

7.5

IBM WebSphere 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、[145 ページの『特記事項』](#)に記載されている情報をお読みください。

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

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

© Copyright International Business Machines Corporation 2007 年, 2024.

目次

構成に関する参照情報.....	5
すべてのプラットフォームの構成例の情報.....	5
通信例の使用法.....	7
構成例 - IBM WebSphere MQ for Windows.....	8
構成例 - IBM WebSphere MQ for AIX.....	17
構成例 - IBM WebSphere MQ for HP-UX.....	23
構成例 - IBM WebSphere MQ for Solaris.....	29
構成例 - IBM WebSphere MQ for Linux.....	34
キュー名.....	40
その他のオブジェクト名.....	42
キュー名の解決.....	43
キュー名解決について.....	45
システムおよびデフォルト・オブジェクト.....	45
Windows デフォルト構成オブジェクト.....	49
SYSTEM.BASE.TOPIC.....	50
スタanzas情報.....	51
分散キューイング用の構成ファイル・スタanzas.....	53
チャンネルの属性.....	54
チャンネル属性とチャンネル・タイプ.....	55
チャンネル属性 (アルファベット順).....	58
IBM WebSphere MQ クラスター・コマンド.....	85
キュー・マネージャー定義コマンド.....	86
チャンネル定義コマンド.....	87
キュー定義コマンド.....	89
DISPLAY CLUSQMGR.....	92
SUSPEND QMGR と RESUME QMGR.....	93
REFRESH CLUSTER.....	94
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.....	95
ワークロード・バランシング.....	97
クラスター・ワークロード出口呼び出しとデータ構造体.....	109
チャンネル・プログラム.....	134
環境変数.....	135
分散プラットフォームにおけるメッセージ・チャンネルの計画例.....	140
例が示す事柄.....	140
例の実行.....	143
別名を使用して MQ ライブラリーを参照する.....	143
特記事項.....	145
プログラミング・インターフェース情報.....	146
商標.....	146

構成に関する参照情報

WebSphere 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。

関連タスク

構成

構成情報の例

構成の例により、WebSphere 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、WebSphere 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、[MQSC リファレンスの 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 WebSphere MQ オブジェクトの概念を表したものです。

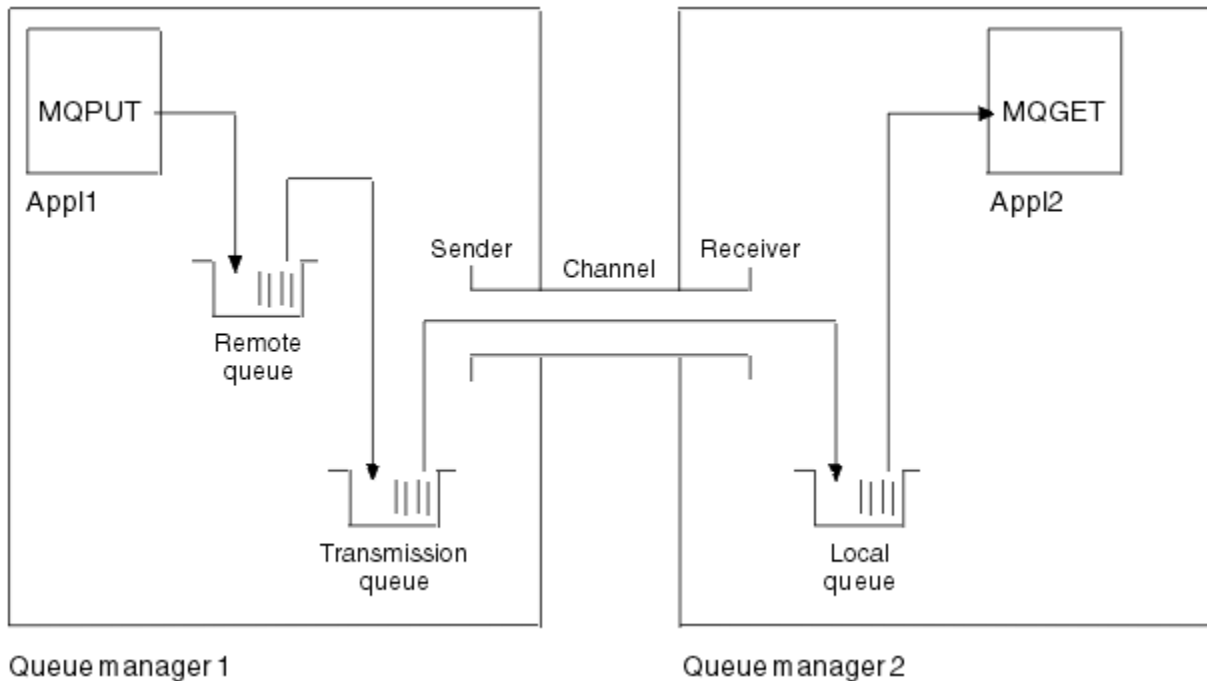


図 1. 構成例でセットアップする WebSphere MQ チャンネル

これは、WebSphere 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[®] は、トークンリングに接続されている 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 は隣接するローカル・エリア・ネットワーク (LAN) 上にある。この LAN も 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) に接続されています。
- 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 では、VTAM[®] およびネットワーク制御プログラム (NCP) 内の必須の定義はすべてそろっていて、LAN 接続プラットフォームに対して活動化されて広域ネットワーク (WAN) で通信す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 パーソナル・コミュニケーションズ Windows 版 V5.9
 - IBM Communications Server for AIX[®]、V6.3
 - Hewlett-Packard SNAplus2
 - IBM i
 - Data Connection SNAP-IX バージョン 7 以降
 - OS/390[®] バージョン 2 リリース 4
- TCP
 - Microsoft Windows

- AIX バージョン 4 リリース 1.4
- HP-UX バージョン 10.2 またはそれ以降
- Sun Solaris リリース 2.4 またはそれ以降
- IBM i
- TCP for z/OS
- HP Tru64 UNIX
- NetBIOS
- SPX

関連タスク

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。

Windows、UNIX、またはLinux® システムを使用している場合には、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。環境変数を設定するには、以下のコマンドのいずれかを入力します。

- Windows の場合：

```
MQ_INSTALLATION_PATH/bin/setmqenv
```

ここで、MQ_INSTALLATION_PATH は、IBM WebSphere MQ がインストールされているロケーションを参照します。

- UNIX and Linux システムの場合：

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

ここで、MQ_INSTALLATION_PATH は、IBM WebSphere MQ がインストールされているロケーションを参照します。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。

- [8 ページの『構成例 - IBM WebSphere MQ for Windows』](#)
- [17 ページの『構成例 - IBM WebSphere MQ for AIX』](#)
- [23 ページの『構成例 - IBM WebSphere MQ for HP-UX』](#)
- [29 ページの『構成例 - IBM WebSphere MQ for Solaris』](#)
- [34 ページの『構成例 - Linux の IBM WebSphere MQ』](#)

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おいては、ネットワーク構成をサポートするための 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あります。WebSphere MQ プラットフォームを WebSphere 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『[構成情報の例](#)』

構成の例により、WebSphere 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、WebSphere 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。

関連資料

[setmqenv](#)

構成例 - IBM WebSphere MQ for Windows

このセクションでは、その他のプラットフォーム上で IBM WebSphere MQ for Windows から IBM WebSphere 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、[11 ページの『IBM WebSphere MQ の構成』](#)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『構成情報の例』](#)を参照してください。

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

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

```
runmqclsr -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。 [11 ページの『IBM WebSphere MQ の構成』](#)へ進んでください。

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 WebSphere 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 WebSphere 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。論理アダプター番号 0 の IBM WebSphere MQ for Windows デフォルトは、インターネット・プロトコル・ネットワークで稼働中の 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) +
    CONNAME(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。

- 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 WebSphere 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、名前の数、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。

SPX 接続の確立

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

このセクションでは、以下の事項について説明しています。

- IPX/SPX パラメーター
- SPX アドレッシング
- SPX での受信

IPX/SPX パラメーター

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

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

SPX アドレッシング

WebSphere MQ は、各マシンの SPX アドレスを使用して、接続を確立します。SPX アドレスは、次の形式で指定されます。

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

この

network

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

node

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

socket

リモート・マシンが listen する 2 バイトのソケット番号。

WebSphere MQ が使用するデフォルトのソケット番号は 5E86 です。Windows レジストリーまたはキュー・マネージャー構成ファイル qm.ini にこの値を指定することによって、デフォルトのソケット番号を変更することができます。Windows レジストリー内の行は、次のようになります。

```
SPX:  
SOCKET=n
```

qm.ini の中に設定できる値の詳細については、[53 ページの『分散キューイング用の構成ファイル・スタンザ』](#)を参照してください。

SPX アドレスは、後で送信側のチャンネル定義の CONNAME パラメーターに指定します。接続されている WebSphere MQ システムが同一ネットワークに存在している場合は、ネットワーク・アドレスを指定する必要はありません。同様に、リモート・システムがデフォルトのソケット番号 (5E86) で listen している場合も、指定する必要はありません。CONNAME パラメーターの完全修飾 SPX アドレスは次のとおりです。

```
CONNAME('network.node(socket)')
```

ただし、これらのシステムが同じネットワーク上にあり、デフォルトのソケット番号を使用する場合、パラメーターは次のように設定します。

```
CONNAME(node)
```

チャンネルの構成パラメーターの詳細な例は、[11 ページの『IBM WebSphere MQ の構成』](#)にあります。

SPX での受信

受信側のチャンネル・プログラムは、送信側チャンネルからの開始要求に応じて開始されます。そのためには、リスナー・プログラムを開始して着信ネットワーク要求を検出し、関連のチャンネルを開始する必要があります。

WebSphere MQ リスナーを使用する必要があります。

WebSphere MQ リスナーの使用

WebSphere MQ が提供するリスナーは、新しいチャンネルをスレッドとして開始させますが、これを実行させるためには、RUNMQLSR コマンドを使用します。以下に例を示します。

```
RUNMQLSR -t spx
```

オプションで、デフォルト設定を使用しない場合に、キュー・マネージャー名またはソケット番号を指定できます。

IBM WebSphere MQ の構成

構成のためのプログラムとコマンドの例です。

注:

1. 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 WebSphere 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。
2. 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
```

3. エラー・ログは、ディレクトリー `MQ_INSTALLATION_PATH\qmgrs\qmgrname\errors` および `MQ_INSTALLATION_PATH\qmgrs\@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、WebSphere MQ がインストールされている上位ディレクトリーを表します。

4. 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。

デフォルト構成

WebSphere 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については、[Postcard アプリケーションを使用したインストールの検査](#)を参照してください。

基本構成

キュー・マネージャーの作成と開始は、IBM WebSphere MQ エクスプローラー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。

1. 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。

チャンネル構成

指定されたチャン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。

以下のセクションでは、[5 ページの図 1](#) で説明されているチャンネルを実装するために Windows キュー・マネージャーで実行する構成について詳しく説明します。

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。

WebSphere MQ for Windows と WebSphere 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 WebSphere 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 WebSphere 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、記載されておりに入力してください。

表 1. WebSphere 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	
WebSphere MQ for AIX との接続				
表のこのセクションで指定する値は、19 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	AIX	
D	リモート・キュー名		AIX リモート・キュー	
E	リモート・システムのキュー名	B	AIX ローカル・キュー	
F	伝送キュー名		AIX	
G	送信側 (SNA) チャンネル名		WINNT.AIX.SNA	
H	送信側 (TCP) チャンネル名		WINNT.AIX.TCP	
I	受信側 (SNA) チャンネル名	G	AIX.WINNT.SNA	
J	受信側 (TCP) チャンネル名	H	AIX.WINNT.TCP	
MQSeries® 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	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	
WebSphere MQ for HP-UX との接続				
表のこのセクションで指定する値は、25 ページの表 3 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	

表 1. WebSphere 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
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	
WebSphere MQ for Solaris との接続				
表のこのセクションで指定する値は、 31 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	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	
WebSphere MQ for Linux との接続				
表のこのセクションで指定する値は、 37 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		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	
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	
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	

表 1. WebSphere 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
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	
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		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	
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	

SNA を使用した IBM WebSphere 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) +
  conname(AIXCPIC) +                         18
  xmitq(AIX) +                               F
  replace

```

SNA を使用した IBM WebSphere 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

```

TCP/IP を使用した IBM WebSphere 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.TCP) chltype(sdr) +      H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(AIX) +                               F
  replace
```

TCP を使用した IBM WebSphere 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.TCP) chltype(rcvr) +    J
  trptype(tcp) +
  replace
```

自動始動

WebSphere MQ for Windows では、キュー・マネージャーとそのチャンネル・イニシエーター、チャンネル、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 WebSphere 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 WebSphere MQ サービスによって読み取られます。

詳しくは、[IBM WebSphere MQ の管理](#)を参照してください。

プロセスまたはスレッドとしてのチャンネルの実行

WebSphere MQ for Windows は、送信チャンネルを 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、送信側チャンネル定義の 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、多数の同時チャン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、送信チャン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・チャンネル・エージェント用に別個プロセスが必要とされます。

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・チャンネル・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、チャン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、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、チャン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。

WebSphere MQ for Windows で、WebSphere MQ エクスプローラーを使用して、レジストリー中に *PipeLineLength* パラメーターを設定します。

注：

1. *PipeLineLength* は V5.2 またはそれ以降の製品だけに適用されます。

2. パイプラインは TCP/IP チャン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ンネルの両側のキュー・マネージャーを構成しなければなりません。

チャン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。

構成例 - IBM WebSphere MQ for AIX

このセクションでは、その他のプラットフォーム上で IBM WebSphere MQ for AIX から IBM WebSphere 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
- HP Tru64 UNIX
- HP-UX
- 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『構成情報の例』](#)を参照してください。

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、オンラインの [IBM 資料 \(Communications Server for AIX\)](#) を参照してください。

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。

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

```
runmqtsr -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。以下に示す行がこのファイルにない場合には、示されているとおりに追加して、`MQ_INSTALLATION_PATH` を WebSphere MQ がインストールされている上位のディレクトリーと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` がありさえすれば、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。18 ページの『[IBM WebSphere MQ for AIX 構成](#)』へ進んでください。

IBM WebSphere MQ for AIX 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が不足している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増やすことができます。(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  
-- Journalled File Systems  
-- Change/Show Characteristics of a Journalled 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。`MQ_INSTALLATION_PATH` は WebSphere MQ がインストールされている上位のディレクトリーを表します。
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、WebSphere MQ 標準のトレース・コマンドまたは AIX システムのトレースを使用して、WebSphere MQ コンポーネントのトレースを開始できます。WebSphere MQ の Trace および AIX システムの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 コマンド行からキュー・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 コマンド行から **runmqsc** を開始して、それを使用して、次のコマンドを入力することによって未配布キュー・マネージャーを作成します。

```
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。

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。

以下のセクションでは、[5 ページの図 1](#) で説明されている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しく説明します。

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 コマンド行から **runmqsc** を開始し、各コマンドを順番に入力するか、コマンドをコマンド・ファイルに作成します。

WebSphere MQ for AIX と WebSphere 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 WebSphere 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 WebSphere 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 ローカル・キュー	

表 2. WebSphere 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
WebSphere MQ for Windows との接続				
表のこのセクションで指定する値は、13 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	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		AIX.WINNT.SNA	
H	送信側 (TCP/IP) チャネル名		AIX.WINNT.TCP	
I	受信側 (SNA) チャネル名	G	WINNT.AIX.SNA	
J	受信側 (TCP) チャネル名	H	WINNT.AIX.TCP	
WebSphere 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	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) チャネル名		DECUX.AIX.TCP	
J	受信側 (TCP) チャネル名	H	AIX.DECUX.TCP	
WebSphere MQ for HP-UX との接続				
表のこのセクションで指定する値は、25 ページの表 3 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) チャネル名		AIX.HPUX.SNA	
H	送信側 (TCP) チャネル名		AIX.HPUX.TCP	
I	受信側 (SNA) チャネル名	G	HPUX.AIX.SNA	
J	受信側 (TCP) チャネル名	H	HPUX.AIX.TCP	
WebSphere MQ for Solaris との接続				
表のこのセクションで指定する値は、31 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	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	

表 2. WebSphere 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.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	
WebSphere MQ for Linux との接続				
表のこのセクションで指定する値は、37 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		AIX.LINUX.SNA	
H	送信側 (TCP/IP) チャンネル名		AIX.LINUX.TCP	
I	受信側 (SNA) チャンネル名	G	LINUX.AIX.SNA	
J	受信側 (TCP/IP) チャンネル名	H	LINUX.AIX.TCP	
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		AIXAS400.SNA	
H	送信側 (TCP) チャンネル名		AIXAS400。TCP (TCP)	
I	受信側 (SNA) チャンネル名	G	AS400.AIX.SNA	
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		AIX.MVS.SNA	
H	送信側 (TCP) チャンネル名		AIX.MVS.TCP	
I	受信側 (SNA) チャンネル名	G	MVS。AIX.SNA	
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	

ID	パラメーター名	参照	使用例	ユーザー値
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	
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	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	

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

```
def ql (WINNT) + usage(xmitq) + replace F
def qr (WINNT.REMOTEQ) + rname(WINNT.LOCALQ) + rqmname(WINNT) + xmitq(WINNT) + replace D
E
C
F
def chl (AIX.WINNT.SNA) chltype(sdr) + trptype(lu62) + conname('WINNTCPIC') + xmitq(WINNT) + replace G
17
F
```

SNA を使用した IBM WebSphere 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) + trptype(lu62) + replace I
```

IBM WebSphere MQ for AIX の TPN 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/interop/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、*MQ_INSTALLATION_PATH*はWebSphere MQがインストールされている上位のディレクトリーです。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。

TCPを使用したIBM WebSphere 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
```

TCPを使用したIBM WebSphere 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
```

構成例 - IBM WebSphere MQ for HP-UX

このセクションでは、その他のプラットフォーム上でIBM WebSphere MQ for HP-UXからIBM WebSphere 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『構成情報の例』](#)を参照してください。

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、オンラインの IBM 資料 ([Communications Server](#))、およびオンラインの HP 資料 ([HP-UX SNAplus2 Installation Guide](#)) を参照してください。

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応答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。以下に示す行がこのファイルにない場合には、示されているとおりに追加して、`MQ_INSTALLATION_PATH` を WebSphere MQ がインストールされている上位のディレクトリーと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` がありさえすれば、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。24 ページの『[IBM WebSphere MQ for HP-UX 構成](#)』へ進んでください。

IBM WebSphere 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 WebSphere MQ がインストールされている上位のディレクトリーを表します。
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。

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。

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 ページの図 1](#) で説明され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。UNIX プロンプトから `runmqsc` を開始し、各コマンドを順番に入力するか、コマンドをコマンド・ファイルに作成します。

WebSphere MQ for HP-UX と WebSphere 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 WebSphere 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 WebSphere 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		HPUX	
B	ローカル・キュー名		HPUX.LOCALQ	

表 3. WebSphere 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
WebSphere MQ for Windows との接続				
表のこのセクションで指定する値は、13 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	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		HPUX.WINNT.SNA	
H	送信側 (TCP/IP) チャネル名		HPUX.WINNT.TCP	
I	受信側 (SNA) チャネル名	G	WINNT.HPUX.SNA	
J	受信側 (TCP) チャネル名	H	WINNT.HPUX.TCP	
WebSphere MQ for AIX との接続				
表のこのセクションで指定する値は、19 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	AIX	
D	リモート・キュー名		AIX リモート・キュー	
E	リモート・システムのキュー名	B	AIX ローカル・キュー	
F	伝送キュー名		AIX	
G	送信側 (SNA) チャネル名		HPUX.AIX.SNA	
H	送信側 (TCP) チャネル名		HPUX.AIX.TCP	
I	受信側 (SNA) チャネル名	G	AIX.HPUX.SNA	
J	受信側 (TCP) チャネル名	H	AIX.HPUX.TCP	
WebSphere 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	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) チャネル名		DECUX.HPUX.TCP	
J	受信側 (TCP) チャネル名	H	HPUX.DECUX.TCP	
WebSphere MQ for Solaris との接続				
表のこのセクションで指定する値は、31 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	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	

表 3. WebSphere 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
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	
WebSphere MQ for Linux との接続				
表のこのセクションで指定する値は、37 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	
AS400 との接続				
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	
I	受信側 (SNA) チャンネル名	G	AS400.HPUX.SNA	
MVS との接続				
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	
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	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	

表 3. WebSphere 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
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	

SNA を使用した IBM WebSphere 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.SNA) chltype(sdr) +  G
  trptype(lu62) +
  conname('WINNTCPIC') + 16
  xmitq(WINNT) +         F
  replace
```

SNA を使用した IBM WebSphere 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
```

IBM WebSphere 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 WebSphere MQ がインストールされている上位のディレクトリです。

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。

TCP を使用した IBM WebSphere 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
```

TCP/IP を使用した IBM WebSphere MQ for HP-UX 受信側チャ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
```

構成例 - IBM WebSphere MQ for Solaris

このセクションでは、その他のプラットフォーム上で IBM WebSphere MQ for Solaris から IBM WebSphere 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『構成情報の例』](#)を参照してください。

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、オンラインの IBM 資料 ([Communications Server](#))、オンラインの MetaSwitch 資料 ([SNAP-IX Administration Guide](#))、およびオンラインの Oracle 資料 ([Configuring Intersystem Communications \(ISC\)](#)) を参照してください。

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/amqcrista amqcrista  
[-m queue.manager.name]
```

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

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。30 ページの『[IBM WebSphere MQ for Solaris 構成](#)』へ進んでください。

IBM WebSphere MQ for Solaris 構成

構成を完了するために定義するチャン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。
`MQ_INSTALLATION_PATH` は、WebSphere MQ がインストールされている上位ディレクトリーを表します。
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。
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。

チャンネル構成

次のセクションでは、チャンネルを実現するために Solaris 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。

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、各コマンドを順番に入力するか、コマンドをコマンド・ファイルに作成します。

WebSphere MQ for Solaris と WebSphere 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 WebSphere 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 WebSphere 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		SOLARIS	
B	ローカル・キュー名		SOLARIS.LOCALQ	
WebSphere MQ for Windows との接続				
表のこのセクションで指定する値は、13 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	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	
WebSphere MQ for AIX との接続				
表のこのセクションで指定する値は、19 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	AIX	
D	リモート・キュー名		AIX リモート・キュー	

表 4. WebSphere 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	AIX ローカル・キュー	
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				
表のこのセクションで指定する値は、Compaq Tru64 UNIX システム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	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	
WebSphere MQ for HP-UX との接続				
表のこのセクションで指定する値は、25 ページの表 3 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		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	
WebSphere MQ for Linux との接続				
表のこのセクションで指定する値は、37 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	

表 4. WebSphere 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
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		SOLARIS.AS400.SNA	
H	送信側 (TCP) チャネル名		SOLARIS.AS400.TCP	
I	受信側 (SNA) チャネル名	G	AS400.SOLARIS.SNA	
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	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		SOLARIS.VSE.SNA	
I	受信側チャネル名	G	VSE.SOLARIS.SNA	

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

```

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

```

SNA を使用した IBM WebSphere 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.SNA) chltype(rcvr) + I
  trptype(lu62) +
  replace
```

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

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

TCP/IPを使用した IBM WebSphere 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
```

構成例- Linux の IBM WebSphere MQ

このセクションでは、その他のプラットフォーム上で IBM WebSphere MQ から IBM WebSphere 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『構成情報の例』](#)を参照してください。

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、WebSphere MQ for Linux (x86 プラットフォーム) にのみ適用されます。WebSphere MQ for Linux (x86-64 プラットフォーム)、WebSphere MQ for Linux (zSeries s390x プラットフォーム)、または WebSphere MQ for Linux (POWER®) 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](#)で、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 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、WebSphere MQ がインストールされている上位ディレクトリーを表します。

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/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
```

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

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、WebSphere 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ンザを追加します。

```
# WebSphere MQ service for XINETD
service MQSeries
{
    disable          = no
    flags            = REUSE
    socket_type      = stream
    wait            = no
    user            = mqm
    server          = MQ_INSTALLATION_PATH/bin/amqcrista
    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` ディレクトリーにファイルを作成するか、以前に作成した WebSphere 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。[36 ページの『IBM WebSphere MQ for Linux 構成』](#)へ進んでください。

IBM WebSphere 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 WebSphere MQ がインストールされている上位のディレクトリーを表します。
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。

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。

チャネル構成

以下のセクションでは、[5 ページの図 1](#) で説明されている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しく説明します。

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。`runmqsc` を UNIX プロンプトから開始し、各コマンドを順番に入力するか、コマンドをコマンド・ファイルに作成します。

WebSphere MQ for Linux および WebSphere MQ for HP-UX への接続の例を示します。他のプラットフォーム上の WebSphere 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 WebSphere 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		LINUX	
B	ローカル・キュー名		LINUX.LOCALQ	
WebSphere MQ for Windows との接続				
表のこのセクションで指定する値は、 13 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	WINNT	

表 5. WebSphere 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
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	

WebSphere MQ for AIX との接続

表のこのセクションで指定する値は、19 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	AIX	
D	リモート・キュー名		AIX リモート・キュー	
E	リモート・システムのキュー名	B	AIX ローカル・キュー	
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	
J	受信側 (TCP) チャネル名	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	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) チャネル名		DECUX.LINUX.TCP	
J	受信側 (TCP) チャネル名	H	LINUX.DECUX.TCP	

WebSphere MQ for HP-UX との接続

表のこのセクションで指定する値は、25 ページの表 3 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) チャネル名		LINUX.HPUX.SNA	
H	送信側 (TCP) チャネル名		LINUX.HPUX.TCP	
I	受信側 (SNA) チャネル名	G	HPUX.LINUX.SNA	

表 5. WebSphere 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/IP) チャンネル名	H	HPUX.LINUX.TCP	
WebSphere MQ for Solaris との接続				
表のこのセクションで指定する値は、31 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	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	
AS400 との接続				
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		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	
MVS との接続				
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	
Connection to MQSeries for VSE/ESA (WebSphere MQ for Linux (x86 プラットフォーム) のみ)				
表のこのセクションで指定する値は、VSE/ESA システム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	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	

SNA を使用した *IBM WebSphere 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
  rname(HPUX.LOCALQ) +                       E
  rqmname(HPUX) +                            C
  xmitq(HPUX) +                               F
  replace

def chl (LINUX.HPUX.SNA) chltype(sdr) +      G
  trptype(lu62) +
  conname('HPUXCPIC') +                     14
  xmitq(HPUX) +                               F
  replace
```

SNA を使用した *IBM WebSphere MQ for Linux (x86 プラットフォーム)* 受信側チャンネル定義
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
```

TCP を使用した *IBM WebSphere MQ for Linux* 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
  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
```

TCP/IP を使用した *IBM WebSphere MQ for Linux* 受信側チャンネル定義
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
```

キュー名

この情報は、キュー名の制約事項および予約済みのキュー名について理解するために使用します。

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。**ALTER** または **DEFINE REPLACE** コマンドを使用して、ご使用のシステムに合わせてこれらのキュー定義を変更することができます。IBM WebSphere 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)
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 WebSphere 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・マネージャー間のリポジトリ変更の通信に使用するキュー (AIX、HP-UX、Linux、IBM i、Solaris、Windows、および z/OS のみ)
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	リポジトリに関する情報を保持するために使用するキュー (AIX、HP-UX、Linux、IBM i、Solaris、Windows、および z/OS のみ)
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・サポートによって管理されるすべての宛先用の伝送キュー (AIX、HP-UX、Linux、IBM i、Solaris、Windows、および z/OS のみ)
SYSTEM.COMMAND.INPUT	z/OS
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

キュー名	説明
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 WebSphere 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・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・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 WebSphere MQ Explorer
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)
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 WebSphere MQ 内部選択評価キュー (z/OS を除く)
SYSTEM.SELECTION.VALIDATION.QUEUE	IBM WebSphere 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名

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 WebSphere 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

オブジェクト名	説明
SYSTEM.AUTO.RECEIVER	自動定義のデフォルト受信側チャンネル (Windows、 UNIX and Linux システムのみ)
SYSTEM.AUTO.SVRCONN	自動定義のデフォルト・サーバー接続チャンネル (IBM i、 Windows、 UNIX and Linux システムのみ)
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
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 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 TCP/IP リスナー (IBM i、 Windows、 UNIX and Linux システムのみ)
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・サービス (IBM i、 Windows、 UNIX and Linux システムのみ)
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

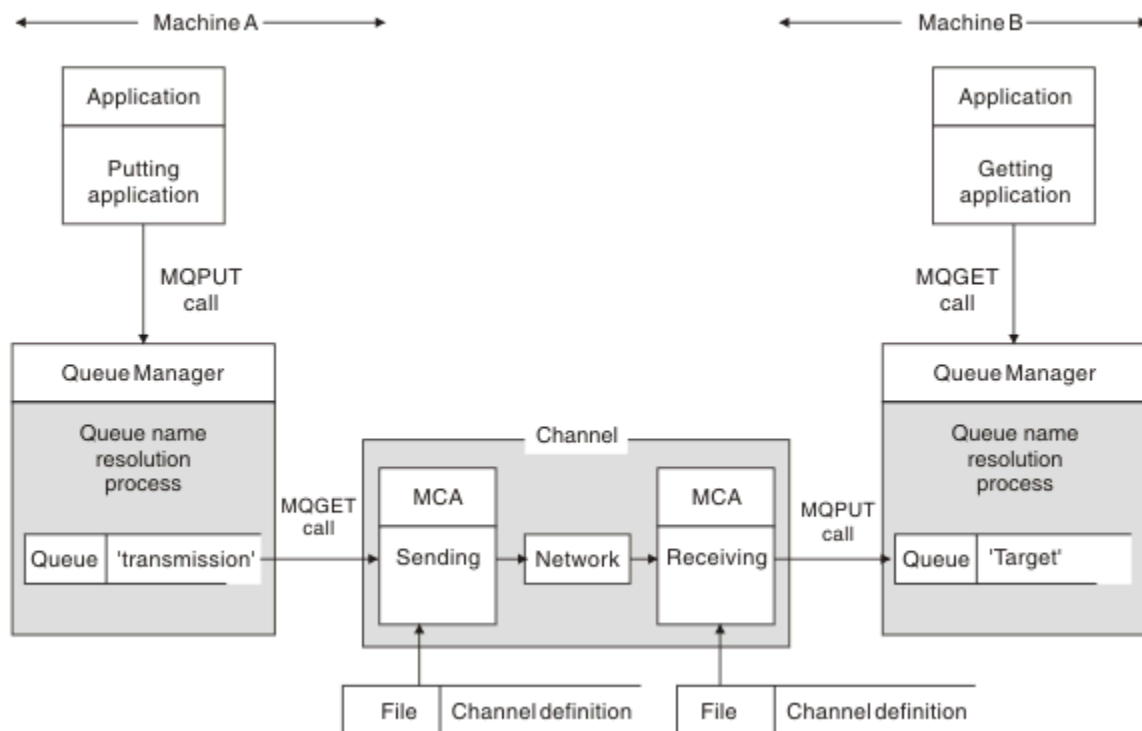


図 2. 名前の解決

44 ページの図 2 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。

- 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。
- 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ケーションは、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` コマンドによって作成されたシステム・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 WebSphere 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。

- [46 ページの表 6](#) 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。
- [47 ページの表 7](#) 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。
- [47 ページの表 8](#) 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。
- [48 ページの表 9](#) 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。
- [48 ページの表 10](#) 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。
- [48 ページの表 11](#) 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。
- [48 ページの表 12](#) 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。
- [49 ページの表 13](#) 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

表 6. 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.CHLAUTH.DATA.QUEUE	IBM WebSphere 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
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 WebSphere MQ Explorer 応答先キュー。これは、IBM WebSphere 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 7. 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
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。
SYSTEM.DEFAULT.TOPIC	デフォルトのトピック定義

表 8. 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 8. 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.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
SYSTEM.DEF.CLNTCONN	デフォルト・クライアント接続チャンネル

表 9. 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 10. 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 11. 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 12. 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 13. 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
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、WebSphere 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、WebSphere 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。

WebSphere 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、49 ページの表 14 に示したオブジェクトが作成されます。

表 14. 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 <code>QM_</code> 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、チャネル・リスナー、およびチャネル・イニシエーターが関連付けられます。チャネル・リスナーは、ポート番号 1414 の標準 WebSphere MQ ポートで <code>listen</code> 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ネル	先頭に文字 <code>TO_QM_</code> 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ネル	クラスター送信側チャネルは、最初は <code>TO_ + QMNAME +</code> という名前で作成されます。WebSphere 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 <code>TO_</code> が付きます。クラスター送信側チャネル名の最大長は 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 <code>default</code> と呼ばれます。

表 14. 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
WebSphere 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	WebSphere 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、[49 ページ](#)の表 14 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 15. SYSTEM.BASE.TOPIC のデフォルト値

パラメーター	値
TOPICSTR	"
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
PUBSCOPE	ALL

表 15. SYSTEM.BASE.TOPIC のデフォルト値 (続き)

パラメーター	値
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 WebSphere 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。

スタンプザ情報

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

スタンプザの構成

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 WebSphere MQ 構成情報の変更は](#)、以下の構成に役立ちます。
 - AllQueueManagers スタンプザ
 - DefaultQueueManager スタンプザ
 - ExitProperties スタンプザ
 - LogDefaults スタンプザ
 - qm.ini ファイルの Security スタンプザ
- [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 スタンプザ (Windows のみ)
 - Service スタンプザ - インストール可能なサービス用
 - Log スタンプザ
 - RestrictedMode スタンプザ (UNIX and Linux システムのみ)
 - XAResourceManager スタンプザ
 - TCP、LU62、および NETBIOS スタンプザ
 - ExitPath スタンプザ
 - QMErrorLog スタンプザ
 - SSL スタンプザ
 - ExitPropertiesLocal スタンプザ
- [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 スタンプザ

– *ServiceComponent* スタンザ

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* スタンザ

– *ApplicationTrace* スタンザ

• 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* スタンザ

– *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

– *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

• 53 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。

• [mqc.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。

mqc.ini ファイル

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

mqc.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 WebSphere 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](#)
- セキュリティー
- [Service](#) および [ServiceComponent](#)

[InstallableServices](#) を構成する方法は次のとおりです。

- UNIX and Linux プラットフォームでは、[Service](#) および [ServiceComponent](#) スタンザを使用します。
- Windows では、**regedit** 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、[crtmqm](#) コマンドを使用して以下のプロパティを変更できます。

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

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 出口を構成することが必要な場合もあります。

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、Windows、UNIX and Linux システム上の IBM WebSphere MQ 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
- SPX (Windows XP および Windows 2003 Server のみ)
- EXITPATH

54 ページの図 3 には、これらのスタンザを使用して設定できる値を示しています。どれか 1 つのスタンザを定義する場合、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".
  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.
  ADOPTNEWMCACHECK=    ; Specifies the type checking required.
                        ; The default is "NAME", "ADDRESS", and "QM".
  typecheck
  TCP:
  PORT=n                ; Port number, the default is 1414
  KEEPALIVE=Yes         ; Switch TCP/IP KeepAlive on
  LIBRARY2=DLLName2    ; Used if code is in two libraries
  EXITPATH:2           ; Location of user exits (MQSeries for AIX,
                        ; HP-UX, and Solaris only)
  EXITPATHS=           ; String of directory paths.
```

図 3. 分散キューイングの *qm.ini* スタンザ

注:

1. MQIBINDTYPE は、IBM WebSphere MQ for AIX、IBM WebSphere MQ for HP-UX、および IBM WebSphere MQ for Solaris にのみ適用されます。
2. EXITPATH は、IBM WebSphere MQ for AIX、IBM WebSphere MQ for HP-UX、および IBM WebSphere MQ for Solaris にのみ適用されます。

関連情報

構成

[Windows、UNIX、および Linux システムでの構成情報の変更](#)

チャンネルの属性

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

この情報は、プロダクト・センシティブ・プログラミング・インターフェース情報です。

チャンネルごとに特定の一組の環境に最適のチャンネル属性を選択してください。ただし、チャン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。[チャン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、ほとんどのチャン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。

注：WebSphere 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、運用システムのシステム・デフォルト・チャンネルから値が取られることを表します。

チャンネル属性とチャンネル・タイプ

さまざまなタイプのチャンネルが、さまざまなタイプのチャンネル属性をサポートします。

WebSphere MQ チャンネル属性のチャンネル・タイプを、55 ページの表 16 にリストします。

属性フィールド	MQSC コマンド・パラメーター	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
Alter date (変更日付)	ALTDATE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Alter time (変更時刻)	ALTTIME	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
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
バッチのサイズ	BATCHSZ	Yes	Yes	Yes	Yes			Yes	Yes
チャンネル名	CHANNEL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
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
クライアント・チャンネルの重み	CLNTWGHT					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)	CLWLPRTY							Yes	Yes
Cluster workload rank (クラスター・ワークロードのランク)	CLWLRANK							Yes	Yes
Cluster workload weight (クラスター・ワークロード・ウェイト)	CLWLWGHT							Yes	Yes
Connection affinity (接続アフィニティー)	AFFINITY					Yes			
接続名	CONNNAME	Yes	Yes		Yes	Yes		Yes	Yes
メッセージ変換	CONVERT	Yes	Yes					Yes	Yes
Data compression (データ圧縮)	COMPMSG	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
説明	DESCR	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
切断インターバル	DISCINT	Yes	Yes					Yes	Yes

表 16. 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・パラメーター	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
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
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		
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
メッセージ・チャンネル・エージェント名	MCANAME	Yes	Yes		Yes			Yes	Yes
Message channel agent type (メッセージ・チャンネル・エージェント・タイプ)	MCATYPE	Yes	Yes		Yes			Yes	Yes
メッセージ・チャンネル・エージェント・ユーザー	MCAUSER	Yes	Yes	Yes	Yes		Yes	Yes	Yes
メッセージ出口名	MSGEXIT	Yes	Yes	Yes	Yes			Yes	Yes
メッセージ出口ユーザー・データ	MSGDATA	Yes	Yes	Yes	Yes			Yes	Yes
メッセージ再試行出口名	MREXIT			Yes	Yes				Yes
Message-retry exit user data (メッセージ再試行出口ユーザー・データ)	MRDATA			Yes	Yes				Yes
メッセージ再試行カウント	MRRTY			Yes	Yes				Yes
メッセージ再試行間隔	MRTMR			Yes	Yes				Yes
モニター	MONCHL	Yes	Yes	Yes	Yes		Yes	Yes	Yes

表 16. 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・パラメーター	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
<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	
<u>プロパティ制御</u>	PROPCTL	Yes	Yes					Yes	Yes
<u>PUT authority (PUT 権限)</u>	PUTAUT			Yes	Yes				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>セキュリティー出口名</u>	SCYEXIT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<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					はい	はい		
<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
<u>SSL クライアント認証</u>	SSLCAUTH		Yes	Yes	Yes		Yes		Yes
<u>SSL ピア</u>	SSLPEER	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<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
<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	

関連概念

58 ページの『チャンネル属性 (アルファベット順)』

このセクションでは、チャンネル・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。

関連資料

[MQSC リファレンス](#)

チャンネル属性 (アルファベット順)

このセクションでは、チャンネル・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 WebSphere 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 (ALTDAT)

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、すべてのチャンネル・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、すべてのチャンネル・タイプで有効です。

バッチ・ハートビート間隔 (BATCHHB)

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、バッチをバックアウトできるため、バッチが未確定とならずに済みます。バッチをバックアウトすることによって、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。

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。

- 送信者
- 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、BATCHSZ で指定されたメッセージ数が送信されるか、または伝送キューが空になると、バッチはクローズします。伝送キューがしばしば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で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。
- 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。

- 送信者
- サーバー
- クラスター送信側
- クラスター受信側

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。値 1 は、WebSphere MQ for MVS 以外のバージョン 1 製品で使用するためのものです。

- 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。

チャンネル統計 (STATCHL)

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 では、この属性はサポートされていません。

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 (WebSphere MQ for Windows システム、および UNIX システムのみ)
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

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

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・バランシングが可能になる)、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・バランシングが実行されず、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・バランシングを有効にするには、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、WebSphere MQ オブジェクトの命名規則に従い、48 文字です。

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

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

クラスター・ワークロード優先順位 (CLWLPRTY)

この属性は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 が最高の優先順位です。

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・ワークロード・ランク (CLWLRANK)

この属性はチャンネルのランクを指定します。

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 が最高のランクです。

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・ワークロード・ウェイト (CLWLWGHT)

この属性は、チャンネルに重み付けの要因を適用するため、そのチャンネルに送信されるメッセージの比率を制御できます。

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 が最高の重み付けです。

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

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 (完全管理対象 .NET を含む) で作成されたクライアント・アプリケーションの場合、および IBM WebSphere MQ classes for Java および IBM WebSphere MQ classes for JMS を使用するアプリケーションの場合、リストの作成後に CCDT が変更されていると、リストが更新されます。

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 (完全管理対象 .NET を含む) で作成されたクライアント・アプリケーションの場合、および IBM WebSphere MQ classes for Java および IBM WebSphere MQ classes for JMS を使用するアプリケーションの場合、リストの作成後に CCDT が変更されていると、リストが更新されます。

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、同じプロパティー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。CLNTWGHT 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 Version 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 Version 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。

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

名前は、z/OS では最大 48 文字 (注 1 を参照)、その他のプラットフォームでは最大 264 文字です。さらに、以下のことが適用されます。

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。またこれには、z/OS 動的 DNS グループまたはネットワーク・ディスプレイ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、接続名は使用される IP スタックを決定するローカル・アドレスと相互作用します。詳細については、69 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

IBM i、Windows システム、および UNIX システムの WebSphere MQ の場合は、TPNAME および MODENAME の指定時にはパートナー LU の完全修飾名を指定します。上記以外のバージョンのとき、あるいは 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用に、CPI-C サイド情報オブジェクト名を指定します。

z/OS では、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・マネージャーの論理装置名。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。

注: 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・マネージャー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。

注: クラスター受信側チャンネルにおけるサイド情報は、クラスター内の他のキュー・マネージャーに関するものです。あるいは、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・マネージャーの適切な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 の総称リソース・グループの名前でかまいません。

伝送プロトコルが NetBIOS である場合

CONNAME 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 WebSphere 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. この 48 文字の長さ制限を回避するには、以下の方法のいずれかが考えられます。
 - 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 [84 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。

- 送信者
- 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、67 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。

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。

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。

サーバー接続の非活動間隔は、クライアントからの WebSphere 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 DISCONT 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。この値はデフォルトです。

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。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。

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 WebSphere 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 Version 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 \(MQLONG\)](#) を参照してください。

関連資料

[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。z/OS 以外のプラットフォームでは、このパラメーターにアクセスおよび変更できますが、保管と転送のみです。パラメーターが機能的に実装されるわけではありません。KAINT パラメーターの機能が必要な場合、68 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。他のプラットフォームでこのような状態にするには、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 スタンザに指定するか、IBM WebSphere MQ エクスプローラーを通して指定します。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オンに切り替える必要があります。

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。KAIN T 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。KAIN T 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。

折衝された HBINT	KAIN T
>0	折衝された HBINT + 60 秒
0	0

KAIN T に AUTO が設定されており、かつサーバー接続チャンネルである場合、キープアライブ間隔の代わりに TCP INTERVAL の値が使用されます。

この属性は、すべてのチャンネル・タイプで有効です。

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。

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 を使用すると、デュアル・スタック・システム上で IPv4 または IPv6 スタックを使用するように、またはシングル・スタック・システム上でデュアル・モード・スタックを使用するようにチャンネルを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む 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。

追加のローカル・アドレスごとに [, [ip-addr][(low-port[,high-port])]] を複数回指定します。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。[, [ip-addr][(low-port[,high-port])]] を使用して、マルチインスタンス・キュー・マネージャー構成の一部である、さまざまなサーバー上の特定のローカル・ネットワーク・アドレスを表すこともできます。

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

IPv6 16 進表記

例えば 2001:DB8:0:0:0:0:0:0

英数字のホスト名書式

例えば WWW.EXAMPLE.COM

low-port and 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、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。
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と、エラーが生成され、チャンネルの開始は失敗します。

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、接続名 (CONNAME) によって、どちらのスタックを使用するかが決まります。ローカル・アドレス (LOCLADDR) および接続名 (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、使用するスタックは、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 IPv4 および IPv6 のデュアル・スタックが構成され、チャンネルのローカル・アドレス (LOCLADDR) が指定されていない場合、使用するスタックはチャンネルで指定された接続名 (CONNAME) によって決定されます。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、使用するスタックは、キュー・マネージャー属性 IPADDRV によって決定されます。

distributed 分散プラットフォームでは、ローカル・アドレスが定義されていないすべての送信側チャ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 は使用する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。

関連概念

[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。

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

はその後、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る自動的な始動はありません)。

(再試行が成功しそうな原因で失敗した場合には、再試行は試みられません。)

チャンネル・イニシエーター (z/OS) またはチャンネル (分散プラットフォーム) 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 (分散プラットフォーム) 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。

注: IBM i、UNIX システム、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。

長期再試行カウント 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

注: UNIX システム、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。

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、短い再試行間隔で *short retry count* 回試行した後、この長い間隔で *long retry count* 回接続を試行します。

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。

- 送信者
- 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、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

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の通信の設定方法について詳しくは、[分散キューイングを使用したアプリケーションの接続](#)を参照してください。

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

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の、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。

クライアント接続機構 (CAF) は、z/OS へのクライアントの接続をサポートする WebSphere MQ for z/OS のオプションです。クライアント接続機能 (CAF) がインストールされていない場合、SYSTEM.ADMIN.SVRCONN チャンネルにおいて属性に設定できるのは 0 から 5 までだけです。5 より大きい値は、CAF がインストールされていなければ、0 と解釈されます。

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。

この属性は、サーバー接続チャンネルにのみ有効です。

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。

クライアント接続機構 (CAF) は、z/OS へのクライアントの接続をサポートする WebSphere MQ for z/OS のオプションです。クライアント接続機能 (CAF) がインストールされていない場合、SYSTEM.ADMIN.SVRCONN チャンネルにおいて属性に設定できるのは 0 から 5 までだけです。5 より大きい値は、CAF がインストールされていなければ、0 と解釈されます。

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を下回っても、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。

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。

WebSphere MQ for UNIX システム、および Windows システムの場合は、キュー・マネージャーの最大メッセージ長以下の、ゼロ以上の値を指定します。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。

プラットフォームごとに WebSphere 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、[WebSphere MQ 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。

WebSphere 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、`RUNMSLSR` 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。

WebSphere 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。

- 送信者
- 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 WebSphere 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 WebSphere 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。このときの `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。

この属性を空白にした場合、`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

関連概念

[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、空白にしておきます。

この属性の形式および最大長は、[79 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

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。80 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

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、79 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・データ収集がオフに切り替わります。

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。

終

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。

- 送信者
- 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・データについて詳しくは、[キューと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。

ネットワーク接続優先順位 (NETPRTY)

この属性で、ネットワーク接続の優先順位を指定します。

分散キューイングでは、使用可能な複数のパスがある場合、優先度が最も高いパスが選択されます。この値は、0 から 9 の範囲でなければなりません。0 が最低の優先順位です。

この属性は、以下のチャンネル・タイプで有効です。

- クラスター受信側

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。

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

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。

WebSphere 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。

- 送信者
- サーバー
- 要求側
- クライアント 接続
- クラスター送信側

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

- 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)

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) と呼ばれます。

デフォルトのユーザー ID が使用されます。

どのプラットフォームにおいても、キューに対するオープン権限の検査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。

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・セキュリティー (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。

どのプラットフォームにおいても、MQOO_SET_ALL_CONTEXT および MQOO_ALTERNATE_USER_AUTHORITY でキューに対するオープン権限の検査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 のキューに対するオープン権限を検査するために使用されるユーザー ID は、メッセージ記述子内の *UserIdentifier* です。

コンテキスト・セキュリティー (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。

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。

セキュリティーの詳細については、以下を参照してください。

- [セキュリティー](#)
- [Windows、UNIX and Linux システムでのセキュリティーのセットアップ \(WebSphere MQ UNIX システムおよび Windows システムの場合\)](#)

キュー・マネージャー名 (QMNAME)

この属性は、WebSphere MQ MQI クライアント・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。

WebSphere 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空白を埋め込みます)。ストリングの最大長は 20 文字です。

- Windows では、以下の形式になります。

```
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 システムでは、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。WebSphere MQ for IBM i では、最大 10 個の出口名をリストできます。WebSphere 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)
```

WebSphere MQ for UNIX システム、および Windows システムでは、ユーザー・データの出口名とストリングのストリングの長さは 500 文字に制限されています。WebSphere 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。WebSphere 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、これを空白のままにします。

名前の形式および最大長は、[79 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、空白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、[79 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。[80 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

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。

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、チャンネルは終了します。

(再試行が成功しそうな原因で失敗した場合には、再試行は試みられません。)

チャンネル・イニシエーター (z/OS) またはチャンネル (分散プラットフォーム) 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 (分散プラットフォーム) 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。

注: UNIX システム、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。

この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

注: UNIX システム、および Windows システムでは、再試行が実行されるために、チャンネル・イニシエーターが稼働し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。

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。

- 送信者
- 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 接続または SSL 接続に対して単一の CipherSpec を指定します。

各 WebSphere 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 属性はブランク値にすることができます。これは、SSL または TLS を使用しないという意味です。チャンネルの片方の SSLCIPH 属性がブランク値の場合は、もう片方の SSLCIPH 属性もブランク値である必要があります。
- また、SSLCIPH に非ブランク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 SSL または 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。

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 SSL クライアントから SSL 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 SSL 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

SSL 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。

SSLCAUTH の値を、SSLCIPH が欠落またはブランクになっている非 SSL チャンネル定義に指定できます。

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
- クラスター受信側

SSLCAUTH の詳細については、[MQSC リファレンス](#)および[セキュリティー](#)を参照してください。

SSL ピア (SSLPEER)

この属性は、IBM WebSphere 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: SSL または 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、SSL または 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。

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 が欠落しているか空白になっている非 SSL 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 SSL 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、SSL 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、「[MQSC リファレンス](#)」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。

関連概念

[チャンネル認証レコード](#)

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。

- 送信者
- 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。

指定できる値は以下のとおりです。

LU62	LU 6.2
TCP	TCP/IP
NETBIOS	NetBIOS (84 ページの『1』)
SPX	SPX (84 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す。	

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。

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 の属性で送達不能キューの名前が指定されている場合は、その名前が使用されます。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 WebSphere 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。

WebSphere 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。

- 送信者
- サーバー
- 要求側
- クライアント 接続
- クラスタ送信側

IBM WebSphere MQ クラスタ・コマンド

IBM WebSphere MQ スクリプト・コマンドである **runmqsc** コマンドには、クラスタ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 WebSphere 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します。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 では、**CSQUTIL** ユーティリティの COMMAND 機能、操作パネル、および制御パネルを使用することも、z/OS コンソールを使用することもできます。
- 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 WebSphere MQ エクスプローラー

IBM WebSphere MQ for z/OS バージョン 6 以降では、エクスプローラー GUI で、リポジトリ・キュー・マネージャーを含むクラスタを管理できます。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 WebSphere MQ for z/OS では、IBM WebSphere MQ エクスプローラーは、リポジトリ・キュー・マネージャーを含むクラスタを管理できません。そのため、IBM WebSphere MQ エクスプローラーが管理できるシステム上で、さらに別のリポジトリを指定する必要があります。

IBM WebSphere MQ for Windows および WebSphere MQ for Linux では、IBM WebSphere MQ エクスプローラー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 WebSphere MQ エクスプローラー・クライアントを使用することもできます。

IBM WebSphere MQ エクスプローラーを使用すると、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 WebSphere MQ エクスプローラーには 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

関連概念

85 ページの『IBM WebSphere MQ クラスター・コマンド』

IBM WebSphere 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。

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 コマンドを使用して、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削減します。例えば、このキュー・マネージャーの保守を実行する前に、このキュー・マネージャーを復元します。

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: Forcibly removing a queue manager from a cluster

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の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 WebSphere 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 コマンドおよび ALTER CHANNEL コマンドの以下の属性は、クラスター・チャンネルに固有のものです。

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、WebSphere 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 CLUSSDR 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。ただし、DISPLAY CLUSQMGR コマンドを使用して、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。

AIX, HP-UX, IBM i, Linux, Solaris, and Windows 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 WebSphere 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 にブランクを指定した場合は、WebSphere 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 が当初はブランク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。ただし、ローカル・キュー・マネージャーから DISPLAY CHANNEL コマンドを発行すると、CONNAME がブランクになります。

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、WebSphere 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 コマンドを使用して、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削減します。例えば、このキュー・マネージャーの保守を実行する前に、このキュー・マネージャーを復元します。

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: Forcibly removing a queue manager from a cluster](#)

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。

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 を指定する場合、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 Version 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 Version 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。

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 または 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 (キューのインスタンスが複数ある場合は、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 コマンドを使用して、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削減します。例えば、このキュー・マネージャーの保守を実行する前に、このキュー・マネージャーを復元します。

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: Forcibly removing a queue manager from a cluster

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。

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。

CLUSSDR

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。

CLUSSDRA

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 CLUSDRB の DEFTYPE に変換するように定義する必要があるからです。

CLUSDRB

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。

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。

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。

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。

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。

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。

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。このプロパティは、z/OS 以外のプラットフォームでのみ使用可能です。

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。

SUSPEND QMGR、RESUME QMGR、およびクラスター

SUSPEND QMGR および RESUME QMGR コマンドを使用して、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削減します。例えば、このキュー・マネージャーの保守を実行する前に、このキュー・マネージャーを復元します。

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: Forcibly removing a queue manager from a cluster

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。

SUSPEND QMGR、RESUME QMGR、およびクラスター

SUSPEND QMGR および RESUME QMGR コマンドを使用して、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削減します。例えば、このキュー・マネージャーの保守を実行する前に、このキュー・マネージャーを復元します。

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。

関連タスク

キュー・マネージャーの保守

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: Forcibly removing a queue manager from a cluster](#)

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。

[SUSPEND QMGR](#)

[RESUME QMGR](#)

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: REFRESH CLUSTER コマンドを使用すると、処理中のクラスターに悪影響が及ぶ可能性があります。例えば、完全リポジトリで、キュー・マネージャー・クラスター・リソースを再伝搬している最中に、

作業負荷が急激に増加するというようなことが起こります。そのような理由から、このコマンドは可能な限り日常の作業では使用せず、別の方法で個々の不整合を修正することが最善策です。

関連概念

[クラスター化: REFRESH CLUSTER と履歴キュー](#)

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 コマンドを使用して、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削減します。例えば、このキュー・マネージャーの保守を実行する前に、このキュー・マネージャーを復元します。

RESET CLUSTER: Forcibly removing a queue manager from a cluster

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。

RESET CLUSTER: Forcibly removing a queue manager from a cluster

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 WebSphere 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。その後、損傷したキュー・マネージャーが再び機能するようになったら、**REFRESH CLUSTER** コマンドを使用して**RESET CLUSTER**の効果を反転し、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、**REFRESH QMGR TYPE (PROXYSUB)** コマンドを発行して、必要なプロキシ・サブスクリプションを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](#)。

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・マネージャーを除去する正しい方法については、[クラスターからキュー・マネージャーを除去する](#)を参照してください。

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** コマンドを使用して、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削減します。例えば、このキュー・マネージャーの保守を実行する前に、このキュー・マネージャーを復元します。

[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 \(クラスターのリセット\)](#)

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、WebSphere 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。クラスター・ワークロード出口プログラムを作成することにより、キュー・マネージャーを選択するワークロード・バランシング・アルゴリズムを提供できます。

最適な宛先は、キュー・マネージャーとキューの可用性とともに、キュー・マネージャー、キュー、およびチャンネルに関連付けられたいくつかのクラスター・ワークロード固有の属性に基づいて選択されます。

ワークロード・バランシング・アルゴリズムの結果が要件に適合しない場合には、独自にクラスター・ワークロード・ユーザー出口プログラムを作成できます。この出口を使用して、クラスター内の自分が選択したキューにメッセージを送付します。

関連概念

クラスター・ワークロード出口呼び出しとデータ構造体

ここでは、クラスター・ワークロード出口と、出口で使用されるデータ構造体について説明します。ここで説明するのは、汎用プログラミング・インターフェース情報です。

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。

このセクションでは、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をリストします。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">• CLWLPRTY• CLWLRANK• CLWLUSEQ	<ul style="list-style-type: none">• CLWLUSEQ• CLWLMRUC	<ul style="list-style-type: none">• CLWLPRTY• CLWLRANK• CLWLWGHT• NETPRTY

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 (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) に対する固有の経路、つまりチャンネルを検索します。

アルゴリズムの後続のステップで、有効な宛先のリストから宛先を除去します。

1. キュー名が指定されている場合は、次のようになります。
 - a. 書き込みが使用可能になっていないキューは、有効な宛先から除去されます。
 - b. ローカル・キュー・マネージャーとクラスターを共用しないキューのリモート・インスタンスは除去されます。
 - c. キューと同じクラスター内にはないリモート CLUSRCVR チャンネルは除去されます。
2. キュー・マネージャー名が指定されている場合は、次のようになります。
 - a. 書き込みが使用可能になっていないキュー・マネージャー別名は除去されます。
 - b. ローカル・キュー・マネージャーと同じクラスターにはないリモート CLUSRCVR チャンネルは除去されます。

3. この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、通常はキューのローカル・インスタンスが使用されます。以下の3つ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。
 - 以下の両方の条件を満たす。
 - a. キューの「キューの使用」属性 CLWLUSEQ が QMGR に設定されている。
 - b. 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・チャンネルを介して受信される。

注：以下の MQWXP_PUT_BY_CLUSTER_CH および MQQF_CLWL_USEQ_ANY フラグが両方とも設定されていない場合は、ユーザー出口内のクラスター・チャンネルからメッセージを検出できます。

 - MQWXP.Flags のフラグ MQWXP_PUT_BY_CLUSTER_CH。
 - MQWQR.QFlags のフラグ MQQF_CLWL_USEQ_ANY。
4. メッセージがクラスター PCF メッセージの場合、パブリケーションまたはサブスクリプションが既に送信されたキュー・マネージャーは除去されます。
5. キュー・マネージャーに対するチャンネルまたはキュー・マネージャー別名のうち、CLWLRANK が残りのすべてのチャンネルまたはキュー・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。
6. 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。
7. 1つのキューのリモート・インスタンスのみが残っている場合、再開されたキュー・マネージャーは、中断されたキュー・マネージャーよりも優先して選択されます。
8. キューの複数のリモート・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 (MQCHS_RETRYING) のすべてのチャンネルが含まれます。
11. キューのリモート・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
12. キューの複数のリモート・インスタンスが残っていて、メッセージがクラスター PCF メッセージである場合は、ローカルに定義された CLUSSDR チャンネルが選択されます。
13. いずれかのキュー・マネージャーに対するキューの複数のリモート・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。
14. キュー・マネージャ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。

15. 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。

16. MQWDR.DestSeqNumber で示される最大値を持ついくつかのチャンネル以外、すべてのチャンネルが除去されます。残りのチャンネルの数が、最後に使用されたチャンネルの最大許容数 CLWLMRUC を超えない場合、除去は停止します。

17. キューの複数のリモート・インスタンスが残っている場合は、最も使用頻度の少ないチャンネルが選択されます。MQWDR.DestSeqFactor に最も小さい値があるチャンネルが、最も使用頻度の少ないチャンネルです。

- 最も小さい値を持つチャンネルが複数ある場合は、MQWDR.DestSeqNumber に最も小さい値があるチャンネルが 1 つ選択されます。
- 選択肢の宛先順序係数は、キュー・マネージャーによっておよそ 1000/CLWLWGHT ずつ増加します。

注:

- a. 使用可能な CLUSRCVR チャンネルのクラスター・ワークロード属性が変更されると、すべての宛先の宛先順序係数がゼロにリセットされます。新しい CLUSRCVR チャンネルが使用可能になると、順序係数がゼロになります。
- b. 手動で定義された CLUSSDR チャンネルのワークロード属性に対して変更を行っても、宛先順序係数はリセットされません。

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、一定になるの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 チャンネル属性では、クラスター・ワークロード分散のために、CLUSSDR チャンネルまたは 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 CLUSSDR または CLUSRCVR 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、CLUSSDR 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 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 キュー属性を使用して、宛先キューの設定を行います。WebSphere 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。

WebSphere 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・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

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 チャンネル属性では、クラスター・ワークロード分散のために、CLUSDR チャンネルまたは 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 CLUSDR または CLUSRCVR 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、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。

WebSphere 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となります。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、WebSphere 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・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。

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 チャンネル属性では、クラスター・ワークロード分散のために、CLUSDR チャンネルまたは 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 CLUSDR または CLUSRCVR 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・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・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 チャンネル属性では、クラスター・ワークロード分散のために、CLUSDR チャンネルまたは 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 CLUSDR または CLUSRCVR 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・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。

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 チャンネル属性では、クラスター・ワークロード分散のために、CLUSDR チャンネルまたは 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 CLUSDR または CLUSRCVR 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・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。

CLWLPRTY チャンネル属性

CLWLPRTY チャンネル属性では、クラスター・ワークロード分散のために、CLUSDR チャンネルまたは 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 CLUSDR または CLUSRCVR 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 チャンネル属性では、クラスター・ワークロード分散のために、CLUSDR チャンネルまたは 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 チャンネル属性は、CLUSDR チャンネルまたは CLUSRCVR チャンネルに優先権を設定する場合に使用します。IBM WebSphere 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。

WebSphere MQ では、チャンネル状況を検査した後でチャンネルの優先順位を取得します。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・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。

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 CLUSSDR または CLUSRCVR 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、CLUSSDR 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 CLUSSDR または CLUSRCVR 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。

WebSphere 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となります。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、WebSphere 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。

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 チャンネル属性では、クラスター・ワークロード分散のために、CLUSDR チャンネルまたは 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。

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。

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。

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 チャンネル属性では、クラスター・ワークロード分散のために、CLUSDR チャンネルまたは 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 CLUSDR または CLUSRCVR 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。

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。

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 チャンネル属性では、クラスター・ワークロード分散のために、CLUSDDR チャンネルまたは 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 CLUSDDR または CLUSRCVR 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、CLUSDDR 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・ワークロード出口呼び出しとデータ構造体

ここでは、クラスター・ワークロード出口と、出口で使用されるデータ構造体について説明します。ここで説明するのは、汎用プログラミング・インターフェース情報です。

クラスター・ワークロード出口は、以下のプログラミング言語で作成することができます。

- C
- System/390® アセンブラー (WebSphere MQ for z/OS)

呼び出しについては、次の節で説明します。

- [110 ページの『MQ_CLUSTER_WORKLOAD_EXIT - 呼び出しの説明』](#)

出口で使用される構造体のデータ・タイプについては、次の各節で説明します。

- [112 ページの『MQXCLWLN - クラスター・ワークロードのレコードのナビゲート』](#)
- [115 ページの『MQWXP - クラスター・ワークロード出口のパラメーター構造体』](#)
- [123 ページの『MQWDR - クラスター・ワークロード宛先レコード構造体』](#)
- [128 ページの『MQWQR - クラスター・ワークロードのキュー・レコード構造体』](#)
- [133 ページの『MQWCR - クラスター・ワークロードのクラスター・レコード構造体』](#)

このセクション全体で、キュー・マネージャー属性とキュー属性は完全に表示されています。MQSC コマンドの資料で使用される同等の名前がその下に表示されます。MQSC コマンドについて詳しくは、[MQSC リファレンス](#) を参照してください。

表 21. キュー・マネージャー属性	
完全な名前	MQSC で使用する名前
<i>ClusterWorkloadData</i>	CLWLDATA
<i>ClusterWorkloadExit</i>	CLWLEXIT
<i>ClusterWorkloadLength</i>	CLWLLEN

表 22. キューの属性	
完全な名前	MQSC で使用する名前
<i>DefBind</i>	DEFBIND
<i>DefPersistence</i>	DEFPSIST
<i>DefPriority</i>	DEFPRTY
<i>InhibitPut</i>	PUT
<i>QDesc</i>	DESCR

関連概念

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、WebSphere 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。クラスター・ワークロード出口プログラムを作成することにより、キュー・マネージャーを選択するワークロード・バランシング・アルゴリズムを提供できます。

MQ_CLUSTER_WORKLOAD_EXIT - 呼び出しの説明

このクラスター・ワークロード出口は、使用可能なキュー・マネージャーにメッセージをルーティングするためにキュー・マネージャーによって呼び出されます。

注：MQ_CLUSTER_WORKLOAD_EXIT という名前の入り口点は、キュー・マネージャーによっては提供されません。代わりに、クラスター・ワークロード出口の名前は ClusterWorkloadExit キュー・マネージャー属性によって定義されます。

MQ_CLUSTER_WORKLOAD_EXIT 出口は、すべてのプラットフォームでサポートされています。

構文

MQ_CLUSTER_WORKLOAD_EXIT (*ExitParms*)

関連資料

MQXCLWLN - クラスター・ワークロードのレコードのナビゲート

MQXCLWLN 呼び出しは、クラスター・キャッシュ内に保管された MQWDR、MQWQR、および MQWCR の各レコードのチェーン全体をナビゲートするために使用します。

MQWXP - クラスター・ワークロード出口のパラメーター構造体

以下の表には、MQWXP - クラスター・ワークロード出口のパラメーター構造体内のフィールドがまとめられています。

MQWDR - クラスター・ワークロード宛先レコード構造体

以下の表には、MQWDR - クラスター・ワークロードの宛先レコード構造体内のフィールドがまとめられています。

MQWQR - クラスター・ワークロードのキュー・レコード構造体

以下の表には、MQWQR - クラスター・ワークロードのキュー・レコード構造体内のフィールドがまとめられています。

MQWCR - クラスター・ワークロードのクラスター・レコード構造体

以下の表には、MQWCR クラスター・ワークロードのレコード構造体内のフィールドがまとめられています。

MQ_CLUSTER_WORKLOAD_EXIT のパラメーター

MQ_CLUSTER_WORKLOAD_EXIT 呼び出しのパラメーターの説明。

ExitParms (MQWXP) - 入出力

出口パラメーター・ブロック。

- 出口では、ワークロードの管理方法を示すために MQWXP 内に情報を設定します。

関連資料

使用上の注意

クラスター・ワークロード出口によって実行される関数は、出口の提供者によって定義されます。ただし、出口は、関連付けられた制御ブロックの MQWXP で定義された規則に従っていなければなりません。

MQ_CLUSTER_WORKLOAD_EXIT の言語呼び出し

MQ_CLUSTER_WORKLOAD_EXIT では、C および高水準アセンブラーの 2 つの言語がサポートされています。

使用上の注意

クラスター・ワークロード出口によって実行される関数は、出口の提供者によって定義されます。ただし、出口は、関連付けられた制御ブロックの MQWXP で定義された規則に従っていなければなりません。

MQ_CLUSTER_WORKLOAD_EXIT という名前の入り口点は、キュー・マネージャーによっては提供されません。ただし、C プログラミング言語では、MQ_CLUSTER_WORKLOAD_EXIT という名前に対して typedef が提供されています。この typedef を使用してユーザー作成の出口を宣言して、パラメーターが正しくなるようにします。

関連資料

MQ_CLUSTER_WORKLOAD_EXIT のパラメーター

MQ_CLUSTER_WORKLOAD_EXIT 呼び出しのパラメーターの説明。

MQ_CLUSTER_WORKLOAD_EXIT の言語呼び出し

MQ_CLUSTER_WORKLOAD_EXIT では、C および高水準アセンブラーの 2 つの言語がサポートされています。

MQ_CLUSTER_WORKLOAD_EXIT の言語呼び出し

MQ_CLUSTER_WORKLOAD_EXIT では、C および高水準アセンブラーの 2 つの言語がサポートされています。

C 言語での呼び出し

```
MQ_CLUSTER_WORKLOAD_EXIT (&ExitParms);
```

MQ_CLUSTER_WORKLOAD_EXIT をクラスター・ワークロード出口関数の名前でも置き換えます。

MQ_CLUSTER_WORKLOAD_EXIT パラメーターを次のように宣言します。

```
MQWXP ExitParms; /* Exit parameter block */
```

高水準アセンブラー呼び出し

```
CALL EXITNAME, (EXITPARMS)
```

パラメーターを次のように宣言します。

関連資料

[MQ_CLUSTER_WORKLOAD_EXIT](#) のパラメーター

[MQ_CLUSTER_WORKLOAD_EXIT](#) 呼び出しのパラメーターの説明。

使用上の注意

クラスター・ワークロード出口によって実行される関数は、出口の提供者によって定義されます。ただし、出口は、関連付けられた制御ブロックの MQWXP で定義された規則に従っていなければなりません。

MQXCLWLN - クラスター・ワークロードのレコードのナビゲート

MQXCLWLN 呼び出しは、クラスター・キャッシュ内に保管された MQWDR、MQWQR、および MQWCR の各レコードのチェーン全体をナビゲートするために使用します。

クラスター・キャッシュは、クラスターに関連した情報を保管するために使用される主記憶の領域です。

クラスター・キャッシュが静的である場合は、固定サイズになっています。クラスター・キャッシュを「動的」に設定すると、このキャッシュを必要に応じて拡張することができます。

クラスター・キャッシュのタイプは、システム・パラメーターまたはシステム・マクロのいずれかを使用して、STATIC または DYNAMIC に設定します。

- z/OS 以外のプラットフォームの場合、システム・パラメーターの ClusterCacheType
- z/OS の場合、CSQ6SYSP マクロ内の CLCACHE パラメーター

構文

MQXCLWLN (*ExitParms*, *CurrentRecord*, *NextOffset*, *NextRecord*, *Compcode*, *Reason*)

関連資料

[MQ_CLUSTER_WORKLOAD_EXIT](#) - 呼び出しの説明

このクラスター・ワークロード出口は、使用可能なキュー・マネージャーにメッセージをルーティングするためにキュー・マネージャーによって呼び出されます。

[MQWXP](#) - クラスター・ワークロード出口のパラメーター構造体

以下の表には、MQWXP - クラスター・ワークロード出口のパラメーター構造体内のフィールドがまとめられています。

[MQWDR](#) - クラスター・ワークロード宛先レコード構造体

以下の表には、MQWDR - クラスター・ワークロードの宛先レコード構造体内のフィールドがまとめられています。

[MQWQR](#) - クラスター・ワークロードのキュー・レコード構造体

以下の表には、MQWQR - クラスター・ワークロードのキュー・レコード構造体内のフィールドがまとめられています。

[MQWCR](#) - クラスター・ワークロードのクラスター・レコード構造体

以下の表には、MQWCR クラスター・ワークロードのレコード構造体内のフィールドがまとめられています。

MQXCLWLN - クラスター・ワークロード・レコードのナビゲート用のパラメーター

MQXCLWLN 呼び出しのパラメーターの説明。

ExitParms (MQWXP) - 入出力

出口パラメーター・ブロック。

この構造体には出口の呼び出しに関する情報があります。出口では、ワークロードの管理方法を示すために、この構造体内に情報を設定します。

CurrentRecord (MQPTR) - 入力

現行レコードのアドレス。

この構造体は、出口によって現在調べられているレコードのアドレスに関連した情報を格納します。レコードは、以下のタイプのいずれかでなければなりません。

- クラスター・ワークロードの宛先レコード (MQWDR)
- クラスター・ワークロードのキュー・レコード (MQWQR)
- クラスター・ワークロードのクラスター・レコード (MQWCR)

NextOffset (MQLONG) - 入力

次のレコードのオフセット。

この構造体は、次のレコードまたは構造体のオフセットに関連した情報を格納します。 *NextOffset* は、現行レコード内の該当するオフセット・フィールドの値であり、以下のフィールドのいずれかでなければなりません。

- MQWDR 内の ChannelDefOffset フィールド
- MQWDR 内の ClusterRecOffset フィールド
- MQWQR 内の ClusterRecOffset フィールド
- MQWCR 内の ClusterRecOffset フィールド

NextRecord (MQPTR) - 出力

次のレコードまたは構造体のアドレス。

この構造体は、次のレコードまたは構造体のアドレスに関連した情報を格納します。 *CurrentRecord* が MQWDR のアドレスであり、 *NextOffset* が ChannelDefOffset フィールドの値である場合、 *NextRecord* はチャンネル定義構造体 (MQCD) のアドレスになります。

次のレコードまたは構造体がない場合は、キュー・マネージャーは *NextRecord* をヌル・ポインターに設定し、呼び出しは完了コード MQCC_WARNING と理由コード MQRC_NO_RECORD_AVAILABLE を戻します。

CompCode (MQLONG) - 出力

完了コード

完了コードには、以下のいずれかの値が設定されます。

MQCC_OK

正常終了。

MQCC_WARNING

警告 (部分完了)。

MQCC_FAILED

呼び出し失敗。

Reason (MQLONG) - 出力

CompCode を限定する理由コード

CompCode が MQCC_OK の場合、次のようになります。

MQRC_NONE

(0, X'0000')

レポートする理由コードはありません。

CompCode が MQCC_WARNING の場合、次のようになります。

MQRC_NO_RECORD_AVAILABLE

(2359, X'0937')

使用可能なレコードはありません。MQXCLWLN 呼び出しが、チェーン内の次のレコードのアドレスを取得するためにクラスター・ワークロード出口から発行されました。現行レコードがチェーン内の最後のレコードです。修正アクション: なし。

CompCode が MQCC_FAILED の場合、次のようになります。

MQRC_CURRENT_RECORD_ERROR

(2357, X'0935')

CurrentRecord パラメーターが無効です。MQXCLWLN 呼び出しが、チェーン内の次のレコードのアドレスを取得するためにクラスター・ワークロード出口から発行されました。*CurrentRecord* パラメーターで指定されたアドレスは、有効なレコードのアドレスではありません。

CurrentRecord は、クラスター・キャッシュ内に存在する宛先レコード (MQWDR)、キュー・レコード (MQWQR)、またはクラスター・レコード (MQWCR) のアドレスでなければなりません。修正アクション: クラスター・ワークロード出口が、クラスター・キャッシュに存在する有効なレコードのアドレスを渡すようにしてください。

MQRC_ENVIRONMENT_ERROR

(2012, X'07DC')

呼び出しが環境内で無効です。MQXCLWLN 呼び出しが発行されましたが、クラスター・ワークロード出口からではありませんでした。

MQRC_NEXT_OFFSET_ERROR

(2358, X'0936')

NextOffset パラメーターが無効です。MQXCLWLN 呼び出しが、チェーン内の次のレコードのアドレスを取得するためにクラスター・ワークロード出口から発行されました。*NextOffset* パラメーターで指定されたオフセットが無効です。*NextOffset* は、以下のフィールドのいずれかの値でなければなりません。

- MQWDR 内の *ChannelDefOffset* フィールド
- MQWDR 内の *ClusterRecOffset* フィールド
- MQWQR 内の *ClusterRecOffset* フィールド
- MQWCR 内の *ClusterRecOffset* フィールド

修正アクション: *NextOffset* パラメーターに指定する値を上にもリストされたフィールドのいずれかの値にしてください。

MQRC_NEXT_RECORD_ERROR

(2361, X'0939')

NextRecord パラメーターが無効です。

MQRC_WXP_ERROR

(2356, X'0934')

ワークロード出口パラメーター構造体が無効です。MQXCLWLN 呼び出しが、チェーン内の次のレコードのアドレスを取得するためにクラスター・ワークロード出口から発行されました。以下のいずれかの理由により、ワークロード出口パラメーター構造体の *ExitParms* が無効です。

- パラメーター・ポインターが無効である。有効ではないパラメーター・ポインターを検出することは必ずしも可能ではありません。検出されない場合は、予測不能な結果が生じます。
- *StrucId* フィールドが MQWXP_STRUC_ID ではない。
- *Version* フィールドが MQWXP_VERSION_2 ではない。
- キュー・マネージャーによって出口に渡された値が *Context* フィールドにない。

修正アクション: *ExitParms* に指定するパラメーターを、出口が呼び出されたときにその出口に渡された MQWXP 構造体にしてください。

関連資料

[MQXCLWLN - クラスター・ワークロード・レコードのナビゲートの使用上の注意](#)

キャッシュが静的な場合でも、MQXCLWLN を使用してクラスター・レコード全体をナビゲートします。

[MQXCLWLN の言語呼び出し](#)

MQXCLWLN では、C および高水準アセンブラの 2 つの言語がサポートされています。

MQXCLWLN - クラスター・ワークロード・レコードのナビゲートの使用上の注意

キャッシュが静的な場合でも、MQXCLWLN を使用してクラスター・レコード全体をナビゲートします。

クラスター・キャッシュが動的な場合、MQXCLWLN 呼び出しを使用してレコード全体をナビゲートする必要があります。レコード全体をナビゲートするために単純なポインターとオフセットの演算を使用すると、出口が異常終了します。

クラスター・キャッシュが静的な場合、レコード全体をナビゲートするために MQXCLWLN 呼び出しを使用する必要はありません。キャッシュが静的な場合も、通常 MQXCLWLN を使用してください。その後、ワークロード出口の変更を必要とせずに、クラスター・キャッシュを「動的」に変更することができます。

関連資料

[MQXCLWLN - クラスター・ワークロード・レコードのナビゲート用のパラメーター](#)
MQXCLWLN 呼び出しのパラメーターの説明。

MQXCLWLN の言語呼び出し

MQXCLWLN では、C および高水準アセンブラの 2 つの言語がサポートされています。

MQXCLWLN の言語呼び出し

MQXCLWLN では、C および高水準アセンブラの 2 つの言語がサポートされています。

C 言語での呼び出し

```
MQXCLWLN (&ExitParms, CurrentRecord, NextOffset, &NextRecord, &CompCode, &Reason);
```

パラメーターを次のように宣言します。

```
typedef struct tagMQXCLWLN {
    MQWXP      ExitParms;          /* Exit parameter block */
    MQPTR      CurrentRecord;     /* Address of current record*/
    MQLONG     NextOffset;        /* Offset of next record */
    MQPTR      NextRecord;        /* Address of next record or structure */
    MQLONG     CompCode;          /* Completion code */
    MQLONG     Reason;            /* Reason code qualifying CompCode */
}
```

高水準アセンブラ呼び出し

```
CALL MQXCLWLN, (CLWLEXITPARMS, CURRENTRECORD, NEXTOFFSET, NEXTRECORD, COMPCODE, REASON)
```

パラメーターを次のように宣言します。

CLWLEXITPARMS	CMQWXP,	Cluster workload exit parameter block
CURRENTRECORD	CMQWDRA,	Current record
NEXTOFFSET	DS F	Next offset
NEXTRECORD	DS F	Next record
COMPCODE	DS F	Completion code
REASON	DS F	Reason code qualifying COMPCODE

関連資料

[MQXCLWLN - クラスター・ワークロード・レコードのナビゲート用のパラメーター](#)
MQXCLWLN 呼び出しのパラメーターの説明。

MQXCLWLN - クラスター・ワークロード・レコードのナビゲートの使用上の注意

キャッシュが静的な場合でも、MQXCLWLN を使用してクラスター・レコード全体をナビゲートします。

MQWXP - クラスター・ワークロード出口のパラメーター構造体

以下の表には、MQWXP - クラスター・ワークロード出口のパラメーター構造体内のフィールドがまとめられています。

表 23. MQWXP のフィールド		
フィールド	説明	参照ページ
StrucId	構造体 ID	StrucId

表 23. MQWXP のフィールド (続き)

フィールド	説明	参照ページ
<i>Version</i>	構造体のバージョン番号	Version
<i>ExitId</i>	出口のタイプ	ExitId
<i>ExitReason</i>	出口を呼び出す理由	ExitReason
<i>ExitResponse</i>	出口からの応答	ExitResponse
<i>ExitResponse2</i>	出口からの 2 次応答	ExitResponse2
<i>Feedback</i>	フィードバック・コード	Feedback
<i>Flags</i>	フラグ値。これらのビット・フラグは、書き込まれるメッセージの情報を示すために使用されます	Flags
<i>ExitUserArea</i>	出口ユーザー域	ExitUserArea
<i>ExitData</i>	出口データ	ExitData
<i>MsgDescPtr</i>	メッセージ記述子 (MQMD) のアドレス	MsgDescPtr
<i>MsgBufferPtr</i>	メッセージ・データの一部または全体が入っているバッファのアドレス	MsgBufferPtr
<i>MsgBufferLength</i>	メッセージ・データが入っているバッファの長さ	MsgBufferLength
<i>MsgLength</i>	完全なメッセージの長さ	MsgLength
<i>QName</i>	キューの名前	QName
<i>QMGrName</i>	ローカル・キュー・マネージャーの名前	QMGrName
<i>DestinationCount</i>	有効な宛先の数	DestinationCount
<i>DestinationChosen</i>	選択された宛先	DestinationChosen
<i>DestinationArrayPtr</i>	宛先レコード (MQWDR) を指すポインタの配列のアドレス	DestinationArrayPtr
<i>QArrayPtr</i>	キュー・レコード (MQWQR) を指すポインタの配列のアドレス	QArrayPtr
注: <i>Version</i> が MQWXP_VERSION_2 より前のものである場合、以下のフィールドは無視されます。		
<i>CacheContext</i>	コンテキスト情報	CacheContext
<i>CacheType</i>	クラスター・キャッシュのタイプ	CacheType
注: <i>Version</i> が MQWXP_VERSION_3 より前のものである場合、以下のフィールドは無視されます。		
<i>CLWLMRUChannels</i>	許可されたアクティブ・アウトバウンド・クラスター・チャンネルの最大数	CLWLMRUChannels
注: <i>Version</i> が MQWXP_VERSION_4 より前のものである場合、以下のフィールドは無視されます。		
<i>pEntryPoints</i>	MQI と DCI の呼び出しを可能にする MQIEP 構造体のアドレス	pEntryPoints

クラスター・ワークロード出口のパラメーター構造体によって、クラスター・ワークロード出口に渡される情報が記述されます。

クラスター・ワークロード出口のパラメーター構造体は、すべてのプラットフォーム上でサポートされます。

また、後方互換性のため、MQWXP1、MQWXP2、および MQWXP3 構造体も用意されています。

関連資料

MQ_CLUSTER_WORKLOAD_EXIT - 呼び出しの説明

このクラスター・ワークロード出口は、使用可能なキュー・マネージャーにメッセージをルーティングするためにキュー・マネージャーによって呼び出されます。

MQXCLWLN - クラスター・ワークロードのレコードのナビゲート

MQXCLWLN 呼び出しは、クラスター・キャッシュ内に保管された MQWDR、MQWQR、および MQWCR の各レコードのチェーン全体をナビゲートするために使用します。

MQWDR - クラスター・ワークロード宛先レコード構造体

以下の表には、MQWDR - クラスター・ワークロードの宛先レコード構造体内のフィールドがまとめられています。

MQWQR - クラスター・ワークロードのキュー・レコード構造体

以下の表には、MQWQR - クラスター・ワークロードのキュー・レコード構造体内のフィールドがまとめられています。

MQWCR - クラスター・ワークロードのクラスター・レコード構造体

以下の表には、MQWCR クラスター・ワークロードのレコード構造体内のフィールドがまとめられています。

MQWXP - クラスター・ワークロード出口のパラメーター構造体内のフィールド

MQWXP - クラスター・ワークロード出口のパラメーター構造体内のフィールドの説明

StrucId (MQCHAR4) - 入力

クラスター・ワークロード出口のパラメーター構造体の構造体 ID。

- StrucId の値は MQWXP_STRUC_ID です。
- C プログラミング言語用に、定数の MQWXP_STRUC_ID_ARRAY も定義されています。この定数には、MQWXP_STRUC_ID と同じ値が含まれています。これは、ストリングではなく文字の配列です。

Version (MQLONG) - 入力

構造体のバージョン番号を示します。Version では、次のいずれかの値を取ります。

MQWXP_VERSION_1

クラスター・ワークロード出口のパラメーター構造体のバージョン 1。

MQWXP_VERSION_1 は、すべての環境でサポートされます。

MQWXP_VERSION_2

クラスター・ワークロード出口のパラメーター構造体のバージョン 2。

MQWXP_VERSION_2 は、AIX、HP-UX、Linux、IBM i、Solaris、および Windows の各環境でサポートされます。

MQWXP_VERSION_3

クラスター・ワークロード出口のパラメーター構造体のバージョン 3。

MQWXP_VERSION_3 は、AIX、HP-UX、Linux、IBM i、Solaris、および Windows の各環境でサポートされます。

MQWXP_VERSION_4

クラスター・ワークロード出口のパラメーター構造体のバージョン 4。

MQWXP_VERSION_4 は、AIX、HP-UX、Linux、IBM i、Solaris、および Windows の各環境でサポートされます。

MQWXP_CURRENT_VERSION

クラスター・ワークロード出口パラメーター構造体の現行バージョン。

ExitId (MQLONG) - 入力

呼び出す出口のタイプを示します。クラスター・ワークロード出口が唯一サポートされる出口です。

- ExitId の値は MQXT_CLUSTER_WORKLOAD_EXIT でなければなりません

ExitReason (MQLONG) - 入力

クラスター・ワークロード出口を呼び出す理由を示します。ExitReason では、次のいずれかの値を取ります。

MQXR_INIT

出口が初めて呼び出されることを示します。

出口で必要になる可能性がある主記憶などのリソースをすべて獲得して初期化します。

MQXR_TERM

出口が終了されようとしていることを示します。

出口が初期化されてから獲得している可能性がある主記憶などのリソースを、すべて解放します。

MQXR_CLWL_OPEN

MQOPEN によって呼び出されます。

MQXR_CLWL_PUT

MQPUT または MQPUT1 によって呼び出されます。

MQXR_CLWL_MOVE

チャンネル状態が変化した場合に MCA によって呼び出されます。

MQXR_CLWL_REPOS

リポジトリ・マネージャーの PCF メッセージ用の MQPUT または MQPUT1 によって呼び出されま
す。

MQXR_CLWL_REPOS_MOVE

チャンネル状態が変化した場合に、リポジトリ・マネージャーの PCF メッセージ用の MCA によ
って呼び出されます。

ExitResponse (MQLONG) - 出力

メッセージの処理を続行するかどうかを示すために ExitResponse を設定します。値は、次のいずれ
かでなければなりません。

MQXCC_OK

メッセージの処理を正常に続行させます。

- DestinationChosen では、メッセージを送信する宛先を特定します。

MQXCC_SUPPRESS_FUNCTION

メッセージの処理を中止します。

- キュー・マネージャーによって実行される処置は、以下に示すように出口が呼び出された理由に
よって決まります。

表 24. キュー・マネージャーによって実行される処置	
ExitReason	実行される処置
<ul style="list-style-type: none"> - MQXR_CLWL_OPEN - MQXR_CLWL_REPOS - MQXR_CLWL_PUT 	MQOPEN 呼び出し、MQPUT 呼び出し、または MQPUT1 呼び出しは、完了コード MQCC_FAILED および理由コード MQRC_STOPPED_BY_CLUSTER_EXIT で失敗します。
<ul style="list-style-type: none"> - MQXR_CLWL_MOVE - MQXR_CLWL_REPOS_MOVE 	メッセージが送達不能キューに配置されます。

MQXCC_SUPPRESS_EXIT

現在のメッセージの処理を正常に続行させます。キュー・マネージャーがシャットダウンするま
で、出口を再度呼び出さないでください。

キュー・マネージャーでは、ClusterWorkloadExit キュー・マネージャー属性がブランクであ
るかのように後続のメッセージを処理します。DestinationChosen では、現在のメッセージを
送信する宛先を特定します。

他のすべての値

MQXCC_SUPPRESS_FUNCTION が指定されているかのようにメッセージが処理されます。

ExitResponse2 (MQLONG) - 入出力

キュー・マネージャーに詳しい情報を提供するには、ExitResponse2 を設定します。

- MQXR2_STATIC_CACHE がデフォルト値で、出口ルーチンへの入り口で設定されます。
- ExitReason に値 MQXR_INIT が設定されている場合、出口では ExitResponse2 に以下の値のいずれかを設定できます。

MQXR2_STATIC_CACHE

出口には、静的クラスター・キャッシュが必要です。

- クラスター・キャッシュが静的な場合、出口では MQXCLWLN 呼び出しを使用してクラスター・キャッシュ内にあるレコードのチェーンをナビゲートする必要はありません。
- クラスター・キャッシュが動的な場合、出口ではキャッシュ内のレコードを正しくナビゲートできません。

注: キュー・マネージャーでは、出口が ExitResponse フィールドに MQXCC_SUPPRESS_EXIT を返したかのように MQXR_INIT 呼び出しからの戻り値を処理します。

MQXR2_DYNAMIC_CACHE

出口は、静的キャッシュまたは動的キャッシュのいずれかと作動可能です。

- 出口がこの値を返した場合、出口では MQXCLWLN 呼び出しを使用してクラスター・キャッシュ内にあるレコードのチェーンをナビゲートしなければなりません。

Feedback (MQLONG) - 入力

予約フィールド。値はゼロです。

Flags (MQLONG) - 入力

出力されるメッセージについての情報を示します。

- Flags の値は MQWXP_PUT_BY_CLUSTER_CHL です。メッセージは、クラスター・チャンネル以外からやローカルではなく、クラスター・チャンネルから発信されています。つまり、メッセージは別のクラスター・キュー・マネージャーから届いています。

Reserved (MQLONG) - 入力

予約フィールド。値はゼロです。

ExitUserArea (MQBYTE16) - 入出力

出口の呼び出し間で通信するには、ExitUserArea を設定します。

- ExitUserArea は、出口の最初の呼び出しの前に 2 進ゼロに初期化されます。このフィールドに対して出口によって加えられる変更は、MQCONN 呼び出しと対応する MQDISC 呼び出しとの間で発生する出口の呼び出しごとにすべて保存されます。このフィールドは、MQDISC 呼び出しが発生すると 2 進ゼロにリセットされます。
- 出口の最初の呼び出しは、値 MQXR_INIT を持つ ExitReason フィールドによって示されます。
- 以下の定数が定義されています。

MQXUA_NONE - ストリング

MQXUA_NONE_ARRAY - 文字配列

ユーザー情報なし。これらの定数は両方とも、フィールドの長さ分の 2 進ゼロです。

MQ_EXIT_USER_AREA_LENGTH

ExitUserArea の長さ。

ExitData (MQCHAR32) - 入力

ClusterWorkloadData キュー・マネージャー属性の値。この属性に値が定義されていない場合、このフィールドはすべてブランクになります。

- ExitData の長さは、MQ_EXIT_DATA_LENGTH により指定されます。

MsgDescPtr (PMQMD) - 入力

処理されるメッセージのメッセージ記述子 (MQMD) のコピーのアドレス。

- 出口によってメッセージ記述子に加えられた変更は、キュー・マネージャーですべて無視されます。
- ExitReason に以下の値のいずれかが設定されている場合、MsgDescPtr は NULL ポインターに設定され、メッセージ記述子は出口に渡されません。
 - MQXR_INIT
 - MQXR_TERM
 - MQXR_CLWL_OPEN

MsgBufferPtr (PMQVOID) - 入力

メッセージ・データの最初の MsgBufferLength バイトのコピーが含まれたバッファのアドレス。

- 出口によってメッセージ・データに加えられた変更は、キュー・マネージャーですべて無視されます。
- 以下の場合、メッセージ・データは出口に渡されません。
 - MsgDescPtr が NULL ポインターである。
 - メッセージにデータがない。
 - ClusterWorkloadLength キュー・マネージャー属性がゼロである。これらの場合、MsgBufferPtr は NULL ポインターです。

MsgBufferLength (MQLONG) - 入力

出口に渡されるメッセージ・データが含まれたバッファの長さ。

- この長さは、ClusterWorkloadLength キュー・マネージャー属性によって制御されます。
- この長さは、完全なメッセージの長さよりも短い場合があります (MsgLength を参照)。

MsgLength (MQLONG) - 入力

出口に渡される完全なメッセージの長さ。

- MsgBufferLength は、完全なメッセージの長さよりも短い場合があります。
- ExitReason が MQXR_INIT、MQXR_TERM、または MQXR_CLWL_OPEN の場合、MsgLength はゼロです。

QName (MQCHAR48) - 入力

宛先キューの名前。このキューはクラスター・キューです。

- QName の長さは MQ_Q_NAME_LENGTH です。

QMgrName (MQCHAR48) - 入力

クラスター・ワークロード出口を呼び出しているローカル・キュー・マネージャーの名前。

- QMgrName の長さは MQ_Q_MGR_NAME_LENGTH です。

DestinationCount (MQLONG) - 入力

可能な宛先の数。宛先は、宛先キューのインスタンスで、宛先レコードによって記述されます。

- 宛先レコードは MQWDR 構造体です。キューのインスタンスへの有効な経路ごとに、1つの構造体があります。
- MQWDR 構造体は、ポインターの配列によってアドレス指定されます (DestinationArrayPtr を参照)。

DestinationChosen (MQLONG) - 入出力

選択された宛先。

- メッセージを送信する経路とキュー・インスタンスを特定する MQWDR 構造体の番号。
- この値の範囲は、1 から DestinationCount までです。

- DestinationChosen は、出口への入力時にキュー・マネージャーで選択している経路とキュー・インスタンスを示します。出口では、この選択を受け入れることができ、また別の経路とキューのインスタンスを選択することもできます。
- 出口によって設定される値は、1 から DestinationCount までの範囲になければなりません。他の値が返された場合、キュー・マネージャーでは出口への入力時に DestinationChosen の値を使用します。

DestinationArrayPtr (PPMQWDR) - 入力

宛先レコード (MQWDR) を指すポインタの配列のアドレス。

- DestinationCount 個の宛先レコードが存在します。

QArrayPtr (PPMQWQR) - 入力

キュー・レコード (MQWQR) を指すポインタの配列のアドレス。

- キュー・レコードが使用できる場合、それらは DestinationCount 個存在します。
- キュー・レコードが使用できない場合、QArrayPtr は NULL ポインタです。

注: DestinationCount がゼロより大きい場合でも、QArrayPtr が NULL ポインタの場合があります。

CacheContext (MQPTR) : バージョン 2 - 入力

CacheContext フィールドは、キュー・マネージャーによる使用のために予約されています。出口では、このフィールドの値を変更してはなりません。

CacheType (MQLONG) : バージョン 2 - 入力

クラスター・キャッシュは、次のいずれかのタイプになります。

MQCLCT_STATIC

キャッシュは静的です。

- キャッシュのサイズは固定されており、キュー・マネージャーの作動に伴って増やすことはできません。
- このタイプのキャッシュ内のレコードをナビゲートするために、MQXCLWLN 呼び出しを使用する必要はありません。

MQCLCT_DYNAMIC

キャッシュは動的です。

- キャッシュのサイズは、変化するクラスター情報を収容するために増やすことができます。
- このタイプのキャッシュ内のレコードをナビゲートするには、MQXCLWLN 呼び出しを使用する必要があります。

CLWLMRUChannels (MQLONG) : バージョン 3 - 入力

クラスター・ワークロード選択アルゴリズムによって使用されると予想される、アクティブ・アウトバウンド・クラスター・チャンネルの最大数を示します。

- CLWLMRUChannels の値は 1 から 999 999 999 までです。

pEntryPoints (PMQIEP) : バージョン 4

MQIEP 構造体のアドレス。これによって、MQI および DCI 呼び出しを実行できます。

MQWXP の初期値および言語ごとの宣言

MQWXP - クラスター・ワークロード出口のパラメーター構造体用の初期値および C と高水準アセンブラー言語の宣言。

フィールド名	定数の名前	定数の値
StrucId	MQWXP_STRUC_ID	'WXP? '
Version	MQWXP_VERSION_2	2

表 25. MQWXP のフィールドの初期値 (続き)

フィールド名	定数の名前	定数の値
<i>ExitId</i>	なし	0
<i>ExitReason</i>	MQXCC_OK	0
<i>ExitResponse</i>	なし	0
<i>ExitResponse2</i>	なし	0
<i>Flags</i>	なし	0
<i>ExitUserArea</i>	{MQXUA_NONE_ARRAY}	0
<i>ExitData</i>	なし	""
<i>MsgDescPtr</i>	なし	NULL
<i>MsgBufferPtr</i>	なし	NULL
<i>MsgBufferLength</i>	なし	0
<i>MsgBufferPtr</i>	なし	0
<i>QName</i>	なし	""
<i>QMgrName</i>	なし	""
<i>DestinationCount</i>	なし	0
<i>DestinationChosen</i>	なし	0
<i>DestinationArrayPtr</i>	なし	NULL
<i>QArrayPtr</i>	なし	NULL
<i>CacheContext</i>	なし	NULL
<i>CacheType</i>	MQCLCT_DYNAMIC	1
<i>CLWLMRUChannels</i>	なし	0
<i>pEntryPoints</i>	なし	NULL

注:

1. 記号? は、単一のブランク文字を表します。
2. C プログラミング言語では、マクロ変数 MQWXP_DEFAULT にデフォルト値が設定されています。このマクロ変数を以下の方法で使用して、構造体のフィールドに初期値を設定します。

```
MQWDR MyWXP = {MQWXP_DEFAULT};
```

C 宣言

```
typedef struct tagMQWXP {
    MQCHAR4   StructId;           /* Structure identifier */
    MQLONG    Version;           /* Structure version number */
    MQLONG    ExitId;            /* Type of exit */
    MQLONG    ExitReason;        /* Reason for invoking exit */
    MQLONG    ExitResponse;      /* Response from exit */
    MQLONG    ExitResponse2;     /* Reserved */
    MQLONG    Feedback;          /* Reserved */
    MQLONG    Flags;             /* Flags */
    MQBYTE16  ExitUserArea;      /* Exit user area */
    MQCHAR32  ExitData;          /* Exit data */
    PMQMD     MsgDescPtr;        /* Address of message descriptor */
};
```

```

PMQVOID    MsgBufferPtr;        /* Address of buffer containing some
                                or all of the message data */
MQLONG     MsgBufferLength;     /* Length of buffer containing message
                                data */
MQLONG     MsgLength;          /* Length of complete message */
MQCHAR48   QName;              /* Queue name */
MQCHAR48   QMgrName;           /* Name of local queue manager */
MQLONG     DestinationCount;    /* Number of possible destinations */
MQLONG     DestinationChosen;   /* Destination chosen */
PPMQWDR    DestinationArrayPtr; /* Address of an array of pointers to
                                destination records */
PPMQWQR    QArrayPtr;          /* Address of an array of pointers to
                                queue records */

/* version 1 */
MQPTR      CacheContext;       /* Context information */
MQLONG     CacheType;          /* Type of cluster cache */
/* version 2 */
MQLONG     CLWLMRUChannels;    /* Maximum number of most recently
                                used cluster channels */

/* version 3 */
PMQIEP     pEntryPoints;       /* Address of the MQIEP structure */
/* version 4 */
};

```

High Level Assembler

```

MQWXP                                DSECT
MQWXP_STRUCID                        DS    CL4      Structure identifier
MQWXP_VERSION                        DS    F        Structure version number
MQWXP_EXITID                         DS    F        Type of exit
MQWXP_EXITREASON                     DS    F        Reason for invoking exit
MQWXP_EXITRESPONSE                   DS    F        Response from exit
MQWXP_EXITRESPONSE2                 DS    F        Reserved
MQWXP_FEEDBACK                       DS    F        Reserved
MQWXP_RESERVED                       DS    F        Reserved
MQWXP_EXITUSERAREA                   DS    XL16     Exit user area
MQWXP_EXITDATA                       DS    CL32     Exit data
MQWXP_MSGDESCPTR                     DS    F        Address of message
*                                     descriptor
MQWXP_MSGBUFFERPTR                   DS    F        Address of buffer containing
*                                     some or all of the message
*                                     data
MQWXP_MSGBUFFERLENGTH                 DS    F        Length of buffer containing
*                                     message data
MQWXP_MSGLENGTH                       DS    F        Length of complete message
MQWXP_QNAME                           DS    CL48     Queue name
MQWXP_QMGRNAME                        DS    CL48     Name of local queue manager
MQWXP_DESTINATIONCOUNT               DS    F        Number of possible
*                                     destinations
MQWXP_DESTINATIONCHOSEN               DS    F        Destination chosen
MQWXP_DESTINATIONARRAYPTR            DS    F        Address of an array of
*                                     pointers to destination
*                                     records
MQWXP_QARRAYPTR                       DS    F        Address of an array of
*                                     pointers to queue records
MQWXP_CACHECONTEXT                   DS    F        Context information
MQWXP_CACHETYPE                       DS    F        Type of cluster cache
MQWXP_CLWLMRUCHANNELS                 DS    F        Number of most recently used
*                                     channels for workload balancing

MQWXP_LENGTH                          EQU    *-MQWXP  Length of structure
MQWXP_AREA                             ORG    MQWXP
MQWXP_AREA                             DS    CL(MQWXP_LENGTH)

```

MQWDR - クラスター・ワークロード宛先レコード構造体

以下の表には、MQWDR - クラスター・ワークロードの宛先レコード構造体内のフィールドがまとめられています。

表 26. MQWDR のフィールド		
フィールド	説明	参照ページ
StrucId	構造体 ID	StrucId

表 26. MQWDR のフィールド (続き)

フィールド	説明	参照ページ
<i>Version</i>	構造体のバージョン番号	Version
<i>StrucLength</i>	MQWDR 構造体の長さ	StrucLength
<i>QMgrFlags</i>	キュー・マネージャー・フラグ	QMgrFlags
<i>QMgrIdentifier</i>	キュー・マネージャーの ID	QMgrIdentifier
<i>QMgrName</i>	キュー・マネージャー名	QMgrName
<i>ClusterRecOffset</i>	最初のクラスター・レコード (MQWCR) の論理オフセット	ClusterRecOffset
<i>ChannelState</i>	チャンネルの状態	ChannelState
<i>ChannelDefOffset</i>	チャンネル定義構造体 (MQCD) の論理オフセット	ChannelDefOffset
注: <i>Version</i> が MQWDR_VERSION_2 より前のものである場合、以下のフィールドは無視されます。		
<i>DestSeqNumber</i>	チャンネル宛先順序番号	DestSeqNumber
<i>DestSeqFactor</i>	加重用のチャンネル宛先順序係数	DestSeqFactor

クラスター・ワークロードの宛先レコード構造体には、メッセージの有効な宛先の 1 つに関連する情報が含まれています。宛先キューのインスタンスごとに、1 つのクラスター・ワークロードの宛先レコード構造体があります。

クラスター・ワークロードの宛先レコード構造体は、すべての環境でサポートされます。

また、後方互換性のために、MQWDR1 構造体および MQWDR2 構造体も用意されています。

関連資料

[MQ_CLUSTER_WORKLOAD_EXIT - 呼び出しの説明](#)

このクラスター・ワークロード出口は、使用可能なキュー・マネージャーにメッセージをルーティングするためにキュー・マネージャーによって呼び出されます。

[MQXCLWLN - クラスター・ワークロードのレコードのナビゲート](#)

MQXCLWLN 呼び出しは、クラスター・キャッシュ内に保管された MQWDR、MQWQR、および MQWCR の各レコードのチェーン全体をナビゲートするために使用します。

[MQWXP - クラスター・ワークロード出口のパラメーター構造体](#)

以下の表には、MQWXP - クラスター・ワークロード出口のパラメーター構造体内のフィールドがまとめられています。

[MQWQR - クラスター・ワークロードのキュー・レコード構造体](#)

以下の表には、MQWQR - クラスター・ワークロードのキュー・レコード構造体内のフィールドがまとめられています。

[MQWCR - クラスター・ワークロードのクラスター・レコード構造体](#)

以下の表には、MQWCR クラスター・ワークロードのレコード構造体内のフィールドがまとめられています。

MQWDR - クラスター・ワークロードの宛先レコード構造体内のフィールド

MQWDR - クラスター・ワークロードの宛先レコード構造体内のパラメーターの説明。

StrucId (MQCHAR4) - 入力

クラスター・ワークロードの宛先レコード構造体の構造体 ID。

- StrucId の値は MQWDR_STRUC_ID です。
- C プログラミング言語用に、定数の MQWDR_STRUC_ID_ARRAY も定義されています。この定数には、MQWDR_STRUC_ID と同じ値が含まれています。これは、ストリングではなく文字の配列です。

Version (MQLONG) - 入力

構造体のバージョン番号。Version では、次のいずれかの値を取ります。

MQWDR_VERSION_1

クラスター・ワークロードの宛先レコードのバージョン 1。

MQWDR_VERSION_2

クラスター・ワークロードの宛先レコードのバージョン 2。

MQWDR_CURRENT_VERSION

クラスター・ワークロード宛先レコードの Current® バージョン。

StrucLength (MQLONG) - 入力

MQWDR 構造体の長さ。StrucLength では、次のいずれかの値を取ります。

MQWDR_LENGTH_1

クラスター・ワークロードの宛先レコードのバージョン 1 の長さ。

MQWDR_LENGTH_2

クラスター・ワークロードの宛先レコードのバージョン 2 の長さ。

MQWDR_CURRENT_LENGTH

クラスター・ワークロードの宛先レコードの現行バージョンの長さ。

QMgrFlags (MQLONG) - 入力

MQWDR 構造体によって記述された宛先キューのインスタンスをホストするキュー・マネージャーのプロパティを示すキュー・マネージャー・フラグ。以下のフラグが定義されます。

MQQMF_REPOSITORY_Q_MGR

宛先は、フル・リポジトリのキュー・マネージャーです。

MQQMF_CLUSSDR_USER_DEFINED

クラスター送信側チャンネルが手動で定義されました。

MQQMF_CLUSSDR_AUTO_DEFINED

クラスター送信側チャンネルが自動で定義されました。

MQQMF_AVAILABLE

宛先キュー・マネージャーが、メッセージを受信できます。

その他の値

このフィールドには、内部目的のためにキュー・マネージャーによって他のフラグが設定される場合があります。

QMgrIdentifier (MQCHAR48) - 入力

キュー・マネージャー ID とは、MQWDR 構造体によって記述された宛先キューのインスタンスをホストするキュー・マネージャーの固有 ID です。

- この ID は、キュー・マネージャーによって生成されます。
- QMgrIdentifier の長さは MQ_Q_MGR_IDENTIFIER_LENGTH です。

QMgrName (MQCHAR48) - 入力

MQWDR 構造体によって記述された宛先キューのインスタンスをホストするキュー・マネージャーの名前。

- QMgrName は、ローカル・キュー・マネージャーおよびクラスター内の別のキュー・マネージャーの名前にすることができます。
- QMgrName の長さは MQ_Q_MGR_NAME_LENGTH です。

ClusterRecOffset (MQLONG) - 入力

MQWDR 構造体に属している最初の MQWCR 構造体の論理オフセット。

- 静的キャッシュの場合、ClusterRecOffset は、MQWDR 構造体に属している最初の MQWCR 構造体のオフセットです。
- オフセット (相対位置) は、MQWDR 構造体の先頭からバイト単位で測定されます。

- ・ 論理オフセットは、動的キャッシュのポインター演算に使用しないでください。次のレコードのアドレスを取得するには、MQXCLWLN 呼び出しを使用する必要があります。

ChannelState (MQLONG) - 入力

MQWDR 構造体で特定されたキュー・マネージャーにローカル・キュー・マネージャーをリンクするチャンネルの状態。属性の値は以下のとおりです。

MQCHS_BINDING

チャンネルはパートナーと折衝中です。

MQCHS_INACTIVE

チャンネルはアクティブではありません。

MQCHS_INITIALIZING

チャンネルは初期化中です。

MQCHS_PAUSED

チャンネルは一時停止しています。

MQCHS_REQUESTING

要求側チャンネルが接続を要求しています。

MQCHS_RETRYING

チャンネルは接続の確立を再試行しています。

MQCHS_RUNNING

チャンネルはメッセージの転送中またはメッセージ待ちの状態です。

MQCHS_STARTING

チャンネルはアクティブになるのを待っています。

MQCHS_STOPPING

チャンネルは停止中です。

MQCHS_STOPPED

チャンネルは停止しました。

ChannelDefOffset (MQLONG) - 入力

MQWDR 構造体で特定されたキュー・マネージャーにローカル・キュー・マネージャーをリンクするチャンネルのチャンネル定義 (MQCD) の論理オフセット。

- ・ ChannelDefOffset は ClusterRecOffset に類似しています
- ・ 論理オフセットはポインター演算に使用することはできません。次のレコードのアドレスを取得するには、MQXCLWLN 呼び出しを使用する必要があります。

DestSeqFactor (MQLONG) - 入力

重みづけに基づいたチャンネルの選択を可能にする宛先順序係数。

- ・ DestSeqFactor が使用されると、キュー・マネージャーはそれを変更します。
- ・ ワークロード・マネージャーでは、メッセージがチャンネルの重みづけに従って各チャンネルに分散されるようにする方法で DestSeqFactor を増やします。

DestSeqNumber (MQLONG) - 入力

キュー・マネージャーが変更する前のクラスター・チャンネル宛先の値。

- ・ メッセージがこのチャンネルに書き込まれるたびに、ワークロード・マネージャーでは DestSeqNumber を増やします。
- ・ ワークロード出口では、メッセージを書き込むチャンネルを決定するために DestSeqNumber を使用できます。

MQWDR の初期値および言語ごとの宣言

MQWDR - クラスター・ワークロードの宛先レコード用の初期値および C と高水準アセンブラー言語の宣言。

表 27. MQWDR のフィールドの初期値

フィールド名	定数の名前	定数の値
<i>StrucId</i>	MQWDR_STRUC_ID	'WDR? '
<i>Version</i>	MQWDR_VERSION_1	1
<i>StrucLength</i>	MQWDR_CURRENT_LENGTH ³	136
<i>QMgrFlags</i>	MQWDR_NONE	0
<i>QMgrIdentifier</i>	なし	""
<i>QMgrName</i>	なし	""
<i>ClusterRecOffset</i>	なし	0
<i>ChannelState</i>	なし	0
<i>ChannelDefOffset</i>	なし	0
<i>DestSeqNumber</i>	なし	0
<i>DestSeqFactor</i>	なし	0

注:

- 記号? は、単一のブランク文字を表します。
- C プログラミング言語では、マクロ変数 MQWDR_DEFAULT にデフォルト値が設定されています。このマクロ変数を以下の方法で使用して、構造体のフィールドに初期値を設定します。

```
MQWDR MyWDR = {MQWDR_DEFAULT};
```

- 初期値では、構造体の長さを構造体のバージョン 1 ではなく、現行バージョンの長さに意図的に設定します。

High Level Assembler

```
MQWDR          DSECT
MQWDR_STRUCID  DS   CL4      Structure identifier
MQWDR_VERSION  DS   F        Structure version number
MQWDR_STRUCLNGTH DS   F        Length of MQWDR structure
MQWDR_QMGRFLAGS DS   F        Queue-manager flags
MQWDR_QMGRIDENTIFIER DS CL48   Queue-manager identifier
MQWDR_QMGRNAME  DS   CL48   Queue-manager name
MQWDR_CLUSTERRECOFFSET DS   F        Offset of first cluster
*              record
MQWDR_CHANNELSTATE DS   F        Channel state
MQWDR_CHANNELDEFOFFSET DS   F        Offset of channel definition
*              structure
MQWDR_LENGTH    EQU   *-MQWDR Length of structure
MQWDR_AREA      ORG   MQWDR
                DS   CL(MQWDR_LENGTH)
```

C 宣言

```
typedef struct tagMQWDR {
    MQCHAR4  StrucId;          /* Structure identifier */
    MQLONG   Version;         /* Structure version number */
    MQLONG   StrucLength;     /* Length of MQWDR structure */
    MQLONG   QMgrFlags;       /* Queue-manager flags */
    MQCHAR48 QMgrIdentifier;   /* Queue-manager identifier */
    MQCHAR48 QMgrName;        /* Queue-manager name */
    MQLONG   ClusterRecOffset; /* Offset of first cluster record */
    MQLONG   ChannelState;    /* Channel state */
    MQLONG   ChannelDefOffset; /* Offset of channel definition structure */
    /* Ver:1 */
    MQLONG   DestSeqNumber;   /* Cluster channel destination sequence number */
};
```

```

MQINT64 DestSeqFactor; /* Cluster channel factor sequence number */
/* Ver:2 */
};

```

MQWQR - クラスター・ワークロードのキュー・レコード構造体

以下の表には、MQWQR - クラスター・ワークロードのキュー・レコード構造体内のフィールドがまとめられています。

表 28. MQWQR のフィールド		
フィールド	説明	参照ページ
<i>StrucId</i>	構造体 ID	StrucId
<i>Version</i>	構造体のバージョン番号	Version
<i>StrucLength</i>	MQWQR 構造体の長さ	StrucLength
<i>QFlags</i>	キュー・フラグ	QFlags
<i>QName</i>	キュー名	QName
<i>QMgrIdentifier</i>	キュー・マネージャーの ID	QMgrIdentifier
<i>ClusterRecOffset</i>	最初のクラスター・レコード (MQWCR) のオフセット	ClusterRecOffset
<i>QType</i>	キュー・タイプ	QType
<i>QDesc</i>	キューの記述	QDesc
<i>DefBind</i>	デフォルトのバインド	DefBind
<i>DefPersistence</i>	デフォルトのメッセージ持続性	DefPersistence
<i>DefPriority</i>	デフォルトのメッセージ優先順位	DefPriority
<i>InhibitPut</i>	キューへの PUT 操作が許可されるかどうか	InhibitPut
注: Version が MQWQR_VERSION_2 より前のものである場合、以下のフィールドは無視されます。		
<i>CWLQueuePriority</i>	キューの優先順位を表す 0 から 9 までの値	CWLQueuePriority
<i>CLWLQueueRank</i>	キューのランクを表す 0 から 9 までの値	CLWLQueueRank
注: Version が MQWQR_VERSION_3 より前のものである場合、以下のフィールドは無視されます。		
<i>DefPutResponse</i>	デフォルトの書き込み応答	DefPutResponse

クラスター・ワークロードのキュー・レコード構造体には、メッセージの有効な宛先の 1 つに関連する情報が含まれています。宛先キューのインスタンスごとに、1 つのクラスター・ワークロードのキュー・レコード構造体があります。

クラスター・ワークロードのキュー・レコード構造体は、すべての環境でサポートされます。

また、後方互換性のために、MQWQR1 構造体および MQWQR2 構造体も用意されています。

関連資料

[MQ_CLUSTER_WORKLOAD_EXIT](#) - 呼び出しの説明

このクラスター・ワークロード出口は、使用可能なキュー・マネージャーにメッセージをルーティングするためにキュー・マネージャーによって呼び出されます。

[MQXCLWLN](#) - クラスター・ワークロードのレコードのナビゲート

[MQXCLWLN](#) 呼び出しは、クラスター・キャッシュ内に保管された MQWDR、MQWQR、および MQWCR の各レコードのチェーン全体をナビゲートするために使用します。

[MQWXP](#) - クラスター・ワークロード出口のパラメーター構造体

以下の表には、MQWXP - クラスター・ワークロード出口のパラメーター構造体内のフィールドがまとめられています。

MQWDR - クラスター・ワークロード宛先レコード構造体

以下の表には、MQWDR - クラスター・ワークロードの宛先レコード構造体内のフィールドがまとめられています。

MQWCR - クラスター・ワークロードのクラスター・レコード構造体

以下の表には、MQWCR クラスター・ワークロードのレコード構造体内のフィールドがまとめられています。

MQWQR - クラスター・ワークロードのキュー・レコード構造体内のフィールド

MQWQR - クラスター・ワークロードのキュー・レコード構造体内のフィールドの説明。

StrucId (MQCHAR4) - 入力

クラスター・ワークロードのキュー・レコード構造体の構造体 ID。

- StrucId の値は MQWQR_STRUC_ID です。
- C プログラミング言語用に、定数の MQWQR_STRUC_ID_ARRAY も定義されています。この定数には、MQWQR_STRUC_ID と同じ値が含まれています。これは、ストリングではなく文字の配列です。

Version (MQLONG) - 入力

構造体のバージョン番号。Version では、次のいずれかの値を取ります。

MQWQR_VERSION_1

クラスター・ワークロードのキュー・レコードのバージョン 1。

MQWQR_VERSION_2

クラスター・ワークロードのキュー・レコードのバージョン 2。

MQWQR_VERSION_3

クラスター・ワークロードのキュー・レコードのバージョン 3。

MQWQR_CURRENT_VERSION

クラスター・ワークロードのキュー・レコードの現行バージョン。

StrucLength (MQLONG) - 入力

MQWQR 構造体の長さ。StrucLength では、次のいずれかの値を取ります。

MQWQR_LENGTH_1

クラスター・ワークロードのキュー・レコードのバージョン 1 の長さ。

MQWQR_LENGTH_2

クラスター・ワークロードのキュー・レコードのバージョン 2 の長さ。

MQWQR_LENGTH_3

クラスター・ワークロードのキュー・レコードのバージョン 3 の長さ。

MQWQR_CURRENT_LENGTH

クラスター・ワークロードのキュー・レコードの現行バージョンの長さ。

QFlags (MQLONG) - 入力

キュー・フラグはキューのプロパティを示します。以下のフラグが定義されます。

MQQF_LOCAL_Q

宛先はローカル・キューです。

MQQF_CLWL_USEQ_ANY

ローカルおよびリモート・キューを書き込みに使用することを許可します。

MQQF_CLWL_USEQ_LOCAL

ローカル・キューへの書き込みのみを許可します。

その他の値

このフィールドには、内部目的のためにキュー・マネージャーによって他のフラグが設定される場合があります。

QName (MQCHAR48) - 入力

メッセージの有効な宛先の 1 つであるキューの名前。

- QName の長さは MQ_Q_NAME_LENGTH です。

QMgrIdentifier (MQCHAR48) - 入力

キュー・マネージャー ID とは、MQWQR 構造体によって記述されたキューのインスタンスをホストするキュー・マネージャーの固有 ID です。

- この ID は、キュー・マネージャーによって生成されます。
- QMgrIdentifier の長さは MQ_Q_MGR_IDENTIFIER_LENGTH です。

ClusterRecOffset (MQLONG) - 入力

MQWQR 構造体に属している最初の MQWCR 構造体の論理オフセット。

- 静的キャッシュの場合、ClusterRecOffset は、MQWQR 構造体に属している最初の MQWCR 構造体のオフセットです。
- オフセット (相対位置) は、MQWQR 構造体の先頭からバイト単位で測定されます。
- 論理オフセットは、動的キャッシュのポインター演算に使用しないでください。次のレコードのアドレスを取得するには、MQXCLWLN 呼び出しを使用する必要があります。

QType (MQLONG) - 入力

宛先キューのキュー・タイプ。属性の値は以下のとおりです。

MQCQT_LOCAL_Q

ローカル・キュー。

MQCQT_ALIAS_Q

別名キュー。

MQCQT_REMOTE_Q

リモート・キュー。

MQCQT_Q_MGR_ALIAS

キュー・マネージャーの別名。

QDesc (MQCHAR64) - 入力

MQWQR 構造体によって記述された宛先キューのインスタンスをホストするキュー・マネージャーに定義されたキューの記述のキュー属性。

- QDesc の長さは MQ_Q_DESC_LENGTH です。

DefBind (MQLONG) - 入力

MQWQR 構造体によって記述された宛先キューのインスタンスをホストするキュー・マネージャーに定義されたデフォルトのバインディングのキュー属性。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。属性の値は以下のとおりです。

MQBND_BIND_ON_OPEN

MQOPEN 呼び出しで固定されたバインディング。

MQBND_BIND_NOT_FIXED

固定されていないバインディング。

MQBND_BIND_ON_GROUP

グループ内のメッセージすべてを同じ宛先のインスタンスに割り振る要求をアプリケーションが行えるようになります。

DefPersistence (MQLONG) - 入力

MQWQR 構造体によって記述された宛先キューのインスタンスをホストするキュー・マネージャーに定義されたデフォルトのメッセージ持続性のキュー属性。属性の値は以下のとおりです。

MQPER_PERSISTENT

メッセージは持続します。

MQPER_NOT_PERSISTENT

メッセージは持続しません。

DefPriority (MQLONG) - 入力

MQWQR 構造体によって記述された宛先キューのインスタンスをホストするキュー・マネージャーに定義されたデフォルトのメッセージ優先順位のキュー属性。優先順位の範囲は、0 から MaxPriority までです。

- 0 が最も低い優先順位です。
- MaxPriority は、宛先キューのこのインスタンスをホストするキュー・マネージャーのキュー・マネージャー属性です。

InhibitPut (MQLONG) - 入力

MQWQR 構造体によって記述された宛先キューのインスタンスをホストするキュー・マネージャーに定義された書き込み禁止のキュー属性。属性の値は以下のとおりです。

MQQA_PUT_INHIBITED

書き込み操作は使用禁止です。

MQQA_PUT_ALLOWED

書き込み操作が許可されています。

CLWLQueuePriority (MQLONG) - 入力

MQWQR 構造体によって記述された宛先キューのインスタンスをホストするキュー・マネージャーに定義されたクラスター・ワークロード・キュー優先順位属性。

CLWLQueueRank (MQLONG) - 入力

MQWQR 構造体によって記述された宛先キューのインスタンスをホストするキュー・マネージャーに定義されたクラスター・ワークロード・キュー・ランク。

DefPutResponse (MQLONG) - 入力

MQWQR 構造体によって記述された宛先キューのインスタンスをホストするキュー・マネージャーに定義されたデフォルトの書き込み応答キュー属性。属性の値は以下のとおりです。

MQPRT_SYNC_RESPONSE

MQPUT 呼び出しまたは MQPUT1 呼び出しへの同期応答。

MQPRT_ASYNC_RESPONSE

MQPUT 呼び出しまたは MQPUT1 呼び出しへの非同期応答。

MQWQR の初期値および言語ごとの宣言

MQWQR - クラスター・ワークロードのキュー・レコード用の初期値および C と高水準アセンブラー言語の宣言。

フィールド名	定数の名前	定数の値
<i>StrucId</i>	MQWQR_STRUC_ID_ARRAY	'WQR? '
<i>Version</i>	MQWQR_VERSION_1	1
<i>StrucLength</i>	MQWQR_CURRENT_LENGTH ³	212
<i>QFlags</i>	なし	0
<i>QName</i>	なし	""
<i>QMgrIdentifier</i>	なし	""
<i>ClusterRecOffset</i>	なし	0
<i>QType</i>	なし	0
<i>QDesc</i>	なし	""
<i>DefBind</i>	なし	0
<i>DefPersistence</i>	なし	0

表 29. MQWQR のフィールドの初期値 (続き)

フィールド名	定数の名前	定数の値
<i>DefPriority</i>	なし	0
<i>InhibitPut</i>	なし	0
<i>CLWLQueuePriority</i>	なし	0
<i>CLWLQueueRank</i>	なし	0
<i>DefPutResponse</i>	なし	1

注:

1. 記号? は、単一のブランク文字を表します。
2. C プログラミング言語では、マクロ変数 `MQWQR_DEFAULT` にデフォルト値が設定されています。このマクロ変数を以下の方法で使用して、構造体のフィールドに初期値を設定します。

```
MQWQR MyWQR = {MQWQR_DEFAULT};
```

3. 初期値では、構造体の長さを構造体のバージョン 1 ではなく、現行バージョンの長さに意図的に設定します。

C 宣言

```
typedef struct tagMQWQR {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    StrucLength;      /* Length of MQWQR structure */
    MQLONG    QFlags;          /* Queue flags */
    MQCHAR48  QName;           /* Queue name */
    MQCHAR48  QMgrIdentifier;   /* Queue-manager identifier */
    MQLONG    ClusterRecOffset; /* Offset of first cluster record */
    MQLONG    QType;           /* Queue type */
    MQCHAR64  QDesc;           /* Queue description */
    MQLONG    DefBind;         /* Default binding */
    MQLONG    DefPersistence;   /* Default message persistence */
    MQLONG    DefPriority;      /* Default message priority */
    MQLONG    InhibitPut;      /* Whether put operations on the queue
                               are allowed */

    /* version 2 */
    MQLONG    CLWLQueuePriority; /* Queue priority */
    MQLONG    CLWLQueueRank;    /* Queue rank */
    /* version 3 */
    MQLONG    DefPutResponse;   /* Default put response */
};
```

High Level Assembler

```
MQWQR          DSECT
MQWQR_STRUCID  DS    CL4      Structure identifier
MQWQR_VERSION  DS    F        Structure version number
MQWQR_STRUCLNGTH DS    F        Length of MQWQR structure
MQWQR_QFLAGS   DS    F        Queue flags
MQWQR_QNAME    DS    CL48     Queue name
MQWQR_QMGRIDENTIFIER DS    CL48 Queue-manager identifier
MQWQR_CLUSTERRECOFFSET DS    F  Offset of first cluster
*              record
MQWQR_QTYPE    DS    F        Queue type
MQWQR_QDESC    DS    CL64     Queue description
MQWQR_DEFBIND  DS    F        Default binding
MQWQR_DEFPERSISTENCE DS    F  Default message persistence
MQWQR_DEFPRIORITY DS    F    Default message priority
MQWQR_INHIBITPUT DS    F     Whether put operations on
*              the queue are allowed
MQWQR_DEFPUTRESPONSE DS    F  Default put response
MQWQR_LENGTH   EQU    *-MQWQR Length of structure
```

MQWCR - クラスター・ワークロードのクラスター・レコード構造体

以下の表には、MQWCR クラスター・ワークロードのレコード構造体内のフィールドがまとめられています。

フィールド	説明	参照ページ
<i>ClusterName</i>	クラスターの名前	ClusterName
<i>ClusterRecOffset</i>	次のクラスター・レコード (MQWCR) のオフセット	ClusterRecOffset
<i>ClusterFlags</i>	クラスター・フラグ	ClusterFlags

クラスター・ワークロードのクラスター・レコード構造体には、クラスターに関する情報が含まれています。宛先キューが属しているクラスターごとに、1つのクラスター・ワークロードのクラスター・レコード構造体があります。

クラスター・ワークロードのクラスター・レコード構造体は、すべての環境でサポートされます。

関連資料

[MQ_CLUSTER_WORKLOAD_EXIT - 呼び出しの説明](#)

このクラスター・ワークロード出口は、使用可能なキュー・マネージャーにメッセージをルーティングするためにキュー・マネージャーによって呼び出されます。

[MQXCLWLN - クラスター・ワークロードのレコードのナビゲート](#)

MQXCLWLN 呼び出しは、クラスター・キャッシュ内に保管された MQWDR、MQWQR、および MQWCR の各レコードのチェーン全体をナビゲートするために使用します。

[MQWXP - クラスター・ワークロード出口のパラメーター構造体](#)

以下の表には、MQWXP - クラスター・ワークロード出口のパラメーター構造体内のフィールドがまとめられています。

[MQWDR - クラスター・ワークロード宛先レコード構造体](#)

以下の表には、MQWDR - クラスター・ワークロードの宛先レコード構造体内のフィールドがまとめられています。

[MQWQR - クラスター・ワークロードのキュー・レコード構造体](#)

以下の表には、MQWQR - クラスター・ワークロードのキュー・レコード構造体内のフィールドがまとめられています。

MQWCR - クラスター・ワークロードのクラスター・レコード構造体内のフィールド。

MQWCR - クラスター・ワークロードのクラスター・レコード構造体内のフィールドの説明。

ClusterName (MQCHAR48) - 入力

MQWCR 構造体を所有する宛先キューのインスタンスが属しているクラスターの名前。宛先キューのインスタンスは、MQWDR 構造体によって記述されます。

- ClusterName の長さは MQ_CLUSTER_NAME_LENGTH です。

ClusterRecOffset (MQLONG) - 入力

次の MQWCR 構造体の論理オフセット。

- それ以上 MQWCR 構造体がない場合、ClusterRecOffset はゼロです。
- オフセット (相対位置) は、MQWCR 構造体の先頭からバイト単位で測定されます。

ClusterFlags (MQLONG) - 入力

クラスター・フラグは、MQWCR 構造体によって特定されたキュー・マネージャーのプロパティを示します。以下のフラグが定義されます。

MQQMF_REPOSITORY_Q_MGR

宛先は、フル・リポジトリのキュー・マネージャーです。

MQQMF_CLUSSDR_USER_DEFINED

クラスター送信側チャンネルが手動で定義されました。

MQQMF_CLUSSDR_AUTO_DEFINED

クラスター送信側チャンネルが自動で定義されました。

MQQMF_AVAILABLE

宛先キュー・マネージャーが、メッセージを受信できます。

その他の値

このフィールドには、内部目的のためにキュー・マネージャーによって他のフラグが設定される場合があります。

関連資料

[MQWCR の初期値および言語ごとの宣言](#)

MQWCR - クラスター・ワークロードのクラスター・レコード構造体用の初期値および C と高水準アセンブラー言語の宣言。

MQWCR の初期値および言語ごとの宣言

MQWCR - クラスター・ワークロードのクラスター・レコード構造体用の初期値および C と高水準アセンブラー言語の宣言。

フィールド名	定数の名前	定数の値
<i>ClusterName</i>	なし	""
<i>ClusterRecOffset</i>	なし	0
<i>ClusterFlags</i>	なし	0

C 宣言

```
typedef struct tagMQWCR {
    MQCHAR48 ClusterName; /* Cluster name */
    MQLONG ClusterRecOffset; /* Offset of next cluster record */
    MQLONG ClusterFlags; /* Cluster flags */
};
```

High Level Assembler

```
MQWCR          DSECT
MQWCR_CLUSTERNAME DS CL48 Cluster name
MQWCR_CLUSTERRECOFFSET DS F Offset of next cluster
* record
MQWCR_CLUSTERFLAGS DS F Cluster flags
MQWCR_LENGTH EQU *-MQWCR Length of structure
ORG MQWCR
MQWCR_AREA DS CL(MQWCR_LENGTH)
```

関連資料

[MQWCR - クラスター・ワークロードのクラスター・レコード構造体内のフィールド。](#)

[MQWCR - クラスター・ワークロードのクラスター・レコード構造体内のフィールドの説明。](#)

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 32. Windows、UNIX and Linux システム用のチャネル・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 (WebSphere MQ リスナーの実行)、runmqchl (WebSphere MQ チャネルの実行)、および runmqchi (WebSphere MQ チャネル・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

- UNIX and Linux システムでは、`export [environment variable]=filename` を使用します。
- Windows システムの場合は、`Set [environment variable]=filename` を使用します。
-

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することによって、`mqs.ini` ファイルのために使用する場所を変更できます。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_SSL_ALLOW_DEFAULT_CERT

AMQ_SSL_ALLOW_DEFAULT_CERT 環境変数が設定されていない場合、証明書に `ibmwebspheremq<userid>` というラベル名が含まれている場合にのみ、アプリケーションは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できます。

AMQ_SSL_ALLOW_DEFAULT_CERT 環境変数が設定されている場合、証明書には `ibmwebspheremq<userid>` 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 `ibmwebspheremq<userid>` 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。

V7.5.0.9

AMQ_SSL_LDAP_SERVER_VERSION

この変数を使用することにより、CRL サーバーによって特定のバージョンの LDAP プロトコルの使用が要求されるケースで、LDAP v2 または LDAP v3 のいずれかが IBM WebSphere MQ 暗号コンポーネントによって使用されるようにすることができます。

キュー・マネージャーまたはチャン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 WebSphere 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 WebSphere MQ MQI クライアントと IBM WebSphere MQ サーバーの両方がご使用の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 アプリケーションは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・バインディング・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、WebSphere 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、WebSphere 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、UNIX and Linux システム上の IBM WebSphere MQ MQI クライアントによって使用される構成情報が含まれます。

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 システム上の IBM WebSphere MQ MQI クライアントによって使用される構成情報が含まれます。

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。

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、WebSphere MQ MQI クライアント・ワークステーションにコピーすることができます。

MQCHLTAB

MQCHLTAB は、クライアント・チャンネル定義テーブル (CCDT) を含むファイルの名前を指定します。デフォルト・ファイル名は AMQCLCHL.TAB です。

MQC_IPC_HOST

IBM WebSphere 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 MQC_IPC_HOST を使用して myHostName を設定します。

MQCLNTCF

この環境変数を使用して、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、[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 WebSphere 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、[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

IBM WebSphere MQ for Windows、UNIX and Linux システムでは、この環境変数を、MQCONNX 呼び出しで使用される MQCNO 構造の Options フィールドに指定されたバインディングのタイプと組み合わせ使用します。[MQCONNX 環境変数を参照してください](#)。

MQ_FILE_PATH

Windows オペレーティング・システムでランタイム・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。"MQIPADDR_IPV4" または "MQIPADDR_IPV6" が考えられるストリング値です。これらの値は、ALTER QMGR IPADDRV において、IPV4 および IPV6 と同様の意味があります。設定されていない場合は、"MQIPADDR_IPV4" が想定されます。

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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere MQ classes for Java に付属しているスクリプトの中には、IVTRun など、この環境変数を使用するものがあります。

MQNAME

MQNAME では、IBM WebSphere 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を使用します。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、最小チャンネルを定義するために使用されます。SSL チャンネルの定義や、チャンネル出口があるチャンネルの定義に MQSERVER を使用することはできません。MQSERVER は、WebSphere MQ サーバーの位置および使用する通信方式を指定します。

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 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

MQSNOAUT では、オブジェクト権限マネージャー (OAM) を無効にして、セキュリティー検査を回避します。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。



警告: OAM を使用可能にするには、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する必要があります。

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 WebSphere 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、SSL 秘密鍵の再ネゴシエーションが行われる前に、SSL チャンネルで送受信される非暗号化バイト数を指定します。

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 WebSphere MQ が TCP 接続呼び出しを待つ時間を指定します。

MQSUIEB

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、MQSUIEB 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、トレース機能がオンに切り替わります。

OMQ_PATH

この環境変数では、ActiveX スクリプトに対応した IBM WebSphere 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 トレースの有効化に関する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 WebSphere MQ で .NET SOAP over JMS サービスのホスト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 WebSphere 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 WebSphere 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、WebSphere MQ Web transport for SOAP をインストールするときにも使用します。
『WebSphere MQ Web transport for SOAP のインストール』を参照してください。

分散プラットフォームにおけるメッセージ・チャネルの計画例

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 WebSphere 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。

例が示す事柄

この例は、使用できる WebSphere 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。2つの方法は似ていますが、コマンド行でコマンドを発行するには、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。

140 ページの図 4 はシナリオ例を示しています。

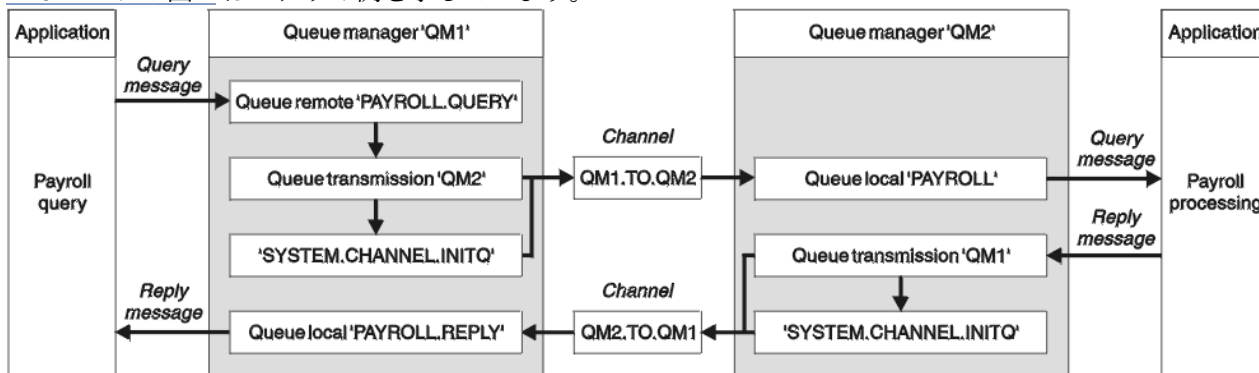


図 4. Windows および UNIX and Linux システム用のメッセージ・チャネルの例

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 [140 ページの図 4](#) に示されています。

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 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。

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) +
CONNAME('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')
```

例の実行

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については、[Windows での通信のセットアップ](#) および [UNIX and Linux システムでの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、それらをどのように使用できるかを理解します。

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 *
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)。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

日本アイ・ビー・エム株式会社

法務・知的財産

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

〒 103-8510

103-8510

東京 103-8510、日本

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 INTERNATIONAL BUSINESS MACHINES CORPORATION は、法律上の瑕疵担保責任、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。"" 国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N

Rochester, MN 55901

U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っていません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、名前や住所が類似する個人や企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

プログラミング・インターフェース情報 (提供されている場合) は、このプログラムで使用するアプリケーション・ソフトウェアの作成を支援することを目的としています。

本書には、プログラムを作成するユーザーが IBM WebSphere MQ のサービスを使用するためのプログラミング・インターフェースに関する情報が記載されています。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

重要: この診断、修正、およびチューニング情報は、変更される可能性があるため、プログラミング・インターフェースとして使用しないでください。

商標

IBM、IBM ロゴ、ibm.com[®]は、世界の多くの国で登録された IBM Corporation の商標です。現時点での IBM の商標リストについては、"Copyright and trademark information" www.ibm.com/legal/copytrade.shtml をご覧ください。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。

Microsoft および Windows は、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

この製品には、Eclipse Project (<http://www.eclipse.org/>) により開発されたソフトウェアが含まれています。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。



部品番号:

(1P) P/N: