

9.0

IBM MQ 構成リファレンス

IBM

注記

本書および本書で紹介する製品をご使用になる前に、[201 ページの『特記事項』](#)に記載されている情報をお読みください。

本書は、IBM® MQ バージョン 9 リリース 0、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

© Copyright International Business Machines Corporation 2007 年, 2023.

目次

構成に関する参照情報.....	5
すべてのプラットフォームでの IBM MQ の構成例.....	5
通信例の使用法.....	7
マルチスレッド・サポート - パイプライン.....	8
AIX での IBM MQ 構成の例.....	9
HP-UX の IBM MQ 構成の例.....	16
IBM i での MQ の構成例.....	21
Linux での MQ の構成例.....	39
Solaris での MQ の構成例.....	45
Windows での IBM MQ の構成例.....	51
z/OS での MQ の構成例.....	58
QSG を使用した z/OS での MQ の構成例.....	63
グループ内キューイングを使用した z/OS での MQ の構成例.....	70
/var/mqm に適用される IBM MQ ファイル・システム権限.....	78
mqm の setuid が設定された /opt/mqm の IBM MQ のファイル権限.....	82
Windows での IBM MQ のファイル・システム権限.....	84
キューの命名上の制約.....	85
その他のオブジェクトの命名上の制約.....	87
キュー名の解決.....	88
キュー名解決について.....	90
システムおよびデフォルト・オブジェクト.....	90
Windows デフォルト構成オブジェクト.....	94
SYSTEM.BASE.TOPIC.....	95
IBM i のシステム・オブジェクトおよびデフォルト・オブジェクト.....	96
スタンプ情報.....	100
分散キューイング用の構成ファイル・スタンプ.....	102
チャンネルの属性.....	104
チャンネル属性とチャンネル・タイプ.....	104
チャンネル属性 (アルファベット順).....	109
IBM MQ クラスター・コマンド.....	142
キュー・マネージャー定義コマンド.....	143
チャンネル定義コマンド.....	144
キュー定義コマンド.....	146
DISPLAY CLUSQMGR.....	149
SUSPEND QMGR、RESUME QMGR、およびクラスター.....	151
REFRESH CLUSTER.....	152
RESET CLUSTER: クラスターからキュー・マネージャーを強制的に除去する.....	153
クラスターでのワークロード・バランシング.....	155
z/OS における CLUSTER コマンドの非同期の動作.....	169
チャンネル・プログラム.....	169
環境変数.....	170
相互通信ジョブ.....	177
IBM i におけるチャンネルの状態.....	177
UNIX, Linux, and Windows におけるメッセージ・チャンネルの計画例.....	177
UNIX, Linux, and Windows の例が示す内容.....	178
UNIX, Linux, and Windows 用のサンプルの実行.....	181
IBM i におけるメッセージ・チャンネルの計画例.....	181
IBM i の例が示す内容.....	181
IBM i 用のサンプルの実行.....	185
IBM i 用のサンプルの拡張.....	185
z/OS におけるメッセージ・チャンネルの計画例.....	186

z/OS の例が示す内容.....	186
z/OS 用のサンプルの実行.....	189
z/OS 用のサンプルの拡張.....	189
キュー共有グループを使用する z/OS におけるメッセージ・チャンネルの計画例.....	190
z/OS のキュー共有グループ例が示す内容.....	190
キュー共有グループ定義.....	191
z/OS 用のキュー・マネージャー QM3 の例.....	192
z/OS 用のキュー共有グループ・サンプルの実行.....	193
別名を使用して MQ ライブラリーを参照する.....	193
mqzOSConnectService エlement.....	194
MQ Service Provider で使用できる HTTP ヘッダー.....	198

特記事項..... 201

プログラミング・インターフェース情報.....	202
商標.....	202

構成に関する参照情報

IBM MQ を構成する際には、このセクションの参照情報を使用してください。

構成についてのリファレンス情報は、以下のサブトピックに記載されています。

関連情報

構成

z/OS z/OS の構成

すべてのプラットフォームでの IBM MQ の構成例

構成の例により、IBM MQ ネットワークを運用環境として設定するために実行する作業について説明します。この作業により、IBM MQ の送信側チャンネルおよび受信側チャンネルが設定され、サポートされているすべてのプロトコルでの、プラットフォーム間の双方向メッセージ・フローが使用可能になります。

送受信以外のチャンネル・タイプを使用するには、[DEFINE CHANNEL](#) コマンドを参照してください。

5 ページの図 1 は、1 つのチャンネルとそれに関連する IBM MQ オブジェクトの概念を表したものです。

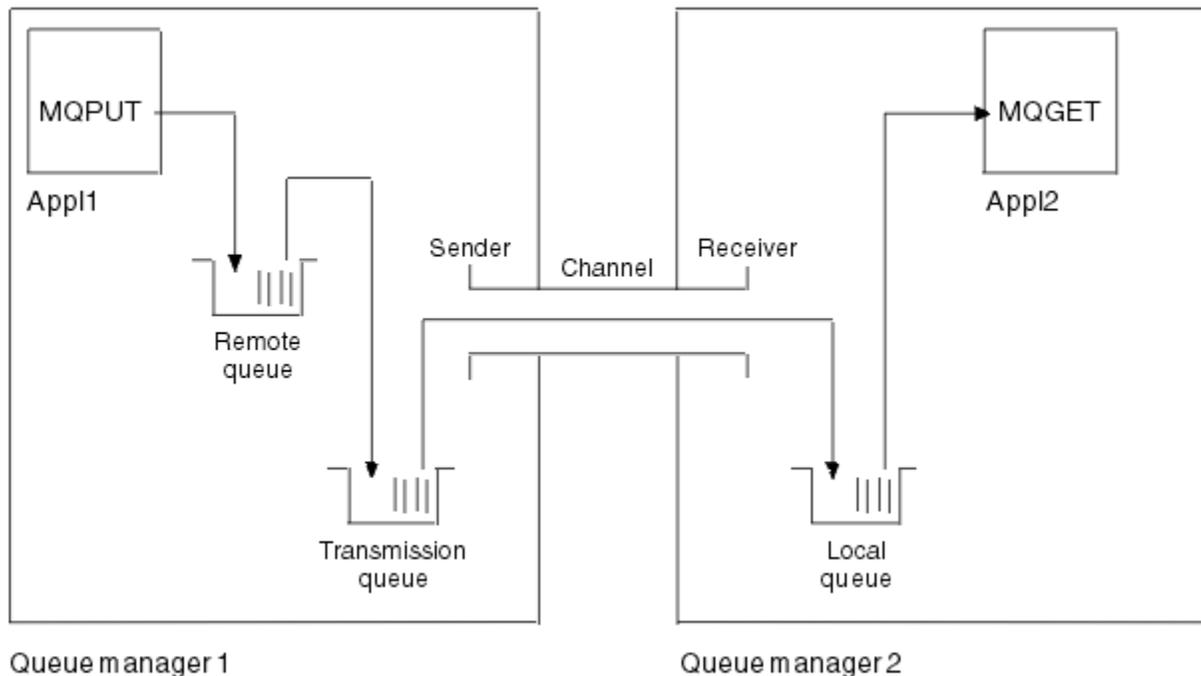


図 1. 構成例でセットアップする IBM MQ チャンネル

これは、IBM MQ ネットワークの基本的なエレメントだけを紹介するための簡単な例です。この例ではトリガー操作の使用法は説明していません。これについては、[チャンネルのトリガー操作](#)で説明しています。

このネットワークのオブジェクトは次のとおりです。

- リモート・キュー
- 伝送キュー
- ローカル・キュー
- 送信側チャンネル
- 受信側チャンネル

Appl1 と Appl2 は、どちらもアプリケーション・プログラムです。Appl1 はメッセージを書き込み、Appl2 はメッセージを受信します。

Appl1 はリモート・キューにメッセージを書き込みます。このリモート・キューの定義は、ターゲット・キュー・マネージャー、そのキュー・マネージャーのローカル・キュー、およびこのローカル・キュー・マネージャーの伝送キューの名前を指定します。

キュー・マネージャーは、リモート・キューへのメッセージ書き込み要求を Appl1 から受け取ったときに、宛先がリモートであることをキュー定義によって判別します。したがって、キュー・マネージャーは、伝送ヘッダーを付けたメッセージを、定義に指定されている伝送キューに直接書き込みます。メッセージは、チャンネルが使用可能になるまで伝送キューに残りますが、即時使用可能になることもあります。

送信側チャンネルには、その定義内に 1 つだけの伝送キューへの参照が備えられています。チャンネルの開始時や、その他の通常操作中のときに、チャンネルはこの伝送キューを参照して、そこにあるすべてのメッセージをターゲット・システムに送信します。メッセージの伝送ヘッダーには、宛先キューとキュー・マネージャーに関する詳細が入っています。

さまざまな組み合わせのプラットフォーム間で相互通信を行う例を使用して、前述の各オブジェクトの作成方法を詳細に説明しています。

ターゲット・キュー・マネージャーでは、ローカル・キューおよびチャンネルの受信側に対する定義が必要です。これらのオブジェクトは、互いに独立して操作し、任意の順序で作成できます。

ローカル・キュー・マネージャーでは、リモート・キュー、伝送キュー、およびチャンネルの送信側に対する定義が必要です。リモート・キュー定義とチャンネル定義の両方が伝送キュー名を参照するので、伝送キューを最初に作成することをお勧めします。

例示するネットワーク・インフラストラクチャー

構成の例では、次のように、特定のプラットフォームに特定のネットワーク・インフラストラクチャーが存在すると仮定しています。

- ▶ **z/OS** z/OS は、トークンリングに接続された 3745 ネットワーク・コントローラー (またはそれに相当するもの) を使用して通信を行う
- ▶ **Solaris** Solaris は、隣接するローカル・エリア・ネットワーク (LAN) 上にあり、これもまた 3745 ネットワーク・コントローラー (またはそれに相当するもの) に接続されている
- 他のすべてのプラットフォームはトークンリング・ネットワークに接続されている

また、SNA については、必要なすべての定義が VTAM およびネットワーク制御プログラム (NCP) に定義済みで、LAN 接続プラットフォームが広域ネットワーク (WAN) を介して通信できるように活動化されていることを前提としています。

同様に、TCP では、ドメイン・ネーム・サーバーまたはローカルに保持されるテーブル (ホスト・ファイルなど) のいずれかを使用して、ネーム・サーバー機能が使用可能であることを前提としています。

例中の通信ソフトウェア

以下に示すネットワーク・ソフトウェア製品について、運用構成例を示します。

- SNA
 - IBM Personal Communications for Windows 5.9
 - IBM Communications Server for AIX® 6.3
 - Hewlett-Packard SNAplus2
 - IBM i
 - Data Connection SNAP-IX バージョン 7 以降
 - OS/390® 2.4
- TCP
 - Microsoft Windows
 - AIX 4.1.4
 - HP-UX 10.2 以降

- Sun Solaris 2.4 以降
- IBM i
- TCP for z/OS
- HP Tru64 UNIX
- NetBIOS
- SPX

関連情報

構成

z/OS z/OS の構成

通信例の使用法

これらの構成例では、1つのプラットフォーム上で別のプラットフォームへの通信をセットアップするために実行する作業について説明します。次に、そのプラットフォームに運用チャンネルを設定する作業について説明します。

できるだけ情報を一般的にするようにしています。したがって、それぞれ異なるプラットフォーム上にある2つのキュー・マネージャーを接続する場合に参照する必要があるのは、関連する2つのセクションだけです。逸脱や特殊な事例は強調表示されています。また、同じプラットフォーム(異なるマシンまたは同一のマシン)で実行される2つのキュー・マネージャーも接続することができます。この場合は、すべての情報は1つのセクションから得られます。

ULW UNIX, Linux®, and Windows では、各プラットフォームに対応する手順を始める前に、各種の環境変数を設定する必要があります。以下のいずれかのコマンドを入力して、これを行います。

- **Windows** Windows の場合:

```
MQ_INSTALLATION_PATH/bin/setmqenv
```

ここで、MQ_INSTALLATION_PATH は IBM MQ がインストールされている場所を示しています。

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

```
. MQ_INSTALLATION_PATH/bin/setmqenv
```

ここで、MQ_INSTALLATION_PATH は IBM MQ がインストールされている場所を示しています。このコマンドは、現在作業しているシェルの環境変数を設定します。別のシェルを開いた場合には、コマンドを再入力する必要があります。

構成例で使用されるパラメーターが記載されたワークシートが用意されています。各パラメーターには短い説明があり、システムでこれに相当する値を探す場所についての指針が示されています。独自に値を設定する場合は、それらの値をワークシートの余白に書き留めておいてください。セクションを進むにつれ、必要に応じてこれらの値を相互参照します。

これらの例は、クラスター化を使用している場合の通信のセットアップ方法については取り上げていません。クラスター化を使用しているときの通信のセットアップについては、[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。その場合でも、ここに示した通信の構成値を適用できます。

以下のプラットフォームを対象とした構成例があります。

- **AIX** 9 ページの『[AIX での IBM MQ 構成の例](#)』
- **HP-UX** 16 ページの『[HP-UX の IBM MQ 構成の例](#)』
- **IBM i** 21 ページの『[IBM i での MQ の構成例](#)』
- **Linux** 39 ページの『[Linux での MQ の構成例](#)』

- ▶ **Solaris** 45 ページの『Solaris での MQ の構成例』
- ▶ **Windows** 51 ページの『Windows での IBM MQ の構成例』
- ▶ **z/OS** 58 ページの『z/OS での MQ の構成例』
- ▶ **z/OS** 63 ページの『QSG を使用した z/OS での MQ の構成例』
- ▶ **z/OS** 70 ページの『グループ内キューイングを使用した z/OS での MQ の構成例』

IT の責務

例中で使用されている用語を理解するためには、以下の指針を理解することから始めてください。

- システム管理者: 特定のプラットフォームにソフトウェアをインストールし、構成する人 (またはグループ) です。
- ネットワーク管理者: LAN 接続、LAN アドレス割り当て、ネットワーク命名規則などのネットワークに関する作業を管理する人です。この管理者は、システム管理グループに属していることも、別のグループに属していることもあります。

ほとんどの z/OS インストール環境には、ネットワーク構成をサポートするために ACF/VTAM、ACF/NCP、および TCP/IP ソフトウェアの更新を担当するグループが存在します。IBM MQ プラットフォームを IBM MQ for z/OS に接続する場合には、主に、このグループのメンバーから必要な情報を入手します。LAN のネットワーク命名規則は、このグループによって指示されたり指定されたりします。独自の定義を作成するときには、このグループの制御スパンを十分確認する必要があります。

- 特定のタイプの管理者、例えば CICS® 管理者は、その人の責務を明確に記述できる場合に指定されています。

構成例のセクションでは、各パラメーター設定の担当者については言及しません。一般的には、数人の担当者で行います。

関連概念

5 ページの『すべてのプラットフォームでの IBM MQ の構成例』

構成の例により、IBM MQ ネットワークを運用環境として設定するために実行する作業について説明します。この作業により、IBM MQ の送信側チャネルおよび受信側チャネルが設定され、サポートされているすべてのプロトコルでの、プラットフォーム間の双方向メッセージ・フローが使用可能になります。

関連情報

setmqenv

Multi マルチスレッド・サポート - パイプライン

オプションで、メッセージ・チャネル・エージェント (MCA) により、複数のスレッドを使用してメッセージを転送できます。このプロセスのことをパイプラインといい、このプロセスを使用すると、MCA によるメッセージ転送の効率が上がり、待ち状態が少なくなり、チャネルのパフォーマンスが向上します。MCA 当たり最大 2 つのスレッドに限定されています。

パイプラインを制御するには、qm.ini ファイル中で *PipeLineLength* パラメーターを使用します。以下のように、このパラメーターを CHANNELS スタンザに追加します。

PipeLineLength= 1 | number

この属性は、チャネルが使用する並行スレッドの最大数を指定します。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。

注: パイプラインは TCP/IP チャネルの場合だけ有効です。

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ネルの両側のキュー・マネージャーを構成する必要があります。

チャネル出口に関する考慮事項

次の理由で、パイプラインによって一部の出口プログラムが失敗します。

- 出口が逐次に呼び出されない。
- 出口が別のスレッドから代替呼び出しされる。

パイプラインを使用する場合は、その前に以下の点について出口プログラムの設計を確認してください。

- 出口はすべての実行段階で再入可能でなければならない。
- MQI 呼び出しを使用する場合は、別々のスレッドから出口が呼び出される時は同一の MQI ハンドルを使用できないことを念頭に置かなければならない。

あるメッセージ出口が、キューをオープンし、それ以降のすべての出口呼び出しでこのハンドルを使用して MQPUT 呼び出しを行うとします。この出口は別のスレッドから呼び出されるので、パイプライン・モードでは失敗します。失敗しないようにするには、スレッドごとにキュー・ハンドルを保持し、出口が呼び出されるたびにスレッド ID を検査してください。

AIX AIX での IBM MQ 構成の例

このセクションでは IBM MQ for AIX から IBM MQ 製品への通信リンクをセットアップする方法を例を挙げて説明します。

以下のプラットフォームが、例でカバーされています。

- **Windows** Windows
- **UNIX** HP Tru64 UNIX
- **HP-UX** HP-UX
- **Solaris** Solaris
- **Linux** Linux
- **IBM i** IBM i
- **z/OS** z/OS
- VSE/ESA

このセクションのバックグラウンド情報およびその使用方法については、5 ページの『すべてのプラットフォームでの IBM MQ の構成例』を参照してください。

AIX LU 6.2 接続の確立

LU 6.2 接続に必要なパラメーターについて説明します。

SNA over TCP/IP の構成の最新情報については、IBM オンライン資料 [Communications Server for AIX](#) を参照してください。

AIX TCP 接続の確立

チャンネルを開始するには、必ずその前にリスナーを明示的に開始してください。これにより、インバウンド送信チャンネルからの要求にตอบสนองして受信チャンネルが自動的に開始されるようになります。

IBM MQ の TCP リスナーを開始させるために使用される IBM MQ コマンドは、次のとおりです。

```
runmqclsr -t tcp
```

もう一つの方法として、UNIX が提供する TCP/IP リスナーを使用する場合は、次のステップを実行します。

1. /etc/services というファイルを編集する。

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。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      1414/tcp      # MQSeries channel listener
```

2. /etc/inetd.conf というファイルを編集する。以下の行がこのファイルにない場合は、示されているとおりに追加して、MQ_INSTALLATION_PATH を IBM MQ がインストールされている上位のディレクトリーに置き換えてください。

```
MQSeries stream tcp nowait root MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
[-m queue.manager.name]
```

3. コマンド `refresh -s inetd` を入力します。

注: mqm グループに **root** を追加しなければなりません。1 次グループを mqm に設定する必要はありません。一連のグループ内に mqm があれば、コマンドを使用することができます。キュー・マネージャーを使用するアプリケーションだけを実行する場合、mqm グループ権限は必要ありません。

次のステップ

これで、接続は確立されました。構成を完成させる準備ができました。10 ページの『IBM MQ for AIX の構成』へ進んでください。

AIX IBM MQ for AIX の構成

チャンネルを定義して構成を完了します。

注:

1. インストール・プロセスを開始する前に、まず、mqm ユーザーおよびグループを作成済みであることを確認してから、パスワードを設定します。
2. ファイル・システムのスペース不足のためにインストールが失敗した場合は、コマンド `smit C sna` を使用して、次のようにサイズを拡大できます。(ファイル・システムの状況を表示するには、`df` を使用します。このコマンドによって、満杯になっている論理ボリュームが分かります。)

```
-- Physical and Logical Storage  
-- File Systems  
-- Add / Change / Show / Delete File Systems  
-- Journaled File Systems  
-- Change/Show Characteristics of a Journaled File System
```

3. 次のコマンドを使用して、チャンネルを開始します。

```
runmqchl -c channel.name
```

4. サンプル・プログラムは MQ_INSTALLATION_PATH/samp にインストールされています。ここで、MQ_INSTALLATION_PATH は IBM MQ がインストールされている上位のディレクトリーを表します。
5. エラー・ログは、/var/mqm/qmgrs/qmgrname/errors に格納されています。
6. AIX では、IBM MQ 標準のトレース・コマンドまたは AIX システム・トレースを使用して、IBM MQ コンポーネントのトレースを開始できます。IBM MQ のトレースおよび AIX システム・トレースの詳細については、[トレースの使用法](#)を参照してください。
7. コマンド・インタプリター `runmqsc` を使用して、管理コマンドを入力している場合、行の終わりにある `+` は、次の行が継続していることを示します。必ず最後のパラメーターと継続文字との間にスペースを入れるようにします。

基本構成

1. 次のコマンドを使用して AIX コマンド・ラインからキュー・マネージャーを作成します。

```
crtmqm -u dlqname -q aix
```

ここで、

aix

キュー・マネージャーの名前

-q

これがデフォルトのキュー・マネージャーになることを示す

-u *dlqname*

未配布メッセージ・キューの名前を指定する

このコマンドは、キュー・マネージャーおよび一連のデフォルト・オブジェクトを作成します。

2. 次のコマンドを使用して AIX コマンド・ラインからキュー・マネージャーを開始します。

```
strmqm aix
```

ここで、*aix* は、作成時に、キュー・マネージャーに指定された名前です。

3. AIX コマンド・ラインから **runmqsc** を開始し、これを使用して、次のコマンドを入力することによって未配布メッセージ・キューを作成します。

```
def ql (dlqname)
```

ここで、*dlqname* は、キュー・マネージャーが作成されたとき、未配布メッセージ・キューに指定された名前です。

AIX AIX 用のチャネル構成

指定されたチャネルおよびプラットフォームのキュー・マネージャーの構成についての情報を提供します。

次のセクションでは、5 ページの『すべてのプラットフォームでの IBM MQ の構成例』に記載したチャネルを実装するために、AIX キュー・マネージャーで実行する構成について詳細に説明します。

それぞれの構成について該当する MQSC コマンドを示しています。AIX コマンド・ラインから **runmqsc** を開始して各コマンドを順次入力するか、コマンドを組み込んだコマンド・ファイルを作成します。

IBM MQ for AIX と IBM MQ for Windows の接続のための例が示されています。他のプラットフォーム上の IBM MQ に接続するには、Windows 用の値の代わりに表から該当する一連の値を使用します。

注: 太字の語は、ユーザーが指定するもので、これらの例で使用される IBM MQ オブジェクトの名前を反映しています。ここで使用される名前を変更する場合は、このセクション全体でこれらのオブジェクトに行われている他の参照も必ず変更してください。他はすべてキーワードなので、記載されているとおりに入力してください。

ID	パラメーター名	参照	使用例	ユーザー値
ローカル・ノードの定義				
A	キュー・マネージャー名		AIX	
B	ローカル・キュー名		AIX.LOCALQ	
IBM MQ for Windows との接続				
表のこのセクションで指定する値は、54 ページの『Windows 用のチャネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	WINNT	
D	リモート・キュー名		WINNT.REMOTEQ	
E	リモート・システムのキュー名	B	WINNT.LOCALQ	
F	伝送キュー名		WINNT	

表 1. IBM MQ for AIX 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
G	送信側 (SNA) チャンネル名		AIX.WINNT.SNA	
H	送信側 (TCP/IP) チャンネル名		AIX.WINNT.TCP	
I	受信側 (SNA) チャンネル名	G	WINNT.AIX.SNA	
J	受信側 (TCP) チャンネル名	H	WINNT.AIX.TCP	
IBM MQ for HP Tru64 UNIX との接続				
表のこのセクションで指定する値は、HP Tru64 UNIX システムで使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	DECUX	
D	リモート・キュー名		DECUX.REMOTEQ	
E	リモート・システムのキュー名	B	DECUX.LOCALQ	
F	伝送キュー名		DECUX	
H	送信側 (TCP) チャンネル名		DECUX.AIXTCP (TCP)	
J	受信側 (TCP) チャンネル名	H	AIX.DECUX.TCP	
IBM MQ for HP-UX との接続				
表のこのセクションで指定する値は、17 ページの『HP-UX 用のチャンネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	HPUX	
D	リモート・キュー名		HPUX.REMOTEQ	
E	リモート・システムのキュー名	B	HPUX.LOCALQ	
F	伝送キュー名		HPUX	
G	送信側 (SNA) チャンネル名		AIX.HPUX.SNA	
H	送信側 (TCP) チャンネル名		AIX.HPUX.TCP	
I	受信側 (SNA) チャンネル名	G	HPUX.AIX.SNA	
J	受信側 (TCP) チャンネル名	H	HPUX.AIX.TCP	
IBM MQ for Solaris との接続				
表のこのセクションで指定する値は、47 ページの『Solaris 用のチャンネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	SOLARIS	
D	リモート・キュー名		SOLARIS.REMOTEQ	
E	リモート・システムのキュー名	B	SOLARIS.LOCALQ	
F	伝送キュー名		SOLARIS	
G	送信側 (SNA) チャンネル名		AIX.SOLARIS.SNA	
H	送信側 (TCP/IP) チャンネル名		AIX.SOLARIS.TCP	
I	受信側 (SNA) チャンネル名	G	SOLARIS.AIX.SNA	
J	受信側 (TCP/IP) チャンネル名	H	SOLARIS.AIX.TCP	

表 1. IBM MQ for AIX 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
IBM MQ for Linux との接続				
表のこのセクションで指定する値は、42 ページの『Linux 用のチャネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	LINUX	
D	リモート・キュー名		LINUX.REMOTEQ	
E	リモート・システムのキュー名	B	LINUX.LOCALQ	
F	伝送キュー名		LINUX	
G	送信側 (SNA) チャネル名		AIX.LINUX.SNA	
H	送信側 (TCP/IP) チャネル名		AIX.LINUX.TCP	
I	受信側 (SNA) チャネル名	G	LINUX.AIX.SNA	
J	受信側 (TCP/IP) チャネル名	H	LINUX.AIX.TCP	
IBM i IBM MQ for IBM i との接続				
表のこのセクションで指定する値は、34 ページの『IBM i 用のチャネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	AS400	
D	リモート・キュー名		AS400.REMOTEQ	
E	リモート・システムのキュー名	B	AS400.LOCALQ	
F	伝送キュー名		AS400	
G	送信側 (SNA) チャネル名		AIX.AS400.SNA	
H	送信側 (TCP) チャネル名		AIX.AS400.TCP	
I	受信側 (SNA) チャネル名	G	AS400.AIX.SNA	
J	受信側 (TCP) チャネル名	H	AS400AIX。TCP (TCP)	
z/OS IBM MQ for z/OS との接続				
表のこのセクションで指定する値は、60 ページの『z/OS 用のチャネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	MVS	
D	リモート・キュー名		MVS.REMOTEQ	
E	リモート・システムのキュー名	B	MVS.LOCALQ	
F	伝送キュー名		MVS	
G	送信側 (SNA) チャネル名		AIX.MVS.SNA	
H	送信側 (TCP) チャネル名		AIX.MVS.TCP	
I	受信側 (SNA) チャネル名	G	MVS.AIX.SNA	

表 1. IBM MQ for AIX 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
▶ z/OS ▶ z/OS J	受信側 (TCP) チャンネル名	H	MVS.AIX.TCP	
▶ z/OS キュー共有グループを使用した IBM MQ for z/OS との接続				
▶ z/OS 表のこのセクションで指定する値は、68 ページの『共有チャンネル構成の例』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	QSG	
D	リモート・キュー名		QSG.REMOTEQ	
E	リモート・システムのキュー名	B	QSG.SHAREDQ	
F	伝送キュー名		QSG	
G	送信側 (SNA) チャンネル名		AIX.QSG.SNA	
H	送信側 (TCP) チャンネル名		AIX.QSG.TCP	
I	受信側 (SNA) チャンネル名	G	QSG.AIX.SNA	
▶ z/OS ▶ z/OS J	受信側 (TCP) チャンネル名	H	QSG.AIX.TCP	
MQSeries® for VSE/ESA との接続				
このセクションの表の値は、VSE/ESA システムで使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	VSE	
D	リモート・キュー名		VSE.REMOTEQ	
E	リモート・システムのキュー名	B	VSE.LOCALQ	
F	伝送キュー名		VSE	
G	送信側チャンネル名		AIX.VSE.SNA	
I	受信側チャンネル名	G	VSE.AIX.SNA	

▶ AIX SNA を使用した IBM MQ for AIX 送信側チャンネル定義コマンドの例。

```
def ql (WINNT) +                               F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                       D
  rname(WINNT.LOCALQ) +                       E
  rqmname(WINNT) +                             C
  xmitq(WINNT) +                               F
  replace

def chl (AIX.WINNT.SNA) chltype(sdr) +        G
  trptype(lu62) +
  conname('WINNTCPIC') +                      17
  xmitq(WINNT) +                               F
  replace
```

AIX SNA を使用した IBM MQ for AIX 受信側チャンネル定義
コマンドの例。

```
def ql (AIX.LOCALQ) replace B
def chl (WINNT.AIX.SNA) chltype(rcvr) + I
  trptype(lu62) +
  replace
```

AIX IBM MQ for AIX TPN のセットアップ

送信側チャンネルが会話を開始するときに、SNA 受信側チャンネルが必ず正しく活動化するようにする別の方法。

AIX Communications Server の構成プロセス中に作成された LU 6.2 TPN プロファイルには、TP 実行可能プログラムへの絶対パスが含まれています。例えば、ファイルは `u/interops/AIX.crs6a` と呼ばれていました。名前を選択することはできますが、その名前に使用するキュー・マネージャーの名前を含めることも検討してください。実行可能ファイルの内容は、次のとおりでなければなりません。

```
#!/bin/sh
MQ_INSTALLATION_PATH/bin/amqcrs6a -m aix
```

ここで、`aix` はキュー・マネージャー名 (A) であり、`MQ_INSTALLATION_PATH` は IBM MQ がインストールされている上位ディレクトリーです。このファイルを作成した後、次のコマンドを実行するとその実行が可能になります。

```
chmod 755 /u/interops/AIX.crs6a
```

実行可能ファイルを作成する代わりに、「LU 6.2 TPN プロファイルの追加」パネルでコマンド行パラメーターを使用して上記のパスを指定することもできます。

これら 2 つの方法のどちらかでパスを指定すると、送信側チャンネルが会話を開始した場合、必ず SNA 受信側チャンネルが正しく活動化します。

AIX TCP を使用した IBM MQ for AIX 送信側チャンネル定義
コマンドの例。

```
def ql (WINNT) + F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) + D
  rname(WINNT.LOCALQ) + E
  rqmname(WINNT) + C
  xmitq(WINNT) + F
  replace

def chl (AIX.WINNT.TCP) chltype(sdr) + H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(WINNT) + F
  replace
```

AIX TCP を使用した IBM MQ for AIX 受信側チャンネル定義
コマンドの例。

```
def ql (AIX.LOCALQ) replace B
def chl (WINNT.AIX.TCP) chltype(rcvr) + J
  trptype(tcp) +
  replace
```

HP-UX HP-UX の IBM MQ 構成の例

このセクションでは IBM MQ for HP-UX から IBM MQ 製品への通信リンクをセットアップする方法を例を挙げて説明します。

以下のプラットフォームが含まれます。

- Windows
- AIX
- HP Tru64 UNIX
- Solaris
- Linux
- IBM i
- z/OS
- VSE/ESA

このセクションのバックグラウンド情報およびその使用方法については、5 ページの『すべてのプラットフォームでの IBM MQ の構成例』を参照してください。

HP-UX LU 6.2 接続の確立

LU 6.2 接続に必要なパラメーターについて説明します。

SNA over TCP/IP の構成に関する最新情報については、IBM オンライン資料「[Communications Server](#)」、および HP オンライン資料「[HP-UX SNAplus2 Installation Guide](#)」を参照してください。

HP-UX TCP 接続の確立

接続を確立する代替方法と、次のステップについて説明します。

チャンネルを開始するには、必ずその前にリスナーを明示的に開始してください。これにより、インバウンド送信チャンネルからの要求に 응답して受信チャンネルが自動的に開始されるようになります。

もう一つの方法として、UNIX が提供する TCP/IP リスナーを使用する場合は、次のステップを実行します。

1. `/etc/services` というファイルを編集する。

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。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      1414/tcp      # MQSeries channel listener
```

2. `/etc/inetd.conf` というファイルを編集する。以下の行がこのファイルにない場合は、示されているとおりに追加して、`MQ_INSTALLATION_PATH` を IBM MQ がインストールされている上位のディレクトリーに置き換えてください。

```
MQSeries stream tcp nowait root MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
[-m queue.manager.name]
```

3. 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
```

4. コマンドを実行します。

```
kill -1 inetd processid
```

注: mqm グループに **root** を追加しなければなりません。1 次グループを mqm に設定する必要はありません。一連のグループ内に mqm があれば、コマンドを使用することができます。キュー・マネージャーを使用するアプリケーションだけを実行する場合、mqm グループ権限は必要ありません。

次のステップ

これで、接続は確立されました。構成を完成させる準備ができました。[17 ページの『IBM MQ for HP-UX 構成』](#)へ進んでください。

HP-UX IBM MQ for HP-UX 構成

構成を完了するためのチャンネルの定義について説明します。

インストール・プロセスを開始する前に、まず、mqm ユーザーおよびグループを作成済みであることを確認してから、パスワードを設定します。

次のコマンドを使用して、チャンネルを開始します。

```
runmqchl -c channel.name
```

注:

1. サンプル・プログラムは `MQ_INSTALLATION_PATH/samp` にインストールされています。ここで、`MQ_INSTALLATION_PATH` は IBM MQ がインストールされている上位のディレクトリーを表します。
2. エラー・ログは、`/var/mqm/qmgrs/qmgrname/errors` に格納されています。
3. コマンド・インタープリター `runmqsc` を使用して、管理コマンドを入力している場合、行の終わりにある `+` は、次の行が継続していることを示します。必ず最後のパラメーターと継続文字との間にスペースを入れるようにします。

基本構成

1. 次のコマンドを使用して、UNIX プロンプトからキュー・マネージャーを作成します。

```
crtmqm -u dlqname -q hpuX
```

ここで、

hpuX

キュー・マネージャーの名前

-q

これがデフォルトのキュー・マネージャーになることを示す

-u dlqname

未配布メッセージ・キューの名前を指定する

このコマンドは、キュー・マネージャーおよび一連のデフォルト・オブジェクトを作成します。このコマンドは、キュー・マネージャーの DEADQ 属性を設定しますが、未配布メッセージ・キューを作成しません。

2. 次のコマンドを使用して、UNIX プロンプトからキュー・マネージャーを開始します。

```
strmqm hpuX
```

ここで、`hpuX` は、作成時に、キュー・マネージャーに指定された名前です。

HP-UX HP-UX 用のチャンネル構成

指定されたチャンネルおよびプラットフォームのキュー・マネージャーの構成についての情報を提供します。

次のセクションでは、[5 ページの『すべてのプラットフォームでの IBM MQ の構成例』](#)に記載したチャンネルを実装するために、HP-UX キュー・マネージャーで実行する構成について詳細に説明します。

それぞれの構成について該当する MQSC コマンドを示しています。UNIX プロンプトから **runmqsc** を開始して各コマンドを順次入力するか、またはコマンドをコマンド・ファイルに組み込みます。

IBM MQ for HP-UX と IBM MQ for Windows の接続のための例が示されています。他のプラットフォーム上の IBM MQ に接続するには、Windows 用の値の代わりに表から該当する一連の値を使用します。

注: 太字の語は、ユーザーが指定するもので、これらの例で使用される IBM MQ オブジェクトの名前を反映しています。ここで使用される名前を変更する場合は、このセクション全体でこれらのオブジェクトに行われている他の参照も必ず変更してください。他はすべてキーワードなので、記載されているとおりに入力してください。

表 2. IBM MQ for HP-UX 用構成ワークシート				
ID	パラメーター名	参照	使用例	ユーザー値
ローカル・ノードの定義				
A	キュー・マネージャー名		HPUX	
B	ローカル・キュー名		HPUX.LOCALQ	
IBM MQ for Windows との接続				
表のこのセクションで指定する値は、54 ページの『Windows 用のチャンネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	WINNT	
D	リモート・キュー名		WINNT.REMOTEQ	
E	リモート・システムのキュー名	B	WINNT.LOCALQ	
F	伝送キュー名		WINNT	
G	送信側 (SNA) チャンネル名		HPUX.WINNT.SNA	
H	送信側 (TCP/IP) チャンネル名		HPUX.WINNT.TCP	
I	受信側 (SNA) チャンネル名	G	WINNT.HPUX.SNA	
J	受信側 (TCP) チャンネル名	H	WINNT.HPUX.TCP	
IBM MQ for AIX との接続				
表のこのセクションで指定する値は、11 ページの表 1 で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	AIX	
D	リモート・キュー名		AIX.REMOTEQ	
E	リモート・システムのキュー名	B	AIX.LOCALQ	
F	伝送キュー名		AIX	
G	送信側 (SNA) チャンネル名		HPUX.AIX.SNA	
H	送信側 (TCP) チャンネル名		HPUX.AIX.TCP	
I	受信側 (SNA) チャンネル名	G	AIX.HPUX.SNA	
J	受信側 (TCP) チャンネル名	H	AIX.HPUX.TCP	
IBM MQ for HP Tru64 UNIX との接続				
表のこのセクションで指定する値は、HP Tru64 UNIX システムで使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	DECUX	
D	リモート・キュー名		DECUX.REMOTEQ	

表 2. IBM MQ for HP-UX 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
E	リモート・システムのキュー名	B	DECUX.LOCALQ	
F	伝送キュー名		DECUX	
H	送信側 (TCP) チャンネル名		DECUX.HPUX.TCP	
J	受信側 (TCP) チャンネル名	H	HPUX.DECUX.TCP	

IBM MQ for Solaris との接続

表のこのセクションで指定する値は、47 ページの表 6 で指定した値と一致させる必要があります。

C	リモート・キュー・マネージャー名	A	SOLARIS	
D	リモート・キュー名		SOLARIS.REMOTEQ	
E	リモート・システムのキュー名	B	SOLARIS.LOCALQ	
F	伝送キュー名		SOLARIS	
G	送信側 (SNA) チャンネル名		HPUX.SOLARIS.SNA	
H	送信側 (TCP/IP) チャンネル名		HPUX.SOLARIS.TCP	
I	受信側 (SNA) チャンネル名	G	SOLARIS.HPUX.SNA	
J	受信側 (TCP/IP) チャンネル名	H	SOLARIS.HPUX.TCP	

IBM MQ for Linux との接続

表のこのセクションで指定する値は、42 ページの表 5 で指定した値と一致させる必要があります。

C	リモート・キュー・マネージャー名	A	LINUX	
D	リモート・キュー名		LINUX.REMOTEQ	
E	リモート・システムのキュー名	B	LINUX.LOCALQ	
F	伝送キュー名		LINUX	
G	送信側 (SNA) チャンネル名		HPUX.LINUX.SNA	
H	送信側 (TCP/IP) チャンネル名		HPUX.LINUX.TCP	
I	受信側 (SNA) チャンネル名	G	LINUX.HPUX.SNA	
J	受信側 (TCP/IP) チャンネル名	H	LINUX.HPUX.TCP	

IBM i IBM MQ for IBM i との接続

表のこのセクションで指定する値は、35 ページの表 4 で指定した値と一致させる必要があります。

C	リモート・キュー・マネージャー名	A	AS400	
D	リモート・キュー名		AS400.REMOTEQ	
E	リモート・システムのキュー名	B	AS400.LOCALQ	
F	伝送キュー名		AS400	
G	送信側 (SNA) チャンネル名		HPUX.AS400.SNA	
H	送信側 (TCP/IP) チャンネル名		HPUX.AS400.TCP	

表 2. IBM MQ for HP-UX 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
I	受信側 (SNA) チャンネル名	G	AS400.HPUX.SNA	
▶ IBM I ▶ IBM I J	受信側 (TCP) チャンネル名	H	AS400.HPUX.TCP	
<p>▶ z/OS IBM MQ for z/OS との接続</p> <p>▶ z/OS 表のこのセクションで指定する値は、60 ページの表 8 で指定した値と一致させる必要があります。</p>				
C	リモート・キュー・マネージャー名	A	MVS	
D	リモート・キュー名		MVS.REMOTEQ	
E	リモート・システムのキュー名	B	MVS.LOCALQ	
F	伝送キュー名		MVS	
G	送信側 (SNA) チャンネル名		HPUX.MVS.SNA	
H	送信側 (TCP) チャンネル名		HPUX.MVS.TCP	
I	受信側 (SNA) チャンネル名	G	MVS.HPUX.SNA	
▶ z/OS ▶ z/OS J	受信側 (TCP) チャンネル名	H	MVS.HPUX.TCP	
<p>MQSeries for VSE/ESA との接続</p> <p>このセクションの表の値は、VSE/ESA システムで使用した値と一致させる必要があります。</p>				
C	リモート・キュー・マネージャー名	A	VSE	
D	リモート・キュー名		VSE.REMOTEQ	
E	リモート・システムのキュー名	B	VSE.LOCALQ	
F	伝送キュー名		VSE	
G	送信側チャンネル名		HPUX.VSE.SNA	
I	受信側チャンネル名	G	VSE.HPUX.SNA	

▶ **HP-UX** SNA を使用した IBM MQ for HP-UX 送信側チャンネル定義コマンドの例。

```
def q1 (WINNT) +                               F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                       D
  rname(WINNT.LOCALQ) +                       E
  rqmname(WINNT) +                             C
  xmitq(WINNT) +                               F
  replace

def chl (HPUX.WINNT.SNA) chltype(sdr) +       G
  trptype(lu62) +
  conname('WINNTCPIC') +                       16
```

```
xmitq(WINNT) +      F
replace
```

HP-UX SNA を使用した IBM MQ for HP-UX 受信側チャンネル定義
コマンドの例。

```
def ql (HPUX.LOCALQ) replace      B
def chl (WINNT.HPUX.SNA) chltype(rcvr) +      I
  trptype(lu62) +
  replace
```

HP-UX IBM MQ for HP-UX の呼び出し可能 TP セットアップ
送信側チャンネルが会話を開始すると、SNA 受信側チャンネルが確実に正しく活動化されるようにします。
HP SNAplus2 リリース 6 の場合、以下の内容は不要です。

HP SNAplus2 構成プロセス中に、実行可能ファイルを指す呼び出し可能 TP 定義が作成されました。例えば、ファイルは /users/interop/HPUX.crs6a と呼ばれていました。このファイルの名前は任意に指定できますが、名前にキュー・マネージャーの名前を含めることを検討してください。実行可能ファイルの内容は、次のとおりでなければなりません。

```
#!/bin/sh
MQ_INSTALLATION_PATH/bin/amqcrs6a -m hpux
```

ここで、*hpux* はキュー・マネージャー A の名前、*MQ_INSTALLATION_PATH* は IBM MQ がインストールされている上位のディレクトリーです。

この場合、送信側チャンネルが会話を開始すると、SNA 受信側チャンネルは、必ず正確に起動します。

HP-UX TCP を使用した IBM MQ for HP-UX 送信側チャンネル定義
コマンドの例。

```
def ql (WINNT) +      F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +      D
  rname(WINNT.LOCALQ) +      E
  rqmname(WINNT) +          C
  xmitq(WINNT) +          F
  replace

def chl (HPUX.WINNT.TCP) chltype(sdr) +      H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(WINNT) +          F
  replace
```

HP-UX TCP/IP を使用した IBM MQ for HP-UX 受信側チャンネル定義
コマンドの例。

```
def ql (HPUX.LOCALQ) replace      B
def chl (WINNT.HPUX.TCP) chltype(rcvr) +      J
  trptype(tcp) +
  replace
```

IBM i IBM i での MQ の構成例

このセクションでは、その他のプラットフォーム上で IBM MQ for IBM i から IBM MQ 製品への通信リンクをセットアップする方法を例を挙げて説明します。

カバーされているその他のプラットフォームは、以下のプラットフォームです。

- Windows
- AIX
- Compaq Tru64 UNIX
- HP-UX
- Solaris
- Linux
- z/OS または MVS
- VSE/ESA

このセクションのバックグラウンド情報およびその使用方法については、5 ページの『すべてのプラットフォームでの IBM MQ の構成例』を参照してください。

IBM i LU 6.2 接続用構成パラメーター

以下のワークシートは、IBM i システムから、他のいずれかの IBM MQ プラットフォームへの通信をセットアップするために必要なすべてのパラメーターをリストしています。このワークシートには、作業環境でテスト済みのパラメーターの例が示されているほか、ユーザー独自の値を入力するスペースもあります。

このセクションのワークシートを使用して、この構成の値を記録してください。このワークシートは、接続先のプラットフォームに対応するセクションのワークシートと共に使用します。

「参照」列に番号が記載されている場合、このセクションの別の場所に記載されている該当するワークシートでの値と一致しなければならないことを示します。このセクションの後に出てくる例では、この表の「ID」列の値を参照しています。

「パラメーター名」列の項目は、24 ページの『用語の説明』で説明します。

ID	パラメーター名	参照	使用例	ユーザー値
ローカル・ノードの定義				
1	ローカル・ネットワーク ID		NETID	
2	ローカル制御点名		AS400PU	
3	LU 名		AS400LU	
4	LAN 宛先アドレス		10005A5962EF	
5	サブシステム記述		QCMN	
6	回線記述		TOKENRINGL	
7	リソース名		LIN041	
8	ローカル・トランザクション・プログラム名		MQSERIES	
Windows システムとの接続				
9	ネットワーク ID	2	NETID	
10	制御点名	3	WINNTCP	
11	LU 名	5	WINNTLU	
12	コントローラー記述		WINNTCP	
13	デバイス		WINNTLU	
14	サイド情報		NTCPIC	

表 3. IBM i システム上の SNA 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
15	トランザクション・プログラム	7	MQSERIES	
16	LAN アダプター・アドレス	9	08005AA5FAB9	
17	モード (Mode)	17	#INTER	
AIX システムとの接続				
9	ネットワーク ID	1	NETID	
10	制御点名	2	AIXPU	
11	LU 名	4	AIXLU	
12	コントローラー記述		AIXPU	
13	デバイス		AIXLU	
14	サイド情報		AIXCPIC	
15	トランザクション・プログラム	6	MQSERIES	
16	LAN アダプター・アドレス	8	123456789012	
17	モード (Mode)	14	#INTER	
HP-UX システムとの接続				
9	ネットワーク ID	4	NETID	
10	制御点名	2	HPUXPU	
11	LU 名	5	HPUXLU	
12	コントローラー記述		HPUXPU	
13	デバイス		HPUXLU	
14	サイド情報		HPUXCPIC	
15	トランザクション・プログラム	7	MQSERIES	
16	LAN アダプター・アドレス	8	100090DC2C7C	
17	モード (Mode)	17	#INTER	
Solaris システムとの接続				
9	ネットワーク ID	2	NETID	
10	制御点名	3	SOLARPU	
11	LU 名	7	SOLARLU	
12	コントローラー記述		SOLARPU	
13	デバイス		SOLARLU	
14	サイド情報		SOLCPIC	
15	トランザクション・プログラム	8	MQSERIES	
16	LAN アダプター・アドレス	5	08002071CC8A	
17	モード (Mode)	17	#INTER	
Linux (x86 プラットフォーム) システムとの接続				

表 3. IBM i システム上の SNA 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
9	ネットワーク ID	4	NETID	
10	制御点名	2	LINUXPU	
11	LU 名	5	LINUXLU	
12	コントローラー記述		LINUXPU	
13	デバイス		LINUXLU	
14	サイド情報		LXCPIC	
15	トランザクション・プログラム	7	MQSERIES	
16	LAN アダプター・アドレス	8	08005AC6DF33	
17	モード (Mode)	6	#INTER	
z/OS システムとの接続				
9	ネットワーク ID	2	NETID	
10	制御点名	3	MVSPU	
11	LU 名	4	MVSLU	
12	コントローラー記述		MVSPU	
13	デバイス		MVSLU	
14	サイド情報		MVSCPIC	
15	トランザクション・プログラム	7	MQSERIES	
16	LAN アダプター・アドレス	8	400074511092	
17	モード (Mode)	6	#INTER	
VSE/ESA システムとの接続				
9	ネットワーク ID	1	NETID	
10	制御点名	2	VSEPU	
11	LU 名	3	VSELU	
12	コントローラー記述		VSEPU	
13	デバイス		VSELU	
14	サイド情報		VSECPIC	
15	トランザクション・プログラム	4	MQ01	MQ01
16	LAN アダプター・アドレス	5	400074511092	
17	モード (Mode)		#INTER	

IBM i 用語の説明

構成ワークシートに使用されている項目の説明。

123

構成値を見つける方法については、25 ページの『ネットワーク属性を見つける方法』で詳しく説明します。

4 LAN 宛先アドレス (LAN destination address)

IBM i システム・トークンリング・アダプターのハードウェア・アドレス。DSPLIND *Line description* (6) コマンドで値を見つけることができます。

5 サブシステム記述

このパラメーターは、キュー・マネージャー使用中にアクティブになる IBM i サブシステムの名前です。QCMN という名前が使用されていますが、それは QCMN が IBM i 通信サブシステムであるためです。

6 回線記述

このパラメーターが指定されている場合、リソースの「リソース名」の「記述」フィールドにそのことが示されます。詳細については、[26 ページの『リソース名の値を見つける方法』](#)を参照してください。値が指定されていない場合は、回線記述を作成する必要があります。

7 リソース名

構成値を見つける方法については、[26 ページの『リソース名の値を見つける方法』](#)で詳しく説明します。

8 ローカル・トランザクション・プログラム名

このワークステーションとの会話を試行する IBM MQ アプリケーションは、受信側で実行するプログラムのシンボル名を指定します。この名前は、送信側のチャンネル定義に定義されています。分かりやすいように、トランザクション・プログラム名には MQSERIES をできるだけ使用するようにしてください。VSE/ESA へ接続している場合は、長さの上限が 4 バイトなので MQTP を使用するようにしてください。

詳細については、[ローカル IBM i システムにおけるリモート・キュー・マネージャー・プラットフォーム用の設定値](#)を参照してください。

12 コントローラー記述

このパラメーターは、パートナー・システムの「制御点名」(または「ノード名」)の別名です。分かりやすいように、この例ではパートナーの実名を使用しています。

13 装置

このパラメーターは、パートナー・システムの LU の別名です。分かりやすいように、この例ではパートナーの LU 名を使用しています。

14 サイド情報

このパラメーターは、CPI-C サイド情報プロファイルに付ける名前です。独自の名前を 8 文字で指定します。

IBM i ネットワーク属性を見つける方法

ローカル・ノードは、IBM i インストールの一部として、すでに部分的に構成されています。現在のネットワーク属性を表示するには、DSPNETA コマンドを入力します。

これらの属性値を変更する場合は、CHGNETA コマンドを使用します。変更を適用するには、IPL が必要になる場合もあります。

```

Display Network Attributes
System: AS400PU
Current system name . . . . . : AS400PU
Pending system name . . . . . :
Local network ID . . . . . : NETID
Local control point name . . . . . : AS400PU
Default local location . . . . . : AS400LU
Default mode . . . . . : BLANK
APPN node type . . . . . : *ENDNODE
Data compression . . . . . : *NONE
Intermediate data compression . . . . . : *NONE
Maximum number of intermediate sessions . . . . . : 200
Route addition resistance . . . . . : 128
Server network ID/control point name . . . . . : NETID NETCP

```

More...
Press Enter to continue.

F3=Exit F12=Cancel

「**Local network ID (ローカル・ネットワーク ID)**」(1)、「**Local control point name (ローカル制御点名)**」(2)、および「**Default local location (デフォルトのローカル・ロケーション)**」(3)の値がワークシートの値に一致していることを確認します。

IBM i リソース名の値を見つける方法

リソース名の値を見つけるには、WRKHDWRSC TYPE(*CMN)を入力して Enter キーを押します。

「Work with Communication Resources (通信リソースの処理)」パネルが表示されます。「リソース名」の値はトークンリング・ポートとして示されています。この例では、LIN041 となっています。

```

Work with Communication Resources
System: AS400PU
Type options, press Enter.
2=Edit 4=Remove 5=Work with configuration description
7=Add configuration description ...

```

構成

Opt	Resource	Description	Type	Description
	CC02			2636 Comm Processor
	LIN04			2636 LAN Adapter
	LIN041	TOKEN-RING		2636 Token-ring Port

Bottom
F3=Exit F5=Refresh F6=Print F11=Display resource addresses/statuses
F12=Cancel F23=More options

IBM i LU 6.2 接続の確立

ここでは、LU 6.2 接続の確立方法について説明します。

IBM i ローカル・ノード構成

ローカル・ノードを構成するには、回線記述を作成し、経路指定項目を追加する必要があります。

回線記述の作成

1. 回線記述をまだ作成していない場合は、CRTLINTRN コマンドを使用して作成します。
2. 「Line description (回線記述)」 (6) と 「Resource name (リソース名)」 (7) に値を指定します。

```
Create Line Desc (token-ring) (CRTLINTRN)

Type choices, press Enter.

回線記述 . . . . . TOKENRINGL Name
資源名 . . . . . LIN041 Name, *NWID
NWI タイプ . . . . . *FR* FR , *ATM
IPL 時のオンライン . . . . . *はい *いいえ、 *いいえ
オンへの構成変更待ち . . . . . *NOWAIT *NOWAIT、 15から180 (1 秒)
最大制御装置 . . . . . 40 - 1 -256
接続された NWI . . . . . *NONE 名前, *NONE

Bottom
F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel
F13=How to use this display F24=More keys
パラメーター LIND が必要です。 +
```

経路指定項目の追加

1. コマンド ADDRTGE を入力して Enter キーを押します。

```
Add Routing Entry (ADDRTGE)

Type choices, press Enter.

サブシステム記述 . . . . . QCMN 名
ライブラリー . . . . . *LIBL 名, *LIBL, *CURLIB
Routing entry sequence number . 1 1-9999
Comparison data:
「値の比較 . . . . .」 MQSERIES'

Starting position . . . . . 37 1-80
コールするプログラム . . . . . AMQCRC6B Name, *RTGDTA
ライブラリー . . . . . QMAS400 Name, * LI BL, *CURLIB
クラス . . . . . * SBSB 名、 *SBSB
ライブラリー . . . . . *LIBL 名, *LIBL, *CURLIB
最大アクティブ・ルーティング・ステップ。 *ノマックス 0-1000、 * ノマックス
Storage pool identifier . . . . . 1 1-10

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
パラメーター SBSB が必要です。 +
```

2. 「Subsystem description (サブシステム記述)」 (5) に値を指定し、さらに 「Routing entry sequence number (経路指定項目のシーケンス番号)」、「Compare value (値の比較)」 (8)、「Starting position (開始位置)」、「Program to call (呼び出すプログラム)」、および呼び出すプログラムが入っている 「Library (ライブラリー)」 に対して、この例で示されている値を指定します。
3. STRSBS *subsystem description* (5) コマンドを入力し、Enter キーを押します。

IBM i パートナー・ノードへの接続

パートナー・ノードに接続するには、コントローラー記述を作成し、装置記述を作成し、CPI-C サイド情報を作成し、APPC の通信項目を追加し、構成リスト項目を追加する必要があります。

この例は、Windows システムへの接続についてのもものですが、他のノードでも手順は同じです。

コントローラー記述の作成

1. コマンド行に CRTCTLAPPC と入力して、Enter キーを押します。

```
Create Ctl Desc (APPC) (CRTCTLAPPC)

Type choices, press Enter.

制御装置記述 . . . WINNTCP      Name
リンク・タイプ . . . . . * LAN *FAX, *FR, *IDLC,
*LAN...
IPL 時のオンライン . . . . . *なし *なし *なし

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys
パラメーター CTLD が必要です。 +
```

2. 「**Controller description (コントローラー記述)**」 (12) に値を設定し、「**Link type (リンク・タイプ)**」を *LAN に、「**Online at IPL (IPL でのオンライン)**」を *NO に設定します。
3. Enter キーを二度押して、F10 キーを押します。

```
Create Ctl Desc (APPC) (CRTCTLAPPC)

Type choices, press Enter.

制御装置記述 . . . > WINNTCP 名
リンク・タイプ . . . . . > *LAN *FAX, *FR, *IDLC, *LAN...
IPL 時のオンライン . . . . . > *なし *なし *なし
APPN 可能 . . . . . *はい *いいえ、 *いいえ
交換回線リスト . . . . . TOKENRINGL Name
+ for more values
最大フレーム・サイズ . . . . . * リンクタイプ 265-16393, 256, 265, 512...
リモート・ネットワーク ID . . . . . NETID      Name, *NETATR, *NONE, *ANY
リモート制御点 . . . . . WINNTCP      Name, *ANY
交換 ID . . . . . 00000000-FFFFFFFF
初期接続 . . . . . * ダイヤル * ダイヤル、 *
ダイヤル開始 . . . . . *リンク・タイプ *リンク・タイプ、 *IMMED、 *遅延
LAN リモート・アダプター・アドレス . . . . . 10005AFC5D83 000000000001-FFFFFFFFFFFFFFF
APPN CP セッション・サポート。 . . . . . *はい *いいえ、 *いいえ
APPN ノード・タイプ . . . . . *エンド・ノード *エンド・ノード、 *レンノード...
APPN transmission group number 1      1-20, *CALC
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

4. 「**Switched line list (交換回線のリスト)**」 (6)、「**Remote network identifier (リモート・ネットワーク ID)**」 (9)、「**Remote control point (リモート制御点)**」 (10)、「**LAN remote adapter address (LAN リモート・アダプター・アドレス)**」に値を指定します (16)。
5. Enter キーを押します。

装置記述の作成

1. コマンド CRTDEVAPPC を入力して、Enter キーを押します。

```

Create Device Desc (APPC) (CRTDEVAPPC)

Type choices, press Enter.

装置記述 . . . . WINNTLU      Name
リモート・ロケーション . . . . WINNTLU      Name
IPL 時のオンライン . . . . . *はい *いいえ、 *いいえ
ローカル・ロケーション . . . . . AS400LU      Name, *NETATR
リモート・ネットワーク ID . . . . . NETID      Name, *NETATR, *NONE
接続された制御装置 . . . . . WINNTCP      Name
モード . . . . . *NETATR 名、 *NETATR
+ for more values
メッセージ待ち行列 . . . . . QSYSOPR      Name, QSYSOPR
ライブラリー . . . . . *LIBL 名、 *LIBL, *CURLIB
APPN 可能 . . . . . *はい *いいえ、 *いいえ
Single session:
単一セッション対応。 . . . *いいえ *  いいえ、 *はい
会話の数。          1-512

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys
パラメーター DEVD が必要である。 +

```

2. 「**Device description (装置記述)**」 (13)、「**Remote location (リモート・ロケーション)**」 (11)、「**Local location (ローカル・ロケーション)**」 (3)、「**Remote network identifier (リモート・ネットワーク ID)**」 (9)、「**Attached controller (接続されているコントローラー)**」 (12) に値を指定します。

注: IBM i 自動構成サービスを利用することにより、コントローラー記述と装置記述の作成の手間を省くことができます。詳細については、IBM i 資料を参照してください。

CPI-C サイド情報の作成

1. CRTCSI と入力し、F10 キーを押します。

```

Create Comm Side Information (CRTCSI)

Type choices, press Enter.

サイド情報 . . . . NTCPIC      Name
ライブラリー . . . . . *CURLIB 名、 *CURLIB
リモート・ロケーション . . . . . WINNTLU      Name
トランザクション・プログラム . . . . . MQSERIES

テキストの「説明」 . . . . *ブランク

Additional Parameters

装置 . . . . . * LOC 名、 *LOC
ローカル・ロケーション . . . . . AS400LU      Name, *LOC, *NETATR
モード . . . . . #INTER 名前、 *NETATR
リモート・ネットワーク ID . . . . . NETID      Name, *LOC, *NETATR, *NONE
権限 . . . . . *LIBCRTAUT 名、 *LIBCRTAUT, *CHANGE...

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Parameter CSI required.

```

2. 「**Side information (サイド情報)**」 (14)、「**Remote location (リモート・ロケーション)**」 (11)、「**Transaction program (トランザクション・プログラム)**」 (15)、「**Local location (ローカル・ロケーション)**」 (3)、「**Mode (モード)**」、「**Remote network identifier (リモート・ネットワーク ID)**」 (9) に値を指定します。
3. Enter キーを押します。

APPC の通信項目の追加

1. コマンド行に ADDCMNE と入力して、Enter キーを押します。

```
Add Communications Entry (ADDCMNE)
Type choices, press Enter.
サブシステム記述 . . . . . QCMN 名
ライブラリー . . . . . *LIBL 名, *LIBL, *CURLIB
装置 . . . . . WINNTLU      Name, generic*, *ALL...
リモート・ロケーション . . . . . 名称
ジョブ記述 . . . . . *USRPRF 名, *USRPRF, *SBSD
ライブラリー . . . . .      Name, *LIBL, *CURLIB
デフォルト・ユーザー・プロファイル . . . . . *NONE の名前, *NONE, *SYS
モード . . . . .      *ANY 名前, *ANY
最大アクティブ・ジョブ数 . . . . . *ノマックス 0-1000、 * ノマックス

Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
Parameter SBSD required.
```

2. 「**Subsystem description (サブシステム記述)**」 (5) と 「**Device (装置)**」 (13) に値を指定して、Enter キーを押します。

構成リスト項目の追加

1. ADDCFGLE *APPNRMT と入力し、F4 キーを押します。

```
Add Configuration List Entries (ADDCFGLE)
Type choices, press Enter.
構成リストのタイプ . . . . . > *APPNRMT *APPNLCL、 *APPNRMT...
APPN remote location entry:
リモート・ロケーション名 . . . . . WINNTLU      Name, generic*, *ANY
リモート・ネットワーク ID 。 NETID      Name, *NETATR, *NONE
ローカル・ロケーション名 . . . . . AS400LU      Name, *NETATR
リモート制御点 . . . . . WINNTCP      Name, *NONE
制御点ネット ID . . . . . NETID      Name, *NETATR, *NONE
ロケーション・パスワード . . . . . *なし
セキュア・ロケーション . . . . . *なし *なし *なし
単一セッション . . . . . *なし *なし *なし
ローカルに制御されたセッション。 *なし *なし *なし
事前に確立されたセッション . . . . . *なし *なし *なし
項目の 'description' . . . . . *ブランク
会話の数。 101-512 まで
+ for more values

Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

2. 「**Remote location name (リモート・ロケーション名)**」 (11)、 「**Remote network identifier (リモート・ネットワーク ID)**」 (9)、 「**Local location name (ローカル・ロケーション名)**」 (3)、 「**Remote control point (リモート制御点)**」 (10)、 「**Control point net ID (制御点ネットワーク ID)**」 (9) に値を指定します。
3. Enter キーを押します。

IBM i 次のステップ

これで LU 6.2 接続は確立されました。 構成を完成させる準備ができました。

32 ページの『IBM MQ for IBM i の構成』へ進んでください。

IBM i TCP 接続の確立

TCP が既に構成されている場合は、これ以上の構成作業はありません。TCP/IP を構成していない場合は、TCP/IP インターフェースの追加、TCP/IP ループバック・インターフェースの追加、およびデフォルト経路の追加を行う必要があります。

TCP/IP インターフェースの追加

1. コマンド行に ADDTCPIFC と入力して、Enter キーを押します。

```
Add TCP/IP Interface (ADDTCPIFC)
Type choices, press Enter.

インターネット・アドレス . . . . . 19.22.11.55
回線記述 . . . . . TOKENRINGL Name, *LOOPBACK
サブネット・マスク . . . . . 255.255.0.0
サービスのタイプ . . . . . *通常 *マインドレー、 *MAXTHRPUT..
最大伝送単位。 *LIND 576-16388, *LIND
自動開始 . . . . . *はい *いいえ、 *いいえ
PVC logical channel identifier      001-FFF
+ for more values
X.25 アイドル・サーキット・タイムアウト。 60 年 - 600 年
X.25 最大仮想回線数。 64 ビット
X.25 DDN インターフェース . . . . . *なし *なし *なし
TRLAN ビット順序付け . . . . . *MSB *MSB, *LSB

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

2. 「IP アドレス」と「回線記述」およびマシンの「サブネット・マスク」を指定します。
3. Enter キーを押します。

TCP/IP ループバック・インターフェースの追加

1. コマンド行に ADDTCPIFC と入力して、Enter キーを押します。

```

Add TCP Interface (ADDTCPIFC)

Type choices, press Enter.

インターネット・アドレス . . . . . 127.0.0.1
回線記述 . . . . . *LOOPBACK 名, *LOOPBACK
サブネット・マスク . . . . . 255.0.0.0
サービスのタイプ . . . . . *通常 *マインドレー、 *MAXTHRPUT..
最大伝送単位。 *LIND 576-16388, *LIND
自動開始 . . . . . *はい *いいえ、 *いいえ
PVC logical channel identifier      001-FFF
+ for more values
X.25 アイドル・サーキット・タイムアウト。 60 年 - 600 年
X.25 最大仮想回線数。 64 ビット
X.25 DDN インターフェース . . . . . *なし *なし *なし
TRLAN ビット順序付け . . . . . *MSB *MSB, *LSB

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

2. IP アドレス、回線記述、およびサブネット・マスクの値を指定します。

デフォルト経路の追加

1. コマンド行に ADDTCP RTE と入力して、Enter キーを押します。

```

Add TCP Route (ADDTCP RTE)

Type choices, press Enter.

経路宛先 . . . . . * DF トラブル
サブネット・マスク . . . . . *なし
サービスのタイプ . . . . . *通常 *マインドレー、 *MAXTHRPUT。
ネクスト・ホップ . . . . . 19.2.3.4
最大伝送単位。 576-16388, *IFC

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Command prompting ended when user pressed F12.

```

2. ご使用のネットワークに適した値を入力して Enter キーを押すと、デフォルトの経路項目が作成されます。

次のステップ

これで TCP 接続は確立されました。構成を完成させる準備ができました。[32 ページの『IBM MQ for IBM i の構成』](#)へ進んでください。

IBM MQ for IBM i の構成

IBM MQ for IBM i を構成するには、WRKMQMQ コマンドを使用して構成メニューを表示します。

STRMQMLSR コマンドを使用して TCP チャネル・リスナーを開始します。

STRMQMCHL CHLNAME(channel_name) コマンドで送信側チャネルを開始します。

WRKMQMQ コマンドを使用して IBM MQ 構成メニューを表示します。

注: AMQ* エラーは、エラーが発生したジョブに関連するログに記録されています。WRKACTJOB コマンドでジョブのリストを表示します。QSYSWRK という名前のサブシステムのもとでジョブを探し出し、5 を入力しそのジョブの処理を行います。IBM MQ ログには、接頭部 AMQ が付きます。

IBM i キュー・マネージャーの作成

基本的な構成キュー・マネージャーをセットアップするには、以下の手順を使用します。

1. 最初にキュー・マネージャーを作成する必要があります。CRTMQM と入力して、Enter キーを押してください。

```
                                Create Message Queue Manager (CRTMQM)

Type choices, press Enter.

Message Queue Manager name . . .
テキストの「説明」 . . . . *ブランク

トリガー間隔 . . . . . 999999999 0-999999999
未配布メッセージ待ち行列 . . . . *なし

デフォルト伝送キュー。 *なし

最大ハンドルの限界 . . . . . 二百五十六 - 999999999
最大未コミット・メッセージ数。 1000 年 - 10000 年
デフォルトのキュー・マネージャー . . . . *なし *なし *なし

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

2. 「**Message Queue Manager name**」フィールドに、AS400 と入力します。「**Undelivered message queue (未送信メッセージ・キュー)**」フィールドに、DEAD.LETTER.QUEUE と入力します。
3. Enter キーを押します。
4. これで、STRMQM MQMNAME (AS400) と入力するとキュー・マネージャーが開始します。
5. 以下のパラメーターを使用して、未配布メッセージ・キューを作成します。(詳細および例については、33 ページの『[キューの定義](#)』を参照してください。)

```
Local Queue
Queue name :   DEAD.LETTER.QUEUE
Queue type  :   *LCL
```

IBM i キューの定義

キューを定義するには、CRTMQMQ コマンドを使用します。

コマンド行で CRTMQMQ と入力します。

```
Create MQM Queue (CRTMQMQ)
Type choices, press Enter.
キュー名 . . . . .
キュー・タイプ . . . . .      *LZ、 *LCL、 *RMT
```

```
Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Parameter QNAME required.
```

このパネルの2つのフィールドに値を入力して、Enter キーを押します。別のパネルに、他のパラメータの入力フィールドが表示されます。他のすべてのキュー属性には、デフォルト値を設定できます。

IBM i IBM iでのチャネルの定義

IBM iでは、CRTMQMCHL コマンドを使用してチャネルを定義できます。

コマンド行で CRTMQMCHL と入力します。

```
Create MQM Channel (CRTMQMCHL)
Type choices, press Enter.
Channel name . . . . .
Channel type . . . . .      *RCVR, *SDR, *SVR, *RQSTR
```

```
Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Parameter CHLNAME required.
```

このパネルの2つのフィールドに値を入力して、Enter キーを押します。別のパネルが表示され、そこで以前設定された他のパラメータに値を指定できます。他のすべてのチャネル属性には、デフォルト値を設定できます。

IBM i IBM i用のチャネル構成

チャネルを構成して、構成チャネルのサンプルを実装する必要があります。

このセクションでは、5 ページの『すべてのプラットフォームでの IBM MQ の構成例』で説明されているチャンネルを実装するために、IBM i キュー・マネージャーで構成を実行する方法について詳しく説明します。

IBM MQ for IBM i と IBM MQ for Windows の接続のための例が示されています。別のプラットフォーム上の IBM MQ に接続するには、Windows 用の値の代わりに、表にある該当する値を使用してください。

注：

1. 太字の語は、ユーザーが指定するもので、これらの例で使用される IBM MQ オブジェクトの名前を反映しています。ここで使用される名前を変更する場合は、このセクション全体でこれらのオブジェクトに行われている他の参照も必ず変更してください。他はすべてキーワードなので、記載されているとおりに入力してください。
2. IBM MQ チャンネルの ping コマンド (PNGMQMCHL) は対話的に実行されますが、チャンネルの開始によってバッチ・ジョブが実行依頼されます。チャンネルの ping は正常に完了するのにチャンネルが開始しないときは、ネットワークと IBM MQ 定義は正しく、バッチ・ジョブの IBM i 環境の定義に誤りがあると思われる場合があります。例えば、QSYS2 が、個人用ライブラリー・リストにだけでなくライブラリー・リストのシステム用の部分にも組み込まれていることを確認してください。

リストされたオブジェクトの作成方法の詳細と例については、33 ページの『キューの定義』と 34 ページの『IBM i でのチャンネルの定義』を参照してください。

表 4. IBM i 用構成ワークシート				
ID	パラメーター名	参照	使用例	ユーザー値
ローカル・ノードの定義				
A	キュー・マネージャー名		AS400	
B	ローカル・キュー名		AS400.LOCALQ	
IBM MQ for Windows との接続				
表のこのセクションで指定する値は、54 ページの『Windows 用のチャンネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	WINNT	
D	リモート・キュー名		WINNT.REMOTEQ	
E	リモート・システムのキュー名	B	WINNT.LOCALQ	
F	伝送キュー名		WINNT	
G	送信側 (SNA) チャンネル名		AS400.WINNT.SNA	
H	送信側 (TCP/IP) チャンネル名		AS400.WINNT.TCP	
I	受信側 (SNA) チャンネル名	G	WINNT.AS400.SNA	
J	受信側 (TCP/IP) チャンネル名	H	WINNT.AS400.TCP	
IBM MQ for AIX との接続				
表のこのセクションで指定する値は、11 ページの『AIX 用のチャンネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	AIX	
D	リモート・キュー名		AIX.REMOTEQ	
E	リモート・システムのキュー名	B	AIX.LOCALQ	
F	伝送キュー名		AIX	
G	送信側 (SNA) チャンネル名		AS400.AIX.SNA	

表 4. IBM i 用構成ワークシート (続き)				
ID	パラメーター名	参照	使用例	ユーザー値
H	送信側 (TCP/IP) チャンネル名		AS400AIX。TCP (TCP)	
I	受信側 (SNA) チャンネル名	G	AIX.AS400.SNA	
J	受信側 (TCP) チャンネル名	H	AIX.AS400.TCP	
MQSeries for Compaq Tru64 Unix との接続				
表のこのセクションで指定する値は、Compaq Tru64 UNIX システムで使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	DECUX	
D	リモート・キュー名		DECUX.REMOTEQ	
E	リモート・システムのキュー名	B	DECUX.LOCALQ	
F	伝送キュー名		DECUX	
H	送信側 (TCP) チャンネル名		DECUX.AS400.TCP	
J	受信側 (TCP) チャンネル名	H	AS400.DECUX.TCP	
IBM MQ for HP-UX との接続				
表のこのセクションで指定する値は、17 ページの『HP-UX 用のチャンネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	HPUX	
D	リモート・キュー名		HPUX.REMOTEQ	
E	リモート・システムのキュー名	B	HPUX.LOCALQ	
F	伝送キュー名		HPUX	
G	送信側 (SNA) チャンネル名		AS400.HPUX.SNA	
H	送信側 (TCP) チャンネル名		AS400.HPUX.TCP	
I	受信側 (SNA) チャンネル名	G	HPUX.AS400.SNA	
J	受信側 (TCP) チャンネル名	H	HPUX.AS400.TCP	
IBM MQ for Solaris との接続				
表のこのセクションで指定する値は、47 ページの『Solaris 用のチャンネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	SOLARIS	
D	リモート・キュー名		SOLARIS.REMOTEQ	
E	リモート・システムのキュー名	B	SOLARIS.LOCALQ	
F	伝送キュー名		SOLARIS	
G	送信側 (SNA) チャンネル名		AS400.SOLARIS.SNA	
H	送信側 (TCP/IP) チャンネル名		AS400.SOLARIS.TCP	
I	受信側 (SNA) チャンネル名	G	SOLARIS.AS400.SNA	
J	受信側 (TCP/IP) チャンネル名	H	SOLARIS.AS400.TCP	

ID	パラメーター名	参照	使用例	ユーザー値
表 4. IBM i 用構成ワークシート (続き)				
IBM MQ for Linux との接続				
表のこのセクションで指定する値は、42 ページの『Linux 用のチャネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	LINUX	
D	リモート・キュー名		LINUX.REMOTEQ	
E	リモート・システムのキュー名	B	LINUX.LOCALQ	
F	伝送キュー名		LINUX	
G	送信側 (SNA) チャネル名		AS400.LINUX.SNA	
H	送信側 (TCP/IP) チャネル名		AS400.LINUX.TCP	
I	受信側 (SNA) チャネル名	G	LINUX.AS400.SNA	
J	受信側 (TCP/IP) チャネル名	H	LINUX.AS400.TCP	
IBM MQ for z/OS との接続				
表のこのセクションで指定する値は、60 ページの『z/OS 用のチャネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	MVS	
D	リモート・キュー名		MVS.REMOTEQ	
E	リモート・システムのキュー名	B	MVS.LOCALQ	
F	伝送キュー名		MVS	
G	送信側 (SNA) チャネル名		AS400.MVS.SNA	
H	送信側 (TCP) チャネル名		AS400.MVS.TCP	
I	受信側 (SNA) チャネル名	G	MVS.AS400.SNA	
J	受信側 (TCP) チャネル名	H	MVS.AS400.TCP	
MQSeries for VSE/ESA との接続				
表のこのセクションで指定する値は、VSE/ESA システムで使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	VSE	
D	リモート・キュー名		VSE.REMOTEQ	
E	リモート・システムのキュー名	B	VSE.LOCALQ	
F	伝送キュー名		VSE	
G	送信側チャネル名		AS400.VSE.SNA	
I	受信側チャネル名	G	VSE.AS400.SNA	

IBM i IBM i での送信側チャネル定義
SNA および TCP の送信側チャネル定義の例です。

SNA の使用

Local Queue

```

Queue name : WINNT F
Queue type : *LCL
Usage : *TMQ

Remote Queue
Queue name : WINNT.REMOTEQ D
Queue type : *RMT
Remote queue : WINNT.LOCALQ E
Remote Queue Manager : WINNT C
Transmission queue : WINNT F

Sender Channel
Channel Name : AS400.WINNT.SNA G
Channel Type : *SDR
Transport type : *LU62
Connection name : WINNTCPIC 14
Transmission queue : WINNT F

```

TCP の使用

```

Local Queue
Queue name : WINNT F
Queue type : *LCL
Usage : *TMQ

Remote Queue
Queue name : WINNT.REMOTEQ D
Queue type : *RMT
Remote queue : WINNT.LOCALQ E
Remote Queue Manager : WINNT C
Transmission queue : WINNT F

Sender Channel
Channel Name : AS400.WINNT.TCP H
Channel Type : *SDR
Transport type : *TCP
Connection name : WINNT.tcpip.hostname
Transmission queue : WINNT F

```

IBM i IBM i での受信側チャネル定義
SNA および TCP の受信側チャネル定義の例です。

SNA の使用

```

Local Queue
Queue name : AS400.LOCALQ B
Queue type : *LCL

Receiver Channel
Channel Name : WINNT.AS400.SNA I
Channel Type : *RCVR
Transport type : *LU62

```

TCP の使用

```

Local Queue
Queue name : AS400.LOCALQ B
Queue type : *LCL

Receiver Channel
Channel Name : WINNT.AS400.TCP J
Channel Type : *RCVR
Transport type : *TCP

```

Linux Linux での MQ の構成例

このセクションでは IBM MQ for Linux から IBM MQ 製品への通信リンクをセットアップする方法を例を挙げて説明します。

以下のプラットフォームでの例が示されています。

- Windows
- AIX
- Compaq Tru64 UNIX
- HP-UX
- Solaris
- IBM i
- z/OS
- VSE/ESA

このセクションのバックグラウンド情報およびその使用方法については、[5 ページの『すべてのプラットフォームでの IBM MQ の構成例』](#)を参照してください。

Linux LU 6.2 接続の確立

このワークシートを使用して、この構成で使用する値を記録します。

注: このセクションの情報は、IBM MQ for Linux (x86 プラットフォーム) にのみ適用されます。IBM MQ for Linux (x86-64 プラットフォーム)、IBM MQ for Linux (zSeries s390x プラットフォーム)、または IBM MQ for Linux (Power プラットフォーム) には適用されません。

SNA over TCP/IP の構成に関する最新情報については、[Communications Server for Linux ライブラリーの資料](#)から、ご使用のバージョンの Linux の管理ガイドを参照してください。

Linux Linux での TCP 接続の確立

現在は、inet デーモン (INETD) 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。次に、inet デーモンまたは拡張 inet デーモンのいずれかを使用した TCP 接続の確立方法を説明します。

inet デーモン (INETD) の使用

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

TCP 接続を確立するには、次の手順に従ってください。

1. `/etc/services` というファイルを編集する。そ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 # MQSeries channel listener
```

注: この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。

2. `/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 ]
```

3. 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
```

4. コマンドを実行します。

```
kill -1 inetd processid
```

システムに複数のキュー・マネージャーがあるために複数のサービスが必要な場合は、追加のキュー・マネージャーごとに `/etc/services` と `inetd.conf` の両方に 1 行追加する必要があります。

以下に例を示します。

```
MQSeries1 1414/tcp
MQSeries2 1822/tcp
```

```
MQSeries1 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM1
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
```

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、[TCP リスナー・バックログ・オプションの使用](#)を参照してください。

Linux 上の `inetd` プロセスは、TCP ポートのインバウンド接続の速度を制限する場合があります。デフォルトは 60 秒間隔内で 40 の接続です。さらに高速であることが必要な場合は、ピリオド (.) に続けて新規制限を `inetd.conf` 内の適切なサービスの `nowait` パラメーターに付加して、60 秒間隔内のインバウンド接続の数に対する新規制限を指定します。例えば、60 秒間隔内で 500 の接続の制限の場合は、以下を使用します。

```
MQSeries stream tcp nowait.500 mqm / MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM1
```

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

拡張 inet デーモン (XINETD) の使用

次に、Red Hat Linux への拡張 `inet` デーモンの実装方法について説明します。別の Linux 配布版を使用している場合は、これらの説明を変更しなければならないことがあります。

TCP 接続を確立するには、次の手順に従ってください。

1. `/etc/services` というファイルを編集する。そ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 # MQSeries channel listener
```

注: この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。

2. XINETD 構成ディレクトリー `/etc/xinetd.d` に、IBM MQ という名前のファイルを作成します。このファイルに以下のスタanzasを追加します。

```
# IBM MQ service for XINETD
service MQSeries
{
    disable          = no
    flags            = REUSE
    socket_type      = stream
    wait             = no
    user             = mqm
    server           = MQ_INSTALLATION_PATH/bin/amqcrsta
    server_args      = -m queue.manager.name
    log_on_failure += USERID
}
```

3. 次のコマンドを実行して、拡張 `inet` デーモンを再始動します。

```
/etc/rc.d/init.d/xinetd restart
```

システムに複数のキュー・マネージャーがあるために複数のサービスが必要な場合は、追加のキュー・マネージャーごとに `/etc/services` に 1 行追加する必要があります。サービスごとに `/etc/xinetd.d` ディレクトリにファイルを作成するか、以前に作成した IBM MQ ファイルにスタンザを追加することができます。

Linux 上の `xinetd` プロセスは、TCP ポートのインバウンド接続の速度を制限する場合があります。デフォルトは 10 秒間隔内で 50 の接続です。さらに高速であることが必要な場合は、`xinetd` 構成ファイル内で 'CPS' 属性を指定することで、インバウンド接続の速度に対する新規制限を指定します。例えば、60 秒間隔内で 500 の接続の制限の場合は、以下を使用します。

```
cps = 500 60
```

次のステップ

これで TCP/IP 接続は確立されました。構成を完成させる準備ができました。41 ページの『IBM MQ for Linux 構成』へ進んでください。

Linux IBM MQ for Linux 構成

インストール・プロセスを開始する前に、まず、`mqm` ユーザー ID および `mqm` グループを作成済みであることを確認してから、パスワードを設定します。

次のコマンドを使用して、チャンネルを開始します。

```
runmqchl -c channel.name
```

注:

1. サンプル・プログラムは `MQ_INSTALLATION_PATH/samp` にインストールされています。ここで、`MQ_INSTALLATION_PATH` は IBM MQ がインストールされている上位のディレクトリを表します。
2. エラー・ログは、`/var/mqm/qmgrs/qmgrname/errors` に格納されています。
3. コマンド・インタープリター `runmqsc` を使用して、管理コマンドを入力している場合、行の終わりにある `+` は、次の行が継続していることを示します。必ず最後のパラメーターと継続文字との間にスペースを入れるようにします。

基本構成

1. 次のコマンドを使用して、UNIX プロンプトからキュー・マネージャーを作成します。

```
crtmqm -u dlqname -q linux
```

ここで、

linux

キュー・マネージャーの名前

-q

これがデフォルトのキュー・マネージャーになることを示す

-u dlqname

送達不能キューの名前を指定します。

このコマンドは、キュー・マネージャーおよび一連のデフォルト・オブジェクトを作成します。

2. 次のコマンドを使用して、UNIX プロンプトからキュー・マネージャーを開始します。

```
strmqm linux
```

ここで、*linux* は、作成時に、キュー・マネージャーに指定された名前です。

Linux Linux 用のチャネル構成

次のセクションでは、5 ページの『すべてのプラットフォームでの IBM MQ の構成例』に記載したチャネルを実装するために、Linux キュー・マネージャーで実行する構成について詳細に説明します。

各オブジェクトを作成するための MQSC コマンドが示されます。UNIX プロンプトから **runmqsc** を開始して各コマンドを順次入力するか、またはコマンドをコマンド・ファイルに組み込みます。

IBM MQ for Linux と IBM MQ for HP-UX の接続のための例が示されています。他のプラットフォーム上の IBM MQ に接続するには、HP-UX 用の値の代わりに表から該当する一連の値を使用します。

注: 太字の語は、ユーザーが指定するもので、これらの例で使用される IBM MQ オブジェクトの名前を反映しています。ここで使用される名前を変更する場合は、このセクション全体でこれらのオブジェクトに行われている他の参照も必ず変更してください。他はすべてキーワードなので、記載されているとおりに入力してください。

表 5. IBM MQ for Linux 用構成ワークシート				
ID	パラメーター名	参照	使用例	ユーザー値
ローカル・ノードの定義				
A	キュー・マネージャー名		LINUX	
B	ローカル・キュー名		LINUX.LOCALQ	
IBM MQ for Windows との接続				
表のこのセクションで指定する値は、54 ページの『Windows 用のチャネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	WINNT	
D	リモート・キュー名		WINNT.REMOTEQ	
E	リモート・システムのキュー名	B	WINNT.LOCALQ	
F	伝送キュー名		WINNT	
G	送信側 (SNA) チャネル名		LINUX.WINNT.SNA	
H	送信側 (TCP/IP) チャネル名		LINUX.WINNT.TCP	
I	受信側 (SNA) チャネル名	G	WINNT.LINUX.SNA	
J	受信側 (TCP) チャネル名	H	WINNT.LINUX.TCP	
IBM MQ for AIX との接続				
表のこのセクションで指定する値は、11 ページの『AIX 用のチャネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	AIX	
D	リモート・キュー名		AIX.REMOTEQ	
E	リモート・システムのキュー名	B	AIX.LOCALQ	
F	伝送キュー名		AIX	
G	送信側 (SNA) チャネル名		LINUX.AIX.SNA	
H	送信側 (TCP) チャネル名		LINUX.AIX.TCP	
I	受信側 (SNA) チャネル名	G	AIX.LINUX.SNA	

表 5. IBM MQ for Linux 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
J	受信側 (TCP) チャンネル名	H	AIX.LINUX.TCP	
MQSeries for Compaq Tru64 UNIX との接続				
表のこのセクションで指定する値は、Compaq Tru64 UNIX システムで使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	DECUX	
D	リモート・キュー名		DECUX.REMOTEQ	
E	リモート・システムのキュー名	B	DECUX.LOCALQ	
F	伝送キュー名		DECUX	
H	送信側 (TCP) チャンネル名		DECUX.LINUX.TCP	
J	受信側 (TCP) チャンネル名	H	LINUX.DECUX.TCP	
IBM MQ for HP-UX との接続				
表のこのセクションで指定する値は、 18 ページの表 2 で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	HPUX	
D	リモート・キュー名		HPUX.REMOTEQ	
E	リモート・システムのキュー名	B	HPUX.LOCALQ	
F	伝送キュー名		HPUX	
G	送信側 (SNA) チャンネル名		LINUX.HPUX.SNA	
H	送信側 (TCP) チャンネル名		LINUX.HPUX.TCP	
I	受信側 (SNA) チャンネル名	G	HPUX.LINUX.SNA	
J	受信側 (TCP/IP) チャンネル名	H	HPUX.LINUX.TCP	
IBM MQ for Solaris との接続				
表のこのセクションで指定する値は、 47 ページの表 6 で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	SOLARIS	
D	リモート・キュー名		SOLARIS.REMOTEQ	
E	リモート・システムのキュー名	B	SOLARIS.LOCALQ	
F	伝送キュー名		GIS	
G	送信側 (SNA) チャンネル名		LINUX.SOLARIS.SNA	
H	送信側 (TCP/IP) チャンネル名		LINUX.SOLARIS.TCP	
I	受信側 (SNA) チャンネル名	G	SOLARIS.LINUX.SNA	
J	受信側 (TCP/IP) チャンネル名	H	SOLARIS.LINUX.TCP	
IBM i IBM MQ for IBM i との接続				
IBM i 表のこのセクションで指定する値は、 35 ページの表 4 で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	AS400	
D	リモート・キュー名		AS400.REMOTEQ	

表 5. IBM MQ for Linux 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
E	リモート・システムのキュー名	B	AS400.LOCALQ	
F	伝送キュー名		AS400	
G	送信側 (SNA) チャンネル名		LINUX.AS400.SNA	
H	送信側 (TCP) チャンネル名		LINUX.AS400.TCP	
I	受信側 (SNA) チャンネル名	G	AS400.LINUX.SNA	
J	受信側 (TCP) チャンネル名	H	AS400.LINUX.TCP	
<p>z/OS IBM MQ for z/OS との接続</p> <p>z/OS 表のこのセクションで指定する値は、60 ページの表 8 で指定した値と一致させる必要があります。</p>				
C	リモート・キュー・マネージャー名	A	MVS	
D	リモート・キュー名		MVS.REMOTEQ	
E	リモート・システムのキュー名	B	MVS.LOCALQ	
F	伝送キュー名		MVS	
G	送信側 (SNA) チャンネル名		LINUX.MVS.SNA	
H	送信側 (TCP) チャンネル名		LINUX.MVS.TCP	
I	受信側 (SNA) チャンネル名	G	MVS.LINUX.SNA	
J	受信側 (TCP) チャンネル名	H	MVS.LINUX.TCP	
<p>MQSeries for VSE/ESA (IBM MQ for Linux (x86 プラットフォーム) のみ)</p> <p>このセクションの表の値は、VSE/ESA システムで使用した値と一致させる必要があります。</p>				
C	リモート・キュー・マネージャー名	A	VSE	
D	リモート・キュー名		VSE.REMOTEQ	
E	リモート・システムのキュー名	B	VSE.LOCALQ	
F	伝送キュー名		VSE	
G	送信側チャンネル名		LINUX.VSE.SNA	
I	受信側チャンネル名	G	VSE.LINUX.SNA	

Linux SNA を使用した IBM MQ for Linux (x86 プラットフォーム) 送信側チャンネル定義コーディングの例。

```
def ql (HPUX) +                               F
  usage(xmitq) +
  replace

def qr (HPUX.REMOTEQ) +                       D
  iname(HPUX.LOCALQ) +                       E
```

```

rqmname(HPUX) +          C
xmitq(HPUX) +           F
replace

def chl (LINUX.HPUX.SNA) chltype(sdr) +      G
trptype(lu62) +
conname('HPUXCPIC') +
xmitq(HPUX) +           F
replace

```

Linux SNA を使用した *IBM MQ for Linux* (x86 プラットフォーム) 受信側チャネル定義コーディングの例。

```

def ql (LINUX.LOCALQ) replace                B

def chl (HPUX.LINUX.SNA) chltype(rcvr) +    I
trptype(lu62) +
replace

```

Linux TCP を使用した *IBM MQ for Linux* 送信側チャネル定義コーディングの例。

```

def ql (HPUX) +                               F
usage(xmitq) +
replace

def qr (HPUX.REMOTEQ) +                       D
rname(HPUX.LOCALQ) +                         E
rqmname(HPUX) +                              C
xmitq(HPUX) +                                F
replace

def chl (LINUX.HPUX.TCP) chltype(sdr) +      H
trptype(tcp) +
conname(remote_tcpip_hostname) +
xmitq(HPUX) +                                F
replace

```

Linux TCP/IP を使用した *IBM MQ for Linux* 受信側チャネル定義コーディングの例。

```

def ql (LINUX.LOCALQ) replace                B

def chl (HPUX.LINUX.TCP) chltype(rcvr) +    J
trptype(tcp) +
replace

```

Solaris Solaris での MQ の構成例

このセクションでは IBM MQ for Solaris から IBM MQ 製品への通信リンクをセットアップする方法を例を挙げて説明します。

例は、以下のプラットフォーム上で示されています。

- Windows
- AIX
- HP Tru64 UNIX
- HP-UX
- Linux
- IBM i
- z/OS
- VSE/ESA

このセクションのバックグラウンド情報およびその使用方法については、[5 ページの『すべてのプラットフォームでの IBM MQ の構成例』](#)を参照してください。

Solaris SNAP-IX を使用した LU 6.2 接続の確立

SNAP-IX を使用した LU 6.2 接続の構成のためのパラメーター。

SNA over TCP/IP の構成に関する最新情報については、IBM オンライン資料「[Communications Server](#)」、MetaSwitch オンライン資料「[SNAP-IX Administration Guide](#)」および Sun オンライン資料「[Configuring Intersystem Communications \(ISC\)](#)」を参照してください。

Solaris TCP 接続の確立

TCP 接続の構成に関する情報と、次のステップについて説明します。

TCP 接続を確立するには、次の手順に従ってください。

1. /etc/services というファイルを編集する。

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。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      1414/tcp      # MQSeries channel listener
```

2. /etc/inetd.conf というファイルを編集する。以下に示す行がこのファイルにない場合には、次のとおりに入力してファイルに追加してください。

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
[-m queue.manager.name]
```

MQ_INSTALLATION_PATH は、IBM MQ 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。

3. 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
```

4. 以下の該当するコマンドを実行する:

- Solaris 9 の場合

```
kill -1 inetd processid
```

- Solaris 10 以降の場合

```
inetconv
```

次のステップ

これで TCP/IP 接続は確立されました。構成を完成させる準備ができました。[46 ページの『IBM MQ for Solaris 構成』](#)へ進んでください。

Solaris IBM MQ for Solaris 構成

構成を完了するために定義するチャネルについて説明します。

インストール・プロセスを開始する前に、まず、mqm ユーザーおよびグループを作成済みであることを確認してから、パスワードを設定します。

次のコマンドを使用して、チャネルを開始します。

```
runmqchl -c channel.name
```

注:

1. サンプル・プログラムは、MQ_INSTALLATION_PATH/samp にインストールされています。
MQ_INSTALLATION_PATH は、IBM MQ がインストールされている上位ディレクトリーを表します。
2. エラー・ログは、/var/mqm/qmgrs/qmgrname/errors に格納されています。
3. コマンド・インタプリタ **runmqsc** を使用して、管理コマンドを入力している場合、行の終わりにある + は、次の行が継続していることを示します。必ず最後のパラメーターと継続文字との間にスペースを入れるようにします。
4. SNA または LU 6.2 チャンネルの場合、通信ライブラリーをロードしようとしてエラーが生じるようであれば、ファイル liblu62.so が見つからない可能性があります。この問題の解決策としては、同ファイルの場所 (おそらく /opt/SUNWlu62) を LD_LIBRARY_PATH に追加することが考えられます。

基本構成

1. 次のコマンドを使用して、UNIX プロンプトからキュー・マネージャーを作成します。

```
crtmqm -u dlqname -q solaris
```

ここで、

solaris

キュー・マネージャーの名前

-q

これがデフォルトのキュー・マネージャーになることを示す

-u dlqname

未配布メッセージ・キューの名前を指定する

このコマンドは、キュー・マネージャーおよび一連のデフォルト・オブジェクトを作成します。

2. 次のコマンドを使用して、UNIX プロンプトからキュー・マネージャーを開始します。

```
strmqm solaris
```

ここで、**solaris** は、作成時に、キュー・マネージャーに指定された名前です。

Solaris Solaris 用のチャンネル構成

次のセクションでは、チャンネルを実装するために Solaris キュー・マネージャーで実行する構成について詳細に説明します。

説明される構成は、[5 ページの図 1](#) で説明するチャンネルの実装のためのものです。

各オブジェクトを作成するための MQSC コマンドが示されます。UNIX プロンプトから **runmqsc** を開始して各コマンドを順次入力するか、またはコマンドをコマンド・ファイルに組み込みます。

IBM MQ for Solaris と IBM MQ for Windows の接続のための例が示されています。他のプラットフォーム上の IBM MQ に接続するには、Windows 用の値の代わりに表から該当する一連の値を使用します。

注: 太字の語は、ユーザーが指定するもので、これらの例で使用される IBM MQ オブジェクトの名前を反映しています。ここで使用される名前を変更する場合は、このセクション全体でこれらのオブジェクトに行われている他の参照も必ず変更してください。他はすべてキーワードなので、記載されているとおりに入力してください。

表 6. IBM MQ for Solaris 用構成ワークシート

ID	パラメーター名	参照	使用例	ユーザー値
	ローカル・ノードの定義			

表 6. IBM MQ for Solaris 用構成ワークシート (続き)				
ID	パラメーター名	参照	使用例	ユーザー値
A	キュー・マネージャー名		SOLARIS	
B	ローカル・キュー名		SOLARIS.LOCALQ	
IBM MQ for Windows との接続				
表のこのセクションで指定する値は、 54 ページの表 7 で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	WINNT	
D	リモート・キュー名		WINNT.REMOTEQ	
E	リモート・システムのキュー名	B	WINNT.LOCALQ	
F	伝送キュー名		WINNT	
G	送信側 (SNA) チャンネル名		SOLARIS.WINNT.SNA	
H	送信側 (TCP/IP) チャンネル名		SOLARIS.WINNT.TCP	
I	受信側 (SNA) チャンネル名	G	WINNT.SOLARIS.SNA	
J	受信側 (TCP) チャンネル名	H	WINNT.SOLARIS.TCP	
IBM MQ for AIX との接続				
表のこのセクションで指定する値は、 11 ページの表 1 で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	AIX	
D	リモート・キュー名		AIX.REMOTEQ	
E	リモート・システムのキュー名	B	AIX.LOCALQ	
F	伝送キュー名		AIX	
G	送信側 (SNA) チャンネル名		SOLARIS.AIX.SNA	
H	送信側 (TCP) チャンネル名		SOLARIS.AIX.TCP	
I	受信側 (SNA) チャンネル名	G	AIX.SOLARIS.SNA	
J	受信側 (TCP) チャンネル名	H	AIX.SOLARIS.TCP	
MQSeries for Compaq Tru64 Unix との接続				
表のこのセクションで指定する値は、Compaq Tru64 UNIX システムで使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	DECUX	
D	リモート・キュー名		DECUX.REMOTEQ	
E	リモート・システムのキュー名	B	DECUX.LOCALQ	
F	伝送キュー名		DECUX	
H	送信側 (TCP) チャンネル名		DECUX.SOLARIS.TCP	
J	受信側 (TCP) チャンネル名	H	SOLARIS.DECUX.TCP	
IBM MQ for HP-UX との接続				
表のこのセクションで指定する値は、 18 ページの表 2 で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	HPUX	
D	リモート・キュー名		HPUX.REMOTEQ	

表 6. IBM MQ for Solaris 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
E	リモート・システムのキュー名	B	HPUX.LOCALQ	
F	伝送キュー名		HPUX	
G	送信側 (SNA) チャンネル名		SOLARIS.HPUX.SNA	
H	送信側 (TCP) チャンネル名		SOLARIS.HPUX.TCP	
I	受信側 (SNA) チャンネル名	G	HPUX.SOLARIS.SNA	
J	受信側 (TCP/IP) チャンネル名	H	HPUX.SOLARIS.TCP	

IBM MQ for Linux との接続

表のこのセクションで指定する値は、[42 ページの表 5](#) で指定した値と一致させる必要があります。

C	リモート・キュー・マネージャー名	A	LINUX	
D	リモート・キュー名		LINUX.REMOTEQ	
E	リモート・システムのキュー名	B	LINUX.LOCALQ	
F	伝送キュー名		LINUX	
G	送信側 (SNA) チャンネル名		SOLARIS.LINUX.SNA	
H	送信側 (TCP/IP) チャンネル名		SOLARIS.LINUX.TCP	
I	受信側 (SNA) チャンネル名	G	LINUX.SOLARIS.SNA	
J	受信側 (TCP/IP) チャンネル名	H	LINUX.SOLARIS.TCP	

IBM i IBM MQ for IBM i との接続

IBM i 表のこのセクションで指定する値は、[35 ページの表 4](#) で指定した値と一致させる必要があります。

C	リモート・キュー・マネージャー名	A	AS400	
D	リモート・キュー名		AS400.REMOTEQ	
E	リモート・システムのキュー名	B	AS400.LOCALQ	
F	伝送キュー名		AS400	
G	送信側 (SNA) チャンネル名		SOLARIS.AS400.SNA	
H	送信側 (TCP) チャンネル名		SOLARIS.AS400.TCP	
I	受信側 (SNA) チャンネル名	G	AS400.SOLARIS.SNA	
J	受信側 (TCP) チャンネル名	H	AS400.SOLARIS.TCP	

z/OS IBM MQ for z/OS との接続

z/OS 表のこのセクションで指定する値は、[60 ページの表 8](#) で指定した値と一致させる必要があります。

C	リモート・キュー・マネージャー名	A	MVS	
D	リモート・キュー名		MVS.REMOTEQ	
E	リモート・システムのキュー名	B	MVS.LOCALQ	

表 6. IBM MQ for Solaris 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
F	伝送キュー名		MVS	
G	送信側 (SNA) チャンネル名		SOLARIS.MVS.SNA	
H	送信側 (TCP) チャンネル名		SOLARIS.MVS.TCP	
I	受信側 (SNA) チャンネル名	G	MVS.SOLARIS.SNA	
J	受信側 (TCP) チャンネル名	H	MVS.SOLARIS.TCP	

MQSeries for VSE/ESA との接続

このセクションの表の値は、VSE/ESA システムで使用した値と一致させる必要があります。

C	リモート・キュー・マネージャー名	A	VSE	
D	リモート・キュー名		VSE.REMOTEQ	
E	リモート・システムのキュー名	B	VSE.LOCALQ	
F	伝送キュー名		VSE	
G	送信側チャンネル名		SOLARIS.VSE.SNA	
I	受信側チャンネル名	G	VSE.SOLARIS.SNA	

Solaris SNAP-IX SNA を使用した IBM MQ for Solaris 送信側チャンネル定義コーディングの例。

```
def q1 (WINNT) +                                     F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                             D
  rname(WINNT.LOCALQ) +                             E
  rqmname(WINNT) +                                   C
  xmitq(WINNT) +                                     F
  replace

def chl (SOLARIS.WINNT.SNA) chltype(sdr) +          G
  trptype(lu62) +
  conname('NTCPIC') +                               14
  xmitq(WINNT) +                                     F
  replace
```

Solaris SNA を使用した IBM MQ for Solaris 受信側チャンネル定義コーディングの例。

```
def q1 (SOLARIS.LOCALQ) replace                     B

def chl (WINNT.SOLARIS.SNA) chltype(rcvr) +        I
  trptype(lu62) +
  replace
```

Solaris TCP を使用した IBM MQ for Solaris 送信側チャンネル定義コーディングの例。

```
def q1 (WINNT) +                                     F
  usage(xmitq) +
  replace
```

```

def qr (WINNT.REMOTEQ) +                               D
  rname(WINNT.LOCALQ) +                               E
  rqmname(WINNT) +                                    C
  xmitq(WINNT) +                                       F
  replace

def chl (SOLARIS.WINNT.TCP) chltype(sdr) +            H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(WINNT) +                                       F
  replace

```

Solaris TCP/IP を使用した IBM MQ for Solaris 受信側チャネル定義コーディングの例。

```

def ql (SOLARIS.LOCALQ) replace                         B

def chl (WINNT.SOLARIS.TCP) chltype(rcvr) +          J
  trptype(tcp) +
  replace

```

Windows Windows での IBM MQ の構成例

このセクションでは、その他のプラットフォーム上で IBM MQ for Windows から IBM MQ 製品への通信リンクをセットアップする方法を例を挙げて説明します。

次のプラットフォーム上での通信リンクのセットアップが示されています。

- AIX
- HP Tru64 UNIX
- HP-UX
- Solaris
- Linux
- IBM i
- z/OS
- VSE/ESA

接続が確立されたら、いくつかのチャネルを定義して、構成を完成させる必要があります。構成のプログラムとコマンドの例は、53 ページの『[IBM MQ for Windows 構成](#)』で説明されています。

このセクションのバックグラウンド情報およびその使用方法については、5 ページの『[すべてのプラットフォームでの IBM MQ の構成例](#)』を参照してください。

Windows LU 6.2 接続の確立

AnyNet[®] SNA over TCP/IP の構成に関する情報の解説書。

AnyNet SNA over TCP/IP の構成に関する最新情報については、IBM オンライン資料 ([AnyNet SNA over TCP/IP](#)、[SNA ノード操作](#)、および [Communications Server for Windows](#)) を参照してください。

Windows TCP 接続の確立

Windows システムとともに出荷される TCP スタックには、*inet* デーモンまたはそれに相当するものは組み込まれていません。

IBM MQ の TCP リスナーを開始させるために使用される IBM MQ コマンドは、次のとおりです。

```
runmqtsr -t tcp
```

チャンネルを開始するには、必ずその前にリスナーを明示的に開始してください。これにより、インバウンド送信チャンネルからの要求にตอบสนองして受信チャンネルが自動的に開始されるようになります。

次のステップ

TCP/IP 接続が確立されたら、いつでも構成を完了できます。53 ページの『IBM MQ for Windows 構成』へ進んでください。

Windows NetBIOS 接続の確立

NetBIOS 接続は、キュー・マネージャーから開始します。このプログラムは、そのチャンネル定義で ConnectionName パラメーターを使用して、ターゲット・リスナーに接続します。

NetBIOS 接続を設定するには、次のステップに従います。

1. チャンネルの両端で、IBM MQ チャンネル・プロセスが使用するローカル NetBIOS 名をキュー・マネージャー構成ファイル qm.ini に指定します。例えば、送信側の Windows の NETBIOS スタンザは、一例として次のようになります。

```
NETBIOS:
LocalName=WNTNETB1
```

受信側は次のようになります。

```
NETBIOS:
LocalName=WNTNETB2
```

各 IBM MQ プロセスが必ず別々のローカル NetBIOS 名を使用するようにします。システム名は既に Windows で使用されているので、NetBIOS 名として使用しないでください。

2. チャンネルの送信側と受信側のシステムで使用している LAN アダプター番号を確認します。IBM MQ for Windows の論理アダプター番号 0 のデフォルトは、インターネット・プロトコル・ネットワークで実行される NetBIOS です。ネイティブの NetBIOS を使用するには、論理アダプター番号 1 を選択する必要があります。LAN アダプター番号の設定を参照してください。

Windows レジストリーの NETBIOS スタンザに正しい LAN アダプター番号を指定します。以下に例を示します。

```
NETBIOS:
AdapterNum=1
```

3. 送信側のチャンネル開始が作動するように、次の MQNAME 環境変数でローカル NetBIOS 名を指定します。

```
SET MQNAME=WNTNETB1I
```

この名前は、固有の名前である必要があります。

4. 送信側で、チャンネルの受信側で使用する NetBIOS 名を指定して、チャンネルを定義します。以下に例を示します。

```
DEFINE CHANNEL (WINNT.OS2.NET) CHLTYPE(SDR) +
TRPTYPE(NETBIOS) +
CONNNAME(WNTNETB2) +
XMITQ(OS2) +
MCATYPE(THREAD) +
REPLACE
```

Windows では送信側チャンネルをスレッドとして実行する必要があるため、必ずオプション MCATYPE(THREAD) を指定します。

5. 受信側で、対応する受信側チャンネルを定義します。以下に例を示します。

```
DEFINE CHANNEL (WINNT.OS2.NET) CHLTYPE(RCVR) +  
TRPTYPE(NETBIOS) +  
REPLACE
```

- 新しく定義したチャンネルはそれぞれ、新しいプロセスとしてではなくスレッドとして開始されるため、チャンネル・イニシエーターを開始します。

```
runmqchi
```

- 受信側で、IBM MQ リスナーを開始します。

```
runmqclsr -t netbios
```

オプションで、キュー・マネージャー名、NetBIOS ローカル名、セッション数、名前数、およびコマンド数に値を指定できます。NetBIOS 接続のセットアップの詳細については、[Windows での NetBIOS 接続の定義](#)を参照してください。

Windows IBM MQ for Windows 構成

構成のためのプログラムとコマンドの例です。

注:

- サンプル・プログラム AMQSBCG を使用して、キュー内のすべてのメッセージのコンテンツとヘッダーを表示できます。以下に例を示します。

```
AMQSBCG q_name qmgr_name
```

これは、キュー・マネージャー *qmgr_name* で定義されるキュー *q_name* のコンテンツを表示します。あるいは、IBM MQ エクスプローラー内でメッセージ・ブラウザーを使用できます。

- コマンドを使用して、コマンド・プロンプトからすべてのチャンネルを開始させることができます。

```
runmqchl -c channel.name
```

- エラー・ログは、ディレクトリー *MQ_INSTALLATION_PATH\mqgrs\qmgrname\errors* および *MQ_INSTALLATION_PATH\mqgrs\@system\errors* にあります。どちらの場合でも、最新メッセージは *amqerr01.log* の最後にあります。
MQ_INSTALLATION_PATH は、IBM MQ がインストールされている上位ディレクトリーを表します。
- コマンド・インタープリター **runmqsc** を使用して、管理コマンドを入力している場合、行の終わりにある **+** は、次の行が継続していることを示します。必ず最後のパラメーターと継続文字との間にスペースを入れるようにします。

Windows デフォルト構成

IBM MQ Postcard アプリケーションのガイダンスに従って、デフォルト構成を作成するプロセスを実行できます。

Postcard アプリケーションの使用方法について詳しくは、『*Postcard* アプリケーションを使用したインストールの検査』で、企業で使用しているプラットフォームについて参照してください。

Windows 基本構成

キュー・マネージャーの作成と開始は、IBM MQ Explorer を使用して行うことも、コマンド・プロンプトから実行することもできます。

コマンド・プロンプトを選択した場合は、次のようにします。

- 次のコマンドを使用して、キュー・マネージャーを作成します。

```
crtmqm -u dlqname -q winnt
```

ここで、

winnt

キュー・マネージャーの名前

-q

これがデフォルトのキュー・マネージャーになることを示す

-u dlqname

未配布メッセージ・キューの名前を指定する

このコマンドは、キュー・マネージャーおよび一連のデフォルト・オブジェクトを作成します。

2. 次のコマンドを使用して、キュー・マネージャーを起動します。

```
strmqm winnt
```

ここで、*winnt* は、作成時に、キュー・マネージャーに指定された名前です。

Windows Windows 用のチャネル構成

指定されたチャネルを実装するために Windows キュー・マネージャーで実行される構成の例です。

次のセクションでは、5 ページの『すべてのプラットフォームでの IBM MQ の構成例』で説明されたチャネルを実現するために Windows キュー・マネージャーで実行される構成について詳細に説明します。

それぞれの構成について該当する MQSC コマンドを示しています。コマンド・プロンプトから **runmqsc** を開始して、次に各コマンドを入力するか、またはコマンドをコマンド・ファイルに組み込みます。

IBM MQ for Windows と IBM MQ for AIX の接続のための例が示されています。他のプラットフォーム上の IBM MQ に接続するには、Windows 用の値の代わりに表から該当する一連の値を使用します。

注: 太字の語は、ユーザーが指定するもので、これらの例で使用される IBM MQ オブジェクトの名前を反映しています。ここで使用される名前を変更する場合は、このセクション全体でこれらのオブジェクトに行われている他の参照も必ず変更してください。他はすべてキーワードなので、記載されているとおりに入力してください。

表 7. IBM MQ for Windows 用構成ワークシート

	パラメーター名	参照	使用例	ユーザー値
ローカル・ノードの定義				
A	キュー・マネージャー名		WINNT	
B	ローカル・キュー名		WINNT.LOCALQ	
IBM MQ for AIX との接続				
表のこのセクションで指定する値は、11 ページの『AIX 用のチャネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	AIX	
D	リモート・キュー名		AIX.REMOTEQ	
E	リモート・システムのキュー名	B	AIX.LOCALQ	
F	伝送キュー名		AIX	
G	送信側 (SNA) チャネル名		WINNT.AIX.SNA	
H	送信側 (TCP) チャネル名		WINNT.AIX.TCP	
I	受信側 (SNA) チャネル名	G	AIX.WINNT.SNA	

表 7. IBM MQ for Windows 用構成ワークシート (続き)

	パラメーター名	参照	使用例	ユーザー値
J	受信側 (TCP) チャンネル名	H	AIX.WINNT.TCP	
IBM MQ for HP Tru64 UNIX との接続				
表のこのセクションで指定する値は、HP Tru64 UNIX システムで使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	DECUX	
D	リモート・キュー名		DECUX.REMOTEQ	
E	リモート・システムのキュー名	B	DECUX.LOCALQ	
F	伝送キュー名		DECUX	
H	送信側 (TCP) チャンネル名		DECUX.WINNT.TCP	
J	受信側 (TCP) チャンネル名	H	WINNT.DECUX.TCP	
IBM MQ for HP-UX との接続				
表のこのセクションで指定する値は、17 ページの『HP-UX 用のチャンネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	HPUX	
D	リモート・キュー名		HPUX.REMOTEQ	
E	リモート・システムのキュー名	B	HPUX.LOCALQ	
F	伝送キュー名		HPUX	
G	送信側 (SNA) チャンネル名		WINNT.HPUX.SNA	
H	送信側 (TCP) チャンネル名		WINNT.HPUX.TCP	
I	受信側 (SNA) チャンネル名	G	HPUX.WINNT.SNA	
J	受信側 (TCP/IP) チャンネル名	H	HPUX.WINNT.TCP	
IBM MQ for Solaris との接続				
表のこのセクションで指定する値は、47 ページの『Solaris 用のチャンネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	SOLARIS	
D	リモート・キュー名		SOLARIS.REMOTEQ	
E	リモート・システムのキュー名	B	SOLARIS.LOCALQ	
F	伝送キュー名		SOLARIS	
G	送信側 (SNA) チャンネル名		WINNT.SOLARIS.SNA	
H	送信側 (TCP) チャンネル名		WINNT.SOLARIS.TCP	
I	受信側 (SNA) チャンネル名	G	SOLARIS.WINNT.SNA	
J	受信側 (TCP) チャンネル名	H	SOLARIS.WINNT.TCP	
IBM MQ for Linux との接続				
表のこのセクションで指定する値は、42 ページの『Linux 用のチャンネル構成』で指定した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	LINUX	

表 7. IBM MQ for Windows 用構成ワークシート (続き)

	パラメーター名	参照	使用例	ユーザー値
D	リモート・キュー名		LINUX.REMOTEQ	
E	リモート・システムのキュー名	B	LINUX.LOCALQ	
F	伝送キュー名		LINUX	
G	送信側 (SNA) チャンネル名		WINNT.LINUX.SNA	
H	送信側 (TCP) チャンネル名		WINNT.LINUX.TCP	
I	受信側 (SNA) チャンネル名	G	LINUX.WINNT.SNA	
J	受信側 (TCP) チャンネル名	H	LINUX.WINNT.TCP	
<p>IBM i IBM MQ for IBM i との接続</p> <p>IBM i 表のこのセクションで指定する値は、34 ページの『IBM i 用のチャンネル構成』で指定した値と一致させる必要があります。</p>				
C	リモート・キュー・マネージャー名	A	AS400	
D	リモート・キュー名		AS400.REMOTEQ	
E	リモート・システムのキュー名	B	AS400.LOCALQ	
F	伝送キュー名		AS400	
G	送信側 (SNA) チャンネル名		WINNT.AS400.SNA	
H	送信側 (TCP) チャンネル名		WINNT.AS400.TCP	
I	受信側 (SNA) チャンネル名	G	AS400.WINNT.SNA	
J	受信側 (TCP) チャンネル名	H	AS400.WINNT.TCP	
<p>z/OS IBM MQ for z/OS との接続</p> <p>z/OS 表のこのセクションで指定する値は、60 ページの『z/OS 用のチャンネル構成』で指定した値と一致させる必要があります。</p>				
C	リモート・キュー・マネージャー名	A	MVS	
D	リモート・キュー名		MVS.REMOTEQ	
E	リモート・システムのキュー名	B	MVS.LOCALQ	
F	伝送キュー名		MVS	
G	送信側 (SNA) チャンネル名		WINNT.MVS.SNA	
H	送信側 (TCP) チャンネル名		WINNT.MVS.TCP	
I	受信側 (SNA) チャンネル名	G	MVS.WINNT.SNA	
J	受信側 (TCP/IP) チャンネル名	H	MVS.WINNT.TCP	

表 7. IBM MQ for Windows 用構成ワークシート (続き)

	パラメーター名	参照	使用例	ユーザー値
<p>z/OS キュー共有グループを使用した IBM MQ for z/OS との接続</p> <p>z/OS 表のこのセクションで指定する値は、68 ページの『共有チャンネル構成の例』で指定した値と一致させる必要があります。</p>				
C	リモート・キュー・マネージャー名	A	QSG	
D	リモート・キュー名		QSG.REMOTEQ	
E	リモート・システムのキュー名	B	QSG.SHAREDQ	
F	伝送キュー名		QSG	
G	送信側 (SNA) チャンネル名		WINNT.QSG.SNA	
H	送信側 (TCP) チャンネル名		WINNT.QSG.TCP	
I	受信側 (SNA) チャンネル名	G	QSG.WINNT.SNA	
J	受信側 (TCP/IP) チャンネル名	H	QSG.WINNT.TCP	
<p>MQSeries for VSE/ESA との接続</p> <p>このセクションの表の値は、VSE/ESA システムで使用した値と一致させる必要があります。</p>				
C	リモート・キュー・マネージャー名	A	VSE	
D	リモート・キュー名		VSE.REMOTEQ	
E	リモート・システムのキュー名	B	VSE.LOCALQ	
F	伝送キュー名		VSE	
G	送信側チャンネル名		WINNT.VSE.SNA	
I	受信側チャンネル名	G	VSE.WINNT.SNA	

Windows SNA を使用した **IBM MQ for Windows** 送信側チャンネル定義
サンプル・コードです。

```
def ql (AIX) +                                     F
  usage(xmitq) +
  replace

def qr (AIX.REMOTEQ) +                             D
  rname(AIX.LOCALQ) +                             E
  rqmname(AIX) +                                   C
  xmitq(AIX) +                                     F
  replace

def chl (WINNT.AIX.SNA) chltype(sdr) +            G
  trptype(lu62) +                                  18
  conname(AIXCPIC) +                               F
  xmitq(AIX) +
  replace
```

Windows SNA を使用した **IBM MQ for Windows** 受信側チャンネル定義
サンプル・コードです。

```
def ql (WINNT.LOCALQ) replace                       B
```

```
def chl (AIX.WINNT.SNA) chltype(rcvr) +      I
  trptype(lu62) +
  replace
```

Windows TCP/IP を使用した IBM MQ for Windows 送信側チャネル定義
サンプル・コードです。

```
def ql (AIX) +                               F
  usage(xmitq) +
  replace

def qr (AIX.REMOTEQ) +                       D
  rname(AIX.LOCALQ) +                       E
  rqmname(AIX) +                             C
  xmitq(AIX) +                               F
  replace

def chl (WINNT.AIX.TCP) chltype(sdr) +      H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(AIX) +                               F
  replace
```

Windows TCP を使用した IBM MQ for Windows 受信側チャネル定義
サンプル・コードです。

```
def ql (WINNT.LOCALQ) replace               B

def chl (AIX.WINNT.TCP) chltype(rcvr) +    J
  trptype(tcp) +
  replace
```

Windows 自動始動

IBM MQ for Windows では、キュー・マネージャーとそのチャネル・イニシエーター、チャネル、リスナー、およびコマンド・サーバーの開始を自動化できます。

キュー・マネージャーのサービスを定義するには、IBM MQ Services スナップインを使用します。通信セットアップのテストが正常に終了したら、スナップイン内で該当するサービスを「自動」に設定します。このファイルは、システム始動時に、指定された IBM MQ サービスによって読み取られます。

詳しくは、[IBM MQ の管理](#)を参照してください。

Windows プロセスまたはスレッドとしてのチャネルの実行

IBM MQ for Windows は、送信チャネルを Windows プロセスまたは Windows スレッドとして実行する柔軟性を備えています。どちらで実行するかは、送信側チャネル定義の MCATYPE パラメーターに指定します。

ほとんどのインストール環境では、多数の同時チャネル接続をサポートするために必要な実メモリと仮想メモリがより少なくすむので、送信チャネルはスレッドとして実行されます。ただし、NetBIOS 接続では、送信メッセージ・チャネル・エージェント用に別個プロセスが必要とされます。

z/OS での MQ の構成例

このセクションでは、その他のプラットフォーム上で IBM MQ for z/OS から IBM MQ 製品への通信リンクをセットアップする方法を例を挙げて説明します。

この例でカバーされているその他のプラットフォームは、次のとおりです。

- Windows
- AIX
- Compaq Tru64 UNIX
- HP-UX
- Solaris

- Linux
- IBM i
- VSE/ESA

次のどの接続も可能です。

- z/OS から z/OS
- z/OS から MVS
- MVS から MVS

このセクションのバックグラウンド情報およびその使用方法については、5 ページの『すべてのプラットフォームでの IBM MQ の構成例』を参照してください。

z/OS 接続の確立

接続を確立するには、構成が必要なものがいくつかあります。

LU 6.2 接続の確立

SNA over TCP/IP の構成の最新情報については、IBM オンライン資料 [Communications Server for z/OS](#) を参照してください。

TCP 接続の確立

キュー・マネージャー・オブジェクトを変更し、以下のコマンドを実行して正しい分散キューイング・パラメーターを使用します。TCPNAME キュー・マネージャー属性に TCP アドレス・スペース名を追加する必要があります。

```
ALTER QMGR TCPNAME(TCPIP)
```

これで TCP 接続は確立されました。構成を完成させる準備ができました。

IBM MQ for z/OS の構成

次のステップでは、IBM MQ の構成方法、およびチャネルとリスナーの開始と構成について概説します。

1. 次のコマンドを使用して、チャネル・イニシエーターを開始します。

```
/cpf START CHINIT 1
```

2. 次のコマンドを使用して、LU 6.2 リスナーを開始します。

```
/cpf START LSTR LUNAME( M1 ) TRPTYPE(LU62)
```

LUNAME の M1 は、LU に指定したシンボル名 (5) です。TRPTYPE(LU62) を指定する必要があります。指定しないと、リスナーは TCP が必要とされていると想定します。

3. 次のコマンドを使用して、TCP リスナーを開始します。

```
/cpf START LSTR
```

1414 (デフォルト値の IBM MQ ポート) 以外のポートを使用する場合は、次のコマンドを使用します。

```
/cpf START LSTR PORT( 1555 )
```

チャネル折衝が、送信側と受信側で異なるメッセージ・シーケンス番号を検出すると、IBM MQ チャネルは正常に初期化しません。これらのチャネルは、手動でリセットすることが必要な場合があります。

z/OS 用のチャネル構成

サンプルのチャネルを実装するために、z/OS キュー・マネージャーで必要とされる構成がいくつかあります。

次のセクションでは、5 ページの『すべてのプラットフォームでの IBM MQ の構成例』で説明されたチャネルを実現するために z/OS キュー・マネージャーで実行される構成について詳細に説明します。

IBM MQ for z/OS と IBM MQ for Windows の接続のための例が示されています。他のプラットフォーム上の IBM MQ に接続するには、Windows 用の値の代わりに表から該当する一連の値を使用します。

注：太字の語は、ユーザーが指定するもので、これらの例で使用される IBM MQ オブジェクトの名前を反映しています。ここで使用される名前を変更する場合は、このセクション全体でこれらのオブジェクトに行われている他の参照も必ず変更してください。他はすべてキーワードなので、記載されているとおりに入力してください。

ID	パラメーター名	参照	使用例	ユーザー値
ローカル・ノードの定義				
A	キュー・マネージャー名		MVS	
B	ローカル・キュー名		MVS.LOCALQ	
IBM MQ for Windows との接続				
表のこのセクションで指定する値は、54 ページの『Windows 用のチャネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	WINNT	
D	リモート・キュー名		WINNT.REMOTEQ	
E	リモート・システムのキュー名	B	WINNT.LOCALQ	
F	伝送キュー名		WINNT	
G	送信側 (LU 6.2) チャネル名		MVS.WINNT.SNA	
H	送信側 (TCP) チャネル名		MVS.WINNT.TCP	
I	受信側 (LU 6.2) チャネル名	G	WINNT.MVS.SNA	
J	受信側 (TCP/IP) チャネル名	H	WINNT.MVS.TCP	
IBM MQ for AIX との接続				
表のこのセクションで指定する値は、11 ページの『AIX 用のチャネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	AIX	
D	リモート・キュー名		AIX.REMOTEQ	
E	リモート・システムのキュー名	B	AIX.LOCALQ	
F	伝送キュー名		AIX	
G	送信側 (LU 6.2) チャネル名		MVS.AIX.SNA	
H	送信側 (TCP/IP) チャネル名		MVS.AIX.TCP	
I	受信側 (LU 6.2) チャネル名	G	AIX.MVS.SNA	
J	受信側 (TCP/IP) チャネル名	H	AIX.MVS.TCP	

表 8. IBM MQ for z/OS 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
MQSeries for Compaq Tru64 Unix との接続				
表のこのセクションで指定する値は、Compaq Tru64 UNIX システムで使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	DECUX	
D	リモート・キュー名		DECUX.REMOTEQ	
E	リモート・システムのキュー名	B	DECUX.LOCALQ	
F	伝送キュー名		DECUX	
H	送信側 (TCP) チャンネル名		DECUX.MVS.TCP	
J	受信側 (TCP) チャンネル名	H	MVS.DECUX.TCP	
IBM MQ for HP-UX との接続				
表のこのセクションで指定する値は、17 ページの『HP-UX 用のチャンネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	HPUX	
D	リモート・キュー名		HPUX.REMOTEQ	
E	リモート・システムのキュー名	B	HPUX.LOCALQ	
F	伝送キュー名		HPUX	
G	送信側 (LU 6.2) チャンネル名		MVS.HPUX.SNA	
H	送信側 (TCP) チャンネル名		MVS.HPUX.TCP	
I	受信側 (LU 6.2) チャンネル名	G	HPUX.MVS.SNA	
J	受信側 (TCP) チャンネル名	H	HPUX.MVS.TCP	
IBM MQ for Solaris との接続				
表のこのセクションで指定する値は、47 ページの『Solaris 用のチャンネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	SOLARIS	
D	リモート・キュー名		SOLARIS.REMOTEQ	
E	リモート・システムのキュー名	B	SOLARIS.LOCALQ	
F	伝送キュー名		SOLARIS	
G	送信側 (LU 6.2) チャンネル名		MVS.SOLARIS.SNA	
H	送信側 (TCP) チャンネル名		MVS.SOLARIS.TCP	
I	受信側 (LU 6.2) チャンネル名	G	SOLARIS.MVS.SNA	
J	受信側 (TCP/IP) チャンネル名	H	SOLARIS.MVS.TCP	
IBM MQ for Linux との接続				
表のこのセクションで指定する値は、42 ページの『Linux 用のチャンネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	LINUX	
D	リモート・キュー名		LINUX.REMOTEQ	

表 8. IBM MQ for z/OS 用構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
E	リモート・システムのキュー名	B	LINUX.LOCALQ	
F	伝送キュー名		LINUX	
G	送信側 (LU 6.2) チャネル名		MVS.LINUX.SNA	
H	送信側 (TCP) チャネル名		MVS.LINUX.TCP	
I	受信側 (LU 6.2) チャネル名	G	LINUX.MVS.SNA	
J	受信側 (TCP/IP) チャネル名	H	LINUX.MVS.TCP	
IBM MQ for IBM i との接続				
表のこのセクションで指定する値は、34 ページの『IBM i 用のチャネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	AS400	
D	リモート・キュー名		AS400.REMOTEQ	
E	リモート・システムのキュー名	B	AS400.LOCALQ	
F	伝送キュー名		AS400	
G	送信側 (LU 6.2) チャネル名		MVS.AS400.SNA	
H	送信側 (TCP/IP) チャネル名		MVS.AS400.TCP	
I	受信側 (LU 6.2) チャネル名	G	AS400.MVS.SNA	
J	受信側 (TCP/IP) チャネル名	H	AS400.MVS.TCP	
MQSeries for VSE/ESA との接続				
表のこのセクションで指定する値は、VSE/ESA システムで使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	VSE	
D	リモート・キュー名		VSE.REMOTEQ	
E	リモート・システムのキュー名	B	VSE.LOCALQ	
F	伝送キュー名		VSE	
G	送信側チャネル名		MVS.VSE.SNA	
I	受信側チャネル名	G	VSE.MVS.SNA	

IBM MQ for z/OS 送信側チャネル定義

このトピックでは、LU 6.2 または TCP を使用した IBM MQ for z/OS の構成に必要とされる送信側チャネル定義の詳細について説明します。

LU 6.2 の場合:

```

Local Queue
  Object type : QLOCAL
  Name       : WINNT           F
  Usage     : X (XmitQ)

Remote Queue
  Object type : QREMOTE
  Name       : WINNT.REMOTEQ  D
Name on remote system : WINNT.LOCALQ E
Remote system name  : WINNT   C
Transmission queue : WINNT   F
    
```

```

Sender Channel
Channel name : MVS.WINNT.SNA      G
Transport type : L (LU6.2)
Transmission queue name : WINNT   F
Connection name : M3              13

```

TCP の場合:

```

Local Queue
Object type : QLOCAL
Name : WINNT      F
Usage : X (XmitQ)

Remote Queue
Object type : QREMOTE
Name : WINNT.REMOTEQ  D
Name on remote system : WINNT.LOCALQ  E
Remote system name : WINNT           C
Transmission queue : WINNT           F

Sender Channel
Channel name : MVS.WINNT.TCP      H
Transport type : T (TCP)
Transmission queue name : WINNT   F
Connection name : winnt.tcpip.hostname

```

IBM MQ for z/OS 受信側チャネル定義

このトピックでは、LU 6.2 または TCP を使用した IBM MQ for z/OS の構成に必要とされる受信側チャネル定義の詳細について説明します。

LU 6.2 の場合:

```

Local Queue
Object type : QLOCAL
Name : MVS.LOCALQ      B
Usage : N (Normal)

Receiver Channel
Channel name : WINNT.MVS.SNA  I

```

TCP の場合:

```

Local Queue
Object type : QLOCAL
Name : MVS.LOCALQ      B
Usage : N (Normal)

Receiver Channel
Channel name : WINNT.MVS.TCP  J

```

z/OS

QSG を使用した z/OS での MQ の構成例

このセクションでは、Windows および AIX 上の IBM MQ 製品からキュー共有グループ (QSG) への通信リンクをセットアップする方法の例を示します。z/OS から z/OS にも接続できます。

キュー共有グループからプラットフォーム (z/OS 以外) への通信リンクのセットアップは、58 ページの『z/OS での MQ の構成例』の説明と同じです。そのセクションには他のプラットフォームについての例が記載されています。

接続が確立されたら、いくつかのチャネルを定義して、構成を完成させる必要があります。この操作については、68 ページの『IBM MQ for z/OS 共有チャネル構成』を参照してください。

このセクションのバックグラウンド情報およびその使用方法については、5 ページの『すべてのプラットフォームでの IBM MQ の構成例』を参照してください。

以下のワークシートは、z/OS システムから、他のいずれかの IBM MQ プラットフォームへの通信をセットアップするために必要なすべてのパラメーターをリストしています。このワークシートには、作業環境でテスト済みのパラメーターの例が示されているほか、ユーザー独自の値を入力するスペースもあります。

接続しているプラットフォームに関するセクションのワークシートと共に、このセクションのワークシートを使用します。

LU 6.2 接続のセットアップに必要な手順については、66 ページの『キュー共有グループへの LU 6.2 接続の確立』でワークシートのパラメーターを番号で相互参照しながら説明しています。

「参照」列にある番号は、このセクションに記載されている該当するワークシートの値と一致させる必要があります。また、このセクションに示されている各例は、「ID」列の値を参照しています。「パラメーター名」列の項目は、65 ページの『用語の説明』で説明します。

ID	パラメーター名	参照	使用例	ユーザー値
表 9. LU 6.2 を使用する z/OS 用の構成ワークシート				
汎用リソースを使用するローカル・ノードの定義				
1	コマンド接頭部		/cpf	
2	ネットワーク ID		NETID	
3	ノード名		MVSPU	
6	MODENAME		#INTER	
7	ローカル・トランザクション・プログラム名		MQSERIES	
8	LAN 宛先アドレス		400074511092	
9	ローカル LU 名		MVSLU1	
10	汎用リソース名		MVSGR	
11	シンボリック宛先		G1	
12	汎用リソース名のシンボリック宛先		G2	
Windows システムとの接続				
13	シンボリック宛先		M3	
14	MODENAME	21	#INTER	
15	リモート・トランザクション・プログラム名	7	MQSERIES	
16	Partner LU name	5	WINNTLU	
21	リモート・ノード ID	4	05D 30F65	
AIX システムとの接続				
13	シンボリック宛先		M4	
14	MODENAME	18	#INTER	
15	リモート・トランザクション・プログラム名	6	MQSERIES	
16	Partner LU name	4	AIXLU	

構成ワークシートに使用されている項目の説明。

1 コマンド接頭部

この項目は IBM MQ for z/OS キュー・マネージャー・サブシステムに固有のコマンド接頭部です。z/OS システム・プログラマーがインストール時にこの値を SYS1.PARMLIB(IEFSSNss) に定義します。システム・プログラマーからこの値の情報を得ることができます。

2 ネットワーク ID

インストール済み環境での VTAM 始動手順は、DDNAME VTAMLST によって参照されるデータ・セットのメンバー ATCSTRxx によって一部カスタマイズされます。ネットワーク ID は、このメンバーの NETID パラメーターに指定された値です。ネットワーク ID には、IBM MQ 通信サブシステムを所有している NETID の名前を指定する必要があります。ネットワーク管理者からこの値の情報を得ることができます。

3 ノード名

VTAM はローエンタリーのネットワーク・ノードで、拡張対等通信ネットワーク機能 (APPN) で使用するための制御点名はありません。ただし、システム・サービス制御点名 (SSCPNAME) があります。ノード名には、IBM MQ 通信サブシステムを所有している SSCP の名前を指定する必要があります。この値は同じメンバー ATCSTRxx で「ネットワーク ID」として定義されています。ネットワーク管理者からこの値の情報を得ることができます。

9 ローカル LU 名

論理装置 (LU) は、トランザクション・プログラムとネットワークの間でインターフェースまたは変換プログラムとして機能するソフトウェアです。LU は、トランザクション・プログラム間のデータ交換を管理します。ローカル LU 名は、この IBM MQ サブシステム固有の VTAM APPLID です。ネットワーク管理者からこの値の情報を得ることができます。

11 12 13 シンボリック宛先

この項目は、CPI-C サイド情報プロファイルに付ける名前です。各 LU 6.2 リスナーにサイド情報項目が必要です。

6 14 モード名

この項目は、LU 6.2 会話を制御するパラメーター・セットに付ける名前です。この名前および同様の属性をもつ項目を、各セッションの最後に定義する必要があります。VTAM では、これはモード・テーブル項目に対応します。このテーブル項目はネットワーク管理者によって割り当てられる場合があります。

7 15 トランザクション・プログラム名

このキュー・マネージャーとの会話を試行する IBM MQ アプリケーションは、受信側で実行するプログラムのシンボル名を指定します。このシンボル名は、送信側ではチャンネル定義の TPNAME 属性に指定されています。分かりやすいように、トランザクション・プログラム名には MQSERIES をできるだけ使用するようにしてください。VSE/ESA へ接続している場合は、長さの上限が 4 バイトなので MQTP を使用するようにしてください。

詳細については、[APPC/MVS を使用した z/OS 用の LU6.2 接続の定義](#)を参照してください。

8 LAN 宛先アドレス

この項目は、パートナー・ノードがこのホストと通信する際に使用する LAN 宛先アドレスです。3745 ネットワーク・コントローラーを使用している場合、このアドレスはパートナーが物理的に接続している回線の回線定義の LOCADD パラメーターに指定されている値です。パートナー・ノードが 317X や 6611 といった他のデバイスを使用する場合は、それらのデバイスをカスタマイズするときにアドレスが設定されます。ネットワーク管理者からこの値の情報を得ることができます。

10 汎用リソース名

汎用リソース名は、キュー共有グループでチャンネル・イニシエーターによって使用される LU 名グループに割り当てられる固有名です。

16 パートナー LU 名

この項目は、通信をセットアップするときに使用するシステム上の IBM MQ キュー・マネージャーの LU 名です。この値は、リモート・パートナーのサイド情報項目の中に指定されています。

21 リモート・ノード ID

Windows への接続の場合、この ID は通信をセットアップするときに使用する Windows システム上のローカル・ノードの ID です。

z/OS キュー共有グループへの LU 6.2 接続の確立

LU 6.2 接続を確立するには、2つのステップがあります。自身をネットワークに定義するステップと、接続をパートナーに定義するステップです。

z/OS 汎用リソースの使用によるネットワークへの自身の登録

VTAM 汎用リソースを使用して、キュー共有グループに接続するための1つの接続名を用意できます。

1. SYS1.PARMLIB(APPCPMxx) には、APPC の始動パラメーターが入っています。このファイルに行を追加して、そこで APPC に sideinfo を見つける場所を知らせる必要があります。この行は次の形式でなければなりません。

```
SIDEINFO
DATASET (APPC.APPCSI)
```

2. 別の行を SYS1.PARMLIB(APPCPMxx) に追加して、そこで IBM MQ LU 6.2 グループ・リスナーに使用するローカル LU 名を定義します。次のような形式で行を追加してください。

```
LUADD ACBNAME(mvslu1)
      NOSCHED
      TPDATA(csq.appctp)
      GRNAME(mvsgrr)
```

ACBNAME (9)、TPDATA、および GRNAME (10) に値を指定します。

NOSCHED パラメーターは、新規 LU が LU 6.2 スケジューラー (ASCH) を所有していても使用していないことを APPC に知らせます。TPDATA は、LU 6.2 がトランザクション・プログラムに関する情報を保管するトランザクション・プログラム・データ・セットです。ここでも IBM MQ はこのパラメーターを使用しませんが、LUADD コマンド構文に必要です。

3. 次のコマンドを指定して APPC サブシステムを始動してください。

```
START APPC, SUB=MSTR, APPC=xxx
```

ここで、*xxx* は、ステップ 1 で LU を追加した PARMLIB メンバーの接尾部です。

注: APPC がすでに実行されている場合は、次のコマンドで更新できます。

```
SET APPC=xx
```

この更新は累積されます。つまり、APPC は、このメンバーや別の PARMLIB メンバー内で APPC に対して今までに定義されたオブジェクトの知識をすべて保持します。

4. 新しい LU を適切な VTAM メジャー・ノード定義に追加します。このノード定義は、通常 SYS1.VTAMLST にあります。APPL 定義は、ここに示すサンプルのようになります。

```

MVSLU APPL ACBNAME=MVSLU1,      9
           APPXC=YES,
           AUTOSES=0,
           DDRAINL=NALLOW,
           DLOGMOD=#INTER,      6
           DMINWML=10,
           DMINWNR=10,
           DRESPL=NALLOW,
           DSESLIM=60,
           LMDENT=19,
           MODETAB=MTCICS,
           PARSESS=YES,
           VERIFY=NONE,
           SECACPT=ALREADYV,
           SRBEXIT=YES
```

5. メジャー・ノードを活動化します。この活動化には、次のコマンドを使用します。

V,NET,ACT,majornode

- LU および汎用リソース名を定義する項目を CPI-C サイド情報データ・セットに追加します。APPC ユーティリティー・プログラム ATBSDFMU を使用して、それを行います。JCL のサンプルは、*thlqual.SCSQPROC(CSQ4SIDE)* にあります (*thlqual* は、インストールでの IBM MQ データ・セットのターゲット・ライブラリー高位修飾子です)。

追加する項目は、次の例のような形式で入力します。

```
SIADD
  DESTNAME(G1)          11
  MODENAME(#INTER)
  TPNAME(MQSERIES)
  PARTNER_LU(MVSLU1)    9
SIADD
  DESTNAME(G2)          12
  MODENAME(#INTER)
  TPNAME(MQSERIES)
  PARTNER_LU(MVSGR)     10
```

- キュー・マネージャー・オブジェクトを変更し、以下のコマンドを実行して正しい分散キューイング・パラメーターを使用します。キュー・マネージャーの LUGROUP 属性にキュー・マネージャーに割り当てるローカル LU (9) を指定する必要があります。

```
ALTER QMGR LUGROUP(MVSLU1)
```

z/OS パートナーへの接続の定義

CPI-C サイド情報データ・セットに項目を追加することにより、パートナーへの接続を定義できます。

注：この例は Windows システムに接続するためのものですが、他のプラットフォームでも作業は同じです。

接続を定義するために、項目を CPI-C サイド情報データ・セットに追加します。この定義を実行する JCL のサンプルは、*thlqual.SCSQPROC(CSQ4SIDE)* にあります。

追加する項目は、次のような形式で入力します。

```
SIADD
  DESTNAME(M3)          13
  MODENAME(#INTER)     14
  TPNAME(MQSERIES)     15
  PARTNER_LU(WINNTLU)  16
```

z/OS 次のステップ

これで、接続は確立されました。構成を完成させる準備ができました。

68 ページの『IBM MQ for z/OS 共有チャネル構成』へ進んでください。

z/OS Sysplex Distributor を使用した TCP 接続の確立

シスプレックス・ディストリビューターをセットアップして、1つの接続名を使用してキュー共有グループに接続します。

- 次のように、分散 DVIPA アドレスを定義します。
 - IPCONFIG に DYNAMICXCF ステートメントを追加します。このステートメントは、動的に作成された XCF TCP/IP リンクを使用したイメージ間接続に使用されます。
 - シスプレックス内の各イメージ上で VIPADYNAMIC ブロックを使用します。
 - 所有イメージ上で、DVIPA を作成する VIPADefine ステートメントをコード化します。次に、それをその他のすべてのイメージまたは選択されたイメージに分散させるために VIPADISTRIBUTE ステートメントをコード化します。

ii) バックアップ・イメージ上で、DVIPA アドレスの VIPABACKUP ステートメントをコード化します。

2. シスプレックスの LPAR で複数のチャンネル・イニシエーターが開始される場合は、SHAREPORT オプションを追加して、ポートが PROFILE データ・セットの PORT 予約リスト内で共有されるようにします。

詳しくは、「z/OS CS: IP 構成ガイド」および「z/OS CS: IP 構成解説書」を参照してください。

シスプレックス・ディストリビューターが LPAR 間のインバウンド接続の平衡を取ります。LPAR に複数のチャンネル・イニシエーターがある場合、SHAREPORT により、そのインバウンド接続が接続の最小番号を持つリスナー・ポートに渡されます。

上記のステップを完了すると、TCP 接続が確立されます。構成を完成させる準備ができました。

68 ページの『IBM MQ for z/OS 共有チャンネル構成』へ進んでください。

z/OS IBM MQ for z/OS 共有チャンネル構成

チャンネル・イニシエーターを開始し、構成に適切なコマンドを実行することによって、共有チャンネルを構成します。

1. 次のコマンドを使用して、チャンネル・イニシエーターを開始します。

```
/cpf START CHINIT
```

2. 次のコマンドを使用して、LU6.2 グループ・リスナーを開始します。

```
/cpf START LSTR TRPTYPE(LU62) LUNAME( G1 ) INDISP(GROUP)
```

LUNAME の G1 は、LU に指定したシンボル名 (11) です。

3. シスプレックス・ディストリビューターで仮想 IP アドレッシングを使用していて、特定アドレス上で listen したい場合は、以下のコマンドを使用します。

```
/cpf START LSTR TRPTYPE(TCP) PORT(1555) IPADDR( musvipa ) INDISP(GROUP)
```

一度に実行される共有チャンネルのインスタンスは 1 つのみです。2 つ目のチャンネル・インスタンスを開始しようとしても失敗します (エラー・メッセージは、その他の要因によって異なります)。共有同期キューが、チャンネル状況を追跡します。

チャンネル折衝が、送信側と受信側で異なるメッセージ・シーケンス番号を検出すると、IBM MQ チャンネルは正常に初期化しません。これは、手動でリセットすることが必要な場合があります。

z/OS 共有チャンネル構成の例

共有チャンネルを構成するために実行する必要があるいくつかのステップがあります。

次のトピックでは、5 ページの『すべてのプラットフォームでの IBM MQ の構成例』で説明されたチャンネルを実現するために z/OS キュー・マネージャーで実行される構成について詳細に説明します。

IBM MQ for z/OS と Windows の接続のための例が示されています。他のプラットフォーム上の IBM MQ に接続するには、Windows 用の値の代わりに表から該当する一連の値を使用します。

注: 太字の語は、ユーザーが指定するもので、これらの例で使用される IBM MQ オブジェクトの名前を反映しています。ここで使用される名前を変更する場合は、このセクション全体でこれらのオブジェクトに行われている他の参照も必ず変更してください。他はすべてキーワードなので、記載されているとおりに入力してください。

ID	パラメーター名	参照	使用例	ユーザー値
ローカル・ノードの定義				
A	キュー・マネージャー名		QSG	

表 10. キュー共有グループを使用する IBM MQ for z/OS 用の構成ワークシート (続き)

ID	パラメーター名	参照	使用例	ユーザー値
B	ローカル・キュー名		QSG.SHAREDQ	
IBM MQ for Windows との接続				
表のこのセクションで指定する値は、54 ページの『Windows 用のチャンネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名	A	WINNT	
D	リモート・キュー名		WINNT.REMOTEQ	
E	リモート・システムのキュー名	B	WINNT.LOCALQ	
F	伝送キュー名		WINNT	
G	送信側 (LU 6.2) チャンネル名		QSG.WINNT.SNA	
H	送信側 (TCP) チャンネル名		QSG.WINNT.TCP	
I	受信側 (LU 6.2) チャンネル名	G	WINNT.QSG.SNA	
J	受信側 (TCP/IP) チャンネル名	H	WINNT.QSG.TCP	
IBM MQ for AIX との接続				
表のこのセクションで指定する値は、11 ページの『AIX 用のチャンネル構成』で使用した値と一致させる必要があります。				
C	リモート・キュー・マネージャー名		AIX	
D	リモート・キュー名		AIX.REMOTEQ	
E	リモート・システムのキュー名	B	AIX.LOCALQ	
F	伝送キュー名		AIX	
G	送信側 (LU 6.2) チャンネル名		QSG.AIX.SNA	
H	送信側 (TCP/IP) チャンネル名		QSG.AIX.TCP	
I	受信側 (LU 6.2) チャンネル名	G	AIX.QSG.SNA	
J	受信側 (TCP/IP) チャンネル名	H	AIX.QSG.TCP	

z/OS IBM MQ for z/OS 共有送信側チャンネル定義

LU 6.2 および TCP の共有送信側チャンネル定義の例です。

LU 6.2 の使用

```

Local Queue
  Object type : QLOCAL
  Name       : WINNT           F
  Usage      : X (XmitQ)
  Disposition : SHARED

  Remote Queue
  Object type : QREMOTE
  Name       : WINNT.REMOTEQ   D
  Name on remote system : WINNT.LOCALQ E
  Remote system name : WINNT   C
  Transmission queue : WINNT   F
  Disposition : GROUP

  Sender Channel
  Channel name : MVS.WINNT.SNA G
    
```

```

Transport type : L (LU6.2)
Transmission queue name : WINNT
Connection name : M3
Disposition : GROUP

```

TCP の使用

```

Local Queue
  Object type : QLOCAL
  Name : WINNT
  Usage : X (XmitQ)
  Disposition : SHARED
Remote Queue
  Object type : QREMOTE
  Name : WINNT.REMOTEQ
  Name on remote system : WINNT.LOCALQ
  Remote system name : WINNT
  Transmission queue : WINNT
  Disposition : GROUP
Sender Channel
  Channel name : QSG.WINNT.TCP
  Transport type : T (TCP)
  Transmission queue name : WINNT
  Connection name : winnt.tcpip.hostname
  Disposition : GROUP

```

z/OS IBM MQ for z/OS 共有受信側チャネル定義
 LU 6.2 および TCP の共有受信側チャネルを定義する例です。

LU 6.2 の使用

```

Local Queue
  Object type : QLOCAL
  Name : QSG.SHAREDQ
  Usage : N (Normal)
  Disposition : SHARED
Receiver Channel
  Channel name : WINNT.QSG.SNA
  Disposition : GROUP

```

TCP の使用

```

Local Queue
  Object type : QLOCAL
  Name : QSG.SHAREDQ
  Usage : N (Normal)
  Disposition : SHARED
Receiver Channel
  Channel name : WINNT.QSG.TCP
  Disposition : GROUP

```

z/OS グループ内キューイングを使用した z/OS での MQ の構成例

このセクションでは、現在分散キューイングを使用するキュー・マネージャー間で小規模なメッセージ転送している典型的な給与計算照会アプリケーションをマイグレーションして、キュー共有グループと共有キューを使用する方法を説明します。

分散キューイング、共有キューを備えたグループ内キューイング、および共有キューの使用法を示すのに、3種類の構成の説明を記載しています。これに関連した図では、キュー・マネージャー QMG1 から QMG3 に向かう一方向のみのデータのフローを示しています。

z/OS 構成 1

構成 1 では、キュー・マネージャー QMG1 と QMG3 がメッセージをやりとりするのに分散キューイングを現在どのように使用しているかを説明します。

構成 1 は、キュー・マネージャー QMG1 で受信された給与計算照会からのメッセージをキュー・マネージャー QMG2 に転送し、最終的にはキュー・マネージャー QMG3 に転送して給与計算サーバーに送信するのに使用される分散キューイング・システムを示しています。

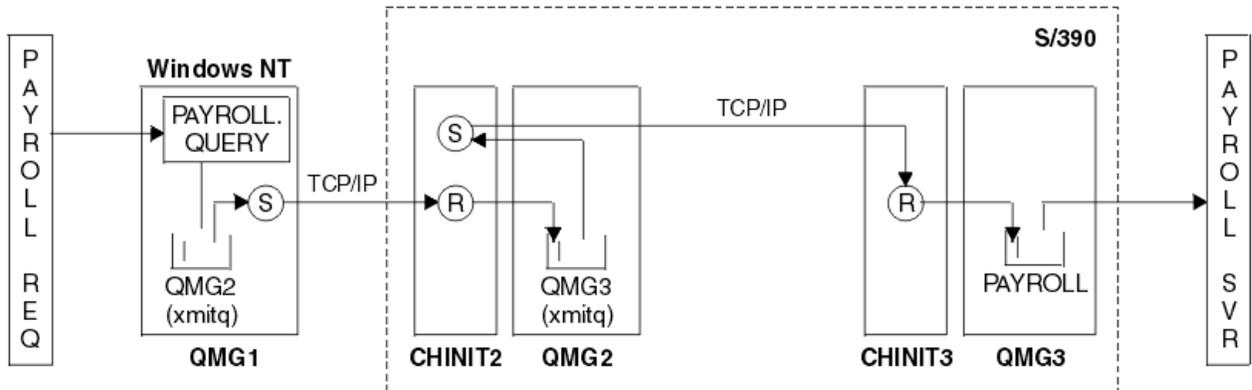


図 2. 構成 1: グループ内キューイングを使用する z/OS

操作のフローは次のとおりです。

1. キュー・マネージャー QMG1 に接続された給与計算要求アプリケーションを使用して照会が入力されます。
2. 給与計算要求アプリケーションは、リモート・キュー PAYROLL.QUERY に照会を書き込みます。キュー PAYROLL.QUERY が伝送キュー QMG2 に解決されると、照会は伝送キュー QMG2 に書き込まれます。
3. キュー・マネージャー QMG1 上の送信側チャネル (S) は、キュー・マネージャー QMG2 上のパートナー受信側チャネル (R) に照会を送達します。
4. キュー・マネージャー QMG2 上の受信側チャネル (R) は、キュー・マネージャー QMG3 上のキュー PAYROLL に照会を書き込みます。QMG3 上のキュー PAYROLL が伝送キュー QMG3 に解決されると、照会は伝送キュー QMG3 に書き込まれます。
5. キュー・マネージャー QMG2 上の送信側チャネル (S) は、キュー・マネージャー QMG3 上のパートナー受信側チャネル (R) に照会を送達します。
6. キュー・マネージャー QMG3 上の受信側チャネル (R) は、ローカル・キュー PAYROLL に照会を書き込みます。
7. キュー・マネージャー QMG3 に接続されている給与計算サーバー・アプリケーションは、ローカル・キュー PAYROLL から照会を取り出して処理してから、該当する応答を生成します。

z/OS 構成 1 の定義

構成 1 に必要な定義は次のとおりです (この定義では、トリガーは考慮に入れられておらず、TCP/IP を使用する通信のチャネル定義のみが示されていることに注意してください)。

QMG1 に関して

リモート・キュー定義:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QMG3') REPLACE +  
PUT(ENABLED) RNAME(PAYROLL) RQMNAME(QMG3) XMITQ(QMG2)
```

伝送キュー定義:

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

送信側チャンネル定義 (TCP/IP の場合):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +  
DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

ここで、MVSQMG2(1415) を使用するキュー・マネージャー接続名とポートに置き換えます。

受信側チャンネル定義 (TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG2')
```

応答先キュー定義:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Reply queue for replies to payroll queries sent to QMG3')
```

QMG2 に関して

伝送キュー定義:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)  
  
DEFINE QLOCAL(QMG3) DESCR('Transmission queue to QMG3') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

送信側チャンネル定義 (TCP/IP の場合):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +  
DESCR('Sender channel to QMG1') XMITQ(QMG1) CONNAME('WINTQMG1(1414)')
```

ここで、WINTQMG1(1414) を使用するキュー・マネージャー接続名とポートに置き換えます。

```
DEFINE CHANNEL(QMG2.TO.QMG3) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +  
DESCR('Sender channel to QMG3') XMITQ(QMG3) CONNAME('MVSQMG3(1416)')
```

ここで、MVSQMG3(1416) を使用するキュー・マネージャー接続名とポートに置き換えます。

受信側チャンネル定義 (TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG1')  
  
DEFINE CHANNEL(QMG3.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG3')
```

QMG3 に関して

ローカル・キュー定義:

```
DEFINE QLOCAL(PAYROLL) DESCR('Payroll query request queue') REPLACE +  
PUT(ENABLED) USAGE(NORMAL) GET(ENABLED) SHARE
```

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

送信側チャンネル定義 (TCP/IP の場合):

```
DEFINE CHANNEL(QMG3.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG2) XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

ここで、MVSQMG2(1415) を使用するキュー・マネージャー接続名とポートに置き換えます。

受信側チャンネル定義 (TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG3) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG2)
```

z/OS 構成 2

構成 2 では、キュー共有グループとグループ内キューイングを使用して、バックエンドの給与計算サーバー・アプリケーションに影響を与えずに、キュー・マネージャー QMG1 と QMG3 でメッセージをやりとりする方法を説明します。

構成 2 は、キュー共有グループとグループ内キューイングを使用して、給与計算要求アプリケーションから給与計算サーバーにメッセージを転送する、分散キューイング・システムを示します。この構成では、キュー・マネージャー QMG2 と QMG3 を結ぶチャンネルの定義は必要ありません。グループ内キューイングを使用して、この 2 つのキュー・マネージャー間でメッセージをやりとりするからです。

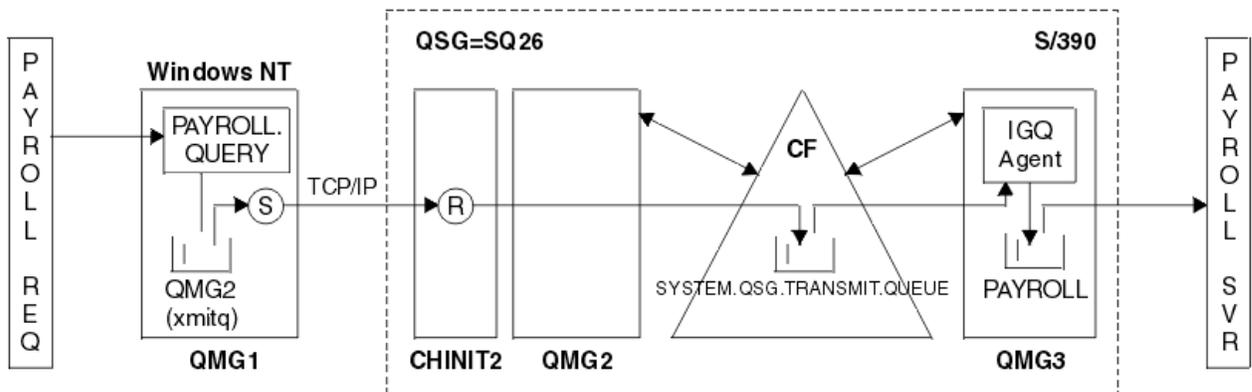


図 3. 構成 2

操作のフローは次のとおりです。

1. キュー・マネージャー QMG1 に接続された給与計算要求アプリケーションを使用して照会が入力されます。
2. 給与計算要求アプリケーションは、リモート・キュー PAYROLL.QUERY に照会を書き込みます。キュー PAYROLL.QUERY が伝送キュー QMG2 に解決されると、照会は伝送キュー QMG2 に書き込まれます。
3. キュー・マネージャー QMG1 上の送信側チャンネル (S) は、キュー・マネージャー QMG2 上のパートナー受信側チャンネル (R) に照会を送達します。
4. キュー・マネージャー QMG2 上の受信側チャンネル (R) は、キュー・マネージャー QMG3 上のキュー PAYROLL に照会を書き込みます。QMG3 上のキュー PAYROLL が共有伝送キュー SYSTEM.QSG.TRANSMIT.QUEUE に解決されると、照会は共有伝送キュー SYSTEM.QSG.TRANSMIT.QUEUE に書き込まれます。
5. キュー・マネージャー QMG3 の IGQ エージェントは、共有伝送キュー SYSTEM.QSG.TRANSMIT.QUEUE から照会を取り出し、キュー・マネージャー QMG3 上のローカル・キュー PAYROLL に書き込みます。
6. キュー・マネージャー QMG3 に接続されている給与計算サーバー・アプリケーションは、ローカル・キュー PAYROLL から照会を取り出して処理してから、該当する応答を生成します。

注: 給与計算照会の例では、小規模メッセージだけが転送されます。持続メッセージと非持続メッセージの両方を転送する必要がある場合は、構成 1 と構成 2 の組み合わせを設定できます。こうすることにより、大規模なメッセージは分散キューイング経路を使用して転送することが可能になり、小規模なメッセージはより高速化できる可能性があるグループ内キューイング経路を使用して転送することができます。

z/OS 構成 2 の定義

構成 2 に必要な定義は次のとおりです (この定義では、トリガーは考慮に入れられておらず、TCP/IP を使用する通信のチャンネル定義のみが示されていることに注意してください)。

キュー・マネージャー QMG2 と QMG3 は、同じキュー共有グループのメンバーとして既に構成済みであることが前提になっています。

QMG1 に関して

リモート・キュー定義:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QMG3') REPLACE +
PUT(ENABLED) RNAME(PAYROLL) RQMNAME(QMG3) XMITQ(QMG2)
```

伝送キュー定義:

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

送信側チャンネル定義 (TCP/IP の場合):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

ここで、MVSQMG2(1415) を使用するキュー・マネージャー接続名とポートに置き換えます。

受信側チャンネル定義 (TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG2')
```

応答先キュー定義:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to payroll queries sent to QMG3')
```

QMG2 に関して

伝送キュー定義:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)

DEFINE QLOCAL(SYSTEM.QSG.TRANSMIT.QUEUE) QSGDISP(SHARED) +
DESCR('IGQ Transmission queue') REPLACE PUT(ENABLED) USAGE(XMITQ) +
GET(ENABLED) INDXTYPE(CORRELID) CFSTRUCT('APPLICATION1') +
DEFSOPT(SHARED) DEFPSIST(NO)
```

ここで、APPLICATION1 をご使用の定義済み CF 構造名に置き換えます。このキューは、共有キューですが、キュー共有グループ内のキュー・マネージャーのうちの 1 つでのみ定義すればよいことにも注意してください。

送信側チャンネル定義 (TCP/IP の場合):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG1') XMITQ(QMG1) CONNAME('WINTQMG1(1414)')
```

ここで、WINTQMG1(1414) を使用するキュー・マネージャー接続名とポートに置き換えます。

受信側チャンネル定義 (TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG1')
```

キュー・マネージャー定義:

```
ALTER QMGR IGQ(ENABLED)
```

QMG3 に関して

ローカル・キュー定義:

```
DEFINE QLOCAL(PAYROLL) DESCR('Payroll query request queue') REPLACE +
PUT(ENABLED) USAGE(NORMAL) GET(ENABLED) SHARE
```

キュー・マネージャー定義:

```
ALTER QMGR IGQ(ENABLED)
```

z/OS 構成 3

構成 3 では、キュー共有グループと共有キューを使用して、バックエンドの給与計算サーバー・アプリケーションに影響を与えずに、キュー・マネージャー QMG1 と QMG3 でメッセージをやりとりする方法を説明します。

構成 3 では、キュー共有グループと共有キューを使用して、キュー・マネージャー QMG1 とキュー・マネージャー QMG3 の間でメッセージをやりとりする、分散キューイング・システムを示します。

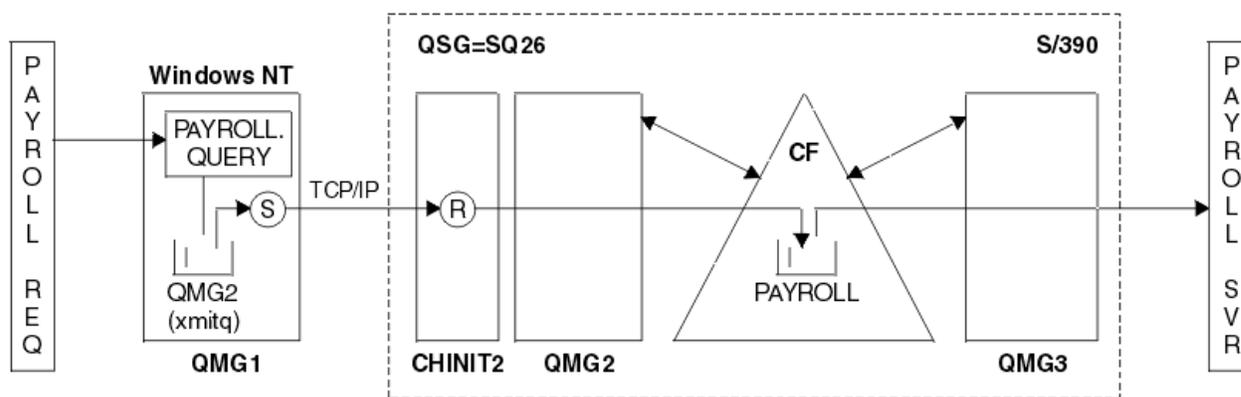


図 4. 構成 3

操作のフローは、次のとおりです。

1. キュー・マネージャー QMG1 に接続された給与計算要求アプリケーションを使用して照会が入力されます。
2. 給与計算要求アプリケーションは、リモート・キュー PAYROLL.QUERY に照会を書き込みます。キュー PAYROLL.QUERY が伝送キュー QMG2 に解決されると、照会は伝送キュー QMG2 に書き込まれます。

3. キュー・マネージャー QMG1 上の送信側チャンネル (S) は、キュー・マネージャー QMG2 上のパートナー受信側チャンネル (R) に照会を送達します。
4. キュー・マネージャー QMG2 上の受信側チャンネル (R) は、共有キュー PAYROLL に照会を書き込みます。
5. キュー・マネージャー QMG3 に接続されている給与計算サーバー・アプリケーションは、共有キュー PAYROLL から照会を取り出して処理してから、該当する応答を生成します。

この構成が最も単純な構成です。ただし、この場合には分散キューイングまたはグループ内キューイングを、(キュー・マネージャー QMG3 に接続された給与計算サーバー・アプリケーションによって生成された) 応答をキュー・マネージャー QMG3 からキュー・マネージャー QMG2 に転送してから、キュー・マネージャー QMG1 に転送するように構成しなければなりません。(給与計算要求アプリケーションに応答を返送するのに使用される構成の詳細は、190 ページの『z/OS のキュー共有グループ例が示す内容』を参照してください。)

QMG3 に関しては定義は不要です。

構成 3 の定義

構成 3 に必要な定義は次のとおりです (この定義では、トリガーは考慮に入れられておらず、TCP/IP を使用する通信のチャンネル定義のみが示されていることに注意してください)。

キュー・マネージャー QMG2 と QMG3 は、同じキュー共有グループのメンバーとして既に構成済みであることが前提になっています。

QMG1 に関して

リモート・キュー定義:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QMG3') REPLACE +
PUT(ENABLED) RNAME(PAYROLL) RQMNAME(QMG3) XMITQ(QMG2)
```

伝送キュー定義:

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

送信側チャンネル定義 (TCP/IP の場合):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

ここで、MVSQMG2(1415) を使用するキュー・マネージャー接続名とポートに置き換えます。

受信側チャンネル定義 (TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG2')
```

応答先キュー定義:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to payroll queries sent to QMG3')
```

QMG2 に関して

伝送キュー定義:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

送信側チャンネル定義 (TCP/IP の場合):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(SDR) TRPTYPE(TCP) +  
REPLACE DESCR('Sender channel to QMG1') XMITQ(QMG1) CONNAME('WINTQMG1(1414)')
```

ここで、WINTQMG1(1414) を使用するキュー・マネージャー接続名とポートに置き換えます。

受信側チャンネル定義 (TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG1')
```

ローカル・キュー定義:

```
DEFINE QLOCAL(PAYROLL) QSGDISP(SHARED) DESCR('Payroll query request queue') +  
REPLACE PUT(ENABLED) USAGE(NORMAL) GET(ENABLED) SHARE +  
DEFSOPT(SHARED) DEFPSIST(NO) CFSTRUCT(APPLICATION1)
```

ここで、APPLICATION1 をご使用の定義済み CF 構造名に置き換えます。このキューは、共有キューですが、キュー共有グループ内のキュー・マネージャーのうちの 1 つでのみ定義すればよいことにも注意してください。

QMG3 に関して

QMG3 に関しては定義は不要です。

例の実行

例をセットアップした後は、例を実行できます。

構成 1 の場合:

1. キュー・マネージャー QMG1、QMG2、および QMG3 を開始します。
2. QMG2 および QMG3 のチャンネル・イニシエーターを開始します。
3. QMG1、QMG2、および QMG3 で、それぞれポート 1414、ポート 1415、ポート 1416 で listen するリスナーを開始します。
4. QMG1、QMG2、および QMG3 で送信側チャンネルを開始します。
5. QMG1 に接続されている給与計算照会要求側アプリケーションを開始します。
6. QMG3 に接続されている給与計算サーバー・アプリケーションを開始します。
7. QMG3 に給与計算照会要求を実行依頼し、給与計算応答を待機します。

構成 2 の場合:

1. キュー・マネージャー QMG1、QMG2、および QMG3 を開始します。
2. QMG2 でチャンネル・イニシエーターを開始します。
3. QMG1 で、ポート 1414 で listen するリスナーを開始し、QMG2 で、ポート 1415 を listen するリスナーを開始します。
4. QMG1 と QMG2 で送信側チャンネルを開始します。
5. QMG1 に接続されている給与計算照会要求側アプリケーションを開始します。
6. QMG3 に接続されている給与計算サーバー・アプリケーションを開始します。

7. QMG3 に給与計算照会要求を実行依頼し、給与計算応答を待機します。

構成 3 の場合:

1. キュー・マネージャー QMG1、QMG2、および QMG3 を開始します。
2. QMG2 でチャンネル・イニシエーターを開始します。
3. QMG1 で、ポート 1414 で listen するリスナーを開始し、QMG2 で、ポート 1415 を listen するリスナーを開始します。
4. QMG1 と QMG2 で送信側チャンネルを開始します。
5. QMG1 に接続されている給与計算照会要求側アプリケーションを開始します。
6. QMG3 に接続されている給与計算サーバー・アプリケーションを開始します。
7. QMG3 に給与計算照会要求を実行依頼し、給与計算応答を待機します。

z/OS 例の拡張

例はさまざまな方法で拡張できます。

以下に例を示します。

- チャンネル・トリガーおよびアプリケーション (PAYROLL および PAYROLL.REPLY キュー) トリガーを使用するための拡張。
- LU 6.2 を使用する通信用の構成。
- キュー共有グループに対するキュー・マネージャー数を増やした構成のための拡張。この場合、サーバー・アプリケーションを複製して、さらに別のキュー・マネージャー・インスタンスを実行し、PAYROLL 照会キュー用に複数のサーバーを用意することができます。
- 給与計算照会要求側アプリケーションのインスタンス数を増やして、複数のクライアントからの要求の処理法を示すための拡張。
- セキュリティー (IGQAUT および IGQUSER) を使用するための拡張。

Linux IBM i UNIX /var/mqm に適用される IBM MQ ファイル・システム権限

以下の情報は、/var/mqm/ の下にあるファイルとディレクトリーに適用されるセキュリティーについて、およびファイル・システム権限がそのように設定される理由について説明します。IBM MQ の適切な運用を確保するには、IBM MQ によって設定されるファイル・システム権限を変更しないでください。

crtmqdir コマンド

IBM MQ 9.0.3 から、何かの理由で /var/mqm のファイル権限を社内に変更した場合は、**crtmqdir** コマンドを使用してその権限を更新したりディレクトリーを追加したりできます。

UNIX、Linux、および IBM i での IBM MQ ファイル・システム・セキュリティー

IBM MQ データ・ディレクトリー (/var/mqm) の下にあるファイルは、以下のデータの保管に使用されます。

- IBM MQ 構成データ
- アプリケーション・データ (IBM MQ オブジェクトおよび IBM MQ メッセージ内に含まれるデータ)
- ランタイム制御情報
- モニタリング情報 (メッセージと FFST ファイル)

このデータへのアクセスは、ファイル・システム許可を使用して制御されます。一部のデータはすべてのユーザーがアクセスできますが、その他のデータは IBM MQ 管理者グループ 'mqm' (または IBM i 上の QMQM) のメンバーのみに制限されます。

アクセス権限は以下の 3 つのカテゴリーで付与されます。

mqm グループのみ

このカテゴリのファイルとディレクトリーにアクセスできるのは、IBM MQ 管理者 (「mqm」グループのメンバー) と IBM MQ キュー・マネージャー・プロセスのみです。

これらのファイルとディレクトリーに対するファイル権限は次のとおりです。

```
-rwxrwx--- mqm:mqm (UNIX and Linux)
-rwxrwx--- QMQMADM:QMQM (IBM i)
```

このカテゴリのファイルおよびディレクトリーには、例えば以下のものがあります。

```
/var/mqm/qmgrs/QMGR/qm.ini
/var/mqm/qmgrs/QMGR/channel/
/var/mqm/qmgrs/QMGR/channel/SYSTEM!DEF!SCRVONN
/var/mqm/qmgrs/QMGR/queues/
/var/mqm/qmgrs/QMGR/queues/SYSTEM!DEFAULT!LOCAL!QUEUES/
/var/mqm/qmgrs/QMGR/errors/
/var/mqm/qmgrs/QMGR/errors/AMQERR01.LOG
/var/mqm/qmgrs/QMGR/ssl/
/var/mqm/qmgrs/QMGR/@qmgr/
/var/mqm/qmgrs/QMGR/@qmpersist/
...
```

全ユーザーに読み取り権限、mqm グループ・メンバーに読み取り/書き込み権限

このカテゴリのファイルとディレクトリーをすべてのユーザーが読み取ることができますが、'mqm' グループのメンバーだけがこれらのファイルを変更したり、これらのディレクトリーを操作したりすることができます。

これらのファイルとディレクトリーに対するファイル権限は次のとおりです。

```
-rwxrwxr-x mqm:mqm (UNIX and Linux)
-rwxrwxr-x QMQMADM:QMQM (IBM i)
```

このカテゴリのファイルおよびディレクトリーには、例えば以下のものがあります。

```
/var/mqm/mqs.ini
/var/mqm/exits/
/var/mqm/qmgrs/
/var/mqm/qmgrs/QMGR/
/var/mqm/qmgrs/QMGR/@app/
/var/mqm/qmgrs/QMGR/@ipcc/
```



重要: 実行可能ファイルとスクリプトにのみ実行権限を設定する必要があります。例えば、Linux では、**crtmqm** コマンドの実行時に以下のファイル権限が設定されます。

```
-rw-rw---- mqm mqm /var/mqm/qmgrs/QMGR/qm.ini
-rw-rw---- mqm mqm /var/mqm/qmgrs/QMGR/channel/SYSTEM!DEF!SCRVONN
-rw-rw---- mqm mqm /var/mqm/qmgrs/QMGR/errors/AMQERR01.LOG
-rw-rw-r-- mqm mqm /var/mqm/mqs.ini
```

IBM MQ 8.0:

```
/var/mqm/sockets/@SYSTEM
/var/mqm/sockets/QMGR/@app/hostname
/var/mqm/sockets/QMGR/@ipcc/hostname
```

全ユーザーに読み取り/書き込み権限

すべてのユーザーに読み取り/書き込み権限が与えられるファイル

IBM MQ には、全ユーザー書き込み可能ファイル許可 (777) を持つ標準ファイルがありません。ただし、全ユーザー書き込み可能ファイル許可を持つように見える特殊ファイルがいくつかあります。

これらの特殊ファイルはセキュリティ上の危険にさらされているわけではありません。権限は 777 と表示されますが、これらは通常のファイルではなく、これらに直接書き込むことはできません。

このような特殊ファイルには、以下のものがあります。

シンボリック・リンク

シンボリック・リンクは、その権限が 'l' 文字で始まることで識別されます。シンボリック・リンクに対する権限は、どのユーザーがターゲット・ファイルにアクセスできるかに影響を与えません。これは、シンボリック・リンクのターゲットに対する権限によって、コマンドへのアクセスが制御されるためです。

ほとんどの UNIX システムおよび Linux システムでは、シンボリック・リンクに対する権限を変更することはできないため、常に `lrwxrwxrwx` と表示されます。

ソケット・ファイル

ソケット・ファイルは、UNIX ドメイン・ソケットの作成プロセスの結果としてオペレーティング・システムによって作成される特殊ファイルです。これらのファイルは、ファイル権限が 's' で始まることで識別されます (つまり `srwxrwxrwx`)。

ファイルに対する権限は、ファイル自体へのアクセス権限を与えるのではなく、どのユーザーが UNIX ドメイン・ソケットに接続できるユーザーを定義するものです。

IBM MQ ではこのようなソケット・ファイルを多数使用し、権限は常に、どのユーザーがソケットとの通信を許可されるかに応じて設定されます。

以下のディレクトリーには、全ユーザーの読み取り/書き込み権限 (`srwxrwxrwx`) が設定されたソケット・ファイルが含まれています。

IBM MQ 8.0:

```
/var/mqm/sockets/QMGR/zsocketEC/hostname/Zsocket_*
```

分離バインディングを使って IBM MQ に接続するアプリケーションで使用されるソケット・ファイル。

```
/var/mqm/sockets/QMGR/@ipcc/ssem/hostname/*
```

すべてのユーザーに読み取り/書き込み権限が与えられるディレクトリー

IBM MQ アプリケーションが IBM MQ データ・ディレクトリーの下にファイルを作成する必要があることがあります。必要な場合にアプリケーションがファイルを作成できるようにするために、いくつかのディレクトリーには全員書き込みアクセス権限が付与されています。つまり、システム上の任意のユーザーがそのディレクトリー内にファイルを作成できます。

'mqm' グループの任意のメンバーによる書き込みが可能なエラー・ログ・ファイルを除き、これらのディレクトリー内に作成されるすべてのファイルには制限付き権限が設定され、ファイル作成者だけが書き込み権限を持ちます。これにより、システム管理者はこれらのディレクトリー内のファイルに書き込まれるすべてのデータのユーザー ID を追跡できます。

`/var/mqm/errors/`

このディレクトリーにはシステム・エラー・ログ・ファイルと FFST ファイルが含まれます。このディレクトリーの権限は `'drwxrwsrwt'` です。つまり、システム上のすべてのユーザーがこのディレクトリー内でファイルを作成できます。

SetGroupId ビット 's' は、このディレクトリー内で作成されるすべてのファイルに 'mqm' グループ所有権が設定されることを示します。

このディレクトリーでは 't' スティッキー・ビットはデフォルトでは設定されませんが、IBM MQ 管理者はこれを明示的に設定して、作成したファイルのみをユーザーが削除できるようにすることができます。

注:  この機能は IBM i では使用できません。

AMQERR0*.LOG

group のメンバーだけがこれらのエラー・ログ・ファイルに直接書き込むことができますが、どのユーザーもこれらのファイルに書き込まれたメッセージを読み取れます (権限は -rw-rw-r--)。

AMQnnnnn*.FDC

これらのファイルには、キュー・マネージャー、またはユーザー作成のアプリケーションでエラーが発生したときに書き込まれる FFST 情報が含まれます。これらのファイルは権限 -rw-r----- 付きで作成されます。

`/var/mqm/trace/`

IBM MQ トレースが有効になっている場合、トレース・ファイルがこのディレクトリーに書き込まれます。IBM MQ トレースは、トレースが有効になっているキュー・マネージャーに関連付けられたすべてのプロセスによって書き込まれます。

このディレクトリーの権限は 'drwxrwsrwt' です。つまり、システムのすべてのユーザーがこのディレクトリー内でファイルを作成できます。

SetGroupId ビット 's' は、このディレクトリーに作成されたすべてのファイルのグループ所有権が 'mqm' であることを示します。

このディレクトリーでは 't' スティッキー・ビットはデフォルトでは設定されませんが、IBM MQ 管理者はこれを明示的に設定して、作成したファイルのみをユーザーが削除できるようにすることができます。

注:  この機能は IBM i では使用できません。

AMQnnnnn*.TRC

これらのファイルは、トレースを行う各プロセスによって書き込まれるトレース・データを格納し、-rw-r----- という権限付きで作成されます。

このディレクトリーに対する権限は drwxrwsrwt で、このディレクトリー内に作成されるソケット・ファイルの権限は srwx----- です。

IBM MQ 8.0:

```
/var/mqm/sockets/QMGR/zsocketapp/hostname/
```

このディレクトリーは、分離 バインディングを使って IBM MQ キュー・マネージャーに接続するアプリケーションで使用されます。接続処理中に、接続元のアプリケーションによってこのディレクトリー内にソケット・ファイルが作成されます。キュー・マネージャーへの接続が行われた後、ソケット・ファイルは削除されます。

このディレクトリーに対する権限は drwxrwsrwt で、このディレクトリー内に作成されるソケット・ファイルの権限は srwx----- です。

このディレクトリーの SetGroupId ビット 's' は、このディレクトリー内で作成されるすべてのファイルに 'mqm' グループ所有権が設定されることを示します。

IBM i を除くすべてのプラットフォームで、このディレクトリーには 't' スティッキー・ビットも設定されるため、ユーザーは自分が所有者ではないファイルを削除できません。これにより、無許可ユーザーが自分の所有していないファイルを削除することを防止できます。

```
/var/mqm/sockets/QMGR/@ipcc/ssem/hostname/  
/var/mqm/sockets/QMGR/@app/ssem/hostname/
```

 共有 バインディングを使って IBM MQ に接続するプロセスの場合、アプリケーションとキュー・マネージャーの間で同期するために UNIX ドメイン・ソケットを使用することもできます。UNIX ドメイン・ソケットが使用される場合、それに関連するソケット・ファイルがこれらのディレクトリー内に作成されます。

これらのディレクトリーに対する権限は drwxrwsrwt で、これらのディレクトリー内に作成されるソケット・ファイルの権限は srwxrwxrwx です。

これらのディレクトリーに関する `SetGroupId` ビット 's' は、これらのディレクトリー内に作成されるすべてのファイルに 'mqm' グループ所有権が必ず設定されるようにします。

IBM i を除くすべてのプラットフォームで、これらのディレクトリーには 's' スティッキー・ビットも設定されるため、ユーザーは自分が所有者ではないファイルを削除できません。これにより、無許可ユーザーが自分の所有していないファイルを削除することを防止できます。

IBM MQ による System V IPC リソースの使用

IBM MQ は System V 共有メモリーおよびセマフォをプロセス間通信 (IPC) のために使用します。これらのリソースは、適切な所有権とアクセス権限を持つそれぞれのグループによってどのように使用されるかに応じてグループ化されます。

システム上のどの System V IPC リソースが IBM MQ に属するかを確認するには、次の方法に従うことができます。

- 所有権を確認します。

UNIX プラットフォームおよび Linux では、IBM MQ System V IPC リソースを所有するユーザーは常に「mqm」ユーザーです。IBM i では所有者ユーザーは 'QMQM' です。

- IBM MQ 8.0 以降では `amqspdbg` ユーティリティーを使用します。

IBM MQ に付属の `amqspdbg` ユーティリティーを使用して、特定のキュー・マネージャーの共有メモリーおよびセマフォ ID を表示できます。

IBM MQ によって作成される System V リソースの 'system' グループに関してコマンドを一度発行する必要があります。

```
# amqspdbg -z -I
```

さらに、システム上の各キュー・マネージャーに関して 4 度発行することにより、IBM MQ で使用される System V リソースの完全なリストを得ることができます。以下の例では、キュー・マネージャー名 QMGR1 を想定します。

```
# amqspdbg -i QMGR1 -I
# amqspdbg -q QMGR1 -I
# amqspdbg -p QMGR1 -I
# amqspdbg -a QMGR1 -I
```

IBM MQ によって作成される System V リソースに対するアクセス権限は、許可されたユーザーに適切なアクセス・レベルのみを付与するよう設定されます。IBM MQ によって作成される多くの System V IPC リソースは、マシン上のすべてのユーザーからアクセス可能であり、`-rw-rw-rw-` という権限が設定されます。

`crtmqm` コマンドで `-g ApplicationGroup` パラメーターを使用すると、キュー・マネージャーへのアクセス権限を、特定のオペレーティング・システム・グループのメンバーのみに限定できます。このグループ限定機能を使用することで、System V IPC リソースに関して付与される権限をさらに制限できます。

Linux

UNIX

mqm の setuid が設定された /opt/mqm の IBM MQ のファイル権限

ファイル権限

ここでは、社内のセキュリティー・チームが、ディレクトリー・ツリー `$MQ_INSTALLATION_PATH` 内の一部の実行可能 IBM MQ ファイルがローカル・セキュリティー・ポリシーに違反していることを問題視しているという状況に対応する説明を行います。AIX ではデフォルトの場所は `/usr/mqm` ですが、その他の UNIX オペレーティング・システムでは `/opt/mqm` です。IBM MQ をデフォルト以外のディレクトリー (`/opt/mqm90` など) にインストールした場合や、複数のインストール済み環境がある場合にも、このトピックの詳細情報が当てはまります。

問題の原因

セキュリティー・チームが `$MQ_INSTALLATION_PATH` 下の次の領域の問題点を指摘しました。

1. /opt/mqm/bin ディレクトリー内のファイルは、それらが存在するディレクトリー・ツリーの所有者の `setuid` です。以下に例を示します。

```
dr-xr-xr-x   mqm mqm   ${MQ_INSTALLATION_PATH}/bin
-r-sr-s---   mqm mqm   ${MQ_INSTALLATION_PATH}/bin/addmqinf
-r-sr-s---   mqm mqm   ${MQ_INSTALLATION_PATH}/bin/amqcrsta
-r-sr-s---   mqm mqm   ${MQ_INSTALLATION_PATH}/bin/amqfcxba
...
```

2. ほとんどすべてのディレクトリーとファイルを "mqm:mqm" が所有していますが、例外として、次のものは `root` が所有しています。

```
dr-xr-x---   root mqm   ${MQ_INSTALLATION_PATH}/bin/security
-r-sr-x---   root mqm   ${MQ_INSTALLATION_PATH}/bin/security/amqoamax
-r-sr-x---   root mqm   ${MQ_INSTALLATION_PATH}/bin/security/amqoampx
```

このサブディレクトリーは、`root` が所有する必要があります。IBM MQ クライアントのユーザーがパスワードを指定した場合に、そのパスワードが有効かどうかを確認するために、IBM MQ キュー・マネージャーはパスワードをオペレーティング・システムに渡しますが、そのときにオペレーティング・システムと対話する実行可能ファイルがこのサブディレクトリーに存在するからです。

3. ユーザーは /opt/mqm/lib/iconv ディレクトリー内のファイルを所有していません (このディレクトリーは AIX には存在しません)。以下に例を示します。

```
dr-xr-xr-x   mqm mqm   ${MQ_INSTALLATION_PATH}/lib/iconv
-r--r--r--   bin bin   ${MQ_INSTALLATION_PATH}/lib/iconv/002501B5.tbl
-r--r--r--   bin bin   ${MQ_INSTALLATION_PATH}/lib/iconv/002501F4.tbl
-r--r--r--   bin bin   ${MQ_INSTALLATION_PATH}/lib/iconv/00250333.tbl
...
```

4. RPM ベース Linux システムのフィックスパック保守ディレクトリー。フィックスパックがインストールされると、既存のファイルは、以下の例に示すような構造でこのディレクトリーの下に保存されます。ただし、この例では、V.R は IBM MQ のバージョンとリリース番号を表しており、表示されるサブディレクトリーは、インストールされるフィックスパックによって異なります。

```
drwx-----   root root   ${MQ_INSTALLATION_PATH}/maintenance
drwxr-xr-x   root root   ${MQ_INSTALLATION_PATH}/maintenance/V.R.0.1
drwxr-xr-x   root root   ${MQ_INSTALLATION_PATH}/maintenance/V.R.0.3
drwxr-xr-x   root root   ${MQ_INSTALLATION_PATH}/maintenance/V.R.0.4
...
```

問題の解決方法

`setuid` プログラムに関する UNIX システム上の懸念事項の 1 つは、`LD*` (AIX では `LD_LIBRARY_PATH`、`LIBPATH` など) などの環境変数を操作することによって、システム・セキュリティーが損なわれる可能性があるということでした。しかし、この点は問題にならなくなりました。各種 UNIX オペレーティング・システムが、`setuid` プログラムのロード時にこのような `LD*` 環境変数を無視するようになったからです。

1. IBM MQ プログラムの一部が `mqm-setuid` や `mqm-setgid` であるのはなぜですか。

IBM MQ では、ユーザー ID 「mqm」と、「mqm」グループの一部であるすべての ID が、IBM MQ 管理ユーザーです。

IBM MQ のキュー・マネージャー・リソースは、このユーザーの認証を行うことで保護されています。キュー・マネージャー・プロセスは、そうしたキュー・マネージャー・リソースを使用して変更するので、リソースにアクセスするために "mqm" 権限を必要とします。したがって、IBM MQ キュー・マネージャー・サポート・プロセスは、有効なユーザー ID 「mqm」で実行するように設計されています。

IBM MQ オブジェクトにアクセスする非管理ユーザーのために、IBM MQ にはオブジェクト権限マネージャー (OAM) 機能が用意されています。この機能によって、非管理ユーザーで実行するアプリケーションの要件に合わせて権限の付与/取り消しを実行できます。

ユーザーにさまざまなレベルの認証を付与する機能と、`setuid` プログラムと `setgid` プログラムが `LD*` 変数を無視するという事実により、IBM MQ バイナリー・ファイルとライブラリー・ファイルは、どのような方法でもシステムのセキュリティーを危険にさらすことはありません。

2. IBM MQ の機能を低下させずに、社内のセキュリティー・ポリシーに合わせて権限を変更することはできません。

IBM MQ バイナリーおよびライブラリーのアクセス権と所有権を変更してはなりません。そのような変更を行うと、IBM MQ の機能に問題が発生するおそれがあります。例えば、キュー・マネージャー・プロセスが一部のリソースにアクセスできなくなる可能性があります。

権限および所有権がシステムに対するセキュリティー上の脅威を引き起こすことはありません。

IBM MQ がインストールされている、または IBM MQ のデータが配置されている Linux ハード・ドライブ/ディスクは、nosuid オプションを使用してマウントしないでください。この構成では IBM MQ の機能が阻害される可能性があります。

詳しくは、78 ページの『[/var/mqm に適用される IBM MQ ファイル・システム権限](#)』を参照してください。

関連情報

[ファイル・システム](#)

Windows Windows での IBM MQ のファイル・システム権限

ここでは、Windows でファイルやディレクトリーに適用されるセキュリティーについて説明します。IBM MQ の適切な運用を確保するには、IBM MQ によって設定されるファイル・システム権限を変更しないでください。

データ・ディレクトリー

注：このディレクトリーのルートに設定した権限が、ディレクトリー構造全体にわたって下位方向に継承されます。

データ・ディレクトリー (DATADIR) の下のディレクトリーには、一部の例外を除き、以下の権限が設定されます。例外については、この後の文章で詳しく説明します。

管理者

フル・コントロール

mqm グループ

フル・コントロール

SYSTEM

フル・コントロール

全員

読み取りと実行

ただし、次の例外があります。

DATADIR¥errors

全員のフル・コントロール

DATADIR¥trace

全員のフル・コントロール

DATADIR¥log

管理者

フル・コントロール

mqm グループ

フル・コントロール

SYSTEM

フル・コントロール

全員

読み取り

DATADIR\log\<qmgrname>\active

管理者

フル・コントロール

mqm グループ

フル・コントロール

SYSTEM

フル・コントロール

全員に与えられるアクセス権はありません。

以前のリリースの製品

IBM MQ 8.0 より前のリリースの製品では、デフォルトのプログラム・ディレクトリーとデフォルトのデータ・ディレクトリーが同じ場所にありました。

元々 IBM MQ 8.0 より前にインストールされていたすべてのインストール済み環境。デフォルトの場所にインストールされてアップグレードされたデータ・ディレクトリーとプログラム・ディレクトリーは、同じ場所 (C:\Program Files\IBM\WebSphere MQ) に残ります。

データ・ディレクトリーとプログラム・ディレクトリーが同じ場所にある場合、前述の情報は、データ・ディレクトリーに属するディレクトリーにのみ当てはまります。プログラム・ディレクトリーに属するディレクトリーには当てはまりません。

キューの命名上の制約

キュー名の長さには制限があります。いくつかのキュー名は、キュー・マネージャーによって定義されたキュー用に予約済みです。

名前の長さの制限

キューの名前には最大 48 文字を使用できます。

予約キュー名

"SYSTEM." で始まる名前は、キュー・マネージャーによって定義されるキュー用に予約されています。ご使用のインストール・システムに合わせてこのキュー定義を変更したい場合は、**ALTER** コマンドまたは **DEFINE REPLACE** コマンドで変更できます。IBM MQ では、次の名前が定義されています。

キュー名	説明
SYSTEM.ADMIN.ACTIVITY.QUEUE	アクティビティー報告書のためのキュー
SYSTEM.ADMIN.CHANNEL.EVENT	チャンネル・イベントのためのキュー
SYSTEM.ADMIN.COMMAND.EVENT	コマンド・イベントのためのキュー
SYSTEM.ADMIN.COMMAND.QUEUE	PCF コマンド・メッセージの送り先となるキュー
SYSTEM.ADMIN.CONFIG.EVENT	構成イベントのためのキュー
SYSTEM.ADMIN.PERFM.EVENT	パフォーマンス・イベントのためのキュー
SYSTEM.ADMIN.PUBSUB.EVENT	システムのパブリッシュ/サブスクライブに関連したイベント・キュー
SYSTEM.ADMIN.QMGR.EVENT	キュー・マネージャー・イベントのためのキュー
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	トレース経路の応答メッセージのためのキュー
SYSTEM.AUTH.DATA.QUEUE	キュー・マネージャー用のアクセス制御リストを保持するキュー (z/OS を除く)

キュー名	説明
SYSTEM.CHANNEL.INITQ	チャンネルの開始キュー
SYSTEM.CHANNEL.SYNCQ	チャンネルの同期データを保持するキュー
SYSTEM.CHLAUTH.DATA.QUEUE	IBM MQ のチャンネル認証データ・キュー
SYSTEM.CICS.INITIATION.QUEUE	トリガーに使用するキュー (z/OS を除く)
SYSTEM.CLUSTER.COMMAND.QUEUE	キュー・マネージャー間でリポジトリの変更をやり取りするために使用されるキュー
SYSTEM.CLUSTER.HISTORY.QUEUE	キューは、サービス目的でクラスター状態情報の履歴を保管するために使用されます。
SYSTEM.CLUSTER.REPOSITORY.QUEUE	リポジトリについての情報を保持するために使用されるキュー
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	このキューを使用して、各クラスター送信側チャンネルの個々の送信キューを作成します。
SYSTEM.CLUSTER.TRANSMIT.QUEUE	クラスター・サポートで管理されるすべての宛先用の伝送キュー
SYSTEM.COMMAND.INPUT	z/OS でコマンド・メッセージの送り先となるキュー
SYSTEM.COMMAND.REPLY.MODEL	コマンド応答用のモデル・キュー定義 (z/OS 用)
SYSTEM.DEAD.LETTER.QUEUE	送達不能キュー (z/OS を除く)
SYSTEM.DEFAULT.ALIAS.QUEUE	デフォルトの別名キュー定義
SYSTEM.DEFAULT.INITIATION.QUEUE	指定したプロセスを起動するためのキュー (z/OS を除く)
SYSTEM.DEFAULT.LOCAL.QUEUE	デフォルトのローカル・キュー定義
SYSTEM.DEFAULT.MODEL.QUEUE	デフォルトのモデル・キュー定義
SYSTEM.DEFAULT.REMOTE.QUEUE	デフォルトのリモート・キュー定義
SYSTEM.DURABLE.SUBSCRIBER.QUEUE	キュー・マネージャー内の永続サブスクリプションの持続コピーを保持するために使用するローカル・キュー
SYSTEM.HIERARCHY.STATE	パブリッシュ/サブスクライブ階層におけるキュー・マネージャー間の関係についての状態情報を保持するためのキュー
SYSTEM.JMS.TEMPQ.MODEL	JMS 一時キューのモデル
SYSTEM.INTERNAL.REPLY.QUEUE	IBM MQ 内部応答キュー (z/OS を除く)
SYSTEM.INTER.QMGR.CONTROL	パブリッシュ/サブスクライブ階層で、プロキシー・サブスクリプションを作成するためにリモート・キュー・マネージャーから要求を受け取るのに使用するキュー
SYSTEM.INTER.QMGR.PUBS	パブリッシュ/サブスクライブ階層で、リモート・キュー・マネージャーからパブリケーションを受け取るために使用するキュー
SYSTEM.INTER.QMGR.FANREQ	パブリッシュ/サブスクライブ階層で、リモート・キュー・マネージャー上にプロキシー・サブスクリプションを作成するための要求を処理するのに使用するキュー
SYSTEM.MQEXPLORER.REPLY.MODEL	IBM MQ Explorer への応答用のモデル・キュー定義
SYSTEM.MQSC.REPLY.QUEUE	MQSC コマンド応答用のモデル・キュー定義 (z/OS を除く)

キュー名	説明
SYSTEM.QSG.CHANNEL.SYNCQ	共有チャンネルの同期情報を含むメッセージの保管に使用される共有ローカル・キュー (z/OS のみ)
SYSTEM.QSG.TRANSMIT.QUEUE	同じキュー共有グループ内のキュー・マネージャー間でメッセージを伝送する際に、グループ内のキューイング・エージェントで使用される共有ローカル・キュー (z/OS のみ)
SYSTEM.RETAINED.PUB.QUEUE	キュー・マネージャーに保存されたそれぞれのパブリケーションのコピーを保持するために使用するローカル・キュー
SYSTEM.SELECTION.EVALUATION.QUEUE	IBM MQ 内部選択評価キュー (z/OS を除く)
SYSTEM.SELECTION.VALIDATION.QUEUE	IBM MQ 内部選択検証キュー (z/OS を除く)

その他のオブジェクトの命名上の制約

オブジェクト名の長さには制限があります。いくつかのオブジェクト名は、キュー・マネージャーによって定義されたオブジェクト用に予約済みです。

名前の長さの制限

プロセス、名前リスト、クラスター、トピック、サービス、および認証情報オブジェクトには、最大 48 文字の名前を付けることができます。

チャンネルの名前に使用できるのは最大 20 文字です。

ストレージ・クラスには 8 文字までの長さの名前を付けることができます。

CF 構造体には、12 文字までの長さの名前を付けることができます。

予約オブジェクト名

SYSTEM で始まる名前。はキュー・マネージャーによって定義されたオブジェクト用に予約済みです。ご使用のインストール・システムに合わせてこのオブジェクト定義を変更する場合は、**ALTER** コマンドまたは **DEFINE REPLACE** コマンドを使用できます。IBM MQ では、次の名前が定義されています。

オブジェクト名	説明
SYSTEM.ADMIN.SVRCONN	キュー・マネージャーのリモート管理に使用されるサーバー接続チャンネル
SYSTEM.AUTO.RECEIVER	自動定義のためのデフォルトの受信側チャンネル (UNIX, Linux, and Windows システムのみ)
SYSTEM.AUTO.SVRCONN	自動定義のためのデフォルトのサーバー接続チャンネル (Multiplatforms のみ)
SYSTEM.BASE.TOPIC	ASPARENT を解決する基本トピック。特定の管理トピック・オブジェクトに親管理トピック・オブジェクトがない場合、ASPARENT 属性があればそれがこのオブジェクトから継承されます。
SYSTEM.DEF.CLNTCONN	デフォルトのクライアント接続チャンネル定義
SYSTEM.DEF.CLUSRCVR	デフォルトのクラスター受信側チャンネル定義
SYSTEM.DEF.CLUSSDR	デフォルトのクラスター送信側チャンネル定義
SYSTEM.DEF.RECEIVER	デフォルトの受信側チャンネル定義

オブジェクト名	説明
SYSTEM.DEF.REQUESTER	デフォルトの要求側チャンネル定義
SYSTEM.DEF.SENDER	デフォルトの送信側チャンネル定義
SYSTEM.DEF.SERVER	デフォルトのサーバー・チャンネル定義
SYSTEM.DEF.SVRCONN	デフォルトのサーバー接続チャンネル定義
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	タイプ CRLLDAP の認証情報オブジェクトを定義するための、デフォルトの認証情報オブジェクト定義
SYSTEM.DEFAULT.AUTHINFO.OCSP	タイプ OCSP の認証情報オブジェクトを定義するための、デフォルトの認証情報オブジェクト定義
SYSTEM.DEFAULT.LISTENER.LU62	デフォルトの SNA リスナー (Windows のみ)
SYSTEM.DEFAULT.LISTENER.NETBIOS	デフォルトの NetBIOS リスナー (Windows のみ)
SYSTEM.DEFAULT.LISTENER.SPX	デフォルトの SPX リスナー (Windows のみ)
SYSTEM.DEFAULT.LISTENER.TCP	デフォルトの TCP/IP リスナー (Multiplatforms のみ)
SYSTEM.DEFAULT.NAMELIST	デフォルト名前リスト定義
SYSTEM.DEFAULT.PROCESS	デフォルトのプロセス定義
SYSTEM.DEFAULT.SERVICE	デフォルトのサービス (Multiplatforms のみ)
SYSTEM.DEFAULT.TOPIC	デフォルトのトピック定義
SYSTEM.QPUBSUB.QUEUE.NAMELIST	モニターするキューに入れられたパブリッシュ/サブスクライブ・インターフェースに対するキューのリスト
  SYSTEMST	デフォルトのストレージ・クラス定義 (z/OS のみ)

キュー名の解決

このトピックには、チャンネルの送信側と受信側の両方のキュー・マネージャーが行うキュー名解決についての情報が含まれています。

大規模なネットワークでは、キュー・マネージャーを使用することは、他の通信形式に比べていくつかの利点があります。これらの利点は、DQM における名前の解決機能から派生するものであり、その主な点を挙げると、次のようになります。

- アプリケーションが経路判断を行う必要がない
- アプリケーションがネットワーク構造体を認識している必要がない
- ネットワーク・リンクはシステム管理者によって作成される
- ネットワーク構造はネットワーク計画者によって制御される
- トラフィックを分割するためにノード間で複数のチャンネルを使用できる

以下の図は、キュー名の解決の例を示しています。図には、ネットワーク内の 2 つのマシンが示されています。1 つは書き込みアプリケーションを実行し、もう 1 つは読み取りアプリケーションを実行しています。これらのアプリケーションは、MCA によって制御される IBM MQ チャンネルを経由して相互に通信します。アプリケーションに関する限り、そのプロセスは、ローカル・キューにメッセージを入れる場合と同じです。

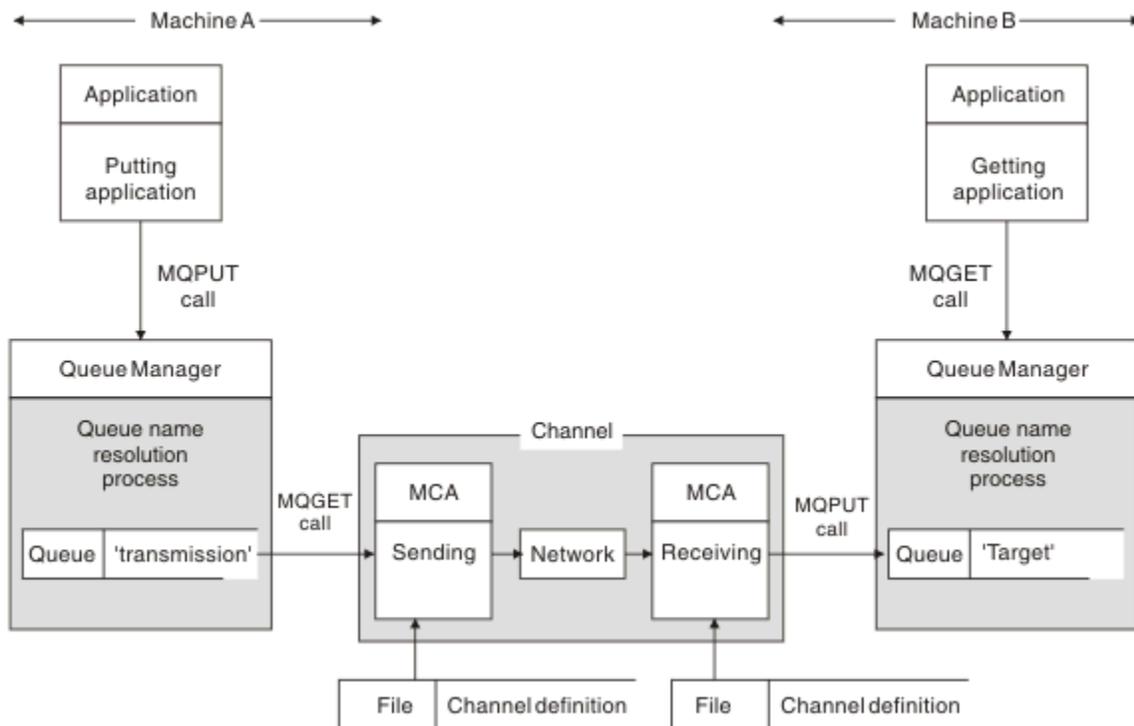


図 5. 名前の解決

89 ページの図 5 に示したように、リモート・キューにメッセージを入れるための基本メカニズムは、アプリケーションに関する限り、ローカル・キューにメッセージを入れる場合と同じです。

- メッセージを書き込むアプリケーションが MQOPEN および MQPUT 呼び出しを発行して、ターゲット・キューにメッセージを入れます。
- メッセージを読み取るアプリケーションが MQOPEN および MQGET 呼び出しを発行して、ターゲット・キューからメッセージを取り出します。

両方のアプリケーションが同じキュー・マネージャーに接続されている場合には、キュー・マネージャー間通信を行う必要はなく、ターゲット・キューが両方のアプリケーションに対してローカルなキューとして記述されます。

ただし、アプリケーションが異なるキュー・マネージャーに接続されている場合は、図に示したように、2 つの MCA とそれらの関連ネットワーク接続が転送のために必要になります。この場合、ターゲット・キューは書き込みを行うアプリケーションにとってリモート・キューであると見なされます。

イベントの順序は次のとおりです。

1. 書き込みを行うアプリケーションが、ターゲット・キューにメッセージを入れるために MQOPEN および MQPUT 呼び出しを発行します。
2. MQOPEN 呼び出し中に、名前の解決機能は、ターゲット・キューがローカルではないことを検出し、適切な伝送キューを決定します。そのあとで、MQOPEN 呼び出しに関連する MQPUT 呼び出しで、すべてのメッセージがこの伝送キューに入れられます。
3. 送信側 MCA が伝送キューからメッセージを入手し、それをリモート側: 計算機の受信側 MCA に渡します。
4. 受信側 MCA がメッセージを 1 つまたは複数のターゲット・キューに書き込みます。
5. 読み取りを行うアプリケーションが、MQOPEN および MQGET 呼び出しを発行してターゲット・キューからメッセージを入手します。

注: アプリケーション・コードに関連するのは、ステップ 1 とステップ 5 だけであり、ステップ 2 から 4 まではローカル・キュー・マネージャーと MCA プログラムによって実行されます。書き込みを行うアプリ

キューションは、ターゲット・キューの位置を認識しません(ターゲット・キューは、同じプロセッサにある場合も、海外の別のプロセッサにある場合もあります)。

送信側 MCA、ネットワーク接続、および受信側 MCA の組み合わせはメッセージ・チャンネルと呼ばれ、単一方向の装置です。通常は、メッセージを両方向に移動させる必要があるため、各方向ごとに1つずつ、合計2つのチャンネルがこの移動のためにセットアップされます。

キュー名解決について

キュー名の解決は DQM にとって不可欠です。キュー名を解決すれば、アプリケーションはキューの物理ロケーションを考慮しないで済み、ネットワークの詳細に関与する必要がなくなります。

システム管理者は、キュー・マネージャーからキュー・マネージャーにキューを移動し、キュー・マネージャーをつなぐ経路を変更できますが、アプリケーションがこのことを認識する必要はありません。

データが通過する正確なパスをアプリケーション設計から切り離すためには、アプリケーションがターゲット・キューを参照するときに使用する名前とフローが行われるチャンネルの名前との間の関係を、ある程度間接的なものにする必要があります。これには、キュー名解決機能を使用します。

実際には、アプリケーションがキュー名を参照すると、解決機能によってこの名前が伝送キューまたはそれ以外のローカル・キューのいずれかにマップされます。伝送キューにマップする場合には、宛先で2回目の名前の解決を行う必要があり、受信されたメッセージはアプリケーション設計者が意図したターゲット・キューに入ります。アプリケーションは、メッセージの移動に使用された伝送キューとチャンネルについては関知しません。

注: キューとチャンネルの定義はシステム管理者によって行われ、オペレーターまたはシステム管理ユーティリティーは、アプリケーションを変更することなくこの定義を変更できます。

メッセージ・フローのシステム管理のための重要な条件として、キュー・マネージャー間で代替パスを用意する必要があります。例えば、業務の必要上から、異なるサービス・クラスを別のチャンネルを介して同一の宛先に送ることがあります。この決定はシステム管理者によって行われ、キュー名解決機構を使用して、柔軟性のある方法で実現できます。「アプリケーション・プログラミング・ガイド」に詳しい説明がありますが、基本となるのは、送信側キュー・マネージャーでキュー名解決を使用して、アプリケーションによって提供されたキュー名を、関連するトラフィックのタイプに適した伝送キューにマップするという考え方です。同様に受信側でも、キュー名解決によって、メッセージ記述子内の名前が(伝送キュー以外の)ローカル・キュー、または再度適切な伝送キューにマップされます。

あるキュー・マネージャーから別のキュー・マネージャーへのトラフィック・パスをさまざまなタイプのトラフィックに分割できるだけでなく、アウトバウンド・メッセージの応答先キュー定義に送られる戻りメッセージにも、同じトラフィック分割を使用できます。キュー名解決がこの条件を満たすため、アプリケーション設計者はこういったトラフィック分割の判断に関与する必要がありません。

送信側と受信側の両方のキュー・マネージャーでマッピングが行われることは、名前の解決における重要な働きの1つです。このマッピングにより、書き込みを行うアプリケーションが提供したキュー名が送信側キュー・マネージャーのローカル・キューまたは伝送キューにマップされ、また受信側キュー・マネージャーのローカル・キューまたは伝送キューにも再びマップされるようになります。

受信側のアプリケーションまたは MCA からの応答メッセージでも、同じ方法で名前の解決が行われ、経路上のすべてのキュー・マネージャーのキュー定義を使用して、特定のパスを介した戻り経路が確立されます。

システムおよびデフォルト・オブジェクト

`crtmqm` により作成されるシステム・オブジェクトおよびデフォルト・オブジェクトをリストします。

`crtmqm` 制御コマンドを使用してキュー・マネージャーを作成する場合、システム・オブジェクトおよびデフォルト・オブジェクトが自動的に作成されます。

- システム・オブジェクトは、キュー・マネージャーまたはチャンネルの操作に必要な IBM MQ オブジェクトです。
- デフォルト・オブジェクトは、オブジェクトのすべての属性を定義します。ローカル・キューなどのオブジェクトを作成する場合、明示的に指定していない属性はすべてデフォルト・オブジェクトから継承されます。

次の表は、**crtmqm** により作成されるシステム・オブジェクトとデフォルト・オブジェクトを示しています。

- [91 ページの表 11](#) は、システム・オブジェクトとデフォルト・キュー・オブジェクトを示しています。
- [92 ページの表 12](#) は、システム・オブジェクトとデフォルト・トピック・オブジェクトを示しています。
- [92 ページの表 13](#) は、システム・オブジェクトとデフォルト・チャンネル・オブジェクトを示しています。
- [93 ページの表 14](#) は、システム・オブジェクトとデフォルトの認証情報オブジェクトを示しています。
- [93 ページの表 15](#) は、システム・オブジェクトとデフォルトのリスナー・オブジェクトを示しています。
- [93 ページの表 16](#) は、システム・オブジェクトとデフォルト名前リスト・オブジェクトを示しています。
- [93 ページの表 17](#) は、システム・オブジェクトとデフォルト・プロセス・オブジェクトを示しています。
- [93 ページの表 18](#) は、システム・オブジェクトとデフォルトのサービス・オブジェクトを示しています。

オブジェクト名	説明
SYSTEM.ADMIN.ACCOUNTING.QUEUE	アカウントリング・モニター・データを保持するキュー。
SYSTEM.ADMIN.ACTIVITY.QUEUE	戻されたアクティビティ報告書を保持するキュー。
SYSTEM.ADMIN.CHANNEL.EVENT	チャンネルのためのイベント・キュー
SYSTEM.ADMIN.COMMAND.EVENT	コマンド・イベントのイベント・キュー
SYSTEM.ADMIN.COMMAND.QUEUE	管理者コマンド・キュー。リモート MQSC コマンドおよび PCF コマンドに使用されます。
SYSTEM.ADMIN.CONFIG.EVENT	構成イベントのイベント・キュー
SYSTEM.ADMIN.PERFM.EVENT	パフォーマンス・イベントのイベント・キュー
SYSTEM.ADMIN.PUBSUB.EVENT	システムのパブリッシュ/サブスクライブに関連したイベント・キュー
SYSTEM.ADMIN.QMGR.EVENT	キュー・マネージャー・イベントのイベント・キュー
SYSTEM.ADMIN.STATISTICS.QUEUE	統計モニター・データを保持するキュー。
SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE	トレース・アクティビティを表示するキュー。
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	戻されたトレース経路応答メッセージを保持するキュー。
SYSTEM.AUTH.DATA.QUEUE	キュー・マネージャー用のアクセス制御リストを保持するキュー
SYSTEM.CHANNEL.INITQ	チャンネル開始キュー
SYSTEM.CHANNEL.SYNCQ	チャンネルの同期データを保持するキュー
SYSTEM.CHLAUTH.DATA.QUEUE	IBM MQ のチャンネル認証データ・キュー
SYSTEM.CICS.INITIATION.QUEUE	デフォルト CICS 開始キュー
SYSTEM.CLUSTER.COMMAND.QUEUE	リポジトリ・キュー・マネージャーにメッセージを転送するために使用されるキュー

表 11. システムおよびデフォルト・オブジェクト: キュー (続き)

オブジェクト名	説明
SYSTEM.CLUSTER.HISTORY.QUEUE	キューは、サービス目的でクラスター状態情報の履歴を保管するために使用されます。
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	このキューを使用して、各クラスター送信側チャンネルの個々の送信キューを作成します。
SYSTEM.CLUSTER.REPOSITORY.QUEUE	すべてのリポジトリ情報を保管するために使用されるキュー
SYSTEM.CLUSTER.TRANSMIT.QUEUE	すべてのクラスターに対するすべてのメッセージのための伝送キュー
SYSTEM.DEAD.LETTER.QUEUE	送達不能 (未配布メッセージ) キュー
SYSTEM.DEFAULT.ALIAS.QUEUE	デフォルト別名キュー
SYSTEM.DEFAULT.INITIATION.QUEUE	デフォルト開始キュー
SYSTEM.DEFAULT.LOCAL.QUEUE	デフォルト・ローカル・キュー
SYSTEM.DEFAULT.MODEL.QUEUE	デフォルト・モデル・キュー
SYSTEM.DEFAULT.REMOTE.QUEUE	デフォルト・リモート・キュー
SYSTEM.JMS.TEMPQ.MODEL	JMS 一時キューのモデル
SYSTEM.MQEXPLORER.REPLY.MODEL	IBM MQ Explorer 応答先キュー。これは、IBM MQ Explorer に対する応答の一時動的キューを作成するモデル・キューです。
SYSTEM.MQSC.REPLY.QUEUE	MQSC コマンド応答先キュー。これは、リモート MQSC コマンドに応答するための一時動的キューを作成するモデル・キューです。
SYSTEM.PENDING.DATA.QUEUE	JMS における据え置きメッセージをサポートします。

表 12. システムおよびデフォルト・オブジェクト: トピック

オブジェクト名	説明
<u>SYSTEM.BASE.TOPIC</u>	ASPARENT を解決する基本トピック。特定のトピックに親管理トピック・オブジェクトがない場合、またはこれらの親オブジェクトに ASPARENT もある場合は、残りのどの ASPARENT 属性もこのオブジェクトから継承されます。
SYSTEM.DEFAULT.TOPIC	デフォルトのトピック定義

表 13. システムおよびデフォルト・オブジェクト: チャンネル

オブジェクト名	説明
SYSTEM.AUTO.RECEIVER	動的受信側チャンネル
SYSTEM.AUTO.SVRCONN	動的サーバー接続チャンネル
SYSTEM.DEF.CLUSRCVR	クラスターのためのデフォルト受信側チャンネル。クラスター内のキュー・マネージャー上に CLUSRCVR チャンネルを作成するときに指定しなかった属性のデフォルト値を提供するために使用されます。

表 13. システムおよびデフォルト・オブジェクト: チャネル (続き)

オブジェクト名	説明
SYSTEM.DEF.CLUSSDR	クラスターのためのデフォルト送信側チャネル。クラスター内のキュー・マネージャー上に CLUSSDR チャネルを作成するときに指定しなかった属性のデフォルト値を提供するために使用されます。
SYSTEM.DEF.RECEIVER	デフォルト受信側チャネル
SYSTEM.DEF.REQUESTER	デフォルト要求側チャネル
SYSTEM.DEF.SENDER	デフォルト送信側チャネル
SYSTEM.DEF.SERVER	デフォルト・サーバー・チャネル
SYSTEM.DEF.SVRCONN	デフォルト・サーバー接続チャネル
SYSTEM.DEF.CLNTCONN	デフォルト・クライアント接続チャネル
V9.0.0 SYSTEM.DEF.AMQP	デフォルト AMQP チャネル。

表 14. システムおよびデフォルト・オブジェクト: 認証情報オブジェクト

オブジェクト名	説明
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	CRLLDAP タイプの認証情報オブジェクトを定義するためのデフォルト認証情報オブジェクト。
SYSTEM.DEFAULT.AUTHINFO.OCSP	OCSP タイプの認証情報オブジェクトを定義するためのデフォルト認証情報オブジェクト。

表 15. システムおよびデフォルト・オブジェクト: リスナー

オブジェクト名	説明
SYSTEM.DEFAULT.LISTENER.TCP	デフォルトの TCP リスナー。
SYSTEM.DEFAULT.LISTENER.LU62 ¹	デフォルトの LU62 リスナー。
SYSTEM.DEFAULT.LISTENER.NETBIOS ¹	デフォルトの NETBIOS リスナー。
SYSTEM.DEFAULT.LISTENER.SPX ¹	デフォルトの SPX リスナー。

1. Windows のみ

表 16. システムおよびデフォルト・オブジェクト: 名前リスト

オブジェクト名	説明
SYSTEM.DEFAULT.NAMELIST	デフォルト名前リスト

表 17. システムおよびデフォルト・オブジェクト: プロセス

オブジェクト名	説明
SYSTEM.DEFAULT.PROCESS	デフォルトのプロセス定義

表 18. システムおよびデフォルト・オブジェクト: サービス

オブジェクト名	説明
SYSTEM.DEFAULT.SERVICE	デフォルト・サービス

表 18. システムおよびデフォルト・オブジェクト: サービス (続き)

オブジェクト名	説明
SYSTEM.BROKER	パブリッシュ/サブスクライブ・ブローカー

Windows デフォルト構成オブジェクト

Windows システムでは、IBM MQ Postcard アプリケーションを使用して、デフォルト構成をセットアップできます。

注: 他のキュー・マネージャーがコンピューター上に存在している場合は、デフォルト構成を設定できません。

Windows デフォルト構成オブジェクトに使用される名前の多くは、短い TCP/IP 名を使用する必要があります。これはドメイン部分を含まないコンピューターの TCP/IP 名です。例えば、コンピューター mycomputer.hursley.ibm.com の短い TCP/IP 名は mycomputer です。この TCP/IP 名を切り捨てなければならない場合、最後の文字がピリオド (.) のときは、そのピリオドは削除されます。

短い TCP/IP 名の中の、IBM MQ オブジェクト名には無効な文字 (例えば、ハイフン) はすべて、下線文字に置き換えられます。

IBM MQ オブジェクト名に有効な文字は、a から z まで、A から Z まで、0 から 9 まで、および /、%、.、_ の 4 つの特殊文字です。

Windows デフォルト構成のクラスター名は DEFAULT_CLUSTER です。

キュー・マネージャーがリポジトリ・キュー・マネージャーでない場合、94 ページの表 19 に示したオブジェクトが作成されます。

表 19. Windows デフォルト構成アプリケーションにより作成されるオブジェクト

オブジェクト	名前
キュー・マネージャー	接頭部に QM_ という文字が付いた短い TCP/IP 名。キュー・マネージャー名の最大長は 48 文字です。この制限を超える名前は、48 文字に切り捨てられます。名前の最後の文字がピリオド (.) の場合は、スペース () に置き換えられます。 キュー・マネージャーには、コマンド・サーバー、チャンネル・リスナー、およびチャンネル・イニシエーターが関連付けられます。チャンネル・リスナーは、ポート番号 1414 の標準 IBM MQ ポートで listen します。このデフォルト構成キュー・マネージャーが存在している限り、このマシン上に作成されたその他のキュー・マネージャーは、ポート 1414 を使用できません。
汎用クラスター受信側チャンネル	先頭に文字 TO_QM_ が付いた短い TCP/IP 名。総称クラスター受信側名の最大長は 20 文字です。この制限を超える名前は、20 文字に切り捨てられます。名前の最後の文字がピリオド (.) の場合は、スペース () に置き換えられます。
クラスター送信側チャンネル	クラスター送信側チャンネルは、最初は TO_ + QMNAME + という名前で作成されます。IBM MQ がデフォルト構成クラスターのリポジトリ・キュー・マネージャーへの接続を確立すると、この名前はデフォルト構成クラスターのリポジトリ・キュー・マネージャーの名前に置き換えられ、接頭部に文字 TO_ が付きます。クラスター送信側チャンネル名の最大長は 20 文字です。この制限を超える名前は、20 文字に切り捨てられます。名前の最後の文字がピリオド (.) の場合は、スペース () に置き換えられます。
ローカル・メッセージ・キュー	ローカル・メッセージ・キューは default と呼ばれます。

表 19. Windows デフォルト構成アプリケーションにより作成されるオブジェクト (続き)

オブジェクト	名前
IBM MQ Postcard アプリケーションで使用されるローカル・メッセージ・キュー	IBM MQ Postcard アプリケーションで使用されるローカル・メッセージ・キューは、postcard と呼ばれます。
サーバー接続チャンネル	サーバー接続チャンネルにより、クライアントはキュー・マネージャーに接続できます。その名前は短い TCP/IP 名で、接頭部に文字 S_ が付きます。サーバー接続チャンネル名の最大長は 20 文字です。この制限を超える名前は、20 文字に切り捨てられます。名前の最後の文字がピリオド (.) の場合は、スペース () に置き換えられます。

キュー・マネージャーがリポジトリ・キュー・マネージャーである場合、デフォルト構成は、[94 ページ](#)の表 19 で説明した構成に類似しますが、次の点が異なります。

- キュー・マネージャーは、デフォルト構成クラスター用のリポジトリ・キュー・マネージャーとして定義されます。
- クラスター送信側チャンネルは定義されていません。
- 先頭に文字 clq_default_ の付いた短い TCP/IP 名であるローカル・クラスター・キューが作成されます。この名前の最大長は 48 文字です。この長さをを超える名前は、48 文字に切り捨てられます。

リモート管理機能を要求すると、サーバー接続チャンネル、SYSTEM.ADMIN.SVRCONN も作成されます。

SYSTEM.BASE.TOPIC

ASPARENT を解決する基本トピック。特定のトピックに親管理トピック・オブジェクトがない場合、またはこれらの親オブジェクトに ASPARENT もある場合は、残りのどの ASPARENT 属性もこのオブジェクトから継承されます。

表 20. SYSTEM.BASE.TOPIC のデフォルト値

パラメーター	値
TOPICSTR	"
CLROUTE	DIRECT
CLUSTER	デフォルト値は空ストリングです。
COMMINFO	SYSTEM.DEFAULT.COMMINFO.MULTICAST
DEFPRESP	SYNC
DEFPRTY	0
DEFPSIST	NO
DESCR	「属性の解決のための基本トピック」
DURSUB	YES
MCAST	DISABLED
MDURMDL	SYSTEM.DURABLE.MODEL.QUEUE
MNDURMDL	SYSTEM.NDURABLE.MODEL.QUEUE
NPMSGDLV	ALLAVAIL
PMSGDLV	ALLDUR
PROXYSUB	FIRSTUSE
PUB	ENABLED

表 20. SYSTEM.BASE.TOPIC のデフォルト値 (続き)

パラメーター	値
PUBSCOPE	ALL
 QSGDISP (z/OS プラットフォームのみ)	QMGR
SUB	ENABLED
SUBSCOPE	ALL
USEDLQ	YES
WILDCARD	PASSTHRU

このオブジェクトが存在しない場合でも、IBM MQ は、トピック・ツリーのうち、さらに上位の親トピックによって解決されない ASPARENT 属性に対して、そのデフォルト値を使用します。

SYSTEM.BASE.TOPIC の PUB 属性または SUB 属性を DISABLED に設定すると、トピック・ツリーに含まれているトピックのパブリッシュまたはサブスクライブの操作をアプリケーションが実行できなくなります。ただし、以下の 2 つの例外があります。

1. トピック・ツリーに含まれているトピック・オブジェクトのうち、PUB または SUB が明示的に ENABLE に設定されているトピック・オブジェクト。アプリケーションは、これらのトピックとその子トピックのパブリッシュまたはサブスクライブの操作を実行できます。
2. SYSTEM.BASE.TOPIC の PUB 属性や SUB 属性を DISABLED に設定しても、SYSTEM.BROKER.ADMIN.STREAM のパブリケーションやサブスクリプションは無効になりません。

PUB パラメーターの特別な処理も参照してください。

IBM i IBM i のシステム・オブジェクトおよびデフォルト・オブジェクト

CRTMQM コマンドを使用してキュー・マネージャーを作成する場合、システム・オブジェクトおよびデフォルト・オブジェクトが自動的に作成されます。

- システム・オブジェクトは、キュー・マネージャーまたはチャネルの操作に必要な IBM MQ オブジェクトです。
- デフォルト・オブジェクトは、オブジェクトのすべての属性を定義します。ローカル・キューなどのオブジェクトを作成する場合、明示的に指定していない属性はすべてデフォルト・オブジェクトから継承されます。

次の表は、**CRTMQM** により作成されるシステム・オブジェクトとデフォルト・オブジェクトを示しています。

- [97 ページの表 21](#) は、システム・オブジェクトとデフォルト・キュー・オブジェクトを示しています。
- [98 ページの表 22](#) は、システム・オブジェクトとデフォルト・チャネル・オブジェクトを示しています。
- [99 ページの表 23](#) は、システム・オブジェクトとデフォルトの認証情報オブジェクトを示しています。
- [99 ページの表 24](#) は、システム・オブジェクトとデフォルトのリスナー・オブジェクトを示しています。
- [99 ページの表 25](#) は、システム・オブジェクトとデフォルトの名前リスト・オブジェクトを示しています。
- [100 ページの表 26](#) は、システム・オブジェクトとデフォルトのプロセス・オブジェクトを示しています。
- [100 ページの表 27](#) システム・オブジェクトとデフォルトのサービス・オブジェクトを示しています。

表 21. システムおよびデフォルト・オブジェクト: キュー

オブジェクト名	説明
SYSTEM.ADMIN.ACCOUNTING.QUEUE	アプリケーションがキュー・マネージャーから切断されたときに生成される、アカウントング・メッセージ・データ。
SYSTEM.ADMIN.ACTIVITY.QUEUE	アクティビティ報告書のメッセージ・データ。
SYSTEM.ADMIN.CHANNEL.EVENT	チャンネルのためのイベント・キュー
SYSTEM.ADMIN.COMMAND.QUEUE	管理者コマンド・キュー。リモート MQSC コマンドおよび PCF コマンドに使用されます。
SYSTEM.ADMIN.LOGGER.EVENT	ロガー・イベント (ジャーナル・レシーバー) のメッセージ・データ。
SYSTEM.ADMIN.PERFM.EVENT	パフォーマンス・イベントのイベント・キュー
SYSTEM.ADMIN.PUBSUB.EVENT	システムのパブリッシュ/サブスクライブに関連したイベント・キュー
SYSTEM.ADMIN.QMGR.EVENT	キュー・マネージャー・イベントのイベント・キュー
SYSTEM.ADMIN.STATISTICS.QUEUE	MQI、キュー、およびチャンネル統計のメッセージ・データ・キュー。
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	経路トレース応答のメッセージ・データ・キュー。
SYSTEM.AUTH.DATA.QUEUE	オブジェクト権限マネージャー (OAM) によって使用されます。
SYSTEM.BROKER.ADMIN.STREAM	キューに入れられたパブリッシュ/サブスクライブ・インターフェースによって使用される管理ストリーム。
SYSTEM.BROKER.CONTROL.QUEUE	パブリッシュ/サブスクライブ・インターフェースの制御キュー。
SYSTEM.BROKER.DEFAULT.STREAM	キューに入れられたパブリッシュ/サブスクライブ・インターフェースによって使用されるデフォルトのストリーム。
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	ブローカー間の通信キュー。
SYSTEM.CHANNEL.INITQ	チャンネル開始キュー
SYSTEM.CHANNEL.SYNCQ	チャンネルの同期データを保持するキュー
SYSTEM.CHLAUTH.DATA.QUEUE	IBM MQ のチャンネル認証データ・キュー
SYSTEM.DURABLE.MODEL.QUEUE	管理対象永続サブスクリプションのモデルとして使用されるキュー。
SYSTEM.DURABLE.SUBSCRIBER.QUEUE	永続サブスクリプションの持続コピーをキュー・マネージャーに保持するために使用されるキュー。
SYSTEM.CICS.INITIATION.QUEUE	デフォルト CICS 開始キュー
SYSTEM.CLUSTER.COMMAND.QUEUE	リポジトリ・キュー・マネージャーにメッセージを転送するために使用されるキュー
SYSTEM.CLUSTER.HISTORY.QUEUE	キューは、サービス目的でクラスター状態情報の履歴を保管するために使用されます。

表 21. システムおよびデフォルト・オブジェクト: キュー (続き)

オブジェクト名	説明
SYSTEM.CLUSTER.REPOSITORY.QUEUE	すべてのリポジトリ情報を保管するために使用されるキュー
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	このキューを使用して、各クラスター送信側チャンネルの個々の送信キューを作成します。
SYSTEM.CLUSTER.TRANSMIT.QUEUE	すべてのクラスターに対するすべてのメッセージのための伝送キュー
SYSTEM.DEAD.LETTER.QUEUE	送達不能 (未配布メッセージ) キュー
SYSTEM.DEFAULT.ALIAS.QUEUE	デフォルト別名キュー
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	デフォルト認証情報定義
SYSTEM.DEFAULT.INITIATION.QUEUE	デフォルト開始キュー
SYSTEM.DEFAULT.LOCAL.QUEUE	デフォルト・ローカル・キュー
SYSTEM.DEFAULT.MODEL.QUEUE	デフォルト・モデル・キュー
SYSTEM.DEFAULT.REMOTE.QUEUE	デフォルト・リモート・キュー
SYSTEM.JMS.TEMPQ.MODEL	JMS 一時キューのモデル
SYSTEM.HIERARCHY.STATE	IBM MQ 分散パブリッシュ/サブスクライブ階層関係の状態。
SYSTEM.INTER.QMGR.CONTROL	IBM MQ 分散パブリッシュ/サブスクライブの制御キュー。
SYSTEM.INTER.QMGR.FANREQ	IBM MQ 分散パブリッシュ/サブスクライブの内部プロキシ・サブスクリプション多分岐プロセスの入力キュー。
SYSTEM.INTER.QMGR.PUBS	IBM MQ 分散パブリッシュ/サブスクライブのパブリケーション
SYSTEM.MQEXPLORER.REPLY.MODEL	IBM MQ Explorer・コマンド応答先キュー。これは、IBM MQ Explorer に対する応答の一時動的キューを作成するモデル・キューです。
SYSTEM.MQSC.REPLY.QUEUE	MQSC コマンド応答先キュー。これは、リモート MQSC コマンドに応答するための一時動的キューを作成するモデル・キューです。
SYSTEM.NDURABLE.MODEL.QUEUE	管理対象非永続サブスクリプションのモデルとして使用されるキュー。
SYSTEM.PENDING.DATA.QUEUE	JMS における据え置きメッセージをサポートします。
SYSTEM.RETAINED.PUB.QUEUE	それぞれの保存パブリケーションのコピーをキュー・マネージャーに保持するために使用されるキュー。

表 22. システムおよびデフォルト・オブジェクト: チャンネル

オブジェクト名	説明
SYSTEM.AUTO.RECEIVER	動的受信側チャンネル
SYSTEM.AUTO.SVRCONN	動的サーバー接続チャンネル

表 22. システムおよびデフォルト・オブジェクト: チャンネル (続き)

オブジェクト名	説明
SYSTEM.DEF.CLNTCONN	デフォルト・クライアント接続チャンネル。キュー・マネージャーに CLNTCONN チャンネルを作成するときに指定しなかった属性のデフォルト値を提供するために使用されます。
SYSTEM.DEF.CLUSRCVR	クラスターのためのデフォルト受信側チャンネル。クラスター内のキュー・マネージャー上に CLUSRCVR チャンネルを作成するときに指定しなかった属性のデフォルト値を提供するために使用されます。
SYSTEM.DEF.CLUSSDR	クラスターのためのデフォルト送信側チャンネル。クラスター内のキュー・マネージャー上に CLUSSDR チャンネルを作成するときに指定しなかった属性のデフォルト値を提供するために使用されます。
SYSTEM.DEF.RECEIVER	デフォルト受信側チャンネル
SYSTEM.DEF.REQUESTER	デフォルト要求側チャンネル
SYSTEM.DEF.SENDER	デフォルト送信側チャンネル
SYSTEM.DEF.SERVER	デフォルト・サーバー・チャンネル
SYSTEM.DEF.SVRCONN	デフォルト・サーバー接続チャンネル

表 23. システムおよびデフォルト・オブジェクト: 認証情報オブジェクト

オブジェクト名	説明
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	認証タイプ CRLLDAP のデフォルト認証情報オブジェクト。
SYSTEM.DEFAULT.AUTHINFO.OCSP	認証タイプ OCSP のデフォルト認証情報オブジェクト。

表 24. システムおよびデフォルト・オブジェクト: リスナー

オブジェクト名	説明
SYSTEM.DEFAULT.LISTENER.TCP	TCP トランスポートのためのデフォルト・リスナー。

表 25. システムおよびデフォルト・オブジェクト: 名前リスト

オブジェクト名	説明
SYSTEM.DEFAULT.NAMELIST	デフォルト名前リスト定義
SYSTEM.QPUBSUB.QUEUE.NAMELIST	キューに入れられたパブリッシュ/サブスクライブ・インターフェースによってモニターされるキュー名のリスト。
SYSTEM.QPUBSUB.SUBPOINT.NAMELIST	トピック・オブジェクトをサブスクリプション・ポイントと一致させるために、キューに入れられたパブリッシュ/サブスクライブ・インターフェースによって使用されるトピック・オブジェクトのリスト。

表 26. システムおよびデフォルト・オブジェクト: プロセス	
オブジェクト名	説明
SYSTEM.DEFAULT.PROCESS	デフォルトのプロセス定義

表 27. システムおよびデフォルト・オブジェクト: サービス	
オブジェクト名	説明
SYSTEM.DEFAULT.SERVICE	デフォルト・サービス

スタanzas情報

ここでは、スタanzas内の情報を構成する際に役立つ情報と、mqc.ini、qm.ini、およびmqclient.ini ファイルの内容の一覧を示します。

スタanzasの構成

以下のリンクは、企業内で1つ以上のシステムを構成する際に役立ちます。

- [IBM MQ 構成情報の変更](#)は、以下の構成に役立ちます。
 - AllQueueManagers スタanzas
 - DefaultQueueManager スタanzas
 - ExitProperties スタanzas
 - LogDefaults スタanzas
 - qm.ini ファイルの Security スタanzas
- [キュー・マネージャー構成情報の変更](#)は、以下の構成に役立ちます。
 -  AccessMode スタanzas (Windows のみ)
 - Service スタanzas - インストール可能なサービス用
 - Log スタanzas
 -   RestrictedMode スタanzas (UNIX and Linux システムのみ)
 - XAResourceManager スタanzas
 - TCP、LU62、および NETBIOS スタanzas
 - ExitPath スタanzas
 - QMErrorLog スタanzas
 - SSL スタanzas
 - ExitPropertiesLocal スタanzas
- [構成サービスおよびコンポーネント](#)は、以下の構成に役立ちます。
 - Service スタanzas
 - ServiceComponent スタanzas

また、UNIX and Linux および Windows プラットフォームでさまざまなサービスにどのように使用されるかを示すリンクも含まれています。
- [API 出口の構成](#)は、以下の構成に役立ちます。
 - AllActivityTrace スタanzas
 - ApplicationTrace スタanzas
- [アクティビティ・トレース動作の構成ファイル](#)は、以下の構成に役立ちます。
 - ApiExitCommon スタanzas

- *ApiExitTemplate* スタンザ
 - *ApiExitLocal* スタンザ
 - クライアントの構成情報は、以下の構成に役立ちます。
 - *CHANNELS* スタンザ
 - *ClientExitPath* スタンザ
 - **Windows** *LU62*、*NETBIOS*、および *SPX* スタンザ (Windows のみ)
 - *MessageBuffer* スタンザ
 - *SSL* スタンザ
 - *TCP* スタンザ
 - 102 ページの『分散キューイング用の構成ファイル・スタンザ』は、以下の構成に役立ちます。
 - *CHANNELS* スタンザ
 - *TCP* スタンザ
 - *LU62* スタンザ
 - *NETBIOS*
 - *ExitPath* スタンザ
 - 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は、以下の構成に役立ちます。
 - *PersistentPublishRetry* 属性
 - *NonPersistentPublishRetry* 属性
 - *PublishBatchSize* 属性
 - *PublishRetryInterval* 属性
- これらは、*Broker* スタンザに含まれています。



重要: *Broker* スタンザが必要な場合は、それを作成する必要があります。

構成ファイル

以下を参照してください。

- [mqs.ini](#) ファイル
- [qm.ini](#) ファイル
- [mqclient.ini](#) ファイル

これらには、各構成ファイルで可能なスタンザの一覧が示されています。



mqs.ini ファイル

UNIX and Linux システム用の IBM MQ 構成ファイルの例に、mqs.ini ファイルの例を示しています。

mqs.ini ファイルには、以下のスタンザを含めることができます。

- [AllQueueManagers](#)
- [DefaultQueueManager](#)
- [ExitProperties](#)
- [LogDefaults](#)

さらに、キュー・マネージャーごとに1つの [QueueManager](#) スタンザがあります。

qm.ini ファイル

UNIX and Linux システム用の IBM MQ のキュー・マネージャー構成ファイルの例に、qm.ini ファイルの例を示します。

qm.ini ファイルには、以下のスタンザを含めることができます。

- [ExitPath](#)
- [ログ](#)
- [QMErrorLog](#)
- [QueueManager](#)
- [セキュリティ](#)
- [Service](#) および [ServiceComponent](#)

 [InstallableServices](#) を構成するには、[Service](#) および [ServiceComponent](#) スタンザを使用します。

- 接続 ([DefaultBind](#) タイプ の場合)



重要: [Connection](#) スタンザが必要な場合は、それを作成する必要があります。

- [SSL](#) および [TLS](#)
- [TCP](#)、[LU62](#)、および [NETBIOS](#)
- [XAResourceManager](#)

さらに、以下のものを変更できます。

-  [AccessMode](#) (Windows のみ)
-  [RestrictedMode](#) (UNIX and Linux システムのみ)

この操作には [crtmqm](#) コマンドを使用します。

mqclient.ini ファイル

mqclient.ini ファイルには、以下のスタンザを含めることができます。

- [CHANNELS](#)
- [ClientExitPath](#)
- [LU62](#)、[NETBIOS](#)、および [SPX](#)
- [MessageBuffer](#)
- [SSL](#)
- [TCP](#)

さらに、[PreConnect](#) スタンザを使用して [PreConnect](#) 出口を構成する必要が生じる場合もあります。

分散キューイング用の構成ファイル・スタンザ

分散キューイングに関連した、キュー・マネージャー構成ファイル qm.ini のスタンザの説明。

このトピックでは、分散キューイングに関連するキュー・マネージャー構成ファイル内のスタンザを示しています。これは、IBM MQ for Multiplatforms 用のキュー・マネージャー構成ファイルに適用されます。すべてのプラットフォームで、このファイル名は [qm.ini](#) です。

分散キューイングに関連するスタンザとして次のスタンザがあります。

- チャンネル
- TCP
- LU62
- NETBIOS
- EXITPATH

103 ページの図 6 には、これらのスタanzasを使用して設定できる値を示しています。どれか1つのスタanzasを定義する場合、各項目ごとに改行する必要はありません。コメントを入れる場合は、セミコロン (;) またはハッシュ文字 (#) を使用できます。

```
CHANNELS:
  MAXCHANNELS=n           ; Maximum number of channels allowed, the
                          ; default value is 100.
  MAXACTIVECHANNELS=n    ; Maximum number of channels allowed to be active at
                          ; any time, the default is the value of MaxChannels.
  MAXINITIATORS=n        ; Maximum number of initiators allowed, the default
                          ; and maximum value is 3.
  MQIBINDTYPE=type1      ; Whether the binding for applications is to be
                          ; "fastpath" or "standard".
                          ; The default is "standard".
  PIPELINELENGTH=n       ; The maximum number of concurrent threads a channel will use.
                          ; The default is 1. Any value greater than 1 is treated as 2.
  ADOPTNEWMCA=chltype    ; Stops previous process if channel fails to start.
                          ; The default is "NO".
  ADOPTNEWMCATIMEOUT=n   ; Specifies the amount of time that the new
                          ; process should wait for the old process to end.
                          ; The default is 60.
  ADOPTNEWMCACHECHECK=   ; Specifies the type checking required.
    typecheck            ; The default is "NAME", "ADDRESS", and "QM".
  CHLAUTHEARLYADOPT=Y/N ; The order in which connection authentication and channel
authentication rules are ; processed. If not present in the qm.ini file the default is "N".
From MQ9.0.4 all
  PASSWORDPROTECTION=    ; queue managers are created with a default of "Y"
than using TLS.         ; From MQ8.0, set protected passwords in the MQCSP structure, rather
    options              ; The options are "compatible", "always", "optional" and "warn"
                          ; The default is "compatible".
  CHLAUTHISSUEWARN=Y     ; If you want message AMQ9787 to be generated when you set the WARN=YES
attribute
                          ; on the SET CHLAUTH command.
TCP:                    ; TCP entries
  PORT=n                ; Port number, the default is 1414
  KEEPALIVE=Yes         ; Switch TCP/IP KeepAlive on
LU62:
  LIBRARY2=DLLName2     ; Used if code is in two libraries
EXITPATH:2             Location of user exits
EXITPATHS=              ; String of directory paths.
```

図 6. 分散キューイングの qm.ini スタanzas

注:

1. MQIBINDTYPE は、次のプラットフォームにのみ適用されます。

-  AIX
-  HP-UX
-  IBM i
-  Solaris

2. EXITPATH は、次のプラットフォームにのみ適用されます。

-  AIX
-  HP-UX
-  Solaris
-  Windows

関連情報

構成

 z/OS の構成

Windows、UNIX、および Linux システムでの構成情報の変更

チャネルの属性

このセクションでは、チャネル定義に保持されるチャネル属性について説明します。

チャネルごとに特定の一組の環境に最適のチャネル属性を選択してください。ただし、チャネルが実行されているときには、開始折衝中に実際の値が変更された可能性があります。[チャネルの作成](#)を参照してください。

多くの属性にはデフォルト値が設定されており、ほとんどのチャネルではこれらの値を使用できます。ただし、デフォルト値が最適な値ではない環境の場合、このセクションを参照して正しい値を選択するための説明をお読みください。

クラスター・チャネルの場合は、ターゲット・キュー・マネージャーにあるクラスター受信側チャネルにクラスター・チャネル属性を指定します。一致するクラスター送信側チャネルで指定したすべての属性は、無視される可能性が高くなります。[クラスター・チャネル](#)を参照してください。

注：IBM MQ for IBM iでは、ほとんどの属性は*SYSDFTCHLとして指定できます。これは、使用するシステムのシステム・デフォルト・チャネルから値が取られることを表します。

チャネル属性とチャネル・タイプ

さまざまなタイプのチャネルが、さまざまなタイプのチャネル属性をサポートします。

IBM MQ チャネル属性のチャネル・タイプは、以下の表にリストされています。

注：クラスター・チャネル(表の CLUSSDR 列と CLUSRCVR 列)では、可能であれば両方のチャネルに属性を設定し、設定を確実に同じにします。これらの設定が一致していない場合、CLUSRCVR チャネルで指定した設定の方が使用されます。これについては、[クラスター・チャネル](#)で説明しています。

属性フィールド	MQSC コマンド・パラメーター	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AMQP
Alter date (変更日付)	ALTDATE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	はい
Alter time (変更時刻)	ALTTIME	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	はい
 AMQP キープアライブ	AMQPKA									はい
Batch heartbeat interval (バッチ・ハートビート間隔)	BATCHHB	Yes	Yes					Yes	Yes	
Batch interval (バッチ間隔)	BATCHINT	Yes	Yes					Yes	Yes	
バッチ制限	BATCHLIM	Yes	Yes					Yes	Yes	

表 28. チャンネル属性とチャンネル・タイプ (続き)

属性フィールド	MQSC コマンド・パラメータ	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	<small>V5.0.0</small> AMQP
バッチのサイズ	BATCHSZ	Yes	Yes	Yes	Yes			Yes	Yes	
証明書ラベル	CERTLABL	Yes	Yes	Yes	Yes	Yes	Yes	はい 109 ページの『1』	Yes	<small>V5.0.0</small> はい
チャンネル名	CHANNEL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	<small>V5.0.0</small> はい
チャンネル統計	STATCHL	Yes	Yes	Yes	Yes			Yes	Yes	
チャンネル・タイプ	CHLTYPE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	<small>V5.0.0</small> はい
クライアント・チャンネルの重み	CLNTWGH T					Yes				
クラスター	CLUSTER							Yes	Yes	
クラスター名前リスト	CLUSNL							Yes	Yes	
Cluster workload priority (クラスター・ワークロードの優先順位)	CLWLPR T Y							Yes	Yes	
Cluster workload rank (クラスター・ワークロードのランク)	CLWLRA N K							Yes	Yes	
Cluster workload weight (クラスター・ワークロード・ウェイト)	CLWLWG H T							Yes	Yes	
Connection affinity (接続アフィニティー)	AFFINITY					Yes				
接続名	CONNAM E	Yes	Yes		Yes	Yes		Yes	Yes	
メッセージ変換	CONVER T	Yes	Yes					Yes	Yes	
Data compression (データ圧縮)	COMPMS G	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
説明	DESCR	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	<small>V5.0.0</small> はい

表 28. チャネル属性とチャネル・タイプ (続き)

属性フィールド	MQSC コマンド・パラメータ	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AMQP
切断インターバル	DISCINT	Yes	Yes				はい 109 ページの『2』	Yes	Yes	
属性指定 109 ページの『2』	QSGDISP	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
ヘッダー圧縮	COMPHDR	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
ハートビート間隔	HBINT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
キープアライブ間隔	KAINT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
Local address (ローカル・アドレス)	LOCLADDR	Yes	Yes		Yes	Yes		Yes	Yes	はい
長期再試行カウント	LONGRTY	Yes	Yes					Yes	Yes	
長期再試行間隔	LONGTMR	Yes	Yes					Yes	Yes	
LU 6.2 モード名	MODENAME	Yes	Yes		Yes	Yes		Yes	Yes	
LU 6.2 トランザクション・プログラム名	TPNAME	Yes	Yes		Yes	Yes		Yes	Yes	
最大インスタンス数	MAXINST						Yes			はい
Maximum instances per client (クライアントごとの最大インスタンス数)	MAXINSTC						Yes			
Maximum message length (最大メッセージ長)	MAXMSGL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	はい
メッセージ・チャネル・エージェント名	MCANAME	Yes	Yes		Yes			Yes	Yes	
Message channel agent type (メッセージ・チャネル・エージェント・タイプ)	MCATYPE	Yes	Yes		Yes			Yes	Yes	

表 28. チャネル属性とチャネル・タイプ (続き)

属性フィールド	MQSC コマンド・パラメータ	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	V9.0.0 AMQP
<u>メッセージ・チャネル・エージェント・ユーザー</u>	MCAUSER	Yes	Yes	Yes	Yes		Yes	Yes	Yes	V9.0.0 はい
<u>メッセージ出口名</u>	MSGEXIT	Yes	Yes	Yes	Yes			Yes	Yes	
<u>メッセージ出口ユーザー・データ</u>	MSGDATA	Yes	Yes	Yes	Yes			Yes	Yes	
<u>メッセージ再試行出口名</u>	MREXIT			Yes	Yes				Yes	
<u>Message-retry exit user data (メッセージ再試行出口ユーザー・データ)</u>	MRDATA			Yes	Yes				Yes	
<u>メッセージ再試行カウント</u>	MRRTY			Yes	Yes				Yes	
<u>メッセージ再試行間隔</u>	MRTMR			Yes	Yes				Yes	
<u>モニター</u>	MONCHL	Yes	Yes	Yes	Yes		Yes	Yes	Yes	
<u>Network-connection priority (ネットワーク接続優先順位)</u>	NETPRTY								Yes	
<u>非持続性メッセージ速度</u>	NPMSPEED	Yes	Yes	Yes	Yes			Yes	Yes	
<u>PASSWORD</u>	パスワード	Yes	Yes		Yes	Yes		Yes		
V9.0.0 V9.0.0 <u>ポート番号</u>	PORT									V9.0.0 はい
<u>プロパティー制御</u>	PROPCTL	Yes	Yes					Yes	Yes	
<u>PUT authority (PUT 権限)</u>	PUTAUT			Yes	Yes		はい 109ページの『2』		Yes	
<u>キュー・マネージャー名</u>	QMNAME					Yes				
<u>受信出口名</u>	RCVEXIT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
<u>受信出口ユーザー・データ</u>	RCVDATA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
<u>セキュリティ出口名</u>	SCYEXIT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	

表 28. チャネル属性とチャネル・タイプ (続き)

属性フィールド	MQSC コマンド・パラメータ	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	<small>V9.0.0</small> AMQP
<u>セキュリティー出口ユーザー・データ</u>	SCYDATA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
<u>送信出口名</u>	SENDEXIT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
<u>送信出口ユーザー・データ</u>	SENDDATA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
<u>シーケンス番号の折り返し</u>	SEQWRAP	Yes	Yes	Yes	Yes			Yes	Yes	
<u>共用接続</u>	SHARECNV					Yes	Yes			
<u>短期再試行カウント</u>	SHORTRTY	Yes	Yes					Yes	Yes	
<u>短期再試行間隔</u>	SHORTTMR	Yes	Yes					Yes	Yes	
<u>SSL 暗号仕様</u>	SSLCIPH	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	<small>V9.0.0</small> はい
<u>SSL クライアント認証</u>	SSLCAUTH		Yes	Yes	Yes		Yes		Yes	<small>V9.0.0</small> はい
<u>SSL ピア</u>	SSLPEER	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	<small>V9.0.0</small> はい
<small>V9.0.0</small> <small>V9.0.0</small> トピック・ルート	TPROOT									<small>V9.0.0</small> はい
<u>伝送キュー名</u>	XMITQ	Yes	Yes							
<u>トランスポート・タイプ</u>	TRPTYPE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
<small>V9.0.0</small> <small>V9.0.0</small> クライアント ID を使用	USECLTID									<small>V9.0.0</small> はい
<u>Use Dead-Letter Queue (送達不能キューの使用)</u>	USEDLQ	Yes	Yes	Yes	Yes			Yes	Yes	
<u>ユーザー ID</u>	ユーザー ID	Yes	Yes		Yes	Yes		Yes		

表 28. チャネル属性とチャネル・タイプ (続き)

属性フィールド	MQSC コマンド・パラメータ	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AMQP
<p>注:</p> <ol style="list-style-type: none"> この属性を照会したり CLUSSDR チャネルに対して設定したりできる管理インターフェースはありません。MQRCCF_WRONG_CHANNEL_TYPE メッセージを受け取ります。ただし、この属性は CLUSSDR チャネル・オブジェクト (MQCD 構造体を含む) に存在しており、必要な場合は CHAD 出口でプログラムマチックに設定できます。  z/OS でのみ有効です。 										

関連概念

109 ページの『チャネル属性 (アルファベット順)』

このセクションでは、チャネル・オブジェクトの各属性を説明し、その有効な値、および該当する場合はその使用法に関する注を記載しています。

関連情報

[MQSC リファレンス](#)

チャネル属性 (アルファベット順)

このセクションでは、チャネル・オブジェクトの各属性を説明し、その有効な値、および該当する場合はその使用法に関する注を記載しています。

一部のプラットフォームの IBM MQ では、このセクションに示されるすべての属性を使用できない場合があります。例外とプラットフォームによる違いについては、関連する個々の属性の記述内で説明しています。

MQSC で指定できるキーワードは、属性ごとに括弧内に表示されています。

これらの属性は、アルファベット順に記載されています。

変更日付 (ALTDATE)

この属性は、定義が最後に変更された日付です。yyyy-mm-dd の形式で表されます。

この属性は、すべてのチャネル・タイプで有効です。

変更時刻 (ALTTIME)

この属性は、定義が最後に変更された時刻です。hh:mm:ss の形式で表されます。

この属性は、すべてのチャネル・タイプで有効です。

AMQP キープアライブ (AMQPKA)

AMQPKA 属性を使用すると、AMQP クライアント接続のキープアライブ時間を指定できます。AMQP クライアントがキープアライブ間隔内にフレームをまったく送信しなかった場合、接続は閉じられます。

AMQPKA 属性により、IBM MQ から AMQP クライアントに送信されるアイドル・タイムアウト属性の値が決まります。この属性は、ミリ秒単位の期間です。

AMQPKA が > 0 の値に設定された場合、IBM MQ は、アイドル・タイムアウト属性としてその半分の値を送ります。例えば、値が 10000 の場合、キュー・マネージャーはアイドル・タイムアウト値として 5000 を送信します。クライアントは、少なくとも 10000 ミリ秒ごとにデータが IBM MQ に送信されるようにする必要があります。その期間内に IBM MQ がデータを受け取らなかった場合、IBM MQ はクライアントとの

接続が失われたと想定して、`amqp:resource-limit-exceeded` エラー条件により接続を強制的にクローズします。

値の `AUTO` または `0` は、IBM MQ がアイドル・タイムアウト属性を AMQP クライアントに送らないことを意味します。

その場合でも、AMQP クライアントは独自のアイドル・タイムアウト値を送ることが可能です。そのようにした場合、IBM MQ はデータ (または空の AMQP フレーム) を少なくともその頻度で送り、使用可能であることをクライアントに通知します。

バッチ・ハートビート間隔 (BATCHEB)

この属性を使用して、送信側チャンネルは、メッセージのバッチをコミットする直前に、受信側チャンネルがまだアクティブであることを検証することができます。

したがって、バッチ・ハートビート間隔を使用すると、受信側チャンネルがアクティブではない場合に、バッチをバックアップできるため、バッチが未確定とならずに済みます。バッチをバックアップすることによって、メッセージは処理可能な状態にとどまるので、例えば、メッセージを別のチャンネルにリダイレクトできます。

バッチ・ハートビート間隔内に、送信側チャンネルに受信側チャンネルからの通信があった場合、受信側チャンネルはアクティブであると見なされます。その他の場合、検査のために「ハートビート」が受信側チャンネルに送信されます。送信側チャンネルは、チャンネル・ハートビート間隔 (HBINT) 属性で指定された秒数に基づいて、チャンネルの受信側からの応答をその間隔待機します。

値はミリ秒単位で、`0` から `999999` までの範囲内でなければなりません。ゼロの値は、バッチ・ハートビートが使用されないことを示します。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- クラスター送信側
- クラスター受信側

バッチ間隔 (BATCHINT)

この属性は、伝送キューにメッセージが無くてもチャンネルがバッチをオープンにしておく時間 (ミリ秒) です。

`0` から `999 999 999` までの任意の数をミリ秒数として指定できます。デフォルト値はゼロです。

バッチ間隔を指定しない場合、次の条件のいずれかが満たされたときにバッチがクローズします。

- `BATCHSZ` に指定された数のメッセージが送信された。
- `BATCHLIM` に指定されたバイト数が送信された。
- 伝送キューが空である。

伝送キューが頻繁に空になる負荷の軽いチャンネルでは、効率的なバッチ・サイズは `BATCHSZ` よりかなり小さい場合があります。

`BATCHINT` 属性を使用して、ショート・バッチ数を減らし、チャンネルをさらに効率的にすることができます。ただし、バッチが長くなるとメッセージがコミットされていないままの時間も長くなるので、応答に時間がかかる可能性があることに注意してください。

`BATCHINT` を指定した場合、バッチは以下の条件のいずれかと一致したときだけクローズします。

- `BATCHSZ` に指定された数のメッセージが送信された。
- `BATCHLIM` に指定されたバイト数が送信された。
- 伝送キューにはもうメッセージが残っておらず、(バッチの最初のメッセージが検索されてから) メッセージを待機する間に `BATCHINT` の時間間隔が経過した。

注: BATCHINT は、メッセージの待機時間の合計を指定します。伝送キューで既に使用可能になっているメッセージの検索時間もメッセージの転送時間も、BATCHINT には含まれません。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- クラスター送信側
- クラスター受信側

バッチ限界 (BATCHLIM)

この属性では、同期点を取る前にチャンネルを介して送信可能なデータ量の限度 (キロバイト単位) を指定します。

限度に達した際のメッセージがチャンネルを通過して送信された後に、同期点が取られます。

値は 0 から 999999 の範囲でなければなりません。デフォルト値は 5000 です。

この属性の値がゼロの場合、それはこのチャンネルに対するバッチに適用されるデータ限度がないことを意味します。

バッチは、次の条件のいずれかが満たされた場合に終了します。

- BATCHSZ メッセージが送信された。
- BATCHLIM バイトが送信された。
- 伝送キューが空で、BATCHINT が経過した。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- クラスター送信側
- クラスター受信側

このパラメーターは、すべてのプラットフォームでサポートされています。

バッチ・サイズ (BATCHSZ)

この属性は、同期点を取る前に送信されるメッセージの最大数です。

バッチ・サイズは、チャンネルがメッセージを転送する方法には影響を与えません。メッセージは常に個別に転送されますが、メッセージのコミットまたはバックアウトはバッチ単位で行われます。

パフォーマンスを向上させるために、バッチ・サイズを設定して、2つの同期点でやりとりされるメッセージの最大数を定義することができます。使用されるバッチ・サイズはチャンネルの開始時に折衝され、2つのチャンネル定義のうちの低いほうの値が採用されます。実現方法によっては、2つのチャンネル定義の低い方の値と2つのキュー・マネージャーの MAXUMSGS 値からバッチ・サイズが計算されます。バッチの実際のサイズはこの値より小さい場合があります。例えば、伝送キューにメッセージが残されていないかバッチ間隔が時間切れになると、バッチは完了してしまいます。

バッチ・サイズに大きな値を設定するとスループットが向上しますが、バックアウトや再送信の対象となるメッセージの数も多くなるため、リカバリーにかかる時間も長くなります。BATCHSZ のデフォルトは 50 であり、この値をまず試してみることをお勧めします。通信の信頼性が低く、リカバリーが必要になることが多い場合には、BATCHSZ の値を小さくした方がよいかもしれません。

同期点手順では、バッチ・コミット手順を調整するために、同期点が取られるたびにリンクを介して固有の論理作業単位 ID を交換する必要があります。

同期化されたバッチ・コミット手順が中断されると、未確定状態が発生する可能性があります。未確定状態は、メッセージ・チャンネルの開始時に自動的に解決されます。この方法で解決しない場合は、RESOLVE コマンドを使用した手操作による介入が必要になる可能性があります。

バッチ・サイズの値を選択する際には、以下の点を考慮してください。

- 値が大きすぎると、リンクの両側で占有されるキュー空間が過大になります。メッセージは、コミットされていないときにはキュー空間を占有し、コミットされるまでキューから除去することはできません。
- メッセージのフローが安定していることが予想される場合は、バッチ・サイズを増やすことによってチャンネルのパフォーマンスを向上できます。これは、同じ数のバイトを転送するのに多くの確認フローは必要ないためです。
- メッセージ・フローの特性から、メッセージが断続的に到着することが予想される場合は、切断時間間隔を比較的大きくしてバッチ・サイズに 1 を指定すると、良好なパフォーマンスが得られます。
- この値の範囲は 1 から 9999 までです。ただし、データ保全性の理由から、どの現行プラットフォーム接続するチャンネルにも 1 より大きいバッチ・サイズを指定する必要があります。1 は、IBM MQ for MVS 以外のバージョン 1 製品に使用する値です。
- 高速チャンネルでの非永続メッセージは、同期点を待たない場合でも、バッチ・サイズに数えられます。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- 受信側
- 要求側
- クラスター送信側
- クラスター受信側

証明書ラベル (CERTLABL)

この属性は、チャンネル定義の証明書ラベルを指定します。

ラベルにより、鍵リポジトリに含まれているどの個人証明書をリモート・ピアに送信するかを指定します。証明書は、[デジタル証明書ラベル](#)に記載されているように定義されます。

インバウンド・チャンネル (チャンネル RCVR、RQSTR、CLUSRCVR、修飾されていない SERVER、および SVRCONN を含む) は、リモート・ピアの IBM MQ のバージョンが証明書ラベル構成を完全にサポートしている、チャンネルが TLS CipherSpec を使用している場合にのみ、構成済みの証明書を送信します。そうでない場合、送信される証明書は、キュー・マネージャーの **CERTLABL** 属性によって決定されます。この制限があるのは、インバウンド・チャンネルの証明書ラベルの選択メカニズムが依存する TLS プロトコル拡張が、すべてのケースでサポートされるわけではないためです。特に、Java クライアント、JMS クライアント、および IBM MQ 8.0 より前のすべてのバージョンの IBM MQ は、この必要なプロトコル拡張をサポートしていないため、チャンネル固有のラベル設定にかかわらず、キュー・マネージャーの **CERTLABL** 属性で構成された証明書を受け取るしかありません。

修飾されていないサーバー・チャンネルとは、CONNAME フィールドが設定されていないチャンネルです。

この属性を照会したり CLUSSDR チャンネルに対して設定したりできる管理インターフェースはありません。MQRCCF_WRONG_CHANNEL_TYPE メッセージを受け取ります。ただし、この属性は CLUSSDR チャンネル・オブジェクト (MQCD 構造体を含む) に存在しており、必要な場合は CHAD 出口でプログラマチックに設定できます。

証明書ラベルに含まれる情報については、[デジタル証明書ラベルの要件に関する説明](#)を参照してください。

この属性は、すべてのチャンネル・タイプで有効です。

注：SSL/TLS では、CERTLABL は QMGR 定義で定義する必要があります。オプションで、CHANNEL 定義に CERTLABL を設定することができますが、CERTLABL をチャンネル属性として定義した場合でも、チャンネルは引き続きキュー・マネージャーのデフォルト CERTLABL を使用します。

CHANNEL 定義で CERTLABL を設定している場合であっても、キュー・マネージャー CERTLABL はチェックされ、有効な個人証明書である必要があります。

CSQ6SYSP モジュールで OPMODE を指定した場合でも、チャンネルは引き続きキュー・マネージャーのデフォルト CERTLABL を使用します。

チャンネル名 (CHANNEL)

この属性は、チャンネル定義の名前を指定します。

この名前には 20 文字まで使用できますが、メッセージ・チャンネルの両端で同じ名前を指定する必要があります。また実現方法によってはサイズが制限されている場合があるため、実際の文字数はより小さくしなければならないことがあります。

可能な場合、チャンネル名は、相互接続されたキュー・マネージャーのネットワーク内にある任意の 2 つのキュー・マネージャーにおいて、1 つのチャンネルに固有の名前にします。

この名前の指定には、以下の文字を使用してください。

英字	(A から Z、a から z。大文字と小文字には意味があります)
数字	(0-9)
ピリオド	(.)
スラッシュ	(/)
下線	(_)
パーセント記号	(%)

注:

1. ブランクを埋め込むことは許されず、また先行ブランクは無視されます。
2. EBCDIC カタカナを使用するシステムでは、小文字を使用することはできません。

この属性は、すべてのチャンネル・タイプで有効です。

Multi

Multiplatforms でのチャンネル統計 (STATCHL)

Multiplatforms で、この属性はチャンネルの統計データの収集方法を制御します。

指定できる値は以下のとおりです。

QMGR

このチャンネルの統計データ収集は、キュー・マネージャー属性 STATCHL の設定に基づいて行われます。この値がデフォルト値です。

OFF

このチャンネルの統計データ収集は、無効になります。

LOW

このチャンネルの統計データ収集は、低いデータ収集率で有効になります。

MEDIUM

このチャンネルの統計データ収集は、中程度のデータ収集率で有効になります。

終

このチャンネルの統計データ収集は、高いデータ収集率で有効になります。

チャンネル統計の詳細については、[モニター・リファレンス](#)を参照してください。

z/OS

z/OS システムでは、このパラメーターを有効にすると、選択した値に関係なく、単に統計データ収集がオンになります。LOW、MEDIUM、または HIGH のどれを指定しても、結果に違いはありません。チャンネル・アカウントング・レコードを収集するには、このパラメーターを有効にしなければなりません。

この属性は、以下のチャンネル・タイプで有効です。

- 送信者
- サーバー
- 受信側
- 要求側

- クラスター送信側
- クラスター受信側

チャンネル・タイプ (CHLTYPE)

この属性は、定義中のチャンネルのタイプを指定します。

指定できるチャンネル・タイプは次のとおりです。

メッセージ・チャンネル・タイプ:

- 送信者
- サーバー
- 受信側
- 要求側
- クラスター送信側
- クラスター受信側

MQI チャンネル・タイプ:

- クライアント接続 (Windows および UNIX のみ)

注: クライアント接続チャンネルを、その他のプラットフォームで使用するために z/OS でも定義することができます。

- サーバー接続
- AMQP

チャンネルの両端は、同じ名前をもち、次のように互換性を備えたタイプでなければなりません。

- 送信側と受信側
- 要求側とサーバー
- 要求側と送信側 (コールバックの場合)
- サーバーと受信側 (サーバーが送信側として使用されます)
- クライアント接続とサーバー接続
- クラスター送信側とクラスター受信側
- AMQP と AMQP

クライアント・チャンネル・ウェイト (CLNTWGHT)

この属性は、どのクライアント接続チャンネル定義を使用するかに影響を与える加重を指定します。

クライアント・チャンネル加重属性を使用すると、複数の適切な定義が選択可能な場合に、クライアント・チャンネル定義を加重に基づいてランダムに選択できます。

先頭がアスタリスクのキュー・マネージャー名を指定して (複数のキュー・マネージャー間でクライアント・ウェイト・บาลancingが可能になる)、クライアントが MQCONN 要求接続をキュー・マネージャー・グループに対して発行し、複数の適切なチャンネル定義がクライアント・チャンネル定義テーブル (CCDT) で選択可能な場合、使用する定義は加重に基づいてランダムに選択されます。適用可能な任意の CLNTWGHT(0) の定義が、アルファベット順に従って最初に選択されます。

0 から 99 の範囲の値を指定します。デフォルトは 0 です。

値として 0 を指定すると、ロード・บาลancingが実行されず、該当する定義がアルファベット順で選択されます。ロード・บาลancingを有効にするには、1 から 99 までの範囲の値を選択します (1 が最低の加重値、99 が最高の加重値です)。非ゼロの加重を持つ 2 つ以上のチャンネル間での接続の分散は、それらの加重の比率に比例したものになります。例えば、CLNTWGHT 値として 2、4、および 14 を持つ 3 つのチャンネルは、ほぼ 10%、20%、および 70% の時間の割合で選択されます。この分散は保証されているわけではありません。接続の AFFINITY 属性が PREFERRED に設定されている場合、最初の接続はクライアントの加重に応じてチャンネル定義を選択し、後続の接続は同じチャンネル定義を使用し続けます。

この属性は、クライアント接続チャンネル・タイプでのみ有効です。

クラスター (CLUSTER)

この属性は、チャンネルが属するクラスターの名前です。

最大長は 48 文字で、IBM MQ オブジェクトの命名規則に従います。

CLUSTER および CLUSNL の結果値は、そのどちらか一方だけ非ブランクにすることができます。一方の値を非ブランクにした場合、もう一方の値は必ずブランクにしてください。

この属性は、以下のチャンネル・タイプで有効です。

- クラスター送信側
- クラスター受信側

クラスターの名前リスト (CLUSNL)

この属性は、そのチャンネルが所属するクラスターのリストを指定した名前リスト名です。

CLUSTER および CLUSNL の結果値は、そのどちらか一方だけ非ブランクにすることができます。一方の値を非ブランクにした場合、もう一方の値は必ずブランクにしてください。

この属性は、以下のチャンネル・タイプで有効です。

- クラスター送信側
- クラスター受信側

CLWLPRTY チャンネル属性

CLWLPRTY チャンネル属性は、クラスター・ワークロード分散のために、チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLPRTY チャンネル属性は、使用可能なクラスター宛先の優先順位を設定するために使用します。IBM MQ は、クラスターの宛先の優先度が最も低い宛先を選択する前に、優先度が最も高い宛先を選択します。優先度が同じである複数の宛先が存在する場合は、最も長期間未使用になっている宛先を選択します。

有効な宛先が 2 つある場合は、この属性を使用して、フェイルオーバーを許可できます。メッセージは、最高の優先順位のチャンネルを持つキュー・マネージャーに移動されます。このキュー・マネージャーが使用不可になった場合、メッセージはその次に高い優先順位のキュー・マネージャーに移動されます。低い優先順位のキュー・マネージャーは、予備としての役割を果たします。

IBM MQ は、チャンネルに優先順位付けする前に、チャンネルの状況を検査します。使用可能なキュー・マネージャーのみが選択候補になります。

注:

- クラスター受信側チャンネルのターゲット・キュー・マネージャーで、この属性を指定します。一致するクラスター送信側チャンネルで指定したすべてのバランシングは、無視される可能性が高くなります。[クラスター・チャンネル](#)を参照してください。
- 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、クラスター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。

CLWLRANK チャンネル属性

CLWLRANK チャンネル属性は、クラスター・ワークロード分散に使用するチャンネルのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLRANK チャンネル属性は、別のクラスター内のキュー・マネージャーに送信されるメッセージの最終宛先を制御する場合に使用します。クラスターの交点にあるゲートウェイ・キュー・マネージャーにキュー・マネージャーを接続するチャンネルのランクを設定することによって、最終宛先の選択を制御します。

CLWLRANK を設定すると、メッセージは、相互接続されたクラスターを経由してランクの高い宛先に向かう指定経路を取ります。例えば、ランク 1 とランク 2 のチャンネルを使用する 2 つのキュー・マネージャーのどちらかにメッセージを送信できるゲートウェイ・キュー・マネージャーにメッセージが到着したとします。これらのメッセージは、最高ランクのチャンネル(この場合はランク 2 のキュー・マネージャーへのチャンネル)で接続されたキュー・マネージャーに自動的に送信されます。

IBM MQ は、チャンネル状況を検査する前にチャンネルのランクを取得します。チャンネル状況を検査する前にランクを取得するという事は、アクセスできないチャンネルでも選択の対象になることを意味します。この方法は、最終宛先が使用不可であったとしても、ネットワークを介してメッセージがルーティングされることを可能にします。

注:

- クラスター受信側チャンネルのターゲット・キュー・マネージャーで、この属性を指定します。一致するクラスター送信側チャンネルで指定したすべてのバランシングは、無視される可能性が高くなります。[クラスター・チャンネル](#)を参照してください。
- 優先順位属性 **CLWLPRTY** も使用した場合、IBM MQ は使用可能な宛先の中から選択します。最高ランクの宛先がチャンネルを使用できない場合、メッセージは伝送キューに保持されます。チャンネルが使用可能になると解放されます。ランク順で次に使用可能な宛先にメッセージが送信されるわけではありません。

CLWLWGHT チャンネル属性

CLWLWGHT チャンネル属性は、クラスター・ワークロード分散用に、**CLUSDR** および **CLUSRCVR** チャンネルに適用される重みを指定します。この値は、1 から 99 の範囲でなければなりません。1 が最低、99 が最高の重みです。

CLWLWGHT を使用して、より多くの処理能力があるサーバーに、より多くのメッセージを送信します。チャンネルの重みが高いほど、そのチャンネルを通して多くのメッセージが送信されます。

注:

- クラスター受信側チャンネルのターゲット・キュー・マネージャーで、この属性を指定します。一致するクラスター送信側チャンネルで指定したすべてのバランシングは、無視される可能性が高くなります。[クラスター・チャンネル](#)を参照してください。
- いずれかのチャンネルで **CLWLWGHT** がデフォルトの 50 から変更された場合、ワークロード・バランシングはいずれかのクラスター・キューに送信されたメッセージのために各チャンネルが合計何回選択されたかに依存します。詳しくは、[165 ページの『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。

接続アフィニティー (AFFINITY)

この属性は、同じキュー・マネージャー名を使用して複数回接続するクライアント・アプリケーションが同じクライアント・チャンネルを使用するかどうかを指定します。

この属性は、該当するチャンネル定義が複数存在する場合に使用します。

指定できる値は以下のとおりです。

PREFERRED

クライアント・チャンネル定義テーブル (CCDT) を読み取るプロセスの最初の接続によって、該当する定義のリストが作成されます。そのリストは、クライアント・チャンネル・ウェイトの値に基づいています。ウェイトの値が 0 の定義が先頭に来て、各定義がアルファベット順で並びます。プロセス内の各接続は、リスト内の最初の定義を使用して接続を試行します。接続が失敗した場合は、次の定義が使用されます。失敗した定義のうち、クライアント・チャンネル・ウェイトの値が 0 以外の定義は、リストの末尾に移動します。クライアント・チャンネル・ウェイトの値が 0 の定義は、リストの先頭に残り、各接続で最初に選択されます。

同じホスト名を持つ各クライアント・プロセスは、常に同じリストを作成します。

C、C++、または .NET プログラミング・フレームワーク (完全管理の .NET を含む) で作成されたクライアント・アプリケーション、および IBM MQ classes for Java と IBM MQ classes for JMS を使用するアプリケーションについては、リストの作成以降に CCDT が変更された場合、リストは更新されます。

この値がデフォルト値です。

NONE

CCDT を読み取るプロセス内の最初の接続が、適用可能な定義のリストを作成します。プロセス内のすべての接続が、クライアント・チャンネル・ウェイトの値に基づいて該当する定義を選択します。アルファベット順で、ウェイトの値が 0 の定義を最初に選択します。

C、C++、または .NET プログラミング・フレームワーク (完全管理の .NET を含む) で作成されたクライアント・アプリケーション、および IBM MQ classes for Java と IBM MQ classes for JMS を使用するアプリケーションについては、リストの作成以降に CCDT が変更された場合、リストは更新されます。

この属性は、クライアント接続チャンネル・タイプでのみ有効です。

接続名 (CONNAME)

この属性は、通信接続 ID です。チャンネルが使用する特定の通信リンクを指定します。

サーバー・チャンネルが起動されていない場合、サーバー・チャンネルではオプションです。その場合、接続名を指定する必要があります。

CONNAME は、記述した **TRPTYPE** のマシンの名前をコンマで区切ったリストとして指定してください。通常、必要なマシン名は 1 つだけです。複数のマシン名を指定して、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、通常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。**CLNTWGT** 属性が指定されている場合は、クライアントに対して順序が変更されます。どの接続も成功しなかった場合、チャンネルの属性によって決められたとおりに、チャンネルは再接続を試みます。クライアント・チャンネルでは、キュー・マネージャー・グループの代わりに、接続リストを使用して複数接続を構成することができます。メッセージ・チャンネルでは、複数インスタンス・キュー・マネージャーの代替アドレスへの接続を構成するために、接続リストが使用されます。

複数の接続名をリストに指定する方法は、初めて IBM WebSphere® MQ 7.0.1 でサポートされました。それによって、**CONNAME** パラメーターの構文が変更になっています。以前のクライアントおよびキュー・マネージャーは、リスト内の最初の接続名を使用して接続し、リスト内の残りの接続名は読み取りません。以前のクライアントやキュー・マネージャーが新しい構文を解析できるようにするために、リスト内の最初の接続名にポート番号を指定してください。IBM WebSphere MQ 7.0.1 より前のレベルで稼働しているクライアントまたはキュー・マネージャーからチャンネルに接続する際に、ポート番号を指定することにより問題を回避できます。

Multi マルチプラットフォームでは、クラスター受信側チャンネルの TCP/IP 接続名パラメーターはオプションです。接続名をブランクにすると、IBM MQ はデフォルト・ポートを想定し、システムの現行 IP アドレスを使用して接続名を自動的に生成します。デフォルト・ポート番号をオーバーライドしても、システムの現行 IP アドレスを引き続き使用できます。各接続名について、IP 名をブランクにして、次のように括弧で囲んだポート番号を指定してください。

(1415)

生成される **CONNAME** は常にドット 10 進 (IPv4) 形式または 16 進 (IPv6) 形式であり、英数字の DNS ホスト名の形式ではありません。

名前の最大長はプラットフォームによって異なります。

- **Multi** 264 文字。
- **z/OS** 48 文字 (注 1 を参照)。

トランスポート・タイプが TCP の場合

CONNAME は、ホスト名、またはリモート・マシン (あるいはクラスター受信側チャンネルのローカル・マシン) のネットワーク・アドレスのいずれかです。例えば、(ABC.EXAMPLE.COM)、

(2001:DB8:0:0:0:0:0:0) または (127.0.0.1)。これには、例えば (MACHINE(123)) のようなポート番号を入れることもできます。

z/OS またこれには、動的 DNS グループまたはネットワーク・ディスパッチャーの入力ポートの IP_name を入れることもできます。

IPv4 のみをサポートするネットワークで IPv6 アドレスを使用する場合、接続名は解決されません。IPv4 および IPv6 の両方を使用するネットワークでは、接続名とローカル・アドレスの相互作用によって、どちらの IP スタックを使用するかが決定されます。詳細については、123 ページの『ローカル・アドレス (LOCLADDR)』を参照してください。

トランスポート・タイプが LU 6.2 の場合

Windows **IBM i** **UNIX** TPNAME および MODENAME を指定する場合は、パートナー LU の完全修飾名を指定します。

Multi TPNAME および MODENAME がブランクの場合は、特定のプラットフォームの CPI-C サイド情報オブジェクト名を指定します。

z/OS 次の 2 とおりの形式を使用して値を指定します。

・論理装置 (LU) 名

キュー・マネージャの論理装置名。論理装置名、TP 名、およびオプション・モード名で構成されます。次の 3 通りの形式のどれかを使用して、この名前を指定します。

形式	例
luname	IGY12355
luname/TPname	IGY12345/APING
luname/TPname/modename	IGY12345/APINGD/#INTER

最初の形式を使用する場合は、TPNAME 属性と MODENAME 属性に対して、それぞれ TP 名とモード名を指定する必要があります。それ以外の形式を使用する場合は、これらの属性は必ずブランクにしてください。クライアント接続チャンネルでは、最初の形式しか使用できません。

・シンボル名

キュー・マネージャの論理装置名を表すシンボリック宛先名。この名前はサイド情報データ・セットに定義されています。TPNAME 属性と MODENAME 属性は、必ずブランクにしてください。クラスター受信側チャンネルでは、サイド情報はクラスター内の他のキュー・マネージャにあることに注意してください。この場合には、チャンネル自動定義出口による名前解決処理の結果、ローカル・キュー・マネージャに適した論理装置情報になるような名前にすることができます。

指定する LU 名または暗黙の LU 名は、VTAM 汎用リソース・グループの名前にすることができます。

伝送プロトコルが NetBIOS である場合

CONNNAME はリモート・マシンで定義された NetBIOS 名です。

伝送プロトコルが SPX である場合

CONNNAME は、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 2 バイトのソケット番号から構成される SPX 形式のアドレスです。ピリオドでネットワーク・アドレスとノード・アドレスを区切り、ソケット数を大括弧で囲み、16 進数でこれらの値を入力してください。以下に例を示します。

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

ソケット数が省略された場合、デフォルトの IBM MQ SPX ソケット数が使用されます。デフォルトは X'5E86' です。

この属性は、以下のチャンネル・タイプで有効です。

- ・送信者
- ・サーバー

- 要求側
- クライアント 接続
- クラスター送信側
- クラスター受信側

サーバー・チャンネルが起動されていない場合、サーバー・チャンネルではオプションです。その場合、接続名を指定する必要があります。

注:

1. 名前の長さについては、以下のいずれかの方法で、48 文字の長さ制限を回避できます。
 - 短いホスト名を使用するように (例えば、「myserver.location.company.com」の代わりに「myserver」というホスト名) DNS サーバーをセットアップする。
 - IP アドレスを使用する。
2. 伝送プロトコルの定義は [140 ページの『トランスポート・タイプ \(TRPTYPE\)』](#) に記載されています。

メッセージ変換 (CONVERT)

この属性は、伝送する前にメッセージを受信側のシステムが要求する形式に変換しなければならないことを指定します。

アプリケーション・メッセージ・データは通常、受信側のアプリケーションによって変換されます。しかし、リモート・キュー・マネージャーのあるプラットフォームがデータ変換をサポートしていない場合は、このチャンネル属性を使用して、伝送する **前に** メッセージを受信側のシステムが要求する形式に変換しなければならないことを指定します。

指定可能できる値は yes と no です。yes を指定すると、標準装備の形式名の 1 つが指定されているか、またはユーザー定義の形式用に変換出口が使用可能である場合に ([データ変換出口の作成を参照](#))、メッセージ内のアプリケーション・データは送信前に変換されます。no を指定すると、メッセージ中のアプリケーション・データは送信前に変換されません。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- クラスター送信側
- クラスター受信側

データ圧縮 (COMPMSG)

この属性は、チャンネルでサポートされるメッセージ・データ圧縮技法のリストです。

送信側チャンネル、サーバー・チャンネル、クラスター送信側チャンネル、クラスター受信側チャンネル、およびクライアント接続チャンネルの場合、望ましい順に値が指定されます。チャンネルのリモート・エンドでサポートされる最初の圧縮手法が使用されます。チャンネルで相互にサポートされる圧縮手法は、送信側チャンネルのメッセージ出口に渡されます。そこでは、使用される圧縮手法をメッセージごとに変更できます。圧縮により、送信および受信出口に渡されたデータが変更されます。メッセージ・ヘッダーの圧縮については、[121 ページの『ヘッダー圧縮 \(COMPHDR\)』](#)を参照してください。

指定できる値は以下のとおりです。

NONE

メッセージ・データ圧縮は実行されません。この値がデフォルト値です。

RLE

ラン・レングス・エンコードを使用してメッセージ・データ圧縮が実行されます。

ZLIBFAST

zlib 圧縮手法を使用してメッセージ・データ圧縮が実行されます。高速圧縮時間を推奨します。

ZLIBFAST はオプションで zEnterprise® データ圧縮ファシリティーにオフロードできます。詳細については、[zEDC Express 機能](#)を参照してください。

ZLIBHIGH

zlib 圧縮手法を使用してメッセージ・データ圧縮が実行されます。ハイレベル圧縮を推奨します。

ANY

チャンネルで、キュー・マネージャーがサポートするすべての圧縮手法のサポートが可能になります。サポート対象は受信側、リクエスター、サーバー接続の各チャンネルのみです。

この属性は、すべてのチャンネル・タイプで有効です。

デフォルト再接続 (DEFRECON)

クライアント接続がクライアント・アプリケーションへの接続から切断した場合に、自動的に再接続するかどうかを指定します。

可能な値は、次のとおりです。

NO

MQCONN 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

MQCONN 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

MQCONN 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

MQCONN 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。

この属性はクライアント接続チャンネルにのみ有効です。

記述 (DESCR)

この属性は、チャンネル定義を記述します。最大 64 バイトのテキストが入ります。

注：システムが 2 バイト文字セット (DBCS) を使用している場合、最大文字数が少なくなります。

テキストを別のメッセージ・キュー・マネージャーに送信する場合には、テキストが正しく変換されるように、メッセージ・キュー・マネージャー用のコード化文字セット ID (CCSID) で識別された文字セットの中の文字を使用してください。

この属性は、すべてのチャンネル・タイプで有効です。

切断インターバル (DISCINT)

この属性は、到達するメッセージがない場合にチャンネルがクローズされるまでの時間の長さを指定します。

この属性は、サーバー・チャンネル、クラスター送信側チャンネル、送信側チャンネル、およびクラスター受信側チャンネルのタイムアウト属性です (秒数で指定)。間隔は、バッチが終了した時点から、つまり、バッチ・サイズに達するか、バッチ間隔の期限が切れて伝送キューが空になった時点から測定されます。指定の時間間隔内に伝送キューに到達するメッセージがない場合には、チャンネルはクローズされます。(この時間は概数です。)

チャンネルの両端がやりとりするクローズのための制御データ交換には、クローズの理由の標識が含まれます。これにより、チャンネルの相手側が再び開始できる状態に保たれます。

0 から 999 999 までの任意の秒数を指定できます。0 は切断しない、つまり無期限の待機を表します。

TCP プロトコルを使用するサーバー接続チャンネルでは、この間隔はクライアントの非アクティブ切断の値 (秒単位) を表します。サーバー接続はパートナーのクライアントから通信を受けない状態がこの長さの時間に達すると、接続を終了します。

サーバー接続の非アクティブ間隔は、クライアントからの IBM MQ API 呼び出しの間に適用されます。

注: 待機呼び出しで長時間実行される可能性のある MQGET は非アクティブとして分類されないため、DISCINT の有効期限が切れた結果としてタイムアウトになることはありません。

この属性は、以下のチャンネル・タイプで有効です。

- 送信者
- サーバー
- サーバー接続
- クラスター送信側
- クラスター受信側

この属性は、TCP 以外のプロトコルを使用するサーバー接続チャンネルには適用されません。

注: 切断間隔に指定する値は、パフォーマンスに影響を与えます。

値を少なくすると (2, 3 秒など) 絶え間なくチャンネルを開始することになり、システム・パフォーマンスにとって好ましくありません。値を大きくすると (1 時間を超える)、システム・リソースが不必要に保留状態になりかねません。ハートビート間隔を指定することで、伝送キューにメッセージがないときには、送信側の MCA が受信側の MCA にハートビート・フローを送信し、切断間隔が経過するまで待たずに受信側の MCA がチャンネルを静止できるようにすることも可能です。これら 2 つの値を共に有効にするには、ハートビート間隔値を切断間隔値よりもかなり小さい値にする必要があります。

デフォルトの DISCINT 値は 100 分に設定されています。ただし、パフォーマンスに影響を与えたり、不必要に長時間にわたってチャンネルを実行し続けたりしないようにするには、多くの場合、この値を数分に設定するのが適切です。ご使用の環境でそうするのが適切と判断される場合は、それぞれ個別のチャンネルでこの値を変更するか、デフォルト・チャンネル定義 (例えば SYSTEM.DEF.SENDER) での値の変更によりこの値を変更できます。

詳しくは、[チャンネルの停止および静止](#)を参照してください。

処理 (QSGDISP)

この属性は、キュー共有グループでのチャンネルの処理を指定します。これは、z/OS でのみ有効です。

値は次のとおりです。

QMGR

チャンネルはコマンドを実行するキュー・マネージャーのページ・セットに定義されます。この値がデフォルトです。

GROUP

チャンネルは共有リポジトリに定義されます。この値は、共有キュー・マネージャー環境が存在する場合にのみ有効です。チャンネルを QSGDISP(GROUP) で定義すると、コマンドの DEFINE CHANNEL(name) NOREPLACE QSGDISP(COPY) が自動的に生成されて、すべてのアクティブなキュー・マネージャーに送信されます。これにより、すべてのアクティブなキュー・マネージャーが、ページ・セット 0 にローカル・コピーを作成します。アクティブでないキュー・マネージャー、または後日キュー共有グループに参加するキュー・マネージャーでは、このコマンドはそのキュー・マネージャーの開始時に生成されます。

COPY

チャンネルはコマンドを実行するキュー・マネージャーのページ・セットに定義されます。その定義は同じ名前の QSGDISP(GROUP) チャンネルからコピーされます。この値は、共有キュー・マネージャー環境が存在する場合にのみ有効です。

この属性は、すべてのチャンネル・タイプで有効です。

ヘッダー圧縮 (COMPHDR)

この属性は、チャンネルでサポートされるヘッダー・データ圧縮技法のリストです。

送信側、サーバー、クラスター送信側、クラスター受信側、およびクライアント接続の各チャンネルでは、指定された値が設定の順序に並び、チャンネルのリモート・エンドでサポートされる最初の圧縮手法が使用されます。チャンネルで相互にサポートされる圧縮手法は、送信側チャンネルのメッセージ出口に渡されます。

ここでは、使用される圧縮手法をメッセージごとに変更できます。圧縮により、送信および受信出口に渡されたデータが変更されます。

指定可能な値は以下のとおりです。

NONE

ヘッダー・データ圧縮は実行されません。この値がデフォルト値です。

SYSTEM

ヘッダー・データ圧縮が実行されます。

この属性は、すべてのチャンネル・タイプで有効です。

ハートビート間隔 (HBINT)

この属性によって、伝送キューにメッセージがなくなったときに送信メッセージ・チャンネル・エージェント (MCA) からハートビート・フローが渡される間の時間の近似値を指定することができます。

ハートビート・フローは、メッセージの到着または切断インターバルの満了を待機している受信 MCA を非ブロック化します。受信 MCA が非ブロック化された場合には、そこで切断インターバルが満了するのを待たずにチャンネルを切断することができます。ハートビート・フローは、また、大きなメッセージのために割り振られているすべてのストレージ・バッファを解放し、チャンネルの受信終了時にオープンされたままになっていたすべてのキューをクローズします。

値は秒単位であり、範囲は 0 から 999 999 でなければなりません。値 0 は、ハートビート・フローが送信されないことを意味します。デフォルト値は 300 です。最も有効にするには、この値が切断インターバル値より大幅に低いものでなければなりません。

IBM MQ classes for Java、JMS、または .NET API を使用するアプリケーションでは、HBINT の値が次のいずれかの方法で決定されます。

- アプリケーションによって使用される SVRCONN チャンネル上の値。
- アプリケーションが CCDT を使用するよう構成されている場合は、CLNTCONN チャンネル上の値。

サーバー接続チャンネルおよびクライアント接続チャンネルでは、ハートビートはサーバー・サイドおよびクライアント・サイドの両方から個々に流れることがあります。ハートビート間隔の時間内にチャンネルを通してデータが転送されないと、クライアント接続 MQI エージェントはハートビート・フローを送信し、それに対してサーバー接続 MQI エージェントは別のハートビート・フローで応答します。これは、チャンネルの状態に関係なく発生します。例えば、API の呼び出し中にチャンネルが非アクティブであっても、クライアント・ユーザーからの入力の待機中にチャンネルが非アクティブであっても関係なく発生します。サーバー接続 MQI エージェントも、同様にチャンネルの状態に関係なく、クライアントへのハートビートを開始することができます。サーバー接続 MQI エージェントとクライアント接続 MQI エージェントが同時に互いにハートビートを送信しないようにするために、サーバーのハートビートは、ハートビート間隔より 5 秒長い時間内にチャンネルを通してデータが転送されなかった場合に流れます。

IBM WebSphere MQ 7.0 より前のチャンネル・モードで動作するサーバー接続およびクライアント接続の各チャンネルでは、サーバー MCA が、WAIT オプションを指定して、クライアント・アプリケーションに代わって発行した MQGET コマンドを待機しているときにのみハートビート・フローが送信されます。

MQI チャンネルを 2 つのモードで動作させる方法については、[SharingConversations \(MQQLONG\)](#) を参照してください。

関連情報

[DEFINE CHANNEL](#)

[ALTER CHANNEL](#)

キープアライブ間隔 (KAINT)

この属性は、チャンネルのタイムアウト値を指定するために使用します。

キープアライブ間隔属性は、チャンネルのキープアライブ・タイミングを指定する通信スタックに渡される値です。このため、チャンネルごとに別々の KeepAlive 値を指定できます。

チャンネルのキープアライブ間隔 (KAINT) 属性はチャンネルごとに設定することができます。

Multi

マルチプラットフォームでは、このパラメーターへのアクセスおよび変更が可能ですが、保管されて転送されるだけです。機能的にはこのパラメーターは実装されていません。KAINТパラメーターの機能が必要な場合、122ページの『[ハートビート間隔 \(HBINT\)](#)』で説明しているとおり、Heartbeat Interval (HBINT) パラメーターを使用してください。

この属性を有効にするには、TCP/IP キープアライブを使用可能にする必要があります。z/OSでキープアライブを使用可能にするには、ALTER QMGR TCPKEEP(YES) MQSC コマンドを発行します。マルチプラットフォームでこれを行うには、KEEPALIVE=YES パラメーターを分散キューイング構成ファイル qm.ini の TCP スタンザで指定するか、IBM MQ エクスプローラーを使用して指定します。TCP プロファイル構成データ・セットを使用して、TCP/IP 自体の中でもキープアライブを使用可能にする必要があります。

この値は秒単位の時間を表し、範囲は 0 から 99999 でなければなりません。キープアライブ間隔の値 0 は、チャンネル固有のキープアライブがチャンネルに対して有効になっておらず、TCP/IP で設定されたシステム全体のキープアライブ値だけが使用されることを示しています。KAINТの値を AUTO (この値がデフォルト) に設定することもできます。KAINТを AUTO に設定すると、キープアライブ値は、以下に示すとおり、折衝ハートビート間隔 (HBINT) の値に基づいたものとなります。

表 29. 折衝された HBINT 値および対応する KAINТ 値。

この表には、2つの列があります。最初の列は折衝された HBINT 値のリストで、2番目の列にはそれぞれの折衝された HBINT に対応する KAINТ 値のリストです。

折衝された HBINT	KAINТ
>0	折衝された HBINT + 60 秒
0	0

この属性は、すべてのチャンネル・タイプで有効です。

この値は、TransportType (TRPTYPE) が TCP または SPX 以外のすべてのチャンネルでは無視されます。

ローカル・アドレス (LOCLADDR)

この属性は、チャンネルのローカル通信アドレスを指定します。

注: AMQP チャンネルは、他の IBM MQ チャンネルと同じ形式の LOCLADDR をサポートしません。詳しくは、126 ページの『[AMQP チャンネルの LOCLADDR](#)』を参照してください。

AMQP チャンネル以外のすべてのチャンネルの LOCLADDR

この属性は、トランスポート・タイプ (TRPTYPE) が TCP/IP の場合にのみ適用されます。その他のすべてのトランスポート・タイプでは、これは無視されます。

LOCLADDR 値を指定すると、停止され再始動されるチャンネルは、引き続き LOCLADDR で指定される TCP/IP アドレスを使用します。リカバリーのシナリオでは、チャンネルがファイアウォールを介して通信を行っている場合に、この属性が役に立つ可能性があります。これは、チャンネルをその接続先の TCP/IP スタックの IP アドレスを使用して再始動することによる問題がなくなるからです。LOCLADDR は、チャンネルがデュアル・スタック・システム上の IPv4 または IPv6 スタック、または単一スタック・システム上のデュアル・モード・スタックを使用するように強制することもできます。

この属性は、以下のチャンネル・タイプで有効です。

- 送信者
- サーバー
- 要求側
- クライアント 接続
- クラスター送信側
- クラスター受信側

LOCLADDR にネットワーク・アドレスが含まれている場合、そのアドレスは、チャンネルが実行されるシステム上のネットワーク・インターフェースに属するネットワーク・アドレスでなければなりません。例えば、MSQC コマンドを使用して、キュー・マネージャー ALPHA の送信側チャンネルをキュー・マネージャー BETA に定義する場合は、次のようにします。

```
DEFINE CHANNEL(TO.BETA) CHLTYPE(SDR) CONNAME(192.0.2.0) XMITQ(BETA) LOCLADDR(192.0.2.1)
```

この LOCLADDR アドレスは、IPv4 アドレス 192.0.2.1 です。この送信側チャンネルはキュー・マネージャー ALPHA のシステム上で実行されるため、IPv4 アドレスは、そのシステムのネットワーク・インターフェースのいずれかに属していなければなりません。

値は、アウトバウンド TCP/IP 通信に使用されるオプションの IP アドレスおよびオプションのポートまたはポート範囲です。この情報の形式は、次のとおりです。

```
LOCLADDR([ip-addr][(low-port[,high-port])][, [ip-addr][(low-port[,high-port])]])
```

複数のアドレスを含めた **LOCLADDR** の最大長は、MQ_LOCAL_ADDRESS_LENGTH です。

LOCLADDR を省略すると、ローカル・アドレスが自動的に割り振られます。

クライアント・チャンネル定義テーブル (CCDT) を使用する C クライアントに **LOCLADDR** を設定できることに注意してください。

どのパラメーターも指定は任意です。アドレスの ip-addr 部分を省略しておくと、IP ファイアウォール用に固定ポート番号を設定できて便利です。ポート番号を省略すると、固有のローカル・ポート番号を指定しなくても、特定のネットワーク・アダプターを選択できます。TCP/IP スタックで固有のポート番号が生成されます。

複数のローカル・アドレスを追加する場合は、それぞれについて [, [ip-addr][(low-port[,high-port])]] を指定します。複数のローカル・アドレスは、ローカル・ネットワーク・アダプターの特定のサブセットを指定する場合に使用します。複数インスタンス・キュー・マネージャー構成に含まれる別々のサーバー上にある特定のローカル・ネットワーク・アドレスを表記する場合にも、[, [ip-addr][(low-port[,high-port])]] を使用できます。

ip-addr

ip-addr は、次の 3 つの形式のいずれかで指定できます。

IPv4 ドット 10 進数

例えば、192.0.2.1 などです。

IPv6 16 進表記

例えば、2001:DB8:0:0:0:0:0:0 などです。

英数字のホスト名書式

WWW.EXAMPLE.COM など

low-port および high-port

low-port および high-port は、括弧で囲まれたポート番号です。

次の表では、**LOCLADDR** パラメーターを使用する方法を示しています。

LOCLADDR	意味
9.20.4.98	チャンネルは、ローカル側でこのアドレスにバインドします。
9.20.4.98, 9.20.4.99	チャンネルはいずれかの IP アドレスにバインドします。このアドレスは、1 つのサーバーの 2 つのネットワーク・アダプターであるかもしれず、複数インスタンス構成された 2 つの別個のサーバーの各ネットワーク・アダプターであるかもしれません。
9.20.4.98(1000)	チャンネルは、このアドレスおよびポート 1000 にローカルにバインドします。

表 30. **LOCLADDR** パラメーターを使用する方法の例 (続き)

LOCLADDR	意味
9.20.4.98(1000,2000)	チャンネルは、このアドレスにバインドし、1000 から 2000 の範囲のポートをローカル側で使用します。
(1000)	チャンネルは、ローカル側でポート 1000 にバインドします。
(1000,2000)	チャンネルは、ローカル側で 1000 から 2000 の範囲のポートにバインドします。

チャンネルが開始されると、接続名 (CONNAME) およびローカル・アドレス (LOCLADDR) に指定した値により、通信に使用される IP スタックが決定します。使用される IP スタックは、以下のように決定します。

- システムに IPv4 スタックのみ構成されている場合、常時 IPv4 スタックを使用します。ローカル・アドレス (LOCLADDR) または接続名 (CONNAME) が IPv6 ネットワーク・アドレスとして指定されていると、エラーが生成され、チャンネルの開始は失敗します。
- システムに IPv6 スタックのみ構成されている場合、常時 IPv6 スタックを使用します。ローカル・アドレス (LOCLADDR) が IPv4 ネットワーク・アドレスとして指定されている場合、エラーが生成され、チャンネルの開始は失敗します。IPv6 マップ・アドレッシングをサポートするプラットフォームでは、接続名 (CONNAME) が IPv4 ネットワーク・アドレスとして指定されていると、その IPv4 アドレスは IPv6 アドレスにマップされます。例えば、xxx.xxx.xxx.xxx が ::ffff:xxx.xxx.xxx.xxx にマップされます。マップされたアドレスを使用するには、プロトコル変換プログラムが必要となる場合があります。マップされたアドレスはできるだけ使用しないようにしてください。
- ローカル・アドレス (LOCLADDR) がチャンネルの IP アドレスとして指定されている場合、その IP アドレスのスタックが使用されます。ローカル・アドレス (LOCLADDR) が、IPv4 アドレスと IPv6 アドレスの両方に解決されるホスト名として指定されている場合、使用するスタックは、接続名 (CONNAME) によって決定されます。ローカル・アドレス (LOCLADDR) と接続名 (CONNAME) の両方が、IPv4 アドレスと IPv6 アドレスの両方に解決されるホスト名として指定されている場合、使用されるスタックはキュー・マネージャー属性 IPADDRV によって決定されます。
- システムにデュアル IPv4 および IPv6 スタックが構成されていて、チャンネルにローカル・アドレス (LOCLADDR) が指定されていない場合、チャンネルに指定された接続名 (CONNAME) によって、使用する IP スタックが決まります。接続名 (CONNAME) が、IPv4 アドレスと IPv6 アドレスの両方に解決されるホスト名として指定されている場合、使用されるスタックはキュー・マネージャー属性 IPADDRV によって決定されます。

Multi マルチプラットフォームでは、ローカル・アドレスが定義されていないすべての送信側チャンネルで使用される、デフォルトのローカル・アドレス値を設定することができます。このデフォルト値を定義するには、キュー・マネージャーの開始前に MQ_LCLADDR 環境変数を設定します。この値の形式は、MQSC 属性の LOCLADDR の形式と同じです。

クラスター送信側チャンネルのローカル・アドレス

クラスター送信側チャンネルは、常に、宛先キュー・マネージャーに定義された対応するクラスター受信側チャンネルの構成を継承します。これは、同じ名前のクラスター送信側チャンネルがローカルに定義されている場合でも行われます。その場合、手動定義は初期通信にしか使用されません。

したがって、クラスター受信側チャンネルに定義された LOCLADDR に依存することはできません。この LOCLADDR の IP アドレスは、クラスター送信側が作成されるシステムには存在しない可能性があるからです。そのため、クラスター受信側で LOCLADDR は使用すべきではありませんが、想定されるすべてのクラスター送信側の IP アドレスではなくポートのみを制限する理由があり、クラスター送信側チャンネルが作成される可能性のあるすべてのシステムでそれらのポートが使用可能であることが既知である場合は例外です。

クラスターで LOCLADDR を使用してアウトバウンド通信チャンネルを特定の IP アドレスにバインドする必要がある場合は、チャンネル自動定義出口を使用するか、または可能であればキュー・マネージャーのデフォルトの LOCLADDR を使用してください。チャンネル出口を使用すると、出口の LOCLADDR 値が、自動的に定義される CLUSSDR チャンネルに強制的に適用されます。

出口を使用してクラスター送信側チャンネルでデフォルト以外の LOCLADDR を使用する場合も、デフォルト値を使用する場合も、手動で定義された一致するクラスター送信側チャンネルがある場合は (完全リポジトリのキュー・マネージャーに対するチャンネルなど)、そのチャンネルによる初期通信を可能にするためにそのチャンネルにも LOCLADDR 値が設定されている必要があります。

注: LOCLADDR で指定されているポート (あるいは、ポート範囲が指定されている場合はすべてのポート) に関してオペレーティング・システムからバインド・エラーが返された場合は、チャンネルが開始されず、システムはエラー・メッセージを出します。

AMQP チャンネルの LOCLADDR

AMQP チャンネルは、他の IBM MQ チャンネルとは異なる形式の LOCLADDR をサポートします。

LOCLADDR (*ip-addr*)

LOCLADDR は、チャンネルのローカル通信アドレスです。このパラメーターは、特定の IP アドレスの使用をクライアントに強制する必要がある場合に使用します。LOCLADDR は、チャンネルで IPv4 または IPv6 アドレスを使用したり (選択可能な場合)、複数のネットワーク・アダプターがあるシステムにおいて特定のネットワーク・アダプターを使用したりすることを強制する場合に役立ちます。

LOCLADDR の最大長は MQ_LOCAL_ADDRESS_LENGTH です。

LOCLADDR を省略すると、ローカル・アドレスが自動的に割り振られます。

ip-addr

ip-addr は、単一のネットワーク・アドレスであり、次の 3 つの形式のいずれかで指定します。

IPv4 ドット 10 進数

192.0.2.1 など

IPv6 16 進表記

2001:DB8:0:0:0:0:0:0 など

英数字のホスト名書式

WWW.EXAMPLE.COM など

IP アドレスを入力すると、アドレス・フォーマットのみが妥当性検査されます。IP アドレス自体は妥当性検査されません。

関連情報

[自動定義クラスター送信側チャンネルの処理](#)

長期再試行カウント (LONGRTY)

この属性は、チャンネルがそのパートナーへのセッションの割り振りを試行する最大回数を指定します。

long retry count 属性に設定できる値は 0 から 999 999 999 です。

この属性は、以下のチャンネル・タイプで有効です。

- 送信者
- サーバー
- クラスター送信側
- クラスター受信側

最初の割り振り試行が失敗すると、*short retry count* の数値が減分され、チャンネルは残りの回数だけ再試行します。それでも失敗する場合は、再試行の間隔を *long retry interval* にして *long retry count* 回再試行します。それでも割り振りが成功しない場合には、チャンネルは停止します。このチャンネルは、後で、コマンドを使用して再始動する必要があります。チャンネル・イニシエーターによって自動的に開始されることはありません。

z/OS では、チャンネルの最大数 (MAXCHL) を超えると、チャンネルが再試行を開始できなくなります。

IBM i、UNIX、および Windows システムで、再試行を行うためには、チャンネル・イニシエーターが実行されている必要があります。チャンネル・イニシエーターは、チャンネルが使用している伝送キューの定義に指定されている開始キューをモニターしている必要があります。

チャンネル・イニシエーター (z/OS) またはチャンネル (マルチプラットフォーム) がチャンネルの再試行中に停止すると、短期再試行カウント および長期再試行カウント は、チャンネル・イニシエーターまたはチャンネルが再始動される時、あるいはメッセージが送信側チャンネルに正常に送信されたときにリセットされます。ただし、チャンネル・イニシエーター (z/OS) またはキュー・マネージャー (Multiplatforms) がシャットダウンおよび再始動されると、短期再試行カウント および長期再試行カウント はリセットされません。チャンネルは、キュー・マネージャーの再始動またはメッセージの送信の前に保持していた再試行カウント値を保持します。

IBM i、UNIX、および Windows システムの場合:

1. チャンネルが RETRYING 状態から RUNNING 状態になる際、短期再試行カウント および長期再試行カウント はすぐにはリセットされません。チャンネルが RUNNING 状態になったあとの最初のメッセージがチャンネルを通して正常に流れたときにだけリセットされます。つまり、ローカルのチャンネルが、相手側に送信したメッセージの数を確認した時点でリセットされます。
2. 短期再試行カウント および長期再試行カウント は、チャンネルが再始動されるとリセットされます。

長期再試行間隔 (LONGTMR)

この属性は、長期再試行モードのときに、チャンネルが接続の確立を再試行するまでに待機するおよその間隔 (秒単位) です。

チャンネルがアクティブになるのを待つ必要がある場合、再試行間隔が延長されることがあります。

チャンネルは、短期再試行間隔で短期再試行カウント 回数を試みた後、この長期間隔で長期再試行カウントの回数、接続を試みます。

この属性に設定できる値は 0 から 999 999 です。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- クラスター送信側
- クラスター受信側

LU 6.2 モード名 (MODENAME)

この属性は、LU 6.2 接続に使用します。これは、通信セッションの割り振りが実行されるときに、接続のセッションの特性について追加の定義を提供します。

サイド情報を SNA 通信で使用するときは、モード名は CPI-C 通信サイド・オブジェクトか APPC サイド情報に定義し、この属性はブランクのままにしておく必要があります。そうでない場合は、SNA モード名に設定する必要があります。

名前は 1 から 8 文字の英数字でなければなりません。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- 要求側
- クライアント接続
- クラスター送信側
- クラスター受信側

これは受信側またはサーバー接続チャンネルに対しては無効です。

LU 6.2 トランザクション・プログラム名 (TPNAME)

この属性は、LU 6.2 接続に使用します。これは、リンクの末端で実行するトランザクション・プログラム (MCA) の名前または総称名です。

サイド情報を SNA 通信で使用するときは、トランザクション・プログラム名は CPI-C 通信サイド・オブジェクトか APPC サイド情報に定義し、この属性はブランクのままにしておく必要があります。そうしないと、送信側チャンネルおよびリクエスター・チャンネルでこの名前が必要になります。

この名前は最大 64 文字まで許容されます。

CONNNAME にサイド・オブジェクト名が指定されていない場合 (この場合ブランクに設定される)、この名前は SNA トランザクション・プログラム名に設定されます。その代わりに、実際の名前は CPI-C コミュニケーション・サイド・オブジェクト、つまり APPC サイド情報データ・セットから取得されます。

この情報は、プラットフォームごとに異なる方法で設定されます。ご使用のプラットフォームにおける通信の設定方法については、[分散キューイングの構成](#)を参照してください。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- 要求側
- クライアント接続
- クラスター送信側
- クラスター受信側

最大インスタンス数 (MAXINST)

この属性は、サーバー接続チャンネルまたは AMQP チャンネルの、開始可能な同時インスタンスの最大数を指定します。

それぞれのチャンネル・タイプでの属性の使用方法については、[子トピック](#)を参照してください。

関連情報

[サーバー接続チャンネルの制限](#)

[DEFINE CHANNEL](#)

サーバー接続チャンネル接続の最大インスタンス数

この属性は、1つのサーバー接続チャンネルで開始可能な同時インスタンスの最大数を指定します。

この属性に設定できる値は 0 から 999 999 999 です。ゼロの値は、このチャンネルでクライアント接続が許可されないことを示します。デフォルト値は 999 999 999 です。

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、実行中のチャンネルは影響を受けません。しかし、新規インスタンスは、十分な数の既存のインスタンスが実行を終了するまで開始できません。

V 9.0.0 AMQP チャンネル接続の最大インスタンス数

この属性は、1つの AMQP チャンネルで開始可能な同時インスタンスの最大数を指定します。

この属性に設定できる値は 0 から 999 999 999 です。ゼロの値は、このチャンネルでクライアント接続が許可されないことを示します。デフォルト値は 999 999 999 です。

クライアントが接続を試行した場合、接続済みクライアント数が MAXINST に達していると、チャンネルはクローズ・フレームを伴って接続を閉じます。クローズ・フレームには以下のメッセージが含まれます。

```
amqp:resource-limit-exceeded
```

既に接続済みの ID を使ってクライアントが接続した場合 (つまりクライアント・テークオーバーを実行した場合)、接続済みクライアント数が MAXINST に達しているかどうかにかかわらず、テークオーバーが成功します。

クライアントあたりの最大インスタンス数 (MAXINSTC)

この属性は、サーバー接続チャンネルの、単一クライアントから開始可能な同時インスタンスの最大数を指定します。

この属性に設定できる値は 0 から 999 999 999 です。ゼロの値は、このチャンネルでクライアント接続が許可されないことを示します。デフォルト値は 999 999 999 です。

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、実行中のチャンネルは影響を受けません。しかし、それらのクライアントからの新規インスタンスは、十分な数の既存のインスタンスが実行を終了するまで開始できません。

この属性は、サーバー接続チャンネルにのみ有効です。

関連情報

[サーバー接続チャンネルの制限](#)

[DEFINE CHANNEL](#)

最大メッセージ長 (MAXMSGL)

この属性は、チャンネルで伝送可能な最大メッセージ長を指定します。

Multi IBM MQ for IBM i、UNIX、および Windows システムでは、0 からキュー・マネージャーの最大メッセージ長までの範囲の値を指定します。詳細は、[ALTER QMGR](#) で説明されている ALTER QMGR コマンドの MAXMSGL パラメーターを参照してください。

z/OS IBM MQ for z/OS では、ゼロ以上 104 857 600 バイト (100 MB) 以下の値を指定してください。

プラットフォームごとに IBM MQ システムの実装環境は異なるため、一部のアプリケーションでメッセージ処理に使用可能なサイズが制限されることがあります。その数値には、システムが滞りなく処理できるサイズが反映されます。チャンネルが開始すると、チャンネルの両端で指定された 2 つの数値のうち小さい方の値が採用されます。

メッセージにデジタル署名と鍵を追加することで、[Advanced Message Security](#) ではメッセージの長さが増します。

注:

1. 最大メッセージ・サイズを 0 にすることもできますが、この値が採用されると、最大メッセージ・サイズはローカル・キュー・マネージャーの最大値に設定されます。

この属性は、すべてのチャンネル・タイプで有効です。

メッセージ・チャンネル・エージェント名 (MCANAME)

この属性は予約属性です。指定する場合、設定できるのはブランクのみです。

最大長は 20 文字です。

メッセージ・チャンネル・エージェント・タイプ (MCATYPE)

この属性では、メッセージ・チャンネル・エージェントに *process* または *thread* を指定することができます。

IBM MQ for z/OS では、この属性がサポートされているのはチャンネル・タイプがクラスター受信側であるチャンネルだけです。

プロセスとして実行することには、次のような利点があります。

- 各チャンネルが分離しているため、安全性が向上する
- チャンネルごとに固有のジョブ権限を指定できる

- ジョブ・スケジュールを制御できる

スレッドには、次のような利点があります。

- 使用するストレージが少なく済む
- コマンド行からの入力で簡単に構成できる
- 実行速度の向上 - プロセスを開始するようオペレーティング・システムに指示するよりも、スレッドを開始する方が早い

送信側、サーバー、要求側のチャンネル・タイプの場合のデフォルトは `process` です。クラスター送信側およびクラスター受信側のチャンネル・タイプの場合のデフォルトは `thread` です。これらのデフォルトはインストール時に変更される可能性があります。

チャンネル定義に `process` を指定すると、`RUNMQCHL` プロセスが開始します。 `thread` を指定すると、`AMQRMPPA` プロセスのスレッド上、または `MQNOREMPOOL` が指定されている場合は `RUNMQCHI` プロセスのスレッド上で `MCA` が実行されます。インバウンド割り振りを受信するマシン上で、`RUNMQLSR` を使用する場合 `MCA` はスレッドとして実行されます。 `inetd` を使用する場合は、プロセスとして実行されません。

IBM MQ for z/OS では、この属性がサポートされているのはチャンネル・タイプがクラスター受信側であるチャンネルだけです。他のプラットフォームでは、この属性は以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- 要求側
- クラスター送信側
- クラスター受信側

メッセージ・チャンネル・エージェント・ユーザー ID (MCAUSER)

この属性は、`MCA` が IBM MQ リソースへのアクセスの許可に使用するユーザー ID (ストリング) です。

注: チャンネルの実行に使用するユーザー ID を提供するための代替手段としては、チャンネル認証の記録を使用するという方法があります。チャンネル認証レコードを使用すると、複数の異なる接続で、それぞれ異なる資格情報を使用して、同一のチャンネルを使用することができます。チャンネルで `MCAUSER` が設定されており、かつチャンネル認証レコードが同じチャンネルに適用されている場合、チャンネル認証レコードが優先されます。チャンネル定義での `MCAUSER` は、チャンネル認証レコードが `USERSRC(CHANNEL)` を使用する場合にのみ使用されます。

この許可には、(PUT 権限が DEF である場合) 受信側チャンネルまたは要求側チャンネルの宛先キューへのメッセージ書き込みが含まれます。

IBM MQ for Windows の場合、ユーザー ID をドメイン修飾して `user@domain` の形式で表すことができます。この `domain` は、ローカル・システムの Windows システム・ドメインか、またはトラステッド・ドメインのどちらかでなければなりません。

この属性をブランクにした場合、`MCA` はデフォルトのユーザー ID を使用します。詳しくは、[DEFINE CHANNEL](#) を参照してください。

この属性は、以下のチャンネル・タイプで有効です。

- 受信側
- 要求側
- サーバー接続
- クラスター受信側

関連情報

[チャンネル認証レコード](#)

メッセージ出口名 (MSGEXIT)

この属性は、チャンネル・メッセージ出口によって実行されるユーザー出口プログラムの名前を指定します。

この属性を、順番に実行されるプログラムの名前のリストにすることができます。有効なチャンネル・メッセージ出口がない場合は、ブランクにしておきます。

この属性の形式および最大長は、[135 ページの『受信出口名 \(RCVEXIT\)』](#)で説明するように、プラットフォームによって異なります。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- 受信側
- 要求側
- クラスター送信側
- クラスター受信側

メッセージ出口ユーザー・データ (MSGDATA)

この属性は、チャンネル・メッセージ出口に渡されるユーザー・データを指定します。

一連のメッセージ出口を実行できます。ユーザー・データ長の制限および複数の出口に対する MSGDATA の指定方法は、RCVDATA について示されているものと同じです。[136 ページの『受信出口ユーザー・データ \(RCVDATA\)』](#)を参照してください。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- 受信側
- 要求側
- クラスター送信側
- クラスター受信側

メッセージ再試行出口名 (MREXIT)

この属性は、メッセージ再試行ユーザー出口によって実行されるユーザー出口プログラムの名前を指定します。

有効なメッセージ再試行出口プログラムがない場合は、ブランクにしておきます。

名前の形式および最大長は、[135 ページの『受信出口名 \(RCVEXIT\)』](#)で説明するように、プラットフォームによって異なります。ただし、指定できるメッセージ再試行出口は1つだけです。

この属性は、以下のチャンネル・タイプで有効です。

- 受信側
- 要求側
- クラスター受信側

メッセージ再試行出口ユーザー・データ (MRDATA)

この属性は、チャンネル・メッセージ再試行出口が呼び出された場合、その出口に引き渡されるデータを指定します。

この属性は、以下のチャンネル・タイプで有効です。

- 受信側
- 要求側

- クラスター受信側

メッセージ再試行カウント (MRRTY)

この属性は、チャンネルがメッセージを再配信しようとする回数を指定します。

この属性は、メッセージ再試行出口名がブランクの場合にのみ MCA の処置を制御します。出口名がブランクではないときは、MRRTY の値は出口に引き渡されます。ただし、再試行される場合、その回数はこの属性ではなく、出口が制御します。

値の範囲は 0 から 999 999 999 でなければなりません。値 0 は、追加の再試行が行われないことを意味します。デフォルトは 10 です。

この属性は、以下のチャンネル・タイプで有効です。

- 受信側
- 要求側
- クラスター受信側

メッセージ再試行間隔 (MRTMR)

この属性は、チャンネルが MQPUT 操作を再試行できようになるまでに経過する必要がある最短の時間間隔を指定します。

この時間間隔は、ミリ秒単位です。

この属性は、メッセージ再試行出口名がブランクの場合にのみ MCA の処置を制御します。出口名がブランクではない場合、MRTMR の値は、その出口に渡され、使用されます。ただし、再試行間隔は、この属性ではなく出口が制御します。

値の範囲は 0 から 999 999 999 でなければなりません。値 0 は、再試行が可能になるとただちに再試行されることを意味します (ただし、MRRTY の値が 0 より大きい場合)。デフォルトは 1000 です。

この属性は、以下のチャンネル・タイプで有効です。

- 受信側
- 要求側
- クラスター受信側

モニター (MONCHL)

この属性は、オンライン・モニター・データの収集を制御します。

指定可能な値は以下のとおりです。

QMGR

オンライン・モニター・データの収集は、キュー・マネージャー・オブジェクトの MONCHL 属性の設定から継承されます。この値がデフォルト値です。

OFF

このチャンネルのオンライン・モニター・データ収集は無効になります。

LOW

パフォーマンスへの影響が最も少ない低いデータ収集率。ただし、表示されるモニター結果は最新になるとは限りません。

MEDIUM

システムのパフォーマンスへの影響が制限される適度なデータ収集率。

HIGH

パフォーマンスに影響する高いデータ収集率。ただし、表示されるモニター結果は最新のものです。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー

- 受信側
- 要求側
- サーバー接続
- クラスター送信側
- クラスター受信側

モニター・データの詳細については、[キューおよびチャンネルのモニター・データの表示を参照してください](#)。

NETPRTY チャンネル属性

NETPRTY チャンネル属性では、CLUSRCVR チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

NETPRTY 属性は、あるネットワークを主ネットワークにし、別のネットワークをバックアップ・ネットワークにする場合に使用します。同じランクの 1 組みのチャンネルが存在すると仮定すると、複数のパスが使用可能な場合、クラスター化では最高の優先順位を持つパスを選択します。

NETPRTY チャンネル属性を使用する典型的な例としては、異なるコストまたは速度を持ち、また同じ複数の宛先を接続する 2 つのネットワークを区別するケースです。

注: クラスター受信側チャンネルのターゲット・キュー・マネージャーで、この属性を指定します。一致するクラスター送信側チャンネルで指定したすべてのバランシングは、無視される可能性が高くなります。[クラスター・チャンネルを参照してください](#)。

非持続メッセージ速度 (NPMSPEED)

この属性は、非持続メッセージが送信される速度を指定します。

指定可能な値は以下のとおりです。

NORMAL

チャンネル上の非持続メッセージはトランザクションの中で転送されます。

FAST

チャンネル上の非持続メッセージは、トランザクションの中では転送されません。

デフォルトは FAST です。この方法には、同期点に達するまで待機しないため、非持続メッセージをより迅速に取り出せるという利点があります。欠点は、トランザクションの一部ではないため、伝送障害があった場合やメッセージの転送中にチャンネルが停止した場合にデータが失われることがある点です。[メッセージの安全性を参照してください](#)。

注:

1. IBM MQ for z/OS のアクティブ・リカバリー・ログの切り替えおよびアーカイブの頻度が予想より多い場合は、チャンネルを介して送信されるのが非持続メッセージであるのならば、チャンネルの送信側と受信側の両方で NPMSPEED(FAST) を設定して SYSTEM.CHANNEL.SYNCQ の更新を最小限にすることができます。
2. SYSTEM.CHANNEL.SYNCQ への更新に関連して高い CPU 使用率が見られる場合は、NPMSPEED(FAST) を設定して CPU 使用率を大幅に下げることができます。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- 受信側
- 要求側
- クラスター送信側
- クラスター受信側

パスワード (PASSWORD)

この属性は、リモート MCA とのセキュア LU 6.2 セッションの開始を試行するときに MCA が使用できるパスワードを指定します。

最大 12 文字の長さのパスワードを指定できます。ただし、最初の 10 文字だけが使用されます。

これは、送信側、サーバー、要求側、またはクライアント接続のチャンネル・タイプの場合に有効です。

IBM MQ for z/OS では、この属性はクライアント接続チャンネルでのみ有効です。他のプラットフォームでは、この属性は以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- 要求側
- クライアント接続
- クラスター送信側

ポート番号 (PORT)

AMQP クライアントへの接続に使用するポート番号を指定します。

AMQP 1.0 接続のデフォルト・ポートは 5672 です。ポート 5672 を既に使用している場合は、異なるポートを指定できます。

PUT 権限 (PUTAUT)

この属性を使用して、MCA で実行されるセキュリティー処理のタイプを指定します。

この属性は、以下のチャンネル・タイプで有効です。

- 受信側
- 要求側
- サーバー接続 (z/OS のみ)
- クラスター受信側

この属性を使用して、以下の実行時に MCA で行われるセキュリティー処理の種類を選びます。

- 宛先キューへの MQPUT コマンド (メッセージ・チャンネルの場合)、または
- MQI 呼び出し (MQI チャンネルの場合)

z/OS z/OS では、検査されるユーザー ID と検査されるユーザー ID の数は、MQADMIN RACF® クラス hlq.RESLEVEL プロファイルの設定により異なります。hlq.RESLEVEL に対してチャンネル・イニシエーターのユーザー ID が持つアクセスのレベルに応じて、0、1、または 2 個のユーザー ID が検査されます。検査されるユーザー ID の数については、[RESLEVEL およびチャンネル・イニシエーター接続を参照してください](#)。どのユーザー ID が検査されるかについては詳しくは、[チャンネル・イニシエーターで使用されるユーザー ID を参照してください](#)。

以下のいずれかを選択することができます。

プロセス・セキュリティー。デフォルトの権限 (DEF) とも呼ばれます。

デフォルトのユーザー ID が使用されます。

z/OS 以外のプラットフォームでは、キューに対するオープン権限をチェックするために使用されるユーザー ID は、メッセージ・チャンネルの受信側で MCA を実行しているプロセスまたはユーザーのユーザー ID です。

z/OS では、チェックされるユーザー ID の数に応じて、ネットワークから受信したユーザー ID と、[MCAUSER](#) から得られたユーザー ID の両方が使用される場合があります。

キューはこのユーザー ID とオープン・オプション MQOO_SET_ALL_CONTEXT でオープンされます。

コンテキスト・セキュリティ (CTX)

メッセージに関連付けられたコンテキスト情報からのユーザー ID が、代替ユーザー ID として使用されます。

メッセージ記述子内の *UserIdentifier* はオブジェクト記述子内の *AlternateUserId* フィールドへ転送されます。キューは、オープン・オプション MQOO_SET_ALL_CONTEXT と MQOO_ALTERNATE_USER_AUTHORITY でオープンされます。

z/OS 以外のプラットフォームでは、MQOO_SET_ALL_CONTEXT および MQOO_ALTERNATE_USER_AUTHORITY の際にキューに対するオープン権限をチェックするために使用されるユーザー ID は、メッセージ・チャンネルの受信側で MCA を実行しているプロセスまたはユーザーのユーザー ID です。MQOO_OUTPUT のキュー上のオープン権限をチェックするのに使用するユーザー ID は、メッセージ記述子の *UserIdentifier* です。

z/OS では、チェックされるユーザー ID の数に応じて、ネットワークから受信したユーザー ID、または MCAUSER から得られたユーザー ID だけでなく、メッセージ記述子内のコンテキスト情報からのユーザー ID も使用される場合があります。

コンテキスト・セキュリティ (CTX) はサーバー接続チャンネル上ではサポートされていません。

MCA セキュリティのみ (ONLYMCA)

MCAUSER から得られたユーザー ID が使用されます。

キューはオープン・オプション MQOO_SET_ALL_CONTEXT でオープンされます。

この値は z/OS にのみ適用されます。

代替 MCA セキュリティ (ALTMCA)

チェックされるユーザー ID の数に応じて、MCAUSER から得られたユーザー ID だけでなく、メッセージ記述子内のコンテキスト情報 (*UserIdentifier* フィールド) からのユーザー ID も使用される場合があります。

この値は z/OS にのみ適用されます。

コンテキスト・フィールドとオープン・オプションの詳細については、[コンテキスト情報の制御](#)を参照してください。

セキュリティの詳細については、以下を参照してください。

- [セキュリティ](#)
-  [UNIX, Linux, and Windows でのセキュリティのセットアップ](#)
-  [IBM i でのセキュリティのセットアップ](#)
-  [z/OS でのセキュリティのセットアップ](#)

キュー・マネージャー名 (QMNAME)

この属性は、IBM MQ MQI client・アプリケーションが接続を要求できるキュー・マネージャーまたはキュー・マネージャー・グループの名前を指定します。

この属性は、以下のチャンネル・タイプで有効です。

- クライアント接続

受信出口名 (RCVEXIT)

この属性は、チャンネル受信ユーザー出口で実行されるユーザー出口プログラムの名前を指定します。

この属性を、順番に実行されるプログラムの名前のリストにすることができます。有効なチャンネル受信ユーザー出口がない場合には、ブランクのままにしておいてください。

この属性の形式および最大長はプラットフォームによって異なります。

- z/OS では、これはロード・モジュール名で、最大長が 8 文字になります。例外として、クライアント接続チャンネルは最大長が 128 文字です。
- IBM i では、次の形式です。

```
libname/progname
```

これは CL コマンドに指定するときの形式です。

IBM MQ コマンド (MQSC) に指定するときは、次の形式になります。

```
progname libname
```

ここで、*progname* は最初の 10 文字を使用し、*libname* はその次の 10 文字を使用します (いずれも必要に応じて右側にブランクを埋め込みます)。ストリングの最大長は 20 文字です。

- Windows では、次の形式です。

```
dllname(functionname)
```

この *dllname* は、接尾部 .DLL を付けずに指定します。ストリングの最大長は 40 文字です。

- UNIX では、次の形式です。

```
libraryname(functionname)
```

ストリングの最大長は 40 文字です。

z/OS でのクラスター送信側のチャンネル自動定義中に、チャンネル出口名は z/OS の形式に変換されます。出口名の変換方法を制御したい場合は、チャンネル自動定義出口を作成できます。詳細については、[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。

受信、送信、またはメッセージ出口プログラム名のリストを指定できます。名前は、コンマ、スペース、またはその両方で分離する必要があります。以下に例を示します。

```
RCVEXIT(exit1 exit2)
MSGEXIT(exit1,exit2)
SENDEXIT(exit1, exit2)
```

特定タイプの出口のユーザー・データ出口名および出口ストリングのストリング全長は、500 文字に制限されています。IBM MQ for IBM i では、最大で 10 個までの出口名を指定できます。IBM MQ for z/OS では、最大で 8 個までの出口名を指定できます。

この属性は、すべてのチャンネル・タイプで有効です。

受信出口ユーザー・データ (RCVDATA)

この属性は、受信出口に渡されるユーザー・データを指定します。

一連の受信出口を実行できます。一連の出口に対するユーザー・データのストリングは、コンマ、スペース、またはその両方で分離する必要があります。以下に例を示します。

```
RCVDATA(exit1_data exit2_data)
MSGDATA(exit1_data,exit2_data)
SENDDATA(exit1_data, exit2_data)
```

IBM MQ for UNIX システム、および Windows システムでは、出口名のストリングとユーザー・データのストリングの長さは 500 文字に制限されています。IBM MQ for IBM i では、最大 10 個の出口名を指定でき、それぞれのユーザー・データの長さは、32 文字に制限されています。IBM MQ for z/OS では、それぞれの長さが 32 文字のユーザー・データのストリングを、最大 8 個まで指定できます。

この属性は、すべてのチャンネル・タイプで有効です。

セキュリティー出口名 (SCYEXIT)

この属性は、チャンネル・セキュリティー出口によって実行される出口プログラムの名前を指定します。

有効なチャンネル・セキュリティー出口がない場合は、これをブランクのままにします。

名前の形式および最大長は、[135 ページ](#)の『[受信出口名 \(RCVEXIT\)](#)』で説明するように、プラットフォームによって異なります。ただし、指定できるセキュリティー出口は1つだけです。

この属性は、すべてのチャンネル・タイプで有効です。

セキュリティー出口ユーザー・データ (SCYDATA)

この属性は、セキュリティー出口に渡されるユーザー・データを指定します。

最大長は 32 文字です。

この属性は、すべてのチャンネル・タイプで有効です。

送信出口名 (SENDEXIT)

この属性は、チャンネル送信出口によって実行される出口プログラムの名前を指定します。

この属性として、順次実行されるプログラムの名前のリストを指定できます。有効なチャンネル送信出口がない場合には、ブランクのままにしておいてください。

この属性の形式および最大長は、[135 ページ](#)の『[受信出口名 \(RCVEXIT\)](#)』で説明するように、プラットフォームによって異なります。

この属性は、すべてのチャンネル・タイプで有効です。

送信出口ユーザー・データ (SENDDATA)

この属性は、送信出口に渡されるユーザー・データを指定します。

一連の送信出口を実行できます。ユーザー・データ長の制限および複数の出口に対する SENDDATA の指定方法は、RCVDATA について示されているものと同じです。[136 ページ](#)の『[受信出口ユーザー・データ \(RCVDATA\)](#)』を参照してください。

この属性は、すべてのチャンネル・タイプで有効です。

シーケンス番号の折り返し (SEQWRAP)

この属性は、メッセージ・シーケンス番号が 1 で再始動する前に到達する最大数を指定します。

前のメッセージがこの数値を使用している間に数値が再発行されることを防ぐため、この数値には十分大きな値を指定する必要があります。チャンネルの両端は、チャンネルの開始時に同じシーケンス番号のラップ値を保持している必要があります。そうでない場合は、エラーが発生します。

値は、100 から 999 999 999 の間で設定できます。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- 受信側
- 要求側
- クラスター送信側
- クラスター受信側

短期再試行カウント (SHORTRTY)

この属性は、チャンネルがそのパートナーへのセッションの割り振りを試行する最大回数を指定します。

short retry count 属性に設定できる値はゼロから 999 999 999 までです。

この属性は、以下のチャンネル・タイプで有効です。

- 送信者
- サーバー
- クラスター送信側
- クラスター受信側

最初の割り振り試行が失敗すると、*short retry count* は減分され、チャンネルは、**short retry interval** 属性で定義された間隔で、各試行の間に残りの回数を再試行します。それでも失敗する場合は、各試行の間隔を *long retry interval* にして *long retry count* 回再試行します。それでも割り振りが成功しない場合には、チャンネルは停止します。

z/OS では、チャンネルの最大数 (**MAXCHL**) を超えると、チャンネルが再試行を開始できなくなります。

IBM i、UNIX、および Windows システムで、再試行を行うためには、チャンネル・イニシエーターが実行されている必要があります。チャンネル・イニシエーターは、チャンネルが使用している伝送キューの定義に指定されている開始キューをモニターしている必要があります。

チャンネル・イニシエーター (z/OS) またはチャンネル (マルチプラットフォーム) がチャンネルの再試行中に停止すると、短期再試行カウント および長期再試行カウント は、チャンネル・イニシエーターまたはチャンネルが再始動されるとき、あるいはメッセージが送信側チャンネルに正常に送信されたときにリセットされます。ただし、チャンネル・イニシエーター (z/OS) またはキュー・マネージャー (Multiplatforms) がシャットダウンおよび再始動されると、短期再試行カウント および長期再試行カウント はリセットされません。チャンネルは、キュー・マネージャーの再始動またはメッセージの送信の前に保持していた再試行カウント値を保持します。

IBM i、UNIX、および Windows システムの場合:

1. チャンネルが **RETRYING** 状態から **RUNNING** 状態になる際、短期再試行カウント および長期再試行カウント はすぐにはリセットされません。チャンネルが **RUNNING** 状態になったあとの最初のメッセージがチャンネルを通して正常に流れたときにだけリセットされます。つまり、ローカルのチャンネルが、相手側に送信したメッセージの数を確認した時点でリセットされます。
2. 短期再試行カウント および長期再試行カウント は、チャンネルが再始動されるとリセットされます。

短期再試行間隔 (SHORTTMR)

この属性は、短期間の再試行モード中に、接続の確立を再試行するまでチャンネルが待機する間隔の概数を秒単位で指定します。

チャンネルがアクティブになるのを待機する必要がある場合、再試行間隔が延長されることがあります。

この属性に設定できる値は 0 から 999 999 です。

この属性は、以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- クラスター送信側
- クラスター受信側

SSL 暗号仕様 (SSLCIPH)

この属性は TLS 接続に対して単一の CipherSpec を指定します。

各 IBM MQ チャンネル定義には SSLCIPH 属性が含まれています。この値は、最大長が 32 文字のストリングです。

次の事項に注意してください。

- SSLCIPH 属性は空白値にすることができます。これは、TLS を使用しないという意味です。チャンネルの片方の SSLCIPH 属性が空白値の場合は、もう片方の SSLCIPH 属性も空白値である必要があります。

- また、SSLCIPH に非空白値が含まれている場合、チャンネルは指定された暗号を使用して TLS を利用しようと試みます。この場合も、チャンネルの両端で同じ SSLCIPH 値を指定する必要があります。
- SSLCIPH はチャンネルの両端で同じでなければならないという規則の唯一の例外として、完全管理の .NET クライアントでは、特殊値 ***NEGOTIATE** を指定することができます。このオプションにより、チャンネルは .NET フレームワークでサポートされている最新バージョンのプロトコルを選択し、サーバーでサポートされる CipherSpec とネゴシエーションすることができます。

これは、トランスポート・タイプ (TRPTYPE) が TCP のチャンネルにのみ有効です。TRPTYPE が TCP でない場合、データは無視され、エラー・メッセージは発行されません。

SSLCIPH の詳細については、『[DEFINE CHANNEL](#)』および『[CipherSpec の指定](#)』を参照してください。

SSL クライアント認証 (SSLCAUTH)

この属性は、チャンネルが TLS クライアントから TLS 証明書を受信して認証する必要があるかどうかを指定します。

指定可能な値は以下のとおりです。

オプション

ピア TLS クライアントが証明書を送信する場合、証明書は通常どおりに処理されますが、証明書が送信されなくても認証は失敗しません。

REQUIRED

TLS クライアントが証明書を送信しない場合、認証は失敗します。

デフォルト値は REQUIRED です。

非 TLS チャンネル定義で SSLCAUTH の値を指定できます。つまり、SSLCIPH 属性が欠落しているか空白になっているチャンネル定義です。

SSLCAUTH はオプションの属性です。

この属性は、チャンネル開始フローを受信できるすべてのチャンネル・タイプに有効です (送信側チャンネルは除く)。

この属性は、以下のチャンネル・タイプで有効です。

- サーバー
- 受信側
- 要求側
- サーバー接続
- クラスタ受信側

SSLCAUTH について詳しくは、>[DEFINE CHANNEL \(MQTT\)](#) および [セキュリティ](#) を参照してください。

SSL ピア (SSLPEER)

この属性は、IBM MQ チャンネルの相手側にあるピア・キュー・マネージャーまたはピア・クライアントから送られてくる証明書の識別名 (DN) を検査するときに使用される属性です。

注: TLS サブジェクト識別名との突き合わせによってチャンネルへの接続を制限する別の方法は、チャンネル認証レコードを使用することです。チャンネル認証レコードを使用すると、TLS のサブジェクト識別名のさまざまなパターンを同じチャンネルに適用することができます。チャンネルで SSLPEER が設定されており、かつチャンネル認証レコードが同じチャンネルに適用されている場合、接続するには、インバウンド証明書が両方のパターンと一致する必要があります。

ピアから受信された DN が SSLPEER 値と一致しない場合は、チャンネルは開始しません。

SSLPEER はオプションの属性です。値を指定しない場合は、チャンネル開始時にピア DN は検査されません。

z/OS では、属性の最大長は 256 バイトです。それ以外のすべてのプラットフォームでは、1024 バイトです。チャンネル認証レコードによって、SSLPEER の使用時に柔軟性が大幅に向上し、すべてのプラットフォームで 1024 バイトがサポートされます。

z/OS では、使用される属性値は確認されません。正しくない値を入力すると、チャンネルは開始時に失敗し、チャンネルの両端のエラー・ログにエラー・メッセージが書き込まれます。チャンネル SSL エラー・イベントも、チャンネルの両端で生成されます。z/OS 以外の SSLPEER をサポートするプラットフォームでは、ストリングの妥当性は最初の入力時に検査されます。

SSLPEER の値を、SSLCIPH が欠落しているかブランクになっている非 TLS チャンネル定義に指定できます。これは、デバッグを行うためにいったん TLS パラメーターをクリアし、あとで再入力しなくてもよいように、TLS を一時的に使用不可にするために使用できます。

SSLPEER の使用について詳しくは、[SET CHLAUTH](#) および [セキュリティー](#) を参照してください。

この属性は、すべてのチャンネル・タイプで有効です。

関連情報

[チャンネル認証レコード](#)

V9.0.0 トピック・ルート (TPROOT)

この属性は AMQP チャンネルのトピック・ルートを指定します。

TPROOT 属性を使用して、AMQP チャンネルのトピック・ルートを指定することができます。この属性を使用すると、MQ Light アプリケーションがキュー・マネージャーにデプロイされたとき、他のアプリケーションで使用されているトピック・ツリー領域との間でメッセージのパブリッシュやサブスクライブが行われないようにすることができます。

TPROOT のデフォルト値は SYSTEM.BASE.TOPIC です。この値を設定した場合、AMQP クライアントがパブリッシュまたはサブスクライブに使用するトピック・ストリングに接頭部が付かず、クライアントは他の MQ パブリッシュ/サブスクライブ・アプリケーションとの間でメッセージを交換できます。トピック接頭部の下で AMQP クライアントにパブリッシュ/サブスクライブさせるには、まず、トピック・ストリングを適切な接頭部に設定して MQ トピック・オブジェクトを作成した後、AMQP チャンネル TPROOT 属性の値を、既に作成した MQ トピック・オブジェクトの名前に変更します。以下の例は、AMQP チャンネル MYAMQP 用に APPGROUP1.BASE.TOPIC に設定されたトピック・ルートを示しています。

```
DEFINE CHANNEL(MYAMQP) CHLTYPE(AMQP) TPROOT(APPGROUP1.BASE.TOPIC) PORT(5673)
```

注：TPROOT 属性値またはその基礎となるトピック・ストリングが変更された場合、既存の AMQP トピックとそのメッセージが孤立する可能性があります。

伝送キュー名 (XMITQ)

この属性は、メッセージの検索元の伝送キューの名前を指定します。

この属性は、タイプが送信側またはサーバーのチャンネルで必要であり、他のチャンネル・タイプでは有効ではありません。

チャンネルの相手側のキュー・マネージャーに対応した、この送信側チャンネルまたはサーバー・チャンネルと関連付けられる伝送キューの名前を指定します。伝送キューに、リモート・エンドのキュー・マネージャーと同じ名前を指定できます。

この属性は、以下のチャンネル・タイプで有効です。

- ・送信側
- ・サーバー

トランスポート・タイプ (TRPTYPE)

この属性は、使用されるトランスポート・タイプを指定します。

可能な値は、次のとおりです。

LU62	LU 6.2
TCP	TCP/IP

NETBIOS	NetBIOS (141 ページの『1』)
SPX	SPX (141 ページの『1』)
注: 1. Windows 上で使用。z/OS でも、Windows で使用するクライアント接続チャンネルの定義に使用できません。	

この属性は、すべてのチャンネル・タイプで有効ですが、応答するメッセージ・チャンネル・エージェントによって無視されます。

クライアント ID の使用 (USECLTID)

AMQP チャンネルへの接続にクライアント ID を使用します。

AMQP チャンネルの接続にクライアント ID を使用するかどうかを指定します。Yes または No に設定します。

送達不能キューの使用 (USEDLQ)

この属性は、チャンネルでメッセージが配信できない場合に、送達不能キュー (または未配布メッセージ・キュー) を使用するかどうかを判別します。

指定可能な値は以下のとおりです。

NO

チャンネルによって送信できないメッセージは、失敗したものとして扱われます。NPMSPEED の設定に従って、チャンネルがこれらのメッセージを破棄するか、チャンネルが終了します。

YES (デフォルト)

キュー・マネージャー DEADQ の属性に送達不能キューの名前が指定されている場合は、そのキューが使用されます。指定されていない場合は、NO が指定されたときの動作になります。

ユーザー ID (USERID)

この属性は、リモート MCA との保護 SNA セッションの開始を試みるときに MCA が使用するユーザー ID を指定します。

20 文字のタスク・ユーザー ID を指定できます。

これは、送信側、サーバー、要求側、またはクライアント接続のチャンネル・タイプの場合に有効です。

この属性は IBM MQ for z/OS ではクライアント接続チャンネル以外には適用されません。

受信側でパスワードが暗号形式で保持され、LU 6.2 ソフトウェアが別の暗号方式を使用している場合、チャンネルを開始しようとする、セキュリティの詳細が無効なために失敗します。この失敗は、受信側の SNA 構成を次のいずれかに変更することによって回避できます。

- パスワード置換をオフにする。
- セキュリティー・ユーザー ID およびパスワードを定義する。

IBM MQ for z/OS では、この属性はクライアント接続チャンネルでのみ有効です。他のプラットフォームでは、この属性は以下のチャンネル・タイプで有効です。

- 送信側
- サーバー
- 要求側
- クライアント接続
- クラスタ送信側

IBM MQ クラスター・コマンド

IBM MQ スクリプト・コマンドである **runmqsc** コマンドには、クラスターに適用される特殊な属性とパラメーターがあります。このほかにも、クラスターを管理するために使用できる管理インターフェースがあります。

MQSC コマンドは、システム管理者がコマンド・コンソールで入力するとおりに示されています。しかし、これが唯一の入力方法ではありません。プラットフォームに応じて、他の複数の方法があります。例を以下に示します。

- IBM MQ for IBM i では、**WRKMQM** のオプション 26 から MQSC コマンドを対話式に実行できます。また、CL コマンドを使用することも、MQSC コマンドをファイルに保管しておいて、**STRMQMMQSC CL** コマンドを使用することもできます。
- **z/OS** z/OS では、**CSQUTIL** ユーティリティの COMMAND 機能、操作パネルと制御パネル、または z/OS コンソールを使用できます。
- 他のすべてのプラットフォームでは、コマンドをファイルに保管して **runmqsc** を使用できます。

MQSC コマンドでは、CLUSTER 属性を使用して指定されるクラスター名の最大長は 48 文字です。

CLUSNL 属性を使用して指定するクラスター名リストには、最大 256 の名前を入れることができます。クラスター名前リストを作成するには、DEFINE NAMELIST コマンドを使用します。

IBM MQ Explorer

IBM MQ Explorer GUI は、IBM WebSphere MQ for z/OS 6 以降のリポジトリ・キュー・マネージャーを持つクラスターを管理できます。別個のシステムで追加のリポジトリを指名する必要はありません。以前のバージョンの IBM MQ for z/OS では、IBM MQ Explorer 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。そのため、IBM MQ Explorer 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。

IBM MQ for Windows および IBM MQ for Linux では、IBM MQ Explorer を使用してクラスターを操作することもできます。また、スタンドアロンの IBM MQ Explorer ・クライアントを使用することもできます。

IBM MQ Explorer を使用すると、クラスター・キューを表示して、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の状況を照会できます。IBM MQ Explorer には 2 種類のウィザードがあり、以下の作業のガイドとして使用できます。

- クラスターを作成する
- 独立キュー・マネージャーをクラスターに結合する

プログラマブル・コマンド・フォーマット (PCF)

runmqsc コマンド	PCF での同等のコマンド
DISPLAY CLUSQMGR	MQCMD_INQUIRE_CLUSTER_Q_MGR
SUSPEND QMGR	MQCMD_SUSPEND_Q_MGR_CLUSTER
RESUME QMGR	MQCMD_RESUME_Q_MGR_CLUSTER
REFRESH CLUSTER	MQCMD_REFRESH_CLUSTER
RESET CLUSTER	MQCMD_RESET_CLUSTER

関連情報

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

キュー・マネージャー定義コマンド

キュー・マネージャー定義コマンドに指定できるクラスター属性です。

キュー・マネージャーでクラスター用のフル・リポジトリを保持することを指定するには、REPOS(*clustername*) 属性を指定した ALTER QMGR コマンドを使用します。複数のクラスター名のリストを指定するには、クラスター名前リストを定義した後、次のように ALTER QMGR コマンドに REPOSNL(*namelist*) 属性を使用します。

```
DEFINE NAMELIST(CLUSTERLIST)
  DESCR('List of clusters whose repositories I host')
  NAMES(CLUS1, CLUS2, CLUS3)
ALTER QMGR REPOSNL(CLUSTERLIST)
```

ALTER QMGR コマンドには、追加のクラスター属性を指定することができます。

CLWLEXIT(*name*)

メッセージがクラスター・キューに書き込まれる際に呼び出すユーザー出口の名前を指定します。

CLWLDATA(*data*)

クラスター・ワークロード・ユーザー出口に渡されるデータを指定します。

CLWLLEN(*length*)

クラスター・ワークロード・ユーザー出口に渡されるメッセージ・データの最大量を指定します。

CLWLMRUC(*channels*)

アウトバウンド・クラスター・チャンネルの最大数を指定します。

CLWLMRUC は、クラスター全体には伝搬されないローカル・キュー・マネージャー属性です。これは、クラスター・ワークロード出口で使用可能で、メッセージの宛先を選択するクラスター・ワークロード・アルゴリズムです。

CLWLUSEQ(LOCAL|ANY)

ターゲット・キューにローカル・インスタンスと少なくとも1つのリモート・クラスター・インスタンスの両方がある場合の MQPUT の動作を指定します。書き込みの発生元がクラスター・チャンネルである場合、この属性は適用されません。CLWLUSEQ は、キュー属性とキュー・マネージャー属性の両方として指定することが可能です。

ANY を指定した場合、ローカル・キューとリモート・キューの両方が MQPUT の指定可能なターゲットになります。

LOCAL を指定した場合、ローカル・キューが MQPUT の唯一のターゲットになります。

同等の PCF は MQCMD_CHANGE_Q_MGR と MQCMD_INQUIRE_Q_MGR です。

関連概念

クラスターでのワークロード・バランシング

1つのクラスターに同じキューのインスタンスが複数含まれる場合、IBM MQ はメッセージを送付するキュー・マネージャーを選択します。また、クラスター・ワークロード管理アルゴリズムと、クラスター・ワークロード固有のいくつかの属性を使用して、使用するのに最適なキュー・マネージャーを判別します。

z/OS における CLUSTER コマンドの非同期の動作

z/OS におけるクラスター・コマンドのコマンド発行者は、コマンドが送信されたことの確認を受け取りませんが、それは正常に完了したということではありません。

関連資料

チャンネル定義コマンド

チャンネル定義コマンドで指定できるクラスター属性

キュー定義コマンド

キュー定義コマンドに指定できるクラスター属性です。

DISPLAY CLUSQMGR

DISPLAY CLUSQMGR コマンドは、クラスター内にあるキュー・マネージャーに関するクラスター情報を表示する場合に使用します。

SUSPEND QMGR、RESUME QMGR、およびクラスター

SUSPEND QMGR および RESUME QMGR コマンドは、このキュー・マネージャーへのインバウンド・クラスター・アクティビティーを一時的に (例えば、このキュー・マネージャーの保守を行う前に) 削減し、後で復元するために使用します。

REFRESH CLUSTER

REFRESH CLUSTER コマンドは、クラスターに関するローカルに保持されたすべての情報を廃棄する場合にキュー・マネージャーから発行します。例外的な状況を除いて、このコマンドを使用する必要はほとんどありません。

RESET CLUSTER: クラスターからキュー・マネージャーを強制的に除去する

RESET CLUSTER コマンドは、例外的な状況にあるクラスターからキュー・マネージャーを強制的に削除する場合に使用します。

チャンネル定義コマンド

チャンネル定義コマンドで指定できるクラスター属性

DEFINE CHANNEL、ALTER CHANNEL、および DISPLAY CHANNEL コマンドには、クラスター用の 2 つの特定 CHLTYPE パラメーター、CLUSRCVR および CLUSSDR があります。クラスター受信側チャンネルを定義するには、DEFINE CHANNEL コマンドで CHLTYPE (CLUSRCVR) を指定します。クラスター受信側チャンネル定義の属性の多くは、受信側チャンネル定義または送信側チャンネルの定義の属性と同じです。クラスター送信側チャンネルを定義するには、DEFINE CHANNEL コマンドで CHLTYPE (CLUSSDR) と指定し、送信側チャンネルの定義に使用すると同じ属性の多くを指定します。

クラスター送信側チャンネルを定義する際に、完全リポジトリ・キュー・マネージャーの名前を指定する必要はなくなりました。クラスター内のチャンネルに使用する命名規則を知っている場合、+QMNAME+ 構造を使用して CLUSSDR 定義を作成できます。+QMNAME+ 構造は z/OS ではサポートされていません。接続後に、IBM MQ はチャンネルの名前を変更し、+QMNAME+ を正しい完全リポジトリ・キュー・マネージャー名に置換します。結果として得られるチャンネル名は 20 文字に切り捨てられます。

命名規則の詳細については、[クラスターの命名規則](#)を参照してください。

この手法が機能するのは、チャンネルの命名の規則にキュー・マネージャーの名前が含まれている場合だけです。例えば、CLUSTER1 というクラスターの QM1 という完全リポジトリ・キュー・マネージャーを、CLUSTER1.QM1.ALPHA というクラスター受信側チャンネルで定義するとします。他のキュー・マネージャーはすべて、このキュー・マネージャーに対して、チャンネル名である CLUSTER1.+QMNAME+.ALPHA を使用して、クラスター送信側チャンネルを定義することができます。

同じ命名規則をすべてのチャンネルに使用する場合、一度に存在できる +QMNAME+ 定義は 1 つだけであることに注意してください。

DEFINE CHANNEL コマンドと ALTER CHANNEL コマンドの次の属性は、クラスター・チャンネルに固有です。

CLUSTER

CLUSTER 属性は、このチャンネルが関連付けられるクラスターの名前を指定します。または、CLUSNL 属性を使用します。

CLUSNL

CLUSNL 属性はクラスター名の名前リストを指定します。

NETPRTY

クラスター受信側だけです。

NETPRTY 属性は、チャンネルのネットワーク優先度を指定します。NETPRTY はワークロード管理ルーチンの働きを助けます。宛先への可能な経路が 2 つ以上ある場合、ワークロード管理ルーチンは優先度が最も高いチャンネルへの経路を選択します。

CLWLPRTY

CLWLPRTY パラメーターは、ワークロード管理のために、同じ宛先へのチャンネルに優先度係数を適用します。このパラメーターは、クラスター・ワークロードを分散するために、チャンネルの優先度を指定します。値の範囲はゼロ (最低の優先度) から 9 (最高の優先度) でなければなりません。

CLWLRANK

CLWLRANK パラメーターは、ワークロード管理のために、チャンネルにランキング係数を適用します。このパラメーターは、クラスター・ワークロードを分散するために、チャンネルのランクを指定します。値の範囲はゼロ (最低ランク) から 9 (最高ランク) でなければなりません。

CLWLWGHT

CLWLWGHT パラメーターは、ワークロード管理のためにチャンネルに加重係数を適用します。CLWLWGHT は、そのチャンネルを経由して送信されるメッセージの比率を制御できるように、チャンネルを重み付けします。クラスター・ワークロード・アルゴリズムは CLWLWGHT を使用して、特定のチャンネルを介してより多くのメッセージを送信できるように宛先の選択を偏らせます。デフォルトでは、すべてのチャンネルの重み属性が同じデフォルト値です。重み属性では、強力な UNIX マシン上のチャンネルには、小さいデスクトップ PC 上の別のチャンネルより大きい重みを割り振ることができます。より大きい重みを割り振ることにより、クラスター・ワークロード・アルゴリズムは、メッセージの宛先として、PC より UNIX マシンをより頻繁に選択します。

CONNAME

クラスター受信側チャンネル定義に指定された CONNAME は、キュー・マネージャーのネットワーク・アドレスを識別するためにクラスター全体で使用されます。CONNAME パラメーターの値は、IBM MQ クラスター全体を通じて解決されるため、注意して選択してください。総称名は使用できません。クラスター受信側チャンネルで指定された値は、対応するクラスター送信側チャンネルに指定されたすべての値に優先することに注意してください。

DEFINE CHANNEL コマンドおよび ALTER CHANNEL コマンドのこれらの属性は、DISPLAY CHANNEL コマンドにも適用されます。

注: 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ても、その属性は対応するクラスター受信側の定義に指定された属性に合わせて自動的に修正されます。例えば、ポート番号を指定せずに CONNAME パラメーターに CLUSRCVR を定義することができます。ただし、ポート番号を指定する CLUSSDR を手操作で定義することになりますので注意してください。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、チャンネルには障害が発生します。

注: DISPLAY CHANNEL コマンドでは、自動定義チャンネルは表示されません。しかし、DISPLAY CLUSQMGR コマンドを使用すると、自動定義クラスター送信側チャンネルの属性を調べることができます。

DISPLAY CHSTATUS コマンドを使用すると、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の状況を表示できます。このコマンドでは、手作業で定義したチャンネルと自動的に定義されたチャンネルの両方の状況が表示されます。

同等の PCF は、MQCMD_CHANGE_CHANNEL、MQCMD_COPY_CHANNEL、MQCMD_CREATE_CHANNEL、および MQCMD_INQUIRE_CHANNEL です。

CLUSRCVR 定義で CONNAME 値を省略する

状況によっては、CLUSRCVR 定義で CONNAME 値を省略することができます。z/OS では、CONNAME 値を省略してはなりません。

Multi マルチプラットフォームでは、クラスター受信側チャンネルの TCP/IP 接続名パラメーターはオプションです。接続名をブランクにすると、IBM MQ はデフォルト・ポートを想定し、システムの現行 IP アドレスを使用して接続名を自動的に生成します。デフォルト・ポート番号をオーバーライドしても、システムの現行 IP アドレスを引き続き使用できます。各接続名について、IP 名をブランクにして、次のように括弧で囲んだポート番号を指定してください。

(1415)

生成される **CONNAME** は常にドット 10 進 (IPv4) 形式または 16 進 (IPv6) 形式であり、英数字の DNS ホスト名の形式ではありません。

この機能は、動的ホスト構成プロトコル (DHCP) を使用するマシンがある場合に役立ちます。CLUSRCVR チャンネル上の CONNAME の値を指定しない場合、CLUSRCVR 定義に変更を加える必要はありません。DHCP により新しい IP アドレスが割り振られます。

CLUSRCVR 定義で CONNAME に空白を指定した場合は、IBM MQ はシステムの IP アドレスから CONNAME を生成します。生成された CONNAME だけがリポジトリに格納されます。クラスター内の他のキュー・マネージャーは、CONNAME が当初は空白だったことを知りません。

DISPLAY CLUSQMGR コマンドを実行すると、生成された CONNAME が表示されます。しかし、ローカル・キュー・マネージャーから DISPLAY CHANNEL コマンドを実行する場合には、CONNAME が空白として表示されます。

キュー・マネージャーが停止し、別の IP アドレスで再始動した場合には、DHCP のために、IBM MQ は CONNAME を再生成し、それに従ってリポジトリを更新します。

関連概念

クラスターでのワークロード・バランシング

1つのクラスターに同じキューのインスタンスが複数含まれる場合、IBM MQ はメッセージを送付するキュー・マネージャーを選択します。また、クラスター・ワークロード管理アルゴリズムと、クラスター・ワークロード固有のいくつかの属性を使用して、使用するのに最適なキュー・マネージャーを判別します。

z/OS における CLUSTER コマンドの非同期の動作

z/OS におけるクラスター・コマンドのコマンド発行者は、コマンドが送信されたことの確認を受け取りませんが、それは正常に完了したということではありません。

関連資料

キュー・マネージャー定義コマンド

キュー・マネージャー定義コマンドに指定できるクラスター属性です。

キュー定義コマンド

キュー定義コマンドに指定できるクラスター属性です。

DISPLAY CLUSQMGR

DISPLAY CLUSQMGR コマンドは、クラスター内にあるキュー・マネージャーに関するクラスター情報を表示する場合に使用します。

SUSPEND QMGR、RESUME QMGR、およびクラスター

SUSPEND QMGR および RESUME QMGR コマンドは、このキュー・マネージャーへのインバウンド・クラスター・アクティビティーを一時的に (例えば、このキュー・マネージャーの保守を行う前に) 削減し、後で復元するために使用します。

REFRESH CLUSTER

REFRESH CLUSTER コマンドは、クラスターに関するローカルに保持されたすべての情報を廃棄する場合にキュー・マネージャーから発行します。例外的な状況を除いて、このコマンドを使用する必要はほとんどありません。

RESET CLUSTER: クラスターからキュー・マネージャーを強制的に除去する

RESET CLUSTER コマンドは、例外的な状況にあるクラスターからキュー・マネージャーを強制的に削除する場合に使用します。

キュー定義コマンド

キュー定義コマンドに指定できるクラスター属性です。

コマンド DEFINE QLOCAL、DEFINE QREMOTE、および DEFINE QALIAS

DEFINE QLOCAL、DEFINE QREMOTE、および DEFINE QALIAS コマンドのクラスター属性、および同等の3つの ALTER コマンドは、以下のとおりです。

CLUSTER

キューが属するクラスターの名前を指定します。

CLUSNL

クラスター名の名前リストを指定します。

DEFBIND

アプリケーションでMQOPEN呼び出しにMQOO_BIND_AS_Q_DEFを指定している場合に使用するバインディングを指定します。この属性のオプションは、以下のとおりです。

- キューがオープンされる際にキュー・ハンドルをクラスター・キューの特定のインスタンスにバインドするには、DEFBIND(OPEN)を指定します。DEFBIND(OPEN)がこの属性のデフォルトです。
- キュー・ハンドルがクラスター・キューのすべてのインスタンスと結合しないようにするには、DEFBIND(NOTFIXED)を指定します。
- メッセージのグループをすべて同じ宛先インスタンスに割り振るようにアプリケーションで要求できるようにするには、DEFBIND(GROUP)を指定します。

同じ名前の複数のキューがキュー・マネージャー・クラスター内で公示されている場合、アプリケーションでは、このアプリケーションから単一インスタンスにすべてのメッセージを送信する(MQOO_BIND_ON_OPEN)か、ワークロード管理アルゴリズムでメッセージごとに最も適切な宛先の選択を可能にする(MQOO_BIND_NOT_FIXED)か、またはアプリケーションでメッセージのグループをすべて同じ宛先インスタンスに割り当てるように要求することを可能にする(MQOO_BIND_ON_GROUP)かを選択することができます。ワークロード・バランシングは、メッセージのグループ間を再度実行します(キューのMQCLOSEおよびMQOPENを要求しないで)。

キュー定義にDEFBINDを指定する際、MQBND_BIND_ON_OPEN、MQBND_BIND_NOT_FIXED、またはMQBND_BIND_ON_GROUPのいずれかの属性がキューに定義されます。クラスターでグループを使用する場合は、MQBND_BIND_ON_OPENまたはMQBND_BIND_ON_GROUPのいずれかを指定する必要があります。

DEFBIND属性は、同じクラスター・キューのすべてのインスタンスで同じ値に設定することをお勧めします。MQOO_BIND_ON_GROUPは、IBM WebSphere MQ 7.1の新規属性であるので、このキューを開くいずれかのアプリケーションがIBM WebSphere MQ 7.0.1以前のキュー・マネージャーに接続している場合は使用してはなりません。

CLWLRANK

ワークロード管理のためにランキング因子をキューに適用します。CLWLRANKパラメーターは、モデル・キューではサポートされません。クラスター・ワークロード・アルゴリズムは、ランクが最も高い宛先キューを選択します。デフォルトでは、すべてのキューのCLWLRANKがゼロに設定されます。

最終宛先が異なるクラスター上のキュー・マネージャーの場合、隣接するクラスターの交差ですべての中間ゲートウェイ・キュー・マネージャーのランクを設定できます。ランク付けされた中間キュー・マネージャーに関し、クラスター・ワークロード・アルゴリズムでは、最終宛先により近い宛先キュー・マネージャーを正しく選択します。

同じロジックが別名キューに適用されます。ランクの選択は、チャンネル状況の検査前に行われるため、アクセス不可能なキュー・マネージャーでも選択できます。これには、(優先順位を指定する場合のように)2つの有効な宛先の中から選択するのではなく、メッセージをネットワークを介して経路指定できるという効果があります。そのため、ランクが示す場所でチャンネルが開始されていない場合、メッセージは次に高いランクに経路指定されません。ただし、チャンネルがその宛先で使用可能になるまで待機します(メッセージは伝送キューに保持されます)。

CLWLPRTY

ワークロード管理のために優先順位因子をキューに適用します。クラスター・ワークロード・アルゴリズムは、優先順位が最も高い宛先キューを選択します。デフォルトでは、すべてのキューの優先順位がゼロに設定されます。

有効な宛先キューが2つある場合は、この属性を使用して、一方の宛先をもう一方の宛先にフェイルオーバーできます。優先順位の選択は、チャンネル状況の検査後に行われます。すべてのメッセージは、優先順位が最も高いキューに送信されます。ただし、その宛先へのチャンネルの状況が、他の宛先へのチャンネルの状況ほど良好ではない場合を除きます。つまり、アクセスできる可能性が最も高い宛先のみを選択できます。これは、どれも使用可能である複数の宛先の間で優先順位付けを行えるという効果があります。

CLWLUSEQ

あるキューのMQPUT 操作の動作を指定します。このパラメーターでは、ターゲット・キューにローカル・インスタンスがあり、少なくとも1つのリモート・クラスター・インスタンスがある場合のMQPUT 操作の動作を指定します(MQPUT がクラスター・チャンネルから発信されている場合を除く)。このパラメーターは、ローカル・キューにのみ有効です。

指定可能な値は、QMGR (この動作は、キュー・マネージャー定義の CLWLUSEQ パラメーターで指定されたとおりになります)、ANY (キュー・マネージャーは、ワークロードを分散するために、ローカル・キューをクラスター・キューの別のインスタンスとして処理します)、LOCAL です (ローカル・キューが書き込み可能であれば、ローカル・キューがMQPUT 操作の唯一の宛先です)。MQPUT の動作は、クラスター・ワークロード管理アルゴリズムによって異なります。

コマンド DISPLAY QUEUE および DISPLAY QCLUSTER

DEFINE QLOCAL、DEFINE QREMOTE、および DEFINE QALIAS コマンドの属性は、DISPLAY QUEUE コマンドにも適用されます。

クラスター・キューに関する情報を表示するには、キュー・タイプの QCLUSTER またはキーワードの CLUSINFO を DISPLAY QUEUE コマンドに指定するか、あるいは DISPLAY QCLUSTER コマンドを使用します。

DISPLAY QUEUE コマンドと DISPLAY QCLUSTER コマンドは、キューをホストするキュー・マネージャーの名前 (複数のキュー・インスタンスがある場合は、すべてのキュー・マネージャーの名前) を返します。また、キューのホストである各キュー・マネージャーのシステム名、キューのタイプ、および定義がローカル・キュー・マネージャーに使用可能になった日時を返します。この情報は、CLUSQMGR、QMID、CLUSQT、CLUSDATE、および CLUSTIME の各属性を使用して返されます。

キュー・マネージャーのシステム名 (QMID) は、キュー・マネージャーに対してシステムによって生成される固有の名前です。

共有キューでもあるクラスター・キューを定義できます。例えば、z/OS では、次のように定義できます。

```
DEFINE QLOCAL(MYQUEUE) CLUSTER(MYCLUSTER) QSGDISP(SHARED) CFSTRUCT(STRUCTURE)
```

同等の PCF は、MQCMD_CHANGE_Q、MQCMD_COPY_Q、MQCMD_CREATE_Q、および MQCMD_INQUIRE_Q です。

関連概念

クラスターでのワークロード・バランシング

1つのクラスターに同じキューのインスタンスが複数含まれる場合、IBM MQ はメッセージを送付するキュー・マネージャーを選択します。また、クラスター・ワークロード管理アルゴリズムと、クラスター・ワークロード固有のいくつかの属性を使用して、使用するのに最適なキュー・マネージャーを判別します。

 z/OS における CLUSTER コマンドの非同期の動作

z/OS におけるクラスター・コマンドのコマンド発行者は、コマンドが送信されたことの確認を受け取りませんが、それは正常に完了したということではありません。

関連資料

キュー・マネージャー定義コマンド

キュー・マネージャー定義コマンドに指定できるクラスター属性です。

チャンネル定義コマンド

チャンネル定義コマンドで指定できるクラスター属性

DISPLAY CLUSQMGR

DISPLAY CLUSQMGR コマンドは、クラスター内にあるキュー・マネージャーに関するクラスター情報を表示する場合に使用します。

SUSPEND QMGR、RESUME QMGR、およびクラスター

SUSPEND QMGR および RESUME QMGR コマンドは、このキュー・マネージャーへのインバウンド・クラスター・アクティビティーを一時的に (例えば、このキュー・マネージャーの保守を行う前に) 削減し、後で復元するために使用します。

REFRESH CLUSTER

REFRESH CLUSTER コマンドは、クラスターに関するローカルに保持されたすべての情報を廃棄する場合にキュー・マネージャーから発行します。例外的な状況を除いて、このコマンドを使用する必要はほとんどありません。

RESET CLUSTER: クラスターからキュー・マネージャーを強制的に除去する

RESET CLUSTER コマンドは、例外的な状況にあるクラスターからキュー・マネージャーを強制的に削除する場合に使用します。

DISPLAY CLUSQMGR

DISPLAY CLUSQMGR コマンドは、クラスター内にあるキュー・マネージャーに関するクラスター情報を表示する場合に使用します。

このコマンドをフル・リポジトリを保持するキュー・マネージャーから発行した場合、返される情報はクラスター内のすべてのキュー・マネージャーに当てはまります。フル・リポジトリを保持しないキュー・マネージャーから発行した場合、返される情報は、関係のあるキュー・マネージャーのみに当てはまります。つまり、メッセージを送信しようとした送信先のキュー・マネージャー、およびフル・リポジトリを保持しているすべてのキュー・マネージャーに関する情報が返されます。

この情報には、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に適用される、ほとんどのチャンネル属性が含まれています。また、以下の属性も表示することができます。

CHANNEL

キュー・マネージャーのクラスター受信側チャンネル名。

CLUSDATE

定義がローカル・キュー・マネージャーに使用可能になった日付。

CLUSTER

キュー・マネージャーがどのクラスターに属しているか。

CLUSTIME

定義がローカル・キュー・マネージャーに有効になった時刻。

DEFTYPE

キュー・マネージャーがどのように定義されたか。DEFTYPE は、次のいずれかになります。

CLUSDR

クラスター送信側チャンネルが管理者によってローカル・キュー・マネージャーに定義されましたが、まだターゲット・キュー・マネージャーには認識されていません。こうした状態になるのは、ローカル・キュー・マネージャーで手動によりクラスター送信側チャンネルを定義したが、受信側キュー・マネージャーがそのクラスター情報をまだ受け入れていない場合です。この原因としては、使用可能な状態ではなかったためにチャンネルが確立されなかったか、送信側と受信側で CLUSTER プロパティーの定義が一致していないなどのクラスター送信側の構成にエラーがあることが考えられます。これは一時的な状態またはエラー状態であり、調査する必要があります。

CLUSDR

この値は自動的に検出されたクラスター・キュー・マネージャーを表します。ローカルにはクラスター送信側チャンネルは定義されていません。これは、ローカル・キュー・マネージャーにはローカル構成は存在しないが、ローカル・キュー・マネージャーに通知されたことのあるクラスター・キュー・マネージャーを表す DEFTYPE です。例えば

- ローカル・キュー・マネージャーが完全リポジトリ・キュー・マネージャーの場合、これは、クラスター内のすべての部分リポジトリ・キュー・マネージャーの DEFTYPE 値になります。
- ローカル・キュー・マネージャーが部分リポジトリの場合、これは、このローカル・キュー・マネージャーから使用されるクラスター・キュー、または、このキュー・マネージャーに連携するように指定した 2 番目の完全リポジトリ・キュー・マネージャーから使用されるクラスター・キューのホストになります。

DEFTYPE 値が CLUSSDRA であり、ローカル・キュー・マネージャーとリモート・キュー・マネージャーの両方が特定のクラスターの完全リポジトリである場合、この構成は正しくありません。ローカルに定義するクラスター送信側チャンネルで、これを CLUSSDRB の DEFTYPE に変換するように定義する必要があるからです。

CLUSSDRB

クラスター送信側チャンネルが管理者によってローカル・キュー・マネージャー上に定義され、ターゲット・キュー・マネージャーにも有効なクラスター・チャンネルとして受け入れられました。これは、部分リポジトリ・キュー・マネージャーの手動構成された完全リポジトリ・キュー・マネージャーに予期される DEFTYPE です。これは、クラスター内の 1 つの完全リポジトリから別の完全リポジトリに対する CLUSQMGR の DEFTYPE でもあります。手動クラスター送信側チャンネルを、複数の部分リポジトリに対して構成することはできません。また、1 つの部分リポジトリ・キュー・マネージャーから複数の完全リポジトリに対して構成することもできません。このような状況で CLUSSDRB である DEFTYPE が見られた場合は、調査して修正する必要があります。

CLUSRCVR

ローカル・キュー・マネージャー上にクラスター受信側チャンネルとして管理者によって定義されました。これはクラスター内のローカル・キュー・マネージャーを表します。

注：どの CLUSQMGR がクラスターの完全リポジトリ・キュー・マネージャーであるかを特定するには、QMTYPE プロパティを調べます。

クラスター・チャンネルの定義について詳しくは、『クラスター・チャンネル』を参照してください。

QMTYPE

完全リポジトリを持つか、部分リポジトリのみを持つか。

STATUS

このキュー・マネージャーのクラスター送信側チャンネルの状況。

SUSPEND

キュー・マネージャーが中断されているかどうか。

VERSION

クラスター・キュー・マネージャーが関連付けられている IBM MQ インストールのバージョン。

バージョンの形式は、以下のような VVRRMMFF です。

- VV: バージョン
- RR: リリース
- MM: 保守レベル
- FF: フィックス・レベル

XMITQ

キュー・マネージャーにより使用されるクラスター伝送キュー。

また、DISPLAY QCLUSTER コマンドも参照してください。これについては、DISPLAY QUEUE、および 146 ページの『キュー定義コマンド』の DISPLAY QUEUE コマンドと DISPLAY QCLUSTER コマンド・セクションで簡単に説明されています。DISPLAY QCLUSTER の使用例については、"DISPLAY QCLUSTER" および "DIS QCLUSTER" に関する情報を検索してください。

関連概念

クラスターでのワークロード・バランシング

1 つのクラスターに同じキューのインスタンスが複数含まれる場合、IBM MQ はメッセージを送付するキュー・マネージャーを選択します。また、クラスター・ワークロード管理アルゴリズムと、クラスター・ワークロード固有のいくつかの属性を使用して、使用するのに最適なキュー・マネージャーを判別します。

z/OS

z/OS における CLUSTER コマンドの非同期的動作

z/OS におけるクラスター・コマンドのコマンド発行者は、コマンドが送信されたことの確認を受け取りませんが、それは正常に完了したということではありません。

関連資料

キュー・マネージャー定義コマンド

キュー・マネージャー定義コマンドに指定できるクラスター属性です。

チャンネル定義コマンド

チャンネル定義コマンドで指定できるクラスター属性

キュー定義コマンド

キュー定義コマンドに指定できるクラスター属性です。

SUSPEND QMGR、RESUME QMGR、およびクラスター

SUSPEND QMGR および RESUME QMGR コマンドは、このキュー・マネージャーへのインバウンド・クラスター・アクティビティーを一時的に (例えば、このキュー・マネージャーの保守を行う前に) 削減し、後で復元するために使用します。

REFRESH CLUSTER

REFRESH CLUSTER コマンドは、クラスターに関するローカルに保持されたすべての情報を廃棄する場合にキュー・マネージャーから発行します。例外的な状況を除いて、このコマンドを使用する必要はほとんどありません。

RESET CLUSTER: クラスターからキュー・マネージャーを強制的に除去する

RESET CLUSTER コマンドは、例外的な状況にあるクラスターからキュー・マネージャーを強制的に削除する場合に使用します。

関連情報

MQSC command **DISPLAY CLUSQMGR**

SUSPEND QMGR、RESUME QMGR、およびクラスター

SUSPEND QMGR および RESUME QMGR コマンドは、このキュー・マネージャーへのインバウンド・クラスター・アクティビティーを一時的に (例えば、このキュー・マネージャーの保守を行う前に) 削減し、後で復元するために使用します。

クラスターのキュー・マネージャーが中断状態になっているとき、そのクラスター内の代替キュー・マネージャーに同じ名前の使用可能キューがある場合、停止中のキュー・マネージャーは、そこでホストするクラスター・キューのメッセージを受け取りません。ただし、このキュー・マネージャーが明示的にメッセージのターゲットになっている場合や、このターゲット・キューを使用できるのがこのキュー・マネージャーだけである場合、メッセージは引き続きこのキュー・マネージャーに送信されます。

このクラスターのクラスター受信側チャンネルを停止すると、キュー・マネージャーが中断状態になっているときにこれ以上インバウンド・メッセージを受け取らないように設定できます。クラスターのクラスター受信側チャンネルを停止するには、**SUSPEND QMGR** コマンドの **FORCE** モードを使用します。

関連概念

クラスターでのワークロード・バランシング

1つのクラスターに同じキューのインスタンスが複数含まれる場合、IBM MQ はメッセージを送付するキュー・マネージャーを選択します。また、クラスター・ワークロード管理アルゴリズムと、クラスター・ワークロード固有のいくつかの属性を使用して、使用するのに最適なキュー・マネージャーを判別します。

z/OS における CLUSTER コマンドの非同期の動作

z/OS におけるクラスター・コマンドのコマンド発行者は、コマンドが送信されたことの確認を受け取りませんが、それは正常に完了したということではありません。

関連資料

キュー・マネージャー定義コマンド

キュー・マネージャー定義コマンドに指定できるクラスター属性です。

チャンネル定義コマンド

チャンネル定義コマンドで指定できるクラスター属性

キュー定義コマンド

キュー定義コマンドに指定できるクラスター属性です。

DISPLAY CLUSQMGR

DISPLAY CLUSQMGR コマンドは、クラスター内にあるキュー・マネージャーに関するクラスター情報を表示する場合に使用します。

REFRESH CLUSTER

REFRESH CLUSTER コマンドは、クラスターに関するローカルに保持されたすべての情報を廃棄する場合にキュー・マネージャーから発行します。例外的な状況を除いて、このコマンドを使用する必要はほとんどありません。

RESET CLUSTER: クラスターからキュー・マネージャーを強制的に除去する

RESET CLUSTER コマンドは、例外的な状況にあるクラスターからキュー・マネージャーを強制的に削除する場合に使用します。

関連情報

SUSPEND QMGR

RESUME QMGR

キュー・マネージャーの保守

REFRESH CLUSTER

REFRESH CLUSTER コマンドは、クラスターに関するローカルに保持されたすべての情報を廃棄する場合にキュー・マネージャーから発行します。例外的な状況を除いて、このコマンドを使用する必要はほとんどありません。

このコマンドの形式には、以下の3つがあります。

REFRESH CLUSTER(clustername) REPOS(NO)

これがデフォルトです。キュー・マネージャーには、ローカルで定義されているすべてのクラスター・キュー・マネージャーとクラスター・キュー、およびフル・リポジトリのすべてのクラスター・キュー・マネージャーに関する情報が保持されます。また、キュー・マネージャーがクラスターのフル・リポジトリの場合は、クラスター内の他のクラスター・キュー・マネージャーに関する情報も保持されます。他のすべてのものはリポジトリのローカル・コピーから除去され、クラスター内の他の完全リポジトリから再作成されます。REPOS(NO)を使用すると、クラスター・チャンネルは停止されません。フル・リポジトリではCLUSDR チャンネルを使用して、そのリフレッシュが完了したことをクラスターの他の部分に通知します。

REFRESH CLUSTER(clustername) REPOS(YES)

デフォルトの動作に加えて、フル・リポジトリのクラスター・キュー・マネージャーを表すオブジェクトもリフレッシュされます。キュー・マネージャーがフル・リポジトリの場合、このオプションを使うことは適切ではありません。使用した場合、コマンドはエラー AMQ9406/CSQX406E をログして失敗します。キュー・マネージャーがフル・リポジトリである場合、まずは、問題のクラスターのフル・リポジトリではなくなるように、変更する必要があります。完全リポジトリの場所は、手作業で定義されたCLUSDR 定義から回復されます。REPOS(YES) によるリフレッシュが発行された後は、キュー・マネージャーに変更を加えることができ、必要な場合はそれを再びフル・リポジトリにすることができます。

REFRESH CLUSTER(*)

キュー・マネージャーは、そのキュー・マネージャーがメンバーであるすべてのクラスターでリフレッシュされます。REPOS(YES) REFRESH CLUSTER(*) と共に使用すると、そのキュー・マネージャーによるローカルのCLUSDR 定義内の情報に基づくフル・リポジトリの検索を強制的に再開するというその他の効果もあります。この検索は、CLUSDR チャンネルでこのキュー・マネージャーを複数のクラスターに接続している場合でも実行されます。

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。

関連概念

クラスターでのワークロード・バランシング

1つのクラスターに同じキューのインスタンスが複数含まれる場合、IBM MQ はメッセージを送付するキュー・マネージャーを選択します。また、クラスター・ワークロード管理アルゴリズムと、クラスター・ワークロード固有のいくつかの属性を使用して、使用するのに最適なキュー・マネージャーを判別します。

 z/OS における CLUSTER コマンドの非同期の動作

z/OS におけるクラスター・コマンドのコマンド発行者は、コマンドが送信されたことの確認を受け取りますが、それは正常に完了したということではありません。

関連資料

キュー・マネージャー定義コマンド

キュー・マネージャー定義コマンドに指定できるクラスター属性です。

チャンネル定義コマンド

チャンネル定義コマンドで指定できるクラスター属性

キュー定義コマンド

キュー定義コマンドに指定できるクラスター属性です。

DISPLAY CLUSQMgr

DISPLAY CLUSQMgr コマンドは、クラスター内にあるキュー・マネージャーに関するクラスター情報を表示する場合に使用します。

SUSPEND QMgr、RESUME QMgr、およびクラスター

SUSPEND QMgr および RESUME QMgr コマンドは、このキュー・マネージャーへのインバウンド・クラスター・アクティビティを一時的に (例えば、このキュー・マネージャーの保守を行う前に) 削減し、後で復元するために使用します。

RESET CLUSTER: クラスターからキュー・マネージャーを強制的に除去する

RESET CLUSTER コマンドは、例外的な状況にあるクラスターからキュー・マネージャーを強制的に削除する場合に使用します。

関連情報

クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス

RESET CLUSTER: クラスターからキュー・マネージャーを強制的に除去する

RESET CLUSTER コマンドは、例外的な状況にあるクラスターからキュー・マネージャーを強制的に削除する場合に使用します。

例外的な状況を除いて、このコマンドを使用する必要はほとんどありません。

RESET CLUSTER コマンドは、フル・リポジトリのキュー・マネージャーからのみ発行することができます。このコマンドは、キュー・マネージャーを名前または ID のいずれかで参照するかによって 2 つの形式があります。

1.

```
RESET CLUSTER( clustername
) QMNAME( qmname ) ACTION(FORCEREMOVE) QUEUES(NO)
```
2.

```
RESET CLUSTER( clustername
) QMID( qmid ) ACTION(FORCEREMOVE) QUEUES(NO)
```

QMNAME と QMID の両方を指定することはできません。QMNAME を使用した場合、その名前を持つキュー・マネージャーがクラスター内に複数あると、このコマンドは実行されません。**RESET CLUSTER** コマンドを確実に実行するには、QMNAME ではなく QMID を使用してください。

RESET CLUSTER コマンドで QUEUES(NO) を指定することがデフォルトです。QUEUES(YES) を指定すると、キュー・マネージャーによって所有されているクラスター・キューへの参照がクラスターから削除されます。つまり、キュー・マネージャーがクラスター自体から削除されるだけでなく、参照も削除されます。

参照は、クラスター・キュー・マネージャーがクラスター内で表示されない場合にも削除されます。これは、おそらくこのクラスター・キュー・マネージャーが前に QUEUES オプションなしで強制的に削除されているためです。

例えば、キュー・マネージャーは削除済みであるものの、クラスター受信側チャンネルが依然クラスターに定義されている場合、**RESET CLUSTER** コマンドを使用できます。これらの定義の IBM MQ による削除 (この削除は自動的に実行されます) を待つのではなく、**RESET CLUSTER** コマンドを発行してすぐに整理

することができます。この場合、次にクラスター内の他のすべてのキュー・マネージャーに、このキュー・マネージャーが使用できなくなったことが通知されます。

キュー・マネージャーが一時的に損傷を受けている場合、クラスター内の他のキュー・マネージャーがそのキュー・マネージャーにメッセージの送信を試みる前に、それらのキュー・マネージャーに通知を行うことがあります。**RESET CLUSTER**では、損傷を受けたキュー・マネージャーが削除されます。後で、損傷を受けたキュー・マネージャーを再び実行する場合は、**REFRESH CLUSTER** コマンドを使用して **RESET CLUSTER** の効果を取り消し、キュー・マネージャーをクラスターに戻します。キュー・マネージャーがパブリッシュ/サブスクライブ・クラスター内にある場合は、必要なプロキシ・サブスクリプションをすべて復元する必要があります。[パブリッシュ/サブスクライブ・クラスターでの REFRESH CLUSTER についての考慮事項](#)を参照してください。

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。

自動定義されたクラスター送信側チャンネルを削除する唯一の方法は、**RESET CLUSTER** コマンドを使用することです。このコマンドは、通常環境で必要になることはありません。IBM サポート・センターでは、クラスター・キュー・マネージャーによって保持されたクラスター情報を整理するために、このコマンドを発行するようにアドバイスする場合があります。このコマンドを、クラスターからキュー・マネージャーを削除するための簡単な方法として使用しないでください。クラスターからキュー・マネージャーを削除する正しい方法は、[クラスターからのキュー・マネージャーの除去](#)で説明されています。

リポジトリでは 90 日間だけ情報を保持しているため、その期間以降は強制的に削除されたキュー・マネージャーをクラスターに再接続することができます。キュー・マネージャーが削除されていない限り、それは自動的に再接続します。キュー・マネージャーがクラスターに再び参加しないようにするには、適切なセキュリティ対策を講じる必要があります。

DISPLAY CLUSQMgr 以外のクラスター・コマンドは、すべて非同期で機能します。クラスター化を含むオブジェクト属性を変更するコマンドでは、オブジェクトが更新され、リポジトリ・プロセッサに要求が送信されます。クラスターを操作するコマンドは、構文が検査されてから要求がリポジトリ・プロセッサに送信されます。

リポジトリ・プロセッサに送信された要求は、クラスターの他のメンバーから受信されたクラスター要求とともに、非同期で処理されます。この処理は、要求が正常に実行されたかどうかを判別するために、要求がクラスター全体に伝搬されなければならない場合、相当な時間がかかる場合があります。

関連概念

[クラスターでのワークロード・バランシング](#)

1つのクラスターに同じキューのインスタンスが複数含まれる場合、IBM MQ はメッセージを送付するキュー・マネージャーを選択します。また、クラスター・ワークロード管理アルゴリズムと、クラスター・ワークロード固有のいくつかの属性を使用して、使用するのに最適なキュー・マネージャーを判別します。

z/OS における CLUSTER コマンドの非同期の動作

z/OS におけるクラスター・コマンドのコマンド発行者は、コマンドが送信されたことの確認を受け取りますが、それは正常に完了したということではありません。

関連資料

[キュー・マネージャー定義コマンド](#)

キュー・マネージャー定義コマンドに指定できるクラスター属性です。

[チャンネル定義コマンド](#)

チャンネル定義コマンドで指定できるクラスター属性

[キュー定義コマンド](#)

キュー定義コマンドに指定できるクラスター属性です。

[DISPLAY CLUSQMgr](#)

DISPLAY CLUSQMgr コマンドは、クラスター内にあるキュー・マネージャーに関するクラスター情報を表示する場合に使用します。

SUSPEND QMGR、RESUME QMGR、およびクラスター

SUSPEND QMGR および RESUME QMGR コマンドは、このキュー・マネージャーへのインバウンド・クラスター・アクティビティーを一時的に (例えば、このキュー・マネージャーの保守を行う前に) 削減し、後で復元するために使用します。

REFRESH CLUSTER

REFRESH CLUSTER コマンドは、クラスターに関するローカルに保持されたすべての情報を廃棄する場合にキュー・マネージャーから発行します。例外的な状況を除いて、このコマンドを使用する必要はほとんどありません。

RESET CLUSTER (クラスターのリセット)

クラスターでのワークロード・バランシング

1つのクラスターに同じキューのインスタンスが複数含まれる場合、IBM MQ はメッセージを送付するキュー・マネージャーを選択します。また、クラスター・ワークロード管理アルゴリズムと、クラスター・ワークロード固有のいくつかの属性を使用して、使用するのに最適なキュー・マネージャーを判別します。

最適な宛先は、キュー・マネージャーとキューが使用可能であるかどうかに基づいて、またキュー・マネージャー、キュー、およびチャンネルに関連付けられたクラスター・ワークロード固有のいくつかの属性に基づいて、クラスター・ワークロード管理アルゴリズムによって選択されます。これらの属性については、サブトピックで説明しています。

注: クラスター・ワークロードのチャンネル属性は、ターゲット・キュー・マネージャーのクラスター受信側チャンネルで指定します。対応するクラスター送信側チャンネルで指定するバランシングは無視される可能性があります。[クラスター・チャンネル](#)を参照してください。

クラスターのワークロード固有の属性を構成した後に、その構成が予期したとおりに動作しない場合は、キュー・マネージャーを選択するアルゴリズムの仕組みの詳細を調べてください。[165 ページの『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。このアルゴリズムの結果がニーズに合っていない場合は、クラスター・ワークロードのユーザー出口プログラムを作成し、この出口を使用してクラスター内の適切なキューにメッセージをルーティングすることができます。[クラスター・ワークロード出口の作成とコンパイル](#)を参照してください。

関連概念

[z/OS における CLUSTER コマンドの非同期の動作](#)

z/OS におけるクラスター・コマンドのコマンド発行者は、コマンドが送信されたことの確認を受け取りませんが、それは正常に完了したということではありません。

関連資料

[キュー・マネージャー定義コマンド](#)

キュー・マネージャー定義コマンドに指定できるクラスター属性です。

[チャンネル定義コマンド](#)

チャンネル定義コマンドで指定できるクラスター属性

[キュー定義コマンド](#)

キュー定義コマンドに指定できるクラスター属性です。

[DISPLAY CLUSQMGR](#)

DISPLAY CLUSQMGR コマンドは、クラスター内にあるキュー・マネージャーに関するクラスター情報を表示する場合に使用します。

SUSPEND QMGR、RESUME QMGR、およびクラスター

SUSPEND QMGR および RESUME QMGR コマンドは、このキュー・マネージャーへのインバウンド・クラスター・アクティビティーを一時的に (例えば、このキュー・マネージャーの保守を行う前に) 削減し、後で復元するために使用します。

REFRESH CLUSTER

REFRESH CLUSTER コマンドは、クラスターに関するローカルに保持されたすべての情報を廃棄する場合にキュー・マネージャーから発行します。例外的な状況を除いて、このコマンドを使用する必要はほとんどありません。

RESET CLUSTER: クラスターからキュー・マネージャーを強制的に除去する

RESET CLUSTER コマンドは、例外的な状況にあるクラスターからキュー・マネージャーを強制的に削除する場合に使用します。

CLWLPRTY キュー属性

CLWLPRTY キュー属性は、クラスター・ワークロードの分散用にローカル・キュー、リモート・キュー、または別名キューの優先度を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLPRTY チャンネル属性は、宛先キューの優先順位を設定するために使用します。IBM MQ は、クラスターの宛先の優先度が最も低い宛先を選択する前に、優先度が最も高い宛先を選択します。優先度が同じである複数の宛先が存在する場合は、最も長期間未使用になっている宛先を選択します。

有効な宛先が 2 つある場合は、この属性を使用して、フェイルオーバーを許可できます。優先度が最も高いキュー・マネージャーは要求を受け取り、優先度が低いキュー・マネージャーは予備として機能します。優先度が最も高いキュー・マネージャーが失敗した場合、次に優先度が高い使用可能なキュー・マネージャーが引き継ぎます。

IBM MQ は、チャンネル状況を検査した後で、キュー・マネージャーの優先度を取得します。使用可能なキュー・マネージャーのみが選択候補になります。

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、クラスターのワークロード管理アルゴリズムの結果より優先度の低い状態もあります。これは、実際に、優先順位がより高い(プライマリー)宛先に対するチャンネルが開始されていても、優先順位がより低い(バックアップ)宛先が選択される可能性があることを意味します。

バックアップ宛先に対してメッセージが送信されないようにする必要がある場合は、CLWLPRTY を使用しないでください。個別のキューを使用するか、または CLWLRANK を使用してプライマリーからバックアップに手動で切り替えることを検討してください。

関連概念

クラスター・ワークロード管理アルゴリズム

ワークロード管理アルゴリズムは、ワークロード・バランシング属性と多数の規則を使用して、クラスター・キューに書き込むメッセージの最終宛先を選択します。

関連資料

CLWLRANK キュー属性

CLWLRANK キュー属性は、クラスター・ワークロードの分散用に、ローカル・キュー、リモート・キュー、または別名キューのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLUSEQ キュー属性

CLWLUSEQ キュー属性では、キューのローカル・インスタンスに宛先としてクラスター内の他のインスタンスよりも優先権を与えるかどうかを指定します。

CLWLUSEQ キュー・マネージャー属性

CLWLUSEQ キュー・マネージャー属性は、クラスター内のキューの他のインスタンスよりも、キューのローカル・インスタンスを宛先として優先するかどうかを指定します。この属性は、CLWLUSEQ キュー属性が QMGR に設定されている場合に適用されます。

CLWLMRUC キュー・マネージャー属性

CLWLMRUC キュー・マネージャー属性は、最後に選択されたチャンネルの数を設定します。クラスター・ワークロード管理アルゴリズムは CLWLMRUC を使用して、アクティブなアウトバウンド・クラスター・チャンネルの数を制限します。値は 1 から 999 999 999 の範囲でなければなりません。

CLWLPRTY チャンネル属性

CLWLPRTY チャンネル属性は、クラスター・ワークロード分散のために、チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK チャンネル属性

CLWLRANK チャンネル属性は、クラスター・ワークロード分散に使用するチャンネルのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLWGHT チャンネル属性

CLWLWGHT チャンネル属性は、クラスター・ワークロード分散用に、**CLUSDR** および **CLUSRCVR** チャンネルに適用される重みを指定します。この値は、1 から 99 の範囲でなければなりません。1 が最低、99 が最高の重みです。

NETPRTY チャンネル属性

NETPRTY チャンネル属性では、**CLUSRCVR** チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK キュー属性

CLWLRANK キュー属性は、クラスター・ワークロードの分散用に、ローカル・キュー、リモート・キュー、または別名キューのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLRANK キュー属性は、別のクラスター内のキュー・マネージャーに送信されるメッセージの最終宛先を制御する場合に使用します。**CLWLRANK** を設定すると、メッセージは、相互接続されたクラスターを経由してランクの高い宛先に向かう指定経路を取ります。

例えば、ゲートウェイの可用性を高めるために、同等に構成された 2 つのゲートウェイ・キュー・マネージャーを定義したとします。クラスターで定義されたローカル・キュー用に、ゲートウェイでクラスター別名キューを定義したとします。ローカル・キューが使用不可になった場合には、キューが再度使用可能になるまで、メッセージがゲートウェイのいずれかで保留されるようにするとします。キューをゲートウェイで保持するには、ゲートウェイのクラスター別名キューよりも高いランクでローカル・キューを定義する必要があります。

キュー別名と同じランクでローカル・キューを定義して、ローカル・キューが使用不可になった場合、メッセージはゲートウェイ間を移動します。ローカル・キューが使用不可であると検出されると、最初のゲートウェイ・キュー・マネージャーはもう一方のゲートウェイにメッセージを経路指定します。もう一方のゲートウェイはターゲット・ローカル・キューに再度メッセージを送信しようとしています。ローカル・キューが使用不可のままである場合、最初のゲートウェイにメッセージが戻されます。メッセージは、ターゲット・ローカル・キューが再度使用可能になるまで、ゲートウェイ間を移動し続けます。ローカル・キューに高位ランクを指定することによって、ローカル・キューが使用不可の場合でも、メッセージが下位ランクの宛先に転送されなくなります。

IBM MQ は、チャンネル状況を検査する前にキューのランクを取得します。チャンネル状況を検査する前にランクを取得するということは、アクセスできないキューでも選択の対象になることになります。この方法は、最終宛先が使用不可であったとしても、ネットワークを介してメッセージがルーティングされることを可能にします。

優先度属性を使用した場合、**IBM MQ** は使用可能な宛先から選択します。最高ランクの宛先がチャンネルを使用できない場合、メッセージは伝送キューに保持されます。チャンネルが使用可能になると解放されます。ランク順で次に使用可能な宛先にメッセージが送信されるわけではありません。

関連概念

クラスター・ワークロード管理アルゴリズム

ワークロード管理アルゴリズムは、ワークロード・balancing属性と多数の規則を使用して、クラスター・キューに書き込むメッセージの最終宛先を選択します。

関連資料

CLWLPRTY キュー属性

CLWLPRTY キュー属性は、クラスター・ワークロードの分散用にローカル・キュー、リモート・キュー、または別名キューの優先度を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLUSEQ キュー属性

CLWLUSEQ キュー属性では、キューのローカル・インスタンスに宛先としてクラスター内の他のインスタンスよりも優先権を与えるかどうかを指定します。

CLWLUSEQ キュー・マネージャー属性

CLWLUSEQ キュー・マネージャー属性は、クラスター内のキューの他のインスタンスよりも、キューのローカル・インスタンスを宛先として優先するかどうかを指定します。この属性は、CLWLUSEQ キュー属性が QMGR に設定されている場合に適用されます。

CLWLMRUC キュー・マネージャー属性

CLWLMRUC キュー・マネージャー属性は、最後に選択されたチャンネルの数を設定します。クラスター・ワークロード管理アルゴリズムは CLWLMRUC を使用して、アクティブなアウトバウンド・クラスター・チャンネルの数を制限します。値は 1 から 999 999 999 の範囲でなければなりません。

CLWLPRTY チャンネル属性

CLWLPRTY チャンネル属性は、クラスター・ワークロード分散のために、チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK チャンネル属性

CLWLRANK チャンネル属性は、クラスター・ワークロード分散に使用するチャンネルのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLWGHT チャンネル属性

CLWLWGHT チャンネル属性は、クラスター・ワークロード分散用に、CLUSDR および CLUSRCVR チャンネルに適用される重みを指定します。この値は、1 から 99 の範囲でなければなりません。1 が最低、99 が最高の重みです。

NETPRTY チャンネル属性

NETPRTY チャンネル属性では、CLUSRCVR チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLUSEQ キュー属性

CLWLUSEQ キュー属性では、キューのローカル・インスタンスに宛先としてクラスター内の他のインスタンスよりも優先権を与えるかどうかを指定します。

CLWLUSEQ キュー属性は、ローカル・キューにのみ有効です。これは、メッセージがアプリケーション、あるいはクラスター・チャンネルではないチャンネルによって書き込まれる場合にのみ適用されます。

LOCAL

ローカル・キューは、書き込み可能であれば、MQPUT の唯一のターゲットです。MQPUT の動作は、クラスター・ワークロード管理によって異なります。

QMGR

この動作は、CLWLUSEQ キュー・マネージャー属性で指定されたとおりになります。

ANY

MQPUT は、ワークロード分散用に、ローカル・キューをクラスター内のキューのその他のインスタンスと同じように見なします。

関連概念

クラスター・ワークロード管理アルゴリズム

ワークロード管理アルゴリズムは、ワークロード・balancing属性と多数の規則を使用して、クラスター・キューに書き込むメッセージの最終宛先を選択します。

関連資料

CLWLPRTY キュー属性

CLWLPRTY キュー属性は、クラスター・ワークロードの分散用にローカル・キュー、リモート・キュー、または別名キューの優先度を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK キュー属性

CLWLRANK キュー属性は、クラスター・ワークロードの分散用に、ローカル・キュー、リモート・キュー、または別名キューのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLUSEQ キュー・マネージャー属性

CLWLUSEQ キュー・マネージャー属性は、クラスター内のキューの他のインスタンスよりも、キューのローカル・インスタンスを宛先として優先するかどうかを指定します。この属性は、**CLWLUSEQ** キュー属性が **QMGR** に設定されている場合に適用されます。

CLWLMRUC キュー・マネージャー属性

CLWLMRUC キュー・マネージャー属性は、最後に選択されたチャンネルの数を設定します。クラスター・ワークロード管理アルゴリズムは **CLWLMRUC** を使用して、アクティブなアウトバウンド・クラスター・チャンネルの数を制限します。値は 1 から 999 999 999 の範囲でなければなりません。

CLWLPRTY チャンネル属性

CLWLPRTY チャンネル属性は、クラスター・ワークロード分散のために、チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK チャンネル属性

CLWLRANK チャンネル属性は、クラスター・ワークロード分散に使用するチャンネルのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLWGHT チャンネル属性

CLWLWGHT チャンネル属性は、クラスター・ワークロード分散用に、**CLUSDR** および **CLUSRCVR** チャンネルに適用される重みを指定します。この値は、1 から 99 の範囲でなければなりません。1 が最低、99 が最高の重みです。

NETPRTY チャンネル属性

NETPRTY チャンネル属性では、**CLUSRCVR** チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLUSEQ キュー・マネージャー属性

CLWLUSEQ キュー・マネージャー属性は、クラスター内のキューの他のインスタンスよりも、キューのローカル・インスタンスを宛先として優先するかどうかを指定します。この属性は、**CLWLUSEQ** キュー属性が **QMGR** に設定されている場合に適用されます。

CLWLUSEQ キュー属性は、ローカル・キューにのみ有効です。これは、メッセージがアプリケーション、あるいはクラスター・チャンネルではないチャンネルによって書き込まれる場合にのみ適用されます。

LOCAL

ローカル・キューは、**MQPUT** の唯一のターゲットです。LOCAL がデフォルトです。

ANY

MQPUT は、ワークロード分散用に、ローカル・キューをクラスター内のキューのその他のインスタンスと同じように見なします。

関連概念

クラスター・ワークロード管理アルゴリズム

ワークロード管理アルゴリズムは、ワークロード・balancing属性と多数の規則を使用して、クラスター・キューに書き込むメッセージの最終宛先を選択します。

関連資料

CLWLPRTY キュー属性

CLWLPRTY キュー属性は、クラスター・ワークロードの分散用にローカル・キュー、リモート・キュー、または別名キューの優先度を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK キュー属性

CLWLRANK キュー属性は、クラスター・ワークロードの分散用に、ローカル・キュー、リモート・キュー、または別名キューのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLUSEQ キュー属性

CLWLUSEQ キュー属性では、キューのローカル・インスタンスに宛先としてクラスター内の他のインスタンスよりも優先権を与えるかどうかを指定します。

CLWLMRUC キュー・マネージャー属性

CLWLMRUC キュー・マネージャー属性は、最後に選択されたチャンネルの数を設定します。クラスター・ワークロード管理アルゴリズムは CLWLMRUC を使用して、アクティブなアウトバウンド・クラスター・チャンネルの数を制限します。値は 1 から 999 999 999 の範囲でなければなりません。

CLWLPRTY チャンネル属性

CLWLPRTY チャンネル属性は、クラスター・ワークロード分散のために、チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK チャンネル属性

CLWLRANK チャンネル属性は、クラスター・ワークロード分散に使用するチャンネルのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLWGHT チャンネル属性

CLWLWGHT チャンネル属性は、クラスター・ワークロード分散用に、CLUSDR および CLUSRCVR チャンネルに適用される重みを指定します。この値は、1 から 99 の範囲でなければなりません。1 が最低、99 が最高の重みです。

NETPRTY チャンネル属性

NETPRTY チャンネル属性では、CLUSRCVR チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLMRUC キュー・マネージャー属性

CLWLMRUC キュー・マネージャー属性は、最後に選択されたチャンネルの数を設定します。クラスター・ワークロード管理アルゴリズムは CLWLMRUC を使用して、アクティブなアウトバウンド・クラスター・チャンネルの数を制限します。値は 1 から 999 999 999 の範囲でなければなりません。

初期デフォルト値は 999 999 999 です。

関連概念

クラスター・ワークロード管理アルゴリズム

ワークロード管理アルゴリズムは、ワークロード・balancing属性と多数の規則を使用して、クラスター・キューに書き込むメッセージの最終宛先を選択します。

関連資料

CLWLPRTY キュー属性

CLWLPRTY キュー属性は、クラスター・ワークロードの分散用にローカル・キュー、リモート・キュー、または別名キューの優先度を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK キュー属性

CLWLRANK キュー属性は、クラスター・ワークロードの分散用に、ローカル・キュー、リモート・キュー、または別名キューのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLUSEQ キュー属性

CLWLUSEQ キュー属性では、キューのローカル・インスタンスに宛先としてクラスター内の他のインスタンスよりも優先権を与えるかどうかを指定します。

CLWLUSEQ キュー・マネージャー属性

CLWLUSEQ キュー・マネージャー属性は、クラスター内のキューの他のインスタンスよりも、キューのローカル・インスタンスを宛先として優先するかどうかを指定します。この属性は、CLWLUSEQ キュー属性が QMGR に設定されている場合に適用されます。

CLWLPRTY チャンネル属性

CLWLPRTY チャンネル属性は、クラスター・ワークロード分散のために、チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK チャンネル属性

CLWLRANK チャンネル属性は、クラスター・ワークロード分散に使用するチャンネルのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLWGHT チャンネル属性

CLWLWGHT チャンネル属性は、クラスター・ワークロード分散用に、CLUSDR および CLUSRCVR チャンネルに適用される重みを指定します。この値は、1 から 99 の範囲でなければなりません。1 が最低、99 が最高の重みです。

NETPRTY チャンネル属性

NETPRTY チャンネル属性では、CLUSRCVR チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLPRTY チャンネル属性

CLWLPRTY チャンネル属性は、クラスター・ワークロード分散のために、チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLPRTY チャンネル属性は、使用可能なクラスター宛先の優先順位を設定するために使用します。IBM MQ は、クラスターの宛先の優先度が最も低い宛先を選択する前に、優先度が最も高い宛先を選択します。優先度が同じである複数の宛先が存在する場合は、最も長期間未使用になっている宛先を選択します。

有効な宛先が2つある場合は、この属性を使用して、フェイルオーバーを許可できます。メッセージは、最高の優先順位のチャンネルを持つキュー・マネージャーに移動されます。このキュー・マネージャーが使用不可になった場合、メッセージはその次に高い優先順位のキュー・マネージャーに移動されます。低い優先順位のキュー・マネージャーは、予備としての役割を果たします。

IBM MQ は、チャンネルに優先順位付けする前に、チャンネルの状況を検査します。使用可能なキュー・マネージャーのみが選択候補になります。

注:

- クラスター受信側チャンネルのターゲット・キュー・マネージャーで、この属性を指定します。一致するクラスター送信側チャンネルで指定したすべてのバランシングは、無視される可能性が高くなります。[クラスター・チャンネル](#)を参照してください。
- 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、クラスターのワークロード管理アルゴリズムの結果より優先度の低い状態もあります。これは、実際に、優先順位がより高い(プライマリー)宛先に対するチャンネルが開始されていても、優先順位がより低い(バックアップ)宛先が選択される可能性があることを意味します。
- バックアップ宛先に対してメッセージが送信されないようにする必要がある場合は、CLWLPRTY を使用しないでください。個別のキューを使用するか、または CLWLRANK を使用してプライマリーからバックアップに手動で切り替えることを検討してください。

関連概念

クラスター・ワークロード管理アルゴリズム

ワークロード管理アルゴリズムは、ワークロード・バランシング属性と多数の規則を使用して、クラスター・キューに書き込むメッセージの最終宛先を選択します。

関連資料

CLWLPRTY キュー属性

CLWLPRTY キュー属性は、クラスター・ワークロードの分散用にローカル・キュー、リモート・キュー、または別名キューの優先度を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK キュー属性

CLWLRANK キュー属性は、クラスター・ワークロードの分散用に、ローカル・キュー、リモート・キュー、または別名キューのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLUSEQ キュー属性

CLWLUSEQ キュー属性では、キューのローカル・インスタンスに宛先としてクラスター内の他のインスタンスよりも優先権を与えるかどうかを指定します。

CLWLUSEQ キュー・マネージャー属性

CLWLUSEQ キュー・マネージャー属性は、クラスター内のキューの他のインスタンスよりも、キューのローカル・インスタンスを宛先として優先するかどうかを指定します。この属性は、CLWLUSEQ キュー属性が QMGR に設定されている場合に適用されます。

CLWLMRUC キュー・マネージャー属性

CLWLMRUC キュー・マネージャー属性は、最後に選択されたチャンネルの数を設定します。クラスター・ワークロード管理アルゴリズムは CLWLMRUC を使用して、アクティブなアウトバウンド・クラスター・チャンネルの数を制限します。値は 1 から 999 999 999 の範囲でなければなりません。

CLWLRANK チャンネル属性

CLWLRANK チャンネル属性は、クラスター・ワークロード分散に使用するチャンネルのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLWGHT チャンネル属性

CLWLWGHT チャンネル属性は、クラスター・ワークロード分散用に、CLUSDR および CLUSRCVR チャンネルに適用される重みを指定します。この値は、1 から 99 の範囲でなければなりません。1 が最低、99 が最高の重みです。

NETPRTY チャンネル属性

NETPRTY チャンネル属性では、CLUSRCVR チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK チャンネル属性

CLWLRANK チャンネル属性は、クラスター・ワークロード分散に使用するチャンネルのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLRANK チャンネル属性は、別のクラスター内のキュー・マネージャーに送信されるメッセージの最終宛先を制御する場合に使用します。クラスターの交点にあるゲートウェイ・キュー・マネージャーにキュー・マネージャーを接続するチャンネルのランクを設定することによって、最終宛先の選択を制御します。

CLWLRANK を設定すると、メッセージは、相互接続されたクラスターを經由してランクの高い宛先に向かう指定経路を取ります。例えば、ランク 1 とランク 2 のチャンネルを使用する 2 つのキュー・マネージャーのどちらかにメッセージを送信できるゲートウェイ・キュー・マネージャーにメッセージが到着したとします。これらのメッセージは、最高ランクのチャンネル (この場合はランク 2 のキュー・マネージャーへのチャンネル) で接続されたキュー・マネージャーに自動的に送信されます。

IBM MQ は、チャンネル状況を検査する前にチャンネルのランクを取得します。チャンネル状況を検査する前にランクを取得するということは、アクセスできないチャンネルでも選択の対象になることを意味します。この方法は、最終宛先が使用不可であったとしても、ネットワークを介してメッセージがルーティングされることを可能にします。

注:

- クラスター受信側チャンネルのターゲット・キュー・マネージャーで、この属性を指定します。一致するクラスター送信側チャンネルで指定したすべてのバランシングは、無視される可能性が高くなります。[クラスター・チャンネル](#)を参照してください。
- 優先順位属性 **CLWLPRTY** も使用した場合、IBM MQ は使用可能な宛先の中から選択します。最高ランクの宛先がチャンネルを使用できない場合、メッセージは伝送キューに保持されます。チャンネルが使用可能になると解放されます。ランク順で次に使用可能な宛先にメッセージが送信されるわけではありません。

関連概念

[クラスター・ワークロード管理アルゴリズム](#)

ワークロード管理アルゴリズムは、ワークロード・バランシング属性と多数の規則を使用して、クラスター・キューに書き込むメッセージの最終宛先を選択します。

関連資料

[CLWLPRTY キュー属性](#)

CLWLPRTY キュー属性は、クラスター・ワークロードの分散用にローカル・キュー、リモート・キュー、または別名キューの優先度を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK キュー属性

CLWLRANK キュー属性は、クラスター・ワークロードの分散用に、ローカル・キュー、リモート・キュー、または別名キューのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLUSEQ キュー属性

CLWLUSEQ キュー属性では、キューのローカル・インスタンスに宛先としてクラスター内の他のインスタンスよりも優先権を与えるかどうかを指定します。

CLWLUSEQ キュー・マネージャー属性

CLWLUSEQ キュー・マネージャー属性は、クラスター内のキューの他のインスタンスよりも、キューのローカル・インスタンスを宛先として優先するかどうかを指定します。この属性は、CLWLUSEQ キュー属性が QMGR に設定されている場合に適用されます。

CLWLMRUC キュー・マネージャー属性

CLWLMRUC キュー・マネージャー属性は、最後に選択されたチャンネルの数を設定します。クラスター・ワークロード管理アルゴリズムは CLWLMRUC を使用して、アクティブなアウトバウンド・クラスター・チャンネルの数を制限します。値は 1 から 999 999 999 の範囲でなければなりません。

CLWLPRTY チャンネル属性

CLWLPRTY チャンネル属性は、クラスター・ワークロード分散のために、チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLWGHT チャンネル属性

CLWLWGHT チャンネル属性は、クラスター・ワークロード分散用に、CLUSDR および CLUSRCVR チャンネルに適用される重みを指定します。この値は、1 から 99 の範囲でなければなりません。1 が最低、99 が最高の重みです。

NETPRTY チャンネル属性

NETPRTY チャンネル属性では、CLUSRCVR チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLWGHT チャンネル属性

CLWLWGHT チャンネル属性は、クラスター・ワークロード分散用に、CLUSDR および CLUSRCVR チャンネルに適用される重みを指定します。この値は、1 から 99 の範囲でなければなりません。1 が最低、99 が最高の重みです。

CLWLWGHT を使用して、より多くの処理能力があるサーバーに、より多くのメッセージを送信します。チャンネルの重みが高いほど、そのチャンネルを通して多くのメッセージが送信されます。

注:

- クラスター受信側チャンネルのターゲット・キュー・マネージャーで、この属性を指定します。一致するクラスター送信側チャンネルで指定したすべてのバランシングは、無視される可能性が高くなります。[クラスター・チャンネル](#)を参照してください。
- いずれかのチャンネルで CLWLWGHT がデフォルトの 50 から変更された場合、ワークロード・バランシングはいずれかのクラスター・キューに送信されたメッセージのために各チャンネルが合計何回選択されたかに依存します。詳しくは、[165 ページ](#)の『[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。

関連概念

クラスター・ワークロード管理アルゴリズム

ワークロード管理アルゴリズムは、ワークロード・バランシング属性と多数の規則を使用して、クラスター・キューに書き込むメッセージの最終宛先を選択します。

関連資料

CLWLPRTY キュー属性

CLWLPRTY キュー属性は、クラスター・ワークロードの分散用にローカル・キュー、リモート・キュー、または別名キューの優先度を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK キュー属性

CLWLRANK キュー属性は、クラスター・ワークロードの分散用に、ローカル・キュー、リモート・キュー、または別名キューのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLUSEQ キュー属性

CLWLUSEQ キュー属性では、キューのローカル・インスタンスに宛先としてクラスター内の他のインスタンスよりも優先権を与えるかどうかを指定します。

CLWLUSEQ キュー・マネージャー属性

CLWLUSEQ キュー・マネージャー属性は、クラスター内のキューの他のインスタンスよりも、キューのローカル・インスタンスを宛先として優先するかどうかを指定します。この属性は、CLWLUSEQ キュー属性が QMGR に設定されている場合に適用されます。

CLWLMRUC キュー・マネージャー属性

CLWLMRUC キュー・マネージャー属性は、最後に選択されたチャンネルの数を設定します。クラスター・ワークロード管理アルゴリズムは CLWLMRUC を使用して、アクティブなアウトバウンド・クラスター・チャンネルの数を制限します。値は 1 から 999 999 999 の範囲でなければなりません。

CLWLPRTY チャンネル属性

CLWLPRTY チャンネル属性は、クラスター・ワークロード分散のために、チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK チャンネル属性

CLWLRANK チャンネル属性は、クラスター・ワークロード分散に使用するチャンネルのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

NETPRTY チャンネル属性

NETPRTY チャンネル属性では、CLUSRCVR チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

NETPRTY チャンネル属性

NETPRTY チャンネル属性では、CLUSRCVR チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

NETPRTY 属性は、あるネットワークを主ネットワークにし、別のネットワークをバックアップ・ネットワークにする場合に使用します。同じランクの 1 組みのチャンネルが存在すると仮定すると、複数のパスが使用可能な場合、クラスター化では最高の優先順位を持つパスを選択します。

NETPRTY チャンネル属性を使用する典型的な例としては、異なるコストまたは速度を持ち、また同じ複数の宛先を接続する 2 つのネットワークを区別するケースです。

注: クラスター受信側チャンネルのターゲット・キュー・マネージャーで、この属性を指定します。一致するクラスター送信側チャンネルで指定したすべてのバランシングは、無視される可能性が高くなります。[クラスター・チャンネル](#)を参照してください。

関連概念

クラスター・ワークロード管理アルゴリズム

ワークロード管理アルゴリズムは、ワークロード・バランシング属性と多数の規則を使用して、クラスター・キューに書き込むメッセージの最終宛先を選択します。

関連資料

CLWLPRTY キュー属性

CLWLPRTY キュー属性は、クラスター・ワークロードの分散用にローカル・キュー、リモート・キュー、または別名キューの優先度を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK キュー属性

CLWLRANK キュー属性は、クラスター・ワークロードの分散用に、ローカル・キュー、リモート・キュー、または別名キューのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLUSEQ キュー属性

CLWLUSEQ キュー属性では、キューのローカル・インスタンスに宛先としてクラスター内の他のインスタンスよりも優先権を与えるかどうかを指定します。

CLWLUSEQ キュー・マネージャー属性

CLWLUSEQ キュー・マネージャー属性は、クラスター内のキューの他のインスタンスよりも、キューのローカル・インスタンスを宛先として優先するかどうかを指定します。この属性は、CLWLUSEQ キュー属性が QMGR に設定されている場合に適用されます。

CLWLMRUC キュー・マネージャー属性

CLWLMRUC キュー・マネージャー属性は、最後に選択されたチャンネルの数を設定します。クラスター・ワークロード管理アルゴリズムは CLWLMRUC を使用して、アクティブなアウトバウンド・クラスター・チャンネルの数を制限します。値は 1 から 999 999 999 の範囲でなければなりません。

CLWLPRTY チャンネル属性

CLWLPRTY チャンネル属性は、クラスター・ワークロード分散のために、チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK チャンネル属性

CLWLRANK チャンネル属性は、クラスター・ワークロード分散に使用するチャンネルのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLWGHT チャンネル属性

CLWLWGHT チャンネル属性は、クラスター・ワークロード分散用に、CLUSDR および CLUSRCVR チャンネルに適用される重みを指定します。この値は、1 から 99 の範囲でなければなりません。1 が最低、99 が最高の重みです。

クラスター・ワークロード管理アルゴリズム

ワークロード管理アルゴリズムは、ワークロード・バランシング属性と多数の規則を使用して、クラスター・キューに書き込むメッセージの最終宛先を選択します。

ワークロード管理アルゴリズムは宛先の選択が必要になるたびに毎回実行されます。

- MQOO_BIND_ON_OPEN オプションを使用してクラスター・キューが開かれた時点で使用されます。
- MQOO_BIND_NOT_FIXED を指定してクラスター・キューが開かれた場合に、メッセージがそのクラスター・キューに書き込まれるたびに使用されます。
- MQOO_BIND_ON_GROUP を使用してクラスター・キューが開かれた場合に、新規メッセージ・グループが開始されるたびに使用されます。
- トピック・ホスト・ルーティング では、メッセージがクラスター・トピックにパブリッシュされるたびに使用されます。ローカル・キュー・マネージャーがそのトピックのホストではない場合、このアルゴリズムを使用して、メッセージのルーティング経路となるホスト・キュー・マネージャーが選択されません。

以下のセクションでは、クラスター・キューに書き込むメッセージの最終宛先を決定する際に使用されるワークロード管理アルゴリズムについて説明します。これらの規則は、キュー、キュー・マネージャー、およびチャンネルの以下の属性に適用される設定の影響を受けます。

「キュー」	キュー・マネージャー	チャンネル
<ul style="list-style-type: none"> • <u>CLWLPRTY</u>¹ • <u>CLWLRANK</u>¹ • <u>CLWLUSEQ</u>¹ • PUT/PUB 	<ul style="list-style-type: none"> • <u>CLWLUSEQ</u>¹ • <u>CLWLMRUC</u> 	<ul style="list-style-type: none"> • <u>CLWLPRTY</u> • <u>CLWLRANK</u> • <u>CLWLWGHT</u> • <u>NETPRTY</u>

¹ この属性は、クラスター・キューを選択する場合にのみ該当します。トピックを選択する場合は該当しません。

最初に、キュー・マネージャーは以下の2つの手順から有効な宛先のリストを構築します。

- ターゲットの `ObjectName` および `ObjectQmgrName` と、このキュー・マネージャーと同じクラスターで共有されるキュー・マネージャー別名定義を突き合わせます。
- `ObjectName` という名前のキューをホストしているキュー・マネージャーへの固有の経路、つまりチャネルのうち、このキュー・マネージャーがメンバーになっているクラスター内にあるものを検索します。

このアルゴリズムでは、以下の各ルールを順次適用して有効な宛先のリストから宛先を除去していきます。

1. ローカル・キュー・マネージャーとクラスターを共有しないキューまたはトピックまたはリモート CLUSRCVR チャネルのリモート・インスタンスは除去されます。
2. キューまたはトピック名が指定されている場合、キューまたはトピックと同じクラスター内がないリモート CLUSRCVR チャネルは除去されます。

注：この段階で残っているすべてのキュー、トピック、およびチャネルは、構成されていれば、クラスター・ワークロード出口で使用可能になります。

3. キュー・マネージャーに対するチャネルまたはキュー・マネージャー別名のうち、`CLWLRANK` が残りのすべてのチャネルまたはキュー・マネージャー別名の最大ランクを下回るものはすべて除去されます。
4. `CLWLRANK` が残りのすべてのキューの最大ランクを下回るキュー (キュー・マネージャー別名ではない) は、すべて除去されます。
5. キュー、トピック、またはキュー・マネージャーの別名のインスタンスが複数残っている場合、パブリッシュ書き込みが有効なものがあれば、書き込みが無効にされたものはすべて除去されます。

注：書き込みが無効にされたインスタンスのみが残っている場合、照会操作のみが成功し、他のすべての操作は `MQRC_CLUSTER_PUT_INHIBITED` で失敗します。

6. キューを選択するときに、結果のキューの集合にキューのローカル・インスタンスが含まれている場合は、通常、そのローカル・インスタンスが使用されます。以下の条件のいずれかが満たされる場合は、キューのローカル・インスタンスが使用されます。

- キューの「キューの使用」属性 `CLWLUSEQ` が `LOCAL` に設定されている。

- 次の記述の両方に当てはまる場合:

- キューの「キューの使用」属性 `CLWLUSEQ` が `QMGR` に設定されている。
- キュー・マネージャーの「キューの使用」属性 `CLWLUSEQ` が `LOCAL` に設定されている。

- メッセージは、ローカル・アプリケーションによって書き込まれずに、クラスター・チャネルを介して受信される。

- `CLWLUSEQ(ANY)` を使用してローカルに定義されたキュー、またはキュー・マネージャーから同じ設定を継承しているローカルに定義されたキューの場合、より広範囲の適用条件において、次の事項が該当します。

- ローカル・キューは、このキューと同じクラスター内にあるローカル定義の CLUSRCVR チャネルの `status` に基づいて選択されます。この状況が、メッセージを同名のリモート定義キューに送信する CLUSSDR チャネルの状況と比較されます。

例えば、キューと同じクラスター内に1つの CLUSRCVR が存在するとします。この CLUSRCVR の状況は `STOPPING` であり、クラスター内の同名のキューの状況が `RUNNING` または `INACTIVE` であるとします。この場合、リモート・チャネルが選択され、ローカル・キューは使用されません。

- ローカル・キューは、同名のリモート定義キューにメッセージを送信する同じ状況の CLUSSDR チャネルと比較した CLUSRCVR チャネルの `number` に基づいて選択されます。

例えば、キューと同じクラスターに4つの CLUSRCVR チャネルがあり、また1つの CLUSSDR チャネルがあるとします。すべてのチャネルの状況が同一 (`INACTIVE` または `RUNNING`) であるとします。この場合、選択可能な5つのチャネルと、2つのキュー・インスタンスが存在します。メッセージの4/5 (80%) はローカル・キューに入れられます。

7. 複数のキュー・マネージャーが残っている場合、中断されていないキュー・マネージャーがあれば、中断されているものはすべて除去されます。

8. キューまたはトピックの複数のリモート・インスタンスが残っている場合は、非アクティブまたは実行状態のチャンネルすべてが含まれます。以下の状態定数がリストされます。
 - MQCHS_INACTIVE
 - MQCHS_RUNNING
9. キューまたはトピックのリモート・インスタンスが1つも残っていない場合は、バインディング、初期化、開始、または停止のいずれかの処理中状態であるすべてのチャンネルが含まれます。以下の状態定数がリストされます。
 - MQCHS_BINDING
 - MQCHS_INITIALIZING
 - MQCHS_STARTING
 - MQCHS_STOPPING
10. キューまたはトピックのリモート・インスタンスが1つも残っていない場合は、再試行中のすべてのチャンネルが含まれます。以下の状態定数がリストされます。
 - MQCHS_RETRYING
11. キューまたはトピックのリモート・インスタンスが1つも残っていない場合は、要求中、休止、または停止状態のすべてのチャンネルが含まれます。以下の状態定数がリストされます。
 - MQCHS_REQUESTING
 - MQCHS_PAUSED
 - MQCHS_STOPPED
 - MQCHS_SWITCHING
12. いずれかのキュー・マネージャーでキューまたはトピックの複数のリモート・インスタンスが残っている場合は、各キュー・マネージャーで最高の NETPRTY 値を持つチャンネルが選択されます。
13. 優先度 (CLWLPRTY) が最も高いチャンネルと別名以外の、残りのチャンネルとキュー・マネージャー別名がすべて除去されます。キュー・マネージャー別名が残っている場合は、キュー・マネージャーに対するチャンネルは保持されます。
14. キューが選択される場合は、次のようになります。
 - 優先度 (CLWLPRTY) が最も高いキュー以外のキューはすべて除去され、チャンネルは保持されます。
15. 残りのチャンネルは、最も低い MQWDR.DestSeqNumber の値を持つチャンネルを除去することにより、最近使用されたチャンネル CLWLMRUC の最大許容数を超えないように削減されます。

注：内部クラスター制御メッセージは、同じクラスター・ワークロード・アルゴリズムを使用して送信されます (適切な場合)。

有効な宛先リストが算出された後、以下のロジックを使用して、それらの宛先の間でメッセージのワークロード・バランシングが行われます。

- 宛先のリモート・インスタンスが複数残っていて、その宛先に対するすべてのチャンネルの CLWLWGHT がデフォルトの設定値 50 に設定されている場合は、最も使用頻度の少ないチャンネルが選択されます。これは、複数のリモート・インスタンスが存在する場合のラウンドロビン方式のワークロード・バランシングとほぼ同一です。
- 宛先のリモート・インスタンスが複数残っていて、それらのキューに対する1つ以上のチャンネルの CLWLWGHT がデフォルト以外の設定値に設定されている場合 (それらすべてがデフォルト以外の同じ値に設定されている場合も含む)、ルーティングは、各チャンネルの相対ウェイトと、各チャンネルがこれまでメッセージの送信時に選択された合計回数によって決まります。
- 複数のインスタンスが存在する単一クラスター・キューのメッセージ分散を観察したときに、一部のキュー・インスタンスの間で分散が平衡化されなくなるように見える場合があります。この原因は、そのキューのメッセージ・トラフィックだけではなく、このキュー・マネージャーの各クラスター送信側チャンネルの過去の使用も平衡化されているからです。この動作が望ましくない場合は、以下の手順のいずれかを実行してください。
 - 均一な分散が必要な場合は、すべてのクラスター受信側チャンネルの CLWLWGHT を 50 に設定します。

- または、特定のキュー・インスタンスに他とは異なるウェイトを設定する必要がある場合は、それらのキューを、専用のクラスター受信側チャンネルが定義された専用のクラスターに定義します。この操作により、これらのキューのワークロード・バランシングは、クラスター内の他のものとは分離されます。
- 使用可能なクラスター受信側チャンネルのクラスター・ワークロード属性が変更されたり、クラスター受信側チャンネルの状況が使用可能になったりすると、チャンネルの平衡化のために使用される履歴データはリセットされます。手動で定義されたクラスター送信側チャンネルのワークロード属性を変更しても、履歴データはリセットされません。
- クラスター・ワークロード出口のロジックについて検討している場合、選択されるチャンネルは、MQWDR.DestSeqFactor が最も小さいチャンネルです。チャンネルが選択されるたびに、この値は約 1000/CLWLWGHT 増分されます。最も小さい値を持つチャンネルが複数ある場合は、MQWDR.DestSeqNumber 値が最も小さいチャンネルのいずれかが選択されます。

クラスターの管理や保守の際には、メッセージがチャンネルを介して流れるため、ユーザー・メッセージの分散は必ずしも正確ではありません。その結果、ユーザー・メッセージが不均一に分散され、一定になるまでに時間がかかることがあります。管理とユーザー・メッセージが混在しているので、ワークロード・バランシング中にメッセージが正確に分散されるとは限りません。

関連資料

CLWLPRTY キュー属性

CLWLPRTY キュー属性は、クラスター・ワークロードの分散用にローカル・キュー、リモート・キュー、または別名キューの優先度を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK キュー属性

CLWLRANK キュー属性は、クラスター・ワークロードの分散用に、ローカル・キュー、リモート・キュー、または別名キューのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLUSEQ キュー属性

CLWLUSEQ キュー属性では、キューのローカル・インスタンスに宛先としてクラスター内の他のインスタンスよりも優先権を与えるかどうかを指定します。

CLWLUSEQ キュー・マネージャー属性

CLWLUSEQ キュー・マネージャー属性は、クラスター内のキューの他のインスタンスよりも、キューのローカル・インスタンスを宛先として優先するかどうかを指定します。この属性は、CLWLUSEQ キュー属性が QMGR に設定されている場合に適用されます。

CLWLMRUC キュー・マネージャー属性

CLWLMRUC キュー・マネージャー属性は、最後に選択されたチャンネルの数を設定します。クラスター・ワークロード管理アルゴリズムは CLWLMRUC を使用して、アクティブなアウトバウンド・クラスター・チャンネルの数を制限します。値は 1 から 999 999 999 の範囲でなければなりません。

CLWLPRTY チャンネル属性

CLWLPRTY チャンネル属性は、クラスター・ワークロード分散のために、チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

CLWLRANK チャンネル属性

CLWLRANK チャンネル属性は、クラスター・ワークロード分散に使用するチャンネルのランクを指定します。値は 0 から 9 の範囲でなければなりません (0 が最低ランク、9 が最高ランク)。

CLWLWGHT チャンネル属性

CLWLWGHT チャンネル属性は、クラスター・ワークロード分散用に、CLUSDR および CLUSRCVR チャンネルに適用される重みを指定します。この値は、1 から 99 の範囲でなければなりません。1 が最低、99 が最高の重みです。

NETPRTY チャンネル属性

NETPRTY チャンネル属性では、CLUSRCVR チャンネルの優先順位を指定します。この値は、0 から 9 の範囲でなければなりません。0 が最低、9 が最高の優先度です。

z/OS における CLUSTER コマンドの非同期の動作

z/OS におけるクラスター・コマンドのコマンド発行者は、コマンドが送信されたことの確認を受け取りませんが、それは正常に完了したということではありません。

REFRESH CLUSTER と RESET CLUSTER の場合、要求が送信されたことを示すメッセージの CSQM130I がコマンド発行者に送信されます。このメッセージには、メッセージ CSQ9022I が続きます。これは、要求が送信されたという点でコマンドが正常に完了したことを示しています。これは、クラスター要求が正常に完了したことを示すものではありません。

エラーはすべて、チャンネル開始プログラムが稼働しているシステムの z/OS コンソールに報告されます。コマンド発行側にはエラーは送られません。

非同期の動作は、CHANNEL コマンドとは対照的です。チャンネル・コマンドが受け付けられたことを示すメッセージは、即座に発行されます。一定の時間が経過した時点でコマンドが完了すると、正常な完了または異常な完了のいずれかを示すメッセージがコマンド発行者に送信されます。

関連概念

クラスターでのワークロード・バランシング

1つのクラスターに同じキューのインスタンスが複数含まれる場合、IBM MQ はメッセージを送付するキュー・マネージャーを選択します。また、クラスター・ワークロード管理アルゴリズムと、クラスター・ワークロード固有のいくつかの属性を使用して、使用するのに最適なキュー・マネージャーを判別します。

関連資料

キュー・マネージャー定義コマンド

キュー・マネージャー定義コマンドに指定できるクラスター属性です。

チャンネル定義コマンド

チャンネル定義コマンドで指定できるクラスター属性

キュー定義コマンド

キュー定義コマンドに指定できるクラスター属性です。

DISPLAY CLUSQMGR

DISPLAY CLUSQMGR コマンドは、クラスター内にあるキュー・マネージャーに関するクラスター情報を表示する場合に使用します。

SUSPEND QMGR、RESUME QMGR、およびクラスター

SUSPEND QMGR および RESUME QMGR コマンドは、このキュー・マネージャーへのインバウンド・クラスター・アクティビティーを一時的に (例えば、このキュー・マネージャーの保守を行う前に) 削減し、後で復元するために使用します。

REFRESH CLUSTER

REFRESH CLUSTER コマンドは、クラスターに関するローカルに保持されたすべての情報を廃棄する場合にキュー・マネージャーから発行します。例外的な状況を除いて、このコマンドを使用する必要はほとんどありません。

RESET CLUSTER: クラスターからキュー・マネージャーを強制的に除去する

RESET CLUSTER コマンドは、例外的な状況にあるクラスターからキュー・マネージャーを強制的に削除する場合に使用します。

関連情報

分散ネットワークに対する非同期コマンドが終了したことの確認

チャンネル・プログラム

このセクションでは、チャンネルで使用可能なさまざまなタイプのチャンネル・プログラム (MCA) について検討します。

MCA の名前が、以下の表に示されています。

表 33. Windows、UNIX and Linux システム用のチャンネル・プログラム

プログラム名	接続の方向	通信
amqrmppa		任意
runmqslr	インバウンド	任意
amqcrs6a	インバウンド	LU 6.2
amqcrsta	インバウンド	TCP
runmqchl	アウトバウンド	任意
runmqchi	アウトバウンド	任意

runmqslr (IBM MQ リスナーの実行)、runmqchl (IBM MQ チャンネルの実行)、および runmqchi (IBM MQ チャンネル・イニシエーターの実行) は、コマンド行で入力できる制御コマンドです。

amqcrsta は、UNIX and Linux システム上のリスナーが開始されていない TCP チャンネルに対して、inetd を使用して呼び出されます。

LU6.2 を使用する場合、amqcrs6a はトランザクション・プログラムとして起動されます。

環境変数

お客様が使用するためのサーバー環境変数およびクライアント環境変数のリスト。

使用例

- Linux
UNIX
 UNIX and Linux システムでは、`export [environment variable]=value` を使用します。
- Windows
 Windows システムでは、`Set [environment variable]=value` を使用します。
- IBM i
 IBM i システムでは、`ADDENVVAR ENVVAR(environment variable) VALUE(xx)` を使用します。

AMQ_BAD_COMMS_DATA_FDCS

この環境変数は、任意の値に設定すると有効になります。

IBM MQ が TCP/IP を介してホストから受信するデータの形式が正しくない場合 (例えば、ネットワーク・クライアントが IBM MQ リスナー・ポートに接続していて、サポートされないアプリケーション・プロトコルと通信しようとした場合)、キュー・マネージャーは **AMQ9207E** エラー・メッセージをキュー・マネージャーのエラー・ログに書き込みます。IBM MQ リスナーは、キュー・マネージャー・メッセージ・チャンネル・エージェント (MCA) および MQI クライアント・アプリケーション、JMS および XMS クライアント・アプリケーションからの TCP/IP 接続をサポートします。

注: IBM MQ リスナーは、AMQP および MQTT クライアントによって使用されるアプリケーション・プロトコルをサポートしません。これらのクライアントは、代わりに、該当する AMQP チャンネルまたは MQXR テレメトリー・サービスで構成されたネットワーク・ポートに接続する必要があります。

IBM MQ が受信した無効データを含む障害データ・キャプチャー機能 (FDC) レコードも書き込まれる可能性があります。ただし、これがリモート・サイドとの会話の開始であり、フォーマットが HTTP Web ブラウザーからの GET 要求などの単純な既知のフォーマットである場合、FFST ファイルは生成されません。これをオーバーライドして、単純な既知の形式を含むすべての不良データに対して FFST ファイルが書き込まれるようにするには、**AMQ_BAD_COMMS_DATA_FDCS** 環境変数を任意の値 (例えば、TRUE) に設定して、キュー・マネージャーを再始動します。

V9.0.0.9 AMQ_NO_BAD_COMMS_DATA_FDCS

この環境変数は、任意の値に設定すると有効になります。

IBM MQ 以外のクライアントを IBM MQ TCP/IP リスナーに接続しようとしたときに IBM MQ が初期データ伝送を認識しない場合、キュー・マネージャーは **AMQ9207E** エラー・メッセージをキュー・マネ

ージャーのエラー・ログに書き込みます。障害データ・キャプチャー機能 (FDC) レコードも書き込まれます。 **AMQ_NO_BAD_COMMS_DATA_FDCS** 環境変数を使用して、これらの診断ファイルの生成を抑止することができます。 **AMQ_NO_BAD_COMMS_DATA_FDCS** が任意の値 (例えば、TRUE) に設定されている場合、これは、初期通信フローで AMQ9207E エラー・メッセージを報告するときに FFST を生成しないように IBM MQ に指示します。有効にするには、キュー・マネージャーおよびリスナー・プロセスを開始する前に環境変数を設定する必要があります。

クライアントが有効な IBM MQ プロトコル・フローをキュー・マネージャーに送信してから、無効なデータを送信する場合は、引き続き FDC が生成されます。これは、さらなる調査を必要とするクライアントの問題を示しています。

V 9.0.0.6 Multi **AMQ_CONVEBCDICNEWLINE**

IBM MQ 9.0.0 Fix Pack 6 から、この環境変数を使用して、IBM MQ が EBCDIC NL 文字を ASCII フォーマットに変換する方法を指定できるようになりました。この環境変数は、mqs.ini の **ConvEBCDICNewline** 属性と同じ値、つまり NL_TO_LF、TABLE、または ISO を取ります (すべての キュー・マネージャー を参照)。例えば、mqs.ini ファイルを使用できない状況でも、**ConvEBCDICNewline** スタンザ属性の代わりに **AMQ_CONVEBCDICNEWLINE** 環境変数を使用すれば、クライアント・サイドで **ConvEBCDICNewline** 機能を利用できます。

スタンザ属性と環境変数の両方が設定されている場合は、スタンザ属性のほうが優先されます。

V 9.0.3 **AMQ_DIAGNOSTIC_MSG_SEVERITY**

IBM MQ 9.0.3 以降、IBM MQ プロセスに対して環境変数 **AMQ_DIAGNOSTIC_MSG_SEVERITY** が 1 に設定されている場合、IBM MQ プロセスがメッセージをエラー・ログまたはコンソールに書き込むと、メッセージ重大度が単一の英大文字としてメッセージ番号に付加されます。

V 9.0.4 IBM MQ 9.0.4 以降、**AMQ_DIAGNOSTIC_MSG_SEVERITY** が有効にする動作はデフォルトで設定されています。この動作は環境変数を 0 に設定してオフにできます。

z/OS LTS **AMQ_DMPMQCFG_QSGDISP_DEFAULT**

IBM MQ 9.0.0 Fix Pack 9 以降、**dmpmqcfg** コマンドを使用して、キュー共有グループのキュー・マネージャーの属性指定を照会できます。

この環境変数で利用できる値の詳細については、「**dmpmqcfg**」を参照してください。

V 9.0.0.9 **AMQ_LDAP_TRACE**

IBM MQ 9.0.0 Fix Pack 9 以降、環境変数 **AMQ_LDAP_TRACE** が非 NULL 値に設定されている場合、キュー・マネージャーを停止または開始せずに LDAP クライアント・トレースのオン/オフを切り替えることができます。詳しくは、LDAP クライアント・ライブラリー・コードの動的トレースの有効化を参照してください。

AMQ_MQS_INI_LOCATION

UNIX and Linux システムでは、この変数で mqs.ini ファイルの場所を設定して、mqs.ini ファイルに使用する場所を変更することができます。この変数は、システム・レベルで設定する必要があります。

AMQ_NO_IPV6

この環境変数は、任意の値に設定すると有効になります。この環境変数を設定すると、接続の試行中に IPv6 が使用不可になります。

AMQ_REVERSE_COMMIT_ORDER

この変数は、XA トランザクションにおいて、IBM MQ キュー・マネージャーによる変更が、対応するデータベース更新が完了した後にコミットされるようにキュー・マネージャーを構成します。

分離レベルのトピックで説明しているシナリオを読んで理解するまでは、**AMQ_REVERSE_COMMIT_ORDER** を設定しないでください。

V 9.0.2 V 9.0.0.1

AMQ_SSL_ALLOW_DEFAULT_CERT

IBM MQ 9.0.0 Fix Pack 1 および IBM MQ 9.0.2 以降は、**AMQ_SSL_ALLOW_DEFAULT_CERT** 環境変数が設定されていない場合、アプリケーションがクライアント鍵ストア内の個人証明書を使用してキュー・マネージャーに接続するためには、その証明書に **ibmwebsphermquserid** のラベル名が含まれてい

なければなりません。AMQ_SSL_ALLOW_DEFAULT_CERT 環境変数が設定されているとき、証明書に `ibmwebsphermquserid` のラベル名は必要ありません。つまり、キュー・マネージャーへの接続に使用される証明書は、鍵リポジトリにデフォルト証明書が存在し、鍵リポジトリに `ibmwebsphermquserid` という接頭部の個人証明書が含まれていない場合は、デフォルト証明書にすることができます。詳しくは、技術情報 [Specifying the userid in the SSL certificate label for an MQ client](#) を参照してください。

値 1 は、デフォルトの証明書の使用を有効にします。

IBM MQ 8.0 以降、アプリケーションは、AMQ_SSL_ALLOW_DEFAULT_CERT 環境変数を使用する代わりに、`mqclient.ini` ファイルの SSL スタンザにある **CertificateLabel** 設定を使用できます。詳しくは、[デジタル証明書ラベルの要件に関する説明およびクライアント構成ファイルの SSL スタンザ](#) を参照してください。

V 9.0.0.2

V 9.0.4

AMQ_SSL_LDAP_SERVER_VERSION

IBM MQ 9.0.0 Fix Pack 2 および IBM MQ 9.0.4 以降、特定のバージョンの LDAP プロトコルの使用が CRL サーバーによって要求される場合に、この変数を使用することにより、IBM MQ 暗号コンポーネントで LDAP v2 と LDAP v3 のどちらを使用するかを設定できます。

キュー・マネージャーまたはチャネルの開始に使用される環境で適切な値を、この変数に設定します。LDAP v2 が使用されるように要求するには、`AMQ_SSL_LDAP_SERVER_VERSION=2` と設定します。LDAP v3 が使用されるように要求するには、`AMQ_SSL_LDAP_SERVER_VERSION=3` と設定します。

この変数は、ユーザー認証またはユーザー許可のために IBM MQ キュー・マネージャーによって設定された LDAP 接続には影響を与えません。

GMQ_MQ_LIB

IBM MQ MQI client と IBM MQ サーバーの両方がシステムにインストールされている場合、デフォルトでは MQAX アプリケーションがサーバーに対して実行されます。MQAX をクライアントに対して実行する場合は、`GMQ_MQ_LIB` 環境変数でクライアント・バイnding・ライブラリーを指定する必要があります。例えば、`GMQ_MQ_LIB=mqic.dll` のように設定します。クライアントだけのインストール環境の場合は、`GMQ_MQ_LIB` 環境変数を設定する必要はありません。この変数を設定しなければ、IBM MQ は、`amqzst.dll` をロードしようとします。クライアントだけのインストール環境のようにこの DLL が存在しなければ、IBM MQ は、`mqic.dll` をロードしようとします。

HOME

この変数には、`mqclient.ini` ファイルの検索時に検索の対象になるディレクトリーの名前を組み込みます。このファイルには、以下のプラットフォームで IBM MQ MQI clients が使用する構成情報が入っています。

-  IBM i
-  UNIX
-  Linux

HOMEDRIVE と HOMEPATH

使用する場合は、両方の変数を設定する必要があります。これらの変数には、`mqclient.ini` ファイルの検索時に検索の対象になるディレクトリーの名前を組み込みます。このファイルには、Windows システムで IBM MQ MQI clients が使用する構成情報が入っています。

LDAP_BASEDN

LDAP サンプル・プログラムを実行するための必須の環境変数。ディレクトリー検索の基本識別名を指定します。

LDAP_HOST

LDAP サンプル・プログラムを実行するための任意指定の変数。LDAP サーバーが稼働しているホストの名前を指定します。指定しなければ、デフォルトでローカル・ホストになります。

LDAP_VERSION

LDAP サンプル・プログラムを実行するための任意指定の変数。使用する LDAP プロトコルのバージョンを指定します。2 または 3 のいずれかを指定できます。ほとんどの LDAP サーバーは、バージョン 3

のプロトコルをサポートするようになりましたが、すべての LDAP サーバーは、旧バージョン 2 もサポートしています。このサンプルは、どちらのバージョンのプロトコルでも同様に機能します。指定しなければ、デフォルトでバージョン 2 になります。

MQAPI_TRACE_LOGFILE

サンプル API 出口プログラムは、MQI トレースをユーザー指定ファイルに生成します。MQAPI_TRACE_LOGFILE 環境変数では、そのファイルの接頭部を定義します。

MQCCSID

使用するコード化文字セット番号を指定します。アプリケーションのネイティブ CCSID をオーバーライドします。

MQCERTLABL

証明書ラベルを定義します。

MQCERTVPOL

使用され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。

MQCHLLIB

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

MQCHLTAB

MQCHLTAB では、クライアント・チャンネル定義テーブル (CCDT) が含まれている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。

MQS_IPC_HOST

IBM MQ のファイルを共有するときに、myHostName の生成値で問題が発生する場合は、環境変数 MQS_IPC_HOST を使用して myHostName を設定します。

MQCLNTCF

この環境変数では、mqclient.ini ファイルのパスを変更します。

MQ_CHANNEL_SUPPRESS_INTERVAL

MQ_CHANNEL_SUPPRESS_MSGS で定義されたメッセージがエラー・ログに書き込まれるのを抑制する時間間隔を秒単位で指定します。この環境変数には、メッセージの発生回数も共に指定します。指定した時間間隔の間に指定した回数を超えて発生したメッセージは抑制されます。デフォルト値は 60,5 です。これは、60 秒の時間間隔で、その間の指定したメッセージの最初の 5 回の発生後はそのメッセージの発生が抑制されることを意味します。詳しくは、[Suppressing channel error messages from error logs on Multiplatforms \(Multiplatforms でエラー・ログからチャンネルのエラー・メッセージを抑制する\)](#) を参照してください。

環境変数の MQ_CHANNEL_SUPPRESS_INTERVAL は、[qm.ini](#) ファイル内の [SuppressInterval](#) と同等です。

MQ_CHANNEL_SUPPRESS_MSGS

エラー・ログに書き込まれる IBM MQ チャンネルのエラー・メッセージを指定します。この書き込みは、MQ_CHANNEL_SUPPRESS_INTERVAL で定義された時間間隔の間、その時間間隔が満了するまでに、指定した回数だけメッセージの発生が許可され、それを超えると書き込みが抑制されます。詳しくは、[Suppressing channel error messages from error logs on Multiplatforms \(Multiplatforms でエラー・ログからチャンネルのエラー・メッセージを抑制する\)](#) を参照してください。

環境変数 MQ_CHANNEL_SUPPRESS_MSGS は、[qm.ini](#) ファイルの [SuppressMessage](#) に相当しますが、指定は異なります。

MQ_CONNECT_TYPE

次のプラットフォームでは、MQCONNX 呼び出しで使用する MQCNO 構造の「Options」フィールドに指定されるバインディングタイプと、この環境変数を組み合わせて使用します。『MQCONNX 環境変数』を参照してください。

-  Windows
-   UNIX and Linux
-  IBM i

 V9.0.0.1

MQ_CROSS_QUEUE_ORDER_ALL

MQ_CROSS_QUEUE_ORDER_ALL 環境変数をゼロ以外の値に設定すると、メッセージの書き込み順序は作業単位で維持されます。つまり、作業単位 (UoW) のメッセージが複数のキュー (例えば Q1 と Q2) に書き込まれる場合、MQCMIT が発行されると、メッセージはキューに書き込まれたときと同じ順序で送信され使用可能になります。

複数のキュー・マネージャーを使用する環境では、各キュー・マネージャーの開始前に、送信側と受信側の両方に MQ_CROSS_QUEUE_ORDER_ALL が存在しており、これに空ではない値が設定されている必要があります。

MQ_FILE_PATH

Windows オペレーティング・システムにランタイム・パッケージをインストールすると、MQ_FILE_PATH という新しい環境変数が構成されます。この環境変数には、Windows レジストリーの以下のキーと同じデータが入ります。

```
HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName\FilePath
```

MQIPADDRV

MQIPADDRV は、チャンネル接続に使用すべき IP プロトコルを指定します。指定できるストリング値は、「MQIPADDR_IPv4」または「MQIPADDR_IPv6」です。これらの値は、ALTER QMGR IPADDRV の IPv4 および IPv6 と同じ意味です。設定されていない場合は、「MQIPADDR_IPv4」が想定されます。

MQ_JAVA_DATA_PATH

ログ出力とトレース出力のディレクトリーを指定します。

MQ_JAVA_INSTALL_PATH

IBM MQ classes for Java のインストール先のディレクトリーを指定します (IBM MQ classes for Java のインストール・ディレクトリーで表示されます)。

MQ_JAVA_LIB_PATH

IBM MQ classes for Java ライブラリーの格納先のディレクトリーを指定します。IBM MQ classes for Java に用意されている一部のスクリプト (IVTRun など) は、この環境変数を使用します。

MQMAXERRORLOGSIZE

MQMAXERRORLOGSIZE は、バックアップにコピーされるキュー・マネージャーのエラー・ログのサイズを指定します。

MQNAME

MQNAME では、IBM MQ プロセスが使用できるローカル NetBIOS 名を指定します。

MQNOREMPOOL

この変数を設定すると、チャンネル・プールがオフに切り替わり、チャンネルがリスナーのスレッドとして実行されるようになります。

MQPSE_TRACE_LOGFILE

出口サンプル・プログラムをパブリッシュするときに使用します。アプリケーション・プロセスのトレースのために、この環境変数では、トレース・ファイルの書き込み先を指定します。『パブリッシュ 出口サンプル・プログラム』を参照してください。

MQSERVER

MQSERVER 環境変数は、最小チャネルを定義するために使用されます。TLS チャネルの定義や、チャネル出口があるチャネルの定義に MQSERVER を使用することはできません。MQSERVER は、IBM MQ サーバーの位置および使用する通信方式を指定します。

MQ_SET_NODELAYACK

この変数を設定すると、TCP 遅延確認応答がオフに切り替わります。

AIX AIX でこの変数を設定すると、オペレーティング・システムの `setsockopt` 呼び出しが `TCP_NODELAYACK` オプションを指定して呼び出されることにより、TCP 遅延確認応答がオフに切り替わります。この関数は AIX でのみサポートされるので、MQ_SET_NODELAYACK 環境変数は AIX でのみ効果があります。

MQSNOAUT

この変数に何らかの値を設定すると、オブジェクト権限マネージャー (OAM) が無効になり、セキュリティ検査が回避されます。MQSNOAUT 変数は、キュー・マネージャーが作成されるときにのみ有効です。

OAM を有効にするには、キュー・マネージャーを削除し、環境変数を削除してから、MQSNOAUT を指定せずにキュー・マネージャーを再作成してください。

MQSPREFIX

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。

MQSSLCRYP

MQSSLCRYP では、システムに存在する暗号ハードウェアを構成するために使用できるパラメーター・ストリングを保持します。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。

MQSSLFIPS

MQSSLFIPS では、IBM MQ で暗号化を実行するときに、FIPS 認定のアルゴリズムだけを使用するかどうかを指定します。この値は、ALTER QMGR コマンドの SSLFIPS パラメーター用と同じです。

MQSSLKEYR

MQSSLKEYR では、ユーザーのデジタル証明書を格納するキー・リポジトリー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。

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。

MQSSLRESET

MQSSLRESET では、TLS 秘密鍵の再ネゴシエーションが行われる前に、TLS チャネルで送受信される非暗号化バイト数を指定します。

MQS_TRACE_OPTIONS

詳細な実行トレース機能およびパラメーター・トレース機能を個々に活動化するには、環境変数 MQS_TRACE_OPTIONS を使用します。

MQTCPTIMEOUT

この変数では、IBM MQ が TCP 接続呼び出しを待つ時間を指定します。

MQSUITEB

この変数は、Suite B 準拠の暗号方式を使用するかどうかを指定します。Suite B 暗号方式を使用する場合、MQSUITEB を以下のいずれかに設定することにより、暗号化の強度を指定できます。

- NONE
- 128_BIT, 192_BIT
- 128_BIT
- 192_BIT

ODQ_MSG

RUNMQDLQ 以外の送達不能キュー処理プログラムを使用する場合は、用意されているサンプルのソースをベースとして使用できます。このサンプルは、製品に用意されている送達不能キュー処理プログラムとよく似ていますが、トレースとエラーの報告の動作が異なります。ODQ_MSG 環境変数では、エラー・メッセージと情報メッセージを格納するファイルの名前を設定します。amqsdlq.msg という名前のファイルが提供されています。

ODQ_TRACE

RUNMQDLQ 以外の送達不能キュー処理プログラムを使用する場合は、用意されているサンプルのソースをベースとして使用できます。このサンプルは、製品に用意されている送達不能キュー処理プログラムとよく似ていますが、トレースとエラーの報告の動作が異なります。ODQ_TRACE 環境変数を YES または yes に設定すると、トレース機能が使用可能になります。

OMQ_PATH

この環境変数では、ActiveX スクリプトに対応した IBM MQ の自動化クラスが失敗した場合に生成される基本障害症状レポートの場所を指定します。

OMQ_TRACE

MQAX は、ユーザー・サイトで問題が発生したときに、保守部門が発生した問題の状況を突き止めるためのトレース機能を備えています。トレース機能は、MQAX スクリプトを実行したときに使用したパスを示します。問題があるとき以外は、システム・リソースの浪費を避けるために、トレースをオフに設定して実行してください。OMQ_TRACE は、トレースを制御するために設定する 3 つの環境変数の 1 つです。OMQ_TRACE にいずれかの値を指定すると、トレース機能がオンに切り替わります。OMQ_TRACE を OFF に設定しても、トレースは引き続きアクティブです。『[トレースの使用](#)』を参照してください。

OMQ_TRACE_PATH

トレースを制御するために設定する 3 つの環境変数の 1 つ。『[トレースの使用](#)』を参照してください。

OMQ_TRACE_LEVEL

トレースを制御するために設定する 3 つの環境変数の 1 つ。『[トレースの使用](#)』を参照してください。

ONCONFIG

Informix® サーバーの構成ファイルの名前。例えば、UNIX and Linux システムでは、以下を使用します。

```
export ONCONFIG=onconfig.hostname_1
```

Windows システムでは、以下を使用します。

```
set ONCONFIG=onconfig.hostname_1
```

WCF_TRACE_ON

WCF カスタム・チャンネルでは、2 つの異なるトレース方式が使用可能です。この 2 つのトレース方式は、単独でアクティブ化することも、一緒にアクティブ化することもできます。それぞれの方式で独自のトレース・ファイルが生成されるので、両方のトレース方式をアクティブ化すると、2 つのトレース出力ファイルが生成されます。この 2 種類のトレース方式を有効/無効にする操作には、4 つの組み合わせがあります。WCF トレースを有効化するためのそれらの組み合わせのほかに、WCF_TRACE_ON 環境変数を使用して XMS .NET トレースを有効にすることもできます。『[WCF トレース構成およびトレース・ファイル名](#)』を参照してください。

WMQSOAP_HOME

IBM MQ で .NET SOAP over JMS サービスのホスティング環境を正しくインストールして構成した後、追加の構成手順を実行する場合に使用します。これは、ローカル・キュー・マネージャーからアクセスできる場所です。『[WCF クライアントから IBM MQ でホストされている .NET サービスへの接続サンプル](#)』と『[WCF クライアントから IBM MQ でホストされている Axis Java サービスへの接続サンプル](#)』を参照してください。

さらに、IBM MQ Web transport for SOAP をインストールするときにも使用します。『[IBM MQ Web transport for SOAP のインストール](#)』を参照してください。

関連情報

IBM MQ 環境変数の使用

IBM i 相互通信ジョブ

以下のジョブは、IBM i 上での相互通信と関連付けられています。これらのジョブの名前を、次の表に示します。

ジョブ名	説明
AMQCLMAA	非スレッド化リスナー
AMQCRSTA	非スレッド化応答側ジョブ
AMQRMPPA	チャンネル・プール・ジョブ
RUNMQCHI	チャンネル・イニシエーター
RUNMQCHL	チャンネル・ジョブ
RUNMQLSR	スレッド化リスナー

IBM i IBM i におけるチャンネルの状態

チャンネルの状態は、「Work with Channels (チャンネルの処理)」パネルに表示されます。

状態名	意味
STARTING	チャンネルは、ターゲット MCA との折衝準備が整っている。
BINDING	セッションと初期データ変更を確立する。
REQUESTING	要求側チャンネルが接続を開始する。
実行中	転送または転送動作可能にする。
PAUSED	メッセージ再試行間隔の間待機する。
STOPPING	再試行または停止を確立する。
RETRYING	次の再試行を試みるまで待機する。
STOPPED	エラーまたはチャンネル終了コマンドが出されたためチャンネルが停止した。
INACTIVE	チャンネルが正常に処理を終了したか、チャンネルが開始しない。
*なし	状態なし (サーバー接続チャンネルのみ)

ULW UNIX, Linux, and Windows におけるメッセージ・チャンネルの計画例

このセクションでは、2つのキュー・マネージャーを接続して、それらの間でメッセージを送信できるようにする方法を示す詳細な例を記載してあります。

この例は、アプリケーションがキュー・マネージャー QM1 を使用してメッセージをキュー・マネージャー QM2 のキューに入れられるようにするために必要な準備を示しています。QM2 で実行されるアプリケーションは、これらのメッセージを検索して、QM1 の応答キューに応答を送信することができます。

この例では、TCP/IP 接続を介しての通信を想定しています。また、各チャンネルは、対応する伝送キューに最初のメッセージが到達したときに、それをトリガーとして始動するものと想定しています。なお、このトリガーがうまく機能するには、チャンネル・イニシエーターを開始しておく必要があります。

この例では、開始キューとして SYSTEM.CHANNEL.INITQ を使用します。このキューは既に IBM MQ で定義されています。異なる開始キューを使用できますが、チャンネル・イニシエーターを開始するときには、開始キューを自分で定義し、キューの名前を指定しなければなりません。

ULW UNIX, Linux, and Windows の例が示す内容

この例は、使用できる IBM MQ コマンド (MQSC) を示しています。

例ではすべて、MQSC コマンドはコマンドのファイル内にあるものとして、またはコマンド行に入力されたものとして示されます。これらの 2 つの方法は同じに見えますが、コマンド行でコマンドを実行するには、まず、runmqsc (デフォルト・キュー・マネージャーの場合) または runmqsc qmname (qmname は必要なキュー・マネージャーの名前) を入力する必要があります。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。

別の方法として、これらのコマンドの入ったファイルを作成することがあります。こうすることで、コマンド内のどのエラーも簡単に訂正することができます。ファイル MQSC.in を呼び出してから、それをキュー・マネージャー QMNAME で実行するためには、次のように入力します。

```
runmqsc QMNAME < mqsc.in > mqsc.out
```

ファイル内のコマンドは、実行する前に、次のコマンドを使用して検査することができます。

```
runmqsc -v QMNAME < mqsc.in > mqsc.out
```

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。1 行を超える場合は、連結文字を使用してください。Windows では、Ctrl-z を使用してコマンド行の入力を終了させます。UNIX and Linux システムでは、Ctrl-d を使用します。あるいは、**end** コマンドを使用します。

178 ページの図 7 はシナリオ例を示しています。

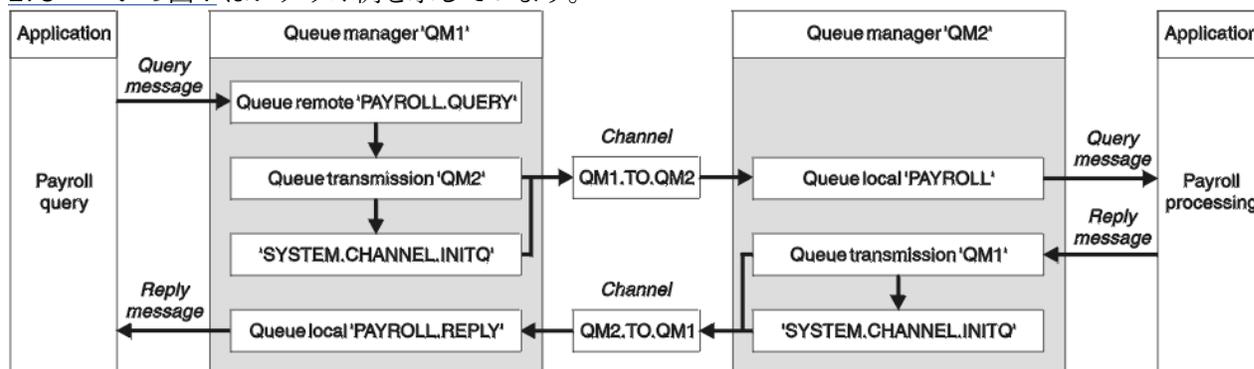


図 7. UNIX, Linux, and Windows システムのメッセージ・チャンネルの例

例には、キュー・マネージャー QM1 に接続された給与計算照会アプリケーションが含まれますが、これは、キュー・マネージャー QM2 で実行している給与計算処理アプリケーションに給与計算照会メッセージを送信します。給与計算照会アプリケーションの照会に対する応答が QM1 に戻されなければなりません。給与計算照会メッセージは QM1.TO.QM2 と呼ばれる送信側 - 受信側チャンネルを介して、QM1 から QM2 へ送信され、その応答メッセージは QM2.TO.QM1 と呼ばれる別の送信側 - 受信側チャンネルを介して、QM2 から QM1 へと戻されます。これらのチャンネルはいずれも、他のキュー・マネージャーに送信するメッセージを得るとただちに、それをトリガーとして始動します。

給与計算照会アプリケーションは、QM1 で定義されているリモート・キュー "PAYROLL.query" に照会メッセージを入れます。このリモート・キュー定義は、QM2 でローカル・キュー "PAYROLL" として解決されます。さらに、給与計算照会アプリケーションでは、照会に対する応答が QM1 のローカル・キュー "PAYROLL.reply" に送信されるように指定されています。給与計算処理アプリケーションは、QM2 のローカル・キュー "PAYROLL" からメッセージを取得して、必要な宛先へ応答を送信します。この例では、QM1 のローカル・キュー "PAYROLL.REPLY" が宛先になります。

TCP/IP の定義の例では、QM1 は、ホスト・アドレスが 192.0.2.0 で、ポート 1411 で listen 中であり、QM2 はホスト・アドレスが 192.0.2.1 で、ポート 1412 で listen 中です。この例では、これらがすでに運用システムで定義済みであり、使用可能であるものと想定しています。

QM1 で作成する必要があるオブジェクト定義は次のとおりです。

- リモート・キュー定義 PAYROLL.QUERY
- 伝送キュー定義 QM2 (デフォルト値 = リモート・キュー・マネージャー名)
- 送信側チャンネル定義 QM1.TO.QM2
- 受信側チャンネル定義 QM2.TO.QM1
- 応答先キュー定義 PAYROLL.REPLY

QM2 で作成する必要があるオブジェクト定義は次のとおりです。

- ローカル・キュー定義 PAYROLL
- 伝送キュー定義 QM1 (デフォルト値 = リモート・キュー・マネージャー名)
- 送信側チャンネル定義 QM2.TO.QM1
- 受信側チャンネル定義 QM1.TO.QM2

接続の詳細は、送信側チャンネル定義の CONNAME 属性内に示されています。

この配置を表すダイアグラムが [178 ページの図 7](#) に示されています。

ULW UNIX, Linux, and Windows 用のキュー・マネージャー QM1 の例

これらのオブジェクト定義によって、キュー・マネージャー QM1 に接続されているアプリケーションは、QM2 の PAYROLL と呼ばれるキューに要求メッセージを送信し、QM1 の PAYROLL.REPLY と呼ばれるキューで応答を受信することができます。

どのオブジェクト定義にも DESCR と REPLACE の属性が指定されています。それ以外には、この例を実施するための最小限の必須属性が与えられています。指定されていない属性には、キュー・マネージャー QM1 に対するデフォルト値が用意されています。

キュー・マネージャー QM1 で以下のコマンドを実行します。

リモート・キュー定義

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QM2') REPLACE +  
PUT(ENABLED) XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

注: リモート・キュー定義は物理キューではありませんが、伝送キュー QM2 にメッセージを送って、それらをキュー・マネージャー QM2 に送信できるようにします。

伝送キュー定義

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') REPLACE +  
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +  
INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(QM1.TO.QM2.PROCESS)
```

最初のメッセージがこの伝送キューに書き込まれると、開始キュー SYSTEM.CHANNEL.INITQ にトリガー・メッセージが送信されます。チャンネル・イニシエーターは、開始キューからメッセージを入手して、指定されたプロセス内に示されているチャンネルを始動させます。

送信側チャンネル定義

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) +  
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +  
CONNAME('192.0.2.1(1412)')
```

受信側チャンネル定義

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QM2')
```

応答先キュー定義

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Reply queue for replies to query messages sent to QM2')
```

応答先キューは PUT(ENABLED) として定義されます。これによって、応答メッセージが確実にキューに書き込まれるようになります。応答先キューに応答を入れることができない場合、応答は、QM1 の送達不能キューに送信されるか、あるいは、このキューが使用できない場合は、キュー・マネージャー QM2 の伝送キュー QM1 に残されます。このキューは GET(ENABLED) として定義され、応答メッセージを検索することができます。

ULW UNIX, Linux, and Windows 用のキュー・マネージャー QM2 の例

以下のオブジェクト定義によって、キュー・マネージャー QM2 に接続されているアプリケーションは、PAYROLL と呼ばれるローカル・キューから要求メッセージを取り出し、これらの要求メッセージに対する応答をキュー・マネージャー QM1 の PAYROLL.REPLY と呼ばれるキューに書き込むことができます。

リモート・キュー定義を用意して、応答を QM1 に戻す必要はありません。ローカル・キュー PAYROLL から取り出されるメッセージのメッセージ記述子には、応答先キューと応答先キュー・マネージャーの両方の名前が入っています。したがって、QM2 が応答先キュー・マネージャー名を、キュー・マネージャー QM2 の伝送キューの応答先キュー・マネージャー名として解決できるのであれば、応答メッセージを送信することができます。この例では、応答先キュー・マネージャー名は QM1 であるため、キュー・マネージャー QM2 には同じ名前の伝送キューが必要です。

どのオブジェクト定義にも DESCR と REPLACE の属性が指定されています。この 2 つの属性は、この例を実施させるための最小限の必須属性です。指定されていない属性には、キュー・マネージャー QM2 に対するデフォルト値が用意されています。

キュー・マネージャー QM2 で以下のコマンドを実行します。

ローカル・キュー定義

```
DEFINE QLOCAL(PAYROLL) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Local queue for QM1 payroll details')
```

このキューは、キュー・マネージャー QM1 における応答先キュー定義の場合と同じ理由で、PUT(ENABLED) および GET(ENABLED) として定義されます。

伝送キュー定義

```
DEFINE QLOCAL(QM1) DESCR('Transmission queue to QM1') REPLACE +  
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +  
INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(QM2.TO.QM1.PROCESS)
```

最初のメッセージがこの伝送キューに書き込まれると、開始キュー SYSTEM.CHANNEL.INITQ にトリガー・メッセージが送信されます。チャンネル・イニシエーターは、開始キューからメッセージを入手して、指定されたプロセス内に示されているチャンネルを始動させます。

送信側チャンネル定義

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(TCP) +
```

```
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +  
CONNNAME('192.0.2.0(1411)')
```

受信側チャンネル定義

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QM1')
```

ULW UNIX, Linux, and Windows 用のサンプルの実行

チャンネル・イニシエーターおよびリスナーの開始に関する情報と、このシナリオの展開についての提案を示します。

これらの定義が作成されたら、以下に示すことを行う必要があります。

- 各キュー・マネージャーに対してチャンネル・イニシエーターを開始させます。
- 各キュー・マネージャーに対してリスナーを開始させます。

チャンネル・イニシエーターおよびリスナーの開始方法については、[Windows における通信のセットアップ](#) および [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。

この例の拡張

この単純な例を、以下の方法で拡張することができます。

- LU 6.2 通信を使用して、CICS システムとの相互接続およびトランザクション処理を行います。
- キュー、プロセス、チャンネル定義をさらに追加して、他のアプリケーションが2つのキュー・マネージャー間でメッセージを送信できるようにします。
- チャンネルにユーザー出口プログラムを追加して、リンク暗号化、セキュリティー検査、または追加のメッセージ処理を行うことができます。
- キュー・マネージャーの別名および応答先キューの別名を使用して、使用されているキュー・マネージャーのネットワーク構成において、それらをどのように使用できるかを理解します。

IBM i IBM i におけるメッセージ・チャンネルの計画例

このセクションでは、2つの IBM i キュー・マネージャーを接続して、それらの間でメッセージを送信できるようにする方法を示す詳細な例を記載してあります。

この例は、アプリケーションがキュー・マネージャー QM1 を使用してメッセージをキュー・マネージャー QM2 のキューに入れるようにするために必要な準備を示しています。QM2 で実行されるアプリケーションは、これらのメッセージを検索して、QM1 の応答キューに応答を送信することができます。

この例では、TCP/IP 接続を介しての通信を想定しています。また、各チャンネルは、対応する伝送キューに最初のメッセージが到達したときに、それをトリガーとして始動するものと想定しています。

この例では、開始キューとして SYSTEM.CHANNEL.INITQ を使用します。このキューは既に IBM MQ で定義されています。異なる開始キューを使用できますが、それを自分で定義し、STRMQMCHLI コマンドを使用してチャンネル・イニシエーターの新規インスタンスを開始し、それに開始キューの名前を付ける必要があります。チャンネルのトリガー操作の詳細については、[チャンネルのトリガー操作](#) を参照してください。

IBM i IBM i の例が示す内容

この例には、キュー・マネージャー QM1 に接続された給与計算照会アプリケーションが含まれますが、これは、キュー・マネージャー QM2 で実行している給与計算処理アプリケーションに給与計算照会メッセージを送信します。給与計算照会アプリケーションの照会に対する応答が QM1 に戻されなければなりません。

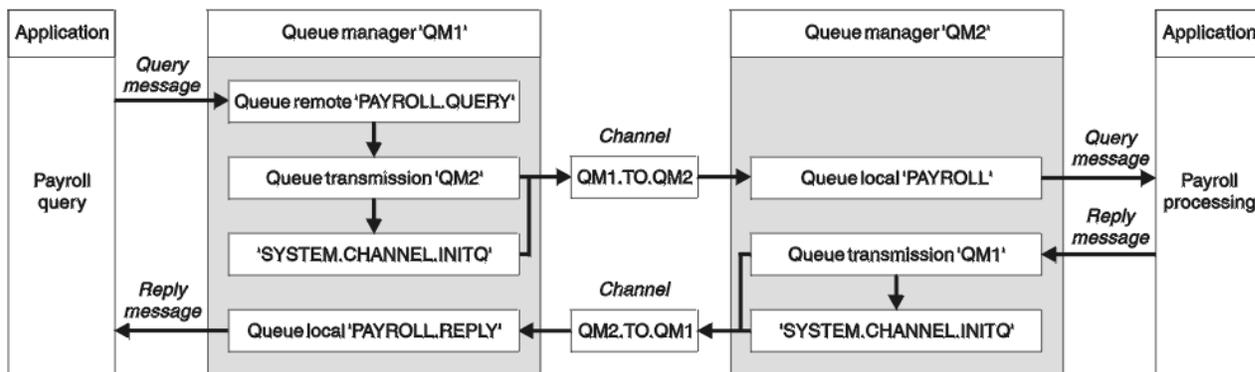


図 8. IBM MQ for IBM i のメッセージ・チャネルの例

給与計算照会メッセージは QM1.TO.QM2 と呼ばれる送信側 - 受信側チャネルを介して、QM1 から QM2 へ送信され、その応答メッセージは QM2.TO.QM1 と呼ばれる別の送信側 - 受信側チャネルを介して、QM2 から QM1 へと戻されます。これらのチャネルはいずれも、他のキュー・マネージャーに送信するメッセージを得るとただちに、それをトリガーとして始動します。

給与計算照会アプリケーションは、QM1 で定義されているリモート・キュー "PAYROLL.query" に照会メッセージを入れます。このリモート・キュー定義は、QM2 でローカル・キュー "PAYROLL" として解決されます。さらに、給与計算照会アプリケーションでは、照会に対する応答が QM1 のローカル・キュー "PAYROLL.reply" に送信されるように指定されています。給与計算処理アプリケーションは、QM2 のローカル・キュー "PAYROLL" からメッセージを取得して、必要な宛先へ応答を送信します。この例では、QM1 のローカル・キュー "PAYROLL.REPLY" が宛先になります。

いずれのキュー・マネージャーも IBM i 上で実行されるものと想定しています。この定義の例では、QM1 は、ホスト・アドレスが 192.0.2.0 で、ポート 1411 で listen 中です。QM2 は、ホスト・アドレスが 192.0.2.1 で、ポート 1412 で listen 中です。この例では、これらのキュー・マネージャーがすでに IBM i システムで定義済みであり、使用可能であるものと想定しています。

QM1 で作成する必要があるオブジェクト定義は次のとおりです。

- リモート・キュー定義 PAYROLL.QUERY
- 伝送キュー定義 QM2 (デフォルト値 = リモート・キュー・マネージャー名)
- 送信側チャネル定義 QM1.TO.QM2
- 受信側チャネル定義 QM2.TO.QM1
- 応答先キュー定義 PAYROLL.REPLY

QM2 で作成する必要があるオブジェクト定義は次のとおりです。

- ローカル・キュー定義 PAYROLL
- 伝送キュー定義 QM1 (デフォルト値 = リモート・キュー・マネージャー名)
- 送信側チャネル定義 QM2.TO.QM1
- 受信側チャネル定義 QM1.TO.QM2

接続の詳細は、送信側チャネル定義の CONNAME 属性内に示されています。

この配置を表すダイアグラムが [182 ページの図 8](#) に示されています。

IBM i IBM i 用のキュー・マネージャー QM1 の例

以下のオブジェクト定義によって、キュー・マネージャー QM1 に接続されているアプリケーションは、QM2 の PAYROLL と呼ばれるキューに要求メッセージを送信し、QM1 の PAYROLL.REPLY と呼ばれるキューで応答を受信することができます。

どのオブジェクト定義にも TEXT 属性が与えられています。それ以外には、この例を実施するための最小限の必須属性が与えられています。指定されていない属性には、キュー・マネージャー QM1 に対するデフォルト値が用意されています。

キュー・マネージャー QM1 で以下のコマンドを実行します。

リモート・キュー定義

次の属性が指定された CRTMQMQ コマンド。

QNAME	'PAYROLL.QUERY'
QTYPE	*RMT
TEXT	'Remote queue for QM2'
PUTENBL	*YES
TMQNAME	'QM2' (デフォルト = リモート・キュー・マネージャー名)
RMTQNAME	'PAYROLL'
RMTMQMNAME	'QM2'

注：リモート・キュー定義は物理キューではありませんが、伝送キュー QM2 にメッセージを送って、それらをキュー・マネージャー QM2 に送信できるようにします。

伝送キュー定義

次の属性が指定された CRTMQMQ コマンド。

QNAME	QM2
QTYPE	*LCL
TEXT	'Transmission queue to QM2'
USAGE	*TMQ
PUTENBL	*YES
GETENBL	*YES
TRGENBL	*YES
TRGTYPE	*FIRST
INITQNAME	SYSTEM.CHANNEL.INITQ
TRIGDATA	QM1.TO.QM2

最初のメッセージがこの伝送キューに書き込まれると、開始キュー SYSTEM.CHANNEL.INITQ にトリガー・メッセージが送信されます。チャンネル・イニシエーターは、開始キューからメッセージを入手して、指定されたプロセス内に示されているチャンネルを始動させます。

送信側チャンネル定義

次の属性が指定された CRTMQMCHL コマンド。

CHLNAME	QM1.TO.QM2
CHLTYPE	*SDR
TRPTYPE	*TCP
TEXT	'Sender channel to QM2'
TMQNAME	QM2
CONNNAME	'192.0.2.1(1412)'

受信側チャンネル定義

次の属性が指定された CRTMQMCHL コマンド。

CHLNAME	QM2.TO.QM1
CHLTYPE	*RCVR

TRPTYPE	*TCP
TEXT	'Receiver channel from QM2'

応答先キュー定義

次の属性が指定された CRTMQMQ コマンド。

QNAME	PAYROLL.REPLY
QTYPE	*LCL
TEXT	'Reply queue for replies to query messages sent to QM2'
PUTENBL	*YES
GETENBL	*YES

応答先キューは PUT(ENABLED) として定義されます。この定義によって、応答メッセージが確実にキューに書き込まれるようになります。応答先キューに応答を入れることができない場合、応答は、QM1 の送達不能キューに送信されるか、あるいは、このキューが使用できない場合は、キュー・マネージャー QM2 の伝送キュー QM1 に残されます。このキューは GET(ENABLED) として定義され、応答メッセージを検索することができます。

IBM i IBM i 用のキュー・マネージャー QM2 の例

以下のオブジェクト定義によって、キュー・マネージャー QM2 に接続されているアプリケーションは、PAYROLL と呼ばれるローカル・キューから要求メッセージを取り出し、これらの要求メッセージに対する応答をキュー・マネージャー QM1 の PAYROLL.REPLY と呼ばれるキューに書き込むことができます。

リモート・キュー定義を用意して、応答を QM1 に戻す必要はありません。ローカル・キュー PAYROLL から取り出されるメッセージのメッセージ記述子には、応答先キューと応答先キュー・マネージャーの両方の名前が入っています。したがって、QM2 が応答先キュー・マネージャー名を、キュー・マネージャー QM2 の伝送キューの応答先キュー・マネージャー名として解決できるのであれば、応答メッセージを送信することができます。この例では、応答先キュー・マネージャー名は QM1 であるため、キュー・マネージャー QM2 には同じ名前の伝送キューが必要です。

どのオブジェクト定義にも、TEXT 属性が与えられています。これらは、この例を機能させるための最小限の必須属性です。指定されていない属性には、キュー・マネージャー QM2 に対するデフォルト値が用意されています。

キュー・マネージャー QM2 で以下のコマンドを実行します。

ローカル・キュー定義

次の属性が指定された CRTMQMQ コマンド。

QNAME	PAYROLL
QTYPE	*LCL
TEXT	'Local queue for QM1 payroll details'
PUTENBL	*YES
GETENBL	*YES

このキューは、キュー・マネージャー QM1 における応答先キュー定義の場合と同じ理由で、PUT(ENABLED) および GET(ENABLED) として定義されます。

伝送キュー定義

次の属性が指定された CRTMQMQ コマンド。

QNAME	QM1
QTYPE	*LCL
TEXT	'Transmission queue to QM1'

USAGE	*TMQ
PUTENBL	*YES
GETENBL	*YES
TRGENBL	*YES
TRGTYPE	*FIRST
INITQNAME	SYSTEM.CHANNEL.INITQ
TRIGDATA	QM2.TO.QM1

最初のメッセージがこの伝送キューに書き込まれると、開始キュー SYSTEM.CHANNEL.INITQ にトリガー・メッセージが送信されます。チャンネル・イニシエーターは、開始キューからメッセージを入手して、トリガー・データで示されているチャンネルを始動させます。

送信側チャンネル定義

次の属性が指定された CRTMQMCHL コマンド。

CHLNAME	QM2.TO.QM1
CHLTYPE	*SDR
TRPTYPE	*TCP
TEXT	'Sender channel to QM1'
TMQNAME	QM1
CONNNAME	'192.0.2.0(1411)'

受信側チャンネル定義

次の属性が指定された CRTMQMCHL コマンド。

CHLNAME	QM1.TO.QM2
CHLTYPE	*RCVR
TRPTYPE	*TCP
TEXT	'Receiver channel from QM1'

IBM i IBM i 用のサンプルの実行

必要なオブジェクトを作成した後は、両方のキュー・マネージャーのチャンネル・イニシエーターおよびチャンネル・リスナーを開始する必要があります。

これで、アプリケーションは相互にメッセージを送信できるようになります。いずれかの伝送キューに最初のメッセージが到達すると、チャンネルがトリガーされて始動するので、STRMQMCHL コマンドを出す必要はありません。

チャンネル・イニシエーターおよびチャンネル・リスナーの開始について詳しくは、[IBM i のチャンネルのモニターと制御](#)を参照してください。

IBM i IBM i 用のサンプルの拡張

例はさまざまな方法で拡張できます。

この例は次のように拡張することができます。

- キュー定義およびチャンネル定義をさらに追加して、他のアプリケーションが2つのキュー・マネージャー間でメッセージを送信できるようにします。
- チャンネルにユーザー出口プログラムを追加して、リンク暗号化、セキュリティー検査、または追加のメッセージ処理を行うことができますようにします。

- キュー・マネージャーの別名および応答先キューの別名を使用して、使用されているキュー・マネージャーのネットワーク構成において、これらのオブジェクトをどのように使用できるかを理解します。

MQSC コマンドを使用するこの例の詳細については、186 ページの『z/OS におけるメッセージ・チャネルの計画例』を参照してください。

z/OS z/OS におけるメッセージ・チャネルの計画例

このセクションでは、z/OS または MVS の各キュー・マネージャーを接続して、互いにメッセージをやりとりできるように、詳細な例を示して説明しています。

この例は、アプリケーションがキュー・マネージャー QM1 を使用してメッセージをキュー・マネージャー QM2 のキューに入れるようにするために必要な準備を示しています。QM2 で実行されるアプリケーションは、これらのメッセージを検索して、QM1 の応答キューに応答を送信することができます。

例では、TCP/IP および LU 6.2 両方の接続の使用方法を図示しています。また、各チャネルは、対応する伝送キューに最初のメッセージが到達したときに、それをトリガーとして始動するものと想定しています。

z/OS z/OS の例が示す内容

この例には、キュー・マネージャー QM1 に接続された給与計算照会アプリケーションが含まれますが、これは、キュー・マネージャー QM2 で実行している給与計算処理アプリケーションに給与計算照会メッセージを送信します。給与計算照会アプリケーションの照会に対する応答が QM1 に戻されなければなりません。

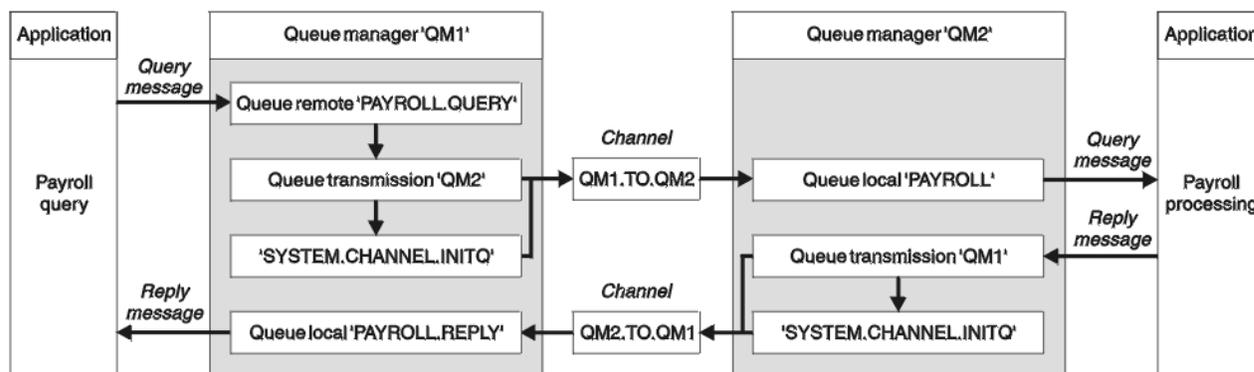


図 9. IBM MQ for z/OS の最初の例

給与計算照会メッセージは QM1.TO.QM2 と呼ばれる送信側 - 受信側チャネルを介して、QM1 から QM2 へ送信され、その応答メッセージは QM2.TO.QM1 と呼ばれる別の送信側 - 受信側チャネルを介して、QM2 から QM1 へと戻されます。これらのチャネルはいずれも、他のキュー・マネージャーに送信するメッセージを得るとただちに、それをトリガーとして始動します。

給与計算照会アプリケーションは、QM1 で定義されているリモート・キュー "PAYROLL.query" に照会メッセージを入れます。このリモート・キュー定義は、QM2 でローカル・キュー "PAYROLL" として解決されます。さらに、給与計算照会アプリケーションでは、照会に対する応答が QM1 のローカル・キュー "PAYROLL.reply" に送信されるように指定されています。給与計算処理アプリケーションは、QM2 のローカル・キュー "PAYROLL" からメッセージを取得して、必要な宛先へ応答を送信します。この例では、QM1 のローカル・キュー "PAYROLL.REPLY" が宛先になります。

いずれのキュー・マネージャーも z/OS 上で実行されるものと想定しています。TCP/IP の定義の例では、QM1 は、ホスト・アドレスが 192.0.2.0 で、ポート 1411 で listen 中であり、QM2 はホスト・アドレスが 192.0.2.1 で、ポート 1412 で listen 中です。LU 6.2 の定義では、QM1 が LUNAME1 と呼ばれるシンボリック論理装置名で listen 中で、QM2 が LUNAME2 と呼ばれるシンボリック論理装置名で listen 中です。この例では、これらの luname が既に z/OS システムで定義済みであり、使用可能になっていることを想定しています。定義する場合の詳細は、58 ページの『z/OS での MQ の構成例』を参照してください。

QM1 で作成する必要があるオブジェクト定義は次のとおりです。

- リモート・キュー定義 PAYROLL.QUERY

- 伝送キュー定義 QM2 (デフォルト値 = リモート・キュー・マネージャー名)
- 送信側チャンネル定義 QM1.TO.QM2
- 受信側チャンネル定義 QM2.TO.QM1
- 応答先キュー定義 PAYROLL.REPLY

QM2 で作成する必要があるオブジェクト定義は次のとおりです。

- ローカル・キュー定義 PAYROLL
- 伝送キュー定義 QM1 (デフォルト値 = リモート・キュー・マネージャー名)
- 送信側チャンネル定義 QM2.TO.QM1
- 受信側チャンネル定義 QM1.TO.QM2

この例では、すべての SYSTEM.COMMAND.* および SYSTEM.CHANNEL.* DQM を実行するために必要なキューは、提供されているサンプル定義 **CSQ4INSG** および **CSQ4INSX** に示すように定義されています。

接続の詳細は、送信側チャンネル定義の CONNAME 属性内に示されています。

この配置を表すダイアグラムが [186 ページの図 9](#) に示されています。

z/OS 用のキュー・マネージャー QM1 の例

以下のオブジェクト定義によって、キュー・マネージャー QM1 に接続されているアプリケーションは、QM2 の PAYROLL という名前のキューに要求メッセージを送信できます。また、アプリケーションは QM1 の PAYROLL.REPLY という名前のキューで応答を受信することもできます。

どのオブジェクト定義にも DESCR と REPLACE の属性が指定されています。それ以外には、この例を実施するための最小限の必須属性が与えられています。指定されていない属性には、キュー・マネージャー QM1 に対するデフォルト値が用意されています。

キュー・マネージャー QM1 で以下のコマンドを実行します。

リモート・キュー定義

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QM2') REPLACE +
PUT(ENABLED) XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

注：リモート・キュー定義は物理キューではありませんが、伝送キュー QM2 にメッセージを送って、それらをキュー・マネージャー QM2 に送信できるようにします。

伝送キュー定義

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QM1.TO.QM2) INITQ(SYSTEM.CHANNEL.INITQ)
```

最初のメッセージがこの伝送キューに書き込まれると、開始キュー SYSTEM.CHANNEL.INITQ にトリガー・メッセージが送信されます。チャンネル・イニシエーターは、開始キューからメッセージを入手して、トリガー・データで示されているチャンネルを始動させます。チャンネル・イニシエーターは、SYSTEM.CHANNEL.INITQ キューからのみトリガー・メッセージを取得できるため、開始キューとして他のキューを使用することはありません。

送信側チャンネル定義

TCP/IP 接続の場合:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +
CONNAME('192.0.2.1(1412)')
```

LU 6.2 接続の場合:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +
CONNAME('LUNAME2')
```

受信側チャネル定義

TCP/IP 接続の場合:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM2')
```

LU 6.2 接続の場合:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QM2')
```

応答先キュー定義

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to query messages sent to QM2')
```

応答先キューは PUT(ENABLED) として定義され、これにより応答メッセージが確実にキューに書き込まれるようになります。応答先キューに応答を入れることができない場合、応答は、QM1 の送達不能キューに送信されるか、あるいは、このキューが使用できない場合は、キュー・マネージャー QM2 の伝送キュー QM1 に残されます。このキューは GET(ENABLED) として定義され、応答メッセージを検索することができます。

z/OS

z/OS 用のキュー・マネージャー QM2 の例

以下のオブジェクト定義によって、キュー・マネージャー QM2 に接続されているアプリケーションは、PAYROLL と呼ばれるローカル・キューから要求メッセージを取り出し、これらの要求メッセージに対する応答をキュー・マネージャー QM1 の PAYROLL.REPLY と呼ばれるキューに書き込むことができます。

リモート・キュー定義を用意して、応答を QM1 に戻す必要はありません。ローカル・キュー PAYROLL から取り出されるメッセージのメッセージ記述子には、応答先キューと応答先キュー・マネージャーの両方の名前が入っています。したがって、QM2 が応答先キュー・マネージャー名を、キュー・マネージャー QM2 の伝送キューの応答先キュー・マネージャー名として解決できるのであれば、応答メッセージを送信することができます。この例では、応答先キュー・マネージャー名は QM1 であるため、キュー・マネージャー QM2 には同じ名前の伝送キューが必要です。

どのオブジェクト定義にも DESCR と REPLACE の属性が指定されています。この 2 つの属性は、この例を実施させるための最小限の必須属性です。指定されていない属性には、キュー・マネージャー QM2 に対するデフォルト値が用意されています。

キュー・マネージャー QM2 で以下のコマンドを実行します。

ローカル・キュー定義

```
DEFINE QLOCAL(PAYROLL) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Local queue for QM1 payroll details')
```

このキューは、キュー・マネージャー QM1 における応答先キュー定義の場合と同じ理由で、PUT(ENABLED) および GET(ENABLED) として定義されます。

伝送キュー定義

```
DEFINE QLOCAL(QM1) DESCR('Transmission queue to QM1') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QM2.TO.QM1) INITQ(SYSTEM.CHANNEL.INITQ)
```

最初のメッセージがこの伝送キューに書き込まれると、開始キュー SYSTEM.CHANNEL.INITQ にトリガー・メッセージが送信されます。チャンネル・イニシエーターは、開始キューからメッセージを入手して、トリガー・データで示されているチャンネルを始動させます。チャンネル・イニシエーターは、SYSTEM.CHANNEL.INITQ からのみトリガー・メッセージを取得できるため、開始キューとして他のキューを使用することはありません。

送信側チャンネル定義

TCP/IP 接続の場合:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +
CONNAME('192.0.2.0(1411)')
```

LU 6.2 接続の場合:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +
CONNAME('LUNAME1')
```

受信側チャンネル定義

TCP/IP 接続の場合:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM1')
```

LU 6.2 接続の場合:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QM1')
```

z/OS 用のサンプルの実行

必要なオブジェクトの作成が完了したら、両方のキュー・マネージャーのチャンネル・イニシエーターとチャンネル・リスナーを起動する必要があります。

これで、アプリケーションは相互にメッセージを送信できるようになります。チャンネルは、各伝送キューに最初のメッセージが届くと開始するようにトリガーが設定されているので、START CHANNEL MQSC コマンドを実行する必要はありません。

チャンネル・イニシエーターの開始についての詳細は [チャンネル・イニシエーターの開始](#) を、リスナーの開始についての詳細は [チャンネル・リスナーの開始](#) を参照してください。

z/OS 用のサンプルの拡張

例はさまざまな方法で拡張できます。

例は、次の方法で拡張できます。

- キューとチャンネル定義をさらに追加して、他のアプリケーションが2つのキュー・マネージャー間でメッセージを送信できるようにします。

- チャンネルにユーザー出口プログラムを追加して、リンク暗号化、セキュリティー検査、または追加のメッセージ処理を行うことができるようにします。
- キュー・マネージャーの別名や応答先キューの別名を使用して、ご使用のキュー・マネージャーのネットワーク編成において、それらの別名を使用する方法についてさらに詳しく理解します。

z/OS キュー共有グループを使用する z/OS におけるメッセージ・チャンネルの計画例

この例は、アプリケーションがキュー・マネージャー QM3 を使用して、キュー・メンバー QM4 および QM5 をもつキュー共有グループのキューにメッセージを書き込めるようにするために必要な準備を示しています。

この例を試してみる前に、186 ページの『[z/OS におけるメッセージ・チャンネルの計画例](#)』に記載している例を理解するようにしてください。

z/OS z/OS のキュー共有グループ例が示す内容

この例は、キュー共有グループを使用する分散キューイング用の IBM MQ for z/OS で使用できる IBM MQ コマンド (MQSC) を示しています。

これは、186 ページの『[z/OS におけるメッセージ・チャンネルの計画例](#)』の例の給与計算照会のシナリオを拡張した例であり、共有キューが利用できるサービスを提供するアプリケーションをさらに追加して照会処理の有用性を高める方法を示しています。

この例では、給与計算照会アプリケーションはキュー・マネージャー QM3 に接続され、QM3 で定義されているリモート・キュー PAYROLL QUERY に照会を書き込みます。このリモート・キュー定義は、キュー共有グループ QSG1 内のキュー・マネージャーをホストとして利用する共有キュー PAYROLL として解決されます。このようにして、給与計算処理アプリケーションは、1 つが QM4 に接続され、もう 1 つが QM5 に接続される 2 つのインスタンスをもつこととなります。

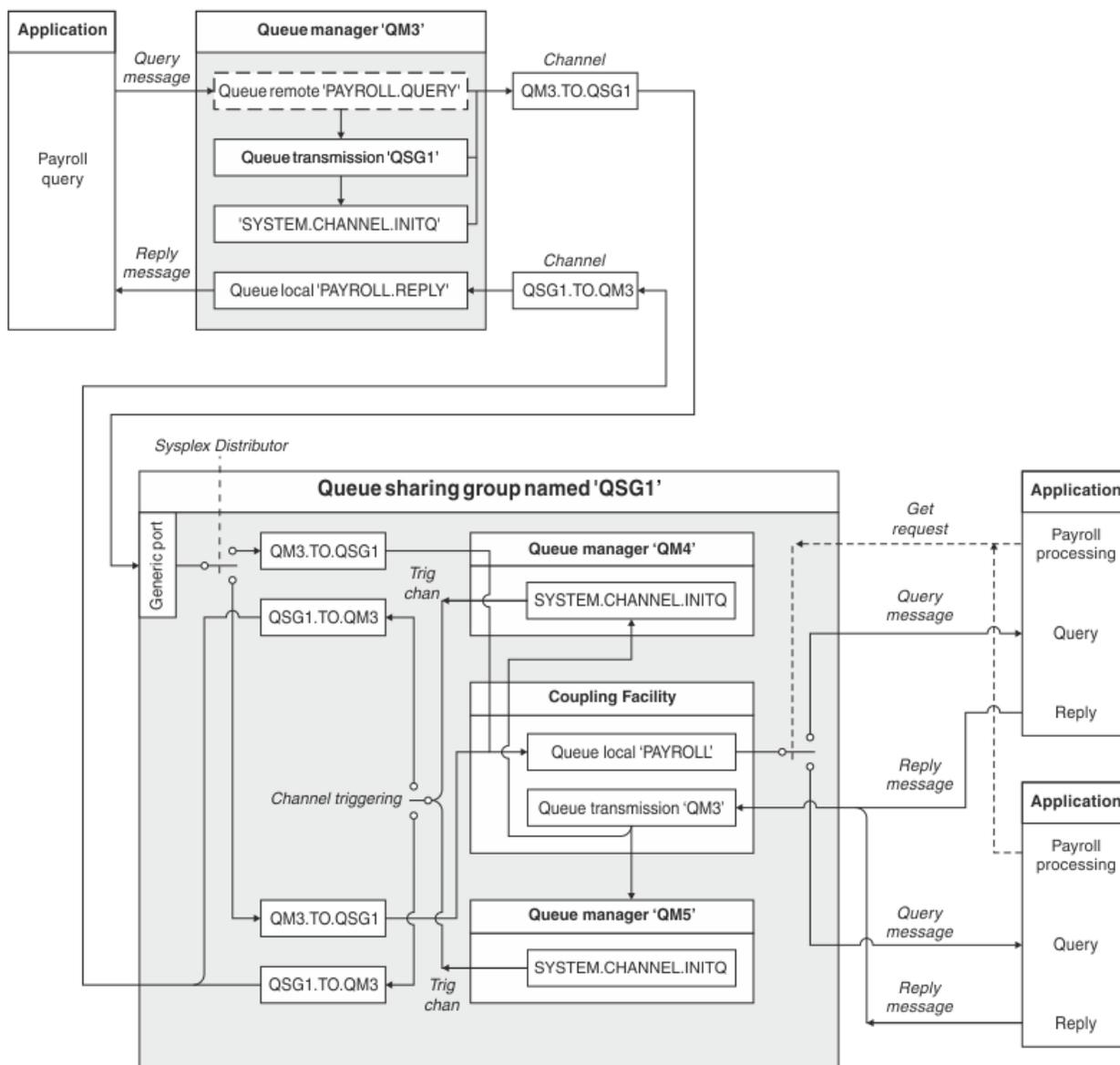


図 10. キュー共有グループを使用する IBM MQ for z/OS におけるメッセージ・チャネルの計画例

3つのキュー・マネージャーは、いずれも z/OS 上で実行されるものと想定しています。TCP/IP 用の定義例では QM4 には MVSIP01 という VIPA アドレスが付き、QM5 には MVSIP02 という VIPA アドレスが付き、どちらのキュー・マネージャーも、ポート 1414 で listen しています。このグループ用に Sysplex Distributor に用意されている汎用アドレスは QSG1.MVSIP です。QM3 は、192.0.2.0 というホスト・アドレスをもち、ポート 1411 で listen します。

LU 6.2 の定義例では、QM3 が LUNAME1 という名前の記号 LU 名で listen します。QM4 および QM5 で listen される LU 名として VTAM 用に定義されている汎用リソースの名前は LUQSG1 です。この例では、これらが既に z/OS システムで定義済みであり、使用可能であるものと想定しています。定義する場合の詳細は、66 ページの『汎用リソースの使用によるネットワークへの自身の登録』を参照してください。

この例では QSG1 はキュー共有グループの名前であり、キュー・マネージャー QM4 と QM5 は、このグループのメンバーの名前です。

z/OS キュー共有グループ定義

キュー共有グループのいずれかのメンバーに対して以下のようなオブジェクト定義を作成すると、他のすべてのメンバーもその定義が使用可能になります。

キュー・マネージャー QM4 および QM5 は、キュー共有グループのメンバーなので、QM4 用に作成された定義は、QM5 でも使用可能です。

カップリング・ファシリティ・リスト構造は APPLICATION1 という名前であることが前提となっています。もしその名前が APPLICATION1 でなければ、この例で自分独自のカップリング・ファシリティ・リスト構造名を使用する必要があります。

共有オブジェクト

共有オブジェクト定義は Db2[®] に保管され、それに関連したメッセージはカップリング・ファシリティ内に保管されます。

```
DEFINE QLOCAL(PAYROLL) QSGDISP(SHARED) REPLACE PUT(ENABLED) GET(ENABLED) +
CFSTRUCT(APPLICATION1) +
DESCR('Shared queue for payroll details')

DEFINE QLOCAL(QM3) QSGDISP(SHARED) REPLACE USAGE(XMITQ) PUT(ENABLED) +
CFSTRUCT(APPLICATION1) +
DESCR('Transmission queue to QM3') TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QSG1.TO.QM3) GET(ENABLED) INITQ(SYSTEM.CHANNEL.INITQ)
```

グループ・オブジェクト

グループ・オブジェクト定義は Db2 に保管され、キュー共有グループ内の各キュー・マネージャーは、定義オブジェクトのローカル・コピーを作成します。

送信側チャンネル定義 (TCP/IP 接続の場合):

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(SDR) QSGDISP(GROUP) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM3') XMITQ(QM3) +
CONNAME('192.0.2.0(1411)')
```

送信側チャンネル定義 (LU 6.2 接続の場合):

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(SDR) QSGDISP(GROUP) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QM3') XMITQ(QM3) +
CONNAME('LUNAME1')
```

受信側チャンネル定義 (TCP/IP 接続の場合):

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM3') QSGDISP(GROUP)
```

受信側チャンネル定義 (LU 6.2 接続の場合):

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QM3') QSGDISP(GROUP)
```

関連資料

[121 ページの『処理 \(QSGDISP\)』](#)

この属性は、キュー共有グループでのチャンネルの処理を指定します。これは、z/OS でのみ有効です。

z/OS

z/OS 用のキュー・マネージャー QM3 の例

QM3 はキュー共有グループのメンバーではありません。QM3 は以下のオブジェクト定義を使用して、キュー共有グループ内のキューにメッセージを書き込むことができます。

このチャンネルの CONNAME は、キュー共有グループの汎用アドレスです。これは、トランスポート・タイプによって異なります。

TCP/IP 接続の場合:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QSG1') XMITQ(QSG1) +
CONNNAME('QSG1.MVSIP(1414)')
```

LU 6.2 接続の場合:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(SDR) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QSG1') XMITQ(QSG1) +
CONNNAME('LUQSG1') TPNAME('MQSERIES') MODENAME('#INTER')
```

その他の定義

最初の例での定義と同じ目的で、以下の定義が必要です。

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QSG1') REPLACE +
PUT(ENABLED) XMITQ(QSG1) RNAME(APPL) RQMNAME(QSG1)

DEFINE QLOCAL(QSG1) DESCR('Transmission queue to QSG1') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QM3.TO.QSG1) INITQ(SYSTEM.CHANNEL.INITQ)

DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QSG1')

DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QSG1')

DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to query messages sent to QSG1')
```

z/OS 用のキュー共有グループ・サンプルの実行

必要なオブジェクトを作成した後は、3つすべてのキュー・マネージャーのチャンネル・イニシエーターを開始する必要があります。また、キュー共有グループ内の両方のキュー・マネージャーに対して、リスナーを開始する必要があります。

TCP/IP 接続の場合、グループのすべてのメンバーが、ポート 1414 で listen するグループ・リスナーを開始しなければなりません。

```
STA LSTR PORT(1414) IPADDR(MVSIP01) INDISP(GROUP)
```

上記のエントリーは、例えば QM4 でリスナーを開始します。

LU6.2 接続の場合、グループのすべてのメンバーが、シンボリック LU 名で listen するグループ・リスナーを開始しなければなりません。この LU 名は、汎用リソース LUQSG1 と対応している必要があります。

- QM3 でのリスナーの開始

```
STA LSTR PORT(1411)
```

別名を使用して MQ ライブラリーを参照する

MQ ライブラリーの名前を直接使用する代わりに、別名を定義して JCL で MQ ライブラリーを参照することができます。その後、MQ ライブラリーの名前が変更された場合には、単に別名を削除して再定義することができます。

例

以下の例では、MQ ライブラリー MQM.V600.SCSQANLE を参照するための別名 MQM.SCSQANLE を定義します。

```
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE (MQM.SCSQANLE)
DEFINE ALIAS (NAME(MQM.SCSQANLE) RELATE(MQM.V600.SCSQANLE))
/*
```

その後、JCLでMQM.V600.SCSQANLE ライブラリーを参照するには、別名MQM.SCSQANLEを使用します。

注: ライブラリー名と別名は同じカタログに含まれる必要があるため、両方に同じ高位修飾子を使用してください(この例で高位修飾子はMQMです)。

V 9.0.1 z/OS mqzOSConnectService エlement

MQ Service Provider は、標準の Liberty 機能として提供されているので、server.xml を使用して構成します。単方向サービスも両方向サービスも、1つのmqzOSConnectService エlementで定義されます。このelementとそのすべての属性は、z/OS Connect V1 および z/OS Connect EE の両方に適用されます。

重要: mqzOSConnectService エlementを使用するには、そのelementをzOSConnectService エlementで参照する必要があります。

いくつかの属性を指定したmqzOSConnectService エlementの例を以下に示します。

```
<mqzOSConnectService id="twoWay "
  connectionFactory="jms/cf1"
  destination="jms/requestQueue"
  replyDestination="jms/replyQueue"
  expiry="-1"
  waitInterval="10000"
  replySelection="msgIDToCorrelID"
  selector=""
  persistence="false"/>
```



重要: MQ Service Provider のインストール方法に応じて、mqzOSConnectService エlementには、ストリングの後に下線が付いた接頭部が付くことがあります(例: `usr_mqzOSConnectService`)。

これについては、[WLP への MQ Service Provider のインストール for z/OS Connect V1 および IBM z/OS Connect EE への MQ Service Provider のインストール \(z/OS Connect EE の場合\)](#) で説明されています。

以下の例に示すフォーマットは、MQ Service Provider が WLP カーネルにインストールされている場所です ([WLP への MQ Service Provider のインストールのオプション 1](#) を参照してください)。

属性名	タイプ	デフォルト値	説明
ID	ストリング		195 ページの『ID』
connectionFactory	JNDI 名(ストリング)。		195 ページの『connectionFactory』
宛先 (destination)	JNDI 名(ストリング)。		195 ページの『宛先 (destination)』
replyDestination	JNDI 名(ストリング)。		196 ページの『replyDestination』
expiry	integer	-1	196 ページの『expiry』
waitInterval	integer		196 ページの『waitInterval』

表 36. *mqzOSConnectService* エレメントの属性 (続き)

属性名	タイプ	デフォルト値	説明
replySelection	ストリング	msgIDToCorrelID	196 ページの『replySelection』
セレクター	ストリング		197 ページの『セレクター』
persistence	boolean	false	197 ページの『persistence』
mqmdFormat	ストリング		197 ページの『mqmdFormat』
userName	ストリング		198 ページの『userName』
パスワード	ストリング		198 ページの『パスワード』
useCallerPrincipal	boolean	false	198 ページの『useCallerPrincipal』
receiveTextCCSID	integer	37	198 ページの『receiveTextCCSID』

ID

id は必須属性であり、*server.xml* 内のすべてのエレメントの中で一意でなければなりません。 **id** は、ターゲット・サービス・プロバイダー・インスタンスを参照するために *zosConnectService* エレメントで使用します。

connectionFactory

connectionFactory は、IBM MQ メッセージング・プロバイダーの接続ファクトリーの JNDI 名を指定します。MQ Service Provider は、接続ファクトリーを使用して IBM MQ に接続します。

connectionFactory は必須属性です。接続ファクトリーについては、[JMS 接続ファクトリー](#)を参照してください。

接続ファクトリーに対して **transportType="BINDINGS"** を指定する必要があります。

宛先 (destination)

destination は、IBM MQ メッセージング・プロバイダーの宛先の JNDI 名を指定します。

destination は必須属性です。

以下を構成する方法については詳しくは、それぞれの資料を参照してください。

- WLP 内のキューについては、[JMS キュー](#)を参照してください。
- WLP 内のトピックについては、[JMS トピック](#)を参照してください。

単方向サービスでは、**destination** が HTTP POST 要求、HTTP GET 要求、および HTTP DELETE 要求の宛先として使用されます。

キュー宛先はこれらの 3 つの要求タイプのすべてでサポートされますが、トピック宛先は HTTP POST 要求でしかサポートされないことに注意してください。

両方向サービスでは、**destination** は、バックエンド・サービスで使用される要求キューを表すキュー宛先でなければなりません。

両方向サービスは、HTTP POST 要求のみをサポートします。

replyDestination

replyDestination は、IBM MQ メッセージング・プロバイダーのキューの JNDI 名を指定します。

replyDestination はオプションの属性です。

WLP 内のキューを構成する方法については、[JMS キュー](#)を参照してください。

replyDestination を指定しなかったサービスは、単方向サービスになります。**replyDestination** を指定したサービスは、両方向サービスになります。

このキューは、バックエンド・サービスが応答メッセージを送信する応答宛先です。

expiry

expiry は、MQ Service Provider が送信するメッセージの有効期間を、送信時からの 1/1000 秒単位で指定します。この時間が経過するまでに宛先キューからメッセージが除去されなかった場合、そのメッセージは廃棄の対象となります。

expiry はオプションの属性であり、MQMD [Expiry](#) フィールドの設定と同等です。

負の値を指定した場合、メッセージは無期限に有効になります。**expiry** のデフォルト値は -1 です。

REST クライアントは、`ibm-mq-md-expiry` HTTP ヘッダーに有効な 16 ビット整数を指定して、**expiry** をオーバーライドできます。

waitInterval

単方向サービスに対する HTTP DELETE 要求の場合、**waitInterval** は、[destination](#) 属性で指定したキュー上で一致するメッセージを待機する時間をミリ秒単位で指定します。

両方向サービスに対する HTTP POST 要求の場合、**waitInterval** は、[replydestination](#) 属性で指定したキュー上で一致するメッセージを待機する時間をミリ秒単位で指定します。

waitInterval は、片方向サービスの場合はオプション属性、両方向サービスの場合は必須属性であり、MQMD [WaitInterval](#) フィールドの設定と同等です。

waitInterval は、HTTP GET 要求ではサポートされません。

waitInterval に応じて次のようになります。

- ゼロの場合、サービスは待機しません。
waitInterval をゼロにすることは、両方向サービスではサポートされません。
- 負の値の場合、サービスはメッセージが使用可能になるまで無期限に待機します。

REST クライアントは、`ibm-mq-gmo-waitInterval` HTTP ヘッダーに有効な 64 ビット整数を指定して、この値をオーバーライドできます。

注: **waitInterval** に大きな値や負の値を指定すると、トランザクション・タイムアウトおよび非同期サービス要求タイムアウトが発生する可能性があります。これらのイベントのいずれかまたは両方が発生した場合は、タイムアウトを延長するか、待機間隔を短くするか、またはその両方を行ってください。

replySelection

replySelection は、応答メッセージと要求メッセージの突き合わせに使用するメカニズムを指定します。

replySelection はオプションであり、両方向サービスでのみ使用されます。**replySelection** を単方向サービスで使用した場合は無視されます。

値は、次のいずれか 1 つです。

msgIDToCorrelID

応答メッセージは、要求メッセージのメッセージ ID の値が相関 ID として設定された状態で生成されると想定されます。サービスは、この情報に基づいて適切なメッセージ・セレクターを生成します。これはデフォルト値です。

なし

応答メッセージを要求メッセージに相関させるメカニズムは使用されません。サービスは、応答キューにある最初のメッセージを取得します。

correlIDToCorrelID

応答メッセージは、要求メッセージの相関 ID の値が相関 ID として設定された状態で生成されると想定されます。サービスは、この情報に基づいて適切なメッセージ・セレクターを生成します。要求メッセージに相関 ID が指定されていない場合は (199 ページの『[ibm-mq-md-correlID](#)』を参照)、サービスがその要求メッセージ用にランダムな相関 ID を生成します。

セレクター

selector は、JMS 仕様に記載されている有効な JMS メッセージ・セレクターでなければなりません。

selector は、単方向サービスでのみ使用するオプションの属性です。selector を両方向サービスに指定した場合は無視されます。セレクターについては、[JMS のメッセージ・セレクター](#)を参照してください。

selector は、HTTP GET および HTTP DELETE 要求で戻されたメッセージを選択するときに使用されます。198 ページの『[ibm-mq-md-msgID](#)』ヘッダーまたは 199 ページの『[ibm-mq-md-correlID](#)』ヘッダーが指定されている場合、selector は無視されます。

セレクター文字のなかには、エンコードしないと server.xml に指定できないものがあります。このためには、以下のような一般的なメカニズムを使用します。

```
" becomes &quot;  
' becomes &apos;  
< becomes &lt;  
> becomes &gt;
```

persistence

persistence は、サービスで送信するメッセージの持続性を指定します。

persistence はオプションであり、MQMD [Persistence](#) フィールドを設定することと同等です。

値は、次のいずれか 1 つです。

false

非持続メッセージであることを意味します。これはデフォルト値です。

true

持続メッセージであることを意味します。

同じ値を取る `ibm-mq-md-persistence` HTTP ヘッダーを使用して、persistence をオーバーライドすることができます。

mqmdFormat

この属性を使用して、MQ Service Provider で送信するメッセージ内の MQMD フォーマット・フィールドの値を設定します。ただし、これは、この MQ Service Provider を z/OS Connect データ変換を使用するように構成した場合にのみ使用されます。それ以外の場合には無視されます。

この属性を指定しなかった場合にデータ変換が使用されると、MQMD フォーマット・フィールドが空白に設定されたメッセージが送信されます。この属性の値は 8 文字以下の長さにする必要があります。

userName

MQ Service Provider が認証と許可のために IBM MQ に渡すユーザー名。

この属性を指定しない場合、**connectionFactory** 属性によって参照される接続ファクトリー内の **userName** 属性が使用されます。

userName 属性を指定した場合は、指定した接続ファクトリーと MQ Service Provider の両方で、その MQ Service Provider 値が使用されます。

この属性を指定する場合は、**password** 属性を指定する必要があります。

パスワード

MQ Service Provider が認証と許可のために IBM MQ に渡すパスワード。

パスワードは平文で指定できますが、そうするべきではありません。代わりに、エンコード・オプションを使用して、z/OS Connect に付属の **securityUtility** ツールを使用してパスワードをエンコードする必要があります。詳しくは、[Liberty: securityUtility コマンド](#)を参照してください。

この属性を指定しない場合は、**connectionFactory** 属性で指定した接続ファクトリーのパスワード属性が使用されます。

パスワード属性を指定した場合は、指定した接続ファクトリーと MQ Service Provider の両方で、その MQ Service Provider 値が使用されます。

この属性を指定する場合は、**userName** 属性も指定する必要があります。

useCallerPrincipal

z/OS Connect に対して要求を発行したときに、呼び出し元の認証を z/OS Connect で行います。認証されたプリンシパルの名前を、認証と許可のために IBM MQ に渡すことができます。

これを行うには、**useCallerPrincipal** の値を true に設定します。

IBM MQ に接続するときには、プリンシパルの名前が(パスワードはなしで)使用されます。**password** 属性および **userName** 属性に指定された値はすべて無視されます。

receiveTextCCSID

データ変換を受け取り、`javax.jms.TextMessage` をコンシュームするとき(つまり、単方向サービスの HTTP GET または HTTP DELETE、あるいは両方向サービスの応答メッセージの取り出し)に使用する CCSID。

メッセージ内のテキストが、**receiveTextCCSID** で指定した CCSID に変換されます。

V 9.0.1

z/OS

MQ Service Provider で使用できる HTTP ヘッダー

MQ Service Provider に特定の HTTP ヘッダーが必要になるのは、HTTP POST を発行する場合のみです。

その場合は、Content-Type ヘッダーに「application/json」を設定する必要があります。このヘッダーの一部として文字セットを指定する場合、その値は utf-8 でなければなりません。

例えば、Content-Type=application/json;charset=utf-8 とします。

他の HTTP ヘッダーを HTTP 要求に指定して、MQ Service Provider の動作を変更することができます。それらについては、以降のセクションで詳しく説明します。その他のすべての HTTP ヘッダーは無視されます。

ibm-mq-md-msgID

このヘッダーは、単方向サービスに対して HTTP GET 要求または HTTP DELETE 要求を発行する場合に指定できます。

このヘッダー値を使用すると、指定のメッセージ ID を持つメッセージを選択するメッセージ・セレクターが生成されます。199 ページの『[ibm-mq-md-correlID](#)』ヘッダーも指定した場合は、両方の ID と一致するメッセージ・セレクターが生成されます。

このヘッダーのフォーマットの値について詳しくは、[msgId: HTTP x-msg-msgId エンティティ・ヘッダー](#)を参照してください。

ibm-mq-md-correlID

このヘッダーは、HTTP POST を発行する場合に指定できます。その場合、これは送信されるメッセージの MQMD の [CorrelID](#) フィールドを設定するために使用されます。

このヘッダーは、単方向サービスに対して HTTP GET 要求または DELETE 要求を発行する場合にも指定できます。このヘッダー値を使用すると、指定の相関 ID を持つメッセージを選択するメッセージ・セレクターが生成されます。198 ページの『[ibm-mq-md-msgID](#)』ヘッダーも指定した場合は、両方と一致するメッセージ・セレクターが生成されます。

このヘッダーのフォーマットの値について詳しくは、[correlId: HTTP x-msg-correlId エンティティ・ヘッダー](#)を参照してください。

ibm-mq-pmo-retain

バックエンドでトピックを使用している単方向サービスに対して HTTP POST 要求を発行する場合に、このヘッダーを TRUE の値にして指定することができます。これにより、保存パブリケーションが生成されます。詳しくは、[保存パブリケーション](#)を参照してください。

ibm-mq-usr

このヘッダーを使用して、単方向サービスと両方向サービスの両方に対する HTTP POST 要求の結果として送信される IBM MQ メッセージにメッセージ・プロパティを設定できます。

このヘッダーのフォーマットの値について詳しくは、[usr: HTTP x-msg-usr エンティティ・ヘッダー](#)を参照してください。

MQ Service Provider によって使用される名前は異なりますが、このヘッダーのフォーマットの値について詳しくは、[require-headers: HTTP x-msg-require-headers 要求ヘッダー](#)を参照してください。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

日本アイ・ビー・エム株式会社

法務・知的財産

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

〒 103-8510

103-8510

東京 103-8510、日本

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 INTERNATIONAL BUSINESS MACHINES CORPORATION は、法律上の瑕疵担保責任、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。"" 国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N

Rochester, MN 55901

U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、名前や住所が類似する個人や企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

プログラミング・インターフェース情報 (提供されている場合) は、このプログラムで使用するアプリケーション・ソフトウェアの作成を支援することを目的としています。

本書には、プログラムを作成するユーザーが WebSphere MQ のサービスを使用するためのプログラミング・インターフェースに関する情報が記載されています。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

重要: この診断、修正、およびチューニング情報は、変更される可能性があるため、プログラミング・インターフェースとして使用しないでください。

商標

IBM、IBM ロゴ、ibm.com® は、世界の多くの国で登録された IBM Corporation の商標です。現時点での IBM の商標リストについては、"Copyright and trademark information" www.ibm.com/legal/copytrade.shtml 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: