

7.5

IBM WebSphere 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、[461 ページの『特記事項』](#)に記載されている情報をお読みください。

本書は、IBM® WebSphere® MQ バージョン 7 リリース 5、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

© Copyright International Business Machines Corporation 2007 年, 2024.

目次

構成	5
UNIX、Linux、および Windows での複数インストールの構成.....	5
複数のインストール環境でのアプリケーションの接続.....	6
プライマリー・インストールの変更.....	16
キュー・マネージャーとインストールの関連付け.....	17
システム上の IBM WebSphere MQ のインストールの見つけ方.....	19
キュー・マネージャーの作成および管理.....	20
デフォルト・キュー・マネージャーの作成.....	23
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.....	24
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.....	25
キュー・マネージャーの開始.....	26
キュー・マネージャーの停止.....	26
キュー・マネージャーの再始動.....	28
キュー・マネージャーの削除.....	28
分散キューイングを使用したアプリケーションの接続.....	29
IBM WebSphere MQ 分散メッセージング技法.....	29
分散キュー管理の概要.....	49
UNIX、Linux、および Windows でのチャンネルのモニターおよび制御.....	74
サーバーとクライアント間の接続の構成.....	99
使用する通信タイプの決定.....	101
拡張トランザクション・クライアントの構成.....	103
MQI チャンネルの定義.....	112
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.....	114
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.....	117
チャンネル出口.....	122
キュー共有グループへのクライアントの接続.....	125
構成ファイルを使用したクライアントの構成.....	127
環境変数を使用したクライアントの構成.....	144
キューに入れられたパブリッシュ/サブスクライブの制御.....	151
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.....	151
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.....	153
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.....	153
ストリームの追加.....	154
ストリームの削除.....	155
サブスクリプション・ポイントの追加.....	156
階層へのキュー・マネージャーの接続.....	157
階層からのキュー・マネージャーの切断.....	158
キュー・マネージャー・クラスターの構成.....	159
アクセス制御と複数のクラスター伝送キュー.....	160
分散キューイングとの比較.....	161
クラスターのコンポーネント.....	164
完全リポジトリを保持するクラスター・キュー・マネージャーの選択方法.....	178
クラスターの編成.....	179
クラスターの命名規則.....	179
クラスターのオーバーラップ.....	180
クラスター化のヒント.....	181
クラスター内での通信の確立.....	183
リポジトリの情報の保存.....	185
IBM WebSphere MQ クラスターの管理.....	186
クラスターとの間のルーティング・メッセージ.....	251
クラスターによるワークロードの管理.....	266
クラスター化: ベスト・プラクティス.....	282

可用性、リカバリー、および再始動.....	314
クライアントの自動再接続.....	315
コンソール・メッセージのモニター.....	321
高可用性の構成.....	321
メッセージの消失を確実に回避する (ログ).....	406
IBM WebSphere MQ キュー・マネージャー・データのバックアップと復元.....	422
構成情報の変更.....	427
UNIX、Linux、および Windows システムでの構成情報の変更.....	427
IBM WebSphere MQ 構成情報を変更するための属性.....	433
キュー・マネージャー構成情報の変更.....	440
構成 HP Integrity NonStop Server.....	457
Gateway プロセスの概要.....	457
Gateway を Pathway のもとで動作するように構成する.....	457
クライアント初期設定ファイルの構成.....	459
チャンネルへのアクセス権付与.....	459
特記事項.....	461
プログラミング・インターフェース情報.....	462
商標.....	462

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

IBM WebSphere 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 WebSphere 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、[20 ページの『キュー・マネージャーの作成および管理』](#)を参照してください。

IBM WebSphere 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、[29 ページの『分散キューイングを使用したアプリケーションの接続』](#)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 WebSphere MQ サーバーに接続する方法については、[99 ページの『クライアントとサーバー間の接続の構成』](#)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、[159 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 WebSphere 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、[427 ページの『IBM WebSphere 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。

関連概念

[WebSphere MQ 技術概要](#)

関連タスク

[ローカル WebSphere MQ オブジェクトの管理](#)

[リモート WebSphere MQ オブジェクトの管理](#)

[計画](#)

UNIX, Linux, and Windows での複数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。

この情報は UNIX, Linux®, and Windows に適用されます。

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。

- [16 ページの『プライマリー・インストールの変更』](#)
- [17 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』](#)
- [6 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』](#)

関連概念

[複数のインストール](#)

関連タスク

[プライマリー・インストールの選択](#)

[インストール名の選択](#)

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 Version 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 WebSphere 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 WebSphere MQ アプリケーションの接続先のキュー・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 WebSphere 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 WebSphere 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。

読み込み

アプリケーションの実行時に、IBM WebSphere 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、[8 ページの『IBM WebSphere MQ Version 7.1 以降の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 WebSphere 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 WebSphere 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 Version 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 WebSphere 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 WebSphere 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。

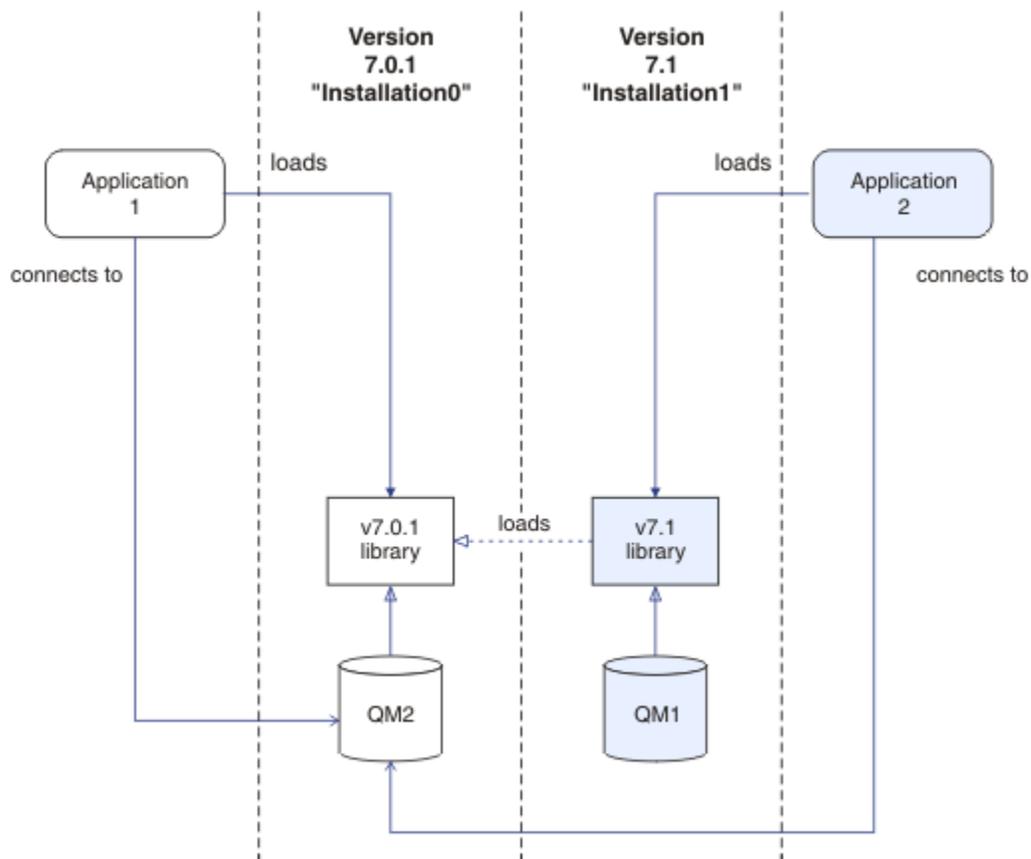


図 1. 複数のインストール環境でのアプリケーションの接続

例えば、7 ページの図 1 は、バージョン 7.0.1 のインストール済み環境 (Installation0) とバージョン 7.1 のインストール済み環境 (Installation1) 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、バージョン 7.0.1 ライブラリーを直接ロードします。application 1 が QM2 に接続されると、バージョン 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、バージョン 7.0.1 ライブラリーが他のライブラリー・バージョンを読み込めないことが原因で発生します。つまり、バージョン 7.0.1 ライブラリーが直接ロードされた場合は、後続のバージョンの IBM WebSphere 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できません。

Application 2 は、バージョン 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、バージョン 7.1 ライブラリーがバージョン 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、バージョン 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移行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、および 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 Version 7.1 ライブラリーのロード方法の詳細については、8 ページの『[IBM WebSphere MQ Version 7.1 以降の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

以下のいずれかのバージョン 7.1 以降のライブラリーが見つかり、ロードされた場合は、IBM WebSphere MQ が自動的に適切なライブラリーをロードし、使用します。

- 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 WebSphere 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、[12 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

[17 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。

[12 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。

[8 ページの『IBM WebSphere MQ Version 7.1 以降のライブラリーのロード』](#)

IBM WebSphere 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 WebSphere MQ のインストール場所、および IBM WebSphere 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

[プライマリー・インストールの選択](#)

[16 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。

IBM WebSphere MQ Version 7.1 以降のライブラリーのロード

IBM WebSphere 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 WebSphere MQ のインストール場所、および IBM WebSphere 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。

IBM WebSphere MQ Version 7.1 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 Version 7.1 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、アプリケーションで `/usr/lib` のシンボリック・リンクが必要な場合は、バージョン 7.1 のインストールをプライマリー・インストールとして選択するか、手動でシンボリック・リンクを作成する必要があります。
- IBM WebSphere MQ Version 7.1 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 Version 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 WebSphere MQ Version 7.1 以降のライブラリーの場所を探索してロードする方法は、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、[11 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。

オペレーティング・システムによってロードされる IBM WebSphere MQ ライブラリーにキュー・マネージャーが関連付けられるようにすることが最善です。

IBM WebSphere 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。

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>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。</p> <p>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、WebSphere MQ アプリケーションの作成を参照してください。</p>	<ul style="list-style-type: none">変更範囲が明確です。	<ul style="list-style-type: none">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。IBM WebSphere 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。

表 1. 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><code>setmqenv</code>、または <code>crtmqenv</code> を使用し、<code>-k</code> または <code>-l</code> オプションを指定して、<code>LD_LIBRARY_PATH</code> 環境変数 (AIX® の場合は <code>LIBPATH</code>) を設定する。</p>	<ul style="list-style-type: none"> 既存のアプリケーションを変更する必要がありません。 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。 IBM WebSphere MQ の場所が変わった場合の変数の変更が容易です。 	<ul style="list-style-type: none"> <code>setuid</code> アプリケーションおよび <code>setgid</code> 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、<code>LD_LIBRARY_PATH</code> が無視される場合があります。 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。 <code>LD_LIBRARY_PATH</code> 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。 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、<code>LD_LIBRARY_PATH</code> が使用不可になっている場合があります。詳しくは、HP-UX 用のランタイム・リンクに関する考慮事項 を参照してください。 Linux: アプリケーションの構築時に使用したコンパイラにより、<code>LD_LIBRARY_PATH</code> が使用不可になっている場合があります。詳しくは、Linux 用のランタイム・リンクに関する考慮事項 を参照してください。
Windows システム	<p><code>setmqenv</code> または <code>crtmqenv</code> を使用して、<code>PATH</code> 変数を設定する。</p>	<ul style="list-style-type: none"> 既存のアプリケーションを変更する必要がありません。 IBM WebSphere MQ の場所が変わった場合の変数の変更が容易です。 	<ul style="list-style-type: none"> 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。 他のアプリケーションに影響する可能性があります。

表 1. 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, Linux, and Windows システム	<p>プライマリー・インストールをバージョン 7.1 以降のインストールに設定する。16 ページの『プライマリー・インストールの変更』を参照。</p> <p>プライマリー・インストールについて詳しくは、プライマリー・インストールの選択を参照してください。</p>	<ul style="list-style-type: none"> 既存のアプリケーションを変更する必要がありません。 IBM WebSphere 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。 前のバージョンの IBM WebSphere MQ と同じような動作になります。 	<ul style="list-style-type: none"> WebSphere MQ バージョン 7.0.1 がインストールされている場合は、プライマリー・インストールをバージョン 7.1 以降に設定することはできません。 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。

HP-UX のライブラリーのロードに関する考慮事項

前のバージョンの IBM WebSphere 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 WebSphere 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、[11 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

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。アプリケーションが影響を受けるかどうかは、`readelf -d applicationName` コマンドを使用して判別できます。 `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

前のバージョンの IBM WebSphere 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、`-lmqmc -lmqzse` 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 WebSphere MQ によって自動的にロードされるようになりました。IBM WebSphere 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、`-lmqmc -lmqzse` 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。

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_LIBRARY_PATH` 環境変数 (AIX の場合は `LIBPATH`、HP-UX の場合は `SHLIB_PATH`) を使用。この変数が設定されている場合、その変数は、必要な WebSphere 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 WebSphere 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 Version 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、WebSphere 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: `dump`
- HP-UX: `chatr`
- Linux: `readelf`
- Solaris: `elfdump`

関連概念

[17 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。

[12 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。

[6 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 Version 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 WebSphere 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 WebSphere MQ アプリケーションの接続先のキュー・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のライブラリーを使用します。

関連タスク

[プライマリー・インストールの選択](#)

[16 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。

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 WebSphere 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® for 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 Version 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 WebSphere MQ ライブラリーを指定できず、IBM WebSphere 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とコンパイル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 Version 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 WebSphere 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。接続しようと試みると、エラーが発生し、理由コード MQRC_INSTALLATION_MISMATCH が返されます。
2. アプリケーションが IBM WebSphere 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 が設定されます。
3. Version 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. Version 7.0.1 のインストールに関連付けられているキュー・マネージャーとの接続と、Version 7.1 (以上) のインストールに関連付けられているキュー・マネージャーとのファスト・パス接続とを組み合わせることはできません。

AMQ_SINGLE_INSTALLATION が設定されていると、キュー・マネージャーへの任意の接続をファスト・パス接続にすることができます。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 WebSphere MQ ランタイム・ライブラリーをロードするの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。異なるインストール済み環境に関連付けられているキュー・マネージャーに接続しようと試みると、その接続は理由コード MQRC_INSTALLATION_MISMATCH で失敗します。AMQ_SINGLE_INSTALLATION が設定されていると、この制限はファスト・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。

関連資料

[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](#)

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 アセンブリとポリシー・ファイルはグローバル・アセンブリ・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 WebSphere 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、[14 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、[15 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。

	下位バージョンの IBM WebSphere MQ でコンパイルされたアプリケーション	より高いバージョンの IBM WebSphere 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 WebSphere 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 2. 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)

	下位バージョンの IBM WebSphere MQ でコンパイルされたアプリケーション	より高いバージョンの IBM WebSphere 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 WebSphere 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"> • <bindingRedirect> タグは、GAC 内にある古いバージョンのアセンブリーの使用を示すために使用します。
新しいバージョンの IBM WebSphere 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"> • <codebase> タグを使用して、上位バージョンのアセンブリーの場所を指定します。 • <bindingRedirect> タグを使用して、上位バージョンのアセンブリーの使用を示します。 	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"> • <codebase> タグを使用して、上位バージョンのアセンブリーの場所を指定します。
古いバージョンの IBM WebSphere 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"> • <codebase> タグを使用して、下位バージョンのアセンブリーの場所を指定します。 • タグ <publisherpolicy Apply=no> を組み込む 	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"> • <codebase> タグを使用して、下位バージョンのアセンブリーの場所を指定します。 • <bindingRedirect> タグを使用して、古いバージョンのアセンブリーの使用を示します。 • タグ <publisherpolicy Apply=no> を組み込む

サンプル・アプリケーション構成ファイル NonPrimaryRedirect.config は、MQ_INSTALLATION_PATH\tools\dotnet\samples\base フォルダーに入っています。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 WebSphere MQ インストール・パスを使用して変更できます。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組み込むこともできます。nmqsget.exe.config および nmqsput.exe.config のサンプルが用意されています。どちらのサンプルも <linkedConfiguration> タグを使用し、NonPrimaryRedirect.config ファイルを組み込みます。

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

6 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 Version 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 WebSphere 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 WebSphere MQ アプリケーションの接続先のキュー・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のライブラリーを使用します。

[複数のインストール](#)

関連タスク

[プライマリー・インストールの選択](#)

[.NET の使用](#)

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 Version 7.1 以降のインストールが IBM WebSphere MQ Version 7.0.1 のインストールと共存している場合は、IBM WebSphere MQ Version 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 Version 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 Version 7.1 以降をプライマリーにすることはできません。

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。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。

16 ページの『[プライマリー・インストールの設定](#)』。

17 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 Version 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 Version 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、[17 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 Version 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. UNIX and Linux システムの root、または Windows システムの Administrators グループのメンバーとして、以下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、以下のようにします。

```
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、以下のようにします。

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -n installationName
```

4. 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、以下のようにします。

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 Version 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 Version 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. UNIX and Linux システムの root、または Windows システムの Administrators グループのメンバーとして、以下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、以下のようにします。

```
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、以下のようにします。

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -n installationName
```

関連概念

[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 および 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のリンク](#)

関連タスク

[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](#)

関連資料

[setmqinst](#)

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。

`setmqm` コマンドは以下の方法で使用できます。

- WebSphere 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 WebSphere MQ から新しいバージョンの WebSphere 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ないさまざまな影響があります。マイグレーションについては、[WebSphere MQ の移行およびアップグレード](#)を参照してください。

キュー・マネージャーをインストールに関連付けるには、以下のようにします。

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。

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、V7.0.1 またはそれよりも前のキュー・マネージャーが初めて V7.1 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済み環境に関連付けられているかに関する情報を表示します。

HA 環境におけるキュー・マネージャーの関連付け

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。

削除された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

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

19 ページの『システム上の IBM WebSphere MQ のインストールの見つけ方』

システムに複数の IBM WebSphere 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。

429 ページの『IBM WebSphere MQ 構成ファイル mqs.ini』

IBM WebSphere 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。

関連タスク

[プライマリー・インストールの選択](#)

関連資料

[setmqm](#)

[strmqm](#)

[dspmq](#)

[dspmqinst](#)

システム上の IBM WebSphere MQ のインストールの見つけ方

システムに複数の IBM WebSphere 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 WebSphere 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** コマンドを使用します。このコマンドが Version 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 WebSphere 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、Version 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 WebSphere 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 WebSphere 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、Version 7.1 以降のインストールから **dspmqver** 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、Version 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。

関連概念

[複数のインストール](#)

関連資料

[dspmqver](#)

[dspmqinst](#)

キュー・マネージャーの作成および管理

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・マネージャーの作成

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 WebSphere 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。

キュー・マネージャーを作成するには、IBM WebSphere MQ 制御コマンド **crtmqm** を使用します (**crtmqm** で説明)。**crtmqm** コマンドは、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 ([システム・デフォルト・オブジェクト](#)を参照) を自動的に作成します。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。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、**stmqm** 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。

注: IBM WebSphere 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 WebSphere 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。

On

キュー・マネージャーを作成する前に、いくつかの考慮すべき点があります (特に、実稼働環境において)。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

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 および Windows システムで IBM WebSphere 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。

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 である場合、キュー・マネージャーに ACCOUNTS.SATURN.QUEUE.MANAGER という名前を付けることができます。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 WebSphere 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。この方法の具体例は示しません。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 および JUPITER.queue.manager という名前の2つのキュー・マネージャーを作成することが可能です。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の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 WebSphere 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、23 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、314 ページの『[可用性、リカバリー、および再始動](#)』を参照してください。

IBM WebSphere MQ for UNIX システムの場合のみ

`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>` を作成することができます。 `crtmqm` を使用するとき、 `/var/mqm/qmgrs/<qmgr>` ディレクトリーが存在し、空であり、mqm によって所有されている場合は、そのディレクトリーがキュー・マネージャー・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、第1障害サポート・テクノロジー (FFST) (First Failure Support Technology (FFST)) メッセージでその作成が失敗します。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。

関連概念

5 ページの『[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。

25 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 WebSphere MQ 構成情報は、Windows、UNIX and Linux システム上の構成ファイルに保管されています。

26 ページの『[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。

26 ページの『[キュー・マネージャーの停止](#)』

キュー・マネージャーを停止する3つの方法、つまり静止シャットダウン、即時シャットダウン、およびプリエンブティブ・シャットダウンがあります。

28 ページの『[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。IBM WebSphere MQ for Windows および IBM WebSphere MQ for Linux (x86 および x86-64 プラットフォーム) システムの場合は、IBM WebSphere MQ エクスプローラーからキュー・マネージャーを再始動できます。

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

IBM WebSphere 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します。

[システムおよびデフォルト・オブジェクト](#)

関連タスク

24 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](#)』

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ことができます。これを行う方法は、使用しているプラットフォームによって異なります。

28 ページの『[キュー・マネージャーの削除](#)』

dltmqm コマンドを使用するか、または WebSphere MQ エクスプローラーを使用して、キュー・マネージャーを削除できます。

distributed 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 WebSphere 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、20 ページの『[キュー・マネージャーの作成および管理](#)』で説明されている考慮事項を参照してください。

UNIX

crtmqm を使用して UNIX and Linux 上にキュー・マネージャーを作成するとき、`/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、空である場合は、そのディレクトリーがキュー・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してい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 WebSphere MQ では、デフォルト伝送キューは自動的には作成されません。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 WebSphere 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。

関連概念

[25 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 WebSphere MQ 構成情報は、Windows、UNIX and Linux システム上の構成ファイルに保管されています。

[キュー・マネージャーの処理](#)

[ローカル・キューの処理](#)

関連資料

[システムおよびデフォルト・オブジェクト](#)

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

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ことができます。これを行う方法は、使用しているプラットフォームによって異なります。

WebSphere MQ for Windows および WebSphere MQ for Linux (x86 および x86-64 プラットフォーム) システム

このタスクについて

WebSphere MQ for Windows および WebSphere MQ for Linux (x86 および x86-64 プラットフォーム) システムで、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、以下の手順を使用します。

手順

1. IBM WebSphere MQ エクスプローラーを開きます。

2. IBM WebSphere MQ を右クリックし、Properties... を選択します。「WebSphere MQ のプロパティ」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。

UNIX and Linux システム

このタスクについて

デフォルト・キュー・マネージャーを作成するときに、デフォルト・キュー・マネージャーの名前が WebSphere MQ 構成ファイル (mqqs.ini) の中の 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、スタンザが存在していない場合に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、mqqs.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。

次のタスク

構成ファイルについては、[427 ページの『IBM WebSphere 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。

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 WebSphere MQ 構成情報は、Windows、UNIX and Linux システム上の構成ファイルに保管されています。

Windows および Linux (x86 および x86-64) システム上で、IBM WebSphere MQ Explorer を使用して、構成ファイルに変更を行います。

Windows システムでは、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。[amqmdain](#) を参照してください。

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 WebSphere MQ 構成ファイル (mqqs.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 mqqs.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。

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。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。

構成ファイルの詳細については、[427 ページの『IBM WebSphere 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。

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。

キュー・マネージャーを開始するには、**strmqm** コマンドを使用します。

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。

例えば、キュー・マネージャー QMB を開始するには、以下のコマンドを入力します。

```
strmqm QMB
```

WebSphere MQ for Windows および WebSphere MQ for Linux (x86 および x86-64 プラットフォーム) システムでは、以下のようにしてキュー・マネージャーを開始できます。

1. IBM WebSphere MQ エクスプローラーを開きます。
2. 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。
3. **Start** をクリックします。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 WebSphere MQ から断続的に発行されます。

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。

キュー・マネージャーを自動的に開始する

WebSphere MQ for Windows では、IBM WebSphere MQ エクスプローラーを使用してシステムを始動すると、キュー・マネージャーを自動的に開始できます。詳しくは、[IBM WebSphere MQ Explorer による管理](#)を参照してください。

キュー・マネージャーの停止

キュー・マネージャーを停止する 3 つの方法、つまり静止シャットダウン、即時シャットダウン、およびプリエンプティブ・シャットダウンがあります。

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。 **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。

例えば、QMB という名前のキュー・マネージャーを停止するには、次のコマンドを入力します。

```
endmqm QMB
```

WebSphere MQ for Windows および WebSphere MQ for Linux (x86 および x86-64 プラットフォーム) システムでは、以下のようにしてキュー・マネージャーを停止できます。

1. IBM WebSphere MQ エクスプローラーを開きます。
2. 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。
3. **Stop...** 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。
4. 「制御」または「即時」を選択します。
5. **OK** をクリックします。キュー・マネージャーが停止します。

静的シャットダウン

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。これは完了するまでに多少の時間がかかります。静的シャットダウンは、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です。

このタイプのシャットダウンを使用して、アプリケーションに対して停止するよう通知します。次のコマンドを発行する場合:

```
endmqm -c QMB
```

すべてのアプリケーションが停止されたのがどの時点かは通知されません。(endmqm -c QMB コマンドは、endmqm QMB コマンドと同等です。)

ただし、次のコマンドを発行する場合:

```
endmqm -w QMB
```

すべてのアプリケーションが停止し、キュー・マネージャーが終了するまで、コマンドは待機します。

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。

即時シャットダウンの場合、次のように入力します。

```
endmqm -i QMB
```

プリエンपティブ・シャットダウン

注: この方法は、**endmqm** コマンドを使用してもキュー・マネージャーを停止できなかったときにのみ使用してください。この方法は、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。

即時シャットダウンが機能しない場合は、**-p** フラグを指定したプリエンプティブ・シャットダウンを用いる必要があります。以下に例を示します。

```
endmqm -p QMB
```

これにより、キュー・マネージャーが即時に停止します。この方法でも解決しない場合には、別の方法として手動によるキュー・マネージャーの停止を参照してください。

endmqm コマンドとそのオプションの詳細については、[endmqm](#) を参照してください。

キュー・マネージャーのシャットダウンに問題がある場合

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フラグを付加します。

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。IBM WebSphere MQ for Windows および IBM WebSphere MQ for Linux (x86 および x86-64 プラットフォーム) システムの場合は、IBM WebSphere MQ エクスプローラーからキュー・マネージャーを再始動できます。

キュー・マネージャーを再始動するには、次のように入力します。

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

IBM WebSphere MQ for Windows および IBM WebSphere MQ for Linux (x86 および x86-64 プラットフォーム) システムでは、キュー・マネージャーを開始するのと同じ方法で、以下のようにしてキュー・マネージャーを再始動できます。

1. IBM WebSphere MQ エクスプローラーを開きます。
2. 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。
3. Start をクリックします。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 WebSphere MQ から断続的に発行されます。

キュー・マネージャーの削除

dltmqm コマンドを使用するか、または WebSphere MQ エクスプローラーを使用して、キュー・マネージャーを削除できます。

始める前に

キュー・マネージャーを停止させます。

手順

- 次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。 **dspm -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。

キュー・マネージャーの削除手順

このタスクについて

WebSphere MQ for Windows および WebSphere MQ for Linux (x86 および x86-64 プラットフォーム) システムでは、以下のようにしてキュー・マネージャーを削除できます。

手順

1. WebSphere MQ エクスプローラーを開きます。
2. 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。
3. 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。
 - a) キュー・マネージャーを右クリックします。
 - b) 「停止」 をクリックします。
4. キュー・マネージャーを右クリックします。
5. 「削除 (Delete)」 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。



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、考えを変えるプロンプトは表示されません。Enter キーを押すと、すべての関連リソースが失われます。
- WebSphere MQ for 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 (26 ページの『[キュー・マネージャーの開始](#)』を参照) から除去されます。コマンドが完了すると、WebSphere 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。詳細は、`dltmqm` の説明にある注を参照してください。

`dltmqm` コマンドとそのオプションの詳細については、[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、[Windows](#)、[UNIX and Linux](#) システムでのセキュリティのセットアップを参照してください。)

分散キューイングを使用したアプリケーションの接続

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、WebSphere 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。

このセクションをお読みにするには、チャンネル、キュー、および[相互通信の概念](#)で紹介されているその他の概念についての知識があると役に立ちます。

以下のリン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。

- 51 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](#)』
- 68 ページの『[チャンネルのトリガー操作](#)』
- 66 ページの『[メッセージの安全性](#)』
- 29 ページの『[IBM WebSphere MQ 分散メッセージング技法](#)』
- 49 ページの『[分散キュー管理の概要](#)』
- [Windows](#) [UNIX](#) [Linux](#) 74 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](#)』

関連概念

99 ページの『[クライアントとサーバー間の接続の構成](#)』

WebSphere MQ MQI クライアント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。

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

IBM WebSphere 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します。

関連タスク

159 ページの『[キュー・マネージャー・クラスターの構成](#)』

クラスターの動作とクラスター構成の設計について調べるには、および単純なクラスターのセットアップ方法の例については、このトピックにあるリンクを使用してください。

IBM WebSphere MQ 分散メッセージング技法

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。

- [分散プラットフォームにおけるメッセージ・チャンネルの計画例](#)

関連概念

29 ページの『[分散キューイングを使用したアプリケーションの接続](#)』

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、WebSphere 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。

[チャンネル](#)

[メッセージ・キューイングの概要](#)

[相互通信の概念](#)

関連資料

[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・チャン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、29 ページの『[分散キューイングを使用したアプリケーションの接続](#)』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、30 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せん。

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・チャンネル
- 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・オブジェクトについては、[オブジェクト](#)で説明されています。メッセージ・チャンネルについては、[分散キューイング・コンポーネント](#)で説明されています。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。[相互通信の概念](#)を参照してください。

関連概念

31 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。

31 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3つの異なる方法で使用されます。[31 ページの表 3](#) 以下の3つの方法のそれぞれを定義する方法について説明します。

- 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。

使用法	キュー・マネージャー名	キュー名	伝送キュー名
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	-

表 3. リモート・キュー定義オブジェクトの 3 つの使用方法 (続き)			
使用法	キュー・マネージャ名	キュー名	伝送キュー名
定義で指定	required	required	オプション
2. キュー・マネージャ別名 (OPEN 呼び出しで)			
呼び出しで指定	(*) 必須かつ非ローカル QM	required	-
定義で指定	required	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	-
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。32 ページの図 2 は、'QA_norm' という名前の論理キューにメッセージを入れるアプリケーションを示しています。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・マネージャに送ります。

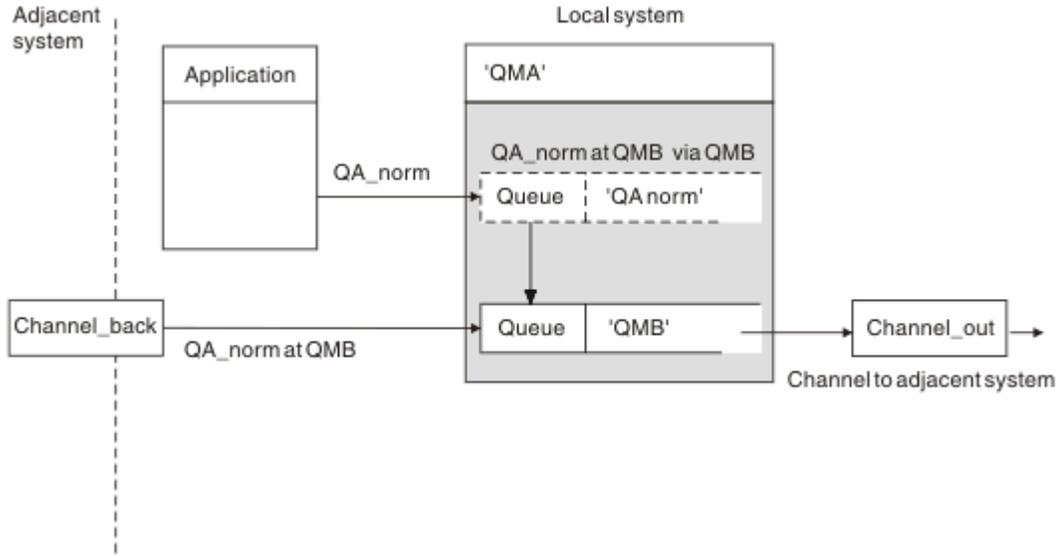


図 2. リモート・キュー定義が使用されて、キュー名が解決され、隣接するキュー・マネージャに通じる伝送キューが得られます

WebSphere 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。詳しくは、[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。

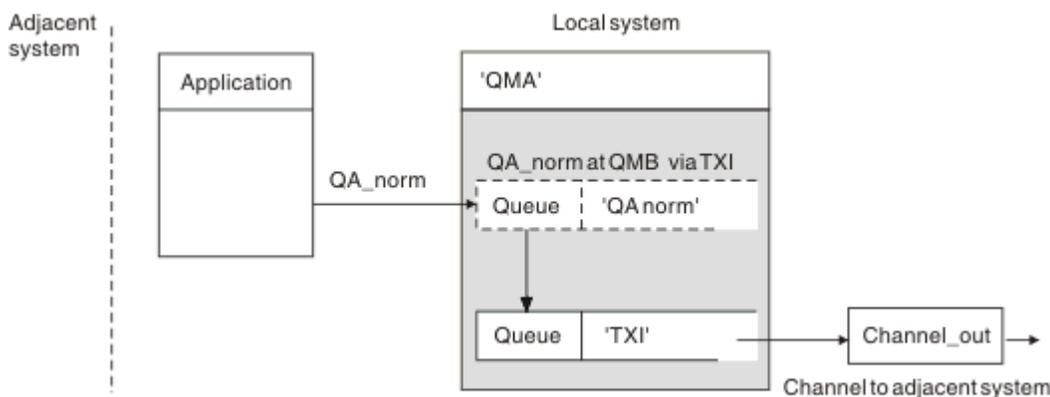


図 3. リモート・キュー定義による異なる伝送キューの使用可能化

分散キューイング環境では、メッセージ・フローをあるチャンネルから別のチャンネルに変更する必要がある場合には、32 ページの『リモート・キューへのメッセージの書き込み』の 32 ページの図 2 と同じシステム構成を使用してください。このトピックの 33 ページの図 3 はリモート・キュー定義を使用し、別の伝送キューを介して (したがってまた、別のチャンネルを介して) 同じ隣接キュー・マネージャーへメッセージを送る方法を示しています。

33 ページの図 3 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。詳しくは、166 ページの『クラスター・キュー』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。

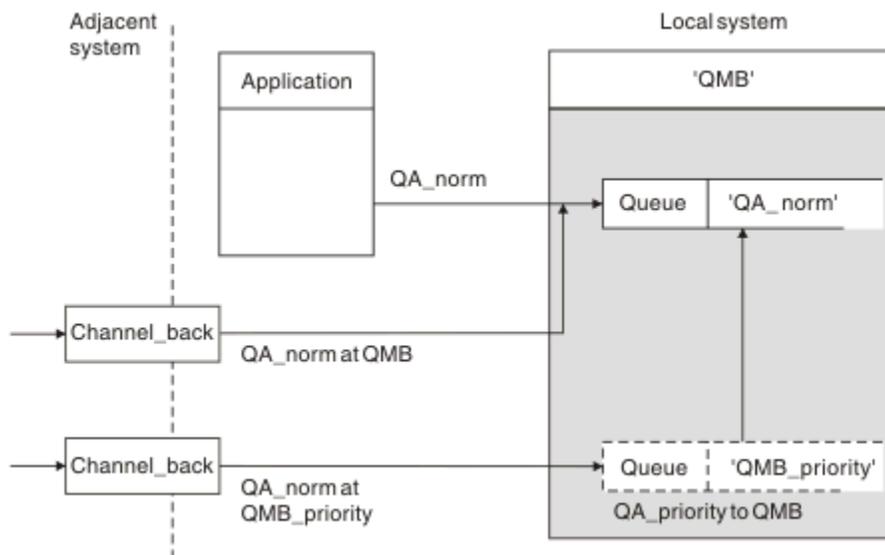


図 4. メッセージの直接受信によるキュー・マネージャー別名の解決

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 [34 ページの図 4](#) 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。

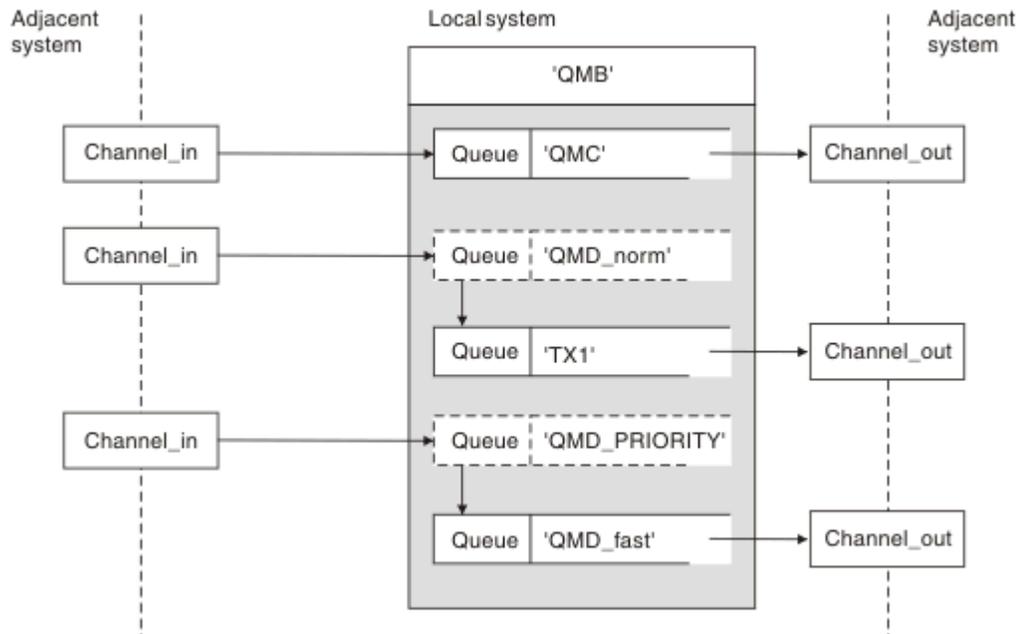


図 5. メッセージが運用システムを通過する 3 つの方法

34 ページの『メッセージの受信』の 34 ページの図 4 に示されている手法は、別名フローがどのように取り込まれるかを示しています。35 ページの図 5 前述の手法をまとめることによってネットワークを構築す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。通常は、単一の伝送キュー (SYSTEM.CLUSTER.TRANSMIT.QUEUE) はすべてのメッセージを、そのキュー・マネージャーがメンバーとなっているすべてのクラスター内のすべてのキュー・マネージャーに転送します。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ることがあります。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。

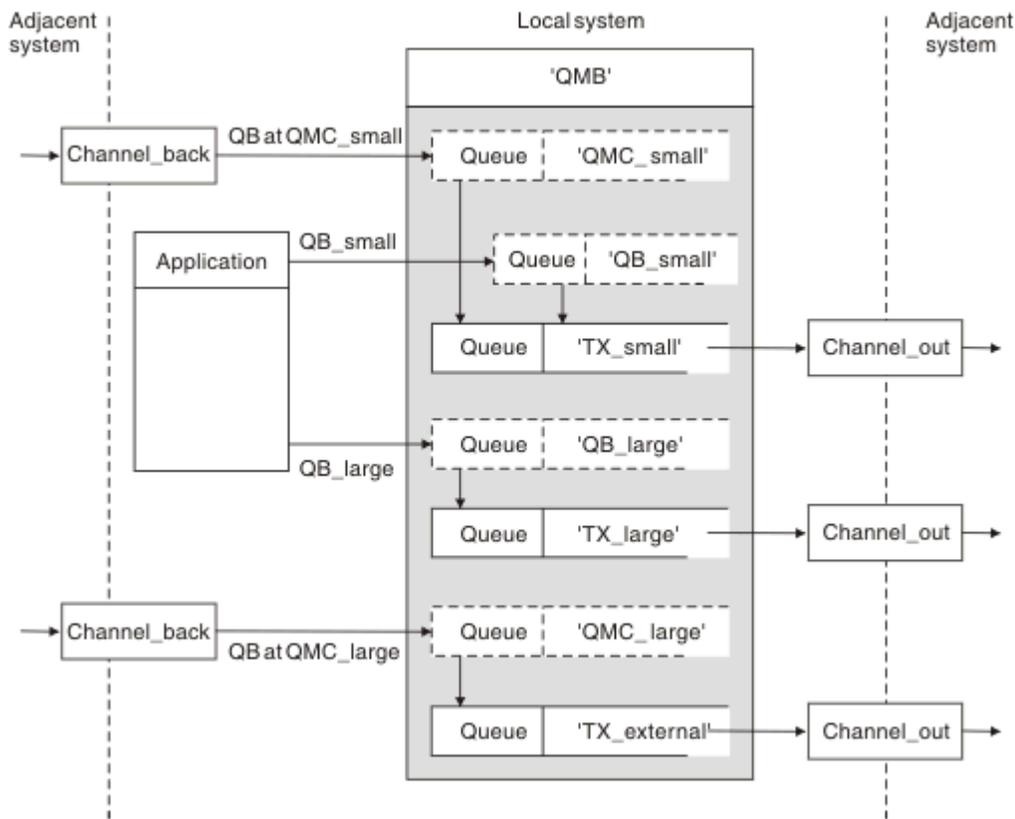


図 6. メッセージ・フローの分離

37 ページの図 6 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てします。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。

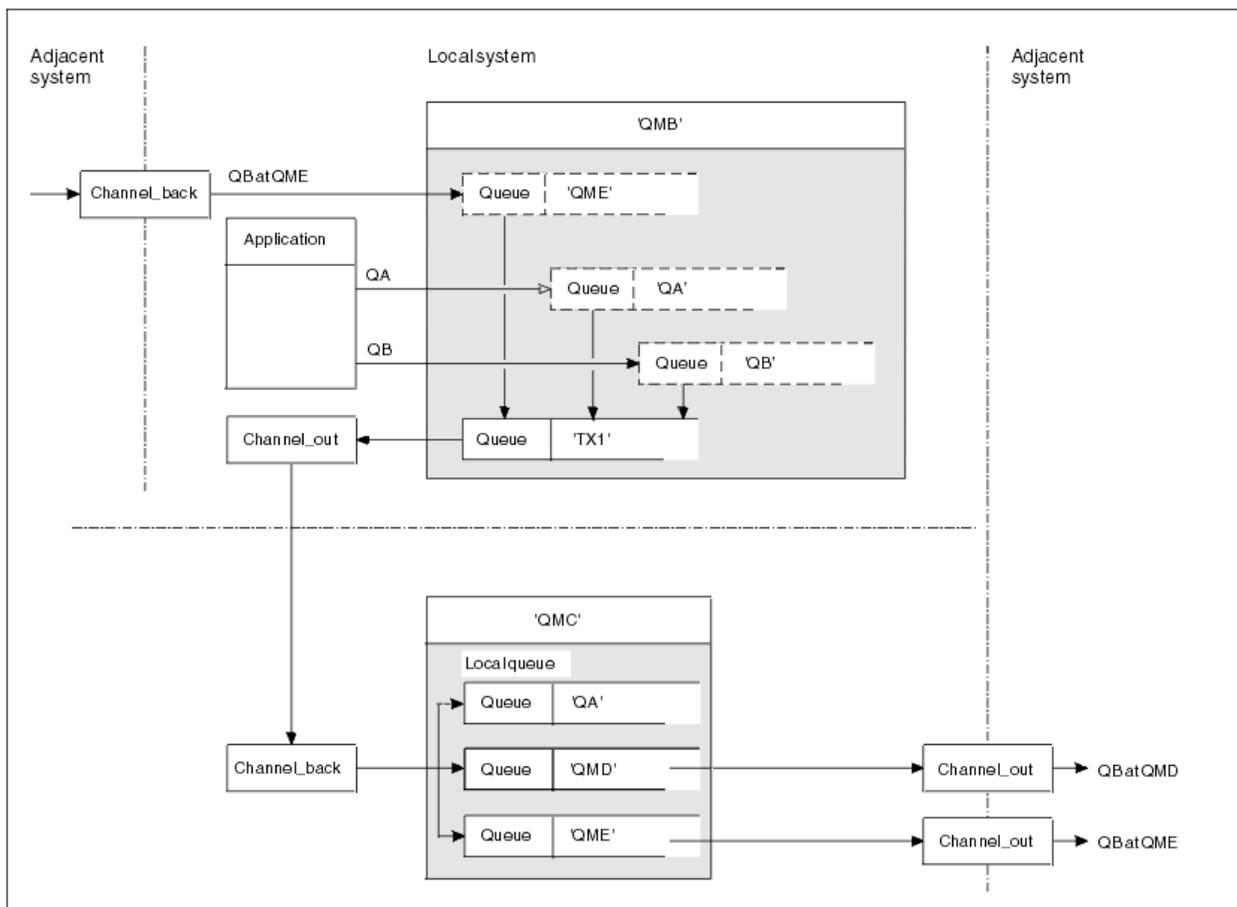


図 7. チャンネルへのメッセージ・フローの結合

38 ページの図 7 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。可能な使用方法には次の 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。

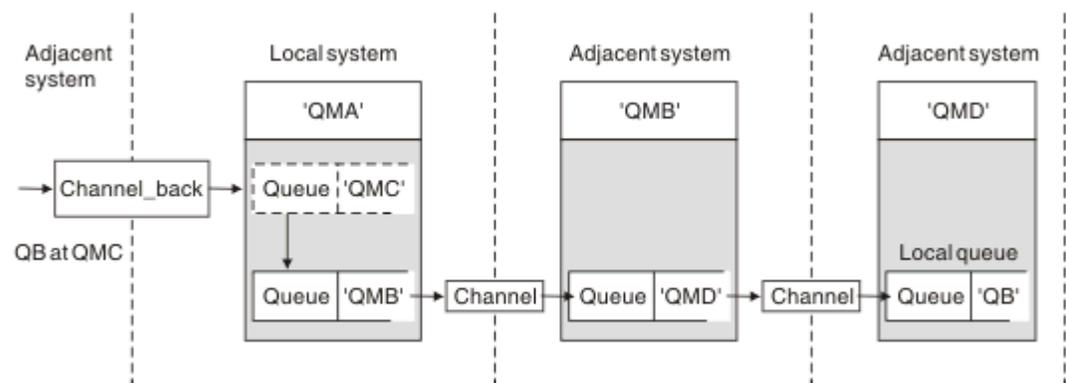


図 8. 別の宛先へのメッセージ・ストリームの転送

39 ページの図 8 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 に通じる関連チャンネル定義

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、262 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®以外のすべてのプラットフォームの WebSphere MQ では、アプリケーションは 1 回の 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、完全なリモート・キュー処理ループを作成できます。

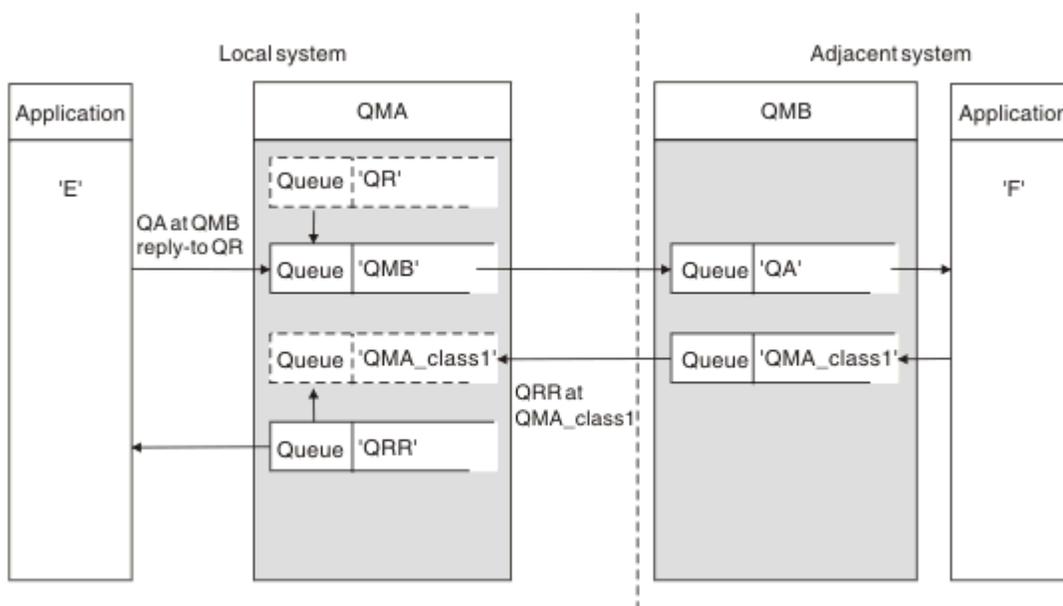


図 9. 書き込み (PUT) 呼び出し時の応答先キュー名の置換

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 40 ページの図 9 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、47 ページの表 7 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

[31 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。

[42 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。

[43 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。

[44 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ウォークスルーです。

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。

42 ページの図 10 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。

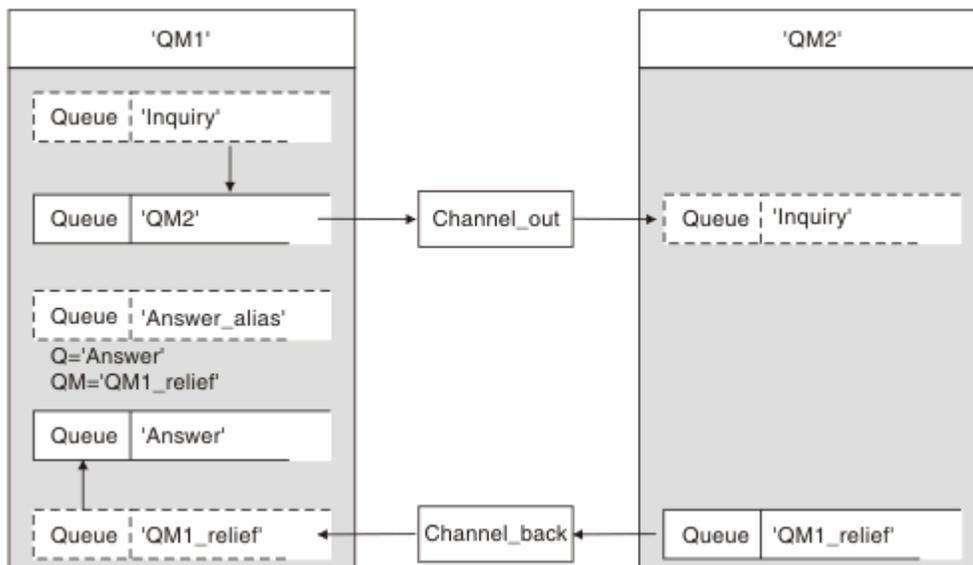


図 10. 応答先キュー別名の例

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 WebSphere 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
	伝送キュー名	QM2 (DEFAULT)
キュー・マネージャー別名	オブジェクト名	QM1_relief *

オブジェクト	定義	
	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*
	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WebSphere 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。

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ウォークスルーです。

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。

応答先キュー名

Answer_alias

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。

応答先キュー名

Answer

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。

ネットワークの考慮事項

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。

クラスター化環境でのこの動作については、製品資料の[クラスターが機能する仕組み](#)セクションの該当するトピック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に、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せん。

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 [46 ページの表 5](#)、[47 ページの表 6](#)、および [47 ページの表 7](#) 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)

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	TX1	リモート・キュー
QB	QMC	QB	QMD	QMB	キュー・マネージャー別名

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・マネージャー別名
(任意)	QMD_PRIORITY	(任意)	QMD_PRIORITY	QMD_fast	キュー・マネージャー別名
(任意)	QMC_small	(任意)	QMC_small	TX_small	キュー・マネージャー別名
(任意)	QMC_large	(任意)	QMC_large	TX_external	キュー・マネージャー別名
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 QMGR	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

メッセージの送達、重複なしの送達、および伝送キューから取り出されたとおりの順序での保管を確実に行うためには、チャンネル・メッセージ・シーケンス番号付けを使います。

シーケンス番号は、チャンネルの送信側で生成され、使用の前に1だけ増やされます。つまり、現在のシーケンス番号は最後に送信されたメッセージの番号であるということです。この情報は、`DISPLAY CHSTATUS (MQSC リファレンスを参照)` を使用して表示することができます。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は、別のマシンに実際にリンクすることなく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に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。

WebSphere 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。

-  [74 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』](#)

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。

-  [分散プラットフォーム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

[50 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際のエンティティ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。

[54 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。

[66 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。

[70 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 WebSphere MQ プラットフォームによって異なります。

[71 ページの『メッセージのデータ変換』](#)

WebSphere 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。

[72 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』](#)

WebSphere 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。

72 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 WebSphere 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、ならびにトラステッド・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。

関連資料

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。

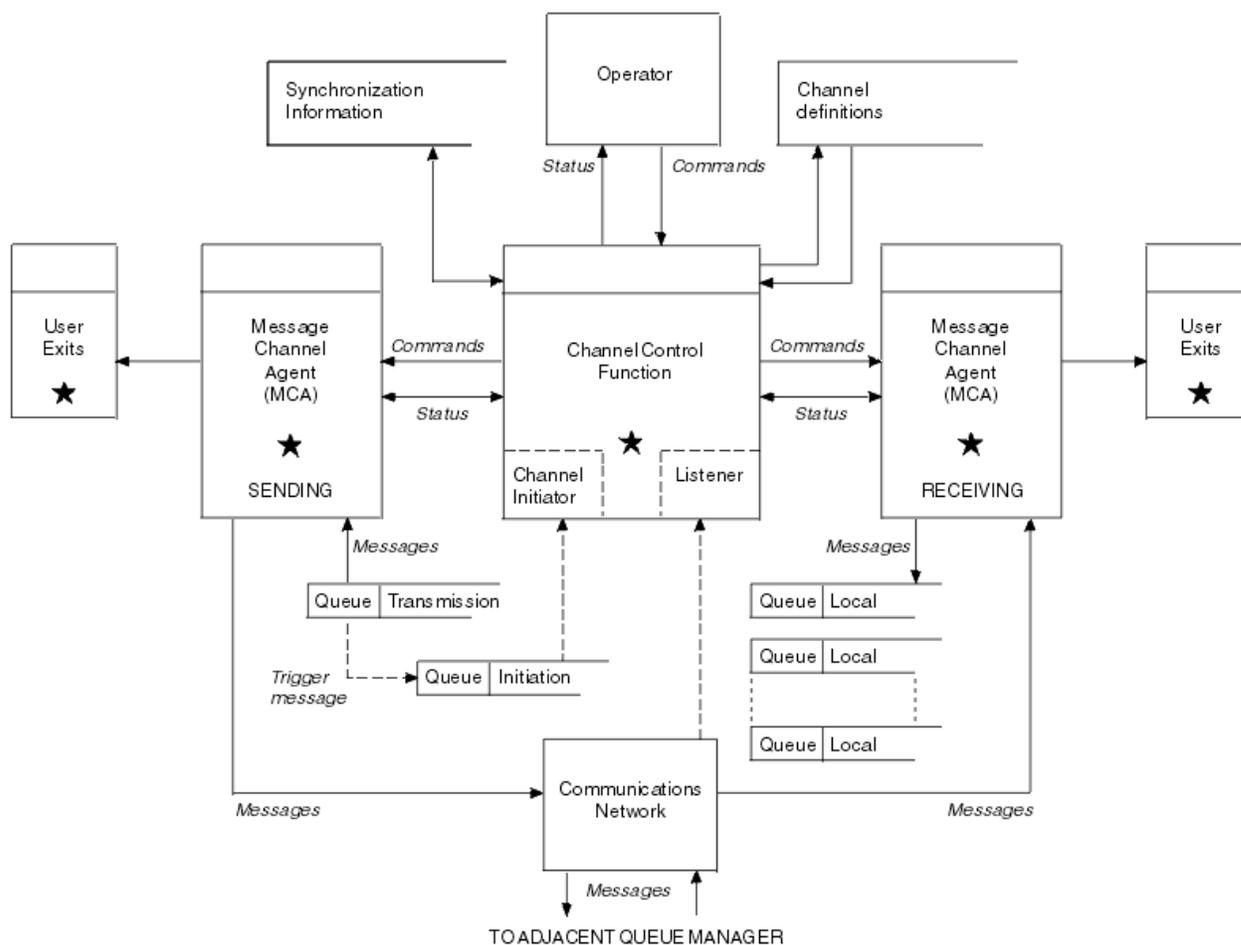


図 11. 分散キュー管理モデル

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 WebSphere MQ セキュリティ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、次のようにします。

- 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 (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 WebSphere 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で、WebSphere MQ スクリプト・コマンド (MQSC) を使用できます。これについては、[MQSC コマンド 管理タスクの自動化](#) で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 WebSphere MQ エクスプローラーで説明されています。

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

[20 ページの『キュー・マネージャーの作成および管理』](#)

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。

[29 ページの『IBM WebSphere MQ 分散メッセージング技法』](#)

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。

[49 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。

[68 ページの『チャンネルのトリガー操作』](#)

WebSphere 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。

[66 ページの『メッセージの安全性』](#)

WebSphere 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。

[74 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 WebSphere 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。

[99 ページの『クライアントとサーバー間の接続の構成』](#)

WebSphere MQ MQI クライアント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。

関連タスク

[159 ページの『キュー・マネージャー・クラスターの構成』](#)

クラスターの動作とクラスター構成の設計について調べるには、および単純なクラスターのセットアップ方法の例については、このトピックにあるリンクを使用してください。

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)。WebSphere 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については、[チャンネル属性](#) を参照してください。

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)。WebSphere 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。

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。

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。

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。

伝送キュー名

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 では必須で、他のプラットフォームではオプションです。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。(他のプラットフォームの場合)。

- 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。

- 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

54 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。

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、WebSphere 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。

チャンネルの開始

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ンネルが開始されるまで伝送キューに保管されます。チャン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ンネルを開始するには、START CHANNEL コマンドを使用します。送信側チャンネルを開始すると、受信側チャンネルは(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 WebSphere MQ コマンド (MQSC) を使用して、[74 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 WebSphere 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 文字のブランクが続く「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 WebSphere 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、[29 ページの『分散キューイングを使用したアプリケーションの接続』](#)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、[112 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。詳しくは、[64 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(CONNAME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、メッセージを伝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。

[57 ページの図 12](#) 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。

[58 ページの図 13](#) 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。

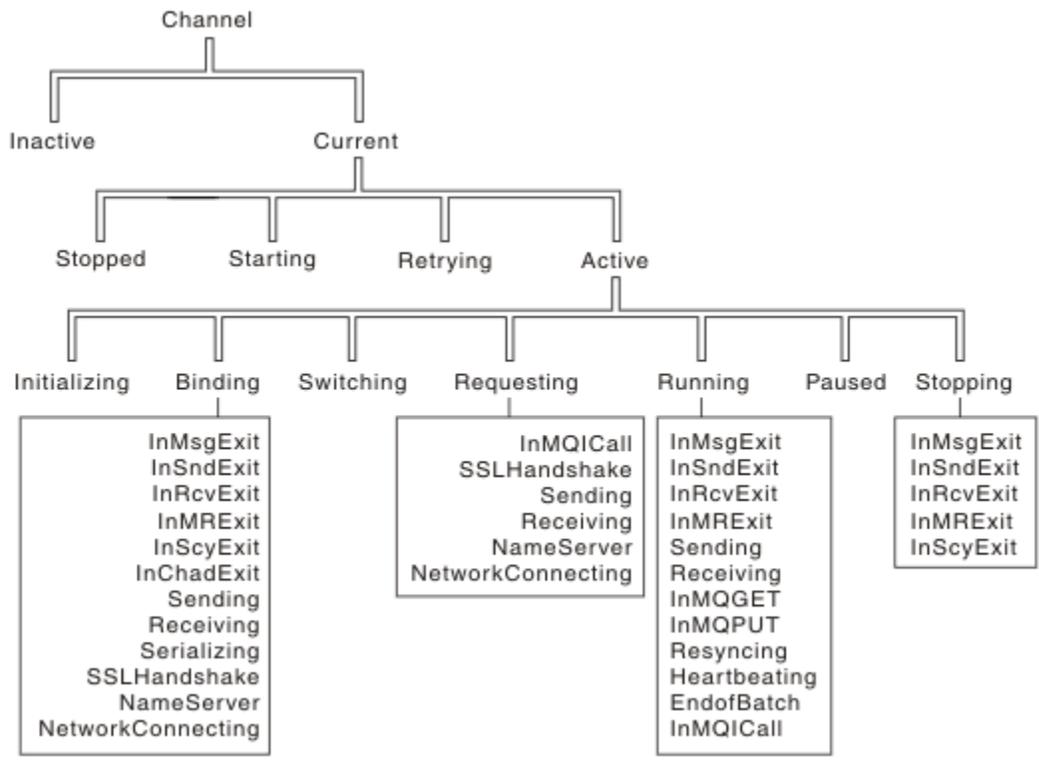


図 12. チャネル状態および副状態

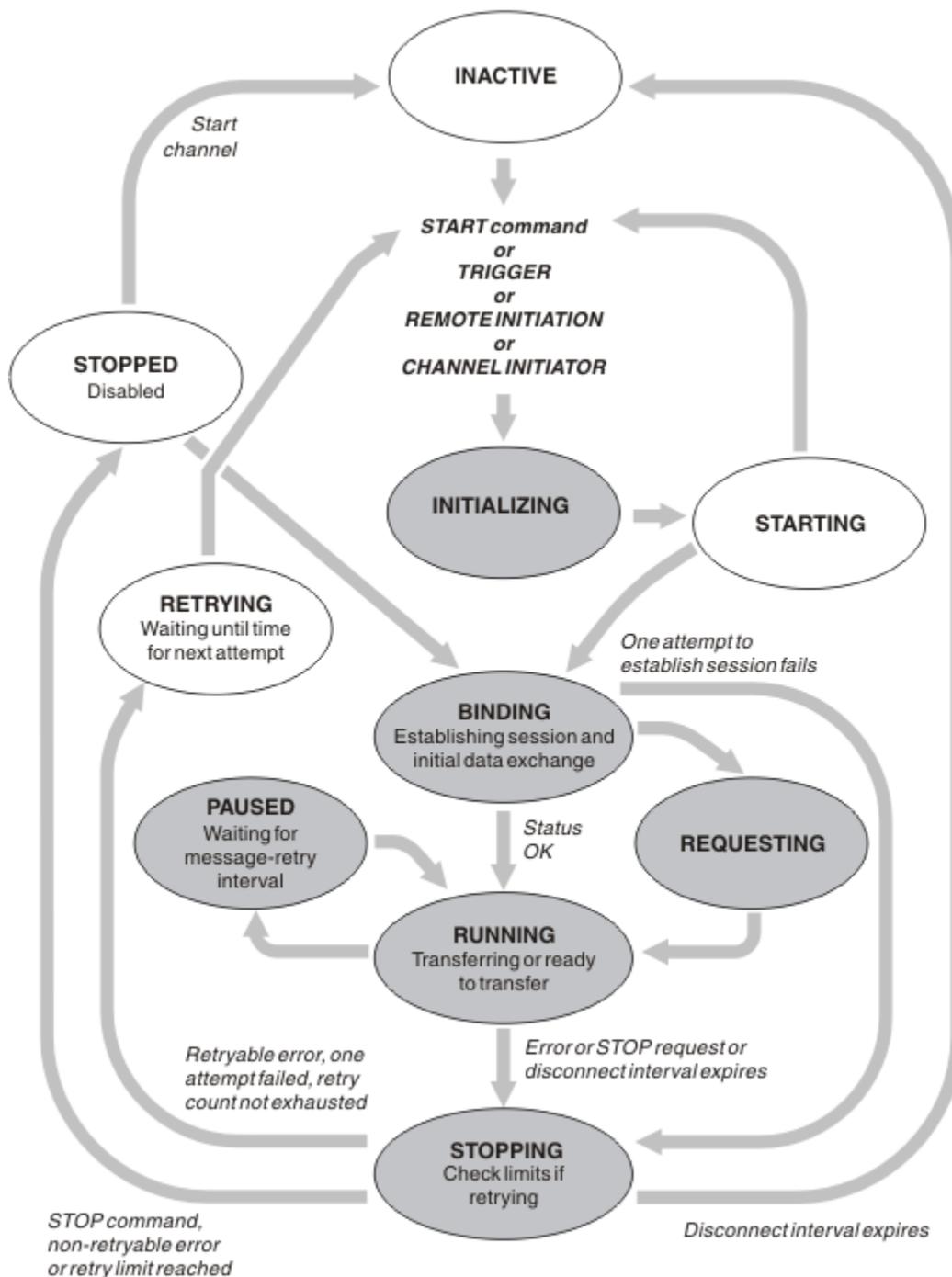


図 13. チャネル状態間のフロー

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 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。58 ページの図 13 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、57 ページの図 12 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。

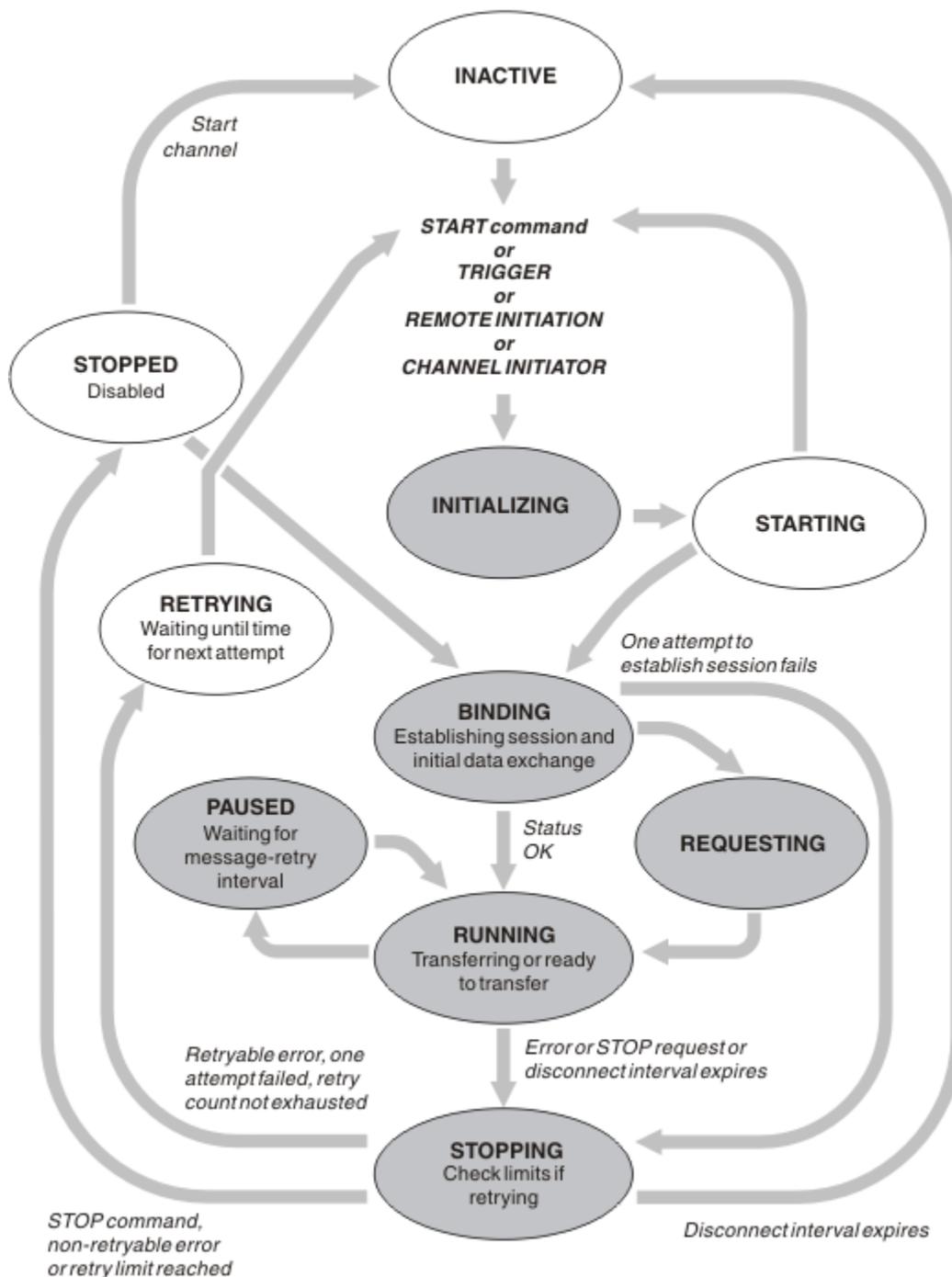


図 14. チャンネル状態間のフロー

注:

1. チャンネルは、59 ページの図 14 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。

現行チャネルの最大数の指定

一度に現行チャネルにしてもよいチャネルの最大数を指定することができます。この数は、チャネル状況テーブルに項目があるチャネルの数で、これには再試行中のチャネルと停止中のチャネルが含まれます。これを指定するには、UNIX and Linux システムのキュー・マネージャー構成ファイル、または WebSphere MQ エクスプローラーを使用します。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anzas](#)を参照してください。チャネルの最大数の指定について詳しくは、「[Administering IBM WebSphere MQ for WebSphere MQ for UNIX and Linux systems, and Windows systems](#)」の管理。

注：

1. この数にサーバー接続チャネルが含まれています。
2. アクティブのチャネルは現行チャネルであることが必要です。開始したチャネルが、現行チャネルにならない場合、開始は失敗します。

アクティブのチャネルの最大数の指定

アクティブ・チャネルの最大数を指定して、多くの開始チャ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、他のチャネルが終了したらすぐに待機中のチャネルを開始できるようにします。

再試行するチャ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ネルは現行のままになります。チャ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、[チャネル属性](#)を参照してください。

チャネルがアクティブのチャ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ネル・インスタンスによって解放されるまで、そのチャネルは待機します。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。

要求側チャネル以外のチャ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。

要求側チャネルは開始状態にはなりません。アクティブにあるチャネルの数がすでに上限に達しているために要求側チャネルを開始できない場合、そのチャネルは異常終了します。

要求側チャネル以外のチャ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、そのチャネルは停止状態になります。チャネル停止イベントが発生します。

アクティブ・チャネルの最大数にサーバー接続チャネルが含まれています。

アクティブ・チャネルの最大数の指定について詳しくは、「[Administering IBM WebSphere MQ for WebSphere MQ for UNIX and Linux systems, and Windows systems](#)」の管理。

チャネル・エラー

チャ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 (64 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: UNIX、Linux および 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。

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すると、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ます。ただし、チャンネルが STOPPED の状況の間にキュー・マネージャーが停止すると、SVRCONN チャンネル・タイプのチャンネル状況は再設定されます。

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。

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。

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

WebSphere MQ for UNIX、Linux、および 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ついて詳しくは、[DEFINE CHANNEL](#) コマンドの [KAINT](#) パラメーター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。

WebSphere MQ for UNIX、Linux、および Windows の各システムでは、*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。

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. WebSphere MQ バージョン 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、 $2 * HBINT$ (HBINT が 60 秒未満の場合) および $HBINT + 60$ (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 秒だけ待っ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。

MCA の採用

MCA の採用機能を使用して、IBM WebSphere MQ Explorer で、受信側チャンネルを取り消してから、代わりに別のチャンネルを起動することができます。

チャンネルで通信障害が起きると、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 WebSphere MQ Explorer 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、代わりに別のチャンネルを起動することができます。

さまざまなオプションを使用してこの機能をセットアップすることができます。 **distributed** 分散プラットフォームの場合は、[管理](#)を参照してください。

チャンネルの停止および静止

このトピックでは、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方法を説明します。

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 WebSphere MQ エクスプローラー

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。

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。

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時、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。

チャンネルを再始動するには、以下のコマンドの1つを発行してください。

- START CHANNEL MQSC コマンド
- START CHANNEL PCF コマンド
- IBM WebSphere MQ エクスプローラー

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。

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(分散プラットフォーム)が停止すると、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 の状況の間にキュー・マネージャーが停止すると再設定されます。

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、WebSphere 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。

WebSphere 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)
```

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。

WebSphere 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。distributed 分散プラットフォームでは、この情報は PCF レコードとして生成されます。詳しくは、[構造データ・タイプ](#)を参照してください。

メッセージおよびコード

問題の 1 次診断に役立つメッセージおよびコードについては、[診断メッセージおよび理由コード](#)を参照してください。

メッセージの安全性

WebSphere 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せん。

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、[66 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。

[67 ページの図 15](#) 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。)

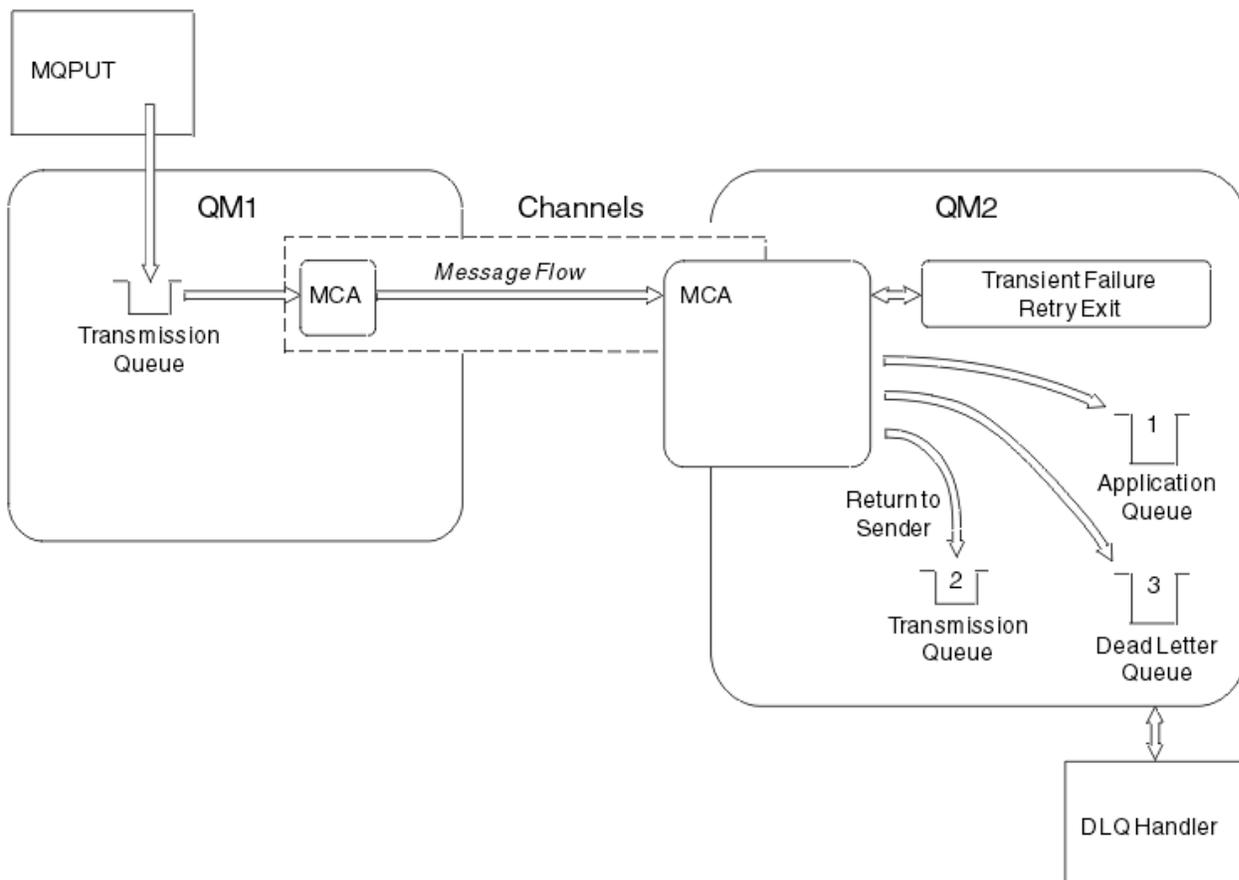


図 15. メッセージを送達できない場合の処理

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、[WebSphere MQ 送達不能キュー・ハンドラーを使用した未配布メッセージの処理 for IBM WebSphere MQ for UNIX、Linux](#) 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 Version 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。

関連資料

[送達不能キューの使用 \(USEDLQ\)](#)

チャンネルのトリガー操作

WebSphere 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 WebSphere 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、69 ページの『[UNIX、Linux および Windows システムでのチャンネルのトリガー操作。](#)』を参照してください。

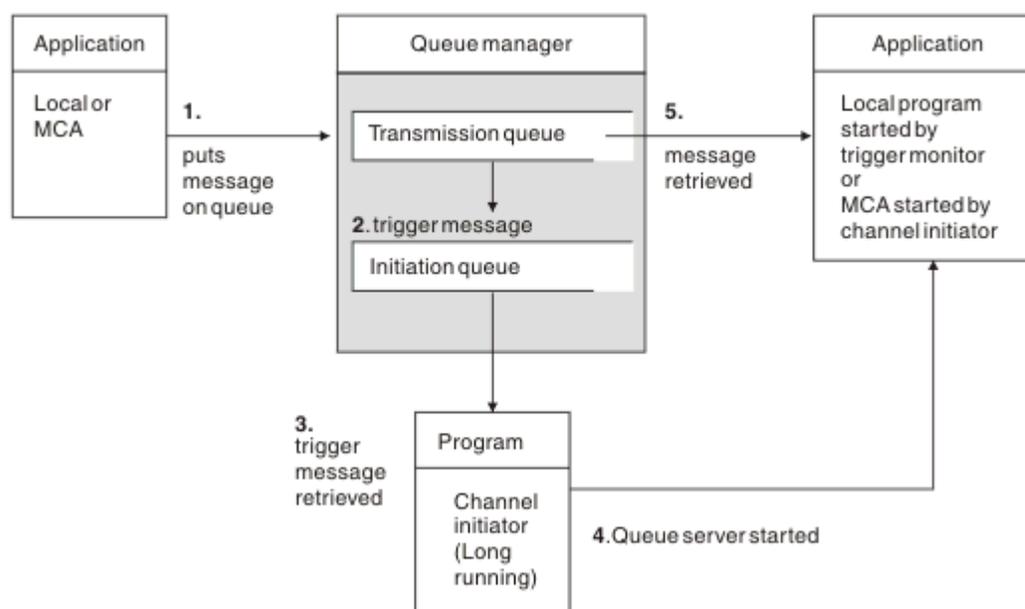


図 16. トリガー操作の概念

トリガー操作に必要なオブジェクトが、68 ページの図 16 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を処理します。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。
- オプションで、トリガー操作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。WebSphere MQ for UNIX、Linux、および Windows システムでは、チャンネル名にブランクを指定できます。その場合、この伝送キューの最初の有効な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 および Windows システム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 WebSphere 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、WebSphere 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。

関連概念

70 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。

29 ページの『[分散キューイングを使用したアプリケーションの接続](#)』

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、WebSphere 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。

関連資料

[UNIX、Linux、および Windows システムでのチャンネル・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。

Linux および Windows システムでは、チャンネル・イニシエーターは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。これは、`MAXINITIATORS` (UNIX and Linux システムの場合は `qm.ini` ファイル、Windows システムの場合はレジストリー内) を使用して変更できます。

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、[WebSphere 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 WebSphere MQ プラットフォームによって異なります。

Windows、UNIX and Linux システム

WebSphere MQ for Windows、UNIX and Linux システムには、WebSphere MQ インストールに関する基本的な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。

WebSphere MQ 構成ファイル

このファイルには、WebSphere 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。詳しくは、[管理 for WebSphere MQ for Windows、UNIX and Linux システム](#)を参照してください。

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 WebSphere 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。

Windows システムの場合:

```
C:\Program Files\IBM\WebSphere 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。

メッセージのデータ変換

WebSphere 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。

WebSphere 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

WebSphere 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、WebSphere 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、WebSphere 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 WebSphere 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。)

WebSphere 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。WebSphere 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考慮し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、WebSphere 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

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 WebSphere 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、ならびにトラステッド・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 とも呼ぶ) 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。DLQ ハンドラーは、WebSphere MQ on UNIX and Linux システムで提供されます。詳しくは、[サンプル DLQ ハンドラー amqsdq](#) 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

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 WebSphere 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が呼び出される正確な場所、および制御ブロックと名前の詳細については、[メッセージング・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、実際にチャン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。不安定なチャンネル出口または STOP CHANNEL MODE(TERMINATE) コマンドを使用していないのであれば、チャン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。

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 WebSphere 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。

以下のタイプのコマンドを使用できます。

IBM WebSphere MQ コマンド (MQSC)

Windows、UNIX and Linux 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。

チャンネル・コマンドは、IBM WebSphere 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

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。詳細については、[制御コマンド](#)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。

IBM WebSphere MQ Explorer

UNIX、Linux および Windows システムでは、IBM WebSphere 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、[75 ページの表 8](#)を参照してください。

関連概念

[77 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。

[83 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、WebSphere MQ for Windows システムにおける通信の 4 つの形式のいずれかを使用して、これを行う方法について説明しています。

[93 ページの『UNIX and Linux システムにおける通信のセットアップ』](#)

DQM は IBM WebSphere 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

[UNIX、Linux、および Windows システムでのチャンネル・プログラム](#)

[分散プラットフォーム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](#)

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 WebSphere 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 WebSphere 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い限りなりません。ただし、IBM WebSphere 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。

- [77 ページの『オブジェクトの準備作業』](#)
- [77 ページの『関連オブジェクトの作成』](#)
- [78 ページの『デフォルト・オブジェクトの作成』](#)
- [78 ページの『チャンネルの作成』](#)
- [79 ページの『チャンネルの表示』](#)
- [79 ページの『チャンネル状況の表示』](#)
- [80 ページの『Ping を使用したリンクの検査』](#)
- [80 ページの『チャンネルの開始』](#)
- [81 ページの『チャンネルの停止』](#)
- [82 ページの『チャンネルの名前変更』](#)
- [82 ページの『チャンネルのリセット』](#)
- [83 ページの『チャンネルの未確定メッセージの解決』](#)

75 ページの表 8 は、必要になる場合のある IBM WebSphere MQ 機能の全リストです。

表 8. UNIX, Linux, and Windows システムで必要な機能			
関数	制御コマンド	MQSC	WebSphere 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	crtmqm		Yes
キュー・マネージャーの削除	dltmqm		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	endmqm		Yes
キュー・マネージャーの ping		PING QMGR	いいえ

表 8. 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	WebSphere MQ エクスプローラーで同等のコマンドがあるか
キュー・マネージャーの始動	strmqm		Yes
コマンド・サーバー機能			
コマンド・サーバーの表示	dspmqcsv		いいえ
コマンド・サーバーの終了	endmqcsv		いいえ
コマンド・サーバーの開始	strmqcsv		いいえ
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
プロセスの作成		DEFINE PROCESS	Yes
プロセスの削除		DELETE PROCESS	Yes
プロセスの表示		DISPLAY PROCESS	Yes
チャンネル機能			
チャンネルの変更		ALTER CHANNEL	Yes
チャンネルの作成		DEFINE CHANNEL	Yes
チャンネルの削除		DELETE CHANNEL	Yes
チャンネルの表示		DISPLAY CHANNEL	Yes

表 8. 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	WebSphere MQ エクスプローラーで同等のコマンドがあるか
チャンネル状況の表示		DISPLAY CHSTATUS	Yes
チャンネルの終了		STOP CHANNEL	Yes
チャンネルの ping		PING CHANNEL	Yes
チャンネルのリセット		RESET CHANNEL	Yes
チャンネルの解決		RESOLVE CHANNEL	Yes
チャンネルの実行	runmqchl	START CHANNEL	Yes
チャンネル・イニシエーターの実行	runmqchi	START CHINIT	いいえ
リスナー ¹ の実行	runmqlsr	START LISTENER	いいえ
リスナーの終了	endmqlsr(Windows システム、AIX、HP-UX、および Solaris のみ)		いいえ
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。			

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。

WebSphere MQ コマンド (MQSC) または IBM WebSphere MQ エクスプローラーを使用して、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。

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、[分散プラットフォーム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。

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。

- 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。
- 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` コマンドについての詳細は、[制御コマンド](#)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。

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 Windows、UNIX または Linux システム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。デフォルト・キュー・マネージャーとして `runmqsc` を入力するか、必要なキュー・マネージャーの名前が `qmname` である場合は、`runmqsc qmname` 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 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。UNIX and Linux システムでは、`Ctrl-d` を使用します。また、UNIX、Linux、または Windows システムで、**end** コマンドを使用します。

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
```

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
- 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。

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、送信側チャンネルおよびサーバー・チャンネルからのみ利用できます。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せん。

メッセージ交換の結果は、Ping complete またはエラー・メッセージとして示されます。

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 WebSphere 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。

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。

- UNIX and Linux システム上の 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。
- UNIX and Linux システムにおける 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 システムの LU 6.2 の場合、SNA Server を使用すると、TpStart (SNA Server で提供されるユーティリティ) 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、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

`MQ_INSTALLATION_PATH\qmgrs\qmname\errors\AMQERR01.LOG` (qmname と呼ばれる各キュー・マネージャーの場合)

`MQ_INSTALLATION_PATH\qmgrs\@SYSTEM\errors\AMQERR01.LOG` (一般エラーの場合)

`MQ_INSTALLATION_PATH` は、WebSphere 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 システムでは、Windows システム・アプリケーション・イベント・ログにもメッセージが表示されます。

UNIX and Linux システム

`/var/mqm/qmgrs/qmname/errors/AMQERR01.LOG` (qmname と呼ばれる各キュー・マネージャーの場合)

`/var/mqm/qmgrs/@SYSTEM/errors/AMQERR01.LOG` (一般エラーの場合)

Windows、UNIX and Linux システムでは、`runmqslsr` コマンドを使用して WebSphere 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。

```
runmqslsr -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。

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、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)
```

2. 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
```

3. 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。

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、[64 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、MQCMT_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ることがあります。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。

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検査する。
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。

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

[77 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。

[54 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。

[29 ページの『分散キューイングを使用したアプリケーションの接続』](#)

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、WebSphere 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。

関連資料

[RESET CHANNEL](#)

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・キュー・マネージャーが実行している。

関連概念

[77 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。

[54 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。

[29 ページの『分散キューイングを使用したアプリケーションの接続』](#)

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、WebSphere 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。

関連資料

[RESOLVE CHANNEL](#)

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、WebSphere MQ for Windows システムにおける通信の4つの形式のいずれかを使用して、これを行う方法について説明しています。

構成例 - [IBM WebSphere MQ for Windows](#) を参照すると役に立つ場合があります。

UNIX and Linux システムについては、[93 ページの『UNIX and Linux システムにおける通信のセットアップ』](#)を参照してください。

接続の決定

次の 4 つの形式の WebSphere MQ for Windows システムの通信から選択します。

- [84 ページの『Windows での TCP 接続の定義』](#)
- [86 ページの『Windows での LU 6.2 接続の定義』](#)
- [88 ページの『Windows での NetBIOS 接続の定義』](#)
- [91 ページの『Windows での SPX 接続の定義』](#) (Windows XP および Windows 2003 Server のみ)

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。

WebSphere MQ クライアントの場合、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。WebSphere MQ クライアントの詳細については、[クライアントの概要](#)を参照してください。

関連概念

[29 ページの『分散キューイングを使用したアプリケーションの接続』](#)

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、WebSphere 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。

[74 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 WebSphere 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。

[99 ページの『クライアントとサーバー間の接続の構成』](#)

WebSphere MQ MQI クライアント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。

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 WebSphere 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を変更するには、IBM WebSphere MQ for Windows の .ini ファイルでそのポート番号を指定します。

```
TCP:  
Port=1822
```

注: IBM WebSphere 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ているポート属性。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 WebSphere 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 WebSphere 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 WebSphere MQ リスナーを使用することができます。

IBM WebSphere 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、`runmqtsr` 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 WebSphere 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ているポート属性。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 WebSphere MQ に割り当てた番号です。

最良のパフォーマンスを得るには、73 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、IBM WebSphere MQ リスナーをトラステッド・アプリケーションとして実行します。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、62 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、Windows KeepAliveTime オプションの HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters レ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を制御します。デフォルト値は2時間です。

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。

リモート・プラットフォーム	TPNAME	TPPATH
z/OS または MVS/ESA (CICS なし)	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)	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、WebSphere 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、以下のオンラインの IBM 資料を参照してください: [AnyNet SNA over TCP/IP](#) および [SNA ノード操作](#)。

関連概念

87 ページの『[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。

87 ページの『[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。

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
```

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
```

最良のパフォーマンスを得るには、「[トラステッド・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、WebSphere MQ リスナーをトラステッド・アプリケーションとして実行します。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 WebSphere MQ リスナーを停止できます。

```
ENDMQLSR [-m QMNAME]
```

Windows の場合でのマイクロソフト 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 での NetBIOS 接続の定義

WebSphere MQ は、他の WebSphere 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、WebSphere 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、システムを再始動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、WebSphere MQ がこ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

[88 ページの『IBM WebSphere MQ ローカル NetBIOS 名の定義』](#)

IBM WebSphere 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。

[89 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。

[89 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 WebSphere MQ *qm.ini* ファイルの NETBIOS スタンザの *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。

[90 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。

[90 ページの『NetBIOS 接続のターゲット・リスナー』](#)

NetBIOS 接続の受信側で行われるステップを定義します。

IBM WebSphere MQ ローカル NetBIOS 名の定義

IBM WebSphere 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 WebSphere 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。

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
```

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 WebSphere MQ qm.ini ファイルの NETBIOS スタンザの AdapterNum 値を使用し、runmqslsr 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。

NetBIOS 接続で、IBM WebSphere MQ によって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。

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
```

NetBIOS 接続のターゲット・リスナー

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. WebSphere 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。

最良のパフォーマンスを得るには、73 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、WebSphere MQ リスナーをトラステッド・アプリケーションとして実行します。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 WebSphere 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。

Windows での SPX 接続の定義

SPX 接続は、Windows XP および Windows 2003 Server を実行中のクライアントおよびサーバーにのみ適用されます。

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。受信側では、リスナー・プログラムを実行する必要があります。

関連概念

91 ページの『SPX での送信側』

ターゲット・マシンがリモートであるとき、チャンネル定義の「Connection name (接続名)」フィールドにターゲット・マシンの SPX アドレスを指定します。

92 ページの『SPX 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。

93 ページの『IPX/SPX パラメーター』

ほとんどの場合、IPX/SPX パラメーターはデフォルト設定から変更する必要はありません。しかし、WebSphere MQ 用にその使用を調整するには、運用環境で設定の一部を変更しなければならない場合があります。

SPX での送信側

ターゲット・マシンがリモートであるとき、チャンネル定義の「Connection name (接続名)」フィールドにターゲット・マシンの SPX アドレスを指定します。

SPX アドレスは、次の形式で指定されます。

```
network.node(socket)
```

ここで、

network

リモート・マシンが存在するネットワークの 4 バイトのネットワーク・アドレス。

node

6 バイトのノード・アドレスであり、リモート・マシンの LAN アダプターの LAN アドレス。

socket

リモート・マシンが listen する 2 バイトのソケット番号。

ローカル・マシンとリモート・マシンが同一ネットワーク上にある場合、ネットワーク・アドレスを指定する必要はありません。リモート側がデフォルトのソケット (5E86) で listen している場合、ソケットを指定する必要はありません。

MQSC コマンドの CONNAME パラメーターに指定された完全指定の SPX アドレスの例は、次のようになります。

```
CONNAME('00000001.08005A7161E5(5E87)')
```

両マシンとも同一ネットワーク上にあるデフォルトの場合は、次のようになります。

```
CONNAME(08005A7161E5)
```

デフォルトのソケット番号を変更するには、キュー・マネージャー構成ファイル (qm.ini) に次のようにソケット番号を指定します。

```
SPX:  
Socket=5E87
```

qm.ini を使用して設定した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。

SPX 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。

WebSphere MQ リスナーを使用します。

SPX リスナー・バックログ・オプション

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、リスナーが要求を受け入れるのを SPX 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、92 ページの表 10 に示すとおりです。

プラットフォーム	デフォルトのリスナー・バックログ値
Windows Server	5
Windows ワークステーション	5

バックログが 92 ページの表 10 の値に到達した場合、MQCONN または MQCONNX を使用してキュー・マネージャーに接続しようとする、理由コード MQRC_Q_MGR_NOT_AVAILABLE を受信します。その場合には、接続を再試行できます。

ただし、qm.ini ファイルまたは Windows のレジストリーに次の項目を追加すれば、このエラーを回避できます。

```
SPX:  
ListenerBacklog = n
```

こうすると、SPX リスナーのデフォルトの最大未解決要求数 (92 ページの表 10 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、それを使用して接続限度に到達するのを回避できます。

backlog オプションをオン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。

- RUNMQLSR -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。

RUNMQLSR コマンドについては、runmqslsr 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。

WebSphere MQ リスナーの使用

WebSphere 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、RUNMQLSR コマンドを使用します。以下に例を示します。

```
RUNMQLSR -t spx [-m QMNAME] [-x 5E87]
```

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・ソケット番号 (5E86) を使用する場合はソケット番号は必要ありません。

パフォーマンスを最高にするには、73 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 WebSphere 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詳しくは、[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 WebSphere 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。

IPX/SPX パラメーター

ほとんどの場合、IPX/SPX パラメーターはデフォルト設定から変更する必要はありません。しかし、WebSphere MQ 用にその使用を調整するには、運用環境で設定の一部を変更しなければならない場合があります。

実パラメーター、およびそれを変更する方法は、SPX 通信サポートのプラットフォームおよびプロバイダーによって異なります。例のセクションでは、これらのパラメーターの一部、特に、WebSphere MQ チャネルおよびクライアント接続の操作に影響を与える可能性のあるパラメーターについて説明します。

Windows システム

NWLink IPX および SPX のパラメーターの使用および設定の詳細については、Microsoft の資料を参照してください。IPX/SPX パラメーターは、レジストリーの次のパスにあります。

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service\NWLinkSPX\Parameters  
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service\NWLinkIPX\Parameters
```

UNIX and Linux システムにおける通信のセットアップ

DQM は IBM WebSphere 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ネル定義は、これらの接続を使用します。

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。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。次のセクションにも役に立つ情報が記載されています。

- [構成例 - IBM WebSphere MQ for AIX](#)
- [構成例 - IBM WebSphere MQ for HP-UX](#)
- [構成例 - IBM WebSphere MQ for Solaris](#)
- [構成例 - IBM WebSphere MQ for Linux](#)

Windows については、[83 ページの『Windows における通信のセットアップ』](#)を参照してください。

UNIX and Linux システム上の WebSphere MQ の通信の 2 つの形式から選ぶことができます。

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

どのチャ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 WebSphere MQ Explorer MQI クライアントの場合、別の伝送プロトコルを使用する代替チャネルがあると役に立つことがあります。IBM WebSphere MQ Explorer MQI クライアントの詳細については、[IBM WebSphere MQ MQI クライアントの概要](#)を参照してください。

関連概念

[29 ページの『分散キューイングを使用したアプリケーションの接続』](#)

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、WebSphere 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。

[74 ページの『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 WebSphere 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。

99 ページの『[クライアントとサーバー間の接続の構成](#)』

WebSphere MQ MQI クライアント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。

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 WebSphere 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anzas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 WebSphere 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

95 ページの『[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。

96 ページの『[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。

97 ページの『[WebSphere MQ リスナーの使用](#)』

WebSphere 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、runmqtsr コマンドを使用します。

97 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。

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 WebSphere 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` ファイルに次の 1 行を追加し、`amqcrsta` プログラムを呼び出します。ここで `MQ_INSTALLATION_PATH` は、WebSphere MQ がインストールされている上位のディレクトリーを表しています。

```
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

```
inetd -c
```

- Solaris 10 以降の場合

```
inetconv
```

- 他の UNIX and Linux システム (Solaris 9 を含む) の場合

```
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、WebSphere 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 WebSphere 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
```

ここで、MQ_INSTALLATION_PATH は、WebSphere MQ がインストールされている上位のディレクトリーを表しています。

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、96 ページの『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。

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、96 ページの表 11 に示すとおりです。

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 サーバー	100
Windows ワークステーション	100
z/OS	255

バックログが 96 ページの表 11 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 (96 ページの表 11 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 オプションをオン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。

- runmq1sr -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。

WebSphere MQ リスナーの使用

WebSphere 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。QMNAME は、デフォルトのキュー・マネージャーには必要ありません。また、デフォルト (1414) を使用している場合は、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。

最良のパフォーマンスを得るには、73 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、WebSphere MQ リスナーをトラステッド・アプリケーションとして実行します。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 WebSphere MQ リスナーを停止できます。

```
endmqclsr [-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。

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、62 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。

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。	-

表 12. ローカル UNIX and Linux 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 (続き)

リモート・プラットフォーム	TPNAME	TPPATH
z/OS CICS の使用	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、WebSphere 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

98 ページの『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。

98 ページの『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。

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、WebSphere MQ は SNA セッションの確立を試みるたびに、それを使用します。

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	rcv
User ID	0

MQ_INSTALLATION_PATH は、WebSphere 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 APPCLU (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。

クライアントとサーバー間の接続の構成

WebSphere MQ MQI クライアント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。

WebSphere 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。WebSphere MQ MQI クライアント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。WebSphere MQ MQI クライアント側の WebSphere 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。この方法についての詳細は、[チャンネルのユーザー定義](#)を参照してください。

MQI チャンネルを定義する前に、次のステップを実行する必要があります。

1. 使用する予定の通信形式を決めます。99 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。

- 接続を設定します。
- 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。
- リスナーを開始して、サーバーによる WebSphere MQ MQI クライアント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。

使用する通信タイプ

サポートされる伝送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、WebSphere MQ MQI 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。

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よって、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。

- LU 6.2
- NetBIOS
- SPX
- TCP/IP

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。WebSphere MQ MQI クライアントの場合、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、WebSphere MQ MQI 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って制限される場合があります。可能な組み合わせが、次の表に示されています。

表 13. 伝送プロトコル - WebSphere MQ MQI クライアント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	WebSphere MQ MQI クライアント	WebSphere MQ サーバー
TCP/IP	UNIX システム Windows	UNIX システム Windows z/OS
LU 6.2	UNIX システム ¹ Windows	UNIX システム ¹ Windows
NetBIOS	Windows	Windows
SPX	Windows	Windows
注:		
1. Linux for Power Systems 以外		

異なるタイプの接続のセットアップについては、以下のリンクを参照してください。

- [84 ページの『Windows での TCP 接続の定義』](#)
- [94 ページの『UNIX and Linux での TCP 接続の定義』](#)
- [102 ページの『TCP/IP 接続の制限』](#)
- [86 ページの『Windows での LU 6.2 接続の定義』](#)
- [97 ページの『UNIX and Linux での LU 6.2 接続の定義』](#)
- [88 ページの『Windows での NetBIOS 接続の定義』](#)
- [91 ページの『Windows での SPX 接続の定義』](#)

関連概念

[103 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。

[112 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。

[114 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。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。

[117 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。

[122 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、および Windows システム上の WebSphere MQ MQI クライアント環境では、3つのタイプのチャンネル出口を使用できます。

[125 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。

[127 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。

関連タスク

[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](#)

関連資料

[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、WebSphere MQ MQI 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。

さまざまなプラットフォーム上の MQI チャンネルによる通信には、次の 4 つのタイプがあります。

- LU 6.2
- NetBIOS
- SPX
- TCP/IP

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。WebSphere MQ MQI クライアントの場合、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、WebSphere 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。

伝送プロトコル	WebSphere MQ MQI クライアント	WebSphere MQ サーバー
TCP/IP	UNIX システム Windows	UNIX システム Windows
LU 6.2	UNIX システム ¹ Windows	UNIX システム ¹ Windows
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

[84 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。

[94 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。

[86 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。

[97 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。

88 ページの『Windows での NetBIOS 接続の定義』

WebSphere MQ は、他の WebSphere 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。

91 ページの『Windows での SPX 接続の定義』

SPX 接続は、Windows XP および Windows 2003 Server を実行中のクライアントおよびサーバーにのみ適用されます。

関連資料

102 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。

TCP/IP 接続の定義

クライアント側でチャンネル定義上の TCP のトランスポート・タイプを指定します。サーバー上でリスナー・プログラムを開始します。

チャンネル定義上の TCP のトランスポート・タイプを指定することで、クライアントでの TCP/IP 接続を指定します。

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。リスナー・プログラムの開始手順は、サーバー・プラットフォームによって異なります。

ご使用のクライアントおよびサーバーのプラットフォームの関連トピックを参照してください。

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 WebSphere 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	255
Solaris	100
Windows Server	100
Windows ワークステーション	100
z/OS	255

接続限度に達すると、クライアントは、MQCONN 呼び出しから MQRC_HOST_NOT_AVAILABLE の戻りコードと、クライアント・エラー・ログ (UNIX and Linux システムでは /var/mqm/errors/AMQERR0n.LOG、Windows では IBM WebSphere 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) で 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。

NetBIOS または SPX 接続の定義

NetBIOS および SPX 接続は Windows システムにのみ適用されます。

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。 [NetBIOS 接続の定義](#)を参照してください。

SPX 接続は、Windows XP または Windows 2003 Server を実行中のクライアントおよびサーバーにのみ適用されます。 [SPX 接続の定義](#)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。

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 WebSphere 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 WebSphere MQ は、WebSphere Application Server バージョン 4 またはバージョン 5 をサポートしていました。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した。WebSphere Application Server バージョン 6 以降には、WebSphere MQ メッセージング・プロバイダーが含まれているので、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

103 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、WebSphere 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。

112 ページの『Microsoft トランザクション・サーバー』

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、WebSphere 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、[IBM WebSphere MQ クライアントのインストール](#)の説明に従って、まず WebSphere MQ ベース・クライアントを構成する必要があります。その後、このセクション内の情報を使用して、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) 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 チャンネルを使用して、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 チャンネルが使用されます。

MQI チャンネルは、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 チャンネルを開始して、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 チャンネルを開始して、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 チャン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 チャンネルを使用して、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。WebSphere 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` ストリング
- 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` ストリング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` ストリング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` ストリングの形式について、および `xa_open` ストリング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、[105 ページの『`xa_open` ストリングの形式』](#)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、[109 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。

- [110 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』](#)
- [111 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

107 ページの『[xa_open ストリングの 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。

108 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

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

SSL チャネルは 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

108 ページの『[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。

108 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。

xa_open ストリングの形式

xa_open ストリングには、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 ストリング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 チャネルの通信プロトコル。有効な値は次のとおりです。

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。

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 QSG 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。

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。

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

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。

CHANNEL および **CONNAME** パラメータが **xa_open** String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と **TRPTYPE** パラメータを使用して、サーバー・キュー・マネージャーとの **MQI** チャンネルを開始します。

CHANNEL および **CONNAME** パラメータが **xa_open** String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** チャンネルを開始します。**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** パラメータによって識別されるクライアント・チャン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** パラメータの値が、**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・クライアントが **V7.0.1** 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。

アプリケーションが **QMNAME** パラメータ・フィールドで **QSG** 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。

アプリケーションが **V7.0.1** より前のバージョンのキュー・マネージャーに接続されている場合、**xa_open** 呼び出しは成功しますが、トランザクションではリカバリー単位属性指定として **QMGR** が使用されます。リカバリー単位属性指定として **GROUP** を使用する必要があるアプリケーションが、**V7.0.1** 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。

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** チャンネルの接続ハンドルを受け取ります。2番目の **MQI** チャン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** パラメータの値が、**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** チャン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** チャンネルを使用します。2番目の **MQI** チャン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** String内の **QMNAME** パラメータの値が、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** チャン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。これらの関数を使用されるのは、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、108 ページの表 16 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
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tring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tring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。

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は、次のとおりです。

- `xa_open` スtring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、109 ページの表 17 に表示されているライブラリー内にあります。

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 ¹

MQ_INSTALLATION_PATH は、WebSphere 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 WebSphere 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

109 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。

SSL 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

SSL 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 SSL 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 `SSL` 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` チャンネルの定義を含む `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ンネル定義と `CipherSpec` の正確な認証に必要な `SSL` 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。

`CICS` オンライン・リソース定義 (`RDO`) コマンド `cicsadd` を使用して、`XAD` リソース定義を追加します。`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` の各プラットフォームで `CICS` が使用するためのスイッチ・ロード・ファイルが 1 つ提供されています。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` を指すポインターを戻す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、[110 ページの表 18](#) を参照してください。

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

`MQ_INSTALLATION_PATH` は、`WebSphere 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="WebSphere MQ queue manager MARS" \  
XAOpen="channel=MARS.SVR,trptype=tcp,connname=MARS(1415),qmname=MARS,tpm=cics" \  
SwitchLoadFile="C:\Program Files\IBM\WebSphere MQ\bin\mqcc4swi.dll"
```

CICS 領域への XAD リソース定義の追加について詳しくは、「*CICS Administration Reference*」およびご使用のプラットフォームの「*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 を使用する場合は、次の情報に注意してください。

- WebSphere MQ の XAD リソース定義は、CICS 領域に 1 つのみ追加できます。これは、1 つの領域に関連付けることができるキュー・マネージャーは 1 つのみであり、その領域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・マネージャーにのみ接続できることを意味します。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する場合は、別の領域で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 チャンネルを開始します。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 WebSphere MQ MQI クライアント・アプリケーションは、同じ MQI チャンネルを使用します。
- MQI チャンネルが開始されるときに、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。Under certain circumstances, the queue manager uses this user ID for authority checks when the server connection MCA subsequently attempts to access the queue manager resources on behalf of a client application. 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は、[セキュリティー](#)を参照してください。
- WebSphere 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、[111 ページの表 19](#) にリストされています。これらの出口は、WebSphere MQ サーバー・システムの対応する出口を構成するのと同じ方法で構成します。したがって、この情報については、[CICS ユーザー出口の使用可能化](#)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	amqczscg
Windows システム	amqzscgn.c	mqqc1415.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` ストリングを指定します。

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 `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¥ 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 トランザクション・サーバー

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は、次のとおりです。

- 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、WebSphere MQ MQI クライアント・マシン上の WebSphere 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 WebSphere 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、WebSphere MQ MQI クライアント・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、[114 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 WebSphere MQ MQI クライアント・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、[117 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

WebSphere z/OS 以外のプラットフォーム上の MQ 製品には、チャンネル定義が存在しない場合にサーバー上に自動的に作成できる機能が組み込まれています。

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、WebSphere 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。

チャンネル定義の自動作成の詳細は、[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](#)を参照してください。

関連概念

[113 ページの『チャンネルの自動定義』](#)

WebSphere z/OS 以外のプラットフォーム上の MQ 製品には、チャンネル定義が存在しない場合にサーバー上に自動的に作成できる機能が組み込まれています。

[113 ページの『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、WebSphere MQ MQI クライアント・マシン上の WebSphere 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 とおあります。

[54 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。

チャンネルの自動定義

WebSphere z/OS 以外のプラットフォーム上の MQ 製品には、チャンネル定義が存在しない場合にサーバー上に自動的に作成できる機能が組み込まれています。

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、WebSphere 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。

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、WebSphere MQ MQI クライアント・マシン上の WebSphere 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 WebSphere 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、WebSphere MQ MQI クライアント・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、[114 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 WebSphere MQ MQI クライアント・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、[117 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。

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。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。

すべてのプラットフォームで、WebSphere 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 WebSphere MQ エクスプローラーを使用して、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。

MQSC コマンドは、WebSphere MQ が WebSphere MQ MQI クライアント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ンネルを定義する必要があります。

関連概念

[115 ページの『IBM WebSphere MQ MQI クライアント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。

関連タスク

[114 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』](#)

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。

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。

```
crtmqm QM1
```

- b) キュー・マネージャーを始動します。

```
strmqm 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 SSL または 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 WebSphere MQ MQI クライアント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
```

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 WebSphere MQ MQI 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

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、[148 ページの『MQSERVER』](#)を参照してください。

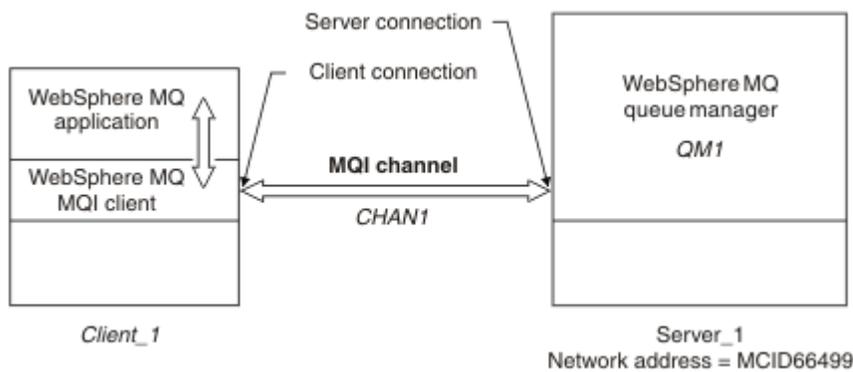


図 17. 簡単なチャンネル定義

簡単なチャンネル定義のさらいくつかの例は、次のとおりです。

- 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
```

この BOX99 は、LU 6.2 ConnectionName です。

IBM WebSphere MQ MQI クライアントでは、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 **MQCONNX** 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、148 ページの『**MQSERVER**』を参照してください。

MQCONNX 呼び出しでの MQCNO 構造体の使用

IBM WebSphere MQ MQI クライアント・アプリケーションは、**MQCONNX** 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が入っているチャンネル定義構造体 MQCD を参照することができます。

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 Secure Sockets Layer (SSL) または 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 SSL または TLS に関連する情報も提供できます。SSL または TLS に関連する追加情報は、SSL または TLS 構成オプション構造体 MQSCO で提供できます。この情報は、**MQCONNX** 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: **MQCONNX** 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。

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、WebSphere 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 WebSphere MQ エクスプローラーを使用して、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。

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。

関連概念

[117 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。z/OS 以外のプラットフォームの場合、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

[119 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。

[120 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。

[121 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』](#)

コピーまたは共有することにより、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可能にして、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。

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。z/OS 以外のプラットフォームの場合、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。

z/OS 以外の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。クライアント接続チャンネルは、ユーザーが **DEFINE CHANNEL** コマンドを使用するときにテーブルに追加され、その定義は、ユーザーが **ALTER CHANNEL** コマンドを発行するときに変更されます。

CCDT を使用してクライアントに認証情報を提供し、SSL 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。

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。

z/OS 以外のプラットフォーム

AMQCLCHL.TAB と呼ばれるデフォルトの CCDT は、ユーザーがキュー・マネージャーを作成するときに作成されます。

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。

- ▶ **UNIX** ▶ **Linux** 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 WebSphere 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。これを行うと、AMQCLCHL.TAB は指定したデータ・パスの @ipcc ディレクトリーに置かれます。

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。

クライアントで CCDT の位置を指定する方法

クライアント・システムでは、2つの方法で 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。

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。

関連資料

145 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、WebSphere MQ MQI クライアント・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](#)

移行およびクライアント・チャンネル定義テーブル (CCDT)

通常、クライアント・チャンネル定義テーブルの内部形式は、IBM WebSphere MQ のリリース・レベルごとに変更されます。結果として、IBM WebSphere MQ MQI クライアントがクライアント・チャンネル定義テ

ブルを使用できるのは、そのクライアントと同じかそれより前のリリース・レベルのサーバー・キュー・マネージャーによってそれが作成された場合のみです。

バージョン 7.1 の IBM WebSphere MQ クライアントは、バージョン 6.0 のキュー・マネージャーによって作成されたクライアント・チャンネル定義テーブルを使用できます。しかし、バージョン 6.0 のクライアントは、バージョン 7.1 のキュー・マネージャーによって作成されたクライアント・チャンネル定義テーブルを使用することはできません。

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 WebSphere MQ は Active Directory で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・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 WebSphere 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 WebSphere 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 WebSphere 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](#) で説明されているように、`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 SSL または 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。

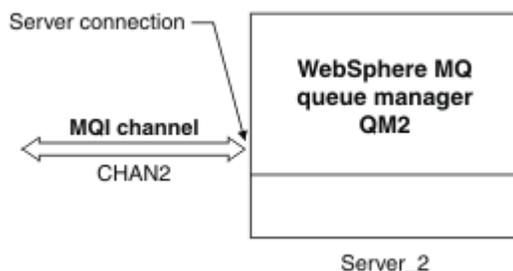


図 18. サーバー接続チャンネルの定義

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 (CONNAME) 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 WebSphere 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) +
CONNAME (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 SSL または 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

z/OS 以外の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、[117 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。

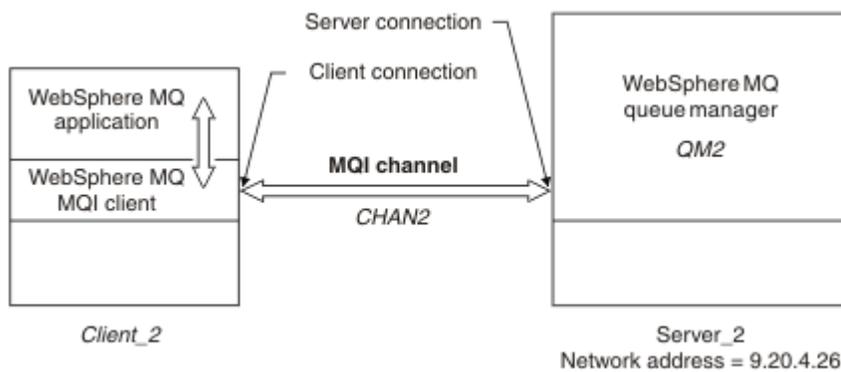


図 19. クライアント接続チャンネルの定義

クライアント接続チャンネル定義へのアクセス

コピーまたは共有することにより、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可能にして、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。

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 の場合、CCDT を作成済みであること。他のプラットフォームの場合、CCDT は自動的に作成および更新されます。

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。

手順

1. 次の 3 つのうちいずれかの方法で、CCDT をクライアント・アプリケーションで使用可能にします。

- 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。
- 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。
- 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。

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。

- 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、[135 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。
- 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。

• HP Integrity NonStop Server および UNIX and Linux システムの場合:

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

ここで、MQ_INSTALLATION_PATH は、WebSphere 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。

- 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

MQSERVER 環境変数が設定されている場合、WebSphere 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。

MQI チャンネル用のチャンネル出口プログラム

UNIX、Linux、および Windows システム上の WebSphere MQ MQI クライアント環境では、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、WebSphere 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、WebSphere 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、WebSphere 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

[123 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。

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

関連タスク

[キュー・マネージャーの機能の拡張](#)

関連資料

[122 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。

[124 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。

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 および Windows システムでは、WebSphere MQ MQI クライアント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。次のスタンザを使用して、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。

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。

124 ページの図 20 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 WebSphere 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。

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 20. 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: WebSphere MQ v7.1 のリリースより前に構成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・パラメーター](#)を参照してください。

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時に使用される 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 125 ページの『1』 , 125 ページの『2』	X'81'	X'91'
MQDISC 125 ページの『1』	X'82'	X'92'
MQOPEN 125 ページの『3』	X'83'	X'93'
MQCLOSE	X'84'	X'94'

表 20. 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 125 ページの『4』	X'85'	X'95'
MQPUT 125 ページの『4』	X'86'	X'96'
MQPUT1 要求 125 ページの『4』	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。

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。

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となります。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、WLM/DNS グループ名または 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。

次のように SVRCONN チャンネルを定義します。

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') 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。

次のように CLNTCONN チャンネルを定義します。

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

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。

関連概念

126 ページの『チャンネル定義の作成』

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。

チャンネル定義の作成

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。

次のように SVRCONN チャンネルを定義します。

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') 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。

次のように CLNTCONN チャンネルを定義します。

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

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ますようになります。

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。

UNIX and Linux プラットフォームで使用されるキュー・マネージャー構成ファイル `qm.ini` と同様に、テキスト・ファイルを使用して IBM WebSphere MQ MQI client を構成します。ファイルにはいくつかのスタンザが含まれており、それぞれのスタンザには **attribute-name=value** という形式の複数の行が含まれています。

この資料では、このファイルを *WebSphere MQ MQI* クライアント構成ファイルと呼びます。このファイル名は通常 `mqclient.ini` ですが、別の名前を付けるように選択することもできます。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 MQI、IBM WebSphere MQ classes for Java、IBM WebSphere MQ classes for JMS、IBM WebSphere MQ classes for .NET、および XMS を使用するクライアントに適用されます。

IBM WebSphere MQ MQI client 構成ファイル内の属性はほとんどの IBM WebSphere 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 WebSphere MQ classes for Java または IBM WebSphere 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、[129 ページの『各属性を読み取ることができる IBM WebSphere 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。

バージョン 7.0 より前の IBM WebSphere MQ のリリースでサポートされていた環境変数は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 WebSphere 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 構成ファイルにプロパティを設定する。
- JMS 構成ファイルもオーバーライドする、Java システム・プロパティを設定する。

.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。

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ない点に注意してください。

クライアント構成ファイルの例

```
## Module Name: mqclient.ini                                ##
## Type       : WebSphere 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=amqldapi
  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)
```

関連資料

[128 ページの『クライアント構成ファイルの場所』](#)

IBM WebSphere MQ MQI クライアント構成ファイルは複数の場所で保持することができます。

[135 ページの『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。

[137 ページの『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ンネル出口のデフォルト位置を指定します。

[138 ページの『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。

[139 ページの『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。

[140 ページの『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、SSL または TLS の使用に関する情報を指定します。

[143 ページの『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。

[144 ページの『WebSphere MQ 環境変数の使用』](#)

このセクションでは、WebSphere MQ MQI クライアント・アプリケーションで使用できる環境変数について説明します。

[440 ページの『キュー・マネージャー構成情報の変更』](#)

ここで説明する属性は、個々のキュー・マネージャーの構成を変更します。これらの属性は、WebSphere MQ の設定値を変更します。

クライアント構成ファイルの場所

IBM WebSphere MQ MQI クライアント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 WebSphere MQ MQI クライアント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ない Java 1.4.2 JRE を許可するには、MQCLNTCF 環境変数を MQCLNTCF Java システム・プロパティーでオーバーライドします。
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 WebSphere MQ データ・ディレクトリー内の mqclient.ini というファイル。

次の事項に注意してください。

- IBM WebSphere MQ データ・ディレクトリーは、例えば、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。クライアントが別の製品で提供された場合も同様です。
 - 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、インストール中に環境変数 MQ_FILE_PATH を構成して、データ・ディレクトリーを指すようにします。通常は C:\Program Files\IBM\WebSphere MQ
 - 環境変数の読み取りをサポートしない Java 1.4.2 JRE を許可するには、MQ_FILE_PATH 環境変数を MQ_FILE_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 クライアントの場合、これは user.home Java システム・プロパティーの値です。
 - 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。

注：IBM WebSphere MQ Client for HP Integrity NonStop Server の場合、mqclient.ini ファイルは OSS ファイル・システムに配置する必要があります。Guardian アプリケーションは、mqclient.ini ファイルを IBM WebSphere MQ データ・ディレクトリーに配置するか、MQCLNTCF 環境変数を OSS ファイル・システム内の場所に設定する必要があります。

各属性を読み取ることができる IBM WebSphere MQ クライアント

IBM WebSphere 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 WebSphere MQ classes for Java や IBM WebSphere 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 21. 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.NET	管理 XMS .NET
CHANNELS スタンザ						
CCSID	使用するコード化文字セット番号。	Yes	No	No	Yes	Yes
ChannelDefinitionDirectory	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

表 21. 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.NET	管理 XMS .NET
ChannelDefinitionFile	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
ReconDelay	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
DefRecon	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
MQReconnectTimeout	クライアントに再接続するタイムアウト(秒)。	Yes	No	No	Yes	No
ServerConnectionParms	IBM WebSphere MQ サーバーの位置および使用する通信方式。	Yes	No	No	Yes	Yes
Put1DefaultAlwaysSync	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。	Yes	Yes	Yes	Yes	Yes
ClientExitPath スタンザ						

表 21. 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.NET	管理 XMS .NET
<u>ExitsDefaultPath</u>	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	No	No	Yes	Yes
<u>ExitsDefaultPath64</u>	クライアントの 64 ビット・チャネル出口の位置を指定します。	Yes	No	No	Yes	Yes
<u>JavaExitsClassPath</u>	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
MessageBuffer スタンザ						
<u>MaximumSize</u>	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
<u>PurgeTime</u>	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
<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
SSL スタンザ						

表 21. 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 .NET	管理 XMS .NET
CDPCheckExtensions	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
CertificateLabel	チャネル定義の証明書ラベル。	Yes	No	No	No	No
CertificateValidationPolicy	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
ClientRevocationChecks	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
EncryptionPolicySuiteB	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 21. 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.NET	管理 XMS .NET
OCSPAuthentication	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 WebSphere MQ の動作を定義します。	Yes	No	No	No	No
OCSPCheckExtensions	IBM WebSphere 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
SSLCryptoHardware	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
SSLFipsRequired	IBM WebSphere 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

表 21. 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 .NET	管理 XMS .NET
<u>SSLHTTPProxyName</u>	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
<u>SSLKeyRepository</u>	ユーザーのデジタル証明書を保持するキー・リポジトリー的位置 (語幹形式で指定)。	Yes	No	No	No	No
<u>SSLKeyResetCount</u>	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
TCP スタンザ						
<u>ClntRcvBufferSize</u>	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

表 21. 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.NET	管理 XMS .NET
<u>ClntSndBufferSize</u>	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)。	Yes	Yes	Yes	Yes	Yes
<u>Connect Timeout</u>	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
<u>IPAddressVersion</u>	チャン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
<u>KeepAlive</u>	KeepAlive 機能をオンまたはオフに切り替えます。	Yes	Yes	Yes	Yes	Yes
Windows Library1	Windows のみ。TCP/IP ソケット DLL の名前。	Yes	No	No	No	No

HP Integrity NonStop Server の場合は、TMF および TmfGateway スタンザを使用して、TMF/Gateway と通信できます。

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。

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。

CCSID 番号は、MQCCSID 環境パラメーターと同等です。

ChannelDefinitionDirectory=パス

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。

Windows システムの場合、デフォルトは IBM WebSphere MQ インストール・ディレクトリー (通常は C:\Program Files\IBM\WebSphere MQ) です。UNIX and Linux システムの場合、デフォルトは /var/mqm です。

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

ChannelDefinitionFile=ファイル名| 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。

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。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。

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い場合は、クライアント・プログラムは自動的に再接続されません。

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

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 ファイル内 DefRecon 値

表 22. 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

表 22. 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 (続き)

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			
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。

IBM WebSphere MQ classes for XMS .NET クライアントは、プロパティ `XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT` を使用して、再接続するタイムアウトを指定できます。このプロパティのデフォルト値は 1800 秒 (30 分) です。

ServerConnectionParms

`ServerConnectionParms` は `MQSERVER` 環境パラメーターと同等で、使用する IBM WebSphere MQ サーバーの場所と通信方式を指定します。 `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、SSL 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ためです。

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。

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` が設定されているか `MQPMO_NO_SYNCPOINT` が設定されているかに関係なく、`MQPMO_SYNC_RESPONSE` が設定されているかのように動作します。

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ンネル出口のデフォルト位置を指定します。

次の属性を `ClientExitPath` スタンザに含めることができます。

ExitsDefaultPath=string

クライアントの 32 ビット・チャンネル出口の位置を指定します。

ExitsDefaultPath64=string

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。

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。

JMS 構成ファイルでは、JavaExitsClassPath 名に標準 com.ibm.mq.cfg が指定されています。接頭部およびこのフルネームは、Websphere MQ V7.0 システム・プロパティでも使用されます。バージョン 6.0 では、この属性は、バージョン 6.0 の README に記載されたシステム・プロパティ 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 しかない場合は、引き続き Websphere MQ V7.0 での使用が受け入れられます。

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

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|_WCPIC32

APPC DLL の名前。

Library2=DLLName|_WCPIC32

コードが 2 つの別個のライブラリーに保管されている場合に使用される Library1 と同じです。

TPName

リモート・サイトで始動する TP 名。

NETBIOS

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=数値|_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=数値|1

割り振るコマンドの数。

NumSess=数値|1

割り振るセッションの数。

SPX

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=数値|_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=数値|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・バッファに関する情報を指定します。

次の属性を MessageBuffer スタンザに含めることができます。

MaximumSize=整数| 1

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)。

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=整数| 600

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)。

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内に保留になります。先読みバッファ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 (除去が行われないことを意味します) です。

UpdatePercentage=整数| -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。

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 SSL スタンザ

SSL スタンザを使用して、SSL または TLS の使用に関する情報を指定します。

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 SSL 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。

この属性に指定できる値は以下のとおりです。

- YES: SSL 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: SSL 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。

CertificateValPolicy=string

使用する証明書妥当性検査のタイプを決定します。

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。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 SSL/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 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。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

セキュリティ強度を 128 ビット・レベルに設定する。

192_BIT

セキュリティ強度を 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、WebSphere MQ の動作を定義します。指定可能な値は以下の 3 つです。

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

WebSphere MQ が AuthorityInfoAccess 証明書拡張に反応するかどうかを制御します。値を NO に設定すると、WebSphere MQ は AuthorityInfoAccess 証明書拡張を無視し、OCSP セキュリティー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。

次の形式のストリングを指定します。GSK_PKCS11=ドライバーのパスおよびファイル名;トークン・ラベル;トークン・パスワード;対称暗号設定;

例: GSK_PKCS11=/usr/lib/pkcs11/
PKCS11_API.so;tokenlabel;passwd;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。

SSLFipsRequired=YES|_NO

WebSphere 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。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。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。

SSLKeyResetCount=整数|_0

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。

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、SSL 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。

次の属性を TCP スタンザに含めることができます。

ClntRcvBuffSize=数値|_32768

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)。値 0 は、WebSphere 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。

ClntSndBuffSize=数値|_32768

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)。値 0 は、WebSphere 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

ソケットへの接続の試行がタイムアウトになるまでの秒数。デフォルトは 0 です。ただし、クライアント・チャンネルの重み付けをゼロ以外に設定してチャンネルを構成した場合のデフォルトは 5 です。

IPAddressVersion=MQIPADDR_IPV4|MQIPADDR_IPV6

チャンネル接続に使用する IP プロトコルを指定します。

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、チャンネルは閉じた状態です。

Windows Library1=DLLName|_WSOCK32

(Windows のみ) TCP/IP ソケット DLL の名前。

TMF および TMF/Gateway のスタンザ

IBM WebSphere MQ が提供する TMF/Gateway は、Pathway 環境で実行されます。TMF および TMF/Gateway のスタンザを使用して、TMF/Gateway と通信するために IBM WebSphere MQ Client for HP Integrity NonStop Server で必要な構成パラメーターを指定します。

TMF を使用する場合、1 つの TMF スタンザと、通信するキュー・マネージャーごとに 1 つの TmfGateway スタンザを定義する必要があります。すべての値は、ユーザーの構成から派生します。

TMF スタンザ

PathMon=name

TMF/Gateway のサーバー・クラスを定義する定義済み Pathmon 処理の名前。

TmfGateway スタンザ

次の属性をこのスタンザに含めることができます。

QManager=name

キュー・マネージャーの名前。

Server=name

そのキュー・マネージャーに対して構成した TMF/Gateway のサーバー・クラス名。

例

以下に、異なるサーバー上の2つの異なるキュー・マネージャーに対して、2つの TmfGateway スタンザを使用して定義された TMF スタンザの例を示します。

```
TMF:
  PathMon=$PSD1P

TmfGateway:
  QManager=MQ5B
  Server=MQ-MQ5B

TmfGateway:
  QManager=MQ5C
  Server=MQ-MQ5C
```

WebSphere MQ 環境変数の使用

このセクションでは、WebSphere MQ MQI クライアント・アプリケーションで使用できる環境変数について説明します。

以下の方法で環境変数を使用できます。

- システム・プロファイルで変数を設定して、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

ユーザーが設定しなかった変数については、WebSphere MQ はデフォルト値を使用します。

コマンドは、特に指定のない限り、すべての WebSphere MQ MQI クライアント・プラットフォームで使用できます。

それぞれの環境変数について、現在の設定値を表示したり、変数の値をリセットするには、ご使用のプラットフォームに関連するコマンドを使用してください。

以下に例を示します。

環境変数の値の設定またはリセット		
機能	コマンド	
	Windows	UNIX and Linux システム
変数を削除する	SET MQSERVER=	unset MQSERVER
現在の設定値を表示する	SET MQSERVER	echo \$MQSERVER
セッションのすべての環境変数を表示する	set	set

個々の変数についての詳細は、以下のサブトピックを参照してください。

関連概念

[127 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。

関連資料

環境変数

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、次のコマンドのうちの 1 つを使用します。

- Windows の場合:

```
SET MQCCSID=number
```

- UNIX and Linux システムの場合:

```
export MQCCSID=number
```

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。

WebSphere MQ での証明書妥当性検査ポリシーについて詳しくは、[WebSphere 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 の場合:

```
SET MQCERTVPOL=value
```

- UNIX and Linux システムの場合:

```
export MQCERTVPOL=value
```

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、WebSphere MQ MQI クライアント・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 の場合: `MQ_INSTALLATION_PATH`

-  UNIX  Linux の場合: `/var/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 の場合: `datapath\@ipcc`

-  UNIX  Linux システムの場合: `datapath/@ipcc`

または:

-  Windows の場合: `MQ_INSTALLATION_PATH\data\qmgrs\qmgrname\@ipcc`

-  UNIX  Linux システムの場合: `/prefix/qmgrs/qmgrname/@ipcc`

ここで、

- `MQ_INSTALLATION_PATH` は、IBM WebSphere 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。接頭部は通常、`/var/mqm` UNIX および Linux プラットフォームの場合です。
- `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・スタンザは、UNIX、および Linux 上の `mqs.ini` ファイル、および Windows 上のレジストリーで定義されます。

注:

1. `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。
2. `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。
3. 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。
4. 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` が設定されている場合は、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。
5. 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。
6. 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。
7. 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。
8. `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** Windows の場合:

```
SET MQCHLLIB=pathname
```

以下に例を示します。

```
SET MQCHLLIB=C:\wmqtest
```

-   UNIX and Linux システムの場合:

```
export MQCHLLIB=pathname
```

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、[117 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 の場合:

```
SET MQCHLTAB=filename
```

- UNIX and Linux システムの場合:

```
export MQCHLTAB=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。

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。"MQIPADDR_IPV4" または "MQIPADDR_IPV6" が考えられるストリング値です。これらの値は、ALTER QMGR IPADDRV において、IPV4 および IPV6 と同様の意味があります。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 の場合:

```
SET MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6
```

- UNIX and Linux システムの場合:

```
export MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6
```

MQNAME

MQNAME は、WebSphere 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、[88 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
```

WebSphere MQ MQI クライアント上で複数の WebSphere 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、WebSphere 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。

SSL 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。SSL チャンネルの定義方法について詳しくは、[SSL 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 の場合:

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

- UNIX and Linux システムの場合:

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

TransportType は、ご使用の IBM WebSphere 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、WebSphere 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 の場合:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName(PortNumber)
```

- UNIX and Linux システムの場合:

```
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
```

- 95 ページの『TCP/IP リスナーの使用』の説明に従って、WebSphere MQ をサービス・ファイルに追加します。

SPX デフォルト・ソケット

SPX の場合、WebSphere MQ は、デフォルトにより、チャンネルがソケット 5E86 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。WebSphere 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、WebSphere MQ MQI クライアント・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。

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
```

MQCONN または MQCONNX 要求はすべて、定義したチャンネルを使用しようとします。ただし、MQCONNX に提供されている MQCNO 構造体から MQCD 構造体が参照されている場合は例外です。この場合、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チャンネルより優先されます。

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 の場合:

```
SET MQSERVER=
```

- UNIX and Linux システムの場合:

```
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 システムの場合:

```
SET MQSSLCRYP=string
```

- UNIX and Linux システムの場合:

```
export MQSSLCRYP=string
```

関連資料

[ALTER QMGR command SSLCRYP パラメーター](#)

MQSSLFIPS

MQSSLFIPS は、暗号化が WebSphere MQ で実行される場合に、FIPS 認定のアルゴリズムのみが使用され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 システムの場合:

```
SET MQSSLFIPS=YES|NO
```

- UNIX and Linux システムの場合:

```
export MQSSLFIPS=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 システムの場合:

```
SET MQSSLKEYR=pathname
```

- UNIX and Linux システムの場合:

```
export MQSSLKEYR=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 システムの場合:

```
SET MQSSLPROXY=string
```

- 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か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。

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 は、秘密鍵が再ネゴシエーションされる前に、SSL チャンネルまたは TLS チャンネルで送信および受信された非暗号化バイト数を表しています。

秘密鍵の再ネゴシエーションについては、[SSL および 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。SSL または 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、SSL または 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、SSL または 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 システムの場合:

```
SET MQSSLRESET=integer
```

- UNIX and Linux システムの場合:

```
export MQSSLRESET=integer
```

キューに入れられたパブリッシュ/サブスクライブの制御

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。

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。

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 23. 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

WebSphere 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

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
- 有効

注: Version 6.0 から移行した場合で、かつアップグレードしたキュー・マネージャーで作業する場合は、**strmqbrk** を使用して、Version 6.0 のパブリッシュ/サブスクライブ・ブローカー状態を移行する必要があります。これは z/OS には適用されません。

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、WebSphere MQ エクスプローラーまたは **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

WebSphere 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

始める前に

キューに入れられたパブリッシュ/サブスクライブは推奨されていません。

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
- 無効化
- 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、WebSphere MQ エクスプローラーまたは **runmqsc** コマンドを使用します。

例

```
ALTER QMGR PSMODE(COMPAT)
```

ストリームの追加

Version 6.0 キュー・マネージャーから移行されたストリームと共存するよう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

トピック、ストリームおよびトピックを読み、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。

このタスクについて

PCF コマンド、**runmqsc**、または IBM WebSphere 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. Version 6.0 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. Version 6.0 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. キューの名前を名前リスト SYSTEM.QPUBSUB.QUEUE.NAMELIST に追加します
4. パブリッシュ/サブスクライブ階層内にある Version 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' の追加中

ストリーム 'Sport' を共有する例では、Version 6.0 キュー・マネージャーと Version 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作業しています。Version 6.0 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。この例は、'Sport' という Version 7.1 キュー・マネージャー上に、バージョン 6 ストリーム 'Sport' と共有されるトピック・ストリング 'Sport' を持つキューとトピックを作成する方法を示しています。

Version 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。Version 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。

Version 6.0 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, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
define qlocal('Sport')
  1 : define qlocal('Sport')
AMQ8006: WebSphere MQ queue created.
define topic('Sport') topicstr('Sport')
  2 : define topic('Sport') topicstr('Sport')
AMQ8690: WebSphere 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: WebSphere 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 Version 6.0 である場合は、Version 6.0 のブローカーとして管理してください。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 Version 7.1 である場合は、階層内の Version 7.1 の各キュー・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 Version 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

IBM WebSphere MQ Version 7.1 では、キューに入れられたパブリッシュ/サブスクライブの使用は推奨されていません。

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。

このタスクについて

接続先の Version 6.0 キュー・マネージャーからストリームを削除する方法については、[ストリームの削除 \(v6.0 資料の ps11870 .htm\)](#) を参照してください。

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

1. 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 Version 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。
2. 他のすべての接続されている Version 6.0 以前のキュー・マネージャーから、ストリームを削除します。

サブスクリプション・ポイントの追加

IBM WebSphere MQ イベント・ブローカーまたは IBM WebSphere MQ メッセージ・ブローカーから **migmbbrk** によって移行されなかったサブスクリプション・ポイントを追加する方法。IBM WebSphere MQ イベント・ブローカーまたは IBM WebSphere MQ メッセージ・ブローカー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・ポイントで拡張します。

始める前に

1. IBM WebSphere MQ イベント・ブローカーおよび IBM WebSphere MQ メッセージ・ブローカーの Version 6.0 から IBM WebSphere MQ Version 7.1 への移行を完了します。
2. サブスクリプション・ポイントが `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST` にまだ定義されていないことを確認します。
3. 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

既存の IBM WebSphere MQ イベント・ブローカー・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す。新しい IBM WebSphere MQ Version 7.1 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。

移行時に使用されていなかったサブスクリプション・ポイントは、**migmbbrk** によって移行されていない可能性があります。

IBM WebSphere MQ イベント・ブローカーから移行された既存のキュー型パブリッシュ/サブスクライブ・プログラムに、サブスクリプション・ポイントを追加することができます。

サブスクリプション・ポイントは、IBM WebSphere MQ Version 6.0 以前から移行された 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・プログラムでは機能しません。

IBM WebSphere MQ Version 7.1 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

次の例は、2つのサブスクリプション・ポイント USD および GBP を追加する **runmqsc** コマンド・ファイルを示しています。

```
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。

ブローカー階層へのキュー・マネージャーの接続

ローカル・キュー・マネージャーを親キュー・マネージャーに接続して、ブローカー階層を変更することができます。

始める前に

1. キュー型パブリッシュ/サブスクライブ・モードを有効にします。[キューに入れられたパブリッシュ/サブスクライブの開始](#)を参照してください。
2. この変更は、IBM WebSphere MQ 接続を使用して親キュー・マネージャーに伝搬されます。この接続を確立する方法は2とおりです。
 - キュー・マネージャーを IBM WebSphere MQ クラスターに接続します。[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。
 - 親キュー・マネージャーと同じ名前の伝送キュー、つまりキュー・マネージャー別名を使用して、Point-to-Point のチャンネル接続を確立する。Point-to-Point チャンネル接続の確立方法については、[WebSphere MQ 分散メッセージング技法](#)を参照してください。

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。

分散パブリッシュ/サブスクライブは、キュー・マネージャー・クラスターおよびクラスター化されたトピック定義を使用して実装されます。IBM WebSphere MQ Version 6.0 および WebSphere Message Broker Version 6.1 および WebSphere Event Broker Version 6.1 以前とのインターオペラビリティのために、キューに入れられたパブリッシュ/サブスクライブ・モードが有効になっている限り、Version 7.1 以降のキュー・マネージャーをブローカー階層に接続することもできます。

手順

```
ALTER QMGR PARENT(PARENT)
```

例

最初の例は、QM2 を QM1 の子として接続し、QM2 にその接続について照会する方法を示しています。

```
C:>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2
alter qmgr parent(QM1)
  1 : alter qmgr parent(QM1)
AMQ8005: WebSphere 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, 2024. 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)
```

次のタスク

接続されたキュー・マネージャー上のパブリッシャーおよびサブスクライバーが使用できる1つのブローカーまたはキュー・マネージャーに、トピックを定義することができます。詳しくは、[管理トピックの定義](#)を参照してください。

関連概念

[ストリームおよびトピック](#)

[WebSphere MQ パブリッシュ/サブスクライブ・メッセージングの概要](#)

関連資料

[DISPLAY PUBSUB](#)

ブローカー階層からのキュー・マネージャーの切断

ブローカー階層内の親キュー・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, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2.
  1 : alter qmgr parent('')
AMQ8005: WebSphere MQ queue manager changed.
  2 : display pubsub type(child)
AMQ8147: WebSphere MQ object not found.
```

```
display pubsub type(parent)
3 : display pubsub type(parent)
AMQ8147: WebSphere 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。

キュー・マネージャー・クラスターの構成

クラスターの動作とクラスター構成の設計について調べるには、および単純なクラスターのセットアップ方法の例については、このトピックにあるリンクを使用してください。

始める前に

クラスター化の概念の概要については、以下のトピックを参照してください。

- [クラスターが機能する仕組み](#)
- [161 ページの『クラスター化と分散キューイングとの比較』](#)
- [164 ページの『クラスターのコンポーネント』](#)

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。まず、クラスター内のどのキュー・マネージャーがクラスター情報の完全リポジトリを保持するかを決定する必要があります。作成するどのキュー・マネージャーもクラスター内で機能できます。この目的のためのキュー・マネージャーはいくつでも選択できますが、理想的な数は2つです。完全リポジトリを保持するキュー・マネージャーの選択については、[178 ページの『完全リポジトリを保持するクラスター・キュー・マネージャーの選択方法』](#)を参照してください。

クラスターの設計について詳しくは、以下のトピックを参照してください。

- [179 ページの『クラスターの編成』](#)
- [179 ページの『クラスターの命名規則』](#)
- [180 ページの『クラスターのオーバーラップ』](#)

例

最小規模のクラスターでは、それに含まれるキュー・マネージャーは2つだけです。この場合は、両方のキュー・マネージャーが完全リポジトリを保有します。クラスターのセットアップに必要な定義がごくわずかで済むにもかかわらず、各キュー・マネージャーに高度の自律性があります。

[160 ページの図 21](#) は、QM1 と QM2 という2つのキュー・マネージャーを持つ DEMOCLSTR というクラスターを示しています。

DEMOCLSTR

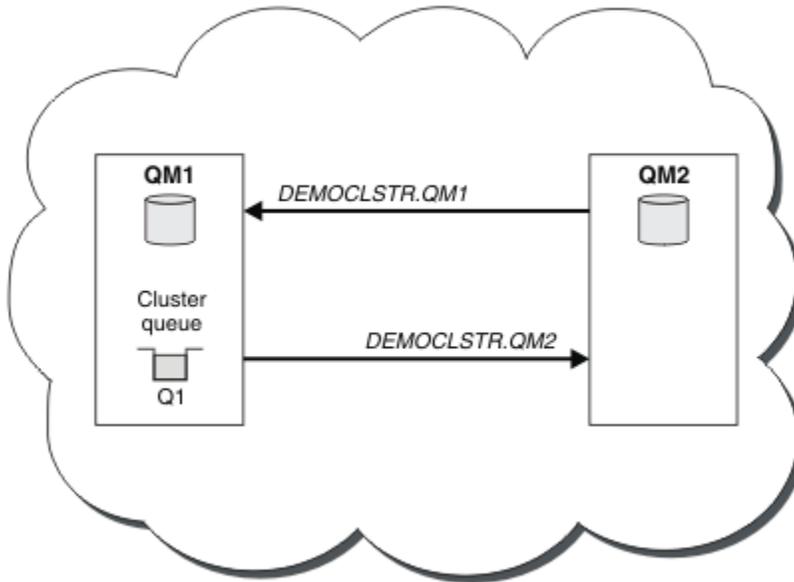


図 21. 2つのキュー・マネージャーで構成される小規模なクラスターの例

- キュー・マネージャーには、LONDON や NEWYORK などのロング・ネームが付けられています。拡張タスクとワークロード・バランシング・タスクでも同じ名前が使用されます。IBM WebSphere MQ for z/OS では、キュー・マネージャー名は 4 文字までに制約されています。
- キュー・マネージャーの名前は、各キュー・マネージャーが別々のマシン上にあることを示すものになっていますが、すべてのキュー・マネージャーが同一マシン上にある場合でも、これらのタスクを実行することができます。
- タスクでは、IBM WebSphere MQ スクリプト・コマンドを、システム管理者が **MQSC** コマンドを使用して入力した場合と同様に使用します。より簡単な IBM WebSphere MQ エクスプローラーを使用するなど、コマンドの入力方法は他にもあります。WebSphere MQ スクリプト・コマンドを使用することのポイントは、どのような IBM WebSphere MQ コマンドがタスクで使用されるかを示すことにあります。

類似のクラスター例のセットアップ手順については、186 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。

次のタスク

クラスターの構成と処理について詳しくは、以下のトピックを参照してください。

- [183 ページの『クラスター内での通信の確立』](#)
- [186 ページの『IBM WebSphere MQ クラスターの管理』](#)
- [251 ページの『クラスターとの間のルーティング・メッセージ』](#)
- [266 ページの『クラスターによるワークロードの管理』](#)

クラスターの構成に役立つ詳細情報は、181 ページの『[クラスター化のヒント](#)』を参照してください。

関連概念

[クラスター](#)

アクセス制御と複数のクラスター伝送キュー

アプリケーションがメッセージをリモート・クラスター・キューに入れるタイミングをチェックするモードを 3 つの中から選択します。これらのモードはそれぞれ、リモートでのクラスター・キューに対するチェック、ローカルでの `SYSTEM.CLUSTER.TRANSMIT.QUEUE` に対するチェック、クラスター・キューまたはクラスター・キュー・マネージャーのローカル・プロファイルに対するチェックを行います。

IBM WebSphere MQ では、ユーザーにメッセージをリモート・キューに入れるためのアクセス権があることをローカルでチェックするか、またはローカルとリモートでチェックするかを選択できます。典型的な IBM WebSphere MQ アプリケーションはローカル・チェックのみを使用し、ローカル・キュー・マネージャーで行われるアクセス・チェックを信用するリモート・キュー・マネージャーに依存します。リモート・チェックが使用されない場合、メッセージはリモート・メッセージ・チャンネル・プロセスの権限でターゲット・キューに入れられます。リモート・チェックを使用するには、受信側チャンネルの書き込み権限をコンテキスト・セキュリティに設定する必要があります。

ローカル・チェックは、アプリケーションがオープンするキューに対して行われます。分散キューイングでは、アプリケーションは一般にリモート・キュー定義をオープンするため、アクセス・チェックはリモート・キュー定義に対して行われます。メッセージが完全なルーティング・ヘッダーで書き込まれている場合には、伝送キューに対するチェックが行われます。アプリケーションがローカル・キュー・マネージャー上にはないクラスター・キューをオープンした場合、チェック対象のローカル・オブジェクトはありません。アクセス制御チェックは、クラスター伝送キュー SYSTEM.CLUSTER.TRANSMIT.QUEUE に対して行われます。複数のクラスター伝送キューがある場合でも (Version 7.5 以降)、リモート・クラスター・キューのローカル・アクセス制御チェックは SYSTEM.CLUSTER.TRANSMIT.QUEUE に対して行われます。

ローカル・チェックとリモート・チェックのどちらを選択するかは、2つの極端な選択となります。リモートでは、チェックが微細化されます。クラスター・キューに書き込むには、すべてのユーザーがすべてのキュー・マネージャーにアクセス制御プロファイルを持っている必要があります。ローカルでは、チェックが粗視化されます。すべてのユーザーに必要なのは、接続先のキュー・マネージャー上のクラスター伝送キューのアクセス制御プロファイルだけです。そのプロファイルを使用して、任意のクラスター内の任意のキュー・マネージャーにある任意のクラスター・キューにメッセージを書き込むことができます。

Version 7.1 以降は、管理者はもう 1 つの方法でクラスター・キューのアクセス制御をセットアップできるようになっています。 **setmqaut** コマンドを使用して、クラスター内の任意のキュー・マネージャー上にクラスター・キューのセキュリティ・プロファイルを作成できます。このプロファイルは、キュー名だけを指定してリモート・クラスター・キューをローカルでオープンする場合に有効になります。リモート・キュー・マネージャーのプロファイルをセットアップすることもできます。その場合、キュー・マネージャーは、完全修飾名を指定して、クラスター・キューをオープンしたユーザーのプロファイルをチェックできます。

新しいプロファイルは、キュー・マネージャーのスタンザ **ClusterQueueAccessControl** を RQMName に変更した場合にのみ機能します。デフォルトは Xmitq です。既存のアプリケーションが使用するすべてのクラスター・キューのプロファイルを作成する必要があります。これらのプロファイルを作成せずに、スタンザを RQMName に変更すると、アプリケーションが失敗する可能性があります。

ヒント: Version 7.1 でクラスター・キューのアクセス・チェックに対して行われた変更は、リモート・キューイングには適用されません。アクセス・チェックは、引き続きローカル定義に対して行われます。これらの変更は、クラスター・キューおよびクラスター・トピックでのアクセス・チェックを構成する場合にも、同じ手法に従えることを意味します。

関連概念

287 ページの『[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](#)』

クラスター内のキュー・マネージャー間のメッセージ・フローは、分離することができます。さまざまなクラスター送信側チャンネルによって転送されるメッセージを、それぞれに異なるクラスター伝送キューに配置できます。この手法は、単一のクラスターでも、オーバーラップするクラスターでも使用できます。このトピックでは、使用する手法を選択する際に参考となる例およびベスト・プラクティスを説明します。

クラスター化と分散キューイングとの比較

キュー・マネージャーに接続するために定義する必要があるコンポーネントを、分散キューイングを使用する場合とクラスター化を使用する場合で比較します。

クラスターを使用しない場合、キュー・マネージャーはそれぞれ独立したプログラムとなり、分散キューイングでデータを送受信します。この場合、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、以下のものを定義する必要があります。

- 伝送キュー
- リモート・キュー・マネージャーへのチャンネル

162 ページの図 22 に、分散キューイングの場合に必要なコンポーネントを示します。

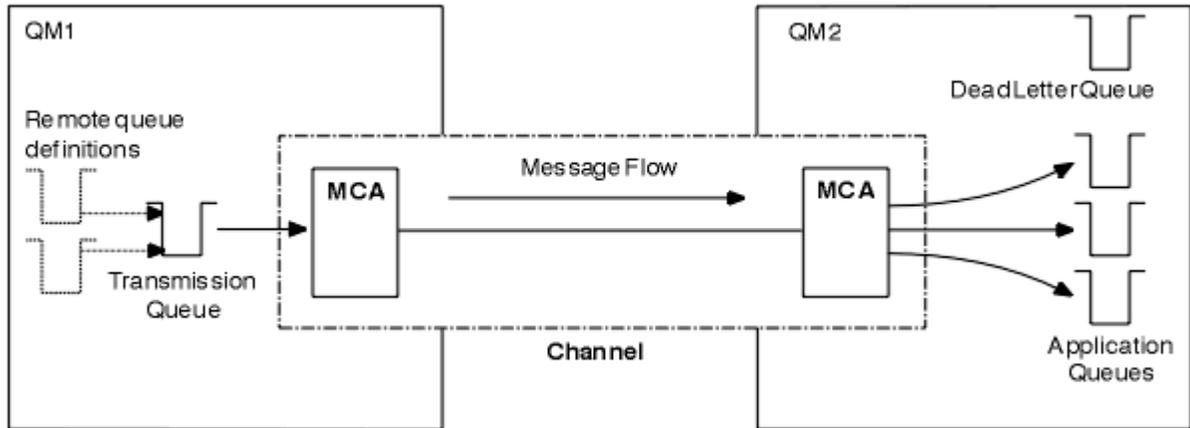


図 22. 分散キュー

複数のキュー・マネージャーを1つのクラスターにグループ化した場合、どのキュー・マネージャー上のキューも、そのクラスター内にある他のキュー・マネージャーから使用できるようになります。どのキュー・マネージャーも、同じクラスター内の別のキュー・マネージャーへ明示的な定義なしでメッセージを送信することができます。宛先ごとにチャンネル定義、リモート・キュー定義、伝送キューを指定しません。クラスターを構成するキュー・マネージャーには伝送キューがそれぞれ1つずつ設定されています。この伝送キューにより、同じ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送信することができます。したがって、クラスター内の各キュー・マネージャーに定義する必要があるのは次の2つだけです。

- クラスター受信側チャンネル。メッセージを受信するために使用します。
- クラスター送信側チャンネル。自分自身の情報を取り込んだり、クラスターを把握したりします。

設定時の定義項目に関するクラスターと分散キューイングとの比較

以降では、4つのキュー・マネージャーで構成されるネットワークを設定する場合を例にして説明します。各キュー・マネージャーにはそれぞれ2つのキューが格納されているものとします。162 ページの図 23 は、そのようなキュー・マネージャーで構成されるネットワークの例です。最初に、これらのキュー・マネージャーを分散キューイングで接続する場合に、いくつの項目を定義する必要があるのかについて説明します。次に、同じネットワークをクラスターとして設定する場合には、いくつの項目を定義する必要があるのかを比較します。

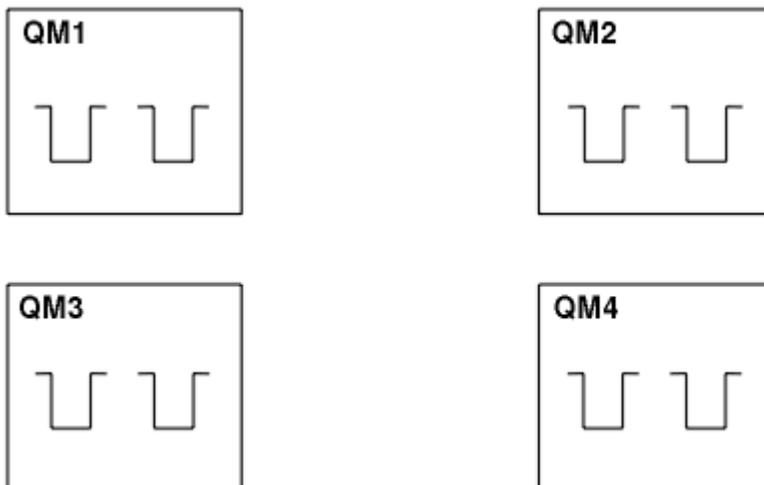


図 23. 4つのキュー・マネージャーで構成されるネットワークの例

分散キューイングによりネットワークを設定する場合に定義する項目

以下の表は、162 ページの図 22 に示すネットワークを分散キューイングにより設定する場合に定義する項目を示しています。

説明	各キュー・マネージャーに定義する数	合計数
送信側チャンネル定義: 他のキュー・マネージャーにメッセージを送信するときに使用するチャンネルを示します。	3	12
受信側チャンネル定義: 他のキュー・マネージャーからメッセージを受信するときに使用するチャンネルを示します。	3	12
伝送キュー定義: 他のキュー・マネージャーにメッセージを送信するときに使用する伝送キューを示します。	3	12
ローカル・キュー定義: 個々のローカル・キューを示します。	2	8
リモート・キュー定義: メッセージの書き込み先となる個々のリモート・キューを示します。	6	24

汎用受信側チャンネルを定義することにより、定義の数を減らすこともできます。各キュー・マネージャーにつき最大で 17 の定義、つまり、ネットワークの合計で 68 の定義が必要です。

クラスター方式によりネットワークを設定する場合に定義する項目

162 ページの図 22 に示すネットワークをクラスター方式により設定する場合は、以下の表に示す項目を定義する必要があります。

説明	各キュー・マネージャーに定義する数	合計数
クラスター送信側チャンネル定義: リポジトリ・キュー・マネージャーにメッセージを送信するときに使用するチャンネルを示します。	1	4
クラスター受信側チャンネル定義: クラスター内にある他のキュー・マネージャーからメッセージを受信するときに使用するチャンネルを示します。	1	4
ローカル・キュー定義: 個々のローカル・キューを示します。	2	8

これらのキュー・マネージャーで構成されるクラスターを (2 つの完全リポジトリと共に) 設定する場合、各キュー・マネージャーにつき 4 つの定義、つまり、合計で 16 の定義が必要です。また、いずれか 2 つのキュー・マネージャーの定義を修正して、それらのキュー・マネージャーをそのクラスターの完全リポジトリ・キュー・マネージャーにする必要があります。

1 つの CLUSSDR チャンネル定義と 1 つの CLUSRCVR チャンネル定義のみ必要です。クラスターを定義すれば、(リポジトリ・キュー・マネージャー以外の) キュー・マネージャーを追加したり削除したりしても、他のキュー・マネージャーが壊れることはありません。

クラスターを使用すると、多数のキュー・マネージャーで構成されるネットワークを設定する場合に、必要な定義の数が減ります。

定義する項目の数が少なければ、エラーが発生することも少なくなります。以下に例を示します。

- オブジェクト名が、例えば送信側と受信側が対になっているチャンネル名と常に一致します。

- チャンネル定義に指定された伝送キュー名は、正しい伝送キュー定義か、またはリモート・キュー定義に指定された伝送キュー名と常に一致します。
- QREMOTE 定義は、リモート・キュー・マネージャーにある正しいキューを常に指し示します。

クラスターをいったん設定したあとは、クラスター内のあるキュー・マネージャーから別のキュー・マネージャーにクラスター・キューを移動しても、その他のキュー・マネージャーでシステム管理上の作業を行う必要がありません。したがって、チャンネル、リモート・キュー、または伝送キューの定義を削除し忘れたり、変更し忘れることはありません。新しいキュー・マネージャーをクラスターに追加する場合にも、既存のネットワーク環境を破壊することなく追加できます。

クラスターのコンポーネント

クラスターは、キュー・マネージャー、クラスター・リポジトリ、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。

各クラスター・コンポーネントについて詳しくは、以下のサブトピックを参照してください。

関連概念

クラスター

[161 ページの『クラスター化と分散キューイングとの比較』](#)

キュー・マネージャーに接続するために定義する必要があるコンポーネントを、分散キューイングを使用する場合とクラスター化を使用する場合で比較します。

[186 ページの『IBM WebSphere MQ クラスターの管理』](#)

IBM WebSphere MQ クラスターの作成、拡張、および保守を行うことができます。

関連タスク

[159 ページの『キュー・マネージャー・クラスターの構成』](#)

クラスターの動作とクラスター構成の設計について調べるには、および単純なクラスターのセットアップ方法の例については、このトピックにあるリンクを使用してください。

[186 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。

クラスター・リポジトリ

リポジトリとは、クラスターを構成する各キュー・マネージャーについての情報の集まりを指します。

リポジトリ情報には、キュー・マネージャーの名前、場所、チャンネル、そのキュー・マネージャーがホストするキュー、および他の情報が含まれます。この情報は、SYSTEM.CLUSTER.REPOSITORY.QUEUE というキューにメッセージ形式で格納されています。このキューは、デフォルト・オブジェクトの1つです。このキューは、WebSphere MQ for z/OS の場合を除き、WebSphere MQ キュー・マネージャーの作成時に定義されます。

通常は、クラスターの中の2つのキュー・マネージャーに完全リポジトリが保持されます。それ以外のすべてのキュー・マネージャーには、部分リポジトリが保持されます。

完全リポジトリおよび部分リポジトリ

クラスター内の各キュー・マネージャーについての全情報をホストしているキュー・マネージャーには、完全リポジトリが含まれます。クラスター内の他のキュー・マネージャーには、完全リポジトリに格納されている情報のサブセットが入っている部分リポジトリが含まれます。

部分リポジトリには、メッセージを交換する必要があるキュー・マネージャーに関する情報だけが格納されています。キュー・マネージャーが必要な情報に対する更新情報を要求すると、その情報が変更されている場合には、完全リポジトリ・キュー・マネージャーは要求元のキュー・マネージャーに新しい情報を送信します。通常、部分リポジトリには、キュー・マネージャーがクラスター内で処理を実行するのに必要な情報がすべて含まれています。その他の情報が必要になると、キュー・マネージャーは完全リポジトリにその情報を照会し、これにより、部分リポジトリを更新します。キュー・マネージャーは、

SYSTEM.CLUSTER.COMMAND.QUEUE というキューを使用して完全リポジトリに更新情報を要求し、これを受信します。このキューはデフォルト・オブジェクトの1つです。

クラスター・キュー・マネージャー

クラスター・キュー・マネージャーは、クラスターのメンバーであるキュー・マネージャーです。

キュー・マネージャーは、複数のクラスターのメンバーにすることができます。各クラスター・キュー・マネージャーには、そのクラスターがメンバーとなっているすべてのクラスター全体で固有の名前がなければなりません。

1つのクラスター・キュー・マネージャーで複数のキューをホストすることができます。その場合、そのキュー・マネージャーはそれらのキューを同じクラスター内にあるその他のキュー・マネージャーに通知します。しかし、クラスター・キュー・マネージャーで複数のキューをホストしたり、それらのキューを通知する必要はありません。クラスター・キュー・マネージャーは、メッセージをクラスターに送信し、そのメッセージに対する応答のうち、通知されたキューではなくそのクラスター・キュー・マネージャーに送るよう明示的に指定された応答だけを受信できます。

WebSphere MQ for z/OS では、クラスター・キュー・マネージャーはキュー共有グループのメンバーになることができます。この場合、キュー・マネージャーは同じキュー共有グループ内の他のキュー・マネージャーとキュー定義を共有します。

クラスター・キュー・マネージャーは自律型のプログラムです。各クラスター・キュー・マネージャーに定義されたキューおよびチャネルをそのキュー・マネージャーが完全に制御することができます。各キュー・マネージャーの定義を別のキュー・マネージャーによって変更することはできません(同じキュー共有グループ内のキュー・マネージャーであれば可能です)。リポジトリ・キュー・マネージャーは、クラスター内の他のキュー・マネージャーでの定義を制御しません。リポジトリ・キュー・マネージャーは、必要に応じて使用できる、すべての定義の完全なセットを保持します。クラスターは、キュー・マネージャーのフェデレーションです。

クラスター・キュー・マネージャーの定義を作成したり変更したりすると、その情報は完全リポジトリ・キュー・マネージャーに送信されます。クラスター内の他のリポジトリは後で更新されます。

完全リポジトリ・キュー・マネージャー

完全リポジトリ・キュー・マネージャーとは、クラスターのリソースの完全な表現を保持するクラスター・キュー・マネージャーです。可用性を確保するために、各クラスターに2つ以上の完全リポジトリ・キュー・マネージャーをセットアップしてください。完全リポジトリ・キュー・マネージャーは、クラスター内にあるその他のキュー・マネージャーから送信された情報を受信し、そのリポジトリを更新します。そして、互いにメッセージを交換することにより、すべての完全リポジトリ・キュー・マネージャーでクラスターに関する情報が常に最新に保たれるようにします。

キュー・マネージャーおよびリポジトリ

すべてのクラスターには、クラスター内のキュー・マネージャー、キュー、およびチャネルに関する情報のフル・リポジトリを保持する、少なくとも1つの(できれば2つの)キュー・マネージャーがあります。また、クラスター内のその他のキュー・マネージャーからこの情報を更新する要求が発行された場合は、その要求も完全リポジトリに格納されます。

他のキュー・マネージャーは、それぞれ部分リポジトリを保持します。このリポジトリには、通信するために必要なキューおよびキュー・マネージャーのサブセットに関する情報が含まれています。これらのキュー・マネージャーは、別のキューまたはキュー・マネージャーに初めてアクセスする必要があるときに照会を行うことによって、それ自身の部分リポジトリを作成します。キュー・マネージャーは、そのキューまたはキュー・マネージャーに関する新しい情報があれば通知するように要求します。

各キュー・マネージャーは、リポジトリ情報を SYSTEM.CLUSTER.REPOSITORY.QUEUE というキューにメッセージ形式で格納します。そして、各キュー・マネージャー間でメッセージ形式のリポジトリ情報を交換するときには SYSTEM.CLUSTER.COMMAND.QUEUE というキューを使用します。

クラスターに参加する各キュー・マネージャーは、いずれかのリポジトリに対するクラスター送信側 (CLUSSDR) チャネルを定義します。キュー・マネージャーはクラスター内のどのキュー・マネージャーが完全リポジトリを保有しているのかをただちに認識します。これ以降、キュー・マネージャーはどのリ

ポジトリーに格納されている情報でも要求することができます。また、キュー・マネージャーが、指定されたリポジトリーに情報を送信すると、その情報は別のリポジトリーがあればそのリポジトリーにも送信されます。

完全リポジトリーをホストしているキュー・マネージャーが、そのプログラムに接続されている別のキュー・マネージャーから新しい情報を受信すると、完全リポジトリーが更新されます。この新しい情報は別のリポジトリーにも送信されます。そのため、稼働していないリポジトリー・キュー・マネージャーがあると情報の更新に遅れが生じるという可能性も少なくなります。どの情報も2回送信されるので、各リポジトリーでは重複する情報を破棄する必要があります。各情報には順序番号が付いているので、各リポジトリーではその番号によって重複する情報を識別することができます。すべてのリポジトリー間では、メッセージを交換することによって互いに同期をとります。

クラスター・キュー

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・マネージャーで利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。クラスター内にあるその他のキュー・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、フルリポジトリーで、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 WebSphere 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用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

z/OS 以外では、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: 旧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 `SYSTEM.CLUSTER.TRANSMIT.QUEUES` を使用している場合は、`SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` の `SHARE/NOSHARE` オプションが **SHARE** に設定されていることを確認してください。

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 を除くすべてのキュー・マネージャーでデフォルトで作成されます。

z/OS 以外のプラットフォームでは、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、モデル・キュー `SYSTEM.CLUSTER.TRANSMIT.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、クラスター受信側チャンネルの命名規則 `ClusterName.QmgrName` に従った場合、総称名 `ClusterName.*` によってさまざまなクラスターに対するメッセージがそれぞれの伝送キューにフィルタリングして転送されます。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。WebSphere MQ は、メッセージを送信できるように、リモート・キュー定義と同等の定義を自動的に作成します。

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
```

関連資料

[ClusterChannelName \(MQCHAR20\)](#)

クラスター・チャンネル

クラスターのキュー・マネージャー用のクラスター受信側チャンネルおよびクラスター送信側チャンネルを定義する必要があります。完全リポジトリには、特に考慮が払われます。

クラスター方式の場合、メッセージは特殊なタイプのチャンネルによってクラスター・キュー・マネージャー間で配布されます。そのために、クラスター受信側チャンネル定義とクラスター送信側チャンネル定義が必要です。

クラスター送信側チャンネル: CLUSSDR

クラスター送信側チャンネルを、クラスター内の各キュー・マネージャーの完全リポジトリに手動で定義します。クラスター送信側を定義することにより、キュー・マネージャーは初めてクラスターと通信できるようになります。このチャンネル定義には、優先的にクラスター情報の送信先とする完全リポジトリ・キュー・マネージャーの名前を指定します。クラスター送信側チャンネルは、キュー・マネージャーの状況の変更をリポジトリに通知するために使用されます。例えば、キューが追加または削除された場合に使用します。また、このチャンネルは、メッセージの送信にも使用されます。

フル・リポジトリ・キュー・マネージャー自体に、お互いを指し示すクラスター送信側チャンネルがあります。フル・リポジトリ・キュー・マネージャーは、このチャンネルを使用してクラスター状況の変更を相互に通信します。

CLUSSDR チャンネル定義がどの完全リポジトリを指すかということはほとんど重要ではありません。一度接続が確立されれば、それ以降はクラスター・キュー・マネージャー・オブジェクトが必要に応じて自動的に定義されます。キュー・マネージャーはクラスター情報をすべての完全リポジトリに送信したり、メッセージをすべてのキュー・マネージャーに送信したりすることができます。

完全リポジトリ・キュー・マネージャーに設定する CLUSSDR チャンネル定義は特殊な定義です。完全リポジトリ間で交換されるすべての更新情報は、このチャンネルでのみ送受信されます。管理者は完全リポジトリのネットワークを明示的に制御します。管理者は、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 チャンネルを定義する必要があります。管理者は、完全リポジトリ・キュー・マネージャーに CLUSSDR を手動で定義する必要があります。自動定義のままにはしないでください。

クラスター送信側チャンネルを定義しなければならないのは、部分リポジトリを完全リポジトリに接続するか、2つの完全リポジトリを相互に接続する場合だけです。部分リポジトリやクラスターに含ま

れないキュー・マネージャーをアドレス指定する CLUSSDR チャンネルを手動で構成すると、AMQ9427 や AMQ9428 などのエラー・メッセージが出されることになります。

これは一時的状況として避けられない場合もありますが (例えば完全リポジトリーの位置を変更するとき)、正しくない定義はできるだけ早く削除してこれらのエラーが出されないようにする必要があります。

クラスター受信側チャンネル: CLUSRCVR

クラスター受信側チャンネル定義は、クラスター・キュー・マネージャーが同じクラスター内にある他のキュー・マネージャーからメッセージを受信することができるチャンネルの側を定義します。

クラスター受信側チャンネルでは、ローカル・リポジトリーに送信するクラスター情報を転送することもできます。キュー・マネージャーは、クラスター受信側チャンネルを定義することにより、そのキュー・マネージャーがメッセージを受信可能であることを他のクラスター・キュー・マネージャーに示します。各クラスター・キュー・マネージャーごとに、少なくとも1つのクラスター受信側チャンネルが必要です。

CLUSRCVR 定義により、他のキュー・マネージャーは対応する CLUSSDR チャンネル定義を自動的に定義できるようになります。

関連概念

169 ページの『クラスター・チャンネルの自動定義』

キュー・マネージャーがリモート宛先にメッセージを送信するためには、キュー・マネージャーのクラスター送信側チャンネルが定義されていなければなりません。キュー・マネージャーの CLUSSDR と CLUSRCVR の初期定義を行うことによってクラスターにキュー・マネージャーを導入すると、WebSphere MQ はクラスター送信側チャンネル定義を必要時に自動的に行います。自動定義のクラスター送信側チャンネルを変更することはできません。これらの動作の変更は、チャンネル自動定義出口を使用して行えます。

クラスター・チャンネルの自動定義

キュー・マネージャーがリモート宛先にメッセージを送信するためには、キュー・マネージャーのクラスター送信側チャンネルが定義されていなければなりません。キュー・マネージャーの CLUSSDR と CLUSRCVR の初期定義を行うことによってクラスターにキュー・マネージャーを導入すると、WebSphere MQ はクラスター送信側チャンネル定義を必要時に自動的に行います。自動定義のクラスター送信側チャンネルを変更することはできません。これらの動作の変更は、チャンネル自動定義出口を使用して行えます。

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定義すると、チャンネルが使用できる状態になります。自動的に定義されたチャンネルは、通常の切断インターバルのルールにより不要になったものと見なされて切断されるまで、活動状態のままです。

自動的に定義されたクラスター送信側チャンネルの属性は、受信側のキュー・マネージャーの対応するクラスター受信側チャンネル定義の属性を基にして設定されます。手動で定義されたクラスター送信側チャンネルがあったとしても、その属性は対応するクラスター受信側の定義に合わせて自動的に変更されます。例えば、CONNAME パラメーターでポート番号を指定せずに CLUSRCVR を定義し、ポート番号を指定する CLUSSDR を手動で定義したとします。手操作で定義した CLUSSDR が自動定義されたもので置き換えられると、ポート番号 (CLUSRCVR から設定された) はブランクになります。デフォルトのポート番号が使用され、チャンネルで障害が発生します。

自動定義されたクラスター送信側の定義は変更できません。

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)
```

作成した CLUSRCVR チャンネルに対応する自動定義 CLUSSDR チャンネル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)
```

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、WebSphere 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。以下の場合、クラスター環境でチャンネル自動定義出口を使用できます。

- 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
- チャネル出口の名前を変更する場合。CLUSSDR チャネル出口の名前を変更する必要があります。CLUSSDR チャネル出口名は、CLUSRCVR チャネル定義から自動生成されるからです。自動生成された名前は不適切な可能性があります。異なるプラットフォーム上にチャネルの両端がある場合は、ほぼ間違いなく不適切な名前になります。出口名の形式は、プラットフォームによって異なります。例えば、Windows の場合は SCYEXIT('drive:\path\library(secexit)') です。

z/OS 以外の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。それ以外の場合は、8 文字に切り捨てられた *library* が使用されます。例:

- /var/mqm/exits/myExit.so(MsgExit) MSGEXIT による変換
- /var/mqm/exits/myExit MYEXIT による変換
- /var/mqm/exits/myExit.so(ExitLongName) EXITLONG による変換

アウトバウンド (TCP) チャネルが特定の 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、チャネル属性 LOCLADDR を使用します。LOCLADDR は、複数のネットワーク・カードがあり、チャ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 チャネルに LOCLADDR 値を強制的に指定するために、チャネル自動定義出口を作成する必要があります。また、手動で定義された CLUSSDR チャネルにこの値を指定する必要もあります。

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 チャネルに伝搬されます。

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 チャネルの LOCLADDR にポート番号またはポート範囲を入力します。

distributed 分散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。

自動定義のクラスター送信側チャネル定義は、実際のチャネル・オブジェクトではありません。z/OS 以外の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、これらの存在を認識していません。自動定義されたクラスター送信側チャネルに対して start、stop、ping、reset、または resolve コマンドを発行しようとする、OAM はコマンドを発行したユーザーがクラスターのクラスター受信側チャ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。

クラスターが PROPCTL を使用して、WebSphere MQ バージョン 7 キュー・マネージャーから以前のレベルの WebSphere 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・ヘッダーを削除する必要がある場合は、PROPCTL を強制的に値 NONE にするチャネル自動定義出口を作成する必要があります。この出口が必要なのは、クラスター送信側チャネルの定義は、対応するクラスター受信側チャネルに基づいているからです。以前のレベルのクラスター受信側チャネルには PROPCTL 属性がないため、この属性は自動クラスター送信側チャネルによって COMPAT に設定されます。手動クラスター送信側チャネルでの設定にかかわらず、この属性は COMPAT に設定されます。

関連資料

[ローカル・アドレス \(LOCLADDR\)](#)

デフォルトのクラスター・オブジェクト

WebSphere MQ クラスターを使用する場合に、デフォルトのクラスター・オブジェクトを作成します。これらのオブジェクトは、キュー・マネージャーを定義するときに自動的に作成されるデフォルト・オブジェクトのセットに含まれます。

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。

SYSTEM.CLUSTER.COMMAND.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.COMMAND.QUEUE という、メッセージを完全リポジトリに転送するために使用するローカル・キューがあります。メッセージには、キュー・マネージャーに関する新しい情報や変更された情報、また他のキュー・マネージャーに関する情報の要求が格納されます。通常、SYSTEM.CLUSTER.COMMAND.QUEUE は空です。

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、サービスを目的として、クラスター状態情報のヒストリーを保管するために使用します。

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。ヒストリー収集を抑止するには、設定を PUT(DISABLED) に変更します。

SYSTEM.CLUSTER.REPOSITORY.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.REPOSITORY.QUEUE というローカル・キューがあります。このキューはすべての完全リポジトリ情報の保管に使用します。このキューは通常は空ではありません。

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 DEFEXMITQ 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。

SYSTEM.DEF.CLUSRCVR

それぞれのクラスターには、SYSTEM.DEF.CLUSRCVR というデフォルトの CLUSRCVR チャンネル定義があります。SYSTEM.DEF.CLUSRCVR は、クラスター内のキュー・マネージャーにクラスター受信側チャンネルを作成するときに指定しないすべての属性にデフォルト値を提供するために使用されます。

SYSTEM.DEF.CLUSSDR

それぞれのクラスターには、SYSTEM.DEF.CLUSSDR というデフォルトの CLUSSDR チャンネル定義があります。SYSTEM.DEF.CLUSSDR は、クラスター内のキュー・マネージャーにクラスター送信側チャンネルを作成するときに指定しないすべての属性にデフォルト値を提供するために使用されます。

クラスター伝送キューとクラスター送信側チャンネル

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。

クラスター送信側チャンネルを表示すると、クラスター送信側チャンネルが伝送キューに関連付けられていることが分かります。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。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。172 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](#)』デフォルト伝送キューを選択するプロセスについて説明します。173 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](#)』では、手動で定義された伝送キューを選択するプロセスについて説明します。

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、174 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。Version 7.5 クラスター接続が単一クラスター伝送キュー SYSTEM.CLUSTER.TRANSMIT.QUEUE を共有するまで。Version 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。291 ページの『[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 208 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。

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、173 ページの図 24 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。

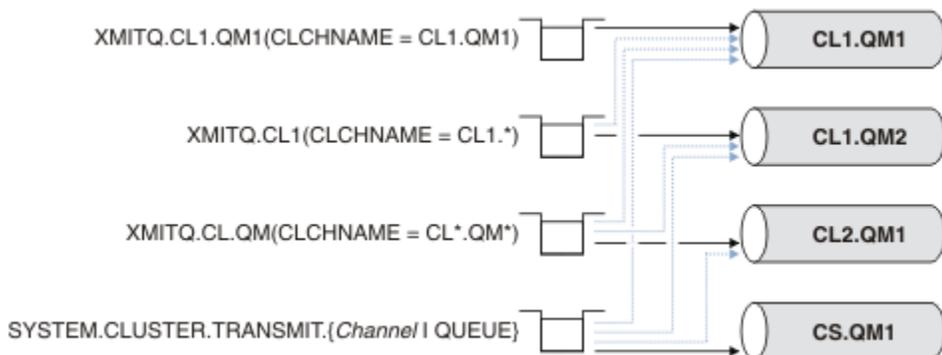


図 24. 伝送キューとクラスター送信側チャンネルの優先順位

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。173 ページの図 24 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、173 ページの図 24 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伝送キュー用に作成したシステム・クラスター伝送キュー SYSTEM.CLUSTER.TRANSMIT.channelName 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・アクティビティは、通常の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。次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。"AMQ7341 チャンネル *ChannelName* 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、[160 ページの『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。

[176 ページの図 25](#) 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、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、トラフィックを異なるクラスター送信側チャンネルに分割する方法を検討してから、特定のチャンネルへのメッセージを異なる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。

176 ページの図 25 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。

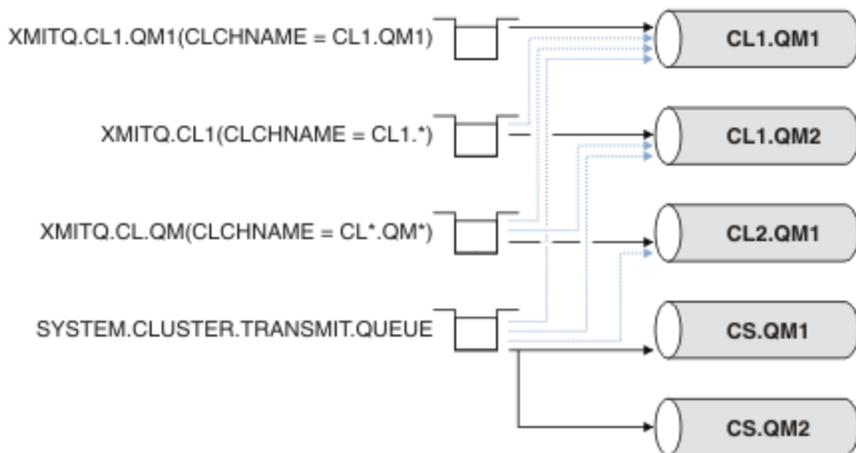


図 25. 複数のクラスター送信側チャンネル

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ません。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。それらの相関 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(174 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

287 ページの『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』

クラスター内のキュー・マネージャー間のメッセージ・フローは、分離することができます。さまざまなクラスター送信側チャンネルによって転送されるメッセージを、それぞれに異なるクラスター伝送キューに配置できます。この手法は、単一のクラスターでも、オーバーラップするクラスターでも使用できます。このトピックでは、使用する手法を選択する際に参考となる例およびベスト・プラクティスを説明します。

411 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

291 ページの『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の選択について手順を追って説明します。1つの共通デフォルト・キュー、個々のデフォルト・キュー、または手動で定義したキューを構成できます。複数のクラスター伝送キューの構成は、z/OS 以外のプラットフォームに適用されます。

219 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。

199 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。

204 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。

208 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。

完全リポジトリを保持するクラスター・キュー・マネージャーの選択方法

それぞれのクラスターで、完全リポジトリを保持するためのキュー・マネージャーを少なくとも1つ、できれば2つ選択する必要があります。きわめて例外的な状況を除き、完全リポジトリは2つあれば十分です。キュー・マネージャーは、可能であれば、永続的に接続された堅固なプラットフォームでホストされ、2台同時に停止しないものを選択します。地理的に中心位置にあるものが理想です。また、システムを完全リポジトリ・ホスト専用にし、他の作業に使用しないシステムにすることも検討してください。

完全リポジトリとは、クラスターの状態に関するすべての情報を保持するキュー・マネージャーのことです。この情報を共有するために、それぞれの完全リポジトリは、クラスター内の他の完全リポジトリそれぞれに CLUSSDR チャンネル (およびそれらに対応する CLUSRCVR 定義) で接続されます。これらのチャンネルは手動で定義する必要があります。

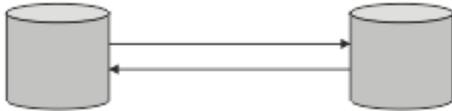


図 26. 接続された 2 つの完全リポジトリ

クラスター内の他のキュー・マネージャーは、それぞれが現在認識している範囲で、クラスターの状態に関する情報を部分リポジトリに保持します。これらのキュー・マネージャーは、自分自身に関する情報をパブリッシュするときや、他のキュー・マネージャーに関する情報を要求するときには、使用可能な 2 つの完全リポジトリを使用します。選択された完全リポジトリが使用不可の場合は、もう 1 つの完全リポジトリが使用されます。選択された完全リポジトリが再び使用可能になると、他の完全リポジトリから最新情報や変更情報を収集して、内容を合わせます。すべての完全リポジトリがサービスを休止した場合、それ以外のキュー・マネージャーは部分リポジトリに保存されている情報を使用します。ただし、使用できるのは自らが持っている情報に限られており、新情報や更新要求を処理することはできません。完全リポジトリがネットワークに再接続すると、メッセージが交換され、すべてのリポジトリ (完全リポジトリと部分リポジトリの両方) が最新状態になります。

完全リポジトリの割り振りを計画する際は、以下の考慮事項を含めてください。

- 完全リポジトリを保持するために選択するキュー・マネージャーを、信頼性があり管理されたものにする必要があるということです。永続的に接続された堅固なプラットフォームでホストされるキュー・マネージャーを選択してください。
- 完全リポジトリをホストするシステムの計画停止を検討して、これらのシステムが同時に停止しないようにします。
- ネットワーク・パフォーマンスを考慮します。つまり、地理的に中心に位置するキュー・マネージャーか、クラスター内の他のキュー・マネージャーと同じシステムを共有するキュー・マネージャーを選択してください。
- キュー・マネージャーが複数のクラスターのメンバーであるかどうかを検討します。いくつかのクラスターの完全リポジトリを同一のキュー・マネージャーを使用してホストすると、管理が容易になる場合があります。ただし、この利点をどの程度重視するかは、キュー・マネージャーの予想稼働率がどの程度になるかを踏まえて、平衡を取ることが必要です。
- 一部のシステムは完全リポジトリだけを収容する専用システムにし、他の作業に使用しないようにすることを検討します。これにより、これらのシステムで必要となるのはキュー・マネージャー構成の保守だけになり、他のビジネス・アプリケーションの保守のためにサービスを停止することがなくなります。また、システム・リソースの使用に関して、リポジトリを保守するタスクがアプリケーションと競合することがなくなります。これは特に、クラスターの状態を保持するために完全リポジトリのワークロードが非常に大きくなる大規模クラスター (例えば、1000 を超えるキュー・マネージャーから成るクラスター) で利点があります。

完全リポジトリを 2 つより多くすることは可能ですが、推奨される状況はほとんどありません。オブジェクト定義 (つまり、キュー、トピック、およびチャンネル) は使用可能なすべての完全リポジトリに流れますが、要求は部分リポジトリから最大 2 つの完全リポジトリにしか流れません。つまり、完全リポジトリを 2 つより多く定義しても、いずれか 2 つの完全リポジトリが使用不可になった場合、部分リ

ポジトリーによっては期待している更新を受け取らない可能性があります。 [MQ Clusters: Why only two Full Repositories?](#) を参照してください。

完全リポジトリーを2つより多く定義することが有用と思われる1つの状況は、既存の完全リポジトリーを新しいハードウェアまたは新しいキュー・マネージャーにマイグレーションする場合です。この場合は、置換用の完全リポジトリーを導入し、そこにデータが完全に取り込まれたことを確認してから、以前の完全リポジトリーを除去する必要があります。完全リポジトリーを追加するときは必ず、他のすべての完全リポジトリーに CLUSSDR チャンネルで直接接続する必要があることに注意してください。

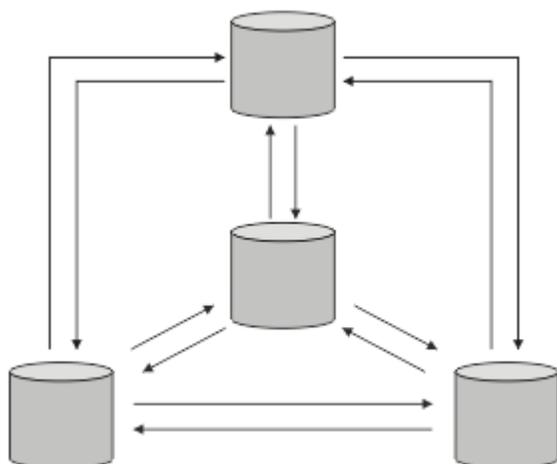


図 27. 2つより多く接続された完全リポジトリー

関連情報

[MQ Clusters: Why only two Full Repositories?](#)

[How big can an MQ Cluster be?](#)

クラスターの編成

どのキュー・マネージャーがどの完全リポジトリーにリンクするかを選択します。パフォーマンスへの影響やキュー・マネージャーのバージョン、および複数の CLUSSDR チャンネルが望ましいかどうかを考慮します。

完全リポジトリーを保持するためのキュー・マネージャーを選択したら、どのキュー・マネージャーをどの完全リポジトリーにリンクするかを決定する必要があります。CLUSSDR チャンネル定義は、クラスター内の他の完全リポジトリーの情報を取り出す完全リポジトリーに、キュー・マネージャーをリンクします。それ以後、このキュー・マネージャーは任意の2つの完全リポジトリーにメッセージを送信します。このとき常に、まず CLUSSDR チャンネル定義がある完全リポジトリーを送信先にしようとします。キュー・マネージャーをいずれかの完全リポジトリーにリンクすることを選択できます。選択する場合には、構成のトポロジーを考慮する必要があります。また、キュー・マネージャーの物理的または地理的な場所も考慮する必要があります。

すべてのクラスター情報が2つの完全リポジトリーに送信されるため、CLUSSDR チャンネル定義をもう1つ作成しなければならない場合があります。多数の完全リポジトリーがあり、広範な領域に広がっている場合は、クラスターに2番目の CLUSSDR チャンネルを定義することができます。これで、情報の送信先となる2つの完全リポジトリーを制御することができます。

クラスターの命名規則

キュー・マネージャーが属するクラスターを識別する命名規則を使用して、同じクラスター内のキュー・マネージャーの命名について考えます。チャンネル名の命名に類似した命名規則を使用し、チャンネルの特性を記述するために命名規則を拡張します。

MQ クラスターの命名時のベスト・プラクティス

クラスター名は最大 48 文字ですが、他のオブジェクトに命名規則を適用する際には比較的短いクラスター名が役立ちます。180 ページの『[クラスター・チャンネル名を選択する際のベスト・プラクティス](#)』を参照してください。

クラスター名を選択する際には、通常、「コンテンツ」ではなく、クラスターの「目的」(存続期間が長い可能性が高い)を表すと役立ちます。例えば、'QM1_QM2_QM3_CLUS'ではなく'B2BPROD'または'ACTTEST'です。

クラスター・キュー・マネージャー名を選択する際のベスト・プラクティス

新規クラスターとそのメンバーを最初から作成する場合は、クラスターの使用法を反映するキュー・マネージャーの命名規則を検討してください。キュー・マネージャーにはそれぞれ異なる名前を付ける必要があります。ただし、クラスター内のキュー・マネージャーに一連の類似した名前を付けることができます。これにより、論理グループ(例えば、「ACTTQM1」、「ACTTQM2」)を識別して記憶することができます。

比較的短いキュー・マネージャー名(例えば 8 文字未満)は、次のセクションで説明する規則、またはチャンネル名に類似した規則を使用することを選択した場合に役立ちます。

クラスター・チャンネル名を選択する際のベスト・プラクティス

キュー・マネージャーおよびクラスターは最大 48 文字の名前を持つことができ、チャンネル名は 20 文字に制限されているため、プロジェクトの途中で命名規則を変更する必要がないように、最初にオブジェクトに命名する際には注意してください(前のセクションを参照してください)。

チャンネルを定義する際には、クラスター内のいずれかのキュー・マネージャーで自動的に作成されたクラスター送信側チャンネルの名前は、クラスター内の受信側キュー・マネージャーで構成された対応するクラスター受信側チャンネルから取られることに注意してください。したがって、これらのチャンネルは固有でなければならず、クラスター内のリモート・キュー・マネージャーで意味を成す。

一般的な方法の 1 つは、クラスター名を前に付けたキュー・マネージャー名を使用することです。例として、クラスター名が CLUSTER1 で、キュー・マネージャーが QM1、QM2 の場合、クラスター受信側チャンネルは CLUSTER1.QM1、CLUSTER1.QM2 となります。

チャンネルの優先順位が異なる場合、または異なるプロトコルを使用する場合は、この規則を拡張することができます。以下に例を示します。

- CLUSTER1.QM1.S1
- CLUSTER1.QM1.N3
- CLUSTER1.QM1.T4

この例では、S1 が最初の SNA チャンネルになり、N3 がネットワーク優先順位が 3 の NetBIOS チャンネルになり、T4 が IPV4 ネットワークを使用する TCP/IP になります。

共用チャンネル定義の命名

単一のチャンネル定義を複数のクラスターで共有することができます。その場合は、ここで推奨されている命名規則を変更する必要があります。ただし、[チャンネル定義の管理](#)で説明されているように、通常はクラスターごとに個別のチャンネルを定義することをお勧めします。

古いチャンネルの命名規則

クラスター環境以外では、従来は「FROMQM.TO.TARGETQM」命名規則を使用することが一般的であったため、既存のクラスターで類似したもの(CLUSTER.TO.TARGET など)が使用されている場合があります。これは、新しいクラスター命名方式の一部としては推奨されません。これにより、チャンネル名内の「有用な」情報を伝達するために使用可能な文字がさらに少なくなるためです。

クラスターのオーバーラップ

クラスターのオーバーラップは、追加の管理機能を提供します。名前リストを使用して、オーバーラップするクラスターの管理に必要なコマンドの数を減らします。

オーバーラップするクラスターを作成できます。オーバーラップするクラスターを定義する理由には、次のようなさまざまなものがあります。

- 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。

181 ページの図 28 では、キュー・マネージャー STF2 は両方のクラスターのメンバーです。1つのキュー・マネージャーが2つ以上のクラスターのメンバーである場合、名前リストの利点を生かして、必要な定義の数を減らすことができます。名前リストには、名前(例えば、クラスター名)のリストを入れます。クラスターを命名する名前リストを作成できます。ALTER QMGR コマンドで STF2 に対してこの名前リストを指定して、両方のクラスターの完全リポジトリ・キュー・マネージャーにすることができます。

ネットワークに2つ以上のクラスターがある場合、それぞれ異なる名前を付ける必要があります。同じ名前の2つのクラスターがマージされると、再び分離することは不可能です。また、クラスターおよびチャンネルに別々の名前を付けるとよいでしょう。これにより、DISPLAY コマンドの出力を確認する際に区別しやすくなります。正しく動作するためにはキュー・マネージャー名がクラスター内で固有でなければなりません。

サービス・クラスの定義

各職員および各学生ごとにキュー・マネージャーを持つ大学を想像してください。職員間のメッセージは、高い優先度と高い帯域幅のチャンネル上でやり取りされます。学生間のメッセージは、安価で低速のチャンネル上でやり取りされます。このネットワークは、従来の分散キューイング技法でセットアップできます。WebSphere MQ は、宛先のキュー名およびキュー・マネージャー名を探すことによって、使用するチャンネルを選択します。

職員と学生をより明確に区別するために、181 ページの図 28 に示されているようにそれらのキュー・マネージャーを2つのクラスターにグループ化することができます。WebSphere MQ は、職員クラスターに定義されているチャンネルを介してのみ、メッセージをそのクラスターの会議キューに移動します。学生クラスターのゴシップ・キューのメッセージは、そのクラスター内に定義されているチャンネルに送られ、適切なサービス・クラスを受けます。

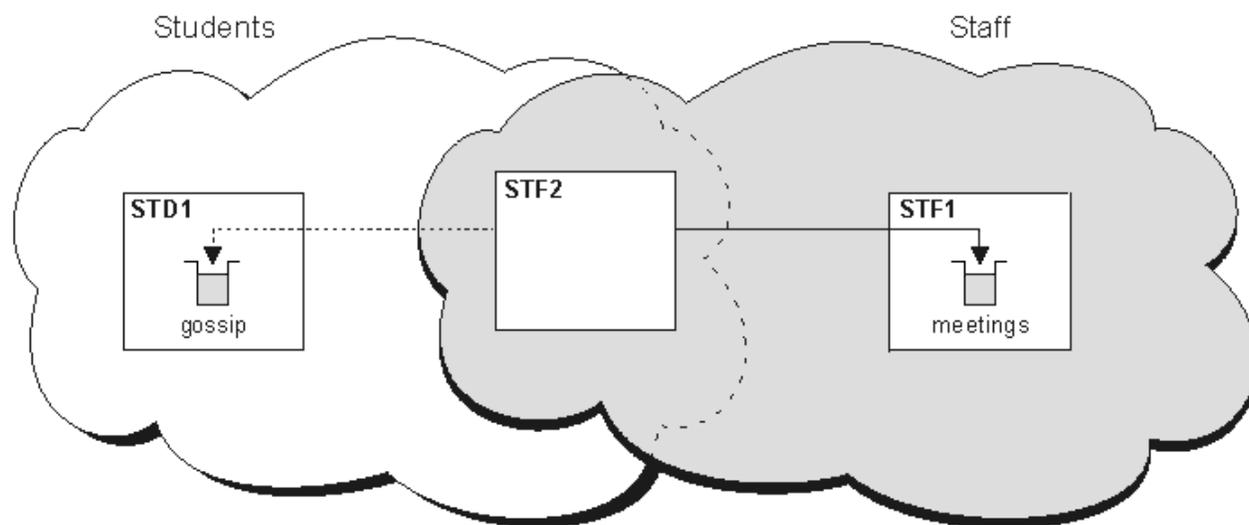


図 28. サービス・クラス

クラスター化のヒント

クラスター化を利用する場合は、その前にシステムやアプリケーションに多少の変更を加える必要が生じることがあります。分散キューイングの動作とは、類似点も相違点もあります。

- バージョン 6.0 より前の WebSphere MQ エクスプローラーでは、WebSphere MQ for z/OS キュー・マネージャーを直接管理することはできません。

- クラスターの外側にあるキュー・マネージャーがクラスター・キューにアクセスできるように、これらのキュー・マネージャーに手動で構成定義を追加しなければなりません。
- 同じ名前の2つのクラスターをマージした場合、それらを再び分離することはできません。したがって、各クラスターには重複しない名前を付けるようにしてください。
- キュー・マネージャーにメッセージが届いたときに、そのキュー・マネージャーにメッセージを受信するキューがないと、メッセージは送達不能キューに書き込まれます。送達不能キューがない場合、そのチャンネルでの受信は失敗しますが、その後処理が再試行されます。送達不能キューの使用法は、分散キューイングの場合と同様です。
- 持続メッセージの健全性が損なわれることはありません。つまり、クラスターの使用によってメッセージが重複したり、消失したりすることはありません。
- クラスターを使用すると、システム管理にかかわる作業が少なくなります。クラスターを使用すると、分散キューイングの場合よりも多数のキュー・マネージャーを大規模なネットワークに簡単に接続できるようになります。クラスター内にあるすべてのキュー・マネージャーの間でデータを通信できるように設定すると、ネットワーク・リソースを消費しすぎることがあるので注意してください。
- WebSphere MQ エクスプローラーは、キュー・マネージャーをツリー構造で表すため、大規模なクラスターの表示には適さない場合があります。
- WebSphere MQ エクスプローラーで、WebSphere MQ for z/OS バージョン 6 以降のリポジトリ・キュー・マネージャーを含むクラスターを管理することができます。個々のシステムに追加リポジトリを指定する必要はありません。以前のバージョンの WebSphere MQ for z/OS では、WebSphere MQ エクスプローラーは、リポジトリ・キュー・マネージャーを含むクラスターを管理できません。WebSphere MQ エクスプローラーが管理できるシステム上で、さらに別のリポジトリを指定する必要があります。
- 配布先リストを使用すると、MQPUT コマンドを1回実行するだけで、同じメッセージを複数の宛先に送信することができます。配布先リストは、WebSphere MQ for AIX、IBM i、HP-UX、Solaris、Linux、および Windows でサポートされています。配布先リストとキュー・マネージャーのクラスターを併用できます。クラスターでは MQPUT の実行時にすべてのメッセージが展開されます。ネットワーク・トラフィックの点においてはクラスターではない環境の場合ほどのメリットはありません。配布先リストを使用すると、多数のチャンネルと伝送キューを手動で定義しなくてすむという利点があります。
- ワークロードのバランスを取るためにクラスターを使用する場合には、使用するアプリケーションを詳しく調べてください。そのアプリケーションに、特定のキュー・マネージャーでメッセージを処理したり、メッセージを特定の順序で処理したりする必要があるかどうかを確認してください。そのようにしてメッセージを処理する必要があるアプリケーションにはメッセージ・アフィニティがあります。そのようなアプリケーションを複雑なクラスターで使用するには、アプリケーションを修正することが必要な場合があります。
- MQOPEN の MQOO_BIND_ON_OPEN オプションによりメッセージを特定の宛先に送信するよう選択することもできます。宛先となるキュー・マネージャーが使用できない状態であると、そのキュー・マネージャーが再び使用可能になるまで、メッセージは送信されません。そのような場合、メッセージを別のキュー・マネージャーに転送するとメッセージが重複してしまうことがあるため、メッセージは転送されません。
- キュー・マネージャーにクラスター・リポジトリをホストさせる場合は、そのホスト名か IP アドレスを知る必要があります。クラスターに加わる他のキュー・マネージャーで CLUSSDR 定義を作成する際に、CONNNAME パラメーターにこの情報を指定する必要があります。DHCP を使用する場合、システムを再始動するたびに DHCP は新しい IP アドレスを割り振ることがあるので、IP アドレスは変わる可能性があります。したがって、この場合 CLUSSDR 定義に IP アドレスを指定することができません。すべての CLUSSDR 定義で、IP アドレスではなくホスト名を指定したとしても、やはりそれらの定義に信頼を置くことはできません。DHCP は、ホストの DNS ディレクトリー項目を新しいアドレスで必ず更新するとは限らないからです。DHCP を使用するシステムでキュー・マネージャーを完全リポジトリとして指名しなければならない場合、DNS ディレクトリーを常に最新の状態に保つことを保証するソフトウェアをインストールしてください。
- チャンネルの接続名として総称名 (例えば、VTAM 総称リソースまたは動的ドメイン名 Server (DDNS) 総称名) を使用しないでください。使用すると、チャンネルは予期しているものとは別のキュー・マネージャーに接続する場合があります。

- メッセージの取得はローカル・クラスター・キューからのみ行えますが、メッセージの書き込みはクラスター内のどのキューに対しても行えます。MQGET コマンドを使用するためにキューをオープンする場合、キュー・マネージャーはローカル・キューをオープンします。
- 単純な構成の WebSphere MQ クラスターをセットアップする場合には、既存のアプリケーションを変更する必要はありません。アプリケーションでは MQOPEN 呼び出しでオープンするキューの名前を指定できます。キュー・マネージャーの格納場所を知っている必要はありません。ワークロード管理用にクラスターをセットアップする場合には、アプリケーションの内容を調べ、必要に応じてアプリケーションを修正する必要があります。
- DISPLAY CHSTATUS および DISPLAY QSTATUS **runmqsc** コマンドを使用して、チャンネルまたはキューの現在のモニター・データおよび状況データを表示できます。モニター情報を使用して、システム・パフォーマンスおよびシステム・ヘルスを測定できます。モニターは、キュー・マネージャー、キュー、およびチャンネルの属性によって制御されます。自動定義されたクラスター送信側チャンネルのモニターは、MONACLS キュー・マネージャー属性によって可能になります。

関連概念

クラスター

クラスターが機能する仕組み

[161 ページの『クラスター化と分散キューイングとの比較』](#)

キュー・マネージャーに接続するために定義する必要があるコンポーネントを、分散キューイングを使用する場合とクラスター化を使用する場合で比較します。

[164 ページの『クラスターのコンポーネント』](#)

クラスターは、キュー・マネージャー、クラスター・リポジトリ、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。

[186 ページの『IBM WebSphere MQ クラスターの管理』](#)

IBM WebSphere MQ クラスターの作成、拡張、および保守を行うことができます。

関連タスク

[159 ページの『キュー・マネージャー・クラスターの構成』](#)

クラスターの動作とクラスター構成の設計について調べるには、および単純なクラスターのセットアップ方法の例については、このトピックにあるリンクを使用してください。

[186 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。

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 WebSphere 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。WebSphere MQ for IBM i、Windows、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。

手順

1. チャンネル・イニシエーターを開始します。

- Windows UNIX Linux

IBM WebSphere MQ for Windows、UNIX and Linux システム

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して開始できます。

2. チャンネル・リスナーを開始します。

- Windows

IBM WebSphere MQ for Windows

WebSphere 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。

WebSphere 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
```

- UNIX Linux

UNIX and Linux システム上の IBM WebSphere MQ

WebSphere 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。

WebSphere MQ 提供のチャンネル・リスナーを起動するときには `runmqslsr` コマンドを使用します。以下に例を示します。

```
runmqslsr -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 WebSphere 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/amqcrista amqcrista  
-m queue.manager.name
```

`MQ_INSTALLATION_PATH` は、WebSphere MQ のインストール先の上位ディレクトリ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 および Linux の場合:

```
kill -1 inetd processid
```

- Solaris 10 以降のバージョンの場合:

```
inetconv
```

キュー・マネージャー・リポジトリに情報が保管される期間

キュー・マネージャー・リポジトリは、情報を 30 日間保管します。自動プロセスは、使用中の情報を効率的にリフレッシュします。

キュー・マネージャーがそれ自体に関する情報を送信したときには、完全および部分リポジトリ・キュー・マネージャーはその情報を 30 日間保管します。例えば、キュー・マネージャーが新しいキューの作成を通知した場合などに、情報は送信されます。この情報の有効期限が切れるのを防ぐために、キュー・マネージャーはそれ自体に関する情報を 27 日後に再送信します。部分リポジトリは、30 日の存続期間の途中で新しい情報要求を送信する場合、有効期限は元の 30 日のままです。

情報は、有効期限が切れた場合、即時にリポジトリから除去されるわけではありません。その情報は、60 日間の猶予期間中、引き続き保持されます。猶予期間中に更新を受け取らなかった場合、その情報は除去されます。猶予期間は、有効期限の日にキュー・マネージャーが一時的にサービスを停止する可能性があることを考慮したものです。キュー・マネージャーがクラスターから切断されている期間が 90 日を超えた場合は、そのキュー・マネージャーはクラスターの一部ではなくなります。ただし、キュー・マネージャーがネットワークに再接続した場合は、再びクラスターの一部になります。完全リポジトリは、他のキュー・マネージャーからの新しい要求に対応するために、期限が満了した情報は使用しません。

また、キュー・マネージャーが完全リポジトリの最新情報に対する要求を送信した場合は、その要求の存続期間は 30 日です。IBM WebSphere MQ は、27 日後に要求をチェックします。27 日の間に参照された場合、その要求は自動的にリフレッシュされます。この期間に参照されなかった場合、その要求は期限満了の対象としてそのまま残され、再び必要になった場合はキュー・マネージャーによりリフレッシュされます。要求が期限満了になると、休止状態のキュー・マネージャーについての情報要求が累積されるのを防ぐことができます。

注: 大規模クラスターでは、多数のキュー・マネージャーが自身に関する全情報を同時に自動的に再送すると悪影響が及ぶ可能性があります。311 ページの『[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。

関連概念

310 ページの『[クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス](#)』

REFRESH CLUSTER コマンドを使用して、クラスターに関するローカルに保持されているすべての情報を破棄し、クラスターの完全リポジトリからその情報を再作成します。例外的な状況を除き、このコマンドを使用する必要はありません。このコマンドを使用する必要がある場合は、使用方法に関する特別な考慮事項があります。この情報は、テストおよびお客様からのフィードバックに基づく指針を示すものです。

IBM WebSphere MQ クラスターの管理

IBM WebSphere MQ クラスターの作成、拡張、および保守を行うことができます。

IBM WebSphere MQ クラスターを管理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

[クラスター](#)

[クラスターが機能する仕組み](#)

[161 ページの『クラスター化と分散キューイングとの比較』](#)

キュー・マネージャーに接続するために定義する必要があるコンポーネントを、分散キューイングを使用する場合とクラスター化を使用する場合で比較します。

[164 ページの『クラスターのコンポーネント』](#)

クラスターは、キュー・マネージャー、クラスター・リポジトリ、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。

関連タスク

[159 ページの『キュー・マネージャー・クラスターの構成』](#)

クラスターの動作とクラスター構成の設計について調べるには、および単純なクラスターのセットアップ方法の例については、このトピックにあるリンクを使用してください。

[186 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。

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 WebSphere 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、**新規** > 「**キュー・マネージャー・クラスター**」をクリックして、ウィザードの指示に従います。
- 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バックグラウンド情報については、[166 ページの『クラスター・キュー』](#)、[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 WebSphere 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。

このクラスターを図に表すと、[187 ページの図 29](#) のようになります。

INVENTORY

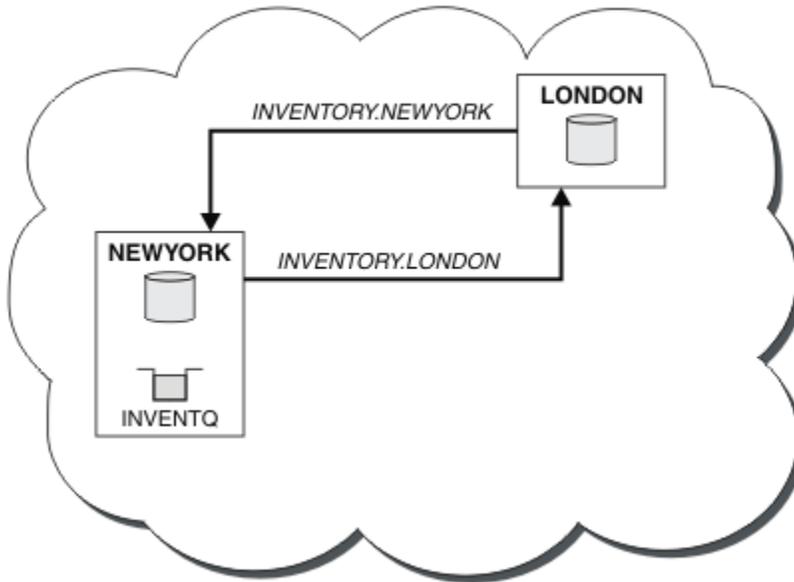


図 29. 2つのキュー・マネージャーで構成される INVENTORY クラスタ

z/OS 上にないクラスタ内の各キュー・マネージャーを、異なるクラスタ伝送キューを使用してクラスタ内の他のキュー・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多少異なります。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。

手順

- [188 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のセットアップ』](#)
- [190 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のセットアップ』](#)
- [193 ページの『z/OS での LU 6.2 を使用するクラスタのセットアップ』](#)
- [195 ページの『クラスタの検査』](#)

タスクの結果

[187 ページの図 29](#) は、このタスクによってセットアップされる INVENTORY クラスタを示しています。

INVENTORY は、明らかに小規模クラスタです。クラスタの概念を具体的に示すのに適しています。このクラスタについて理解する上で重要なのは、このクラスタには今後規模を拡大できる余地があるということです。

関連概念

クラスタ

クラスタが機能する仕組み

[161 ページの『クラスタ化と分散キューイングとの比較』](#)

キュー・マネージャーに接続するために定義する必要があるコンポーネントを、分散キューイングを使用する場合とクラスタ化を使用する場合で比較します。

[164 ページの『クラスタのコンポーネント』](#)

クラスタは、キュー・マネージャー、クラスタ・リポジトリ、クラスタ・チャネル、およびクラスタ・キューで構成されます。

186 ページの『IBM WebSphere MQ クラスターの管理』

IBM WebSphere MQ クラスターの作成、拡張、および保守を行うことができます。

関連タスク

159 ページの『キュー・マネージャー・クラスターの構成』

クラスターの動作とクラスター構成の設計について調べるには、および単純なクラスターのセットアップ方法の例については、このトピックにあるリンクを使用してください。

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

始める前に

- AIX, HP-UX, IBM i, Linux, Solaris, and Windows では、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

AIX, HP-UX, IBM i, Linux, Solaris, and Windows 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。

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 (197 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 WebSphere 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: Websphere 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)
```

注: CONNAME フィールドで IP アドレスを使用し、ポート番号がデフォルトのポート (1414) ではない場合は、リスナーを定義する際にポート番号を定義する必要があります。以下に例を示します。

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(1415)
```

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。CLUSRCVR 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。

クラスター内のキュー・マネージャーごとに、クラスター送信側チャンネルを定義します。キュー・マネージャーは、クラスター送信側チャンネル上の完全リポジトリ・キュー・マネージャーの 1 つにメッセージを送信します。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、もう一方のキュー・マネージャーに定義された CLUSRCVR チャンネルを指す CLUSSDR 定義を必ず設定してください。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を定義すると、クラスター送信側チャンネルが起動します。

```
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。

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

このタスクについて

AIX, HP-UX, IBM i, Linux, Solaris, and Windows 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、[199 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。z/OS では、別々のクラスター伝送キューを使用するキュー・マネージャーをセットアップすることはできません。

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 ([197 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 WebSphere 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: Websphere 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。[197 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 [199 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)
```

注: CONNAME フィールドで IP アドレスを使用し、ポート番号がデフォルトのポート (1414) ではない場合は、リスナーを定義する際にポート番号を定義する必要があります。以下に例を示します。

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(1415)
```

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。CLUSRCVR 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。

クラスター内のキュー・マネージャーごとに、クラスター送信側チャンネルを定義します。キュー・マネージャーは、クラスター送信側チャンネル上の完全リポジトリ・キュー・マネージャーの 1 つにメッセージを送信します。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、もう一方のキュー・マネージャーに定義された CLUSRCVR チャンネルを指す CLUSSDR 定義を必ず設定してください。

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を定義すると、クラスター送信側チャンネルが起動します。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(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)
CONNNAME(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)
CONNNAME(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。

z/OS での LU 6.2 を使用するクラスターのセットアップ

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 (197 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: Websphere 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。

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。CLUSRCVR 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)
CONNAME(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)
CONNAME(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)
CONNAME(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。

クラスター内のキュー・マネージャーごとに、クラスター送信側チャンネルを定義します。キュー・マネージャーは、クラスター送信側チャンネル上の完全リポジトリ・キュー・マネージャーの1つにメッセージを送信します。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

キュ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、もう一方のキュー・マネージャーに定義された CLUSRCVR チャンネルを指す CLUSSDR 定義を必ず設定してください。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を定義すると、クラスター送信側チャンネルが起動します。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(CPIC) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(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)
CONNAME(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。

クラスターの検査

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)

```

amqspmt 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 **amqspmt 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

- 186 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ネルを定義する必要があります。このチャネルにより、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。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')
```

4. オプション: このキュー・マネージャーがクラスターに再結合する場合は、追加の手順をいくつか実行します。

a) 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は、クラスター・メンバーとして表示されるようになったことを確認します。表示されない場合は、以下の追加の手順を実行します。

i) 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、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。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。

ii) 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)。

iii) CLUSRCVR チャンネルを再始動します。

b) クラスターがパブリッシュ/サブスクライブのクラスターであり、再結合するキュー・マネージャーにサブスクリプションがある場合は、以下のコマンドを実行して、プロキシ・サブスクリプションがクラスター全体で正しく同期するようにします。

```
REFRESH QMGR TYPE(PROXYSUB)
```

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。

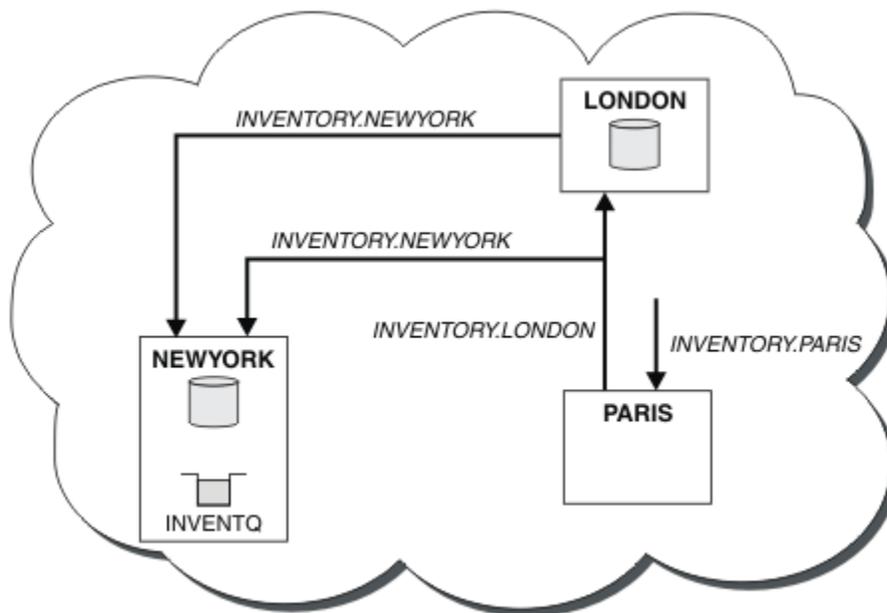


図 30. 3つのキュー・マネージャーで構成される INVENTORY クラスター

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、z/OS 以外のプラットフォームに定義されます。
- 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、[186 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 [197 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。

キューの定義数を少なくするために、デフォルトでは単一の伝送キューを使用します。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` から作成され、`SYSTEM.CLUSTER.TRANSMIT.ChannelName` という名前が付けられます。



重要: 旧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 `SYSTEM.CLUSTER.TRANSMIT.QUEUES` を使用している場合は、`SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` の SHARE/NOSHARE オプションが **SHARE** に設定されていることを確認してください。

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)
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。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。

4. 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')
```

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。

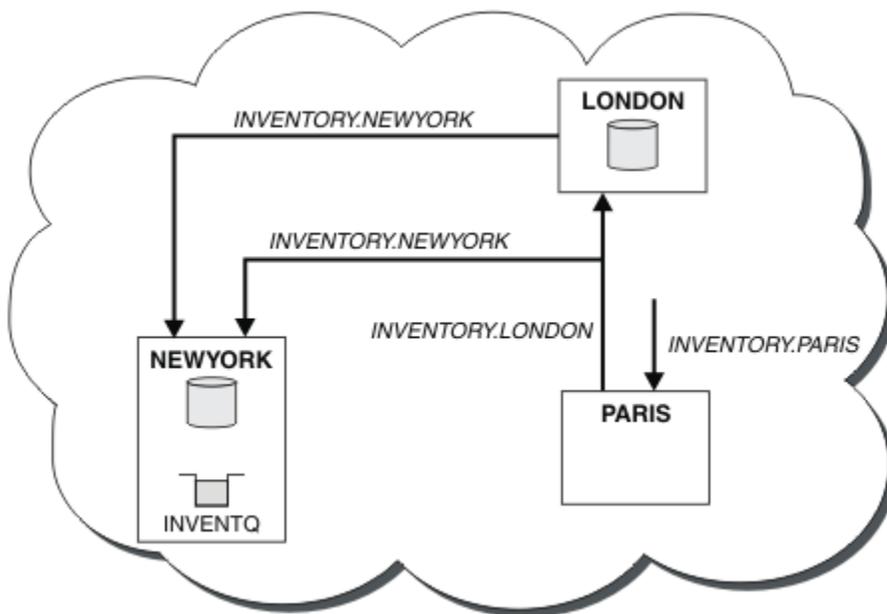


図 31. 3つのキュー・マネージャーで構成される INVENTORY クラスター

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。

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

219 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219 ページの図 37 に示されているオーバーラップするクラスターを、その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。202 ページの図 32 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、204 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。

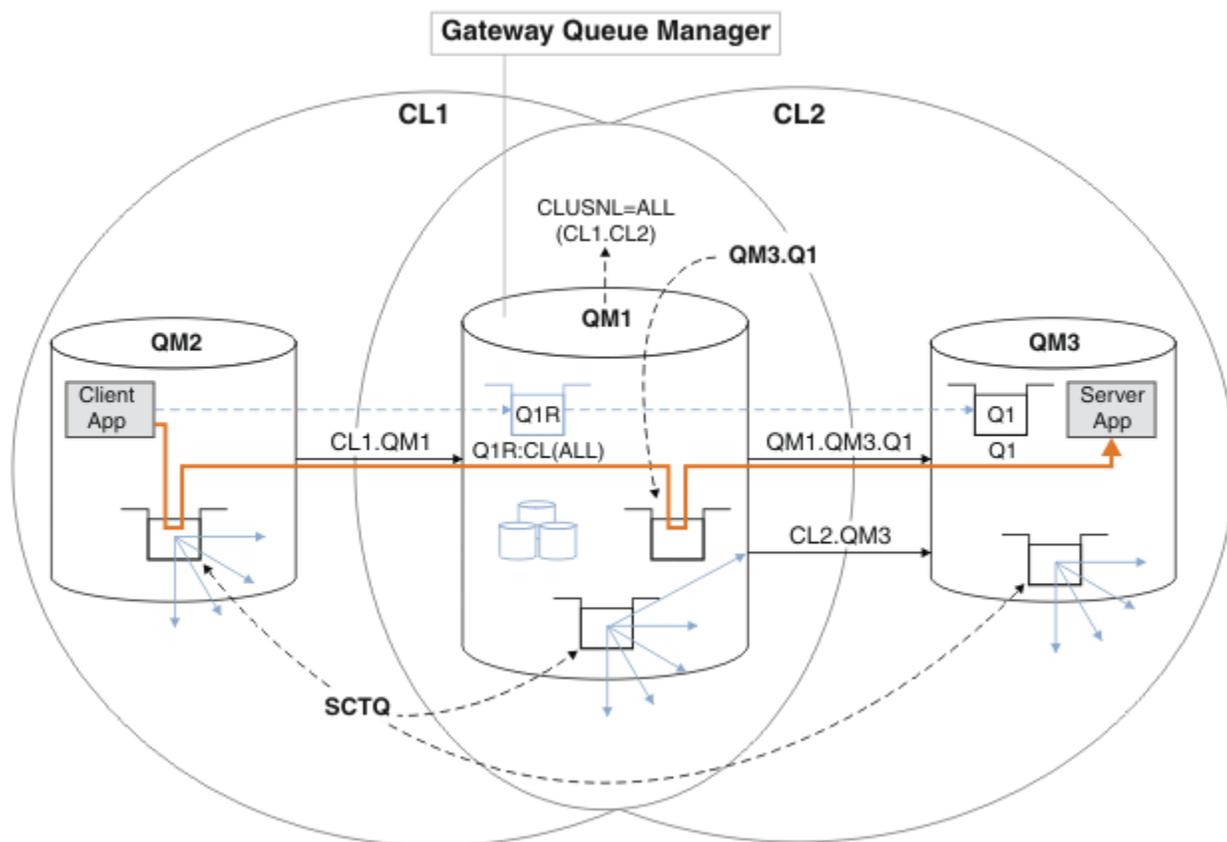


図 32. リモート・キュー定義を使用してハブ・スポーク・クラスター・アーキテクチャーにデPLOYされたクライアント/サーバー・アプリケーション

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
```

関連概念

[160 ページの『アクセス制御と複数のクラスター伝送キュー』](#)

アプリケーションがメッセージをリモート・クラスター・キューに入れるタイミングをチェックするモードを 3 つの中から選択します。これらのモードはそれぞれ、リモートでのクラスター・キューに対するチェック、ローカルでの SYSTEM.CLUSTER.TRANSMIT.QUEUE に対するチェック、クラスター・キューまたはクラスター・キュー・マネージャーのローカル・プロファイルに対するチェックを行います。

[287 ページの『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』](#)

クラスター内のキュー・マネージャー間のメッセージ・フローは、分離することができます。さまざまなクラスター送信側チャンネルによって転送されるメッセージを、それぞれに異なるクラスター伝送キューに配置できます。この手法は、単一のクラスターでも、オーバーラップするクラスターでも使用できます。このトピックでは、使用する手法を選択する際に参考となる例およびベスト・プラクティスを説明します。

関連タスク

[199 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。

[219 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。

[201 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』](#)

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。

224 ページの『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。

204 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。

208 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。

291 ページの『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の選択について手順を追って説明します。1つの共通デフォルト・キュー、個々のデフォルト・キュー、または手動で定義したキューを構成できます。複数のクラスター伝送キューの構成は、z/OS 以外のプラットフォームに適用されます。

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。

始める前に

1. ゲートウェイ・キュー・マネージャーは、バージョン Version 7.5 以降、および z/OS 以外のプラットフォーム上に配置されていなければなりません。
2. 219 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 219 ページの図 37 に示されているオーバーラップするクラスターを、その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 (205 ページの図 33 を参照)。

このソリューションには、201 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・バランシングをサポートします。

- 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 に対するメッセージがまだオン SYSTEM.CLUSTER.TRANSMIT.QUEUE になっても、切り替えが行われます。

205 ページの図 33 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、201 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。CL2 のキュー・マネージャー QM3 が多数の異なるクラスター・キューおよびサーバー・アプリケーションをホストしている場合、それらのすべてのキューは、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。追加のフローは、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・トラフィックを表すグレイの矢印によって 205 ページの図 33 に示されます。

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 (208 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)。

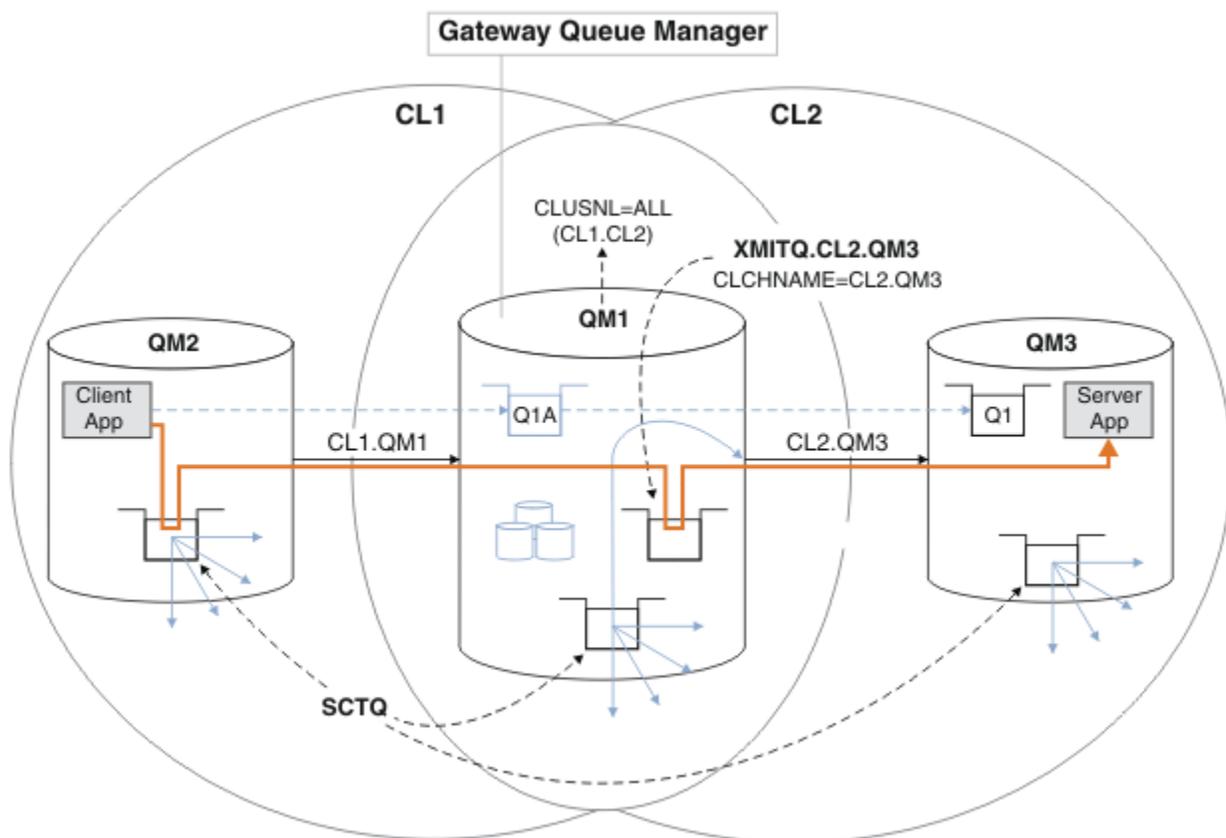


図 33. 追加のクラスター伝送キューを使用してハブ・スポーク・アーキテクチャーにデプロイされたクライアント/サーバー・アプリケーション

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 WebSphere 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て **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)  
SUBSTATE(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 WebSphere 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。

ゲートウェイ・キュー・マネージャーのエラー・ログでメッセージ "AMQ7341 をモニターします。チャンネル CL2.QM3 は XMITQ.CL2.QM3"。

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 AMQSGET0 end
```

関連概念

[160 ページの『アクセス制御と複数のクラスター伝送キュー』](#)

アプリケーションがメッセージをリモート・クラスター・キューに入れるタイミングをチェックするモードを3つの中から選択します。これらのモードはそれぞれ、リモートでのクラスター・キューに対するチェック、ローカルでの SYSTEM.CLUSTER.TRANSMIT.QUEUE に対するチェック、クラスター・キューまたはクラスター・キュー・マネージャーのローカル・プロファイルに対するチェックを行います。

[287 ページの『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』](#)

クラスター内のキュー・マネージャー間のメッセージ・フローは、分離することができます。さまざまなクラスター送信側チャンネルによって転送されるメッセージを、それぞれに異なるクラスター伝送キューに配置できます。この手法は、単一のクラスターでも、オーバーラップするクラスターでも使用できます。このトピックでは、使用する手法を選択する際に参考となる例およびベスト・プラクティスを説明します。

[171 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』](#)

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。

関連タスク

[199 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。

[219 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。

[201 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』](#)

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。

[224 ページの『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。

[204 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。

[208 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。

291 ページの『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の選択について手順を追って説明します。1つの共通デフォルト・キュー、個々のデフォルト・キュー、または手動で定義したキューを構成できます。複数のクラスター伝送キューの構成は、z/OS以外のプラットフォームに適用されます。

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。

始める前に

このタスクで説明する手順は、[205 ページの図 33](#) 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・マネージャーは、バージョン Version 7.5 以降、および z/OS 以外のプラットフォーム上に配置されていなければなりません。
2. [219 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 [219 ページの図 37](#) に示されているオーバーラップするクラスターを、そのタスクで説明している手順に従って作成します。
3. [204 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 [205 ページの図 33](#) 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

[204 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、[204 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、[209 ページの図 34](#) 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ぬことを意味します。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 WebSphere MQ は、204 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。

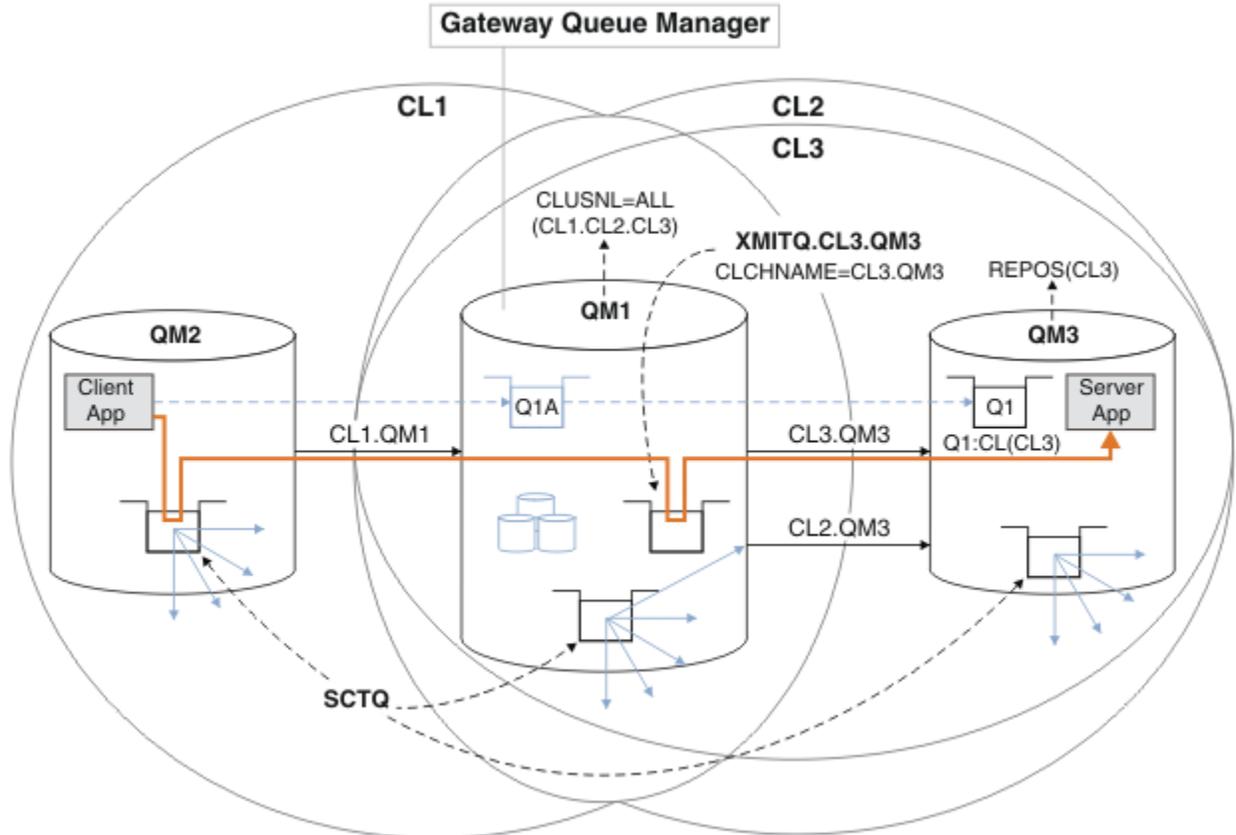


図 34. 追加のクラスターを使用して、同じキュー・マネージャー上の多数のクラスター・キューのうちの1つに向かうメッセージ・トラフィックをゲートウェイ・キュー・マネージャー内で分離する

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 NAMLIST(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)
```

4. 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
```

5. 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 WebSphere 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 WebSphere 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て **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)
CONNAME (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 WebSphere 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)
CONNAME (127.0.0.1(1413)) CURRENT
```

```
RQMNAME(QM3)          STATUS(RUNNING)
SUBSTATE(MQGET)       XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE/CL2.QM3)
```

- f) ゲートウェイ・キュー・マネージャーのエラー・ログでメッセージ "AMQ7341 をモニターします。チャネル CL2.QM3 は SYSTEM.CLUSTER.TRANSMIT.QUEUE / CL2 です。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 **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 AMQSGET0 end
```

関連概念

[160 ページの『アクセス制御と複数のクラスター伝送キュー』](#)

アプリケーションがメッセージをリモート・クラスター・キューに入れるタイミングをチェックするモードを 3 つの中から選択します。これらのモードはそれぞれ、リモートでのクラスター・キューに対するチェック、ローカルでの SYSTEM.CLUSTER.TRANSMIT.QUEUE に対するチェック、クラスター・キューまたはクラスター・キュー・マネージャーのローカル・プロファイルに対するチェックを行います。

[287 ページの『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』](#)

クラスター内のキュー・マネージャー間のメッセージ・フローは、分離することができます。さまざまなクラスター送信側チャネルによって転送されるメッセージを、それぞれに異なるクラスター伝送キューに配置できます。この手法は、単一のクラスターでも、オーバーラップするクラスターでも使用できます。このトピックでは、使用する手法を選択する際に参考となる例およびベスト・プラクティスを説明します。

[171 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』](#)

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。

関連タスク

[199 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。

[219 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。

[201 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』](#)

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。

224 ページの『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。

204 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。

208 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。

291 ページの『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の選択について手順を追って説明します。1つの共通デフォルト・キュー、個々のデフォルト・キュー、または手動で定義したキューを構成できます。複数のクラスター伝送キューの構成は、z/OS以外のプラットフォームに適用されます。

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

- 186 ページの『新規クラスターのセットアップ』で説明されているように、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。システムから接続名を検出するよう IBM WebSphere MQ に要求することができます。そのためには CONNAME を省略するか、または CONNAME(' ') を指定します。IBM WebSphere MQ は、システムの現行 IP アドレスを使用して CONNAME 値を生成します。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・マネージャー宛での 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)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

4. オプション: このキュー・マネージャーがクラスターに再結合する場合は、追加の手順をいくつか実行します。

a) 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は、クラスター・メンバーとして表示されるようになったことを確認します。表示されない場合は、以下の追加の手順を実行します。

i) 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、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。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。

ii) 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)。

iii) CLUSRCVR チャンネルを再始動します。

b) クラスターがパブリッシュ/サブスクライブのクラスターであり、再結合するキュー・マネージャーにサブスクリプションがある場合は、以下のコマンドを実行して、プロキシ・サブスクリプションがクラスター全体で正しく同期するようにします。

```
REFRESH QMGR TYPE(PROXYSUB)
```

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 197 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。

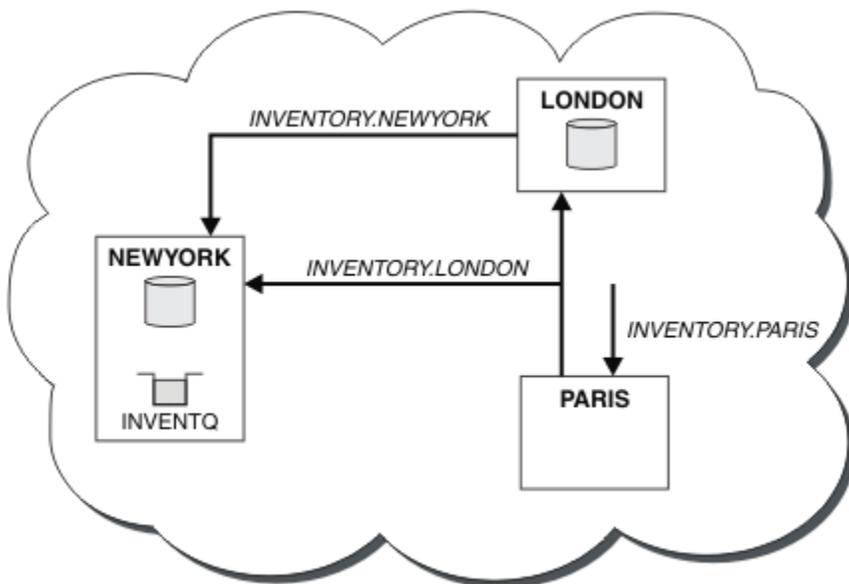


図 35. 3つのキュー・マネージャーで構成される INVENTORY クラスター

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、`STRING+QMNAME+` を含む **CLUSSDR** が開始します。LONDON システムでは、IBM WebSphere MQ は `+QMNAME+` をキュー・マネージャー名 (LONDON) に解決します。次に、IBM WebSphere MQ は、`INVENTORY.LONDON` というチャンネルの定義を、対応する **CLUSRCVR** 定義と突き合わせます。

WebSphere 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 전송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。パリで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 INVENTQ, PARIS にメッセージを書き込もうとすると、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が自動的に定義されます。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。

関連資料

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

- [197 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。処理能力を強化するため、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、[231 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)
CONNAME(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)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from TORONTO to repository at NEWYORK')
```

4. オプション: このキュー・マネージャーがクラスターに再結合する場合は、追加の手順をいくつか実行します。

a) 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は、クラスター・メンバーとして表示されるようになったことを確認します。表示されない場合は、以下の追加の手順を実行します。

i) 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、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。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。

ii) 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)。

iii) CLUSRCVR チャンネルを再始動します。

b) クラスターがパブリッシュ/サブスクライブのクラスターであり、再結合するキュー・マネージャーにサブスクリプションがある場合は、以下のコマンドを実行して、プロキシ・サブスクリプションがクラスター全体で正しく同期するようにします。

```
REFRESH QMGR TYPE(PROXYSUB)
```

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

218 ページの図 36 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。

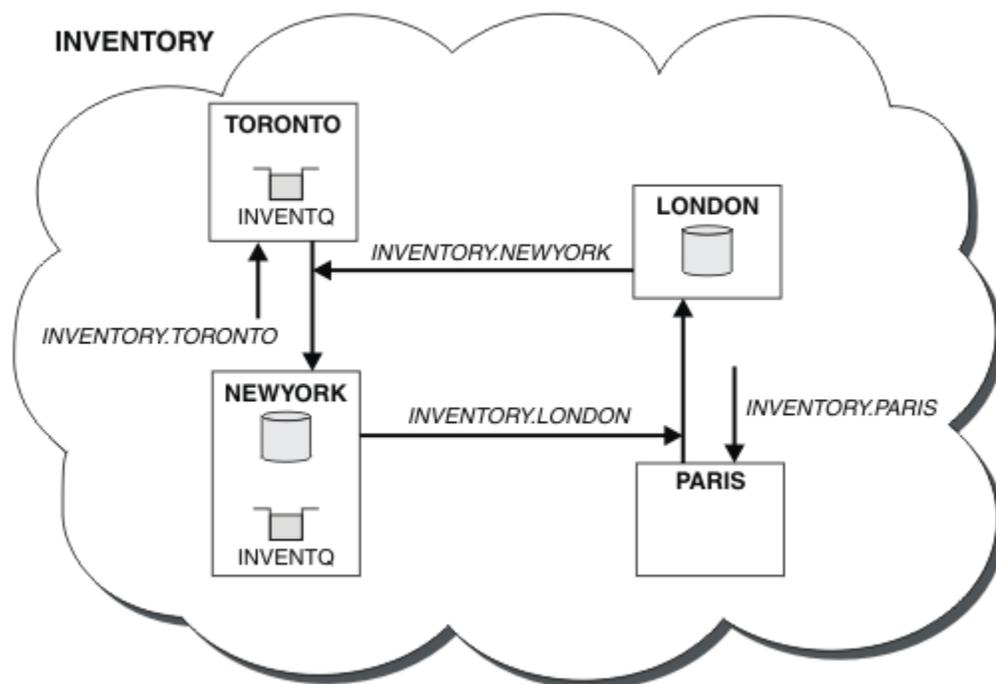


図 36. 4 つのキュー・マネージャーが格納されている INVENTORY クラスター

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ーは、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 か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 WebSphere 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。

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。

このタスクについて

219 ページの図 37 に、クラスター・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ー構成例を示します。この例については、287 ページの『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』で説明しています。

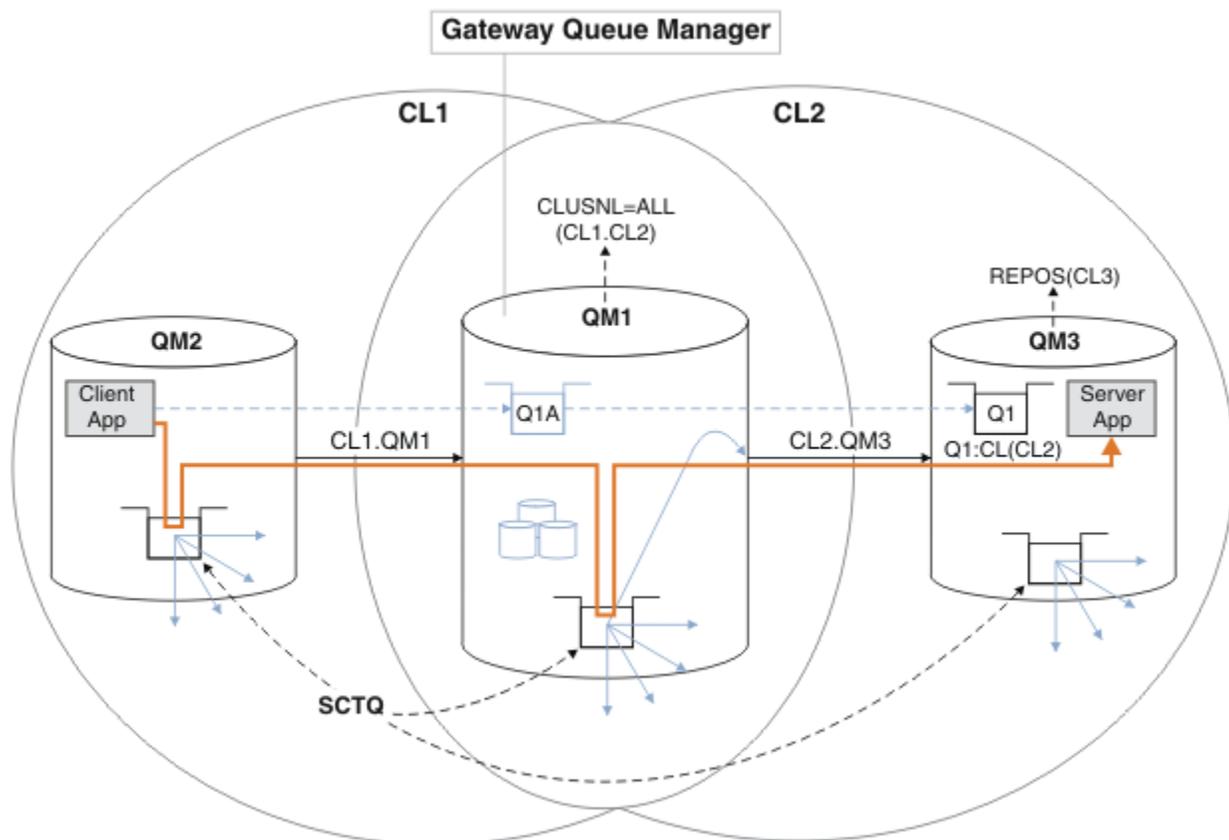


図 37. ハブ・スポーク・アーキテクチャにデPLOYされた、IBM WebSphere MQ クラスターを使用するクライアント/サーバー・アプリケーション

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 つのクラスターの統合を表

しています。より現実的なシナリオについては、291 ページの『[クラスター化: クラスター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. キュー・マネージャー QM1、QM2、QM3、QM4、QM5 を作成して開始します。

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QMn
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 QMn
DEFINE LISTENER(TCP141n) TRPTYPE(TCP) IPADDR(hostname) PORT(141n) CONTROL(QMGR) REPLACE
START LISTENER(TCP141n)
```

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 および QM4 完全リポジトリを CL1 に、QM3 および QM5 完全リポジトリを CL2 に作成します。

- 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、[220 ページの表 26](#) 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。

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 QMm
DEFINE CHANNEL(CLc.QMn) CHLTYPE(CLUSSDR) CONNAME('localhost(141n)') CLUSTER(CLc) REPLACE
DEFINE CHANNEL(CLc.QMm) CHLTYPE(CLUSRCVR) CONNAME('localhost(141m)') CLUSTER(CLc) REPLACE
```

6. ゲートウェイ・キュー・マネージャー QM1 を各クラスターに追加します。

```
*... 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
```

- ローカル・キュー Q1 をクラスター CL2 内のキュー・マネージャー QM3 に追加します。

```
*... On QM3
DEFINE QLOCAL(Q1) CLUSTER(CL2) REPLACE
```

- クラスター化されたキュー・マネージャーの別名 Q1A をゲートウェイ・キュー・マネージャーに追加します。

```
*... On QM1
DEFINE QALIAS(Q1A) CLUSNL(ALL) TARGET(Q1) TARGTYPE(Queue) DEFBIND(NOTFIXED) REPLACE
```

注: QM1 以外のキュー・マネージャーでキュー・マネージャーの別名を使用するアプリケーションは、別名キューを開くときに 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開かれるときに、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。デフォルト値 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 が存在しないため、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 **amqspout** などのアプリケーションが正しく動作します。

- すべてのクラスター・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ゲートウェイ・キュー・マネージャー QM1 に追加します。

```
*... 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

- キュー別名定義 Q1A を使用して QM3 上の QM2 から Q1 にメッセージを送信して、キュー別名定義をテストします。
 - 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
```

- 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
```

- 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。

この図は、応答メッセージが (RQ と呼ばれる) 一時動的キューに戻すためにたどるパスを示しています。QM3 に接続されているサーバー・アプリケーションは、キュー・マネージャー名 QM2 を使用して応答キューを開きます。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に経路指定します。QM1は、メッセージをQM2に経路指定します。

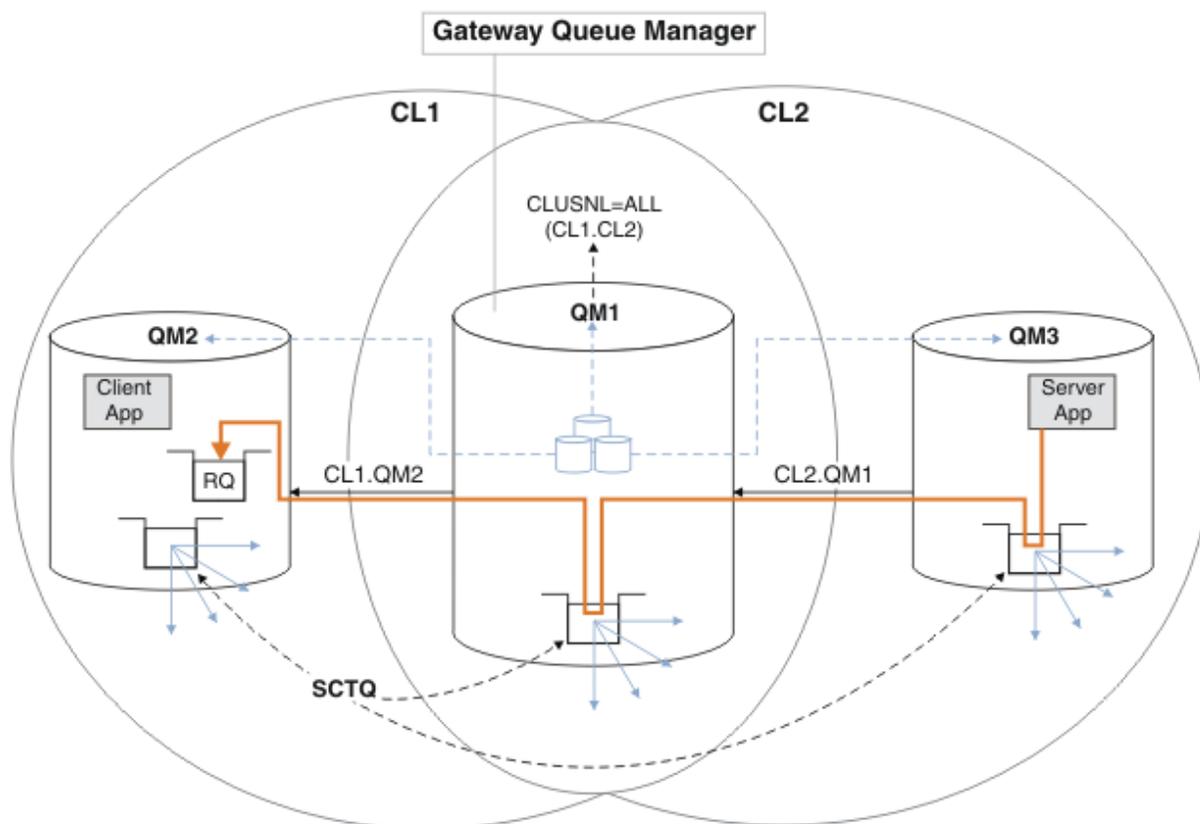


図 38. 応答メッセージを異なるクラスターに戻すためのキュー・マネージャー別名の使用

このルーティングは、次のようにして行われます。QM1では、各クラスター内のすべてのキュー・マネージャーにキュー・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。この場合、QM2別名はクラスターCL1とCL2の両方でクラスター化され、CL1内の実際のキュー・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、応答先キュー名RQ、および応答先キュー・マネージャー名QM2を使用して応答メッセージを作成します。キュー・マネージャーの別名定義QM2がクラスターCL2内のQM1で定義されており、キュー・マネージャーQM2がクラスターCL2内にないため、メッセージはQM1に経路指定されます。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、メッセージをQM1上のクラスター伝送キューに入れ、QM2に転送します。QM1はメッセージをQM2に経路指定します。これは、QM1のQM2でのキュー・マネージャー別名定義がQM2を実際のターゲット・キュー・マネージャーとして定義しているためです。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の両方が同じクラスターCL1にあるため、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の接続情報を検出し、メッセージをQM2に経路指定します。メッセージがQM1によって転送されるようにするには、サーバー・アプリケーションが、オプションDEFBINDをMQBND_BIND_NOT_FIXEDに設定した状態で応答キューを開いている必要があります。サーバー・アプリケーションがオプションMQBND_BIND_ON_OPENで応答キューを開いている場合、メッセージは転送されず、送達不能キューに入ります。

- a. QM3 でトリガーを使用して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・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. キュー別名定義 QRA を使用して QM3 上の QM2 から QR に要求を送信して、キュー・マネージャー別名定義をテストします。

- 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, 2024. ALL RIGHTS RESERVED.
01/02/2012 16:17:15: WebSphere 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
```

関連概念

[160 ページの『アクセス制御と複数のクラスター伝送キュー』](#)

アプリケーションがメッセージをリモート・クラスター・キューに入れるタイミングをチェックするモードを3つの中から選択します。これらのモードはそれぞれ、リモートでのクラスター・キューに対するチェック、ローカルでの SYSTEM.CLUSTER.TRANSMIT.QUEUE に対するチェック、クラスター・キューまたはクラスター・キュー・マネージャーのローカル・プロファイルに対するチェックを行います。

[287 ページの『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』](#)

クラスター内のキュー・マネージャー間のメッセージ・フローは、分離することができます。さまざまなクラスター送信側チャンネルによって転送されるメッセージを、それぞれに異なるクラスター伝送キューに配置できます。この手法は、単一のクラスターでも、オーバーラップするクラスターでも使用できます。このトピックでは、使用する手法を選択する際に参考となる例およびベスト・プラクティスを説明します。

関連タスク

199 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。

219 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。

201 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』

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。

224 ページの『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。

204 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。

208 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。

291 ページの『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の選択について手順を追って説明します。1 つの共通デフォルト・キュー、個々のデフォルト・キュー、または手動で定義したキューを構成できます。複数のクラスター伝送キューの構成は、z/OS 以外のプラットフォームに適用されます。

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。

始める前に

1. ゲートウェイ・キュー・マネージャーは、バージョン Version 7.5 以降、および z/OS 以外のプラットフォーム上に配置されていなければなりません。
2. 219 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 219 ページの図 37 に示されているオーバーラップするクラスターを、その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を実装するには、ゲートウェイ・キュー・マネージャーが Version 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。225 ページの図 39 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はない限り、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、199 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。

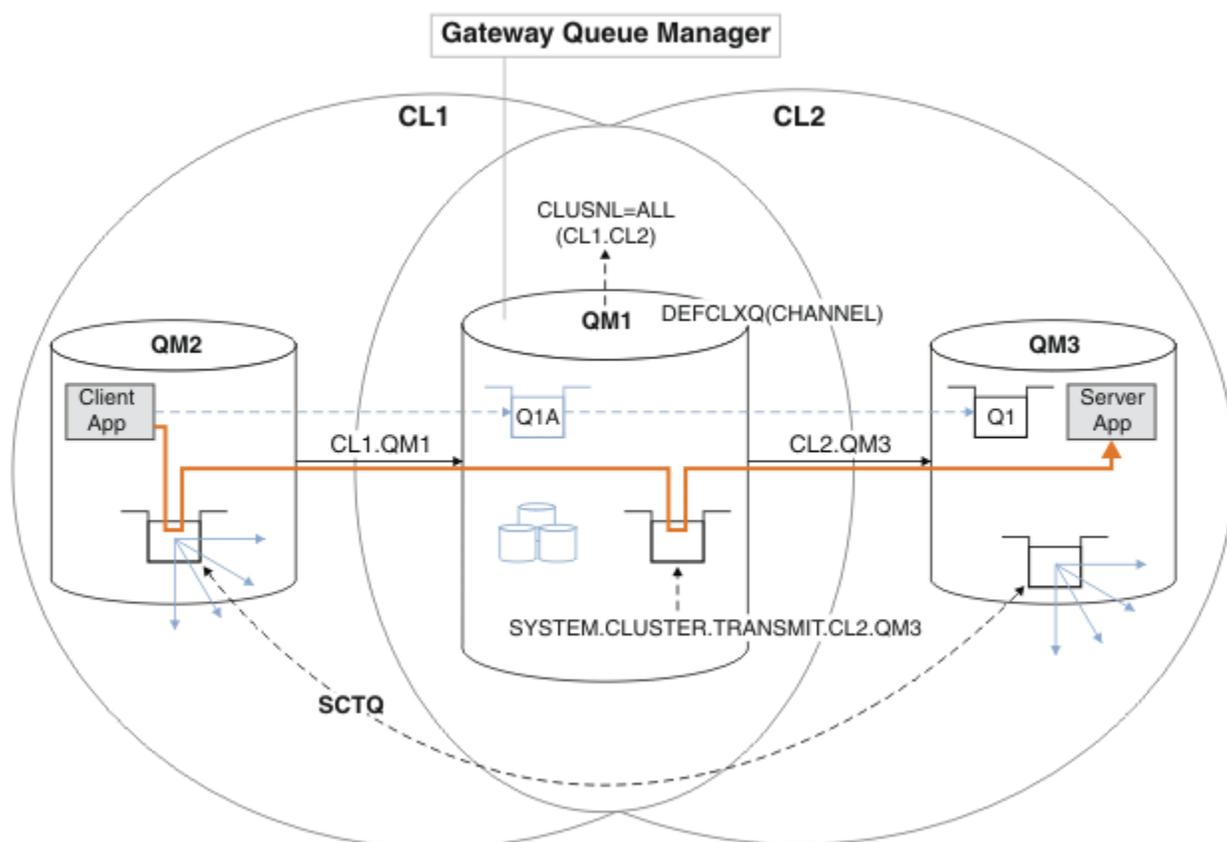


図 39. ゲートウェイ・キュー・マネージャーで個々のクラスター伝送キューを使用するハブ・スポーク・アーキテクチャにデプロイされたクライアント/サーバー・アプリケーション

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)
CONNAME (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)
CONNAME (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)
CONNAME (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)
```

ここで、*ChannelName* は、それぞれの CL1.QM2, CL1.QM4, CL1.QM3, CL1.QM5 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 WebSphere 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)
CONNAME (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)
CONNAME (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 WebSphere MQ channel accepted.
```

e) 切り替えられる伝送キューをモニターします。

ゲートウェイ・キュー・マネージャーのエラー・ログでメッセージ "AMQ7341 をモニターします。チャンネル CL2.QM3 は SYSTEM.CLUSTER.TRANSMIT.QUEUE / CL2 です。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: Channel Status not found.
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)
CONNAME (127.0.0.1(1415))
RQMNAME (QM5)
SUBSTATE (MQGET)
XMITQ (SYSTEM.CLUSTER.TRANSMIT.CL2.QM5)
AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM4)
CONNAME (127.0.0.1(1414))
RQMNAME (QM4)
SUBSTATE (MQGET)
XMITQ (SYSTEM.CLUSTER.TRANSMIT.CL1.QM4)
CHLTYPE (CLUSSDR)
CURRENT
STATUS (RUNNING)
CHLTYPE (CLUSSDR)
CURRENT
STATUS (RUNNING)

```

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。

関連概念

[160 ページの『アクセス制御と複数のクラスター伝送キュー』](#)

アプリケーションがメッセージをリモート・クラスター・キューに入れるタイミングをチェックするモードを 3 つの中から選択します。これらのモードはそれぞれ、リモートでのクラスター・キューに対するチェック、ローカルでの SYSTEM.CLUSTER.TRANSMIT.QUEUE に対するチェック、クラスター・キューまたはクラスター・キュー・マネージャーのローカル・プロファイルに対するチェックを行います。

[287 ページの『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』](#)

クラスター内のキュー・マネージャー間のメッセージ・フローは、分離することができます。さまざまなクラスター送信側チャンネルによって転送されるメッセージを、それぞれに異なるクラスター伝送キューに配置できます。この手法は、単一のクラスターでも、オーバーラップするクラスターでも使用できます。このトピックでは、使用する手法を選択する際に参考となる例およびベスト・プラクティスを説明します。

関連タスク

[199 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。

[219 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。

201 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』

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。

224 ページの『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。

204 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。

208 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。

291 ページの『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の選択について手順を追って説明します。1つの共通デフォルト・キュー、個々のデフォルト・キュー、または手動で定義したキューを構成できます。複数のクラスター伝送キューの構成は、z/OS以外のプラットフォームに適用されます。

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。すべての在庫メッセージをニューヨークに送信し、空の場合はトロントの INVENTQ キューを削除します。

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

- 216 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 は両方とも完全リポジトリを保持します。PARIS および TORONTO は、部分リポジトリを保持します。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。TORONTO 上の INVENTQ を変更して、クラスターの残りの部分からアクセスできないようにします。

```
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・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)
```

MQOO_BIND_ON_OPEN を使用してこのキューに転送中のメッセージは、送達不能キューに送られます。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

チャンネル INVENTORY.TORONTO に未確定メッセージがないことを確認するには、他の各キュー・マネージャー上の INVENTORY.TORONTO というクラスター送信側チャンネルをモニターします。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。

ステップ 1 でキューをサービス休止にすると、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

ユー・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、TORONTO キュー・マネージャーは、INVENTQ だけでなく、PAYROLLQ、SALESQ、および PURCHASESQ もホストすることができます。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)
```

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

- [197 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ネルが必要です。このチャネルでは、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)  
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)  
DESCR('Cluster-sender channel from PARIS to repository at NEWYORK')
```

3. 宛先が PARIS の CLUSSDR チャネルを 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ネルを NEWYORK に追加する必要があります。

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNNAME(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 CLUSQMGR(*) 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)
```

- クラスター内のクラスター送信側チャンネルのうち、宛先が LONDON である他のすべてのチャンネルを、宛先が NEWYORK または PARIS のチャンネルに置き換えます。この小規模な例では、他のチャンネルはありません。忘れたものが他にあるかどうかを確認するには、TYPE (CLUSSDR) を指定して、各キュー・マネージャーから 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)
```

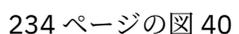
完全リポジトリを LONDON から PARIS に移動した後、できるだけ早くこの作業を実行することが重要です。この作業を実行するまでの間に、INVENTORY.LONDON という CLUSSDR チャンネルを手動で定義したキュー・マネージャーが、このチャンネルを使用する情報の要求を送信する可能性があります。

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: Unexpected publication of a cluster queue object received
- AMQ9432: Query received by a non-repository queue manager

キュー・マネージャー LONDON は完全リポジトリではないので、情報の要求に応答しません。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

234 ページの  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。

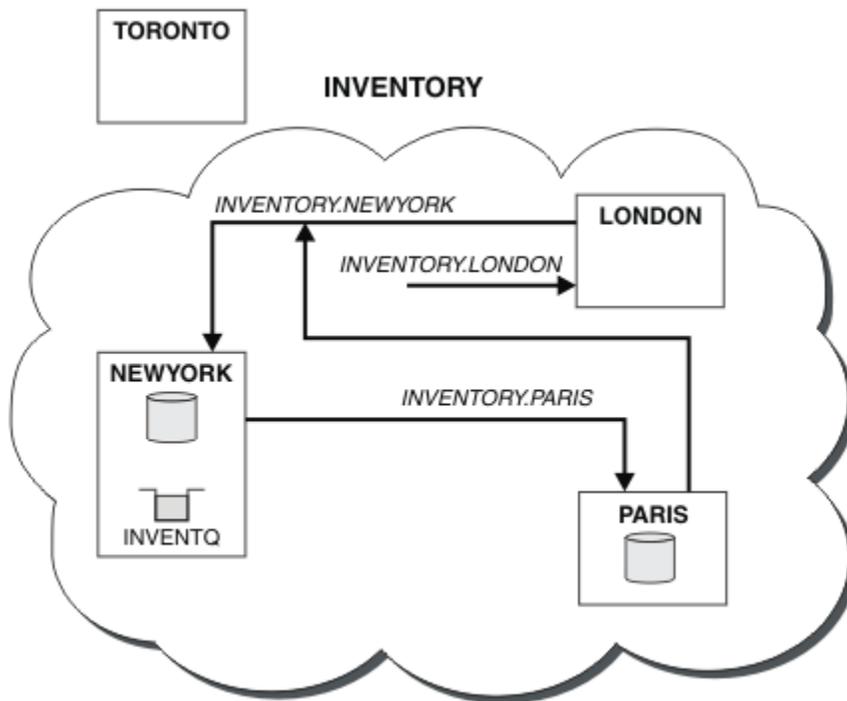


図 40. 完全リポジトリが PARIS に移動した INVENTORY クラスタ

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

231 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 186 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 WebSphere 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 [235 ページの図 41](#) に示します。

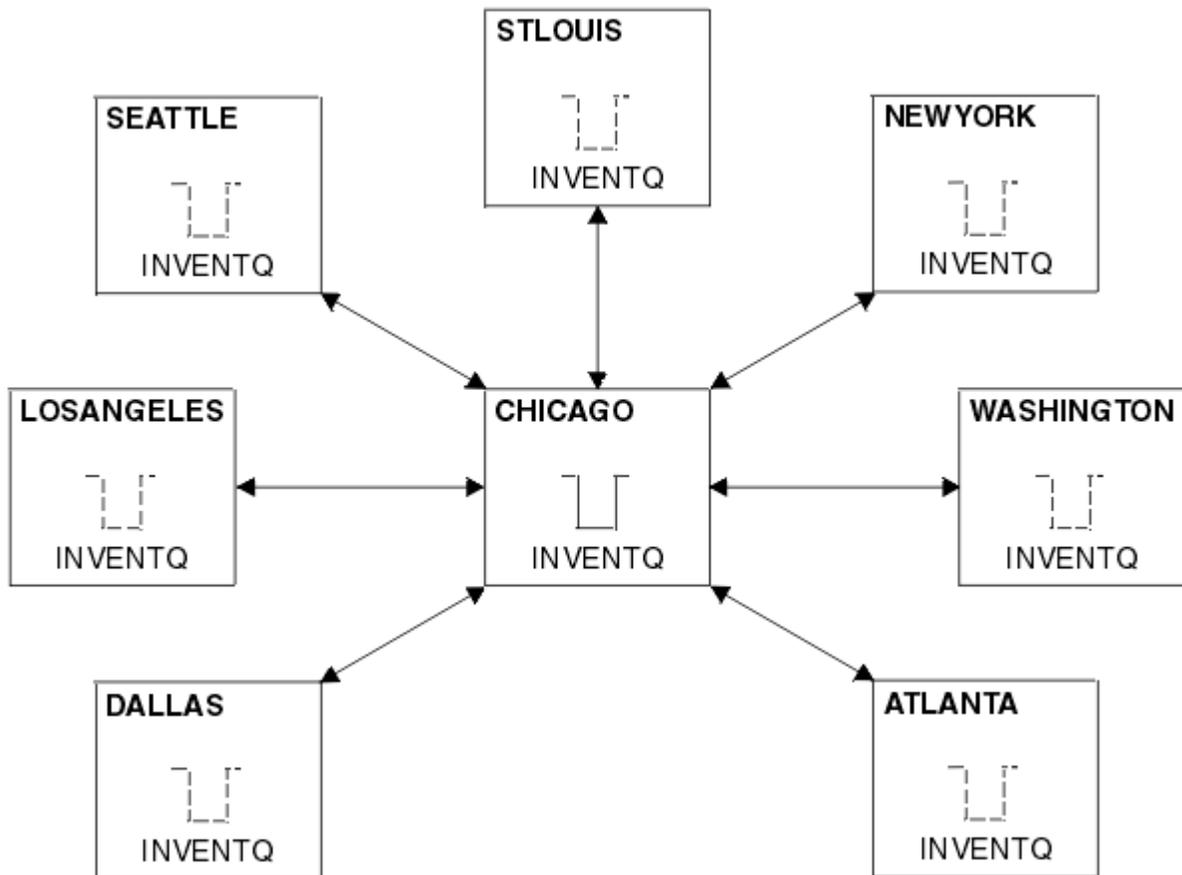


図 41. ハブ・スポーク・ネットワーク

- 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` 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、[279 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。

```
DIS QCLUSTER(INVENTQ)
```

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。
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

- 234 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、WebSphere 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。このクラスターは、CHNSTORE クラスター内の 4 つのキュー・マネージャー (CHICAGO、CHICAGO2、SEATTLE、および ATLANTA) と、2 つの追加キュー・マネージャー (HARTFORD および OMAHA) で構成されます。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。これは、MORDERQ キューにメッセージを書き込むクラスター内の他のキュー・マネージャーによって駆動されません。
- MAILORDER クラスターの完全リポジトリーは、2 つのキュー・マネージャー CHICAGO および CHICAGO2 で保守されます。
- 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

これで、CHICAGO および CHICAGO2 のフル・リポジトリー・キュー・マネージャーは、クラスター CHNSTORE および MAILORDER の両方のフル・リポジトリーを保持することになります。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 NAMLIST(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、これらの定義は、キュー・マネージャーがクラスター CHNSTORE の完全リポジトリーを保持していることを示しています。その定義を変更して、キュー・マネージャーが CHAINMAIL 名前リストにリストされているすべてのクラスターの完全リポジトリーを保持するようにします。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 上の CLUSRCVR チャンネルを変更します。

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 の CLUSSDR チャンネルを変更します。

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します。

SEATTLE と ATLANTA は複数のクラスターのメンバーになるため、クラスターの名前を含む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 上の CLUSRCVR チャンネルを変更します。

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 の CLUSSDR チャンネルを変更します。

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)
CONNNAME(HARTFORD.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for HARTFORD')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSDR) TRPTYPE(TCP)
CONNNAME(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)
CONNNAME(OMAHA.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for OMAHA')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSDR) TRPTYPE(TCP)
CONNNAME(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、[240 ページの図 42](#) 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。両方のクラスターの完全リポジトリは、CHICAGO および CHICAGO2 に保持されます。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。

クラスター CHNSTORE 内の NEWYORK およびクラスター MAILORDER 内の OMAHA に、ACCOUNTQ というキューがあるとします。クラスターをオーバーラップし、SEATTLE 上のアプリケーションがメッセージをキュー ACCOUNTQ に書き込むと、メッセージは ACCOUNTQ のいずれかのインスタンスに送信されます。

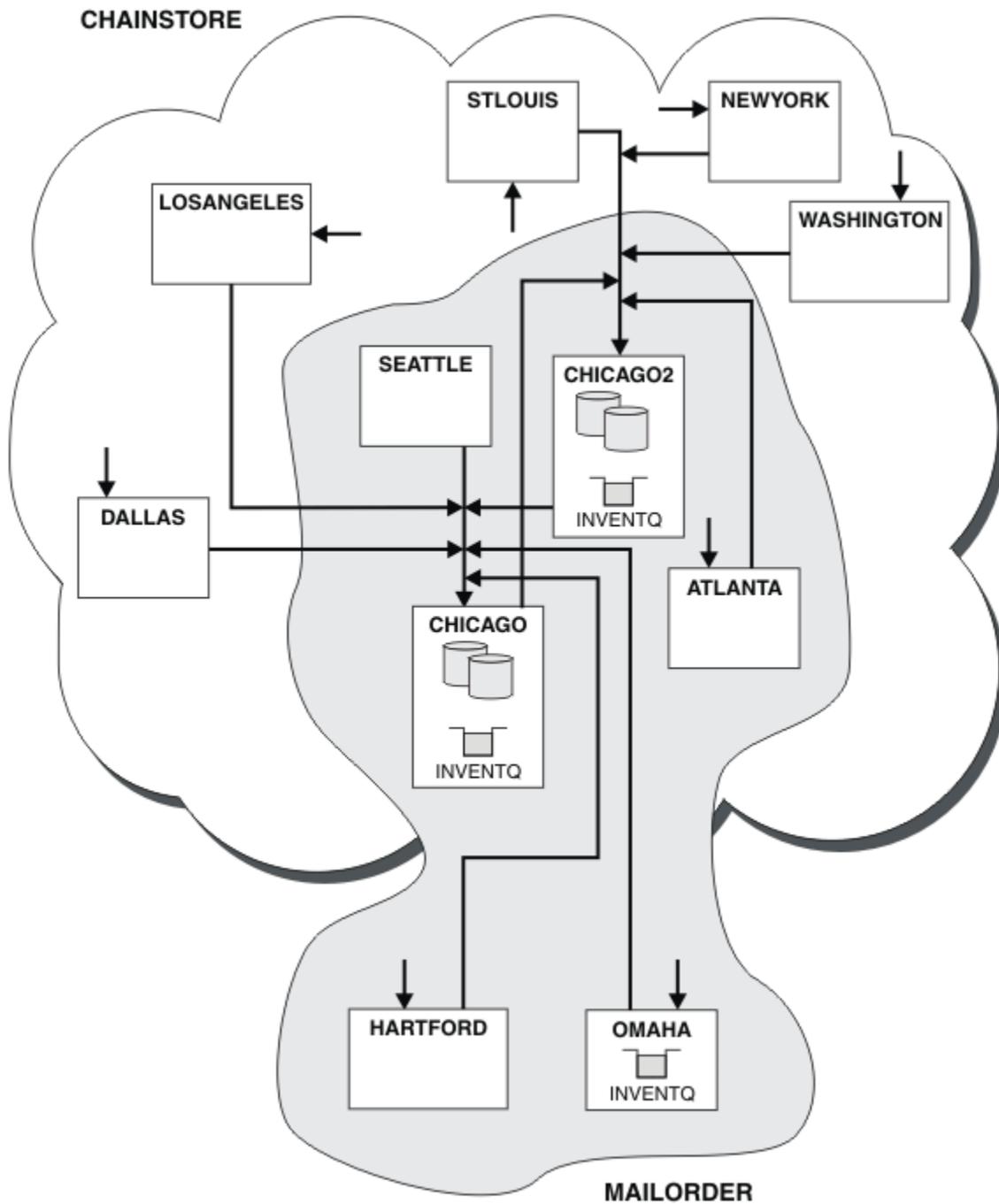


図 42. 相互接続されたクラスター

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。

MAILORDER クラスターを CHNSTORE クラスターとマージして、CHICAGO と CHICAGO2 が完全リポジトリを保持するようにするには、以下のようにします。

- CHICAGO および CHICAGO2 のキュー・マネージャー定義を変更し、名前リスト (CHAINMAIL) を指定する REPOSNL 属性を削除し、クラスター名 (CHNSTORE) を指定する REPOS 属性に置き換えます。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)
```

- クラスター名前リスト CHAINMAIL を指定するすべての定義 (つまり、CHICAGO、CHICAGO2、SEATTLE、および ATLANTA にある CLUSRCVR および CLUSSDR チャンネル定義) を変更して、代わりにクラスター CHNSTORE 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、名前リスト CHAINMAIL の値を変更することができます。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。

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

- 234 ページの『[既存のネットワークのクラスターへの変換](#)』で説明されているように、IBM WebSphere MQ クラスターがセットアップされました。
- 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、完全リポジトリー内の情報が更新され、クラスター全体に伝搬されます。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 DEFBIND(NOTFIXED) を使用してキューが定義されている MQOO_BIND_AS_Q_DEF を使用するアプリケーションは、次に試行される MQPUT 呼び出しまたは MQPUT1 呼び出しで失敗します。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。 次に、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) です。

クラスター CHNSTORE の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、[246 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. クラスターを整理する

必要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。

クラスターからのキュー・マネージャーの除去

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ー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ーが使用可能で、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ーと接続できない場合、[244 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。

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する必要があります。

- キュー・マネージャーが完全リポジトリーをホストする場合、[231 ページの『完全リポジトリーの別のキュー・マネージャーへの移動』](#)の1から4の手順を実行します。
- 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、[229 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、それらのクラスター・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。

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、[197 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、[229 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ー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・リポジトリー以外のクラスター・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 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(' ')
```

2. 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。

3. 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.LONDON) 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、[197 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、[246 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。

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(243 ページの『クラスターからのキュー・マネージャーの除去』)を使用してください。

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する必要があります。

- 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、231 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 4 の手順を実行します。
- 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、229 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、それらのクラスター・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。

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、197 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、229 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. 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)
```

2. 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。

3. 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)
```

4. 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)
```

除去されたキュー・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は、除去されたキュー・マネージャーの **QMID** を指定します。

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)
```

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、[197 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、[246 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。

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。

重要: キュー・マネージャーをクラスターから除去し、REPOS(YES) を使用してリフレッシュした場合、単純に CLUSRCVR の CLUSTER 属性を変更するだけでは、そのキュー・マネージャーを追加して戻すことはできません。CLUSRCVR の CLUSTER 属性を非ブランク（つまり、クラスター名）に変更した後で、REPOS(NO) を指定してクラスターのリフレッシュを実行する必要があります。この時点で、CLUSRCVR の内部シーケンス番号が最新の状態に更新されます。その後、キュー・マネージャーを、完全にポジトリーおよび残りのクラスター・メンバーに正常に再追加できるようになります。(CLUSRCVR チャンネルに正しいクラスター名が与えられた後で、コマンドの REPOS(NO) バージョンを実行する必要があることに注意してください。)

この制約事項は、IBM WebSphere MQ Version 7.5 にのみ適用されます。

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。243 ページの『[クラスターからのキュー・マネージャーの除去](#)』で説明されているように、LONDON キュー・マネージャーが INVENTORY クラスターから削除されました。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)
```

関連概念

[クラスター](#)

[161 ページの『クラスター化と分散キューイングとの比較』](#)

キュー・マネージャーに接続するために定義する必要があるコンポーネントを、分散キューイングを使用する場合とクラスター化を使用する場合で比較します。

[164 ページの『クラスターのコンポーネント』](#)

クラスターは、キュー・マネージャー、クラスター・リポジトリー、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。

[186 ページの『IBM WebSphere MQ クラスターの管理』](#)

IBM WebSphere MQ クラスターの作成、拡張、および保守を行うことができます。

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. SUSPEND QMGR **runmqsc** コマンドを発行して、キュー・マネージャーを中断します。

```
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 クラスター内のキュー・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ー・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。

WebSphere 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・バランスング・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、クラスター内の他のキュー・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. 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にポジティブに通知します。完全にポジティブ・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。

クラスター伝送キューの保守

クラスター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、クラスターのパフォーマンスには不可欠です。

始める前に

- クラスター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。
- クラスター伝送キューを保管するメディアが満杯にならないようにしてください。

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、大規模なクラスターの場合、**REFRESH CLUSTER** コマンドを使用すると、処理中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。[310 ページの『クラスター化: REFRESH CLUSTER の使用に関するベスト・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 コマンドにより、キュー・マネージャーはその完全リポジトリ内容に関してコールド・スタートが可能になります。IBM WebSphere MQ は、キューのデータの消失が起こらないことを保証しています。

関連概念

[310 ページの『クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス』](#)

REFRESH CLUSTER コマンドを使用して、クラスターに関するローカルに保持されているすべての情報を破棄し、クラスターの完全リポジトリからその情報を再作成します。例外的な状況を除き、このコマンドを使用する必要はありません。このコマンドを使用する必要がある場合は、使用方法に関する特別な考慮事項があります。この情報は、テストおよびお客様からのフィードバックに基づく指針を示すものです。

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・マネージャーを回復するには、キュー・マネージャーを復元してから、`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、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。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。

関連概念

310 ページの『[クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス](#)』

`REFRESH CLUSTER` コマンドを使用して、クラスターに関するローカルに保持されているすべての情報を破棄し、クラスターの完全リポジトリからその情報を再作成します。例外的な状況を除き、このコマンドを使用する必要はありません。このコマンドを使用する必要がある場合は、使用方法に関する特別な考慮事項があります。この情報は、テストおよびお客様からのフィードバックに基づく指針を示すものです。

可用性のためのクラスター・チャネルの構成

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ネルの技術が使用されています。クラスター・チャ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。

- チャネルの動作方法
- チャネル状況の検索方法
- チャ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ネルおよびクラスター受信側チャネルの `HBINT` または `KAINT` 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 以外の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、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。

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

クラスター

クラスターが機能する仕組み

161 ページの『クラスター化と分散キューイングとの比較』

キュー・マネージャーに接続するために定義する必要があるコンポーネントを、分散キューイングを使用する場合とクラスター化を使用する場合で比較します。

164 ページの『クラスターのコンポーネント』

クラスターは、キュー・マネージャー、クラスター・リポジトリ、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。

186 ページの『IBM WebSphere MQ クラスターの管理』

IBM WebSphere MQ クラスターの作成、拡張、および保守を行うことができます。

262 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。

265 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。

265 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

159 ページの『キュー・マネージャー・クラスターの構成』

クラスターの動作とクラスター構成の設計については、および単純なクラスターのセットアップ方法の例については、このトピックにあるリンクを使用してください。

186 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。

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

252 ページの図 43 は、DEMO というクラスターの外部にある QM3 というキュー・マネージャーを示しています。QM3 は、クラスターをサポートしない WebSphere MQ 製品のキュー・マネージャーの場合があります。QM3 は、以下のように定義された Q3 というキューをホストします。

```
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)
```

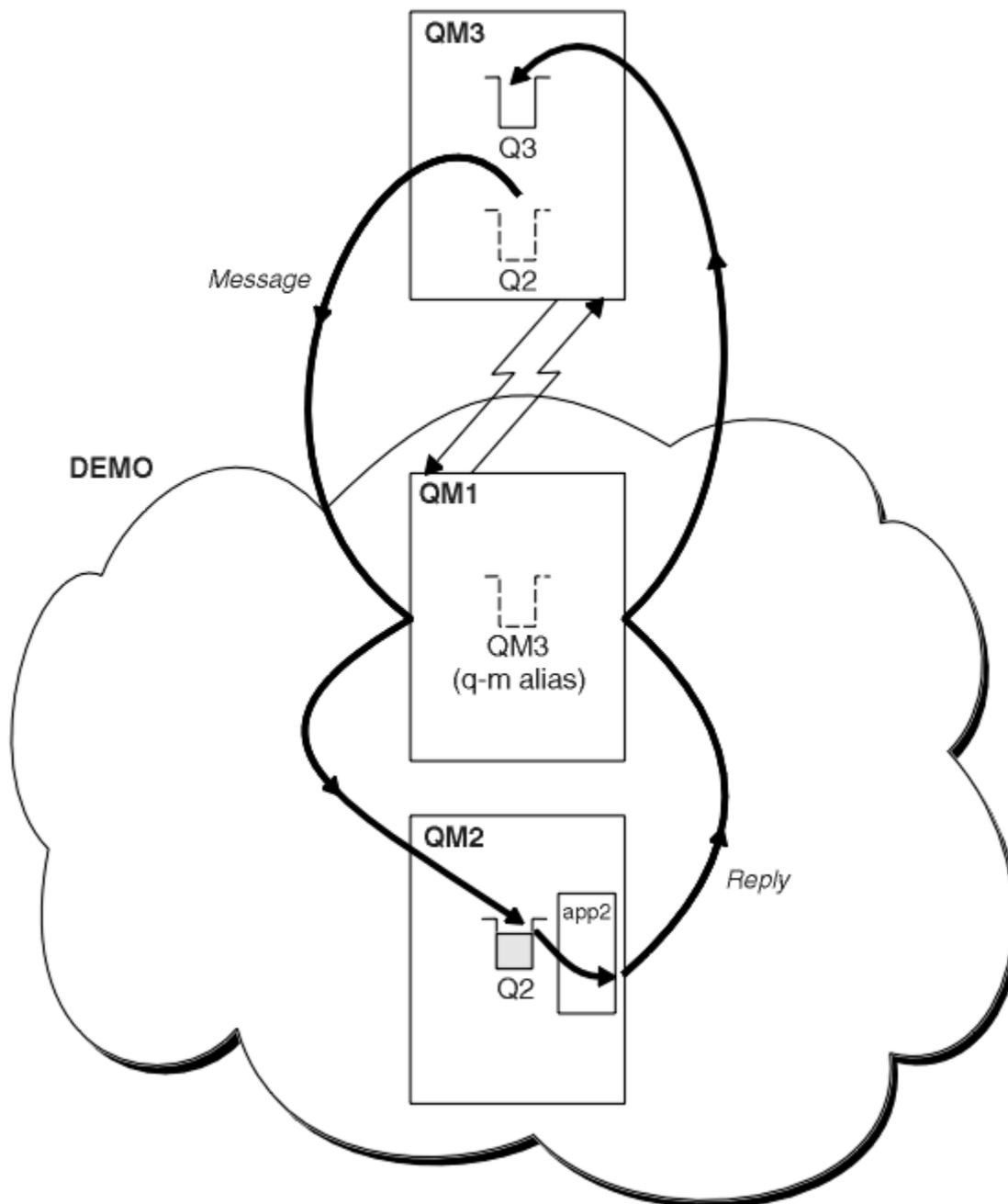


図 43. クラスターの外側にあるキュー・マネージャーからの書き込み

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 上の Q2 のリモート・キューを定義します。

```
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、252 ページの図 43 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、メッセージは、QM1 伝送キューからメッセージを取得する送信側チャンネルを使用して QM2 の Q2 に経路指定されます。

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 QM3 の外部にあるキュー・マネージャーのキュー・マネージャー別名を公示します。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、クラスター内のキュー・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)
```

応答を QM1 から QM3 に転送するために使用する伝送キューの名前の選択を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。ただし、QM3 は、キュー・マネージャー別名を使用してクラスターの残りの部分に通知する予定の名前と同じです。WebSphere MQ では、伝送キューとキュー・マネージャー別名の両方に、同じ名前を使用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・マネージャー別名は、QM3.XMIT という新しい伝送キューを QM3 キュー・マネージャー別名と結合します。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 上のサーバーは、応答で QM3 応答先キュー・マネージャー名を使用して QM3 をアドレス指定します。このソリューションでは、応答メッセージを返すキュー・マネージャーの別名として QM1 が QM3 を公示する必要がありましたが、QM1 は伝送キューの名前として QM3 を使用できませんでした。

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 自体ではなく QM3 への応答のためにキュー・マネージャーの別名を通知し、伝送キューの名前との競合を回避します。

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 は、キュー・マネージャーの別名 QM3.ALIAS をクラスターの残りの部分に通知し、それを実際のキュー・マネージャーの名前 QM3 に結合します。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 によって構成される応答先キュー・マネージャー名フィールドに QM3.ALIAS をコーディングします。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。

関連タスク

254 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・バル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。
 - 254 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特例として妥当です。

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

251 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

この例では、[256 ページの図 44](#) を参照して、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(' ')
```

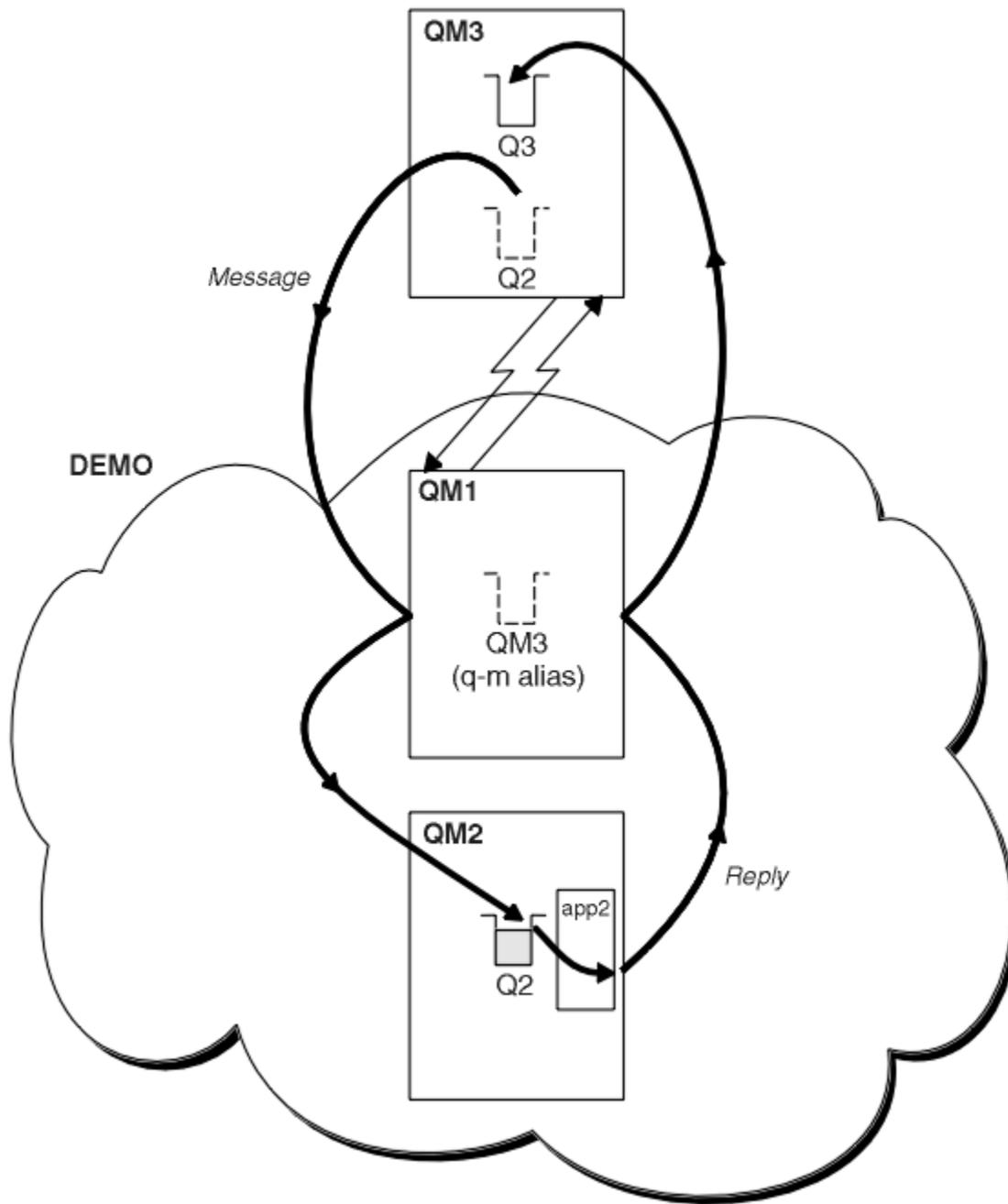


図 44. クラスターの外側にあるキュー・マネージャーからの書き込み

QM1 上の QREMOTE 定義により、キュー・マネージャー別名 DEMO がゲートウェイ・キュー・マネージャーに認識されます。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する必要なく、キュー・マネージャーの別名 DEMO を使用して、DEMO 上のクラスター・キューにメッセージを送信できます。

クラスター名を使用してクラスターに接続する伝送キューに名前を付けるという規則を採用した場合、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)
```

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

258 ページの図 45 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。

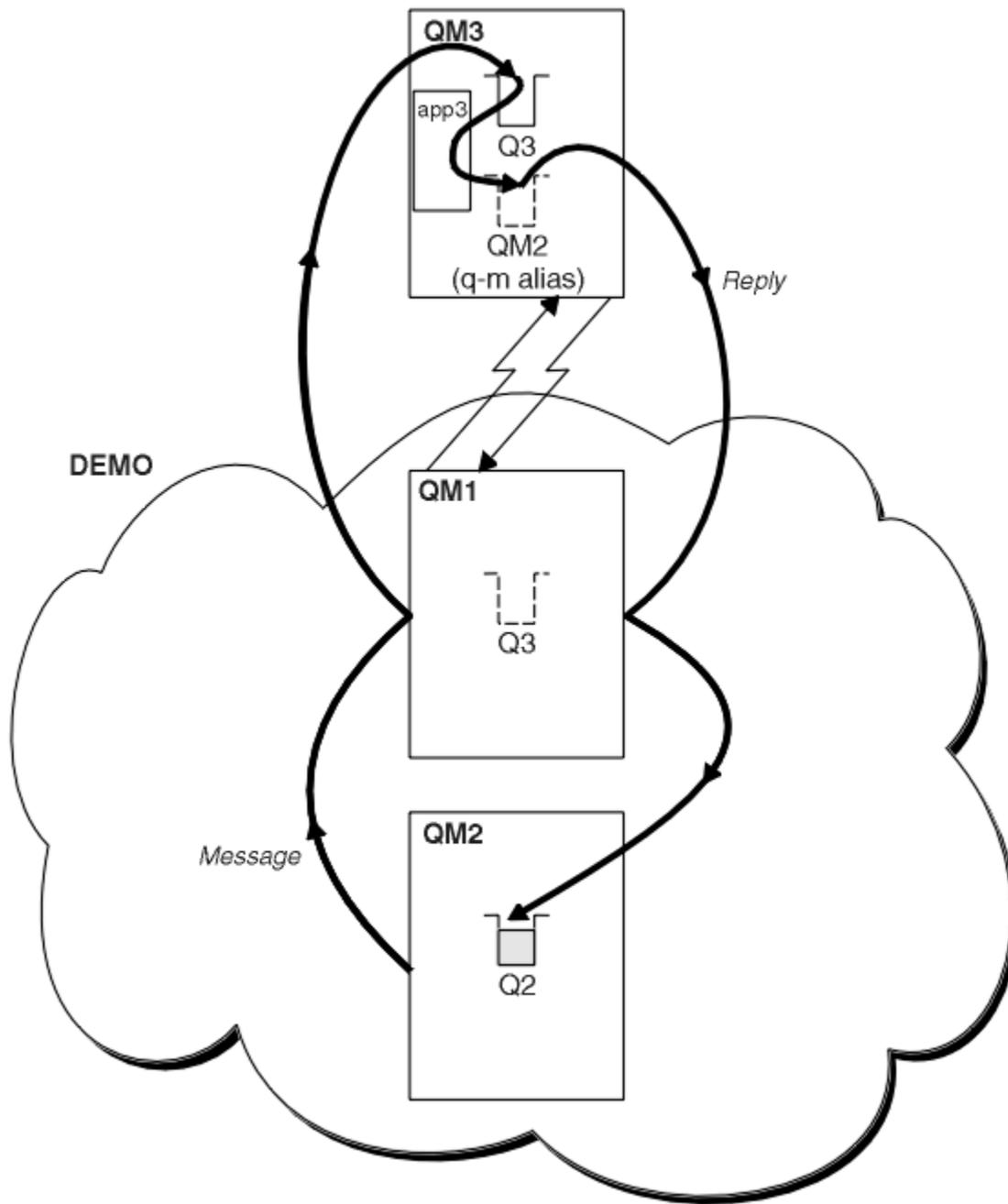


図 45. クラスターの外側にあるキュー・マネージャーへの書き込み

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、[258ページの図45](#)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。

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

[251ページの『クラスターへの要求/応答の構成』](#)の[252ページの図43](#)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、[260ページの図46](#)のクラスター外部のキュー・マネージャーQM3が、キューQ2に要求を送信します。ワークロード・バランシングを使用するために、Q2はクラスターDEMO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。Q2という名前のキューはキュー・マネージャーQM2およびQM4では定義しますが、ゲートウェイ・キュー・マネージャーQM1では定義しません。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、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、[260ページの図46](#)の中では明示的に表示されていません。

以下の手順は、[251ページの『クラスターへの要求/応答の構成』](#)の[252ページの図43](#)を拡張しています。

手順

1. 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)
```

2. 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。

3. QM3 でキュー・マネージャー別名 Q3 を作成します。

```
DEFINE QREMOTE(Q3) RNAME(' ') RQMNAME(' ') CLUSTER(DEMO) DEFBIND(NOTFIXED)
```

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 と空白 ' ' を同等にするキュー・マネージャー別名定義の名前です。

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

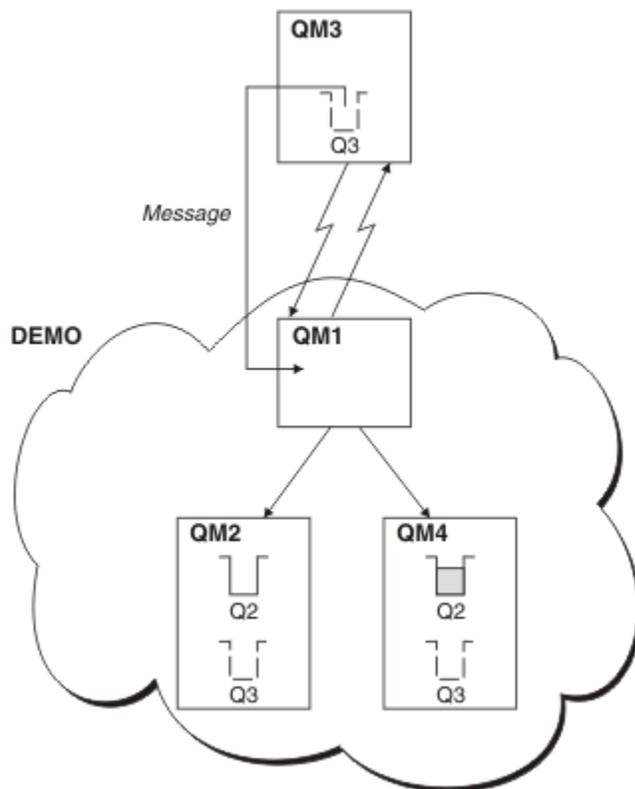


図 46. クラスターの外側にあるキュー・マネージャーからの書き込み

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、このメッセージは、QREMOTE 定義のために、ゲートウェイ・キュー・マネージャー QM1 を経由して転送されます。QM1 は、ワークロード・บาลancingを使用して、キュー Q2 に書き込まれたメッセージを、2 つのキュー・マネージャー (QM2 と QM4) 上で複数の Q2 に分散します。これらのキュー・マネージャーには、Q3 のクラスター・キュー・マネージャー別名があります。

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。

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。[180 ページの『クラスターのオーバーラップ』](#)を参照。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。

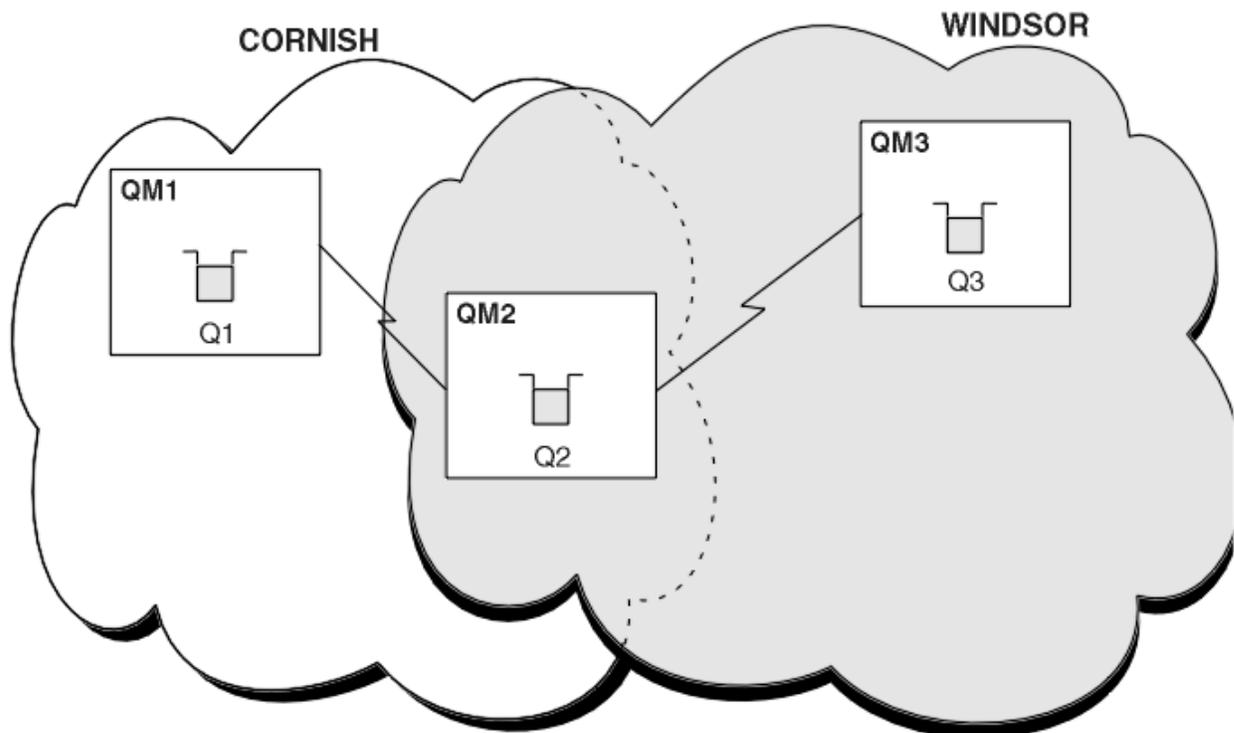


図 47. 複数のクラスターにわたるブリッジ化

[261 ページの図 47](#)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 NAMLIST(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。261 ページの図 47 の例では、QM2 で次の定義が必要です。

```
DEFINE QALIAS(MYQ3) TARGET(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。

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。これは、ROME をキュー・マネージャーとして定義し、公示します。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 の宛先に到達することができます。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、両方とも同じにすることができます。

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、219 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。

この図は、応答メッセージが (RQ と呼ばれる) 一時動的キューに戻すためにたどるパスを示しています。QM3 に接続されているサーバー・アプリケーションは、キュー・マネージャー名 QM2 を使用して応答キューを開きます。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 に経路指定します。QM1 は、メッセージを QM2 に経路指定します。

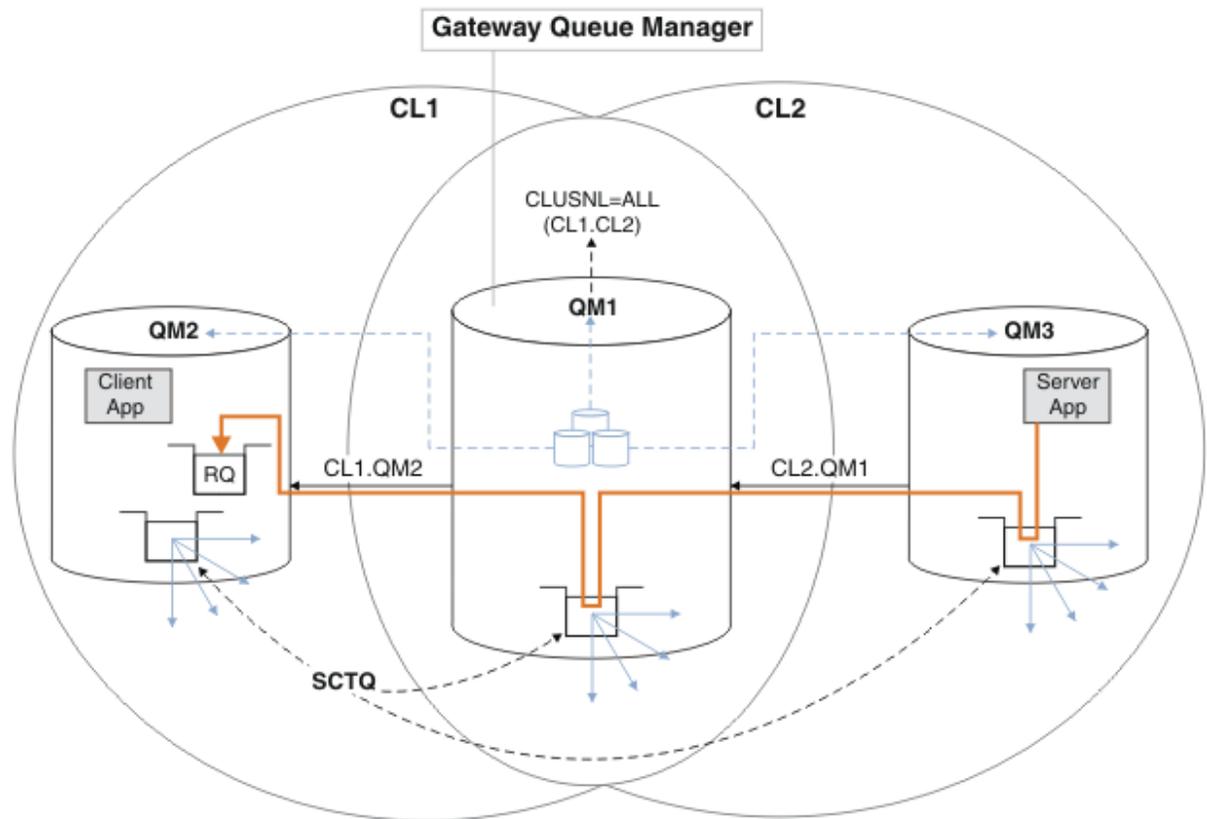


図 48. 応答メッセージを異なるクラスターに返すためのキュー・マネージャー別名の使用

このルーティングは、次のようにして行われます。QM1では、各クラスター内のすべてのキュー・マネージャーにキュー・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。この場合、QM2別名はクラスターCL1とCL2の両方でクラスター化され、CL1内の実際のキュー・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、応答先キュー名RQ、および応答先キュー・マネージャー名QM2を使用して応答メッセージを作成します。キュー・マネージャーの別名定義QM2がクラスターCL2内のQM1で定義されており、キュー・マネージャーQM2がクラスターCL2内にないため、メッセージはQM1に経路指定されます。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、メッセージをQM1上のクラスター伝送キューに入れ、QM2に転送します。QM1はメッセージをQM2に経路指定します。これは、QM1のQM2でのキュー・マネージャー別名定義がQM2を実際のターゲット・キュー・マネージャーとして定義しているためです。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の両方が同じクラスターCL1にあるため、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の接続情報を検出し、メッセージをQM2に経路指定します。メッセージがQM1によって転送されるようにするには、サーバー・アプリケーションが、オプションDEFBINDをMQBND_BIND_NOT_FIXEDに設定した状態で応答キューを開いている必要があります。サーバー・アプリケーションがオプションMQBND_BIND_ON_OPENで応答キューを開いている場合、メッセージは転送されず、送達不能キューに入ります。

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・マネージャーには、EDINBURGH キューのインスタンスが含まれていません。
- 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、[259 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。

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。

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) TARGET(LOCAL) CLUSTER(C)
```

このコマンドは、PUBLIC というキューをクラスター C 内のキュー・マネージャーに通知します。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) TARGET(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。

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。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がメッセージを受信し、相互に独立して動作します。

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すべてのキュー・マネージャーが、そのキューを宛先とするメッセージを処理できます。アプリケーションはメッセージを送信するときにキュー・マネージャーを指定しません。どのキュー・マネージャーがメッセージを処理するかは、ワークロード管理アルゴリズムが決定します。

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

[クラスター](#)

[クラスターが機能する仕組み](#)

[161 ページの『クラスター化と分散キューイングとの比較』](#)

キュー・マネージャーに接続するために定義する必要があるコンポーネントを、分散キューイングを使用する場合とクラスター化を使用する場合で比較します。

[164 ページの『クラスターのコンポーネント』](#)

クラスターは、キュー・マネージャー、クラスター・リポジトリ、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。

[186 ページの『IBM WebSphere MQ クラスターの管理』](#)

IBM WebSphere MQ クラスターの作成、拡張、および保守を行うことができます。

[251 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

[159 ページの『キュー・マネージャー・クラスターの構成』](#)

クラスターの動作とクラスター構成の設計について調べるには、および単純なクラスターのセットアップ方法の例については、このトピックにあるリンクを使用してください。

[186 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。

[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。

267 ページの図 49 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・インスタンスが存在する QM2 または QM4, 上でアプリケーションが実行されている場合、デフォルトで Q3 のローカル・インスタンスが開きます。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。

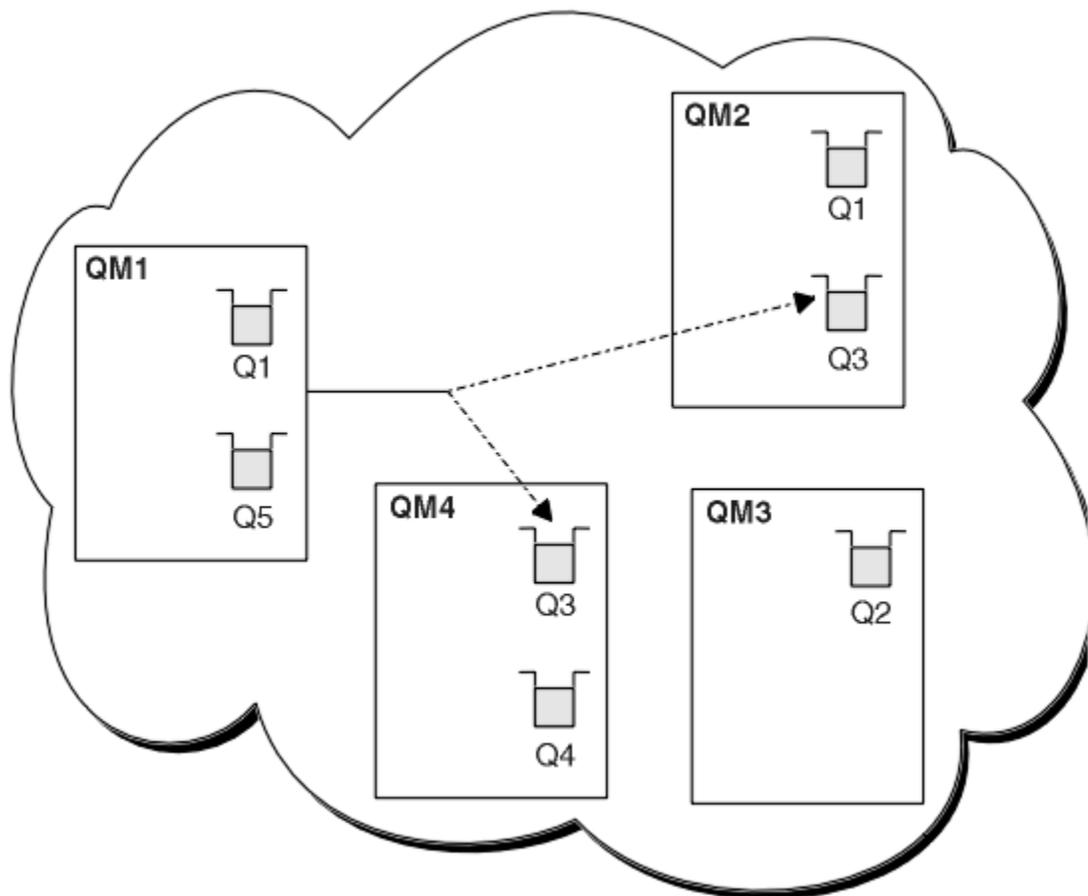


図 49. 同じキューの複数のインスタンスがあるクラスター

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。

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。詳しくは、[279 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。

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。WebSphere 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

269 ページの図 50 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。

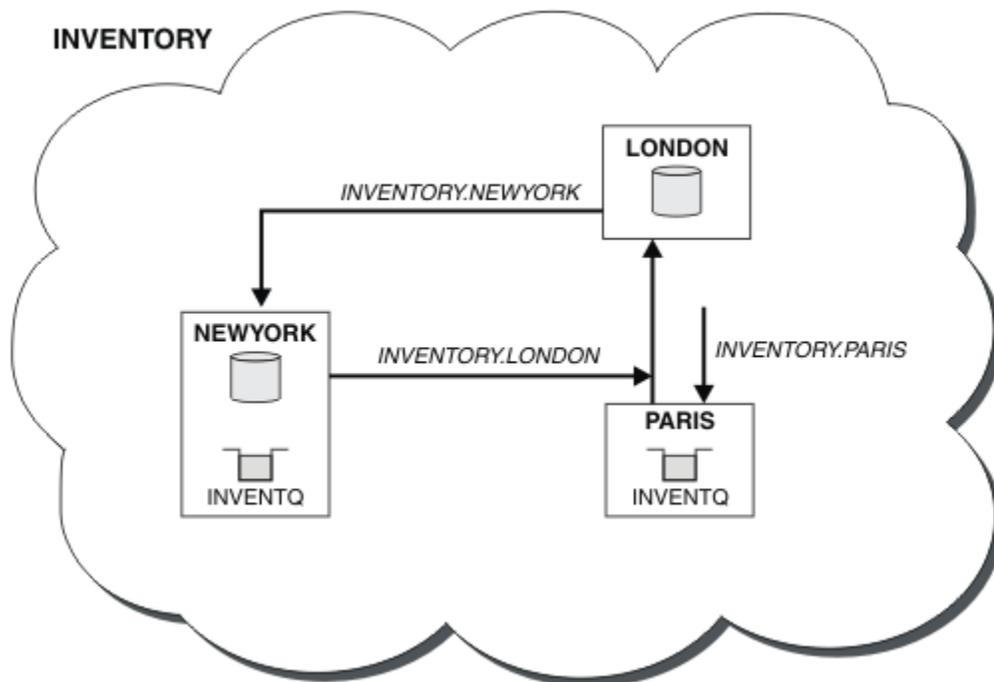


図 50. 3つのキュー・マネージャーで構成される INVENTORY クラスター

このクラスターの変更では、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。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。

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』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。
- 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。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。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. で CLUSSDR チャンネルを定義します。

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)
CONNAME(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。WebSphere 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

[271 ページの図 51](#) 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。

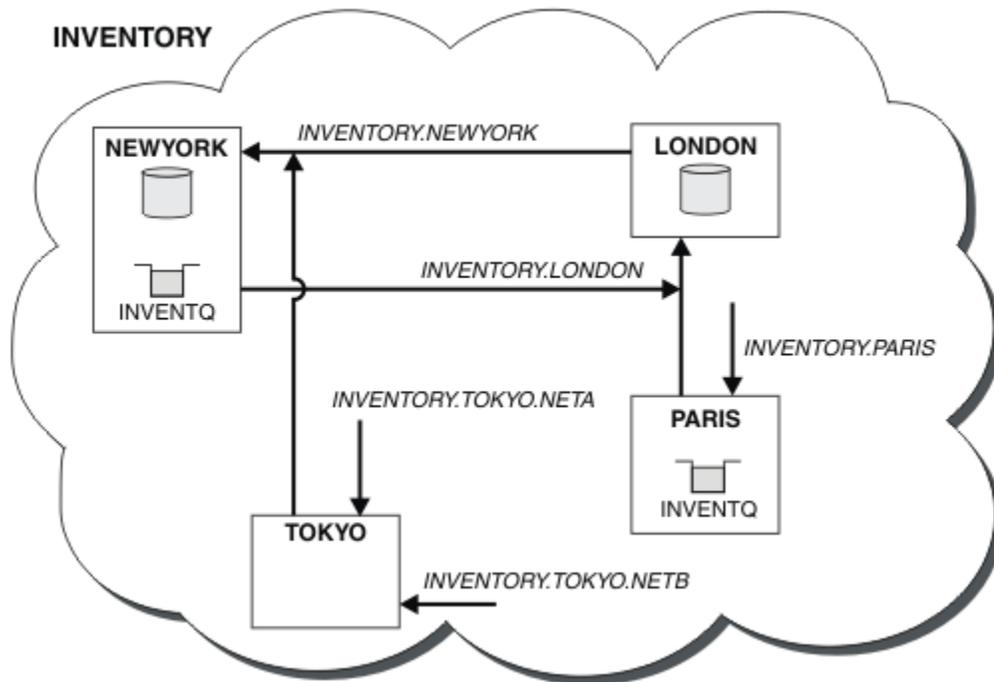


図 51. 4つのキュー・マネージャーで構成される INVENTORY クラスタ

3つの定義のみを作成して、クラスタにキュー・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。

関連タスク

197 ページの『クラスタにキュー・マネージャーを追加する』

以下の指示に従って、作成したクラスタにキュー・マネージャーを追加します。クラスタ・キューおよびトピックへのメッセージは、単一のクラスタ伝送キュー SYSTEM.CLUSTER.TRANSMIT.QUEUE を使用して転送されます。

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。

始める前に

注: クラスタへの変更をクラスタ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

- 269 ページの『クラスタ内での2つのネットワークの使用』での説明に従って、INVENTORY クラスタを設定しました。このクラスタには、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) DESCR('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。

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

- [197 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。
- 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にのみ使用されます。

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。特定のキュー・マネージャーに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。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。WebSphere 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

274 ページの図 52 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。

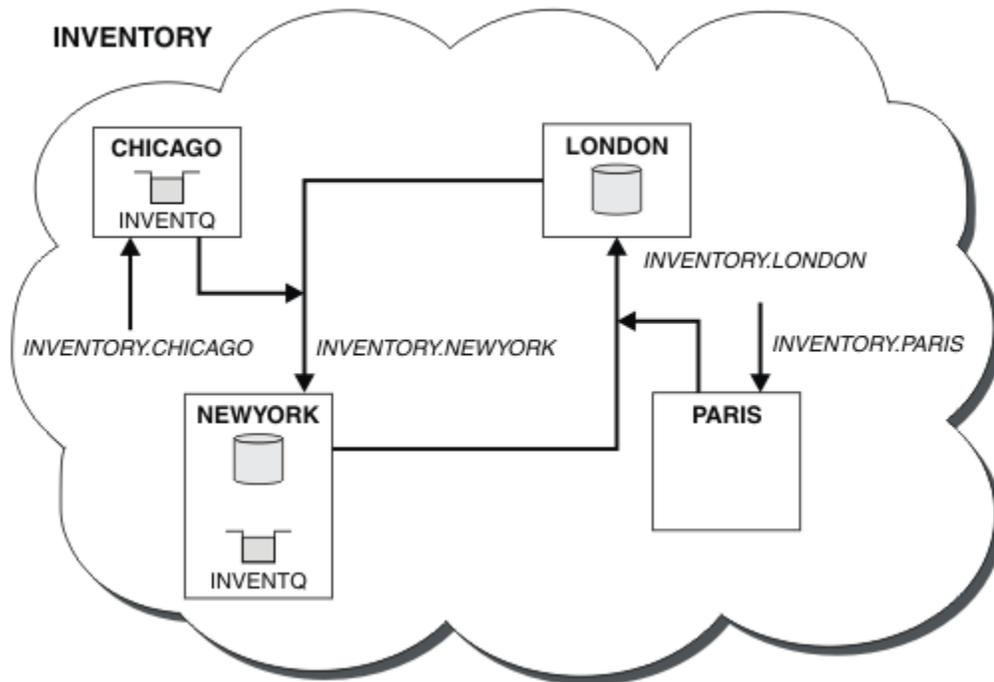


図 52. 4つのキュー・マネージャーで構成される INVENTORY クラスタ

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。

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：クラスタへの変更をクラスタ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)  
CONNAME(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. WebSphere 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

276 ページの [図 53](#) 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。

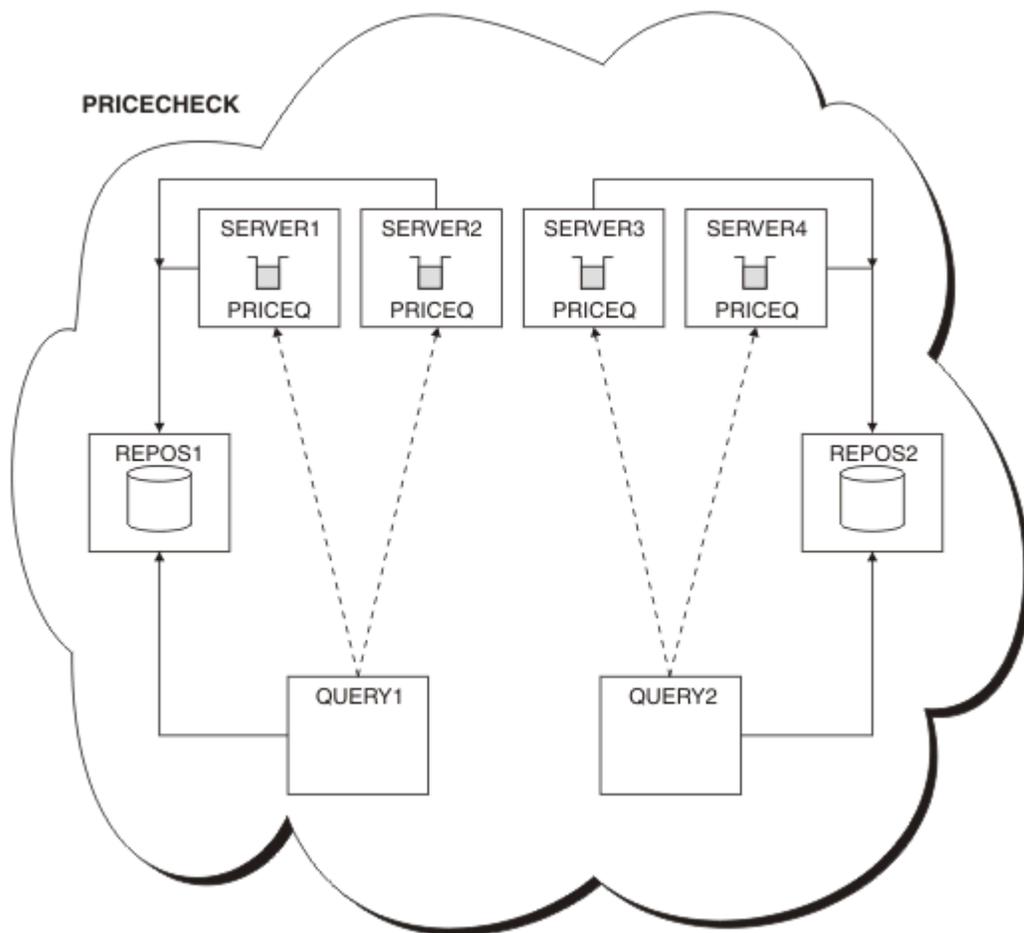


図 53. 4 つのサーバー・キュー・マネージャー、2 つのリポジトリ、および 2 つの照会キュー・マネージャーを持つ PRICECHECK クラスタ

PRICECHECK クラスタには、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。

次のタスク

PRICECHECK クラスタには、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。

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

- 197 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ーを保有します。PARIS は部分リポジトリーを保有して、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ーを決める。
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。

```
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ネルを定義する必要があります。このチャネルにより、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、ロサンゼルのキュー・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 チャネルを変更する。
ロサンゼルのキュー・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。WebSphere 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

276 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トアップされるクラスターを示します。

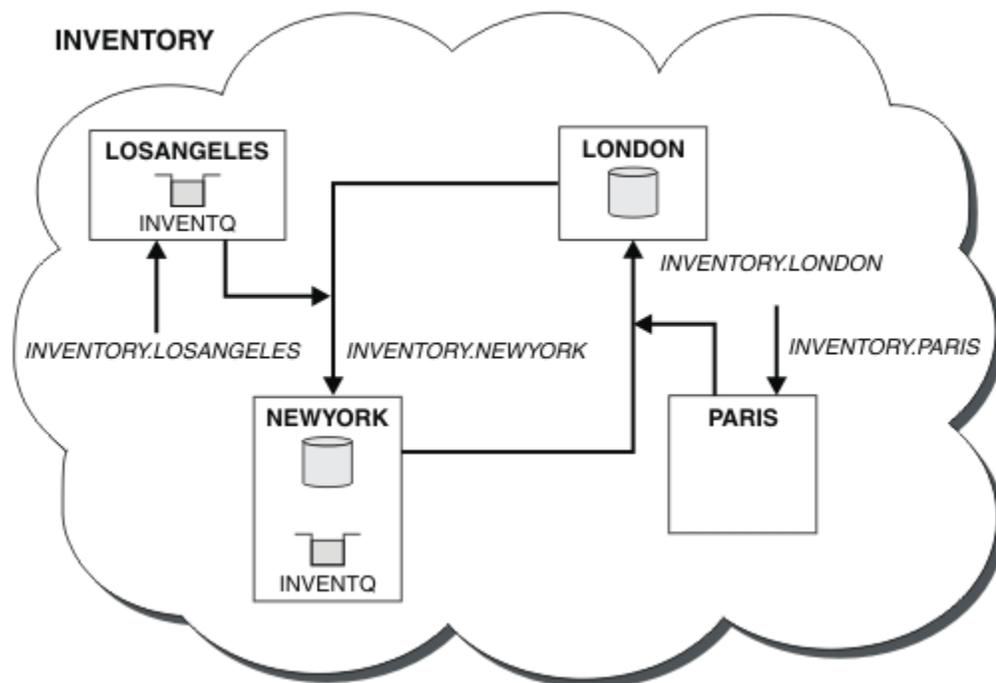


図 54. 4 つのキュー・マネージャーが格納されている INVENTORY クラスター

クラスターをこのように変更するときに、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。

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・マネージャーはCLWLUSEQ(ANY) on ALTER QMGRを指定します。
 - キュー・マネージャーのキューがCLWLUSEQ(ANY)を指定しています。

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る
- 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 呼び出しで MQOO_BIND_ON_OPEN オプションを使用して、強制的にすべてのメッセージが同じ宛先に書き込まれるようにします。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

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 を使用するようになります。

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_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 が指定されているが、メッセージがグループ化されていない場合、動作は MQOO_BIND_NOT_FIXED と同等です。

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。

クラスター化: ベスト・プラクティス

クラスターは、キュー・マネージャーが相互接続するためのメカニズムを提供します。このセクションで説明しているベスト・プラクティスは、テストおよびお客様からのフィードバックに基づいています。

クラスターのセットアップを成功させるには、適切な計画を立てることと IBM WebSphere MQ に関する基礎知識 (適切なアプリケーション管理やネットワーク設計など) を十分に理解することが必要です。先に進む前に、以下にリストされている関連トピックで提供されている情報をよく理解してください。

関連概念

[クラスタリング](#)

[相互通信の概念](#)

[クラスターが機能する仕組み](#)

クラスター化: オーバーラップするクラスターについての特殊な考慮事項

このトピックでは、IBM WebSphere MQ クラスターを計画および管理するための指針について記載しています。この情報は、テストおよびお客様からのフィードバックに基づく指針を示すものです。

クラスター所有権

以下の情報を読む前に、クラスターのオーバーラップについて十分に理解してください。必要な情報については、[180 ページの『クラスターのオーバーラップ』](#)および [260 ページの『クラスター間のメッセージパスの構成』](#)を参照してください。

オーバーラップするクラスターで構成されたシステムを構成および管理する際には、以下に従うことが最善です。

- 前述のとおり、IBM WebSphere MQ クラスターは「疎結合」されていますが、クラスターを1つの管理単位として考えると役立ちます。この概念を使用する理由は、クラスターがスムーズに機能するためには、個々のキュー・マネージャーに関する定義の相互作用が重要となるためです。例えば、ワークロード・バランシングが行われるクラスター・キューを使用する場合には、一人の管理者または1つのチームがメッセージに考えられるすべての宛先一式を理解していることが重要ですが、この宛先一式は、クラスター全体に分散された定義に依存します。さらに当然のことながら、クラスター送信側/受信側チャネルのペアは、クラスター全体で互換性がなければなりません。
- 上記の概念を考えると、(別々のチームや個人によって管理されることになる) 複数のクラスターが集まる場合には、ゲートウェイ・キュー・マネージャーの管理を制御する明確なポリシーを実施することが重要です。
- オーバーラップするクラスターは単一の名前空間として扱うと便利です。チャンネル名とおキュー・マネージャー名は、1つのクラスター全体で固有でなければなりません。トポロジー全体で固有であると、さらに管理しやすくなります。適切な命名規則に従うことが最善です。使用可能な命名規則については、[179 ページの『クラスターの命名規則』](#)を参照してください。
- 場合によっては、管理とシステム管理の連携が必須であり、それを避けることができないことがあります。例えば、オーバーラップする必要がある別個のクラスターのそれぞれを所有する組織間での連携です。クラスターをオーバーラップする際には、誰が何を所有しているのか、および実施可能な規則と慣例をはっきりと理解しておく、クラスター化をスムーズに実行するのに役立ちます。

オーバーラップするクラスター: ゲートウェイ

一般に、複数のクラスターを管理するよりも、1つのクラスターを管理するほうが容易です。したがって、通常は、多数の小さなクラスター (例えばアプリケーションごとのクラスター) を作成するようなことを避けるのが適切です。

ただし、サービス・クラスを提供する目的で、オーバーラップするクラスターを実装することもできます。以下に例を示します。

- 少数のクラスターがパブリッシュ/サブスクライブ用である同心円クラスターがある場合。詳しくは、[システムのサイズ変更の方法](#)を参照してください。

- 一部のキュー・マネージャーが異なるチームによって管理される場合。詳しくは、前のセクション [282 ページ](#)の『[クラスター所有権](#)』を参照してください。
- 組織的または地理的観点から意味がある。
- 同等のクラスターがネーム・レゾリューションで機能する場合(例えば、既存のクラスターに SSL または TLS を実装する場合)。

クラスターをオーバーラップさせることによるセキュリティ上の利点はありません。2つの異なるチームで管理されるクラスターをオーバーラップさせることで、それらのチームおよびトポロジーが効率的に結合されます。次のとおりです。

- そのようなクラスターで公示される名前は、もう1つのクラスターでアクセス可能になります。
- 1つのクラスターで公示される名前は、適格となるメッセージを処理するためにもう1つのクラスターで公示可能になります。
- ゲートウェイに隣接するキュー・マネージャー上の非公示オブジェクトは、ゲートウェイがメンバーとして属しているクラスターから解決可能です。

ネーム・スペースは両方のクラスターの和集合であり、1つのネーム・スペースとして扱われる必要があります。したがって、オーバーラップするクラスターの所有権は、両方のクラスターのすべての管理者の間で共有されます。

複数のクラスターが存在するシステムで、あるクラスターのキュー・マネージャーから別のクラスターのキュー・マネージャーにメッセージを経路指定することが要件となる場合もあります。そのような場合には、複数のクラスターを何らかの方法で相互接続する必要があり、優れたパターンとして、クラスター間でゲートウェイ・キュー・マネージャーを使用することが挙げられます。この配置によって、Point-to-Point チャネルが網目のように積み重なって管理しにくくなるという事態を回避し、セキュリティ・ポリシーなどの問題を一箇所で管理できるようになります。この配置を達成するには、次に示す2種類の方法があります。

1. 2つ目のクラスター受信側を定義して、1つ(または複数)のキュー・マネージャーを両方のクラスターに配置する。このような配置にすると管理定義は少なくなりますが、既に述べたとおり、オーバーラップするクラスターの所有権は両方のクラスターのすべての管理者の間で共有されます。
2. 従来の Point-to-Point チャネルを使用して、クラスター1のキュー・マネージャーとクラスター2のキュー・マネージャーをペアとして組み合わせる。

上記のいずれの場合でも、トラフィックを適切にルーティングするために各種のツールを使用できます。特に、別のクラスターへのルーティングを行うには、キュー別名またはキュー・マネージャー別名を使用できます。ブランクの **QPMNAME** プロパティを設定したキュー・マネージャー別名は、必要に応じてワークロード・バランシングを再駆動します。

関連概念

[179 ページ](#)の『[クラスターの命名規則](#)』

キュー・マネージャーが属するクラスターを識別する命名規則を使用して、同じクラスター内のキュー・マネージャーの命名について考えます。チャネル名の命名に類似した命名規則を使用し、チャネルの特性を記述するために命名規則を拡張します。

クラスター化: トポロジー設計上の考慮事項

このトピックでは、IBM WebSphere MQ クラスターを計画および管理するための指針について記載しています。この情報は、テストおよびお客様からのフィードバックに基づく指針を示すものです。

ユーザー・アプリケーションおよび内部管理プロセスをあらかじめどこに配置するのかについて考えておくことにより、多くの問題を回避できたり、あるいは後で最小化したりできます。このトピックでは、クラスターが大きくなっていった場合にパフォーマンスを向上させ、保守タスクを簡単な操作で行えるようにするための、設計上の決定事項について説明します。

- [284 ページ](#)の『[クラスター・インフラストラクチャーのパフォーマンス](#)』
- [284 ページ](#)の『[完全リポジトリ](#)』
- [285 ページ](#)の『[アプリケーションは完全リポジトリでキューを使用しなければなりませんか](#)』
- [286 ページ](#)の『[チャネル定義の管理](#)』

クラスター・インフラストラクチャーのパフォーマンス

アプリケーションがクラスター内のキュー・マネージャー上にあるキューを開こうとすると、そのキュー・マネージャーは、クラスター内における該当のキューの場所を認識できるようにするため、自らのインタレストをキューの完全リポジトリに登録します。キューの場所または構成に対する更新は、完全リポジトリからインタレスト・キュー・マネージャーへ自動的に送信されます。このようなインタレストの登録は、内部的に、サブスクリプションと呼ばれます (IBM WebSphere MQ でメッセージのパブリッシュ/サブスクライブに使用される IBM WebSphere MQ サブスクリプションとは異なります)。

クラスターに関するすべての情報は、あらゆる完全リポジトリを経由して送信されます。したがって、完全リポジトリは、クラスター内の管理メッセージ・トラフィックのために常時使用されています。これらのサブスクリプションの管理、伝送、および結果的に生成される構成メッセージに大量のシステム・リソースが使用されると、クラスター・インフラストラクチャーの負荷が大幅に増加する可能性があります。この負荷を可能な限り認識し、最小限に抑えるために、いくつかの考慮事項があります。

- クラスター・キューを使用する個々のキュー・マネージャーが多いほど、システム内のサブスクリプションが増加します。したがって、変更が発生し、関連するサブスクライバーに通知する必要がある場合には、管理オーバーヘッドがさらに大きくなります。このことは、特に、完全リポジトリ・キュー・マネージャーに当てはまります。不要なトラフィックおよび完全リポジトリの負荷を最小限に抑える方法の1つは、類似するアプリケーション (すなわち、同一のキューを処理するアプリケーション) をより少ない数のキュー・マネージャーに接続することです。
- システム内のサブスクリプションの数だけでなく、クラスター・オブジェクトの構成に対する変更の頻度 (クラスター・キューの構成の頻繁な変更など) もパフォーマンスに影響を与える可能性があります。
- キュー・マネージャーが複数のクラスターのメンバーである (つまり、オーバーラップしているクラスター・システムの一部である) 場合、キューでインタレストが作成されると、メンバーとして属している各クラスターについてサブスクリプションが発生することになります。同一のキュー・マネージャーが複数のクラスターの完全リポジトリであっても同様です。このような配置はシステムの負荷を増加させます。したがって、単一のクラスターではなく複数のオーバーラップ・クラスターが必要かどうかを検討する理由の1つにもなります。
- アプリケーション・メッセージ・トラフィック (つまり、IBM WebSphere MQ アプリケーションによってクラスター・キューに送信されるメッセージ) は、完全リポジトリを経由せずに宛先キュー・マネージャーに到達します。このメッセージ・トラフィックは、クラスターへのメッセージの入口となるキュー・マネージャーとクラスター・キューが存在するキュー・マネージャーの間で直接送信されます。したがって、完全リポジトリ・キュー・マネージャーが偶然にもそれらの2つのキュー・マネージャーの1つである場合を除き、完全リポジトリ・キュー・マネージャーに関しては高速のアプリケーション・メッセージ・トラフィックに対応する必要はありません。そのため、クラスター・インフラストラクチャーの負荷が高いクラスターでは、完全リポジトリ・キュー・マネージャーをアプリケーション・メッセージ・トラフィックに使用しないことをお勧めします。

完全リポジトリ

リポジトリとは、クラスターを構成する各キュー・マネージャーについての情報の集まりを指します。クラスター内の各キュー・マネージャーについての全情報をホストしているキュー・マネージャーには、完全リポジトリが含まれます。完全リポジトリおよび部分リポジトリの詳細については、[164 ページの『クラスター・リポジトリ』](#)を参照してください。

完全リポジトリは、信頼性が高く、可能な限り可用性の高いサーバーに保持する必要があり、単一障害点を避けなければなりません。クラスター設計には常に2つの完全リポジトリが必要です。1つの完全リポジトリに障害が発生した場合でも、クラスターは動作を継続することができます。

クラスター内のキュー・マネージャーによって行われたクラスター・リソース (クラスター・キューなど) に対する更新の詳細は、そのキュー・マネージャーからクラスター内の最大2つの完全リポジトリ (クラスター内に完全リポジトリ・キュー・マネージャーが1つしかない場合は1つ) に送信されます。それらの完全リポジトリは、情報を保持し、クラスター内のインタレスト・キュー・マネージャー (つまり、完全リポジトリにサブスクライブしているキュー・マネージャー) にその情報を伝搬します。クラスターの各メンバーでクラスター・リソースの最新の状況が認識されるようにするため、各キュー・マネージャーが常に少なくとも1つの完全リポジトリ・キュー・マネージャーと通信できなければなりません。

キュー・マネージャーは、何らかの理由でいずれの完全リポジトリ・キュー・マネージャーとも通信できない場合には、既にキャッシュに入れられているレベルの情報を基にしばらくは機能し続けることができますが、新しい更新情報を取得することや、以前は使用されていなかったクラスター・リソースにアクセスすることはできません。

このため、常に2つの完全リポジトリを使用可能にしておくことを目指してください。ただし、1つの完全リポジトリがない場合でも短い期間であればクラスターは十分に機能するので、この配置は、極端な手段を講じなければならないということを意味するものではありません。

クラスターに2つの完全リポジトリ・キュー・マネージャーが必要な理由は、クラスター情報を使用可能にしておくため以外に、もう1つあります。その理由とは、リカバリーのため、完全リポジトリのキャッシュに保持されるクラスター情報が2つの場所に保管されるようにすることです。完全リポジトリが1つだけであり、そこで保持されているクラスター情報が失われた場合、クラスターが再び機能するようにするには、クラスター内のすべてのキュー・マネージャーに対して手操作による介入を行う必要があります。しかし、完全リポジトリが2つあれば、情報は常に2つの完全リポジトリにパブリッシュされ、サブスクライブされるので、最小限の作業で、障害が発生した完全リポジトリを回復することができます。

- 2つの完全リポジトリを採用するクラスター設計では、クラスターのユーザーに影響を及ぼすことなく、完全リポジトリ・キュー・マネージャーの保守作業を行うことができます。クラスターはリポジトリが1つしかなくても機能し続けるので、可能であれば、一度に1つずつリポジトリを停止して、保守作業を行い、稼働状態に戻します。2番目の完全リポジトリが故障している場合でも、実行中のアプリケーションは、最低でも3日間は影響を受けません。
- 地理的理由から特定の場所にある完全リポジトリを使用しているなど、3番目のリポジトリを使用する理由がある場合を除き、これら2つのリポジトリを採用する設計を使用してください。完全リポジトリが3つあると、どの2つが現在使用されているのか分からず、複数のワークロード管理パラメーター間の相互作用によって管理上の問題が発生することもあります。完全リポジトリを3つ以上使用することは推奨されません。
- より優れた可用性が必要な場合は、完全リポジトリ・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ホストすることや、プラットフォーム固有の高可用性サポート機能を使用して可用性を改善することを検討してください。
- 手動で定義されたクラスター送信側チャンネルを使用して、すべての完全リポジトリ・キュー・マネージャーを完全に相互接続する必要があります。何らかの正当な理由があり、クラスターに3つ以上の完全リポジトリが確保されていない場合は、特に注意しなければなりません。そのような場合、1つまたは複数のチャンネルが見逃され、そのことがすぐに判明しない可能性が高くなります。完全な相互接続が確立されない場合、問題の診断が困難になるということがしばしば発生します。なぜならば、いくつかの完全リポジトリですべてのリポジトリ・データが保持されず、その結果、クラスター内のキュー・マネージャーが認識するクラスターの全体像が、接続している完全リポジトリによって異なることになるからです。

アプリケーションは完全リポジトリでキューを使用しなければなりませんか

完全リポジトリは、ほとんどの場合、他のキュー・マネージャーとまったく同じようなものです。したがって、完全リポジトリでアプリケーション・キューをホストし、アプリケーションを他のキュー・マネージャーに直接接続することが可能です。アプリケーションは完全リポジトリでキューを使用しなければなりませんか

一般的に受け入れられている回答は「いいえ」です。そのような構成を使用することはできますが、多くのお客様は完全リポジトリ・キュー・マネージャーを完全リポジトリ・クラスター・キャッシュの保守専用にしておくことを好まれます。ここでは、どちらを選択するか決定する際の考慮事項を紹介していますが、最終的には、ご使用の環境における個々の要望に対応するにはクラスター・アーキテクチャーが適しているものと思われま

- アップグレード: 通常、新規リリースの IBM WebSphere MQ で新しいクラスター機能を使用するには、最初に、該当のクラスターの完全リポジトリ・キュー・マネージャーをアップグレードします。クラスター内のアプリケーションで新機能を使用する必要がある場合に、共存している多数のアプリケーションをテストしなくても完全リポジトリ (および部分リポジトリの一部のサブセット) をアップデートできるようになるので便利です。

- 保守: 同様に、完全リポジトリに緊急保守を適用する必要がある場合は、アプリケーションには触れずに、それらのリポジトリを再始動するか、**REFRESH** コマンドを使用してリフレッシュすることができます。
- パフォーマンス: クラスターが成長し、完全リポジトリ・クラスター・キャッシュの保守に対する要求が強まってきても、アプリケーションを切り離しておくことで、アプリケーションのパフォーマンスがシステム・リソースの競合による影響を受けるというリスクが軽減されます。
- ハードウェア要件: 通常、強力な完全リポジトリは不要であり、例えば、可用性が十分に期待できる単純な UNIX サーバーで十分です。あるいは、非常に大規模なクラスターまたは頻繁に変更が加えられるクラスターの場合は、完全リポジトリ・コンピューターのパフォーマンスを考慮する必要があります。
- ソフトウェア要件: 通常、完全リポジトリでアプリケーション・キューをホストすることにする場合、これらの要件が主な理由となります。小規模なクラスターでは、コロケーションは、全体としてキュー・マネージャーまたはサーバーの数を削減する必要があることを意味します。

チャンネル定義の管理

単一クラスター内でも、2つのキュー・マネージャー間で複数の経路を提供する複数のチャンネル定義が存在することがあります。

単一クラスター内に並列チャンネルが存在していることが利点である場合もありますが、この設計上の決定事項は十分に検討する必要があります。この設計では、複雑さが増すばかりでなく、チャンネルの使用効率低下が下がり、パフォーマンスが低下する可能性があります。そのような状況が発生するのは、通常、テストでは、一定の速度で多くのメッセージが送信されるので、並列チャンネルがフルに使用されるためです。しかし、メッセージのストリームが一定ではない現実世界の状態では、メッセージ・フローがチャンネルからチャンネルへと切り替えられていくにつれて、ワークロード・バランシングのアルゴリズムが原因でパフォーマンスが低下します。

キュー・マネージャーが複数のクラスターのメンバーである場合は、クラスターごとに別々の CLUSRCVR チャンネルを定義する代わりに、クラスター名前リストを指定して1つのチャンネル定義を使用するという選択肢があります。ただし、そのようにセットアップすると、後で管理が困難になることがあります。例えば、SSLが1番目のクラスターに適用されるが2番目のクラスターには適用されない場合があります。そのため、別々の定義を作成することが望ましく、179 ページの『[クラスターの命名規則](#)』で推奨されている命名規則はこのような場合に対応するようになっています。

複数チャンネルのワークロード・バランシング

この情報は、このテーマを高度に理解してもらうことを意図したものです。この主題の基礎の説明(ここでの情報を使用する前に理解しておかなければならない知識)については、266 ページの『[クラスターによるワークロードの管理](#)』、[ワークロード・バランシング](#)、および[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。

クラスター・ワークロード管理アルゴリズムは、多くのツールからなるセットを提供しますが、それらのツールはすべて、それらの動作方法と相互作用を十分に理解してから相互に使用すべきです。ワークロード・バランシング・プロセスに対するチャンネルの重要度がただちに明確にならない場合があります。ワークロード管理のラウンドロビン・アルゴリズムでは、クラスター・キューを所有するキュー・マネージャーへの複数のクラスター・チャンネルが、そのキューの複数のインスタンスであるかのように処理されます。このプロセスについては、以下の例で詳しく説明します。

1. クラスターにはキューをホストする2つのキュー・マネージャー (QM1 と QM2) があります。
2. QM1 へのクラスター受信側チャンネルは5つあります。
3. QM2 へのクラスター受信側チャンネルは1つのみです。
4. **MQPUT** または **MQOPEN on QM3** がインスタンスを選択すると、アルゴリズムは QM2 よりも QM1 にメッセージを送信する可能性が5倍高くなります。
5. ステップ4の状況が発生する理由は、このアルゴリズムに見えているのは、(5+1)からの選択肢としての6つのオプションであるため、このアルゴリズムは、QM1 への5つのチャンネルすべてと、QM2 への1つのチャンネルについてラウンドロビンを行うからです。

その他の優れた動作は、たまたまローカル・キュー・マネージャー上に構成されているインスタンスを1つもつクラスター・キューにメッセージを送信する場合でも、IBM WebSphere MQ は、ローカル・クラスター受信側チャンネルの状態に基づいて、メッセージをそのキューのローカル・インスタンスとリモート・インスタンスのどちらに送信するかが決まることです。このシナリオでは、以下のようになります。

1. メッセージを送信する場合、ワークロード管理アルゴリズムでは、個々のクラスター・キューを調べるのではなく、宛先に到達できるクラスター・チャンネルを調べます。
2. ローカル宛先に到達するため、(メッセージの送信には使用されませんが) ローカル受信側チャンネルがこのリストに含められます。
3. ローカル受信側チャンネルが停止すると、ワークロード管理アルゴリズムでは、デフォルトで代替りのインスタンスを優先的に使用します(ただし、その CLUSRCVR が停止していない場合)。宛先のローカル CLUSRCVR インスタンスが複数存在し、少なくとも1つが停止状態でない場合、そのローカル・インスタンスは引き続き適格になります。

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

クラスター内のキュー・マネージャー間のメッセージ・フローは、分離することができます。さまざまなクラスター送信側チャンネルによって転送されるメッセージを、それぞれに異なるクラスター伝送キューに配置できます。この手法は、単一のクラスターでも、オーバーラップするクラスターでも使用できます。このトピックでは、使用する手法を選択する際に参考となる例およびベスト・プラクティスを説明します。

アプリケーションをデプロイするときには、他のアプリケーションと共有する IBM WebSphere MQ リソース、および共有しないリソースを選択できます。共有できるリソースには、さまざまなタイプがあります。そのうち主なタイプは、サーバー自体、キュー・マネージャー、チャンネル、およびキューです。少ない数の共有リソースでアプリケーションを構成し、個々のアプリケーションに個別のキュー、チャンネル、キュー・マネージャー、さらにはサーバーを割り振るといった方法があります。この方法では、システム構成全体が大きくなり、複雑さも増してきます。IBM WebSphere MQ クラスターを使用すると、多数のサーバー、キュー・マネージャー・キュー、およびチャンネルを管理する場合の複雑さは軽減されますが、別の共有リソースであるクラスター伝送キュー `SYSTEM.CLUSTER.TRANSMIT.QUEUE` が導入されます。

288 ページの図 55 は、大規模な IBM WebSphere MQ デプロイメントからのスライスです。ここでは、`SYSTEM.CLUSTER.TRANSMIT.QUEUE` を共有する重要性が示されています。この図のアプリケーション Client App は、クラスター CL1 内のキュー・マネージャー QM2 に接続されています。Client App からのメッセージは、アプリケーション Server App によって処理されます。このメッセージは、Server App が CLUSTER2 内のキュー・マネージャー QM3 にあるクラスター・キュー Q1 から取得します。クライアント・アプリケーションとサーバー・アプリケーションは同じクラスター内に配置されていないため、メッセージはゲートウェイ・キュー・マネージャー QM1 によって転送されます。

クラスター・ゲートウェイを構成する通常の方法は、ゲートウェイ・キュー・マネージャーをすべてのクラスターのメンバーにすることです。ゲートウェイ・キュー・マネージャーには、すべてのクラスター内のクラスター・キューに対応するクラスター別名キューを定義します。クラスター化されたキュー別名は、すべてのクラスター内で使用可能になります。クラスター化されたキュー別名に入れられたメッセージは、ゲートウェイ・キュー・マネージャーを介して、それぞれの正しい宛先にルーティングされます。ゲートウェイ・キュー・マネージャーは、クラスター別名キューに送信されたメッセージを QM1 上の共通 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` に入れます。

ハブ・スポーク・アーキテクチャーでは、クラスター間のすべてのメッセージが、ゲートウェイ・キュー・マネージャーを通過しなければなりません。したがって、すべてのメッセージ・フローが、QM1 上の1つのクラスター伝送キュー `SYSTEM.CLUSTER.TRANSMIT.QUEUE` を通過することになります。

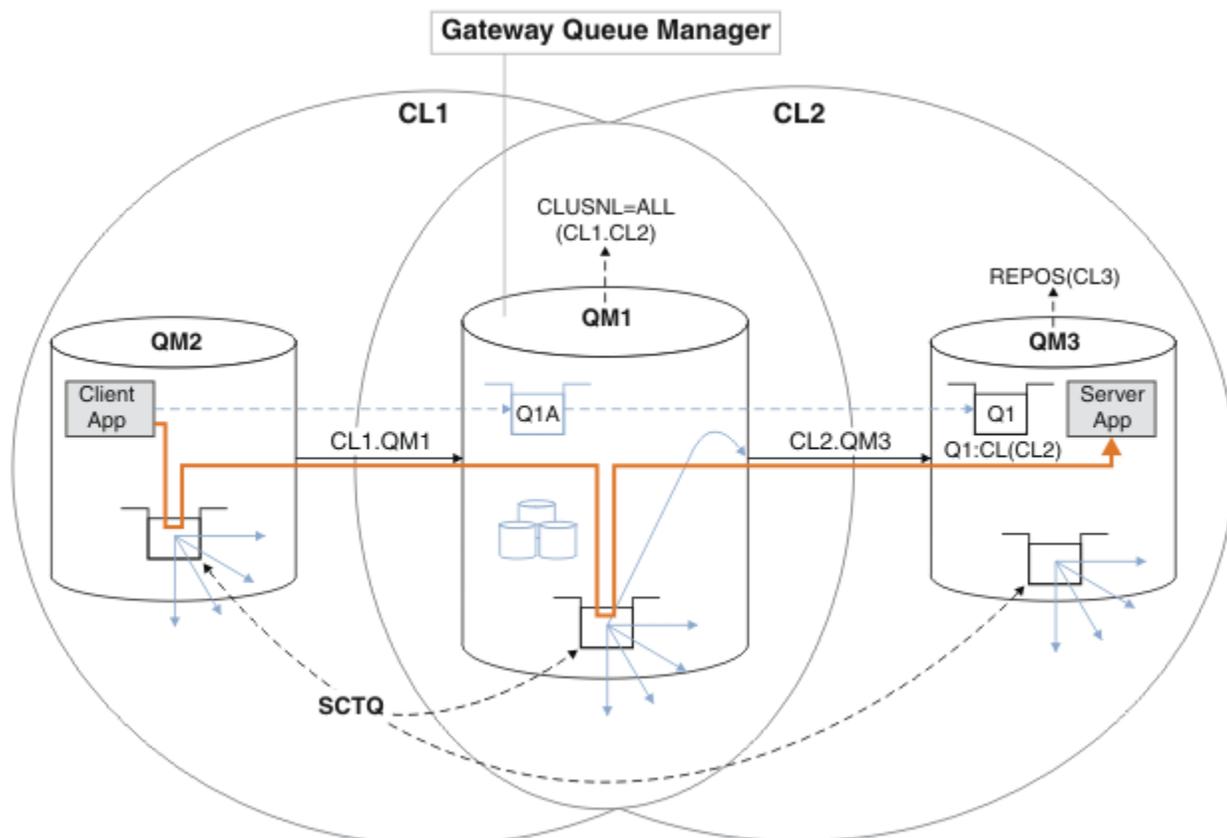
パフォーマンスの観点からすると、キューが1つでも問題にはなりません。共通伝送キューは、一般にパフォーマンス・ボトルネックを意味しません。ゲートウェイでのメッセージ・スループットは、主にそのゲートウェイに接続するチャンネルのパフォーマンスによって決まります。キューの数やチャンネルが使用するキューに置かれたメッセージの数は通常、スループットに影響を与えません。

その他の観点から見ると、複数のアプリケーションに単一の伝送キューを使用することには、以下の欠点があります。

- ある宛先へのメッセージ・フローを、別の宛先へのメッセージ・フローから分離することができません。メッセージがそれぞれに異なるクラスター内の異なるキュー・マネージャーを宛先としているとしても、メッセージのストレージを分離してからメッセージを転送することは不可能です。

1つのクラスター宛先が使用不可になると、その宛先のメッセージが単一の伝送キューに蓄積され、最終的には、その伝送キューがこれらのメッセージで満杯になってしまいます。伝送キューが満杯になると、あらゆるクラスター宛先の伝送キューへのメッセージの配置が中断されます。

- さまざまなクラスター宛先へのメッセージの転送をモニターするのは容易ではありません。すべてのメッセージは、単一の伝送キューに配置されます。伝送キューの深さを表示しても、すべての宛先にメッセージが転送されているかどうかは、ほとんど明らかになりません。



注：288 ページの図 55 および以降の図には、異なるタイプの矢印が示されています。実線の矢印は、メッセージ・フローを表します。実線矢印のラベルは、メッセージ・チャンネル名です。グレーの実線矢印は、SYSTEM.CLUSTER.TRANSMIT.QUEUE からクラスター送信側チャンネルへの潜在的メッセージ・フローです。黒の破線は、ラベルをそのターゲットに結びます。グレーの破線矢印は、参照を示します。例えば、Client App による MQOPEN 呼び出しから、クラスター別名キュー定義 Q1A への参照です。

図 55. ハブ・スポーク・アーキテクチャにデプロイされた、IBM WebSphere 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

288 ページの図 55 では、Server App のクライアントはキュー Q1A をオープンします。QM2 の SYSTEM.CLUSTER.TRANSMIT.QUEUE に入れられたメッセージは、QM1 の SYSTEM.CLUSTER.TRANSMIT.QUEUE に転送されてから、QM3 の Q1 に転送されます。このキューから、Server App アプリケーションはメッセージを受け取ります。

Client App からのメッセージは、QM2 および QM1 のシステム・クラスター伝送キューを通過します。288 ページの図 55 で目標となっているのは、ゲートウェイ・キュー・マネージャーでメッセージ・フローをクライアント・アプリケーションから分離することなので、メッセージは SYSTEM.CLUSTER.TRANSMIT.QUEUE には保管されません。他のすべてのクラスター・キュー・マネージャーでも、フローを分離できます。逆の方向で、クライアントに戻るフローを分離することも可能です。

ソリューションの説明を簡潔にするため、以降の説明ではクライアント・アプリケーションからの1つのフローのみを検討します。

クラスター・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・トラフィックを分離するためのソリューション

問題を解決する1つの方法は、キュー・マネージャー別名(リモート・キュー定義)を使用して、クラスター間にブリッジを確立することです。クラスター化されたリモート・キュー定義、伝送キュー、およびチャンネルを作成し、それぞれのメッセージ・フローをゲートウェイ・キュー・マネージャーで分離します(201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](#)』を参照)。

Version 7.5 からは、クラスター・キュー・マネージャーは単一のクラスター伝送キューに制限されません。次のいずれかを選択できます。

1. 追加のクラスター伝送キューを手動で定義し、それぞれの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定義します(204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)。
2. キュー・マネージャーが追加のクラスター伝送キューを自動的に作成して管理できるようにします。この場合、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定義されます(224ページの『[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](#)』を参照)。

一部のクラスター送信側チャンネルに手動で定義したクラスター伝送キューは、残りのクラスター送信側チャンネルを管理するキュー・マネージャーに結合できます。伝送キューを組み合わせるという方法は、204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](#)』場合に採用しています。このソリューションでは、クラスター間のほとんどのメッセージが共通 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` を使用します。非常に重要な1つのアプリケーションがあり、そのすべてのメッセージ・フローは、手動で定義された1つのクラスター伝送キューを使用して他のメッセージ・フローから分離されます。

204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](#)』場合の構成には、制限があります。あるクラスター・キューへと流れるメッセージ・トラフィックは、同じクラスター内の同じキュー・マネージャー上の別のクラスター・キューへと流れるメッセージ・トラフィックから分離されません。個々のキューへのメッセージ・トラフィックを分離するには、分散キューイングの一部であるリモート・キュー定義を使用するという方法があります。クラスターでは、複数のクラスター伝送キューを使用して、異なるクラスター送信側チャンネルに向かうメッセージ・トラフィックを分離できます。同じクラスター内の同じキュー・マネージャーにある複数のクラスター・キューは、クラスター送信側チャンネルを共有します。これらのキューのメッセージは、同じ伝送キューに保管された後、ゲートウェイ・キュー・マネージャーから転送されます。208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](#)』場合の構成では、この制限を回避するために、もう1つのクラスターを追加し、キュー・マネージャーとクラスター・キューをその新しいクラスターのメンバーにします。新しいキュー・マネージャーが、そのクラスター内の唯一のキュー・マネージャーである場合もあります。クラスターにさらにキュー・マネージャーを追加し、同じクラスターを使用して、これらのキュー・マネージャーでもクラスター・キューを分離するという方法もあります。

関連概念

160ページの『[アクセス制御と複数のクラスター伝送キュー](#)』

アプリケーションがメッセージをリモート・クラスター・キューに入れるタイミングをチェックするモードを3つの中から選択します。これらのモードはそれぞれ、リモートでのクラスター・キューに対するチェック、ローカルでの `SYSTEM.CLUSTER.TRANSMIT.QUEUE` に対するチェック、クラスター・キューまたはクラスター・キュー・マネージャーのローカル・プロファイルに対するチェックを行います。

282ページの『[クラスター化: オーバーラップするクラスターについての特殊な考慮事項](#)』

このトピックでは、IBM WebSphere MQ クラスターを計画および管理するための指針について記載しています。この情報は、テストおよびお客様からのフィードバックに基づく指針を示すものです。

171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](#)』

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。

180 ページの『クラスターのオーバーラップ』

クラスターのオーバーラップは、追加の管理機能を提供します。名前リストを使用して、オーバーラップするクラスターの管理に必要なコマンドの数を減らします。

関連タスク

リモート・クラスター・キューへのメッセージ書き込み権限の付与

201 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』

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。

204 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。

208 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。

224 ページの『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。

219 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。

260 ページの『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。

291 ページの『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の選択について手順を追って説明します。1つの共通デフォルト・キュー、個々のデフォルト・キュー、または手動で定義したキューを構成できます。複数のクラスター伝送キューの構成は、z/OS以外のプラットフォームに適用されます。

関連資料

439 ページの『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。

`setmqaut`

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の選択について手順を追って説明します。1つの共通デフォルト・キュー、個々のデフォルト・キュー、または手動で定義したキューを構成できます。複数のクラスター伝送キューの構成は、z/OS以外のプラットフォームに適用されます。

始める前に

294 ページの『[使用するクラスター伝送キュー・タイプの選択方法](#)』を参照してください。

このタスクについて

キュー・マネージャーを構成する方法を計画する際には、クラスター伝送キューに関して以下の選択肢があります。

1. クラスター・メッセージを転送するデフォルト・クラスター伝送キューをどれにするか。
 - a. 共通クラスター伝送キュー `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。
 - b. 個々のクラスター伝送キュー。キュー・マネージャーが、個々のクラスター伝送キューを管理します。これらのキューは、モデル・キュー `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` から永続動的キューとして作成します。使用するクラスター送信側チャンネルごとに1つのクラスター伝送キューを作成します。
2. 手動で作成することにしたクラスター伝送キューについては、さらに2つの選択肢があります。
 - a. 手動で構成することにしたクラスター送信側チャンネルごとに個別の伝送キューを定義する。この場合、伝送キューの **CLCHNAME** キュー属性をクラスター送信側チャンネルの名前に設定します。この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選択します。
 - b. クラスター送信側チャンネルのグループのメッセージ・トラフィックを同じクラスター伝送キューに結合する (292 ページの [図 56](#) を参照)。この場合、それぞれの共通伝送キューの **CLCHNAME** キュー属性を総称クラスター送信側チャンネル名に設定します。総称クラスター送信側チャンネル名は、クラスター送信側チャンネル名をグループ化するためのフィルターです。例えば、`SALES.*` は、名前が `SALES.` で始まるすべてのクラスター送信側チャンネルをグループ化します。フィルター・ストリングには、任意の場所に複数のワイルドカード文字を配置できます。ワイルドカード文字は、アスタリスク「`*`」です。これは、ゼロから任意の数の文字を表します。

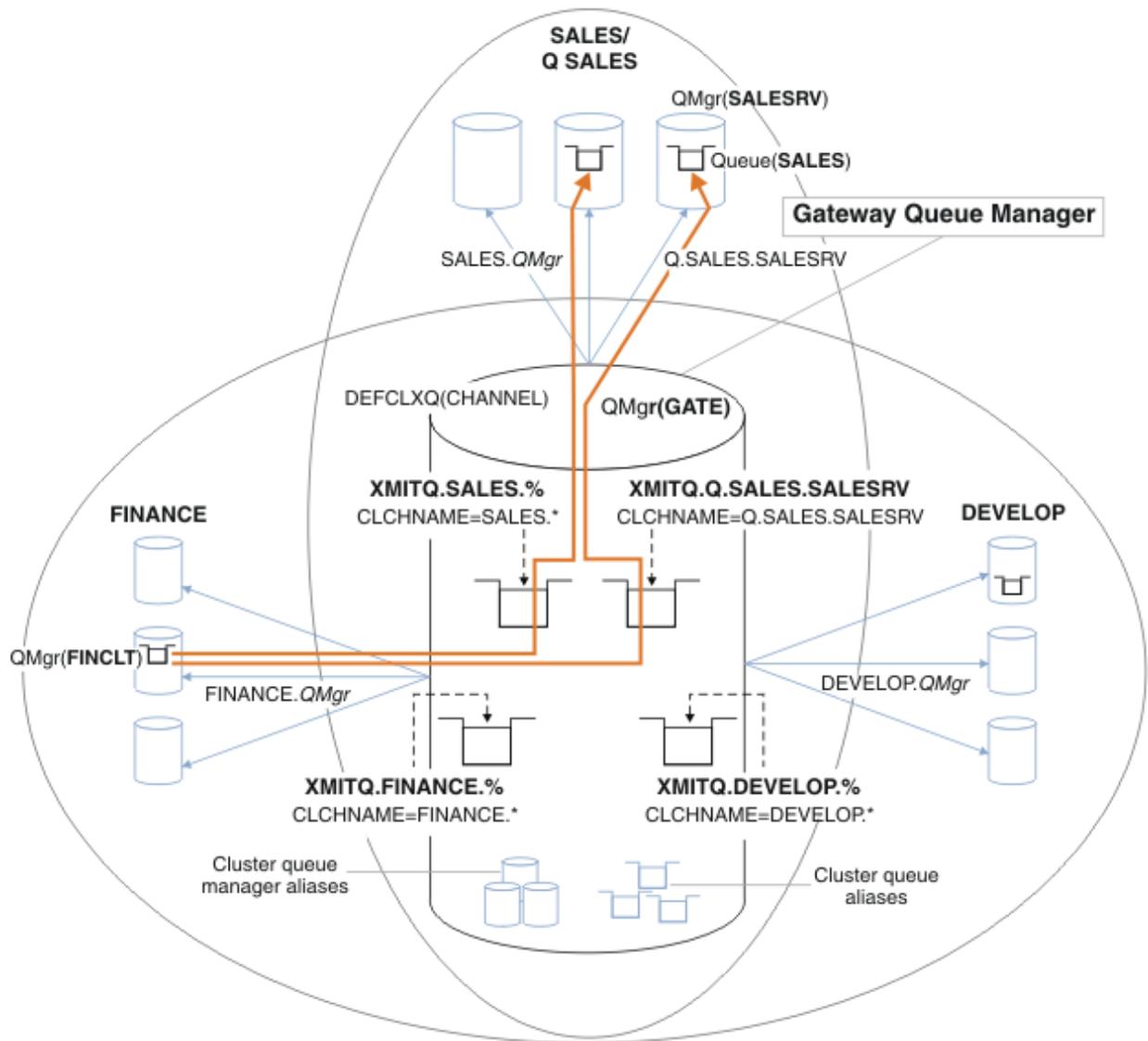


図 56. さまざまな部門別 IBM WebSphere MQ クラスタに固有の伝送キューの例

手順

1. 使用するデフォルト・クラスター伝送キューのタイプを選択します。
 - 単一のクラスター伝送キューを選択するか、またはクラスター接続ごとに個別のキューを選択します。デフォルト設定のままにするか、または以下の **MQSC** コマンドを実行します。

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. 他のフローとクラスター伝送キューを共有させないメッセージ・フローを分離します。
 - [296 ページの『クラスター化: 複数のクラスター伝送キューの構成例』](#)を参照してください。この例で、分離しなければならない SALES キューは SALES クラスタのメンバーであり、SALESRV 上にあります。SALES キューを分離するには、新しいクラスター Q.SALES を作成し、SALESRV キュー・マネージャーをそのメンバーにして、SALES が Q.SALES に属するように変更します。
 - メッセージを SALES に送信するキュー・マネージャーも、新しいクラスターのメンバーにしなければなりません。この例のようにクラスター・キュー別名とゲートウェイ・キュー・マネージャーを使用すると、多くの場合、ゲートウェイ・キュー・マネージャーを新しいクラスターのメンバーにするための変更を制限できます。

- 一方、ゲートウェイから宛先へのフローを分離しても、ゲートウェイからソース・キュー・マネージャーへのフローを分離することにはなりません。ただし、ゲートウェイへのフローを分離しなくても、ゲートウェイからのフローを分離するだけで十分である場合もあります。それで十分でない場合には、ソース・キュー・マネージャーを新しいクラスターに追加してください。メッセージがゲートウェイを通過するようにするには、クラスター別名を新しいクラスターに移動し、メッセージをターゲット・キュー・マネージャーに直接送信するのではなく、引き続きゲートウェイ上のクラスター別名に送信します。

以下の手順に従って、メッセージ・フローを分離します。

- フローの宛先を構成して、それぞれのフローのターゲット・キューが特定のクラスター内の当該キュー・マネージャー上の唯一のキューとなるようにします。
- 体系的な命名規則に従って、作成した新規クラスター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。
 - 282 ページの『[クラスター化: オーバーラップするクラスターについての特殊な考慮事項](#)』を参照してください。
- メッセージをターゲット・キューに送信するすべてのキュー・マネージャーに、分離された各宛先のクラスター伝送キューを定義します。
 - クラスター伝送キューの命名規則は、XMITQ. を先頭に付けた、クラスター・チャンネル名属性 CLCHNAME の値を使用することです。

3. ガバナンスまたはモニター要件を満たすクラスター伝送キューを作成します。

- 典型的なガバナンスおよびモニター要件では、クラスターごとに1つの伝送キュー、またはキュー・マネージャーごとに1つの伝送キューを作成することになります。クラスター・チャンネルの命名規則 `ClusterName.QueueManagerName` に従うと、キュー・マネージャーのクラスター、またはキュー・マネージャーがメンバーとなっているすべてのクラスターを選択する汎用チャンネル名を容易に作成できます (296 ページの『[クラスター化: 複数のクラスター伝送キューの構成例](#)』を参照)。
- 汎用チャンネル名に対応するように、アスタリスク記号を % 記号に置き換えてクラスター伝送キューの命名規則を拡張します。例:

```
DEFINE QLOCAL(XMITQ.SALES.%) USAGE(XMITQ) CLCHNAME(SALES.*)
```

関連概念

171 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](#)』

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。

287 ページの『[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](#)』

クラスター内のキュー・マネージャー間のメッセージ・フローは、分離することができます。さまざまなクラスター送信側チャンネルによって転送されるメッセージを、それぞれに異なるクラスター伝送キューに配置できます。この手法は、単一のクラスターでも、オーバーラップするクラスターでも使用できます。このトピックでは、使用する手法を選択する際に参考となる例およびベスト・プラクティスを説明します。

160 ページの『[アクセス制御と複数のクラスター伝送キュー](#)』

アプリケーションがメッセージをリモート・クラスター・キューに入れるタイミングをチェックするモードを3つの中から選択します。これらのモードはそれぞれ、リモートでのクラスター・キューに対するチェック、ローカルでの `SYSTEM.CLUSTER.TRANSMIT.QUEUE` に対するチェック、クラスター・キューまたはクラスター・キュー・マネージャーのローカル・プロファイルに対するチェックを行います。

282 ページの『[クラスター化: オーバーラップするクラスターについての特殊な考慮事項](#)』

このトピックでは、IBM WebSphere MQ クラスターを計画および管理するための指針について記載しています。この情報は、テストおよびお客様からのフィードバックに基づく指針を示すものです。

180 ページの『[クラスターのオーバーラップ](#)』

クラスターのオーバーラップは、追加の管理機能を提供します。名前リストを使用して、オーバーラップするクラスターの管理に必要なコマンドの数を減らします。

関連タスク

201 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](#)』

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。

204 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。

208 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。

224 ページの『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。

219 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。

260 ページの『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。

使用するクラスター伝送キュー・タイプの選択方法

各種のクラスター伝送キュー構成オプションの中から選択する方法。

Version 7.5 以降では、クラスター送信側チャンネルを関連付けるクラスター伝送キューを選択できるようになっています。

1. すべてのクラスター送信側チャンネルを単一のデフォルト・クラスター伝送キュー `SYSTEM.CLUSTER.TRANSMIT.QUEUE` に関連付けることができます。このオプションはデフォルトであり、バージョン Version 7.1 以前を実行するキュー・マネージャーでは唯一の選択肢です。
 2. すべてのクラスター送信側チャンネルが別個のクラスター伝送キューと自動的に関連付けられるように設定することができます。これらのキューは、キュー・マネージャーによってモデル・キュー `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` から作成され、`SYSTEM.CLUSTER.TRANSMIT.ChannelName` という名前が付けられます。キュー・マネージャーの属性 `DEFCLXQ` が `CHANNEL` に設定されている場合、チャンネルは一意的に名前が付けられたクラスター伝送キューを使用します。
-  **重要:** 旧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 `SYSTEM.CLUSTER.TRANSMIT.QUEUES` を使用している場合は、`SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` の `SHARE/NOSHARE` オプションが **SHARE** に設定されていることを確認してください。
3. 単一のクラスター伝送キューで処理する特定のクラスター送信側チャンネルを設定することができます。このオプションを選択するには、伝送キューを作成し、その `CLCHNAME` 属性をクラスター送信側チャンネルの名前に設定します。

4. 単一のクラスター伝送キューで処理するクラスター送信側チャンネルのグループを選択することができます。このオプションを選択するには、伝送キューを作成し、その **CLCHNAME** 属性を総称チャンネル名 (`ClusterName.*` など) に設定します。『282 ページの『クラスター化: オーバーラップするクラスターについての特殊な考慮事項』』で説明されている命名規則に従ってクラスター・チャンネルの名前を設定する場合、この名前によって、クラスター `ClusterName` 内のキュー・マネージャーに接続されたすべてのクラスター・チャンネルが選択されます。

一部のクラスター送信側チャンネルのデフォルト伝送キュー・オプションのいずれかを、任意の数の特定および汎用クラスター伝送キュー構成と組み合わせることができます。

ベスト・プラクティス

ほとんどの場合、既存の IBM WebSphere MQ インストール済み環境ではデフォルトの構成が最適な選択となります。クラスター・キュー・マネージャーは、クラスター・メッセージを単一のクラスター伝送キュー `SYSTEM.CLUSTER.TRANSMIT.QUEUE` に保管します。このデフォルトを変更して、異なるキュー・マネージャーと異なるクラスターのメッセージを個々の伝送キューに保管することも、独自の伝送キューを定義することもできます。

ほとんどの場合、新しい IBM WebSphere MQ インストール済み環境でもデフォルトの構成が最適な選択となります。デフォルト構成から、クラスター送信側チャンネルごとに1つの伝送キューを使用する代替デフォルトへの切り替えプロセスは、自動的に行われます。元の状態への切り替えも自動です。どちらを選択するかは重要ではありません。選択は元に戻すことができます。

別の構成を選択する理由は、機能やパフォーマンスよりも、ガバナンスおよび管理に関係します。いくつかの例外を除き、複数のクラスター伝送キューを構成することによって、キュー・マネージャーの動作に利益がもたらされることはありません。結果的にキューの数が増えて、すでにセットアップした、単一の伝送キューを参照するモニターおよび管理プロシージャを変更しなければなりません。そのため、結局は、別の選択を行う強力なガバナンス上または管理上の理由がない限り、デフォルト構成のままにすることが最善の選択です。

これらの例外は、どちらも、`SYSTEM.CLUSTER.TRANSMIT.QUEUE` に保管されるメッセージの数が増加した場合に起こる事態に関係しています。ある宛先へのメッセージを他の宛先へのメッセージから分離するためにあらゆる措置を講じると、ある宛先でのチャンネルおよび配信の問題が別の宛先への配信に影響しなくなるはずですが、ただし、`SYSTEM.CLUSTER.TRANSMIT.QUEUE` に保管されるメッセージの数は、ある1つの宛先へのメッセージの配信速度が遅いために増加することもあります。1つの宛先に対する `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上のメッセージ数が他の宛先へのメッセージの配信に影響することがあります。

1つの伝送キューがいっぱいになったために問題が発生するということがないようにするため、構成で十分なキャパシティを確保するようにしてください。そうすれば、宛先で障害が発生し、メッセージ・バックログが溜まり始めたとしても、問題を修正する時間をとることができます。

メッセージが、クラスター・ゲートウェイなどのハブ・キュー・マネージャー経由でルーティングされる場合、これらのメッセージは共通伝送キュー `SYSTEM.CLUSTER.TRANSMIT.QUEUE` を共有します。ゲートウェイ・キュー・マネージャー上の `SYSTEM.CLUSTER.TRANSMIT.QUEUE` に保管されているメッセージ数が最大件数に達すると、キュー・マネージャーは、件数が減るまで、この伝送キューへの新しいメッセージを拒否し始めます。この輻輳(ふくそう)は、ゲートウェイ経由でルーティングされるすべての宛先へのメッセージに影響を与えます。メッセージは、メッセージをゲートウェイに送信する他のキュー・マネージャー上の伝送キューにバックアップされます。この問題は、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、メッセージのスループットが低下し、メッセージが送信されてからその宛先に到着するまでの経過時間が長くなるという形で現れます。

単一の伝送キューでの輻輳の影響は、伝送キューがフルになる前から明らかになる可能性があります。メッセージ・トラフィックに大規模な非持続メッセージと小さなメッセージが混在する場合、小さなメッセージの配信時間は伝送キューがいっぱいになるにつれ長くなります。この遅延は、通常はディスクに書き込まれない大規模な非持続メッセージがディスクに書き込まれるために発生します。速度が重視される重大なメッセージ・フローが、他の混在メッセージ・フローとクラスター伝送キューを共有している場合には、そのメッセージ・フローを他のメッセージ・フローと切り分けるための特別メッセージ・パスを構成することを検討する価値があります(208 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)。

個々のクラスター伝送キューを構成するもう1つの理由は、ガバナンス要件を満たすため、あるいは、さまざまなクラスター宛先に送信されるメッセージのモニターを簡素化するためです。例えば、ある宛先へのメッセージが、他の宛先へのメッセージと伝送キューを決して共有しないことを実証しなければならない場合があります。

すべてのクラスター送信側チャンネルに異なるクラスター伝送キューを作成するには、デフォルト・クラスター伝送キューを制御するキュー・マネージャー属性 **DEFCLXQ** を変更します。複数の宛先が1つのクラスター送信側チャンネルを共有することは可能であるため、この目標を完全に満たすクラスターを計画する必要があります。208 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』の方法を、すべてのクラスター・キューに体系的に適用してください。目標とする結果は、別のクラスター宛先とクラスター送信側チャンネルを共有するクラスター宛先が1つもなくなることです。その結果、クラスター宛先へのメッセージはいずれも、別の宛先へのメッセージとクラスター伝送キューを共有することがなくなります。

特定のメッセージ・フローに別個のクラスター伝送キューを作成すると、その宛先へのメッセージのフローを簡単にモニターできるようになります。新しいクラスター伝送キューを使用するには、キューを定義し、そのキューをクラスター送信側チャンネルに関連付けて、チャンネルを停止して開始します。これは永続的な変更である必要はありません。しばらくの間、伝送キューをモニターするためにメッセージ・フローを切り分けて、後でデフォルトの伝送キューを再び使用するように戻すこともできます。

関連タスク

クラスター化: 複数のクラスター伝送キューの構成例

このタスクでは、複数のクラスター伝送キューの計画手順を、3つのオーバーラップするクラスターに適用します。ここで要件となるのは、1つのクラスター・キューへのメッセージ・フローを、その他すべてのメッセージ・フローから分離すること、そして各クラスターへのメッセージをそれぞれ異なるクラスター伝送キューに保管することです。

クラスター化: クラスター伝送キューの切り替え

既存の実働キュー・マネージャーのクラスター伝送キューに加えた変更を有効にしていく方法を計画します。

クラスター化: 複数のクラスター伝送キューの構成例

このタスクでは、複数のクラスター伝送キューの計画手順を、3つのオーバーラップするクラスターに適用します。ここで要件となるのは、1つのクラスター・キューへのメッセージ・フローを、その他すべてのメッセージ・フローから分離すること、そして各クラスターへのメッセージをそれぞれ異なるクラスター伝送キューに保管することです。

このタスクについて

このタスクのステップでは、291 ページの『クラスター化: クラスター伝送キューの構成方法の計画』で説明した手順を適用して、297 ページの図 57 に示す構成を達成する方法を説明します。これは、個々のクラスター伝送キューを設定して構成されたゲートウェイ・キュー・マネージャーを使用した、3つのオーバーラップするクラスターの一例です。クラスターを定義するための MQSC コマンドについては、299 ページの『サンプル・クラスターの作成』で説明しています。

この例には、2つの要件があります。1つは、ゲートウェイ・キュー・マネージャーから売り上げをログに記録する販売アプリケーションへのメッセージ・フローを分離することです。2つ目の要件は、さまざまな部門領域への送信を待機しているメッセージの数を任意の時点で照会することです。SALES、FINANCE、および DEVELOP の各クラスターがすでに定義されています。クラスター・メッセージは現在、SYSTEM.CLUSTER.TRANSMIT.QUEUE から転送されています。

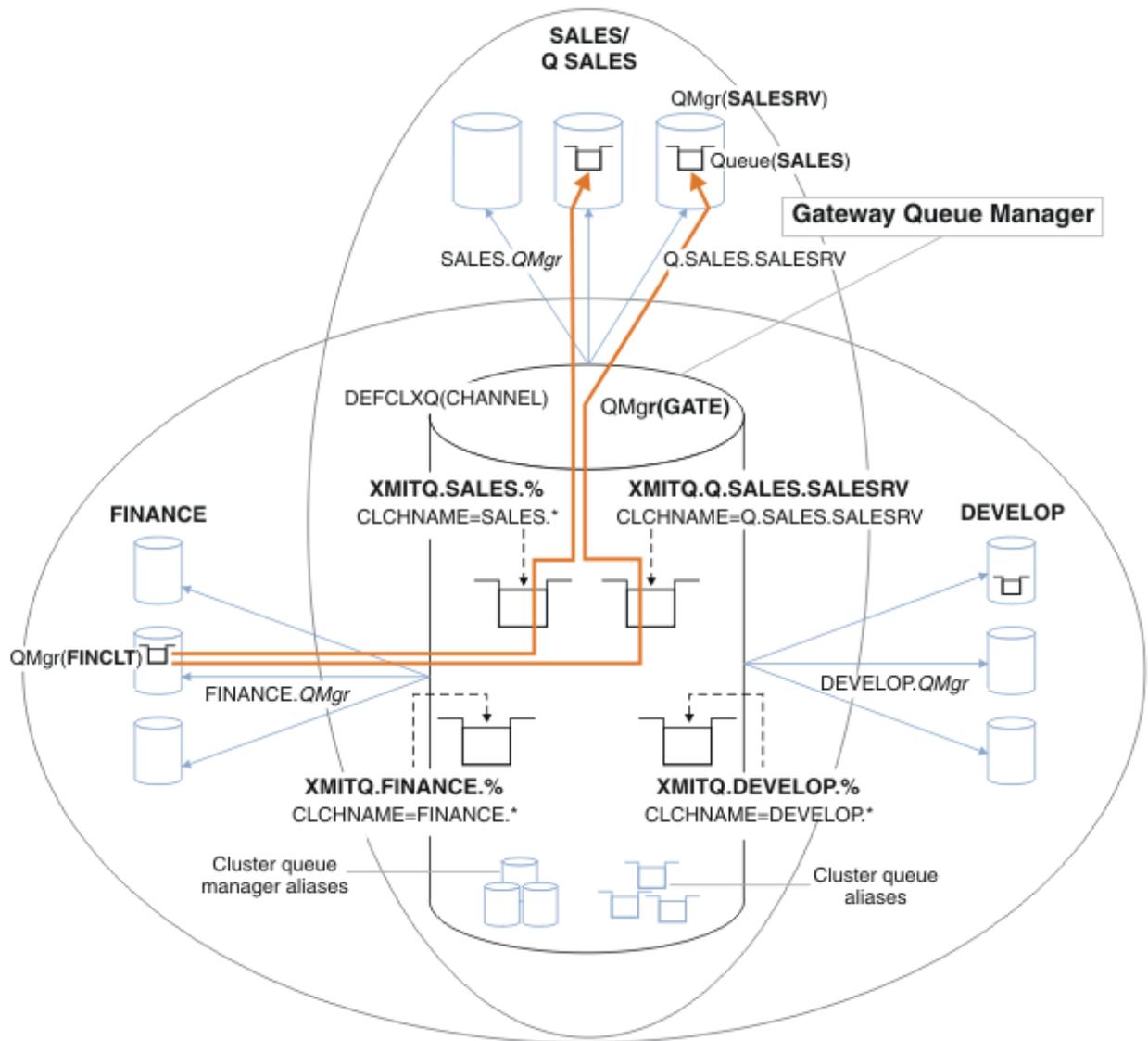


図 57. さまざまな部門別 IBM WebSphere MQ クラスタに固有の伝送キューの例

クラスタを変更する手順は以下のとおりです。定義については、新規クラスタで販売キューを分離し、ゲートウェイ・クラスタ伝送キューを分離するための変更を参照してください。

手順

1. 最初の構成ステップは、『" 使用するデフォルト・クラスタ伝送キューのタイプを選択します"』に対するものです。

決定内容は、個々のデフォルト・クラスタ伝送キューを作成することです。それには、GATE キュー・マネージャーで以下の **MQSC** コマンドを実行します。

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

手動でクラスタ伝送キューを定義することが目的なので、このデフォルトを選択する強力な理由はありません。この選択を診断する価値は低いものです。手動による定義が誤って行われ、メッセージがデフォルト・クラスタ伝送キューに流れる場合、それは、永続動的クラスタ伝送キューの作成という形で現れます。

2. 2 番目の構成ステップは、『" 他のフローとクラスタ伝送キューを共有させないメッセージ・フローを分離します"』です。

この例では、SALESRV 上のキュー SALES からメッセージを受信する販売アプリケーションを分離する必要があります。ゲートウェイ・キュー・マネージャーからのメッセージの分離のみが必要です。3つのサブステップは以下のとおりです。

- a) "フローの宛先を構成して、それぞれのフローのターゲット・キューが特定のクラスター内の当該キュー・マネージャー上の唯一のキューとなるようにします"。

この例では、キュー・マネージャー SALESRV を販売部門内の新しいクラスターに追加する必要があります。分離を必要とするキューがほとんどない場合、SALES キューに固有のクラスターを作成することもできます。クラスター名の命名規則としては、そのようなクラスターに Q. QueueName という名前を付けることが考えられます (例: Q.SALES)。多数のキューを分離しなければならない場合には、必要な場所と時機に応じて、分離されたキューのクラスターを作成するほうが、より実用的な代替手段となります。クラスター名は、QUEUES.n のようになります。

この例では、新しいクラスターを Q.SALES と呼びます。新規クラスターを追加するには、新規クラスター内の販売キューを分離し、ゲートウェイ・クラスター伝送キューを分離するための変更の定義を参照してください。定義の変更内容を以下に要約します。

- i) リポジトリ・キュー・マネージャーのクラスターの名前リストに、Q.SALES を追加します。この名前リストは、キュー・マネージャーの **REPOSNL** パラメーターで参照されます。
- ii) ゲートウェイ・キュー・マネージャーのクラスターの名前リストに、Q.SALES を追加します。この名前リストは、ゲートウェイ・キュー・マネージャーのすべてのクラスター・キュー別名定義およびクラスター・キュー・マネージャー別名定義で参照されます。
- iii) キュー・マネージャー SALESRV に、このキュー・マネージャーがメンバーとなるクラスター両方の名前リストを作成し、SALES キューのクラスター・メンバーシップを変更します。

```
DEFINE NAMLIST(CLUSTERS) NAMES(SALES, Q.SALES) REPLACE  
ALTER QLOCAL(SALES) CLUSTER(' ') CLUSNL(SALESRV.CLUSTERS)
```

SALES キューは、遷移の目的でのみ、両方のクラスターのメンバーになります。新しい構成が実行中になった後、SALES クラスターから SALES キューを削除します (302 ページの図 58 を参照)。

- b) "体系的な命名規則に従って、作成した新規クラスター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"。

- i) リポジトリ・キュー・マネージャーのそれぞれに、クラスター受信側チャンネル Q.SALES.RepositoryQMGr を追加します。
- ii) リポジトリ・キュー・マネージャーのそれぞれに、クラスター送信側チャンネル Q.SALES.OtherRepositoryQMGr を追加し、他方のリポジトリ・マネージャーに接続します。これらのチャンネルを開始します。
- iii) いずれか実行中のリポジトリ・キュー・マネージャーに、クラスター受信側チャンネル Q.SALES.SALESRV および Q.SALES.GATE を追加します。
- iv) SALESRV および GATE キュー・マネージャーに、クラスター送信側チャンネル Q.SALES.SALESRV および Q.SALES.GATE を追加します。クラスター送信側チャンネルを、クラスター受信側チャンネルを作成したりリポジトリ・キュー・マネージャーに接続します。

- c) "メッセージをターゲット・キューに送信するすべてのキュー・マネージャーに、分離された各宛先のクラスター伝送キューを定義します"。

ゲートウェイ・キュー・マネージャーに、Q.SALES.SALESRV クラスター送信側チャンネルのクラスター伝送キュー XMITQ.Q.SALES.SALESRV を定義します。

```
DEFINE QLOCAL(XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
```

- 3.3 番目の構成ステップは、『"ガバナンスまたはモニター要件を満たすクラスター伝送キューを作成します"』です。

ゲートウェイ・キュー・マネージャーに、クラスター伝送キューを定義します。

```
DEFINE QLOCAL(XMITQ.SALES) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
DEFINE QLOCAL(XMITQ.DEVELOP) USAGE(XMITQ) CLCHNAME(DEVELOP.*) REPLACE
DEFINE QLOCAL(XMITQ.FINANCE) USAGE(XMITQ) FINANCE(SALES.*) REPLACE
```

次のタスク

ゲートウェイ・キュー・マネージャーで新しい構成に切り替えます。

切り替えをトリガーするには、新規チャンネルを開始し、別の伝送キューに関連付けられるようになったチャンネルを再始動します。あるいは、ゲートウェイ・キュー・マネージャーを停止してから再始動するという方法もあります。

1. ゲートウェイ・キュー・マネージャーで以下のチャンネルを停止します。

```
SALES.Qmgr
DEVELOP.Qmgr
FINANCE.Qmgr
```

2. ゲートウェイ・キュー・マネージャーで以下のチャンネルを開始します。

```
SALES.Qmgr
DEVELOP.Qmgr
FINANCE.Qmgr
Q.SALES.SAVESRV
```

切り替えが完了したら、SALES クラスターから SALES キューを削除します。302 ページの図 58 を参照してください。

関連概念

[使用するクラスター伝送キュー・タイプの選択方法](#)

各種のクラスター伝送キュー構成オプションの中から選択する方法。

関連タスク

[クラスター化: クラスター伝送キューの切り替え](#)

既存の実働キュー・マネージャーのクラスター伝送キューに加えた変更を有効にしていく方法を計画します。

サンプル・クラスターの作成

サンプル・クラスターの定義、およびサンプル・クラスターを作成し、それを変更して SALES キューを分離し、ゲートウェイ・キュー・マネージャーでメッセージを分離する手順を説明します。

このタスクについて

FINANCE クラスター、SALES クラスター、および Q.SALES クラスターを作成するための完全な MQSC コマンドは、[基本クラスターの定義](#)、[新規クラスターで販売キューを分離し、ゲートウェイ・クラスター伝送キューを分離するための変更](#)、および [キュー・マネージャー SALESRV の販売キューを販売クラスターから削除する](#)に記載されています。定義を短縮するため、DEVELOP クラスターは定義から省略されています。

手順

1. SALES および FINANCE の各クラスター、およびゲートウェイ・キュー・マネージャーを作成します。

- a) キュー・マネージャーを作成します。

300 ページの表 27 のキュー・マネージャー名ごとに、コマンド `crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QmgrName` を実行します。

表 27. キュー・マネージャー名およびポート番号		
説明	キュー・マネージャー名	ポート番号
財務リポジトリ	FINR1	1414
財務リポジトリ	FINR2	1415
財務クライアント	FINCLT	1418
販売リポジトリ	SALER1	1416
販売リポジトリ	SALER2	1417
販売サーバー	SALESRV	1419
ゲートウェイ	GATE	1420

b) すべてのキュー・マネージャーを開始します。

300 ページの表 27 に記載されている各キュー・マネージャー名に対して、コマンド `stimqm QmgrName` を実行します。

c) 各キュー・マネージャーの定義を作成します。

コマンド `runmqsc QmgrName <filename>` を実行します。これらのファイルは 基本クラスターの定義 にリストされており、ファイル名はキュー・マネージャー名と一致します。

基本クラスターの定義

finr1.txt

```
DEFINE LISTENER(1414) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1414) REPLACE
START LISTENER(1414)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
```

finr2.txt

```
DEFINE LISTENER(1415) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1415) REPLACE
START LISTENER(1415)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
```

finclt.txt

```
DEFINE LISTENER(1418) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1418) REPLACE
START LISTENER(1418)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINCLT) CHLTYPE(CLUSRCVR) CONNAME('localhost(1418)')
CLUSTER(FINANCE) REPLACE
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

saler1.txt

```
DEFINE LISTENER(1416) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1416) REPLACE
START LISTENER(1416)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
```

saler2.txt

```
DEFINE LISTENER(1417) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1417) REPLACE
START LISTENER(1417)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
```

salesrv.txt

```
DEFINE LISTENER(1419) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1419) REPLACE
START LISTENER(1419)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(SALES) REPLACE
DEFINE QLOCAL(SALES) CLUSTER(SALES) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

gate.txt

```
DEFINE LISTENER(1420) TRPTYPE(TCP) IPADDR(LOCALHOST) CONTROL(QMGR) PORT(1420) REPLACE
START LISTENER(1420)
DEFINE NAMELIST(ALL) NAMES(SALES, FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(SALES) REPLACE
DEFINE QALIAS(A.SALES) CLUSNL(ALL) TARGET(SALES) TARGTYPE(Queue) DEFBIND(NOTFIXED)
REPLACE
DEFINE QREMOTE(FINCLT) RNAME(' ') RQMNAME(FINCLT) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(SALESRV) RNAME(' ') RQMNAME(SALESRV) CLUSNL(ALL) REPLACE
```

2. サンプル要求プログラムを実行して、構成をテストします。

- a) SALESRV キュー・マネージャーでトリガー・モニター・プログラムを開始します。

Windows で、コマンド・ウィンドウを開き、コマンド `runmqtrm -m SALESRV` を実行します。

- b) サンプル要求プログラムを実行し、要求を送信します。

Windows で、コマンド・ウィンドウを開き、コマンド `amqsreq A.SALES FINCLT` を実行します。

要求メッセージがエコー出力され、15 秒後にサンプル・プログラムが終了します。

3. ゲートウェイ・キュー・マネージャーで、Q.SALES クラスター内の SALES キューを分離して、SALES および FINANCE クラスターへのクラスター・メッセージを分離するための定義を作成します。

コマンド `runmqsc QmgrName <filename>` を実行します。これらのファイルは以下のリストにリストされており、ファイル名はキュー・マネージャー名とほぼ一致しています。

新規クラスター内で販売キューを分離し、ゲートウェイ・クラスター伝送キューを分離するための変更 **chgsaler1.txt**

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
```

chgsaler2.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
```

```
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
```

chgsalesrv.txt

```
DEFINE NAMELIST (CLUSTERS) NAMES(SALES, Q.SALES)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SAVESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(Q.SALES) REPLACE
ALTER QLOCAL (SALES) CLUSTER(' ') CLUSNL(CLUSTERS)
```

chggate.txt

```
ALTER NAMELIST(ALL) NAMES(SALES, FINANCE, Q.SALES)
ALTER QMGR DEFCLXQ(CHANNEL)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.GATE) CHLTYPE(CLUSRCVR) CONNAME('localhost(1420)')
CLUSTER(Q.SALES) REPLACE
DEFINE QLOCAL (XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
DEFINE QLOCAL (XMITQ.SALES) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
DEFINE QLOCAL (XMITQ.FINANCE) USAGE(XMITQ) CLCHNAME(FINANCE.*) REPLACE
```

4. SALES キューを SALES クラスターから削除します。

302 ページの [図 58](#) で **MQSC** コマンドを実行します。

```
ALTER QLOCAL(SALES) CLUSTER('Q.SALES') CLUSNL(' ')
```

[図 58](#). 販売クラスターからのキュー・マネージャー SALESRV 上の販売キューの削除

5. チャンネルを新しい伝送キューに切り替えます。

要件は、GATE キュー・マネージャーが使用しているすべてのチャンネルを停止してから開始することです。これを最も少ない数のコマンドで行うには、キュー・マネージャーを停止して、始動します。

```
endmqm -i GATE
strmqm GATE
```

次のタスク

1. サンプル要求プログラムを再実行して、新しい構成が機能することを検証します。ステップ [301 ページ](#) の『2』を参照してください。
2. GATE キュー・マネージャー上のすべてのクラスター伝送キューを介したメッセージ・フローをモニターします。

- a. クラスター伝送キューごとの定義を変更し、キュー・モニターを有効にします。

```
ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.
name) STATQ(ON)
```

- b. キュー・マネージャー統計モニターが OFF になっていることを確認し(出力を最小限にするため)、モニター間隔を低い値に設定します(複数のテストを効率的に行うため)。

```
ALTER QMGR STATINT(60) STATCHL(OFF) STATQ(OFF) STATMQI(OFF) STATACLS(OFF)
```

- c. GATE キュー・マネージャーを再始動します。
- d. サンプル要求プログラムを何回か実行し、SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV および SYSTEM.CLUSTER.TRANSMIT.QUEUE を経由するメッセージが同じ数であることを確認します。要求は SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV を経由し、応答は SYSTEM.CLUSTER.TRANSMIT.QUEUE を経由します。

```
amqsmon -m GATE -t statistics
```

- e. いくつかの間隔の結果は以下のとおりです。

```

C:\Documents and Settings\Admin>amqsmon -m GATE -t statistics
MonitoringType: QueueStatistics
QueueManager: 'GATE'
IntervalStartDate: '2012-02-27'
IntervalStartTime: '14.59.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.00.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
  QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
  CreateDate: '2012-02-24'
  CreateTime: '15.58.15'
  ...
  Put1Count: [0, 0]
  Put1FailCount: 0
  PutBytes: [435, 0]
  GetCount: [1, 0]
  GetBytes: [435, 0]
  ...
QueueStatistics: 1
  QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
  CreateDate: '2012-02-24'
  CreateTime: '16.37.43'
  ...
  PutCount: [1, 0]
  PutFailCount: 0
  Put1Count: [0, 0]
  Put1FailCount: 0
  PutBytes: [435, 0]
  GetCount: [1, 0]
  GetBytes: [435, 0]
  ...
MonitoringType: QueueStatistics
QueueManager: 'GATE'
IntervalStartDate: '2012-02-27'
IntervalStartTime: '15.00.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.01.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
  QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
  CreateDate: '2012-02-24'
  CreateTime: '15.58.15'
  ...
  PutCount: [2, 0]
  PutFailCount: 0
  Put1Count: [0, 0]
  Put1FailCount: 0
  PutBytes: [863, 0]
  GetCount: [2, 0]
  GetBytes: [863, 0]
  ...
QueueStatistics: 1
  QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
  CreateDate: '2012-02-24'

```

```
CreateTime: '16.37.43'  
...  
PutCount: [2, 0]  
PutFailCount: 0  
Put1Count: [0, 0]  
Put1FailCount: 0  
PutBytes: [863, 0]  
GetCount: [2, 0]  
GetBytes: [863, 0]  
...  
2 Records Processed.
```

最初の間隔では1つの要求および応答メッセージが送信され、2回目の間隔では2つ送信されました。要求メッセージは `SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV` に配置され、応答メッセージは `SYSTEM.CLUSTER.TRANSMIT.QUEUE` に配置されたと推測できます。

クラスター化: クラスター伝送キューの切り替え

既存の実働キュー・マネージャーのクラスター伝送キューに加えた変更を有効にしていく方法を計画します。

始める前に

切り替えプロセスによって新規伝送キューに転送する必要のあるメッセージの数を減らすと、切り替えはより迅速に完了します。さらに続行する前に、伝送キューを空にすることを試みるべき理由については、174 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](#)』を参照してください。

このタスクについて

クラスター伝送キューに加える変更を有効にする方法には、次の2つの選択肢があります。

1. キュー・マネージャーが自動で変更するようにする。これがデフォルトです。次回クラスター送信側チャンネルが開始するときに、キュー・マネージャーはクラスター送信側チャンネルを、保留中の伝送キューの変更内容に切り替えます。
2. 手動で変更する。クラスター送信側チャンネルが停止しているときに、それに対する変更を行うことができます。クラスター伝送キューを別のものに切り替えてから、クラスター送信側チャンネルを開始します。

これら2つのオプションのどちらを選ぶか、また切り替えをどのように行うかを決定するに当たって、どのような要素を考慮すればよいのでしょうか。

手順

- オプション 1: キュー・マネージャーが自動で変更するようにする。306 ページの『[アクティブなクラスター送信側チャンネルを別のクラスター送信側キューのセットに切り替える](#)』を参照。

キュー・マネージャー側で自動に切り替わるようにするには、このオプションを選択します。

言い換えれば、このオプションの場合、ユーザーが強制的にチャンネルを停止しなくても、キュー・マネージャーはクラスター送信側チャンネルを切り替えます。早く切り替わるように、チャンネルを強制的に停止してから開始することもできます。このスイッチは、チャンネルの開始時に開始され、チャンネルの実行中に実行されます。これはオプション 2 とは異なります。オプション 2 では、チャンネルが停止しているときにスイッチが実行されます。

このオプションを選択して、切り替えが自動的に行われるようにすると、クラスター送信側チャンネルが開始したときに、切り替えプロセスが開始します。チャンネルが停止していない場合は、処理するメッセージがあれば、チャンネルが活動状態でなくなってから、切り替えプロセスが開始します。チャンネルが停止している場合は、`START CHANNEL` コマンドでチャンネルを開始します。

切り替えプロセスは、チャンネルがサービスを提供する伝送キューにクラスター送信側チャンネルのメッセージがなくなるとすぐに完了します。この状態になると、それ以後、クラスター送信側チャンネル

の新着メッセージは新しい伝送キューに直接格納されます。それまでは、メッセージは古い伝送キューに格納され、切り替えプロセスがメッセージを古い伝送キューから新しい伝送キューに転送します。クラスター送信側チャンネルは、切り替えプロセスの間中、メッセージを新しいクラスター伝送キューから転送します。

切り替えプロセスがいつ完了するかは、システムの状態によって左右されます。保守時間枠で変更を行う場合は、切り替えプロセスが時間内で完了するかどうかを前もって査定しておいてください。時間内に完了するかどうかは、古い伝送キューからの転送を待機しているメッセージの数がゼロになるかどうかによって決まります。

1 番目の方法の利点は、自動であることです。欠点としては、構成変更を行えるのが保守時間枠に限られている場合、保守時間枠内で切り替えプロセスを完了するようにシステムを制御できるという確信がなければならぬということです。その確信がない場合は、オプション 2 をお勧めします。

- オプション 2: 手動で変更する。307 ページの『[停止したクラスター送信側チャンネルから別のクラスター伝送キューへの切り替え](#)』を参照。

切り替えプロセス全体を手動で制御する場合や、停止したまたは活動状態にないチャンネルを切り替える場合は、このオプションを選択します。いくつかのクラスター送信側チャンネルを切り替える場合、その切り替えを保守時間枠に行うのであれば、このオプションが適しています。

言い換えれば、このオプションの場合、クラスター送信側チャンネルが停止している間に、自分でクラスター送信側チャンネルを切り替えます。

このオプションを選択すると、いつ切り替えるかをすべて自分で決めることができます。

保守時間枠内の一定の時間内に切り替えプロセスが完了することを確信できます。切り替えにかかる時間は、一方の伝送キューから他方の伝送キューに転送しなければならないメッセージの数によって左右されます。継続的にメッセージが着信する場合は、プロセスがすべてのメッセージを転送するのに時間がかかる可能性があります。

古い伝送キューからメッセージを転送せずにチャンネルを切り替えるというオプションもあります。

スイッチは"一瞬"にして終わります。

クラスター送信側チャンネルを再始動すると、新たに割り当てられた伝送キュー上のメッセージの処理が開始します。

2 番目の方法の利点は、切り替えプロセス全体をコントロールできることです。欠点は、切り替えるクラスター送信側チャンネルを識別し、必要なコマンドを実行し、クラスター送信側チャンネルを停止できなくさせかねない未確定チャンネルがあればそれを解決しなければならないことです。

関連概念

[使用するクラスター伝送キュー・タイプの選択方法](#)

各種のクラスター伝送キュー構成オプションの中から選択する方法。

関連タスク

[クラスター化: 複数のクラスター伝送キューの構成例](#)

このタスクでは、複数のクラスター伝送キューの計画手順を、3 つのオーバーラップするクラスターに適用します。ここで要件となるのは、1 つのクラスター・キューへのメッセージ・フローを、その他すべてのメッセージ・フローから分離すること、そして各クラスターへのメッセージをそれぞれ異なるクラスター伝送キューに保管することです。

306 ページの『[アクティブなクラスター送信側チャンネルを別のクラスター送信側キューのセットに切り替える](#)』

この作業には、アクティブなクラスター送信側チャンネルを切り替えるための 3 つのオプションがあります。1 つのオプションは、キュー・マネージャーが自動で切り替えるようにする方法で、アプリケーションの実行への影響はありません。その他のオプションは、チャンネルを手動で停止および開始する、またはキュー・マネージャーを再始動するという方法です。

307 ページの『[停止したクラスター送信側チャンネルから別のクラスター伝送キューへの切り替え](#)』

関連情報

174 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](#)』

アクティブなクラスター送信側チャンネルを別のクラスター送信側キューのセットに切り替える
この作業には、アクティブなクラスター送信側チャンネルを切り替えるための3つのオプションがあります。
1つのオプションは、キュー・マネージャーが自動で切り替えるようにする方法で、アプリケーションの実行への影響はありません。その他のオプションは、チャンネルを手動で停止および開始する、またはキュー・マネージャーを再始動するという方法です。

始める前に

クラスター伝送キューの構成を変更します。DEFCLXQ キュー・マネージャー属性を変更するか、または伝送キューのCLCHNAME 属性を追加または変更することができます。

切り替えプロセスによって新規伝送キューに転送する必要のあるメッセージの数を減らすと、切り替えはより迅速に完了します。さらに続行する前に、伝送キューを空にすることを試みるべき理由については、174 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](#)』を参照してください。

このタスクについて

この作業のステップを見本として使用し、クラスター伝送キュー構成変更を実行する独自の計画を立ててください。

手順

1. オプション: 現在のチャンネル状況を記録します。

クラスター伝送キューにサービスを提供している現在のチャンネルおよび保存済みのチャンネルの状況を記録します。次のコマンドを実行すると、システム・クラスター伝送キューに関連する状況が表示されます。自分で定義したクラスター伝送キューに関連する状況を表示するために、独自のコマンドを追加できます。XMITQ.ChannelName などの規則を使用して、定義するクラスター伝送キューに名前を付け、それらの伝送キューのチャンネル状況を簡単に表示できるようにします。

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

2. 伝送キューを切り替えます。

- 何もしない。クラスター送信側チャンネルが停止または非アクティブ状態から再始動すると、キュー・マネージャーはクラスター送信側チャンネルを切り替えます。

キュー・マネージャー構成の変更に関して規則や考慮事項がない場合は、このオプションを選択します。実行中のアプリケーションは、この変更の影響を受けません。

- キュー・マネージャーを再始動する。要求に応じて、すべてのクラスター送信側チャンネルは自動的に停止して再始動します。

すべての変更の適用を即時に開始するには、このオプションを選択します。実行中のアプリケーションは、キュー・マネージャーがシャットダウンして再始動する際に、そのキュー・マネージャーによって中断されます。

- 個々のクラスター送信側チャンネルを停止して再始動します。

いくつかのチャンネルを即時に変更するには、このオプションを選択します。メッセージ・チャンネルを一度停止して再び開始するまでの間に、実行中のアプリケーションにメッセージ転送の短い遅延が生じます。クラスター送信側チャンネルは、停止させた時間を除いて、実行を継続します。切り替えプロセス中、メッセージは古い伝送キューに送達され、切り替えプロセスによって新しい伝送キューに転送され、クラスター送信側チャンネルによって新しい伝送キューから転送されます。

3. オプション: 切り替え時のチャンネルをモニターします。

切り替え中のチャンネル状況と伝送キュー・サイズを表示します。次の例では、システム・クラスター伝送キューの状況が表示されます。

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

- オプション: 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 "AMQ7341 チャンネル *ChannelName* の伝送キューがキュー *QueueName* から *QueueName* に切り替えられました。" をモニターします。

停止したクラスター送信側チャンネルから別のクラスター伝送キューへの切り替え

始める前に

何らかの構成変更を行った後に、影響を受けるクラスター送信側チャンネルを始動せずに、変更を有効にすることが必要な場合があります。あるいは、作業のステップの1つとして、必要な構成変更を行うこともできます。

切り替えプロセスによって新規伝送キューに転送する必要のあるメッセージの数を減らすと、切り替えはより迅速に完了します。さらに続行する前に、伝送キューを空にすることを試みるべき理由については、174 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](#)』を参照してください。

このタスクについて

この作業では、停止または非アクティブのクラスター送信側チャンネルによって扱われる伝送キューを切り替えます。クラスター送信側チャンネルが停止しており、その伝送キューを即時に切り替えることが必要な場合に、この作業を行います。例えば、何らかの理由でクラスター送信側チャンネルが開始されていない場合や、構成に関連した他の問題がある場合が該当します。問題を解決するには、クラスター送信側チャンネルを作成し、古いクラスター送信側チャンネルの伝送キューを、新しく定義したクラスター送信側チャンネルと関連付けるようにします。

さらによくある状況は、クラスター伝送キューの再構成を実行するタイミングを制御することが必要となるケースです。再構成を完全に制御するには、チャンネルを停止し、構成を変更してから、伝送キューを切り替えます。

手順

- 切り替えるチャンネルを停止します。
 - 切り替える対象で実行中のチャンネルや非アクティブなチャンネルがあれば、すべて停止します。非アクティブ・クラスター送信側チャンネルを停止すると、構成の変更中にこのチャンネルが開始しなくなります。

```
STOP CHANNEL(ChannelName) MODE(QUIESCSE) STATUS(STOPPED)
```

- オプション: 構成を変更します。

例については、296 ページの『[クラスター化: 複数のクラスター伝送キューの構成例](#)』を参照してください。

- クラスター送信側チャンネルを新しいクラスター伝送キューに切り替えます。

```
runswchl -m QmgrName -c ChannelName
```

runswchl コマンドを実行すると、古い伝送キュー上のすべてのメッセージが新しい伝送キューに転送されます。このチャンネルの古い伝送キュー上にあるメッセージの数がゼロになると、切り替えが完了します。コマンドは同期的に実行されます。コマンドは、切り替えプロセス中、進行状況のメッセージをウィンドウに表示します。

転送フェーズ中は、クラスター送信側チャンネルに送信された既存および新規メッセージが順番に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停止しているため、メッセージは新しい伝送キューに蓄積されます。停止しているクラスター送信側チャンネルを、306 ページの『[アクティブなクラスター送信側チャンネルを別の](#)』

[クラスター送信側キューのセットに切り替える](#)のステップ 306 ページの『2』と対比してください。そのステップでは、クラスター送信側チャンネルが実行中であるため、必ずしもメッセージが新しい伝送キューに蓄積されるとは限りません。

4. オプション: 切り替え時のチャンネルをモニターします。

別のコマンド・ウィンドウに、切り替え中の伝送キュー・サイズを表示します。次の例では、システム・クラスター伝送キューの状況が表示されます。

```
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

5. オプション: 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 "AMQ7341 チャンネル *ChannelName* の伝送キューがキュー *QueueName* から *QueueName* に切り替えられました。" をモニターします。
6. 停止したクラスター送信側チャンネルを再始動します。
チャンネルを停止し、STOPPED 状況にしたため、それらは自動では始動しません。

```
START CHANNEL(ChannelName)
```

関連資料

[runswchl](#)

[RESOLVE CHANNEL](#)

[STOP CHANNEL](#)

クラスター化: マイグレーションと変更に関するベスト・プラクティス

このトピックでは、IBM WebSphere MQ クラスターを計画および管理するための指針について記載しています。この情報は、テストおよびお客様からのフィードバックに基づく指針を示すものです。

1. [308 ページの『クラスター内でのオブジェクトの移動』](#) (IBM WebSphere MQ のフィックスパックや新バージョンはインストールせずに、クラスター内でオブジェクトを移動する場合のベスト・プラクティスです)。
2. [310 ページの『インストールのアップグレードと保守』](#) (作業中のクラスター・アーキテクチャーを稼働させたまま、保守またはアップグレードの適用および新規アーキテクチャーのテストを行う場合のベスト・プラクティスです)。

クラスター内でのオブジェクトの移動

アプリケーションとそのキュー

あるキュー・マネージャーでホストされているキュー・インスタンスを別のキュー・マネージャーに移動して、そこでホストされるように必要がある場合は、ワークロード・บาลancingのパラメーターを操作して、スムーズな移行を行うことができます。

キューのインスタンスを、それが新たにホストされる場所に作成します。ただし、アプリケーションの切り替え準備が完了するまで、クラスター・ワークロード・บาลancing設定を使用して、引き続き、元のインスタンスにメッセージが送信されるようにしてください。そのためには、以下の手順を実行します。

1. 既存のキューの **CLWLRANK** プロパティを高い値 (例えば、5) に設定します。
2. キューの新規インスタンスを作成し、その **CLWLRANK** プロパティを 0 に設定します。
3. 新規システムのためのその他の構成作業 (キューの新規インスタンスに対するコンシューム側アプリケーションのデプロイおよび始動など) を完了させます。
4. 新規キュー・インスタンスの **CLWLRANK** プロパティを元のインスタンスよりも高い値 (例えば、9) に設定します。
5. システムでキューに入っているすべてのメッセージを元のキュー・インスタンスが処理できるようにしてから、そのキューを削除します。

キュー・マネージャー全体の移動

キュー・マネージャーが同じホストに留まっても、その IP アドレスを変更する場合、プロセスは以下ようになります。

- DNS は、正しく使用されれば、このプロセスを簡単にするのに役立ちます。「接続名 (CONNAME)」チャンネル属性を設定して DNS を使用する方法については、[ALTER CHANNEL](#) を参照してください。
- 完全リポジトリを移動する場合は、変更を加える前に、順調に (例えば、チャンネル状況に関する問題なく) 稼働している完全リポジトリが少なくとも 1 つは他にあることを確認します。
- トラフィックが蓄積していくのを避けるために、[SUSPEND QMGR](#) コマンドを使用してキュー・マネージャーを中断します。
- コンピューターの IP アドレスを変更します。CLUSRCVR チャンネル定義の CONNAME フィールドで IP アドレスを使用する場合、この IP アドレスの項目を変更します。更新内容がどこでも有効になるようにするため、DNS キャッシュをフラッシュすることが必要な場合があります。
- キュー・マネージャーが完全リポジトリに再接続すると、チャンネルの自動定義が自動的にそれ自体を解決します。
- キュー・マネージャーで完全リポジトリをホストしており、IP アドレスを変更する場合は、新しい場所に対して手動で定義されているあらゆる CLUSSDR チャンネルをポイントするように、部分リポジトリが可能な限り速やかに切り替えられるようにすることが重要です。この切り替えが実施されるまで、それらのキュー・マネージャーは残りの (変更されていない) 完全リポジトリとは通信できるかもしれませんが、チャンネル定義が不正であるという警告メッセージが表示されることがあります。
- [RESUME QMGR](#) コマンドを使用してキュー・マネージャーを再開します。

キュー・マネージャーを新しいホストに移動する必要がある場合は、キュー・マネージャー・データをコピーして、バックアップから復元することが可能です。ただし、他に選択肢がない場合を除き、そのようなプロセスは推奨されません。前のセクションで説明しているように、新しいマシンでキュー・マネージャーを作成し、キューおよびアプリケーションを複製することをお勧めします。そのようにすることで、円滑なロールオーバーおよびロールバック・メカニズムが実現します。

バックアップを使用してキュー・マネージャー全体を移動する場合は、以下のベスト・プラクティスに従ってください。

- バックアップからキュー・マネージャーを復元するものとしてプロセス全体を処理し、オペレーティング・システム環境に応じてシステム・リカバリーで通常使用するあらゆるプロセスを適用してください。
- マイグレーション後に [REFRESH CLUSTER](#) コマンドを使用して、ローカルに保持されているすべてのクラスター情報 (未確定の自動定義チャンネルを含む) を破棄し、強制的に再作成します。

注: 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。[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。](#)

キュー・マネージャーを作成し、クラスター内の既存のキュー・マネージャーからセットアップを複製するときには (このトピックの前述の説明を参照)、決して 2 つの異なるキュー・マネージャーを実際には同じものとして扱わないでください。特に、新しいキュー・マネージャーに同じキュー・マネージャー名と IP アドレスを指定しないようにしてください。代替キュー・マネージャーに「ドロップイン」しようとする、IBM WebSphere MQ クラスターで問題が発生する原因となることがよくあります。キャッシュは、**QMID** 属性を含む更新情報を受信するものと予期しているため、破損状態になる可能性があります。

誤って 2 つの異なるキュー・マネージャーを同じ名前で作成してしまった場合には、[RESET CLUSTER QMID](#) コマンドを使用して、誤ったエントリーをクラスターから排除することをお勧めします。

インストールのアップグレードと保守

「ビッグ・バン・シナリオ」(例えば、クラスターとキュー・マネージャー・アクティビティをすべて停止し、すべてのアップグレードと保守を全部のキュー・マネージャーに適用してから、すべてを同時に開始するなど)は避けてください。クラスターは複数のバージョンのキュー・マネージャーが共存している場合でも動作するよう設計されているので、十分な計画を立てたうえで段階的な保守を行うことをお勧めします。

以下に示しているバックアップ・プランを立ててください。

- z/OS の場合、後方マイグレーション PTF を適用しましたか
- バックアップを取りましたか
- すぐに新しいクラスター機能を使用するのは避けてください。すべてのキュー・マネージャーが新しいレベルにアップグレードされたことを確認し、それらをいずれもロールバックしないことを確認するまでは、使用を待ってください。一部のキュー・マネージャーがまだ初期レベルにあるクラスターで新しいクラスター機能を使用すると、動作が未定義になる可能性があります。例えば、IBM WebSphere MQ Version 6.0 から IBM WebSphere MQ Version 7.1 への移行において、キュー・マネージャーがクラスター・トピックを定義している場合、IBM WebSphere MQ Version 6.0 キュー・マネージャーはその定義を理解できず、このトピックにパブリッシュすることもできません。

最初に完全リポジトリをマイグレーションしてください。完全リポジトリは、自らが認識できない情報を転送できますが、そのような状態を維持することはできません。したがって、絶対必要な場合を除き、そのようなことはお勧めできません。詳しくは、[キュー・マネージャー・クラスターの移行](#)を参照してください。

関連概念

310 ページの『[クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス](#)』

REFRESH CLUSTER コマンドを使用して、クラスターに関するローカルに保持されているすべての情報を破棄し、クラスターの完全リポジトリからその情報を再作成します。例外的な状況を除き、このコマンドを使用する必要はありません。このコマンドを使用する必要がある場合は、使用方法に関する特別な考慮事項があります。この情報は、テストおよびお客様からのフィードバックに基づく指針を示すものです。

クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス

REFRESH CLUSTER コマンドを使用して、クラスターに関するローカルに保持されているすべての情報を破棄し、クラスターの完全リポジトリからその情報を再作成します。例外的な状況を除き、このコマンドを使用する必要はありません。このコマンドを使用する必要がある場合は、使用方法に関する特別な考慮事項があります。この情報は、テストおよびお客様からのフィードバックに基づく指針を示すものです。

REFRESH CLUSTER は本当に実行する必要がある場合にのみ実行する

IBM WebSphere MQ クラスター・テクノロジーを使用することにより、クラスター構成に対するあらゆる変更(クラスター・キューへの変更など)が、その情報を認識する必要があるクラスターのすべてのメンバーに自動的に認識されるようになります。このような情報伝達を実現するためにさらに管理手順を実行する必要はありません。

そのような情報を必要とするクラスター内のキュー・マネージャーに情報が届かない場合(あるクラスター・キューをアプリケーションが初めて開こうとしたときにそのキューがクラスター内の他のキュー・マネージャーに認識されない場合など)、クラスター・インフラストラクチャーに問題があることを意味します。例えば、キュー・マネージャーと完全リポジトリ・キュー・マネージャーの間のチャンネルが開始されないということがあり得ます。そのため、不整合が認められる場合は調査する必要があります。可能であれば、**REFRESH CLUSTER** コマンドを使用しないでこの状況を解決してください。

この製品資料の他の場所に記載されている稀な状況、または IBM サポートから要求された場合は、**REFRESH CLUSTER** コマンドを使用して、クラスターに関するローカルに保持されているすべての情報を破棄し、クラスター内の完全リポジトリからその情報を再作成することができます。

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

REFRESH CLUSTER コマンドを使用すると、処理中のクラスターに悪影響が生じる可能性があります。例えば、完全リポジトリがキュー・マネージャー・クラスター・リソースの再伝搬を処理する際に、完全リポジトリの作業が突然増加することがあります。大規模なクラスター (数百のキュー・マネージャーが存在するクラスター) をリフレッシュする場合、このコマンドは可能な限り日常の作業では使用せず、別の方法で個々の不整合を修正してください。例えば、クラスター・キューがクラスター全体で正しく伝搬されていない場合、クラスター・キュー定義の更新 (定義の記述の変更など) を調査するために最初に適用する手法によって、キュー構成がクラスター全体に再伝搬されます。このプロセスは、問題を特定するのに役立ちます。また、一時的な不整合を解消するのに役立つこともあります。

代替方法を使用できず、大規模クラスターで **REFRESH CLUSTER** を実行する必要がある場合は、ユーザー・ワークロードへの影響を回避するために、オフピーク時または保守期間中に実行する必要があります。また、単一のバッチで大規模なクラスターをリフレッシュするのではなく、[311 ページの『クラスター・オブジェクトが自動更新を送信するときにパフォーマンスおよび可用性の問題を回避する』](#) で説明されているように、時間をずらしてアクティビティーを実行してください。

クラスター・オブジェクトが自動更新を送信するときにパフォーマンスおよび可用性の問題を回避する

新規クラスター・オブジェクトがキュー・マネージャーで定義されると、定義された時刻から 27 日ごとにこのオブジェクトの更新が生成され、クラスター内のすべての完全リポジトリおよびその他関係するキュー・マネージャーに送信されます。**REFRESH CLUSTER** コマンドをキュー・マネージャーに発行すると、指定したクラスターのローカルで定義されているすべてのオブジェクトでこの自動更新のクロックが再設定されます。

単一のバッチで大規模なクラスター (数百のキュー・マネージャーが存在するクラスター) をリフレッシュする場合、または構成バックアップからシステムを再作成するなどその他の状況では、それらすべてのキュー・マネージャーは、27 日後にすべてのオブジェクト定義を完全リポジトリに同時に再通知します。この動作によってもシステムの実行速度が著しく低下する場合があります。すべての更新が完了するまで利用不可になる可能性さえあります。そのため、大規模なクラスターで複数のキュー・マネージャーをリフレッシュするまたは再作成する必要がある場合は、数時間または数日を空けてこのアクティビティーを実行し、後続の自動更新がシステム・パフォーマンスに定期的に影響を与えないようにしてください。

システム・クラスター履歴キュー

REFRESH CLUSTER が実行されると、キュー・マネージャーは、クラスターの状態をリフレッシュして **SYSTEM.CLUSTER.HISTORY.QUEUE (SCHQ)** に保存する前に、そのスナップショットをとりま (キュー・マネージャーで定義されている場合)。このスナップショットは、後でシステムの問題が発生した場合に備えて、IBM サービスに使用することのみを目的として作成されます。デフォルトでは SCHQ は、始動時に分散キュー・マネージャーに対して定義されます。z/OS マイグレーションの場合は、SCHQ を手動で定義する必要があります。SCHQ 上のメッセージの有効期限は 3 カ月です。

関連概念

[REFRESH CLUSTER の実行中に発生するアプリケーションの問題](#)

[パブリッシュ/サブスクライブ・クラスターの REFRESH CLUSTER についての考慮事項](#)

関連資料

[MQSC コマンドのリファレンス: REFRESH CLUSTER](#)

クラスター化: 可用性、複数インスタンス、および災害復旧

このトピックでは、IBM WebSphere MQ クラスターを計画および管理するための指針について記載しています。この情報は、テストおよびお客様からのフィードバックに基づく指針を示すものです。

IBM WebSphere MQ クラスター化自体は高可用性ソリューションではありませんが、状況によっては、IBM WebSphere MQ を使用してサービスの可用性を向上させることができます。例えば、異なるキュー・マネージャーに複数のキュー・インスタンスを作成するというような方法をとります。このセクションでは、

IBM WebSphere MQ インフラストラクチャーをそのようなアーキテクチャーで使用できるように、インフラストラクチャーで最大限の可用性が実現されるようにするための指針を提供します。

クラスター・リソースの可用性

2つの完全リポジトリを保持することが推奨される一般的な理由は、1つが失われてもクラスターの円滑な稼働に重大な影響が出ないことです。両方が使用不可になった場合でも、部分リポジトリに既存のナレッジが保持されるため、60日の猶予期間が与えられます。ただし、このイベントでは、新規リソースまたは以前にアクセスされたリソース(キューなど)は使用できません。

クラスターによるアプリケーションの可用性の向上

クラスターは、キューとアプリケーションの複数のインスタンスを使用することにより、高い可用性を備えたアプリケーション(要求/応答タイプのサーバー・アプリケーションなど)の設計に役立ちます。必要な場合は、例えばキュー・マネージャーまたはチャンネルが使用不可になっているというようなことがない限り、優先順位属性により、稼働中のアプリケーションが優先されるようにすることができます。これは、問題が発生した場合に速やかに切り替えが行われ、引き続き新しいメッセージが処理されるようにするために極めて有効です。

ただし、クラスター内の特定のキュー・マネージャーに配信されたメッセージは、当該キュー・インスタンスでのみ保持されるため、そのキュー・マネージャーが回復するまで処理できません。このため、本当の意味で高いデータ可用性を実現するには、複数インスタンス・キュー・マネージャーなど、他のテクノロジーの利用を検討することもできます。

複数インスタンス・キュー・マネージャー

ソフトウェアの高可用性(複数インスタンス)は、既存のメッセージを常に使用可能にしておくための最も優れた組み込みオフリングです。詳しくは、[321 ページの『高可用性構成と共に WebSphere MQ を使用する』](#)、[350 ページの『複数インスタンス・キュー・マネージャーの作成』](#)、および次のセクションを参照してください。この手法を利用してクラスター内の任意のキュー・マネージャーで高い可用性を実現することができます。ただし、クラスター内のすべてのキュー・マネージャーが、IBM WebSphere MQ Version 7.0.1 以降を使用して実行されていることが前提になります。クラスター内のいずれかのキュー・マネージャーが以前のレベルである場合、そのキュー・マネージャーは、セカンダリー IP へのフェイルオーバーを行うと、複数インスタンス・キュー・マネージャーへの接続を失う可能性があります。

このトピックで既に説明したように、2つの完全リポジトリを構成している限り、ほとんどの場合は、本来、可用性が高くなります。必要に応じて、IBM WebSphere MQ ソフトウェア高可用性/複数インスタンス・キュー・マネージャーを完全リポジトリに使用することができます。これらの方法を使用することを強く勧める根拠はありません。むしろ、一時的な障害の場合、これらの方法を使用すると、フェイルオーバーの際に余分にパフォーマンス・コストが発生することもあります。2つの完全リポジトリを稼働させる代わりにソフトウェア高可用性を使用することはお勧めしません。なぜならば、例えば、チャンネルが1つだけ停止した場合に、フェイルオーバーは必要ないかもしれませんが、部分リポジトリがクラスター・リソースを照会できないままになる可能性があります。

災害時回復の場合

災害復旧(キュー・マネージャーのデータが保管されているディスクが破損した場合の復旧など)にきちんと対処するのは難しいものです。IBM WebSphere MQ は役立ちますが、自動的に災害復旧できるわけではありません。IBM WebSphere MQ (オペレーティング・システムおよび他の基本的複製テクノロジーは除く)における「真の」災害復旧オプションは、バックアップからの回復だけです。そのような状態におけるクラスター固有の考慮事項として、以下のようなものがあります。

- 災害復旧シナリオのテスト時には注意してください。例えば、バックアップ・キュー・マネージャーの動作をテストする場合に、同じネットワーク内でそれらのキュー・マネージャーをオンラインにするときには注意が必要です。稼働中のクラスター・キュー・マネージャーのキューと同じ名前が付けられているキューがホストされることにより、誤ってバックアップ・キュー・マネージャーが稼働中のクラスターに加わり、メッセージを「盗み」始める可能性があります。
- 災害復旧のテストは、稼働中のクラスターに干渉してはなりません。干渉を避けるための手法を以下に示します。
 - 完全なネットワーク分離、またはファイアウォール・レベルでのネットワーク分離。
 - 実際の災害復旧シナリオが実施されるまで、または実施されない限り、災害復旧システムには稼働中の SSL 証明書を発行しない。

- 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、バックアップがそのクラスターの残りの部分と同期していない可能性もあります。 **REFRESH CLUSTER** コマンドを使用すると、更新を解決してクラスターと同期することができます。ただし、 **REFRESH CLUSTER** コマンドは最後の手段として使用してください。 310 ページの『クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス』を参照してください。組織内の手続き関連の資料および IBM WebSphere MQ の資料を読み返して、最後の手段としてこのコマンドを使用する前に、簡単な手段を見逃していないかどうか確認してください。
- どのようなリカバリーにおいても、アプリケーションはデータの再生と損失に対処する必要があります。キューをクリアして既知の状態にするかどうか、あるいは再生を管理するのに十分な情報が他の場所にあるかどうかを判断する必要があります。

クラスター化: モニター

このトピックでは、IBM WebSphere MQ クラスターを計画および管理するための指針について記載しています。この情報は、テストおよびお客様からのフィードバックに基づく指針を示すものです。

クラスター内のアプリケーション・メッセージのモニター

通常、キュー・マネージャーから送り出されるすべてのクラスター・メッセージは、どのクラスター送信側チャンネルを使用して送信されるかに関係なく、**SYSTEM.CLUSTER.TRANSMIT.QUEUE** を通過します。各チャンネルは、そのチャンネルを宛先とするメッセージを、他のすべてのクラスター送信側チャンネルと並行して、排出します。このキューに蓄積されるメッセージが徐々に増える場合は、1つまたは複数のチャンネルで問題が発生している可能性があるため、以下のことを調べる必要があります。

- クラスターを設計するためには、キューの深さを適切にモニターする必要があります。
- 次のコマンドを実行すると、伝送キューで複数のメッセージが待機しているすべてのチャンネルが返されません。

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

すべてのクラスター・メッセージが単一のキューに入っている場合は、キューがいっぱいになり始めたときに、どのチャンネルに問題があるのかを確認するのが困難である可能性があります。このコマンドを使用すると、どのチャンネルが原因なのかを簡単に確認することができます。

複数の送信キューを使用するように、クラスター・キュー・マネージャーを構成できます。キュー・マネージャー属性 **DEFCLXQ** を **CHANNEL** に変更すると、すべてのクラスター送信側チャンネルが異なるクラスター伝送キューに関連付けられます。また、送信キューを個別に手動で構成することもできます。クラスター送信側チャンネルに関連付けられているすべてのクラスター伝送キューを表示するには、次のコマンドを実行します。

```
DISPLAY CLUSQMGR (qmgrName) XMITQ
```

キュー名の左側に特定の語幹を付けるというパターンに従うように、クラスター送信キューを定義します。次に、汎用キュー名を使用して、**DISPLAY CLUSMGR** コマンドによって返されたすべてのクラスター伝送キューの深さを照会できます。

```
DISPLAY QUEUE (qname*) CURDEPTH
```

クラスター内の制御メッセージのモニター

SYSTEM.CLUSTER.COMMAND.QUEUE キューは、キュー・マネージャーのすべてのクラスター制御メッセージ（ローカル・キュー・マネージャーによって生成されるものと、クラスター内の他のキュー・マネージャーからこのキュー・マネージャーに送信されるもののいずれか）を処理するために使用されます。キュー・マネージャーが自らのクラスター状態を正しく維持している場合、このキューは 0 に近づく傾向にあります。このキューに蓄積されるメッセージの数が一時的に増加することがありますが、以下のように考えられます。

- キューに多くのメッセージがあるということは、クラスターの状態が激しく変動していることを示しています。

- 重大な変更を行う場合は、次の変更までの間にキューの状態を落ち着かせます。例えば、リポジトリを移動する場合は、第2のリポジトリを移動する前に、キューが0になるようにします。

このキューにメッセージのバックログがある間は、クラスター状態の更新およびクラスター関連のコマンドは処理されません。メッセージが長い間このキューから削除されない場合は、さらなる調査が必要です。最初に、キュー・マネージャーのエラー・ログを調べてください。この状態を引き起こしているプロセスが明らかになる場合があります。

SYSTEM.CLUSTER.REPOSITORY.QUEUE では、クラスター・リポジトリのキャッシュの情報が多数のメッセージとして保持されます。このキューにメッセージが常に存在するのはよくあることであり、大規模なクラスターでは、そのようなことがより頻繁に起こります。したがって、このキューに蓄積されているメッセージの数は懸念すべき問題ではありません。

ログのモニター

クラスター内で発生する問題は、情報がキャッシュされること、およびクラスター化の分散されているという性質により、発生時から数日間(あるいは、数カ月間も)経っても、アプリケーションで表立った症状として現れないことがあります。ただし、大本の問題は、多くの場合、IBM WebSphere MQ エラー・ログで報告されます。そのため、それらのログを頻繁に調べて、クラスター化に関連するメッセージが書き込まれていないか確認することが重要です。メッセージを読み取って理解し、必要であれば処置を取ってください。

例えば、クラスターは定期的に情報をリパブリッシュしてクラスター・リソースを再評価しているため、クラスター内のキュー・マネージャーとの通信が途絶えたという結果から、特定のクラスター・リソースが削除されたことを知ることができます。このようなイベントが発生する可能性があることを示す警告が、メッセージ [AMQ9465](#) によって報告されます。これらのメッセージは、問題を調べる必要があることを示します。

ロード・バランシングに関する特別な考慮事項

キューの複数のインスタンスでクラスターのロード・バランシングが行われる場合は、各インスタンスでコンシューム側アプリケーションがメッセージを処理します。1つまたは複数のコンシューム側アプリケーションが強制終了されたり、メッセージの処理を停止したりすると、クラスター化ではキューのそれらのインスタンスにメッセージが引き続き送信される可能性があります。そのような場合、それらのメッセージは、アプリケーションが再び正常に機能するようになるまで処理されません。そのため、ソリューションではアプリケーションのモニターが重要であり、そのような状況でメッセージが再経路指定されるように処置をとる必要があります。このようなモニターを自動化するメカニズムの例については、[クラスター・キュー・モニター・サンプル・プログラム \(AMQSCLM\)](#) を参照してください。

可用性、リカバリー、および再始動

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することによってアプリケーションの可用性を高め、サーバーまたはストレージの障害後にメッセージをリカバリーします。

クライアント・アプリケーションの可用性を向上させるために、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ます。あるいは、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ます。クライアントの自動再接続機能は、WebSphere MQ classes for Java ではサポートされていません。

Windows、UNIX、Linux、および IBM i プラットフォームの場合、サーバー・アプリケーションを複数インスタンス・キュー・マネージャーにデプロイします。複数インスタンス・キュー・マネージャーは、複数サーバー上で単一のキュー・マネージャーとして実行するように構成されています。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。

WebSphere MQ は、Microsoft Cluster Server、または PowerHA for AIX (以前の AIX 上の HACMP) などのプラットフォーム固有のクラスタリング・ソリューションやその他の UNIX and Linux クラスタリング・ソリューションの一部として構成することができます。

サーバー・アプリケーションの可用性を増加させる別の方法は、サーバー・アプリケーションをキュー・マネージャー・クラスター内の複数のコンピューターにデプロイすることです。

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。WebSphere 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。

WebSphere MQ では、メッセージの受信、伝送、および送達を取り扱うキュー・マネージャーの活動のリカバリー・ログを維持することにより、メッセージが失われないようにしています。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 WebSphere MQ を停止した場合
2. 障害リカバリー。障害によって WebSphere 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不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。

一方、手動再接続では、クライアント・アプリケーションが `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。

316 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がある、IBM WebSphere 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。316 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` を指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 環境に統合された WebSphere MQ リソース・アダプターまたは JMS クライアントを使用する場合、自動クライアント再接続が使用できないことがあります。一部の管理対象環境には制限があります。詳しくは、[Java SE 環境および Java EE 環境での自動クライアント再接続の使用](#)を参照してください。

注：WebSphere MQ classes for Java では、自動クライアント再接続はサポートされていません。

表 28. サポートされるクライアント

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コンテナ)	7.0.1.3	7.0.1.3
	WebSphere 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 ¹⁾)	サポート対象外	7.0.1
	Windows Communication Foundation (管理対象 ¹)	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。

表 29. 自動再接続の構成要件

コンポーネント	要件	要件に適合しない場合の影響
WebSphere MQ MQI クライアントのインストール済み環境	316 ページの表 28 を参照してください。	MQRC_OPTIONS_ERROR
WebSphere MQ サーバーのインストール済み環境	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
MQI	以下のいずれかです。 <ul style="list-style-type: none"> MQCNO Options が MQCNO_RECONNECT または MQCNO_RECONNECT_Q_MGR に設定されている MQCONNX。 Defrecon=YES QMGR (mqclient.ini) JMS 内で、接続ファクトリーの CLIENTRECONNECTOPTIONS プロパティを設定します。 	接続が切断された場合、またはキュー・マネージャーが終了または失敗した場合は MQCC_FAILED。

317 ページの図 59 は、クライアントの再接続に関係するコンポーネント間の主な相互作用を示しています。

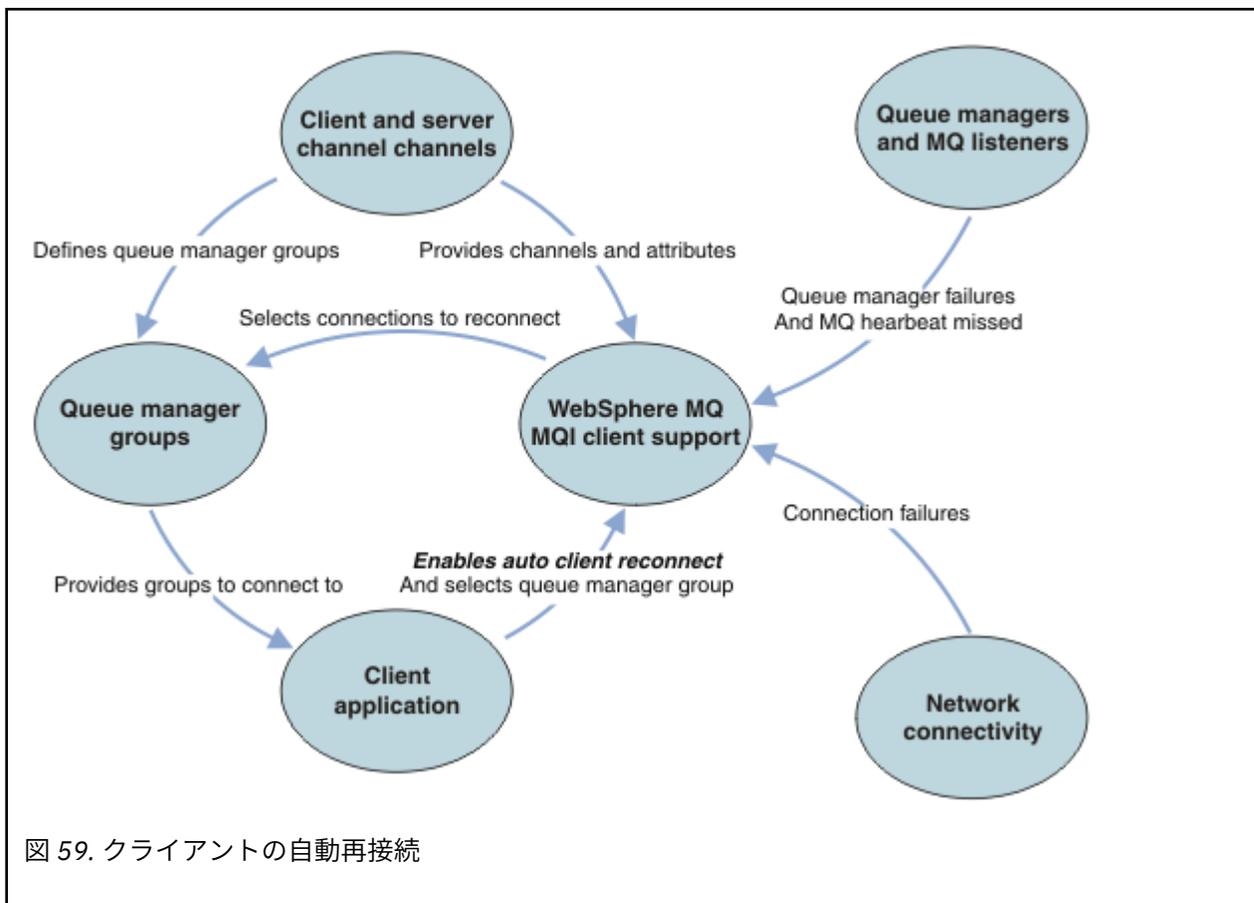


図 59. クライアントの自動再接続

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 WebSphere MQ MQI クライアントです。

- 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、MQCONNX 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 構成ファイルのチャンネル・スタンザにデフレコンの属性を設定することによって、コーディングの変更を行わずに、既存のプログラムの自動再接続を使用可能にします。
- 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、[399 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た 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。複数の再接続が試行されると、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

再接続可能な接続で MQCNO_RECONNECT_Q_MGR が指定され、接続が別のキュー・マネージャーへの再接続を試みました。

MQRC_RECONNECT_Q_MGR_REQD

MQGET 呼び出しの MQMO_MATCH_MSG_TOKEN などのオプションが、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、再接続オプションが設定されている場合に、アプリケーションが MQPMO_LOGICAL_ORDER (MQPUT および MQPUT1 を使用) または MQGMO_LOGICAL_ORDER (MQGET を使用) を使用しようとするたびに返されます。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 WebSphere MQ MQI クライアント・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 パラメーターにネットワーク・アドレスのリストを指定するか、CCDT 内のキュー・マネージャーに複数の行を定義してください。

一般的に、WebSphere MQ MQI クライアントはキュー・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。WebSphere MQ MQI クライアントが、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 オプションを MQCNO_RECONNECT_Q_MGR に設定します。WebSphere MQ MQI クライアントは、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 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 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

接続アフィニティー (AFFINITY)AFFINITY

接続のアフィニティー。

クライアント・チャンネル・ウェイト (CLNTWGHT)CLNTWGHT

クライアント・チャンネル・ウェイト。

接続名 (CONNAME)CONNAME

接続情報。

ハートビート間隔 (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。

キープアライブ間隔 (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。

KAINT は z/OS にのみ適用されることに注意してください。

キュー・マネージャー名 (QMNAME)QMNAME

キュー・マネージャー名。

サーバー接続の属性

ハートビート間隔 (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。

キープアライブ間隔 (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 WebSphere 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。KAIMT 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 WebSphere MQ MQI クライアント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。

- トランスポートによって自動的に実行される再接続は、IBM WebSphere MQ には認識されません。
- HBINT を設定すると、WebSphere 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 WebSphere 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。

WebSphere MQ MQI クライアントの自動再接続サポート

WebSphere MQ MQI クライアント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。 [402 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。

再接続には通常 1 分ほどしかかかりませんが、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 WebSphere 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` ファイルに設定されます。

```
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。

コンソール・メッセージのモニター

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出される情報メッセージには、特に重要で検討の必要なものが多数存在します。これらのメッセージは直接問題を示すものではありませんが、対処が必要になる可能性のある潜在的な問題を示しているため、トラッキングに役立つ可能性があります。

この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。

高可用性構成と共に WebSphere MQ を使用する

WebSphere MQ キュー・マネージャーを高可用性 (HA) 構成で操作する場合、キュー・マネージャーを PowerHA for AIX (以前の HACMP) や Microsoft Cluster Service (MSCS) などの HA マネージャー、または WebSphere MQ 複数インスタンス・キュー・マネージャーのいずれかと共に作動するようにセットアップできます。

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。

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によるグループです。相互に接続されて、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 WebSphere MQ クラスターと HA クラスターとの間の関係は、322 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ると、HACMP や MSCS などのクラスター・テクノロジーを必要とせずに、WebSphere MQ に基づいて独自の高可用性メッセージング・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。

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。

- WebSphere 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
- WebSphere MQ エクスプローラーとの統合

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。

UNIX and Linux 上で高可用性クラスターと共に WebSphere MQ を使用する

UNIX and Linux プラットフォーム上で高可用性 (HA) クラスターとともに WebSphere MQ を使用できます。例えば、PowerHA for AIX (旧称 HACMP)、Veritas Cluster Server、HP Serviceguard、または Red Hat Cluster Suite を使用する Red Hat Enterprise Linux クラスターなどです。

WebSphere MQ バージョン 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。WebSphere MQ バージョン 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、[322 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](#)、[323 ページの『WebSphere MQ クライアント』](#)、および [323 ページの『HA クラスター内で作動する WebSphere 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。WebSphere 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 WebSphere 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。

WebSphere 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 WebSphere 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。WebSphere MQ バージョン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。

WebSphere MQ MQI クライアントが2フェーズ・トランザクションに参加することを可能にする拡張トランザクション・クライアント (ETC) は、同じキュー・マネージャーに常に接続する必要があります。ETC は、キュー・マネージャーのリストから選択するために IP ロード・バランサーなどの手法を使用することができません。HA 製品を使用するとき、キュー・マネージャーはそれが実行しているノードの ID (名前およびアドレス) を保持するので、ETC は HA 制御の下にあるキュー・マネージャーと共に使用できます。

HA クラスター内で作動する WebSphere MQ

すべての HA クラスターには、フェイルオーバーの単位の概念があります。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。

WebSphere 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に予約されている、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 WebSphere 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ますようにします。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。

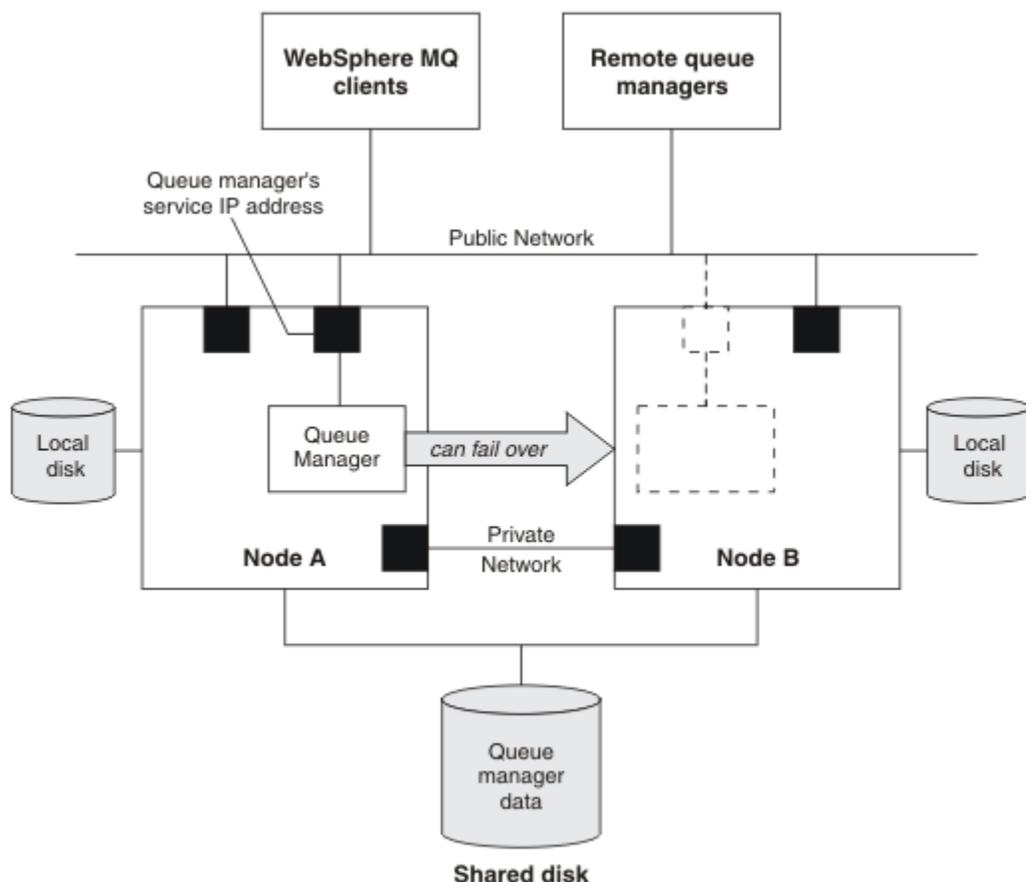


図 60. HA クラスター

図 1 は、2 つのノードのある HA クラスターを示しています。HA クラスターは、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ため、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 クラスター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 クラスターがノード 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 クラスターに複数のキュー・マネージャーが含まれる場合は、その HA クラスター構成のために、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 クラスター内の各キュー・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ー・ノードで使用されます。

通常、HA クラスターは root ユーザーとして実行されます。WebSphere MQ は、mqm ユーザーとして実行されます。WebSphere 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 クラスター・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、クラスター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 WebSphere 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: WebSphere 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。

共有ディスクの構成

HA クラスター内の WebSphere 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。

共有ディスクを構成するには、以下のステップを完了します。

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。

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 stanza には、LogPath の値があります。

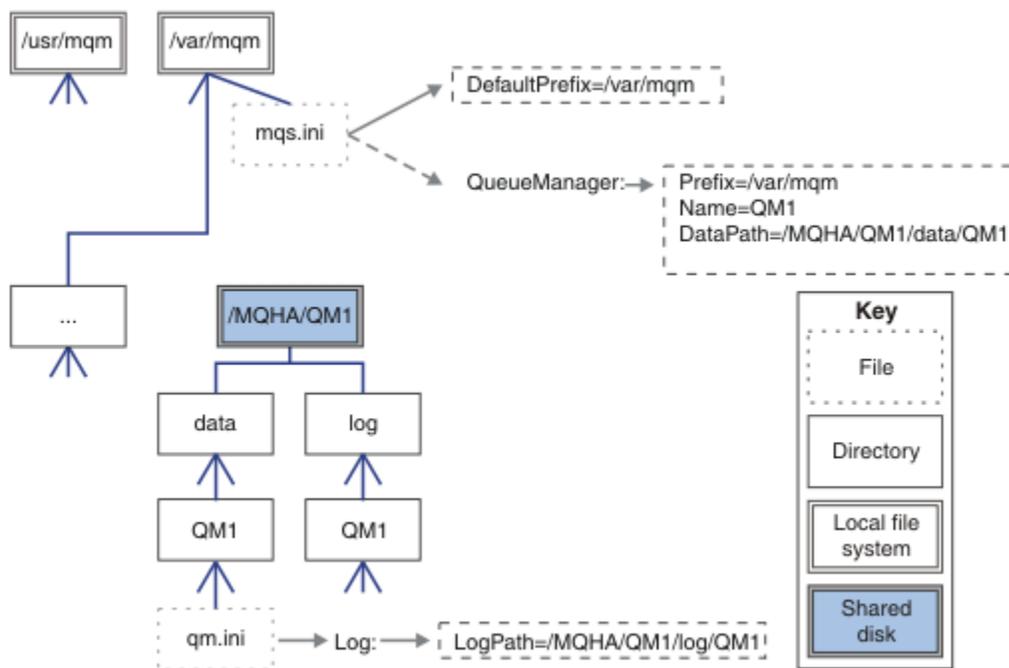


図 61. 共有に指定された data および log ディレクトリー

高可用性 (HA) クラスターで使用するキュー・マネージャーの作成

高可用性クラスターでキュー・マネージャーを使用するための最初のステップは、いずれかのノードでキュー・マネージャーを作成することです。

HA クラスターで使用するキュー・マネージャーを作成するには、キュー・マネージャーを作成するクラスター内のノードの1つを選択します。このノードで、以下のステップを実行します。

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・マネージャーの初期構成 (キューおよびチャネルの作成など) を完了し、キュー・マネージャーの開始時にリスナーを自動的に開始するようにキュー・マネージャーを設定します。
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. **dspmqinf** コマンドを以下のように使用して、後のタスクで使用する **addmqinf** コマンドを表示します。これについては、[326 ページの『キュー・マネージャー構成情報を高可用性 \(HA\) クラスターの他のノードに追加する』](#)で説明されています。

```
dspmqinf -o command qmgrname
```

ここで、`qmgrname` は、キュー・マネージャーの名前です。

7. 表示される **addmqinf** コマンドは、以下の例に示すようなものとなります。

```
addmqinf -sQueueManager -vName=qmgrname -vDirectory=qmgrname \  
-vPrefix=/var/mqm -vDataPath=/MQHA/qmgrname/data/qmgrname
```

表示されたコマンドを注意深くメモしてください。

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。

これで、[326 ページの『キュー・マネージャー構成情報を高可用性 \(HA\) クラスターの他のノードに追加する』](#)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。

キュー・マネージャー構成情報を高可用性 (HA) クラスターの他のノードに追加する

キュー・マネージャーの構成を HA クラスター内の他のノードに追加する必要があります。

この作業を完了する前に、[326 ページの『高可用性 \(HA\) クラスターで使用するキュー・マネージャーの作成』](#)のステップを完了している必要があります。

キュー・マネージャーの構成情報を HA クラスター内の各ノードに追加するには、他の各ノードで以下のステップを完了します。

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. `/var/mqm/mqs.ini` を直接編集するか、または [326 ページの『高可用性 \(HA\) クラスターで使用するキュー・マネージャーの作成』](#) のステップ 6 および 7 で **dspmqinf** コマンドにより表示された **addmqinf** コマンドを発行することにより、キュー・マネージャーの構成情報をノードに追加する。
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と停止に使用するコマンドは、**addmqinf** コマンドと同じ IBM WebSphere MQ インストール済み環境から発行する必要があります。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。

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。

キュー・マネージャーを高可用性 (HA) クラスターの制御下で始動する

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 amqzsmuc0 amqzma0 amqfcxba amqfcpub amqpcsea amqzlaa0 \
    amqzlsa0 runmqchi runmqlsr amqcrsta amqirmfa amqrmppa \
    amqzfuma amqzmaa 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。

キュー・マネージャーを高可用性 (HA) クラスターの制御下で停止する

ほとんどの場合、キュー・マネージャー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 クラスターの制御下でキュー・マネージャー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・マネージャーを即時に停止する方法を示す例です。 このスクリプトは、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 クラスター・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、[326 ページの『キュー・マネージャーを高可用性 \(HA\) クラスターの制御下で始動する』](#)にあります)。

```
#!/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 amqfcpub amqpcsea amqzlaa0 \
            amqzlsa0 runmqchi runmqslr amqcrsta amqirmfa amqrmppa \
            amqzfuma amqzmaa 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

```

キュー・マネージャーのモニター

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。

WebSphere MQ バージョン 7.1 から、WebSphere 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。バージョン 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、バージョン 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. バージョン 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 WebSphere 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 クラスター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 クラスターがキュー・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 クラスターは、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 クラスター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 クラスター・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、[326 ページの『キュー・マネージャーを高可用性 \(HA\) クラスターの制御下で始動する』](#)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

```

キュー・マネージャーを高可用性 (HA) クラスターの制御下に置く

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。

リソース・グループがキュー・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むように定義するには、以下のステップを完了します。

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。

キュー・マネージャーを高可用性 (HA) クラスター・ノードから削除する

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。

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、以下のステップを完了します。

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
```

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。ただし、これはキュー・マネージャーのデータ・ファイルおよびログ・ファイルを完全に削除す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。

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 **WebSphere MQ for Windows** にのみ適用されます。

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。

- [331 ページの『MSCS クラスターの紹介』](#)
- [332 ページの『MSCS クラスター化用の IBM WebSphere 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 WebSphere MQ を構成する方法を説明します。

- [334 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』](#)
- [335 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』](#)
- [337 ページの『MSCS 制御下にキュー・マネージャーを置く』](#)
- [343 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、以下の各項で、WebSphere MQ で MSCS を使用するうえで役立つヒントを挙げ、WebSphere 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。

- [344 ページの『MSCS の使用に関するヒント』](#)
- [347 ページの『IBM WebSphere MQ MSCS サポート・ユーティリティー・プログラム』](#)

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。

[321 ページの『高可用性構成と共に WebSphere MQ を使用する』](#)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 WebSphere 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 WebSphere 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こと、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 つのコンピューターからなるクラスターの標準的なセットアップを、[332 ページの図 62](#) に示します。

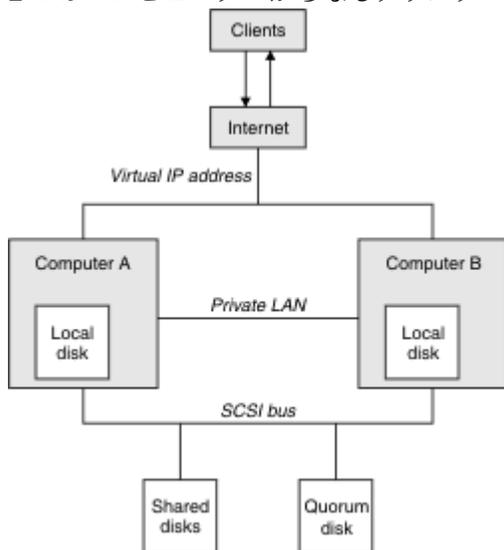


図 62. 2 台のコンピューターにおける MSCS クラスター

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。

MSCS クラスター化用の IBM WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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も存在している必要があります。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 WebSphere 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。335 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、334 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。

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。337 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、343 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。

対称性のセットアップ

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とき、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 333 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。

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、WebSphere MQ に関係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いてください。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、WebSphere 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、コードは、管理者グループと mqm グループ、および SYSTEM アカウントが完全な制御権を持っていること、および Everyone が古いノードに対する読み取りアクセス権を持っている場合は、その許可が宛先ノードに追加されていることを検査します。

ドメイン・アカウントを使用して WebSphere 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。

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。

クラスター・モード

WebSphere MQ でクラスター・システムを実行する場合、アクティブ/パッシブ、またはアクティブ/アクティブの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、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つの共有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。

WebSphere 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。

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 WebSphere MQ エクスプローラーで作成するかの 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 WebSphere MQ エクスプローラーの場合、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^に、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:\WebSphere 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 WebSphere MQ デフォルト・ディレクトリーで作成されます。
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・ディレクトリーに `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. [335 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。

WebSphere MQ エクスプローラーを使用してキュー・マネージャーを作成する

MSCS で使用するよう^に、IBM WebSphere MQ エクスプローラーを使用してキュー・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 WebSphere MQ エクスプローラーを始動します。
2. Navigator ビューで、ツリー・ノードを展開して、Queue Managers ツリー・ノードを見つけます。
3. Queue Managers ツリー・ノードを右クリックし、New->Queue Manager を選択します。「キュー・マネージャーの作成」パネルが表示されます。
4. ダイアログを完了し (ステップ 1)、Next> をクリックします。
5. ダイアログを完了し (ステップ 2)、Next> をクリックします。
6. ダイアログ (ステップ 3) を完了し、「Start Queue Manager」および「Create Server Connection Channel」が選択されていないことを確認してから、「Next>」をクリックします。
7. ダイアログを完了し (ステップ 4)、Finish をクリックします。
8. [335 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。

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\に、このトピックで説明されているように、WebSphere 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 SSL 接続を使用し、その SSL のキー・リポジトリーがローカル・マシン上のキュー・マネージャーのデータ・ディレクトリー内にある場合は、そのキー・リポジトリーは、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、SSL 鍵リポジトリーの場所を指定するキュー・マネージャー属性 SSLKEYR は、キュー・マネージャーのデータ・ディレクトリーの下にある `MQ_INSTALLATION_PATH\qmgrs\QMGRNAME\ssl\key` に設定されます。

`MQ_INSTALLATION_PATH` WebSphere 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。

`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 WebSphere

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 が新しい SSL 鍵リポジトリ・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. 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、[337 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 [346 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・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・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・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・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・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:\
WebSphere MQ" /ld "e:\
WebSphere 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、*WebSphere 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 WebSphere MQ エクスプローラーを使用し、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・ツリー・ノードを右クリックして、**Start** を選択します。キュー・マネージャーが開始します。
 - b. **Queues** ツリー・ノードを右クリックし、**New->Local Queue...** を選択して、キューに名前を付けます。
 - c. **Finish** をクリックします。
 - d. キューを右クリックし、**Put Test Message...** 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 **Put Test Message** をクリックし、パネルを閉じます。
 - f. キューを右クリックし、**Browse Messages...** 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にあることを確認し、**Close** 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、**Clear Messages...** 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、**Delete...** を選択します。確認パネルが表示されたら、**OK** 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・ツリー・ノードを右クリックして、**Stop...** 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. WebSphere 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 WebSphere MQ エクスプローラーのキュー・マネージャー・プロパティ・パネルで、「Startup (始動)」フィールドを manual に設定します。

10. [337 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。

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 WebSphere 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、[332 ページの『MSCS クラスター化用の IBM WebSphere 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、WebSphere 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、[347 ページの『IBM WebSphere MQ MSCS サポート・ユーティリティー・プログラム』](#)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、[334 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』](#)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、[335 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 WebSphere 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



重要: 32 ビットの DLL を使用する WebSphere MQ 7.5 では、MSCS サポートを提供しています。Windows 2012 の制限により、IBM WebSphere MQ キュー・マネージャーは再始動後にフェイルオーバーしません。

Microsoft は Windows 2012 での 32 ビット DLL の使用を推奨していないため、この問題に対するオペレーティング・システムのフィックスは現在提供されていません。IBM は IBM WebSphere MQ 7.5 では 64 ビット・ライブラリーを提供していません。

IBM MQ 8.0 以降では 64 ビット・ライブラリーを提供しているため、Windows 2012 以降の MSCS 機能を最大限に使用するには、このバージョンの製品を使用する必要があります。

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、「**IBM 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、IBM 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 IBM 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**。どのキュー・マネージャーのキュー・マネージャーであるかを識別しやすくする名前を選択します。
- **Run in a separate Resource Monitor**。分離を向上させます。
- **Possible owners**。両方のノードを設定します。
- **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 WebSphere 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 WebSphere MQ プロセスの実行がブロックされる場合があります。

- **Parameters**。以下ようになります。

- **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、[347 ページの『PostOnlineCommand および 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、[347 ページの『PostOnlineCommand および 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、[343 ページの『looksAlive および isAlive ポーリングの要約』](#)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、[347 ページの『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 WebSphere 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. Queues ツリー・ノードを右クリックし、New->Local Queue...を選択して、キューに名前を付けます。
 - b. Finish 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. キューを右クリックし、Put Test Message...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 Put Test Message をクリックし、パネルを閉じます。
 - e. キューを右クリックし、Browse Messages...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にあることを確認して、Close 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、Clear Messages...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、Delete...を選択します。確認パネルが表示されたら、OK 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、[335 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。) Initiate Failure を右クリックして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、「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、「**IBM 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、IBM 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 IBM 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**。どのキュー・マネージャーのキュー・マネージャーであるかを識別しやすくする名前を選択します。
- **Run in a separate Resource Monitor**。分離を向上させます。
- **Possible owners**。両方のノードを設定します。
- **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、WebSphere 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 WebSphere MQ プロセスの実行がブロックされる場合があります。

- **Parameters**。以下ようになります。

- **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、347 ページの『[PostOnlineCommand および 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、347 ページの『[PostOnlineCommand および 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、343 ページの『[looksAlive および isAlive](#) ポーリングの要約』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、[347 ページの『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 WebSphere 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. Queues ツリー・ノードを右クリックし、New->Local Queue...を選択して、キューに名前を付けます。
 - b. Finish 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. キューを右クリックし、Put Test Message...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 Put Test Message をクリックし、パネルを閉じます。
 - e. キューを右クリックし、Browse Messages...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にあることを確認して、Close 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、Clear Messages...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、Delete...を選択します。確認パネルが表示されたら、OK 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、[335 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。) Initiate Failure を右クリックして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 ([333 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 WebSphere MQ MSCS* のリソースを作成します。ウィザードでさまざまな項目を指定する必要があります。指定項目は以下のとおりです。

- **Name。**どのキュー・マネージャーのキュー・マネージャーであるか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します。
- **Run in a separate Resource Monitor。**分離を向上させます。
- **Possible owners。**両方のノードを設定します。
- **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、WebSphere 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 WebSphere MQ プロセスの実行がブロックされる場合があります。

- **Parameters。** 以下のようになります。
 - **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、[347 ページの『PostOnlineCommand および 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、[347 ページの『PostOnlineCommand および 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、[343 ページの『looksAlive および isAlive ポーリングの要約』](#)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、[347 ページの『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 WebSphere 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. Queues ツリー・ノードを右クリックし、New->Local Queue...を選択して、キューに名前を付けます。
 - b. Finish 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. キューを右クリックし、Put Test Message...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 Put Test Message をクリックし、パネルを閉じます。
 - e. キューを右クリックし、Browse Messages...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にあることを確認して、Close 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、Clear Messages...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、Delete...を選択します。確認パネルが表示されたら、OK 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、[335 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。) [Initiate Failure](#) を右クリックして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。

looksAlive および isAlive ポーリングの要約

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。

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 SSL 接続を使用する場合は、WebSphere MQ エクスプローラーまたは MQSC コマンド ALTER QMGR を使用して、ローカル・ディレクトリー上の SSL キー・リポジトリー・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. 344 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・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、344 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・アドミニストレーターを始動します。
2. クラスタへの接続をオープンします。
3. Groups を選択し、移動するキュー・マネージャーを含む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 WebSphere 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\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・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・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 346 ページの『[キュー・マネージャーのログ・ファイル](#)』を参照)。
3. 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。
4. ファイルが現在収められている共用ディスクが、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・ノード上でオンラインになっていることを確認します。
5. 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:\
WebSphere MQ" /ld "c:\
WebSphere 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、*WebSphere 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。

6. 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(335 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 の使用に関するヒント

このセクションには、MSCS の WebSphere 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 WebSphere 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。

MSCS の動作確認

以下のステップに従って、稼働中の MSCS クラスタがあることを確認してください。

334 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・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. そのグループ内に、システム・クロック (パス名 C:\winnt\system32\clock.exe および C:\ の作業ディレクトリー) を指定して、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。

手動による始動

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、WebSphere 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 IBM MQSeries サービスを再開できます。

WebSphere 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。詳細については、335 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 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

クラスタリングで WebSphere MQ を使用するには、ノード A のキュー・マネージャーごとに、ノード B のキュー・マネージャーが同一でなければなりません。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 335 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ーがありません。WebSphere 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、347 ページの『[IBM WebSphere MQ MSCS サポート・ユーティリティー・プログラム](#)』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(332 ページの図 62 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。

- [334 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』
- [335 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』](#)
- [337 ページの『MSCS 制御下にキュー・マネージャーを置く』](#)

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、[335 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

WebSphere 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。

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 WebSphere MQ エクスプローラーを使用して、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。

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、WebSphere 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。

- WebSphere 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、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。

PostOnlineCommand および *PreOfflineCommand*

これらのコマンドを使用して、WebSphere 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 WebSphere 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 WebSphere 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、WebSphere 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。Windows 64 ビット・システムで WebSphere 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、これは `\Windows\system32` ではなく `\Windows\SysWOW64` ディレクトリーです。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。WebSphere 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 WebSphere MQ コマンドを実行し、例えばリクエスト・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。

優先ノードの使用

キュー・マネージャーごとに優先ノードを構成すると、便利な場合があります。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。これにより、`immediate` フェイルバック設定が使用できなくなります。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。これにより、`preferred node` オプションが使用できなくなります。

アプリケーション・イベント・ログで COM+ エラーが発生した場合

新しくインストールされた MSCS クラスターに WebSphere 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) がそのような環境で実行するように構成されていない場合に、Microsoft Cluster Server (MSCS) 環境で WebSphere MQ を実行しようとしていることを意味します。クラスター環境で MSDTC を構成する方法に関する詳細については、Microsoft の資料を参照してください。

IBM WebSphere MQ MSCS サポート・ユーティリティー・プログラム

コマンド・プロンプトで実行できる MSCS の IBM WebSphere 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 WebSphere 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 WebSphere MQ MSCS リソース・タイプを登録抹消すると、そのタイプのリソースを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、[344 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 WebSphere MQ のインストール後まだシステムをリポートしていない場合は、リポ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

[335 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 WebSphere MQ Explorer 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ーは、ご使用の IBM WebSphere 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る)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。

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。

[349 ページの図 63](#) は、QM1 に対する複数インスタンス構成を示しています。2つのサーバーに IBM WebSphere 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。

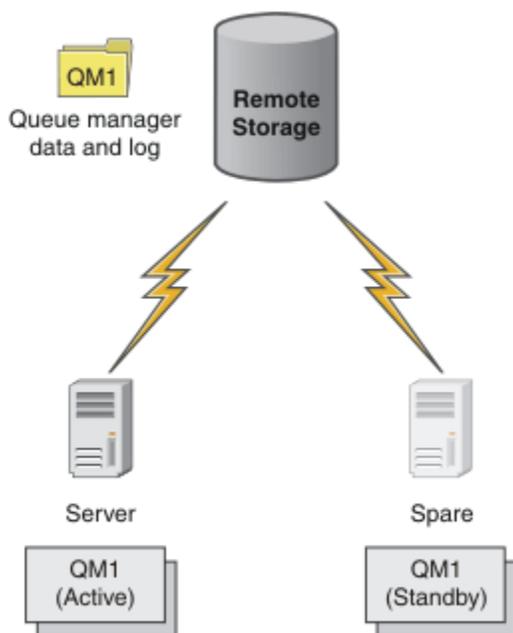


図 63. 複数インスタンス・キュー・マネージャー

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ということ。

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こると、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・システムの間接続障害。
- 無応答のキュー・マネージャー・プロセス。WebSphere 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。

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 WebSphere 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 WebSphere 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。

WebSphere MQ MQI のクライアントおよびチャネルは、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。クライアントの自動再接続機能は、IBM WebSphere MQ classes for Java ではサポートされていません。

サポートされているオペレーティング・システム

複数インスタンス・キュー・マネージャーは、バージョン 7.0.1 以降のいずれかの非 z/OS プラットフォーム上で作成することができます。

クライアントの自動再接続は、バージョン 7.0.1 以降の MQI クライアントで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 WebSphere 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、`mqm` の `uid` と `gid` をすべてのシステムで同じにする必要があります。各システム上の `/etc/passwd` を編集して、`mqm` 用の共通の `uid` および `gid` を設定してから、システムをリブートする必要がある場合があります。

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。378 ページの『[Windows でキュー・マネージャーの共有のデータとログのディレクトリーおよびファイルを保護する](#)』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、353 ページの『[ドメイン・ワークステーションまたは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](#)』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、368 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 WebSphere 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 WebSphere 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、`dspmqlinf` コマンドと `addmqinf` コマンドが役立ちます。あるいは、`mqm.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。391 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](#)』、353 ページの『[ドメイン・ワークステーションまたは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](#)』、および 368 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 `mqm.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。
2. すべての IBM WebSphere 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。

3. 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。

4. 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 WebSphere 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。
5. 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。
6. 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。

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、[382 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 [378 ページの『Windows でキュー・マネージャーの共有のデータとログのディレクトリーおよびファイルを保護する』](#)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、[353 ページの『ドメイン・ワークステーションまたは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』](#)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

Version 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。Version 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。[353 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ー・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ー・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、代替セキュリティー・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、Version 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ー・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、[353 ページの『ドメイン・ワークステーションまたは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』](#)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、[382 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、[384 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・ファイルおよびログ・ファイルの読み取りおよび書き込み』](#)と [387 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、[378 ページの『Windows でキュー・マネージャーの共有のデータとログのディレクトリーおよびファイルを保護する』](#)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 WebSphere 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. [356 ページの『IBM WebSphere MQ』](#)。
2. [360 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への IBM WebSphere MQ のインストール』](#)。
3. [363 ページの『キュー・マネージャー・データおよびログ・ファイル用の共有ディレクトリーの作成』](#)。
4. [366 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、[353 ページの『ドメイン・ワークステーションまたは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』](#)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

Version 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。[378 ページの『Windows でキュー・マネージャーの共有のデータとログのディレクトリーおよびファイルを保護する』](#)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 WebSphere MQ for Windows をインストールした場合は、キュー・マネージャーのデータ・ファイルとログ・ファイルがドメイン mqm グループで保護され、共有が可能になります。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、[368 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・マネージャーの作成』](#)の作業の手順を実行してください。

関連情報

[Windows 2000、Windows Server 2003、および Windows Server 2008 クラスタ・ノードをドメイン・コントローラーとして構成する](#)

ドメイン・ワークステーションまたは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 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 WebSphere MQ サーバーとして使用される 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 WebSphere MQ サーバーとして使用される 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。検査手順「[共有ファイル・システムの動作の検査](#)」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。これらの作業によって、ドメイン・コントローラーとドメインを作成し、1つのサーバーに IBM WebSphere MQ for Windows をインストールし、データ・ファイルとログ・ファイルのファイル共有フォルダーを作成します。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. [356 ページの『IBM WebSphere MQ』](#)。
2. [360 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への IBM WebSphere MQ のインストール』](#)。
3. [363 ページの『キュー・マネージャー・データおよびログ・ファイル用の共有ディレクトリーの作成』](#)。
4. [366 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』](#)。

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 *QMGR* 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 *QMGR* 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・マネージャーを複数インスタンス・キュー・マネージャー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・マネージャーでも複数インスタンス・キュー・マネージャーでも共通です。

手順

1. IBM WebSphere 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 360 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への IBM WebSphere MQ のインストール』](#)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 WebSphere 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 WebSphere MQ 構成ファイル *mqs.ini* の **QMGR QueueManager** スタンザにある **Prefix** 変数と **InstallationName** 変数を調整する必要があります。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 366 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、**dspmqinf** コマンドを実行します。

```
dspmqinf QMGR
```

システム応答:

```
QueueManager:  
  Name=QMGR  
  Directory=QMGR  
  Prefix=C:\Program Files\IBM\WebSphere 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 **dspmqinf** コマンドを再度実行します。

```
dspmqinf -o command QMGR
```

システム応答:

```
addmqinf -s QueueManager -v Name=QMGR  
-v Directory=QMGR -v Prefix="C:\Program Files\IBM\WebSphere 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:\Program Files\IBM\WebSphere MQ"  
-v DataPath=\\sun\wmq\data\QMGR
```

WebSphere 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:

```
WebSphere 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'.  
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

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:

```
WebSphere 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。

```
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere 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、[376 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。

IBM WebSphere MQ

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 WebSphere MQ サーバーとして使用される 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 WebSphere MQ サーバーとして使用される 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. このタスク・ステップは、インストールされているがロールで構成されていない Windows Server 2008 と整合しています。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 WebSphere 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ことが必要です。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. ユーザーを追加し、それをグローバル・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、352 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 WebSphere 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) **開始** > 「**実行 ...**」 をクリックします。 `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 によってそ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、**新規フォレスト内に新規ドメインを作成** > 「**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、「**フォレストの機能レベル**」 > 「**次へ**」のリストから「**Windows Server 2003**」以降を選択します。

IBM WebSphere MQ でサポートされる Windows Server の最も古いレベルは、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 WebSphere MQ" 準備ウィザードの動作が変更されます。これにより、"「IBM WebSphere MQ」の準備」ウィザードは、ドメイン内の IBM WebSphere 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` グローバル・グループを持たないドメインにワークステーションまたはサーバーをインストールすることができます。その場合は、`Domain mqm` グループと同じプロパティでグループを定義する必要があります。IBM WebSphere MQ がドメインにインストールされている場合は、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。各ドメイン・グループを、異なる IBM WebSphere MQ インストール済み環境のローカル `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。

- ドメイン内のワークステーションまたはサーバーに IBM WebSphere MQ をインストールするときに指定するドメイン・ユーザーは、Domain mqm グループのメンバー、または 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. Domain mqm グローバル・グループの権限に「読み取りグループ・メンバーシップを許可」権限と「読み取り groupMembershipSAM を許可」権限を追加します。
- 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」
 - 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) 「ユーザー」ウィンドウで、Domain mqm > 「プロパティ」を右クリックします。
 - d) 「セキュリティ」 > 「拡張」 > 「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、ストリング 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」リストの下部にある「下位のユーザー オブジェクト」を選択します。
 - f) 「許可」リストから、「グループ・メンバーシップの読み取り」と「読み取り groupMembershipSAM」 「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 (例では *wmuser1*) が IBM IBM WebSphere MQ サービスを実行し、もう 1 人のユーザー (*wmuser2*) 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らなければなりません。
- この例では、*wmuser1* と *wmuser2* を Domain mqm グローバル・グループのメンバーにします。「IBM WebSphere 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。
- 単一のコンピューターに IBM WebSphere MQ をインストールするたびに IBM IBM WebSphere MQ サービスを実行するには、別のユーザーを指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。
- a) 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」
 - b) 「新しいオブジェクト - ユーザー」ウィンドウの「ユーザー ログオン名」フィールドに *wmuser1* 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。
 - c) 「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。
 - d) 「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。
 - e) ステップ a から d を繰り返して、*WebSphere MQ2* を *wmuser2* として追加します。
15. サービスとして IBM WebSphere MQ を実行します。
- IBM WebSphere 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。
- a) 「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。

- 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。
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。
- d) Windows セキュリティー・ウィンドウでプロンプトが出されたら、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 Vista および Windows Server 2008 では、ユーザー・アカウント制御 (UAC) はデフォルトで有効になっています。

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、360 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への IBM WebSphere MQ のインストール](#)』の作業に進みます。

Windows ドメイン内のサーバーまたはワークステーションへの *IBM WebSphere MQ* のインストール
この作業では、*wmq.example.com* Windows 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に *IBM WebSphere MQ* をインストールして構成します。

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 WebSphere MQ* サーバーとして使用される 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 WebSphere MQ* サーバーとして使用される 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

- 356 ページの『[IBM WebSphere MQ](#)』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。
- IBM WebSphere 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 WebSphere MQ* をインストールし、*wmq.example.com* ドメインのメンバーとして実行するように設定します。

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、352 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 WebSphere 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* に *mars* を DNS サーバーとして追加します。
 - 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、「インターネットプロトコルバージョン4」または「インターネットプロトコルバージョン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。
 - タイプ *wmq.example.com* と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でのタイプ *wmq.example.com*。
 - 「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・ウィンドウを開き、コマンド **ipconfig /all** を入力して、TCP/IP 設定を確認します。
- mars* で、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。
 - 「コンピューター」 > 「プロパティ」を右クリックします。「コンピューター名、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*」を入力して、**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」" というメッセージに応答して、**OK** > **OK** > 「> 今すぐ再始動」をクリックします。
- コンピューターが *wmq.example.com* ドメインのメンバーであ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」を開き、*mars* が「コンピューター」ウィンドウに正しくリストされていることを確認します。
- IBM WebSphere MQ for Windows を *mars* にインストールします。

IBM WebSphere 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 WebSphere MQ サーバーのインストール](#) を参照してください。

- a) *mars* に、ローカル管理者 *mars\Administrator* としてログオンします。
- b) **Setup** コマンドを IBM WebSphere MQ for Windows インストール・メディアで実行します。
- IBM WebSphere MQ のランチパッド・アプリケーションが起動します。
- c) 「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。
- d) 「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。
- 356 ページの『IBM WebSphere MQ』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) 「WebSphere 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 IBM WebSphere 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 WebSphere MQ がインストールされ、インストーラーによって "IBM WebSphere 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 IBM WebSphere 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。
- ローカル *mqm* グループを構成するか、Domain *mqm* グループを構成するか、右側で IBM IBM WebSphere MQ サービスを実行するユーザーを構成するかを選択します。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) 「セキュリティ設定」 > ローカル・ポリシー > 「ユーザー権限の割り当て」を開きます。ポリシーの中で、「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 WebSphere MQ" 準備ウィザードを実行します。
- "IBM WebSphere MQ" 準備ウィザードの実行について詳しくは、[WebSphere MQ 準備ウィザードを使用した WebSphere MQ の構成](#) を参照してください。
- a) IBM IBM WebSphere MQ インストーラーは、"準備 IBM WebSphere 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 WebSphere MQ** フォルダで "準備 IBM WebSphere 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 WebSphere 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) 「次へ」 をクリックし、"「ネットワーク内に Windows 2000 以降のドメイン・コントローラーが存在するかどうかを識別します」" という質問に対して 「はい」 をクリックしたままにします。
- c) 最初の「IBM WebSphere 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 WebSphere MQ for Windows for Windows ドメイン・ユーザーの構成」ウィンドウの「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* と入力します。パスワードを設定する場合は、「パスワード」フィールドにパスワードを入力します。「次へ」 をクリックします。
- ウィザードによって、IBM IBM WebSphere 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. 384 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・ファイルおよびログ・ファイルの読み取りおよび書き込み』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. 363 ページの『キュー・マネージャー・データおよびログ・ファイル用の共有ディレクトリーの作成』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。

関連概念

WebSphere MQ Windows サービスに必要なユーザー権限

キュー・マネージャー・データおよびログ・ファイル用の共有ディレクトリーの作成

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 WebSphere MQ サーバーとして使用される 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 WebSphere MQ サーバーとして使用される 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、356 ページの『IBM WebSphere MQ』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、352 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。これは、IBM WebSphere MQ サーバー上のローカル mqm グループ、および *wmqha* グローバル・グループのメンバーで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* および *venus* IBM WebSphere 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 WebSphere 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、「**拡張**」 > 「**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。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** の許可のリストで「フルコントロール」チェック・ボックスを選択し、「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。「管理者の許可」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** > 「**OK**」をクリックします。

次のタスク

それぞれの IBM WebSphere 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 WebSphere 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* と入力して、「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。366 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。

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

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、代替セキュリティー・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 WebSphere MQ サーバーとして使用される 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 WebSphere MQ サーバーとして使用される 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。これらの作業によって、ドメイン・コントローラーとドメインを作成し、1つのサーバーに IBM WebSphere MQ for Windows をインストールし、データ・ファイルとログ・ファイルのファイル共有フォルダーを作成します。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. [356 ページの『IBM WebSphere MQ』](#).
2. [360 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への IBM WebSphere MQ のインストール』](#).
3. [363 ページの『キュー・マネージャー・データおよびログ・ファイル用の共有ディレクトリーの作成』](#).

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、[352 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・フォルダーが含まれているディレクトリーを共有するために、グローバル・グループ *wmqha* にフル・コントロール権限を与えます。

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 WebSphere 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, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.  
および:  
5724-H72 (C) Copyright IBM Corp. 1994, 2024. 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・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:

```
WebSphere MQ queue manager created.  
Directory '\\sun\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.
```

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:

```
WebSphere 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'.  
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

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, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)
```

```
AMQ8006: WebSphere 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 **amqspout** を使用してテスト・メッセージを書き込みます。

```
echo 'A test message' | amqspout 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:

```
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere 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:

```
WebSphere 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: /Q オプションをコマンドに追加して、各ファイルまたはディレクトリーを削除するためのプロンプトがコマンドに表示されないようにします。

```
del /F /S C:\wmq\*.*  
rmdir /S C:\wmq
```

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。

この構成では、mini-domain または "domainlet" の概念を使用します。Windows 2000、Windows Server 2003、および Windows Server 2008 のクラスター・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、353 ページの『ドメイン・ワークステーションまたは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』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

2番目のドメイン・コントローラー IBM WebSphere MQ サーバーとして使用される 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。

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. 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。検査手順「[共有ファイル・システムの動作の検査](#)」は、UNIX and Linux に適用されます。Windows では検査は必ず成功します。
2. [356 ページの『IBM WebSphere MQ』](#)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. [372 ページの『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する操作』](#)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 WebSphere 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. [374 ページの『wmq.example.com ドメイン内のドメイン・コントローラーへの IBM WebSphere 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 WebSphere 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、「拡張」 > 「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 の「セキュリティー」ウィンドウに回答して、Domain Administrator の名前とパスワードを入力し、「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 の「セキュリティー」ウィンドウに回答して、Domain Administrator の名前とパスワードを入力し、「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:

```
WebSphere 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:

```
WebSphere 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'.
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

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:\Program Files\IBM\WebSphere 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、`-o command` パラメーターを指定して **dspmqlinf** コマンドを再度実行します。

```
dspmqlinf -o command QMGR
```

システム応答:

```
addmqinf -s QueueManager -v Name=QMGR  
-v Directory=QMGR -v Prefix="C:\Program Files\IBM\WebSphere 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
```

WebSphere 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:

```
WebSphere 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:

```
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere 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、[376 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

[372 ページ](#)の『[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する操作](#)』

[374 ページ](#)の『[wmq.example.com ドメイン内のドメイン・コントローラーへの IBM WebSphere 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・ノードをドメイン・コントローラーとして構成する](#)

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する操作

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

2 番目のドメイン・コントローラー IBM WebSphere MQ サーバーとして使用される 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。

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. [356 ページ](#)の『[IBM WebSphere MQ](#)』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、[352 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

- 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。
 - earth* で、*earth\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、「インターネット プロトコルバージョン 4」または「インターネット プロトコルバージョン 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。
 - タイプ *wmq.example.com* と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でのタイプ *wmq.example.com*。
 - 「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・ウィンドウを開き、コマンド **ipconfig /all** を入力して、TCP/IP 設定を確認します。
- 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。
- 「Active Directory ドメイン サービス」ウィザードを実行します。
 - 開始 > 「実行 ...」 をクリックします。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 によってそれらのファイルが自動的にインストールされます。
- 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。
 - 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。
 - 「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。
 - 「追加のドメイン・コントローラー・オプション」ウィンドウで、選択した「DNS サーバー」オプションと「グローバル・カタログ」オプションを受け入れます。「次へ」 > 「次へ」 をクリックします。
 - ディレクトリー・サービス復元モードの管理者パスワードで、「パスワード」と「パスワードの確認」を入力し、「次へ」 > 「次へ」 をクリックします。
 - 「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。
 - 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 WebSphere MQ のインストールに進みます。374 ページの『[wmq.example.com ドメイン内のドメイン・コントローラーへの IBM WebSphere MQ のインストール](#)』を参照してください。

関連タスク

356 ページの『[IBM WebSphere MQ](#)』

wmq.example.com ドメイン内のドメイン・コントローラーへの *IBM WebSphere 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 WebSphere MQ* をインストールして構成します。

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

2 番目のドメイン・コントローラー *IBM WebSphere MQ* サーバーとして使用される 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。

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

- 356 ページの『[IBM WebSphere MQ](#)』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。
- 372 ページの『[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する操作](#)』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 WebSphere 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 WebSphere MQ* をインストールして構成します。

手順

- IBM WebSphere MQ* で *sun* と *earth* をインストールします。

IBM WebSphere 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 WebSphere MQ* サーバーのインストール](#) を参照してください。

a) *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。

b) **Setup** コマンドを *IBM WebSphere MQ for Windows* インストール・メディアで実行します。

IBM WebSphere MQ のランチパッド・アプリケーションが始動します。

c) 「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。

d) 「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) 「**WebSphere 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 IBM WebSphere 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 WebSphere MQ がインストールされ、インストーラーによって "IBM WebSphere MQ" 準備ウィザードが開始されます。

IBM WebSphere 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 mqmmqm` が構成されます。の `Domain mqmmqm` メンバーになります。同じドメイン内の後続のドメイン・コントローラーは、`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 WebSphere MQ" 準備ウィザードを実行します。

"「IBM WebSphere MQ の準備」" ウィザードの実行について詳しくは、[WebSphere MQ 準備ウィザードを使用した WebSphere MQ の構成](#) を参照してください。

a) IBM WebSphere MQ インストーラーは、"準備 IBM WebSphere 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 WebSphere MQ** フォルダーで "準備 IBM WebSphere 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 WebSphere 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) 「**次へ**」 をクリックし、"「ネットワーク内に Windows 2000 以降のドメイン・コントローラーがあるかどうかを識別します」" という質問に対して 「**いいえ**」 をクリックしたままにします。¹

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 WebSphere 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。ウィザードは、サービスにログオンするユーザーとして `MUSR_MQADMIN` または `MUSR_MQADMIN1` を使用して、各コントローラー上に IBM WebSphere MQ サービスを作成します。

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。

ユーザーはローカルにログオンする権限を持っている必要があり、ドメイン・ローカル `mqm` グループ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。このタスクでは、ドメイン管理者をドメインのローカル `mqm` グループに追加します。

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、「**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 WebSphere 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。

¹ 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範囲はドメイン・範囲なので、どちらにしても違いはありません。IBM WebSphere MQ をドメイン内にないものとして、インストールする方が簡単です。

- b. 右クリック **MUSR_MQADMIN (管理者)** > **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. 「セキュリティ設定」 > ローカル・ポリシー > 「ユーザー権限の割り当て」を開きます。ポリシーの中で、「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。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. 384 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・ファイルおよびログ・ファイルの読み取りおよび書き込み』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. 368 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。

関連概念

WebSphere MQ Windows サービスに必要なユーザー権限

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**、**amqsphac**、および **amqsmhac** 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、368 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、Windows XP Service Pack 2 を実行している 400 MB の VMware イメージにインストールされます。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。

WebSphere MQ エクスプローラー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、WebSphere MQ エクスプローラー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。

注：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) +
CONNAME('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、WebSphere MQ エクスプローラーを使用して、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. サーバー上の共有ディレクトリ `C:\MQHA` をクライアント上の `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 amqsmnac -s SOURCE -t TARGET -m QM1
start amqspnac 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。
- WebSphere MQ エクスプローラーを使用して、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:\>amqspshac 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:\>amqsgshac 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 でキュー・マネージャーの共有のデータとログのディレクトリーおよびファイルを保護する
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 WebSphere 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 WebSphere 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。リモート・ファイル・サーバー上のフォルダーが、mars というサーバー上のローカルmqmグループによって保護されているとします。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。キュー・マネージャーが *mars* とは異なるサーバー上で実行されている場合、*mars* 上で実行されたときに作成されたデータ・ファイルおよびログ・ファイルにアクセスすることはできません。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ー・グループを設定することによって解決できます。379 ページの図 64 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。グローバル・グループを *mars* 上で作成するときに、そのグローバル・グループの名前をキュー・マネージャーに渡します。そのグローバル・グループ名を代替セキュリティー・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、そのセキュリティー・グループの名前も一緒に移動します。この名前は、qm.ini ファイルの **AccessMode** スタンザで SecurityGroup として転送されます。以下に例を示します。

```
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。

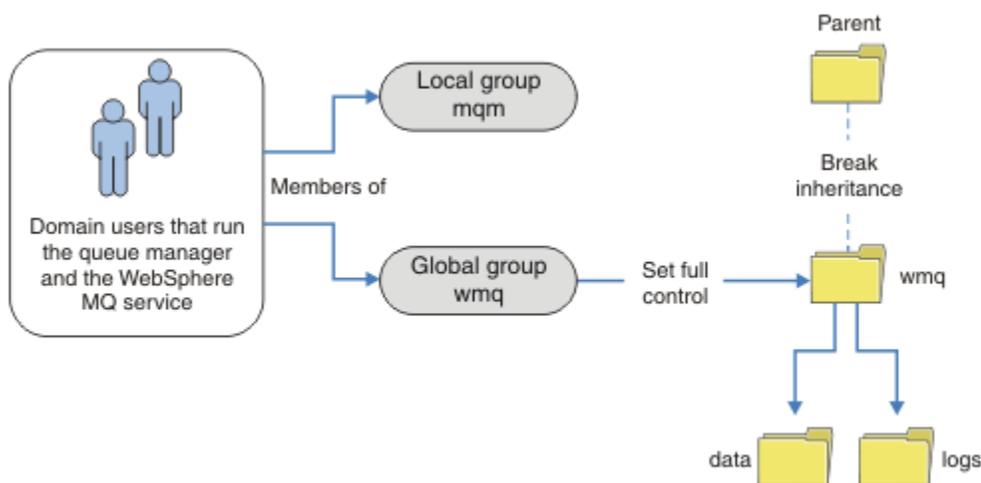


図 64. 代替グローバル・セキュリティー・グループを使用してキュー・マネージャーのデータとログを保護する操作 (1)

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・セキュリティー・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必要があります。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。379 ページの図 64 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 WebSphere MQ サーバーをデプロイする場合は、379 ページの図 64 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 WebSphere 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、IBM WebSphere MQ を実行するユーザーを Domain mqm グループのメンバーにします。380 ページの図 65 を参照してください。IBM WebSphere 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 WebSphere MQ の準備」ウィザードによって、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 の両方に入ります。

ヒント：同じユーザーが別々のサーバーで IBM WebSphere MQ を実行することは可能ですが、1つのサーバーで IBM WebSphere 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。

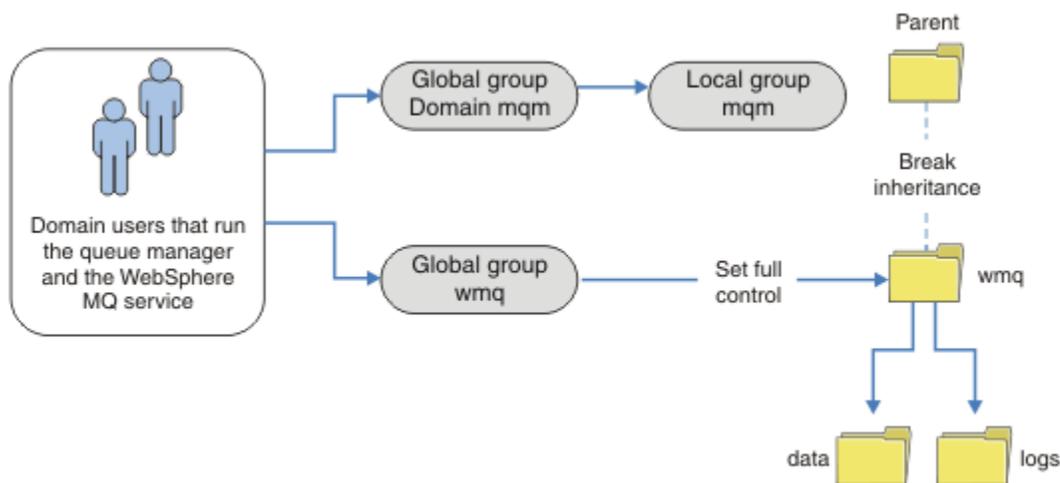


図 65. 代替グローバル・セキュリティ・グループを使用してキュー・マネージャーのデータとログを保護する操作 (2)

380 ページの図 65 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 381 ページの図 66 です。

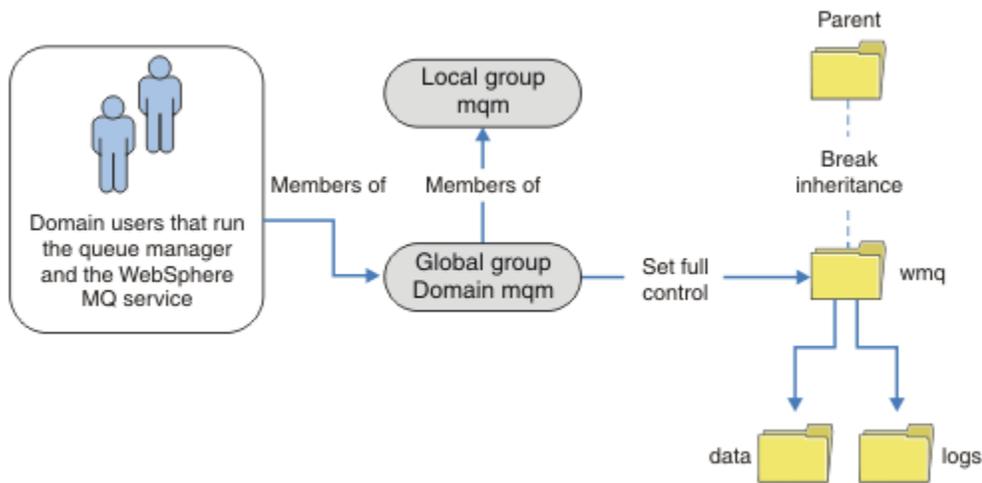


図 66. 代替グローバル・セキュリティ・グループを使用してキュー・マネージャーのデータとログを保護する操作 (3)

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。382 ページの図 67 を参照してください。382 ページの図 67 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 WebSphere 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 つのセットに区分化し、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。グローバル・グループ wmq1 および wmq2 は、テスト部門であるか実動部門であるかに応じて、正しいキュー・マネージャーに手動で追加する必要があります。2 つのユーザー・グループがあるため、381 ページの図 66 の場合と同様に、IBM WebSphere 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。

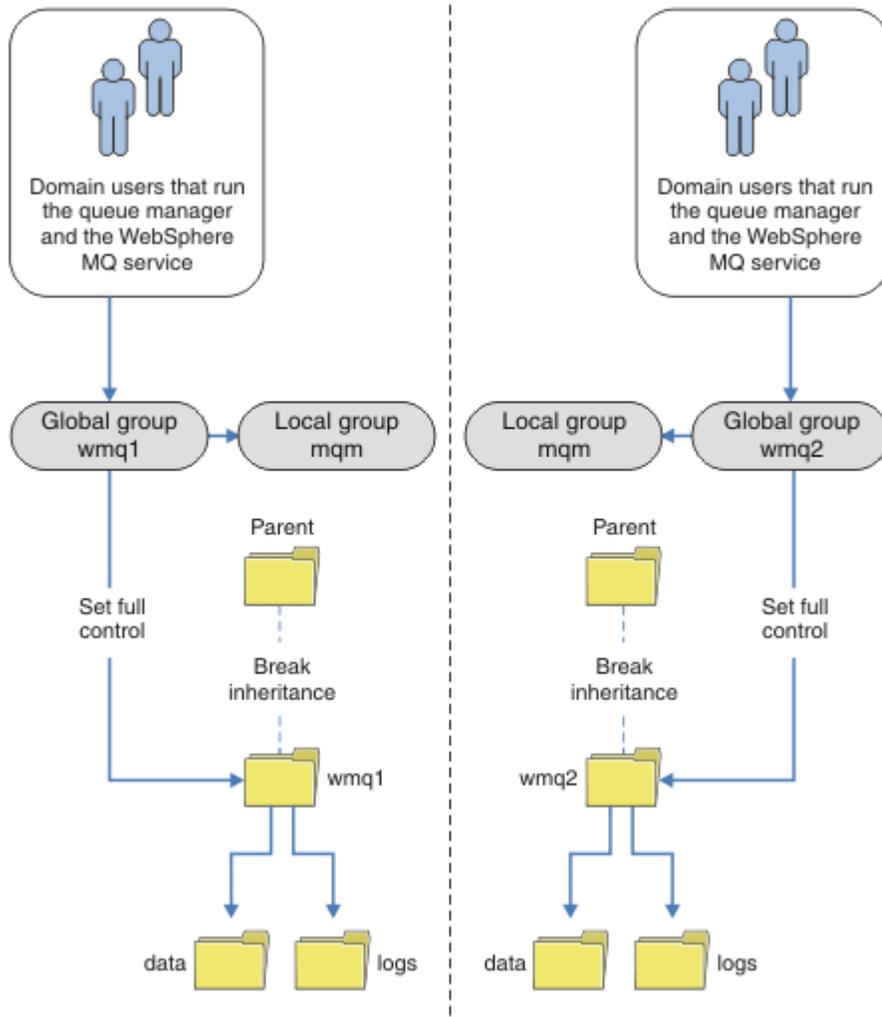


図 67. 代替グローバル・セキュリティ・プリンシパルを使用してキュー・マネージャーのデータとログを保護する操作 (4)

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、381ページの図66にあるようなシンプルなモデルに戻ることができます。

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 WebSphere 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 WebSphere MQサービスのユーザーIDが構成されます。

IBM WebSphere 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(384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・ファイルおよびログ・ファイルの読み取りおよび書き込み』)では、ローカルmqmグループによって、キュ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(387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。378ページの『[Windows でキュー・マネージャーの共有のデータとログのディレクトリーおよびファイルを保護する](#)』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 WebSphere 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 WebSphere 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、384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・ファイルおよびログ・ファイルの読み取りおよび書き込み](#)』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。384ページの図 68 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。

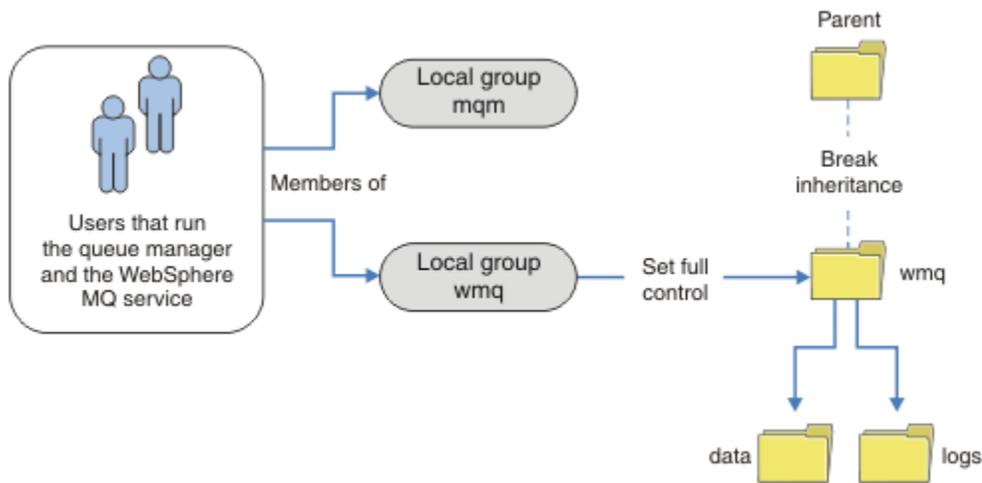


図 68. 代替ローカル・セキュリティ・グループ *wmq* を使用してキュー・マネージャーのデータとログを保護する操作

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、[378 ページの『Windows でキュー・マネージャーの共有のデータとログのディレクトリーおよびファイルを保護する』](#)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。

ローカル *mqm* グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りおよび書き込み

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。ファイルへのアクセスは、ローカル *mqm* グループによって保護されます。ディレクトリーは共有しません。

始める前に

1. IBM WebSphere 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 WebSphere 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、[352 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、352 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 WebSphere 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。インストールおよび構成ウィザードは、ローカル mqm グループ、およびキュー・マネージャー・プロセスを実行するユーザー ID に、ディレクトリーへのアクセス権限を自動的に付与します。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 WebSphere 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. ローカル mqm グループに読み取りおよび書き込みアクセスを許可するように、ディレクトリーに許可を設定します。

```
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。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 WebSphere MQ インストールがドメインの一部として構成されている場合は、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 WebSphere 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:

```
WebSphere 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. ローカル `mqm` グループに対する読み取りと書き込み、またはフル・コントロール権限がファイルに付与されてい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:

```
WebSphere 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'.  
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

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, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: WebSphere 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:

```
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere 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:

```
WebSphere 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: /Q オプションをコマンドに追加して、各ファイルまたはディレクトリーを削除するためのプロンプトがコマンドに表示されないようにします。

```
del /F /S C:\wmq\*.*  
rmdir /S C:\wmq
```

関連概念

[352 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

[387 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』](#)

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。

[366 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』](#)

[353 ページの『ドメイン・ワークステーションまたは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』](#)

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 WebSphere 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 WebSphere 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、[352 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、[352 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 WebSphere 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。インストールおよび構成ウィザードは、ローカル `mqm` グループ、およびキュー・マネージャー・プロセスを実行するユーザー ID に、ディレクトリーへのアクセス権限を自動的に付与します。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 WebSphere 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 WebSphere 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 WebSphere 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 `altmqm` と入力し、「作成」 > 「閉じる」をクリックします。
- d) IBM WebSphere MQ サービスを実行するユーザー ID を識別します。
 - i) 「スタート」 > 「ファイル名を指定して実行...」をクリックします。 `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 WebSphere 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 WebSphere MQ サービスを実行するユーザー ID を *altmqm* 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。ユーザー ID は、ディレクトリーを許可した *altmqm* グループのグローバル・グループを介して、直接的または間接的にメンバーでなければなりません。

ドメインの一部として IBM WebSphere MQ をインストールした環境で、[353 ページの『ドメイン・ワークステーションまたは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』](#)の作業を実行する場合、[356 ページの『IBM WebSphere MQ』](#)で作成するドメイン・ユーザー ID は、*wmquser1* と *wmquser2* になります。

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 WebSphere 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。

以下の手順で *wmquser1* と *wmquser2* を *altmqm* 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、**altmqm** > 「プロパティー」 > 「追加 ...」を右クリックします。
- ii) 「ユーザー、コンピュータ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 の「セキュリティ」ウィンドウで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 WebSphere 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, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

および:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. 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. ローカル・ユーザーに *user* 読み取りおよび書き込みアクセスを許可するように、ディレクトリーの許可を設定します。

```
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。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 WebSphere MQ インストールがドメインの一部として構成されている場合は、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 WebSphere 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:

```
WebSphere 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. ローカル mqm グループに対する読み取りと書き込み、またはフル・コントロール権限がファイルに付与されてい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:

```
WebSphere 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'.  
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

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, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: WebSphere 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:

```
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere 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:

```
WebSphere 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: /Q オプションをコマンドに追加して、各ファイルまたはディレクトリーを削除するためのプロンプトがコマンドに表示されないようにします。

```
del /F /S C:\wmq\*.*  
rmdir /S C:\wmq
```

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 Hat Enterprise5に基づいています。この手順は、他のUNIXプラットフォームでは異なります。

この例は、Windows XP Service Pack 2 を実行する 3 GB RAM を備えた 2 GHz ノートブック・コンピューター上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 Hat Enterprise5 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 XP Service Pack 2 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。

例

Server1	Server2
root としてログインします。	
<p>IBM WebSphere MQ のインストールの説明に従って WebSphere 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て (存在しない場合)、/var/mqm を定義します。</p> <p>最初のマシンの /etc/passwd の uid と gid の内容を mqm 用に表示します。以下に例を示します。</p> <pre>mqm:x:501:100:MQ User:/var/mqm:/bin/bash</pre> <p>2 番目のマシンの /etc/passwd にある mqm の uid と gid を突き合わせて、これらの値が同じであることを確認します。値を変更する必要がある場合、このマシンをリブートします。</p>	
<p>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。</p>	
<p>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。</p> <ol style="list-style-type: none"> mkdir /MQHA mkdir /MQHA/logs mkdir /MQHA/qmgrs 	<p>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。例えば、以下のようになります。</p> <ol style="list-style-type: none"> mkdir /MQHA
<p>MQHA ディレクトリーがユーザーとグループ mqm によって所有されていること、およびユーザーとグループのアクセス許可が rwx に設定されていることを確認します。例えば、ls -al と指定すると、以下のように表示されます。</p> <pre>drwxrwxr-x mqm mqm 4096 Nov 27 14:38 MQDATA</pre> <ol style="list-style-type: none"> chown -R mqm:mqm /MQHA chmod -R ug+rwx /MQHA 	
<p>以下のようにしてキュー・マネージャーを作成します。</p> <pre>crtmqm -ld /MQHA/logs -md /MQHA/qmgrs QM1</pre>	

表 30. Linux での複数インスタンス・キュー・マネージャー構成の説明 (続き)	
Server1	Server2
追加 ² /MQHA *(rw, sync, no_wdelay, fsid=0) を /etc/exports に追加します。	
NFS デーモンを開始します: /etc/init.d/nfs start	エクスポートされたファイル・システム /MQHA を以下のようにしてマウントします。 mount -t nfs4 -o hard,intr Server1:/ /MQHA
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。 dspmqinf -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 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM1	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
x パラメーターを指定して、いずれかの順序でキュー・マネージャー・インスタンスを開始します: strmqm -x QM1 キュー・マネージャー・インスタンスを開始するために使用するコマンドは、 addmqinf コマンドと同じ IBM WebSphere MQ インストール済み環境から発行する必要があります。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 での複数インスタンス・キュー・マネージャーの検査

サンプル・プログラム **amqsgbac**、**amqspbac**、および **amqsmbac** 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、391 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 XP Service Pack 2 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。

² '*' 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。

WebSphere MQ エクスプローラー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、WebSphere MQ エクスプローラー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・システムをホストするサーバーの一つを選択した可能性があります。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：MCAUSER を mqm に設定して SVRCONN チャネルを実行すると、この例の構成ステップの数を減らすことができます。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と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) +
CONNNAME('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、WebSphere MQ エクスプローラーを使用して、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. 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。
 - WebSphere MQ エクスプローラーを使用して、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>
```

複数インスタンス・キュー・マネージャーの削除

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、次いで **rmvmqinf** または **dltmqm** コマンドのいずれかを使用して他のサーバーからインスタンスを除去する必要があります。

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anzas は **dltmqm** コマンドを発行したサーバー上の **mqs.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 **mqs.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anzas を除去する必要があります。

UNIX and Linux システム上では、**mqs.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 **mqs.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
```

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

単一インスタンスまたは複数インスタンスのいずれかのキュー・マネージャーとして構成されたキュー・マネージャーを開始および停止します。

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、WebSphere 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。

共有ファイル・システム

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。
- WebSphere 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。

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。

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** コマンドに応答して、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。クライアントの自動再接続機能は、WebSphere MQ classes for Java ではサポートされていません。

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確実に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します。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。

チャンネルおよびクライアントの再接続

チャン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 WebSphere MQ のチャン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ンネルは、新規のネットワーク・アドレスにある新たにアクテ

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

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。WebSphere MQ 再接続では、IP アドレスは変更も転送もされません。これはチャン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、チャン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。複数インスタンス・キュー・マネージャーへのクライアントの再接続についての詳細は、[315 ページの『クライアントの自動再接続』](#)を参照してください。クライアントは WebSphere MQ バージョン 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する必要があります。また、クライアントはバージョン 7.0.1 以上のキュー・マネージャーと接続される必要があります。

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かないキュー・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。

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・リカバリーの問題は、WebSphere MQ MQI クライアントとキュー・マネージャーの間で共有され、キュー・マネージャーに保管されているコンテキストを失うことです。WebSphere MQ MQI クライアント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 WebSphere 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、再接続可能な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、MQBACK を発行してトランザクションをロールバックすることが必要です。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、障害の前に MQCMIT が 2 フェーズ・コミットの第 2 フェーズを開始した場合、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。再接続可能な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。再接続可能な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。クライアントの自動再接続機能は、WebSphere MQ classes for Java ではサポートされていません。

WebSphere MQ MQI クライアント・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 呼び出しを発行して新しい接続を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、WebSphere 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。

WebSphere 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. WebSphere MQ MQI クライアント・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进行处理し、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入れます。MQBACK を発行し、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。

ただし、2番目の例では、アプリケーションが MQGET オプション MQGMO_MSG_UNDER_CURSOR を使用して、処理するメッセージを選択するためにキューをブラウズしているとします。再接続によってブラウズ・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、ブラウズ・カーソルを復元する必要があります。

ブラウズ・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、402 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 WebSphere MQ MQI クライアント・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、そのすべての WebSphere MQ 状態情報を廃棄し、新規の MQCONN 呼び出しまたは MQCONNX 呼び出しを発行して新規接続を確立してから、処理する必要がある WebSphere 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 で MQCNO_RECONNECT_DISABLED オプションを設定することと同等であり、フェイルオーバー後のアプリケーションの再接続を防止します。

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。イベント・ハンドラーは、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 MQRC_RECONNECTING で通

知され、再接続が成功した後に MQRC_RECONNECTED で通知されます。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。

前のセクションでは、ローカル・トランザクションを使用して MQGET と MQPUT を調整する WebSphere MQ MQI クライアントの例 401 ページの『2』を示しました。クライアントは、MQRC_BACKED_OUT エラーに回答して MQCMIT または MQBACK 呼び出しを発行してから、バックアウトされたトランザクションを再サブミットします。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 WebSphere MQ MQI クライアント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、WebSphere 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。

WebSphere MQ MQI クライアント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 WebSphere 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。

新しいエラーを受け取る例としては、再接続前にコンテキストを保存した後で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・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・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、WebSphere 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。

- 非永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。
- キュー内のブラウザ・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` を指定した `MQGET` 呼び出しは、理由コード `MQRC_NO_MSG_AVAILABLE` で失敗します。
 - 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ザ・マーク付きメッセージのマークが解除され、再度ブラウザされることがあります。
 - 多くの場合、共有ブラウザ・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。
- キュー・マネージャーにメッセージが正常に送信されたかどうかを WebSphere MQ `MQI` クライアントが認識していない場合、同期点外での `MQPUT` 呼び出し時の再接続で、`MQRC_CALL_INTERRUPTED` が返されます。 `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、WebSphere MQ `MQI` クライアントがキュー・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(異なるチャネルになる場合もあります)、チャ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となります。

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、WebSphere MQ `MQI` クライアント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・バインデ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・バインディング・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。

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、WebSphere 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、WebSphere 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。

- WebSphere 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。この構成情報は、Windows、UNIX and Linux システム上の mqs.ini ファイルのように WebSphere MQ によって管理される場合もありますし、MQSC 構成スクリプトのようにユーザー独自の構成管理の一部である場合もあります。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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere MQ クラスタによるキューの可用性の向上

1つのクラスタ・キュー(クラスタ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・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

クラスタ内のキュー・マネージャーに障害が起きた場合、クラスタ内の他のキュー・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・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内のどのメンバーにも送信できるか、などです。

クラスタ内の同じキュー・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・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を構成する可能な1つの方法は、クラスタ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、クラスタの可用性を向上させます。

クラスタは、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

クラスタは無中断サービスを提供します。クラスタ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

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。

WebSphere 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

WebSphere MQ MQI クライアント・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。クライアントの自動再接続機能は、WebSphere MQ classes for Java ではサポートされていません。

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。

メッセージの消失を確実に回避する (ログ)

WebSphere MQ は、キュー・マネージャーの障害からのリカバリーに必要なすべての情報をログに記録します。

WebSphere MQ は、キュー・マネージャーが制御する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、WebSphere MQ は、自身のファイルをホスティングするディスク・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。

WebSphere 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・エクステンションとも呼ばれます。

記録されているデータを含んでいるいくつかのログ・ファイルがあります。それらの数とサイズを (427 ページの『[IBM WebSphere MQ およびキュー・マネージャー構成情報の変更](#)』で説明されているように) 定義することもできますし、あるいはシステム・デフォルト値の 3 ファイルを取ることもできます。

WebSphere MQ for Windows では、3 ファイルそれぞれのデフォルト値は 1 MB です。WebSphere MQ for UNIX and Linux システムでは、3 ファイルそれぞれのデフォルト値は 4 MB です。

キュー・マネージャーを作成するときに、定義するログ・ファイルの数は、割り振られる 1 次 ログ・ファイルの数です。数を指定しないと、デフォルト値が使用されます。

WebSphere MQ for Windows では、ログのパスを変更していなければ、ログ・ファイルは次のディレクトリに作成されます。

```
C:\Program Files\IBM\WebSphere MQ\log\<QMGrName>
```

WebSphere MQ for UNIX and Linux システムでは、ログ・パスを変更していない場合、ログ・ファイルは次のディレクトリに作成されます。

```
/var/mqm/log/<QMGrName>
```

WebSphere MQ は、これらの 1 次ログ・ファイルで開始しますが、1 次ログ・スペースが足りなくなった場合、2 次 ログ・ファイルを割り振ります。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・ファイル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、427 ページの『[IBM WebSphere 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。

ログ制御ファイル

ログ制御ファイルには、ログ・ファイルの使用状況を制御するために必要な情報 (サイズと位置、次に使用可能なファイルの名前など) が入っています。

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。

注: キュー・マネージャーを開始するときに作成されるログが、アプリケーションで取り扱うメッセージのサイズとボリュームに十分見合った大きさになるようにしてください。デフォルトのログの数とサイズは、要件に合わせて修正する必要があるでしょう。詳しくは、411 ページの『[ログのサイズの計算](#)』を参照してください。

ログのタイプ

WebSphere MQ では、ログに必要なファイルの数は、ファイル・サイズ、受け取ったメッセージの数、およびメッセージ長によって異なります。キュー・マネージャーの記録を維持する方法には、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 WebSphere 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、[409 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・データを連続したファイルに保持します。この方法では、スペースが再利用されないため、削除されていないログ・エクステンツに記録されているレコードをいつでも取得することができます。

ディスク・スペースは限られているので、保存の形式について考慮を要する場合があります。管理者は、必要に応じて既存のスペースを再利用したり拡張したりして、ログのディスク・スペースを管理してください。

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・フローやキュー・マネージャーの持続期間によっては、非常に大きくなる可能性があります。ただし、アクティブと呼ばれているいくつかのファイルがあります。アクティブ・ファイルには、キュー・マネージャーを再始動するのに必要なログ項目が入ります。アクティブなログ・ファイル群を集合的にアクティブ・ログと呼びます。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、[411 ページの『ログのサイズの計算』](#)を参照してください。)

ログ・ファイルがアクティブと呼ばれるかそうでないかを制御する重要なイベントは、チェックポイントです。WebSphere MQ のチェックポイントは、リカバリー・ログとオブジェクト・ファイル間の整合点です。チェックポイントにより、再始動リカバリーの実行に必要なログ・ファイル・セットが決定されます。アクティブでないログ・ファイルは、再始動リカバリーに必要ではなく、非アクティブと呼ばれます。非アクティブ・ログは場合によっては、メディア回復で必要になります。(チェックポイント機能の詳細については、[409 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。)

非アクティブ・ログ・ファイルは、再始動リカバリーでは必要でないため、アーカイブ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、不要なログ・ファイルと見なすことができます。不要なログ・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。ログ・ファイルの属性指定の詳細については、[413 ページの『ログの管理』](#)を参照してください。

新しいチェックポイントが、2 番目以降の1次ログ・ファイルに記録された場合、最初のファイルを非アクティブにすることができ、新しい1次ファイルがフォーマットされて1次プールの最後に追加されます。こうしてロギングに使用可能な1次ファイルの数を回復します。このようにして、1次ログ・ファイル・プールは、絶えず拡張されるログ・ファイル・リスト中にある現行のファイル・セットのように見えます。また、管理者は操作の要件に応じて非アクティブ・ファイルを管理してください。

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ら、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、MQRC_RESOURCE_PROBLEM 戻りコードがアプリケーションに戻されます。

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

チェックポイントによって、キュー・マネージャー・データおよびログ・ファイルを同期させ、ログ・レコードを廃棄できる整合点にマークを付けます。頻繁にチェックポイントを作成すれば、より速くリカバリーできます。

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。まず、更新を表すレコードがログに書き込まれ、その後にキュー・ファイルが更新されます。つまり、ログ・ファイルの方が、キュー・ファイルよりも最新のものになります。再始動処理が確実に整合点から開始されるようにするために、WebSphere MQ はチェックポイントを使用します。チェックポイントとは、ログに記述されているレコードとキューのレコードが一致する時点(ポイント)です。チェックポイント自体は、キュー・マネージャーを再始動するのに必要な一連のログ・レコードです。例えば、チェックポイントの時点でアクティブであったすべてのトランザクション(作業単位)の状態です。

WebSphere MQ は、自動的にチェックポイントを生成します。チェックポイントは、キュー・マネージャーの開始時、シャットダウン時、ログ・スペースが不足してきたとき、および 10,000 回の操作の記録ごとにとられます。

キューが取り扱うメッセージが増えるにつれて、チェックポイント・レコードは、キューの現在の状態と一致しなくなります。

WebSphere 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・レコードは、ログとデータの間の最新の整合点を表すものです。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。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。

WebSphere 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、不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。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 が出力されます。

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。解決フェーズの経過を示すために、メッセージ AMQ7233 が出力されます。

WebSphere MQ は、ログのヘッドとテールを指し示す内部ポインターを維持します。WebSphere MQ は、ヘッドを指すポインターを、リカバリー・メッセージ・データとの整合性がある最新のチェックポイントに動かします。

チェックポイントは、リカバリーを効率よく行ったり、1次ログ・ファイルおよび2次ログ・ファイルの再利用を制御するために使用されます。

410 ページの図 69 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、WebSphere 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。

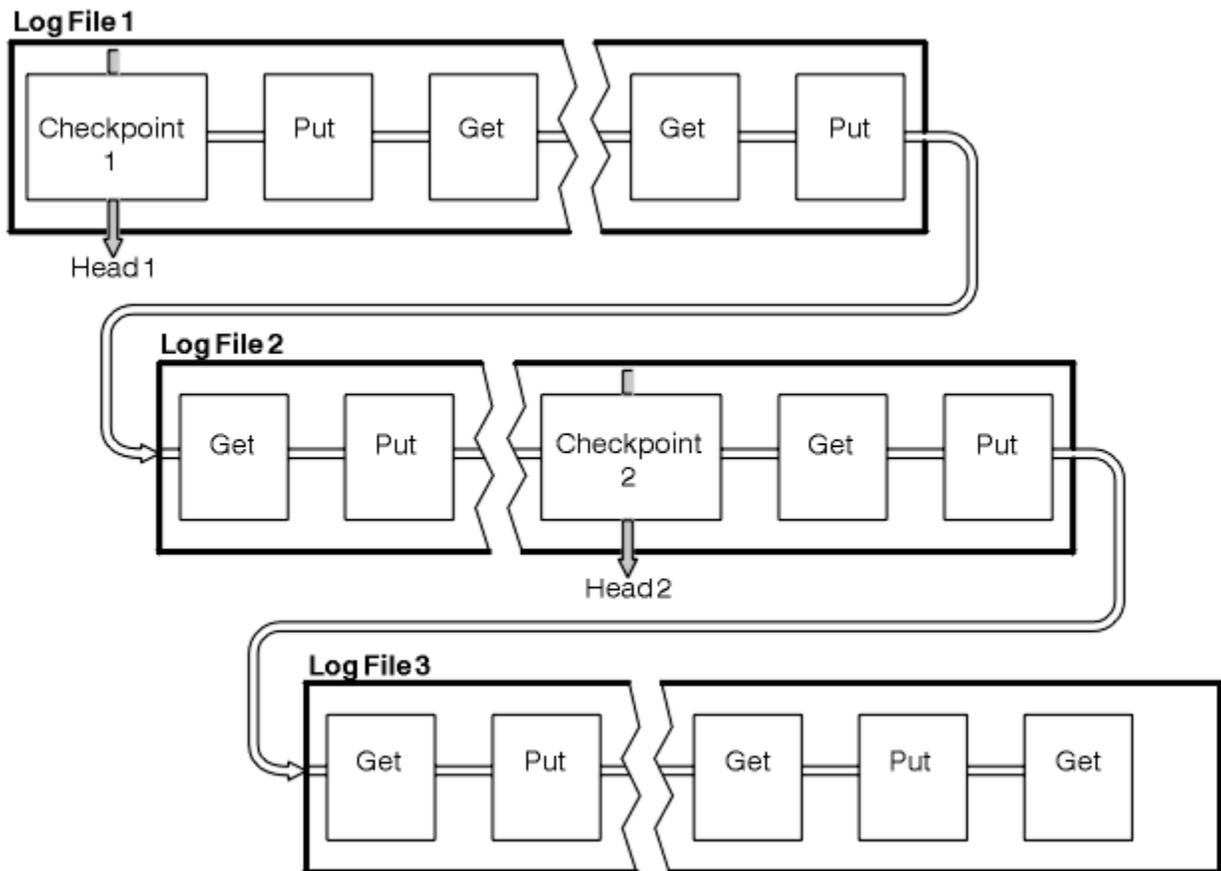


図 69. チェックポイント機能

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。

411 ページの図 70 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・ファイルがすべて記入された場合には、ログが満杯になるのを避けるために、可能ならば 2 次ログ・ファイルが使用されます。

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。

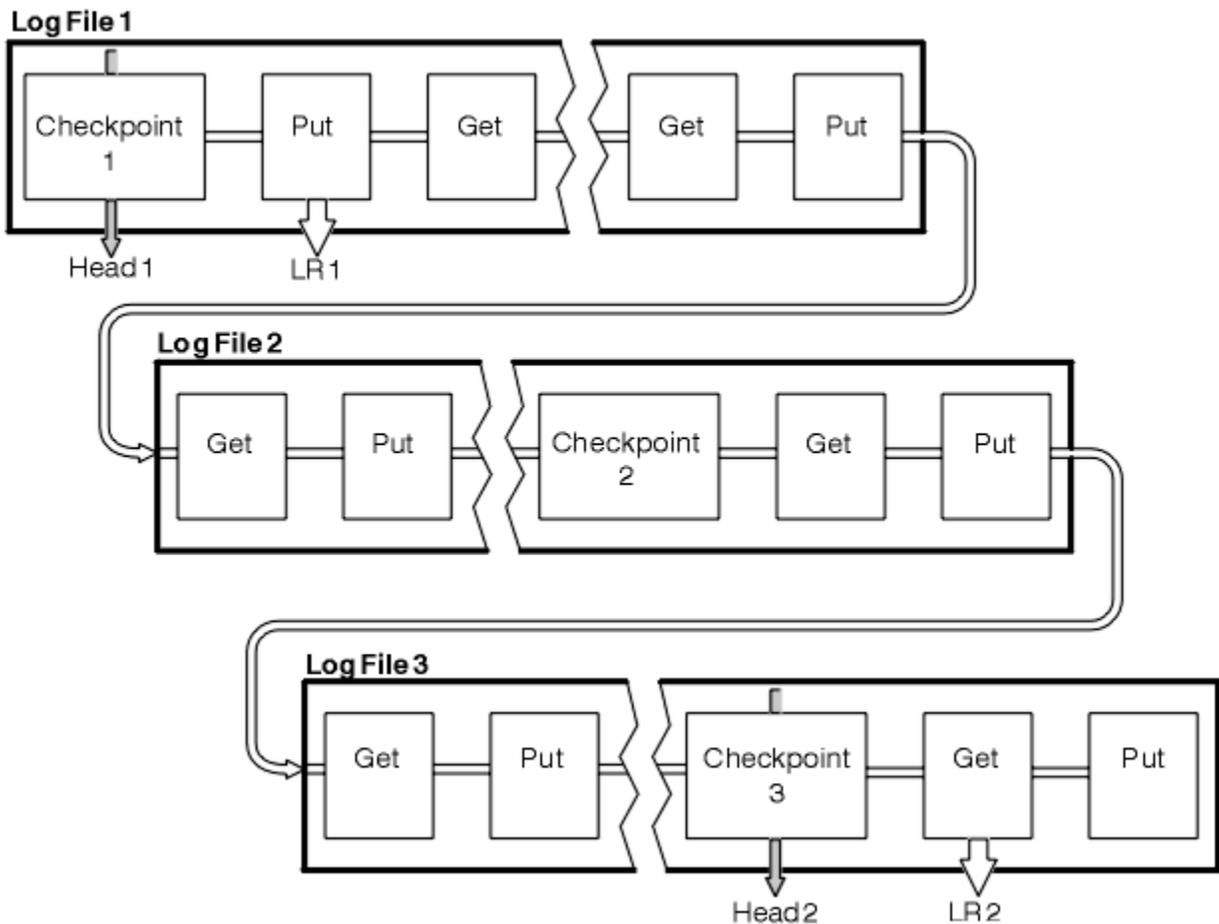


図 70. 持続期間の長いトランザクションでのチェックポイント機能

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ログのサイズを見積もる必要があります。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

412 ページの表 31 に、各種の操作の場合にキュー・マネージャーがログに記録するデータの量を示します。キュー・マネージャーが実行するほとんどの操作では、最少量のログ・スペースしか必要ありません。しかし、持続メッセージをキューに入れる場合は、メッセージのリカバリーを可能にするために、すべてのメッセージ・データをログに書き込む必要があります。ログのサイズは、通常、キュー・マネージャーが取り扱う必要がある持続メッセージの数とサイズによって決まります。

表 31. ログ項目のサイズ (すべて概算値)

Operation	サイズ
持続メッセージの書き込み	750 バイト + メッセージ長 大きいメッセージは 261844 バイトのセグメントに分割され、セグメントごとに 300 バイトが追加されます。
メッセージの読み取り	260 バイト
同期点、コミット	750 バイト
同期点、ロールバック	1000 バイト + 12 バイト (ロールバックされる読み取りまたは書き込みごとに)
オブジェクトの作成	1500 バイト
オブジェクトの削除	300 バイト
属性の変更	1024 バイト
メディア・イメージの記録	800 バイト + イメージ イメージは 260 000 バイトのセグメントに分割され、それぞれのセグメントに 300 バイトが余分に追加されます。
チェックポイント	750 バイト + 200 バイト (各アクティブ作業単位ごとに) + 380 バイト (キュー・マネージャーごとに複数のクラスター伝送キューを使用している場合、各クラスター送信側チャンネルごとに)。 パフォーマンス上の理由でバッファーに入れられている、コミットされていない書き込みまたは読み取りについて、さらに他のデータがログに記録される可能性があります。 クラスター送信側チャンネルがある場合は、各チェックポイントで、クラスター送信側チャンネル当たり 380 バイトが余分にログに書き込まれます。

注:

- 1 次ログ・ファイルと 2 次ログ・ファイルの数は、キュー・マネージャーが始動するたびに変更することができます。
- 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 次ログ・スペースを再利用します。これは、キュー・マネージャーがログに記録する必要がある量として見積もったデータ量より、キュー・マネージャーのログの方が小さくなる可能性があることを意味します。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。
- 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。

定義した1次ログ・ファイルの数が2次ログ・ファイルより少なかった場合は、チェックポイントがとられる前にキュー・マネージャーが2次ログ・ファイルを割り振る可能性があります。これで、1次ログ・ファイルの再利用ができる状態になります。

ログの管理

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管理が必要になることがあります。

時間の経過と共に、作成されたログ・レコードの一部はキュー・マネージャーの再始動には不要になります。循環ロギングを使用している場合、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。

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。チェックポイントが長い間取られなかったり、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は、ログが満杯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。

持続期間の長いトランザクションがログを満杯にしてしまうと、ログ・レコードを書き込む試みは失敗し、一部のMQI呼び出しはMQRC_RESOURCE_PROBLEMを戻します。(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ため、MQCMITまたはMQBACKは失敗しません。)

キュー・マネージャーは、過度にログ・スペースを消費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。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。

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。

最後に、ログが処理できる以上の速度でレコードがログに書き込まれていると、メッセージAMQ7466が出されます。このメッセージが出されたら、ログ・ファイルの数を増やすか、あるいはキュー・マネージャーが処理するデータの量を減らしてください。

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AMQ6708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・レコードが書き込まれるにつれて拡張されるのではなく、最大のサイズのものが作成されます。これは、WebSphere 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。WebSphere 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。

ログ・ファイルを含んでいるドライブが満杯になった場合には、一部のディスク・スペースを解放することができます。リニア・ログを使用している場合は、別のドライブまたは装置にコピーできる非アクティブ・ログ・ファイルがログ・ディレクトリーにいくつかある場合があります。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らないように、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・ファイルが同じサイズであると想定しています。

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(435 ページの『IBM WebSphere MQ のログのデフォルト』および 443 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。以下に示すのは、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です。

ロガー・イベント・メッセージ

使用可能にした場合、キュー・マネージャーが新規ログ・ファイルへのログ・レコードの書き込みを開始すると、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 AMQ7467 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 AMQ7468 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。

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。

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。

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 AMQ6767 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。

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・ディレクトリー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。キュー・マネージャーですべてのオブジェクトのメディア・イメージを取るには、以下の例で示されているように、**rcdmqing** コマンドを実行します。

Windows の場合

```
rcdmqing -m QMNAME -t all *
```

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、418 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。

不要なログ・ファイルの判別

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。

キュー・マネージャーを再始動するのに必要な最も古いログ・ファイルを判別するには、コマンド **DISPLAY QMSTATUS RECLOG** を発行します。

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・ファイルを判別するには、コマンド **DISPLAY QMSTATUS MEDIALOG** を発行します。

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。

ただし、ログ・ファイルの回転率が非常に高い場合、あるいは一般的なケースに確実に対処しておきたい場合、通常は以下のアルゴリズムを使用できます。

```
Let S == 再始動ログ・ファイル番号  
          (DISPLAY QMSTATUS RECLOG から)。  
Let M == メディア・リカバリー・ログ・ファイル番号  
          (DISPLAY QMSTATUS MEDIALOG から)。  
Let L == 削除またはアーカイブの対象となるログ・ファイル番号  
          決定する必要があります。
```

```
function minlog (a, b) {  
  if (abs (a - b) < 5000000)  
    return min (a, b); # Not wrapped.  
  else  
    return max (a, b); # 折り返し。}
```

ログ・ファイル L は、以下の場合に削除できます。
(L! = S && L! = M && minlog (L, minlog (S, M)) == L)。

ログ・ファイル L は、以下の場合にアーカイブできます。
(L! = S && minlog (L, S) == L)。

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 WebSphere 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の少なくとも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・マネージャーのデータとは別個のディスク・ドライブに置くのが理想的です。これは、パフォーマンスの面での利点があります。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、WebSphere 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

ログを使用して、障害からリカバリーします。

データが損傷する原因は、いくつか考えられます。WebSphere MQ を利用すると次のような状況からリカバリーするのに役立ちます。

- 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

WebSphere 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。さらに、場合によっては、不注意によるファイルの削除といった、その他の種類の障害からもリカバリーすることができます。

通信障害の場合、メッセージは、受信側のアプリケーションがそれらを取り除く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、WebSphere 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、WebSphere 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 WebSphere 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

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があります。

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 WebSphere MQ オブジェクトのメディア・イメージを記録します。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

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トランザクショ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する必要があります。このコマンドは、IBM WebSphere 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 WebSphere MQ オブジェクトが損傷した場合、実行できる有効なアクションは、そのオブジェクトを削除するか、このメソッドでオブジェクトを再作成するかのいずれかのみです。非持続メッセージは、今述べた方法ではリカバリできません。

`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・リカバリを試みる際には、メディア・リカバリ・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にある状態で、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 が出力され、メディア・リカバリ操作は失敗します。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。

始動時に損傷オブジェクトをリカバリする

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 WebSphere 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ています。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 WebSphere 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、損傷が通知されません。

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)。

IBM WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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` 出力の `nextlsn` を指定できます。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・エクステントからログ・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、Log File Header 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。

<code>logactive</code>	1 次ログ・エクステントの数
<code>loginactive</code>	2 次ログ・エクステントの数
ログ・サイズ	エクステントあたり 4 KB ページ数
ベースライン (<code>baselsn</code>)	ログのヘッドを含むログ・エクステントの中の最初の LSN
<code>nextlsn</code>	書き込まれる次のログ・レコードの LSN
ヘッドレス	ログのヘッドでのログ・レコードの LSN
<code>tailsn</code>	ログのテール位置を識別する LSN
<code>hflag1</code>	ログが CIRCULAR であるか LOG RETAIN (リニア) であるかどうか
<code>HeadExtentID</code>	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。

<code>LSN</code>	ログの順序番号
<code>LogRecd</code> タイプ	ログ・レコードのタイプ
「 <code>XTranid</code> 」	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 <code>TranType</code> は WebSphere MQ トランザクションのみを示しています。XA の <code>TranType</code> 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 <code>XTranid</code> があります。
<code>QueueName</code>	このログ・レコード (ある場合) と関連したキュー
<code>QID</code>	キュー用の固有の内部 ID
<code>PrevLSN</code>	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。

<i>StartDate</i>	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。

<i>StopDate</i>	キュー・マネージャーを停止した日付
<i>StopTime</i>	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。

<i>ChkPtLSN</i>	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 nul 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。

<i>MapIndex</i>	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 <i>MQGET</i> を突き合わせる場合に使用できます。この場合、同じ <i>QueueName</i> と <i>MapIndex</i> を含む後続の <i>Get Message</i> ログ・レコードを見つけることができます。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 <i>MapIndex</i> ID を再使用できます。
データ	このログ・レコードに 16 進ダンプで含まれているものは、メッセージ記述子 (目印 MD) およびメッセージ・データ自体が後に続く内部データです。

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。*Put Part* レコードがある場合、*PrevLSN* フィールドは、*Put Part* レコードと最終 *Put Message* レコードをチェーニングします。

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 nul 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。

<i>MapIndex</i>	キューから得られたメッセージを識別します。同じ <i>QueueName</i> および <i>MapIndex</i> を含む最新の <i>Put Message</i> ログ・レコードは、取得されたメッセージを識別します。
「 <i>QPriority</i> 」	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 WebSphere 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>	トランザクションと関連した最初のログ・レコードの LSN
<i>LastLSN</i>	トランザクションと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>RM</i> 名	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、リソース・マネージャーが参加しているグローバル・トランザクションを記録する後続の <i>Transaction Prepared</i> ログ・レコードにも記録されます。
<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与している場合は、*RMID* がゼロの *Participant Entry* が存在します。

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。コミット決定が各参加者に送信されると、*Transaction Prepared* ログ・レコードの後に続きます。

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、WebSphere 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
----------------	---------------

UserId

作成を実行するユーザー ID

オブジェクトの削除

これは、WebSphere MQ オブジェクトの削除を記録します。

ObjName

削除したオブジェクトの名前

IBM WebSphere MQ キュー・マネージャー・データのバックアップと復元

キュー・マネージャーとキュー・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・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。

- [423 ページの『キュー・マネージャー・データのバックアップ』](#)。
- [423 ページの『キュー・マネージャー・データの復元』](#)。

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。

バックアップ・キュー・マネージャーを作成およびアクティブ化するには、以下を参照してください。

- [424 ページの『バックアップ・キュー・マネージャーの作成』](#)。
- [426 ページの『バックアップ・キュー・マネージャーの開始』](#)。

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。

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。

- [426 ページの『キュー・マネージャー構成のバックアップ』](#)
- [426 ページの『キュー・マネージャー構成の復元』](#)

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・マネージャーが実行中でないことを確認します。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た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。

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、[427 ページの『IBM WebSphere 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: ディレクトリーに表示されている名前を理解するのが困難な場合があります。使用している WebSphere MQ のプラットフォームとの互換性を持たせるために、名前が変換されているためです。名前の変換について詳しくは、[WebSphere 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、[407 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、[70 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。

3. ファイルの所有権も存続させます。WebSphere MQ for UNIX and 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: WebSphere MQ バージョン 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 WebSphere MQ に戻すために使用できます。

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

キュー・マネージャーが実行中でないことを確認します。

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を復元するには、次の作業が必要です。

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。
4. 構成情報ファイルを更新します。

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 WebSphere MQ のディレクトリーとサブディレクトリーについて詳しくは、[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。

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 WebSphere 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、WebSphere 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。

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。

注：WebSphere MQ バージョン 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 WebSphere MQ に戻すために使用できます。

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。

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・マネージャーを保持することができます。

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・マネージャーをアクティブ化した後は、更新できなくなります。

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。

- [424 ページの『バックアップ・キュー・マネージャーの作成』](#)
- [425 ページの『バックアップ・キュー・マネージャーの更新』](#)
- [426 ページの『バックアップ・キュー・マネージャーの開始』](#)

バックアップ・キュー・マネージャーの作成

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。

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するには、以下の手順を実行します。

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。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。
2. 423 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. バックアップ・キュー・マネージャー上で次の制御コマンドを実行します。

```
strmqm -i BackupQMName
```

これにより、キュー・マネージャーが WebSphere 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・エクステントを再生して、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。

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。

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ために、キュー・マネージャーを停止する必要はありません。

バックアップ・キュー・マネージャーを更新するには、以下の手順を実行します。

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、現行キュー・マネージャーのログ・ディレクトリーからバックアップ・キュー・マネージャーのログ・ディレクトリーにコピーします。最後の更新以降のすべてのログ・エクステントを、ステップ 2 でメモした現行エクステントまで(ただし、そのエクステントは含まない)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. バックアップ・キュー・マネージャー上で次の制御コマンドを発行します。

```
strmqm -i 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。

警告: non-contiguous のログ・セットをバックアップ・キュー・マネージャーのログ・ディレクトリーにコピーすると、最初に欠落しているログが見つかった時点までのログのみが再生されます。

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。

これを行うには、以下のステップを実行します。

1. バックアップ・キュー・マネージャーを活動化する次の制御コマンドを実行します。

```
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。

2. バックアップ・キュー・マネージャーを始動する次の制御コマンドを実行します。

```
strmqm BackupQMName
```

WebSphere 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。

バックアップ・キュー・マネージャーの代わりにリカバリー不能のキュー・マネージャーが使用された場合、リカバリー不能のキュー・マネージャーのキュー・マネージャー・データの一部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。

3. 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。

WebSphere MQ ディレクトリーおよびサブディレクトリーについて詳しくは、[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、WebSphere 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、WebSphere 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のかも確認してください。

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。

注: キュー・マネージャーのデータおよびログ・ファイルが異なるディレクトリーに保持されているとしても、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。たとえ始動したとしても、データはほぼ間違いなく破壊されます。

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、その定義から容易にキュー・マネージャーを再作成できます。

キュー・マネージャーの構成のバックアップ・コピーを取るには、次の作業が必要です。

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. a. AIX、HP-UX、Linux、Solaris、または Windows の場合: デフォルトのフォーマット設定オプション (-f mqsc) MQSC およびすべての属性 (-a) を使用して Dump MQ Configuration コマンド (dmpmqcfg) を実行し、標準出力リダイレクトを使用して定義をファイルに保管します。以下に例を示します。

```
dmpmqcfg -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
```

キュー・マネージャー構成の復元

キュー・マネージャーの構成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。

キュー・マネージャーの構成のバックアップを復元するには、次の作業が必要です。

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、その他の方法では復旧不能だった場合、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実行します。
 - a. AIX、HP-UX、Linux、Solaris、または Windows の場合: キュー・マネージャーに対して `runmqsc` を実行し、標準入力リダイレクトを使用して、MQ 構成のダンプ (`dmpmqcfg`) 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。以下に例を示します。

```
runmqsc MYQMGR < /mq/backups/MYQMGR.mqsc
```

関連資料

[dmpmqcfg](#)

IBM WebSphere MQ およびキュー・マネージャー構成情報の変更

IBM WebSphere 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します。

IBM WebSphere MQ 構成情報を変更するには、IBM WebSphere 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 WebSphere MQ 構成ファイルを編集して、属性情報を変更します。IBM WebSphere MQ for Windows および Linux (x86 および x86-64 プラットフォーム) では、IBM WebSphere MQ 構成ファイルは、IBM WebSphere MQ Explorer を使用して編集できます。

`amqmdain` で説明されているように、Windows システムでは、`amqmdain` を使用して構成情報を変更することもできます。

ご使用のプラットフォームでの IBM WebSphere MQ およびキュー・マネージャーの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

[5 ページの『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

[計画](#)

[WebSphere MQ の管理](#)

UNIX, Linux, and Windows システムでの構成情報の変更

構成属性は、キュー・マネージャーのノードのレベルで構成ファイルに保持されます。

Windows、UNIX and Linux プラットフォームでは、以下の中にある IBM WebSphere MQ 構成属性を変更することができます。

- ノード全体で IBM WebSphere MQ の変更を有効にするための IBM WebSphere MQ 構成ファイル (`mqqs.ini`)。1 ノードに 1 つの `mqqs.ini` ファイルがあります。
- 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。

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内に個別に保持されます。

構成ファイル (または **スタンザ**・ファイル) には、1 つ以上のスタンザが含まれています。スタンザは、`.ini` ファイル内の行のグループで、全体で 1 つの機能を表しているか、ログ機能、チャネル機能、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 WebSphere 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 WebSphere 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。

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。

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 WebSphere MQ Explorer から構成情報を編集できます。

Windows システムでは、amqmdain コマンドを使用して、構成ファイルを編集することもできます。

Windows、UNIX and Linux システムでの構成オプションについての詳細は、以下のサブトピックを参照してください。

関連概念

5 ページの『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。

427 ページの『IBM WebSphere MQ およびキュー・マネージャー構成情報の変更』

IBM WebSphere 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します。

関連タスク

計画

WebSphere MQ の管理

関連資料

433 ページの『IBM WebSphere MQ 構成情報を変更するための属性』

IBM WebSphere MQ for Windows システムおよび IBM WebSphere MQ for Linux (x86 および x86-64 プラットフォーム) システムでは、IBM WebSphere 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、mq.ini 構成ファイルを編集して構成情報を変更します。

440 ページの『キュー・マネージャー構成情報の変更』

ここで説明する属性は、個々のキュー・マネージャーの構成を変更します。これらの属性は、WebSphere MQ の設定値を変更します。

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。

WebSphere 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。

- WebSphere 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。

IBM WebSphere MQ 構成ファイル mqs.ini

IBM WebSphere 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。

IBM WebSphere MQ for UNIX and Linux 製品では、データ・ディレクトリーとログ・ディレクトリーは常に /var/mqm および /var/mqm/log です。

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。

さらに、Windows システムでは、Windows 上に mqinst.ini ファイルがないため、インストール構成情報 (UNIX and Linux システムの IBM WebSphere MQ 上の mqinst.ini に含まれています) がレジストリー内にあります。

Windows システムの mqs.ini ファイルは、HKLM\SOFTWARE\IBM\WepSphere 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 IBM WebSphere 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、430 ページの図 71 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。

```

#*****#
#* Module Name: mqs.ini                                     *#
#* Type       : WebSphere MQ Machine-wide Configuration File *#
#* Function   : Define WebSphere 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 71. UNIX システム用の IBM WebSphere MQ 構成ファイルの例

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。

V7.5.0.9 IBM WebSphere MQ Version 7.5.0、フィックスパック 9 以降の場合、**strmqm** コマンドによって、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 スタンザと SSL スタンザの構文が検査されます。これにより、**strmqm** は、qm.ini ファイルにエラーが含まれていることを検出した場合に、問題を容易に判別して迅速に修正できるようになります。詳しくは、[strmqm](#) を参照してください。

qm.ini ファイルの場所

Windows UNIX Linux

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 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、次のとおりです。

```
C:\Program Files\IBM\WebSphere 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、[WebSphere 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

UNIX Linux

以下の例は、UNIX and Linux システムの IBM WebSphere MQ 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 : WebSphere 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=13

ServiceComponent:
  Service=AuthorizationService
  Name=MQSeries.UNIX.auth.service
  Module=opt/mqm/bin/amqzfu
  ComponentDataSize=0

Log:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=01
  LogPath=/var/mqm/log/saturn!queue!manager/

AccessMode:
  SecurityGroup=wmq\wmq

XAResourceManager:
  Name=DB2 Resource Manager Bank
  SwitchFile=/usr/bin/db2swit
  XAOpenString=MQBankDB
  XACloseString=
  ThreadOfControl=THREAD
```

```
Channels: 2
  MaxChannels=200
  MaxActiveChannels=100
  MQIBindType=STANDARD

AccessMode:
  SecurityGroup=wmq\wmq
TCP:
  KeepAlive = Yes
  SvrSndBuffSize=32768
  SvrRcvBuffSize=32768
  Connect_Timeout=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anzas の詳細については、[70 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anzas 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anzas のみを使用してください。

Unix では、WebSphere MQ はファイル拡張子が .ini の構成ファイル (例: qm.ini) を使用します。WebSphere MQ には、このようなファイルの一時的なバックアップ・コピーを作成するユーティリティー (**setmqm** など) がいくつかあります。例えば、ファイル qm.ini については、qm.ini.bak というバックアップ・コピーが作成されます。ユーティリティーは、qm.ini ファイルを変更し、更新したファイルを保管してから、qm.ini.bak ファイルを削除します。ユーティリティーは、qm.ini ファイルを保管できない場合、qm.ini の内容をバックアップ・ファイル qm.ini.bak から復元し、qm.ini.bak ファイルを削除します。

既存の qm.ini.bak ファイルがある場合、ユーティリティーは、その qm.ini.bak の内容を使用して qm.ini ファイルを元に戻し、qm.ini.bak ファイルを削除します。したがって、*.ini ファイルのバックアップ・コピーを、.bak ファイル拡張子を使用して作成すべきではありません。そのようなバックアップ・ファイルは、WebSphere MQ ユーティリティーによって削除される可能性があるからです。

変更が有効になるタイミングについては、[427 ページの『UNIX, Linux, and Windows システムでの構成情報の変更』](#)を参照してください。

インストール構成ファイル、mqinst.ini

UNIX and Linux システム

インストール構成ファイル mqinst.ini には、UNIX または Linux システム上のすべての IBM WebSphere MQ インストールに関する情報が含まれています。

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

この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。その代わりに、以下のコマンドを使用して、mqinst.ini ファイル内の値を作成、削除、照会、および変更します。

[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。

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。

Windows システム

Windows システムでは、インストール構成情報は以下のキーで保持されます。

```
HKLM\SOFTWARE\IBM\WebSphere MQ\Installation\<InstallationName>
```

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、レジストリー内の値を照会および変更します。

[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。

IBM WebSphere MQ 構成情報を変更するための属性

IBM WebSphere MQ for Windows システムおよび IBM WebSphere MQ for Linux (x86 および x86-64 プラットフォーム) システムでは、IBM WebSphere 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、mqcs.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『[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。

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

IBM WebSphere 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します。

関連タスク

[計画](#)

[WebSphere MQ の管理](#)

関連資料

440 ページの『[キュー・マネージャー構成情報の変更](#)』

ここで説明する属性は、個々のキュー・マネージャーの構成を変更します。これらの属性は、WebSphere MQ の設定値を変更します。

すべてのキュー・マネージャー

IBM WebSphere MQ エクスプローラーの General および Extended WebSphere MQ プロパティ・ページ、または mqcs.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、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 WebSphere MQ を停止してください。WebSphere 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。

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、WebSphere 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。

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、[434 ページの表 32](#) 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。

デフォルト・キュー・マネージャー

IBM WebSphere MQ エクスプローラーの General WebSphere MQ プロパティ・ページ、または mqs.ini ファイルの DefaultQueueManager スタンザを使用して、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 WebSphere MQ エクスプローラーの Extended IBM WebSphere MQ プロパティ・ページ、または mqs.ini ファイルの ExitProperties スタンザを使用して、キュー・マネージャー出口プログラムで使用す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) 出口を使用すると、MQI 呼び出し (例えば、MQOPEN、MQPUT) に応答して、クラスター内のどのクラスター・キューを開くかを指定できます。CLWL 出口は、CLWLMode 属性に指定した値に応じて、FAST モードまたは SAFE モードのいずれかで実行されます。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 WebSphere MQ のログのデフォルト

IBM WebSphere MQ Explorer の Default log settings IBM WebSphere MQ プロパティ・ページ、または mqs.ini ファイルの LogDefaults スタンザを使用して、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。

スタンザが存在しない場合、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、[443 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 (433 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 (443 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、411 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、WebSphere 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 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

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、408 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、408 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。キュー・マネージャーを作成した後にロギングの方法を変更することはできません。

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。WebSphere MQ バージョン 7.0 の場合、512 (2048 KB) です。

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 WebSphere MQ for Windows。ここで、<DefaultPrefix> は、All Queue Managers WebSphere MQ プロパティ・ページの DefaultPrefix 属性に指定された値です。この値はインストール時に設定されます。
- /var/mqm/log (WebSphere MQ for UNIX and Linux システム)

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となります。

crtmqm コマンドで -ld を指定しなかった場合は、mqqs.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。

LogWrite 健全性 =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、WebSphere MQ リカバリー・ログをホストするファイル・システムまたはデバイスで最小単位 4 KB の書き込みが明確に保証され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。

Advanced Configuration and Power Interface (ACPI)

IBM WebSphere MQ エクスプローラーの ACPI WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere MQ Explorer から Exits IBM WebSphere MQ プロパティ・ページを使用するか、mq.ini ファイル内の ApiExitTemplate スタンザおよび 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、453 ページの『API 出口』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、[WebSphere 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 WebSphere 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・マネージャーがバージョン 7.1 より前の WebSphere MQ インストールに関連付けられます。

関連概念

17 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。

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**、**dspmqa**、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。

キュー・マネージャー構成情報の変更

ここで説明する属性は、個々のキュー・マネージャーの構成を変更します。これらの属性は、WebSphere MQ の設定値を変更します。

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anzas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icolon (;) またはハッシュ文字 (#) を使用できます。

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 WebSphere 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 WebSphere 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、Windows では **regedit** を使用し、UNIX and Linux では qm.ini ファイルを編集して、インストール可能サービスに変更を加える必要があります。

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『[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。

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

IBM WebSphere 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します。

関連タスク

計画

[WebSphere MQ の管理](#)

関連資料

433 ページの『[IBM WebSphere MQ 構成情報を変更するための属性](#)』

IBM WebSphere MQ for Windows システムおよび IBM WebSphere MQ for Linux (x86 および x86-64 プラットフォーム) システムでは、IBM WebSphere 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、mqsc.ini 構成ファイルを編集して構成情報を変更します。

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anzasは、**crtmqm** コマンドの **-a [r]** オプションによって設定されます。キュー・マネージャーの作成後には、AccessMode スタanzas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 \¥ 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` グループにはキュー・マネージャー・データ・ファイルへのアクセス権限が付与されません。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・セキュリティー・グループです。グローバル・セキュリティー・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ー・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、追加のセキュリティー・アクセス・グループを、`mqm` という名前のローカル・グループを使用する代わりに使用することもできます。ローカル・グループ `mqm` とは異なり、追加のセキュリティー・アクセス・グループをローカル・グループまたはグローバル・グループに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、Windows システムによって検査されるユーザー ID は、IBM WebSphere MQ 準備ウィザードで構成したユーザー ID です。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には、ユーザー ID がローカル `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

[382 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』](#)

[378 ページの『Windows でキュー・マネージャーの共有のデータとログのディレクトリーおよびファイルを保護する』](#)

関連タスク

[353 ページの『ドメイン・ワークステーションまたは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』](#)

関連資料

[crtmqm](#)

インストール可能サービス

Windows では `regedit` を使用して、UNIX and Linux では `qm.ini` ファイルの `Service` スタンザを使用して、インストール可能サービスを変更します。

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 WebSphere MQ エクスプローラーでは、インストール可能サービスを読み取り専用としています。

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` を使用するか、UNIX and Linux システムでは `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

WebSphere 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。

SecurityPolicy=Default|NTSIDsRequired (WebSphere MQ for Windows のみ)

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ー・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。

Default

有効にするデフォルトのセキュリティー・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ー・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

セキュリティー検査を行うときに、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。

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。

Default

UserIdentifier フィールドの値は、実ユーザー ID として設定されます。

実数

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。

Default

UserIdentifier フィールドの値は、実ユーザー ID として設定されます。

実数

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。

Default

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。

実数

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。

インストール可能サービスおよびコンポーネントについて詳しくは、[UNIX、Linux、および Windows のインストール可能サービスおよびコンポーネント](#)を参照してください。

セキュリティ・サービス全般について詳しくは、[Windows、UNIX and Linux システムでのセキュリティのセットアップ](#)を参照してください。

関連資料

[インストール可能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 WebSphere 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。

インストール可能サービスおよびコンポーネントについて詳しくは、[UNIX、Linux、および Windows のインストール可能サービスおよびコンポーネント](#)を参照してください。

キュー・マネージャーのログ

IBM WebSphere MQ エクスプローラーの Log 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 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(435 ページの『[IBM WebSphere 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、411 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、WebSphere 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、435 ページの『[IBM WebSphere 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、408 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、408 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。WebSphere MQ バージョン 7.0 の場合、512 (2048 KB) です。

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。

- 「WebSphere MQ for Windows」の「C:\Program Files\IBM\WebSphere MQ\log」。
- /var/mqm/log (WebSphere MQ for UNIX and 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 crtmqm 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。

crtmqm 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。

WebSphere MQ for UNIX and Linux システムでは、ユーザー ID mqm およびグループ mqm はログ・ファイルについての全権限を持っている必要があります。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。

LogWrite 健全性 =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、WebSphere MQ リカバリー・ログをホストするファイル・システムまたはデバイスで最小単位 4 KB の書き込みが明確に保証され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。

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 WebSphere 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 WebSphere 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・グループに含まれていないユーザーが発行した MQCONN または MQCONNX 呼び出しは、理由コード MQRC_Q_MGR_NOT_AVAILABLE で失敗します。

制限モードは、IBM WebSphere MQ 許可サービスで作動します。そのため、ユーザーには、IBM WebSphere MQ に接続し、IBM WebSphere 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。

 IBM WebSphere 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](#)を参照してください。

IBM WebSphere 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のみ使用してください。

XA リソース・マネージャー

IBM WebSphere MQ エクスプローラーの XA resource manager 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 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tring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。

WebSphere 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へロードされます。

64 ビット・ファイルは、ExitsDefaultPath64 で指定されたパスからキュー・マネージャーにロードされます。

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 String。この String 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®](#)

また、適切な String 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 エントリー・ポイントに渡されるデータの String。この String 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 String 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

WebSphere MQ for Windows の場合は、これは必須属性です。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 WebSphere 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 WebSphere 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。

チャンネル・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。

この情報は、z/OS プラットフォーム用の WebSphere MQ には適用されません。

WebSphere MQ エクスプローラーの 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。

値は 1 から 65535 の範囲でなければなりません。デフォルトは 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

WebSphere 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場合もあります。AdoptNewMCA 属性を使用すると、この方法で終了できるチャンネルのタイプを制御できます。

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、WebSphere 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: リクエスター・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。これは、適切なチャンネル・タイプも指定されている場合にものみ発生します (例: AdoptNewMCA=RCVR, SVR, FASTPATH)。

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 テークオーバーを行う 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・ balancer 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。

関連概念

56 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。

TCP、LU62、NETBIOS、SPX

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、チャンネルのデフォルトの属性を指定変更します。

TCP

IBM WebSphere MQ エクスプローラーの 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。WebSphere MQ の事前割り当てポート番号は 1414 です。

Library1 =DLLName1 (WebSphere 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、チャン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、[450 ページの表 33](#) に示すとおりです。

プラットフォーム	ListenerBacklog のデフォルト値
Windows サーバー	100
Windows ワークステーション	5
Linux	100
Solaris	100
HP-UX	20
AIX V4.2 以降	100
AIX V4.1 以前	10

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 [450 ページの表 33](#) 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。

SvrSndBuffSize=32768|number

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)。

SvrRcvBuffSize=32768|number

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)。

Connect_Timeout=0|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。

LU62 (WebSphere MQ for Windows のみ)

IBM WebSphere MQ エクスプローラーの 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。

NETBIOS (WebSphere MQ for Windows のみ)

IBM WebSphere MQ エクスプローラーの 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。

SPX (WebSphere 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 WebSphere MQ エクスプローラーの 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、[452 ページの表 34](#) に示すとおりです。

プラットフォーム	ListenerBacklog のデフォルト値
Windows サーバー	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 [452 ページの表 34](#) 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 WebSphere MQ エクスプローラーの Exits 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 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 WebSphere 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、`mqlclient.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、438 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。

QMErrorLog スタンザ (UNIX, Linux, and Windows)

WebSphere MQ エクスプローラーの 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 `QMErrorLog` スタンザを使用して、キュー・マネージャー・エラー・ログの操作と内容を調整します。



重要: 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にのみ、WebSphere MQ エクスプローラーを使用して変更を行うことができます。

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 2097152 バイト (2 MB) が使用されます。

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。WebSphere MQ が、多くのチャンネルの停止および開始により、頻繁に使用される場合は、大量の通知メッセージが 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。WebSphere 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)
- 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)
- 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。WebSphere MQ が、多くのチャンネルの停止および開始により、頻繁に使用される場合は、大量の通知メッセージが 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。WebSphere 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、メッセージを抑制すると、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)
- 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)
- 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。

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。

キュー・マネージャーのデフォルトのバインド・タイプ

IBM WebSphere MQ エクスプローラーの 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anzas を使用して、デフォルトのバインド・タイプを指定します。

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。

キュー・マネージャー構成ファイルの SSL スタンザおよび TLS 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 SSL チャネルまたは TLS チャネルを構成します。

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 チャネルまたは TLS チャネルが 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 チャネルまたは TLS チャネルが 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 SSL チャネルまたは TLS チャ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 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: SSL チャネルまたは TLS チャネルは、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: SSL チャネルまたは TLS チャネルは、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、SSL または TLS のチャネル・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 SSL 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: SSL 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: SSL 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。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。

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 WebSphere 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 (435 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) 出口を使用すると、MQI 呼び出し (例えば、`MQOPEN`、`MQPUT`) に応答して、クラスター内のどのクラスター・キューを開くかを指定できます。CLWL 出口は、`CLWLMode` 属性に指定した値に応じて、FAST モードまたは SAFE モードのいずれかで実行されます。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

このスタンプは、WebSphere MQ によって作成されます。変更はしないでください。

スタンプ Subpool およびスタンプ内の属性 ShortSubpoolName は、キュー・マネージャーを作成するときに、WebSphere MQ によって自動的に作成されます。WebSphere 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、WebSphere MQ が実行中のプロセス間の内部通信に使用します。

構成 HP Integrity NonStop Server

この情報は、IBM WebSphere MQ Client for HP Integrity NonStop Server インストール済み環境を構成するのに役立ちます。

構成ファイルを使用したクライアントの構成について詳しくは、[127 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。

環境変数を使用したクライアントの構成について詳しくは、[144 ページの『WebSphere MQ 環境変数の使用』](#)を参照してください。

TMF/Gateway の下で IBM WebSphere MQ client for HP Integrity NonStop Server 操作を実行する場合、TMF/Gateway の構成方法についてサブトピックを参照してください。Gateway プロセスの概要、Gateway を Pathway のもとで動作するように構成する方法、および IBM WebSphere MQ Client for HP Integrity NonStop Server が TMF Gateway に到達できるようにクライアント初期設定ファイルを構成する方法が記載されています。

このセクションには、チャンネルへのアクセス権付与に関する IBM WebSphere MQ Client for HP Integrity NonStop Server 固有の情報も含まれています。

Gateway プロセスの概要

HP NonStop Transaction Management Facility (TMF) は、ゲートウェイ・プロセスをリソース・マネージャーとして登録できるようにするためのサービスを提供します。IBM WebSphere MQ が提供する TMF/Gateway プロセスは、Pathway のもとで動作します。

IBM WebSphere MQ は TMF によって調整されるキュー・マネージャーごとに単一のゲートウェイ・プロセスを登録するので、TMF によって調整される作業単位に参加するキュー・マネージャーごとに別個の TMF/Gateway を構成する必要があります。この登録により、各キュー・マネージャーが独立リソース・マネージャーになるとともに、管理上、各キュー・マネージャーを HP NonStop TMF に一度登録するとマッピングが理解しやすくなります。

IBM WebSphere MQ の複数インストール済み環境の場合は、TMF によって調整されるキュー・マネージャーごとに、それらのインストール済み環境のいずれかにおける単一のゲートウェイ・プロセスを指定する必要があります。

ゲートウェイ・プロセスへのインターフェースは、同じバージョンまたはそれより前のどのクライアントもサポートします。

ゲートウェイ・プロセスの管理について詳しくは、[HP Integrity NonStop Server の管理](#)を参照してください。

Gateway を Pathway のもとで動作するように構成する

TMF/Gateway は、HP NonStop Transaction Management Facility (TMF) と TMF を IBM WebSphere MQ トランザクションのトランザクション・コーディネーターにするための IBM WebSphere MQ との間のインターフェースです。

IBM WebSphere MQ に備わっている TMF/Gateway は、リモート・キュー・マネージャーと通信するために、TMF 調整からのトランザクションを eXtended Architecture (XA) トランザクション調整に変換します。

調整を必要とするキュー・マネージャーごとに TMF/Gateway が 1 つ必要であり、正しい Gateway にクライアントが接続できるようにクライアント構成が必要です。

TMF/Gateway は、クライアントが使用できるすべてのメカニズムを使用してキュー・マネージャーと通信できます。他のアプリケーションの場合と同様に TMF/Gateway を構成します。

TMF/Gateway は HP Integrity NonStop Server プロセスの対ではなく、Pathway 環境で動作するように設計されています。TMF/Gateway は TMF 内に永続リソースを作成し、それらの永続リソースを TMF がその後の実行で再利用します。したがって、TMF/Gateway は常に同じユーザー権限で実行されなければなりません。

サーバー・クラスの定義

TMF/Gateway は Pathway 環境内でサーバー・クラスとしてホストされます。サーバー・クラスを定義するには、以下のサーバー属性を設定する必要があります。

PROCESSTYPE=OSS

サーバー・クラス内のサーバーのタイプを指定します。Gateway プロセスはマルチスレッド OSS プログラムです。この属性は必須であり、OSS に設定する必要があります。

MAXSERVERS=1

このサーバー・クラス内で同時に実行できるサーバー・プロセスの最大数を指定します。どのようなキュー・マネージャーについても、存在できる Gateway プロセスは 1 つだけです。この属性は必須であり、1 に設定する必要があります。

NUMSTATIC=1

このサーバー・クラス内の統計サーバーの最大数を指定します。Gateway プロセスは統計サーバーとして実行する必要があります。この属性は必須であり、1 に設定する必要があります。

TMF=ON

TMF サブシステムによって監査されるデータ・ファイルをこのサーバー・クラス内のサーバーがロックして更新できるかどうかを指定します。Gateway プロセスは IBM WebSphere MQ クライアント・アプリケーションの TMF トランザクションに参加するので、この属性を ON に設定する必要があります。

PROGRAM=<OSS installation path>/opt/mqm/bin/runmqtmf

IBM WebSphere MQ 用 IBM WebSphere MQ クライアントの場合、この属性は runmqtmf でなければなりません。この属性は OSS の絶対パス名である必要があります。大/小文字の区別があります。

ARGLIST=-m <QMgr name>[-c <channel name>] [-p <port>] [-h <host name>] [-n <max threads>]

これらの属性は Gateway プロセスのパラメーターを指定します。それぞれの意味は次のとおりです。

- QMgrName は、この Gateway プロセスのキュー・マネージャーの名前です。キュー共有グループ (または他のポート配分テクノロジー) を使用する場合、このパラメーターは特定のキュー・マネージャーをターゲットとしなければなりません。このパラメーターは必須です。
- channel name は、Gateway プロセスによって使用されるキュー・マネージャー上のサーバー・チャンネルの名前です。このパラメーターはオプションです。
- port は、キュー・マネージャーの TCP/IP ポートです。このパラメーターはオプションです。
- host name は、キュー・マネージャーのホスト名です。このパラメーターはオプションです。
- max threads は、Gateway プロセスによって作成されるワーカー・スレッドの最大数です。このパラメーターは 10 以上の値にすることができます。10 より小さい値が指定されても、使用される最小値は 10 です。値が指定されない場合、Gateway プロセスは最大 50 までスレッドを作成します。

459 ページの『[環境変数を使用した TMF/Gateway の構成](#)』に記載されている方式に加えて、Gateway に接続情報を提供する代替方式として、-c、-p、および -h 属性を使用します。-c、-p、および -h 属性を 1 つ以上指定するがすべては指定しない場合、指定しない属性はデフォルトで次の値になります。

- channel name はデフォルトで SYSTEM.DEF.SVRCONN になります。
- host name はデフォルトで localhost になります。

- port はデフォルトで 1414 になります。

指定したパラメーターのいずれかが無効である場合、TMF/Gateway は診断メッセージ [AMQ5379](#) をエラー・ログに発行して強制終了します。

OWNER=ID

Gateway を実行するユーザー ID。このユーザー ID には、キュー・マネージャーに対する接続権限が付与されていなければなりません。

SECURITY="value"

Owner 属性に関連して、IBM WebSphere MQ クライアント・アプリケーションから Gateway にアクセスできるユーザーを指定します。

LINKDEPTH および MAXLINKS は、Gateway と同時に通信する可能性のある IBM WebSphere MQ クライアント・アプリケーションの予想数に対して適切な値に構成されなければなりません。これらの値の設定が小さすぎると、クライアント・アプリケーションからエラー・メッセージ [AMQ5399](#) が出される場合があります。

これらのサーバー属性については、「[HP NonStop TS/MP 2.5 System Management Manual](#)」を参照してください。

環境変数を使用した TMF/Gateway の構成

TMF/Gateway を定義する際に最も一般的に使用される方式の 1 つとして、MQSERVER 環境変数を設定する方式があります。次に例を示します。

```
ENV MQSERVER=<channel name>/<transport>/<host name>(<listener port>)
```

このコマンドの最初の ENV は、Pathway 表記です。

クライアント初期設定ファイルの構成

HP NonStop Transaction Management Facility (TMF) を使用する場合は、HP Integrity NonStop Server が TMF ゲートウェイに到達するために IBM WebSphere MQ クライアントを有効にするための IBM WebSphere MQ クライアント初期設定ファイルが必要です。

HP Integrity NonStop Server 用の IBM WebSphere MQ クライアント初期設定ファイルは、いくつかの場所に保持することができます。詳しくは、[128 ページ](#)の『[クライアント構成ファイルの場所](#)』を参照してください。

構成ファイルの内容の詳細と例については、[127 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。TMF スタンザを使用して TMF キュー・マネージャーおよびサーバーの詳細を指定します。詳しくは、[143 ページ](#)の『[TMF および TMF/Gateway のスタンザ](#)』を参照してください。

IBM WebSphere MQ Client for HP Integrity NonStop Server の項目の例を以下に示します。

```
TMF:
  PathMon=$PSD1P

TmfGateway:
  QManager=MQ5B
  Server=MQ-MQ5B

TmfGateway:
  QManager=MQ5C
  Server=MQ-MQ5C
```

環境変数を使用したクライアントの構成については、[144 ページ](#)の『[WebSphere MQ 環境変数の使用](#)』を参照してください。

チャンネルへのアクセス権付与

IBM WebSphere MQ Client for HP Integrity NonStop Server のチャンネルへのアクセス権付与は他のオペレーティング・システムと同じですが、ゲートウェイを実行する所有者の ID が分かっている必要があります。

その上で、ゲートウェイの所有者の ID を使用して、適切なアクセス権を付与できます。大きな違いは、キュー・マネージャーのチャンネルへのアクセス権付与は、どのアプリケーションの権限のもとでも行われな
いという点です。

setmqaut コマンドを使用して、許可を付与します。つまり、IBM WebSphere MQ プリンシパルまたはユーザー・グループに操作を実行する許可を付与し、許可を取り消す (つまり、操作を実行する許可を削除する) ことができます。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

日本アイ・ビー・エム株式会社

法務・知的財産

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

〒 103-8510

103-8510

東京 103-8510、日本

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 INTERNATIONAL BUSINESS MACHINES CORPORATION は、法律上の瑕疵担保責任、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。"" 国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N

Rochester, MN 55901

U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っていません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、名前や住所が類似する個人や企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

プログラミング・インターフェース情報 (提供されている場合) は、このプログラムで使用するアプリケーション・ソフトウェアの作成を支援することを目的としています。

本書には、プログラムを作成するユーザーが IBM WebSphere MQ のサービスを使用するためのプログラミング・インターフェースに関する情報が記載されています。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

重要: この診断、修正、およびチューニング情報は、変更される可能性があるため、プログラミング・インターフェースとして使用しないでください。

商標

IBM、IBM ロゴ、ibm.com[®]は、世界の多くの国で登録された IBM Corporation の商標です。現時点での IBM の商標リストについては、"Copyright and trademark information" www.ibm.com/legal/copytrade.shtml 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: