9.3

IBM MQ の構成

注記

本書および本書で紹介する製品をご使用になる前に、1093 ページの『特記事項』に記載されている情報をお読みください。

本書は、 IBM® MQ バージョン 9 リリース 3、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

© Copyright International Business Machines Corporation 2007 年, 2025.

目次

構成.....7
Multiplatforms でのキュー・マネージャーの作成.....7
構成可能な一時ディレクトリー.....10
userdata ディレクトリー.....11
デフォルト・キュー・マネージャーの作成.....12
既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする方法.....13
キュー・マネージャーの作成後の構成ファイルのバックアップ.....14
クライアントとサーバー間の接続の構成.....15
使用する通信タイプ.....16
IBM MQ MQI client のセットアップ方法.....18
拡張トランザクション・クライアントの構成.....19
MQI チャネルの定義.....29
AMQP チャネルの作成および使用.....30
異なるプラットフォーム上でのサーバー接続およびクライアント接続の定義の作成.....35
サーバー側でのサーバー接続およびクライアント接続の定義の作成.....41
MQI チャネル用のチャネル出口プログラム.....57
キュー共有グループへのクライアントの接続.....61
IBM MQ 環境変数の使用.....62
環境変数の説明.....64
Multiplatforms での .ini ファイルの IBM MQ 構成情報の変更.....84
IBM MQ 構成ファイル mqs.ini.....86
キュー・マネージャー構成ファイル qm.ini.....98
インストール構成ファイル、mqinst.ini.....157
IBM MQ MQI client 構成ファイル、 mqclient.ini.....158
アクティビティー・トレース構成ファイル mqat.ini.....188
分散キューイングの構成.....190
IBM MQ 分散キューイング技法.....191
分散キュー管理の概要.....211
AIX, Linux, and Windows 上のチャネルのモニターおよび制御.....242
IBM i 上のチャネルのモニターおよび制御.....265
キュー・マネージャー・クラスターの構成.....287
均一クラスターの構成.....409
パブリッシュ/サブスクライブ・メッセージングの構成.....432
キューに入れられたパブリッシュ/サブスクライブのメッセージ属性の設定.....432
キューに入れられたパブリッシュ/サブスクライブの開始.....433
キューに入れられたパブリッシュ/サブスクライブの停止.....434
ストリームの追加.....434
ストリームの削除.....435
サブスクリプション・ポイントの追加.....436
分散パブリッシュ/サブスクライブ・ネットワークの構成.....437
複数のインストールの構成.....455
複数のインストール環境でのアプリケーションの接続.....455
プライマリー・インストールの変更.....463
キュー・マネージャーとインストールの関連付け.....464
システム上の IBM MQ のインストールの見つけ方.....466
高可用性、リカバリー、および再始動の構成.....467
クライアントの自動再接続.....468
コンソール・メッセージのモニター.....474
高可用性の構成.....479
ロギング: メッセージが失われないようにするための機能.....639
IBM MQ キュー・マネージャー・データのバックアップと復元.....671
クラスター・エラー・リカバリーの変更 (z/OS 以外のサーバー).....678

JMS および Jakarta Messaging リソースの構成....680
JNDI ネーム・スペースでの接続ファクトリーおよび宛先の構成....681
IBM MQ Explorer を使用した JMS 2.0 オブジェクトの構成....685
管理ツールを使用した JMS および Jakarta Messaging オブジェクトの構成....686
WebSphere Application Server での JMS 2.0 リソースの構成....696
最新のリソース・アダプター保守レベルを使用するための WebSphere Application Server の構成....706
JMS **PROVIDERVERSION** プロパティーの構成....708
WebSphere Application Server 永続サブスクリプションの削除....716
Managed File Transfer の構成....719
MFT 構成オプション (Multiplatforms)....719
z/OS での MFT 構成オプション....721
Redistributable Managed File Transfer components のダウンロードおよび構成....721
MFT エージェントまたはロガーのコマンド・データ・セットの作成....727
Managed File Transfer for z/OS の構成....728
IBM i での MFT の構成....760
初めて使用する際の MFT の構成....762
MFT エージェント・キュー・マネージャーの構成....772
MFT ロガーの構成....782
Connect:Direct ブリッジの構成....816
IBM MQ Console および REST API の構成....821
mqweb サーバーの基本構成....822
スタンドアロン IBM MQ Web Server の構成....826
セキュリティーの構成....827
HTTP ホスト名の構成....827
HTTP および HTTPS ポートの構成....828
応答タイムアウトの構成....830
自動始動の構成....831
ロギングの構成....832
LTPA トークンの構成....835
IBM MQ Console のリモート・キュー・マネージャー接続の動作の構成....837
administrative REST API ゲートウェイの構成....839
messaging REST API の構成....840
REST API for MFT の構成....846
mqweb サーバーの JVM のチューニング....852
IBM MQ Console および REST API のインストール・コンポーネントのファイル構造....853
mqweb サーバーの構成のバックアップとリストア....856
Linux または Windows プラットフォームでの Aspera gateway 接続の定義....858
IBM Cloud Private 計量サービスで使用するための IBM MQ の構成....863
IBM Cloud Private の計量サービス・インスタンスで使用するためのキュー・マネージャーの構成....864
HTTP プロキシーを介した IBM Cloud Private 計量サービスへの接続....867
計量サービスへの接続のトラブルシューティング....867
Salesforce プッシュ・トピックおよびプラットフォーム・イベントで使用するための IBM MQ の構成....868
IBM MQ Bridge to Salesforce の構成....870
IBM MQ Bridge to Salesforce の追加の構成オプション....874
Salesforce プラットフォーム・イベントのイベント・メッセージの作成....877
IBM MQ Bridge to Salesforce の実行....883
ブロックチェーンで使用するための IBM MQ の構成....885
IBM MQ Bridge to blockchain の構成ファイルの作成....887
Hyperledger Fabric ネットワーク資格情報ファイルの例....889
IBM MQ 9.2.0 からの IBM MQ Bridge to blockchain のメッセージ・フォーマット....891
IBM MQ Bridge to blockchain の実行....892
IBM MQ Bridge to blockchain の追加の構成オプション....897
ブロックチェーンで使用するための IBM MQ Advanced for z/OS VUE の構成....898
z/OS での IBM MQ Bridge to blockchain の構成ファイルの作成....900
z/OS 上の IBM MQ Bridge to blockchain の IBM MQ セキュリティー構成....902
z/OS での IBM MQ Bridge to blockchain の実行....903

z/OS でのキュー・マネージャーの構成 908
z/OS でのキュー・マネージャーのカスタマイズ準備 909
IBM MQ for z/OS の設定 914
z/OS でのキュー・マネージャーのテスト 980
z/OS での他のキュー・マネージャーとの通信のセットアップ 988
IBM MQ と IMS の使用 1019
IBM MQ と CICS の使用 1027
言語環境プログラムまたは z/OS 呼び出し可能サービスのアップグレードおよび保守の適用 1027
IMS における OTMA 出口の使用 1029
IBMz/OSMF を使用して自動化します IBM MQ 1033
MFT エージェントからリモート z/OS キュー・マネージャーへの接続を有効にする 1045
IBM MQ Internet Pass-Thru の構成 1045
MQIPT での HTTP のサポート 1046
MQIPT での SOCKS サポート 1047
MQIPT での SSL/TLS サポート 1049
Java security manager (MQIPT) 1077
MQIPT のセキュリティー出口 1080
MQIPT のポート番号制御 1084
MQIPT に保管されているパスワードの暗号化 1085
MQIPT のその他のセキュリティーに関する考慮事項 1087
MQIPT の接続ログ 1088
コンテナーを使用した IBM MQ Internet Pass-Thru の構成 1089
ストリーミング・キューの構成 1090

特記事項 1093
プログラミング・インターフェース情報 1094
商標 1094

IBM MQ の構成

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

このタスクについて

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

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

キュー・マネージャーを作成、開始、停止、および削除する方法については、7 ページの『Multiplatforms でのキュー・マネージャーの作成』を参照してください。

IBM MQ インストール済み環境とアプリケーションを結び付けるのに必要なコンポーネントを作成する方法については、190 ページの『分散キューイングの構成』を参照してください。

さまざまな方式でクライアントを IBM MQ サーバーに接続する方法については、15 ページの『クライアントとサーバー間の接続の構成』を参照してください。

キュー・マネージャー・クラスターを構成する方法については、287 ページの『キュー・マネージャー・クラスターの構成』を参照してください。

構成情報を変更することによって、IBM MQ またはキュー・マネージャーの動作を変更できます。 詳しくは、84 ページの『Multiplatforms での .ini ファイルの IBM MQ 構成情報の変更』を参照してください。 一般に、この製品資料で示されている場合を除いて、構成変更を有効にするためにキュー・マネージャーを再始動する必要はありません。

z/OS IBM MQ for z/OS® の構成方法については、908 ページの『z/OS でのキュー・マネージャーの構成』を参照してください。

関連概念

IBM MQ の技術概要

関連タスク

ローカル IBM MQ オブジェクトの管理

リモート IBM MQ オブジェクトの管理

IBM i IBMi の管理

z/OS IBM MQ for z/OS の管理

計画

z/OS z/OS での IBM MQ 環境の計画

908 ページの『z/OS でのキュー・マネージャーの構成』

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

Multi Multiplatforms でのキュー・マネージャーの作成

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

© Copyright IBM Corp. 2007 年, 2025

始める前に

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

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

キュー・マネージャーとインストール済み環境の関連付け
キュー・マネージャーを作成するには、IBM MQ 制御コマンド **crtmqm** を使用します。 **crtmqm** コマンドは、キュー・マネージャーを、その **crtmqm** コマンドを発行したインストール済み環境と自動的に関連付けます。 キュー・マネージャー上で作動するコマンドの場合は、そのキュー・マネージャーと関連付けられたインストール済み環境から、そのコマンドを発行する必要があります。 キュー・マネージャーに関連付けられたインストール済み環境は、setmqm コマンドを使用して変更できます。 Windows インストーラーは、インストールを実行するユーザーを mqm グループに追加しないことに注意してください。詳しくは、 AIX®, Linux®, and Windows 上の IBM MQ を管理する権限を参照してください。

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

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

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

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

注 : 制御コマンドでは、キュー・マネージャー名は大文字小文字の区別がされます。 つまり、jupiter.queue.manager および JUPITER.queue.manager という名前の 2 つのキュー・マネージャーを作成することが可能です。 しかし、複雑な問題点もあります。

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

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

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

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

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

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

デフォルトのキュー・マネージャーの作成については、12 ページの『デフォルト・キュー・マネージャーの作成』で説明されています。

送達不能キューの指定

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

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

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

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

キュー・マネージャーを作成するときは、**-u** フラグを使用して送達不能キューの名前を指定します。MQSC コマンドを使用することによって、既に定義したキュー・マネージャーの属性を変更し、使用する送達不能キューを指定することもできます。 MQSC コマンドの ALTER の例については、キュー・マネージャーの属性の表示および変更を参照してください。

デフォルト伝送キューの指定

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

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

必要なロギング・パラメーターの指定

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

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

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

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

ロギング・パラメーターの詳細については、467 ページの『高可用性、リカバリー、および再始動の構成』を参照してください。

AIX **IBM MQ for UNIX システムにのみ該当します**

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

このタスクについて

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

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

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

手順

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

関連概念

キュー・マネージャーの処理

関連タスク

QM1 という名前のキュー・マネージャーの作成

84 ページの『Multiplatforms での .ini ファイルの IBM MQ 構成情報の変更』

構成 (.ini) ファイル内の情報を編集することにより、IBM MQ または個々のキュー・マネージャーの動作を、ご使用のインストール済み環境のニーズに合わせて変更できます。 IBM MQ MQI clients の構成オプションを変更することもできます。

908 ページの『z/OS でのキュー・マネージャーの構成』

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

関連資料

システムおよびデフォルト・オブジェクト

crtmqm

Linux AIX 構成可能な一時ディレクトリー

構成可能な一時ディレクトリーは、キュー・マネージャーに対する一時データの移動先を定義します。 これを使用すると、AIX and Linux ドメイン・ソケットを Red Hat® OpenShift® 環境内のマウントされていないファイル・システムに配置できます。

それ以前の IBM MQ 9.2.0、AIX and Linux プラットフォームでは、キュー・マネージャーを実行すると、/var/mqm/sockets ディレクトリの下に AIX and Linux ドメイン・ソケットが作成されます。 /var/mqm をマウントされたファイル・システムとしてコンテナー内でキュー・マネージャーを実行すると、一部の Linux プラットフォームではこれらのドメイン・ソケットが作成されないようにすることができます。これは、コンテナーの外部からのプロセスがコンテナー内の操作を妨害することができるためです。 この問題により、デフォルトのセキュリティー・コンテキストにおいては IBM MQ が Red Hat OpenShift コンテナー・プラットフォームで実行できなくなります。

IBM MQ 9.2.0 以降、**EphemeralPrefix** 属性を使用して一時ディレクトリーの場所を構成できるようになりました。 この属性を使用しない場合、動作は変わりません。

mqs.ini で (**crtmqm** または **addmqinf** コマンドを使用して) キュー・マネージャー項目が作成されると、以下の場合に **EphemeralPrefix** 属性が追加されます。

- 90 ページの『mqs.ini ファイルの AllQueueManagers スタンザ』に **DefaultEphemeralPrefix** 属性を設定します。
- **MQ_EPHEMERAL_PREFIX** 環境変数を設定します。
- **-v** EphemeralPrefix は、 **addmqinf** コマンドにのみ指定してください。

EphemeralPrefix 属性は、既存のキュー・マネージャーの停止時に明示的に追加することもできます。これは、キュー・マネージャーの再始動時に追加されます。

EphemeralPrefix 属性を指定すると、キュー・マネージャーの開始時に、キュー・マネージャーに対する一時データが、通常のロケーションではなくその接頭部の下に作成されます。 具体的には、以下の数を示します。

- 通常、/var/mqm/sockets/<QM>の下にあるソケット・ファイルは、現在 /<EphemeralPrefix>/sockets/<QM> 下にあります
- /<Prefix>/qmgrs/<QM>/@<Subpool>の下に存在するサブプール・ファイルは、現在/<EphemeralPrefix>/qmgrs/<QM>/@<Subpool>の下にあります

注:

- /var/mqm/sockets/@SYSTEM は固定ロケーションに残り、**EphemeralPrefix** 属性の一部ではありません。
- AMQCLCHL.TAB は /<Prefix>/qmgrs/<QM>/@ipcc の下に残り、**EphemeralPrefix** 属性の一部ではありません。

EphemeralPrefix 属性に含めることができる文字数は、ご使用のプラットフォームによって異なります。

- Linux AIX AIX and Linux プラットフォームでは、12 文字に制限されています。
- IBM i IBM i では、24 文字に制限されています。

指定する **EphemeralPrefix** 属性が長すぎる、または存在しない場合、AMQ7001E メッセージを受信します:

```
AMQ7001E:キュー・マネージャーに対して指定されたロケーションが無効です
```

.

Multi userdata ディレクトリー

userdata ディレクトリーを使用して、永続的なアプリケーション状況を保管することができます。

各 IBM MQ キュー・マネージャーには、その永続的な状態の専用ファイル・システムがあります (キュー・データとリカバリー・ログの両方を含む)。 ファイル・システムには、アプリケーションの永続状況情報を保管するために使用できる userdata ディレクトリーが含まれています。 UNIX および Linux システムでのディレクトリーの内容および Windows システムでのディレクトリーの内容を参照してください。

userdata ディレクトリーは、以下のようないくつかの状況で役立つ場合があります:

- RDQM 構成において、キュー・マネージャーが別のノードにフェイルオーバーしたときにアプリケーション情報も移動するようにする (581 ページの『アプリケーションの持続状況の保管』を参照)。
- 複数インスタンスのキュー・マネージャーの場合、それらのアプリケーションの状態が、そのキュー・マネージャー・データと共に共有ネットワーク・ファイル・システム上に配置されるようにする。
- 一般に、アプリケーションがキュー・マネージャー・サービスとして構成される場合。

アプリケーション状態を userdata ディレクトリーに保管することを選択した場合は、このロケーションに書き込まれるデータが、キュー・マネージャーに割り振られた使用可能なディスク・スペースを消費する可能性があることを認識しておく必要があります。 キュー・マネージャーがキュー・データ、ログ、およびその他の永続的な状態の情報を書き込むために使用できるだけの十分なディスク・スペースが残っていることを確認する必要があります。

userdata ディレクトリーには mqm ユーザーおよびグループの所有権があり、ユーザーは IBM MQ 管理者グループ(つまり mqm)でなくても、ユーザーがアクセスできるようになっています。 userdata ディレクトリーの許可を変更することはできませんが、必要な所有権および許可を使用して、ディレクトリーにコンテンツを作成することができます。

Multi デフォルト・キュー・マネージャーの作成

デフォルト・キュー・マネージャーとは、アプリケーションが MQCONN 呼び出しでキュー・マネージャー名を指定しなかった場合に、そのアプリケーションが接続されるキュー・マネージャーのことです。 また、これは、キュー・マネージャー名を指定せずに **runmqsc** コマンドを呼び出したときに MQSC コマンドを処理するキュー・マネージャーでもあります。 キュー・マネージャーを作成するには、IBM MQ 制御コマンド **crtmqm** を使用します。

始める前に

デフォルト・キュー・マネージャーを作成する前に、7 ページの『Multiplatforms でのキュー・マネージャーの作成』で説明されている考慮事項を参照してください。

Linux AIX **crtmqm** を使用して AIX and Linux 上にキュー・マネージャーを作成するときに、 /var/mqm/qmgrs/*qmgr* ディレクトリーが既に存在し、mqm によって所有され、空である場合は、そのディレクトリーがキュー・マネージャー・データ用に使用されます。 ディレクトリーが mqm によって所有されていない場合、キュー・マネージャーの作成は失敗し、First Failure Support Technology (FFST) メッセージが表示されます。 ディレクトリーが空でない場合は、キュー・マネージャー・データ用に新たにディレクトリーが作成されます。

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

このタスクについて

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

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

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

crtmqm コマンドおよびその構文の詳細については、**crtmqm** を参照してください。

手順

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

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

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

ここで、

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

-d MY.DEFAULT.XMIT.QUEUE
このキュー・マネージャーによって使用されるデフォルト伝送キューの名前です。

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

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

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

次のタスク

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

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

キュー・マネージャーの属性の表示および変更
関連資料
システムおよびデフォルト・オブジェクト

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

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

このタスクについて

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

Windows Linux Windows および Linux (x86 および x86-64 プラットフォーム) システムで、IBM MQ Explorer を使用してこの変更を行う場合は、14 ページの『IBM MQ Explorer を使用してキュー・マネージャーをデフォルトにする方法』を参照してください。

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

手順

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

関連タスク

84 ページの『Multiplatforms での .ini ファイルの IBM MQ 構成情報の変更』

構成 (.ini) ファイル内の情報を編集することにより、IBM MQ または個々のキュー・マネージャーの動作を、ご使用のインストール済み環境のニーズに合わせて変更できます。 IBM MQ MQI clients の構成オプションを変更することもできます。

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

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

このタスクについて

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

テキスト・エディターを使用してこの変更を手動で行うには、13 ページの『既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする方法』を参照してください。

手順

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

ALW キュー・マネージャーの作成後の構成ファイルのバックアップ

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

このタスクについて

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

構成ファイルには、次の 2 つのタイプがあります。

- 製品をインストールすると、IBM MQ 構成ファイル(mqs.ini)が作成されます。 これには、キュー・マネージャーのリストが含まれます。このファイルは、キュー・マネージャーを作成したり、削除したりするたびに更新されます。 ノードごとに 1 つの mqs.ini ファイルがあります。

- 新しいキュー・マネージャーを作成すると、新しい構成ファイル(qm.ini)が自動的に作成されます。 これには、そのキュー・マネージャーの構成パラメーターが含まれます。

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

- Windows Windows システムで:amqp_win.properties
- Linux AIX AIX and Linux システムで:amqp_unix.properties

関連タスク

84 ページの『Multiplatforms での .ini ファイルの IBM MQ 構成情報の変更』
構成 (.ini) ファイル内の情報を編集することにより、IBM MQ または個々のキュー・マネージャーの動作を、ご使用のインストール済み環境のニーズに合わせて変更できます。 IBM MQ MQI clients の構成オプションを変更することもできます。

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

クライアントとサーバー間の接続の構成

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

このタスクについて

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

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

別々の物理ネットワーク上に存在する、またはファイアウォールを介して通信する、IBM MQ MQI client とキュー・マネージャーの間に MQI チャネルを定義する場合は、IBM MQ Internet Pass-Thru を使用すると構成が簡単になる可能性があります。 詳しくは、IBM MQ Internet Pass-Thru を参照してください。

手順

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

関連概念

158 ページの『IBM MQ MQI client 構成ファイル、 mqclient.ini』
テキスト・ファイル内の属性を使用して、クライアントを構成することができます。 この属性は、環境変数、または他のプラットフォーム固有の方法を使用してオーバーライドすることができます。

関連タスク

62 ページの『IBM MQ 環境変数の使用』

コマンドを使用して、現在の設定を表示したり、IBM MQ 環境変数の値をリセットしたりすることができます。

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

関連資料

DISPLAY CHLAUTH

SET CHLAUTH

使用する通信タイプ

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

MQI チャネル用の伝送プロトコルのタイプ

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

- TCP/IP
- LU 6.2
- NetBIOS
- SPX

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

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

表 1. 伝送プロトコル - *IBM MQ MQI client* とサーバーのプラットフォームの組み合わせ

伝送プロトコル	IBM MQ MQI client	IBM MQ サーバー
TCP/IP 17 ページの『1』	IBM i AIX Linux Windows	IBM i AIX Linux Windows z/OS
LU 6.2	AIX Linux17 ページの『2』 Windows	IBM i AIX Linux17 ページの『2』 Windows z/OS
NetBIOS	Windows	Windows
SPX	Windows	Windows

注:

1. MQ Adv. CD TCP/IP を使用するメッセージ・チャネルは、ネットワーク・スループットを大幅に向上させることができる高速 TCP/IP トンネルを提供する IBM Aspera® faspio Gateway を指すことができます。 Linux または Windows での Aspera gateway 接続の定義を参照してください。
2. Linux (POWER プラットフォーム) を除く

関連概念

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

260 ページの『AIX and Linux での TCP 接続の定義』
送信側のチャネル定義では、宛先のアドレスを指定します。 受信側の接続については、リスナーまたは inet デーモンが構成されます。

280 ページの『IBM i での TCP 接続の定義』
「接続名」フィールドを使用して、チャネル定義内に TCP 接続を定義できます。

1010 ページの『z/OS での TCP 接続の定義』
TCP 接続を定義するために、構成するいくつかの設定があります。

255 ページの『Windows での LU 6.2 接続の定義』
2 つのマシン間で LU 6.2 の会話が確立できるように SNA を構成する必要があります。

264 ページの『AIX and Linux での LU 6.2 接続の定義』
2 つのマシン間で LU 6.2 の会話が確立できるように SNA を構成する必要があります。

282 ページの『IBM i での LU 6.2 接続の定義』
完全修飾された LU 6.2 接続のモード名、TP 名、および接続名を使用して、LU 6.2 通信の詳細を定義します。

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

関連タスク

858 ページの『Linux または Windows プラットフォームでの Aspera gateway 接続の定義』
IBM Aspera faspio Gateway は、IBM MQ のネットワーク・スループットを大幅に向上させることができる高速 TCP/IP トンネルを提供します。 資格のあるプラットフォームで実行されているキュー・マネージャーは、 Aspera gateway を介して接続できます。 ゲートウェイ自体は、 Red Hat または Ubuntu Linux、あるいは Windows にデプロイされます。

関連資料

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

1012 ページの『APPC/MVS を使用した z/OS 用の LU6.2 接続の定義』
LU6.2 接続を定義するために、構成するいくつかの設定があります。

TCP/IP 接続の制限

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

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

表 2. 1 つの TCP/IP ポートでキューに入れられる未解決の接続要求の最大数	
サーバーのプラットフォーム	**接続要求の最大数**
AIX	100
Linux	100
IBM i	255
Windows サーバー	100
Windows ワークステーション	100
z/OS	255

接続の制限に達した場合、クライアントは MQCONN を呼び出してから、MQRC_HOST_NOT_AVAILABLE というリターン・コードを受け取り、クライアント・エラー・ログに AMQ9202 エラーを記録します(システム/var/mqm/errors/AMQERR0n.LOG の上 AIX and Linux、または amqerr0n.logIBM MQ クライアント・インストールのエラーサブディレクトリにあります Windows)。 クライアントが MQCONN を再試行すると、正常に処理される場合があります。

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

IBM MQ MQI client のセットアップ方法

クライアントを設定するには、次の指示に従ってください。

始める前に

IBM MQ MQI client を設定するには、IBM MQ サーバーがインストール済みで作動している必要があり、クライアントはこのサーバーに接続します。

手順

1. IBM MQ MQI クライアントに適したプラットフォームがあること、およびハードウェアとソフトウェアが要件を満たしていることを確認してください。

   プラットフォーム・サポートについては、 IBM MQ クライアントのプラットフォーム・サポートで説明されています。
2. クライアント・ワークステーションに IBM MQ をインストールする方法を決定し、クライアントとサーバーのプラットフォームの特定の組み合わせに応じた指示に従います。

   インストールについては、以下のトピックで説明します。

   - AIX での IBM MQ クライアントのインストール
   - Linux での IBM MQ クライアントのインストール
   - Windows での IBM MQ クライアントのインストール
   - IBM i での IBM MQ クライアントのインストール
3. 通信リンクが設定され、接続されていることを確認します。

   通信リンクの構成については、サーバーとクライアント間の接続の構成で説明されています。
4. インストールが正しく行われたことを確認します。

   社内で使用しているプラットフォームに該当するインストール手順の検証のセクションを参照してください。

5. IBM MQ MQI client のインストールを確認した後、クライアントを保護する必要があるかどうか考慮します。

   クライアントのセキュリティーについては、IBM MQ MQI client・セキュリティーのセットアップで説明されています。
6. IBM MQ MQI クライアントと、クライアント上で実行する IBM MQ アプリケーションが必要とするサーバーとの間のチャネルをセットアップします。

   チャネルの設定については、MQI チャネルの定義で説明されています。 TLS を使用している場合はその他の考慮事項がいくつかあります。

   これらの考慮事項については、MQI チャネルで TLS を使用するよう指定するで説明されています。 IBM MQ MQI client 構成ファイルまたは IBM MQ 環境変数を使用してチャネルを設定することが必要な場合があります。 IBM MQ 環境変数については、IBM MQ 環境変数の使用で説明されています。
7. IBM MQ アプリケーションの詳細については、 アプリケーションの開発 を参照してください。
8. IBM MQ MQI client 環境でアプリケーションを設計、ビルド、および実行する際には、キュー・マネージャー環境との違いを考慮する必要があります。

   これらの相違点については、以下を参照してください。

   - クライアント・アプリケーションでのメッセージ・キュー・インターフェース (MQI) の使用
   - IBM MQ MQI clients 用のアプリケーションの作成
   - キュー・マネージャーへの IBM MQ MQI client・アプリケーションの接続
   - IBM MQ MQI clients に関する問題の解決

拡張トランザクション・クライアントの構成

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

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

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

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

WebSphere® Application Server
WebSphere Application Server 6 以降には IBM MQ メッセージング・プロバイダーが組み込まれているため、拡張トランザクション・クライアントを使用する必要はありません。

XA 対応トランザクション・マネージャーの構成

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

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

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

- AIX AIX での IBM MQ クライアントのインストール
- Linux Linux での IBM MQ クライアントのインストール
- Windows Windows での IBM MQ クライアントのインストール
- IBM i IBM i での IBM MQ クライアントのインストール

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

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

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

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

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

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

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

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

- xa_open ストリング
- XA スイッチ構造を指すポインター

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

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

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

XA スイッチ構造により、トランザクション・マネージャーは、拡張トランザクション・クライアントが提供する xa_ 関数を見付けることができます。また、XA スイッチ構造は、キュー・マネージャーが動的登録を使用するかどうかを指定します。 拡張トランザクション・クライアントに付属の XA スイッチ構造については、25 ページの『XA スイッチ構造』を参照してください。

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

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

関連概念

24 ページの『xa_open ストリングの CHANNEL、TRPTYPE、CONNAME、および QMNAME パラメーター』
以下の情報を使用して、拡張トランザクション・クライアントがこれらのパラメーターを使用して、接続先のキュー・マネージャーを判別する方法を理解します。

25 ページの『xa_open の追加のエラー処理』
特定の状況で、xa_open 呼び出しは失敗します。

関連タスク

26 ページの『TLS チャネルを持つ拡張トランザクション・クライアントの使用』
TLS チャネルは xa_open ストリングを使用してセットアップすることができません。 クライアント・チャネル定義テーブル (CCDT) を使用するには、以下の指示に従ってください。

関連資料

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

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

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

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

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

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

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

この状態が発生したトランザクション・マネージャーの場合は、以下のアクションを実行して、トランザクション・マネージャーが IBM MQ for z/OS と対話できるようにします。

IBM MQ for z/OS 上の XA クライアント接続に対する変更が有効になっていることを確認してください。 これにより、トランザクション・マネージャーが別の接続でトランザクションを準備できるようになります。

注 :

- デフォルトでは、この変更は有効になっていません。 変更を利用するには、XA クライアントが使用する SVRCONN チャネルの記述フィールドの任意の場所に、キーワード CSQSERVICE1 (大文字) を指定する必要があります。
- CSQSERVICE1 キーワードを使用するチャネルには、以下の制約事項があります。
  - リカバリー単位 GROUP の処理を実行できません。 リカバリー単位 QMGR の処理のみが可能です。 後処理は、xa_open 呼び出しで指定された名前によって決定されます。 キュー共有グループ名が使用されている場合、XA 接続はグループ・リカバリー単位を要求します。

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

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

この変更は、 IBM MQ for z/OS 9.0 で APAR P173410 によって有効にされました。

関連概念

19 ページの『XA 対応トランザクション・マネージャーの構成』
まず、IBM MQ ベース・クライアントを構成し、次に以下のトピックにある情報を使用して、拡張トランザクション機能を構成します。

xa_open ストリングの形式

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

xa_open ストリングの形式は次のとおりです。

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

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

パラメーターの名前、意味、および有効な値は、次のとおりです。

名前
意味と有効な値

CHANNEL
MQI チャネルの名前。

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

TRPTYPE
MQI チャネルの通信プロトコル。 以下のプロトコルは有効な値です。

LU62
SNA LU 6.2

NETBIOS
NetBIOS

SPX
IPX/SPX

TCP
TCP/IP

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

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

LU62
CPI-C サイド情報項目を識別するシンボリック宛先名。

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

NETBIOS
NetBIOS の名前。

SPX
4 バイトのネットワーク・アドレス、6 バイトのノード・アドレス、およびオプションの 2 バイトのソケット番号。 これらの値は、16 進数表記で指定する必要があります。 ネットワーク・アドレ

スとノード・アドレスの間をピリオドで区切る必要があります。ソケット番号が指定される場合、小括弧で囲む必要があります。 以下に例を示します。

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

ソケット番号を省略すると、デフォルト値 5e86 が取られます。

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

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

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

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

これは必須パラメーターです。

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

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

z/OS GROUP リカバリー単位の使用方法について詳しくは、キュー共用グループ内のリカバリー単位を参照してください。

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

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

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

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

これは任意指定のパラメーターです。

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

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

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

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

xa_open ストリングの例は次のとおりです。

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

xa_open ストリングの *CHANNEL*、*TRPTYPE*、*CONNAME*、および *QMNAME* パラメーター

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

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

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

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

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

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

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

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

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

TPM および *AXLIB* パラメーター

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

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

表 *3.* 想定される *AXLIB* パラメーターの値

TPM の値	**プラットフォーム**	**想定される AXLIB の値**
CICS	AIX	/usr/lpp/encina/lib/libEncServer.a(EncServer_shr.o)
CICS	Windows Windows システム	libEncServer
Tuxedo	AIX	/usr/lpp/tuxedo/lib/libtux.a(libtux.so.60)
Tuxedo	Windows Windows システム	libtux

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

xa_open の追加のエラー処理

特定の状況で、xa_open 呼び出しは失敗します。

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

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

拡張トランザクション処理における障害後のリカバリー

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

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

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

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

XA スイッチ構造

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

これらのスイッチ構造は、次のとおりです。

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

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

これらのスイッチ構造は、26 ページの表 4 に表示されているライブラリー内にあります。

表 *4. XA* スイッチ構造が入っている *IBM MQ* ライブラリー

プラットフォーム	XA スイッチ構造が入っているライブラリー
AIX AIX Linux Linux	*MQ_INSTALLATION_PATH*/lib/libmqcxa
Windows Windows システム	*MQ_INSTALLATION_PATH*¥bin¥mqcxa.dll [1]

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

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

関連概念
26 ページの『動的登録および拡張トランザクション処理』
動的登録の使用は、最適化の 1 つの形式です。これは、動的登録が、トランザクション・マネージャーによって発行される xa_ 関数呼び出しの数を減らすことができるからです。

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

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

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

TLS チャネルを持つ拡張トランザクション・クライアントの使用
TLS チャネルは xa_open ストリングを使用してセットアップすることができません。 クライアント・チャネル定義テーブル (CCDT) を使用するには、以下の指示に従ってください。

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

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

手順

1. 必須の qmname (キュー・マネージャーの名前) パラメーターだけを含む xa_open ストリングを指定します (例えば XA_Open_String=qmname=MYQM)。
2. キュー・マネージャーを使って、必須の TLS パラメーターと共に CLNTCONN (クライアント接続) チャネルを定義します。 CLNTCONN 定義の QMNAME 属性にキュー・マネージャーの名前を組み込みます。これは xa_open ストリングの qmname とマッチングされます。
3. クライアント・システムのクライアント・チャネル定義テーブル (CCDT) または Windows の場合はアクティブ・ディレクトリーで CLNTCONN 定義を利用できるようにします。
4. CCDT を使用している場合は、環境変数 MQCHLLIB および MQCHLTAB を使用している CLNTCONN チャネルの定義を含む CCDT を識別します。 クライアント・アプリケーションとトランザクション・マネージャーの両方で使用される環境でこれらの変数を設定します。

タスクの結果

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

CICS 用の拡張トランザクション・クライアントの構成

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

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

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

以下のプラットフォームごとに、CICS が使用するためのスイッチ・ロード・ファイルが 1 つ用意されています。

- AIX
- Windows

各スイッチ・ロード・ファイルには、動的登録に使用される XA スイッチ構造 MQRMIXASwitchDynamic を指すポインターを戻す関数が含まれています。 各スイッチ・ロード・ファイルの完全修飾パス名については、27 ページの表 5 を参照してください。

表 *5.* スイッチ・ロード・ファイル

プラットフォーム	スイッチ・ロード・ファイル
AIX Linux	*MQ_INSTALLATION_PATH*/lib/amqczsc (lib/amqczsc)
Windows	*MQ_INSTALLATION_PATH*¥bin¥mqcc4swi.dll [1]

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

Windows システム用の XAD リソース定義の例は、次のとおりです。

```
cicsadd -c xad -r REGION1 WMQXA \
    ResourceDescription="IBM MQ queue manager MARS" \
    XAOpen="channel=MARS.SVR,trptype=tcp,conname=MARS(1415),qmname=MARS,tpm=cics" \
    SwitchLoadFile="C:&#xa5;Program Files&#xa5;IBM&#xa5;MQ\bin\mqcc4swi.dll"
```

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

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

- CICS 領域に追加できる IBM MQ の XAD リソース定義は 1 つのみです。 つまり、1 つの領域には 1 つのキュー・マネージャーしか関連付けることができません。また、領域内で実行されるすべての CICS アプリケーションは、そのキュー・マネージャーにしか接続できません。 別のキュー・マネージャーに接続する CICS アプリケーションを実行したい場合は、別の領域でそのアプリケーションを実行する必要があります。
- 領域内の各アプリケーション・サーバーは、初期化時に xa_open を呼び出し、その領域に関連したキュー・マネージャーとの MQI チャネルを開始します。 つまり、アプリケーション・サーバーが始動する前に、キュー・マネージャーを始動しておく必要があります。そうしないと、xa_open 呼び出しが失敗します。 それ以降にアプリケーション・サーバーによって処理されるすべての IBM MQ MQI client・アプリケーションは、同じ MQI チャネルを使用します。
- MQI チャネルが開始するときに、チャネルのクライアント側にセキュリティー出口がない場合、クライアント・システムからサーバー接続 MCA に流れるユーザー ID は `cics` です。 特定の状況では、サーバー接続 MCA がクライアント・アプリケーションに代わって後でキュー・マネージャー・リソースにアクセスしようとするときに、キュー・マネージャーはこのユーザー ID を権限検査に使用します。 このユーザー ID が権限検査に使用される場合、必要なすべてのリソースにアクセスする権限がこのユーザー ID にあることを確認する必要があります。

  キュー・マネージャーがこのユーザー ID を権限検査に使用する場合については、保護を参照してください。
- IBM MQ クライアント・システムで使用するために提供されている CICS タスク終了出口は、 28 ページの表 6 にリストされています。 これらの出口の構成方法は、IBM MQ サーバー・システム用の対応する出口の構成方法と同じです。 したがって、この情報については、CICS ユーザー出口の有効化を参照してください。

表 6. *CICS* タスク終了出口

プラットフォーム	Source	ライブラリー
AIX AIX Linux Linux	amqzscgx.c	amqczscg
Windows Windows システム	amqzscgn.c	mqcc1415.dll

Tuxedo 用の拡張トランザクション・クライアントの構成

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

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

- アプリケーション用の Tuxedo UBBCONFIG ファイルの GROUPS セクションで、**OPENINFO** パラメーターを使用して xa_open ストリングを指定します。 この方法の例は、UBBCONFIG サンプル・ファイルを参照してください。このファイルは、Tuxedo サンプル・プログラムで使用するために提供されています。

  AIX 以下のプラットフォームでは、ファイルの名前は `ubbstxcx.cfg` です:

  – AIX

  Windows Windows、ファイルの名前は `ubbstxcn.cfg` です。
- Tuxedo リソース・マネージャー・テーブル内のキュー・マネージャーの項目で、XA スイッチ構造の名前、およびその構造が入っているライブラリーの完全修飾パス名を指定します。

  – AIX で AIX、`udataobj/RM` を指定する。

– Windows Windows で、udataobj\rm を指定します。

プラットフォームごとにこれを行う方法の例については、TUXEDO サンプルを参照してください。 Tuxedo は、リソース・マネージャーの動的登録をサポートするので、MQRMIXASwitch か MQRMIXASwitchDynamic のどちらでも使用できます。

Windows Microsoft トランザクション・サーバー

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

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

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

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

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

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

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

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

MQI チャネルの定義

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

ユーザー定義チャネル

サーバーがチャネルを自動的に定義しない場合、チャネル定義を作成し、 IBM MQ MQI client マシン上の IBM MQ アプリケーションにチャネルへのアクセス権限を付与する方法は 2 つあります。

この 2 つの方法については、以下で詳しく説明します。

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

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

   この方式を使用する方法についての詳細は、35 ページの『異なるプラットフォーム上でのサーバー接続およびクライアント接続の定義の作成』を参照してください。
2. 両方のチャネル定義をサーバー・マシン側で作成する方法

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

   この方式を使用する方法についての詳細は、41 ページの『サーバー側でのサーバー接続およびクライアント接続の定義の作成』を参照してください。

チャネルの自動定義

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

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

関連概念

220 ページの『チャネル制御機能』
チャネル制御機能は、チャネルを定義、モニター、および制御するための機能を備えています。

ALW AMQP チャネルの作成および使用

AMQP サービス・コンポーネントの IBM MQ サポートを IBM MQ インストール済み環境にインストールすると、IBM MQ MQSC コマンド (**runmqsc**) を実行して、チャネルの定義、変更、削除、開始、および停止を行うことができます。 チャネルの状況を表示することもできます。

始める前に

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

- AIX AIX システム用の IBM MQ コンポーネント
- Linux Linux システム用の IBM MQ rpm コンポーネント
- Linux Linux Ubuntu システム用の IBM MQ Debian コンポーネント
- Windows Windows システム用の IBM MQ 機能

注 : SERVICE コンポーネントの例と、AMQP サービスが正常に機能しなくなった場合の詳細については、AMQP 用の IBM MQ サービスの再始動 を参照してください。

この作業は、既存のキュー・マネージャーがあることも前提としています。

キュー・マネージャーへのテスト接続を行うには、OASIS AMQP 1.0 プロトコルを実装する任意の AMQP クライアント (Apache Qpid Proton や Apache Qpid JMS などの MQ Light および Apache Qpid クライアントなど) を使用できます。

V 9.3.0 IBM MQ 9.3.0 からは、デフォルト・チャネル SYSTEM.DEF.AMQP のみを使用して、キュー・マネージャーへの MQ Light 接続をテストすることができます。 以下の手順では、デフォルトのチャネルを使用します。

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

注 : AMQP チャネルはユーザー定義の AMQP サービスをサポートしていません。 AMQP チャネルはシステム・デフォルトの SYSTEM.AMQP.SERVICE サービスのみをサポートしています。 このサービスのインスタンスは、キュー・マネージャーごとに 1 つだけ定義できます。

手順

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

   ```
   runmqsc QMNAME
   ```

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

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

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

   注 : IBM MQ 9.1 以降、SYSTEM.AMQP.SERVICE の **CONTROL** 属性が *QMGR* に設定されます。 そのため、キュー・マネージャーの開始時にサービスが自動的に開始されます。 **CONTROL** 属性を *MANUAL* に設定すると、キュー・マネージャーの開始時にサービスが開始されないようにすることができます。

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

3. MCAUSER ユーザー ID を設定します。

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

   注 : Windows Windows では、IBM MQ 9.2.0 より前は、MCAUSER ユーザー ID として設定できるのは、長さが 12 文字以内のユーザー ID のみとなります。 IBM MQ 9.2.0 より、12 文字の長さ制限は取り除かれています。

   a) **ALTER CHANNEL** コマンドを使用して、 MCAUSER ユーザー ID を設定します。

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

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

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

   および

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

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

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

4. **START CHANNEL** コマンドを使用して、デフォルトの SYSTEM.DEF.AMQP チャネル:

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

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

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

チャネルが正しく実行されると、コマンド出力に STATUS(RUNNING) が表示されます。

6. デフォルトのポートを変更します。

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

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

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

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

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

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

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

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

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

- デフォルトでは、キュー・マネージャーの **CERTLABL** 属性に対応するラベルを持つキュー・マネージャーのキー・リポジトリー証明書は、チャネルの SSL/TLS 暗号化によって使用される名前です。 **CERTLABL** を設定することにより、別の証明書を選択できます。 **ALTER CHANNEL** コマンドを使用して、必要な証明書のラベルを指定します。

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

- チャネルが SSL/TLS クライアント接続で証明書を必要とするように設定することができます。 **SSLCAUTH** 属性を設定することにより、SSL/TLS クライアント接続からの証明書が必要かどうかを選択できます。 **ALTER CHANNEL** コマンドを使用して、SSL/TLS クライアント接続で証明書を必要とするかどうかを設定します。 以下に例を示します。

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

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

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

あるいは、チャネル認証レコードを使用して接続を許可またはブロックすることもできます。この方法は、**SSLPEER** 属性を使用する場合よりも細分性が高いためです。 **SSLPEER** の設定および代替方法としてのチャネル認証レコードの使用についての詳細情報は、SSL ピアを参照してください。

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

```
npm install mqlight
```

10. node_modules/mqlight/samples ディレクトリーに移動し、サンプルの受信側アプリケーションを実行します:
    - デフォルトのポート番号を使用している場合は、このサンプルの受信側アプリケーションを実行できます。

      ```
      node recv.js
      ```

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

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

    デフォルトのチャネルへの接続が成功すると、次のメッセージが表示されます。

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

    アプリケーションはキュー・マネージャーに接続され、メッセージの受信を待機しています。 これはトピック「public」にサブスクライブされます。

    注: -i パラメーターを使用して指定しない限り、client-id は自動的に生成されます。
11. 新規コマンド・ウィンドウで、node_modules/mqlight/samples ディレクトリーに移動し、以下のコマンドを実行してサンプルの送信側アプリケーションを実行します:

    ```
    node send.js
    ```

    受信側アプリケーションのコマンド・ウィンドウに、Hello World メッセージが表示されます。
12. **AMQSSUB** IBM MQ サンプルを使用して、MQ Light サンプル・メッセージを受信します。

    Linux および Windows では、サンプルは以下の場所にあります:
    - Linux Linux 上の *mqinstall*/samp/bin ディレクトリー
    - Windows Windows 上の *mqinstall*/Tools\c\Samples\Bin ディレクトリー

    a) 次のコマンドを実行して、サンプルを実行します。

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

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

    ```
    node send.js
    ```

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

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

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

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

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

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

次のタスク

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

DISPLAY CONN(*) TYPE(*CONN*) WHERE (*CHANNEL EQ SYSTEM.DEF.AMQP*)
AMQP チャネルによってキュー・マネージャーに作成された IBM MQ 接続を表示します。

DISPLAY CHSTATUS(*) CHLTYPE(*AMQP*) CLIENTID(*) ALL
指定されたチャネルに接続されている AMQP クライアントのリストを表示します。

STOP CHANNEL (*MY.AMQP.CHANNEL*)
AMQP チャネルを停止し、リスニング中のポートを閉じます。

DELETE CHANNEL (*MY.AMQP.CHANNEL*)
作成したすべてのチャネルを削除します。

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

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

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

関連タスク
AMQP クライアント・アプリケーションの開発
AMQP クライアントの保護
関連資料
strmqm

ALW キュー・マネージャーからの AMQP チャネルの削除

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

手順

1. キュー・マネージャーを停止させます。
2. AMQP サービス・コンポーネント API の IBM MQ サポートを削除します。
   - AIX AIX では、 以下のコマンドを実行します。

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

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

     ```
     rpm -e MQSeriesAMQP
     ```

   - Windows Windows で、IBM MQ インストール済み環境から amqp フォルダーを除去します。 それ以外のファイルやフォルダーを IBM MQ インストール・パスから削除することがないように注意してください。
3. キュー・マネージャーを再始動する。

関連タスク
AMQP クライアント・アプリケーションの開発
AMQP クライアントの保護

ALW AMQP チャネルのログ・ファイル

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

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

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

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

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

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

すべての FDC ファイルは、AMQP*.FDC ファイルとして作成されます。このファイルは、*data-directory*/errors フォルダーに書き込まれます。

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

関連概念

AIX, Linux, and Windows 用エラー・ログ

関連タスク

AMQP クライアント・アプリケーションの開発

AMQP クライアントの保護

異なるプラットフォーム上でのサーバー接続およびクライアント接続の定義の作成

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

このタスクについて

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

z/OS z/OS では、操作パネルおよび制御パネルも使用できます。

IBM i IBM i では、パネル・インターフェースも使用できます。

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

runmqsc の場合、以下の考慮事項が適用されます。

- **-c** パラメーターを指定し、オプションで **-u** パラメーターを指定して、 管理するキュー・マネージャーへのクライアントとして **runmqsc** を接続することができます。
- **-u** パラメーターを使用してユーザー ID を指定する場合、対応するパスワードを求めるプロンプトが出されます。
- CHCKLOCL(REQUIRED) または CHCKLOCL(REQDADM) を指定して CONNAUTH AUTHINFO レコードを構成した場合、**-u** パラメーターを使用する必要があります。このパラメーターを使用しないと **runmqsc** でキュー・マネージャーを管理することはできません。

手順

- サーバーでサーバー接続チャネルを定義するには、36 ページの『サーバー側でのサーバー接続チャネルの定義』を参照してください。
- **MQSERVER** 環境変数を使用して IBM MQ MQI client 上にクライアント接続チャネルを作成するには、36 ページの『MQSERVER を使用した IBM MQ MQI client でのクライアント接続チャネルの作成』を参照してください。
- MQCONNX 呼び出しで MQCNO 構造を使用して IBM MQ MQI client 上にクライアント接続チャネルを作成するには、40 ページの『MQCNO を使用した IBM MQ MQI client でのクライアント接続チャネルの作成』を参照してください。

サーバー側でのサーバー接続チャネルの定義

必要に応じて、MQSC を開始し、次にサーバー接続チャネルを定義します。

手順

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

   a) 例えば QM1 と呼ばれるキュー・マネージャーを作成します。

   ```
   crtmqm QM1
   ```

   b) キュー・マネージャーを始動します。

   ```
   strmqm QM1
   ```

   c) MQSC コマンドを始動します。

   ```
   runmqsc QM1
   ```

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

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

   このチャネル定義は、サーバー上で稼働するキュー・マネージャーに関連しています。
3. 以下のコマンドを使用して、キュー・マネージャーへのインバウンド接続アクセスを許可します。

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

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

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

MQSERVER を使用した IBM MQ MQI client でのクライアント接続チャネルの作成

MQSERVER 環境変数を使用して、クライアント・ワークステーション上にクライアント接続チャネルを定義できます。

このタスクについて

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

MQSERVER 環境変数を使用して IBM MQ MQI client マシンとサーバー・マシンの間のチャネルを定義する場合、これはアプリケーションで使用可能な唯一のチャネルであり、クライアント・チャネル定義テーブル (CCDT) に対する参照は行われません。

MQCONN 要求または MQCONNX 要求で、リスナーが接続されているキュー・マネージャー以外のキュー・マネージャーが指定されている場合、または **MQSERVER** パラメーター *TransportType* が認識されない場合、MQCONN 要求または MQCONNX 要求は戻りコード MQRC_Q_MGR_NAME_ERROR で失敗します。

Linux AIX AIX and Linux では、以下のいずれかの例のように **MQSERVER** を定義できます。

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56(2002)'
export MQSERVER=CHANNEL1/LU62/BOX99
```

すべての MQCONN または MQCONNX 要求は、MQCD 構造体が MQCONNX に提供された MQCNO 構造体から参照されていない限り、定義されたチャネルを使用しようとします。参照されている場合、MQCD 構造体によって指定されたチャネルが、 **MQSERVER** 環境変数によって指定されたチャネルより優先されます。

MQSERVER 環境変数は、 **MQCHLLIB** および **MQCHLTAB** 環境変数が指すどのクライアント・チャネル定義よりも優先されます。

手順

- ご使用のプラットフォームに応じて、以下のいずれかのコマンドを使用して、**MQSERVER** でチャネル定義を指定します。
  - Windows Windows では、以下のように単純なチャネル定義を指定します。

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

    以下に例を示します。

    ```
    export  MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
    ```

  - Linux AIX AIX and Linux では、以下のように単純なチャネル定義を指定します。

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

    以下に例を示します。

    ```
    SET  MQSERVER=SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)
    ```

  - IBM i IBM i では、以下のように単純なチャネル定義を指定します。

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

    以下に例を示します。

    ```
    ADDENVVAR ENVVAR(MQSERVER) VALUE('SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)')
    ```

    注:

  - *ChannelName* は、サーバーで定義されているものと同じ名前でなければなりません。 スラッシュ (/) 文字を含めることはできません。この文字は、チャネル名、トランスポート・タイプ、および接続名を区切るために使用されるためです。 **MQSERVER** 環境変数を使用してクライアント・チャネルを定義する場合、最大メッセージ長 (**MAXMSGL**) として 100 MB が使用されます。 そのため、チャネルの最大メッセージ・サイズは、実際にはサーバーの SVRCONN チャネルに指定した値となります。

- *TransportType* は、 IBM MQ クライアント・プラットフォームに応じて、 LU62、 TCP、 NETBIOS、 SPX のいずれかになります。
- Linux AIX AIX and Linux では、 *TransportType* に大/小文字の区別があり、大文字にする必要があります。 トランスポート・タイプが認識されない場合、 MQCONN または MQCONNX 呼び出しは 2058 を返します。
- *ConnectionName* は、通信プロトコル (*TransportType*) に定義されているサーバーの名前です。 これは、完全修飾ネットワーク名 (例えば、 AMACHINE.ACOMPANY.COM(1414)) でなければなりません。
- *ConnectionName* は、接続名のコンマ区切りリストにすることができます。 リスト内の接続名は、クライアント接続テーブルでの複数接続の場合と同じように使用されます。 接続名リストは、クライアントが試行する複数の接続を指定するために、キュー・マネージャー・グループの代わりに使用される場合があります。 複数インスタンス・キュー・マネージャーを構成する場合は、接続名リストを使用して別のキュー・マネージャー・インスタンスを指定することができます。

- **MQSERVER** を取り消して、 **MQCHLLIB** および **MQCHLTAB** が指すクライアント・チャネル定義テーブルに戻るには、次のコマンドを入力します。
  - Linux AIX On AIX and Linux:

    ```
    unset MQSERVER
    ```

  - Windows On Windows:

    ```
    SET MQSERVER=
    ```

例

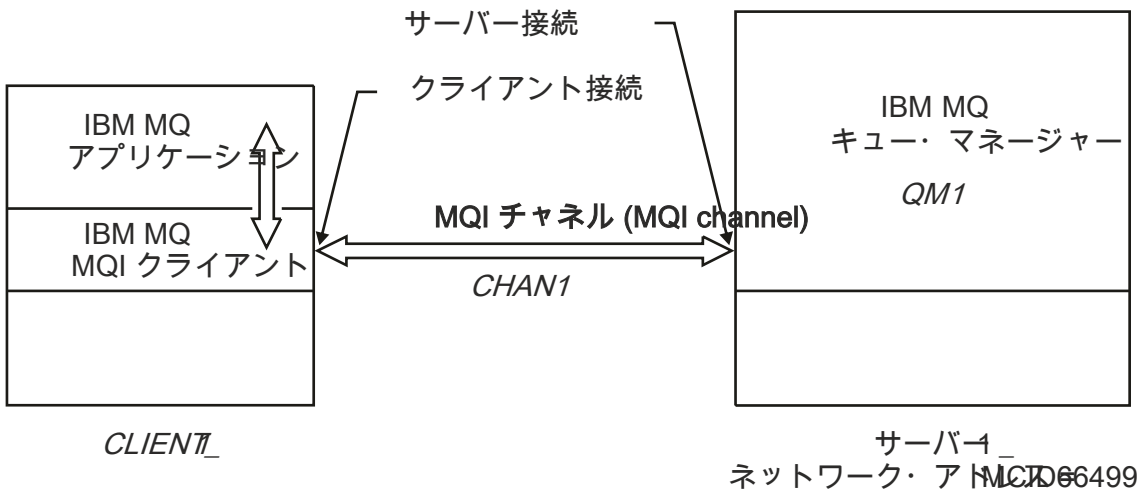


図 1. 単純なチャネル定義の例

38 ページの図 1 に示されている単純なチャネル定義を作成するには、以下のコマンドを使用します。

- Linux AIX On AIX and Linux:

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

- Windows On Windows:

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

注 : TCP/IP ポート番号の変更方法については、 39 ページの『TCP/IP デフォルト・ポートの変更』を参照してください。

簡単なチャネル定義の例をさらにいくつか挙げます。

- Windows On Windows:

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

- Linux AIX On AIX and Linux:

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

この B0X99 は、LU 6.2 ConnectionName です。

- IBM i On IBM i:

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

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

関連タスク

62 ページの『IBM MQ 環境変数の使用』
コマンドを使用して、現在の設定を表示したり、IBM MQ 環境変数の値をリセットしたりすることができます。

40 ページの『MQCNO を使用した IBM MQ MQI client でのクライアント接続チャネルの作成』
MQCONNX 呼び出しで MQCNO 構造を使用することにより、クライアント・ワークステーションでクライアント接続チャネルを定義できます。

TCP/IP デフォルト・ポートの変更

TCP/IP の場合、IBM MQ は、デフォルトではチャネルがポート 1414 に接続されるものと想定します。 必要に応じて、デフォルト値を変更できます。

このタスクについて

以下の 3 つのオプションのいずれかを使用して、ポート番号を変更できます。

- **MQSERVER** 環境変数を使用します。
- `mqclient.ini` ファイルを変更する。
- IBM MQ をサービス・ファイルに追加します。

手順

- 環境変数'**MQSERVER** を使ってポート番号を変更するには、例えば、*ConnectionName* の最後の部分に括弧でポート番号を追加する：
  - Linux AIX On AIX and Linux:

    ```
    export  MQSERVER='ChannelName/TransportType/ConnectionName(PortNumber)'
    ```

  - Windows On Windows:

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

- `mq.ini` ファイルを使用してポート番号を変更するには、プロトコル名にポート番号を追加します。以下に例を示します。

```
TCP:
port=2001
```

- IBM MQ をサービス・ファイルに追加してポート番号を変更するには、 261 ページの『AIX and Linux での TCP/IP リスナーの使用』で説明されている手順を実行します。

Windows *SPX* デフォルト・ソケットの変更

デフォルトでは、SPX の場合、IBM MQ は、チャネルがソケット 5E86 に接続されることを前提としています。 必要に応じて、デフォルト値を変更できます。

このタスクについて

以下のいずれかのオプションを使用して、ポート番号を変更できます。

- **MQSERVER** 環境変数を使用します。

  SPX 接続の場合は、ConnectionName とソケットを network.node(socket) の形式で指定します。 IBM MQ クライアントとサーバーが同じネットワークにある場合は、ネットワークを指定する必要はありません。 デフォルトのソケットを使用する場合は、ソケットを指定する必要はありません。
- mqclient.ini file.Changing qm.ini ファイルを変更する。

手順

- **MQSERVER** 環境変数を使用して SPX 接続のポート番号を変更するには、以下の例に示すように、*ConnectionName* とソケットを network.node(socket) の形式で指定します。

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

  注: IBM MQ クライアントとサーバーが同じネットワーク上にある場合は、ネットワークを指定する必要はありません。 デフォルト・ソケットを使用している場合は、ソケットを指定する必要はありません。
- qm.ini ファイルを使用してポート番号を変更するには、プロトコル名にポート番号を追加します。以下に例を示します。

  ```
  SPX:
  socket=5E87
  ```

MQCNO を使用した IBM MQ MQI client でのクライアント接続チャネルの作成

MQCONNX 呼び出しで MQCNO 構造を使用することにより、クライアント・ワークステーションでクライアント接続チャネルを定義できます。

このタスクについて

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

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

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

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

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

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

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

関連タスク

36 ページの『MQSERVER を使用した IBM MQ MQI client でのクライアント接続チャネルの作成』
MQSERVER 環境変数を使用して、クライアント・ワークステーション上にクライアント接続チャネルを定義できます。

サーバー側でのサーバー接続およびクライアント接続の定義の作成

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

このタスクについて

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

- すべてのプラットフォームで、IBM MQ Script (MQSC) コマンド、プログラマブル・コマンド・フォーマット (PCF) コマンドを使用して、サーバー・マシンでサーバー接続チャネルを定義できます。
- Windows Linux Linux および Windows では、IBM MQ Explorer を使用することもできます。
- z/OS z/OS では、操作パネルおよび制御パネルも使用できます。
- IBM i IBM i では、パネル・インターフェースも使用できます。

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

手順

1. サーバー接続チャネルを定義するには、54 ページの『サーバー側でのサーバー接続チャネルの定義』を参照してください。
2. クライアント接続チャネルを定義するには、55 ページの『サーバー側でのクライアント接続チャネルの定義』を参照してください。

関連タスク

42 ページの『バイナリー形式の CCDT の構成』
クライアント・チャネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャネル定義と認証情報を判別します。 Multiplatforms では、キュー・マネージャーの作成時に、デフォルト設定が組み込まれたバイナリー CCDT が自動的に作成されます。 バイナリー CCDT を更新する場合は、**runmqsc** コマンドを使用します。

54 ページの『サーバー側でのサーバー接続チャネルの定義』
キュー・マネージャー用にサーバー接続チャネルの定義を作成します。

55 ページの『サーバー側でのクライアント接続チャネルの定義』
サーバー接続チャネルを定義した後、対応するクライアント接続チャネルを定義することができます。

56 ページの『クライアント接続チャネル定義へのアクセス』
クライアント・チャネル定義テーブル (CCDT) をクライアント・アプリケーションで使用できるようにするには、CCDT をコピーまたは共有してからクライアント・コンピューターでその位置と名前を指定します。URL からクライアント・チャネル定義テーブル（CCDT）を探すこともできる。

クライアント・チャネル定義テーブルの構成

クライアント・チャネル定義テーブル (CCDT) では、クライアント接続チャネルとそれぞれの属性を定義します。 クライアントはこのファイルを読み取って、接続先のキュー・マネージャーを判別します。 CCDT ファイルは、JSON 形式またはバイナリー形式のいずれかにすることができます。

このタスクについて

キュー・マネージャーは、CCDT ファイルを読み取りません。 用途は、クライアントにチャネル定義と認証情報を提供することに限られています。

IBM MQ 9.2.0 の前は、バイナリー形式の CCDT しかありませんでした。 IBM MQ 9.2.0 から、JavaScript Object Notation (JSON) 形式の CCDT も作成できるようになりました。

バイナリー形式の CCDT は、キュー・マネージャーの作成時に自動的に作成されます。 このテーブルに保管されているクライアント・チャネル定義を更新するには、**runmqsc** コマンドのみを使用します。

JSON 形式の CCDT は、.json という拡張子の付いたフラット・テキスト・ファイルです。 このテーブルの作成や更新は手作業で行うので、**runmqsc** コマンドを使用する場合よりも制限が少なくなります。

z/OS アプリケーション・サーバーで実行する z/OS JMS クライアントは、CCDT を使用して、リモート・キュー・マネージャーへの接続の詳細情報を参照します。 IBM MQ for z/OS 9.1 以降、IBM MQ Advanced for z/OS Value Unit Edition では、JMS クライアントが他の z/OS LPAR にあるキュー・マネージャーにリモート接続できるようになりました。 したがって、そのクライアントも CCDT を使用できます。

クライアントで使用する CCDT の構成に役立つ資料を以下のタスクの中から選択してください。

手順

- 42 ページの『バイナリー形式の CCDT の構成』
- 44 ページの『JSON 形式の CCDT の構成』
- 51 ページの『CCDT の場所』
- 52 ページの『CCDT への URL アクセス』

関連概念

MQI クライアント: クライアント・チャネル定義テーブル (CCDT)

関連タスク

409 ページの『均一クラスターの構成』

均一クラスターによって、スケーリングと可用性を重視したアプリケーションを設計することができます。また均一クラスターは、均一クラスター内のどのキュー・マネージャーにも接続できます。

バイナリー形式の *CCDT* の構成

クライアント・チャネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャネル定義と認証情報を判別します。 Multiplatforms では、キュー・マネージャーの作成時に、デフォルト設定が組み込まれたバイナリー CCDT が自動的に作成されます。 バイナリー CCDT を更新する場合は、**runmqsc** コマンドを使用します。

始める前に

IBM MQ 9.1.2 以降、JavaScript Object Notation (JSON) 形式でも CCDT を作成できるようになりました。この代替形式には、バイナリー CCDT に比べて幾つかのメリットがあります。 44 ページの『JSON 形式の CCDT の構成』を参照してください。

どのプラットフォームのクライアントも CCDT を表示したり使用したりできます。 ただし、バイナリー CCDT を作成したり変更したりできるのは、IBM MQ for Multiplatforms の場合に限られます。

このタスクについて

Multi On マルチプラットフォーム:

- バイナリー CCDT は、キュー・マネージャーのデータ・ディレクトリーの下の`@ipcc` ディレクトリーに自動的に作成されます。
- キュー・マネージャーに関連付けられたバイナリー CCDT は、自動的に作成されるだけでなく、オブジェクト定義との同期状態が保持されます。 クライアント・チャネル・オブジェクトを定義、変更、削除す

る場合、同じ操作の一部としてキュー・マネージャー・オブジェクト定義と CCDT 内のエントリーの両方が更新されます。

注:

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

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

- バイナリー CCDT の場所や内容を変更する場合は、**runmqsc** コマンドを使用します。

どのプラットフォームのクライアントもバイナリー CCDT を表示したり使用したりできます。

手順

- Multi
  デフォルトのバイナリー CCDT を作成します。

  マルチプラットフォームでは、キュー・マネージャーを作成すると、デフォルトのバイナリー CCDT が AMQCLCHL.TAB という名前で作成されます。

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

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

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

  - Linux AIX AIX and Linux システムの場合:

    /prefix/qmgrs/*QUEUEMANAGERNAME*/@ipcc

    AIX and Linux システムでは、*QUEUEMANAGERNAME* によって参照されるディレクトリーの名前には大/小文字の区別があります。 キュー・マネージャー名に特殊文字が含まれている場合、ディレクトリー名は、キュー・マネージャー名と同じにならない可能性があります。

  - Windows On Windows:

    MQ_INSTALLATION_PATH\data\qmgrs*QUEUEMANAGERNAME*\@ipcc

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

  ただし、キュー・マネージャー・データ用に別のディレクトリーの使用を選択している場合もあります。 **crtmqm** コマンドを使用した場合は、パラメーター **-md** *DataPath* を指定できます。 これを行うと、AMQCLCHL.TAB は指定したデータ・パスの@ipcc ディレクトリーに置かれます。
- 以下で CCDT を見つけます。
  - クライアント・コンピューター上
  - 複数のクライアントが共有する場所
  - サーバー上で、共有ファイルとして

  51 ページの『CCDT の場所』 を参照してください。

  a) クライアント・マシンでバイナリー CCDT を直接作成します。
  - **-n** パラメーターを指定して runmqsc コマンドを使用します。
  - CCDT は、**MQCHLLIB** で指定した場所に、**MQCHLTAB** で指定したファイル名 (デフォルトは AMQCLCHL.TAB) で作成されます。
  - **重要:-n** パラメーターを指定する場合は、他のパラメーターを指定してはなりません。

b) 場所を変更します。

CCDT のパスを変更する場合は、**MQCHLLIB** を設定します。 なお、同じサーバーに複数のキュー・マネージャーがある場合は、どのキュー・マネージャーも CCDT の場所を共有します。

- CCDT にアクセスします。

  以下のようにして、CCDT にアクセスできます。

  - **MQCCDTURL** 環境変数を定義することにより、ファイル、FTP、または HTTP URL からリモートでアクセスする。
  - **MQCHLLIB** 環境変数と **MQCHLTAB** 環境変数を設定することによりローカルでアクセスする。
  - クライアント構成ファイル内の CHANNELS スタンザの **ChannelDefinitionDirectory** 属性および **ChannelDefinitionFile** 属性を定義することにより、ローカルでアクセスする。

  さまざまな例については、51 ページの『CCDT の場所』 を参照してください。
- CCDT の内容を表示したり編集したりします。

  CCDT の内容を表示するには、**runmqsc** コマンドを使用します。

  1. CCDT にアクセスするための環境変数を設定します。
  2. `runmqsc -n` コマンドを実行します。
  3. DISPLAY CHANNEL(*) コマンドなどを実行します。

  Multi マルチプラットフォーム では、**runmqsc** コマンドを使用してバイナリー CCDT の内容を編集することもできます。 CCDT の各項目は、特定のキュー・マネージャーへのクライアント接続を表します。 **DEFINE CHANNEL** コマンドを使用してクライアント接続チャネルを定義すると新しい項目が追加され、**ALTER CHANNEL** コマンドを使用してクライアント接続チャネルを変更すると項目が更新されます。 このコマンドの使用例をさらに確認したい場合は、**runmqsc** を参照してください。
- クライアントに認証情報を提供し、TLS 証明書が失効しているかどうかの検査を行うことができます。

  a) 認証情報オブジェクトを含んだ名前リストを定義します。

  b) キュー・マネージャー属性 **SSLCRLNL** を名前リストの名前に設定します。

関連概念

取り消された証明書の取り扱い

関連タスク

44 ページの『JSON 形式の CCDT の構成』

クライアント・チャネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャネル定義と認証情報を判別します。 テキスト・エディターを使用して、JavaScript Object Notation (JSON) CCDT を作成したり更新したりできます。

JSON 形式の CCDT の構成

クライアント・チャネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャネル定義と認証情報を判別します。 テキスト・エディターを使用して、JavaScript Object Notation (JSON) CCDT を作成したり更新したりできます。

始める前に

Multi IBM MQ for Multiplatforms を使用している場合は、キュー・マネージャーの作成時に自動的に作成されるバイナリー CCDT を使用することも可能です。 42 ページの『バイナリー形式の CCDT の構成』 を参照してください。

このタスクについて

JSON 形式の CCDT スキーマのファイル名は次のとおりです。

Linux

```
/opt/mqm/lib/ccdt_schema.json
```

Windows
```
C:\Program Files\IBM\MQ\bin\ccdt_schema.json
```

デフォルトの JSON CCDT はありません。また IBM MQ には、JSON 形式の CCDT を作成したり編集したりするためのツールは用意されていません。 それでも、**runmqsc** コマンドでバイナリー CCDT を操作するよりも、JSON CCDT を手動で作成する方が、構成オプションの幅が広がります。

- IBM MQ for Multiplatforms を使用していなくても、JSON CCDT ファイルの作成や編集は可能です。
- JSON 形式では、同じ名前の重複チャネル定義を作成できます。 クラウドに IBM MQ をデプロイする時には、この形式を使用して、デプロイメントのスケーラビリティーと可用性を強化できます。
- JSON ファイルは人間が読めるので、キュー・マネージャーの構成がシンプルになります。
- フラット・ファイル形式なので、以下のツールとの統合が可能です。
  - CCDT のヒストリーをトラッキングするためのバージョン管理ツール
  - 継続的デリバリーのためのオートメーション・ツール
- CCDT ファイルの保守のために特別なツールは必要ありません。
- ファイルのサイズが小さくなります。
- この形式には、逆方向と順方向の互換性があります。

注 :

1. JSON 標準は重複キーを有効と見なしていますが、JSON パーサーは、重複キーの最終読み取り値だけを取り込んで属性に割り当てます。 したがって、重複するチャネルを定義する場合、各チャネルは、「channel」キーに割り当てられる配列値のエレメントでなければなりません。
2. JSON CCDT のキーと属性値は、スキーマで指定されたとおりに提供さ**れなければならない**。 特に、JSON の解析では大文字と小文字が区別されることに注意。
3. JSON CCDT では、Lightweight Directory Access Protocol (LDAP) サーバー上の証明書失効リスト (CRL) の場所と Online Certificate Status Protocol (OCSP) の応答側の場所に関する情報を保管できません。

表 7. プラットフォームごとのエンコード要件

プラットフォーム	JMS クライアントのエンコード	C クライアントのエンコード
IBM i	ASCII	EBCDIC
AIX, Linux, and Windows	ASCII	ASCII
z/OS	ASCII または EBCDIC	適用外

重要 : JSON CCDT (すべての属性を含まない スパース を含む) を使用してチャネルの定義を指定すると、JSON で指定されていないものにデフォルトを使用して、すべての属性が定義された完全なチャネル定義が作成されます。

したがって、デフォルト値を必要としない属性ごとに特定の値を指定する必要があります。

手順

- JSON CCDT を作成します。

  a) 汎用テキスト・エディターを使用して、.json 拡張子を持つフラット・ファイルを作成します。

  b) CCDT を定義します。

  48 ページの『JSON CCDT の例』 および 47 ページの『JSON CCDT でサポートされているチャネル属性』 を参照してください。
- 以下で CCDT を見つけます。
  - クライアント・コンピューター上
  - 複数のクライアントが共有する場所

– サーバー上で、共有ファイルとして

51 ページの『CCDT の場所』 を参照してください。

- JSON CCDT を検証します。

  JSON 検証ツールを使用して、スキーマに照らして CCDT を検証します。

  2 つのチャネルを使用して CCDT ファイルを作成し、それが機能することを検証する方法については、スキーマに照らして IBM MQ CCDT JSON ファイルを検証する方法 を参照してください。

  CCDT スキーマは、製品パッケージとクライアント・パッケージに含まれています。

  – Linux AIX AIX and Linux システムの場合:

    製品パッケージおよびクライアント・パッケージ内の`$MQ_INSTALLATION_PATH/lib` と `/lib`。

  – Windows On Windows:

    製品パッケージおよびクライアント・パッケージ内の`%MQ_INSTALLATION_PATH%\bin` と `\bin`。

  注:

  – JSON 検証ツールはオンラインで入手できます。
  – スキーマは、「`required`」キーを使用して必須属性を定義します。
  – スキーマは、「`type`」キーを使用して属性データ・タイプを定義します。

- CCDT にアクセスします。

  以下のようにして、CCDT にアクセスできます。

  – **MQCCDTURL** 環境変数を定義することにより、ファイル、FTP、または HTTP URL からリモートでアクセスする。
  – **MQCHLLIB** 環境変数と **MQCHLTAB** 環境変数を設定することによりローカルでアクセスする。
  – クライアント構成ファイル内の CHANNELS スタンザの **ChannelDefinitionDirectory** 属性および **ChannelDefinitionFile** 属性を定義することにより、ローカルでアクセスする。

  さまざまな例については、51 ページの『CCDT の場所』 を参照してください。

- CCDT の内容を表示したり編集したりします。

  CCDT の各項目は、特定のキュー・マネージャーへのクライアント接続を表します。 テキスト・エディターで CCDT の内容を表示したり編集したりできます。

  CCDT を表示するだけなら、以下の要領で **runmqsc** コマンドを使用するという方法もあります。

  1. 環境変数で CCDT へのアクセス権を設定します (前の手順を参照)。
  2. コマンド `runmqsc -n` を実行します。 詳しくは、runmqsc を参照してください。
  3. **DISPLAY CHANNEL** コマンドを実行します。 例えば、 `DISPLAY CHANNEL(*)` を実行します。

関連概念

取り消された証明書の取り扱い

関連タスク

42 ページの『バイナリー形式の CCDT の構成』

クライアント・チャネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャネル定義と認証情報を判別します。 Multiplatforms では、キュー・マネージャーの作成時に、デフォルト設定が組み込まれたバイナリー CCDT が自動的に作成されます。 バイナリー CCDT を更新する場合は、**runmqsc** コマンドを使用します。

409 ページの『均一クラスターの構成』

均一クラスターによって、スケーリングと可用性を重視したアプリケーションを設計することができます。また均一クラスターは、均一クラスター内のどのキュー・マネージャーにも接続できます。

JSON CCDT でサポートされているチャネル属性
JSON CCDT でサポートされているクライアント接続チャネル属性のリストです。 このリストは、バイナリー CCDT でサポートされている属性のサブセットになります。

属性マッピング

以下のチャネル・オブジェクトに属性が挿入されます。

```
{ "channel": [ {$CHANNEL_1_KEY_VALUE_LIST },..., {$CHANNEL_N_KEY_VALUE_LIST } ] }
```

$CHANNEL_X_KEY_VALUE_LIST は、以下の表にまとめられている属性のコンマ区切りリストです。

基本的なユース・ケースについては、48 ページの『JSON CCDT の例』を参照してください。

JSON スキーマは /opt/mqm/lib/ccdt_schema.json に同梱されています。 各属性に有効な値を確認するには、JSON スキーマを参照してください。

JSON オブジェクト、JSON キー、JSON データ型、対応するバイナリー・チャネル属性定義を以下の表にまとめます。

重要 : 必須属性は、channel の **name** と channel の **type** です。 **portRange** を定義する場合は、*low* と *high* も必須属性になります。

JSON オブジェクト	JSON キー	JSON データ型	バイナリー属性定義
channel (配列)	名前	STRING	CHANNEL
channel (配列)	タイプ	STRING	CHLTYPE
channel.clientConnection	queueManager	STRING	QMNAME
channel.clientConnection.connection (配列)	host	STRING	CONNAME
channel.clientConnection.connection	port	INT	CONNAME
channel.compression.header (配列)	ヘッダー	STRING	COMPHDR
channel.compression.message (配列)	メッセージ	STRING	COMPMSG
channel.connectionManagement	アフィニティー (affinity)	STRING	AFFINITY
channel.connectionManagement	clientWeight	INT	CLNTWGHT
channel.connectionManagement	defaultReconnect	STRING	DEFRECON
channel.connectionManagement	disconnectInterval	INT	DISCINT
channel.connectionManagement	heartbeatInterval	INT	HBINT
channel.connectionManagement	keepAliveInterval	INT	KAINT
channel.connectionManagement	sharingConversations	INT	SHARECNV
channel.connectionManagement.localAddress (配列)	host	STRING	LOCLADDR
channel.connectionManagement.localAddress (配列)	port	INT	LOCLADDR
channel.connectionManagement.localAddress.portRange	high	INT	LOCLADDR
channel.connectionManagement.localAddress.portRange	low	INT	LOCLADDR
channel.exits.receive (配列)	名前	STRING	RCVEXIT

JSON オブジェクト	JSON キー	JSON データ型	バイナリー属性定義
channel.exits.receive (配列)	userData	STRING	RCVDATA
channel.exits.security	名前	STRING	SCYEXIT
channel.exits.security	userData	STRING	SCYDATA
channel.exits.send (配列)	名前	STRING	SENDEXIT
channel.exits.send (配列)	userData	STRING	SENDDATA
channel.general	description	STRING	DESCR
channel.general	maximumMessageLength	INT	MAXMSGL
channel.timestamps	altered	STRING	ALTDATE と ALTTIME
channel.transmissionSecurity	certificateLabel	STRING	CERTLABL
channel.transmissionSecurity	certificatePeerName	STRING	SSLPEER
channel.transmissionSecurity	cipherSpecification	STRING	SSLCIPH

注:

- `channel.connectionManagement.localAddress` は、以下のいずれかのキーの組み合わせで定義できます。
  - host と port
  - host と portRange
  - port
  - portRange
- `channel.timestamps altered` JSON キーはオプションであり、定義されていない場合、値はデフォルトで JSON CCDT ファイルの最終変更日時になります。 ただし、環境が URL から CCDT をフェッチするように構成されている場合、デフォルト値はファイルが最後にダウンロードされた時刻です。
- `channel.clientConnection.connection` には、host と port の両方のキーを組み込む必要があります。
- altered キーは、ALTDATE 属性と ALTTIME 属性の両方をカプセル化した 1 つのストリングです。
- トランスポート・タイプは TCP だけなので、以下の属性はスキーマで定義されていません。
  - **TRPTYPE**
  - **USERID**
  - **PASSWORD**
  - **MODENAME**
  - **TPNAME**

関連資料

チャネル・タイプのチャネル属性

JSON CCDT の例

このトピックで示す例に基づいて、それぞれの要件に合わせて構成できます。

汎用のテキスト・エディターを開いて、以下の例の 1 つをコピーします。

- 49 ページの『シンプルなクライアント接続の定義』
- 49 ページの『TLS を使用して 1 つのチャネルと 1 つのキュー・マネージャーを定義する』

- 49 ページの『TLS を使用しない 1 つのチャネルと 1 つのキュー・マネージャーの定義』
- 50 ページの『同じ名前の 2 つのチャネルの定義』
- 50 ページの『クライアント接続チャネルの CCDT チャネル属性定義の完全なリスト』

シンプルなクライアント接続の定義

```
{
  "channel":
  [
    {
      "general":
      {
        "description": "a channel"
      },
      "name": "channel",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "localhost",
            "port": 1414
          }
        ],
        "queueManager": "QM1"
      },
      "type": "clientConnection"
    }
  ]
}
```

TLS を使用して 1 つのチャネルと 1 つのキュー・マネージャーを定義する

```
{
  "channel": [
    {
      "name": "SSL.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "aztlan1.fyre.ibm.com",
            "port": 1419
          }
        ],
        "queueManager": "QM92TLS"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "TLS_AES_128_GCM_SHA256",
        "certificateLabel": "ibmwebspheremqadministrator",
      },
      "type": "clientConnection"
    }
  ]
}
```

TLS を使用しない 1 つのチャネルと 1 つのキュー・マネージャーの定義

```
{
  "channel": [
    {
      "name": "SYSTEM.DEF.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "aztlan1.fyre.ibm.com",
            "port": 1414
          }
        ],
        "queueManager": "QM92"
      },
      "type": "clientConnection"
```

```
    }
  ]
}
```

同じ名前の 2 つのチャネルの定義

各チャネルを 2 つの別々のキュー・マネージャーに接続します。

```
{
  "channel":
  [
    {
      "general":
      {
        "description": "First channel"
      },
      "name": "channel",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "localhost",
            "port": 1414
          }
        ],
        "queueManager": "QM1"
      },
      "type": "clientConnection"
    },
    {
      "general":
      {
        "description": "Second channel"
      },
      "name": "channel",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "localhost",
            "port": 1415
          }
        ],
        "queueManager": "QM2"
      },
      "type": "clientConnection"
    }
  ]
}
```

クライアント接続チャネルの CCDT チャネル属性定義の完全なリスト

```
{
  "channel":
  [
    {
      "compression":
      {
        "header": [ "system" ],
        "message": [ "zlibfast" ]
      },
      "connectionManagement":
      {
        "sharingConversations": 10,
        "clientWeight": 1,
        "affinity": "none",
        "defaultReconnect": "yes",
        "heartbeatInterval": 600,
        "keepAliveInterval": -1,
        "localAddress":
        [
          {
            "portRange":
            {
```

```
              "low": 2020,
              "high": 3030
            }
          }
        ]
      },
      "exits":
      {
        "receive":
        [
          {
          "name": "",
          "userData": ""
          }
        ],
        "security":
        {
          "name": "",
          "userData": ""
        },
        "send":
        [
           {
             "name": "",
             "userData": ""
           }
        ]
      },
      "general":
      {
        "description": "First channel",
        "maximumMessageLength": 4194304
      },
      "name": "the_channel",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "localhost",
            "port": 1414
          }
        ],
        "queueManager": "QM1"
      },
      "timestamps":
      {
        "altered": "2018-12-04T15:37:22.000Z"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "",
        "certificateLabel": "",
        "certificatePeerName": ""
      },
      "type": "clientConnection"
    }
  ]
}
```

関連資料

チャネル・タイプのチャネル属性

チャネル属性 (アルファベット順)

CCDT の場所

IBM MQ では、ファイル、FTP、または HTTP の URL からの CCDT の取得がサポートされます。 CCDT がサーバー上に置かれたまま、共有ファイルとしてクライアントからアクセス可能にすることができます。 あるいは、CCDT を分散させることも可能です。その場合は、各クライアント・コンピューターに CCDT をコピーするか、複数のクライアントが共有できる場所に CCDT をコピーします。

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

サーバー上で CCDT ファイルをホストする方法

IBM MQ 9.0 以降、URL でアクセス可能な中央の位置に CCDT をホストすることができるため、デプロイされた各クライアントの CCDT を個々に更新する必要はなくなりました。 IBM MQ 9.0 では、ネイティブ (C/C++、COBOL および RPG) および非管理 .NET アプリケーション用の機能が追加され、ローカル・ファイル、FTP、または HTTP リソースでも、CCDT を URL から取得できるようになりました。

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

- **CCDTUrlPtr** と **CCDTUrlOffset** (MQCNO 構造を MQCONNX MQI 呼び出しに渡す)
- **MQCCDTURL** 環境変数
- mqclient.ini の Channels スタンザの **ChannelDefinitionDirectory** 属性

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

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

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

このサポートを FTP や HTTP で使用する場合は、引き続きサーバーで CCDT ファイルをホストする必要がありますが、IBM MQ 9.0 で追加されたサポートを使用すれば、すべてのクライアント・アプリケーションが自動的にチャネル定義の変更内容を取得できるので、手動で更新を提供したり、ネットワーク・ファイル・システムを各クライアントでマウントしたりする必要はありません。 詳しくは、52 ページの『CCDT への URL アクセス』を参照してください。

クライアントで CCDT の位置を指定する方法

クライアント・システムでは、以下の方法で CCDT の位置を指定できます。

- 環境変数 **MQCHLLIB** を使用してテーブルが置かれているディレクトリーを指定し、 **MQCHLTAB** を使用してテーブルのファイル名を指定します。
- クライアント構成ファイルを使用します。 CHANNELS スタンザで、 **ChannelDefinitionDirectory** 属性を使用してテーブルが置かれているディレクトリーを指定し、 **ChannelDefinitionFile** 属性を使用してファイル名を指定します。
- 中央の位置でホストされている CCDT の URL (ファイル、FTP、HTTP) を指定します (前述の説明を参照)。

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

URL を使用して CCDT の位置を指定する場合、ネイティブ・クライアント・アプリケーションがクライアント・チャネル定義を検索する順序は、52 ページの『CCDT への URL アクセス』に記載している順序になります。

CCDT への URL アクセス

URL でアクセスできる中央ロケーションにクライアント・チャネル定義テーブル (CCDT) をホストできるので、デプロイされたクライアントごとに CCDT を個別に更新する必要がありません。

IBM MQ 9.0 以降、クライアント・チャネル定義テーブルは、次のいずれかの方法で、URL を使用して位置指定できるようになりました。

- MQCNO を使用したプログラミング
- 環境変数を使う。 キュー・マネージャへの接続に使用されるクライアント・チャネル定義は、 以下の環境変数を使用して、優先順位の高い順に配置される： '**MQSERVER** 、 '**MQCHLLIB** 、 '**MQCHLTAB** 。

> **重要 :** 環境変数オプションを使用すると、クライアントとして接続するネイティブ・プログラム、つまり C、COBOL、C++アプリケーションだけに URL を提供することができる。 Java、JMS、および管理対象 .NET アプリケーションの場合、環境変数は効果がありません。

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

- mqclient.ini ファイルの CHANNELS スタンザを使用する。

環境変数'**MQCCDTURL** を使用すると、クライアント・チャネル定義テーブルを 取得するためのファイル、ftp、または http の URL を 1 つの値として指定する ことができる。

また、'**MQCHLLIB** 環境変数で指定されたディレクトリ・パス (または'172 ページの『クライアント構成ファイルの CHANNELS スタンザ』の'**ChannelDefinitionDirectory** 属性で指定されたパス) を使って、既存のローカル・ファイル・システム・ディレクトリに加えて、 ファイル、ftp、http URL のいずれかを使って CCDT ファイルを探すこともできる (つまり、'/var/mqm)。**MQCHLLIB** 値はディレクトリのステムであり、'**MQCHLTAB** と組み合わせて完全修飾 URL を導き出すことに注意してください。

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

認証される接続

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

認証されない接続

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

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

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

1. MQCNO 内の **ClientConnOffset** および **ClientConnPtr** によって提供される MQCD。
2. MQCNO の **CCDTUrlOffset** および **CCDTUrlPtr** によって指定される URL。
3. *MQSERVER* 環境変数。
4. mqclient.ini ファイルが定義されていて、Channels スタンザに **ServerConnectionParms** 属性が含まれている場合は、そのファイルで定義されているチャネルが使用されます。 詳細については、158 ページの『IBM MQ MQI client 構成ファイル、 mqclient.ini』および 172 ページの『クライアント構成ファイルの CHANNELS スタンザ』を参照してください。
5. **MQCCDTURL** 環境変数。
6. **MQCHLLIB** および **MQCHLTAB** 環境変数。
7. 172 ページの『クライアント構成ファイルの CHANNELS スタンザ』の **ChannelDefinitionDirectory** および **ChannelDefinitionFile**。

重要 : URL を使用して、CCDT ファイルにアクセスすると、file://プロトコルを使用している場合でも、読み取り専用のファイルのコピーが常に開けます。

書き込みアクセスのために CCDT ファイルを開こうとすると (例えば、クライアントから **DEFINE CHANNEL** MQSC コマンドを使用する場合)、書き込みアクセスのためにファイルを開くことができなかったことを示すエラー・メッセージが返されます。

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

関連タスク

56 ページの『クライアント接続チャネル定義へのアクセス』
クライアント・チャネル定義テーブル (CCDT) をクライアント・アプリケーションで使用できるようにするには、CCDT をコピーまたは共有してからクライアント・コンピューターでその位置と名前を指定します。URL からクライアント・チャネル定義テーブル（CCDT）を探すこともできる。

42 ページの『バイナリー形式の CCDT の構成』
クライアント・チャネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャネル定義と認証情報を判別します。 Multiplatforms では、キュー・マネ

ージャーの作成時に、デフォルト設定が組み込まれたバイナリー CCDT が自動的に作成されます。 バイナリー CCDT を更新する場合は、**runmqsc** コマンドを使用します。

IBM MQ classes for JMS での CCDT の使用

関連資料

CCDTURL

MQCNO - 接続オプション

XMSC_WMQ_CCDTURL

Windows Active Directory 内のクライアント接続チャネル

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

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

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

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

サーバー側でのサーバー接続チャネルの定義

キュー・マネージャー用にサーバー接続チャネルの定義を作成します。

手順

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

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

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

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

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

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

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

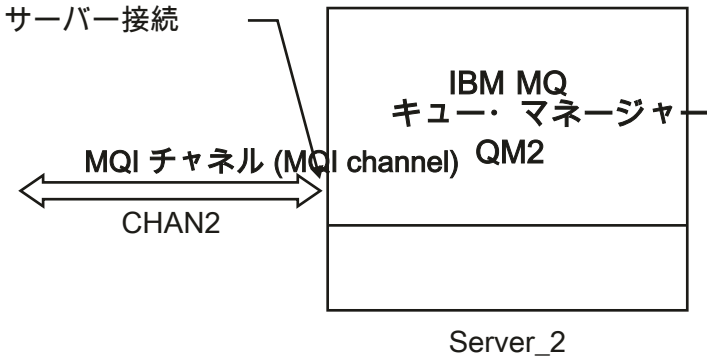


図 2. サーバー接続チャネルの定義

関連タスク

55 ページの『サーバー側でのクライアント接続チャネルの定義』
サーバー接続チャネルを定義した後、対応するクライアント接続チャネルを定義することができます。

サーバー側でのクライアント接続チャネルの定義

サーバー接続チャネルを定義した後、対応するクライアント接続チャネルを定義することができます。

始める前に

サーバー接続チャネルを定義します。 詳細については、54 ページの『サーバー側でのサーバー接続チャネルの定義』を参照してください。

手順

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

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

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

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

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

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

タスクの結果

Multi マルチプラットフォーム では、このチャネル定義は、キュー・マネージャーに関連付けられたクライアント・チャネル定義テーブル (CCDT) と呼ばれるファイルに格納されます。 クライアント・チャネル定義テーブルには、複数のクライアント接続チャネル定義を 組み込むことができます。 クライアント・チャネル定義テーブルの詳細と、クライアント接続チャネル定義を z/OS に格納する方法については、42 ページの『バイナリー形式の CCDT の構成』を参照してください。

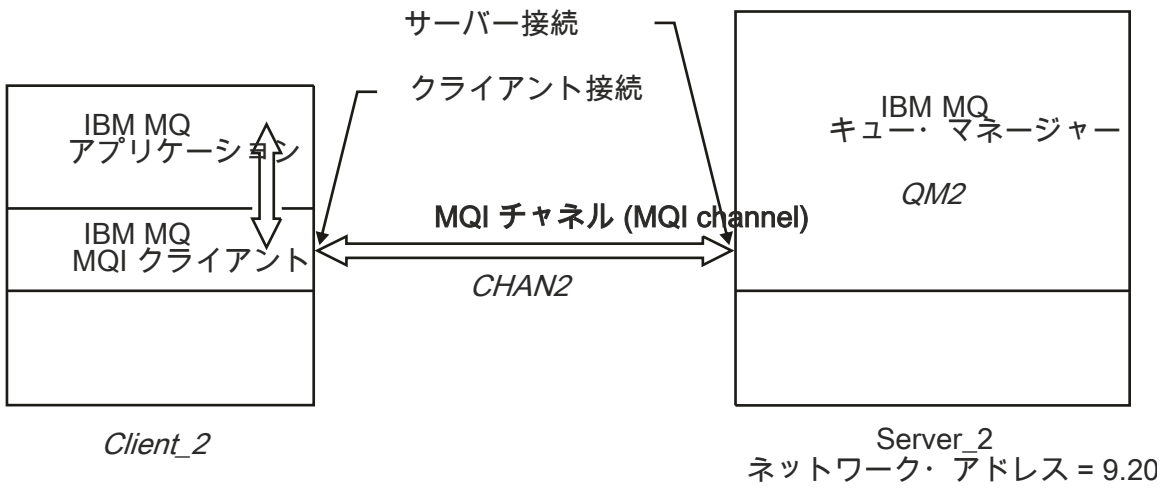


図 3. クライアント接続チャネルの定義

関連資料
DEFINE CHANNEL (新規チャネルの定義)
SET CHLAUTH (チャネル認証レコードの作成または変更)

クライアント接続チャネル定義へのアクセス

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

始める前に

必要なクライアント接続チャネルがすでに CCDT で定義されていることがこのタスクの前提になります。41 ページの『クライアント・チャネル定義テーブルの構成』 を参照してください。

このタスクについて

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

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

IBM MQ ネイティブ（C/C++、COBOL、RPG）およびアンマネージド.NET アプリケーションは、ローカルファイル、ftp、http リソースを問わず、URL から中央の場所にホストされている CCDT を引き出すことができる。

手順

1. 以下のいずれかの方法で、CCDT をクライアント・アプリケーションで使用できるようにします。
   a) オプション: クライアントのコンピューターに CCDT をコピーします。
   b) オプション: 複数のクライアントが共有する位置に CCDT をコピーします。
   c) オプション: CCDT をサーバーに残しますが、クライアントが共有できるようにします。

d) オプション: 中央の位置でホストする CCDT に対するローカル・ファイル、ftp、または http の URL を定義して、その URL からネイティブ (C/C++、 COBOL および RPG) および非管理 .NET アプリケーションが CCDT を取得できるようにします。

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

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

a) オプション: クライアント構成ファイルの CHANNELS スタンザを使用します。 詳細については、172 ページの『クライアント構成ファイルの CHANNELS スタンザ』を参照してください。

b) オプション: 環境変数 **MQCHLLIB** および **MQCHLTAB** を使用します。

例えば、次のように入力して環境変数を設定できます。

- Linux AIX AIX and Linux システムの場合:

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

- IBM i IBM i システムの場合:

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

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

c) オプション: Windows でのみ、**setmqscp** 制御コマンドを使用して、Active Directory でクライアント接続チャネル定義を公開することができます。

d) 環境変数を使用して、MQCNO を使用するか、環境変数を使用してプログラミングするか、または mqclient.ini ファイル・スタンザを使用することにより、中央でホストされる CCDT のロケーションを URL で提供します。 詳しくは、51 ページの『CCDT の場所』および52 ページの『CCDT への URL アクセス』を参照してください。

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

関連タスク

42 ページの『バイナリー形式の CCDT の構成』
クライアント・チャネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャネル定義と認証情報を判別します。 Multiplatforms では、キュー・マネージャーの作成時に、デフォルト設定が組み込まれたバイナリー CCDT が自動的に作成されます。 バイナリー CCDT を更新する場合は、**runmqsc** コマンドを使用します。

関連資料

MQI クライアント: クライアント・チャネル定義テーブル (CCDT)

ALW MQI チャネル用のチャネル出口プログラム

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

次のとおりです。

- 送信出口
- 受信出口
- セキュリティー出口

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

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

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

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

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

関連概念

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

ユーザー出口、API 出口、および IBM MQ インストール可能サービス

関連タスク

キュー・マネージャーの機能の拡張

関連資料

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

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

ALW 出口へのパス

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

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

```
ClientExitPath:
ExitsDefaultPath= string
ExitsDefaultPath64= string
```

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

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

ALW クライアント接続でのセキュリティー出口

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

60 ページの図 4 は、クライアント接続でのセキュリティー出口の使用を示しており、ここでは IBM MQ オブジェクト権限マネージャーを使用してユーザーを認証します。

MQCNO 構造体の SecurityParmsPtr または SecurityParmsOffset フィールドのいずれかがクライアントによって設定され、チャネルの両端にセキュリティー出口があります。 通常のセキュリティー・メッセージ交換が終了し、チャネルが実行可能になると、MQCSP 構造がクライアント・セキュリティー出口に渡されます。 出口は、MQCXP 構造体の SecurityParms フィールドを使用して MQCSP 構造体にアクセスできます。 出口タイプは MQXR_SEC_PARMS に設定されています。 セキュリティー出口は、MQCSP 構造内の資格情報を変更するか、変更しないままにすることができます。

その後、出口から返されたデータは、チャネルのサーバー接続側に送信されます。 MQCSP 構造体は、チャネルのサーバー接続側で再作成され、サーバー接続セキュリティー出口に渡されます。 出口は、MQCXP 構造体の SecurityParms フィールドを使用して MQCSP 構造体にアクセスできます。 セキュリティー出口はこのデータを受信して処理します。 この処理は通常、キュー・マネージャー接続を許可するために使用される、クライアント出口によって MQCSP 構造体内の資格情報に加えられた変更を元に戻すためのものです。 結果の MQCSP 構造体は、キュー・マネージャー・システム上の MQCNO 構造体の SecurityParmsPtr を使用して参照されます。

MQCXP 構造体の SecurityParms フィールドで返されるメモリー・アドレスは、MQXR_TERM までアドレス可能のままで変更されないようにする必要があります。 出口が MQXR_TERM に対して呼び出される前に、その出口がメモリーを無効にしたり、解放してシステムに戻したりすることはできません。

MQCNO 構造体の SecurityParmsPtr または SecurityParmsOffset フィールドが設定されていて、チャネルの一方の側にのみセキュリティー出口がある場合、セキュリティー出口は MQCSP 構造体を受信して処理します。 補足的な処置を実行する出口がないため、暗号化などの処置は単一のユーザー出口には不適切です。

MQCNO 構造体の SecurityParmsPtr フィールドと SecurityParmsOffset フィールドが設定されておらず、チャネルの一方または両方にセキュリティー出口がある場合は、セキュリティー出口が呼び出されます。 どちらのセキュリティー出口も、 SecurityParmsPtr フィールドによってアドレス指定される独自の MQCSP 構造体を返すことができます。 セキュリティー出口は、終了するまで再び呼び出されることはありません (MQXR_TERM の ExitReason)。 出口ルーチンは、その段階で MQCSP に使用されるメモリーを解放することができます。

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

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

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

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

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

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

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

ALW 送信または受信出口プログラムでの API 呼び出しの識別

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

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

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

表 *8. API* 呼び出しの識別

API 呼び出し	**要求用のバイト 10 の値**	**応答用のバイト 10 の値**
MQCONN 61 ページの『1』, 61 ページの『2』	X'81'	X'91'
MQDISC 61 ページの『1』	X'82'	X'92'
MQOPEN 61 ページの『3』	X'83'	X'93'

表 8. API 呼び出しの識別 (続き)

API 呼び出し	要求用のバイト 10 の値	応答用のバイト 10 の値
MQCLOSE	X'84'	X'94'
MQGET 61 ページの『4』	X'85'	X'95'
MQPUT 61 ページの『4』	X'86'	X'96'
MQPUT1 要求 61 ページの『4』	X'87'	X'97'
MQSET 要求	X'88'	X'98'
MQINQ 要求	X'89'	X'99'
MQCMIT 要求	X'8A'	X'9A'
MQBACK 要求	X'8B'	X'9B'
MQSTAT 要求	X'8D'	X'9D'
MQSUB 要求	X'8E'	X'9E'
MQSUBRQ 要求	X'8F'	X'9F'
xa_start 要求	X'A1'	X'B1'
xa_end 要求	X'A2'	X'B2'
xa_open 要求	X'A3'	X'B3'
xa_close 要求	X'A4'	X'B4'
xa_prepare 要求	X'A5'	X'B5'
xa_commit 要求	X'A6'	X'B6'
xa_rollback 要求	X'A7'	X'B7'
xa_forget 要求	X'A8'	X'B8'
xa_recover 要求	X'A9'	X'B9'
xa_complete 要求	X'AA'	X'BA'

注 :

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

z/OS キュー共有グループへのクライアントの接続

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

このタスクについて

キュー共有グループは、同じ共有キューのセットにアクセスできるキュー・マネージャーのセットにより構成されます。 共用キューについて詳しくは、共用キューおよびキュー共用グループを参照してください。

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

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

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

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

手順

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

1. 次の例に示されているように SVRCONN チャネルを定義します。

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

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

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

タスクの結果

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

IBM MQ 環境変数の使用

コマンドを使用して、現在の設定を表示したり、IBM MQ 環境変数の値をリセットしたりすることができます。

このタスクについて

以下の方法で環境変数を使用できます。

- システム・プロファイル内の変数を設定して永続的な変更を行う
- コマンド・ラインからコマンドを発行し、このセッションのみの変更を行う
- 実行中のアプリケーションに応じて 1 つ以上の変数に特定の値を指定するために、そのアプリケーションが使用するコマンド・スクリプト・ファイルにコマンドを追加する

環境変数ごとに、コマンドを使用して現在の設定を表示したり、環境変数の値をリセットしたりすることができます。 これらのコマンドは、特に明記されていない限り、サポートされているすべてのプラットフ

ォームで使用可能です。 コマンドの形式は、ご使用のプラットフォームによって異なります。 以下に例を示します。

- Linux AIX On AIX and Linux:

```
export [environment variable]=value
```

- Windows On Windows:

```
Set [environment variable]=value
```

- IBM i On IBM i:

```
ADDENVVAR ENVVAR(environment variable) VALUE(xx)
```

- MQ Appliance IBM MQ Appliance については、 IBM MQ Appliance 資料の「IBM MQ Appliance での環境変数の構成」を参照してください。

該当する場合、 IBM MQ は、設定されていない環境変数に対してデフォルト値を使用します。

注 : z/OS IBM MQ for z/OS は、いずれの IBM MQ 環境変数もサポートしていません。 このプラットフォームをサーバーとして使用している場合は、z/OS でのクライアント・チャネル定義テーブルの生成方法について、クライアント・チャネル定義テーブル を参照してください。 クライアント・プラットフォームでは、引き続き IBM MQ 環境変数を使用できます。

手順

- Windows
  Windows では、環境変数ごとに次のコマンドを使用して、現在の設定を表示するか、変数の値をリセットします。
  - 環境変数の値を削除するには、以下のコマンドを使用します。

    ```
    SET MQSERVER=
    ```

  - 環境変数の現行設定を表示するには、以下のコマンドを使用します。

    ```
    SET MQSERVER
    ```

  - セッションのすべての環境変数を表示するには、以下のコマンドを使用します。

    ```
    set
    ```

- Linux AIX
  AIX and Linux では、環境変数ごとに次のコマンドを使用して、現在の設定を表示するか、変数の値をリセットします。
  - 環境変数の値を削除するには、以下のコマンドを使用します。

    ```
    unset MQSERVER
    ```

  - 環境変数の現行設定を表示するには、以下のコマンドを使用します。

    ```
    echo $MQSERVER
    ```

  - セッションのすべての環境変数を表示するには、以下のコマンドを使用します。

    ```
    set
    ```

関連タスク
IBM MQ classes for JMS/Jakarta Messaging の環境変数の設定
IBM MQ classes for Java に関連する環境変数

service.env ファイルでの追加の環境変数の定義
84 ページの『Multiplatforms での .ini ファイルの IBM MQ 構成情報の変更』
構成 (.ini) ファイル内の情報を編集することにより、IBM MQ または個々のキュー・マネージャーの動作を、ご使用のインストール済み環境のニーズに合わせて変更できます。 IBM MQ MQI clients の構成オプションを変更することもできます。

関連資料
MFT プロパティーでの環境変数の使用

環境変数の説明

お客様が使用することを意図したサーバー環境変数およびクライアント環境変数の説明。

使用例

- Linux AIX AIX and Linux システムでは、`export [environment variable]=value` という形式を使用します。
- Windows Windows システムでは、 `Set [environment variable]=value` という形式を使用します。
- IBM i IBM i システムでは、`ADDENVVAR ENVVAR(environment variable) VALUE(xx)` という形式を使用します。
- MQ Appliance IBM MQ Appliance については、 IBM MQ Appliance 資料の「IBM MQ Appliance での環境変数の構成」を参照してください。

AMQ_ALLOWED_CIPHERS

Multi

IBM MQ 9.2.0 以降では、 **AMQ_ALLOWED_CIPHERS** 環境変数を使用して、Multiplatforms の IBM MQ チャネルで使用可能な CipherSpecs のカスタム・リストを指定できます。 環境変数は、 `.ini` ファイルの **AllowedCipherSpecs** SSL スタンザ属性と同じ値を取ります。

- 単一の CipherSpec 名
- 再有効化する IBM MQ CipherSpec 名のコンマ区切りリスト。または
- すべての CipherSpec を表す特殊値 ALL (推奨されません)。

注 : ALL CipherSpecs を有効化すると、SSL 3.0 プロトコルおよび TLS 1.0 プロトコルと多数の脆弱な暗号アルゴリズムが有効化されるため、これは推奨されません。

詳しくは、 TLS ハンドシェークでの CipherSpec の順序の「Multiplatforms での有効な CipherSpecs のカスタム・リストの提供」を参照してください。

amq_bad_comms_data_fdcs

AMQ_BAD_COMMS_DATA_FDCS 環境変数は、任意の値に設定すると有効になります。

IBM MQ が TCP/IP を介してホストから受信するデータの形式が正しくない場合 (例えば、ネットワーク・クライアントが IBM MQ リスナー・ポートに接続していて、サポートされないアプリケーション・プロトコルと通信しようとした場合)、キュー・マネージャーは AMQ9207E エラー・メッセージをキュー・マネージャーのエラー・ログに書き込みます。 IBM MQ リスナーは、キュー・マネージャー・メッセージ・チャネル・エージェント (MCA) および MQI クライアント・アプリケーション、 JMS および XMS クライアント・アプリケーションからの TCP/IP 接続をサポートします。

注 : IBM MQ リスナーは、AMQP および MQTT クライアントによって使用されるアプリケーション・プロトコルをサポートしません。これらのクライアントは、代わりに、該当する AMQP チャネルまたは MQXR テレメトリー・サービスで構成されたネットワーク・ポートに接続する必要があります。

IBM MQ が受信した無効データを含む障害データ・キャプチャー機能 (FDC) レコードも書き込まれる可能性があります。 ただし、リモート側との会話の開始時で、 HTTP ウェブブラウザからの GET リクエストのような単純な既知のフォーマットである場合は、FFST ファイルは生成されません。 これをオーバーライドして、単純な既知の形式を含むすべての不良データに対して FFST ファイルが書き込まれるようにするには、**AMQ_BAD_COMMS_DATA_FDCS** 環境変数を任意の値 (例えば、 TRUE) に設定して、キュー・マネージャーを再始動します。

AMQ_CONVEBCDICNEWLINE

Multi

AMQ_CONVEBCDICNEWLINE 環境変数を使用して、 IBM MQ が EBCDIC NL 文字を ASCII フォーマットに変換する方法を指定できます。 環境変数は、 `mqs.ini` の **ConvEBCDICNewline** 属性 (`NL_TO_LF`、 `TABLE`、または `ISO`) と同じ値を取ります (mqs.ini ファイルのすべてのキュー・マネージャー・スタンザを参照)。 例えば、`mqs.ini` ファイルを使用できない状況でも、**ConvEBCDICNewline** スタンザ属性の代わりに **AMQ_CONVEBCDICNEWLINE** 環境変数を使用すれば、クライアント・サイドで **ConvEBCDICNewline** 機能を利用できます。 スタンザ属性と環境変数の両方が設定されている場合は、スタンザ属性のほうが優先されます。

詳しくは、 コード化文字セット間のデータ変換 を参照してください。

AMQ_DIAGNOSTIC_MSG_SEVERITY

IBM MQ プロセスの **AMQ_DIAGNOSTIC_MSG_SEVERITY** 環境変数が 1 に設定されている場合、IBM MQ プロセスがメッセージをエラー・ログまたはコンソールに書き込むときに、メッセージ重大度が単一の大文字の英字としてメッセージ番号に付加されます。

AMQ_DIAGNOSTIC_MSG_SEVERITY が有効にする動作は、デフォルトで設定されています。 この動作をオフにするには、環境変数を `0` に設定します。

詳しくは、『エラー・ログの使用』を参照してください。

amq_disable_client_ams

旧バージョンの製品のキュー・マネージャーに接続しようとしたときに 2085 (`MQRC_UNKNOWN_OBJECT_NAME`) エラーが報告され、以下のいずれかのクライアントを使用している場合は、 **AMQ_DISABLE_CLIENT_AMS** 環境変数を使用してクライアントで IBM MQ Advanced Message Security (AMS) を無効にすることができます。

- Java runtime environment (JRE) (IBM Java runtime environment (JRE) 以外)
- IBM MQ IBM MQ classes for JMS または IBM MQ classes for Java クライアント。

注 : C クライアントに対して **AMQ_DISABLE_CLIENT_AMS** 環境変数を使用することはできません。 代わりに、 **MQS_DISABLE_ALL_INTERCEPT** 環境変数を使用する必要があります。

詳しくは、 クライアントでの Advanced Message Security の無効化を参照してください。

AMQ_DMPMQCFG_QSGDISP_DEFAULT

dmpmqcfg コマンドで使用されるキュー・マネージャーの属性指定に関する照会では、デフォルトで QSGDISP (QMGR) 定義のみが照会されます。 **AMQ_DMPMQCFG_QSGDISP_DEFAULT** 環境変数を使用して、追加の定義を照会できます。この環境変数は、以下のいずれかの値に設定できます。

LIVE
QSGDISP(QMGR) または QSGDISP(COPY) で定義されたオブジェクトのみを含みます。

ALL
QSGDISP(QMGR) および QSGDISP(COPY) で定義されたオブジェクトを含みます。 キュー・マネージャーがキュー共有グループのメンバーである場合、QSGDISP(GROUP) および QSGDISP(SHARED) も含まれます。

COPY
QSGDISP(COPY) で定義されたオブジェクトのみを含みます。

GROUP
QSGDISP(GROUP) で定義されたオブジェクトのみを含みます。ターゲット・キュー・マネージャーがキュー共有グループのメンバーである必要があります。

QMGR
QSGDISP(QMGR) で定義されたオブジェクトのみを含みます。 これは、この環境変数を使用している場合、**dmpmqcfg** の既存の動作と一致させるためのデフォルトの動作です。

PRIVATE
QSGDISP(QMGR) または QSGDISP(COPY) で定義されたオブジェクトのみを含みます。

SHARED
QSGDISP(SHARED) で定義されたオブジェクトのみを含みます。

AMQ_IODELAY、AMQ_IODELAY_INMS および AMQ_IODELAY_FFST

V 9.3.4 Multi

IBM MQ は、ログの読み取りと書き込み、または入出力操作に予想より長い時間がかかることを検出します。 これは、オペレーティング・システムまたはストレージ・システムの問題が原因である可能性があり、キュー・マネージャーのパフォーマンスに影響を与える可能性があります。 IBM MQ 9.3.4 以降、**AMQ_IODELAY** 環境変数を使用して、キュー・マネージャー・ログおよびストレージ・ファイル・システムの入出力が遅い場合の診断とタイミングを微調整できます。 キュー・マネージャーのエラー・ログに AMQ6729W「ログ入出力操作がしきい値を超えました (Log I/O operation exceeded threshold)」というメッセージが表示された場合は、原因を調査し、それに応じて調整してください。以下の例に示すように、変数を使用します。

AMQ 入出力遅延
しきい値時間 (秒)。デフォルトは 1 秒です。 入出力操作にかかる時間がこのしきい値を超えると、エラー・メッセージ AMQ6729W が IBM MQ ログ・ファイルに報告されます。 遅延が続く場合、警告メッセージは最大で 10 秒ごとに繰り返されます。 これを増やしてエラーを抑止するか、特定のパフォーマンス問題を調査するために減らすことができます。 例えば、

```
export AMQ_IODELAY=200000
```

AMQ 入出力遅延 (INMS)
時間測定を秒ではなくマイクロ秒に変更してください。 これは、キュー・マネージャー・ログにメッセージ AMQ6729 が記録される前に下限しきい値を設定するために使用します。

```
export AMQ_IODELAY_INMS=YES
```

AMQ 入出力遅延 (FFST)
エラー・ログ内の警告メッセージに加えて、しきい値を超えるたびに診断情報を含む FFST ファイルが生成されます。

```
export AMQ_IODELAY_FFST=YES
```

この例のようにキュー・マネージャーを開始すると、入出力操作に 200000 マイクロ秒 (0.2s) より長い時間がかかる場合に、FDC ファイルまたは FFST ファイルが書き込まれます。これは比較的大きなしきい値です。

詳しくは、 キュー・マネージャーのヘルス・チェックの動作を参照してください。

AMQ_LDAP_TRACE

AMQ_LDAP_TRACE 環境変数がヌル以外の値に設定されている場合、キュー・マネージャーを停止または開始することなく、LDAP クライアント・トレースのオン/オフを切り替えることができます。

詳しくは、LDAP クライアント・ライブラリー・コードの動的トレースの有効化を参照してください。

AMQ_LICENSING_METRIC

Multi

AMQ_LICENSING_METRIC=VPCMonthlyPeak 環境変数を設定すると、キュー・マネージャーは、時間単位のコンテナー・ベースのライセンスに関連するデータをアップロードするというデフォルトの動作ではなく、月単位の VPC ライセンス・タイプに関連するデータをアップロードします。

IBM Cloud® Private 計量サービスで使用するための IBM MQ の構成について詳しくは、 IBM Cloud Private 資料の IBM Cloud Private 計量サービス を参照してください。

AMQ_MQS_INI_LOCATION

Linux AIX

AIX and Linux システムでは、**AMQ_MQS_INI_LOCATION** 環境変数に mqs.ini ファイルの場所を設定することで、 mqs.ini ファイルに使用される場所を変更できます。 この環境変数は、システム・レベルで設定する必要があります。

ディレクトリーの場所など、 mqs.ini ファイルについて詳しくは、 IBM MQ 構成ファイル mqs.ini を参照してください。

AMQ_NO_BAD_COMMES_DATA_FDCS

AMQ_NO_BAD_COMMS_DATA_FDCS 環境変数は、任意の値に設定すると有効になります。

IBM MQ 以外のクライアントを IBM MQ TCP/IP リスナーに接続しようとしたときに IBM MQ が初期データ伝送を認識しない場合、キュー・マネージャーは AMQ9207E エラー・メッセージをキュー・マネージャーのエラー・ログに書き込みます。 障害データ・キャプチャー機能 (FDC) レコードも書き込まれます。 **AMQ_NO_BAD_COMMS_DATA_FDCS** 環境変数を使用して、これらの診断ファイルの生成を抑止することができます。 **AMQ_NO_BAD_COMMS_DATA_FDCS** が任意の値 (例えば、 TRUE) に設定されている場合、これは、初期通信フローで AMQ9207E エラー・メッセージを報告するときに FFST を生成しないように IBM MQ に指示します。 有効にするには、キュー・マネージャーおよびリスナー・プロセスを開始する前に環境変数を設定する必要があります。

クライアントが有効な IBM MQ プロトコル・フローをキュー・マネージャーに送信してから、無効なデータを送信する場合は、引き続き FDC が生成されます。これは、さらなる調査を必要とするクライアントの問題を示しています。

注 : IBM MQ 9.2.0 以降、初期通信フローに関する AMQ9207E エラー・メッセージを報告する際の FFST のキャプチャーは、デフォルトでは抑止されます。

AMQ_NO_IPV6

AMQ_NO_IPV6 環境変数は、任意の値に設定すると有効になります。 この環境変数を設定すると、接続の試行中に IPv6 を使用できなくなります。

AMQ_REVERSE_COMMIT_ORDER

AMQ_REVERSE_COMMIT_ORDER 環境変数は、XA トランザクションで、対応するデータベース更新が完了した後に IBM MQ キュー・マネージャーの変更がコミットされるように、キュー・マネージャーを構成します。 キューからメッセージを読み取るアプリケーションには、対応するデータベース更新が完了した後にのみメッセージが表示されます。

注 : 分離レベルで説明されているシナリオを読んで理解しない限り、**AMQ_REVERSE_COMMIT_ORDER** を設定しないでください。

AMQ_SSL_ALLOW_DEFAULT_CERT

AMQ_SSL_ALLOW_DEFAULT_CERT 環境変数が設定されていない場合、アプリケーションは、証明書に ibmwebspheremq*userid* のラベル名が含まれている場合にのみ、クライアント鍵ストア内の個人証明書

を使用してキュー・マネージャーに接続できます。 **AMQ_SSL_ALLOW_DEFAULT_CERT** 環境変数が設定されているとき、証明書に ibmwebspheremq*userid* のラベル名は必要ありません。 つまり、キュー・マネージャーへの接続に使用される証明書をデフォルトの証明書にすることができます。ただし、デフォルトの証明書が鍵リポジトリーに存在し、鍵リポジトリーに接頭部 ibmwebspheremq*userid* を持つ個人証明書が含まれていない場合に限ります。

値 1 は、デフォルトの証明書の使用を有効にします。

アプリケーションは、 **AMQ_SSL_ALLOW_DEFAULT_CERT** 環境変数を使用する代わりに、 mqclient.ini ファイル内の SSL スタンザの **CertificateLabel** 設定を使用できます。 詳しくは、デジタル証明書ラベルの要件に関する説明およびクライアント構成ファイルの SSL スタンザを参照してください。

AMQ_SSL_LDAP_SERVER_VERSION

CRL サーバーで特定のバージョンの LDAP プロトコルを使用する必要がある場合、 **AMQ_SSL_LDAP_SERVER_VERSION** 環境変数を使用して、LDAP v2 または LDAP v3 のいずれかが IBM MQ 暗号コンポーネントによって使用されるようにすることができます。

環境変数を、キュー・マネージャーまたはチャネルの開始に使用される環境内の適切な値に設定します。

- LDAP v2 が使用されるように要求するには、AMQ_SSL_LDAP_SERVER_VERSION=2 と設定します。
- LDAP v3 が使用されるように要求するには、AMQ_SSL_LDAP_SERVER_VERSION=3 と設定します。

この環境変数は、ユーザー認証またはユーザー許可のために IBM MQ キュー・マネージャーによって確立された LDAP 接続には影響しません。

AMQ_USE_ZLIBNX

V 9.3.0 AIX

AIX では、ZLIBFAST または ZLIBHIGH 技法を使用する場合に、 **AMQ_USE_ZLIBNX** 環境変数を使用して、メッセージ・チャネル・エージェント (MCA) がハードウェア加速 zlibNX ライブラリーを使用してメッセージ・データの圧縮および圧縮解除を行えるようにすることができます。

ヒント: 2 KB を超えるサイズの高度に圧縮可能なメッセージは、CPU 使用量を削減することにより、zlibNX ライブラリーの使用を選択することで最も効果が得られる可能性があります。

zlibNX ライブラリーは、 IBM AIX 7.2 (Technology Level 4 Expansion Pack 適用済み) 以降で使用可能です。環境変数が設定されていても、zlibNX ライブラリー (/usr/opt/zlibNX/lib/libz.a) がインストールされていなければ、メッセージング・チャネル・エージェントは IBM MQ for AIX のインストール済み環境に用意されている標準の zlib ライブラリーを使用することになります。

HOME

Linux IBM i AIX

AIX、 Linux 、および IBM i では、 **HOME** 環境変数は、 mqclient.ini ファイルを検索するディレクトリーの名前を指定します。 このファイルには、 IBM MQ MQI clients で使用される構成情報が含まれています。

詳しくは、 IBM MQ MQI クライアント構成ファイル mqclient.ini および クライアント構成ファイルの場所を参照してください。

HOMEDRIVE と HOMEPATH

Windows

使用するには、**HOMEDRIVE** 環境変数と **HOMEPATH** 環境変数の両方を設定する必要があります。 これらは、mqclient.ini ファイルを検索するディレクトリーの名前を指定するために Windows システムで使用されます。 このファイルには、 IBM MQ MQI clients によって使用される構成情報が含まれています。

詳しくは、 IBM MQ MQI クライアント構成ファイル mqclient.ini および クライアント構成ファイルの場所を参照してください。

LDAP_BASEDN

LDAP_BASEDN は、LDAP サンプル・プログラムを実行するために必要な環境変数です。 ディレクトリー検索の基本識別名を指定します。

LDAP_HOST

LDAP_HOST は、LDAP サンプル・プログラムを実行するためのオプションの環境変数です。 LDAP サーバーが稼働しているホストの名前を指定します。指定しなければ、デフォルトでローカル・ホストになります。

LDAP_VERSION

LDAP_VERSION は、LDAP サンプル・プログラムを実行するためのオプションの環境変数です。 使用する LDAP プロトコルのバージョンを指定します。2 または 3 のいずれかを指定できます。 ほとんどの LDAP サーバーは、バージョン 3 のプロトコルをサポートするようになりましたが、すべての LDAP サーバーは、旧バージョン 2 もサポートしています。 このサンプルは、どちらのバージョンのプロトコルでも同様に機能します。指定しなければ、デフォルトでバージョン 2 になります。

MQ_CHANNEL_SUPPRESS_INTERVAL

MQ_CHANNEL_SUPPRESS_INTERVAL 環境変数は、 **MQ_CHANNEL_SUPPRESS_MSGS** で定義されたメッセージがエラー・ログに書き込まれないように抑止される時間間隔を秒単位で指定します。また、指定された時間間隔内にメッセージが発生できる回数も指定します。この回数を超えると、そのメッセージは抑止されます。 デフォルト値は 60,5 です。これは、指定されたメッセージの最初の 5 回の出現の後、60 秒の間隔でそれ以降のすべての出現が抑制されることを意味します。 詳しくは、 Suppressing channel error messages from error logs on Multiplatforms (Multiplatforms で エラー・ログからチャネルのエラー・メッセージを抑制する) を参照してください。

MQ_CHANNEL_SUPPRESS_INTERVAL 環境変数は、 `qm.ini` ファイルの SuppressInterval に相当します。

MQ_CHANNEL_SUPPRESS_MSGS

MQ_CHANNEL_SUPPRESS_MSGS 環境変数は、エラー・ログ内のチャネル・エラー・メッセージを抑止します。 抑制されるメッセージのリストを指定できます。 **MQ_CHANNEL_SUPPRESS_MSGS** は、抑止される前に各メッセージが表示される回数、およびメッセージが抑止される時間の長さを指定する **MQ_CHANNEL_SUPPRESS_INTERVAL** と一緒に使用されます。 詳しくは、 Suppressing channel error messages from error logs on Multiplatforms (Multiplatforms で エラー・ログからチャネルのエラー・メッセージを抑制する) を参照してください。

MQ_CHANNEL_SUPPRESS_MSGS 環境変数は、 `qm.ini` ファイル内の SuppressMessage と同等ですが、環境変数を使用してチャネル・メッセージを抑止できる点が異なります。一方、 `qm.ini` メソッドには制限リストがあります。

MQ_CONNECT_TYPE

Multi

Multiplatforms では、 **MQ_CONNECT_TYPE** 環境変数を、MQCONNX 呼び出しで使用される MQCNO 構造の Options フィールドで指定されたバインディングのタイプと組み合わせて使用できます。
MQ_CONNECT_TYPE は、STANDARD バインディングに対してのみ有効です。 その他のバインディングの場合、 **MQ_CONNECT_TYPE** は無視されます。

詳しくは、 MQ_CONNECT_TYPE での MQCONNX 呼び出しオプションの使用を参照してください。

MQ_CROSS_QUEUE_ORDER_ALL

MQ_CROSS_QUEUE_ORDER_ALL 環境変数をゼロ以外の値に設定すると、メッセージ書き込み順序は作業単位内で維持されます。 つまり、作業単位 (UoW) のメッセージが複数のキュー (例えば Q1 と Q2) に書き込

まれる場合、MQCMIT が発行されると、メッセージはキューに書き込まれたときと同じ順序で送信され使用可能になります。

複数キュー・マネージャー環境では、各キュー・マネージャーを開始する前に、送信側と受信側の両方に **MQ_CROSS_QUEUE_ORDER_ALL** が存在し、空でない値を持っている必要があります。

MQ_EPHEMERAL_PREFIX

MQ_EPHEMERAL_PREFIX 環境変数は、キュー・マネージャーの実行中に一時キュー・マネージャー・データが保持される、キュー・マネージャーの一時ディレクトリーへのパスを指定します。

`mqs.ini` ファイルの AllQueueManagers スタンザの'**DefaultEphemeralPrefix** 属性内の'**EphemeralPrefix** 属性を変更してエフェメラル接頭辞を変更する代わりに、環境変数'**MQ_EPHEMERAL_PREFIX** を使用して、'**crtmqm** コマンドの'**EphemeralPrefix** を上書きすることができる。 詳しくは、構成可能な一時ディレクトリーを参照してください。

MQ_FILE_PATH

Windows

MQ_FILE_PATH 環境変数は、 Windows プラットフォームへのランタイム・パッケージのインストール時に構成されます。 この環境変数には、Windows レジストリーの以下のキーと同じデータが入ります。

```
HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName\FilePath
```

詳しくは、 setmqenv (IBM MQ 環境の設定) および crtmqenv (IBM MQ 環境の作成)を参照してください。

MQ_JAVA_DATA_PATH

MQ_JAVA_DATA_PATH 環境変数は、 IBM MQ classes for JMS および IBM MQ classes for Jakarta Messaging と IBM MQ classes for Java のログおよびトレース出力のディレクトリーを指定します。 これは、 IBM MQ classes for JMS 、 IBM MQ classes for Jakarta Messaging 、および IBM MQ classes for Java で提供されるスクリプトによって使用されます。

詳しくは、JMS/Jakarta Messaging 用の IBM MQ クラスの環境変数の設定 および IBM MQ classes for Java に関連する環境変数を参照してください。

MQ_JAVA_INSTALL_PATH

MQ_JAVA_INSTALL_PATH 環境変数は、 IBM MQ classes for JMS および IBM MQ classes for Jakarta Messaging (What is installed for IBM MQ classes for JMS を参照)、および IBM MQ classes for Java (IBM MQ classes for Java インストール・ディレクトリー を参照) がインストールされているディレクトリーを指定します。

詳しくは、JMS/Jakarta Messaging 用の IBM MQ クラスの環境変数の設定 および IBM MQ classes for Java に関連する環境変数を参照してください。

MQ_JAVA_LIB_PATH

MQ_JAVA_LIB_PATH 環境変数は、 IBM MQ classes for JMS と IBM MQ classes for Jakarta Messaging、および IBM MQ classes for Java ライブラリーが保管されるディレクトリーを指定します。 IBM MQ classes for JMS および IBM MQ classes for Jakarta Messaging または IBM MQ classes for Java に用意されている一部のスクリプト (`IVTRun` など) では、この環境変数を使用します。

詳しくは、JMS/Jakarta Messaging 用の IBM MQ クラスの環境変数の設定 および IBM MQ classes for Java に関連する環境変数を参照してください。

MQ_OVERRIDE データ・パス

MQ_OVERRIDE_DATA_PATH 環境変数を使用して、 IBM MQ データ・パスのデフォルト・ディレクトリーを変更できます。

MQ_SET_NODELAYACK

AIX

MQ_SET_NODELAYACK 環境変数は、 AIX 上の TCP 遅延確認応答をオフに切り替えます。

この環境変数を設定すると、オペレーティング・システムの setsockopt 呼び出しに TCP_NODELAYACK オプションを指定して呼び出すことにより、TCP 遅延確認応答がオフに切り替わります。 AIX のみがこの機能をサポートするため、 **MQ_SET_NODELAYACK** 環境変数は AIX にのみ影響します。

MQ ユーザー名 (MQ_USER_NAME)

Linux

ユーザーにパスワードファイルのエントリがない場合、またはユーザー名がユーザー識別子と一致する場合、 **MQ_USER_NAME** 環境変数を使用して、 IBM MQ の非インストールイメージを Linux で選択したユーザー名にすることができます。 これは、例えば、 OpenShift のパブリッシュ/サブスクライブ階層を使用するために必要です。

MQ_USER_NAME の値は、システム上の既存のユーザーと一致してはならず、12 バイト以下でなければなりません。

MQAPI_TRACE_LOGFILE

サンプル API 出口プログラムは、 **MQAPI_TRACE_LOGFILE** 環境変数で定義された接頭部を持つユーザー指定ファイルに MQI トレースを生成します。

詳しくは、 API 出口サンプル・プログラムを参照してください。

MQAPPLNAME

ALW

アプリケーション名がまだ選択されていない場合は、キュー・マネージャーへの接続を識別するために使用する名前として **MQAPPLNAME** 環境変数を使用できます。 最初の 28 文字だけが使用されます。すべてをブランクにしたりすべてを NULL にしたりすることはできません。

詳しくは、 サポートされるプログラミング言語でのアプリケーション名の使用を参照してください。

MQCCSID

MQCCSID 環境変数は、使用されるコード化文字セット番号を指定し、サーバーの構成に使用された CCSID 値をオーバーライドします。 **MQCCSID** を使用して、アプリケーションのネイティブ CCSID をオーバーライドし、使用するコード化文字セット番号を指定することができます。例えば、ネイティブ CCSID がサポートされていない CCSID である場合や、必要な CCSID ではない場合などです。

MQCCSID を設定するには、以下のいずれかのコマンドを使用します。

- Linux AIX On AIX and Linux:

```
export MQCCSID=number
```

- Windows On Windows:

```
SET MQCCSID=number
```

- IBM i On IBM i:

```
ADDENVVAR ENVVAR(MQCCSID) VALUE(number)
```

詳しくは、 クライアントまたはサーバーの CCSID の選択を参照してください。

MQCCDTURL

MQCCDTURL 環境変数は、 **MQCHLLIB** 環境変数と **MQCHLTAB** 環境変数の組み合わせを設定する機能と同等の機能を提供します。 これにより、ファイル、FTP、または http URL を単一の値として提供し、クライアントとして接続するネイティブプログラム（C、COBOL、C++アプリケーションなど）用のクライアントチャネル定義テーブルを取得することができます。

注 : URL に環境変数を使用しても、 Java 、 JMS 、または管理された .NET アプリケーションには影響しません。

IBM MQ では、ファイル、FTP、または HTTP の URL からの CCDT の取得がサポートされます。 しかし、「**MQCCDTURL**」は FTP 値を受け付けない。 URL の値、または「file://」プロトコルを使用して指定されたファイルのみを受け入れます。

例えば、'**MQCHLLIB** と'**MQCHLTAB** 代わりに'**MQCCDTURL** 使い、'AIX と'Linux にローカルファイルを置く場合、このコマンドを実行する：

```
export MQCCDTURL=file:///var/mqm/qmgrs/QMGR/@ipcc/MYCHL.TAB
```

これは、下記と同じです。

```
export MQCHLLIB=/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLTAB=MYCHL.TAB
```

Windows の以下の例に示すように、JSON ファイルを指定することもできます。

```
set MQCCDTURL=file:/c:/mq-channels/CCDT-QMGR1.json
```

これは、下記と同じです。

```
set MQCHLLIB=C:\mq-channels
set MQCHLTAB=CCDT-QMGR1.json
```

詳細については 、 URL の CCDT へのアクセスをご覧ください。

MQCERTLABL

MQCERTLABL 環境変数は、TLS ハンドシェーク中に送信される個人証明書を見つけるために IBM MQ が使用するチャネル定義の証明書ラベルを定義します。

詳しくは、 デジタル証明書ラベル、要件の理解を参照してください。

MQCERTVPOL

MQCERTVPOL 環境変数は、使用する証明書妥当性検査ポリシーのタイプを指定します。 この環境変数は、クライアント構成ファイルの SSL スタンザの **CertificateValPolicy** 属性をオーバーライドします。

MQCERTVPOL は、以下の 2 つの値のいずれかに設定できます。

ANY
基礎となるセキュア・ソケット・ライブラリーによってサポートされているいずれかの証明書妥当性検査ポリシーを使用します。 この設定はデフォルト設定です。

RFC5280
RFC 5280 標準に準拠する証明書妥当性検査のみを使用します。

MQCERTVPOL を設定するには、以下のいずれかのコマンドを使用します。

- Linux AIX AIX and Linux システムの場合:

```
export MQCERTVPOL= value
```

- Windows Windows システムの場合:

```
SET MQCERTVPOL= value
```

- IBM i IBM i システムの場合:

```
ADDENVVAR ENVVAR(MQCERTVPOL) VALUE(value)
```

詳しくは、 IBM MQ での証明書妥当性検査ポリシー および IBM MQ を参照してください。

MQCHLLIB

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

MQCHLLIB を設定するには、以下のいずれかのコマンドを使用します。

- Windows On Windows:

```
SET MQCHLLIB=pathname
```

以下に例を示します。

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

- Linux AIX AIX and Linux システムの場合:

```
export MQCHLLIB=pathname
```

- IBM i IBM i の場合:

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE(pathname)
```

MQCHLLIB を設定しない場合、クライアントのパスは次のようにデフォルト指定されます。

- Linux AIX AIX and Linux 上: /var/mqm/
- Windows Windows 上: *MQ_INSTALLATION_PATH*
- IBM i IBM i の場合、/QIBM/UserData/mqm/

crtmqm コマンドおよび **strmqm** コマンドの場合、パスはデフォルトで以下の 2 つのパス・セットのいずれかになります。 *datapath* が設定される場合、パスはデフォルトで 1 つ目のセットのいずれかになります。 *datapath* が設定されない場合、パスはデフォルトで 2 つ目のセットのいずれかになります。

- Linux AIX AIX and Linux 上: *datapath*/@ipcc
- Windows Windows 上: *datapath*\@ipcc
- IBM i IBM i 上: *datapath*/&ipcc

または:

- Linux AIX AIX and Linux 上: /prefix/qmgrs/*qmgrname*/@ipcc
- Windows Windows 上: *MQ_INSTALLATION_PATH*\data\qmgrs*qmgrname*\@ipcc

- IBM i IBM i 上: /*prefix*/qmgrs/*qmgrname*/&ipcc

ここで、

- *MQ_INSTALLATION_PATH* は、IBM MQ がインストールされている上位ディレクトリーを表します。
- *datapath* は、キュー・マネージャー・スタンザで定義される DataPath の値です (存在する場合)。
- *prefix* は、キュー・マネージャー・スタンザで定義されるプレフィックスの値です。 プレフィックスは、通常は以下のいずれかの値になります。
  - Linux AIX AIX and Linux システム上の。/var/mqm
  - IBM i IBM i 上の /QIBM/UserData/mqm/。
- *qmgrname* は、キュー・マネージャー・スタンザで定義される Directory 属性の値です。 この値は、実際のキュー・マネージャー名とは異なる場合があります。 この値は、特殊文字の置換により変更されている可能性があります。
- キュー・マネージャー・スタンザが定義されている場所は、プラットフォームによって異なります。
  - Linux IBM i AIX mqs.ini ファイル (IBM i、 AIX and Linux の場合)。
  - Windows レジストリー (Windows の場合)。

注:

1. z/OS IBM MQ for z/OS をサーバーとして使用する場合、このファイルは IBM MQ クライアント・ワークステーション上に保持する必要があります。
2. MQCHLLIB を設定すると、CCDT の配置に使用されるパスをオーバーライドします。
3. MQCHLLIB には、MQCHLTAB 環境変数との組み合わせで機能する URL を含めることができます (52 ページの『CCDT への URL アクセス』 を参照してください)。
4. **MQCHLLIB** などの環境変数は、プラットフォーム固有の方法で、単一のプロセスまたはジョブをスコープにするか、あるいはシステム全体をスコープにすることができます。
5. サーバーで **MQCHLLIB** をシステム全体に設定すると、そのサーバー上のすべてのキュー・マネージャーの CCDT ファイルに同じパスが設定されます。 **MQCHLLIB** 環境変数を設定しないと、このパスはキュー・マネージャーごとに違うものになります。 キュー・マネージャーは、 **MQCHLLIB**(設定されている場合) の値を **crtmqm** コマンドまたは **strmqm** コマンドで読み取ります。
6. 単一のサーバーに複数のキュー・マネージャーを作成する場合には、以下の理由でそれらの区別が重要となります。 **MQCHLLIB** をシステム全体に設定すると、各キュー・マネージャーが同じ CCDT ファイルを更新します。 ファイルには、そのサーバー上にあるすべてのキュー・マネージャーからのクライアント接続定義が含まれます。 複数のキュー・マネージャーに同じ定義 (例えば、SYSTEM.DEF.CLNTCONN) が存在する場合、ファイルには最新の定義が含まれます。 キュー・マネージャーの作成時に **MQCHLLIB** を設定すると、CCDT で SYSTEM.DEF.CLNTCONN が更新されます。 この更新により、別のキュー・マネージャーによって作成された SYSTEM.DEF.CLNTCONN が上書きされます。 以前の定義を変更していた場合、その変更は失われます。 そのため、サーバーで **MQCHLLIB** をシステム全体の環境変数に設定する代わりに、他の方法を探すことを検討する必要があります。
7. クライアント接続定義の MQSC および PCF NOREPLACE オプションは、CCDT ファイルの内容を検査しません。 以前に別のキュー・マネージャーによって作成された同じ名前のクライアント接続チャネル定義は、NOREPLACE オプションの指定に関係なく置換されます。 以前に同じキュー・マネージャーによって作成された定義の場合は、置換されません。
8. コマンド **rcrmqobj** -t clchltab は、CCDT ファイルを削除し、再作成します。 このファイルは、そのコマンドの実行対象となるキュー・マネージャーで作成されたクライアント接続定義のみを使用して再作成されます。
9. CCDT を更新する他のコマンドは、同じチャネル名を持つクライアント接続チャネルのみを変更します。 そのファイル内の他のクライアント接続チャネルは変更されません。
10. **MQCHLLIB** のパスには、引用符は必要ありません。

詳細は、「CCDT の場所」、「URL への CCDT へのアクセス」、および「環境変数を使用したクライアントアプリケーションのキューマネージャへの接続」を参照してください。

MQCHLTAB

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

MQCHLTAB を設定するには、以下のいずれかのコマンドを使用します。

- Linux AIX On AIX and Linux:

```
export MQCHLTAB=filename
```

- Windows On Windows:

```
SET MQCHLTAB=filename
```

- IBM i On IBM i:

```
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(filename)
```

以下に例を示します。

```
SET MQCHLTAB=ccdf1.tab
```

クライアントの場合と同様に、サーバー上の **MQCHLTAB** 環境変数は、クライアント・チャネル定義テーブルの名前を指定します。

詳細は、「CCDT の場所」、「URL への CCDT へのアクセス」、および「環境変数を使用したクライアントアプリケーションのキューマネージャへの接続」を参照してください。

MQCLNTCF

MQCLNTCF 環境変数は、 IBM MQ MQI client 構成ファイルの場所を指定します。 このファイルには、 IBM MQ MQI clients によって使用される構成情報が含まれています。

MQCLNTCF 環境変数を使用して、 mqclient.ini ファイルのファイル・パスを変更できます。

この環境変数の形式は、完全な URL です。 これは、ファイル名が必ずしも mqclient.ini であるとは限らないことを意味します。これにより、ネットワーク接続されたファイル・システム上にファイルを配置することが容易になります。 詳細については、 IBM、 MQ、MQI クライアント構成ファイル、mqclient.ini、およびクライアント構成ファイルの場所を参照してください。

mqdotnet_trace_on

MQDOTNET_TRACE_ON 環境変数は、IBM MQ .NET 再配布可能クライアントのトレースを有効にするために使用されます。 0 以下の値はトレースを使用可能にしません。1 はデフォルトのトレースを使用可能にし、1 より大きい値は詳細トレースを使用可能にします。

詳しくは、 環境変数を使用した IBM MQ .NET アプリケーションのトレース および 環境変数を使用した IBM MQ .NET アプリケーションのトレースを参照してください。

MQIPADDRV

MQIPADDRV 環境変数は、チャネル接続に使用する IP プロトコルを指定します。 可能なストリング値は、"MQIPADDR_IPV4" または "MQIPADDR_IPV6"です。 これらの値は、**ALTER QMGR IPADDRV** の IPv4 および IPv6 、およびクライアント構成ファイルの TCP スタンザの **IPAddressVersion** 属性と同じ意味を持ちます。 環境変数が設定されていない場合は、 "MQIPADDR_IPV4" が想定されます。

MQIPADDRV を設定するには、以下のいずれかのコマンドを使用します。

- Linux AIX On AIX and Linux:

```
export MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6"/>
```

- Windows On Windows:

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

- IBM i On IBM i:

```
ADDENVVAR ENVVAR(MQIPADDRV) VALUE(MQIPADDR_IPV4|MQIPADDR_IPV6)
```

MQKEYRPWD

V 9.3.0 V 9.3.0

MQKEYRPWD 環境変数を設定する場合は、ユーザーに属するデジタル証明書を保持する鍵リポジトリーのパスワードを指定します。 *MQKEYRPWD* を使用する場合は、環境変数の値を設定する前にパスワードを暗号化する必要があります。

MQKEYRPWD を設定するには、以下のいずれかのコマンドを使用します。

- Linux AIX AIX and Linux システムの場合:

```
export MQKEYRPWD=passphrase
```

- Windows Windows システムの場合:

```
SET MQKEYRPWD=passphrase
```

- IBM i On IBM i:

```
ADDENVVAR ENVVAR(MQKEYRPWD) VALUE(passphrase)
```

この環境変数にはデフォルト値はありません。

V 9.3.0 V 9.3.0 詳しくは、以下を参照してください

- ALW AIX, Linux, and Windows 上の IBM MQ MQI client の鍵リポジトリー・パスワードの指定 および 鍵リポジトリー・パスワードの暗号化
- IBM i IBM i 上の IBM MQ MQI client の鍵リポジトリー・パスワードの指定 および 鍵リポジトリー・パスワードの暗号化。

mqlicense

Linux

Linux システムでは、 **MQLICENSE** 環境変数を使用して、製品のインストール後に IBM MQ ライセンスを受け入れるか表示することができます。

これを行う必要がある理由について詳しくは、 IBM MQ for Linux でのライセンスの受け入れ を参照してください。

MQLICENSE 環境変数は、以下の 2 つの値のいずれかに設定できます。

accept
インストール後にライセンスに同意します。

ビュー
ライセンスが受け入れられている場合は、ライセンスを表示します。

ライセンスのインストール後に同意するには、次のコマンドを使用します。

```
export MQLICENSE=accept
```

ライセンスを表示するには、次のコマンドを使用します。

```
export MQLICENSE=view
```

注 : 以下のコマンドを使用して、ライセンスを受け入れて表示することもできます。

- mqlicense (インストール後のライセンスの受け入れ)
- dspmqlic (IBM MQ ライセンスの表示)

MQMAXERRORLOGSIZE

Multi

MQMAXERRORLOGSIZE 環境変数は、バックアップにコピーされるキュー・マネージャー・エラー・ログのサイズを指定します。

詳しくは、『エラー・ログの使用』を参照してください。

MQNAME

Windows

MQNAME 環境変数は、 IBM MQ プロセスが使用できるローカル NetBIOS 名を指定します。 NetBIOS 接続は、Windows 稼働中のクライアントおよびサーバーにのみ適用されます。

MQNAME を設定するには、次のコマンドを使用します。

```
SET MQNAME=Your_env_Name
```

以下に例を示します。

```
SET MQNAME=CLIENT1
```

一部の NetBIOS 実装では、 IBM MQ MQI client 上で複数の IBM MQ アプリケーションを同時に実行する場合、アプリケーションごとに **MQNAME** によって設定された固有の名前が必要です。

詳しくは、 IBM MQ ローカル NetBIOS 名の定義を参照してください。

MQNOREMPOOL

MQNOREMPOOL 環境変数を設定すると、チャネル・プールがオフに切り替わり、チャネルがリスナーのスレッドとして実行されます。

詳しくは、 MCATYPE (メッセージ・チャネル・エージェント・タイプ)を参照してください。

MQPSE_TRACE_LOGFILE

パブリッシュ出口サンプル・プログラム AMQSPSE0 を実行するときに、 **MQPSE_TRACE_LOGFILE** 環境変数を使用します。これは、パブリケーションがサブスクライバーに配信される前にインターセプトする出口のサンプル C プログラムです。 アプリケーション・プロセスのトレースのために、この環境変数では、トレース・ファイルの書き込み先を指定します。

詳しくは、 パブリッシュ出口サンプル・プログラムを参照してください。

mqs_amscred_keyfile

MQS_AMSCRED_KEYFILE 環境変数を使用して、 IBM MQ Advanced Message Security (AMS) アプリケーションの実行時、または **runamscred** コマンドを使用して鍵ストア構成ファイルを保護する場合に使用する初期鍵ファイルをオーバーライドまたは指定できます。

詳しくは、AMS での鍵ストアと証明書の使用 および IBM MQ コンポーネント構成ファイルでのパスワードの保護を参照してください。

mqs_disable_all_intercept

旧バージョンの製品のキュー・マネージャーに接続しようとして、ネイティブ C クライアントで IBM MQ を使用している場合に 2085 (MQRC_UNKNOWN_OBJECT_NAME) エラーが報告される場合は、**MQS_DISABLE_ALL_INTERCEPT** 環境変数を使用して IBM MQ Advanced Message Security (AMS) を無効にすることができます。

注 : MQS_DISABLE_ALL_INTERCEPT 環境変数は、C クライアントにのみ使用できます。 Java クライアントの場合は、代わりに **AMQ_DISABLE_CLIENT_AMS** 環境変数を使用する必要があります。

詳しくは、 クライアントでの Advanced Message Security の無効化を参照してください。

MQS_IPC_HOST

IPC ファイル・システム・オブジェクトはシステムによって区別される必要があるため、キュー・マネージャーが実行される各システムのサブディレクトリーがディレクトリー・パスに追加されます。 生成されたホスト名の値に問題がある場合は、 **MQS_IPC_HOST** 環境変数を使用してホスト名を設定できます。

詳しくは、 Multiplatforms での IBM MQ ファイルの共有を参照してください。

MQ_keystore_conf

MQS_KEYSTORE_CONF 環境変数は、 IBM MQ Advanced Message Security (AMS) の鍵ストア構成ファイルがデフォルトの場所 *home_directory*/.mqs/keystore.conf にない場合に、そのファイルの場所を指定します。

詳しくは、 AMS での鍵ストアおよび証明書の使用を参照してください。

Managed File Transfer で問題が発生している場合は、 MFT が AMS の鍵ストア・プロパティーを読み取らない場合のトラブルシューティングを参照してください。

mqs_mqi_keyfile

V 9.3.0 V 9.3.0

MQS_MQI_KEYFILE 環境変数を設定すると、パスワード保護操作に使用される初期鍵を含む初期鍵ファイルの場所が指定されます。 初期鍵ファイルが指定されていない場合、デフォルトの初期鍵が IBM MQ パスワード保護システムによって使用されます。

MQS_MQI_KEYFILE を設定するには、以下のいずれかのコマンドを使用します。

- Linux AIX AIX and Linux システムの場合:

```
export MQS_MQI_KEYFILE=key file location
```

- Windows Windows システムの場合:

```
SET MQS_MQI_KEYFILE=key file location
```

- IBM i On IBM i:

```
ADDENVVAR ENVVAR(MQS_MQI_KEYFILE) VALUE(key file location)
```

V 9.3.0 V 9.3.0 詳しくは、AIX, Linux, and Windows での IBM MQ MQI client の初期キーの提供 および IBM i での IBM MQ MQI client の初期キーの提供を参照してください。

mqs_sslcryp_keyfile

V 9.3.0

MQS_SSLCRYP_KEYFILE 環境変数は、 qm.ini の SSL スタンザ で **SSLCryptoHardwareKeyFile** 属性を指定する代わりに、PKCS #11 暗号ハードウェア構成ストリングでパスワードを暗号化するために使用される初期鍵を含むファイルの絶対パスと名前を指定する別の方法です。 **MQS_SSLCRYP_KEYFILE** の優先順位は qm.ini ファイルよりも高いため、その値が他のどの値よりも優先されます。 詳しくは、暗号化ハードウェアを使用する IBM MQ クライアントを参照してください。

MQS_TRACE_OPTIONS

AIX

AIX でのコンポーネント・トレースを選択するには、 **MQS_TRACE_OPTIONS** 環境変数を使用して、詳細およびパラメーターのトレース機能を個別に活動化します。

注 : MQS_TRACE_OPTIONS 環境変数を設定するのは、IBM サポートから指示された場合のみにしてください。

詳しくは、 AIX and Linux でのトレースを参照してください。

mqs_getgrouplist_api

Linux

環境変数 **MQS_GETGROUPLIST_API** は、より低速な getgrent API ではなく、より高速な getgrouplist API を使用することで、パフォーマンスを向上させる。 何千ものグループを検索するような企業では、キュー・マネージャーのタイムアウトを回避するのに役立ちます。

環境変数 **MQS_GETGROUPLIST_API** を有効にするには、以下を実行する必要がある：

1. キュー・マネージャーを停止させます。
2. 以下のコマンドを発行します。

```
export MQS_GETGROUPLIST_API=1
```

3. キュー・マネージャーを再始動する。

この変更により RC2035 エラーを解決することもできます。 **MQS_GETGROUPLIST_API** 変数の詳細については、 Linux の「特定のグループ・メンバーシップの問題を解決する 」を参照のこと

MQSERVER

MQSERVER 環境変数は、最小チャネルを定義するために使用されます。 **MQSERVER** は、 IBM MQ サーバーの場所と使用する通信方式を指定します。

注 : MQSERVER を使用して、TLS チャネルまたはチャネル出口を持つチャネルを定義することはできません。 TLS チャネルの定義方法について詳しくは、 TLS を使用したチャネルの保護を参照してください。

以下の例は、 **MQSERVER** の設定方法を示しています。

- Linux AIX On AIX and Linux:

```
export  MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
```

- Windows On Windows:

```
SET  MQSERVER=SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)
```

- IBM i On IBM i:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)')
```

注 :

- チャネル名にスラッシュ (/) 文字を含めることはできません。この文字は、チャネル名、トランスポート・タイプ、および接続名を区切るために使用されるためです。 **MQSERVER** 環境変数を使用してクライアント・チャネルを定義する場合、最大メッセージ長 (MAXMSGL) として 100 MB が使用されます。 そのため、チャネルの最大メッセージ・サイズは、実際にはサーバーの SVRCONN チャネルに指定した値となります。
- トランスポート・タイプは、 IBM MQ クライアント・プラットフォームに応じて、 LU62 、 TCP 、 NETBIOS、 SPX のいずれかになります。
- 接続名は完全修飾ネットワーク名でなければなりません。 例えば、AMACHINE.ACOMPANY.COM(1414) などです。
- 接続名は、接続名のコンマ区切りリストにすることができます。 リスト内の接続名は、クライアント接続テーブルでの複数接続の場合と同じように使用されます。 接続名リストは、クライアントが試行する複数の接続を指定するために、キュー・マネージャー・グループの代わりに使用される場合があります。 複数インスタンス・キュー・マネージャーを構成する場合は、接続名リストを使用して別のキュー・マネージャー・インスタンスを指定することができます。

MQSERVER 環境変数を使用して IBM MQ MQI client マシンとサーバー・マシンの間のチャネルを定義する場合、これはアプリケーションで使用可能な唯一のチャネルであり、クライアント・チャネル定義テーブル (CCDT) に対する参照は行われません。

詳しくは、MQSERVER を使用した IBM MQ MQI クライアントでのクライアント接続チャネルの作成を参照してください。

MQSNOAUT

警告 : この機能は推奨されません。

MQSNOAUT 環境変数を任意の値に設定すると、オブジェクト権限マネージャー (OAM) が使用不可になり、セキュリティー検査が行われなくなります。 テスト環境では、その設定が適している場合もあります。 これには、許可機能と接続認証機能の両方が含まれます。 TLS、チャネル認証レコード、およびセキュリティー出口は影響を受けません。

MQSNOAUT 環境変数は、キュー・マネージャーの作成時にのみ有効になります。

警告 : OAM を使用可能にするには、キュー・マネージャーを削除し、環境変数を削除してから、 **MQSNOAUT** を指定せずにキュー・マネージャーを再作成する必要があります。

詳しくは、 AIX、 Linux、および Windows システムでのセキュリティー・アクセス検査の防止を参照してください。

MQSPREFIX

デフォルトの接頭部を変更する代わりに、 **MQSPREFIX** 環境変数を使用して、 **crtmqm** コマンドの **DefaultPrefix** をオーバーライドすることもできます。

詳細は「IBM MQ ファイル名」と「mqs.ini ファイルの AllQueueManagers スタンザ参照。

MQSSLCRYP

ALW

MQSSLCRYP 環境変数は、システム上に存在する暗号ハードウェアを構成するために使用できるパラメーター・ストリングを保持します。

V 9.3.0 許可される値は、クライアント構成ファイルの SSL スタンザ内の SSLCryptoHardware フィールド用と同じです。

MQSSLCRYP を設定するには、以下のいずれかのコマンドを使用します。

- Linux AIX AIX and Linux システムの場合:

```
export MQSSLCRYP=string
```

- Windows Windows システムの場合:

```
SET MQSSLCRYP=string
```

詳しくは、 IBM MQ コンポーネント構成ファイルでのパスワードの保護の「暗号ハードウェアを使用する AIX, Linux, and Windows および IBM MQ clients での暗号ハードウェアの構成」を参照してください。

MQSSLFIPS

MQSSLFIPS 環境変数は、 IBM MQ で暗号化が実行される場合に FIPS 認証アルゴリズムのみを使用するかどうかを指定します。 この環境変数を YES または NO に設定できます。デフォルトは NO です。 これらの値は、 **ALTER QMGR** コマンドの **SSLFIPS** パラメーターと同じです。

MQSSLFIPS を設定するには、以下のいずれかのコマンドを使用します。

- Linux AIX AIX and Linux システムの場合:

```
export MQSSLFIPS=YES|NO
```

- Windows Windows システムの場合:

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

- IBM i On IBM i:

```
ADDENVVAR ENVVAR(MQSSLFIPS) VALUE(YES|NO)
```

FIPS 認証アルゴリズムの使用は、暗号ハードウェアの使用によって影響を受けます。 詳しくは、 MQI クライアントでの実行時に FIPS 認証 CipherSpecs のみを使用することの指定を参照してください。

MQSSLKEYR

MQSSLKEYR 環境変数は、ユーザーなし) が含まれていることを意味します。

V 9.3.0 V 9.3.0 キー・リポジトリーの絶対パスとファイル名を指定します。 ファイル・サフィックスが指定されていない場合は、 .kdb と想定されます。

MQSSLKEYR を設定するには、以下のいずれかのコマンドを使用します。

- Linux AIX AIX and Linux システムの場合:

```
export MQSSLKEYR=pathname
```

- Windows Windows システムの場合:

```
SET MQSSLKEYR=pathname
```

- IBM i On IBM i:

```
ADDENVVAR ENVVAR(MQSSLKEYR) VALUE(pathname)
```

この環境変数にはデフォルト値はありません。

詳しくは、 **ALTER QMGR** コマンドの **SSLKEYR** パラメーターを参照してください。

MQSSLPROXY

MQSSLPROXY 環境変数は、 GSKit が OCSP チェックに使用する HTTP プロキシサーバーのホスト名とポート番号を指定します。

MQSSLPROXY を設定するには、以下のいずれかのコマンドを使用します。

- Linux AIX AIX and Linux システムの場合:

```
export MQSSLPROXY="string"
```

- Windows Windows システムの場合:

```
SET MQSSLPROXY= string
```

MQSSLPROXY で指定する文字列は、 GSKit が OCSP チェックに使用する HTTP プロキシサーバーのホスト名またはネットワークアドレスのいずれかです。 このアドレスの後に、オプションのポート番号が括弧で囲まれて続く場合もあります。 ポート番号を指定しないと、デフォルトの HTTP ポートである 80 が使用されます。

Linux AIX 例えば、AIX and Linux システムでは、以下のコマンドのいずれかを使用できます。

-
```
export MQSSLPROXY="proxy.example.com(80)"
```

-
```
export MQSSLPROXY="127.0.0.1"
```

詳しくは、 Online Certificate Status Protocol (OCSP) の操作を参照してください。

MQSSLRESET

MQSSLRESET 環境変数は、TLS 秘密鍵が再ネゴシエーションされる前に TLS チャネルで送受信される暗号化されていないバイト数を指定します。 これは、0 から 999 999 999 の範囲の整数に設定することができます。 デフォルトは 0 で、これは秘密鍵が再ネゴシエーションされないことを示します。 TLS 秘密鍵のリセット・カウントを 1 バイトから 32 KB の範囲で指定すると、TLS チャネルは 32 KB の秘密鍵リセット・カウントを使用します。 この秘密鍵リセット・カウントは、TLS の秘密鍵のリセット値が小さい場合に生じる可能性のある、鍵の過度のリセットを防ぐためにあります。

MQSSLRESET を設定するには、以下のいずれかのコマンドを使用します。

- Linux AIX AIX and Linux システムの場合:

```
export MQSSLRESET=integer
```

- Windows Windows システムの場合:

```
SET MQSSLRESET=integer
```

- IBM i On IBM i:

```
ADDENVVAR ENVVAR(MQSSLRESET) VALUE(integer)
```

詳しくは、 SSL および TLS 秘密鍵のリセットを参照してください。

MQSUITEB

ALW

IBM MQ は、AIX, Linux, and Windows プラットフォーム上で、NSA Suite B 規格に準拠して動作するよう構成することができます。

MQSUITEB 環境変数は、Suite B 準拠の暗号方式を使用するかどうかを指定します。 Suite B 暗号方式を使用する場合は、 **MQSUITEB** を以下のいずれかに設定して、暗号方式の強度を指定できます。

- NONE
- 128_BIT, 192_BIT
- 128_BIT
- 192_BIT

コンマ区切りリストを使用して、複数の値を指定することができます。 値 NONE を他の値と一緒に使用することは無効です。

詳しくは、 Suite B 用の IBM MQ の構成を参照してください。

MQTCPTIMEOUT

MQTCPTIMEOUT 環境変数は、 IBM MQ が TCP 接続呼び出しを待機する時間を指定します。

ODQ_MSG

runmqdlq 以外の送達不能キュー・ハンドラーを使用する場合は、サンプルのソース amqsdlq をベースとして使用することができます。 このサンプルは、製品に用意されている送達不能キュー処理プログラムとよく似ていますが、トレースとエラーの報告の動作が異なっています。 **ODQ_MSG** 環境変数を使用して、エラー・メッセージおよび情報メッセージを含むファイルの名前を設定します。 提供されているファイルの名前は amqsdlq.msg です。

詳しくは、 送達不能キュー・ハンドラー・サンプルを参照してください。

ODQ_TRACE

runmqdlq 以外の送達不能キュー・ハンドラーを使用する場合は、サンプルのソース amqsdlq をベースとして使用することができます。 このサンプルは、製品に用意されている送達不能キュー処理プログラムとよく似ていますが、トレースとエラーの報告の動作が異なっています。 トレースを有効にするには、 **ODQ_TRACE** 環境変数を YES または yes に設定します。

詳しくは、 送達不能キュー・ハンドラー・サンプルを参照してください。

WCF_TRACE_ON

WCF カスタム・チャネルでは、2 つの異なるトレース方式が使用可能です。 これらの 2 つのトレース方式は、独立して、または一緒に活動化されます。 それぞれの方式で独自のトレース・ファイルが生成されるので、両方のトレース方式をアクティブ化すると、2 つのトレース出力ファイルが生成されます。 この 2 種類のトレース方式を有効/無効にする操作には、4 つの組み合わせがあります。 WCF トレースを使用可能にするためのこれらの組み合わせだけでなく、 **WCF_TRACE_ON** 環境変数を使用して XMS .NET トレースを使用可能にすることもできます。

詳しくは、 IBM MQ の WCF カスタム・チャネルのトレースを参照してください。

WMQSOAP_HOME

WMQSOAP_HOME 環境変数は、 .NET SOAP over JMS サービス・ホスティング環境が正しくインストールされ、 IBM MQ で構成された後に、追加の構成ステップを実行する際に使用されます。 これは、ローカル・キュー・マネージャーからアクセスできる場所です。

詳しくは、 WCF client to a .NET service hosted by IBM MQ sample および WCF client to an Axis Java service hosted by IBM MQ sample を参照してください。

XMS_TRACE_ON、XMS_TRACE_FILE_NUMBER、XMS_TRACE_FILE_PATH、XMS_TRACE_FILE_SIZE、XMS_TRACE_FORMAT、XMS_TRACE_SPECIFICATION

IBM MQ classes for XMS .NET Framework を使用している場合は、 XMS 環境変数だけでなく、アプリケーション構成ファイルからもトレースを構成できます。 IBM MQ classes for XMS .NET (.NET Standard および .NET 6 ライブラリー)を使用している場合は、 XMS 環境変数からトレースを構成する必要があります。 一般に、トレースは IBM サポートの指導に従って使用します。

XMS .NET アプリケーションのトレースを有効にして構成するには、アプリケーションを実行する前に以下の環境変数を設定します。

XMS_TRACE_ON
XMS_TRACE_ON 環境変数が設定されている場合、デフォルトですべてのトレースが有効になります。

xms_trace_file_number
XMS_TRACE_FILE_NUMBER 環境変数は、保持するトレースファイルの数を指定する。 デフォルトは 4 です (1 つのアクティブ・ファイルと 3 つのアーカイブ・ファイル)。 最小数は 2 です。

XMS_TRACE_FILE_PATH
XMS_TRACE_FILE_PATH 環境変数は、トレース・レコードおよび FFDC レコードが書き込まれるディレクトリーの完全修飾パス名を指定します (これらのレコードを現行作業ディレクトリーから別の場所に書き込む場合)。

xms_trace_file_size
XMS_TRACE_FILE_SIZE 環境変数は、トレースファイルの最大許容サイズを指定する。 ファイルは、このサイズに達すると、アーカイブされて名前変更されます。 デフォルトの最大値は 20MB。 ファイルサイズはバイトで指定する。 例えば、`XMS_TRACE_FILE_SIZE="20000000"`などです。

XMS_TRACE_FORMAT
XMS_TRACE_FORMAT 環境変数は、必要なトレース・フォーマット (`BASIC` または `ADVANCED` のいずれか) を指定します。

XMS_TRACE_SPECIFICATION
XMS_TRACE_SPECIFICATION 環境変数は、 アプリケーション構成ファイルのトレース・セクション で定義されたトレース設定をオーバーライドします。 **XMS_TRACE_SPECIFICATION** は IBM MQ classes for XMS .NET Framework にのみ適用されます。

詳しくは、 XMS .NET アプリケーションのトレース および XMS 環境変数を使用した XMS .NET アプリケーションのトレースを参照してください。

Multi Multiplatforms での .ini ファイルの IBM MQ 構成情報の変更

構成 (`.ini`) ファイル内の情報を編集することにより、IBM MQ または個々のキュー・マネージャーの動作を、ご使用のインストール済み環境のニーズに合わせて変更できます。 IBM MQ MQI clients の構成オプションを変更することもできます。

このタスクについて

IBM MQ を管理する一連の構成属性 (またはパラメーター) に指定されている値を変更することにより、ノードまたはキュー・マネージャーのレベルで IBM MQ 構成情報を変更できます。

構成ファイル(またはスタンザ・ファイル)には、一つ以上のスタンザが含まれています。これらのスタンザは、`.ini` ファイル内の行のグループです。これらの行のグループは、共通の機能を持つか、ログ機能、チャネル機能、およびインストール可能サービスなどのシステムの一部を定義します。 IBM MQ 構成属性は、以下の構成ファイル内で変更できます。

IBM MQ 構成ファイル、 `mqs.ini`
`mqs.ini` ファイルは、ノード全体に影響を与えます。 `mqs.ini` ファイルは、IBM MQ のインストール済み環境ごとに 1 つあります。

IBM MQ 構成ファイルは、キュー・マネージャーに関連したデータを見つけるために使用されるので、構成ファイルが存在しなかったり、正しくなかったりすると、一部または全部の MQSC コマンドが失敗

します。 また、アプリケーションは、IBM MQ 構成ファイルに定義されていないキュー・マネージャーには接続できません。

インストール構成ファイル、mqinst.ini

Linux AIX AIX and Linux システムの場合、インストール構成ファイル `mqinst.ini` には、すべての IBM MQ インストールに関する情報が含まれています。 `mqinst.ini` ファイルの形式は固定されておらず、変更される可能性があるため、このファイルは編集したり、直接参照したりしないでください。 代わりに、コマンドを使用して編集する必要があります。

キュー・マネージャー構成ファイル、 qm.ini

`qm.ini` ファイルは、特定のキュー・マネージャーの変更に影響します。 ノード上のキュー・マネージャーごとに一つの `qm.ini` ファイルがあります。

IBM MQ MQI client 構成ファイル、 mqclient.ini

IBM MQ MQI clients の構成オプションは、クライアント構成ファイル (通常は `mqclient.ini` という名前) に別個に保持されます。

アクティビティー・トレース構成ファイル mqat.ini

`mqat.ini` ファイルは、アクティビティー・トレースの動作を構成するために使用されます。

構成ファイルを編集する必要があるのは、次のような場合です。

- 構成ファイルが失われる場合。 (可能であれば、バックアップからリカバリーしてください。)
- 1 以上のキュー・マネージャーを新しいディレクトリーに移す必要がある場合。
- デフォルトのキュー・マネージャーを変更する必要があります。 これは、既存のキュー・マネージャーを誤って削除した場合に発生する可能性があります。
- IBM サポートからこれを行うようお勧めします。

重要 : 通常、構成ファイルに対して行った変更は、キュー・マネージャーが次に開始されるまで有効になりません。

構成ファイルの編集に関する注意点 :

- 構成ファイルの属性の値は、以下の優先順位に従って設定されます。
  - コマンド行に入力されたパラメーターは、構成ファイルに定義されている値よりも優先されます。
  - `qm.ini` ファイルに定義されている値は、 `mqs.ini` ファイルに定義されている値よりも優先されます。
- インストール後に、 IBM MQ 構成ファイル内のデフォルト値を編集できます。
- キュー・マネージャーをバックアップする場合は、その構成ファイル (`qm.ini`) と中央の IBM MQ 構成ファイル (`mqs.ini`) の両方を必ず含めてください。
- 構成ファイル属性に誤った値を設定すると、属性全体が欠落しているのと同じ結果になります。 値は無視され、問題を示すオペレーター・メッセージが出されます。
- IBM i IBM i では、 `.ini` ファイルは IFS に常駐するストリーム・ファイルです。
- `mqat.ini` ファイルのフォーマットには、いくつかの構文規則があります。 詳しくは、「アプリケーション・アクティビティー・トレース」 「`mqat.ini` を使用したアクティビティー・トレース動作の構成」 を参照してください。

手順

1. 構成ファイルを編集する前に、構成ファイルをバックアップして、必要に応じて元に戻すことができるコピーを用意します。
2. 以下のいずれかの方法で `.ini` 構成ファイルを編集します。
   - 標準のテキスト・エディターを使用して手動で行う。 コメントは、コメント・テキストの前に";"または"#"文字を追加することによって、構成ファイルに組み込むことができます。 コメントを表示しない ";" または "#" 文字を使用する場合は、文字の前に "¥" 文字を付けることができます。 この文字は、構成データの一部として使用されます。

- ノード上のキュー・マネージャーの構成を変更するコマンドを使用して、自動的に。 詳しくは、コマンド・リファレンスを参照してください。

  Windows 例えば、 Windows specific コマンド **amqmdain** は、 qm.ini プロパティーのサブセットを自動的に更新します。 詳しくは、amqmdain を参照してください。

- Windows Linux Linux (x86 および x86-64) および Windows では、 IBM MQ Explorer を使用して qm.ini プロパティーのサブセットを更新できます。 詳しくは、MQ エクスプローラーを使用した IBM MQ の構成 を参照してください。

  注 : インストール可能サービスとそのコンポーネントの変更には重大な影響があるため、 IBM MQ Explorer ではインストール可能サービスは読み取り専用です。 そのため、qm.ini ファイルを編集して、インストール可能サービスに変更を加える必要があります。 詳しくは、141 ページの『qm.ini ファイルの Service スタンザ』を参照してください。

関連タスク
IBM MQ の管理

Multi IBM MQ 構成ファイル mqs.ini

IBM MQ 構成ファイル mqs.ini には、ノード上のすべてのキュー・マネージャーに関連する情報が含まれています。 これは、インストール時に自動的に作成されます。

注 : mqs.ini ファイルを編集する方法とタイミング、およびファイルに加えた変更が有効になるタイミングについて詳しくは、 84 ページの『Multiplatforms での .ini ファイルの IBM MQ 構成情報の変更』を参照してください。

ディレクトリーの場所

Linux AIX AIX and Linux では、データ・ディレクトリーとログ・ディレクトリーはそれぞれ、/var/mqm と/var/mqm/log です。

Windows Windows システムでは、データ・ディレクトリー mqs.ini のロケーションとログ・ディレクトリーのロケーションがレジストリーに保管されます。これは、そのロケーションが変わる可能性があるためです。 Windows に mqinst.ini ファイルがないため、AIX and Linux システム上に mqinst.ini 含まれているインストール設定情報もレジストリにあります（157 ページの『インストール構成ファイル、mqinst.ini』を参照）。

Windows Windows システムの mqs.ini ファイルは、 HKLM\SOFTWARE\IBM\IBM MQ キーで指定された WorkPath によって指定されます。 このファイルの内容は、次のとおりです。

- キュー・マネージャーの名前
- デフォルト・キュー・マネージャーの名前
- 各キュー・マネージャーに関連したファイルの位置

IBM i IBM i では、mqs.ini ファイルは /QIBM/UserData/mqm に保管されます。 ファイルには以下が含まれます。

- キュー・マネージャーの名前。
- デフォルト・キュー・マネージャーの名前。
- 各キュー・マネージャーに関連したファイルの位置。
- API 出口を識別する情報 (詳細については、API 出口の構成を参照)。

特に、mqs.ini 構成ファイルは、各キュー・マネージャーに関連したデータを見つけるために使用されます。

AIX and Linux 用の mqs.ini ファイルの例

Linux AIX

```
#*******************************************************************#
#* Module Name: mqs.ini                                            *#
#* Type       : IBM MQ Machine-wide Configuration File             *#
#* Function   : Define IBM MQ resources for an entire machine      *#
#*******************************************************************#
#* Notes      :                                                    *#
#* 1) This is the installation time default configuration          *#
#*                                                                 *#
#*******************************************************************#
AllQueueManagers:
#*******************************************************************#
#* The path to the qmgrs directory, below which queue manager data *#
#* is stored                                                       *#
#*******************************************************************#
DefaultPrefix=/var/mqm

LogDefaults:
   LogPrimaryFiles=3
   LogSecondaryFiles=2
   LogFilePages=4096
   LogType=CIRCULAR
   LogBufferPages=0
   LogDefaultPath=/var/mqm/log

QueueManager:
   Name=saturn.queue.manager
   Prefix=/var/mqm
   Directory=saturn!queue!manager
   InstallationName=Installation1

QueueManager:
   Name=pluto.queue.manager
   Prefix=/var/mqm
   Directory=pluto!queue!manager
   InstallationName=Installation2

DefaultQueueManager:
   Name=saturn.queue.manager

ApiExitTemplate:
   Name=OurPayrollQueueAuditor
   Sequence=2
   Function=EntryPoint
   Module=/usr/ABC/auditor
   Data=123

ApiExitCommon:
   Name=MQPoliceman
   Sequence=1
   Function=EntryPoint
   Module=/usr/MQPolice/tmqp
   Data=CheckEverything
```

Windows 用の mqs.ini ファイルの例

Windows

```
#*******************************************************************#
#* Module Name: mqs.ini                                            *#
#* Type       : IBM MQ Machine-wide Configuration File             *#
#* Function   : Define IBM MQ resources for an entire machine      *#
#*******************************************************************#
#* Notes      :                                                    *#
#* 1) This is the installation time default configuration          *#
#*                                                                 *#
#*******************************************************************#
AllQueueManagers:
#*******************************************************************#
#* The path to the qmgrs directory, below which queue manager data *#
#* is stored                                                       *#
#*******************************************************************#
DefaultPrefix=C:\ProgramData\IBM\MQ

LogDefaults:
   LogPrimaryFiles=3
   LogSecondaryFiles=2
   LogFilePages=4096
   LogType=CIRCULAR
```

```
   LogBufferPages=0
   LogDefaultPath=C:\ProgramData\IBM\MQ\log

QueueManager:
   Name=saturn.queue.manager
   Prefix=C:\ProgramData\IBM\MQ
   Directory=saturn!queue!manager
   InstallationName=Installation1

QueueManager:
   Name=pluto.queue.manager
   Prefix=C:\ProgramData\IBM\MQ
   Directory=pluto!queue!manager
   InstallationName=Installation2

DefaultQueueManager:
   Name=saturn.queue.manager

ApiExitTemplate:
   Name=OurPayrollQueueAuditor
   Sequence=2
   Function=EntryPoint
   Module=C:\usr\ABC\auditor
   Data=123

ApiExitCommon:
   Name=MQPoliceman
   Sequence=1
   Function=EntryPoint
   Module=C:\usr\MQPolice\tmpq
   Data=CheckEverything
```

IBM i 用の mqs.ini ファイルの例

IBM i

```
#********************************************************************#
#* Module Name: mqs.ini                                             *#
#* Type       : IBM MQ Configuration File                           *#
#* Function   : Define IBM MQ resources for the node                *#
#*                                                                  *#
#********************************************************************#
#* Notes      :                                                     *#
#* 1) This is an example IBM MQ configuration file                  *#
#*                                                                  *#
#********************************************************************#
AllQueueManagers:
#********************************************************************#
#* The path to the qmgrs directory, within which queue manager data *#
#* is stored                                                        *#
#********************************************************************#
DefaultPrefix=/QIBM/UserData/mqm

QueueManager:
Name=saturn.queue.manager
Prefix=/QIBM/UserData/mqm
Library=QMSATURN.Q
Directory=saturn!queue!manager

QueueManager:
Name=pluto.queue.manager
Prefix=/QIBM/UserData/mqm
Library=QMPLUTO.QU
Directory=pluto!queue!manager

DefaultQueueManager:
Name=saturn.queue.manager
```

注:

1. ノード上の IBM MQ は、キュー・マネージャー用とジャーナル用にデフォルト・ロケーションを使用します。
2. キュー・マネージャー saturn.queue.manager は、ノードでのデフォルト・キュー・マネージャーです。このキュー・マネージャーに関連したファイルのディレクトリーは、ファイル・システムで有効なファイル名に自動的に変換されています。

3. IBM MQ 構成ファイルは、キュー・マネージャーに関連するデータを見つけるために使用されるので、構成ファイルが存在しなかったり、正しくなかったりすると、一部または全部の IBM MQ コマンドが失敗します。 また、アプリケーションは、IBM MQ 構成ファイルに定義されていないキュー・マネージャーには接続できません。

mqs.ini スタンザ

重要: このトピックは、 mqs.ini ファイル内のスタンザに関する詳細情報にリンクしています。各スタンザには、そのスタンザ内のパラメーターに関する情報が含まれています。

Multi mqs.ini ファイルのスタンザと属性の要約

IBM MQ 構成ファイル mqs.ini のスタンザの属性の要約で、より多くの情報へのリンクがあります。

表 9. *mqs.ini* ファイルのスタンザ

スタンザと属性	属性の説明
AllQueueManagers スタンザ	
DefaultPrefix	qmgrs ディレクトリーへのパスはキュー・マネージャーのデータが保持されます。
ALW DefaultEphemeral 接頭部	キュー・マネージャーの一時データが保管されているディレクトリーへのパス。
Multi ConvEBCDICNewline	IBM MQ が EBCDIC の NL 文字を ASCII 形式に変換する方法
ApiExitCommon スタンザと ApiExitTemplate スタンザ	
名前	MQAXP 構造体の ExitInfoName フィールドで、API 出口に渡される API 出口の記述名。
機能	API 出口コードを含むモジュールへの関数エントリー・ポイントの名前。
モジュール	API 出口コードを含むモジュール。
データ	MQAXP 構造体の ExitData フィールドで API 出口に渡されるデータ。
Sequence	その他の API 出口に関してこの API 出口が呼び出される順序。
DefaultQueueManager スタンザ	
名前	キュー・マネージャー名が明示的に指定されていないコマンドを処理するキュー・マネージャーの名前。
ExitProperties スタンザ	
CLWLMode	クラスター・ワークロード (CLWL) 出口を FAST モードと SAFE モードのどちらで実行するか。
LogDefaults スタンザ	
LogPrimaryFiles	キュー・マネージャーの作成時に割り振られるログ・ファイル。
LogSecondaryFiles	1 次ファイルが足りなくなったときに割り振られるログ・ファイル。
LogFilePages	ログ・ファイルのページ数。 (ログ・ファイル・サイズは、4 KB ページ単位で指定します。)

表 9. *mqs.ini* ファイルのスタンザ (続き)	
スタンザと属性	**属性の説明**
LogType	キュー・マネージャーで使用するロギングのタイプ (循環またはリニア)。
LogBufferPages	書き込み用としてバッファー・レコードに割り振るメモリーの量。バッファーのサイズは 4 KB ページ単位で指定します。
LogDefaultPath	キュー・マネージャーのログ・ファイルが置かれるディレクトリー。
LogWriteIntegrity	高信頼ログ・レコードを書き込むためにロガーが使用する方式。
QueueManager スタンザ	
名前	キュー・マネージャーの名前。
PREFIX	キュー・マネージャーのファイルが保管される場所。
Directory	キュー・マネージャー・ファイルが保管されている `prefix\QMGRS` ディレクトリーの下にあるサブディレクトリーの名前。
DataPath	キュー・マネージャーの作成時に指定された明示的なデータ・パス。 このパスは、キュー・マネージャー・データのパスとして設定された Prefix と Directory をオーバーライドします。
InstallationName	このキュー・マネージャーに関連付けられている IBM MQ インストールの名前。
EphemeralPrefix	キュー・マネージャーの一時的データが保管される場所。

Multi mqs.ini ファイルの AllQueueManagers スタンザ

AllQueueManagers スタンザは、キュー・マネージャーに関連付けられたファイルが保管される `qmgrs` ディレクトリーへのパス、実行可能ライブラリーへのパス、および EBCDIC 形式のデータを ASCII 形式に変換する方法を指定できます。

`mqs.ini` ファイル内の AllQueueManagers スタンザを使用して、すべてのキュー・マネージャーに関する情報を指定します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer `General` および `Extended` IBM MQ のプロパティー・ページを使用します。

DefaultPrefix=*directory_name*
この属性は、キュー・マネージャーのデータが保持される `qmgrs` ディレクトリーへのパスを指定します。

キュー・マネージャーのデフォルト・プレフィックスを変更した場合、インストール時に作成されたディレクトリー構造を複製してください。 特に、qmgrs 構造を作成する必要があります。 デフォルト接頭部を変更する前に IBM MQ を停止してください。IBM MQ の再始動は、構造を新しい位置に移し、デフォルト接頭部を変更した後に行ってください。

注 : ALW AIX and Linux システムの場合は `/var/mqm/errors` ディレクトリー、 Windows システムの場合は `\errors` ディレクトリーを削除しないでください。

デフォルトの接頭部を変更する代わりに、環境変数 **MQSPREFIX** を使用して、 **crtmqm** コマンドの **DefaultPrefix** をオーバーライドすることができます。

オペレーティング・システムの制約があるため、指定するパスをできるだけ短くして、パス長とすべてのキュー・マネージャー名の長さの合計が最大 70 文字になるようにしてください。

Multi **DefaultEphemeralPrefix= *directory_name***
この属性には、キュー・マネージャーの一時データ (IPC ソケットなど) を保持し、キュー・マネージャーの作成時にキュー・マネージャーの **EphemeralPrefix** を設定するときにのみ使用される、ディレクトリーのパスを指定します。 また、デフォルト値を変更する場合は、ディレクトリーを自分で作成する必要があります。 IBM MQ グループがそのディレクトリーに書き込むためのアクセス許可を持つ一時データ・ディレクトリーを作成する必要があります。

`mqs.ini` ファイルを変更する代わりに、環境変数 **MQ_EPHEMERAL_PREFIX** を使用して、 **crtmqm** コマンドの **DefaultEphemeralPrefix** をオーバーライドすることもできます。

オペレーティング・システムの制約事項のため、デフォルトの一時接頭部は以下のように制限されます。

- Linux AIX 12 文字 (AIX and Linux プラットフォーム)。
- IBM i 24 文字 (IBM i)。

MQ Appliance **DefaultEphemeralPrefix** は IBM MQ Appliance ではサポートされていません。

Multi **ConvEBCDICNewline=NL_TO_LF|TABLE|ISO**
EBCDIC コード・ページには、ASCII コード・ページではサポートされない改行 (NL) 文字が含まれています (ただし、ISO 仕様の ASCII には、この改行文字と同等の文字が含まれているものもいくつかあります)。 **ConvEBCDICNewline** 属性を使用して、IBM MQ が EBCDIC の NL 文字を ASCII フォーマットに変換する方法を指定します。

IBM i IBM MQ for IBM i では CCSID 1253 は ISO CCSID と見なされ、NL_TO_LF は ISO と ASCII の両方の変換に影響を及ぼします。

z/OS **ConvEBCDICNewline** 属性は、z/OS では使用できません。 z/OS での動作は、`ConvEBCDICNewline=TABLE` と同等です。 他のプラットフォームでのデフォルトは異なる場合があるので注意してください。

NL_TO_LF
EBCDIC から ASCII へのすべての変換について、EBCDIC NL 文字 (X'15') を ASCII 改行 (LF) 文字 (X'0A') に変換します。

NL_TO_LF はデフォルトです。

TABLE
EBCDIC から ASCII への 変換に使用している変換テーブルに従って、EBCDIC の NL 文字を変換します。

このタイプの変換の結果は、プラットフォームと言語によって 異なる場合があります。同じプラットフォーム上でも、使用する CCSID が異なれば動作が違ってくる場合があります。

ISO
次のものを変換します。

- ISO CCSID (TABLE メソッドを使用)
- その他のすべての CCSID (NL_TO_CF メソッドを使用)

使用できる ISO CCSID は、91 ページの表 10 に示されています。

表 *10.* 使用できる *ISO CCSID* のリスト

CCSID	**コード・セット**
819	ISO8859-1
912	ISO8859-2

表 10. 使用できる ISO CCSID のリスト (続き)	
CCSID	**コード・セット**
915	ISO8859-5
1089	ISO8859-6
813	ISO8859-7
916	ISO8859-8
920	ISO8859-9
1051	roman8

ASCII CCSID が ISO サブセットの 1 つではない場合は、デフォルトにより、**ConvEBCDICNewline** は NL_TO_LF に設定されます。

IBM MQ 9.1.0 Fix Pack 2 および IBM MQ 9.1.2 以降では、 **ConvEBCDICNewline** スタンザ属性の代わりに **AMQ_CONVEBCDICNEWLINE** 環境変数 を使用できます。例えば、mqs.ini ファイルを使用できない状況で、クライアント・サイドで **ConvEBCDICNewline** 機能を提供することができます。 この環境変数では、**ConvEBCDICNewline** 属性と同じ値 (NL_TO_LF、TABLE、ISO) を設定できます。 スタンザ属性と環境変数の両方が設定されている場合は、スタンザ属性の方が優先されます。

Multi mqs.ini ファイルの ApiExitCommon スタンザおよび ApiExitTemplate スタンザ

ApiExit テンプレートおよび ApiExit 共通スタンザは、すべてのキュー・マネージャーの API 出口ルーチンを識別します。

mqs.ini ファイル内の ApiExit テンプレートおよび ApiExit 共通スタンザを使用して、すべてのキュー・マネージャーの API 出口ルーチンを識別します。 (個々のキュー・マネージャーの API 出口ルーチンを識別するには、 110 ページの『qm.ini ファイルの ApiExitLocal スタンザ』で説明されているように、 ApiExit ローカル・スタンザを使用します。)

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer Exits IBM MQ のプロパティー・ページを使用します。

Windows Windows では、 **amqmdain** コマンドを使用して、API 出口の項目を変更することもできます。

これらの属性の使用方法について詳しくは、API 出口の構成を参照してください。

Name=ApiExit_name
MQAXP 構造体の ExitInfoName フィールドで、API 出口に渡される API 出口の記述名。

この名前は一意で、48 文字以内であり、さらに IBM MQ オブジェクト名 (例えばキュー名) に有効な文字のみを含む必要があります。

Function=function_name
API 出口コードを含むモジュールへの関数エントリー・ポイントの名前。 このエントリー・ポイントは MQ_INIT_EXIT 関数です。

このフィールドの長さは MQ_EXIT_NAME_LENGTH に限定されます。

Module=module_name
API 出口コードを含むモジュール。

このフィールドにモジュールの絶対パス名が入っている場合、それがそのまま使用されます。 このフィールドにモジュール名のみが入っている場合、モジュールは、 qm.ini ファイルの ExitPath スタンザ内の **ExitsDefaultPath** 属性を使用して配置されます。

異なるスレッド化ライブラリーをサポートするプラットフォームで、API 出口モジュールの非スレッド化バージョンとスレッド化バージョンの両方を提供する必要があります。 スレッド化バージョンは、

接尾部 _r を持っている必要があります。 IBM MQ アプリケーション・スタブのスレッド化バージョンは、指定のモジュールがロードされる前にその名前に _r を暗黙的に追加します。

このフィールドの長さは、プラットフォームがサポートする最大パス長に限定されます。

Data=data_name
MQAXP 構造体の ExitData フィールドで API 出口に渡されるデータ。

この属性を指定すると、先行および末尾のブランクが除去されて残りのストリングは 32 文字に切り捨てられ、その結果が出口に渡されます。 この属性を省略すると、デフォルト値の 32 文字のブランクが出口に渡されます。

このフィールドの最大長は、32 文字です。

Sequence=sequence_number
その他の API 出口に関してこの API 出口が呼び出される順序。 小さなシーケンス番号の出口は、より大きなシーケンス番号の出口よりも先に呼び出されます。 出口のシーケンス番号は連続である必要はありません。 つまり、1、2、3 の順序は、7、42、1096 の順序と同じ結果となります。 2 つの出口のシーケンス番号が同じ場合は、キュー・マネージャーが最初に呼び出す出口を決定します。 MQAXP の ExitChainAreaPtr で示される ExitChainArea に時刻またはマーカーを設定するかまたは独自のログ・ファイルを作成して、イベントの後に呼び出された出口を判別できます。

この属性は、符号なし数値です。

Multi mqs.ini ファイルの DefaultQueueManager スタンザ

DefaultQueue マネージャー・スタンザは、ノードのデフォルト・キュー・マネージャーを指定します。

mqs.ini ファイルの DefaultQueue マネージャー・スタンザを使用して、デフォルト・キュー・マネージャーを指定します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer General IBM MQ のプロパティー・ページを使用します。

Name=*default_queue_manager*
デフォルト・キュー・マネージャーは、キュー・マネージャー名が明示的に指定されていないすべてのコマンドを処理します。 **DefaultQueueManager** 属性は、新規のデフォルト・キュー・マネージャーを作成すると、自動的に更新されます。 誤って新規のデフォルト・キュー・マネージャー作成した場合、それを元のプログラムに戻すときは、**DefaultQueueManager** 属性を手動で変更してください。

Multi mqs.ini ファイルの ExitProperties スタンザ

ExitProperties スタンザは、キュー・マネージャー出口プログラムによって使用される構成オプションを指定します。

mqs.ini ファイルの ExitProperties スタンザを使用して、キュー・マネージャー出口プログラムによって使用される構成オプションを指定します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer Extended IBM MQ のプロパティー・ページを使用します。

CLWLMode = SAFE (デフォルト) | 高速
クラスター・ワークロード (CLWL) 出口を使用すると、MQI 呼び出し (例えば、 MQOPEN、 MQPUT) への応答として、クラスター内のどのクラスター・キューを開くかを指定できます。 CLWL 出口は、**CLWLMode** 属性に指定した値に応じて、FAST モードまたは SAFE モードのいずれかで実行されます。 **CLWLMode** 属性を省略すると、クラスター・ワークロード出口は SAFE モードで実行されます。

SAFE
CLWL 出口をキュー・マネージャーから独立したプロセスで実行します。 これはデフォルトです。

SAFE モードでの実行中にユーザー作成の CLWL に問題が起きた場合は、次のような結果が生じます。

- CLWL サーバー・プロセス (amqzlwa0) は失敗します。

- キュー・マネージャーは CLWL サーバー・プロセスを再始動します。
- エラーがエラー・ログに記録されます。 進行中の MQI 呼び出しがある場合は、戻りコードが戻されます。

キュー・マネージャーの保全性は維持されます。

注 : CLWL 出口を別のプロセスで実行すると、パフォーマンスに影響を与える可能性があります。

FAST
クラスター出口は、キュー・マネージャー・プロセスでインラインで実行します。

このオプションを指定すると、SAFE モードで実行した場合のプロセスの切り替えコストを回避できるのでパフォーマンスは向上しますが、代わりにキュー・マネージャーの保全性が保証されないことになります。 CLWL 出口を FAST モードで実行する必要があるのは、CLWL 出口に問題がなく、特にパフォーマンスについて懸念があると確信している場合のみです。

CLWL 出口を FAST モードで実行しているときに問題が発生すると、キュー・マネージャーは失敗し、キュー・マネージャーの保全性が損なわれる恐れがあります。

Multi mqs.ini ファイルの LogDefaults スタンザ

LogDefaults スタンザは、すべてのキュー・マネージャーのログ・デフォルトに関する情報を指定します。

`mqs.ini` ファイルの LogDefaults スタンザを使用して、すべてのキュー・マネージャーのログ・デフォルトに関する情報を指定します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer `Default log settings` IBM MQ のプロパティー・ページを使用します。

デフォルト以外の値が必要な場合は、 LogDefaults スタンザにその値を明示的に指定する必要があります。

LogDefaults スタンザが存在しない場合、 IBM MQ デフォルトが使用されます。 ログ属性は、ユーザーがキュー・マネージャーを作成するときにデフォルト値として使用されますが、**crtmqm** コマンドにログ属性を指定することにより、指定変更することができます。 このコマンドについて詳しくは、「**crtmqm**」を参照してください。

キュー・マネージャーが作成された後は、そのキュー・マネージャーのログ属性は、132 ページの『qm.ini ファイルの Log スタンザ』に説明されている設定値から取られます。

注 : 新規 IBM MQ インストール用に提供された LogDefaults スタンザには、属性の明示的な値は含まれません。 属性が欠けている場合は、新規キュー・マネージャーの作成時にこの値のデフォルトが使用されます。 LogDefaults スタンザのデフォルト値は、86 ページの『AIX and Linux 用の mqs.ini ファイルの例』および87 ページの『Windows 用の mqs.ini ファイルの例』に示されています。 `LogBufferPages` 属性に値 0 を指定した場合は、512 を意味します。

90 ページの『mqs.ini ファイルの AllQueueManagers スタンザ』に指定されているデフォルトの接頭部、および132 ページの『qm.ini ファイルの Log スタンザ』に指定されている特定のキュー・マネージャーのログ・パスにより、キュー・マネージャーとそのログを異なる物理ドライブに置くことができます。 デフォルトではこの 2 つは同じドライブに入りますが、別々のドライブを使用することをお勧めします。

ログ・サイズの計算方法については、646 ページの『ログのサイズの計算』を参照してください。

注 : 次のパラメーター・リストに示す制限は、IBM MQ により設定される制限です。 オペレーティング・システムの制限により、最大可能ログ・サイズがさらに減少することもあります。

LogPrimaryFiles = 3 (デフォルト) |*2-254* (Windows) |*2-510* (AIX and Linux)
キュー・マネージャーの作成時に割り振られるログ・ファイル。

1 次ログ・ファイルの最小数は 2 であり、最大数は Windows では 254、AIX and Linux では 510 です。 デフォルトは 3 です。

1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 3 以上にする必要があり、かつ Windows では 255 以下、AIX and Linux では 511 以下にする必要があります。

この値は、キュー・マネージャーの作成時または開始時に調べられます。 キュー・マネージャーが作成された後に、この値を変更することができます。 ただし、この変更された値は、キュー・マネージャーが再始動されるまで有効にならないので、効果はただちに現れません。

LogSecondaryFiles = 2 (デフォルト) |*1-253* (Windows) |*1-509* (AIX and Linux)
1 次ファイルが足りなくなったときに割り振られるログ・ファイル。

2 次ログ・ファイルの最小数は 1 であり、最大数は Windows では 253、AIX and Linux では 509 です。デフォルトの数は 2 です。

1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 3 以上にする必要があり、かつ Windows では 255 以下、AIX and Linux では 511 以下にする必要があります。

この値は、キュー・マネージャーの始動時に検査されます。 この値は変更することができます。ただし、変更された値は、キュー・マネージャーが再始動されるまでは有効にはなりません。有効になった場合でも効果がただちに現れるとは限りません。

LogFilePages=*number*
ログ・データは、ログ・ファイルと呼ばれる一連のファイルに保持されます。 ログ・ファイル・サイズは、4 KB ページ単位で指定します。

ログ・ファイルのデフォルトのページ数は 4096 です。 これは、16 MB のログ・ファイル・サイズと同じです。

AIX and Linux では、ログ・ファイルの最小ページ数は 64 です。 Windows では、ログ・ファイルの最小ページ数は 32 です。最大ページ数はどちらも 65 535 です。

注 : キュー・マネージャーの作成時に指定したログ・ファイルのサイズを、個々のキュー・マネージャーについて変更することはできません。

LogType = 循環 (デフォルト) | LINEAR
使用するログのタイプ。 デフォルトは 循環です。

CIRCULAR
システムの停止時に進行中だったトランザクションをロールバックするログを使用して、再始動リカバリーを開始します。

循環ロギングの詳細については、641 ページの『ログのタイプ』を参照してください。

LINEAR
再始動リカバリーと、メディアまたは順方向リカバリー (ログの内容を再生することにより、消失または損傷したデータを作成する) の両方を可能にします。

リニア・ロギングの詳細については、641 ページの『ログのタイプ』を参照してください。

デフォルトを変更する場合は、LogType 属性を編集するか、**crtmqm** コマンドでリニア・ロギングを指定します。

IBM MQ 9.1.0 から、キュー・マネージャーの作成後にロギング方式を変更できるようになりました。詳しくは、migmqlog を参照してください。

LogBufferPages=0 (デフォルト) |*0* から *4096*
書き込み用としてバッファー・レコードに割り振るメモリーの量。バッファーのサイズは 4 KB ページ単位で指定します。

バッファーの最小ページ数は 18 であり、最大ページ数は 4096 です。 バッファーが大きいとスループットが高くなりますが、これは特に大きなメッセージに適用されます。

0 (デフォルト) を指定すると、キュー・マネージャーは 512 (2048 KB) のサイズを選択します。

1 から 17 の範囲の数を指定すると、キュー・マネージャーはデフォルトで 18 (72 KB) になります。 18 から 4096 の範囲の数を指定すると、キュー・マネージャーは指定された数を使用して割り振るメモリーを設定します。

LogDefaultPath=*directory_name*
キュー・マネージャーのログ・ファイルが置かれるディレクトリー。 ディレクトリーは、キュー・マネージャーが書き込みを行える ローカル装置に置くか、できればメッセージ・キューとは異なるドライブに置きます。 異なるドライブを指定すると、システム障害時の保護機能が加わります。

デフォルトは次のとおりです。

- Windows *DefaultPrefix*\log for IBM MQ for Windows 。ここで、 *DefaultPrefix* は、 All Queue Managers IBM MQ プロパティー・ページの DefaultPrefix 属性で指定された値です。 この値は、インストール時に設定されます。
- Linux AIX AIX and Linux システムの場合 /var/mqm/log

あるいは、 **-ld** フラグを使用して、 **crtmqm** コマンドでディレクトリーの名前を指定することもできます。 キュー・マネージャーが作成されると、キュー・マネージャー・ディレクトリーの下にもう 1 つディレクトリーが作成され、ログ・ファイルの保存に使用されます。 このディレクトリーの名前はキュー・マネージャー名に基づいて付けられます。 これによってログ・ファイル・パスが確実に固有になり、ディレクトリー名の長さに関する制限にも抵触しないことになります。

crtmqm コマンドで **-ld** を指定しない場合、 mqs.ini ファイル内の **LogDefaultPath** 属性の値が使用されます。

複数のキュー・マネージャーで別々のログ・ディレクトリーが使用されるように、キュー・マネージャー名がディレクトリー名に付加されます。

キュー・マネージャーが作成されると、 構成情報内のログ属性の中に **LogPath** 値が作成されて、キュー・マネージャーのログに完全ディレクトリー名が与えられます。 この値は、キュー・マネージャーの開始時または削除時に、ログを見つけるために使用されます。

LogWriteIntegrity=SingleWrite|DoubleWrite|TripleWrite（デフォルト）
高信頼ログ・レコードを書き込むためにロガーが使用する方式。

TripleWrite (デフォルト)

なお、DoubleWrite を選択可能ですが、そのようにした場合、システムでは TripleWrite と解釈されます。

SingleWrite
SingleWrite は、IBM MQ リカバリー・ログをホスティングしているファイル・システムおよびデバイスが、最小単位 4KB の書き込みを明示的に保証している場合にのみ使用してください。

すなわち、何らかの原因で 4KB ページの書き込みが失敗した場合、変更前イメージまたは変更後イメージの 2 つの状態しかありません。 中間の状態は考えられません。

注 : 持続するワークロードにおける並行性が十分であるなら、デフォルト値 TripleWrite 以外の値に設定することで得られる利点はほとんどありません。

詳細については、135 ページの『LogWriteIntegrity-using SingleWrite または TripleWrite』を参照してください。

Multi mqs.ini ファイルの QueueManager スタンザ

QueueManager スタンザは、キュー・マネージャー・ディレクトリーの場所を指定します。

キュー・マネージャーごとに 1 つの QueueManager スタンザがあります。 このスタンザの属性は、キュー・マネージャー名と、そのキュー・マネージャーに関連付けられたファイルを含むディレクトリーの名前を指定します。 ディレクトリーの名前はキュー・マネージャー名に基づいていますが、キュー・マネージャー名が有効なファイル名でない場合は、変換されます。 名前変換について詳しくは、 IBM MQ ファイル名についての理解を参照してください。

Name=*queue_manager_name*
キュー・マネージャーの名前。

Prefix=*prefix*
キュー・マネージャーのファイルが保管される場所。 デフォルトでは、この値は、 mqs.ini ファイル内の All Queue Managers スタンザの **DefaultPrefix** 属性で指定された値と同じです。

Directory=*name*
キュー・マネージャー・ファイルが保管されている *prefix*\QMGRS ディレクトリーの下にあるサブディレクトリーの名前。 この名前はキュー・マネージャー名に基づいていますが、重複する名前がある場合、またはキュー・マネージャー名が無効なファイル名の場合は、変換される場合があります。

DataPath=*path*
キュー・マネージャーの作成時に指定された明示的なデータ・パス。これは、キュー・マネージャー・データへのパスとして **Prefix** および **Directory** をオーバーライドします。

InstallationName=*name*
このキュー・マネージャーに関連付けられている IBM MQ インストールの名前。 このキュー・マネージャーと対話するときには、このインストールからコマンドを使用する必要があります。

IBM i **Library=*name***
このキュー・マネージャーに関連する IBM i オブジェクト (例えばジャーナルやジャーナル・レシーバー) を保管するライブラリーの名前。 この名前はキュー・マネージャー名に基づくものですが、重複する名前がある場合、またはキュー・マネージャー名が無効なライブラリー名の場合は、変換される場合があります。

EphemeralPrefix= *name*
キュー・マネージャーの一時的データが保管される場所。

デフォルトでは、この値は存在しません。つまり、データは接頭部の場所の下に保管されます。

この値は、キュー・マネージャーの作成時に、**MQ_EPHEMERAL_PREFIX** 環境変数の値、または mqs.ini ファイル内の AllQueueManagers スタンザの **DefaultEphemeralPrefix** 属性から設定されます。

IBM i オペレーティング・システムの制約事項のため、デフォルトの一時接頭部は IBM i では 24 文字に制限されます。

関連タスク
464 ページの『キュー・マネージャーとインストールの関連付け』
キュー・マネージャーを作成すると、そのキュー・マネージャーは、**crtmqm** コマンドを実行したインストールに自動的に関連付けられます。 AIX, Linux, and Windows では、**setmqm** コマンドを使用して、キュー・マネージャーに関連付けられているインストールを変更できます。

Windows Advanced Configuration and Power Interface (ACPI)

Windows は、Advanced Configuration and Power Interface (ACPI) 規格をサポートしています。 この規格により、ACPI 対応のハードウェアを使用する Windows ユーザーは、システムがサスペンド・モードに入るときと、サスペンド・モードからレジュームするときに、チャネルを停止および再始動することができます。

IBM MQ Explorer の ACPI IBM MQ プロパティー・ページを使用して、システムが中断要求を受信したときの IBM MQ の動作を指定します。

ACPI IBM MQ のプロパティー・ページで指定されている設定は、アラート・モニターの実行時にのみ適用されることに注意してください。 アラート・モニター・アイコンは、アラート・モニター が実行中である場合にタスクバーに表示されます。

DoDialog=Y | N
サスペンド要求があったときにダイアログを表示します。

DenySuspend=Y | N
サスペンド要求を拒否します。 これは、DoDialog=N の場合、または DoDialog=Y を指定していても、ノートブックのふたが閉じられているなどの理由でダイアログを表示できない場合に使用されます。

CheckChannelsRunning=Y | N
実行中のチャネルがないか確認します。 この結果によって、他の設定の結果が分かります。

次の表は、これらのパラメーターのそれぞれの組み合わせの効果を示しています。

DoDialog	DenySuspend	CheckChannels Running	アクション
N	N	N	サスペンド要求を受け入れます。
N	N	Y	サスペンド要求を受け入れます。
N	Y	N	サスペンド要求を拒否します。
N	Y	Y	チャネルが実行中の場合はサスペンド要求を拒否し、チャネルが実行中でない場合はサスペンド要求を受け入れます。
Y	N	N	ダイアログを表示します (注を参照。サスペンド要求を受け入れます)。 これはデフォルトです。
Y	N	Y	実行中のチャネルがない場合はサスペンド要求を受け入れ、チャネルが実行中の場合はダイアログを表示します (注を参照。サスペンド要求を受け入れます)。
Y	Y	N	ダイアログを表示します (注を参照。サスペンド要求を拒否します)。
Y	Y	Y	実行中のチャネルがない場合はサスペンド要求を受け入れ、チャネルが実行中の場合はダイアログを表示します (注を参照。サスペンド要求を拒否します)。

注: アクションがダイアログを表示するようになっている場合で、ダイアログを表示できない場合 (ノートブックのふたが閉じられているなどの理由で) は、サスペンド要求を受け入れるか拒否するかを判別するために、DenySuspend オプションが使用されます。

Multi キュー・マネージャー構成ファイル qm.ini

キュー・マネージャー構成ファイル qm.ini には、特定のキュー・マネージャーに関連する情報が含まれています。 個々のキュー・マネージャーの構成を変更するために使用できる属性によって、IBM MQ の設定がオーバーライドされます。

各キュー・マネージャーごとに、1 つのキュー・マネージャー構成ファイルがあります。 qm.ini ファイルは、それが関連付けられているキュー・マネージャーが作成されると、自動的に作成されます。

注: qm.ini ファイルを編集する方法とタイミング、およびファイルに加えた変更がいつ有効になるかについて詳しくは、 84 ページの『Multiplatforms での .ini ファイルの IBM MQ 構成情報の変更』 を参照してください。

IBM MQ 9.0.4 および IBM MQ 9.0.0 Fix Pack 2 以降、**strmqm** コマンドは、キュー・マネージャーを完全に開始する前に qm.ini ファイル内の CHANNELS および SSL スタンザの構文をチェックします。これにより、**strmqm** によって qm.ini ファイルのエラーが検出された場合に、誤りを容易に見つけたり、それを迅速に修正したりできます。 詳しくは、strmqm を参照してください。

qm.ini ファイルの場所

Linux AIX AIX and Linux システムでは、qm.ini ファイルは、キュー・マネージャーがディレクトリー・ツリーのルートに保持されます。 例えば、QMNAME という名前のキュー・マネージャーの構成ファイルのパスと名前は、次のとおりです。

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

Windows Windows システムでは、 qm.ini ファイルの場所は、 HKLM\SOFTWARE\IBM\WebSphere MQ キーで指定された WorkPath によって指定されます。 例えば、QMNAME という名前のキュー・マネージャーの構成ファイルのパスと名前は、以下のとおりです。

```
C:\ProgramData\ IBM \ MQ\qmgrs\QMNAME\qm.ini
```

IBM i qm.ini ファイルは、*mqmdata directory*/*QMNAME*/qm.ini で保持されています。ここで、*mqmdata directory* デフォルトでは/QIBM/UserData/mqm*QMNAME* は初期設定ファイルが適用されるキュー・マネージャーの名前です。

注: mqs.ini ファイル内の *mqmdata directory* を変更することができます。

キュー・マネージャーの名前の長さは 48 文字までです。 ただし、この名前が有効である、または固有であるかは保証されません。 このため、キュー・マネージャー名に基づいてディレクトリー名が生成されます。 このプロセスは名前変換 と呼ばれています。 詳しくは、IBM MQ ファイル名 および IBM i 上のオブジェクト名を参照してください。

qm.ini スタンザ

重要:

- このトピックは、 qm.ini ファイル内のスタンザに関する詳細情報にリンクしています。 各スタンザには、該当する場合の例を含め、そのスタンザ内のパラメーターに関する情報が含まれています。
- 各スタンザは、そのスタンザが適用される IBM MQ for Multiplatforms のプラットフォームを示します。

Multi 始動時の qm.ini の自動構成

IBM MQ 9.2.0 から、キュー・マネージャーを構成して、すべてのキュー・マネージャーを開始する時に、qm.ini オーバーライドを含むファイルの内容またはファイルのセットを自動的に適用するように構成することができます。

これを利用すると、構成を変更してキュー・マネージャーの次の再始動時に自動的に再生することができます。 例えば、qm.ini オーバーライドがマウントされたドライブ上にある場合は、開始時にすべてのキュー・マネージャーに最新バージョンが適用される集中構成を使用することができます。

この機能を使用すると、均一なクラスターを自動クラスター機能で作成する作業が簡単になります。 この例については、425 ページの『均一クラスターの新規作成』 を参照してください。

注: これらのオーバーライドは、キュー・マネージャーの始動時にのみ適用されるので、キュー・マネージャーの作成には影響を与えられません。 例えば、この機能を使用して 1 次ログ・ファイルの数を設定することはできません。

始める前に

以下を使用できます。

1. 単一のファイルで、qm.ini ファイルに対する変更を含むテキスト・ファイルを作成します。
2. qm.ini フォーマット・ファイルのセット:
   - 構成が存在するディレクトリーを指定する。
   - そのディレクトリーでファイルを作成し、それぞれに拡張子.ini を付けてください。例えば、qminisettings.ini などです。

ファイルには、変更する項目のスタンザおよび **attribute**=*value* 設定のみを含める必要があります。 例えば、Channels スタンザの **MaxChannels** 属性を更新する場合は、以下のようなファイルになります。

```
Channels:
 MaxChannels=1234
```

qm.ini オーバーライド・ファイルでは、# から始まる行はコメントとして扱われます。

qm.ini ファイル属性の自動構成の有効化

新しいキュー・マネージャーを構成するには、 **crtmqm** コマンドに対して **-ii** フラグを使用し、特定のファイルまたはディレクトリーを指定します。 指定された値は、属性 **IniConfig** として **AutoConfig** スタンザの下の qm.ini ファイルに保管されます。

有効なファイルまたはディレクトリーを指す **AutoConfig** スタンザ属性 **IniConfig** を追加することにより、自動 MQSC 構成を有効にするように既存のキュー・マネージャーを構成できます。 以下に例を示します。

```
AutoConfig:
   IniConfig=C:\MQ_Configuration\uniclus.ini
```

自動構成の仕組み

キュー・マネージャーの始動時に、AutoConfig スタンザ属性 **IniConfig** によって識別される構成が検証され、有効な構文が確認されてから、キュー・マネージャーのデータ・ツリーで単一の cached.ini ファイルとして autocfg ディレクトリーに保管されます。

ディレクトリー内の複数のファイルが処理される場合、それらのファイルはアルファベット順に処理されます。

キュー・マネージャーの最初の始動時にファイルまたはディレクトリーを読み取れないと、キュー・マネージャーは始動できません。 この場合は、該当するエラー・メッセージがコンソール、およびキュー・マネージャーのエラー・ログの両方に出力されます。

2 回目以降の始動時に、指定されたファイルまたはディレクトリーを読み取れなかった場合は、前にキャッシュされたファイルが使用され、 キュー・マネージャーのエラー・ログに、そのことを示すメッセージが書き込まれます。

strmqm コマンドを使用する場合は、キュー・マネージャーが呼び出される前に、cached.ini ファイルの内容がオーバーライドとして qm.ini ファイルに適用されます。

つまり、スタンバイ・キュー・マネージャーでは、キュー・マネージャーがアクティブになるときではなく、**strmqm** コマンドが処理されるときに設定が読み取られることになります。

置換 qm.ini ファイルが構築される方法

自動初期設定構成が初めて構成され、キュー・マネージャーが始動すると、現行の qm.ini ファイルのコピーが、base_qm.ini としてキュー・マネージャーのデータ・ディレクトリー内の autoconfig サブディレクトリーにコピーされます。 これが、これ以降のベースラインと見なされます。

キュー・マネージャーが始動するたびに、つまり **strmqm** が実行されるたびに、現行のアクティブな qm.ini ファイルが破棄されて、base_qm.ini のコピーに置き換えられます。 その後、cached.ini ファイルからの構成がこのファイルに適用されます。

キュー・マネージャーが自動構成制御下に置かれると、qm.ini ファイルに対するすべての変更は、AutoConfig スタンザの **IniConfig** 属性を使用して指定する 1 つ以上のファイルを介して実行する必要があります。

キュー・マネージャーの始動時に既存の qm.ini ファイルは削除されるので、 **IniConfig** 属性を使用して提供した qm.ini ファイル内の構成のみが、キュー・マネージャーのベースラインに適用されます。

前にキュー・マネージャーが始動したときに自動初期設定構成によってスタンザまたは属性が変更されていた場合、それらの変更は削除されます。 ただし、**IniConfig** 属性で指定した 1 つ以上のファイルにも指定されている変更は別です。

キュー・マネージャーの開始時に、qm.ini ファイルを再作成することにより、qm.ini ファイルに対する手動変更が失われることを意味します。 実際に変更を永続的なものにする必要があり、**IniConfig** 属性を使用してその変更を行うことができない場合は、以下のいずれかを実行できます。

- base_qm.ini ファイルに変更を加えます。
- base_qm.ini ファイルを削除します。

  このファイルを削除すると、base_qm.ini は、qm.ini ファイルの現在の内容に基づいて、次のキュー・マネージャーの開始時に再作成されます。 これにより、現在のすべての変更が、将来の始動のための新しいベースラインとして「固定化」されます。

関連概念

101 ページの『qm.ini ファイルのスタンザと属性の要約』
キュー・マネージャー構成ファイル qmi.ini のスタンザの属性の要約。詳細情報へのリンクが含まれています。

Multi qm.ini ファイルのスタンザと属性の要約

キュー・マネージャー構成ファイル qmi.ini のスタンザの属性の要約。詳細情報へのリンクが含まれています。

表 *11. qm.ini* ファイルのスタンザ

スタンザと属性	属性の説明
Windows **AccessMode スタンザ**	
Windows アクセス・グループ [1]	Windows のセキュリティー・グループ。このグループのメンバーには、 キュー・マネージャーのすべてのデータ・ファイルに対する全アクセス権限が付与されます。
ApiExitLocal スタンザ	
名前	MQAXP 構造体の ExitInfoName フィールドで、API 出口に渡される API 出口の記述名。
機能	API 出口コードを含むモジュールへの関数エントリー・ポイントの名前。
モジュール	API 出口コードを含むモジュール。
データ	MQAXP 構造体の ExitData フィールドで API 出口に渡されるデータ。
Sequence	その他の API 出口に関してこの API 出口が呼び出される順序。
V 9.3.4 Linux AIX **AuthToken スタンザ**	
KeyStore	トラステッド発行者の公開鍵証明書または対称鍵を含む鍵ストアのファイル・パス。
KeyStorePwdFile	鍵ストアの暗号化されたパスワードを含むファイルのファイル・パス。
CertLabel	認証トークンの検証に使用される、鍵ストア内の公開鍵証明書または対称鍵の証明書ラベル。
UserClaim	キュー・マネージャーが許可検査に採用できるユーザー ID 情報を含むトークン内のクレーム。
AllowOSGroups	この属性は、借用ユーザーのグループ・メンバーシップを検査するかどうかを決定します。
AutoCluster スタンザ	
タイプ	自動クラスターのタイプ。 有効なオプションは、均一クラスターを表す Uniform のみです。

表 11. qm.ini ファイルのスタンザ (続き)	
スタンザと属性	**属性の説明**
ClusterName	自動クラスターの名前。
RepositoryName1	自動クラスター内の最初の完全リポジトリーのキュー・マネージャー名。
Repository1Conname	自動クラスターのメンバーがキュー・マネージャーに接続する方法を示す接続名 (CONNAME) の値。
RepositoryName2	自動クラスター内の 2 番目の完全リポジトリーのキュー・マネージャー名。
Repository2Conname	自動クラスターのメンバーがキュー・マネージャーに接続する方法を示す接続名 (CONNAME) の値。
AutoConfig スタンザ	
MQSCConfig 構成	キューマネージャの起動ごとに、すべての`*.mqsc` ファイルのフルパスまたはディレクトリーへのパスのいずれかファイルがキュー・マネージャーに適用される。
IniConfig	完全なファイルパス、または `qm.ini` ディレクトリへのパスのどちらかで、キューマネージャの起動ごとにすべての `*.ini` ファイルが適用されます。
Channels スタンザ	
MaxChannels	使用できる現行チャネルの最大数。
MaxActiveChannels	任意の時点でアクティブにできるチャネルの最大数。
MaxInitiators	開始プログラムの最大数。
MQIBindType	アプリケーションのバインディング。
IBM i ThreadedListener	RUNMQLSR または AMQCLMAA をリスナーとして起動するかどうか。
PipeLineLength	チャネルが使用する並行スレッドの最大数。
AdoptNewMCA	IBM MQ がチャネルの開始要求を受け取ったが、 そのチャネルのインスタンスが既に実行中であることが検出された場合に、新しいチャネル・インスタンスが開始できるように、 既存のチャネル・インスタンスを停止させることができるチャネルのタイプ。
AdoptNewMCATimeout	新しいチャネル・インスタンスが古いチャネル・インスタンスの終了を待つ時間 (秒)。
AdoptNewMCACheck	**AdoptNewMCA** 属性を使用可能にするときに必要な検査のタイプ。
ChlauthEarlyAdopt	接続認証規則とチャネル認証規則の処理順序。
PasswordProtection	チャネルが TLS 暗号化を使用しない場合に、アプリケーションによって指定された資格情報を MQCSP パスワード保護で保護する必要があるかどうか。
IgnoreSeqNumberMismatch	チャネル始動時にキュー・マネージャーがシーケンス番号の不一致を処理する方法を制御します。
Connection スタンザ	

表 11. qm.ini ファイルのスタンザ (続き)	
スタンザと属性	**属性の説明**
DefaultBindType	別々のプロセスで実行されているアプリケーションとキュー・マネージャーの間で、一部のリソースを共有するか、リソースをまったく共有しないか。
DiagnosticMessages スタンザ	
name	スタンザの名前。
サービス	このスタンザで有効にするサービス。
ExcludeMessage	キュー・マネージャーのエラー・ログに書き込まれないメッセージ。
SuppressMessage	指定した時間内に 1 回のみ、キュー・マネージャーのエラー・ログに書き込まれるメッセージ。
IBM i SuppressInterval	秒単位の時間。**SuppressMessage** に指定したメッセージは、この時間内に 1 回しかキュー・マネージャーの エラー・ログに書き込まれません。
Severities	重大度レベルをコンマで区切ったリスト。
FilePath	ログ・ファイルの書き込み先のパス。 (Service 属性を File に設定した場合にのみ サポートされます。)
FilePrefix	ログ・ファイルの接頭部。 (Service 属性を File に設定した場合にのみ サポートされます。)
FileSize	ログがロールオーバーするサイズ。 (Service 属性を File に設定した場合にのみ サポートされます。)
形式	ファイルの形式。 (Service 属性を File に設定した場合にのみ サポートされます。)
Linux AIX Syslog	JSON 形式の診断メッセージ仕様を使用して、 フィルタリングされていないすべてのメッセージを syslog に送信する Syslog サービス。
Linux AIX Ident	syslog 項目に関連付けられた ident 値。 (Service 属性を Syslog に設定した場合にのみ サポートされます。)
ExitPath スタンザ	
ExitsDefaultPath	キュー・マネージャー・システム (32 ビット) 上のユーザー出口プログラムのパス。
ExitsDefaultPath64	キュー・マネージャー・システム (64 ビット) 上のユーザー出口プログラムのパス。
ExitPropertiesLocal スタンザ	
CLWLMode	クラスター・ワークロード (CLWL) 出口を FAST モードと SAFE モードのどちらで実行するか。
Linux IBM i AIX **Filesystem スタンザ**	
Linux IBM i AIX ValidateAuth	mqm グループのメンバーでないユーザーがエラーのディレクトリーとファイルに アクセスできるようにします。
Log スタンザ	

表 11. qm.ini ファイルのスタンザ (続き)	
スタンザと属性	**属性の説明**
LogPrimaryFiles	キュー・マネージャーの作成時に割り振られるログ・ファイル。
LogSecondaryFiles	1 次ファイルが足りなくなったときに割り振られるログ・ファイル。
LogFilePages	ログ・ファイルのページ数。 (ログ・ファイル・サイズは、4 KB ページ単位で指定します。)
LogType	キュー・マネージャーで使用するロギングのタイプ (循環またはリニア)。
LogBufferPages	書き込み用としてバッファー・レコードに割り振るメモリーの量。バッファーのサイズは 4 KB ページ単位で指定します。
LogPath	キュー・マネージャーのログ・ファイルが置かれるディレクトリー。
LogWriteIntegrity	高信頼ログ・レコードを書き込むためにロガーが使用する方式。
LogManagement	ログ・エクステントを手動またはキュー・マネージャーによって管理するために使用される方式。
Windows **LU62 スタンザ**	
Windows TPName	リモート・サイトで始動する TP 名。
Windows Library1	APPC DLL の名前。
Windows Library2	Library1 と同様に、コードが 2 つの別個のライブラリーに格納されている場合に使用されます。
CP4I **NativeHAInstance stanza**	
136 ページの『名前』	キュー・マネージャー・インスタンスが作成されたときに使用されたインスタンス名。
136 ページの『ReplicationAddress』	インスタンスのホスト名、小数点付き 10 進数 IPv4、または小数点付き 16 進数 IPv6 アドレス。
CP4I **NativeHALocalInstance stanza**	
136 ページの『LocalName』	NativeHALocalInstance スタンザの名前。ネイティブ HA キュー・マネージャーの作成時に指定されたログ・レプリカ・インスタンス名から取得されます。
137 ページの『KeyRepository』	ログ複製トラフィックの保護のために、使用するデジタル証明書を保持する鍵リポジトリーのロケーション。
137 ページの『CertificateLabel』	ログ複製トラフィックの保護に使用するデジタル証明書を識別する証明書ラベル。
137 ページの『CipherSpec』	ログ複製トラフィックを保護するために使用する MQCipherSpec。
137 ページの『LocalAddress』	ログ複製トラフィックを受け入れるローカル・ネットワーク・インターフェース・アドレス。

表 11. qm.ini ファイルのスタンザ (続き)	
スタンザと属性	**属性の説明**
137 ページの『HeartbeatInterval』	ハートビート間隔は、ネイティブ HA キュー・マネージャーのアクティブ・インスタンスがネットワーク・ハートビートを送信する頻度をミリ秒単位で定義します。
137 ページの『HeartbeatTimeout』	ハートビート・タイムアウトは、ネイティブ HA キュー・マネージャーのレプリカ・インスタンスが、アクティブ・インスタンスが応答しないと判断するまで待機する時間の長さを定義します。
138 ページの『RetryInterval』	再試行間隔は、障害が発生した複製リンクがネイティブ HA キュー・マネージャーで再試行される頻度をミリ秒単位で定義します。
Windows **NETBIOS スタンザ**	
Windows LocalName	このマシンが LAN 上で識別される名前。
Windows AdapterNum	LAN アダプターの番号。
Windows NumSess	割り振るセッションの数。
Windows NumCmds	割り振るコマンドの数。
Windows NumNames	割り振る名前の数。
Windows Library1	NetBIOS DLL の名前。
QMErrorLog スタンザ	
ErrorLogSize	キュー・マネージャーのエラー・ログがバックアップにコピーされるときのサイズを指定します。
ExcludeMessage	キュー・マネージャーのエラー・ログに書き込まれないメッセージを指定します。
SuppressMessage	指定した時間間隔内で 1 回だけキュー・マネージャーのエラー・ログに書き込まれるメッセージを指定します。
SuppressInterval	SuppressMessage に指定したメッセージを 1 回だけキュー・マネージャーのエラー・ログに書き込む時間間隔を秒単位で指定します。
Linux AIX **制限モード・スタンザ (RestrictedMode スタンザ)** [2]	
Linux AIX ApplicationGroup	伝送キューが宛先について明示的に定義されなかったときにリモート・メッセージが置かれるローカル伝送キューの名前。
Security スタンザ	
ClusterQueueAccessControl	クラスター・キュー・マネージャーでホストされているクラスター・キューまたは完全修飾キューのアクセス制御を検査します。
Windows GroupModel	オブジェクト権限マネージャー (OAM) が Windows のユーザーのグループ・メンバーシップを判別する際に、グローバル・グループを検査するかどうか。
サービス・スタンザ	

表 11. qm.ini ファイルのスタンザ (続き)	
スタンザと属性	**属性の説明**
名前	必要なサービスの名前。
EntryPoints	このサービス用として定義するエントリー・ポイントの数。
Windows SecurityPolicy	Windows のキュー・マネージャーごとのセキュリティー・ポリシー
Linux AIX SecurityPolicy	AIX and Linux のキュー・マネージャーがユーザー・ベースと グループ・ベースのどちらの権限を使用するか。 V 9.3.0 IBM MQ 9.3.0 以降は、稼働していないシステムのシステム・ユーザー名も作成できるようになりました。
SharedBindingsUserId	共有バインディングの場合にのみ、MQZ_AUTHENTICATE_USER 関数の IdentityContext 構造体の UserIdentifier フィールドが 有効ユーザー ID と実ユーザー ID のどちらであるか。
FastpathBindingsUserId	ファースト・パス・バインディングの場合にのみ、MQZ_AUTHENTICATE_USER 関数の IdentityContext 構造体の UserIdentifier フィールドが 有効ユーザー ID と実ユーザー ID のどちらであるか。
IsolatedBindingsUserId	分離されたバインディングの場合にのみ、MQZ_AUTHENTICATE_USER 関数の IdentityContext 構造体の UserIdentifier フィールドが 有効ユーザー ID と実ユーザー ID のどちらであるか。
ServiceComponent スタンザ	
サービス	必要なサービスの名前。
名前	サービス・コンポーネントの記述名を指定します。
モジュール	このコンポーネントのコードを含むモジュールの名前。
ComponentDataSize	各呼び出しでコンポーネントに渡される コンポーネント・データ域のバイト単位のサイズ。
Windows **SPX スタンザ**	
Windows Socket	16 進表記の SPX ソケット番号。
Windows BoardNum	LAN アダプターの番号。
Windows KeepAlive	KeepAlive 機能のオン/オフを切り替えます。
Windows Library1	SPX DLL の名前。
Windows Library2	LibraryName1 と同じもので、コードが 2 つの異なるライブラリーに保管されている場合に使用されます。
Windows ListenerBacklog	SPX リスナーの場合のデフォルトの未解決要求数を指定変更します。
SSL スタンザ	
V 9.3.0 OutboundSNI	SNI 対応クライアントが、TLS 接続の開始時に、SNI をリモート・システムへのターゲット IBM MQ チャネル名に設定するか、ホスト名に設定するかを指定します。

表 11. qm.ini ファイルのスタンザ (続き)	
スタンザと属性	**属性の説明**
AllowOutboundSNI	SNI 対応クライアントが、TLS 接続の開始時に、SNI をリモート・システムへのターゲット IBM MQ チャネル名に設定するかどうかを指定します。 **重要 :** Deprecated V 9.3.0 IBM MQ 9.3.0 以降、このプロパティーは非推奨になりました。 代わりに **OutboundSNI** を使用してください。
AllowedCipherSpecs	Multiplatforms の IBM MQ チャネルで順序付けされ、使用可能な CipherSpec のカスタム・リストを指定します。
AllowTLSV13	キュー・マネージャーが TLS 1.3 CipherSpec を使用できるかどうか。
CDPCheckExtensions	このキュー・マネージャーの TLS チャネルが、CrlDistributionPoint 証明書拡張で指定されている CDP サーバーの 検査を試行するかどうか。
MinimumRSAKeySize	RSA 証明書を受け入れるために必要な最小の鍵サイズを指定します。
OCSPAuthentication	失効状況を OCSP サーバーで判別できない場合に実行するアクション。
OCSPCheckExtensions	このキュー・マネージャーの TLS チャネルが、AuthorityInfoAccess 証明書拡張で指定されている OCSP サーバーの 検査を試行するかどうか。
OCSPTimeout (OCSPTimeout)	失効検査の実行時に OCSP 応答側を待機する秒数。
ALW PeerCertChainValidation	IBM Global Security Kit (GSKit) 証明書の妥当性検査設定。
SSLHTTPProxyName	GSKit が OCSP 検査に使用する HTTP プロキシー・サーバーのホスト名またはネットワーク・アドレスのいずれか。
SSLHTTPConnectTimeout	失効検査の実行時に HTTP サーバーへのネットワーク接続が正常に確立されるのを待機する秒数。
サブプール・スタンザ 109 ページの『3』	このスタンザは、IBM MQ によって作成されます。 変更はしないでください。
ShortSubpoolName 109 ページの『3』	/var/mqm/sockets ディレクトリー内に作成されたディレクトリーおよびシンボリック・リンクに対応する名前。 IBM MQ は、実行中のプロセス間の内部通信に、この名前を使用します。
TCP スタンザ	
ポート	TCP/IP セッション用の 10 進表記のデフォルトのポート番号。
Windows Library1	TCP/IP ソケット DLL の名前。
KeepAlive	KeepAlive 機能のオン/オフを切り替えます。
ListenerBacklog	TCP/IP リスナーの場合のデフォルトの未解決要求数を指定変更します。
Connect_Timeout	ソケットへの接続試行がタイムアウトになるまでの秒数。

表 11. qm.ini ファイルのスタンザ (続き)	
スタンザと属性	**属性の説明**
SndBuffSize	チャネルの送信側で使用される TCP/IP 送信バッファーのサイズ (バイト)。
RcvBuffSize	チャネルの受信側で使用される TCP/IP 受信バッファーのサイズ (バイト)。
RcvSndBuffSize	受信側チャネルの送信側で使用される TCP/IP 送信バッファーのサイズ (バイト)。
RcvRcvBuffSize	受信側チャネルの受信側で使用される TCP/IP 受信バッファーのサイズ (バイト)。
SvrSndBuffSize	クライアント接続チャネルとサーバー接続チャネルのサーバー側で使用される TCP/IP 送信バッファーのサイズ (バイト)。
SvrRcvBuffSize	クライアント接続チャネルとサーバー接続チャネルのサーバー側で使用される TCP/IP 受信バッファーのサイズ (バイト)。
V 9.3.0 Multi SecureCommsOnly	プレーン・テキスト通信を許可するか、デフォルト値を使用するか、許可しないかを指定します。
チューニング・パラメーター・スタンザ	
SuppressDspAuthFail	接続にオブジェクトへの+dsp 権限がない場合、キューマネージャが認証チェックに失敗したときの認証イベントの生成とエラーログへの AMQ8077 エラーメッセージの書き込みを抑止するかどうかを指定します。
ImplSyncOpenOutput	書き込みのためにキューを開いているアプリケーションの最小数。これより多ければ、 同期点以外で、永続書き込みのために暗黙的同期点を使用できることがあります。
UniformClusterName	均一クラスターとして使用している IBM MQ クラスターの名前。
OAMLdapConnectTimeout	LDAP クライアントがサーバーへの TCP 接続を確立するために待機する最大時間(秒単位)。
OAMLdapQueryTimeLimit	LDAP クライアントが LDAP 要求に対する応答をサーバーから受信するまでに待機する最大時間(秒)。
V 9.3.2 OAMLdapResponseWarningTime	LDAP サーバーへの接続にかかった時間が、**OAMLdapResponseWarningTime** パラメーターで指定されたしきい値 (秒数) を超えた場合、 AMQ5544W メッセージがエラー・ログに書き込まれます。
ExpiryInterval	キュー・マネージャーがキューをスキャンして、他のキュー・アクティビティーによってまだクリーンアップされていない有効期限切れメッセージを探す頻度を示します。これは秒単位の時間間隔です。
LivenessHeartBeatLen	ログへの書き込みを行うキュー・マネージャー検査が適切な頻度で行われる頻度を構成します。
ECHeartBeatLen	キュー・マネージャーの一般的なヘルス・チェックの頻度を構成します。

表 11. qm.ini ファイルのスタンザ (続き)	
スタンザと属性	**属性の説明**
FileLockHeartBeatLen	実行コントローラーが定期的に実行する複数インスタンス・キュー・マネージャーのファイル・ロック検査のデフォルト値を変更して、1 次複数インスタンス・ファイルの排他ロックを保持していることを確認します。
Variables スタンザ	
属性 = 値	MQSC の定義中に挿入するために使用する名前と関連値。
XAResourceManager スタンザ	
名前	リソース・マネージャー・インスタンス。
SwitchFile	リソース・マネージャーの XA スイッチ構造を収容するロード・ファイルの完全修飾名。
XAOpenString	リソース・マネージャーの xa_open エントリー・ポイントに渡されるデータのストリング。
XACloseString	リソース・マネージャーの xa_close エントリー・ポイントに渡されるデータのストリング。
ThreadOfControl	キュー・マネージャーが、自身のマルチスレッド・プロセスの 1 つからリソース・マネージャーを呼び出す 必要がある場合に、シリアライゼーション (直列化) に使用する値。Windows の場合は必須。

注 :

1. AccessMode スタンザは、**crtmqm** コマンドの **-a [r]** オプションによって設定されます。 キュー・マネージャーの作成後には、AccessMode スタンザを変更しないでください。
2. RestrictedMode スタンザは、**crtmqm** コマンドの **-g** オプションによって設定されます。 キュー・マネージャーの作成後には、このスタンザを変更しないでください。 **-g** オプションを使用しない場合、スタンザは qm.ini ファイルに作成されません。
3. Subpool スタンザおよびこのスタンザ内の属性 ShortSubpoolName は、キュー・マネージャーの作成時に、 IBM MQ によって自動的に書き込まれます。 IBM MQ は ShortSubpoolName の値を選択します。 この値は変更しないでください。

Windows qm.ini ファイルの AccessMode スタンザ

アクセス・モードは Windows サーバーのみに適用されます。 qm.ini ファイルの AccessMode スタンザは、**crtmqm** コマンドの -a [r] オプションで設定します。 キュー・マネージャーの作成後には、AccessMode スタンザを変更しないでください。

アクセス・グループの使用 (-a [r]) **crtmqm** コマンドのオプション。 Windows セキュリティー・グループを指定します。このグループのメンバーには、すべてのキュー・マネージャー・データ・ファイルに対する全アクセス権限が付与されます。 このグループは、使用する構文に応じて、ローカル・グループかグローバル・グループのいずれかになります。 グループ名の有効な構文は次のとおりです。

LocalGroup
Domain name ¥ *GlobalGroup name*
GlobalGroup name@*Domain name*

-a [r] オプションを指定して **crtmqm** コマンドを実行する前に、追加のアクセス・グループを定義する必要があります。

-a の代わりに -ar を使用してグループを指定すると、ローカル mqm グループはキュー・マネージャー・データ・ファイルへのアクセス権を付与されません。 キュー・マネージャー・データ・ファイルをホスト

するファイル・システムが、ローカルに定義されたグループのアクセス制御項目をサポートしていない場合は、このオプションを使用してください。

このグループは通常はグローバル・セキュリティー・グループです。グローバル・セキュリティー・グループは、複数インスタンス・キュー・マネージャーに、キュー・マネージャーの共有データ・フォルダーと共有ログ・フォルダーに対するアクセス権を付与するために使用されます。 このような追加のセキュリティー・アクセス・グループを使用すれば、キュー・マネージャーのデータ・ファイルとログ・ファイルが含まれているフォルダーや共有フォルダーに対する読み取り/書き込み権限を設定できます。

キュー・マネージャーのデータとログが含まれているフォルダーに対する権限を設定するために、追加のセキュリティー・アクセス・グループを、mqm という名前のローカル・グループを使用する代わりに使用することもできます。 ローカル・グループ mqm とは異なり、追加のセキュリティー・アクセス・グループをローカル・グループまたはグローバル・グループにすることができます。 複数インスタンス・キュー・マネージャーが使用するデータ・ファイルとログ・ファイルが含まれている共有フォルダーに対する権限を設定する場合は、グローバル・グループを使用する必要があります。

Windows オペレーティング・システムは、キュー・マネージャーのデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限を検査します。 検査の対象になるのは、キュー・マネージャーのプロセスを実行しているユーザー ID の権限です。 検査対象になるユーザー ID は、キュー・マネージャーをサービスとして開始したか、それとも対話式に開始したかによって異なります。 キュー・マネージャーをサービスとして開始した場合、Windows システムによって検査されるユーザー ID は、「IBM MQ の**準備**」ウィザードで構成したユーザー ID です。 キュー・マネージャーを対話式に開始した場合、Windows システムによって検査されるユーザー ID は、**strmqm** コマンドを実行したユーザー ID です。

キュー・マネージャーを開始するには、ユーザー ID がローカル mqm グループのメンバーでなければなりません。 そのユーザー ID が追加のセキュリティー・アクセス・グループのメンバーになっていれば、キュー・マネージャーで、そのグループに基づいて権限が与えられているファイルを読み書きすることが可能になります。

制約事項 : 追加のセキュリティー・アクセス・グループを指定できるのは、Windows オペレーティング・システムだけです。 他のオペレーティング・システムで追加のセキュリティー・アクセス・グループを指定すると、**crtmqm** コマンドでエラーが返されます。

スタンザの例

```
AccessMode:
SecurityGroup=wmq\wmq
```

関連概念

539 ページの『Windows でキュー・マネージャーのデータとログの非共有ディレクトリーおよびファイルを保護する操作』
536 ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 (Windows)』

関連タスク

510 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 (Windows)』

関連資料

crtmqm (キュー・マネージャーの作成)

Multi qm.ini ファイルの ApiExitLocal スタンザ

ApiExitLocal スタンザは、キューマネージャの API 終了ルーチンを指定します。

サーバでは、キューマネージャの API 終了ルーチンを特定するために、'qm.ini ファイルの ApiExitLocal スタンザを変更してください。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer Exits キュー・マネージャーのプロパティー・ページを使用します。

クライアントの場合、キューマネージャの API 終了ルーチンを特定するために、 'mqclient.ini ファイル内の ApiExitLocal スタンザを変更する。

概要

ApiExitLocal スタンザでは、単一の Module のみを指定できます。 また、以下のように 4 つのモジュールを提供する必要があります。

- 32 ビット、スレッドなし
- 32 ビット、スレッドあり
- 64 ビット、スレッドなし
- 64 ビット、スレッドあり

IBM MQ は提供されるモジュール名に _r を付加して、スレッドのあるバージョンの出口を識別しますが、IBM MQ は 32 ビットと 64 ビットのバリアントに関しては、同等のメカニズムを直接提供していないことに注意してください。

prefix/mqm/samp/bin に同梱されている amqsaxe0 および amqsaxe0_r のバージョンは、作成されたプラットフォーム上のキュー・マネージャーのネイティブ・サイズ (現在はすべて 64 ビット) 用に作成され、 同じネイティブ・サイズで実行されているアプリケーションでのみ使用できます。

非修飾のモジュール名が提供されている場合は、/var/mqm/exits と /var/mqm/exits64 で IBM MQ32 ビットバリアントと 64 ビットバリアントを検索します

例えば、module=amqsaxe は以下を暗黙に示します。

```
/var/mqm/exits/amqsaxe  - 32 bit unthreaded variant
/var/mqm/exits/amqsaxe_r - 32 bit threaded variant
/var/mqm/exits64/amqsaxe - 64 bit unthreaded variant
/var/mqm/exits64/amqsaxe_r - 64 bit threaded variant
```

Windows Windows システムでは、API 出口の項目を変更するために、**amqmdain** コマンドを使用することもできます。 (すべてのキュー・マネージャーの API 出口ルーチンを識別するには、92 ページの『mqs.ini ファイルの ApiExitCommon スタンザおよび ApiExitTemplate スタンザ』に説明されているとおりに、ApiExitCommon および ApiExitTemplate スタンザを使用します。)

API 出口が正しく機能するためには、サーバーからのメッセージを変換されていない状態でクライアントに送信する必要があるので、注意してください。 そのメッセージは、API 出口がメッセージを処理した後に、クライアントで変換される必要があります。 そのため、すべての変換出口がクライアントにインストールされている必要があります。

これらの属性の使用方法について詳しくは、API 出口の構成を参照してください。

パラメーター

Name=ApiExit_name
MQAXP 構造体の ExitInfoName フィールドで、API 出口に渡される API 出口の記述名。

この名前は一意で、48 文字以内であり、さらに IBM MQ オブジェクト名 (例えばキュー名) に有効な文字のみを含む必要があります。

Function=function_name
API 出口コードを含むモジュールへの関数エントリー・ポイントの名前。 このエントリー・ポイントは MQ_INIT_EXIT 関数です。

このフィールドの長さは MQ_EXIT_NAME_LENGTH に限定されます。

Module=module_name
API 出口コードを含むモジュール。

このフィールドにモジュールの絶対パス名が入っている場合、それがそのまま使用されます。 このフィールドにモジュール名のみが入っている場合、モジュールは、 qm.ini ファイルの ExitPath スタンザ内の **ExitsDefaultPath** 属性を使用して配置されます。

異なるスレッド化ライブラリーをサポートするプラットフォームで、API 出口モジュールの非スレッド化バージョンとスレッド化バージョンの両方を提供する必要があります。 スレッド化バージョンは、接尾部 _r を持っている必要があります。 IBM MQ アプリケーション・スタブのスレッド化バージョンは、指定のモジュールがロードされる前にその名前に _r を暗黙的に追加します。

このフィールドの長さは、プラットフォームがサポートする最大パス長に限定されます。

Data=data_name
MQAXP 構造体の ExitData フィールドで API 出口に渡されるデータ。

この属性を指定すると、先行および末尾のブランクが除去されて残りのストリングは 32 文字に切り捨てられ、その結果が出口に渡されます。 この属性を省略すると、デフォルト値の 32 文字のブランクが出口に渡されます。

このフィールドの最大長は、32 文字です。

Sequence=sequence_number
その他の API 出口に関してこの API 出口が呼び出される順序。 小さなシーケンス番号の出口は、より大きなシーケンス番号の出口よりも先に呼び出されます。 出口のシーケンス番号は連続である必要はありません。 つまり、1、2、3 の順序は、7、42、1096 の順序と同じ結果となります。 2 つの出口のシーケンス番号が同じ場合は、キュー・マネージャーが最初に呼び出す出口を決定します。 MQAXP の ExitChainAreaPtr で示される ExitChainArea に時刻またはマーカーを設定するかまたは独自のログ・ファイルを作成して、イベントの後に呼び出された出口を判別できます。

この属性は、符号なし数値です。

スタンザの例

```
ApiExitLocal:
   Name=ClientApplicationAPIchecker
   Sequence=3
   Function=EntryPoint
   Module=/usr/Dev/ClientAppChecker
   Data=9.20.176.20
```

V 9.3.4 Linux AIX qm.ini ファイルの AuthToken スタンザ

AuthToken スタンザを使用して、接続アプリケーションによって提供される認証トークンを検証するようにキュー・マネージャーを構成します。

AuthToken スタンザ

KeyStore= ストリング
トラステッド発行者の公開鍵証明書および対称鍵を含む鍵ストアのファイル・パス。 既存の鍵ストアに鍵を追加することも、新しい鍵ストアを作成することもできます。 詳しくは、 認証トークンを受け入れるためのキュー・マネージャーの構成を参照してください。 キュー・マネージャーは、鍵ストア内の鍵を使用して、アプリケーションが提示する認証トークンがトラステッド発行者によって署名されていることを検証します。

ファイル拡張子が .kdb の CMS 鍵ストア、またはファイル拡張子が .p12 の PKCS#12 鍵ストアのいずれかを使用できます。 鍵ストア・ファイルが存在しないか、アクセスできない場合、「AMQ7076E: ini ファイルの属性の値が無効です (Invalid value for attribute in ini file)」というエラーがキュー・マネージャーのエラー・ログに出力されます。

鍵ストア・タイプが、鍵ストアのファイル名拡張子と一致していることを確認してください。 IBM MQ は、鍵ストアの正しい形式を検出します。不整合があると、鍵ストア・タイプとファイル名拡張子が一致しない場合に、他の管理上の問題が発生する可能性があります。

鍵ストア・ファイル・パスの最大長は 256 文字です。

KeyStorePwdFile= ストリング
鍵ストアの暗号化されたパスワードを含むファイルのファイル・パス。 ファイルには、暗号化されたパスワードが 1 行のテキストとして含まれている必要があります。 プレーン・テキストのパスワードは受け入れられません。

パスワードを鍵ストア・パスワード・ファイルに保存する前に、 **runqmcred** コマンドを使用してパスワードを暗号化します。 鍵ストア・パスワード・ファイルには、 **runqmcred** コマンドを実行して作成された暗号化パスワードのみが含まれている必要があります。

暗号化される前のプレーン・テキスト・パスワードの最大長は 1024 文字です。

このパラメーターはオプションです。 これが指定されていない場合、キュー・マネージャーは、パスワードを持つ stash ファイルを同じディレクトリー内で、鍵ストアと同じ名前 (ファイル拡張子は .sth) で検索します。 stash ファイルが見つからない場合、構成は拒否され、エラー・メッセージ AMQ7006E がキュー・マネージャーのエラー・ログに出力されます。 鍵ストア・パスワードを保管するためのオプションについて詳しくは、 鍵リポジトリー・パスワードの暗号化を参照してください。

パスワード・ファイル・パスの最大長は 256 文字です。

CertLabel= ストリング
認証トークンの検証に使用される、鍵ストア内の公開鍵証明書または対称鍵の証明書ラベル。 **CertLabel** 属性を繰り返すことにより、最大 32 個の証明書ラベルを指定できます。

キュー・マネージャーの鍵ストアに証明書を追加する場合は、それらの証明書に意味のあるラベルを付けます。 証明書ラベルには大/小文字の区別があります。 英数字、句読文字、およびスペースを含めることができます。 無効文字が検出されると、エラーが戻され、エラー・メッセージが IBM MQ エラー・ログに書き込まれます。

トラステッド・トークン発行者は、複数の公開鍵証明書と対称鍵を提供する場合があります。 例えば、公開鍵証明書には有効期間があります。 有効期限が近づいている場合、トークン発行者は、新しい有効期限日付を持つ新しい証明書を提供します。 しばらくの間、両方の証明書が有効である可能性があります。

アプリケーションが認証用のトークンを提示すると、トークンの署名に使用された有効な鍵が見つかるまで、 **CertLabels** のリストが検査されます。 一致するものが見つかると、トークン署名が検証されます。

CertLabel が指定されていない場合、トークンを提示するアプリケーションからの接続は理由コード 2063 MQRC_SECURITY_ERROR で失敗し、メッセージ「AMQ5786E: 認証トークン構成エラー」がキュー・マネージャーのエラー・ログに書き込まれます。

証明書ラベルの最大長は 64 文字です。

例:

```
AuthToken:
   KeyStore=/var/mqm/qmgrs/qmgrs/qm1/tokenissuer/key.kdb
   KeyStorePwdFile=/var/mqm/qmgrs/qm1/tokenissuer/key.pw
   CertLabel=token
   CertLabel=rsakey
   CertLabel=mark
   ... up to 32 CertLabel fields
```

UserClaim= ストリング
キュー・マネージャーが許可検査に採用するユーザー ID を含むトークン内のクレーム。

キュー・マネージャーが **ADOPTCTX(NO)** で構成されている場合、このパラメーターはオプションです。 **ADOPTCTX(YES)** を使用する場合、このパラメーターは必須です。 **ADOPTCTX** は、キュー・マネージャーの **CONNAUTH** 属性によって参照される認証情報 (AUTHINFO) オブジェクト内に存在する属性です。

ID を採用するには、**AuthToken** スタンザの **UserClaim** 属性に指定されている名前を持つクレームがトークンに含まれている必要があり、 **ADOPTCTX(YES)** を使用する必要があります。

例えば、トークンにクレーム "AppUser": "MyUserName"が含まれている場合、qm.ini ファイルの AuthToken スタンザに UserClaim=AppUser を指定して、許可のために ID "MyUserName" を採用する必要があります。

UserClaim 属性値の最大長は 128 文字です。

注: IBM MQ 9.3.4 以降、 AuthToken スタンザが指定されている場合、Service スタンザの **SecurityPolicy** 属性の有効値は UserExternal に設定されます。 Service スタンザで **SecurityPolicy** が明示的に Group に設定されている場合、トークン認証は使用できません。

SecurityPolicy が Group に設定されている場合は、Service スタンザから **SecurityPolicy** 属性を削除してから、キュー・マネージャーを再始動します。 詳しくは、SecurityPolicy を参照してください。

注 : 認証情報オブジェクトの **ADOPTCTX** 属性を使用して、トークン内のユーザー ID が許可検査に採用されるかどうかを制御します。 キュー・マネージャーを作成すると、この属性は **ADOPTCTX(YES)** に設定されます。 この値により、トークンからのユーザー ID が採用されます。 ユーザー ID は、認証トークン内のユーザー ID の要件を満たしている必要があります。 詳しくは、認証トークン内のユーザー ID を参照してください。 トークン・ユーザー・クレームに、要件を満たさないユーザー ID が含まれている場合、接続は理由コード **2035 MQRC_NOT_AUTHORIZED** で拒否されます。 **ADOPTCTX(NO)** が設定されている場合、トークンは認証にのみ使用され、許可には別のユーザーを使用する必要があります。

AllowOSGroups=NO (デフォルト) |YES
デフォルト値は NO です。 トークンから採用された ID がオペレーティング・システム (OS) ユーザーとして扱われるかどうか、および一致する OS ユーザーのグループ・メンバーシップが許可時に受け入れられるかどうかを決定します。

AllowOSGroups= NO | N
許可検査は、トークンから採用されたユーザーの名前にのみ基づいて行われます。

AllowOSGroups= YES | Y
許可検査はユーザーの名前に基づいて行われ、ユーザーが属する可能性のあるグループも検査されます。

スタンザの例-認証のみ

AuthToken スタンザは、以下の 2 つの最小必須パラメーターのみを使用して有効にすることができます。

- **KeyStore** ファイル・パス、および
- **CertLabel** 名。

```
AuthToken:
   KeyStore=/var/mqm/qmgrs/qmgrs/qm1/tokenissuer/key.kdb
   CertLabel=token
   ... up to 32 CertLabel fields
```

2 つの最小パラメーターのみを指定した場合は、以下のようになります。

- 鍵ストア・パスワード・ファイルが必要ないように、暗号化された鍵ストア・パスワードを持つ stash ファイル key.sth が存在している必要があります。
- 許可のために IBM MQ に渡されるユーザー名がトークンに含まれていません。 アプリケーションは接続して認証することができますが、アプリケーションが接続後に作業を行うための許可を提供するためには、別のメカニズムを設定する必要があります。

キュー・マネージャーの構成に応じて、許可に使用されるユーザー名は、MCA 規則を介してチャネルで定義されたユーザー名、またはサーバー上に存在し、権限を持つグループに属しているクライアント・アプリケーションの実行に使用されたユーザー名のいずれかになります。 トークンを使用する場合は、以下の点に留意してください。

- キュー・マネージャーが **UserExternal** モードになります。これは、キュー・マネージャーが実行されているオペレーティング・システムに存在しないユーザーを認証に使用できることを意味します。
- **AuthToken** qm.ini スタンザに **AllowOSGroups** オプションを含めない場合でも、デフォルトは No に設定されます。 したがって、 **UserClaim** を含めるが、 **AllowOSGroups**=Yes を指定しない場合、許可に採用されるトークン・ユーザーは、キュー・マネージャーが実行されているオペレーティング・システムに属する可能性のあるグループについては検査されません。

スタンザの例-認証と許可

すべての **AuthToken** パラメーターを定義できます。

- **KeyStore** ファイル・パス
- **KeyStorePwdFile** ファイル・パス
- **CertLabel** 名、
- **UserClaim** 名および
- **AllowOSGroups** オプション。

```
AuthToken:
   KeyStore=/var/mqm/qmgrs/qmgrs/QMJWT/ssl/key.kdb
   KeyStorePwdFile=/var/mqm/qmgrs/QMJWT/ssl/key.pw
   CertLabel=token
   CertLabel=rsakey
   CertLabel=mark
   ... up to 32 CertLabel fields
   UserClaim=AppUser
   AllowOSGroups=Y
```

使用可能なすべてのパラメーターを含めた場合は、以下のようになります。

- **runqmcred** コマンドを使用して、鍵ストアのパスワードを暗号化します。 ファイルに保存してから、**AuthToken** スタンザにファイル・パスを含めます。
- 認証トークン・ユーザー・クレーム内のユーザー名は、認証と許可の両方に使用されます。
  - トークン・ユーザーは、キュー・マネージャーが実行されているオペレーティング・システム上のユーザーとして存在する可能性があります。
  - ユーザー検査を使用可能にする認証情報オブジェクトを定義しました。
  - チャネル認証レコードをセットアップして、チャネル認証または MCA 規則に基づいて、IBM MQ オブジェクトと対話する許可を持つユーザーを採用します。

トークン・ユーザーを認証および許可するための戦略は、要件と、 IBM MQ キュー・マネージャーが既にどのように構成されているかによって異なります。 詳しくは、 認証トークンの処理を参照してください。

関連概念

トークンの処理

関連タスク

AuthTokens を受け入れるためのキュー・マネージャーの構成

アプリケーションでの認証トークンの使用

Multi qm.ini ファイルの AutoCluster スタンザ

AutoCluster スタンザは、キュー・マネージャーがクラスターが自動クラスターのメンバーであるかどうかの識別を開始し、クラスターの完全リポジトリーを識別できる場合に使用されます。

次の属性は、AutoCluster スタンザの必須の属性です。

Type=*Uniform*
自動クラスターのタイプを指定します。有効なオプションは、均一クラスターを表す *Uniform* のみです。

ClusterName=*<String>*
クラスターの名前 (自動クラスター名)。

次の属性は、AutoCluster スタンザのオプションの属性ですが、指定はペアで行う必要があります。

RepositoryName1 =< ストリング>
これは、自動クラスター内の最初の完全リポジトリーのキュー・マネージャー名です。 この名前は、このキュー・マネージャーの名前でも、別のキュー・マネージャーの名前でも構いません。

Repository1Conname=< 接続名ストリング>
これは、自動クラスターのメンバーがこのキュー・マネージャーに接続する方法を示す接続名 (CONNAME) の値です。

Repository2Name=< ストリング>
これは、自動クラスター内の 2 番目の完全リポジトリーのキュー・マネージャー名です。 この名前は、このキュー・マネージャーの名前でも、別のキュー・マネージャーの名前でも構いません。

Repository2Conname=< 接続名ストリング>
これは、自動クラスターのメンバーがこのキュー・マネージャーに接続する方法を示す接続名 (CONNAME) の値です。

スタンザの例

```
AutoCluster:
        Repository1Name=QM1
        Repository2Name=QM2
        Repository1Conname=127.0.0.1(1414)
        Repository2Conname=127.0.0.1(1415)
        ClusterName=UNIFORMCLUSTER1
        Type=Uniform
```

関連概念
412 ページの『自動アプリケーション・バランシング』
IBM MQ の均一クラスターによってクラスター内のアプリケーションの分散度を緊密に管理できるようになり、ランダム化や特定のキュー・マネージャーにアプリケーションを手動でピン留めする操作に依存しなくてよくなりました。その結果、自動アプリケーション・バランシングによるアプリケーションの分散度と可用性が大幅に高まっています。

関連タスク
425 ページの『均一クラスターの新規作成』
新しい均一クラスターの作成方法。

関連資料
429 ページの『自動クラスター構成の使用』
qm.ini コンフィギュレーション情報を変更することで、IBM MQ 自動コンフィギュレーションを有効にするように設定します。

Multi qm.ini ファイルの AutoConfig スタンザ

AutoConfig スタンザの属性は、均一クラスターのセットアップの一部として頻繁に使用されます。

注 : AutoCluster スタンザは、均一クラスターでのみ使用できます。

MQSCConfig=*<Path>*
このパスは、完全ファイル・パスまたはディレクトリーへのパスのいずれかです。すべての*.mqsc ファイルは、キュー・マネージャーの開始時にキュー・マネージャーに適用されます。

詳しくは、開始時の MQSC スクリプトからの自動構成を参照してください。

IniConfig=*<Path>*
このパスは、完全ファイル・パスまたはディレクトリーへのパスのいずれかです。すべての *.ini ファイルは、キュー・マネージャーの開始時に qm.ini ファイルに適用されます。

詳細については、99 ページの『始動時の qm.ini の自動構成』を参照してください。

V 9.3.0 **ConfigTimeout**
キュー・マネージャーが自動構成の完了を待機する時間の長さ (秒単位)。 この時間が経過すると、キュー・マネージャーは起動を続行し、アプリケーションの接続ができるようになります。

デフォルトの動作はタイムアウトなしです。 つまり、すべての自動構成コマンドが完了するまで、アプリケーションの接続にキュー・マネージャーを使用することはできません。

この属性を構成すべきではない理由は、単純に構成に時間がかかること、そしてアプリケーションに適用される構成が完了する前に、アプリケーションが接続できるようになる可能性がある (アプリケーションに必要なキューの作成など) からです。

スタンザの例

```
AutoConfig:
        MQSCConfig=/tmp/auto.mqsc
        IniConfig=/tmp/auto.ini
        ConfigTimeout=120
```

関連概念

412 ページの『自動アプリケーション・バランシング』
IBM MQ の均一クラスターによってクラスター内のアプリケーションの分散度を緊密に管理できるようになり、ランダム化や特定のキュー・マネージャーにアプリケーションを手動でピン留めする操作に依存しなくてよくなりました。その結果、自動アプリケーション・バランシングによるアプリケーションの分散度と可用性が大幅に高まっています。

関連タスク

425 ページの『均一クラスターの新規作成』
新しい均一クラスターの作成方法。

関連資料

429 ページの『自動クラスター構成の使用』
qm.ini コンフィギュレーション情報を変更することで、IBM MQ 自動コンフィギュレーションを有効にするように設定します。

Multi qm.ini ファイルの Channels スタンザ

Channels スタンザの属性は、チャネルの構成を決定します。

z/OS ただし、この情報は IBM MQ for z/OS には適用されません。

qm.ini ファイル内の CHANNELS スタンザを使用して、チャネルに関する情報を指定します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer Channels キュー・マネージャーのプロパティー・ページを使用します。

MaxChannels = 100 (デフォルト) |*number*
使用できる現行チャネルの最大数。

デフォルトは 100 です。

必要に応じて、**MaxChannels** を別の値に設定し、現行チャネルの最大数を制限することができます。IBM MQ Appliance では、デフォルト値は 999 999 999 です。この値は変更しないでください。

MaxActiveChannels= *MaxChannels_value*
任意のある時点でアクティブ にできるチャネルの最大数。 デフォルトは、**MaxChannels** 属性に指定されている値です。

MaxInitiators = 3 (デフォルト) |*number*
開始プログラムの最大数。 デフォルト値と最大値は 3 です。

MQIBindType=FASTPATH|STANDARD
アプリケーション用のバインディング方法。次のとおりです。

FASTPATH
チャネルは MQCONNX FASTPATH を使用して接続されます。つまり、エージェント・プロセスはありません。

STANDARD
チャネルは STANDARD を使用して接続されます。

IBM i **ThreadedListener=NO|YES**
RUNMQLSR（YES）または AMQCLMAA（NO）をリスナーとして起動するかどうか。

ThreadedListener=YES を指定すると、すべてのチャネルは単一ジョブのスレッドとして実行されます。この場合単一ジョブで使用可能なリソースへの接続数が制限されます。

ThreadedListener=NO を指定すると、非スレッド化リスナー (AMQCLMAA) はインバウンドの各 TCP/IP チャネルで新しいレスポンダー・ジョブ (AMQCRSTA) を開始します。 この手法の欠点は、新しい AMQCRSTA ジョブを開始するのが、RUNMQLSR ジョブ内でスレッドを開始するほどには速くないことです。 そのため、非スレッド化リスナーの接続時間はスレッド化リスナーのものより遅くなります。

PipeLineLength=1|*number*
チャネルが使用する並行スレッドの最大数。 デフォルトは 1 です。 1 より大きい値は 2 として扱われます。

パイプラインを使用する場合は、**PipeLineLength** が 1 より大きくなるようにチャネルの両端のキュー・マネージャーを構成してください。

注 : パイプラインは、TCP/IP チャネルでのみ有効です。

詳しくは、 マルチスレッド・サポート-パイプライン を参照してください。

AdoptNewMCA=NO（デフォルト）| SVR|SDR|RCVR|CLUSRCVR|ALL|FASTPATH
IBM MQ がチャネル開始要求を受け取ったときに、チャネルのインスタンスが既に実行中であることが検出された場合、新しいインスタンスを開始する前に、既存のチャネル・インスタンスを停止する必要がある場合もあります。 **AdoptNewMCA** 属性を使用すると、どの種類のチャネルをこの方法に従って終了可能にするかを制御できます。

特定のチャネル・タイプについて **AdoptNewMCA** 属性を指定すると、一致するチャネル・インスタンスがすでに実行中であるために新規チャネルが始動できない場合は、次のようになります。

1. 新規チャネルは、終了要求を出して既存のチャネルを停止しようとします。
2. 既存のチャネル・サーバーが、**AdoptNewMCATimeout** に指定されている待ち時間が満了するまでにこの要求に応じない場合は、既存のチャネル・サーバーのスレッドまたはプロセスが終了されます。
3. ステップ 2 の実行後も既存のチャネル・サーバーが終了していない場合は、**AdoptNewMCATimeout** の待ち時間が 2 回目に満了した時点で、IBM MQ はそのチャネルを終了し、CHANNEL IN USE エラーを返します。

AdoptNewMCA 機能は、サーバー、送信側、受信側、およびクラスター受信側の各チャネルに適用されます。 送信側チャネルまたはサーバー・チャネルの場合、受信側のキュー・マネージャーでは、特定の名前を持つチャネルの 1 つのインスタンスだけが実行可能です。 受信側チャネルまたはクラスター受信側チャネルの場合、受信側のキュー・マネージャーでは、特定の名前を持つチャネルの複数のインスタンスが実行可能です。 ただし、特定のリモート・キュー・マネージャーからは、ある時点で 1 つのインスタンスだけが実行可能です。

注 : リクエスター・チャネルまたはサーバー接続チャネルに対しては、**AdoptNewMCA** はサポートされません。

以下のリストから 1 つ以上の値を選んで、コンマまたはブランクで区切って指定します。

NO
AdoptNewMCA 機能が不要であることを意味します。 これはデフォルトです。

SVR
サーバー・チャネルが採用されます。

SDR
送信側チャネルが採用されます。

RCVR
受信側チャネルが採用されます。

CLUSRCVR
クラスター受信側チャネルが採用されます。

ALL
FASTPATH チャネル以外のすべてのチャネル・タイプが採用されます。

FASTPATH
該当チャネルが FASTPATH チャネルのときは、そのチャネルが採用されます。 これが起きるのは、AdoptNewMCA=RCVR,SVR,FASTPATH などのように 該当するチャネル・タイプも指定されている場合のみです。

重要! : FASTPATH チャネルの場合は、AdoptNewMCA 属性が予期しない動作をすることがあります。 FASTPATH チャネルの AdoptNewMCA 属性を使用可能にするときは、十分な注意が必要です。

AdoptNewMCATimeout=60（デフォルト）｜1 - 3600
新しいチャネル・インスタンスが古いチャネル・インスタンスの終了を待つ時間 (秒)。 1 から 3600 までの範囲の値を指定します。 デフォルト値は 60 です。

AdoptNewMCACheck=QM|ADDRESS|NAME|ALL
AdoptNewMCA 属性を使用可能にするときに必要な検査のタイプ。 可能な場合は、故意または不注意によりチャネルがシャットダウンされるのを防ぐために、すべての検査を実行してください。 最低限、チャネル名が一致することだけは確認してください。

QM、*NAME*、または *ALL* の場合はコンマまたはブランクで区切って、以下の 1 つ以上の値を指定します。

QM
キュー・マネージャー名が一致することを検査します。

QMID ではなく、キュー・マネージャー名自体が突き合わされることに注意してください。

ADDRESS
通信ソースの IP アドレスを検査します。 例えば、TCP/IP アドレス。

注 : コンマ区切りの CONNAME 値は、ターゲット・アドレスに適用されるため、このオプションには関係ありません。

複数インスタンス・キュー・マネージャーが hosta から hostb にフェイルオーバーした場合、そのキュー・マネージャーのすべてのアウトバウンド・チャネルは hostb のソース IP アドレスを使用します。 これが hosta と異なる場合、AdoptNewMCACheck=*ADDRESS* の突き合わせは失敗します。

相互認証に SSL または TLS を使用することで、実行中の既存のチャネルがアタッカーによって中断されるのを防止できます。 別の方法として、複数インスタンス・キュー・マネージャーの代わりに、IP テークオーバーを行う HACMP タイプのソリューションを使用する方法と、ネットワーク・ロード・バランサーを使用してソース IP アドレスをマスクする方法があります。

名前
チャネル名が一致することを検査します。

ALL
キュー・マネージャー名の一致、通信アドレス、およびチャネル名の一致を検査します。

デフォルトは AdoptNewMCACheck=NAME,ADDRESS,QM です。

ChlauthEarlyAdopt=Y（デフォルト）｜N
IBM MQ クライアント・アプリケーション接続のセキュリティー・コンテキストを決定するときに、接続認証規則とチャネル認証規則の処理順序が重要な要素になります。

重要 : ChlauthEarlyAdopt が qm.ini ファイルに存在しない場合のデフォルトは N です。ただし、IBM MQ 9.0.4 以降、すべてのキュー・マネージャーは **ChlauthEarlyAdopt**=Y で作成され、自動的に qm.ini ファイルに追加されます。

キュー・マネージャー上の接続認証 AUTHINFO オブジェクトに ADOPTCTX(YES) が設定されている場合、**ChlauthEarlyAdopt** は、接続認証のためにキュー・マネージャーに指定されたユーザー ID のみを採用します。

ChlauthEarlyAdopt の有効値は、以下の値です。

Y
チャネルは、チャネル認証規則を適用する前に、キュー・マネージャー接続認証を使用して、アプリケーションから提供されたユーザー ID とパスワードの資格情報を検証し、採用します。 この操

作モードでは、チャネル認証規則は、接続認証チェックの結果生成されるユーザー ID と照合されます。

N
チャネルは、チャネル認証規則が適用されるまで、アプリケーションから提供されたユーザー ID とパスワードの資格情報の接続認証検証を遅らせます。 この操作モードでは、チャネル認証のブロッキング規則とマッピング規則は、ユーザー ID とパスワードの検証結果を考慮できません。

例えば、デフォルトの認証情報オブジェクトが **ADOPTCTX(YES)** に設定され、ユーザー fred がログインしているとします。 次の 2 つの CHLAUTH 規則が構成されています。

```
SET CHLAUTH('MY.CHLAUTH') TYPE(ADDRESSMAP) DESCR('Block all access by
default') ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
SET CHLAUTH('MY.CHLAUTH') TYPE(USERMAP) DESCR('Allow user bob and force
CONNAUTH') CLNTUSER('bob') CHCKCLNT(REQUIRED) USERSRC(CHANNEL)
```

次のコマンドを、ユーザー bob のセキュリティー・コンテキストを採用してコマンドを認証する目的で発行します。

```
runmqsc -c -u bob QMGR
```

実際には、キュー・マネージャーは bob ではなく fred のセキュリティー・コンテキストを使用するため、接続に失敗します。

bob のセキュリティー・コンテキストを使用するには、**ChlauthEarlyAdopt** を Y に設定する必要があります。

PasswordProtection = Compatible (デフォルト) |always|optional|warn

IBM MQ 8.0 以降、 IBM MQ client アプリケーションがキュー・マネージャーに接続するときに指定する認証資格情報は、接続で TLS 暗号化を使用しない場合、 IBM MQ MQCSP パスワード保護機能を使用して保護することができます。

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

MQCSP 構造で資格情報を保護する方法、およびこの属性に設定できる値について詳しくは、 MQCSP パスワード保護を参照してください。

IgnoreSeqNumberMismatch=NO（デフォルト） | YES
チャネルの両端のメッセージ・チャネル・エージェント (MCA) はそれぞれ、同期を維持するために、チャネルを介して送信されたメッセージ数を記録します。 例えば、片側のチャネル定義が削除された後、再作成されると、同期が失われる可能性があります。 このような状況では、同期データが失われていることを確認し、チャネルの始動の続行を許可するするために RESET CHANNEL が必要になることがあります。

IgnoreSeqNumberMismatch 属性は、受信側キュー・マネージャー上で設定する必要があります。

事実上、この属性は受信側チャネル上で reset channel コマンドを実行します。

この属性は、以下の値を使用して、チャネル始動時にキュー・マネージャーがシーケンス番号の不一致を処理する方法を制御します。

NO
チャネル再同期中にチャネル・シーケンス番号が検査されます。2 つの MCA が同じシーケンス番号で一致していない場合は、エラー・メッセージ AMQ9526 が報告され、チャネルは始動に失敗します。

YES
チャネル再同期中にチャネル・シーケンス番号が検査されます。2 つの MCA が同じシーケンス番号で一致していない場合は、警告メッセージ AMQ9703 が報告され、チャネルの始動が継続されます。 この属性値は、通常の環境では必要ありません。 災害復旧時などに、同期データが失われたことが判明した場合、このオプションにより、各シーケンス番号の不一致を手動で確認する必要がなくなります。 この値を指定すると、各シーケンス番号の不一致に対応して、管理者が自動的に **RESET CHANNEL** を発行する場合と同様の効果が得られます。

ChlauthIgnoreUserCase=N（デフォルト）｜Y

キュー・マネージャーが CHLAUTH 規則内で、ユーザー名を大/小文字を区別しないで一致させるようにします。 このオプションを選択すると、以下が許可されます。

- CHLAUTH TYPE(USERMAP) 規則の CLNTUSER で、大/小文字を区別しない一致。
- CHLAUTH TYPE(BLOCKUSER) 規則の USERLIST で、大/小文字を区別しない一致。

ChlauthIgnoreUserCase の有効値は、以下の値です。

N

チャネル認証規則は、大/小文字を区別して、クライアント・ユーザー ID を一致させようとします (例: CLNTUSER('Fred') を指定する規則が 'fred' または 'FRED' と一致せず、'Fred' のユーザー ID にのみ一致)。 これがデフォルト値です。

Y

チャネル認証規則は、大/小文字を区別しないで、クライアント・ユーザー ID と一致させようとします (例: CLNTUSER('Fred') を指定した TYPE (USERMAP) または TYPE(USERBLOCK) のチャネル認証規則は、ユーザー ID 'Fred'、'FRED'、'fred' など、大/小文字のあらゆる組み合わせの ID と一致)。

チャネル認証規則と一致させるときに、ユーザー ID の大/小文字を無視する場合は、複数の規則が一致する可能性があることに注意してください。 これが発生した場合、一致する規則は不明確になります。 例えば、次のような規則では、ユーザー 'fred' が CLIENT チャネルを介してキュー・マネージャーに接続した場合、それらは 'mquser1' または 'mquser2' にマップされる可能性があります。

```
SET CHLAUTH('CLIENT') TYPE(USERMAP) CLNTUSER('fred') USERSRC(MAP) MCAUSER('mquser1')
SET CHLAUTH('CLIENT') TYPE(USERMAP) CLNTUSER('FRED') USERSRC(MAP) MCAUSER('mquser2')
```

ChlauthIgnoreUserCase=Y の使用時に生じうる不確実性を回避するには、大/小文字を区別しない一致を使用すると重複し、結果として異なる動作を引き起こす CHLAUTH 規則は、定義しないようにします。

ChlauthIssueWarn = y

SET CHLAUTH コマンドで WARN = YES 属性を設定した場合にメッセージ AMQ9787 を生成するには、この属性を設定します。

スタンザの例

```
Channels:
   MaxChannels=200
   MaxActiveChannels=100
   MQIBindType=STANDARD
   PipelineLength=2
```

関連概念

222 ページの『チャネルの状態』

チャネルは、いつでも多くの状態のうちの 1 つであることができます。 いくつかの状態には、副状態もあります。 チャネルは、指定された状態から他の状態に移動することができます。

Multi qm.ini ファイルの Connection スタンザ

Connection スタンザは、デフォルトのバインド・タイプを定義します。

qm.ini ファイル内の Connection スタンザを使用して、デフォルトのバインド・タイプを指定します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer Extended キュー・マネージャーのプロパティー・ページを使用します。

注 : Connection スタンザが必要な場合は、作成する必要があります。

DefaultBindType=SHARED（デフォルト）|ISOLATED

DefaultBindType を ISOLATED に設定すると、アプリケーションとキュー・マネージャーは別個のプロセスで実行され、両者間でリソースは共有されません。

DefaultBindType を SHARED に設定すると、アプリケーションとキュー・マネージャーは別個のプロセスで実行されますが、いくつかのリソースは両者間で共有されます。

デフォルト値 SHARED です。

重要 : DefaultBindType はすべての MQCONN 呼び出しに適用されます。また、MQCNO_STANDARD_BINDING を指定して MQCONNX を使用する場合にも適用されます。

DefaultBindType を変更すると、一部のアプリケーションでパフォーマンスが低下する場合があります。

スタンザの例

```
Connection:
  DefaultBindType=SHARED
```

Multi 診断メッセージ・ロギング

IBM MQ の診断メッセージ・ログは、IBM MQ システムのさまざまなコンポーネントを使用して IBM MQ 構成やランタイムの状態変更および問題に関連する診断メッセージを報告するメカニズムです。

これらのログは、IBM MQ エラー・ログ と呼ばれる場合もありますが、IBM MQ エラー・メッセージ以外に、情報メッセージや警告メッセージが常に含まれます。 これらのログに報告を行う 3 つの主な IBM MQ コンポーネントは以下のとおりです。

- キュー・マネージャー
- IBM MQ クライアント
- IBM MQ システムの残り

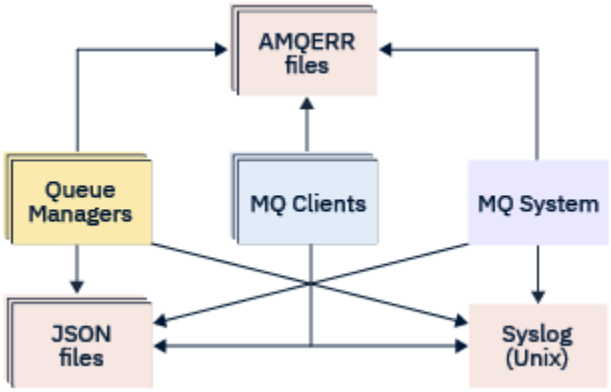
IBM MQ では、診断メッセージ・サービス と呼ばれるさまざまな方式を介した診断メッセージの報告がサポートされているため、以下の情報を記録およびコンシュームするためのアプローチを調整できます。

- AMQERRnn ログ・ファイル
- JSON 形式のログ・ファイル
- AIX JSON 形式の Syslog

IBM MQ による JSON 出力は、1 行の JSON オブジェクトとしてフォーマットされます。つまり、JSON ログまたは Syslog レコードの各行は、有効な JSON オブジェクトを表します。 ログの全体は、単一の JSON オブジェクトとしてカプセル化されません。

以下の図は、キュー・マネージャー、IBM MQ クライアント、および IBM MQ システムのすべてが、説明される方式を使用して診断メッセージを報告できることを示しています。

図 *5. IBM MQ* の各部分が診断メッセージを報告する方法



IBM MQ 診断ログの構成方法

診断ログは、qm.ini ファイル内のスタンザを使用して、それを必要とする IBM MQ コンポーネントに応じて、スタンザを使用して定義およびカスタマイズされます。 それぞれ固有のロギング・エンドポイントが、ini ファイル内の独自のスタンザ・ヘッディングの下で、カスタマイズと共に定義されます。 カスタマイズには以下のことが含まれます。

- ロールオーバーが発生する前に循環させるログ・ファイルのサイズ (Syslog には適用されない)
- ログ・メッセージの重大度に基づくフィルター処理
- 特定のメッセージ・コードの抑止

IBM MQ は 3 つのタイプのロギング・エンドポイントのいずれかまたはすべてに書き込むように構成できるため、特定のログ・スタンザが特定の役割を果たすようにできます。 同様に、複数のファイル・サービスも定義できます。 以下に例を示します。

- JSON 形式を使用すると、ローカル環境および Cloud 環境内で自動化ツールを介して容易に解析できるようになります。
- Syslog 出力を使用すると、IBM MQ コンポーネントで、システムの他の製品に合わせた共通 OS ロギングの場所に診断情報を統合できます。
- 重大度に基づいてフィルター処理されるログ・エンドポイントでは、例えば、システム内の重大エラーのみを特定のログ・ファイルに記録できます。

構成されている診断ロギングのスタイルに関係なく、IBM MQ システム・ログ・ディレクトリー(/var/mqm/errors/AMQERRnn.log)および特定のキュー・マネージャー・ログ・ディレクトリー(/var/mqm/qmgrs/<qmgr_name>/errors/AMQERRnn.log)の下に保持されている従来の診断ファイルは、使用されるその他のロギング構成に加えて、常に書き込まれます。

キュー・マネージャーの場合のみ、125 ページの『診断メッセージ・サービス・スタンザ』の属性を指定して、これらの必須ログのオプション構成を実行できます。

さまざまなスタンザ領域

IBM MQ のさまざまな領域に追加スタンザを適用できます。

Qmgr(qm.ini)
キュー・マネージャーによって生成されるログ・メッセージに適用されます。

System(mqs.ini)
システムによって生成されるログ・メッセージに適用されます。 このオプションは、キュー・マネージャーがそれ自体のログにアクセスまたは書き込みできない場合を除き、キュー・マネージャーに固有ではありません。

Templates(mqs.ini)
テンプレートとしての 1 つ以上のスタンザ。キュー・マネージャーが作成されるときに qm.ini にコピーされます。

Client(mqclient.ini)
クライアント操作に適用されます (例えば、リモート・キュー・マネージャーへのクライアント・モードでの **runmqsc**)。

JSON 形式のログと従来形式のログの間の変換

mqrc コマンドが拡張され、JSON 形式のログと従来形式のログの間、および異なる言語間での多くの変換が可能になりました。

関連資料

125 ページの『診断メッセージ・サービス・スタンザ』
使用可能な診断メッセージ・サービス・オプションによって IBM MQ 診断ロギングをカスタマイズできるので、さまざまな IBM MQ コンポーネントのさまざまなログ・エンドポイントにログ出力を送信できます。

124 ページの『QMErrorLog スタンザ』

qm.ini ファイル内のキュー・マネージャー・エラー・ログ・スタンザ QMErrorLog を使用して、IBM MQ エラー・ログの操作と内容を調整します。

128 ページの『診断メッセージ・サービス』
以下の診断メッセージ・サービスおよびそれらのサービス固有の属性 (構成ファイルの DiagnosticSystemMessages、DiagnosticMessages および DiagnosticMessagesTemplate の各スタンザの下で指定される) を定義できます。

Multi *QMErrorLog スタンザ*

qm.ini ファイル内のキュー・マネージャー・エラー・ログ・スタンザ QMErrorLog を使用して、IBM MQ エラー・ログの操作と内容を調整します。

QMErrorLog サービスは、キュー・マネージャーに関する診断メッセージの出力に使用される従来型の IBM MQ 診断ロギング・サービスです。 QMErrorLog サービスは、継続的に実行され、オフにすることはできませんが、ある程度カスタマイズすることができます。

qm.ini ファイル内の QMErrorLog スタンザを使用すると、特定のメッセージがキュー・マネージャーのエラー・ログに書き込まれないようにすることができます。 また、特定の期間、エラー・ログにメッセージが書き込まれないようにすることも可能です。

Windows Linux また、qm.ini ファイルを直接編集する代わりに、Extended Queue Manager のプロパティページで IBM MQ Explorer、**Excluded messages**、**Suppressed messages**、**Suppressed messages interval** の属性を使用して、メッセージを除外したり抑制したりすることもできます。

重要 :

- Windows IBM MQ Explorer を使用して、Windows プラットフォーム上でローカル・キュー・マネージャーを使用している場合にのみ変更を行うことができます。
- QMErrorLog スタンザは、IBM MQ システム構成ファイル、mqs.ini、またはクライアント構成ファイル(一般的には mqclient.ini という名前)には適用されません。

次の属性を QMErrorLog スタンザに含めることができます。

ErrorLogSize=*maxsize*
バックアップにコピーされるキュー・マネージャー・エラー・ログのサイズを指定します。 *maxsize* は、32768 から 2147483648 バイトの範囲でなければなりません。 **ErrorLogSize** が指定されていないと、デフォルト値である 33554432 バイト (32 MB) が使用されます。

この属性を使用して、最大サイズを必要に応じて以前の最大サイズである 2 MB に戻すことができます。

ログのサイズは、 **MQMAXERRORLOGSIZE** 環境変数を使用して設定できます。

ExcludeMessage= *msgIds*
キュー・マネージャーのエラー・ログに書き込まれないメッセージを指定します。

詳しくは、125 ページの『診断メッセージ・サービス・スタンザ』の ExcludeMessage を参照してください。

SuppressMessage= *msgIds*
指定した時間間隔内で 1 回だけキュー・マネージャーのエラー・ログに書き込まれるメッセージを指定します。 同一のメッセージ ID が SuppressMessage と ExcludeMessage の両方で指定されていると、そのメッセージは除外されます。

オプションは、mqclient.ini で定義された診断メッセージ・サービスには適用できません。 詳しくは、125 ページの『診断メッセージ・サービス・スタンザ』の SuppressMessage を参照してください。

SuppressInterval=*length*
SuppressMessage に指定したメッセージを 1 回だけキュー・マネージャーのエラー・ログに書き込む時間間隔を秒単位で指定します。 *length* は、1 から 86400 秒の範囲でなければなりません。 SuppressInterval が指定されて いないと、デフォルト値の 30 秒が使用されます。

スタンザの例

```
QMErrorLog:
   ErrorLogSize=262144
   ExcludeMessage=7234
   SuppressMessage=9001,9002,9202
   SuppressInterval=30
```

関連概念

98 ページの『キュー・マネージャー構成ファイル qm.ini』

キュー・マネージャー構成ファイル qm.ini には、特定のキュー・マネージャーに関連する情報が含まれています。 個々のキュー・マネージャーの構成を変更するために使用できる属性によって、IBM MQ の設定がオーバーライドされます。

関連資料

125 ページの『診断メッセージ・サービス・スタンザ』

使用可能な診断メッセージ・サービス・オプションによって IBM MQ 診断ロギングをカスタマイズできるので、さまざまな IBM MQ コンポーネントのさまざまなログ・エンドポイントにログ出力を送信できます。

Multi 診断メッセージ・サービス・スタンザ

使用可能な診断メッセージ・サービス・オプションによって IBM MQ 診断ロギングをカスタマイズできるので、さまざまな IBM MQ コンポーネントのさまざまなログ・エンドポイントにログ出力を送信できます。

以下のいずれかの名前のスタンザを使用して、追加の診断メッセージ・サービスを有効にします。

- **DiagnosticSystemMessages**

  システム・エラー・ログに送信される診断メッセージの生成時に使用されるサービスを定義します。 mqs.ini または mqclient.ini ファイルで有効です。

  クライアント・アプリケーションは **DiagnosticSystemMessages** スタンザを mqclient.ini および mqs.ini ファイルで使用し、**DiagnosticSystemMessages** スタンザはキュー・マネージャー・コンテキストがないサーバー・アプリケーションのメッセージを制御します。

  すべてのメッセージを syslog サービスに追加で書き込むキュー・マネージャーとアプリケーションを構成できます。

- **DiagnosticMessages**

  キュー・マネージャー・エラー・ログに送信される診断メッセージの生成時に使用されるサービスを定義します。 qm.ini ファイルでのみ有効です。

- **DiagnosticMessagesTemplate**

  キュー・マネージャーの作成時に mqs.ini ファイルから qm.ini ファイルの **DiagnosticMessages** にコピーされるスタンザです。

診断メッセージを表示するには、mqrc コマンドを使用します。

スタンザの属性

重要 : Service およびスタンザ名は必須です。

name=<stanzaname>
スタンザの名前。 値は ini ファイル内で一意である必要があります。

Service= ***type of service***

この属性では、このスタンザで有効にするサービスを定義します。サービスの名前は、大/小文字を区別しません。

例えば、syslog を追加サービスとして有効にするには、以下のように入力します。

```
Service=syslog
```

128 ページの『診断メッセージ・サービス』と、診断メッセージ・サービス・スタンザで使用可能なそれらのサービス固有の属性を参照してください。

以下のオプション属性をスタンザに追加できます。

- ExcludeMessage
- SuppressMessage
- SuppressInterval
- 127 ページの『Severities』

ExcludeMessage= *msgIds*
キュー・マネージャーのエラー・ログに書き込まれないメッセージを指定します。 IBM MQ システムが大量に使用されている場合、多数のチャネルが停止して開始している場合は、多数の情報メッセージが z/OS コンソールおよびハードコピー・ログに送信されます。 IBM MQ-IMS ブリッジとバッファー管理プログラムも、大量の通知メッセージを生成する場合があるので、必要な場合に、メッセージを除外すると、大量のメッセージは受信されません。 *msgIds* には、次のメッセージ ID のコンマ区切りリストが含まれています。

5211 - プロパティー名の最大長を超えました。
5973 - 分散パブリッシュ/サブスクライブのサブスクリプションは禁止されています
5974 - 分散パブリッシュ/サブスクライブのパブリッシュは禁止されています
6254 - システムは共有ライブラリーを動的にロードできませんでした

7163 - Job started message (ジョブがメッセージを開始しました) (IBM i のみ)
7234 - Number of messages loaded (いくつかのメッセージが読み込まれました)
8245 - エンティティーには、オブジェクトを表示するための十分な権限がありません (Entity has insufficient authority to display object)
9001 - Channel program ended normally (チャネル・プログラムが正常に終了しました)
9002 - Channel program started (チャネル・プログラムが開始されました)
9202 - Remote host not available (リモート・ホストは使用できません)
9208 - ホストからの受信エラー
9209 - 接続の閉止
9228 - チャネル・レスポンダーの開始不能
9489 - SVRCONN インスタンス数の最大限度を超えました
9490 - クライアントあたりの SVRCONN インスタンス数の最大限度を超えました
9508 - キュー・マネージャーに接続不能
9524 - Remote queue manager unavailable (リモート・キュー・マネージャーは使用できません)
9528 - User requested closure of channel (ユーザーがチャネルの閉止を要求しました)
9545 - 切断間隔の有効期限切れ
9558 - リモート・チャネルは使用不可
9637 - チャネルに証明書がありません。
9776 - ユーザー ID によるチャネルのブロック
9777 - NOACCESS マップによるチャネルのブロック
9782 - アドレスによる接続のブロック
9999 - Channel program ended abnormally (チャネル・プログラムが異常終了しました)

SuppressMessage= *msgIds*
指定した時間間隔内で 1 回だけキュー・マネージャーのエラー・ログに書き込まれるメッセージを指定します。 IBM MQ システムが大量に使用されている場合、多数のチャネルが停止して開始している場合は、多数の情報メッセージが z/OS コンソールおよびハードコピー・ログに送信されます。 IBM MQ-IMS ブリッジとバッファー管理プログラムも、大量の通知メッセージを生成する場合があるので、必要な場合に、メッセージを抑止すると、大量の反復メッセージは受信されません。 時間間隔は、

SuppressInterval で指定します。 *msgIds* には、次のメッセージ ID のコンマ区切りリストが含まれています。

5211 - プロパティー名の最大長を超えました。
5973 - 分散パブリッシュ/サブスクライブのサブスクリプションは禁止されています
5974 - 分散パブリッシュ/サブスクライブのパブリッシュは禁止されています
6254 - システムは共有ライブラリーを動的にロードできませんでした

7163 - Job started message (ジョブがメッセージを開始しました) (IBM i のみ)
7234 - Number of messages loaded (いくつかのメッセージが読み込まれました)
8245 - エンティティーには、オブジェクトを表示するための十分な権限がありません (Entity has insufficient authority to display object)
9001 - Channel program ended normally (チャネル・プログラムが正常に終了しました)
9002 - Channel program started (チャネル・プログラムが開始されました)
9202 - Remote host not available (リモート・ホストは使用できません)
9208 - ホストからの受信エラー
9209 - 接続の閉止
9228 - チャネル・レスポンダーの開始不能
9489 - SVRCONN インスタンス数の最大限度を超えました
9490 - クライアントあたりの SVRCONN インスタンス数の最大限度を超えました
9508 - キュー・マネージャーに接続不能
9524 - Remote queue manager unavailable (リモート・キュー・マネージャーは使用できません)
9528 - User requested closure of channel (ユーザーがチャネルの閉止を要求しました)
9545 - 切断間隔の有効期限切れ
9558 - リモート・チャネルは使用不可
9637 - チャネルに証明書がありません。
9776 - ユーザー ID によるチャネルのブロック
9777 - NOACCESS マップによるチャネルのブロック
9782 - アドレスによる接続のブロック
9999 - Channel program ended abnormally (チャネル・プログラムが異常終了しました)

同一のメッセージ ID が SuppressMessage と ExcludeMessage の両方で指定されていると、そのメッセージは除外されます。

オプションは、MQ client.ini で定義された診断メッセージ・サービスには適用できません。

SuppressInterval=*length*
SuppressMessage に指定したメッセージを 1 回だけキュー・マネージャーのエラー・ログに書き込む時間間隔を秒単位で指定します。 *length* は、1 から 86400 秒の範囲でなければなりません。 **SuppressInterval** が指定されて いないと、デフォルト値の 30 秒が使用されます。

Severities
重大度レベルのコンマ区切りリスト。重大度レベルの名前は大/小文字を区別しません。 許容値は以下のとおりです。

- I (または、Information か 0)
- W (または、Warning か 10)
- E (または、Error か 20 と 30)
- S (または、Stop か 40)
- T (または、System か 50)

注 :

1. デフォルト値は all です。
2. 選択した重大度レベルのメッセージのみがサービスに提示されます。

または、プラス文字 (+) を使用して、指定したエラー・レベル以上のメッセージを表示することもできます。 例えば、すべてのエラーを表示するには、以下のように入力します。

```
Severities=E+
```

関連資料

124 ページの『QMErrorLog スタンザ』
qm.ini ファイル内のキュー・マネージャー・エラー・ログ・スタンザ QMErrorLog を使用して、IBM MQ エラー・ログの操作と内容を調整します。

128 ページの『診断メッセージ・サービス』
以下の診断メッセージ・サービスおよびそれらのサービス固有の属性 (構成ファイルの DiagnosticSystemMessages、DiagnosticMessages および DiagnosticMessagesTemplate の各スタンザの下で指定される) を定義できます。

Multi 診断メッセージ・サービス

以下の診断メッセージ・サービスおよびそれらのサービス固有の属性 (構成ファイルの DiagnosticSystemMessages、DiagnosticMessages および DiagnosticMessagesTemplate の各スタンザの下で指定される) を定義できます。

以下の診断メッセージ・サービスが定義されています。

ファイル

このサービスでは、QMErrorLog サービスと同様の方法でフィルター処理されていないメッセージをファイルに送信します。 **Format** の指定値に基づいて、指定されている既存のテキスト形式か JSON 形式のどちらかが使用されます。 デフォルトでは、 **Format** プロパティーに応じて AMQERR01.LOG、AMQERR02.LOG、および AMQERR03.LOG または AMQERR01.json、 AMQERR02.json、および AMQERR03.json という 3 つのファイルがあり、これらのロールオーバーは構成されたサイズに基づいています。

以下の属性は、File スタンザだけでサポートされています。

FilePath

ログ・ファイルの書き込み先のパス。 デフォルトは、システムまたはキュー・マネージャーである AMQERR01.log ファイルと同じロケーションです。 このパスは絶対パスである必要がありますが、置き換え可能な挿入を含めることができます。 以下に例を示します。

+MQ_Q_MGR_DATA_PATH+

キュー・マネージャー診断メッセージ・ディレクトリーの親への絶対パス。 デフォルト値は以下のとおりです。

- Linux AIX AIX and Linux プラットフォームで: /var/mqm/qmgrs/<QM_name>
- Windows で Windows、C:\Program Data\IBM\MQ\qmgrs\<QM_name>

+MQ_DATA_PATH+

システム診断メッセージ・ディレクトリーの親への絶対パス。 デフォルト値は以下のとおりです。

- Linux AIX で AIX and Linux プラットフォーム:/var/mqm
- Windows Windows 上: C:\Program Data\IBM\MQ

既存のエラー・ディレクトリーを使用しない場合は、適切な権限を使用してこのパスを作成する必要があります。

FilePrefix

ログ・ファイルの接頭部。 デフォルトは AMQERR です。

FileSize

ログがロールオーバーするサイズ。 124 ページの『QMErrorLog スタンザ』の **ErrorLogSize** プロパティーと同様に、デフォルトは 32MB です。これは意味的には同一です。

注 : ErrorLogSize プロパティーは、カスタム診断サービスではなく、デフォルトのエラー・ログ・サービスにのみ適用されます。

ログのサイズは、 **MQMAXERRORLOGSIZE** 環境変数を使用して設定できます。

Format
ファイルの形式。 値は、*text* (追加の QMErrorLog スタイル・サービスの場合) または *json* (デフォルト) のどちらかです。

ファイルの接尾部は、この属性の設定に基づいて、 `.LOG` と `.json` のどちらかになります。

例えば、キュー・マネージャーの `qm.ini` ファイルを編集し、以下のスタンザを追加します:

```
DiagnosticMessages:
   Service = File
   Name = JSONLogs
   Format = json
   FilePrefix = AMQERR
```

再始動後、キュー・マネージャーは、ERRORS ディレクトリーに `AMQERR0x.json` ファイルを持つことになります。

複数の File サービスを定義できます。 これにより、以下の例に示す構成が可能になります。異なるタグのメッセージは、異なるログのセットに分割されます。

```
DiagnosticMessages:
  Name=ErrorsToFile
  Service=File
  Severities=E+
  FilePrefix=OnlyErrors

DiagnosticMessages:
  Name=NonErrorstoFile
  Service=File
  Severities=I,W
  FilePrefix=InformationandWarning
```

Linux AIX **Syslog**
Syslog サービスは、Windows または IBM i では使用できません。

定義できる Syslog サービスは 1 つのみであり、Syslog サービスでは、JSON 形式の診断メッセージ仕様を使用して、フィルター処理されていないメッセージが syslog に送信されます。 情報は表に示されている順序で syslog に追加され、msgID と挿入で開始されます。

メッセージの重大度は、以下のように syslog レベルにマップされます。

表 *12.* メッセージの重大度と *syslog* レベルのマッピング

重大度	**レベル**
0	LOG_INFO
10	LOG_WARNING
20	LOG_ERR
30	LOG_ERR
40	LOG_ALERT
50	LOG_ALERT

以下の属性は、syslog スタンザだけでサポートされています。

Ident
syslog 項目に関連付けられた **ident** 値を定義します。 デフォルト値は *ibm-mq* です。

Syslog に送信されるエラー・メッセージの例を以下に示します。

```
DiagnosticMessages:
  Name=ErrorsToSyslog
  Ident=mq
  Service=Syslog
  Severities=E+
```

スタンザの一般的な属性について詳しくは、125 ページの『診断メッセージ・サービス・スタンザ』を参照してください。

注:

1. File サービスの場合のみ、それぞれ名前が異なる複数のスタンザを持つことができます。 シーケンス内の最後の名前を使用した定義のみが有効になります。
2. スタンザの値に加えた変更は、キュー・マネージャーを再始動した場合にのみ有効になります。

Multi qm.ini ファイルの ExitPath スタンザ

ExitPath スタンザは、キュー・マネージャー・システム上のユーザー出口プログラムのパスを指定します。

qm.ini ファイルの ExitPath スタンザを使用して、キュー・マネージャー・システム上のユーザー出口プログラムのパスを指定します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer Exits キュー・マネージャーのプロパティー・ページを使用します。

ExitsDefaultPath=*string*
ExitsDefaultPath 属性は、次の出口の位置を指定します。

- クライアントの 32 ビット・チャネル出口
- サーバーの 32 ビット・チャネル出口およびデータ変換出口
- 非修飾 XA スイッチ・ロード・ファイル

ExitsDefaultPath64= *string*
ExitsDefaultPath64 属性は、次の出口の位置を指定します。

- クライアントの 64 ビット・チャネル出口
- サーバーの 64 ビット・チャネル出口およびデータ変換出口
- 非修飾 XA スイッチ・ロード・ファイル

スタンザの例

```
ExitPath:
   ExitsDefaultPath=/var/mqm/exits
   ExitsDefaultPath64=/var/mqm/exits64
```

Multi qm.ini ファイルの ExitPropertiesLocal スタンザ

ExitPropertiesLocal スタンザは、キューマネージャの終了プロパティに関する情報を指定します。

キューマネージャの終了プロパティに関する情報を指定するには、'qm.ini ファイルの ExitPropertiesLocal スタンザを使用します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer クラスター・キュー・マネージャーのプロパティー・ページを使用します。

Windows あるいは、 Windows では、 **amqmdain** コマンドを使用してこの情報を指定できます。

デフォルトでは、この設定は、マシン全体の構成の ExitProperties スタンザに含まれている **CLWLMode** 属性から継承されます (93 ページの『mqs.ini ファイルの ExitProperties スタンザ』を参照してください)。 こ

の設定を変更するのは、このキュー・マネージャーを別の方法で構成する場合のみにしてください。 この値は、クラスター・キュー・マネージャー・プロパティー・ページのクラスター・ワークロード・モード属性を使用して、キュー・マネージャーごとに個別にオーバーライドできます。

mqs.ini ファイルの ExitProperties スタンザを使用して、キュー・マネージャー出口プログラムによって使用される構成オプションを指定します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer Extended IBM MQ のプロパティー・ページを使用します。

CLWLMode = SAFE (デフォルト) | 高速
クラスター・ワークロード (CLWL) 出口を使用すると、MQI 呼び出し (例えば、 MQOPEN、 MQPUT) への応答として、クラスター内のどのクラスター・キューを開くかを指定できます。 CLWL 出口は、**CLWLMode** 属性に指定した値に応じて、FAST モードまたは SAFE モードのいずれかで実行されます。 **CLWLMode** 属性を省略すると、クラスター・ワークロード出口は SAFE モードで実行されます。

SAFE
CLWL 出口をキュー・マネージャーから独立したプロセスで実行します。 これはデフォルトです。

SAFE モードでの実行中にユーザー作成の CLWL に問題が起きた場合は、次のような結果が生じます。

- CLWL サーバー・プロセス (amqzlwa0) は失敗します。
- キュー・マネージャーは CLWL サーバー・プロセスを再始動します。
- エラーがエラー・ログに記録されます。 進行中の MQI 呼び出しがある場合は、戻りコードが戻されます。

キュー・マネージャーの保全性は維持されます。

注 : CLWL 出口を別のプロセスで実行すると、パフォーマンスに影響を与える可能性があります。

FAST
クラスター出口は、キュー・マネージャー・プロセスでインラインで実行します。

このオプションを指定すると、SAFE モードで実行した場合のプロセスの切り替えコストを回避できるのでパフォーマンスは向上しますが、代わりにキュー・マネージャーの保全性が保証されないことになります。 CLWL 出口を FAST モードで実行する必要があるのは、CLWL 出口に問題がなく、特にパフォーマンスについて懸念があると確信している場合のみです。

CLWL 出口を FAST モードで実行しているときに問題が発生すると、キュー・マネージャーは失敗し、キュー・マネージャーの保全性が損なわれる恐れがあります。

スタンザの例

```
ExitPropertiesLocal:
      CLWLMode=SAFE
```

Linux IBM i AIX qm.ini ファイルの Filesystem スタンザ

Filesystem スタンザは、キュー・マネージャーのエラー・ログに設定された許可を未変更のままにするか、デフォルト値に戻すかを指定します。

エラー・ログ・ファイルに設定されたデフォルトの権限は、ほとんどの環境で役に立つことが予想されます。したがって、ほとんどの IBM MQ 管理者は、これらを変更する必要はありません。

ただし、 IBM MQ 管理者は、エラー・ログ・ファイルの許可を変更したい場合があります。その場合は、Filesystem スタンザ・オプション **ValidateAuth**=*No* を設定する必要があります。これにより、キュー・マネージャーは許可を未変更のままにします。

デフォルトの動作 (**ValidateAuth**=*No* を指定しない場合) では、キュー・マネージャーはキュー・マネージャーのエラー・ログのファイル許可を検査し、それらをデフォルト値に戻します。 検査は、キュー・マネージャーの終了時や操作開始時など、いつでも発生することがあります。

スタンザの例

```
Filesystem:
      ValidateAuth=No
```

Multi qm.ini ファイルの Log スタンザ

Log スタンザは、キュー・マネージャーでのロギングに関する情報を指定します。

qm.ini ファイル内の Log スタンザを使用して、キュー・マネージャーでのロギングに関する情報を指定します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer 「**ログ・キュー・マネージャー**」 プロパティー・ページを使用します。

デフォルトでは、これらの設定は、キュー・マネージャーのデフォルト・ログ設定に指定された設定から継承されます (94 ページの『mqs.ini ファイルの LogDefaults スタンザ』を参照)。 これらの設定を変更するのは、このキュー・マネージャーを別の方法で構成する場合のみにしてください。

ログ・サイズの計算について詳しくは、 646 ページの『ログのサイズの計算』を参照してください。

注 : 以下のパラメーター・リストに示されている制限は、 IBM MQ によって設定されます。 オペレーティング・システムの制限により、最大可能ログ・サイズがさらに減少することもあります。

LogPrimaryFiles=3(デフォルト)/*2-254*('Windows)/*2-510*('AIX and Linux システム)
キュー・マネージャーの作成時に割り振られるログ・ファイル。

1 次ログ・ファイルの最小数は 2 であり、最大数は Windows では 254、AIX and Linux システムでは 510 です。 デフォルトは 3 です。

1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 3 以上にする必要があり、かつ Windows では 255 以下、AIX and Linux システムでは 511 以下にする必要があります。

この値は、キュー・マネージャーの作成時または開始時に調べられます。 キュー・マネージャーが作成された後に、この値を変更することができます。 ただし、この変更された値は、キュー・マネージャーが再始動されるまで有効にならないので、効果はただちに現れません。

LogSecondaryFiles=2(デフォルト)/*1-253*('Windows)/*1-509*('AIX and Linux システム)
1 次ファイルが足りなくなったときに割り振られるログ・ファイル。

2 次ログ・ファイルの最小数は 1 であり、最大数は Windows では 253、AIX and Linux システムでは 509 です。 デフォルトの数は 2 です。

1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 3 以上にする必要があり、かつ Windows では 255 以下、AIX and Linux システムでは 511 以下にする必要があります。

この値は、キュー・マネージャーの始動時に検査されます。 この値は変更することができます。ただし、変更された値は、キュー・マネージャーが再始動されるまでは有効にはなりません。有効になった場合でも効果がただちに現れるとは限りません。

LogFilePages=*number*
ログ・データは、ログ・ファイルと呼ばれる一連のファイルに保持されます。 ログ・ファイル・サイズは、4 KB ページ単位で指定します。

ログ・ファイルのデフォルトのページ数は 4096 です。 これは、16 MB のログ・ファイル・サイズと同じです。

AIX and Linux システムの場合、ログ・ファイルの最小ページ数は 64 です。Windows の場合、ログ・ファイルの最小ページ数は 32 です。最大ページ数はどちらも 65 535 です。

注 : キュー・マネージャーの作成時に指定されたログ・ファイルのサイズは、キュー・マネージャーでは変更できません。

LogType = 循環 (デフォルト) | LINEAR| REPLICATED
キュー・マネージャーで使用するロギングのタイプ。 デフォルトは 循環です。 必要なタイプのロギングを使用してキュー・マネージャーを作成する方法について詳しくは、94 ページの『mqs.ini ファイルの LogDefaults スタンザ』の **LogType** 属性の説明を参照してください。

CIRCULAR
ログを使用して再始動リカバリーを開始し、システムの停止時に進行中であったトランザクションをロールバックします。

循環ロギングの詳細については、641 ページの『ログのタイプ』を参照してください。

LINEAR
再始動リカバリーと、メディアまたは順方向リカバリー (ログの内容を再生することにより、消失または損傷したデータを作成する) の両方を可能にします。

リニア・ロギングの詳細については、641 ページの『ログのタイプ』を参照してください。

CP4I **REPLICATED**
ログ・データをアクティブ・インスタンスから複製インスタンスに複製するために、Native HA グループによって使用されます。

複製ロギングの詳細については、641 ページの『ログのタイプ』を参照してください。

注 : `qm.ini` ファイル内のこの属性を変更してキュー・マネージャーの **LogType** を変更することはできません。 キュー・マネージャーの **LogType** を変更するには、**migmqlog** コマンドを使用する必要があります。

LogBufferPages=0 (デフォルト) |*0* から *4096*
書き込み用としてバッファー・レコードに割り振るメモリーの量。バッファーのサイズは 4 KB ページ単位で指定します。

バッファーの最小ページ数は 18 であり、最大ページ数は 4096 です。 バッファーが大きいとスループットが高くなりますが、これは特に大きなメッセージに適用されます。

0 (デフォルト) を指定すると、サイズはキュー・マネージャーによって選択されます。

1 から 17 の範囲の数を指定すると、キュー・マネージャーはデフォルトで 18 (72 KB) になります。 18 から 4096 の範囲の数値を指定すると、キュー・マネージャーは指定された数値を使用して、割り振られるメモリーの量を設定します。

この値は、キュー・マネージャーの始動時に検査されます。 値は、制限内で増減できます。 ただし、この値の変更は、 キュー・マネージャーが次に始動されるまで有効になりません。

LogPath=*directory_name*
キュー・マネージャーのログ・ファイルが置かれるディレクトリー。 ディレクトリーは、キュー・マネージャーが書き込みを行える ローカル装置に置くか、できればメッセージ・キューとは異なるドライブに置く必要があります。 異なるドライブを指定すると、システム障害時の保護機能が加わります。

デフォルトは次のとおりです。

- Windows Windows の `C:\ProgramData\ IBM \ MQ\log`。
- Linux AIX AIX and Linux システムの `/var/mqm/log`。

-ld フラグを使用して、 **crtmqm** コマンドでディレクトリーの名前を指定できます。 キュー・マネージャーが作成されると、キュー・マネージャー・ディレクトリーの下にもう 1 つディレクトリーが作成され、ログ・ファイルの保存に使用されます。 このディレクトリーの名前はキュー・マネージャー名に基づいて付けられます。 これによってログ・ファイル・パスが確実に固有になり、ディレクトリー名の長さに関する制限にも抵触しないことになります。

crtmqm コマンドで `-ld` を指定しない場合は、 **LogDefaultPath** 属性の値が使用されます。

Linux AIX AIX and Linux システムでは、ユーザー ID mqm およびグループ mqm に、ログ・ファイルに対する全権限が必要です。 これらのファイルの位置を変更する場合は、それらの権

限を取得する必要があります。 ログ・ファイルがプロダクトで提供されたデフォルト位置にある場合、これは必要ありません。

LogWriteIntegrity=SingleWrite|DoubleWrite|TripleWrite（デフォルト）
高信頼ログ・レコードを書き込むためにロガーが使用する方式。

TripleWrite (デフォルト)

なお、DoubleWrite を選択可能ですが、そのようにした場合、システムでは TripleWrite と解釈されます。

SingleWrite
SingleWrite は、IBM MQ リカバリー・ログをホスティングしているファイル・システムおよびデバイスが、最小単位 4KB の書き込みを明示的に保証している場合にのみ使用してください。

すなわち、何らかの原因で 4KB ページの書き込みが失敗した場合、変更前イメージまたは変更後イメージの 2 つの状態しかありません。 中間の状態は考えられません。

注 : 持続するワークロードにおける並行性が十分であるなら、デフォルト値 TripleWrite 以外の値に設定することで得られる利点はほとんどありません。

詳細については、135 ページの『LogWriteIntegrity-using SingleWrite または TripleWrite』を参照してください。

LogManagement = Manual (デフォルト) | Automatic | Archive
ログ・エクステントを手動またはキュー・マネージャーによって管理するために使用される方式。 デフォルト値は 「手動」です。

この属性は、 **LogType** が LINEAR の場合にのみ適用されます。

LogManagement の値を変更しても、キュー・マネージャーを再始動するまでその変更は有効になりません。

属性に認識されない値が見つかると、その値が修正されるまでキュー・マネージャーは始動しません。

IBM i **LogManagement** プロパティーは、IBM i では無効です。

手動 (デフォルト)
手動でログ・エクステントを管理します。 このオプションを指定すると、リカバリー用のログ・エクステントが不要になった場合でも、キュー・マネージャーは、ログ・エクステントを再使用したり削除したりしません。

自動
キュー・マネージャーによってログ・エクステントが自動的に管理されます。 このオプションを指定すると、リカバリー用のログ・エクステントが不要になった時点で、キュー・マネージャーがログ・エクステントを再使用したり削除したりできるようになります。 アーカイブ保存はできません。

アーカイブ
キュー・マネージャーによってログ・エクステントが管理されますが、それぞれのログ・エクステントのアーカイブ保存が完了したことをキュー・マネージャーに通知する必要があります。

このオプションを指定すると、不要になったリカバリー用のログ・エクステントのアーカイブ保存が完了したことをキュー・マネージャーに通知した時点で、キュー・マネージャーがログ・エクステントを自由に再使用したり削除したりできるようになります。

この通知は、 **RESET QMGR** MQSC コマンドまたは Reset Queue Manager PCF コマンドを使用して行います。

スタンザの例

```
Log:
   LogPrimaryFiles=3
   LogSecondaryFiles=2
   LogFilePages=4096
   LogType=CIRCULAR
   LogBufferPages=0
```

```
LogPath=/var/mqm/log/saturn!queue!manager/
```

注 : **LogBufferPages** の値を 0 にすると、値 512 が使用されます。

Multi *LogWriteIntegrity-using SingleWrite または TripleWrite*

qm.ini ファイルの Log スタンザにある **LogWriteIntegrity** オプションの設定によって、IBM MQ のロガーがログ・レコードをリカバリー・ログに書き込むために使用するアルゴリズムが決まります。 デフォルト設定は TripleWrite で、この設定はほぼすべてのシナリオで安全です。

LogWriteIntegrity の設定は、部分的なログ・ページが書き込まれる場合にのみ効果があります。 並行アクティビティーの量が適切なキュー・マネージャーでは、このような状況はほとんど発生しません。

SingleWrite

SingleWrite は、非常にまれな状況でデフォルトの TripleWrite 設定よりも優れたパフォーマンスを出すアルゴリズムを選択します。 SingleWrite 設定は、 IBM MQ リカバリー・ログに同期的に書き込まれる 4KB ページがアトミックに書き込まれることを、基礎となるストレージ・プラットフォームがあらゆる状況下で絶対に保証できる場合にのみ安全です。

SingleWrite 設定は、IBM MQ リカバリー・ログをホストするファイル・システムまたはデバイスが 4KB 書き込みのアトミシティを明示的に保証する場合にのみ使用してください。 つまり、何らかの理由で 4 KB のページの書き込みが失敗した場合は、変更前イメージまたは変更後イメージの 2 つの状態しか存在せず、中間の状態が存在しないことになります。 それ以外の場合は、 TripleWrite を使用してください。

並行性が十分にあるシステムでは、キュー・マネージャーはログ・データのフル・ページのみを書き込みます。 フル・ページの割合が高い場合は、SingleWrite と TripleWrite の間にパフォーマンス上の大きな違いはありません。

並行性がほとんどないシステムでは、SingleWrite にパフォーマンス上の大きな利点があります。 ただし、通常は、SingleWrite を使用するよりも並行性を高める方法をお勧めします。

4 KB 書き込みのアトミシティを確実に判断することは難しく、また、基礎のソフトウェアまたはハードウェアを変更すると、そのような保証が無効になる可能性があることに注意してください。

現在かつ今後いかなるときも、あらゆる状況下で、必要な保証を与えてくれるストレージ・インフラストラクチャーであるという確信がない場合は、 TripleWrite を使用する必要があります。

Windows qm.ini ファイルの LU62 スタンザ (Windows のみ)

LU62 スタンザは、SNA LU 6.2 プロトコル構成パラメーターを指定します。 これらのパラメーターは、チャネルのデフォルト属性をオーバーライドします。

qm.ini ファイルの LU62 スタンザを使用して、SNA LU 6.2 プロトコル構成パラメーターを指定します。 これらのスタンザは、チャネルのデフォルトの属性を指定変更します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer LU6.2 キュー・マネージャーのプロパティー・ページを使用します。

TPName
リモート・サイトで始動する TP 名。

Library1=*DLLName 1*
APPC DLL の名前。

デフォルト値は WCPIC32 です。

Library2= *DLLName2*
Library1 と同様に、コードが 2 つの別個のライブラリーに格納されている場合に使用されます。

デフォルト値は WCPIC32 です。

CP4I qm.ini ファイルの NativeHAInstance スタンザ

コンテナー内の IBM MQ の場合、 NativeHAInstance スタンザは、ネイティブ HA 構成内の 3 つのノードが相互に通信する方法を指定します。

注 : この情報は、コンテナー環境にのみ適用されます。 IBM MQ オペレーターを使用したネイティブ HA の構成 、または 独自のコンテナーを作成する場合のネイティブ HA グループの作成を参照してください。

ネイティブ HA グループ内のキュー・マネージャー・インスタンス (ローカル・インスタンスを含む) ごとに 1 つずつ、合計 3 つの NativeHAInstance スタンザを追加します。 以下の属性を追加します。

名前
キュー・マネージャー・インスタンスの作成時に使用したインスタンス名を指定します。

ReplicationAddress
インスタンスのホスト名、 IPv4 ドット 10 進または IPv6 16 進形式のアドレスを指定します。 アドレスは、ホスト名、 IPv4 小数点付き 10 進数、または IPv6 16 進形式のアドレスとして指定できます。 複製アドレスは、グループ内の各インスタンスから解決可能かつルーティング可能でなければなりません。 ログ複製に使用するポート番号は、大括弧で囲んで指定する必要があります。以下に例を示します。

```
ReplicationAddress=host1.example.com(4444)
```

スタンザの例

以下の例は、ネイティブ HA 構成の 3 つのノードを指定するために qm.ini ファイルで使用される NativeHAInstance スタンザを示しています。

```
NativeHAInstance:
  Name=node-1
  ReplicationAddress=host1.example.com(4444)
NativeHAInstance:
  Name=node-2
  ReplicationAddress=host2.example.com(4444)
NativeHAInstance:
  Name=node-3
  ReplicationAddress=host3.example.com(4444)
```

関連概念
136 ページの『qm.ini ファイルの NativeHALocalInstance スタンザ』
コンテナー内の IBM MQ の場合、 NativeHALocalInstance スタンザはネイティブ HA 構成の操作を制御します。

CP4I qm.ini ファイルの NativeHALocalInstance スタンザ

コンテナー内の IBM MQ の場合、 NativeHALocalInstance スタンザはネイティブ HA 構成の操作を制御します。

注 : この情報は、コンテナー環境にのみ適用されます。 IBM MQ オペレーターを使用したネイティブ HA の構成 、または 独自のコンテナーを作成する場合のネイティブ HA グループの作成を参照してください。

ネイティブ HA 構成を作成すると、各ノードの qm.ini ファイルに NativeHALocalInstance スタンザが自動的に追加されます。 その後、 qm.ini ファイルを編集し、 NativeHALocalInstance スタンザ内の属性をカスタマイズできます。

LocalName
NativeHALocalInstance スタンザの名前は、ネイティブ HA キュー・マネージャーの作成時に指定されたログ・レプリカ・インスタンス名から取得されます。

オプションで、以下の属性を NativeHALocalInstance スタンザに追加できます。

KeyRepository

V 9.3.0 V 9.3.0 ログ複製トラフィックを保護するために使用されるデジタル証明書を保持する鍵リポジトリーの絶対パスおよびファイル名。 ファイル拡張子が指定されていない場合、.kdb であると想定されます。

KeyRepository スタンザ属性を省略すると、インスタンス間でログ複製データがプレーン・テキストで交換されます。

V 9.3.2 **KeyRepositoryPassword**

鍵リポジトリーには機密情報が含まれているため、鍵リポジトリーはパスワードで保護されます。 鍵リポジトリーの内容にアクセスできるようにするには、IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。 パスワードが鍵リポジトリー stash ファイルに保管されていない場合は、KeyRepositoryPassword 属性にパスワードを指定できます。 以下に例を示します。

```
KeyRepositoryPassword=passw0rd
```

重要: この属性を使用してパスワードを指定する場合は、IBM MQ パスワード保護システムを使用してパスワードを暗号化します。 詳しくは、138 ページの『鍵リポジトリー・パスワードの暗号化』を参照してください。

V 9.3.2 **InitialKeyFile**

KeyRepositoryPassword 属性で指定された鍵リポジトリー・パスワードが特定の初期鍵で暗号化されている場合は、この属性を指定します。 初期鍵を含むファイルの名前は、 **runmqicred** コマンドを使用して鍵リポジトリー・パスワードを暗号化するときに、 **-sf** パラメーターを使用して指定できます。

この属性の値を、パスワードの暗号化に使用される初期鍵を含むファイルの名前に設定します。 例えば、 mykey.key というファイルに初期キーが含まれているとします。

```
InitialKeyFile=/mykey.key
```

詳細については、138 ページの『鍵リポジトリー・パスワードの暗号化』を参照してください。

CertificateLabel

ログ複製トラフィックの保護に使用するデジタル証明書を識別する証明書ラベル。 KeyRepository が指定されているが、 CertificateLabel が省略されている場合は、デフォルト値 ibmwebspheremq*queue_manager* が使用されます。

CipherSpec

ログ複製トラフィックを保護するために使用する CipherSpec 。 このスタンザ属性を指定する場合は、KeyRepository も指定する必要があります。 KeyRepository が指定されているが、 CipherSpec が省略されている場合は、デフォルト値 ANY が使用されます。

LocalAddress

ログ複製トラフィックを受け入れるローカル・ネットワーク・インターフェース・アドレス。 このスタンザ属性が指定されている場合は、「[addr] [(port)]」の形式を使用してローカル・ネットワーク・インターフェースまたはポート (あるいはその両方) を識別します。 ネットワーク・アドレスは、ホスト名、 IPv4 小数点付き 10 進数、または IPv6 16 進形式で指定できます。 この属性を省略すると、キュー・マネージャーはすべてのネットワーク・インターフェースへのバインドを試行し、ローカル・インスタンス名と一致する NativeHAInstances スタンザの ReplicationAddress に指定されているポートを使用します。

HeartbeatInterval

ハートビート間隔は、ネイティブ HA キュー・マネージャーのアクティブ・インスタンスがネットワーク・ハートビートを送信する頻度をミリ秒単位で定義します。 ハートビート間隔値の有効範囲は 500 (0.5 秒) から 60000 (1 分) で、この範囲外の値を使用するとキュー・マネージャーの開始が失敗します。 この属性を省略すると、デフォルト値の 5000 (5 秒) が使用されます。 各インスタンスで同じハートビート間隔を使用する必要があります。

HeartbeatTimeout

ハートビート・タイムアウトは、ネイティブ HA キュー・マネージャーのレプリカ・インスタンスが、アクティブ・インスタンスが応答しないと判断するまで待機する時間の長さを定義します。 ハートビ

ート間隔タイムアウト値の有効範囲は 500 (0.5 秒) から 120000 (2 分) です。 ハートビート・タイムアウトの値は、ハートビート間隔以上でなければなりません。

無効な値を指定すると、キュー・マネージャーの開始が失敗します。 この属性が省略されると、レプリカは、新しいアクティブ・インスタンスを選択するプロセスを開始する前に 2 x HeartbeatInterval 待機します。 各インスタンスで同じハートビート・タイムアウトを使用する必要があります。

RetryInterval

再試行間隔は、障害が発生した複製リンクがネイティブ HA キュー・マネージャーで再試行される頻度をミリ秒単位で定義します。 再試行間隔の有効範囲は 500 (0.5 秒) から 120000 (2 分) です。 この属性が省略されると、レプリカは、障害が発生した複製リンクを再試行する前に 2 x HeartbeatInterval 待機します。

SSLFipsRequired

ログ複製トラフィックの送信に暗号化が使用される場合に、FIPS 認証アルゴリズムのみを使用するかどうかを指定します。 Yes または No に設定します。

EncryptionPolicySuiteB

ログ複製トラフィックが Suite-B 準拠の暗号化を使用するかどうか、および使用する強度のレベルを指定します。 以下のいずれかの値に設定します。

NONE
Suite-B 準拠の暗号化を使用しない。 この設定はデフォルト設定です。

128_BIT,192_BIT
セキュリティー強度を 128 ビットと 192 ビットの両方のレベルに設定する。

128_BIT
セキュリティー強度を 128 ビット・レベルに設定する。

192_BIT
セキュリティー強度を 192 ビット・レベルに設定する。

鍵リポジトリー・パスワードの暗号化

V 9.3.2

鍵リポジトリー・パスワードは、 IBM MQ パスワード保護システムまたは鍵リポジトリー stash ファイルを使用して保護できます。 これら 2 つの方法について詳しくは、 鍵リポジトリー・パスワードの暗号化を参照してください。

NativeHALocalInstance スタンザの KeyRepositoryPassword 属性を使用してリポジトリー・パスワードを指定する場合は、 IBM MQ パスワード保護システムを使用してパスワードを暗号化します。 パスワードを暗号化するには、 **runmqicred** コマンドを使用します。 このコマンドは、 KeyRepositoryPassword 属性で指定できる暗号化されたパスワードを返します。

固有の初期鍵を使用して、パスワードを安全に暗号化します。 初期キーを含むファイルの名前は、 **runmqicred** コマンドに対して **-sf** パラメーターを使用して指定できます。 固有キーを指定しない場合は、デフォルトのキーが使用されます。

固有の初期鍵を使用して鍵リポジトリー・パスワードを暗号化する場合は、 NativeHALocalInstance スタンザの InitialKeyFile 属性を使用して、同じ初期鍵も指定する必要があります。

スタンザの例

以下の例は、ノードのローカル名を指定するために qm.ini ファイルで使用される NativeHALocalInstance スタンザを示しています。

```
NativeHALocalInstance:
  LocalName=node-1
```

関連概念

136 ページの『qm.ini ファイルの NativeHAInstance スタンザ』

コンテナー内の IBM MQ の場合、 NativeHAInstance スタンザは、ネイティブ HA 構成内の 3 つのノードが相互に通信する方法を指定します。

関連資料

runmqicred (IBM MQ クライアント・パスワードの保護)

Windows qm.ini ファイルの NETBIOS スタンザ (Windows のみ)

qm.ini ファイル内の NETBIOS スタンザは、 NetBIOS プロトコル構成パラメーターを指定します。 これらのパラメーターは、チャネルのデフォルト属性をオーバーライドします。

NetBIOS プロトコル構成パラメーターを指定するには、 qm.ini ファイルの NETBIOS スタンザを使用します。 これらのスタンザは、チャネルのデフォルトの属性を指定変更します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer Netbios キュー・マネージャー・プロパティー・ページを使用します。

LocalName=*name*
このマシンが LAN 上で識別される名前。

AdapterNum = 0 (デフォルト) | *adapter_number*
LAN アダプターの番号。 デフォルトはアダプター 0 です。

NumSess = 1 (デフォルト) | *number_of_sessions*
割り振るセッションの数。 デフォルトは 1 です。

NumCmds = 1 (デフォルト) | *number_of_commands*
割り振るコマンドの数。 デフォルトは 1 です。

NumNames = 1 (デフォルト) | *number_of_names*
割り振る名前の数。 デフォルトは 1 です。

Library1= *DLLName1*
NetBIOS DLL の名前。

デフォルト値は NETAPI32 です。

関連概念

257 ページの『IBM MQ ローカル NetBIOS 名の定義』

IBM MQ チャネル・プロセスで使用されるローカル NetBIOS 名は、3 つの方法で指定できます。

Linux AIX qm.ini ファイルの RestrictedMode スタンザ

RestrictedMode スタンザは、MQI アプリケーションの実行、すべての IPCC リソースの更新、および一部のキュー・マネージャー・ディレクトリーの内容の変更を許可されるメンバーを含むグループの名前を指定します。 このスタンザは、 AIX and Linux システムにのみ適用されます。

RestrictedMode スタンザは、**crtmqm** コマンドの **-g** オプションにより設定します。 **-g** オプションを使用しなかった場合は、qm.ini ファイル内にこのスタンザは作成されません。

キュー・マネージャー・データ・ディレクトリー内には、 IBM MQ アプリケーションが、キュー・マネージャーに接続している間に ファイルを作成するディレクトリーがあります。 これらのディレクトリー内にファイルを作成できるように、 アプリケーションには、以下の world write アクセス権限が付与されます。

- /var/mqm/sockets/*QMgrName*/@ipcc/ssem/*hostname*/
- /var/mqm/sockets/*QMgrName*/@app/ssem/*hostname*/
- /var/mqm/sockets/*QMgrName*/zsocketapp/*hostname*/

ここで、*QMGRNAME* はキュー・マネージャーの名前で、*hostname* はホスト名です。

一部のシステムでは、すべてのユーザーにこれらのディレクトリーへの書き込みアクセス権限を付与することは望ましくありません。 例えば、ユーザーによっては、キュー・マネージャーへのアクセス権限は必要ありません。 制限モードでは、キュー・マネージャー・データを保管するディレクトリーに関する許可が変更されます。 その後、これらのディレクトリーには指定されたアプリケーション・グループのメンバーだけがアクセスできるようになります。 キュー・マネージャーとの通信に使用される System V IPC 共有メモリーに対する許可も、同様の仕方で変更されます。

アプリケーション・グループは、以下の内容を許可されるメンバーで構成されるグループの名前です。

- MQI アプリケーションの実行
- すべての IPCC リソースの更新
- 一部のキュー・マネージャー・ディレクトリーの内容の変更

キュー・マネージャーの制限モードを使用するには、 次の条件を満たす必要があります。

- キュー・マネージャーの作成者は、mqm グループとアプリケーション・グループに含まれている必要があります。
- mqm ユーザー ID は、アプリケーション・グループに含まれている必要があります。
- キュー・マネージャーを管理するすべてのユーザーは、mqm グループとアプリケーション・グループに含まれている必要があります。
- IBM MQ アプリケーションを実行するすべてのユーザーは、 アプリケーション・グループに含まれている必要があります。

アプリケーション・グループに入っていないユーザーが実行した MQCONN または MQCONNX の呼び出しは、理由コード MQRC_Q_MGR_NOT_AVAILABLE で失敗します。

重要 : 多くのオペレーティング・システムでは、グループにユーザーを追加して認識させるには、問題のユーザーがログオフしてから再度ログオンする必要があります。

制限モードは、IBM MQ 許可サービスで作動します。 そのため、 ユーザーには、IBM MQ に接続し、 IBM MQ 許可サービスを使用して必要なリソースにアクセスする権限も付与する必要があります。

ALW IBM MQ 許可サービスの構成について詳しくは、 AIX, Linux, and Windows システムでのセキュリティーのセットアップをご覧ください。

IBM MQ の制限モードは、 許可サービスによって提供される制御によってキュー・マネージャー・リソースが十分に分離されない場合にのみ使用してください。

関連資料

crtmqm (キュー・マネージャーの作成)

Multi qm.ini ファイルの Security スタンザ

Security スタンザは、オブジェクト権限マネージャー (OAM) のオプションを指定します。

ClusterQueueAccessControl=RQMName|Xmitq

クラスター・キュー・マネージャー上でホストされるクラスター・キューまたは完全修飾キューのアクセス制御を検査するには、この属性を設定します。

RQMName

リモート側でホストされるキューのアクセス制御について検査されるプロファイルは、指定されたキューのプロファイルまたは指定されたキュー・マネージャーのプロファイルです。

Xmitq

リモート側でホストされるキューのアクセス制御について検査されるプロファイルは、SYSTEM.CLUSTER.TRANSMIT.QUEUE に解決されます。

Xmitq はデフォルト値です。

Windows **GroupModel=GlobalGroups**

この属性は、Windows 上でユーザーのグループ・メンバーシップを判別する際に、OAM がグローバル・グループを検査するかどうかを決定します。

デフォルトでは、グローバル・グループを検査しません。

GlobalGroups
OAM はグローバル・グループを検査します。

GlobalGroups を設定すると、許可コマンド **setmqaut**、 **dspmqaut**、および **dmpmqaut** はグローバル・グループ名を受け入れます。 **setmqaut** -g パラメーターを参照してください。

注 : ClusterQueueAcessControl=RQMName を設定したときに、許可サービスのカスタム実装が MQZAS_VERSION_6 未満である場合は、キュー・マネージャーは開始しません。 この場合には、ClusterQueueAcessControl=Xmitq を設定するか、またはカスタム許可サービスを MQZAS_VERSION_6 以上にアップグレードしてください。

スタンザの例

```
Security:
      ClusterQueueAccessControl=Xmitq
      GroupModel=GlobalGroups
```

Multi qm.ini ファイルの Service スタンザ

サービス・スタンザは、インストール可能サービスに変更を加えるために使用されます。 このスタンザには、サービスの名前と、サービスに定義されているエントリー・ポイントの数が含まれます。

注 : Windows Linux インストール可能サービスとそのコンポーネントを変更することには、大きな影響があります。 このため、IBM MQ Explorer では、インストール可能サービスは読み取り専用です。

1 つのサービス内の各コンポーネントについて、そのコンポーネントのコードを含むモジュールの名前とパスも指定する必要があります。 これには、 ServiceComponent スタンザを使用します。

Service および **ServiceComponent** スタンザは任意の順序で指定できます。スタンザにあるスタンザ・キーも、任意の順序で指定できます。 これらのスタンザの両方に、すべてのスタンザ・キーが存在する必要があります。 スタンザ・キーが重複する場合は、最後のキーが使用されます。

起動時に、キュー・マネージャーは構成ファイル内の各サービス・コンポーネント項目を順番に処理します。 その後、指定のコンポーネント・モジュールをロードし、コンポーネントのエントリー・ポイント (これはコンポーネントの初期化のエントリー・ポイントでなければなりません) を呼び出して、それに構成ハンドルを渡します。

名前 = AuthorizationService (デフォルト) |NameService
必要なサービスの名前。

AuthorizationService
IBM MQ の場合、 **AuthorizationService** コンポーネントはオブジェクト権限マネージャー (OAM) と呼ばれます。 **Service** スタンザおよびその関連 **ServiceComponent** スタンザは、キュー・マネージャーの作成時に自動的に追加されますが、 *MQSNOAUT* 環境変数によってオーバーライドすることができます。 その他の **ServiceComponent** スタンザは、手動で追加します。

Linux AIX 以下の qm.ini ファイル内のスタンザの例では、 IBM MQ for AIX 上に 2 つの許可サービス・コンポーネントを定義します。 *MQ_INSTALLATION_PATH* は、IBM MQ がインストールされている上位ディレクトリーを表します。

```
 Service:
    Name=AuthorizationService
    EntryPoints=13

 ServiceComponent:
    Service=AuthorizationService
    Name=MQSeries.UNIX.auth.service
Module=MQ_INSTALLATION_PATH/lib/amqzfu
    ComponentDataSize=0

 ServiceComponent:
    Service=AuthorizationService
```

```
    Name=user.defined.authorization.service
    Module=/usr/bin/udas01
    ComponentDataSize=96
```

Linux AIX ServiceComponent スタンザ `MQSeries.UNIX.auth.service` は、デフォルトの許可サービス・コンポーネントである OAM を定義します。 このスタンザを削除してキュー・マネージャーを再始動すると、OAM は使用不可となり、許可検査は行われません。

注意 : Windows Windows レジストリには、ファイルの場所、ドライバ、起動プロセスなどの重要な情報が含まれています。 IBM MQ が正しく機能するためには、これらの特定のレジストリ設定に依存している。 適切な情報なしにこれらの設定を変更すると、 IBM MQ アプリケーションが使用できなくなったり、完全に機能しなくなったりすることがあります。 不適切な変更は、アプリケーションの基本的な機能を停止させる可能性があるからです。

Windows さらに、**SecurityPolicy** 属性も IBM MQ サービスを使用して追加できます。 **SecurityPolicy** 属性は、サービス・スタンザに指定されているサービスが許可サービス、つまりデフォルトの OAM である場合にのみ適用されます。 **SecurityPolicy** 属性は、各キュー・マネージャーの セキュリティー・ポリシーを指定するために使用できます。 指定できる値は以下のとおりです。

デフォルト
デフォルトのセキュリティー・ポリシーを有効にする場合は、`Default` を指定します。 ある特定のユーザー ID の Windows セキュリティー ID (NT SID) が OAM に渡されなかった場合は、関連のセキュリティー・データベース内の検索によって、該当する SID が取得されます。

NTSIDsRequired
セキュリティー検査を行うときに、NT SID を OAM に渡す必要があります。

Windows ServiceComponent スタンザ `MQSeries.WindowsNT.auth.service` は、デフォルトの許可サービス・コンポーネントである OAM を定義します。 このスタンザを削除してキュー・マネージャーを再始動すると、OAM は使用不可となり、許可検査は行われません。

NameService
デフォルトではネーム・サービスが提供されていません。 ネーム・サービスが必要な場合は、NameService スタンザを手動で追加する必要があります。

Linux AIX 以下のネーム・サービスの AIX and Linux `qm.ini` ファイル・スタンザの例では、(架空の) ABC 社によって提供されるネーム・サービス・コンポーネントを指定します。

```
# Stanza for name service
 Service:
    Name=NameService
    EntryPoints=5

# Stanza for name service component, provided by ABC
 ServiceComponent:
    Service=NameService
    Name=ABC.Name.Service
    Module=/usr/lib/abcname
    ComponentDataSize=1024
```

注 : Windows Windows システムでは、 NameService スタンザ情報はレジストリーに保管されます。

EntryPoints=*number-of-entries*
このサービス用として定義するエントリー・ポイントの数。

これには、初期化エントリー・ポイントと終了エントリー・ポイントも含まれています。

Windows **SecurityPolicy= Default|NTSIDsRequired**
Windows システムでは、**SecurityPolicy** 属性が適用されるのは、指定されたサービスがデフォルトの許可サービス、つまり OAM である場合のみです。 **SecurityPolicy** 属性は、各キュー・マネージャーの セキュリティー・ポリシーを指定するために使用できます。

指定できる値は以下のとおりです。

デフォルト
有効にするデフォルトのセキュリティー・ポリシーを使用します。 ある特定のユーザー ID の Windows セキュリティー ID (NT SID) が OAM に渡されなかった場合は、関連のセキュリティー・データベース内の検索によって、該当する SID が取得されます。

NTSIDsRequired
セキュリティー検査を行うときに、NT SID を OAM に渡します。

詳しくは、 Windows セキュリティー ID (SID)を参照してください。

許可サービス・スタンザーの構成: Windows システムも参照してください。

Linux AIX **SecurityPolicy=ユーザー|グループ|ユーザー外部|デフォルト**
AIX and Linux システムでは、この値は、キュー・マネージャーがユーザー・ベースの許可を使用するか、グループ・ベースの許可を使用するかを指定します。 値には大/小文字の区別はありません。

値は、次の値のうちのいずれかです。

group
キュー・マネージャーはグループ・ベースの許可を使用します。 リソースにアクセスする権限は、グループに付与されます。

ユーザーは、そのユーザーが属する各グループに付与されたすべての権限の集合を受け取ります。

ユーザー ID とグループは、ローカル・オペレーティング・システムに対して定義する必要があります。

ユーザー
キュー・マネージャーはユーザー・ベースの許可を使用します。 リソースにアクセスする権限は、グループまたは特定のユーザー ID に付与できます。

ユーザーは、以下の権限の集合を受け取ります。

- 特定のユーザーに付与される権限。
- ユーザーが所属する各グループに付与される権限。

ユーザー ID とグループは、ローカル・オペレーティング・システムに対して定義する必要があります。

V 9.3.0 **UserExternal**
キュー・マネージャーはユーザー・ベースの許可を使用します。 ただし、ローカル・オペレーティング・システムに認識されていないユーザー ID に権限を付与することはできます。

リソースにアクセスする権限は、グループまたは特定のユーザー ID に付与できます。

ユーザーは、以下の権限の集合を受け取ります。

- 特定のユーザーに付与される権限。
- ユーザーが所属する各グループに付与される権限。

ユーザーがローカル・オペレーティング・システムに認識されていない場合、そのユーザーは nobody グループにのみ属していると見なされます。 グループについて詳しくは、AIX, Linux, and Windows のプリンシパルおよびグループを参照してください。 ユーザー ID は than12 以内で、オブジェクトの命名規則 IBM MQ に従わなければなりません。

現在の構成を失うことなく、この追加オプションを使用するように既存のキュー・マネージャーを変更できます。

V 9.3.4 これは、 AuthToken スタンザが指定されている場合のデフォルト値です。

default
キュー・マネージャーはグループ・ベースの許可を使用します。 この動作は、 `group` オプションの場合と同じです。

これは、 AuthToken スタンザが指定されていない場合のデフォルト値です。

属性値に対する変更を有効にするには、キュー・マネージャーを再始動します。

注 : V 9.3.4 Linux AIX IBM MQ 9.3.4 以降、 AuthToken スタンザが指定されている場合、Service スタンザの **SecurityPolicy** 属性の有効値は `UserExternal` に設定されます。 Service スタンザで **SecurityPolicy** が明示的に `Group` に設定されている場合、トークン認証は使用できません。 **SecurityPolicy** が `Group` に設定されている場合は、Service スタンザから **SecurityPolicy** 属性を削除してから、キュー・マネージャーを再始動します。 詳細については、112 ページの『qm.ini ファイルの AuthToken スタンザ』を参照してください。

SharedBindingsUserId=*user-type*

SharedBindingsUserId 属性は、指定されたサービスがデフォルトの承認サービス、つまり OAM である場合にのみ適用されます。 **SharedBindingsUserId** 属性は、共有バインディングとの関係でのみ使用されます。 この値により、MQZ_AUTHENTICATE_USER 関数から、*IdentityContext* 構造体の *UserIdentifier* フィールドを有効ユーザー ID と実ユーザー ID のどちらかに指定できます。

MQZ_AUTHENTICATE_USER 関数については、MQZ_AUTHENTICATE_USER - ユーザーの認証を参照してください。

指定できる値は以下のとおりです。

デフォルト
UserIdentifier フィールドの値は、実ユーザー ID として設定されます。

Real
UserIdentifier フィールドの値は、実ユーザー ID として設定されます。

Effective
UserIdentifier フィールドの値は、有効ユーザー ID として設定されます。

FastpathBindingsUserId=*user-type*

FastpathBindingsUserId 属性は、指定されたサービスがデフォルトの承認サービス、つまり OAM である場合にのみ適用されます。 **FastpathBindingsUserId** 属性は、ファースト・パス・バインディングとの関係でのみ使用されます。 この値により、MQZ_AUTHENTICATE_USER 関数から、*IdentityContext* 構造体の *UserIdentifier* フィールドを有効ユーザー ID と実ユーザー ID のどちらかに指定できます。

MQZ_AUTHENTICATE_USER 関数については、MQZ_AUTHENTICATE_USER - ユーザーの認証を参照してください。

指定できる値は以下のとおりです。

デフォルト
UserIdentifier フィールドの値は、実ユーザー ID として設定されます。

Real
UserIdentifier フィールドの値は、実ユーザー ID として設定されます。

Effective
UserIdentifier フィールドの値は、有効ユーザー ID として設定されます。

IsolatedBindingsUserId=*user-type*

IsolatedBindingsUserId 属性は、指定されたサービスがデフォルトの承認サービス、つまり OAM である場合にのみ適用されます。 **IsolatedBindingsUserId** 属性は、分離されたバインディングとの関係でのみ使用されます。 この値により、MQZ_AUTHENTICATE_USER 関数から、*IdentityContext* 構造体の *UserIdentifier* フィールドを有効ユーザー ID と実ユーザー ID のどちらかに指定できます。

MQZ_AUTHENTICATE_USER 関数については、MQZ_AUTHENTICATE_USER - ユーザーの認証を参照してください。

指定できる値は以下のとおりです。

デフォルト
UserIdentifier フィールドの値は、有効ユーザー ID として設定されます。

Real
UserIdentifier フィールドの値は、実ユーザー ID として設定されます。

Effective
UserIdentifier フィールドの値は、有効ユーザー ID として設定されます。

インストール可能なサービスとコンポーネントの詳細については、AIX, Linux, and Windows 用のインストール可能なサービスとコンポーネントを参照してください。

セキュリティー・サービス全般について詳しくは、AIX and Linux システムでのセキュリティーのセットアップを参照してください。

スタンザの例

```
Service:
   Name=AuthorizationService
   EntryPoints=14
```

関連概念

AIX、 Linux、および Windows 用のインストール可能サービスとコンポーネント

関連資料

IBM i でのインストール可能サービスとコンポーネント

インストール可能サービスの参照情報

Multi qm.ini ファイルの ServiceComponent スタンザ

ServiceComponent スタンザは、サービス・コンポーネントの情報を指定します。 新しいインストール可能サービスを追加する場合は、サービス・コンポーネント情報を指定する必要があります。 デフォルトでは、許可サービス・スタンザが事前に提供され、関連するコンポーネントである OAM がアクティブになっています。

Service および **ServiceComponent** スタンザは任意の順序で指定できます。スタンザにあるスタンザ・キーも、任意の順序で指定できます。 これらのスタンザの両方に、すべてのスタンザ・キーが存在する必要があります。 スタンザ・キーが重複する場合は、最後のキーが使用されます。

起動時に、キュー・マネージャーは構成ファイル内の各サービス・コンポーネント項目を順番に処理します。 その後、指定のコンポーネント・モジュールをロードし、コンポーネントのエントリー・ポイント (これはコンポーネントの初期化のエントリー・ポイントでなければなりません) を呼び出して、それに構成ハンドルを渡します。

Service=*service_name*
: 必要なサービスの名前。 これは、サービス構成情報の Name 属性に指定されている値に一致していることが必要です。

Name=*component_name*
: サービス・コンポーネントの記述名を指定します。 これは、固有名でなければならず、しかも IBM MQ オブジェクトの名前 (例えばキュー名) として有効な文字のみが使用されていなければなりません。 この名前は、サービスで生成されるオペレーター・メッセージに表示されます。 この名前は、企業の商標やそれに類似した識別性の高いストリングで 始めることをお勧めします。

Module=*module_name*
: このコンポーネントのコードを含むモジュールの名前。 これは、絶対パス名でなければなりません。

ComponentDataSize=*size*
: 各呼び出しでコンポーネントに渡される コンポーネント・データ域のバイト単位のサイズ。 コンポーネント・データが必要ない場合は、ゼロを指定します。

スタンザの例

```
ServiceComponent:
   Service=AuthorizationService
   Name=MQSeries.UNIX.auth.service
   Module=amqzfu
   ComponentDataSize=0
```

注意 : Windows Windows レジストリには、ファイルの場所、ドライバ、起動プロセスなどの重要な情報が含まれています。 IBM MQ が正しく機能するためには、これらの特定のレジストリ設定に依存している。 適切な情報なしにこれらの設定を変更すると、IBM MQ アプリケーションが使用できなくなったり、完全に機能しなくなったりすることがあります。不適切な変更は、アプリケーションの基本的な機能を停止させる可能性があるからです。

AuthorizationService スタンザとそれに関連する ServiceComponent スタンザ、および NameService スタンザとそれに関連する ServiceComponent スタンザを示すその他の例については、 141 ページの『qm.ini ファイルの Service スタンザ』を参照してください。

関連概念

AIX、 Linux、および Windows 用のインストール可能サービスとコンポーネント

関連資料

141 ページの『qm.ini ファイルの Service スタンザ』
サービス・スタンザは、インストール可能サービスに変更を加えるために使用されます。 このスタンザには、サービスの名前と、サービスに定義されているエントリー・ポイントの数が含まれます。

IBM i でのインストール可能サービスとコンポーネント

インストール可能サービスの参照情報

Windows qm.ini ファイルの SPX スタンザ (Windows のみ)

SPX スタンザは、SPX プロトコル構成パラメーターを指定します。 これらのパラメーターは、チャネルのデフォルト属性をオーバーライドします。

SPX プロトコル構成パラメーターを指定するには、 `qm.ini` ファイル内の SPX スタンザを使用します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer SPX キュー・マネージャーのプロパティー・ページを使用します。

Socket = 5E86 (デフォルト) | *socket_number*
16 進表記の SPX ソケット番号。 デフォルトは X'5E86' です。

BoardNum = 0 (デフォルト) | *adapter_number*
LAN アダプターの番号。 デフォルトはアダプター 0 です。

KeepAlive=NO|YES
KeepAlive 機能のオン/オフを切り替えます。

KeepAlive=YES を指定すると、SPX は、接続の相手側がまだ使用可能かどうかを定期的に検査します。有効でない場合、チャネルは閉じた状態です。

Library1= *DLLName1*
SPX DLL の名前。

デフォルトは WSOCK32.DLL です。

Library2= *DLLName2*
LibraryName1 と同じもので、コードが 2 つの異なるライブラリーに保管されている場合に使用されます。

デフォルトは WSOCK32.DLL です。

ListenerBacklog=number
SPX リスナーの場合のデフォルトの未解決要求数を指定変更します。

SPX で受信するときに、未解決の接続要求の最大数を設定します。 これは、SPX ソケット上でリスナーからの受け入れを待っている要求のバックログと考えることができます。 デフォルトのリスナー・バックログ値は、146 ページの表 13 に示すとおりです。

表 *13.* デフォルトの未解決接続要求数 *(SPX)*

プラットフォーム	**ListenerBacklog のデフォルト値**
Windows Server	100

表 13. デフォルトの未解決接続要求数 (SPX) (続き)	
プラットフォーム	**ListenerBacklog のデフォルト値**
Windows ワークステーション	5

注 : オペレーティング・システムの中には、表に示されているデフォルトより大きい値をサポートするものもあります。 接続限度に達しないようにするために、この表の値を使用してください。

これとは反対に、オペレーティング・システムの中には、SPX バックログのサイズを制限する可能性のあるものがあり、有効な SPX バックログは、ここで要求されるよりも小さい可能性があります。

バックログが 146 ページの表 13 に示されている値に達すると、SPX 接続は拒否され、チャネルは開始できなくなります。 メッセージ・チャネルの場合は、結果としてチャネルは RETRY 状態になり、後で接続を再試行することになります。 クライアント接続の場合は、クライアントは、MQCONN から理由コード MQRC_Q_MGR_NOT_AVAILABLE を受け取り、後で接続を再試行することが必要になります。

Multi qm.ini ファイルの SSL スタンザ

SSL スタンザは、キュー・マネージャー上の TLS チャネルを構成するために使用されます。

Online Certificate Status Protocol (OCSP)

証明書には、AuthorityInfoAccess 拡張を含めることができます。 この拡張は、サーバーへの通信に Online Certificate Status Protocol (OCSP) を使用することを指定します。 ご使用のキュー・マネージャーの SSL チャネルまたは TLS チャネルが AuthorityInfoAccess 拡張を使用できるようにするには、それらの拡張で指定されている OCSP サーバーが使用可能で、正しく構成されており、ネットワークを通じてアクセスできることを確認してください。 詳しくは、失効した証明書の取り扱いを参照してください。

CrlDistributionPoint (CDP)

証明書には、CrlDistributionPoint 拡張を含めることができます。 この拡張には、証明書取り消しリスト (CRL) のダウンロードに使用されるプロトコルと通信先のサーバーの両方を識別する URL が含まれます。

ご使用のキュー・マネージャーの SSL チャネルまたは TLS チャネルが CrlDistributionPoint 拡張を使用できるようにするには、それらの拡張で指定されている CDP サーバーが使用可能で、正しく構成されており、ネットワークを通じてアクセスできることを確認してください。

SSL スタンザの属性

qm.ini ファイル内の SSL スタンザを使用して、キュー・マネージャー上の TLS チャネルが以下の機能を使用しようとする方法、およびそれらを使用するときに問題が発生した場合の対応方法を構成します。

以下の各ケースでは、指定された値がリストされている有効な値のいずれでもない場合、デフォルトの値が使用されます。 無効な値が指定されたことを示すエラー・メッセージは書き込まれません。

V 9.3.0 **OutboundSNI = CHANNEL | HOSTNAME**

OutboundSNI がチャネルに設定されている場合、SNI 対応クライアントは、TLS 接続の開始時に SNI をリモート・システムのターゲット IBM MQ チャネル名に設定します。

この属性が HOSTNAME に設定されている場合、SNI 対応クライアントは、アウトバウンド接続要求によって TLS ハンドシェーク時にリモート・キュー・マネージャーのデフォルト証明書を受信するように SNI ヘッダーをホスト名に設定します。 したがって、チャネルごとの証明書は使用できません。

注 : OutboundSNI=HOSTNAME を使用して、証明書ラベルが構成されているリモート・チャネルに接続する場合、接続は MQRC_SSL_INITIALIZATION_ERROR で拒否され、 AMQ9673 メッセージがリモート・キュー・マネージャーのエラー・ログに出力されます。

AllowOutboundSNI=YES（デフォルト） | NO

使用可能にすると、SNI 対応クライアントは、TLS 接続の開始時に、SNI をリモート・システムへのターゲット IBM MQ チャネル名に設定します。 この属性が NO に設定されている場合、SNI 対応クライア

ントは、アウトバウンド接続要求によって TLS ハンドシェーク時にリモート・キュー・マネージャーのデフォルト証明書を受信するように SNI ヘッダーを設定しません。したがって、チャネルごとの証明書は使用できません。

重要 : Deprecated V 9.3.0 IBM MQ 9.3.0 以降、**AllowOutboundSNI** プロパティーは非推奨になり、後方互換性の目的でのみ使用できます。

はいに設定された **AllowOutboundSNI** は、チャネルに設定された **OutboundSNI** と同じ機能を提供します。一方、いいえに設定された **AllowOutboundSNI** は、ホスト名に設定された **OutboundSNI** と同じ機能を提供します。

AllowOutboundSNI 属性と **OutboundSNI** 属性の両方が SSL スタンザに存在する場合は、**OutboundSNI** の設定が優先されます。

AllowedCipherSpecs=*name*|*name list*|ALL

Multiplatforms の IBM MQ チャネルで順序付けされ、使用可能な CipherSpec のカスタム・リストを指定します。 **AllowedCipherSpecs** を指定すると、 IBM MQ リスナーが CipherSpecs のいずれかを使用しない限り、入ってくるチャンネル開始リクエストを受け付けないように制限される。

- 単一の CipherSpec 名。
- 再有効化する IBM MQ CipherSpec 名のコンマ区切りリスト。
- すべての CipherSpec を表す特殊値 ALL (推奨されません)。

注 : ALL CipherSpecs を選択しないでください。これを選択すると、SSL 3.0 プロトコルと TLS 1.0 プロトコル、および多数の脆弱な暗号アルゴリズムが有効になります。

詳しくは、TLS ハンドシェークでの CipherSpec の順序の「IBM MQ for Multiplatforms での順序付けと有効化された CipherSpecs のカスタム・リストの提供」を参照してください。

IBM i ALW **AllowTLSV13 =Y | YES | T | TRUE (デフォルト) | N | NO | F | FALSE**

キュー・マネージャーが TLS 1.3 CipherSpec を使用できるかどうかを指定します。

- Y (デフォルト)、 YES (デフォルト)、 T (デフォルト)、または TRUE (デフォルト): TLS 1.3 を有効にして、キュー・マネージャーが TLS 1.3 CipherSpecs を使用できるようにします。
- N、NO、F、または FALSE: TLS 1.3 を無効にします。 これは、キュー・マネージャーが TLS 1.3 CipherSpec を使用できないことを意味します。

詳しくは、CipherSpec の有効化を参照してください。

CDPCheckExtensions=YES|NO（デフォルト）

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

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

ALW **MinimumRSAKeySize=int**

TLS のハンドシェーク中に RSA 証明書を受け入れるために必要な最小の鍵サイズを指定します。 0 以上の任意の値が許可されます。 指定しない場合は、デフォルトの 1 になります。

OCSPAuthentication=REQUIRED (デフォルト) | WARN | OPTIONAL

取り消し状況を OCSP サーバーから判別できない場合に実行されるアクションを指定します。

OCSP 検査が使用可能になっている場合、TLS のチャネル・プログラムは、OCSP サーバーとの通信を試行します。

チャネル・プログラムがどの OCSP サーバーとも通信できない場合、あるいは証明書の取り消し状況を提供できるサーバーがない場合は、OCSPAuthentication パラメーターの値が使用されます。

- REQUIRED (デフォルト): 取り消し状況を判別できない場合、接続はエラーでクローズされます。 この値はデフォルトです。
- WARN: 取り消し状況の判別に失敗すると、キュー・マネージャーのエラー・ログに警告メッセージを書き込みますが、接続の続行は許可します。
- OPTIONAL: 取り消し状況の判別に失敗しても、通知なしで接続の続行を許可します。 警告やエラーは生成されません。

OCSPCheckExtensions = YES (デフォルト) | いいえ

このキュー・マネージャーの TLS チャネルおよび TLS チャネルが、AuthorityInfoAccess 証明書拡張で指定されている OCSP サーバーの検査を試行するかどうかを指定します。

- YES (デフォルト): TLS チャネルは、デジタル証明書が失効しているかどうかを判別するために OCSP サーバーの検査を試行します。 この値はデフォルトです。
- NO: TLS チャネルおよび TLS チャネルは、OCSP サーバーの検査を試行しません。

ALW **OCSPTimeout= *number***

失効検査の実行時に OCSP 応答側を待機する秒数。

IBM MQ 9.3.0 以降では、値 0 が設定されている場合、デフォルトのタイムアウトである 30 秒が使用されます。

値を設定しない場合は、IBM MQ のデフォルトの 30 秒が使用されます。

SSLHTTPProxyName=*string*

この文字列は、 IBM Global Security Kit (GSKit) が OCSP チェックに使用する HTTP プロキシサーバーのホスト名またはネットワークアドレスのいずれかである。 このアドレスの後に、オプションのポート番号が括弧で囲まれて続く場合もあります。 ポート番号を指定しないと、デフォルトの HTTP ポートである 80 が使用されます。

AIX 32 ビット・クライアントでは AIX の上で、ネットワーク・アドレスは IPv4 アドレスのみとなります。

その他のプラットフォームでは、ネットワーク・アドレスには IPv4 または IPv6 アドレスを使用できます。

例えば、ファイアウォールが OCSP 応答側の URL へのアクセスを回避する場合など、この属性が必要になる場合があります。

ALW **PeerCertChainValidation=*string***

このストリングは、次のいずれかの値に設定できます。

- Usepeerchain **[デフォルト]**: ピアによって提供される証明書チェーンを使用して、証明書の検証時にトラスト・チェーンのギャップを埋めることができます。 ルート証明書は例外です。
- Truststoreonly **[推奨されません]**: トラストストア内の証明書のみがピアの証明書の検証に使用されます。

ALW **SSLHTTPConnectTimeout = *number*|0**

失効検査の実行時に HTTP サーバーへのネットワーク接続が正常に確立されるのを待機する秒数。

値を設定しない場合は、IBM MQ のデフォルトの 0 (オフ) が使用されます。

スタンザの例

```
SSL:
       OutboundSNI=CHANNEL
      AllowedCipherSpecs=TLS13 CipherSpec list
      AllowTLSV13=Y
      CDPCheckExtensions=NO
      MinimumRSAKeySize=1
      OCSPAuthentication=REQUIRED
      OCSPCheckExtensions=YES
      OCSPTimeout=30
```

```
        PeerCertChainValidation=Usepeerchain
        SSLHTTPConnectTimeout=0
```

注:

- **OutboundSNI** のデフォルト値は **Channel** です。
- **TLS13 CipherSpec** リストは、別名暗号ではなく、特定の CipherSpecs のリストです。 TLS1.3 暗号のみが必要な場合は、それらをリストする必要があります。 以下に例を示します。

  TLS_CHACHA20_POLY1305_SHA256
  TLS_AES_256_GCM_SHA384
  TLS_AES_128_GCM_SHA256
  TLS_AES_128_CCM_SHA256
  TLS_AES_128_CCM_8_SHA256
- 弱い暗号を使用可能にしていない限り、 **AllowTLSV13** のデフォルト値は Y です。弱い暗号を使用可能にしている場合は、(明示的にオンにしない限り) オフになります。
- **CDPCheckExtensions** の値は、 「はい」 または 「いいえ」のみです。
- **PeerCertChainValidation** の値は、 Usepeerchain または Truststoreonly のみです。

Multi qm.ini ファイルの Subpool スタンザ

このスタンザは、IBM MQ によって作成されます。 変更はしないでください。

Subpool スタンザとそのスタンザ内の属性 **ShortSubpoolName** は、キュー・マネージャーを作成するときに、IBM MQ によって自動的に作成されます。 IBM MQ は **ShortSubpoolName** の値を選択します。 この値は変更しないでください。

この名前は、 /var/mqm/sockets ディレクトリー内に作成されたディレクトリーおよびシンボリック・リンクに対応します。 IBM MQ は、実行中のプロセス間の内部通信にこのディレクトリーを使用します。

Multi qm.ini ファイルの TCP スタンザ

TCP スタンザは、伝送制御プロトコル/Internet Protocol (TCP/IP) 構成パラメーターを指定します。 これらのパラメーターは、チャネルのデフォルト属性をオーバーライドします。

qm.ini ファイル内の TCP スタンザを使用して、TCP/IP 構成パラメーターを指定します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer SPX TCP キュー・マネージャーのプロパティー・ページを使用します。

ポート = 1414 (デフォルト) | *port_number*
TCP/IP セッション用の 10 進表記のデフォルトのポート番号。 IBM MQ の 事前割り当て ポート番号は 1414 です。

Windows **Library1= *DLLName1* (Windows のみ)**
TCP/IP ソケット DLL の名前。

デフォルトは WSOCK32 です。

V 9.3.0 Multi **SecureCommsOnly=NO|N|FALSE|F(デフォルト|TRUE|T|YES|Y)**
非暗号化テキストの通信を許可するかどうかを指定します。

SecureCommsOnly=NO|N|FALSE|F
非暗号化テキストの通信が許可され、キュー・マネージャーの開始時に警告メッセージが出力されます。

SecureCommsOnly=YES|Y|TRUE|T
非暗号化テキストの通信は許可されません。また、キュー・マネージャーの開始時に情報メッセージが出力されます。

KeepAlive = NO (デフォルト) | はい
KeepAlive 機能のオン/オフを切り替えます。 KeepAlive=YES を指定すると、TCP/IP は、接続の相手側がまだ使用可能かどうかを定期的に検査します。 有効でない場合、チャネルは閉じた状態です。

ListenerBacklog=number
TCP/IP リスナーの場合のデフォルトの未解決要求数を指定変更します。

TCP/IP で受信するときに、未解決の接続要求の最大数を設定します。 これは、TCP/IP ポート上で listener からの受け入れを待っている要求のバックログと考えることができます。 デフォルトのリスナー・バックログ値は、151 ページの表 14 に示すとおりです。

表 *14.* デフォルトの未解決接続要求数 *(TCP)*

プラットフォーム	ListenerBacklog のデフォルト値
Windows Windows サーバー	100
Linux Linux	100
AIX AIX V5.3 以降	100

注 : オペレーティング・システムの中には、表に示されているデフォルトより大きい値をサポートするものもあります。 接続限度に達しないようにするために、この表の値を使用してください。

これとは反対に、オペレーティング・システムの中には、TCP バックログのサイズを制限する可能性のあるものがあり、有効な TCP バックログは、ここで要求されるよりも小さい可能性があります。

バックログが 151 ページの表 14 に示されている値に達すると、TCP/IP 接続は拒否され、チャネルは開始できなくなります。 メッセージ・チャネルの場合は、結果としてチャネルは RETRY 状態になり、後で接続を再試行することになります。 クライアント接続の場合は、クライアントは、MQCONN から理由コード MQRC_Q_MGR_NOT_AVAILABLE を受け取り、後で接続を再試行します。

以下のプロパティーのグループは、TCP/IP によって使用されるバッファーのサイズを制御するために使用できます。 値は、オペレーティング・システムの TCP/IP 層に直接渡されます。 これらのプロパティーを使用する際は、十分注意する必要があります。 値が正しく設定されないと、TCP/IP のパフォーマンスに悪影響が及ぶ可能性があります。 値が正しく設定されない場合にパフォーマンスに及ぶ影響について詳しくは、ご使用の環境の TCP/IP の資料を参照してください。 値 0 は、IBM MQ によってバッファー・サイズが修正されるのではなく、オペレーティング・システムがバッファー・サイズを管理することを意味します。

Connect_Timeout = 0 (デフォルト) |number
ソケットへの接続試行がタイムアウトになるまでの秒数。 デフォルト値である 0 は、接続のタイムアウトがないことを指定します。

IBM MQ チャネル・プロセスは非ブロッキング・ソケットを介して接続します。 このため、ソケットのもう一方の側の準備ができていない場合、connect() は即時に *EINPROGRESS* または *EWOULDBLOCK* を返します。 これに続いて接続が最大 20 回まで再試行された後、通信エラーがレポートされます。

Connect_Timeout がゼロ以外の値に設定されている場合、IBM MQ は、ソケットの準備ができるまで、select() 呼び出しに対して規定された期間待機します。 これにより、後続の connect() 呼び出しが成功する確率は上昇します。 このオプションは、ネットワークの負荷が高くなっていて、接続に多少の待機時間が必要なときに役に立つ場合があります。

SndBuffSize = number |0 (デフォルト)
チャネルの送信側で使用される TCP/IP 送信バッファーのサイズ (バイト)。 このスタンザ値は、このチャネル・タイプをさらに具体的に指定するスタンザ (例えば RcvSndBuffSize) でオーバーライドされる場合があります。 値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。 値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

Multi IBM MQ 8.0 以降、新規キュー・マネージャーは自動的に 0 にデフォルト設定された状態で作成されます (152 ページの『スタンザの例』 を参照してください)。

RcvBuffSize = number |0 (デフォルト)
チャネルの受信側で使用される TCP/IP 受信バッファーのサイズ (バイト)。 このスタンザ値は、このチャネル・タイプをさらに具体的に指定するスタンザ (例えば RcvRcvBuffSize) でオーバーライドされる場合があります。 値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。 値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

Multi IBM MQ 8.0 以降、新規キュー・マネージャーは自動的に 0 にデフォルト設定された状態で作成されます (152 ページの『スタンザの例』 を参照してください)。

RcvSndBuffSize = number |0 (デフォルト)
受信側チャネルの送信側で使用される TCP/IP 送信バッファーのサイズ (バイト)。 値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。 値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

Multi IBM MQ 8.0 以降、新規キュー・マネージャーは自動的に 0 にデフォルト設定された状態で作成されます (152 ページの『スタンザの例』 を参照してください)。

RcvRcvBuffSize = number |0 (デフォルト)
受信側チャネルの受信側で使用される TCP/IP 受信バッファーのサイズ (バイト)。 値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。 値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

Multi IBM MQ 8.0 以降、新規キュー・マネージャーは自動的に 0 にデフォルト設定された状態で作成されます (152 ページの『スタンザの例』 を参照してください)。

SvrSndBuffSize = number |0 (デフォルト)
クライアント接続チャネルとサーバー接続チャネルのサーバー側で使用される TCP/IP 送信バッファーのサイズ (バイト)。 値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。 値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

Multi IBM MQ 8.0 以降、新規キュー・マネージャーは自動的に 0 にデフォルト設定された状態で作成されます (152 ページの『スタンザの例』 を参照してください)。

SvrRcvBuffSize = number |0 (デフォルト)
クライアント接続チャネルとサーバー接続チャネルのサーバー側で使用される TCP/IP 受信バッファーのサイズ (バイト)。 値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。 値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

Multi IBM MQ 8.0 以降、新規キュー・マネージャーは自動的に 0 にデフォルト設定された状態で作成されます (152 ページの『スタンザの例』 を参照してください)。

スタンザの例

```
TCP:
   SndBuffSize=0
   RcvBuffSize=0
   RcvSndBuffSize=0
   RcvRcvBuffSize=0
   ClntSndBuffSize=0
   ClntRcvBuffSize=0
   SvrSndBuffSize=0
   SvrRcvBuffSize=0
```

注 : Multi Multiplatforms での新規キュー・マネージャーの場合、 `qm.ini file` の TCP スタンザ内のデフォルトの TCP 送受信バッファー・サイズは、オペレーティング・システムによって管理されるように設定されます。 前の例に示すように、新しいキュー・マネージャーは、送信バッファーと受信バッファーのデフォルト設定 0 で自動的に作成されます。 これは、新しいキュー・マネージャーにのみ適用されます。 以前のバージョンの IBM MQ からマイグレーションしたキュー・マネージャーの TCP 送受信バッファー設定は保持されます。

TCP バッファー・サイズ・プロパティーが qm.ini ファイルから削除されると、デフォルト・バッファーは 32K に設定されます。 すべてのメッセージング・シナリオで 32K が適切なバッファーではない可能性があるため、このデフォルトを使用する場合は注意が必要です。

TCP 送信および受信バッファーのプロパティーがゼロに設定されている場合は、OS のデフォルト値が使用されます。 これらのデフォルトを選択する方法はオペレーティング・システムによって異なりますが、通常は「tcp」または get/setsockopt () OS マニュアル・ページにあります。

Multi qm.ini ファイルの TuningParameters スタンザ

TuningParameters スタンザは、キュー・マネージャーをチューニングするためのオプションを指定します。

SuppressDspAuthFail= YES |NO (デフォルト)
YES に設定すると、キュー・マネージャーは、オブジェクトに対する + dsp 権限が接続にない場合、許可検査が失敗したときに、許可イベントの生成とエラー・ログへの AMQ8077 エラー・メッセージの書き込みを抑止します。

ImplSyncOpenOutput=*value*
ImplSyncOpenOutput は、書き込みのためにキューを開いておくアプリケーションの最小数です。その数に達すると、同期点以外で永続書き込みのための暗黙的同期点が有効になることがあります。 **ImplSyncOpenOutput** のデフォルト値は 2 です。

PUT 操作のためにキューを開いておくアプリケーションが 1 つだけの場合は、**ImplSyncOpenOutput** がオフに切り替えられます。

ImplSyncOpenOutput=1 を指定すると、暗黙の同期が常に考慮されます。 任意の正整数値を設定できます。 暗黙の同期を追加したくない場合は、ImplSyncOpenOutput=OFF を設定します。

UniformClusterName=*name of cluster*
ユニフォーム・クラスターとして使用している IBM MQ クラスターの名前。

OAMLdapConnectTimeout=time|0 (デフォルト)
LDAP クライアントがサーバーへの TCP 接続を確立するために待機する最大時間(秒単位)。 接続名前リストを介して、複数の LDAP サーバーを提供している場合、タイムアウトは個々の接続試行に適用されるため、このタイムアウトに達すると、名前リスト内の次のエントリーへの接続が試行されます。

time の最大値は 3600 秒で、デフォルト値だけでなく最小値も 0 という値は、待機が無制限であることを意味します。

OAMLdapQueryTimeLimit=time|0 (デフォルト)
接続が確立され、LDAP 要求が送信された後、LDAP クライアントがサーバーからの LDAP 要求への応答を受信するのを待機する最大時間を秒単位で指定します。

time の最大値は 3600 秒で、デフォルト値だけでなく最小値も 0 という値は、待機が無制限であることを意味します。

V 9.3.2 V 9.3.0.5 **OAMLdapResponseWarningTime=しきい値**
LDAP サーバーへの接続にかかった時間が、 **OAMLdapResponseWarningTime** パラメーターで指定されたしきい値 (秒数) を超えた場合、 AMQ5544W メッセージがエラー・ログに書き込まれます。 デフォルトのしきい値は 10 秒です。

ExpiryInterval
キュー・マネージャーがキューをスキャンして、他のキュー・アクティビティーによってまだクリーンアップされていない有効期限切れメッセージを探す頻度を示します。 これは秒単位の時間間隔です。

デフォルトでは、有効期限スキャナーは、実動 IBM MQ ビルドで約 5 分ごとに実行されます。

注意 : 通常、 **ExpiryInterval** 値を変更する必要はありません。この値は、 IBM サポートの指導の下でのみ変更してください。

LivenessHeartBeatLen
ログへの書き込みを行うキュー・マネージャー検査が適切な頻度で行われる頻度を構成します。 **LivenessHeartBeatLen** の最大値は 600 秒 (10 分) で、最小値は 0 です。 これは、チェックを完全に無効にする効果があります。

注意 : ほとんどの場合、これらのチェックの頻度を変更する必要はありません。 IBM サポートからの指示がない限り、変更を加えないでください。

ECHeartBeatLen
キュー・マネージャーの一般的なヘルス・チェックの頻度を構成します。 **ECHeartBeatLen** の最小値は 10000 ミリ秒 (10 秒) で、最大値は 60000 ミリ秒 (60 秒) です。

注意 : ほとんどの場合、これらのチェックの頻度を変更する必要はありません。 IBM サポートからの指示がない限り、変更を加えないでください。

FileLockHeartBeatLen
実行コントローラーが定期的に実行する複数インスタンス・キュー・マネージャーのファイル・ロック検査のデフォルト値を変更して、1 次複数インスタンス・ファイルの排他ロックを保持していることを確認します。 デフォルトでは、これらのファイル・ロック検査は 20 秒ごとに行われます。
FileLockHeartBeatLen の最小値は 10 秒で、最大値は 600 秒 (10 分) です。

注意 : ほとんどの場合、これらのチェックの頻度を変更する必要はありません。 IBM サポートからの指示がない限り、変更を加えないでください。

スタンザの例

V 9.3.2 V 9.3.0.5

```
TuningParameters:
       SuppressDspAuthFail=NO
       ImplSyncOpenOutput=2
       OAMLdapConnectTimeout=60
       OAMLdapQueryTimeLimit=60
       OAMLdapResponseWarningTime=10
       ExpiryInterval=300
```

関連概念
暗黙の同期点

Multi qm.ini ファイルの Variables スタンザ

変数スタンザは、自動均一クラスターで使用する構成変数を指定します。

Variables スタンザにリストされる属性は、クラスター受信側チャネルの CONNAME およびチャネル名 MQSC フィールドの自動クラスター構成時に使用できます。 構成変数は、MQSC スクリプトの他のエレメントでは使用できません。

attribute=value
MQSC の定義中に挿入するために使用する名前と関連値を指定します。

attribute=value のペアは、キュー・マネージャーの作成時に **crtmqm** コマンドで **-iv** コマンド行オプションを使用して指定できます。

スタンザの例

```
Variables:
       CONNAME=127.0.0.1(1414)
```

関連概念
412 ページの『自動アプリケーション・バランシング』
IBM MQ の均一クラスターによってクラスター内のアプリケーションの分散度を緊密に管理できるようになり、ランダム化や特定のキュー・マネージャーにアプリケーションを手動でピン留めする操作に依存しなくてよくなりました。その結果、自動アプリケーション・バランシングによるアプリケーションの分散度と可用性が大幅に高まっています。

関連タスク
425 ページの『均一クラスターの新規作成』
新しい均一クラスターの作成方法。

関連資料
429 ページの『自動クラスター構成の使用』
qm.ini コンフィギュレーション情報を変更することで、IBM MQ 自動コンフィギュレーションを有効にするように設定します。

Multi qm.ini ファイルの XAResourceManager スタンザ

XAResourceManager スタンザ。キュー・マネージャーによって調整されるグローバル作業単位に関与するリソース・マネージャーに関する情報を指定します。

qm.ini ファイル内の XAResourceManager スタンザを使用して、キュー・マネージャーによって調整されるグローバル作業単位に関与するリソース・マネージャーに関する情報を指定します。

Windows Linux あるいは、Linux (x86 および x86-64) および Windows では、IBM MQ Explorer XA リソース・マネージャーのキュー・マネージャー・プロパティー・ページを使用します。

グローバル作業単位に関係するリソース・マネージャーのインスタンスごとに XA リソース・マネージャー構成情報を手動で追加します。デフォルト値はありません。

リソース・マネージャーの属性について詳しくは、データベースの調整を参照してください。

Name=*name* (mandatory)
この属性は、リソース・マネージャーのインスタンスを識別します。

Name 値は長さが 31 文字までです。 XA スイッチ構造で定義されている リソース・マネージャーの名前を使用することができます。 ただし、同じリソース・マネージャーの 複数のインスタンスを使用している場合は、各インスタンスごとに固有の名前を作成する必要があります。 例えば、次のように、データベースの名前を Name ストリングに組み込むことによって、それぞれの名前を固有に保つことができます。

IBM MQ は、メッセージおよび dspmqtrn コマンドからの出力で Name 値を使用します。

関連するキュー・マネージャーが開始され、リソース・マネージャー名が有効になったら、リソース・マネージャーのインスタンスの名前変更、あるいは構成情報からの項目の削除は行わないでください。

SwitchFile=*name* (mandatory)
リソース・マネージャーの XA スイッチ構造を収容するロード・ファイルの完全修飾名。

64 ビットのキュー・マネージャーを 32 ビット・アプリケーションと一緒に使用する場合、 name 値に含まれるのは、リソース・マネージャーの XA スイッチ構造を含むロード・ファイルのベース名のみでなければなりません。

32 ビット・ファイルは、ExitsDefaultPath によって指定されたパスからアプリケーションへロードされます。

64 ビット・ファイルは、ExitsDefaultPath64 によって指定されたパスからキュー・マネージャーへロードされます。

XAOpenString=*string* (オプション)
リソース・マネージャーの xa_open エントリー・ポイントに渡されるデータのストリング。 このストリングの内容は、リソース・マネージャーによって異なります。 例えば、このリソース・マネージャーのインスタンスがアクセスする データベースを識別します。 この属性の定義方法については、以下のセクションを参照してください。

- Db2 のリソース・マネージャー構成情報の追加
- Oracle のリソース・マネージャー構成情報の追加
- Sybase 用のリソース・マネージャー構成情報の追加
- Informix®のリソース・マネージャー構成情報の追加

また、適切なストリングについては、リソース・マネージャーの資料を参照してください。

XACloseString=*string* (オプション)
リソース・マネージャーの xa_close エントリー・ポイントに渡されるデータのストリング。 このストリングの内容は、リソース・マネージャーによって異なります。 この属性の定義方法については、以下のセクションを参照してください。

- Db2 のリソース・マネージャー構成情報の追加
- Oracle のリソース・マネージャー構成情報の追加
- Sybase 用のリソース・マネージャー構成情報の追加
- Informix のリソース・マネージャー構成情報の追加

また、適切なストリングについては、データベースの資料を参照してください。

ThreadOfControl=THREAD|PROCESS
Windows この属性は、Windows に必須です。 キュー・マネージャーは、自身のマルチスレッド・プロセスの 1 つから リソース・マネージャーを呼び出す必要がある場合に、この値を使用してシリアライゼーションの処理をします。

スレッド
リソース・マネージャーは完全にスレッド認識型 です。 マルチスレッド IBM MQ プロセスでは、外部のリソース・マネージャーに対して複数のスレッドから XA 関数呼び出しを同時に行うことができます。

PROCESS
リソース・マネージャーはスレッド安全 ではありません。 マルチスレッド IBM MQ プロセスでは、リソース・マネージャーに対して行える XA 関数呼び出しは一度に 1 つのみです。

ThreadOfControl 項目は、マルチスレッド・アプリケーション・プロセスの キュー・マネージャーによって発行される XA 関数呼び出しには 適用されません。 通常、別々のスレッドに並行する複数の作業単位を持つアプリケーションは、各リソース・マネージャーでサポートされるためにこのモードの操作が必要となります。

スタンザの例

```
XAResourceManager:
   Name=DB2 Resource Manager Bank
   SwitchFile=/usr/bin/db2swit
   XAOpenString=MQBankDB
   XACloseString=
   ThreadOfControl=THREAD
```

注 : XAResourceManager スタンザの最大数は 255 に制限されています。 ただし、トランザクションの性能低下を防ぐため、少数のスタンザのみを使用してください。

IBM i IBM i 用の qm.ini ファイルの例

IBM i のキュー・マネージャー構成ファイルで属性のグループがどのように配置されるかを示す例。

```
#********************************************************************#
#* Module Name: qm.ini                                              *#
#* Type       : IBM MQ queue manager configuration file             *#
# Function    : Define the configuration of a single queue manager *#
#*                                                                  *#
#********************************************************************#
#* Notes      :                                                     *#
#* 1) This file defines the configuration of the queue manager     *#
#*                                                                  *#
#********************************************************************#
Log:
LogPath=QMSATURN.Q
LogReceiverSize=65536

CHANNELS:
MaxChannels = 20       ; Maximum number of channels allowed.
                       ; Default is 100.
MaxActiveChannels = 10 ; Maximum number of channels allowed to be
                       ; active at any time. The default is the
                       ; value of MaxChannels.
```

```
TCP:                    ; TCP/IP entries.
KeepAlive = Yes         ; Switch KeepAlive on.
SvrSndBuffSize=20000    ; Size in bytes of the TCP/IP send buffer for each
                        ; channel instance. Default is 32768.
SvrRcvBuffSize=20000    ; Size in bytes of the TCP/IP receive buffer for each
                        ; channel instance. Default is 32768.
Connect_Timeout=10000   ; Number of seconds before an attempt to connect the
                        ; channel instance times out. Default is zero (no timeout).

QMErrorLog:
ErrorLogSize = 262144
ExcludeMessage = 7234
SuppressMessage = 9001,9002,9202
SuppressInterval = 30

TuningParameters:
   ImplSyncOpenOutput=2
```

ALW インストール構成ファイル、mqinst.ini

AIX and Linux システムの場合、インストール構成ファイル `mqinst.ini` には、すべての IBM MQ インストールに関する情報が含まれています。 Windows システムでは、インストール構成情報はレジストリーにあります。

mqinst.ini ファイルのロケーション

Linux AIX

`mqinst.ini` ファイルは、AIX and Linux システム上の`/etc/opt/mqm` ディレクトリーにあります。 これには、どのインストール済み環境がプライマリーのインストール済み環境 (もしあれば) であるかが、各インストール済み環境に関する以下の情報とともに書き込まれています。

- インストール名
- インストールの説明
- インストール ID
- インストール・パス

重要: `mqinst.ini` ファイルは、フォーマットが固定されておらず、変更される可能性があるため、編集または参照することはできません。

内部でのみ使用されるインストール ID は、自動的に設定され、変更することはできません。

`mqinst.ini` ファイルを直接編集する代わりに、以下のコマンドを使用して、ファイル内の値を作成、削除、照会、および変更する必要があります:

crtmqinst は、項目を作成します。
dltmqinst は、項目を削除します。
dspmqinst は、項目を表示します。
setmqinst は、項目を設定します。

Windows でのインストール構成情報

Windows

Windows 上に `mqinst.ini` ファイルがありません。 インストール構成情報はレジストリーにあり、以下のキーで保持されます。

```
HKLM\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName
```

重要: このキーの形式は固定されておらず、変更される可能性があるため、このキーは編集したり、直接参照したりしないでください。

その代わりに、以下のコマンドを使用して、レジストリーの値を照会および変更する必要があります。

dspmqinst は、項目を表示します。

setmqinst は、項目を設定します。

Windows では、**crtmqinst** コマンドおよび **dltmqinst** コマンドは使用できません。 必要なレジストリー項目の作成および削除は、インストール・プロセスおよびアンインストール・プロセスによって処理されます。

Multi IBM MQ MQI client 構成ファイル、 mqclient.ini

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

IBM MQ MQI clients は、キュー・マネージャー構成ファイル qm.ini と同様に、テキスト・ファイルを使用して構成します。 ファイルには多数のスタンザがあり、各スタンザには **attribute-name**=*value* 形式の行が多数含まれています。

IBM MQ MQI client 通常、構成ファイルは mqclient.ini という名前になりますが、別の名前を付けることを選択できます。 このファイルの構成情報は、以下のプラットフォームに適用されます。

- ALW AIX, Linux, and Windows
- IBM i IBM i

  注: IBM i では、デフォルトの mqclient.ini ファイルはありません。 ただし、 IBM i Integrated File System (IFS) でファイルを作成することはできます。

詳細については、159 ページの『クライアント構成ファイルの場所』を参照してください。

注: z/OS z/OS プラットフォームを使用して IBM MQ クライアントを実行することはできません。したがって、 IBM MQ for z/OS には mqclient.ini ファイルは存在しません。

IBM MQ MQI client 構成ファイル内の属性は、以下を使用するクライアントに適用されます。

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

IBM MQ MQI client 構成ファイル内の属性はほとんどの IBM MQ クライアントに適用されますが、管理対象の.NET クライアントまたは XMS .NET クライアントによって、読み取られない属性、IBM MQ classes for Java または IBM MQ classes for JMS を使用するクライアントによって読み取られない属性がいくつかあります。 詳しくは、160 ページの『各属性を読み取ることができる IBM MQ クライアント』を参照してください。

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

以下にクライアント構成ファイルの例を示します。

```
#* Module Name: mqclient.ini                                          *#
#* Type       : IBM MQ MQI client configuration file                  *#
#  Function   : Define the configuration of a client                  *#
#*                                                                    *#
#**********************************************************************#
#* Notes      :                                                       *#
#* 1) This file defines the configuration of a client                 *#
#*                                                                    *#
#**********************************************************************#

ClientExitPath:
   ExitsDefaultPath=/var/mqm/exits
   ExitsDefaultPath64=/var/mqm/exits64

TCP:
```

```
   Library1=DLLName1
   KeepAlive = Yes
   ClntSndBuffSize=0
   ClntRcvBuffSize=0
   Connect_Timeout=0

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

LU62:
   TPName
   Library1=DLLName1
   Library2=DLLName2

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

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

Connection:
     ApplName=ExampleApplName
```

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

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

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

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

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

構成ファイル内のコメント

Linux AIX

セミコロン「;」およびハッシュ文字「#」を使用して、構成ファイル内のコメントの開始を示すことができます。 これにより、行全体をコメントとしてマークしたり、設定の値に含まれることのない行末のコメントを表示したりできます。

値がこれらの文字のいずれかを必要とする場合は、円記号 ('\') を使用してその文字をエスケープする必要があります。

以下の例は、構成ファイルでのコメントの使用方法を示しています。

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

関連概念

IBM MQ classes for Java 構成ファイル

Multi クライアント構成ファイルの場所

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

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

1. 環境変数 **MQCLNTCF** で指定された場所。

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

   注:

   - C、 .NET 、および XMS クライアントは、 `file:` プロトコルのみをサポートします。 URL ストリングが `protocol:` で始まらない場合は、 `file:` プロトコルが想定されます。
   - 環境変数の読み取りをサポートしない Java 1.4.2 JRE を許可するために、 **MQCLNTCF** 環境変数を **MQCLNTCF** Java システム・プロパティーでオーバーライドできます。
2. アプリケーションの現行作業ディレクトリー内の `mqclient.ini` というファイル。
3. AIX, Linux, and Windows システムの IBM MQ データ・ディレクトリー内の `mqclient.ini` というファイル。

   注:

   - IBM MQ データ・ディレクトリーは、IBM i や z/OS などの特定のプラットフォームには存在しません。また、クライアントが別の製品と一緒に提供されている場合もあります。

     IBM i IBM i では、デフォルトの `mqclient.ini` ファイルはありません。 ただし、このファイルは、 IBM i Integrated File System (IFS) のディレクトリー `/QIBM/UserData/mqm/`、およびこのファイルを指すように定義された環境変数 **MQCLNTCF** に作成することができます。 以下に例を示します。

     ```
     ADDENVVAR ENVVAR(MQCLNTCF) VALUE('QIBM/UserData/mqm/mqclient.ini') REPLACE(*YES)
     ```

     その他の環境変数の例については、 64 ページの『環境変数の説明』を参照してください。

     z/OS z/OS プラットフォームを使用して IBM MQ クライアントを実行することはできません。 したがって、 IBM MQ for z/OS には `mqclient.ini` ファイルは存在しません。
   - Linux AIX AIX and Linux システムの場合、このディレクトリーは `/var/mqm` です。
   - Windows Windows プラットフォームでは、データ・ディレクトリーを指すように、インストール時に環境変数 **MQ_DATA_PATH** を構成します。 通常は `C:\ProgramData\ IBM \ MQ.` です。

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

Multi 各属性を読み取ることができる IBM MQ クライアント

IBM MQ MQI client 構成ファイル内のほとんどの属性は、C クライアントおよび非管理対象 .NET クライアントで使用できます。 ただし、一部の属性は、管理対象 .NET および XMS .NET クライアントによって、ま

たは IBM MQ classes for Java や IBM MQ classes for JMS を使用するクライアントによって読み取られません。

表 *15.* 各タイプのクライアントに適用される属性

mqclient.ini スタンザ名および属性	**説明**	**C および非管理対象 .NET**	**Java**	**JMS**	**マネージド.NET**	**マネージド XMS .NET**
CHANNELS スタンザ						
CCSID	使用するコード化文字セット番号。	はい	いいえ	いいえ	はい	はい
ChannelDefinitionDirectory	クライアント・チャネル定義テーブルを含むファイルのディレクトリー・パス。	はい	いいえ	いいえ	はい	はい
ChannelDefinitionFile	クライアント・チャネル定義テーブルを含むファイルの名前。	はい	いいえ	いいえ	はい	はい
ReconDelay	自動再接続が可能なクライアント・プログラムに対して再接続遅延を構成するための管理オプション。	はい	いいえ	はい	はい	はい
DefRecon	クライアント・プログラムが自動的に再接続できるようにしたり、自動的に再接続するように作成されたクライアント・プログラムの自動再接続を無効にしたりするための管理オプション。	はい	いいえ	はい	はい	はい
MQReconnectTimeout	クライアントに再接続するタイムアウト (秒)。	はい	いいえ	いいえ	はい	いいえ

表 15. 各タイプのクライアントに適用される属性 (続き)

mqclient.ini スタンザ名および属性	説明	C および非管理対象 .NET	Java	JMS	マネージド.NET	マネージド XMS .NET
ServerConnectionParms	IBM MQ サーバーの位置および使用する通信方式。	はい	いいえ	いいえ	はい	はい
Put1DefaultAlwaysSync	オプション MQPMO_RESPONSE_AS_Q_DEF を指定して MQPUT1 関数呼び出しの動作を制御します。	はい	はい	はい	はい	はい
PasswordProtection	SSL または TLS を使用する代わりに、保護されたパスワードを MQCSP 構造に設定できます。	はい	はい	はい	はい	はい
ClientExitPath スタンザ						
ExitsDefaultPath	クライアントの 32 ビット・チャネル出口の位置を指定します。	はい	はい	はい	はい	はい
ExitsDefaultPath64	クライアントの 64 ビット・チャネル出口の位置を指定します。	はい	はい	はい	はい	はい
JavaExitsClassPath	Java 出口の実行時に、クラスパスに追加される値。	いいえ	はい	はい	いいえ	いいえ
Connection スタンザ						
ApplName	クライアント構成ファイルで指定されているアプリケーション名。	はい	いいえ	いいえ	いいえ	いいえ
JMQI スタンザ						

表 15. 各タイプのクライアントに適用される属性 (続き)

mqclient.ini スタンザ名および属性	説明	C および非管理対象 .NET	Java	JMS	マネージド.NET	マネージド XMS .NET
useMQCSPauthentication	IBM MQ classes for Java および IBM MQ classes for JMS のアプリケーションがキュー・マネージャーへの認証で互換モードと MQCSP 認証モードのどちらを使用するかを制御します。	いいえ	はい	はい	いいえ	いいえ
MessageBuffer スタンザ						
MaximumSize	1 から 999 999 の範囲の先読みバッファーのサイズ (キロバイト単位)。	はい	はい	はい	はい	はい
PurgeTime	先読みバッファーに残ったメッセージが除去される間隔 (秒単位)。	はい	はい	はい	はい	はい
UpdatePercentage	クライアント・アプリケーションがサーバーに対して新しい要求をいつ行うかを決定するしきい値の計算に使用される、1 から 100 の範囲の更新のパーセント値。	はい	はい	はい	はい	はい
PreConnect スタンザ						
データ	接続定義が保管されるリポジトリーの URL。	はい	いいえ	いいえ	いいえ	いいえ

表 15. 各タイプのクライアントに適用される属性 (続き)						
mqclient.ini スタンザ名および属性	**説明**	**C および非管理対象 .NET**	**Java**	**JMS**	**マネージド.NET**	**マネージド XMS .NET**
機能	PreConnect 出口コードを含むライブラリーへの、関数のエントリー・ポイントの名前。	はい	いいえ	いいえ	いいえ	いいえ
モジュール	API 出口コードを含むモジュールの名前。	はい	いいえ	いいえ	いいえ	いいえ
Sequence	その他の出口に関してこの出口が呼び出される順序。	はい	いいえ	いいえ	いいえ	いいえ
セキュリティー・スタンザ						
DisableClientAMS	キュー・マネージャーへのクライアント接続で AMS を無効/有効にします。	はい	はい	はい	いいえ	いいえ
SSL スタンザ						
V 9.3.0 OutboundSNI	SNI 対応クライアントが、TLS 接続の開始時に、SNI をリモート・システムへのターゲット IBM MQ チャネル名に設定するか、ホスト名に設定するかを指定します。	はい	はい	はい	はい	いいえ

表 15. 各タイプのクライアントに適用される属性 (続き)

mqclient.ini スタンザ名および属性	説明	C および非管理対象 .NET	Java	JMS	マネージド.NET	マネージド XMS .NET
AllowOutbound「SNI」	SNI 対応クライアントが、TLS 接続の開始時に、SNI をリモート・システムへのターゲット IBM MQ チャネル名に設定するかどうかを指定します。 **重要：** Deprecated V 9.3.0 IBM MQ 9.3.0 以降、このプロパティーは非推奨になりました。代わりに **OutboundSNI** を使用してください。	はい	はい	はい	いいえ	いいえ
AllowTLSV13	キュー・マネージャーが TLS 1.3 CipherSpec を使用できるかどうか。	はい (C/C++ クライアント)	いいえ	いいえ	いいえ	いいえ
CDPCheckExtensions	このキュー・マネージャーの SSL チャネルまたは TLS チャネルが、CrlDistributionPoint 証明書拡張で指定されている CDP サーバーの検査を試行するかどうかを指定します。	はい	いいえ	いいえ	いいえ	いいえ
CertificateLabel	チャネル定義の証明書ラベル。	はい	いいえ	いいえ	いいえ	いいえ

表 15. 各タイプのクライアントに適用される属性 (続き)

mqclient.ini スタンザ名および属性	説明	C および非管理対象 .NET	Java	JMS	マネージド.NET	マネージド XMS .NET
CertificateValPolicy	使用する証明書妥当性検査のタイプを決定します。	はい	いいえ	いいえ	いいえ	いいえ
ClientRevocationChecks	クライアント接続呼び出しで SSL/TLS チャネルを使用する場合の証明書の失効検査をどのように構成するかを指定します。	はい	いいえ	いいえ	いいえ	いいえ
EncryptionPolicySuiteB	チャネルで Suite-B 準拠の暗号化を使用するかどうか、および使用する際の強度のレベルを決定します。	はい	いいえ	いいえ	いいえ	いいえ
V 9.3.0 EnvironmentScope	IBM MQ がプロセス全体に対して単一の IBM Global Security Kit (GSKit) 環境を使用するか、接続ごとに GSKit 環境を使用するかを制御します。	はい (C クライアント)	いいえ	いいえ	いいえ	いいえ
MinimumRSAKey サイズ	RSA 証明書を受け入れるために必要な最小の鍵サイズを指定します。	はい (C/C++ クライアント)	いいえ	いいえ	いいえ	いいえ

表 15. 各タイプのクライアントに適用される属性 (続き)						
mqclient.ini スタンザ名および属性	**説明**	**C および非管理対象 .NET**	**Java**	**JMS**	**マネージド.NET**	**マネージド XMS .NET**
OCSPAuthentication	OCSP が有効な状態で、OCSP 取り消し検査で証明書取り消し状況を判別できない場合の、IBM MQ の動作を定義します。	はい	いいえ	いいえ	いいえ	いいえ
OCSPCheckExtensions	IBM MQ が AuthorityInfoAccess 証明書拡張に反応するかどうかを制御します。	はい	いいえ	いいえ	いいえ	いいえ
OCSPTimeout (OCSPTimeout)	失効検査の実行時に OCSP 応答側を待機する秒数。	はい	いいえ	いいえ	いいえ	いいえ
ALW PeerCertChainValidation	GSKit 証明書の妥当性検査設定。	はい	いいえ	いいえ	いいえ	いいえ
SSLCryptoHardware	システム上に存在する PKCS #11 暗号ハードウェアの構成に必要なパラメーター・ストリングを設定します。	はい	いいえ	いいえ	いいえ	いいえ
SSLCryptoHardwareKeyFile	**SSLCryptoHardware** 属性で指定された PKCS #11 暗号ハードウェア構成ストリング内のパスワードを暗号化するために使用された初期鍵を含むファイルの絶対パスと名前を指定します。	はい	いいえ	いいえ	いいえ	いいえ

表 15. 各タイプのクライアントに適用される属性 (続き)

mqclient.ini スタンザ名および属性	説明	C および非管理対象 .NET	Java	JMS	マネージド.NET	マネージド XMS .NET
SSLFipsRequired	IBM MQ で暗号化を実行する場合に FIPS 認証アルゴリズムのみを使用するかどうかを指定します。	はい	いいえ	いいえ	いいえ	いいえ
SSLHTTPProxyName	このストリングは、 GSKit が OCSP 検査に使用する HTTP プロキシー・サーバーのホスト名またはネットワーク・アドレスのいずれかです。	はい	いいえ	いいえ	いいえ	いいえ
SSLHTTPConnectTimeout	失効検査の実行時に HTTP サーバーへのネットワーク接続が正常に確立されるのを待機する秒数。	はい	いいえ	いいえ	いいえ	いいえ
SSLKeyRepository	ユーザーのデジタル証明書を保持するキー・リポジトリーの位置 (語幹形式で指定)。	はい	いいえ	いいえ	いいえ	いいえ
V 9.3.0 V 9.3.0 SSLKeyRepository パスワード	鍵リポジトリーにアクセスするためのパスフレーズ。	はい	いいえ	いいえ	いいえ	いいえ

表 15. 各タイプのクライアントに適用される属性 (続き)

mqclient.ini スタンザ名および属性	説明	C および非管理対象 .NET	Java	JMS	マネージド.NET	マネージド XMS .NET
SSLKeyResetCount	秘密鍵が再ネゴシエーションされる前に、SSL チャネルまたは TLS チャネルで送受信された暗号化されていないバイトの数。	はい	いいえ	いいえ	いいえ	いいえ
TCP スタンザ						
ClntRcvBuffSize	クライアント接続チャネルとサーバー接続チャネルのクライアント側で使用される TCP/IP 受信バッファーのサイズ (バイト)。	はい	はい	はい	はい	はい
ClntSndBuffSize	クライアント接続チャネルとサーバー接続チャネルのクライアント側で使用される TCP/IP 送信バッファーのサイズ (バイト)。	はい	はい	はい	はい	はい
Connect_Timeout	ソケットへの接続試行がタイムアウトになるまでの秒数。	はい	はい	はい	いいえ	いいえ
IPAddressVersion	チャネル接続に使用する IP プロトコルを指定します。	はい	いいえ	いいえ	はい	はい
KeepAlive	KeepAlive 機能をオンまたはオフに切り替えます。	はい	はい	はい	はい	はい

表 15. 各タイプのクライアントに適用される属性 (続き)

mqclient.ini スタンザ名および属性	説明	C および非管理対象 .NET	Java	JMS	マネージド.NET	マネージド XMS .NET
Windows Library1	Windows のみ。TCP/IP ソケット DLL の名前。	はい	いいえ	いいえ	いいえ	いいえ
Trace スタンザ **注 :** Trace スタンザは、 IBM MQ .NET および XMS .NET クライアントにのみ適用されます。						
V 9.3.3 MQDotnetTrace レベル	IBM MQ .NET トレースを使用可能にするために使用されます。	いいえ	いいえ	いいえ	はい	いいえ
V 9.3.3 MQDotnetTrace パス	IBM MQ .NET トレース・ファイルが作成されるフォルダーを指します。	いいえ	いいえ	いいえ	はい	いいえ
V 9.3.3 MQDotnetError パス	IBM MQ .NET トレース用のエラー・ログ・ファイルが作成されるフォルダーを指します。	いいえ	いいえ	いいえ	はい	いいえ
V 9.3.3 XMSDotnetTrace レベル	XMS .NET トレースを使用可能にするために使用されます。	いいえ	いいえ	いいえ	いいえ	はい
V 9.3.3 XMSDotnetTraceFilePath	XMS .NET トレース・ファイルが作成されるフォルダーを指します。	いいえ	いいえ	いいえ	いいえ	はい
V 9.3.3 XMSDotnetTrace 仕様	XMS .NET についてトレースしたいクラスの名前を指定します。	いいえ	いいえ	いいえ	いいえ	はい
V 9.3.3 XMSDotnetTrace 仕様	XMS .NET 用に生成するトレース・ファイルの最大サイズを指定します。	いいえ	いいえ	いいえ	いいえ	はい

表 15. 各タイプのクライアントに適用される属性 (続き)						
mqclient.ini スタンザ名および属性	**説明**	**C および非管理対象 .NET**	**Java**	**JMS**	**マネージド.NET**	**マネージド XMS .NET**
V 9.3.3 XMSDotnetTraceFileSize	XMS .NET 用に保存されるトレース・ファイルの数。	いいえ	いいえ	いいえ	いいえ	はい

V 9.3.0 クライアント構成ファイルの Application スタンザ

Application スタンザを使用して、この構成を使用して接続する特定のアプリケーションの均一クラスター・バランシング動作に影響を与える属性を指定します。 このスタンザ内の値は、ApplicationDefaults スタンザよりも優先されますが、プログラムと提供される MQBNO 構造体によって、オーバーライドすることができます。

注 : このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。 すべての IBM MQ MQI client 構成ファイルのスタンザを示す要約表は、「Which IBM MQ attributes can be read by each client」を参照してください。

アプリケーション・スタンザには、以下の属性を含めることができます:

Name = ApplicationName
オプションが適用されるアプリケーション名を識別します。

タイプ = Simple,ReqRep
このアプリケーションが参加するアクティビティーの一般的な IBM MQ パターンを IBM MQ に示します。

BalanceTimeout = *Never,Immediate,0-999999999*、デフォルト
アプリケーション・アクティビティーが再平衡化を許可するために中断される前のタイムアウトを IBM MQ に指示します。再平衡化を許可しないか、または最大 999,999,999 秒 (デフォルト値は 10 秒) までの値を指定します。

BalanceOptions=*None,IgnTrans*
バランシング・オプションが設定されていないか、トランザクションに現在関与しているアプリケーションの即時中断を許可します。

V 9.3.0 Multi クライアント構成ファイルの ApplicationDefaults スタンザ

ApplicationDefaults スタンザを使用して、この構成を使用して、接続するクライアント・アプリケーションのデフォルトのクラスター・バランシング動作に影響を与える属性を指定します。 これらのデフォルトは、アプリケーション固有の Application スタンザ、またはプログラムによって提供される MQBNO 構造体のいずれかによってオーバーライドできます。

注 : このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。 すべての IBM MQ MQI client 構成ファイルのスタンザを示す要約表は、「Which IBM MQ attributes can be read by each client」を参照してください。

ApplicationDefaults スタンザには、以下の属性を含めることができます:

タイプ = Simple,ReqRep
このアプリケーションが参加するアクティビティーの一般的な IBM MQ パターンを IBM MQ に示します。

BalanceTimeout = *Never,Immediate,0-999999999*、デフォルト
アプリケーション・アクティビティーが再平衡化を許可するために中断される前のタイムアウトを IBM MQ に指示します。再平衡化を許可しないか、または最大 999,999,999 秒 (デフォルト値は 10 秒) までの値を指定します。

BalanceOptions=*None,IgnTrans*
バランシング・オプションが設定されていないか、トランザクションに現在関与しているアプリケーションの即時中断を許可します。

Multi クライアント構成ファイルの CHANNELS スタンザ

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

注 : このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。 すべての IBM MQ MQI client 構成ファイルのスタンザを示す要約表は、「Which IBM MQ attributes can be read by each client」を参照してください。

次の属性を CHANNELS スタンザに含めることができます。

CCSID = *number*
使用するコード化文字セット番号。

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

CCSID 番号は、MQCCSID 環境変数と同等です。

ChannelDefinitionDirectory = *path*
クライアント・チャネル定義テーブルを含むファイルのディレクトリー・パス。

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

Windows Windows システムでは、デフォルトは IBM MQ データ・ファイルおよびログ・ファイル・ディレクトリー(通常は `C:\ProgramData\ IBM \ MQ`)です。

Linux AIX AIX and Linux システムでは、デフォルトは`/var/mqm` です。

ChannelDefinitionDirectory には、ChannelDefinitionFile 属性との組み合わせで機能する URL を含めることができます (52 ページの『CCDT への URL アクセス』 を参照してください)。

ChannelDefinitionDirectory パスは、**MQCHLLIB** 環境変数と同等です。

ChannelDefinitionFile = *filename*|AMQCLCHL.TAB
クライアント・チャネル定義テーブルを含むファイルの名前。

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

クライアント・チャネル定義テーブルは、**MQCHLTAB** 環境変数と同等です。

ReconDelay=(delay[、rand])(delay[、rand])...
ReconDelay 属性は、自動的に再接続できるクライアント・プログラムに対して再接続遅延を構成するための管理オプションを提供します。

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

構成の例は以下のとおりです。

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

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

DefRecon = NO|YES|QMGR |DISABLED
DefRecon 属性は、クライアント・プログラムによる自動再接続を有効にするか、自動的に再接続するように作成されたクライアント・プログラムの自動再接続を無効にするための 管理オプションを提供します。 プログラムが再接続との互換性がない MQPMO_LOGICAL_ORDER のようなオプションを使用する場合は、後者の設定を選択できます。

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

IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

自動クライアント再接続は通常、以下の 2 つの値に依存します。

- アプリケーション MQCONNX (または JMS 接続ファクトリー) で設定された再接続オプション
- 使用中の任意のクライアント接続定義 (例えば、CCDT ファイルを使用して提供される MQCD 構造体) で提供されるデフォルトの再接続オプション。

`mqclient.ini` ファイル属性は、**DefReconnect** 属性を設定するチャネル定義が使用されていない場合に **のみ** 適用されます。この場合、この属性は指定された場合と同様に動作します。 チャネルの **DefReconnect** 属性 (および該当する場合はこの属性):

- いずれかが DISABLED に設定されている場合は、アプリケーション・コードをオーバーライドします。
- オプションが MQCONNX で指定されている場合は、他のすべての場合にアプリケーション・コードによってオーバーライドされます。

アプリケーションとチャネル定義の提供値の考えられるすべての組み合わせを示す表については、DEFRECON の説明を参照してください。

注:

- MQCD が使用されているが、 MQCD_VERSION_10 より前の日付の場合、 **DefReconnnect** パラメーターは構造体の一部ではありません。 この場合、欠落しているパラメーターの値には、 `mqclient.ini` file **DefReconnnect** 値が指定されている場合はその値が取り込まれます。 これは、例えば、旧バージョンの IBM MQ で生成されたバイナリー・フォーマットの CCDT がクライアント・アプリケーションによって引き続き使用されている場合に発生する可能性があります。
- IBM MQ クライアント・コード (44 ページの『JSON 形式の CCDT の構成』を参照) によって解釈される場合、JSON CCDT は常に最新バージョンの MQCD 構造体を生成するため、明示的に別の値を指定しない限り、この属性のデフォルト値 (NO) を常に提供します。

MQReconnectTimeout

クライアント内の自動クライアント再接続機能が接続の再確立を試行する最大時間 (秒数)。 デフォルト値は 1800 秒 (30 分) です。

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

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

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

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

ServerConnectionParms

ServerConnectionParms は *MQSERVER* 環境変数に相当し、「IBM MQ サーバーの場所と使用する通信方法を指定する。

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

ServerConnectionParms 属性で定義するのは単純なチャネルのみです。この属性を使用して、TLS チャネルやチャネル出口があるチャネルを定義することはできません。 これは *ChannelName*/*TransportType*/*ConnectionName* という形式のストリングで、*ConnectionName* は完全修飾ネットワー

ク名であることが必要です。 *ChannelName* に、スラッシュ文字 (/) を使用することはできません。スラッシュ文字はチャネル名、トランスポート・タイプ、および接続名を区切るために使用されるからです。

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

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

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

この場合、2 番目だけが使用されます。

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

Put1DefaultAlwaysSync = NO (デフォルト) | YES
オプション MQPMO_RESPONSE_AS_Q_DEF を指定して MQPUT1 関数呼び出しの動作を制御します。

この属性は、C、管理されていない.NET、IBM MQ classes for Java、IBM MQ classes for JMS、管理対象.NET、および管理対象 XMS .NET クライアントから読み取ることができる。

NO
MQPUT1 が MQPMO_SYNCPOINT で設定される場合、MQPMO_ASYNC_RESPONSE として動作します。同様に、MQPUT1 が MQPMO_NO_SYNCPOINT で設定される場合、MQPMO_SYNC_RESPONSE として動作します。 これがデフォルト値です。

YES
MQPUT1 は、MQPMO_SYNCPOINT または MQPMO_NO_SYNCPOINT が設定されているかどうかに関係なく MQPMO_SYNC_RESPONSE が設定されている場合のように動作します。

PasswordProtection = Compatible (デフォルト) |always|オプション

IBM MQ 8.0 以降、 IBM MQ client アプリケーションがキュー・マネージャーに接続するときに指定する認証資格情報は、接続で TLS 暗号化を使用しない場合、 IBM MQ MQCSP パスワード保護機能を使用して保護することができます。

この属性は、C、管理されていない.NET、IBM MQ classes for Java、IBM MQ classes for JMS、管理対象.NET、および管理対象 XMS .NET クライアントから読み取ることができる。

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

MQCSP 構造で資格情報を保護する方法、およびこの属性に設定できる値について詳しくは、 MQCSP パスワード保護を参照してください。

関連タスク
キュー・マネージャーへの IBM MQ MQI アプリケーションの接続

Multi クライアント構成ファイルの ClientExitPath スタンザ

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

注 : このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。 すべての IBM MQ MQI client 構成ファイルのスタンザを示す要約表は、「Which IBM MQ attributes can be read by each client」を参照してください。

次の属性を ClientExitPath スタンザに含めることができます。

ExitsDefaultPath = *string*
クライアント用の 32 ビット・チャネル出口の場所を指定します。

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

ExitsDefaultPath64 = *string*
クライアントに対する 64 ビット・チャネル出口の場所を指定します。

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

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

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

JMS 構成ファイルでは、 JavaExitsClass パス名に標準 com.ibm.mq.cfg が与えられます。 接頭部とこのフルネームは、 IBM MQ システム・プロパティーでも使用されます。

Multi クライアント構成ファイルの Connection スタンザ

Connection スタンザを使用してアプリケーション名を指定します。

注 : このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。 すべての IBM MQ MQI client 構成ファイルのスタンザを示す要約表は、「Which IBM MQ attributes can be read by each client」を参照してください。

Connection スタンザに以下の属性を組み込めます。

ApplName = ExampleApplname
クライアント構成ファイルでアプリケーション名を指定できます。

この属性は、C および非管理対象 .NET クライアントが使用することができます。

Multi クライアント構成ファイルの JMQI スタンザ

JMQI スタンザを使用して、 IBM MQ classes for Java および IBM MQ classes for JMS によって使用される Java Message Queuing Interface (JMQI) の構成パラメーターを指定します。

注 : このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。 すべての IBM MQ MQI client 構成ファイルのスタンザを示す要約表は、「Which IBM MQ attributes can be read by each client」を参照してください。

次の属性を JMQI スタンザに含めることができます。

useMQCSPauthentication = NO | YES
IBM MQ classes for Java および IBM MQ classes for JMS のアプリケーションがキュー・マネージャーへの認証で互換モードと MQCSP 認証モードのどちらを使用するかを制御します。

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

この属性には、以下の値を設定できます。

NO
キュー・マネージャーへの認証で互換モードを使用します。 これは IBM MQ 9.3.0 より前のバージョンでのデフォルト値です。

YES

キュー・マネージャーへの認証で MQCSP 認証モードを使用します。 V 9.3.0 これは IBM MQ 9.3.0 以降のデフォルト値です。

useMQCSPauthentication 属性の値よりも優先される認証モードを設定する方法は、他にもいくつかあります。 互換モードと MQCSP 認証モードの詳細については、Java クライアントを使用した接続認証を参照してください。

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

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

LU62 スタンザ

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

Library1 = *DLLName*|WCPIC32
APPC DLL の名前。

Library2 = *DLLName*|WCPIC32
Library1 と同様に、コードが 2 つの別個のライブラリーに格納されている場合に使用されます。

TPName
リモート・サイトで始動する TP 名。

NETBIOS スタンザ

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

AdapterNum = *number*|0
LAN アダプターの番号。

Library1 = *DLLName*|NETAPI32
NetBIOS DLL の名前。

LocalName = *name*
このコンピューターを LAN 上で認識する際に使用する名前。

これは MQNAME 環境変数と同等です。

NumCmds = *number*|1
割り振るコマンドの数。

NumSess = *number*|1
割り振るセッションの数。

SPX スタンザ

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

BoardNum = *number*|0
LAN アダプターの番号。

KeepAlive = YES|NO
KeepAlive 機能のオン/オフを切り替えます。

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

Library1 = *DLLName*|WSOCK32.DLL
SPX DLL の名前。

Library2 = *DLLName*|WSOCK32.DLL
Library1 と同様に、コードが 2 つの別個のライブラリーに格納されている場合に使用されます。

Socket = *number*|5E86
16 進表記の SPX ソケット番号。

Multi クライアント構成ファイルの MessageBuffer スタンザ

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

注 : このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。 すべての IBM MQ MQI client 構成ファイルのスタンザを示す要約表は、「Which IBM MQ attributes can be read by each client」を参照してください。

次の属性を MessageBuffer スタンザに含めることができます。

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

この属性は、C、管理されていない.NET、IBM MQ classes for Java、IBM MQ classes for JMS、管理対象.NET、および管理対象 XMS .NET クライアントから読み取ることができる。

以下の特殊値があります。

-1
クライアントが適切な値を決定します。

0
クライアントでは先読みは無効です。

PurgeTime = *integer*|600
先読みバッファーに残ったメッセージが除去される間隔 (秒単位)。

この属性は、C、管理されていない.NET、IBM MQ classes for Java、IBM MQ classes for JMS、管理対象.NET、および管理対象 XMS .NET クライアントから読み取ることができる。

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

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

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

この属性は、C、管理されていない.NET、IBM MQ classes for Java、IBM MQ classes for JMS、管理対象.NET、および管理対象 XMS .NET クライアントから読み取ることができる。

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

しきい値 T は次のように計算されます。

```
T = Upper - Lower
```

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

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

```
Lower = Upper x UpdatePercentage / 100
```

例 1:

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

クライアントは、MQGET を呼び出してメッセージを取得して、これを繰り返します。 これは、MQGET が n バイトをコンシュームするまで続きます。

次の計算を使用します。

```
T = Upper - Lower
```

T は (100 - 20) = 80 キロバイトとなります。

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

例 2:

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

クライアントは、MQGET を呼び出してメッセージを取得して、これを繰り返します。 これは、MQGET が n バイトをコンシュームするまで続きます。

次の計算を使用します。

```
T = Upper - Lower
```

T は (100 - 40) = 60 キロバイトとなります。

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

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

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

Multi クライアント構成ファイルの PreConnect スタンザ

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

注 : このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。 すべての IBM MQ MQI client 構成ファイルのスタンザを示す要約表は、「Which IBM MQ attributes can be read by each client」を参照してください。

次の属性を PreConnect スタンザに含めることができます。

Data = *user_data*

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

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

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

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

Function = *myFunc*

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

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

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

このフィールドの最大長は MQ_EXIT_NAME_LENGTH です。

Module = *myMod*
API 出口コードを含むモジュールの名前。

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

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

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

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

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

関連タスク
リポジトリーから接続前出口を使用した接続定義の参照

Multi クライアント構成ファイルの Security スタンザ

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

注 : このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。 すべての IBM MQ MQI client 構成ファイルのスタンザを示す要約表は、「Which IBM MQ attributes can be read by each client」を参照してください。

セキュリティー・スタンザには、以下の属性を含めることができます:

DisableClientAMS = NO|YES
DisableClientAMS 属性を使用すると、IBM MQ クライアントを使用して旧バージョンの製品のキュー・マネージャーに接続していて、 2085 (MQRC_UNKNOWN_OBJECT_NAME) エラーが報告された場合に、IBM MQ Advanced Message Security (AMS) を無効にすることができます。

IBM MQ Advanced Message Security (AMS) は IBM MQ クライアントで自動的に使用可能になるため、デフォルトでは、クライアントはキュー・マネージャーでオブジェクトのセキュリティー・ポリシーを検査しようとします。

DisableClientAMS 属性の使用例を以下に示します。

- AMS を無効にするには、次のようにします。

```
Security:
DisableClientAMS=Yes
```

- を使用可能にする場合 AMS：

```
Security:
DisableClientAMS=No
```

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

V 9.3.0 V 9.3.0 **MQIInitialKey ファイル = パス名**
クライアントによって提供された資格情報を暗号化するために使用された初期鍵を含むファイルの絶対パスと名前。 鍵リポジトリー・パスフレーズが **runmqicred** ユーティリティーを使用して暗号化されたときに初期鍵ファイルが指定された場合は、初期鍵を指定する必要があります。

この属性は、C および非管理の.NET クライアントによって、読み取ることができます。

関連タスク
クライアントでの Advanced Message Security の無効化

Multi クライアント構成ファイルの SSL スタンザ

SSL スタンザを使用して、TLS の使用に関する情報を指定します。

注 : このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。 すべての IBM MQ MQI client 構成ファイルのスタンザを示す要約表は、「Which IBM MQ attributes can be read by each client」を参照してください。

次の属性を SSL スタンザに含めることができます。

V 9.3.0 **OutboundSNI = CHANNEL | HOSTNAME**
OutboundSNI がチャネルに設定されている場合、SNI 対応クライアントは、TLS 接続の開始時に SNI をリモート・システムのターゲット IBM MQ チャネル名に設定します。

この属性が HOSTNAME に設定されている場合、SNI 対応クライアントは、アウトバウンド接続要求によって TLS ハンドシェーク時にリモート・キュー・マネージャーのデフォルト証明書を受信するように SNI ヘッダーをホスト名に設定します。 したがって、チャネルごとの証明書は使用できません。

この属性は C、管理されていない.NET、IBM MQ classes for Java、および IBM MQ classes for JMS クライアントによって、読み取ることができます。

プロパティー値は、Java/JMS クライアントによって、大/小文字が区別されるため、YES/NO 値は大文字で設定する必要があります。

IBM MQ 9.3.0 以降、 IBM MQ 管理対象 .NET クライアントが更新され、 **OutboundSNI** プロパティーが HOSTNAME に設定されている場合に SERVERNAME がそれぞれのホスト名に設定されるようになりました。これにより、 IBM MQ 管理対象 .NET クライアントは Red Hat OpenShift 経路を使用してキュー・マネージャーに接続できます。

注 : OutboundSNI が HOSTNAME に設定されているアプリケーションが、証明書ラベルが構成されているチャネルに接続すると、アプリケーションは MQRC_SSL_INITIALIZATION_ERROR で拒否され、キュー・マネージャーのエラー・ログに AMQ9673 メッセージが出力されます。

AllowOutboundSNI = YES (デフォルト) |NO
使用可能にすると、SNI 対応クライアントは、TLS 接続の開始時に、SNI をリモート・システムへのターゲット IBM MQ チャネル名に設定します。 この属性が NO に設定されている場合、SNI 対応クライアントは、アウトバウンド接続要求によって TLS ハンドシェーク時にリモート・キュー・マネージャーのデフォルト証明書を受信するように SNI ヘッダーを設定しません。したがって、チャネルごとの証明書は使用できません。

この属性は C、管理されていない.NET、IBM MQ classes for Java、および IBM MQ classes for JMS クライアントによって、読み取ることができます。

プロパティー値は、Java/JMS クライアントによって、大/小文字が区別されるため、YES/NO 値は大文字で設定する必要があります。

重要 : Deprecated V 9.3.0 IBM MQ 9.3.0 以降、**AllowOutboundSNI** プロパティーは非推奨になり、後方互換性の目的でのみ使用できます。

はいに設定された **AllowOutboundSNI** は、チャネルに設定された **OutboundSNI** と同じ機能を提供します。一方、いいえに設定された **AllowOutboundSNI** は、ホスト名に設定された **OutboundSNI** と同じ機能を提供します。

AllowOutboundSNI 属性と **OutboundSNI** 属性の両方が SSL スタンザに存在する場合は、**OutboundSNI** の設定が優先されます。

IBM i ALW **AllowTLSV13 =Y | YES | T | TRUE (デフォルト) | N | NO | F | FALSE**
キュー・マネージャーが TLS 1.3 CipherSpec を使用できるかどうかを指定します (CipherSpec の有効化を参照)。

この属性は、C/C++ クライアントが読み取ることができます。

この属性に指定できる値は以下のとおりです。

- Y (デフォルト)、 YES (デフォルト)、 T (デフォルト)、または TRUE (デフォルト): TLS 1.3 を有効にして、キュー・マネージャーが TLS 1.3 CipherSpecs を使用できるようにします。
- N、NO、F、または FALSE: TLS 1.3 を無効にします。 これは、キュー・マネージャーが TLS 1.3 CipherSpec を使用できないことを意味します。

注 : MQI クライアントを使用している場合、**AllowTLSV13** の値は、アプリケーションによって使用されている 180 ページの『クライアント構成ファイルの SSL スタンザ』 ファイルの SSL スタンザで明示的に指定されていない限り、推測されます。 詳しくは、IBM MQ MQI クライアントと TLS 1.3 を参照してください。

CDPCheckExtensions = YES|NO (デフォルト)

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

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

この属性に指定できる値は以下のとおりです。

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

CertificateLabel = *string*

チャネル定義の証明書ラベル。

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

詳しくは、証明書ラベル (CERTLABL) を参照してください。

CertificateValPolicy = *string*

使用する証明書妥当性検査のタイプを決定します。

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

この属性に指定できる値は以下のとおりです。

ANY
基礎となるセキュア・ソケット・ライブラリーによってサポートされているいずれかの証明書妥当性検査ポリシーを使用します。 この設定はデフォルト設定です。

RFC5280
RFC 5280 標準に準拠する証明書妥当性検査のみを使用します。

ClientRevocationChecks = REQUIRED|OPTIONAL|DISABLED

クライアント接続呼び出しで TLS チャネルを使用する場合の証明書の失効検査をどのように構成するかを指定します。 **OCSPAuthentication** も参照してください。

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

この属性に指定できる値は以下のとおりです。

REQUIRED (デフォルト)
CCDT から証明書失効構成をロードし、構成に従って失効検査を実行することを試みます。 CCDT ファイルを開くことができない場合、または証明書を検証できない場合 (例えば、OCSP または CRL サーバーを使用できないため) には、MQCONN 呼び出しは失敗します。 CCDT に失効構成が含まれない場合には、失効検査が実行されませんが、これによりチャネルが失敗することはありません。

Windows Windows システムでは、CRL 失効検査に Active Directory を使用することもできます。 OCSP 失効検査に Active Directory を使用することはできません。

MQSCO または CCDT のいずれかを使用している場合、接続は成功します。 CCDT ファイルがなく、MQSCO も指定されていない場合、接続は理由コード 2059 で失敗し、エラー・ログに AMQ9518E: File '/var/mqm/AMQCLCHL.TAB' not found。

オプション
REQUIRED と同様ですが、証明書失効構成をロードできない場合にはチャネルは失敗しません。

DISABLED
CCDT から証明書失効構成をロードしようとせず、証明書の失効検査を実行しません。

注 : MQCONN 呼び出しではなく、MQCONNX 呼び出しを使用する場合は、MQSCO により認証情報レコード (MQAIR) を設定することもできます。 そのため、CCDT ファイルを開くことができない場合の MQCONNX のデフォルト動作は、失敗することではなく、MQAIR が設定されていると見なすことです (設定しないことにした場合も含む)。

EncryptionPolicySuiteB = *string*
チャネルで Suite-B 準拠の暗号化を使用するかどうか、および使用する際の強度のレベルを決定します。

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

この属性に指定できる値は以下のとおりです。

NONE
Suite-B 準拠の暗号化を使用しない。 この設定はデフォルト設定です。

128_BIT,192_BIT
セキュリティー強度を 128 ビットと 192 ビットの両方のレベルに設定する。

128_BIT
セキュリティー強度を 128 ビット・レベルに設定する。

192_BIT
セキュリティー強度を 192 ビット・レベルに設定する。

V 9.3.0 ALW **EnvironmentScope=PROCESS|CONNECTION**
IBM MQ がプロセス全体に対して単一の IBM Global Security Kit (GSKit) 環境を使用するか、接続ごとに GSKit 環境を使用するかを制御します。

この属性は C クライアントによって、読み取ることができます。

この属性に指定できる値は以下のとおりです。

PROCESS
プロセスによって作成される複数の接続には、単一の GSKit 環境が使用されます。 この設定を使用すると、プロセス内のすべてのアクティブな TLS 接続が停止するまで、TLS 鍵ストアの変更は使用可能になりません。

この値がデフォルト値です。

接続
GSKit 環境は、同じプロセス内の接続ごとに作成されます。 これを有効にすると、プロセスによって開始された新しい TLS 接続によって、TLS 鍵ストアの変更が即時に取得されます。

警告 : この操作モードを有効にすると、アプリケーションは追加の CPU およびメモリー・リソースを使用して各 GSKit 環境を作成します。 このリソース使用量は、それぞれの追加の同時 TLS 接続が増加する。

ALW **MinimumRSAKeySize=int**
RSA 証明書を受け入れるために必要な最小の鍵サイズを指定します。 0 以上の任意の値が許可されます。 指定しない場合は、デフォルトの 1 になります。

この属性は、C/C++ クライアントが読み取ることができます。

OCSPAuthentication = OPTIONAL|REQUIRED|WARN
OCSP が有効な状態で、OCSP 取り消し検査で証明書取り消し状況を判別できない場合の、IBM MQ の動作を定義します。 **ClientRevocationChecks** も参照してください。

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

OCSP レスポンダーが定義されていない場合、OCSP 失効チェックは実行されず、接続の継続が許可される。

この属性に指定できる値は以下のとおりです。

オプション
OCSP 検査で取り消し状況を判別できない証明書が受け入れられ、警告メッセージやエラー・メッセージは生成されません。 取り消し検査が行われなかった場合と同じように、SSL または TLS 接続が続行されます。

REQUIRED
OCSP 検査で、検査されたすべての SSL または TLS 証明書に対して最終的な取り消し結果を示す必要があります。 取り消し状況が確認できない SSL または TLS 証明書はすべて拒否され、エラー・メッセージが表示されます。 キュー・マネージャーの SSL イベント・メッセージが有効な場合は、ReasonQualifier が MQRQ_SSL_HANDSHAKE_ERROR である MQRC_CHANNEL_SSL_ERROR メッセージが生成されます。 接続はクローズされます。

この値がデフォルト値です。

WARN
OCSP 取り消し検査で任意の SSL または TLS 証明書の取り消し状況を判別できない場合に、キュー・マネージャーのエラー・ログに警告が報告されます。 キュー・マネージャーの SSL イベント・メッセージが有効な場合は、ReasonQualifier が MQRQ_SSL_UNKNOWN_REVOCATION である MQRC_CHANNEL_SSL_WARNING メッセージが生成されます。 接続は続行可能です。

OCSPCheckExtensions=YES|NO
IBM MQ が AuthorityInfoAccess 証明書拡張に反応するかどうかを制御します。

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

値を NO に設定すると、IBM MQ は AuthorityInfoAccess 証明書拡張を無視し、OCSP セキュリティー検査を試行しません。 デフォルト値は YES です。

ALW **OCSPTimeout = *number***
失効検査の実行時に OCSP 応答側を待機する秒数。

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

IBM MQ 9.3.0 以降では、値 0 が設定されている場合、デフォルトのタイムアウトである 30 秒が使用されます。

値を設定しない場合は、IBM MQ のデフォルトの 30 秒が使用されます。

ALW **PeerCertChainValidation=*string***
この属性は、C および非管理の.NET クライアントによって、読み取ることができます。

このストリングは、次のいずれかの値に設定できます。

- `Usepeerchain` **[デフォルト]**: ピアによって提供される証明書チェーンを使用して、証明書の検証時にトラスト・チェーンのギャップを埋めることができます。 ルート証明書は例外です。
- `Truststoreonly` **[推奨されません]**: トラストストア内の証明書のみがピアの証明書の検証に使用されます。

SSLCryptoHardware = *string*
システム上に存在する PKCS #11 暗号ハードウェアの構成に必要なパラメーター・ストリングを設定します。

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

`GSK_PKCS11 = driver path and filename;token label;token password;symmetric cipher setting;` の形式でストリングを指定します。

例 : `GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;passw0rd;SYMMETRIC_CIPHER_ON`

ドライバーのパスは、PKCS #11 カードをサポートする共有ライブラリーの絶対パスです。 ドライバーのファイル名は共有ライブラリーの名前です。 PKCS #11 ドライバー・パスおよびファイル名に必要な値の例は、/usr/lib/pkcs11/PKCS11_API.so です。 GSKit を介して対称暗号操作にアクセスするには、対称暗号設定パラメーターを指定します。 このパラメーターの値は次のいずれかです。

SYMMETRIC_CIPHER_OFF
対称暗号操作を使用しません。 この設定はデフォルト設定です。

SYMMETRIC_CIPHER_ON
対称暗号操作を使用します。

Linux AIX ストリングのさまざまな構成要素を指定する場合、円記号 (¥) 文字を使用してセミコロン文字をエスケープする必要があります。セミコロン文字はコメントとして扱われるためです。 例: '\;'

V 9.3.0 **SSLCryptoHardware** 属性のストリングに含まれているトークン・パスワードを保護する必要があります。 詳しくは、暗号ハードウェアを使用する IBM MQ クライアント を参照してください。

V 9.3.0 暗号化されたパスワードを処理する場合、ストリングの長さに制限がなくなりました。

デフォルト値はブランクです。 形式が正しくないストリングを指定すると、エラーが発生します。

SSLCryptoHardwareKeyFile = パス名
SSLCryptoHardware 属性で指定された PKCS #11 暗号化ハードウェア構成ストリング内のパスワードの暗号化に使用された初期鍵を含むファイルの絶対パスと名前。 暗号ハードウェア構成ストリング内のパスワードが **runp11cred** コマンドを使用して暗号化されたときに初期鍵ファイルが指定された場合は、初期鍵を指定する必要があります。 詳しくは、暗号化ハードウェアを使用する IBM MQ クライアントを参照してください。

この属性は、C および非管理の.NET クライアントによって、読み取ることができます。

SSLFipsRequired = YES|NO
IBM MQ で暗号化を実行する場合に FIPS 認証アルゴリズムのみを使用するかどうかを指定します。

この属性は、C および非管理の.NET クライアントによって、読み取ることができます。

暗号ハードウェアが構成されている場合、ハードウェア製品で提供される暗号モジュールが使用されます。 それらは、使用されているハードウェア製品によって一定レベルまで FIPS の認定を受けている場合もあれば、そうではない場合もあります。

SSLHTTPProxyName = *string*
この文字列は、GSKit が OCSP チェックに使用する HTTP プロキシサーバーのホスト名またはネットワークアドレスです。 このアドレスの後に、オプションのポート番号が括弧で囲まれて続く場合もあります。 ポート番号を指定しないと、デフォルトの HTTP ポートである 80 が使用されます。

この属性は、C および非管理の.NET クライアントによって、読み取ることができます。

AIX 32 ビットクライアントの場合 AIX、ネットワークアドレスは IPv4 アドレスのみである可能性があります。

その他のプラットフォームでは、ネットワーク・アドレスには IPv4 または IPv6 アドレスを使用できます。

例えば、ファイアウォールが OCSP 応答側の URL へのアクセスを回避する場合など、この属性が必要になる場合があります。

ALW **SSLHTTPConnectTimeout = *number*|0**
失効検査の実行時に HTTP サーバーへのネットワーク接続が正常に確立されるのを待機する秒数。

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

値を設定しない場合は、IBM MQ のデフォルトの 0 (オフ) が使用されます。

SSLKeyRepository = *pathname*

V 9.3.0 V 9.3.0 ユーザーのデジタル証明書を保持するキー・リポジトリーの絶対パスおよびファイル名。 ファイル拡張子が指定されていない場合、 .kdb であると想定されます。

この属性は、C および非管理の.NET クライアントによって、読み取ることができます。

V 9.3.0 V 9.3.0 **SSLKeyRepositoryPassword = パスフレーズ**

鍵リポジトリーにアクセスするためのパスフレーズ。 値は、プレーン・テキスト・ストリング、または **runmqicred** ユーティリティーを使用して暗号化されたパスフレーズにすることができます。

この属性は、C および非管理の.NET クライアントによって、読み取ることができます。

SSLKeyResetCount = *integer*|0

秘密鍵が再ネゴシエーションされる前に、TLS チャネルまたは TLS チャネルで送受信された暗号化されていないバイトの数。

この属性は、C および非管理の.NET クライアントによって、読み取ることができます。

値は 0 から 999999999 の範囲でなければなりません。

デフォルトは 0 で、これは秘密鍵が絶対に再ネゴシエーションされないことを意味します。

1 から 32768 の値を指定すると、TLS チャネルと TLS チャネルは、秘密鍵のリセット・カウントとして 32768 (32 キロバイト) を使用します。 これは、秘密鍵のリセット値が小さい場合に生じる可能性のある、鍵の過度のリセットを防ぐためです。

Multi クライアント構成ファイルの TCP スタンザ

TCP スタンザを使用して、TCP ネットワーク・プロトコル構成パラメーターを指定します。

注 : このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。 すべての IBM MQ MQI client 構成ファイルのスタンザを示す要約表は、「Which IBM MQ attributes can be read by each client」を参照してください。

次の属性を TCP スタンザに含めることができます。

ClntRcvBuffSize = *number*|0

クライアント接続チャネルとサーバー接続チャネルのクライアント側で使用される TCP/IP 受信バッファーのサイズ (バイト)。

この属性は、C、アンマネージド .NET、 IBM MQ classes for Java、 IBM MQ classes for JMS、 IBM MQ classes for Jakarta Messaging、マネージド .NET、およびマネージド XMS .NET クライアントが読み取ることができる。

値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。 値 0 は、IBM MQ によってバッファー・サイズが修正されるのではなく、オペレーティング・システムがバッファー・サイズを管理することを意味します。

V 9.3.0.30 IBM MQ 9.3.0 Fix Pack 30 から、 IBM MQ Java クライアントで値がゼロに設定されている場合、 Java メッセージキューイングインターフェース（JMQI）は、ネットワークレイヤーが提供するバッファサイズを使用する。 これは、このプロパティをゼロに設定すると、キューマネージャとの通信時に JMQI が誤ってサイズ 32K の TCP/IP 受信バッファを使用してしまうという以前の問題を解決するものです。 受信バッファーのサイズは JMSCS0062IBM MQ classes for Java IBM MQ classes for JMS IBM MQ classes for Jakarta Messaging メッセージに書き込まれる。

ClntSndBuffSize = *number*|0

クライアント接続チャネルとサーバー接続チャネルのクライアント側で使用される TCP/IP 送信バッファーのサイズ (バイト)。

この属性は、C、アンマネージド .NET、 IBM MQ classes for Java、 IBM MQ classes for JMS、 IBM MQ classes for Jakarta Messaging、マネージド .NET、およびマネージド XMS .NET クライアントが読み取ることができる。

値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。 値 0 は、IBM MQ によってバッファー・サイズが修正されるのではなく、オペレーティング・システムがバッファー・サイズを管理することを意味します。

V 9.3.0.30 IBM MQ 9.3.0 Fix Pack 30 から、 IBM MQ Java クライアントで値がゼロに設定されている場合、 Java メッセージキューイングインターフェース（JMQI）は、ネットワークレイヤーが提供するバッファサイズを使用する。 これは、このプロパティをゼロに設定すると、キューマネージャとの通信時に JMQI が誤ってサイズ 32K の TCP/IP 送信バッファを使用してしまうという以前の問題を解決するものです。 送信バッファ・サイズは JMSCS0061IBM MQ classes for Java IBM MQ classes for JMS IBM MQ classes for Jakarta Messaging メッセージに書き込まれる。

Connect_Timeout = *number*
ソケットへの接続試行がタイムアウトになるまでの秒数。

ConnectTimeout = 0 で、非同期 connect () 呼び出しの前に SOCK_NONBLOCK が発行された場合、呼び出しは非ブロック化されます。 デフォルトのタイムアウト値である 20 秒 (CONNECT_WAIT_MAX) は、接続状況の検査に適用されます。

この属性は C、管理されていない.NET、IBM MQ classes for Java、および IBM MQ classes for JMS クライアントによって、読み取ることができます。

IBM MQ チャネル・プロセスは非ブロッキング・ソケットを介して接続します。 このため、ソケットのもう一方の側の準備ができていない場合、connect() は即時に *EINPROGRESS* または *EWOULDBLOCK* を返します。 その後、再接続は試行されません。

Connect_Timeout がゼロ以外の値に設定されている場合、IBM MQ は、ソケットの準備ができるまで、select() 呼び出しに対して規定された期間待機します。 これにより、後続の connect() 呼び出しが成功する確率は上昇します。 このオプションは、ネットワークの負荷が高くなっていて、接続に多少の待機時間が必要なときに役に立つ場合があります。

Connect_Timeout、ClntSndBuffSize、および ClntRcvBuffSize パラメーターの間に関係はありません。

IPAddressVersion = MQIPADDR_IPV4|MQIPADDR_IPV6
チャネル接続に使用する IP プロトコルを指定します。

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

指定可能なストリング値は MQIPADDR_IPV4 または MQIPADDR_IPV6 です。 これらの値は、 **ALTER QMGR IPADDRV** の IPV4 および IPV6 および **MQIPADDRV** 環境変数と同じ意味を持ちます。

KeepAlive = YES|NO
KeepAlive 機能のオン/オフを切り替えます。 KeepAlive=YES を指定すると、TCP/IP は、接続の相手側がまだ使用可能かどうかを定期的に検査します。 有効でない場合、チャネルは閉じた状態です。

この属性は、C、管理されていない.NET、IBM MQ classes for Java、IBM MQ classes for JMS、管理対象.NET、および管理対象 XMS .NET クライアントから読み取ることができる。

Windows **Library1 = *DLLName*|WSOCK32**
(Windows のみ) TCP/IP ソケット DLL の名前。

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

Windows Linux V 9.3.3 クライアント構成ファイルのトレース・スタンザ

Trace スタンザを使用して、 IBM MQ .NET および XMS .NET クライアント・ライブラリーのトレースを有効にします。

TRACE スタンザには、以下の属性を含めることができます。

MQDotnetTraceLevel=0 (デフォルト) |1|2
MQDotnetTraceLevel 属性は、 IBM MQ .NET トレースを開始または停止するために使用されます。

- 0: トレースを停止します - これはデフォルト値です。
- 1: より少ない詳細でトレースを開始します。
- 2: 詳細トレースを開始します - 推奨。

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

MQDotnetTrace パス =パス名
MQDotnetTracePath 属性は、 IBM MQ .NET トレース・ファイルが作成されるフォルダーを指します。 パスがブランクの場合、または **MQDotnetTracePath** プロパティーが定義されていない場合は、アプリケーションの現行ディレクトリーが使用されます。

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

MQDotnetError パス =パス名
MQDotnetErrorPath 属性は、 IBM MQ .NET トレース用のエラー・ログ・ファイルが作成されるフォルダーを指します。 パスがブランクの場合、または **MQDonetErrorPath** 属性が定義されていない場合は、アプリケーションの現行ディレクトリーが使用されます。

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

XMSDotnetTraceLevel=0 (デフォルト) |1|2
XMSDotnetTraceLevel 属性は、 XMS .NET トレースを開始または停止するために使用されます。

- 0: トレースを停止します - これはデフォルト値です。
- 1: 基本フォーマットでトレースを開始します。
- 2: 拡張フォーマットでトレースを開始します。

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

XMSDotnetTraceFilePath=*filename*
XMSDotnetTraceFilePath 属性に値が設定されていない場合、またはこの属性が存在するが空ストリングが含まれている場合は、 XMS .NET のトレース・ファイルが現行ディレクトリーに配置されます。 指定されたディレクトリーにトレース・ファイルを保管するには、 **XMSDotnettraceFilePath** にディレクトリー名を指定します (例: XMSDotnetTraceFilePath="c:*somepath*")。

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

XMSDotnetTrace 仕様 =*ComponentName*=*type*=*state*
XMSDotnetTraceSpecification 属性は、トレースするクラスの名前と、 XMS .NET に必要なトレースのタイプを指定します。

- *ComponentName* は、トレースするクラスの名前です。 この名前には、ワイルドカード文字 * を使用できます。 例えば *=all=enabled はすべてのクラスをトレースすることを指定し、 IBM.XMS.impl.*=all=enabled は API トレースのみが必要であることを指定します。
- *type* には、all、debug、event、 EntryExit のいずれかのトレース・タイプを指定できます。
- *state* は、「enabled」または「disabled」のいずれかに指定できます。

複数のトレース・エレメントを続けて入力するには、「:」(コロン) 区切り文字を使用します。

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

XMSDotnetTraceFileSize=*size*
XMSDotnetTraceFileSize 属性は、 XMS .NET 用に生成されるトレース・ファイルの最大サイズを指定します。 デフォルトの最大値は 20 MB です。これは、XMSDotnetTraceFileSize=20 として指定されます。

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

XMSDotnetTraceFileNumber=*number*
XMSDotnetTraceFileNumber 属性は、 XMS .NET 用に保存するトレース・ファイルの数を指定します。 デフォルトは 4 (1 つのアクティブ・ファイルと 3 つのアーカイブ・ファイル) です。 許可される最小数は 2 です。

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

関連タスク

mqclient.ini を使用した IBM MQ .NET アプリケーションのトレース

mqclient.ini を使用した XMS .NET アプリケーションのトレース

Multi アクティビティー・トレース構成ファイル mqat.ini

アクティビティー・トレースの動作を構成するには、アクティビティー・トレース構成ファイル mqat.ini を使用します。 このファイルは、アクティビティー・トレース・データの報告のレベルと頻度の定義に使用します。 また、このファイルでは、アプリケーションの名前に基づいてアクティビティー・トレースを有効化および無効化するルールを定義できます。

mqat.ini ファイルは、mqs.ini ファイルおよび qm.ini ファイルと同じスタンザ・キーとパラメーターと値のペアのフォーマットに従います。 このファイルは、単一のスタンザ AllActivityTrace で構成されます。このスタンザは、すべてのアクティビティー・トレースについてデフォルトでアクティビティー・トレース・データを報告するレベルと頻度を構成するために使用されます。 このファイルには、複数の ApplicationTrace スタンザを含めることもできます。 これらのスタンザのそれぞれが、接続のアプリケーション名とルールのマッチングに基づいて、1 つ以上の接続のトレース動作のルールを定義します。 詳しくは、 アプリケーション・アクティビティー・トレース および mqat.ini を使用したアクティビティー・トレース動作の構成を参照してください。

キュー・マネージャーはいくつかの規則を適用して、接続に使用するスタンザ設定を決定します。 オプションで、 ApplicationTrace スタンザに一致する接続について、 AllActivityTrace スタンザの下のグローバル・トレース・レベルと頻度の設定をオーバーライドできます。 詳しくは、 mqat.ini を使用したアクティビティー・トレース動作の構成を参照してください。

ディレクトリーの場所

Linux IBM i AIX AIX and Linux および IBM i システムでは、mqat.ini はキュー・マネージャーのデータ・ディレクトリーにあります。このディレクトリーは、 qm.ini ファイルと同じ場所です。

Windows Windows システムでは、mqat.ini はキュー・マネージャーのデータ・ディレクトリー C:\Program Files\IBM\WebSphere MQ\qmgrs\queue_manager_name に配置されています。 トレースされるアプリケーションを実行するユーザーには、このファイルの読み取りアクセス権が必要です。

Multi AllActivity mqat.ini ファイルの Trace スタンザ

mqat.ini 構成ファイルの AllActivityTrace スタンザは、キュー・マネージャーのトレース・レベルを構成するために使用されるパラメーターを指定します。

単一の AllActivityTrace スタンザは、オーバーライドされない限り、すべての IBM MQ 接続に適用されるアクティビティー・トレースの設定を定義します。

AllActivityTrace スタンザの個々の値は、 ApplicationTrace スタンザのより具体的な情報によってオーバーライドできます。

複数の AllActivityTrace スタンザが指定された場合、最後のスタンザにある値が使用されます。 選択された AllActivityTrace に欠落しているパラメーターは、デフォルト値を取ります。 以前の AllActivityTrace スタンザのパラメーターと値は無視されます。

ActivityInterval

トレース・メッセージ間の時間間隔 (秒単位)。 アクティビティー・トレースではタイマー・スレッドが使用されないため、トレース・メッセージは、この時間が経過した瞬間に書き込まれるのではなく、この時間が経過した後に初めて MQI 操作が実行されたときに書き込まれます。 この値が 0 の場合、トレース・メッセージは、接続が切断されたとき (またはアクティビティー・カウントに達したとき) に書き込まれます。 デフォルトは 1 です。

ActivityCount
トレース・メッセージ間の MQI 操作の数。 この値が 0 の場合、トレース・メッセージは、接続が切断されたとき (またはアクティビティー間隔が経過したとき) に書き込まれます。 デフォルトは 100 です。

TraceLevel
各操作ごとにトレースされるパラメーター詳細の量。 個々の操作の説明では、どのパラメーターが各トレース・レベルに含まれているかを詳しく示しています。 LOW、 MEDIUM、または HIGH に設定します。 デフォルトは MEDIUM です。

TraceMessageData
MQGET、MQPUT、 MQPUT1、およびコールバック操作でトレースされたメッセージ・データの量 (バイト単位)。 デフォルトは 0 です。

StopOnGetTraceMsg
ON または OFF に設定できます。 デフォルトは ON です。

SubscriptionDelivery
BATCHED または IMMEDIATE に設定できます。 1 つ以上のアクティビティー・トレース・サブスクリプションが存在する場合に、 **ActivityInterval** パラメーターと **ActivityCount** パラメーターを使用するかどうかを決定します。 このパラメーターを IMMEDIATE に設定すると、トレース・データに一致するサブスクリプションがある場合に、 **ActivityInterval** および **ActivityCount** の値が有効値 1 でオーバーライドされます。 すべてのアクティビティー・トレース・レコードが、同じ接続に由来する他のレコードとまとめられることなく、遅延なしで即時にサブスクリプションに配信されます。 IMMEDIATE 設定を使用すると、アクティビティー・トレース・データを収集する際のパフォーマンス・オーバーヘッドが増加します。 デフォルト設定は BATCHED です。

関連タスク
mqat.ini を使用したアクティビティー・トレース動作の構成

Multi mqat.ini ファイルの ApplicationTrace スタンザ

mqat.ini 構成ファイルには、複数の ApplicationTrace スタンザを含めることができます。 これらのスタンザはそれぞれ、接続のアプリケーション名とルールのマッチングに基づいて、1 つ以上の接続のトレース動作のルールを定義します。

ApplicationTrace スタンザには、以下の値を設定できます。

トレース
ON または OFF に設定できるアクティビティー・トレース・スイッチ。 **Trace** パラメーターは、デフォルト値のない必須パラメーターです。 これをアプリケーション固有のスタンザで使用して、現行アプリケーション・スタンザの有効範囲に対してアクティビティー・トレースがアクティブであるかどうかを判別できます。 この値は、キュー・マネージャーの **ACTVTRC** 設定および **ACTVCONO** 設定をオーバーライドすることに注意してください。

ApplName
ApplName パラメーターは、文字ストリングとして指定され、デフォルトのない必須パラメーターです。 この値を使用して、ApplicationTrace スタンザが適用されるアプリケーションを決定します。 これは、API 出口コンテキスト構造体 (MQMD.PutApplName と同等) の **ApplName** 値と一致します。 **ApplName** 値の内容は、アプリケーション環境によって異なります。

Multiplatforms では、 MQAXC.ApplName は、スタンザ内の値と一致します。 比較が行われるときに、右端のパス・セパレーターの左側の文字は無視されます。

ApplName 値の末尾に単一のワイルドカード文字 (*) を使用して、そのポイント以降の任意の数の文字に一致させることができます。 **ApplName** 値が単一のワイルドカード文字 (*) に設定されている場合、**ApplName** 値はすべてのアプリケーションに一致します。

IBM i **ApplFunction**
ApplFunction パラメーターは、文字ストリングとして指定されます。 デフォルト値は * です。 このパラメーターの値は、ApplicationTrace スタンザおよび **ApplName** 値が適用されるアプリケーション・プログラムを修飾するために使用されます。

このスタンザはオプションであり、IBM i キュー・マネージャーに対してのみ有効です。 単一のワイルドカード文字 (*) を **ApplName** 値の最後に使用すると、任意の数の文字に一致するようにできます。 例えば、 **ApplName** = * および **ApplFunction** = *AMQSPUT0* を指定した ApplicationTrace スタンザは、任意のジョブからの AMQSPUT0 プログラムのすべての呼び出しに適用されます。

ApplClass
ApplClass パラメーターは、アプリケーションのクラスを定義し、以下の値に設定できます。

- USER
- MCA
- ALL (これはデフォルト値です)

AppType 値が IBM MQ 接続にどのように対応するかについては、 mqat.ini を使用したアクティビティー・トレース動作の構成の 表 3 を参照してください。

オプションで、 ApplicationTrace スタンザと一致する接続について、 AllActivityTrace スタンザの下のグローバル・トレース・レベルおよび頻度の設定をオーバーライドすることができます。

ApplicationTrace スタンザの下に以下のパラメーターを設定できます。 これらが設定されていない場合、値は AllActivity トレース・スタンザ 設定から継承されます。

ActivityInterval
トレース・メッセージ間の時間間隔 (秒単位)。 アクティビティー・トレースではタイマー・スレッドが使用されないため、トレース・メッセージは、この時間が経過した瞬間に書き込まれるのではなく、この時間が経過した後に初めて MQI 操作が実行されたときに書き込まれます。 この値が 0 の場合、トレース・メッセージは、接続が切断されたとき (またはアクティビティー・カウントに達したとき) に書き込まれます。 デフォルトは 1 です。

ActivityCount
トレース・メッセージ間の MQI 操作の数。 この値が 0 の場合、トレース・メッセージは、接続が切断されたとき (またはアクティビティー間隔が経過したとき) に書き込まれます。 デフォルトは 100 です。

TraceLevel
各操作ごとにトレースされるパラメーター詳細の量。 個々の操作の説明では、どのパラメーターが各トレース・レベルに含まれているかを詳しく示しています。 LOW、 MEDIUM、または HIGH に設定します。 デフォルトは MEDIUM です。

TraceMessageData
MQGET、MQPUT、 MQPUT1、およびコールバック操作でトレースされたメッセージ・データの量 (バイト単位)。 デフォルトは 0 です。

StopOnGetTraceMsg
ON または OFF に設定できます。 デフォルトは ON です。

関連タスク
mqat.ini を使用したアクティビティー・トレース動作の構成

分散キューイングの構成

このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

始める前に

このセクションを読む前に、チャネル、キュー、および分散キューイングおよびクラスターで導入されたその他の概念について理解しておくと役に立ちます。

別々の物理ネットワーク上に存在する 2 つのキュー・マネージャー、またはファイアウォールを介して通信する 2 つのキュー・マネージャーを接続する必要がある場合は、IBM MQ Internet Pass-Thru を使用すると構成が簡単になる可能性があります。 詳しくは、 IBM MQ Internet Pass-Thru を参照してください。

手順

- 以下のサブトピックにある情報を使用し、分散キューイングを使用してアプリケーションを接続します。
  - 191 ページの『IBM MQ 分散キューイング技法』
  - 211 ページの『分散キュー管理の概要』
  - 214 ページの『別のキュー・マネージャーへのメッセージの送信方法』
  - 235 ページの『チャネルのトリガー操作』
  - 233 ページの『メッセージの安全性』
  - ALW 242 ページの『AIX, Linux, and Windows 上のチャネルのモニターおよび制御』
  - IBM i 265 ページの『IBM i 上のチャネルのモニターおよび制御』

関連概念

914 ページの『IBM MQ for z/OS の設定』
このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

関連タスク

15 ページの『クライアントとサーバー間の接続の構成』
IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャネルを定義します。

287 ページの『キュー・マネージャー・クラスターの構成』
クラスターによって提供されるキュー・マネージャーを相互接続するためのメカニズムにより、初期構成と継続的な管理を簡単に行えます。 クラスター・コンポーネントを定義し、クラスターを作成および管理できます。

84 ページの『Multiplatforms での .ini ファイルの IBM MQ 構成情報の変更』
構成 (`.ini`) ファイル内の情報を編集することにより、IBM MQ または個々のキュー・マネージャーの動作を、ご使用のインストール済み環境のニーズに合わせて変更できます。 IBM MQ MQI clients の構成オプションを変更することもできます。

908 ページの『z/OS でのキュー・マネージャーの構成』
IBM MQ for z/OS でキュー・マネージャーを構成するには、この指示に従ってください。

988 ページの『z/OS での他のキュー・マネージャーとの通信のセットアップ』
このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

IBM MQ 分散キューイング技法

このセクションのサブトピックでは、チャネルを計画する際に役立つ技法について説明します。 これらのサブトピックでは、キュー・マネージャー同士を接続して、アプリケーション間のメッセージのフローを管理する方法を計画するのに役立つ技法について説明します。

メッセージ・チャネルの計画の例については、以下を参照してください。

- ALW AIX, Linux, and Windows におけるメッセージ・チャネルの計画例
- IBM i IBM i におけるメッセージ・チャネルの計画例
- z/OS z/OS におけるメッセージ・チャネルの計画例
- z/OS キュー共有グループを使用する z/OS におけるメッセージ・チャネルの計画例

関連概念

チャネル
メッセージ・キューイングの概要
分散キューイングとクラスター

関連タスク

190 ページの『分散キューイングの構成』
このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

関連資料

構成情報の例

メッセージ・フロー制御

メッセージ・フロー制御は、キュー・マネージャーを結ぶメッセージ経路を準備し、維持するタスクです。これは、多数のキュー・マネージャーを介するマルチ・ホップ経路の場合には重要です。 このセクションでは、運用システム上のキュー、別名キュー定義、およびメッセージ・チャネルを使用してメッセージ・フロー制御を行う方法を説明します。

メッセージ・フローは、190 ページの『分散キューイングの構成』で紹介したいくつかの技法で制御します。 キュー・マネージャーがクラスター内にある場合は、メッセージ・フローは、192 ページの『メッセージ・フロー制御』で説明されているように、別の方法を使用して制御されます。 z/OS キュー・マネージャーがキュー共有グループ内にあり、グループ内キューイング (IGQ) が使用可能である場合は、メッセージ・フローを IGQ エージェントで制御することができます。 これらのエージェントについては、グループ内キューイングで説明します。

以下のオブジェクトを使用して、メッセージ・フロー制御を行うことができます。

- 伝送キュー
- メッセージ・チャネル
- リモート・キュー定義
- キュー・マネージャー別名定義
- 応答先キュー別名定義

キュー・マネージャーおよびキュー・オブジェクトは、オブジェクト・タイプで説明されています。 メッセージ・チャネルについては、分散キューイング・コンポーネントに説明します。 以下の技法では、このようなオブジェクトを使用して運用システム内のメッセージ・フローを作成します。

- リモート・キューへのメッセージの書き込み
- 特定の伝送キューを介した転送
- メッセージの受信
- メッセージのユーザー・システム通過
- メッセージ・フローの分離
- 別の宛先へのメッセージ・フローの切り替え
- 応答先キュー名から別名への解決

注記

このセクションで説明するすべての概念は、ネットワーク内のすべてのノードに関連するものであり、メッセージ・チャネルの送信側と受信側に関与します。 したがって、ほとんどの例ではノードは 1 つだけが示されています。 例外は、メッセージ・チャネルの反対側で管理者が明示的な協調を行う必要がある例です。

個々の技法の説明に進む前に、解決の概念、およびリモート・キュー定義の 3 つの使用法を復習しておくと役に立ちます。 分散キューイングおよびクラスターを参照してください。

関連概念

193 ページの『伝送ヘッダー内のキュー名』
宛先キュー名は、宛先キューに到達するまで、伝送ヘッダーをメッセージと共に移動します。

193 ページの『キュー・マネージャーと応答先別名の作成方法』

このトピックでは、リモート・キュー定義を作成できる 3 つの方法について説明しています。

伝送ヘッダー内のキュー名

宛先キュー名は、宛先キューに到達するまで、伝送ヘッダーをメッセージと共に移動します。

アプリケーションによって使用されるキュー名 (すなわち論理キュー名) は、キュー・マネージャーによって、宛先キュー名として解決されます。 すなわち、物理キュー名です。 この宛先キュー名は、宛先キューに到達するまで、別個のデータ域 (伝送ヘッダー) をメッセージと共に移動します。 その後、伝送ヘッダーは除去されます。

並列のサービス・クラスを作成するときには、このキュー名のキュー・マネージャー部分を変更します。 サービス・クラスの迂回経路の終わりに達したときには、キュー・マネージャー名を必ず元の名前に戻してください。

キュー・マネージャーと応答先別名の作成方法

このトピックでは、リモート・キュー定義を作成できる 3 つの方法について説明しています。

リモート・キュー定義オブジェクトは 3 つの異なる方法で使用されます。 193 ページの表 16 は、この 3 つの定義方法を示しています。

- リモート・キュー定義を使用してローカル・キュー名を再定義する。

  アプリケーションはキューのオープン時にキュー名だけを提供します。このキュー名は、リモート・キュー定義の名前です。

  リモート・キュー定義にはターゲット・キューとキュー・マネージャーの名前が含まれています。 オプションで、この定義には、使用される伝送キューの名前を含めることができます。 伝送キュー名が指定されないと、キュー・マネージャーは、リモート・キュー定義からとったキュー・マネージャー名を伝送キュー名として使用します。 この名前の伝送キューが定義されていないが、デフォルト伝送キューが定義されている場合は、デフォルト伝送キューを使用します。
- リモート・キュー定義を使用してキュー・マネージャー名を再定義する。

  アプリケーションまたはチャネル・プログラムは、キューのオープン時にリモート・キュー・マネージャー名と共にキュー名を指定します。

  すでに指定されているリモート・キュー定義がキュー・マネージャー名と同じ名前になっている場合、定義内のキュー名がブランクのままになっていると、キュー・マネージャーは、オープン・コールで指定されたキュー・マネージャー名を定義内のキュー・マネージャー名で置き換えます。

  さらにこの定義には、使用される伝送キューの名前を指定することができます。 伝送キュー名が指定されないと、キュー・マネージャーは、リモート・キュー定義からとったキュー・マネージャー名を伝送キュー名としてとります。 この名前の伝送キューが定義されていないが、デフォルト伝送キューが定義されている場合は、デフォルト伝送キューを使用します。
- リモート・キュー定義を使用して応答先キュー名を再定義する。

  アプリケーションは、キューにメッセージを入れるたびに、応答メッセージ用に、キュー・マネージャー名をブランクにしたまま応答先キューの名前を指定することがあります。

  応答先キューと同じ名前のリモート・キュー定義を指定すると、ローカル・キュー・マネージャーは、その応答先キュー名を定義で指定されたキュー名によって置き換えます。

  キュー・マネージャー名を定義で指定することはできますが、伝送キュー名は指定できません。

表 *16.* リモート・キュー定義オブジェクトの *3* つの使用方法

使用法	**キュー・マネージャー名**	**キュー名**	**伝送キュー名**
1. リモート・キュー定義 (OPEN 呼び出しで)			
呼び出しで指定	ブランクまたはローカル QM	(*) 必須	適用外
定義で指定	required	required	オプション

表 16. リモート・キュー定義オブジェクトの 3 つの使用方法 (続き)			
使用法	**キュー・マネージャー名**	**キュー名**	**伝送キュー名**
2. キュー・マネージャー別名 (OPEN 呼び出しで)			
呼び出しで指定	(*) 必須かつ非ローカル QM	required	適用外
定義で指定	required	ブランク	オプション
3. 応答先キュー別名 (PUT 呼び出しで)			
呼び出しで指定	ブランク	(*) 必須	適用外
定義で指定	オプション	オプション	ブランク

注 : (*) は、この名前が定義オブジェクトの名前であることを示しています。

詳しくは、キュー名の解決を参照してください。

リモート・キューへのメッセージの書き込み

リモート・キュー定義オブジェクトを使用して、キュー名を隣接するキュー・マネージャーに通じる伝送キューとして解決できます。

分散キューイング環境では、伝送キューとチャネルは、メッセージが運用しているローカル・システムのアプリケーションから出される場合にも、隣接システムからチャネルを介して到達する場合にも、あるロケーションに送られるすべてのメッセージの中心拠点になります。 194 ページの図 6 では、アプリケーションが 'QA_norm' という名前の論理キューにメッセージを入れています。 このネーム・レゾリューションでは、'QA_norm' というリモート・キュー定義を使用して伝送キュー QMB を選択します。 次に、'QA_norm at QMB' というメッセージに伝送ヘッダーを追加します。

隣接システムから 'Channel_back' に到着するメッセージには伝送ヘッダーが付いていて、 例えば 'QA_norm at QMB' などの物理キュー名を示します。 これらのメッセージは、未変更のまま伝送キュー QMB に入ります。

チャネルは、このメッセージを隣接するキュー・マネージャーに送ります。

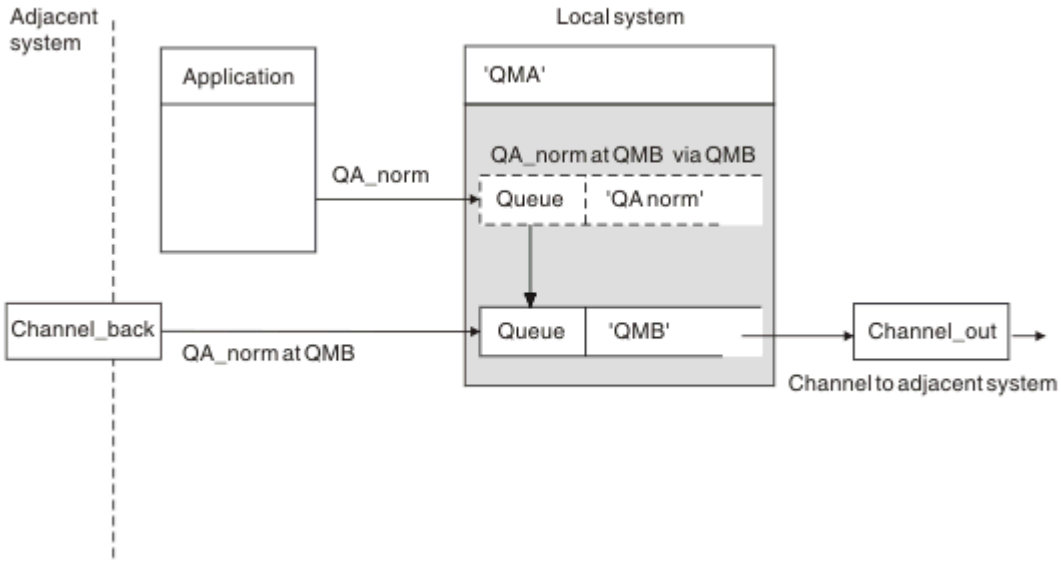


図 6. リモート・キュー定義が使用されて、キュー名が解決され、隣接するキュー・マネージャーに通じる伝送キューが得られます

IBM MQ システム管理者は、以下を行ってください。

- 隣接システムからのメッセージ・チャネルを定義する。
- 隣接システムに向かうメッセージ・チャネルを定義する。
- 伝送キュー QMB を作成する。
- リモート・キュー・オブジェクト 'QA_norm' を定義し、 アプリケーションによって使用されるキュー名を解決して宛先キュー名、宛先キュー・マネージャー名、および伝送キュー名が得られるようにする。

クラスター化環境では、ローカル・キュー・マネージャーにクラスター受信側チャネルを定義するだけで十分です。 伝送キューまたはリモート・キュー・オブジェクトを定義する必要はありません。 クラスターを参照します。

解決に関する補足説明

リモート・キュー定義により、物理的な宛先キュー名およびキュー・マネージャー名が定義されます。 これらの名前は、メッセージの伝送ヘッダーに書き込まれます。

隣接システムからの着信メッセージでは、既に元のキュー・マネージャーによってこのタイプのネーム・レゾリューションが実行されています。 したがって、物理的な宛先キュー名およびキュー・マネージャー名が示されている伝送ヘッダーがあります。 これらのメッセージは、リモート・キュー定義による影響を受けません。

関連資料
キュー名の解決

伝送キューの選択

リモート・キュー定義を使用すると、別の伝送キューが同じ隣接キュー・マネージャーにメッセージを送信することができます。

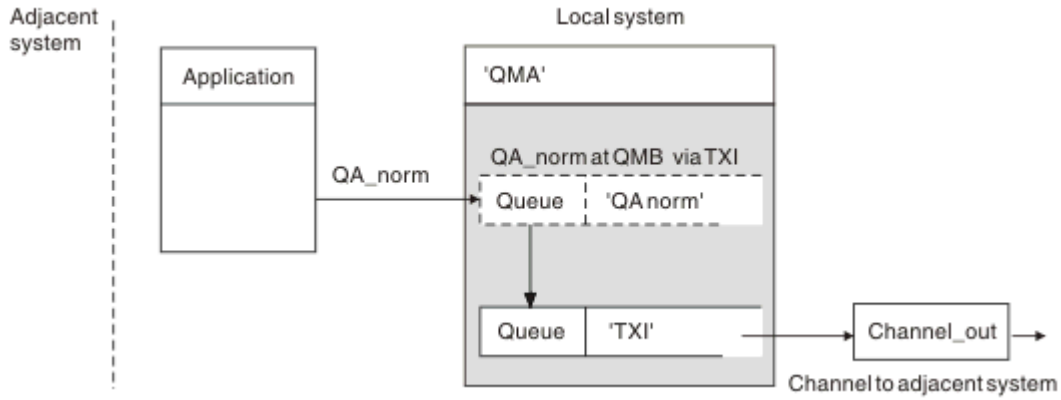


図 7. リモート・キュー定義による異なる伝送キューの使用可能化

分散キューイング環境では、メッセージ・フローをあるチャネルから別のチャネルに変更する必要がある場合には、194 ページの『リモート・キューへのメッセージの書き込み』の 194 ページの図 6 と同じシステム構成を使用してください。 このトピックの 195 ページの図 7 はリモート・キュー定義を使用し、別の伝送キューを介して (したがってまた、別のチャネルを介して) 同じ隣接キュー・マネージャーへメッセージを送る方法を示しています。

195 ページの図 7 に示されている構成の場合には、リモート・キュー・オブジェクト「QA_norm」および伝送キュー「TX1」を提供する必要があります。 「QA_norm」を提供して、リモート・キュー・マネージャーで「QA_norm」キュー、伝送キュー「TX1」、およびキュー・マネージャー「QMB_priority」を選択する必要があります。 隣接システムへのチャネルの定義で、「TX1」を指定します。

メッセージは、'QA_norm at QMB_priority' という内容の伝送ヘッダーと共に伝送キュー 'TX1' に入り、チャネルを介して隣接システムに送られます。

channel_back には、キュー・マネージャー別名が必要なため、この図には示されていません。

クラスター化環境では、伝送キューまたはリモート・キュー定義を定義する必要はありません。 詳しくは、288 ページの『クラスター・キューの定義』 を参照してください。

メッセージの受信

他のキュー・マネージャーからメッセージを受信するようにキュー・マネージャーを構成できます。 意図されないネーム・レゾリューションが起きないことを確認する必要があります。

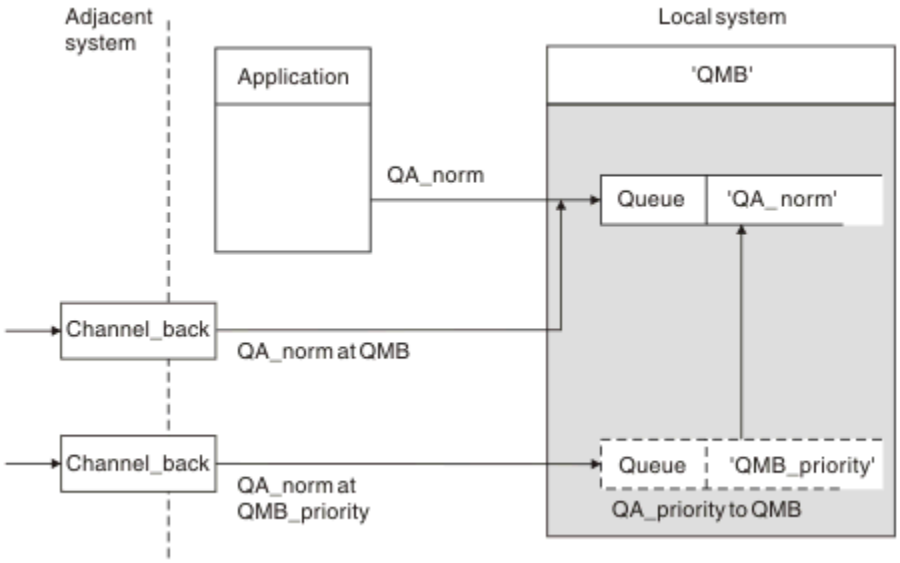


図 *8.* メッセージの直接受信によるキュー・マネージャー別名の解決

システム管理者は、メッセージを送信する準備のほかに、隣接キュー・マネージャーからのメッセージの受信を準備する必要もあります。 受信されたメッセージの伝送ヘッダーには、宛先キュー・マネージャーとキューの物理名が指定されています。 これらのメッセージは、キュー・マネージャー名とキュー名の両方が指定されたローカル・アプリケーションからのメッセージと同じように扱われます。 この処理のため、運用システムに入ってくるメッセージに対して意図されないネーム・レゾリューションが行われないようにしなければなりません。 このシナリオについては 196 ページの図 8 を参照してください。

この構成では、次のものを準備する必要があります。

- 隣接キュー・マネージャーからメッセージを受信するメッセージ・チャネル
- 着信メッセージ・フロー 'QMB_priority' を解決してローカル・キュー・マネージャー名 'QMB' を得るための、キュー・マネージャー別名定義
- ローカル・キュー 'QA_norm' (存在していない場合)

キュー・マネージャー別名の受信

この図におけるキュー・マネージャー別名定義の使用では、別の宛先キュー・マネージャーは選択されていません。 'QMB_priority' とアドレッシングされて、このローカル・キュー・マネージャーを通過するメッセージは、キュー・マネージャー 'QMB' に送られます。 このキュー・マネージャー別名が使用されて、別のメッセージ・フローが作成されます。

メッセージのユーザー・システム通過

ロケーション名の使用、キュー・マネージャーの別名の使用、または伝送キューの選択の 3 つの方法で、メッセージにシステムを通過させることができます。

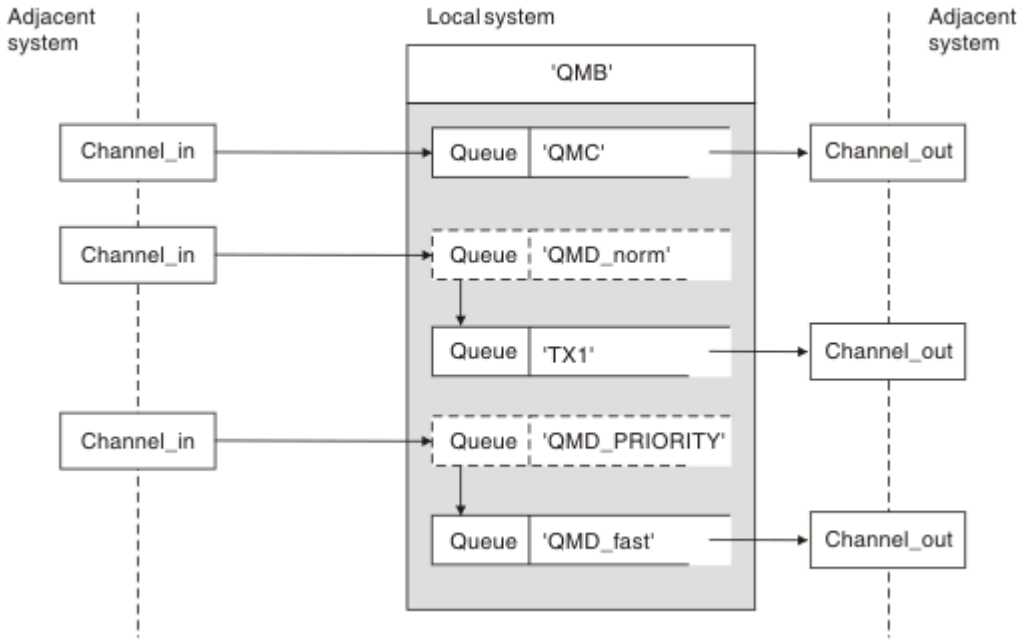


図 9. メッセージが運用システムを通過する 3 つの方法

196 ページの『メッセージの受信』の 196 ページの図 8 では、別名フローを取り込む方法を説明しました。 197 ページの図 9 では、前に説明した技法を組み合わせてネットワークを構成する方法を説明しています。

この構成で示されるチャネルは、次のようなさまざまな宛先を指定された 3 つのメッセージを送達します。

1. QMC の QB
2. QMD_norm の QB
3. QMD_PRIORITY の QB

最初のメッセージ・フローは変更せずにシステムを通過させる必要があります。 2 番目のメッセージ・フローは別の伝送キューおよびチャネルを通過させる必要があります。 2 番目のメッセージ・フローでは、別名キュー・マネージャー名 QMD_norm のメッセージをキュー・マネージャー QMD に解決する必要もあります。 3 番目のメッセージ・フローでは、別の伝送キューが選択されるだけで、それ以外の変更は行われません。

クラスター化環境では、メッセージがクラスター伝送キューを通過します。 通常、単一の伝送キュー SYSTEM.CLUSTER.TRANSMIT.QUEUE は、そのキュー・マネージャーがメンバーになっているすべてのクラスター内のすべてのキュー・マネージャーにすべてのメッセージを転送します。 キュー・マネージャーのクラスターを参照してください。 キュー・マネージャーがメンバーとなっているクラスター内のすべてのキュー・マネージャー、またはその一部に使用する別個の伝送キューを定義できます。

次の方法は、分散キューイング環境に適用できる技法を説明しています。

これらの方法の使用法

これらの構成では、次のものを準備する必要があります。

- 入力チャネル定義
- 出力チャネル定義
- 伝送キュー
  - QMC
  - TX1
  - QMD_fast

- キュー・マネージャー別名定義
  - TX1 経由の QMD への QMD_norm を指定した QMD_norm
  - QMD_fast 経由の QMD_PRIORITY への QMD_PRIORITY を指定した QMD_PRIORITY

注 : この例で示されているいずれのメッセージ・フローでも、宛先キューは変更されていません。 キュー・マネージャー名の別名では、メッセージ・フローの分離が行われています。

方法 1: 着信ロケーション名の使用

伝送ヘッダーに別のロケーション名 (例えば、QMC) が格納されたメッセージを受信するとします。 その場合、最も単純な構成となるのは、その名前 (QMC) を設定した伝送キューを作成することです。 この伝送キューを処理するチャネルが、メッセージをそのままの状態で次の宛先に配信します。

方法 2: キュー・マネージャーの別名の使用

2 番目の方法は、キュー・マネージャー別名オブジェクト定義を使用することです。ただし、新しいロケーション名 QMD および特定の伝送キュー TX1 を指定します。 この処置を行うと、

- キュー・マネージャー名の別名 QMD_norm でセットアップされた別名メッセージ・フローが終了します。これは、指定されたサービス・クラス QMD_norm です。
- メッセージの伝送ヘッダーを QMD_norm から QMD に変更します。

方法 3: 伝送キューの選択

3 番目の方法は、宛先ロケーション QMD_PRIORITY と同じ名前を指定してキュー・マネージャー別名オブジェクトを定義することです。 このキュー・マネージャー別名定義を使用して、特定の伝送キュー QMD_fast、つまり別チャネルを選択します。 これらのメッセージの伝送ヘッダーは、未変更のままになります。

メッセージ・フローの分離

キュー・マネージャー別名を使用して、同じキュー・マネージャーにメッセージを送信するための別個のメッセージ・フローを作成できます。

メッセージを異なるメッセージ・フローに分離する理由

分散キューイング環境では、いくつかの理由で、同じキュー・マネージャーに送られるメッセージを別のメッセージ・フローに分離する必要が生じることがあります。 以下に例を示します。

- 大中小のそれぞれの大きさのメッセージに合わせて別々のフローを用意する必要がある場合。 この必要はクラスター化環境にも当てはまり、その場合、重なり合ったクラスターを作成することもできます。 そのようにする理由は、次のようにいろいろあります。
  - 組織ごとに独自の管理ができるようにする。
  - 独立したアプリケーションを個別に管理できるようにする。
  - サービス・クラスを作成するため。 例えば、STUDENTS というクラスターのサブセットとして STAFF というサブセットを作成できます。 STAFF クラスター内で公示されたキューへメッセージを入れた場合は、制限されたチャネルが使用されます。 STUDENTS クラスター内で公示されたキューへメッセージを入れた場合は、一般チャネルと制限されたチャネルのどちらかが使用されます。
  - テスト環境または実稼働環境を作成する。
- 着信メッセージを、ローカルで生成されたメッセージのパスと異なるパスで経路指定する必要がある場合。
- インストール先で特定の時刻 (例えば、深夜) にメッセージの移動をスケジュールする必要があり、そのメッセージを、スケジュールされるまで用意されたキューに保管しておく必要がある場合。

メッセージ・フローの例

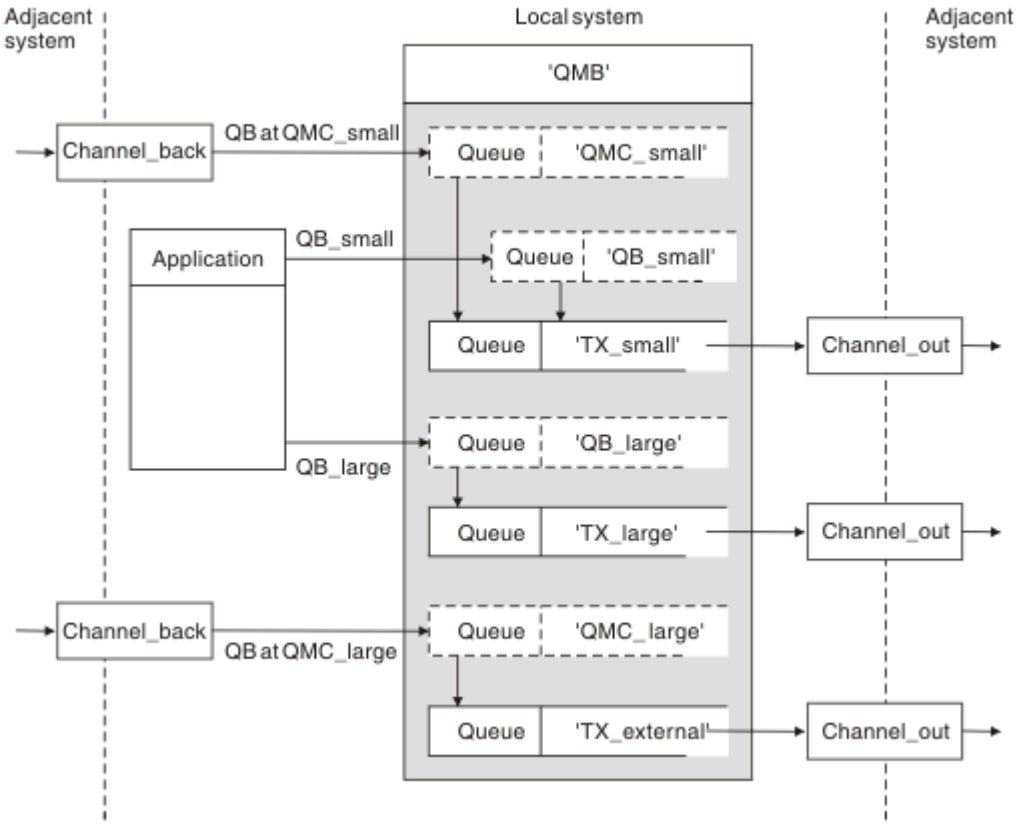


図 *10.* メッセージ・フローの分離

199 ページの図 10 の例で、2 つの着信フローはキュー・マネージャー名の別名 'QMC_small' および 'QMC_large' を宛先としています。 これらのフローにキュー・マネージャー別名定義を提供して、ローカル・キュー・マネージャーのためにこれらのフローを獲得できるようにしてください。 2 つのリモート・キューをアドレッシングするアプリケーションがあり、これらのメッセージ・フローを別個に分けておく必要があるものとします。 ロケーションには同じ値 'QMC' を指定し、伝送キューには別の値を指定して、2 つのリモート・キュー定義を提供します。 この定義は、フローが分離されます。伝送ヘッダーに同じ宛先キュー・マネージャー名が入っているため、相手側での追加処置は必要ありません。 ローカル・システムでは次のものを指定してください。

- 着信チャネル定義
- 2 つのリモート・キュー定義 QB_small および QB_large
- 2 つのキュー・マネージャー別名定義 QMC_small および QMC_large
- 3 つの送信チャネル定義
- 3 つの伝送キュー TX_small、TX_large、および TX_external

隣接システムとの調整

別個のメッセージ・フローを作成するためにキュー・マネージャー別名を使用するときには、メッセージ・チャネルのリモート側のシステム管理者と連携してこの作業を行い、リモート側で対応するキュー・マネージャーの別名が有効になるようにする必要があります。

さまざまなロケーションに送られるメッセージの集中

さまざまなロケーションにあてられたメッセージを単一のチャネルに集中させます。

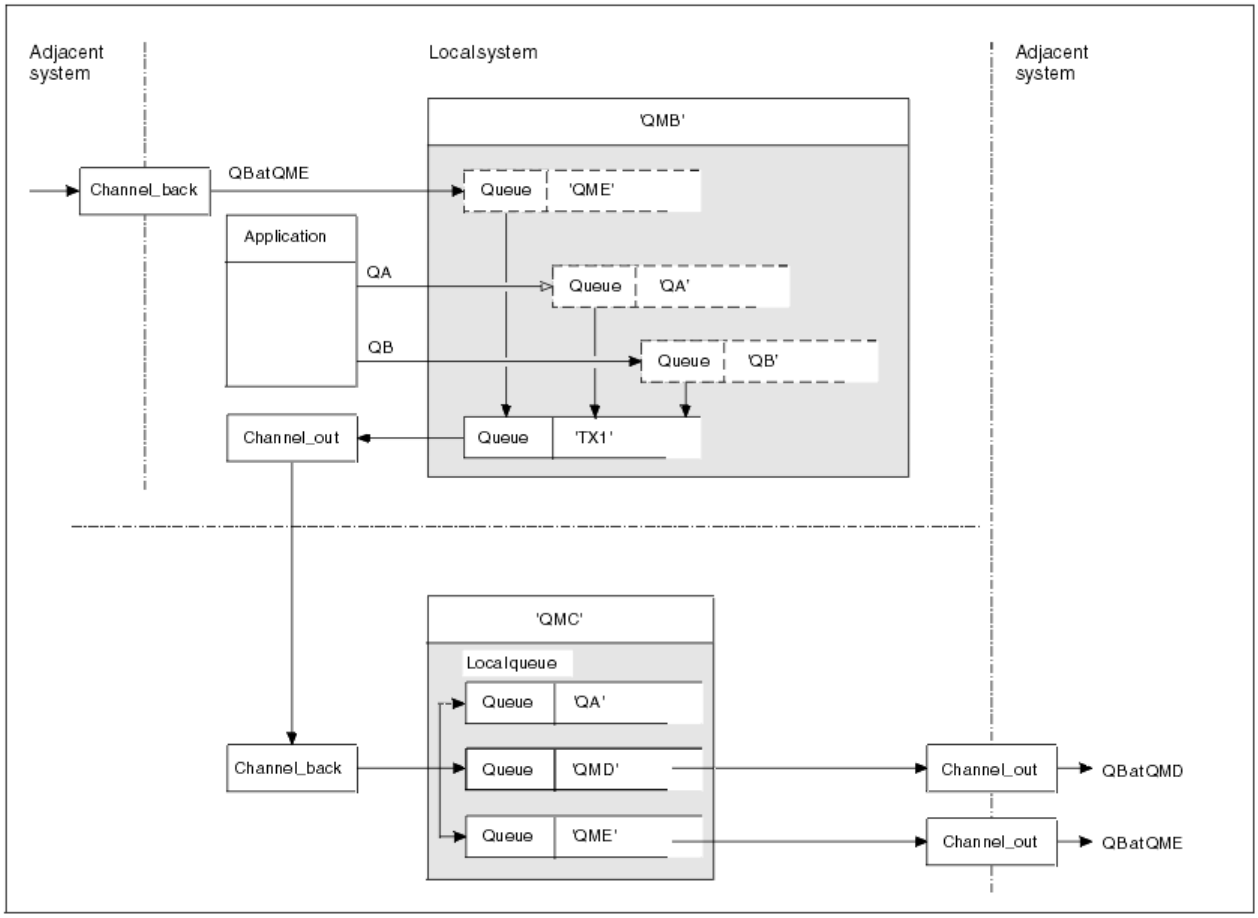


図 *11.* チャネルへのメッセージ・フローの結合

200 ページの図 11 は、さまざまなロケーションにあてられたメッセージを 1 つのチャネルに集中させる分散キューイングの技法を示しています。 可能な使用法には次の 2 つがあります。

- ゲートウェイを使用してメッセージ・トラフィックを集中させる
- ノード間で通信容量の高い高速経路を使用する

この例では、ローカル・システムおよび隣接システムのさまざまなソースからメッセージが発行されます。これらメッセージは、宛先キューも取り扱うキュー・マネージャーも異なっていますが、いずれも伝送キュー 'TX1' を介してキュー・マネージャー QMC に送られています。 キュー・マネージャー QMC は、宛先に応じてメッセージを送信します。 あるものは伝送キュー 'QMD' に、さらにキュー・マネージャー QMD に伝送されるように設定されます。 別のものは、伝送キュー 'QME' に、さらにキュー・マネージャー QME に伝送されるように設定されます。 他のメッセージは、ローカル・キュー 'QA' に書き込まれます。

以下を指定してください。

- チャネル定義
- 伝送キュー TX1
- リモート・キュー定義
  - 'TX1 経由の QMC の QA' を指定した QA
  - 'TX1 経由の QMD の QB' を指定した QB
- キュー・マネージャー別名定義
  - 'TX1 経由の QME' を指定した QME

QMC を構成する管理者アシスタントは以下を指定してください。

- 同じチャネル名を指定した受信側チャネル定義

- 関連する送信側チャネル定義を指定した伝送キュー QMD
- 関連する送信側チャネル定義を指定した伝送キュー QME
- ローカル・キュー・オブジェクト QA

別の宛先へのメッセージ・フローの転送

キュー・マネージャー別名および伝送キューを使用して、特定のメッセージの宛先を再定義できます。

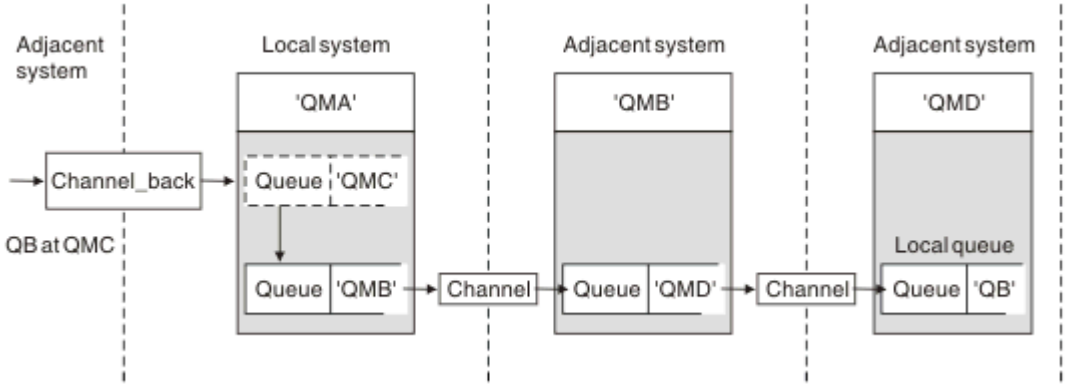


図 *12.* 別の宛先へのメッセージ・ストリームの転送

201 ページの図 12 では、特定のメッセージの宛先を再定義する方法を示しています。 QMA への着信メッセージは 'QMC の QB' に宛てて送られます。 通常は QMA に到達して、QMC に至るチャネルの一部である QMC という伝送キューに入ります。 QMA はこのメッセージを QMD に転送する必要がありますが、QMD には QMB を通らなければ到達できません。 この方法は、あるロケーションから別のロケーションにサービスを移動する場合に便利であり、加入者は、メッセージが新しいアドレスに合わせて調整されるまでの間、一時的な方法でメッセージの送信を続けることができます。

特定のキュー・マネージャーに向けられた着信メッセージを別のキュー・マネージャーに再経路指定する方法では、次のものを使用します。

- 宛先キュー・マネージャーを別のキュー・マネージャーに変更して、隣接システムに至る伝送キューを選択するための、キュー・マネージャー別名
- 隣接キュー・マネージャーで使用される伝送キュー
- 宛先キュー・マネージャーに向けてさらにルーティングするための、隣接キュー・マネージャーにおける伝送キュー

以下を指定してください。

- Channel_back 定義
- QMB 経由の QMD の QB を指定したキュー・マネージャー別名オブジェクト定義 QMC
- Channel_out 定義
- 関連の伝送キュー QMB

QMB を構成する管理者アシスタントは以下を指定してください。

- 対応する channel_back 定義
- 伝送キュー QMD
- QMD に通じる関連チャネル定義

クラスター化環境の中では、別名が使用できます。 詳細については、384 ページの『キュー・マネージャー別名とクラスター』を参照してください。

配布先リストへのメッセージの送信

単一の MQPUT 呼び出しを使用して、アプリケーションが複数の宛先にメッセージを送信するようにできます。

z/OS 以外のすべてのプラットフォームの IBM MQ では、アプリケーションは 1 回の MQPUT 呼び出しで複数の宛先にメッセージを送信できます。 分散キューイング環境とクラスター化環境の両方でそれを行うことができます。 配布先リストで説明されているように、宛先を配布先リストで定義する必要があります。

すべてのキュー・マネージャーが配布先リストをサポートしているとは限りません。 MCA は、パートナーとの接続を確立するときに、パートナーが配布先リストをサポートしているかどうかを判別し、 それに応じて伝送キューにフラグを設定します。 アプリケーションが、配布先リストに指定されているメッセージを送信しようとしたところ、パートナーが配布先リストをサポートしていない場合、送信側の MCA がメッセージを代行受信して、各宛先ごとに 1 回、伝送キューに書き込みます。

受信側の MCA は、配布先リストに送信したメッセージがすべての宛先で安全に受信されたことを確認します。 いずれかの宛先で受信が失敗した場合、MCA は、どれが失敗したかを特定します。 その後、例外報告書を作成し、メッセージの送信の再試行を行うことができます。

応答先キュー

応答先キューを使用すると、完全なリモート・キュー処理ループを 作成できます。

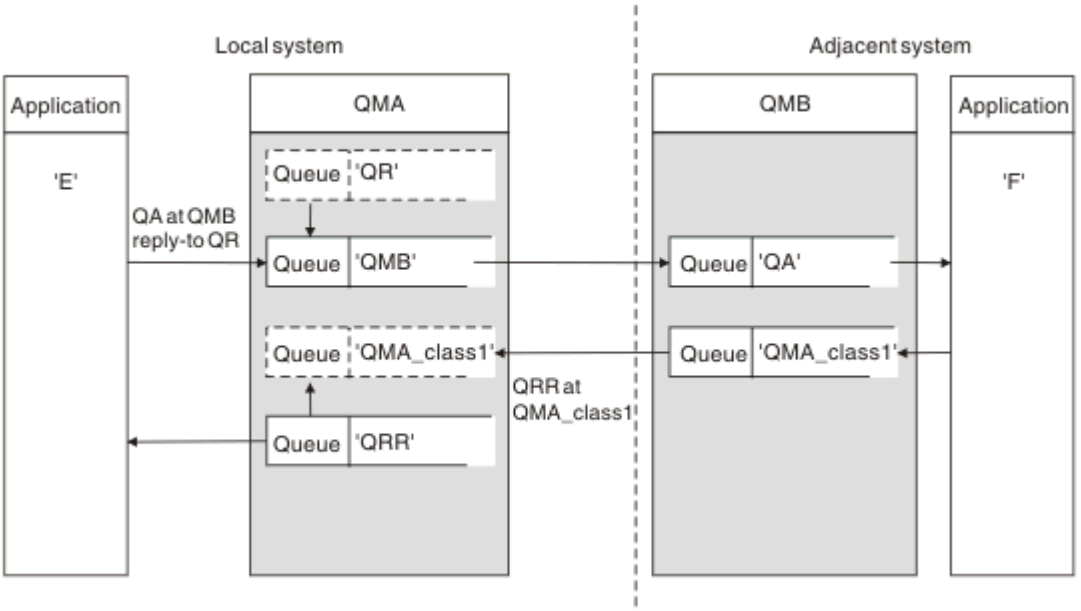


図 *13.* 書き込み *(PUT)* 呼び出し時の応答先キュー名の置換

応答先キューを使用する完全なリモート・キュー処理ループが 202 ページの図 13 に示されています。 このループは、分散キューイング環境とクラスター化環境の両方に適用されます。 詳細については、209 ページの表 20 を参照してください。

アプリケーションは QMB にある QA をオープンし、そのキューにメッセージを書き込みます。 このメッセージには QR という応答先キュー名が指定され、キュー・マネージャー名は指定されません。 キュー・マネージャー QMA は応答先キュー・オブジェクト QR を検出し、その中から QRR という別名とキュー・マネージャー名 QMA_class1 を取り出します。 これらの名前はメッセージの応答先フィールドに書き込まれます。

QMB にあるアプリケーションから発行されたメッセージは、QMA_class1 にある QRR に向けてアドレッシングされます。 キュー・マネージャーによってキュー・マネージャー別名定義 QMA_class1 が使用され、メッセージがそのキュー・マネージャー自体およびキュー QRR に流れるようになります。

このシナリオでは、応答メッセージのサービス・クラスを選択する機能を、 アプリケーションに与える方法を示しています。 クラスは、 QMB 上の伝送キュー QMA_class1 と、 QMA 上のキュー・マネージャー別名定義 QMA_class1 を併用することによって実装されています。 この方法で、アプリケーションの応答先キューを変更して、アプリケーションには影響を与えずにフローを分離できます。 アプリケーションでは、常に、この特定のサービス・クラスの QR を選択します。 この応答先キュー定義 QR を使用して、 サービス・クラスを変更することができます。

以下を作成してください。

- 応答先キュー定義 QR
- 伝送キュー・オブジェクト QMB
- Channel_out 定義
- Channel_back 定義
- キュー・マネージャー別名定義 QMA_class1
- ローカル・キュー・オブジェクト QRR (まだ存在していない場合)

隣接システムの管理者アシスタントは、以下を作成してください。

- 受信チャネル定義
- 伝送キュー・オブジェクト QMA_class1
- 関連する送信チャネル
- ローカル・キュー・オブジェクト QA

アプリケーション・プログラムでは以下のものを使用します。

- 書き込み呼び出しで応答先キュー名 QR
- 読み取り呼び出しでキュー名 QRR

このようにして、アプリケーションの関与なしで、 必要に応じてサービス・クラスを変更することができます。 応答先別名「QR」を、伝送キュー「QMA_class1」およびキュー・マネージャー別名「QMA_class1」と共に変更します。

メッセージがキューに書き込まれるときに応答先別名オブジェクトが検出されなかった場合は、 ブランクの応答先キュー・マネージャー名フィールドに ローカル・キュー・マネージャー名が挿入されます。 応答先キュー名は未変更のままになります。

解決の制約事項

元のメッセージが書き込まれたときに 「QMA」上で応答先キューのネーム・レゾリューションが行われているため、「QMB」上ではそれ以上のネーム・レゾリューションは実行できません。 したがって、応答側アプリケーションは、 応答先キューの物理名を指定してメッセージをキューに書き込みます。

応答先キューとして使用する名前と、戻りメッセージが検出される実際のキューの名前とは異なることを、アプリケーションが認識している必要があります。

例えば、アプリケーションが使用するための 2 つのサービス・クラスが、「C1_alias」および「C2_alias」という応答先キュー別名で指定された場合、アプリケーションはメッセージ書き込み呼び出しでこれらの名前を応答先キュー名として使用します。 ただし、アプリケーションは、実際にはメッセージが、「C1_alias」の場合はキュー「C1」に、 「C2_alias」の場合はキュー「C2」に入ることを予期します。

ただし、アプリケーションは、応答先別名キューに対して照会呼び出しを出して、応答メッセージを取得するために使用する必要のある実際のキューの名前を調べることができます。

関連概念

193 ページの『キュー・マネージャーと応答先別名の作成方法』
このトピックでは、リモート・キュー定義を作成できる 3 つの方法について説明しています。

204 ページの『応答先キュー別名の例』
この例では、戻されるメッセージ用に別の経路 (伝送キュー) を選択するために応答先別名を使用する方法が示されています。 この機能を使用するためには、アプリケーションと連携して応答先キュー名を変更する必要があります。

205 ページの『この例の作業内容』
例およびキュー・マネージャーが応答先キュー別名を使用する方法についての説明。

206 ページの『応答先キュー別名の概要』
アプリケーションがリモート・キューにメッセージを書き込んでから、同じアプリケーションが応答先キューの別名から応答メッセージを削除するまでのプロセスのウォークスルーです。

応答先キュー別名の例

この例では、戻されるメッセージ用に別の経路 (伝送キュー) を選択するために応答先別名を使用する方法が示されています。 この機能を使用するためには、アプリケーションと連携して応答先キュー名を変更する必要があります。

204 ページの図 14 に示すように、応答メッセージが、伝送キュー、チャネル、およびキュー・マネージャー別名の入った戻り経路を利用できなければなりません。

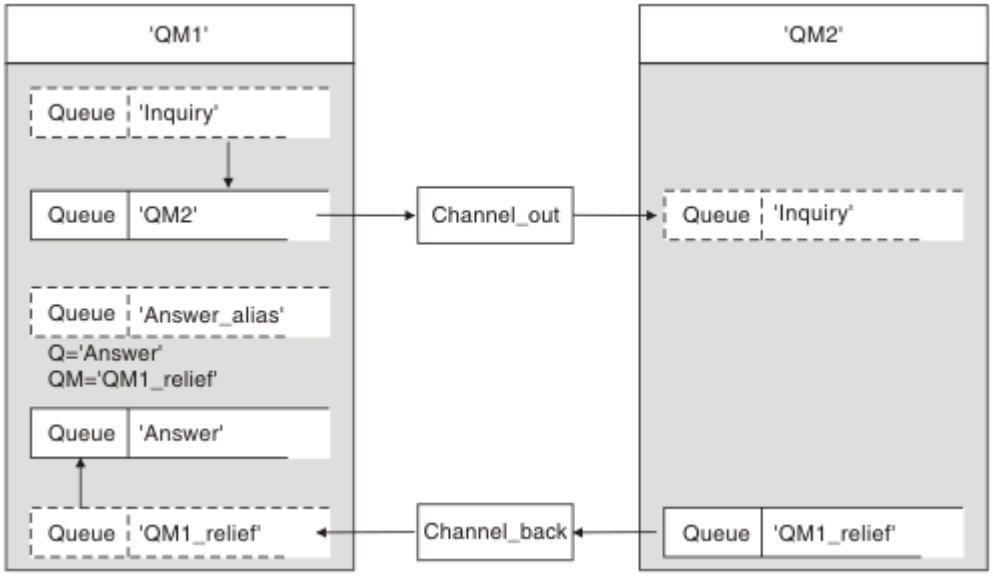


図 *14.* 応答先キュー別名の例

この例は、'QM2' にあるサーバー・アプリケーションにメッセージを送る、'QM1' にある要求側アプリケーションを表しています。 サーバーのメッセージは、伝送キュー 'QM1_relief' を使用して、代替チャネルを介して返されます (デフォルトの戻りチャネルは、伝送キュー 'QM1' によって提供されます)。

この応答先キュー別名には、'Answer_alias' というリモート・キュー定義が特定の方法で使用されています。 QM1 にあるアプリケーションは、キュー 'Inquiry' に入れるすべてのメッセージの応答先フィールドに、この 'Answer_alias' という名前を書き込みます。

応答先キュー定義 'Answer_alias' は、'Answer at QM1_relief' として定義されます。 QM1 にあるアプリケーションは、応答が 'Answer' というローカル・キューに入るものと想定します。

QM2 にあるサーバー・アプリケーションは、受信したメッセージの応答先フィールドを使用して、QM1 にある要求側に対する応答メッセージ用のキューおよびキュー・マネージャーの名前を取得します。

この例の QM1 で使用されている定義

QM1 の IBM MQ システム管理者は、他のオブジェクトと共に応答先キュー 'Answer' が作成されるようにしなければなりません。 '*' が付いているキュー・マネージャー別名の名前は、'*' が付いている応答先のキュー別名定義のキュー・マネージャー名と一致している必要があります。

オブジェクト	**定義**	
ローカル伝送キュー	QM2	
リモート・キュー定義	オブジェクト名	照会
	リモート・キュー・マネージャー名	QM2
	リモート・キュー名	照会
	伝送キュー名	QM2 (DEFAULT)
キュー・マネージャー別名	オブジェクト名	QM1_relief *

オブジェクト	定義	
	キュー・マネージャー名	QM1
	キュー名	(ブランク)
応答先キュー別名	オブジェクト名	Answer_alias
	リモート・キュー・マネージャー名	QM1_relief *
	リモート・キュー名	Answer

QM1 での書き込み定義

アプリケーションは応答先フィールドに応答先キュー別名を入れ、キュー・マネージャー名フィールドをブランクのままにします。

フィールド	内容
キュー名	照会
キュー・マネージャー名	(ブランク)
応答先キュー名	Answer_alias
応答先キュー・マネージャー	(ブランク)

この例の QM2 で使用されている定義

QM2 の IBM MQ システム管理者は、着信メッセージのためのローカル・キューが存在していることを確認し、正しい名前の伝送キューを応答メッセージで利用できるようにしなければなりません。

オブジェクト	定義
ローカル・キュー	照会
伝送キュー	QM1_relief

QM2 での書き込み定義

QM2 にあるアプリケーションは元のメッセージから応答先キュー名とキュー・マネージャー名を取り出し、応答先キューに応答メッセージを書き込むときにはその名前を使用します。

フィールド	内容
キュー名	Answer
キュー・マネージャー名	QM1_relief

この例の作業内容

例およびキュー・マネージャーが応答先キュー別名を使用する方法についての説明。

この例で、QM1 にある要求側アプリケーションは常に、書き込み呼び出しの関連フィールド内で応答先キューとして 'Answer_alias' を使用します。 それらは常に 'Answer' というキューからメッセージを取り出します。

応答先キュー 'Answer' の名前および戻り経路 'QM1_relief' の名前を変更するために、応答先キュー別名定義を QM1 のシステム管理者が使用できるようになっています。

QM1 のアプリケーションは応答がキューに入ることを予想しているため、このキュー名 'Answer' を変更することは通常は有効ではありません。 ただし、QM1 システム管理者は、必要に応じて戻り経路 (サービス・クラス) を変更できます。

キュー・マネージャーが応答先キュー別名を使用する方法

キュー・マネージャー QM1 は、アプリケーションによって書き込み呼び出しで指定された応答先キュー名が応答先キュー別名と同じであって、キュー・マネージャー部分がブランクになっている場合は、応答先キュー別名から定義を取り出します。

キュー・マネージャーは、書き込み呼び出しで指定された応答先キュー名を定義内の応答先キュー名によって置き換えます。 また、書き込み呼び出しでブランクになっているキュー・マネージャー名を定義内のキュー・マネージャー名によって置き換えます。

これらの名前は、メッセージ記述子に入ってメッセージと共に送られます。

表 *17.* 応答先キュー別名

フィールド名	**書き込み呼び出し**	**伝送ヘッダー**
応答先キュー名	Answer_alias	Answer
応答先キュー・マネージャー名	(ブランク)	QM1_relief

応答先キュー別名の概要

アプリケーションがリモート・キューにメッセージを書き込んでから、同じアプリケーションが応答先キューの別名から応答メッセージを削除するまでのプロセスのウォークスルーです。

この例を完成するために、プロセスについて説明します。

1. アプリケーションが 'Inquiry' というキューをオープンして、その中にメッセージを書き込みます。 アプリケーションは、メッセージ記述子の応答先フィールドに次の値を入れます。

応答先キュー名	**Answer_alias**
応答先キュー・マネージャー名	(ブランク)

2. キュー・マネージャー 'QM1' が 'Answer_alias' という名前のリモート・キュー定義の有無を調べて、ブランクのキュー・マネージャー名に応答します。 この名前のリモート・キュー定義がない場合、キュー・マネージャーは、メッセージ記述子の応答先キュー・マネージャー・フィールドにそれ自身の名前 'QM1' を入れます。
3. 'Answer_alias' という名前のリモート・キュー定義が見つかった場合、キュー・マネージャーは、その定義からキュー名 ('Answer') およびキュー・マネージャー名 ('QM1_relief') を取り出します。 次に、それをメッセージ記述子の応答先フィールドに入れます。
4. キュー・マネージャー「QM1」は、リモート・キュー定義「Inquiry」を使用して、送信先キューがキュー・マネージャー 「QM2」 であるかどうかを判別し、メッセージが伝送キュー「QM2」に置かれていることを判別します。 「QM2」は、キュー・マネージャー「QM2」のキューに向けられたメッセージのデフォルトの送信キュー名である。
5. キュー・マネージャー 'QM1' は、メッセージをこの伝送キューに入れるときに、メッセージに伝送ヘッダーを追加します。 このヘッダーには、宛先キューの名前 'Inquiry' と宛先キュー・マネージャー 'QM2' が入っています。
6. メッセージはキュー・マネージャー 'QM2' に到達し、ローカル・キュー 'Inquiry' に入ります。
7. アプリケーションはこのキューからメッセージを受け取り、メッセージを処理します。 そして、応答メッセージを用意して、元のメッセージのメッセージ記述子から得られた応答先キュー名にその応答メッセージを入れます。

応答先キュー名	**Answer**
応答先キュー・マネージャー名	QM1_relief

8. キュー・マネージャー 'QM2' は書き込みコマンドを実行します。 さらに、キュー・マネージャー名 'QM1_relief' がリモート・キュー・マネージャーであることを調べ、同じ名前の伝送キュー 'QM1_relief' にメッセージを入れます。 このメッセージには、宛先キューの名前 'Answer' と宛先キュー・マネージャー 'QM1_relief' の入った伝送ヘッダーが付けられます。

9. メッセージはキュー・マネージャー 'QM1' に転送されます。 キュー・マネージャーは、キュー・マネージャー名 'QM1_relief' が別名であることを認識し、別名定義 'QM1_relief' から物理キュー・マネージャー名 'QM1' を取り出します。
10. キュー・マネージャー 'QM1' は、伝送ヘッダーに収められているキュー名 'Answer' にメッセージを入れます。
11. アプリケーションがキュー 'Answer' から応答メッセージを取り出します。

ネットワーキングの考慮事項

分散キューイング環境では、メッセージの宛先は、キュー名とキュー・マネージャー名だけでアドレッシングされるため、特定の規則が適用されます。

1. キュー・マネージャー名が指定され、その名前がローカル・キュー・マネージャーの名前と異なっている場合、次のいずれかが成立すること。
   - 伝送キューは同じ名前が使用可能でなければならない。 この伝送キューが別のキュー・マネージャーにメッセージを移動するためのメッセージ・チャネルの一部でなければならない。
   - キュー・マネージャー名を解決して同じキュー・マネージャー名または別のキュー・マネージャー名、およびオプションの伝送キューを得るためにキュー・マネージャー別名定義が存在していること。
   - 伝送キュー名が解決できないで、デフォルト伝送キューが定義されている場合は、デフォルト伝送キューが使用される。
2. キュー名だけが指定された場合は、同じ名前の任意のタイプのキューがローカル・キュー・マネージャーで使用可能でなければなりません。 このキューは、隣接キュー・マネージャーへの伝送キュー、キュー・マネージャー名、およびオプションの伝送キューを解決する、リモート・キュー定義である場合があります。

これがクラスタリング環境でどのように機能するかを調べるには、クラスターを参照してください。

z/OS キュー共有グループ内でキュー・マネージャーを実行している場合に、グループ内キューイング (IGQ) が使用可能であると、SYSTEM.QSG.TRANSMIT.QUEUE を使うことができます。 詳しくは、グループ内キューイングを参照してください。

分散キューイング環境で、あるキュー・マネージャーから別のキュー・マネージャーへメッセージを移動するメッセージ・チャネルのシナリオについて考えてみます。

移動されるメッセージは、ネットワーク内の他のいずれかのキュー・マネージャーから出されたものであり、それらのメッセージのうちには、未知のキュー・マネージャー名が宛先として指定されて到達するものもあります。 例えば、キュー・マネージャー名が変更されたりシステムから削除されたりした場合に、この問題が発生することがあります。

チャネル・プログラムは、これらのメッセージの伝送キューを見つけられない場合にこの状況を認識し、メッセージを未配布メッセージ (送達不能) キューに入れます。 このようなメッセージの有無を調べて正しい宛先に転送できるようにする必要があります。 または、宛先が確認できる、発信元にメッセージを戻します。

元のメッセージでレポート・メッセージが要求されていた場合は、このような状況になると例外報告書が生成されます。

解決の規則

宛先キューの識別名を変更するネーム・レゾリューション (すなわち、論理名から物理名への変更) は、発信元のキュー・マネージャーでのみ、1 回だけなされます。

それ以降は、各種の可能な別名は、メッセージ・フローの分離および組み合わせを行うときにのみ使用しなければなりません。

戻り経路

メッセージには、キューおよびキュー・マネージャーの名前の形式で戻りアドレスが入ることがあります。この戻りアドレスの形式は、分散キューイング環境とクラスター化環境の両方で使用できます。

このアドレスは、通常はメッセージを作成するアプリケーションによって指定されます。 それは、あとでそのメッセージを取り扱うアプリケーション (ユーザー出口アプリケーションを含む) によって変更できます。

このアドレスの発信元とは関係なく、メッセージを扱うアプリケーションは、応答、状況、またはレポート・メッセージをメッセージ発信元アプリケーションに戻すためにこのアドレスを選択できます。

これらの応答メッセージの経路指定方法は、元のメッセージの経路指定方法と変わりません。 別のキュー・マネージャーに向けて作成するメッセージ・フローには、対応する戻りフローが必要です。

物理名の矛盾

宛先の応答先キュー名は、解決されて元のキュー・マネージャーにある物理キュー名になっています。 応答元のキュー・マネージャーで再び解決してはなりません。

そのようにすると、名前の矛盾の問題が発生することがあります。 この矛盾は、ネットワーク全体で物理キュー名と論理キュー名に関する合意を得なければ回避できません。

キュー名変換の管理

キュー・マネージャー別名定義またはリモート・キュー定義を作成するときには、その名前を伴う各メッセージごとにネーム・レゾリューションが行われます。 この状態は管理されていなければなりません。

以下の説明は、隣接システムへのメッセージ・チャネルを備えた個々のシステムを扱うアプリケーション設計者およびチャネル計画者向けの説明です。 ここでは、ローカル側から見たチャネルの計画と制御について説明します。

キュー・マネージャー別名定義またはリモート・キュー定義を作成するときには、メッセージのソースとは無関係に、その名前を伴う各メッセージごとに解決が行われます。 キュー・マネージャー・ネットワーク内の多数のキューが関連している可能性があるため、この状況を把握するには、以下の内容を示す表を保持する必要があります。

- 発信元キューと発信元キュー・マネージャーの名前、それに関連する解決されたキュー名、解決されたキュー・マネージャー名、解決された伝送キュー名、および解決方法
- 発信元キューの名前と以下のものとの関係
  - 解決された宛先キュー名
  - 解決された宛先キュー・マネージャー名
  - 伝送キュー
  - メッセージ・チャネル名
  - 隣接システム名
  - 応答先キュー名

注 : このコンテキストで使用されるソース *(*発信元*)* という用語は、アプリケーションが提供するキュー名またはキュー・マネージャー名、あるいはメッセージを書き込むためにキューをオープンする場合のチャネル・プログラムを指します。

これらのそれぞれの表の例が 209 ページの表 18、209 ページの表 19、および 209 ページの表 20 に示されています。

これらの表で使用されている名前は、このセクションの例から取られたものであり、この表は 1 つのノードのキュー名解決の実例として示されたものではありません。

表 18. キュー・マネージャー *QMA* におけるキュー名解決

キューがオープンされるときに、入力元キューが指定される	キューのオープン時に、指定されたソース・キュー・マネージャー	解決されたキュー名	解決されたキュー・マネージャー名	解決された伝送キュー名	解決タイプ
QA_norm	-	QA_norm	QMB	QMB	リモート・キュー
(任意)	QMB	-	-	QMB	(なし)
QA_norm	-	QA_norm	QMB	TX1	リモート・キュー
QB	QMC	QB	QMD	QMB	キュー・マネージャー別名

表 19. キュー・マネージャー *QMB* におけるキュー名解決

キューがオープンされるときに、入力元キューが指定される	キューのオープン時に、指定されたソース・キュー・マネージャー	解決されたキュー名	解決されたキュー・マネージャー名	解決された伝送キュー名	解決タイプ
QA_norm	-	QA_norm	QMB	-	(なし)
QA_norm	QMB	QA_norm	QMB	-	(なし)
QA_norm	QMB_PRIORITY	QA_norm	QMB	-	キュー・マネージャー別名
(任意)	QMC	(任意)	QMC	QMC	(なし)
(任意)	QMD_norm	(任意)	QMD_norm	TX1	キュー・マネージャー別名
(任意)	QMD_PRIORITY	(任意)	QMD_PRIORITY	QMD_fast	キュー・マネージャー別名
(任意)	QMC_small	(任意)	QMC_small	TX_small	キュー・マネージャー別名
(任意)	QMC_large	(任意)	QMC_large	TX_external	キュー・マネージャー別名
QB_small	QMC	QB_small	QMC	TX_small	リモート・キュー
QB_large	QMC	QB_large	QMC	TX_large	リモート・キュー
(任意)	QME	(任意)	QME	TX1	キュー・マネージャー別名
QA	QMC	QA	QMC	TX1	リモート・キュー
QB	QMD	QB	QMD	TX1	リモート・キュー

表 20. キュー・マネージャー *QMA* における応答先キュー名変換

アプリケーション設計		応答先別名定義	
ローカル QMGR	メッセージ用のキュー名	応答先キュー別名	定義後

表 20. キュー・マネージャー *QMA* における応答先キュー名変換 (続き)			
アプリケーション設計		**応答先別名定義**	
QMA	QRR	QR	QMA_class1 の QRR

チャネル・メッセージ・シーケンス番号付け

チャネルはシーケンス番号を使用して、メッセージが伝送キューから取り出された順序と同じ順序で送達されることを検査します。

チャネルのシーケンス番号の不一致は、メッセージがチャネルを流れ始めて初めて検出される。 不一致が発生した場合、チャネルのいずれかの側で永続的な同期データが失われたことを意味する。例えば、ディザスタリカバリ（DR）構成や、チャネルが疑わしいときにバッチ処理の終了が中断された場合などである。

シーケンス番号の不一致をリセットまたは無視する (*qm.ini* ファイルの *Channels* スタンザの **IgnoreSeqNumberMismatch** を参照) と、メッセージのバッチが失われたり重複したりするリスクはなく、チャネルの未確定状況はリセットされません。

この情報は、DISPLAY CHSTATUS を使用して表示できます。 最後にバッチで転送されたメッセージのシーケンス番号と LUWID という名前の ID は、持続ストレージに保管されます。 これらの値をチャネルの開始時に使用して、どのメッセージの転送が正常に完了したかをリンクの両側で確認することができます。

メッセージの順次検索

あるアプリケーションが同じ宛先キューに一連のメッセージを投入した場合、以下の条件を満たせば、それらのメッセージは 1 つのアプリケーションから一連の MQGET 操作で順番に取り出すことができる：

- すべての書き込み要求が同じアプリケーションから出された。
- すべての書き込み要求が同じ作業単位から行われたか、あるいはすべての書き込み要求が作業単位の外部で行われた。
- どのメッセージも同じ優先順位をもっている。
- どのメッセージも同じ持続性をもっている。
- リモート・キューイングの場合に、書き込み要求を行うアプリケーションから、キュー・マネージャーまたは相互通信を経由して、宛先キュー・マネージャーとターゲット・キューにつながる経路が 1 つしかないように構成されている。
- メッセージが送達不能キューに入れられない (例えば、キューが一時的に満杯になっているため)。
- メッセージを読み取るアプリケーションが、例えば特定の *MsgId* または *CorrelId* を指定するなどの方法で、検索順序を意図的に変更していない。
- 宛先キューからのメッセージを検索するために、1 つのアプリケーションだけが読み取り操作を行う。 複数のアプリケーションがある場合、送信側アプリケーションによって書き込まれた各メッセージ順序内のすべてのメッセージを読み取れるようにすべてのアプリケーションを設計しなければなりません。

注 : 一連のメッセージが単一の作業単位によって書き込まれた場合であっても、他のタスクおよび作業単位からのメッセージがこれらのメッセージ順序の間に散在していることがあります。

上記の条件が満たされない場合に、ターゲット・キューでのメッセージの順序は重要なので、その順序を守りたいときは、メッセージに独自のメッセージ・シーケンス番号を組み込むようにアプリケーションをコード化してもかまいません。

高速、非持続メッセージの順次検索

高速チャネル上の非持続メッセージは、同じチャネル上の持続メッセージに置き換わるため、順序どおりに到着しません。 受信 MCA は、非持続メッセージを宛先キューに即時に書き込み、それらを可視にします。 持続性メッセージは次の同期点まで可視になりません。

ループバック・テスト

ループバック・テスト は、z/OS 以外のプラットフォームで、接続は実際に別のマシンにリンクしないで通信リンクをテストできる手法です。

接続は、別々のマシン上にあるかのように 2 つのキュー・マネージャー間にセットアップされますが、同じマシンの別のプロセスにループバックして接続をテストします。 この技法は、アクティブなネットワークを必要とせずに通信コードをテストできることを意味します。

それを行う方法は、使用している製品およびプロトコルによって異なります。

Windows システムでは、「ループバック」アダプターを使用できます。

詳細については、使用している製品の資料を参照してください。

経路トレースおよびアクティビティー記録

2 つの方法で、メッセージが一連のキュー・マネージャーを通る経路を確認することができます。

IBM MQ 経路表示アプリケーション (制御コマンド **dspmqrte** によって使用可能) を使用するか、アクティビティー記録を使用できます。 これらの両方のトピックは、監視リファレンスで説明されています。

分散キュー管理の概要

分散キュー管理 (DQM) は、キュー・マネージャー間の通信を定義および制御するのに使用します。

分散キュー管理は以下のことを行います。

- キュー・マネージャー間の通信チャネルを定義および制御できるようにします。
- あるタイプのローカル・キュー (伝送キュー) からローカル・システム上の通信リンクに、また通信リンクから宛先キュー・マネージャーのロケーションにあるローカル・キューにメッセージを移動するための、メッセージ・チャネル・サービスを提供します。
- パネル、コマンド、およびプログラムを使用して、チャネルの作動のモニターと問題の診断を行うための機能を提供します。

チャネル定義は、伝送キュー、通信リンク ID 、およびチャネル属性にチャネル名を関連付けます。 チャネル定義の実現方法は、プラットフォームごとに異なります。 メッセージの送受信はメッセージ・チャネル・エージェント (MCA) と呼ばれるプログラムによって制御されます。このプログラムは、チャネル定義を使用して、通信の開始および制御を行います。

MCA は DQM によって直接制御されます。 この制御構造はプラットフォームによって決まりますが、一般的にはオペレーター・コマンドとパネルの他、リスナーとトリガー・モニターが含まれます。

メッセージ・チャネル とは、あるキュー・マネージャーから別のキュー・マネージャーにメッセージを移動させるための、1 方向のパイプです。 したがって、1 つのメッセージ・チャネルは、MCA のペアによって表される 2 つのエンドポイントをもつことになります。 各エンドポイントで、メッセージ・チャネルの側が定義されています。 例えば、一方の側で送信側を定義し、他方の側で受信側を定義します。

チャネルの定義方法については、以下の章を参照してください。

- ALW 242 ページの『AIX, Linux, and Windows 上のチャネルのモニターおよび制御』
- z/OS 992 ページの『z/OS 上のチャネルのモニターおよび制御』
- IBM i 265 ページの『IBM i 上のチャネルのモニターおよび制御』

メッセージ・チャネルの計画の例については、以下を参照してください。

- ALW AIX, Linux, and Windows におけるメッセージ・チャネルの計画例
- IBM i IBM i におけるメッセージ・チャネルの計画例
- z/OS z/OS におけるメッセージ・チャネルの計画例
- z/OS キュー共有グループを使用する z/OS におけるメッセージ・チャネルの計画例

チャネル出口については、メッセージ・チャネル用のチャネル出口プログラムを参照してください。

関連概念

212 ページの『メッセージの送受信』
以下の図は、メッセージが送信されるときのエンティティー間の関係を詳しく表す、分散キュー管理モデルを示しています。 また、制御のフローも示しています。

220 ページの『チャネル制御機能』
チャネル制御機能は、チャネルを定義、モニター、および制御するための機能を備えています。

233 ページの『メッセージを送達できない場合の処理』
メッセージを送達できない場合、MCA はメッセージをいくつかの方法で処理できます。 再試行するか、送信側へ返信するか、または送達不能キューへ書き込むことができます。

238 ページの『初期設定および構成ファイル』
チャネル初期設定データの処理は、運用している IBM MQ プラットフォームによって異なります。

239 ページの『メッセージのデータ変換』
IBM MQ メッセージは、異なるキュー・マネージャーでのキュー間で送信したときには、データ変換を必要とする場合があります。

240 ページの『独自のメッセージ・チャネル・エージェントの作成』
IBM MQ を使用すれば、独自のメッセージ・チャネル・エージェント (MCA) プログラムの記述、または独立ソフトウェア・ベンダーからのインストールを行うことができます。

240 ページの『分散キュー管理に関するその他の考慮事項』
分散キュー管理用に IBM MQ を準備する際に考慮する、その他のトピックです。 このトピックでは、未配布メッセージ・キュー、 使用中のキュー、システム拡張およびユーザー出口プログラム、ならびにトラステッド・アプリケーションとしてチャネルおよびリスナーを実行する方法 について説明します。

関連資料

構成情報の例

メッセージの送受信

以下の図は、メッセージが送信されるときのエンティティー間の関係を詳しく表す、分散キュー管理モデルを示しています。 また、制御のフローも示しています。

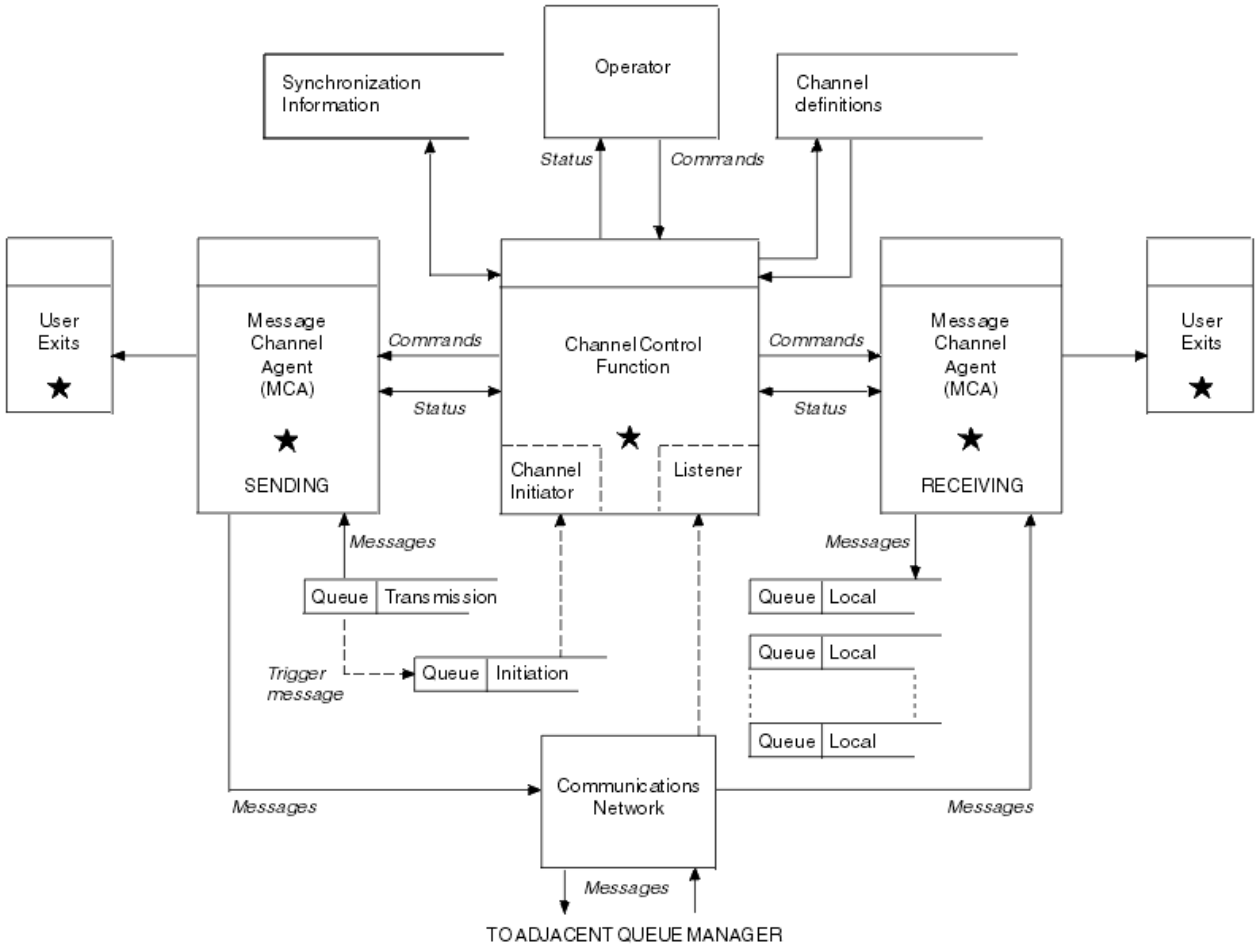


図 *15.* 分散キュー管理モデル

注 :

1. プラットフォームに応じて、チャネルごとに 1 つの MCA があります。 特定のキュー・マネージャーには、1 つまたは複数のチャネル制御機能が備わっていることがあります。
2. MCA の実装およびチャネル制御機能は、プラットフォームに強く依存しています。 これらは、プログラム、プロセス、またはスレッドのいずれかです。それぞれは、1 つのエンティティーの場合もあり、いくつかの独立した部分またはリンクしている部分から構成される、複数のエンティティーの場合もあります。
3. 星印の付いたコンポーネントはすべて MQI を使用できます。

チャネル・パラメーター

MCA は、次のようないくつかの方法でパラメーターを受信します。

- コマンドで開始された場合は、データ域にチャネル名が渡されます。 すると、MCA はチャネル定義を直接読み取り、その属性を取得します。
- 送信側の場合、および一部のサーバー・チャネルの場合、MCA は、キュー・マネージャーのトリガー操作によって、自動的に開始されます。 必要に応じてトリガー・プロセス定義からチャネル名が取り出され、MCA に渡されます。 それ以外の処理は、上記の場合と同じです。 サーバー・チャネルが起動するように設定されるのは、完全修飾されている、つまり CONNAME を指定して接続する場合のみです。
- 送信側、サーバー、要求側またはクライアント接続からリモートで開始された場合、チャネル名はパートナーのメッセージ・チャネル・エージェントからの初期データに入って渡されます。 MCA はチャネル定義を直接読み取り、その属性を取得します。

チャネル定義内で定められていない以下の属性は折衝することもできます。

分割メッセージ
いずれかの側で分割メッセージがサポートされていない場合には、分割メッセージは送信されません。

変換機能
コード・ページ変換または数値エンコード変換が必要なときに、一方の側でこのような変換が行えない場合は、他方の側で行う必要があります。 必要な場合にいずれの側でもこの変換がサポートされない場合には、チャネルは開始できません。

配布リスト・サポート
一方の側が配布先リストをサポートしない場合、パートナーの MCA は伝送キューにフラグを設定して、複数の宛先に向けられたメッセージを代行受信することが分かるようにします。

チャネル状況およびシーケンス番号

メッセージ・チャネル・エージェント・プログラムは、各チャネルの現行シーケンス番号、論理作業単位番号、およびチャネルの一般状況の記録を保持します。 プラットフォームによっては、この状況情報を表示して、チャネルの制御に役立てることができるものもあります。

別のキュー・マネージャーへのメッセージの送信方法

このセクションでは、キュー・マネージャー間でメッセージを送信する最も簡単な方法を、前提条件および必要な権限を含めて説明します。 他の方法を使用してリモート・キュー・マネージャーにメッセージを送信することもできます。

あるキュー・マネージャーから別のキュー・マネージャーへメッセージを送信する前に、以下のステップを実行する必要があります。

1. 選択した通信プロトコルが使用可能であることをチェックします。
2. キュー・マネージャーを開始します。
3. チャネル・イニシエーターを開始します。
4. リスナーを開始します。

また、必要なオブジェクトを作成するには、適切な IBM MQ セキュリティー権限をもっている必要があります。

あるキュー・マネージャーから別のキュー・マネージャーへメッセージを送信するには、次のようにします。

- 発信元キュー・マネージャーに次のオブジェクトを定義します。
  - 送信側チャネル
  - リモート・キュー定義
  - 開始キュー (z/OS z/OS の場合は必須、その他はオプション)
  - 伝送キュー
  - 送達不能キュー
- ターゲット・キュー・マネージャーに次のオブジェクトを定義します。
  - 受信側チャネル
  - ターゲット・キュー
  - 送達不能キュー

ご使用の IBM MQ プラットフォームに応じてそれぞれ異なる方法を使用してこれらのオブジェクトを定義することができます。

- すべてのプラットフォームで、 MQSC コマンド 管理タスクの自動化で説明されているプログラマブル・コマンド・フォーマット (PCF) コマンド、または IBM MQ エクスプローラーで説明されている IBM MQ スクリプト・コマンド (MQSC) を使用できます。
- z/OS z/OS では、IBM MQ for z/OS の管理で説明されている操作パネルおよび制御パネルも使用できます。

- IBM i IBM i では、パネル・インターフェースも使用できます。

別のキュー・マネージャーにメッセージを送信するためのコンポーネントの作成についての詳細は、以下のサブトピックを参照してください。

関連概念

191 ページの『IBM MQ 分散キューイング技法』
このセクションのサブトピックでは、チャネルを計画する際に役立つ技法について説明します。 これらのサブトピックでは、キュー・マネージャー同士を接続して、アプリケーション間のメッセージのフローを管理する方法を計画するのに役立つ技法について説明します。

211 ページの『分散キュー管理の概要』
分散キュー管理 (DQM) は、キュー・マネージャー間の通信を定義および制御するのに使用します。

235 ページの『チャネルのトリガー操作』
IBM MQ には、キューで特定の条件が満たされると自動的にアプリケーションを開始する機能が用意されています。 この機能は、トリガー操作と呼ばれています。

233 ページの『メッセージの安全性』
IBM MQ の通常のリカバリー機能の他に、分散キュー管理は、メッセージ・チャネルの両側で調整された同期点プロシージャーを使用して、メッセージが適切に送達されることを確認します。 このプロシージャーはエラーを検出すると、問題を調査できるようにチャネルをクローズし、チャネルが再開されるまでメッセージを伝送キューに安全に保管します。

関連タスク

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

242 ページの『AIX, Linux, and Windows 上のチャネルのモニターおよび制御』
DQM では、リモート・キュー・マネージャーに対するチャネルの作成、モニターおよび制御が必要です。 コマンド、プログラム、IBM MQ Explorer、チャネル定義ファイル、および同期情報のストレージ域を使用して、チャネルを制御できます。

265 ページの『IBM i 上のチャネルのモニターおよび制御』
リモート・キュー・マネージャーにつながるチャネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。 各キュー・マネージャーには、互換性のあるリモート・キュー・マネージャーとの相互接続を制御するための DQM プログラムが備わっています。

15 ページの『クライアントとサーバー間の接続の構成』
IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャネルを定義します。

287 ページの『キュー・マネージャー・クラスターの構成』
クラスターによって提供されるキュー・マネージャーを相互接続するためのメカニズムにより、初期構成と継続的な管理を簡単に行えます。 クラスター・コンポーネントを定義し、クラスターを作成および管理できます。

988 ページの『z/OS での他のキュー・マネージャーとの通信のセットアップ』
このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

チャネルの定義

あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信するには、2 つのチャネルを定義する必要があります。 1 つのチャネルはソース・キュー・マネージャーに、もう 1 つのチャネルはターゲット・キュー・マネージャーに定義する必要があります。

発信元キュー・マネージャーの場合

チャネル・タイプが SENDER のチャネルを定義します。 以下を指定する必要があります。

- 使用される伝送キューの名前 (XMITQ 属性)。

- パートナー・システムの接続名 (CONNAME 属性)。
- 使用している通信プロトコル名 (TRPTYPE 属性)。 IBM MQ for z/OS では、プロトコルは TCP または LU6.2 でなければなりません。 他のプラットフォームでは、これを指定する必要はありません。 そのままにしておき、デフォルトのチャネル定義から値を選ぶことができます。

すべてのチャネル属性の詳細は、チャネル属性に記載されています。

ターゲット・キュー・マネージャーの場合
チャネル・タイプ RECEIVER で、送信側チャネルと同じ名前のチャネルを定義します。

使用している通信プロトコル名を指定します (TRPTYPE 属性)。 IBM MQ for z/OS では、プロトコルは TCP または LU6.2 でなければなりません。 他のプラットフォームでは、これを指定する必要はありません。 そのままにしておき、デフォルトのチャネル定義から値を選ぶことができます。

受信側チャネルの定義は汎用にすることができます。 これは、同じ受信側と通信する複数のキュー・マネージャーがある場合に、送信側チャネルですべて受信側に同じ名前を指定でき、1 つの受信側の定義がそれらすべてに適用されることを意味します。

チャネルを定義するとき、PING CHANNEL コマンドを使用してテストできます。 このコマンドは、送信側チャネルから受信側チャネルに特別なメッセージを送信し、それが返されることをチェックします。

注 : TRPTYPE パラメーターの値は、応答メッセージ・チャネル・エージェントで無視されます。 例えば、送信側チャネル定義の TRPTYPE が TCP であっても、パートナーの受信側チャネル定義の TRPTYPE は正しく LU62 で始まります。

キューの定義

あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信するには、最大 6 個のキューを定義する必要があります。 そのうち 4 個以下のキューはソース・キュー・マネージャーに、2 個以下のキューはターゲット・キュー・マネージャーに定義する必要があります。

発信元キュー・マネージャーの場合

- リモート・キュー定義

  この定義では、以下を指定します。

  リモート・キュー・マネージャー名
  ターゲット・キュー・マネージャーの名前です。

  リモート・キュー名
  ターゲット・キュー・マネージャーのターゲット・キューの名前です。

  伝送キュー名
  伝送キューの名前です。 この伝送キューの名前を指定する必要はありません。 指定しない場合、ターゲット・キュー・マネージャーと同じ名前の伝送キューが使用されます。 それが存在しない場合には、デフォルトの伝送キューが使用されます。 伝送キューをターゲット・キュー・マネージャーと同じ名前にして、デフォルトでキューが検出されるようにすることをお勧めします。

- 開始キュー定義

  z/OS これは必須です。 SYSTEM.CHANNEL.INITQ という開始キューを使用しなければなりません。

  Multi これはオプションです。 開始キュー SYSTEM.CHANNEL.INITQ の名前を指定することを考慮してください。

- 伝送キュー定義

  XMITQ に設定された USAGE 属性をもつローカル・キューです。 IBM i IBM MQ for IBM i 固有のインターフェースを使用する場合、USAGE 属性は *TMQ となります。

- 送達不能キュー定義

  未配布メッセージを書き込むことのできる送達不能キューの定義を選択してください。

ターゲット・キュー・マネージャーの場合

- ローカル・キュー定義

ターゲット・キュー。 このキューの名前は、発信元キュー・マネージャーのリモート・キュー定義のリモート・キュー名フィールドに指定された名前と同じでなければなりません。

- 送達不能キュー定義

  未配布メッセージを書き込むことのできる送達不能キューの定義を選択してください。

関連概念

217 ページの『伝送キューの作成』
チャネル (要求側チャネル以外) を開始する前に、このセクションの説明に従って伝送キューを定義しておく必要があります。 チャネル定義において、伝送キューの名前が指定されている必要があります。

217 ページの『IBM i での伝送キューの作成』
「MQM キューの作成」パネルを使用して、IBM i プラットフォームで伝送キューを作成できます。

伝送キューの作成

チャネル (要求側チャネル以外) を開始する前に、このセクションの説明に従って伝送キューを定義しておく必要があります。 チャネル定義において、伝送キューの名前が指定されている必要があります。

それぞれの送信側メッセージ・チャネルについて、USAGE 属性を XMITQ に設定して、ローカル・キューを定義してください。 リモート・キュー定義で、特定の伝送キューを使用したい場合は、以下で示すように、リモートのキューを作成してください。

伝送キューを作成するには、次の例に示すように、IBM MQ コマンド (MQSC) を使用してください。

伝送キューの作成例

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') USAGE(XMITQ)
```

リモート・キューの作成例

```
DEFINE QREMOTE(PAYROLL) DESCR('Remote queue for QM2') +
XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

伝送キューには、上記の例のように、リモート・システムのキュー・マネージャーの名前を付けることを考慮してください。

IBM i での伝送キューの作成

「MQM キューの作成」パネルを使用して、IBM i プラットフォームで伝送キューを作成できます。

それぞれの送信側メッセージ・チャネルについて、「Usage」フィールド属性を *TMQ に設定して、ローカル・キューを定義する必要があります。

リモート・キュー定義を利用する場合、同じコマンドを使用して、タイプが *RMT で、使用法が *NORMAL のキューを作成してください。

伝送キューを作成するには、コマンド行から CRTMQMQ コマンドを使用して、最初のキュー作成パネルを表示します。218 ページの図 16 を参照してください。

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Queue name . . . . . .

Queue type . . . . . .  ____     *ALS, *LCL, *MDL, *RMT

Message Queue Manager name . . .  *DFT_____________________________
_____









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

図 *16.* キューの作成 *(1)*

キューの名前を入力し、作成したいキューのタイプとしてローカル (*LCL)、リモート (*RMT)、または別名 (*ALS) のいずれかを指定してください。 伝送キューの場合、このパネルにローカル (*LCL) を指定して Enter キーを押してください。

「Create MQM Queue (MQM キューの作成)」パネルの 2 ページ目が表示されます。218 ページの図 17 を参照してください。

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Queue name . . . . . > HURS.2.HURS.PRIORIT

Queue type . . . . . > *LCL     *ALS, *LCL, *MDL, *RMT
Message Queue Manager name . . .  *DFT
Replace . . . . . .  *NO      *NO, *YES
Text 'description' . . . . . .  '                    '
Put enabled . . . . . .  *YES     *SYSDFTQ, *NO, *YES
Default message priority . . . .  0       0-9, *SYSDFTQ
Default message persistence . .  *NO      *SYSDFTQ, *NO, *YES
Process name . . . . . .  '                    '
Triggering enabled . . . . . .  *NO      *SYSDFTQ, *NO, *YES
Get enabled . . . . . .  *YES     *SYSDFTQ, *NO, *YES
Sharing enabled . . . . . .  *YES     *SYSDFTQ, *NO, *YES




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

図 *17.* キューの作成 *(2)*

表示されたデフォルト値を変更したい場合は、変更してください。 Page Down (次ページ) キーを押して次の画面を表示させます。219 ページの図 18 を参照してください。

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Default share option . . . . . *YES     *SYSDFTQ, *NO, *YES
Message delivery sequence . . .  *PTY     *SYSDFTQ, *PTY, *FIFO
Harden backout count . . . . . *NO      *SYSDFTQ, *NO, *YES
Trigger type . . . . . *FIRST    *SYSDFTQ, *FIRST, *ALL...
Trigger depth . . . . . 1       1-999999999, *SYSDFTQ
Trigger message priority . . . . 0       0-9, *SYSDFTQ
Trigger data . . . . . '                   '
Retention interval . . . . . 999999999   0-999999999, *SYSDFTQ
Maximum queue depth . . . . . 5000     1-24000, *SYSDFTQ
Maximum message length . . . . . 4194304    0-4194304, *SYSDFTQ
Backout threshold . . . . . 0       0-999999999, *SYSDFTQ
Backout requeue queue . . . . . '                  '
Initiation queue . . . . . '                  '



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

図 *18.* キューの作成 *(3)*

このパネルの「Usage (使用法)」フィールドに伝送キューを意味する *TMQ を入力し、その他のフィールドに表示されているデフォルト値を必要に応じて変更してください。

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Usage . . . . . *TMQ     *SYSDFTQ, *NORMAL, *TMQ
Queue depth high threshold . . .  80      0-100, *SYSDFTQ
Queue depth low threshold . . .  20      0-100, *SYSDFTQ
Queue full events enabled . . .  *YES     *SYSDFTQ, *NO, *YES
Queue high events enabled . . .  *YES     *SYSDFTQ, *NO, *YES
Queue low events enabled . . . .  *YES     *SYSDFTQ, *NO, *YES
Service interval . . . . . 999999999   0-999999999, *SYSDFTQ
Service interval events . . . .  *NONE     *SYSDFTQ, *HIGH, *OK, *NONE
Distribution list support . . .  *NO      *SYSDFTQ, *NO, *YES
Cluster Name . . . . . *SYSDFTQ
Cluster Name List . . . . . *SYSDFTQ
Default Binding . . . . . *SYSDFTQ   *SYSDFTQ, *OPEN, *NOTFIXED




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

図 *19.* キューの作成 *(4)*

フィールドに適切なデータが入っていることを確認してから、Enter キーを押してキューを作成してください。

チャネルの開始

発信元キュー・マネージャーで定義されるリモート・キューにメッセージを書き込むとき、これらのメッセージはチャネルが開始されるまで伝送キューに保管されます。 チャネルが開始されると、メッセージはリモート・キュー・マネージャーのターゲット・キューに送達されます。

送信側キュー・マネージャーでチャネルを開始するには、START CHANNEL コマンドを使用します。 送信側チャネルを開始すると、受信側チャネルは (リスナーによって) 自動的に開始され、メッセージがターゲット・キューに送信されます。 メッセージを転送するには、メッセージ・チャネルの両側が共に実行されていなければなりません。

チャネルの両側には別のキュー・マネージャーがあるため、チャネルは別の属性で定義されている場合があります。 この差を解消するには、チャネルの開始時にチャネルの両側で初期データ折衝が行われます。 一般に、チャネルの両側はより少ないリソースを必要とする属性で作動します。 これにより、より大規模なシステムが、メッセージ・チャネルの片側でより小規模なシステムのより少ないリソースに適応できます。

送信 MCA は、チャネル間の送信前に大きなメッセージを分割します。 分割されたメッセージは、リモート・キュー・マネージャーで再度組み立てられます。 これは、ユーザーからは分かりません。

MCA は、複数のスレッドを使用してメッセージを転送できます。 パイプライン化 と呼ばれるこのプロセスにより、MCA は待ち状態になることが少なくなり、メッセージをより効率的に転送できます。 パイプライン化により、チャネルのパフォーマンスが向上します。

チャネル制御機能

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

コマンドは、パネル、プログラム、またはコマンド行から出されて、チャネル制御機能に渡されます。 パネル・インターフェースは、チャネル状況およびチャネル定義データの表示も行います。 プログラマブル・コマンド・フォーマットまたはそれらの IBM MQ コマンド (MQSC) を使用して、242 ページの『AIX, Linux, and Windows 上のチャネルのモニターおよび制御』に説明されているコマンドを制御することができます。

これらのコマンドは、以下のグループに分類されます。

- チャネル管理
- チャネル制御
- チャネル状況モニター

チャネル管理コマンドは、チャネルの定義について扱います。 これらを使用して以下のことができます。

- チャネル定義の作成
- チャネル定義のコピー
- チャネル定義の変更
- チャネル定義の削除

チャネル制御コマンドは、チャネルの動作を管理します。 これらを使用して以下のことができます。

- チャネルの開始
- チャネルの停止
- パートナーとの再同期 (一部のインストール先で)
- メッセージ・シーケンス番号のリセット
- メッセージの未確定バッチの解決
- ping。チャネルを介するテスト通信の送信

チャネル・モニターでは、以下のようなチャネルの状態が表示されます。

- 現行のチャネル設定
- チャネルがアクティブか非アクティブか
- チャネルが同期化状態で終了したか

関連概念

問題判別に役立つ情報の入手先

チャネルの準備

メッセージ・チャネルまたは MQI チャネルを開始しようとする前に、そのチャネルを準備しておく必要があります。 ローカルおよびリモートのチャネル定義の全属性が 正しく、かつ互換性のある設定になっていなければなりません。

チャネル属性は、チャネルの定義と属性を説明しています。

明示的にチャネル定義を設定しても、 チャネル開始時に行われるチャネル折衝によって定義の値が部分的に 指定変更される場合があります。 この動作は正常であり、ユーザーには意識されない仕方で行われます。 これは、そのままでは矛盾する定義同士を一緒に使用できるようにするために調整されている動作です。

受信側チャネルおよびサーバー接続チャネルの自動定義

z/OS 以外のすべてのプラットフォームの IBM MQ では、適切なチャネル定義がない場合、自動定義が有効になっている受信側チャネルまたはサーバー接続チャネルに対して定義が自動的に作成されます。 定義は、以下のものを使用して作成されます。

1. 適切なモデルのチャネル定義、SYSTEM.AUTO.RECEIVER、または SYSTEM.AUTO.SVRCONN。 自動定義用のモデル・チャネル定義は、システム・デフォルトの SYSTEM.DEF.RECEIVER、および SYSTEM.DEF.SVRCONN と同じです。ただし、49 文字のブランクが続く「Auto-defined by (自動定義に使用するモデル・チャネル定義)」の説明フィールドを除きます。 システム管理者は、システムに備えられたモデル・チャネル定義の任意の部分を選択して変更できます。
2. パートナー・システムからの情報。 チャネル名およびシーケンス番号の折り返し値には、パートナーの値が使用されます。
3. 自動定義で作成される値を変更するために使用できる、チャネル出口プログラム。 チャネル自動定義出口プログラムを参照してください。

その後、記述がチェックされ、自動定義出口プログラムによって定義が変更されたのかまたはモデル定義が変更されたためなのかを判別します。 最初の 44 文字がまだ 29 文字のブランクが続く「Auto-defined by (自動定義に使用するモデル・チャネル定義)」の場合、キュー・マネージャー名が追加されます。 最終の 20 文字がまだすべてブランクの場合は、ローカルの時刻と日付が追加されます。

定義が作成され、保管されると、 その定義が以前から引き続き存在していた場合と同じ条件でチャネルの開始が行われます。 バッチ・サイズ、伝送サイズ、メッセージ・サイズがパートナーと折衝されます。

その他のオブジェクトの定義

メッセージ・チャネルを開始するには、 チャネルの両端をそれぞれのキュー・マネージャーで 定義しておく (または自動定義を使用可能にしておく) 必要があります。 チャネルがサービスを行う伝送キューを、送信側でキュー・マネージャーに対して定義しておく必要があります。 通信リンクを定義し、使用可能にしておかなければなりません。 場合によっては、その他の IBM MQ オブジェクト (リモート・キュー定義、キュー・マネージャー別名定義、応答先キュー別名定義など) を準備して、190 ページの『分散キューイングの構成』に記載しているシナリオを実現できるようにする必要もあります。

MQI チャネルの定義については、29 ページの『MQI チャネルの定義』を参照してください。

伝送キュー当たりの複数メッセージ・チャネル

1 つの伝送キューに複数のチャネルを定義することが可能ですが、これらのチャネルのうち、一度にアクティブにしておけるのは 1 つだけです。 このオプションは、通信量の平衡化およびリンク障害の訂正アクション用に、 キュー・マネージャー間に代替経路を用意する方法と考えてください。 伝送キューを使用した直前のチャネルが送信側でメッセージのバッチを未確定なままで終了した場合、別のチャネルが伝送キューを使用することはできません。 詳しくは、231 ページの『未確定チャネルの処理』 を参照してください。

チャネルの開始

チャネルは、4 つの方法のいずれかでメッセージの送信することができます。 次の方法にすることができます:

- オペレーターによって (受信側チャネル、クラスター受信側チャネル、サーバー接続チャネル以外)。
- 伝送キューから起動。 この方式は、送信側チャネルおよび完全修飾サーバー・チャネル (CONNAME を指定するチャネル) の場合にのみ 使用できます。 チャネルをトリガーするために、 必要なオブジェクトを準備する必要があります。
- アプリケーション・プログラムからの開始 (受信側チャネル、クラスター受信側チャネル、またはサーバー接続チャネル以外)。
- 送信側、クラスター送信側、要求側、サーバー、またはクライアント接続チャネルによるネットワークからのリモート開始。 受信側チャネル、クラスター受信側チャネル、および場合によってはサーバー・チャネルと要求側チャネルによる伝送は、この方法で開始できます。サーバー接続チャネルもこの方法で開始できます。 チャネル自体はすでに開始済み (つまり使用可能) になっていなければなりません。

注 : チャネルが 「開始済み」であっても、メッセージを伝送しているとは限りません。 前述の 4 つのイベントのいずれかが発生して、 伝送の開始が「使用可能になっている」ことも考えられます。 チャネルは、START および STOP オペレーター・コマンドを使用して使用可能または使用不可にします。

チャネルの状態

チャネルは、いつでも多くの状態のうちの 1 つであることができます。 いくつかの状態には、副状態もあります。 チャネルは、指定された状態から他の状態に移動することができます。

222 ページの図 20 は、考えられるすべてのチャネル状態と各チャネル状態に対応する副状態の階層を示しています。

223 ページの図 21 は、チャネル状態間のリンクを示しています。 これらのリンクは、すべてのタイプのメッセージ・チャネルおよびサーバー接続チャネルに適用されます。

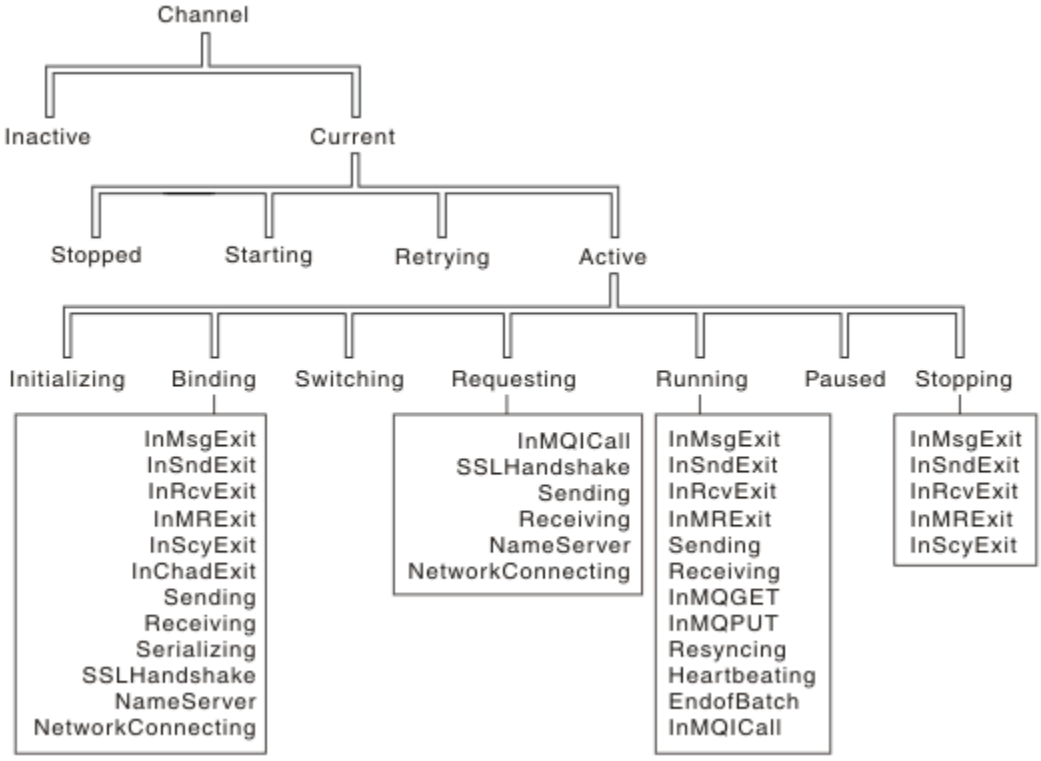


図 *20.* チャネル状態および副状態

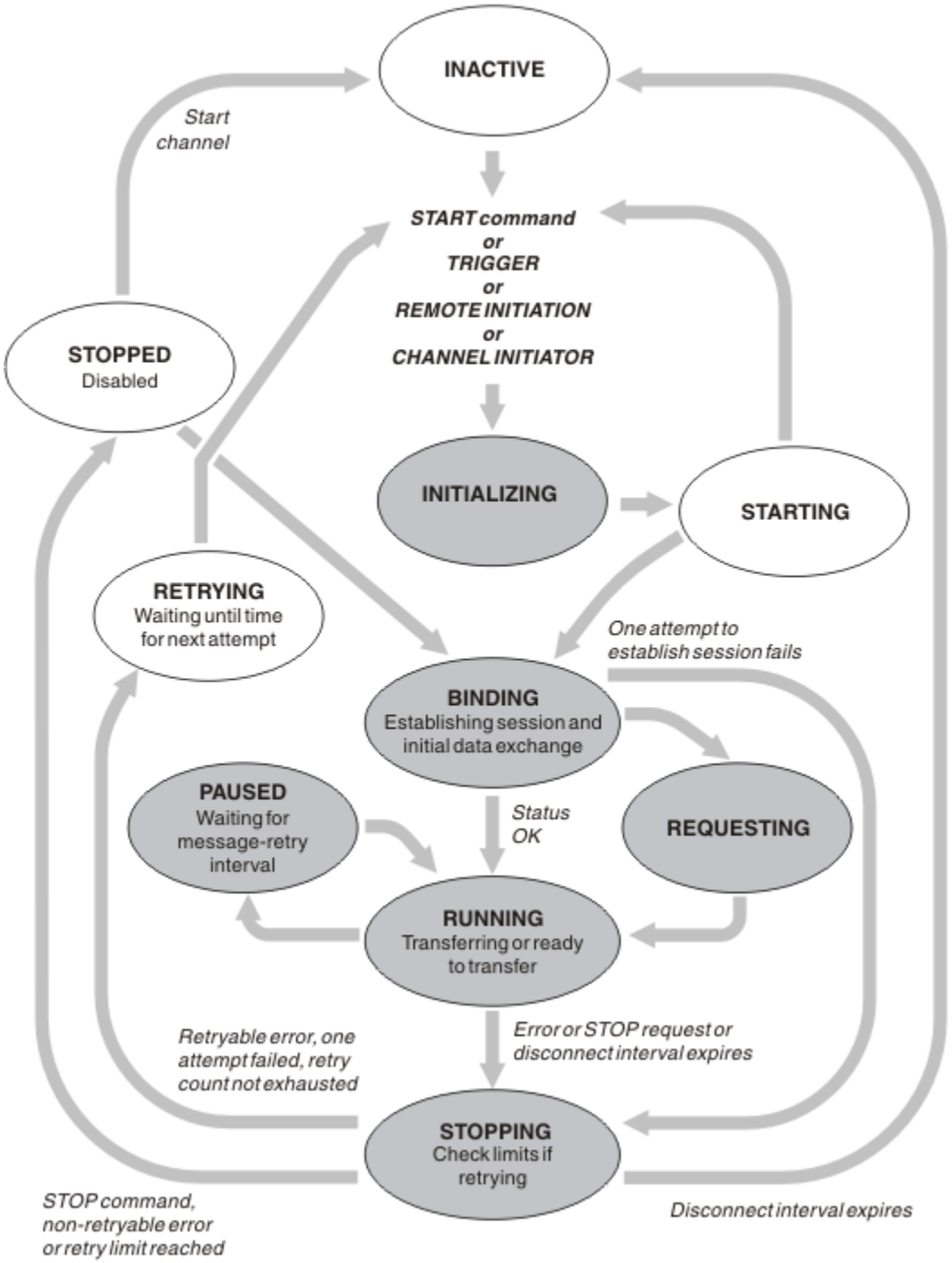


図 *21.* チャネル状態間のフロー

現行およびアクティブ

チャネルは、非アクティブ以外の状態であれば、現行 です。 現行チャネルは、RETRYING、STOPPED、または STARTING 状態でない限り、アクティブです。 チャネルはアクティブであるとき、リソースを消費しており、プロセスまたはスレッドが実行されています。 223 ページの図 21 では、アクティブ・チャネルの 7 つの可能な状態 (INITIALIZING、BINDING、SWITCHING、REQUESTING、RUNNING、PAUSED、または STOPPING) が強調表示されています。

アクティブ・チャネルは、そのチャネルが今どの処理を実行しているかを詳細に表す副状態も示すことができます。 各状態の副状態は、222 ページの図 20 に示されています。

現行およびアクティブ
チャネルは、非アクティブ以外の状態であれば、"現行"です。 現行チャネルは、RETRYING、STOPPED、または STARTING 状態でない限り、"アクティブ"です。

チャネルが「アクティブ」の状態であれば、そのチャネルが今どの処理を実行しているかを詳細に表す副状態も示されます。

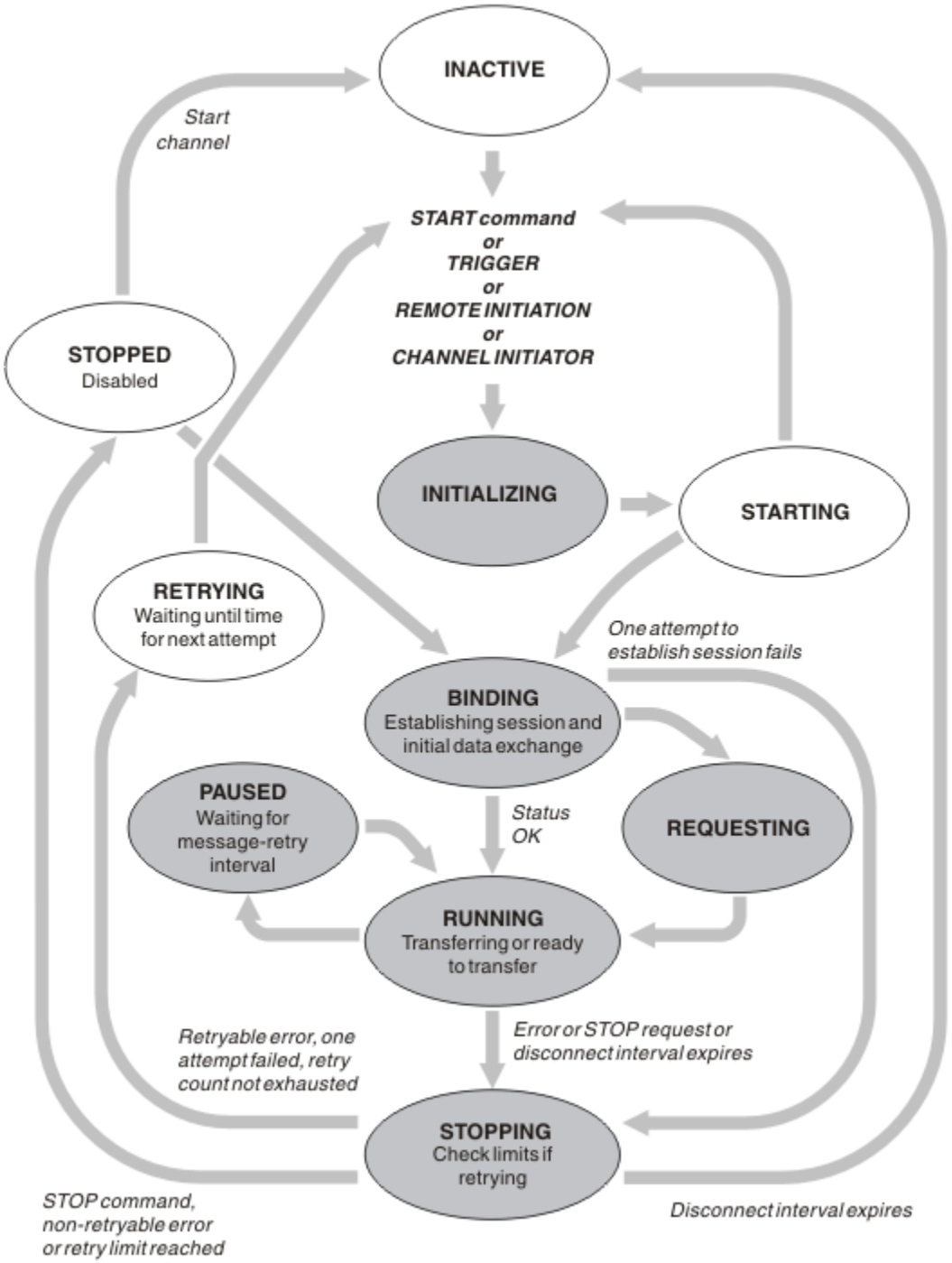


図 22. チャネル状態間のフロー

注 :

1. チャネルは、224 ページの図 22 で強調表示されている 6 種類の状態 (初期化、バインディング、要求、実行、休止、または停止) のいずれかのときは、リソースを 消費しており、プロセスまたはスレッドが実行されています。すなわち、チャネルはアクティブ です。

2. チャネル停止状態のとき、次の状態が何か分からないため、セッションがアクティブである場合があります。

現行チャネルの最大数の指定

一度に現行チャネルにしてもよいチャネルの最大数を指定することができます。 この数は、チャネル状況テーブルに項目があるチャネルの数で、これには再試行中のチャネルと停止中のチャネルが含まれます。 ご使用のプラットフォームに合わせて指定してください。

- z/OS ALTER QMGR MAXCHL コマンドを使用します。
- IBM i キュー・マネージャー初期設定ファイルを編集します。
- Linux AIX キュー・マネージャー構成ファイルを編集します。
- IBM MQ Explorer を使用する。

初期設定ファイルまたは構成ファイルを使用して設定する値について詳しくは、分散キューイング用の構成ファイル・スタンザを参照してください。 チャネルの最大数を指定する方法の詳細については、以下のトピックを参照してください。

- ALW IBM MQ の管理。
- IBM i IBM MQ for IBM i の管理。
- z/OS IBM MQ for z/OS の管理。

注 :

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

アクティブのチャネルの最大数の指定

アクティブ・チャネルの最大数を指定して、多くの開始チャネルによってシステムの負荷が過大になるのを防ぐことができます。 この方法を使用する場合は、切断時間間隔属性を低い値に設定して、他のチャネルが終了したらすぐに待機中のチャネルを開始できるようにします。

再試行するチャネルがパートナーとの接続を確立しようとするたびに、アクティブのチャネルになる必要があります。 試行が失敗した場合、次の試行が再開されるまで、非アクティブのチャネルは現行のままになります。 チャネルを再試行する回数および頻度は、再試行カウント属性および再試行間隔チャネル属性によって決まります。 これらの属性には両方とも長い値と短い値があります。 詳細については、チャネル属性を参照してください。

チャネルがアクティブのチャネルになる必要があるが (START コマンドが出されたか、または起動されたか、または別の再試行が試みられたため)、アクティブのチャネル数がすでに最大数に達しているためにアクティブにできない場合、アクティブのスロットの 1 つが、アクティブになるのをやめた別のチャネル・インスタンスによって解放されるまで、そのチャネルは待機します。 ただし、リモートで開始されたためにチャネルが始動し、その時点で使用できるアクティブのスロットがない場合は、リモート開始は拒否されます。

要求側チャネル以外のチャネルがアクティブになる直前には必ず、開始状態になります。 アクティブなスロットをすぐに利用できる場合でもこの状態が発生しますが、開始状態になるのは短時間のみです。 ただし、チャネルがアクティブのスロットを待っている必要がある場合は、その待機中に開始状態になります。

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

要求側チャネル以外のチャネルがアクティブのスロットを確保できないため、待機している場合はいつでも、メッセージがログ z/OS または z/OS コンソールに書き込まれ、イベントが生成されます。 その後、スロットが解放され、チャネルがスロットを獲得できる場合、別のメッセージおよびイベントが生

成されます。 チャネルがスロットをじかに獲得できる場合は、これらのイベントおよびメッセージはどちらも生成されません。

チャネルがアクティブになるのを待っている間に STOP CHANNEL コマンドが出された場合、そのチャネルは停止状態になります。 チャネル停止イベントが発生します。

アクティブ・チャネルの最大数にサーバー接続チャネルが含まれています。

アクティブ・チャネルの最大数を指定する方法の詳細については、以下のトピックを参照してください。

- ALW IBM MQ の管理。
- IBM i IBM MQ for IBM i の管理。
- z/OS IBM MQ for z/OS の管理。

チャネル・エラー
チャネルでエラーが発生すると、それ以降の伝送が停止することになります。 チャネルが送信側またはサーバーの場合、問題が自然に解決する場合もあるので、チャネルは再試行状態に入ります。 再試行状態にできない場合、チャネルは停止状態になります。

送信側チャネルの場合は、関連する伝送キューは GET(DISABLED) に設定され、トリガー操作はオフにされます。 (STATUS(STOPPED) を指定した STOP コマンドによってもチャネルは停止状態になります。切断時間間隔の満了時に、または STATUS(INACTIVE) を指定した STOP コマンドによってのみ、チャネルは正常に終了し、非アクティブになります。) STOPPED 状態のチャネルは、再始動する前にオペレーターの介入が必要になります (231 ページの『停止したチャネルの再始動』を参照してください)。

注 : IBM i IBM i、 AIX, Linux, and Windows システムの場合、再試行を行うにはチャネル・イニシエーターを実行しなければなりません。 チャネル・イニシエーターが利用不能の場合は、チャネルは非アクティブになるので手動で再始動する必要があります。 スクリプトを使用してチャネルを開始する場合は、スクリプトを実行する前にチャネル・イニシエーターを実行するようにしてください。

長期再試行カウント (LONGRTY) は、再試行がどのように機能するかを説明しています。 エラーが解消されると、自動的にチャネルが再始動されて、伝送キューが再び使用可能になります。 エラーが解消されずに再試行限界に達すると、チャネルは停止状態になります。 停止状態のチャネルは、オペレーターが手動で再始動する必要があります。 エラーが持続している場合は、もう一度再始動を行うことはありません。 正常に開始されると、再び伝送キューが使用可能になります。

z/OS チャネルが RETRYING または STOPPED の状況の間にチャネル・イニシエーターが停止した場合、チャネル・イニシエーターの再開始時にチャネル状況が記憶されています。 ただし、SVRCONN チャネル・タイプのチャネル状況は、チャネルが STOPPED の状況の間にチャネル・イニシエーターが停止すると再設定されます。

Multi チャネルが RETRYING または STOPPED の状況の間にキュー・マネージャーが停止した場合、キュー・マネージャーの再開始時にチャネル状況が記憶されています。 IBM MQ 8.0 以降、これは SVRCONN チャネルにも当てはまります。 以前は、SVRCONN チャネル・タイプのチャネル状況は、チャネルが STOPPED の状況の間にチャネル・イニシエーターが停止するとリセットされていました。

キューが満杯または使用禁止のためにチャネルがターゲット・キューにメッセージを書き込めない場合、チャネルはある時間間隔 (メッセージ再試行間隔属性に指定された時間) で何度も (メッセージ再試行カウント属性に指定された回数) 操作を再試行します。 また、メッセージ再試行出口を作成して、どのような状態のときに何回再試行を行うかを決めることもできます。 チャネルは、メッセージ再試行間隔が終わるのを待つ間、休止状態になります。

チャネル属性についてはチャネル属性を、メッセージ再試行出口についてはメッセージ・チャネル用のチャネル出口プログラムを参照してください。

サーバー接続チャネルの制限

サーバー接続チャネル制限を設定して、クライアント・アプリケーションが **MAXINST** パラメーターを使用してキュー・マネージャー・チャネル・リソースを使い果たさないようにしたり、単一のクライアント・アプリケーションが **MAXINSTC** パラメーターを使用してサーバー接続チャネルの容量を使い果たさないようにしたりすることができます。

DEFINE CHANNEL コマンドを使用して、 **MAXINST** および **MAXINSTC** を設定します。

最大合計数のチャネルは、個々のキュー・マネージャーでいつでもアクティブにできます。 サーバー接続チャネル・インスタンスの合計数は、アクティブ・チャネルの最大数に含まれます。

サーバー接続チャネルの同時インスタンスを最大でいくつ開始できるかを指定しないと、単一のサーバー接続チャネルに接続する単一のクライアント・アプリケーションが、使用可能なアクティブ・チャネルの最大数を使い果たす可能性があります。 アクティブ・チャネルの最大数に達すると、キュー・マネージャー上で他のチャネルは開始できなくなります。 この状態を避けるために、どのクライアントが開始したかに関係なく、個々のサーバー接続チャネルの開始可能な同時インスタンスの数を制限する必要があります。

制限の値が、現在実行中のサーバー接続チャネルのインスタンス数より少ない数まで引き下げられる場合でも (たとえ 0 でも)、実行中のチャネルは影響を受けません。 十分な数の既存のインスタンスが実行を終了して、現在実行中のインスタンスの数が制限の値を下回らないと、新規インスタンスは開始できません。

多くの異なるクライアント接続チャネルが、個々のサーバー接続チャネルに接続することもできます。 どのクライアントが開始したかに関係なく、個々のサーバー接続チャネルの開始可能な同時インスタンス数の制限により、いずれのクライアントもキュー・マネージャーの最大アクティブ・チャネル容量を使い果たせなくなります。 個々のクライアントから開始可能な個々のサーバー接続チャネルの同時インスタンス数も制限しておかないと、欠陥のある単一のクライアント・アプリケーションが多くの接続をオープンして、個々のサーバー接続チャネルに割り振られているチャネル容量を使い果たしてしまうことがあります。したがって、チャネルを使用する必要がある他のクライアントはそれに接続できなくなります。 この状態を避けるために、個々のクライアントから開始可能な、個々のサーバー接続チャネルの同時インスタンスの数を制限する必要があります。

個々のクライアントの制限値が、個々のクライアントから現在実行中のサーバー接続チャネルのインスタンス数より少ない数まで引き下げられる場合でも (たとえ 0 でも)、実行中のチャネルは影響を受けません。しかし、クライアントが開始した十分な数の既存のインスタンスが実行を終了して、現在実行中のインスタンスの数がこのパラメーターの値を下回るまで、新しい制限を超過する個々のクライアントからサーバー接続チャネルの新規インスタンスを開始することはできません。

関連資料
チャネル属性とチャネル・タイプ
DEFINE CHANNEL

チャネルの相手側がまだ使用可能であるかどうかの検査
チャネルの相手側が使用可能であることを検査するために、ハートビート間隔、キープアライブ間隔、およびタイムアウトの受信を使用できます。

ハートビート

ハートビート間隔チャネル属性を使用して、ハートビート間隔 (HBINT) の説明のように、伝送キューにメッセージがない場合にそのフローが送信 MCA から渡されるように指定できます。

キープアライブ

z/OS z/OS では、トランスポート・プロトコルとして TCP/IP を使用する場合、 **Keepalive** インターバル・チャネル属性 (**KAINT**) の値を指定することもできます。 **Keepalive** 間隔には、ハートビート間隔より大きい値と、切断値より小さい値を指定することをお勧めします。 この属性を使用して、キープアライブ間隔 (KAINT) の説明のように、各チャネルのタイムアウト値を指定できます。

Multi IBM i、 AIX, Linux, and Windows システムでは、トランスポート・プロトコルとして TCP を使用する場合、 keepalive=yes を設定できます。 このオプションを指定すると、TCP は、接続の相手側がまだ使用可能であるかどうかを定期的に検査します。 使用可能でない場合、チャネルは終了されます。このオプションについては、キープアライブ間隔 (KAINT)を参照してください。

TCP エラーを報告する信頼性の低いチャネルがある場合は、**Keepalive** オプションを使用するとチャネルがリカバリーしやすくなります。

時間間隔を指定して、**Keepalive** オプションの動作を制御することができます。 時間間隔を変更するとき、変更後に開始される TCP/IP チャネルだけが影響を受けます。 時間間隔として 選択する値は、チャネルの切断間隔の値よりも小さい値にしなければなりません。

Keepalive オプションの使用について詳しくは、 **DEFINE CHANNEL** コマンドの **KAINT** パラメーターを参照してください。

受信タイムアウト

トランスポート・プロトコルとして TCP を使用している場合、 一定の期間データを受信しないと、使用されていない非 MQI チャネル接続の受信側もクローズされます。 この期間 (受信タイムアウト 値) は、**HBINT** (ハートビート間隔) 値に従って決定されます。

IBM MQ for IBM i、AIX, Linux, and Windows システムでは、 *receive time-out* 値は次のように設定されます。

1. 初期フロー数の場合、ネゴシエーションが行われる前の *receive time-out* 値は、チャネル定義の **HBINT** 値の 2 倍です。
2. チャネルが **HBINT** 値をネゴシエーションした後、 **HBINT** が 60 秒未満に設定されている場合、 *receive time-out* 値はこの値の 2 倍に設定されます。 **HBINT** が 60 秒以上に設定されている場合、 *receive time-out* 値は、HBINT の値より 60 秒大きい値に設定されます。

z/OS z/OS では、 *receive time-out* 値は以下のように設定されます。

1. 初期フロー数の場合、ネゴシエーションが行われる前の *receive time-out* 値は、チャネル定義の **HBINT** 値の 2 倍です。
2. **RCVTIME** が設定されている場合、タイムアウトは、 **RCVTTYPE** パラメーターに応じて以下のいずれかの値に設定され、適用される場合は **RCVTMIN** によって課される制限に従います。
   - 折衝された HBINT に定数を掛けた値
   - 折衝された HBINT に定数 (秒数) を加えた値
   - 定数 (秒数)

   RCVTTYPE(EQUAL) が構成されている場合、**RCVTMIN** は適用されません。 定数値 **RCVTIME** を使用し、ハートビート間隔を使用する場合は、ハートビート間隔より小さい **RCVTIME** を指定しないでください。 **RCVTIME** 属性、 **RCVTMIN** 属性、および **RCVTTYPE** 属性について詳しくは、 **ALTER QMGR** コマンドを参照してください。

注 :

1. これらの値のいずれかがゼロの場合、タイムアウトは存在しません。
2. ハートビートをサポートしない接続の場合、 **HBINT** 値はステップ 2 でゼロにネゴシエーションされるため、タイムアウトはないため、 TCP/IP KEEPALIVE を使用する必要があります。
3. 会話の共有を使用するクライアント接続では、MQGET が未解決のときだけでなく、常に (両端から) ハートビートがチャネルを流れることができます。
4. 会話の共有が使用されていないクライアント接続では、クライアントが待機条件付きの MQGET 呼び出しを発行した場合のみ、ハートビートがサーバーから流れます。 この理由から、クライアント・チャネルではハートビート間隔を極端に短く設定しないことをお勧めします。 例えば、ハートビートが 10 秒に設定されている場合に、 コミット処理にかかった時間が 20 秒を超えていると、 この処理中には何もデータが流れていないので MQCMIT 呼び出しは失敗します (MQRC_CONNECTION_BROKEN)。 このエラーは、使用している作業単位が大きいほど、よく発生します。 ただし、ハートビート間隔に対して適切な値が選択されている場合、 極端に長い時間がかかるのは、待機条件を指定された MQGET しかないため、このエラーが発生することはありません。

   SHARECNV がゼロでない場合、クライアントは全二重接続を使用します。これは、クライアントがすべての MQI 呼び出し中にハートビートを実行できる (および実行する) ことを意味します。
5. ハートビート間隔の 2 倍の時間が経過した後で 接続を打ち切ることも有効です。これは、データまたはハートビート・フローが、少なくとも各ハートビート間隔おきに発生するためです。 ハートビート間隔

を小さく設定しすぎると、特にチャネル出口を使用している場合に問題が生じる可能性があります。 例えば、 **HBINT** 値が 1 秒で、送信出口または受信出口が使用されている場合、受信側はチャネルを取り消す前に 2 秒間だけ待機します。 MCA がメッセージの暗号化などの作業を実行している場合は、この値は小さすぎる可能性があります。

推奨設定

z/OS IBM MQ for z/OS

初期の開始点として、次を使用できます。

```
/cpf ALTER QMGR TCPKEEP(YES) RCVTTYPE(ADD) RCVTIME(60) ADOPTMCA(ALL) ADOPTCHK(ALL)
```

ここで、cpf はキュー・マネージャー・サブシステムのコマンド接頭部です。

各種パラメーターについて詳しくは、「**ALTER QMGR**」および「IBM MQ ネットワークの可用性」を参照してください。

送信側の IP アドレスが複数のアドレスに変換される可能性がある場合は、 **ADOPTCHK** を ALL ではなく QMNAME に設定する必要があります。

Multi IBM MQ for Multiplatforms

qm.ini で、以下の情報を追加します:

```
TCP:
KeepAlive=Yes
CHANNELS:
AdoptNewMCA=ALL
AdoptNewMCACheck=ALL
```

詳しくは、 **ALTER QMGR**、分散キューイング用の構成ファイル・スタンザ、および 117 ページの『qm.ini ファイルの Channels スタンザ』 を参照してください。

送信側の IP アドレスが複数のアドレスに変換される可能性がある場合は、**AdoptNewMCACheck** を **ALL** ではなく QMNAME に設定する必要があります。

MCA の採用

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

チャネルで通信が失われると、受信側チャネルは「通信受信」状態のままになる可能性があります。 通信が再確立されたら、送信側チャネルは再接続を試みます。 受信側チャネルがすでに実行されていることがリモート・キュー・マネージャーで検出されると、同じ受信側チャネルの別のバージョンを開始することはできません。 この問題が生じたら、ユーザーが介入してそれを解決するか、またはシステム・キープアライブを使用する必要があります。

Adopt MCA 機能は、この問題を自動的に解決します。 IBM MQ でこの機能を使用して、受信側チャネルを取り消してから、代わりの別のチャネルを起動することができます。

関連タスク

IBM MQ の管理

IBM MQ for z/OS の管理

IBM MQ for IBM i の管理

チャネルの停止および静止

切断時間間隔が満了する前に、チャネルを停止および静止することができます。

メッセージ・チャネルは、キュー・マネージャー間の接続を長時間保持するように設計されており、正規の終了が切断時間間隔チャネル属性だけによって制御されるようになっています。 このメカニズムは、切断時間間隔が満了する前にオペレーターがチャネルを終了する必要がない限り、効率的に働きます。 このような必要が生じるのは、以下の場合です。

- システムの静止
- リソースの保存
- チャネルの片側からの一方的な処置

このような場合、チャネルを停止することができます。 以下の方法を使用します。

- STOP CHANNEL MQSC コマンド
- STOP CHANNEL PCF コマンド
- IBM MQ エクスプローラー
- z/OS IBM i プラットフォーム固有のその他のメカニズムは以下のとおりです。

  z/OS **z/OS の場合:**
  「Stop a Channel (チャネルの停止)」パネル

  IBM i **IBM i の場合:**
  ENDMQMCHL CL コマンドまたは WRKMQMCHL パネルの END オプション

これらのコマンドを使用してチャネルを停止する場合は、以下の 3 つのオプションを指定できます。

QUIESCE
QUIESCE オプションを指定すると、チャネルを停止する前にメッセージの現行バッチの終了が試みられます。

FORCE
FORCE オプションを指定すると、即時にチャネルの停止が試みられますが、再始動時には、チャネルが未確定のままである可能性があるため、再同期化が必要になることがあります。

z/OS IBM MQ for z/OS では、FORCE は、進行中のメッセージ再割り振りをすべて中断します。これにより、BIND_NOT_FIXED メッセージの再割り振りが不完全になったり、正しくない順序になったりする場合があります。

TERMINATE
TERMINATE オプションを指定すると、即時にチャネルの停止が試みられ、 チャネルのスレッドまたは処理が終了します。

z/OS IBM MQ for z/OS では、TERMINATE は、進行中のメッセージ再割り振りをすべて中断します。これにより、BIND_NOT_FIXED メッセージの再割り振りが不完全になったり、正しくない順序になったりする場合があります。

これらのすべてのオプションは、 チャネルを停止状態のままにしておくので、オペレーターの介入によって再始動する必要があります。

送信側でチャネルを停止するのは効果的ですが、 再始動には、オペレーターの介入が必要です。 チャネルの受信側では、MCA は送信側からのデータを待機しており、受信側からチャネルの正規の 終了を開始する方法はないので、受信側でチャネルを終了させるのは困難です。停止コマンドは、MCA がデータ待ちから解放されるまで保留されます。

必要とされる操作の特性によって、次の 3 つのチャネル使用法を推奨します。

- チャネルを長期間実行させたい場合は、 正規の方法でチャネルを終了できるのは、送信側からだけであることに注意してください。 チャネルに割り込みが行われる (停止される) と、そのチャネルの再始動にはオペレーターの介入 (START CHANNEL コマンド) が必要です。
- 送信するメッセージがあるときだけチャネルをアクティブにする場合は、 切断間隔を非常に小さな値に設定してください。 デフォルト設定は高い値であるため、 このようなレベルの制御が必要なチャネルにはお勧めできません。 受信側チャネルに割り込みを行うのは困難なため、最も効率的なオプションはワークロードの需要に応じて自動的にチャネルの切断および再接続を行うことです。 ほとんどのチャネルでは、切断時間間隔に適切な値を見つけて設定することができます。
- ハートビート間隔属性を使用して、送信するメッセージがない期間に送信側 MCA から受信側 MCA へハートビート・フローを送信させることができます。 このアクションによって、受信側 MCA は待機状態から解放され、切断時間間隔の満了を待たずにチャネルを静止させる機会が与えられます。 ハートビート間隔は、切断時間間隔値よりも小さい値にします。

注:

1. サーバー・チャネルには特に小さい切断時間間隔値を設定するかまたはハートビートを使用することをお勧めします。 小さい値を使用するのは、サーバー・チャネルから送信すべきメッセージがないときに、 (例えば、チャネルが取り消されて) 要求側のチャネルが異常終了するような場合に備えるためです。 切断間隔の値が高く設定され、 ハートビートが使用されていない場合、 サーバーでは、要求側の終了は検出されません (検出されるのは、次に要求側にメッセージを送信しようとした場合のみです)。 サーバーがまだ実行中の間、さらに伝送キューに入ってくるメッセージを入手するために、サーバーは、入力専用としてそのキューをオープンしたままにしておきます。 要求側からチャネルを再始動しようとすると、サーバーが入力専用としてまだ伝送キューをオープンにしているので、開始要求はエラーを受け取ります。 サーバー・チャネルを停止して、要求側からサーバー・チャネルを再始動する必要があります。

停止したチャネルの再始動

チャネルが STOPPED 状態になるときは、手動でチャネルを再始動しなければなりません。

このタスクについて

送信側またはサーバー・チャネルでは、チャネルが停止状態になった場合、関連する伝送キューは GET(DISABLED) に設定され、トリガー操作はオフに設定されます。 開始要求が受信されたとき、これらの属性は自動的にリセットされます。

z/OS チャネルが RETRYING または STOPPED の状況の間にチャネル・イニシエーターが停止した場合、チャネル・イニシエーターの再開始時にチャネル状況が記憶されています。 ただし、SVRCONN チャネル・タイプのチャネル状況は、チャネルが STOPPED の状況の間にチャネル・イニシエーターが停止すると再設定されます。

Multi チャネルが RETRYING または STOPPED の状況の間にキュー・マネージャーが停止した場合、キュー・マネージャーの再開始時にチャネル状況が記憶されています。 IBM MQ 8.0 以降、これは SVRCONN チャネルにも当てはまります。 以前は、SVRCONN チャネル・タイプのチャネル状況は、チャネルが STOPPED の状況の間にチャネル・イニシエーターが停止するとリセットされていました。

手順

- 以下のいずれかの方法で、チャネルを再始動します。
  - START CHANNEL MQSC コマンドを使用します。
  - Start Channel PCF コマンドを使用します。
  - IBM MQ Explorer を使用します。
  - z/OS z/OS で、「チャネルの開始 (Start a channel)」パネルを使用します。
  - IBM i IBM i で、STRMQMCHL CL コマンドまたは WRKMQMCHL パネルの START オプションを使用します。

未確定チャネルの処理

未確定チャネルとは、メッセージの送受信の対象となっているリモート・チャネルが未確定のチャネルのことです。

このタスクについて

これと、キューに対するメッセージ・コミットの対象になるキュー・マネージャーが未確定であることとの相違点に注意してください。

バッチ・ハートビート・チャネル・パラメーター (**BATCHHB**) を使用すると、チャネルを未確定にする機会を減らすことができます。 このパラメーターの値を指定すると、送信側チャネルは、リモート・チャネルがまだアクティブであることを確認してから、追加のアクションを実行します。 応答が受信されない場合は、受信側チャネルはアクティブではないと見なされます。 メッセージのロールバックおよび再経路指定が可能なので、送信側チャネルは未確定になりません。 このため、チャネルが未確定になるのは、受信側

チャネルがアクティブになっていることを送信側チャネルが確認する期間と、送信されたメッセージを受信側チャネルが受信したことを送信側チャネルが確認する期間だけとなります。 バッチ・ハートビート・パラメーターの詳細については、チャネル属性を参照してください。

未確定チャネルの問題は、通常は自動的に解決されます。 通信が中断され、受信状態が不明のメッセージ・バッチが送信側に存在する状況でチャネルが未確定になる場合でも、その通信が再度確立されたときに、この状態は解決されます。 そのために、シーケンス番号と LUWID レコードが保持されています。 LUWID 情報が交換されるまでチャネルは未確定になります。また、未確定になるメッセージ・バッチはチャネルごとに 1 つだけです。

必要な場合は、チャネルを手動で再同期化できます。 マニュアルという用語には、 IBM MQ システム管理コマンドを含むオペレーターまたはプログラムの使用が含まれます。 手動による再同期プロセスは次のように行われます。 ここでは MQSC コマンドを使用しますが、PCF の同等コマンドを使用することもできます。

手順

1. **DISPLAY CHSTATUS** コマンドを使用して、チャネルの各サイドの最後にコミットされた作業論理単位 ID (LUWID) を見つけます。
   次のコマンドを使用してこれを行います。
   - 未確定側のチャネルの場合

     ```
     DISPLAY CHSTATUS(name) SAVED CURLUWID
     ```

     CONNAME パラメーターと **XMITQ** パラメーターを使用して、チャネルをさらに詳しく識別できます。
   - 受信側チャネルの場合

     ```
     DISPLAY CHSTATUS( name ) SAVED LSTLUWID
     ```

     CONNAME パラメーターを使用して、チャネルをさらに詳しく識別できます。

   注 : 未確定の可能性があるのはチャネルの送信側だけであるため、コマンドは異なります。 受信側が未確定になることはありません。

   IBM i IBM i では、 **DISPLAY CHSTATUS** コマンドは、 **STRMQMMQSC** コマンドまたは「MQM チャネル状況の処理」CL コマンド (**WRKMQMCHST**) を使用してファイルから実行できます。
2. 2 つの LUWID が同じ場合は、 **RESOLVE CHANNEL** コマンドを使用して未確定メッセージをコミットします。
   2 つの LUWID が同じである場合、受信側は送信側が未確定であると見なす作業単位をすでにコミットしています。 したがって、送信側は伝送キューから未確定のメッセージを除去して、それを再度使用可能にすることができます。 これは、以下の **RESOLVE CHANNEL** コマンドを使用して行います。

   ```
   RESOLVE CHANNEL(name) ACTION(COMMIT)
   ```
3. 2 つの LUWID が異なる場合は、 **RESOLVE CHANNEL** コマンドを使用して未確定メッセージをバックアウトします。
   2 つの LUWID が異なっている場合、受信側は送信側が未確定であると見なす作業単位をコミットしていません。 送信側では、伝送キューに未確定のメッセージを保存しておき、それらを再送信する必要があります。 これは、以下の **RESOLVE CHANNEL** コマンドを使用して行います。

   ```
   RESOLVE CHANNEL( name ) ACTION(BACKOUT)
   ```

   IBM i IBM i では、「MQM チャネルの解決」コマンド **RSVMQMCHL** を使用できます。

タスクの結果

以上の処理で、チャネルは未確定ではなくなります。 必要であれば伝送キューを別のチャネルで使用できます。

関連資料
DISPLAY CHSTATUS (チャネル状況の表示)
RESOLVE CHANNEL (未確定メッセージを解決するためのチャネルへの要求)

メッセージの安全性

IBM MQ の通常のリカバリー機能の他に、分散キュー管理は、メッセージ・チャネルの両側で調整された同期点プロシージャーを使用して、メッセージが適切に送達されることを確認します。 このプロシージャーはエラーを検出すると、問題を調査できるようにチャネルをクローズし、チャネルが再開されるまでメッセージを伝送キューに安全に保管します。

同期点プロシージャーには、チャネルの開始時に未確定状態をリカバリーしようとするという利点が追加されています。(未確定は、同期点が要求されたが、要求の結果がまだ認識されていない、リカバリー単位の状況です。) また、この機能には、次の 2 つの機能が関連付けられています。

1. コミットまたはバックアウトによる解決
2. シーケンス番号のリセット

ほとんどの場合、チャネルは自動的にリカバリーするので、これらの機能は例外状況でのみ使用されます。

高速、非持続メッセージ

非持続メッセージ速度 (NPMSPEED) チャネル属性を使用して、チャネルのすべての非持続メッセージの送達速度を上げるように指定できます。 この属性の詳細については、非持続メッセージ速度 (NPMSPEED) を参照してください。

高速で、非持続メッセージの転送中にチャネルが終了した場合、メッセージは失われることがあります。必要な場合にリカバリーの措置をとることはアプリケーション側の処理範囲です。

受信側チャネルが宛先キュー内にメッセージを入れられない場合、送達不能キューが定義されていれば、メッセージはそのキューに入れられます。 定義されていないと、メッセージは廃棄されます。

注 : チャネルの片側でこのオプションをサポートしていない場合、そのチャネルは通常の速度で実行されます。

未配布メッセージ

メッセージを送達できない場合の処理については、233 ページの『メッセージを送達できない場合の処理』を参照してください。

メッセージを送達できない場合の処理

メッセージを送達できない場合、MCA はメッセージをいくつかの方法で処理できます。 再試行するか、送信側へ返信するか、または送達不能キューへ書き込むことができます。

234 ページの図 23 は、MCA が宛先キューにメッセージを入れることができないときに行われる処理を示したものです。 (図に示したオプションはすべてのプラットフォームには当てはまりません。)

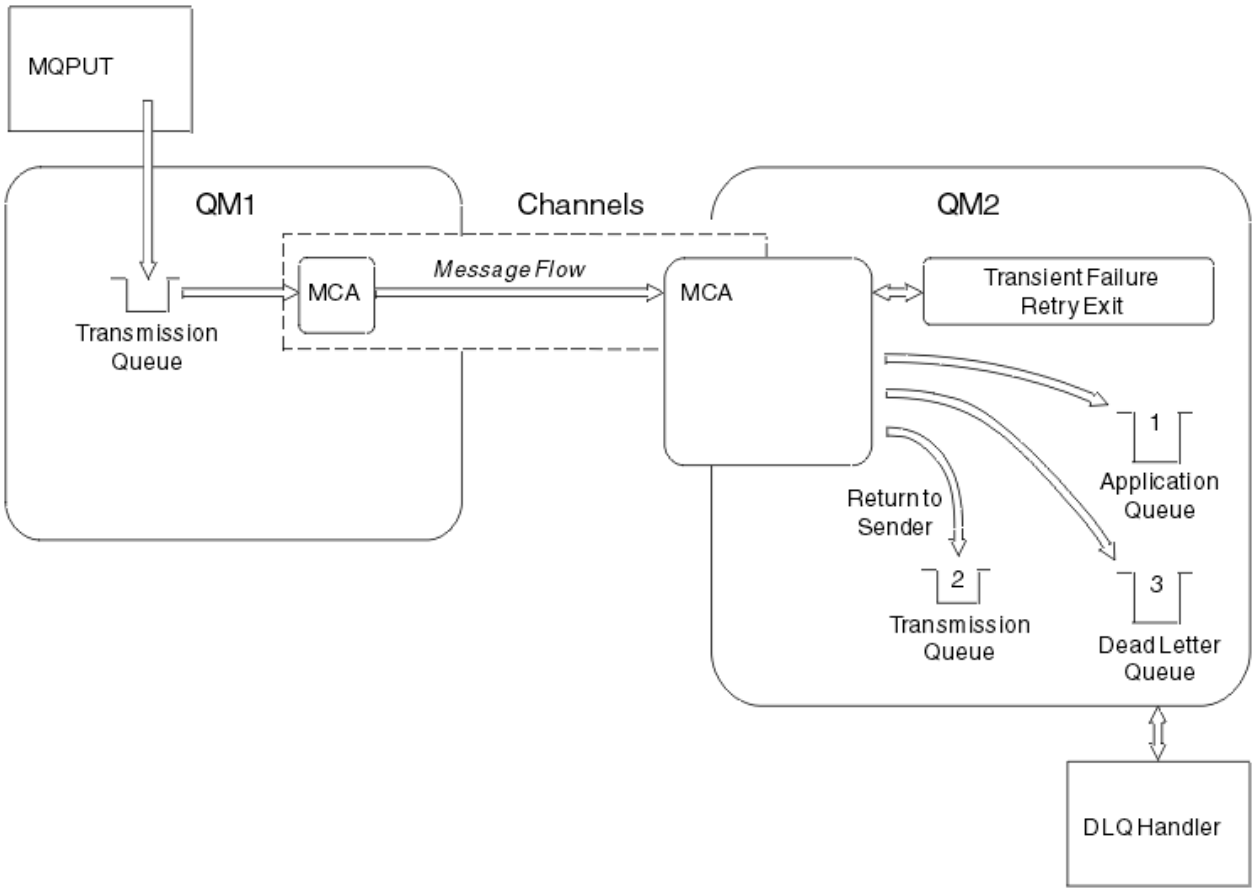


図 *23.* メッセージを送達できない場合の処理

図に示すように、MCA は、送達できないメッセージについていくつかの処置を行うことができます。 行われる処置は、チャネルの定義時に指定したオプションおよびメッセージの MQPUT レポート・オプションで指定したオプションによって決まります。

1. メッセージ再試行

   一時的な理由 (キューが満杯であるなど) で MCA がターゲット・キューにメッセージを書き込むことができない場合、MCA は待機して、あとで操作を再試行できます。 MCA が待機するかどうか、待機時間、および試行の回数を決めることができます。

   - チャネルを定義するときに、MQPUT エラーの場合のメッセージ再試行時間および間隔を指定することができます。 キューが満杯または書き込み禁止のためメッセージを宛先キューに書き込むことができない場合、MCA は、指定された時間間隔で指定された回数だけ操作を試行します。
   - 独自のメッセージ再試行出口を作成することができます。 この出口を使用して、MCA に MQPUT 操作または MQOPEN 操作を再試行させる条件を指定することができます。 チャネルの定義時に出口の名前を指定します。

2. 送信側への返信

   メッセージ再試行が失敗した場合、または別のタイプのエラーが検出された場合、MCA は発信元にメッセージを送り返すことができます。 送信側への返信を可能にするには、メッセージを元のキューに書き込むとき、メッセージ記述子に以下のオプションを指定する必要があります。

   - MQRO_EXCEPTION_WITH_FULL_DATA レポート・オプション
   - MQRO_DISCARD_MSG レポート・オプション
   - 応答先キューおよび応答先キュー・マネージャーの名前

   宛先キューにメッセージを書き込むことができない場合、MCA は、元のメッセージが入った例外レポートを生成し、伝送キューに書き込んで、元のメッセージで指定されている応答先キューに送信されるよ

うにします。 (応答先キューが MCA と同じキュー・マネージャー上にある場合は、メッセージは伝送キューには送られず、その応答先キューに直接書き込まれます。)

3. 送達不能キュー

   送達することも戻すこともできないメッセージは、送達不能キュー (DLQ) に入れられます。 DLQ ハンドラーを使用してメッセージを処理することができます。 この処理については、送達不能キューのメッセージの処理 (IBM MQ for UNIX、Linux、および Windows システムの場合)、および送達不能キュー・ハンドラー・ユーティリティー (CSQUDLQH) (z/OS システムの場合) で説明されています。 送達不能キューが使用できない場合、送信側 MCA は、伝送キューにメッセージを残したまま、チャネルは停止します。 高速チャネルでは、送達不能キューに書き込めない非持続メッセージは失われます。

   IBM WebSphere MQ 7.0 では、ローカル送達不能キューが定義されていない場合、リモート・キューは使用できないか、定義されません。また、リモート送達不能キューがない場合、送信側チャネルは RETRY 状態となり、メッセージは自動的に伝送キューにロールバックされます。

関連資料

送達不能キューの使用 (USEDLQ)

チャネルのトリガー操作

IBM MQ には、キューで特定の条件が満たされると自動的にアプリケーションを開始する機能が用意されています。 この機能は、トリガー操作と呼ばれています。

ここでの説明は、トリガー操作のおおよその概念を示すことを目的としています。 完全な説明については、トリガーによる IBM MQ アプリケーションの開始を参照してください。

プラットフォーム特有の情報については、以下のセクションを参照してください。

- AIX, Linux, and Windows については、236 ページの『AIX, Linux, and Windows でのチャネルのトリガー操作』を参照してください。
- IBM i IBM i については、237 ページの『IBM MQ for IBM i におけるチャネルのトリガー操作』を参照してください。
- z/OS z/OS については、991 ページの『伝送キューおよびチャネルのトリガー操作』を参照してください。

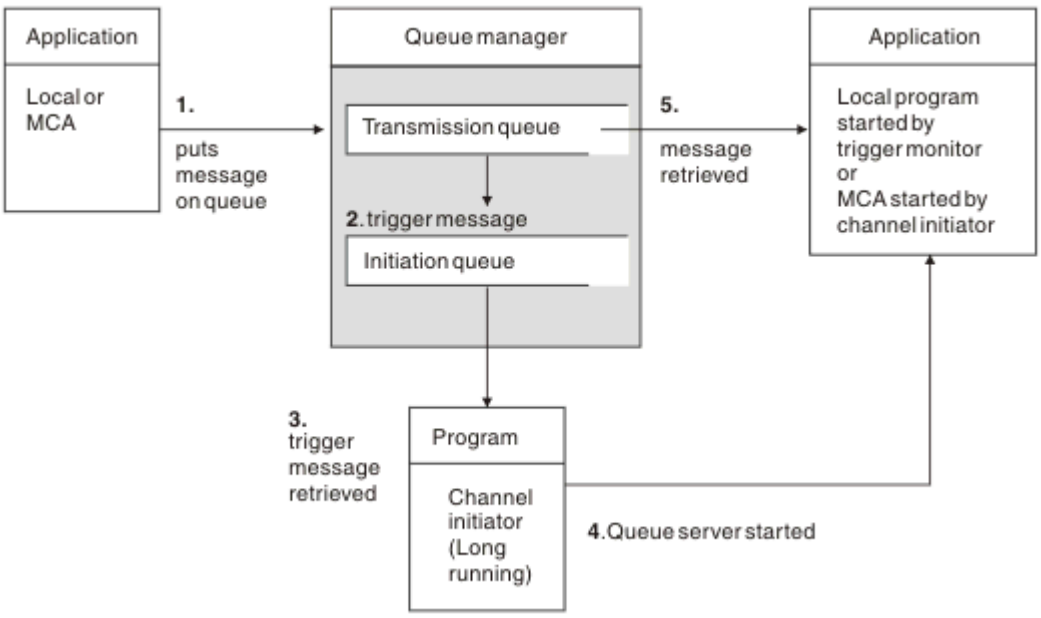


図 *24.* トリガー操作の概念

トリガー操作に必要なオブジェクトが、235 ページの図 24 に示されています。 この図には、以下のイベント順序が示されています。

1. アプリケーションまたはメッセージ・チャネル・エージェント (MCA) からのメッセージがローカル・キュー・マネージャーによって伝送キューに書き込まれます。
2. トリガー操作の条件が満たされると、開始キューにトリガー・メッセージがローカル・キュー・マネージャーによって書き込まれます。
3. 長期実行チャネル・イニシエーター・プログラムは開始キューをモニターし、到達したメッセージを取り出します。
4. チャネル・イニシエーターは、トリガー・メッセージに入れられた情報に従ってそれらを処理します。この情報には、チャネル名も入っていることがあります。その場合、対応する MCA が開始されます。
5. ローカル・アプリケーションまたは MCA は、起動されると、伝送キューからのメッセージを検索します。

このシナリオを設定するには、以下のことを行う必要があります。

- 対応する属性の開始キューの名前 (つまり SYSTEM.CHANNEL.INITQ) でローカル・キューを作成する。
- 開始キュー (つまり、SYSTEM.CHANNEL.INITQ) が存在することを確認する。
- チャネル・イニシエーターが使用可能であって、しかも実行されていることを確認する。 チャネル・イニシエーターには、開始コマンドに入れて開始キューの名前を提供しなければなりません。
  z/OS z/OS では、開始キューの名前は固定であるため、開始コマンドでは使用されません。
- オプションで、トリガー操作用のプロセス定義が存在していない場合はそれを作成し、*UserData* フィールドにサービス対象となるチャネルの名前が入っていることを確認する。 プロセス定義を作成する代わりに、伝送キューの **TriggerData** 属性にチャネル名を指定することもできます。 IBM MQ (IBM i IBM i、 AIX, Linux, and Windows システムの場合) では、チャネル名をブランクとして指定できます。この場合、この伝送キューを持つ使用可能な最初のチャネル定義が使用されます。
- サービス対象となるプロセス定義名 (適用できる場合)、開始キュー名、および最適と思われるトリガー操作特性が伝送キュー定義に入っていることを確認する。 トリガー制御属性は、必要に応じてトリガー操作を有効/無効にします。

注 :

1. チャネル・イニシエーターは、チャネルの開始に使われる開始キューをモニターするための「トリガー・モニター」として働きます。
2. 開始キューおよびトリガー・プロセスを使用して、任意の数のチャネルを起動できます。
3. 開始キューとトリガー・プロセスは、いくつでも定義できます。
4. FIRST トリガー・タイプを使用して、システムで過剰にチャネルが開始されないようにすることをお勧めします。

AIX, Linux, and Windows でのチャネルのトリガー操作

ALW

トリガー操作の対象となるプロセスを定義するプロセス定義を IBM MQ 内に作成します。 MQSC コマンド DEFINE PROCESS を使用して、伝送キューにメッセージが到達したときに起動されるプロセスを指定して、プロセス定義を作成します。 プロセス定義の USERDATA 属性には、その伝送キューでサービスを受けるチャネルの名前が含まれます。

ローカル・キュー (QM4) を定義します。そのためには、トリガー・メッセージを開始キュー (IQ) に書き込んで、アプリケーションにトリガー操作を行うように指定します。それによって、チャネル (QM3.TO.QM4) が開始されます。

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(P1) USAGE(XMITQ)
```

開始するアプリケーション (プロセス P1) を定義します。

```
DEFINE PROCESS(P1) USERDATA(QM3.TO.QM4)
```

また、IBM MQ for UNIX、Linux および Windows システムの場合、伝送キューの TRIGDATA 属性にチャネル名を指定すると、プロセスを定義する必要がなくなります。

ローカル・キュー (QM4) を定義します。 次のように、トリガー・メッセージをデフォルト開始キュー SYSTEM.CHANNEL.INITQ に書き込んで、チャネル (QM3.TO.QM4) を開始させるアプリケーション (プロセス P1) を起動するように指定します。

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ)
USAGE(XMITQ) TRIGDATA(QM3.TO.QM4)
```

チャネル名を指定しないと、チャネル・イニシエーターが、指定された伝送キューと関連付けられたチャネルを発見するまで、チャネル定義ファイルを探します。

IBM MQ for IBM i におけるチャネルのトリガー操作

IBM i

IBM MQ for IBM i におけるチャネルのトリガー操作は、チャネル・イニシエーター・プロセスで実現されます。 開始キュー SYSTEM.CHANNEL.INITQ のチャネル・イニシエーター・プロセスは、キュー・マネージャーで自動的に開始されます (キュー・マネージャーの SCHINIT 属性を変更してそれを使用不可にした場合を除く)。

SYSTEM.CHANNEL.INITQ を開始キューとして指定し、そのキューでのトリガーを使用可能にすることで、チャネルの伝送キューをセットアップします。 チャネル・イニシエーターは、この伝送キューを指定する、使用可能な最初のチャネルを開始します。

```
CRTMQMQ QNAME(MYXMITQ1) QTYPE(*LCL) MQMNAME(MYQMGR)
TRGENBL(*YES) INITQNAME(SYSTEM.CHANNEL.INITQ)
USAGE(*TMQ)
```

Deprecated STRMQMCHLI コマンドで最大 3 つのチャネル・イニシエーター・プロセスを手動で開始でき、異なる開始キューを指定できます。 また、伝送キューを処理できるチャネルを複数指定し、開始するチャネルを選択することもできます。 この機能は、以前のリリースと互換性を持たせるために今でも提供されています。 この使用は推奨されません。

注 : 一度に 1 つのチャネルのみが伝送キューを処理できます。

```
STRMQMCHLI QNAME(MYINITQ)
```

TRGENBL(*YES) を指定して、チャネルの伝送キューをセットアップします。開始しようとするチャネルを選択するには、TRIGDATA フィールドにチャネル名を指定します。 以下に例を示します。

```
CRTMQMQ QNAME(MYXMITQ2) QTYPE(*LCL) MQMNAME(MYQMGR)
TRGENBL(*YES) INITQNAME(MYINITQ)
USAGE(*TMQ) TRIGDATA(MYCHANNEL)
```

関連概念

238 ページの『チャネル・イニシエーターの開始と停止』
トリガー操作は、チャネル・イニシエーターのプロセスによって実行されます。

関連タスク

190 ページの『分散キューイングの構成』
このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

関連資料

AIX, Linux, and Windows でのチャネル・プログラム

IBM i IBM i における相互通信ジョブ

IBM i IBM i におけるチャネルの状態

チャネル・イニシエーターの開始と停止

トリガー操作は、チャネル・イニシエーターのプロセスによって実行されます。

このチャネル・イニシエーターの処理は、MQSC コマンド START CHINIT を使用して開始します。 デフォルトの始動キューを使用していない場合、コマンドの始動キューの名前を指定します。 例えば、START CHINIT コマンドを使用してデフォルト・キュー・マネージャーのキュー IQ を開始するには、以下のように入力します。

```
START CHINIT INITQ(IQ)
```

デフォルトでは、チャネル・イニシエーターはデフォルトの開始キュー SYSTEM.CHANNEL.INITQ を使用して自動的に開始されます。 すべてのチャネル・イニシエーターを手動で開始したい場合は、以下のステップに従ってください:

1. キュー・マネージャーを作成して開始します。
2. キュー・マネージャーの SCHINIT プロパティーを MANUAL に変更します。
3. キュー・マネージャーを終了して再始動します。

IBM MQ for Multiplatforms システムでは、チャネル・イニシエーターが自動的に開始されます。 開始できるチャネル・イニシエーターの数は制限されています。 デフォルト値と最大値は 3 です。 AIX and Linux システムの qm.ini ファイル内の MAXINITIATORS、および Windows システムのレジストリー内で変更できます。

チャネル・イニシエーター実行コマンド **runmqchi**、およびその他の制御コマンドについて詳しくは、IBM MQ 制御コマンド を参照してください。

チャネル・イニシエーターの停止

デフォルトのチャネル・イニシエーターは、キュー・マネージャーを開始すると自動的に開始します。 すべてのチャネル・イニシエーターは、キュー・マネージャーが停止すると自動的に停止します。

初期設定および構成ファイル

チャネル初期設定データの処理は、運用している IBM MQ プラットフォームによって異なります。

IBM MQ for z/OS

z/OS

IBM MQ for z/OS では、 **ALTER QMGR** MQSC コマンドを使用して初期化情報と構成情報を指定します。 **ALTER QMGR** コマンドを CSQINP2 初期設定入力データ・セットに入れると、キュー・マネージャーが開始されるたびにそれらのコマンドが処理されます。

チャネル・イニシエーターを開始するたびに **START LISTENER** などの MQSC コマンドを実行するには、それらのコマンドを CSQINPX 初期設定入力データ・セットに入れ、オプションの DD ステートメント CSQINPX をチャネル・イニシエーター開始タスク・プロシージャーに指定します。

CSQINP2 および CSQINPX について詳しくは、 初期化入力データ・セットのカスタマイズ、および ALTER QMGR を参照してください。

IBM MQ for Multiplatforms

Multi

IBM MQ for Multiplatforms には、 IBM MQ インストールに関する基本構成情報を保持するための構成ファイルがあります。

構成ファイルは 2 種類あり、1 つはマシンに適用されるもので、もう 1 つは個々のキュー・マネージャーに適用されるものです。

IBM MQ 構成ファイル

このファイルには、IBM MQ システムのすべてのキュー・マネージャーに関連する情報が保持されています。 このファイルは mqs.ini と呼ばれます。 86 ページの『IBM MQ 構成ファイル mqs.ini』を参照してください。

キュー・マネージャー構成ファイル

このファイルには、1 つの特別なキュー・マネージャーに関する構成情報が保持されています。 このファイルは qm.ini と呼ばれます。

このファイルは、キュー・マネージャーの作成時に作成され、キュー・マネージャーのどの面に関する構成情報も保持することができます。 ファイルに保管される情報には、ログ構成が IBM MQ 構成ファイルのデフォルト設定とどのように異なるかの詳細情報が入っています。

キュー・マネージャー構成ファイルは、キュー・マネージャーが所有するディレクトリー・ツリーのルート内に入っています。 例えば、**DefaultPath** 属性の場合、QMNAME というキュー・マネージャーのキュー・マネージャー設定ファイルは次のようになります。

AIX and Linux システムの場合:

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

Windows システムの場合:

```
C:\ProgramData\ IBM \ MQ\qmgrs\QMNAME\qm.ini
```

IBM i の場合:

```
/QIBM/UserData/mqm/qmgrs/QMNAME/qm.ini
```

これは、qm.ini からの抽出データです。 これは、TCP/IP リスナーがポート 2500 にあり、現行チャネルの最大数が 200、アクティブのチャネルの最大数が 100 であることを指定します。

```
TCP:
Port=2500
CHANNELS:
MaxChannels=200
MaxActiveChannels=100
```

アウトバウンド・チャネルが使用する TCP/IP ポートの範囲を指定できます。 1 つの方法は、qm.ini ファイルを使用して、ポート値の範囲の開始と終了を指定することです。 以下の例は、チャネルの範囲を指定する qm.ini ファイルを示しています:

```
TCP:
StrPort=2500
EndPort=3000
CHANNELS:
MaxChannels=200
MaxActiveChannels=100
```

StrPort または **EndPort** に値を指定する場合は、両方に値を指定する必要があります。 **EndPort** の値は、常に **StrPort** の値より大きくなければなりません。

チャネルは、指定した範囲の両方のポート値の使用を試みます。 接続が成功すると、ポート値はチャネルが使用するポートになります。

qm.ini ファイルについて詳しくは、 98 ページの『キュー・マネージャー構成ファイル qm.ini』を参照してください。

メッセージのデータ変換

IBM MQ メッセージは、異なるキュー・マネージャーでのキュー間で送信したときには、データ変換を必要とする場合があります。

IBM MQ メッセージは次の 2 つの部分から構成されています。

- メッセージ記述子内の制御情報
- アプリケーション・データ

2 つのいずれの部分についても、異なるキュー・マネージャーでのキュー間で送信したときにはデータ変換を必要とする場合があります。 アプリケーション・データ変換について詳しくは、アプリケーション・データの変換を参照してください。

独自のメッセージ・チャネル・エージェントの作成

IBM MQ を使用すれば、独自のメッセージ・チャネル・エージェント (MCA) プログラムの記述、または独立ソフトウェア・ベンダーからのインストールを行うことができます。

独自の MCA プログラムの作成は、IBM MQ にユーザー独自の専用通信プロトコルとの相互運用性をもたせたり、IBM MQ がサポートしていないプロトコルを介してメッセージを送信したい場合に、必要になることがあります。 (相手側にある IBM MQ 提供の MCA と相互運用性をもつユーザー独自の MCA を作成することはできません。)

IBM MQ で提供されたもの以外の MCA を使用することに決めた場合は、以下の点を考慮する必要があります。

メッセージの送受信
アプリケーションがメッセージを置く場所、例えば伝送キューなどからメッセージを取得し、それらのメッセージを、通信に使用するプロトコルで送信する送信アプリケーションを作成する必要があります。 また、このプロトコルからメッセージを受け取って宛先キューに入れる受信アプリケーションも作成する必要があります。 送信アプリケーションと受信アプリケーションは、特別のインターフェースではなくメッセージ・キュー・インターフェース (MQI) 呼び出しを使用します。

メッセージが一度しか送達されないようにする必要があります。 この送達には、同期点調整が役立ちます。

チャネル制御機能
チャネルを制御するための独自の管理機能を用意する必要があります。 IBM MQ チャネル管理機能は、チャネルの構成用 (例えば、DEFINE CHANNEL コマンド) にもモニター用 (例えば、DISPLAY CHSTATUS) にも使用できません。

初期設定ファイル
初期設定ファイルが必要な場合は、独自に用意する必要があります。

アプリケーション・データの変換
別のシステムに送信するメッセージのためのデータ変換を考えておく必要もあります。 その場合は、アプリケーションが書き込む先の伝送キューなどの場所からメッセージを検索するときに、MQGET 呼び出しで MQGMO_CONVERT オプションを使用してください。

ユーザー出口
ユーザー出口が必要かどうかを検討してください。 ユーザー出口が必要な場合は、IBM MQ が使用するのと同じインターフェース定義を使用できます。

トリガー
アプリケーションが伝送キューにメッセージを書き込む場合、伝送キュー属性を設定して、メッセージがキューに到着したときに送信 MCA が起動されるように設定できます。

チャネル・イニシエーター
独自のチャネル・イニシエーターを用意する必要があります。

分散キュー管理に関するその他の考慮事項

分散キュー管理用に IBM MQ を準備する際に考慮する、その他のトピックです。 このトピックでは、未配布メッセージ・キュー、 使用中のキュー、システム拡張およびユーザー出口プログラム、ならびにトラステッド・アプリケーションとしてチャネルおよびリスナーを実行する方法 について説明します。

未配布メッセージ・キュー

未配布メッセージ・キュー (送達不能キュー、略して DLQ とも呼ぶ) に 到達したメッセージが確実に処理されるように、 一定の間隔でトリガーされ、実行される、これらのメッセージを処理するプログラムを作成します。

Linux AIX DLQ ハンドラーは、 AIX and Linux システム上の IBM MQ で提供されます。詳しくは、 サンプル DLQ ハンドラー amqsdlq を参照してください。

IBM i IBM MQ for IBM i について詳しくは、IBM MQ for IBM i 送達不能キュー・ハンドラーを参照してください。

使用中のキュー

受信側チャネルの MCA は、メッセージが伝送されていないときでも宛先キューをオープンしておくことができます。 これにより、キューが"使用中"になることが示されます。

チャネルの最大数

IBM i IBM MQ for IBM i の場合は、システムで許可されるチャネルの最大数と、一度にアクティブ化できるチャネルの最大数を指定できます。 これらの番号は、ディレクトリー QIBM/UserData/mqm/qmgrs/*queue_manager_name* 内の `qm.ini` ファイルに指定します。 分散キューイング用の構成ファイル・スタンザを参照してください。

システム拡張部分およびユーザー出口プログラム

チャネル定義には、 メッセージ処理中の特定の時期に追加プログラムを実行できるようにする機能が 用意されています。 これらのプログラムは IBM MQ 提供ではありませんが、ローカル要件に応じてそれぞれのインストール先で用意することができます。

これらのユーザー出口プログラムを実行するには、定義済みの名前を割り当て、チャネル・プログラムに対する呼び出しで利用できるようにしなければなりません。 ユーザー出口プログラムの名前はメッセージ・チャネル定義に入っています。

これらのプログラムに制御を渡したり、これらのプログラムから制御が戻ってきたりするときに必要な、定義済み制御ブロック・インターフェースがあります。

これらのプログラムを呼び出す正確な位置、および制御ブロックと名前の詳細については、メッセージ・チャネル用のチャネル出口プログラムを参照してください。

承認されたアプリケーションとしてのチャネルおよびリスナーの実行

ご使用の環境でパフォーマンスが重要な考慮事項であり、 その環境が安定している場合は、FASTPATH バインディングを使用して、 チャネルおよびリスナーを承認されたものとして実行することができます。 チャネルおよびリスナーが承認されたものとして実行されるかどうかを左右する要因は 2 つあります。

- 環境変数 MQ_CONNECT_TYPE=FASTPATH または MQ_CONNECT_TYPE=STANDARD。 これは、大文字小文字が区別されます。 無効な値を指定すると、無視されます。
- `qm.ini` またはレジストリー・ファイルのチャネル・スタンザ内の MQIBindType。 これを FASTPATH または STANDARD に設定できます。これは、大文字小文字を区別しません。 デフォルトは STANDARD です。

必要な効果を達成するために、次のように環境変数と関連付けて MQIBindType を使用できます。

MQIBindType	環境変数	結果
STANDARD	UNDEFINED	STANDARD
FASTPATH	UNDEFINED	FASTPATH
STANDARD	STANDARD	STANDARD

MQIBindType	環境変数	結果
FASTPATH	STANDARD	STANDARD
STANDARD	FASTPATH	STANDARD
FASTPATH	FASTPATH	FASTPATH
STANDARD	CLIENT	CLIENT
FASTPATH	CLIENT	STANDARD
STANDARD	ローカル	STANDARD
FASTPATH	ローカル	STANDARD

つまり、実際にチャネルおよびリスナーを承認されたものとして実行させるには、次の 2 つの方法しかありません。

1. qm.ini またはレジストリーで MQIBindType=FASTPATH を指定し、環境変数を指定していない。
2. qm.ini またはレジストリーで MQIBindType=FASTPATH を指定し、環境変数を FASTPATH に設定することによって、以下のようにします。

リスナーは安定したプロセスなので、承認されたものとして実行することを検討してください。 不安定なチャネル出口または STOP CHANNEL MODE(TERMINATE) コマンドを使用していないのであれば、 チャネルを承認されたものとして実行することを検討してください。

ALW AIX, Linux, and Windows 上のチャネルのモニターおよび制御

DQM では、リモート・キュー・マネージャーに対するチャネルの作成、モニターおよび制御が必要です。 コマンド、プログラム、IBM MQ Explorer、チャネル定義ファイル、および同期情報のストレージ域を使用して、チャネルを制御できます。

このタスクについて

以下のタイプのコマンドを使用して、チャネルを制御できます。

IBM MQ コマンド (MQSC)
AIX, Linux, and Windows システムの MQSC セッションでは、MQSC を単一コマンドとして使用することができます。 より複雑な、あるいは複数のコマンドを出すためには、MQSC をファイルとして組み立て、それをコマンド行から実行することができます。 詳細については、MQSC コマンドを参照してください。 このセクションでは、分散キューイングに対して MQSC を使用する簡単な例をいくつか記載しています。

チャネル・コマンドは、IBM MQ コマンド・セット (MQSC) のサブセットです。 MQSC と制御コマンドを使用して、以下のことを行うことができます。

- チャネル定義の作成、コピー、表示、変更、および削除
- チャネルの開始と停止、ping、チャネル・シーケンス番号のリセット、およびリンクを再確立できないときの未確定メッセージの解決
- チャネルに関する状況情報の表示

制御コマンド
ある種の機能については、コマンド行から制御コマンド を出すこともできます。 詳しくは、 制御コマンドを使用した IBM MQ for Multiplatforms の管理を参照してください。

プログラム式コマンド形式コマンド
詳細については、PCF コマンドを参照してください。

Windows Linux **IBM MQ Explorer**
Linux、および Windows システムでは、IBM MQ Explorer を使用できます。 これは、制御コマンドまたは MQSC コマンドを使用しないで管理タスクを実行す るためのグラフィカル管理インターフェースを提供します。 チャネル定義はキュー・マネージャー・オブジェクトとして保持されます。

各キュー・マネージャーには、互換性のあるリモート・キュー・マネージャーとの相互接続を制御するための DQM コンポーネントが備わっています。 保管域には、シーケンス番号と、作業論理単位 *(LUW)* ID があります。 これらはチャネルの同期をとる目的で使用されます。

異なるタイプのコマンドを使用して、メッセージ・チャネルのセットアップおよび制御を行う場合に使用できる機能のリストについては、244 ページの表 21 を参照してください。

手順

- 243 ページの『チャネルのセットアップおよび制御に必要な機能』
- 245 ページの『オブジェクトの準備作業』
- 252 ページの『Windows における通信のセットアップ』
- 259 ページの『AIX and Linux における通信のセットアップ』

関連タスク

265 ページの『IBM i 上のチャネルのモニターおよび制御』
リモート・キュー・マネージャーにつながるチャネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。 各キュー・マネージャーには、互換性のあるリモート・キュー・マネージャーとの相互接続を制御するための DQM プログラムが備わっています。

関連資料

ALW AIX, Linux, and Windows でのチャネル・プログラム

ALW AIX, Linux, and Windows におけるメッセージ・チャネルの計画例

構成情報の例

チャネルの属性

ALW チャネルのセットアップおよび制御に必要な機能

いくつかの IBM MQ 機能が、チャネルをセットアップおよび制御するのに必要となる場合があります。 チャネルの機能についてはこのトピックで説明します。

チャネル定義を作成する際には、IBM MQ が提供するデフォルト値を使用して、チャネル名、作成しようとするチャネルのタイプ、使用する通信方式、伝送キュー名、および接続名を指定することができます。

チャネル名はチャネルの両端で同じ名前になっていなければならず、またネットワーク内で固有の名前でなければなりません。 ただし、IBM MQ オブジェクト名として有効な文字のみを使用するようにしてください。

他のチャネル関連機能についての詳細は、以下のトピックを参照してください。

- 245 ページの『オブジェクトの準備作業』
- 246 ページの『関連オブジェクトの作成』
- 246 ページの『デフォルト・オブジェクトの作成』
- 246 ページの『チャネルの作成』
- 247 ページの『チャネルの表示』
- 247 ページの『チャネル状況の表示』
- 248 ページの『Ping を使用したリンクの検査』
- 248 ページの『チャネルの開始』
- 250 ページの『チャネルの停止』
- 250 ページの『チャネルの名前変更』
- 251 ページの『チャネルのリセット』
- 251 ページの『チャネルの未確定メッセージの解決』

244 ページの表 21 は、必要になる場合のある IBM MQ 機能の全リストです。

表 *21. AIX, Linux, and Windows* システムで必要な機能			
Function	**制御コマンド**	**MQSC**	**IBM MQ エクスプローラーで同等のコマンドがあるか**
キュー・マネージャー機能			
キュー・マネージャーの変更		ALTER QMGR	はい
キュー・マネージャーの作成	crtmqm		はい
キュー・マネージャーの削除	dltmqm		はい
キュー・マネージャーの表示		DISPLAY QMGR	はい
キュー・マネージャーの終了	endmqm		はい
キュー・マネージャーの ping		PING QMGR	いいえ
キュー・マネージャーの始動	strmqm		はい
コマンド・サーバー機能			
コマンド・サーバーの表示	dspmqcsv		いいえ
コマンド・サーバーの終了	endmqcsv		いいえ
コマンド・サーバーの開始	strmqcsv		いいえ
キュー機能			
キューの変更		ALTER QALIAS ALTER QLOCAL ALTER QMODEL ALTER QREMOTE ALTER (キュー) を参照してください。	はい
キューのクリア		CLEAR QLOCAL	はい
キューの作成		DEFINE QALIAS DEFINE QLOCAL DEFINE QMODEL DEFINE QREMOTE DEFINE (キュー) を参照してください。	はい
キューの削除		DELETE QALIAS DELETE QLOCAL DELETE QMODEL DELETE QREMOTE DELETE (キュー) を参照してください。	はい
キューの表示		DISPLAY QUEUE	はい
プロセス機能			
プロセスの変更		ALTER PROCESS	はい
プロセスの作成		DEFINE PROCESS	はい
プロセスの削除		Delete Process	はい

表 21. AIX, Linux, and Windows システムで必要な機能 (続き)			
Function	**制御コマンド**	**MQSC**	**IBM MQ エクスプローラーで同等のコマンドがあるか**
プロセスの表示		DISPLAY PROCESS	はい
チャネル機能			
チャネルの変更		ALTER CHANNEL	はい
チャネルの作成		DEFINE CHANNEL	はい
チャネルの削除		Delete Channel	はい
チャネルの表示		DISPLAY CHANNEL	はい
チャネル状況の表示		DISPLAY CHSTATUS	はい
チャネルの終了		STOP CHANNEL	はい
チャネルの ping		PING CHANNEL	はい
チャネルのリセット		RESET CHANNEL	はい
チャネルの解決		RESOLVE CHANNEL	はい
チャネルの実行	runmqchl	START CHANNEL	はい
チャネル・イニシエーターの実行	runmqchi	START CHINIT	いいえ
リスナー [1] の実行	runmqlsr	START LISTENER	いいえ
リスナーの終了	endmqlsr、以下のプラットフォームでのみ使用可能: • AIX AIX • Windows Windows システム		いいえ

注:

1. リスナーは、キュー・マネージャーが開始されると自動的に開始されます。

ALW オブジェクトの準備作業

チャネルを開始する前に、チャネルが定義されていて、それに関連するオブジェクトが存在し、使用可能になっていなければなりません。 このセクションでは、その方法を示します。

IBM MQ コマンド (MQSC) または IBM MQ Explorer を使用して、以下のことを行います。

1. メッセージ・チャネルおよび関連オブジェクトの定義
2. メッセージ・チャネルのモニターと制御

定義が必要な関連オブジェクトは次のとおりです。

- 伝送キュー
- リモート・キュー定義
- キュー・マネージャー別名定義
- 応答先キュー別名の定義
- 応答先ローカル・キュー
- トリガー操作のプロセス (MCA)

- メッセージ・チャネル定義

各チャネルに関する特定の通信リンクが定義されて、使用可能になっていなければ、チャネルを実行することはできません。 LU 6.2、TCP/IP、NetBIOS、SPX、および DECnet リンクの定義方法については、ご使用のシステムに関する特定の通信の手引きを参照してください。 構成情報の例も参照してください。

オブジェクトの作成および処理についての詳細は、以下のサブトピックを参照してください。

ALW 関連オブジェクトの作成

MQSC は関連オブジェクトを作成するのに使用されます。

MQSC を使用して、次のようなキューおよび別名オブジェクトを作成します。伝送キュー、リモート・キュー定義、キュー・マネージャー別名定義、応答先キュー別名定義、および応答先ローカル・キュー。

トリガー操作のプロセス (MCA) の定義も同様に作成する必要があります。

必要なすべてのオブジェクトを作成する方法を示す例については、AIX, Linux, and Windows におけるメッセージ・チャネルの計画例を参照してください。

ALW デフォルト・オブジェクトの作成

デフォルト・オブジェクトは、キュー・マネージャーが作成されると自動的に作成されます。 これらのオブジェクトとは、キュー、チャネル、プロセス定義、管理キューです。 デフォルト・オブジェクトが作成された後、-c オプションで strmqm コマンドを実行することによっていつでもそれらを置換できます。

crtmqm コマンドを使用して、キュー・マネージャーを作成すると、コマンドは、プログラムを開始して、一連のデフォルト・オブジェクトも作成します。

1. 次に、各デフォルト・オブジェクトが作成されます。 プログラムは、正しく定義されたオブジェクト、存在していて置換されたオブジェクト、および試みが失敗したオブジェクトのカウントを保持します。
2. プログラムは作成者に結果を表示し、エラーが発生すると、詳細については該当するエラー・ログを参照するように指示します。

プログラムが実行を終了したら、strmqm コマンドを使用して、キュー・マネージャーを開始します。

crtmqm および strmqm コマンドについて詳しくは、 制御コマンドを使用した IBM MQ for Multiplatforms の管理 を参照してください。

デフォルト・オブジェクトの変更

-c オプションを指定すると、オブジェクトが作成されているとき、キュー・マネージャーが一時的に開始されて、その後、再度終了します。 strmqm に -c オプションを付けて発行すると、既存のシステム・オブジェクトがデフォルト値でリフレッシュされます (例えば、チャネル定義の MCAUSER 属性はブランクに設定されます)。 キュー・マネージャーを開始したい場合は、-c オプションなしで、再度 strmqm コマンドを使用する必要があります。

デフォルト・オブジェクトを変更する場合は、旧 amqscoma.tst ファイルの自分用のバージョンを作成して、それを編集できます。

ALW チャネルの作成

2 つのチャネル定義、つまり接続の両端に 1 つずつの定義を作成します。 第 1 のキュー・マネージャーで最初のチャネルを作成します。 次に、リンクのもう一方の第 2 のキュー・マネージャーで 2 番目のチャネル定義を作成します。

両端とも同じチャネル名を使用して定義する必要があります。 チャネルの両端は、送信側と受信側のように、互換性を備えたチャネル・タイプでなければなりません。

リンクの一端でチャネル定義を作成するためには、MQSC コマンド DEFINE CHANNEL を使用します。 これには、チャネル名、該当接続側のチャネル・タイプ、接続名、記述 (必要な場合)、伝送キューの名前 (必要な場合)、および伝送プロトコルを指定します。 また、すでに収集した情報を使用して、必要なチャネル・タイプについて、システムのデフォルト値とは異なる値にしたい属性を指定します。

チャネル属性で説明されているチャネル属性の値を決めるためのヘルプ情報が用意されています。

注 : 使用するネットワーク内のすべてのチャネルに固有の名前を付けることをお勧めします。 発信元および宛先のキュー・マネージャーの名前をチャネル名に入れるのもこのための 1 つの方法です。

チャネル作成の例

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) +
DESCR('Sender channel to QM2') +
CONNAME(QM2) TRPTYPE(TCP) XMITQ(QM2) CONVERT(YES)
```

MQSC のすべての例で、このコマンドがコマンドのファイル内にあるように、また AIX, Linux, and Windows で入力されているように表示されます。 2 つの方法は同じに思われますが、コマンドを対話形式で入力するにはまず、MQSC セッションを開始しておかなければならない点が異なります。 `runmqsc`(デフォルト・キュー・マネージャーの場合) または `runmqsc` *qmname* (*qmname* は必要なキュー・マネージャーの名前) を入力します。 その後で、例に示されているように任意の数のコマンドを入力します。

移植性を考慮して、コマンド行の長さは、最高 72 文字に限定されています。 複数行にわたる場合は、上で示したとおり、連結文字 + を使用します。

- Windows では、Ctrl-z を使用してコマンド・ラインの入力を終了します。
- AIX and Linux では、Ctrl-d を使用します。
- また、AIX, Linux, and Windows で、**end** コマンドを使用します。

ALW チャネルの表示

MQSC コマンド DISPLAY CHANNEL は、チャネルの属性を表示するために使用されます。

特定の属性が要求されておらず、指定されているチャネル名が汎用でない場合、DISPLAY CHANNEL コマンドのすべてのパラメーターがデフォルトで使用されます。

属性については チャネル属性を参照してください。

チャネル表示の例

```
DISPLAY CHANNEL(QM1.TO.QM2) TRPTYPE,CONVERT

DISPLAY CHANNEL(QM1.TO.*) TRPTYPE,CONVERT

DISPLAY CHANNEL(*) TRPTYPE,CONVERT

DISPLAY CHANNEL(QM1.TO.QMR34) ALL
```

ALW チャネル状況の表示

MQSC コマンド DISPLAY CHSTATUS は、チャネル名、およびチャネルの現在状況または保管済み情報の状況を入手したいかどうかを指定して使用します。

DISPLAY CHSTATUS はすべてのメッセージ・チャネルに適用されます。 サーバー接続チャネル以外の MQI チャネルには適用されません。

表示される情報は以下のとおりです。

- チャネル名
- 通信接続名
- チャネルの未確定状況 (該当する場合)
- 最終シーケンス番号
- 伝送キュー名 (該当する場合)
- 未確定 ID (該当する場合)
- 最後にコミットされたシーケンス番号
- 論理作業単位 ID
- プロセス ID

- Windows スレッド ID (Windows のみ)

チャネル状況の表示例

```
DISPLAY CHSTATUS(*) CURRENT
DISPLAY CHSTATUS(QM1.TO.*) SAVED
```

保存された状況は、メッセージの少なくとも 1 つのバッチがチャネルに伝送されるまでは適用されません。状況は、チャネルが (STOP CHL コマンドで) 停止した場合、およびキュー・マネージャーが終了した場合にも保管されます。

ALW *Ping* を使用したリンクの検査

MQSC コマンド **PING CHANNEL** は、リモート・エンドと固定データ・メッセージを交換するために使用します。

Ping により、システム監視プログラムは、リンクが使用可能であって機能していることをある程度確認することができます。

Ping を選択しても、伝送キューとターゲット・キューが使用されることはありません。 チャネル定義、関連する通信リンク、およびネットワーク設定が使用されます。 それは、現在チャネルがアクティブでない場合にのみ使用できます。

Ping は、送信側チャネル、サーバー・チャネル、クラスター送信側チャネルからのみ使用できます。 リンクの反対側で対応チャネルが開始され、開始パラメーターの折衝が実施されます。 エラーがある場合には、通常どおり通知されます。

メッセージ交換の結果は、`Ping complete` またはエラー・メッセージで示されます。

LU 6.2 での ping

ping が呼び出されるとき、デフォルトでは、受信側には、ユーザー ID またはパスワードは送られません。ユーザー ID やパスワードが必要な場合は、開始側のチャネル定義で作成することができます。 チャネル定義にパスワードを入れると、それは、保管される前に IBM MQ によって暗号化されます。 その後、暗号解読されて、会話を介して流されます。

関連タスク

ping を使用した通信のテスト

接続を検査するためのチャネルの Ping

関連資料

PING CHANNEL (チャネル応答の検査)

ALW チャネルの開始

MQSC コマンド START CHANNEL は、送信側、サーバー、および要求側のチャネルに対して使用します。アプリケーションがメッセージ交換を行えるようにするには、インバウンド接続に対してリスナー・プログラムを開始する必要があります。

START CHANNEL は、チャネルがキュー・マネージャーによるトリガー操作で設定されている場合には必要ありません。

送信側 MCA は、開始されると、チャネル定義を読み取り、伝送キューをオープンします。 チャネル開始シーケンスが実行され、リモートで受信側またはサーバー側のチャネルの対応する MCA が開始されます。 それらのチャネルが開始すると、送信側プロセスおよびサーバー・プロセスはメッセージが伝送キューに到達するのを待ち、到達したメッセージを伝送します。

トリガー操作を使用するか、チャネルをスレッドとして実行する場合には、チャネル・イニシエーターが開始キューをモニターできるようにしておいてください。 チャネル・イニシエーターは、キュー・マネージャーの一部としてデフォルトで始動します。

ただし、TCP および LU 6.2 は、その他に以下の機能を提供します。

- Linux AIX AIX and Linux 上の TCP では、inetd を構成してチャネルを開始できます。 inetd は、独立したプロセスとして開始されます。
- Linux AIX AIX and Linux における LU 6.2 の場合、SNA 製品を構成して LU 6.2 応答側プロセスを開始します。
- Windows Windows での LU 6.2 では、SNA サーバーを使用すると、TpStart (SNA サーバーに付属のユーティリティー) を使用してチャネルを開始できます。 TpStart は、独立したプロセスとして開始されます。

「Start (開始)」オプションを使用すると、必要な場合には必ずチャネルが再同期化されます。

開始手続きを正常に行うためには、次の条件が必要です。

- ローカルおよびリモートのチャネル定義が存在している。 受信側またはサーバー接続のチャネルに該当するチャネル定義がない場合は、チャネルが自動定義されていると、自動的にデフォルトのチャネル定義が作成されます。 チャネル自動定義出口プログラムを参照してください。
- 伝送キューが存在していて、しかも他のチャネルで使用されていない。
- ローカルおよびリモートの MCA が存在している。
- 通信リンクが利用可能である。
- ローカルおよびリモートのキュー・マネージャーが実行されている。
- メッセージ・チャネルがまだ実行されていない状態である。

チャネル開始要求が受け入れられたことを確認させるメッセージが画面に戻されます。 開始コマンドが正常に行われたかどうかを確認するためには、エラー・ログを調べるか、DISPLAY CHSTATUS を使用してください。 エラー・ログは次のとおりです。

Windows **Windows**

MQ_DATA_PATH\qmgrs\qmname\errors\AMQERR01.LOG (qmname というキュー・マネージャーごとに)

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

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

注 : Windows では、Windows システム・アプリケーション・イベント・ログでメッセージを取得することもできます。

Linux AIX **AIX and Linux**

/var/mqm/qmgrs/qmname/errors/AMQERR01.LOG (qmname というキュー・マネージャーごとに)

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

AIX, Linux, and Windows では、**runmqlsr** コマンドを使用して IBM MQ リスナー・プロセスを開始します。 デフォルトでは、チャネル接続インバウンド要求によってリスナー・プロセスが MCA を amqrmppa プロセスのスレッドとして開始します。

```
runmqlsr -t tcp -m QM2
```

アウトバウンド接続については、次の 3 つの方法のどれか 1 つでチャネルを開始する必要があります。

1. MCATYPE パラメーターに基づいて、プロセスまたはスレッドとしてチャネルを開始させるには、チャネル名を指定して、MQSC コマンド START CHANNEL を使用します。 (チャネルがスレッドとして開始されるときには、チャネルがチャネル・イニシエーターのスレッドです。)

   ```
   START CHANNEL(QM1.TO.QM2)
   ```

2. 制御コマンド runmqchl を使用して、プロセスとしてチャネルを開始させます。

```
runmqchl -c QM1.TO.QM2 -m QM1
```

3. チャネル・イニシエーターを使用して、チャネルを起動します。

ALW チャネルの停止

MQSC コマンド STOP CHANNEL は、チャネルに活動の停止を要求するために使用します。 オペレーターがチャネルを再始動させるまで、チャネルは新しいメッセージのバッチを開始しません。

停止したチャネルの再始動に関する情報については、231 ページの『停止したチャネルの再始動』を参照してください。

このコマンドは、MQCHT_CLNTCONN 以外のどのタイプのチャネルに対しても発行できます。

必要な停止のタイプを選択できます。

静止の停止例

```
STOP CHANNEL(QM1.TO.QM2) MODE(QUIESCE)
```

このコマンドは、通常の手順でチャネルがクローズされることを要求します。 現行のメッセージ・バッチは完了され、チャネルの他方の側で同期点手順が行われます。 チャネルがアイドル状態であると、このコマンドは受信側チャネルを終了させません。

強制停止の例

```
STOP CHANNEL(QM1.TO.QM2) MODE(FORCE)
```

このオプションは、チャネルを即時に停止しますが、チャネルのスレッドやプロセスを終了しません。 チャネルは、メッセージの現行バッチの処理を完了しないため、そのチャネルが未確定のままになることがあります。 一般的には、静止の停止オプションの使用を考慮します。

終了停止の例

```
STOP CHANNEL(QM1.TO.QM2) MODE(TERMINATE)
```

このオプションは、チャネルを即時に停止し、チャネルのスレッドやプロセスを終了します。

停止 (静止) 停止の例

```
STOP CHANNEL(QM1.TO.QM2) STATUS(STOPPED)
```

このコマンドでは MODE を指定しないため、デフォルトで MODE(QUIESCE) になります。 このコマンドはチャネルの停止を要求するので、チャネルの自動的な再始動は行えません。したがって、手動で開始する必要があります。

停止 (静止) の非アクティブ状態の例

```
STOP CHANNEL(QM1.TO.QM2) STATUS(INACTIVE)
```

このコマンドでは MODE を指定しないため、デフォルトで MODE(QUIESCE) になります。 このコマンドはチャネルの非アクティブ化を要求するので、必要に応じて、チャネルは自動的に再始動されます。

ALW チャネルの名前変更

MQSC を使用して、メッセージ・チャネルを名前変更します。

MQSC を使用して、以下のステップを行ってください。

1. STOP CHANNEL を使用してチャネルを停止させる。

2. DEFINE CHANNEL を使用して、新しい名前で重複したチャネル定義を作成する。
3. DISPLAY CHANNEL を使用して正常に作成されたかを検査する。
4. DELETE CHANNEL を使用して元のチャネル定義を削除する。

メッセージ・チャネルの名前を変更することを決めた場合は、チャネルは両側に 1 つずつ、計 2 個のチャネル定義をもっていることに注意してください。 必ず両側のチャネルを同時に名前変更するようにします。

ALW チャネルのリセット

MQSC コマンド RESET CHANNEL は、メッセージ・シーケンス番号を変更するために使用します。

RESET CHANNEL コマンドはどのメッセージ・チャネルに対しても使用できますが、MQI チャネル (クライアント接続またはサーバー接続) には使用できません。 次にそのチャネルが開始されると、最初のメッセージは新しいシーケンス番号になります。

送信側またはサーバーのチャネルでコマンドが出されると、他方のチャネルが再始動されたときに、変更が通知されます。

関連概念

245 ページの『オブジェクトの準備作業』
チャネルを開始する前に、チャネルが定義されていて、それに関連するオブジェクトが存在し、使用可能になっていなければなりません。 このセクションでは、その方法を示します。

220 ページの『チャネル制御機能』
チャネル制御機能は、チャネルを定義、モニター、および制御するための機能を備えています。

関連タスク

190 ページの『分散キューイングの構成』
このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

関連資料

RESET CHANNEL

ALW チャネルの未確定メッセージの解決

MQSC コマンド RESOLVE CHANNEL は、送信側またはサーバーによってメッセージが未確定のままにされているときに使用します。 例えば、リンクの反対側が終了していて、メッセージが戻される見込みがないなどの理由によります。

RESOLVE CHANNEL コマンドは、2 つのパラメーター、BACKOUT または COMMIT のいずれかを受信します。 バックアウトではメッセージは伝送キューに復元され、コミットではメッセージは廃棄されます。

チャネル・プログラムは、パートナーとのセッションの確立を試みません。 その代わりに、未確定メッセージを表す作業論理単位 ID (LUWID) を判別します。 そのあとで、要求に応じて次のいずれかを発行します。

- メッセージを伝送キューに復元するための BACKOUT
- メッセージを伝送キューから削除するための COMMIT

解決を正常に行うためには、次の条件が必要です。

- チャネルが非アクティブになっている。
- チャネルが未確定になっている。
- チャネル・タイプが送信側、サーバー、またはクラスター送信側である。
- ローカルのチャネル定義が存在している。
- ローカル・キュー・マネージャーが実行している。

関連概念

245 ページの『オブジェクトの準備作業』
チャネルを開始する前に、チャネルが定義されていて、それに関連するオブジェクトが存在し、使用可能になっていなければなりません。 このセクションでは、その方法を示します。

220 ページの『チャネル制御機能』
チャネル制御機能は、チャネルを定義、モニター、および制御するための機能を備えています。

関連タスク

190 ページの『分散キューイングの構成』
このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

関連資料

RESOLVE CHANNEL

Windows における通信のセットアップ

分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 これが成功するためには、接続が定義されていて、使用可能になっていなければなりません。 このセクションでは、IBM MQ for Windows システムで使用可能な通信の形式を使用して、これを行う方法について説明しています。

始める前に

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

MQ Adv. CD TCP/IP を使用するメッセージ・チャネルは、ネットワーク・スループットを大幅に向上させることができる高速 TCP/IP トンネルを提供する IBM Aspera faspio Gateway を指すことができます。 資格のあるプラットフォームで実行されているキュー・マネージャーは、Aspera gateway を介して接続できます。 ゲートウェイ自体は、 Red Hat または Ubuntu Linux、あるいは Windows にデプロイされます。 Linux または Windows での Aspera gateway 接続の定義を参照してください。

このタスクについて

Windows 上の IBM MQ の通信をセットアップする場合、以下のタイプの通信から選択できます。

- TCP/IP
- LU 6.2
- NetBIOS

手順

- Windows システムの通信のセットアップについては、選択した通信タイプのサブトピックを参照してください。
  - 253 ページの『Windows での TCP 接続の定義』
  - 255 ページの『Windows での LU 6.2 接続の定義』
  - 256 ページの『Windows での NetBIOS 接続の定義』

  TCP/IP 以外の通信プロトコルを使用している環境では、IBM MQ for Windows のすべての機能を利用できるとは限りません。 使用できない項目は IBM MQ Explorer です。

関連タスク

242 ページの『AIX, Linux, and Windows 上のチャネルのモニターおよび制御』
DQM では、リモート・キュー・マネージャーに対するチャネルの作成、モニターおよび制御が必要です。 コマンド、プログラム、IBM MQ Explorer、チャネル定義ファイル、および同期情報のストレージ域を使用して、チャネルを制御できます。

15 ページの『クライアントとサーバー間の接続の構成』
IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャネルを定義します。

259 ページの『AIX and Linux における通信のセットアップ』

分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 これが成功するためには、接続が定義されていて、使用可能になっていなければなりません。 このセクションでは、IBM MQ for UNIX or Linux システムで使用可能な通信の形式を使用して、これを行う方法について説明しています。

関連資料

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

Windows *Windows での TCP 接続の定義*

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

始める前に

MQ Adv. CD TCP/IP を使用するメッセージ・チャネルは、ネットワーク・スループットを大幅に向上させることができる高速 TCP/IP トンネルを提供する IBM Aspera faspio Gateway を指すことができます。 資格のあるプラットフォームで実行されているキュー・マネージャーは、Aspera gateway を介して接続できます。 ゲートウェイ自体は、 Red Hat または Ubuntu Linux、あるいは Windows にデプロイされます。 Linux または Windows での Aspera gateway 接続の定義を参照してください。

送信側

チャネル定義の「Connection name (接続名)」フィールドにホスト名またはターゲット・マシンの TCP アドレスを指定します。

接続されるポートのデフォルト値は 1414 です。 Internet Assigned Numbers Authority (インターネットによる番号割り当て権限) によってポート番号 1414 が IBM MQ に割り当てられます。

デフォルト以外のポート番号を使用する場合は、チャネル・オブジェクト定義の接続名フィールドに次のようにポート番号を指定します。

```
DEFINE CHANNEL('channel name') CHLTYPE(SDR) +
          TRPTYPE(TCP) +
          CONNAME('OS2ROG3(1822)') +
          XMITQ('XMitQ name') +
          REPLACE
```

ここで、OS2ROG3 はリモート・キュー・マネージャーの DNS ホスト名、1822 は必要なポート番号です。 (これは受信側のリスナーが待機するポートでなければなりません。)

チャネル・オブジェクト定義の変更を反映させるには、実行中のチャネルを停止して再始動する必要があります。

IBM MQ for Windows の `.ini` ファイルでデフォルトのポート番号を指定することにより、デフォルトのポート番号を変更できます:

```
TCP:
Port=1822
```

注 : IBM MQ は、使いたい TCP/IP ポート番号を選択するときに、次のような順序で最初に見つけたポート番号を使います。

1. チャネル定義またはコマンド行で明示的に指定されたポート番号。 この番号を指定すると、チャネルのデフォルト・ポート番号を変更することができます。
2. `.ini` ファイルの TCP スタンザに指定されているポート属性。 この番号を指定すると、キュー・マネージャーのデフォルト・ポート番号を変更することができます。
3. デフォルト値 1414。 これは、インバウンドとアウトバウンドの両方の接続用に Internet Assigned Numbers Authority が IBM MQ に割り当てた番号です。

qm.ini を使用して設定する値について詳しくは、分散キューイング用の構成ファイル・スタンザを参照してください。

TCP での受信

受信側のチャネル・プログラムを開始するには、着信ネットワーク要求を検出し、関連チャネルを開始するリスナー・プログラムが起動されている必要があります。 IBM MQ リスナーを使用することができます。

受信側のチャネル・プログラムは、送信側チャネルからの開始要求に応じて開始されます。

受信側のチャネル・プログラムを開始するには、着信ネットワーク要求を検出し、関連チャネルを開始するリスナー・プログラムが起動されている必要があります。 IBM MQ リスナーを使用することができます。

IBM MQ が提供するリスナーは、新しいチャネルをスレッドとして開始させますが、これを実行させるためには、runmqlsr コマンドを使用します。

runmqlsr コマンドを使用する基本的な例を次に示します。

```
runmqlsr -t tcp [-m QMNAME] [-p 1822]
```

大括弧は、オプション・パラメーターを示します。デフォルト・キュー・マネージャーの場合は QMNAME は必要ありません。また、デフォルト・ポート番号 (1414) を使用する場合はポート番号は必要ありません。 ポート番号は 65535 以下でなければなりません。

注 : IBM MQ は、使いたい TCP/IP ポート番号を選択するときに、次のような順序で最初に見つけたポート番号を使います。

1. チャネル定義またはコマンド行で明示的に指定されたポート番号。 この番号を指定すると、チャネルのデフォルト・ポート番号を変更することができます。
2. `.ini` ファイルの TCP スタンザに指定されているポート属性。 この番号を指定すると、キュー・マネージャーのデフォルト・ポート番号を変更することができます。
3. デフォルト値 1414。 これは、インバウンドとアウトバウンドの両方の接続用に Internet Assigned Numbers Authority が IBM MQ に割り当てた番号です。

最良のパフォーマンスを得るには、241 ページの『承認されたアプリケーションとしてのチャネルおよびリスナーの実行』で説明されているように、 IBM MQ リスナーをトラステッド・アプリケーションとして実行します。 承認されたアプリケーションについては、承認されたアプリケーションに関する制限を参照してください。

TCP/IP SO_KEEPALIVE オプションの使用

Windows SO_KEEPALIVE オプションを使用する場合、レジストリーに以下の項目を追加する必要があります。

```
TCP:
KeepAlive=yes
```

SO_KEEPALIVE オプションの詳細については、227 ページの『チャネルの相手側がまだ使用可能であるかどうかの検査』を参照してください。

Windows では、 Windows **KeepAliveTime** オプションの `HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters` レジストリー値によって、接続が検査される前に経過する間隔が制御されます。 デフォルト値は 2 時間です。

TCP リスナー・バックログ・オプションの使用

TCP では、サーバーとクライアントとの間で 3 ウェイ・ハンドシェークが実行されない限り、接続は不完全として扱われます。 これらの接続は、未解決接続要求と呼ばれます。 最大値をこれらの未解決接続要求に設定し、リスナーが要求を受け入れるのを TCP ポート上で待っている要求のバックログと考えることができます。

詳細については、 262 ページの『IBM MQ for Multiplatforms での TCP リスナー・バックログ・オプションの使用』 を参照してください。また、 Windows の特定の値を参照してください。

Windows *Windows での LU 6.2 接続の定義*

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

SNA が構成された後、次のように進んでください。

詳細は以下の表を参照してください。

表 *22.* ローカル *Windows* システムにおけるリモート・キュー・マネージャー・プラットフォーム用の設定値

リモート・プラットフォーム	TPNAME	TPPATH
z/OS または MVS/ESA (CICS を使用しない)	リモート・キュー・マネージャーの対応するサイド情報内と同じ。	-
z/OS または MVS/ESA (CICS を使用する)	CKRC (送信側) CKSV (要求側) CKRC (サーバー)	-
IBM i	IBM i システムにおいて経路指定項目の比較値と同じ。	-
AIX and Linux システム	リモート・キュー・マネージャーの対応するサイド情報内と同じ。	*MQ_INSTALLATION_PATH*/bin/amqcrs6a
Windows	Windows Run Listener コマンドで指定されたとおり、または Windows において TpSetup を使用して定義された呼び出し可能なトランザクション・プログラム。	*MQ_INSTALLATION_PATH*\bin\amqcrs6a

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

同一のマシン上に複数のキュー・マネージャーがある場合は、チャネル定義内の TPname をそれぞれ固有にする必要があります。

AnyNet® SNA over TCP/IP の構成に関する最新情報については、オンラインの IBM 資料である AnyNet SNA over TCP/IP および SNA ノード操作を参照してください。

関連概念

255 ページの『Windows における LU 6.2 での送信側』
使用している LU 6.2 製品の管理アプリケーションから CPI-C 側オブジェクト (シンボリック宛先) を作成します。 その名前をチャネル定義の「接続名」フィールドに入れます。 また、パートナーに対する LU 6.2 リンクも作成します。

256 ページの『Windows における LU 6.2 での受信』
受信側のチャネル・プログラムは、送信側チャネルからの開始要求に応じて開始されます。

Windows *Windows における LU 6.2 での送信側*

使用している LU 6.2 製品の管理アプリケーションから CPI-C 側オブジェクト (シンボリック宛先) を作成します。 その名前をチャネル定義の「接続名」フィールドに入れます。 また、パートナーに対する LU 6.2 リンクも作成します。

CPI-C サイド・オブジェクトには、受信側マシンのパートナー LU 名、TP 名、およびモード名を入れます。以下に例を示します。

```
Partner LU Name              OS2ROG2
Partner TP Name              recv
Mode Name                    #INTER
```

Windows *Windows における LU 6.2 での受信*

受信側のチャネル・プログラムは、送信側チャネルからの開始要求に応じて開始されます。

受信側のチャネル・プログラムを開始するには、着信ネットワーク要求を検出し、関連チャネルを開始するリスナー・プログラムが起動されている必要があります。 このリスナー・プログラムは、待機の対象となる TpName を指定して、RUNMQLSR コマンドを使用して開始させます。 または、Windows では、SNA サーバー配下の TpStart を使用することもできます。

RUNMQLSR コマンドの使用

次は、リスナーを開始するコマンドの一例です。

```
RUNMQLSR -t LU62 -n RECV -m QMNAME
```

ここで RECV は、相手側 (送信側) で「TpName to start on the remote side (リモート側で開始する際の TpName)」として指定されている TpName です。 このコマンドの最後の部分で使用される **-m** パラメーターはオプションであり、デフォルトのキュー・マネージャーでは必須ではありません。

1 つのマシンで、複数のキュー・マネージャーを実行することが可能です。 どのキュー・マネージャーにも異なる TpName を割り当ててから、それぞれに対してリスナー・プログラムを開始する必要があります。以下に例を示します。

```
RUNMQLSR -t LU62 -m QM1 -n TpName1
RUNMQLSR -t LU62 -m QM2 -n TpName2
```

最良のパフォーマンスを得るには、 トラステッド・アプリケーションとしてのチャネルおよびリスナーの実行で説明されているように、 IBM MQ リスナーをトラステッド・アプリケーションとして実行します。 承認されたアプリケーションについては、承認されたアプリケーションの制約事項を参照してください。

次のコマンドを使用して、非アクティブであるキュー・マネージャーで実行されているすべての IBM MQ リスナーを停止できます。

```
ENDMQLSR -m QMNAME
```

Windows での MicrosoftSNA サーバーの使用

TpSetup を (SNA サーバー SDK から) 使用して呼び出し可能 TP を定義して、amqcrs6a.exe を取り出すか、または手動で各種の登録値を設定することができます。 amqcrs6a.exe に渡す必要のあるパラメーターは、次のとおりです。

```
-m QM -n TpName
```

ここで、*QM* はキュー・マネージャー名であり、*TpName* は TP 名です。 詳細については、「*Microsoft SNA Server APPC Programmers Guide*」または「*Microsoft SNA Server CPI-C Programmers Guide*」を参照してください。

キュー・マネージャー名を指定しないと、デフォルトのキュー・マネージャーが想定されます。

Windows *Windows での NetBIOS 接続の定義*

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

タイプに関係なく、実行中のチャネルは、1 つの NetBIOS セッションおよび 1 つの NetBIOS コマンドを使用します。 IBM NetBIOS 環境では、複数のプロセスで同一のローカル NetBIOS 名を使用できます。 したがって、IBM MQ で使用する場合は、1 つの NetBIOS 名のみが使用可能になっている必要があります。 例えば、Novell の NetBIOS エミュレーションなど、他のベンダー環境では、プロセスごとに異なるローカル名が要求されます。 使用している NetBIOS 製品の資料で、必要な条件を確認してください。

必ず各タイプの十分なリソースがすでに使用可能になっていることを確認してください。あるいは、構成情報として指定されている最大値を大きくします。 値を変更した場合は常に、システムを再開する必要があります。

システム再開中、NetBIOS 装置ドライバーは、アプリケーションで使用するときに使用可能なセッション、コマンド、および名前の数を表示します。 これらのリソースは、同一システムで実行しているすべての NetBIOS ベース・アプリケーションに使用可能です。 したがって、IBM MQ がそれらのリソースの獲得を必要とする前に、別のアプリケーションがそれらを消費してしまう可能性があります。 この問題への対処法については、LAN ネットワーク管理者に相談してください。

関連概念

257 ページの『IBM MQ ローカル NetBIOS 名の定義』
IBM MQ チャネル・プロセスで使用されるローカル NetBIOS 名は、3 つの方法で指定できます。

258 ページの『キュー・マネージャーの NetBIOS セッション、コマンド、および名前制限の設定』
NetBIOS セッション、コマンド、および名前に関するキュー・マネージャー制限は、2 つの方法で指定できます。

258 ページの『LAN アダプター番号の設定』
チャネルが NetBIOS 間で正常に動作するためには、それぞれの側のアダプター・サポートに互換性がなければなりません。 IBM MQ では、qm.ini ファイルの NETBIOS スタンザの中で AdapterNum 値を使用し、runmqlsr コマンドに **-a** パラメーターを指定することによって、LAN アダプター (LANA) 番号の選択を制御できます。

258 ページの『NetBIOS 接続の開始』
接続を開始するのに必要なステップを定義します。

259 ページの『NetBIOS 接続のターゲット・リスナーの定義』
NetBIOS 接続の受信側で行われるステップを定義します。

Windows *IBM MQ* ローカル *NetBIOS* 名の定義

IBM MQ チャネル・プロセスで使用されるローカル NetBIOS 名は、3 つの方法で指定できます。

3 つの方法を優先順位に従って示してあります。

1. **runmqlsr** コマンドの **-l** パラメーターに指定された値。以下に例を示します。

   ```
   runmqlsr -t netbios -l my_station
   ```

2. コマンドによって設定された値を持つ **MQNAME** 環境変数:

   ```
   SET MQNAME= my_station
   ```

   以下に例を示します。

   ```
   SET MQNAME=CLIENT1
   ```

   プロセスごとに **MQNAME** 値を設定できます。 または、システム・レベルで、Windows レジストリー内に設定することもできます。

   固有の名前を必要とする NetBIOS 実装を使用している場合は、 IBM MQ プロセスが開始される各ウィンドウで **SET MQNAME** コマンドを発行する必要があります。 **MQNAME** 値は任意ですが、プロセスごとに固有でなければなりません。

3. キュー・マネージャー構成ファイル qm.ini の NETBIOS スタンザ。 以下に例を示します。

   ```
   NETBIOS:
   LocalName= my_station
   ```

注:

1. サポートされる NetBIOS 製品の環境のバリエーションに応じて、ネットワークでは、各 NetBIOS 名を固有にしておくことが重要です。 名前を固有にしておかないと、予測不能な結果が生じることがありま

す。 NetBIOS チャネルの確立で問題が生じ、キュー・マネージャー・エラー・ログに NetBIOS 戻りコード X'15' を示すエラー・メッセージがある場合は、NetBIOS 名の使用を検討してください。

2. Windows では、マシン名を NetBIOS 名として使用することはできません。それは、Windows が既にその名前を使用しているためです。
3. 送信側のチャネルを開始するには、NetBIOS 名が、qm.ini ファイルの中で、MQNAME 環境変数または LocalName を使用して指定されていなければなりません。

Windows キュー・マネージャーの *NetBIOS* セッション、コマンド、および名前制限の設定

NetBIOS セッション、コマンド、および名前に関するキュー・マネージャー制限は、2 つの方法で指定できます。

これらの方法を優先順位に従って示してあります。

1. RUNMQLSR コマンドで指定された値。

```
-s Sessions
-e Names
-o Commands
```

   コマンドに -m オペランドが指定されていない場合、値はデフォルトのキュー・マネージャーにのみ適用されます。

2. キュー・マネージャー構成ファイル qm.ini の中の NETBIOS スタンザ。 以下に例を示します。

```
NETBIOS:

NumSess= Qmgr_max_sess
NumCmds= Qmgr_max_cmds
NumNames= Qmgr_max_names
```

Windows *LAN* アダプター番号の設定

チャネルが NetBIOS 間で正常に動作するためには、それぞれの側のアダプター・サポートに互換性がなければなりません。 IBM MQ では、qm.ini ファイルの NETBIOS スタンザの中で AdapterNum 値を使用し、runmqlsr コマンドに **-a** パラメーターを指定することによって、LAN アダプター (LANA) 番号の選択を制御できます。

IBM MQ によって、NetBIOS 接続に使用されるデフォルトの LAN アダプター番号は 0 です。 次のように、システムで使用されている番号を確認します。

Windows 上では、オペレーティング・システムから直接 LAN アダプター番号を照会することはできません。 代わりに、Microsoft から入手可能な LANACFG.EXE コマンド行ユーティリティーを使用します。 ツールの出力は、仮想 LAN アダプター番号と、その有効なバインディングを示します。 LAN アダプター番号の詳細については、Microsoft サポート技術情報の記事 138037『*HOWTO: Use LANA Numbers in a 32-bit Environment*』を参照してください。

次のように、キュー・マネージャー構成ファイル qm.ini の NETBIOS スタンザに正しい値を指定します。

```
NETBIOS:
AdapterNum= n
```

ここで、n は、このシステムの正しい LAN アダプター番号です。

Windows *NetBIOS* 接続の開始

接続を開始するのに必要なステップを定義します。

接続を開始するには、送信側で次のステップに従ってください。

1. MQNAME または LocalName 値を使用して、NetBIOS ステーション名を定義します。
2. システムで使用している LAN アダプター番号を確認し、AdapterNum を使用して正しいファイルを指定します。

3. チャネル定義の「ConnectionName」フィールドで、ターゲット・リスナー・プログラムが使用する NetBIOS 名を指定します。 Windows では、NetBIOS チャネルをスレッドとして実行しなければなりません。 チャネル定義で MCATYPE (THREAD) を指定して、これを実行してください。

```
DEFINE CHANNEL (chname) CHLTYPE(SDR) +
TRPTYPE(NETBIOS) +
CONNAME(your_station) +
XMITQ(xmitq) +
MCATYPE(THREAD) +
REPLACE
```

Windows *NetBIOS 接続のターゲット・リスナーの定義*
NetBIOS 接続の受信側で行われるステップを定義します。

受信側では、次のステップに従ってください。

1. MQNAME または LocalName 値を使用して、NetBIOS ステーション名を定義します。
2. システムで使用している LAN アダプター番号を確認し、AdapterNum を使用して正しいファイルを指定します。
3. 次のように受信側チャネルを定義します。

```
DEFINE CHANNEL (chname) CHLTYPE(RCVR) +
TRPTYPE(NETBIOS) +
REPLACE
```

4. IBM MQ リスナー・プログラムを開始してステーションを設定し、連絡を受けられる状態にします。 以下に例を示します。

```
RUNMQLSR -t NETBIOS -l your_station [-m qmgr]
```

   このコマンドは、your_station を連絡待ちの NetBIOS ステーションとして設定します。 NetBIOS ステーション名は、NetBIOS ネットワークで固有のものでなければなりません。

最良のパフォーマンスを得るには、241 ページの『承認されたアプリケーションとしてのチャネルおよびリスナーの実行』で説明されているように、 IBM MQ リスナーをトラステッド・アプリケーションとして実行します。 承認されたアプリケーションについては、承認されたアプリケーションの制約事項を参照してください。

次のコマンドを使用して、非アクティブであるキュー・マネージャーで実行されているすべての IBM MQ リスナーを停止できます。

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

キュー・マネージャー名を指定しないと、デフォルトのキュー・マネージャーが想定されます。

Linux AIX AIX and Linux における通信のセットアップ

分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 これが成功するためには、接続が定義されていて、使用可能になっていなければなりません。 このセクションでは、IBM MQ for UNIX or Linux システムで使用可能な通信の形式を使用して、これを行う方法について説明しています。

始める前に

次のセクションを参照すると役立つ場合があります。

- AIX 構成例 - IBM MQ for AIX
- Linux 構成例 - IBM MQ for Linux

MQ Adv. CD TCP/IP を使用するメッセージ・チャネルは、ネットワーク・スループットを大幅に向上させることができる高速 TCP/IP トンネルを提供する IBM Aspera faspio Gateway を指すことができます。 資格のあるプラットフォームで実行されているキュー・マネージャーは、Aspera gateway を介

して接続できます。 ゲートウェイ自体は、 Red Hat または Ubuntu Linux、あるいは Windows にデプロイされます。 Linux または Windows での Aspera gateway 接続の定義を参照してください。

このタスクについて

分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。 このセクションでは、そのための方法を説明します。

AIX and Linux 上の IBM MQ の通信をセットアップする場合、以下のタイプの通信から選択できます。

- TCP/IP
- LU 6.2

どのチャネル定義も伝送プロトコル (Transport Type) 属性を 1 つだけ指定する必要があります。 1 つのキュー・マネージャーが 1 つまたは複数のプロトコルを使用することができます。

IBM MQ MQI clients の場合、別の伝送プロトコルを使用する代替チャネルがあると役に立つことがあります。 IBM MQ MQI clients を参照してください。

手順

AIX システムまたは Linux システムの通信のセットアップについては、選択した通信タイプのサブトピックを参照してください。

- 260 ページの『AIX and Linux での TCP 接続の定義』
- 264 ページの『AIX and Linux での LU 6.2 接続の定義』
- MQ Adv. MQ Adv. VUE 858 ページの『Linux または Windows プラットフォームでの Aspera gateway 接続の定義』

関連タスク

242 ページの『AIX, Linux, and Windows 上のチャネルのモニターおよび制御』
DQM では、リモート・キュー・マネージャーに対するチャネルの作成、モニターおよび制御が必要です。 コマンド、プログラム、IBM MQ Explorer、チャネル定義ファイル、および同期情報のストレージ域を使用して、チャネルを制御できます。

15 ページの『クライアントとサーバー間の接続の構成』
IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャネルを定義します。

252 ページの『Windows における通信のセットアップ』
分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 これが成功するためには、接続が定義されていて、使用可能になっていなければなりません。 このセクションでは、IBM MQ for Windows システムで使用可能な通信の形式を使用して、これを行う方法について説明しています。

関連資料

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

Linux AIX *AIX and Linux での TCP 接続の定義*

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

始める前に

MQ Adv. CD TCP/IP を使用するメッセージ・チャネルは、ネットワーク・スループットを大幅に向上させることができる高速 TCP/IP トンネルを提供する IBM Aspera faspio Gateway を指すことができます。 資格のあるプラットフォームで実行されているキュー・マネージャーは、Aspera gateway を介

して接続できます。 ゲートウェイ自体は、 Red Hat または Ubuntu Linux、あるいは Windows にデプロイされます。 Linux または Windows での Aspera gateway 接続の定義を参照してください。

送信側

チャネル定義の「Connection Name (接続名)」フィールドにホスト名またはターゲット・マシンの TCP アドレスを指定します。 接続されるポートのデフォルト値は 1414 です。 Internet Assigned Numbers Authority (インターネットによる番号割り当て権限) によってポート番号 1414 が IBM MQ に割り当てられます。

デフォルト以外のポート番号を使用するには、接続名フィールドを次のように変更します。

```
Connection Name REMHOST(1822)
```

ここで、REMHOST はリモート・マシンのホスト名、1822 は必要なポート番号です。 (これは受信側のリスナーが待機するポートでなければなりません。)

または、次のようにキュー・マネージャーの構成ファイル (qm.ini) にポート番号を指定して、変更することもできます。

```
TCP:
Port=1822
```

qm.ini を使用して設定する値について詳しくは、分散キューイング用の構成ファイル・スタンザを参照してください。

TCP での受信

inet デーモンである TCP/IP リスナー (inetd)、または IBM MQ リスナーのいずれかを使用することができます。

現在は、inet デーモンではなく、拡張 inet デーモン (xinetd) を使用する Linux 配布版もあります。 Linux システムで拡張 inet デーモンを使用する方法について詳しくは、例: Linux での IBM MQ クロスプラットフォーム通信のセットアップの ステップ 2 を参照してください。

関連概念

261 ページの『AIX and Linux での TCP/IP リスナーの使用』
AIX and Linux 上でチャネルを開始するには、`/etc/services` ファイルと `inetd.conf` ファイルを編集する必要があります

262 ページの『IBM MQ for Multiplatforms での TCP リスナー・バックログ・オプションの使用』
TCP では、サーバーとクライアントとの間で 3 ウェイ・ハンドシェークが実行されない限り、接続は不完全として扱われます。 これらの接続は、未解決接続要求と呼ばれます。 最大値をこれらの未解決接続要求に設定し、リスナーが要求を受け入れるのを TCP ポート上で待っている要求のバックログと考えることができます。

263 ページの『IBM MQ リスナーの使用』
IBM MQ に提供されているリスナーは、新しいチャネルをスレッドとして起動します。このリスナーを実行するには、`runmqlsr` コマンドを使用します。

264 ページの『TCP/IP SO_KEEPALIVE オプションの使用』
一部の AIX and Linux システムでは、接続がまだ利用できるかどうか検査する前の TCP の待機時間を指定できます。また、最初の検査が失敗した場合の接続の再試行回数も指定できます。 それには、カーネル調整パラメーターを使うか、またはコマンド行で指定できます。

AIX and Linux での *TCP/IP* リスナーの使用

AIX and Linux 上でチャネルを開始するには、`/etc/services` ファイルと `inetd.conf` ファイルを編集する必要があります

次の指示に従ってください。

1. `/etc/services` ファイルを編集します:

注: /etc/services ファイルを編集するには、スーパーユーザーまたは root ユーザーとしてログインする必要があります。 このポート番号は変更できますが、送信側で指定したポート番号と一致しなければなりません。

次の行をファイルに追加します。

```
MQSeries 1414/tcp
```

ここで、1414 は IBM MQ に必要なポート番号です。 ポート番号は 65535 以下でなければなりません。

2. inetd.conf ファイルに行を追加して、プログラム amqcrsta を呼び出します。ここで、*MQ_INSTALLATION_PATH* は、IBM MQ がインストールされている上位ディレクトリーを表します:

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta
[-m Queue_Man_Name]
```

これらの更新は、inetd が構成ファイルを再度読み取ったあとでアクティブになります。 そのために、ルート・ユーザー ID で次のコマンドを出します。

- AIX On AIX:

```
refresh -s inetd
```

- Linux Linux システムの場合:

```
kill -1 process_number
```

inetd が開始したリスナー・プログラムが inetd からロケールを継承している場合は、MQMDE が使用 (マージ) されずに、メッセージ・データとしてキューに書き込まれる可能性があります。 MQMDE を確実に使用するには、ロケールを正しく設定しなければなりません。 inetd によって設定されるロケールは、IBM MQ プロセスが使用する他のロケールとして選んだロケールと一致していないことがあります。 ロケールの設定は以下の方法で行ってください。

1. ロケール環境変数 LANG、LC_COLLATE、LC_CTYPE、LC_MONETARY、LC_NUMERIC、LC_TIME、および LC_MESSAGES を、他の IBM MQ プロセスが使用するロケールに設定するシェル・スクリプトを作成します。
2. 同じシェル・スクリプトで、リスナー・プログラムを呼び出します。
3. inetd.conf ファイルを変更して、リスナー・プログラムの代わりにシェル・スクリプトを呼び出すようにします。

1 つのサーバーに複数のキュー・マネージャーを存在させることができます。 各キュー・マネージャーの 2 つのファイルのそれぞれに、1 行を追加します。 以下に例を示します。

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

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

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

このようにすると、1 つの TCP ポートのキューに入る未解決の接続要求の数に制限がある場合、エラー・メッセージの生成が回避されます。 未解決の接続要求の数については、262 ページの『IBM MQ for Multiplatforms での TCP リスナー・バックログ・オプションの使用』を参照してください。

Multi *IBM MQ for Multiplatforms での TCP リスナー・バックログ・オプションの使用*

TCP では、サーバーとクライアントとの間で 3 ウェイ・ハンドシェークが実行されない限り、接続は不完全として扱われます。 これらの接続は、未解決接続要求と呼ばれます。 最大値をこれらの未解決接続要求に設定し、リスナーが要求を受け入れるのを TCP ポート上で待っている要求のバックログと考えることができます。

デフォルトのリスナー・バックログ値は、263 ページの表 23 に示すとおりです。

表 23. 1 つの TCP/IP ポートでキューに入れられる未解決の接続要求の最大数	
サーバーのプラットフォーム	**接続要求の最大数**
AIX	100
Linux	100
IBM i	255
Windows サーバー	100

バックログが 263 ページの表 23 に示された値に到達すると、TCP/IP 接続は拒否され、チャネルを開始できません。

MCA チャネルでは、結果としてチャネルが RETRY 状態になり、後から接続を再試行します。

ただし、このエラーを回避するために、qm.ini ファイルにエントリーを追加することができます:

```
TCP:
ListenerBacklog = n
```

こうすると、TCP/IP リスナーのデフォルトの最大未解決要求数 (263 ページの表 23 を参照) が上書きされます。

注 : 一部のオペレーティング・システムでは、デフォルトより大きい値がサポートされています。 必要であれば、この値を使用して接続限度に到達するのを回避できます。

backlog オプション有効にしたままリスナーを実行するには、以下のいずれかを使用します。

- runmqlsr -b コマンドを使用する。
- MQSC コマンドの **DEFINE LISTENER** を、 必要な値に設定した BACKLOG 属性を指定して使用する。

runmqlsr コマンドについては、runmqlsr を参照してください。 DEFINE LISTENER コマンドの詳細については、DEFINE LISTENER を参照してください。

関連概念

1012 ページの『z/OS での TCP リスナー・バックログ・オプションの使用』
TCP/IP で受信するときに、未解決の接続要求の最大数を設定します。 これらの未解決の要求は、リスナーが要求を受け入れるのを TCP/IP ポート上で待っている要求のバックログ と考えることができます。

Linux AIX *IBM MQ* リスナーの使用

IBM MQ に提供されているリスナーは、新しいチャネルをスレッドとして起動します。このリスナーを実行するには、runmqlsr コマンドを使用します。

以下に例を示します。

```
runmqlsr -t tcp [-m QMNAME] [-p 1822]
```

大括弧は、オプション・パラメーターを示します。デフォルト・キュー・マネージャーの場合は QMNAME は必要ありません。また、デフォルト・ポート番号 (1414) を使用する場合はポート番号は必要ありません。 ポート番号は 65535 以下でなければなりません。

最良のパフォーマンスを得るには、 241 ページの『承認されたアプリケーションとしてのチャネルおよびリスナーの実行』で説明されているように、 IBM MQ リスナーをトラステッド・アプリケーションとして実行します。 承認されたアプリケーションについては、承認されたアプリケーションの制約事項を参照してください。

次のコマンドを使用して、非アクティブであるキュー・マネージャーで実行されているすべての IBM MQ リスナーを停止できます。

```
endmqlsr [-m QMNAME]
```

キュー・マネージャー名を指定しないと、デフォルトのキュー・マネージャーが想定されます。

Linux AIX *TCP/IP SO_KEEPALIVE* オプションの使用

一部の AIX and Linux システムでは、接続がまだ利用できるかどうか検査する前の TCP の待機時間を指定できます。また、最初の検査が失敗した場合の接続の再試行回数も指定できます。 それには、カーネル調整パラメーターを使うか、またはコマンド行で指定できます。

SO_KEEPALIVE オプション (詳細は、227 ページの『チャネルの相手側がまだ使用可能であるかどうかの検査』 を参照) を使用する場合、キュー・マネージャー構成ファイル (qm.ini) に次の項目を追加する必要があります。

```
TCP:
KeepAlive=yes
```

詳細については、ご使用の AIX システムまたは Linux システムの資料を参照してください。

Linux AIX *AIX and Linux での LU 6.2 接続の定義*

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

SNA over TCP/IP の構成に関する最新情報については、次に示すオンラインの IBM 資料を参照してください: Communications Server。

2 つのシステム間で LU 6.2 の会話が確立できるように SNA を構成する必要があります。

詳細については、「*Multiplatform APPC Configuration Guide*」と次の表を参照してください。

表 *24.* ローカルの *AIX* システムまたは *Linux* システムにおけるリモート・キュー・マネージャー・プラットフォーム用の設定値

リモート・プラットフォーム	**TPNAME**	**TPPATH**
z/OS (CICS なし)	リモート・キュー・マネージャーのサイド情報の中の対応する TPName と同じ。	-
z/OS 使用 CICS	CKRC (送信側) CKSV (要求側) CKRC (サーバー)	-
IBM i	IBM i システムにおいて経路指定項目の比較値と同じ。	-
AIX and Linux システム	リモート・キュー・マネージャーのサイド情報の中の対応する TPName と同じ。	*MQ_INSTALLATION_PATH*/bin/amqcrs6a
Windows	Windows Run Listener コマンドで指定されたとおり、または Windows において TpSetup を使用して定義された呼び出し可能なトランザクション・プログラム。	*MQ_INSTALLATION_PATH*\bin\amqcrs6a

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

同一のマシン上に複数のキュー・マネージャーがある場合は、チャネル定義内の TPname をそれぞれ固有にする必要があります。

関連概念

265 ページの『AIX and Linux における LU 6.2 での送信側』
AIX and Linux システムの場合、CPI-C サイド・オブジェクト (シンボリック宛先) を作成して、その名前をチャネル定義の「Connection name (接続名)」フィールドに入れます。 また、パートナーに対する LU 6.2 リンクも作成します。

265 ページの『AIX and Linux における LU 6.2 での受信』
AIX and Linux システムでは、受信側で listen 接続機構、LU 6.2 論理接続プロファイル、TPN プロファイルを作成します。

Linux AIX *AIX and Linux における LU 6.2 での送信側*

AIX and Linux システムの場合、CPI-C サイド・オブジェクト (シンボリック宛先) を作成して、その名前をチャネル定義の「Connection name (接続名)」フィールドに入れます。 また、パートナーに対する LU 6.2 リンクも作成します。

CPI-C サイド・オブジェクトには、受信側マシンのパートナー LU 名、トランザクション・プログラム名、およびモード名を入れます。 以下に例を示します。

```
Partner LU Name              REMHOST
Remote TP Name               recv
Service Transaction Program  no
Mode Name                    #INTER
```

SECURITY PROGRAM が CPI-C にサポートされていれば、IBM MQ は SNA セッションの確立を試みるときに、それを使用します。

Linux AIX *AIX and Linux における LU 6.2 での受信*

AIX and Linux システムでは、受信側で listen 接続機構、LU 6.2 論理接続プロファイル、TPN プロファイルを作成します。

TPN プロファイルには、実行可能ファイルへの絶対パス、およびトランザクション・プログラム名を入力します。

```
Full path to TPN executable    MQ_INSTALLATION_PATH/bin/amqcrs6a
Transaction Program name       recv
User ID                        0
```

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

ユーザー ID を設定できるシステムでは、mqm グループのメンバーであるユーザーを指定します。

AIX AIX では、APPCTPN (トランザクション名) および APPCLLU (ローカル LU 名) 環境変数を設定してください (呼び出すトランザクション・プログラム用の構成パネルを使用します)。

デフォルトのキュー・マネージャー以外のキュー・マネージャーを使用しなければならないこともあります。 その場合には、次のものを呼び出すコマンド・ファイルを定義します。

```
amqcrs6a -m Queue_Man_Name
```

その後で、コマンド・ファイルを呼び出します。

IBM i IBM i 上のチャネルのモニターおよび制御

リモート・キュー・マネージャーにつながるチャネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。 各キュー・マネージャーには、互換性のあるリモート・キュー・マネージャーとの相互接続を制御するための DQM プログラムが備わっています。

このタスクについて

次のリストに、チャネル制御機能のコンポーネントについて簡単に説明します。

- チャネル定義はキュー・マネージャー・オブジェクトとして保持されます。
- チャネル・コマンドは、IBM MQ for IBM i のコマンド・セットのサブセットです。

  IBM MQ for IBM i コマンド・セット全体を表示するには、GO CMDMQM コマンドを使用します。
- チャネル定義パネルまたはコマンドは、以下の用途に使用します。
  - チャネル定義の作成、コピー、表示、変更、および削除
  - チャネルの開始と停止、ping、チャネル・シーケンス番号のリセット、およびリンクを再確立できないときの未確定メッセージの解決
  - チャネルに関する状況情報の表示
- MQSC を使用してチャネルを管理することもできます。

- IBM MQ エクスプローラーを使用してチャネルを管理することもできます。
- シーケンス番号と作業論理単位 *(LUW)* ID は同期ファイルに格納され、チャネル同期化のために使用されます。

コマンドおよびパネルを使用して、メッセージ・チャネルと関連するオブジェクトを定義し、メッセージ・チャネルをモニターおよび制御できます。 F4=プロンプト・キーを使用することにより、関係のあるキュー・マネージャーを指定できます。 プロンプトを使用しないと、デフォルトのキュー・マネージャーが想定されます。 F4=プロンプトを使用すると、追加のパネルが表示されます。そのパネルで、関係のあるキュー・マネージャーの名前と、場合によっては他のデータを入力することができます。

パネルを使用して定義するオブジェクトは、次のとおりです。

- 伝送キュー
- リモート・キュー定義
- キュー・マネージャー別名定義
- 応答先キュー別名の定義
- 応答先ローカル・キュー
- メッセージ・チャネル定義

これらのオブジェクトの使用に関連する概念の詳細については、190 ページの『分散キューイングの構成』を参照してください。

チャネルは、完全に定義されていて、それに関連するオブジェクトが存在し、使用可能になっていなければ、開始することはできません。

また、各チャネルに関する特定の通信リンクが定義されて、使用可能になっていなければ、チャネルを実行することはできません。 LU 6.2、および TCP/IP リンクの定義方法については、ご使用のシステムの特定の通信の手引きを参照してください。

手順

- オブジェクトの作成および処理についての詳細は、以下を参照してください。
  - 267 ページの『オブジェクトの作成 (IBM i)』
  - 267 ページの『チャネルの作成 (IBM i)』
  - 269 ページの『チャネルの開始 (IBM i)』
  - 269 ページの『チャネルの選択 (IBM i)』
  - 270 ページの『チャネルの表示 (IBM i)』
  - 272 ページの『チャネルの名前変更 (IBM i)』
  - 272 ページの『チャネル状況の操作 (IBM i)』
  - 273 ページの『チャネル操作の選択項目 (IBM i)』

関連概念

279 ページの『IBM i における通信のセットアップ』
分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 これを正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。

関連タスク

15 ページの『クライアントとサーバー間の接続の構成』
IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャネルを定義します。

関連資料

構成例 - IBM MQ for IBM i
IBM MQ for IBM i におけるメッセージ・チャネルの計画例
IBM MQ for IBM i CL コマンド

IBM i オブジェクトの作成 (IBM i)

CRTMQMQ コマンドを使用して、キュー・オブジェクトおよび別名オブジェクトを作成できます。

伝送キュー、リモート・キュー定義、キュー・マネージャー別名の定義、応答先キュー別名の定義、応答先ローカル・キューなどのキュー・オブジェクトおよび別名オブジェクトを作成できます。

デフォルト・オブジェクトのリストについては、システムおよびデフォルト・オブジェクトを参照してください。

IBM i チャネルの作成 (IBM i)

「チャネルの作成」パネルから、またはコマンド行で CRTMQMCHL コマンドを使用して、チャネルを作成できます。

チャネルを作成するには、次のようにします。

1. 「Work with MQM Channels (MQM チャネルの処理)」パネル (WRKMQMCHL) で F6 キーを使用します。

   あるいは、コマンド行から CRTMQMCHL コマンドを使用します。

   どちらの場合にも、「チャネルの作成」パネルが表示されます。 タイプ:

   - 表示されたフィールドにチャネルの名前を入力します。
   - リンク側のチャネル・タイプを入力します。
2. Enter キーを押します。

注 : 使用するネットワーク内のすべてのチャネルに固有の名前を付ける必要があります。 すべてのチャネルを示すネットワーク・ダイアグラムに示すように、発信元および宛先のキュー・マネージャー名をチャネル名に含める方法を推奨します。

入力した項目は検査され、エラーがある場合にはただちに報告されます。 エラーがある場合には訂正し、続行してください。

選択したチャネルのタイプに対応するチャネル設定パネルが表示されます。 あらかじめまとめておいた情報を各フィールドに入力します。 Enter キーを押してチャネルを作成します。

チャネル定義パネルの記述を行うさまざまなフィールドの内容を決める際には、ヘルプ・パネルにヘルプ情報が提供されます。また、チャネル属性を参照してください。

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Channel name . . . . . > CHANNAME_____________
Channel type . . . . . > *SDR__    *RCVR, *SDR, *SVR, *RQSTR...
Message Queue Manager name     *DFT_____________________________
_____
Replace . . . . . .  *NO_     *NO, *YES
Transport type . . . . . .  *TCP____    *LU62, *TCP, *SYSDFTCHL
Text 'description' . . . . . > 'Example Channel Definition'_____________
_____________________________
Connection name . . . . . .  *SYSDFTCHL_____________________________
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
_____________________________________________
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

図 *25.* チャネルの作成 *(1)*

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Transmission queue . . . . . .  'TRANSMISSION_QUEUE_NAME'_____________________
_____________________
Message channel agent . . . . .  *NONE_______  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________  Name
Message channel agent user ID .  *SYSDFTCHL__ Character value...
Coded Character Set Identifier   *SYSDFTCHL__ 0-9999, *SYSDFTCHL
Batch size . . . . . .  50_________  1-9999, *SYSDFTCHL
Disconnect interval . . . . . .  6000_______  1-999999, *SYSDFTCHL
Short retry interval . . . . . .  60_________  0-999999999, *SYSDFTCHL
Short retry count . . . . . .  10_________  0-999999999, *SYSDFTCHL
Long retry interval . . . . . .  1200_______  0-999999999, *SYSDFTCHL
Long retry count . . . . . .  999999999__  0-999999999, *SYSDFTCHL
Security exit . . . . . .  *NONE_____  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________  Name
Security exit user data . . . .  *SYSDFTCHL_____________________

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

図 *26. チャネルの作成 (2)*

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Send exit . . . . . .  *NONE_______  Name, *SYSDFTCHL, *NONE
Library . . . . . .  ___________ Name
+ for more values   __________
Send exit user data . . . . . .  _________________________________
+ for more values  _________________________________
Receive exit . . . . . .  *NONE_____  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________  Name
+ for more values  __________
__________
Receive exit user data . . . . .  _________________________________
+ for more values  _________________________________
Message exit . . . . . .  *NONE_____  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________  Name
+ for more values  __________
__________
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

図 *27. チャネルの作成 (3)*

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Message exit user data . . . . .  _________________________________
+ for more values  _________________________________
Convert message . . . . . .  *SYSDFTCHL_  *YES, *NO, *SYSDFTCHL
Sequence number wrap . . . . . .  999999999__  100-999999999, *SYSDFTCHL
Maximum message length . . . . .  4194304____  0-4194304, *SYSDFTCHL
Heartbeat interval . . . . . .  300________  0-999999999, *SYSDFTCHL
Non Persistent Message Speed . .  *FAST_____  *FAST, *NORMAL, *SYSDFTCHL
Password . . . . . .  *SYSDFTCHL_  Character value, *BLANK...
Task User Profile . . . . . .  *SYSDFTCHL_  Character value, *BLANK...
Transaction Program Name . . . .  *SYSDFTCHL




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

図 *28.* チャネルの作成 *(4)*

IBM i チャネルの開始 (IBM i)

「チャネルの処理」パネルから、またはコマンド行で STRMQMCHL コマンドを使用して、チャネルを開始できます。

リスナーは TCP にのみ有効です。 SNA リスナーの場合は、通信サブシステムを構成しなければなりません。

アプリケーションがメッセージ交換を行えるようにするには、STRMQMLSR コマンドを使用して、インバウンド接続に対してリスナー・プログラムを開始する必要があります。

アウトバウンド接続については、次のいずれかの方法でチャネルを開始させる必要があります。

1. MCATYPE パラメーターに基づいて、プロセスまたはスレッドとしてチャネルを開始させるには、チャネル名を指定して、CL コマンド STRMQMCHL を使用します (チャネルがスレッドとして開始されるときには、チャネルがチャネル・イニシエーターのスレッドです。)

   ```
   STRMQMCHL CHLNAME(QM1.TO.QM2) MQNAME(MYQMGR)
   ```

2. チャネル・イニシエーターを使用して、チャネルを起動します。 キュー・マネージャーが開始されると、1 つのチャネル・イニシエーターが自動的に開始されます。 この自動開始は、そのキュー・マネージャーの qm.ini ファイルの中の chinit スタンザを変更すると、除去できます。
3. WRKMQMCHL コマンドを使用して「Work with Channels (チャネルの処理)」パネルを開始し、オプション 14 を選択してチャネルを開始します。

IBM i チャネルの選択 (IBM i)

「チャネルの処理」パネルから、チャネルを選択できます。

チャネルを選択するには、WRKMQMCHL コマンドを使用して「Work with Channels (チャネルの処理)」パネルで次の手順を実行してください。

1. 必要なチャネル名に関連付けられたオプション・フィールドにカーソルを移動させます。
2. オプション番号を入力します。
3. Enter キーを押して選択項目をアクティブ化させます。

複数のチャネルを選択すると、オプションは順番にアクティブ化されます。

```
Work with MQM Channels

Queue Manager Name . . :  CNX

Type options, press Enter.
2=Change  3=Copy  4=Delete  5=Display  8=Work with Status  13=Ping
14=Start  15=End  16=Reset  17=Resolve

Opt   Name           Type    Transport   Status
CHLNIC          *RCVR    *TCP     INACTIVE
CORSAIR.TO.MUSTANG    *SDR    *LU62    INACTIVE
FV.CHANNEL.MC.DJE1    *RCVR    *TCP     INACTIVE
FV.CHANNEL.MC.DJE2    *SDR    *TCP     INACTIVE
FV.CHANNEL.MC.DJE3    *RQSTR   *TCP     INACTIVE
FV.CHANNEL.MC.DJE4    *SVR    *TCP     INACTIVE
FV.CHANNEL.PETER     *RCVR    *TCP     INACTIVE
FV.CHANNEL.PETER.LU   *RCVR    *LU62    INACTIVE
FV.CHANNEL.PETER.LU1   *RCVR    *LU62    INACTIVE
More...
Parameters or command
===>
F3=Exit  F4=Prompt  F5=Refresh  F6=Create  F9=Retrieve  F12=Cancel
F21=Print
```

図 *29.* チャネルの操作

IBM i チャネルの表示 (IBM i)

「チャネルの表示」パネルから、またはコマンド行で DSPMQMCHL コマンドを使用して、チャネルをブラウズできます。

チャネルの設定をブラウズするには、WRKMQMCHL コマンドを使用して「Display Channel (チャネルの表示)」パネルで次の手順を実行してください。

1. 必要なチャネル名に対してオプション 5 (表示) を入力します。
2. Enter キーを押して選択項目をアクティブ化させます。

複数のチャネルを選択すると、チャネルは順番に表示されます。

あるいは、コマンド行から DSPMQMCHL コマンドを使用します。

すると、そのチャネルの現行設定値の詳細を示した適切な「チャネルの表示」パネルが表示されます。 フィールドについては、チャネル属性を参照してください。

```
Display MQM Channel

Channel name . . . . . :  ST.JST.2TO1
Queue Manager Name . . . . . :  QMREL
Channel type . . . . . :  *SDR
Transport type . . . . . :  *TCP
Text 'description' . . . . . :  John's sender to WINSDOA1

Connection name . . . . . :  MUSTANG


Transmission queue . . . . . :  WINSDOA1

Message channel agent . . . . :
Library . . . . . :
Message channel agent user ID :  *NONE
Batch interval . . . . . :  0
Batch size . . . . . :  50
Disconnect interval . . . . . :  6000



F3=Exit  F12=Cancel  F21=Print
```

図 *30. TCP/IP* チャネルの表示 *(1)*

```
Display MQM Channel

Short retry interval . . . . . :  60
Short retry count . . . . . :  10
Long retry interval . . . . . :  6000
Long retry count . . . . . :  10
Security exit . . . . . :
Library . . . . . :
Security exit user data . . . :
Send exit . . . . . :
Library . . . . . :
Send exit user data . . . . . :
Receive exit . . . . . :
Library . . . . . :
Receive exit user data . . . . :
Message exit . . . . . :
Library . . . . . :
Message exit user data . . . . :
More...



F3=Exit  F12=Cancel  F21=Print
```

図 *31. TCP/IP* チャネルの表示 *(2)*

```
Display MQM Channel

Sequence number wrap . . . . . :  999999999
Maximum message length . . . . :  10000
Convert message . . . . . :  *NO
Heartbeat interval . . . . . .  300
Nonpersistent message speed . .  *FAST









Bottom



F3=Exit  F12=Cancel  F21=Print
```

図 *32. TCP/IP* チャネルの表示 *(3)*

IBM i チャネルの名前変更 (IBM i)

「チャネルの処理」パネルから、チャネルの名前を変更できます。

メッセージ・チャネルの名前を変更するためには、「Work with Channels (チャネルの処理)」パネルで次の手順を実行してください。

1. チャネルを終了させます。
2. オプション「3 (Copy (コピー))」を選び、新しい名前でチャネルの複製を作ります。
3. オプション「5 (Display (表示))」を選び、その複製チャネルが正しく作成されたか確認します。
4. オプション「4 (Delete (削除))」を選び、元のチャネルを削除します。

メッセージ・チャネルの名前を変更する際には、チャネルの両側が同時に名前変更されるようにしてください。

IBM i チャネル状況の操作 (IBM i)

「チャネル状況の処理」パネルからチャネル状況を処理できます。

WRKMQMCHST コマンドを使用すると、使用チャネルの状況を示す一連のパネルのうちの最初のパネルが表示されます。 「パネルの変更」(F11) を選択すると、状況パネルを順番に見ることができます。

あるいは、「MQM チャネルの処理」パネルからオプション「8 (状況の処理)」を選択しても、最初の状況パネルを表示することができます。

```
MQSeries Work with Channel Status

Type options, press Enter.
5=Display  13=Ping  14=Start  15=End  16=Reset  17=Resolve


Opt Name          Connection      Indoubt   Last Seq
CARTS_CORSAIR_CHAN  GBIBMIYA.WINSDOA1    NO        1
CHLNIC        9.20.2.213       NO        3
FV.CHANNEL.PETER2   9.20.2.213       NO      6225
JST.1.2        9.20.2.201       NO       28
MP_MUST_TO_CORS    9.20.2.213       NO       100
MUSTANG.TO.CORSAIR  GBIBMIYA.WINSDOA1    NO       10
MP_CORS_TO_MUST    9.20.2.213       NO       101
JST.2.3        9.5.7.126        NO       32
PF_WINSDOA1_LU62   GBIBMIYA.IYA80020    NO       54
PF_WINSDOA1_LU62   GBIBMIYA.WINSDOA1    NO       500
ST.JCW.EXIT.2TO1.CHL 9.20.2.213       NO       216

Bottom
Parameters or command
===>
F3=Exit  F4=Prompt  F5=Refresh  F6=Create  F9=Retrieve  F11=Change view
F12=Cancel  F21=Print
```

図 *33.* 一連のチャネル状況パネルの最初のパネル

「Work with Channel Status (チャネル状況の処理)」パネルで利用できるオプションは、次のとおりです。

メニュー・オプション	説明
5=Display	チャネル設定値を表示します。
13=Ping	該当する場合には、ping アクションを開始します。
14=Start	チャネルを開始します。
15=End	チャネルを停止します。
16=Reset	チャネル・シーケンス番号をリセットします。
17=Resolve	未確定チャネル状況を手動で解決します。

IBM i チャネル操作の選択項目 (IBM i)

「Work with Channels (チャネルの処理)」パネルは、WRKMQMCHL コマンドによって表示でき、これを使用して、リストされたすべてのチャネルの状況をモニターしたり、選択したチャネルに対してコマンドを実行したりできます。

「Work with Channels (チャネルの処理)」パネルで利用できるオプションは、次のとおりです。

メニュー・オプション	説明
274 ページの『2=Change』	チャネルの属性を変更します。
274 ページの『3=Copy』	あるチャネルの属性を新しいチャネルにコピーします。
274 ページの『4=Delete』	チャネルを削除します。
274 ページの『5=Display』	チャネルの現行設定値を表示します。
274 ページの『6=Create』	「チャネルの作成」パネルの表示
275 ページの『8=Work with Status』	チャネル状況パネルを表示します。
276 ページの『13=Ping』	ping 機能を実行し、リモート側と固定データ・メッセージを交換して、隣接システムとの接続をテストします。

メニュー・オプション	説明
276 ページの『14=Start』	選択されたチャネルを開始するか、あるいは使用不可の受信側チャネルをリセットします。
277 ページの『15=End』	チャネルの終了を要求します。
278 ページの『16=Reset』	リンク先でシーケンス番号をリセットするようにチャネルに要求します。 シーケンス番号は、チャネルの両側で同じになっていなければ開始できません。
278 ページの『17=Resolve』	反対側の接続を確立しないで未確定メッセージを解決するようにチャネルに要求します。
278 ページの『18=Display authority』	IBM MQ オブジェクト権限を表示します。
278 ページの『19=Grant authority』	IBM MQ オブジェクト権限を付与します。
279 ページの『20=Revoke authority』	IBM MQ オブジェクト権限を取り消します。
279 ページの『21=Recover object』	IBM MQ オブジェクトをリカバリーします。
279 ページの『22=Record image』	IBM MQ オブジェクト・イメージを記録します。

IBM i 2=Change

「Change (変更)」オプションは、既存のチャネル定義を変更するときに使用します。

「Change (変更)」オプションまたは CHGMQMCHL コマンドは、チャネル名以外の既存のチャネル定義を変更します。 チャネル定義パネル内で変更したいフィールドを上書きし、Enter キーを押して更新済みの定義を保存してください。

IBM i 3=Copy

「Copy (コピー)」オプションは、既存のチャネルをコピーするときに使用します。

「Copy (コピー)」オプションは CPYMQMCHL コマンドを使用して、既存のチャネルをコピーします。「Copy (コピー)」パネルを使用すると、新しいチャネル名を定義できます。 ただし、IBM i オブジェクト名として有効な文字だけを使用するようにしてください。 IBM MQ for IBM i の管理を参照してください。

「Copy (コピー)」パネルで Enter キーを押して、現行設定値の詳細を表示してください。 任意の新規チャネル設定値を変更できます。 Enter キーを押して、新しいチャネル定義を保管してください。

IBM i 4=Delete

「Delete (削除)」オプションは、特定のチャネルを削除するために使用します。

要求の確認または取り消しを行うためのパネルが表示されます。

IBM i 5=Display

「Display (表示)」オプションは、チャネルの現行定義を表示するために使用します。

これを選択すると、フィールドにパラメーターの現行値が入っているパネルが表示されます。これらのフィールドは、ユーザー入力が行われないように保護されています。

IBM i 6=Create

「Create (作成)」オプションは、「Create channel (チャネルの作成)」パネルを表示するときに使用します。

「Create (作成)」オプションを使用するか、またはコマンド行で CRTMQMCHL コマンドを入力して、「Create Channel (チャネルの作成)」パネルを表示します。 267 ページの図 25 以降には、「Create Channel (チャネルの作成)」パネルの例があります。

このパネルを使用すると、IBM MQ for IBM i によって提供されるデフォルト値がフィールドに入った画面に基づいて、チャネル定義を作成できます。 チャネルの名前を入力し、作成するチャネルのタイプおよび使用する通信方式を入力してください。

Enter キーを押すと、パネルが表示されます。 このパネルとその他のパネル内の必要なすべてのフィールドに情報を入力してから、Enter キーを押して定義を保存します。

チャネル名はチャネルの両端で同じ名前になっていなければならず、またネットワーク内で固有の名前でなければなりません。 ただし、IBM MQ for IBM i オブジェクト名として有効な文字だけを使用するようにしてください。

どのパネルでも、いくつかのフィールドには IBM MQ for IBM i によってデフォルト値が提供されます。 これらの値をカスタマイズしたり、またはチャネルの作成もしくはコピー時に値を変更したりできます。 値のカスタマイズについては、「*IBM MQ for IBM i* システム管理ガイド」を参照してください。

独自のチャネル・デフォルト値のセットを作成するには、各チャネル・タイプに必要なデフォルト値を指定したダミー・チャネルを設定しておいて、新しいチャネル定義の作成が必要になるたびにそれらのチャネルをコピーすることができます。

関連資料
チャネルの属性

IBM i *8=Work with Status*

「状況の処理」は、詳細なチャネルの状況情報を表示するときに使用します。

「status (状況)」列は、チャネルがアクティブ、または非アクティブかを示し、「Work with MQM channels (MQM チャネルの処理)」パネルでも引き続き表示されます。 さらに状況情報を表示するには、オプション「8 (Work with Status (状況の処理))」を選択します。 あるいは、コマンド行から WRKMQMCHST コマンドを使用してこの情報を表示させることもできます。 272 ページの『チャネル状況の操作 (IBM i)』 を参照してください。

- チャネル名
- チャネル・タイプ
- チャネル状況
- チャネル・インスタンス
- リモート・キュー・マネージャー
- 伝送キュー名
- 通信接続名
- チャネルの未確定状況
- 最終シーケンス番号
- 未確定メッセージ数
- 未確定シーケンス番号
- 伝送キュー上のメッセージ数
- 論理作業単位 ID
- 未確定作業論理単位 ID
- チャネル副状態
- チャネル・モニター
- ヘッダー圧縮
- メッセージ圧縮
- 圧縮時間標識
- 圧縮率標識
- 伝送キュー時間標識
- ネットワーク時間標識

- 終了時間標識
- バッチ・サイズ標識
- 現行共有会話
- 最大共有会話

IBM i *13=Ping*

固定データ・メッセージをリモート側に交換するには、Ping オプションを使用します。

IBM MQ の Ping に成功すると、システム監視プログラムは、チャネルが使用可能であって機能していることをある程度確認することができます。

Ping を選択しても、伝送キューとターゲット・キューが使用されることはありません。 チャネル定義、関連する通信リンク、およびネットワーク設定が使用されます。

Ping は、送信側チャネル、およびサーバー・チャネルからのみ使用できます。 リンクの反対側で対応チャネルが開始され、開始パラメーターの折衝が実施されます。 エラーがある場合には、通常どおり通知されます。

メッセージ交換の結果は Ping パネルに表示されます。これには、戻されたメッセージ・テキスト、メッセージの送信時刻、および応答の受信時刻も表示されます。

LU 6.2 での ping

IBM MQ for IBM i で ping が呼び出されるときには、その機能を要求したユーザーのユーザー ID で実行されます。これに対し、チャネル・プログラムの通常の実行方法では、QMQM ユーザー ID がチャネル・プログラム用に使用されます。 ユーザー ID は受信側まで流れるため、これが受信側で有効でなければ LU 6.2 の会話を割り振ることはできません。

IBM i *14=Start*

「Start (開始)」オプションは、チャネルを手動で開始するときに使用します。

「Start (開始)」オプションは送信側、サーバー、および要求側のチャネルで利用できます。 これは、チャネルがキュー・マネージャーによるトリガー操作でセットアップされている場合には必要ありません。

「Start (開始)」オプションは、受信側、サーバー接続、クラスター送信側、およびクラスター受信側の各チャネルでも使用されます。 STOPPED 状態になっている受信側チャネルの開始は、リモート・チャネルからそのチャネルを開始できることを意味しています。

送信側 MCA は、開始されると、チャネル定義ファイルを読み取り、伝送キューをオープンします。 チャネル開始シーケンスが実行され、リモートで受信側またはサーバー側のチャネルの対応する MCA が開始されます。 それらのチャネルが開始すると、送信側プロセスおよびサーバー・プロセスはメッセージが伝送キューに到達するのを待ち、到達したメッセージを伝送します。

トリガー操作を使用する場合には、絶えず実行されているトリガー・プロセスを開始して、開始キューをモニターする必要があります。 プロセスを開始するために、STRMQMCHLI コマンドを使用します。

チャネルの反対側では、送信側からのチャネル開始に応答して受信側プロセスが開始することがあります。そのために使用される方式は、以下のように、LU 6.2 接続チャネルと TCP/IP 接続チャネルとでは異なります。

- LU 6.2 接続チャネルの場合、チャネルの受信側では明示的なアクションは不要です。
- TCP 接続チャネルの場合は、リスナー・プロセスが絶えず実行されている必要があります。 このプロセスはリンクのリモート側からのチャネル開始要求を待ち、その接続に関してチャネル定義で定義されたプロセスを開始します。

  リモート・システムが IBM i の場合には、STRMQMLSR コマンドを使用できます。

「Start (開始)」オプションを使用すると、必要な場合には必ずチャネルが再同期化されます。

開始手続きを正常に行うためには、次の条件が必要です。

- ローカルおよびリモートのチャネル定義が存在している。 受信側またはサーバー接続のチャネルに該当するチャネル定義がない場合は、チャネルが自動定義されていると、自動的にデフォルトのチャネル定義が作成されます。 チャネル自動定義出口プログラムを参照してください。
- 伝送キューが存在していて、読み取りのために使用することが可能になっており、しかも他のチャネルで使用されていない。
- ローカルおよびリモートの MCA が存在している。
- 通信リンクが利用可能になっている。
- ローカルおよびリモートのキュー・マネージャーが実行されている。
- メッセージ・チャネルが非アクティブになっている。

メッセージを転送するには、リモート・キューとリモート・キュー定義が存在していなければなりません。

チャネル開始要求が受け入れられたことを示すメッセージがパネルに表示されます。 開始プロセスが成功したかどうかを確認するためには、システム・ログを調べるか、あるいは F5 (画面最新表示) キーを押してください。

IBM i ***15=End***

「End (終了)」は、チャネル活動を停止するために使用します。

「End (終了)」オプションは、チャネル活動の停止を要求するために使用します。 チャネルは、これ以上メッセージを送信しません。

Enter キーを押す前に F4 を選択し、チャネルが STOPPED と INACTIVE のどちらになるか、さらに CONTROLLED と IMMEDIATE のどちらの停止を使用してチャネルを停止するかを選択します。 停止状態のチャネルを再度アクティブにするは、オペレーターが再開する必要があります。 非アクティブのチャネルはトリガーできます。

即時停止

「即時停止」を使用すると、 作業単位がある場合でもそれを完了せずにチャネルを停止します。

このオプションは、チャネル・プロセスを終了してしまいます。 チャネルはメッセージの現行バッチの処理を完了しないため、そのチャネルを未確定のままにしておくことはできません。 一般に、オペレーターは制御された停止オプションを使用する方が良いです。

制御された停止

「制御された停止」を使用すると、現在の作業単位が終わったらチャネルを停止します。

これを選択すると、チャネルが正常にクローズし、メッセージの現行バッチが完了するとともに、チャネルの反対側の端点で同期点手順が実行されます。

停止したチャネルの再始動

チャネルが STOPPED 状態になるときは、手動でチャネルを再始動しなければなりません。 以下の方法でチャネルを再始動できます。

- **START CHANNEL** MQSC コマンドを使用します。
- **Start Channel** PCF コマンドを使用します。
- IBM MQ Explorer を使用します。
- z/OS z/OS で、「チャネルの開始 (Start a channel)」パネルを使用します。
- IBM i IBM i で、**STRMQMCHL CL** コマンドまたは WRKMQMCHL パネルの **START** オプションを使用します。

送信側またはサーバー・チャネルでは、チャネルが停止状態になった場合、関連する伝送キューは GET(DISABLED) に設定され、トリガー操作はオフに設定されます。 開始要求が受信されたとき、これらの属性は自動的にリセットされます。

z/OS チャネルが RETRYING または STOPPED の状況の間にチャネル・イニシエーターが停止した場合、チャネル・イニシエーターの再開始時にチャネル状況が記憶されています。 ただし、SVRCONN チャネル・タイプのチャネル状況は、チャネルが STOPPED の状況の間にチャネル・イニシエーターが停止すると再設定されます。

Multi チャネルが RETRYING または STOPPED の状況の間にキュー・マネージャーが停止した場合、キュー・マネージャーの再開始時にチャネル状況が記憶されています。 IBM MQ 8.0 以降、これは SVRCONN チャネルにも当てはまります。 以前は、SVRCONN チャネル・タイプのチャネル状況は、チャネルが STOPPED の状況の間にチャネル・イニシエーターが停止するとリセットされていました。

IBM i *16=Reset*

「Reset (リセット)」オプションは、メッセージ・シーケンスを強制的に新しくするときに使用します。

「Reset (リセット)」オプションは、メッセージ・シーケンス番号を変更するために使用します。 このオプションを使用する際には注意が必要です。未確定状態がある場合は、必ず、「Resolve (解決)」オプションを使用してそれを解決してからこのオプションを使用するようにしてください。 このオプションは、送信側チャネルまたはサーバー・チャネルだけで使用可能です。 次にそのチャネルが開始されると、最初のメッセージは新しいシーケンス番号になります。

IBM i *17=Resolve*

「Resolve (解決)」オプションは、伝送キューに保持される未確定メッセージのローカル側でのコミットまたはバックアウトを強制するときに使用します。

リンクの送信側かサーバー側が終了してしまいリカバリーの見込みがないなどの理由で、メッセージがいずれかの側で未確定のままの場合は、「(Resolve (解決)」オプションを使用します。 「Resolve (解決)」オプションには、BACKOUT か COMMIT のどちらかのパラメーターを設定できます。 バックアウトではメッセージは伝送キューに復元され、コミットではメッセージは廃棄されます。

チャネル・プログラムは、パートナーとのセッションの確立を試みません。 その代わりに、未確定メッセージを表す作業論理単位 ID (LUWID) を判別します。 そのあとで、要求に応じて次のいずれかを発行します。

- メッセージを伝送キューに復元するための BACKOUT
- メッセージを伝送キューから削除するための COMMIT

解決を正常に行うためには、次の条件が必要です。

- チャネルが非アクティブになっている。
- チャネルが未確定になっている。
- チャネル・タイプが送信側またはサーバーである。
- ローカルのチャネル定義が存在している。
- ローカルのキュー・マネージャーが実行されている。

IBM i *18=Display authority*

「Display authority (権限の表示)」オプションは、特定の IBM MQ オブジェクトに対してユーザーがどのアクションの実行を許可されているかを表示するときに使用します。

DSPMQAUT コマンドは、選択したオブジェクトおよびユーザーについて、IBM MQ オブジェクトに対してアクションを実行するための、ユーザーが持つ権限を表示します。 そのユーザーが複数のグループのメンバーである場合は、そのオブジェクトへのすべてのグループの権限を組み合わせたものがこのコマンドで表示されます。

IBM i *19=Grant authority*

「Grant authority (権限の付与)」オプションは、IBM MQ オブジェクトに対するアクションを実行する権限を別のユーザーまたはユーザー・グループに付与するときに使用します。

GRTMQMAUT コマンドは、QMQMADM グループのユーザーのみに使用できます。 QMQMADM のユーザーは、ユーザーを名前で指定するか、または *PUBLIC のすべてのユーザーに権限を付与するかのいずれかに

より、コマンドで指定された IBM MQ オブジェクトに対してアクションを実行する権限を他のユーザーに付与します。

IBM i *20=Revoke authority*

「Revoke authority (権限の取り消し)」は、オブジェクトに対するアクション実行の許可をユーザーから除去するときに使用します。

RVKMQMAUT コマンドは、QMQMADM グループのユーザーのみに使用できます。 QMQMADM グループのユーザーは、ユーザーを名前で指定するか、または *PUBLIC のすべてのユーザーから権限を取り消すかのいずれかにより、コマンドで指定された IBM MQ オブジェクトに対してアクションを実行する権限を他のユーザーから除去します。

IBM i *21=Recover object*

「Recover object (オブジェクトのリカバリー)」は、損傷したオブジェクトを、IBM MQ ジャーナルに保管されている情報を使用してリカバリーするときに使用します。

「Recover object (オブジェクトのリカバリー)」では、「MQ オブジェクトの再作成」コマンド (RCRMQMOBJ) を使用して、コマンドで指定された損傷したすべてのオブジェクトをリカバリーします。 オブジェクトが損傷していなければ、そのオブジェクトに対してはアクションは実行されません。

IBM i *22=Record image*

「Record image (イメージの記録)」は、一連のオブジェクトのリカバリーのために必要なジャーナル・レシーバーの数を削減するため、さらにリカバリー時間を最小化するために使用します。

RCDMQMIMG コマンドは、コマンドで選択されたすべてのオブジェクトでチェックポイントを取ります。 このコマンドは、統合ファイル・システム (IFS) 内のオブジェクトの現行値を、ジャーナル・レシーバーに記録された MQPUT や MQGET などの、そのオブジェクトに関する後の情報と同期化します。

コマンドが完了すると、IFS 内のオブジェクトは最新になり、このオブジェクトをリカバリーするためのそれらジャーナル・レシーバーは、もはや必要なくなります。 切断されたジャーナル・レシーバーは切り離すことができます (これらがその他のオブジェクトをリカバリーするために必要である場合を除く)。

IBM i IBM i における通信のセットアップ

分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 これを正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。

DQM は IBM MQ for IBM i 用のリモート・キューイング機能です。 これは IBM MQ for IBM i キュー・マネージャーにチャネル制御プログラムを提供し、このプログラムによって、通信リンクに対する、システム・オペレーターからの制御が可能なインターフェースが形成されます。

分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 これを正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。 このセクションでは、確実に接続が定義され、使用可能になる方法について説明します。

チャネルを開始する前に、このセクションの説明に従って伝送キューを定義して、メッセージ・チャネル定義に組み込んでおく必要があります。

IBM MQ for IBM i システム間の通信形式は次の 2 つから選択できます。

- 280 ページの『IBM i での TCP 接続の定義』

  TCP の場合、ホスト・アドレスを使用することができ、これらの接続は「*IBM i Communication Configuration Reference*」で説明されているようにセットアップされます。

  TCP 環境では、各分散サービスには、そのサービスにアクセスするためにリモートのマシンで使用される可能性がある、固有の TCP アドレスが割り当てられます。 TCP アドレスは、ホストの名前/番号およびポート番号で構成されます。 すべてのキュー・マネージャーは、この番号を使用して、TCP を介して相互に通信します。

- 281 ページの『TCP での受信』

  この通信を行うには、ローカル・キュー・マネージャーを備えた IBM i システムとリモート・キュー・マネージャーを備えたシステムとの間の物理リンクを使用できるように、IBM i SNA 論理装置タイプ 6.2

(LU 6.2) を定義しなければなりません。 IBM i での通信の構成について詳しくは、「*IBM i Communication Configuration Reference* 」を参照してください。

必要に応じて、必要なプロセスとキューを定義してトリガーの準備も整えておく必要があります。

MQ Adv. CD TCP/IP を使用するメッセージ・チャネルは、ネットワーク・スループットを大幅に向上させることができる高速 TCP/IP トンネルを提供する IBM Aspera faspio Gateway を指すことができます。 資格のあるプラットフォームで実行されているキュー・マネージャーは、Aspera gateway を介して接続できます。 ゲートウェイ自体は、 Red Hat または Ubuntu Linux、あるいは Windows にデプロイされます。 Linux または Windows での Aspera gateway 接続の定義を参照してください。

関連タスク

265 ページの『IBM i 上のチャネルのモニターおよび制御』
リモート・キュー・マネージャーにつながるチャネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。 各キュー・マネージャーには、互換性のあるリモート・キュー・マネージャーとの相互接続を制御するための DQM プログラムが備わっています。

関連資料

構成例 - IBM MQ for IBM i

IBM MQ for IBM i におけるメッセージ・チャネルの計画例

IBM i における相互通信ジョブ

IBM i におけるチャネルの状態

IBM i *IBM i での TCP 接続の定義*

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

チャネル定義には CONNECTION NAME フィールドがあり、ターゲットの TCP ネットワーク・アドレスか、ホスト名 (例えば ABCHOST) が含まれています。 TCP ネットワーク・アドレスは、IPv4 小数点付き 10 進数形式 (例えば、127.0.0.1) または IPv6 16 進数形式 (例えば、2001:DB8:0:0:0:0:0:0) になります。 CONNECTION NAME がホスト名またはネーム・サーバーである場合、ホスト名を TCP ホスト・アドレスに変換するために、IBM i ホスト・テーブルが使用されます。

TCP アドレスを完成させるには、ポート番号が必要です。 ポート番号を指定しなければ、デフォルト・ポート番号 1414 が使用されます。 接続の開始側 (送信側、要求側、およびサーバーのチャネル・タイプ) では、例えば、次のようにオプションの接続用ポート番号を指定できます。

```
Connection name 127.0.0.1 (1555)
```

この場合、接続の開始側ではポート 1555 で受信プログラムに接続しようとします。

MQ Adv. CD TCP/IP を使用するメッセージ・チャネルは、ネットワーク・スループットを大幅に向上させることができる高速 TCP/IP トンネルを提供する IBM Aspera faspio Gateway を指すことができます。 資格のあるプラットフォームで実行されているキュー・マネージャーは、Aspera gateway を介して接続できます。 ゲートウェイ自体は、 Red Hat または Ubuntu Linux、あるいは Windows にデプロイされます。 Linux または Windows での Aspera gateway 接続の定義を参照してください。

TCP リスナー・バックログ・オプションの使用

TCP では、サーバーとクライアントとの間で 3 ウェイ・ハンドシェークが実行されない限り、接続は不完全として扱われます。 これらの接続は、未解決接続要求と呼ばれます。 最大値をこれらの未解決接続要求に設定し、リスナーが要求を受け入れるのを TCP ポート上で待っている要求のバックログと考えることができます。

詳細については、262 ページの『IBM MQ for Multiplatforms での TCP リスナー・バックログ・オプションの使用』 を参照してください。また、 IBM i の特定の値を参照してください。

関連概念

281 ページの『TCP での受信』
受信側のチャネル・プログラムは、送信側チャネルからの開始要求に応じて開始されます。 開始要求に応答するには、リスナー・プログラムを開始して着信ネットワーク要求を検出し、関連のチャネルを開始する必要があります。 このリスナー・プログラムは、STRMQMLSR コマンドで開始してください。

IBM i *TCP* での受信

受信側のチャネル・プログラムは、送信側チャネルからの開始要求に応じて開始されます。 開始要求に応答するには、リスナー・プログラムを開始して着信ネットワーク要求を検出し、関連のチャネルを開始する必要があります。 このリスナー・プログラムは、STRMQMLSR コマンドで開始してください。

キュー・マネージャーごとに複数のリスナーを開始することができます。 デフォルトでは、STRMQMLSR コマンドはポート 1414 を使用しますが、この値は指定変更できます。 デフォルト設定を指定変更するには、選択したキュー・マネージャーの qm.ini ファイルに以下のステートメントを追加します。 次の例では、リスナーはポート 2500 を使用する必要があります。

```
TCP:
Port=2500
```

qm.ini ファイルは次の IFS ディレクトリーにあります。/QIBM/UserData/mqm/qmgrs/*queue manager name*

この新しい値は、TCP リスナー・プログラムが開始するときにだけ読み取られます。 リスナーがすでに開始されている場合、この変更はそのプログラムからは見えません。 新しい値を使用するためには、リスナーを停止して STRMQMLSR コマンドを実行し直します。 現時点では、STRMQMLSR コマンドを使用した場合、リスナーは常に新しいポートにデフォルト解釈されます。

また、STRMQMLSR コマンドに異なるポート番号を指定することもできます。 以下に例を示します。

```
STRMQMLSR MQMNAME( queue manager name ) PORT(2500)
```

このように変更すると、リスナー・ジョブが続く限りリスナーのデフォルト値は新しいポートになります。

TCP SO_KEEPALIVE オプションの使用

SO_KEEPALIVE オプション (詳細は、227 ページの『チャネルの相手側がまだ使用可能であるかどうかの検査』を参照) を使用するためには、キュー・マネージャー構成ファイル (IFS ディレクトリー /QIBM/UserData/mqm/qmgrs/*queue manager name* にある qm.ini) に次の項目を追加する必要があります。

```
TCP:
KeepAlive=yes
```

そのあと、次のコマンドを実行する必要があります。

```
CFGTCP
```

オプション「3 (Change TCP Attributes (TCP 属性の変更))」を選択します。 ここで、時間間隔を分単位で指定できます。 値は、1 から 40320 分の範囲内で指定できます。デフォルトは 120 です。

TCP リスナー・バックログ・オプションの使用

TCP 上で受信する場合、未解決の接続要求の最大数が設定されます。 この数は、リスナーが要求を受け入れるのを TCP ポート上で待っている要求のバックログ と考えることができます。

IBM i でのデフォルト・リスナー・バックログ値は 255 です。 バックログがこの値に到達すると、TCP 接続は拒否され、チャネルを開始できません。

MCA チャネルの場合は、チャネルが RETRY 状態になり、後で接続が再試行されます。

クライアント接続の場合は、クライアントは、MQCONN から理由コード MQRC_Q_MGR_NOT_AVAILABLE を受け取り、後で接続を再試行することができます。

しかし、このエラーを回避するため、qm.ini ファイルに次の項目を追加できます。

```
ListenerBacklog = n
```

この項目は、TCP リスナーのデフォルトの最大未解決要求数 (255) を上書きします。

注 : 一部のオペレーティング・システムでは、デフォルトより大きい値がサポートされています。 必要であれば、この値を使用して接続限度に到達するのを回避できます。

IBM i *IBM i での LU 6.2 接続の定義*

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

リンクが開始される側では、この CSI オブジェクトを完成するための経路指定項目の定義が必要です。 リモート LU 6.2 システムからの作業要求の管理についての詳細は、「*IBM i Programming: Work Management Guide*」に記載されています。

詳細については、「*Multiplatform APPC Configuration Guide*」と次の表を参照してください。

表 *25.* ローカル *IBM i* システムにおけるリモート・キュー・マネージャー・プラットフォーム用の設定値

リモート・プラットフォーム	**TPNAME**
z/OS または MVS	リモート・キュー・マネージャーの対応するサイド情報内と同じ。
IBM i	IBM i システムにおいて経路指定項目の比較値と同じ。
AIX and Linux システム	リモート LU 6.2 構成で定義された呼び出し可能なトランザクション・プログラム。
Windows	Windows Run Listener コマンドで指定されたとおり、または Windows において TpSetup を使用して定義された呼び出し可能なトランザクション・プログラム。

同一のコンピューター上に複数のキュー・マネージャーがある場合は、チャネル定義内の TPname をそれぞれ固有にする必要があります。

関連概念

282 ページの『開始される側 (送信側)』
トランスポート・タイプ *LU62 のチャネルを定義するには、CRTMQMCHL コマンドを使用してください。

285 ページの『開始される側 (受信側)』
トランスポート・タイプ *LU62 のメッセージ・チャネル・リンクの受信側を定義するには、CRTMQMCHL コマンドを使用してください。

IBM i 開始される側 (送信側)

トランスポート・タイプ *LU62 のチャネルを定義するには、CRTMQMCHL コマンドを使用してください。

IBM MQ for IBM i V5.3 以降ではオプションとして、CSI オブジェクトを使用できます。

開始される側のパネルを、図 LU 6.2 通信セットアップ・パネル - 開始側に示しています。 図示されているパネルの全体を確認するには、最初のパネルで F10 を押してください。

```
Create Comm Side Information (CRTCSI)

Type choices, press Enter.

Side information . . . . . > WINSDOA1   Name
Library . . . . . >  QSYS    Name, *CURLIB
Remote location . . . . . > WINSDOA1   Name
Transaction program . . . . . > MQSERIES

Text 'description' . . . . . .  *BLANK


Additional Parameters

Device . . . . . .  *LOC     Name, *LOC
Local location . . . . . .  *LOC     Name, *LOC, *NETATR
Mode . . . . . .  JSTMOD92   Name, *NETATR
Remote network identifier . . .  *LOC     Name, *LOC, *NETATR, *NONE
Authority . . . . . .  *LIBCRTAUT  Name, *LIBCRTAUT, *CHANGE...

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

図 *34. LU 6.2* 通信セットアップ - 開始側

次のようにして、開始側のフィールドを完成させてください。

サイド情報
作成するサイド情報オブジェクトを保管するために使用する名前 (例えば、WINSDOA1) を、この定義に付けてください。

注 : LU 6.2 の場合、メッセージ・チャネル定義と通信接続の間のリンクは、送信側のメッセージ・チャネル定義における「**Connection name (接続名)**」フィールドになります。 このフィールドには、CSI オブジェクトの名前が入ります。

ライブラリー
この定義が保管されるライブラリーの名前です。

CSI オブジェクトは、メッセージ・チャネルをサービスするプログラムからアクセスできるライブラリー (例えば、QSYS、QMQM、および QGPL) で使用可能でなければなりません。

この 名前が誤っていたり、欠落していたり、検出できなかったりする場合には、 チャネル開始時にエラーが発生します。

リモート・ロケーション (Remote location)
プログラムが通信するリモート・ロケーションの名前を指定してください。

この必須パラメーターには、2 つのシステム間の通信リンクに使用される装置記述で定義された、リモート・システムのパートナーの論理装置名が入ります。

「**Remote location (リモート・ロケーション)**」名は、リモート・システムで DSPNETA コマンドを実行し、デフォルト・ローカル・ロケーション名を調べることによって得られます。

トランザクション・プログラム (Transaction program)
開始するリモート・システムのトランザクション・プログラムの名前を指定します (64 文字まで)。 トランザクション・プロセス名、プログラム名、チャネル名、または経路指定項目の「**Compare value (値の比較)**」に一致する文字ストリングを指定することができます。

このパラメーターは必須です。

注 : SNA サービス・トランザクション・プログラム名を指定するには、サービス・トランザクション・プログラム名を 16 進表記で入力してください。 例えば、16 進表記で 21F0F0F1 となるサービス・トランザクション・プログラム名を指定する場合には、X'21F0F0F1' と入力します。

SNA サービス・トランザクション・プログラム名の詳細については、 LU タイプ 6.2 用の「*SNA Transaction Programmer's Reference*」に 記載されています。

受信側が別の IBM i システムである場合、送信側の CSI オブジェクトと受信側の経路指定項目を突き合わせるために「**Transaction program (トランザクション・プログラム)**」名が使用されます。 この名前は、宛先 IBM i システム上のキュー・マネージャーごとに固有でなければなりません。 開始される側 (受信側) の「**呼び出すプログラム**」パラメーターを参照してください。「経路指定項目の追加」パネルの「**比較データ: 値の比較**」パラメーターも参照してください。

テキスト記述 (Text description)
この接続の使用目的を記録しておくための記述です (50 文字まで)。

デバイス
リモート・システム用の装置記述の名前を指定します。 指定できる値は以下のとおりです。

***LOC**
この装置はシステムによって決められます。

装置名 (Device-name)
リモート・ロケーションと関連付けられる装置の名前を指定します。

ローカル・ロケーション (Local location)
ローカル・ロケーション名を指定します。 指定できる値は以下のとおりです。

***LOC**
ローカル・ロケーション名はシステムによって決定されます。

***NETATR**
システム・ネットワーク属性で指定した LCLLOCNAME 値が使用されます。

ローカル・ロケーション名 (Local-location-name)
ローカル・ロケーションの名前を指定します。 リモート・ロケーションとして特定のロケーション名を指示したい場合には、ローカル・ロケーション名を指定してください。 ロケーション名は、DSPNETA コマンドを使用して調べることができます。

Mode
セッションの制御に使用するモードを指定します。 この名前は、共通プログラミング・インターフェース (CPI) - 通信 Mode_Name と同じです。 指定できる値は以下のとおりです。

***NETATR**
ネットワーク属性で指定されたモードが使用されます。

BLANK
8 つのブランク文字が使用されます。

モード名
リモート・ロケーションのモード名を指定します。

注 : 通信セッションの伝送優先順位はモードによって決まるため、送られるメッセージの優先順位に基づいて、 例えば MQMODE_HI、MQMODE_MED、MQMODE_LOW のように、さまざまなモードを定義しておくと便利な場合があります (同じロケーションを指す複数の CSI を持つことができます)。

リモート・ネットワーク ID (Remote network identifier)
リモート・ロケーションで使用されるリモート・ネットワーク ID を指定します。 指定できる値は以下のとおりです。

***LOC**
リモート・ロケーションのリモート・ネットワーク ID が使用されます。

***NETATR**
ネットワーク属性で指定されたリモート・ネットワーク ID が使用されます。

***NONE 値**
このリモート・ネットワークには名前がありません。

リモート・ネットワーク ID (Remote-network-id)
リモート・ネットワーク ID を指定します。 このネットワーク ID を調べるには、リモート・ロケーションで DSPNETA コマンドを使用してください。 これはリモート・ロケーションの「ローカル・ネットワーク ID」です。

権限
このオブジェクトに対する特別な権限をもたないユーザー、 許可リストに記載されていないユーザー、およびグループ・プロファイルでそのオブジェクトに対する特別な権限が指定されていないユーザーに対して与える権限を指定します。 指定できる値は以下のとおりです。

***LIBCRTAUT**
このオブジェクトに関する共通権限は、指定されたライブラリーの CRTAUT パラメーターから得られます。 この値は作成時に決定されます。 オブジェクトの作成後にライブラリーに関する CRTAUT 値が変更された場合には、新しい値は既存のオブジェクトに影響を与えません。

***CHANGE**
変更権限を与えられたユーザーは、そのオブジェクトに対して基本機能を実行できます。ただし、ユーザーがオブジェクトを変更することはできません。 変更権限は、オブジェクト操作権限およびすべてのデータ権限が与えられます。

***ALL**
ユーザーは、所有者に限定されている操作や権限リスト管理権限によって制御される操作を除き、すべての操作を実行できます。 ユーザーは、そのオブジェクトの存在を制御したり、オブジェクトのセキュリティーを指定したり、オブジェクトを変更したり、 オブジェクトに対する基本機能を実行したりすることができます。 また、オブジェクトの所有権を変更できます。

***USE**
使用権限は、オブジェクト操作権限と読み取り権限を提供します。

***EXCLUDE**
除外権限は、そのユーザーがオブジェクトにアクセスできないようにします。

許可リスト (Authorization-list)
サイド情報として使用する権限の入った許可リストの名前を指定します。

IBM i **開始される側 (受信側)**
トランスポート・タイプ *LU62 のメッセージ・チャネル・リンクの受信側を定義するには、CRTMQMCHL コマンドを使用してください。

「Connection name (接続名)」フィールドはブランクのままにし、また対応する詳細がチャネルの送信側と一致するようにしてください。 詳細については、チャネルの作成を参照してください。

開始する側が受信側チャネルを開始できるように、開始される側のサブシステムに経路指定項目を追加してください。 このサブシステムは、LU 6.2 セッションで使用する APPC デバイスを割り振るもの でなければなりません。 したがって、そのデバイスの有効な通信項目を保持している必要があります。 経路指定項目は、メッセージ・チャネルの受信側を開始させるプログラムを呼び出します。

リンク上で通信セッションによって開始される側を定義するには、IBM i コマンド (例えば、ADDRTGE) を使用してください。

開始される側のパネルを、図 LU 6.2 通信設定パネル - Add Routing Entry に示しています。

```
Add Routing Entry (ADDRTGE)

Type choices, press Enter.

Subsystem description . . . . .  QCMN     Name
Library . . . . . .   *LIBL    Name, *LIBL, *CURLIB
Routing entry sequence number .  1       1-9999
Comparison data:
Compare value . . . . . .  MQSERIES

Starting position . . . . . .  37      1-80
Program to call . . . . . .  AMQCRC6B   Name, *RTGDTA
Library . . . . . .  QMAS400    Name, *LIBL, *CURLIB
Class . . . . . .  *SBSD     Name, *SBSD
Library . . . . . .  *LIBL     Name, *LIBL, *CURLIB
Maximum active routing steps . .  *NOMAX    0-1000, *NOMAX
Storage pool identifier . . . .  1       1-10



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

図 *35. LU 6.2* 通信設定パネル - 開始される側

サブシステム記述
この定義が置かれるサブシステムの名前です。 経路指定項目に該当するサブシステム記述を表示および更新するには、IBM i WRKSBSD コマンドを使用してください。

経路指定項目のシーケンス番号 (Routing entry sequence number)
この通信定義を識別するための、サブシステム内で固有の番号です。 値は 1 から 9999 の範囲内で指定できます。

比較データ: 値の比較 (Comparison data: Compare value)
Figure 1 に示されているように、**トランザクション・プログラム**パラメーターによって、セッションが開始されたときに受信されるストリングと比較するテキスト・ストリング。 文字ストリングは、送信側 CSI の「トランザクション・プログラム」フィールドから導出されます。

比較データ: 開始位置 (Comparison data: Starting position)
ストリング内の、比較が開始される文字位置です。

注 : この開始位置フィールドは、 ストリング内の比較を行うための文字位置であり、常に 37 です。

呼び出すプログラム (Program to call)
セッションを開始するために呼び出されるインバウンド・メッセージ・プログラムを実行する、プログラムの名前です。

プログラム AMQCRC6A は、デフォルト・キュー・マネージャーの場合に呼び出されます。 このプログラムは、IBM MQ for IBM i に付属のプログラムであり、環境をセットアップしてから AMQCRS6A を呼び出します。

追加キュー・マネージャーの場合:

- 各キュー・マネージャーには、そのライブラリーに置かれた特定の LU 6.2 呼び出し可能プログラムが備わっています。 このプログラムは AMQCRC6B と呼ばれ、キュー・マネージャーの作成時に自動的に生成されます。
- 各キュー・マネージャーには、追加する固有経路指定データを指定した特定の経路指定項目が必要です。 この経路指定データは、 要求側のシステムで指定する**トランザクション・プログラム**の名前 (開始する側 (送信側)を参照) と一致している必要があります。

例を以下の LU 6.2 通信設定パネル - Display Routing Entries に示します。

```
Display Routing Entries
System:  MY400
Subsystem description:  QCMN      Status:  ACTIVE

Type options, press Enter.
5=Display details

Start
Opt  Seq Nbr  Program    Library    Compare Value   Pos
10   *RTGDTA           'QZSCSRVR'     37
20   *RTGDTA           'QZRCSRVR'     37
30   *RTGDTA           'QZHQTRG'     37
50   *RTGDTA           'QVPPRINT'     37
60   *RTGDTA           'QNPSERVR'     37
70   *RTGDTA           'QNMAPINGD'    37
80   QNMAREXECD  QSYS     'AREXECD'     37
90   AMQCRC6A   QMQMBW    'MQSERIES'     37
100   *RTGDTA           'QTFDWNLD'     37
150   *RTGDTA           'QMFRCVR'     37



F3=Exit  F9=Display all detailed descriptions  F12=Cancel
```

図 *36. LU 6.2* 通信設定パネル - 開始される側

LU 6.2 通信設定パネル - Display Routing Entries のシーケンス番号 90 はデフォルト・キュー・マネージャーを表しており、これによって、 IBM MQ for IBM i の旧リリース (すなわち、V3R2、V3R6、V3R7、および V4R2) の構成との互換性を実現しています。 これらのリリースは、1 つのキュー・マネージャーだけを使用できます。 シーケンス番号 92 および 94 は、ライブラリー QMALPHA および QMBETA を使用して作成された ALPHA および BETA という 2 つの追加キュー・マネージャーを表します。

注 : 異なる経路指定データを使用することにより、それぞれのキュー・マネージャーごとに別々の経路指定項目を指定できます。 これらの項目によって、 使用するクラスに応じてジョブ優先順位を変えることができます。

クラス
この経路指定項目を使用して開始されるステップに使用されるクラスの、名前とライブラリーです。 このクラスは、 経路指定ステップの実行環境の属性を定義し、ジョブ優先順位を指定します。 該当のクラス項目を指定しなければなりません。 既存クラスの表示 または新規クラスの作成には、WRKCLS コマンドなどを使用してください。 リモート LU 6.2 システムからの作業要求の管理についての詳細は、「*IBM i Programming: Work Management Guide*」に記載されています。

ワーク・マネジメントに関する注意

AMQCRS6A ジョブは、他の IBM MQ ジョブとは異なる方法で開始されるため、 実行管理機能 に記載されている通常の IBM i 実行管理機能を利用できません。 LU 62 受信側ジョブの実行時プロパティーを変更するには、 以下のいずれかの変更を行います。

- AMQCRS6A ジョブの経路指定項目に指定されているクラス記述の変更
- 通信項目に対するジョブ記述の変更

通信ジョブの構成についての詳細は、「*IBM i Programming: Work Management Guide*」を参照してください。

キュー・マネージャー・クラスターの構成

クラスターによって提供されるキュー・マネージャーを相互接続するためのメカニズムにより、初期構成と継続的な管理を簡単に行えます。 クラスター・コンポーネントを定義し、クラスターを作成および管理できます。

始める前に

クラスタリングの概念の導入については、クラスターを参照してください。

キュー・マネージャー・クラスターを設計するときは、いくつかの決定を行う必要があります。 サンプル・クラスターおよびクラスターの設計を参照してください。

関連タスク

440 ページの『別のキュー・マネージャーへのクラスター・トピック定義の移動』
トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

関連資料

DELETE TOPIC

クラスターのコンポーネントの定義

クラスターは、キュー・マネージャー、クラスター・チャネル、およびクラスター・キューで構成されます。 クラスター・キューを定義したり、デフォルトのクラスター・オブジェクトのいくつかの側面を変更したりできます。 個別のクラスター送信側チャネルと伝送キューの間の関係に関して、さらに自動定義チャネルに関しての構成情報や状況情報を取得できます。

各クラスター・コンポーネントの定義については、以下のサブトピックを参照してください。

関連概念

クラスターのコンポーネント

クラスター・チャネル

関連タスク

クラスター・トピックの定義

301 ページの『新規クラスターのセットアップ』
クラスター例をセットアップするには、これらの手順に従ってください。 クラスターを設定する方法は、TCP/IP、LU 6.2、および単一の伝送キューまたは複数の伝送キューを使用する場合で別々に説明されています。 あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信することによって、クラスターの動作をテストします。

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

クラスター・キューの定義

クラスター・キューとは、クラスター・キュー・マネージャーでホストされ、同じクラスター内の別のキュー・マネージャーで使用できるキューです。 クラスター・キューは、キューをホストするクラスター・キュー・マネージャーのローカル・キューとして定義します。 キューが属するクラスターの名前を指定します。

次に示すのは、CLUSTER オプションを指定してクラスター・キューを定義する **runmqsc** コマンドの例です。

```
DEFINE QLOCAL(Q1) CLUSTER(SALES)
```

クラスター・キュー定義は、クラスター内の他のキュー・マネージャーに通知されます。 クラスター内にあるその他のキュー・マネージャーは、対応するリモート・キュー定義がなくても、クラスター・キューにメッセージを書き込むことができます。 クラスター名前リストを使用して、クラスター・キューを複数のクラスターに通知できます。

キューが通知されると、クラスター内のキュー・マネージャーはそのキューにメッセージを書き込めるようになります。 メッセージを書き込むときには、キュー・マネージャーが、フルリポジトリーで、そのキ

ューがホストされている場所を調べる必要があります。 格納場所が分かったら、宛先情報をメッセージに追加して、クラスター伝送キューにメッセージを書き込みます。

z/OS クラスター・キューは、IBM MQ for z/OS でのキュー共用グループの メンバーによって共用されるキューにすることができます。

バインド中

クラスターを構築するときには、同じクラスター・キューをホストしている複数のキュー・マネージャーで構成されるクラスターを構築することができます。 シーケンスに含まれるすべてのメッセージがキューの同一インスタンスに送信されるようにします。 このとき、MQOPEN 呼び出しの MQOO_BIND_ON_OPEN オプションを使用すると、一連のメッセージを特定のキューにバインドすることができます。

クラスター伝送キュー

キュー・マネージャーは、同じクラスター内の他のキュー・マネージャーのメッセージを複数の伝送キューに格納することができます。 複数のクラスター伝送キューにメッセージを格納するようキュー・マネージャーを構成する方法は 2 つあります。 キュー・マネージャー属性 **DEFCLXQ** を CHANNEL に設定すると、クラスター送信側チャネルごとに異なるクラスター伝送キューが、SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE から自動的に作成されます。 CLCHNAME 伝送キュー・オプションを 1 つ以上のクラスター送信側チャネルに一致するように設定すると、キュー・マネージャーは、一致しているチャネルのメッセージを、そのチャネルの伝送キューに格納できます。

重要 : IBM WebSphere MQ 7.5 より前のバージョンの製品からアップグレードされたキュー・マネージャーで専用の SYSTEM.CLUSTER.TRANSMIT.QUEUES を使用する場合は、SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE の SHARE/NOSHARE オプションが **SHARE** に設定されている必要があります。

別のキュー・マネージャー上のクラスター・キューへのメッセージは、クラスター伝送キューに配置されてから送信されます。 クラスター送信側チャネルは、メッセージをクラスター伝送キューから他のキュー・マネージャーのクラスター受信側チャネルに転送します。 デフォルトでは、システムで定義された 1 つのクラスター伝送キューが、他のクラスター・キュー・マネージャーに転送するすべてのメッセージを保持します。 このキューを SYSTEM.CLUSTER.TRANSMIT.QUEUE といいます。 クラスターの一部であるキュー・マネージャーは、 このクラスター伝送キューに入っているメッセージを、同じクラスター内の他のキュー・マネージャーに送信できます。

単一の SYSTEM.CLUSTER.TRANSMIT.QUEUE キューの定義は、z/OS の場合を除いて、デフォルトではすべてのキュー・マネージャーに作成されます。 z/OS z/OS では、提供サンプル **CSQ4INSX** を使用して定義できます。

複数の伝送キューを使用して他のクラスター化されたキュー・マネージャーにメッセージを転送するように、キュー・マネージャーを構成できます。 追加のクラスター伝送キューを手動で定義するか、またはキュー・マネージャーが自動的にキューを作成するように設定することもできます。

キューがキュー・マネージャーによって自動的に作成されるようにするには、キュー・マネージャー属性 DEFCLXQ を SCTQ から CHANNEL に変更します。 その結果、キュー・マネージャーはクラスター送信側チャネルが作成されるたびに別個のクラスター伝送キューを作成します。 伝送キューは、モデル・キュー SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE から永続動的キューとして作成されます。 各永続動的キューの名前は SYSTEM.CLUSTER.TRANSMIT. *ChannelName* です。 各永続動的クラスターの伝送キューが関連付けられたクラスター送信側チャネルの名前は、ローカル伝送キュー属性 CLCHNAME に設定されます。 リモート・クラスター化されたキュー・マネージャーのメッセージは、SYSTEM.CLUSTER.TRANSMIT.QUEUE ではなく、関連付けられたクラスター送信側チャネルの永続動的クラスターの伝送キューに置かれます。

クラスター伝送キューを手動で作成するには、USAGE 属性を XMITQ に設定し、CLCHNAME 属性を 1 つ以上のクラスター送信側チャネルに解決する汎用チャネル名に設定して、ローカル・キューを作成します。 ClusterChannelName を参照してください。 クラスター伝送キューを手動で作成する場合は、伝送キューを 1 つのクラスター送信側チャネルに関連付けるかまたは複数のクラスター送信側チャネルに関連付ける

かを選択できます。 CLCHNAME の属性は総称名です。これは、名前に複数のワイルドカード文字"*"を入れることができることを意味します。

キュー・マネージャーを完全リポジトリーに接続するために手動で作成する初期クラスター送信側チャネルを除いて、クラスター送信側チャネルは自動的に作成されます。 自動的に作成されるのは、クラスター・キュー・マネージャーに転送するメッセージがあるときです。 クラスター送信側チャネルは、宛先キュー・マネージャーで特定のクラスターに対するクラスター・メッセージを受信するクラスター受信側チャネルの名前と同じ名前で作成されます。

クラスター受信側チャネルの命名規則に従うと、CLCHNAME の総称値を定義して、さまざまなクラスター・メッセージをそれぞれの伝送キューにフィルタリングして転送することができます。 例えば、*ClusterName. QmgrName* のクラスター受信側チャネルの命名規則に従う場合、総称名 *ClusterName.** は、異なるクラスターのメッセージを異なる伝送キューにフィルタリングします。 伝送キューを手動で定義し、各伝送キューの CLCHNAME を *ClusterName.** に設定してください。

クラスター伝送キューからクラスター送信側チャネルへの関連付けを変更しても、即時には有効になりません。 現在関連付けられていて、クラスター送信側チャネルがサービス中の伝送キューには、クラスター送信側チャネルが転送処理中のメッセージが含まれている可能性があります。 キュー・マネージャーがクラスター送信側チャネルの関連付けを別の伝送キューに変更できるのは、現在関連付けられている伝送キューにクラスター送信側チャネルで処理中のメッセージがないときにのみです。 それに該当するのは、クラスター送信側チャネルで処理するメッセージが伝送キューに残っていないとき、またはメッセージの処理が中断していてクラスター送信側チャネルに"未完了"メッセージがないときです。 この状態になると、クラスター送信側チャネルの未処理メッセージはどれも新たに関連付けられた伝送キューに転送され、クラスター送信側チャネルの関連付けが変更されます。

解決結果がクラスター伝送キューになるリモート・キュー定義を作成できます。 この定義では、ローカル・キュー・マネージャーと同じクラスター内にキュー・マネージャー QMX があり、伝送キュー QMX はありません。

```
DEFINE QREMOTE(A) RNAME(B) RQMNAME(QMX)
```

キュー名の解決時は、デフォルト伝送キューよりクラスター伝送キューが優先されます。 A に書き込まれたメッセージは、クラスター伝送キューに保管されてから、QMX 上のリモート・キュー B に送信されます。

また、キュー・マネージャーは、クラスターの一部ではない他のキュー・マネージャーと通信することもできます。 分散キューイング環境の場合と同じように、他のキュー・マネージャーへのチャネルと伝送キューを定義する必要があります。

注 : アプリケーションでは、解決結果がクラスター伝送キューになるキューに書き込むようにし、クラスター伝送キューには直接書き込まないようにしてください。

リモート・キューの自動定義

クラスター内のキュー・マネージャーは、クラスター内のリモート・キューに対して、リモート・キュー定義を必要としません。 クラスター・キュー・マネージャーは、完全リポジトリーからリモート・キューのロケーションを見つけます。 メッセージにルーティング情報を追加し、それをクラスター伝送キューに書き込みます。 IBM MQ は、リモート・キュー定義と同じ内容の定義を自動的に作成するため、メッセージを送信することができます。

自動作成されたリモート・キュー定義を修正したり削除したりすることはできません。 しかし、CLUSINFO 属性を指定した DISPLAY QUEUE **runmqsc** コマンドを使用することにより、キュー・マネージャー上のローカル・キューをすべて表示できるだけでなく、リモート・キュー・マネージャー上のクラスター・キューも含めてすべてのクラスター・キューを表示できます。 以下に例を示します。

```
DISPLAY QUEUE(*) CLUSINFO
```

関連概念
クラスター・キュー
使用するクラスター伝送キュー・タイプの選択方法
関連資料
ClusterChannelName (MQCHAR20)

自動定義クラスター送信側チャネルの処理

最初の CLUSSDR 定義と CLUSRCVR 定義を作成してキュー・マネージャーをクラスターに導入した後、IBM MQ は、クラスター内の別のキュー・マネージャーにメッセージを移動する必要があるときに、他のクラスター送信側チャネル定義を自動的に作成します。 自動定義されたクラスター送信側チャネルについては、情報を表示することはできますが、それを変更することはできません。 その動作を変更するには、チャネル自動定義出口を使用することができます。

始める前に

自動定義チャネルの概要については、自動定義されたクラスター送信側チャネルを参照してください。

このタスクについて

自動定義クラスター送信側チャネルは、クラスターにより必要に応じて作成されます。その後、通常の切断間隔規則を使ってシャットダウンされるまでアクティブ状態になります。

クラスター送信側チャネル (CLUSSDR) は、アプリケーション・メッセージと内部クラスター管理メッセージの両方を移動するために自動定義できます。 例えば、パブリッシュ/サブスクライブ・クラスター (クラスター・トピックが定義されているもの) では、部分リポジトリー間でチャネルを定義して「プロキシー・サブスクリプション」状態の交換を許可できます。 長期間にわたって必要ではない (非アクティブ) 場合、自動定義 CLUSSDR が部分リポジトリーのクラスター情報のキャッシュから削除され、そのキュー・マネージャーに表示されなくなります。

Multi マルチプラットフォーム では、OAM (オブジェクト権限マネージャー) は、自動定義クラスター送信側チャネルの存在を認識しません。 自動定義クラスター送信側チャネルに対してコマンド **start**、**stop**、**ping**、**reset**、または **resolve** を発行すると、OAM はコマンドを発行したユーザーが対応するクラスター受信側チャネルに対して同じアクションを実行する権限があるかどうかを検査します。

z/OS z/OS では、自動定義クラスター送信側チャネルを、他のチャネルと同じように保護することができます。

手順

- 特定のクラスター・キュー・マネージャーの自動定義チャネルに関する情報を表示します。

  自動的に定義されたチャネルは、DISPLAY CHANNEL **runmqsc** コマンドで表示することができません。自動定義チャネルを調べるには、次のコマンドを使用します。

  ```
  DISPLAY CLUSQMGR(qMgrName)
  ```

- 特定の CLUSRCVR の自動定義チャネルの状況を表示します。

  作成した CLUSRCVR チャネル定義に対応する自動定義チャネル CLUSSDR の状況を表示するには、次のコマンドを使用します。

  ```
  DISPLAY CHSTATUS(channelname)
  ```

- チャネル自動定義出口を使用して、自動定義チャネルの動作を変更します。

  クラスター送信側チャネルまたはクラスター受信側チャネルをユーザーの要求に合わせてカスタマイズするために、ユーザー出口プログラムを作成したい場合、IBM MQ のチャネル自動定義出口を使用することができます。 例えば、クラスター環境でチャネル自動定義出口を使用すると、以下のような変更を行えます。

  - 通信の定義 (SNA LU6.2 名) を変更する場合。
  - 他の出口 (セキュリティー出口など) を追加または削除する場合。
  - チャネル出口の名前を変更する場合。

    CLUSSDR チャネル出口の名前は CLUSRCVR チャネル定義から自動生成されるため、希望の名前にならない場合があります。異なるプラットフォーム上にチャネルの両端がある場合は特にそう言えます。

出口名の形式は、プラットフォームによって異なります。 以下に例を示します。

- z/OS z/OS プラットフォームでは、SCYEXIT (セキュリティー出口名) パラメーターの形式は SCYEXIT('SECEXIT') です
- Windows Windows プラットフォームでは、SCYEXIT (セキュリティー出口名) パラメーターの形式は SCYEXIT(' *drive*:*path**library* (secexit)') です。

注 : z/OS チャネル自動定義出口がない場合、z/OS キュー・マネージャーは CLUSSDR チャネル出口名をチャネルの他方の端の CLUSRCVR チャネル定義から派生するものにします。 z/OS 以外に派生する名前から z/OS 出口名を付けるために、以下のアルゴリズムが使用されます。

- マルチプラットフォーム における出口名の一般的な形式は、*path*/*library*(*function*) です。
- *function* が存在する場合は、8 文字までが使用されます。
- 存在しない場合は、*library* に含まれる最大 8 文字が使用されます。

以下に例を示します。

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

- クラスターで **PROPCTL** を使用して、 IBM MQ キュー・マネージャーから前のバージョンの製品のキュー・マネージャーに送信されるメッセージから RFH2 などのアプリケーション・ヘッダーを除去する必要がある場合は、 **PROPCTL** を NONE の値に設定するチャネル自動定義出口を作成する必要があります。

- チャネル属性 LOCLADDR を使用して、アドレッシングの側面を制御します。
  - アウトバウンド (TCP) チャネルが特定の IP アドレス、ポート、またはポートの範囲を使用できるようにするには、チャネル属性 LOCLADDR を使用します。 これは、複数のネットワーク・カードがあり、 チャネルでアウトバウンド通信に特定のネットワーク・カードを使用する場合に有益です。
  - CLUSSDR チャネルの仮想 IP アドレスを指定するには、手動で定義した CLUSSDR の LOCLADDR での IP アドレスを使用します。 ポート範囲を指定するには、CLUSRCVR でのポート範囲を使用します。
  - クラスターで LOCLADDR を使用して、アウトバウンド通信チャネルを特定の IP アドレスにバインドする必要がある場合は、自動的に定義された CLUSSDR チャネルに LOCLADDR 値を強制的に指定するために、チャネル自動定義出口を作成することができます。 また、手動で定義された CLUSSDR チャネルにもこの値を指定する必要があります。
  - クラスター内のすべてのキュー・マネージャーで、すべてのアウトバウンド通信に特定のポートまたはポートの範囲を使用するには、CLUSRCVR チャネルの LOCLADDR にポート番号またはポート範囲を入力します。

  注 : すべてのキュー・マネージャーが同一サーバー上にない限り、CLUSRCVR チャネルの LOCLADDR フィールドに IP アドレスを入力しないでください。 LOCLADDR IP アドレスは、CLUSRCVR チャネルを使用して接続するすべてのキュー・マネージャーの自動定義 CLUSSDR チャネルに伝搬されます。

  Multi マルチプラットフォーム では、ローカル・アドレスが定義されていないすべての送信側チャネルで使用される、デフォルトのローカル・アドレス値を設定することができます。 このデフォルト値を定義するには、キュー・マネージャーの開始前に MQ_LCLADDR 環境変数を設定します。 この値の形式は、MQSC 属性の LOCLADDR の形式と同じです。

関連資料
ローカル・アドレス (LOCLADDR)

デフォルト・クラスター・オブジェクトの処理

デフォルト・チャネル定義は、他のすべてのチャネル定義と同様に、MQSC または PCF コマンドを実行することで変更できます。 SYSTEM.CLUSTER.HISTORY.QUEUE を除き、デフォルトのキュー定義は変更しないでください。

これらのオブジェクトの完全なリストについては、デフォルトのクラスタオブジェクトを参照してください。 以下のリストには、変更可能なオブジェクトしか含まれていません。

SYSTEM.CLUSTER.HISTORY.QUEUE
クラスター内の各キュー・マネージャーには、SYSTEM.CLUSTER.HISTORY.QUEUE という名前のローカル・キューがあります。 SYSTEM.CLUSTER.HISTORY.QUEUE は、サービス目的でクラスター状態情報の履歴を保管するために使用されます。

デフォルトのオブジェクト設定では、SYSTEM.CLUSTER.HISTORY.QUEUE は PUT (ENABLED) に設定されます。 履歴収集を抑止するには、設定を PUT (DISABLED) に変更します。

SYSTEM.CLUSTER.TRANSMIT.QUEUE
各キュー・マネージャーには、SYSTEM.CLUSTER.TRANSMIT.QUEUE というローカル・キューの定義があります。 SYSTEM.CLUSTER.TRANSMIT.QUEUE は、クラスター内のすべてのキューおよびキュー・マネージャーに対するすべてのメッセージのデフォルト伝送キューです。 キュー・マネージャー属性 DEFXMITQ z/OS (z/OS を除く) を変更することにより、各クラスター送信側チャネルのデフォルト伝送キューを SYSTEM.CLUSTER.TRANSMIT.*ChannelName* に変更できます。
SYSTEM.CLUSTER.TRANSMIT.QUEUE は削除できません。 また、使用されるデフォルト伝送キューが SYSTEM.CLUSTER.TRANSMIT.QUEUE であるか SYSTEM.CLUSTER.TRANSMIT.*ChannelName* であるかの許可検査を定義するためにも使用されます。

関連概念
デフォルトのクラスター・オブジェクト

クラスター伝送キューとクラスター送信側チャネルの操作
クラスター・キュー・マネージャー間のメッセージは、クラスター伝送キューに保管されてから、クラスター送信側チャネルによって転送されます。 どの時点でも、クラスター送信側チャネルが関連付けられる伝送キューは 1 つです。 チャネルの構成を変更すると、そのチャネルは次回開始したときに、別の伝送キューに切り替わる可能性があります。 この切り替え処理は自動化されていて、トランザクションで実行されます。

クラスター送信側チャネルが関連付けられている伝送キューを表示するには、以下の MQSC コマンドを実行します。

```
DISPLAY CHSTATUS(*) WHERE(CHLTYPE EQ CLUSSDR)
```

```
AMQ8417: Display Channel Status details.
CHANNEL(TO.QM2)          CHLTYPE(CLUSSDR)
CONNAME(9.146.163.190(1416))    CURRENT
RQMNAME(QM2)             STATUS(STOPPED)
SUBSTATE( )            XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

停止したクラスター送信側チャネルの保存されたチャネル状況に示される伝送キューは、チャネルが再び開始すると変更される可能性があります。 デフォルト伝送キューの選択プロセスについては、294 ページの『クラスター送信側チャネルによるデフォルト伝送キューの選択』で説明しています。手動で定義した伝送キューの選択プロセスについては、295 ページの『クラスター送信側チャネルによる手動で定義された伝送キューの選択』で説明しています。

すべてのクラスター送信側チャネルは、開始時にその伝送キューとの関連付けを再チェックします。 伝送キューの構成またはキュー・マネージャーのデフォルトが変更されている場合、チャネルと伝送キューの関連付けが変更される可能性があります。 構成変更の結果として、チャネルが異なる伝送キューで再始動した場合、新しく関連付けられた伝送キューにメッセージを転送するプロセスが発生します。 ある伝送キューから別の伝送キューへのクラスター送信側チャネルの転送プロセスについては、295 ページの『クラスター送信側チャネルを異なる伝送キューに切り替えるプロセスの仕組み』で説明しています。

クラスター送信側チャネルの動作は、送信側およびサーバー・チャネルとは異なります。 これらのチャネルは、チャネル属性 **XMITQ** が変更されるまで、同じ伝送キューとの関連付けを維持します。 送信側またはサーバー・チャネルで伝送キュー属性を変更して、チャネルを再始動しても、メッセージは古い伝送キューから新しい伝送キューには転送されません。

クラスター送信側チャネルと送信者、またはサーバー・チャネルの相違点の一つは、複数のクラスター送信側チャネルがクラスター伝送キューを開くことができるという点ですが、通常の伝送キューをオープンできる送信側チャネルまたはサーバー・チャネルだけです。 伝送キューを共用しないクラスター送信側チャネルのオプションがあります。 排他性は強制されません。これは、構成の結果です。 クラスター内でメッセージが通るパスを構成して、そのメッセージが他のアプリケーション間を流れるメッセージと伝送キューやチャネルを共有しないようにすることができます。 クラスター化: クラスター伝送キューの構成方法の計画および 347 ページの『クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』 を参照してください。

クラスター送信側チャネルによるデフォルト伝送キューの選択

クラスター伝送キューは、名前が SYSTEM.CLUSTER.TRANSMIT で始まるシステム・デフォルト・キュー、または手動で定義されたキューのいずれかです。 クラスター送信側チャネルは、2 つの方法のいずれかでクラスター伝送キューに関連付けられます。一方の方法はデフォルト・クラスター伝送キュー・メカニズム、もう一方の方法は手動による構成です。

デフォルト・クラスター伝送キューは、キュー・マネージャー属性 **DEFCLXQ** として設定されます。 その値は、SCTQ または CHANNEL です。 新しいキュー・マネージャーおよびマイグレーションされたキュー・マネージャーは、SCTQ に設定されます。 この値を CHANNEL に変更することができます。

SCTQ が設定されている場合、デフォルト・クラスター伝送キューは SYSTEM.CLUSTER.TRANSMIT.QUEUE となります。 このキューは、すべてのクラスター送信側チャネルがオープンできます。 このキューをオープンしないクラスター送信側チャネルは、手動で定義されたクラスター伝送キューに関連付けられていないものです。

CHANNEL が設定されている場合には、キュー・マネージャーがクラスター送信側チャネルごとに個別の永続動的伝送キューを作成できます。 各キューは SYSTEM.CLUSTER.TRANSMIT. *ChannelName* という名前で、モデル・キュー SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE から作成されます。 手動で定義されたクラスター伝送キューに関連付けられていないクラスター送信側チャネルは、それぞれ永続動的クラスター伝送キューに関連付けられます。 このキューは、このクラスター送信側チャネルが処理するクラスター宛先に別個のクラスター伝送キューが必要であるものの、キューが存在していない場合に、キュー・マネージャーによって作成されます。

一部のクラスター宛先は、手動で定義された伝送キューに関連付けられたクラスター送信側チャネルによって処理できます。その他のクラスター宛先は、デフォルト・キューによって処理されます。 クラスター送信側チャネルと伝送キューの関連付けでは、常に、手動で定義された伝送キューがデフォルト伝送キューに優先されます。

クラスター伝送キューの優先順位については、294 ページの図 37 で説明されています。 手動で定義されたクラスター伝送キューに関連付けられていないクラスター送信側チャネルは、CS.QM1 だけです。 これが手動で定義された伝送キューに関連付けられない理由は、伝送キューの **CLCHNAME** 属性に設定されたチャネル名の中に、CS.QM1 と一致する名前がないためです。

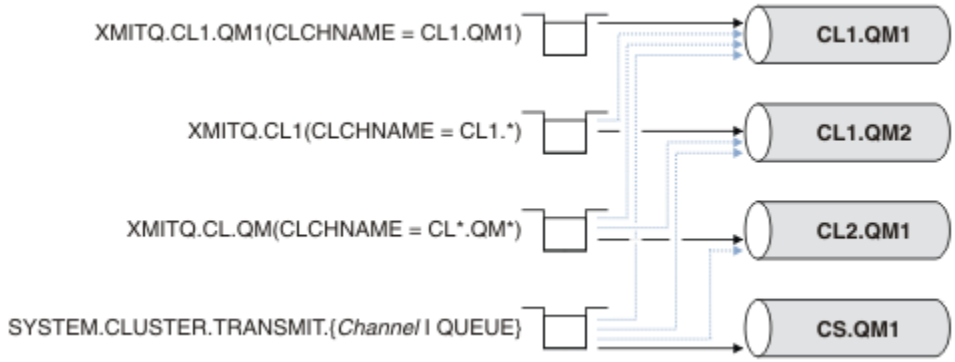


図 37. 伝送キューとクラスター送信側チャネルの優先順位

クラスター送信側チャネルによる手動で定義された伝送キューの選択

手動で定義されたキューの伝送キュー属性 **USAGE** は XMITQ に設定され、クラスター・チャネル名属性 **CLCHNAME** は特定のチャネル名または総称チャネル名に設定されています。

CLCHNAME キュー属性に設定された名前がクラスター送信側チャネル名と一致する場合、そのチャネルがキューに関連付けられます。 この名前は、完全一致 (名前にワイルドカードが含まれていない場合)、または最適一致 (名前にワイルドカードが含まれている場合) のいずれかです。

複数の伝送キューの **CLCHNAME** 定義が同じクラスター送信側チャネルに一致する場合、「定義がオーバーラップしている」と言います。 あいまいさを解決するため、複数の一致の間には優先順位があります。 完全一致が常に優先されます。 294 ページの図 37 に、伝送キューとクラスター送信側チャネルとの間の関連付けを示します。 黒の矢印は実際の関連付けを表し、グレイの矢印は潜在的な関連付けを表します。 伝送キューの優先順位については、294 ページの図 37 で説明されています。

XMITQ.CL1.QM1
: 伝送キュー XMITQ.CL1.QM1 の **CLCHNAME** 属性は CL1.QM1 に設定されています。 **CLCHNAME** 属性の定義である CL1.QM1 はワイルドカードを取らず、他の伝送キューに定義されたワイルドカードに一致する他のすべての CLCHNAME 属性よりも優先されます。 キュー・マネージャーは、CL1.QM1 クラスター送信側チャネルによって転送されるすべてのクラスター・メッセージを XMITQ.CL1.QM1 伝送キューに格納します。 唯一の例外は、複数の伝送キューで **CLCHNAME** 属性が CL1.QM1 に設定されている場合です。 その場合、キュー・マネージャーは CL1.QM1 クラスター送信側チャネルのメッセージをそれらの伝送キューのいずれかに格納します。 チャネルが開始すると、キュー・マネージャーはキューを任意に選択します。 再びチャネルが開始して、別のキューが選択されることもあります。

XMITQ.CL1
: 伝送キュー XMITQ.CL1 の **CLCHNAME** 属性は CL1.* に設定されています。 **CLCHNAME** 属性の定義である CL1.* は末尾に 1 つのワイルドカードを取り、これは CL1. で始まる任意のクラスター送信側チャネルの名前に一致します。 キュー・マネージャーは、CL1. で始まる名前を持つクラスター送信側チャネルによって転送されるすべてのクラスター・メッセージを伝送キュー XMITQ.CL1 に格納します。 ただし、キュー XMITQ.CL1.QM1 などのように、より具体的に一致する伝送キューがあれば、そのキューが使用されます。 末尾に 1 つのワイルドカードがある定義は、具体性の点で、ワイルドカードがない定義よりも劣り、複数のワイルドカードがある定義や、ワイルドカードの後にさらに文字が続く定義よりも優れています。

XMITQ.CL.QM
: XMITQ.CL.QM は、**CLCHNAME** 属性が CL*.QM* に設定された伝送キューの名前です。 CL*.QM* という定義には 2 つのワイルドカードがあり、それらは任意のクラスター送信側チャネルの名前が CL. で始まり、その名前が QM を含むかまたはそれで終わることに一致します。 ワイルドカードが 1 文字であるときの一致と比較すると、この一致は具体性の点で劣ります。

SYSTEM.CLUSTER.TRANSMIT. *channelName*|QUEUE
: 伝送キューの **CLCHNAME** 属性にキュー・マネージャーが使用する予定のクラスター送信側チャネルの名前と一致するものがない場合、キュー・マネージャーはデフォルトのクラスター伝送キューを使用します。 デフォルトのクラスター伝送キューは、単一システム・クラスター伝送キュー SYSTEM.CLUSTER.TRANSMIT.QUEUE、またはキュー・マネージャーが特定のクラスター送信側チャネル SYSTEM.CLUSTER.TRANSMIT. *channelName* 用に作成したシステム・クラスター伝送キューのいずれかです。 どちらがデフォルトであるかは、キュー・マネージャーの **DEFXMITQ** 属性の設定によって決まります。

ヒント: オーバーラップする定義が明らかに必要であるのでなければ、構成がわかりにくくて複雑にならないようにするために、そのような定義は使わないようにしてください。

クラスター送信側チャネルを異なる伝送キューに切り替えるプロセスの仕組み

クラスター送信側チャネルとクラスター伝送キューとの関連付けを変更するには、随時、任意の伝送キューの **CLCHNAME** パラメーターを変更するか、キュー・マネージャー・パラメーター **DEFCLXQ** を変更します。 すぐには何も起こりません。 変更はチャネルが開始した時にのみ行われます。 開始時に、引き続き同じ伝送キューからメッセージを転送するかどうかをチェックします。 次の 3 種類の変更によって、クラスター送信側チャネルと伝送キューとの関連付けは変更されます。

1. クラスター送信側チャネルが現在関連付けられている伝送キューの **CLCHNAME** パラメーターをより限定的でなくなるように再定義するか、またはブランクに再定義する。あるいは、チャネルが停止されているときに、クラスター伝送キューを削除する。

   他のクラスター伝送キューが、チャネル名により一致するようになる可能性があります。 あるいは、クラスター送信側チャネルの名前と一致する伝送キューが他にない場合、関連付けはデフォルト伝送キューに戻る必要があります。
2. 他の任意のクラスター伝送キューの **CLCHNAME** パラメーターを再定義するか、クラスター伝送キューを追加する。

   クラスター送信側チャネルに現在関連付けられている伝送キューよりも、別の伝送キューの **CLCHNAME** パラメーターのほうがクラスター送信側チャネルに一致するようになる可能性があります。 クラスター送信側チャネルが現在デフォルト・クラスター伝送キューに関連付けられている場合、手動で定義されたクラスター伝送キューに関連付けられることになる可能性があります。
3. クラスター送信側チャネルが現在デフォルト・クラスター伝送キューに関連付けられている場合、**DEFCLXQ** キュー・マネージャー・パラメーターを変更する。

クラスター送信側チャネルの関連付けが変更された場合、チャネルは開始した時点で、その関連付けを新しい伝送キューに切り替えます。 この切り替えの間、メッセージが失われないようにします。 メッセージは、チャネルによってリモート・キュー・マネージャーに転送される順序に従って、新しい伝送キューに転送されます。

要確認 : クラスター内でメッセージを転送する場合と同様、メッセージをグループに書き出して、順番に送信する必要のあるメッセージが順番に送信されているかどうかを確認します。 まれに、クラスター内でメッセージの順序が狂うことがあります。

切り替えプロセスは、以下のトランザクション・ステップに従って行われます。 切り替えプロセスが中断された場合は、チャネルが再始動した時点で、現行のトランザクション・ステップが再開されます。

ステップ 1 - 元の伝送キューのメッセージの処理
クラスター送信側チャネルが関連付けられた新しい伝送キューは、他のクラスター送信側チャネルと共有している可能性があります。 クラスター送信側チャネルへのメッセージは、引き続き元の伝送キューに配置されます。 トランザクション切り替えプロセスにより、メッセージは元の伝送キューから新しい伝送キューに転送されます。 クラスター送信側チャネルは、メッセージを新しい伝送キューからクラスター受信側チャネルに転送します。 チャネル状況には、まだ古い伝送キューに関連付けられたクラスター送信側チャネルが示されます。

切り替えプロセスは、新しく到着したメッセージの転送も継続します。 このステップは、切り替えプロセスによって転送される残りのメッセージの数がゼロに達するまで続きます。 メッセージの数がゼロに達すると、プロシージャーはステップ 2 に移ります。

ステップ 1 の間、チャネルのディスク・アクティビティーは増加します。 持続メッセージは、最初の伝送キューからコミットされ、2 番目の伝送キューに渡されます。 このディスク・アクティビティーは、通常のメッセージ転送の一環として、メッセージが伝送キューに配置され、そこから削除される際に、コミットされるメッセージに追加されます。 概念的には、切り替えプロセス中に到着するメッセージはないため、この遷移は可能な限り短時間で行われるはずです。 メッセージが到着した場合は、切り替えプロセスによって処理されます。

ステップ 2 - 新しい伝送キューのメッセージの処理
クラスター送信側チャネルの元の伝送キューに残っているメッセージがなくなると同時に、新しいメッセージは直接新しい伝送キューに配置されるようになります。 チャネル状況には、クラスター送信側チャネルが新しい伝送キューに関連付けられていることが示されます。 次のメッセージがキュー・マネージャーのエラー・ログに書き込まれます。 " AMQ7341 チャネル *ChannelName* の伝送キューは *QueueName* です。"

複数のクラスター伝送キューとクラスター伝送キュー属性

クラスター・メッセージを別のキュー・マネージャーへ転送する場合、1 つのクラスター伝送キューに格納するか、または複数のキューに格納するかを選択できます。 1 つのキューを使用する場合は、クラスター伝送キュー属性の 1 つのセットを設定および照会し、複数のキューを使用する場合は、複数のセットを設定および照会します。 一部の属性では、複数のセットを設定すると有利です。例えば、キュー項目数を照

会することによって、すべてのチャネルではなく、1 つのチャネルまたはチャネルのセットについて、転送を待機中のメッセージ数が分かります。 その他の属性については、複数のセットを設定すると不利になります。例えば、すべてのクラスター伝送キューに同じアクセス権を構成することはしたくないはずです。この理由から、アクセス権は特定のクラスター伝送キューのプロファイルではなく、常に SYSTEM.CLUSTER.TRANSMIT.QUEUE のプロファイルに対してチェックされます。 より粒度の細かいセキュリティー検査を適用する必要がある場合には、アクセス制御と複数のクラスター伝送キューを参照してください。

複数のクラスター送信側チャネルと複数の伝送キュー

キュー・マネージャーは、メッセージをクラスター送信側チャネルに転送する前に、クラスター伝送キューに保管します。 そして、メッセージの宛先に接続されたクラスター送信側チャネルを選択します。 共通して同じ宛先に接続された複数のクラスター送信側チャネルから選択できる場合もあります。 宛先は、複数のクラスター送信側チャネルによって単一のキュー・マネージャーに接続された同じ物理的キューである場合があります。 宛先が、同じクラスター内の異なる複数のキュー・マネージャーでホストされた、同じキュー名を持つ多数の物理的キューである場合もあります。 宛先に接続されたクラスター送信側チャネルから選択できる場合には、ワークロード・バランシング・アルゴリズムがクラスター送信側チャネルを選択します。 この選択は、複数の要因に基づいて行われます。クラスター・ワークロード管理アルゴリズムを参照してください。

298 ページの図 38 では、CL1.QM1、CL1.QM2、および CS.QM1 は、いずれも同じ宛先に至る可能性のあるチャネルです。 例えば、CL1 内での Q1 を QM1 および QM2 に定義した場合、CL1.QM1 と CL1.QM2 の両方が、2 つの異なるキュー・マネージャー上にある同じ宛先 Q1 への経路を提供します。 チャネル CS.QM1 も CL1 内にある場合、Q1 へのメッセージはこのチャネルも使用できます。 CS.QM1 のクラスター・メンバーシップはクラスター名前リストで定義できます。チャネル名がその構造にクラスター名を含めないのは、このためです。 ワークロード・バランシング・パラメーターおよび送信側アプリケーションによっては、Q1 への一部のメッセージが伝送キュー XMITQ.CL1.QM1、XMITQ.CL1、および SYSTEM.CLUSTER.TRANSMIT.CS.QM1 のそれぞれに配置されることもあります。

メッセージ・トラフィックを分離して、同じ宛先へのメッセージが、それとは異なる宛先へのメッセージとキューやチャネルを共有しないようにする場合には、トラフィックを複数の異なるクラスター送信側チャネルに分割する方法をまず検討してから、特定のチャネルのメッセージを別個の伝送キューに分離する方法を検討する必要があります。 同じクラスターの同じキュー・マネージャー上にあるクラスター・キューは、一般に同じクラスター・チャネルを共有します。 複数のクラスター伝送キューを定義するだけでは、クラスター・メッセージ・トラフィックを異なるキューに分離するには不十分です。 異なる宛先キューへのメッセージをそれぞれに異なるチャネルに分離しない限り、メッセージは同じクラスター伝送キューを共有します。

メッセージが使用するチャネルを分離する簡単な方法は、複数のクラスターを作成することです。 各クラスターのあらゆるキュー・マネージャーには、1 つのクラスター・キューだけを定義します。 そうしてから、クラスターとキュー・マネージャーの組み合わせごとに異なるクラスター受信側チャネルを定義すれば、各クラスター・キューへのメッセージがクラスター・チャネルを他のクラスター・キューへのメッセージと共有することはありません。 クラスター・チャネルに個々の伝送キューを定義すると、送信側キュー・マネージャーは、各伝送キューに 1 つのクラスター・キューへのメッセージだけを保管します。 例えば、2 つのクラスター・キューにリソースを共有させないようにするには、これらのクラスター・キューを同じキュー・マネージャーでそれぞれに異なるクラスターに配置するか、あるいは同じクラスター内の異なるキュー・マネージャーに配置することができます。

クラスター伝送キューの選択は、ワークロード・バランシング・アルゴリズムには影響しません。 ワークロード・バランシング・アルゴリズムが、メッセージを転送するクラスター送信側チャネルを選択します。そして、そのチャネルによって処理される伝送キューにメッセージを配置します。 チャネルを選択し直すためにワークロード・バランシング・アルゴリズムが必要になった場合 (例えば、チャネルが停止した場合)、アルゴリズムはメッセージを転送するための別のチャネルを選択できる場合があります。 別のチャネルを選択し、その新しいチャネルが別のクラスター伝送キューからのメッセージを転送する場合、ワークロード・バランシング・アルゴリズムはメッセージをその別の伝送キューに転送します。

298 ページの図 38 では、2 つのクラスター送信側チャネル CS.QM1 と CS.QM2 が、デフォルト・システム伝送キューに関連付けられています。 ワークロード・バランシング・アルゴリズムがメッセージを SYSTEM.CLUSTER.TRANSMIT.QUEUE、または他のクラスター伝送キューに保管すると、そのメッセージ

の転送先クラスター送信側チャネルの名前が、メッセージの相関 ID に保管されます。 各チャネルは、相関 ID がチャネル名と一致するメッセージのみを転送します。

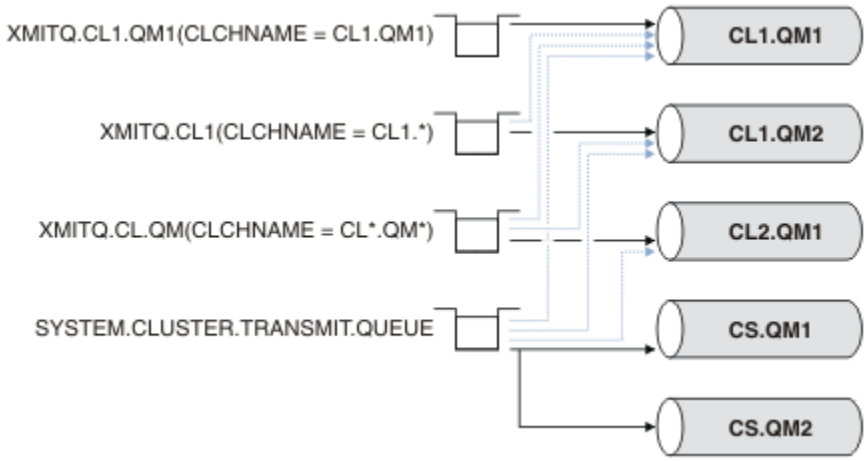


図 *38.* 複数のクラスター送信側チャネル

CS.QM1 が停止すると、このクラスター送信側チャネルの伝送キューに配置されたメッセージが検査されます。 別のチャネルで転送可能なメッセージは、ワークロード・バランシング・アルゴリズムによって再処理されます。 それらの相関 ID は、代替クラスター送信側チャネル名に再設定されます。 代替クラスター送信側チャネルが CS.QM2 の場合、メッセージは SYSTEM.CLUSTER.TRANSMIT.QUEUE に残ります。 代替チャネルが CL1.QM1 の場合には、ワークロード・バランシング・アルゴリズムはメッセージを XMITQ.CL1.QM1 に転送します。 クラスター送信側チャネルが再始動すると、新しいメッセージおよび異なるクラスター送信側チャネルのフラグが立てられなかったメッセージは、このチャネルによって再び転送されます。

実行中のシステムで、伝送キューとクラスター送信側チャネルの関連付けを変更することもできます。 その場合、伝送キューの **CLCHNAME** パラメーターを変更するか、または **DEFCLXQ** キュー・マネージャー・パラメーターを変更します。 変更によって影響を受けるチャネルが再始動すると、そのチャネルは伝送キュー切り替えプロセスを開始します (295 ページの『クラスター送信側チャネルを異なる伝送キューに切り替えるプロセスの仕組み』を参照)。

チャネルを再開すると、伝送キューを切り替えるプロセスが開始します。 チャネルが停止すると、ワークロードの再バランシング・プロセスが開始します。 この 2 つのプロセスは、並列に実行できます。

単純なケースでは、クラスター送信側チャネルが停止しても、キューのあらゆるメッセージを転送するクラスター送信側チャネルを変更するための再バランシング・プロセスは発生しません。 このケースは、メッセージを正しい宛先に転送できるクラスター送信側チャネルが他にない場合です。 メッセージをその宛先に転送する代替クラスター送信側チャネルがないため、クラスター送信側チャネルが停止した後も、メッセージには同じクラスター送信側チャネルのフラグが立てられたままになります。 切り替えが保留されている場合、チャネルが開始すると、切り替えプロセスはメッセージを異なる宛先キューに移動します。 その宛先キューで、メッセージは同じクラスター送信側チャネルによって処理されます。

より複雑なケースでは、同じ宛先へのメッセージを複数のクラスター送信側チャネルが処理できます。 伝送キュー切り替えをトリガーするには、クラスター送信側チャネルを停止してから再始動します。 多くの場合、チャネルを再始動するまでには、ワークロード・バランシング・アルゴリズムが、すでにメッセージを元の伝送キューから別のクラスター送信側チャネルが処理する別の伝送キューに移動し終えています。 別のクラスター送信側チャネルで転送できないメッセージのみが、新しい伝送キューに転送されるために残ります。 チャネルが短時間で再始動された場合には、ワークロード・バランシング・アルゴリズムによって転送できなかったメッセージが残ることもあります。 その場合、残されたメッセージのなかには、ワークロード・バランシング・プロセスによって切り替えられるものも、伝送キューを切り替えるプロセスによって切り替えられるものもあります。

関連概念

クラスター・チャネル

クラスター化: 複数のクラスター伝送キューの使用によるアプリケーションの分離

646 ページの『ログのサイズの計算』
キュー・マネージャーが必要とするログのサイズを見積もります。

関連タスク
クラスター化: クラスター伝送キューの構成方法の計画
336 ページの『ゲートウェイ・キュー・マネージャーを使用した 2 つのオーバーラップするクラスターの作成』
このタスクの手順に従って、ゲートウェイ・キュー・マネージャーを使用したオーバーラップするクラスターを構成します。 このクラスターを、以降で説明するあるメッセージへのメッセージをクラスター内の他のアプリケーションへのメッセージから分離する例の開始点として使用します。

315 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』
以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。 クラスター・キューおよびトピックへのメッセージは、複数のクラスター伝送キューを使用して転送されます。

343 ページの『クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』
ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後 このソリューションは、追加のクラスター伝送キューを使用して、メッセージ・トラフィックをクラスター内の 1 つのキュー・マネージャーに分離します。

347 ページの『クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』
ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後、これは、追加のクラスターを使用して、特定のクラスター・キューにメッセージを分離します。

クラスター内での通信の確立

送達するメッセージがある場合に、通信チャネルを開始するには、チャネル開始プログラムが必要です。チャネル・リスナーは、チャネルの相手側が開始されてメッセージが受信されるのを待ちます。

始める前に

クラスター内にあるキュー・マネージャー間の通信を確立するには、サポートされている通信プロトコルのいずれかを使用してリンクを構成します。 サポートされるプロトコルは、以下のとおりです。

- TCP または LU 6.2 (すべてのプラットフォーム)
- Windows NetBIOS または SPX (Windows システム)

分散キューイングの場合と同様に、この設定処理でもチャネル開始プログラムおよびチャネル・リスナーを設定する必要があります。

このタスクについて

すべてのクラスター・キュー・マネージャーには、システム定義の開始キュー SYSTEM.CHANNEL.INITQ をモニターするためのチャネル・イニシエーターが必要です。 SYSTEM.CHANNEL.INITQ は、クラスター伝送キューを含むすべての伝送キューの開始キューです。

各キュー・マネージャーには、チャネル・リスナーが必要です。 チャネル・リスナー・プログラムは、着信ネットワーク要求を待機し、必要な場合は、該当する受信側チャネルを起動します。 チャネル・リスナーの実装方法は、プラットフォームによって異なりますが、一部の機能は共通しています。

すべての IBM MQ プラットフォームで、リスナーは **START LISTENER** コマンドを使用して起動可能です。

Multi Multiplatforms では、キュー・マネージャーと同時にリスナーを自動的に開始できます。 リスナーを自動的に起動するには、LISTENER オブジェクトの CONTROL 属性を QMGR または STARTONLY に設定してください。

z/OS 非共用リスナー・ポート(INDISP(QMGR))は、z/OS の CLUSRCVR チャネルおよび CLUSSDR チャネルの場合は z/OS に使用する必要があります。

手順

1. チャネル・イニシエーターを開始します。
   - z/OS z/OS では、キュー・マネージャーごとにチャネル・イニシエーターが 1 つずつあり、別々のアドレス・スペースとして実行されます。 **MQSC** START CHINIT コマンドを使用してチャネル開始プログラムを開始しますが、このコマンドは、キュー・マネージャーの始動時に発行します。
   - ALW AIX, Linux, and Windows で、キュー・マネージャー属性 SCHINIT が QMGR に設定されている場合は、キュー・マネージャーを始動すると、チャネル開始プログラムが自動的に開始されます。 それ以外の場合は、**runmqsc** START CHINIT コマンドまたは **runmqchi** 制御コマンドを使用することで開始できます。
   - IBM i IBM i で、キュー・マネージャー属性 SCHINIT が QMGR に設定されている場合は、キュー・マネージャーを始動すると、チャネル開始プログラムが自動的に開始されます。 それ以外の場合は、**runmqsc** START CHINIT コマンドまたは **runmqchi** 制御コマンドを使用することで開始できます。
2. チャネル・リスナーを開始します。
   - z/OS z/OS では、IBM MQ 提供のチャネル・リスナー・プログラムを使用します。 IBM MQ チャネル・リスナーを起動するには、**MQSC** コマンドの START LISTENER を使用しますが、このコマンドは、チャネル開始プログラムの始動時に発行します。 以下に例を示します。

     ```
     START LISTENER PORT(1414) TRPTYPE(TCP)
     ```

     または:

     ```
     START LISTENER LUNAME(LONDON.LUNAME) TRPTYPE(LU62)
     ```

     キュー共有グループのメンバーは、キュー・マネージャーごとにリスナーを使用する代わりに、共有リスナーを使用できます。 共有リスナーをクラスターと併用しないでください。 具体的に言うと、CLUSRCVR チャネルの CONNAME を キュー共有グループの共有リスナーのアドレスに使用しないでください。 使用すると、キュー・マネージャーは、定義していないキューに関するメッセージを受信する可能性があります。
   - IBM i IBM i では、IBM MQ 提供のチャネル・リスナー・プログラムを使用します。 IBM MQ チャネル・リスナーを起動するには、**CL** コマンドの STRMQMLSR を使用します。 以下に例を示します。

     ```
     STRMQMLSR MQMNAME(QM1) PORT(1414)
     ```

   - Windows Windows では、IBM MQ 提供のチャネル・リスナー・プログラムまたはオペレーティング・システム提供の機能を使用します。

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

     ```
     RUNMQLSR -t tcp -p 1414 -m QM1
     ```

   - Linux AIX AIX and Linux では、IBM MQ 提供のチャネル・リスナー・プログラムか、オペレーティング・システム提供の機能 (例えば、TCP 通信の場合には **inetd**) のどちらかを使用します。

IBM MQ 提供のチャネル・リスナーを起動するときには **runmqlsr** コマンドを使用します。 以下に例を示します。

```
runmqlsr -t tcp -p 1414 -m QM1
```

inetd によってチャネルを起動する場合には、次の 2 つのファイルを構成してください。

a. /etc/services ファイルを編集します。 スーパーユーザーまたはルート・ユーザーとしてログインする必要があります。 このファイルに次の行がない場合は、それを以下のように追加します。

```
MQSeries    1414/tcp   # WebSphere MQ channel listener
```

ここで、1414 は、IBM MQ に必要なポート番号です。 このポート番号は変更できますが、送信側で指定したポート番号と一致しなければなりません。

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

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

MQ_INSTALLATION_PATH は、IBM MQ のインストール先の上位ディレクトリーに置き換えられます。

inetd により構成ファイルが再度読み取られると、新しい設定が有効になります。 次に、ルート・ユーザー ID により、以下に示すコマンドを実行します。

AIX On AIX:

```
refresh -s inetd
```

Linux Linux の場合:

a. 以下のコマンドにより、**inetd** のプロセス ID を調べる

```
ps -ef | grep inetd
```

b. 該当するコマンドを実行する

Linux の場合:

```
kill -1 inetd processid
```

新規クラスターのセットアップ

クラスター例をセットアップするには、これらの手順に従ってください。 クラスターを設定する方法は、TCP/IP、LU 6.2、および単一の伝送キューまたは複数の伝送キューを使用する場合で別々に説明されています。 あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信することによって、クラスターの動作をテストします。

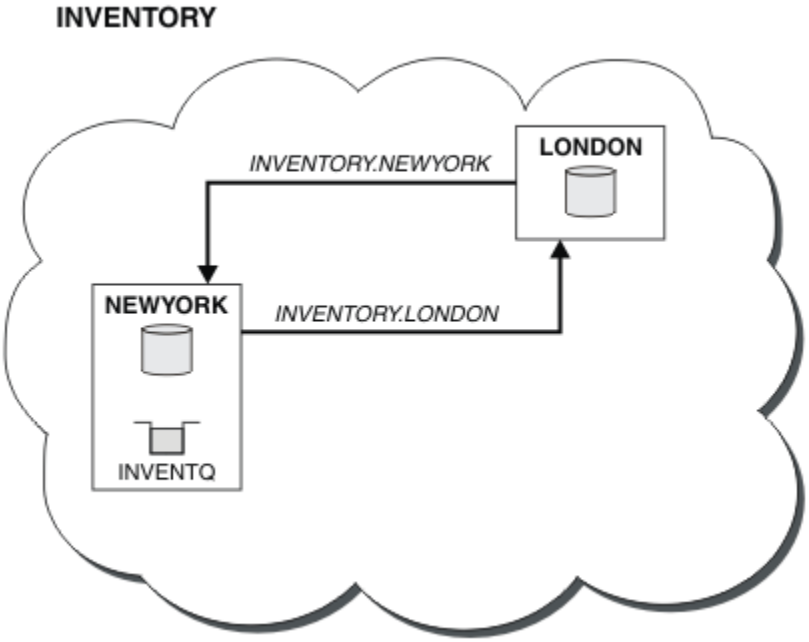
始める前に

- 上記の手順に従う代わりに、IBM MQ Explorer に付属のウィザードの 1 つを使用して、このタスクで作成されるものと同様のクラスターを作成することができます。 「キュー・マネージャー・クラスター」フォルダーを右クリックしてから、「**新規**」 > 「**キュー・マネージャー・クラスター**」をクリックして、ウィザードに示される指示に従います。
- クラスタのセットアップ手順を理解するための背景情報については、クラスタ・チャネルとリスナー 288 ページの『クラスター・キューの定義』を参照してください。

このタスクについて

これから、あるチェーン店で使用する新しい IBM MQ ネットワークを セットアップします。 このチェーン店には、ロンドンとニューヨークにそれぞれ 1 つの支店があります。 各支店のデータとアプリケーションは、各キュー・マネージャーを実行するシステムがそれぞれホストします。 この 2 つのキュー・マネージャーの名前を LONDON および NEWYORK とします。 在庫管理アプリケーションは、ニューヨーク支店のシステムで実行します。このシステムは、キュー・マネージャー NEWYORK に接続されます。 このアプリケーションは、NEWYORK がホストしている INVENTQ というキューにメッセージが届くと起動されるようにします。 キュー・マネージャー LONDON および NEWYORK を INVENTORY というクラスターにリンクさせて、これらのキュー・マネージャーが INVENTQ キューにメッセージを書き込むことができるようにします。

このクラスターは以下のようになります。



クラスター内の各キュー・マネージャーを、異なるクラスター伝送キューを使用してクラスター内の他のキュー・マネージャーにメッセージを送信するように構成できます。

クラスターをセットアップする手順は、トランスポート・プロトコル、伝送キューの数、またはプラットフォームによって少し異なります。 3 つの組み合わせから選択できます。 どの組み合わせでも検査の手順は同じです。

INVENTORY は小さいクラスターです。 クラスターの概念を具体的に示すのに適しています。 このクラスターについて理解する上で重要なのは、このクラスターには今後規模を拡大できる余地があるということです。

手順

- 303 ページの『キュー・マネージャーごとに 1 つの伝送キューを持ち、TCP/IP を使用するクラスターのセットアップ』
- 306 ページの『キュー・マネージャーごとに複数の伝送キューを持ち TCP/IP を使用するクラスターのセットアップ』
- 309 ページの『z/OS での LU 6.2 を使用するクラスターのセットアップ』
- 311 ページの『クラスターの検査』

関連概念

クラスター

クラスター化と分散キューイングとの比較

クラスターのコンポーネント

関連タスク

287 ページの『キュー・マネージャー・クラスターの構成』
クラスターによって提供されるキュー・マネージャーを相互接続するためのメカニズムにより、初期構成と継続的な管理を簡単に行えます。 クラスター・コンポーネントを定義し、クラスターを作成および管理できます。

キュー・マネージャーごとに *1* つの伝送キューを持ち、*TCP/IP* を使用するクラスターのセットアップ

これは、単純なクラスターのさまざまな構成について説明している 3 つのトピックのうちの 1 つです。

始める前に

作成されるクラスターの概要については、 301 ページの『新規クラスターのセットアップ』を参照してください。

キュー・マネージャー属性 **DEFCLXQ** はデフォルト値 SCTQ のままにしておいてください。

このタスクについて

マルチプラットフォームでトランスポート・プロトコル TCP/IP を使用するクラスターをセットアップするには、以下のステップに従います。 z/OS z/OS では、ステップ 304 ページの『4』でリスナーを定義するのではなく、1010 ページの『z/OS での TCP 接続の定義』の説明に従って TCP/IP 接続をセットアップする必要があります。 それ以外の場合、エラー・メッセージがキュー・マネージャーのエラー・ログではなくコンソールに書き込まれることを除いて、手順は z/OS と同じです。

手順

1. クラスターの編成およびクラスター名を決める。

   この例では、LONDON、NEWYORK という 2 つのキュー・マネージャーをリンクさせてクラスターを構築します。 2 つのキュー・マネージャーのみが含まれるクラスターは、分散キューイングを使用するネットワークに比べて利点はごくわずかです。 しかし、スタートするには有効な方法であり、将来拡張できる余地も備えています。 新しい支店をオープンするときに、新しいキュー・マネージャーをクラスターに簡単に追加できます。 新しいキュー・マネージャーを追加しても、既存のネットワークが切断されるわけではありません (313 ページの『クラスターにキュー・マネージャーを追加する』を参照)。

   当分の間、実行するアプリケーションは在庫管理アプリケーションのみとします。 クラスター名は INVENTORY とします。
2. 完全リポジトリーを保持するキュー・マネージャーを決定する。

   クラスター内の 1 つ以上の (できれば 2 つの) キュー・マネージャーに完全リポジトリーを格納するようにしてください。 この例では、キュー・マネージャーは 2 つだけ (LONDON と NEWYORK) であるため、両方のキュー・マネージャーに完全リポジトリーを格納します。

   a. 以降の手順については、任意の順序で実行できます。

   b. ステップを進めていくうちに、キュー・マネージャー・ログに警告メッセージが書き込まれることがあります。 このメッセージは、まだ追加する必要がある定義が欠落していることの結果です。

   ```
   Examples of the responses to the commands are shown in a box
   like this after each step in this task.
   These examples show the responses returned by IBM MQ for AIX.
   The responses vary on other platforms.
   ```

   c. 以降の手順に進む前に、上述のキュー・マネージャーが 2 つとも起動していることを確認してください。
3. キュー・マネージャー定義を変更してリポジトリー定義を追加する。

完全リポジトリーを保持する各キュー・マネージャーで、次のように ALTER QMGR コマンドを使用して REPOS 属性を指定することにより、ローカル・キュー・マネージャー定義を変更します。

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: IBM MQ queue manager changed.
```

例えば、以下のように入力したとします。

a. `runmqsc LONDON`

b. `ALTER QMGR REPOS(INVENTORY)`

LONDON が完全リポジトリーに変更されます。

4. リスナーを定義します。

   クラスター内のすべてのキュー・マネージャーに対して、他のキュー・マネージャーからのネットワーク要求を受け入れるリスナーを定義します。 LONDON キュー・マネージャーに以下のコマンドを発行します。

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

   CONTROL 属性は、キュー・マネージャーの開始および停止と同時に、このリスナーも開始および停止するようにします。

   リスナーは定義と同時に開始されないため、最初は以下の MQSC コマンドを使用して手動で開始する必要があります。

   ```
   START LISTENER(LONDON_LS)
   ```

   クラスター内の他のすべてのキュー・マネージャーに対して同様のコマンドを発行します。ただし、各リスナーの名前は変更します。

   リスナーで示すように、これらのリスナーを定義する方法はいくつかあります。

5. LONDON キュー・マネージャーの CLUSRCVR チャネルを定義する。

   クラスター内のすべてのキュー・マネージャーで、キュー・マネージャーがメッセージを受信できるクラスター受信側チャネルを定義します。 クラスター受信側チャネル:CLUSRCVR を参照してください。 CLUSRCVR チャネルは、キュー・マネージャーの接続名を定義します。 接続名はリポジトリーに格納され、それを他のキュー・マネージャーが参照できます。 CLUSTER キーワードは、キュー・マネージャーがクラスター内の他のキュー・マネージャーからメッセージを受信できることを示します。

   この例では、チャネル名を INVENTORY.LONDON とし、接続名 (CONNAME) は、キュー・マネージャーがあるマシンのネットワーク・アドレスである LONDON.CHSTORE.COM とします。 ネットワーク・アドレスは、英数字の DNS ホスト名、または IPv4 小数点付き 10 進数形式の IP アドレスとして入力できます。 例えば、192.0.2.0 です。あるいは IPv6 16 進数形式の場合は、例えば 2001:DB8:0204:acff:fe97:2c34:fde0:3485 です。 ポート番号は指定されていないため、 デフォルト・ポート (1414) が使用されます。

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-receiver channel for queue manager LONDON')
   ```

   ```
   1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-receiver channel for queue manager LONDON')
   AMQ8014: WebSphere MQ channel created.
   07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
   ```

6. NEWYORK キュー・マネージャーの CLUSRCVR チャネルを定義する。

チャネル・リスナーがデフォルト・ポート (通常は 1414) を使用し、クラスターに z/OS 上のキュー・マネージャーが含まれていない場合は、次のように CONNAME を省略できます。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

7. LONDON キュー・マネージャーに対して CLUSSDR チャネルを定義します。

   クラスターの各完全リポジトリー・キュー・マネージャーから、他の各完全リポジトリー・キュー・マネージャーへの CLUSSDR チャネルを手動で定義します。 クラスター送信側チャネル:CLUSSDR を参照します。 この場合、キュー・マネージャーは 2 つだけであるため、両方のキュー・マネージャーに完全リポジトリーを格納します。 各キュー・マネージャーには、他方のキュー・マネージャーで定義された CLUSRCVR チャネルを指す、手動で定義された CLUSSDR が必要です。 CLUSSDR 定義で指定するチャネル名は、対応する CLUSRCVR 定義でのチャネル名と一致しなければなりません。 キュー・マネージャーに、同じクラスターのクラスター受信側チャネルとクラスター送信側チャネル両方の定義が設定されると、クラスター送信側チャネルが開始します。

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

   ```
   1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
   AMQ8014: WebSphere MQ channel created.
   07/09/98 13:00:18  Channel program started.
   ```

8. NEWYORK キュー・マネージャーに対して CLUSSDR チャネルを定義します。

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

9. クラスター・キュー INVENTQ を定義する。

   キーワード CLUSTER を指定して、キュー・マネージャー NEWYORK に INVENTQ というキューを定義します。

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
   ```

   ```
   1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
   AMQ8006: WebSphere MQ queue created.
   ```

   CLUSTER というキーワードによって、クラスターはキューを認識できるようになります。 キューは、定義されるとすぐに、クラスター内の他のキュー・マネージャーでも使用できるようになります。 このキューにメッセージを送信する際、キュー・マネージャーは、リモート・キュー定義を作成する必要はありません。

   すべての定義が完了しました。 すべてのプラットフォームで、各キュー・マネージャー上でリスナー・プログラムを開始します。 リスナー・プログラムは、着信ネットワーク要求を待機し、必要な場合にクラスター受信側チャネルを始動します。

次のタスク

これで、 クラスターを検証する準備ができました。

関連タスク

306 ページの『キュー・マネージャーごとに複数の伝送キューを持ち TCP/IP を使用するクラスターのセットアップ』
これは、単純なクラスターのさまざまな構成について説明している 3 つのトピックのうちの 1 つです。

309 ページの『z/OS での LU 6.2 を使用するクラスターのセットアップ』
これは、単純なクラスターのさまざまな構成を説明するツリー・トピックの 1 つです。

キュー・マネージャーごとに複数の伝送キューを持ち *TCP/IP* を使用するクラスターのセットアップ

これは、単純なクラスターのさまざまな構成について説明している 3 つのトピックのうちの 1 つです。

始める前に

作成されるクラスターの概要については、301 ページの『新規クラスターのセットアップ』を参照してください。

このタスクについて

マルチプラットフォームでトランスポート・プロトコル TCP/IP を使用するクラスターをセットアップするには、以下のステップに従います。リポジトリー・キュー・マネージャーは、リポジトリー・キュー・マネージャー同士およびクラスター内の他のキュー・マネージャーとの間でメッセージを送信する際に、別個のクラスター伝送キューを使用するように構成されています。 別々の伝送キューを使用するキュー・マネージャーをクラスターに追加する場合は、315 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』に説明されている手順に従ってください。

手順

1. クラスターの編成およびクラスター名を決める。

   この例では、LONDON、NEWYORK という 2 つのキュー・マネージャーをリンクさせてクラスターを構築します。 2 つのキュー・マネージャーのみが含まれるクラスターは、分散キューイングを使用するネットワークに比べて利点はごくわずかです。 しかし、スタートするには有効な方法であり、将来拡張できる余地も備えています。 新しい支店をオープンするときに、新しいキュー・マネージャーをクラスターに簡単に追加できます。 新しいキュー・マネージャーを追加しても、既存のネットワークが切断されるわけではありません (313 ページの『クラスターにキュー・マネージャーを追加する』を参照)。

   当分の間、実行するアプリケーションは在庫管理アプリケーションのみとします。 クラスター名は INVENTORY とします。

2. 完全リポジトリーを保持するキュー・マネージャーを決定する。

   クラスター内の 1 つ以上の (できれば 2 つの) キュー・マネージャーに完全リポジトリーを格納するようにしてください。 この例では、キュー・マネージャーは 2 つだけ (LONDON と NEWYORK) であるため、両方のキュー・マネージャーに完全リポジトリーを格納します。

   a. 以降の手順については、任意の順序で実行できます。

   b. ステップを進めていくうちに、キュー・マネージャー・ログに警告メッセージが書き込まれることがあります。 このメッセージは、まだ追加する必要がある定義が欠落していることの結果です。

   ```
   Examples of the responses to the commands are shown in a box
   like this after each step in this task.
   These examples show the responses returned by IBM MQ for AIX.
   The responses vary on other platforms.
   ```

   c. 以降の手順に進む前に、上述のキュー・マネージャーが 2 つとも起動していることを確認してください。

3. キュー・マネージャー定義を変更してリポジトリー定義を追加する。

   完全リポジトリーを保持する各キュー・マネージャーで、次のように ALTER QMGR コマンドを使用して REPOS 属性を指定することにより、ローカル・キュー・マネージャー定義を変更します。

   ```
   ALTER QMGR REPOS(INVENTORY)
   ```

   ```
   1 : ALTER QMGR REPOS(INVENTORY)
   AMQ8005: IBM MQ queue manager changed.
   ```

例えば、以下のように入力したとします。

a. `runmqsc LONDON`

b. `ALTER QMGR REPOS(INVENTORY)`

LONDON が完全リポジトリーに変更されます。

4. キュー・マネージャー定義を変更して、宛先ごとに別々のクラスター伝送キューを作成する。

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

クラスターに追加するキュー・マネージャーのそれぞれで、個別の伝送キューを使用するかどうかを決定します。 313 ページの『クラスターにキュー・マネージャーを追加する』および 315 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』のトピックを参照してください。

5. リスナーを定義します。

クラスター内のすべてのキュー・マネージャーに対して、他のキュー・マネージャーからのネットワーク要求を受け入れるリスナーを定義します。 LONDON キュー・マネージャーに以下のコマンドを発行します。

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

CONTROL 属性は、キュー・マネージャーの開始および停止と同時に、このリスナーも開始および停止するようにします。

リスナーは定義と同時に開始されないため、最初は以下の MQSC コマンドを使用して手動で開始する必要があります。

```
START LISTENER(LONDON_LS)
```

クラスター内の他のすべてのキュー・マネージャーに対して同様のコマンドを発行します。ただし、各リスナーの名前は変更します。

リスナーで示すように、これらのリスナーを定義する方法はいくつかあります。

6. LONDON キュー・マネージャーの CLUSRCVR チャネルを定義する。

クラスター内のすべてのキュー・マネージャーで、キュー・マネージャーがメッセージを受信できるクラスター受信側チャネルを定義します。 クラスター受信側チャネル:CLUSRCVR を参照してください。 CLUSRCVR チャネルは、キュー・マネージャーの接続名を定義します。 接続名はリポジトリーに格納され、それを他のキュー・マネージャーが参照できます。 CLUSTER キーワードは、キュー・マネージャーがクラスター内の他のキュー・マネージャーからメッセージを受信できることを示します。

この例では、チャネル名を INVENTORY.LONDON とし、接続名 (CONNAME) は、キュー・マネージャーがあるマシンのネットワーク・アドレスである LONDON.CHSTORE.COM とします。 ネットワーク・アドレスは、 英数字の DNS ホスト名、または IPv4 小数点付き 10 進数形式の IP アドレスとして入力できます。 例えば、192.0.2.0 です。あるいは IPv6 16 進数形式の場合は、例えば 2001:DB8:0204:acff:fe97:2c34:fde0:3485 です。 ポート番号は指定されていないため、 デフォルト・ポート (1414) が使用されます。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

7. NEWYORK キュー・マネージャーの CLUSRCVR チャネルを定義する。

チャネル・リスナーがデフォルト・ポート (通常は 1414) を使用し、クラスターに z/OS 上のキュー・マネージャーが含まれていない場合は、次のように CONNAME を省略できます。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

8. LONDON キュー・マネージャーに対して CLUSSDR チャネルを定義します。

   クラスターの各完全リポジトリー・キュー・マネージャーから、他の各完全リポジトリー・キュー・マネージャーへの CLUSSDR チャネルを手動で定義します。 クラスター送信側チャネル:CLUSSDR を参照します。 この場合、キュー・マネージャーは 2 つだけであるため、両方のキュー・マネージャーに完全リポジトリーを格納します。 各キュー・マネージャーには、他方のキュー・マネージャーで定義された CLUSRCVR チャネルを指す、手動で定義された CLUSSDR が必要です。 CLUSSDR 定義で指定するチャネル名は、対応する CLUSRCVR 定義でのチャネル名と一致しなければなりません。 キュー・マネージャーに、同じクラスターのクラスター受信側チャネルとクラスター送信側チャネル両方の定義が設定されると、クラスター送信側チャネルが開始します。

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

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18  Channel program started.
```

9. NEWYORK キュー・マネージャーに対して CLUSSDR チャネルを定義します。

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

10. クラスター・キュー INVENTQ を定義する。

    キーワード CLUSTER を指定して、キュー・マネージャー NEWYORK に INVENTQ というキューを定義します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

    CLUSTER というキーワードによって、クラスターはキューを認識できるようになります。 キューは、定義されるとすぐに、クラスター内の他のキュー・マネージャーでも使用できるようになります。 このキューにメッセージを送信する際、キュー・マネージャーは、リモート・キュー定義を作成する必要はありません。

    すべての定義が完了しました。 すべてのプラットフォームで、各キュー・マネージャー上でリスナー・プログラムを開始します。 リスナー・プログラムは、着信ネットワーク要求を待機し、必要な場合にクラスター受信側チャネルを始動します。

次のタスク

これで、 クラスターを検証する準備ができました。

関連タスク

303 ページの『キュー・マネージャーごとに 1 つの伝送キューを持ち、TCP/IP を使用するクラスターのセットアップ』
これは、単純なクラスターのさまざまな構成について説明している 3 つのトピックのうちの 1 つです。

309 ページの『z/OS での LU 6.2 を使用するクラスターのセットアップ』
これは、単純なクラスターのさまざまな構成を説明するツリー・トピックの 1 つです。

z/OS での LU 6.2 を使用するクラスターのセットアップ

これは、単純なクラスターのさまざまな構成を説明するツリー・トピックの 1 つです。

始める前に

作成されるクラスターの概要については、 301 ページの『新規クラスターのセットアップ』を参照してください。

手順

1. クラスターの編成およびクラスター名を決める。

   この例では、LONDON、NEWYORK という 2 つのキュー・マネージャーをリンクさせてクラスターを構築します。 2 つのキュー・マネージャーのみが含まれるクラスターは、分散キューイングを使用するネットワークに比べて利点はごくわずかです。 しかし、スタートするには有効な方法であり、将来拡張できる余地も備えています。 新しい支店をオープンするときに、新しいキュー・マネージャーをクラスターに簡単に追加できます。 新しいキュー・マネージャーを追加しても、既存のネットワークが切断されるわけではありません (313 ページの『クラスターにキュー・マネージャーを追加する』を参照)。

   当分の間、実行するアプリケーションは在庫管理アプリケーションのみとします。 クラスター名は INVENTORY とします。

2. 完全リポジトリーを保持するキュー・マネージャーを決定する。

   クラスター内の 1 つ以上の (できれば 2 つの) キュー・マネージャーに完全リポジトリーを格納するようにしてください。 この例では、キュー・マネージャーは 2 つだけ (LONDON と NEWYORK) であるため、両方のキュー・マネージャーに完全リポジトリーを格納します。

   a. 以降の手順については、任意の順序で実行できます。

   b. ステップを進めていくうちに、z/OS システム・コンソールに警告メッセージが書き込まれることがあります。 このメッセージは、まだ追加する必要がある定義が欠落していることの結果です。

   c. 以降の手順に進む前に、上述のキュー・マネージャーが 2 つとも起動していることを確認してください。

3. キュー・マネージャー定義を変更してリポジトリー定義を追加する。

   完全リポジトリーを保持する各キュー・マネージャーで、次のように ALTER QMGR コマンドを使用して REPOS 属性を指定することにより、ローカル・キュー・マネージャー定義を変更します。

   ```
   ALTER QMGR REPOS(INVENTORY)
   ```

   ```
   1 : ALTER QMGR REPOS(INVENTORY)
   AMQ8005: IBM MQ queue manager changed.
   ```

   例えば、以下のように入力したとします。

   a. `runmqsc LONDON`

   b. `ALTER QMGR REPOS(INVENTORY)`

   LONDON が完全リポジトリーに変更されます。

4. リスナーを定義します。

   z/OS z/OS のチャネル・イニシエーターおよび 1013 ページの『LU 6.2 での受信』を参照してください。

   リスナーは定義と同時に開始されないため、最初は以下の MQSC コマンドを使用して手動で開始する必要があります。

   ```
   START LISTENER(LONDON_LS)
   ```

クラスター内の他のすべてのキュー・マネージャーに対して同様のコマンドを発行します。ただし、各リスナーの名前は変更します。

5. LONDON キュー・マネージャーの CLUSRCVR チャネルを定義する。

   クラスター内のすべてのキュー・マネージャーで、キュー・マネージャーがメッセージを受信できるクラスター受信側チャネルを定義します。 クラスター受信側チャネル:CLUSRCVR を参照してください。 CLUSRCVR チャネルは、キュー・マネージャーの接続名を定義します。 接続名はリポジトリーに格納され、それを他のキュー・マネージャーが参照できます。 CLUSTER キーワードは、キュー・マネージャーがクラスター内の他のキュー・マネージャーからメッセージを受信できることを示します。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

6. NEWYORK キュー・マネージャーの CLUSRCVR チャネルを定義する。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager NEWYORK')
```

7. LONDON キュー・マネージャーに対して CLUSSDR チャネルを定義します。

   クラスターの各完全リポジトリー・キュー・マネージャーから、他の各完全リポジトリー・キュー・マネージャーへの CLUSSDR チャネルを手動で定義します。 クラスター送信側チャネル:CLUSSDR を参照します。 この場合、キュー・マネージャーは 2 つだけであるため、両方のキュー・マネージャーに完全リポジトリーを格納します。 各キュー・マネージャーには、他方のキュー・マネージャーで定義された CLUSRCVR チャネルを指す、手動で定義された CLUSSDR が必要です。 CLUSSDR 定義で指定するチャネル名は、対応する CLUSRCVR 定義でのチャネル名と一致しなければなりません。 キュー・マネージャーに、同じクラスターのクラスター受信側チャネルとクラスター送信側チャネル両方の定義が設定されると、クラスター送信側チャネルが開始します。

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

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18  Channel program started.
```

8. NEWYORK キュー・マネージャーに対して CLUSSDR チャネルを定義します。

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

9. クラスター・キュー INVENTQ を定義する。

   キーワード CLUSTER を指定して、キュー・マネージャー NEWYORK に INVENTQ というキューを定義します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

CLUSTER というキーワードによって、クラスターはキューを認識できるようになります。 キューは、定義されるとすぐに、クラスター内の他のキュー・マネージャーでも使用できるようになります。 このキューにメッセージを送信する際、キュー・マネージャーは、リモート・キュー定義を作成する必要はありません。

すべての定義が完了しました。 すべてのプラットフォームで、各キュー・マネージャー上でリスナー・プログラムを開始します。 リスナー・プログラムは、着信ネットワーク要求を待機し、必要な場合にクラスター受信側チャネルを始動します。

次のタスク

これで、 クラスターを検証する準備ができました。

関連タスク

303 ページの『キュー・マネージャーごとに 1 つの伝送キューを持ち、TCP/IP を使用するクラスターのセットアップ』
これは、単純なクラスターのさまざまな構成について説明している 3 つのトピックのうちの 1 つです。

306 ページの『キュー・マネージャーごとに複数の伝送キューを持ち TCP/IP を使用するクラスターのセットアップ』
これは、単純なクラスターのさまざまな構成について説明している 3 つのトピックのうちの 1 つです。

クラスターの検査

ピア・トピックでは、単純なクラスターの 3 つの異なる構成について説明します。 このトピックでは、クラスターを検査する方法について説明します。

始める前に

このトピックでは、以下のいずれかのタスクを使用して作成したクラスターを検証することを想定しています。

- 303 ページの『キュー・マネージャーごとに 1 つの伝送キューを持ち、TCP/IP を使用するクラスターのセットアップ』.
- 306 ページの『キュー・マネージャーごとに複数の伝送キューを持ち TCP/IP を使用するクラスターのセットアップ』.
- 309 ページの『z/OS での LU 6.2 を使用するクラスターのセットアップ』.

作成されたクラスターの概要については、 301 ページの『新規クラスターのセットアップ』を参照してください。

このタスクについて

クラスターの検査は、以下の方法から 1 つ以上を使って行うことができます。

1. クラスター属性およびチャネル属性を表示する管理コマンドを実行する。
2. クラスター・キューに対してメッセージを送受信するサンプル・プログラムを実行する。
3. クラスター・キューに要求メッセージを送信し、クラスター化されていない応答キューに応答メッセージで応答する独自のプログラムを作成する。

手順

DISPLAY **runmqsc** コマンドを実行して、クラスターを検査する。

表示される応答は、以下の手順の応答のようになります。

1. NEWYORK キュー・マネージャーから、**DISPLAY CLUSQMGR** コマンドを実行する。

```
dis clusqmgr(*)
```

```
1 : dis clusqmgr(*)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(NEWYORK)       CLUSTER(INVENTORY)
CHANNEL(INVENTORY.NEWYORK)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(LONDON)       CLUSTER(INVENTORY)
CHANNEL(INVENTORY.LONDON)
```

2. NEWYORK キュー・マネージャーから、**DISPLAY CHANNEL STATUS** コマンドを実行する。

```
dis chstatus(*)
```

```
1 : dis chstatus(*)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.NEWYORK)  XMITQ( )
CONNAME(192.0.2.0)      CURRENT
CHLTYPE(CLUSRCVR)       STATUS(RUNNING)
RQMNAME(LONDON)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.LONDON) XMITQ(SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON)
CONNAME(192.0.2.1)      CURRENT
CHLTYPE(CLUSSDR)       STATUS(RUNNING)
RQMNAME(LONDON)
```

amqsput を使用して、2 つのキュー・マネージャーの間でメッセージを送信します。

3. LONDON で、コマンド **amqsput INVENTQ LONDON** を実行します。

   何らかのメッセージを入力して、ブランク行を 1 行追加します。

4. NEWYORK で、コマンド **amqsget INVENTQ NEWYORK** を実行します。

   これで、LONDON で入力したメッセージが表示されるようになりました。 プログラムは 15 秒後に終了します。

独自のプログラムを使用して、2 つのキュー・マネージャーの間でメッセージを送信する。

次の手順では、キュー・マネージャー LONDON から NEWYORK の INVENTQ キューにメッセージを書き込み、そのメッセージに対する応答を LONDON_reply というキューに受信します。

5. LONDON で、クラスター・キューにメッセージを書き込む。
   a) LONDON_reply という名前のローカル・キューを定義します。
   b) MQOPEN オプションを MQOO_OUTPUT に設定します。
   c) MQOPEN 呼び出しを発行して、キュー INVENTQ をオープンします。
   d) メッセージ記述子の *ReplyToQ* 名を'LONDON_reply に設定する。
   e) MQPUT 呼び出しを発行して、メッセージを書き込みます。
   f) メッセージをコミットします。
6. NEWYORK で、クラスター・キューにあるメッセージを受信して、応答キューに応答を書き込む。
   a) MQOPEN オプションを MQOO_BROWSE に設定します。
   b) MQOPEN 呼び出しを発行して、キュー INVENTQ をオープンします。
   c) MQGET 呼び出しを発行して、 INVENTQ からメッセージを取得します。
   d) メッセージ記述子から *ReplyToQ* 名を取り出す。
   e) オブジェクト記述子の ObjectName フィールドに *ReplyToQ* の名前を入れる。
   f) MQOPEN オプションを MQOO_OUTPUT に設定します。
   g) MQOPEN 呼び出しを発行して、キュー・マネージャー LONDON で LONDON_reply を開きます。
   h) MQPUT 呼び出しを発行して、メッセージを LONDON_reply に書き込みます。
7. LONDON で、応答を受け取ります。
   a) MQOPEN オプションを MQOO_BROWSE に設定します。

b) MQOPEN 呼び出しを発行して、キュー LONDON_reply をオープンします。

c) MQGET 呼び出しを発行して、 LONDON_reply からメッセージを取得します。

クラスターにキュー・マネージャーを追加する

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

始める前に

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- 301 ページの『新規クラスターのセットアップ』での説明に従って、INVENTORY クラスターを設定しました。 このクラスターは、LONDON、NEWYORK という 2 つのキュー・マネージャーで構成されています。そして、両方のキュー・マネージャーに完全リポジトリーが格納されています。
- キュー・マネージャー PARIS は、最初のプライマリー・インストールが所有しています。 所有関係が異なる場合は、**setmqenv** コマンドを実行して、PARIS が属するインストールのコマンド環境をセットアップしてください。
- TCP 接続は、3 つすべてのシステム間で存在します。キュー・マネージャーは、キュー・マネージャーの制御下で開始される TCP リスナーを使用して構成されます。

このタスクについて

1. その後、このチェーン店ではパリに新しい支店をオープンすることになったため、このクラスターに PARIS というキュー・マネージャーの追加が必要になりました。
2. キュー・マネージャー PARIS は、INVENTQ キューにメッセージを書き込むことにより、ニューヨーク支店のシステムで実行されているアプリケーションに新しい在庫情報を送信するものとします。

以下の手順に従って、クラスターにキュー・マネージャーを追加します。

手順

1. PARIS が最初に参照する完全リポジトリーを決める。

   クラスター内の各キュー・マネージャーは、いずれかの完全リポジトリーを参照しなければなりません。 キュー・マネージャーは、クラスターに関する情報を完全リポジトリーから収集し、自分用の部分リポジトリーを作成します。 いずれかのリポジトリーを完全リポジトリーとして選択してください。 クラスターに新しいキュー・マネージャーを追加すると、そのキュー・マネージャーは他方のリポジトリーもすぐに認識できるようになります。 キュー・マネージャーに加えられた変更に関する情報は 2 つのリポジトリーに直接送信されます。 ここでは、(単に地理的な理由により) キュー・マネージャー PARIS をキュー・マネージャー LONDON にリンクします。

   注 : キュー・マネージャー PARIS の開始後は、残りの手順は任意の順序で実行できます。

2. キュー・マネージャー PARIS に CLUSRCVR チャネルを定義する。

   クラスター内の各キュー・マネージャーにクラスター受信側チャネルを定義する必要があります。このチャネルにより、キュー・マネージャーはメッセージを受信することができます。 キュー・マネージャー PARIS には、以下のように定義します。

   ```
   DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for queue manager PARIS')
   ```

クラスター受信側チャネルは、このキュー・マネージャーがクラスター INVENTORY 内の他のキュー・マネージャーからメッセージを受信できる状態になったことを通知します。 他のキュー・マネージャーでは、クラスター受信側チャネル INVENTORY.PARIS の送信側用の定義を作成しないでください。 他の定義は、必要な場合に自動的に行われます。 クラスター・チャネルを参照します。

3. z/OS
   IBM MQ for z/OS でチャネル・イニシエーターを開始します。
4. キュー・マネージャー PARIS に CLUSSDR チャネルを定義する。

   完全リポジトリーではないキュー・マネージャーをクラスターに追加する際は、完全リポジトリーとの初期接続を行うためのクラスター送信側チャネルを 1 つだけ定義します。 クラスター送信側チャネル: クラススドルを参照します。

   PARIS に、LONDON.CHSTORE.COM というネットワーク・アドレスが割り当てられているキュー・マネージャー宛ての INVENTORY.LONDON という CLUSSDR チャネルを以下のように定義します。

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from PARIS to repository at LONDON')
   ```

5. オプション: 以前にクラスターから除去したキュー・マネージャーを同じクラスターに追加する場合、今回クラスター・メンバーとして表示されることを確認します。 表示されない場合は、以下の追加の手順を実行します。

   a) 追加するキュー・マネージャーで **REFRESH CLUSTER** コマンドを発行します。

   このステップを行うと、クラスター・チャネルが停止され、クラスター内の他のキュー・マネージャー内で確実に最新である新しい一連のシーケンス番号が、ローカル・クラスター・キャッシュに設定されます。

   ```
   REFRESH CLUSTER(INVENTORY) REPOS(YES)
   ```

   注 : 大規模クラスターでは、処理中のクラスターに **REFRESH CLUSTER** コマンドを使用すると、破壊的な影響を及ぼす恐れがあります。その後、クラスター・オブジェクトが 27 日間隔で対象のキュー・マネージャーすべてに状況の更新を自動的に送信する際にも同様のことが起こり得ます。 大規模クラスターでのリフレッシュはクラスターのパフォーマンスと可用性に影響を与える可能性があるを参照してください。

   b) CLUSSDR チャネルを再始動します。

   (例えば、START CHANNEL コマンドを使用します)。

   c) CLUSRCVR チャネルを再始動します。

タスクの結果

以下の図は、このタスクによってセットアップされるクラスターを示します。

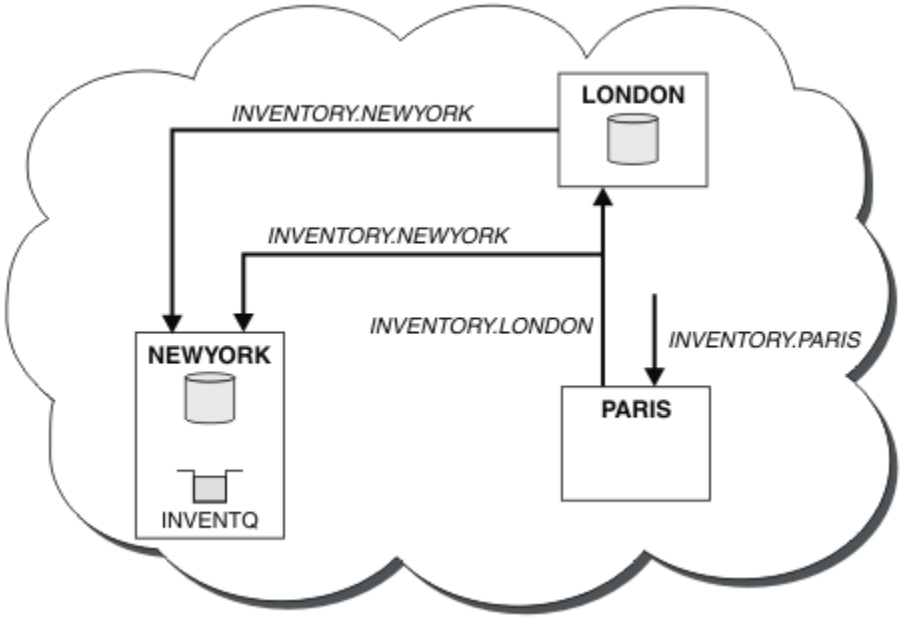


図 *39. 3* つのキュー・マネージャーで構成される *INVENTORY* クラスター

CLUSRCVR チャネルと CLUSSDR チャネルの 2 つを定義するだけで、キュー・マネージャー PARIS がクラスターに追加されます。

これで、キュー・マネージャー PARIS は、キュー・マネージャー LONDON の完全リポジトリーを参照することにより、INVENTQ キューがキュー・マネージャー NEWYORK にホストされていることを認識できるようになります。 パリ支店のシステムでホストされているアプリケーションが INVENTQ キューにメッセージを書き込む場合、キュー・マネージャー PARIS はクラスター受信側チャネル INVENTORY.NEWYORK に接続するクラスター送信側チャネルを自動的に定義します。 このとき、そのアプリケーションのあるキュー・マネージャーの名前を指定して、応答先キューを定義すると、そのアプリケーションはメッセージに対する応答を受信することができます。

クラスターへのキュー・マネージャーの追加: 別個の伝送キュー

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

始める前に

- キュー・マネージャーはクラスターのメンバーではありません。
- クラスターは存在します。つまり、このキュー・マネージャーが直接接続できる完全リポジトリーがあり、そのリポジトリーが使用可能になっています。 クラスターを作成するステップについては、301 ページの『新規クラスターのセットアップ』を参照してください。

このタスクについて

このタスクは 313 ページの『クラスターにキュー・マネージャーを追加する』 (クラスター・メッセージを単一の伝送キューに配置するクラスターにキュー・マネージャーを追加) の代替方法です。

このタスクでは、クライアント送信側チャネルごとに個別のクライアント伝送キューを自動作成するクラスターにキュー・マネージャーを追加します。

キューの定義数を小さくするために、デフォルトでは単一の伝送キューが使用されます。 個々の伝送キューを使用することが有利となるのは、複数の異なるキュー・マネージャーおよび異なるクラスターを宛先とするトラフィックをモニターする必要がある場合です。 また、分離またはパフォーマンスの目標を達成するために、さまざまな宛先へのトラフィックを分離しなければならないこともあります。

手順

1. デフォルト・クラスター・チャネル伝送キュー・タイプを変更します。

   以下のようにして、キュー・マネージャー PARIS を変更します。

   ```
   ALTER QMGR DEFCLXQ(CHANNEL)
   ```

   キュー・マネージャーは、メッセージをクラスター・マネージャーに送信するためのクラスター送信側チャネルを作成するたびに、クラスター伝送キューを作成します。 これは、このクラスター送信側チャネル専用の伝送キューです。 この伝送キューは、永続動的キューです。 これは、モデル・キュー SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE から作成され、SYSTEM.CLUSTER.TRANSMIT.*ChannelName* という名前が付けられます。

   重要 : IBM WebSphere MQ 7.5 より前のバージョンの製品からアップグレードされたキュー・マネージャーで専用の SYSTEM.CLUSTER.TRANSMIT.QUEUES を使用する場合は、SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE の SHARE/NOSHARE オプションが **SHARE** に設定されている必要があります。

2. PARIS が最初に参照する完全リポジトリーを決める。

   クラスター内の各キュー・マネージャーは、いずれかの完全リポジトリーを参照しなければなりません。 キュー・マネージャーは、クラスターに関する情報を完全リポジトリーから収集し、自分用の部分リポジトリーを作成します。 いずれかのリポジトリーを完全リポジトリーとして選択してください。 クラスターに新しいキュー・マネージャーを追加すると、そのキュー・マネージャーは他方のリポジトリーもすぐに認識できるようになります。 キュー・マネージャーに加えられた変更に関する情報は 2 つのリポジトリーに直接送信されます。 ここでは、(単に地理的な理由により) キュー・マネージャー PARIS をキュー・マネージャー LONDON にリンクします。

   注 : キュー・マネージャー PARIS の開始後は、残りの手順は任意の順序で実行できます。

3. キュー・マネージャー PARIS に CLUSRCVR チャネルを定義する。

   クラスター内の各キュー・マネージャーにクラスター受信側チャネルを定義する必要があります。このチャネルにより、キュー・マネージャーはメッセージを受信することができます。 キュー・マネージャー PARIS には、以下のように定義します。

   ```
   DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for queue manager PARIS')
   ```

   クラスター受信側チャネルは、このキュー・マネージャーがクラスター INVENTORY 内の他のキュー・マネージャーからメッセージを受信できる状態になったことを通知します。 他のキュー・マネージャーでは、クラスター受信側チャネル INVENTORY.PARIS の送信側用の定義を作成しないでください。 他の定義は、必要な場合に自動的に行われます。 クラスター・チャネルを参照します。

4. キュー・マネージャー PARIS に CLUSSDR チャネルを定義する。

   完全リポジトリーではないキュー・マネージャーをクラスターに追加する際は、完全リポジトリーとの初期接続を行うためのクラスター送信側チャネルを 1 つだけ定義します。 クラスター送信側チャネル: クラススドルを参照します。

   PARIS に、LONDON.CHSTORE.COM というネットワーク・アドレスが割り当てられているキュー・マネージャー宛ての INVENTORY.LONDON という CLUSSDR チャネルを以下のように定義します。

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from PARIS to repository at LONDON')
   ```

   キュー・マネージャーは、モデル・キュー SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE から自動的に永続動的クラスター伝送キュー SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON を作成します。 伝送キューの CLCHNAME 属性は、INVENTORY.LONDON に設定されます。

タスクの結果

以下の図は、このタスクによってセットアップされるクラスターを示します。

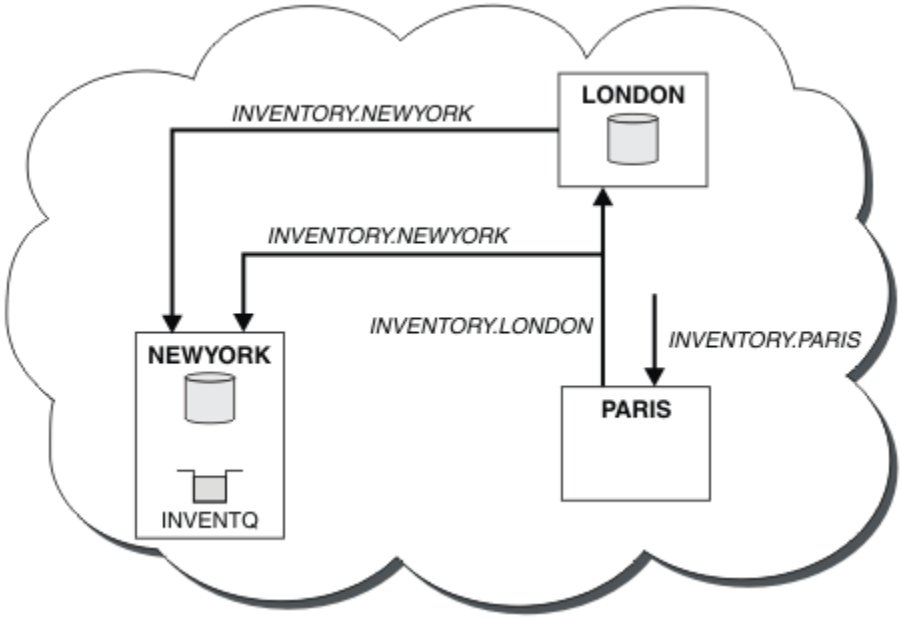


図 *40. 3* つのキュー・マネージャーで構成される *INVENTORY* クラスター

CLUSRCVR チャネルと CLUSSDR チャネルの 2 つを定義するだけで、キュー・マネージャー PARIS がクラスターに追加されます。

これで、キュー・マネージャー PARIS は、キュー・マネージャー LONDON の完全リポジトリーを参照することにより、INVENTQ キューがキュー・マネージャー NEWYORK にホストされていることを認識できるようになります。 パリ支店のシステムでホストされているアプリケーションが INVENTQ キューにメッセージを書き込む場合、キュー・マネージャー PARIS はクラスター受信側チャネル INVENTORY.NEWYORK に接続するクラスター送信側チャネルを自動的に定義します。 このとき、そのアプリケーションのあるキュー・マネージャーの名前を指定して、応答先キューを定義すると、そのアプリケーションはメッセージに対する応答を受信することができます。

関連概念

使用するクラスター伝送キュー・タイプの選択方法

関連タスク

DHCP を使用したクラスターへのキュー・マネージャーの追加

DHCP を使用して、クラスターにキュー・マネージャーを追加します。 この作業では、CLUSRCVR 定義で CONNAME 値を省略することについて説明します。

DHCP を使用したクラスターへのキュー・マネージャーの追加

DHCP を使用して、クラスターにキュー・マネージャーを追加します。 この作業では、CLUSRCVR 定義で CONNAME 値を省略することについて説明します。

始める前に

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

この作業では、以下の 2 つの特殊フィーチャーについて説明します。

- CLUSRCVR 定義で CONNAME 値を省略する機能。
- CLUSSDR 定義で +QMNAME+ を使用する機能。

どちらのフィーチャーも、z/OS では提供されません。

シナリオ

- 301 ページの『新規クラスターのセットアップ』で説明されているように、INVENTORY クラスターがセットアップされています。 このクラスターは、LONDON、NEWYORK という 2 つのキュー・マネージャーで構成されています。そして、両方のキュー・マネージャーに完全リポジトリーが格納されています。
- その後、このチェーン店ではパリに新しい支店をオープンすることになったため、このクラスターに PARIS というキュー・マネージャーの追加が必要になりました。
- キュー・マネージャー PARIS は、INVENTQ キューにメッセージを書き込むことにより、ニューヨーク支店のシステムで実行されているアプリケーションに新しい在庫情報を送信するものとします。
- 各支店のシステム間にはネットワークの接続性があるものとします。
- ネットワーク・プロトコルは、TCP です。
- PARIS キュー・マネージャーシステムは DHCP を使用します。したがって、システム再始動時に IP アドレスが変更される可能性があります。
- PARIS および LONDON のシステム間のチャネルは、定義された命名規則に従って命名されます。 この規則には、LONDON にある完全リポジトリー・キュー・マネージャーのキュー・マネージャー名が使用されます。
- PARIS キュー・マネージャーの管理者は、LONDON リポジトリーのキュ ー・マネージャーの名前に関する情報を知りません。 LONDON リポジトリーのキュー・マネージャーの名前は変わる可能性があります。

このタスクについて

以下の手順に従って、DHCP を使用して、クラスターにキュー・マネージャーを追加します。

手順

1. PARIS が最初に参照する完全リポジトリーを決める。

   クラスター内の各キュー・マネージャーは、いずれかの完全リポジトリーを参照しなければなりません。 キュー・マネージャーは、クラスターに関する情報を完全リポジトリーから収集し、自分用の部分リポジトリーを作成します。 いずれかのリポジトリーを完全リポジトリーとして選択してください。 クラスターに新しいキュー・マネージャーを追加すると、そのキュー・マネージャーは他方のリポジトリーもすぐに認識できるようになります。 キュー・マネージャーに加えられた変更に関する情報は 2 つのリポジトリーに直接送信されます。 ここでは、(単に地理的な理由により) キュー・マネージャー PARIS をキュー・マネージャー LONDON にリンクすることにします。

   注: キュー・マネージャー PARIS の開始後は、残りの手順は任意の順序で実行できます。

2. キュー・マネージャー PARIS に CLUSRCVR チャネルを定義する。

   クラスター内の各キュー・マネージャーにクラスター受信側チャネルを定義する必要があります。このチャネルにより、キュー・マネージャーはメッセージを受信することができます。 キュー・マネージャー PARIS には、以下のように定義します。

   ```
   DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR)
   TRPTYPE(TCP) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for queue manager PARIS')
   ```

   クラスター受信側チャネルは、このキュー・マネージャーがクラスター INVENTORY 内の他のキュー・マネージャーからメッセージを受信できる状態になったことを通知します。 クラスター受信側チャネルで CONNAME を指定する必要はありません。 システムから接続名を検出するよう IBM MQ に要求することができます。そのためには CONNAME を省略するか、または CONNAME(' ') を指定します。 IBM MQ は、システムの現行 IP アドレスを使用して、CONNAME 値を生成します。 CONNAME を参照してください。 このキュー・マネージャーのクラスター受信側チャネル INVENTORY.PARIS に対応する送信側を、他のキュー・マネージャーに定義する必要はありません。 他の定義は、必要な場合に自動的に行われます。

3. キュー・マネージャー PARIS に CLUSSDR チャネルを定義する。

クラスター内の各キュー・マネージャーにクラスター送信側チャネルを 1 つ定義する必要があります。このチャネルにより、キュー・マネージャーは初期完全リポジトリーにメッセージを送信することができます。 PARIS に、LONDON.CHSTORE.COM というネットワーク・アドレスが割り当てられているキュー・マネージャー宛ての INVENTORY.+QMNAME+ というチャネルを以下のように定義します。

```
DEFINE CHANNEL(INVENTORY.+QMNAME+) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

4. オプション: 以前にクラスターから除去したキュー・マネージャーを同じクラスターに追加する場合、今回クラスター・メンバーとして表示されることを確認します。 表示されない場合は、以下の追加の手順を実行します。

   a) 追加するキュー・マネージャーで **REFRESH CLUSTER** コマンドを発行します。

   このステップを行うと、クラスター・チャネルが停止され、クラスター内の他のキュー・マネージャー内で確実に最新である新しい一連のシーケンス番号が、ローカル・クラスター・キャッシュに設定されます。

   ```
   REFRESH CLUSTER(INVENTORY) REPOS(YES)
   ```

   注 : 大規模クラスターでは、処理中のクラスターに **REFRESH CLUSTER** コマンドを使用すると、破壊的な影響を及ぼす恐れがあります。その後、クラスター・オブジェクトが 27 日間隔で対象のキュー・マネージャーすべてに状況の更新を自動的に送信する際にも同様のことが起こり得ます。 大規模クラスターでのリフレッシュはクラスターのパフォーマンスと可用性に影響を与える可能性があるを参照してください。

   b) CLUSSDR チャネルを再始動します。

   (例えば、START CHANNEL コマンドを使用します)。

   c) CLUSRCVR チャネルを再始動します。

タスクの結果

この作業によってセットアップされるクラスターは 313 ページの『クラスターにキュー・マネージャーを追加する』の場合と同じです。

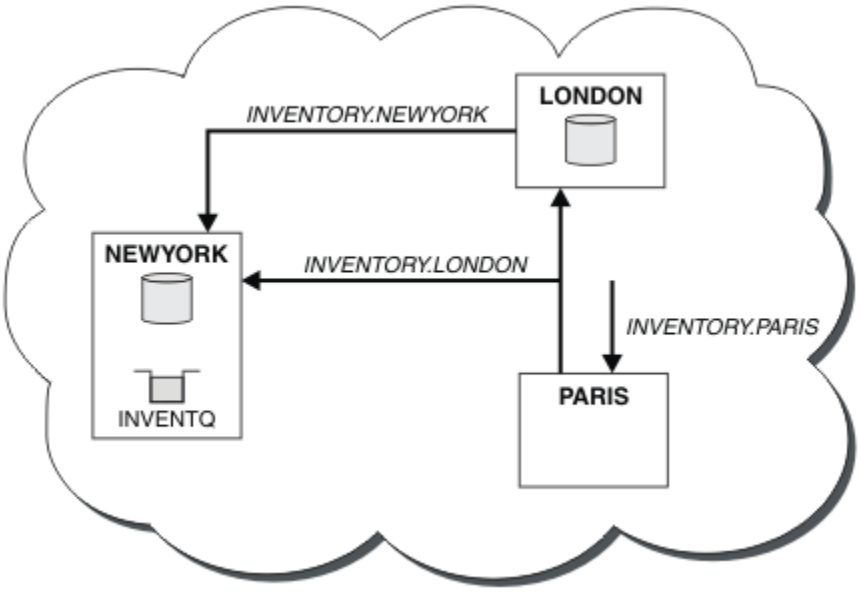


図 *41. 3* つのキュー・マネージャーで構成される *INVENTORY* クラスター

CLUSRCVR チャネルと CLUSSDR チャネルの 2 つを定義するだけで、キュー・マネージャー PARIS がクラスターに追加されます。

PARIS キュー・マネージャーで、ストリング +QMNAME+ を含む CLUSSDR が開始します。 LONDON システム IBM MQ では、+QMNAME+をキュー・マネージャー名(LONDON)に解決します。IBM MQ は、INVENTORY.LONDON というチャネルの定義を、対応するクラースィーヴル定義に一致させます。

IBM MQ は、解決されたチャネル名を PARIS キュー・マネージャーに戻します。 PARIS では、INVENTORY.+QMNAME+ というチャネルの CLUSSDR チャネル定義が、INVENTORY.LONDON に対して内部的に生成された CLUSSDR 定義によって置き換えられます。 この定義には解決済みのチャネル名が含まれていますが、その他はユーザーが作成した +QMNAME+ 定義と同じです。 クラスター・リポジトリーも、新たに解決されたチャネル名を伴うチャネル定義によって、最新の状態にされます。

注 :

1. +QMNAME+ 名で作成されたチャネルは即時に非アクティブになります。 それを使用してデータを伝送することは決してありません。
2. チャネル出口では、ある呼び出しと次の呼び出しの間にチャネル名が変化した場合、それを確認することができます。

これで、キュー・マネージャー PARIS は、キュー・マネージャー LONDON のリポジトリーを参照することにより、INVENTQ キューがキュー・マネージャー NEWYORK にホストされていることを認識できるようになります。 システムによって、ホストされているアプリケーションが、メッセージを INVENTQ, PARIS に自動的に設定しようとすると、クラスター送信側チャネルは、クラスター受信側チャネル INVENTORY.NEWYORK に接続するためのクラスター送信側チャネルを自動的に定義します。 このとき、そのアプリケーションのあるキュー・マネージャーの名前を指定して、応答先キューを定義すると、そのアプリケーションはメッセージに対する応答を受信することができます。

関連タスク

クラスターへのキュー・マネージャーの追加: 別個の伝送キュー

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

関連資料

DEFINE CHANNEL

キューをホストするキュー・マネージャーの追加

別の INVENTQ キューをホストするために、別のキュー・マネージャーをクラスターに追加します。 各キュー・マネージャー上のキューに要求が交互に送信されます。 既存の INVENTQ ホストを変更する必要はありません。

始める前に

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- 313 ページの『クラスターにキュー・マネージャーを追加する』 で説明されているように、INVENTORY クラスターがセットアップされています。 このクラスターには、3 つのキュー・マネージャーが格納されています。このうち、LONDON と NEWYORK の 2 つは完全リポジトリーを保有し、PARIS は部分リポジトリーを保有しています。 在庫管理アプリケーションは、ニューヨークのシステムで動作します。このシステムは、キュー・マネージャー NEWYORK に接続されています。 このアプリケーションは、INVENTQ キューのメッセージが到着すると実行されます。
- トロントでは、新しい店舗の準備が進められています。 処理能力を増強するため、ニューヨークのシステムと同様、トロントのシステムでも在庫管理アプリケーションを実行する意向があります。
- ネットワークは、4 つのシステム間すべてに接続されています。
- ネットワーク・プロトコルは、TCP です。

注 : キュー・マネージャー TORONTO には、部分リポジトリーしか入っていません。 完全リポジトリー・キュー・マネージャーをクラスターに追加する場合は、325 ページの『完全リポジトリーの別のキュー・マネージャーへの移動』を参照してください。

このタスクについて

以下の手順に従って、キューをホストするキュー・マネージャーを追加します。

手順

1. TORONTO が最初に参照する完全リポジトリーを決める。

   クラスター内の各キュー・マネージャーは、いずれかの完全リポジトリーを参照しなければなりません。 キュー・マネージャーは、クラスターに関する情報を完全リポジトリーから収集し、自分用の部分リポジトリーを作成します。 どのリポジトリーが選択されるかは、あまり重要ではありません。 この例では、NEWYORK を選択します。 新しいキュー・マネージャーをクラスターに追加すると、このキュー・マネージャーは両方のリポジトリーと通信します。

2. CLUSRCVR チャネルを定義する。

   クラスター内の各キュー・マネージャーにクラスター受信側チャネルを定義する必要があります。このチャネルにより、キュー・マネージャーはメッセージを受信することができます。 TORONTO では、CLUSRCVR チャネルを次のように定義します。

   ```
   DEFINE CHANNEL(INVENTORY.TORONTO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(TORONTO.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for TORONTO')
   ```

   TORONTO キュー・マネージャーは、クラスター受信側チャネルを使用して INVENTORY クラスター内の他のキュー・マネージャーからメッセージを受信できる状態であることを通知します。

3. キュー・マネージャー TORONTO に CLUSSDR チャネルを定義する。

   クラスター内の各キュー・マネージャーは、その最初の完全リポジトリーへのメッセージの送信に使用できるクラスター送信側チャネルを 1 つ定義する必要があります。 この例では、NEWYORK を選択します。 TORONTO には次の定義が必要です。

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from TORONTO to repository at NEWYORK')
   ```

4. オプション: 以前にクラスターから除去したキュー・マネージャーを同じクラスターに追加する場合、今回クラスター・メンバーとして表示されることを確認します。 表示されない場合は、以下の追加の手順を実行します。

   a) 追加するキュー・マネージャーで **REFRESH CLUSTER** コマンドを発行します。

   このステップを行うと、クラスター・チャネルが停止され、クラスター内の他のキュー・マネージャー内で確実に最新である新しい一連のシーケンス番号が、ローカル・クラスター・キャッシュに設定されます。

   ```
   REFRESH CLUSTER(INVENTORY) REPOS(YES)
   ```

   注 : 大規模クラスターでは、処理中のクラスターに **REFRESH CLUSTER** コマンドを使用すると、破壊的な影響を及ぼす恐れがあります。その後、クラスター・オブジェクトが 27 日間隔で対象のキュー・マネージャーすべてに状況の更新を自動的に送信する際にも同様のことが起こり得ます。 大規模クラスターでのリフレッシュはクラスターのパフォーマンスと可用性に影響を与える可能性があるを参照してください。

   b) CLUSSDR チャネルを再始動します。

   (例えば、START CHANNEL コマンドを使用します)。

   c) CLUSRCVR チャネルを再始動します。

5. 在庫管理アプリケーションを調べてメッセージの類縁性を調べる。

これを調べる前に、在庫管理アプリケーションがメッセージの処理順序に依存しないことを確認して、トロントのシステムにアプリケーションをインストールしてください。

6. クラスター・キュー INVENTQ を定義する。

   INVENTQ キューは、キュー・マネージャー NEWYORK に既にホストされていますが、TORONTO によってもホストされます。 キュー・マネージャー TORONTO に次のように定義します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

タスクの結果

322 ページの図 42 は、このタスクによってセットアップされる INVENTORY クラスターを示しています。

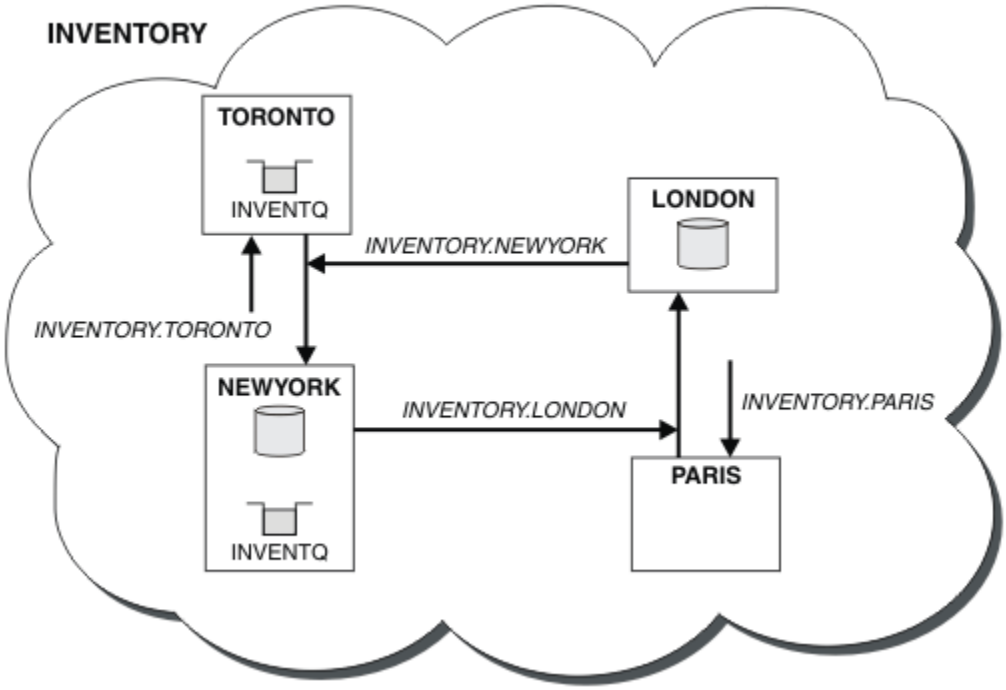


図 *42. 4* つのキュー・マネージャーが格納されている *INVENTORY* クラスター

現在、INVENTQ キューと在庫管理アプリケーションは、このクラスター内の 2 つのキュー・マネージャーによってホストされています。 これにより、これらの可用性が増大してメッセージのスループットが向上し、ワークロードが 2 つのキュー・マネージャーに分散されるようになります。 TORONTO または NEWYORK によって INVENTQ に書き込まれたメッセージは、可能なときは必ずローカル・キュー・マネージャーのインスタンスによって処理されます。 LONDON または PARIS によって書き込まれたメッセージは、TORONTO または NEWYORK に交互に経路指定されるため、ワークロードのバランスを取ることができます。

クラスターをこのように変更するときに、NEWYORK、LONDON、および PARIS の各キュー・マネージャーの定義を変更する必要はありませんでした。 これらのキュー・マネージャーの完全リポジトリーは、TORONTO の INVENTQ にメッセージを送信できるようにするために必要な情報によって、自動的に更新されます。 在庫管理アプリケーションは、NEWYORK または TORONTO のうちの 1 つのキュー・マネージャーが使用不可の状態になり、それ自身に十分な処理能力がある場合、動作を継続できます。 在庫管理アプリケーションが両方の場所でホストされている場合、正しく動作するはずです。

この作業結果から分かるように、複数のキュー・マネージャーで同じアプリケーションを実行することが可能です。 ワークロードを均等に分散するためにクラスタリングが可能です。

アプリケーションは両方の場所でレコードを処理できない可能性があります。 例えば、カスタマー・アカウント照会を追加し、LONDON および NEWYORK で実行されているアプリケーションを更新することを決めたと仮定します。 アカウント・レコードは 1 カ所でのみ保有できます。 データ区分化の手法を用いて、要

求の分散方法を制御することを選択できます。 レコードの配布を分割できます。 レコードの半分を整理して、例えばアカウント番号 00000 から 49999 までを LONDON で保有するようにすることができます。 残りの半分である 50000 から 99999 までは、NEWYORK で保有します。 その後、すべてのメッセージのアカウント・フィールドを調べ、該当するキュー・マネージャーにメッセージを送信するクラスター・ワークロード出口プログラムを作成することができます。

次のタスク

これですべての定義が完了しました。IBM MQ for z/OS でまだチャネル・イニシエーターを開始していない場合は開始します。 すべてのプラットフォームで、キュー・マネージャー TORONTO でリスナー・プログラムを始動します。 リスナー・プログラムは、着信ネットワーク要求を待機し、必要な場合にクラスター受信側チャネルを始動します。

z/OS 既存のクラスターへのキュー共有グループの追加

z/OS のキュー共有グループを既存のクラスターに追加します。

始める前に

注 :

1. クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。
2. キュー共有グループは、IBM MQ for z/OS でのみサポートされています。 この作業は、他のプラットフォームには該当しません。

シナリオ

- 301 ページの『新規クラスターのセットアップ』で説明されているように、 INVENTORY クラスターがセットアップされている。 これは、LONDON および NEWYORK という 2 つのキュー・マネージャーで構成されています。
- このクラスターにキュー共有グループを追加することにします。 グループ QSGP は、3 つのキュー・マネージャー P1、P2、および P3 で構成されます。 これらは、P1 によって定義される INVENTQ キューのインスタンスを共有します。

このタスクについて

以下の手順に従って、共有キューをホストする新しいキュー・マネージャーを追加します。

手順

1. キュー・マネージャーが最初に参照する完全リポジトリーを決める

   クラスター内の各キュー・マネージャーは、いずれかの完全リポジトリーを参照しなければなりません。 キュー・マネージャーは、クラスターに関する情報を完全リポジトリーから収集し、自分用の部分リポジトリーを作成します。 どの完全リポジトリーが選択されるかは、あまり重要ではありません。 この例では、NEWYORK を選択します。 キュー共有グループをクラスターに追加すると、それは両方の完全リポジトリーと通信します。
2. CLUSRCVR チャネルを定義する。

   クラスター内の各キュー・マネージャーにクラスター受信側チャネルを定義する必要があります。このチャネルにより、キュー・マネージャーはメッセージを受信することができます。 P1、P2、および P3 で、以下のように定義します。

```
DEFINE CHANNEL(INVENTORY.Pn) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(Pn.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for sharing queue manager')
```

クラスター受信側チャネルは、各キュー・マネージャーがクラスター INVENTORY 内の他のキュー・マネージャーからメッセージを受信できる状態であることを通知します。

3. キュー共有グループの CLUSSDR チャネルを定義する

   クラスター内の各メンバーは、そのプログラムの最初の完全リポジトリーにメッセージを送信できるクラスター送信側チャネルを 1 つ定義する必要があります。 この例では、NEWYORK を選択しました。 キュー共有グループ内のキュー・マネージャーの 1 つには、以下のようなグループ定義が必要です。 この定義により、すべてのキュー・マネージャーが必ずクラスター送信側チャネル定義を持つようになります。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) QSGDISP(GROUP)
DESCR('Cluster-sender channel to repository at NEWYORK')
```

4. 共有キューを定義する

   次のように P1 にキュー INVENTQ を定義します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) QSGDISP(SHARED) CFSTRUCT(STRUCTURE)
```

   新規キュー・マネージャーで、チャネル・イニシエーターおよびリスナー・プログラムを始動します。 リスナー・プログラムは、着信ネットワーク要求がないかどうか listen し、必要な場合にクラスター受信側チャネルを起動します。

タスクの結果

324 ページの図 43 は、このタスクによってセットアップされるクラスターを示します。

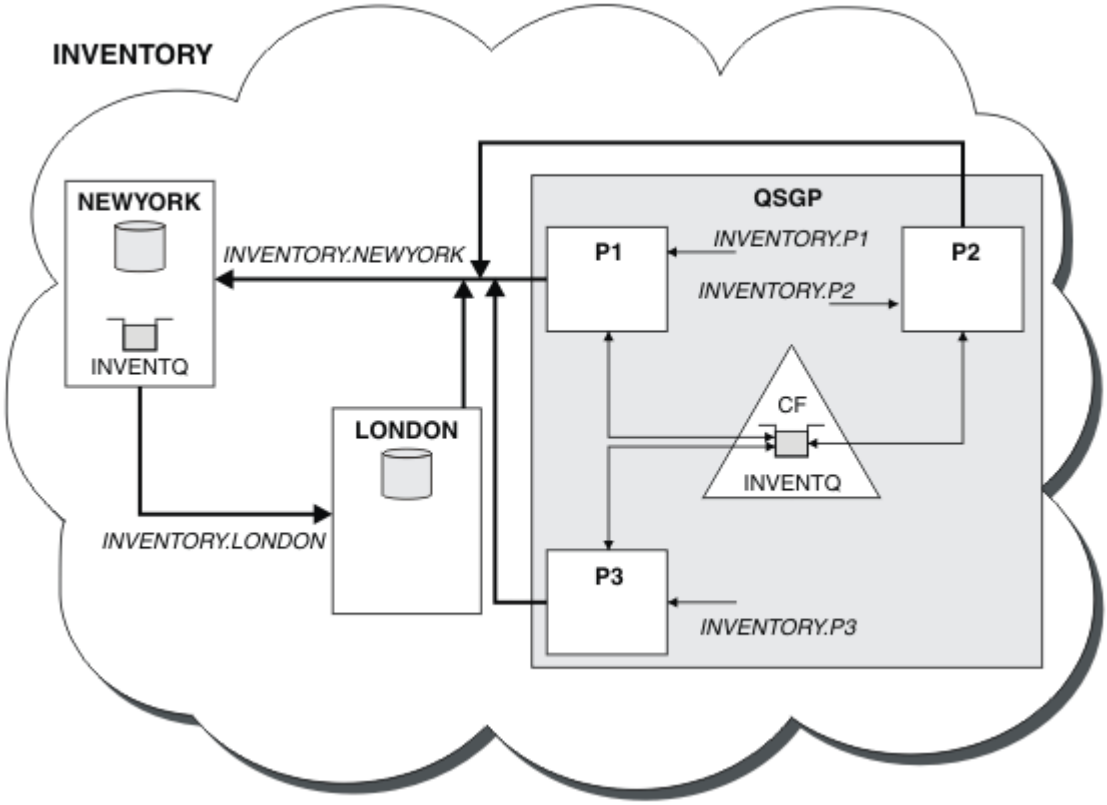


図 *43.* クラスターとキュー共有グループ

これで、LONDON によって INVENTQ キューに入れられるメッセージは、キューをホストすると公示されている 4 つのキュー・マネージャーの間で経路指定されます。

次のタスク

キュー共有グループのメンバーがクラスター・キューをホストする利点は、グループの他のメンバーが要求に対して応答できるということです。 この例では、P1 が、共有キューでメッセージを受信した後で利用不可になったとします。 キュー共有グループの他のメンバーが代わりに応答できます。

完全リポジトリーの別のキュー・マネージャーへの移動

完全リポジトリーを 1 つのキュー・マネージャーから別のキュー・マネージャーへ移動し、2 番目のリポジトリーで保有された情報から新しいリポジトリーを作成します。

始める前に

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- 313 ページの『クラスターにキュー・マネージャーを追加する』 で説明されているように、INVENTORY クラスターがセットアップされています。
- 業務上の理由により、キュー・マネージャー LONDON から完全リポジトリーを削除して、これをキュー・マネージャー PARIS の完全リポジトリーで置きかえる必要が生じました。 キュー・マネージャー NEWYORK は、引き続き完全リポジトリーを保有します。

このタスクについて

以下の手順に従って、完全リポジトリーを別のキュー・マネージャーへ移動します。

手順

1. PARIS を完全リポジトリー・キュー・マネージャーに変更する

   PARIS で、以下のコマンドを発行します。

   ```
   ALTER QMGR REPOS(INVENTORY)
   ```

2. PARIS で CLUSSDR チャネルを追加する

   PARIS には、現在宛先が LONDON になっているクラスター送信側チャネルがあります。 LONDON はクラスターの完全リポジトリーを保有しなくなりました。 PARIS には、NEWYORK を指す新しいクラスター送信側チャネルが必要です。このチャネルでは、もう一方のフル・リポジトリーが保持されています。

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from PARIS to repository at NEWYORK')
   ```

3. 宛先が PARIS の CLUSSDR チャネルを NEWYORK に定義する

   現在、NEWYORK には、宛先が LONDON になっているクラスター送信側チャネルがあります。 他の完全リポジトリーが PARIS に移動したので、宛先が PARIS の新しいクラスター送信側チャネルを NEWYORK に追加する必要があります。

   ```
   DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from NEWYORK to repository at PARIS')
   ```

   PARIS にクラスター送信側チャネルを追加すると、PARIS は NEWYORK からクラスターに関する情報を取得します。 そして、NEWYORK からの情報を利用して、固有の完全リポジトリーを作成します。

4. キュー・マネージャー PARIS に現在完全リポジトリーが存在することを確認する

キュー・マネージャー PARIS が、キュー・マネージャー NEWYORK の完全リポジトリーから、固有の完全リポジトリーを作成したことを確認します。 次のコマンドを発行します。

```
DIS QCLUSTER(*) CLUSTER (INVENTORY)
DIS CLUSQMGR(*) CLUSTER (INVENTORY)
```

これらのコマンドによって、このクラスター内の NEWYORK にあるものと同じリソースの詳細が表示されることを確認します。

注: キュー・マネージャー NEWYORK が使用できない場合、この情報の作成を完了することはできません。 この作業が完了するまで、次の手順に進まないでください。

5. LONDON のキュー・マネージャー定義を変更する

   最後に LONDON のキュー・マネージャーを変更して、このキュー・マネージャーがクラスターの完全リポジトリーを保有しないようにします。 LONDON で、次のコマンドを発行します。

   ```
   ALTER QMGR REPOS(' ')
   ```

   キュー・マネージャーは、クラスター情報を受け取ることがなくなりました。 30 日後、その完全リポジトリーに格納されている情報は、期限が切れます。 このため、キュー・マネージャー LONDON は、固有の部分リポジトリーを作成します。
6. 未解決の定義を削除または変更する

   クラスターの新しい配置が予期されるとおりに機能していることを確認したら、正確ではなくなった、手動で定義された CLUSSDR 定義を削除または変更します。

   - キュー・マネージャー PARIS で LONDON へのクラスター送信側チャネルを停止して削除する必要があります。次に、開始チャネル・コマンドを発行して、クラスターが自動チャネルを再度使用することができるようにします。

     ```
     STOP CHANNEL(INVENTORY.LONDON)
     DELETE CHANNEL(INVENTORY.LONDON)
     START CHANNEL(INVENTORY.LONDON)
     ```

   - キュー・マネージャー NEWYORK で LONDON へのクラスター送信側チャネルを停止して削除する必要があります。次に、開始チャネル・コマンドを発行して、クラスターが自動チャネルを再度使用することができるようにします。

     ```
     STOP CHANNEL(INVENTORY.LONDON)
     DELETE CHANNEL(INVENTORY.LONDON)
     START CHANNEL(INVENTORY.LONDON)
     ```

   - クラスター内のすべてのキュー・マネージャーにおいて、宛先が LONDON の、手動で定義された他のすべてのクラスター送信側チャネルを、宛先が NEWYORK または PARIS のチャネルに置き換えます。チャネルを削除した後には、必ず **start channel** コマンドを発行して、クラスターで自動チャネルを再び使用できるようにしてください。 この小規模な例では、他のチャネルはありません。 他に忘れていたチャネルがないか確認するには、各キュー・マネージャーから、TYPE(CLUSSDR) を指定して DISPLAY CHANNEL コマンドを発行します。 以下に例を示します。

     ```
     DISPLAY CHANNEL(*) TYPE(CLUSSDR)
     ```

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

   LONDON では、完全リポジトリーではなくなった後にそのような要求を受け取った場合、キュー・マネージャーのエラー・ログにエラー・メッセージを書き込みます。 以下の例は、LONDON で表示される可能性のあるエラー・メッセージを示します。

   - AMQ9428: Unexpected publication of a cluster queue object received
   - AMQ9432: Query received by a non-repository queue manager

キュー・マネージャー LONDON は完全リポジトリーではないので、情報の要求に応答しません。 LONDON から情報を要求しているキュー・マネージャーは、手動で定義された CLUSSDR 定義が PARIS を宛先とするよう修正されるまで NEWYORK を使用してクラスター情報を得る必要があります。 この状態を、有効な構成として長い期間容認するべきではありません。

タスクの結果

327 ページの図 44 は、このタスクによってセットアップされるクラスターを示します。

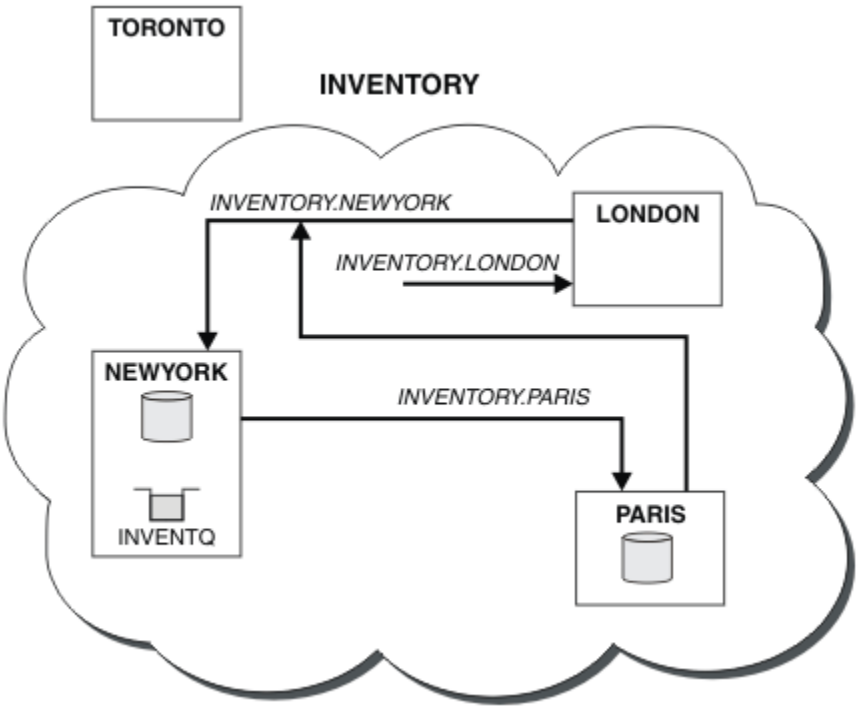


図 *44.* 完全リポジトリーが *PARIS* に移動した *INVENTORY* クラスター

既存のネットワークのクラスターへの変換

既存の分散キューイング・ネットワークをクラスターに変換し、キュー・マネージャーを追加して容量を増やします。

始める前に

325 ページの『完全リポジトリーの別のキュー・マネージャーへの移動』 から 301 ページの『新規クラスターのセットアップ』 では、新規クラスターを作成して拡張しました。 次の 2 つの作業では、キュー・マネージャーの既存のネットワークをクラスターに変換するためのさまざまな手法について調べます。

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- IBM MQ ネットワークは既に構築されており、これによりチェーン店の支店が全国規模で接続されています。 ネットワークの構造はハブ・スポーク構造であり、すべてのキュー・マネージャーは中心にある 1 つのキュー・マネージャーに接続されています。 中心にあるキュー・マネージャーは、在庫管理アプリケーションが動作するシステム上にあります。 アプリケーションは、INVENTQ キューにメッセージが到着すると実行されます。このキューの各キュー・マネージャーには、リモート・キュー定義があります。

  このネットワークを 328 ページの図 45 に示します。

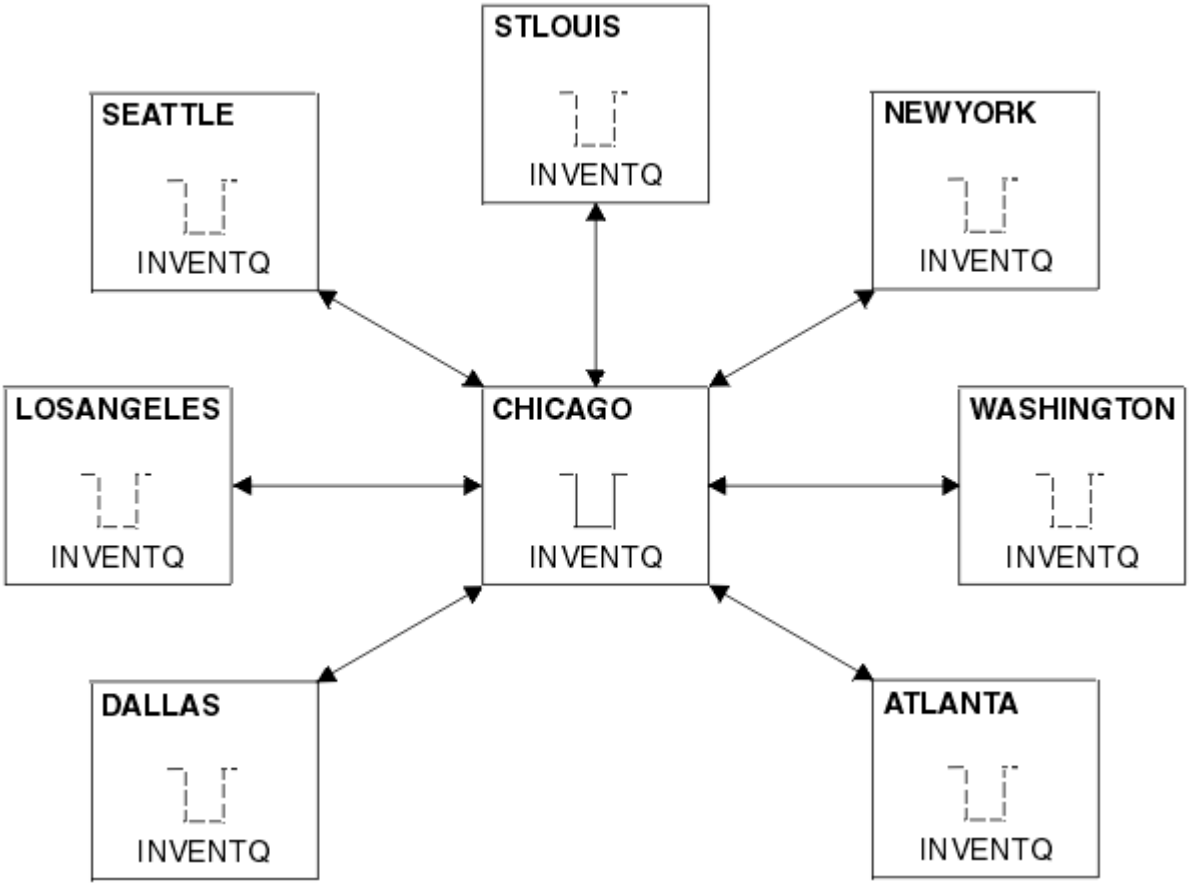


図 *45.* ハブ・スポーク・ネットワーク

- 管理を容易にするため、このネットワークをクラスターに変換して、中心の場所に別のキュー・マネージャーを作成して、ワークロードを共有する予定です。

  クラスター名は CHNSTORE とします。

  注: クラスター名 CHNSTORE が選択され、最大長 20 文字を超えない *cluster_name.queue_manager_name* 形式の名前 (例えば、CHNSTORE.WASHINGTON) を使用してクラスター受信側チャネル名を作成できるようになりました。
- 中心にある 2 つのキュー・マネージャーは、完全リポジトリーをホストし、在庫管理アプリケーションにアクセス可能です。
- 在庫管理アプリケーションは、中心にある 2 つのキュー・マネージャーのいずれかによってホストされた INVENTQ キューのメッセージが到着すると実行されます。
- 在庫管理アプリケーションは、同時に実行される唯一のアプリケーションとなり、複数のキュー・マネージャーによってアクセスが可能です。 他のすべてのアプリケーションは、以前と同様の動作を継続します。
- すべての支店は、中心にある 2 つのキュー・マネージャーにネットワークで接続されます。
- ネットワーク・プロトコルは、TCP です。

このタスクについて

以下の手順に従って、既存のネットワークをクラスターに変換します。

手順

1. 在庫管理アプリケーションを調べてメッセージの類縁性を調べる。

   先に進む前に、アプリケーションがメッセージの類縁性を処理できることを確認してください。 メッセージの類縁性とは、2 つのアプリケーション間で交換される会話型メッセージ間の関係で、それらのメ

ッセージは特定のキュー・マネージャーによって、または特定の順序で処理されなければなりません。メッセージの類縁性について詳しくは、407 ページの『メッセージの類縁性の処理』を参照してください。

2. 中心にある 2 つのキュー・マネージャーを完全リポジトリー・キュー・マネージャーに変更する。

   2 つのキュー・マネージャー CHICAGO および CHICAGO2 が、このネットワークのハブにあります。CHAINSTORE クラスターに対応する活動をこれら 2 つのキュー・マネージャーに集約することに決めました。 在庫管理アプリケーションおよび INVENTQ キューの定義以外に、クラスターの 2 つの完全リポジトリーをこれらのキュー・マネージャーにホストさせることが必要です。 2 つのキュー・マネージャーで、それぞれ次のコマンドを出します。

   ```
   ALTER QMGR REPOS(CHNSTORE)
   ```

3. 各キュー・マネージャーで CLUSRCVR チャネルを定義する。

   クラスター内の各キュー・マネージャーで、クラスター受信側チャネルとクラスター送信側チャネルを定義してください。 どちらのチャネルを最初に定義しても構いません。

   CLUSRCVR 定義を作成して、各キュー・マネージャー、そのネットワーク・アドレス、および他の情報をクラスターに通知します。 例えば、キュー・マネージャー ATLANTA では、次のように定義します。

   ```
   DEFINE CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(ATLANTA.CHSTORE.COM) CLUSTER(CHNSTORE)
   DESCR('Cluster-receiver channel')
   ```

4. 各キュー・マネージャーで CLUSSDR チャネルを定義する。

   各キュー・マネージャーで CLUSSDR を定義して、そのキュー・マネージャーをいずれかの完全リポジトリー・キュー・マネージャーにリンクします。 例えば、ATLANTA を CHICAGO2 にリンクします。

   ```
   DEFINE CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(CHICAGO2.CHSTORE.COM) CLUSTER(CHNSTORE)
   DESCR('Cluster-sender channel to repository queue manager')
   ```

5. 在庫管理アプリケーションを CHICAGO2 にインストールする。

   キュー・マネージャー CHICAGO には、既に在庫管理アプリケーションがあります。 したがって、キュー・マネージャー CHICAGO2 にこのアプリケーションをコピーする必要があります。

6. 中心にあるキュー・マネージャーに INVENTQ キューを定義する。

   CHICAGO で、INVENTQ キューのローカル・キュー定義を変更して、そのキューをそのクラスターで使用できるようにします。 次のコマンドを実行します。

   ```
   ALTER QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
   ```

   CHICAGO2 で、同じキューの定義を作成します。

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
   ```

   z/OS では、 **CSQUTIL** の COMMAND 機能の MAKEDEF オプションを使用して、 INVENTQ on CHICAGO の CHICAGO2 上に正確なコピーを作成できます。

   これらを定義すると、CHICAGO と CHICAGO2 の完全リポジトリーにメッセージが送信され、それらの中の情報が更新されます。 キュー・マネージャーは、メッセージを INVENTQ に書き込む際に、メッセージの宛先を選択できることを完全リポジトリーから判断します。

7. クラスターの変更が伝搬されたことを確認する。

   前の手順で作成した定義がクラスター全体に伝搬されたことを確認します。 完全リポジトリー・キュー・マネージャーで以下のコマンドを発行します。

   ```
   DIS QCLUSTER(INVENTQ)
   ```

相互接続された新しいクラスターの追加

既存のクラスターと一部のキュー・マネージャーを共有する新しいクラスターを追加します。

始める前に

注:

1. クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。
2. この作業を開始する前に、キュー名が衝突していないかどうかを確認し、衝突による影響を把握します。 作業を進める前に、キューの名前を変更するか、キューの別名を設定する必要があることがあります。

シナリオ

- 327 ページの『既存のネットワークのクラスターへの変換』で説明されているように、 IBM MQ クラスターがセットアップされました。
- MAILORDER と呼ばれる新しいクラスターが実行される予定です。 このクラスターには、CHNSTORE クラスターにあるキュー・マネージャーのうちの 4 つ (CHICAGO、CHICAGO2、SEATTLE、ATLANTA) が格納され、さらに 2 つのキュー・マネージャー (HARTFORD および OMAHA) が追加されます。 MAILORDER アプリケーションは、オマハにあるシステムで実行され、キュー・マネージャー OMAHA に接続します。 このアプリケーションは、クラスター内にある他のキュー・マネージャーのうち、MORDERQ キューにメッセージを書き込むキュー・マネージャーによって実行されます。
- MAILORDER クラスターの完全リポジトリーは、2 つのキュー・マネージャー CHICAGO および CHICAGO2 によって保守されます。
- ネットワーク・プロトコルは、TCP です。

このタスクについて

以下の手順に従って、新規の内部接続されたクラスターを追加します。

手順

1. クラスター名の名前リストを作成する

   CHICAGO と CHICAGO2 の完全リポジトリー・キュー・マネージャーは、CHNSTORE と MAILORDER の両方のクラスターの完全リポジトリーを保有する予定です。 まず、クラスターの名前を含む名前リストを作成します。 CHICAGO と CHICAGO2 の名前リストを次のように定義します。

   ```
   DEFINE NAMELIST(CHAINMAIL)
   DESCR('List of cluster names')
   NAMES(CHNSTORE, MAILORDER)
   ```

2. 2 つのキュー・マネージャー定義を変更する

   この時点で、CHICAGO および CHICAGO2 の 2 つのキュー・マネージャー定義を変更します。 現在、これらの定義では、これらのキュー・マネージャーが CHNSTORE クラスターの完全リポジトリーを保有することになっています。 これらのキュー・マネージャーが、CHAINMAIL 名前リストに登録されているすべてのクラスターの完全リポジトリーを保有するように、この定義を変更してください。 次のように、CHICAGO および CHICAGO2 キュー・マネージャー定義を変更します。

   ```
   ALTER QMGR REPOS(' ') REPOSNL(CHAINMAIL)
   ```

3. CHICAGO および CHICAGO2 で CLUSRCVR チャネルを変更する

   CHICAGO および CHICAGO2 の CLUSRCVR チャネル定義では、チャネルは、CHNSTORE クラスターで使用可能であることになっています。 クラスター受信側定義を変更して、CHAINMAIL 名前リストに含

まれるすべてのクラスターに対してチャネルが使用可能であることを示す必要があります。 次のようにして、CHICAGO でクラスター受信側定義を変更します。

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

CHICAGO2 で、次のコマンドを入力します。

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

4. CHICAGO および CHICAGO2 で CLUSSDR チャネルを変更する。

   2 つの CLUSSDR チャネル定義を変更して、名前リストを追加します。 CHICAGO で、次のコマンドを入力します。

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

   CHICAGO2 で、次のコマンドを入力します。

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

5. SEATTLE および ATLANTA で名前リストを作成する。

   SEATTLE および ATLANTA は複数のクラスターのメンバーになるので、複数のクラスターの名前が入っている名前リストを作成する必要があります。 SEATTLE および ATLANTA で、次のように名前リストを定義します。

   ```
   DEFINE NAMELIST(CHAINMAIL)
   DESCR('List of cluster names')
   NAMES(CHNSTORE, MAILORDER)
   ```

6. SEATTLE および ATLANTA で CLUSRCVR チャネルを変更する

   SEATTLE および ATLANTA の CLUSRCVR チャネル定義では、チャネルは、CHNSTORE クラスターで使用可能であることになっています。 クラスター受信側チャネル定義を変更して CHAINMAIL 名前リストに含まれるすべてのクラスターに対してチャネルが使用可能であることを示す必要があります。 SEATTLE で、次のコマンドを入力します。

   ```
   ALTER CHANNEL(CHNSTORE.SEATTLE) CHLTYPE(CLUSRCVR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

   ATLANTA で、次のコマンドを入力します。

   ```
   ALTER CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

7. SEATTLE および ATLANTA で CLUSSDR チャネルを変更する。

   2 つの CLUSSDR チャネル定義を変更して、名前リストを追加します。 SEATTLE で、次のコマンドを入力します。

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

   ATLANTA で、次のコマンドを入力します。

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

8. CLUSRCVR および CLUSSDR チャネルを HARTFORD および OMAHA で定義する。

2 つの新しいキュー・マネージャー HARTFORD および OMAHA で、クラスター受信側チャネルとクラスター送信側チャネルを定義してください。 どちらを先に定義しても構いません。 HARTFORD で、次のコマンドを入力します。

```
DEFINE CHANNEL(MAILORDER.HARTFORD) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(HARTFORD.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for HARTFORD')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-sender channel from HARTFORD to repository at CHICAGO')
```

OMAHA で、次のコマンドを入力します。

```
DEFINE CHANNEL(MAILORDER.OMAHA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(OMAHA.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for OMAHA')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-sender channel from OMAHA to repository at CHICAGO')
```

9. OMAHA で MORDERQ キューを定義する。

   この作業の最終段階では、MORDERQ キューをキュー・マネージャー OMAHA で定義します。 OMAHA で、次のコマンドを入力します。

   ```
   DEFINE QLOCAL(MORDERQ) CLUSTER(MAILORDER)
   ```

10. クラスターの変更が伝搬されたことを確認する。

    前の手順で作成した定義がクラスター全体に伝搬されたことを確認します。 完全リポジトリー・キュー・マネージャーで、次のコマンドを発行します。

    ```
    DIS QCLUSTER (MORDERQ)
    DIS CLUSQMGR
    ```

11.

タスクの結果

上述の作業によって設定されるクラスターは、333 ページの図 46 で示されています。

現在の設定では、クラスターが 2 つ重なり合っています。 2 つのクラスターの完全リポジトリーは、CHICAGO および CHICAGO2 に保有されています。 OMAHA で実行される MAILORDER アプリケーションは、CHICAGO で実行される在庫管理アプリケーションとは独立しています。 しかし、一部のキュー・マネージャーは、CHNSTORE クラスターと MAILORDER クラスターの両方に属しているので、それらのキュー・マネージャーは、どちらのアプリケーションにもメッセージを送信できます。 この作業を実行して 2 つのクラスターを重ね合わせるときは、キュー名が衝突しないよう注意してください。

CHNSTORE クラスターの NEWYORK と MAILORDER クラスターの OMAHA に、ACCOUNTQ と呼ばれるキューがあると仮定します。 クラスターをオーバーラップさせた後で、SEATTLE にあるアプリケーションが ACCOUNTQ キューにメッセージを書き込んだ場合、メッセージは ACCOUNTQ のどちらのインスタンスにも送信される可能性があります。

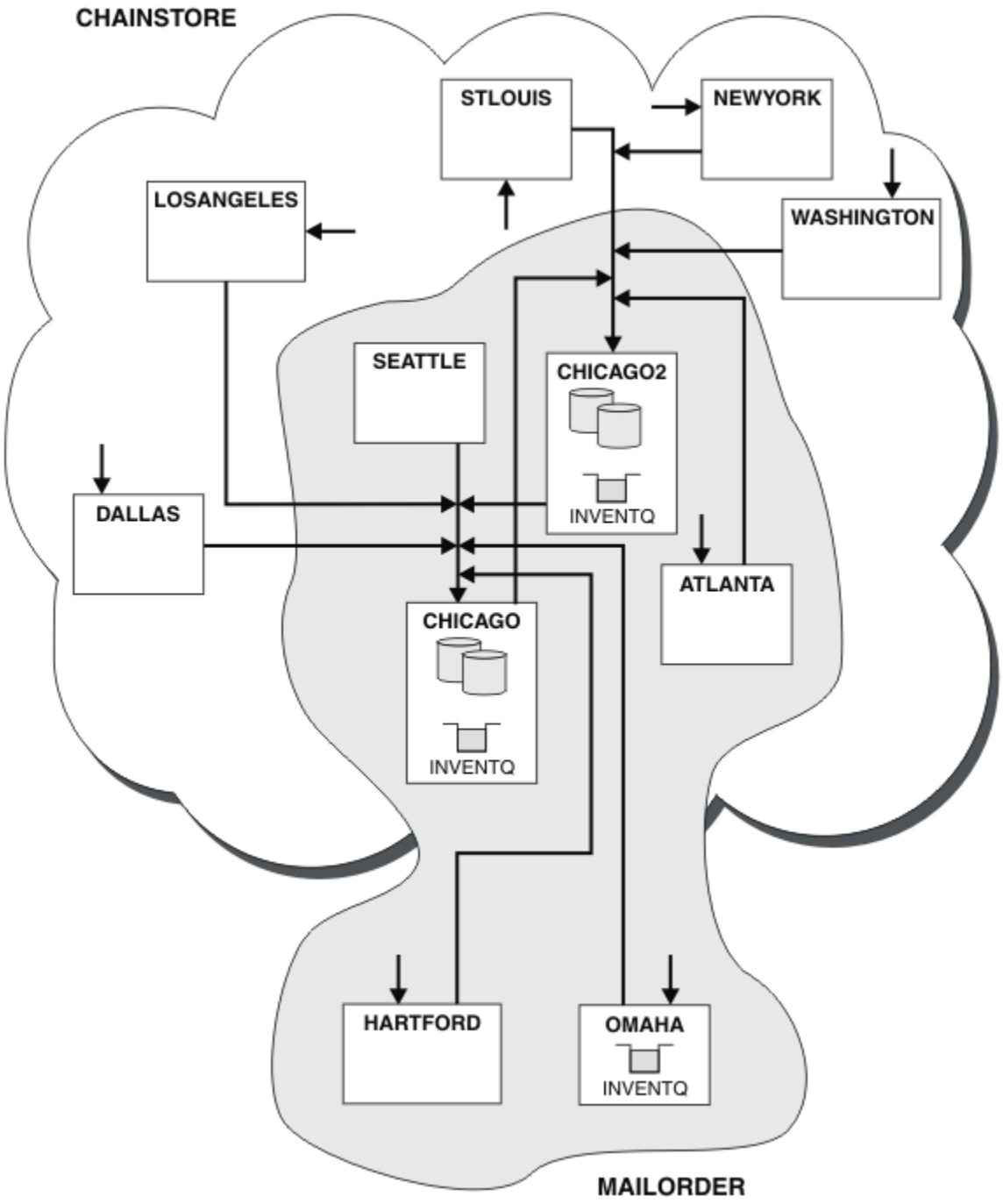


図 *46.* 相互接続されたクラスター

次のタスク

MAILORDER クラスターと CHNSTORE クラスターをマージして、CHNSTORE と呼ばれる 1 つの大型クラスターを形成することにしたと想定します。

CHNSTORE クラスターと MAILORDER クラスター、つまり、CHICAGO と CHICAGO2 が完全リポジトリーを保有するようなクラスターをマージするには、以下のようにします。

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

```
ALTER QMGR(CHICAGO) REPOSNL(' ') REPOS(CHNSTORE)
```

- MAILORDER クラスターの各キュー・マネージャーで、すべてのチャネル定義およびキュー定義を変更して、CLUSTER 属性の値を MAILORDER から CHNSTORE に変更します。 例えば、HARTFORD で、次のように入力します。

```
ALTER CHANNEL(MAILORDER.HARTFORD) CLUSTER(CHNSTORE)
```

  OMAHA で、次のコマンドを入力します。

```
ALTER QLOCAL(MORDERQ) CLUSTER(CHNSTORE)
```

- CHAINMAIL というクラスター名前リストを指定するすべての定義 (つまり CHICAGO、CHICAGO2、SEATTLE、ATLANTA での CLUSRCVR チャネル定義および CLUSSDR チャネル) を変更して、その代わりに CHNSTORE クラスターを指定します。

この例から、名前リストを使用する利点が理解できます。 CHICAGO および CHICAGO2 のキュー・マネージャー定義を変更する代わりに、CHAINMAIL 名前リストの値を変更できます。 同様に CLUSRCVR および CLUSSDR チャネル定義を CHICAGO、CHICAGO2、SEATTLE、ATLANTA で変更する代わりに、名前リストを変更することによって必要な結果を得ることができます。

関連タスク

クラスター・ネットワークの削除
ネットワークからクラスターを削除し、分散キューイング構成を復元します。

クラスター・ネットワークの削除

ネットワークからクラスターを削除し、分散キューイング構成を復元します。

始める前に

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- 327 ページの『既存のネットワークのクラスターへの変換』で説明されているように、 IBM MQ クラスターがセットアップされました。
- 今度は、このクラスターをシステムから除去します。 キュー・マネージャーのネットワークは、クラスターが設定される前と同じように引き続き機能します。

このタスクについて

以下の手順に従って、クラスター・ネットワークを除去します。

手順

1. CHNSTORE クラスターからクラスター・キューを削除します。

   CHICAGO と CHICAGO2 の両方で、キュー INVENTQ のローカル・キュー定義を変更して、クラスターからキューを削除します。 次のコマンドを実行します。

```
ALTER QLOCAL(INVENTQ) CLUSTER(' ')
```

   キューを変更すると、完全リポジトリー内の情報が更新され、クラスター全体に伝搬されます。 DEFBIND(NOTFIXED) によってキューが定義されている、MQOO_BIND_NOT_FIXED を使用するアク

ティブ・アプリケーション、および MQOO_BIND_AS_Q_DEF を使用するアプリケーションは、次に MQPUT 呼び出しまたは MQPUT1 呼び出しの試行に失敗します。 理由コード MQRC_UNKNOWN_OBJECT_NAME が戻されます。

ステップ 1 を最初に行う必要はありませんが、最初に行わない場合はステップ 4 の後で行ってください。

2. クラスター・キューにアクセスするすべてのアプリケーションを停止する

   クラスター・キューにアクセスするすべてのアプリケーションを停止してください。 そうしないと、ステップ 5 でクラスターをリフレッシュすると、一部のクラスター情報がローカル・キュー・マネージャー上に残ることがあります。 この情報は、すべてのアプリケーションが停止し、クラスター・チャネルが切断されたときに除去されます。

3. 完全リポジトリー・キュー・マネージャーからリポジトリー属性を削除する

   CHICAGO と CHICAGO2 の両方で、キュー・マネージャーの定義を変更してリポジトリー属性を削除します。 これを実行するには、次のコマンドを入力します。

   ```
   ALTER QMGR REPOS(' ')
   ```

   キュー・マネージャーは、完全リポジトリーを保有しなくなったことをクラスター内の他のキュー・マネージャーに通知します。 他のキュー・マネージャーがこの情報を受け取ると、完全リポジトリーが終了したことを示すメッセージが表示されます。 さらに、クラスター CHNSTORE に関して利用できるリポジトリーがなくなったことを示す 1 つまたは複数のメッセージが表示されます。

4. クラスター・チャネルを削除する

   CHICAGO でクラスター・チャネルを削除するために、次のコマンドを発行します。

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) CLUSTER(' ')
   ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR) CLUSTER(' ')
   ```

   注: 最初に CLUSSDR コマンド、次に CLUSRCVR コマンドの順序で発行することが重要です。 最初に CLUSRCVR コマンド、次に CLUSSDR コマンドの順序で発行しないでください。 これを行うと、STOPPED 状況の未確定チャネルが作成されます。 この場合は、START CHANNEL コマンドを発行して、停止したチャネルをリカバリーする必要があります (例えば、START CHANNEL(CHNSTORE.CHICAGO))。

   クラスター CHNSTORE に関するリポジトリーがないことを示すメッセージが表示されます。

   手順 1 で説明されているようにクラスター・キューを削除していない場合は、ここで削除してください。

5. クラスター・チャネルを停止する

   CHICAGO でクラスター・チャネルを停止するために、次のコマンドを発行します。

   ```
   STOP CHANNEL(CHNSTORE.CHICAGO2)
   STOP CHANNEL(CHNSTORE.CHICAGO)
   ```

6. クラスター内のキュー・マネージャーごとに手順 4 と 5 を繰り返す
7. クラスター・チャネルを停止し、クラスター・チャネルおよびクラスター・キューに関するすべての定義をそれぞれのキュー・マネージャーから除去します。
8. オプション: キュー・マネージャーによって保持されるクラスター情報のキャッシュの消去

   キュー・マネージャーがクラスターのメンバーでなくなっても、各キュー・マネージャーにはクラスターに関する情報のキャッシュ・コピーが保存されています。 このデータを削除する場合は、364 ページの『キュー・マネージャーのクラスター化以前の状態への復元』に説明されている作業を参照してください。

9. INVENTQ のリモート・キュー定義を置き換える

   ネットワークが引き続き機能するように、キュー・マネージャーごとに INVENTQ のリモート・キュー定義を置き換えてください。

10. クラスターを整理する

必要のなくなったキューやチャネルの定義をすべて削除してください。

関連タスク

相互接続された新しいクラスターの追加

既存のクラスターと一部のキュー・マネージャーを共有する新しいクラスターを追加します。

ゲートウェイ・キュー・マネージャーを使用した 2 つのオーバーラップするクラスターの作成

このタスクの手順に従って、ゲートウェイ・キュー・マネージャーを使用したオーバーラップするクラスターを構成します。 このクラスターを、以降で説明するあるメッセージへのメッセージをクラスター内の他のアプリケーションへのメッセージから分離する例の開始点として使用します。

このタスクについて

336 ページの図 47 に、クラスター・メッセージ・トラフィックの分離方法を説明するために使用するクラスター構成例を示します。 この例は、クラスター化: 複数のクラスター伝送キューの使用によるアプリケーションの分離で説明されています。

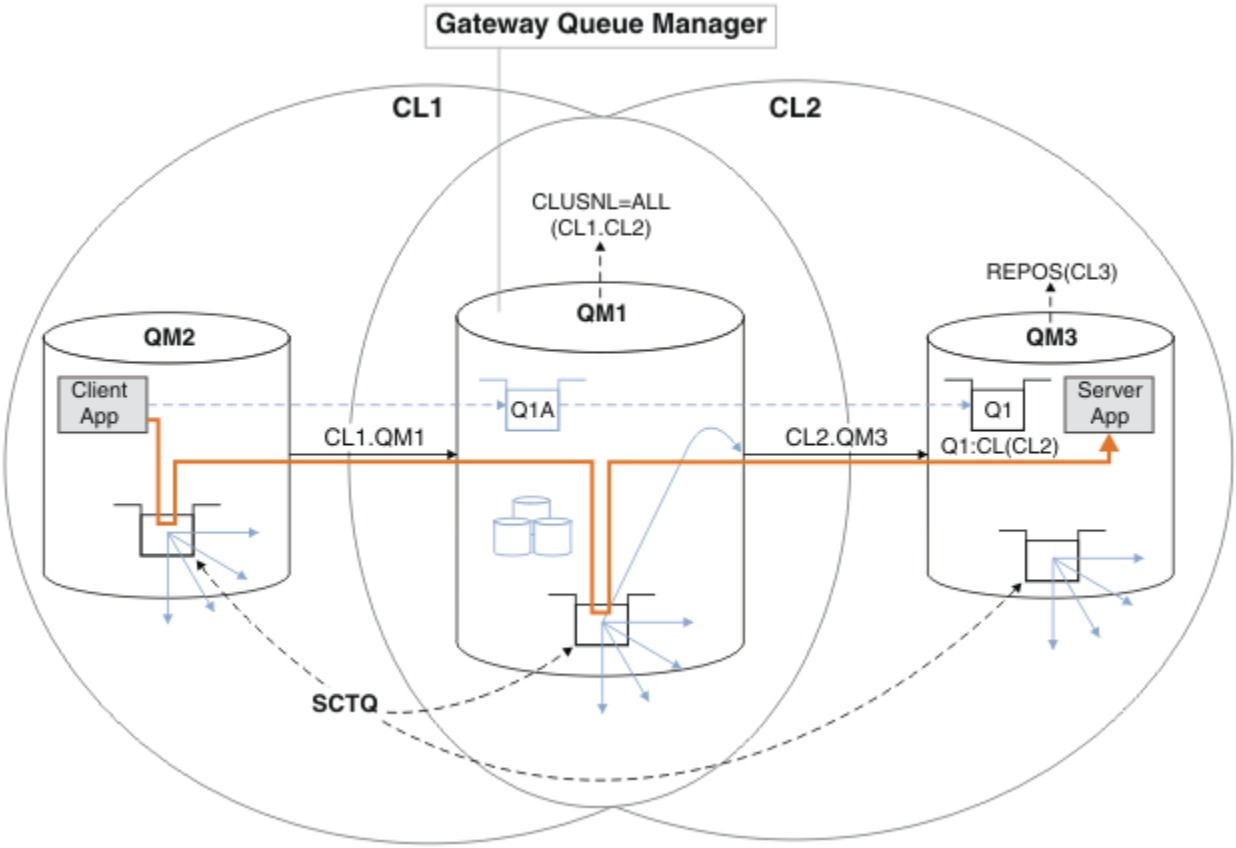


図 *47. ハブ・スポーク・アーキテクチャーにデプロイされた、IBM MQ クラスターを使用するクライアント/サーバー・アプリケーション*

例を構築する手順数を最小限に抑えるため、構成は現実的ではなく単純に保たれています。 この例は、2 つの別々の組織で作成された 2 つのクラスターの統合を表しています。 より現実的なシナリオについては、クラスター化: クラスター伝送キューの構成方法の計画を参照してください。

手順に従って、クラスターを構成します。 これらのクラスターが、以降で説明するクライアント・アプリケーションからサーバー・アプリケーションへのメッセージ・トラフィックを分離する例で使用されます。

この手順では、さらにいくつかのキュー・マネージャーを追加して、各クラスターが 2 つのリポジトリーを持つようにします。 ゲートウェイキューマネージャー QM1 は、パフォーマンス上の理由からリポジトリとしては使用されていません。

手順

1. ゲートウェイキューマネージャー QM1 、およびキューマネージャー QM2 、 QM3 、 QM4 、 QM5 を作成し、起動します。

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QM n
strmqm QmgrName
```

注 : QM4 および QM5 は、クラスターのバックアップ完全リポジトリーです。

2. 各キューマネージャーのリスナーを定義し、開始します。

```
*... On QM n
DEFINE LISTENER(TCP141 n) TRPTYPE(TCP) IPADDR(hostname) PORT(141 n) CONTROL(QMGR) REPLACE
START LISTENER(TCP141 n)
```

3. すべてのクラスターのクラスター名前リストを作成します。

```
*... On gateway queue manager QM1
DEFINE NAMELIST(ALL) NAMES(CL1, CL2) REPLACE
```

4. QM2 および QM4 完全リポジトリーを CL1 に、QM3 および QM5 完全リポジトリーを CL2 に作成します。

   a) CL1 の場合:

```
*... On QM2 and QM4
ALTER QMGR REPOS(CL1) DEFCLXQ(SCTQ)
```

   b) CL2 の場合:

```
*... On QM3 and QM5
ALTER QMGR REPOS(CL2) DEFCLXQ(SCTQ)
```

5. キュー・マネージャーとクラスターのそれぞれに、クラスター送信側チャネルとクラスター受信側チャネルを追加します。

   QM2、QM3、QM4、および QM5 で以下のコマンドを実行します。ここで、*c*、*n*、および *m* は、各キュー・マネージャーの 337 ページの表 26 に示されている値です。

表 *26.* クラスター *1* および *2* を作成する際のパラメーター値

キュー・マネージャー	クラスター *c*	その他のリポジトリー *n*	このリポジトリー *m*
QM2	1	4	2
QM4	1	2	4
QM3	2	5	3
QM5	2	3	5

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

6. 各クラスタにゲートウェイキューマネージャ QM1 を追加します。

```
*... On gateway queue manager QM1
DEFINE CHANNEL(CL1.QM2) CHLTYPE(CLUSSDR) CONNAME('localhost(1412)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL1.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL2.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL2) REPLACE
DEFINE CHANNEL(CL2.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL2) REPLACE
```

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

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

8. クラスタ化された各キューに対して、ゲートウェイキューマネージャ QM1 にキューエイリアスを追加します。

```
*... On gateway queue manager QM1
DEFINE QALIAS(Q1A) CLUSNL(ALL) TARGET(Q1) TARGTYPE(QUEUE) DEFBIND(NOTFIXED) REPLACE
```

9. クラスタ化されたすべてのキュー・マネージャーのクラスタ・キュー・マネージャーのエイリアス定義をゲートウェイ・キュー・マネージャーに追加します。 QM1 。

   レスポンス（返信）が必要な各部分リポジトリには、キューマネージャエイリアスが必要です。 キュー・マネージャー・エイリアスの目的は次の通りです。アプリケーションが（異なるクラスターのキュー・マネージャーから発信された）**ReplyTo** 属性を含むリクエスト・メッセージを処理している場合、アプリケーションはキュー・マネージャー・エイリアスを通じてリクエストに応答することができます。

   注 : 次の例では、キュー・マネージャのエイリアス名は QM2 と QM3 であり、これらは実際のキュー・マネージャ名（ **RQMNAME** ）である QM2 と QM3 と同じです。 アプリケーションに実際のキューマネージャー名を知られたくない場合は、別の名前を使用することができます。 例えば、キューマネージャのエイリアスは、 QM2 よりもアプリケーションにとって意味のある名前であれば、 PAYROLLQM とすることもできます。

```
*... On gateway queue manager QM1
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSNL(ALL) REPLACE
```

   ヒント : ゲートウェイキューマネージャー上のキューマネージャーエイリアスの定義 QM1 は、他のクラスター内のキューマネージャーを参照するメッセージを転送します。 クラスタ化キューマネージャのエイリアスを参照してください。

次のタスク

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

   a. QM2 でサンプル・プログラム **amqsput** を実行して、メッセージを書き込みます。

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```

   b. サンプル・プログラム **amqsget** を実行して、QM3 上の Q1 からメッセージを取得します。

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

2. キュー・マネージャー別名定義をテストするために、要求メッセージを送信し、一時動的応答キューで応答メッセージを受信します。

   この図は、応答メッセージが (RQ と呼ばれる) 一時動的キュー に戻すためにたどるパスを示しています。 QM3 に接続されているサーバー・アプリケーションは、キュー・マネージャー名 QM2 を使用して応答キューを開きます。 キューマネージャ名 QM2 は、ゲートウェイキューマネージャ QM1 上のクラスタキューマネージャエイリアスとして定義されています。 QM3 返信メッセージをゲートウェイキューマネージャ にルーティングします。 QM1 QM1 は、メッセージを QM2 に経路指定します。

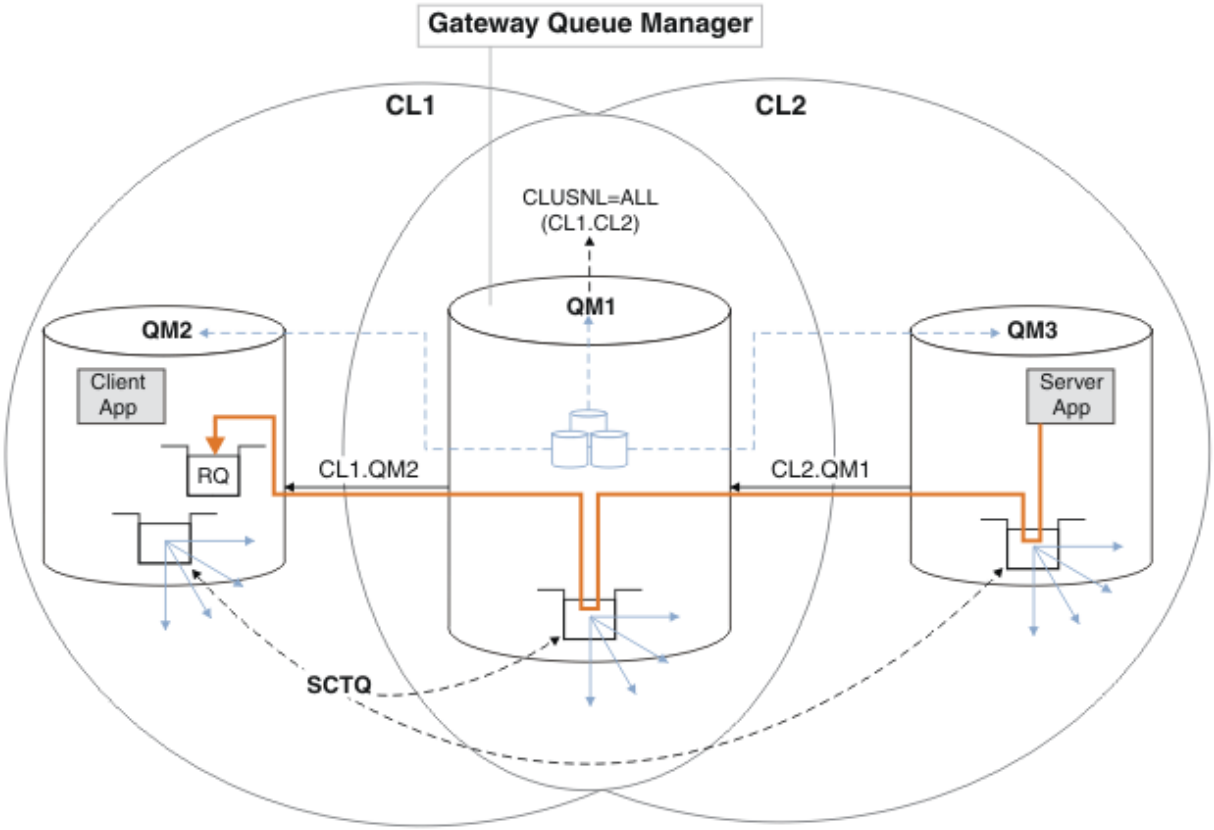


図 *48.* 応答メッセージを異なるクラスターに返すためのキュー・マネージャー別名の使用

このルーティングは、次のようにして行われます。 各クラスタのキュー・マネージャーには、ゲートウェイ・キュー・マネージャー QM1 上のキュー・マネージャー・エイリアス定義があります。 これらの別名は、すべてのクラスターでクラスター化されます。 それぞれの別名からキュー・マネージャーへと向かうグレーの破線矢印は、各キュー・マネージャー別名が、クラスターのうちの少なくとも 1 つにある実際のキュー・マネージャーに解決されることを示します。 この場合、QM2 別名はクラスター CL1 と CL2 の両方でクラスター化され、CL1 内の実際のキュー・マネージャー QM2 に解決されます。 サーバー・アプリケーションは、応答先キュー名 RQ、および応答先キュー・マネージャー名 QM2 を使用して応答メッセージを作成します。 メッセージはゲートウェイキューマネージャー QM1 にルーティングされます。キューマネージャーエイリアスの定義 QM2 がクラスター CL2 の QM1 で定義されている一方で、キューマネージャー QM2 はクラスター CL2 にないためです。 メッセージをターゲット・キュー・マネージャーに送信できないことから、この別名定義を持つキュー・マネージャーにメッセージが送信されます。

ゲートウェイキューマネージャー QM1 は、メッセージをクラスタ送信キューに配置し、 QM2 に転送します。 ゲートウェイキューマネージャー QM1 は、 QM1 上の QM2 のキューマネージャーエイリアスの定義で、 QM2 が実際のターゲットキューマネージャーとして定義されているため、メッセージを QM2 に転送します。 別名定義が参照できるのは、実際の定義だけで、自身を参照することはできないため、この定義は循環しません。 実際の定義は、ゲートウェイキューマネージャー QM1 によって解決されます。 QM1 と QM2 は同じクラスターである CL1 にあるためです。 ゲートウェイキューマネージャー QM1 は、CL1 のリポジトリから QM2 の接続情報を取得し、メッセージを QM2 に転送します。 メッセージを QM1 によって転送するためには、サーバー・アプリケーションが、オプション DEFBIND を MQBND_BIND_NOT_FIXED に設定して応答キューをオープンしていなければなりません。 サーバー・アプリケーションがオプション MQBND_BIND_ON_OPEN を設定して応答キューをオープンしている場合、メッセージは転送されず、送達不能キューに入れられます。

a. QM3 でトリガーを使用してクラスター化された要求キューを作成します。

```
*... On QM3
DEFINE QLOCAL(QR) CLUSTER(CL2) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
```

b. ゲートウェイキューマネージャ QM1 上で、 QR のクラスタキューエイリアスの定義を作成します。

```
*... On gateway queue manager QM1
DEFINE QALIAS(QRA) CLUSNL(ALL) TARGET(QR) TARGTYPE(QUEUE) DEFBIND(NOTFIXED) REPLACE
```

c. QM3 に、サンプル・エコー出力プログラム **amqsech** を開始するためのプロセス定義を作成します。

```
*... On QM3
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

d. QM2 に、サンプル・プログラム **amqsreq** のモデル・キューを作成し、一時動的応答キューを作成します。

```
*... On QM2
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

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

i) QM3 でトリガー・モニター・プログラムを実行します。

```
runmqtrm -m QM3
```

出力は次のとおりです。

```
C:\IBM\MQ>runmqtrm -m QM3
5724-H72 (C) Copyright IBM Corp. 1994, 2025. ALL RIGHTS RESERVED.
01/02/2012  16:17:15: IBM MQ trigger monitor started.

------------------------------------------------------
01/02/2012  16:17:15: Waiting for a trigger message
```

ii) QM2 でサンプル・プログラム **amqsreq** を実行して要求を書き込み、応答を待機します。

```
C:\IBM\MQ>amqsreq QRA QM2
Sample AMQSREQ0 start
server queue is QRA
replies to 4F2961C802290020
A request message from QM2 to QR on QM3

response <A request message from QM2 to QR on QM3>
no more replies
Sample AMQSREQ0 end
```

関連概念

アクセス制御と複数のクラスター伝送キュー

クラスター化: 複数のクラスター伝送キューの使用によるアプリケーションの分離

関連タスク

クラスター化: クラスター伝送キューの構成方法の計画

315 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』

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

リモート・キュー定義を追加して、ゲートウェイ・キュー・マネージャーから送信されたメッセージを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。 このソリューションは、クラスター化されたキュー・リモート定義と、別個の送信側チャネルおよび伝送キューを使用します。

始める前に

336 ページの『ゲートウェイ・キュー・マネージャーを使用した 2 つのオーバーラップするクラスターの作成』のハブ・スポーク・アーキテクチャーにデプロイされた IBM MQ クラスターを使用するクライアント/サーバー・アプリケーションに示されているオーバーラップするクラスターを、タスクで説明している手順に従って作成します。

このタスクについて

このソリューションは、分散キューイングを使用して、Server App アプリケーションのメッセージをゲートウェイ・キュー・マネージャーの他のメッセージ・トラフィックから分離します。 メッセージを他別の伝送キューおよび他のチャネルに方向転換するには、QM1 にクラスター化されたリモート・キュー定義を定義しなければなりません。 リモート・キュー定義には、QM3 上の Q1 へのメッセージのみを保管する、特定の伝送キューへの参照を組み込む必要があります。 342 ページの図 49 では、クラスター・キュー別名 Q1A がリモート・キュー定義 Q1R で補足され、伝送キューと送信側チャネルが追加されています。

このソリューションでは、すべての応答メッセージが共通 SYSTEM.CLUSTER.TRANSMIT.QUEUE を使用して返されます。

このソリューションには、同じクラスター内の同じキュー・マネージャー上にある複数の宛先キューへのトラフィックを簡単に分離できるという利点があります。 このソリューションの欠点は、それぞれに異なるキュー・マネージャー上にある Q1 の複数のコピーの間でクラスター・ワークロード・バランシングを使用できないことです。 この欠点を克服するには、343 ページの『クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』を参照してください。 また、ある伝送キューから別の伝送キューへの切り替えも管理する必要があります。

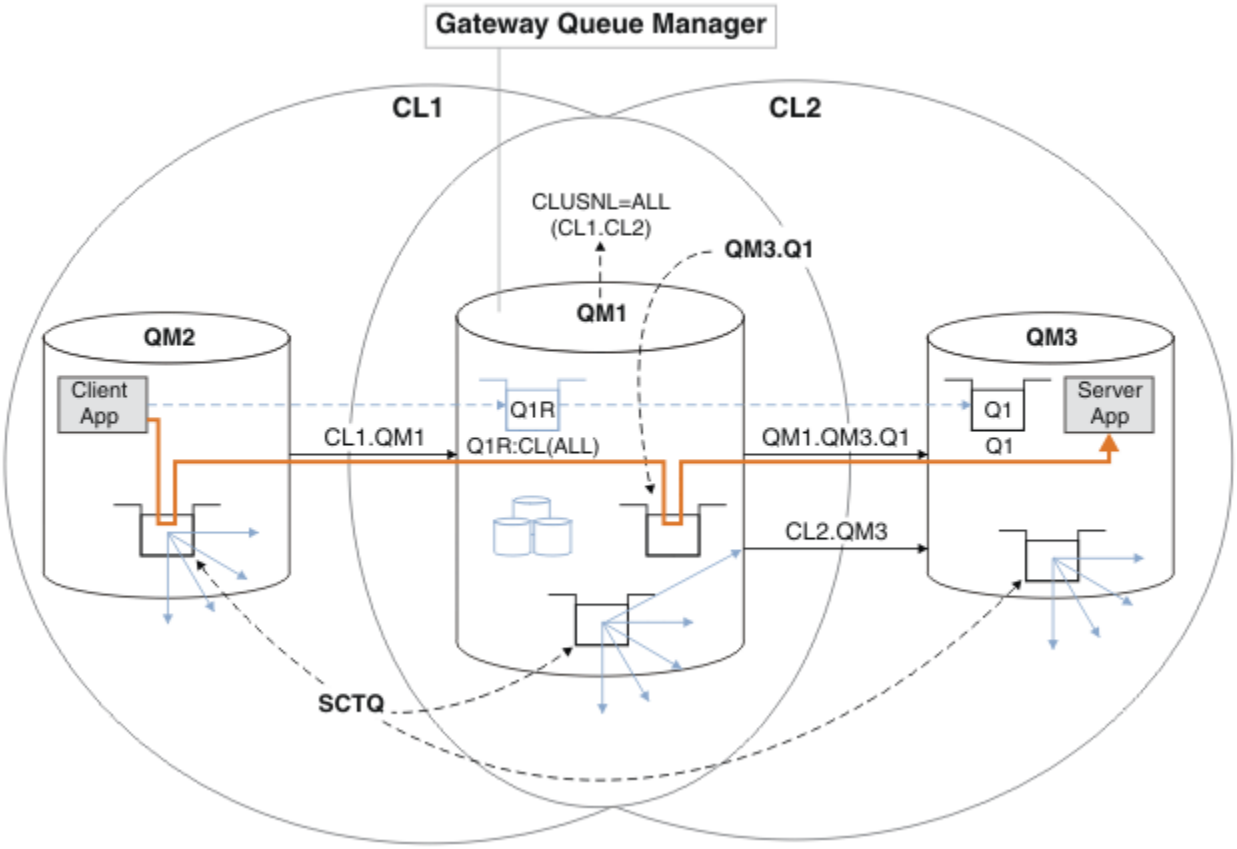


図 *49.* リモート・キュー定義を使用してハブ・スポーク・クラスター・アーキテクチャーにデプロイされたクライアント/サーバー・アプリケーション

手順

1. Q1 へのメッセージ・トラフィックをゲートウェイ・キュー・マネージャーから分離するためのチャネルを作成します。
   a) ゲートウェイ・キュー・マネージャー QM1 に、ターゲット・キュー・マネージャー QM3 への送信側チャネルを作成します。

   ```
   DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(SDR) CONNAME(QM3HostName(1413)) XMITQ(QM3.Q1) REPLACE
   ```

   b) ターゲット・キュー・マネージャー QM3 に、受信側チャネルを作成します。

   ```
   DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(RCVR) REPLACE
   ```

2. ゲートウェイ・キュー・マネージャーに、Q1 へのメッセージ・トラフィック用の伝送キューを作成します。

   ```
   DEFINE QLOCAL(QM3.Q1) USAGE(XMITQ) REPLACE
   START CHANNEL(QM1.QM3.Q1)
   ```

   伝送キューに関連付けるチャネルを開始すると、伝送キューがそのチャネルに関連付けられます。 伝送キューがチャネルに関連付けられると、チャネルが自動的に開始します。
3. ゲートウェイ・キュー・マネージャー上の Q1 のクラスター化されたキュー別名定義を、クラスター化されたリモート・キュー定義で補足します。

   ```
   DEFINE QREMOTE CLUSNL(ALL) RNAME(Q1) RQMNAME(QM3) XMITQ(QM3.Q1) REPLACE
   ```

次のタスク

構成をテストするために、ゲートウェイ・キュー・マネージャー QM1 上のクラスター化されたキュー・リモート定義 Q1R を使用して、メッセージを QM2 から QM3 上の Q1 にメッセージを送信します。

1. QM2 でサンプル・プログラム **amqsput** を実行して、メッセージを書き込みます。

```
C:\IBM\MQ>amqsput Q1R QM2
Sample AMQSPUT0 start
target queue is Q1R
Sample request message from QM2 to Q1 using Q1R

Sample AMQSPUT0 end
```

2. サンプル・プログラム **amqsget** を実行して、QM3 上の Q1 からメッセージを取得します。

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1R>
no more messages
Sample AMQSGET0 end
```

関連概念

クラスター化: 複数のクラスター伝送キューの使用によるアプリケーションの分離

アクセス制御と複数のクラスター伝送キュー

関連タスク

クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後 このソリューションは、追加のクラスター伝送キューを使用して、メッセージ・トラフィックをクラスター内の 1 つのキュー・マネージャーに分離します。

クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後、これは、追加のクラスターを使用して、特定のクラスター・キューにメッセージを分離します。

クラスター伝送キューを区別するようにデフォルトを変更して、メッセージ・トラフィックを分離する

キュー・マネージャーがクラスター・キューまたはトピックへのメッセージを伝送キューに保管するデフォルトの方法を変更できます。 デフォルトを変更することは、ゲートウェイ・キュー・マネージャーでクラスター・メッセージを分離する手段となります。

クラスター化: クラスター伝送キューの構成方法の計画

315 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』

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

クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後 このソリューションは、追加のクラスター伝送キューを使用して、メッセージ・トラフィックをクラスター内の 1 つのキュー・マネージャーに分離します。

始める前に

1. ゲートウェイ・キュー・マネージャーが IBM MQ 上になければなりません。
2. 336 ページの『ゲートウェイ・キュー・マネージャーを使用した 2 つのオーバーラップするクラスターの作成』のハブ・スポーク・アーキテクチャーにデプロイされた IBM MQ クラスターを使用するクライアント/サーバー・アプリケーションに示されているオーバーラップするクラスターを、タスクで説明している手順に従って作成します。

このタスクについて

ゲートウェイ・キュー・マネージャー QM1 で、伝送キューを追加して、そのキュー属性 CLCHNAME を設定します。 CLCHNAME は、QM3 上のクラスター受信側チャネルの名前に設定します (345 ページの図 50 を参照)。

このソリューションには、341 ページの『リモート・キュー定義を追加して、ゲートウェイ・キュー・マネージャーから送信されたメッセージを分離する』で説明したソリューションに勝る数々の利点があります。

- 追加しなければならない定義の数が少なくなります。
- 同じクラスター CL2 内の異なるキュー・マネージャー上にあるターゲット・キュー Q1 の複数のコピーの間でのワークロード・バランシングをサポートします。
- チャネルが再始動すると、ゲートウェイ・キュー・マネージャーが自動的に新しい構成に切り替え、メッセージが失われることがありません。
- ゲートウェイ・キュー・マネージャーは、引き続きメッセージを受信した順に転送します。 これは、QM3 のキュー Q1 へのメッセージがまだ SYSTEM.CLUSTER.TRANSMIT.QUEUE にあるときに切り替えが発生した場合にも該当します。

345 ページの図 50 でクラスター・メッセージ・トラフィックを分離するための構成では、341 ページの『リモート・キュー定義を追加して、ゲートウェイ・キュー・マネージャーから送信されたメッセージを分離する』で紹介されているリモート・キューを使用した構成ほど効果的にトラフィックを分離することはできません。 クラスター CL2 内のキュー・マネージャー QM3 が多数の異なるクラスター・キューとサーバー・アプリケーションをホストしている場合、これらのキューのすべてが、QM1 を QM3 に接続するクラスター・チャネル CL2.QM3 を共有します。 その他のフローは、345 ページの図 50 のグレイの矢印で示されています。この矢印は、SYSTEM.CLUSTER.TRANSMIT.QUEUE からクラスター送信側チャネル CL2.QM3 への潜在的なクラスター・メッセージ・トラフィックを表しています。

解決策は、キュー・マネージャーが特定のクラスター内で 1 つのクラスター・キューだけをホストするように制限することです。 キュー・マネージャーがすでに多数のクラスター・キューをホストしている場合には、この制限に対応するために、別のキュー・マネージャーを作成するか、または別のクラスターを作成する必要があります (347 ページの『クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』を参照)。

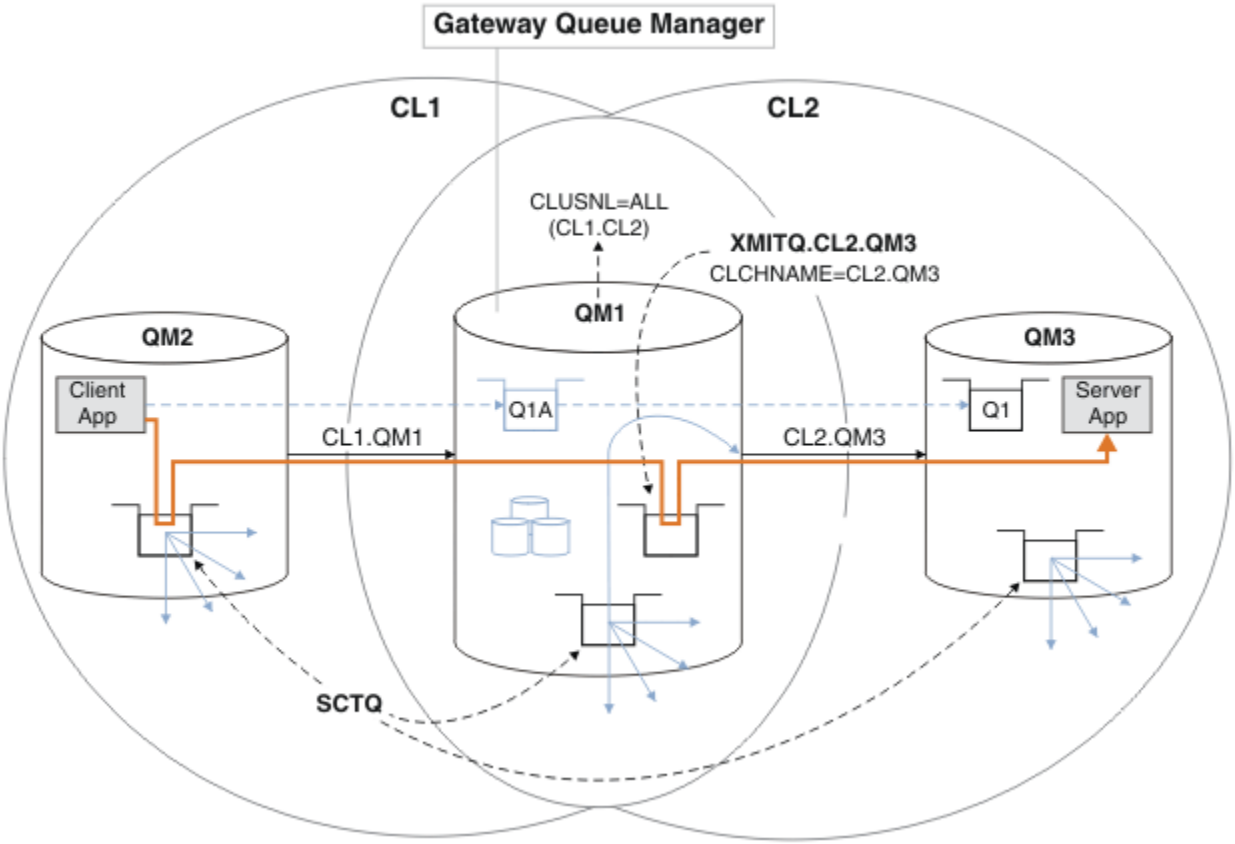


図 *50.* 追加のクラスター伝送キューを使用してハブ・スポーク・アーキテクチャーにデプロイされたクライアント/サーバー・アプリケーション

手順

1. ゲートウェイ・キュー・マネージャー QM1 で、クラスター送信側チャネル CL2.QM3 用の追加のクラスター伝送キューを作成します。

```
 *... on QM1
DEFINE QLOCAL(XMITQ.CL2.QM3) USAGE(XMITQ) CLCHNAME(CL2.QM3)
```

2. 伝送キュー XMITQ.CL2.QM3 を使用するように切り替えます。
   a) クラスター送信側チャネル CL2.QM3 を停止します。

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

   コマンドが受け入れられたという応答が返されます。

```
AMQ8019: Stop IBM MQ channel accepted.
```

   b) チャネル CL2.QM3 が停止していることを確認します。

   チャネルが停止しない場合は、FORCE オプションを指定した **STOP CHANNEL** コマンドを再度実行できます。 FORCE オプションを設定する一例は、チャネルが停止しないため、そのチャネルを同期化するもう一方のキュー・マネージャーを再始動できない場合です。

```
*... On QM1
start
```

   応答は、チャネル状況の要約です。

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)              CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))      CURRENT
RQMNAME(QM3)                  STATUS(STOPPED)
SUBSTATE(MQGET)               XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

c) チャネル CL2.QM3 を開始します。

```
*... On QM1
START CHANNEL(CL2.QM3)
```

コマンドが受け入れられたという応答が返されます。

```
AMQ8018: Start IBM MQ channel accepted.
```

d) チャネルが開始したことを確認します。

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

応答は、チャネル状況の要約です。

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))        CURRENT
RQMNAME(QM3)                    STATUS(RUNNING)
SUBSTATE(MQGET)                 XMITQ(XMITQ.CL2.QM3)
```

e) 伝送キューが切り替えられたことを確認します。

メッセージ " アンカル 7341 チャネル CL2.QM3 の伝送キューは XMITQ です。" CL2"第 3 四半期"のゲートウェイ・キュー・マネージャーのエラー・ログをモニターしてください。

次のタスク

単独の伝送キューをテストするために、キュー別名定義 Q1A を使用して、メッセージを QM2 から QM3 上の Q1 に送信します。

1. QM2 でサンプル・プログラム **amqsput** を実行して、メッセージを書き込みます。

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```

2. サンプル・プログラム **amqsget** を実行して、QM3 上の Q1 からメッセージを取得します。

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

関連概念

アクセス制御と複数のクラスター伝送キュー

クラスター化: 複数のクラスター伝送キューの使用によるアプリケーションの分離
293 ページの『クラスター伝送キューとクラスター送信側チャネルの操作』
クラスター・キュー・マネージャー間のメッセージは、クラスター伝送キューに保管されてから、クラスター送信側チャネルによって転送されます。 どの時点でも、クラスター送信側チャネルが関連付けられる伝送キューは 1 つです。 チャネルの構成を変更すると、そのチャネルは次回開始したときに、別の伝送キューに切り替わる可能性があります。 この切り替え処理は自動化されていて、トランザクションで実行されます。

関連タスク
リモート・キュー定義を追加して、ゲートウェイ・キュー・マネージャーから送信されたメッセージを分離する
ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。 このソリューションは、クラスター化されたキュー・リモート定義と、別個の送信側チャネルおよび伝送キューを使用します。

クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する
ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後、これは、追加のクラスターを使用して、特定のクラスター・キューにメッセージを分離します。

クラスター伝送キューを区別するようにデフォルトを変更して、メッセージ・トラフィックを分離する
キュー・マネージャーがクラスター・キューまたはトピックへのメッセージを伝送キューに保管するデフォルトの方法を変更できます。 デフォルトを変更することは、ゲートウェイ・キュー・マネージャーでクラスター・メッセージを分離する手段となります。

クラスター化: クラスター伝送キューの構成方法の計画
315 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』
以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。 クラスター・キューおよびトピックへのメッセージは、複数のクラスター伝送キューを使用して転送されます。

クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後、これは、追加のクラスターを使用して、特定のクラスター・キューにメッセージを分離します。

始める前に

このタスクで説明する手順は、345 ページの図 50 に示されている構成を変更するために作成されています。

1. ゲートウェイ・キュー・マネージャーが IBM MQ 上になければなりません。
2. 336 ページの『ゲートウェイ・キュー・マネージャーを使用した 2 つのオーバーラップするクラスターの作成』のハブ・スポーク・アーキテクチャーにデプロイされた IBM MQ クラスターを使用するクライアント/サーバー・アプリケーションに示されているオーバーラップするクラスターを、タスクで説明している手順に従って作成します。
3. 343 ページの『クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』の 345 ページの図 50 に示されている手順に従って、クラスターが追加されていないソリューションを作成します。 これを、このタスクの手順のベースとして使用します。

このタスクについて

343 ページの『クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』で説明した、単一アプリケーションへのメッセージ・

トラフィックを分離するためのソリューションが有効に働くのは、ターゲット・クラスター・キューがキュー・マネージャー上の唯一のクラスター・キューである場合です。 そうでない場合には、2 つの選択肢があります。 1 つはキューを別のキュー・マネージャーに移動すること、もう 1 つはキューをキュー・マネージャー上の他のクラスター・キューと分離するクラスターを作成することです。

このタスクでは、クラスターを追加してターゲット・キューを分離する手順を説明します。 クラスターは、その目的のためだけに追加します。 実際には、クラスターおよびクラスターの命名体系を設定するプロセスの中で、体系的に、特定のアプリケーションを分離するタスクに取り組んでください。 キューの分離が必要になるたびにクラスターを追加すると、最終的に多数のクラスターを管理しなければならなくなります。 このタスクでは、343 ページの『クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』の構成を変更して、QM3 上の Q1 を分離するためのクラスター CL3 を追加します。 アプリケーションは、この変更作業の初めから終わりまで実行し続けます。

新しい定義と変更された定義は、349 ページの図 51 に強調表示されています。 変更内容を要約すると、まず、クラスターを作成します。これは、クラスターの新しい完全リポジトリーも作成しなければならないことを意味します。 この例では、QM3 が CL3 の完全リポジトリーの 1 つにされます。 新しいクラスターにゲートウェイ・キュー・マネージャーを追加するために、QM1 のクラスター送信側チャネルとクラスター受信側チャネルを作成します。 Q1 の定義を変更して、CL3 に切り替えます。 ゲートウェイ・キュー・マネージャーのクラスター名前リストを変更し、新規クラスター・チャネルを使用するクラスター伝送キューを追加します。 最後に、キュー別名 Q1A を新規クラスター名前リストに切り替えます。

IBM MQ は、 343 ページの『クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』で追加した伝送キュー XMITQ.CL2.QM3 から新しい伝送キュー XMITQ.CL3.QM3 にメッセージを自動的に転送することはできません。 自動的にメッセージを転送できるのは、両方の伝送キューが同じクラスター送信側チャネルによって処理される場合のみです。 代わりに、このタスクでは、適切だと考えられる手動による切り替え方法の 1 つを説明します。 転送が完了した時点で、QM3 上の他の CL2 クラスター・キューのデフォルト・クラスター伝送キューを使用するように戻すことを選択できます。 あるいは、引き続き XMITQ.CL2.QM3 を使用することもできます。 デフォルト・クラスター伝送キューに戻すことにした場合、ゲートウェイ・キュー・マネージャーが自動的に切り替えを管理します。

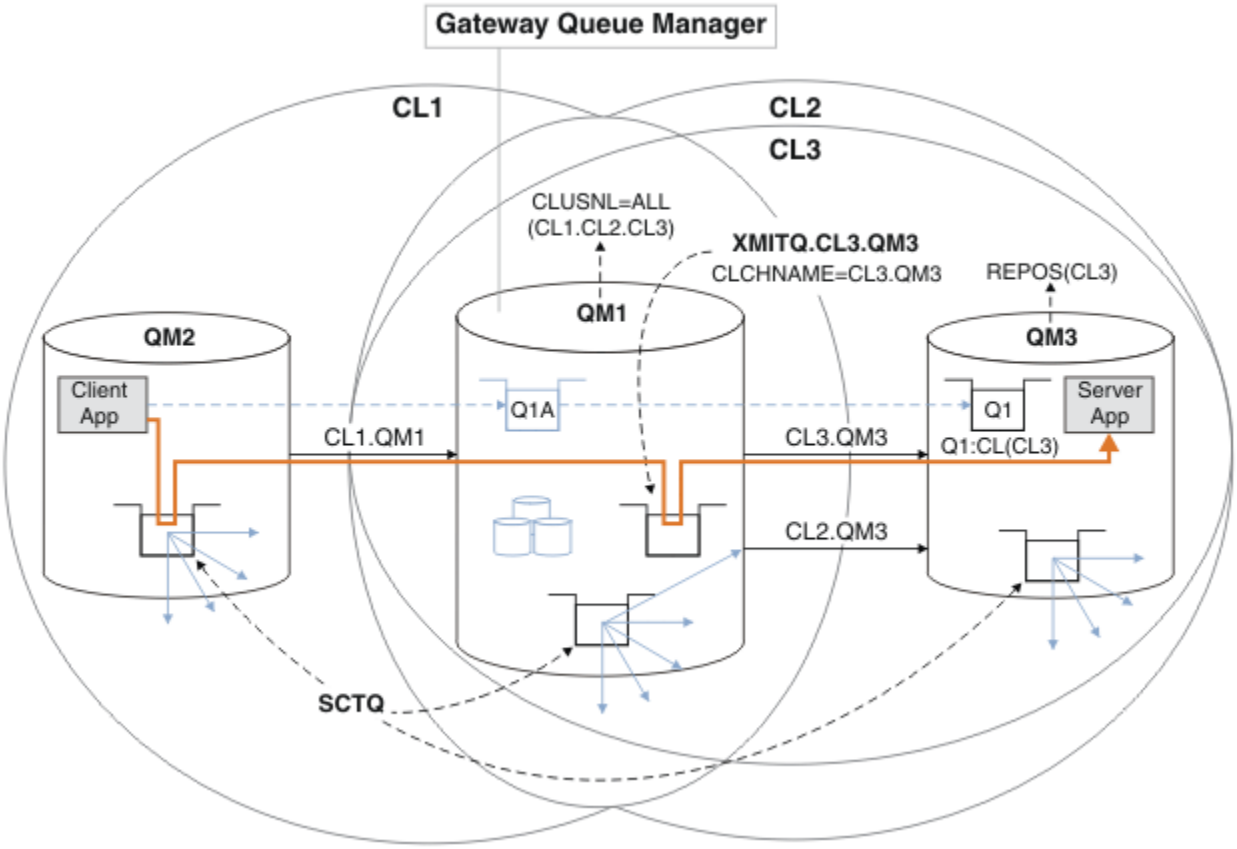


図 *51.* 追加のクラスターを使用して、同じキュー・マネージャー上の多数のクラスター・キューのうちの *1* つに向かうメッセージ・トラフィックをゲートウェイ・キュー・マネージャー内で分離する

手順

1. キュー・マネージャー QM3 および QM5 を変更して、CL2 と CL3 両方のリポジトリーにします。

   キュー・マネージャーを複数のクラスターのメンバーにするには、キュー・マネージャーがクラスター名前リストを使用して、それがメンバーとなっているクラスターを識別する必要があります。

   ```
   *... On QM3 and QM5
   DEFINE NAMELIST(CL23) NAMES(CL2, CL3) REPLACE
   ALTER QMGR REPOS(' ') REPOSNL(CL23)
   ```

2. CL3 のキュー・マネージャー QM3 と QM5 との間のチャネルを定義します。

   ```
   *... On QM3
   DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
   DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSRCVR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE

   *... On QM5
   DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
   DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
   ```

3. ゲートウェイ・キュー・マネージャーを CL3 に追加します。

   ゲートウェイ・キュー・マネージャーを追加するために、QM1 を部分リポジトリーとして CL3 に追加します。 部分リポジトリーを作成するには、クラスター送信側チャネルとクラスター受信側チャネルを QM1 に追加します。

   さらに、ゲートウェイ・キュー・マネージャーに接続されたすべてのクラスターの名前リストに CL3 を追加します。

   ```
   *... On QM1
   DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
   ```

```
DEFINE CHANNEL(CL3.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL3) REPLACE
ALTER NAMELIST(ALL) NAMES(CL1, CL2, CL3)
```

4. ゲートウェイ・キュー・マネージャー QM1 に、QM3 の CL3 に向かうメッセージ用のクラスター伝送キューを追加します。

   最初に、伝送キューを切り替える準備ができるまで、伝送キューからメッセージを転送するクラスター送信側チャネルを停止します。

```
*... On QM1
DEFINE QLOCAL(XMITQ.CL3.QM3) USAGE(XMITQ) CLCHNAME(CL3.QM3) GET(DISABLED) REPLACE
```

5. 既存のクラスター伝送キュー XMITQ.CL2.QM3 からメッセージを排出します。

   以下のサブ手順は、メッセージがゲートウェイ・キュー・マネージャーに到着した順序と一致するように、Q1 内でメッセージの順序を保持することを目的としています。 クラスターでは、メッセージの順序は完全に保証されませんが、おそらく順序は保持されます。 メッセージの順序を保証することが必要な場合には、アプリケーションでメッセージの順序を定義する必要があります (メッセージがキューから取り出される順序を参照)。

   a) QM3 上のターゲット・キュー Q1 を CL2 から CL3 に変更します。

```
*... On QM3
ALTER QLOCAL(Q1) CLUSTER(CL3)
```

   b) メッセージの配信が開始されるまで、XMITQ.CL3.QM3 をモニターします。

   Q1 から CL3 への切り替えがゲートウェイ・キュー・マネージャーに伝搬されると、XMITQ.CL3.QM3 へのメッセージ配信が開始されます。

```


*... On QM1
DISPLAY QUEUE(XMITQ.CL3.QM3) CURDEPTH
```

   c) QM3 上の Q1 への配信を待機中のメッセージがなくなるまで、XMITQ.CL2.QM3 をモニターします。

   注: XMITQ.CL2.QM3 は、CL2 のメンバーとなっている他の QM3 上のキューへのメッセージを保管している可能性があります。その場合、キューの項目数はゼロになりません。

```


*... On QM1
DISPLAY QUEUE(XMITQ.CL2.QM3) CURDEPTH
```

   d) 新規クラスター伝送キュー XMITQ.CL3.QM3 からの取得を有効にします。

```
*... On QM1
ALTER QLOCAL(XMITQ.CL3.QM3) GET(ENABLED)
```

6. 前のクラスター伝送キュー XMITQ.CL2.QM3 が必要ない場合には、これを削除します。

   QM3 上の CL2 内にあるクラスター・キューへのメッセージは、ゲートウェイ・キュー・マネージャー QM1 上のデフォルト・クラスター伝送キューを使用する状態に戻ります。 デフォルト・クラスター伝送キューは、SYSTEM.CLUSTER.TRANSMIT.QUEUE または SYSTEM.CLUSTER.TRANSMIT.CL2.QM3 のいずれかです。 このどちらであるかは、QM1 のキュー・マネージャー属性 **DEFCLXQ** の値が、SCTQ または CHANNEL のどちらであるかによって決まります。 クラスター送信側チャネル CL2.QM3 が次に開始した時点から、キュー・マネージャーは自動的に XMITQ.CL2.QM3 からメッセージを転送します。

   a) 伝送キュー XMITQ.CL2.QM3 を、クラスター伝送キューから一般の伝送キューに変更します。

   これにより、伝送キューとすべてのクラスター送信側チャネルとの関連付けが解除されます。 これに応じて、クラスター送信側チャネルが次に開始した時点から、IBM MQ は自動的にメッセージを XMITQ.CL2.QM3 からデフォルト・クラスター伝送キューに転送します。 それまでは、QM3 上の CL2 へのメッセージは、引き続き XMITQ.CL2.QM3 に配置されます。

```
*... On QM1
ALTER QLOCAL(XMITQ.CL2.QM3) CLCHNAME(' ')
```

b) クラスター送信側チャネル CL2.QM3 を停止します。

クラスター送信側チャネルを停止してから再始動すると、XMITQ.CL2.QM3 からデフォルト・クラスター伝送キューへのメッセージの転送が開始します。 通常は、手動でチャネルを手動で停止して開始することによって、転送を開始します。 チャネルの切断間隔が満了してチャネルがシャットダウンした後、再始動する場合には、自動的に転送が開始します。

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

コマンドが受け入れられたという応答が返されます。

```
AMQ8019: Stop IBM MQ channel accepted.
```

c) チャネル CL2.QM3 が停止していることを確認します。

チャネルが停止しない場合は、FORCE オプションを指定した **STOP CHANNEL** コマンドを再度実行できます。 FORCE オプションを設定する一例は、チャネルが停止しないため、そのチャネルを同期化するもう一方のキュー・マネージャーを再始動できない場合です。

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

応答は、チャネル状況の要約です。

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))                CURRENT
RQMNAME(QM3)                            STATUS(STOPPED)
SUBSTATE(MQGET)                         XMITQ(XMITQ.CL2.QM3)
```

d) チャネル CL2.QM3 を開始します。

```
*... On QM1
START CHANNEL(CL2.QM3)
```

コマンドが受け入れられたという応答が返されます。

```
AMQ8018: Start IBM MQ channel accepted.
```

e) チャネルが開始したことを確認します。

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

応答は、チャネル状況の要約です。

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)      CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))  CURRENT
RQMNAME(QM3)        STATUS(RUNNING)
SUBSTATE(MQGET)      XMITQ(SYSTEM.CLUSTER.TRANSMIT. QUEUE|CL2.QM3)
```

f) メッセージのゲートウェイ・キュー・マネージャーのエラー・ログをモニターする"AMQ734 チャネル CL2.QM3 の伝送キューは SYSTEM.CLUSTER.TRANSMIT です。 *QUEUE*|*CL2.QM3* ".

g) クラスター伝送キュー XMITQ.CL2.QM3 を削除します。

```
*... On QM1
DELETE QLOCAL(XMITQ.CL2.QM3)
```

次のタスク

クラスター・キューを単独でテストするために、QM2 からキュー別名定義 Q1A を使用して、メッセージを QM3 上の Q1 に送信します。

1. QM2 でサンプル・プログラム **amqsput** を実行して、メッセージを書き込みます。

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```

2. サンプル・プログラム **amqsget** を実行して、QM3 上の Q1 からメッセージを取得します。

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

関連概念

アクセス制御と複数のクラスター伝送キュー

クラスター化: 複数のクラスター伝送キューの使用によるアプリケーションの分離

293 ページの『クラスター伝送キューとクラスター送信側チャネルの操作』

クラスター・キュー・マネージャー間のメッセージは、クラスター伝送キューに保管されてから、クラスター送信側チャネルによって転送されます。 どの時点でも、クラスター送信側チャネルが関連付けられる伝送キューは 1 つです。 チャネルの構成を変更すると、そのチャネルは次回開始したときに、別の伝送キューに切り替わる可能性があります。 この切り替え処理は自動化されていて、トランザクションで実行されます。

関連タスク

リモート・キュー定義を追加して、ゲートウェイ・キュー・マネージャーから送信されたメッセージを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。 このソリューションは、クラスター化されたキュー・リモート定義と、別個の送信側チャネルおよび伝送キューを使用します。

クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後 このソリューションは、追加のクラスター伝送キューを使用して、メッセージ・トラフィックをクラスター内の 1 つのキュー・マネージャーに分離します。

クラスター伝送キューを区別するようにデフォルトを変更して、メッセージ・トラフィックを分離する

キュー・マネージャーがクラスター・キューまたはトピックへのメッセージを伝送キューに保管するデフォルトの方法を変更できます。 デフォルトを変更することは、ゲートウェイ・キュー・マネージャーでクラスター・メッセージを分離する手段となります。

クラスター化: クラスター伝送キューの構成方法の計画

315 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』

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

クラスター伝送キューを区別するようにデフォルトを変更して、メッセージ・トラフィックを分離する

キュー・マネージャーがクラスター・キューまたはトピックへのメッセージを伝送キューに保管するデフォルトの方法を変更できます。 デフォルトを変更することは、ゲートウェイ・キュー・マネージャーでクラスター・メッセージを分離する手段となります。

始める前に

1. ゲートウェイ・キュー・マネージャーが IBM MQ 上になければなりません。
2. 336 ページの『ゲートウェイ・キュー・マネージャーを使用した 2 つのオーバーラップするクラスターの作成』のハブ・スポーク・アーキテクチャーにデプロイされた IBM MQ クラスターを使用するクライアント/サーバー・アプリケーションに示されているオーバーラップするクラスターを、タスクで説明している手順に従って作成します。

このタスクについて

複数のクラスター・キューを使用するアーキテクチャーを実装するには、ゲートウェイ・キュー・マネージャーが IBM MQ 上になければなりません。 複数のクラスター伝送キューを使用するために必要な作業は、ゲートウェイ・キュー・マネージャーでデフォルト・クラスター伝送キュー・タイプを変更することだけです。 QM1 上のキュー・マネージャー属性 **DEFCLXQ** の値を SCTQ から CHANNEL に変更します。 354 ページの図 52 を参照。 この図には、1 つのメッセージ・フローが示されています。 他のキュー・マネージャーや他のクラスターへのフローには、キュー・マネージャーが追加の永続動的クラスター伝送キューを作成します。 各クラスター送信側チャネルは、それぞれに異なるクラスター伝送キューからメッセージを転送します。

ゲートウェイ・キュー・マネージャーをクラスターに初めて接続しているのでない限り、変更は即時に適用されません。 このタスクには、既存の構成に対する変更を管理する際の標準的な手順が含まれます。 キュー・マネージャーが初めてクラスターに参加するときに、個々のクラスター伝送キューを使用するようにセットアップするには、315 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』を参照してください。

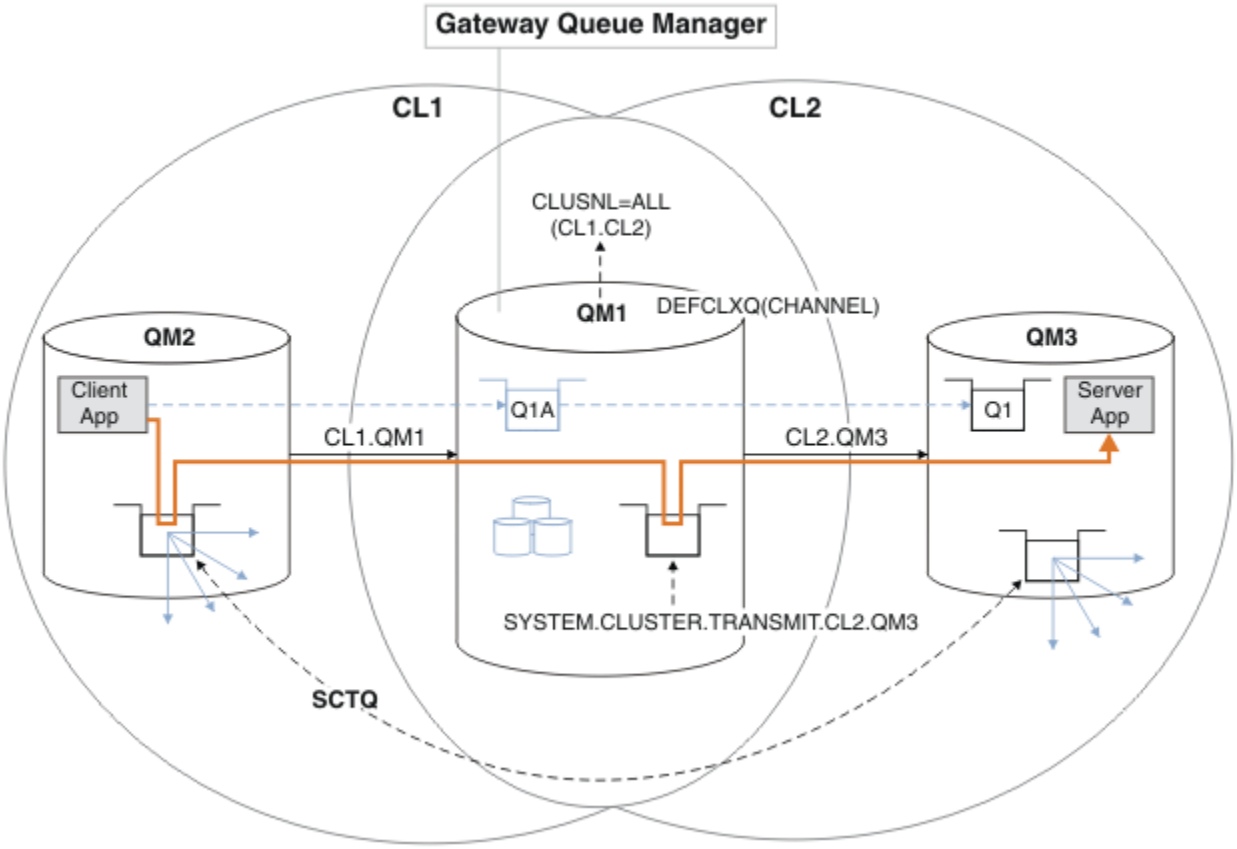


図 *52.* ゲートウェイ・キュー・マネージャーで個々のクラスター伝送キューを使用するハブ・スポーク・アーキテクチャーにデプロイされたクライアント/サーバー・アプリケーション

手順

1. 個々のクラスター伝送キューを使用するようにゲートウェイ・キュー・マネージャーを変更します。

```
*... On QM1
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. 個々のクラスター伝送キューに切り替えます。

   実行中でないクラスター送信側チャネルは、いずれも次に開始するときに、個々の伝送キューを使用するように切り替わります。

   実行中のチャネルを切り替えるには、キュー・マネージャーを再始動するか、または以下のステップに従います。

   a) SYSTEM.CLUSTER.TRANSMIT.QUEUE を使用して実行中のクラスター送信側チャネルをリストします。

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
```

   応答は、チャネル状況レポートのリストです。

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)             CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1412))     CURRENT
RQMNAME(QM2)                 STATUS(RUNNING)
SUBSTATE(MQGET)              XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
```

```
CHANNEL(CL2.QM3)            CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))    CURRENT
RQMNAME(QM3)                STATUS(RUNNING)
SUBSTATE(MQGET)             XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)            CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1415))    CURRENT
RQMNAME(QM5)                STATUS(RUNNING)
SUBSTATE(MQGET)             XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)            CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1414))    CURRENT
RQMNAME(QM4)                STATUS(RUNNING)
SUBSTATE(MQGET)             XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

b) 実行中のチャネルを停止します。

リストに含まれるチャネルごとに、以下のコマンドを実行します。

```
*... On QM1
STOP CHANNEL(ChannelName)
```

ここで、 *ChannelName* は、それぞれの CL1.QM2, CL1.QM4, CL1.QM3, CL1.QM5 です。

コマンドが受け入れられたという応答が返されます。

```
AMQ8019: Stop IBM MQ channel accepted.
```

c) 停止されたチャネルをモニターします。

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
```

応答は、まだ実行中のチャネルと停止されたチャネルのリストです。

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)            CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1412))    CURRENT
RQMNAME(QM2)                STATUS(STOPPED)
SUBSTATE( )                 XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)            CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))    CURRENT
RQMNAME(QM3)                STATUS(STOPPED)
SUBSTATE( )                 XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)            CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1415))    CURRENT
RQMNAME(QM5)                STATUS(STOPPED)
SUBSTATE( )                 XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)            CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1414))    CURRENT
RQMNAME(QM4)                STATUS(STOPPED)
SUBSTATE( )                 XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

d) 停止されたチャネルのそれぞれを開始します。

このステップは、実行中であったすべてのチャネルに対して行ってください。 チャネルが停止しない場合は、FORCE オプションを指定した **STOP CHANNEL** コマンドを再度実行できます。 FORCE オ

プションを設定する一例は、チャネルが停止しないため、そのチャネルを同期化するもう一方のキュー・マネージャーを再始動できない場合です。

```
*... On QM1
START CHANNEL(CL2.QM5)
```

コマンドが受け入れられたという応答が返されます。

```
AMQ8018: Start IBM MQ channel accepted.
```

e) 切り替えられる伝送キューをモニターします。

メッセージのゲートウェイ・キュー・マネージャーのエラー・ログをモニターする"AMQ734 チャネル CL2.QM3 の伝送キューは SYSTEM.CLUSTER.TRANSMIT です。 *QUEUE*|*CL2.QM3* ".

f) SYSTEM.CLUSTER.TRANSMIT.QUEUE が使用されていないことを確認します。

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
DISPLAY QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE) CURDEPTH
```

応答として、以下のような、チャネル状況レポートのリストおよび SYSTEM.CLUSTER.TRANSMIT.QUEUE の深さが返されます。

```
AMQ8420: Channel Status not found.
AMQ8409: Display Queue details.
QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE)    TYPE(QLOCAL)
CURDEPTH(0)
```

g) 開始されたチャネルをモニターします。

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

応答として、チャネルのリストが返されます。この場合は、新しいデフォルト・クラスター伝送キューを使用して既に実行されているチャネルであり、以下のようになります。

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1412))                CURRENT
RQMNAME(QM2)                            STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL1.QM2)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))                CURRENT
RQMNAME(QM3)                            STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM3)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1415))                CURRENT
RQMNAME(QM5)                            STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM5)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1414))                CURRENT
RQMNAME(QM4)                            STATUS(RUNNING)
```

```
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL1.QM4)
```

次のタスク

1. 自動的に定義されたクラスター伝送キューをテストするために、メッセージを QM2 から QM3 上の Q1 に送信し、キュー別名定義 Q1A を使用してキュー名を解決します。

   a. QM2 でサンプル・プログラム **amqsput** を実行して、メッセージを書き込みます。

   ```
   C:\IBM\MQ>amqsput Q1A QM2
   Sample AMQSPUT0 start
   target queue is Q1A
   Sample request message from QM2 to Q1 using Q1A

   Sample AMQSPUT0 end
   ```

   b. サンプル・プログラム **amqsget** を実行して、QM3 上の Q1 からメッセージを取得します。

   ```
   C:\IBM\MQ>amqsget Q1 QM3
   Sample AMQSGET0 start
   message <Sample request message from QM2 to Q1 using Q1A>
   no more messages
   Sample AMQSGET0 end
   ```

2. クラスター・キューに対するメッセージが発生したキュー・マネージャーでクラスター・キューのセキュリティーを構成して、セキュリティーを再構成するかどうかを検討します。

関連概念

アクセス制御と複数のクラスター伝送キュー

クラスター化: 複数のクラスター伝送キューの使用によるアプリケーションの分離

関連タスク

リモート・キュー定義を追加して、ゲートウェイ・キュー・マネージャーから送信されたメッセージを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。 このソリューションは、クラスター化されたキュー・リモート定義と、別個の送信側チャネルおよび伝送キューを使用します。

クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後 このソリューションは、追加のクラスター伝送キューを使用して、メッセージ・トラフィックをクラスター内の 1 つのキュー・マネージャーに分離します。

クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。 変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後、これは、追加のクラスターを使用して、特定のクラスター・キューにメッセージを分離します。

クラスター化: クラスター伝送キューの構成方法の計画

315 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』

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

キュー・マネージャーからのクラスター・キューの削除

トロントで INVENTQ キューを無効にします。 すべての在庫管理メッセージをニューヨークに送信し、トロントの INVENTQ キューが空になったらそれを削除します。

始める前に

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- 320 ページの『キューをホストするキュー・マネージャーの追加』で説明されているように、INVENTORY クラスターがセットアップされています。 このクラスターには、4 つのキュー・マネージャーが格納されています。 LONDON と NEWYORK はどちらも完全リポジトリーを保有しています。 PARIS と TORONTO は、部分リポジトリーを保有しています。 在庫管理アプリケーションはニューヨークおよびトロントのシステムで動作し、INVENTQ キューのメッセージが到着すると実行されます。
- ワークロードが軽減されたので、在庫管理アプリケーションをトロントで実行する必要はなくなりました。 キュー・マネージャー TORONTO によってホストされている INVENTQ キューを無効にし、さらに TORONTO によって NEWYORK の INVENTQ キューにメッセージを送信することが必要です。
- ネットワークは、4 つのシステム間すべてに接続されています。
- ネットワーク・プロトコルは、TCP です。

このタスクについて

以下の手順に従って、クラスター・キューを除去します。

手順

1. キューが使用できないことを示す

   キューをクラスターから削除するには、クラスター名をローカル・キュー定義から削除します。 TORONTO 上の INVENTQ を変更して、クラスターの残りの部分からアクセスできないようにします。

   ```
   ALTER QLOCAL(INVENTQ) CLUSTER(' ')
   ```

2. キューが使用できないことを確認する

   LONDON または NEWYORK のいずれかの完全リポジトリー・キュー・マネージャーで、次のコマンドを発行して、キューがキュー・マネージャー TORONTO にホストされていないことを確認します。

   ```
   DIS QCLUSTER (INVENTQ)
   ```

   ALTER コマンドが正常に実行されると、TORONTO は結果にリストされません。

3. キューを無効にする

   次のように TORONTO で INVENTQ キューを無効にして、このキューにこれ以上メッセージを書き込むことができないようにします。

   ```
   ALTER QLOCAL(INVENTQ) PUT(DISABLED)
   ```

   この結果、MQOO_BIND_ON_OPEN によってこのキューに転送中のメッセージは、送達不能キューに転送されます。 すべてのアプリケーションを停止して、このキュー・マネージャーのキューにメッセージが明示的に書き込まれることのないようにする必要があります。

4. キューが空になるまでキューをモニターする

   属性 IPPROCS、 OPPROCS、および CURDEPTH を指定して DISPLAY QUEUE コマンドを使用するか、IBM i で **WRKMQMSTS** コマンドを使用して、キューをモニターします。 入出力処理の数および現在のキューのサイズがすべてゼロのとき、キューは空です。

5. チャネルをモニターして、未確定メッセージがないことを確認する

   INVENTORY.TORONTO チャネルに未確定メッセージがないことを確認するには、INVENTORY.TORONTO と呼ばれるクラスター送信側チャネルを他のキュー・マネージャーのそれぞれについてモニターします。 次のように、各キュー・マネージャーから INDOUBT パラメーターを指定して DISPLAY CHSTATUS コマンドを発行します。

   ```
   DISPLAY CHSTATUS(INVENTORY.TORONTO) INDOUBT
   ```

   未確定メッセージがある場合は、必ずそれを解決してから作業を進めてください。 例えば、RESOLVE チャネル・コマンドの発行や、チャネルの停止および再起動などが考えられます。
6. ローカル・キューを削除する

   TORONTO の在庫管理アプリケーションに転送するメッセージがなくなったことを確認したら、次のようにキューを削除します。

   ```
   DELETE QLOCAL(INVENTQ)
   ```
7. これで、在庫管理アプリケーションをトロントのシステムから削除できます。

   アプリケーションを削除することによって、重複が発生しないようになり、システムのスペースが節約されます。

タスクの結果

この作業によって設定されるクラスターは、前の作業によって設定されたクラスターと類似しています。異なる点は、INVENTQ キューがキュー・マネージャー TORONTO で使用できなくなっていることです。

手順 1 でキューを処理対象から外すと、TORONTO キュー・マネージャーは 2 つの完全リポジトリー・キュー・マネージャーにメッセージを送信します。 キュー・マネージャーは、完全リポジトリー・キュー・マネージャーに状況の変化を通知します。 完全リポジトリー・キュー・マネージャーは、この情報をクラスター内の他のキュー・マネージャーのうち、INVENTQ に関する情報の更新を要求していたキュー・マネージャーに渡します。

キュー・マネージャーが INVENTQ キューにメッセージを書き込むと、更新された部分リポジトリーにより、INVENTQ キューが NEWYORK キュー・マネージャーでのみ使用可能であることが示されます。 メッセージは NEWYORK キュー・マネージャーに送信されます。

次のタスク

この作業では、削除するキューは 1 つだけであり、そのキューの削除元のクラスターも 1 つだけです。

クラスター名が多数登録されている名前リストを参照しているキューが多数あるとします。 例えば、キュー・マネージャー TORONTO は、INVENTQ 以外に PAYROLLQ、SALESQ、PURCHASESQ をホストすることもあります。 TORONTO は、これらのキューをそれぞれ該当するすべてのクラスター INVENTORY、PAYROLL、SALES、PURCHASES で使用可能にします。 TORONTO キュー・マネージャー上でクラスター名の名前リストを次のように定義します。

```
DEFINE NAMELIST(TOROLIST)
DESCR('List of clusters TORONTO is in')
NAMES(INVENTORY, PAYROLL, SALES, PURCHASES)
```

各キュー定義に名前リストを追加します。

```
DEFINE QLOCAL(INVENTQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PAYROLLQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(SALESQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PURCHASESQ) CLUSNL(TOROLIST)
```

ここで、SALES 演算は PURCHASES 演算に引き継がれるため、これらのすべてのキューを SALES クラスターから削除する必要があると仮定します。 これを実行するには、TOROLIST 名前リストを変更して、SALES クラスターの名前をこの名前リストから削除するだけで済みます。

名前リストに含まれるいずれかのクラスターから 1 つのキューを削除する場合は、それ以外のクラスター名のリストを登録した名前リストを作成します。 次に、この新しい名前リストを使用するようキュー定義を変更します。 PAYROLLQ を INVENTORY クラスターから削除するには、以下の手順を実行します。

1. 次のようにして、名前リストを作成します。

```
DEFINE NAMELIST(TOROSHORTLIST)
DESCR('List of clusters TORONTO is in other than INVENTORY')
NAMES(PAYROLL, SALES, PURCHASES)
```

2. 次のようにして、PAYROLLQ キュー定義を変更します。

```
ALTER QLOCAL(PAYROLLQ) CLUSNL(TOROSHORTLIST)
```

クラスターからのキュー・マネージャーの削除: ベスト・プラクティス

キュー・マネージャーがクラスター内の少なくとも 1 つの完全リポジトリーと正常に通信できるシナリオで、クラスターから 1 つのキュー・マネージャーを除去します。

始める前に

この方法は、少なくとも 1 つの完全リポジトリーが使用可能で、かつ、除去されるキュー・マネージャーから接続できるシナリオでのベスト・プラクティスです。 この方法では、手操作による介入を最小限に抑え、キュー・マネージャーがクラスターからの制御された離脱をネゴシエーションすることができます。 除去されるキュー・マネージャーが完全リポジトリーと接続できない場合、362 ページの『クラスターからのキュー・マネージャーの削除: 代替方式』を参照してください。

このタスクについて

以下のタスク例では、キュー・マネージャー LONDON を INVENTORY クラスターから除去します。 INVENTORY クラスターは 313 ページの『クラスターにキュー・マネージャーを追加する』の説明に従ってセットアップされ、358 ページの『キュー・マネージャーからのクラスター・キューの削除』の説明に従って変更されます。

キュー・マネージャーをクラスターから除去する処理は、キュー・マネージャーを追加する処理よりも複雑です。

キュー・マネージャーがクラスターに加わるとき、クラスターの既存メンバーには新規キュー・マネージャーについての情報はなく、したがって、そのことについての相互作用はありません。 加えられるキュー・マネージャーに送信側チャネルと受信側チャネルを新規に作成し、そのキュー・マネージャーが完全リポジトリーに接続できるようにする必要があります。

キュー・マネージャーがクラスターから削除されるときに、そのキュー・マネージャーに接続しているアプリケーションが、クラスター内のどこかでホストされているキューなどのオブジェクトを使用している可能性があります。 また、クラスター内の他のキュー・マネージャーに接続しているアプリケーションが、ターゲット・キュー・マネージャーでホストされているオブジェクトを使用している場合もあります。 このようなアプリケーションがある結果、現行のキュー・マネージャーは、クラスターに加わっていたフル・リポジトリー以外のクラスター・メンバーと通信を確立するために、追加の送信側チャネルを作成する場合があります。 そのクラスター内のすべてのキュー・マネージャーには、他のクラスター・メンバーについて記述するデータのキャッシュ・コピーがあります。 これには、除去されるメンバーが含まれる可能性があります。

手順

1. キュー・マネージャーをクラスターから除去する前に、クラスターで必要なリソースをキュー・マネージャーがホストしなくなったことを確認します。
   - キュー・マネージャーが完全リポジトリーをホストする場合、325 ページの『完全リポジトリーの別のキュー・マネージャーへの移動』の 1 から 6 の手順を実行します。 除去するキュー・マネージャーの完全リポジトリー機能が別のキュー・マネージャーに移動しない場合は、5 と 6 の手順のみを実行します。

- キュー・マネージャーがクラスター・キューをホストする場合、358 ページの『キュー・マネージャーからのクラスター・キューの削除』の 1 から 7 の手順を実行します。
- キュー・マネージャーがクラスター・トピックをホストする場合、トピックを削除するか (例えば、DELETE TOPIC コマンドを使用する)、または 440 ページの『別のキュー・マネージャーへのクラスター・トピック定義の移動』の説明に従って他のホストに移動します。

  注 : キュー・マネージャーをクラスターから除去する際にキュー・マネージャーがまだクラスター・トピックをホストしている場合、 そのキュー・マネージャーは、トピックが削除されるまで、クラスターに残っているキュー・マネージャーにパブリケーションの送信を引き続き試行する可能性があります。

2. キュー・マネージャー LONDON で、手動で定義したクラスター受信側チャネルを変更して、クラスターから削除します。

   ```
   ALTER CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) CLUSTER(' ')
   ```

3. キュー・マネージャー LONDON で、手動で定義したクラスター送信側チャネルを変更して、クラスターから削除します。

   ```
   ALTER CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) CLUSTER(' ')
   ```

   クラスター内の他のキュー・マネージャーは、このキュー・マネージャーとそのクラスター・リソースが、そのクラスターの一部でなくなったことを知ります。
4. クラスター内で完全リポジトリーへのフローを待機しているメッセージがなくなるまで、キュー・マネージャー LONDON のクラスター伝送キューをモニターします。

   ```
   DISPLAY CHSTATUS(INVENTORY.PARIS) XQMSGSA
   ```

   メッセージが送信キューに残っている場合は、先へ進む前に、それらのメッセージが完全リポジトリー PARIS と NEWYORK に送られていない理由を特定する必要があります。

タスクの結果

キュー・マネージャー LONDON は、クラスターの一部ではなくなりました。 しかし、依然として独立したキュー・マネージャーとして機能することができます。

次のタスク

クラスターの残りのメンバー上で以下のコマンドを発行して、これらの変更の結果を確認できます。

```
DISPLAY CLUSQMGR(LONDON)
```

キュー・マネージャーは、その自動定義クラスター送信側チャネルが停止するまで表示され続けます。 停止するまで待つこともできますし、次のコマンドを発行して、アクティブ・インスタンスがないかどうかモニターすることもできます。

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

このキュー・マネージャーにこれ以上メッセージが送られないことが確実な場合には、クラスターの残りのメンバーに対して次のコマンドを発行して、LONDON へのクラスター送信側チャネルを停止することもできます。

```
STOP CHANNEL(INVENTORY.LONDON) STATUS(INACTIVE)
```

変更がクラスター全体に伝搬して、このキュー・マネージャーに送達されるメッセージがなくなった後、LONDON の CLUSRCVR チャネルを停止して削除します。

```
STOP CHANNEL(INVENTORY.LONDON)
DELETE CHANNEL(INVENTORY.LONDON)
```

手動で定義された伝送キューがこのチャネルに使用されていて、CLCHNAME パターンが他の既存のチャネルまたは計画されたチャネルと一致しない場合は、伝送キューを削除することができます。 以下に例を示します。

```
DELETE QLOCAL(PARIS.CUSTOM.XMITQ)
```

注 : 自動定義伝送キューまたは共用 SYSTEM.CLUSTER.TRANSMIT.QUEUE は使用中です。このステップは必要ありません。

除去されたキュー・マネージャーは、313 ページの『クラスターにキュー・マネージャーを追加する』の説明に従って、後でクラスターに再び追加することができます。 除去されたキュー・マネージャーは、クラスターの残りのメンバーの情報を引き続き最大 90 日間キャッシュに入れます。 このキャッシュの有効期限が切れるまでは待たないことにした場合、364 ページの『キュー・マネージャーのクラスター化以前の状態への復元』の説明に従って強制的に除去できます。

関連タスク

クラスターからのキュー・マネージャーの除去 (IBM MQ Explorer を使用)

関連資料

ALTER CHANNEL (チャネル設定の変更)

DISPLAY CHANNEL (チャネル定義の表示)

DISPLAY CHSTATUS (チャネル状況の表示)

DISPLAY CLUSQMGR (クラスター・キュー・マネージャーのチャネル情報の表示)

STOP CHANNEL (チャネルの停止)

クラスターからのキュー・マネージャーの削除: 代替方式

重大なシステムまたは構成上の問題によりキュー・マネージャーがクラスター内のどの完全リポジトリーとも通信できないシナリオで、クラスターから 1 つのキュー・マネージャーを除去します。

始める前に

クラスターからキュー・マネージャーを除去するために使用するこの代替方式により、除去されるキュー・マネージャーをクラスターにリンクしているすべてのクラスター・チャネルを手動で停止して削除し、キュー・マネージャーをクラスターから強制的に除去します。 この方法は、除去されるキュー・マネージャーがどの完全リポジトリーとも通信できないシナリオで使用されます。 このシナリオの原因の例として、キュー・マネージャーが動作を停止していることや、キュー・マネージャーとクラスターの間で発生した通信障害が長引いていることなどが考えられます。 そうでなければ、最も一般的な方法 (360 ページの『クラスターからのキュー・マネージャーの削除: ベスト・プラクティス』) を使用してください。

このタスクについて

以下のタスク例では、キュー・マネージャー LONDON を INVENTORY クラスターから除去します。 INVENTORY クラスターは 313 ページの『クラスターにキュー・マネージャーを追加する』の説明に従ってセットアップされ、358 ページの『キュー・マネージャーからのクラスター・キューの削除』の説明に従って変更されます。

キュー・マネージャーをクラスターから除去する処理は、キュー・マネージャーを追加する処理よりも複雑です。

キュー・マネージャーがクラスターに加わるとき、クラスターの既存メンバーには新規キュー・マネージャーについての情報はなく、したがって、そのことについての相互作用はありません。 加えられるキュー・マネージャーに送信側チャネルと受信側チャネルを新規に作成し、そのキュー・マネージャーが完全リポジトリーに接続できるようにする必要があります。

キュー・マネージャーがクラスターから削除されるときに、そのキュー・マネージャーに接続しているアプリケーションが、クラスター内のどこかでホストされているキューなどのオブジェクトを使用している可能性があります。 また、クラスター内の他のキュー・マネージャーに接続しているアプリケーションが、ターゲット・キュー・マネージャーでホストされているオブジェクトを使用している場合もあります。 このようなアプリケーションがある結果、現行のキュー・マネージャーは、クラスターに加わっていたフル・

リポジトリー以外のクラスター・メンバーと通信を確立するために、追加の送信側チャネルを作成する場合があります。 そのクラスター内のすべてのキュー・マネージャーには、他のクラスター・メンバーについて記述するデータのキャッシュ・コピーがあります。 これには、除去されるメンバーが含まれる可能性があります。

この手順が適していると考えられるのは緊急時です。 緊急時にはキュー・マネージャーがクラスターから正規の手順で除外されるのを待つことはできません。

手順

1. キュー・マネージャーをクラスターから除去する前に、クラスターで必要なリソースをキュー・マネージャーがホストしなくなったことを確認します。
   - キュー・マネージャーが完全リポジトリーをホストする場合、325 ページの『完全リポジトリーの別のキュー・マネージャーへの移動』の 1 から 6 の手順を実行します。 除去するキュー・マネージャーの完全リポジトリー機能が別のキュー・マネージャーに移動しない場合は、5 と 6 の手順のみを実行します。
   - キュー・マネージャーがクラスター・キューをホストする場合、358 ページの『キュー・マネージャーからのクラスター・キューの削除』の 1 から 7 の手順を実行します。
   - キュー・マネージャーがクラスター・トピックをホストする場合、トピックを削除するか (例えば、DELETE TOPIC コマンドを使用する)、または 440 ページの『別のキュー・マネージャーへのクラスター・トピック定義の移動』の説明に従って他のホストに移動します。

     注 : キュー・マネージャーをクラスターから除去する際にキュー・マネージャーがまだクラスター・トピックをホストしている場合、 そのキュー・マネージャーは、トピックが削除されるまで、クラスターに残っているキュー・マネージャーにパブリケーションの送信を引き続き試行する可能性があります。
2. クラスター内の他のキュー・マネージャーと通信するために使用されているすべてのチャネルを停止します。 キュー・マネージャー LONDON 上の CLUSRCVR チャネルを停止するには、MODE(FORCE) を使用します。 そうでない場合は、送信側キュー・マネージャーがチャネルを停止するまで待たなければならないことがあります。

   ```
   STOP CHANNEL(INVENTORY.LONDON) MODE(FORCE)
   STOP CHANNEL(INVENTORY.TORONTO)
   STOP CHANNEL(INVENTORY.PARIS)
   STOP CHANNEL(INVENTORY.NEWYORK)
   ```
3. キュー・マネージャー LONDON 上で、チャネルが停止するまでチャネルの状態をモニターします。

   ```
   DISPLAY CHSTATUS(INVENTORY.LONDON)
   DISPLAY CHSTATUS(INVENTORY.TORONTO)
   DISPLAY CHSTATUS(INVENTORY.PARIS)
   DISPLAY CHSTATUS(INVENTORY.NEWYORK)
   ```

   チャネルが停止した後は、クラスター内の他のキュー・マネージャーがそれ以上アプリケーション・メッセージを送受信することはなくなります。
4. キュー・マネージャー LONDON で、手動で定義したクラスター・チャネルを削除します。

   ```
   DELETE CHANNEL(INVENTORY.NEWYORK)
   DELETE CHANNEL(INVENTORY.TORONTO)
   ```
5. クラスター内の残りのキュー・マネージャーは、削除されるキュー・マネージャーの情報を保持し、引き続きメッセージをそこに送信する場合もあります。 残りのキュー・マネージャーからこの情報を消去するには、いずれかの完全リポジトリー上のクラスターから、除去されるキュー・マネージャーを以下のようにリセットします。

   ```
   RESET CLUSTER(INVENTORY) ACTION(FORCEREMOVE) QMNAME(LONDON) QUEUES(YES)
   ```

   削除されるキュー・マネージャーと同じ名前の別のキュー・マネージャーがクラスター内に存在する可能性がある場合、削除されるキュー・マネージャーの **QMID** を指定してください。

タスクの結果

キュー・マネージャー LONDON は、クラスターの一部ではなくなりました。 しかし、依然として独立したキュー・マネージャーとして機能することができます。

次のタスク

クラスターの残りのメンバー上で以下のコマンドを発行して、これらの変更の結果を確認できます。

```
DISPLAY CLUSQMGR(LONDON)
```

キュー・マネージャーは、その自動定義クラスター送信側チャネルが停止するまで表示され続けます。 停止するまで待つこともできますし、次のコマンドを発行して、アクティブ・インスタンスがないかどうかモニターすることもできます。

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

変更がクラスター全体に伝搬して、このキュー・マネージャーに送達されるメッセージがなくなった後、LONDON の CLUSRCVR チャネルを削除します。

```
DELETE CHANNEL(INVENTORY.LONDON)
```

除去されたキュー・マネージャーは、313 ページの『クラスターにキュー・マネージャーを追加する』の説明に従って、後でクラスターに再び追加することができます。 除去されたキュー・マネージャーは、クラスターの残りのメンバーの情報を引き続き最大 90 日間キャッシュに入れます。 このキャッシュの有効期限が切れるまでは待たないことにした場合、364 ページの『キュー・マネージャーのクラスター化以前の状態への復元』の説明に従って強制的に除去できます。

関連資料

DELETE CHANNEL (チャネルの削除)
DISPLAY CHANNEL (チャネル定義の表示)
DISPLAY CHSTATUS (チャネル状況の表示)
DISPLAY CLUSQMGR (クラスター・キュー・マネージャーのチャネル情報の表示)
STOP CHANNEL (チャネルの停止)
RESET CLUSTER (クラスターのリセット)

キュー・マネージャーのクラスター化以前の状態への復元

キュー・マネージャーには、クラスターから除去されても、残りのクラスター・メンバーに関する情報が保持されます。 最終的にはこの情報は有効期限切れになり、自動的に削除されます。 しかし、この情報をすぐに削除する場合は、このトピックの手順で行うことができます。

始める前に

対象となるキュー・マネージャーがクラスターから除去されており、クラスター内で作業を実行しなくなったことを前提としています。 例えば、そのキューがクラスターからメッセージを受け取らなくなり、それらのキューでメッセージが届くのを待っているアプリケーションもない状態です。

このタスクについて

キュー・マネージャーには、クラスターから除去されても、残りのクラスター・メンバーに関する情報が最大 90 日間保持されます。 これにはシステム上の利点がある場合があり、特に、キュー・マネージャーがすぐにクラスターに再び加入する場合にはそう言えます。 この情報は、最終的に有効期限切れになると、自動的に削除されます。 しかし、この情報の手動での削除を選択する理由もあります。 以下に例を示します。

- 以前クラスター・リソースを使用していたこのキュー・マネージャー上のすべてのアプリケーションを停止したことを確認することができます。 残りのクラスター・メンバーの情報の有効期限が切れるまで、

そのようなアプリケーションは伝送キューに書き込みを続けます。 クラスターの情報が削除された後、そのようなアプリケーションがクラスター・リソースの使用を試みると、システムでエラー・メッセージが生成されます。

- キュー・マネージャーの状況情報を表示する際に、残りのクラスター・メンバーに関する有効期限情報を表示しない方がよい場合もあります。

このタスクでは、例として INVENTORY クラスターを使用します。 LONDON キュー・マネージャーは、360 ページの『クラスターからのキュー・マネージャーの削除: ベスト・プラクティス』で説明されているように、INVENTORY クラスターから除去されています。 クラスターの残りのメンバーの情報を削除するには、LONDON キュー・マネージャーで以下のコマンドを発行します。

手順

1. このキュー・マネージャーから、クラスター内の他のキュー・マネージャーのメモリーをすべて除去します。

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

2. すべてのクラスター・リソースが除去されるまで、キュー・マネージャーをモニターします。

```
DISPLAY CLUSQMGR(*) CLUSTER(INVENTORY)
DISPLAY QCLUSTER(*) CLUSTER(INVENTORY)
DISPLAY TOPIC(*) CLUSTER(INVENTORY)
```

関連概念

クラスター

クラスターのコンポーネント

関連資料

クラスター化と分散キューイングとの比較

キュー・マネージャーの保守

保守を実行するために、キュー・マネージャーをクラスターから中断および再開します。

このタスクについて

クラスターの一部であるキュー・マネージャーの保守は、毎日のように行う必要があります。 例えば、キューの中のデータのバックアップや、ソフトウェアに対する修正の適用が必要になります。 キュー・マネージャーがいずれかのキューをホストしている場合は、その活動を中断する必要があります。 保守が完了したら、キュー・マネージャーの活動を再開できます。

手順

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

```
SUSPEND QMGR CLUSTER(SALES)
```

SUSPEND **runmqsc** コマンドは、このキュー・マネージャーが中断されたことを SALES クラスター内のキュー・マネージャーに通知します。

SUSPEND QMGR コマンドの目的は、このキュー・マネージャーへのメッセージの送信をできれば避けるように、他のキュー・マネージャーにアドバイスすることだけです。 キュー・マネージャーが使用禁止になるわけではありません。 このキュー・マネージャーが処理する必要があるメッセージの中には、引き続きキュー・マネージャーに送信されるものもあります。例えば、このキュー・マネージャーがクラスター・キューの唯一のホストである場合などです。

キュー・マネージャーが中断している間、ワークロード管理ルーチンは、メッセージをこのキュー・マネージャーに送信することを避けます。 このキュー・マネージャーが処理する必要があるメッセージには、ローカル・キュー・マネージャーが送信するメッセージがあります。

IBM MQ は、可能な場合は常にローカル・キュー・マネージャーを選択するのではなく、ワークロード・バランシング・アルゴリズムを使用して、どの宛先が適しているかを判別します。

a) SUSPEND QMGR コマンドで FORCE オプションを使用して、以下のようにキュー・マネージャーの中断を強制します。

```
SUSPEND QMGR CLUSTER(SALES) MODE(FORCE)
```

MODE(FORCE) は、クラスター内の他のキュー・マネージャーからのインバウンド・チャネルを強制的に停止します。 MODE(FORCE) を指定しない場合は、デフォルトの MODE(QUIESCE) が適用されます。

2. 必要な保守タスクがあれば、すべて実行します。
3. 以下のように RESUME QMGR **runmqsc** コマンドを発行して、キュー・マネージャーを再開します。

```
RESUME QMGR CLUSTER(SALES)
```

タスクの結果

RESUME **runmqsc** コマンドは、キュー・マネージャーが再び使用可能になったことを完全リポジトリーに通知します。 完全リポジトリー・キュー・マネージャーは、このキュー・マネージャーに関係する情報の更新を要求していた他のキュー・マネージャーに、この通知情報を配布します。

クラスター伝送キューの保守

クラスター伝送キューを使用可能な状態に保つためにあらゆる作業を行います。 これらは、クラスターのパフォーマンスには不可欠です。 z/OS z/OS では、クラスター伝送キューの INDXTYPE を CORRELID に設定します。

始める前に

- クラスター伝送キューが満杯にならないようにしてください。
- ALTER **runmqsc** コマンドを実行して、誤って GET DISABLED または PUT DISABLED に設定されることがないように注意してください。
- クラスター伝送キューを保管するメディア z/OS (例えば、z/OS ページ・セット) が満杯にならないようにしてください。

このタスクについて

z/OS

以下の手順は、z/OS にのみ適用されます。

手順

クラスター伝送キューの INDXTYPE を CORRELID に設定します。

クラスター・キュー・マネージャーのリフレッシュ

REFRESH CLUSTER コマンドを使用して、ローカル・リポジトリーから自動定義チャネルおよび自動定義クラスター・オブジェクトを除去できます。 メッセージは失われません。

始める前に

IBM サポートがこのコマンドの使用を要求する場合があります。 このコマンドを安易に使用しないでください。 例えば、大規模なクラスターの場合、 **REFRESH CLUSTER** コマンドを使用すると、処理中のクラスターに悪影響が及ぶ可能性があります。その後、クラスター・オブジェクトが 27 日間隔で対象のキュー・マネージャーすべてに状況の更新を自動的に送信する際にも同様のことが起こり得ます。 クラスタリング: REFRESH CLUSTER の使用のベスト・プラクティスを参照してください。

このタスクについて

キュー・マネージャーはクラスター内でフレッシュ・スタートを行うことができます。 通常の環境では、REFRESH CLUSTER コマンドを使用する必要はありません。

手順

キュー・マネージャーから REFRESH CLUSTER **MQSC** コマンドを実行して、自動定義クラスター・キュー・マネージャーおよびキュー・オブジェクトをローカル・リポジトリーから除去します。

このコマンドにより、他のキュー・マネージャーを参照するオブジェクトのみが削除され、ローカル・キュー・マネージャーに関連したオブジェクトは除去されません。 このコマンドにより、自動定義チャネルも除去されます。 クラスター伝送キューにメッセージがないチャネルと、完全リポジトリー・キュー・マネージャーに接続されていないチャネルも、このコマンドによって除去されます。

タスクの結果

事実上、REFRESH CLUSTER コマンドにより、キュー・マネージャーはその完全リポジトリー内容に関してコールド・スタートが可能になります。 IBM MQ は、キューのデータの消失が起こらないことを保証しています。

関連情報

クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス

クラスター・キュー・マネージャーのリカバリー

REFRESH CLUSTER **runmqsc** コマンドを使用して、キュー・マネージャーに関するクラスター情報を最新の状態にします。 特定時点のバックアップからキュー・マネージャーをリカバリーした後で、以下の手順に従います。

始める前に

特定時点のバックアップから、クラスター・キュー・マネージャーがリストアされています。

このタスクについて

クラスター内のキュー・マネージャーをリカバリーするには、キュー・マネージャーをリストアし、REFRESH CLUSTER **runmqsc** コマンドを使用して、クラスター情報を最新の状態にします。

注 : 大規模クラスターでは、処理中のクラスターに **REFRESH CLUSTER** コマンドを使用すると、破壊的な影響を及ぼす恐れがあります。その後、クラスター・オブジェクトが 27 日間隔で対象のキュー・マネージャーすべてに状況の更新を自動的に送信する際にも同様のことが起こり得ます。 大規模クラスターでのリフレッシュはクラスターのパフォーマンスと可用性に影響を与える可能性があるを参照してください。

手順

キュー・マネージャーが参加しているすべてのクラスターについて、リストアされたキュー・マネージャーで REFRESH CLUSTER コマンドを実行します。

次のタスク

その他のキュー・マネージャーで REFRESH CLUSTER コマンドを実行する必要はありません。

関連概念

クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス

可用性のためのクラスター・チャネルの構成

適切な構成手順に従って、ネットワークの偶発的な停止が生じた場合に、クラスター・チャネルの稼働が問題なく続行されるようにします。

始める前に

クラスターを使用すると、チャネルを定義する必要から解放されますが、引き続き保守は必要です。 クラスター内のキュー・マネージャー間の通信には、分散キューイングで使用されているものと同じチャネルの技術が使用されています。 クラスター・チャネルを理解するには、次の事項を理解する必要があります。

- チャネルの動作方法
- チャネル状況の検索方法
- チャネル出口の使用方法

このタスクについて

特に以下の点について考慮してください。

手順

クラスター・チャネルを構成する際には、以下の点について検討してください。

- クラスター送信側チャネルおよびクラスター受信側チャネルの HBINT または KAINT の値には、多数のハートビートまたはキープアライブのフローによりネットワークに負荷がかからないようなものを選択します。 10 秒未満のインターバルを設定した場合に、ネットワークの速度が低下して設定した長さの遅延が発生すると、偽の障害が報告されます。
- 障害を起こしたチャネル上に未確定のメッセージがあると、それは孤立メッセージになりますが、BATCHHB 値を設定して、このようなメッセージの発生を許可する時間を短縮します。 バッチを埋めるのに長い時間がかかる場合、障害を起こしたチャネル上に未確定バッチが生じる可能性が高まります。 チャネル上のメッセージ・トラフィックが散発的でメッセージのバースト間隔が長い場合、バッチの障害の発生の可能性が高くなります。
- チャネルのクラスター送信側が障害を起こし、その後、ハートビートまたはキープアライブによってその障害が検出される前に再始動が試行された場合、問題が生じます。 チャネルのクラスター受信側がアクティブのままだと、チャネル送信側の再始動が拒否されます。 この障害を避けるために、クラスター送信側チャネルが再始動を試行するときにクラスター受信側チャネルの終了と再始動が行われるように調整してください。

  z/OS **IBM MQ for z/OS 上**
  ALTER QMGR の **ADOPTMCA** および **ADOPTCHK** パラメーターを使用して、アクティブなままのチャネルのクラスター受信側の側の問題を制御します。

  Multi **マルチプラットフォーム 上**
  qm.ini ファイルまたは Windows レジストリー内の **AdoptNewMCA** 属性、**AdoptNewMCATimeout** 属性、および **AdoptNewMCACheck** 属性を使用して、チャネルのクラスター受信側がアクティブのままである問題を制御します。

例

IBM MQ for z/OS と IBM MQ for Multiplatforms の両方でこれらの設定を実装する方法の例については、229 ページの『推奨設定』 を参照してください。

分散ネットワークに対する非同期コマンドが終了したことの確認

分散ネットワークでは、多くのコマンドが非同期コマンドとして使用されます。 コマンドによっては、またコマンド発行時のネットワークの状態によっては、完了までにかなりの時間がかかる場合があります。 キュー・マネージャーは完了時にメッセージを発行しないため、コマンドが終了したことを他の方法で確認する必要があります。

このタスクについて

クラスターに対して行う構成変更のほとんどすべてがたいてい非同期で完了します。 これは、クラスター内で発生する内部管理やサイクルの更新によるものです。 パブリッシュ/サブスクライブ階層では、サブス

クリプションに影響を与えるすべての構成変更がたいてい非同期で完了します。 このことはコマンドの名前からは分からないこともあります。

以下の MQSC コマンドはすべて非同期で完了します。 これらのコマンドにはそれぞれ PCF に相当するものがあり、そのほとんどは IBM MQ Explorer 内からも入手できます。 これらのコマンドは通常、ワークロードのない小規模なネットワークで実行される場合は、数秒で完了します。 ただし、これは大規模で負荷の高いネットワークには当てはまりません。 また、**REFRESH CLUSTER** コマンドは、特に複数のキュー・マネージャーで同時に発行される場合にはそれよりもかなり長い時間がかかることもあります。

これらのコマンドが終了したことを確認するには、予期するオブジェクトがリモート・キュー・マネージャーに存在することを確認します。

手順

- ALTER QMGR

  ALTER QMGR PARENT コマンドの場合、DISPLAY PUBSUB TYPE(PARENT) ALL を使用して、要求した親との関係の状況を追跡してください。

  ALTER QMGR REPOS コマンドおよび ALTER QMGR REPOSNL コマンドの場合、DISPLAY CLUSQMGR QMTYPE を使用して完了を確認してください。

- DEFINE CHANNEL、ALTER CHANNEL、および DELETE CHANNEL

  ALTER CHANNEL パラメーターの表にリストされているすべてのパラメーターについて、DISPLAY CLUSQMGR コマンドを使用して、クラスターにいつ変更が伝搬されたかをモニターしてください。

- DEFINE NAMELIST、ALTER NAMELIST、および DELETE NAMELIST。

  NAMELIST を **QMgr** オブジェクトの **CLUSNL** 属性で使用する場合、そのオブジェクトはキューまたはクラスター・チャネルの影響を受ける可能性があります。 必要に応じて、影響を受けるオブジェクトをモニターしてください。

  SYSTEM.QPUBSUB.QUEUE.NAMELIST に対する変更は、パブリッシュ/サブスクライブ階層でのプロキシー・サブスクリプションの作成または取り消しに影響を与える可能性があります。 DISPLAY SUB SUBTYPE(PROXY) コマンドを使用してこれをモニターしてください。

- DEFINE キュー、ALTER キュー、および DELETE キュー。

  DISPLAY QUEUE コマンドからの戻り値として可能なパラメーターの表にリストされているすべてのパラメーターについては、DISPLAY QCLUSTER コマンドを使用して、クラスターにいつ変更が伝搬されたかをモニターしてください。

- DEFINE SUB、および DELETE SUB

  最初のサブスクリプションをトピック・ストリングで定義する際、パブリッシュ/サブスクライブ階層またはパブリッシュ/サブスクライブ・クラスターにプロキシー・サブスクリプションを作成することもできます。 同様に、トピック・ストリング上の最後のサブスクリプションを削除する際、パブリッシュ/サブスクライブ階層またはパブリッシュ/サブスクライブ・クラスターでプロキシー・サブスクリプションを取り消すこともできます。

  サブスクリプションを定義または削除しているコマンドが終了したことを確認するには、予期するプロキシー・サブスクリプションが分散ネットワーク内の他のキュー・マネージャーに存在するかどうか確認します。 クラスターで直接ルーティング を使用する場合、クラスター内の他の部分リポジトリーに予期するプロキシー・サブスクリプションが存在することを確認してください。 クラスターでトピック・ホスト・ルーティング を使用する場合、一致するトピック・ホスト上に予期するプロキシー・サブスクリプションが存在することを確認してください。 以下の MQSC コマンドを使用します。

  ```
  DISPLAY SUB(*) SUBTYPE(PROXY)
  ```

  これに相当する以下のサブスクライブおよびアンサブスクライブ MQI 呼び出しがクラスターまたは階層で発行される場合には、これらの呼び出しに対して同じ確認を行ってください。

  - MQSUB を使用してサブスクライブします。
  - MQCO_REMOVE_SUB とともに MQCLOSE を使用してアンサブスクライブします。

- DEFINE TOPIC、ALTER TOPIC、および DELETE TOPIC

  クラスター化されたトピックを定義、変更、または削除するコマンドが終了したことを確認するには、クラスター内の他の部分リポジトリー内のトピック (直接ルーティング を使用している場合)、または他のトピック・ホスト上のトピック (トピック・ホスト・ルーティング を使用している場合) を表示します。

  DISPLAY TOPIC コマンドで表示できるパラメーターの表にリストされているすべてのパラメーターについては、DISPLAY TCLUSTER コマンドを使用して、クラスターにいつ変更が伝搬されたかをモニターしてください。

  注:

  - **CLUSTER** パラメーターは、パブリッシュ/サブスクライブ・クラスターでのプロキシー・サブスクリプションの作成または取り消しに影響を与える可能性があります。
  - **PROXYSUB** パラメーターおよび **SUBSCOPE** パラメーターは、パブリッシュ/サブスクライブ階層またはパブリッシュ/サブスクライブ・クラスターでのプロキシー・サブスクリプションの作成または取り消しに影響を与える可能性があります。
  - DISPLAY SUB SUBTYPE(PROXYSUB) コマンドを使用してこれをモニターしてください。

- REFRESH CLUSTER

  REFRESH CLUSTER コマンドを実行している場合、クラスターのコマンド・キュー項目数をポーリングしてください。 それがゼロになるまで待機し、ゼロの状態にしたまま、オブジェクトを検索してください。

  1. 以下の MQSC コマンドを使用して、クラスター・コマンド・キューのサイズがゼロであることを確認します。

     ```
     DISPLAY QL(SYSTEM.CLUSTER.COMMAND.QUEUE) CURDEPTH
     ```

  2. キュー項目数がゼロに達するまで確認を繰り返し、後続の確認でゼロを維持します。

  REFRESH CLUSTER コマンドは、オブジェクトを削除して再作成します。大きな構成の場合、完了までにかなりの時間がかかることがあります。 パブリッシュ/サブスクライブ・クラスターでの REFRESH CLUSTER についての考慮事項を参照してください。

- REFRESH QMGR タイプ (PROXYSUB)

  REFRESH QMGR TYPE(PROXYSUB) コマンドが終了したことを確認するには、分散ネットワーク内の他のキュー・マネージャー上でプロキシー・サブスクリプションが修正されていることを確認してください。 クラスターで直接ルーティング を使用する場合、クラスター内の他の部分リポジトリー上でプロキシー・サブスクリプションが修正されていることを確認してください。 クラスターでトピック・ホスト・ルーティング を使用する場合、一致するトピック・ホスト上で、予期するプロキシー・サブスクリプションが修正されていることを確認してください。 以下の MQSC コマンドを使用します。

  ```
  DISPLAY SUB(*) SUBTYPE(PROXYSUB)
  ```

- Reset Cluster

  RESET CLUSTER コマンドが完了したことを確認するには、DISPLAY CLUSQMGR を使用します。

- RESET QMGR タイプ (PUBSUB)

  RESET QMGR コマンドが完了したことを確認するには、DISPLAY PUBSUB TYPE(PARENT|CHILD) を使用します。

  注: RESET QMGR コマンドにより、パブリッシュ/サブスクライブ階層またはパブリッシュ/サブスクライブ・クラスターでプロキシー・サブスクリプションが取り消される可能性があります。 DISPLAY SUB SUBTYPE(PROXYSUB) コマンドを使用してこれをモニターしてください。

- コマンドの完了が近づくにつれて、また完了したときにキュー項目数がゼロになる傾向がある他のシステム・キューを、モニターすることもできます。

  例えば、SYSTEM.INTER.QMGR.CONTROL キューや SYSTEM.INTER.QMGR.FANREQ キューをモニターすることもできます。 クラスターにおけるプロキシー・サブスクリプションのトラフィックをモニター

するおよびパブリッシュ/サブスクライブ・ネットワークにおけるプロデューサーとコンシューマーのバランシングを参照してください。

次のタスク

これらの確認によって非同期コマンドが終了したことを確認できない場合、エラーが発生する可能性があります。 調査を行うには、コマンドが発行されたキュー・マネージャーのログをまず確認し、その後、(クラスターで) クラスターのフル・リポジトリー・ログを確認してください。

関連資料

z/OS z/OS における CLUSTER コマンドの非同期の動作

クラスターとの間のルーティング・メッセージ

キュー別名、キュー・マネージャー別名、およびリモート・キュー定義を使用して、クラスターを外部のキュー・マネージャーおよび他のクラスターに接続します。

クラスターとの間のメッセージのルーティングについて詳しくは、以下のサブトピックを参照してください。

関連概念

クラスター

クラスターのコンポーネント

384 ページの『キュー・マネージャー別名とクラスター』
クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

388 ページの『キュー別名とクラスター』
キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

387 ページの『応答先キュー別名およびクラスター』
応答先キューの別名定義は、応答情報の代替名を指定するために使用します。 応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

関連タスク

287 ページの『キュー・マネージャー・クラスターの構成』
クラスターによって提供されるキュー・マネージャーを相互接続するためのメカニズムにより、初期構成と継続的な管理を簡単に行えます。 クラスター・コンポーネントを定義し、クラスターを作成および管理できます。

301 ページの『新規クラスターのセットアップ』
クラスター例をセットアップするには、これらの手順に従ってください。 クラスターを設定する方法は、TCP/IP、LU 6.2、および単一の伝送キューまたは複数の伝送キューを使用する場合で別々に説明されています。 あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信することによって、クラスターの動作をテストします。

関連資料

クラスター化と分散キューイングとの比較

クラスターへの要求/応答の構成

クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。 クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

始める前に

372 ページの図 53 は、DEMO と呼ばれるクラスターの外部にある QM3 という名前のキュー・マネージャーを示します。 QM3 は、クラスターをサポートしない IBM MQ 製品のキュー・マネージャーの場合があります。 QM3 は、Q3 という名前のキューをホストします。このキューは、次のように定義されています。

```
DEFINE QLOCAL(Q3)
```

クラスターの内側には、QM1 および QM2 と呼ばれる 2 つのキュー・マネージャーがあります。 QM2 は、Q2 と呼ばれるクラスター・キューをホストしています。このキューは、次のように定義されています。

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO)
```

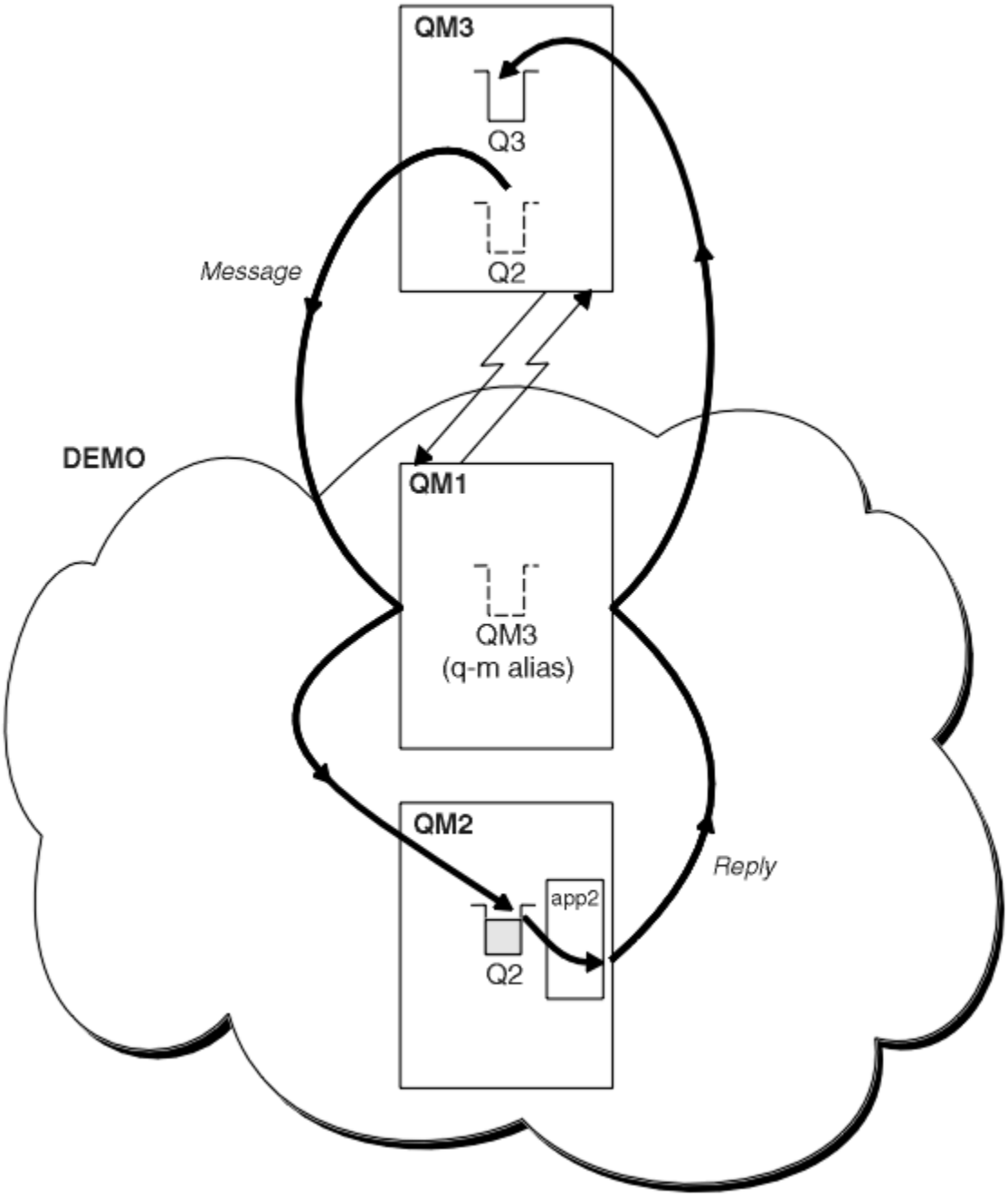


図 *53.* クラスターの外側にあるキュー・マネージャーからの書き込み

このタスクについて

以下の手順のアドバイスに従い、要求および応答メッセージのパスをセットアップしてください。

手順

1. クラスターに要求メッセージを送信する。

   クラスターの外部にあるキュー・マネージャーが、クラスターの内部にある QM2 のキュー Q2 にメッセージを書き込む方法について考えてみます。 クラスターの外部にあるキュー・マネージャーには、メッセージを書き込む対象となるクラスターのキューごとに QREMOTE 定義が必要です。

   a) QM3 に Q2 のリモート・キューを定義する。

   ```
   DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(QM2) XMITQ(QM1)
   ```

   QM3 はクラスターの一部ではないので、QM3 は、分散キューイング技術を使用して通信を行う必要があります。 したがって、QM3 には、送信側チャネルおよび QM1 への伝送キューも必要です。 QM1 には、これに対応する受信側チャネルが必要です。 このチャネルおよび伝送キューは、372 ページの図 53 の中では明示的に表示されていません。

   この例では、QM3 のアプリケーションが MQPUT 呼び出しを発行してメッセージを Q2 に書き込みます。 QREMOTE 定義により、そのメッセージは、QM1 伝送キューからメッセージを取得している送信側チャネルを使用して、QM2 の Q2 に転送されます。

2. クラスターから応答メッセージを受信する。

   キュー・マネージャーの別名を使用して、クラスターの外部にあるキュー・マネージャーへの応答の戻りパスを作成します。 ゲートウェイ QM1 は、クラスターの外部にあるキュー・マネージャー QM3 の別名をキュー・マネージャーに通知します。 QM3 のキュー・マネージャー別名定義にクラスター属性を追加することにより、クラスターの内側にあるキュー・マネージャーに QM3 を通知します。 キュー・マネージャーの別名定義は、RNAME がブランクである以外は、リモート・キュー定義と同じです。

   a) QM1 に QM3 のキュー・マネージャー別名を定義します。

   ```
   DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
   ```

   QM1 から QM3 に応答を転送するために使用する伝送キューの名前の選択について考慮する必要があります。 QREMOTE 定義で XMITQ 属性を省略することにより、伝送キューの名前は暗黙的に QM3 になります。 しかし、QM3 は、キュー・マネージャー別名を使用して、クラスター内の他のキュー・マネージャーに通知しようとしている名前と同じ名前です。 IBM MQ では、伝送キューとキュー・マネージャー別名の両方に、同じ名前を使用することはできません。 解決策の 1 つとして、QM3 へメッセージを転送するための伝送キューをキュー・マネージャー別名と異なる名前で作成します。

   b) 伝送キュー名を QREMOTE 定義で指定する。

   ```
   DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO) XMITQ(QM3.XMIT)
   ```

   新しいキュー・マネージャー別名により、QM3.XMIT という名前の新しい伝送キューと QM3 キュー・マネージャー別名が関連付けられます。 これは、簡単で正確な解決策ですが、まったく問題がないわけではありません。 これは、伝送キューがターゲット・キュー・マネージャーと同じ名前を持つという、 伝送キューの命名規則に反しています。 伝送キューの命名規則を維持する別の解決策はあるのでしょうか?

   リクエスターは、QM3 から送信される要求メッセージで、応答先キュー・マネージャー名としてデフォルトで QM3 を渡すため、問題が発生します。 QM2 のサーバーは、QM3 の応答先キュー・マネージャー名を使用して、応答で QM3 をアドレス指定します。 この解決策では、QM1 が、応答メッセージを返信するためのキュー・マネージャー別名として QM3 を通知する必要があるため、QM1 は QM3 を伝送キューの名前として使用できなくなります。

   キュー・マネージャー名への応答として QM3 を提供するデフォルトの代わりに、QM3 のアプリケーションは、QM1 に応答メッセージ用の応答先キュー・マネージャー別名を渡す必要があります。 ゲ

ートウェイ・キュー・マネージャー QM1 は、QM3 への応答に、QM3 そのものではなくキュー・マネージャー別名を通知して、伝送キューの名前と競合しないようにします。

c) QM1 に QM3 のキュー・マネージャー別名を定義します。

```
DEFINE QREMOTE(QM3.ALIAS) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

構成コマンドに 2 つ変更を加える必要があります。

i) QM1 の QREMOTE で、クラスターの他のキュー・マネージャーに対してキュー・マネージャー別名 QM3.ALIAS を通知し、この別名を実際のキュー・マネージャー QM3 の名前と関連付けます。 QM3 は、再び、QM3 への応答キューを返信するための伝送キューの名前になります。

ii) クライアント・アプリケーションで、要求メッセージを構成する際に、応答先キュー・マネージャーの名前として QM3.ALIAS を指定する必要があります。 以下の 2 つの方法のいずれかで、クライアント・アプリケーションに対して QM3.ALIAS を指定できます。

- MQMD 内の MQPUT によって構成されている応答先キュー・マネージャー名フィールドで、QM3.ALIAS をコーディングする。 応答に動的キューを使用している場合は、この方法で行わなければなりません。
- 応答先キュー名を指定する場合は、応答先キューではなく、応答先キュー別名の Q3.ALIAS を使用する。

```
DEFINE QREMOTE(Q3.ALIAS) RNAME(Q3) RQMNAME(QM3.ALIAS)
```

次のタスク

注 : 応答先キュー別名の使用を **AMQSREQ0** によって示すことはできません。 AMQSREQ0 は、パラメーター 3 に指定されているキュー名か、デフォルトの SYSTEM.SAMPLE.REPLY モデル・キューを使用して、応答先キューをオープンします。 MQPUT の応答先キュー・マネージャー別名を指定するためには、応答先キュー別名が含まれている別のパラメーターを指定してサンプルを変更する必要があります。

関連概念

キュー・マネージャー別名とクラスター
クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

応答先キュー別名およびクラスター
応答先キューの別名定義は、応答情報の代替名を指定するために使用します。 応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

キュー別名とクラスター
キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

関連タスク

クラスターからの要求/応答の構成
クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。 ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

クラスター外部からのワークロード・バランシングの構成
クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。 その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

クラスター間のメッセージ・パスの構成
ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターを一緒に接続します。 ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

375 ページの『クラスターのターゲット・キュー・マネージャーの名前の非公表』
キュー・マネージャー名を指定せずに、クラスター内のキュー・マネージャーで定義されているクラスター・キューにメッセージを転送します。

クラスターのターゲット・キュー・マネージャーの名前の非公表
キュー・マネージャー名を指定せずに、クラスター内のキュー・マネージャーで定義されているクラスター・キューにメッセージを転送します。

始める前に

- クラスター内部にあるキュー・マネージャーの名前が、クラスター外部にあるキュー・マネージャーに知られないようにします。
  - クラスター内部のキューをホストするキュー・マネージャーへの参照を解決すると、ワークロード・バランシングを行うための柔軟性がなくなります。
  - また、クラスター内のキューをホストするキュー・マネージャーを変更しにくくなります。
  - 代わりに、RQMNAME をクラスター管理者が指定したキュー・マネージャー別名に置き換えます。
  - 375 ページの『クラスターのターゲット・キュー・マネージャーの名前の非公表』では、キュー・マネージャー別名を使用して、クラスター外部のキュー・マネージャーをクラスター内部のキュー・マネージャーの管理から分離することについて説明されています。
- しかし、伝送キューの名前の付け方としては、ターゲット・キュー・マネージャーの名前を付けることが勧められています。 伝送キューの名前により、クラスター内のキュー・マネージャーの名前が分かります。 従うルールを選択する必要があります。 伝送キューの名前にキュー・マネージャー名を使用するか、それともクラスター名を使用するか選択できます。

  ゲートウェイ・キュー・マネージャー名を使用して伝送キューに名前を付ける
  クラスター外部のキュー・マネージャーにゲートウェイ・キュー・マネージャー名を開示することは、クラスター・キュー・マネージャー名を公表しないというルールの例外として妥当です。

  クラスターの名前を使用して伝送キューに名前を付ける
  ターゲット・キュー・マネージャーの名前を使用するという伝送キューの命名規則に従わない場合は、クラスター名を使用してください。

このタスクについて

371 ページの『クラスターへの要求/応答の構成』の作業を変更して、クラスター内部にあるターゲット・キュー・マネージャーの名前を非公開にします。

手順

例 (376 ページの図 54 を参照) で、ゲートウェイ・キュー・マネージャー QM1 に DEMO というキュー・マネージャー別名を定義します。

```
DEFINE QREMOTE(DEMO) RNAME(' ') RQMNAME(' ')
```

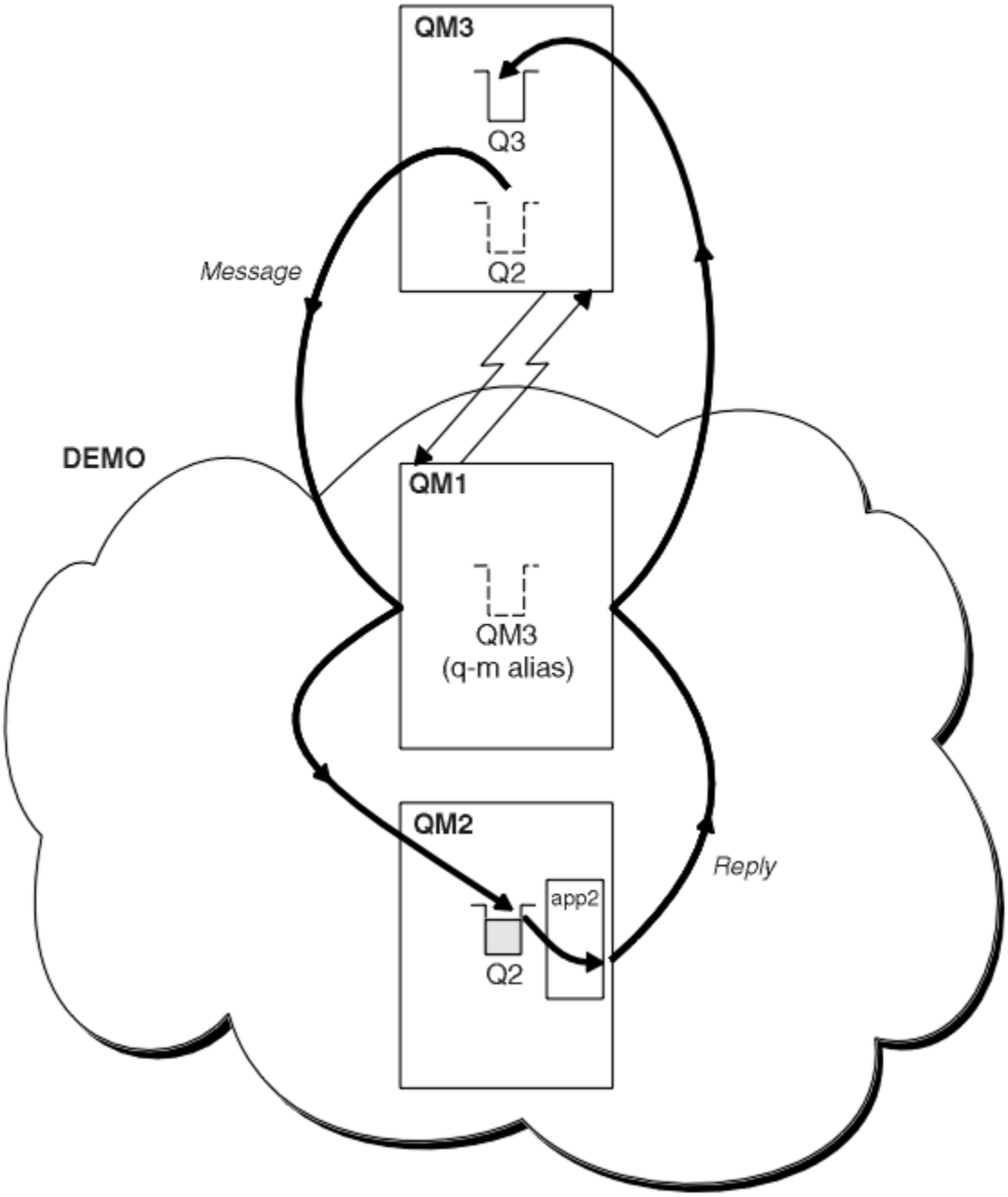


図 *54.* クラスターの外側にあるキュー・マネージャーからの書き込み

QM1 に対する 'QREMOTE 定義は、キューマネージャのエイリアス 'DEMO をゲートウェイのキューマネージャに知らしめる。 QM3 、 クラスタ外のキューマネージャは、実際のキューマネージャ名を使用するのではなく、キューマネージャエイリアス 'DEMO を使用して、'DEMO 上のクラスタキューにメッセージを送信することができます。

クラスターに接続する伝送キューの名前にクラスター名を使用する規則を採用する場合、Q2 のリモート・キュー定義は次のようになります。

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(DEMO) XMIT(DEMO)
```

タスクの結果

DEMO の Q2 宛のメッセージは、DEMO 伝送キューに入れられます。 伝送キューから送信側チャネルによりゲートウェイ・キュー・マネージャー QM1 に転送されます。 ゲートウェイ・キュー・マネージャーは、クラスター・キュー Q2 をホストする、クラスター内のいずれかのキュー・マネージャーにメッセージを転送します。

クラスターからの要求/応答の構成

クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。 ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

始める前に

378 ページの図 55 は、クラスター DEMO 内部のキュー・マネージャー QM2 を示します。 このキュー・マネージャーは、クラスター外部のキュー・マネージャーにホストされるキュー Q3 に要求を送信します。 応答は、クラスター内部にある QM2 の Q2 に返されます。

クラスター外部にあるキュー・マネージャーと通信するために、クラスター内部にある 1 つ以上のキュー・マネージャーがゲートウェイとして機能します。 ゲートウェイ・キュー・マネージャーには、クラスター外部にあるキュー・マネージャーへの通信パスがあります。 この例では、QM1 がゲートウェイです。

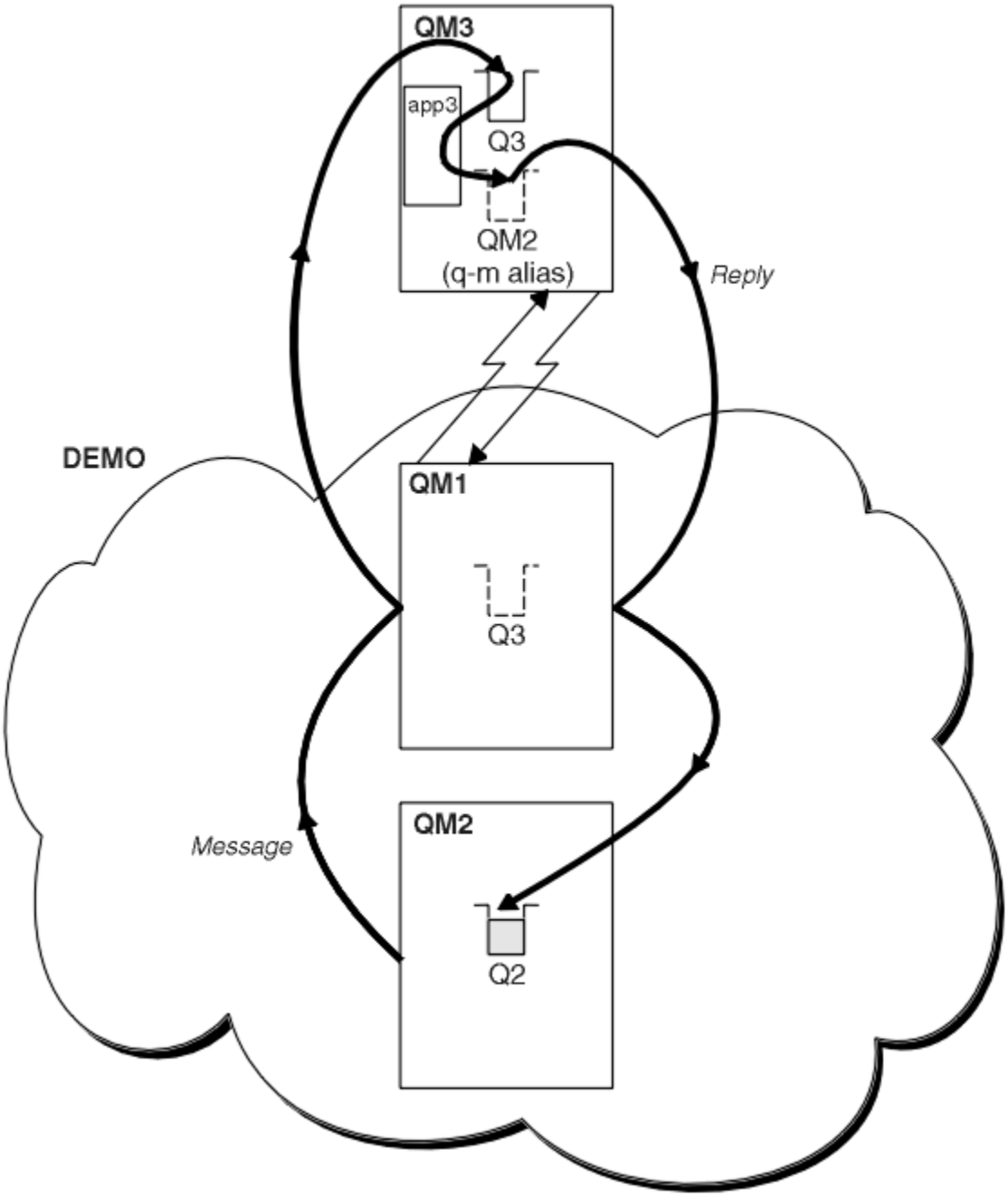


図 *55.* クラスターの外側にあるキュー・マネージャーへの書き込み

このタスクについて

以下の指示に従い、要求および応答メッセージのパスをセットアップしてください。

手順

1. クラスターから要求メッセージを送信する。

   クラスター内部にあるキュー・マネージャー QM2 が、クラスター外部にある QM3 のキュー Q3 にメッセージを書き込む方法について考えてみます。

   a) QM1 で QREMOTE 定義を作成して、リモート・キュー Q3 をクラスターに通知します。

   ```
   DEFINE QREMOTE(Q3) RNAME(Q3) RQMNAME(QM3) CLUSTER(DEMO)
   ```

このキュー・マネージャーには、クラスター外部にあるキュー・マネージャーへの送信側チャネルと伝送キューもあります。 QM3 には、これに対応する受信側チャネルがあります。 このチャネルは、378 ページの図 55 には示されていません。

QM2 のアプリケーションは、宛先キューと応答の送信先となるキューを指定して、MQPUT 呼び出しを発行します。 宛先キューは Q3 で、応答先キューは Q2 です。

メッセージは、QM1 に送信されます。QM1 では、そのリモート・キュー定義を使用して、QM3 の Q3 へのキュー名を解決します。

2. クラスター外部にあるキュー・マネージャーから応答メッセージを受信する。

   クラスター外部にあるキュー・マネージャーには、メッセージの送信先となるクラスター内のキュー・マネージャーごとにキュー・マネージャー別名が必要です。 キュー・マネージャーの別名には、ゲートウェイ・キュー・マネージャーへの伝送キューの名前を指定する必要もあります。 この例では、QM3 には QM2 に関するキュー・マネージャーの別名定義が必要です。

   a) QM3 でキュー・マネージャー別名 QM2 を作成します。

   ```
   DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) XMITQ(QM1)
   ```

   QM3 には QM1 への送信側チャネルおよび伝送キューも必要になります。また、QM1 にはこれに対応する受信側チャネルが必要になります。

   QM3 のアプリケーション **app3** は、次に、MQPUT 呼び出しを発行してキュー名 Q2 およびキュー・マネージャー名 QM2 を指定することによって、QM2 に応答できます。

次のタスク

クラスターからの経路を複数定義することができます。

関連概念

キュー・マネージャー別名とクラスター
クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

応答先キュー別名およびクラスター
応答先キューの別名定義は、応答情報の代替名を指定するために使用します。 応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

キュー別名とクラスター
キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

関連タスク

クラスターへの要求/応答の構成
クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。 クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

クラスター外部からのワークロード・バランシングの構成
クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。 その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

クラスター間のメッセージ・パスの構成
ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターを一緒に接続します。 ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

クラスター外部からのワークロード・バランシングの構成

クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。 その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

始める前に

371 ページの『クラスターへの要求/応答の構成』の 372 ページの図 53 に示されているように、例を構成します。

このタスクについて

このシナリオでは、 381 ページの図 56 のクラスター外部のキュー・マネージャー QM3 が、キュー Q2 に要求を送信します。 Q2 は、クラスター DEMO 内の 2 つのキュー・マネージャー QM2 および QM4 によってホストされます。 両方のキュー・マネージャーは、ワークロード・バランシングを使用するために、デフォルトのバインド・オプション NOTFIXED で構成されています。 クラスター外部にあるキュー・マネージャー QM3 からの要求は、QM1 を介して Q2 のいずれかのインスタンスに送信されます。

QM3 はクラスターの一部ではないので、分散キューイング技術を使用して通信します。 このキュー・マネージャーには、送信側チャネルと QM1 への伝送キューが必要です。 QM1 には、これに対応する受信側チャネルが必要です。 このチャネルおよび伝送キューは、381 ページの図 56 の中では明示的に表示されていません。

以下の手順は、371 ページの『クラスターへの要求/応答の構成』の 372 ページの図 53 を拡張しています。

手順

1. QM3 に Q2 の QREMOTE 定義を作成します。

   ```
   DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(Q3) XMITQ(QM1)
   ```

   QM3 がメッセージを書き込むクラスター内のキューごとに QREMOTE 定義を作成します。
2. QM1 でキュー・マネージャー別名 Q3 を作成します。

   ```
   DEFINE QREMOTE(Q3) RNAME(' ') RQMNAME(' ')
   ```

   Q3 は実際のキュー・マネージャー名ではありません。 クラスター内の、キュー・マネージャー別名 Q3 とブランク ' ' を同等にするキュー・マネージャー別名定義の名前です。
3. QM2 と QM4 のそれぞれで、Q2 というローカル・キューを定義します。

   ```
   DEFINE QLOCAL(Q2) CLUSTER(DEMO) DEFBIND(NOTFIXED)
   ```
4. ゲートウェイ・キュー・マネージャー QM1 には、特別な定義はありません。

タスクの結果

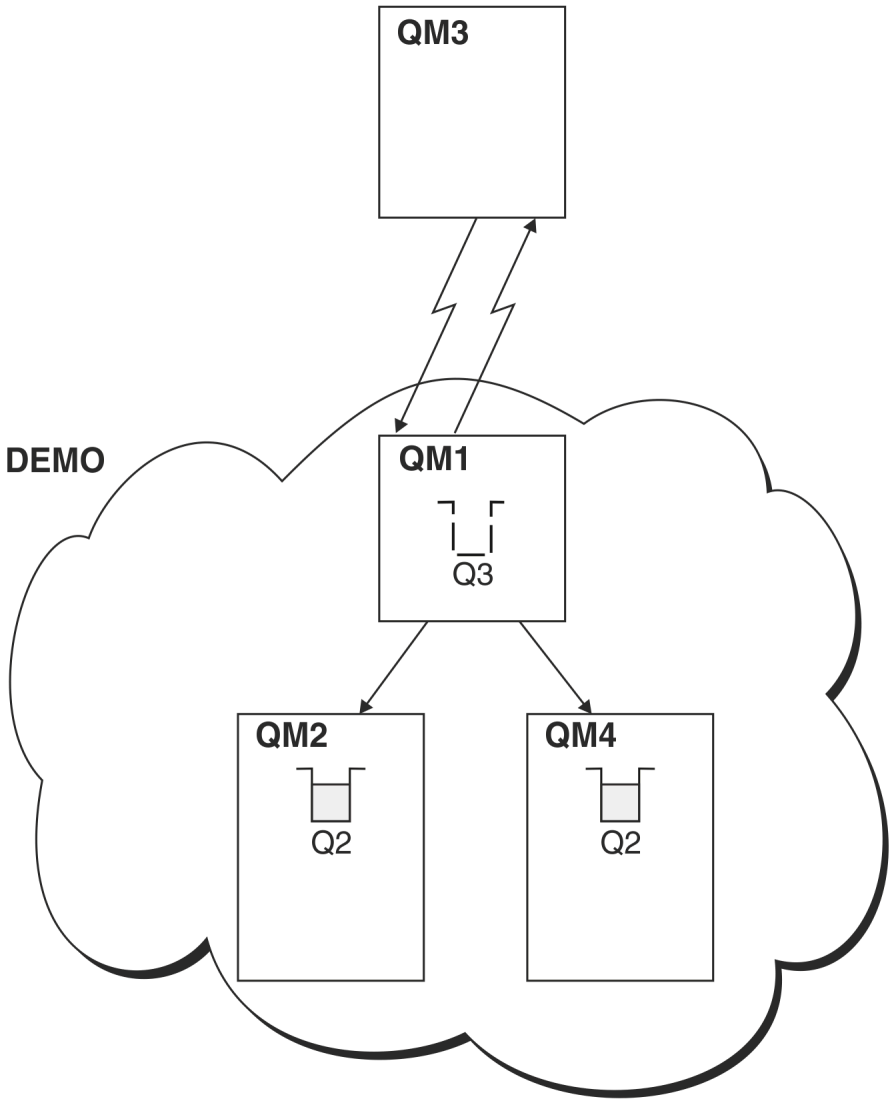


図 *56.* クラスターの外側にあるキュー・マネージャーからの書き込み

QM3 のアプリケーションが MQPUT 呼び出しを発行してメッセージを Q2 に書き込むとき、QM3 上には QREMOTE 定義があるため、このメッセージは、ゲートウェイ・キュー・マネージャー QM1 を経由してルーティングされます。 QM1 はメッセージを受け取ると、このメッセージが Q2 というキューを対象としていると認識し、ネーム・レゾリューションを実行します。 QM1 は、自身のローカル定義を調べますが、Q2 については検出されません。 次に、QM1 は自身のクラスター構成を調べ、クラスター DEMO 内の Q2 の 2 つのインスタンスを認識していることを突き止めます。 これで QM1 は、ワークロード・バランシングを使用して QM2 および QM4 に置かれている Q2 のインスタンス間でメッセージを分散できるようになりました。

関連概念

キュー・マネージャー別名とクラスター
クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

応答先キュー別名およびクラスター
応答先キューの別名定義は、応答情報の代替名を指定するために使用します。 応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

キュー別名とクラスター

キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

名前の解決

関連タスク

クラスターへの要求/応答の構成
クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。 クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

クラスターからの要求/応答の構成
クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。 ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

クラスター間のメッセージ・パスの構成
ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターを一緒に接続します。 ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

関連資料

キュー名の解決

クラスター間のメッセージ・パスの構成

ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターを一緒に接続します。 ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

このタスクについて

すべてのキュー・マネージャーを 1 つの大型クラスターにまとめる代わりに、多数の小型クラスターを使用できます。 この場合、各クラスター内の、1 つ以上のキュー・マネージャーをブリッジとして機能させます。 この長所は、複数のクラスター間でのキュー名およびキュー・マネージャー名の可視性を制限できることです。 クラスターのオーバーラップを参照してください。 別名を使用してキューの名前とキュー・マネージャーの名前を変更し、名前が競合しないようにしたり、ローカル命名規則に適合させたりすることができます。

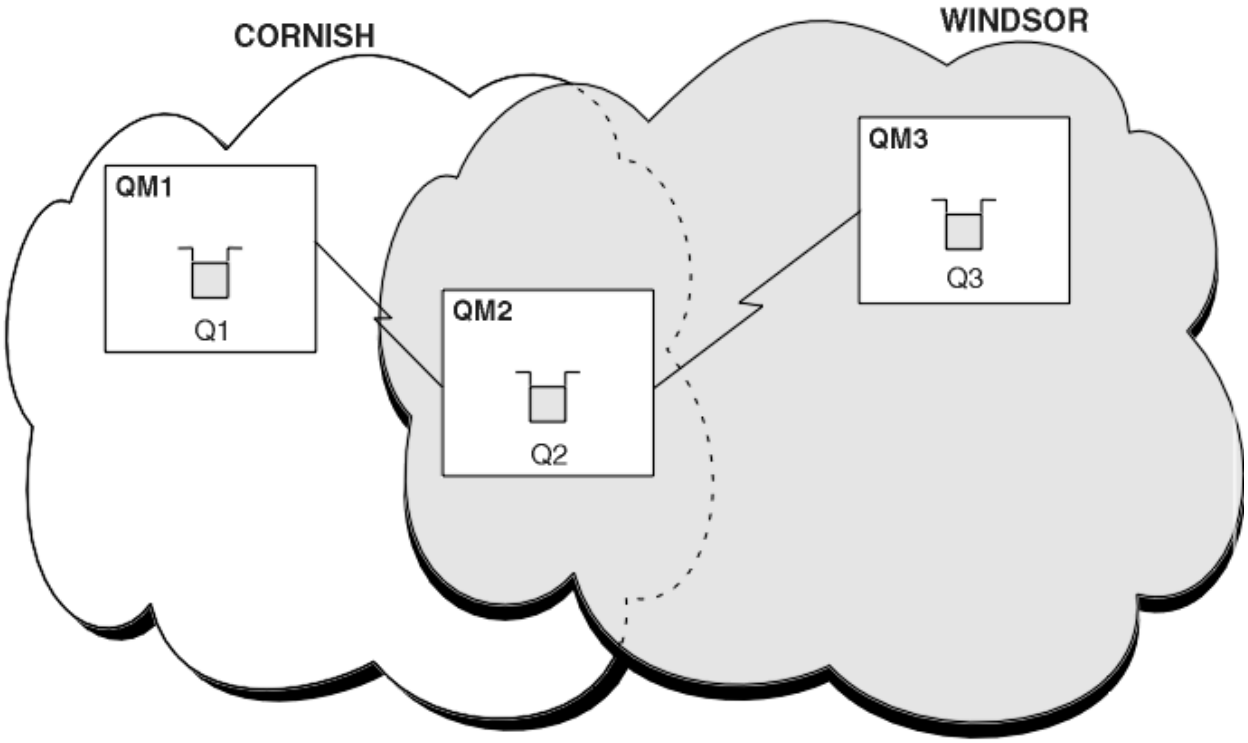


図 57. 複数のクラスターにわたるブリッジ化

383 ページの図 57 には、2 つのクラスターと両者の間をつなぐブリッジを示します。 ブリッジは、複数の場合もあります。

次の手順を使用して、クラスターを構成してください。

手順

1. QM1 にクラスター・キュー Q1 を定義する。

```
DEFINE QLOCAL(Q1) CLUSTER(CORNISH)
```

2. QM3 にクラスター・キュー Q3 を定義する。

```
DEFINE QLOCAL(Q3) CLUSTER(WINDSOR)
```

3. QM2 で、CORNISHWINDSOR と呼ばれる名前リストを作成する。これには両方のクラスターの名前が格納されます。

```
DEFINE NAMELIST(CORNISHWINDSOR) DESCR('CornishWindsor namelist')
NAMES(CORNISH, WINDSOR)
```

4. QM2 にクラスター・キュー Q2 を定義する

```
DEFINE QLOCAL(Q2) CLUSNL(CORNISHWINDSOR)
```

次のタスク

QM2 は、両方のクラスターのメンバーであり、両者の間のブリッジでもあります。 ブリッジを介して認識させる必要があるキューごとに、ブリッジの QALIAS 定義が必要になります。 383 ページの図 57 の例では、QM2 で次の定義が必要です。

```
DEFINE QALIAS(MYQ3) TARGET(Q3) CLUSTER(CORNISH) DEFBIND(NOTFIXED)
```

CORNISH 内のキュー・マネージャー (例えば QM1) に接続されたアプリケーションは、キュー別名を使用して、Q3 にメッセージを書き込むことができます。 Q3 は MYQ3 として参照されます。 メッセージは、QM3 の Q3 に転送されます。

キューをオープンするときに、DEFBIND を NOTFIXED か QDEF のいずれかに設定する必要があります。 DEFBIND がデフォルト (OPEN) のままの場合には、キュー・マネージャーは別名定義を、そのホストとなるブリッジ・キュー・マネージャーに解決します。 この場合、ブリッジはメッセージを転送しません。

認識させたいキュー・マネージャーごとに、キュー・マネージャーの別名定義が必要です。 例えば、QM2 で次の定義が必要です。

```
DEFINE QREMOTE(QM1) RNAME(' ') RQMNAME(QM1) CLUSTER(WINDSOR)
```

MQOPEN 呼び出しに QM1 の名前を明示的に指定することによって、WINDSOR 内のキュー・マネージャー (例えば QM3) に接続されたアプリケーションが QM1 のいずれかのキューにメッセージを書き込むことができます。

関連概念

キュー・マネージャー別名とクラスター
クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

応答先キュー別名およびクラスター
応答先キューの別名定義は、応答情報の代替名を指定するために使用します。 応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

キュー別名とクラスター
キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

関連タスク

クラスターへの要求/応答の構成
クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。 クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

クラスターからの要求/応答の構成
クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。 ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

クラスター外部からのワークロード・バランシングの構成
クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。 その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

キュー・マネージャー別名とクラスター

クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

キュー・マネージャーの別名は、ブランクの RNAME を持つリモート・キュー定義を使用して作成されますが、これには以下に示す 5 つの使用法があります。

メッセージ送信時のキュー・マネージャー名の再マップ

キュー・マネージャーの別名を使用すると、MQOPEN 呼び出しで指定されたキュー・マネージャー名を別のキュー・マネージャーに再マップすることができます。 クラスター・キュー・マネージャーも可能です。 例えば、キュー・マネージャーには、次のようなキュー・マネージャーの別名定義が存在する場合があります。

```
DEFINE QREMOTE(YORK) RNAME(' ') RQMNAME(CLUSQM)
```

CLUSQM というキュー・マネージャーの別名として YORK を使用できます。 これが定義されたキュー・マネージャーのアプリケーションがキュー・マネージャー YORK にメッセージを書き込むと、ローカル・キュー・マネージャーは名前を CLUSQM に解決します。 ローカル・キュー・マネージャーが CLUSQM という名前でない場合、メッセージはクラスター伝送キューに書き込まれて CLUSQM に転送されます。 伝送ヘッダーも変更され、YORK の代わりに CLUSQM が使用されます。

注 : 定義は、それを行うキュー・マネージャーにのみ適用されます。 クラスター全体に別名を通知するには、リモート・キュー定義に CLUSTER 属性を追加する必要があります。 次に、他のキュー・マネージャーからの YORK 宛だったメッセージが、CLUSQM に送信されます。

メッセージ送信時の伝送キューの変更または指定

別名割り当てを使用して、非クラスター・システムにクラスターを参加させることができます。 例えば、クラスター ITALY 内のキュー・マネージャーが、そのクラスター外の PALERMO というキュー・マネージャーと通信することができます。 通信するには、クラスター内のキュー・マネージャーの 1 つがゲートウェイの機能を果たす必要があります。 ゲートウェイ・キュー・マネージャーから、次のコマンドを発行します。

```
DEFINE QREMOTE(ROME) RNAME(' ') RQMNAME(PALERMO) XMITQ(X) CLUSTER(ITALY)
```

このコマンドは、キュー・マネージャー別名定義です。 ROME をキュー・マネージャーと定義して通知します。クラスター ITALY 内のキュー・マネージャーからのメッセージは、このキュー・マネージャーを介して、PALERMO の宛先にマルチホップして到達できます。 キュー・マネージャー名を ROME に設定して開かれたキューに書き込まれたメッセージは、キュー・マネージャー別名が定義されたゲートウェイ・キュー・マネージャーに送信されます。 到達したメッセージは伝送キュー X に書き込まれ、非クラスター・チャネルによってキュー・マネージャー PALERMO に転送されます。

この例で ROME という名前を選択したことは、重要ではありません。 QREMOTE と RQMNAME の値は、同じ場合があります。

メッセージ受信時の宛先指定

キュー・マネージャーはメッセージを受信すると、宛先キューとキュー・マネージャーの名前を伝送ヘッダーから抽出します。 伝送ヘッダー内のキュー・マネージャーと同じ名前のキュー・マネージャー別名定義を探します。 見つかると、伝送ヘッダー内のキュー・マネージャー名の代わりに、キュー・マネージャー別名定義での RQMNAME を使用します。

キュー・マネージャーの別名をこのように使用する理由は、次の 2 つです。

- メッセージを別のキュー・マネージャーに送信するため
- キュー・マネージャー名を変更して、ローカル・キュー・マネージャーと同じ名前にするため

ゲートウェイ・キュー・マネージャーでキュー・マネージャー別名を使用して、それぞれ異なるクラスター内にあるキュー・マネージャーの間でメッセージをルーティングする

アプリケーションは、キュー・マネージャー別名を使用して、別のクラスター内のキューにメッセージを送信できます。 このキューは、クラスター・キューである必要はありません。 キューは、1 つのクラスター内に定義されます。 アプリケーションは、別のクラスター内にあるキュー・マネージャーに接続されます。 ゲートウェイ・キュー・マネージャーが、この 2 つのクラスターを接続します。 キューがクラスター・キューとして定義されていない場合、正しいルーティングが行われるためには、アプリケーションがキュー名とクラスター化されたキュー・マネージャー別名を使用してキューをオープンする必要があります。 構成例については、336 ページの『ゲートウェイ・キュー・マネージャーを使用した 2 つのオーバーラップするクラスターの作成』を参照してください。図 1 に示されている応答メッセージ・フローは、その構成例から引用しています。

この図は、応答メッセージが (RQ と呼ばれる) 一時動的キュー に戻すためにたどるパスを示しています。 QM3 に接続されているサーバー・アプリケーションは、キュー・マネージャー名 QM2 を使用して応答キューを開きます。 キューマネージャ名 QM2 は、ゲートウェイキューマネージャ QM1 上のクラスタキューマネージャエイリアスとして定義されています。 QM3 返信メッセージをゲートウェイキューマネージャ にルーティングします。 QM1 QM1 は、メッセージを QM2 に経路指定します。

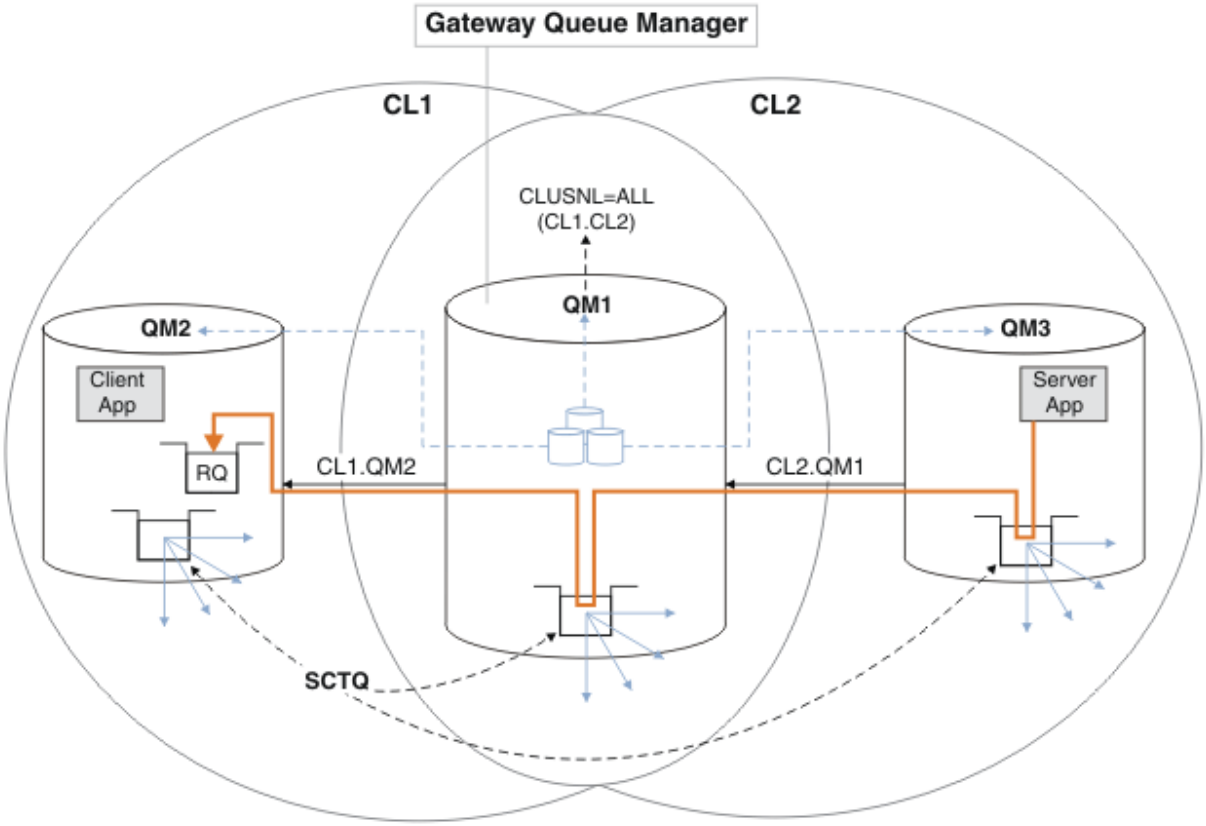


図 *58.* 応答メッセージを異なるクラスターに返すためのキュー・マネージャー別名の使用

このルーティングは、次のようにして行われます。 各クラスタのキュー・マネージャーには、ゲートウェイ・キュー・マネージャー QM1 上のキュー・マネージャー・エイリアス定義があります。 これらの別名は、すべてのクラスターでクラスター化されます。 それぞれの別名からキュー・マネージャーへと向かうグレーの破線矢印は、各キュー・マネージャー別名が、クラスターのうちの少なくとも 1 つにある実際のキュー・マネージャーに解決されることを示します。 この場合、QM2 別名はクラスター CL1 と CL2 の両方でクラスター化され、CL1 内の実際のキュー・マネージャー QM2 に解決されます。 サーバー・アプリケーションは、応答先キュー名 RQ、および応答先キュー・マネージャー名 QM2 を使用して応答メッセージを作成します。 メッセージはゲートウェイキューマネージャー QM1 にルーティングされます。キューマネージャーエイリアスの定義 QM2 がクラスター CL2 の QM1 で定義されている一方で、キューマネージャー QM2 はクラスター CL2 にないためです。 メッセージをターゲット・キュー・マネージャーに送信できないことから、この別名定義を持つキュー・マネージャーにメッセージが送信されます。

ゲートウェイキューマネージャー QM1 は、メッセージをクラスタ送信キューに配置し、 QM2 に転送します。 ゲートウェイキューマネージャー QM1 は、QM1 上の QM2 のキューマネージャーエイリアスの定義で、 QM2 が実際のターゲットキューマネージャーとして定義されているため、メッセージを QM2 に転送します。 別名定義が参照できるのは、実際の定義だけで、自身を参照することはできないため、この定義は循環しません。 実際の定義は、ゲートウェイキューマネージャー QM1 によって解決されます。 QM1 と QM2 は同じクラスターである CL1 にあるためです。 ゲートウェイキューマネージャー QM1 は、CL1 のリポジトリから QM2 の接続情報を取得し、メッセージを QM2 に転送します。 メッセージを QM1 によって転送するためには、サーバー・アプリケーションが、オプション DEFBIND を MQBND_BIND_NOT_FIXED に設定して応答キューをオープンしていなければなりません。 サーバー・アプリケーションがオプション MQBND_BIND_ON_OPEN を設定して応答キューをオープンしている場合、メッセージは転送されず、送達不能キューに入れられます。

キュー・マネージャーをクラスターへのゲートウェイとして使用して、クラスター外から来るメッセージのワークロード・バランシングを行う

クラスター内の複数のキュー・マネージャーに EDINBURGH というキューを定義します。 クラスター外からそのキューに来るメッセージのワークロードのバランスを取るためのクラスター化メカニズムが必要です。

クラスターの外側にあるキュー・マネージャーは、クラスター内の 1 つのキュー・マネージャーへの伝送キューと送信側チャネルを必要とします。 このキューをゲートウェイ・キュー・マネージャーといいます。 デフォルトの ワークロード・バランシング・メカニズムを利用するためには、以下のいずれかのルールが該当しなければなりません。

- ゲートウェイ・キュー・マネージャーに、EDINBURGH キューのインスタンスが含まれていないこと。
- ゲートウェイ・キュー・マネージャーが ALTER QMGR で CLWLUSEQ(ANY) を指定していること。

クラスター外に起因するワークロード・バランシングの例については、380 ページの『クラスター外部からのワークロード・バランシングの構成』を参照してください。

関連概念

応答先キュー別名およびクラスター
応答先キューの別名定義は、応答情報の代替名を指定するために使用します。 応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

キュー別名とクラスター
キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

関連タスク

クラスターへの要求/応答の構成
クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。 クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

クラスターからの要求/応答の構成
クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。 ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

クラスター外部からのワークロード・バランシングの構成
クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。 その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

クラスター間のメッセージ・パスの構成
ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターを一緒に接続します。 ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

応答先キュー別名およびクラスター

応答先キューの別名定義は、応答情報の代替名を指定するために使用します。 応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

以下に例を示します。

- キュー・マネージャー VENICE にあるアプリケーションは、MQPUT 呼び出しを使用して、メッセージをキュー・マネージャー PISA に送信します。 アプリケーションは、以下の応答先キュー情報をメッセージ記述子に提供します。

```
ReplyToQ='QUEUE'
ReplyToQMgr=''
```

- QUEUE に送信された応答を PISA の OTHERQ で受信できるようにするには、VENICE で応答先キューの別名として使用するリモート・キューの定義を作成します。 この別名が有効なのは、別名が作成されたシステム上だけです。

```
DEFINE QREMOTE(QUEUE) RNAME(OTHERQ) RQMNAME(PISA)
```

RQMNAME 自体がクラスター・キュー・マネージャーである場合でも、RQMNAME と QREMOTE に同じ名前を指定することができます。

関連概念

キュー・マネージャー別名とクラスター
クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

キュー別名とクラスター
キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

関連タスク

クラスターへの要求/応答の構成
クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。 クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

クラスターからの要求/応答の構成
クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。 ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

クラスター外部からのワークロード・バランシングの構成
クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。 その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

クラスター間のメッセージ・パスの構成
ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターを一緒に接続します。 ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

キュー別名とクラスター

キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

QALIAS 定義は、キューを認識するための別名を作成するために使用されます。 別名を作成する理由には、以下のようにさまざまなものがあります。

- 別のキューを使用してアプリケーションを始動するが、そのアプリケーションは変更しない場合。
- メッセージの書き込み先であるキューの実名をアプリケーションに認識させない場合。
- キューが定義された場所での命名規則とは異なる命名規則を適用している場合。
- 使用しているアプリケーションがキューに実名でアクセスすることは許可されず、別名によってのみアクセスが許可される場合。

キュー・マネージャーの QALIAS 定義は、DEFINE QALIAS コマンドを使用して作成します。 例えば、以下のコマンドを実行します。

```
DEFINE QALIAS(PUBLIC) TARGET(LOCAL) CLUSTER(C)
```

このコマンドは、PUBLIC というキューをクラスター C 内のキュー・マネージャーに通知します。 PUBLIC は、LOCAL というキューに解決される別名です。 PUBLIC に送信されたメッセージは、LOCAL と呼ばれるキューに送信されます。

キューの別名定義を使用して、キュー名をクラスター・キューに解決することもできます。 例えば、以下のコマンドを実行します。

```
DEFINE QALIAS(PRIVATE) TARGET(PUBLIC)
```

このコマンドによって、キュー・マネージャーは PRIVATE という名前を使用して、 PUBLIC という名前でクラスター内の他の場所に通知されたキューにアクセスすることができます。 この定義には CLUSTER 属性が設定されていないので、この定義はこの定義を作成したキュー・マネージャーだけに適用されます。

関連概念

キュー・マネージャー別名とクラスター
クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

応答先キュー別名およびクラスター
応答先キューの別名定義は、応答情報の代替名を指定するために使用します。 応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

関連タスク

クラスターへの要求/応答の構成
クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。 クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

クラスターからの要求/応答の構成
クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。 ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

クラスター外部からのワークロード・バランシングの構成
クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。 その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

クラスター間のメッセージ・パスの構成
ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターを一緒に接続します。 ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

クラスターによるワークロードの管理

クラスター内の異なるキュー・マネージャーにキューの複数インスタンスを定義することによって、キュー・サービス作業を複数のサーバーに分散させることができます。 障害が発生した場合にメッセージが別のキュー・マネージャーに再キューイングされるのを妨げる要因はいくつかあります。

システム管理を軽減するためにクラスターをセットアップすることに加えて、複数のキュー・マネージャーが同じキューの 1 つのインスタンスをホストするクラスターを作成することができます。

クラスター内の複数のキュー・マネージャーが互いにクローンになるように、クラスターを編成できます。 各キュー・マネージャーは同じアプリケーションを実行でき、同じキューのローカル定義を持つことができます。 z/OS 例えば、z/OS 並列シスプレックスでは、クローン・アプリケーションが共有 Db2 または仮想記憶アクセス方式 (VSAM) データベースのデータにアクセスできます。 アプリケーションのインスタンスをいくつか備えることによって、キュー・マネージャーの間でワークロードを分散させることができます。 アプリケーションの各インスタンスはメッセージを受信し、それぞれ独立して実行されます。

このようなクラスター編成には、次のような利点があります。

- キューおよびアプリケーションの可用性の向上。
- メッセージのスループットの高速化。
- ネットワーク内でのワークロード分布の均等化。

特定のキューのインスタンスをホストするどのキュー・マネージャーも、そのキューを宛先とするメッセージを処理できるので、アプリケーションはメッセージ送信時にキュー・マネージャーを指定しません。1 つのクラスターに同じキューのインスタンスが複数含まれる場合、IBM MQ はメッセージを送付するキュー・マネージャーを選択します。 最適な宛先は、キュー・マネージャーとキューの可用性とともに、キュー・マネージャー、キュー、 およびチャネルに関連付けられたいくつかのクラスター・ワークロード固有の属性に基づいて選択されます。 クラスターでのワークロード・バランシングを参照してください。

z/OS IBM MQ for z/OS では、キュー共有グループ内のキュー・マネージャーは共有キューとしてクラスター・キューをホストすることができます。 共有クラスター・キューは、同一キュー共有グループ内のすべてのキュー・マネージャーが使用可能です。 例えば、同じキューに複数のインスタンスがある場合のクラスターでは、キュー・マネージャー QM2 と QM4 のどちらかまたは両方を共有キュー・マネージャーにすることができます。 それぞれにキュー Q3 の定義があります。 QM4 と同じキュー共有グループ内のどのキュー・マネージャーも、共有キュー Q3 に書き込まれたメッセージを読み取ることができます。 キュー共有グループそれぞれに最大 32 のキュー・マネージャーを含めることができ、各キュー・マネージャーが同じデータにアクセスできます。 キュー共有は、メッセージのスループットを著しく高めます。

ワークロード管理のためのクラスター構成について詳しくは、以下のサブトピックを参照してください。

関連概念

クラスター化と分散キューイングとの比較

分散キューイングとクラスター

クラスターのコンポーネント

クラスター・チャネル

クラスター・キューが MQPUT に対して無効になったときの処置

クラスター送信側チャネルで設定されたワークロード・バランシングが機能しない

371 ページの『クラスターとの間のルーティング・メッセージ』
キュー別名、キュー・マネージャー別名、およびリモート・キュー定義を使用して、クラスターを外部のキュー・マネージャーおよび他のクラスターに接続します。

関連タスク

クラスター・ワークロード出口の作成とコンパイル

287 ページの『キュー・マネージャー・クラスターの構成』
クラスターによって提供されるキュー・マネージャーを相互接続するためのメカニズムにより、初期構成と継続的な管理を簡単に行えます。 クラスター・コンポーネントを定義し、クラスターを作成および管理できます。

301 ページの『新規クラスターのセットアップ』
クラスター例をセットアップするには、これらの手順に従ってください。 クラスターを設定する方法は、TCP/IP、LU 6.2、および単一の伝送キューまたは複数の伝送キューを使用する場合で別々に説明されています。 あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信することによって、クラスターの動作をテストします。

380 ページの『クラスター外部からのワークロード・バランシングの構成』
クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。 その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

関連資料

クラスター・キュー・モニターのサンプル・プログラム (AMQSCLM)

同じキューに複数のインスタンスがある場合のクラスターの例

この例では、1 つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。 1 つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか 1 つのキュー・マネージャーに送信することを選択したりできます。

391 ページの図 59 は、キュー Q3 の定義が複数あるクラスターを示しています。 QM1 のアプリケーションが Q3 へメッセージを書き込む場合、Q3 のどのインスタンスがメッセージを処理するかは必ずしも分かりません。 Q3 のローカル・インスタンスが存在する QM2 または QM4, 上でアプリケーションが実行されてい

る場合、デフォルトで Q3 のローカル・インスタンスが開きます。 CLWLUSEQ キュー属性を設定すると、キューのローカル・インスタンスをキューのリモート・インスタンスと同様に処理できます。

MQOPEN のオプション DefBind は、ターゲットのキュー・マネージャーが選択されるのが、MQOPEN 呼び出しを発行する際か、それとも伝送キューからメッセージを転送する際かを制御します。

DefBind を MQBND_BIND_NOT_FIXED に設定すると、メッセージの伝送時に使用可能になるキューのインスタンスにメッセージを送信できます。 これにより、次のような問題が避けられます。

- メッセージがターゲット・キュー・マネージャーに着信したときに、ターゲット・キューが使用不可になっている。
- キューの状態が変更されている。
- クラスター・キュー別名を使用してメッセージが書き込まれているのに、クラスター・キュー別名のインスタンスが定義されているキュー・マネージャーにターゲット・キューのインスタンスが存在しない。

伝送時にこれらの問題のいずれかが見つかった場合、ターゲット・キューの別の使用可能なインスタンスが検索され、メッセージが転送されます。 使用可能なキューのインスタンスがない場合は、送達不能キューにメッセージが入れられます。

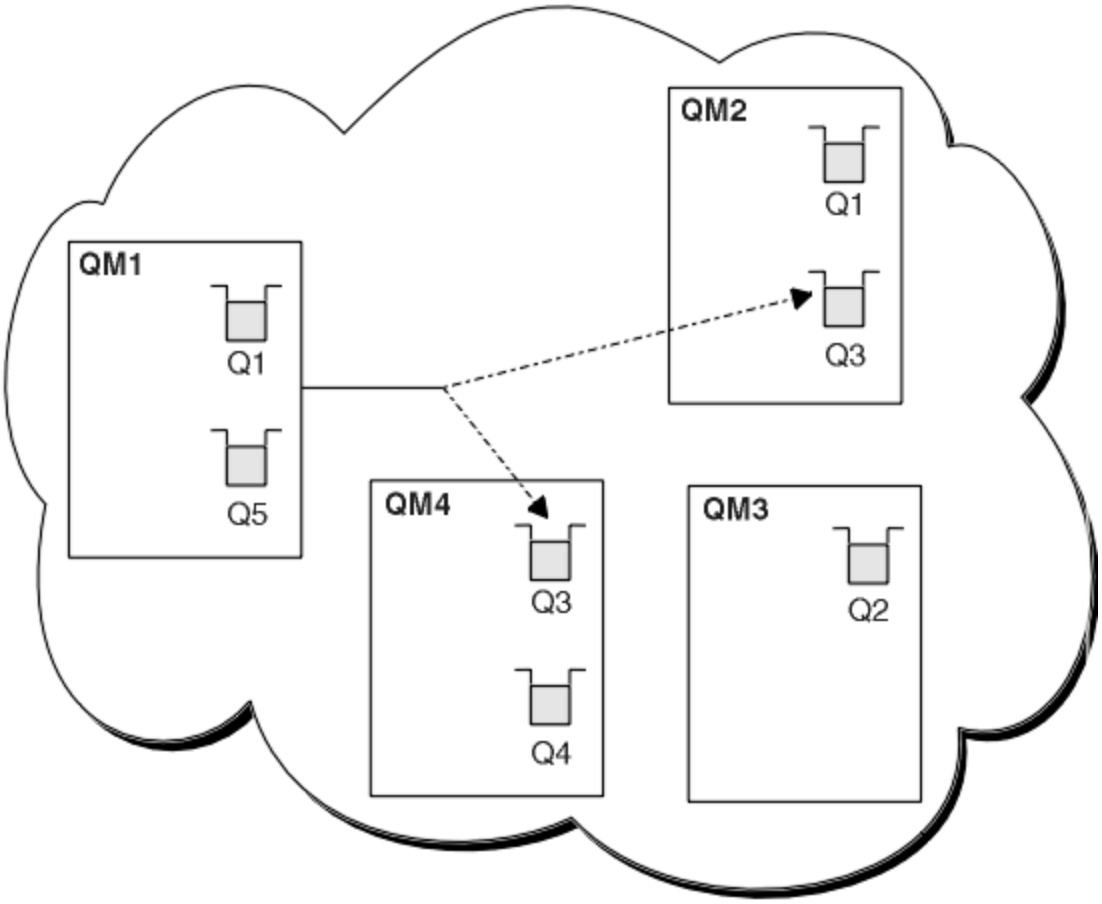


図 *59.* 同じキューの複数のインスタンスがあるクラスター

メッセージを転送できなくなる可能性がある 1 つの要因として、MQBND_BIND_ON_OPEN を使用してメッセージが固定のキュー・マネージャーまたはチャネルに割り当てられている場合があります。 MQOPEN 時にバインドされるメッセージは、別のチャネルに再割り振りされることはありません。 メッセージの再割り振りは、クラスター・チャネルの障害処理が実際に行われているときにのみ行われることに注意してください。 チャネルが既に失敗している場合は、再割り振りは行われません。

宛先キュー・マネージャーがサービス休止になると、システムはメッセージを転送しようとします。 この場合、メッセージが失われたり重複メッセージが作られたりしてメッセージの完全性が損なわれることはありません。 キュー・マネージャーの故障によりメッセージの完全性に疑問がある場合、そのメッセージの転送は行われません。

z/OS IBM MQ for z/OS では、メッセージ再割り振りプロセスが完了するまで、チャネルは完全に停止しません。 モードを FORCE または TERMINATE に設定した状態でチャネルを停止すると、このプロセスが中断するため、その場合は一部の BIND_NOT_FIXED メッセージがすでに別のチャネルに再割り振りされているか、メッセージの順番が狂っている可能性があります。

注 : z/OS

1. 同じキューに複数のインスタンスがあるクラスターを設定する前に、メッセージ相互に依存関係がないことを確認してください。 例えば、特定の順序で処理する必要や同じキュー・マネージャーで処理する必要がないようにします。
2. 同じキューの複数のインスタンスの定義は同じにしてください。 同じにしない場合、MQINQ 呼び出しが違うと、得られる結果にも違いが生じます。

関連概念

アプリケーション・プログラミングおよびクラスター
同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。 ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

関連タスク

キューをローカル側でホストするキュー・マネージャーの追加
以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

クラスター内での 2 つのネットワークの使用
以下の指示に従って、2 つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。 どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

クラスター内での 1 次ネットワークと 2 次ネットワークの使用
以下の説明に従って、あるネットワークを 1 次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。 バックアップ・ネットワークは、1 次ネットワークに問題が発生した場合に使用します。

バックアップとして機能するキューの追加
以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。 シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

使用されるチャネルの数の制限
以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャネルの数を制限します。

キューをホストするより強力なキュー・マネージャーの追加
以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの 2 倍のメッセージ数を処理できるようにします。

キューをローカル側でホストするキュー・マネージャーの追加

以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

始める前に

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- クラスターへの新規キュー・マネージャーの追加での説明に従って、INVENTORY クラスターをセットアップしました。 これには 3 つのキュー・マネージャーが含まれており、LONDON と NEWYORK はどちらも完全リポジトリーを保持し、 PARIS は部分リポジトリーを保持します。 在庫管理アプリケーションは、

ニューヨークのシステムで動作します。このシステムは、キュー・マネージャー NEWYORK に接続されています。 このアプリケーションは、INVENTQ キューのメッセージが到着すると実行されます。

- パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

このタスクについて

以下の手順に従って、ローカルでキューをホストするキュー・マネージャーを追加します。

手順

1. PARIS キュー・マネージャーを変更します。

   パリのアプリケーションがパリとニューヨークの INVENTQ を使用できるようにするには、キュー・マネージャーに通知する必要があります。 PARIS で、以下のコマンドを発行します。

   ```
   ALTER QMGR CLWLUSEQ(ANY)
   ```

2. 在庫管理アプリケーションを調べてメッセージの類縁性を調べる。

   これを調べる前に、在庫管理アプリケーションがメッセージの処理順序に依存しないことを確認してください。 詳細については、メッセージの類縁性の処理を参照してください。

3. 在庫管理アプリケーションをパリのシステムにインストールする
4. クラスター・キュー INVENTQ を定義する。

   INVENTQ キューは、キュー・マネージャー NEWYORK に既にホストされていますが、PARIS によってもホストされます。 キュー・マネージャー PARIS に次のように定義します。

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
   ```

   これですべての定義が完了しました。IBM MQ for z/OS でまだチャネル・イニシエーターを始動していない場合は始動してください。 すべてのプラットフォームで、キュー・マネージャー PARIS 上でリスナー・プログラムを開始してください。 リスナーは、着信ネットワーク要求がないかどうか listen し、必要な場合にクラスター受信側チャネルを起動します。

タスクの結果

394 ページの図 60 は、このタスクによってセットアップされるクラスターを示します。

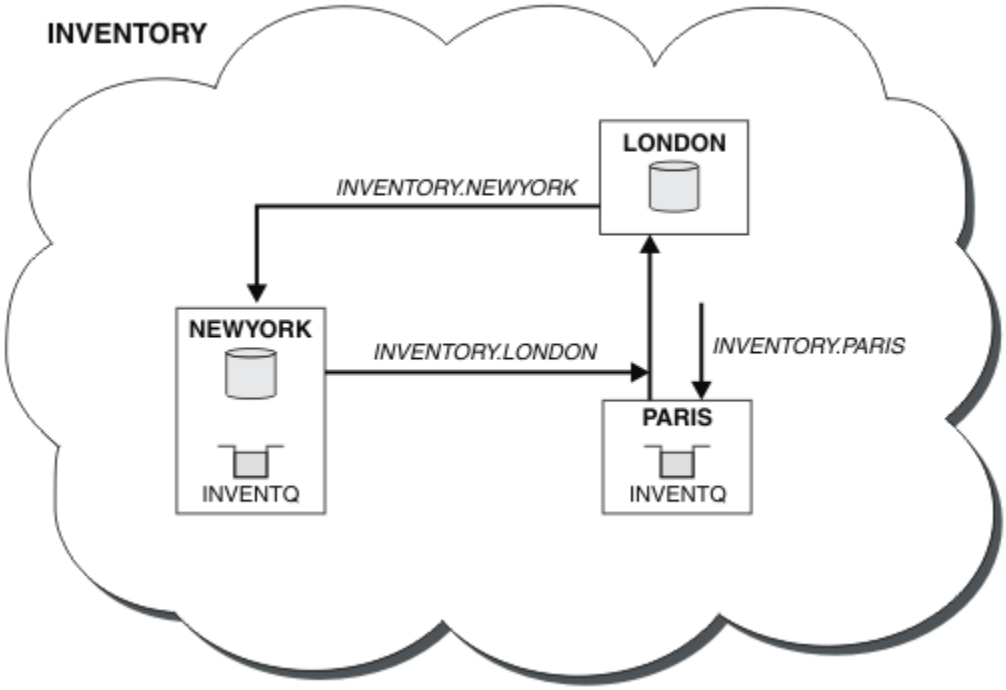


図 *60. 3* つのキュー・マネージャーで構成される *INVENTORY* クラスター

このクラスターの変更では、キュー・マネージャー NEWYORK および LONDON のいずれも変更されませんでした。 これらのキュー・マネージャーの完全リポジトリーは、PARIS の INVENTQ にメッセージを送信できるようにするために必要な情報によって、自動的に更新されます。

次のタスク

現在、INVENTQ キューと在庫管理アプリケーションは、このクラスター内の 2 つのキュー・マネージャーによってホストされています。 これにより、これらの可用性が増大してメッセージのスループットが向上し、ワークロードが 2 つのキュー・マネージャーに分散されるようになります。 キュー・マネージャー LONDON、NEWYORK、PARIS のいずれかによって INVENTQ に書き込まれたメッセージは、PARIS または NEWYORK に交互に経路指定されるため、ワークロードのバランスを取ることができます。

関連概念

同じキューに複数のインスタンスがある場合のクラスターの例
この例では、1 つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。 1 つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか 1 つのキュー・マネージャーに送信することを選択したりできます。

アプリケーション・プログラミングおよびクラスター
同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。 ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

関連タスク

クラスター内での 2 つのネットワークの使用
以下の指示に従って、2 つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。 どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

クラスター内での 1 次ネットワークと 2 次ネットワークの使用
以下の説明に従って、あるネットワークを 1 次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。 バックアップ・ネットワークは、1 次ネットワークに問題が発生した場合に使用します。

バックアップとして機能するキューの追加
以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。 シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

使用されるチャネルの数の制限
以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャネルの数を制限します。

キューをホストするより強力なキュー・マネージャーの追加
以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの 2 倍のメッセージ数を処理できるようにします。

クラスター内での *2* つのネットワークの使用

以下の指示に従って、2 つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。 どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

始める前に

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- 『クラスターへの新規キュー・マネージャーの追加』の説明に従って、INVENTORY クラスターがセットアップされています。 このクラスターには、3 つのキュー・マネージャーが格納されています。このうち、LONDON と NEWYORK は完全リポジトリーを保有し、PARIS は部分リポジトリーを保有しています。在庫管理アプリケーションは、ニューヨークのシステムで動作します。このシステムは、キュー・マネージャー NEWYORK に接続されています。 このアプリケーションは、INVENTQ キューのメッセージが到着すると実行されます。
- 新しい店舗が、2 つの異なるネットワークが存在する TOKYO に追加されます。 どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

このタスクについて

以下の手順に従って、クラスター内の 2 つのネットワークを使用します。

手順

1. TOKYO が最初に参照する完全リポジトリーを決める。

   クラスター内の各キュー・マネージャーは、クラスターに関する情報を収集するために、いずれかの完全リポジトリーを参照する必要があります。 キュー・マネージャーは、独自の部分リポジトリーを作成します。 どのリポジトリーが選択されるかは、あまり重要ではありません。 この例では、NEWYORK が選択されます。 新しいキュー・マネージャーをクラスターに追加すると、このキュー・マネージャーは両方のリポジトリーと通信します。
2. CLUSRCVR チャネルを定義する。

   クラスター内の各キュー・マネージャーに、メッセージを受信できるクラスター受信側を定義する必要があります。 このキュー・マネージャーは、各ネットワークで通信できる必要があります。

   ```
   DEFINE CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME('TOKYO.NETB.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
   channel using network B for TOKYO')
   ```

   ```
   DEFINE CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME('TOKYO.NETA.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
   channel using network A for TOKYO')
   ```
3. キュー・マネージャー TOKYO. で CLUSSDR チャネルを定義します。

クラスター内の各キュー・マネージャーは、その最初の完全リポジトリーへのメッセージの送信に使用できるクラスター送信側チャネルを 1 つ定義する必要があります。 この場合、NEWYORK を選択したので、TOKYO には次の定義が必要になります。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
channel from TOKYO to repository at NEWYORK')
```

これですべての定義が完了しました。IBM MQ for z/OS でまだチャネル・イニシエーターを開始していない場合は開始します。 すべてのプラットフォームで、キュー・マネージャー PARIS 上でリスナー・プログラムを開始してください。 リスナー・プログラムは、着信ネットワーク要求がないかどうか listen し、必要な場合にクラスター受信側チャネルを起動します。

タスクの結果

396 ページの図 61 は、このタスクによってセットアップされるクラスターを示します。

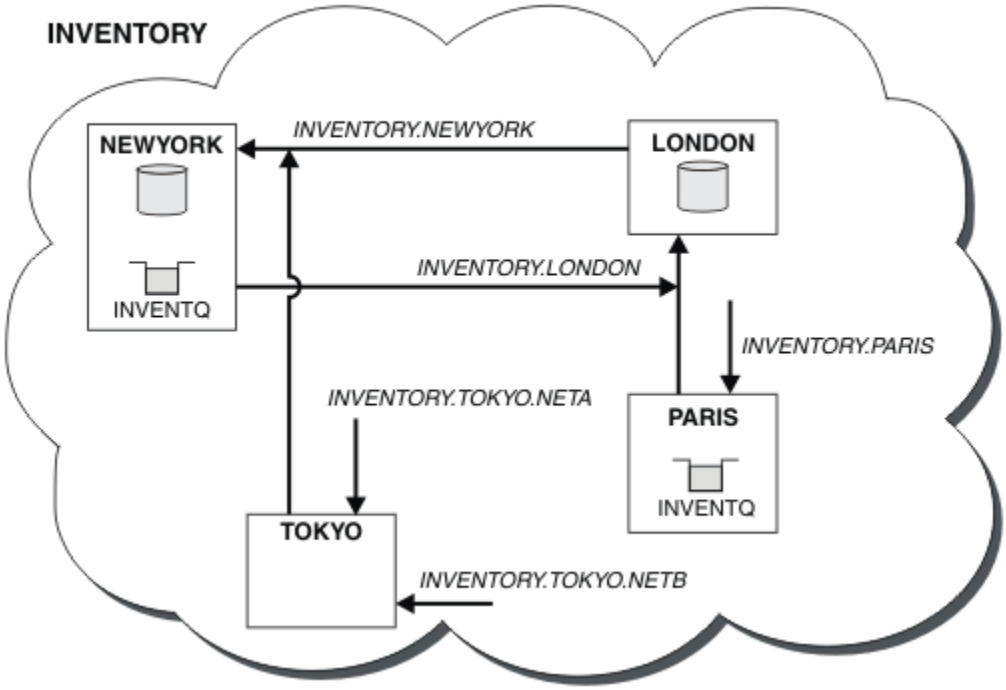


図 *61. 4* つのキュー・マネージャーで構成される *INVENTORY* クラスター

3 つの定義のみを作成して、クラスターにキュー・マネージャー TOKYO を追加し、2 つの異なるネットワーク経路を使用できるようにしました。

関連概念

同じキューに複数のインスタンスがある場合のクラスターの例
この例では、1 つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。 1 つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか 1 つのキュー・マネージャーに送信することを選択したりできます。

アプリケーション・プログラミングおよびクラスター
同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。 ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

関連タスク

キューをローカル側でホストするキュー・マネージャーの追加

以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

クラスター内での 1 次ネットワークと 2 次ネットワークの使用
以下の説明に従って、あるネットワークを 1 次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。 バックアップ・ネットワークは、1 次ネットワークに問題が発生した場合に使用します。

バックアップとして機能するキューの追加
以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。 シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

使用されるチャネルの数の制限
以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャネルの数を制限します。

キューをホストするより強力なキュー・マネージャーの追加
以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの 2 倍のメッセージ数を処理できるようにします。

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

クラスター内での *1* 次ネットワークと *2* 次ネットワークの使用

以下の説明に従って、あるネットワークを 1 次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。 バックアップ・ネットワークは、1 次ネットワークに問題が発生した場合に使用します。

始める前に

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- 395 ページの『クラスター内での 2 つのネットワークの使用』 で説明されているように、INVENTORY クラスターがセットアップされています。 このクラスターには、4 つのキュー・マネージャーが格納されています。このうち、LONDON と NEWYORK はどちらも完全リポジトリーを保有し、PARIS と TOKYO は部分リポジトリーを保有しています。 在庫管理アプリケーションは、ニューヨークのシステムで実行され、キュー・マネージャー NEWYORK に接続されます。 TOKYO キュー・マネージャーには、通信可能な 2 つの異なるネットワークがあります。
- ネットワークの 1 つを 1 次ネットワークにし、もう 1 つのネットワークをバックアップ・ネットワークにするつもりです。 さらに、1 次ネットワークに問題が発生した場合にバックアップ・ネットワークを使用することを計画しています。

このタスクについて

NETPRTY 属性を使用して、クラスター内の 1 次ネットワークと 2 次ネットワークを構成します。

手順

TOKYO で既存の CLUSRCVR チャネルを変更する。

ネットワーク A チャネルが 1 次チャネルであり、ネットワーク B チャネルが 2 次チャネルであることを示すために、次のコマンドを使用します。

a) `ALTER CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) NETPRTY(2) DESCR('Main cluster-receiver channel for TOKYO')`

```
b) ALTER CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) NETPRTY(1)
   DESCR('Backup cluster-receiver channel for TOKYO')
```

次のタスク

異なるネットワーク優先度を使用するようにチャネルを構成することにより、クラスターに対して、1 次ネットワークと 2 次ネットワークが存在することを定義しました。 これらのチャネルを使用するクラスター内のキュー・マネージャーは、1 次ネットワークが使用可能には必ずこれを自動的に使用します。 キュー・マネージャーは、1 次ネットワークが使用不可の場合は 2 次ネットワークを使用するためにフェイルオーバーします。

関連概念

同じキューに複数のインスタンスがある場合のクラスターの例
この例では、1 つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。 1 つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか 1 つのキュー・マネージャーに送信することを選択したりできます。

アプリケーション・プログラミングおよびクラスター
同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。 ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

関連タスク

キューをローカル側でホストするキュー・マネージャーの追加
以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

クラスター内での 2 つのネットワークの使用
以下の指示に従って、2 つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。 どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

バックアップとして機能するキューの追加
以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。 シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

使用されるチャネルの数の制限
以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャネルの数を制限します。

キューをホストするより強力なキュー・マネージャーの追加
以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの 2 倍のメッセージ数を処理できるようにします。

バックアップとして機能するキューの追加

以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。 シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

始める前に

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- 313 ページの『クラスターにキュー・マネージャーを追加する』 で説明されているように、INVENTORY クラスターがセットアップされています。 これには 3 つのキュー・マネージャーが含まれており、LONDON と NEWYORK はどちらも完全リポジトリーを保持し、 PARIS は部分リポジトリーを保持します。在庫管理アプリケーションは、ニューヨークのシステムで動作します。このシステムは、キュー・マネージャー NEWYORK に接続されています。 このアプリケーションは、INVENTQ キューのメッセージが到着すると実行されます。

- 現在ニューヨークで稼働している在庫管理システムのバックアップを提供するため、シカゴで新しい店舗の準備が進められています。 シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

このタスクについて

以下の手順に従って、バックアップとして機能するキューを追加します。

手順

1. CHICAGO が最初に参照する完全リポジトリーを決める

   クラスター内の各キュー・マネージャーは、クラスターに関する情報を収集するために、いずれかの完全リポジトリーを参照する必要があります。 キュー・マネージャーは、独自の部分リポジトリーを作成します。 特定のキュー・マネージャーにどのリポジトリーが選択されるかは、あまり重要ではありません。 この例では、NEWYORK が選択されます。 新しいキュー・マネージャーをクラスターに追加すると、このキュー・マネージャーは両方のリポジトリーと通信します。

2. CLUSRCVR チャネルを定義する。

   クラスター内の各キュー・マネージャーに、メッセージを受信できるクラスター受信側を定義する必要があります。 CHICAGO には、以下のように定義します。

   ```
   DEFINE CHANNEL(INVENTORY.CHICAGO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(CHICAGO.CMSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-receiver
   channel for CHICAGO')
   ```

3. キュー・マネージャー CHICAGO に CLUSSDR チャネルを定義する。

   クラスター内の各キュー・マネージャーは、その最初の完全リポジトリーへのメッセージの送信に使用できるクラスター送信側チャネルを 1 つ定義する必要があります。 この場合、NEWYORK を選択したので、CHICAGO には次の定義が必要になります。

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
   channel from CHICAGO to repository at NEWYORK')
   ```

4. 既存のクラスター・キュー INVENTQ を変更する。

   キュー・マネージャー NEWYORK がホストしている INVENTQ は、キューのメイン・インスタンスです。

   ```
   ALTER QLOCAL(INVENTQ) CLWLPRTY(2)
   ```

5. 在庫管理アプリケーションを調べてメッセージの類縁性を調べる。

   これを調べる前に、在庫管理アプリケーションがメッセージの処理順序に依存しないことを確認してください。

6. 在庫管理アプリケーションを CHICAGO にインストールする。

7. バックアップ・クラスター・キュー INVENTQ を定義する

   INVENTQ は、キュー・マネージャー NEWYORK にすでにホストされていますが、CHICAGO によってもバックアップとしてホストされます。 キュー・マネージャー CHICAGO に次のように INVENTQ を定義します。

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) CLWLPRTY(1)
   ```

   これですべての定義が完了しました。IBM MQ for z/OS でまだチャネル・イニシエーターを開始していない場合は開始します。 すべてのプラットフォームで、キュー・マネージャー CHICAGO でリスナー・プログラムを始動します。 リスナー・プログラムは、着信ネットワーク要求がないかどうか listen し、必要な場合にクラスター受信側チャネルを起動します。

タスクの結果

400 ページの図 62 は、このタスクによってセットアップされるクラスターを示します。

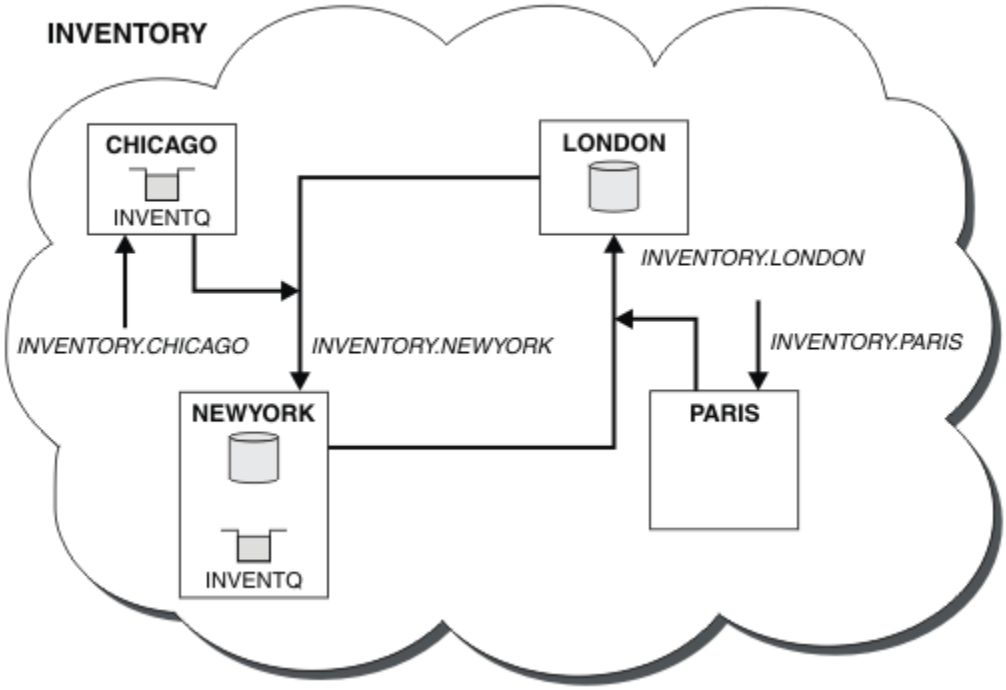


図 *62. 4* つのキュー・マネージャーで構成される *INVENTORY* クラスター

現在、INVENTQ キューと在庫管理アプリケーションは、このクラスター内の 2 つのキュー・マネージャーによってホストされています。 CHICAGO キュー・マネージャーはバックアップです。 INVENTQ に書き込まれたメッセージは、CHICAGO の代わりに、送信されたときに使用不可でない限り NEWYORK に経路指定されます。

注 :

リモート・キュー・マネージャーの可用性は、 そのキュー・マネージャーに対するチャネルの状況に基づきます。 チャネルの開始時に、チャネルの状態は何度か変わります。その中には、クラスターのワークロード管理アルゴリズムにとって好ましくない状態もあります。 これは、実際に、優先順位がより高い (プライマリー) 宛先に対するチャネルが開始されていても、 優先順位がより低い (バックアップ) 宛先が選択される可能性があることを意味します。

バックアップ宛先に対してメッセージが送信されないようにする必要がある場合は、CLWLPRTY を使用しないでください。 個別のキューを使用するか、または CLWLRANK を使用してプライマリーからバックアップに手動で切り替えることを検討してください。

関連概念

同じキューに複数のインスタンスがある場合のクラスターの例
この例では、1 つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。 1 つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか 1 つのキュー・マネージャーに送信することを選択したりできます。

アプリケーション・プログラミングおよびクラスター
同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。 ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

関連タスク

キューをローカル側でホストするキュー・マネージャーの追加
以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

クラスター内での 2 つのネットワークの使用

以下の指示に従って、2 つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。 どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

クラスター内での 1 次ネットワークと 2 次ネットワークの使用
以下の説明に従って、あるネットワークを 1 次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。 バックアップ・ネットワークは、1 次ネットワークに問題が発生した場合に使用します。

使用されるチャネルの数の制限
以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャネルの数を制限します。

キューをホストするより強力なキュー・マネージャーの追加
以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの 2 倍のメッセージ数を処理できるようにします。

使用されるチャネルの数の制限

以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャネルの数を制限します。

始める前に

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- 価格チェック・アプリケーションはさまざまなキュー・マネージャーにインストールされます。 使用されるチャネルの数を少なく抑えるために、各サーバーが実行するアクティブ・チャネルの数が制限されます。 このアプリケーションは、PRICEQ キューにメッセージが到着すると実行されます。
- 4 つのサーバー・キュー・マネージャーが価格チェック・アプリケーションをホストします。 2 つの照会キュー・マネージャーがメッセージを PRICEQ に送信して価格を照会します。 残り 2 つのキュー・マネージャーは、完全リポジトリーとして構成されます。

このタスクについて

以下の手順に従って、使用されるチャネルの数を制限します。

手順

1. 2 つの完全リポジトリーを選択します。

   価格チェック・クラスターの完全リポジトリーとなる 2 つのキュー・マネージャーを選択します。 これらは、REPOS1 および REPOS2 と呼ばれます。

   以下のコマンドを発行します。

   ```
   ALTER QMGR REPOS(PRICECHECK)
   ```

2. 各キュー・マネージャーで CLUSRCVR チャネルを定義する。

   クラスター内の各キュー・マネージャーで、クラスター受信側チャネルとクラスター送信側チャネルを定義してください。 どちらを最初に定義しても構いません。

   ```
   DEFINE CHANNEL(PRICECHECK.SERVE1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(SERVER1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-receiver channel')
   ```

3. 各キュー・マネージャーで CLUSSDR チャネルを定義する。

   各キュー・マネージャーで CLUSSDR を定義して、そのキュー・マネージャーをいずれかの完全リポジトリー・キュー・マネージャーにリンクします。

```
DEFINE CHANNEL(PRICECHECK.REPOS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(REPOS1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-sender channel to
repository queue manager')
```

4. 価格チェック・アプリケーションをインストールする
5. すべてのサーバー・キュー・マネージャーで PRICEQ キューを定義する

   各サーバー・キュー・マネージャーで次のコマンドを発行します。

   ```
   DEFINE QLOCAL(PRICEQ) CLUSTER(PRICECHECK)
   ```

6. 照会で使用されるチャネルの数を制限する。

   各照会キュー・マネージャーで、次のコマンドを発行して、使用されるアクティブ・チャネルの数を制限します。

   ```
   ALTER QMGR CLWLMRUC(2)
   ```

7. IBM MQ for z/OS でまだチャネル・イニシエーターを始動していない場合は始動する。 すべてのプラットフォームで、リスナー・プログラムを始動する。

   リスナー・プログラムは、着信ネットワーク要求がないかどうか listen し、必要な場合にクラスター受信側チャネルを起動します。

タスクの結果

402 ページの図 63 は、このタスクによってセットアップされるクラスターを示します。

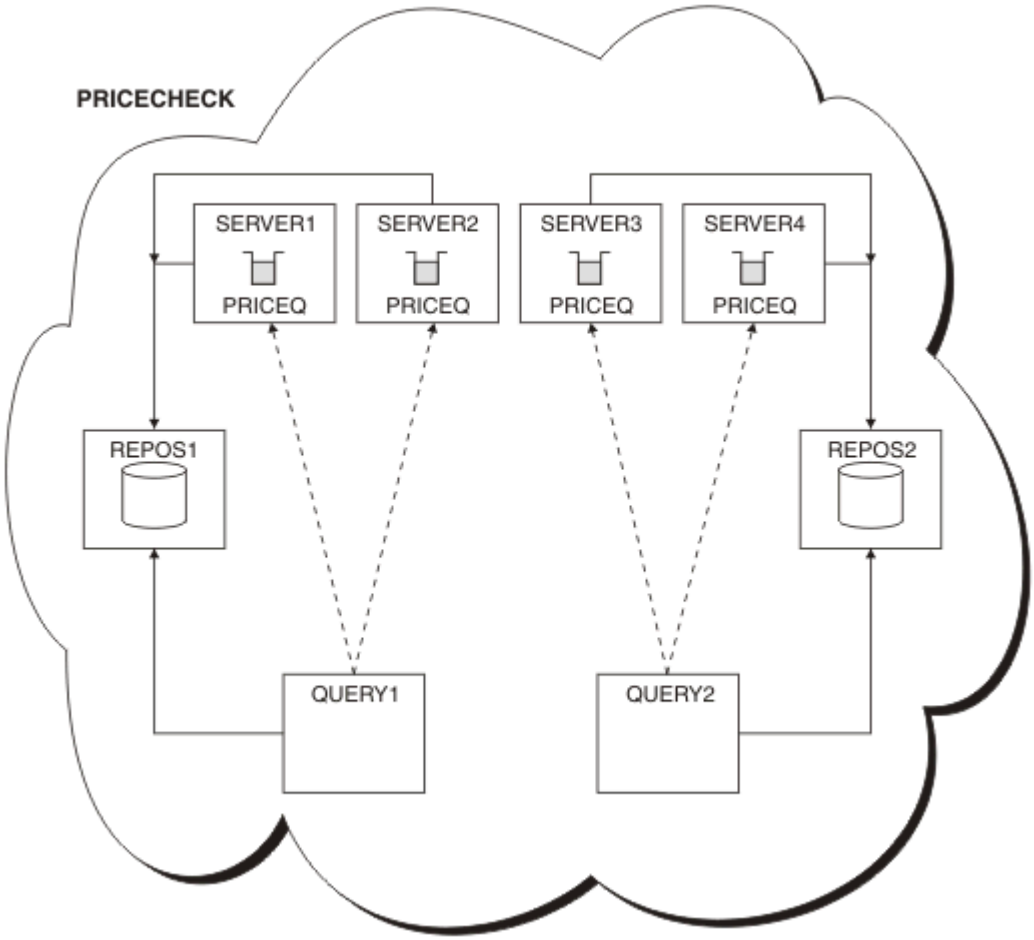


図 *63. 4* つのサーバー・キュー・マネージャー、*2* つのリポジトリー、および *2* つの照会キュー・マネージャーを持つ *PRICECHECK* クラスター

PRICECHECK クラスターには、使用可能な PRICEQ キューの 4 つのインスタンスがありますが、各照会キュー・マネージャーはそのうちの 2 つのみを使用します。 例えば、QUERY1 キュー・マネージャーは、SERVER1 および SERVER2 キュー・マネージャーへのアクティブ・チャネルのみを持っています。 SERVER1 が使用不可になった場合、QUERY1 キュー・マネージャーは、SERVER3 などの別のキュー・マネージャーを使用し始めます。

関連概念

同じキューに複数のインスタンスがある場合のクラスターの例
この例では、1 つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。 1 つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか 1 つのキュー・マネージャーに送信することを選択したりできます。

アプリケーション・プログラミングおよびクラスター
同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。 ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

関連タスク

キューをローカル側でホストするキュー・マネージャーの追加
以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

クラスター内での 2 つのネットワークの使用
以下の指示に従って、2 つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。 どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

クラスター内での 1 次ネットワークと 2 次ネットワークの使用
以下の説明に従って、あるネットワークを 1 次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。 バックアップ・ネットワークは、1 次ネットワークに問題が発生した場合に使用します。

バックアップとして機能するキューの追加
以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。 シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

キューをホストするより強力なキュー・マネージャーの追加
以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの 2 倍のメッセージ数を処理できるようにします。

キューをホストするより強力なキュー・マネージャーの追加

以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの 2 倍のメッセージ数を処理できるようにします。

始める前に

注 : クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- 313 ページの『クラスターにキュー・マネージャーを追加する』で説明されているように、 INVENTORY クラスターがセットアップされています。 このクラスターには、3 つのキュー・マネージャーが格納されています。このうち、LONDON と NEWYORK はどちらも完全リポジトリーを保有します。PARIS は部分リポジトリーを保有して、INVENTQ からメッセージを書き込みます。 在庫管理アプリケーションは、ニューヨークのシステムで動作します。このシステムは、キュー・マネージャー NEWYORK に接続されています。 このアプリケーションは、INVENTQ キューのメッセージが到着すると実行されます。

- ロサンゼルスでは、新しい店舗の準備が進められています。 処理能力を増強するために、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させます。 新しいキュー・マネージャーは、ニューヨークの 2 倍のメッセージ数を処理できます。

このタスクについて

以下の手順に従って、キューをホストする、より強力なキュー・マネージャーを追加します。

手順

1. LOSANGELES が最初に参照する完全リポジトリーを決める。
2. クラスター内の各キュー・マネージャーは、クラスターに関する情報を収集するために、いずれかの完全リポジトリーを参照する必要があります。 キュー・マネージャーは、独自の部分リポジトリーを作成します。 どのリポジトリーが選択されるかは、あまり重要ではありません。 この例では、NEWYORK が選択されます。 新しいキュー・マネージャーをクラスターに追加すると、このキュー・マネージャーは両方のリポジトリーと通信します。

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from LOSANGELES to repository at NEWYORK')
   ```

3. キュー・マネージャー LOSANGELES に CLUSRCVR チャネルを定義する。

   クラスター内の各キュー・マネージャーにクラスター受信側チャネルを定義する必要があります。このチャネルにより、キュー・マネージャーはメッセージを受信することができます。 LOSANGELES には、以下のように定義します。

   ```
   DEFINE CHANNEL(INVENTORY.LOSANGELES) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(LOSANGELES.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for queue manager LOSANGELES')
   CLWLWGHT(2)
   ```

   クラスター受信側チャネルは、このキュー・マネージャーがクラスター INVENTORY 内の他のキュー・マネージャーからメッセージを受信できる状態になったことを通知します。 CLWLWGHT を 2 に設定すると、ロサンゼルスのキュー・マネージャーがニューヨークの 2 倍の在庫メッセージを受け取れるようになります (NEWYORK のチャネルを 1 に設定した場合)。
4. キュー・マネージャー NEWYORK で CLUSRCVR チャネルを変更する。

   ロサンゼルスのキュー・マネージャーがニューヨークの 2 倍の在庫メッセージを受け取るようにします。 そのために、クラスター受信側チャネルの定義を変更します。

   ```
   ALTER CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) CLWLWGHT(1)
   ```

5. 在庫管理アプリケーションを調べてメッセージの類縁性を調べる。

   これを調べる前に、在庫管理アプリケーションがメッセージの処理順序に依存しないことを確認してください。
6. 在庫管理アプリケーションをロサンゼルスのシステムにインストールする。
7. クラスター・キュー INVENTQ を定義する。

   INVENTQ キューは、キュー・マネージャー NEWYORK に既にホストされていますが、LOSANGELES によってもホストされます。 キュー・マネージャー LOSANGELES に次のように INVENTQ キューを定義します。

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
   ```

   これですべての定義が完了しました。IBM MQ for z/OS でまだチャネル・イニシエーターを開始していない場合は開始します。 すべてのプラットフォームで、キュー・マネージャー LOSANGELES 上でリスナー・プログラムを開始します。 リスナー・プログラムは、着信ネットワーク要求がないかどうか listen し、必要な場合にクラスター受信側チャネルを起動します。

タスクの結果

403 ページの『キューをホストするより強力なキュー・マネージャーの追加』 は、このタスクによってセットアップされるクラスターを示します。

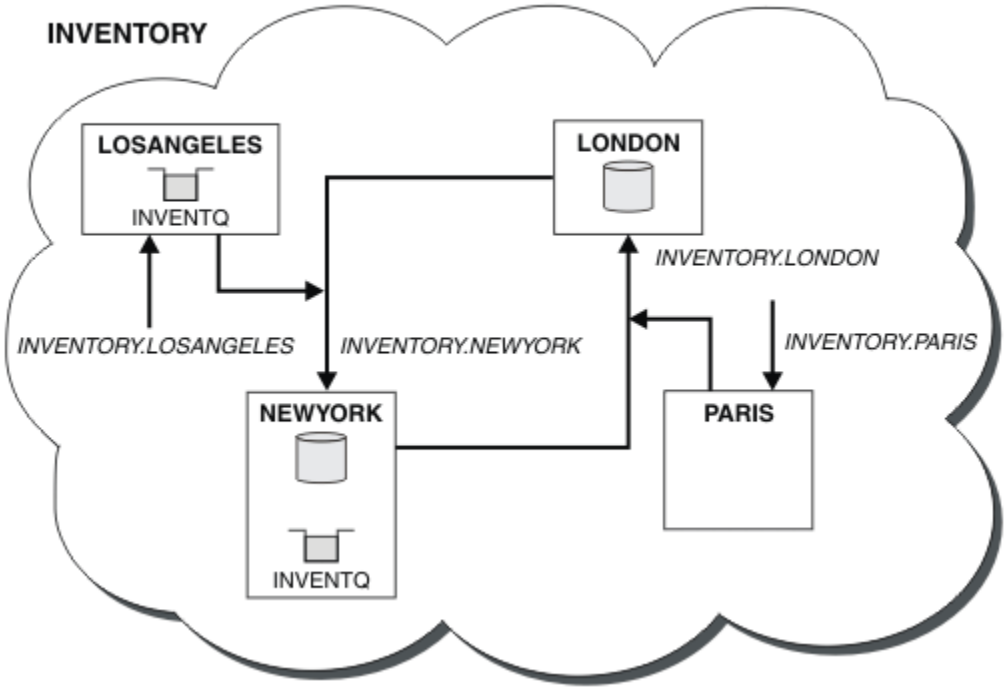


図 *64. 4* つのキュー・マネージャーが格納されている *INVENTORY* クラスター

クラスターをこのように変更するときに、キュー・マネージャー LONDON および PARIS を変更する必要はありませんでした。 これらのキュー・マネージャーのリポジトリーは、LOSANGELES の INVENTQ にメッセージを送信できるよう設定するために必要な情報によって、自動的に更新されます。

次のタスク

INVENTQ キューと在庫管理アプリケーションは、このクラスター内の 2 つのキュー・マネージャーによってホストされます。 この構成により、可用性が増大してメッセージのスループットが向上し、ワークロードが 2 つのキュー・マネージャーに分散されるようになります。 LOSANGELES または NEWYORK によって INVENTQ に書き込まれたメッセージは、可能なときは必ずローカル・キュー・マネージャーのインスタンスによって処理されます。 LONDON または PARIS によって書き込まれたメッセージは、LOSANGELES または NEWYORK に経路指定され、2 倍のメッセージ数が LOSANGELES に送信されます。

関連概念

同じキューに複数のインスタンスがある場合のクラスターの例
この例では、1 つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。 1 つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか 1 つのキュー・マネージャーに送信することを選択したりできます。

アプリケーション・プログラミングおよびクラスター
同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。 ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

関連タスク

キューをローカル側でホストするキュー・マネージャーの追加

以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

クラスター内での 2 つのネットワークの使用
以下の指示に従って、2 つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。 どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

クラスター内での 1 次ネットワークと 2 次ネットワークの使用
以下の説明に従って、あるネットワークを 1 次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。 バックアップ・ネットワークは、1 次ネットワークに問題が発生した場合に使用します。

バックアップとして機能するキューの追加
以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。 シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

使用されるチャネルの数の制限
以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャネルの数を制限します。

アプリケーション・プログラミングおよびクラスター

同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。 ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

アプリケーションは MQOPEN 呼び出しを使用して、キューをオープンすることができます。 アプリケーションは MQPUT 呼び出しを使用して、メッセージをオープン・キューに書き込みます。 また、MQPUT1 呼び出しを使用すると、まだオープンしていないキューに単一のメッセージを書き込むことができます。

同じキューの複数のインスタンスを持つクラスターを設定する場合、アプリケーション・プログラミングの考慮事項は特にありません。 しかし、クラスター化をワークロード管理に役立てるためには、アプリケーションの変更が必要な場合があります。 同じキューの複数の定義があるネットワークを設定する場合は、メッセージの類縁性について、アプリケーションを確認してください。

例えば、質問と回答の形式の一連のメッセージの流れを使用する 2 つのアプリケーションがあるとします。 質問を送信したキュー・マネージャーと同じキュー・マネージャーに回答を戻すことが想定されます。 この場合、ワークロード管理ルーチンが、応答キューのコピーをホストしているどのキュー・マネージャーにもメッセージを送信しないことが重要です。

メッセージの順次処理を必要とするアプリケーションもあります (例えば、順番に取り出す必要があるメッセージのバッチを送信するデータベース複製アプリケーション)。 セグメント化されたメッセージを使用するときも、類縁性の問題が起こることがあります。

ローカル・バージョンまたはリモート・バージョンの宛先キューのオープン

キュー・マネージャーが、ローカル・バージョンまたはリモート・バージョンの宛先キューのいずれを使用するかを選択する方法を理解してください。

1. キュー・マネージャーは、メッセージの読み取りまたはキューの属性設定を行う場合には、ローカル・バージョンの宛先キューをオープンします。
2. 以下の条件が 1 つ以上該当する場合、キュー・マネージャーは、メッセージを書き込む宛先キューの任意のインスタンスをオープンします。
   - ローカル・バージョンの宛先キューが存在していません。
   - キュー・マネージャーが ALTER QMGR で CLWLUSEQ(ANY) を指定しています。
   - キュー・マネージャーのキューが CLWLUSEQ(ANY) を指定しています。

関連概念
同じキューに複数のインスタンスがある場合のクラスターの例
この例では、1 つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。 1 つのメッセージをキューの特定のインスタンスに強制的に送

信したり、一連のメッセージをいずれか 1 つのキュー・マネージャーに送信することを選択したりできます。

関連タスク

キューをローカル側でホストするキュー・マネージャーの追加
以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

クラスター内での 2 つのネットワークの使用
以下の指示に従って、2 つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。 どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

クラスター内での 1 次ネットワークと 2 次ネットワークの使用
以下の説明に従って、あるネットワークを 1 次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。 バックアップ・ネットワークは、1 次ネットワークに問題が発生した場合に使用します。

バックアップとして機能するキューの追加
以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。 シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

使用されるチャネルの数の制限
以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャネルの数を制限します。

キューをホストするより強力なキュー・マネージャーの追加
以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの 2 倍のメッセージ数を処理できるようにします。

メッセージの類縁性の処理
メッセージの類縁性が、優れたプログラミング設計に含まれることはほとんどありません。 クラスタリングを最大限に使用するには、メッセージの類縁性を除去する必要があります。 メッセージ類縁性を除去できない場合は、関連メッセージが、必ず同じチャネルにより同じキュー・マネージャーに送信されるようにすることができます。

メッセージの類縁性があるアプリケーションがある場合、クラスターの使用を開始する前に、メッセージの類縁性を除去してください。

メッセージの類縁性を除去すると、アプリケーションの可用性が向上します。 アプリケーションは、メッセージの類縁性があるメッセージのバッチをキュー・マネージャーに送信します。 バッチの一部のみを受信した後に、キュー・マネージャーに障害が起こります。 送信側のキュー・マネージャーは、メッセージの続きを送信するには、キュー・マネージャーが復旧するのを待ってから、完了していないメッセージのバッチを処理する必要があります。

メッセージの類縁性を除去すると、アプリケーションのスケーラビリティーも改善されます。 類縁性のあるメッセージのバッチは、宛先キュー・マネージャーのリソースを、後続のメッセージを待つ間ロックすることがあります。 これらのリソースのロックが長期間になると、他のアプリケーションの処理の妨げになります。

さらに、メッセージの類縁性があると、クラスター・ワークロード管理ルーチンで最適のキュー・マネージャーを選択することができなくなります。

類縁性を除去するには、次の可能性を検討してください。

- メッセージに状態情報を入れておく
- すべてのキュー・マネージャーからアクセスできる不揮発性記憶装置 (例えば、Db2 データベース) に状態情報を保持する
- 複数のキュー・マネージャーからアクセスできるように読み取り専用データを複製する

メッセージの類縁性を除去するためにアプリケーションを変更するのが適切でない場合は、いろいろな解決法があります。

MQOPEN 呼び出しで特定の宛先を指定する

それぞれの MQOPEN 呼び出しにリモート・キュー名およびキュー・マネージャー名を指定すると、そのオブジェクト・ハンドルを使用してキューに書き込まれるメッセージはすべて同じキュー・マネージャー (ローカル・キュー・マネージャーの場合もある) に送信されます。

それぞれの MQOPEN 呼び出しにリモート・キュー名およびキュー・マネージャー名を指定することには、次の欠点があります。

- ワークロード・バランシングが実行されません。 クラスター・ワークロード・バランシングの利点を利用できません。
- 宛先キュー・マネージャーがリモートであって、そのキュー・マネージャーに複数のチャネルが接続されている場合、メッセージがいろいろな経路で送信されるため、メッセージの順序が保持されない可能性があります。
- ローカル・キュー・マネージャーに宛先キュー・マネージャーと同じ名前の伝送キューの定義がある場合は、メッセージはその伝送キューに送信され、クラスター伝送キューには送信されません。

応答先キュー・マネージャー・フィールドにキュー・マネージャー名を戻す

バッチの最初のメッセージを受け取るキュー・マネージャーは、その名前を応答の中で戻すことができます。 これはメッセージ記述子の ReplyToQMgr フィールドを使用して行われます。 送信側のキュー・マネージャーは、応答先キュー・マネージャー名を抜き出し、後続のすべてのメッセージにその名前を指定することができます。

応答からの ReplyToQMgr 情報を使用することには、次のような欠点があります。

- 要求中のキュー・マネージャーは最初のメッセージに対する応答を待つ必要がある
- 後続のメッセージを送信する前に、ReplyToQMgr 情報を見つけて使用するために追加のコードを作成する必要がある
- キュー・マネージャーへの経路が複数ある場合、メッセージの順序が保持されないことがある

MQOPEN 呼び出しで MQOO_BIND_ON_OPEN オプションを設定する

MQOPEN 呼び出しの MQOO_BIND_ON_OPEN オプションを使用して、すべてのメッセージを強制的に同じ宛先に書き込みます。 メッセージ・グループをクラスターで使用する場合は、'MQOO_BIND_ON_OPEN か'MQOO_BIND_ON_GROUP どちらか を指定しなければならない。

キューをオープンして MQOO_BIND_ON_OPEN を指定すると、そのキューに送信されるメッセージはすべてそのキューの同じインスタンスに強制的に送信されます。 MQOO_BIND_ON_OPEN はすべてのメッセージを同じキュー・マネージャーにバインドし、しかも同じ経路にバインドします。 例えば、同じ宛先への経路として IP 経路と NetBIOS 経路がある場合、キューをオープンするときにその 1 つが選択され、獲得されたオブジェクト・ハンドルを使用して同じキューに書き込まれるすべてのメッセージにこの選択が適用されます。

MQOO_BIND_ON_OPEN を指定すると、すべてのメッセージが同じ宛先に経路指定されます。 このため、メッセージの類縁性があるアプリケーションが分析されません。 宛先が使用できない場合は、それが再び使用可能になるまでメッセージは伝送キューにとどまります。

MQOO_BIND_ON_OPEN は、キューをオープンするときオブジェクト記述子にキュー・マネージャー名を指定する場合にも適用されます。 指定したキュー・マネージャーには複数の経路があることがあります。 例えば、複数のネットワーク・パスがある場合や、別のキュー・マネージャーが別名を定義している場合があります。 MQOO_BIND_ON_OPEN を指定すると、キューをオープンするときに経路が選択されます。

注: これがお勧めする方法です。 ただし、キュー・マネージャーがクラスター・キューの別名を通知するマルチホップ構成では、この方法は使用できません。 また、アプリケーションがメッセージ・グループごとに同じキュー・マネージャーの別々のキューを使用する場合も、この方法は役に立ちません。

MQOPEN 呼び出しで MQOO_BIND_ON_OPEN を指定する代わりに、キュー定義を変更することもできます。 キュー定義に DEFBIND(OPEN) を指定し、MQOPEN 呼び出しの DefBind オプションでデフォルトの MQOO_BIND_AS_Q_DEF を使用するようにします。

MQOPEN 呼び出しで MQOO_BIND_ON_GROUP オプションを設定する

MQOPEN 呼び出しの MQOO_BIND_ON_GROUP オプションを使用して、グループ内のすべてのメッセージを強制的に同じ宛先に書き込みます。 メッセージ・グループをクラスターで使用する場合は、'MQOO_BIND_ON_OPEN か'MQOO_BIND_ON_GROUP どちらか を指定しなければならない。

キューをオープンして MQOO_BIND_ON_GROUP を指定すると、そのキューに送信されるグループのメッセージはすべてそのキューの同じインスタンスに強制的に送信されます。 MQOO_BIND_ON_GROUP はグループのすべてのメッセージを同じキュー・マネージャーにバインドし、しかも同じ経路にバインドします。 例えば、同じ宛先への経路として IP 経路と NetBIOS 経路がある場合、キューをオープンするときにその 1 つが選択され、獲得されたオブジェクト・ハンドルを使用して同じキューに書き込まれるグループのすべてのメッセージにこの選択が適用されます。

MQOO_BIND_ON_GROUP を指定すると、グループのすべてのメッセージが強制的に同じ宛先に経路指定されます。 このため、メッセージの類縁性があるアプリケーションが分析されません。 宛先が使用できない場合は、それが再び使用可能になるまでメッセージは伝送キューにとどまります。

MQOO_BIND_ON_GROUP は、キューをオープンするときオブジェクト記述子にキュー・マネージャー名を指定する場合にも適用されます。 指定したキュー・マネージャーには複数の経路があることがあります。 例えば、複数のネットワーク・パスがある場合や、別のキュー・マネージャーが別名を定義している場合があります。 MQOO_BIND_ON_GROUP を指定すると、キューをオープンするときに経路が選択されます。

MQOO_BIND_ON_GROUP を有効にするには、MQPUT に MQPMO_LOGICAL_ORDER Put オプションを組み込む必要があります。 メッセージの MQMD の **GroupId** を MQGI_NONE に設定できます。 また、メッセージの MQMD **MsgFlags** フィールド内に次のメッセージ・フラグを含める必要があります。

- グループの最後のメッセージ: MQMF_LAST_MSG_IN_GROUP
- グループの他のすべてのメッセージ: MQMF_MSG_IN_GROUP

MQOO_BIND_ON_GROUP が指定されていても、メッセージがグループ化されていない場合、MQOO_BIND_NOT_FIXED が指定された場合と同じ振る舞いになります。

注 : これはグループ内のメッセージが同じ宛先に送信されることを確実にする推奨手法です。 ただし、キュー・マネージャーがクラスター・キューの別名を通知するマルチホップ構成では機能しません。

MQOPEN 呼び出しで MQOO_BIND_ON_GROUP を指定する代わりに、キュー定義を変更することもできます。 キュー定義に DEFBIND(GROUP) を指定し、MQOPEN 呼び出しの DefBind オプションでデフォルトの MQOO_BIND_AS_Q_DEF を使用するようにします。

カスタマイズされたクラスター・ワークロード出口プログラムを作成する

アプリケーションを変更する代わりに、クラスター・ワークロード出口プログラムを作成して、メッセージの類縁性の問題を回避することもできます。 クラスター・ワークロード出口プログラムを作成することは簡単ではないので、お勧めできる解決策ではありません。 このプログラムは、メッセージの内容を検査して類縁性を認識するように設計する必要があります。 そして、類縁性の認識に基づいて、ワークロード管理ユーティリティーにすべての関連メッセージを同じキュー・マネージャーに転送させる必要があります。

Multi 均一クラスターの構成

均一クラスターによって、スケーリングと可用性を重視したアプリケーションを設計することができます。 また均一クラスターは、均一クラスター内のどのキュー・マネージャーにも接続できます。

始める前に

クラスタリングの概要については、クラスターを参照してください。 均一クラスターの概要については、410 ページの『均等クラスターについて』を参照してください。

このタスクについて

均一クラスターでは、キュー・マネージャー同士の通信やキュー間のワークロード・バランシングのために、IBM MQ クラスタリングを使用します。 ただし、通常の IBM MQ クラスターとは以下の点が違います。

- 通常、均一クラスターはクラスター内のキュー・マネージャーの数が少なめです。 均一クラスターでは、キュー・マネージャーの数が 10 を超えないようにします。
- クラスターの各メンバーの構成がほぼ同じです。
- 通常、クラスターを使用するのは 1 つのアプリケーションか、相互に関連する 1 群のアプリケーションです。
- クラスターに接続するアプリケーション・インスタンスの数をキュー・マネージャーの数以上にします。

自動構成および自動クラスタリング・サポートを使用することにより、均一クラスターの作成と、その後その均一クラスターのメンバー間で構成を同一に保つ作業の両方を単純化できます。

手順

- 均一クラスターについて
- 均一クラスターの作成
- 均一クラスターの作成
- 均一クラスターからのキュー・マネージャーの中断

Multi 均等クラスターについて

均一クラスター・デプロイメントの目的は、スケーリングと可用性を重視したアプリケーションを設計して、均一クラスター内のどのキュー・マネージャーにも接続できるようにすることです。 そうすれば、特定のキュー・マネージャーに依存することがなくなり、可用性が向上します。メッセージ・トラフィックのワークロード・バランシングの強化も可能です。 z/OS 均一クラスターは、IBM MQ for z/OS では使用できません。キュー共有グループによって、均一クラスターの多くの機能を実行できます。

均一クラスターとは、可用性の高い水平スケーリングの小規模なキュー・マネージャー・コレクションを組み込んだ IBM MQ クラスターの 1 つのパターンです。 これらのキュー・マネージャーはほぼ同じ構成なので、アプリケーションはこれらのキュー・マネージャーを 1 つのグループと見なして対話できます。 これにより、キュー・マネージャー間でアプリケーション・インスタンスを均等に分散させる処理が自動的に行われるので、クラスター内の各キュー・マネージャーの有効活用が容易になります。

それぞれ独立した相互接続のキュー・マネージャーを作成して管理するには管理者の側に手動ステップが必要になりますが、均一クラスターによってそうした手動ステップをいくらか削減できます。 クライアント接続のロジックの一部がクライアント側からキュー・マネージャー側に移動します。アプリケーション・アクティビティーのレベルに関する情報がクライアント側に通知され、どのキュー・マネージャーに接続するかについての決定で活用されます。

自動構成および自動クラスタリング・サポートを使用することにより、均一クラスターの最初の作成と、その後その均一クラスターのメンバー間で構成を同一に保つ作業の両方を単純化できます。 この機能を使用する場合は、1 つの構成ファイルでクラスターを記述し、別の構成ファイルでその均一クラスター内のすべてのキュー・マネージャーに適用する MQSC 構成を表します。 キュー・マネージャーを再始動するたびに、構成が再適用され、クラスターが自動的に形成されます。 この機能の使用法について詳しくは、425 ページの『均一クラスターの作成』を参照してください。

均一クラスターのメリットをフルに活用するには、多数の同一インスタンスに各アプリケーションをスケーリングすることも必要です。可能なら、インスタンスの数を最低でもキュー・マネージャーと同数にします。もっと多くできれば、それに越したことはありません。

どんなサイズの IBM MQ クラスターにも、さまざまな機能があります。

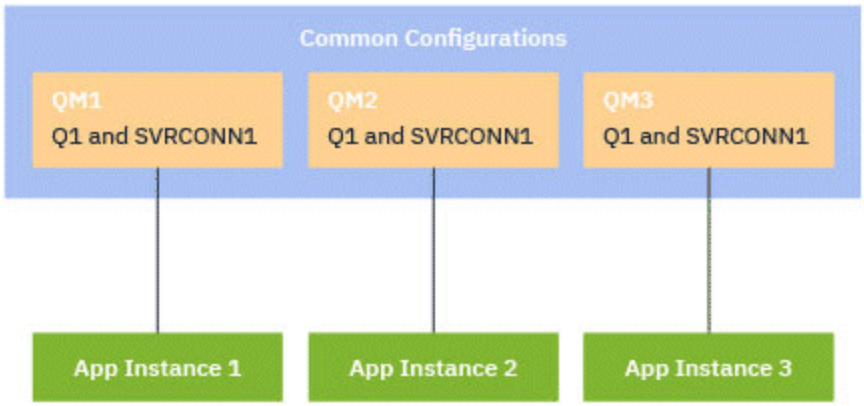
- すべてのクラスタリング・リソースのディレクトリー (クラスター内のどのメンバーからも検出が可能)
- 自動チャネル作成/接続
- 多数の同一キューによる水平スケーリング (メッセージ・ワークロード・バランシングの使用)
- 可用性に基づく動的メッセージ・ルーティング

均一クラスターでは、キュー・マネージャー同士の通信やキュー間のワークロード・バランシングのために、IBM MQ クラスタリングを使用します。 ただし、通常の IBM MQ クラスターとは以下の点が違います。

- 通常、均一クラスターはクラスター内のキュー・マネージャーの数が少なめです。 均一クラスターでは、キュー・マネージャーの数が 10 を超えないようにします。
- クラスターの各メンバーの構成がほぼ同じです。
- 通常、クラスターを使用するのは 1 つのアプリケーションか、相互に関連する 1 群のアプリケーションです。
- クラスターに接続するアプリケーション・インスタンスの数をキュー・マネージャーの数以上にします。

均一クラスター・パターンでは、クラスター内のすべてのキュー・マネージャーが同じメッセージング・サービスを提供します。 例えば、すべてのクラスター・メンバーに同じローカル・キューが定義されるように構成して、クライアント・アプリケーションからクラスター内のどのメンバーにも接続できるようにすることができます。 また、同じサーバー接続チャネルを定義することもできます。場合によっては、同じ権限レコードやチャネル認証ルールなども定義できます。 ただし、クラスターの各メンバーのオブジェクトや構成にいくらか違いがあるのは構いません。 例えば、一部のアプリケーションでは、キュー・マネージャーに接続している間に、一時動的キューを作成するかもしれません。 あるいは、一定の時間をかけて構成の更新 (証明書の新規作成や更新) を各メンバーに展開していくこともあり得ます。 通常の IBM MQ クラスターと同様に、キュー・マネージャーのうちの二つは、フル・リポジトリー・キュー・マネージャーにするための追加設定が必要です。

各キュー・マネージャーを同じ構成にした環境の図を以下に示します。 `Q1` という同じキューと、`SVRCONN1` という同じサーバー接続チャネルが定義されています。



同じサーバー接続チャネル名になっている複数のキュー・マネージャーが 1 つのクライアント・チャネル定義テーブル (CCDT) を使用する場合は、IBM MQ 9.1.2 で導入された新しいフォーマットの CCDT が必要です。 44 ページの『JSON 形式の CCDT の構成』を参照してください。

アプリケーション名とアプリケーション・インスタンス

アプリケーション名は、 **DISPLAY CONN(*) TYPE CONN** コマンドの APPLTAG 属性として表示されます。IBM MQ 9.1.2 以降では、アプリケーション名が設定される方法が 変更されています。

アプリケーションのインスタンスは、そのアプリケーションに対する 1 つの実行単位を提供する、密接に関連した一連の接続を表します。 通常これは、単一のオペレーティング・システム・プロセスで、複数のスレッドおよび関連する IBM MQ 接続を持つことができます。

アプリケーション名およびアプリケーション・インスタンスについて詳しくは、 アプリケーション開発の概念を参照してください。

再接続可能クライアント

ワークロードの均等な分散のために移動できるのは、再接続可能なクライアントです。それに対し、再接続不可能なクライアントは、その定義からして、別のキュー・マネージャーに再接続できません。 しかし、それでも再接続不可能なクライアントを均一クラスターに接続するメリットはあります。例えば、クライ

アントがある種の永続的な状態になる場合や、他のメカニズムを使ってそれぞれのキュー・マネージャーでアプリケーションの各インスタンスが実行されるようにする場合です。

ローカル・バインド・アプリケーション

均一クラスターでは、IBM MQ アプリケーションがローカル・バインド・アプリケーションではなくクライアント・アプリケーションとして接続することが想定されています。 ローカル・バインド・アプリケーションからクラスター・メンバーに接続することは禁止されてはいませんが、ローカル・バインド・アプリケーションはクラスターの他のメンバーに接続できないため、均一クラスターによる均等なワークロード分散はできません。

関連タスク

サポートされるプログラミング言語でのアプリケーション名の指定

Multi 自動アプリケーション・バランシング

IBM MQ の均一クラスターによってクラスター内のアプリケーションの分散度を緊密に管理できるようになり、ランダム化や特定のキュー・マネージャーにアプリケーションを手動でピン留めする操作に依存しなくてよくなりました。その結果、自動アプリケーション・バランシングによるアプリケーションの分散度と可用性が大幅に高まっています。

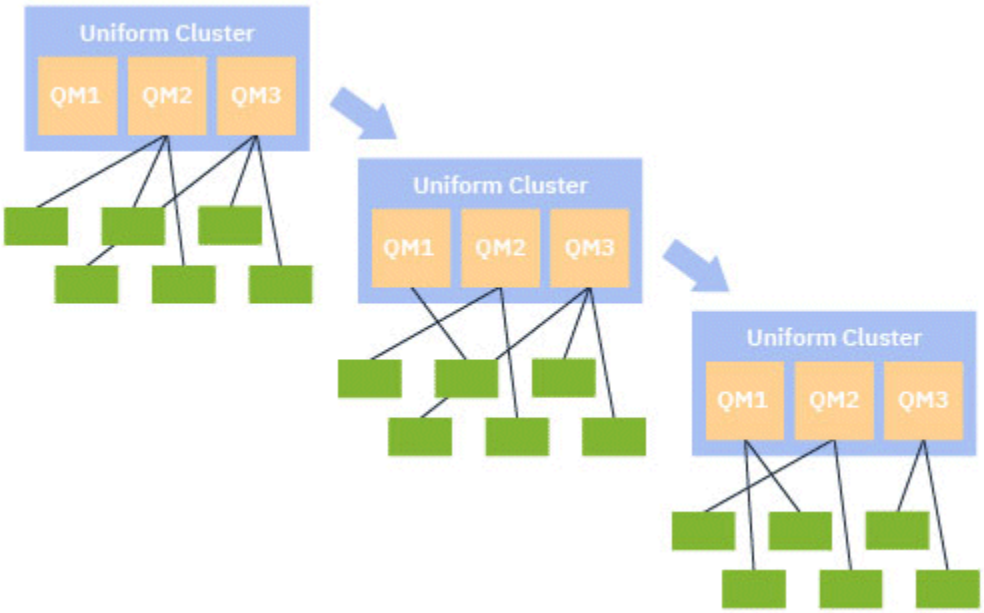
IBM MQ 9.2.0 以降、クラスター化されたキュー・マネージャーのセット全体にわたる自動バランシングは、C、JMS、IBM MQ .NET、XMS .NET で書かれたアプリケーションでサポートされます。

同じアプリケーションのインスタンスの数がキュー・マネージャーの数以上になっていると、均一クラスターによって、すべてのキュー・マネージャーが常に少なくとも 1 つのアプリケーション・インスタンスと接続するようになります。

アプリケーションでは、キュー・マネージャーへの特定のアフィニティーを除去できます。そして代わりにクライアント・チャネル定義テーブル (CCDT) を使用して、均一クラスター内のキュー・マネージャー・グループへの接続を安全にランダム化できます。 アプリケーションでこれを実現できる理由は次のとおりです。

- コンシューム側のアプリケーション・インスタンスが十分にあると、メッセージを処理するアプリケーション・インスタンスが常に存在することになります。
- 1 つのキュー・マネージャーを停止すると、接続済みのアプリケーション・インスタンスは、クラスター内の残りのキュー・マネージャーの間で均等に分散されます。
- 1 つのキュー・マネージャーを開始すると、クラスター内の他のキュー・マネージャーに接続していたアプリケーション・インスタンスが、その新しいキュー・マネージャーを含むキュー・マネージャー間で自動的に再バランシングされます。

その結果、均一クラスターでは、計画内の停止の場合も計画外の停止の場合も、継続的にアプリケーションの分散が最適化されてメッセージの処理が最大化されます。



自動バランシングのために、均一クラスター内のキュー・マネージャーは、 定期的に情報を共有します。 これは、トピックツリーの予約$SYS/MQ ブランチの下にあるシステムトピックのメタデータを公開することによって行われます。 均一クラスター内の各キュー・マネージャーは、他のキュー・マネージャーからパブリッシュされたメッセージにサブスクライブして、均一クラスター内のアプリケーションの状況を把握します。

キュー・マネージャーは、クラスター全体のクライアント・アプリケーションの分散状況をモニターします。 特定のキュー・マネージャーに接続しているアプリケーションの数がかなり少なく、そのキュー・マネージャーが、クラスター内のバランスが崩れていると判断した場合は、そのキュー・マネージャーがクラスター内の他のキュー・マネージャーの 1 つに対して、システム・トピックに要求をパブリッシュします。

メッセージを受け取ったターゲット・キュー・マネージャーは、クライアント・アプリケーションの 1 つに、要求元のキュー・マネージャーへのリダイレクトを要求します。 そのクライアント・アプリケーションはリダイレクト要求を受け取ると、接続を閉じて、要求元のキュー・マネージャーに再接続します。 この自動バランシング・メカニズムは、アプリケーションには認識されません。 詳細については、414 ページの『自動バランシングの機能』を参照してください。

均一クラスターは、接続済みのアプリケーションに関するメタデータを周期的に配布することによって、クライアント・アプリケーションとキュー・マネージャーの比率のバランスを幅広い範囲で時間の経過とともに改善していきます。 リダイレクト・イベントが短時間で連続して発生する事態を防ぐために、自動バランシング・アルゴリズムでは、リダイレクト要求の発生率が制限されています。

クラスター内の複数のキュー・マネージャーにわたってアプリケーションの現在の状態をモニターし、アプリケーション・インスタンスをモニターすることができます。 詳しくは、アプリケーション・バランシングのモニターを参照してください。 また、アプリケーション・バランシングに関する様々な問題を解決する方法については、アプリケーション・バランシングのトラブルシューティングを参照してください。

再バランシングが効果的なのは、接続時間の長いアプリケーションに限られます。 接続時間が短いクライアント・アプリケーション (例えば、定期的に異なるキュー・マネージャーに接続および切断するように作成されたクライアント・アプリケーション) の場合は、再接続不可能として構成するのが適切です。 これにより、そのアプリケーションは、キュー・マネージャーがバランシングを試行する対象のアプリケーションのセットから外れます。

関連概念

416 ページの『自動バランシングによる自動再接続の使用方法』
IBM MQ 9.2.0 以降、均一クラスターの自動バランシングで、IBM MQ の既存の自動再接続機能の機能強化を利用できるようになりました。

Multi 自動バランシングの機能

均一クラスターでは、アプリケーション名に基づいてクライアント接続をグループ化します。 同じアプリケーション名で均一クラスターのメンバーに接続する各アプリケーションは、同じアプリケーション名を使用するその他のアプリケーションと同等と見なされます。

自動バランシングにより、クラスターのメンバー間でアプリケーション・インスタンスが均等に分散されます。詳しくは、411 ページの『アプリケーション名とアプリケーション・インスタンス』を参照してください。 DISPLAY APSTATUS コマンドは、キュー・マネージャーまたは均一クラスターに接続されている 1 つ以上のアプリケーションおよびアプリケーション・インスタンスの状況を表示するために使用します。

例えば、保険要求アプリケーションのすべてのインスタンスが「INSURANCE.REQUESTS". このアプリケーションからの関連する接続は、必要に応じて自動的にインスタンスでグループ化され、インスタンス単位ですべてのバランシングが行われます。

そのアプリケーションの新しいインスタンスが均一クラスターのメンバーに接続する時には、自動バランシング・アルゴリズムによって、INSURANCE.REQUESTS インスタンス数の最も少ないキュー・マネージャーが判別され、いくつかの接続がそれらのキュー・マネージャーにリダイレクトされます。

自動バランシングが有効になるのは、以下の場合に限られます。

- チャネルの SHARECNV 値がゼロより大きい場合。
- 以下のいずれかが該当する。
  - クライアント・アプリケーションで MQCNO_RECONNECT が指定されている場合。
  - `mqclient.ini` ファイルで **Defrecon**=YES が指定されている場合。

注 : キュー・マネージャー・アフィニティーを持つアプリケーション (例えば、永続サブスクリプションやキューへの動的応答など) は、安全に再バランシングができないため、MQCNO_RECONNECT_QMGR を使用する方法をとるか、そうでない場合は再接続のオプションを使用しないでください。

クライアントが代替キュー・マネージャーにリダイレクトされると、通常どおりローカル・クライアント・チャネル定義テーブル (CCDT) を使用して、新しいターゲットの接続情報を見つけます。 したがって、自動バランスをスムーズかつ効率的に運用するために重要なのは、クライアントが、均一クラスターの各メンバーの項目が含まれている CCDT と、最初の接続時のバランシングに使用されたキュー・マネージャー・グループを使用することです。

複数の接続で同じサーバー接続名を使用するには、JSON 形式の CCDT を使用する方法が簡単です。 詳細については、44 ページの『JSON 形式の CCDT の構成』を参照してください。

関連概念

416 ページの『自動バランシングによる自動再接続の使用方法』
IBM MQ 9.2.0 以降、均一クラスターの自動バランシングで、IBM MQ の既存の自動再接続機能の機能強化を利用できるようになりました。

ALW *JMS* アプリケーションの自動バランシング

Jakarta Messaging 3.0 または Java Message Service 2.0 アプリケーションのバランスが自動的に取られると、 JMS アプリケーションが作成する IBM MQ 接続の基礎となるグループが一緒に移動されます。

V 9.3.0 IBM MQ 9.3.0 以降では、 ActivationSpecs の構成時に **dynamicallyBalanced** プロパティーを使用できます。 このプロパティーは、均一クラスター内のアプリケーション・バランシングの一部として、MDB が別のキュー・マネージャーからメッセージを受信するように要求できるかどうかを指定します。 詳しくは、インバウンド通信のリソース・アダプターの構成を参照してください。

JMS 接続を処理するために、均一クラスターにはアプリケーション・インスタンスという概念があります。 JMS では、アプリケーション・インスタンス は JMS 接続およびそれに関連するすべての JMS セッションとして定義されています。

JMS 接続に対応するクライアント接続に固有の接続タグが割り振られ、その JMS 接続によって作成された JMS セッションに対応するクライアント接続に同じタグが適用されます。

例えば、クライアント・アプリケーションのペアが、単一のアクティブ・キュー・マネージャー (キュー・マネージャー 1) を含んだ均一クラスターに対して JMS アプリケーションを実行しているとします。

- クライアント 1 は、アプリケーション名「App1」を設定する接続ファクトリーを作成し、JMS 接続と 3 つの JMS セッションを作成します。 クライアント 1 はキュー・マネージャー 1 に、それぞれが同じ接続タグを共有する 4 つのクライアント接続を作成します。これは「App1」という単一インスタンスとして扱われます。
- クライアント 2 は、アプリケーション名「App1」を設定する接続ファクトリーを作成し、JMS 接続と 2 つの JMS セッションを作成します。 クライアント 2 は、それぞれが同じ接続タグ (クライアント 1 に割り当てられたものとは別) を共有する 3 つのクライアント接続を作成します。これは別の「App1」単一インスタンスとして扱われます。
- したがって、キュー・マネージャーからは、2 つの「App1」が見えることになります。

自動バランシングが実行されると、アプリケーション・インスタンスは移動されます。 キュー・マネージャーは、アプリケーション・インスタンス (同じ接続タグを共用するクライアント接続のグループ) を選択し、そのインスタンスを別のキュー・マネージャーに移動するように要求します。 クライアント・コードは要求を受け取り、関連するすべての接続 (JMS 接続とその関連 JMS セッションに対応するもの) が新しいキュー・マネージャーに移動されるようにします。

例えば、前述の一連のアプリケーション・インスタンスを例に取って、新しいキュー・マネージャー (キュー・マネージャー 2) が均一クラスターに追加されたとします。

キュー・マネージャー 2 には作業が割り当てられていませんが、キュー・マネージャー 1 には「App1」のインスタンスが 2 つあります。そこでキュー・マネージャー 2 は、「App1」のインスタンスの 1 つをキュー・マネージャー 2 に転送するようキュー・マネージャー 1 に要求します。

キュー・マネージャー 1 は、移動する「App1」のインスタンスの 1 つを選択します。 この例では、クライアント 1 によって作成されたインスタンスを選択するものとします。

- キュー・マネージャー 1 は、「App1」のインスタンスを QM2 に移動する要求をクライアント 1 に送信します。
- クライアントは、キュー・マネージャー 1 への 4 つの既存のクライアント接続を閉じ、キュー・マネージャー 2 への 4 つの新規接続を作成します。
- JMS 接続とその JMS セッションは、処理の間の短い一時停止を除き、通常は中断されることはありません。

注 :

アプリケーション・インスタンスの移動時に特定の操作が進行中の場合、アプリケーションは JMS 例外を受け取ることがあります。

JMS 例外には、リンクされた IBM MQ 例外があります。そこから理由コードを取得して、失敗の原因を判別することができます。

予期される理由コードは、以下のとおりです。

MQRC_CALL_INTERRUPTED
これは、例えば持続メッセージ (JMS のデフォルト) が同期点の外側に置かれたものの、再接続によって操作が中断された場合に発生します。

MQRC_BACKED_OUT
これは、例えば、同期点の内側にメッセージを書き込もうとしたものの、再接続によってそれが中断された場合に発生します。

関連概念

414 ページの『自動バランシングの機能』
均一クラスターでは、アプリケーション名に基づいてクライアント接続をグループ化します。 同じアプリケーション名で均一クラスターのメンバーに接続する各アプリケーションは、同じアプリケーション名を使用するその他のアプリケーションと同等と見なされます。

416 ページの『自動バランシングによる自動再接続の使用方法』
IBM MQ 9.2.0 以降、均一クラスターの自動バランシングで、IBM MQ の既存の自動再接続機能の機能強化を利用できるようになりました。

Multi 自動バランシングによる自動再接続の使用方法

IBM MQ 9.2.0 以降、均一クラスターの自動バランシングで、IBM MQ の既存の自動再接続機能の機能強化を利用できるようになりました。

IBM MQ 9.2.0 より前のバージョンの IBM MQ では、自動再接続機能は、指定された接続の詳細 (通常は接続名リストまたはクライアント・チャネル定義テーブル (CCDT)) に基づいて、キュー・マネージャーのスタンバイ・インスタンスまたは別のキュー・マネージャーに自動的に再接続します。

IBM MQ クライアントは、状況によっては通知なしで再接続を実行します。アプリケーションはその再接続に気付きません。 どのキュー・マネージャーに再接続するかの決定は、接続名リストに載っている接続名の順序か、CCDT で設定されているワークロード・バランシングの構成に基づきます。

IBM MQ 9.2.0 以降、どのキュー・マネージャーに再接続するかに関するヒントが含まれた再接続要求をクライアントに送信することが可能になりました。 キュー・マネージャーで障害が発生する、または管理者が **endmqm -r** コマンドを実行するといった多くの再接続シナリオでは、ヒント情報の中にキュー・マネージャー名が含まれていません。自動再接続の動作はこれまでと同じです。

一方、均一クラスターを構成した場合は、クラスター内のバランス調整のために、自動アプリケーション・バランシング機能からクライアントに再接続要求が周期的に送信されます。 その場合は、均一クラスターが再接続のヒントの中でキュー・マネージャー名を指定するので、接続数の最も少ないキュー・マネージャーにクライアント接続が移動します。

自動バランシングを機能させるには、以下のようにすることが重要です。

- IBM MQ アプリケーションで CCDT を使用して接続情報を取得します。
- CCDT に均一クラスター内の各キュー・マネージャーの項目を組み込みます。

そのようにしない場合は、クラスターがすべてのクラスター・メンバーの間でアプリケーションのバランス調整を自動的に行うことは不可能です。

アプリケーションが IBM MQ 9.2.0 より前のバージョンの IBM MQ クライアントを使用しており、自動クライアント再接続をサポートするように構成されている場合、均一クラスターによって再接続ステップを実行するように要求が送信される可能性があります。

特定のキュー・マネージャーへの再接続がクライアントに要求されることはなく、他の再接続イベントの場合と同じ順序で再接続ロジックが実行されます。 クライアントを、各クラスター・メンバーの項目の重みが均等になっている CCDT を使用するように構成することで、IBM MQ 9.2.0 より前のクライアント・アプリケーションの均一クラスターでの均等な分散を実現できます。

アプリケーションは、追加のインスタンスを必要とするキュー・マネージャーに接続するまで再接続を数回試行することがあるので、クラスター内でアプリケーションを均等に分散させる方法としては、あまり効率がよくありません。 そうした環境では、自動バランシングに時間がかかる可能性があります。

自動クライアント再接続をサポートしていない IBM MQ クライアント

自動クライアント再接続をサポートしていないバージョンの IBM MQ クライアントをアプリケーションで使用している場合は、MQI 呼び出しから失敗の戻りコードを受け取ることがあります。

失敗を処理して手動で再接続を実行する設計が組み込まれていないアプリケーションについては、自動バランシングを無効にする必要があるかもしれません。

注 : 再接続可能として識別されるアプリケーション (つまり、有効な接続オプションで MQCNO_RECONNECT が設定されているアプリケーション) では、自動バランシングが有効になります。

関連タスク

425 ページの『均一クラスターの新規作成』
新しい均一クラスターの作成方法。

V 9.3.0 Multi 均一クラスターでのアプリケーションの再バランシングに影響を与える

自動アプリケーション・バランシング (均一クラスターの機能) を使用すると、アプリケーションの接続が、そのライフサイクルのあらゆる時点で、代替のキュー・マネージャーに移動するように要求されるようになります。

概要

IBM MQ 9.3.0 以降、バランシング・アルゴリズムは、アプリケーション・フローの中断を最小限にするために、アプリケーションの状態を自動的に考慮しようとします。 これは、アプリケーションのタイプ、またはこのアプリケーションによって実行される IBM MQ アクティビティーのパターンに関する追加情報を IBM MQ に提供することにより、特定のアプリケーションまたはアプリケーション・インスタンスに合うように調整できます。

通常は、クライアント・アプリケーションを開発またはデプロイする人が、このパターンを理解し、この情報をキュー・マネージャーに提供するのに最適な立場にある可能性が高いですが (柔軟で拡張性の高いクライアント・アプリケーションのデプロイ (Deploying flexible and scalable client applications) を参照)、管理者がこの調整を実行するか、追加的に実行する場合もあります。

キュー・マネージャーが妥当な期間内にアプリケーションの均等な分散を達成できない場合でも、他のキュー・マネージャーの IBM MQ フローの都合のよいタイミングを待つことなく、アプリケーションの接続が他のキュー・マネージャーに再バランシングされる可能性があることに注意してください。

これは、要件を満たすように調整することもできます。 アプリケーションの均等な分散を迅速に実現することがより重要な場合は、アプリケーションの再バランシングに適したタイミングを見つけるまでの待機時間を短縮するように製品を構成することができます。 あるいは、アプリケーションの中断を防ぐことがより重要な場合は、アプリケーションを移動するのに都合のよいタイミングを常に待つように製品を構成することができます。

概要情報の詳細については、柔軟で拡張性の高いクライアント・アプリケーションのデプロイ (Deploying flexible and scalable client applications) を参照してください。

.NET アプリケーションの場合の詳細については、420 ページの『.NET でのアプリケーションの再バランシングへの影響』を参照してください。

.XMS.NET アプリケーションの場合の詳細については、ConnectionFactory のプロパティーを参照してください。

V 9.3.4 JMS アプリケーションについて詳しくは、421 ページの『IBM MQ classes for JMS でのアプリケーションの再バランシングへの影響』 を参照してください。

デフォルトのアプリケーション・バランシング動作

デフォルトでは、キュー・マネージャーとのアプリケーションの相互作用のトランザクション/作業単位の状態は、すべてのアプリケーションについて考慮されます。

ローカル・トランザクションの場合、自動アプリケーション・バランシングは、現在トランザクションに関係しているアプリケーションへの再バランシング要求の発行を回避します。 これは、アプリケーションがバックアウトされた戻りコードを受け取る可能性を排除するわけではありませんが (構成された再バランシングのタイムアウトに達するか、実際の障害が発生したことによって、依然としてそのような戻りコードが発行される場合があるため)、通常はトランザクションの最中にアプリケーションが再接続を求められることがないことを意味します。

前のトランザクションの完了直後に新しいトランザクションを開始するアプリケーションの場合、再バランシングが完了するまで、新しいトランザクションでの最初の呼び出しに遅延が発生する可能性があります。 これにより、自動アプリケーション・バランシングは、引き続き均一クラスター内のキュ・ーマネージャー間でアプリケーションを均等に分散することができます。

実行時間の長いトランザクションを使用するアプリケーションがある場合は、リバランス・タイムアウトの値を増やすか、この制約を完全に無効にすることを検討してください。 MQI および .NET でこれを制御する方法、または同等のコード・レベルの「フォールト・トレランスおよびスケーラビリティーのためのクライアント・アプリケーションの設計」のリンクについては、418 ページの『バランシング動作の構成』を参照してください。

要求/応答バランシング

アプリケーション・タイプが **Request-Reply** として指定されている場合、アプリケーション・インスタンスが実行する PUT ごとに 1 つの応答 GET が予期されます。 アプリケーション・インスタンスに複数の

スレッドが含まれている場合、またはバッチでの要求と応答を処理する場合、複数の要求および応答をいつでもフライトに乗せることができます。

送信された要求の数が受信された応答の数と等しくなるか、またはタイムアウトのバックストップ値を超えるまで、アプリケーションは移動に適格と見なされません。

例外として、メッセージの有効期限が要求メッセージに対して構成されている場合があります。 応答は、要求メッセージの有効期限間隔内に受信する必要があり、すべての要求メッセージが期限切れになると、バランシング・アルゴリズムは、移動の対象となるインスタンスを考慮する前に、追加の応答を待機しなくなります。

未処理の要求が複数ある場合は、送信された要求メッセージのうち最新の有効期限のみが考慮されます。 意味のある有効期限値が使用されている場合は、予期される要求/応答の有効期限期間を短くしないように、アプリケーションの **Timeout** バランシング・パラメーターを、少なくとも送信されたメッセージの有効期限と同じ値に構成する必要があります。

先行するパターンは、未処理の要求がない期間があることを予期するアプリケーションにのみ適しています。 常にメッセージを送受信する複雑なマルチスレッド・アプリケーションは、このパターンの下でリバランスに適格にならない可能性があります。

注 :

- 特定の要求と応答を相関させる試みは行われないため、フライト・メッセージのバッチ内で以前の応答が期限切れになると、アプリケーションは、最新の要求の有効期限が切れるまで待機してから、バランスを取ることができます。
- 特に、同様の理由から、無制限の有効期限とメッセージの有効期限を組み合わせる場合には注意が必要です。

  有効期限が制限された要求メッセージが未解決であり、新規メッセージが無制限の有効期限付きで送信される場合、バランシング・アルゴリズムでは無制限の有効期限タイムアウトは考慮され ません 。これにより、現在の最新の有効期限時刻が引き続き考慮されます。

  そうしないと、以前の応答が有効期限切れになったため、アプリケーションが移動できなくなります。 同様に、無制限の有効期限切れタイムアウト応答が未解決の場合でも、有効期限切れの要求が後で送信されると、待ち時間は最長の (限定された) 有効期限に削減されます。

  一般的に、バランスの取れたアプリケーションで有効期限切れと非期限切れの両方の要求メッセージを送信する単一のアプリケーション・インスタンスを避ける必要があります。 これは、開発者または管理者が、正確な追跡または定義を行うことがより困難になるためです。
- 送信側アプリケーションによって指定された有効期限時刻のみ (例えば、MQI では MQMD.**Expiry** の値) 応答を待機する時間を決定する際に考慮されます。 例えば、この値に対する後続の変更では、CAPEXPRY の使用は、待機時間には影響しません。

バランシング動作の構成

IBM MQ がいつアプリケーションを再バランシングするかに明確に影響を与えるために、特定のクライアント・アプリケーション環境では、使用中のメッセージング・パターンに関する情報を接続時に提供することができます。

この情報は、バランシング・オプションと呼ばれる新しい構造で提供されます。

MQI の場合は、419 ページの『MQI を使用したバランシング動作の構成』を参照してください。

この構造に相当する .NET クライアントについては、420 ページの『.NET でのアプリケーションの再バランシングへの影響』を参照してください。

V 9.3.4 これらのオプションを設定する JMS アプローチについて詳しくは、421 ページの『IBM MQ classes for JMS でのアプリケーションの再バランシングへの影響』 を参照してください。

他のクライアント環境では、現在、接続時にこの構造を提供することはサポートされていません。

V 9.3.0 Multi *MQI* を使用したバランシング動作の構成

IBM MQ がいつアプリケーションを再バランシングするかに明確に影響を与えるために、特定のクライアント・アプリケーション環境では、使用中のメッセージング・パターンに関する情報を接続時に提供することができます。

MQI では、平衡化オプション構造体は ムクブノ と呼ばれます。

プログラムに バランシング・オプション が指定されていない場合、サポート・クライアントは、クライアント・アプリケーションとともにデプロイされる `client.ini` ファイルの Application スタンザ または ApplicationDefaults スタンザ からこの情報を派生させます。

注 : これらのスタンザは同じですが、`Application` バージョンには、これらのオプションが適用されるアプリケーションを識別するための **Name** フィールドが含まれています。

いずれかの形式のスタンザが指定されている場合は、すべてのフィールドが存在する必要があります。ただし、 **BalanceOptions** は例外で、明示的に設定されていない場合は none であると想定されます。

提供するオプションの優先順位は、次のとおりです:

1. MQBNO 構造体は、CONNX でアプリケーションによって提供され、全体として使用されます
2. または、 `Application` という名前のマッチング・スタンザが存在する場合は、それを生成するためにのみ使用されます。
3. または、 `ApplicationDefaults` スタンザが存在する場合は、このスタンザのみを使用して生成されます。
4. または、この接続に対する MQBNO フローはありません。

MQBNO 構造体または `client.ini` ファイルのいずれかから、以下の三つのキーに情報を提供することができます:

1. **ApplicationType** またはアプリケーションのパターン。

   フィールドは、アプリケーションが参加する IBM MQ アクティビティの一般的なパターン IBM MQ を示します。

   以下の三つのタイプのアプリケーションがサポートされています:

   単純
   417 ページの『デフォルトのアプリケーション・バランシング動作』で説明されているデフォルト以外の特定の規則は適用されません。

   要求/応答
   各 MQPUT 呼び出しの後で、対応する MQGET 呼び出しが応答メッセージに対して予期されています。 詳しくは、417 ページの『要求/応答バランシング』を参照してください。

   管理対象クライアント
   再平衡化要求は常に、クライアントに即時にディスパッチされます。これにより、例えば、JEE リソース・アダプターがこの方法で登録されるなど、適切な時点で再平衡化が行われます。
2. **Timeout** 。これを過ぎると、リバランスによってアプリケーション・アクティビティーが中断されます。
3. 特定の **BalanceOptions**

アプリケーションが再平衡化される可能性がある場合の例

例 1

メッセージを同期点で書き込み、MQCMIT 呼び出しを発行してメッセージのバッチをコミットするアプリケーションを作成しました。 MQCMIT 呼び出しが完了すると、アプリケーションは新しい同期点の下にメッセージを書き込みます。

推奨される IBM MQ 構成
十分なデフォルト・オプション

結果
構成された数のトランザクションが満たされると、MQCMIT 呼び出しが成功した(または失敗した)後に、アプリケーション・インスタンスが移動されます。

デフォルトでは、メッセージのバッチが 10 秒を超える場合は、リバランスが要求された場合にロールバックされる可能性があります。 トランザクションが定期的にこの制限を超え、これを許可する必要がある場合は、 **Timeout** を適切に拡張できます。

例 2

一つのメッセージをクラスター・キュー・インスタンスに書き込むアプリケーションが作成され、その要求を処理した後に、別のアプリケーションがメッセージを含むローカル一時動的キューに応答します。 要求がローカル・キューから破壊的に読み込まれた場合、アプリケーションは次の要求メッセージを出します。

推奨される IBM MQ 構成
種類を MQBNO_BALTYPE_REQREP に設定します

結果
アプリケーションが MQGET 呼び出しを完了すると、アプリケーション・インスタンスが移動されます。この時点で、アプリケーション・インスタンスは別のキュー・マネージャーに移動します。 後続の MQPUT 呼び出しは、新しいキュー・マネージャー上で実行されます。

MQBNO
ApplicationType

V 9.3.0 Multi *.NET* でのアプリケーションの再バランシングへの影響

IBM MQ 9.3.0 以降、MQQueueManager クラスを使用してキュー・マネージャーに接続するときに、アプリケーションからハッシュ・テーブルを使用してバランシング・オプションのプロパティーを設定できる定数が追加されました。

以下の定数は、.NET でアプリケーション・バランシングに影響を与えるために使用する定数です。

アプリケーション・タイプの再バランシング
バランシング・アクションのタイプ。定数 **MQC.BALANCING_APPLICATION_TYPE_PROPERTY** で表されています。

- MQBNO 構造体の **ApplicationType** フィールドを設定するには、このプロパティーを使用する必要があります。

  整数型の値を設定する必要があり、可能な値は以下のとおりです:

  MQC.BALANCING_APPLICATION_TYPE_SIMPLE
  単純なバランシング；416 ページの『均一クラスターでのアプリケーションの再バランシングに影響を与える』で説明されているものに加えて、特定のルールは適用されません。 これがデフォルト値です。

  MQC.BALANCING_APPLICATION_TYPE_REQUEST_REPLY
  要求/応答バランシング。各 **MQPUT** 呼び出しの後に、応答メッセージに対して一致する **MQGET** 呼び出しが必要です。 バランシングは、そのようなメッセージが受信されるか、または要求メッセージ **EXPIRY** が超過するまで遅延されます。

  アプリケーションによって再接続が有効になっており、このプロパティーが設定されていない場合、**MQC.BALANCING_APPLICATION_TYPE_SIMPLE** が使用されます。

再平衡化オプション
発行側アプリケーションによって設定されるバランシング・オプション。定数 **MQC.BALANCING_OPTIONS_PROPERTY** で表されています。

- MQBNO 構造体の **BalanceOptions** フィールドを設定するには、このプロパティーを使用する必要があります。

  整数型の値を設定する必要があり、可能な値は以下のとおりです:

MQC.BALANCING_OPTIONS_NONE
オプションは設定されていません。 これはデフォルト値です

MQC.BALANCING_OPTIONS_IGNORE_TRANSACTIONS
このオプションを設定すると、トランザクションの途中であっても、アプリケーションをリバランスすることができます。

アプリケーションによって再接続が有効になっており、このプロパティーが設定されていない場合は、**MQC.BALANCING_OPTIONS_NONE** が使用されます。

再平衡化タイムアウト
再バランシングによってアプリケーション・アクティビティーが中断されるまでのタイムアウト。定数 **MQC.BALANCING_TIMEOUT_PROPERTY** で表されます。

- プロパティーを使用して、MQBNO 構造体のタイムアウトフィールドを設定する必要があります。

  整数型の値を設定する必要があり、可能な値は以下のとおりです:

  MQC.BALANCING_TIMEOUT_AS_DEFAULT
  設定されたデフォルトのタイムアウト値。 これはデフォルト値です

  MQC.BALANCING_TIMEOUT_IMMEDIATE
  即時タイムアウトが発生

  MQC.BALANCING_TIMEOUT_NEVER
  タイムアウト発生なし

  注 : 定義された値から 1 つの値、または 0 から 999999999 秒の値を指定する必要があります。

柔軟でスケーラブルなクライアントアプリケーションの導入
MQBNO

V 9.3.4 Multi *IBM MQ classes for JMS* でのアプリケーションの再バランシングへの影響
IBM MQ 9.3.4 以降、 **ConnectionFactory** でバランシング・オプション・プロパティーを設定するための追加の定数を使用できるようになりました。 これらの定数は、**WMQ_PROVIDER_VERSION** が 7 に設定されている場合にのみ適用されます。 均一クラスター内の Request_reply アプリケーションでは、応答が欠落する可能性を考慮する必要があります。

- 421 ページの『使用可能な定数』.
- 422 ページの『REQUEST_REPLY アプリケーションのバランシング時にメッセージが失われる可能性がある』.

使用可能な定数

以下の定数は、IBM MQ classes for JMS でアプリケーション・バランシングに影響を与えるために使用する定数です。

アプリケーション・タイプの再バランシング
バランシング・アクションのタイプ。定数 **WMQConstants.WMQ_BALANCING_APPLICATION_TYPE** で表されています。

- MQBNO 構造の **ApplicationType** フィールドを設定するには、このプロパティーを使用する必要があります。

  整数型の値を設定する必要があります。 指定できる値は以下のとおりです。

  WMQConstants.WMQ_BALANCING_APPLICATION_TYPE_SIMPLE (デフォルト)
  単純なバランシング；416 ページの『均一クラスターでのアプリケーションの再バランシングに影響を与える』で説明されているものに加えて、特定のルールは適用されません。

  WMQConstants.WMQ_BALANCING_APPLICATION_TYPE_REQUEST_REPLY
  要求/応答バランシング。各 **MQPUT** 呼び出しの後に、応答メッセージに対して一致する **MQGET** 呼び出しが必要です。 バランシングは、そのようなメッセージが受信されるか、または要求メッセージ **EXPIRY** が超過するまで遅延されます。

アプリケーションによって再接続が有効になっていて、このプロパティーが設定されていない場合は、 **WMQConstants.WMQ_BALANCING_APPLICATION_TYPE_SIMPLE** が使用されます。

再平衡化オプション
発行側アプリケーションによって設定されるバランシング・オプション。定数 **WMQConstants.WMQ_BALANCING_OPTIONS** で表されています。

- MQBNO 構造の **BalanceOptions** フィールドを設定するには、このプロパティーを使用する必要があります。

  整数型の値を設定する必要があります。 指定できる値は以下のとおりです。

  WMQConstants.WMQ_BALANCING_OPTIONS_NONE (デフォルト)
  オプションは設定されていません。

  WMQConstants.WMQ_BALANCING_OPTIONS_IGNORE_TRANSACTIONS
  このオプションを設定すると、トランザクションの途中であっても、アプリケーションをリバランスすることができます。

  アプリケーションによって再接続が有効になっていて、このプロパティーが設定されていない場合は、 **WMQConstants.WMQ_BALANCING_OPTIONS_NONE** が使用されます。

再平衡化タイムアウト
リバランスによってアプリケーション・アクティビティーが中断されるまでのタイムアウト。定数 **WMQConstants.WMQ_BALANCING_TIMEOUT** で表されます。

- MQBNO 構造の **Timeout** フィールドを設定するには、このプロパティーを使用する必要があります。

  整数型の値を設定する必要があります。 指定できる値は以下のとおりです。

  WMQConstants.WMQ_BALANCING_TIMEOUT_AS_DEFAULT (デフォルト)
  設定されたデフォルトのタイムアウト値。 デフォルトでは、この値は 10 秒です。

  WMQConstants.WMQ_BALANCING_TIMEOUT_IMMEDIATE
  即時タイムアウトが発生します。

  WMQConstants.WMQ_BALANCING_TIMEOUT_NEVER
  タイムアウトは起こりません。

  1 から 999999999 までの値
  これは秒単位の値を表します。

  注 : 定義された値から 1 つの値、または 0 から 999999999 秒の値を指定する必要があります。

これらのプロパティーは、 JMSAdmin または IBM MQ Explorer インターフェースを使用する接続ファクトリーの JNDI 表現でも設定できます。

REQUEST_REPLY アプリケーションのバランシング時にメッセージが失われる可能性がある

IBM MQ classes for JMS (および IBM MQ classes for Jakarta Messaging) では、要求/応答機能は、要求メッセージに **JMSReplyTo** プロパティーを設定することによって実装されます。このプロパティーは、応答を送信するかどうかを決定するために応答アプリケーションによって使用されます。 JMS の用語では、 **JMSReplyTo** プロパティーは **Destination** です。

これが IBM MQ 操作に変換されると、 **JMSReplyTo** プロパティーは、特定のキュー・マネージャー上のキューを識別する完全修飾キュー URI として送信されます。

再接続のバランシングの処理は非同期であるため、**JMSReplyTo** プロパティーが完全修飾 URI に変換された後、要求メッセージが要求キューに書き込まれる前に、再接続が開始されることがあります。 このような状況では、応答側アプリケーションは、元のキュー・マネージャーの元の応答キューに応答を送信する可能性がありますが、要求側アプリケーションは現在、新しいキュー・マネージャーで応答を待機している可能性があります。

したがって、均一クラスター内の Request_reply アプリケーションでは、応答が欠落する可能性を考慮する必要があります。

柔軟でスケーラブルなクライアントアプリケーションの導入
MQBNO - バランシング・オプション

Multi 均一クラスターの制限事項と注意点

均一クラスターを構成する時に考慮すべき制限事項など。

注 : 均一クラスターを構成する場合の一般的な要件については、 425 ページの『均一クラスターの新規作成』も参照してください。

キュー・マネージャー間の均一性の重要性

デフォルトでは、 reconnectable として自身を宣言するアプリケーションは、いつでも均一クラスター内の代替キュー・マネージャーにリバランスされる可能性があります。 つまり、そのようなアプリケーションが必要とするキュー、トピック、または権限レコードなどのリソースは、均一クラスター内のすべてのキュー・マネージャーで宣言する必要があります。

キュー・マネージャー構成の整合性はポリシングされません。 クラスターのメンバーを同じような構成にするかどうかは、システム管理者にかかっています。

ただし、「始動時に MQSC スクリプトから自動構成」 機能を使用して、クラスターのオブジェクトを定義する MQSC スクリプトを共有することにより、整合性を保つことができます。これにより、すべてのスクリプトの定義が同じになります。 詳しくは、425 ページの『均一クラスターの新規作成』を参照してください。

この均一性は、クラスターのフル・リポジトリー・キュー・マネージャーにまで及びます。 従来の IBM MQ クラスターの場合、多くの場合、完全リポジトリーをスタンドアロン・システムに分離することがベスト・プラクティスと見なされますが、均一クラスターでは、完全リポジトリーがクラスターに完全に参加し、他のノードとともにアプリケーション・ワークロードを処理します。

均一クラスターと従来の IBM MQ クラスターのオーバーラップ

均一クラスターのキュー・マネージャーは 1 つの均一クラスターにしか参加できません。標準の IBM MQ クラスターにはいくつでも参加してそのメンバーになれます。 均一クラスターは、より広いクラスター内の単一のキュー・マネージャーとして機能すると考えると役立つ場合があります。

次の考慮事項に注意してください。

- 完全なリポジトリとして動作するユニフォームクラスターキューマネージャは、ユニフォームクラスタ自身のための完全なリポジトリでなければなりません。
- 同様に、ユニフォームクラスタのメンバーでありながら、より広い従来の IBM MQ クラスタにも属しているかもしれない部分的なリポジトリキューマネージャは、ユニフォームクラスタの外ではリポジトリとして使用することはできません。

  詳細については、完全なリポジトリを保持するクラスタ・キュー・マネージャの選択方法を参照してください。

その理由は、従来の IBM MQ クラスタとユニフォームクラスタの組み合わせのための完全なリポジトリであるキューマネージャは、ユニフォームクラスタのメンバー間でクラスタキャッシュに保持されるデータの発散を促し、したがってユニフォームクラスタの機能を意図したとおりに使用することに反するからです。

単一のフル・リポジトリー・キュー・マネージャーを均一クラスターに置き換えるには、フル・リポジトリーをそのキュー・マネージャーで進行中のアプリケーション作業から分離し、アプリケーション作業のみを均一クラスターに移動します。

均一クラスターの自動定義を使用している場合、他のクラスターで使用するためにクラスター・チャネルを共有することはできません。つまり、**CLUSTER** 属性を自動クラスターに設定し、**CLUSNL** 属性を空にする必要があります。

アプリケーション・バランシングの考慮事項

特に以下の状況では、アプリケーション・インスタンスのバランスが常に均等であるとは限りません。

- アプリケーション・インスタンスの数がクラスター内のキュー・マネージャーの数より少ない場合。
- クライアント・アプリケーションがクラスターに接続した後、またはクラスターから離脱した後の短期間。

クライアント・アプリケーションのリバランスが過度に頻繁に行われないようにするために (特に、アプリケーション接続が頻繁に行われる場合)、均一クラスターがクライアント・アプリケーションにリバランスを要求する頻度に制限が設定されます。 接続または切断のアクティビティーが高い期間が経過すると、残りのアプリケーション・インスタンスが均一クラスター全体で均等にバランスが取られるまでに数分かかる場合があります。

詳しくは、 アプリケーション・バランシングのトラブルシューティングを参照してください。

アプリケーション親和性

すべてのアプリケーションが均一クラスター間の自動リバランスに適しているわけではありません。 **MQCNO_RECONNECT** を指定するアプリケーションのみがリバランスされます。 特定のキュー・マネージャーに対してアフィニティーを持つアプリケーションは、 **MQCNO_NO_RECONNECT** オプションまたは **MQCNO_RECONNECT_Q_MGR** のいずれかを指定する必要があります。 後者では HA フェイルオーバーが許可されますが、リバランスは許可されません。

キュー・マネージャーへの暗黙的な親和性を作成するアプリケーションの例を以下に示します。

- 永続サブスクリプションを作成するアプリケーション。
- 例えば、応答メッセージを受信するために、永続動的キューを作成するアプリケーション。
- 厳密なメッセージ順序付けを必要とするアプリケーション、またはシーケンス内のすべてのメッセージを必要とするアプリケーションは、同じアプリケーション・インスタンスまたはその両方によって処理されます。

これらのアプリケーションは、 **MQCNO_RECONNECT** ではなく、 **MQCNO_NO_RECONNECT** または **MQCNO_RECONNECT_Q_MGR** オプションを指定する必要があります。

詳しくは、 再接続オプションを参照してください。

メッセージの可用性

アプリケーション・バランシングでは、障害が発生したキュー・マネージャーまたは一時的に使用不可になったキュー・マネージャーの間で接続のバランスを再調整できますが、均一クラスターでは、メンバー間でメッセージ・データを複製しません。 データの可用性のために、ノードに障害が発生した場合は、均一クラスターの各メンバーも高可用性になるように構成する必要があります。 多くのデータ複製と高可用性ソリューションが使用可能であり、サービスとデータの可用性を最大限に高めるために均一クラスターと組み合わせることができます。以下に例を示します。

- コンテナー・オーケストレーションによって自動的に再始動されるコンテナー・インスタンスをサポートする複製ストレージ。 詳しくは、 単一の回復力を持つキュー・マネージャーを参照してください。
- RDQM キュー・マネージャー。 詳しくは、RDQM 高可用性を参照してください。
- 複数インスタンス・キュー・マネージャー。 詳しくは、複数インスタンスのキュー・マネージャーを参照してください。
- ネイティブ HA。 詳しくは、ネイティブ HA を参照してください。
- IBM MQ Appliance HA。 詳しくは、高可用性を参照してください。

均一クラスターのスケーラビリティーとパフォーマンス

均一クラスター内のキュー・マネージャー間でアプリケーションの状態をより緊密に統合および共有できるようにするには、従来の IBM MQ クラスターよりも高いレベルの相互通信が必要です。 したがって、単一の均一クラスターで多数のキュー・マネージャーにスケーリングすることは推奨されません。これは、追加の通信がパフォーマンスに悪影響を及ぼすためです。

パフォーマンスと管理の両方の理由から、均一クラスターは、多数の関連アプリケーションにメッセージングを提供する単一の従来型キュー・マネージャーとして機能するが、企業全体で唯一のメッセージング・

サービスではないと考えることをお勧めします。 このパターンでは、多数のクライアント・アプリケーション接続をサポートするには、通常、少数 (最大 10 個) のキュー・マネージャーで十分です。 アプリケーション・バランシングにより、少数のキュー・マネージャー (例えば、3 つのキュー・マネージャー) から始めて、キュー・マネージャーを追加することでスケールアップすることが簡単になります。

重要 : 推奨特性のないクラスター (特に多数のキュー・マネージャーがあるクラスター) で均一クラスターの動作を有効にすると、パフォーマンスに深刻な影響が出る可能性があります。

関連概念

412 ページの『自動アプリケーション・バランシング』

IBM MQ の均一クラスターによってクラスター内のアプリケーションの分散度を緊密に管理できるようになり、ランダム化や特定のキュー・マネージャーにアプリケーションを手動でピン留めする操作に依存しなくてよくなりました。その結果、自動アプリケーション・バランシングによるアプリケーションの分散度と可用性が大幅に高まっています。

ALW 均一クラスターの作成

自動構成および自動クラスタリング・サポートを使用することにより、均一クラスターの最初の作成と、その後その均一クラスターのメンバー間で構成を同一に保つ作業の両方を単純化できます。

始める前に

均一クラスターを作成する前に、423 ページの『均一クラスターの制限事項と注意点』をお読みください。

このタスクについて

特定の IBM MQ クラスターを均一クラスターとして扱うことを指定するには、 `qm.ini` ファイルで、**Type**=一様 および **ClusterName**=< 均一クラスター名>以上の AutoCluster のセクションを指定します。

自動クラスター作成を使用して、基本となる IBM MQ クラスターを同じ `.ini` スタンザを介して構成することができます。 この自動クラスター・サポートを使用してクラスターをセットアップする場合は、クラスターとその完全リポジトリーを記述した構成ファイルを提供します。

開始するキュー・マネージャーが完全リポジトリーの 1 つとしてリストされている場合、それは自動的に完全リポジトリーになります。 同様に、クラスター受信側チャネルが定義されている場合、完全リポジトリー (複数の場合もある) に対するクラスター送信側チャネルが自動的に定義されます。

手順

均一クラスターを必要とする追加の機能を活用するには、以下のいずれかのステップを実行する必要があります。

- 410 ページの『均等クラスターについて』に記されているパターンに合致する均一クラスターへの既存のクラスターの変換。
- これを目的とした均一クラスターの新規作成。

ALW 均一クラスターの新規作成

新しい均一クラスターの作成方法。

手順

1. クラスターの完全リポジトリーについて記述するファイルを作成します。

   どのクラスターでも、2 つの完全リポジトリーが、そのクラスターについての情報の中央ストアとして機能します。

   つまり、そのクラスター内の 2 つの完全リポジトリーの名前と接続名を記述する必要があります。

   注 : すべての作成作業 (キュー・マネージャーの作成を含む) は、この記述の後に行います。 以下のプロセスには、これらのキュー・マネージャーの作成が含まれています。

例として、キュー・マネージャー・メンバー QMA、QMB、QMC、および QMD を含む UNICLUS という名前の均一クラスターをセットアップするとします。 この例では、QMA と QMB が完全リポジトリーで、QMC と QMD が部分リポジトリーです。 構成ファイル uniclus.ini を以下に示します:

```
AutoCluster:
   Repository2Conname=QMA.dnsname(1414)
   Repository2Name=QMA
   Repository1Conname=QMB.dnsname(1414)
   Repository1Name=QMB
   ClusterName=UNICLUS
   Type=Uniform
```

RepositoryNConname フィールドは、他のクラスター・メンバーがクラスター送信側 (CLUSSDR) を定義するための *conname* 属性として使用され、複数インスタンス・キュー・マネージャーの接続リストにすることができ、オプションでポートを含めることもできます。

2. すべてのクラスター・メンバーに適用される MQSC 定義を含むサンプル構成ファイル(uniclus.mqsc)を作成します。

   このファイルには、クラスター受信側チャネル (CLUSRCVR) の定義の 1 行を必ず含める必要があります。この行には、自動クラスター名 (通常は +AUTOCL+ 挿入を使用) の CLUSTER 属性と、+QMNAME+ 挿入を含むチャネル名を指定します。

   これは、均一クラスターの他のメンバーが各キュー・マネージャーに接続する方法を記述するもので、他のキュー・マネージャーへの接続方法のテンプレートとしても使用されます。 定義の例を以下に示します。

```
define channel('+AUTOCL+_+QMNAME+') chltype(clusrcvr) trptype(tcp)
conname(+CONNAME+) cluster('+AUTOCL+') replace
```

   自動クラスターが構成されている場合、均一クラスター内のすべてのキュー・マネージャーでクラスター受信側チャネルの定義を同一にするために、この定義の CLUSTER、CONNAME、および CHANNEL のフィールドにいくつかの追加の挿入を使用することができます。 これには、以下が含まれます。

   +AUTOCL+
   自動クラスター名

   + キュー名 +
   作成するキュー・マネージャーの名前

   +CONNAME+
   接続名パラメーター・ストリングで使用するために、**-iv** パラメーターを使用するか、または Variables qm.ini スタンザ内で、キュー・マネージャーの作成時に定義される変数。 変数名には任意の値を指定できます。

   チャネル名は 20 文字に制限されているので、挿入の使用時と挿入の置換後のいずれにおいても、値がこの制限内に収まっている必要があります。 例えば、ファイルは次のようになります。

```
*##########################################################################
* Compulsory section for all uniform cluster queue managers
*##########################################################################
define channel('+AUTOCL+_+QMNAME+') chltype(clusrcvr) trptype(tcp) conname(+CONNAME+)
cluster('+AUTOCL+') replace
*
*##########################################################################
* Configuration for all queue managers
*##########################################################################
define QL(APPQ) maxdepth(99999999) replace
define QL(APPQ2) maxdepth(99999999) replace
define channel(CLIENTCHL) chltype(svrconn) trptype(tcp) replace
```

3. これら 2 つのファイルを、均一クラスター・メンバーをホストする各マシンで利用できるようにします。

   例えば、/shared/uniclus.ini と/shared/uniclus.mqsc などです。

4. これらのマシンごとに、キュー・マネージャーを作成します。

   コマンド・ラインで、以下を指定します。

   a. 予期されるポートでのリスナーの開始の要求

b. 自動クラスター・セットアップ・ファイル (uniclus.ini) を指す自動 INI 構成 (**-ii**) の要求

c. 均一クラスターの CLUSRCVR 定義が含まれている MQSC 構成ファイルを指す自動 MQSC 構成 (**-ic**) の要求。

d. このキュー・マネージャーの CONNAME。

QMA のホストで、次のようにします。

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMA.dnsname(1414) QMA
strmqm QMA
```

均一クラスター内の各キュー・マネージャーは、ほとんど同じコマンド・ラインで作成されます。 均一クラスターでは、完全リポジトリーと部分リポジトリーの違いはすべて自動的に処理されます。

QMB のホストで、次のようにします。

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMB.dnsname(1414) QMB
strmqm QMB
```

QMC のホストで、次のようにします。

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMC.dnsname(1414) QMC
strmqm QMC
```

QMD のホストで、次のようにします。

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMD.dnsname(1414) QMD
strmqm QMD
```

自動処理の内容:

キュー・マネージャーが開始されると、`uniclus.ini` ファイルからの定義が `qm.ini` ファイルに適用されます。 詳しくは、99 ページの『始動時の qm.ini の自動構成』 を参照してください。 これにより、**AutoCluster** 定義が `qm.ini` ファイルに追加されます。

キュー・マネージャーが **AutoCluster** スタンザで完全リポジトリーの 1 つとして指定されている場合は、MQSC コマンド ALTER QMGR REPOS (**ClusterName**) を発行する場合と同様に、完全リポジトリーに自動的に変換されます。そうでない場合は、MQSC コマンド ALTER QMGR REPOS ('') を発行する場合と同様に、部分リポジトリーに変換されます。

自動クラスターのクラスター受信側チャネルの定義が処理されると、このキュー・マネージャーから **AutoCluster** スタンザ内のすべての完全リポジトリー (これが完全リポジトリーの 1 つである場合はローカル・キュー・マネージャーを除く) に対してクラスター送信側チャネルが定義されます。 これらの送信側チャネルは、定義されているローカル・クラスター受信側からすべての共通チャネル属性を継承します。

重要 : チャネルはそれ以上の手操作による介入なしで作成されますが、これらのチャネルは管理チャネル・オブジェクトであり、他のチャネル定義の場合と同様に表示および管理することができます。 これらのオブジェクトを「自動定義された」クラスター送信側チャネルと混同しないでください。「自動定義された」クラスター送信側チャネルは、メッセージ・トラフィックを経路指定するために、クラスターによって一時的に作成され、オンデマンドで作成されます。

次のタスク

均一クラスターのセットアップの検証

ClusterName パラメーターが正しく設定されていて、キュー・マネージャーが指定されたクラスターのメンバーである場合、クラスターが均一クラスターとして識別されるようになったことを確認するメッセージ AMQ9883 が発行されます。

その後、均一クラスターの機能 (自動アプリケーション・バランシングなど) を利用できます。 キュー・マネージャーの開始時にこのパラメーターが設定されていても、その名前が有効な IBM MQ クラスター名でなければ、その名前は無視され、エラー・メッセージ AMQ9882 が生成されます。

その名前が有効なクラスター名でも、識別されたクラスターにクラスター・チャネルがなければ、警告メッセージ AMQ9881 がキュー・マネージャーのエラー・ログに書き込まれ、管理者がその状況を確認して修正できるようになります。

自動クラスターのセットアップの検証

自動クラスター・サポートを使用して均一クラスターをセットアップした場合は、以下の runmqsc コマンドを使用して、完全リポジトリーとして指定したキュー・マネージャーがそのとおり正しく構成されているか確認できます。

```
QMA:
     1 : dis qmgr repos
AMQ8408I: Display Queue Manager details.
   QMNAME(QMA)                             REPOS(UNICLUS)
```

一方、部分リポジトリーは、リポジトリーとしては構成されていません。

```
QMC:
     1 : dis qmgr repos
AMQ8408I: Display Queue Manager details.
   QMNAME(QMC)                             REPOS( )
```

さらに、各キュー・マネージャーから他の完全リポジトリーに対してクラスター送信側チャネル (CLUSSDR) が構成されていることを、構成 MQSC ファイルのチャネル名を使用して確認できます。

```
QMA:
     1 : dis chl(UNICLUS*) conname
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMA)                    CHLTYPE(CLUSRCVR)
   CONNAME(QMA.dnsname(1414))
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMB)                    CHLTYPE(CLUSSDR)
   CONNAME(QMB.dnsname(1414))

QMC:
     1 : dis chl(UNICLUS*) conname
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMA)                    CHLTYPE(CLUSSDR)
   CONNAME(QMA.dnsname(1414))
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMB)                    CHLTYPE(CLUSSDR)
   CONNAME(QMB.dnsname(1414))
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMC)                    CHLTYPE(CLUSRCVR)
   CONNAME(QMC.dnsname(1414))
```

関連概念

410 ページの『均等クラスターについて』
均一クラスター・デプロイメントの目的は、スケーリングと可用性を重視したアプリケーションを設計して、均一クラスター内のどのキュー・マネージャーにも接続できるようにすることです。 そうすれば、特定のキュー・マネージャーに依存することがなくなり、可用性が向上します。メッセージ・トラフィックのワークロード・バランシングの強化も可能です。 z/OS 均一クラスターは、IBM MQ for z/OS では使用できません。キュー共有グループによって、均一クラスターの多くの機能を実行できます。

423 ページの『均一クラスターの制限事項と注意点』
均一クラスターを構成する時に考慮すべき制限事項など。

ALW 均一クラスターへの既存のクラスターの変換

この手順を使用して既存のクラスターを均一クラスターに変換できます。

このタスクについて

既存のクラスターを均一クラスターに変換する場合は、キュー・マネージャー間のアプリケーション・バランシングのサポートに必要な定義がすべてのクラスター・メンバーに存在することを確認する必要があります。

手順

1. すべてのキュー・マネージャーで、IBM MQ のパブリッシュ/サブスクライブを有効にします。リモートの (クラスター化した) パブリッシュ/サブスクライブも有効にします。

   これは、均一クラスターが機能するための前提条件になります。したがって、必ずキュー・マネージャーの PSMODE 属性と PSCLUS 属性の両方を、デフォルト値の ENABLED に設定してください。
2. qm.ini ファイルの **AutoCluster** セクションを、クラスター・チャネルなどの MQSC オブジェクト定義で使用されている IBM MQ クラスターの名前に追加します。

   例えば、クラスタの名前が UNICLUS の場合、qm.ini ファイル内で AutoCluster stanza を次のように追加または変更します:

   ```
   AutoCluster:
         ClusterName=UNICLUS
       Type=Uniform
   ```
3. キュー・マネージャーを再始動して、新しい設定を適用します。
4. 始動時から、均一クラスターのすべてのメンバーに同じ構成が適用されるようにするためのメカニズムとして、自動構成を使用することを検討してください。

   詳しくは、始動時の MQSC スクリプトからの自動構成を参照してください。

関連概念

410 ページの『均等クラスターについて』
均一クラスター・デプロイメントの目的は、スケーリングと可用性を重視したアプリケーションを設計して、均一クラスター内のどのキュー・マネージャーにも接続できるようにすることです。 そうすれば、特定のキュー・マネージャーに依存することがなくなり、可用性が向上します。メッセージ・トラフィックのワークロード・バランシングの強化も可能です。 z/OS 均一クラスターは、IBM MQ for z/OS では使用できません。キュー共有グループによって、均一クラスターの多くの機能を実行できます。

423 ページの『均一クラスターの制限事項と注意点』
均一クラスターを構成する時に考慮すべき制限事項など。

Multi 自動クラスター構成の使用

qm.ini コンフィギュレーション情報を変更することで、IBM MQ 自動コンフィギュレーションを有効にするように設定します。

注 : AutoCluster スタンザは、均一クラスターでのみ使用できます。

構成するスタンザ

以下のスタンザを変更できます。

AutoConfig
qm.ini ファイルで定義されています。 キュー・マネージャーは始動すると、適用する自動構成ファイルを識別します。

均一クラスターを使用している場合は、このメカニズムを使用して同一のクラスター構成を配布する必要があります。

AutoCluster
qm.ini ファイルで定義されています。 クラスターが自動クラスターのメンバーであるかどうかの識別をキュー・マネージャーが開始するときに使用されます。クラスターの完全リポジトリーを識別することができます。

変数
qm.ini ファイルで定義されています。 いくつかのキュー・マネージャー変数が含まれます。

AutoConfig スタンザの属性

AutoConfig スタンザには、次の 2 つの属性を指定できます。

MQSCConfig=*<Path>*
このパスは、すべてのキュー・マネージャーの開始時に、すべてのファイル*.mqsc マネージャーに適用される絶対ファイル・パスまたはディレクトリーへのパスのいずれかです。

詳しくは、開始時の MQSC スクリプトからの自動構成を参照してください。

IniConfig=*<Path>*
このパスは、完全ファイル・パスか、またはすべてのファイル*.iniqm.ini に適用されるディレクトリーへのパス(すべてのキュー・マネージャーの開始時)になります。

詳細については、99 ページの『始動時の qm.ini の自動構成』を参照してください。

これらの属性は、均一クラスターのセットアップの一部として頻繁に使用されます。 詳細については、425 ページの『均一クラスターの新規作成』を参照してください。

スタンザの例:

```
AutoConfig:
MQSCConfig=C:\MQ_Configuration\uniclus.mqsc
IniConfig=C:\MQ_Configuration\uniclus.ini
```

AutoCluster スタンザの属性

次の属性は、AutoCluster スタンザの必須の属性です。

Type=*Uniform*
自動クラスターのタイプを指定します。有効なオプションは、均一クラスターを表す *Uniform* のみです。

ClusterName=*<String>*
クラスターの名前 (自動クラスター名)。

上記の属性が存在すると、均一クラスターのアプリケーション・バランシングが有効になります。 詳細については、412 ページの『自動アプリケーション・バランシング』 を参照してください。

また、このスタンザでクラスターを記述すると、クラスターのセットアップを簡素に実行できます。 詳細については、425 ページの『均一クラスターの新規作成』を参照してください。 これを使用する場合は、2 つのキュー・マネージャーの名前を指定し、それらの接続名をこの自動クラスターの完全リポジトリーに指定することができます。

次の属性は、AutoCluster スタンザのオプションの属性ですが、指定はペアで行う必要があります。

RepositoryName1 =< ストリング>
これは、自動クラスター内の最初の完全リポジトリーのキュー・マネージャー名です。 この名前は、このキュー・マネージャーの名前でも、別のキュー・マネージャーの名前でも構いません。

Repository1Conname=< 接続名ストリング>
これは、自動クラスターのメンバーがこのキュー・マネージャーに接続する方法を示す接続名 (CONNAME) の値です。

さらに、クラスターの 2 番目の完全リポジトリーを指定することができます。

Repository2Name=< ストリング>
Repository2Conname=< 接続名ストリング>

スタンザの例:

```
AutoCluster:
   Repository2Conname=myFR1.hostname(1414)
   Repository2Name=QMFR1
```

```
   Repository1Conname= myFR2.hostname(1414)
   Repository1Name=QMFR2
   ClusterName=UNICLUS
   Type=Uniform
```

Variables スタンザの属性

この属性フィールドでは、attribute=value のペアが有効です。 これらは、キュー・マネージャーの作成時に **crtmqm** コマンドの **-iv** コマンド行オプションを使用して指定できます。

Variables スタンザにリストされる属性は、クラスター受信側チャネルの CONNAME およびチャネル名 MQSC フィールドの自動クラスター構成時に使用できます。

均一クラスターからのキュー・マネージャーの中断

均一クラスターの通常操作時には、再接続可能なクライアント・アプリケーション・インスタンスは、クラスター内の任意のキュー・マネージャーにいつでも自動的にリバランスされる可能性があります。 アプリケーションが一定期間、特定のキュー・マネージャーに接続できないようにするには (例えば、保守操作時や問題判別時など)、SUSPEND QMGR コマンドを使用します。

コマンド SUSPEND QMGR CLUSTER(*uniform cluster name*) を発行します。

均一クラスターでは、IBM MQ クラスターから中断する通常の影響に加えて、SUSPEND コマンドを使用すると、再接続可能なアプリケーションがこのキュー・マネージャーにリバランスされないようにすることもできます。

そのような既存のキュー・マネージャーへの接続は、コマンドの発行時に、クラスター内の他の使用可能なキュー・マネージャーに即時に再バランシングされます。

注 :

- キュー・マネージャーがクラスターから中断されると、 DIS APSTATUS はそれらを ACTIVE (NO) として表示します。ただし、ローカル・キュー・マネージャーは例外で、ローカル・キュー・マネージャー自体の状況項目には常に ACTIVE (YES) と表示されます。
- 均一クラスター内のすべてのキュー・マネージャーが中断されている場合、アプリケーションは 1 つ以上の中断状態のキュー・マネージャーに接続されたままになります。

キュー・マネージャーに新しい接続が追加されないようにするには、例えば、以下の **runmqsc** コマンドを発行して、クライアント・アプリケーションが使用するサーバー接続チャネル (複数可) を停止する必要があります。

```
STOP CHANNEL(svrconn channel name)
```

これらのチャネルが保守期間中に必要な管理アプリケーションに接続するために使用される場合などには、このコマンドを使用できない可能性があります。 このため、中断状態のキュー・マネージャーは、接続されている再接続可能アプリケーションを定期的に検査します。

再接続可能なアプリケーションが存在する場合、それらのアプリケーションはクラスター内の他の使用可能なキュー・マネージャーにリバランスされます。 これで、中断状態のキュー・マネージャーで保守を実行できるようになります。

注 : 移動できないと見なされるアプリケーションは、最初のコマンドおよび後続の再スキャンのいずれの影響も受けず、中断状態のキュー・マネージャーに接続されたままになります。詳細については、MOVCOUNT を参照してください。

中断状態のキュー・マネージャーを再開するには、次のようにします。

1. 必要に応じて、次のコマンドを発行して、サーバー接続チャネルを開始し、新規アプリケーション接続の受け入れを再開します。

   ```
   START CHANNEL(svrconn channel name)
   ```

2. 次の **runmqsc** コマンドを発行します。

```
RESUME QMGR CLUSTER(uniform cluster name)
```

キュー・マネージャーは、残りの均一クラスターとの通信を再開し、バランスを復元する必要がある場合は、再接続可能なクライアント・アプリケーション・インスタンスがこのキュー・マネージャーにリダイレクトされます。

パブリッシュ/サブスクライブ・メッセージングの構成

キュー型パブリッシュ/サブスクライブの開始、停止、および状況の表示を行えます。 ストリームの追加および除去、ブローカー階層に対するキュー・マネージャーの追加および削除も行えます。

手順

- キュー型パブリッシュ/サブスクライブの制御の詳細については、以下のサブトピックを参照してください。
  - 432 ページの『キューに入れられたパブリッシュ/サブスクライブのメッセージ属性の設定』
  - 433 ページの『キューに入れられたパブリッシュ/サブスクライブの開始』
  - 434 ページの『キューに入れられたパブリッシュ/サブスクライブの停止』
  - 434 ページの『ストリームの追加』
  - 435 ページの『ストリームの削除』
  - 436 ページの『サブスクリプション・ポイントの追加』
  - 445 ページの『パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合』

キューに入れられたパブリッシュ/サブスクライブのメッセージ属性の設定

パブリッシュ/サブスクライブ・メッセージの一部の属性の振る舞いは、キュー・マネージャー属性によって制御します。 その他の属性は、qm.ini ファイルの *Broker* スタンザで制御します。

このタスクについて

以下のパブリッシュ/サブスクライブ属性を設定することができます。 詳しくは、キュー・マネージャーのパラメーターを参照してください

表 27. パブリッシュ/サブスクライブ構成パラメーター

説明	MQSC パラメーター名
コマンド・メッセージの再試行回数	**PSRTYCNT**
配信不能なコマンド入力メッセージの廃棄	**PSNPMSG**
配信不能なコマンド応答メッセージの後の振る舞い	**PSNPRES**
同期点でのコマンド・メッセージの処理	**PSSYNCPT**

Broker スタンザは、以下の構成設定を管理するために使用します。

- `PersistentPublishRetry=yes | force`

  Yes を指定した場合に、キュー型パブリッシュ/サブスクライブ・インターフェース経由の持続メッセージのパブリケーションに失敗すると、否定応答を要求していなければ、パブリッシュ操作が再試行されます。

  否定応答メッセージを要求していた場合は、否定応答が送信され、それ以上の再試行は行われません。

  Force を指定した場合に、キュー型パブリッシュ/サブスクライブ・インターフェース経由の持続メッセージのパブリケーションに失敗すると、正常に処理されるまでパブリッシュ操作が再試行されます。 否定応答は送信されません。

- NonPersistentPublishRetry=yes | force

  Yes を指定した場合に、キュー型パブリッシュ/サブスクライブ・インターフェース経由の非持続メッセージのパブリケーションに失敗すると、否定応答を要求していなければ、パブリッシュ操作が再試行されます。

  否定応答メッセージを要求していた場合は、否定応答が送信され、それ以上の再試行は行われません。

  Force を指定した場合に、キュー型パブリッシュ/サブスクライブ・インターフェース経由の非持続メッセージのパブリケーションに失敗すると、正常に処理されるまでパブリッシュ操作が再試行されます。否定応答は送信されません。

  注: 非持続メッセージに対してこの機能を有効にする場合は、NonPersistentPublishRetry 値の設定に加えて、必ず、キュー・マネージャー属性 **PSSYNCPT** も Yes に設定してください。

  同期点では STREAM キューの **MQGET** が実行されるようになったため、これを行うと、非持続パブリケーションの処理のパフォーマンスにも影響を与える可能性があります。

- PublishBatchSize=*number*

  ブローカーは通常、同期点でパブリッシュ・メッセージを処理します。 パブリケーションごとに個別にコミットすると効率が落ちる可能性があり、状況によっては、ブローカーは複数のパブリッシュ・メッセージを 1 つの作業単位として処理できます。 このパラメーターでは、1 つの作業単位で処理できるパブリッシュ・メッセージの最大数を指定します。

  PublishBatchSize のデフォルト値は 5 です。

- PublishBatchInterval=*number*

  ブローカーは通常、同期点でパブリッシュ・メッセージを処理します。 パブリケーションごとに個別にコミットすると効率が落ちる可能性があり、状況によっては、ブローカーは複数のパブリッシュ・メッセージを 1 つの作業単位として処理できます。 このパラメーターでは、バッチに含まれている最初のメッセージから、その同じバッチに含まれている後続パブリケーションまでの間の最大時間をミリ秒単位で指定します。

  メッセージがすぐに使用可能であれば、バッチの間隔を 0 にすると、最大 PublishBatchSize 個のメッセージを処理できます。

  PublishBatchInterval のデフォルト値はゼロです。

手順

IBM MQ エクスプローラー、プログラマブル・コマンド、または **runmqsc** コマンドを使用して、パブリッシュ/サブスクライブの振る舞いを制御するキュー・マネージャー属性を変更します。

例

ALTER QMGR PSNPRES(SAFE)

キューに入れられたパブリッシュ/サブスクライブの開始

キュー・マネージャーの PSMODE 属性を設定することで、キューに入れられたパブリッシュ/サブスクライブを開始します。

始める前に

パブリッシュ/サブスクライブの次の 3 つのモードを理解するには、PSMODE の説明を参照してください。

- COMPAT
- disabled
- 有効

このタスクについて

QMGR PSMODE 属性を設定して、キューに入れられたパブリッシュ/サブスクライブ・インターフェース (ブローカーとも呼ばれる) またはパブリッシュ/サブスクライブ・エンジン (バージョン 7 パブリッシュ/サブスクライブとも呼ばれる) のいずれか、あるいはその両方を開始します。 キューに入れられたパブリッシュ/サブスクライブを開始するには、PSMODE を ENABLED に設定する必要があります。 デフォルトは ENABLED です。

手順

キューに入れられたパブリッシュ/サブスクライブ・インターフェースがまだ使用可能になっていなければ、IBM MQ Explorer または **runmqsc** コマンドを使用して使用可能にします。

例

```
ALTER QMGR PSMODE (ENABLED)
```

次のタスク

IBM MQ は、キューに入れられたパブリッシュ/サブスクライブ・コマンドおよびパブリッシュ/サブスクライブの Message Queue Interface (MQI) 呼び出しを処理します。

キューに入れられたパブリッシュ/サブスクライブの停止

キュー・マネージャーの PSMODE 属性を設定することで、キューに入れられたパブリッシュ/サブスクライブを停止します。

始める前に

パブリッシュ/サブスクライブの次の 3 つのモードを理解するには、PSMODE の説明を参照してください。

- COMPAT
- DISABLED
- ENABLED

このタスクについて

QMGR PSMODE 属性を設定して、キューに入れられたパブリッシュ/サブスクライブ・インターフェース (ブローカーとも呼ばれる) またはパブリッシュ/サブスクライブ・エンジン (バージョン 7 パブリッシュ/サブスクライブとも呼ばれる) のいずれか、あるいはその両方を停止します。 キューに入れられたパブリッシュ/サブスクライブを停止するには、PSMODE を COMPAT に設定する必要があります。 キューに入れられたパブリッシュ/サブスクライブ・エンジンを完全に停止するには、PSMODE を DISABLED に設定します。

手順

キューに入れられたパブリッシュ/サブスクライブ・インターフェースを無効にするには、IBM MQ Explorer または **runmqsc** コマンドを使用します。

例

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

ストリームの追加

アプリケーション間でデータを分離できるようにするため、または IBM MQ パブリッシュ/サブスクライブ階層の相互協調処理を可能にするために、ストリームを手動で追加することができます。

始める前に

パブリッシュ/サブスクライブ・ストリームの動作についてよく理解しておいてください。 ストリームおよびトピックを参照します。

このタスクについて

PCF コマンド、**runmqsc**、または IBM MQ Explorer を使用して、以下のステップを実行します。

注: ステップ 1 と 2 は、任意の順序で実行できます。 ステップ 3 だけは、ステップ 1 と 2 の両方が完了した後に実行してください。

手順

1. 以前のバージョンの IBM MQ のストリームと同じ名前でローカル・キューを定義します。
2. IBM MQ の旧バージョンのストリームと同じ名前でローカル・トピックを定義します。
3. キューの名前を名前リスト SYSTEM.QPUBSUB.QUEUE.NAMELIST に追加します
4. パブリッシュ/サブスクライブ階層にある、新しいバージョンの IBM MQ のすべてのキュー・マネージャーについて、この手順を繰り返します。

'Sport' の追加中

ストリーム 'Sport'を共有する例では、前のバージョンのキュー・マネージャーと後のバージョンの IBM MQ キュー・マネージャーが同じパブリッシュ/サブスクライブ階層で動作しています。 以前のバージョンのキュー・マネージャーは、 'Sport'というストリームを共有します。 この例では、 'Sport'という新しいバージョンのキュー・マネージャー上にキューとトピックを作成する方法を示します。このトピック・ストリングは、 'Sport' 前のバージョンのキュー・マネージャー・ストリームと共有されます 'Sport'。

新しいバージョンのキュー・マネージャー・パブリッシュ・アプリケーションで、トピック・ストリング 'Soccer/Results'を使用してトピック 'Sport'にパブリッシュすると、結果としてトピック・ストリング 'Sport/Soccer/Results'が作成されます。 新しいバージョンのキュー・マネージャーでは、トピック・ストリング 'Soccer/Results' を持つトピック 'Sport'へのサブスクライバーがパブリケーションを受け取ります。

以前のバージョンのキュー・マネージャーでは、トピック・ストリング 'Soccer/Results' を持つストリーム 'Sport'へのサブスクライバーがパブリケーションを受け取ります。

```
runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
define qlocal('Sport')
     1 : define qlocal('Sport')
AMQ8006: IBM MQ queue created.
define topic('Sport') topicstr('Sport')
     2 : define topic('Sport') topicstr('Sport')
AMQ8690: IBM MQ topic created.
alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM',
'SYSTEM.BROKER.ADMIN.STREAM')
     3 : alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM',
'SYSTEM.BROKER.ADMIN.STREAM')
AMQ8551: IBM MQ namelist changed.
```

注: alter namelist コマンドには、名前リスト・オブジェクト内の既存の名前と、追加する新規の名前の両方を指定する必要があります。

次のタスク

ストリームに関する情報が、階層内の他のブローカーに渡されます。

階層内の各 IBM MQ キュー・マネージャーを手動で構成する必要があります。

ストリームの削除

IBM MQ キュー・マネージャーからストリームを削除できます。

始める前に

ストリームを削除する前に、そのストリームへのサブスクリプションが残っていないことを確認し、そのストリームを使用するすべてのアプリケーションを静止する必要があります。 削除されたストリームにパブリケーションが流れ続けた場合、システムをクリーンな処理状態に復元するために、多くの管理努力が必要となります。

手順

1. このストリームをホストする、すべての接続済みのブローカーを検索します。
2. すべてのブローカー上にある、このストリームへのすべてのサブスクリプションを取り消します。
3. 名前リスト SYSTEM.QPUBSUB.QUEUE.NAMELIST からキュー (ストリームと同じ名前を持つもの) を削除します。
4. ストリームと同じ名前を持つキューからのすべてのメッセージを削除または消去します。
5. ストリームと同じ名前を持つキューを削除します。
6. 関連するトピック・オブジェクトを削除します。

次のタスク

ストリームをホストしている他のすべての接続された IBM MQ キュー・マネージャーで、ステップ 3 から 5 を繰り返します。

サブスクリプション・ポイントの追加

IBM Integration Bus の旧バージョンから移行した既存のキュー型パブリッシュ/サブスクライブ・アプリケーションを、新規のサブスクリプション・ポイントで拡張する方法を示します。

始める前に

1. サブスクリプション・ポイントが SYSTEM.QPUBSUB.SUBPOINT.NAMELIST でまだ定義されていないことを確認してください。
2. サブスクリプション・ポイントと同じ名前を持つトピック・オブジェクトまたはトピック・ストリングが存在するかどうかを確認します。

このタスクについて

IBM MQ では、アプリケーションはサブスクリプション・ポイントを使用しませんが、サブスクリプション・ポイント移行メカニズムを使用して、サブスクリプション・ポイントを使用する既存のアプリケーションと相互運用することができます。

重要 : サブスクリプション・ポイント移行メカニズムは、IBM MQ 8.0 から削除されています。 既存のアプリケーションを移行する必要がある場合は、使用している製品のバージョン用の資料に示された手順を実行してから、最新バージョンに移行する必要があります。

IBM MQ のバージョン用に作成された統合パブリッシュ/サブスクライブ・アプリケーションを使用するために、サブスクリプション・ポイントを追加する必要はありません。

手順

1. サブスクリプション・ポイントの名前を SYSTEM.QPUBSUB.SUBPOINT.NAMELIST に追加します。
   - z/OS では、**NLTYPE** はデフォルトで NONE になっています。
   - 同じパブリッシュ/サブスクライブ・トポロジー内で接続されているすべてのキュー・マネージャーに対して、このステップを繰り返します。
2. サブスクリプション・ポイントの名前と一致するトピック・ストリングを持つトピック・オブジェクトを追加し、できればこれにサブスクリプション・ポイントの名前を付けます。

- サブスクリプション・ポイントがクラスター内にある場合は、トピック・オブジェクトをクラスター・トピック・ホスト上のクラスター・トピックとして追加します。
- サブスクリプション・ポイントの名前と同じトピック・ストリングを持つトピック・オブジェクトが存在している場合は、その既存のトピック・オブジェクトを使用します。 サブスクリプション・ポイントが既存のトピックを再利用した場合の結果について、理解しておく必要があります。 既存のトピックが既存のアプリケーションの一部である場合には、同じ名前を持つ 2 つのトピック間の衝突を解決する必要があります。
- サブスクリプション・ポイントと同じ名前を持つトピック・オブジェクトは存在しているものの、トピック・ストリングが異なる場合は、別の名前を持つトピックを作成します。

3. **Topic** の属性 WILDCARD を値 BLOCK に設定します。

   #または*のサブスクリプションをブロック化すると、ワイルドカード・サブスクリプションがサブスクリプション・ポイントに分離されます。ワイルドカードおよびサブスクリプション・ポイントを参照してください。
4. トピック・オブジェクトで、任意の必要な属性を設定します。

例

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

```
DEFINE TOPIC(USD) TOPICSTR(USD)
DEFINE TOPIC(GBP) TOPICSTR(GBP) WILDCARD(BLOCK)
ALTER NL(SYSTEM.QPUBSUB.SUBPOINT.NAMELIST) NAMES(SYSTEM.BROKER.DEFAULT.SUBPOINT, USD, GBP)
```

注 :

1. デフォルトのサブスクリプション・ポイントを、**ALTER** コマンドを使用して追加されたサブスクリプション・ポイントのリストに組み込みます。 **ALTER** は、名前リスト内の既存の名前を削除します。
2. 名前リストを変更する前に、トピックを削除します。 キュー・マネージャーが名前リストを検査するのは、キュー・マネージャーの開始時および名前リストの変更時のみです。

分散パブリッシュ/サブスクライブ・ネットワークの構成

分散パブリッシュ/サブスクライブ・トポロジーにまとめて接続されるキュー・マネージャーは、共通のフェデレーテッド・トピック・スペースを共有します。 1 つのキュー・マネージャー上で作成されるサブスクリプションは、トポロジー内の別のキュー・マネージャーに接続されたアプリケーションによってパブリッシュされるメッセージを受け取ることができます。

クラスターまたは階層の中で複数のキュー・マネージャーをまとめて接続することによって、作成されるトピック・スペースの範囲を制御できます。 パブリッシュ/サブスクライブ・クラスターでは、クラスター内のトピック・スペースの各ブランチで、トピック・オブジェクトを「クラスター化」する必要があります。 階層内では、階層の「親」を識別するように各キュー・マネージャーを構成する必要があります。

それぞれのパブリケーションおよびサブスクリプションをローカルまたはグローバルとして選択することにより、トポロジー内のパブリケーションおよびサブスクリプションのフローをさらに制御できます。 ローカル・パブリケーションおよびサブスクリプションは、パブリッシャーまたはサブスクライバーの接続先であるキュー・マネージャーを超えて伝搬されることはありません。

関連概念

分散パブリッシュ/サブスクライブのネットワーク

パブリケーション有効範囲

サブスクリプション有効範囲

トピック・スペース

関連タスク

クラスター・トピックの定義

パブリッシュ/サブスクライブ・クラスターの構成

キュー・マネージャーでトピックを定義します。 トピックをクラスター・トピックにするには、**CLUSTER** プロパティーを設定します。 このトピックのパブリケーションおよびサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティーを設定します。

始める前に

一部のクラスター構成では、直接ルーティング型パブリッシュ/サブスクライブのオーバーヘッドに対応できません。 この構成を使用する前に、パブリッシュ/サブスクライブ・クラスターの設計で詳述されている考慮事項およびオプションを調べてください。

クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。 この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

パブリッシュ/サブスクライブ・クラスターのルーティング: 動作に関する注意も参照してください。

シナリオ

- 313 ページの『クラスターにキュー・マネージャーを追加する』 で説明されているように、INVENTORY クラスターがセットアップされています。 これには 3 つのキュー・マネージャーが含まれており、LONDON と NEWYORK はどちらも完全リポジトリーを保持し、 PARIS は部分リポジトリーを保持します。

このタスクについて

クラスター内のキュー・マネージャーにトピックを定義する際に、トピックがクラスター・トピックであるかどうかを指定する必要があります。クラスター・トピックである場合、このトピックのパブリケーションおよびサブスクリプションに使用するクラスター内でのルーティングも指定する必要があります。 トピックをクラスター・トピックにするには、TOPIC オブジェクトの **CLUSTER** プロパティーをクラスターの名前を使用して構成します。 クラスター内の一つのキュー・マネージャーにクラスター・トピックを定義することによって、トピックをクラスター全体で使用できるようにします。クラスター内で使用するメッセージ・ルーティングを選択するには、TOPIC オブジェクトの **CLROUTE** プロパティーを以下のいずれかの値に設定します。

- **DIRECT**
- **TOPICHOST**

デフォルトでは、トピック・ルーティングは **DIRECT** です。 これは IBM MQ 8.0 より前の唯一のオプションでした。 直接経路指定されたクラスター・トピックをキュー・マネージャーで構成すると、クラスター内のすべてのキュー・マネージャーがクラスター内の他のすべてのキュー・マネージャーを認識するようになります。 各キュー・マネージャーは、パブリッシュ操作およびサブスクライブ操作を実行するときに、クラスター内の他のすべてのキュー・マネージャーに直接接続できます。 直接ルーティング型パブリッシュ/サブスクライブ・クラスターを参照してください。

IBM MQ 8.0 以降、代わりにトピック・ルーティングを **TOPICHOST** として構成できるようになりました。トピック・ホスト経路指定を使用すると、クラスター内のすべてのキュー・マネージャーは、経路指定されたトピック定義をホストするクラスター・キュー・マネージャー (つまり、トピック・オブジェクトを定義したキュー・マネージャー) を認識するようになります。 パブリッシュ操作およびサブスクライブ操作を行うとき、クラスター内のキュー・マネージャーは、それらのトピック・ホスト・キュー・マネージャーにのみ接続し、相互に直接接続されることはありません。 トピック・ホスト・キュー・マネージャーは、パブリケーションがパブリッシュされるキュー・マネージャーから、一致するサブスクリプションがあるキュー・マネージャーへのパブリケーションの経路指定を担当します。 トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターを参照してください。

注 : トピック オブジェクトが (**CLUSTER** プロパティを設定して) クラスタ化された後は、 **CLROUTE** プロパティの値を変更することはできません。 値を変更するには、その前にオブジェクトのクラスター化を解除 (**CLUSTER** を ' ' に設定) する必要があります。 トピックのクラスター化を解除すると、トピック定義はローカル・トピックに変換されます。これによって、パブリケーションがリモート・キュー・マネージャーのサブスクリプションに送信されない期間ができます。この変更を行う場合は、この点を考慮する必要があります。 別のキュー・マネージャーのクラスター・トピックと同じ名前で非クラスター・トピックを定

義する効果を参照してください。 クラスタリング中に **CLROUTE** プロパティの値を変更しようとすると、MQRCCF_CLROUTE_NOT_ALTERABLE 例外が発生します。

手順

1. トピックをホストするキュー・マネージャーを選択します。

   どのクラスター・キュー・マネージャーでもトピックをホストできます。 3 つのキュー・マネージャー (LONDON、NEWYORK、または PARIS) のいずれかを選択し、TOPIC オブジェクトのプロパティーを構成します。 直接ルーティングを使用する場合、どのキュー・マネージャーを選択しても操作に違いは生じません。 トピック・ホスト・ルーティングを使用する場合、選択したキュー・マネージャーには、パブリケーションのルーティングを行う追加の責任が生じます。 そのため、トピック・ホスト・ルーティングにおいては、性能の良いシステムにホストされる、ネットワーク接続の状態が良いキュー・マネージャーを選択してください。
2. キュー・マネージャーでトピックを定義します。

   トピックをクラスター・トピックにするには、トピックを定義するときにクラスター名を使用し、このトピックのパブリケーションおよびサブスクリプションに使用するルーティングを設定します。 例えば、LONDON キュー・マネージャー上に直接ルーティング・クラスター・トピックを作成するには、以下のようにトピックを作成します。

   ```
   DEFINE TOPIC(INVENTORY) TOPICSTR('/INVENTORY') CLUSTER(INVENTORY) CLROUTE(DIRECT)
   ```

   クラスター内の一つのキュー・マネージャーにクラスター・トピックを定義することによって、トピックをクラスター全体で使用できるようにします。

   CLROUTE の使用についての詳細情報は、DEFINE TOPIC (CLROUTE) および パブリッシュ/サブスクライブ・クラスターのルーティング: 動作に関する注意を参照してください。

タスクの結果

クラスターは、トピックのパブリケーションおよびサブスクリプションを受信する準備が整いました。

次のタスク

トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターを構成した場合には、このトピック用にたいてい 2 つ目のトピック・ホストを追加することになります。 442 ページの『トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加』を参照。

お客様の組織が地理的に分散しているなどの理由で、いくつかの別個のパブリッシュ/サブスクライブ・クラスターが存在する場合、一部のクラスター・トピックをすべてのクラスターに伝搬することもできます。 階層でクラスターに接続すると、これを行うことができます。 447 ページの『複数のクラスターのトピック・スペースの結合』を参照。 あるクラスターから別のクラスターへ流れるパブリケーションを制御することもできます。 448 ページの『複数のクラスター内でのトピック・スペースの結合および分離』を参照してください。

関連概念

パブリケーションとサブスクリプションの有効範囲の結合
IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合
キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。 パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

関連タスク

別のキュー・マネージャーへのクラスター・トピック定義の移動
トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時

間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加
トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。 この方式を利用して、アベイラビリティーとワークロード・バランシングを向上させることができます。 同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続
階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。 子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断
パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

パブリッシュ/サブスクライブ・クラスターの設計
分散パブリッシュ/サブスクライブの問題のトラブルシューティング
クラスター化されたパブリッシュ/サブスクライブの禁止

別のキュー・マネージャーへのクラスター・トピック定義の移動

トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

このタスクについて

1 つのクラスター内に、同じクラスター・トピック・オブジェクトの複数の定義を持つことができます。 これは、トピック・ホスト・ルーティング型クラスターにとっては通常の状態であり、直接ルーティング型クラスターにとっては異常な状態です。 詳しくは、同じ名前の複数のクラスター・トピック定義を参照してください。

パブリケーションのフローを中断せずに、クラスター内の別のキュー・マネージャーにクラスター・トピック定義を移動させるには、以下のステップを実行します。 この手順では、定義をキュー・マネージャー QM1 からキュー・マネージャー QM2 に移動します。

手順

1. クラスター・トピック定義の複製を QM2 上に作成します。

   直接ルーティング型の場合は、すべての属性を QM1 の定義と一致するように設定します。

   トピック・ホスト・ルーティングの場合、最初に新しいトピック・ホストを PUB(DISABLED) として定義します。 これによって、QM2 はクラスター内のサブスクリプションを認識できますが、ルーティング・パブリケーションを開始することはできません。
2. クラスター内に情報が伝搬されるのを待ちます。

   新しいクラスター・トピック定義がフル・リポジトリー・キュー・マネージャーによってクラスター内のすべてのキュー・マネージャーに伝搬されるのを待ちます。 **DISPLAY CLUSTER** コマンドを使用して各クラスター・メンバーのクラスター・トピックを表示し、QM2 から発信された定義があるかどうかを確認します。

   トピック・ホスト・ルーティング型の場合は、QM2 上の新しいトピック・ホストがすべてのサブスクリプションを認識するのを待ちます。 QM2 が認識しているプロキシー・サブスクリプションと、QM1 が認識しているプロキシー・サブスクリプションを比較します。 キュー・マネージャーのプロキシー・サブ

スクリプションを表示する 1 つの方法として、以下のように **runmqsc** コマンドを発行する方法があります。

```
DISPLAY SUB(*) SUBTYPE(PROXY)
```

3. トピック・ホスト・ルーティングの場合、QM2 上のトピック・ホストを PUB(ENABLED) として再定義してから、QM1 上のトピック・ホストを PUB(DISABLED)として再定義します。

   これで、QM2 上の新規トピック・ホストが他のキュー・マネージャーのすべてのサブスクリプションを認識し、トピック・ホストがルーティング・パブリケーションを開始できるようになりました。

   PUB(DISABLED) 設定を使用して QM1 のメッセージ・トラフィックを静止し、クラスター・トピック定義を削除するときに QM1 を通過するパブリケーションがない状態にしてください。
4. QM1 のクラスター・トピック定義を削除します。

   キュー・マネージャーが使用可能である場合は、QM1 から定義を削除するだけです。 使用不可である場合は、両方の定義が存在する状態で QM1 が再始動するまで実行するか、または強制的に削除されるまで実行する必要があります。

   QM1 が長期にわたって使用不可のままである場合で、その間に QM2 のクラスター・トピック定義を変更する必要がある場合は、QM2 定義は QM1 定義より新しいので、通常は優先されます。

   この間に、QM1 と QM2 の定義の間で差異がある場合は、両方のキュー・マネージャーのエラー・ログに、クラスター・トピック定義が競合していることを警告するエラーが書き込まれます。

   例えばハードウェアが故障した結果予期せぬ使用廃止が発生したために、QM1 が決してクラスターに戻されない場合、最後の手段として、RESET CLUSTER コマンドを使用してキュー・マネージャーを強制的に排除することができます。 **RESET CLUSTER** コマンドを実行すると、自動的に、ターゲット・キュー・マネージャーでホストされているすべてのトピック・オブジェクトが削除されます。

関連概念

パブリケーションとサブスクリプションの有効範囲の結合
IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合
キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。 パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

関連タスク

パブリッシュ/サブスクライブ・クラスターの構成
キュー・マネージャーでトピックを定義します。 トピックをクラスター・トピックにするには、**CLUSTER** プロパティーを設定します。 このトピックのパブリケーションおよびサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティーを設定します。

トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加
トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。 この方式を利用して、アベイラビリティーとワークロード・バランシングを向上させることができます。 同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続
階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。 子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断
パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加

トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。 この方式を利用して、アベイラビリティーとワークロード・バランシングを向上させることができます。 同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

始める前に

複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することが機能上役立つのは、トピック・ホスト・ルーティング型クラスターの場合に限られます。 直接ルーティング型クラスターでは、一致する複数のトピックを定義しても動作に変化はありません。 この作業は、トピック・ホスト・ルーティング型クラスターのみが適用対象です。

この作業は、同じ名前の複数のクラスター・トピック定義という記事、特に以下のセクションを読んでいることを前提としています。

- トピック・ホストで経路指定されるクラスター内の複数のクラスター・トピック定義
- PUB パラメーターの特別な処理

このタスクについて

キュー・マネージャーがルーティング型トピック・ホストにされた場合、まずキュー・マネージャーはクラスター内でサブスクライブされているすべての関連トピックの存在を検知する必要があります。 トピック・ホストがさらに追加されるときにそれらのトピックにパブリケーションがパブリッシュされ、かつその新規ホストがクラスター内の他のキュー・マネージャー上のサブスクリプションの存在を検知する前にその新規ホストに送付されるパブリケーションがあるなら、そのホストはそのパブリケーションをそれらのサブスクリプションに転送しません。 この結果、サブスクリプションでパブリケーションの欠落が生じます。

パブリケーションは、クラスター・トピック・オブジェクトの **PUB** パラメーターを明示的に DISABLED に設定したトピック・ホスト・キュー・マネージャーを介してルーティングされないため、この設定を使用して、トピック・ホストを追加するプロセス中にサブスクリプションがパブリケーションを見逃さないようにすることができます。

注 : キュー・マネージャーは PUB(DISABLED) として定義されているクラスター・トピックをホストしますが、そのキュー・マネージャーに接続されているパブリッシャーはメッセージをパブリッシュできません。また、そのキュー・マネージャー上の一致するサブスクリプションは、クラスター内の他のキュー・マネージャーにパブリッシュされたパブリケーションを受信しません。 このため、サブスクリプションが存在し、パブリッシュ・アプリケーションが接続されているキュー・マネージャーでトピック・ホスト・ルーティング型トピックを定義する際には、注意深く検討する必要があります。

手順

1. 新規トピック・ホストを構成し、最初にその新規トピック・ホストを PUB(DISABLED) として定義します。

   これにより、新規トピック・ホストはクラスター内のサブスクリプションを検知しますが、パブリケーションの送付を開始しません。

   トピック・ホストの構成については、438 ページの『パブリッシュ/サブスクライブ・クラスターの構成』を参照してください。
2. 新規トピック・ホストがすべてのサブスクリプションを検知した時点を見極めます。

   これを行うには、新規トピック・ホストで認識されているプロキシー・サブスクリプションと、既存のトピック・ホストで認識されているプロキシー・サブスクリプションとを比較します。 プロキシー・サ

ブスクリプションを表示する 1 つの方法として、次の **runmqsc** コマンドを発行することができます。
DISPLAY SUB(*) SUBTYPE(PROXY)

3. 新しいトピック・ホストを PUB(ENABLED) として再定義します。

   新規トピック・ホストが他のキュー・マネージャー上のすべてのサブスクリプションを検知した後、トピックではパブリケーションの送付を開始することができます。

関連概念

パブリケーションとサブスクリプションの有効範囲の結合
IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合
キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。 パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

関連タスク

パブリッシュ/サブスクライブ・クラスターの構成
キュー・マネージャーでトピックを定義します。 トピックをクラスター・トピックにするには、**CLUSTER** プロパティーを設定します。 このトピックのパブリケーションおよびサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティーを設定します。

別のキュー・マネージャーへのクラスター・トピック定義の移動
トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続
階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。 子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断
パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

パブリケーションとサブスクリプションの有効範囲の結合

IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

パブリケーションは、パブリッシュ/サブスクライブ・トポロジー内で接続されているすべてのキュー・マネージャーに流れるようにすることも、ローカル・キュー・マネージャーのみに流れるようにすることもできます。 プロキシー・サブスクリプションに関しても同様です。 どのパブリケーションがサブスクリプションに一致するかは、それら 2 つのフローの組み合わせによって制御されます。

パブリケーションとサブスクリプションは両方とも、有効範囲を QMGR または ALL に設定できます。 パブリッシャーとサブスクライバーが両方とも同じキュー・マネージャーに接続されている場合、有効範囲の設定が、サブスクライバーがパブリッシャーからどのパブリケーションを受け取るかに影響を及ぼすことはありません。

パブリッシャーとサブスクライバーが異なるキュー・マネージャーに接続されている場合、リモート・パブリケーションを受け取るには、両方の設定を ALL にする必要があります。

例えば、パブリッシャーが異なるキュー・マネージャーに接続されているとします。 サブスクライバーがあらゆるパブリッシャーからパブリケーションを受け取るようにするには、サブスクリプションの有効範囲を ALL に設定します。 その後、パブリッシャーごとに、そのパブリケーションの有効範囲をそのパブリッシャーにローカルのサブスクライバーに制限するかどうかを決定できます。

サブスクライバーが異なるキュー・マネージャーに接続されているとします。 あるパブリッシャーからのパブリケーションをすべてのサブスクライバーに送信するようにする場合、パブリケーション有効範囲を

ALL に設定します。 サブスクライバーが同じキュー・マネージャーに接続されているパブリッシャーだけからパブリケーションを受け取るようにする場合、サブスクリプション有効範囲を QMGR に設定します。

例: フットボールの試合結果サービス

フットボール・リーグのメンバーのチームであるとします。 各チームは、キュー・マネージャーをパブリッシュ/サブスクライブ・クラスター内の他のすべてのチームと接続させています。

チームは、トピック Football/result/*Home team name/Away team name* を使用して、ホーム・グラウンドで行われたすべてのゲーム結果を公開します。 イタリック体のストリングはトピック名の変数で、パブリケーションは試合の結果です。

また、各クラブは、トピック・ストリング Football/myteam/*Home team name/Away team name* を使用して、そのクラブの結果のみをリパブリッシュします。

両方のトピックがクラスター全体にパブリッシュされます。

リーグは以下のサブスクリプションをセットアップして、どのチームのファンも 3 つの関心のある方法で試合結果をサブスクライブできるようにします。

SUBSCOPE(QMGR) を使用してクラスター・トピックをセットアップすることができます。 トピック定義はクラスターの各メンバーに伝搬されますが、サブスクリプションの有効範囲はローカル・キュー・マネージャーのみです。 そのため、各キュー・マネージャーのサブスクライバーは、同じサブスクリプションからのさまざまなパブリケーションを受け取ります。

すべての試合結果を受け取る

```
DEFINE TOPIC(A) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(ALL)
```

ホームの試合結果すべてを受け取る

```
DEFINE TOPIC(B) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(QMGR)
```

サブスクリプションの有効範囲は QMGR なので、ホーム・グラウンドでパブリッシュされた試合結果のみが一致します。

自分のチームの試合結果すべてを受け取る

```
DEFINE TOPIC(C) TOPICSTR('Football/myteam/') CLUSTER SUBSCOPE(QMGR)
```

サブスクリプションの有効範囲は QMGR なので、ローカルでリパブリッシュされた、ローカルのチームの結果のみが一致します。

関連概念

パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合
キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。 パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

分散パブリッシュ/サブスクライブのネットワーク

パブリケーション有効範囲

サブスクリプション有効範囲

関連タスク

パブリッシュ/サブスクライブ・クラスターの構成
キュー・マネージャーでトピックを定義します。 トピックをクラスター・トピックにするには、**CLUSTER** プロパティーを設定します。 このトピックのパブリケーションおよびサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティーを設定します。

別のキュー・マネージャーへのクラスター・トピック定義の移動
トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加
トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。 この方式を利用して、アベイラビリティーとワークロード・バランシングを向上させることができます。 同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続
階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。 子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断
パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合

キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。 パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

CLUSTER、**PUBSCOPE**、および **SUBSCOPE** 属性のビルディング・ブロック、パブリッシュ/サブスクライブ・クラスター、およびパブリッシュ/サブスクライブ階層を使用して、さまざまなパブリッシュ/サブスクライブ・トピック・スペースを作成できます。

以下のシナリオでは、1 つのキュー・マネージャーからパブリッシュ/ サブスクライブ・クラスターに拡大する例に始まり、さまざまなパブリッシュ/サブスクライブ・トポロジーについて説明します。

関連概念
パブリケーションとサブスクリプションの有効範囲の結合
IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

分散パブリッシュ/サブスクライブのネットワーク
トピック・スペース
関連タスク
パブリッシュ/サブスクライブ・クラスターの構成
キュー・マネージャーでトピックを定義します。 トピックをクラスター・トピックにするには、**CLUSTER** プロパティーを設定します。 このトピックのパブリケーションおよびサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティーを設定します。

別のキュー・マネージャーへのクラスター・トピック定義の移動
トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加
トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。 この方式を利用して、アベイラビリティーとワークロード・バランシングを向上させることができます。 同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続
階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。 子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断

パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

クラスター・トピックの定義

パブリッシュ/サブスクライブ・クラスターでの単一のトピック・スペースの作成

複数のキュー・マネージャー上で実行されるように、パブリッシュ/サブスクライブ・システムを拡大します。 各パブリッシャーおよびサブスクライバーに単一で同一のトピック・スペースを提供するために、パブリッシュ/サブスクライブ・クラスターを使用します。

始める前に

単一のバージョン 7 キュー・マネージャーにパブリッシュ/サブスクライブ・システムを実装しています。

SYSTEM.BASE.TOPIC の属性の継承に依存するのではなく、常に独自のルート・トピックを使用してトピック・スペースを作成してください。 パブリッシュ/サブスクライブ・システムをクラスターに拡大する場合、クラスター・トピック・ホストで、ルート・トピックをクラスター・トピックとして定義でき、そうするとすべてのトピックがクラスター全体で共有されます。

このタスクについて

より多くのパブリッシャーおよびサブスクライバーをサポートするためにシステムを拡大し、すべてのトピックをクラスター全体で可視にしようとしています。

手順

1. パブリッシュ/サブスクライブ・システムで使用するクラスターを作成します。
   従来のクラスターが既に存在する場合、パフォーマンス上の理由で、新しいパブリッシュ/サブスクライブ・システム用に新しいクラスターをセットアップする方が良いでしょう。 両方のクラスターのクラスター・リポジトリーに、同じサーバーを使用できます。
2. クラスター・トピック・ホストにするキュー・マネージャーを 1 つ (おそらく、リポジトリーのいずれか) 選択します。
3. パブリッシュ/サブスクライブ・クラスター全体で可視にするすべてのトピックが、管理トピック・オブジェクトに解決されるようにします。
   パブリッシュ/サブスクライブ・クラスターの名前を指定する **CLUSTER** 属性を設定します。

次のタスク

パブリッシャーおよびサブスクライバー・アプリケーションをクラスター内の任意のキュー・マネージャーに接続します。

CLUSTER 属性を持つ管理トピック・オブジェクトを作成します。 これらのトピックもクラスター全体に伝搬されます。 パブリッシャーおよびサブスクライバー・プログラムは、クラスター内の異なるキュー・マネージャーに接続しても動作が変わらないように、管理トピックを使用します。

すべてのキュー・マネージャーで SYSTEM.BASE.TOPIC をクラスター・トピックのように機能させる必要がある場合は、キュー・マネージャーごとに変更する必要があります。

関連概念

分散パブリッシュ/サブスクライブのネットワーク

トピック・スペース

関連タスク

複数のクラスターのトピック・スペースの結合

複数のクラスターにまたがるトピック・スペースを作成します。 1 つのクラスターでトピックにパブリッシュし、別のクラスターでそれに対してサブスクライブします。

複数のクラスター内でのトピック・スペースの結合および分離

いくつかのトピック・スペースを特定のクラスターに分離して、他のトピック・スペースを結合し、接続されているすべてのクラスター内でそれらにアクセスできるようにします。

複数のクラスター内のトピック・スペースに対するパブリッシュおよびサブスクライブ

オーバーラップされたクラスターを使用して、複数のクラスター内のトピックに対するパブリッシュおよびサブスクライブを行います。 クラスター内でトピック・スペースがオーバーラップしない限り、この手法を使用できます。

クラスター・トピックの定義

複数のクラスターのトピック・スペースの結合

複数のクラスターにまたがるトピック・スペースを作成します。 1 つのクラスターでトピックにパブリッシュし、別のクラスターでそれに対してサブスクライブします。

始める前に

この作業では、直接ルーティング型パブリッシュ/サブスクライブ・クラスターが既に存在しており、いくつかのクラスター・トピックをすべてのクラスターに伝搬させようとしていることを前提としています。

注 : トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターの場合は、この作業を実行できません。

このタスクについて

あるクラスターから別のクラスターへパブリケーションを伝搬するには、それらのクラスターをまとめて 1 つの階層に結合する必要があります。447 ページの図 65 を参照してください。 階層型の接続は、接続されているキュー・マネージャー間でサブスクリプションおよびパブリケーションを伝搬します。クラスターは各クラスター内でクラスター・トピックを伝搬しますが、クラスター間では伝搬しません。

これらの 2 つのメカニズムの組み合わせにより、すべてのクラスター間でクラスター・トピックが伝搬されます。 クラスターごとに、クラスター・トピック定義を繰り返す必要があります。

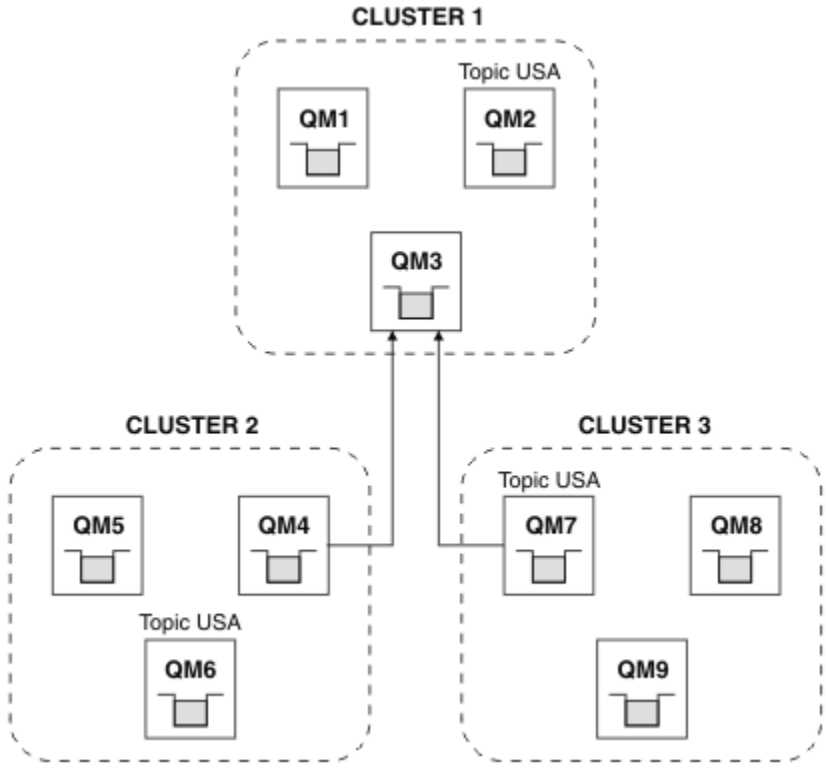


図 65. 階層を使用したクラスターの接続

以下のステップにより、クラスターが階層に接続されます。

手順

1. 二つの送信側と受信側チャネルのセットを作成して、QM3 と QM4、QM3 と QM7 を両方の方向に接続します。 従来の送信側と受信側のチャネルおよび伝送キュー (クラスターではなく) を使用して、階層に接続しなければなりません。
2. ターゲット・キュー・マネージャーの名前を指定して、3 つの伝送キューを作成します。 何らかの理由で、伝送キュー名としてターゲット・キュー・マネージャーの名前を使用できない場合は、キュー・マネージャーの別名を使用します。
3. 送信側チャネルをトリガーするように、伝送キューを構成します。
4. QM3、QM4、および QM7 の **PSMODE** が ENABLE に設定されていることを確認します。
5. QM4 および QM7 の **PARENT** 属性を QM3 に変更します。
6. キュー・マネージャー間の親子関係の状況が両方向でアクティブであることを検査します。
7. 属性 **CLUSTER** (「CLUSTER 1」)、**CLUSTER** (「CLUSTER 2」)、および **CLUSTER** (「CLUSTER 3」) を使用してクラスター 1、2、および 3 内の 3 つのクラスター・トピック・ホスト・キュー・マネージャーのそれぞれに管理トピック USA を作成します。 クラスター・トピック・ホストは、階層的に接続されたキュー・マネージャーである必要はありません。

次のタスク

これで、447 ページの図 65 でクラスター・トピック「USA」をパブリッシュまたはサブスクライブできます。 パブリケーション/サブスクリプションは、3 つのすべてのクラスター内のパブリッシャーおよびサブスクライバーに流れます。

USA を他のクラスター内のクラスター・トピックとして作成しなかったとします。 USA のみで定義されている場合、QM7、パブリケーションとサブスクリプションは USAQM7、QM8、QM9 と QM3 の間で交換されます。 QM7、QM8、QM9 で実行されているパブリッシャーとサブスクライバーは管理トピックの属性 USA を継承します。 QM3 上のパブリッシャーおよびサブスクライバーは、QM3 上の SYSTEM.BASE.TOPIC の属性を継承します。

448 ページの『複数のクラスター内でのトピック・スペースの結合および分離』 も参照してください。

関連概念

分散パブリッシュ/サブスクライブのネットワーク

トピック・スペース

関連タスク

パブリッシュ/サブスクライブ・クラスターでの単一のトピック・スペースの作成

複数のキュー・マネージャー上で実行されるように、パブリッシュ/サブスクライブ・システムを拡大します。 各パブリッシャーおよびサブスクライバーに単一で同一のトピック・スペースを提供するために、パブリッシュ/サブスクライブ・クラスターを使用します。

複数のクラスター内でのトピック・スペースの結合および分離

いくつかのトピック・スペースを特定のクラスターに分離して、他のトピック・スペースを結合し、接続されているすべてのクラスター内でそれらにアクセスできるようにします。

複数のクラスター内のトピック・スペースに対するパブリッシュおよびサブスクライブ

オーバーラップされたクラスターを使用して、複数のクラスター内のトピックに対するパブリッシュおよびサブスクライブを行います。 クラスター内でトピック・スペースがオーバーラップしない限り、この手法を使用できます。

クラスター・トピックの定義

複数のクラスター内でのトピック・スペースの結合および分離

いくつかのトピック・スペースを特定のクラスターに分離して、他のトピック・スペースを結合し、接続されているすべてのクラスター内でそれらにアクセスできるようにします。

始める前に

447 ページの『複数のクラスターのトピック・スペースの結合』のトピックを調べてください。 ブリッジとしてさらにキュー・マネージャーを追加しなくても、このトピックの手順で十分に必要を満たせる場合があります。

注 : この作業は、直接ルーティング型パブリッシュ/サブスクライブ・クラスターを使用する場合にのみ実行できます。 トピック・ホスト・ルーティング型クラスターを使用して実行することはできません。

このタスクについて

すべてのクラスター間で共有されないクラスター・トピックを分離させることにより、447 ページの『複数のクラスターのトピック・スペースの結合』の 447 ページの図 65 で示されているトポロジーを向上させることができる可能性があります。 どのクラスターにも含まれていないブリッジング・キュー・マネージャーを作成することにより、クラスターを分離します (449 ページの図 66 を参照)。 ブリッジング・キュー・マネージャーを使用して、どのパブリケーションおよびサブスクリプションが、あるクラスターから別のクラスターへ流れるようにするかをフィルター処理します。

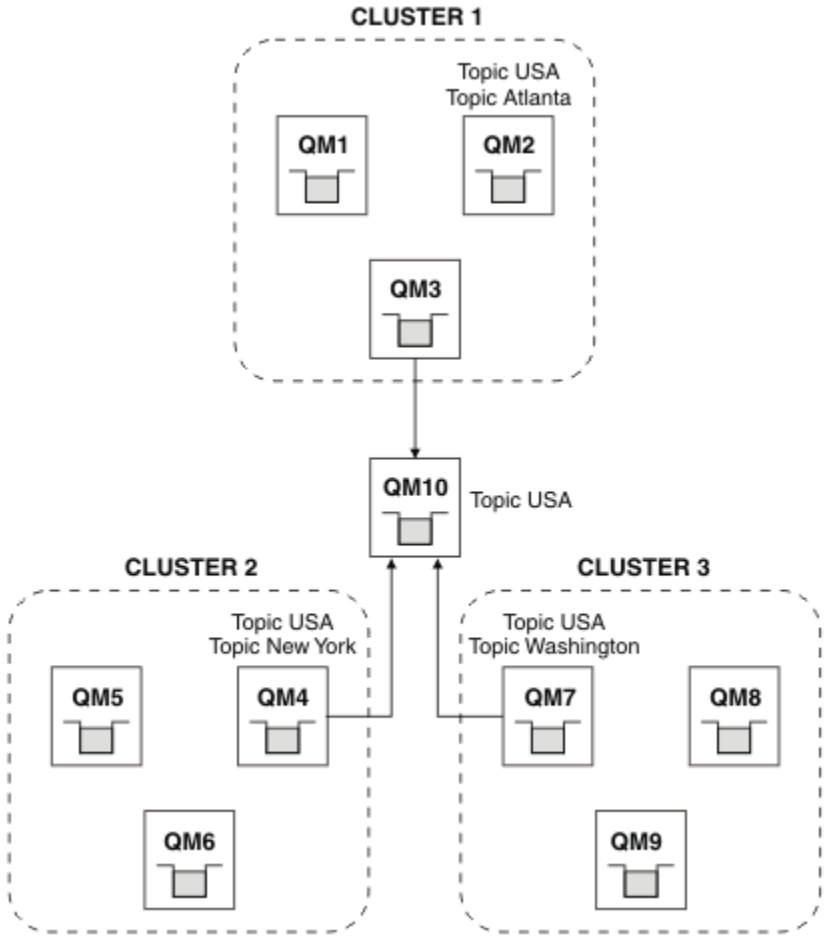


図 66. ブリッジされるクラスター

ブリッジを使用すると、ブリッジを超えて他のクラスターで公開しないクラスター・トピックを分離できます。 において 449 ページの図 66、USA は全クラスタで共有されるクラスタトピック、Atlanta、New York および Washington それぞれ 1 つのクラスタでのみ共有されるクラスタトピックである。

以下の手順を使用して、構成をモデル化しまてください。

手順

1. すべてのキュー・マネージャーで **SUBSCOPE** (QMGR) および **PUBSCOPE** (QMGR) を持つようにすべての SYSTEM.BASE.TOPIC トピック・オブジェクトを変更します。

   クラスター・トピックのルート・トピックに **SUBSCOPE** (ALL) および **PUBSCOPE** (ALL) を明示的に設定しない限り、トピック (クラスター・トピックも含む) は他のキュー・マネージャーに伝搬されません。
2. 各クラスターで共有する 3 つのクラスター・トピック・ホスト・キュー・マネージャーのトピックを、属性 **CLUSTER** (*clustername*)、**SUBSCOPE** (ALL)、および **PUBSCOPE** (ALL) を使用して定義します。

   いくつかのクラスター・トピックをすべてのクラスター間で共有するには、それぞれのクラスターで同じトピックを定義します。 クラスター属性として、各クラスターのクラスター名を使用します。
3. すべてのクラスター間で共有するクラスター・トピックについて、属性 **SUBSCOPE** (ALL) および **PUBSCOPE** (ALL) を使用して、ブリッジ・キュー・マネージャー (QM10) でトピックを再定義します。

例

449 ページの図 66 の例では、USA から継承するトピックのみが、三つのクラスターの間で伝搬されます。

次のタスク

SUBSCOPE (ALL) および **PUBSCOPE** (ALL) を使用してブリッジ・キュー・マネージャーで定義されたトピックのサブスクリプションがクラスター間で伝搬されます。

属性 **CLUSTER** (*clustername*)、**SUBSCOPE** (ALL) および **PUBSCOPE** (ALL) を使用して各クラスター内で定義されたトピックのサブスクリプションが各クラスター内で伝搬されます。

他のサブスクリプションはすべて、キュー・マネージャーにとってローカルです。

関連概念

分散パブリッシュ/サブスクライブのネットワーク
トピック・スペース
パブリケーション有効範囲
サブスクリプション有効範囲

関連タスク

パブリッシュ/サブスクライブ・クラスターでの単一のトピック・スペースの作成
複数のキュー・マネージャー上で実行されるように、パブリッシュ/サブスクライブ・システムを拡大します。 各パブリッシャーおよびサブスクライバーに単一で同一のトピック・スペースを提供するために、パブリッシュ/サブスクライブ・クラスターを使用します。

複数のクラスターのトピック・スペースの結合
複数のクラスターにまたがるトピック・スペースを作成します。 1 つのクラスターでトピックにパブリッシュし、別のクラスターでそれに対してサブスクライブします。

複数のクラスター内のトピック・スペースに対するパブリッシュおよびサブスクライブ
オーバーラップされたクラスターを使用して、複数のクラスター内のトピックに対するパブリッシュおよびサブスクライブを行います。 クラスター内でトピック・スペースがオーバーラップしない限り、この手法を使用できます。

クラスター・トピックの定義

複数のクラスター内のトピック・スペースに対するパブリッシュおよびサブスクライブ

オーバーラップされたクラスターを使用して、複数のクラスター内のトピックに対するパブリッシュおよびサブスクライブを行います。 クラスター内でトピック・スペースがオーバーラップしない限り、この手法を使用できます。

始める前に

従来のクラスターを複数作成して、クラスター間の交差部分にいくつかのキュー・マネージャーを配置します。

このタスクについて

さまざまな理由で、クラスターをオーバーラップさせることを選択する場合があります。

1. 高可用性サーバーまたはキュー・マネージャーの数が限られています。 クラスター・リポジトリーおよびクラスター・トピック・ホストはすべてそれらにデプロイすることにしています。
2. ゲートウェイ・キュー・マネージャーを使用して接続される従来のキュー・マネージャー・クラスターが既に存在します。 パブリッシュ/サブスクライブ・アプリケーションを同じクラスター・トポロジーにデプロイする予定です。
3. 自己完結型のパブリッシュ/サブスクライブ・アプリケーションがいくつか存在します。 パフォーマンス上の理由で、パブリッシュ/サブスクライブ・クラスターを小さいままにし、従来のクラスターから分離させる方が適切です。 それらのアプリケーションを別のクラスターにデプロイすることにしています。 ただし、モニタリング・アプリケーションのライセンス交付を受けたコピーが 1 つだけであるため、1 つのキュー・マネージャー上ですべてのパブリッシュ/サブスクライブ・アプリケーションをモニターする必要があります。 このキュー・マネージャーは、すべてのクラスター内のクラスター・トピックに対するパブリケーションにアクセスできなければなりません。

オーバーラップしないトピック・スペースでトピックが定義されるようにすることにより、オーバーラップしているパブリッシュ/サブスクライブ・クラスターにそのトピックをデプロイすることができます (451 ページの図 67 を参照)。 トピック・スペースがオーバーラップしている場合、オーバーラップするクラスターにデプロイすると問題が発生します。

パブリッシュ/サブスクライブ・クラスターがオーバーラップするため、オーバーラップ内のキュー・マネージャーを使用して、任意のトピック・スペースにパブリッシュおよびサブスクライブすることができます。

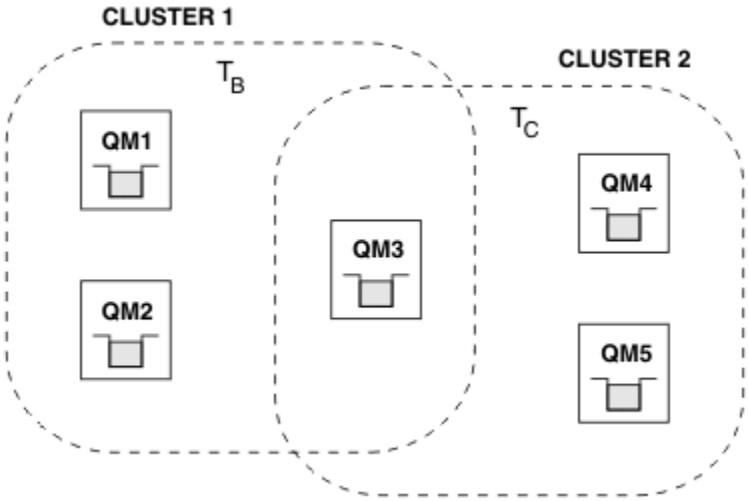


図 67. オーバーラップするクラスター、オーバーラップしないトピック・スペース

手順

トピック・スペースがオーバーラップしないようにする方法を作成します。

例えば、トピック・スペースごとに固有のルート・トピックを定義します。 そのルート・トピックをクラスター・トピックにします。

a) `DEFINE TOPIC(B) TOPICSTR('B') CLUSTER('CLUSTER 1') ...`

b) `DEFINE TOPIC(C) TOPICSTR('C') CLUSTER('CLUSTER 2') ...`

例

451 ページの図 67 では、QM3 に接続されたパブリッシャーおよびサブスクライバーは、T B または T C にパブリッシュまたはサブスクライブできます

次のタスク

両方のクラスター内のトピックを使用するパブリッシャーおよびサブスクライバーを、オーバーラップ内のキュー・マネージャーに接続します。

特定のクラスター内のトピックのみを使用する必要があるパブリッシャーおよびサブスクライバーを、オーバーラップ内にないキュー・マネージャーに接続します。

関連概念

分散パブリッシュ/サブスクライブのネットワーク

トピック・スペース

関連タスク

パブリッシュ/サブスクライブ・クラスターでの単一のトピック・スペースの作成

複数のキュー・マネージャー上で実行されるように、パブリッシュ/サブスクライブ・システムを拡大します。 各パブリッシャーおよびサブスクライバーに単一で同一のトピック・スペースを提供するために、パブリッシュ/サブスクライブ・クラスターを使用します。

複数のクラスターのトピック・スペースの結合

複数のクラスターにまたがるトピック・スペースを作成します。 1 つのクラスターでトピックにパブリッシュし、別のクラスターでそれに対してサブスクライブします。

複数のクラスター内でのトピック・スペースの結合および分離

いくつかのトピック・スペースを特定のクラスターに分離して、他のトピック・スペースを結合し、接続されているすべてのクラスター内でそれらにアクセスできるようにします。

クラスター・トピックの定義

パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続

階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。 子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

始める前に

1. パブリッシュ/サブスクライブ階層内のキュー・マネージャーには、固有のキュー・マネージャー名が必要です。
2. パブリッシュ/ サブスクライブ階層は、"キュー型パブリッシュ/サブスクライブ"のキュー・マネージャー機能に依存します。 親と子のキュー・マネージャー両方でこれが有効になっている必要があります。 433 ページの『キューに入れられたパブリッシュ/サブスクライブの開始』を参照してください。
3. パブリッシュ/サブスクライブ関係は、キュー・マネージャーの送信側チャネルと受信側チャネルに依存します。 チャネルを確立する方法は 2 つあります。
   - 親および子キュー・マネージャーの両方を IBM MQ クラスターに追加します。 313 ページの『クラスターにキュー・マネージャーを追加する』を参照してください。
   - 送信側/受信側チャネルのペアを、子キュー・マネージャーから親へ、および親キュー・マネージャーから子へ確立します。 各チャネルは、ターゲット・キュー・マネージャーと同じ名前の伝送キューか、ターゲット・キュー・マネージャーと同じ名前のキュー・マネージャー別名のいずれかを使用する必要があります。 Point-to-Point チャネル接続を確立する方法の詳細については、191 ページの『IBM MQ 分散キューイング技法』を参照してください。

   チャネル構成の各タイプにわたる階層を構成する例については、以下の一連のパブリッシュ/サブスクライブ階層シナリオを参照してください。

   - シナリオ 1: キュー・マネージャー別名を使用した Point-to-Point チャネルの使用
   - シナリオ 2: 伝送キューとリモート・キュー・マネージャーに同じ名前を使用した Point-to-Point チャネルの使用
   - シナリオ 3: クラスター・チャネルを使用したキュー・マネージャーの追加

このタスクについて

ALTER QMGR PARENT（*PARENT_NAME*）**runmqsc** コマンドを使用して、子を親に接続します。 この構成は子キュー・マネージャー上で実行されます。*PARENT_NAME* は親キュー・マネージャーの名前です。

手順

ALTER QMGR PARENT(*PARENT_NAME*)

例

最初の例は、キュー・マネージャー QM2 を QM1 の子として接続し、QM2 を照会して、**STATUS** が ACTIVE の子に正常になったことを確認する方法を示しています。

```
C:>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2
alter qmgr parent(QM1)
     1 : alter qmgr parent(QM1)
AMQ8005: IBM MQ queue manager changed.
display pubsub all
     2 : display pubsub all
AMQ8723: Display pub/sub status details.
   QMNAME(QM2)                             TYPE(LOCAL)
   STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
   QMNAME(QM1)                             TYPE(PARENT)
STATUS(ACTIVE)
```

次の例は、QM1 にその接続について照会した結果を示しています。

```
C:\Documents and Settings\Admin>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
display pubsub all
     2 : display pubsub all
AMQ8723: Display pub/sub status details.
   QMNAME(QM1)                             TYPE(LOCAL)
   STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
   QMNAME(QM2)                             TYPE(CHILD)
   STATUS(ACTIVE)
```

STATUS が ACTIVE と表示されない場合は、子と親の間のチャネルが正しく構成され、実行されていることを確認してください。 両方のキュー・マネージャーのエラー・ログで、考えられるエラーを確認します。

次のタスク

1 つのキュー・マネージャーでパブリッシャーとサブスクライバーが使用するトピックは、デフォルトで階層内の他のキュー・マネージャー上のパブリッシャーとサブスクライバーによって共有されます。 管理対象トピックは、**SUBSCOPE** および **PUBSCOPE** トピック・プロパティーを使用して共有のレベルを制御するように構成できます。 437 ページの『分散パブリッシュ/サブスクライブ・ネットワークの構成』を参照してください。

関連概念

パブリケーションとサブスクリプションの有効範囲の結合

IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合

キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。 パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

関連タスク

パブリッシュ/サブスクライブ・クラスターの構成

キュー・マネージャーでトピックを定義します。 トピックをクラスター・トピックにするには、**CLUSTER** プロパティーを設定します。 このトピックのパブリケーションおよびサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティーを設定します。

別のキュー・マネージャーへのクラスター・トピック定義の移動
トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加
トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。 この方式を利用して、アベイラビリティーとワークロード・バランシングを向上させることができます。 同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断
パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

関連資料
ストリームおよびトピック
DISPLAY PUBSUB
パブリッシュ/サブスクライブ・メッセージング

パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断

パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

このタスクについて

ALTER QMGR コマンドを使用して、キュー・マネージャーをブローカー階層から切断します。 キュー・マネージャーは、いつでも任意の順序で切断することができます。

親の更新に対応する要求は、キュー・マネージャー間の接続が稼働中に送信されます。

手順

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

例

```
C:\Documents and Settings\Admin>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2.
    1 : alter qmgr parent('')
AMQ8005: IBM MQ queue manager changed.
    2 : display pubsub type(child)
AMQ8147: IBM MQ object  not found.
display pubsub type(parent)
    3 : display pubsub type(parent)
AMQ8147: IBM MQ object not found.
```

次のタスク

不要になったすべてのストリーム、キュー、および手動で定義したチャネルをすべて削除できます。

関連概念
パブリケーションとサブスクリプションの有効範囲の結合
IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合

キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。 パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

関連タスク

パブリッシュ/サブスクライブ・クラスターの構成
キュー・マネージャーでトピックを定義します。 トピックをクラスター・トピックにするには、**CLUSTER** プロパティーを設定します。 このトピックのパブリケーションおよびサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティーを設定します。

別のキュー・マネージャーへのクラスター・トピック定義の移動
トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加
トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。 この方式を利用して、アベイラビリティーとワークロード・バランシングを向上させることができます。 同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続
階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。 子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

ALW 複数のインストールの構成

同一のシステムで複数のインストールを使用している場合は、それらのインストールとキュー・マネージャーを構成する必要があります。

このタスクについて

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

手順

- 以下のリンクの情報を使用してインストールを構成してください。
  - 463 ページの『プライマリー・インストールの変更』
  - 464 ページの『キュー・マネージャーとインストールの関連付け』
  - 455 ページの『複数のインストール環境でのアプリケーションの接続』

ALW 複数のインストール環境でのアプリケーションの接続

AIX, Linux, and Windows システムでは、 IBM MQ ライブラリーがロードされると、 IBM MQ は自動的に適切なライブラリーを使用します。これ以上の処置は必要ありません。 IBM MQ は、アプリケーションが接続されるキュー・マネージャーと関連付けられているインストール済み環境からのライブラリーを使用します。

以下の概念は、アプリケーションを IBM MQ に接続する方法を説明する際に使用されます。

リンク
アプリケーションは、コンパイル時に IBM MQ ライブラリーにリンクされ、機能エクスポートを取得します。このエクスポートはアプリケーションの実行時に読み込まれます。

ロード中
アプリケーションの実行時に、IBM MQ ライブラリーの場所が探索され、ロードされます。 ライブラリーを見つけるための具体的なメカニズムは、オペレーティング・システムと、アプリケーションの構築方法によって異なります。 複数インストール環境でのライブラリーの場所探索方法とロード方法の詳細については、457 ページの『IBM MQ ライブラリーのロード』を参照してください。

接続中
アプリケーションは、MQCONN または MQCONNX 呼び出しなどを使用して、実行中のキュー・マネージャーに接続する際に、読み込まれた IBM MQ ライブラリーを使用して接続を行います。

サーバー・アプリケーションをキュー・マネージャーに接続する場合は、そのキュー・マネージャーに関連付けられているインストール済み環境から読み込まれたライブラリーを使用する必要があります。 システム上に複数のインストール済み環境がある場合、この制限により、オペレーティング・システムが、ロードする IBM MQ ライブラリーを見つけるために使用するメカニズムを選択する際に、以下のような新たな課題が生じます。

- **setmqm** コマンドを使用して、キュー・マネージャーに関連付けられているインストール済み環境を変更する場合、読み込む必要があるライブラリーも変わる。
- アプリケーションを、異なるインストール済み環境が所有している複数のキュー・マネージャーに接続する場合、複数のライブラリー・セットを読み込む必要がある。

ただし、IBM MQ(ライブラリー) が配置されてロードされている場合、IBM MQ は適切なライブラリーをロードして使用するため、追加のアクションを行う必要はありません。 アプリケーションがキュー・マネージャーに接続されると、IBM MQ は、そのキュー・マネージャーに関連付けられているインストール済み環境からライブラリーを読み込みます。

複数のインストール済み環境でのマイグレーション・シナリオおよびアプリケーションの接続について詳しくは、AIX, Linux, and Windows での複数インストール済み環境のキュー・マネージャーの共存を参照してください。

IBM MQ ライブラリーのロード方法の詳細については、457 ページの『IBM MQ ライブラリーのロード』を参照してください。

サポートおよび制約事項

以下のいずれかの IBM MQ ライブラリーが見つかり、ロードされると、製品は自動的に適切なライブラリーをロードして使用することができます。

- C サーバー・ライブラリー
- C++ サーバー・ライブラリー
- XA サーバー・ライブラリー
- COBOL サーバー・ライブラリー
- COM+ サーバー・ライブラリー
- 非管理対象モードの .NET

IBM MQ は、バインディング・モードの Java および JMS アプリケーションに適したライブラリーも自動的にロードし、使用します。

複数のインストール済み環境を使用するアプリケーションに関する制約事項がいくつかあります。 詳しくは、460 ページの『複数のインストールを使用するアプリケーションの制限』 を参照してください。

関連概念

460 ページの『複数のインストールを使用するアプリケーションの制限』
CICS サーバー・ライブラリー、ファスト・パス接続、メッセージ・ハンドル、および出口を複数インストール環境で使用する場合、いくつかの制限があります。

457 ページの『IBM MQ ライブラリーのロード』
IBM MQ ライブラリーのロード方法を決定する際には、使用環境、既存のアプリケーションを変更できるかどうか、プライマリー・インストールを行うかどうか、IBM MQ のインストール場所、および IBM MQ の場所を変更する可能性があるかどうかなど、いくつかの要素を考慮する必要があります。

関連タスク

プライマリー・インストールの選択

463 ページの『プライマリー・インストールの変更』

setmqinst コマンドを使用すると、インストールをプライマリー・インストールとして設定する、あるいはその設定を解除することができます。

464 ページの『キュー・マネージャーとインストールの関連付け』

キュー・マネージャーを作成すると、そのキュー・マネージャーは、**crtmqm** コマンドを実行したインストールに自動的に関連付けられます。 AIX, Linux, and Windows では、**setmqm** コマンドを使用して、キュー・マネージャーに関連付けられているインストールを変更できます。

ALW IBM MQ ライブラリーのロード

IBM MQ ライブラリーのロード方法を決定する際には、使用環境、既存のアプリケーションを変更できるかどうか、プライマリー・インストールを行うかどうか、IBM MQ のインストール場所、および IBM MQ の場所を変更する可能性があるかどうかなど、いくつかの要素を考慮する必要があります。

IBM MQ ライブラリーの場所を探索しロードする方法は、ご使用のインストール済み環境に応じて異なります。

- AIX and Linux システムでは、 IBM MQ バージョンのコピーがデフォルト・ロケーションにインストールされている場合、既存のアプリケーションは引き続き前のバージョンと同じように機能します。 ただし、アプリケーションが `/usr/lib` でシンボリック・リンクを必要とする場合は、 IBM MQ バージョンのインストール済み環境をプライマリー・インストールとして選択するか、シンボリック・リンクを手動で作成する必要があります。
- IBM MQ がデフォルト以外の場所にインストールされている場合は、正しいライブラリーがロードされるように既存のアプリケーションを変更する必要がある場合があります。

IBM MQ ライブラリーの場所を探索しロードする方法は、既存のアプリケーションがライブラリーをロードするためにどのようにセットアップされたかによっても異なります。 ライブラリーのロード方法の詳細については、459 ページの『オペレーティング・システムのライブラリー・ロード・メカニズム』を参照してください。

最善策として、オペレーティング・システムによってロードされる IBM MQ ライブラリーが、キュー・マネージャーと関連付けられたライブラリーとなるようにしてください。

IBM MQ ライブラリーをロードする方法は、プラットフォームによって異なり、それぞれ利点と欠点があります。

表 *28.* ライブラリーをロードするためのオプションの利点と欠点

プラットフォーム	オプション	利点	欠点
Linux AIX AIX and Linux システム	アプリケーションの埋め込みランタイム検索パス (RPath) を設定または変更する。 このオプションでは、アプリケーションを再コンパイルしてリンクする必要があります。 アプリケーションのコンパイルとリンクについて詳しくは、プロシージャー型アプリケーションの構築を参照してください。	• 変更範囲が明確です。	• アプリケーションを再コンパイルしてリンクできる必要があります。 • IBM MQ の場所が変わった場合に、RPath を変更する必要があります。

表 28. ライブラリーをロードするためのオプションの利点と欠点 (続き)			
プラットフォーム	**オプション**	**利点**	**欠点**
AIX and Linux システム	**-k** オプションまたは **-l** オプションを指定して、setmqenv または crtmqenv を使用し、*LD_LIBRARY_PATH* 環境変数を設定します。(AIX では、この環境変数は *LIBPATH*)	• 既存のアプリケーションを変更する必要がありません。 • アプリケーションの埋め込み RPath をオーバーライドします。 • IBM MQ の場所が変わった場合の変数の変更が容易です。	• setuid アプリケーションおよび setgid アプリケーション、または他の方法で構築されたアプリケーションでは、セキュリティー上の理由で、*LD_LIBRARY_PATH* が無視される場合があります。 • 環境固有であるため、アプリケーションを実行する環境ごとに設定する必要があります。 • *LD_LIBRARY_PATH* に依存する他のアプリケーションに影響する可能性があります。 • Linux: アプリケーションの構築時に使用したコンパイラーにより、*LD_LIBRARY_PATH* が使用不可になっている場合があります。 詳しくは、Linux 用のランタイム・リンクに関する考慮事項を参照してください。
Windows Windows システム	setmqenv または crtmqenv を使用して、PATH 変数を設定する。	• 既存のアプリケーションを変更する必要がありません。 • IBM MQ の場所が変わった場合の変数の変更が容易です。	• 環境固有であるため、アプリケーションを実行する環境ごとに設定する必要があります。 • 他のアプリケーションに影響する可能性があります。
ALW AIX, Linux, and Windows システム	プライマリー・インストールを IBM MQ 以降のインストールに設定する。 463 ページの『プライマリー・インストールの変更』を参照してください。 プライマリー・インストールについて詳しくは、プライマリー・インストールの選択を参照してください。	• 既存のアプリケーションを変更する必要がありません。 • IBM MQ の場所が変わった場合のプライマリー・インストールの変更が容易です。 • 前のバージョンの IBM MQ と同じような動作になります。	• Linux AIX AIX and Linux: /usr/lib がデフォルトの検索パスにない場合は、機能しません。

Linux のライブラリーのロードに関する考慮事項

Linux

バージョン 3.2.x など、一部のバージョンの gcc を使用してコンパイルされたアプリケーションには、*LD_LIBRARY_PATH* 環境変数を使用してオーバーライドできない RPath を埋め込むことができます。 `readelf -d applicationName` コマンドを使用して、アプリケーションが影響を受けるかどうかを判別できます。 RPATH シンボルは存在するが、RUNPATH シンボルが存在しない場合は、RPath をオーバーライドすることはできません。

オペレーティング・システムのライブラリー・ロード・メカニズム

Windows システムでは、ライブラリーを見つける際に以下のいくつかのディレクトリーが検索されます。

- アプリケーションの読み込み元のディレクトリー。
- 現行ディレクトリー。
- *PATH* 環境変数 (グローバル *PATH* 変数および現行ユーザーの *PATH* 変数の両方) のディレクトリー。

Linux AIX AIX and Linux システムでは、以下のように、ロードするライブラリーを見つけるために使用されてきた方法が多数あります。

- *LD* ライブラリー・パス 環境変数を使用する (AIX では ライブラリー・パス も使用)。 この変数が設定されている場合、その変数は、必要な IBM MQ ライブラリーを検索する一連のディレクトリーを定義します。 これらのディレクトリーにライブラリーがある場合、他の方法を使用して検出されるどのライブラリーよりもそのライブラリーが優先されます。
- 埋め込み検索パス (RPath) を使用。 アプリケーションには、IBM MQ ライブラリーを検索する一連のディレクトリーが含まれている場合があります。 *LD_LIBRARY_PATH* が設定されていない場合、または必要なライブラリーが変数を使用して検出されなかった場合は、ライブラリーを見つけるために RPath が検索されます。 既存のアプリケーションで RPath を使用しているが、アプリケーションを再コンパイルしてリンクできない場合は、デフォルトの場所に IBM MQ をインストールするか、ライブラリーの検出に別の方法を使用する必要があります。
- デフォルトのライブラリー・パスを使用。 *LD_LIBRARY_PATH* 変数と RPath の場所を検索しても IBM MQ ライブラリーが見つからない場合は、デフォルトのライブラリー・パスが検索されます。 通常、パスには `/usr/lib` または `/usr/lib64` が含まれます。 デフォルトのライブラリー・パスを検索してもライブラリーが見つからなかった場合は、依存関係の欠落により、アプリケーションは開始できません。

アプリケーションに埋め込み検索パスがあるかどうかについては、オペレーティング・システム・メカニズムを使用して調べることができます。 以下に例を示します。

- AIX AIX: **dump**
- Linux Linux: **readelf**

関連概念

460 ページの『複数のインストールを使用するアプリケーションの制限』
CICS サーバー・ライブラリー、ファスト・パス接続、メッセージ・ハンドル、および出口を複数インストール環境で使用する場合、いくつかの制限があります。

455 ページの『複数のインストール環境でのアプリケーションの接続』
AIX, Linux, and Windows システムでは、 IBM MQ ライブラリーがロードされると、 IBM MQ は自動的に適切なライブラリーを使用します。これ以上の処置は必要ありません。 IBM MQ は、アプリケーションが接続されるキュー・マネージャーと関連付けられているインストール済み環境からのライブラリーを使用します。

関連タスク

プライマリー・インストールの選択

463 ページの『プライマリー・インストールの変更』
setmqinst コマンドを使用すると、インストールをプライマリー・インストールとして設定する、あるいはその設定を解除することができます。

464 ページの『キュー・マネージャーとインストールの関連付け』
キュー・マネージャーを作成すると、そのキュー・マネージャーは、**crtmqm** コマンドを実行したインストールに自動的に関連付けられます。 AIX, Linux, and Windows では、**setmqm** コマンドを使用して、キュー・マネージャーに関連付けられているインストールを変更できます。

ALW 複数のインストールを使用するアプリケーションの制限

CICS サーバー・ライブラリー、ファスト・パス接続、メッセージ・ハンドル、および出口を複数インストール環境で使用する場合、いくつかの制限があります。

CICS サーバー・ライブラリー

CICS サーバー・ライブラリーを使用している場合、IBM MQ は正しいライブラリー・レベルを自動的に選択しません。 アプリケーションが接続するキュー・マネージャーに対する適切なライブラリー・レベルを使用して、ユーザーがアプリケーションをコンパイルおよびリンクする必要があります。 詳しくは、TXSeries®Multiplatforms バージョン 5 で使用するためのライブラリーの作成を参照してください。

メッセージ・ハンドル

特殊値 MQHC_UNASSOCIATED_HCONN を使用するメッセージ・ハンドルは、プロセスでロードされる最初のインストールでの使用に制限されています。 特定のインストールによってこのメッセージ・ハンドルが使用できない場合は、理由コード MQRC_HMSG_NOT_AVAILABLE が返されます。

この制限は、メッセージ・プロパティーに影響を与えます。 あるインストール上のキュー・マネージャーから、メッセージ・ハンドルを使用してメッセージ・プロパティーを取得し、それらを別のインストール上のキュー・マネージャーに書き込むことはできません。 メッセージ・ハンドルについて詳しくは、MQCRTMH - メッセージ・ハンドルの作成を参照してください。

出口

複数インストール環境では、 IBM MQ インストール済み環境で使用するために、既存の出口を更新する必要があります。 **crtmqcvx** コマンドを使用して生成されたデータ変換出口は、更新されたコマンドを使用して再生成する必要があります。

すべての出口は、MQIEP 構造体を使用して記述される必要があり、埋め込まれた RPATH を使用して IBM MQ ライブラリーを指定できず、IBM MQ ライブラリーにリンクすることはできません。 詳しくは、出口とインストール可能サービスの作成 (AIX, Linux, and Windows) を参照してください。

ファスト・パス

複数のインストール済み環境があるサーバーでは、 IBM MQ へのファスト・パス接続を使用するアプリケーションは、以下の規則に従う必要があります。

1. キュー・マネージャーは、アプリケーションが IBM MQ ランタイム・ライブラリーをロードしたときに使用したのと同じインストールに関連付けられている必要があります。 アプリケーションは、別のインストールに関連付けられているキュー・マネージャーに対してファスト・パス接続を使用してはなりません。 接続しようと試みると、エラーが発生し、理由コード MQRC_INSTALLATION_MISMATCH が返されます。
2. アプリケーションが IBM MQ ランタイム・ライブラリーをロードしたときに使用したのと同じインストールに関連付けられているキュー・マネージャーに対して非ファスト・パスで接続すると、以下の条件が両方とも真である場合を除いて、このアプリケーションはファスト・パス接続できなくなります。
   - このアプリケーションが、同じインストールに関連付けられているキュー・マネージャーとの最初の接続に、ファスト・パス接続を使用した。
   - 環境変数 AMQ_SINGLE_INSTALLATION が設定されます。
3. IBM MQ インストール済み環境に関連付けられたキュー・マネージャーに非ファスト・パスを接続しても、アプリケーションがファスト・パスを接続できるかどうかには影響しません。

AMQ_SINGLE_INSTALLATION が設定されていると、キュー・マネージャーへの任意の接続をファスト・パス接続にすることができます。 それ以外は、以下のように、ほぼ同様の制限が適用されます。

- インストールは、IBM MQ ランタイム・ライブラリーをロードするために使用したものと同じでなければならない。

- 同一プロセス上のすべての接続は、同一インストールに対してでなければならない。 異なるインストール済み環境に関連付けられているキュー・マネージャーに接続しようと試みると、その接続は理由コード MQRC_INSTALLATION_MISMATCH で失敗します。 AMQ_SINGLE_INSTALLATION が設定されていると、この制限はファスト・パス接続だけでなく、すべての接続に適用されることに注意してください。
- 1 つのキュー・マネージャーに対してのみ、ファスト・パス接続する。

関連資料

MQCONNX - キュー・マネージャーの接続 (拡張)
MQIEP 構造体
2583 (0A17) (RC2583): MQRC_INSTALLATION_MISMATCH
2587 (0A1B) (RC2587): MQRC_HMSG_NOT_AVAILABLE
2590 (0A1E) (RC2590): MQRC_FASTPATH_NOT_AVAILABLE

ALW 複数のインストール環境での .NET アプリケーションの接続

デフォルトでは、プライマリー・インストール済み環境の .NET アセンブリーがアプリケーションで使用されます。 プライマリー・インストール済み環境がない場合、またはプライマリー・インストール済み環境のアセンブリーを使用しない場合は、アプリケーションの構成ファイル、または *DEVPATH* 環境変数を更新する必要があります。

システム上にプライマリー・インストール済み環境がある場合、そのインストール済み環境の .NET アセンブリーとポリシー・ファイルはグローバル・アセンブリー・キャッシュ (GAC) に登録されています。 他のすべてのインストール済み環境の .NET アセンブリーは、各インストール済み環境のインストール・パスにありますが、それらのアセンブリーは GAC には登録されていません。 したがって、デフォルトでは、プライマリー・インストール済み環境の .NET アセンブリーを使用して、アプリケーションが実行されます。 以下のいずれかの状況に当てはまる場合は、アプリケーション構成ファイルを更新する必要があります。

- プライマリー・インストール済み環境がない。
- アプリケーションでプライマリー・インストール済み環境のアセンブリーを使用しない。
- アプリケーションのコンパイル時に使用したバージョンよりも古いバージョンの IBM MQ が、プライマリー・インストール済み環境として使用されている。

アプリケーション構成ファイルの更新方法については、461 ページの『アプリケーション構成ファイルを使用する .NET アプリケーションの接続』を参照してください。

次の状況に当てはまる場合は、*DEVPATH* 環境変数を更新する必要があります。

- アプリケーションでプライマリー以外のインストール済み環境のアセンブリーを使用するが、プライマリー・インストールのバージョンがそのプライマリー以外のインストールのバージョンと同じである。

DEVPATH 変数の更新方法について詳しくは、462 ページの『DEVPATH を使用する .NET アプリケーションの接続』を参照してください。

アプリケーション構成ファイルを使用する .NET アプリケーションの接続

アプリケーション構成ファイル内に、さまざまなタグを設定して、プライマリー・インストール済み環境以外のアセンブリーを使用するようにアプリケーションをリダイレクトする必要があります。

以下の表には、特定のアセンブリーを使用して .NET アプリケーションを接続できるようにするために、アプリケーション構成ファイルに加える必要がある具体的な変更の内容が示されています。

表 29. 特定のアセンブリーを使用するためのアプリケーションの構成		
	古いバージョンの IBM MQ でコンパイルされたアプリケーション	**新しいバージョンの IBM MQ でコンパイルされたアプリケーション**
新しいバージョンの IBM MQ プライマリー・インストール済み環境でアプリケーションを実行する場合 (GAC の新しいバージョンのアセンブリーを使用):	変更の必要はありません。	変更の必要はありません。
古いバージョンの IBM MQ プライマリー・インストール済み環境でアプリケーションを実行する場合 (GAC の古いバージョンのアセンブリーを使用):	変更の必要はありません。	アプリケーション構成ファイルで以下のようにします。 • *bindingRedirect* タグを使用して、GAC にある古いバージョンのアセンブリーを使用するように指示する。
新しいバージョンの IBM MQ のプライマリー以外のインストール済み環境でアプリケーションを実行する場合 (インストール・フォルダーの新しいバージョンのアセンブリーを使用):	アプリケーション構成ファイルで以下のようにします。 • *codebase* タグを使用して、新しいバージョンのアセンブリーの場所を指すようにする。 • *bindingRedirect* タグを使用して、新しいバージョンのアセンブリーを使用するように指示する。	アプリケーション構成ファイルで以下のようにします。 • *codebase* タグを使用して、新しいバージョンのアセンブリーの場所を指すようにする。
古いバージョンの IBM MQ のプライマリー以外のインストール済み環境でアプリケーションを実行する場合 (インストール・フォルダーの古いバージョンのアセンブリーを使用):	アプリケーション構成ファイルで以下のようにします。 • *codebase* タグを使用して、古いバージョンのアセンブリーの場所を指すようにする。 • *publisherpolicy Apply=no* タグを組み込む。	アプリケーション構成ファイルで以下のようにします。 • *codebase* タグを使用して、古いバージョンのアセンブリーの場所を指すようにする。 • *bindingRedirect* タグを使用して、古いバージョンのアセンブリーを使用するように指示する。 • *publisherpolicy Apply=no* タグを組み込む。

サンプルのアプリケーション構成ファイル NonPrimaryRedirect.config は、*MQ_INSTALLATION_PATH*\tools\dotnet\samples\base フォルダーに入れられて出荷されています。 このファイルは、プライマリー以外のインストール済み環境の IBM MQ インストール・パスを使用して変更できます。 また、このファイルは、*linkedConfiguration* タグを使用して、他の構成ファイルに直接含めることができます。 nmqsget.exe.config および nmqsput.exe.config のサンプルが用意されています。 両方のサンプルで *linkedConfiguration* タグが使用されます。これらのサンプルには NonPrimaryRedirect.config ファイルが含まれています。

DEVPATH を使用する .NET アプリケーションの接続

DEVPATH 環境変数を使用して、アセンブリーを検索することができます。 *DEVPATH* 変数で指定されたアセンブリーは、GAC のどのアセンブリーよりも優先して使用されます。 この変数をいつ使用するかについて詳しくは、該当する Microsoft の *DEVPATH* に関する資料を参照してください。

DEVPATH 環境変数を使用してアセンブリーを検索するには、使用するアセンブリーを含むフォルダーに *DEVPATH* 変数を設定する必要があります。 その後、アプリケーション構成ファイルを更新し、以下のランタイム構成情報を追加する必要があります。

```
<configuration>
<runtime>
<developmentMode developerInstallation="true" />
</runtime>
</configuration>
```

関連概念

455 ページの『複数のインストール環境でのアプリケーションの接続』

AIX, Linux, and Windows システムでは、 IBM MQ ライブラリーがロードされると、 IBM MQ は自動的に適切なライブラリーを使用します。これ以上の処置は必要ありません。 IBM MQ は、アプリケーションが接続されるキュー・マネージャーと関連付けられているインストール済み環境からのライブラリーを使用します。

複数のインストール

関連タスク

プライマリー・インストールの選択

.NET の使用

ALW プライマリー・インストールの変更

setmqinst コマンドを使用すると、インストールをプライマリー・インストールとして設定する、あるいはその設定を解除することができます。

このタスクについて

このタスクは AIX, Linux, and Windows に適用されます。

プライマリー・インストールとは、必要とされている、システム全体が関係する場所が参照するインストールです。 プライマリー・インストールと、プライマリー・インストールを選択するための考慮事項について詳しくは、プライマリー・インストールの選択を参照してください。

Windows Windows でのインストール・プロセス中に、そのインストールをプライマリー・インストールにすることを指定できます。

Linux AIX AIX and Linux システムの場合は、インストール後に **setmqinst** コマンドを実行して、そのインストールをプライマリー・インストールとして設定する必要があります。

手順

- インストールをプライマリー・インストールとして設定するには、以下のようにします。
  a) 次のコマンドを入力して、インストールが既にプライマリー・インストールになっているかどうかを確認します。

  ```
  MQ_INSTALLATION_PATH/bin/dspmqinst
  ```

  ここで、 *MQ_INSTALLATION_PATH* は IBM MQ インストールのインストール・パスです。
  b) 既存の IBM MQ インストール済み環境がプライマリー・インストールとして設定されている場合は、次のステップに進む前に その設定を解除 してください。
  c) 適切な権限を使用してログオンしていることを確認します。
  – Linux AIX AIX and Linux では root として。
  – Windows Windows システムでは Administrators グループのメンバーとして。
  d) 次のいずれかのコマンドを入力します。

– プライマリー・インストールにするインストールのパスを使用してプライマリー・インストールを設定するには、以下のようにします。

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -p MQ_INSTALLATION_PATH
```

– プライマリー・インストールにするインストールの名前を使用してプライマリー・インストールを設定するには、以下のようにします。

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

e) Windows
Windows システムの場合は、システムを再始動します。

- プライマリー・インストールとして設定されているインストールの設定解除を行うには、以下のようにします。

a) 次のコマンドを入力して、プライマリー・インストールになっているインストールを確認します。

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

ここで、 *MQ_INSTALLATION_PATH* は IBM MQ インストールのインストール・パスです。

b) 適切な権限を使用してログオンしていることを確認します。

– Linux AIX AIX and Linux では root として。

– Windows Windows システムでは Administrators グループのメンバーとして。

- 次のいずれかのコマンドを入力します。

– プライマリー・インストールにしておく必要がなくなったインストールのパスを使用してプライマリー・インストールを設定解除するには、以下のようにします。

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -p MQ_INSTALLATION_PATH
```

– プライマリー・インストールにしておく必要がなくなったインストールの名前を使用してプライマリー・インストールを設定解除するには、以下のようにします。

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

関連タスク
プライマリー・インストールのアンインストール、アップグレード、および保守
インストール名の選択
関連資料
Windows 上のプライマリー・インストールでのみ使用できる機能
AIX and Linux 上のプライマリー・インストールへの外部ライブラリーおよび制御コマンド・リンク
setmqinst

ALW キュー・マネージャーとインストールの関連付け

キュー・マネージャーを作成すると、そのキュー・マネージャーは、**crtmqm** コマンドを実行したインストールに自動的に関連付けられます。 AIX, Linux, and Windows では、**setmqm** コマンドを使用して、キュー・マネージャーに関連付けられているインストールを変更できます。

このタスクについて

キュー・マネージャーが関連付けられているインストールは、そのキュー・マネージャーに制限を課します。 つまり、そのキュー・マネージャーは、そのインストールからのコマンドでしか管理できないのです。これには、以下に示した重要な 3 つの例外があります。

- **setmqm** は、キュー・マネージャーに関連付けられているインストールを変更します。 このコマンドは、キュー・マネージャーが現在関連付けられているインストールからではなく、キュー・マネージャーに関

連付けたいインストールから実行しなければなりません。 **setmqm** コマンドによって指定されたインストール名は、このコマンドの実行元であるインストールに一致している必要があります。

- **strmqm** は、キュー・マネージャーに関連付けられているインストール済み環境から発行する必要があります。
- **dspmq** は、**dspmq** コマンドと同じインストールに関連付けられているキュー・マネージャーだけでなく、システム上のすべてのキュー・マネージャーに関する情報を表示します。 `dspmq -o installation` コマンドは、どのキュー・マネージャーがどのインストールに関連付けられているのかについての情報を表示します。

HA 環境の場合、**addmqinf** コマンドは **addmqinf** コマンドの実行元のインストールにキュー・マネージャーを自動的に関連付けます。 **strmqm** コマンドが **addmqinf** コマンドと同じインストールから実行される限り、これ以上のセットアップは必要ありません。 別のインストールを使用しているキュー・マネージャーを始動するには、最初に **setmqm** コマンドを使用して、そのキュー・マネージャーに関連付けられているインストールを変更する必要があります。

キュー・マネージャーをインストール済み環境に関連付ける場合は、**setmqm** コマンドを以下の方法で使用できます。

- IBM MQ の等価のバージョン間で個々のキュー・マネージャーを移動する。 例えば、テスト・システムから実動システムにキュー・マネージャーを移動します。
- 古いバージョンの IBM MQ から新しいバージョンの IBM MQ に個々のキュー・マネージャーをマイグレーションする。 キュー・マネージャーをバージョン間でマイグレーションすることには、留意しなければならないさまざまな影響があります。 マイグレーションの詳細については、保守とマイグレーションを参照してください。

手順

1. 現在キュー・マネージャーに関連付けられているインストールから **endmqm** コマンドを使用してそのキュー・マネージャーを停止します。
2. 別のインストールから **setmqm** コマンドを使用して、キュー・マネージャーをそのインストールに関連付けます。

   例えば、Installation2 という名前のインストールに関連付けられるようキュー・マネージャー QMB を設定するには、Installation2 から次のコマンドを入力します。

   ```
   MQ_INSTALLATION_PATH/bin/setmqm -m QMB -n Installation2
   ```

   ここで、*MQ_INSTALLATION_PATH* は、Installation2 がインストールされている場所のパスです。
3. 現在キュー・マネージャーに関連付けられているインストールから **strmqm** コマンドを使用して、そのキュー・マネージャーを始動します。

   このコマンドは、必要とされるキュー・マネージャーのマイグレーションを実行し、結果として、そのキュー・マネージャーを使用する準備が完了します。

次のタスク

キュー・マネージャーが関連付けられているインストールが削除されていた場合、またはキュー・マネージャーの状況情報を使用できない場合、**setmqm** コマンドはキュー・マネージャーと別のインストールとの関連付けに失敗します。 この状況では、以下のアクションを実行します。

1. **dspmqinst** コマンドを使用して、システム上の別のインストールを表示します。
2. mqs.ini 内の QueueManager スタンザの InstallationName フィールドを手動で変更し、別のインストールを指定します。
3. そのインストールから **dltmqm** コマンドを使用して、キュー・マネージャーを削除します。

関連概念

466 ページの『システム上の IBM MQ のインストールの見つけ方』

システムに複数の IBM MQ インストールがある場合、インストールされているバージョンとそれらの場所を確認できます。

86 ページの『IBM MQ 構成ファイル mqs.ini』
IBM MQ 構成ファイル `mqs.ini` には、ノード上のすべてのキュー・マネージャーに関連する情報が含まれています。 これは、インストール時に自動的に作成されます。

関連タスク
プライマリー・インストールの選択
関連資料
addmqinf
dspmq
dspmqinst
endmqm
setmqm
strmqm

ALW システム上の IBM MQ のインストールの見つけ方

システムに複数の IBM MQ インストールがある場合、インストールされているバージョンとそれらの場所を確認できます。

システム上の IBM MQ のインストールを見つけるには、以下の方法を使用します。

- IBM MQ がインストールされた場所を照会するには、プラットフォームのインストール・ツールを使用します。 次に、 IBM MQ インストール済み環境から **dspmqver** コマンドを使用します。 以下のコマンドは、IBM MQ のインストール先を照会するために使用できるコマンドの例です。
  - AIX AIX システムの場合は、**lslpp** コマンドを使用できます。

    ```
    lslpp -R ALL -l mqm.base.runtime
    ```

  - Linux Linux システムの場合は、**rpm** コマンドを使用できます。

    ```
    rpm -qa --qf "%{NAME}-%{VERSION}-%{RELEASE}\t%{INSTPREFIXES}\n" | grep MQSeriesRuntime
    ```

  - Windows Windows システムの場合は、**wmic** コマンドを使用できます。 このコマンドでは、wmic クライアントをインストールできます。

    ```
    wmic product where "(Name like '%MQ%') AND (not Name like '%bitSupport')" get Name,
    Version, InstallLocation
    ```

- Linux AIX AIX and Linux システムでは、以下のコマンドを実行して、IBM MQ がインストールされている場所を見つけます。

  ```
  cat /etc/opt/mqm/mqinst.ini
  ```

  次に、 IBM MQ インストール済み環境から **dspmqver** コマンドを使用します。
- Windows 32 ビット Windows でシステム上のインストール済み環境の詳細を表示するには、次のコマンドを実行します。

  ```
  reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation" /s
  ```

- Windows 64 ビット Windows では、次のコマンドを実行します。

  ```
  reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IBM\WebSphere MQ\Installation" /s
  ```

関連資料
dspmqver

dspmqinst
複数のインストール

高可用性、リカバリー、および再始動の構成

キュー・マネージャーに障害が起きた場合にキューの可用性を維持したり、サーバーやストレージの障害後にメッセージをリカバリーしたりして、アプリケーションの可用性を高めます。

このタスクについて

z/OS z/OS では、高可用性がプラットフォームに組み込まれています。 共用キューおよびキュー共用グループを参照してください。

Multi マルチプラットフォーム では、クライアント・アプリケーションの可用性を高めるために、クライアントの再接続機能を使用して、キュー・マネージャーのグループ間でクライアントを自動的に切り替えたり、キュー・マネージャーの障害後に複数インスタンス・キュー・マネージャーの新しいアクティブ・インスタンスに切り替えたりできます。 IBM MQ classes for Java は自動クライアント再接続をサポートしていません。 複数インスタンスキュー・マネージャーを複数のサーバーで 1 つのキュー・マネージャーとして実行するように構成します。 そのキュー・マネージャーにサーバー・アプリケーションをデプロイしてください。 アクティブ・インスタンスを実行しているサーバーに障害が起きた場合、別のサーバー上にある同じキュー・マネージャーのスタンバイ・インスタンスに実行が自動的に切り替えられます。 サーバー・アプリケーションをキュー・マネージャー・サービスとして実行するように構成した場合、スタンバイ・インスタンスが、アクティブに実行しているキュー・マネージャー・インスタンスになると、サーバー・アプリケーションは再始動されます。

Multiplatforms でサーバー・アプリケーションの可用性を高めるためのもう 1 つの方法は、キュー・マネージャー・クラスター内の複数のコンピューターにサーバー・アプリケーションをデプロイすることです。 IBM WebSphere MQ 7.1 からは、クラスター・エラー・リカバリー機能によって、問題が解決されるまで、問題の原因となった操作が再実行されます。 z/OS 以外のサーバーにおけるクラスター・エラー・リカバリーの変更を参照してください。 IBM MQ for Multiplatforms をプラットフォーム固有の次のようなクラスタリング・ソリューションの一部として構成することも可能です。

- Microsoft Cluster Server
- IBM i IBM i での HA クラスター
- Linux AIX PowerHA®forAIX(以前 HACMP オン AIX)およびその他の UNIX and Linux クラスタリングの対応策

Linux Linux システムでは、高可用性や災害復旧のソリューションを実装するための、複製データ・キュー・マネージャー (RDQM) を構成できます。 高可用性に対しては、3 つの Linux サーバーからなるグループ内の各ノードで同じキュー・マネージャーのインスタンスを構成します。 3 つのインスタンスのいずれかがアクティブ・インスタンスです。 アクティブ・キュー・マネージャーからのデータは、他の 2 つのインスタンスに同期的に複製されるため、障害が発生した場合はこれらのインスタンスのいずれかが処理を引き継ぐことができます。 災害復旧に対しては、1 つのサイトの 1 次ノードで 1 つのキュー・マネージャーを実行し、別のサイトのリカバリー・ノードでそのキュー・マネージャーの 2 次インスタンスを実行します。 1 次インスタンスと 2 次インスタンスの間でデータを複製し、何かの理由で 1 次ノードが失われた場合は、2 次インスタンスを 1 次インスタンスにして開始します。

CP4I ネイティブ HA は、コンテナーを対象とした高可用性ソリューションです。 ネイティブ HA は、ログ複製を使用して、異なるノード上で稼働しているキュー・マネージャーの 3 つのインスタンスを最新の状態に維持します。 一度に 1 つのインスタンスがアクティブになり、メッセージが処理されます。 アクティブ・キュー・マネージャーは、ログ更新を他の 2 つのインスタンスに送信して、それらを更新します。 アクティブ・インスタンスに障害が発生すると、レプリカ・インスタンスの 1 つが自動的にアクティブ・ロールを引き継ぎます。

MQ Appliance 高可用性または災害復旧のソリューションの別のオプションは、IBM MQ アプライアンスのペアをデプロイすることです。 IBM MQ Appliance 資料の 高可用性 および 災害復旧 を参照してください。

メッセージング・システムでは、システムに入ってきたメッセージが確実にその宛先に配布されます。 IBM MQ は **dspmqrte** コマンドを使用して、あるキュー・マネージャーから別のキュー・マネージャーに移動するときに、メッセージの経路をトレースできます。 システムに障害が起きた場合、障害のタイプやシステムの構成方法によって、様々な方法でメッセージをリカバリーできます。 IBM MQ では、メッセージの受信や送信を取り扱うキュー・マネージャーのアクティビティーのリカバリー・ログが維持されています。 WebSphere MQ は、次の 3 つのタイプのリカバリーに関してログを使用します。

1. 再始動リカバリー。計画どおりに IBM MQ を停止した場合
2. 障害リカバリー。障害によって IBM MQ が停止した場合
3. メディア・リカバリー。損傷したオブジェクトを復元する場合

いずれの場合も、リカバリーによってキュー・マネージャーは、停止した時の状態に復元されます。ただし、不完了トランザクションはロールバックされ、キュー・マネージャーの停止時に未完了であった更新がある場合は、この更新がキューから除去されます。 リカバリーによって、持続メッセージはすべて復元されます。このプロセス中に、非持続メッセージが失われることがあります。

注意 : リカバリー・ログを別のオペレーティング・システムに移動することはできません。

クライアントの自動再接続

追加のコードを作成しなくても、いくつかのコンポーネントを構成することによって、クライアント・アプリケーションが自動的に再接続するようにすることができます。

クライアントの自動再接続はインラインです。 接続はクライアント・アプリケーション・プログラムのどのポイントでも自動的に復元され、オブジェクトを開くためのハンドルがすべて復元されます。

対照的に、手動再接続では、クライアント・アプリケーションで MQCONN または MQCONNX を使用して接続を再作成し、オブジェクトを再オープンする必要があります。 クライアントの自動再接続は多くのクライアント・アプリケーションに適していますが、すべてのクライアント・アプリケーションに適しているわけではありません。

469 ページの表 30 は、クライアント・ワークステーションにインストールする必要のある、IBM MQ クライアント・サポートの最新のリリースのリストです。 アプリケーションがクライアントの自動再接続を使用できるようにするには、示されているレベルの 1 つにクライアント・ワークステーションをアップグレードする必要があります。 469 ページの表 31 には、クライアント自動再接続を有効にするためのその他の要件がリストされています。

再接続オプションへのプログラム・アクセスによって、クライアント・アプリケーションは再接続オプションを設定できます。 さらに、JMS および XMS クライアント以外では、クライアント・アプリケーションが再接続オプションにアクセスする場合、再接続イベントを扱うためにイベント・ハンドラーを作成することができます。

既存のクライアント・アプリケーションでは、以下のようにして、再コンパイルおよびリンクせずに、再接続のサポートのメリットを受けられます。

- JMS 以外のクライアントの場合は、mqclient.ini 環境変数 DefRecon を設定して再接続オプションを設定します。 キュー・マネージャーに接続するには CCDT を使用します。 クライアントがマルチ・インスタンスのキュー・マネージャーに接続する場合、CCDT にあるアクティブで待機状態のキュー・マネージャー・インスタンスのネットワーク・アドレスを提供します。 複製したデータ・キュー・マネージャーの場合、または IBM MQ Appliance 上の HA キュー・マネージャーの場合、構成を単純化するために、アクティブおよびスタンバイの両方のキュー・マネージャーで使用される浮動 IP アドレスを指定できます。
- JMS クライアントでは、接続ファクトリー構成に再接続オプションを設定します。 Java EE サーバーの EJB コンテナー内で実行している場合、MDB は、IBM MQ リソース・アダプターのアクティベーション・スペック (WebSphere Application Server で実行している場合はリスナー・ポート) によって提供される再接続メカニズムを使用して、 IBM MQ に再接続できます。 ただし、アプリケーションが MDB でない (または Web コンテナー内で実行されている) 場合、このシナリオではクライアントの自動再接続がサポートされていないため、アプリケーションは独自の再接続ロジックを実装する必要があります。 IBM MQ リソース・アダプターは、メッセージ駆動型 Bean にメッセージを送達するための再接続機能を提供しま

すが、サーブレットのようなその他の Java EE エレメントは、それ自体の再接続を実装する必要があります。

注 : IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

表 *30.* サポートされるクライアント

クライアント・インターフェース	クライアント	再接続オプションへのプログラム・アクセス	再接続サポート
API メッセージング	C、C++、COBOL、非管理の Visual Basic、XMS (Windows 上の管理対象でない XMS)	7.0.1	7.0.1
	JMS (JSE、および Java EE クライアント・コンテナーと管理対象のコンテナー)	7.0.1.3	7.0.1.3
	IBM MQ classes for Java	サポート対象外	サポート対象外
	管理対象の XMS および管理対象の .NET クライアント: C#、Visual Basic	7.1	7.1
その他の API	Windows Communication Foundation (管理対象でない [1])	サポート対象外	7.0.1
	Windows Communication Foundation (管理対象 [1])	サポート対象外	サポート対象外
	Axis 1	サポート対象外	サポート対象外
	Axis 2	サポート対象外	7.0.1.3
	HTTP (Web 2.0)	サポート対象外	7.0.1.3

1. WCF バインディング構成で、管理対象か管理対象でないかを設定します。

自動再接続には、以下の構成要件があります。

表 *31.* 自動再接続の構成要件

コンポーネント	要件	要件に適合しない場合の影響
IBM MQ MQI client インストール	469 ページの表 30 を参照	MQRC_OPTIONS_ERROR
IBM MQ サーバー・インストール	レベル 7.0.1	MQRC_OPTIONS_ERROR
チャネル	SHARECNV > 0	MQRC_ENVIRONMENT_ERROR
アプリケーション環境	スレッド化必須	MQRC_ENVIRONMENT_ERROR

表 31. 自動再接続の構成要件 (続き)		
コンポーネント	**要件**	**要件に適合しない場合の影響**
MQI	以下のいずれかです。 • MQCONNX で MQCNO オプションを MQCNO_RECONNECT または MQCNO_RECONNECT_Q_MGR に設定。 • Defrecon=YES|QMGR (mqclient.ini) • JMS セットで、接続ファクトリーの **CLIENTRECONNECTOPTIONS** プロパティーを設定します。	接続が切断された場合、またはキュー・マネージャーが終了または失敗した場合は MQCC_FAILED。

470 ページの図 68 は、クライアントの再接続に関係するコンポーネント間の主な相互作用を示しています。

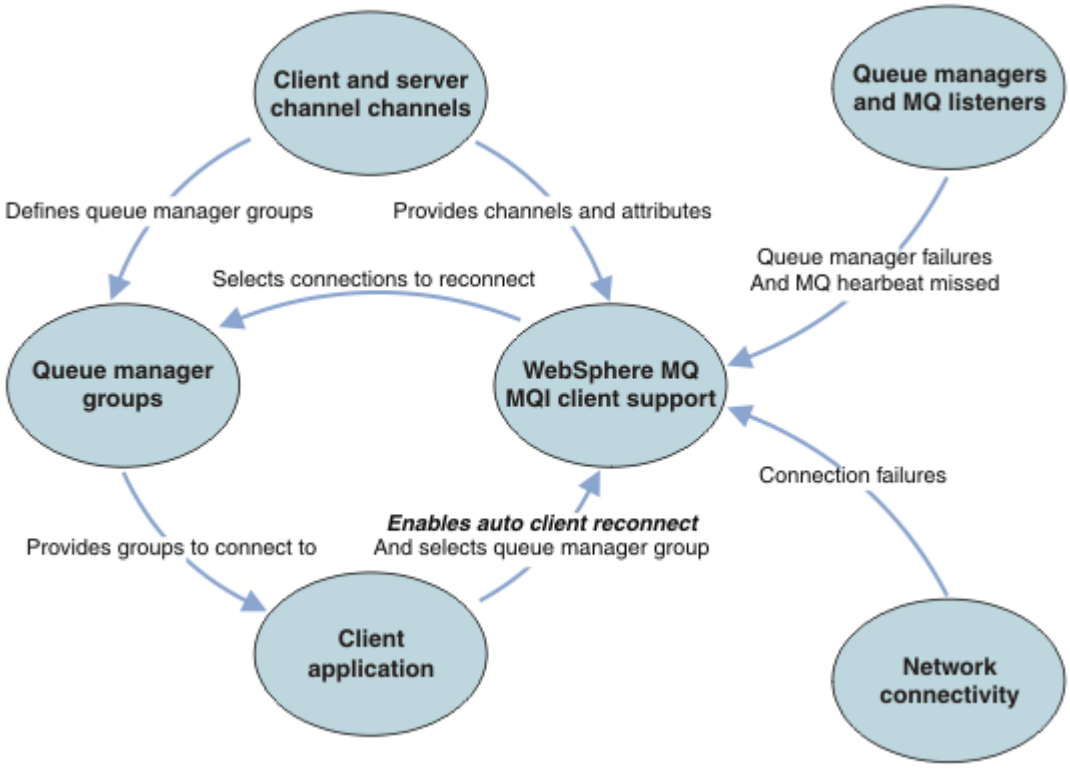


図 68. クライアントの自動再接続

クライアント・ アプリケーション

クライアント・アプリケーションは IBM MQ MQI client です。 JMS クライアントの自動クライアント再接続について詳しくは、自動 JMS クライアント再接続の使用を参照してください。

- デフォルトでは、クライアントの自動再接続は行われません。 クライアントの自動再接続は、MQCONNX で MQCNO オプション MQCNO_RECONNECT または MQCNO_RECONNECT_Q_MGR を設定することにより有効になります。

- 追加のコーディングを行わなくても自動再接続を利用できるように、多くのアプリケーションがこのような方法で作成されています。 mqclient.ini 構成ファイルのチャネル・スタンザにデフレクコンの属性を設定することによって、コーディングの変更を行わずに、既存のプログラムの自動再接続を使用可能にします。
- 以下の 3 つの選択肢のいずれかを使用します。

  1. ロジックが再接続の影響を受けないようにプログラムを変更する。 例えば、同期点内で MQI 呼び出しを発行し、バックアウトされたトランザクションを再実行依頼しなければならない場合があります。 非同期コンシューマーは、トランザクションがバックアウトされた場合に「中断」されているかどうかを確認する必要があります。
  2. 再接続を検出するイベント・ハンドラーを追加し、接続が再度確立されたときに、クライアント・アプリケーションの状態を復元する。
  3. 自動再接続を有効にしない。代わりにクライアントを切断し、MQCONN または MQCONNX いずれかの MQI 呼び出しを新たに発行して、同じキュー・マネージャー・グループで実行されている別のキュー・マネージャー・インスタンスを見つける。

  この 3 つの選択肢について、557 ページの『アプリケーションの復旧』のトピックで詳しく説明されています。
- 同じ名前のキュー・マネージャーに再接続しても、それは同じキュー・マネージャーのインスタンスに再接続したという保証にはなりません。

  同じキュー・マネージャーのインスタンスに再接続するには、MQCNO オプション MQCNO_RECONNECT_Q_MGR を使用します。
- クライアントでは、再接続の状態に関する通知を受けられるようにイベント・ハンドラーを登録することができます。 イベント・ハンドラーで渡された MQHCONN は使用できません。 以下の理由コードが用意されています。

  MQRC_RECONNECTING
  接続に障害が起きたため、システムが再接続を試行しています。 複数の再接続が試行されると、複数の MQRC_RECONNECTING イベントを受け取ります。

  MQRC_RECONNECTED
  再接続が実行され、すべてのハンドルが正常に再設定されました。

  MQRC_RECONNECT_FAILED
  再接続は成功しませんでした。

  MQRC_RECONNECT_QMID_MISMATCH
  再接続可能な接続で MQCNO_RECONNECT_Q_MGR が指定され、接続が別のキュー・マネージャーへの再接続を試みました。

  MQRC_RECONNECT_Q_MGR_REQD
  MQGET 呼び出しの MQMO_MATCH_MSG_TOKEN といったオプションが、同じキュー・マネージャーへの再接続を必要とするクライアント・プログラムで指定されました。
- 再接続可能クライアントは、接続後に初めて自動再接続可能になります。 すなわち、MQCONNX 呼び出し自体は、失敗した場合は再試行されません。 例えば、MQCONNX から戻りコード 2543 - MQRC_STANDBY_Q_MGR を受け取った場合は、少し経ってから呼び出しを再発行する必要があります。

  MQRC_RECONNECT_INCOMPATIBLE
  この理由コードは、アプリケーションが、再接続オプションを設定する際に、MQPMO_LOGICAL_ORDER (MQPUT および MQPUT1 で)、または MQGMO_LOGICAL_ORDER (MQGET で) の使用を試みた場合に戻されます。 理由コードが戻される理由は、今後同様なケースにおいてアプリケーションが二度と再接続を使用しないようにするためです。

  MQRC_CALL_INTERRUPTED
  この理由コードは、Commit 呼び出しの実行およびクライアント再接続の最中に接続が中断された場合に戻されます。 同期点の外側の持続メッセージの MQPUT によっても、アプリケーションに同じ理由コードが戻される結果になります。

高可用性キュー・マネージャー

高可用性キュー・マネージャーには、キュー・マネージャーの 1 つのアクティブ・インスタンスと 1 つ以上のスタンバイ・インスタンスがあります。 アクティブ・キュー・マネージャーはスタンバイ・キュー・マネージャーと同期されるので、アクティブ・インスタンスで障害が発生した場合にはスタンバイが自動的にテークオーバーできます。 高可用性キュー・マネージャーを提供するための様々なソリューションが多数あり、詳しくは 479 ページの『高可用性の構成』 を参照してください。

高可用性キュー・マネージャーがスタンバイ・インスタンスをアクティブ化した後で、クライアント自動再接続を使って、単純に IBM MQ MQI client・アプリケーションを再始動できます。

高可用性キュー・マネージャーのスタンバイ・インスタンスは、通常、アクティブ・インスタンスとは異なるネットワーク・アドレス上にあります。 クライアント接続定義表 (CCDT) には、両方のインスタンスのネットワーク・アドレスが含まれます。 **CONNAME** パラメーターにネットワーク・アドレスのリストを指定するか、CCDT 内のキュー・マネージャーに複数の行を定義してください。 複製されたデータ・キュー・マネージャーと IBM MQ Appliance 高可用性キュー・マネージャーは浮動 IP アドレスをサポートします。 浮動 IP アドレスでは、アクティブ・キュー・マネージャーまたはスタンバイ・キュー・マネージャーで使用する、1 つのアドレスを指定します。

キュー・マネージャー・ グループ

一般的に、IBM MQ MQI clients はキュー・マネージャー・グループ内の任意のキュー・マネージャーに再接続します。 IBM MQ MQI client が、同じキュー・マネージャーにだけ再接続するようにしたい場合があります。 1 つのキュー・マネージャーに対して、親和性を持つ可能性があります。

クライアント・アプリケーションの接続先および再接続先が、常に同じ名前のキュー・マネージャーか、同じキュー・マネージャーか、またはクライアント接続テーブルで同じ QMNAME で定義された一連のキュー・マネージャーのうちのいずれかであるかを選択できます。

- クライアント・チャネル定義にあるキュー・マネージャーの名前属性 QMNAME は、キュー・マネージャー・グループの名前です。
- クライアント・アプリケーションで、MQCONN または MQCONNX の QmgrName パラメーターの値をキュー・マネージャー名に設定した場合、クライアントはその名前のキュー・マネージャーにのみ接続します。 キュー・マネージャー名にアスタリスク (*) の接頭部を付けた場合、クライアントは、同じ QMNAME 値を持つキュー・マネージャー・グループ内の任意のキュー・マネージャーに接続します。 詳しい説明については、CCDT のキュー・マネージャー・グループを参照してください。

クライアントが異なるキュー・マネージャーに再接続しないようにすることができます。 MQCNO オプション、MQCNO_RECONNECT_Q_MGR を設定します。 IBM MQ MQI client は、異なるキュー・マネージャーに再接続された場合、失敗します。 MQCNO オプション、MQCNO_RECONNECT_Q_MGR を設定した場合は、同じキュー・マネージャー・グループに他のキュー・マネージャーを含めないでください。 クライアントが再接続するキュー・マネージャーが以前接続したのと同じでない場合、クライアントはエラーを戻します。

キュー共有グループ

z/OS z/OS キュー共有グループへのクライアント自動再接続は、他の任意の環境における再接続と同じ仕組みを使用します。 クライアントは、元の接続のために構成したのと同じキュー・マネージャーのセットに対して再接続します。 例えば、クライアント・チャネル定義テーブルを使用する場合、管理者は、テーブルのすべての項目が、同じ z/OS キュー共有グループに解決されるようにする必要があります。

クライアントおよびサーバーのチャネル定義

クライアントおよびサーバーのチャネル定義では、クライアント・アプリケーションが再接続可能なキュー・マネージャーのグループを定義します。 この定義は、再接続の選択内容やタイミング、およびセキュリティーなどの他の要因を管理します (関連トピックを参照)。 再接続の面で最も関係のあるチャネル属性を、以下の 2 つのグループにリストします。

クライアント接続の属性
接続アフィニティー (AFFINITY) AFFINITY
接続のアフィニティー。

クライアント・チャネル・ウェイト (CLNTWGHT) CLNTWGHT
クライアント・チャネル・ウェイト。

接続名 (CONNAME) CONNAME
接続情報。

ハートビート間隔 (HBINT) HBINT
ハートビート間隔。 サーバー接続チャネルのハートビート間隔を設定します。

キープアライブ間隔 (KAINT) KAINT
キープアライブ間隔。 サーバー接続チャネルのキープアライブ間隔を設定します。

z/OS KAINT は z/OS にのみ適用されることに注意してください。

キュー・マネージャー名 (QMNAME)QMNAME
キュー・マネージャー名。

サーバー接続の属性
ハートビート間隔 (HBINT) HBINT
ハートビート間隔。 クライアント接続チャネルのハートビート間隔を設定します。

キープアライブ間隔 (KAINT) KAINT
キープアライブ間隔。 クライアント接続チャネルのキープアライブ間隔を設定します。

z/OS KAINT は z/OS にのみ適用されることに注意してください。

KAINT はネットワーク層のハートビートで、HBINT はクライアントとキュー・マネージャーの間の IBM MQ ハートビートです。 これらのハートビートをより短い時間に設定すると、以下の 2 つの効果があります。

1. 接続上のアクティビティーがシミュレートされるので、非アクティブな接続のクローズを受け持つネットワーク層ソフトウェアが接続をシャットダウンする可能性が低くなります。
2. 接続がシャットダウンされた場合に、障害の起きた接続を検出するまでの遅延時間が短くなります。

デフォルトの TCP/IP キープアライブ間隔は 2 時間です。 KAINT および HBINT 属性の設定時間を短くすることを検討してください。 ネットワークの通常動作が自動再接続の要件に適しているとは限りません。 例えば、ファイアウォールによっては、非アクティブの TCP/IP 接続をシャットダウンするまでに、少なくとも 10 分を要します。

ネットワーク 接続

ネットワークから IBM MQ MQI client に渡されるネットワーク障害のみが、クライアントの自動再接続機能によって処理されます。

- トランスポートによって自動的に実行される再接続は、IBM MQ には認識されません。
- HBINT を設定すると、IBM MQ では認識されないネットワーク障害を処理する上で役立ちます。

キュー・マネージャーおよび IBM MQ リスナー

クライアントの再接続は、サーバーの障害、キュー・マネージャーの障害、ネットワーク接続の障害、さらに管理者による別のキュー・マネージャー・インスタンスへの切り替えによって起動されます。

- 複数インスタンス・キュー・マネージャーを使用している場合は、アクティブなキュー・マネージャー・インスタンスからスタンバイ・インスタンスに制御を切り替えるときにも、クライアントの再接続が発生します。
- デフォルトの **endmqm** コマンドを使用してキュー・マネージャーを終了しても、クライアントの自動再接続はトリガーされません。 **endmqm** コマンドに、クライアント自動再接続を要求する `-r` オプション、またはシャットダウン後にスタンバイ・キュー・マネージャー・インスタンスに転送する `-s` オプションを追加してください。

IBM MQ MQI client の自動再接続サポート

IBM MQ MQI client でクライアント自動再接続サポートを使用すると、キュー・マネージャーに再接続するための MQCONN MQI 呼び出しまたは MQCONNX MQI 呼び出しを発行しなくても、クライアント・アプリケーションの再接続が自動的に実行され、クライアント・アプリケーションが処理を続行します。

- クライアント自動再接続は、以下の状況のいずれかに該当する場合にトリガーされます。
  - キュー・マネージャー障害
  - **endmqm** コマンドで -r(再接続) オプションを指定してキュー・マネージャーを終了する
- MQCONNX MQCNO オプションは、クライアント自動再接続を有効にするかどうかを制御します。 オプションについては、再接続オプションで説明されています。
- クライアント自動再接続は、アプリケーションの代わりに MQI 呼び出しを発行して、接続ハンドルおよび他の開いているオブジェクトへのハンドルを復元します。それにより、プログラムは、接続の障害により発生した MQI エラーを処理した後で、通常の処理を再開することができます。 559 ページの『自動再接続されるクライアントの復旧』を参照。
- 接続に対してチャネル出口プログラムを作成した場合は、その出口がこれらの追加の MQI 呼び出しを受け取ります。
- 再接続の開始時および終了時にトリガーされる再接続イベント・ハンドラーを登録することができます。

再接続にかかる時間は 1 分ほどですが、キュー・マネージャーに大量の管理対象リソースがあるためにそれ以上の時間がかかることがあります。 この間、クライアント・アプリケーションが IBM MQ リソースに属さないロックを保持し続ける可能性があります。 クライアントの再接続の待ち時間を制限するために構成可能なタイムアウト値が 1 つあります。 この値(秒単位)は、mqclient.ini ファイルに設定されます。

```
Channels:
MQReconnectTimeout = 1800
```

タイムアウト後、再接続は試行されなくなります。 システムは、タイムアウトになったことを検出すると、MQRC_RECONNECT_FAILED エラーを返します。

関連概念

再接続可能クライアント

関連タスク

キュー・マネージャーの停止

z/OS コンソール・メッセージのモニター

IBM MQ for z/OS では、キュー・マネージャーまたはチャネル・イニシエーターから送信される情報メッセージの中に、特に重要と見なすべきメッセージが多数存在します。 そうしたメッセージは、問題があることを示しているわけではありませんが、対応が必要になる可能性のある問題が潜んでいることを示しているので、トラッキングしておくと役立ちます。

これらのコンソール・メッセージが出されたということは、ユーザー・アプリケーションが多数のメッセージをそのページ・セットに入れたことを示している場合もあり、以下のような大きな問題の徴候である可能性があります。

- メッセージを PUT するユーザー・アプリケーションに、制御不能なループなどの問題がある。
- キューからメッセージを GET するユーザー・アプリケーションが機能しなくなっている。

モニターするコンソール・メッセージ

以下のリストは、大きな問題を示している可能性があるメッセージの概略を示しています。 システムの自動化を使用してこれらのメッセージをトラッキングする必要があるかどうかを判別し、適切な文書を用意して、すべての潜在的な問題を効果的に追跡できるようにしてください。

CSQI004I: *csect-name connection-type* の接続 *connection-name* のために、キュー *queue-name* に索引タイプ *index-type* の索引を作成することを検討してください。*num-msgs* 個のメッセージがスキップされました

- アプリケーションが、索引が定義されていないキューからメッセージ ID または相関 ID を使用してメッセージを受け取ったことを、キュー・マネージャーが検出しました。
- 特定されたキューに索引を設定するために、ローカル・キュー・オブジェクト *queue-name* の INDXTYPE 属性の値が *index-type* となるように変更することを検討してください。

CSQI031I: *csect-name* ページ・セット *psid* の新規エクステントは正常にフォーマットされました。

- このページ・セットに割り振られているキューの curdepth を確認します。
- メッセージを処理するために、障害の原因を調べます。

.

CSQI041I: *csect-name* ジョブ *jobname* ユーザー *userid* がページ・セット *psid* へのアクセス中にエラー。

- ページ・セットがキュー・マネージャーに割り振られているかどうかを判別します。
- **DISPLAY USAGE** コマンドを発行して、ページ・セットの状態を判別します。
- キュー・マネージャーのジョブ・ログに追加のエラー・メッセージがあるかどうかを確認します。

.

CSQI045I: *csect-name* ログ RBA が *rba* に達しました。 ログのリセットを計画します。

- 適当なときにキュー・マネージャーを停止し、ログをリセットする。
- キュー・マネージャーが 6 バイトのログ RBA を使用している場合は、8 バイトのログ RBA を使用するようキュー・マネージャーを変換することを検討してください。

CSQI046E: *csect-name* ログ RBA が *rba* に達しました。 ログのリセットを実行します。

- 適当なときにキュー・マネージャーを停止し、ログをリセットする。
- キュー・マネージャーが 6 バイトのログ RBA を使用している場合は、8 バイトのログ RBA を使用するようキュー・マネージャーを変換することを検討してください。

CSQI047E: *csect-name* ログ RBA が *rba* に達しました。 キュー・マネージャーを停止しログをリセットします。

- 即座にキュー・マネージャーを停止し、ログをリセットする。
- キュー・マネージャーが 6 バイトのログ RBA を使用している場合は、8 バイトのログ RBA を使用するようキュー・マネージャーを変換することを検討してください。

CSQJ004I: アクティブ・ログのコピー *n* は非活動で、単一モードです。ENDRBA=*ttt*

- キュー・マネージャーによって「single」ロギング・モードがアクティブ化されました。 これは通常、ログのオフロードの問題を示しています。
- **DISPLAY LOG** コマンドを発行して、二重のアクティブ・ログおよびアーカイブ・ログに関する設定を判別します。 この表示は、オフロード処理を必要とするアクティブ・ログの数も示します。
- キュー・マネージャーのジョブ・ログに追加のエラー・メッセージがあるかどうかを確認します。

CSQJ031D: *csect-name*、ログ RBA 範囲をリセットする必要があります。 起動を続行するには「Y」を、シャットダウンするには「N」を入力してください

- できるだけ早くキュー・マネージャーを停止し、ログをリセットします。
- キュー・マネージャーが 6 バイトのログ RBA を使用している場合は、8 バイトのログ RBA を使用するようキュー・マネージャーを変換することを検討してください。

CSQJ032E: *csect-name alert-lvl* - ログ RBA 範囲の終端 *max-rba* に近づきました。 現行のログ RBA は *current-rba* です。

- できるだけ早くキュー・マネージャーを停止してログをリセットする計画を立ててください。
- キュー・マネージャーが 6 バイトのログ RBA を使用している場合は、8 バイトのログ RBA を使用するようキュー・マネージャーを変換することを検討してください。

CSQJ110E: 活動ログ・データ・セットの最新コピー *n* は *nnn* パーセントまできています

- ログのオフロード処理に関連する未解決の要求を判別するための表示要求を実行して、他の待ちオフロード・タスクを完了する手順を実行する。 要求を満たすのに必要な処置を取り、オフロード処理を続行できるようにする。
- 十分なアクティブ・ログ・データ・セットがあるかどうかを確認する。 必要に応じて、DEFINE LOG コマンドを使用して、動的にログ・データ・セットを追加できる。

CSQJ111A: 活動ログ・データ・セットにスペースがありません

- 表示要求を実行し、ログのオフロード処理に関連して未解決の要求がないかどうかを確かめる。 要求を満たすのに必要な処置を取り、オフロード処理を続行できるようにする。
- 十分なアクティブ・ログ・データ・セットがあるかどうかを確認する。 必要に応じて、DEFINE LOG コマンドを使用して、動的にログ・データ・セットを追加できる。
- オフロードに必要な資源が不足しているために遅延が生じた場合は、 必要な資源を使用可能にして、オフロードを完了させ、ロギングを続行できるようにしなければならない。 この状態から回復する方法について詳しくは、アーカイブ・ログの問題を参照。

CSQJ114I: 保存データ・セットにエラーが発生し、オフロードは生成中の保存データ・セットでだけ継続します。

- キュー・マネージャーのジョブ・ログに追加のエラー・メッセージがあるかどうかを確認します。
- アーカイブ・ログのもう 1 つのコピーを作成して、ご使用の BSDS を手動で更新します。

CSQJ115E: オフロードが失敗しました。保存データ・セットを割り振ることができませんでした。
メッセージ CSQJ103E または CSQJ073E のエラー状況情報を調べる。 データ・セット割り振りエラーの原因となった状況を訂正し、再試行時にオフロードが行われるようにする。

CSQJ136I: 磁気テープ装置を接続 ID=*xxxx* 相関 ID=*yyyyyy* に割り振れません。*m* 個割り振られ、*n* 個許可されました。

- キュー・マネージャーのジョブ・ログに追加のエラー・メッセージがあるかどうかを確認します。

.

CSQJ151I: *csect-name* RBA *rrr* の読み取り中エラー。CONNECTION-ID=*xxxx* CORRELATION-ID=*yyyyyy* REASON CODE=*ccc*

- キュー・マネージャーのジョブ・ログに追加のメッセージがあるかどうかを確認します。
- **DISPLAY CONN** コマンドを発行して、活動をコミットしていない接続を判別します。
- アプリケーションが更新をコミットできることを確認します。

CSQJ160I: 長時間実行されている UOW が見つかりました。URID=*urid* CONNECTION NAME=*name*

- キュー・マネージャーのジョブ・ログに追加のメッセージがあるかどうかを確認します。
- **DISPLAY CONN** コマンドを発行して、活動をコミットしていない接続を判別します。
- アプリケーションが更新をコミットできることを確認します。

CSQJ161I: *n* オフロードの後、UOW が解決されませんでした。 URID=*urid* CONNECTION NAME=*name*

- ページ・セットがキュー・マネージャーに割り振られているかどうかを判別します。
- **DISPLAY USAGE** コマンドを発行して、ページ・セットの状態を判別します。
- キュー・マネージャーのジョブ・ログに追加のメッセージがあるかどうかを確認します。

CSQP011E: ページ・セット *psid* の接続エラー状況 *ret-code*

- このページ・セットに割り振られているキューの curdepth を確認します。
- メッセージを処理するために、障害の原因を調べます。

CSQP013I: *csect-name* ページ・セット psid に作成された新規エクステントです。 新規エクステントは今フォーマットされます。

- このページ・セットに割り振られているキューの curdepth を確認します。

- メッセージを処理するために、障害の原因を調べます。
- キューを別のページ・セットに再配置する必要があるかどうかを判別します。
- ボリュームが満杯の場合には、ページ・セットをマルチボリュームのデータ・セットにする必要があるかどうかを判別します。 ページ・セットがすでにマルチボリュームになっている場合は、使用しているストレージ・グループにさらにボリュームを追加することを検討します。 使用できるスペースが増えたら、ページ・セットの **EXPAND** メソッドを **SYSTEM** に設定して、拡張を再試行します。 再試行が必要な場合は、 **EXPAND** を **SYSTEM** に切り替えて、通常の設定に戻します。

CSQP014E: *csect-name* ページ・セット psid の拡張が失敗しました。 これ以降の拡張要求は拒否されます。

- このページ・セットに割り振られているキューの curdepth を確認します。
- メッセージを処理するために、障害の原因を調べます。
- キューを別のページ・セットに再配置する必要があるかどうかを判別します。

CSQP016E: *csect-name* ページ・セット *psid* は範囲の最大数に達しました。 もう一度拡張はできません。

- このページ・セットに割り振られているキューの curdepth を確認します。
- メッセージを処理するために、障害の原因を調べます。

CSQP017I: *csect-name* ページ・セット *psid* の拡張を開始しました。
DISPLAY THREAD コマンドを発行して、IBM MQ 内の作業単位の状態を判別します。

CSQP047E: ページ・セットが使用不可である場合、問題が発生する可能性があります。この状態を訂正するアクションをとってください。

- システム・プログラマー応答に従ってください。

CSQQ008I: *nn* の回復単位がキュー・マネージャー *qqqq* において、まだ未確定です。

- 送達不能キューの状態を調べます。 送達不能キューの PUT が使用不可に設定されていないことを確認します。
- 送達不能キューが MAXMSG の限度に達していないことを確認します。

CSQQ113I: *psb-name region-id* このメッセージを処理できません。

- CSQOUTX データ・セットを確認して、CSQINPX 障害の原因を判別します。
- 一部のコマンドは処理されない可能性があります。

CSQX035I: *csect-name* キュー・マネージャー *qmgr-name* への接続が停止中、または壊れています。 MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)。

- MQRC を確認して、障害の原因を判別します。
- これらのコードについては、 IBM MQ for z/OS のメッセージ、完了コード、および理由コードに記載されています。

CSQX032I: *csect-name* 初期化コマンド・ハンドラーが終了しました。

- MQRC を確認して、障害の原因を判別します。
- これらのコードについては、 IBM MQ for z/OS のメッセージ、完了コード、および理由コードに記載されています。

CSQX048I: *csect-name name* のメッセージを変換できません。 MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)。

- ジョブ・ログを確認して、TCP/IP 障害の原因を判別します。
- TCP/IP アドレス・スペースのエラーを確認します。

CSQX234I: *csect-name* リスナーが停止しました。 TRPTYPE=*trptype* INDISP=*disposition*

- リスナーが停止しない場合は、**STOP** コマンドに続いて、TCP/IP アドレス・スペースのエラーを確認してください。
- システム・プログラマー応答に従ってください。

CSQX407I: *csect-name* クラスター・キュー *q-name* の定義が矛盾しています。

- クラスター内の複数のクラスター・キューの値が矛盾しています。 相違を調べて解決してください。

CSQX411I: *csect-name* リポジトリー・マネージャーが停止しました。

- エラーが原因でリポジトリー・マネージャーが停止した場合は、ジョブ・ログでメッセージを確認してください。

CSQX417I: *csect-name* 除去されたキュー・マネージャー *qmgr-name* のクラスター送信者が残っています。

- システム・プログラマー応答に従ってください。

CSQX418I: *csect-name* クラスター *cluster_name* にリポジトリーが 1 つしかありません。

- 高可用性をさらに高めるには、2 つの完全リポジトリーを使用するようにクラスターを構成する必要があります。

CSQX419I: *csect-name* クラスター *cluster_name* にクラスター受信側がありません。

- システム・プログラマー応答に従ってください。

CSQX420I: *csect-name* クラスター *cluster_name* にリポジトリーがありません。

- システム・プログラマー応答に従ってください。

CSQX448E: *csect-name* リポジトリー・マネージャーがエラーのため停止しています。 *n* 秒後に再始動します。

- システム・プログラマー応答に従ってください。

  このメッセージは、以下のコマンドを使用して SYSTEM.CLUSTER.COMMAND.QUEUE を有効にするまで 600 秒 (10 分) ごとに書き出されます。

  ```
  ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
  ```

  キューを有効にする前に、最初の CSQX448E メッセージが発行される前にリポジトリー・マネージャーが終了する原因となった問題を解決するために、手操作による介入が必要になる場合があります。

CSQX548E: *csect-name* ローカルの送達不能キューに送信されたメッセージ。チャネル *channel-name* 理由 =*mqrc* (*mqrc-text*)

- システム・プログラマー応答に従ってください。

CSQX788I: *csect-name* 関数 '*func*' を使用したアドレス *address* の DNS ルックアップに *n* 秒かかりました

- システム・プログラマー応答に従ってください。

CSQY222E: *csect-name* キューマネージャーのローカルストレージが極端に不足しています。対処してください。

- キュー・マネージャーの拡張専用領域における仮想ストレージが大幅に不足している。 この状態を緩和して、キュー・マネージャーが異常終了する可能性を回避する必要がある。

CSQY225E: *csect-name* キュー・マネージャーで、2 GB 境界より上のローカル・ストレージが非常に不足しています。アクションを実行してください。

- キュー・マネージャーの 2 GB 境界より上の仮想ストレージ不足が深刻である。 この状態を緩和して、キュー・マネージャーが異常終了する可能性を回避する必要がある。

CSQ5038I: *csect-name* hh.mm.ss.nnnnnn 以降、サービス・タスク service-task が応答しませんでした。 Db2 の問題を確認します。

- システム・プログラマー応答に従ってください。

高可用性の構成

IBM MQ キューマネージャを高可用性（HA）構成で運用する場合、PowerHAforAIX（旧 HACMP）または Microsoft クラスタサービス（MSCS）などの高可用性マネージャ、または IBM MQ マルチインスタンスのキューマネージャで動作するように設定することができます。 Linux システムでは、クォーラム・ベースのグループを使用して高可用性を提供する複製データ・キュー・マネージャー (RDQM) をデプロイすることもできます。 別のオプション (ネイティブ HA) は、コンテナー・デプロイメントを対象としています。

MQ Appliance 高可用性または災害復旧のソリューションの別のオプションは、IBM MQ アプライアンスのペアをデプロイすることです。 IBM MQ Appliance 資料の 高可用性 および 災害復旧 を参照してください。

以下の構成定義に留意する必要があります。

キュー・マネージャー・クラスター
1 つ以上のコンピューター上にある 2 つ以上のキュー・マネージャーのグループ。 自動相互接続を提供し、グループ間でロード・バランシングと冗長度が適切になるようにキューを共有できます。 IBM WebSphere MQ 7.1 からは、クラスター・エラー・リカバリー機能によって、問題が解決されるまで、問題の原因となった操作が再実行されます。

HA クラスター
HA クラスターは、ディスクやネットワークなどの 2 つ以上のコンピューターとリソースのグループであり、一緒に接続され、障害が発生すると、高可用性マネージャー(HACMP(AIX and Linux)や MSCS(Windows)などがフェイルオーバーを実行するように構成されます。 フェイルオーバーは、障害が起きたコンピューターからアプリケーションの状態データをクラスター内の別のコンピューターに転送して、そこで操作が再開されるようにします。 これにより、HA クラスター内で実行されるサービスに対する高可用性が提供されます。 IBM MQ クラスターと HA クラスターとの間の関係は、480 ページの『HA クラスターとキュー・マネージャー・クラスターとの関係』で説明されています。

複数インスタンス・キュー・マネージャー
2 台のコンピュータに同じキューマネージャーが設定されている例。 複数のインスタンスを起動すると、1 つのインスタンスがアクティブインスタンスとなり、もう 1 つのインスタンスがスタンバイインスタンスとなります。 稼働中のインスタンスが故障した場合、別のコンピュータ上で稼働中のスタンバイインスタンスが自動的に引き継ぎます。 複数インスタンス・キュー・マネージャーを使用すると、HACMP や MSCS などのクラスター・テクノロジーを必要とせずに、 IBM MQ に基づいて独自の高可用性メッセージング・システムを構成することができます。 HA クラスターおよび複数インスタンス・キュー・マネージャーは、キュー・マネージャーの可用性を高くするための代替方法です。 複数インスタンス・キュー・マネージャーを HA クラスター内に入れてそれらを結合しないでください。

高可用性複製データ・キュー・マネージャー (HA RDQM)
3 つの Linux サーバーのグループ内の各ノード上で構成されている同じキュー・マネージャーのインスタンス。 3 つのインスタンスのいずれかがアクティブ・インスタンスです。 アクティブ・キュー・マネージャーからのデータは、他の 2 つのインスタンスに同期的に複製されるため、障害が発生した場合はこれらのインスタンスのいずれかが処理を引き継ぐことができます。 サーバーのグループ化は Pacemaker によって制御され、複製は DRBD によって制御されます。

災害復旧複製データ・キュー・マネージャー (DR RDQM)
1 つのサイトの 1 次ノードで 1 つのキュー・マネージャーを実行し、別のサイトのリカバリー・ノードでそのキュー・マネージャーの 2 次インスタンスを実行します。 1 次インスタンスと 2 次インスタンスの間でデータを複製し、何かの理由で 1 次ノードが失われた場合は、2 次インスタンスを 1 次インスタンスにして開始します。 どちらのノードも Linux サーバーでなければなりません。 複製は DRBD によって制御します。

災害復旧/高可用性複製データ・キュー・マネージャー (DR/HA RDQM)
1 つのサイトの高可用性グループで実行される複製データ・キュー・マネージャー (RDQM) を構成できますが、最初のグループを使用不可にする災害が発生した場合は、別のサイトの別の高可用性グループにフェイルオーバーすることができます。 これを DR/HA RDQM と呼びます。

CP4I **ネイティブ HA**
ネイティブ HA は、IBM MQ のコンテナー・デプロイメントを対象とした高可用性ソリューションです。 ネイティブ HA は、ログ複製を使用して、異なるノード上で稼働しているキュー・マネージャーの 3 つのインスタンスを最新の状態に維持します。 一度に 1 つのインスタンスがアクティブになり、メッセージが処理されます。 アクティブ・キュー・マネージャーは、ログ更新を他の 2 つのインスタンスに

送信して、それらを更新します。 アクティブ・インスタンスに障害が発生すると、レプリカ・インスタンスの 1 つが自動的にアクティブ・ロールを引き継ぎます。

複数インスタンス・キュー・マネージャーと HA クラスターとの相違

複数インスタンス・キュー・マネージャーおよび HA クラスターは、キュー・マネージャーの高可用性を達成するための代替方法です。 以下は、2 つのアプローチの間の相違点を明白にするいくつかのポイントです。

複数インスタンス・キュー・マネージャーには、以下の機能が含まれます。

- IBM MQ に統合された基本的なフェイルオーバー・サポート
- HA クラスターより高速のフェイルオーバー
- 簡単な構成および操作
- IBM MQ Explorer との統合

複数インスタンス・キュー・マネージャーには、以下の制約があります。

- 可用性の高い、ハイパフォーマンスのネットワーク・ストレージが必要とされる
- キュー・マネージャーがフェイルオーバーするときに IP アドレスが変更されるので、ネットワーク構成がより複雑になる

HA クラスターには、以下の機能が含まれます。

- アプリケーション・サーバーやデータベースなど、複数のリソースを調整する機能
- 2 つよりも多いノードから成るクラスターを含む、より柔軟な構成オプション
- オペレーターの介入なしで複数回フェイルオーバーできる機能
- フェイルオーバーの一部としてキュー・マネージャーの IP アドレスを引き継ぐ機能

HA クラスターには、以下の制約があります。

- 追加の製品購入およびスキルが必要となる
- クラスターのノード間で切り替え可能なディスクが必要となる
- HA クラスターの構成が比較的に複雑
- 従来はフェイルオーバーが比較的に低速でしたが、最近の HA クラスター製品では改善されています
- キュー・マネージャーなどのリソースのモニターに使用されるスクリプトに不備がある場合、不必要なフェイルオーバーが発生することがあります

HA クラスターとキュー・マネージャー・クラスターとの関係

キュー・マネージャー・クラスターは、キュー・マネージャー・クラスター・キューの使用可能な複数インスタンスに渡るメッセージのロード・バランシングを提供します。 この機能は、単一のキュー・マネージャーよりも高い可用性を提供します。 なぜなら、キュー・マネージャーに障害が生じた後に、メッセージング・アプリケーションがキュー・マネージャー・クラスター・キューの残りの正常なインスタンスに引き続きメッセージを送ったり、アクセスしたりできるためです。 ただし、キュー・マネージャー・クラスターはクラスター内の使用可能なキュー・マネージャーに新規メッセージを自動的にルーティングしますが、使用できないキュー・マネージャーのキューに現時点で入れられているメッセージは、そのキュー・マネージャーを再始動するまで使用できません。 そのため、キュー・マネージャー・クラスター自体は、すべてのメッセージ・データの高可用性は提供せず、 キュー・マネージャーの障害の自動検知、およびキュー・マネージャーの再始動やフェイルオーバーの自動起動の機能もありません。 高可用性 (HA) クラスターには、それらの機能があります。 これら 2 つのタイプのクラスターは、効果的に共有することができます。 キュー・マネージャー・クラスターの概要については、クラスターの設計を参照してください。

関連概念

IBM MQ Advanced container の高可用性

Linux AIX AIX and Linux での HA クラスター

AIX and Linux プラットフォーム上の高可用性 (HA) クラスターで IBM MQ を使用できます。例えば、PowerHA for AIX (旧称 HACMP)、Veritas Cluster Server、HP Serviceguard、 Red Hat Cluster Suite を備えた Red Hat Enterprise Linux クラスターなどです。

このセクションでは、481 ページの『HA クラスターの構成』、HA クラスターとキュー・マネージャー・クラスターとの関係、481 ページの『IBM MQ クライアント』、および 482 ページの『HA クラスター内で作動する IBM MQ』について紹介し、キュー・マネージャーを HA クラスターと共に構成するための手順を示して、改造可能なスクリプトの例を提供します。

このセクションで説明される構成の手順に関する支援として、使用する環境に応じた HA クラスターの文書を参照してください。

HA クラスターの構成

このセクションでノードという語は、オペレーティング・システムおよび HA ソフトウェアを実行しているエンティティーを指して使用されます。このような使用法での同義語としては、「コンピューター」、「システム」、「マシン」、「パーティション」、「ブレード」などがあります。 IBM MQ を使用すると、スタンバイ構成またはテークオーバー構成のいずれか (すべてのクラスター・ノードが IBM MQ ワークロードを実行する相互テークオーバーを含む) をセットアップするのに役立ちます。

スタンバイ構成は最も基本的な HA クラスター構成で、1 つのノードが作業を実行している間、もう 1 つのノードはスタンバイとして機能します。 スタンバイ・ノードは作業を実行しないので、アイドルと呼ばれます。 この構成は、コールド・スタンバイと呼ばれることもあります。 そのような構成では、高度のハードウェア冗長度が必要となります。 ハードウェアに関して節約するために、この構成を拡張して、単一のスタンバイ・ノードに対して複数の作業ノードを組み合わせることも可能です。 この方法のポイントは、スタンバイ・ノードが他のいずれの作業ノードによる作業も引き継ぐことができることです。 この構成もスタンバイ構成と呼ばれ、「N+1」構成と呼ばれることもあります。

テークオーバー構成は、すべてのノードが何らかの作業を実行し、ノード障害の際には重要な作業を引き継ぐことが可能な、より高度な構成です。

片側テークオーバー構成は、スタンバイ・ノードが重要ではない、そして移動できない何らかの作業を追加で実行する構成です。 この構成はスタンバイ構成と似ていますが、(重要ではない) 作業がスタンバイ・ノードによって実行される点が異なります。

相互テークオーバー構成は、すべてのノードが可用性の高い (移動可能な) 作業を実行する構成です。 このタイプの HA クラスター構成は、すべてのノードが重要なワークロードをアクティブに処理していることを示すために、「アクティブ/アクティブ」と呼ばれることもあります。

拡張されたスタンバイ構成またはいずれかのテークオーバー構成では、他のノードの作業をテークオーバーできるノードに負わせる可能性のあるピーク・ロード量を検討することは重要です。 そのようなノードは、受容可能なレベルのパフォーマンスを保つために十分な能力を持つ必要があります。

HA クラスターとキュー・マネージャー・クラスターとの関係

キュー・マネージャー・クラスターは管理作業を減らし、キュー・マネージャー・クラスター・キューの複数インスタンスに渡るメッセージのロード・バランシングを提供します。 それらはまた、単一のキュー・マネージャーよりも高い可用性を提供します。 キュー・マネージャーに障害が生じた後に、メッセージング・アプリケーションがキュー・マネージャー・クラスター・キューの機能しているインスタンスに引き続きアクセスできるためです。 ただし、キュー・マネージャー・クラスター自体には、キュー・マネージャーの障害の自動検知、およびキュー・マネージャーの再始動やフェイルオーバーの自動起動の機能はありません。 HA クラスターには、それらの機能があります。 これら 2 つのタイプのクラスターは、効果的に共有することができます。

IBM MQ クライアント

再始動またはテークオーバーに依存する可能性のあるキュー・マネージャーと通信している IBM MQ クライアントは、接続の失敗に対応できるように、そして再接続を繰り返し試行するように記述する必要があります。 IBM MQ には、接続の可用性とワークロード・バランシングを支援するクライアント・チャネル

定義テーブル (CCDT) の処理の機能が含まれています。ただし、これらの機能は、フェイルオーバー・システムで作業する場合には直接関係ありません。

トランザクション機能により、IBM MQ MQI client は、同じキュー・マネージャーに接続されている限り、2 フェーズ・トランザクションに参加できます。 トランザクション機能で、IP ロード・バランサーなどの手法を使用してキュー・マネージャーのリストから選択することはできません。 HA 製品を使用する場合、キュー・マネージャーはそれが実行されているノードの ID (名前およびアドレス) を保持するので、HA 制御の下にあるキュー・マネージャーと共にトランザクション機能を使用できます。

HA クラスター内で作動する IBM MQ

すべての HA クラスターには、フェイルオーバーの単位の概念があります。 これは、可用性の高いサービスを形成するすべてのリソースを含む定義のセットです。 フェイルオーバーの単位には、サービス自体と、それが依存する他のすべてのリソースが含まれます。

HA ソリューションは、フェイルオーバーの単位に対して異なる用語を使用します。

- PowerHA for AIX では、フェイルオーバーの単位はリソース・グループ と呼ばれます。
- Veritas Cluster Server では、サービス・グループと呼ばれます。
- Serviceguard では、パッケージと呼ばれます。

このトピックでは、フェイルオーバーの単位を意味する用語として、リソース・グループ を使用します。

IBM MQ のフェイルオーバーの単位として最も小さいものは、キュー・マネージャーです。 通常、キュー・マネージャーを含むリソース・グループには、リソース・グループによる使用のために専用に予約されている、ボリューム・グループまたはディスク・グループ内の共有ディスクも含まれます。また、キュー・マネージャーへの接続に使用される IP アドレスが含まれます。 さらに、同じリソース・グループ内のリスナーまたはトリガー・モニターなど、別のリソースとして、またはキュー・マネージャー自体の制御下で、他の IBM MQ リソースを含めることも可能です。

HA クラスターで使用されるキュー・マネージャーでは、データおよびログがクラスター内のノード間で共有されるディスク上に存在する必要があります。 HA クラスターは、一度にクラスター内の 1 つのノードだけがディスクに書き込むことができるようにします。 HA クラスターはモニター・スクリプトを使用して、キュー・マネージャーの状態をモニターできます。

キュー・マネージャーに関連したデータおよびログの両方のために、単一の共有ディスクを使用することが可能です。 ただし通常は、別々の共有ファイル・システムを使用して、それらを個別にサイズ変更および調整できるようにします。

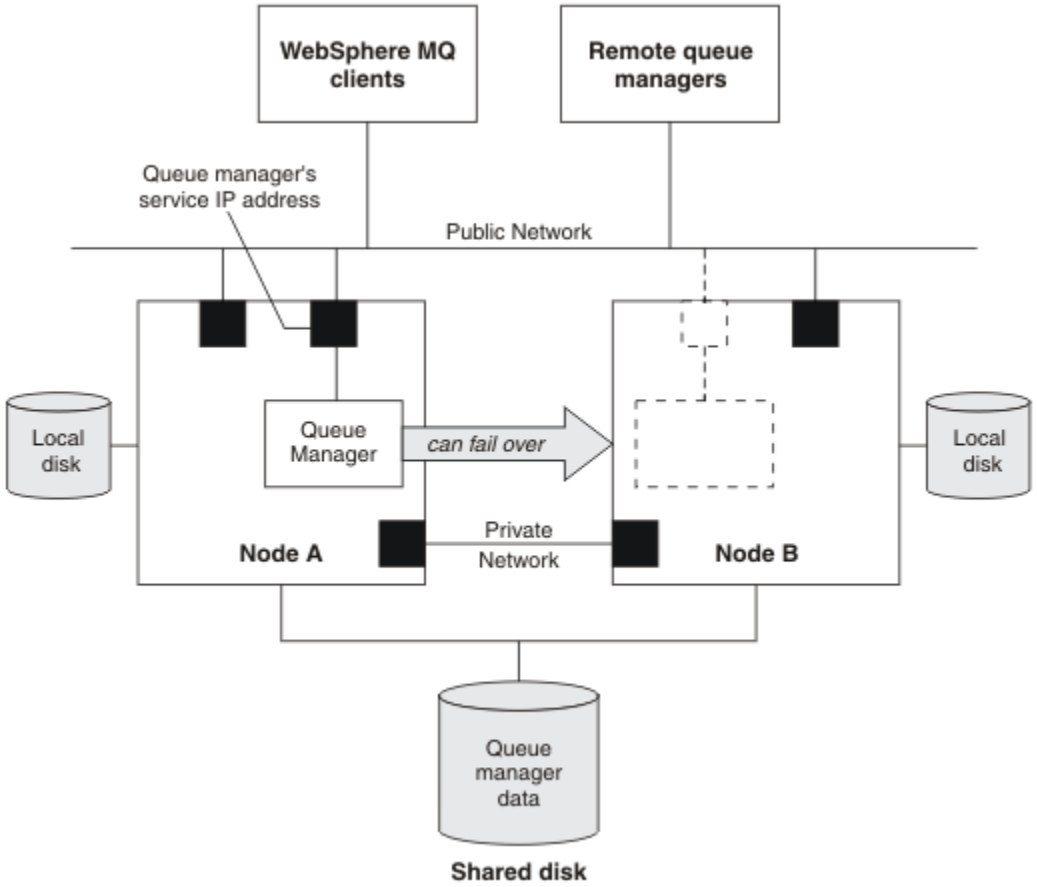


図 *69. HA* クラスター

図 1 は、2 つのノードのある HA クラスターを示しています。 HA クラスターは、リソース・グループ内で定義されたキュー・マネージャーの可用性を管理しています。 1 つのノード (ノード A) だけが現在キュー・マネージャーを実行しているので、これはアクティブ/パッシブまたはコールド・スタンバイ構成です。 キュー・マネージャーは、共有ディスク上のデータおよびログ・ファイルと共に作成されました。 キュー・マネージャーには、同じく HA クラスターによって管理されるサービス IP アドレスがあります。 キュー・マネージャーは、共有ディスクおよびそのサービス IP アドレスに依存します。 HA クラスターがノード A からノード B にキュー・マネージャーをフェイルオーバーするときは、最初にキュー・マネージャーの従属リソースをノード B に移動してから、キュー・マネージャーを始動します。

HA クラスターに複数のキュー・マネージャーが含まれる場合は、その HA クラスター構成のために、 フェイルオーバーの後に同じノード上に複数のキュー・マネージャーが実行する結果になることがあります。 HA クラスター内の各キュー・マネージャーには、独自のポート番号が割り当てられます。 これは、いずれかの特定の時点でアクティブになっているクラスター・ノードで使用されます。

通常、HA クラスターは root ユーザーとして実行されます。 IBM MQ は、mqm ユーザーとして実行されます。 IBM MQ の管理は、mqm グループのメンバーに認可されます。 mqm ユーザーおよびグループの両方が、すべての HA クラスター・ノード上に存在することを確認してください。 ユーザー ID およびグループ ID は、クラスター全体で一貫性のあるものでなければなりません。 root ユーザーが IBM MQ を管理することは許可されていません。スクリプトを開始、停止、またはモニターするスクリプトは、mqm ユーザーに切り替える必要があります。

注 : IBM MQ は、すべてのノードに正しくインストールする必要があります。製品の実行可能ファイルを共有することはできません。

Linux AIX 共有ディスクの構成 *(AIX and Linux)*

HA クラスター内の IBM MQ キュー・マネージャーでは、データ・ファイルおよびログ・ファイルが、共有ディスク上で指定された共通のリモート・ファイル・システム内にあることが必要です。

このタスクについて

図 1 は、HA クラスター内のキュー・マネージャー用に可能なレイアウトを示しています。 キュー・マネージャーのデータおよびログ・ディレクトリーは、どちらも /MQHA/QM1 にマウントされる共有ディスク上にあります。 フェイルオーバーが生じるとき、このディスクは HA クラスターのノード間で切り替えられるので、キュー・マネージャーをどこで再始動してもデータが使用可能になります。 `mqs.ini` ファイルには、QM1 キュー・マネージャーのスタンザがあります。 `qm.ini` ファイル内のログ・スタンザには、LogPath の値が含まれています。

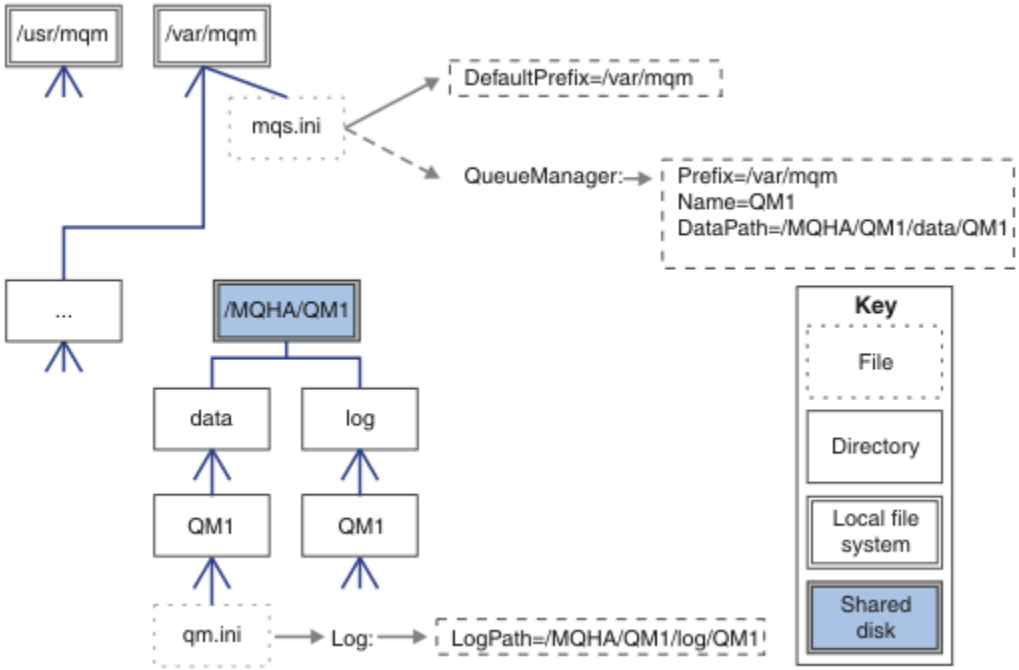


図 *70.* 共有名 *data* と *log* ディレクトリー

手順

1. キュー・マネージャーのファイル・システム用のマウント・ポイントの名前を決めます。
   例えば、キュー・マネージャーのデータ・ファイルの場合は`/MQHA/qmgrname/data`、ログ・ファイルの場合は`/MQHA/qmgrname/log` を指定します。
2. キュー・マネージャーのデータ・ファイルおよびログ・ファイルを入れるためのボリューム・グループ (またはディスク・グループ) を作成します。
   このボリューム・グループは、キュー・マネージャーと同じリソース・グループ内の高可用性 (HA) クラスターによって管理されます。
3. キュー・マネージャーのデータ・ファイルおよびログ・ファイル用のファイル・システムをボリューム・グループ内に作成します。
4. 次にノードごとに、ファイル・システム用のマウント・ポイントを作成して、ファイル・システムがマウント可能であることを確認します。
   mqm ユーザーはマウント・ポイントを所有する必要があります。

Linux AIX *HA クラスター・キュー・マネージャーの作成 (AIX and Linux)*

キュー・マネージャーを高可用性クラスターで使用するための最初のステップは、ノードの 1 つにキュー・マネージャーを作成することです。

このタスクについて

HA クラスターで使用するためのキュー・マネージャーを作成するには、まずキュー・マネージャーを作成するクラスター内のノードを 1 つ選択し、次にそのノードで以下の手順を実行する必要があります。

手順

1. キュー・マネージャーのファイル・システムをノードにマウントします。
2. **crtmqm** コマンドを使用してキュー・マネージャーを作成します。
   以下に例を示します。

   ```
   crtmqm -md /MQHA/qmgrname/data -ld /MQHA/qmgrname/log qmgrname
   ```
3. **strmqm** コマンドを使用して、手動でキュー・マネージャーを開始します。
4. キューとチャネルの作成、およびキュー・マネージャーの始動時にリスナーが自動的に開始するようにキュー・マネージャーを設定するなど、 キュー・マネージャーの初期構成を完了します。
5. **endmqm** コマンドを使用してキュー・マネージャーを停止させます。
6. 以下のように **dspmqinf** コマンドを使用して、**addmqinf** コマンドを表示します。

   ```
   dspmqinf -o command qmgrname
   ```

   ここで、qmgrname は、キュー・マネージャーの名前です。

   addmqinf コマンドの使用について詳しくは、 485 ページの『他の HA クラスター・ノードへのキュー・マネージャー構成の追加 (AIX and Linux)』を参照してください。
   addmqinf コマンドは、以下の例のように表示されます。

   ```
   addmqinf -sQueueManager -vName=qmgrname -vDirectory=qmgrname \
   -vPrefix=/var/mqm -vDataPath=/MQHA/qmgrname/data/qmgrname
   ```
7. 表示されるコマンドを注意深く記録してください。
8. キュー・マネージャーのファイル・システムをアンマウントします。

次のタスク

これで、485 ページの『他の HA クラスター・ノードへのキュー・マネージャー構成の追加 (AIX and Linux)』で説明されているステップを完了する準備ができました。

Linux AIX 他の *HA* クラスター・ノードへのキュー・マネージャー構成の追加 *(AIX and Linux)*

キュー・マネージャーの構成情報を HA クラスター内の他のノードに追加する必要があります。

始める前に

この作業を完了する前に、484 ページの『HA クラスター・キュー・マネージャーの作成 (AIX and Linux)』のステップを完了している必要があります。 キュー・マネージャーを作成したら、他の各ノードで以下の手順を実行して、HA クラスター内の他のノードのそれぞれにキュー・マネージャーの構成情報を追加する必要があります。

このタスクについて

HA クラスターで使用するためのキュー・マネージャーを作成する場合は、まず、キュー・マネージャーを作成するクラスター内のノードを 1 つ選択する必要があります (詳しくは、484 ページの『HA クラスター・キュー・マネージャーの作成 (AIX and Linux)』を参照)。

手順

1. キュー・マネージャーのファイル・システムをマウントする。
2. ノードにキュー・マネージャー構成情報を追加する。
   構成情報を追加するには、以下の 2 つの方法があります。

   - `/var/mqm/mqs.ini` を直接的に編集する。

- 484 ページの『HA クラスター・キュー・マネージャーの作成 (AIX and Linux)』の手順 6 の **dspmqinf** コマンドで表示された **addmqinf** コマンドを実行する。

3. キュー・マネージャーを開始してから停止して、構成を検査する。

   キュー・マネージャーの開始と停止に使用するコマンドは、**addmqinf** コマンドと同じ IBM MQ インストール済み環境から発行する必要があります。 キュー・マネージャーに現在関連付けられているものとは別のインストール済み環境からキュー・マネージャーを開始および停止するには、まず、**setmqm** コマンドを使用して、インストール済み環境にキュー・マネージャーとの関連付けを設定する必要があります。 詳細については、setmqm を参照してください。
4. キュー・マネージャーのファイル・システムをアンマウントする。

Linux AIX *HA クラスター・キュー・マネージャーを開始するためのシェル・スクリプトの例 (AIX and Linux)*

キュー・マネージャーは、HA クラスター内でリソースとして表わされます。 HA クラスターは、キュー・マネージャーを始動および停止できることが必要です。 ほとんどの場合、キュー・マネージャーを始動するためにシェル・スクリプトを使用できます。 これらのスクリプトは、ネットワーク・ファイル・システムを使用するかまたは各ローカル・ディスクにコピーすることにより、クラスター内のすべてのノードの同じ場所で使用可能にする必要があります。

注 : 失敗したキュー・マネージャーを再始動する前に、アプリケーションをキュー・マネージャーのそのインスタンスから切断する必要があります。 そうしない場合、キュー・マネージャーは正しく再始動されない場合があります。

適切なシェル・スクリプトの例をここに示します。 これらを必要に合うように作り替えて、HA クラスターの制御下でキュー・マネージャーを始動するために使用できます。

以下のシェル・スクリプトは、HA クラスター・ユーザーから mqm ユーザーに切り替えて、キュー・マネージャーを正常に始動できるようにする方法を示す例です。

```
#!/bin/ksh

# A simple wrapper script to switch to the mqm user.

su mqm -c name_of_your_script $*
```

以下のシェル・スクリプトは、キュー・マネージャーの現在の状態について何も想定せずに、キュー・マネージャーを始動する方法を示す例です。 キュー・マネージャーに属するすべてのプロセスを突然に終了させる方法が使用されていることに注意してください。

```
#!/bin/ksh
#
# This script robustly starts the queue manager.
#
# The script must be run by the mqm user.

# The only argument is the queue manager name. Save it as QM variable
QM=$1

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

# End any queue manager processes which might be running.

srchstr="( |-m)$QM *.*$"
for process in amqzmuc0 amqzxma0 amqfcxba amqfqpub amqpcsea amqzlaa0 \
               amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqrmppa \
               amqzfuma amqzmuf0 amqzmur0 amqzmgr0
 do
  ps -ef | tr "\t" " " | grep $process | grep -v grep | \
    egrep "$srchstr" | awk '{print $2}'| \
       xargs kill -9 > /dev/null 2>&1
done

# It is now safe to start the queue manager.
```

```
# The strmqm command does not use the -x flag.
strmqm ${QM}
```

このスクリプトを変更して、関連する他のプログラムを開始することができます。

Linux AIX *HA クラスター・キュー・マネージャーを停止するためのシェル・スクリプトの例 (AIX and Linux)*

ほとんどの場合、キュー・マネージャーを停止するためにシェル・スクリプトを使用できます。 適切なシェル・スクリプトの例をここに示します。 これらを必要に合うように作り替えて、HA クラスターの制御下でキュー・マネージャーを停止するために使用できます。

以下のスクリプトは、キュー・マネージャーの現在の状態について何も想定せずに、キュー・マネージャーを即時に停止する方法を示す例です。 スクリプトは mqm ユーザーが実行する必要があります。 そのため、このスクリプトをシェル・スクリプト内にラップして、ユーザーを HA クラスター・ユーザーから mqm に切り替えることが必要な場合があります (シェル・スクリプトの例は 486 ページの『HA クラスター・キュー・マネージャーを開始するためのシェル・スクリプトの例 (AIX and Linux)』 に記載されています)。

```
#!/bin/ksh
#
# The script ends the QM by using two phases, initially trying an immediate
# end with a time-out and escalating to a forced stop of remaining
# processes.
#
# The script must be run by the mqm user.
#
# There are two arguments: the queue manager name and a timeout value.
QM=$1
TIMEOUT=$2

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

if [ -z "$TIMEOUT" ]
then
  echo "ERROR! No timeout specified"
  exit 1
fi

for severity in immediate brutal
do
  # End the queue manager in the background to avoid
  # it blocking indefinitely. Run the TIMEOUT timer
  # at the same time to interrupt the attempt, and try a
  # more forceful version. If the brutal version fails,
  # nothing more can be done here.

  echo "Attempting ${severity} end of queue manager '${QM}'"
  case $severity in

  immediate)
    # Minimum severity of endmqm is immediate which severs connections.
    # HA cluster should not be delayed by clients
    endmqm -i ${QM} &
    ;;

  brutal)
    # This is a forced means of stopping queue manager processes.

    srchstr="( |-m)$QM *.*$"
    for process in amqzmuc0 amqzxma0 amqfcxba amqfqpub amqpcsea amqzlaa0 \
                amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqrmppa \
                amqzfuma amqzmuf0 amqzmur0 amqzmgr0
    do
      ps -ef | tr "\t" " " | grep $process | grep -v grep | \
         egrep "$srchstr" | awk '{print $2}'| \
            xargs kill -9 > /dev/null 2>&1
    done

  esac

  TIMED_OUT=yes
  SECONDS=0
  while (( $SECONDS < ${TIMEOUT} ))
```

```
  do
   TIMED_OUT=yes
   i=0
   while [ $i -lt 5 ]
   do
     # Check for execution controller termination
     srchstr="( |-m)$QM *.*$"
     cnt=`ps -ef | tr "\t" " " | grep amqzxma0 | grep -v grep | \
       egrep "$srchstr" | awk '{print $2}' | wc -l `
     i=`expr $i + 1`
     sleep 1
     if [ $cnt -eq 0 ]
     then
       TIMED_OUT=no
       break
     fi
   done

   if [ ${TIMED_OUT} = "no" ]
   then
     break
   fi

   echo "Waiting for ${severity} end of queue manager '${QM}'"
   sleep 1
  done # timeout loop

  if [ ${TIMED_OUT} = "yes" ]
  then
    continue        # to next level of urgency
  else
    break           # queue manager is ended, job is done
  fi

done # next phase
```

注 : 特定のキュー・マネージャーに対して実行されているプロセスによっては、このスクリプトに含まれるキュー・マネージャー・プロセスのリストが完全ではなかったり、対象キュー・マネージャーで実行されているプロセス以外のプロセスが含まれたりする可能性があります。

```
for process in amqzmuc0 amqzxma0 amqfcxba amqfqpub amqpcsea amqzlaa0 \
               amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqrmppa \
               amqzfuma amqzmuf0 amqzmur0 amqzmgr0
```

構成されている機能と、特定のキュー・マネージャーで実行されているプロセスに基づいて、プロセスをこのリストに包含したり、このリストから除外したりできます。 プロセスの完全な一覧と、特定の順序でプロセスを停止する方法については、UNIX および Linux におけるキュー・マネージャーの手動停止を参照してください。

Linux AIX *HA クラスター・キュー・マネージャーのモニター (AIX and Linux)*

通常、高可用性 (HA) クラスターがキュー・マネージャーの状態を定期的にモニターするための手段を提供します。 ほとんどの場合、このためにシェル・スクリプトを使用できます。 適切なシェル・スクリプトの例をここに示します。 これらのスクリプトを必要に合うように作り替えて、環境に特定の追加のモニター検査を行うために使用できます。

IBM MQ の複数のインストール済み環境を 1 つのシステムに共存させることができます。 複数のインストール環境について詳しくは、複数のインストールを参照してください。 複数のインストールでモニター・スクリプトを使用する場合は、いくつかの追加ステップを実行する必要があります。 プライマリー・インストールがある場合は、スクリプトを使用するために *MQ_INSTALLATION_PATH* を指定する必要はありません。 それ以外の場合は、以下の手順を使用して、 *MQ_INSTALLATION_PATH* が正しく識別されていることを確認します。

1. IBM MQ インストール済み環境から **crtmqenv** コマンドを使用して、キュー・マネージャーの正しい *MQ_INSTALLATION_PATH* を識別します。

   ```
   crtmqenv -m qmname
   ```

   このコマンドは、*qmname* で指定されたキュー・マネージャーの正しい *MQ_INSTALLATION_PATH* 値を返します。

2. 該当する *qmname* パラメーターと *MQ_INSTALLATION_PATH* パラメーターを使用して、モニター・スクリプトを実行します。

注 : PowerHA for AIX では、モニター・プログラムにキュー・マネージャー用のパラメーターを指定する手段が提供されていません。 キュー・マネージャーごとに、キュー・マネージャー名をカプセル化する別個のモニター・プログラムを作成する必要があります。 キュー・マネージャー名をカプセル化するための、AIX 上で使用するスクリプトを以下に示します。

```
#!/bin/ksh
su mqm -c name_of_monitoring_script qmname  MQ_INSTALLATION_PATH
```

ここで、 *MQ_INSTALLATION_PATH* は、キュー・マネージャー *qmname* が関連付けられている IBM MQ のインストール済み環境へのパスを指定するオプション・パラメーターです。

下記のスクリプトは、**runmqsc** がハングする可能性に対して耐久力がありません。 通常、HA クラスターはハングしたモニター・スクリプトを障害として処理するので、それ自体はこの可能性に対して耐久力があります。

しかしこのスクリプトは、キュー・マネージャーが開始状態にあることに関しては容認します。 それは、HA クラスターがキュー・マネージャーを開始した直後に、そのモニターを開始することが一般的であるためです。 一部の HA クラスターは、リソースの開始フェーズと実行フェーズとを区別しますが、開始フェーズの所要時間を構成することが必要です。 キュー・マネージャーの開始に必要な時間はそれが行う作業の量に依存するので、キュー・マネージャーの開始に必要な最大時間を選択することは困難です。 小さ過ぎる値を選択すると、キュー・マネージャーが開始処理をまだ完了していないだけの場合でも、障害が発生したと、HA クラスターが間違って判断してしまいます。 これにより、フェイルオーバーが無限に続く結果になる可能性があります。

スクリプトは、mqm ユーザーによって実行される必要があります。そのため、このスクリプトをシェル・スクリプト内にラップして、ユーザーを HA クラスター・ユーザーから mqm に切り替えることが必要な場合があります (シェル・スクリプトの例は、486 ページの『HA クラスター・キュー・マネージャーを開始するためのシェル・スクリプトの例 (AIX and Linux)』にあります)。

```
#!/bin/ksh
#
# This script tests the operation of the queue manager.
#
# An exit code is generated by the runmqsc command:
# 0  => Either the queue manager is starting or the queue manager is running and responds.
#       Either is OK.
# >0 => The queue manager is not responding and not starting.
#
# This script must be run by the mqm user.
QM=$1
MQ_INSTALLATION_PATH=$2

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

if [ -z "$MQ_INSTALLATION_PATH" ]
then
  # No path specified, assume system primary install or MQ level < 7.1.0
  echo "INFO: Using shell default value for MQ_INSTALLATION_PATH"
else
  echo "INFO: Prefixing shell PATH variable with $MQ_INSTALLATION_PATH/bin"
  PATH=$MQ_INSTALLATION_PATH/bin:$PATH
fi

# Test the operation of the queue manager. Result is 0 on success, non-zero on error.
echo "ping qmgr" | runmqsc ${QM} > /dev/null 2>&1
pingresult=$?

if [ $pingresult -eq 0 ]
then # ping succeeded

  echo "Queue manager '${QM}' is responsive"
  result=0

else # ping failed
```

```
  # Don't condemn the queue manager immediately, it might be starting.
  srchstr="( |-m)$QM *.*$"
  cnt=`ps -ef | tr "\t" " " | grep strmqm | grep "$srchstr" | grep -v grep \
                | awk '{print $2}' | wc -l`
  if [ $cnt -gt 0 ]
  then
    # It appears that the queue manager is still starting up, tolerate
    echo "Queue manager '${QM}' is starting"
    result=0
  else
    # There is no sign of the queue manager starting
    echo "Queue manager '${QM}' is not responsive"
    result=$pingresult
  fi

fi

exit $result
```

Linux AIX キュー・マネージャーを *HA* クラスターの制御下に置く方法 *(AIX and Linux)*

キュー・マネージャーを、HA クラスターの制御下で、キュー・マネージャーの IP アドレスおよび共有ディスクと共に構成する必要があります。

このタスクについて

キュー・マネージャーを HA クラスターの制御下に置くには、キュー・マネージャーおよび関連付けられたすべてのリソースを含めるためのリソース・グループを定義する必要があります。

手順

1. キュー・マネージャー、キュー・マネージャーのボリュームまたはディスク・グループ、およびキュー・マネージャーの IP アドレスを含むリソース・グループを作成します。

   IP アドレスは仮想 IP アドレスであり、コンピューターの IP アドレスではありません。
2. HA クラスターがクラスター・ノード間でリソースを適正に切り替えて、キュー・マネージャーを制御する準備ができたことを確認してください。

Linux AIX *HA* クラスター・キュー・マネージャーの削除 *(AIX and Linux)*

キュー・マネージャーの実行が必要ではなくなったノードからキュー・マネージャーを除去できます。

このタスクについて

HA クラスター内のノードからキュー・マネージャーを除去するには、その構成情報を除去する必要があります。

手順

1. HA クラスターからノードを除去して、HA クラスターがこのノードでのキュー・マネージャーのアクティブ化を試行しなくなるようにします。
2. 以下の **rmvmqinf** コマンドを使用して、キュー・マネージャーの構成情報を除去します。

   rmvmqinf *qmgrname*
3. オプション: キュー・マネージャーを完全に削除するには、**dltmqm** コマンドを使用します。

   重要 : dltmqm コマンドを使用してキュー・マネージャーを削除すると、そのキュー・マネージャーのデータおよびログ・ファイルが完全に削除されることに注意してください。

   キュー・マネージャーを削除した後に、**rmvmqinf** コマンドを使用して他のノードから残りの構成情報を除去することができます。

Windows Microsoft Cluster Service (MSCS) のサポート

仮想サーバーのフェイルオーバーをサポートする MSCS の紹介およびセットアップ。 MSCS は Windows Server Failover Clustering (WSFC) とも呼ばれます。

この情報は IBM MQ for Windows のみに適用されます。

注 : Windows Server 2016 以降、 Microsoft Cluster Service (MSCS) の新しい名前は Windows Server Failover Clustering (WSFC) になりました。

MSCS/WSFC を使用すると、サーバーをクラスターに接続して、データとアプリケーションの可用性を高め、システムの管理を容易にすることができます。 MSCS/WSFC は、サーバーまたはアプリケーションの障害を自動的に検出し、リカバリーすることができます。

MSCS/WSFC は、アプリケーション、Web サイト、印刷キュー、またはファイル共用 (例えば、ディスク・スピンドル、ファイル、および IP アドレスを含む) に対応する仮想サーバーのフェイルオーバーをサポートします。

フェイルオーバー とは、MSCS/WSFC がクラスター内の 1 つのコンピューター上のアプリケーションで障害を検出し、中断されたアプリケーションを正常にシャットダウンし、その状態データを他のコンピューターに転送し、そこでアプリケーションを再開始するプロセスです。

フェイルオーバー・クラスターを構成して使用する方法については、サブトピックを参照してください。

Windows *MSCS* クラスターの紹介

Microsoft Cluster Service (MSCS) クラスターは複数のコンピューターのグループで、相互に接続され、1 つに障害が発生した場合に MSCS が フェイルオーバーを実行し、障害が発生したコンピューターからクラスター内の別のコンピューターにアプリケーションの状態データを転送して、そこで操作を再開するように構成されます。

注 : Windows Server 2016 以降、 Microsoft Cluster Service (MSCS) の新しい名前は Windows Server Failover Clustering (WSFC) になりました。

479 ページの『高可用性の構成』には、MSCS クラスター、複数インスタンス・キュー・マネージャー、および IBM MQ クラスターの間の比較が含まれています。

このセクションおよび従属するトピックで、用語クラスター が単独で使用されるときには、**常に** MSCS クラスターを意味しています。 これは、このガイドの他の場所で説明されている IBM MQ クラスターとは異なります。

2 台のマシンのクラスターは、仮想 *IP* アドレス を使用して、クライアント・アクセス用のネットワークに一緒に接続される 2 台のコンピューター (例えば、A と B) で構成されます。 さらに、1 つ以上の専用ネットワークによって相互に接続されていることもあります。 A および B は、使用するそれぞれのコンピューター上で、サーバー・アプリケーション用に少なくとも 1 つのディスクを共有します。 また、MSCS が排他的に使用するための別の共有ディスクがあります。これは独立ディスク (*RAID*) レベル 1 の冗長配列でなければなりません。このディスクを *quorum* ディスクと言います。 MSCS は、両方のコンピューターをモニターして、ハードウェアとソフトウェアが正しく実行していることを確認します。

このような単純なセットアップでは、どちらのコンピューターにもすべてのアプリケーションがインストールされますが、ライブ・アプリケーションはコンピューター A でのみ実行され、コンピューター B は稼働しても待ち状態になります。 コンピューター A に何らかの問題が発生すると、障害が起きたアプリケーションは MSCS によって通常の方法でシャットダウンされ、他のコンピューターにその状態データが転送され、アプリケーションはそこで再開されます。 これがフェイルオーバー です。 アプリケーションをクラスター指向 にしておくと、MSCS との対話が十分に行われ、フェイルオーバーの機能が高まります。

2 台のコンピューターのクラスター用の一般的なセットアップを 492 ページの図 71 に示します。

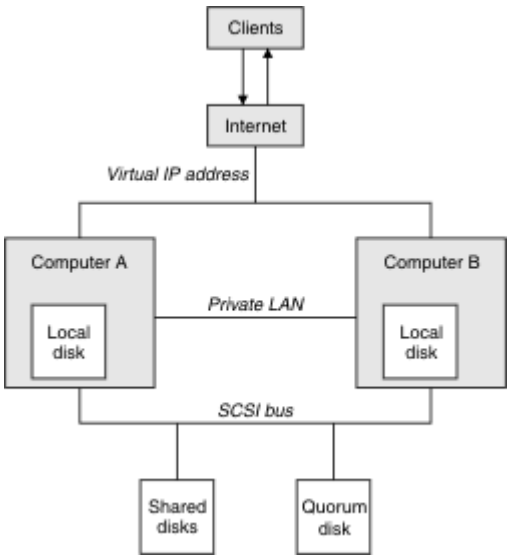


図 *71. 2* 台のコンピューターにおける *MSCS* クラスター

それぞれのコンピューターが共有ディスクにアクセスできますが、MSCS 制御の下では、共有ディスクにアクセスできるのは一度に 1 台のコンピューターだけです。 フェイルオーバーの際、MSCS はアクセスを他のコンピューターに切り替えます。 共有ディスク自体は通常 RAID ですが、それ以外でも構いません。

各コンピューターは、クライアント・アクセス用に外部ネットワークに接続され、それぞれが IP アドレスを持っています。 ただし、このクラスターと通信する外部クライアントは、1 つの仮想 *IP* アドレス だけを認識し、MSCS がクラスター内で IP トラフィックの経路を適切に定めます。

また、MSCS は、1 つ以上の専用接続、または公衆ネットワークを介して 2 台のコンピューター間で独自の通信を実行し、例えば、ハートビートを使用して 2 台のコンピューターの状態をモニターしたり、データベースを同期したりします。

Windows *MSCS クラスター化用の IBM MQ のセットアップ*

キュー・マネージャーを MSCS へのフェイルオーバーの単位にすることにより、クラスター化用に IBM MQ を構成します。 キュー・マネージャーを MSCS へのリソースとして定義すると、キュー・マネージャーをモニターし、問題がある場合にはクラスター内の別のコンピューターにキュー・マネージャーを転送できます。

注 : Windows Server 2016 以降、 Microsoft Cluster Service (MSCS) の新しい名前は Windows Server Failover Clustering (WSFC) になりました。

システムに合わせて設定するには、クラスター内の各コンピューター上に IBM MQ をインストールすることから始めます。

キュー・マネージャーは IBM MQ インストール名と関連付けられるため、クラスター内のすべてのコンピューター上で IBM MQ インストール名が同じでなければなりません。 インストールおよびアンインストールを参照してください。

キュー・マネージャー自体は、それらが作成されるコンピューター上にのみ存在している必要があります。フェイルオーバーの際、MSCS は他のコンピューターでキュー・マネージャーを始動します。 ただし、キュー・マネージャーは、ローカル・ドライブ上ではなくクラスター共有ディスク上にログおよびデータ・ファイルを持つ必要があります。 キュー・マネージャーが既にローカル・ドライブにインストールされている場合、IBM MQ で提供されているツールを使用してこれを移行することができます。495 ページの『キュー・マネージャーを MSCS ストレージに移動する』を参照してください。 MSCS で使用する新しいキュー・マネージャーを作成するには、494 ページの『MSCS で使用するようにキュー・マネージャーを作成する』を参照してください。

インストールおよび移行後、MSCS クラスター・アドミニストレーターを使用して MSCS にキュー・マネージャーを認識させます。497 ページの『MSCS 制御下にキュー・マネージャーを置く』を参照してください。

MSCS 制御からキュー・マネージャーを除去する場合、499 ページの『MSCS 制御からキュー・マネージャーを除去する』で説明されている手順に従ってください。

Windows セットアップの対称性と *MSCS*

アプリケーションがあるノードから別のノードに切り替わるとき、ノードに関係なく同じ方法で動作する必要があります。 これを確実にする最善の方法は、環境を同一にすることです。

可能であれば、各コンピューター上で同一のハードウェア、オペレーティング・システム・ソフトウェア、製品ソフトウェア、および構成でクラスターをセットアップします。 特に、2 台のコンピューターにインストールされた必須ソフトウェアがすべて、バージョン、保守レベル、SupportPacs、パスおよび出口の点で同一であるように、さらに493 ページの『MSCS のセキュリティー』で説明されているとおり共通のネーム・スペース (セキュリティー環境) が存在するようにしてください。

Windows *MSCS* のセキュリティー

MSCS セキュリティーを確実にするために、次のガイドラインに従ってください。

ガイドラインは次のとおりです。

- クラスター内の各コンピューターに同一のソフトウェアがインストールされていることを確認してください。
- クラスター間で共通のネーム・スペース (セキュリティー環境) を作成します。
- MSCS クラスターのノードをドメインのメンバーにし、そのドメイン内ではクラスター所有者 のユーザー・アカウントをそのドメイン・アカウントにします。
- クラスターのその他のユーザー・アカウントもドメイン・アカウントにして、それらのユーザー・アカウントを両方のノードで使用できるようにします。 既にドメインがあり、IBM MQ に関係するアカウントがドメイン・アカウントになっている場合は、これが自動的に当てはまります。 現在ドメインがない場合、クラスター・ノードおよび関連するアカウントを提供するために、小さなドメイン をセットアップすることを検討してください。 その目的は、2 台のコンピューターのクラスターを単一のコンピューター・リソースのようにすることです。

  あるコンピューターに対してローカルのアカウントが、もう 1 台のコンピューター上には存在しないということを覚えておいてください。 もう 1 台のコンピューター上に同じ名前でアカウントを作成しても、そのセキュリティー ID (SID) が異なるため、アプリケーションが別のノードに移動したとき、そのノードでの許可はありません。

フェイルオーバーまたは移動時に、IBM MQ MSCS サポートにより、キュー・マネージャー・オブジェクトを収めたどのファイルも必ず宛先ノードに対して同等の許可を持つようになります。 明示的に、コードは、管理者グループと mqm グループ、および SYSTEM アカウントが完全な制御権を持っていること、および Everyone が古いノードに対する読み取りアクセス権を持っている場合は、その許可が宛先ノードに追加されていることを検査します。

ドメイン・アカウントを使用して IBM MQ サービスを実行できます。 クラスター内の各コンピューター上のローカル mqm グループにそのサービスがあることを確認してください。

Windows 複数のキュー・マネージャーを *MSCS* と共に使用する

1 台のコンピューター上で複数のキュー・マネージャーを実行する場合、以下のセットアップのいずれかを選択できます。

セットアップは次のとおりです。

- すべてのキュー・マネージャーが 1 つのグループで。 この構成では、いずれかのキュー・マネージャーで問題が発生した場合、グループ内のすべてのキュー・マネージャーがグループとして他のコンピューターにフェイルオーバーします。
- 各グループのキュー・マネージャーが単独で。 この構成では、あるキュー・マネージャーに問題が発生した場合、他のキュー・マネージャーに影響を与えずに、そのキュー・マネージャーだけが他のコンピューターにフェイルオーバーします。
- 上記 2 つのセットアップの混合。

Windows クラスター・モードと *MSCS*

Windows 上の IBM MQ を使用してクラスター・システムを実行するモードには、アクティブ/パッシブとアクティブ/アクティブの 2 つがあります。

注 : MSCS を Microsoft Transaction Server (COM+) で使用する場合、Active/Active モードは使用できません。

アクティブ/パッシブ・モード

アクティブ/パッシブ・モードでは、コンピューター A 上でアプリケーションが実行され、コンピューター B はバックアップ専用で、MSCS が障害を検出した場合のみに使用されます。

このモードは、1 つの共有ディスクでのみ使用できますが、いずれかのアプリケーションがフェイルオーバーを引き起こした場合、**すべての**アプリケーションをグループとして転送する必要があります (共有ディスクにアクセスできるのは一度に 1 つのコンピューターのみであるためです)。

MSCS を構成する際に、A を優先 コンピューターとすることができます。 これを行うと、コンピューター A が修理されるかまたは取り替えられて、再び正常に機能すると、MSCS がこれを検出してアプリケーションを自動的にコンピューター A に切り替えます。

複数のキュー・マネージャーを実行する場合、キュー・マネージャーごとに個別の共有ディスクを持つことを考慮してください。 さらに、各キュー・マネージャーを MSCS 内の別々のグループに設置してください。 これで、あるキュー・マネージャーが他のキュー・マネージャーに影響を与えずに、別のコンピューターにフェイルオーバーすることができます。

アクティブ/アクティブ・モード

アクティブ/アクティブ・モードでは、コンピューター A とコンピューター B の両方でアプリケーションが実行され、各コンピューター上のグループは一方のコンピューターをバックアップとして使用するように設定されます。 コンピューター A 上で障害が検出された場合、MSCS は状態データをコンピューター B に転送し、そこでアプリケーションを再開します。 コンピューター B は独自のアプリケーションと A のアプリケーションを実行します。

このセットアップでは、少なくとも 2 つの共用ディスクが必要です。 MSCS を構成する際に、A を A のアプリケーションの優先コンピューター、B を B のアプリケーションの優先コンピューターとして設定できます。 フェイルオーバーおよび修理後に、各アプリケーションは自動的に元のコンピューターに戻ります。

IBM MQ では、これは、例えば 2 つのキュー・マネージャー (コンピューター A 上とコンピューター B 上でそれぞれ 1 つずつ) が実行されており、それぞれのキュー・マネージャーが自分のコンピューターの完全な能力を活用できることを意味します。 コンピューター A で障害が発生すると、両方のキュー・マネージャーがコンピューター B で稼働します。 これは、一台のコンピューターの能力を共有することで、大量のデータを高速で処理する能力を低下させます。 ただし、A の障害を見つけて修理する間も、重要なアプリケーションは依然として使用可能になっています。

Windows *MSCS* で使用するようにキュー・マネージャーを作成する

この手順により、 Microsoft Cluster Service (MSCS) 制御下での準備と配置に適した方法で新しいキュー・マネージャーが作成されるようになります。

注 : Windows Server 2016 以降、 Microsoft Cluster Service (MSCS) の新しい名前は Windows Server Failover Clustering (WSFC) になりました。

まず、キュー・マネージャーとそのすべてのリソースをローカル・ドライブ上に作成してから、ログ・ファイルとデータ・ファイルを共有ディスクに移行します。 (この操作は逆にしても構いません。) キュー・マネージャーとそのリソースを共有ドライブ上に作成**しないでください**。

MSCS で使用するようにキュー・マネージャーを作成するには、コマンド・プロンプトから作成するか、または IBM MQ Explorer で作成するかの 2 つの方法があります。 コマンド・プロンプトを使用する利点は、キュー・マネージャーが停止状態 で作成され、手動による始動 に設定されて MSCS の準備が整う点です。 (IBM MQ Explorer の場合、自動的に新しいキュー・マネージャーが始動し、作成後は自動的に始動するように設定します。 これを変更する必要があります。)

コマンド・プロンプトからキュー・マネージャーを作成する

MSCS で使用するように、コマンド・プロンプトからキュー・マネージャーを作成するには、以下のステップに従います。

1. 環境変数 MQSPREFIX がローカル・ドライブ (例えば、 C:\IBM MQ) を参照するように設定されていることを確認します。 これを変更した場合、システム・アカウントが変更を適用するよう、マシンをリブートしてください。 変数を設定しない場合、キュー・マネージャーは、キュー・マネージャーの IBM MQ デフォルト・ディレクトリーで作成されます。
2. **crtmqm** コマンドを使用してキュー・マネージャーを作成します。 例えば、デフォルト・ディレクトリーに mscs_test というキュー・マネージャーを作成するには、次のように入力します。

   ```
   crtmqm mscs_test
   ```

3. 495 ページの『キュー・マネージャーを MSCS ストレージに移動する』に進みます。

IBM MQ Explorer を使用してキュー・マネージャーを作成する

MSCS で使用するように、IBM MQ Explorer を使用してキュー・マネージャーを作成するには、以下のステップに従います。

1. 「スタート」メニューから IBM MQ Explorer を始動します。
2. ナビゲーター・ビューで、ツリー・ノードを展開して、「Queue Managers (キュー・マネージャー)」ツリー・ノードを探します。
3. 「キュー・マネージャー」ツリー・ノードを右クリックし、「**新規**」 > 「**キュー・マネージャー**」を選択します。 「Create Queue Manager (キュー・マネージャーの作成)」パネルが表示されます。
4. ダイアログを完成 (ステップ 1) してから、「**次へ>**」をクリックします。
5. ダイアログを完成 (ステップ 2) してから、「**次へ>**」をクリックします。
6. ダイアログを完成 (ステップ 3) してから、「**キュー・マネージャーの開始**」および「**サーバー接続チャネルの作成**」が選択されていないことを確認し、「**次へ>**」をクリックします。
7. ダイアログを完成 (ステップ 4) してから、「**Finish (完了)**」をクリックします。
8. 495 ページの『キュー・マネージャーを MSCS ストレージに移動する』に進みます。

Windows キュー・マネージャーを *MSCS* ストレージに移動する

この手順では、 Microsoft Cluster Service (MSCS) 制御下に置くのに適したものになるように、既存のキュー・マネージャーを構成します。

この操作を行うには、ログ・ファイルとデータ・ファイルを共有ディスクに移動して、障害が起きた場合に他のコンピューターで使用できるようにしておく必要があります。 例えば、既存のキュー・マネージャーには、C:\WebSphere MQ\log*QMname* や C:\WebSphere MQ\qmgrs*QMname* などのパスが存在する場合があります。

重要 : ファイルは手動で移動せずに、このトピックで説明されているように、IBM MQ MSCS サポートの一部として提供されるユーティリティー・プログラムを使用してください。

移動するキュー・マネージャーが TLS 接続を使用し、その TLS のキー・リポジトリーがローカル・マシン上のキュー・マネージャーのデータ・ディレクトリー内にある場合は、そのキー・リポジトリーは、他のキュー・マネージャーによって共用ディスクに移動されます。 デフォルトでは、TLS 鍵リポジトリーの場所を指定するキュー・マネージャー属性 SSLKEYR は、キュー・マネージャー・データ・ディレクトリーの下にある *MQ_INSTALLATION_PATH*\qmgrs\QMGRNAME\ssl\key に設定されます。
MQ_INSTALLATION_PATH は、IBM MQ がインストールされている上位ディレクトリーを表します。
hamvmqm コマンドは、このキュー・マネージャー属性を変更しません。 この状態では、IBM MQ Explorer または MQSC コマンド ALTER QMGR を使用して、キュー・マネージャー属性 SSLKEYR が新しい TLS キー・リポジトリー・ファイルを指すように変更する必要があります。

注 : Windows Server 2016 以降、 Microsoft Cluster Service (MSCS) の新しい名前は Windows Server Failover Clustering (WSFC) になりました。

手順は次のとおりです。

1. キュー・マネージャーをシャットダウンし、エラーがないことを確認します。
2. キュー・マネージャーのログ・ファイルまたはキュー・ファイルがすでに共有ディスクに保管されている場合は、この手順の以下の部分を省略し、497 ページの『MSCS 制御下にキュー・マネージャーを置く』へ直接進みます。
3. キュー・ファイルおよびログ・ファイルの完全メディア・バックアップを作成し、バックアップを安全な位置に保管します (この重要性については 502 ページの『キュー・マネージャーのログ・ファイル』を参照)。
4. 適切な共用ディスク・リソースを既に持っている場合は、ステップ 6 に進みます。 そうでない場合は、MSCS クラスター・アドミニストレーターを使用して、キュー・マネージャーのログ・ファイルおよびデータ(キュー)ファイルを保管するための十分な容量を持つタイプ共有ディスクのリソースを作成します。
5. 共有ディスクをテストするため、MSCS クラスター・アドミニストレーターを使用して共有ディスクをクラスター・ノードから別のノードへ移動し、また元に戻します。
6. 共用ディスクが、キュー・マネージャーのログ・ファイルとデータ・ファイルをローカルで保管している クラスター・ノード上でオンラインになっていることを確認します。
7. ユーティリティー・プログラムを実行して、次のようにキュー・マネージャーを移動します。

   ```
   hamvmqm /m qmname /dd " e: \
   IBM MQ " /ld " e: \
   IBM MQ \log"
   ```

   qmname は該当するキュー・マネージャー名に、*e* は共有ディスク・ドライブ文字に、*IBM MQ* は選択したディレクトリーに、それぞれ置き換えて入力します。 ディレクトリーは、なければ作成されます。
8. IBM MQ Explorer を使用し、キュー・マネージャーをテストして、機能することを確認します。 以下に例を示します。
   a. キュー・マネージャーのツリー・ノードを右マウス・ボタン・クリックして、「**Start (開始)**」を選択します。 キュー・マネージャーが開始します。
   b.「**キュー**」 ツリー・ノードを右クリックし、 「**新規**」 > 「**ローカル・キュー ...**」を選択します。 キューに名前を付けます。
   c.「**完了 (Finish)**」 をクリックします。
   d. キューを右マウス・ボタン・クリックして、「**Put Test Message... (テスト・メッセージの書き込み...)**」を選択します。 「Put Test Message (テスト・メッセージの書き込み)」パネルが表示されます。
   e. メッセージ・テキストを入力してから、「**Put Test Message (テスト・メッセージの書き込み)**」をクリックし、パネルをクローズします。
   f. キューを右クリックして、「**メッセージの参照...**」を選択します。 「メッセージ・ブラウザー」パネルが表示されます。
   g. メッセージがキュー上にあることを確認して、「**Close (クローズ)**」をクリックします。 「メッセージ・ブラウザー」パネルが閉じます。
   h. キューを右クリックして、「**Clear Messages... (メッセージの消去...)**」を選択します。 キューにあるメッセージがクリアされます。
   i. キューを右クリックして、「**削除...**」を選択します。 確認パネルが表示されるため、「**OK (了解)**」をクリックします。 キューが削除されます。
   j. キュー・マネージャーのツリー・ノードを右マウス・ボタン・クリックして、「**Stop... (停止...)**」を選択します。 「キュー・マネージャーの終了」パネルが表示されます。
   k.「**OK**」をクリックします。 キュー・マネージャーが停止します。
9. IBM MQ のように、キュー・マネージャーの始動属性が手動に設定されていることを確認します。 IBM MQ Explorer のキュー・マネージャー・プロパティー・パネルで、「Startup (始動)」フィールドを `manual` に設定します。
10. 497 ページの『MSCS 制御下にキュー・マネージャーを置く』に進みます。

Windows *MSCS 制御下にキュー・マネージャーを置く*

キュー・マネージャーを Microsoft Cluster Service (MSCS) 制御下に置く方法 (前提条件タスクを含む)。

注 : Windows Server 2016 以降、 Microsoft Cluster Service (MSCS) の新しい名前は Windows Server Failover Clustering (WSFC) になりました。

キュー・マネージャーを MSCS/WSFC 制御下に置く前

キュー・マネージャーを MSCS/WSFC 制御下に置く前に、以下のステップを実行します。

1. 492 ページの『MSCS クラスター化用の IBM MQ のセットアップ』で説明されているように、 IBM MQ とその MSCS/WSFC サポートがクラスター内の両方のマシンにインストールされていること、および各コンピューター上のソフトウェアが同一であることを確認してください。
2. **haregtyp** ユーティリティー・プログラムを使用し、IBM MQ を MSCS リソース・タイプとして、すべてのクラスター・ノードに登録してください。 504 ページの『MSCS ユーティリティー・プログラムのサポート』を参照してください。
3. まだ作成していない場合は、 MSCS/WSFC で使用するキュー・マネージャーを作成します。
4. キュー・マネージャーをすでに作成してある場合、またはキュー・マネージャーがすでに存在している場合は、495 ページの『キュー・マネージャーを MSCS ストレージに移動する』にある手順を実行したかどうかを確認してください。
5. キュー・マネージャーが実行中の場合は、コマンド・プロンプトまたは IBM MQ エクスプローラーを使用して停止します。
6. このトピックの以下の Windows 手順のいずれかに進む前に、共用ドライブの MSCS/WSFC 操作をテストしてください。

Windows Server 2019 または 2022 年

Windows Server 2019 または 2022 上でキュー・マネージャを MSCS/WSFC 制御下に置くには、以下の手順を使用します：

1. キュー・マネージャーのホストであるクラスター・ノード・コンピューターにログインします。または、クラスター管理権限を持つユーザーとしてリモート・ワークステーションにログインし、キュー・マネージャーのホストであるクラスター・ノードに接続します。
2. フェイルオーバー・クラスター管理ツールを開始します。
3. 「**フェイルオーバー・クラスター管理**」 > 「**クラスターの接続 ...**」 を右クリックします。 クラスターへの接続を開きます。
4. Windows の以前のバージョン上の MSCS Cluster Administrator で使用されたグループ方式とは異なり、フェイルオーバー・クラスター管理ツールはサービスおよびアプリケーションの概念を使用します。 構成されたサービスまたはアプリケーションには、1 つのアプリケーションをクラスター化するために必要なすべてのリソースが含まれています。 以下のようにして、WSFC の下にキュー・マネージャーを構成することができます。
   a. クラスターを右クリックし、「**役割の構成**」を選択して構成ウィザードを開始します。
   b. 「サービスまたはアプリケーションの選択」パネルで、「**他のサーバー**」を選択します。
   c. クライアント・アクセス・ポイントとして、適切な IP アドレスを選択します。

      このアドレスは未使用の IP アドレスである必要があり、仮想 キュー・マネージャーへ接続するためにクライアントと他のキュー・マネージャーによって使用されます。 この IP アドレスはどちらかのノードの通常の (静的な) アドレスではなく、両方の間を浮動する 追加のアドレスです。 WSFC はこのアドレスのルーティングを処理しますが、アドレスに到達できるかどうかは検査 **しません** 。
   d. キュー・マネージャー専用のストレージ・デバイスを割り当てます。 このデバイスは、リソース・インスタンスとして作成してからでなければ割り当てることができません。

      1 つのドライブにログとキュー・ファイルの両方を保管することもできますし、複数のドライブに分割することもできます。 いずれの場合も、各キュー・マネージャーが独自の共用ディスクを持っている場合は、キュー・マネージャーが使用するすべてのドライブが、キュー・マネージャー(つまり、ドライブに依存していないもの)に対して、排他的なものであることを確認してください。 また、キ

ュー・マネージャーによって使用されるすべての ドライブ用のリソース・インスタンスを作成する必要があります。

ドライブのリソース・タイプは、使用する SCSI サポートに応じて異なります。SCSI アダプターの説明を参照してください。 各共有ドライブにすでにグループとリソースが存在する可能性もあります。 その場合は、各ドライブのリソース・インスタンスを作成する必要はありません。 そのインスタンスを現在のグループからキュー・マネージャー用に作成したグループへ移動します。

各ドライブ・リソースごとに、両方のノードの所有者を設定します。 従属リソースを「none (なし)」に設定します。

e.「リソース・タイプの選択」パネルで、「**MQSeries MSCS**」リソースを選択します。

f. ウィザードでの残りのステップを完了します。

5. リソースをオンラインにする前に、MQSeries® MSCS リソースは以下の追加の構成を必要とします。

a.「新しい MQSeries MSCS」という名前のリソースを含む、新しく定義されたサービスを選択します。

b. MQ リソースで、「**プロパティー**」を右クリックします。

c. 以下のように、リソースを構成します。

- `Name` 対象のキュー・マネージャーを識別しやすくする名前を選択します。
- `Run in a separate Resource Monitor` より良い隔離のため
- `Possible owners` 両方のノードの設定
- `Dependencies` このキュー・マネージャーのドライブと IP アドレスを追加します。

  警告 : これらの依存関係を追加できないことは、IBM MQ がフェイルオーバー時にキュー・マネージャーの状況を誤ったクラスターのディスクに書き込もうとしていることを意味しています。 多くのプロセスが同時にこのディスクに書き込もうとしているため、IBM MQ プロセスの実行がブロックされる場合があります。
- `Parameters` 以下のとおりです。
  - `QueueManagerName` (必須)。このリソースを制御するキュー・マネージャーの名前です。 このキュー・マネージャーはローカル・コンピューター上になければなりません。
  - `PostOnlineCommand` (オプショナル)。キュー・マネージャー・リソースの状態が オフラインからオンラインに変わった時に必ず実行されるプログラムを指定できます。 詳細については、503 ページの『MSCS の PostOnlineCommand と PreOfflineCommand』を参照してください。
  - `PreOfflineCommand` (オプショナル)。キュー・マネージャー・リソースの状態が オンラインからオフラインに変わった時に必ず実行されるプログラムを指定できます。 詳細については、503 ページの『MSCS の PostOnlineCommand と PreOfflineCommand』を参照してください。

    注 : *looksAlive* ポーリング間隔は、デフォルト値の 5000 ミリ秒に設定されます。 *isAlive* ポーリング間隔は、デフォルト値の 60000 ミリ秒に設定されます。 これらのデフォルトは、リソース定義が完了した後にのみ変更できます。 詳細については、499 ページの『MSCS での looksAlive と isAlive のポーリング』を参照してください。

d. 任意で、優先ノードを設定します (ただし、503 ページの『MSCS の優先ノードの使用』の内容に注意してください)。

e. フェイルオーバー・ポリシー は、デフォルトで適切な値に設定されますが、リソース・フェイルオーバー とグループ・フェイルオーバー を制御するしきい値および期間を調整して、その値をキュー・マネージャーにかかる負荷に合わせることもできます。

6. MSCS クラスター・アドミニストレーターでキュー・マネージャーをオンラインにし、テスト・ワークロードをかけることによってキュー・マネージャーをテストします。 テスト用のキュー・マネージャーで実験する場合は、IBM MQ エクスプローラーを使用します。 以下に例を示します。

a.「**キュー**」 ツリー・ノードを右クリックし、 「**新規**」 >「**ローカル・キュー ...**」を選択します。 キューに名前を付けます。

b.「**完了 (Finish)**」 をクリックします。 キューが作成され、コンテンツ・ビューに表示されます。

c. キューを右マウス・ボタン・クリックして、「**Put Test Message... (テスト・メッセージの書き込み...)**」を選択します。 「Put Test Message (テスト・メッセージの書き込み)」パネルが表示されます。

d. メッセージ・テキストを入力してから、「**Put Test Message (テスト・メッセージの書き込み)**」をクリックし、パネルをクローズします。

e. キューを右クリックして、「**メッセージの参照...**」を選択します。 「メッセージ・ブラウザー」パネルが表示されます。

f. メッセージがキュー上にあることを確認して、「**Close (クローズ)**」をクリックします。 「メッセージ・ブラウザー」パネルが閉じます。

g. キューを右クリックして、「**Clear Messages... (メッセージの消去...)**」を選択します。 キューにあるメッセージがクリアされます。

h. キューを右クリックして、「**削除...**」を選択します。 確認パネルが表示されるため、「**OK (了解)**」をクリックします。 キューが削除されます。

7. MSCS クラスター・アドミニストレーターを使用して、キュー・マネージャーをオフラインにできるかテストし、さらにオンラインに戻せるかテストします。

8. フェイルオーバーをシミュレートします。

   MSCS クラスター・アドミニストレーターで、キュー・マネージャーを含むグループを右クリックして、`Move Group` を選択します。 これが完了するには数分かかることがあります。 (キュー・マネージャーをもう一方のノードへ移動する操作を別の機会に行うことによって時間を節約する場合は、495 ページの『キュー・マネージャーを MSCS ストレージに移動する』の手順に従ってください。) 右クリックして `Initiate Failure` を選択することもできます。 アクション (ローカル再始動またはフェイルオーバー) は、現在の状態と構成設定によって異なります。

Windows *MSCS での looksAlive と isAlive のポーリング*

looksAlive および *isAlive* は、 Microsoft Cluster Service (MSCS) がリソース・タイプ提供のライブラリー・コードにコールバックし、リソースが自身の作業状況を判別するために検査を実行することを要求する間隔です。 これは最終的には、MSCS がリソースのフェイルオーバーを行うかどうかを判別します。

注 : Windows Server 2016 以降、 Microsoft Cluster Service (MSCS) の新しい名前は Windows Server Failover Clustering (WSFC) になりました。

looksAlive 間隔が経過すると (デフォルトは 5000 ms)、キュー・マネージャー・リソースが呼び出され、その状況が適切なものであるかどうかを判別する独自の検査を実行します。

isAlive 間隔が経過すると (デフォルトは 30000 ms)、キュー・マネージャー・リソースへの別の呼び出しが行われ、リソースが正しく機能しているかどうかを判別する別の検査が実行されます。 これにより、2 つのレベルのリソース・タイプ検査を行うことができます。

1. *looksAlive* 状況検査では、リソースが機能しているかどうかが明らかにされます。
2. *isAlive* はより重要な検査であり、キュー・マネージャー・リソースがアクティブであるかどうかが判別されます。

キュー・マネージャー・リソースがアクティブでないと判断されると、MSCS は、他の拡張 MSCS オプションに基づいて、クラスター内の別のノードへのリソースおよび関連する従属リソースのフェイルオーバーをトリガーします。 詳細については、MSCS の資料を参照してください。

Windows *MSCS 制御からキュー・マネージャーを除去する*

Microsoft Cluster Service (MSCS) 制御からキュー・マネージャーを除去して、手動管理に戻すことができます。

注 : Windows Server 2016 以降、 Microsoft Cluster Service (MSCS) の新しい名前は Windows Server Failover Clustering (WSFC) になりました。

保守操作のために、MSCS 制御からキュー・マネージャーを除去する必要はありません。 この除去は、MSCS クラスター・アドミニストレーターを使用して、キュー・マネージャーを一時的にオフラインにすることによって行えます。 MSCS 制御下からのキュー・マネージャーの除去は、より永続的な状態への変更であり、MSCS によるキュー・マネージャーの制御がこれ以上必要なくなったと判断できた時にだけ行います。

除去するキュー・マネージャーが TSL 接続を使用する場合は、IBM MQ エクスプローラーまたは MQSC コマンド ALTER QMGR を使用して、キュー・マネージャー属性 SSLKEYR がローカル・ディレクトリー上の TLS キー・リポジトリー・ファイルを指すように変更する必要があります。

手順は次のとおりです。

1. 500 ページの『MSCS からキュー・マネージャーをオフラインにする』で説明されているように、MSCS クラスター・アドミニストレーターを使用してキュー・マネージャー・リソースをオフラインにします。
2. リソース・インスタンスを破棄します。 この操作では、キュー・マネージャーは破棄されません。
3. 必要に応じて、共用ドライブからローカル・ドライブへキュー・マネージャー・ファイルを移行します。 その方法については、500 ページの『キュー・マネージャーを MSCS ストレージから戻す』を参照してください。
4. キュー・マネージャーをテストします。

MSCS からキュー・マネージャーをオフラインにする

MSCS からキュー・マネージャーをオフラインにするには、以下のステップを実行します。

1. MSCS クラスター・アドミニストレーターを始動します。
2. クラスターへの接続をオープンします。
3. Groups または Role (Windows 2012 を使用している場合) を選択し、移動するキュー・マネージャーが含まれているグループを開きます。
4. キュー・マネージャーのリソースを選択します。
5. それを右クリックして、 Offline を選択します。
6. アクションが完了するまで待ちます。

キュー・マネージャーを MSCS ストレージから戻す

この手順に従うと、キュー・マネージャーがコンピューターのローカル・ドライブに戻るように構成されます。つまり、通常の IBM MQ キュー・マネージャーになります。 この操作を行うには、ログ・ファイルとデータ・ファイルを共有ディスクから移動します。 例えば、既存のキュー・マネージャーには、E:\WebSphere MQ\log*QMname* や E:\WebSphere MQ\qmgrs*QMname* などのパスが存在する場合があります。 ファイルを手動で移動しないでください。 IBM MQ MSCS サポートの一部として提供されている **hamvmqm** ユーティリティー・プログラムを使用してください。

1. キュー・ファイルおよびログ・ファイルの完全メディア・バックアップを作成し、バックアップを安全な位置に保管します (この重要性については 502 ページの『キュー・マネージャーのログ・ファイル』を参照)。
2. 使用するローカル・ドライブを決めて、キュー・マネージャーのログ・ファイルと データ (キュー) ファイルを保管するのに十分な容量があるか確認します。
3. ファイルが現在収められている共用ディスクが、キュー・マネージャーのログ・ファイルと データ・ファイルを移動する先のクラスター・ノード上でオンラインになっていることを確認します。
4. ユーティリティー・プログラムを実行して、次のようにキュー・マネージャーを移動します。

   ```
   hamvmqm /m qmname /dd " c:\
   IBM MQ " /ld "c:\
   IBM MQ \log"
   ```

   qmname は該当するキュー・マネージャー名に、*c* はローカル・ディスク・ドライブ文字に、 *IBM MQ* は選択したディレクトリー (選択したディレクトリーが存在していない場合は作成される) に、それぞれ置き換えて入力します。
5. キュー・マネージャーをテストして、機能することを確認します (495 ページの『キュー・マネージャーを MSCS ストレージに移動する』を参照してください)。

Windows *MSCS* の使用に関するヒント

このセクションには、 IBM MQ support for Microsoft Cluster Service (MSCS) を効果的に使用するために役立つ一般情報が記載されています。

注 : Windows Server 2016 以降、 Microsoft Cluster Service (MSCS) の新しい名前は Windows Server Failover Clustering (WSFC) になりました。

あるマシンから別のマシンにキュー・マネージャーをフェイルオーバーするには、どれくらいの時間がかかるでしょうか。 これは、キュー・マネージャーのワークロード量、およびトラフィックの構成 (例えば、トラフィックがどれくらい持続的か、同期点内のトラフィック、障害の前にトラフィックがどれくらいコミットされていたか) によってかなり異なります。 IBM の実験では、フェイルオーバーおよびフェイルバック時間はおよそ 1 分でした。 この実験は、非常に負荷の少ないキュー・マネージャーで行われましたが、実際の時間は負荷によってかなり違いがあります。

Windows *MSCS* の動作確認

以下のステップに従って、稼働中の MSCS クラスターがあることを確認してください。

494 ページの『MSCS で使用するようにキュー・マネージャーを作成する』の手順で始まるタスクの説明では、リソースの作成、移動、および破棄が可能な MSCS クラスターが稼働中であることが前提になります。 そのようなクラスターがあることを確認する方法は次のとおりです。

1. MSCS クラスター・アドミニストレーターを使用して、グループを作成します。
2. そのグループ内に、システム・クロック (パス名 `C:\winnt\system32\clock.exe` および `C:\`の作業ディレクトリー) を指定して、汎用アプリケーション・リソースのインスタンスを作成します。
3. リソースをオンラインにできること、そのリソースを含むグループを他のノードに移動できること、およびリソースをオフラインにできることを確認します。

Windows 手動始動と *MSCS*

MSCS が管理するキュー・マネージャーでは、必ず始動属性を手動に設定しなければなりません。 これによって、IBM MQ MSCS サポートは、即時にキュー・マネージャーを開始しなくても MQSeries サービスを再開できます。

IBM MQ MSCS サポートは、モニターおよび制御を行えるように、サービスを再始動できなければなりませんが、どのキュー・マネージャーがどのマシンで実行されるかを自身で管理し続ける必要があります。 詳しくは、495 ページの『キュー・マネージャーを MSCS ストレージに移動する』を参照してください。

Windows *MSCS* およびキュー・マネージャー

MSCS を使用しているときのキュー・マネージャーに関する考慮事項。

他のノードに対応するキュー・マネージャーを作成する

クラスタリングが IBM MQ で機能するためには、ノード A 上のキュー・マネージャーとノード B 上のキュー・マネージャーを同一にする必要があります。 ただし、二番目のものを明示的に作成する必要はありません。 1 つのノードでキュー・マネージャーを作成または準備し、それを495 ページの『キュー・マネージャーを MSCS ストレージに移動する』で説明されている方法で他のノードに移動すれば、そのキュー・マネージャーは完全にそのノード上で複製されます。

デフォルト・キュー・マネージャー

MSCS 制御下では、デフォルト・キュー・マネージャーは使用しないでください。 キュー・マネージャーには、そのキュー・マネージャーをデフォルトにするためのプロパティーがありません。 IBM MQ には独自のレコードが個別に存在します。 デフォルトに設定された キュー・マネージャーをフェイルオーバー時に他のコンピューターへ移動しても、移動先では デフォルトになりません。 すべてのアプリケーションが特定のキュー・マネージャーをその名前で 参照するように設定してください。

キュー・マネージャーの削除

キュー・マネージャーをノード間で移動すると、その詳細が両方のコンピューターのレジストリーに記録されます。 これを削除する場合は、一方のコンピューターで削除し、504 ページの『MSCS ユーティリティー・プログラムのサポート』 で説明しているようにユーティリティーを実行し、もう一方のコンピューターのレジストリーを クリーンアップします。

既存キュー・マネージャーのサポート

キュー・マネージャーのログ・ファイルとキュー・ファイルを、2 つのマシン間の共有 SCSI バス上の ディスクに書き込むことが可能な場合は、既存のキュー・マネージャーを MSCS 制御下に置くことが できます (492 ページの図 71 を参照してください)。 MSCS リソースの作成中は、キュー・マネージャーを一時的にオフラインにする必要があります。

キュー・マネージャーを新規作成する場合は、MSCS から独立して作成し、テストしてから、MSCS 制御下に置きます。 以下を参照してください。

- 494 ページの『MSCS で使用するようにキュー・マネージャーを作成する』
- 495 ページの『キュー・マネージャーを MSCS ストレージに移動する』
- 497 ページの『MSCS 制御下にキュー・マネージャーを置く』

管理するキュー・マネージャーを MSCS に指示する

MSCS 制御の下にどのキュー・マネージャーを置くかは、MSCS クラスター・アドミニストレーターを使用して、キュー・マネージャーごとにリソース・インスタンスを作成することによって選択できます。 このプロセスで、そのインスタンスを管理するキュー・マネージャーを選択できる、リソースのリストが表示されます。

キュー・マネージャーのログ・ファイル

キュー・マネージャーを MSCS ストレージへ移動すると、そのログ・ファイルとデータ・ファイルは共有ディスクへ移動されます (この例については、495 ページの『キュー・マネージャーを MSCS ストレージに移動する』 を参照してください)。

移動する前に、キュー・マネージャーを完全にシャットダウンし、データ・ファイルとログ・ファイルの全バックアップを取ることをお勧めします。

複数のキュー・マネージャー

IBM MQ MSCS サポートを使用すると、各マシンで複数のキュー・マネージャーを実行することができ、個々のキュー・マネージャーを MSCS 制御下に置くことができます。

Windows 常に *MSCS* を使用してクラスターを管理する

制御コマンドか、または IBM MQ Explorer を使用して、MSCS の制御下にあるキュー・マネージャーに対して直接始動および停止操作を実行しないでください。 その代わり、MSCS クラスター・アドミニストレーターを使用して、キュー・マネージャーをオンラインまたはオフラインにします。

MSCS クラスター・アドミニストレーターの使用は一部には、実際にはキュー・マネージャーを MSCS の制御外で始動したにもかかわらず、キュー・マネージャーがオフラインであることを MSCS が報告するために引き起こされる混乱を避けるためです。 さらに重要なことに、MSCS を使用せずにキュー・マネージャーを停止すると、MSCS はこれを障害として検出し、他のノードへのフェイルオーバーが開始されます。

Windows *MSCS* のアクティブ/アクティブ・モードでの作業

アクティブ/アクティブ・モードでは、MSCS クラスター内にある 2 つのコンピューターの両方がキュー・マネージャーを実行していることがあります。 完全に使用されていないマシンを待機用として用意する必要はありません (その必要があれば、これはアクティブ/パッシブ・モードで可能です)。

両方のマシンでワークロードを実行する場合、十分なパフォーマンス・レベルでクラスター・ワークロード全体を実行するには、それぞれのマシンに十分な容量 (プロセッサー、メモリー、2 次ストレージ) が必要です。

注 : MSCS を Microsoft Transaction Server (COM+) で使用する場合、Active/Active モードは使用**できません**。 これは、IBM MQ を MSCS および COM+ と共に使用して、次のことを行うためです。

- IBM MQ の COM+ サポートを使用するアプリケーション・コンポーネントは、COM+ の一部である分散トランザクション調整プログラム (Distributed Transaction Coordinator (DTC)) と同じコンピューター上で実行する必要があります。
- キュー・マネージャーも、同じコンピューター上で実行する必要があります。
- DTC は MSCS リソースとして構成する必要があります。そのように構成しておくと、いつでもクラスター内のコンピューターの 1 つでのみ実行できます。

Windows *MSCS の PostOnlineCommand と PreOfflineCommand*

これらのコマンドを使用して、IBM MQ MSCS サポートをその他のシステムと統合することができます。 これらを使用して IBM MQ コマンドを発行できますが、いくつかの制限があります。

これらのコマンドは、タイプ `IBM MQ MSCS` のリソースに対するパラメーターで指定します。 これらのコマンドを使用して、IBM MQ MSCS サポートをその他のシステムまたはプロシージャーと統合することができます。 例えば、メール・メッセージの送信、ページャーのアクティブ化、または別のモニター・システムによって キャプチャーされる別の形式のアラートの生成を行うプログラムの名前を指定することが可能です。

PostOnlineCommand は、リソースがオフラインからオンラインに変更されると起動され、PreOfflineCommand は、リソースがオンラインからオフラインに変更されると起動されます。 起動時、これらのコマンドはデフォルトで Windows システム・ディレクトリーから実行されます。 IBM MQ は 32 ビット・リソース・モニター・プロセスを使用するため、Windows64 ビット・システムでは、`\Windows\system32` ディレクトリーではなく、`\Windows\SysWOW64` ディレクトリーになります。 詳しくは、 Windows x64 環境でのファイル・リダイレクトに関する Microsoft 資料を参照してください。 どちらのコマンドも、MSCS クラスター・サービスを実行するのに使用されるユーザー・アカウントの下で実行され、また非同期で起動されます。 IBM MQ MSCS サポートは、それらのコマンドの完了を待たずに処理を続行します。 このため、この後のクラスター操作を妨害したり、遅らせたりする危険はありません。

また、これらのコマンドを使用して IBM MQ コマンドを実行し、例えばリクエスター・チャネルを再始動できます。 ただし、コマンドはキュー・マネージャーの状態変更の時点で実行されたため、長時間実行される機能を実行するようには意図されていません。また、キュー・マネージャーの現在の状態を憶測することはできません。キュー・マネージャーがオンラインになった直後に、アドミニストレーターがオフライン・コマンドを発行する可能性があります。

キュー・マネージャーの状態に依存するプログラムを実行する場合は、`MSCS Generic Application` リソース・タイプのインスタンスを作成し、それらをキュー・マネージャー・リソースと同じ MSCS グループに配置し、それらをキュー・マネージャー・リソースに依存させることを検討してください。

Windows *MSCS の優先ノードの使用*

MSCS でアクティブ/アクティブ・モードを使用する場合は、各キュー・マネージャーの優先ノード を構成すると便利です。 しかし、一般的には優先ノードを設定しないで、手動フェイルバックのみを使用することをお勧めします。

その他の比較的ステートレスなリソースとは異なり、キュー・マネージャーでは、あるノードから別のノードへのフェイルオーバー (またはフェイルバック) に時間がかかることがあります。 不必要な停止を避けるため、キュー・マネージャーをリカバリー済みノードにフェイルバックする前にそのノードをテストしてください。 これにより、 `immediate` フェイルバック設定が使用できなくなります。 1 日のうちのある時間にフェイルバックが行われるように構成できます。

ノードが完全にリカバリーされたことを確認できたときに、 キュー・マネージャーを手動で必要なノードに移動する方法が最も安全です。 これにより、 `preferred node` オプションを使用できなくなります。

Windows *MSCS へのインストール時の COM+ エラー*

新しくインストールされた MSCS クラスターに IBM MQ をインストールする場合、アプリケーション・イベント・ログに、ソース COM+ のエラーおよびイベント ID 4691 が報告されることがあります。

これは、 Microsoft 分散トランザクション・コーディネーター (MSDTC) が Microsoft Cluster Server (MSCS) 環境で実行するように構成されていない場合に、MSCS 環境で IBM MQ を実行しようとしていることを意味します。 クラスター環境で MSDTC を構成する方法については、Microsoft の資料を参照してください。

Windows *MSCS* ユーティリティー・プログラムのサポート

コマンド・プロンプトで実行できる IBM MQ Support for Microsoft Cluster Service (MSCS) ユーティリティー・プログラムのリストです。

注 : Windows Server 2016 以降、 Microsoft Cluster Service (MSCS) の新しい名前は Windows Server Failover Clustering (WSFC) になりました。

MSCS の IBM MQ サポートには、以下のユーティリティー・プログラムが含まれています。

リソース・タイプの登録/登録解除
`haregtyp.exe`

IBM MQ MSCS リソース・タイプを 登録抹消 すると、そのタイプのリソースを作成できなくなります。 MSCS では、クラスター内にまだ存在しているインスタンスのタイプとして指定されているリソース・タイプを登録解除することはできません。

1. MSCS 制御下で実行されているキュー・マネージャーを停止するため、MSCS クラスター・アドミニストレーターを使用して、500 ページの『MSCS からキュー・マネージャーをオフラインにする』の手順に従ってキュー・マネージャーをオフラインにします。
2. MSCS クラスター・アドミニストレーターを使用して、リソース・インスタンスを削除します。
3. リソース・タイプを登録解除するため、コマンド・プロンプトで次のコマンドを入力します。

```
haregtyp /u
```

タイプを後から登録 (または後から再登録) する場合は、コマンド・プロンプトで次のコマンドを入力します。

```
haregtyp /r
```

MSCS ライブラリーを正常に登録した後、IBM MQ のインストール後まだシステムをリブートしていない場合は、リブートする必要があります。

キュー・マネージャーを MSCS ストレージに移動する
`hamvmqm.exe`

495 ページの『キュー・マネージャーを MSCS ストレージに移動する』 を参照してください。

キュー・マネージャーをノードから削除する
`hadltmqm.exe`

クラスターにキュー・マネージャーがあり、そのキュー・マネージャーがあるノードから別のノードに移動し、今はそれを破棄したいという場合を考えてください。 現在キュー・マネージャーが存在するノードで、そのキュー・マネージャーを削除するには、IBM MQ エクスプローラーを使用します。 削除後も、キュー・マネージャーのレジストリー項目は別のコンピューター上に引き続き存在します。 これらを削除するには、そのコンピューター上でプロンプトに次のコマンドを入力してください。

```
hadltmqm /m qmname
```

qmname は、除去されるキュー・マネージャーの名前です。

セットアップの詳細を確認および保管する
`amqmsysn.exe`

このユーティリティーは、ご使用の IBM MQ MSCS サポートのセットアップについての詳細 (これは IBM サポートに連絡する場合に必要になる可能性があります) が示されるダイアログを表示します。 詳細をファイルに保管するためのオプションがあります。

Multi 複数インスタンス・キュー・マネージャー

複数インスタンス・キュー・マネージャーは、異なる複数のサーバー上で構成されている同じキュー・マネージャーのインスタンスです。 キュー・マネージャーの 1 つのインスタンスはアクティブ・インスタンスとして定義され、もう 1 つのインスタンスはスタンバイ・インスタンスとして定義されます。 アクティブ・インスタンスで障害が発生すると、複数インスタンス・キュー・マネージャーは、スタンバイ・サーバーで自動的に再始動します。

複数インスタンス・キュー・マネージャーの構成の例

505 ページの図 72 は、キュー・マネージャー QM1 の複数インスタンスの構成の例を示しています。 2 つのサーバーに IBM MQ がインストールされ、そのうちの 1 つは予備です。 1 つのキュー・マネージャー QM1 は作成済みです。 QM1 の 1 つのインスタンスはアクティブで、1 サーバー上で実行しています。 QM1 の他方のインスタンスは、他方のサーバー上でスタンバイとして実行しており、アクティブな処理は実行していませんが、QM1 のアクティブ・インスタンスに障害が起きた場合には、そのアクティブ・インスタンスを引き継ぐ準備ができています。 (複数インスタンス構成では、キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスは 1 つしか存在できません。)

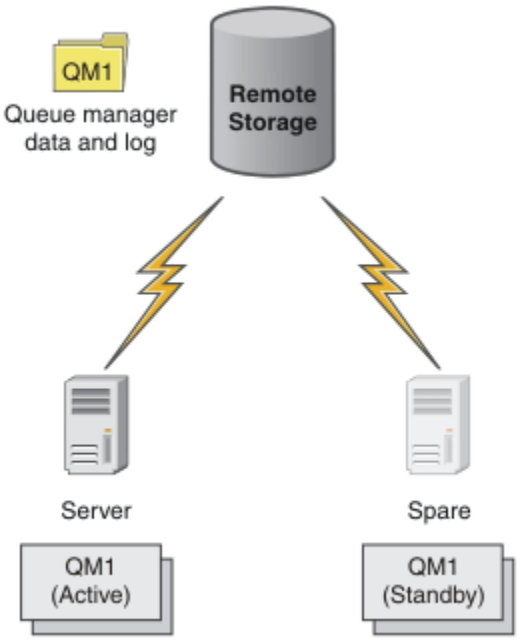


図 72. 複数インスタンス・キュー・マネージャー

キュー・マネージャーを複数インスタンス・キュー・マネージャーとして使用しようとしている場合は、**crtmqm** コマンドを使用して一方のサーバー上で単一のキュー・マネージャーを作成し、そのキュー・マネージャーのデータおよびログを共有ネットワーク・ストレージに配置します。 もう一方のサーバー上では、キュー・マネージャーを再び作成するのではなく、**addmqinf** コマンドを使用して、ネットワーク・ストレージ上のキュー・マネージャーのデータおよびログへの参照を作成します。

これで、キュー・マネージャーはどちらのサーバーからでも実行できるようになりました。 それぞれのサーバーは、同じキュー・マネージャーのデータおよびログを参照します。キュー・マネージャーは、1 つしかなく、一度に 1 サーバー上でのみアクティブになります。

キュー・マネージャーは、単一インスタンス・キュー・マネージャーまたは複数インスタンス・キュー・マネージャーのいずれかとして実行できます。 どちらの場合も、1 インスタンスのキュー・マネージャーだけが実行し、要求を処理します。 相違点となっているのは、複数インスタンス・キュー・マネージャーとして実行している場合には、キュー・マネージャーのアクティブ・インスタンスを実行していないサーバーはスタンバイ・インスタンスとして実行し、アクティブ・サーバーに障害が起きた場合に、アクティブ・インスタンスから自動的に引き継ぐ準備ができているということです。

どちらのインスタンスが最初にアクティブになるかを制御するのは、2 つのサーバーでキュー・マネージャーを開始する順序だけです。 キュー・マネージャーのデータに対する読み取り/書き込みロックを最初に獲得したインスタンスが、アクティブ・インスタンスになります。

もう一方のサーバーが開始しているのであれば、切り替えオプションを使用してアクティブ・インスタンスを停止し、制御をスタンバイに転送することで、アクティブ・インスタンスをそのもう一方のサーバーと交換できます。

QM1 のアクティブ・インスタンスは、実行時に共有キュー・マネージャーのデータおよびログ・フォルダーに対して排他的アクセスを行います。 アクティブ・インスタンスに障害が起こると、QM1 のスタンバイ・インスタンスがこれを検出し、アクティブ・インスタンスになります。 アクティブ・インスタンスが残したままの状態の QM1 のデータおよびログを引き継いで、クライアントおよびチャネルからの再接続を受け入れます。

アクティブ・インスタンスでは、以下のさまざまな理由から障害が起こる可能性があり、その結果としてスタンバイが引き継ぐことになります。

- アクティブ・キュー・マネージャー・インスタンスをホストするサーバーの障害。
- アクティブ・キュー・マネージャー・インスタンスをホストするサーバーとファイル・システムの間の接続障害。
- 無応答のキュー・マネージャー・プロセス。IBM MQ はこれを検出すると、キュー・マネージャーをシャットダウンします。

キュー・マネージャーの構成情報を複数のサーバーに追加し、アクティブ/スタンバイのペアとして実行する任意の 2 つのサーバーを選択することができます。 インスタンスの数は合計 2 つまでという制限があります。 2 つのスタンバイ・インスタンスと 1 つのアクティブ・インスタンスを設定することはできません。

高可用性ソリューションを構築するために必要な追加のコンポーネント

複数インスタンス・キュー・マネージャーは、高可用性ソリューションの一部です。 有用な高可用性ソリューションを構築するには、いくつかの追加のコンポーネントが必要です。

- アクティブ・キュー・マネージャー・インスタンスの実行を引き継ぐコンピューターに IBM MQ 接続を転送するための、クライアントおよびチャネルの再接続。
- ロックを正しく管理し、メディアおよびファイル・サーバーを障害から保護するハイパフォーマンス共有ネットワーク・ファイル・システム (NFS)。

  重要 : NFS ドライブの保守を実行するには、その前に環境内で実行されているすべての複数インスタンス・キュー・マネージャー・インスタンスを停止する必要があります。 NFS に障害が発生した場合に備えて、リカバリー用のキュー・マネージャー構成のバックアップを取っておいてください。
- 基本インフラストラクチャー内の Single Point of Failure を排除する、弾力性のあるネットワークおよび電源機構。
- フェイルオーバーを許容するアプリケーション。 特に、トランザクション・アプリケーションの動作、および IBM MQ キューを参照するアプリケーションには十分注意する必要があります。
- アクティブ・インスタンスとスタンバイ・インスタンスの実行を確認し、障害が起きたアクティブ・インスタンスを再始動するための、それらのインスタンスのモニターおよび管理。 複数インスタンス・キュー・マネージャーが自動的に再始動するとしても、スタンバイ・インスタンスが実行していて引き継ぎを行える状態であることと、障害が起きたインスタンスが新しいスタンバイ・インスタンスとしてオンラインに復帰できることを確認する必要があります。

IBM MQ MQI clients およびチャネルは、アクティブになったスタンバイ・キュー・マネージャーに自動的に再接続します。 再接続および高可用性ソリューションにおけるその他のコンポーネントについての詳細は、関連トピックにあります。 IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

サポートされているプラットフォーム

複数インスタンス・キュー・マネージャーは、z/OS 以外の任意のプラットフォームで作成できます。

MQI クライアントでは、自動クライアント再接続がサポートされています。

複数インスタンス・キュー・マネージャーの作成

1 つのサーバーでキュー・マネージャーを作成し、別のサーバーで IBM MQ を構成することで、複数インスタンス・キュー・マネージャーを作成します。 複数インスタンス・キュー・マネージャーは、キュー・マネージャーのデータとログを共有します。

複数インスタンス・キュー・マネージャーを作成する作業の大半は、キュー・マネージャーの共有のデータ・ファイルとログ・ファイルをセットアップする作業です。 ネットワーク・ストレージに共有ディレクトリーを作成し、ネットワーク共有の機能を使用してそれらのディレクトリーを他のサーバーからも利用できるようにする必要があります。 これらのタスクは、 AIX and Linux システムの *root* など、管理権限を持つユーザーが実行する必要があります。 手順は、次のとおりです。

1. データ・ファイルとログ・ファイルの共有フォルダーを作成します。
2. 1 つのサーバーでキュー・マネージャーを作成します。
3. 最初のサーバーで **dspmqinf** コマンドを実行し、キュー・マネージャーの構成データを収集して、クリップボードにコピーします。
4. コピーしたデータを対象にして **addmqinf** コマンドを実行し、第 2 のサーバーでキュー・マネージャーの構成を作成します。

第 2 のサーバーで **crtmqm** を実行して、キュー・マネージャーを再び作成することはしません。

ファイル・アクセス制御

他のすべてのサーバー上にあるユーザーおよびグループ mqm が、シェアへのアクセス許可を持っていることを確認する必要があります。

AIX and Linux では、すべてのシステム上で mqm の uid と gid を同じものにする必要があります。 このために、各システムの /etc/passwd を編集し、mqm に共通の uid および gid を設定して、ご使用のシステムをリブートすることが必要になる場合があります。

Microsoft Windows の場合、キュー・マネージャーのプロセスを実行するユーザー ID には、キュー・マネージャーのデータ・ファイルとログ・ファイルが含まれているディレクトリーに対するフル・コントロール権限が必要です。 この権限を構成するには、2 つの方法があります。

1. 代替セキュリティー・プリンシパルとしてグローバル・グループを使用してキュー・マネージャーを作成します。 そのグローバル・グループに、キュー・マネージャーのデータ・ファイルとログ・ファイルが含まれているディレクトリーに対するフル・コントロール権限を与えます。 536 ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 (Windows)』を参照してください。 キュー・マネージャーを実行するユーザー ID をそのグローバル・グループのメンバーにします。 ローカル・ユーザーをグローバル・グループのメンバーにすることはできないので、キュー・マネージャーのプロセスは、ドメイン・ユーザー ID で実行しなければなりません。 ドメイン・ユーザー ID は、ローカル・グループ mqm のメンバーでなければなりません。 このような方法で保護したファイルを使用する複数インスタンス・キュー・マネージャーをセットアップする方法については、510 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 (Windows)』の作業を参照してください。
2. ドメイン・コントローラー上にキュー・マネージャーを作成して、ローカル mqm グループにドメイン・スコープ "ドメイン・ローカル"を設定します。 ドメイン・ローカル mqm でファイル共有を保護し、キュー・マネージャーのすべてのインスタンスで、同じドメイン・ローカル mqm グループの下にあるキュー・マネージャー・プロセスを実行します。 このような方法で保護したファイルを使用する複数インスタンス・キュー・マネージャーをセットアップする方法については、526 ページの『Windows ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの作成』の作業を参照してください。

構成情報

それぞれのキュー・マネージャー・インスタンスを実行する各サーバーに関する IBM MQ キュー・マネージャー構成情報を変更することによって、必要な数だけキュー・マネージャー・インスタンスを構成できます。 各サーバーでは、同じバージョンの互換性のあるフィックス・レベルの IBM MQ をインストールしておく必要があります。 追加のキュー・マネージャー・インスタンスを構成するときには、**dspmqinf** コ

マンドと **addmqinf** コマンドが役立ちます。 あるいは、mqs.ini ファイルと qm.ini ファイルを直接編集することもできます。 549 ページの『Linux での複数インスタンス・キュー・マネージャーの作成』、510 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 (Windows)』、および 526 ページの『Windows ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの作成』のトピックは、複数インスタンス・キュー・マネージャーの構成方法を示す例です。

AIX, Linux, and Windows システムでは、単一の mqs.ini ファイルをネットワーク・シェアに置き、**AMQ_MQS_INI_LOCATION** 環境変数がそれを指すように設定することで、そのファイルを共有できます。

制約事項

1. 同じキュー・マネージャーの複数インスタンスは、同じオペレーティング・システム、アーキテクチャー、およびエンディアンを持つサーバー上でのみ構成してください。 例えば、両方のマシンを 32 ビットか 64 ビットで統一する必要があります。
2. すべての IBM MQ インストールは、リリース・レベル 7.0.1 以上でなければなりません。
3. 一般に、アクティブ・インストールとスタンバイ・インストールは、同じ保守レベルで保守されます。すべてのインストールをまとめてアップグレードする必要があるかどうかを確認するために、それぞれのアップグレードの保守の手順を参照してください。

   アクティブおよびパッシブのキュー・マネージャーの保守レベルは同一でなければならないことに注意してください。
4. キュー・マネージャーのデータとログを共有するのは、同じ IBM MQ ユーザー、同じグループ、同じアクセス制御メカニズムで構成したキュー・マネージャー同士の間に限定してください。 IBM i 例えば、Linux サーバー上にセットアップされたネットワーク・シェアには、AIX and Linux キュー・マネージャーのそれぞれのデータおよびログを入れることができますが、IBM i が使用するキュー・マネージャーのデータは入れることができません。

   IBM i IBM i システム用と AIX and Linux システム用の複数の共有場所を同じネットワーク・ストレージ上に作成できます。ただし、共有場所を別々にすることが必要です。 異なる所有者にそれぞれ異なるシェアを付与することができます。 この制限は、 AIX and Linux と IBM i の間で IBM MQ ユーザーおよびグループに使用される異なる名前の結果として生じます。 そのユーザーとグループの uid と gid が同じ場合もあり得ますが、この制約事項に変わりはありません。
5. AIX and Linux システム上では、ネットワーク・ストレージ上のファイル共有システムを、ソフト・マウントではなく、割り込み可能なハード・マウントで構成します。 割り込み可能なハード・マウントは、システム呼び出しにより中断されるまで、強制的にキュー・マネージャーを停止させます。 ソフト・マウントでは、サーバーの障害後のデータ整合性は保証されません。
6. この共有ログおよびデータ・ディレクトリーは、FAT または NFSv3 ファイル・システムには格納できません。 Windows の複数インスタンスのキュー・マネージャーの場合、Windows ネットワークによって使用される Common Internet File System (CIFS) プロトコルでネットワーク・ストレージにアクセスする必要があります。
7. z/OS z/OS は、複数インスタンス・キュー・マネージャーをサポートしません。 キュー共有グループを使用してください。

   再接続可能クライアントは、z/OS キュー・マネージャーと連携可能です。

Windows *Windows ドメインおよび複数インスタンス・キュー・マネージャー*

Windows で複数インスタンス・キュー・マネージャーを実行する場合は、データとログを共有する必要があります。 その共有フォルダーは、それぞれのサーバーやワークステーションで稼働するキュー・マネージャーのすべてのインスタンスからアクセスできる状態でなければなりません。 キュー・マネージャーと共有フォルダーを 1 つの Windows ドメインの中で構成してください。 キュー・マネージャーは、ドメイン・ワークステーションでも、ドメイン・サーバーでも、ドメイン・コントローラーでも実行できます。

重要 : デフォルトでは、Windows10 バージョン 1607 および WindowsServer 2016 で始まるコンピューターは、以前のバージョンの Windows よりも制限が厳しくなっています。

この変更により、セキュリティー・アカウント・マネージャー(SAM)へのリモート呼び出しを許可するクライアントが制限され、キュー・マネージャーによる IBM MQ の開始が失敗する可能性があります。 SAM へのアクセスは、IBM MQ がドメイン・アカウントとして構成されている場合に IBM MQ を機能させるために重要です。

複数インスタンス・キュー・マネージャーを構成する前に、539 ページの『Windows でキュー・マネージャーのデータとログの非共有ディレクトリーおよびファイルを保護する操作』と 536 ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 (Windows)』を読んで、キュー・マネージャーのデータ・ファイルとログ・ファイルへのアクセスを制御する方法を確認してください。 これらのトピックは学習用です。 Windows ドメインで複数インスタンス・キュー・マネージャーの共有ディレクトリーをセットアップする作業をすぐに始める場合は、510 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 (Windows)』を参照してください。

ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの実行

IBM WebSphere MQ 7.1 から、ドメインのメンバーであるワークステーションやサーバーで複数インスタンス・キュー・マネージャーを実行できるようになりました。 Windows で複数インスタンス・キュー・マネージャーを実行するには、ドメイン・コントローラーとファイル・サーバーのほかに、同じドメインに接続する同じキュー・マネージャーを実行するワークステーションまたはサーバーが 2 つ必要です。

ドメインに含まれているサーバーまたはワークステーションで複数インスタンス・キュー・マネージャーを実行できるように設計を変更したのは、追加のセキュリティー・グループを使用してキュー・マネージャーを作成できるようにするためです。 追加のセキュリティー・グループは、**crtmqm** コマンドの -a パラメーターで渡します。 その場合は、キュー・マネージャーのデータとログが含まれているディレクトリーをそのグループで保護します。 キュー・マネージャーのプロセスを実行するユーザー ID は、そのグループのメンバーでなければなりません。 キュー・マネージャーがそれらのディレクトリーにアクセスすると、それらのディレクトリーにアクセスする権限がユーザー ID にあるかどうかを Windows が検査します。 そのグループとユーザー ID の両方にドメイン・スコープを与えれば、キュー・マネージャーのプロセスを実行するユーザー ID は、そのグローバル・グループから資格情報を引き継ぐことになります。 キュー・マネージャーを別のサーバーで実行する場合でも、キュー・マネージャーのプロセスを実行するユーザー ID は、同じ資格情報を使用できます。 ユーザー ID 自体は、同じでなくてもかまいません。 いずれにしても、代替セキュリティー・グループとローカル mqm グループの両方のメンバーになっていることが必要です。

複数インスタンス・キュー・マネージャーの作成について詳しくは、 510 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 (Windows)』を参照してください。

ドメイン、ドメイン・サーバー、ドメイン・ワークステーションの構成には、複数の手順が必要になります。 Windows でデータ・ディレクトリーとログ・ディレクトリーにアクセスする権限をキュー・マネージャーに与える方法も理解しておかなければなりません。 ログ・ファイルとデータ・ファイルにアクセスする権限をキュー・マネージャーのプロセスに与える方法が分からない場合は、539 ページの『Windows でキュー・マネージャーのデータとログの非共有ディレクトリーおよびファイルを保護する操作』のトピックを参照してください。 このトピックには、必要な手順を理解するために役立つ 2 つの作業が含まれています。 その作業とは、541 ページの『ローカル mqm グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りおよび書き込み』と 545 ページの『代替ローカル・セキュリティー・グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りと書き込み』です。 さらに、536 ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 (Windows)』のトピックでは、キュー・マネージャーのデータ・ファイルとログ・ファイルが含まれている共有ディレクトリーを代替セキュリティー・グループで保護する方法が取り上げられています。 このトピックには、4 つの作業が含まれています。 つまり、Windows ドメインのセットアップ、ファイル共有の作成、IBM MQ for Windows のインストール、キュー・マネージャーでファイル共有を使用するための構成です。 それらの作業を以下に示します。

1. 513 ページの『Active Directory と DNS のドメインの作成 (Windows)』.
2. 517 ページの『Windows ドメイン内のサーバーまたはワークステーションへの IBM MQ のインストール』.

3. 520 ページの『キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 (Windows)』.
4. 523 ページの『代替グローバル・セキュリティー・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み』.

次に、ドメインを使用して、510 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 (Windows)』の作業を実行します。 実動ドメインで知識を活用する前に、これらの作業を実行して、複数インスタンス・キュー・マネージャーのセットアップについて検討してください。

ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの実行

キュー・マネージャーのデータは、ドメイン mqm グループで保護できました。 536 ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 (Windows)』のトピックで説明されているように、ワークステーションやサーバーでは、ローカル mqm グループで保護されているディレクトリーを共有できません。 一方、ドメイン・コントローラーでは、すべてのグループとプリンシパルのスコープがドメイン・スコープになります。 ドメイン・コントローラーに IBM MQ for Windows をインストールした場合は、キュー・マネージャーのデータ・ファイルとログ・ファイルがドメイン mqm グループで保護され、共有が可能になります。 ドメイン・コントローラーで複数インスタンス・キュー・マネージャーを構成する場合は、526 ページの『Windows ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの作成』の作業の手順を実行してください。

関連情報

Managing Authorization and Access Control
Windows サーバー・クラスター・ノードをドメイン・コントローラーとして使用する方法

Windows ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 *(Windows)*

Windows ドメインの一部であるワークステーションまたはサーバーの Windows 上で複数インスタンス・キュー・マネージャーをセットアップする方法の例を示します。 サーバーは、ドメイン・コントローラーである必要はありません。 ここで取り上げるのは、実動スケールのセットアップではなく、関連する概念を説明する程度のセットアップです。 この例は、Windows Server 2008 をベースとしています。 それぞれのステップは、Windows Server の他のバージョンでは異なる場合があります。

実動スケール構成では、構成を既存のドメインに合わせて調整しなければならない可能性があります。 例えば、さまざまなドメイン・グループを定義して、それぞれに異なる共有を許可したり、キュー・マネージャーを実行するユーザー ID をグループ化したりします。

この構成例では、3 つのサーバーを使用します。

sun
: Windows Server 2008 ドメイン・コントローラー。 *Sun*、*mars* および *venus* を含む *wmq.example.com* ドメインを所有しています。 例示目的で、これをファイル・サーバーとしても使用します。

mars
: 最初の IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの一つのインスタンスが入っています。

venus
: 2 番目の IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの二番目のインスタンスが含まれています。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

始める前に

Windows では、キュー・マネージャー・データとログ・ファイルを保管するファイル・システムを検証する必要はありません。 検査の手順、共有ファイル・システムの動作の検証は、AIX and Linux に適用されます。 Windows では検査は必ず成功します。

以下の作業手順を実行します。 これらのタスクは、ドメイン・コントローラーおよびドメインを作成し、サーバーに IBM MQ for Windows をインストールし、データおよびログ・ファイル用のファイル共用を作成します。既存のドメイン・コントローラーを構成する場合でも、新しい WindowsServer 2008 でこれらの手順を試してみる価値はあります。 それぞれのドメインに合わせてこれらの手順を調整できます。

1. 513 ページの『Active Directory と DNS のドメインの作成 (Windows)』.
2. 517 ページの『Windows ドメイン内のサーバーまたはワークステーションへの IBM MQ のインストール』.
3. 520 ページの『キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 (Windows)』.
4. 523 ページの『代替グローバル・セキュリティー・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み』.

このタスクについて

このタスクは、キュー・マネージャーのインスタンスを実行するように、ドメイン・コントローラーと 2 つのサーバーをドメイン内で構成するための、一連のタスクの 1 つです。 このタスクでは、二番目のサーバー *venus* を構成して、キュー・マネージャー *QMGR* の別のインスタンスを実行します。 キュー・マネージャー *QMGR* の 2 番目のインスタンスを作成し、機能するかどうかをテストするには、このタスクの手順に従います。

このタスクは、前のセクションの 4 つのタスクとは別のものです。 単一インスタンス・キュー・マネージャーを複数インスタンス・キュー・マネージャーに変換する手順が含まれています。 他のすべての手順は、単一インスタンス・キュー・マネージャーでも複数インスタンス・キュー・マネージャーでも共通です。

手順

1. IBM MQ for Windows を実行するように 2 番目のサーバーを構成します。
   a) タスク 517 ページの『Windows ドメイン内のサーバーまたはワークステーションへの IBM MQ のインストール』の手順を実行して、2 番目のドメイン・サーバーを作成します。 一連のタスクでは、二番目のサーバーは *venus* と呼ばれます。

   ヒント: 2 つのサーバーそれぞれにおいて、IBM MQ に同じインストール・デフォルトを使用して、2 番目のインストール済み環境を作成します。 デフォルトが異なる場合は、 IBM MQ 構成ファイル `mqs.ini` の *QMGR* **QueueManager** スタンザにある `Prefix` 変数と `InstallationName` 変数を調整する必要があります。 変数は、インストール済み環境ごと、および各サーバー上のキュー・マネージャーごとに異なる可能性のあるパスを参照します。 パスがすべてのサーバーで同一であると、より簡単に複数インスタンス・キュー・マネージャーを構成できます。
2. で *venus* の二番目 *QMGR* のインスタンスを作成します。
   a) *QMGR* *mars* が存在しない場合は、タスク 523 ページの『代替グローバル・セキュリティー・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み』を作成して作成します
   b) `Prefix` および `InstallationName` パラメーターの値が *venus* に対して正しいことを確認してください。

   mars で、**dspmqinf** コマンドを実行します。

   ```
   dspmqinf QMGR
   ```

   システム応答:

   ```
   QueueManager:
   Name=QMGR
   Directory=QMGR
   Prefix=C:\ProgramData\ IBM \ MQ
   ```

```
DataPath=\\sun\wmq\data\QMGR
InstallationName=Installation1
```

c) 機械可読形式の **QueueManager** スタンザをクリップボードにコピーします。

mars で、 -o command パラメーターを指定して **dspmqinf** コマンドを再度実行します。

```
dspmqinf -o command QMGR
```

システム応答:

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\ProgramData\ IBM \ MQ"
-v DataPath=\\sun\wmq\data\QMGR
```

d) *venus* で **addmqinf** コマンドを実行し、クリップボードの情報を使用して、キュー・マネージャーのインスタンスを *venus* で作成します。

必要に応じて、Prefix パラメーターまたは InstallationName パラメーターの違いに合わせてコマンドを調整します。

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\ProgramData\ IBM \ MQ"
-v DataPath=\\sun\wmq\data\QMGR
```

```
IBM MQ configuration information added.
```

3. *venus* 上のキュー・マネージャー *QMGR* を開始し、スタンバイ・インスタンスを許可します。

a) *QMGRmars* のチェック・マークが停止されている。

mars で、**dspmq** コマンドを実行します。

```
dspmq -m QMGR
```

システム応答は、キュー・マネージャーが停止した方法に応じて異なります。 以下に例を示します。

```
C:\Users\Administrator>dspmq -m QMGR
QMNAME(QMGR) STATUS(Ended immediately)
```

b) *venus* 上で、**strmqm** コマンドを実行し、スタンバイを許可して *QMGR* を開始します。

```
strmqm -x QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

タスクの結果

複数インスタンス・キュー・マネージャーが切り替わるかどうかをテストするには、以下の手順に従います。

1. *mars* 上で、**strmqm** コマンドを実行し、スタンバイを許可して *QMGR* を開始します。

   ```
   strmqm -x QMGR
   ```

   システム応答:

   ```
   IBM MQ queue manager 'QMGR' starting.
   The queue manager is associated with installation 'Installation1'.
   A standby instance of queue manager 'QMGR' has been started.
   The active instance is running elsewhere.
   ```

2. *venus* で、**endmqm** コマンドを実行します。

   ```
   endmqm -r -s -i QMGR
   ```

   venus 上のシステム応答:

   ```
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ended, permitting switchover to
   a standby instance.
   ```

   そして、で *mars*:

   ```
   dspmq
   QMNAME(QMGR) STATUS(Running as standby)
   C:\Users\wmquser2>dspmq
   QMNAME(QMGR) STATUS(Running as standby)
   C:\Users\wmquser2>dspmq
   QMNAME(QMGR)  STATUS(Running)
   ```

次のタスク

サンプル・プログラムを使用して複数インスタンス・キュー・マネージャーを検証するには、534 ページの『Windows での複数インスタンス・キュー・マネージャーの検査』を参照してください。

Windows *Active Directory と DNS のドメインの作成 (Windows)*

このタスクは、*sun* という Windows2008 ドメイン・コントローラー上でドメイン *wmq.example.com* を作成します。 さらに、そのドメインで Domain mqm グローバル・グループを構成し、適切な権限と 1 人のユーザーを設定します。

実動スケール構成では、構成を既存のドメインに合わせて調整しなければならない可能性があります。 例えば、さまざまなドメイン・グループを定義して、それぞれに異なる共有を許可したり、キュー・マネージャーを実行するユーザー ID をグループ化したりします。

この構成例では、3 つのサーバーを使用します。

sun
: Windows Server 2008 ドメイン・コントローラー。 *Sun*、*mars* および *venus* を含む *wmq.example.com* ドメインを所有しています。 例示目的で、これをファイル・サーバーとしても使用します。

mars
: 最初の IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの一つのインスタンスが入っています。

venus
2 番目の IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの二番目のインスタンスが含まれています。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

始める前に

1. タスク・ステップは、インストールされているが役割を使用して構成されていない WindowsServer 2008 と整合しています。 既存のドメイン・コントローラーを構成する場合でも、新しい WindowsServer 2008 でこれらの手順を試してみる価値はあります。 それぞれのドメインに合わせてこれらの手順を調整できます。

このタスクについて

この作業では、新しいドメイン・コントローラーで Active Directory と DNS のドメインを作成します。 その後、そのドメインを構成し、そのドメインに加わる他のサーバーやワークステーションに IBM MQ をインストールする準備を整えます。 Active Directory のインストールと構成を実行して Windows ドメインを作成する作業に慣れていない場合でも、以下の手順を実行してみてください。 複数インスタンス・キュー・マネージャーの構成を作成するには、Windows ドメインを作成することが必要です。 ここで取り上げる作業手順が Windows ドメインを構成するための最適な方法であるというわけではありません。 実稼働環境で複数インスタンス・キュー・マネージャーをデプロイする場合は、Windows の資料を参照してください。

この作業では、以下の流れで手順を実行します。

1. Active Directory をインストールします。
2. ドメインを追加します。
3. ドメインを DNS に追加します。
4. グローバル・グループ `Domain mqm` を作成して、適切な権限を与えます。
5. 1 人のユーザーを追加して、そのユーザーをグローバル・グループ `Domain mqm` のメンバーにします。

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。 これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。 さらに、508 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』の作業も必要です。

このタスクでは、ドメインコントローラーのホスト名は *sun* です。2 つの IBM MQ サーバーを *mars* と *venus* 呼ぶことにする。 ドメインは *wmq.example.com* と呼ばれます。 この作業で斜体で表記している名前はすべて、それぞれの好みの名前に置き換えてもかまいません。

手順

1. ローカル管理者または `Workgroup` 管理者として、ドメイン・コントローラー *sun* にログオンします。

   サーバーが既にドメイン・コントローラーとして構成されている場合は、ドメイン管理者としてログオンする必要があります。
2. 「Active Directory ドメイン サービス」ウィザードを実行します。

   a) クリック**スタート** > **実行...** `dcpromo` と入力して、 「**OK**」をクリックします。

   Active Directory のバイナリー・ファイルがインストールされていなければ、Windows によってそれらのファイルが自動的にインストールされます。
3. ウィザードの最初のウィンドウで、「**詳細モード インストールを使用する**」チェック・ボックスは、クリアされた状態のままにしておきます。 「**次へ**」 > 「**次へ**」をクリックし、「**新しいフォレストに新しいドメインを作成する**」 > 「**次へ**」をクリックします。
4. 「**フォレストルートドメインの FQDN**」 フィールドに *wmq.example.com* と入力します。 **次へ** をクリックします。

5. 「フォレスト機能レベルの設定」ウィンドウで、「**フォレスト機能レベル**」 > 「**次へ**」のリストから「**Windows Server 2003**」以降を選択します。

   IBM MQ でサポートされている最も古いレベルの Windows サーバーは、WindowsServer 2003 です。
6. オプション:「ドメイン機能レベルの設定」ウィンドウで、「**ドメイン機能レベル**」 > 「**次へ**」のリストから「 **Windows Server 2003** 以降」を選択します。

   このステップは、フォレストの機能レベルを「**Windows Server 2003**」に設定した場合にのみ必要です。
7. 「追加のドメイン コントローラ オプション」ウィンドウが開きます。 追加のオプションとして「**DNS サーバー**」が選択された状態になります。 「**次へ**」をクリックし、「**はい**」をクリックして警告ウィンドウを消去します。

   ヒント: DNS サーバーが既にインストールされていると、このオプションは表示されません。 この作業を厳密に実行したいと思う場合は、このドメイン・コントローラーからすべての役割を削除してから、再び作業を開始してください。
8. から移動 `DatabaseLog Files` と `SYSVOL` ディレクトリは変更せずに、**次**をクリックする。
9. 「ディレクトリー サービス復元モード Administrator パスワード」ウィンドウの「**パスワード**」フィールドと「**パスワードの確認入力**」フィールドにパスワードを入力します。 「**次へ**」 > 「**次へ**」をクリックします。 ウィザードの最後のウィンドウで「**完了時に再起動する**」を選択します。
10. ドメイン・コントローラーがリブートされたら、*wmq*\`Adminstrator` としてログオンします。

    サーバー・マネージャーが自動的に始動します。
11. *wmq.example.com*\`Users` フォルダーを開きます

    a) **[サーバーマネージャー]** > **[ロール]** > **Active Directory Domain Services** > ***wmq.example.com*** > **[ユーザー]**を開きます。
12. 「**ユーザー**」 > **新規** > 「**グループ**」を右クリックします。

    a)「**グループ名**」フィールドにグループ名を入力します。

    注: 優先的に使用するべきグループ名は、`Domain mqm` です。 このとおりに入力してください。

    - このグループに `Domain mqm` という名前を付けると、ドメイン・ワークステーションまたはドメイン・サーバーでの Prepare IBM MQ Wizard の動作が変更されます。 これにより、 Prepare IBM MQ Wizard は、ドメイン内の IBM MQ の新規インストールのたびに、グループ `Domain mqm` をローカル `mqm` グループに自動的に追加します。
    - `Domain mqm` グローバル・グループを持たないドメインにワークステーションまたはサーバーをインストールすることができます。 これを行う場合は、`Domain mqm` グループと同じプロパティーを持つグループを定義する必要があります。 IBM MQ がドメインにインストールされている場合は、そのグループ、またはそのグループのメンバーであるユーザーを、ローカル `mqm` グループのメンバーにする必要があります。 ドメイン・ユーザーを複数のグループに組み込むことも可能です。 複数のドメイン・グループを作成し、別々に管理するインストール環境のセットごとに、それぞれのグループを対応させます。 管理対象のインストール環境に基づいて、ドメイン・ユーザーをそれぞれのドメイン・グループに振り分けます。 各ドメイン・グループまたはグループを、異なる IBM MQ インストール済み環境のローカル `mqm` グループに追加します。 特定のローカル `mqm` グループのメンバーであるドメイン・グループ内のドメイン・ユーザーのみが、そのインストールのキュー・マネージャーを作成、管理、および実行することができます。
    - ドメインに含まれているワークステーションまたはサーバーに IBM MQ をインストールするときに指名するドメイン・ユーザーは、`Domain mqm` グループのメンバーであるか、`Domain mqm` グループと同じプロパティーで定義した代替グループのメンバーでなければなりません。

    b)「**グループのスコープ**」では、「**グローバル**」がクリックされたままの状態にしておくか、設定を「**ユニバーサル**」に変更します。 「**グループの種類**」では、「**セキュリティ**」がクリックされたままの状態にしておきます。 「**OK**」をクリックします。
13. 「**グループのメンバーシップの読み取り**」の「**許可**」と「**groupMembershipSAM の読み取り**」の「**許可**」の権限を `Domain mqm` グローバル・グループに追加します。

    a) サーバー・マネージャーのアクション・バーで、「**表示**」 > 「**高度な機能**」をクリックします

b) サーバー・マネージャーのナビゲーション・ツリーで、「**ユーザー**」をクリックします。

c)「Users (ユーザー)」ウィンドウで「**Domain mqm (ドメイン mqm)**」 > 「**Properties (プロパティー)**」を右クリックします。

d)「**セキュリティ**」 > 「**詳細設定**」 > 「**追加...**」をクリックします。 Domain mqm と入力し、「**名前の確認**」 > 「**OK**」をクリックします。

「**名前**」 フィールドには、ストリング Domain mqm (*domain name*\Domain mqm)が事前に入力されています。

e)「**プロパティー**」をクリックします。 「**適用先**」リストの「**下位のユーザー オブジェクト**」を選択します。

f)「**アクセス許可**」リストから、「**グループのメンバーシップの読み取り**」と「**groupMembershipSAM の読み取り**」の「**許可**」チェック・ボックスを選択し、「**OK**」 > 「**適用**」 > 「**OK**」 > 「**OK**」をクリックします。

14. Domain mqm グローバル・グループに 2 人以上のユーザーを追加します。

1 人のユーザー (この例では *wmquser1*) が IBM MQ サービスを実行し、もう 1 人のユーザー (*wmquser2*) が対話式に使用されます。

ドメイン構成で代替セキュリティー・グループを使用するキュー・マネージャーを作成するには、ドメイン・ユーザーが必要です。 管理者には **crtmqm** コマンドを実行する権限があるものの、管理者権限を持ったユーザー ID だけでは十分ではありません。 このドメイン・ユーザーは、管理者である場合もありますが、いずれにしてもローカル mqm グループと代替セキュリティー・グループの両方のメンバーになっていなければなりません。

この例では、*wmquser1* と *wmquser2* を Domain mqm グローバル・グループのメンバーにします。 Prepare IBM MQ Wizard を実行すると、ウィザードの実行場所がどこであっても、Domain mqm がローカル mqm グループのメンバーとして自動的に構成されます。

1 台のコンピューターに存在する IBM MQ のインストール環境ごとに、IBM MQ サービスを実行するユーザーを別々に設定する必要があります。 別々のコンピューターで同じユーザーを再利用することは可能です。

a) サーバー・マネージャーのナビゲーション・ツリーで、「**ユーザー**」 > 「**新規作成**」 > 「**ユーザー**」をクリックします。

b)「新規オブジェクト-ユーザー」ウィンドウで、 「**ユーザー・ログオン名**」 フィールドに *wmquser1* と入力します。 「**名**」 フィールドに *WebSphere* と入力し、 「**姓**」 フィールドに *MQ1* と入力します。 「**次へ**」をクリックします。

c)「**パスワード**」フィールドと「**パスワードの確認入力**」フィールドにパスワードを入力し、「**ユーザーは次回ログオン時にパスワードの変更が必要**」チェック・ボックスをクリアします。 「**次へ**」 > 「**完了**」をクリックします。

d)「ユーザー」ウィンドウで、***WebSphere MQ*** >「**グループに追加 ...**」を右クリックします。 Domain mqm と入力し、「**名前の確認**」 > 「**OK**」 > 「**OK**」をクリックします。

e) ステップ a から d を繰り返して、 *WebSphere MQ2* を *wmquser2* として追加します。

15. サービスとして IBM MQ を実行します。

IBM MQ をサービスとして実行し、サービスとして実行する権限をドメイン・ユーザー (ドメイン管理者から取得した) に付与する必要がある場合、以下の手順を実行します。

a)「**スタート**」 **>** 「**実行...**」をクリックします。

コマンド secpol.msc を入力して「**OK**」をクリックします。

b)「**セキュリティの設定**」 **>** 「**ローカル ポリシー**」 **>** 「**ユーザー権利の割り当て**」を開きます。

ポリシーのリストで、「**サービスとしてログオン**」 **>** 「**プロパティー**」を右クリックします。

c)「**ユーザーまたはグループの追加...**」をクリックします。

ドメイン管理者から取得したユーザーの名前を入力し、「**名前の確認**」をクリックします。

d)「Windows セキュリティ」ウィンドウが表示されたら、十分な権限を持つアカウント・ユーザーまたは管理者のユーザー名とパスワードを入力し、「**OK**」 **>** 「**適用**」 **>** 「**OK**」をクリックします。

「ローカル セキュリティ ポリシー」ウィンドウを閉じます。

注 : Windows Server Server 2008 および Windows Server 2012 では、ユーザー・アカウント制御 (UAC) 機能はデフォルトで有効です。

UAC 機能は、ユーザーが Administrators グループのメンバーである場合でも、特定のオペレーティング・システム機能に対してユーザーが実行できる操作を制限します。 この制限に対応するため、適切な手段を講じる必要があります。

次のタスク

次に、517 ページの『Windows ドメイン内のサーバーまたはワークステーションへの IBM MQ のインストール』の作業に進みます。

関連タスク

Windows Windows ドメイン内のサーバーまたはワークステーションへの IBM MQ のインストール

Windows キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 (Windows)

Windows 代替グローバル・セキュリティー・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み

Windows *Windows ドメイン内のサーバーまたはワークステーションへの IBM MQ のインストール*

このタスクでは、*wmq.example.com*Windows ドメイン内のサーバーまたはワークステーションにインストールし、IBM MQ 設定します。

実動スケール構成では、構成を既存のドメインに合わせて調整しなければならない可能性があります。 例えば、さまざまなドメイン・グループを定義して、それぞれに異なる共有を許可したり、キュー・マネージャーを実行するユーザー ID をグループ化したりします。

この構成例では、3 つのサーバーを使用します。

sun
Windows Server 2008 ドメイン・コントローラー。 *Sun*、*mars* および *venus* を含む *wmq.example.com* ドメインを所有しています。 例示目的で、これをファイル・サーバーとしても使用します。

mars
最初の IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの一つのインスタンスが入っています。

venus
2 番目の IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの二番目のインスタンスが含まれています。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

始める前に

重要 : デフォルトでは、Windows10 バージョン 1607 および WindowsServer 2016 で始まるコンピューターは、以前のバージョンの Windows よりも制限が厳しくなっています。

この変更により、セキュリティー・アカウント・マネージャー(SAM)へのリモート呼び出しを許可するクライアントが制限され、キュー・マネージャーによる IBM MQ の開始が失敗する可能性があります。 SAM へのアクセスは、IBM MQ がドメイン・アカウントとして構成されている場合に IBM MQ を機能させるために重要です。

1. 513 ページの『Active Directory と DNS のドメインの作成 (Windows)』のステップを実行して、ドメイン *wmq.example.com* 用のドメイン・コントローラー *sun* を作成します。 斜体の名前は、それぞれの構成に合わせて変更してください。
2. IBM MQ を実行できるその他の Windows バージョンについては、 Windows システムでのハードウェア要件およびソフトウェア要件 を参照してください。

このタスクについて

このタスクでは、*mars* という名前の Windows サーバー 2008 を、*wmq.example.com* ドメインのメンバーとして構成します。 IBM MQ をインストールし、*wmq.example.com* ドメインのメンバーとして実行するように設定します。

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。 これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。 さらに、508 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』の作業も必要です。

このタスクでは、ドメインコントローラーのホスト名は *sun* です。2 つの IBM MQ サーバーを *mars* と *venus* 呼ぶことにする。 ドメインは *wmq.example.com* と呼ばれます。 この作業で斜体で表記している名前はすべて、それぞれの好みの名前に置き換えてもかまいません。

手順

1. ドメイン・コントローラー *sun.wmq.example.com*to*mars* を DNS サーバーとして追加します。
   a) *mars* で、*mars*\Administrator としてログオンし、**スタート**をクリックします。
   b)「**ネットワーク**」 >「**プロパティー**」 >「**ネットワーク接続の管理**」を右クリックします。
   c) ネットワーク・アダプターを右クリックして、「**プロパティ**」をクリックします。

      システムの応答として、「ローカル エリア接続のプロパティ」ウィンドウが開き、接続で使用する項目のリストが表示されます。
   d)「ローカル エリア接続のプロパティ」ウィンドウの項目リストから、「**インターネット プロトコル バージョン 4**」または「**インターネット プロトコル IBM WebSphere MQ 6**」を選択します。 「**プロパティー**」 >「**拡張 ...**」 をクリックします。 「**DNS**」 タブをクリックします。
   e) DNS サーバー・アドレスで、「**追加...**」をクリックします。
   f) DNS サーバーでもあるドメイン・コントローラーの IP アドレスを入力して、「**追加**」をクリックします。
   g)「**以下の DNS サフィックスを順に追加する**」 >「**追加...**」をクリックします。
   h) タイプ *wmq.example.com* と**追加**をクリックします。
   i) **この接続の DNS サフィックス**フィールドでのタイプ *wmq.example.com*。
   j)「**この接続のアドレスを DNS に登録する**」と「**この接続の DNS サフィックスを DNS 登録に使う**」を選択します。 「**OK**」 >「**OK**」 >「**閉じる**」をクリックします。
   k) コマンド・ウィンドウで、**ipconfig /all** というコマンドを入力して、TCP/IP 設定を確認します。
2. *mars* で、コンピューターを *wmq.example.com* ドメインに追加します。
   a)「**スタート**」をクリックします。
   b) **コンピュータ > プロパティ**を右クリックします。 「コンピュータ名、ドメインおよびワークグループの設定」のセクションで、「**設定の変更**」をクリックします。
   c) システム・プロパティー・ウィンドウで、「**変更...**」をクリックします。
   d)「ドメイン」をクリックし、「*wmq.example.com*」を入力して、**OK** をクリックします。
   e) ドメイン・コントローラーの管理者 (このコンピューターをドメインに加わらせる権限を持った管理者) の**ユーザー名**と**パスワード** を入力して、「**OK**」をクリックします。
   f) "*wmq.example.com* ドメインへようこそ"メッセージに応答して、**OK > OK > クローズ > 即時再始動**をクリックします。
3. コンピューターが *wmq.example.com* ドメインのメンバーであることを確認します
   a) *sun* で、ドメイン・コントローラーに *wmq*\Administrator としてログオンします。
   b) **サーバー・マネージャー > Active Directory ドメイン・サービス > *wmq.example.com* > コンピューターズ**を開き、ウィンドウに *mars* が正しくリストされていることを確認します。

4. に IBM MQ for Windows インストールします *mars*。

   IBM MQ for Windows インストール・ウィザードの実行について、詳しくは、Windows への IBM MQ サーバーのインストールを参照してください。

   a) *mars* に、ローカル管理者 *mars*\Administrator としてログオンします。

   b) **Setup** コマンドを IBM MQ for Windows インストール・メディアで実行します。

      IBM MQ のランチパッド・アプリケーションが始動します。

   c)「**ソフトウェア要件**」をクリックして、前提ソフトウェアがインストールされていることを確認します。

   d)「**ネットワーク構成**」 > 「**はい**」をクリックして、ドメイン・ユーザー ID を構成します。

      513 ページの『Active Directory と DNS のドメインの作成 (Windows)』の作業によって、この一連の作業のドメイン・ユーザー ID を構成します。

   e)「**IBM MQ インストール**」をクリックし、インストール言語を選択し、「IBM MQ インストーラーの起動」をクリックします。

   f) 使用条件を確認し、「**次へ**」 > 「**次へ**」 > 「**インストール**」をクリックして、デフォルト構成を受け入れます。 インストールが完了するのを待って、「**完了**」をクリックします。

      インストール環境の名前を変更したり、別のコンポーネントをインストールしたり、キュー・マネージャーのデータとログのために別のディレクトリーを構成したり、別のディレクトリーにインストールしたりする場合もあります。 その場合は、「**標準**」ではなく「**カスタム**」をクリックします。

      IBM MQ がインストールされ、インストーラーによって Prepare IBM MQ Wizard が開始されます。

      重要 : ウィザードはまだ実行しないでください。

5. IBM MQ サービスを実行するユーザーを構成して、「**サービスとして実行**」の権限を与えます。

   ローカル mqm グループ、 Domain mqm グループ、または権限を使用して IBM MQ サービスを実行するユーザーのいずれを構成するかを選択します。 この例では、ユーザーにその権限を与えます。

   a)「**スタート**」 > 「**ファイル名を指定して実行 ...**」をクリックします。 コマンド **secpol.msc** を入力し、 「**OK**」をクリックします。

   b)「**セキュリティの設定**」 > 「**ローカル ポリシー**」 > 「**ユーザー権利の割り当て**」を開きます。 ポリシーの中で、**サービスとしてログオン > プロパティ**を右クリックします。

   c)「**ユーザーまたはグループの追加...**」をクリックします。 *wmquser1* と入力し、 「**名前の確認**」 をクリックします。

   d) ドメイン管理者 *wmq*\Administrator のユーザー名とパスワードを入力し、 「**OK**」 > 「**適用**」 > 「**OK**」をクリックします。 「ローカル セキュリティ ポリシー」ウィンドウを閉じます。

6. Prepare IBM MQ Wizard を実行します。

   詳しくは、 Prepare IBM MQ Wizard を使用した IBM MQ の構成を参照してください。

   a) IBM MQ インストーラーによって Prepare IBM MQ Wizard が自動的に実行されます。

      ウィザードを手動で開始するには、 「**スタート**」 > 「**すべてのプログラム**」 > **IBM MQ** フォルダーで Prepare IBM MQ Wizard へのショートカットを見つけます。 複数インストール構成の場合は、IBM MQ のインストールに対応するショートカットを選択します。

   b)「**次**」をクリックし、"ネットワーク内に Windows 2000 またはそれ以降のドメイン・コントローラーがあるかどうかを判別する"ために、応答として、**Yes** をクリックします。

   c) 最初の「 IBM MQ for Windows for Windows ドメイン・ユーザーの構成」ウィンドウで 「**はい**」 > 「**次へ**」 をクリックします。

   d) 2 番目の「 IBM MQ for Windows for Windows ドメイン・ユーザーの構成」ウィンドウで、 「**ドメイン**」 フィールドに *wmq* と入力します。 「**ユーザー名**」 フィールドに *wmquser1* と入力し、 「**パスワード**」 フィールドにパスワード (設定した場合) を入力します。 「**次へ**」をクリックします。

      ウィザードによって、IBM MQ が構成され、*wmquser1* で開始されます。

e) ウィザードの最終ページで、必要に応じて各種のチェック・ボックスを選択したりクリアしたりして、「**完了**」をクリックします。

次のタスク

1. 541 ページの『ローカル mqm グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りおよび書き込み』の作業を実行して、インストールと構成が正しく機能していることを検証します。
2. 520 ページの『キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 (Windows)』の作業を実行して、複数インスタンス・キュー・マネージャーのデータ・ファイルとログ・ファイルのファイル共有を構成します。

関連タスク

Windows Active Directory と DNS のドメインの作成 (Windows)

Windows キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 (Windows)

Windows 代替グローバル・セキュリティー・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み

関連資料

IBM MQ Windows サービスに必要なユーザー権限

Windows キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 *(Windows)*

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。 これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。

実動スケール構成では、構成を既存のドメインに合わせて調整しなければならない可能性があります。 例えば、さまざまなドメイン・グループを定義して、それぞれに異なる共有を許可したり、キュー・マネージャーを実行するユーザー ID をグループ化したりします。

この構成例では、3 つのサーバーを使用します。

sun
: Windows Server 2008 ドメイン・コントローラー。 *Sun*、*mars* および *venus* を含む *wmq.example.com* ドメインを所有しています。 例示目的で、これをファイル・サーバーとしても使用します。

mars
: 最初の IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの一つのインスタンスが入っています。

venus
: 2 番目の IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの二番目のインスタンスが含まれています。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

始める前に

1. 文書化されているとおりにこのタスクを実行するには、513 ページの『Active Directory と DNS のドメインの作成 (Windows)』、タスクの手順を実行して、ドメインコントローラー *sun* の上にドメイン *sun.wmq.example.com* を作成します。 斜体の名前は、それぞれの構成に合わせて変更してください。

このタスクについて

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。 これらの作業によって、任意のディレクトリーに格納されているデータ・

ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。 さらに、508 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』の作業も必要です。

この作業では、データとログのディレクトリーが含まれている共有フォルダーと、グローバル・グループを作成し、その共有フォルダーに対するアクセス権限をそのグローバル・グループに与えます。 **crtmqm** コマンドでは、その共有フォルダーに対するアクセス権限を与えるグローバル・グループの名前を -a パラメーターで渡します。 このグローバル・グループを使用すれば、この共有フォルダーのユーザーと他の共有フォルダーのユーザーを柔軟に分離できるようになります。 そのような柔軟性が不要な場合は、新しいグローバル・グループを作成する代わりに、その共有フォルダーに対する権限を Domain mqm グループに与えます。

このタスクで共有するために使用されるグローバル・グループは *wmqha* と呼ばれ、共用は *wmq* と呼ばれます。 これらは Windows ドメイン *wmq.example.com* のドメイン・コントローラー *sun* で定義されます。 この共有には、グローバル・グループ *wmqha* に対する完全な制御権限があります。 この作業で斜体で表記している名前をそれぞれの好みの名前に置き換えてください。

この作業では、ドメイン・コントローラーとして、ファイル・サーバーと同じサーバーを使用します。 実際のアプリケーション環境では、パフォーマンスと可用性の観点から、ディレクトリー・サービスとファイル・サービスを別々のサーバーに分離するようにしてください。

キュー・マネージャーを実行するユーザー ID は、2 つのグループのメンバーとして構成する必要があります。 これは、 IBM MQ サーバー上のローカル mqm グループ、および *wmqha* グローバル・グループのメンバーでなければなりません。

タスク・セットでは、キュー・マネージャーがサービスとして実行されている場合は、ユーザー ID*wmquser1* の下で実行されるため、*wmquser1* は *wmqha* のメンバーでなければなりません。 キュー・マネージャーが対話式に実行されている場合は、ユーザー ID*wmquser2* の下で実行されるため、*wmquser2* は *wmqha* のメンバーでなければなりません。 *wmquser1* も *wmquser2* も、グローバル・グループ Domain mqm のメンバーです。 Domain mqm は、 *mars* サーバーおよび *venus* IBM MQ サーバー上のローカル mqm グループのメンバーです。 したがって、*wmquser1* と *wmquser2* は、両方の IBM MQ サーバー上のローカル mqm グループのメンバーです。

手順

1. *sun.wmq.example.com* ドメイン管理者として、ドメインコントローラーにログオンします。
2. グローバル・グループ *wmqha* を作成します。
   a) **[サーバーマネージャー] > [ロール] > Active Directory Domain Services > *wmq.example.com* > [ユーザー]**を開きます。
   b) *wmq.example.com*\Users フォルダーを開きます
   c)「**ユーザー**」 > **新規** > 「**グループ**」を右クリックします。
   d)「**グループ名**」 フィールドに *wmqha* と入力します。
   e)「**グループのスコープ**」では、「**グローバル**」がクリックされたままの状態にしておき、「**グループの種類**」では、「**セキュリティ**」がクリックされたままの状態にしておきます。 「**OK**」 をクリックします。
3. ドメイン・ユーザー *wmquser1* および *wmquser2* をグローバル・グループ *wmqha* に追加します。
   a) サーバー・マネージャーのナビゲーション・ツリーで、**利用者**をクリックし、ユーザーのリストで ***wmqha*** **> プロパティ**を右クリックします。
   b) *wmqha* プロパティーウィンドウの「メンバー」タブをクリックします。
   c)「**追加 ...**」 をクリックします。 *wmquser1* ; *wmquser2* と入力し、 「**名前の確認**」 > 「**OK**」 > 「**適用**」 > 「**OK**」をクリックします。
4. キュー・マネージャーのデータ・ファイルとログ・ファイルを格納するディレクトリー・ツリーを作成します。
   a) コマンド・プロンプトを開きます。

b) 次のコマンドを入力します。

```
md c:\wmq\data, c:\wmq\logs
```

5. グローバル・グループ *wmqha* に対して、*c:\wmq* ディレクトリーおよび共有に対する全制御権限を付与する権限を付与します。
   a) Windows エクスプローラーで、 ***c:¥wmq*** > 「**プロパティー**」を右クリックします。
   b)「**セキュリティ**」タブをクリックし、「**詳細設定**」 > 「**編集...**」をクリックします。
   c)「**このオブジェクトの親からの継承可能なアクセス許可を含める**」チェック・ボックスをクリアします。 「Windows セキュリティー」ウィンドウで 「**コピー**」 をクリックします。
   d)「**アクセス許可エントリ**」のリストで「Users」の各行を選択し、「**削除**」をクリックします。 「**アクセス許可エントリ**」のリストにある「SYSTEM」、「Administrators」、「CREATOR OWNER」の各行はそのままにしておきます。
   e)「**追加 ...**」をクリックします。 グローバル・グループ *wmqha* の名前を入力します。 「**名前の確認**」 > 「**OK**」をクリックします。
   f)「wmq の許可エントリー」ウィンドウで、**アクセス権**のリストで「 **完全制御** 」を選択します。
   g)「**OK**」 > 「**適用**」 > 「**OK**」 > 「**OK**」 > 「**OK**」をクリックします。
   h) Windows エクスプローラーで、 ***c:¥wmq*** > 「**共有 ...**」を右クリックします。
   i)「**拡張共有 ...**」 をクリックします。 次に、 「**このフォルダーを共有する**」 チェック・ボックスを選択します。 共用名は *wmq* のままにします。
   j) **許可** > 「**追加 ...**」をクリックします。 グローバル・グループ *wmqha* の名前を入力します。 「**名前の確認**」 > 「**OK**」をクリックします。
   k) **グループ名またはユーザー名**のリストで「*wmqha*」を選択します。 ***wmqha* の許可** のリストにある「**フルコントロール**」 チェック・ボックスを選択します。 「**適用**」をクリックします
   l) **グループ名またはユーザー名**のリストで「*Administrators*」を選択します。 「 **管理者の権限**」 のリストで 「**フルコントロール**」 チェック・ボックスを選択します。 「**適用**」 > 「**OK**」 > 「**OK**」 > 「**閉じる**」をクリックします。

次のタスク

それぞれの IBM MQ サーバーから、共有ディレクトリーのファイルを読み取ったり、共有ディレクトリーにファイルを書き込んだりする操作を実行できることを確認します。 IBM MQ サービスのユーザー ID *wmquser1* と対話式ユーザー ID *wmquser2* を確認します。

1. リモート・デスクトップを使用している場合は、 *mars* 上のローカル・グループ Remote Desktop Users に *wmq**wmquser1* および *wmquser2* を追加する必要があります。
   a. *wmq*\Administrator として *mars* にログオンします
   b. **lusrmgr.msc** コマンドを実行して、「ローカル ユーザーとグループ」ウィンドウを開きます。
   c.「**グループ**」をクリックします。 「**リモート・デスクトップ・ユーザー**」 > 「**プロパティー**」 > 「**追加 ...**」を右クリックします。 *wmquser1 ; wmquser2* と入力し、 「**名前の確認**」をクリックします。
   d. ドメイン管理者 *wmq*\Administrator のユーザー名とパスワードを入力し、 「**OK**」 > 「**適用**」 > 「**OK**」をクリックします。
   e.「ローカル ユーザーとグループ」ウィンドウを閉じます。
2. *wmq**wmquser1* として *mars* にログオンします。
   a. Windows エクスプローラーのウィンドウを開き、*sun**wmq* を入力します。

      システムは *sun.wmq.example.com* で *wmq* 共有を開くことで応答し、データとログのディレクトリを一覧表示します。
   b. *wmquser1* の許可を確認するには、データ・サブディレクトリー内にファイルを作成し、いくつかのコンテンツを追加し、それを読み取ってから削除します。

3. *wmq\wmquser2* として *mars* にログオンし、検査を繰り返します。
4. 次に、データとログの共有ディレクトリーを使用するキュー・マネージャーを作成する作業を実行します。 523 ページの『代替グローバル・セキュリティー・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み』を参照してください。

関連タスク

Windows Active Directory と DNS のドメインの作成 (Windows)

Windows Windows ドメイン内のサーバーまたはワークステーションへの IBM MQ のインストール

Windows 代替グローバル・セキュリティー・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み

Windows 代替グローバル・セキュリティー・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み

この作業では、**crtmqm** コマンドで -a フラグを使用する方法を確認します。 -a フラグを使用すると、キュー・マネージャーは、代替セキュリティー・グループを使用して、リモート・ファイル共有フォルダーにあるログ・ファイルとデータ・ファイルにアクセスできるようになります。

実動スケール構成では、構成を既存のドメインに合わせて調整しなければならない可能性があります。 例えば、さまざまなドメイン・グループを定義して、それぞれに異なる共有を許可したり、キュー・マネージャーを実行するユーザー ID をグループ化したりします。

この構成例では、3 つのサーバーを使用します。

sun
Windows Server 2008 ドメイン・コントローラー。 *Sun*、*mars* および *venus* を含む *wmq.example.com* ドメインを所有しています。 例示目的で、これをファイル・サーバーとしても使用します。

mars
最初の IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの一つのインスタンスが入っています。

venus
2 番目の IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの二番目のインスタンスが含まれています。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

始める前に

以下の作業手順を実行します。 これらのタスクは、ドメイン・コントローラーおよびドメインを作成し、サーバーに IBM MQ for Windows をインストールし、データおよびログ・ファイル用のファイル共用を作成します。既存のドメイン・コントローラーを構成する場合でも、新しい WindowsServer 2008 でこれらの手順を試してみる価値はあります。 それぞれのドメインに合わせてこれらの手順を調整できます。

1. 513 ページの『Active Directory と DNS のドメインの作成 (Windows)』.
2. 517 ページの『Windows ドメイン内のサーバーまたはワークステーションへの IBM MQ のインストール』.
3. 520 ページの『キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 (Windows)』.

このタスクについて

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。 これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。 さらに、508 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』の作業も必要です。

この作業では、ファイル・サーバーのリモート・ディレクトリーにデータとログを格納するキュー・マネージャーを作成します。 この例では、ファイル・サーバーとして、ドメイン・コントローラーと同じサーバーを使用します。 データ・フォルダーとログ・フォルダーを含むディレクトリーは、グローバル・グループ wmqha に付与された完全な制御権限で共有されます。

手順

1. ローカル管理者 *mars*\Administrator として、ドメイン・サーバー *mars* にログオンします。
2. コマンド・ウィンドウを開きます。
3. IBM MQ サービスを再始動します。

   サービスを実行するユーザー ID で、ここで構成した追加のセキュリティー資格情報を有効にするには、サービスを再始動する必要があります。

   次のコマンドを入力します。

   ```
   endmqsvc
   strmqsvc
   ```

   システム応答:

   ```
   5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
   The MQ service for installation 'Installation1' ended successfully.
   ```

   および:

   ```
   5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
   The MQ service for installation 'Installation1' started successfully.
   ```
4. キュー・マネージャーを作成します。

   ```
   crtmqm -a wmq\wmqha -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\sun\wmq\data -ld \\sun\wmq\logs
   QMGR
   ```

   グローバル・グループ"*wmq**wmqha*"の完全なドメイン名を指定することにより、代替セキュリティー・グループ *wmqha* のドメイン *wmq* を指定する必要があります。

   共有*sun**wmq* の汎用命名規則(UNC)を入力し、マップされたドライブ参照を使用しないようにする必要があります。

   システム応答:

   ```
   IBM MQ queue manager created.
   Directory '\\sun\wmq\data\QMGR' created.
   The queue manager is associated with installation '1'
   Creating or replacing default objects for queue manager 'QMGR'
   Default objects statistics : 74 created. 0 replaced.
   Completing setup.
   Setup completed.
   ```

次のタスク

キューにメッセージを書き込み、キューからメッセージを取り出して、キュー・マネージャーをテストします。

1. キュー・マネージャーを始動します。

   ```
   strmqm QMGR
   ```

   システム応答:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation '1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. テスト・キューを作成します。

```
echo define qlocal(QTEST) | runmqsc QMGR
```

システム応答:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QMGR.


1 : define qlocal(QTEST)
AMQ8006: IBM MQ queue created.
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

3. サンプル・プログラム **amqsput** を使用してテスト・メッセージを書き込みます。

```
echo 'A test message' | amqsput QTEST QMGR
```

システム応答:

```
Sample AMQSPUT0 start
target queue is QTEST
Sample AMQSPUT0 end
```

4. サンプル・プログラム **amqsget** を使用してテスト・メッセージを取り出します。

```
amqsget QTEST QMGR
```

システム応答:

```
Sample AMQSGET0 start
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGET0 end
```

5. キュー・マネージャーを停止させます。

```
endmqm -i QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended.
```

6. キュー・マネージャーを削除してください。

```
dltmqm QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' deleted.
```

7. 作成したディレクトリーを削除します。

   ヒント: /Q オプションをコマンドに追加して、各ファイルまたはディレクトリーを削除するためのプロンプトがコマンドに表示されないようにします。

```
del /F /S C:\wmq\*.*
rmdir /S C:\wmq
```

関連タスク

Windows Active Directory と DNS のドメインの作成 (Windows)

Windows Windows ドメイン内のサーバーまたはワークステーションへの IBM MQ のインストール

Windows キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 (Windows)

Windows *Windows* ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの作成

Windows のドメイン・コントローラーで複数インスタンス・キュー・マネージャーをセットアップする例を取り上げます。 ここで取り上げるのは、実動スケールのセットアップではなく、関連する概念を説明する程度のセットアップです。 この例は、Windows Server 2008 をベースとしています。 それぞれのステップは、Windows Server の他のバージョンでは異なる場合があります。

構成では、ミニ・ドメインの概念、または"ドメインレット"の概念を使用します。Windows 2000、WindowsServer 2003、および WindowsServer 2008 クラスター・ノードをドメイン・コントローラーとして、を参照してください。 複数インスタンス・キュー・マネージャーを既存のドメインに追加する場合は、510 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 (Windows)』を参照してください。

この構成例では、3 つのサーバーを使用します。

sun
: 第 1 のドメイン・コントローラーとして使用する Windows Server 2008 サーバーです。 これは、*sun*、*earth*、および *mars* を含む *wmq.example.com* ドメインを定義します。 これには、*QMGR* と呼ばれるの複数インスタンス・キュー・マネージャーの 一つのインスタンスが含まれています。

earth
: 2 番目のドメイン・コントローラー IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの二番目のインスタンスが含まれています。

mars
: ファイル・サーバーとして使用する Windows Server 2008 です。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

始める前に

1. Windows では、キュー・マネージャー・データとログ・ファイルを保管するファイル・システムを検証する必要はありません。 検査の手順、共有ファイル・システムの動作の検証は、AIX and Linux に適用されます。 Windows では検査は必ず成功します。
2. 513 ページの『Active Directory と DNS のドメインの作成 (Windows)』の手順を実行して、第 1 のドメイン・コントローラーを作成します。
3. 530 ページの『サンプル・ドメインへの 2 番目の Windows ドメイン・コントローラーの追加』の手順を実行して、第 2 のドメイン・コントローラーを追加し、両方のドメイン・コントローラーに IBM MQ for Windows をインストールし、インストール環境を検証します。

4. 531 ページの『サンプル・ドメインでの Windows ドメイン・コントローラーへの IBM MQ のインストール』の手順を実行して、2 つのドメイン・コントローラーに IBM MQ をインストールします。

このタスクについて

同じドメインにあるファイル・サーバーで、キュー・マネージャーのログ・ディレクトリーとデータ・ディレクトリーのための共有フォルダーを作成します。 次に、そのファイル共有フォルダーを使用する複数インスタンス・キュー・マネージャーの最初のインスタンスをどちらかのドメイン・コントローラーで作成します。 もう 1 つのインスタンスをもう 1 つのドメイン・コントローラーで作成し、最後に構成を検証します。 ドメイン・コントローラーでファイル共有フォルダーを作成することも可能です。

このサンプルでは、*sun* は最初のドメイン・コントローラー *earth* 二番目のドメイン・コントローラー、および *mars* がファイル・サーバーです。

手順

1. キュー・マネージャーのデータ・ファイルとログ・ファイルを格納するディレクトリーを作成します。
   a) *mars* の場合、次のコマンドを入力します:

   ```
   md c:\wmq\data , c:\wmq\logs
   ```

2. キュー・マネージャーのデータ・ファイルとログ・ファイルを格納するディレクトリーの共有を設定します。

   ドメイン・ローカル・グループ mqm への全制御アクセスを許可し、キュー・マネージャーを作成するために使用するユーザー ID を許可する必要があります。 この例では、Domain Administrators のメンバーであるユーザー ID は、キュー・マネージャーを作成する権限を持っています。

   ファイル共有は、ドメイン・コントローラーと同じドメインにあるサーバーで設定する必要があります。 この例では、サーバー *mars* は、ドメイン・コントローラーと同じドメインにあります。

   a) Windows エクスプローラーで、 ***c:¥wmq*** > 「**プロパティー**」を右クリックします。
   b)「**セキュリティ**」タブをクリックし、「**詳細設定**」 > 「**編集...**」をクリックします。
   c)「**このオブジェクトの親からの継承可能なアクセス許可を含める**」チェック・ボックスをクリアします。 「Windows セキュリティー」ウィンドウで 「**コピー**」 をクリックします。
   d)「**アクセス許可エントリ**」のリストで「Users」の各行を選択し、「**削除**」をクリックします。 「**アクセス許可エントリ**」のリストにある「SYSTEM」、「Administrators」、「CREATOR OWNER」の各行はそのままにしておきます。
   e)「**追加 ...**」をクリックします。 ドメイン・ローカル・グループ *mqm* の名前を入力します。 「**名前の確認**」をクリックします。
   f)「Windows セキュリティー」ウィンドウに対する応答として、「Domain Administrator」の名前とパスワードを入力して、「**OK** > **OK**」をクリックします。
   g)「wmq の許可エントリー」ウィンドウで、**アクセス権**のリストで「 **完全制御** 」を選択します。
   h)「**OK**」 > 「**適用**」 > 「**OK**」 > 「**OK**」 > 「**OK**」をクリックします。
   i) Domain Administrators を追加するには、e から h を繰り返します。
   j) Windows エクスプローラーで、 ***c:¥wmq*** > 「**共有 ...**」を右クリックします。
   k)「**拡張共有 ...**」 をクリックします。 次に、 「**このフォルダーを共有する**」 チェック・ボックスを選択します。 共用名は *wmq* のままにします。
   l) **許可** > 「**追加 ...**」をクリックします。 ドメイン・ローカル・グループ *mqm* ; Domain Administrators の名前を入力します。 「**名前の確認**」をクリックします。
   m)「Windows セキュリティー」ウィンドウに対する応答として、「Domain Administrator」の名前とパスワードを入力して、「**OK** > **OK**」をクリックします。
3. 最初のドメイン・コントローラー *sun* の上に、キュー・マネージャー *QMGR* を作成します。

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\mars\wmq\data -ld \\mars\wmq\logs QMGR
```

システム応答:

```
IBM MQ queue manager created.
Directory '\\mars\wmq\data\QMGR' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QMGR'.
Default objects statistics : 74 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

4. *sun* 上のキュー・マネージャーを開始し、スタンバイ・インスタンスを許可します。

```
strmqm -x QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

5. で *earth* の二番目 *QMGR* のインスタンスを作成します。

a) Prefix および InstallationName パラメーターの値が *earth* に対して正しいことを確認してください。

sun で **dspmqinf** コマンドを実行します。

```
dspmqinf QMGR
```

システム応答:

```
QueueManager:
Name=QMGR
Directory=QMGR
Prefix=C:\ProgramData\ IBM \ MQ
DataPath=\\mars\wmq\data\QMGR
InstallationName=Installation1
```

b) 機械可読形式の **QueueManager** スタンザをクリップボードにコピーします。

sun で **dspmqinf** コマンドを再び実行します。 今回は、-o command パラメーターを付けます。

```
dspmqinf -o command QMGR
```

システム応答:

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\ProgramData\ IBM \ MQ"
-v DataPath=\\mars\wmq\data\QMGR
```

c) *earth* で **addmqinf** コマンドを実行し、クリップボードの情報を使用して、キュー・マネージャーのインスタンスを *earth* で作成します。

必要に応じて、Prefix パラメーターまたは InstallationName パラメーターの違いに合わせてコマンドを調整します。

```
addmqinf -s QueueManager -v Name= QMGR
-v Directory= QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"
-v DataPath=\\mars\wmq\data\QMGR
```

```
IBM MQ configuration information added.
```

6. *earth* 上のキュー・マネージャーのスタンバイ・インスタンスを開始します。

```
strmqm -x QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
A standby instance of queue manager 'QMGR' has been started. The active
instance is running elsewhere.
```

タスクの結果

キュー・マネージャーが *sun* から *earth* に切り替わることを確認します:

1. *sun* で、次のコマンドを実行します:

```
endmqm -i -r -s QMGR
```

sun 上のシステム応答:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended, permitting switchover to
a standby instance.
```

2. *earth* では、以下のコマンドを繰り返し入力します:

```
dspmq
```

システム応答:

```
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running)
```

次のタスク

サンプル・プログラムを使用して複数インスタンス・キュー・マネージャーを検証するには、534 ページの『Windows での複数インスタンス・キュー・マネージャーの検査』を参照してください。

関連タスク

530 ページの『サンプル・ドメインへの 2 番目の Windows ドメイン・コントローラーの追加』
531 ページの『サンプル・ドメインでの Windows ドメイン・コントローラーへの IBM MQ のインストール』

関連情報

Windows 2000、Windows Server 2003、および Windows Server 2008 クラスター・ノードをドメイン・コントローラーとして構成する

Windows サンプル・ドメインへの *2* 番目の *Windows* ドメイン・コントローラーの追加

二番目のドメイン・コントローラーを *wmq.example.com* ドメインに追加して、ドメイン・コントローラーおよびファイル・サーバー上でマルチインスタンス・キュー・マネージャーを実行する Windows ドメインを構成します。

この構成例では、3 つのサーバーを使用します。

sun
: 第 1 のドメイン・コントローラーとして使用する Windows Server 2008 サーバーです。 これは、*sun*、*earth*、および *mars* を含む *wmq.example.com* ドメインを定義します。 これには、*QMGR* と呼ばれるの複数インスタンス・キュー・マネージャーの 一つのインスタンスが含まれています。

earth
: 2 番目のドメイン・コントローラー IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの二番目のインスタンスが含まれています。

mars
: ファイル・サーバーとして使用する Windows Server 2008 です。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

始める前に

1. 513 ページの『Active Directory と DNS のドメインの作成 (Windows)』のステップを実行して、ドメイン *wmq.example.com* 用のドメイン・コントローラー *sun* を作成します。 斜体の名前は、それぞれの構成に合わせて変更してください。
2. WindowsServer 2008 をデフォルトのワークグループ WORKGROUP 内のサーバーにインストールします。 この例では、サーバーの名前は *earth* になります。

このタスクについて

このタスクでは、*earth* という名前の WindowsServer 2008 を、*wmq.example.com* ドメイン内の二番目のドメイン・コントローラーとして構成します。

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。 これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。 さらに、508 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』の作業も必要です。

手順

1. ドメイン・コントローラー *sun.wmq.example.com*to*earth* を DNS サーバーとして追加します。
   a) *earth* で、*earth*\Administrator としてログオンし、**スタート**をクリックします。
   b)「**ネットワーク**」 >「**プロパティー**」 >「**ネットワーク接続の管理**」を右クリックします。
   c) ネットワーク・アダプターを右クリックして、「**プロパティ**」をクリックします。

      システムの応答として、「ローカル エリア接続のプロパティ」ウィンドウが開き、接続で使用する項目のリストが表示されます。
   d)「ローカル エリア接続のプロパティ」ウィンドウの項目リストから、「**インターネット プロトコル バージョン 4**」または「**インターネット プロトコル IBM WebSphere MQ 6**」を選択します。 「**プロパティー**」 >「**拡張 ...**」 をクリックします。 「**DNS**」 タブをクリックします。
   e) DNS サーバー・アドレスで、「**追加...**」をクリックします。

f) DNS サーバーでもあるドメイン・コントローラーの IP アドレスを入力して、「**追加**」をクリックします。

g)「**以下の DNS サフィックスを順に追加する**」 > 「**追加...**」をクリックします。

h) タイプ *wmq.example.com* と**追加**をクリックします。

i) **この接続の DNS サフィックス**フィールドでのタイプ *wmq.example.com*。

j)「**この接続のアドレスを DNS に登録する**」と「**この接続の DNS サフィックスを DNS 登録に使う**」を選択します。 「**OK**」 > 「**OK**」 > 「**閉じる**」をクリックします。

k) コマンド・ウィンドウで、**ipconfig /all** というコマンドを入力して、TCP/IP 設定を確認します。

2. ローカル管理者または Workgroup 管理者として、ドメイン・コントローラー *sun* にログオンします。

サーバーが既にドメイン・コントローラーとして構成されている場合は、ドメイン管理者としてログオンする必要があります。

3.「Active Directory ドメイン サービス」ウィザードを実行します。

a) クリック**スタート > 実行...** dcpromo と入力して、 「**OK**」をクリックします。

Active Directory のバイナリー・ファイルがインストールされていなければ、Windows によってそれらのファイルが自動的にインストールされます。

4. *earth* を *wmq.example.com* ドメインの二番目のドメイン・コントローラーとして構成します。

a) ウィザードの最初のウィンドウで、「**詳細モード インストールを使用する**」チェック・ボックスは、クリアされた状態のままにしておきます。 「**次へ**」 >「**次へ**」をクリックし、「**既存のドメインにドメイン コントローラを追加する**」 > 「**次へ**」をクリックします。

b)「**このフォレスト内の任意のドメインの名前を入力してください ...**」に *wmq* と入力します。 設定します。 「**代替の資格情報**」ラジオ・ボタンがクリックされた状態になっていることを確認し、「**設定...**」をクリックします。 ドメイン管理者の名前とパスワードを入力し、「**OK**」 > 「**次へ**」 > 「**次へ**」 > 「**次へ**」をクリックします。

c)「追加のドメイン コントローラ オプション」ウィンドウで、「**DNS サーバー**」オプションと「**グローバル カタログ**」オプションが選択された状態になります。 それらのオプションをそのまま受け入れ、「**次へ**」 > 「**次へ**」をクリックします。

d)「ディレクトリー サービス復元モード Administrator パスワード」ウィンドウの「**パスワード**」と「**パスワードの確認入力**」にパスワードを入力します。 「**次へ**」 > 「**次へ**」をクリックします。

e)「**ネットワーク資格情報**」のプロンプト画面が表示されたら、ドメイン管理者のパスワードを入力します。 ウィザードの最後のウィンドウで「**完了時に再起動する**」を選択します。

f) しばらくすると、DNS の委任に関する **DCPromo** エラーのウィンドウが表示されることがあります。「**OK**」をクリックします。 サーバーがリブートします。

タスクの結果

earth がリブートされたら、ドメイン管理者としてログオンします。 wmq.example.com ドメインが *earth* に複製されていることを確認します。

次のタスク

IBM MQ のインストールに進みます。 531 ページの『サンプル・ドメインでの Windows ドメイン・コントローラーへの IBM MQ のインストール』を参照してください。

関連タスク

Windows サンプル・ドメインでの Windows ドメイン・コントローラーへの IBM MQ のインストール
513 ページの『Active Directory と DNS のドメインの作成 (Windows)』

Windows サンプル・ドメインでの *Windows* ドメイン・コントローラーへの *IBM MQ* のインストール
wmq.example.com ドメイン内の両方のドメイン・コントローラーに IBM MQ のインストールをインストールおよび構成します。

この構成例では、3 つのサーバーを使用します。

sun
第 1 のドメイン・コントローラーとして使用する Windows Server 2008 サーバーです。 これは、*sun*、*earth*、および *mars* を含む *wmq.example.com* ドメインを定義します。 これには、*QMGR* と呼ばれるの複数インスタンス・キュー・マネージャーの 一つのインスタンスが含まれています。

earth
2 番目のドメイン・コントローラー IBM MQ サーバーとして使用される Windows Server 2008。 これには、*QMGR* という名前の複数インスタンス・キュー・マネージャーの二番目のインスタンスが含まれています。

mars
ファイル・サーバーとして使用する Windows Server 2008 です。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

始める前に

1. 513 ページの『Active Directory と DNS のドメインの作成 (Windows)』のステップを実行して、ドメイン *wmq.example.com* 用のドメイン・コントローラー *sun* を作成します。 斜体の名前は、それぞれの構成に合わせて変更してください。
2. 530 ページの『サンプル・ドメインへの 2 番目の Windows ドメイン・コントローラーの追加』のステップを実行して、ドメイン *wmq.example.com* 用の二番目のドメイン・コントローラー *earth* を作成します。 斜体の名前は、それぞれの構成に合わせて変更してください。
3. IBM MQ を実行できるその他の Windows バージョンについては、 Windows システムでのハードウェア要件およびソフトウェア要件 を参照してください。

このタスクについて

wmq.example.com ドメイン内の両方のドメイン・コントローラーに IBM MQ のインストールをインストールおよび構成します。

手順

1. IBM MQ で *sun* と *earth* をインストールします。

   詳しくは、 Windows への IBM MQ サーバーのインストール を参照してください。

   a) *sun* と *earth* の両方で、ドメイン管理者 *wmq*\Administrator としてログオンします。

   b) **Setup** コマンドを IBM MQ for Windows インストール・メディアで実行します。

   IBM MQ のランチパッド・アプリケーションが始動します。

   c)「**ソフトウェア要件**」をクリックして、前提ソフトウェアがインストールされていることを確認します。

   d)「**ネットワーク構成**」 > 「**いいえ**」をクリックします。

   このインストール環境では、ドメイン・ユーザー ID を構成してもしなくてもかまいません。 作成されるユーザー ID は、ドメイン・ローカル・ユーザー ID になります。

   e)「**IBM MQ インストール**」をクリックし、インストール言語を選択し、「IBM MQ インストーラーの起動」をクリックします。

   f) 使用条件を確認し、「**次へ**」 > 「**次へ**」 > 「**インストール**」をクリックして、デフォルト構成を受け入れます。 インストールが完了するのを待って、「**完了**」をクリックします。

   インストール環境の名前を変更したり、別のコンポーネントをインストールしたり、キュー・マネージャーのデータとログのために別のディレクトリーを構成したり、別のディレクトリーにインストールしたりする場合は、「**標準**」ではなく「**カスタム**」をクリックします。

   IBM MQ がインストールされ、インストーラーによって Prepare IBM MQ Wizard が開始されます。

IBM MQ for Windows インストールでは、ドメイン・ローカル・グループ mqm、およびドメイン・グループ Domain mqm を構成します。 の Domain mqmmqm メンバーになります。 同じドメイン内の後続のドメイン・コントローラーは、mqm グループと Domain mqm グループを共有します。

2. *earth* と *sun* の両方で、Prepare IBM MQ Wizard を実行します。

   詳しくは、 Prepare IBM MQ Wizard を使用した IBM MQ の構成を参照してください。

   a) IBM MQ インストーラーによって Prepare IBM MQ Wizard が自動的に実行されます。

      ウィザードを手動で開始するには、 「**スタート**」 > 「**すべてのプログラム**」 > **IBM MQ** フォルダーで Prepare IBM MQ Wizard へのショートカットを見つけます。 複数インストール構成の場合は、IBM MQ のインストールに対応するショートカットを選択します。

   b)「**次**」をクリックして、質問に答えて**いいえ**を「クリック」してください。"ネットワーク内に Windows 2000 またはそれ以降のドメイン・コントローラーがあるかどうかを確認する"[1].

   c) ウィザードの最終ページで、必要に応じて各種のチェック・ボックスを選択したりクリアしたりして、「**完了**」をクリックします。

   Prepare IBM MQ Wizard によって、第 1 のドメイン・コントローラーにドメイン・ローカル・ユーザー MUSR_MQADMIN が作成され、第 2 のドメイン・コントローラーに別のドメイン・ローカル・ユーザー MUSR_MQADMIN1 が作成されます。 さらに、このウィザードによって、各コントローラーに IBM MQ サービスが作成され、このサービスにログオンするユーザーが MUSR_MQADMIN または MUSR_MQADMIN1 になります。

3. キュー・マネージャーを作成する権限を持ったユーザーを定義します。

   ユーザーはローカルにログオンする権限を持っている必要があり、ドメイン・ローカル mqm グループのメンバーでなければなりません。 ドメイン・コントローラーの場合、ローカル・ログオンの権限は、ドメイン・ユーザーではなく管理者にあります。 デフォルトでその両方の属性を兼ね備えているユーザーはいません。 このタスクでは、ドメイン管理者をドメインのローカル mqm グループに追加します。

   a) **[サーバーマネージャー] > [ロール] > Active Directory Domain Services > *wmq.example.com* > [ユーザー]**を開きます。

   b) **ドメイン管理者** > 「**グループに追加 ...**」 を右クリックします。 次に mqm と入力します。 「**名前の確認**」 > **OK** > 「**OK**」 をクリックします。

タスクの結果

1. Prepare IBM MQ Wizard によってドメイン・ユーザー MUSR_MQADMIN が作成されたことを確認します。

   a. **[サーバーマネージャー] > [ロール] > Active Directory Domain Services > *wmq.example.com* > [ユーザー]**を開きます。

   b. 右クリック **MUSR_MQADMIN** > **プロパティー... > 次のメンバー**。これが Domain users および mqm のメンバーであることを確認します。

2. MUSR_MQADMIN に、サービスとして実行する権限があることを確認します。

   a.「**スタート**」 > 「**ファイル名を指定して実行 ...**」をクリックします。 コマンド **secpol.msc** を入力し、 「**OK**」をクリックします。 をクリックします。

   b.「**セキュリティの設定**」 > 「**ローカル ポリシー**」 > 「**ユーザー権利の割り当て**」を開きます。 ポリシーの中で、**サービスとしてログオン** > **プロパティ**を右クリックします。サービスとしてログオンする権限があるユーザーとして MUSR_MQADMIN が表示されることを確認します。 **OK** をクリックします。

[1] ドメインのインストールを構成することができます。 ドメイン・コントローラーに存在するすべてのユーザーとグループのスコープはドメイン・スコープなので、どちらにしても違いはありません。 IBM MQ をドメイン内にないものとして、インストールする方が簡単です。

次のタスク

1. 541 ページの『ローカル mqm グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りおよび書き込み』の作業を実行して、インストールと構成が正しく機能していることを検証します。
2. 526 ページの『Windows ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの作成』の作業に戻り、ドメイン・コントローラーで複数インスタンス・キュー・マネージャーを構成する作業を完了します。

関連タスク

Windows サンプル・ドメインへの 2 番目の Windows ドメイン・コントローラーの追加

関連資料

IBM MQ Windows サービスに必要なユーザー権限

Windows *Windows* での複数インスタンス・キュー・マネージャーの検査

複数インスタンス・キュー・マネージャー構成を検査するには、サンプル・プログラム **amqsghac**、**amqsphac**、および **amqsmhac** を使用します。 このトピックでは、Windows Server 2003 上で複数インスタンス・キュー・マネージャーの構成を検査するためのサンプル構成を記載しています。

高可用性サンプル・プログラムは、クライアントの自動再接続を使用します。 接続されたキュー・マネージャーに障害が起きた場合、クライアントは同じキュー・マネージャー・グループ内のキュー・マネージャーへの再接続を試行します。 サンプルの説明、高可用性のサンプル・プログラムでは、単純にするために単一インスタンス・キュー・マネージャーを使用してクライアントの再接続を説明しています。 複数インスタンス・キュー・マネージャーがある同じサンプルを使用して、複数インスタンス・キュー・マネージャー構成を検査することができます。

このサンプルでは、526 ページの『Windows ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの作成』で説明されている複数インスタンス構成を使用しています。 この構成を使用して、複数インスタンス・キュー・マネージャーがスタンバイ・インスタンスに切り替わることを検査します。 **endmqm** コマンドを使用してキュー・マネージャーを停止し、-s 切り替えオプションを使用します。 クライアント・プログラムは、新規キュー・マネージャー・インスタンスに再接続して、少しの遅延の後に新規インスタンスの処理を続行します。

クライアントは、Windows7 Service Pack 1 を実行している 400MB の VMware イメージにインストールされています。 セキュリティー上の理由から、マルチインスタンス・キュー・マネージャーを実行しているドメイン・サーバーと同じ VMware ホスト専用ネットワーク上で接続されています。 これは、構成を単純化するために、クライアント接続テーブルを含む/MQHA フォルダーを共用しています。

IBM MQ Explorer を使用したフェイルオーバーの検査

フェイルオーバーを検査するためにサンプル・アプリケーションを使用する前に、IBM MQ Explorer を各サーバー上で実行します。 「**リモート・キュー・マネージャーの追加**」>「**複数インスタンス・キュー・マネージャーへの直接接続**」ウィザードを使用して、両方のキュー・マネージャー・インスタンスを各エクスプローラーに追加します。 両方のインスタンスをスタンバイを許可して実行しておきます。 アクティブ・インスタンスがある VMware イメージを実行しているウィンドウを閉じて仮想的にサーバーの電源をオフにするか、またはアクティブ・インスタンスを停止して、スタンバイ・インスタンスおよび再接続する再接続可能クライアントへの切り替えを許可します。

重要 : サーバーの電源をオフにしている場合は、サーバーが MQHA フォルダーをホストしていないことを確認してください!

注 :「**キュー・マネージャーの停止**」ダイアログで「**スタンバイ・インスタンスへの切り替えを許可**」オプションが表示されない場合があります。 このオプションがないのは、キュー・マネージャーが単一インスタンス・キュー・マネージャーとして実行されているためです。 それを開始したときに、「**スタンバイ・インスタンスの許可**」オプションを指定しませんでした。 キュー・マネージャーの停止要求が拒否された場合は、「**詳細**」ウィンドウを参照してください。実行中のスタンバイ・インスタンスがない可能性があります。

サンプル・プログラムを使用したフェイルオーバーの検査

アクティブ・インスタンスを実行するサーバーの選択

MQHA ディレクトリーまたはファイル・システムをホストするサーバーの一つを選択した可能性があります。 アクティブ・サーバーを実行している VMware ウィンドウを閉じて、フェイルオーバーをテストする場合は、MQHA をホスティングしているものではないことを確認してください！

アクティブ・キュー・マネージャー・インスタンスを実行するサーバー上

1. *ipaddr1* および *ipaddr2* を変更し、以下のコマンドを N:\hasample.tst.に保管する

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN)  TRPTYPE(TCP) +
MCAUSER(' ') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(' ipaddr1 (1414), ipaddr2 (1414)') QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

   注 : **MCAUSER** パラメーターをブランクのままにすると、クライアント・ユーザー ID がサーバーに送信されます。 クライアント・ユーザー ID には、サーバーに対する正しいアクセス権が必要です。 別の方法として、SVRCONN チャネル内の **MCAUSER** パラメーターを、サーバー上で構成済みのユーザー ID に設定します。

2. パス N:\を指定して、コマンド・プロンプトを開き、以下のコマンドを実行します:

```
runmqsc -m QM1 < hasample.tst
```

3. **runmqsc** コマンドの出力を検査することで、リスナーが実行中であり、キュー・マネージャー制御があることを確認します。

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

   または、IBM MQ Explorer を使用して、TCPIP リスナーが実行中で、Control = Queue Manager があることを確認します。

クライアント上

1. サーバー上の共有ディレクトリー C:\MQHA、クライアント上の N:\にマップします。
2. パス N:\を指定してコマンド・プロンプトを開きます。 次のようにして、環境変数 MQCHLLIB を、サーバー上のクライアント・チャネル定義テーブル (CCDT) を指すように設定します。

```
SET MQCHLLIB=N:\data\QM1\@ipcc
```

3. コマンド・プロンプトで、以下のコマンドを入力します。

```
start amqsghac TARGET QM1
start amqsmhac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

   注 : 問題がある場合は、コマンド・プロンプトでアプリケーションを開始し、理由コードがコンソールに出力されるようにするか、N:\data\QM1\errors フォルダー内の AMQERR01.LOG ファイルを調べてください。

アクティブ・キュー・マネージャー・インスタンスを実行するサーバー上

1. 以下のいずれかを実行します。
   - アクティブ・サーバー・インスタンスがある VMware イメージを実行するウィンドウを閉じます。
   - IBM MQ Explorer を使用して、アクティブ・キュー・マネージャー・インスタンスを停止して、スタンバイ・インスタンスへの切り替えを許可し、再接続可能クライアントに再接続を指示します。

2. 最終的には 3 つのクライアントで接続が切断していることが検出され、次いで再接続されます。 この構成では、サーバー・ウィンドウを閉じた場合、3 つの接続がすべて再確立されるには約 7 分かかります。 他の接続よりも速く再確立される接続もあります。

結果

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start

17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsghac TARGET QM1
Sample AMQSGHAC start
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

Windows 共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 *(Windows)*

このトピックでは、グローバル代替セキュリティー・グループを使用して、キュー・マネージャーのデータ・ファイルとログ・ファイルの共有場所を保護する方法を取り上げます。 別々のサーバーで実行するキュー・マネージャーの複数インスタンスで、その場所を共有できます。

通常、キュー・マネージャーのデータ・ファイルとログ・ファイルの共有格納場所をセットアップすることはありません。 IBM MQ for Windows をインストールするときに、そのサーバーで作成するキュー・マネージャーのために指定したホーム・ディレクトリーがインストール・プログラムによって作成されます。 キュー・マネージャーのディレクトリーは、ローカル mqm グループによって保護され、それらのディレクトリーにアクセスする IBM MQ サービスのユーザー ID が構成されます。

セキュリティー・グループを使用して共有フォルダーを保護する場合、そのフォルダーにアクセスする権限を持ったユーザーには、そのグループの資格情報が必要になります。 例えば、*mars* というサーバーのローカル mqm グループによって、リモート・ファイル・サーバーにあるフォルダーを保護するとしましょう。 その場合は、キュー・マネージャーのプロセスを実行するユーザーを *mars* のローカル mqm グループのメンバーにします。 そのユーザーには、リモート・ファイル・サーバーにあるフォルダーの資格情報に合致

する資格情報を設定します。 その資格情報を使用して、キュー・マネージャーは、そのフォルダーにあるデータ・ファイルとログ・ファイルにアクセスできます。 別のサーバーでキュー・マネージャーのプロセスを実行するユーザーは、別のローカル mqm グループのメンバーであり、資格情報が合致しません。 キュー・マネージャーは、*mars* に対して、異なるサーバー上で実行される場合、*mars* 上で実行されたときに、作成したデータおよびログ・ファイルにアクセスできません。 そのユーザーをドメイン・ユーザーにしたとしても、そのユーザーの資格情報は異なります。 そのユーザーは、*mars* のローカル mqm グループから資格情報を取得する必要があり、別のサーバーからその資格情報を取得することはできないからです。

この問題は、キュー・マネージャーのためにグローバル代替セキュリティー・グループを設定することによって解決できます。 537 ページの図 73 を参照してください。 つまり、グローバル・グループによってリモート・フォルダーを保護するということです。 グローバル・グループを *mars* 上で作成するときに、そのグローバル・グループの名前をキュー・マネージャーに渡します。 そのグローバル・グループ名を代替セキュリティー・グループとして渡すときには、**crtmqm** コマンドの `-a[r]` パラメーターを使用します。 キュー・マネージャーを別のサーバーに移動する場合は、そのセキュリティー・グループの名前も一緒に移動します。 具体的には、`qm.ini` ファイルの **AccessMode** スタンザにある `SecurityGroup` でその名前を移動することになります。 例えば、以下のとおりです。

```
AccessMode:
SecurityGroup=wmq\wmq
```

`qm.ini` の **AccessMode** スタンザには、`RemoveMQMAccess` も含まれます。例えば、以下のとおりです。

```
AccessMode:
RemoveMQMAccess=true|false
```

値 `true` を使用してこの属性を指定し、アクセス・グループも指定していると、ローカル mqm グループはキュー・マネージャー・データ・ファイルへのアクセス権を付与されません。

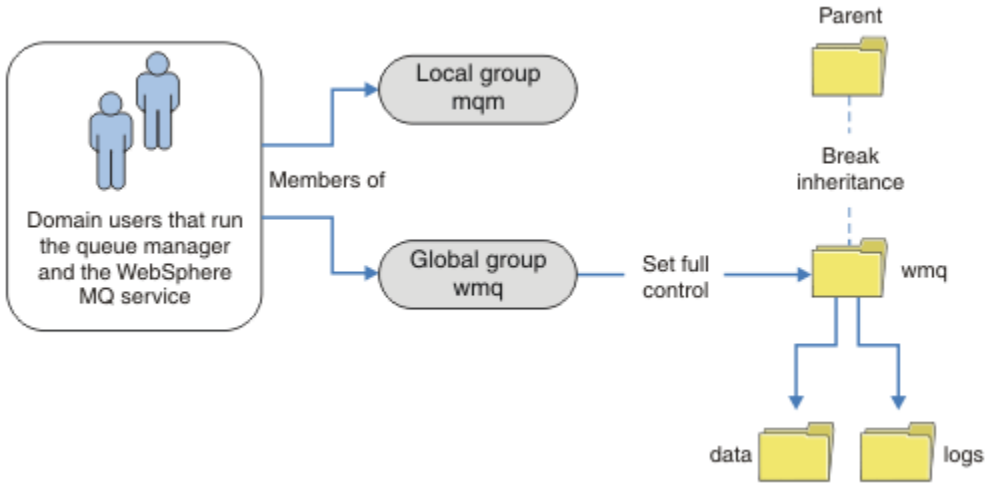


図 *73.* 代替グローバル・セキュリティー・グループを使用してキュー・マネージャーのデータとログを保護する操作 *(1)*

キュー・マネージャーのプロセスを実行するユーザー ID に、グローバル・セキュリティー・グループと合致する資格情報を設定するには、そのユーザー ID のスコープがグローバル・スコープになっていることも必要です。 ローカル・グループまたはローカル・プリンシパルをグローバル・グループのメンバーにすることはできません。 537 ページの図 73 では、キュー・マネージャーのプロセスを実行するユーザーをドメイン・ユーザーとしています。

多数の IBM MQ サーバーをデプロイする場合は、537 ページの図 73 にあるようにユーザーをグループ化する作業が面倒になります。 ローカル・グループにユーザーを追加するプロセスを IBM MQ サーバーごとに繰り返さなければならないからです。 代わりに、ドメイン・コントローラー上に `Domain mqm` グローバル・グループを作成し、`Domain mqm` グループの IBM MQ メンバーを実行するユーザーを作成します。

538 ページの図 74 を参照してください。 IBM MQ をドメイン・インストールとしてインストールすると、Prepare IBM MQ Wizard は自動的に `Domain mqm` ローカル `mqm` グループのメンバーをグループ化します。 同じユーザーが、グローバル・グループ `Domain mqm` と `wmq` の両方に存在します。

ヒント: 同じユーザーが別々のサーバーで IBM MQ を実行することは可能ですが、1 つのサーバーで IBM MQ をサービスとして実行するユーザーと、対話式に実行するユーザーは、別々にしなければなりません。 1 つのサーバーにあるインストール環境ごとに別々のユーザーを使用することも必要です。 したがって、`Domain mqm` には通常、多数のユーザーが含まれます。

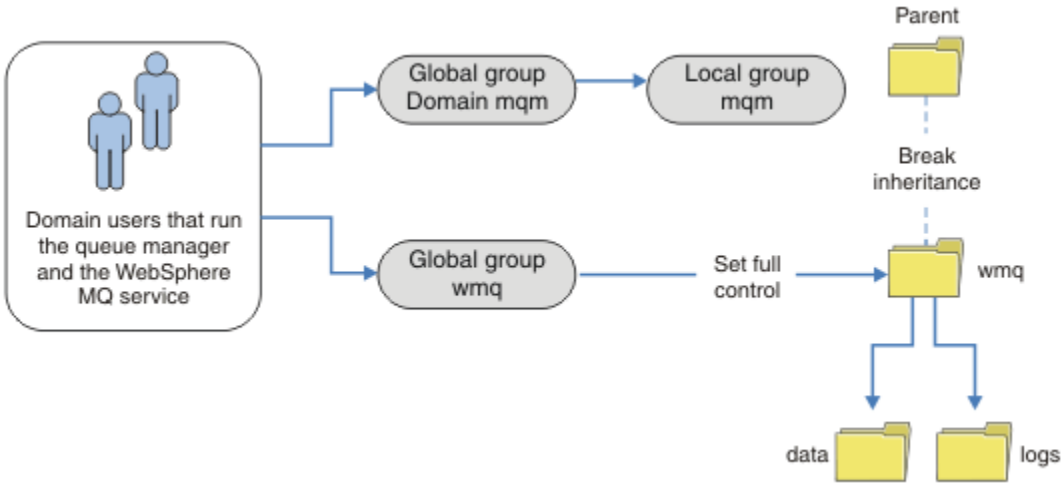


図 *74.* 代替グローバル・セキュリティー・グループを使用してキュー・マネージャーのデータとログを保護する操作 *(2)*

538 ページの図 74 の編成は、必要以上に複雑になっています。 この配置では、2 つのグローバル・グループに同じメンバーが入っています。 この編成を簡略化して、グローバル・グループを 1 つだけ定義した環境が 538 ページの図 75 です。

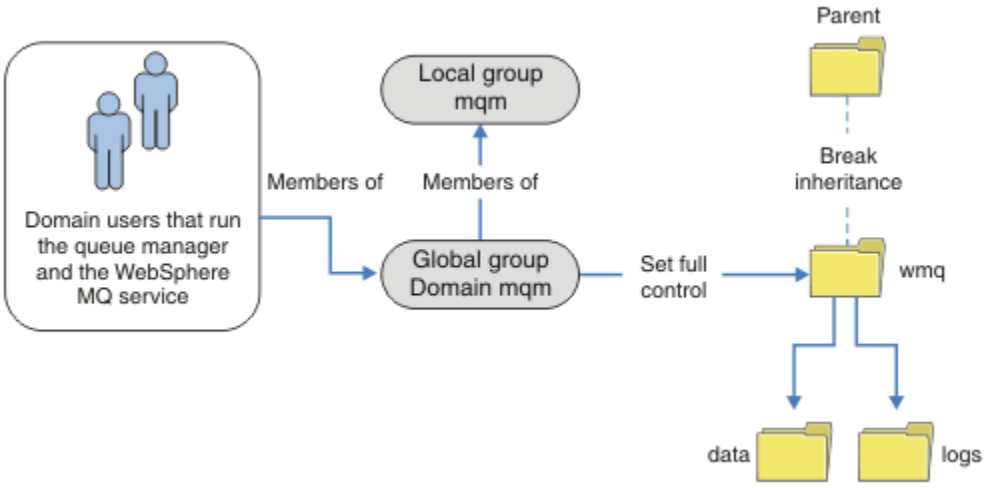


図 *75.* 代替グローバル・セキュリティー・グループを使用してキュー・マネージャーのデータとログを保護する操作 *(3)*

場合によっては、それぞれのキュー・マネージャーからアクセスできるフォルダーを制限することによって、アクセス制御をきめ細かく設定しなければならない環境もあります。 539 ページの図 76 を参照してください。 539 ページの図 76 では、ドメイン・ユーザーの 2 つのグループを定義し、別々のグローバ

ル・グループで、別々のキュー・マネージャーのログ・ファイルとデータ・ファイルを保護しています。 二つの異なる IBM MQ ローカルグループが表示されていますが、これらは異なる mqm サーバーにあるはずです。 この例では、各キュー・マネージャーを 2 つのセットに区分化し、その 2 つのセットに別々のユーザーを割り振ります。 例えば、テスト用のキュー・マネージャーと実動用のキュー・マネージャーという 2 つのセットを設定する環境などが考えられます。 それぞれの代替セキュリティー・グループの名前は、wmq1 と wmq2 です。 テストの区分に所属するのか、実動の区分に所属するのかに応じて、グローバル・グループ wmq1 と wmq2 を正しいキュー・マネージャーに手動で追加しなければなりません。 2 つのユーザー・グループがあるため、538 ページの図 75 の場合と同様に、IBM MQ のインストールによって Domain mqm がローカル mqm グループに伝搬されるということを構成で利用することはできません。

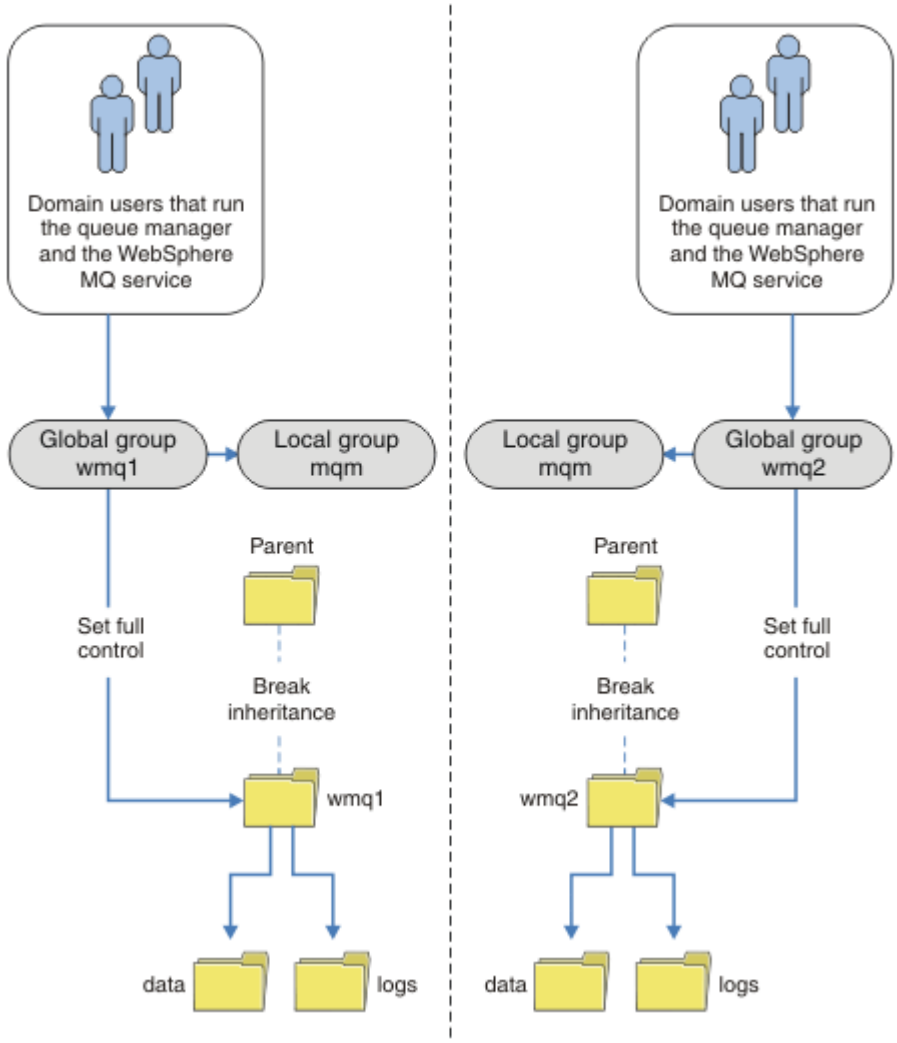


図 76. 代替グローバル・セキュリティー・プリンシパルを使用してキュー・マネージャーのデータとログを保護する操作 *(4)*

2 つに区分するためのもう 1 つの方法は、それぞれの区分を 2 つの Windows ドメインに配置することです。 その場合は、538 ページの図 75 にあるようなシンプルなモデルに戻ることができます。

Windows *Windows* でキュー・マネージャーのデータとログの非共有ディレクトリーおよびファイルを保護する操作

このトピックでは、ローカル mqm グループと代替セキュリティー・グループの両方を使用して、キュー・マネージャーのデータ・ファイルとログ・ファイルの代替格納場所を保護する方法を取り上げます。

通常、キュー・マネージャーのデータ・ファイルとログ・ファイルの代替格納場所をセットアップすることはありません。 IBM MQ for Windows をインストールするときに、作成するキュー・マネージャーのた

めに指定したホーム・ディレクトリーがインストール・プログラムによって作成されます。 キュー・マネージャーのディレクトリーは、ローカル mqm グループによって保護され、それらのディレクトリーにアクセスする IBM MQ サービスのユーザー ID が構成されます。

IBM MQ のアクセス制御を構成する方法を 2 つの例で確認します。 これらの例では、インストール時に作成されるデータとログのパスに存在しないディレクトリーにデータとログを格納するキュー・マネージャーを作成する方法を見ていきます。 最初の例 (541 ページの『ローカル mqm グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りおよび書き込み』) では、ローカル mqm グループによって、キューとログのディレクトリーに対するアクセスを許可します。 2 番目の例 (545 ページの『代替ローカル・セキュリティー・グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りと書き込み』) では、代替セキュリティー・グループによって、それらのディレクトリーに対するアクセスを許可します。 1 つのサーバーだけで実行するキュー・マネージャーがそれらのディレクトリーにアクセスする場合は、代替セキュリティー・グループでデータ・ファイルとログ・ファイルを保護することによって、他のローカル・グループやローカル・プリンシパルで他のキュー・マネージャーを保護するという選択肢が可能になります。 一方、複数のサーバーで実行するキュー・マネージャー (複数インスタンス・キュー・マネージャーなど) がそれらのディレクトリーにアクセスする場合は、代替セキュリティー・グループによってデータ・ファイルとログ・ファイルを保護することが唯一の選択肢になります。 536 ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 (Windows)』 を参照してください。

Windows の場合、キュー・マネージャーのデータ・ファイルとログ・ファイルに関するセキュリティー権限を構成する作業は、一般的な作業ではありません。 IBM MQ for Windows のインストール時に、キュー・マネージャーのデータとログのディレクトリーを指定するか、デフォルトのディレクトリーをそのまま受け入れます。 インストール・プログラムは、これらのディレクトリーをローカル mqm グループで自動的に保護し、完全な制御権限を付与します。 インストール・プロセスによって、キュー・マネージャーを実行するユーザー ID は、そのローカル mqm グループのメンバーになります。 それぞれのアクセス要件に合わせて、それらのディレクトリーに対する他のアクセス権限を変更することもできます。

データ・ファイルとログ・ファイルのディレクトリーを新しい場所に移動する場合は、その新しい場所のセキュリティーを構成する必要があります。 それらのディレクトリーの場所を変更する可能性があるのは、キュー・マネージャーをバックアップして別のコンピューターにリストアする場合や、キュー・マネージャーを複数インスタンス・キュー・マネージャーに変更する場合などです。 新しい場所に移したキュー・マネージャーのデータとログのディレクトリーを保護する方法には、2 つの選択肢があります。 つまり、アクセス権をローカル mqm グループに限定する方法と、任意のセキュリティー・グループに限定する方法のいずれかで、それらのディレクトリーを保護できます。

それらのディレクトリーを保護するための手順の数が少なくてすむのは、ローカル mqm グループを使用する場合です。 データとログのディレクトリーに対するアクセス許可を設定し、ローカル mqm グループにフル・コントロール権限を与えます。 そのための標準的な方法は、既存のアクセス許可をコピーして、親フォルダーからの継承を解除することです。 その後、他のプリンシパルのアクセス許可を削除するか、制限できます。

「IBM MQ の準備」ウィザードによってセットアップされるサービスのユーザー ID とは異なるユーザー ID でキュー・マネージャーを実行する場合は、そのユーザー ID をローカル mqm グループのメンバーにしなければなりません。 そのための手順については、541 ページの『ローカル mqm グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りおよび書き込み』 の作業を参照してください。

代替セキュリティー・グループを使用して、キュー・マネージャーのデータ・ファイルとログ・ファイルを保護することもできます。 代替セキュリティー・グループを使用して、キュー・マネージャーのデータ・ファイルとログ・ファイルを保護するプロセスでは、いくつかの手順を実行しなければなりません。 541 ページの図 77 を参照してください。 ローカル・グループ wmq は、代替セキュリティー・グループの例です。

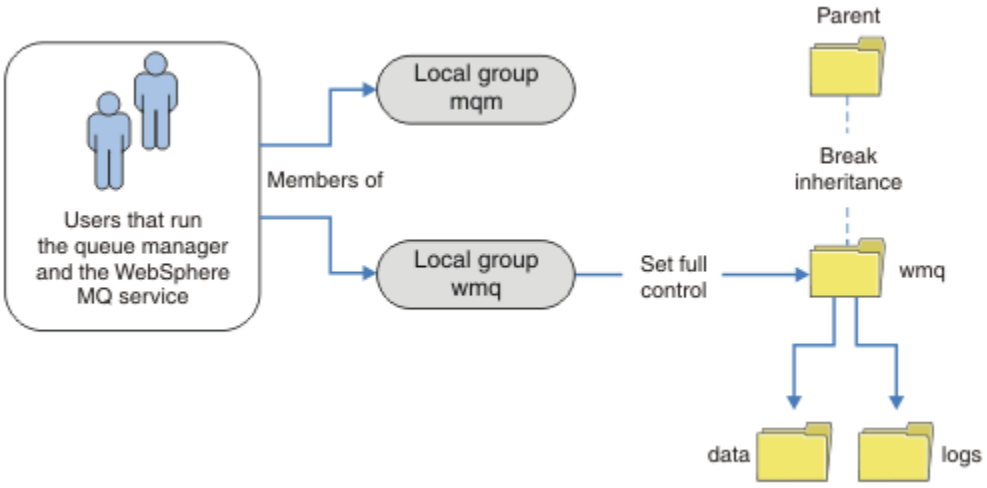


図 77. 代替ローカル・セキュリティー・グループを使用して、キュー・マネージャーのデータおよびログを保護する。*wmq*

1. キュー・マネージャーのデータとログのために別々のディレクトリーを作成するか、共通のディレクトリーを作成するか、共通の親ディレクトリーを作成します。
2. 既存の継承アクセス許可をそれらのディレクトリーまたは親ディレクトリーにコピーして、それぞれの要件に合わせて変更します。
3. キュー・マネージャーおよびログを含むディレクトリーを、ディレクトリーに対する代替グループ wmq、フル・コントロール許可を付与することによって保護します。
4. キュー・マネージャーのプロセスを実行するすべてのユーザー ID に、代替セキュリティー・グループまたは代替セキュリティー・プリンシパルの資格情報を与えます。
   a. ユーザーを代替セキュリティー・プリンシパルとして定義する場合、そのユーザーは、キュー・マネージャーを実行するユーザーと同じユーザーでなければなりません。 そのユーザーは、ローカル mqm グループのメンバーでなければなりません。
   b. ローカル・グループを代替セキュリティー・グループとして定義する場合は、その代替グループに、キュー・マネージャーを実行するユーザーを追加しなければなりません。 そのユーザーは、ローカル mqm グループのメンバーでもなければなりません。
   c. グローバル・グループを代替セキュリティー・グループとして定義する場合は、536 ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 (Windows)』を参照してください。
5. **crtmqm** コマンドの -a パラメーターで代替セキュリティー・グループまたは代替セキュリティー・プリンシパルを指定して、キュー・マネージャーを作成します。

Windows ローカル *mqm* グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りおよび書き込み

この作業では、キュー・マネージャーを作成し、そのデータ・ファイルとログ・ファイルを任意のディレクトリーに格納する方法を確認します。 ファイルへのアクセスは、ローカル mqm グループによって保護されます。 ディレクトリーは共有しません。

始める前に

1. IBM MQ for Windows をプライマリー・インストールとしてインストールします。
2. Prepare IBM MQ Wizard を実行します。

   詳しくは、 Prepare IBM MQ Wizard を使用した IBM MQ の構成を参照してください。

この作業では、ローカル・ユーザー ID またはドメイン・ユーザー ID のどちらかでインストール環境を実行するように構成します。 最終的に、508 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』のすべての作業を完了するには、ドメインの一部としてインストール環境を構成する必要があります。

3. この作業の最初の部分を実行するために、管理者権限でログオンします。

このタスクについて

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。 これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。 さらに、508 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』の作業も必要です。

Windows では、IBM MQ for Windows のデフォルトのデータ・パスとログ・パスを任意のディレクトリーに作成できます。 インストールおよび構成ウィザードは、ローカル mqm グループ、およびキュー・マネージャー・プロセスを実行するユーザー ID に、ディレクトリーへのアクセス権限を自動的に付与します。 キュー・マネージャーのデータ・ファイルとログ・ファイルのために別のディレクトリーを指定してキュー・マネージャーを作成する場合は、それらのディレクトリーに対するフル・コントロール権限を構成する必要があります。

この例では、ディレクトリー *c:\wmq* にローカル mqm グループ許可に与えることによって、データ・ファイルとログ・ファイルに対するフル・コントロール権限をキュー・マネージャーに与えます。

crtmqm コマンドを実行して、ワークステーションが IBM MQ サービスを使用し始める時点で自動的に始動するキュー・マネージャーを作成します。

この作業は 1 つの例にすぎません。具体的な値を使用していますが、変更することも可能です。 変更できる値は、斜体で示しています。 作業の最終段階で、すべての変更を削除する手順を実行してください。

手順

1. コマンド・プロンプトを開きます。
2. 次のコマンドを入力します。

```
md c:\wmq\data, c:\wmq\logs
```

3. ローカル mqm グループに読み取りおよび書き込みアクセスを許可するように、ディレクトリーに許可を設定します。

```
cacls c:\wmq/T /E /G mqm:F
```

システム応答:

```
processed dir: c:\wmq
processed dir: c:\wmq\data
processed dir: c:\wmq\logs
```

4. オプション: ローカル mqm グループのメンバーになっているユーザー ID に切り替えます。

この後も管理者として作業を続けてかまいませんが、実際の実動構成では、権限を限定したユーザー ID で作業を進めるようにしてください。 ユーザー ID は、少なくともローカル mqm グループのメンバーでなければなりません。

IBM MQ インストールがドメインの一部として構成されている場合は、そのユーザー ID を Domain mqm グループのメンバーにします。 "IBM MQ の準備" ウィザードは、Domain mqm グローバル・グループをローカル mqm グループのメンバーにするため、ユーザー ID を直接ローカル mqm グループのメンバーにする必要はありません。

5. キュー・マネージャーを作成します。

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

システム応答:

```
IBM MQ queue manager created.
Directory 'c:\wmq\data\QMGR' created.
The queue manager is associated with installation '1'
Creating or replacing default objects for queue manager 'QMGR'
Default objects statistics : 74 created. 0 replaced.
Completing setup.
Setup completed.
```

6. キュー・マネージャーによって、作成されたディレクトリーが *c:\wmq* ディレクトリー内にあることを確認します。

```
dir c:\wmq/D /B /S
```

7. ローカル mqm グループに対する読み取りと書き込み、またはフル・コントロール権限がファイルに付与されていることを確認します。

```
cacls c:\wmq\*.*
```

次のタスク

キューにメッセージを書き込み、キューからメッセージを取り出して、キュー・マネージャーをテストします。

1. キュー・マネージャーを始動します。

```
strmqm QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation '1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. テスト・キューを作成します。

```
echo define qlocal(QTEST) | runmqsc QMGR
```

システム応答:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QMGR.


1 : define qlocal(QTEST)
AMQ8006: IBM MQ queue created.
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

3. サンプル・プログラム **amqsput** を使用してテスト・メッセージを書き込みます。

   ```
   echo 'A test message' | amqsput QTEST QMGR
   ```

   システム応答:

   ```
   Sample AMQSPUT0 start
   target queue is QTEST
   Sample AMQSPUT0 end
   ```

4. サンプル・プログラム **amqsget** を使用してテスト・メッセージを取り出します。

   ```
   amqsget QTEST QMGR
   ```

   システム応答:

   ```
   Sample AMQSGET0 start
   message A test message
   Wait 15 seconds ...
   no more messages
   Sample AMQSGET0 end
   ```

5. キュー・マネージャーを停止させます。

   ```
   endmqm -i QMGR
   ```

   システム応答:

   ```
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ended.
   ```

6. キュー・マネージャーを削除してください。

   ```
   dltmqm QMGR
   ```

   システム応答:

   ```
   IBM MQ queue manager 'QMGR' deleted.
   ```

7. 作成したディレクトリーを削除します。

   ヒント: /Q オプションをコマンドに追加して、各ファイルまたはディレクトリーを削除するためのプロンプトがコマンドに表示されないようにします。

   ```
   del /F /S C:\wmq\*.*
   rmdir /S C:\wmq
   ```

関連概念

508 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』
Windows で複数インスタンス・キュー・マネージャーを実行する場合は、データとログを共有する必要があります。 その共有フォルダーは、それぞれのサーバーやワークステーションで稼働するキュー・マネージャーのすべてのインスタンスからアクセスできる状態でなければなりません。 キュー・マネージャーと共有フォルダーを 1 つの Windows ドメインの中で構成してください。 キュー・マネージャーは、ドメイン・ワークステーションでも、ドメイン・サーバーでも、ドメイン・コントローラーでも実行できます。

関連タスク

Windows 代替ローカル・セキュリティー・グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りと書き込み

この作業では、**crtmqm** コマンドで -a フラグを使用する方法を確認します。 このフラグによって、ログ・ファイルとデータ・ファイルに対するアクセス権をキュー・マネージャーに与えるための代替ローカル・セキュリティー・グループを指定します。

523 ページの『代替グローバル・セキュリティー・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み』
510 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 (Windows)』

Windows 代替ローカル・セキュリティー・グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りと書き込み
この作業では、**crtmqm** コマンドで -a フラグを使用する方法を確認します。 このフラグによって、ログ・ファイルとデータ・ファイルに対するアクセス権をキュー・マネージャーに与えるための代替ローカル・セキュリティー・グループを指定します。

始める前に

1. IBM MQ for Windows をプライマリー・インストールとしてインストールします。
2. Prepare IBM MQ Wizard を実行します。

   詳しくは、 Prepare IBM MQ Wizard を使用した IBM MQ の構成を参照してください。

   この作業では、ローカル・ユーザー ID またはドメイン・ユーザー ID のどちらかでインストール環境を実行するように構成します。 最終的に、508 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』のすべての作業を完了するには、ドメインの一部としてインストール環境を構成する必要があります。
3. この作業の最初の部分を実行するために、管理者権限でログオンします。

このタスクについて

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。 これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。 さらに、508 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』の作業も必要です。

Windows では、IBM MQ for Windows のデフォルトのデータ・パスとログ・パスを任意のディレクトリーに作成できます。 インストールおよび構成ウィザードは、ローカル mqm グループ、およびキュー・マネージャー・プロセスを実行するユーザー ID に、ディレクトリーへのアクセス権限を自動的に付与します。 キュー・マネージャーのデータ・ファイルとログ・ファイルのために別のディレクトリーを指定してキュー・マネージャーを作成する場合は、それらのディレクトリーに対するフル・コントロール権限を構成する必要があります。

この例では、キュー・マネージャーのために、対象のディレクトリーに対するフル・コントロール権限を持った代替セキュリティー・ローカル・グループを設定します。 その代替セキュリティー・グループによって、対象のディレクトリーに含まれているファイルを管理する権限をキュー・マネージャーに与えます。 代替セキュリティー・グループの主な目的は、代替セキュリティー・グローバル・グループに権限を与えることです。 複数インスタンス・キュー・マネージャーをセットアップするときには、代替セキュリティー・グローバル・グループを使用します。 この例では、代替セキュリティー・グループの使い方に習熟するためにローカル・グループを構成するだけにして、IBM MQ をドメインにインストールする作業は行いません。 通常、代替セキュリティー・グループとしてローカル・グループを構成することはありません。

crtmqm コマンドを実行して、ワークステーションが IBM MQ サービスを使用し始める時点で自動的に始動するキュー・マネージャーを作成します。

この作業は 1 つの例にすぎません。具体的な値を使用していますが、変更することも可能です。 変更できる値は、斜体で示しています。 作業の最終段階で、すべての変更を削除する手順を実行してください。

手順

1. 代替セキュリティー・グループをセットアップします。

   通常、代替セキュリティー・グループは、ドメイン・グループです。 この例では、ローカル代替セキュリティー・グループを使用するキュー・マネージャーを作成します。 ドメインの一部ではない IBM MQ のインストール環境に関する作業は、ローカル代替セキュリティー・グループでも実行できます。

   a) **lusrmgr.msc** コマンドを実行して、「ローカル ユーザーとグループ」ウィンドウを開きます。

   b)「**グループ**」 > 「**新しいグループ...**」を右クリックします。

   c)「**グループ名**」 フィールドに *altmqm* と入力し、 「**作成**」 > 「**閉じる**」をクリックします。

   d) IBM MQ サービスを実行するユーザー ID を識別します。

      i) **開始 > 実行 ...**をクリックし、 services.msc と入力し、 「**OK**」をクリックします。

      ii) サービスのリストから IBM MQ サービスをクリックし、「ログオン」タブをクリックします。

      iii) ユーザー ID を記録し、「サービス」エクスプローラーを閉じます。

   e) IBM MQ サービスを実行するユーザー ID を *altmqm* グループに追加します。 さらに、キュー・マネージャーを作成して対話式に実行するためにログオン時に使用するユーザー ID も追加します。

      Windows は、キュー・マネージャーのプロセスを実行するユーザー ID の権限を検査することによって、データとログのディレクトリーに対するアクセス権がキュー・マネージャーにあるかどうかを確認します。 ユーザー ID は、ディレクトリーを許可した *altmqm* グループのグローバル・グループを介して、直接的または間接的にメンバーでなければなりません。
      IBM MQ ドメインの一部としてインストールされ、510 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 (Windows)』でタスクを実行する場合、513 ページの『Active Directory と DNS のドメインの作成 (Windows)』で作成されたドメインユーザー ID は、*wmquser1* と *wmquser2* で作成されたドメインユーザー ID です。
      ドメインの一部としてキュー・マネージャーをインストールしなかった場合、IBM MQ サービスを実行するデフォルトのローカル・ユーザー ID は MUSR_MQADMIN です。 管理者権限なしでタスクを実行する場合は、ローカル mqm グループのメンバーであるユーザーを作成します。
      以下の手順で *wmquser1* と *wmquser2* を *altmqm* に追加します。 構成内容が異なる場合は、ここで取り上げるユーザー ID とグループの代わりに該当する名前を使用してください。

      i) グループのリストで、**altmqm > プロパティー > 追加 ...** を右クリックします。

      ii)「ユーザー、コンピューター、またはグループの選択」ウィンドウで *wmquser1 ; wmquser2* と入力し、 「**名前の確認**」をクリックします。

      iii)「Windows セキュリティ」ウィンドウにドメイン管理者の名前とパスワードを入力し、「**OK**」 > 「**OK**」 > 「**適用**」 > 「**OK**」をクリックします。

2. コマンド・プロンプトを開きます。

3. IBM MQ サービスを再始動します。

   サービスを実行するユーザー ID で、ここで構成した追加のセキュリティー資格情報を有効にするには、サービスを再始動する必要があります。

   次のコマンドを入力します。

   ```
   endmqsvc
   strmqsvc
   ```

   システム応答:

   ```
   5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
   The MQ service for installation 'Installation1' ended successfully.
   ```

   および:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
The MQ service for installation 'Installation1' started successfully.
```

4. 次のコマンドを入力します。

```
md c:\wmq\data, c:\wmq\logs
```

5. ローカル・ユーザーに *user* 読み取りおよび書き込みアクセスを許可するように、ディレクトリーの許可を設定します。

```
cacls c:\wmq/T /E /G altmqm:F
```

システム応答:

```
processed dir: c:\wmq
processed dir: c:\wmq\data
processed dir: c:\wmq\logs
```

6. オプション: ローカル mqm グループのメンバーになっているユーザー ID に切り替えます。

この後も管理者として作業を続けてかまいませんが、実際の実動構成では、権限を限定したユーザー ID で作業を進めるようにしてください。 ユーザー ID は、少なくともローカル mqm グループのメンバーでなければなりません。
IBM MQ インストールがドメインの一部として構成されている場合は、そのユーザー ID を Domain mqm グループのメンバーにします。 "IBM MQ の準備" ウィザードは、Domain mqm グローバル・グループをローカル mqm グループのメンバーにするため、ユーザー ID を直接ローカル mqm グループのメンバーにする必要はありません。

7. キュー・マネージャーを作成します。

```
crtmqm -a altmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

システム応答:

```
IBM MQ queue manager created.
Directory 'c:\wmq1\data\QMGR' created.
The queue manager is associated with installation '1'
Creating or replacing default objects for queue manager 'QMGR'
Default objects statistics : 74 created. 0 replaced.
Completing setup.
Setup completed.
```

8. キュー・マネージャーによって、作成されたディレクトリーが *c:\wmq* ディレクトリー内にあることを確認します。

```
dir c:\wmq/D /B /S
```

9. ローカル mqm グループに対する読み取りと書き込み、またはフル・コントロール権限がファイルに付与されていることを確認します。

```
cacls c:\wmq\*.*
```

次のタスク

キューにメッセージを書き込み、キューからメッセージを取り出して、キュー・マネージャーをテストします。

1. キュー・マネージャーを始動します。

```
strmqm QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation '1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. テスト・キューを作成します。

```
echo define qlocal(QTEST) | runmqsc QMGR
```

システム応答:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QMGR.


1 : define qlocal(QTEST)
AMQ8006: IBM MQ queue created.
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

3. サンプル・プログラム **amqsput** を使用してテスト・メッセージを書き込みます。

```
echo 'A test message' | amqsput QTEST QMGR
```

システム応答:

```
Sample AMQSPUT0 start
target queue is QTEST
Sample AMQSPUT0 end
```

4. サンプル・プログラム **amqsget** を使用してテスト・メッセージを取り出します。

```
amqsget QTEST QMGR
```

システム応答:

```
Sample AMQSGET0 start
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGET0 end
```

5. キュー・マネージャーを停止させます。

```
endmqm -i QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended.
```

6. キュー・マネージャーを削除してください。

```
dltmqm QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' deleted.
```

7. 作成したディレクトリーを削除します。

   ヒント: /Q オプションをコマンドに追加して、各ファイルまたはディレクトリーを削除するためのプロンプトがコマンドに表示されないようにします。

```
del /F /S C:\wmq\*.*
rmdir /S C:\wmq
```

関連タスク

Windows ローカル mqm グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りおよび書き込み

この作業では、キュー・マネージャーを作成し、そのデータ・ファイルとログ・ファイルを任意のディレクトリーに格納する方法を確認します。 ファイルへのアクセスは、ローカル mqm グループによって保護されます。 ディレクトリーは共有しません。

Linux *Linux* での複数インスタンス・キュー・マネージャーの作成

Linux で複数インスタンス・キュー・マネージャーをセットアップする方法を示す例。 セットアップは、関係した概念を例示するために小規模なものとしています。 このステップは、他の UNIX プラットフォームで異なります。

始める前に

この例は、 Red Hat Enterprise Linux 5 と Windows 7 に基づいています。 IBM MQ の現行バージョンでは、これら両製品の最新バージョンが必要です。 ただし、セットアップの手順は、ここで説明されている手順とほぼ同じです。

このタスクについて

この例は、3GBRAM が稼働している 2GHz のノートブック・コンピューター(Windows7ServicePack1)にセットアップされています。 VMware の仮想マシンである Server1 と Server2 の 2 つが、640MB のイメージで Red Hat Enterprise Linux 5 を実行します。 サーバー 1 は、ネットワーク・ファイル・システム (NFS)、キュー・マネージャー・ログおよび HA インスタンスをホストします。 通常の手法では、NFS サーバーがキュー・マネージャー・インスタンスのいずれかもホストすることはありません。これは、例を単純化するためのものです。 サーバー 2 は、サーバー 1 のキュー・マネージャー・ログをスタンバイ・インスタンスでマウントします。 IBM MQ MQI クライアントは、 VMware イメージに追加で 400MB の容量を確保してインストールされ、 Windows 7 Service Pack 1 を実行し、サンプルの高可用性アプリケーションを実行します。 すべての仮想マシンは、セキュリティー上の理由から、VMware ホスト専用ネットワークの一部として構成されています。

注: NFS サーバーにはキュー・マネージャー・データのみを書き込む必要があります。 NFS では、mount コマンドで次の 3 つのオプションを使用して、システムを保護します。

- **noexec**
  このオプションを使用すると、バイナリー・ファイルを NFS 上で実行できなくなります。 こうすることで、リモート・ユーザーがシステム上で望ましくないコードを実行できないようにします。
- **nosuid**
  このオプションを使用すると、セット・ユーザー ID ビットとセット・グループ ID ビットを使用できなくなります。 こうすることで、リモート・ユーザーが上位の特権を取得できないようにします。
- **nodev**
  このオプションを使用すると、文字およびブロック特殊装置が使用または定義できなくなります。 こうすることで、リモート・ユーザーが chroot ジェイルから出られないようにします。

手順

1. root としてログインします。
2. IBM MQ のインストール- 概要を読み取り、IBM MQ をインストールし、mqm ユーザーとグループを作成し、/var/mqm を定義するための適切なリンクに従います。
3. 共有ファイル・システムの動作の検証の作業を実行し、ファイル・システムが複数インスタンス・キュー・マネージャーをサポートしていることを確認します。
4. Server1 で、以下の手順を実行します。
   a. 共用される共通フォルダー/MQHA に、ログ・ディレクトリーおよびデータ・ディレクトリーを作成します。 以下に例を示します。
      i) **mkdir** */MQHA*
      ii) **mkdir** */MQHA/logs*
      iii) **mkdir** */MQHA/qmgrs*
5. Server2 で、以下の手順を実行します。
   a. 共有ファイル・システムをマウントするためのフォルダー /MQHA を作成します。 このパスをサーバー 1 と同じにしておきます。 以下に例を示します。
      i) **mkdir** */MQHA*
6. MQHA ディレクトリーがユーザーおよびグループ mqm により所有されており、ユーザーおよびグループのアクセス権が rwx に設定されていることを確認します。 例えば、**ls -al** は drwxrwxr-x mqm mqm 4096 Nov 27 14:38 MQDATA を表示します。
   a. **chown -R** mqm:mqm */MQHA*
   b. **chmod -R** ug+rwx */MQHA*
7. コマンド **crtmqm -ld** */MQHA/logs* **-md** */MQHA/qmgrs QM1* を入力して、キュー・マネージャーを作成します。
8. 追加 [2]/MQHA *(rw,sync,no_wdelay,fsid=0) から /etc/exports へ
9. Server1 で、以下の手順を実行します。
   a. NFS デーモンを開始します。 */etc/init.d/* **nfs** start
   b. 以下のように、キュー・マネージャー構成の詳細をサーバー 1 からコピーします。

      ```
      dspmqinf -o command QM1
      ```

      そして、結果をクリップボードにコピーします。

      ```
      addmqinf -s QueueManager
      -v Name=QM1
      -v Directory=QM1
      -v Prefix=/var/mqm
      -v DataPath=/MQHA/qmgrs/QM1
      ```
10. Server2 で、以下の手順を実行します。
    a. コマンド **mount -t** nfs4 **-o** hard,intr *Server1:/ /MQHA* を入力して、エクスポートされたファイル・システム /MQHA をマウントします。
    b. 以下のように、キュー・マネージャー構成コマンドをサーバー 2 に貼り付けます。

       ```
       addmqinf -s QueueManager
       -v Name=QM1
       -v Directory=QM1
       -v Prefix=/var/mqm
       -v DataPath=/MQHA/qmgrs/QM1
       ```

[2] '*' では、読み取り/書き込み用に一つのマウント/MQHA に到達できるすべてのマシンを使用できます。 実動マシンでのアクセスを制限します。

11. **-x** パラメーターを使用して、キュー・マネージャー・インスタンスをいずれかの順序で開始します (**strmqm -x** QM1)。

    キュー・マネージャー・インスタンスを開始するために使用するコマンドは、 **addmqinf** コマンドと同じ IBM MQ インストール済み環境から発行する必要があります。 異なるインストール済み環境からキュー・マネージャーを開始および停止するには、まず **setmqm** コマンドを使用してキュー・マネージャーと関連するインストール済み環境を設定する必要があります。 詳細については、setmqm を参照してください。

Linux *Linux* での複数インスタンス・キュー・マネージャーの検査

複数インスタンス・キュー・マネージャー構成を検査するには、サンプル・プログラム **amqsghac**、**amqsphac**、および **amqsmhac** を使用します。 このトピックでは、Linux Red Hat Enterprise 5 上で複数インスタンス・キュー・マネージャーの構成を検査するためのサンプル構成を記載しています。

高可用性サンプル・プログラムは、クライアントの自動再接続を使用します。 接続されたキュー・マネージャーに障害が起きた場合、クライアントは同じキュー・マネージャー・グループ内のキュー・マネージャーへの再接続を試行します。 サンプルの説明、高可用性のサンプル・プログラムでは、単純にするために単一インスタンス・キュー・マネージャーを使用してクライアントの再接続を説明しています。 複数インスタンス・キュー・マネージャーがある同じサンプルを使用して、複数インスタンス・キュー・マネージャー構成を検査することができます。

サンプルでは、549 ページの『Linux での複数インスタンス・キュー・マネージャーの作成』で説明されている複数インスタンス構成を使用しています。 この構成を使用して、複数インスタンス・キュー・マネージャーがスタンバイ・インスタンスに切り替わることを検査します。 **endmqm** コマンドを使用してキュー・マネージャーを停止し、-s 切り替えオプションを使用します。 クライアント・プログラムは、新規キュー・マネージャー・インスタンスに再接続して、少しの遅延の後に新規インスタンスの処理を続行します。

この例では、クライアントは Windows 7 Service Pack 1 システム上で実行されます。 システムは、複数インスタンス・キュー・マネージャーを実行する 2 つの VMware Linux サーバーをホスティングします。

IBM MQ Explorer を使用したフェイルオーバーの検査

フェイルオーバーを検査するためにサンプル・アプリケーションを使用する前に、IBM MQ Explorer を各サーバー上で実行します。「**リモート・キュー・マネージャーの追加**」>「**複数インスタンス・キュー・マネージャーへの直接接続**」ウィザードを使用して、両方のキュー・マネージャー・インスタンスを各エクスプローラーに追加します。 両方のインスタンスをスタンバイを許可して実行しておきます。 アクティブ・インスタンスがある VMware イメージを実行中のウィンドウを閉じて仮想的にサーバーの電源をオフにするか、またはスタンバイ・インスタンスへの切り替えを許可してアクティブ・インスタンスを停止します。

注 : サーバーの電源をオフにする場合は、それが/MQHA をホスティングしているものではないことを確認してください！

注 :「**キュー・マネージャーの停止**」ダイアログで「**スタンバイ・インスタンスへの切り替えを許可**」オプションが表示されない場合があります。 このオプションがないのは、キュー・マネージャーが単一インスタンス・キュー・マネージャーとして実行されているためです。 それを開始したときに、「**スタンバイ・インスタンスの許可**」オプションを指定しませんでした。 キュー・マネージャーの停止要求が拒否された場合は、「**詳細**」ウィンドウを参照してください。実行中のスタンバイ・インスタンスがないことが原因である可能性があります。

サンプル・プログラムを使用したフェイルオーバーの検査

アクティブ・インスタンスを実行するサーバーの選択

MQHA ディレクトリーまたはファイル・システムをホストするサーバーの一つを選択した可能性があります。 アクティブ・サーバーを実行している VMware ウィンドウを閉じて、フェイルオーバーをテストする場合は、MQHA をホスティングしているものではないことを確認してください！

アクティブ・キュー・マネージャー・インスタンスを実行するサーバー上

注 : SVRCONN チャネルを、MCAUSER を mqm に設定して実行することは、サンプルの構成ステップの数を減らすことに役立ちます。 別のユーザー ID を選択し、システムがサンプルで使用されているものとは異なるセットアップとなっている場合、アクセス権問題が発生する可能性があります。 mqm を公開

システム上で MCAUSER として使用しないでください。これはセキュリティーが大きく危険に晒される可能性があります。

1. *ipaddr1* および *ipaddr2* を変更し、以下のコマンドを/MQHA/hasamples.tst.に保管する

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN)  TRPTYPE(TCP) +
MCAUSER('mqm') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(' ipaddr1 (1414), ipaddr2
(1414)') QMNAME(QM1) REPLACE
START  CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
START  LISTENER(LISTENER.TCP)
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

2. パス/MQHA を指定して、端末ウィンドウを開き、以下のコマンドを実行します:

```
runmqsc -m QM1 < hasamples.tst
```

3. **runmqsc** コマンドの出力を検査することで、リスナーが実行中であり、キュー・マネージャー制御があることを確認します。

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

   または、IBM MQ Explorer を使用して、TCPIP リスナーが実行中で、Control = Queue Manager があることを確認します。

クライアント上

1. クライアント接続テーブル AMQCLCHL.TAB を、サーバー上の/MQHA/qmgrs/QM1.000/@ipcc からクライアント上の C:\にコピーします。
2. パス C:\ を指定してコマンド・プロンプトを開き、クライアント・チャネル定義テーブル (CCDT) を指すように環境変数 MQCHLLIB を設定します。

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

3. コマンド・プロンプトで、以下のコマンドを入力します。

```
start amqsghac TARGET QM1
start amqsmhac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

アクティブ・キュー・マネージャー・インスタンスを実行するサーバー上

1. 以下のいずれかを実行します。
   - アクティブ・サーバー・インスタンスがある VMware イメージを実行するウィンドウを閉じます。
   - IBM MQ Explorer を使用して、アクティブ・キュー・マネージャー・インスタンスを停止して、スタンバイ・インスタンスへの切り替えを許可し、再接続可能クライアントに再接続を指示します。
2. 最終的には 3 つのクライアントで接続が切断していることが検出され、次いで再接続されます。 この構成では、サーバー・ウィンドウを閉じた場合、3 つの接続がすべて再確立されるには約 7 分かかります。 他の接続よりも速く再確立される接続もあります。

結果

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start

17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsghac TARGET QM1
Sample AMQSGHAC start
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

Multi 複数インスタンス・キュー・マネージャーの削除

Multiplatforms で複数インスタンス・キュー・マネージャーを完全に削除するには、**dltmqm** コマンドを使用してキュー・マネージャーを削除してから、**rmvmqinf** コマンドか **dltmqm** コマンドを使用して他のサーバーからインスタンスを削除します。

dltmqm コマンドを実行して、他のサーバー (キュー・マネージャーが定義されているすべてのサーバー) 上で定義されているインスタンスがあるキュー・マネージャーを削除します。 **dltmqm** コマンドは、それを作成した同じサーバー上で実行する必要はありません。 次いで、**rmvmqinf** または **dltmqm** コマンドを、キュー・マネージャーの定義がある他のすべてのサーバー上で実行します。

キュー・マネージャーは停止しているときにのみ削除できます。 削除時には実行中のインスタンスはなく、厳密に言えばキュー・マネージャーは単一または複数のどちらのインスタンス・キュー・マネージャーでもありません。それはリモート・シェアにキュー・マネージャー・データとログを持つ単なるキュー・マネージャーです。 キュー・マネージャーの削除時には、そのキュー・マネージャー・データとログは削除され、キュー・マネージャー・スタンザは **dltmqm** コマンドを発行したサーバー上の `mqs.ini` ファイルから除去されます。 キュー・マネージャーを削除するときには、キュー・マネージャー・データとログを含むネットワーク・シェアへのアクセス権限が必要です。

以前にキュー・マネージャーのインスタンスが作成されている他のサーバーでは、それらのサーバー上の `mqs.ini` ファイルにも項目があります。 各サーバーに順にアクセスし、**rmvmqinf** *Queue manager stanza name* コマンドを実行して、キュー・マネージャー・スタンザを除去する必要があります。

Linux AIX AIX and Linux システム上では、mqs.ini 共通ファイルをネットワーク・ストレージ内に置いていて、各サーバー上で AMQ_MQS_INI_LOCATION 環境変数を設定してすべてのサーバーからそれを参照していた場合は、更新するファイルは mqs.ini ファイル 1 つしかないので、そのサーバーの 1 つだけからキュー・マネージャーを削除すれば済みます。

例

最初のサーバー

dltmqm QM1

インスタンスが定義されている他のサーバー

rmvmqinf QM1 、または

dltmqm QM1

Multi 複数インスタンス・キュー・マネージャーの開始および停止

Multiplatforms で単一インスタンスまたは複数インスタンスのキュー・マネージャーとして構成されたキュー・マネージャーの開始と停止。

1 組みのサーバー上に複数インスタンス・キュー・マネージャーを定義した場合は、いずれかのサーバー上で、キュー・マネージャーを単一インスタンス・キュー・マネージャー、または複数インスタンス・キュー・マネージャーのいずれかとして実行することができます。

複数インスタンス・キュー・マネージャーを実行するには、**strmqm** -x *QM1* コマンドを使用して、一方のサーバー上でキュー・マネージャーを開始します。-x は、インスタンスのフェイルオーバーを許可するオプションです。 これがアクティブ・インスタンス になります。 同じ **strmqm** -x *QM1* コマンドを使用して、もう一方のサーバー上でスタンバイ・インスタンスを開始します。-x は、インスタンスのスタンバイとしての開始を許可するオプションです。

キュー・マネージャーは現在、すべての要求を処理している 1 つのアクティブ・インスタンスと、アクティブ・インスタンスに障害が起きた場合の引き継ぎに備えている 1 つのスタンバイ・インスタンスがある状態で実行されています。 アクティブ・インスタンスには、キュー・マネージャーのデータおよびログに対する排他的アクセス権限が付与されています。 スタンバイ・インスタンスは、キュー・マネージャーのデータおよびログに対する排他的アクセス権限が付与されるのを待っています。 排他的アクセス権限が付与されたスタンバイ・インスタンスは、アクティブ・インスタンスになります。

アクティブ・インスタンス上で **endmqm** -s コマンドを発行することにより、手動でスタンバイ・インスタンスに制御を切り替えることもできます。 **endmqm** -s コマンドは、スタンバイ・インスタンスをシャットダウンすることなく、アクティブ・インスタンスをシャットダウンします。 キュー・マネージャーのデータおよびログに対する排他的アクセス・ロックが解除され、スタンバイ・インスタンスに引き継がれます。

異なるサーバー上に複数のインスタンスを置いて構成したキュー・マネージャーを、単一インスタンス・キュー・マネージャーとして開始および停止することもできます。 **strmqm** コマンドで -x オプションを使用せずにキュー・マネージャーを開始した場合、他のマシン上に構成されたキュー・マネージャーのインスタンスをスタンバイ・インスタンスとして開始できなくなります。 別のインスタンスを開始しようとすると、そのキュー・マネージャー・インスタンスをスタンバイ・インスタンスとして実行することは認められていないという応答を受け取ります。

endmqm コマンドで -s オプションを使用せずに複数インスタンス・キュー・マネージャーのアクティブ・インスタンスを停止すると、アクティブ・インスタンスとスタンバイ・インスタンスの両方が停止します。-x オプションを指定した **endmqm** コマンドを実行してスタンバイ・インスタンスを停止すると、そのインスタンスのスタンバイ状態は解除されますが、アクティブ・インスタンスの実行は続行します。 -x オプションを使用していない **endmqm** をスタンバイ・インスタンスに対して発行することはできません。

同時に実行できるのは、アクティブ・インスタンスとスタンバイ・インスタンスの、2 つのキュー・マネージャー・インスタンスだけです。 2 つのインスタンスを同時に開始した場合、IBM MQ は、どちらがアクティブ・インスタンスになるかを制御しません。どちらがアクティブ・インスタンスになるかはネットワーク・ファイル・システムにより決定されます。 キュー・マネージャーのデータに対する排他的アクセス権限を最初に獲得したインスタンスが、アクティブ・インスタンスになります。

注 : 失敗したキュー・マネージャーを再始動する前に、アプリケーションをキュー・マネージャーのそのインスタンスから切断する必要があります。 そうしない場合、キュー・マネージャーは正しく再始動されない場合があります。

Multi 共有ファイル ・システム

Mulitplatforms の場合、複数インスタンス・キュー・マネージャーは、ネットワーク・ファイル・システムを使用してキュー・マネージャーのインスタンスを管理します。

複数インスタンス・キュー・マネージャーは、ファイル・システム・ロック、およびキュー・マネージャーのデータおよびログの共有を組み合わせて使用して、フェイルオーバーを自動化します。 キュー・マネージャーのインスタンスのうち、キュー・マネージャーの共有データおよびログに対する排他的アクセス権限を持つことができるのは 1 つのみです。 アクセス権限を取得したインスタンスが、アクティブ・インスタンスになります。 排他的アクセス権限を取得できなかったもう一方のインスタンスは、キュー・マネージャーのデータおよびログが使用可能になるまで、スタンバイ・インスタンスとして待機します。

ネットワーク・ファイル・システムは、キュー・マネージャーのアクティブ・インスタンス用に保持しているロックの解除を担当します。 アクティブ・インスタンスに何らかの障害が起こった場合、ネットワーク・ファイル・システムはアクティブ・インスタンス用に保持しているロックを解除します。 排他ロックが解除されるとすぐに、ロックを待機していたスタンバイ・キュー・マネージャーがロックを獲得しようとします。 獲得できた場合は、それがアクティブ・インスタンスになり、ファイル共有システム上のキュー・マネージャーのデータおよびログに対する排他的アクセス権限を保持します。 その後、このインスタンスは開始段階に進みます。

関連トピックのファイル・システム・サポートの計画では、セットアップ方法、およびファイル・システムが複数インスタンス・キュー・マネージャーをサポートするかどうかを確認する方法を説明します。

複数インスタンス・キュー・マネージャーは、ファイル・システムを障害から保護するわけではありません。 データを保護する方法はいくつかあります。

- 新磁気ディスク制御機構 (RAID) などの高信頼ストレージに投資して、それらをネットワーク回復力のあるネットワーク・ファイル・システム内に組み込む。
- IBM MQ のリニア・ログを代替メディアにバックアップして、1 次ログ・メディアに障害が起きた場合に、その代替メディア上のログを使用してリカバリーする。 このプロセスは、バックアップ・キュー・マネージャーを使用して管理できます。

Multi 複数のキュー・マネージャー・インスタンス

複数インスタンス・キュー・マネージャーは、障害発生後にスタンバイ・キュー・マネージャー・インスタンスを使用してキュー・マネージャーの可用性を取り戻すことができるため、回復力があります。

キュー・マネージャー・インスタンスの複製は、キュー・マネージャー・プロセスの可用性を向上させる上で非常に有効な方法です。 単に説明を目的とした単純な可用性モデルの場合で考えると、キュー・マネージャーのインスタンスが 1 つである場合の信頼性が 99% (年間の累積ダウン時間が 3.65 日) であるとき、キュー・マネージャーのインスタンスをもう 1 つ追加することで、可用性が 99.99% (年間の累積ダウン時間が約 1 時間) に上昇します。

このモデルは、実際的な可用性の数値見積もりを得るには単純過ぎます。 可用性を現実的にモデル化するには、平均故障間隔 (MTBF) および平均修理時間 (MTTR) の統計、および故障間隔と修理時間の確率分布を収集する必要があります。

複数インスタンス・キュー・マネージャーという用語は、キュー・マネージャーのデータおよびログを共有する、キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスの組み合わせを意味しています。 複数インスタンス・キュー・マネージャーでは、一方のサーバーにあるキュー・マネージャー・インスタンスはアクティブにし、もう一方のサーバーにあるもう一方キュー・マネージャー・インスタンスはスタンバイにすることで、アクティブ・インスタンスで障害が起きた場合に自動的に引き継ぎが行われるようにして、キュー・マネージャー・プロセスを障害から保護します。

Multi フェイルオーバーまたは切り替え

スタンバイ・キュー・マネージャー・インスタンスは、要求 (切り替え) 時またはアクティブ・インスタンスで障害が起きたとき (フェイルオーバー) に、アクティブ・インスタンスの処理を引き継ぎます。

- 切り替え は、アクティブ・キュー・マネージャー・インスタンスに対して発行された **endmqm** -s コマンドに応答して、スタンバイ・インスタンスが開始されたときに実行されます。 **endmqm** のパラメーター -c、-i、または -p を指定することで、キュー・マネージャーをどれほど急速に停止するかを制御できます。

  注: 切り替えは、スタンバイ・キュー・マネージャー・インスタンスが既に開始されている場合にのみ実行されます。 **endmqm** -s コマンドは、アクティブ・キュー・マネージャーのロックを解除し、切り替えを許可します。ただしスタンバイ・キュー・マネージャー・インスタンスの開始は行いません。
- フェイルオーバー は、アクティブ・インスタンスが突然停止した可能性があるために、そのインスタンスがキュー・マネージャー・データに対して保持していたロックが解除されたときに (つまり、**endmqm** コマンドの発行なしに)、実行されます。

スタンバイ・インスタンスは、アクティブ・インスタンスとして処理を引き継ぐと、キュー・マネージャーのエラー・ログにメッセージを書き込みます。

再接続可能クライアントは、キュー・マネージャーの障害発生時または切り替え時に自動的に再接続します。 クライアント再接続を要求するために **endmqm** コマンドに -r フラグを含める必要はありません。 IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

フェイルオーバーが発生し、スタンバイ・インスタンスがアクティブになったにもかかわらず、失敗したインスタンスを再開できない場合は、失敗したインスタンスにローカルで接続しているアプリケーションがその失敗したインスタンスから切断されていないかどうかを確認してください。

失敗したインスタンスを再開できるようにするには、ローカルで接続しているアプリケーションを終了するか、失敗したキュー・マネージャー・インスタンスから切断する必要があります。 共有バインディングを使用しており (デフォルトの設定です)、失敗したインスタンスへの接続を保持しているローカル接続のアプリケーションが存在すると、そのインスタンスを再開する妨げになります。

ローカル接続のアプリケーションを終了できない場合、あるいはローカル・キュー・マネージャーの障害発生時にそれらのアプリケーションが確実に切断されるようにできない場合は、隔離されたバインディングの使用を検討してください。 隔離されたバインディングを使用するローカル接続のアプリケーションは、切断されなくても、ローカル・キュー・マネージャー・インスタンスの再開を妨げません。

Multi チャネルおよびクライアントの再接続

チャネルおよびクライアントの再接続は、スタンバイ・キュー・マネージャー・インスタンスがアクティブになった後、メッセージ処理を回復させるためには不可欠の作業です。

複数インスタンス・キュー・マネージャー・インスタンスは、異なるネットワーク・アドレスを持つ複数のサーバー上にインストールされています。 すべてのキュー・マネージャー・インスタンスの接続情報を使用して IBM MQ のチャネルおよびクライアントを構成する必要があります。 スタンバイが引き継ぎを行うと、クライアントおよびチャネルは、新規のネットワーク・アドレスにある新たにアクティブになったキュー・マネージャー・インスタンスに自動的に再接続します。 IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

このような設計は、HA-CMP 処理などの高可用性環境の方法とは異なります。 HA-CMP は、クラスターの仮想 IP アドレスを指定して、そのアドレスをアクティブ・サーバーに転送します。 IBM MQ の再接続では、IP アドレスを変更または転送しません。 これはチャネル定義およびクライアント接続で定義したネットワーク・アドレスを使用して再接続することにより機能します。 管理者は、複数インスタンス・キュー・マネージャーのすべてのインスタンスに対して、チャネル定義およびクライアント接続でネットワーク・アドレスを定義する必要があります。 複数インスタンス・キュー・マネージャーに対してネットワーク・アドレスを構成するための最善の方法は、以下の接続に応じて異なります。

キュー・マネージャー・チャネル
チャネルの CONNAME 属性は、接続名のコンマ区切りリストです (例: CONNAME('127.0.0.1(1234),192.0.2.0(4321)'))。 接続は、正常に確立されるまで、この接続リストに指定された順序に従って試行されます。 正常に確立された接続がない限り、チャネルは再接続を試みます。

クラスター・チャネル

一般に、複数インスタンス・キュー・マネージャーがクラスター内で機能するために必要な追加の構成はありません。

キュー・マネージャーがリポジトリー・キュー・マネージャーに接続する場合は、リポジトリーがキュー・マネージャーのネットワーク・アドレスを検出します。 リポジトリーは、キュー・マネージャーの CLUSRCVR チャネルの CONNAME を参照します。 TCPIP では、CONNAME が省略されたりブランクに構成されたりしている場合は、キュー・マネージャーがそれを自動的に設定します。 スタンバイ・インスタンスが引き継ぎを行う場合、その IP アドレスが CONNAME として以前のアクティブ・インスタンスの IP アドレスと置き換わります。

必要な場合、キュー・マネージャー・インスタンスのネットワーク・アドレスのリストを使用して、CONNAME を手動で構成することができます。

クライアント接続
クライアント接続は、接続リストを使用することも、キュー・マネージャー・グループを使用して代替接続を選択することもできます。

フェイルオーバー実行時の再接続には、ある程度の時間がかかります。 スタンバイ・キュー・マネージャーは、始動を完了する必要があります。 障害が起きたキュー・マネージャーに接続されていたクライアントは、接続障害を検出し、新規クライアント接続を開始する必要があります。 新規クライアント接続が、新たにアクティブになったスタンバイ・キュー・マネージャーを選択する場合は、クライアントは同じキュー・マネージャーに再接続されます。

再接続時にクライアントが MQI 呼び出しの途中である場合は、その呼び出しが完了するまで待つ必要があります。

メッセージ・チャネルでバッチ転送中に障害が発生した場合、そのバッチはロールバック後、リスタートされます。

切り替えはフェイルオーバーよりも短時間であり、1 つのキュー・マネージャー・インスタンスを停止して、別の 1 つを開始するだけの時間で済みます。 再生するログ・レコード数がわずかしかないキュー・マネージャーの場合、切り替えにかかる時間は多くても数秒程度です。 フェイルオーバーにかかる時間を見積もるには、障害が検出されるまでの時間を加算する必要があります。 検出にかかる時間は、速ければ 10 秒程度ですが、ネットワークやファイル・システムによっては、数分かかる場合もあります。

Multi アプリケーションの復旧

アプリケーションの復旧は、フェイルオーバーの後に自動で継続されるアプリケーション処理です。 フェイルオーバー後のアプリケーションの復旧は、慎重に設計する必要があります。 一部のアプリケーションでは、フェイルオーバーが実行されたことを認識する必要があります。

アプリケーションの復旧の目的は、短い遅延時間でアプリケーション処理を続行することです。 新しい処理を続行する前に、アプリケーションは障害時に処理していた作業単位をバックアウトして再実行依頼する必要があります。

アプリケーションの復旧における問題は、IBM MQ MQI client とキュー・マネージャーが共有する、キュー・マネージャーに保管されたコンテキストが失われることです。 IBM MQ MQI client はコンテキストの大部分を復元しますが、忠実には復元できない部分があります。 以下の各セクションでは、アプリケーションの復旧に関するいくつかのプロパティー、およびそれらが複数インスタンス・キュー・マネージャーに接続したアプリケーションの復旧にどう影響するかについて説明します。

トランザクション・メッセージング

メッセージ送信の観点からすると、フェイルオーバーによって IBM MQ メッセージングの持続プロパティーが変更されることはありません。 メッセージが持続メッセージで、作業単位内で正しく管理されている場合、フェイルオーバー中にそれらのメッセージが失われることはありません。

トランザクション処理の観点からすると、フェイルオーバー後にトランザクションはバックアウトされるかまたはコミットされます。

コミットされていないトランザクションはロールバックされます。 フェイルオーバーの後に、再接続可能アプリケーションではトランザクションが失敗したことを示す MQRC_BACKED_OUT 理由コードを受け取ります。 次いでトランザクションを再開する必要があります。

コミット済みトランザクションとは、2 フェーズ・コミットの 2 番目のフェーズに達しているトランザクション、または MQCMIT を開始した単一フェーズ (メッセージのみ) トランザクションのことです。

キュー・マネージャーがトランザクション・コーディネーターであり、障害前に MQCMIT がその 2 フェーズ・コミットの第 2 フェーズを開始済みであった場合、トランザクションは正常に完了します。 その完了はキュー・マネージャーの制御下にあり、キュー・マネージャーが再び実行されると続行します。 再接続可能アプリケーションでは、MQCMIT 呼び出しは正常に完了します。

メッセージだけが関係する単一フェーズ・コミットでは、コミット処理を開始したトランザクションは、キュー・マネージャーが再度実行すると、その制御下で正常に完了します。 再接続可能アプリケーションでは、MQCMIT は正常に完了します。

再接続可能クライアントは、トランザクション・コーディネーターとして機能するキュー・マネージャーの制御下では、単一フェーズのトランザクションを使用できます。 拡張トランザクション・クライアントは再接続をサポートしません。 トランザクション・クライアントの接続時に再接続が要求される場合、接続は成功しますが、再接続することはできません。 その接続は、再接続不可として動作します。

アプリケーションの再始動または再開

フェイルオーバーはアプリケーションを中断させます。 障害解決後、アプリケーションは最初から再始動することもできますし、中断後の処理を再開することもできます。 後者の機能のことをクライアントの自動再接続 といいます。 IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

IBM MQ MQI client・アプリケーションでは、クライアントが自動的に再接続されるように接続オプションを設定できます。 該当するオプションは、MQCNO_RECONNECT または MQCNO_RECONNECT_Q_MGR です。 オプションを設定しなければ、クライアントは、自動的に再接続しようとはしません。キュー・マネージャーの障害で、MQRC_CONNECTION_BROKEN がクライアントに返されます。 新しい MQCONN または MQCONNX 呼び出しを発行することによって新しい接続を試行して開始するようにクライアントを設計することもできます。

サーバー・プログラムは再始動する必要があります。キュー・マネージャーが、これらをキュー・マネージャーまたはサーバーで障害が発生した際に処理していたポイントで自動的に再接続することはできません。 複数インスタンス・キュー・マネージャー・インスタンスで障害が発生した場合、IBM MQ サーバー・プログラムは通常、スタンバイ・キュー・マネージャー・インスタンス上では再始動しません。

IBM MQ サーバー・プログラムをスタンバイ・サーバー上で自動的に再始動するには、次の 2 つの方法があります。

1. サーバー・アプリケーションをキュー・マネージャー・サービスとしてパッケージする。 これはスタンバイ・キュー・マネージャーの再始動時に再始動します。
2. 例えば、スタンバイ・キュー・マネージャー・インスタンスの開始時に書き込まれるフェイルオーバーのログ・メッセージなどによってトリガーされる独自のフェイルオーバー・ロジックを作成する。 その後、アプリケーション・インスタンスは、開始後に MQCONN または MQCONNX を呼び出して、キュー・マネージャーへの接続を作成する必要があります。

フェイルオーバーの検出

フェイルオーバーを確実に認識する必要があるアプリケーションもあれば、そうでないアプリケーションもあります。 以下の 2 つの例について考えてみます。

1. メッセージング・チャネルを介してメッセージを取得または受信するメッセージング・アプリケーションでは、通常、そのチャネルの反対側にあるキュー・マネージャーが稼働している必要はありません。したがって、チャネルの反対側にあるキュー・マネージャーがスタンバイ・インスタンスで再始動しても、影響が出る可能性はほとんどありません。
2. IBM MQ MQI client アプリケーションは、1 つのキューからの持続メッセージ入力を処理し、持続メッセージ応答を 1 つの作業単位の一部として別のキューに書き込みます。作業単位を再始動することによって、同期点内で MQPUT、MQGET 、または MQCMIT からの MQRC_BACKED_OUT 理由コードを処理する場合、メッセージは失われません。 さらに、このアプリケーションでは接続障害に対応して何らかの特別な処理を実行する必要もありません。

しかし、2 番目の例で、アプリケーションが MQGET のオプション MQGMO_MSG_UNDER_CURSOR を使用して、処理すべきメッセージを選択するためにキューを参照しているとします。 再接続によってブラウズ・カーソルがリセットされるため、MQGET 呼び出しは正しいメッセージを返しません。 この例の場合、アプ

リケーションは、フェイルオーバーが実行されたことを認識する必要があります。 さらにアプリケーションは、カーソルのあるメッセージに対して別の MQGET を発行する前に、ブラウズ・カーソルを復元する必要があります。

ブラウズ・カーソルの消失は、再接続後にアプリケーション・コンテキストが変化する例の 1 つです。 他のケースは、559 ページの『自動再接続されるクライアントの復旧』に記載されています。

フェイルオーバー後の IBM MQ MQI client・アプリケーションには、3 つの代替設計パターンがあります。 フェイルオーバーを検出する必要がないのは、1 つだけです。

再接続なし

このパターンでは、接続が失敗した場合に、アプリケーションは現行接続上のすべての処理を停止します。 アプリケーションが処理を続行するには、キュー・マネージャーとの新規接続を確立する必要があります。 新規接続での処理を続行するために必要な、すべての状態情報の転送全体を受け持つのは、アプリケーションです。 接続が失われた後にキュー・マネージャーと再接続する既存のクライアント・アプリケーションについては、この記述が当てはまります。

クライアントは、接続が失われた後に次の MQI 呼び出しから MQRC_CONNECTION_BROKEN または MQRC_Q_MGR_NOT_AVAILABLE などの理由コードを受け取ります。 アプリケーションはキュー・ハンドルなど、そのすべての IBM MQ 状態情報を廃棄し、新規の MQCONN 呼び出しまたは MQCONNX 呼び出しを発行して新規接続を確立してから、処理する必要のある IBM MQ オブジェクトを再オープンする必要があります。

デフォルトの MQI の動作では、キュー・マネージャーとの接続が失われた後には、キュー・マネージャーの接続ハンドルは使用不可になります。 デフォルトは、MQCONNX に MQCNO_RECONNECT_DISABLED オプションを設定して、フェイルオーバー後のアプリケーション再接続が行われないようにすることと同じです。

フェイルオーバーへの耐性

フェイルオーバーの影響を受けないアプリケーションを作成します。 場合によっては、注意深くエラー処理を行うだけでフェイルオーバーを処理できます。

再接続の認識

キュー・マネージャーに MQCBT_EVENT_HANDLER イベント・ハンドラーを登録します。 クライアントがサーバーへの再接続の試行を開始したときには MQRC_RECONNECTING、再接続が正常に行われた後には MQRC_RECONNECTED を使用してイベント・ハンドラーに通知が行われます。 ユーザーはその後、クライアント・アプリケーションが処理を続行できるように、予測可能な状態を再度確立するためのルーチンを実行できます。

自動再接続されるクライアントの復旧

フェイルオーバーは予期しないイベントであり、自動的に再接続されたクライアントが再接続の結果を設計できるように設計されている場合は、予測可能なものでなければなりません。

予測不可の障害を、予測可能で確かなリカバリーに変える主要な要素は、トランザクションの使用です。

前のセクションでは、ローカル・トランザクションを使用して MQGET と MQPUT を調整する IBM MQ MQI client の例 558 ページの『2』を示しました。 クライアントは MQRC_BACKED_OUT エラーに対する応答として MQCMIT または MQBACK 呼び出しを発行した後、バックアウトされたトランザクションを再実行依頼します。 キュー・マネージャーの障害が発生するとトランザクションはバックアウトされ、クライアント・アプリケーションの動作により、トランザクションもメッセージも失われることはありません。

コールバックの場合、コールバック・コンシューマー・パラメーターの状態が MQCS_SUSPENDED_USER_ACTION であれば、コンシューマー・アプリケーションを再開する必要がある場合があることに注意してください。

すべてのプログラム状態がトランザクションの一部として管理されるわけではないため、再接続の結果がさらに分かりにくくなります。 キュー・マネージャーのフェイルオーバーの影響を受けないクライアント・アプリケーションを設計するには、再接続によって IBM MQ MQI client の状態がどのように変化するかを知る必要があります。

特殊なフェイルオーバー・コードを一切含まないアプリケーションを設計し、再接続エラーを他のエラーと同じロジックで処理するようにすることもできます。 あるいは、再接続に特殊なエラー処理を必要とすることが分かった場合は、IBM MQ にイベント・ハンドラーを登録して、フェイルオーバーを処理するルーチンを実行することもできます。 ルーチンで再接続処理そのものを扱うこともできますし、メインプログラム・スレッドが処理を再開するときに、復旧処理を実行する必要があることを指示するフラグを設定することもできます。

IBM MQ MQI client 環境はフェイルオーバー自体を認識し、再接続後に可能な限りコンテキストを復元します。そのために、この環境では、クライアント内に何らかの状態情報を格納して、その IBM MQ 状態を回復するためにクライアント・アプリケーションに代わって追加の MQI 呼び出しを発行します。 例えば、障害発生時点でオープンされていたオブジェクトへのハンドルが復元され、一時動的キューが同じ名前でオープンされます。 しかし、避けられない変化があるため、こうした変化に対応できる設計が必要です。 これらの変化は、以下の 5 種類に分類できます。

1. アプリケーション・プログラムが一貫性のある新規のコンテキスト状態を復元するまで、MQI 呼び出しから新規の (つまり以前に診断されていない) エラーが返される。

   新規のエラーを受け取る例として、コンテキストを保存してから再接続するまでの間にコンテキストを渡そうとしたときの戻りコード MQRC_CONTEXT_NOT_AVAILABLE があります。 セキュリティー・コンテキストは無許可のクライアント・プログラムには渡されないため、再接続後にコンテキストを復元することはできません。 復元してしまうと、悪意のあるアプリケーション・プログラムがセキュリティー・コンテキストを取得してしまう可能性があります。

   通常、アプリケーションは、一般的で予測可能なエラーは綿密に設計された方法で処理し、一般的でないエラーについては汎用エラー・ハンドラーに任せます。 あらゆるエラーに対応するエラー・ハンドラーは、IBM MQ から切断して再接続したり、場合によってはプログラムを完全に停止したりします。 継続性を高めるには、いくつかのエラーを別の方法で処理しなければならない場合もあります。
2. 非持続メッセージが失われることがある。
3. トランザクションはロールバックされます (非同期コンシューマーも中断される可能性があります。前のテキストを参照してください)。
4. 同期点外で使用された MQGET 呼び出しまたは MQPUT 呼び出しが中断され、メッセージが失われることがある。
5. MQI 呼び出しでの待機時間の延長により、タイミングに起因したエラーが発生する。

コンテキストの消失に関する詳細を以下のセクションにリストします。

- NPMCLASS(HIGH) オプションを指定してキューに入れていて、かつ、キュー・マネージャーの障害が、シャットダウン時に非持続メッセージを保管するオプションを妨げなかった場合を除き、非持続メッセージは廃棄されます。
- 非永続サブスクリプションは、接続が切断されると失われます。 再接続時に再確立されます。 永続サブスクリプションを使用することを検討してください。
- get-wait 間隔が再計算され、その限度を超えている場合には、MQRC_NO_MSG_AVAILABLE が返されます。 同様に、同じ総有効期限時間となるようにサブスクリプションの有効期限が再計算されます。
- キュー内のブラウズ・カーソルの位置が失われますが、通常、これは最初のメッセージの前に再設定されます。
  - MQGMO_BROWSE_MSG_UNDER_CURSOR または MQGMO_MSG_UNDER_CURSOR を指定する MQGET 呼び出しが失敗し、理由コード MQRC_NO_MSG_AVAILABLE が返されます。
  - ブラウズのためにロックされたメッセージがアンロックされます。
  - ハンドルの有効範囲を持つブラウズ・マーク付きメッセージのマークが解除され、再度ブラウズされることがあります。
  - 多くの場合、共有ブラウズ・マーク付きメッセージのマークが解除されます。
- セキュリティー・コンテキストが失われます。 MQPMO_PASS_ALL_CONTEXT を指定してメッセージをキューに入れる場合など、メッセージ・コンテキストの保存を使用しようとすると失敗し、MQRC_CONTEXT_NOT_AVAILABLE が返されます。

- メッセージ・トークンが失われます。 メッセージ・トークンを使用した MQGET で、理由コード MQRC_NO_MSG_AVAILABLE が返されます。

  注: *MsgId* および *CorrelId* はメッセージの一部であるため、フェイルオーバー時にメッセージとともに保存されます。そのため、MsgId または CorrelId を使用した MQGET は、予期したとおりに機能します。
- コミットされていないトランザクションの同期点でキューに書き込まれたメッセージは、使用できなくなります。
- 論理順序に従ってメッセージを処理したり、メッセージ・グループ内のメッセージを処理したりすると、再接続後に戻りコード MQRC_RECONNECT_INCOMPATIBLE が返されます。
- MQI 呼び出しによって、クライアントが現在通常受け取る、より一般的な MQRC_CONNECTION_BROKEN ではなく、MQRC_RECONNECT_FAILED が返される場合があります。
- メッセージがキュー・マネージャーに正常に配信されたかどうかを IBM MQ MQI client が認識しない場合、同期点外での MQPUT 呼び出し中の再接続は MQRC_CALL_INTERRUPTED を返します。 MQCMIT の際の再接続も同様に動作します。
- MQRC_CALL_INTERRUPTED が正常な再接続の後に戻されるのは、以下の成功または失敗に関して、IBM MQ MQI client がキュー・マネージャーから応答を受け取らない場合です。
  - 同期点の外部で MQPUT 呼び出しを使用することによる持続メッセージの配信。
  - 同期点の外部で MQPUT1 呼び出しを使用することによる持続メッセージまたはデフォルトで持続となるメッセージの配信。
  - MQCMIT 呼び出しを使用することによるトランザクションのコミット。 再接続が正常に行われた後にのみ、応答が戻されます。
- チャネルは新規インスタンスとして再始動するため (異なるチャネルになる場合もあります)、チャネル出口の状態は保持されません。
- 一時動的キューは、一時動的キューが開かれていた再接続可能クライアントの復旧プロセスの一部として復元されます。 一時動的キュー上のメッセージは復元されませんが、キューが開かれていた、つまりキューの名前を保存していたアプリケーションは、処理を続行できます。

  キューがそれを作成したアプリケーション以外のアプリケーションによって使用されている場合、キューは復元が間に合わないために次に参照される時点でまだ存在しない可能性があります。 例えば、クライアントが応答先キューとして一時動的キューを作成し、そのキューにチャネルが応答メッセージを置く場合は、キューの復旧が間に合わない可能性があります。 この場合、チャネルは通常、応答先メッセージを送達不能キューに置くことになります。

  再接続可能なクライアント・アプリケーションが一時動的キューを (別のアプリケーションが既にそのキューを作成したために) 名前を指定して開いた場合、再接続の発生時に、IBM MQ MQI client が一時動的キューを再作成できなくなります。これは、作成の基になるモデルがないためです。 MQI では、モデルで一時動的キューを開けるのは単一のアプリケーションのみです。 一時動的キューの使用を必要とする他のアプリケーションは、MQPUT1 またはサーバー・バインディングを使用する必要があります。 また、失敗した場合に、再接続を再び試行できるようにする必要があります。

  一時動的キューに入れることができるのは非持続メッセージのみで、それらのメッセージはフェイルオーバー中に失われます。 失われるのは、再接続中に MQPUT1 を使用して一時動的キューに入れられたメッセージです。 MQPUT1 中にフェイルオーバーが発生した場合、MQPUT1 が成功してもメッセージは入れられません。 この問題の回避策の 1 つは、永続動的キューを使用することです。 サーバー・バインディング・アプリケーションはすべて、再接続できないため、一時動的キューを名前で開くことができます。

Multi データ・リカバリーおよび高可用性

複数インスタンス・キュー・マネージャーを使用した高可用性ソリューションには、ストレージの故障後にデータをリカバリーするメカニズムが含まれている必要があります。

複数インスタンス・キュー・マネージャーを使用した場合、キュー・マネージャー・プロセスの可用性は向上しますが、キュー・マネージャーがメッセージや他の情報を保管するために使用するファイル・システムなどの他のコンポーネントの可用性は向上しません。

データの可用性を高める 1 つの方法は、回復力のあるネットワーク・データ・ストレージを使用することです。 ネットワーク・ファイル・システムおよび回復力のあるデータ・ストレージを使用して独自のソリ

ューションを構築することも、統合ソリューションを購入することもできます。 回復力と災害復旧を組み合わせたものとしては、数十または数百キロメートル離れた場所でのディスク複製を可能にする非同期ディスク複製があります。

ストレージ・メディアを最大限に活用できるように、IBM MQ ディレクトリーをメディアに異なる方法でマップするように構成することができます。 複数インスタンス・キュー・マネージャーでは、IBM MQ のディレクトリーおよびファイルの 2 つのタイプの間に重要な区別があります。

キュー・マネージャーのインスタンス間で共有する必要があるディレクトリー。
キュー・マネージャーの異なるインスタンス間で共有される必要がある情報は、`qmgrs` ディレクトリーと `logs` ディレクトリーの 2 つのディレクトリーにあります。 これらのディレクトリーは、共有ネットワーク・ファイル・システム上になければなりません。 メッセージの作成や削除に伴ってデータが絶えず変わるので、持続的な高可用性と優れたパフォーマンスを備えたストレージ・メディアを使用することをお勧めします。

キュー・マネージャーのインスタンス間で共有する必要 がないディレクトリーおよびファイル。
他のディレクトリーの中には、キュー・マネージャーの各インスタンス間で共有する必要がないために、ミラーリングされたファイル・システムを使用する以外の方法で短時間でリストアされるものもあります。

- IBM MQ 実行可能ファイルおよびツール用ディレクトリー。 再インストールするか、バックアップを行いバックアップされたファイル・アーカイブからリストアすることによって置き換えます。
- インストール済み環境全体に関して変更される構成情報。 構成情報は、 AIX, Linux, and Windows システム上の `mqs.ini` ファイルなどの IBM MQ によって管理されるか、 **MQSC** 構成スクリプトなどの独自の構成管理の一部として管理されます。 バックアップを行い、ファイル・アーカイブを使用してリストアします。
- トレース、エラー・ログ、FFDC ファイルなど、インストール済み環境全体の出力。 これらのファイルは、デフォルト・データ・ディレクトリー内の `errors` サブディレクトリーと `trace` サブディレクトリーに保管されます。 AIX and Linux システム上のデフォルトのデータ・ディレクトリーは`/var/mqm` です。 Windows の場合、デフォルトのデータ・ディレクトリーは IBM MQ のインストール・ディレクトリーです。

バックアップ・キュー・マネージャーにより、複数インスタンス・キュー・マネージャーのメディア・バックアップをリニア・ログを使用して定期的に実行することもできます。 バックアップ・キュー・マネージャーによるリカバリーは、ミラーリングされたファイル・システムからのリカバリーほど高速ではありません。また、最新バックアップ以降に行われた変更はリカバリーされません。 バックアップ・キュー・マネージャー・メカニズムは、ローカルのストレージ故障の後でキュー・マネージャーをリカバリーする場合よりも、オフサイトの災害復旧シナリオで使用する場合に適しています。

IBM MQ 可用性ソリューションの結合

アプリケーションでは、可用性を向上させるために IBM MQ の他の機能を使用します。 複数インスタンス・キュー・マネージャーは、他の高可用性機能を補完します。

IBM MQ クラスターによるキューの可用性の向上

1 つのクラスター・キュー (クラスター内の各マネージャー上のすべてのキューのうちの最大で 1 つ) に複数の定義を作成することで、キューの可用性を向上させることができます。

クラスターの 1 メンバーに障害が起こった後に新しいメッセージがクラスター・キューに送信されるとします。 障害が起きたキュー・マネージャーにメッセージが送信される必要 がない限り、そのメッセージは、キューの定義がある、クラスター内の別の実行中のキュー・マネージャーに送信されます。

クラスターにより可用性は大幅に向上しますが、結果的にメッセージの遅延となる、関連する 2 つの障害シナリオがあります。 複数インスタンス・キュー・マネージャーを使用してクラスターを構築すると、メッセージが遅延する可能性が減ります。

孤立メッセージ
クラスター内のキュー・マネージャーに障害が起きた場合、クラスター内の他のキュー・マネージャーに送信可能なメッセージは、障害が起きたキュー・マネージャーにはそれ以上送信されません。 既に送信済みのメッセージは、障害が起きたキュー・マネージャーが再始動するまで孤立します。

アフィニティー

アフィニティーとは、本来は別のものである 2 つの処理の間で共有される情報を表すために使用される用語です。 例えば、サーバーに要求メッセージを送信するアプリケーションと、応答を処理することになっている同じアプリケーションの間には、アフィニティーが存在します。 その他の例としては、各メッセージの処理が先行するメッセージに依存する、メッセージのシーケンスなどがあります。

メッセージをクラスター・キューに送信する場合は、アフィニティーを考慮する必要があります。 つまり、連続するメッセージを同じキュー・マネージャーに送信する必要があるか、またはそれぞれのメッセージをクラスター内のどのメンバーにも送信できるか、などです。

クラスター内の同じキュー・マネージャーにメッセージを送信する必要があり、そのキュー・マネージャーに障害が発生した場合には、障害が起きたクラスター・キュー・マネージャーが再び実行されるまで、そのメッセージは送信元の伝送キュー内で待機します。

クラスターが複数インスタンス・キュー・マネージャーを使用して構成されている場合は、障害が起きたキュー・マネージャーの再始動を待機する遅延はほんの数分程度に制限され、スタンバイにより引き継がれます。 スタンバイが実行中になっていれば、孤立メッセージの処理が再開され、新たにアクティブ化されたキュー・マネージャー・インスタンスへのチャネルが開始され、伝送キュー内で待機していたメッセージが流れ始めます。

障害が起きたキュー・マネージャーによるメッセージ遅延を克服するようにクラスターを構成する可能な 1 つの方法は、クラスター内の各サーバーに対して 2 つの異なるキュー・マネージャーをデプロイし、その異なるキュー・マネージャーの 1 つはアクティブ・インスタンスとして、もう 1 つはスタンバイ・インスタンスとして配置することです。 これはアクティブ/スタンバイ構成であり、クラスターの可用性を向上させます。

クラスターは、管理の手間が減り、スケーラビリティーが向上するという利点があることに加え、複数インスタンス・キュー・マネージャーを補完する可用性の追加要素を継続的に提供します。 クラスターには、キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスの両方に影響する他のタイプの障害からの保護機能があります。

無中断サービス

クラスターは無中断サービスを提供します。 クラスターが受信した新規のメッセージは、アクティブ・キュー・マネージャーに送信されて処理されます。 無中断サービスの提供を複数インスタンス・キュー・マネージャーに依存しないようにしてください。なぜなら、スタンバイ・キュー・マネージャーが障害を検出して開始を完了し、そのチャネルが再接続され、失敗したメッセージのバッチが再実行依頼されるまでに時間を要するからです。

局所的障害

許容可能なパフォーマンスを実現するには、アクティブ・サーバー、スタンバイ・サーバー、およびファイル・システム・サーバーはミリ秒単位の速度で対話する必要があるため、それらのサーバーの物理的距離には実用面での制約があります。

クラスター・キュー・マネージャーで必要になる対話速度はほぼ秒単位であり、これらのキュー・マネージャーは地理的に世界のどこにでも分散させて配置できます。

操作上のエラー

異なる 2 つのメカニズムを使用して可用性を向上させることにより、人為的エラーなどの操作上のエラーが発生して、可用性の実現が危うくなる可能性を減らすことができます。

キュー共有グループによるメッセージ処理の可用性の向上

z/OS キュー共有グループ (z/OS 上でのみ提供) を使用すると、キュー・マネージャーのグループがキューのサービス提供を共有できるようになります。 あるキュー・マネージャーで障害が起きた場合には、引き続き他のキュー・マネージャーがそのキューにあるすべてのメッセージを処理します。 複数インスタンス・キュー・マネージャーは、z/OS ではサポートされておらず、幅広いメッセージング・アーキテクチャーの中の限られた一部分としてのみキュー共有グループを補完します。

IBM MQ クライアントによるアプリケーションの可用性の向上

IBM MQ MQI client・プログラムは、キュー・マネージャーの可用性、接続の重み、およびアフィニティーに基づいて、1 つのキュー・マネージャー・グループ内の複数の異なるキュー・マネージャーに接続できま

す。 キュー・マネージャーが実行しているマシンとは異なるマシン上でアプリケーションを実行することで、ソリューションの総合的な可用性を向上させることができます。ただしこれは、接続先のキュー・マネージャー・インスタンスに障害が起きた場合に、アプリケーションを再接続する方法があることが前提です。

キュー・マネージャー・グループを使用すると、停止したキュー・マネージャーとクライアントの結合が解除され、IP スプレイヤーのように、キュー・マネージャーのグループ内でクライアント接続のロード・バランシングが行われるため、クライアントの可用性が向上します。 クライアント・アプリケーションには、障害が起きたキュー・マネージャーとのアフィニティー (特定のキューへの依存など) があってはなりません。そうでないと、処理を再開できません。

クライアントの自動再接続および複数インスタンス・キュー・マネージャーを使用すると、いくつかのアフィニティーの問題が解決されるため、クライアントの可用性は向上します。 IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

MQCNO のオプション MQCNO_RECONNECT_Q_MGR を設定して、クライアントを同じキュー・マネージャーに強制的に再接続させることができます。

1. 以前に接続していた単一インスタンスのキュー・マネージャーが実行していない場合には、そのキュー・マネージャーが再度実行中になるまで接続が再試行されます。
2. キュー・マネージャーが複数インスタンス・キュー・マネージャーとして構成されている場合、クライアントはアクティブであるいずれかのインスタンスに再接続します。

同じキュー・マネージャーに自動的に再接続することで、クライアントの代わりにキュー・マネージャーが保持していた多くの状態情報 (開いたキューや、サブスクライブしていたトピック) は、復元されます。要求への応答を受け取るためにクライアントが動的応答先キューを開いていた場合、応答先キューへの接続も復元されます。

Linux MQ Adv. RDQM 高可用性

RDQM (複製データ・キュー・マネージャー) は、 Red Hat Enterprise Linux for x86-64 プラットフォームで使用可能な高可用性ソリューションです。

RDQM 構成は、高可用性 (HA) グループとして構成されている 3 つのサーバーから成り、その個々にキュー・マネージャーのインスタンスがあります。 インスタンスの 1 つは実行中のキュー・マネージャーで、データを他の 2 つのインスタンスに同期的に複製します。 このキュー・マネージャーを実行しているサーバーに障害が発生すると、別のキュー・マネージャーのインスタンスが開始され、このインスタンスには操作に使用する現行データがあります。 キュー・マネージャーのインスタンスは 3 つとも浮動 IP アドレスをオプションで共有できるので、クライアントの構成時に使用する必要がある IP アドレスは 1 つだけです。 ネットワークの問題のために HA グループがパーティション化されていても、同時に実行できるキュー・マネージャーのインスタンスは 1 つだけです。 キュー・マネージャーを実行しているサーバーは「1 次」と呼ばれ、その他の 2 つのサーバーはどちらも「2 次」と呼ばれます。

3 つのノードを使用すると、スプリット・ブレーンという状態になる可能性が大幅に減少します。 2 ノードの高可用性システムでは、2 つのノード間の接続が切断されるとスプリット・ブレーンが発生する可能性があります。 接続のない状態だと、両方のノードでキュー・マネージャーが同時に実行され、別々のデータが累積される可能性があります。 そうなると、接続が復元された時に 2 つの別々のバージョンのデータ (「スプリット・ブレーン」) が存在することになるので、保持するデータ・セットと破棄するデータ・セットを決定するために手操作による介入が必要になります。

RDQM では、スプリット・ブレーン状態を回避するために、クォーラムを設定した 3 ノード・システムを使用します。 1 つのノードが他の 1 つ以上のノードと通信できる場合は、その通信可能なノード同士がクォーラムを形成します。 キュー・マネージャーは、クォーラムを形成するノードでしか実行できません。他の 1 つ以上のノードと接続していないノードではキュー・マネージャーを実行できないので、接続の切れた 2 つのノードでキュー・マネージャーが同時に実行されることは絶対にありません。

- 1 つのノードで障害が発生しても、他の 2 つのノードのいずれかでキュー・マネージャーを実行できます。 2 つのノードで障害が発生すると、残りの 1 つのノードではキュー・マネージャーを実行できません。そのノードはクォーラムを形成していないからです (残りのノードは、他の 2 つのノードで障害が失敗したのか、それとも他の 2 つのノードは実行中だけれども接続が失われたのかを判別できません)。

- 1 つのノードが接続を失うと、そのノードではキュー・マネージャーを実行できません。そのノードはクォーラムを形成していないからです。 残りの 2 つのノードのいずれかではキュー・マネージャーを実行できます。その 2 つがクォーラムを形成しているからです。 すべてのノードが接続を失うと、どのノードでもキュー・マネージャーを実行できません。どのノードもクォーラムを形成していないからです。

注 : IBM MQ Console は、複製データ・キュー・マネージャーをサポートしていません。 IBM MQ Explorer は複製データ・キュー・マネージャーに使用できますが、 RDQM 機能に固有の情報は表示されません。

3 つのノードのグループ構成は、Pacemaker が実施します。 3 つのノード間の複製は、DRBD が実施します。 (Pacemaker については https://clusterlabs.org/pacemaker/ を、DRBD については https://docs.linbit.com/docs/users-guide-9.0/ を参照してください。)

複製データ・キュー・マネージャーは、672 ページの『キュー・マネージャー・データのバックアップ』で説明しているプロセスを使用してバックアップできます。 このキュー・マネージャーを停止してバックアップしても、 RDQM 構成で実行されているノード・モニターには影響を与えません。

以下の図に、HA グループ内の 3 つの各ノード上で 1 つずつ RDQM が実行されている標準的なデプロイメントを示します。

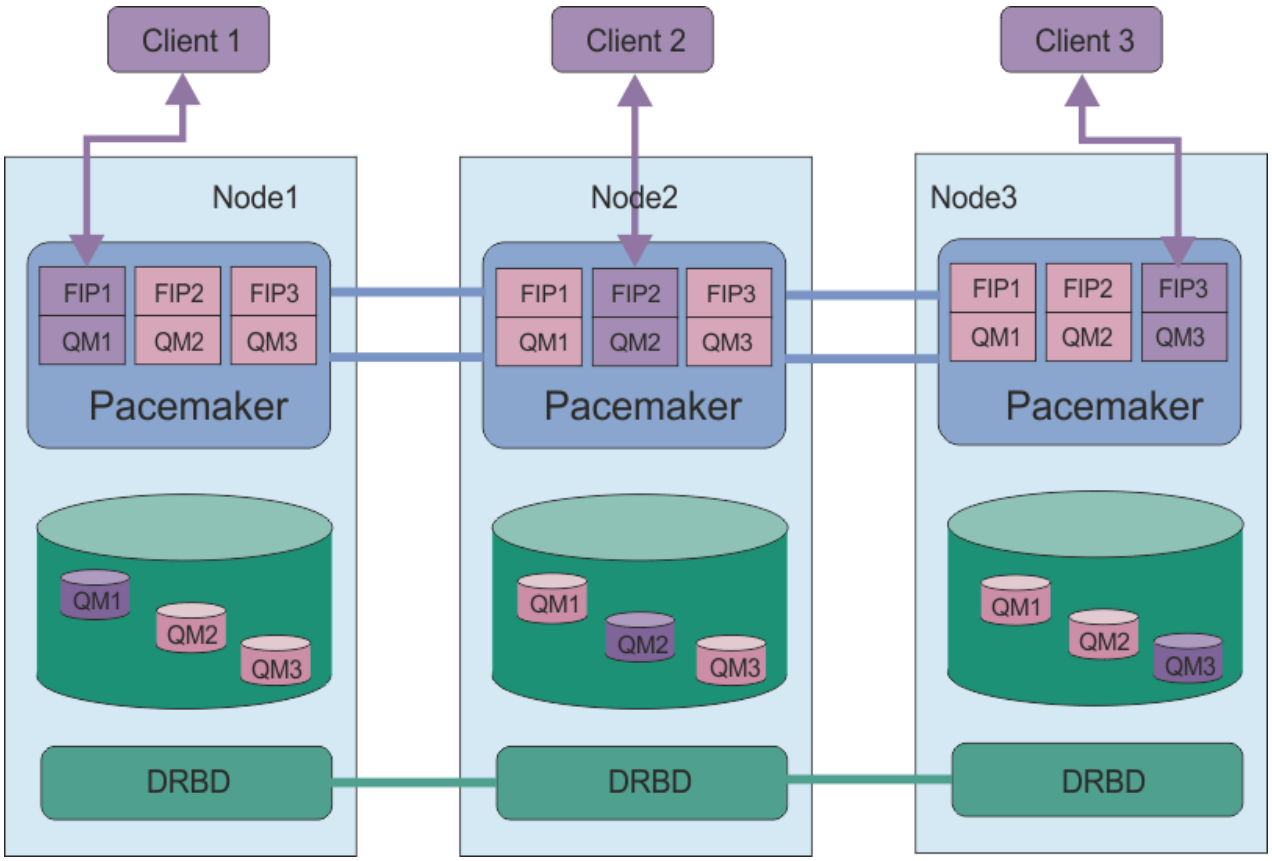


図 *78. 3 つの RDQM がある HA グループの例*

以下の図では、Node3 で障害が発生し、Pacemaker リンクが失われ、キュー・マネージャー QM3 が Node2 で実行されています。

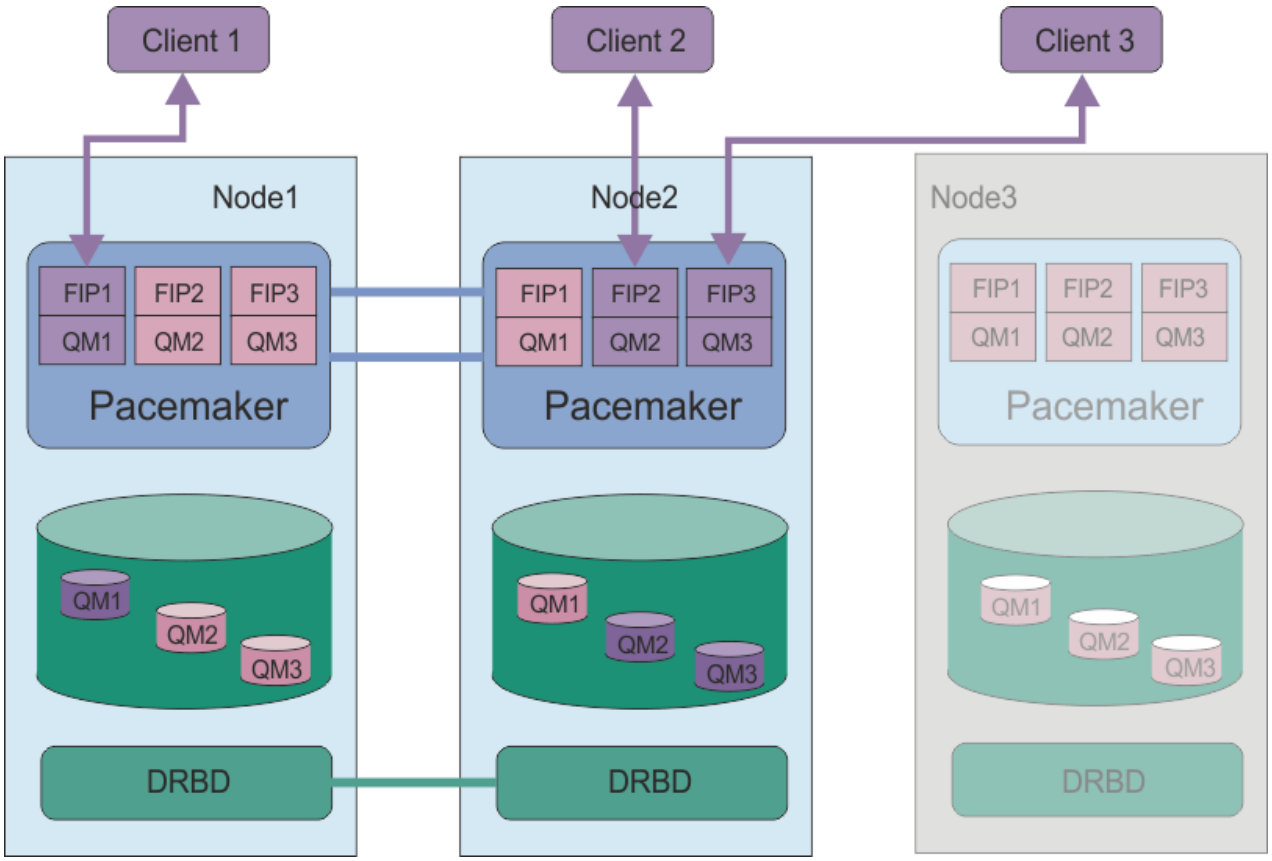


図 79. *Node3* で障害が発生した後の例

注 : キュー・マネージャーが別のノードにフェイルオーバーする際、それらのキュー・マネージャーはフェイルオーバー時の状態を保持します。 実行中のキュー・マネージャーが開始され、停止されたキュー・マネージャーは停止したままです。

関連タスク

RDQM (複製データ・キュー・マネージャー) のインストール

RDQM のための保守レベルの更新の適用

複製データ・キュー・マネージャーのマイグレーション

RDQM 構成のトラブルシューティング

Linux *RDQM HA ソリューションの要件*

RDQM 高可用性 (HA) グループを構成する前に、いくつかの要件を満たしている必要があります。

システム要件

RDQM HA グループを構成する前に、HA グループの一部となる 3 つのサーバーのそれぞれで、いくつかの構成を完了する必要があります。

- 各ノードには、`drbdpool` という名前のボリューム・グループが必要です。 各複製データ・キュー・マネージャーのストレージは、このボリューム・グループからのキュー・マネージャーごとに別個の論理ボリュームとして割り振られます。 最高のパフォーマンスを得るためには、このボリューム・グループを内部ディスク・ドライブ (SSD) に対応する 1 つ以上の物理ボリュームで構成する必要があります。 RDQM HA ソリューションのインストール前またはインストール後に `drbdpool` を作成できますが、実際に RDQM を作成する前に `drbdpool` を作成する必要があります。 **vgs** コマンドを使用してボリューム・グループ構成を確認します。 出力は次のようになります。

```
VG       #PV #LV #SN Attr   VSize   VFree
drbdpool   1   9   0 wz--n- <16.00g <7.00g
rhel       1   2   0 wz--n- <15.00g     0
```

特に、属性の 6 番目の列に c 文字 (つまり、wz--nc) がないことを確認してください。c は、クラスタリングが有効になっていることを示します。有効になっている場合は、ボリューム・グループを削除して、クラスタリングなしで再作成する必要があります。

- drbdpool ボリューム・グループを作成した後は、それ以外の操作は何も行いません。 IBM MQ は、drbdpool で作成された論理ボリューム、およびそれらがマウントされる方法と場所を管理します。
- 各ノードには、RDQM サポートの構成に使用されるインターフェースが最大で 3 つ必要です。
  - HA グループをモニターするための Pacemaker の基本インターフェース。
  - HA グループをモニターするための Pacemaker の代替インターフェース。
  - 同期データ複製用のインターフェース。複製インターフェースと呼ばれます。 HA グループで実行されるすべての複製データ・キュー・マネージャーの予期されるワークロードを前提として、複製要件をサポートするために十分な帯域幅が必要です。

  3 つのすべてのインターフェースに同じ IP アドレスを使用するか、各インターフェースに別個の IP アドレスを使用するか、または基本インターフェースと代替インターフェースに同じ IP アドレスを使用して、複製インターフェースに別個の IP アドレスを使用するように HA グループを構成できます。

  最大限の耐障害性を実現するためには、これらのインターフェースは独立ネットワーク・インターフェース・カード (NIC) である必要があります。
- DRBD では、RFC 1123 で修正された RFC 952 によって定義されているとおり、HA グループ内の各ノードが有効なインターネット・ホスト名 (uname -n によって返される値) を持っている必要があります。
- HA グループのノード間にファイアウォールがある場合は、ファイアウォールでポートの範囲のノード間のトラフィックが許可される必要があります。 RHEL で標準のファイアウォールを実行している場合は、必要なポートを開くサンプル・スクリプト(/opt/mqm/samp/rdqm/firewalld/configure.sh)が提供されています。 このスクリプトは root として実行する必要があります。 他のファイアウォールを使用している場合は、サービス定義 /usr/lib/firewalld/services/rdqm* を調べて、どのポートを開く必要があるかを確認してください。 スクリプトは、DRBD、Pacemaker、および IBM MQ に関する以下の永続的なファイアウォール ID サービス規則を追加します。
  - MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-drbd.xml は TCP ポート 7000-7100 を使用できるようにします。
  - MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-pacemaker.xml は UDP ポート 5404-5407 を使用できるようにします。
  - MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-mq.xml は TCP ポート 1414 を使用できるようにします (別のポートが必要な場合はスクリプトを編集する必要があります)。
- システムが SELinux を Enforcing モードで使用している場合は、以下のコマンドを実行する必要がある可能性があります。

```
semanage permissive -a drbd_t
```

V 9.3.0.1 V 9.3.2 drbd-selinux パッケージをインストールした場合は、**semanage** を実行する必要はありません。 このパッケージをすべてのノードにインストールするか、すべてのノードで **semanage** を実行する必要があります。

ネットワーク要件

RDQM HA グループの 3 つのノードを同じデータ・センターに配置することをお勧めします。

そうしたノードを別々のデータ・センターに配置する場合は、以下の制約に注意する必要があります。

- データ・センター間の待ち時間が長くなると、パフォーマンスが急速に低下します。 IBM では最大 5 ミリ秒の待ち時間をサポートしていますが、アプリケーション・パフォーマンスの観点からすれば、待ち時間が 1 から 2 ミリ秒より長くなるのは許容できないかもしれません。
- 複製リンクで送信するデータは、IBM MQ AMS での暗号化は別として、それ以外の暗号化の対象になりません。

浮動 IP アドレスを任意で構成できます。これにより、HA グループ内のどのノードでクライアントが実行されているかに関わりなく、クライアントが複製データ・キュー・マネージャー (RDQM) に関して同じ IP アドレスを使用することが可能になります。 浮動アドレスは、RDQM の 1 次ノードの指定した物理インターフェースにバインドします。 RDQM がフェイルオーバーし、別のノードが 1 次ノードになると、浮動 IP は、新しい 1 次上の同じ名前のインターフェースにバインドされます。 3 つのノードの物理インターフェースは、すべて同じ名前でなければなりません。また、浮動 IP アドレスと同じサブネットに属していることが必要です。

クラスターを構成するためのユーザー要件

RDQM HA グループをユーザー root として構成できます。 root として構成しない場合は、代わりに mqm グループのユーザーとして構成します。 mqm グループのユーザーが RDQM クラスターを構成するには、以下の要件を満たす必要があります。

- mqm ユーザーは、RDQM HA グループを構成する 3 つのサーバーのそれぞれで sudo を使用してコマンドを実行できる必要があります。
- mqm ユーザーがパスワードなしで SSH を使用して、RDQM HA グループを構成する 3 つのサーバーのそれぞれでコマンドを実行できる場合、ユーザーはそれらのサーバーの 1 つでのみコマンドを実行する必要があります。
- mqm ユーザーは、3 つのサーバーすべてで同じ UID を持っている必要があります。
- mqm グループは、3 つのサーバーすべてで同じ GID を持っている必要があります。

mqm ユーザーが root 権限で以下のコマンドを実行できるように、sudo を構成する必要があります。

```
/opt/mqm/bin/crtmqm
/opt/mqm/bin/dltmqm
/opt/mqm/bin/rdqmadm
/opt/mqm/bin/rdqmstatus
```

キュー・マネージャーで作業するためのユーザー要件

複製データ・キュー・マネージャー (RDQM) を作成、削除、または構成するには、mqm グループと haclient グループの両方に属するユーザー ID を使用する必要があります (haclient グループは、Pacemaker のインストール時に作成されます)。

Linux パスワードなしの *SSH* のセットアップ

HA グループの 1 つのノードで構成コマンドを発行するだけで済むように、パスワードなしの SSH をセットアップできます。 (パスワードなしの SSH のセットアップはオプションです。代わりに、各ノードに手動でコマンドをコピーすることもできます。)

このタスクについて

パスワードなしの SSH をセットアップするには、各ノードで mqm ID を構成して、そのユーザーに対して各ノードで鍵を生成する必要があります。 次に、鍵を他のノードに配布して、接続をテストし、既知のホストのリストに各ノードを追加します。 最後に、 mqm ID をロックダウンします。

注 : この説明は、個別の 1 次、代替、および複製のインターフェースで HA グループを定義していることが前提となっています。したがって、1 次および代替のインターフェースを介してパスワードなしの SSH アクセスを定義します。 単一 IP アドレスでシステムを構成する予定の場合は、その単一インターフェースを介してパスワードなしの SSH アクセスを定義します。

RDQM は、 ssh コマンドが対話なしで (つまり、パスワードの入力を求めるプロンプトを出さずに) 機能することを必要とします。

手順

1. 3 つのノードそれぞれで以下の手順を実行し、mqm ユーザーをセットアップし、SSH 鍵を生成します。
   a) mqm ホーム・ディレクトリーを /home/mqm に変更します。

```
usermod -d /home/mqm mqm
```

b) /home/mqm ディレクトリーを作成します:

```
mkhomedir_helper mqm
```

c) mqm パスワードを追加します。

```
passwd mqm
```

d) 対話式シェルを mqm として実行します。

```
su mqm
```

e) mqm 認証鍵を生成します。

```
ssh-keygen -t rsa -f /home/mqm/.ssh/id_rsa -N ''
```

2. 3 つのノードそれぞれで以下の手順を実行し、ノードの鍵を他の 2 つのノードに追加し、各ノードの 1 次および代替 (使用している場合) のアドレスの接続をテストします。

a) リモート・ノードに鍵を追加します。

```
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_primary_address
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_alternate_address
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_primary_address
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_alternate_address
```

b) パスワードなしの SSH を確認して、リモート・ノードの known_hosts を更新します。

```
ssh remote_node1_primary_address uname -n
ssh remote_node1_alternate_address uname -n
ssh remote_node2_primary_address uname -n
ssh remote_node2_alternate_address uname -n
```

接続ごとに、続行するかどうかを確認するプロンプトが出されます。 それぞれが known_hosts を更新できるように確認します。 パスワードなしの SSH を使用して HA グループを構成する前に、これを完了する必要があります。

c) mqm として対話式シェルを終了します。

```
exit
```

3. 各ノードで root として以下の手順を実行し、mqm パスワードを削除して ID をロックします。

a) mqm パスワードを削除します。

```
passwd -d mqm
```

b) mqm をロックします。

```
passwd -l mqm
```

4. 各ノードで root として次の手順を実行して、mqm ユーザーの sudo アクセス権限をセットアップします。

a) **visudo** コマンドを使用して、sudoers ファイルを編集します。

```
visudo
```

b)「## Allows people in group wheel to run all commands」という行を探し、 その行の下に次のテキストを追加します。

```
#%mqm    ALL=(ALL)       ALL
```

c)「## Same thing without a password」という行を探し、その行の下に次のテキストを追加します。

```
%mqm            ALL=(ALL)       NOPASSWD: ALL
```

Linux *Pacemaker クラスター (HA グループ) の定義*

HA グループは、Pacemaker クラスターです。 Pacemaker クラスターを定義するには、`/var/mqm/rdqm.ini` ファイルを編集し、**rdqmadm** コマンドを実行します。

このタスクについて

Pacemaker については、https://clusterlabs.org/pacemaker/を参照してください。 `mqm` ユーザーが `sudo` を使用できる場合は、`mqm` グループのユーザーとして Pacemaker クラスターを作成できます。 ユーザーが各サーバーに対してパスワードなしの SSH を使用することもできる場合は、`rdqm.ini` ファイルを編集し、いずれかのサーバーで **rdqmadm** を実行するだけで Pacemaker クラスターを作成できます。 それ以外の場合は、ファイルを作成し、ノードとなる各サーバー上で `root` としてコマンドを実行する必要があります。

`rdqm.ini` ファイルは、 Pacemaker クラスター内のノード用に RDQM によって使用される IP アドレスを指定します。 RHEL 8 および RHEL 9 のインストール済み環境では、各ノードの名前を指定する必要があります。これは、 **uname -n** コマンドによって返されるホスト名でなければなりません。 RHEL 7 インストール済み環境の場合、ノード名の指定はオプションです。

RDQM HA グループは、1 つ、2 つ、または 3 つの IP アドレスを使用するように構成できます。

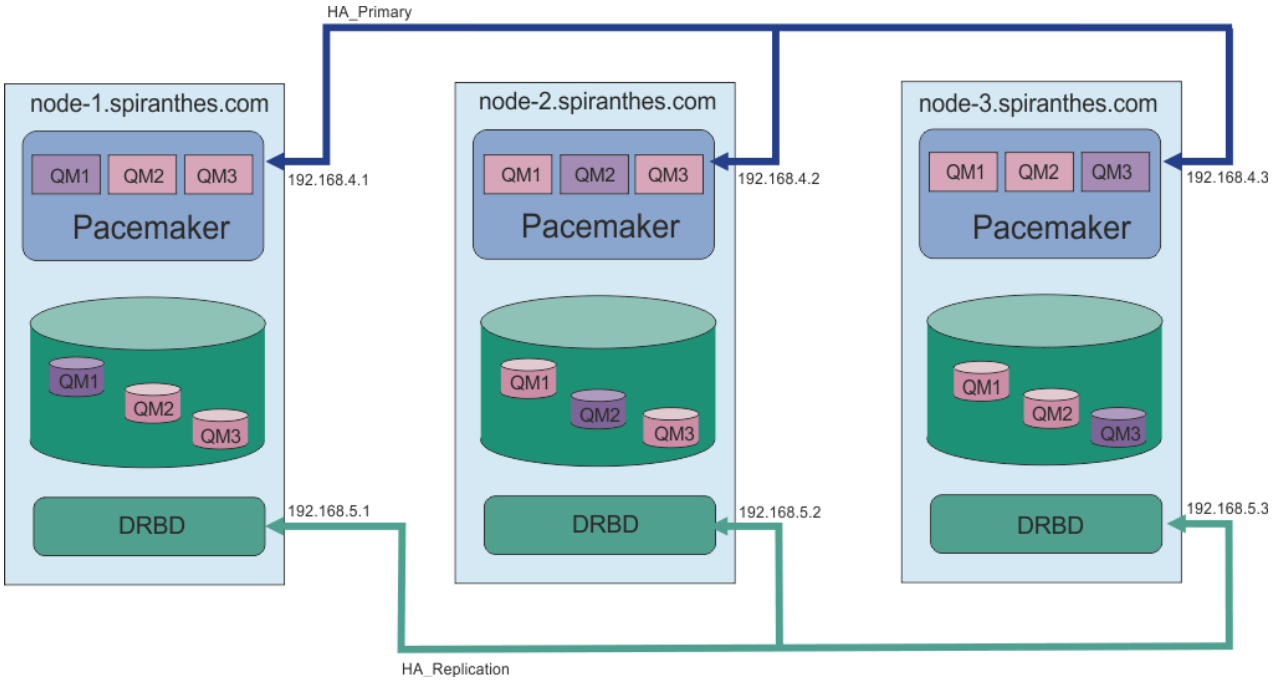
- 1 つの IP アドレス: ハートビートと複製は同じリンクを共有します
- 2 つの IP アドレス: ハートビートと複製は別々のリンクを使用します
- 3 つの IP アドレス: 複製用に 1 つのリンク、およびハートビート用に 2 つの別個のリンク

これらのオプションは、RDQM のさまざまなデプロイメント・パターンをサポートするために提供されています。 さまざまなオプションを使用して、使用されている環境に基づいて RDQM ソリューションの回復力を最大限に高めることができます。 2 つまたは 3 つの IP アドレスを使用する構成は、主に、ノード間の接続の冗長性を構成するために、どの物理ネットワークがハートビートと複製トラフィックをリンクするかを細かく制御する必要があるデプロイメントを対象としています。 あるいは、例えばリンク集約を使用して、ネットワーク層で高可用性と回復力のある接続を実装することもできます。 リンク集約では、複数の物理ネットワーク・リンクを使用して、個々の物理リンクに障害が発生した場合に機能し続けることができる単一の論理リンクを提供します。 ネットワーク接続が仮想化されている環境や、ネットワーク層で回復力のある接続が実装されている環境に RDQM がデプロイされている場合は、通常、ハートビートと複製の両方に単一の IP アドレスを使用することをお勧めします。

以下の例は、2 つの IP アドレスの使用法を示しています。 `rdqm.ini` ファイルには、ノードごとに `HA_Primary` フィールドと `HA_Replication` フィールドがありますが、`HA_Alternate` フィールドはありません。

```
Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Primary=192.168.4.1
  HA_Replication=192.168.5.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Primary=192.168.4.2
  HA_Replication=192.168.5.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Primary=192.168.4.3
  HA_Replication=192.168.5.3
```

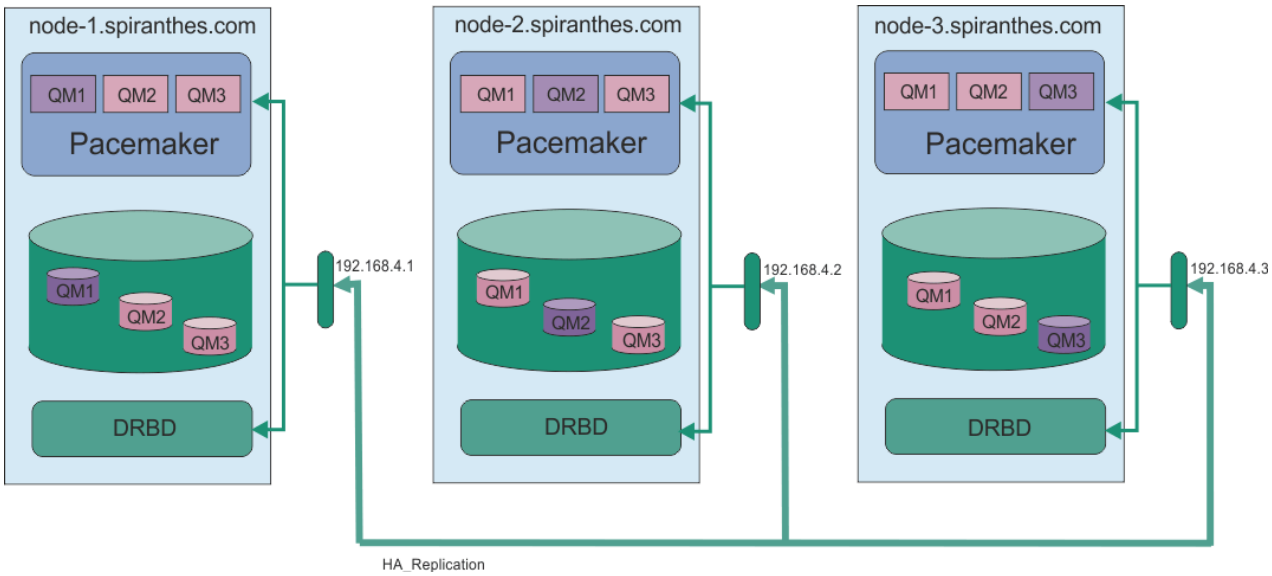
次の図は、この構成を示しています:



以下のサンプル・ファイルは、モニター用に `HA_Replication` インターフェースを使用する Pacemaker クラスターの例の構成を示しています (例えば、PoC (概念検証) デプロイメントに使用できます)。 この場合は、 `HA_Replication` インターフェースのみを指定します。

```
Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Replication=192.168.4.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Replication=192.168.4.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Replication=192.168.4.3
```

次の図は、この構成を示しています:

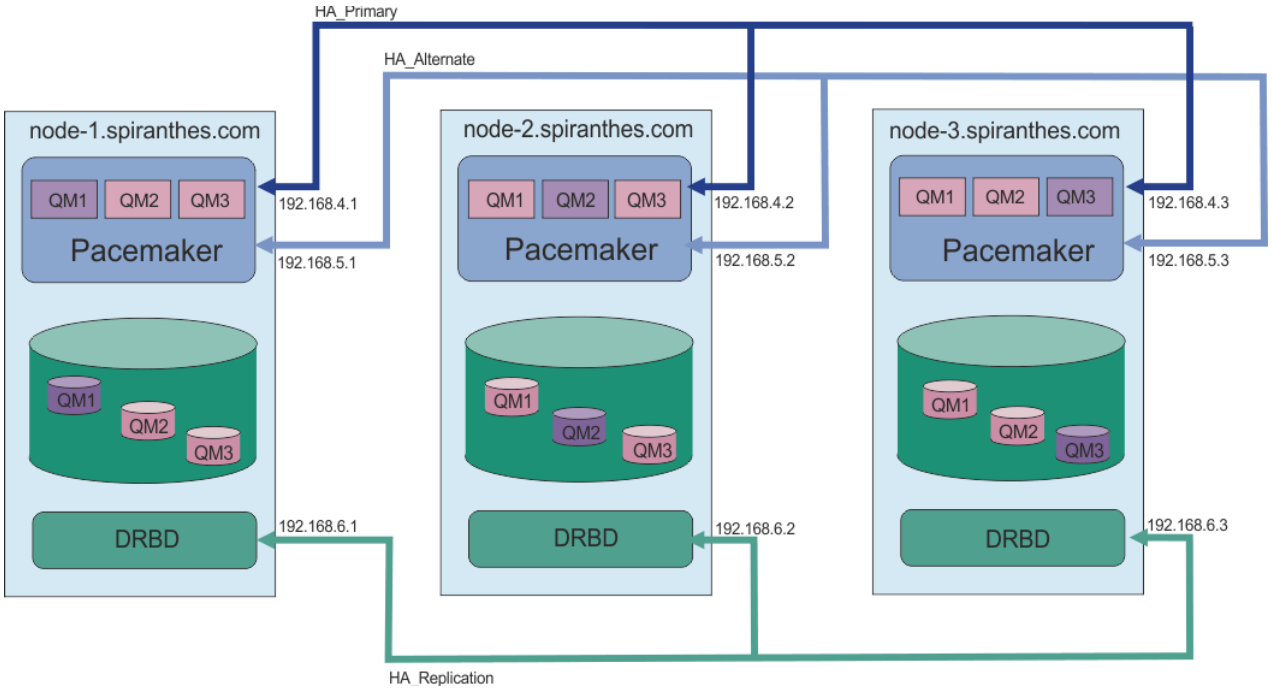


以下のサンプル・ファイルは、各インターフェースに別個の IP アドレスを使用するサンプル Pacemaker クラスターの構成を示しています。

```
Node:
  Name=rdqm-node-1.spiranthes.com
```

```
  HA_Primary=192.168.4.1
  HA_Alternate=192.168.5.1
  HA_Replication=192.168.6.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Primary=192.168.4.2
  HA_Alternate=192.168.5.2
  HA_Replication=192.168.6.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Primary=192.168.4.3
  HA_Alternate=192.168.5.3
  HA_Replication=192.168.6.3
```

次の図は、この構成を示しています:



ノードを指定する順序は、構成内のすべての `rdqm.ini` ファイルで同じでなければなりません。 3 つのノードには、どれが Node1 でどれが Node2 であるかなどについて共通のビューが必要です。

手順

- Pacemaker クラスターを `root` ユーザーとして定義するには、次の手順を実行します。
  a) 3 つのサーバーのいずれかにある `/var/mqm/rdqm.ini` ファイルを編集して、ファイルがクラスターを定義するようにします。
  b) このファイルを、Pacemaker クラスター内のノードとなる他の 2 つのサーバーにコピーします。
  c) 3 つのサーバーそれぞれで、以下のコマンドを `root` として実行します。

  ```
  rdqmadm -c
  ```

- 各ノードの `mqm` グループのユーザーとして Pacemaker クラスターを定義するには、以下のようにします。
  a) ユーザー `mqm` が **sudo** を使用してコマンドを実行できることを確認してください。
  b) 3 つのサーバーのいずれかで `/var/mqm/rdqm.ini` ファイルを編集して、ファイルが Pacemaker クラスターを定義するようにします。
  c) `/var/mqm/rdqm.ini` を、Pacemaker クラスター内のノードになる他の 2 つのサーバーにコピーします。
  d) 各サーバーで次のコマンドを実行します。

```
rdqmadm -c
```

- 1 つのノードの mqm グループのユーザーとして Pacemaker クラスターを定義するには、以下のようにします。
  a) ユーザー mqm が **sudo** を使用してコマンドを実行でき、パスワードなしで SSH を使用して各サーバーに接続できることを確認します。
  b) 3 つのサーバーのいずれかで/var/mqm/rdqm.ini ファイルを編集して、ファイルが Pacemaker クラスターを定義するようにします。
  c) 以下のコマンドを実行します。

```
rdqmadm -c
```

関連資料
rdqmadm (複製データ・キュー・マネージャー・クラスターの管理)

Linux *Pacemaker クラスター (HA グループ) の削除*
HA グループは、Pacemaker クラスターです。 -u オプションを指定して **rdqmadm** コマンドを実行すると、Pacemaker クラスター構成を削除できます。

このタスクについて

いずれかのノードに複製データ・キュー・マネージャーが存在する場合は、Pacemaker クラスター構成を削除できません。

手順

- Pacemaker クラスター構成を削除するには、任意のノードから以下のコマンドを入力します。

```
rdqmadm -u
```

関連資料
rdqmadm (複製データ・キュー・マネージャー・クラスターの管理)

Linux ***HA RDQM の作成***
crtmqm コマンドを使用して、高可用性複製データ・キュー・マネージャー (RDQM) を作成します。

このタスクについて

mqm ユーザーが sudo を使用できる場合は、mqm グループのユーザーとして高可用性複製データ・キュー・マネージャー (RDQM) を作成できます。 ユーザーが各サーバーに対してパスワードなしの SSH を使用することもできる場合は、1 つのノードで create RDQM コマンドを実行するだけで 3 つすべてのノードで RDQM を作成できます。 その他の場合、RDQM を作成するには root である必要があり、3 つのすべてのノードでコマンドを実行する必要があります。

注 : HA グループには、129 のキュー・マネージャーという絶対的な制限があります。 これを超えて作成しようとすると、失敗します。 実際には、HA グループに 50 を超えるキュー・マネージャーを追加すると、タイムアウトの問題が発生する可能性があります。

以下に、キュー・マネージャーのファイル・システムのサイズ変更に関するガイダンスを示します。

1. RDQM キュー・マネージャーの作成時に、キュー・マネージャーのデータとログを保管するためにファイル・システムが割り振られます。 キュー・マネージャーが進行中のアクティビティーをログに記録し、アプリケーション・メッセージをキューに保管できるように、このファイル・システムを適切にサイズ変更することが重要です。 ファイル・システムをサイジングする際には、ピーク時のメッセージング要件、将来のワークロードの増加、およびメッセージがキューに蓄積される原因となる可能性のあるアプリケーション障害を考慮してください。 キュー・マネージャーの回復ログのサイズを計算するためのガイダンスについては、 651 ページの『ログ・ファイル・システムの大きさの決定方法』を参照し

てください。 アプリケーション・メッセージのストレージ要件を計算する際には、メッセージのサイズと数に加えて、MQMD ヘッダーおよびメッセージ・プロパティーを考慮する必要があります。

2. RDQM キュー・マネージャーのファイル・システムを動的にサイズ変更することはできません。 これが必要な場合は、より大きなファイル・システムを使用して RDQM キュー・マネージャーをバックアップしてからリストアする必要があります。 579 ページの『HA RDQM キュー・マネージャーのファイル・システムのサイズ変更』を参照してください。
3. MAXDEPTH や MAXFSIZE などのローカル・キュー属性を使用して、ディスク上の個々のキューのサイズを制限することができます。 IBM MQ キュー・ファイルの変更を参照してください。
4. ファイル・システムの使用量が重要になる前に、進行中のディスク使用量をモニターし、ディスク使用量が増加した場合は適切に対応する必要があります。 ファイル・システムの使用状況をモニターするには、プラットフォーム/オペレーティング・システムの機能を使用するか、 システム・トピックでパブリッシュされるメトリックで説明されている IBM MQ システム・トピックにパブリッシュされるメトリックをサブスクライブします。

手順

- mqm グループのユーザーとして RDQM を作成するには、次のようにします。

  a) mqm ユーザーが **sudo** を使用してコマンドを実行できること、およびパスワードなしの SSH を使用して各サーバーに接続できることを確認します。

  b) 以下のコマンドを入力します。

  ```
  crtmqm -sx [-fs FilesystemSize] qmname
  ```

  ここで、*qmname* は、複製データ・キュー・マネージャーの名前です。 オプションでキュー・マネージャーのファイル・システム・サイズ (drbdpool ボリューム・グループで作成される論理ボリュームのサイズ) を指定できます。

  このコマンドは、mqm ユーザーとして SSH を使用して、クラスター内の他のノードへの接続を試行します。 接続が成功すると、ノードでキュー・マネージャーの 2 次インスタンスが作成されます。 成功しなかった場合は、2 次インスタンスを作成して、**crtmqm -sx** コマンドを実行する必要があります (ユーザー root 用の説明に従います)。

- root ユーザーとして RDQM を作成するには、次のようにします。

  a) RDQM の 2 次インスタンスをホストする各ノードで以下のコマンドを入力します。

  ```
  crtmqm -sxs [-fs FilesystemSize] qmname
  ```

  ここで、*qmname* は、複製データ・キュー・マネージャーの名前です。 オプションでキュー・マネージャーのファイル・システム・サイズ (drbdpool ボリューム・グループで作成される論理ボリュームのサイズ) を指定できます。 HA グループ内の 3 つのノードすべての RDQM に同じファイル・システム・サイズを指定する必要があります。 サイズは、GB 単位で指定される数値です。 値を MB 単位で指定するには、値の後に文字 M を入力します。

  このコマンドは、RDQM の 2 次インスタンスを作成します。

  b) 残りのノードで、以下のコマンドを入力します。

  ```
  crtmqm -sx [-fs FilesystemSize] qmname
  ```

  ここで、*qmname* は、複製データ・キュー・マネージャーの名前です。 オプションでキュー・マネージャーのファイル・システム・サイズを指定できます。 サイズは、GB 単位で指定される数値です。 値を MB 単位で指定するには、値の後に文字 M を入力します。

  このコマンドは、キュー・マネージャーの 2 次インスタンスが他の 2 つのノードに存在するかどうかを判別します。 2 次インスタンスが存在する場合、コマンドは、1 次キュー・マネージャーを作成し、開始します。 2 次が存在しない場合は、各ノードで **crtmqm -sxs** コマンドを実行するように指示されます。

  DataPath (**-md**) 引数および LogPath (**-ld**) 引数を除き、標準 Linux キュー・マネージャーの作成に有効なすべての引数は、1 次複製データ・キュー・マネージャーにも有効です。

注 : RDQM を作成すると、その複製リンクには 7000 以上で空いている次のポート番号が割り振られます。 選択されたポートが別のアプリケーションによって使用されていることが検出された場合、**crtmqm** コマンドは AMQ6543 エラーで失敗し、そのポートが除外リストに追加されます。 キュー・マネージャーの 2 次インスタンスを削除してから、**crtmqm** コマンドを再実行する必要があります。

関連資料
crtmqm

Linux *HA RDQM の削除*
dltmqm コマンドを使用して、高可用性複製データ・キュー・マネージャー (RDQM) を削除します。

このタスクについて

RDQM の 1 次ノードから RDQM を削除するには、コマンドを実行する必要があります。 最初に RDQM を終了する必要があります。 mqm ユーザーが必要な sudo 特権を持っている場合は、mqm ユーザーとしてコマンドを実行できます。 その他の場合は、root としてコマンドを実行する必要があります。 1 次キュー・マネージャーに関連したリソースが削除された後、コマンドは他のノードに接続するための ssh を使用して、2 次キュー・マネージャーの削除を試みます。 この削除が失敗した場合は、他のノードで `dltmqm` を手動で実行してプロセスを完了する必要があります。 1 次キュー・マネージャーがまだ削除されていない場合は、2 次ノードでコマンドが失敗します。

手順

- RDQM を削除するには、以下のコマンドを入力します。

```
dltmqm RDQM_name
```

関連資料
dltmqm

Linux MQ Adv. キュー・マネージャーをマイグレーションして *HA RDQM* キュー・マネージャーにする
既存のキュー・マネージャーをマイグレーションして高可用性 (HA) 複製データ・キュー・マネージャー (RDQM) にするには、その永続データをバックアップしてから、新規に作成された同じ名前の RDQM キュー・マネージャーにデータをリストアします。

このタスクについて

HA 複製データ・キュー・マネージャーには、専用の論理ボリューム (ファイル・システム) と、ディスク複製および HA 制御の構成が必要です。 これらのコンポーネントは、新規キュー・マネージャーが作成されたときにのみ構成されます。 既存のキュー・マネージャーをマイグレーションして RDQM を使用するには、その永続データをバックアップしてから、新規に作成された同じ名前の RDQM キュー・マネージャーにデータをリストアします。 この手順では、バックアップの作成時にキュー・マネージャーの構成、状態、および持続メッセージを保存します。

注 : キュー・マネージャーは、RDQM がインストールされているバージョンと同じかそれより前のバージョンの IBM MQ からのみマイグレーションできます。 オペレーティング・システムとアーキテクチャーも同じでなければなりません。 それ以外の場合は、ターゲット・プラットフォーム上に新しいキュー・マネージャーを作成する必要があります。別のオペレーション・システムへのキュー・マネージャーの移動を参照してください。

キュー・マネージャーをマイグレーションする前に、以下の条件を満たす必要があります。

- 高可用性の要件を評価し、564 ページの『RDQM 高可用性』を参照してください。
- キュー・マネージャーに接続するアプリケーションおよびキュー・マネージャーを確認します。 キュー・マネージャーが実行されている RDQM ノードに接続をルーティングするために必要な変更を検討します。 例えば、RDQM 高可用性を構成する場合は、浮動 IP アドレスの使用を検討します。582 ページの『浮動 IP アドレスの作成および削除』を参照してください。

- 選択した構成の RDQM ノードをプロビジョンするか、既存の RDQM ノードを識別します。 RDQM のシステム要件について詳しくは、566 ページの『RDQM HA ソリューションの要件』を参照してください。
- 各ノードに、RDQM フィーチャーを含む IBM MQ Advanced をインストールします。
- RDQM HA グループ構成を構成します。570 ページの『Pacemaker クラスター (HA グループ) の定義』を参照してください。
- オプションで、テスト・キュー・マネージャーを使用して RDQM 構成を検証します。このキュー・マネージャーはあとで削除できます。 キュー・マネージャーをマイグレーションする前に問題を特定して解決するために、構成をテストすることをお勧めします。
- キュー・マネージャーのセキュリティー構成を確認し、必要なローカル・ユーザーおよびグループを各 RDQM ノードに複製します。
- キュー・マネージャーおよびチャネル構成を調べて、API 出口、チャネル出口、またはデータ変換出口が使用されているかどうかを判別します。 必要な出口を各 RDQM ノードにインストールします。
- 定義されているキュー・マネージャー・サービスを確認し、必要なプロセスを各 RDQM ノードでインストールして構成します。

手順

1. 既存のキュー・マネージャーをバックアップします。
   a) 待機シャットダウン・コマンド `endmqm -w` または即時シャットダウン・コマンド `endmqm -i` を発行して、既存のキュー・マネージャーを停止します。 このステップは、バックアップ内のデータの整合性を確保するために重要です。
   b) IBM MQ 構成ファイル `mqs.ini` を表示して、キュー・マネージャーのデータ・ディレクトリーの位置を判別します。 Linux では、ファイルは`/var/mqm` ディレクトリー内にあります。 `mqs.ini` について詳しくは、86 ページの『IBM MQ 構成ファイル mqs.ini』を参照してください。

      ファイル内でキュー・マネージャーの `QueueManager` スタンザを見つけます。 スタンザに `DataPath` という名前のキーが含まれている場合、その値はキュー・マネージャーのデータ・ディレクトリーです。 キーが存在しない場合、キュー・マネージャーのデータ・ディレクトリーは、`Prefix` キーと `Directory` キーの値を使用して判別できます。 キュー・マネージャーのデータ・ディレクトリーは、これらの値を *prefix*/qmgrs/*directory* の形式で連結したものです。 QueueManager スタンザについて詳しくは、96 ページの『mqs.ini ファイルの QueueManager スタンザ』を参照してください。
   c) キュー・マネージャーのデータ・ディレクトリーのバックアップを作成します。 Linux の場合、これを行うには **tar** コマンドを使用します。 例えば、キュー・マネージャーのデータ・ディレクトリーをバックアップするには、以下のコマンドを使用できます。 コマンドの最後のパラメーターである単一ピリオド (ドット) に注意してください。

      ```
      tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
      ```

   d) IBM MQ キュー・マネージャー構成ファイル `qm.ini` を表示して、キュー・マネージャーのログ・ディレクトリーの位置を判別します。 このファイルは、キュー・マネージャーのデータ・ディレクトリーにあります。 このファイルについて詳しくは、98 ページの『キュー・マネージャー構成ファイル qm.ini』を参照してください。

      キュー・マネージャーのログ・ディレクトリーは、`Log` スタンザの `LogPath` キーの値として定義されます。 スタンザについて詳しくは、132 ページの『qm.ini ファイルの Log スタンザ』を参照してください。
   e) キュー・マネージャーのログ・ディレクトリーのバックアップを作成します。 Linux の場合、これを行うには tar コマンドを使用します。 例えば、キュー・マネージャーのログ・ディレクトリーをバックアップするには、以下のコマンドを使用できます。 コマンドの最後のパラメーターである単一ピリオド (ドット) に注意してください。

      ```
      tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
      ```

   f) キュー・マネージャーが使用する証明書リポジトリーがキュー・マネージャーのデータ・ディレクトリーにない場合は、そのバックアップを作成します。 鍵データベース・ファイルとパスワード・ス

タッシュ・ファイルの両方がバックアップされていることを確認してください。 キュー・マネージャーの鍵リポジトリーについては、SSL/TLS 鍵リポジトリーおよびキュー・マネージャーの鍵リポジトリーの検索を参照してください。 AMS メッセージ・チャネル・エージェント (MCA) インターセプトを使用するようにキュー・マネージャーが構成されている場合の AMS 鍵ストアの検索については、メッセージ・チャネル・エージェント (MCA) インターセプトを参照してください。

g) 既存のキュー・マネージャーは不要になったため、削除できます。 ただし、可能であれば、既存のキュー・マネージャーは、ターゲット・システムで正常にリストアされた後にのみ削除してください。 削除を保留しておくと、マイグレーション・プロセスが正常に完了しない場合にキュー・マネージャーを再始動できます。

注 : 既存のキュー・マネージャーの削除を保留する場合は、再始動しないでください。 構成または状態に追加で加えられる変更がマイグレーション中に失われるため、キュー・マネージャーは終了したままであることが重要です。

2. 1 次 RDQM ノードを準備します。

a) バックアップしたキュー・マネージャーと同じ名前の新規 RDQM キュー・マネージャーを作成します。 **crtmqm** によって RDQM キュー・マネージャーに割り振られたファイル・システムが、既存のキュー・マネージャーのデータ、1 次ログ、2 次ログ、および将来の拡張用の追加スペースを含めるのに十分な大きさであることを確認します。 RDQM キュー・マネージャーを作成する方法について詳しくは、573 ページの『HA RDQM の作成』を参照してください。

b) キュー・マネージャーの 1 次 RDQM ノードを判別します。 1 次ノードの判別方法については、rdqmstatus (RDQM 状況の表示) を参照してください。

c) 1 次 RDQM ノードで RDQM キュー・マネージャーが開始されている場合は、`endmqm -w` コマンドまたは `endmqm -i` コマンドを使用して停止します。

d) 1 次 RDQM ノードで、RDQM キュー・マネージャーのデータ・ディレクトリーとログ・ディレクトリーの場所を判別します (ステップ 1b および 1d で説明されている方法を使用します)。

e) 1 次 RDQM ノードで、RDQM キュー・マネージャーのデータ・ディレクトリーとログ・ディレクトリーの内容を削除します。ただし、ディレクトリー自体は削除しないでください。

3. 1 次 RDQM ノードでキュー・マネージャーをリストアします。

a) キュー・マネージャーのデータ・ディレクトリーとログ・ディレクトリーのバックアップを 1 次 RDQM ノードにコピーし、キュー・マネージャーが使用する証明書リポジトリーの別個のバックアップをコピーします。

b) キュー・マネージャーのデータ・ディレクトリーのバックアップを新しい RDQM キュー・マネージャーの空のデータ・ディレクトリーにリストアし、ファイルの所有権と許可が保持されるようにします。 ステップ 1c のサンプル tar コマンドを使用してバックアップを作成した場合は、root ユーザーが以下のコマンドを使用してバックアップをリストアできます。

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

c) キュー・マネージャーのログ・ディレクトリーのバックアップを新しい RDQM キュー・マネージャーの空のログ・ディレクトリーにリストアし、ファイルの所有権と許可が保持されるようにします。 ステップ 1e のサンプル tar コマンドを使用してバックアップを作成した場合は、root ユーザーが以下のコマンドを使用してバックアップをリストアできます。

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

d) RDQM キュー・マネージャーのデータ・ディレクトリーに、リストアされたキュー・マネージャー構成ファイル(`qm.ini`)を編集します。 `Log` スタンザ内の `LogPath` キーの値を更新して、RDQM キュー・マネージャーのログ・ディレクトリーを指定します。

構成ファイルに定義されている他のファイル・パスを確認し、必要に応じて更新します。 例えば、以下のパスを更新しなければならない可能性があります。

- 診断メッセージ・サービスによって生成されるエラー・ログ・ファイルのパス。
- キュー・マネージャーで必要な出口のパス。

- キュー・マネージャーが XA トランザクション・コーディネーターである場合のスイッチ・ロード・ファイルのパス。

e) AMS メッセージ・チャネル・エージェント (MCA) インターセプトを使用するようにキュー・マネージャーが構成されている場合は、AMS 鍵ストアを新しい RDQM インストール済み環境にコピーしてから、構成を確認して更新します。 鍵ストアは各 RDQM ノードで使用可能でなければならないため、キュー・マネージャーの複製ファイル・システムにない場合は、代わりに各ノードにコピーする必要があります。 詳細については、メッセージ・チャネル・エージェント (MCA) インターセプトを参照してください。

f) **dspmq** コマンドによってキュー・マネージャーが表示され、その状況が「終了」と報告されていることを確認します。 以下の例は、RDQM HA キュー・マネージャーの出力例を示しています。

```
$ dspmq -o status -o ha

QMNAME(QM1) STATUS(Ended normally) HA(Replicated)
```

g) **rdqmstatus** コマンドを使用してキュー・マネージャーの状況を表示することにより、リストアされたキュー・マネージャー・データが 2 次 RDQM ノードに複製されたことを確認します。 HA 状況は、各ノードで Normal として報告される必要があります。 以下の例は、RDQM HA キュー・マネージャーの出力例を示しています。

```
$ rdqmstatus -m QM1
Node:                           mqhavm10-adm
Queue manager status:           Ended normally
Queue manager file system:      50MB used, 0.2GB allocated [42%]
HA role:                        Primary
HA status:                      Normal
HA control:                     Disabled
HA current location:            This node
HA preferred location:          This node
HA floating IP interface:       None
HA floating IP address:         None

Node:                           mqhavm11-adm
HA status:                      Normal

Node:                           mqhavm12-adm
HA status:                      Normal
```

h) 1 次 RDQM ノードでキュー・マネージャーを開始します。

i) キュー・マネージャーに接続し、SSLKEYR キュー・マネージャー属性の値を更新して、キュー・マネージャー証明書リポジトリーの新しい場所を指定します。 デフォルトで、この属性の値には *queue_manager_data_directory*/ssl/key が設定されます。 証明書リポジトリーは、各 RDQM ノード上の同じ場所に配置する必要があります。 リポジトリーがキュー・マネージャーの複製ファイル・システムにない場合は、代わりに各ノードにコピーする必要があります。

j) キュー・マネージャーの IBM MQ オブジェクト定義を確認し、変更されたネットワーク設定、IBM MQ インストール・ディレクトリー、またはキュー・マネージャー・データ・ディレクトリーを参照するオブジェクト属性の値を更新します (以下のオブジェクトを含む)。

- リスナーが使用するローカル IP アドレス (IPADDR 属性)。
- チャネルが使用するローカル IP アドレス (LOCLADDR 属性)。
- クラスター受信側チャネル用に定義されたローカル IP アドレス (CONNAME 属性)。
- コミュニケーション情報オブジェクト用に定義されたローカル IP アドレス (GRPADDR 属性)。
- プロセスおよびサービス・オブジェクト定義用に定義されたシステム・パス。

k) キュー・マネージャーをいったん停止したあとで再始動して、変更内容が有効になるようにします。

l) マイグレーションされたキュー・マネージャーに接続するリモート・キュー・マネージャーと、それに相当するアプリケーションの設定 (以下を含む) について、ステップ 3j を繰り返します。

- チャネル接続名 (CONNAME 属性)。
- IP アドレスまたはホスト名に基づいてキュー・マネージャーからのインバウンド接続を制限するチャネル認証規則。

- クライアント・チャネル定義テーブル (CCDT)、ドメイン・ネーム設定 (DNS)、ネットワーク・ルーティング、または同等の接続情報。

m) 各 RDQM ノードへのキュー・マネージャーの管理フェイルオーバーを実行して、必要な構成が正常に確立されたことを確認します。581 ページの『RDQM の優先ロケーションの設定』を参照してください。

HA RDQM キュー・マネージャーのファイル・システムのサイズ変更
既存の高可用性 (HA) 複製データ・キュー・マネージャー (RDQM) のファイル・システムをサイズ変更するには、その永続データをバックアップしてから、同じ名前で、サイズが異なるファイル・システムを持つ新しく作成された RDQM キュー・マネージャーにデータをリストアします。

このタスクについて

HA 複製データ・キュー・マネージャーには、専用の論理ボリューム (ファイル・システム) と、ディスク複製および HA 制御の構成が必要です。 これらのコンポーネントは、新規キュー・マネージャーが作成されたときにのみ構成されます。 ファイル・システムは、各ノードで同じサイズでなければならないため、作成後にサイズ変更することはできません。 既存の複製データ・キュー・マネージャー (RDQM) のファイル・システムをサイズ変更するには、その永続データをバックアップしてから、同じ名前で、サイズが異なるファイル・システムを持つ新しく作成された RDQM キュー・マネージャーにデータをリストアします。 この手順では、バックアップの作成時にキュー・マネージャーの構成、状態、および持続メッセージを保存します。

手順

1. 1 次 RDQM ノード上の既存の RDQM キュー・マネージャーをバックアップします。

   a) キュー・マネージャーの 1 次 RDQM ノードを判別します。 1 次ノードの判別方法については、rdqmstatus (RDQM 状況の表示) を参照してください。

   b) 1 次 RDQM ノードで RDQM キュー・マネージャーが開始されている場合は、**endmqm -w** コマンドまたは **endmqm -i** コマンドを使用して停止します。

   c) IBM MQ 構成ファイル `mqs.ini` を表示して、キュー・マネージャーのデータ・ディレクトリーの位置を判別します。 Linux では、ファイルは`/var/mqm` ディレクトリー内にあります。 `mqs.ini` について詳しくは、86 ページの『IBM MQ 構成ファイル mqs.ini』を参照してください。

      ファイル内でキュー・マネージャーの `QueueManager` スタンザを見つけます。 キュー・マネージャーのデータ・ディレクトリーは、`DataPath` というキーの値です。 `QueueManager` スタンザについて詳しくは、96 ページの『mqs.ini ファイルの QueueManager スタンザ』を参照してください。

   d) キュー・マネージャーのデータ・ディレクトリーのバックアップを作成します。 Linux の場合、これを行うには **tar** コマンドを使用します。 例えば、キュー・マネージャーのデータ・ディレクトリーをバックアップするには、以下のコマンドを使用できます。 コマンドの最後のパラメーターである単一ピリオド (.) に注意してください。

      ```
      tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
      ```

   e) IBM MQ キュー・マネージャー構成ファイル `qm.ini` を表示して、キュー・マネージャーのログ・ディレクトリーの位置を判別します。 このファイルは、キュー・マネージャーのデータ・ディレクトリーにあります。 このファイルについて詳しくは、98 ページの『キュー・マネージャー構成ファイル qm.ini』を参照してください。

      キュー・マネージャーのログ・ディレクトリーは、`Log` スタンザの `LogPath` キーの値として定義されます。 スタンザについて詳しくは、132 ページの『qm.ini ファイルの Log スタンザ』を参照してください。

   f) キュー・マネージャーのログ・ディレクトリーのバックアップを作成します。 Linux の場合、これを行うには **tar** コマンドを使用します。 例えば、キュー・マネージャーのログ・ディレクトリーをバックアップするには、以下のコマンドを使用できます。 コマンドの最後のパラメーターである単一ピリオド (.) に注意してください。

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

g) 既存の RDQM キュー・マネージャーを削除してください。

2. 必要なサイズのファイル・システムでキュー・マネージャーをリストアします。

a) バックアップしたキュー・マネージャーと同じ名前の新規 RDQM キュー・マネージャーを作成します。 **crtmqm** によって RDQM キュー・マネージャーに割り振られたファイル・システムが、必要なサイズで、既存のキュー・マネージャーのデータ、1 次ログ、2 次ログ、および将来の拡張用の追加スペースを含めるのに十分な大きさであることを確認します。 RDQM キュー・マネージャーを作成する方法について詳しくは、573 ページの『HA RDQM の作成』を参照してください。

b) キュー・マネージャーの 1 次 RDQM ノードを判別します。 1 次ノードの判別方法については、rdqmstatus (RDQM 状況の表示) を参照してください。

c) 1 次 RDQM ノードで、RDQM キュー・マネージャーが開始されている場合は、 **endmqm -w** または **endmqm -i** コマンドを使用して停止します。

d) 1 次 RDQM ノードで、RDQM キュー・マネージャーのデータ・ディレクトリーとログ・ディレクトリーの新しい場所を判別します (ステップ 1c および 1e で説明されている方法を使用します)。

e) 1 次 RDQM ノードで、RDQM キュー・マネージャーのデータ・ディレクトリーとログ・ディレクトリーの内容を削除します。ただし、ディレクトリー自体は削除しないでください。

f) 1 次 RDQM ノードで、キュー・マネージャーのデータ・ディレクトリーのバックアップを新しい RDQM キュー・マネージャーの空のデータ・ディレクトリーにリストアし、ファイルの所有権と許可が保持されるようにします。 ステップ 1d のサンプル **tar** コマンドを使用してバックアップを作成した場合は、root ユーザーが以下のコマンドを使用してバックアップをリストアできます。

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

g) 1 次 RDQM ノードで、キュー・マネージャーのログ・ディレクトリーのバックアップを新しい RDQM キュー・マネージャーの空のログ・ディレクトリーにリストアし、ファイルの所有権と許可が保持されるようにします。 ステップ 1f のサンプル **tar** コマンドを使用してバックアップを作成した場合は、root ユーザーが以下のコマンドを使用してバックアップをリストアできます。

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

h) 一次 RDQM ノードで、リストアされたキュー・マネージャー構成ファイル `qm.ini` を、新しい RDQM キュー・マネージャーのデータ・ディレクトリーに編集します。 `Log` スタンザ内の `LogPath` キーの値を更新して、ステップ 2d で判別した新しい RDQM キュー・マネージャーのログ・ディレクトリーを指定します。 構成ファイルに定義されている他のファイル・パスを確認し、必要に応じて更新します。 例えば、以下のパスを更新しなければならない可能性があります。

- 診断メッセージ・サービスによって生成されるエラー・ログ・ファイルのパス。
- キュー・マネージャーで必要な出口のパス。
- キュー・マネージャーが XA トランザクション・コーディネーターである場合のスイッチ・ロード・ファイルのパス。

i) **dspmq** コマンドによってキュー・マネージャーが表示され、その状況が「終了」と報告されていることを確認します。 以下の例は、RDQM HA キュー・マネージャーの出力例を示しています。

```
$ dspmq -o status -o ha
QMNAME(QM1) STATUS(Ended normally) HA(Replicated)
```

j) **rdqmstatus** コマンドを使用してキュー・マネージャーの状況を表示することにより、リストアされたキュー・マネージャー・データが 2 次 RDQM ノードに複製されたことを確認します。 HA 状況は、各ノードで `Normal` として報告される必要があります。 以下の例は、RDQM HA キュー・マネージャーの出力例を示しています。

```
$ rdqmstatus -m QM1
Node: mqhavm10-adm
Queue manager status:         Ended normally
Queue manager file system:    50MB used, 0.2GB
allocated [42%]
HA role:                      Primary
```

```
HA status:                    Normal
HA control:                   Disabled
HA current location:          This node
HA preferred location:        This node
HA floating IP interface:     None
HA floating IP address:       None
Node:                         mqhavm11-adm
HA status:                    Normal
Node:                         mqhavm12-adm
HA status:                    Normal
```

k) 1 次 RDQM ノードでキュー・マネージャーを開始します。

l) 各 RDQM ノードへのキュー・マネージャーの管理フェイルオーバーを実行して、必要な構成が正常に確立されたことを確認します。581 ページの『RDQM の優先ロケーションの設定』を参照してください。

アプリケーションの持続状況の保管
アプリケーションに関連する持続状況情報を、他のキュー・マネージャー・データと一緒に保管できます。

各 IBM MQ キュー・マネージャーには、その永続的な状態の専用ファイル・システムがあります (キュー・データとリカバリー・ログの両方を含む)。 RDQM 構成では、ファイル・システムは Linux システム (ノード) 間で複製される論理ボリュームを基盤としています。 ファイル・システムには、アプリケーションの永続状況情報を保管するために使用できる userdata ディレクトリーが含まれています。 したがって、複製データ・キュー・マネージャーが、RDQM 構成内の別のノードで実行するよう移動されると、キュー・マネージャー・コンテキストだけでなく、アプリケーション・コンテキストも使用できます。 Unix および Linux システムでのディレクトリーの内容を参照してください。

アプリケーション状態を userdata ディレクトリーに保管することを選択した場合は、このロケーションに書き込まれるデータが、キュー・マネージャーに割り振られた使用可能なディスク・スペースを消費する可能性があることを認識しておく必要があります。 キュー・マネージャーがキュー・データ、ログ、およびその他の永続的な状態の情報を書き込むために使用できるだけの十分なディスク・スペースが残っていることを確認する必要があります。

userdata ディレクトリーには mqm ユーザーおよびグループの所有権があり、ユーザーは IBM MQ 管理者グループ(つまり mqm)でなくても、ユーザーがアクセスできるようになっています。 userdata ディレクトリーの許可を変更することはできませんが、必要な所有権および許可を使用して、ディレクトリーにコンテンツを作成することができます。

RDQM キュー・マネージャーのフェイルオーバー中に、キュー・マネージャーが終了し、そのファイル・システムが現行 RDQM ノード上でアンマウントされます。 その後、ファイル・システムがマウントされ、RDQM 構成内の別のノードでキュー・マネージャーが再始動されます。 プロセスがそのファイルの 1 つに対してオープン・ハンドルを持っている場合は、ファイル・システムをアンマウントできません。 キュー・マネージャーのフェイルオーバーを確実に完了できるようにするため、キュー・マネージャーのファイル・システムをアンマウントできない場合、オープン・ファイル・ハンドルを持つプロセスには SIGTERM シグナルが送信され、その後、オープン・ハンドルが解放されない場合は SIGKILL が続きます。 アプリケーションは、SIGTERM に正しく応答するように設計されている必要があります。 アプリケーションまたはプロセスがキュー・マネージャー・サービスとして構成されている場合は、管理フェイルオーバー時に、ファイル・システムがアンマウントされる前に、キュー・マネージャーのシャットダウン中に終了できます。 アプリケーションまたはプロセスがキュー・マネージャー・サービスとして構成されていない場合、または非管理フェイルオーバーが発生した場合 (クォーラムの喪失など)、シグナルがファイル・システムを解放するために送信される可能性があります。

Linux *RDQM の優先ロケーションの設定*

複製データ・キュー・マネージャー (RDQM) の優先ロケーションは、そのノードが使用可能な場合に RDQM が実行されるノードを識別します。

このタスクについて

優先ロケーションは、HA グループが通常の状態 (すべてのノードおよび接続が使用可能) である場合に、Pacemaker がキュー・マネージャーを実行する必要があるノードの名前です。 優先ロケーションは、キュー・マネージャーの作成時に 1 次ノードの名前に初期化されます。 コマンドを実行して、3 つのノードの

いずれかに優先ロケーションを設定できます。 この場合、mqm グループおよび haclient グループに属するユーザーである必要があります。

手順

- ローカル・ノードまたは指定したノードを、指定したキュー・マネージャーの優先ロケーションとして割り当てるには、以下のコマンドを入力します。

  ```
  rdqmadm -p -m qmname [ -n nodename[,nodename  ]
  ```

  ここで、*qmname* は優先ロケーションを指定する RDQM の名前で、*nodename* はオプションの優先ノードの名前です。

  HA グループが通常の状態で、優先ロケーションが現在の 1 次ノードではない場合、キュー・マネージャーが停止し、新しい優先ロケーションで再始動されます。 2 つのノード名のコンマ区切りリストを指定して、優先ロケーションの 2 つ目の設定を割り当てることができます。
- キュー・マネージャーがリストア時に自動的にノードに戻ることがないように、優先ロケーションをクリアするには、以下のコマンドを入力します。

  ```
  rdqmadm -p -m qmname -d
  ```

関連資料

rdqmadm (複製データ・キュー・マネージャー・クラスターの管理)

Linux 浮動 *IP* アドレスの作成および削除

浮動 IP アドレスを使用すると、HA グループ内のどのノードでクライアントが実行されているかにかかわらず、複製データ・キュー・マネージャー (RDQM) と同じ IP アドレスを使用することができます。 (浮動 IP アドレスの使用は任意です。)

このタスクについて

浮動 IP アドレスを作成または削除するには、**rdqmint** コマンドを使用します。 浮動アドレスは、RDQM の 1 次ノードの指定した物理インターフェースにバインドします。 RDQM がフェイルオーバーし、別のノードが 1 次ノードになると、浮動 IP は、新しい 1 次上の同じ名前のインターフェースにバインドされます。 3 つのノード上の物理インターフェースは、浮動 IP アドレスと同じサブネットに属している必要があります。 以下の図は、浮動 IP アドレスの使用を示しています。

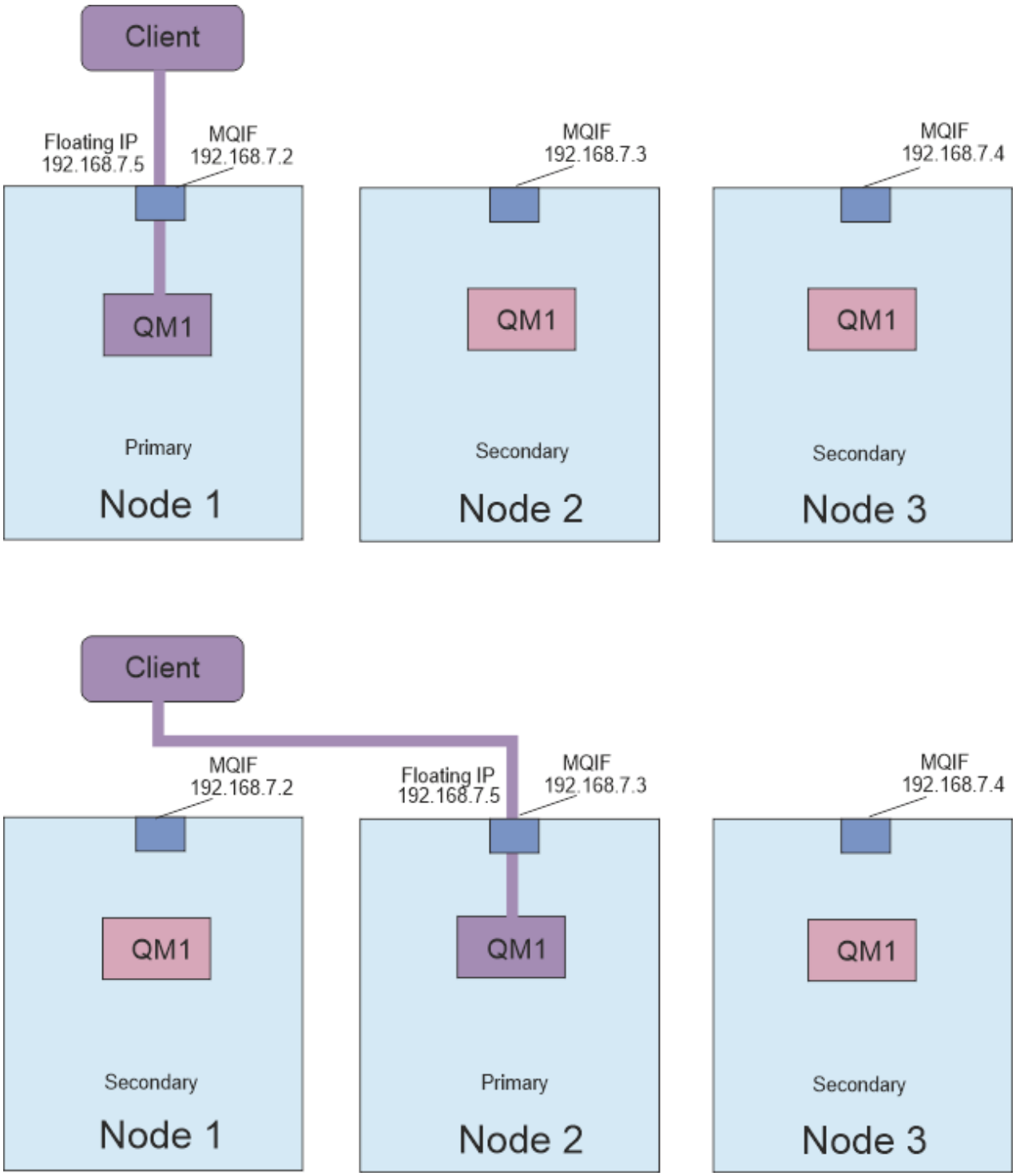


図 *80.* 浮動 *IP* アドレス

rdqmint コマンドを実行するには、mqm グループと haclient グループの両方のユーザーである必要があります。 RDQM の 1 次ノードまたは 2 次ノードのいずれかで浮動 IP アドレスを作成または削除できます。

注 : 複数の RDQM に同じ浮動 IP アドレスを使用することはできません。それぞれの RDQM の浮動 IP アドレスは固有でなければなりません。

手順

- RDQM の浮動 IP アドレスを作成するには、以下のコマンドを入力します。

```
rdqmint -m qmname -a  -f ipv4address -l interfacename
```

ここで、

qmname
浮動 IP アドレスを作成する RDQM の名前。

ipv4address
ipv4 形式の浮動 IP アドレス。

この浮動 IP アドレスは、どちらの HA ノードにもまだ定義されていない有効な IPv4 アドレスでなければならず、またローカル・インターフェースに定義されている静的 IP アドレスと同じサブネットに属していなければなりません。

interfacename
1 次ノードがバインドする物理インターフェースの名前。

以下に例を示します。

```
rdqmint -m QM1 -a -f 192.168.7.5 -l MQIF
```

- 既存の浮動 IP アドレスを削除するには、以下のコマンドを入力します。

```
rdqmint -m qmname -d
```

関連資料
rdqmint (RDQM の浮動 IP アドレスの追加または削除)

Linux *HA RDQM の開始、停止、および状態の表示*

標準 IBM MQ 制御コマンドのバリアントを使用して、複製データ・キュー・マネージャー (RDQM) を開始または停止したり、現在の状態を表示したりできます。

このタスクについて

mqm グループと haclient グループの両方に属するユーザーとして、複製データ・キュー・マネージャー (RDQM) を開始または停止したり、現在の状態を表示したりするコマンドを実行する必要があります。

そのキュー・マネージャーの 1 次ノードでキュー・マネージャーを開始および停止するには、コマンドを実行する必要があります。

手順

- RDQM を開始するには、RDQM の 1 次ノードで以下のコマンドを入力します。

```
strmqm qmname
```

ここで、*qmname* は、開始する RDQM の名前です。

RDQM が開始され、Pacemaker が RDQM の管理を開始します。 その他の strmqm オプションを指定する場合は、strmqm と共に -ns オプションを指定する必要があります。

- RDQM を停止するには、RDQM の 1 次ノードで以下のコマンドを入力します。

```
endmqm qmname
```

ここで、*qmname* は、停止する RDQM の名前です。

Pacemaker は、RDQM の管理を停止し、次に RDQM が終了します。 他のすべての **endmqm** パラメーターは、RDQM を停止するときに使用できます。

- RDQM の状態を表示するには、以下のコマンドを入力します。

```
dspmq
```

出力される状態情報は、コマンドを RDQM の 1 次ノードまたは 2 次ノードのいずれで実行するかによって異なります。 1 次ノードで実行する場合は、**dspmq** によって返された正常状況メッセージの 1 つが表示されます。 2 次ノードでコマンドを実行する場合は、状況 `running elsewhere` が表示されます。 例えば、**dspmq** がノード RDQM7 で実行されている場合、以下の情報が返されることがあります。

```
QMNAME(RDQM8)                           STATUS(Running elsewhere)
QMNAME(RDQM9)                           STATUS(Running elsewhere)
QMNAME(RDQM7)                           STATUS(Running)
```

1 次ノードが使用不可の場合、または **dspmq** が `root` ではないユーザーまたは `haclient` グループのメンバーによって実行された場合、`Unavailable` 状態が報告されます。 以下に例を示します。

```
QMNAME(RDQM8)               STATUS(Unavailable)
QMNAME(RDQM9)               STATUS(Unavailable)
QMNAME(RDQM7)               STATUS(Unavailable)
```

コマンド **dspmq -o ha** (または **dspmq -o HA**) を入力して、ノードで認識されているキュー・マネージャーのリストやキュー・マネージャーが RDQM であるかどうかなどを表示できます。

```
dspmq -o ha

QMNAME(RDQM8)                                           HA(Replicated)
QMNAME(RDQM9)                                           HA(Replicated)
QMNAME(RDQM7)                                           HA(Replicated)
QMNAME(QM7)                                             HA()
```

関連資料

dspmq (キュー・マネージャーの表示)

endmqm (キュー・マネージャーの終了)

strmqm (キュー・マネージャーの始動)

V 9.3.0 失敗したリソースのアクション

RDQM 高可用性構成の Pacemaker コンポーネントが、HA グループ内のいずれかのノード上のリソースに関連した何らかの問題を検出すると、失敗したリソースのアクションが発生します。

RDQM HA ソリューションは、Pacemaker を使用してリソースをモニターおよび管理します (564 ページの『RDQM 高可用性』を参照)。 Pacemaker は、ノード上のリソースで特定の操作を実行したときにエラーを検出した場合、失敗したリソース・アクションを使用してこの情報を記録します。 失敗したリソース・アクションの一部は、リソースの実行を妨げるため、Pacemaker がリソースを再始動する前にクリアする必要があります。

rdqmstatus -m コマンドを使用して、キュー・マネージャーが 1 つ以上のノードで開始するのを妨げている、失敗したリソース・アクションが存在するかどうかを確認できます。

次に、**rdqmstatus -m** *qmname* **-a** コマンドを使用して、キュー・マネージャーと関連付けられた、失敗したリソース・アクションの詳細を表示できます。 **rdqmclean** コマンドを使用してこのアクションを実行し、失敗したこれらのリソース・アクションをクリアして、制限されているリソースをすべて解放します。 (また、最初の場所で失敗したリソース・アクションの原因となった問題を解決するためのアクションを実行する必要もあります。)

以下のリソースは、RDQM HA 構成内で Pacemaker によって制御されており、失敗したリソース・アクションの対象となる可能性があります。

- キュー・マネージャー
- 浮動 IP
- RDQM コントロール
- ファイル・システム
- DR レプリケーション (DRBD)
- HA レプリケーション (DRBD)

各リソース・タイプは、以下の失敗のタイプの対象になります。

ソフト
ソフト失敗は一時的なものであり、Pacemaker はタイムアウトするか、停止されるまで、そのリソースのリカバリーを試行し続けます。

ハード
ハード・エラーは管理介入を必要とします。 ハード・エラーが発生すると、特定のノード上でのリソースの実行がブロックされます。

致命的
致命的エラーは管理介入を必要とします。 致命的エラーが発生すると、すべてのノード上でのリソースの実行がブロックされます。

失敗したリソース・キュー・アクションを含む状況の例については、586 ページの『RDQM および HA グループの状況の表示』を参照してください。

rdqmclean コマンドを使用して、指定したキュー・マネージャーに関連付けられているすべての失敗したリソース・アクション、または RDQM HA 構成内のすべての失敗したリソース・アクションをクリアできます。

注 : 一部の失敗したリソース・アクションでは、キュー・マネージャーはノードでブロックされません。 例えば、予期しないキュー・マネージャーの終了後、Pacemaker は、実行されていないことが判明したノードでキュー・マネージャーを再起動しようとします。 起動に成功した場合、そのキュー・マネージャーはノードでの実行がブロックされません。 この場合、失敗したリソース・アクションを把握するための唯一の方法は、**rdqmstatus -m** *qmname* **-a** を実行することです。

関連タスク
586 ページの『RDQM および HA グループの状況の表示』
HA グループおよび個々の複製データ・キュー・マネージャー (RDQM) の状況を表示できます。

関連資料
rdqmclean
rdqmstatus

Linux *RDQM および HA グループの状況の表示*

HA グループおよび個々の複製データ・キュー・マネージャー (RDQM) の状況を表示できます。

このタスクについて

個々の RDQM および HA グループ全体の状況を表示するには、**rdqmstatus** コマンドを使用します。

V 9.3.0 ノードの要約状況には、RDQM が依存している DRBD カーネル・モジュールに関する情報も表示されます。 RDQM をアップグレードする場合は、システム上で実行されている RHEL カーネルのバージョン用に正しいバージョンの DRBD カーネル・モジュールがインストールされていることを確認することが重要です。 この状況には、OS カーネルのバージョン、DRBD モジュールが作成されたカーネル・バージョン、DRBD バージョン、および DRBD カーネル・モジュールのロード状況が表示されます。

rdqmstatus コマンドを実行するには、mqm グループと haclient グループのユーザーである必要があります。 このコマンドは、3 つのノードのいずれでも実行できます。

手順

- HA 構成の一部であるノードおよび RDQM の要約状況を表示するには、次のようにします。

```
rdqmstatus
```

コマンドを実行したノードの ID 、そのノードのカーネルと DRBD の詳細、および HA 構成の RDQM の状況が、以下のように表示されます。

```
Node:                                 mqhavm07.exampleco.com
OS kernel version:                    3.10.0-1160.15.2
DRBD OS kernel version:               3.10.0-1160
DRBD version:                         9.1.1
DRBD kernel module status:            Loaded
```

```
Queue manager name:                     RDQM8
Queue manager status:                   Running elsewhere
HA current location:                    mqhavm08.exampleco.com
HA preferred location:                  mqhavm08.exampleco.com
HA blocked location:                    None

Queue manager name:                     RDQM9
Queue manager status:                   Running elsewhere
HA current location:                    mqhavm09.exampleco.com
HA preferred location:                  mqhavm09.exampleco.com
HA blocked location:                    None

Queue manager name:                     RDQM7
Queue manager status:                   Running
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
```

V 9.3.0 DRBD カーネル・モジュール状況は、以下のいずれかの値です。

ロード済み (Loaded)
DRBD モジュールがロードされたことを示します。

一部ロード (Partially loaded)
DRBD モジュールがロードされたが、不一致が原因で正しく機能しない場合に発生する可能性があります。

ロードされていません (Not loaded)
DRBD モジュールはロードされていません。 これは、RDQM キュー・マネージャーがまだ作成されていない場合に、新規にインストールされた構成で表示される可能性があります。

インストールされていません
DRBD モジュールがインストールされていないか、 IBM MQ が DRBD モジュールの OS カーネル・バージョンを判別できなかったことを示します。

以前にインストールされたバージョンがまだロードされています (Previously installed version still loaded)
この状況は、既存の DRBD モジュールの実行中 (つまり、RDQM キュー・マネージャーの実行中) に新規 DRBD モジュールがインストールされた場合に発生する可能性があります。 新規にインストールされたモジュールが状況で報告されますが、これは実際に実行されているモジュールではありません。

- HA グループの 3 つのノードの状況を表示するには、以下のコマンドを入力します。

```
rdqmstatus -n
```

各ノードのオンラインまたはオフライン状況が報告されます。 以下に例を示します。

```
Node mqha04(mqhavm04.example.com) is online
Node mqha05(mqhavm05.example.com) is offline
Node mqha06(mqhavm06.example.com) is online
```

- HA グループのすべてのノードの特定のキュー・マネージャーの状況を表示するには、以下のコマンドを入力します。

```
rdqmstatus -m qmname
```

ここで、*qmname* は、状況を表示する RDQM の名前です。 現行ノードの RDQM の状況が表示され、その後に現行ノードの観点からの他の 2 つのノードの状況の要約が表示されます。

- V 9.3.0
HA グループのすべてのノードの特定のキュー・マネージャーの状況を、失敗したリソース・アクションの詳細を含めて表示するには、以下のコマンドを入力します。

```
rdqmstatus -m qmname -a
```

ここで、*qmname* は、状況を表示する RDQM の名前です。 現行ノードの RDQM の状況が表示され、その後に現行ノードの観点からの他の 2 つのノードの状況の要約が表示されます。 この後に、RDQM に関連付けられている失敗したリソース・アクションに関する詳細が続きます。

- 以下の表は、RDQM の `rdqmstatus -m` *qmname* コマンドによって返される可能性がある現行ノードに関する情報を要約したものです。

表 *32.* 現行ノードの状況

状況属性	**可能な値**	**表示される時期**
ノード名	*nodename*	常に表示されます。
キュー・マネージャー状況	実行中 別の場所で実行中 Ended 使用不可	常に表示されます。
CPU	*n.nn*%	現行ノードに 1 次の役割がある場合にのみ表示されます (つまり、RDQM がそのノードで実行されています)。
メモリー	*nnn*MB used、*y.y*GB allocated	現行ノードに 1 次の役割がある場合にのみ表示されます (つまり、RDQM がそのノードで実行されています)。
キュー・マネージャー・ファイル・システム	*nnn*MB used、*y.y*GB allocated [*z*%]	現行ノードに 1 次の役割がある場合にのみ表示されます (つまり、RDQM がそのノードで実行されています)。
HA 役割	Primary Secondary Unknown	常に表示されます。
HA 状況	すべてのノードがスタンバイ状態です このノードはスタンバイ状態です Remote nodes in standby 混合 リモート・ノードの状況	すべてのノードがスタンバイ状態です 現行ノードがスタンバイ状態の場合 両方のリモート・ノードがスタンバイ状態の場合 リモート・ノードごとに状況が異なる場合 (個々の状況については次の表を参照) 両方のリモート・ノードの状況が同じ場合 (すべての値については次の表を参照)
HA 制御	有効 無効 不明	常に表示されます。 RDQM が Pacemaker 制御下にあるかどうかが示されます。
HA 優先ロケーション	なし このノード 不明 *nodename*	常に表示されます。

表 32. 現行ノードの状況 (続き)		
状況属性	**可能な値**	**表示される時期**
V 9.3.0 V 9.3.0 HA ブロック・ロケーション	なし - このキュー・マネージャーはどのノードでも実行がブロックされることはありません。 このノード - 1 つ以上のリソース・アクションが失敗したため、現在のノードでのキュー・マネージャーの実行がブロックされています。 *nodename* - 1 つ以上のリソース・アクションが失敗したため、*nodename* でのキュー・マネージャーの実行がブロックされています。 *nodename1*、*nodename2* - 1 つ以上のリソース・アクションが失敗したため、*nodename1* および *nodename2* でのキュー・マネージャーの実行がブロックされています。 すべてのノード - 1 つ以上のリソース・アクションが失敗したため、すべてのノードでキュー・マネージャーの実行がブロックされています。	常に表示されます。
HA 浮動 IP インターフェース	*Interface_name*	常に表示されます。
HA 浮動 IP アドレス	*IPV4_address*	常に表示されます。

以下の表は、HA グループ内の他のノードに対する `rdqmstatus -m qmname` コマンドによって返される情報の要約を示しています。

表 33. その他のノードの状況		
状況属性	**可能な値**	**表示される時期**
ノード名	*nodename*	常に表示されます。
HA 状況	通常 同期が進行中 リモートは使用不可です 不整合 休止 リモート・ノードはスタンバイ状態です 不明	ノードが互いに同期している場合 リモート・ノードと同期している場合 リモート・ノードと通信できない場合 リモート・ノードとの同期化が行われておらず、同期していない場合 複製が一時停止している場合 リモート・ノードはスタンバイ状態です
HA 同期が進行中	*n.n*%	同期の進行中、コマンドが `root` として実行されたときに表示されます
HA 予想される同期時間	*yyyy-mm-dd hh:mm:ss.nnn*	同期化が進行中の場合に表示されます。
HA 非同期データ	*n*KB	リモート・ノードが使用できないか、または不整合である場合に表示されます。

表 33. その他のノードの状況 (続き)		
状況属性	**可能な値**	**表示される時期**
V 9.3.0 V 9.3.0 最後の HA 同期日時	*yyyy-mm-dd hh:mm:ss.nnn*	HA データが同期していない状況 (初期同期の後) で表示されます。データが最後に同期したときの日時を指定します。

例

1 次ノードの正常状況の例:

```
Node:                                  mqhavm07.exampleco.com
Queue manager status:                   Running
CPU:                                    0.00
Memory:                                 123MB
Queue manager file system:              606MB used, 1.0GB allocated [60%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4


Node:                                   mqhavm08.exampleco.com
HA status:                              Normal


Node:                                   mqhavm09.exampleco.com
HA status:                              Normal
```

2 次ノードの正常状況の例:

```
Node:                                   mqhavm08.exampleco.com
Queue manager status:                   Running elsewhere
HA role:                                Secondary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    mqhavm07.exampleco.com
HA preferred location:                  mqhavm07.exampleco.com
HA blocked location:                    None
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm07.exampleco.com
HA status:                              Normal

Node:                                   mqhavm09.exampleco.com
HA status:                              Normal
```

同期化が進行中の場合の 1 次ノードの状況の例:

```
Node:                                   mqhavm07.exampleco.com
Queue manager status:                   Running
CPU:                                    0.53
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Synchronization in progress
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm08.exampleco.com
HA status:                              Synchronization in progress
HA synchronization progress:            11.0%
```

```
HA estimated time to completion:        2017-09-06 14:55:05

Node:                                   mqhavm09.exampleco.com
HA status:                              Synchronization in progress
HA synchronization progress:            11.0%
HA estimated time to completion:        2017-09-06 14:55:06
```

V 9.3.0 同期が失われた場合の 1 次ノードの状況の例:

```
Node:                                   mqhavm07.exampleco.com
Queue manager status:                   Running
CPU:                                    0.53
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Mixed
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm08.exampleco.com
HA status:                              Normal

Node:                                   mqhavm09.exampleco.com
HA status:                              Inconsistent
HA out of sync data:                    15932KB
HA last in sync:                        2017-09-06 14:55:06
```

複数の状態を示す 1 次ノードの例:

```
Node:                                   mqhavm07.exampleco.com
Queue manager status:                   Running
CPU:                                    0.02
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Mixed
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm08.exampleco.com
HA status:                              Normal

Node:                                   mqhavm09.exampleco.com
HA status:                              Inconsistent
```

V 9.3.0 失敗したリソース・アクションを示す 1 次ノードの例:

```
Node:                                   mqhavm07.exampleco.com
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              606MB used, 1.0GB allocated [60%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  mqhavm08.exampleco.com
HA blocked location:                    mqhavm08.exampleco.com
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm08.exampleco.com
HA status:                              Normal

Node:                                   mqhavm09.exampleco.com
HA status:                              Normal

Failed resource action:                 Start
Resource type:                          Filesystem
```

```
Failure node:                           mqhavm08.exampleco.com
Failure time:                           2017-09-06 12:00:00
Failure reason:                         Couldn't find directory  [/var/mqm/vols/qmname] to use
as a mount point
Blocked location:                       mqhavm08.exampleco.com
```

このステータスは、Pacemaker が 12:00:00 にノード mqhavm08.exampleco.com でファイル・システムを開始できなかったことを示しています。 この失敗したリソース・アクションは、mqhavm08.exampleco.com でのキュー・マネージャーの実行がブロックされていることを意味します。 失敗したリソース・アクションの原因となった根本的な問題が解決された後に **rdqmclean** コマンドを実行して、失敗したアクションを消去し、Pacemaker がアクションを再試行できるようにします (必要な場合)。

V 9.3.0 以下に、OS カーネル・バージョン (RHEL 7.9) と DRBD カーネル・モジュール (ターゲットは RHEL 7.8) の間の不一致を示す要約状況の例を示します。 DRBD カーネル・モジュールがロードされ、キュー・マネージャーが実行されていることが状況に報告される場合でも、この状況で実行されている OS カーネルをターゲットとするバージョンで DRBD カーネル・モジュールを更新する必要があります。

```
Node:                                   mqhavm07.exampleco.com
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-1127
DRBD version:                           9.1.1
DRBD kernel module status:              Loaded

Queue manager name:                     RDQM7
Queue manager status:                   Running
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
```

V 9.3.0 以下に、OS カーネル・バージョン (RHEL 7.9) と DRBD カーネル・モジュール (ターゲットは RHEL 7.6) の間の不一致を示す要約状況の例を示します。 この例では、バージョンの違いがさらに大きいため、DRBD カーネル・モジュールを正常にロードできません。 その結果、キュー・マネージャーは Unknown で優先ノードおよび HA 状況で開始できません。 この障害を解決するには、稼働中の OS カーネルをターゲットとするバージョンで DRBD カーネル・モジュールを更新する必要があります。

```
Node:                                   mqhavm57.exampleco.com
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-957
DRBD version:                           9.1.2+ptf.3
DRBD kernel module status:              Partially loaded

Queue manager name:                     QM2
Queue manager status:                   Running elsewhere
HA status:                              Unknown
HA current location:                    mqhavm58.exampleco.com
HA preferred location:                  This node
HA blocked location:                    All nodes
```

関連資料

Linux rdqmstatus

高可用性構成での *IP* アドレスの変更

高可用性構成内のいずれかのインターフェースの IP アドレスを変更すると、高可用性操作は使用できなくなり、キュー・マネージャーは、アドレスが変更されたノード上で実行されなくなります。

`rdqm.ini` ファイル内の HA 操作には、最大 3 つの IP アドレスを指定します。 Pacemaker モニター・アドレスを既に変更している場合は、手順に従う前に、それらのアドレスを元の値に一時的に戻す必要があります。 値を戻さないと、HA RDQM キュー・マネージャーを削除できません。

1. 各ノードで HA 構成を削除します。 HA を削除するには、キューマネージャをバックアップしてから削除し、671 ページの『IBM MQ キュー・マネージャー・データのバックアップと復元』および 575 ページの『HA RDQM の削除』を参照します。573 ページの『Pacemaker クラスター (HA グループ) の削除』を参照して、HA グループ自体を削除する必要があります。
2. 新しい IP アドレスを使用して HA 構成を再作成します (570 ページの『Pacemaker クラスター (HA グループ) の定義』 を参照してください)。

3. HA キュー・マネージャーを再作成し、バックアップを復元します (573 ページの『HA RDQM の作成』および 671 ページの『IBM MQ キュー・マネージャー・データのバックアップと復元』を参照してください)。

Linux 高可用性の構成の障害ノードの交換

HA グループ内のノードの 1 つに障害が発生した場合は、それを置き換えることができます。

このタスクについて

ノードを置き換えるためのステップは、シナリオによって異なります。

- 障害が発生したノードを同じ構成のノードで置き換える場合は、HA グループを中断せずにノードを置き換えることができます。
- 新しいノードの構成が異なる場合は、HA グループを削除して再構築する必要があります。 ノードで実行しているキュー・マネージャーのバックアップをまず作成してから、HA グループの再構築後にリストアします。

手順

- 障害が発生したノードのように交換ノードが構成されている場合 (同じホスト名、同じ IP アドレスなど) は、新規ノードで以下のステップを実行します。
  a) その他のノードのファイルと一致する `rdqm.ini` ファイルを作成し、`rdqmadm -c` コマンドを実行します (570 ページの『Pacemaker クラスター (HA グループ) の定義』を参照)。
  b) `crtmqm -sxs` *`qmanager`* コマンドを実行して、各複製データ・キュー・マネージャーを再作成します (573 ページの『HA RDQM の作成』を参照)。
- 交換ノードの構成が、障害が発生したノードとは異なる場合:
  a) 必要に応じて、キュー・マネージャーのバックアップを作成します (671 ページの『IBM MQ キュー・マネージャー・データのバックアップと復元』を参照)。
  b) **`dltmqm`** コマンドを使用して、HA グループ内の他のノードから複製データ・キュー・マネージャーを削除します (575 ページの『HA RDQM の削除』を参照)。
  c) **`rdqmadm -u`** コマンドを使用して、 Pacemaker クラスターを構成解除します (573 ページの『Pacemaker クラスター (HA グループ) の削除』を参照)。
  d) **`rdqmadm -c`** コマンドを使用して、新規ノードの情報を含む Pacemaker クラスターを再構成します (570 ページの『Pacemaker クラスター (HA グループ) の定義』を参照)。
  e) 必要に応じて (つまり、他のノードへの SSH アクセス権限がない場合)、`crtmqm -sxs` *`qmanager`* コマンドを実行して、他のノード上に各複製データ・キュー・マネージャーを再作成します (573 ページの『HA RDQM の作成』を参照)。
  f) `crtmqm -sx` *`qmanager`* コマンドを実行して、交換ノードでキュー・マネージャーを作成します。
  g) 必要に応じて、キュー・マネージャーにデータをリストアして構成します (671 ページの『IBM MQ キュー・マネージャー・データのバックアップと復元』を参照)。

Linux MQ Adv. RDQM 災害復旧

Linux プラットフォームのサブセットでは、RDQM (複製データ・キュー・マネージャー) を利用して、災害復旧ソリューションを構築できます。

詳細については、Software Product Compatibility Reports を参照してください。

1 つのサーバーで災害復旧キュー・マネージャーの 1 次インスタンスを作成し、リカバリー・ノードとして実行する別のサーバーでキュー・マネージャーの 2 次インスタンスを作成できます。 キュー・マネージャーのインスタンス間でデータを複製します。 1 次キュー・マネージャーを失った場合は、手動で 2 次インスタンスを 1 次インスタンスにしてキュー・マネージャーを開始し、同じ場所から処理を再開できます。 2 次の役割になっているキュー・マネージャーを開始することはできません。 2 つのノード間でのデータの複製は DRBD によって処理されます。

1 次キュー・マネージャーと 2 次キュー・マネージャーの間のデータ複製では、同期複製と非同期複製のいずれかを選択できます。 非同期オプションを選択すると、IBM MQ の PUT や GET などの操作が完了してアプリケーションに戻ってから、そのイベントが 2 次キュー・マネージャーに複製されます。 非同期で複製されるので、リカバリー状態となった後に、一部のメッセージング・データが失われる可能性があります。 しかし、2 次キュー・マネージャーは一貫性のある状態を保ち、メッセージ・ストリームの少し前の部分で開始する場合でも、即時に実行を開始することができます。

既存のキュー・マネージャーが RDQM キュー・マネージャーになるようにマイグレーションする (601 ページの『キュー・マネージャーをマイグレーションして DR RDQM キュー・マネージャーにする』を参照) ことはできますが、既存のキュー・マネージャーに災害復旧構成を追加することはできません。

さまざまなサーバーで実行する RDQM キュー・マネージャーについては、いろいろな組み合わせが可能です。 例えば、1 次災害復旧キュー・マネージャーを別々のノードで実行し、すべての 2 次災害復旧キュー・マネージャーを同じノードで実行することもできます。 いくつかの構成例を以下の図に示します。

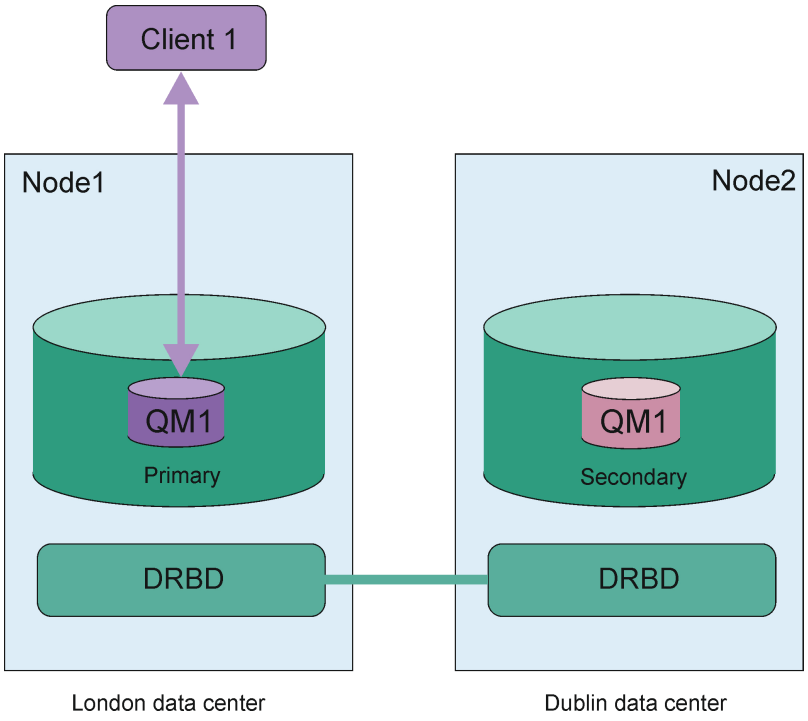


図 *81. 1 つの RDQM ペア*

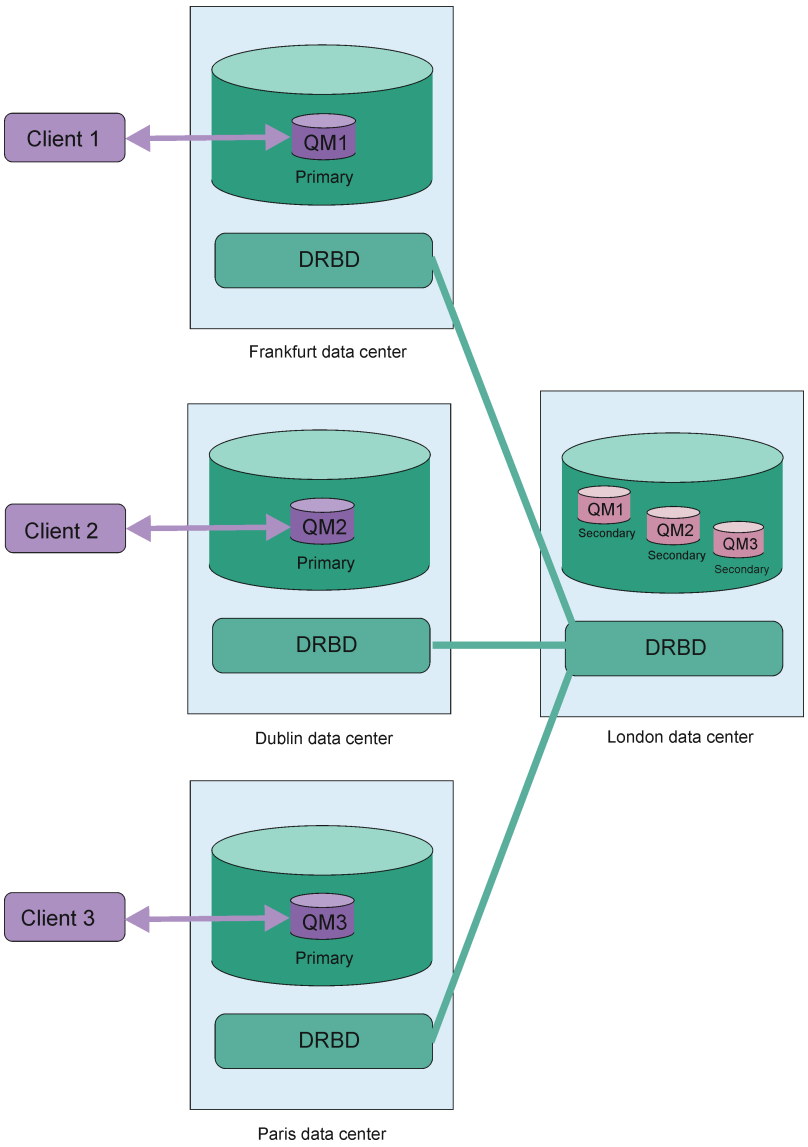


図 *82. 2* 次キュー・マネージャーを同じノードで構成する例

複製、同期、スナップショット

災害復旧構成にした 2 つのノードを接続すると、災害復旧キュー・マネージャーの永続データに対するすべての更新がキュー・マネージャーの 1 次インスタンスから 2 次インスタンスに転送されます。 これを**複製**といいます。

ノード間のネットワーク接続が失われると、キュー・マネージャーの 1 次インスタンスの永続データに対する変更が追跡されます。 ネットワーク接続が復旧すると、2 次インスタンスの内容の取り込みができるだけ早く追い付くようにするために、別のプロセスが使用されます。 これを**同期**といいます。

同期が進行中のとき、2 次インスタンス内のデータは不整合の状態になっています。 2 次キュー・マネージャー・データの状態の**スナップショット**が取得されます。 同期中にメイン・ノードまたはネットワーク接続で障害が発生すると、2 次インスタンスがそのスナップショットに戻って、キュー・マネージャーを開始できるようになります。 ただし、元のネットワーク障害の発生後に行われたすべての更新情報は失われます。

区分データ (スプリット・ブレーン)

DR RDQM 構成では、キュー・マネージャーの 1 次インスタンスの消失後に、リカバリー・ノードで 2 次インスタンスをレベル上げして実行するためのユーザー・アクションが必要になります。 2 次インスタンスをレベル上げするのがだれでも、それまでの 1 次キュー・マネージャーが停止していることを必ず確認し

なければなりません。 従来の 1 次が実行されている場合にはメッセージを処理中の可能性があり、通常操作を復元すると、キュー・マネージャーの 2 つのインスタンスに異なるデータ・ビューが含まれることがあります。 これは区分化またはスプリット・ブレーンと呼ばれる状態です。

以下の状況について考慮してみましょう。

- 1 次キュー・マネージャーが稼働しているノードが完全に故障しています。 2 次インスタンスが 1 次になるようレベル上げしますが、従来の 1 次が実行されていないため、停止するためのアクションを実行できません。 従来のノードが修復または置き換えられると、対象ノード上のキュー・マネージャーが最初に 2 次となり、リカバリー・ノード上の 1 次キュー・マネージャーと同期されます。 次に、2 次のキュー・マネージャーの役割が逆になり、通常稼働状態が再開されます。 この状況で可能性があるデータ損失は、対象ノードで障害が発生する前に 1 次が 2 次に複製を完了していなかったデータのみとなります。
- キュー・マネージャーの 1 次と 2 次のインスタンスを実行しているノード間における複製リンクに影響を及ぼすネットワーク障害が発生します。 この状況では、2 次をレベル上げする前に従来の 1 次を停止する必要があります。 従来の 1 次に他のネットワーク接続が依然として存在する場合、実質的に 2 つの 1 次インスタンスが同時実行され、区分データが生じる可能性があります (複製リンクが作動している場合、1 次インスタンスが実行中であると、2 次キュー・マネージャーをレベル上げできず、コマンドが失敗します)。
- キュー・マネージャーの 1 次インスタンスを実行しているノードでネットワーク障害が発生します。 この場合も、2 次をレベル上げする前に、1 次インスタンスが停止していることを確認しなければなりません。 ネットワークが復元されるときに従来の 1 次が実行中であると、1 次インスタンスが 2 つ存在することになり、区分データがやはり生じます。

管理されたフェイルオーバーを実行する場合、キュー・マネージャー・インスタンスで DR 状況が `partitioned` になることはないはずです。 管理されたフェイルオーバーによって 1 次ノード上のキュー・マネージャーが終了し、その後、データがすべて複製された後にリカバリー・ノード上でキュー・マネージャーが開始されます。 キュー・マネージャーが終了し、ノード間でデータが同期されてから、リカバリー・ノードで開始されるため、パーティション状況になることは予期されません。 ノード間で接続が失われている状況でリカバリー・ノードでキュー・マネージャーが開始されると、接続が失われたときにメイン・ノードでキュー・マネージャーがアクティブである場合にはデータの不一致が生じる可能性があります。 このシナリオでは、キュー・マネージャーのデータが同期されていないため、接続が復元されると、パーティション状態が報告されることが予想されます。 パーティション状態が生じると、2 つのデータ・セットを調べ、保持するセットについて情報に基づいて判断する必要が生じます。 616 ページの『DR RDQM でのパーティション (スプリット・ブレーン) 問題の解決』を参照してください。

Linux *RDQM DR ソリューションの要件*

RDQM 災害復旧 (DR) キュー・マネージャー・ペアを構成する前に、いくつかの要件を満たさなければなりません。

システム要件

RDQM DR を構成する前に、RDQM DR キュー・マネージャーをホストする各サーバーでいくつかの構成を完了する必要があります。

- 各ノードには、`drbdpool` という名前のボリューム・グループが必要です。 各災害復旧複製データ・キュー・マネージャー (DR RDQM) のストレージは、このボリューム・グループのキュー・マネージャーごとに 2 つの別個の論理ボリュームとして割り振られます。 (それぞれのキュー・マネージャーは、スナップショットへの復帰操作をサポートするために 2 つの論理ボリュームを必要とするので、各 DR RDQM に、作成時の指定値の 2 倍のストレージが割り振られます。) 最高のパフォーマンスを得るためには、このボリューム・グループを内部ディスク・ドライブ (SSD) に対応する 1 つ以上の物理ボリュームで構成する必要があります。
- `drbdpool` ボリューム・グループを作成した後は、それ以外の操作は何も行いません。 IBM MQ は、`drbdpool` で作成された論理ボリューム、およびそれらがマウントされる方法と場所を管理します。
- 各ノードにデータ複製用のインターフェースが必要です。 すべての複製データ・キュー・マネージャーの予期されるワークロードを前提として、複製要件をサポートするために十分な帯域幅が必要です。

  フォールト・トレランスを最大にするために、このインターフェースとして、独立したネットワーク・インターフェース・カード (NIC) を使用してください。

- RFC 1123 で修正された RFC 952 で定義されているように、DRBD では、有効なインターネット・ホスト名 (uname -n によって返される値) が RDQM 用の各ノードで必要になります。
- DR RDQM 用のノード同士の間にファイアウォールがある場合は、そのファイアウォールで、ノード間のトラフィックを複製用のポートで許可する必要があります。 RHEL で標準のファイアウォールを実行している場合は、必要なポートを開くサンプル・スクリプト(/opt/mqm/samp/rdqm/firewalld/configure.sh)が提供されています。 このスクリプトは root として実行する必要があります。 他のファイアウォールを使用している場合は、サービス定義 /usr/lib/firewalld/services/rdqm* を調べて、開く必要があるポートを確認します。 このスクリプトは、DRBD および IBM MQ 用の以下の永続 firewallD サービス・ルールを追加します (HA を使用していない場合は、スクリプトを編集して Pacemaker ポートを省略できます)。
  - MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-drbd.xml は TCP ポート 7000-7100 を使用できるようにします。
  - MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-mq.xml は TCP ポート 1414 を使用できるようにします (別のポートが必要な場合はスクリプトを編集する必要があります)。
- SELinux を permissive 以外のモードで使用しているシステムでは、以下のコマンドを実行する必要があります。

```
semanage permissive -a drbd_t
```

ネットワーク要件

災害復旧用のノードをさまざまなデータ・センターに配置することをお勧めします。

次の制約に注意してください。

- データ・センター間の待ち時間が長くなると、パフォーマンスが急速に低下します。 IBM は、最大 5 ミリ秒 (同期複製の場合) および 100 ミリ秒 (非同期複製の場合) の待ち時間をサポートします。
- 複製リンクで送信するデータは、IBM MQ AMS での暗号化は別として、それ以外の暗号化の対象になりません。
- RDQM キュー・マネージャーを災害復旧用に構成すると、2 つの RDQM ノード間のデータ複製の要件によるオーバーヘッドが発生します。 同期複製の方が非同期複製よりもオーバーヘッドが大きくなります。 同期複製では、データが両方のノードに書き込まれるまで、ディスク入出力操作がブロックされます。 非同期複製では、データを 1 次ノードに書き込むだけで処理の継続が可能になります。

キュー・マネージャーで作業するためのユーザー要件

複製データ・キュー・マネージャー (RDQM) の作成/削除/構成の作業は、root ユーザーとして実行するか、以下のコマンドに対する sudo 権限を持った mqm グループに属するユーザー ID で実行する必要があります。

- **crtmqm**
- **dltmqm**
- **rdqmdr**

mqm グループに属するユーザーは、以下のコマンドを使用して DR RDQM の状態と状況を表示できます。

- **dspmq**
- **rdqmstatus**

mqm ユーザーは両方のサーバー上で同じ UID を持っている必要があり、mqm グループは両方のサーバーで同じ GID を持つ必要があります。

Linux 災害復旧 *RDQM* の作成

crtmqm コマンドを使用して、災害復旧構成の 1 次キュー・マネージャーまたは 2 次キュー・マネージャーとして機能する複製データ・キュー・マネージャー (RDQM) を作成します。

このタスクについて

ユーザーが sudo を使用できる場合は、mqm グループのユーザーとして複製データ・キュー・マネージャー (RDQM) を作成できます。 そうでない場合は、RDQM を root として作成する必要があります。

1 つのノードで 1 次 RDQM DR キュー・マネージャーを作成する必要があります。 次に、別のノードで同じキュー・マネージャーの 2 次インスタンスを作成する必要があります。 1 次インスタンスと 2 次インスタンスの名前は同じでなければならず、同じ量のストレージを割り振る必要があります。

以下に、キュー・マネージャーのファイル・システムのサイズ変更に関するガイダンスを示します。

1. RDQM キュー・マネージャーの作成時に、キュー・マネージャーのデータとログを保管するためにファイル・システムが割り振られます。 キュー・マネージャーが進行中のアクティビティーをログに記録し、アプリケーション・メッセージをキューに保管できるように、このファイル・システムを適切にサイズ変更することが重要です。 ファイル・システムをサイジングする際には、ピーク時のメッセージング要件、将来のワークロードの増加、およびメッセージがキューに蓄積される原因となる可能性のあるアプリケーション障害を考慮してください。 キュー・マネージャーの回復ログのサイズを計算するためのガイダンスについては、651 ページの『ログ・ファイル・システムの大きさの決定方法』を参照してください。 アプリケーション・メッセージのストレージ要件を計算する際には、メッセージのサイズと数に加えて、MQMD ヘッダーおよびメッセージ・プロパティーを考慮する必要があります。
2. RDQM キュー・マネージャーのファイル・システムを動的にサイズ変更することはできません。 これが必要な場合は、より大きなファイル・システムを使用して RDQM キュー・マネージャーをバックアップしてからリストアする必要があります。 579 ページの『HA RDQM キュー・マネージャーのファイル・システムのサイズ変更』を参照してください。
3. MAXDEPTH や MAXFSIZE などのローカル・キュー属性を使用して、ディスク上の個々のキューのサイズを制限することができます。 IBM MQ キュー・ファイルの変更を参照してください。
4. ファイル・システムの使用量が重要になる前に、進行中のディスク使用量をモニターし、ディスク使用量が増加した場合は適切に対応する必要があります。 ファイル・システムの使用状況をモニターするには、プラットフォーム/オペレーティング・システムの機能を使用するか、システム・トピックでパブリッシュされるメトリックで説明されている IBM MQ システム・トピックにパブリッシュされるメトリックをサブスクライブします。

手順

- 1 次 DR RDQM を作成するには、以下のようにします。

  a) 以下のコマンドを入力します。

```
crtmqm -rr p [-rt (a | s)] -rl Local_IP -ri Recovery_IP -rn Recovery_Name -rp Port
[other_crtmqm_options] [-fs size] QMname
```

  ここで、

  -rr p
  キュー・マネージャーの 1 次インスタンスを作成することを指定します。

  -rt a | s
  -rt s では、DR 構成で同期複製を使用することを指定し、**-rt a** では、DR 構成で非同期複製を使用することを指定します。 非同期複製がデフォルトです。

  -rl *Local_IP*
  このキュー・マネージャーの DR 複製で使用するローカル IP アドレスを指定します。

  -ri *Recovery_IP*
  キュー・マネージャーの 2 次インスタンスをホスティングするサーバーでの複製に使用するインターフェースの IP アドレスを指定します。

  -rn *Recovery_Name*
  キュー・マネージャーの 2 次インスタンスをホスティングするシステムの名前を指定します。 この名前は、そのサーバーで uname -n を実行した時に返される値です。 そのサーバーで 2 次キュー・マネージャーを明示的に作成する必要があります。

-rp *Port*
DR レプリケーションに使用するポートを指定します。

other_crtmqm_options
(オプション) 以下の一般的な **crtmqm** オプションを 1 つ以上指定することもできます。

- -z
- -q
- -c *Text*
- -d *DefaultTransmissionQueue*
- -h *MaxHandles*
- -g *ApplicationGroup*
- -oa *user|group*
- -t *TrigInt*
- -u *DeadQ*
- -x *MaxUMsgs*
- -lp *LogPri*
- -ls *LogSec*
- -lc | -l
- -lla | -lln
- -lf *LogFileSize*
- -p *Port*

-fs *size*
(オプション) キュー・マネージャー用に作成するファイル・システムのサイズ、つまり drbdpool ボリューム・グループ内で作成する論理ボリュームのサイズを指定します。 スナップショットへの復帰操作をサポートするために、そのサイズの別の論理ボリュームも作成するので、DR RDQM の合計ストレージは、ここで指定した値の 2 倍になります。

Size は、GB 単位で指定される数値です。 値を MB 単位で指定するには、値の後に文字 M を入力します。 例えば、3 GB のファイル・システム・サイズを指定するには、3 を入力します。 1024 MB のファイル・システム・サイズを指定するには、 1024M と入力します。 (G 接尾部を明示的に状態 GB に追加することもできます。)

QMname
複製データ・キュー・マネージャーの名前を指定します。 この名前には大/小文字の区別があります。

コマンドが完了すると、キュー・マネージャーの 2 次インスタンスを作成するために 2 次ノードで入力する必要のあるコマンドが出力されます。 1 次ノードで **rdqmdr** コマンドを使用して、2 次キュー・マネージャーを作成するために 2 次ノードで実行する必要がある **crtmqm** コマンドを取得することもできます。 607 ページの『DR RDQM の 1 次特性と 2 次特性の管理』を参照してください。

- 2 次 DR RDQM を作成するには、以下のようにします。

  a) RDQM の 2 次インスタンスをホストするノードで、以下のコマンドを入力します。

```
crtmqm  -rr s [-rt (a | s)] -rl Local_IP -ri Primary_IP -rn Primary_Name  -rp Port
[other_crtmqm_options] [-fs size] QMname
```

  説明

  -rr s
  キュー・マネージャーの 2 次インスタンスを作成することを指定します。

  -rt a | s
  -rt s は、DR 構成が同期複製を使用することを指定し、 **-rt a** は DR 構成が非同期複製を使用することを指定します。

-rl *Local_IP*
このキュー・マネージャーの DR 複製で使用するローカル IP アドレスを指定します。

-ri *Primary_IP*
キュー・マネージャーの 1 次インスタンスをホスティングするサーバーでの複製に使用するインターフェースの IP アドレスを指定します。

-rn *Primary_Name*
キュー・マネージャーの 1 次インスタンスをホスティングするシステムの名前を指定します。 この名前は、そのサーバーで uname -n を実行した時に返される値です。

-rp *Port*
DR レプリケーションに使用するポートを指定します。

other_crtmqm_options
(オプション) 以下の一般的な **crtmqm** オプションを 1 つ以上指定することもできます。

– -z

-fs *size*
キュー・マネージャー用に作成するファイル・システムのサイズ、つまり drbdpool ボリューム・グループ内で作成する論理ボリュームのサイズを指定します。 1 次キュー・マネージャーの作成時にデフォルト以外のサイズを指定した場合は、ここで同じ値を指定する必要があります。

Size は、GB 単位で指定される数値です。 値を MB 単位で指定するには、値の後に文字 M を入力します。 例えば、3 GB のファイル・システム・サイズを指定するには、3 を入力します。 1024 MB のファイル・システム・サイズを指定するには、 1024M と入力します。 (G 接尾部を明示的に状態 GB に追加することもできます。)

QMname
複製データ・キュー・マネージャーの名前を指定します。 キュー・マネージャーの 1 次インスタンスで指定した名前と同じ名前でなければなりません。 名前には大/小文字の区別があります。

次のタスク

キュー・マネージャーの 1 次インスタンスと 2 次インスタンスを作成したら、両方のノードの状況を調べて、両方が正しいことを確認する必要があります。 両方のノードで **rdqmstatus** コマンドを使用します。 どちらのノードでも正常状況が表示されるはずです (608 ページの『DR RDQM 状況の表示』を参照)。 その状況が表示されない場合は、2 次インスタンスをいったん削除して再作成してください。その時に、正しい引数を使用するように注意してください。

関連資料

crtmqm

Linux *DR RDQM の削除*

dltmqm コマンドを使用して、災害復旧複製データ・キュー・マネージャー (RDQM) を削除します。

このタスクについて

RDQM を削除するために、RDQM の 1 次ノードと 2 次ノードの両方でこのコマンドを実行する必要があります。 最初に RDQM を終了する必要があります。 mqm ユーザーが必要な sudo 特権を持っている場合は、mqm ユーザーとしてコマンドを実行できます。 その他の場合は、root としてコマンドを実行する必要があります。

手順

- DR RDQM を削除するには、以下のコマンドを入力します。

```
dltmqm RDQM_name
```

関連資料

dltmqm

Linux MQ Adv. キュー・マネージャーをマイグレーションして *DR RDQM* キュー・マネージャーにする

既存のキュー・マネージャーをマイグレーションして災害復旧 (DR) 複製データ・キュー・マネージャー (RDQM) にするには、その永続データをバックアップしてから、新規に作成された同じ名前の RDQM キュー・マネージャーにデータをリストアします。

このタスクについて

DR 複製データ・キュー・マネージャーには、専用の論理ボリューム (ファイル・システム) と、ディスク複製の構成が必要です。 これらのコンポーネントは、新規キュー・マネージャーが作成されたときにのみ構成されます。 既存のキュー・マネージャーをマイグレーションして RDQM を使用するには、その永続データをバックアップしてから、新規に作成された同じ名前の RDQM キュー・マネージャーにデータをリストアします。 この手順では、バックアップの作成時にキュー・マネージャーの構成、状態、および持続メッセージを保存します。

注 : キュー・マネージャーは、RDQM がインストールされているバージョンと同じかそれより前のバージョンの IBM MQ からのみマイグレーションできます。 オペレーティング・システムとアーキテクチャーも同じでなければなりません。 それ以外の場合は、ターゲット・プラットフォーム上に新しいキュー・マネージャーを作成する必要があります。別のオペレーション・システムへのキュー・マネージャーの移動を参照してください。

キュー・マネージャーをマイグレーションする前に、以下の条件を満たす必要があります。

- 災害復旧の要件を評価し、593 ページの『RDQM 災害復旧』を参照してください。
- キュー・マネージャーに接続するアプリケーションおよびキュー・マネージャーを確認します。 キュー・マネージャーが実行されている RDQM ノードに接続をルーティングするために必要な変更を検討します。
- 選択した構成の RDQM ノードをプロビジョンするか、既存の RDQM ノードを識別します。 RDQM のシステム要件について詳しくは、596 ページの『RDQM DR ソリューションの要件』を参照してください。
- 各ノードに、RDQM フィーチャーを含む IBM MQ Advanced をインストールします。
- オプションで、テスト・キュー・マネージャーを使用して RDQM 構成を検証します。このキュー・マネージャーはあとで削除できます。 キュー・マネージャーをマイグレーションする前に問題を特定して解決するために、構成をテストすることをお勧めします。
- キュー・マネージャーのセキュリティー構成を確認し、必要なローカル・ユーザーおよびグループを各 RDQM ノードに複製します。
- キュー・マネージャーおよびチャネル構成を調べて、API 出口、チャネル出口、またはデータ変換出口が使用されているかどうかを判別します。 必要な出口を各 RDQM ノードにインストールします。
- 定義されているキュー・マネージャー・サービスを確認し、必要なプロセスを各 RDQM ノードでインストールして構成します。

手順

1. 既存のキュー・マネージャーをバックアップします。
   a) 待機シャットダウン・コマンド `endmqm -w` または即時シャットダウン・コマンド `endmqm -i` を発行して、既存のキュー・マネージャーを停止します。 このステップは、バックアップ内のデータの整合性を確保するために重要です。
   b) IBM MQ 構成ファイル `mqs.ini` を表示して、キュー・マネージャーのデータ・ディレクトリーの位置を判別します。 Linux では、ファイルは `/var/mqm` ディレクトリー内にあります。 `mqs.ini` について詳しくは、86 ページの『IBM MQ 構成ファイル mqs.ini』を参照してください。

      ファイル内でキュー・マネージャーの `QueueManager` スタンザを見つけます。 スタンザに `DataPath` という名前のキーが含まれている場合、その値はキュー・マネージャーのデータ・ディレクトリーです。 キーが存在しない場合、キュー・マネージャーのデータ・ディレクトリーは、`Prefix` キーと `Directory` キーの値を使用して判別できます。 キュー・マネージャーのデータ・ディレクトリーは、これらの値を *prefix*/qmgrs/*directory* の形式で連結したものです。 QueueManager スタン

ザについて詳しくは、96 ページの『mqs.ini ファイルの QueueManager スタンザ』を参照してください。

c) キュー・マネージャーのデータ・ディレクトリーのバックアップを作成します。 Linux の場合、これを行うには **tar** コマンドを使用します。 例えば、キュー・マネージャーのデータ・ディレクトリーをバックアップするには、以下のコマンドを使用できます。 コマンドの最後のパラメーターである単一ピリオド (ドット) に注意してください。

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

d) IBM MQ キュー・マネージャー構成ファイル `qm.ini` を表示して、キュー・マネージャーのログ・ディレクトリーの位置を判別します。 このファイルは、キュー・マネージャーのデータ・ディレクトリーにあります。 このファイルについて詳しくは、98 ページの『キュー・マネージャー構成ファイル qm.ini』を参照してください。

キュー・マネージャーのログ・ディレクトリーは、Log スタンザの LogPath キーの値として定義されます。 スタンザについて詳しくは、132 ページの『qm.ini ファイルの Log スタンザ』を参照してください。

e) キュー・マネージャーのログ・ディレクトリーのバックアップを作成します。 Linux の場合、これを行うには tar コマンドを使用します。 例えば、キュー・マネージャーのログ・ディレクトリーをバックアップするには、以下のコマンドを使用できます。 コマンドの最後のパラメーターである単一ピリオド (ドット) に注意してください。

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

f) キュー・マネージャーが使用する証明書リポジトリーがキュー・マネージャーのデータ・ディレクトリーにない場合は、そのバックアップを作成します。 鍵データベース・ファイルとパスワード・スタッシュ・ファイルの両方がバックアップされていることを確認してください。 キュー・マネージャーの鍵リポジトリーについては、SSL/TLS 鍵リポジトリーおよびキュー・マネージャーの鍵リポジトリーの検索を参照してください。 AMS メッセージ・チャネル・エージェント (MCA) インターセプトを使用するようにキュー・マネージャーが構成されている場合の AMS 鍵ストアの検索については、メッセージ・チャネル・エージェント (MCA) インターセプトを参照してください。

g) 既存のキュー・マネージャーは不要になったため、削除できます。 ただし、可能であれば、既存のキュー・マネージャーは、ターゲット・システムで正常にリストアされた後にのみ削除してください。 削除を保留しておくと、マイグレーション・プロセスが正常に完了しない場合にキュー・マネージャーを再始動できます。

注 : 既存のキュー・マネージャーの削除を保留する場合は、再始動しないでください。 構成または状態に追加で加えられる変更がマイグレーション中に失われるため、キュー・マネージャーは終了したままであることが重要です。

2. 1 次 RDQM ノードを準備します。

a) バックアップしたキュー・マネージャーと同じ名前の新規 RDQM キュー・マネージャーを作成します。 **crtmqm** によって RDQM キュー・マネージャーに割り振られたファイル・システムが、既存のキュー・マネージャーのデータ、1 次ログ、2 次ログ、および将来の拡張用の追加スペースを含めるのに十分な大きさであることを確認します。 RDQM キュー・マネージャーを作成する方法について詳しくは、597 ページの『災害復旧 RDQM の作成』を参照してください。

b) キュー・マネージャーの 1 次 RDQM ノードを判別します。 1 次ノードの判別方法については、rdqmstatus (RDQM 状況の表示) を参照してください。

c) 1 次 RDQM ノードで RDQM キュー・マネージャーが開始されている場合は、`endmqm -w` コマンドまたは `endmqm -i` コマンドを使用して停止します。

d) RDQM キュー・マネージャーのデータ・ディレクトリーとログ・ディレクトリーの場所を判別します (ステップ 1b および 1d で説明されている方法を使用します)。

e) RDQM キュー・マネージャーのデータ・ディレクトリーとログ・ディレクトリーの内容を削除します。ただし、ディレクトリー自体は削除しないでください。

3. 1 次 RDQM ノードでキュー・マネージャーをリストアします。

a) キュー・マネージャーのデータ・ディレクトリーとログ・ディレクトリーのバックアップを 1 次 RDQM ノードにコピーし、キュー・マネージャーが使用する証明書リポジトリーの別個のバックアップをコピーします。

b) キュー・マネージャーのデータ・ディレクトリーのバックアップを新しい RDQM キュー・マネージャーの空のデータ・ディレクトリーにリストアし、ファイルの所有権と許可が保持されるようにします。 ステップ 1c のサンプル tar コマンドを使用してバックアップを作成した場合は、root ユーザーが以下のコマンドを使用してバックアップをリストアできます。

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

c) キュー・マネージャーのログ・ディレクトリーのバックアップを新しい RDQM キュー・マネージャーの空のログ・ディレクトリーにリストアし、ファイルの所有権と許可が保持されるようにします。 ステップ 1e のサンプル tar コマンドを使用してバックアップを作成した場合は、root ユーザーが以下のコマンドを使用してバックアップをリストアできます。

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

d) RDQM キュー・マネージャーのデータ・ディレクトリーに、リストアされたキュー・マネージャー構成ファイル(qm.ini)を編集します。 Log スタンザ内の LogPath キーの値を更新して、RDQM キュー・マネージャーのログ・ディレクトリーを指定します。

構成ファイルに定義されている他のファイル・パスを確認し、必要に応じて更新します。 例えば、以下のパスを更新しなければならない可能性があります。

- 診断メッセージ・サービスによって生成されるエラー・ログ・ファイルのパス。
- キュー・マネージャーで必要な出口のパス。
- キュー・マネージャーが XA トランザクション・コーディネーターである場合のスイッチ・ロード・ファイルのパス。

e) AMS メッセージ・チャネル・エージェント (MCA) インターセプトを使用するようにキュー・マネージャーが構成されている場合は、AMS 鍵ストアを新しい RDQM インストール済み環境にコピーしてから、構成を確認して更新します。 鍵ストアは各 RDQM ノードで使用可能でなければならないため、キュー・マネージャーの複製ファイル・システムにない場合は、代わりに各ノードにコピーする必要があります。 詳細については、メッセージ・チャネル・エージェント (MCA) インターセプトを参照してください。

f) **dspmq** コマンドによってキュー・マネージャーが表示され、その状況が「終了」と報告されていることを確認します。 以下の例は、RDQM DR キュー・マネージャーの出力例を示しています。

```
$ dspmq -o status -o dr

QMNAME(QM1) STATUS(Ended normally) DRROLE(Primary)
```

g) **rdqmstatus** コマンドを使用してキュー・マネージャーの状況を表示することにより、リストアされたキュー・マネージャー・データが 2 次 RDQM ノードに複製されたことを確認します。 DR 状況は、各ノードで Normal として報告される必要があります。 以下の例は、RDQM DR キュー・マネージャーの出力例を示しています。

```
$ rdqmstatus -m QM1
Queue manager status:                   Ended normally
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Normal
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
```

h) 1 次 RDQM ノードでキュー・マネージャーを開始します。

i) キュー・マネージャーに接続し、SSLKEYR キュー・マネージャー属性の値を更新して、キュー・マネージャー証明書リポジトリーの新しい場所を指定します。 デフォルトで、この属性の値には *queue_manager_data_directory*/ssl/key が設定されます。 証明書リポジトリーは、各

RDQM ノード上の同じ場所に配置する必要があります。 リポジトリーがキュー・マネージャーの複製ファイル・システムにない場合は、代わりに各ノードにコピーする必要があります。

j) キュー・マネージャーの IBM MQ オブジェクト定義を確認し、変更されたネットワーク設定、IBM MQ インストール・ディレクトリー、またはキュー・マネージャー・データ・ディレクトリーを参照するオブジェクト属性の値を更新します (以下のオブジェクトを含む)。

- リスナーが使用するローカル IP アドレス (IPADDR 属性)。
- チャネルが使用するローカル IP アドレス (LOCLADDR 属性)。
- クラスター受信側チャネル用に定義されたローカル IP アドレス (CONNAME 属性)。
- コミュニケーション情報オブジェクト用に定義されたローカル IP アドレス (GRPADDR 属性)。
- プロセスおよびサービス・オブジェクト定義用に定義されたシステム・パス。

k) キュー・マネージャーをいったん停止したあとで再始動して、変更内容が有効になるようにします。

l) マイグレーションされたキュー・マネージャーに接続するリモート・キュー・マネージャーと、それに相当するアプリケーションの設定 (以下を含む) について、ステップ 3j を繰り返します。

- チャネル接続名 (CONNAME 属性)。
- IP アドレスまたはホスト名に基づいてキュー・マネージャーからのインバウンド接続を制限するチャネル認証規則。
- クライアント・チャネル定義テーブル (CCDT)、ドメイン・ネーム設定 (DNS)、ネットワーク・ルーティング、または同等の接続情報。

m) 各 RDQM ノードへのキュー・マネージャーの管理フェイルオーバーを実行して、必要な構成が正常に確立されたことを確認します。613 ページの『リカバリー・ノードへの切り替え』を参照してください。

DR RDQM キュー・マネージャーのファイル・システムのサイズ変更
既存の災害復旧 (DR) 複製データ・キュー・マネージャー (RDQM) のファイル・システムをサイズ変更するには、その永続データをバックアップしてから、同じ名前で、サイズが異なるファイル・システムを持つ新しく作成された RDQM キュー・マネージャーにデータをリストアします。

このタスクについて

DR 複製データ・キュー・マネージャーには、専用の論理ボリューム (ファイル・システム) と、ディスク複製の構成が必要です。 これらのコンポーネントは、新規キュー・マネージャーが作成されたときにのみ構成されます。 ファイル・システムは、各ノードで同じサイズでなければならないため、作成後にサイズ変更することはできません。 既存の複製データ・キュー・マネージャー (RDQM) のファイル・システムをサイズ変更するには、その永続データをバックアップしてから、同じ名前で、サイズが異なるファイル・システムを持つ新しく作成された RDQM キュー・マネージャーにデータをリストアします。 この手順では、バックアップの作成時にキュー・マネージャーの構成、状態、および持続メッセージを保存します。

手順

1. 1 次 RDQM ノード上の既存の RDQM キュー・マネージャーをバックアップします。
   a) キュー・マネージャーの 1 次 RDQM ノードを判別します。 1 次ノードの判別方法については、rdqmstatus (RDQM 状況の表示) を参照してください。
   b) 1 次 RDQM ノードで RDQM キュー・マネージャーが開始されている場合は、**endmqm -w** コマンドまたは **endmqm -i** コマンドを使用して停止します。
   c) IBM MQ 構成ファイル mqs.ini を表示して、キュー・マネージャーのデータ・ディレクトリーの位置を判別します。 Linux では、ファイルは/var/mqm ディレクトリー内にあります。 mqs.ini について詳しくは、86 ページの『IBM MQ 構成ファイル mqs.ini』を参照してください。

      ファイル内でキュー・マネージャーの QueueManager スタンザを見つけます。 キュー・マネージャーのデータ・ディレクトリーは、DataPath というキーの値です。 QueueManager スタンザについて詳しくは、96 ページの『mqs.ini ファイルの QueueManager スタンザ』を参照してください。

d) キュー・マネージャーのデータ・ディレクトリーのバックアップを作成します。 Linux の場合、これを行うには **tar** コマンドを使用します。 例えば、キュー・マネージャーのデータ・ディレクトリーをバックアップするには、以下のコマンドを使用できます。 コマンドの最後のパラメーターである単一ピリオド (.) に注意してください。

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

e) IBM MQ キュー・マネージャー構成ファイル qm.ini を表示して、キュー・マネージャーのログ・ディレクトリーの位置を判別します。 このファイルは、キュー・マネージャーのデータ・ディレクトリーにあります。 このファイルについて詳しくは、98 ページの『キュー・マネージャー構成ファイル qm.ini』を参照してください。

キュー・マネージャーのログ・ディレクトリーは、Log スタンザの LogPath キーの値として定義されます。 スタンザについて詳しくは、132 ページの『qm.ini ファイルの Log スタンザ』を参照してください。

f) キュー・マネージャーのログ・ディレクトリーのバックアップを作成します。 Linux の場合、これを行うには **tar** コマンドを使用します。 例えば、キュー・マネージャーのログ・ディレクトリーをバックアップするには、以下のコマンドを使用できます。 コマンドの最後のパラメーターである単一ピリオド (.) に注意してください。

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

g) 既存の RDQM キュー・マネージャーを削除してください。

2. 必要なサイズのファイル・システムでキュー・マネージャーをリストアします。

a) バックアップしたキュー・マネージャーと同じ名前の新規 RDQM キュー・マネージャーを作成します。 **crtmqm** によって RDQM キュー・マネージャーに割り振られたファイル・システムが、必要なサイズで、既存のキュー・マネージャーのデータ、1 次ログ、2 次ログ、および将来の拡張用の追加スペースを含めるのに十分な大きさであることを確認します。 RDQM キュー・マネージャーを作成する方法について詳しくは、597 ページの『災害復旧 RDQM の作成』を参照してください。

b) キュー・マネージャーの 1 次 RDQM ノードを判別します。 1 次ノードの判別方法については、rdqmstatus (RDQM 状況の表示) を参照してください。

c) 1 次 RDQM ノードで、RDQM キュー・マネージャーが開始されている場合は、 **endmqm -w** または **endmqm -i** コマンドを使用して停止します。

d) 1 次 RDQM ノードで、RDQM キュー・マネージャーのデータ・ディレクトリーとログ・ディレクトリーの新しい場所を判別します (ステップ 1c および 1e で説明されている方法を使用します)。

e) 1 次 RDQM ノードで、RDQM キュー・マネージャーのデータ・ディレクトリーとログ・ディレクトリーの内容を削除します。ただし、ディレクトリー自体は削除しないでください。

f) 1 次 RDQM ノードで、キュー・マネージャーのデータ・ディレクトリーのバックアップを新しい RDQM キュー・マネージャーの空のデータ・ディレクトリーにリストアし、ファイルの所有権と許可が保持されるようにします。 ステップ 1d のサンプル **tar** コマンドを使用してバックアップを作成した場合は、root ユーザーが以下のコマンドを使用してバックアップをリストアできます。

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

g) 1 次 RDQM ノードで、キュー・マネージャーのログ・ディレクトリーのバックアップを新しい RDQM キュー・マネージャーの空のログ・ディレクトリーにリストアし、ファイルの所有権と許可が保持されるようにします。 ステップ 1f のサンプル **tar** コマンドを使用してバックアップを作成した場合は、root ユーザーが以下のコマンドを使用してバックアップをリストアできます。

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

h) 一次 RDQM ノードで、リストアされたキュー・マネージャー構成ファイル qm.ini を、新しい RDQM キュー・マネージャーのデータ・ディレクトリーに編集します。 Log スタンザ内の LogPath キーの値を更新して、ステップ 2d で判別した新しい RDQM キュー・マネージャーのログ・ディレクトリーを指定します。 構成ファイルに定義されている他のファイル・パスを確認し、必要に応じて更新します。 例えば、以下のパスを更新しなければならない可能性があります。

- 診断メッセージ・サービスによって生成されるエラー・ログ・ファイルのパス。
- キュー・マネージャーで必要な出口のパス。
- キュー・マネージャーが XA トランザクション・コーディネーターである場合のスイッチ・ロード・ファイルのパス。

i) **dspmq** コマンドによってキュー・マネージャーが表示され、その状況が ended として報告されていることを確認します。 以下の例は、RDQM DR キュー・マネージャーの出力例を示しています。

```
$ dspmq -o status -o dr
QMNAME(QM1) STATUS(Ended normally) DR(Primary)
```

j) **rdqmstatus** コマンドを使用してキュー・マネージャーの状況を表示することにより、リストアされたキュー・マネージャー・データが 2 次 RDQM ノードに複製されたことを確認します。 DR 状況は、各ノードで Normal として報告される必要があります。 以下の例は、1 次ノード上の RDQM DR キュー・マネージャーの出力例を示しています。

```
$ rdqmstatus -m QM1
Queue manager status:                   Running
CPU:                                    0.00
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Normal
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
```

以下の例は、リカバリー・ノード上の RDQM DR キュー・マネージャーの出力例を示しています。

```
Queue manager status:                   Ended immediately
DR role:                                Secondary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.20.2
DR remote IP address:                   192.168.20.1
```

k) 1 次 RDQM ノードでキュー・マネージャーを開始します。

l) キュー・マネージャーのリカバリー・ノードへの切り替えを実行して、必要な構成が正常に確立されたことを確認します。613 ページの『リカバリー・ノードへの切り替え』を参照してください。

アプリケーションの持続状況の保管
アプリケーションに関連する持続状況情報を、他のキュー・マネージャー・データと一緒に保管できます。

各 IBM MQ キュー・マネージャーには、その永続的な状態の専用ファイル・システムがあります (キュー・データとリカバリー・ログの両方を含む)。 RDQM 構成では、ファイル・システムは Linux システム (ノード) 間で複製される論理ボリュームを基盤としています。 ファイル・システムには、アプリケーションの永続状況情報を保管するために使用できる userdata ディレクトリーが含まれています。 したがって、複製データ・キュー・マネージャーが、RDQM 構成内の別のノードで実行するよう移動されると、キュー・マネージャー・コンテキストだけでなく、アプリケーション・コンテキストも使用できます。 Unix および Linux システムでのディレクトリーの内容を参照してください。

アプリケーション状態を userdata ディレクトリーに保管することを選択した場合は、このロケーションに書き込まれるデータが、キュー・マネージャーに割り振られた使用可能なディスク・スペースを消費する可能性があることを認識しておく必要があります。 キュー・マネージャーがキュー・データ、ログ、およびその他の永続的な状態の情報を書き込むために使用できるだけの十分なディスク・スペースが残っていることを確認する必要があります。

userdata ディレクトリーには mqm ユーザーおよびグループの所有権があり、ユーザーは IBM MQ 管理者グループ(つまり mqm)でなくても、ユーザーがアクセスできるようになっています。 userdata ディレクトリーの許可を変更することはできませんが、必要な所有権および許可を使用して、ディレクトリーにコンテンツを作成することができます。

RDQM キュー・マネージャーのフェイルオーバー中に、キュー・マネージャーが終了し、そのファイル・システムが現行 RDQM ノード上でアンマウントされます。 その後、ファイル・システムがマウントされ、

RDQM 構成内の別のノードでキュー・マネージャーが再始動されます。 プロセスがそのファイルの 1 つに対してオープン・ハンドルを持っている場合は、ファイル・システムをアンマウントできません。 キュー・マネージャーのフェイルオーバーを確実に完了できるようにするため、キュー・マネージャーのファイル・システムをアンマウントできない場合、オープン・ファイル・ハンドルを持つプロセスには SIGTERM シグナルが送信され、その後、オープン・ハンドルが解放されない場合は SIGKILL が続きます。 アプリケーションは、SIGTERM に正しく応答するように設計されている必要があります。 アプリケーションまたはプロセスがキュー・マネージャー・サービスとして構成されている場合は、管理フェイルオーバー時に、ファイル・システムがアンマウントされる前に、キュー・マネージャーのシャットダウン中に終了できます。 アプリケーションまたはプロセスがキュー・マネージャー・サービスとして構成されていない場合、または非管理フェイルオーバーが発生した場合 (クォーラムの喪失など)、シグナルがファイル・システムを解放するために送信される可能性があります。

Linux *DR RDQM の 1 次特性と 2 次特性の管理*

2 次災害復旧複製データ・キュー・マネージャー (DR RDQM) を 1 次 DR RDQM に変更できます。 1 次インスタンスを 2 次インスタンスに変更することもできます。

このタスクについて

rdqmdr コマンドを使用して、RDQM の 2 次インスタンスを 1 次インスタンスに変更できます。 何かの理由で 1 次インスタンスが失われた時に、このアクションを実行しなければならない場合があります。 その後、キュー・マネージャーを開始し、リカバリー・ノードで実行できます。

rdqmdr コマンドを使用して、RDQM の 1 次インスタンスを 2 次インスタンスに変更することもできます。システムを再構成する時などに、このアクションを実行しなければならない場合があります。

1 次キュー・マネージャーで **rdqmdr** を使用して、リカバリー・ノードでそのキュー・マネージャーの 2 次インスタンスを作成するのに必要な正確なコマンドを取得することもできます。

mqm グループのユーザーが sudo を使用できるのであれば、そのユーザーとして **rdqmdr** コマンドを使用できます。 そうでなければ、root としてログインする必要があります。

手順

- DR RDQM の 2 次インスタンスを 1 次インスタンスに変更するには、以下のコマンドを入力します。

  ```
  rdqmdr -m QMname -p
  ```

  キュー・マネージャーの 1 次インスタンスが実行を続けていて、DR 複製リンクがまだ機能している場合、このコマンドは失敗します。
- キュー・マネージャーの 1 次インスタンスを 2 次インスタンスに変更するには、以下のコマンドを入力します。

  ```
  rdqmdr -m QMname -s
  ```
- キュー・マネージャーの 2 次インスタンスを構成するのに必要な **crtmqm** コマンドを表示するには、1 次ノードで以下のコマンドを入力します。

  ```
  rdqmdr -d -m QMname
  ```

  返された **crtmqm** コマンドを 2 次ノードで入力すれば、RD RDQM の 2 次インスタンスを作成できます。

Linux *DR RDQM の開始、停止、および状態の表示*

IBM MQ の標準的な制御コマンドのバリアントを使用して、災害復旧複製データ・キュー・マネージャー (DR RDQM) を開始/停止したり、現在の状態を表示したりできます。

このタスクについて

複製データ・キュー・マネージャー (RDQM) を開始/停止したり、現在の状態を表示したりするコマンドは、mqm グループに属するユーザーとして実行する必要があります。

そのキュー・マネージャー (つまり、キュー・マネージャーが現在実行されているノード) の 1 次ノードでキュー・マネージャーを開始および停止するには、コマンドを実行する必要があります。

手順

- DR RDQM を開始するには、RDQM の 1 次ノードで以下のコマンドを入力します。

  ```
  strmqm qmname
  ```

  ここで、*qmname* は、開始する RDQM の名前です。
- RDQM を停止するには、RDQM の 1 次ノードで以下のコマンドを入力します。

  ```
  endmqm qmname
  ```

  ここで、*qmname* は、停止する RDQM の名前です。
- RDQM の状態を表示するには、以下のコマンドを入力します。

  ```
  dspmq -m QMname
  ```

  出力される状態情報は、コマンドを RDQM の 1 次ノードまたは 2 次ノードのいずれで実行するかによって異なります。 1 次ノードで実行する場合は、**dspmq** によって返された正常状況メッセージの 1 つが表示されます。 このコマンドを 2 次ノードで実行すると、状況 `Ended immediately` が表示されます。 例えば、**dspmq** がノード RDQM7 で実行されている場合、以下の情報が返されることがあります。

  ```
  QMNAME(DRQM8)                           STATUS(Ended immediately)
  QMNAME(DRQM7)                           STATUS(Running)
  ```

  dspmq の引数を使用して、RDQM が災害復旧用として構成されているかどうか、現時点で 1 次インスタンスになっているか 2 次インスタンスになっているかを確認できます。

  ```
  dspmq -m QMname -o (dr | DR)
  ```

  以下のいずれかの応答が表示されます。

  DRROLE()
  キュー・マネージャーが災害復旧用に構成されていないことを示しています。

  DRROLE(Primary)
  キュー・マネージャーが DR 1 次として構成されていることを示しています。

  DRROLE(Secondary)
  キュー・マネージャーが DR 2 次として構成されていることを示しています。

関連資料
dspmq
endmqm
strmqm

Linux *DR RDQM 状況の表示*

1 つのノードにあるすべての災害復旧複製データ・キュー・マネージャー (DR RDQM) の状況や、指定した DR RDQM の詳細情報を表示できます。

このタスクについて

rdqmstatus コマンドで、すべての DR RDQM の状況や個々の RDQM の状況を表示できます。

V 9.3.0 ノードの要約状況には、RDQM が依存している DRBD カーネル・モジュールに関する情報も表示されます。 RDQM をアップグレードする場合は、システム上で実行されている RHEL カーネルのバージョン用に正しいバージョンの DRBD カーネル・モジュールがインストールされていることを確認することが重要です。 この状況には、OS カーネルのバージョン、DRBD モジュールが作成されたカーネル・バージョン、DRBD バージョン、および DRBD カーネル・モジュールのロード状況が表示されます。

rdqmstatus コマンドを実行するには、mqm グループのユーザーである必要があります。 このコマンドは、DR RDQM ペアのどちらのノードでも実行できます。

手順

- 1 つのノードにあるすべての DR RDQM の要約状況を表示するには、そのノードで以下のコマンドを実行します。

```
rdqmstatus
```

そのノードにある DR RDQM の状況が表示されます。以下に例を示します。

```
Node:                                 mqhavm07.exampleco.com
OS kernel version:                    3.10.0-1160.15.2
DRBD OS kernel version:               3.10.0-1160
DRBD version:                         9.1.1
DRBD kernel module status:            Loaded

Queue manager name:                   DRQM8
Queue manager status:                 Ended immediately
DR role:                              Secondary

Queue manager name:                   DRQM7
Queue manager status:                 Running
DR role:                              Primary
```

V 9.3.0 DRBD カーネル・モジュール状況は、以下のいずれかの値です。

ロード済み (Loaded)
DRBD モジュールがロードされたことを示します。

一部ロード (Partially loaded)
DRBD モジュールがロードされたが、不一致が原因で正しく機能しない場合に発生する可能性があります。

ロードされていません (Not loaded)

DRBD モジュールはロードされていません。 これは、RDQM キュー・マネージャーがまだ作成されていない場合に、新規にインストールされた構成で表示される可能性があります。

インストールされていません
DRBD モジュールがインストールされていないか、 IBM MQ が DRBD モジュールの OS カーネル・バージョンを判別できなかったことを示します。

以前にインストールされたバージョンがまだロードされています (Previously installed version still loaded)
この状況は、既存の DRBD モジュールの実行中 (つまり、RDQM キュー・マネージャーの実行中) に新規 DRBD モジュールがインストールされた場合に発生する可能性があります。 新規にインストールされたモジュールが状況で報告されますが、これは実際に実行されているモジュールではありません。

- 1 つの RDQM の状況を表示するには、以下のコマンドを入力します。

```
rdqmstatus -m  qmname
```

以下の表に、返される情報の要約を示します。

表 *34. Status attributes*

状況属性	可能な値	表示される時期
キュー・マネージャー状況	状態 (dspmq で表示される値)	常に表示されます。
CPU	*n.nn*%	現行ノードの RDQM に 1 次の役割がある場合に限り表示されます

表 34. *Status attributes* (続き)

状況属性	可能な値	表示される時期
メモリー	*nnn*MB	現行ノードの RDQM に 1 次の役割がある場合に限り表示されます
キュー・マネージャー・ファイル・システム	*nnn*MB used,*n.n*GB allocated [*n*%]	現行ノードの RDQM に 1 次の役割がある場合に限り表示されます
DR 役割	1 次 2 次 不明	常に表示されます。
DR 状況	通常	通常の操作
	同期が進行中	同期が進行中です
	パーティション	DR 複製ネットワークが使用不可の状態になっている時に、両方のノードでキュー・マネージャーが開始されました。
	リモート・システムが使用不可	もう一方のノードへの接続が失われました
	不整合	同期が進行中でしたが、中断されました
	スナップショットへの復帰	キュー・マネージャーが不整合状態になった時に取られたスナップショットに復帰することをユーザーが選択しました。
	リモート・システムが未構成	RDQM の 1 次インスタンスは構成されていますが、2 次インスタンスが構成されていません
	ネゴシエーションの失敗	一方のノードで同期複製が設定され、もう一方のノードで非同期複製が設定されています
DR タイプ	同期または非同期	常に表示されます。
DR ポート	*port_number* (このキュー・マネージャーのデータを複製するための TCP/IP ポート)	常に表示されます。
DR ローカル IP アドレス	このキュー・マネージャーが DR の複製を実行している複製元のローカル IP アドレス	常に表示されます。
DR リモート IP アドレス	このキュー・マネージャーが DR の複製を実行している複製先のリモート IP アドレス	常に表示されます。
DR 非同期データ	*n*KB	リモート・ノードが使用できないか、または不整合である場合に表示されます。
DR 同期の進行状況	*n*%	同期の進行中に表示されます

表 34. Status attributes (続き)		
状況属性	**可能な値**	**表示される時期**
DR 完了までの推定時間	*YYYY-MM-DD HH:MM:SS*	同期の進行中に表示されます
スナップショットへの復帰の進行中	*n*%	DR 状況が `Reverting to snapshot` の場合に表示されます。 状況がカウントダウンで表示されるので、0% になったら完了です。
V 9.3.0 V 9.3.0 最後の DR 同期日時	*YYYY-MM-DD HH:MM:SS*	DR データが同期していない状況 (初期同期の後) で表示されます。 データが最後に同期したときの日時を指定します。

例

1 次ノードの正常状況の例:

```
Queue manager status:                   Running
CPU:                                    0.00
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Normal
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
```

2 次ノードの正常状況の例:

```
Queue manager status:                   Ended immediately
DR role:                                Secondary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.20.2
DR remote IP address:                   192.168.20.1
```

同期化が進行中の場合の 1 次ノードの状況の例:

```
Queue manager status:                   Running
CPU:                                    0.53
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Synchronization in progress
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
DR synchronization progress:            11.0%
DR estimated time to completion:        2017-09-06 14:55:05
```

1 次ノードがパーティション化されていることを示す例:

```
Queue manager status:                   Running
CPU:                                    0.02
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Partitioned
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
```

V 9.3.0 1 次ノードが 2 次ノードと同期していないことを示す例:

```
Queue manager status:                   Running
CPU:                                    0.00
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Remote unavailable
DR type:                                Asynchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
DR out of sync data:                    15932KB
DR last in sync:                        2020-07-27 16:01:47
```

V 9.3.0 以下に、OS カーネル・バージョン (RHEL 7.9) と DRBD カーネル・モジュール (ターゲットは RHEL 7.8) の間の不一致を示す要約状況の例を示します。 DRBD カーネル・モジュールがロードされ、必要なキュー・マネージャーが実行されていることが状況に報告される場合でも、この状況で実行されている OS カーネルをターゲットとするバージョンで DRBD カーネル・モジュールを更新する必要があります。

```
Node:                                   mqhavm07.exampleco.com
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-1127
DRBD version:                           9.1.1
DRBD kernel module status:              Loaded

Queue manager name:                     DRQM8
Queue manager status:                   Ended immediately
DR role:                                Secondary

Queue manager name:                     DRQM7
Queue manager status:                   Running
DR role:                                Primary
```

V 9.3.0 以下に、OS カーネル・バージョン (RHEL 7.9) と DRBD カーネル・モジュール (ターゲットは RHEL 7.6) の間の不一致を示す要約状況の例を示します。 この例では、バージョンの違いがさらに大きいため、DRBD カーネル・モジュールを正常にロードできません。 QM3 は DR キュー・マネージャーであり、1 次インスタンスとなるものですが、DRBD カーネル・モジュールが完全にロードされなかったため、Unknown の DR 状況で 2 次インスタンスとして報告されます。 この障害を解決するには、稼働中の OS カーネルをターゲットとするバージョンで DRBD カーネル・モジュールを更新する必要があります。

```
Node:                                   mqhavm57.exampleco.com
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-957
DRBD version:                           9.1.2+ptf.3
DRBD kernel module status:              Partially loaded

Queue manager name:                     QM3
Queue manager status:                   Status not available
DR role:                                Secondary
DR status:                              Unknown
```

関連資料

Linux rdqmstatus

Linux 災害復旧環境での操作

災害復旧構成の 2 次キュー・マネージャーに切り替えたい状況がいくつかあります。

災害時リカバリー

メイン・サイトで 1 次キュー・マネージャーが完全に失われた場合は、リカバリー・サイトで 2 次キュー・マネージャーを開始します。 アプリケーションは、リカバリー・サイトのキュー・マネージャーに再接続し、その 2 次キュー・マネージャーがアプリケーション・メッセージを処理します。 以前の構成に戻すための手順は、失敗の原因によって異なります。 例えば、メイン・ノードが完全に失われた場合と一時的に失われた場合とでは、手順が異なります。

メイン・サイトが一時的に失われた後に実行する手順については、613 ページの『リカバリー・ノードへの切り替え』を参照してください。 永久的な障害が発生した場合に実行する手順については、614 ページの『災害復旧構成の障害ノードの交換』を参照してください。

災害復旧テストのサポート
災害復旧構成をテストするには、一時的に 2 次インスタンスに切り替えて、アプリケーションが正常に接続できるかどうかを確認します。 1 次ノードで一時的な障害が発生した場合の切り替えと同じ手順を実行します。613 ページの『リカバリー・ノードへの切り替え』を参照してください。

スナップショットへの復帰
同期の進行中に 1 次ノードで障害が発生した場合は、同期の開始直前の 2 次キュー・マネージャー・データのスナップショットに復帰できます。 2 次サーバーを整合状態に復元できたら、1 次として実行します。 613 ページの『リカバリー・ノードへの切り替え』の説明に沿って 2 次を 1 次にすれば、スナップショットに戻ります。 キュー・マネージャーを開始する前に、スナップショットへの復帰が完了したことを (**rdqmstatus** コマンドを使用して) 確認する必要があります。

Linux リカバリー・ノードへの切り替え
メイン・サイトで災害が発生した場合、リカバリー・サイトへの切り替え手順を実行します。

このタスクについて

メイン・サイトで 1 次キュー・マネージャーが失われた場合は、リカバリー・サイトの 2 次キュー・マネージャーを 1 次にして開始します。 アプリケーションは、リカバリー・サイトのキュー・マネージャーに再接続し、そのキュー・マネージャーがアプリケーション・メッセージを処理します。 この手順を使用して、リカバリー・ノードをテストすることもできます。

重要 : 元のセカンダリ・インスタンスをプロモートする前に、キュー・マネージャーのプライマリ・インスタンスが実行できないか、または停止して、セカンダリ・インスタンスに作成されていることを確認する必要があります。 そうしないと、区分データが発生する可能性があります。

ルートとしてログインするか、mqm グループに属し、必要な sudo 構成を持つユーザーとしてログインする必要があります。

手順

1. この手順を使用して 2 次キュー・マネージャーをテストする場合 (つまり、1 次インスタンスがまだ実行中の場合) は、1 次インスタンスを停止して 2 次インスタンスとして再指定する必要があります。

   ```
   endmqm qmname
   rdqmdr -m qmname -s
   ```

2. リカバリー・ノードで以下のコマンドを入力して、2 次キュー・マネージャーを 1 次にします。

   ```
   rdqmdr -m qmname -p
   ```

3. 次のコマンドを入力して、キュー・マネージャーを開始します。

   ```
   strmqm qmname
   ```

4. アプリケーションがリカバリー・キュー・マネージャー上のキュー・マネージャーに再接続していることを確認します。 代替接続名のリストに 1 次と 2 次のキュー・マネージャーを指定してチャネルを定義しておくと、アプリケーションは自動的に新しい 1 次キュー・マネージャーに接続します。

次のタスク

障害が発生したノードがリストアされると、2 つのノードの間のリンクが機能している場合、キュー・マネージャーは、セカンダリ・キュー・マネージャー・インスタンスをプロモートしたリカバリー・ノード上で実行されているため、このノード上で開始できません。 通常の操作に戻るには、リカバリー・ノード上のキュー・マネージャーを停止してから、元のノード上のキュー・マネージャーをプライマリ・ロールに昇格させる必要があります。

関連資料
strmqm

rdqmdr

リカバリー *RDQM* キュー・マネージャーのテスト
メイン・サイトを中断することなく、RDQM 災害復旧構成内のキュー・マネージャーのリカバリー・インスタンスが正しく動作していることをテストできます。

このタスクについて

リカバリー・キュー・マネージャーをテストするには、メイン・ノードとリカバリー・ノードの間のインターフェースを無効にします。 2 次キュー・マネージャーを 1 次キュー・マネージャーにした後、スタンドアロン・キュー・マネージャーをテストできます。 テストが完了したら、インターフェースを復元し、テスト・キュー・マネージャーを削除します。 そして、キュー・マネージャーを、災害復旧構成の 2 次キュー・マネージャーとして再作成します。

手順

1. メイン・ノードとリカバリー・ノードの間のネットワーク接続を無効にします。
2. リカバリー・ノードで、キュー・マネージャーを 1 次キュー・マネージャーにします。

   ```
   rdqmdr -m QMname -p
   ```

   QMname は、キュー・マネージャーの名前です。
3. キュー・マネージャーを始動します。

   ```
   strmqm QMname
   ```
4. アプリケーションをキュー・マネージャーに接続し、予期されたとおりに動作するかテストします。
5. 以下の手順を実行して、キュー・マネージャーを終了します。

   ```
   endmqm QMname
   ```
6. 以下の手順を実行して、キュー・マネージャーを削除します。

   ```
   dltmqm QMname
   ```
7. メイン・アプライアンスとリカバリー・アプライアンスの間のネットワーク接続を復元します。
8. メイン・ノードで、以下のコマンドを実行して、災害復旧を最初に構成したときに使用した **crtmqm** コマンドを取得します。

   ```
   rdqmdr -d -m QMname
   ```
9. リカバリー・ノードで取得した **crtmqm** コマンドを実行して、2 次キュー・マネージャーを再作成します。 メイン・ノード上の 1 次キュー・マネージャーは、そのデータを 2 次キュー・マネージャーと同期して最新の状態にします。

Linux 災害復旧構成の障害ノードの交換
災害復旧構成のいずれかのノードを失った場合は、以下の手順を実行し、ノードを置き換えて災害復旧構成を復元できます。

このタスクについて

メインサイトのノードの修復が不可能になるような災害が発生した場合は、キュー・マネージャーがリカバリー・ノードで実行されている間に、障害が発生したノードを置き換えて、元の災害復旧構成を復元します。 置換後のノードでは、障害が発生したノードの ID を使用する必要があります。同じ名前と同じ IP アドレスを使用しなければなりません。

ルートとしてログインするか、mqm グループに属し、必要な sudo 構成を持つユーザーとしてログインする必要があります。

手順

メイン・サイトのキュー・マネージャーが失われた場合は、以下の手順を実行します。

1. リカバリー・ノードで以下のコマンドを実行して、2 次キュー・マネージャーに 1 次の役割を与えます。

   ```
   rdqmdr -m QMname -p
   ```

   QMname は、キュー・マネージャーの名前です。
2. 災害復旧を再構成するために置換後の 1 次ノードで実行する必要があるコマンドを取得します。

   ```
   rdqmdr -m QMname -d
   ```

   このコマンドの出力をコピーします。
3. 以下のコマンドを実行して、キュー・マネージャーを開始します。

   ```
   strmqm QMname
   ```
4. アプリケーションがリカバリー・ノード上のキュー・マネージャーに再接続したことを確認します。 代替接続名のリストに 1 次と 2 次のキュー・マネージャーを指定してチャネルを定義しておくと、アプリケーションは自動的に新しい 1 次キュー・マネージャーに接続します。
5. メイン・サイトで障害が発生したノードを置き換え、元のノードの災害復旧構成で使用していた同じ名前と同じ IP アドレスを構成します。 次に、ステップ 2 でコピーした **crtmqm** コマンドを実行して、災害復旧を構成します。 これで、キュー・マネージャーの二次インスタンスが作成され、プライマリーインスタンスはデータを二次インスタンスと同期化します。
6. 現在の 1 次インスタンスを終了します。
7. 同期が完了したら、リカバリー・ノードで実行している 1 次インスタンスをもう一度 2 次インスタンスに戻します。

   ```
   rdqmdr -m QMname -s
   ```
8. 置換後の 1 次ノードで、キュー・マネージャーの 2 次インスタンスを 1 次インスタンスにします。

   ```
   rdqmdr -m QMname -p
   ```
9. 置換後の 1 次ノードで、キュー・マネージャーを開始します。

   ```
   strmqm QMname
   ```

   これで、メイン・サイトで障害が発生する前の構成を復元できました。

関連資料

strmqm

rdqmdr

endmqm

DR RDQM での不整合の問題の解決

キュー・マネージャーの 1 次インスタンスと 2 次インスタンスの間で同期が失敗すると、DR 状況 `inconsistent` (不整合) が報告されることがあります。

このタスクについて

同期操作中に 1 次インスタンスへの複製接続が失われるため、キュー・マネージャーの 2 次インスタンスで不整合状態が報告されます。 この状態を解決するためのアクションをとることが必要な場合があります。 以下のイベント・シーケンスを考えてみましょう。

1. DR 1 次キュー・マネージャーが DR 2 次キュー・マネージャーと同期する
2. 1 次および 2 次の間で複製リンクが失われる
3. 1 次および 2 次の間で複製リンクがリストアされる

4. DR 2 次キュー・マネージャーが DR 1 次キュー・マネージャーに追いつくと、再同期が発生します。 この間、synchronization in progress (同期が進行中) の DR 状況が両方のキュー・マネージャーについて報告されます。
5. 再同期中に複製が再び失われた場合、DR 2 次側の状況は Inconsistent (不整合) と報告されます。

1 次キュー・マネージャーをホストするノードがまだ作動可能であり、複製リンクを復元できる場合、再同期は自動的に行われます。 不整合状態は、何もアクションを実行せずに解決されます。

1 次キュー・マネージャーをホストするノードが作動可能でなくなった場合、2 次キュー・マネージャーでのスナップショットへの復帰を実装することにより、不整合状態を解決できます。 この操作は、データを最後に認識された良好な状態に戻します。

手順

不整合状態を解決するには、以下のようにします。

1. リカバリー・ノードで、2 次インスタンスを 1 次インスタンスにします。

   ```
   rdqmdr -m qmname -p
   ```

   スナップショットへの復帰操作が開始されます。
2. リカバリー・ノードで、キュー・マネージャーの状況を調べて、スナップショットへの復帰操作が完了したことを確認します。

   ```
   rdqmstatus -m qmname
   ```

3. キュー・マネージャーの状況が Normal (正常) の場合は、キュー・マネージャーを開始します。

   ```
   strmqm qmname
   ```

DR RDQM でのパーティション (スプリット・ブレーン) 問題の解決
災害時リカバリー・ペアの両方のキュー・マネージャーが同時に 1 次ロールで実行されると、パーティション化問題が発生する可能性があります。

このタスクについて

元の 1 次インスタンスがメイン・ノードで実行を継続している間に、リカバリー・ノードでキュー・マネージャーの 2 次インスタンスをプロモートした場合、実質的に同じキュー・マネージャーの 2 つのバージョンが実行され、それぞれにキュー・マネージャー・データの独自のビューを持ちます。 各ノード上のキュー・マネージャーの DR 状況は、Partitioned と報告されます。

2 つのキュー・マネージャーのどちらがデータを最も正確に表示しているかを判断し、そのセットを保持して、他方を破棄する必要があります。 この操作を実行するには、**rdqmdr** コマンドを使用します。

2 つの手順があります。 最初の説明はメイン・ノードからのデータの保持、2 番目の説明はリカバリー・ノードからのデータの保持です。

手順

- メイン・ノード上のキュー・マネージャーのデータを維持するには、次のようにします。
  a) 両方のキュー・マネージャー・インスタンスが停止していることを確認します。
  b) リカバリー・ノード上のキュー・マネージャーが 2 次であることを指定します。

  ```
  rdqmdr -m qmname -s
  ```

  c) メイン・ノード上のキュー・マネージャーが 1 次であることを指定します。

  ```
  rdqmdr -m qmname -p
  ```

  同期が開始され、メイン・ノード上のキュー・マネージャーからのデータがリカバリー・ノードにコピーされます。

d) 同期の状況を確認します:

```
rdqmstatus -m qmname
```

e) 同期が完了したら、メイン・ノードでキュー・マネージャーを開始します。

```
strmqm qmname
```

- リカバリー・ノード上のキュー・マネージャーのデータを維持するには、次のようにします。
  a) 両方のキュー・マネージャー・インスタンスが停止していることを確認します。
  b) メイン・ノード上のキュー・マネージャーが 2 次であることを指定します。

```
rdqmdr -m qmname -s
```

  c) リカバリー・ノード上のキュー・マネージャーが 1 次であることを指定します。

```
rdqmdr -m qmname -p
```

  同期が開始され、リカバリー・ノード上のキュー・マネージャーからのデータがメイン・ノードにコピーされます。
  d) 同期の状況を確認します:

```
rdqmstatus -m qmname
```

  e) 同期が完了したら、リカバリー・ノード上のキュー・マネージャーをデモートします。

```
rdqmdr -m qmname -s
```

  f) メイン・ノード上のキュー・マネージャーをプロモートし、開始します。

```
rdqmdr -m qmname -p
strmqm qmname
```

災害復旧構成での *IP* アドレスの変更

災害復旧構成におけるインターフェースのいずれかの IP アドレスを変更すると、2 つのノード間の複製ができなくなります。

ご使用の DR ノードのいずれかの複製インターフェースについて、IP アドレスを変更する必要がある場合は、以下の手順を使用する必要があります。

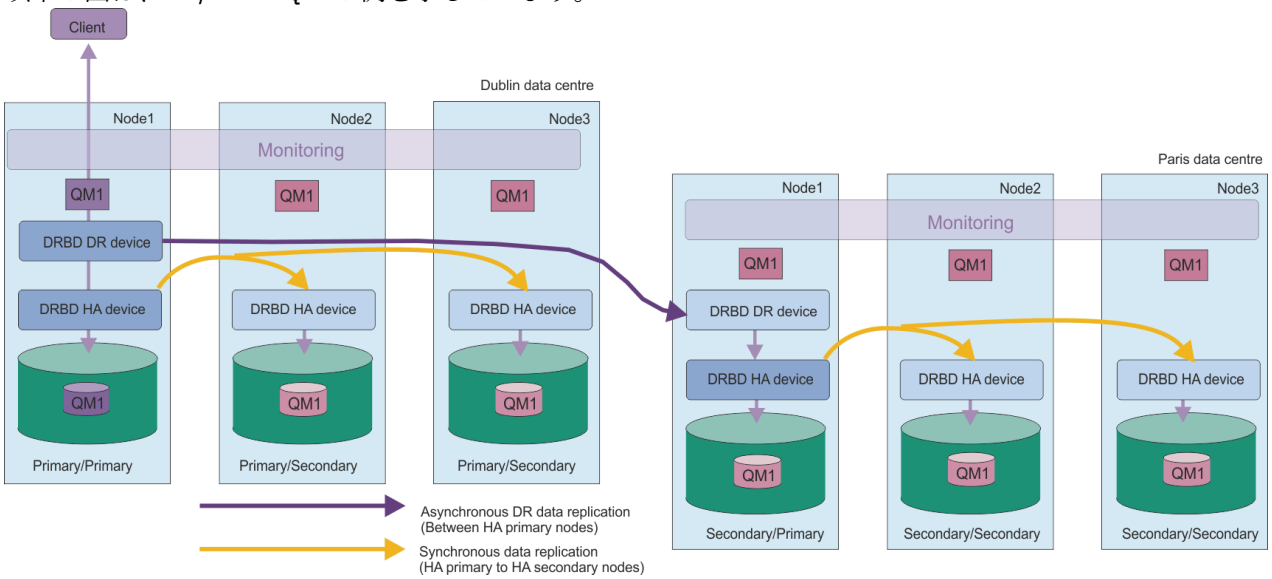
1. 1 次ノードで、DR キュー・マネージャーをバックアップしてから削除します。 リカバリー・ノードで、キュー・マネージャーを削除します。 671 ページの『IBM MQ キュー・マネージャー・データのバックアップと復元』 および 600 ページの『DR RDQM の削除』 を参照してください。
2. DR キュー・マネージャーを再作成し、新しい IP アドレスを指定して、バックアップをリストアします (597 ページの『災害復旧 RDQM の作成』 および 671 ページの『IBM MQ キュー・マネージャー・データのバックアップと復元』を参照してください)。

Linux RDQM 災害復旧および高可用性

1 つのサイトの高可用性グループで実行される複製データ・キュー・マネージャー (RDQM) を構成できますが、最初のグループを使用不可にする災害が発生した場合は、別のサイトの別の高可用性グループにフェイルオーバーすることができます。 これを DR/HA RDQM と呼びます。

DR/HA RDQM は、高可用性 RDQM (564 ページの『RDQM 高可用性』を参照) と災害復旧 RDQM (593 ページの『RDQM 災害復旧』を参照) の機能を結合しています。

以下の図は、DR/HA RDQM の例を示しています。



メイン・サイトと災害復旧サイトの DR/HA RDQM の複製は、常に非同期です。 非同期複製では、IBM MQ の PUT や GET などの操作が完了してアプリケーションに戻ってから、そのイベントが 2 次キュー・マネージャーに複製されます。

必要に応じて、メイン・サイトと復旧サイトではなく、2 つのアクティブ・サイトを持つことができます。これにより、通常の操作中は、一部の DR/HA RDQM が 1 つのサイトで実行され、一部の DR/HA RDQM が別のサイトで実行されます。 災害が発生して 1 つのサイトが使用不可になった場合は、すべての DR/HA RDQM が同じサイトの同じ HA グループで実行されます。

各 HA グループは、通常の HA グループと同様に構成されます。 各 HA グループ内の DR/HA RDQM の浮動 IP アドレスを定義できます。 浮動 IP アドレスは、HA グループごとに同じものにしたり、別のものにしたりすることができます。

既存の RDQM を DR/HA RDQM にアップグレードすることはできないため、DR/HA RDQM を作成する必要があります (必要な場合は、既存の RDQM のデータをバックアップしてから削除し、DR/HA RDQM として再作成してからデータを復元できます。671 ページの『IBM MQ キュー・マネージャー・データのバックアップと復元』を参照してください)。

DR/HA RDQM を構成するには、以下のステップを実行する必要があります。

1. 「メイン」サイトで HA グループを構成します。
2. 「復旧」サイトで HA グループを構成します。
3. 「メイン」サイトにある HA グループの 1 つのノード上に 1 次/1 次 DR/HA RDQM を作成します。
4. 「メイン」サイトの他の 2 つのノード上に、1 次/2 次 DR/HA RDQM を作成します。
5. DR/HA RDQM が「メイン」サイト上の HA グループのいずれかのノードで実行されている場合に、DR/HA RDQM にアクセスするアプリケーションの浮動 IP アドレスを定義します。
6. 「復旧」サイトにある HA グループの 1 つのノード上に 2 次/1 次 DR/HA RDQM を作成します。
7. 「復旧」サイトの他の 2 つのノード上に、2 次/2 次 DR/HA RDQM を作成します。
8. DR/HA RDQM が「復旧」サイト上の HA グループのいずれかのノードで実行されている場合に、DR/HA RDQM にアクセスするアプリケーションの浮動 IP アドレスを定義します。

上記の各ステップについて詳しくは、以下の各トピックで説明します。

Linux *DR/HA RDQM ソリューションの要件*

DR/HA RDQM ソリューションの要件は、HA RDQM ソリューションおよび DR RDQM ソリューションの要件と同じです。

構成の HA 部分の要件について詳しくは、566 ページの『RDQM HA ソリューションの要件』を参照してください。

構成の DR 部分について詳しくは、596 ページの『RDQM DR ソリューションの要件』を参照してください。

Linux *DR/HA RDQM 用の HA グループの構成*

メイン・サイトと復旧サイトの両方で HA グループを作成する必要があります。 いずれかのサイトに既存の HA グループがある場合は、その HA グループ内に DR/HA RDQM を作成できます (既存の RDQM は以前と同様に動作し続けます)。

この手順は RDQM 高可用性についての説明にあるものと同じです。570 ページの『Pacemaker クラスター (HA グループ) の定義』を参照してください。

高可用性グループを定義する場合は、`rdqm.ini` ファイル内の各ノードによるモニターおよび複製に使用する IP アドレスを指定します。 DR/HA RDQM をサポートする HA グループを作成する場合は、定義する HA グループによって DR 複製に使用される IP アドレス、および DR ペアの他方の HA グループ内のノードによって DR 複製に使用される IP アドレスを指定することもできます (DR 複製 IP アドレスを `rdqm.ini` ファイルに指定しない場合は、DR/HARDQM を作成するときに、コマンド行で IP アドレスを指定することができます。)

既存の HA グループを構成している場合は、DR 複製 IP アドレスを既存の `rdqm.ini` ファイルに追加できます。 `rdqm.ini` の更新後に **rdqmadm** を再度実行する必要はありませんが、`rdqm.ini` を更新してから DR/HA RDQM を作成する必要があります。

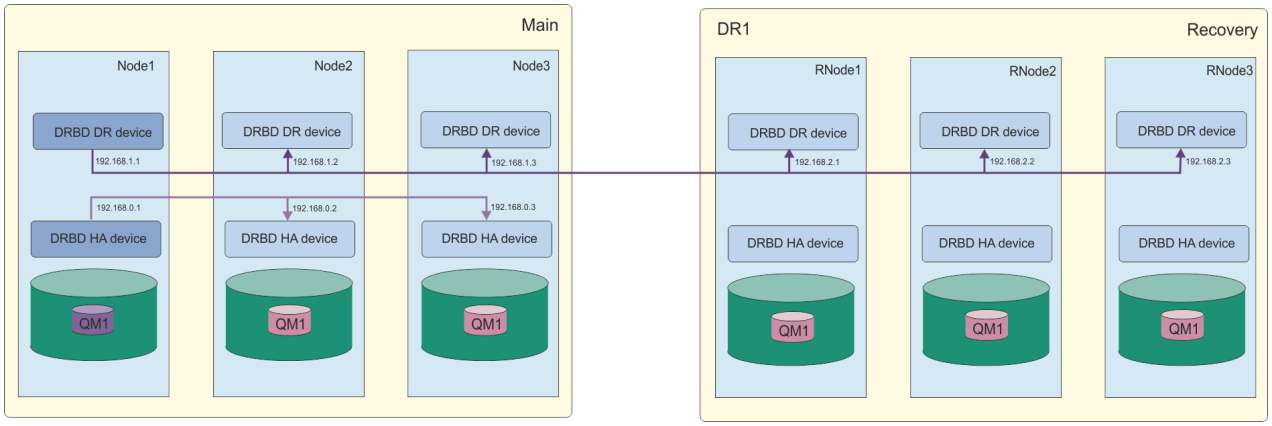
Node スタンザの DR_Replication 属性を使用して、定義する HA グループ上の DR 複製インターフェースを指定します。以下に例を示します。

```
Node:
  Name=Node1
  HA_Replication=192.168.0.1
  DR_Replication=192.168.1.1
Node:
  Name=Node2
  HA_Replication=192.168.0.2
  DR_Replication=192.168.1.2
Node:
  Name=Node3
  HA_Replication=192.168.0.3
  DR_Replication=192.168.1.3
```

DRGroup スタンザを使用して、リモート HA グループの DR 複製アドレスを指定します。以下に例を示します。

```
DRGroup:
  Name=DR1
  DR_Replication=192.168.2.1
  DR_Replication=192.168.2.2
  DR_Replication=192.168.2.3
```

次の図は、この構成を示しています:



`rdqm.ini` ファイルまたはコマンド・ラインのいずれかでローカル HA グループ内のノードに DR 複製の IP アドレスを指定しなかった場合、DR/HA RDQM を作成すると、各ノードに定義された HA_Replication

インターフェースが DR 複製に使用されます。 リモート HA グループの DR 複製アドレスは、rdqm.ini ファイルまたは **crtmqm** コマンド・ラインのいずれかで指定する必要があります。

Linux *DR/HA RDQM の作成*

crtmqm コマンドを使用して、DR/HA 構成内に複製データ・キュー・マネージャー (RDQM) を作成します。

このタスクについて

ユーザーが sudo を使用できる場合は、mqm グループのユーザーとして DR/HA RDQM を作成できます。 そうでない場合は、RDQM を root として作成する必要があります。

複数の DR/HA RDQM を作成する必要があります。

- 「メイン」サイトの HA グループで、以下を実行します。
  - キュー・マネージャーを通常の条件下で実行するノードで、1 次/1 次 DR/HA RDQM を作成します。
  - HA グループ内のその他の 2 つのノードのそれぞれで、1 次/2 次 DR/HA RDQM を作成します。
- 「復旧」サイトの HA グループで、以下を実行します。
  - キュー・マネージャーが復旧サイトにフェイルオーバーした場合に、キュー・マネージャーが実行されるノードで 1 次/2 次 DR/HA RDQM を作成します。 「メイン」サイトで 1 次/1 次キュー・マネージャーを作成したときのコマンド出力を使用できます。
  - HA グループ内のその他の 2 つのノードのそれぞれで、2 次/2 次 DR/HA RDQM を作成します。

すべてのキュー・マネージャー・インスタンスの名前は同じにし、同じ量のストレージを割り振る必要があります。

以下に、キュー・マネージャーのファイル・システムのサイズ変更に関するガイダンスを示します。

1. RDQM キュー・マネージャーの作成時に、キュー・マネージャーのデータとログを保管するためにファイル・システムが割り振られます。 キュー・マネージャーが進行中のアクティビティーをログに記録し、アプリケーション・メッセージをキューに保管できるように、このファイル・システムを適切にサイズ変更することが重要です。 ファイル・システムをサイジングする際には、ピーク時のメッセージング要件、将来のワークロードの増加、およびメッセージがキューに蓄積される原因となる可能性のあるアプリケーション障害を考慮してください。 キュー・マネージャーの回復ログのサイズを計算するためのガイダンスについては、651 ページの『ログ・ファイル・システムの大きさの決定方法』を参照してください。 アプリケーション・メッセージのストレージ要件を計算する際には、メッセージのサイズと数に加えて、MQMD ヘッダーおよびメッセージ・プロパティーを考慮する必要があります。
2. RDQM キュー・マネージャーのファイル・システムを動的にサイズ変更することはできません。 これが必要な場合は、より大きなファイル・システムを使用して RDQM キュー・マネージャーをバックアップしてからリストアする必要があります。 579 ページの『HA RDQM キュー・マネージャーのファイル・システムのサイズ変更』を参照してください。
3. MAXDEPTH や MAXFSIZE などのローカル・キュー属性を使用して、ディスク上の個々のキューのサイズを制限することができます。 IBM MQ キュー・ファイルの変更を参照してください。
4. ファイル・システムの使用量が重要になる前に、進行中のディスク使用量をモニターし、ディスク使用量が増加した場合は適切に対応する必要があります。 ファイル・システムの使用状況をモニターするには、プラットフォーム/オペレーティング・システムの機能を使用するか、システム・トピックでパブリッシュされるメトリックで説明されている IBM MQ システム・トピックにパブリッシュされるメトリックをサブスクライブします。

手順

- 1 次/1 次 DR/HA RDQM を作成するには、以下のようにします。

  a) 以下のコマンドを入力します。

```
crtmqm -sx -rr p
              [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
              (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
              -rp DRPort
              [-z] [-q] [-c Text] [-d DefXmitQ] [-h MaxHandles]
```

```
[-g ApplicationGroup] [-oa user|group]
[-t TrigInt] [-u DeadQ] [-x MaxUMsgs]
[-lp LogPri] [-ls LogSec]
[-lc | -ll | -lla | -lln] [-lf LogFileSize]
[-p Port] [-fs FilesystemSize] QMgrName
```

説明

-sx
初期 HA 役割が 1 次であることを示します。

-rr p
初期 DR 役割が 1 次であることを示します。

-rl *DRLocalIP1*, *DRLocalIP2*, *DRLocalIP3*
オプションで、ローカル・サイトの 3 つのノード上の DR インターフェースの IP アドレスを指定します (「メイン」サイト)。 指定しない場合は、`rdqm.ini` ファイルで指定された IP アドレスが使用されます。

-ri *DRRemoteIP1*, *DRRemoteIP2*, *DRRemoteIP3*
リモート・サイトの 3 つのノード上の DR インターフェースの IP アドレスを指定します (「復旧」サイト)。 このパラメーターまたは `-rn` パラメーターのいずれかを指定する必要があります。

-rn *GroupName*
`rdqm.ini` ファイルに指定されているリモート HA グループ名を指定します。 `-ri` または `-rn` のいずれかを指定する必要があります。

-rp *Port*
DR レプリケーションに使用するポートを指定します。

other_crtmqm_options
(オプション) 以下の一般的な **crtmqm** オプションを 1 つ以上指定することもできます。

- -z
- -q
- -c *Text*
- -d *DefaultTransmissionQueue*
- -h *MaxHandles*
- -g *ApplicationGroup*
- -oa user | group
- -t *TrigInt*
- -u *DeadQ*
- -x *MaxUMsgs*
- -lp *LogPri*
- -ls *LogSec*
- -lc | -l
- -lla | -lln
- -lf *LogFileSize*
- -p *Port*

-fs *size*
(オプション) キュー・マネージャー用に作成するファイル・システムのサイズ、つまり drbdpool ボリューム・グループ内で作成する論理ボリュームのサイズを指定します。 スナップショットへの復帰操作をサポートするために、そのサイズの別の論理ボリュームも作成するので、DR RDQM の合計ストレージは、ここで指定した値の 2 倍になります。

Size は、GB 単位で指定される数値です。 値を MB 単位で指定するには、値の後に文字 `M` を入力します。 例えば、3 GB のファイル・システム・サイズを指定するには、3 を入力します。 1024 MB のファイル・システム・サイズを指定するには、 1024M と入力します。 (G 接尾部を明示的に状態 GB に追加することもできます。)

QMname
複製データ・キュー・マネージャーの名前を指定します。 この名前には大/小文字の区別があります。

コマンドが完了すると、キュー・マネージャーの 2 次/1 次インスタンスを作成するために復旧サイトで入力できるコマンドが出力されます。

- HA グループのその他の 2 つのノードで 1 次/2 次 DR/HA RDQM を作成するには、以下のようにします。

  a) 各ノードで次のコマンドを入力します。

```
crtmqm -sxs -rr p
             [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
             (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
             -rp DRPort
             [-fs FilesystemSize] QMgrName
```

  説明

  -sxs
  初期 HA 役割が 2 次であることを示します。

  -rr p
  初期 DR 役割が 1 次であることを示します。

  -rl *DRLocalIP1, DRLocalIP2, DRLocalIP3*
  オプションで、ローカル・サイトの 3 つのノード上の DR インターフェースの IP アドレスを指定します (「メイン」サイト)。 指定しない場合は、`rdqm.ini` ファイルで指定された IP アドレスが使用されます。

  -ri *DRRemoteIP1, DRRemoteIP2, DRRemoteIP3*
  リモート・サイトの 3 つのノード上の DR インターフェースの IP アドレスを指定します (「復旧」サイト)。 このパラメーターまたは `-rn` パラメーターのいずれかを指定する必要があります。

  -rn *GroupName*
  `rdqm.ini` ファイルに指定されているリモート HA グループ名を指定します。 `-ri` または `-rn` のいずれかを指定する必要があります。

  -rp *Port*
  DR レプリケーションに使用するポートを指定します。

  -fs *size*
  キュー・マネージャー用に作成するファイル・システムのサイズ、つまり drbdpool ボリューム・グループ内で作成する論理ボリュームのサイズを指定します。 1 次/1 次 RDQM の作成時にデフォルト以外のサイズを指定した場合は、ここで同じ値を指定する必要があります。

  Size は、GB 単位で指定される数値です。 値を MB 単位で指定するには、値の後に文字 `M` を入力します。 例えば、3 GB のファイル・システム・サイズを指定するには、3 を入力します。 1024 MB のファイル・システム・サイズを指定するには、 1024M と入力します。 (G 接尾部を明示的に状態 GB に追加することもできます。)

  QMname
  1 次/2 次 RDQM の名前を指定します。 RDQM の 1 次/1 次インスタンスで指定した名前と同じである必要があります。 名前には大/小文字の区別があります。

- キュー・マネージャーが復旧サイトにフェイルオーバーした場合に、キュー・マネージャーが実行されるノードで 2 次/1 次 DR/HA RDQM を作成するには、以下のようにします。

  a) メイン・サイトで 1 次/1 次 DR/HA を作成したときのコマンド出力を使用するか、以下のコマンドを入力します。

```
crtmqm -sx -rr s
             [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
             (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
             -rp DRPort
             [-fs FilesystemSize] QMgrName
```

  -sx
  初期 HA 役割が 1 次であることを示します。

-rr s
初期 DR 役割が 2 次であることを示します。

-rl *DRLocalIP1*, *DRLocalIP2*, *DRLocalIP3*
オプションで、ローカル・サイトの 3 つのノード上の DR インターフェースの IP アドレスを指定します (「復旧」サイト)。 指定しない場合は、`rdqm.ini` ファイルで指定された IP アドレスが使用されます。

-ri *DRRemoteIP1*, *DRRemoteIP2*, *DRRemoteIP3*
リモート・サイトの 3 つのノード上の DR インターフェースの IP アドレスを指定します (「メイン」サイト)。 このパラメーターまたは `-rn` パラメーターのいずれかを指定する必要があります。

-rn *GroupName*
`rdqm.ini` ファイルに指定されているリモート HA グループ名を指定します。 `-ri` または `-rn` のいずれかを指定する必要があります。

-rp *Port*
DR レプリケーションに使用するポートを指定します。

-fs *size*
(オプション) キュー・マネージャー用に作成するファイル・システムのサイズ、つまり drbdpool ボリューム・グループ内で作成する論理ボリュームのサイズを指定します。 スナップショットへの復帰操作をサポートするために、そのサイズの別の論理ボリュームも作成するので、DR RDQM の合計ストレージは、ここで指定した値の 2 倍になります。

QMname
複製データ・キュー・マネージャーの名前を指定します。 この名前には大/小文字の区別があります。

- 復旧サイトのその他の 2 つのノードで 2 次/2 次 HA/DR RDQM を作成するには、以下のようにします。
  a) 各ノードで次のコマンドを入力します。

```
crtmqm -sxs -rr s
              [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
              (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
              -rp DRPort
              [-fs FilesystemSize] QMgrName
```

-sxs
初期 HA 役割が 1 次であることを示します。

-rr s
初期 DR 役割が 2 次であることを示します。

-rl *DRLocalIP1*, *DRLocalIP2*, *DRLocalIP3*
オプションで、ローカル・サイトの 3 つのノード上の DR インターフェースの IP アドレスを指定します。 指定しない場合は、`rdqm.ini` ファイルで指定された IP アドレスが使用されます。

-ri *DRRemoteIP1*, *DRRemoteIP2*, *DRRemoteIP3*
リモート・サイト上の 3 つのノード上の DR インターフェースの IP アドレスを指定します。 このパラメーターまたは `-rn` パラメーターのいずれかを指定する必要があります。

-rn *GroupName*
`rdqm.ini` ファイルに指定されているリモート HA グループ名を指定します。 `-ri` または `-rn` のいずれかを指定する必要があります。

-rp *Port*
DR レプリケーションに使用するポートを指定します。

-fs *size*
(オプション) キュー・マネージャー用に作成するファイル・システムのサイズ、つまり drbdpool ボリューム・グループ内で作成する論理ボリュームのサイズを指定します。 スナップショットへの復帰操作をサポートするために、そのサイズの別の論理ボリュームも作成するので、DR RDQM の合計ストレージは、ここで指定した値の 2 倍になります。

QMname
複製データ・キュー・マネージャーの名前を指定します。 この名前には大/小文字の区別があります。

注 : RDQM を作成すると、HA 複製リンクには 7000 より上の空いている次のポート番号が割り振られます。 選択されたポートが別のアプリケーションによって使用されていることが検出された場合、**crtmqm** コマンドは AMQ6543 エラーで失敗し、そのポートが除外リストに追加されます。 キュー・マネージャーの 2 次インスタンスを削除してから、**crtmqm** コマンドを再実行する必要があります。

次のタスク

すべての DR/HA RDQM を作成したら、1 次/1 次インスタンスと 2 次/1 次インスタンスの状況を調べて、すべてが正しいことを確認する必要があります。 ノードで **rdqmstatus** コマンドを使用します。 どちらのノードでも正常状況が表示されるはずです (626 ページの『DR/HA RDQM および HA グループの状況の表示』を参照)。 その状況が表示されない場合は、2 次/1 次インスタンスをいったん削除して再作成してください。その際に、正しい引数を使用するように注意してください。

関連タスク

620 ページの『DR/HA RDQM の作成』

crtmqm コマンドを使用して、DR/HA 構成内に複製データ・キュー・マネージャー (RDQM) を作成します。

関連資料

crtmqm

Linux

DR/HA RDQM の削除

dltmqm コマンドを使用して、DR/HA 複製データ・キュー・マネージャー (RDQM) を削除します。

このタスクについて

1 次/1 次ノードと 2 次/1 次ノードの両方で RDQM を削除するには、このコマンドを実行する必要があります。 最初に RDQM を終了する必要があります。 mqm ユーザーが必要な sudo 特権を持っている場合は、mqm ユーザーとしてコマンドを実行できます。 その他の場合は、root としてコマンドを実行する必要があります。

手順

- DR/HA RDQM を削除するには、以下のコマンドを入力します。

```
dltmqm RDQM_name
```

関連資料

dltmqm

Linux

浮動 *IP* アドレスの作成

DR/HA RDQM 構成内の各 HA グループに浮動 IP アドレスを作成できます。

浮動 IP アドレスを使用すると、HA グループ内のどのノードでクライアントが実行されているかにかかわらず、DR/HA RDQM と同じ IP アドレスを使用することができます。 2 つの HA グループにアプリケーション接続用の分離された専用ネットワークがある場合は、両方のグループに同じ浮動 IP アドレスを定義できます。 ただし、この浮動 IP アドレスは、HA グループごとに 1 回ずつ、2 回定義する必要があります。

浮動 IP アドレスは、HA RDQM の場合と同じ方式を使用して作成および削除します。 582 ページの『浮動 IP アドレスの作成および削除』を参照してください。

Linux

DR/HA RDQM の開始、停止、および状態の表示

標準 IBM MQ 制御コマンドのバリアントを使用して、DR/HA RDQM を開始または停止したり、現在の状態を表示したりできます。

このタスクについて

mqm グループと haclient グループの両方に属するユーザーとして、DR/HA RDQM を開始または停止したり、現在の状態を表示したりするコマンドを実行する必要があります。

そのキュー・マネージャーの 1 次ノードでキュー・マネージャーを開始および停止するには、コマンドを実行する必要があります。

手順

- RDQM を開始するには、RDQM の 1 次ノードで以下のコマンドを入力します。

  ```
  strmqm qmname
  ```

  ここで、*qmname* は、開始する DR/HA RDQM の名前です。

  RDQM が開始され、Pacemaker が RDQM の管理を開始します。 その他の `strmqm` オプションを指定する場合は、`strmqm` と共に `-ns` オプションを指定する必要があります。

- RDQM を停止するには、DR/HA RDQM の 1 次ノードで以下のコマンドを入力します。

  ```
  endmqm qmname
  ```

  ここで、*qmname* は、停止する RDQM の名前です。

  Pacemaker は、RDQM の管理を停止し、次に RDQM が終了します。 他のすべての **endmqm** パラメーターは、RDQM を停止するときに使用できます。

- RDQM の状態を表示するには、以下のコマンドを入力します。

  ```
  dspmq -m QMname
  ```

  出力される状態情報は、コマンドを RDQM の 1 次ノードまたは 2 次ノードのいずれで実行するかによって異なります。 1 次ノードで実行する場合は、**dspmq** によって返された正常状況メッセージの 1 つが表示されます。 このコマンドを 2 次ノードで実行すると、状況 `Ended immediately` が表示されます。 例えば、**dspmq** がノード RDQM7 で実行されている場合、以下の情報が返されることがあります。

  ```
  QMNAME(DRQM8)                           STATUS(Ended immediately)
  QMNAME(DRQM7)                           STATUS(Running)
  ```

  `dspmq` の引数を使用して、RDQM が災害復旧用として構成されているかどうか、現時点で 1 次インスタンスになっているか 2 次インスタンスになっているかを確認できます。

  ```
  dspmq -m QMname -o (dr | DR)
  ```

  以下のいずれかの応答が表示されます。

  DRROLE()
  キュー・マネージャーが災害復旧用に構成されていないことを示しています。

  DRROLE(Primary)
  キュー・マネージャーが DR 1 次として構成されていることを示しています。

  DRROLE(Secondary)
  キュー・マネージャーが DR 2 次として構成されていることを示しています。

  DR/HA RDQM の災害復旧および高可用性情報を表示するには、**dspmq -o all** コマンドを使用します。 例えば、DR/HA RDQM が実行されているノード上で **dspmq -o all** を実行すると、以下の状態情報が表示されます。

  ```
  QMNAME(TESTQM1)                                           STATUS(Running) HA(Replicated)
  DRROLE(Primary)
  ```

関連資料
dspmq (キュー・マネージャーの表示)
endmqm (キュー・マネージャーの終了)
strmqm (キュー・マネージャーの始動)

V 9.3.0 *DR/HA* 構成での失敗したリソース・アクション

RDQM 高可用性構成の Pacemaker コンポーネントが、HA グループ内のいずれかのノード上のリソースに関連した何らかの問題を検出すると、失敗したリソースのアクションが発生します。

失敗したリソース・アクションは、RDQM DR/HA 構成のいずれかの HA 構成で発生する可能性があります。**rdqmstatus** コマンドを使用して、失敗したリソース・アクションを表示し、**rdqmclean** コマンドを使用して (失敗の原因が解決された後で) それらをクリアすることができます。 このプロセスは、DR コンポーネントを持たない RDQM HA 構成と同じです。 詳細については、585 ページの『失敗したリソースのアクション』を参照してください。

関連タスク

626 ページの『DR/HA RDQM および HA グループの状況の表示』
DR/HA 複製データ・キュー・マネージャー (RDQM) の HA の状況および DR 役割を表示できます。

586 ページの『RDQM および HA グループの状況の表示』
HA グループおよび個々の複製データ・キュー・マネージャー (RDQM) の状況を表示できます。

関連資料

rdqmclean

rdqmstatus

Linux *DR/HA RDQM* および *HA* グループの状況の表示

DR/HA 複製データ・キュー・マネージャー (RDQM) の HA の状況および DR 役割を表示できます。

このタスクについて

個々の RDQM の状況を表示したり、HA グループに認識されているすべての RDQM の状況の概要を表示したりするには、**rdqmstatus** コマンドを使用します。

V 9.3.0 ノードの要約状況には、RDQM が依存している DRBD カーネル・モジュールに関する情報も表示されます。 RDQM をアップグレードする場合は、システム上で実行されている RHEL カーネルのバージョン用に正しいバージョンの DRBD カーネル・モジュールがインストールされていることを確認することが重要です。 この状況には、OS カーネルのバージョン、DRBD モジュールが作成されたカーネル・バージョン、DRBD バージョン、および DRBD カーネル・モジュールのロード状況が表示されます。

注 : HA/DR 構成では、DR 構成は常に非同期複製を使用し、HA 構成は常に同期複製を使用することに注意してください。 これらの値は、結合された HA/DR 構成の `rdqmstatus -m qmgr` コマンドの出力には表示されません。

rdqmstatus コマンドを実行するには、`mqm` グループと `haclient` グループのユーザーである必要があります。 このコマンドは、HA グループのいずれかのノードで実行できます。

手順

- HA 構成の一部であるノードおよび RDQM の要約状況を表示するには、次のようにします。

  ```
  rdqmstatus
  ```

  コマンドを実行したノードの ID と HA 構成の RDQM の状況および現在の役割が、以下のように表示されます。

  ```
  Node:                                 main-alice
  OS kernel version:                    3.10.0-1160.15.2
  DRBD OS kernel version:               3.10.0-1160
  DRBD version:                         9.1.1
  DRBD kernel module status:            Loaded

  Queue manager name:                   RDQM1
  Queue manager status:                 Running elsewhere
  HA current location:                  main-charlie
  HA preferred location:                main-charlie
  HA blocked location:                  None

  Queue manager name:                   RDQM9
  ```

```
Queue manager status:                   Running elsewhere
HA current location:                    main-bob
HA preferred location:                  main-bob
HA blocked location:                    None
DR role:                                Primary

Queue manager name:                     RDQM7
Queue manager status:                   Running
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
DR role:                                Primary
```

この例では、RDQM7 および RDQM8 は両方とも DR/HA RDQM です。RDQM1 は HA RDQM であり、災害復旧サイトに切り替えられるように構成されていません。

V 9.3.0 DRBD カーネル・モジュール状況は、以下のいずれかの値です。

ロード済み (Loaded)
DRBD モジュールがロードされたことを示します。

一部ロード (Partially loaded)
DRBD モジュールがロードされたが、不一致が原因で正しく機能しない場合に発生する可能性があります。

ロードされていません (Not loaded)
DRBD モジュールはロードされていません。 これは、RDQM キュー・マネージャーがまだ作成されていない場合に、新規にインストールされた構成で表示される可能性があります。

インストールされていません
DRBD モジュールがインストールされていないか、 IBM MQ が DRBD モジュールの OS カーネル・バージョンを判別できなかったことを示します。

以前にインストールされたバージョンがまだロードされています (Previously installed version still loaded)
この状況は、既存の DRBD モジュールの実行中 (つまり、RDQM キュー・マネージャーの実行中) に新規 DRBD モジュールがインストールされた場合に発生する可能性があります。 新規にインストールされたモジュールが状況で報告されますが、これは実際に実行されているモジュールではありません。

- HA グループのすべてのノードの特定のキュー・マネージャーの状況を表示するには、以下のコマンドを入力します。

```
rdqmstatus -m qmname
```

ここで、*qmname* は、状況を表示する RDQM の名前です。 現行ノードの RDQM の状況が表示され、その後に現行ノードの観点からの他の 2 つのノードの状況の要約が表示されます。

- V 9.3.0

HA グループのすべてのノードの特定のキュー・マネージャーの状況を、失敗したリソース・アクションの詳細を含めて表示するには、以下のコマンドを入力します。

```
rdqmstatus -m qmname -a
```

ここで、*qmname* は、状況を表示する RDQM の名前です。 現行ノードの RDQM の状況が表示され、その後に現行ノードの観点からの他の 2 つのノードの状況の要約が表示されます。 この後に、RDQM に関連付けられている失敗したリソース・アクションに関する詳細が続きます。

以下の表は、RDQM の `rdqmstatus -m qmname` コマンドによって返される可能性がある現行ノードに関する情報を要約したものです。

表 *35.* 現行ノードの状況

状況属性	可能な値	表示される時期
ノード名	*nodename*	常に表示されます。

表 35. 現行ノードの状況 (続き)		
状況属性	**可能な値**	**表示される時期**
キュー・マネージャー状況	キュー・マネージャーの状態 (**dspmq** コマンドに対して有効な状態の 1 つ)	常に表示されます。
CPU	*n.nn*%	このノードで RDQM が実行されている場合にのみ表示されます。
メモリー	*nnn*MB used	このノードで RDQM が実行されている場合にのみ表示されます。
キュー・マネージャー・ファイル・システム	*nnn*MB used、*y.y*GB allocated [*z*%]	このノードで RDQM が実行されている場合にのみ表示されます。
HA 役割	1 次 2 次 不明	常に表示されます。
HA 状況	すべてのノードがスタンバイ状態です このノードはスタンバイ状態です Remote nodes in standby 混合	すべてのノードがスタンバイ状態です 現行ノードがスタンバイ状態の場合 両方のリモート・ノードがスタンバイ状態の場合 リモート・ノードごとに状況が異なる場合
HA 制御	有効 無効 不明	常に表示されます。 RDQM が Pacemaker 制御下にあるかどうかが示されます。
HA 優先ロケーション	なし このノード 不明 *nodename*	常に表示されます。

表 35. 現行ノードの状況 (続き)		
状況属性	**可能な値**	**表示される時期**
V 9.3.0 V 9.3.0 HA ブロック・ロケーション	なし - このキュー・マネージャーはどのノードでも実行がブロックされることはありません。 このノード - 1 つ以上のリソース・アクションが失敗したため、現在のノードでのキュー・マネージャーの実行がブロックされています。 *nodename* - 1 つ以上のリソース・アクションが失敗したため、*nodename* でのキュー・マネージャーの実行がブロックされています。 *nodename1*、*nodename2* - 1 つ以上のリソース・アクションが失敗したため、*nodename1* および *nodename2* でのキュー・マネージャーの実行がブロックされています。 すべてのノード - 1 つ以上のリソース・アクションが失敗したため、すべてのノードでキュー・マネージャーの実行がブロックされています。	常に表示されます。
HA 浮動 IP インターフェース	*Interface_name*	常に表示されます。
HA 浮動 IP アドレス	*IPV4_address*	常に表示されます。
DR 役割	1 次 2 次 2 次保留 不明	常に表示されます。

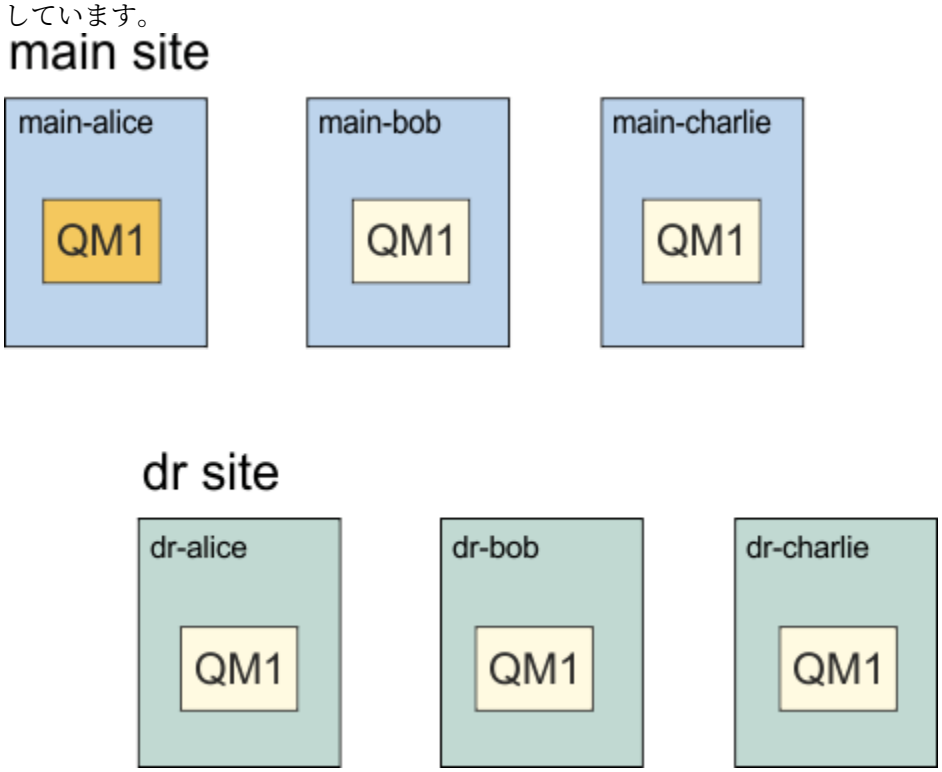
表 35. 現行ノードの状況 (続き)

状況属性	可能な値	表示される時期
DR 状況	通常 同期が進行中 パーティション リモート・システムが使用不可 不整合 スナップショットへの復帰 リモート・システムが未構成 ネゴシエーションの失敗	すべて正常です。 同期が進行中です。 ユーザーがキューを開始した 各ノードの管理者 DR レプリケーション・ネットワーク使用不可。 他のノードへの接続 失われました。 同期が進行中でしたが、中断されました。 ユーザーは、復帰先を選択しました 使用されたときに取られたスナップショット 入力されたキュー・マネージャー 不整合状態です。 1 次が構成されましたが、 2 次は構成されませんでした。 1 次ノードと 2 次ノードの間の初期ネゴシエーションが失敗しました。 これは、複製タイプが非互換であるか、または 2 次ノードがより小さいファイル・システム・サイズを使用して構成されている場合に発生します。
DR 状況 (HA 2 次ノード上)	*HA_Primary_Node* を参照してください。	DR 状況が HA 1 次ノードでのみ認識されているため、HA 2 次ノードで表示されます。
DR ポート	このキュー・マネージャーのデータを複製するための TCP/IP ポート。	常に表示されます。
DR ローカル IP アドレス	このキュー・マネージャーが DR 複製に使用するローカル IP アドレス	常に表示されます。
DR リモート IP アドレス・リスト	このキュー・マネージャーが DR 複製に使用するリモート IP アドレス。 3 つの IP アドレスのコンマ区切りリスト。	常に表示されます。
DR 現行リモート IP アドレス	このキュー・マネージャーが DR 複製用に接続されている現在のリモート IP	アクティブな DR 接続を持つ HA 1 次の場合
DR 現行リモート IP アドレス (HA 2 次ノード上)	*HA_Primary_Node* を参照してください。	DR 接続が HA 1 次ノード上にのみあるため、HA 2 次ノードで表示されます。
DR 非同期データ	*x*KB	リモート・ノードが使用できないか、または不整合である場合に表示されます。
DR 同期の進行状況	*y*%	同期の進行中に表示されます。
DR 完了までの推定時間	*yyyy-MM-dd HH:mm:ss*	同期の進行中に表示されます。

表 35. 現行ノードの状況 (続き)		
状況属性	**可能な値**	**表示される時期**
スナップショットへの復帰の進行中	*y*%	DR 状況が「スナップショットに戻ります」の場合に表示されます。
V 9.3.0 V 9.3.0 最後の DR 同期日時	*yyyy-MM-dd HH:mm:ss*	DR データが同期していない状況 (初期同期の後) で表示されます。データが最後に同期したときの日時を指定します。

例

これらの例は、以下の DR/HA 構成のさまざまなノードで実行されるコマンド **rdqmstatus -m qm1** を示しています。



DR 1 次および HA 1 次であるノードでの正常状況の例:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           192.168.2.1

Node:                                   main-bob
HA status:                              Normal
```

```
Node:                                   main-charlie
HA status:                              Normal
```

DR 1 次および HA 2 次であるノードでの正常状況の例:

```
Node:                                   main-bob
Queue manager status:                   Running elsewhere
HA role:                                Secondary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    main-alice
HA preferred location:                  main-alice
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              See main-alice
DR port:                                3000
DR local IP address:                    192.168.1.2
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           See main-alice

Node:                                   main-alice
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

DR 2 次および HA 1 次であるノードでの正常状況の例:

```
Node:                                   dr-alice
Queue manager status:                   Ended immediately
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Secondary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.2.1
DR remote IP address list:              192.168.1.1,192.168.1.2,192.168.1.3
DR current remote IP address:           192.168.1.1

Node:                                   dr-bob
HA status:                              Normal

Node:                                   dr-charlie
HA status:                              Normal
```

DR 2 次および HA 2 次であるノードでの正常状況の例:

```
Node:                                   dr-bob
Queue manager status:                   Ended immediately
HA role:                                Secondary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    dr-alice
HA preferred location:                  dr-alice
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Secondary
DR status:                              See dr-alice
DR port:                                3000
DR local IP address:                    192.168.2.2
DR remote IP address list:              192.168.1.1,192.168.1.2,192.168.1.3
DR current remote IP address:           See dr-alice

Node:                                   dr-alice
HA status:                              Normal

Node:                                   dr-charlie
HA status:                              Normal
```

DR 1 次および HA 1 次であるノードで進行中の DR 同期化の例:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           192.168.2.1
DR synchronization progress:            11.0%
DR estimated time to completion:        2018-09-06 14:55:05

Node:                                   main-bob
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

DR 1 次および HA 1 次であるノードでパーティション化された DR の例:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Partitioned
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           192.168.2.1
DR out of sync data:                    372KB

Node:                                   main-bob
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

V 9.3.0 DR 1 次および HA 1 次であるノードでの DR 非同期の例:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Remote unavailable
DR port:                                3000
DR local IP address:                    192.168.1.1
```

```
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           Unknown
DR out of sync data:                    372KB
DR last in sync:                        2020-02-02 20:22:02

Node:                                   main-bob
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

V 9.3.0 DR 1 次および HA 1 次であるノードでの HA 非同期の例:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           192.168.2.1

Node:                                   main-bob
HA status:                              Inconsistent
HA out of sync data:                    15932KB
HA last in sync:                        2020-02-02 20:22:02

Node:                                   main-charlie
HA status:                              Normal
```

V 9.3.0 以下に、OS カーネル・バージョン (RHEL 7.9) と DRBD カーネル・モジュール (ターゲットは RHEL 7.8) の間の不一致を示す要約状況の例を示します。 DRBD カーネル・モジュールがロードされ、キュー・マネージャーが実行されていることが状況に報告される場合でも、この状況で実行されている OS カーネルをターゲットとするバージョンで DRBD カーネル・モジュールを更新する必要があります。

```
Node:                                   main-alice
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-1127
DRBD version:                           9.1.1
DRBD kernel module status:              Loaded

Queue manager name:                     QM1
Queue manager status:                   Running
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
DR role:                                Primary
```

V 9.3.0 以下に、OS カーネル・バージョン (RHEL 7.9) と DRBD カーネル・モジュール (ターゲットは RHEL 7.6) の間の不一致を示す要約状況の例を示します。 この例では、バージョンの違いがさらに大きいため、DRBD カーネル・モジュールを正常にロードできません。 QM1 は、HA/DR キュー・マネージャーであり、別のノードに移動されます。その HA 状況と DR 状況は不明です。 この障害を解決するには、稼働中の OS カーネルをターゲットとするバージョンで DRBD カーネル・モジュールを更新する必要があります。

```
Node:                                   main-alice
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-957
DRBD version:                           9.1.2+ptf.3
DRBD kernel module status:              Partially loaded

Queue manager name:                     QM1
Queue manager status:                   Running elsewhere
HA status:                              Unknown
```

```
HA current location:                    main-bob
HA preferred location:                  This node
HA blocked location:                    None
DR role:                                Primary
DR status:                              Unknown
```

関連資料

Linux rdqmstatus

Linux *DR/HA* 環境での操作

DR/HA 環境で操作する場合は、高可用性および災害復旧に関して、それぞれ考慮事項があります。

DR/HA RDQM が実行されているノードで障害が発生すると、RDQM は自動的にその HA グループ内の別のノードにフェイルオーバーします。 サイト全体で障害が発生した場合は、復旧サイト上の HA グループ内の優先ノード上で RDQM を手動で開始する必要があります。 ここでの考慮事項は、通常の DR RDQM の場合と同じです。詳しくは、612 ページの『災害復旧環境での操作』を参照してください。

いずれかのノードが完全に故障し、交換する必要がある場合は、ガイダンスとして 614 ページの『災害復旧構成の障害ノードの交換』 および 593 ページの『高可用性の構成の障害ノードの交換』を参照してください。

Linux *DR/HA* 構成の障害ノードの交換

いずれかの HA グループ内のいずれかのノードに障害が発生した場合は、そのノードを置き換えることができます。

このタスクについて

この手順は、置き換えられるノードが DR 構成内の 1 次または 2 次のいずれであるかどうかによって異なります。 いずれの場合も、新規ノードの構成は置き換えるノードと同じである必要があります。つまり、ホスト名、IP アドレスなどが同じである必要があります。

メイン・サイトまたは復旧サイトで HA グループが完全に失われ、HA グループ全体を置き換える必要がある状況が発生する場合もあります。

手順

- DR 構成内の 1 次ノードである交換ノードの場合は、新規ノードで以下の手順を実行します。
  a) その他のノードのファイルと一致する `rdqm.ini` ファイルを作成し、`rdqmadm -c` コマンドを実行します (570 ページの『Pacemaker クラスター (HA グループ) の定義』を参照)。
  b) `crtmqm -sxs -rr p` *qmanager* コマンドを実行して、各 DR/HA RDQM を再作成します (620 ページの『DR/HA RDQM の作成』 を参照)。
- DR 構成の 2 次ノードである交換ノードの場合は、新規ノードで以下の手順を実行します。
  a) その他のノードのファイルと一致する `rdqm.ini` ファイルを作成し、`rdqmadm -c` コマンドを実行します (570 ページの『Pacemaker クラスター (HA グループ) の定義』を参照)。
  b) `crtmqm -sx -rr s` *qmanager* コマンドを実行して、各 DR/HA RDQM を再作成します (620 ページの『DR/HA RDQM の作成』 を参照)。
- HA グループ全体を置き換えるには、以下の手順を実行します。
  a) DR 1 次サイト (メイン・サイト) で HA グループ全体を失った場合は、DR 2 次サイトへの管理フェイルオーバーを実行するための手順に従って、DR/HA RDQM の実行を維持する必要があります (612 ページの『災害復旧環境での操作』を参照)。 (復旧サイトで HA グループ全体が失われると、DR/HA RDQM は介入なしでメイン・サイトで引き続き実行されます。)
  b) 619 ページの『DR/HA RDQM 用の HA グループの構成』で説明されているように、3 つの交換ノード上で HA グループを再作成します。
  c) 620 ページの『DR/HA RDQM の作成』で説明されているように、新規 HA グループ上で DR/HA RDQM を再作成します。
  d) 必要に応じて、復旧サイトからメイン・サイトへの管理フェイルオーバーを実行します。

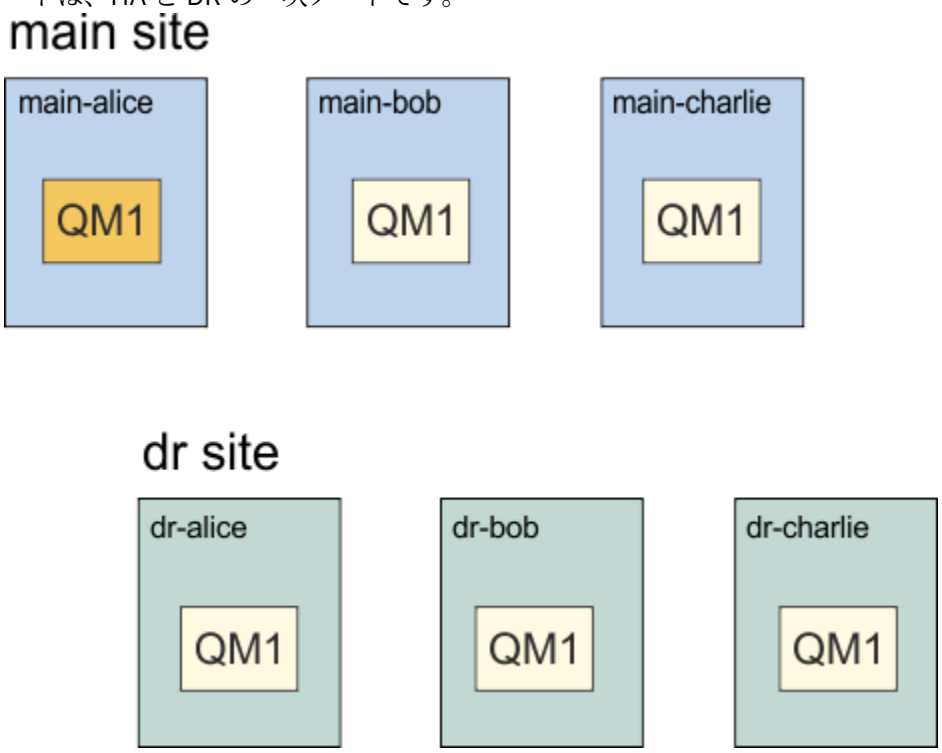
Linux *DR/HA RDQM 作業例*

この例は、DR/HA RDQM を作成および削除する方法を示しています。

DR/HA RDQM の作成

この構成例には、「main」および「dr」という名前の 2 つのサイトが含まれています。 各サイトには、「alice」、「bob」、および「charlie」という名前の 3 つのノードがあります。 これらのノードのフルネームはサイト名と名前で構成され、「main-alice」、「dr-alice」のようになります。

以下の手順では、main-alice で実行される QM1 という名前の DR/HA RDQM を作成します。 main-alice ノードは、HA と DR の一次ノードです。



ローカルおよびリモートの DR IP アドレスが `rdqm.ini` ファイルに指定されている場合は、コマンド行に IP アドレスを指定する必要はなく、QM1 という名前の DR/HA RDQM を作成するには、main-alice 上で以下のコマンドを実行します:

```
crtmqm -sx -rr p -rn DR1 -rp 7001 QM1
```

ローカル DR IP アドレスが `rdqm.ini` ファイルに指定されている場合は、コマンド行でリモート DR IP アドレスを指定することができます:

```
crtmqm -sx -rr p -ri 192.168.2.1,192.168.2.2,192.168.2.3 -rp 7001 QM1
```

`rdqm.ini` ファイルに DR IP アドレスが指定されていない場合は、コマンド行でリモートおよびローカルの DR IP アドレスの両方を指定できます:

```
crtmqm -sx -rr p -rl 192.168.1.1,192.168.1.2,192.168.1.3 -ri
192.168.2.1,192.168.2.2,192.168.2.3 -rp 7001 QM1
```

QM1 の作成に対する応答としての出力は、次の例のようになります。

```
Creating replicated data queue manager configuration.
Secondary queue manager created on 'main-bob'.
Secondary queue manager created on 'main-charlie'.
IBM MQ queue manager created.
Directory '/var/mqm/vols/qm1/qmgr/qm1' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QM1'.
Default objects statistics : 83 created. 0 replaced. 0 failed.
```

```
Completing setup.
Setup completed.
Enabling replicated data queue manager.
Replicated data queue manager enabled.
Issue the following command on the remote HA group to create the DR/HA secondary queue manager:
crtmqm -sx -rr s -rl 192.168.2.1,192.168.2.2,192.168.2.3 -ri
192.168.1.1,192.168.1.2,192.168.1.3 -rp 7001 -fs 3072M QM1
```

次のコマンドをメッセージからコピーして、dr-alice 上に QM1 の DR 2 次インスタンスを作成します。

```
crtmqm -sx -rr s -rl 192.168.2.1,192.168.2.2,192.168.2.3 -ri
192.168.1.1,192.168.1.2,192.168.1.3 -rp 7001 -fs 3072M QM1
```

以下のメッセージが dr-alice で出力されます。

```
Creating replicated data queue manager configuration.
Secondary queue manager created on 'dr-bob'.
Secondary queue manager created on 'dr-charlie'.
IBM MQ secondary queue manager created.
Enabling replicated data queue manager.
```

DR 2 次のテスト

QM1 の災害復旧機能をテストするには、main-alice で以下のコマンドを実行して、QM1 を DR 2 次インスタンスにします。

```
rdqmdr -m QM1 -s
Queue manager 'QM1' has been made the DR secondary on this node.
```

dr-alice で以下のコマンドを実行して、そのノード上の DR 1 次インスタンスを QM1 にします。

```
rdqmdr -m QM1 -p
Queue manager 'QM1' has been made the DR primary on this node.
```

DR/HA RDQM の削除

QM1 という名前の DR/HA RDQM を削除するには、まず main-alice 上でキュー・マネージャーを終了します。

```
endmqm -w QM1
Replicated data queue manager disabled.
Waiting for queue manager 'QM1' to end.
IBM MQ queue manager 'QM1' ended.
```

次に、main-alice で以下のコマンドを実行して、QM1 を削除します。

```
dltmqm QM1
Removing replicated data queue manager configuration.
Secondary queue manager deleted on 'main-bob'.
Secondary queue manager deleted on 'main-charlie'.
IBM MQ queue manager 'QM1' deleted.
```

最後に、dr-alice で QM1 を削除する必要があります。

```
dltmqm QM1
Removing replicated data queue manager configuration.
Secondary queue manager deleted on 'dr-bob'.
Secondary queue manager deleted on 'dr-charlie'.
IBM MQ queue manager 'QM1' deleted.
```

関連概念

635 ページの『DR/HA 環境での操作』
DR/HA 環境で操作する場合は、高可用性および災害復旧に関して、それぞれ考慮事項があります。

関連タスク

620 ページの『DR/HA RDQM の作成』
crtmqm コマンドを使用して、DR/HA 構成内に複製データ・キュー・マネージャー (RDQM) を作成します。

624 ページの『DR/HA RDQM の削除』
dltmqm コマンドを使用して、DR/HA 複製データ・キュー・マネージャー (RDQM) を削除します。

CP4I MQ Adv. ネイティブ HA

ネイティブ HA は、 IBM MQ のコンテナー・デプロイメントで使用可能な高可用性ソリューションです。

ネイティブ HA 構成は、3 つのノード (例えば、3 つの Kubernetes ポッド) で構成され、それぞれにキュー・マネージャーのインスタンスがあります。 1 つのインスタンスはアクティブ・キュー・マネージャーで、メッセージを処理し、そのログに書き込みます。 ログが書き込まれるたびに、アクティブ・キュー・マネージャーはデータを他の 2 つのインスタンス (「レプリカ」と呼ばれる) に送信します。 各レプリカは、独自のログに書き込み、データを確認し、複製されたログから独自のキュー・データを更新します。 アクティブなキュー・マネージャーを実行しているノードに障害が発生した場合、キュー・マネージャーのレプリカ・インスタンスの 1 つがアクティブな役割を引き継ぎ、操作対象の現行データを持ちます。

詳細な概要については、この資料の「コンテナー」セクションの ネイティブ HA を参照してください。

以下の図に、1 つのキュー・マネージャーの 3 つのインスタンスが 3 つのコンテナー内にデプロイされている、標準的なデプロイメントを示します。

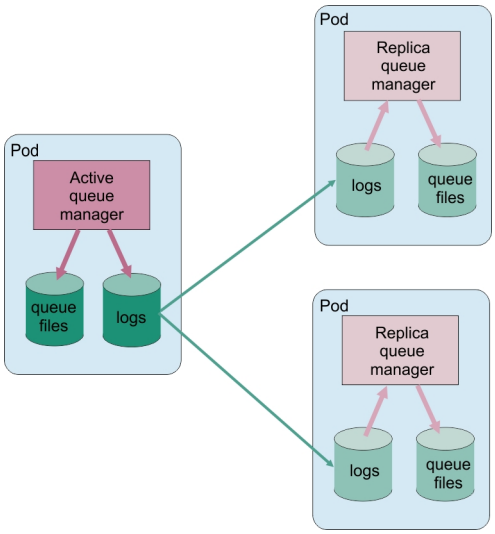


図 *83.* ネイティブ *HA* 構成の例

CP4I MQ Adv. ネイティブ *HA* ソリューションの作成

ネイティブ HA ソリューションを作成するために推奨される方法は、 IBM MQ Operator を使用することです。 あるいは、独自のコンテナーを作成し、ネイティブ HA を手動で構成することもできます。

注 : この情報は、コンテナー環境にのみ適用されます。

IBM MQ Operator を使用してネイティブ HA ソリューションを作成するには、概要について ネイティブ HA を参照し、詳細な手順については 例: ネイティブ HA キュー・マネージャーの構成 を参照してください。

独自のコンテナーを作成し、ネイティブ HA を手動で構成するには、 独自のコンテナーを作成する場合のネイティブ HA グループの作成を参照してください。

CP4I MQ Adv. ネイティブ *HA* キュー・マネージャーの終了

コンテナー内の IBM MQ の場合、 **endmqm** コマンドを使用して、ネイティブ HA グループの一部であるアクティブ・キュー・マネージャーまたはレプリカ・キュー・マネージャーを終了できます。

このタスクについて

注 : この情報は、コンテナー環境にのみ適用されます。

ネイティブ HA グループの一部であるキュー・マネージャーを停止する手順は、そのキュー・マネージャーがアクティブ・インスタンスであるかレプリカ・インスタンスであるかによって異なります。 いずれかの

タイプのインスタンスを終了すると、インスタンスの終了によってネイティブ HA グループのクォーラムが中断されないことを確認するための検査が行われます。 クォーラムが破損すると、 **endmqm** コマンドは失敗します。

endmqm コマンドを発行すると、接続の切断時にエラーが報告されないように、グループ内の他のインスタンスに警告が出されます。

多数のレプリカ・インスタンスの終了または切断が原因でアクティブ・インスタンスがクォーラムを失った場合、アクティブ・インスタンスは、完全に終了する前に、構成可能な時間だけ待機します。 これにより、アプリケーションの接続が切断されるだけではなく、処理を正常にシャットダウンする期間が設けられます。 このタイムアウト値は、 `qm.ini` ファイルの `NativeHALocalInstance` スタンザの `QuorumConnectivityTimeout` 属性で指定できます。 デフォルト値は 0 秒です。

手順

- キュー・マネージャーのアクティブ・インスタンスを終了するには、アクティブ・インスタンスが実行されているノードで次のコマンドを発行します。

  ```
  endmqm -s QMgrName
  ```

  - `-r` オプションを指定して、クライアント・アプリケーションが別のインスタンスに再接続できるようにします。
  - このインスタンスが Native HA グループのアクティブ・インスタンスでない場合、コマンドは失敗します。
  - このアクティブ・インスタンスを終了してグループ・クォーラムが失敗した場合、コマンドは失敗します。 (このコマンドの実行と同時に他のインスタンスが終了するか、または使用できなくなった場合、クォーラム・チェックでこれが検出されない可能性があります。その場合、Native HA グループは終了し、十分な数のインスタンスが使用可能な場合にのみ再起動できます。)

  アクティブなキュー・マネージャーが終了すると、レプリカ・インスタンスの 1 つがアクティブな役割を引き継ぎます。 どのレプリカが引き継ぐかを指定することはできません。これは、グループ内のネゴシエーションによって決定され、どのレプリカが最新のトランザクション・ログを持っているかに依存します。
- キュー・マネージャーのレプリカ・インスタンスを終了するには、次のコマンドを発行します。

  ```
  endmqm -x QMgrName
  ```

  - このインスタンスがアクティブ・インスタンスである場合、コマンドは失敗します。
  - このレプリカ・インスタンスを終了してグループ・クォーラムが失敗した場合、コマンドは失敗します。 (このコマンドの実行と同時に他のインスタンスが終了するか、または使用できなくなった場合、クォーラム・チェックでこれが検出されない可能性があります。その場合、Native HA グループは終了し、十分な数のインスタンスが使用可能な場合にのみ再起動できます。)

  注 : また、ネイティブ HA インスタンスの **endmqm** コマンドで -c、-i、-p、または -w スイッチを使用することもできます。これらのスイッチがどのロールにあるかは関係ありません。 キュー・マネージャー・インスタンスは、グループ・クォーラムに与える影響を無視して終了します。 ただし、情報は引き続きグループ内の他のインスタンスと共有されます。 これらのスイッチは、アクティブ・インスタンスの -s と一緒に使用できます。 レプリカ・インスタンスの場合、これらのスイッチを -x スイッチと一緒に使用することはできません。

関連資料

endmqm (キュー・マネージャーの終了)

ロギング: メッセージが失われないようにするための機能

IBM MQ は、キュー・マネージャーが制御する持続データに対する重要な変更をすべてリカバリー・ログに記録します。

ログには、オブジェクトの作成および削除、持続メッセージの更新、トランザクション状態、オブジェクト属性の変更、およびチャネル活動が含まれます。 ログは、メッセージ・キューのすべての更新をリカバリーする ために必要な情報を、次の方法によって格納します。

- キュー・マネージャーの変更の記録を保持する
- 再始動プロセス用としてキューの更新の記録を保持する
- ハードウェア障害またはソフトウェア障害の後にデータを復元できるようにする

しかし、IBM MQ は、ログ・ファイルを含め、MQ のファイルをホストするディスク・システムにも依存しています。 ディスク・システムそれ自体の信頼性がなければ、 依然としてログ情報などの情報が消失するおそれがあります。

注意 : リカバリー・ログを別のオペレーティング・システムに移動することはできません。

ログの概要

ログは、1 次ファイルと 2 次ファイル、 および制御ファイルで構成されます。 ログ・ファイルの数とサイズ、 およびファイル・システム内の保管場所を定義します。

IBM MQ のログは、次の 2 つのコンポーネントからなっています。

1. 1 つ以上のログ・データ・ファイル。
2. ログ制御ファイル

ログ・データ・ファイルは、ログ・エクステントとも呼ばれます。

記録するデータは、複数のログ・エクステントに含められます。 その数とサイズは、(94 ページの『mqs.ini ファイルの LogDefaults スタンザ』で 説明しているように) 定義することも、システム・デフォルト (1 次エクステントが 3 つで 2 次エクステントが 2 つ) を取ることもできます。

この 3 つの 1 次エクステントと 2 つの 2 次エクステントのデフォルトは 16 MB です。

キュー・マネージャーを作成するときに、割り当てた *1* 次 ログ・エクステントの数が、事前に割り振られるログ・エクステントの数になります。 数を指定しないと、デフォルト値が使用されます。

IBM MQ では、以下の 2 つのタイプのロギングを使用します。

- 循環
- リニア

  リニア・ロギングで使用されるログ・エクステントの数は、メディア・イメージの記録頻度に応じて、非常に大きくなる可能性があります。

詳しくは、 641 ページの『ログのタイプ』 を参照してください。

ALW IBM MQ for AIX or Linux システムでは、ログのパスを変更していなければ、ログ・エクステントは次のディレクトリーに作成されます。

```
/var/mqm/log/QMgrName
```

Windows IBM MQ for Windows では、ログのパスを変更していなければ、ログ・エクステントは次のディレクトリーに作成されます。

```
C:\ProgramData\ IBM \ MQ\log\QMgrName
```

IBM MQ は、これらの 1 次ログ・エクステントで開始しますが、1 次ログ・スペースが足りなくなった場合、*2* 次 ログ・エクステントを割り振ります。 WebSphere MQ はこれを動的に行い、ログ・スペースの要求が少なくなったら、それを除去します。 デフォルトでは、最高 2 個の 2 次ログ・エクステントを割り振ることができます。 このデフォルトの割り振りを、84 ページの『Multiplatforms での .ini ファイルの IBM MQ 構成情報の変更』で説明されている方法で変更することができます。

ログ・エクステントの接頭部は、文字 S または文字 R のいずれかです。 アクティブ、非アクティブ、および不必要なエクステントの接頭部は S、再利用エクステントの接頭部は R となります。

キュー・マネージャーのバックアップまたは復元を行うときは、すべてのアクティブ、非アクティブ、および不必要なエクステントをログ制御ファイルと共にバックアップおよび復元します。

注 : 再利用エクステントをバックアップおよび復元する必要はありません。

ログ制御ファイル

ログ制御ファイルには、サイズや場所、次に使用できるエクステントの名前など、ログ・エクステントの状態を表すのに必要な情報が含まれています。

重要 : ログ制御ファイルは、専らキュー・マネージャーの内部使用向けです。

キュー・マネージャーは、リカバリー・ログの状態に関連付けられた制御データを、ログ制御ファイルに保持します。 ログ制御ファイルの内容を変更しないでください。

ログ制御ファイルは、ログ・パスにあり、ファイル名は `amqhlctl.lfh` です。 キュー・マネージャーのバックアップまたは復元を行うときは、ログ・エクステントと共に、ログ制御ファイルがバックアップおよび復元されることを確認してください。

ログのタイプ

IBM MQ では、キュー・マネージャー・アクティビティーの記録を維持する方法として、循環ロギングとリニア・ロギングの 2 つがあります。 ロギングの 3 番目のタイプである「複製」は、Native HA 構成でのみ使用されます。

循環ロギング

ログを使用してシステムの停止時に処理中であったトランザクションを ロールバックするだけの再始動リカバリーでよい場合には、循環ロギングを使用してください。

循環ロギングでは、すべての再始動データをログ・ファイルのリングに保持 します。 ロギングでは、リングの最初のファイルに出力され、それから次のファイル に進むといった方法で、すべてのファイルに出力されます。 その後、リングの最初のファイルに戻って、再び出力が開始されます。 この操作は、プロダクトが使用されている間ずっと行われるので、ログ・ファ イルが不足するということは決してありません。

キュー・マネージャーをデータの消失なしに再始動するために必要なログ項目は、キュー・マネージャーのデータ・リカバリーの必要性がなくなるまで IBM MQ によって保持されます。 再利用のためにログ・ファイルを解放するためのメカニズムについては、643 ページの『チェックポイント機能を使用してリカバリーの完了を確認する』で説明されています。

リニア・ロギング

再始動リカバリーおよびメディア・リカバリーの両方が必要な場合には、リニア・ロギングを使用します。メディア・リカバリーは、ログの内容を再生することにより、消失または損傷したデータを再作成するものです。 リニア・ロギングでは、ログ・データを連続したログ・ファイルに保持します。

オプションで、ログ・ファイルを以下のようにすることもできます。

- 再利用する。ただし、これはリスタート・リカバリーやメディア・リカバリー用に必要ではなくなったときだけです。
- 手動でアーカイブして、より長い期間保存したり分析したりする。

メディア・イメージの頻度により、リニア・ログ・ファイルの再利用が可能となる時期が決まります。 これは、リニア・ログ・ファイル用に使用可能にする必要のあるディスク・スペースのサイズを判別するための主な要素となります。

時間またはログの使用量に基づいて定期的なメディア・イメージの取得を自動的に行うようにキュー・マネージャーを構成すること、またはメディア・イメージを手動でスケジュールすることができます。

管理者は、実装するポリシーを決定し、ディスク・スペース使用量に与える影響を判断します。 再始動リカバリーに必要なログ・ファイルは常に使用可能でなければなりませんが、メディア・リカバリーだけに必要なログ・ファイルは長期保管用にアーカイブすることが可能です (磁気テープなど)。

管理者が自動ログ管理および自動メディア・イメージを有効にした場合、リニア・ロギングは非常に大きな循環ログと同様の方法で動作しますが、メディア・リカバリーによってメディア障害に対する冗長性は向上します。

IBM MQ 9.1.0 以降、migmqlog コマンドを使用して、キュー・マネージャーの既存のログ・タイプをリニアから循環に、または循環からリニアに変更できます。

複製ロギング

CP4I

Native HA 構成を構成するには複製ロギングを使用します。 Native HA グループを作成するときには、別々のノードに 3 つのキュー・マネージャーを作成します。 複製ロギングを指定する場合は、各キュー・マネージャーの固有のインスタンス名も指定します。 Native HA 構成では、アクティブ・インスタンスの複製ログ・データを 2 つの複製インスタンスに持つことにより、高可用性ソリューションを提供します。 アクティブ・インスタンスに障害が発生すると、レプリカ・インスタンスの 1 つがアクティブ・ロールを引き継ぎます。 ログを複製することで、データ損失が発生したとしても、それを最小限にすることができます。詳しくは、638 ページの『ネイティブ HA』を参照してください。 複製されたログは、自動ログ管理および自動メディア・イメージが有効になっているリニア・ログに相当します。

非アクティブなリニア・ログ・エクステント

Multi

IBM MQ 9.1.0 から、アーカイブ作成を含む自動ログ管理を使用している場合、ロガーは非アクティブなリニア・ログ・エクステントを追跡します。

重要 : アーカイブなしの自動ログ管理を使用している場合、バックアップ・キュー・マネージャーの使用はこのプロセスではサポートされません。

ALW ログ・エクステントがリカバリーに必要なくなったときに、必要に応じてアーカイブされると、ロガーは適切な時点でログ・エクステントを削除するか、またはそのログ・エクステントを再利用します。

再利用されるログ・エクステントは、ログ・シーケンスで次の位置になるように名前変更されます。 メッセージ AMQ7490 が定期的に出力されて、作成、削除、または再利用されたエクステントの数を知らせます。

ロガーは、再利用に備えて保持するエクステントの数を選択し、それらのエクステントを削除するタイミングを選択します。

アクティブ・ログ

リニア・ロギングでも循環ロギングでも、アクティブ と呼ばれるいくつかのファイルがあります。 循環ロギングとリニア・ロギングのどちらを使用するかにかかわらず、アクティブ・ログはログ・スペースの最大量であり、再始動リカバリーで参照される可能性があります。

アクティブ・ログ・ファイルの数は、通常は、構成ファイルに定義された 1 次ログ・ファイルの数より少なくなります。 (数の定義方法の詳細については、646 ページの『ログのサイズの計算』を参照してください。)

アクティブ・ログ・スペースにはメディア・リカバリーに必要なスペースが含まれないこと、またリニア・ロギングで使用されるログ・ファイルの数はメッセージ・フローやメディア・イメージの頻度によっては非常に多くなることに注意してください。

非アクティブ・ログ

ログ・ファイルが再始動リカバリーに必要なくなると、それは非アクティブになります。 再始動リカバリーにもメディア・リカバリーにも必要ないログ・ファイルは、不必要なログ・ファイルと見なすことができます。

自動ログ管理を使用しているとき、それらの不必要なログ・ファイルの処理はキュー・マネージャーによって制御されます。 手動のログ管理を選択した場合、操作の対象でなくなったときに不必要なログ・ファイルを管理 (削除やアーカイブなど) することは、管理者の責任となります。

ログ・ファイルの属性指定の詳細については、652 ページの『ログの管理』を参照してください。

2 次ログ・ファイル

2 次ログ・ファイルはリニア・ロギング用として定義 されますが、このファイルは通常の操作では使用されません。 持続期間の長いトランザクションのために、アクティブ・プールからファイルを解放できないような状態が生じた場合には (そのファイルが再始動に必要とされる可能性がまだ存在するため)、2 次ファイルが フォーマットされて、アクティブ・ログ・ファイル・プールに追加されます。

使用可能な 2 次ファイルの数が使い果たされた場合、ログ活動を必要とするそれ以降のほとんどの操作の要求は拒否されて、MQRC_RESOURCE_PROBLEM 戻りコードがアプリケーションに戻されます。 また、長期実行トランザクションは非同期ロールバックの対象と見なされます。

重要 : すべてのタイプのロギングは不意の停電に対処できます。ただし、ハードウェア障害が起こらないことを想定した場合です。

チェックポイント機能を使用してリカバリーの完了を確認する

循環ロギングとリニア・ロギングの両方のキュー・マネージャーが再始動リカバリーをサポートしています。 直前のキュー・マネージャーのインスタンスがどれほど突然に終了した場合でも (例えば電源異常の場合など)、再始動の際に、キュー・マネージャーはその永続的な状態を終了時点での正しいトランザクション状態に復元します。

再始動リカバリーは、ディスクの保全性が維持されていることに依存します。 同様に、オペレーティング・システムには、オペレーティング・システムがどれほど突然に終了する場合でもディスクの保全性が維持される機能が必要です。

ディスクの保全性が維持されない稀なケースでは、リニア・ロギング (およびメディア・リカバリー) により、追加の冗長性と回復可能性のオプションが提供されます。 普及を続ける RAID などのテクノロジーによって、ディスクの保全性の問題が生じる可能性はますます小さくなっているので、多くの企業では循環ロギングを構成して、再始動リカバリーだけを使用しています。

IBM MQ は、従来型のログ先行書き込みリソース・マネージャーとして設計されています。 メッセージ・キューに対する持続更新は、2 段階で行われます。

1. 更新を表すログ・レコードがリカバリー・ログに確実に書き込まれます。
2. キュー・ファイルやバッファーは、システムにとって最も効率的な方法で更新されますが、一貫して同じ方法であるとは限りません。

そのため、ログ・ファイルの方が、基礎となるキュー・バッファーやファイル状態よりも新しい状態になることがあります。

この状態が収まることなく続くと、クラッシュ・リカバリーの後、キューの状態を整合性のあるものとするために大量のログ再生が必要となります。

IBM MQ は、クラッシュ・リカバリーの後に必要なログ再生の量を制限するために、 `checkpoints` を使用します。 ログ・ファイルをアクティブと呼ぶかどうかを制御するキー・イベントは、 `checkpoint` です。

IBM MQ チェックポイントは、以下のようなポイントです。

- リカバリー・ログとオブジェクト・ファイル間の整合点。
- 後続のログ・レコードの順方向再生により、キュー・マネージャーが終了した可能性のある時点での正しい論理状態にキューが復元されることが保証される、ログ内の場所を示す点。

チェックポイントの際に、IBM MQ は必要に応じて以前の更新内容をキュー・ファイルに移動し、クラッシュ・リカバリーの後にキューを整合性のある状態に戻すために再生しなければならないログ・レコードの量を制限します。

最新の完全なチェックポイントは、クラッシュ・リカバリーの際に再生を実行する開始点とする、ログ内のポイントになります。 したがって、チェックポイントの頻度は、チェックポイントを記録するためのオーバーヘッドと、それらのチェックポイントによる潜在的なリカバリー時間の改善との間のトレードオフによって決まります。

IBM MQ 9.1.0 から、ロガーが設定するチェックポイント・スケジュールの頻度が上がりました (前のチェックポイントの完了前に次のチェックポイントのスケジュールが設定されます)。ロガーは、1 次ログ・エクステントでアクティブ・ログを保持しようとするからです。 これが可能でない場合は、AMQ7466 エラーがログに記録されます。

最新の完全なチェックポイントが開始するログ内の位置は、ログ・ファイルがアクティブか非アクティブかを判別する主な要因の 1 つとなります。 もう 1 つの主な要因は、現行のアクティブ・トランザクションによる最初の持続更新に関連した最初のログ・レコードのログ内での位置です。

新しいチェックポイントが 2 番目以降のログ・ファイルに記録された場合、現在のトランザクションが最初のログ・ファイル内のログ・レコードを参照していなければ、最初のログ・ファイルは非アクティブになります。 循環ロギングの場合、これにより最初のログ・ファイルの再利用が可能になります。 リニア・ロギングの場合、メディア・リカバリーのために通常は最初のログ・ファイルが引き続き必要となります。

循環ロギングまたは自動ログ管理のいずれかを構成した場合、キュー・マネージャーが非アクティブ・ログ・ファイルを管理します。 手動ログ管理によってリニア・ロギングを構成した場合、操作の要件に従って非アクティブ・ファイルを管理することは、管理タスクの 1 つになります。

IBM MQ は、自動的にチェックポイントを生成します。 チェックポイント取得のタイミングは以下のとおりです。

- キュー・マネージャーの開始時
- シャットダウン時
- ログ・スペースの不足時
- Multi 直前のチェックポイントが取得されて以来 50,000 件の操作がログに記録された後
- z/OS 前回のチェックポイント取得時以降、*number_of_operations* 件の操作がログに記録された後 (*number_of_operations* は、**LOGLOAD** プロパティーで設定された操作数です)。

IBM MQ は、再始動時に、ログの中の最新のチェックポイント・レコードを見つけます。 その情報は、すべてのチェックポイントの最後に更新される チェックポイント・ファイルに保持されています。 チェックポイント以降に発生した操作すべてが、順方向に再生されます。 これを、やり直しフェーズと呼びます。

やり直しフェーズによって、キューがシステム障害やシャットダウンの前の論理的状態に戻ります。 やり直しフェーズの間に、システム障害やシャットダウンが発生したときに未完了であったトランザクションのリストが作成されます。

Multi やり直しフェーズの経過を示すために、メッセージ AMQ7229 and AMQ7230 が出されます。

IBM MQ は、バックアウトまたはコミットする操作を知るために、不完了トランザクションに関連付けられた各アクティブ・ログ・レコードにアクセスします。 これを、リカバリー・フェーズと呼びます。

Multi リカバリー・フェーズの経過を示すために、メッセージ AMQ7231、AMQ7232、および AMQ7234 が出されます。

リカバリー・フェーズの間に必要なすべてのログ・レコードにアクセスした後、次に各アクティブ・トランザクションが解決され、トランザクションに関連付けられた各操作がバックアウトまたはコミットされます。 これを、解決フェーズと呼びます。

Multi 解決フェーズの経過を示すために、メッセージ AMQ7233 が出されます。

z/OS z/OS では、再始動処理はさまざまなフェーズで構成されます。

1. ページ・セットに必要なメディア・リカバリー、および作業単位をバックアウトするためと未確定の作業単位のロックを取得するために必要な最も古いログ・レコードに基づいて、リカバリー・ログの範囲が確立されます。

2. ログ範囲が決定されると、順方向のログの読み取りが行われて、ページ・セットが最新の状態になり、未確定または未完了の作業単位に関連したすべてのメッセージがロックされます。
3. 順方向のログ読み取りが完了すると、ログは逆方向に読み取られて、障害発生時に未確定または未完了だった作業単位がバックアウトされます。

z/OS 表示される可能性のあるメッセージの例を以下に示します。

```
CSQR001I +MQOX RESTART INITIATED
CSQR003I +MQOX RESTART - PRIOR CHECKPOINT RBA=00000001E48C0A5E
CSQR004I +MQOX RESTART - UR COUNTS -  806
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN BACKOUT=0
CSQR030I +MQOX Forward recovery log range  815
from RBA=00000001E45FF7AD to RBA=00000001E48C1882
CSQR005I +MQOX RESTART - FORWARD RECOVERY COMPLETE -  816
IN COMMIT=0, INDOUBT=0
CSQR032I +MQOX Backward recovery log range  817
from RBA=00000001E48C1882 to RBA=00000001E48C1882
CSQR006I +MQOX RESTART - BACKWARD RECOVERY COMPLETE -  818
INFLIGHT=0, IN BACKOUT=0
CSQR002I +MQOX RESTART COMPLETED
```

注 : 読み取るログが大量にある場合は、メッセージ CSQR031I (順方向リカバリー) および CSQR033I (逆方向リカバリー) が定期的に発行されて、進捗が示されます。

645 ページの図 84 では、最新のチェックポイント (チェックポイント 2) より前のすべてのレコードは、IBM MQ にはもう必要ありません。 このチェックポイント情報およびその後のログ項目から、キューをリカバリーするこ とができます。 循環ロギングでは、チェックポイントより前に解放されたファイルは再利用できます。 リニア・ロギングでは、解放されたログ・ファイルは、通常の操作では アクセスする必要がなくなり、非アクティブになります。 この例では、キュー・ヘッド・ポインターが最新のチェックポイントであるチェックポイント 2 をポイントするように移動されます。Checkpoint 2 は、新しいキュー・ヘッド、ヘッド 2 になります。 ログ・ファイル 1 を再使用できるようになりました。

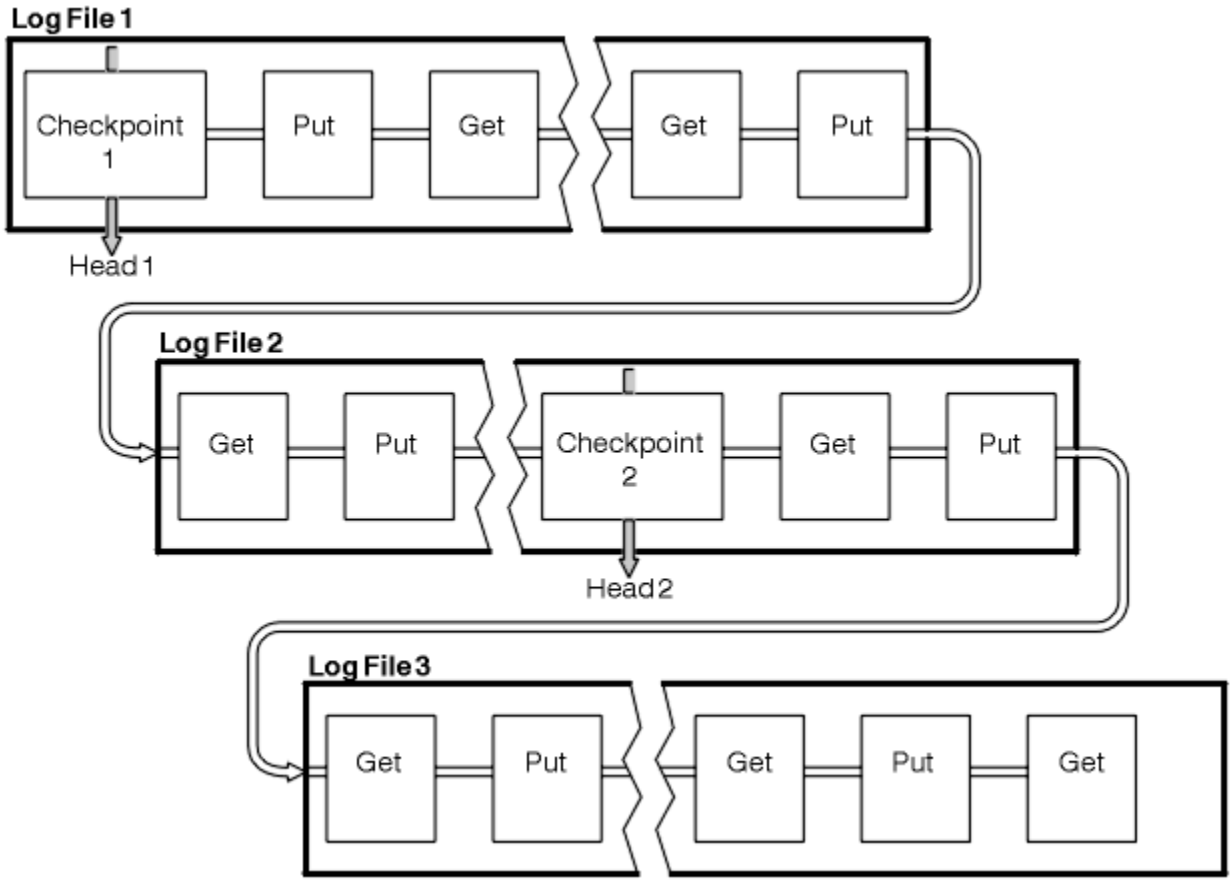


図 *84.* チェックポイント機能

持続期間の長いトランザクションを使用したチェックポイント機能

持続期間の長いトランザクションが、ログ・ファイルの再利用にどんな影響を与えるかを示したものです。

646 ページの図 85 は、持続期間の長いトランザクションが、ログ・ファイルの再利用にどんな影響を与えるかを示したものです。 例では、持続期間の長いトランザクションによって、示されている最初の チェックポイントの後に LR 1 で示されているログへの入力が開始されました。 3 番目のチェックポイントの後までトランザクションは完了しません (LR 2 の地点)。 LR 1 以降のすべてのログ情報は、このトランザクションのリカバリーを行うために、必要であればトランザクションが完了するまで保持されます。

持続期間の長いトランザクションが完了した後に (LR 2)、ログのヘッドがチェックポイント 3 (ログに記録された最も新しいチェックポイント) に論理的に移されます。 チェックポイント 3 (ヘッド 2) より前のログ・レコードの入ったファイルは、不要になります。 循環ロギングを使用している場合には、スペースは再利用できます。

持続期間の長いトランザクションが完了する前に 1 次ログ・ファイルがすべて満杯になった場合は、ログフルを防止するために、2 次ログ・ファイルが使用される可能性があります。

完全にキュー・マネージャーの制御下にあるアクティビティー (チェックポイントなど) については、そのアクティビティーを 1 次ログ内に保持する試行が行われるようにスケジュールされています。

ただし、キュー・マネージャーの制御下にない動作 (いずれかのトランザクションの期間など) をサポートするために 2 次ログ・スペースが必要なとき、そのアクティビティーが完了するように、キュー・マネージャーは定義されたいずれかの 2 次ログ・スペースの使用を試みます。

合計ログ・スペースの 80% を使用してもそのアクティビティーが完了しない場合、キュー・マネージャーは、それがアプリケーションに影響を与えることになるとしても、ログ・スペースを再要求するアクションを開始します。

循環ロギングを使用している場合に、ログのヘッドが移されたときは、1 次ログ・ファイルが再利用できます。ロガーは、カレント・ファイル が満杯になった後に、使用できる最初の 1 次ファイルを再利用します。リニア・ロギングを使用している場合は、ログのヘッドはアクティブ・プールの さらに後方に移されて、最初のファイルが非アクティブになります。 新しい 1 次ファイルがフォーマットされて、将来のロギング活動に備えて プールの下部に追加されます。

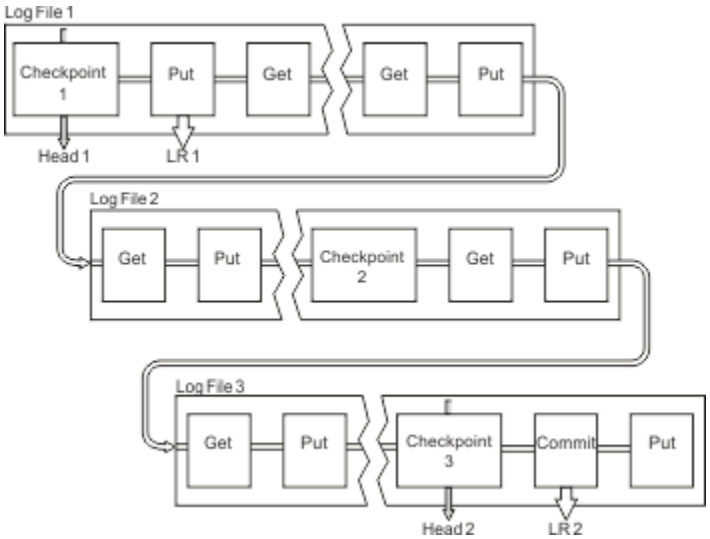


図 *85.* 持続期間の長いトランザクションでのチェックポイント機能

ログのサイズの計算

キュー・マネージャーが必要とするログのサイズを見積もります。

キュー・マネージャーで循環ロギングとリニア・ロギングのどちらを使用するかを決めた後、キュー・マネージャーに必要なアクティブ・ログのサイズを見積もる必要があります。 アクティブ・ログのサイズは、次のログ構成パラメーターによって決まります。

LogFilePages
4K ページを 1 単位として表した、個々の 1 次ログ・ファイルまたは 2 次ログ・ファイルのサイズ

LogPrimaryFiles
事前に割り振る 1 次ログ・ファイルの数

LogSecondaryFiles
1 次ログ・ファイルが満杯になってきたときに使用するために作成可能な 2 次ログ・ファイルの数

注 :

1. 1 次ログ・ファイルと 2 次ログ・ファイルの数は、キュー・マネージャーを始動するたびに変更することができます。ただし、2 次ログに対して行った変更の効果はすぐにはわからない可能性があります。
2. ログ・ファイル・サイズは変更できません。キュー・マネージャーの作成 **前**に決定する必要があります。
3. 1 次ログ・ファイルの数およびログ・ファイル・サイズによって、キュー・マネージャーの作成時に事前に割り振るログ・スペースの量が決まります。
4. 1 次ログ・ファイルと 2 次ログ・ファイルの合計数は、AIX and Linux システムでは 511 以下、Windows では 255 以下にする必要があります。このため、長時間続くトランザクションでは、キュー・マネージャーが再始動リカバリーに使用できるログ・スペースの最大量が制限されます。 メディア・リカバリーのためにキュー・マネージャーが使用するログ・スペースの量は、この制限には拘束されません。
5. 循環 ロギングを使用している場合は、キュー・マネージャーは 1 次および 2 次ログ・スペースを再利用します。 ログ・ファイルが満杯になり、シーケンス内の 次の 1 次ログ・ファイルが使用できない場合には、キュー・マネージャーは 2 次ログ・ファイルを割り振り、限度に達するまでこれを続けます。

   割り振る必要があるログの数については、647 ページの『アクティブ・ログの大きさの決定方法』を参照してください。 1 次ログのエクステントは順番に使用され、その順番は変わりません。

   例えば、3 つの 1 次ログ 0、1、および 2 があるとすると、0、1、2 の順に使用され、続いて 1、2、0、2、0、1 が使用され、0、1、2 に戻ります。 割り振った 2 次ログが、必要に応じてこの間に使用されます。
6. チェックポイントの時点で、1 次ログ・ファイルは再利用できるようになります。 ログ・スペースの量は徐々に小さくなるので、キュー・マネージャーは、1 次ログ・スペースおよび 2 次ログ・スペースの両方を考慮に入れてチェックポイントをとります。

   キュー・マネージャーは、ログの使用が 1 次エクステント内に保持されるようにチェックポイントをスケジュールすることを試みます。

詳しくは、94 ページの『mqs.ini ファイルの LogDefaults スタンザ』を参照してください。

アクティブ・ログの大きさの決定方法

キュー・マネージャーが必要とするアクティブ・ログのサイズを見積もります。

アクティブ・ログのサイズの上限を次に示します。

```
logsize = (primaryfiles + secondaryfiles) * logfilepages * 4096
```

ログは、キュー・マネージャーがディスクに書き込む 1 秒あたりのデータ量が最大の状態で、実行時間が最長のトランザクションを処理できる十分な大きさにする必要があります。

実行時間が最長のトランザクションの実行時間が N 秒で、キュー・マネージャーがディスクに書き込む 1 秒あたりの最大データ量がログで 1 秒あたり B バイトである場合、ログを次のサイズ以上にする必要があります。

```
logsize >= 2 * (N+1) * B
```

キュー・マネージャーがディスクに書き込む 1 秒あたりのデータ量が最大になる可能性があるのは、ワークロードのピーク時に実行する場合、あるいはメディア・イメージを記録する場合です。

トランザクションの実行時間が長いために、最初のログ・レコードが入ったログ・エクステントがアクティブ・ログ内に含まれなくなると、キュー・マネージャーはアクティブ・ログを一度に 1 つずつ、ログ・レコードが最も古いものからロール・バックします。

1 次ファイルおよび 2 次ファイルの最大数が使用される前に、キュー・マネージャーは、古いログ・エクステントを非アクティブにして、他のログ・エクステントを割り振る必要があります。

実行時間が最長のトランザクションを実行する時間を決定します (その時間を超えると、キュー・マネージャーはそのトランザクションをロールバックできます)。 実行時間が最長のトランザクションは低速なネットワークのトラフィックを待機している可能性があります。あるいは、トランザクションの設計が不十分でユーザー入力を待機している可能性があります。

実行時間が最長のトランザクションの実行時間を調べるには、次の **runmqsc** コマンドを実行します。

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

dspmqtrn -a コマンドを実行すると、すべての状態のすべての XA および非 XA コマンドが表示されます。

このコマンドを実行すると、すべての現行トランザクションについて最初のログ・レコードが書き込まれた日時がリストされます。

重要 : ログ・サイズを計算するという目的で重要なのは、アプリケーション/トランザクションが開始されてから経過した時間ではなく、最初のログ・レコードが書き込まれてから経過した時間です。実行時間が最長のトランザクションの実行時間を最も近い秒に切り上げます。 これは、キュー・マネージャーの最適化のためです。

例えば、MQGET 呼び出しの発行によって開始されるアプリケーションの場合は、実際にメッセージを取得するまでに一定の時間待機するので、最初のログ・レコードが書き込まれるのはアプリケーションが開始されてからかなり後になります。

以下からの最大監視日時出力を調べます。

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

このコマンドは、最初に発行した、現在の日時からのものです。これにより、実行時間が最長のトランザクションの実行時間を推定できます。

実行時間が最長のトランザクションの実行時間を過小評価しないように、そのトランザクションをワークロードのピーク時に実行し、その間にこの **runmqsc** コマンドを繰り返し実行してください。

IBM MQ 8.0 では、オペレーティング・システム・ツール (例えば、 UNIX プラットフォーム上の **iostat**) を使用します。

IBM MQ 9.0 以降では、次のコマンドを実行して、キュー・マネージャーがログに書き込んだ 1 秒あたりのバイト数を調べることができます。

```
amqsrua -m qmgr -c DISK -t Log
```

「logical bytes written」が、キュー・マネージャーがログに書き込んだ 1 秒あたりのバイト数を示しています。 以下に例を示します。

```
$ amqsrua -m mark -c DISK -t Log
Publication received PutDate:20160920 PutTime:15383157 Interval:4 minutes,39.579 seconds
Log - bytes in use 37748736
Log - bytes max 50331648
Log file system - bytes in use 316243968
Log file system - bytes max 5368709120
Log - physical bytes written 4334030848 15501948/sec
Log - logical bytes written 3567624710 12760669/sec
Log - write latency 411 uSec
```

この例では、1 秒あたりにログに書き込まれた論理バイト数は 12760669/sec (約 12 MiB/秒) です。

次を使用して

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

実行時間が最長のトランザクションが以下であることがわかります。

```
CONN(57E14F6820700069)
```

```
EXTCONN(414D51436D61726B2020202020202020)
TYPE(CONN)
APPLTAG(msginteg_r)                     UOWLOGDA(2016-09-20)
UOWLOGTI(16.44.14)
```

現在の日時が 2016-09-20 16.44.19 なので、このトランザクションは 5 秒間実行されていたことになります。 しかし、キュー・マネージャーでロールバックするまでにトランザクションを 10 秒間実行することを許容することにします。 したがって、必要なログ・サイズは次のようになります。

```
2 * (10 + 1) * 12 = 264 MiB
```

.

ログ・ファイルの数は、(前のテキストで計算された) 予期される最大サイズのログを格納できるものでなければなりません。 これは次のようになります。

ログ・ファイルの最小数 = (必要なログ・サイズ) / (**LogFilePages** * ログ・ファイルのページ・サイズ (4096))

LogFilePages のデフォルト値 4096 と、前のテキストで計算されたログ・サイズの見積もり 264MiB を使用すると、ログ・ファイルの最小数は次のようになります。

```
264MiB / (4096 x 4096) = 16.5
```

つまり、17 個のログ・ファイルです。

予想されるワークロードの実行が 1 次ファイル内に収まるようにログのサイズを設定する場合

- 2 次ファイルは、追加のログ・スペースが必要になった場合のための、緊急用になります。
- 事前割り振りされた 1 次ファイルを常に使用する循環ロギングは、2 次ファイルの割り振りと解放を行うよりわずかに速くなります。
- キュー・マネージャーは 1 次ファイルに残されているスペースのみを使用して、次のチェックポイントを取る時点を計算します。

したがって、上記の例では、ワークロードが 1 次ログ・ファイルの中で実行するように、次の値を設定します。

- **LogFilePages** = 4096
- **LogPrimaryFiles** = 17
- **LogSecondaryFiles** = 5

次の事項に注意してください。

- この例では、5 つの 2 次ファイルがアクティブ・ログ・スペースの 20 パーセントを超えています。

  IBM MQ 9.1.0 から、ロガーはワークロードを 1 次ファイルの中だけに保管するように試行します。 そのため、ロガーは、1 次ファイルの一部だけがフルになるとチェックポイントをスケジュールします。

  2 次ファイルは、予期せぬ長期実行トランザクションが生じた場合に備えて、緊急用に用意します。

  合計ログ・スペースの 80 パーセントより多くが使用されると、キュー・マネージャーはログ・スペースの使用を減らすための処理を行うことに注意してください。
- リニア・ロギングまたは循環ロギングのどちらを使用する場合も同じ計算を行います。

  アクティブ・ログの概念はリニア・ロギングでも循環ロギングでも同じであるため、リニア・アクティブ・ログと循環アクティブ・ログのどちらを計算しているかは問題ではありません。
- メディア・リカバリーだけに必要なログ・エクステントはアクティブ・ログ内にないため、1 次ファイルおよび 2 次ファイルの数にはカウントされません。
- IBM MQ 9.1.0 以降、 DISPLAY QMSTATUS LOG の *LOGUTIL* フィールドを使用して、必要なアクティブ・ログのサイズを概算することができます。

  このフィールドは、長期実行トランザクションの期間、またはキュー・マネージャーのスループットのピークを判別するために、常にサンプリングすることなく、必要なログ・サイズの妥当な見積もりを作成できるように設計されています。

LogFilePages の大きさの決定方法

通常は、LogFilePages を十分な大きさにしてください。そうすれば、1 次ファイルの最大数に達しないようにアクティブ・ログのサイズを簡単に大きくできます。 小さなログ・ファイルがたくさんあるよりも、大きなログ・ファイルが少ない数だけある方が望ましいと言えます。その方が、必要に応じてログのサイズを大きくする作業を柔軟に行えるからです。

リニア・ロギングの場合は、ログ・ファイルが非常に大きくなると、パフォーマンスが変わりやすくなります。 ログ・ファイルが非常に大きくなると、新しいログ・ファイルを作成して形式を設定したり古いログ・ファイルをアーカイブに保存したりする手順が大掛かりになります。 この問題が大きくなるのは、手動ログ管理やアーカイブ・ログ管理の場合です。自動ログ管理では、新しいログ・ファイルが作成されることはまれだからです。

ログを小さくしすぎた場合の結果

ログの最小サイズを見積もる際に考慮すべきポイントについて説明します。

ログを小さくしすぎた場合:

- 実行時間の長いトランザクションがバックアウトされます。
- 前のチェックポイントが終了する前に次のチェックポイントが開始しようとします。

重要 : ログ・サイズの見積もりが不正確でも、データの保全性は保たれます。

チェックポイントの説明については、643 ページの『チェックポイント機能を使用してリカバリーの完了を確認する』を参照してください。 アクティブ・ログ・エクステントに残されたログ・スペース量が不足してくると、キュー・マネージャーがチェックポイントをスケジュールする頻度が高くなります。

チェックポイントは一瞬ではなく一定の時間がかかります。 チェックポイントで記録しなければならないデータが多くなるほど、チェックポイントにかかる時間は長くなります。 ログが小さいとチェックポイントがオーバーラップする可能性があります。つまり、前のチェックポイントが終了する前に次のチェックポイントが要求されます。 これが発生すると、エラー・メッセージが書き込まれます。

実行時間の長いトランザクションがバックアウトされたり、チェックポイントがオーバーラップしたりしても、キュー・マネージャーはワークロードの処理を続けます。 実行時間の短いトランザクションは通常どおり実行できます。

しかし、キュー・マネージャーの実行状況は好ましいものではなく、パフォーマンスが低下する可能性もあります。 十分なログ・スペースを割り当ててキュー・マネージャーを再始動する必要があります。

ログを大きくしすぎた場合の結果

ログの最大サイズを見積もる際に考慮すべきポイントについて説明します。

ログを大きくしすぎた場合:

- めったにありませんが、緊急時再始動にかかる時間が長くなる可能性があります。
- 不要なディスク・スペースを使用することになります。
- 実行時間の非常に長いトランザクションが許容されます。

重要 : ログ・サイズの見積もりが不正確でも、データの保全性は保たれます。

ログ使用率統計を使用すると、ログの最大サイズを見積もるために役立ちます。 詳しくは、658 ページの『IMGLOGLN と IMGINTVL の設定値の決定』 および ALTER QMGR を参照してください。

キュー・マネージャーが再始動時にログを読み取る方法について詳しくは、643 ページの『チェックポイント機能を使用してリカバリーの完了を確認する』を参照してください。 キュー・マネージャーは最後のチェックポイントからログを再生し、キュー・マネージャーの終了時にアクティブだったすべてのトランザクションを解決します。

トランザクションを解決するために、キュー・マネージャーはそのトランザクションに関連するすべてのログ・レコードを読み直します。 それらのログ・レコードが最後のチェックポイントより前のものである場合もあります。

キュー・マネージャーに非常に大きなログを割り振れば、キュー・マネージャーは再始動時にログ内のすべてのログ・レコードを読み取ることができますが、通常、これが必要になることはありません。 これが行われるような状況が万が一発生した場合、そのプロセスには非常に長い時間がかかります。

チェックポイントが予期せず停止した後にキュー・マネージャーが終了した場合、ログが大きいと、キュー・マネージャーの再始動時間が大幅に増加します。 ログ・サイズを制限することで、緊急時再始動にかかる時間を制限できます。

このような問題を回避するには、次のようにする必要があります。

- ワークロードが十分に収まるが、大きすぎないログにする。
- 実行時間の長いトランザクションは回避する。

ログ・ファイル・システムの大きさの決定方法

キュー・マネージャーが必要とするログ・ファイル・システムのサイズの見積もり。

ログ・ファイル・システムを大きくして、キュー・マネージャーがログ記録を書き込むためのスペースが十分にあるようにすることは大切です。 ログ・ファイル・システムがキュー・マネージャーによって完全にフルにされると、FFDC が書き出され、トランザクションはロールバックして、 キュー・マネージャーが突然終了することがあります。

ログのために予約するディスク・スペースの量は、少なくともアクティブ・ログと同じ大きさでなければなりません。 その正確な大きさは、以下の要素に依存します。

- ログ・タイプの選択 (リニアまたは循環)
- アクティブ・ログのサイズ (1 次ファイル、2 次ファイル、ログ・ファイル・ページ)
- ログ管理の選択 (手動、自動、またはアーカイブ)
- オブジェクトが損傷した場合のコンティンジェンシー・プラン

循環ログを選択した場合、ログ・ファイル・システムには次のサイズが必要です。

```
LogFilesystemSize >= (PrimaryFiles + SecondaryFiles + 1) * LogFileSize
```

これにより、キュー・マネージャーはすべての 1 次ファイルと 2 次ファイルに書き込むことができます。 例外的な状況では、キュー・マネージャーが 2 次ファイルの数を超えて追加のエクステントを書き込むことがあります。 前述のアルゴリズムは、それを考慮に入れたものです。

リニア・ログを選択した場合、ログ・ファイル・システムはアクティブ・ログよりもかなり大きくなります。

手動ログ管理を選択すると、キュー・マネージャーは、必要に応じて新しいログ・エクステントへの書き込みを続行します。 ユーザーは、必要なくなったときにそれらを削除 (およびアーカイブ) する責任があります。

ログ・ファイル・システムに必要な大きさは、不必要なエクステントや非アクティブのエクステントを削除する戦略に大きく依存しています。

非アクティブになったエクステント (再始動リカバリーには不要) をすぐにアーカイブして削除するように決めること、または不必要なエクステント (メディア・リカバリーや再始動リカバリーには不要) だけをアーカイブして削除するように決めることができます。

不必要なエクステントだけをアーカイブおよび削除する場合、損傷したオブジェクトがあれば、**MEDIALOG** は進行しません。 そのため、さらに多くのエクステントが不必要になることはありません。 オブジェクトをリカバリーするなどして、問題を解決するまでは、エクステントのアーカイブと削除を停止します。

ワークロードを停止しない場合、問題を解決するために使用できる時間は、ログ・ファイル・システムのサイズによって決まります。 そのため、ベスト・プラクティスとして、リニア・ロギングを使用する場合には十分に大きなログ・ファイル・システムを用意してください。

リニア・ログと、自動またはアーカイブ・ログ管理を選択した場合、キュー・マネージャーはログ・エクステントを再利用します。

再使用できるログ・エクステントには、接頭部として R という文字が付けられます。 メディア・イメージが記録される時に、余分なエクステントがアーカイブされるので、キュー・マネージャーはそれらのエクステントを再利用することができます。

そのため、再使用エクステントは、メディア・イメージの間のログに書き込まれるデータの長さ未満になります。

```
ReuseExtents <= LogDataLengthBetweenMediaImages
```

メディア・イメージを自動的に記録して **IMGLOGLN** を設定するとき、**IMGLOGLN** は固定された最大値ではなく目標値なので、LogDataLengthBetweenMediaImages は **IMGLOGLN** の 2 倍の大きさになることもあります。

メディア・イメージを手動で記録するとき、またはインターバルに基づいて自動的に記録するとき、LogDataLengthBetweenMediaImages は、ワークロードおよび複数のイメージを取得する間隔によって決まります。

アクティブ・エクステントおよび再利用エクステントに加えて、非アクティブ・エクステント (メディア・リカバリーにのみ必要) および不必要なエクステント (再始動リカバリーやメディア・リカバリーには不要) もあります。

自動またはアーカイブ・ログ管理を使用する場合、キュー・マネージャーは、メディア・リカバリーに必要なエクステントを再利用しません。 したがって、非アクティブ・エクステントの数は、メディア・イメージを取得する頻度、およびそれらを手動または自動のどちらで取得するかによって異なります。

IMGINTVL および **IMGLOGLN** は目標値であり、メディア・イメージ間の固定された最小値や最大値ではありません。 ただし、必要となる可能性のあるログ・ファイル・システムの最大サイズを見積もるとき、自動メディア・イメージが、**IMGINTVL** または **IMGLOGLN** の 2 倍よりも大きく記録されることはまずありません。

自動またはアーカイブ・ログ管理を使用してログ・ファイル・システムのサイズを決める際には、キューまたは他のオブジェクトに損傷が生じた場合にどうなるかをも検討する必要があります。 その場合、キュー・マネージャーは損傷したオブジェクトのメディア・イメージを取得できないので、**MEDIALOG** は進行しません。

ワークロードが続く場合、メディア・リカバリーに必要な最も古いエクステントは引き続き必要となり、再利用できないので、非アクティブ・ログが無制限に拡大します。 ワークロードが続く場合には、ログ・ファイル・システムが完全にフルになり、キュー・マネージャーがトランザクションのロールバックを開始して、突然停止する可能性さえ生じる前に、問題を解決する必要があります。

そのため、自動ログ管理とアーカイブ・ログ管理:

```
LogFilesystemSize > (PrimaryFiles + SecondaryFiles +
(((TimeBetweenMediaImages *2) + TimeNeededToResolveDamagedObject) * ExtentsUsedPerHour))
* LogFilePages
```

注 : 上記のアルゴリズムでは、アーカイブ・ログ管理のために、 **SET LOG ARCHIVED** がエクステントごとに、メディア・リカバリーに必要でなくなるとすぐに呼び出されることを前提としています。

ログの管理

IBM MQ 9.1.0 から、本製品は自動ログ管理およびリニア・ログの自動メディア・リカバリーをサポートします。 循環ログは、ほぼ自己管理されますが、スペースの問題を解決するために介入が必要になることがあります。

注 : IBM i 自動ログ管理およびアーカイブ・ログ管理は、IBM i では無効です。

循環ロギングでは、キュー・マネージャーはログ・ファイルの空きスペースを再利用します。 この活動はユーザーには気付かれないで行われます。割り振られるスペースは迅速に再利用されるため、使用されるディスク・スペースの量が減少するのは通常は分かりません。

IBM MQ 9.1.0 から、循環ロギングを使用するときに 2 次ファイルを削除できます。 詳しくは、RESET QMGR **TYPE(REDUCELOG)** を参照してください。

リニア・ロギングでは、長い間チェックポイントが取られていない場合や、持続期間の長いトランザクションがはるか前にログ・レコードを作成した場合に、ログがフルになる可能性があります。 キュー・マネージャーは、最初の問題を回避するために、頻繁にチェックポイントを取ろうとします。

Multi ログが満杯になると、メッセージ AMQ7463 が出されます。 さらに、持続期間の長いトランザクションがスペースの解放を妨げているためにログが満杯になった場合、メッセージ AMQ7465 が出されます。

ログ・レコードのうち、最後の完全なチェックポイントの開始以降に 書き込まれたもの、およびアクティブ・トランザクションによって 書き込まれたもののみが、キュー・マネージャーの再始動に必要です。

時間の経過と共に、作成された最も古いログ・レコードはキュー・マネージャーの再始動には不要になります。

持続期間の長いトランザクションが検出されると、そのトランザクションを非同期でロールバックするようにアクティビティーがスケジュールされます。 何らかの予期しない理由により、その非同期ロールバックが失敗した場合、一部の MQI 呼び出しはその状況で MQRC_RESOURCE_PROBLEM を戻します。

MQCMIT や **MQBACK** が失敗しないように、すべての未完了トランザクションをコミットまたはロールバックするためのスペースが予約されていることに注意してください。

このようにしてトランザクションをロールバックしたアプリケーションは、同じトランザクションの下で同期点を指定して後続の **MQPUT** または **MQGET** 操作を実行することはできません。

この状態の同期点でメッセージの書き込みまたは読み取りを行おうとすると、MQRC_BACKED_OUT が戻されます。 この場合、アプリケーションは、**MQCMIT** を出す (MQRC_BACKED_OUT が戻される) か、**MQBACK** を出して、新しいトランザクションを開始することができます。 過度にログ・スペースを消費するトランザクションがロールバックされると、そのログ・スペースが解放され、キュー・マネージャーは操作を正常に続行します。

自動メディアイメージング

リニアロギングを使用している場合、メディア画像を定期的に撮影することができます。 これらのイメージは、オブジェクトが損傷した場合に回復することができます。

デフォルトでは、メディアイメージングのスケジューリングは手動で行われます。 メディアイメージングを自動に設定することもできます。 自動メディアイメージングを設定した場合、キューマネージャは作業負荷が許す限りイメージングをスケジュールします。 管理が簡素化され、パフォーマンスへの影響も最小限に抑えられます。

ALTER QMGRMQSC コマンドを使用して以下のキューマネージャ属性を設定することで、 自動メディアイメージングを有効にし、キューマネージャがそれをどのように 実装するかを定義することができる：

IMGSCHED
キューマネージャの自動メディアイメージングを有効にするには、この属性を 'AUTO に設定します。 自動メディアイメージングを有効にしている場合は、この 2 つの属性を使用して、画像を撮影するタイミングを調整します：

IMGINTVL
メディア画像を記録するターゲット周波数を指定する。 分単位で値を指定する。デフォルトは 60 分。 メディア画像に影響を与える時間間隔を使用しない場合は、「OFF」に設定する。

IMGLOGLN
メディア画像が撮影されるまでに書き込まれるログの目標量を指定する。 OFF」に設定すると、メディアイメージングに影響を与えるログの書き込み量を使用しない。 デフォルトは OFF です。

IMGRCOVO
この属性を使用して、回復可能なオブジェクトのタイプを指定します。 認証情報、チャネル、クライアント接続、リスナー、ネームリスト、プロセス、エイリアスキュー、リモートキュー、およびサービスオブジェクトのメディアイメージを取得するには、この属性を'YES に設定します。

IMGRCOVQ
この属性を使用して、ローカルキューと PERMADYN キュータイプのデフォルトを設定します。 メディア画像を撮影する場合は'YES、そうでない場合は'NO を設定する。

関連情報
ALTER QMGR (キュー・マネージャー設定の変更)

自動ログ管理
ログエクステントを再利用できるように、リニアログの自動ログ管理を設定できます。

デフォルトでは、リニアログは手動で管理しなければならない。つまり、ディスク容量が不足しないように、リカバリに不要になったログファイルを削除またはアーカイブしなければならない。

自動ログ管理を有効にすると、ログ・エクステントが自動的にリサイクルされるため、容量が不足することはありません。

キューマネージャの作成時にロギング管理の設定を指定することもできますし、'qm.ini ファイルを編集して既存のキューマネージャのロギング管理を変更することもできます。 3 つのログ管理モードから 1 つを選択できる：

手動
ログ・エクステントは手動で管理する。 これはデフォルト設定です。

自動
キューマネージャは、再起動やメディアリカバリのためにログエクステントが不要になると、自動的に再利用します

アーカイブ
これは、キュー・マネージャが、エクステントを再利用する前に、エクステントがアーカイブされたという通知を待つことを除いて、Automatic と同じです。

キュー・マネージャを作成する際には、'**crtmqm** コマンドで以下の引数を使用することで、ロギング管理設定を指定します：

-lla
自動ログ管理を使用することを指定する。

-lln
アーカイブログ管理を使用することを指定する。

-ll
手動ログ管理を使用することを指定する（デフォルト）。

既存のキューマネージャについては、キューマネージャの 'qm.ini ファイルの 'Log スタンザを編集することで、ログ管理設定を変更することができます。 以下のいずれかの設定を追加する：

LogManagement=Automatic
自動ログ管理を使用することを指定する。

LogManagement=Archive
アーカイブログ管理を使用することを指定する。

アーカイブ・ログ管理方法を選択した場合、ログ・エクステントをアーカイブするときにキュー・マネージャに通知する必要があります。 そのためには、次のコマンドを使用します。

```
SET LOG ARCHIVED(logfilename)
```

PCF コマンド '**MQCMD_SET_LOG** を使うこともできる。

関連情報
crtmqm (キュー・マネージャーの作成)
Multiplatforms での SET LOG (ログ・アーカイブ完了の通知)
AIX, Linux, and Windows での MQCMD_SET_LOG (ログ・アーカイブ完了の通知)

ディスクが満杯になったときに何が起きるか
リニア・ログを使用するようにキュー・マネージャーを構成した場合、キュー・マネージャーのロギング・コンポーネントは、ディスクが満杯になった状態に以下の方法で反応します。

ログ・ファイルを含むディスクが満杯になると、以下のようになります。

- キュー・マネージャーは、必要なサイズの新しいログ・ファイルを作成するときにのみ、この状態を検出します。これは、必要に応じて事前に検出されます。
- ファイルを必要なサイズに拡張する要求に対して、オペレーティング・システムによってエラーが返された場合、ディスクが満杯になっている状態を検出します。
- キュー・マネージャーは、メッセージ AMQ6708 をキュー・マネージャーのエラー・ログに発行します。
- First Failure Support Technology (FFST) のレコードが、システム全体にわたるエラー・ディレクトリーに書き込まれます。 このレコードは、ディスクが満杯になった状態の詳細を提供するもので、IBM サポートに連絡する必要がある場合に備えて保持する必要があります。

ログ・ファイルは、ログ記録を書き込むにつれて拡張されるのではなく、固定サイズで作成されます。 これは、IBM MQ が新しいファイルを作成しているときにのみ、ディスク・スペースが不足する可能性があることを意味します。レコードをログに書き込んでいるときにスペースが不足することはありません。 IBM MQ は、既存のログ・ファイルに使用可能なスペースがどれだけあるかを常に認識しており、ファイル内のスペースをそれに応じて管理します。

IBM MQ 9.1.0 以降では、リニア・ロギングを使用する場合に、以下のオプションが選択できます。

- ログ・エクステントの自動管理。

  新しいログ属性の詳細については、DISPLAY QMSTATUS を参照してください。

  また、以下のコマンド、または PCF でこれらに相当するコマンドも参照してください。

  - RESET QMGR
  - SET LOG (分散プラットフォームの場合)
- メディア・イメージの使用を制御するオプション。

  以下の詳細については、ALTER QMGR コマンドと ALTER QUEUES コマンドを参照してください:

  - IMGINTVL
  - IMGLOGLN
  - IMGRCOVO
  - IMGRCOVQ
  - IMGSCHED

循環ロギングでは、リソース問題が返されます。

それでもスペースが足りない場合には、キュー・マネージャーの構成ファイル 内のログの構成が正しいかどうか確認してください。 使用可能なスペースよりもログが大きくならないように、1 次または 2 次ログ・ファイルの数を減らすこともできます。

既存のキュー・マネージャーでは、ログ・ファイルのサイズを変更することはできません。 キュー・マネージャーは、すべてのログ・エクステントが同じサイズであることを必要とします。

ログ・ファイルを管理する

ログ・ファイルに十分なスペースを割り振ります。 リニア・ログでは、古いログ・ファイルが必要ではなくなったときに削除することができます。

循環ロギングに固有の情報

循環ロギングを使用している場合、システムの構成時にはログ・ファイルを保持できるだけの十分なスペースを確保してください (94 ページの『mqs.ini ファイルの LogDefaults スタンザ』および 132 ページの『qm.ini ファイルの Log スタンザ』を参照)。 ログが使用するディスク・スペースの量は、必要な時に作成される 2 次ファイ ルのスペースも含めて、構成されたサイズを超えることはありません。

リニア・ロギングに固有の情報

リニア・ログを使用している場合は、データがログに記録されるにつれて、ログ・ファイルが連続的に追加され、使用されるディスク・スペースの量は 時間と共に増大します。 データのログの速度が速いと、ディスク・スペースは新しいログ・ファイルによって急速に消費されます。

時間が経過すると、リニア・ログの古いログ・ファイルは、キュー・マネージャーを再始動したり、損傷したオブジェクトのメディア・リカバリーを実行するために必要なくなります。 以下に、この時点でどのログ・ファイルが必要かを判断するための方法を示します。

ロガー・イベント・メッセージ
重要なイベント (例えば、メディア・イメージの記録など) が発生すると、ロガー・イベント・メッセージが生成されます。 ロガー・イベント・メッセージの内容により、キュー・マネージャーの再開とメディアの回復のために現時点で必要なログ・ファイルが指定されます。 ロガー・イベント・メッセージについて詳しくは、ロガー・イベントを参照してください。

キュー・マネージャー状況
MQSC コマンドの DISPLAY QMSTATUS、または PCF コマンドの Inquire Queue Manager Status を実行すると、必要なログ・ファイルの詳細を含む、キュー・マネージャー情報が戻されます。 MQSC コマンドについて詳しくは、 MQSC コマンドを使用した IBM MQ の管理を参照してください。PCF コマンドについて詳しくは、 管理タスクの自動化を参照してください。

キュー・マネージャー・メッセージ
キュー・マネージャーは、次のような一対のメッセージを定期的に発行して、どのログ・ファイルが必要かを示します。

- メッセージ AMQ7467I は、キュー・マネージャーを再始動するために必要な最も古いログ・ファイルの名前を示しています。 このログ・ファイルおよびそれより新しいログ・ファイルはすべて、キュー・マネージャーの再始動時に使用できる必要があります。
- メッセージ AMQ7468I は、メディア・リカバリーを実行するために必要な最も古いログ・ファイルの名前を示しています。

ログ・ファイルが "古い" か "新しい" かを判別するには、ファイル・システムによって適用される変更時刻ではなく、ログ・ファイル番号を使用します。

両方のタイプのロギングに適用される情報

キュー・マネージャーの再開に必要なログ・ファイル、つまりアクティブ・ログ・ファイルのみを、オンラインにする必要があります。 非アクティブ・ログ・ファイルは、災害復旧用にテープなどのアーカイブ・メディアにコピーして、ログ・ディレクトリーから削除することができます。 メディア回復に必要とされない非アクティブ・ログ・ファイルは、不必要なログ・ファイルと見なすことができます。 不必要なログ・ファイルは、操作の対象でなくなった場合、削除することができます。

必要なログ・ファイルが見つからないと、オペレーター・メッセージ AMQ6767E が出されます。 そのログ・ファイル、およびそれ以降のすべてのログ・ファイルをキュー・マネージャーが使用できるようにして、操作を再試行してください。

ログ・エクステントを自動的にクリーンアップする - リニア・ロギングのみ

Multi

IBM MQ 9.1.0 以降には、リカバリーのために必要なくなったリニア・ログ・エクステントの自動管理を使用するオプションが用意されています。

自動管理をセットアップするには、qm.ini ファイルのログ・スタンザにある **LogManagement** 属性を使用するか、IBM MQ Explorer を使用します。 詳しくは、132 ページの『qm.ini ファイルの Log スタンザ』 を参照してください。

ログの操作に関する詳細、およびログを使用するための以下のコマンドについては、**DISPLAY QMSTATUS** の LOG パラメーターを参照してください。

- RESET QMGR
- SET LOG

メディア・イメージを自動的に作成する - リニア・ロギングのみ

IBM MQ 9.1.0 以降、キュー・マネージャーが自動的にメディア・イメージを書き込むかどうかを制御するための全体的なスイッチが装備されます。デフォルトでは、このスイッチは設定されていません。

自動メディア・イメージを実行するかどうかと、そのプロセスの頻度は、以下のキュー・マネージャー属性を使用して制御できます。

IMGSCHED
キュー・マネージャーがメディア・イメージを自動的に書き込むかどうか

IMGINTVL
メディア・イメージを書き込む頻度 (分)

IMGLOGLN
オブジェクトの前のメディア・イメージより後に書き込まれたログのメガバイト数。

1 日の中でワークロードが非常に重くなる重要な時間があり、システム・スループットが自動メディア・イメージ作成による影響を受けないようにする必要がある場合は、**IMGSCHED***(MANUAL)* を設定して自動メディア・イメージ作成を一時的にオフにすることができます。

IMGSCHED の切り替えは、ワークロード中にいつでも実行できます。

重要 : メディア・イメージを取得しない場合、 **MEDIALOG** は前進しません。そのため、エクステントをアーカイブするか、十分なディスク・スペースがあることを確認する必要があります。

IMGRCOVO 属性を使用して、他のユーザー定義オブジェクトの自動および手動メディア・イメージを制御することもできます。

- 認証情報
- チャネル
- クライアント接続
- リスナー
- 名前リスト
- プロセス
- 別名キュー
- ローカル・キュー
- サービス
- トピック

オブジェクト・カタログやキュー・マネージャー・オブジェクトなどの内部システム・オブジェクトについては、必要に応じてキュー・マネージャーが自動的にメディア・イメージを書き込みます。

この属性について詳しくは、ALTER QMGR を参照してください。

ローカル・キューと永続動的キューに対してのみ、自動および手動のメディア・イメージを有効または無効にすることもできます。 これを行うには、**IMGRCOVQ** キュー属性を使用します。

IMGRCOVQ 属性について詳しくは、 ALTER QUEUES を参照してください。

注 :

1. メディア・イメージは、リニア・ロギングを使用している場合にのみサポートされます。 自動メディア・イメージを有効にした場合に、循環ロギングが使用されているときは、エラー・メッセージが出され、キュー・マネージャーの自動メディア・イメージ属性が無効になります。
2. 自動メディア・イメージを有効にしているが、ログの頻度 (分またはメガバイト) を指定していない場合は、エラー・メッセージが出され、自動メディア・イメージは書き込まれなくなります。
3. **IMGSCHED***(AUTO)*を設定した場合は、必要に応じて rcdmqimg を使用してメディア・イメージを手動で記録できます。

   これを利用すると、企業の都合のよい時間 (例えば、システムが静止しているとき) に、メディア・イメージを作成することができます。 自動メディア・イメージ作成では、このような手動によるメディア・イメージを考慮に入れます。つまり、手動メディア・イメージを作成した時点で、次回の自動メディア・イメージ作成までの間隔とログの長さがリセットされます。

4. IBM MQ 9.1.0 から、キュー・マネージャーは、永続メッセージのみをメディア・イメージに書き込み、非永続メッセージは書き込みません。 このため、IBM MQ 9.1.0 以降にマイグレーションすると、メディア・イメージのサイズを縮小できます。

IMGLOGLN と IMGINTVL の設定値の決定

V 9.3.4 デフォルトでは、ネイティブ HA キュー・マネージャー以外のキュー・マネージャーの場合、**IMGLOGLN** は **off** に設定されます。 (ネイティブ HA キュー・マネージャーは、リカバリー・ログが書き込まれるボリューム上の使用可能スペースの 25% の値に設定された **IMGLOGLN** を使用して作成されます。)

V 9.3.4 デフォルトでは、**IMGINTVL** は 60 分に設定されています。 **IMGINTVL** で指定された間隔は、新しいイメージを記録する価値があるほど十分に新しい作業がキュー・マネージャーで実行されたときに受け入れられます。 そうしないと、新しいイメージの取得が遅延します。

IMGLOGLN および **IMGINTVL** の値を変更して、ご使用の構成に最適なソリューションを実現することができます。 **IMGINTVL** と **IMGLOGLN** の値は、キュー・マネージャーがメディア・イメージの記録に費やす時間をわずかに抑えるのに十分な大きさに設定する必要があります。ただし、次の目的を達成するのに十分なだけ、小さく設定する必要があります。

- 損傷したオブジェクトを妥当な時間内に回復できるようにする。
- ログの大きさがディスクに見合っており、スペース不足にならないようにする。

IMGLOGLN を設定する場合は、**IMGLOGLN** をキュー上のデータ量の数倍、かつワークロードのデータ速度の数倍にすることをお勧めします。 **IMGLOGLN** を大きくすると、キュー・マネージャーがメディア・イメージの記録に費やす時間が少なくなります。

同様に、**IMGINTVL** を設定する場合は、**IMGINTVL** を、キュー・マネージャーがメディア・イメージを記録するために費やす時間の数倍にすることをお勧めします。 メディア・イメージを記録するのにかかる時間は、手動で記録して調べることができます。

IMGLOGLN と **IMGINTVL** が大きすぎると、損傷したオブジェクトをリカバリーするときに、最後のメディア・イメージ以後のすべての範囲を再生する必要があるため、非常に長時間かかることがあります。

損傷したオブジェクトの回復にかかる最大時間を許容範囲内に収めるには、**IMGLOGLN** と **IMGINTVL** の値を十分に小さくします。

IMGINTVL と **IMGLOGLN** を非常に大きい値にすると、メディア・イメージを記録する頻度が下がるため、ログが非常に大きくなります。

重要: ログ・ファイル・システムが完全に満杯になると、ワークロードがバックアウトされることになるため、このサイズのログをログ・ファイル・システムに無理なく格納できることを確認してください。

IMGINTVL と **IMGLOGLN** の両方を設定できます。 両方を設定すると、ワークロードが重い間に自動メディア・イメージが定期的に作成される (**IMGLOGLN** によって制御) 一方で、ワークロードが非常に軽い時にも時折作成されます (**IMGINTVL** によって制御)。

IMGINTVL と **IMGLOGLN** は、自動メディア・イメージが作成される間隔とログ・データ長の目標値です。

これらの属性を、固定された最大値または最小値と見なすべきではありません。 実際、キュー・マネージャーは、以下のような場合には、適切な機会を見計らって、早期に自動メディア・イメージをスケジュールすることがあります。

- キューが空の場合。パフォーマンスの観点から、メディア・イメージ作成の効率が最も高いと判断されます。
- しばらくの間メディア・イメージが記録されていない場合。

時折、自動メディア・イメージの作成間隔が **IMGINTVL** と **IMGLOGLN** の一方または両方より少し長くなることがあります。

キュー上のデータの量が **IMGLOGLN** に近づくと、メディア・イメージの間隔が **IMGLOGLN** より大きくなることがあります。 メディア・イメージの記録に **IMGINTVL** とほぼ同じ時間がかかる場合は、メディア・イメージの間隔が **IMGINTVL** より大きくなることがあります。

これでは、キュー・マネージャーが大半の時間をメディア・イメージの記録に費やすことになるので、望ましくありません。

自動メディア・イメージ記録を使用している場合、キュー・マネージャーはオブジェクトごとおよびキューごとに個別にメディア・イメージを記録します。そのため、キュー・マネージャーは、イメージの間隔とログ長をオブジェクトごとに個別に追跡します。

時間の経過とともに、メディア・イメージの記録は、すべてのオブジェクトについて同時にメディア・イメージを記録する状態から、交互に記録する状態に変化していきます。 この状態になると、メディア・イメージの記録がパフォーマンスに与える影響が分散します。これは、メディア・イメージの手動記録よりも自動記録を使用するほうが優れているもう 1 つの利点です。

メディア・イメージを手動で作成する - リニア・ロギングのみ

キューのメディア・イメージを記録するには、そのキューのすべての永続メッセージをログに書き込む必要があります。 大量のメッセージ・データを含むキューの場合、ログに大量のデータが書き込まれるので、このプロセスが実行されている間、システムのパフォーマンスが影響を受ける可能性があります。

その他のオブジェクトのメディア・イメージの記録は、その他のオブジェクトのメディア・イメージにはユーザー・データが含まれていないため、比較的速く実行できます。

キューのメディア・イメージを記録するタイミングは、そのプロセスがピーク時のワークロードを邪魔しないように、注意深く検討する必要があります。

すべてのオブジェクトのメディア・イメージを定期的に記録して、メディア・リカバリーに必要な最も古いログ・エクステントを更新する必要があります。

キューのメディア・イメージを記録する良いタイミングは、キューが空のときです。その時点ではログに書き込まれるメッセージ・データがないからです。 反対に、悪いタイミングは、キューが非常に深いときや、非常に大きなメッセージがキューにあるときです。

キューのメディア・イメージを記録するのに良いタイミングは、システムが静かなときで、悪いタイミングはピーク・ワークロードの最中です。 例えば、常に午前 0 時のワークロードが静かな場合は、メディア・イメージを毎晩午前 0 時に記録することを決定できます。

各キューの記録を別々のタイミングにすると、パフォーマンスへの影響を分散できるため、記録の影響を減らすことができます。 メディア・イメージを最後に記録してからの経過時間が長くなるほど、メディア・イメージを記録する重要性は高くなります。メディア・リカバリーに必要なログ・エクステント数が増えるからです。

注 : メディア・リカバリーの実行中に、必要なログ・ファイルはすべてログ・ファイル・ ディレクトリーの中で同時に使用可能である必要があります。 必要なログ・ファイルをすべて保持するためのディスク・スペースが不足しないようにするために、 リカバリーするオブジェクトのメディア・イメージを必ず定期的にとるようにしてください。

例えば、キュー・マネージャー内のすべてのオブジェクトのメディア・イメージを作成するには、次の例で示されているように **rcdmqimg** コマンドを実行します。

Windows **Windows 上**

```
rcdmqimg -m QMNAME -t all *
```

Linux AIX **AIX and Linux 上**

```
rcdmqimg -m QMNAME -t all "*"
```

rcdmqimg コマンドの実行により、メディアのログ順序番号 (LSN) が順方向に移動します。 ログ順序番号の詳細については、667 ページの『dmpmqlog コマンドを使用したログの内容のダンプ』を参照してください。 **rcdmqimg** は自動的には実行されません。 したがって、手動で実行するか、または作成した自動タ

スクから実行する必要があります。 このコマンドの詳細については、rcdmqimg および dmpmqlog を参照してください。

キュー・マネージャーに制御される自動メディア・イメージでリニア・ロギングを使用する場合は、ログ・スペースを管理するために **rcdmqimg** を使用してメディア・イメージを手動記録する必要はありません。

注 : rcdmqimg コマンドの実行時に、メッセージ AMQ7467 および AMQ7468 が出されることもあります。

部分メディア・イメージ

IBM MQ メッセージは、近い将来にコンシュームされるデータに対してのみ使用することをお勧めします。各メッセージがキューに保持される時間が比較的短くなるからです。

一方で、データベースのように長期間データを保管するために IBM MQ メッセージを使用するのは望ましくありません。

また、キューを比較的浅い状態に保持することもお勧めします。逆に、メッセージをキューに長期間保持してキューを深い状態にするのは望ましくありません。

このようなガイドラインに従うと、キュー・マネージャーによるメディア・イメージの自動記録のパフォーマンスを最適化できます。

空のキューのメディア・イメージを記録するのは、(パフォーマンスの観点から) 非常に効率の良い方法です。逆に、大量のデータが入ったキューのメディア・イメージを作成するのは、それらのデータをすべてメディア・イメージのログに書き込む必要があるため、非常に効率の悪い方法です。

直前にメッセージを書き込んだ浅いキューの場合、キュー・マネージャーはさらに最適化を行うことができます。

現在キューに入っているすべてのメッセージが最近書き込まれたものであるとすると、キュー・マネージャーは、すべてのメッセージが書き込まれた直前 (リカバリー・ポイント) のメディア・イメージを記録することができます。つまり、空のキューのイメージを記録できます。 このプロセスは、パフォーマンス上、非常に低コストです。

リカバリー・ポイントの時点でキューに存在していたすべてのメッセージが、その後でキューから取得された場合は、それらのメッセージがメディア・イメージに記録されている必要はありません。それらのメッセージは既にキューに存在しないからです。

これは、部分メディア・イメージ と呼ばれます。 その後、稀なケースですがキューのリカバリーが必要になった場合は、最後のメディア・イメージより後の、このキューに関連したすべてのログ・レコードを再生することにより、最近書き込まれたメッセージをすべて復元します。

仮に、リカバリー・ポイントの時点で少数のメッセージがキューにあり、現在もキューにある (そのため、部分メディア・イメージに記録する必要がある) としても、すべてのメッセージの完全なメディア・イメージを作成するよりも、小さな部分メディア・イメージを記録するほうが効率的です。

メッセージが短時間だけキューに滞在するようにすれば、メディア・イメージの自動記録のパフォーマンスが向上する可能性があります。

不必要なログ・ファイルの判別 - リニア・ロギングのみ

循環ロギングでは、ログ・ディレクトリーからデータを決して削除しないでください。 リニア・ログ・ファイルを管理する場合、どのファイルを削除またはアーカイブできるかを確認することが重要です。 以下の情報は、この決定を行う上で役立ちます。

ファイル・システムの変更日時によって "古い" ログ・ファイルの判別を行うことはしないでください。 ログ・ファイル番号のみを使用してください。 キュー・マネージャーによるログ・ファイルの使用は、複雑な規則 (必要になる前に行われる事前割り振りやログ・ファイルの形式設定など) に従っています。 変更日時を含むログ・ファイルを参照する際に、それらの変更日時によって相対的な経過時間を判別しようとすると、誤解が生じる場合があります。

必要な最も古いログ・ファイルを判別するために、以下の 3 つの場所を使用できます。

- DISPLAY QMSTATUS コマンド
- ロガー・イベント・メッセージ、そして最後に

- エラー・ログ・メッセージ

DISPLAY QMSTATUS コマンドで、それぞれの目的のために必要な最も古いログ・エクステントを判別する方法は以下のとおりです。

- キュー・マネージャーを再始動するためには、コマンド DISPLAY QMSTATUS RECLOG を発行します。
- メディア・リカバリーを実行するためには、コマンド DISPLAY QMSTATUS MEDIALOG を発行します。
- アーカイブ通知の名前を判別し、コマンド DISPLAY QMSTATUS ARCHLOG を発行します。

コマンド **RESET QMGR TYPE(REDUCELOG)** を発行することによって、循環ロギングを使用する際に 2 次ログのエクステントの数を削減できます。

一般的に、ログ・ファイル番号が小さい方が、より古いログ・ファイルであることを意味します。 ログ・ファイルの回転率が非常に高い場合 (10 年間で毎日 3000 個のログ・ファイルの配列) を除いて、9 999 999 での数の折り返しを考慮に入れる必要はありません。 この場合、RECLOG 値よりも小さい番号のログ・ファイルをアーカイブし、RECLOG 値と MEDIALOG 値の両方よりも小さい番号のログ・ファイルを削除することができます。

重要 : ログ・ファイルは循環するので、9 999 999 の次の番号はゼロになります。

ログ・ファイルの位置
ログ・ファイルの位置を選択する際には、ディスク・スペースの不足のために IBM MQ が新しいログをフォーマットできないと、操作に重大な影響が及ぶので注意してください。

循環ログを使用している場合は、少なくとも構成された 1 次ログ・ファイル 用の十分なスペースをドライブ上に確保してください。 また、ログが大きくなる場合に必要な少なくとも 1 つの 2 次ログ・ファイル用のスペースも残しておいてください。

リニア・ログを使用している場合は、十分にスペースを考慮しておく必要があります。というのは、ログが使用するスペースは、データがログに記録されるにつれて絶えず増大するからです。

ログ・ファイルは、キュー・マネージャー・データとは別のディスク・ドライブに配置する必要があります。

このデバイスではデータ保全性が非常に重要です - 組み込みの冗長性を許可する必要があります。

また、ミラーリング配置された複数のディスク・ドライブにログ・ファイルを置く こともできます。 これは、ログを含んでいるドライブに障害が起きた場合に保護として機能します。 ミラーリングがない場合にはおそらく、IBM MQ システムの最新のバックアップを使用する以外に方法はありません。

コールド・スタート: ログ・エクステントが欠落しているか壊れている場合の対処

企業で再始動リカバリーを行うために必要なログ・エクステントの一部またはすべてが失われると、キュー・マネージャーがリカバリー・ログを再生できなくなり、再始動に失敗します。 リカバリー・ログで何らかの破損が生じるときにキュー・マネージャーを再始動する必要がある場合には、データの整合性が犠牲になるため極力避けるべきですが、再始動を行うことが可能です。 この処理は、キュー・マネージャーのコールド・スタート とも呼ばれます。

重要 : キュー・マネージャーのコールド・スタートは例外的な状況でのみ検討すべきであり、実行する場合はこのページで説明されているデータ保全性リスクが伴います。 IBM では、データ・ファイル破損に対応するうえで、コールド・スタートよりもキュー・マネージャーの再作成を推奨しています。

運用上の理由からコールド・スタートが必要な場合は、問題の根本原因を確認するために IBM サポート担当者に連絡してください。 コールド・スタートさせたキュー・マネージャーは、再作成したキュー・マネージャーとなるべく早く取り替える必要があります。

コールド・スタートの影響

コールド・スタートを行うと、キュー・マネージャーによって空のリカバリー・ログが作成され、キュー・ファイルにあるデータと他のオブジェクト・ファイル内にあるデータを既存の状態のまま使用します。 キュー・ファイル内のデータは不整合な状態になっている可能性があるため、メッセージが失われたり、重複したり、破損したり、不整合な状態になったりすることがあります。

キュー・マネージャーは、他のすべての永続オブジェクトの構成をリカバリー・ログおよびオブジェクト・ファイルに保管します。 その他の内部状態データもリカバリー・ログに記録されるので、コールド・スタート時には内部状態データはリセットされ、他のすべての構成データは不正確になる可能性があります。

コールド・スタートの影響を事前に予測することはできず、広範囲に及ぶため、コールド・スタートは絶対に必要な場合を除き行わないでください。 コールド・スタート後、キュー・ファイルとオブジェクト・ファイル内の情報は不整合な状態になり、キュー・マネージャーがまったく再始動できなくなる可能性があります。

キュー・マネージャーが再始動する場合でも、信頼できるメッセージ・データまたは構成と、信頼できないものとを簡単に特定することはできません。 また、コールド・スタート後にはキューが破損し、まったく使用できなくなる可能性もあります。

さらに、特定のキューとの間で読み書きができる場合でも、そのメッセージが破損、欠落、重複していることもあります。 トランザクションおよびチャネルが未確定のままになっている可能性があります。 キュー・マネージャーによるコールド・スタートが正常に実行され、キューが変更されていないように見えても、コールド・スタートが後ほどどのような影響を及ぼすかについては予測できず、時間が経過しないと分かりません。

コールド・スタートが必要な場合の処置

コールド・スタートの実行は標準的な運用慣例と見なすべきではなく、IBM ではこれを実行しないことを強く推奨します。 ただし、キュー・マネージャーのコールド・スタートがどうしても必要な場合には、IBMMQ サポートにお問い合わせください。

キュー・マネージャーのコールド・スタートのプロセスは、リニア・キュー・マネージャーのほうが循環キュー・マネージャーと比べて複雑でした。 IBM MQ 9.1.3 では、コールド・スタート・プロセスが簡単になり、ログ・エクステントのコピーまたは名前変更が必要なくなりました。

IBM MQ 9.1.3 以降では IBM サポートに問い合わせ、キュー・マネージャーをコールド・スタートするために **strmqm** コマンドに渡すキーをお受け取りください。

重要 : IBM MQ 9.1.3 コールド・スタート・コマンドも手動のコールド・スタート同様、データ整合性を失う危険性があり、IBM ではこれを実行しないことを強く推奨します。

今後のコールド・スタートの必要性を減らす: お願い

strmqm コマンドにはコールドスタートのキーが必要です。IBM MQ これは、コールドスタートが必要な場合に IBM MQ サポートに連絡する必要があるためです。また、IBM MQ この状況に陥った経緯を理解することにも熱心です。

コールド・スタートは行わないのが最善であることははっきりしています。 IBM MQ ではキュー・マネージャーのコールド・スタートが不要になるよう真摯な努力を傾けてきました。同時に、IBM ではコールド・スタートが必要となる状況を回避するために製品側でできることの把握に努めています。

コールド・スタートを回避するための予防措置

キュー・マネージャーの作成時におけるデフォルトのロギング方式は循環ロギングです。 循環ロギングを使用すると、キュー・マネージャーで、特定数の、指定サイズの 1 次ログ・エクステントと 2 次ログ・エクステントが確保されます。 1 次と 2 次のログ・エクステントすべてを含めることのできる十分な大きさのログ・ファイル・システムを作成します。このようにすると、その後は管理が不要です。

あるいは、循環ではなくリニア・ロギングを使用することもできます。 リニア・ロギングを使用すると、万が一、キューや他のオブジェクトが損傷を受けたときにリカバリーする機能が追加されます。 ただしデフォルトでは、リニア・ロギングでは、再始動またはメディア・リカバリーに不要なログ・エクステントを削除する必要があります。 これは、手動ログ管理と呼ばれます。

このような方法でログ・エクステントを管理する場合、誤って多すぎる数のログ・エクステントを削除してしまい、結果的にコールド・スタートしなければならなくなることがあります。 このリスクを軽減するには、自動ログ管理を使用し、ユーザーの代わりにキュー・マネージャーがログ・エクステントを管理するようにします。

ベスト・プラクティスは、リカバリー・ログだけが含まれる、別個のログ・ファイル・システムにリカバリー・ログを配置することです。 リカバリー・ログをキュー・マネージャーのその他のファイルと同じファイル・システムに置くと、大きなキュー・ファイルなどが原因となり、ファイル・システムが意図せずいっぱいになってしまう可能性があります。 キュー・マネージャーのログ・ディレクトリーを別のファイル・システムに配置するか、**crtmqm** コマンドで **-ld** コマンド・ライン・オプションを使用して別のログ・ファイル・システムを指定します。

キュー・ファイルを保持しているファイル・システムがいっぱいになると、それらのキューに書き込むことができなくなりますが、キュー・マネージャーは実行を続けます。 リカバリー・ログを含むファイル・システムが満杯になると、キュー・マネージャーは突然に終了し、空き容量を確保するまで再始動できなくなります。

再始動リカバリーに必要なログ・エクステントを削除しないように注意してください。削除すると、コールド・スタートが必要になることがあります。 リカバリー・ログを含むディスクに障害が発生したためにコールド・スタートが必要になる場合があります。 ベスト・プラクティスは、複製したディスク上にリカバリー・ログを配置し、ディスク・クラッシュによるリスクを軽減する方法です。

メッセージと構成を新しい交換キュー・マネージャーに移動すると、以前にコールド・スタートしたキュー・マネージャーで継続的問題が生じる可能性が排除されます。

以前にコールド・スタートしたキュー・マネージャーについて記録を取ってください。コールド・スタートしたのがかなり前で、長期間停止されていたり、その後、再始動したり、マイグレーションしたりした場合であっても記録してください。 IBM サポートに連絡する場合、キュー・マネージャーを以前にコールド・スタートしたことがある場合にはそのことを伝え、コールド・スタートが必要となった原因について可能な限り詳細をお知らせください。

ログをリカバリーに使用する

ログの情報を障害からのリカバリーに役立てることができます。

データが損傷する原因は、いくつか考えられます。 IBM MQ を利用すると次のような状況からリカバリーするのに役立ちます。

- データ・オブジェクトの損傷
- システムの停電
- 通信障害

このセクションでは、これらの問題からリカバリーするのにログがどのように使用されるかを 調べます。

停電や通信障害からのリカバリー

IBM MQ は、通信障害や停電による障害からリカバリーすることができます。 他の種類の問題 (不注意によるファイルの削除など) からのリカバリーも可能です。

通信障害の場合、持続メッセージは、受信側のアプリケーションによって取り除かれるまでキューに残っています。 メッセージが伝送中である場合は、正常に伝送できるまで、メッセージは 伝送キューにとどまっています。 通信障害からのリカバリーは、通常の場合、障害を起こしたリンクを使用して いるチャネルを再始動することで可能です。

停電の場合には、キュー・マネージャーが再始動されたときに、IBM MQ が障害が起きたときのコミットした状態にキューを復元します。 これで、持続メッセージが失われることはありません。 非持続メッセージは廃棄されます。非持続メッセージは、IBM MQ が不意に停止した場合には残存しません。

損傷オブジェクトをリカバリーする

IBM MQ オブジェクトが使用不能になる場合があります。 例えば、不注意で損傷した場合などです。 そのようなときには、システム全体かまたはその一部をリカバリーする必要があります。 どのような処置が必要かは、損傷が検出された時期、選択されているログ方法がメディア・リカバリーをサポートするかどうか、およびどのオブジェクトが損傷したかによって異なります。

メディア・リカバリー

オブジェクトのメディア・イメージを記録して、損傷した場合にリカバリーできるようにすることができます。 この機能は、リニア・ロギング または複製ロギング を使用するキュー・マネージャーでのみ使用で

きます。リニア・ロギングの場合は、リカバリー可能として定義されているオブジェクトでのみ使用できます。 **IMGRCOVO** および **IMGRCOVQ** キュー・マネージャー属性を使用して、オブジェクトのタイプがリカバリー可能であることを定義します。 ALTER QMGR を参照してください。 リカバリー可能として定義されていないオブジェクトが損傷した場合、リカバリー・オプションは循環ロギングの場合と同じです。

メディア・リカバリーは、リニア・ログ または複製されたログに記録された情報からオブジェクトを再作成します。 例えば、オブジェクト・ファイルが不注意で削除された場合、あるいはその他の理由で使用不能になった場合、それをメディア・リカバリーによって再作成できます。 オブジェクトのメディア・リカバリーに必要なログ内の情報のことを、メディア・イメージ と呼びます。

メディア・イメージは、オブジェクトのイメージを含んでいる一連のログ・レコード であり、そこからオブジェクト自体を再作成できます。

オブジェクトを再作成するのに必要な最初のログ・レコードは、メディア・リカバリー・レコード として知られています。これは、オブジェクトの最新のメディア・イメージの開始点です。 各オブジェクトのメディア・リカバリー・レコードは、チェックポイント時に記録された情報の一部です。

メディア・イメージからオブジェクトを再作成する際には、最新のイメージが 取られた時以降にオブジェクトに対して行われた更新を記述しているログ・ ファイルを再生する必要もあります。

例えば、持続メッセージがキューに書き込まれる前にとられたキュー・ オブジェクトのイメージを持っているローカル・キューを考えてみます。 そのオブジェクトの最新のイメージを再作成するには、イメージ自体を 再生するだけでなく、キューにメッセージが書き込まれたことを記録 しているログ項目を再生することも必要です。

オブジェクトが作成されるときに、作成されるログ・レコードには、そのオブジェクトを完全に再作成するのに十分な情報が入れられます。 それらのレコードは、オブジェクトの最初のメディア・イメージを構成す ることになります。 続いて、シャットダウンするたびに、キュー・マネージャーは自動的に以下のようなメディア・イメージを記録します。

- すべてのプロセス・オブジェクトおよび非ローカル・キューのイメージ
- 空のローカル・キューのイメージ

メディア・イメージは、 rcdmqimg で説明されているように、 **rcdmqimg** コマンドを使用して手動で記録することもできます。 このコマンドは、 IBM MQ オブジェクトのメディア・イメージを記録します。

IMGSCHED*(AUTO)* が設定されている場合、キュー・マネージャーはメディア・イメージを自動的に記録します。 詳しくは、 ALTER QMGR を参照して、 **IMGINTVL** および **INGLOGLN** に関する情報を確認してください。

メディア・イメージの書き込みが行われた後は、損傷しているオブジェクトを再作成するために必要なのは、そのメディア・イメージを持つログと、この時点以後に作成されたすべてのログだけです。 メディア・イメージを作成する利点は、利用できるフリー・ストレージの容量およびログ・ファイルが作成される速度などの要因によって異なります。

メディア・イメージからのリカバリー

キュー・マネージャーは、キュー・マネージャーの始動時に、そのメディア・イメージから一部のオブジェクトを自動的にリカバリーします。 キュー・マネージャーの最後のシャットダウン時に未完了だったトランザクションにキューが関与していて、再始動処理時にそのキューが破損または損傷していることが分かると、キュー・マネージャーはそのキューを自動的にリカバリーします。

その他のオブジェクトについては、**rcrmqobj** コマンドを使用して手動でリカバリーする必要があります。このコマンドは、IBM MQ オブジェクトの再作成のためにログのレコードを再生するものです。 オブジェクトは、ログの中にある最新のイメージから再作成されます。また、その際に、イメージが保管されてから再作成コマンドが出されるまでの間の 適当なログ・イベントもすべて再作成に使用されます。 IBM MQ オブジェクトが損傷した場合には、実行できる有効な処置は、オブジェクトを削除するか、あるいはこの方法でオブジェクトを再作成することです。 非持続メッセージは、今述べた方法ではリカバリーできません。

rcrmqobj コマンドの詳細については、 rcrmqobj を参照してください。

オブジェクトのメディア・リカバリーを試みる際には、メディア・リカバリー・レコードを含んでいるログ・ファイル、およびそれ以降のすべてのログ・ファイルがログ・ファイル・ディレクトリーの中にあって、使用可能状態になっていなければなりません。 必要なファイルが見つからない場合には、オペレーター・メッセージ AMQ6767 が出され、メディア・リカバリー操作は失敗します。 再作成するオブジェクトのメディア・イメージを定期的に取っていないと、オブジェクトを再作成するために必要なログ・ファイルすべてを保持するためのディスク・スペースが、不足する可能性があります。

V 9.3.3 ネイティブ HA キュー・マネージャーは、複製ロギングを使用します。 このようなキュー・マネージャーは、損傷が検出されると、適格なオブジェクトの自動リカバリーを試みます。 ネイティブ HA キュー・マネージャーは、開始されると、オブジェクトの損傷が検出されたときに、デフォルトで自動的に非同期リカバリーを試みます。 例えば、オブジェクトがアプリケーションによって使用されている場合や、メディア・リカバリーに必要なログ・エクステントが使用できない場合には、リカバリーを即時に実行できないことがあります。 このような状況では、非同期リカバリー処理は定期的に再試行します。 リカバリーを妨げた問題が解決された場合、オブジェクトは次の再試行時にリカバリーされるか、または **rcrmqobj** コマンドを使用してオブジェクトを手動でリカバリーすることができます。

存在するオブジェクト・ファイル

キュー・マネージャーは **runmqsc** に定義されたオブジェクトの属性をディスク上のファイルに保管します。 これらのオブジェクト・ファイルはキュー・マネージャーのデータ・ディレクトリーの下にあるサブディレクトリーにあります。

Linux AIX 例えば、AIX and Linux プラットフォームでは、チャネルは/var/mqm/qmgrs/*qmgr*/channel に保管されます。

これらのオブジェクト・ファイル内のデータは、オブジェクトのメディア・イメージです。 これらのオブジェクト・ファイルが削除されたり破損したりすると、ファイル内に保管されたオブジェクトが破損します。 リニア・ロギングのキュー・マネージャーを使用する場合は、rcrmqobj コマンドを使用して、破損したオブジェクトをログからリカバリーできます。 複製ロギング (ネイティブ HA) キュー・マネージャーは、損傷したオブジェクトが検出されると、それらのリカバリーを自動的に試行します。

ほとんどのオブジェクト・ファイルには、そのオブジェクトの属性のみが含まれています。そのため、チャネル・ファイルにはチャネルの属性が含まれています。 ただし、次の例外があります。

- カタログ

  オブジェクト・カタログは、すべてのタイプのすべてのオブジェクトをカタログし、qmanager/QMQMOBJCAT に保管されます。

- 同期ファイル

  同期ファイルには、すべてのチャネルに関連する内部状態データが含まれています。

- キュー

  キュー・ファイルには、そのキューのメッセージとキューの属性の両方が含まれています。

カタログ・オブジェクトも同期ファイル・オブジェクトも **runmqsc** または IBM MQ Explorer で公開されないことに注意してください。

カタログおよびキュー・マネージャーは、記録はできますがリカバリーはできません。 これらのオブジェクトが破損した場合、キュー・マネージャーは先回りして終了し、再始動時にこれらのオブジェクトが自動的にリカバリーされます。

サブスクリプションは、永続サブスクリプションがシステム・キューに保管されているため、記録またはリカバリーするオブジェクトにリストされません。 永続サブスクリプションを記録またはリカバリーするには、代わりに SYSTEM.DURABLE.SUBSCRIBER.QUEUE を記録またはリカバリーします。

始動時に損傷オブジェクトをリカバリーする

始動時に損傷オブジェクトを見つけた場合、キュー・マネージャーが行うアクションは、オブジェクトのタイプや、キュー・マネージャーがメディア・リカバリーをサポートするように構成されているかどうかによって異なります。

キュー・マネージャーのオブジェクトが損傷している場合、キュー・マネージャーはそのオブジェクトをリカバリーできなければ、開始できません。 キュー・マネージャーがリニア・ログと共に構成されていて、メディア・リカバリーをサポートしている場合には、IBM MQ は、自動的にメディア・イメージからキュー・マネージャーのオブジェクトを再作成しようとします。 選択したログの方法がメディア・リカバリーをサポートしていない場合には、キュー・マネージャーのバックアップを復元するか、あるいはキュー・マネージャーを削除することができます。

キュー・マネージャーが停止したときにアクティブ状態のトランザクションが あると、それらのトランザクションの中で書き込まれたり 読み取られたりした持続メッセージ (コミットされていないもの) のあるローカル・キューも、キュー・マネージャーを正常に始動するためには必要です。 キュー・マネージャーがメディア・リカバリーをサポートしている場合で、これらの キューのいずれかが損傷していることが分かった場合は、キュー・マネージャーは、メディア・イメージからそれらのキューを自動的に再作成しようとします。 それらのキューのいずれかがリカバリーできないときは、IBM MQ は始動できません。

メディア・リカバリーをサポートしないキュー・マネージャーの始動処理時に、コミット されていないメッセージを含んでいる損傷ローカル・キューが検出された 場合、それらのキューには損傷オブジェクトのマークが付けられ、そのキューにあるコミットされていないメッセージは無視されます。 その理由は、そのようなキュー・マネージャーでは、損傷オブジェクトのメディア・リカバリーを実行できず、そのオブジェクトを削除するという処置しか行えないため です。 メッセージ AMQ7472 が出て、損傷が通知されます。

その他の場合の損傷オブジェクトのリカバリー

オブジェクトのメディア・リカバリーは、始動時にのみ自動的に行われます (デフォルトで自動リカバリーを使用するネイティブ HA キュー・マネージャーの場合を除く)。 また、オブジェクト損傷が検出されると、オペレーター・メッセージ AMQ7472 が出され、そのオブジェクトを使用するほとんどの操作は失敗し、戻りコード MQRC_OBJECT_DAMAGED が返されます。 キュー・マネージャーの始動後の任意の時点でキュー・マネージャー・オブジェクトが損傷した場合、キュー・マネージャーはプリエンプティブ・シャットダウンを実行します。 オブジェクトが損傷している場合には、それを削除することができます。あるいは、キュー・マネージャーがリニア・ログを使用している場合には、**rcrmqobj** コマンドを使用して、メディア・イメージからそのオブジェクトのリカバリーを試みて ください (詳細については、rcrmqobj を参照してください)。

キュー (または他のオブジェクト) が損傷を受けた場合、**MEDIALOG** は進行しません。 これは、**MEDIALOG** がメディア・リカバリーに必要な最も古いエクステントであるためです。 ワークロードが続いている場合、**CURRLOG** は引き続き進行しているので、新しいエクステントが書き込まれます。 構成 (**LogManagement** 設定を含む) によっては、ログ・ファイル・システムが満杯になり始める可能性があります。 ログ・ファイル・システムが完全にフルになると、トランザクションはロールバックされて、キュー・マネージャーが突然終了する可能性があります。 そのため、キューが損傷すると、キュー・マネージャーが終了する前に対処するには限られた時間しかない可能性があります。 どれだけの時間があるかは、キュー・マネージャーがワークロードによる新しいエクステントを書き込んでいる速度、およびログ・ファイル・システムに存在する空きスペースのサイズによって異なります。

手動のログ管理を使用している場合には、再始動リカバリーには必要のないエクステントをアーカイブした後に、それらがメディア・リカバリー用にまだ必要であるにもかかわらず、ログ・ファイル・システムから削除している可能性があります。 これは、それらが必要なときにアーカイブから復元できる限りは許容されることです。 このポリシーでは、キューが損傷して **MEDIALOG** が進行を停止しても、ログ・ファイル・システムがフルになることはありません。 ただし、再始動リカバリーとメディア・リカバリーのいずれにも必要のないエクステントをアーカイブして削除するだけであれば、キューが損傷した場合に、ログ・ファイル・システムはフルになっていきます。

自動またはアーカイブ・ログ管理を使用している場合、SET LOG ARCHIVED を使用してアーカイブしてからキュー・マネージャーに通知した場合でも、キュー・マネージャーはメディア・リカバリーに必要なエクステントを再利用しません。 その結果、キューが損傷した場合には、ログ・ファイル・システムがフルになっていきます。

キューが損傷すると、OBJECT DAMAGED FFDC が書き込まれて、**MEDIALOG** は進行を停止します。 損傷したオブジェクトは、FFDC から識別できます。 または、**runmqsc** で状態を表示したときに、最も古い **MEDIALOG** を持つオブジェクトとして識別することもできます。

ログ・ファイル・システムがフルになっていく場合、ログ・ファイル・システムがフルになってワークロードがバックアウトする心配が生じます。 オブジェクトをリカバリーすることにより、またはワークロードを静止させることにより、そのような事態を回避できることがあります。

V 9.3.3 (複製ロギングを使用する) ネイティブ HA キュー・マネージャーの場合、損傷したオブジェクトの自動リカバリーが試行されます。 ネイティブ HA キュー・マネージャーは、開始されると、オブジェクトの損傷が検出されたときに、デフォルトで自動的に非同期リカバリーを試みます。 例えば、オブジェクトがアプリケーションによって使用されている場合や、メディア・リカバリーに必要なログ・エクステントが使用できない場合には、リカバリーを即時に実行できないことがあります。 このような状況では、非同期リカバリー処理は定期的に再試行します。 リカバリーを妨げた問題が解決された場合、オブジェクトは次の再試行時にリカバリーされるか、または **rcrmqobj** コマンドを使用してオブジェクトを手動でリカバリーすることができます。

IBM MQ ログ・ファイルの保護

キュー・マネージャーの実行中はログ・ファイルに触らないでください。 リカバリーできなくなる場合があります。 誤って変更しないように、 スーパーユーザー権限または mqm 権限を使用してログ・ファイルを保護してください。

IBM MQ キュー・マネージャーの実行中は、アクティブ・ログ・ファイルを手動で削除しないでください。 キュー・マネージャーの再始動に必要なログ・ファイルをユーザーが誤って削除した場合でも、IBM MQ はエラー・メッセージを**発行しない**で、持続メッセージが入っているデータの処理を続けます。 この場合、キュー・マネージャーは正常にシャットダウンしますが、再始動できない場合あります。 そして、メッセージのリカバリーも不可能になります。

アクティブなキュー・マネージャーで使用中のログを削除する権限を持つユーザーは、キュー・マネージャーのその他の重要なリソース (キュー・ファイル、オブジェクト・カタログ、IBM MQ 実行可能ファイルなど) を削除する権限も持っています。 したがって、そのようなユーザーは、おそらくは経験不足から、IBM MQ が防ぎきれない方法で実行中または休止中のキュー・マネージャーに損傷を与える場合があります。

スーパーユーザー権限または mqm 権限を付与する場合には、十分に注意してください。

dmpmqlog コマンドを使用したログの内容のダンプ

dmpmqlog コマンドを使用して、キュー・マネージャーのログの内容をダンプする方法。

キュー・マネージャーのログの内容をダンプするには、dmpmqlog コマンドを使用します。 デフォルトでは、アクティブなログ・レコードがすべてダンプされます。つまり、このコマンドは、ログのヘッドからダンプを開始します (通常、最後に完了したチェックポイントの開始)。

通常、キュー・マネージャーが実行されていない場合にのみログをダンプできます。 終了中にキュー・マネージャーはチェックポイントを取るため、通常ログのアクティブ部分には、少数のログ・レコードが入ります。 ただし、次のオプションのいずれかを使用すると、dmpmqlog コマンドを使用してログ・レコードをより多くダンプし、ダンプの開始位置を変更することができます。

- ログのベース からダンプを開始します。 ログのベースとは、ログのヘッドを含むログ・ファイルの中にある最初のログ・レコ ードです。 この場合、ダンプされるデータ追加量は、ログのヘッドがログ・ファイルの中で 位置付けされる場所によって異なります。 ログ・ファイルの最初に近い場合には、少量の追加データしかダンプされません。 ログ・ファイルの終わりに近い場合には、極めて多量のデータがダンプされます。
- ダンプの開始位置を個々のログ・レコードとして指定します。 各ログ・レコードは、固有のログ順序番号 *(LSN)* により識別され ます。 循環ログの場合、この開始ログ・レコードをログのベースの前に置くことはで きません。この制限は、リニア・ログには適用されません。 コマンドの実行前に、非アクティブ・ログ・ファイルを復元する必要があるかもしれません。 開始位置として、前の dmpmqlog 出力から取った、有効な LSN を指定することが必要です。

  例えば、リニア・ロギングでは、最後の dmpmqlog 出力から nextlsn を指定します。 すると、次の nextlsn が Log File Header に表示され、次のログ ・レコードの LSN が書き込まれることを示しています。 ログが最後にダンプされた後に、書き込まれるログ・レコードを すべてフォーマットするため、開始位置としてこれを使用することができます。

- **リニア・ログのみの場合**、指定のログ・ファイル・エクステントからログ・レコードのフォーマット化を開始するよう dmpmqlog に指示できます。 この場合、dmpmqlog により、このログ・ファイル、およびアクティ ブ・ログ・ファイルと同じディレクトリーにある次のファイルが検索されます。 このオプションは循環ログには適用されません。dmpmqlog がログのベース前のログ・レコードにアクセスできないためです。

dmpmqlog コマンドからの出力は、 Log File Header および一連のフォーマット済みログ・レコードです。 キュー・マネージャーは、いくつかのログ・レコードを使用して、データに対 する変更を記録します。

フォーマットされる一部の情報は、内部的に使用されるものです。 次のリストには、最も役立つログ・レコードを示しています。

ログ・ファイルのヘッダー
各ログには単一のログ・ファイルのヘッダーがあり、dmpmqlog コマンドにより必ず最初にフォーマットされます。 これには、次のようなフィールドがあります。

logactive	1 次ログ・エクステントの数
loginactive	2 次ログ・エクステントの数
logsize	エクステントあたり 4 KB ページ数
baselsn	ログのヘッドを含むログ・エクステントの中の最初の LSN
nextlsn	書き込まれる次のログ・レコードの LSN
headlsn	ログのヘッドでのログ・レコードの LSN
tailsn	ログのテール位置を識別する LSN
hflag1	ログが CIRCULAR であるか LOG RETAIN (リニア) であるかどうか
拡張ヘッダー *ID*	ログのヘッドを含むログ・エクステント

ログ・レコードのヘッダー
ログ内の各ログ・レコードには以下の情報を含む固定のヘッダーがあります。

LSN	ログの順序番号
LogRecdType	ログ・レコードのタイプ
XTranid	このログ・レコード (ある場合) と関連したトランザクション ID MQI の *TranType* は IBM MQ トランザクションのみを示しています。 XA の *TranType* は、他のリソース・マネージャーに関連するトランザクションを示しています。 同じ作業単位内に含まれる更新には同じ *XTranid* があります。
QueueName	このログ・レコード (ある場合) と関連したキュー
Qid	キュー用の固有の内部 ID
PrevLSN	同じトランザクション内の前のログ・レコード (ある場合) の LSN

キュー・マネージャーの始動
これは、キュー・マネージャーを始動した記録を取ります。

StartDate	キュー・マネージャーを始動した日付
StartTime	キュー・マネージャーを始動した時刻

キュー・マネージャーの停止
これは、キュー・マネージャーを停止した記録を取ります。

StopDate	キュー・マネージャーを停止した日付

StopTime キュー・マネージャーを停止した時刻

ForceFlag 使用した終了のタイプ

チェックポイントの開始
これはキュー・マネージャーのチェックポイントの開始を示しています。

チェックポイントの終了
これはキュー・マネージャーのチェックポイントの終了を示しています。

ChkPtLSN このチェックポイントを開始したログ・レコードの LSN

メッセージの書き込み
これはキューに書き込んだ持続メッセージの記録を取ります。 メッセージが同期点で書き込まれた場合、ログ・レコードのヘッダーには非ヌル文字 *XTranid* が入ります。 レコードの残りの部分には以下のものが入ります。

MapIndex キュー上のメッセージの ID。 これは、キューからこのメッセージを得るのに使用した該当する MQGET を 突き合わせる場合に使用できます。 この場合、同じ QueueName および MapIndex が含まれている次の *Get Message* ログ・レコードを検出できます。 この時点で、キューにメッセージを続けて 書き込むために MapIndex ID を再使用できます。

データ このログ・レコードには 16 進ダンプで、さまざまな内部データと、それに続くメッセージ記述子 (目印 MD) の表現と、それに続くメッセージ・データ本体が入っています。

部分の書き込み
単一のログ・レコードとしては長すぎる持続メッセージは、複数の *Put Part* ログ・レコードと、その後に続く単一の *Put Message* レコードとして記録されます。 複数の *Put Part* レコードがある場合は、*PrevLSN* フィールドによって、*Put Part* レコードと最後の *Put Message* レコードが連結されます。

データ 前のログ・レコードが終了した場所のメッセージ・データが続きます。

メッセージの読み取り
持続メッセージの読み取りのみが記録されます。 メッセージが同期点で読み取られた場合、ログ・レコードのヘッダーには非ヌル文字 *XTranid* が入ります。 レコードの残りの部分には以下のものが入ります。

MapIndex キューから得られたメッセージを識別します。 同じ *QueueName* および *MapIndex* を 含む最新の *Put Message* ログ・レコードは得られたメッセージを識別します。

QPriority キューから得られたメッセージの優先順位。

トランザクションの開始
新規トランザクションの開始を示しています。 MQI の TranType は IBM MQ トランザクションのみを示しています。 XA の TranType は、他のリソース・マネージャーに関連する トランザクションを示しています。 このトランザクションが行ったすべての更新には同じ *XTranid* があります。

トランザクションの準備
キュー・マネージャーは、指定した *XTranid* と関連し た更新をコミットする準備が整っていることを示しています。 このログ・レコードは、他のリソース・マネージャーに 関連する 2 フェーズ・コミットの一部として書き込まれます。

トランザクションのコミット
トランザクションが行ったすべての更新をキュー・マネージャーがコミットし たことを示しています。

トランザクションのロールバック
これは、キュー・マネージャーがトランザクションをロールバックしようとしていることを表しています。

トランザクションの終了
これはロールバック・トランザクションの終了を表しています。

トランザクション表
このレコードは同期点の最中に書き込まれます。 これは持続更新を行ったトランザクションそれぞれの状態を記録します。 トランザクションごとに、次の情報が記録されます。

XTranid	トランザクション ID。
FirstLSN	トランザクションと関連した最初のログ・レコードの LSN
LastLSN	トランザクションと関連した最後のログ・レコードの LSN

トランザクションに関連するプログラム
このログ・レコードは、キュー・マネージャーの XA トランザクション・マネージャーのコンポーネントによって書き込まれます。 これは、トランザクションに関連する外部リソース・マネージャーを記録します。 関連するプログラムごとに次の情報が記録されます。

RMName	リソース・マネージャーの名前
RMID	リソース・マネージャー ID。 これは後続の *Transaction Prepared* ログ・レコードにもログインします。このレコードはリソース・マネージャーが関連しているグローバル・トランザクションを記録するものです。
SwitchFile	このリソース・マネージャー用のスイッチ・ロード・ファイル
XAOpenString	このリソース・マネージャー用の XA オープン・ストリング
XACloseString	このリソース・マネージャー用の XA クローズ・ストリング

準備済みトランザクション
このログ・レコードは、キュー・マネージャーの XA トランザクション・マネージャーのコンポーネントによって書き込まれます。 指定したグローバル・トランザクションが正常に作成されたことを示しています。 関連するリソース・マネージャーのそれぞれにコミットが指示されます。 準備済みの各リソース・マネージャーの *RMID* は、ログ・レコードに記録されます。 キュー・マネージャー自体がトランザクションに関連している場合、*RMID* がゼロである *Participant Entry* が示されます。

忘れられたトランザクション
このログ・レコードは、キュー・マネージャーの XA トランザクション・マネージャーのコンポーネントによって書き込まれます。 コミット決定が各参加者に送信されると、 *Transaction Prepared* ログ・レコードの後に続きます。

キューの除去
これは、キュー上のすべてのメッセージが、例えば MQSC コマンド CLEAR QUEUE を使用して除去されたという事実の記録を取ります。

キューの属性
これは、キューの属性の初期化または変更の記録を取ります。

オブジェクトの作成
これは、IBM MQ オブジェクトの作成を記録します。

ObjName	作成したオブジェクトの名前
UserId	作成を実行するユーザー ID

オブジェクトの削除
これは、IBM MQ オブジェクトの削除を記録します。

ObjName	削除したオブジェクトの名前

IBM MQ キュー・マネージャー・データのバックアップと復元

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

このタスクについて

注意 : キュー・マネージャーを別のオペレーティング・システムに移動する場合は、十分に注意する必要があります。 詳細については、別のオペレーティング・システムへのキュー・マネージャーの移動を参照してください。

ハードウェア障害に起因する破壊からキュー・マネージャーを保護する対策を定期的に講じることができます。 キュー・マネージャーには、次に示す 3 種類の保護方法があります。

キュー・マネージャー・データのバックアップ

ハードウェア障害が発生した場合、キュー・マネージャーは強制的に停止されることがあります。 ハードウェア障害によってキュー・マネージャーのログ・データが失われた場合、キュー・マネージャーが再始動できないことがあります。 キュー・マネージャー・データをバックアップしておくと、失われたキュー・マネージャー・データの一部またはすべてを回復できる可能性があります。

一般に、ハードウェア障害が発生したために、リカバリー・ログの整合性が失われたとしても、キュー・マネージャー・データのバックアップの頻度を高くするほど、失われるデータ量は少なくなります。

キュー・マネージャー・データをバックアップするには、キュー・マネージャーが実行中であってはなりません。

キュー・マネージャー構成のみのバックアップ

ハードウェア障害が発生した場合、キュー・マネージャーは強制的に停止されることがあります。 ハードウェア障害によってキュー・マネージャー構成とログ・データの両方が失われた場合、キュー・マネージャーは再始動したり、ログからリカバリーしたりできなくなります。 キュー・マネージャー構成をバックアップした場合は、保存されている定義から、キュー・マネージャーとそのすべてのオブジェクトを再作成できます。

キュー・マネージャー構成をバックアップするには、キュー・マネージャーが実行中でなければなりません。

バックアップ・キュー・マネージャーの使用

重大なハードウェア障害が起きた場合は、キュー・マネージャーをリカバリーできない可能性があります。 この状態では、リカバリー不能なキュー・マネージャーに専用のバックアップ・キュー・マネージャーが存在すれば、バックアップ・キュー・マネージャーをリカバリー不能なキュー・マネージャーの代わりに活動化することができます。 定期的に更新していた場合は、バックアップ・キュー・マネージャーのログに、リカバリー不能なキュー・マネージャーからの最新の完全なログが含まれたログ・データがある可能性があります。

バックアップ・キュー・マネージャーは、既存のキュー・マネージャーの実行中に更新できます。

手順

- キュー・マネージャー・データをバックアップおよびリストアするには、以下を参照してください。
  - 672 ページの『キュー・マネージャー・データのバックアップ』.
  - 673 ページの『キュー・マネージャー・データの復元』.
- キュー・マネージャー構成をバックアップおよびリストアするには、以下を参照してください。
  - 673 ページの『キュー・マネージャー構成のバックアップ』
  - 674 ページの『キュー・マネージャー構成の復元』
- バックアップ・キュー・マネージャーを作成、更新、および始動するには、675 ページの『バックアップ・キュー・マネージャーの使用』を参照してください。

キュー・マネージャー・データのバックアップ

キュー・マネージャーのバックアップを行うと、ハードウェア・エラーによって生じる可能性のあるデータの消失から保護することができます。

始める前に

キュー・マネージャーのバックアップを開始する前に、キュー・マネージャーが実行されていないことを確認してください。 実行中のキュー・マネージャーのバックアップを取ろうとすると、ファイルがコピーされるときに更新が進行中であるため、バックアップが一貫していない可能性があります。 可能な場合は、**endmqm -w** コマンド (待機シャットダウン) を実行してキュー・マネージャーを停止します。 これが失敗した場合にのみ、 **endmqm -i** コマンドを使用します (即時シャットダウン)。

このタスクについて

キュー・マネージャーのデータのバックアップ・コピーを取るには、次のタスクを実行します。

手順

1. 構成ファイル中の情報を使用して、キュー・マネージャーのデータとログ・ファイルが置かれているディレクトリーを検索します。

   詳細については、84 ページの『Multiplatforms での .ini ファイルの IBM MQ 構成情報の変更』を参照してください。

   注 : IBM MQ を使用しているプラットフォームとの互換性を持たせるために、ディレクトリーに表示される名前が変換されています。 名前変換について詳しくは、IBM MQ ファイル名の理解を参照してください。
2. キュー・マネージャーのデータおよびログ・ファイルのディレクトリーすべてを、すべてのサブディレクトリーと一緒に、コピーしてください。

   必ずすべてのファイルをコピーしてください。 特に、640 ページの『ログの概要』で説明されているログ制御ファイルと、238 ページの『初期設定および構成ファイル』で説明されている構成ファイルのコピーを忘れないでください。 一部のディレクトリーは空である場合がありますが、後日バックアップを復元 する際にはすべて必要になります。

   循環ログでは、キュー・マネージャー・データとログ・ファイル・ディレクトリーを一緒にバックアップしてください。これにより、整合したキュー・マネージャー・データとログの集合を復元できます。

   リニア・ロギングの場合は、キュー・マネージャー・データとログ・ファイル・ディレクトリーを一緒にバックアップしてください。 キュー・マネージャーのデータ・ファイルに対応する一連の全ログ・ファイルが使用可能である場合に、キュー・マネージャーのデータ・ファイルのみを復元することが可能です。
3. ファイルの所有権も存続させます。

   Linux AIX IBM MQ for UNIX および Linux システムでは、**tar** コマンドを使用してこれを行うことができます。 (2 GB より大きいキューがある場合、**tar** コマンドは使用できません。 詳しくは、大規模キューの使用可能化を参照してください。

   注 : 新しいバージョンの IBM MQ にアップグレードする前に、 `qm.ini` ファイルとレジストリエントリのバックアップを取ってください。 キュー・マネージャーの情報は、 `qm.ini` ファイルに保存され、以前のバージョンに戻すために使用することができる。

関連タスク

キュー・マネージャーの停止

14 ページの『キュー・マネージャーの作成後の構成ファイルのバックアップ』

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

キュー・マネージャー・データの復元

キュー・マネージャーのデータのバックアップを復元するには、以下のステップに従います。

始める前に

バックアップを開始する前に、キュー・マネージャーが実行されていないことを確認してください。

クラスター内のキュー・マネージャーのバックアップをリストアする場合の詳細については、367 ページの『クラスター・キュー・マネージャーのリカバリー』 および「クラスタリング: 可用性、マルチインスタンス、および災害復旧」を参照してください。

注 : より新しいバージョンの IBM MQ にアップグレードする場合は、必ず **.ini** ファイルとレジストリー項目のバックアップを作成してください。 キュー・マネージャー情報は **.ini** ファイルに保管されており、以前のバージョンの IBM MQ に戻すために使用できます。

手順

1. 構成ファイル中の情報を使用して、キュー・マネージャーのデータとログ・ファイルが置かれているディレクトリーを探します。
2. バックアップ・データの格納先となるディレクトリーを空にしてください。
3. バックアップされたキュー・マネージャーのデータおよび ログ・ファイルを、正しい場所にコピーします。

   ログ・ファイルだけでなく、ログ制御ファイルもあることを確認してください。

   循環ログでは、キュー・マネージャー・データとログ・ファイル・ディレクトリーを一緒にバックアップしてください。これにより、整合したキュー・マネージャー・データとログの集合を復元できます。

   リニア・ロギングの場合は、キュー・マネージャー・データとログ・ファイル・ディレクトリーを一緒にバックアップしてください。 キュー・マネージャーのデータ・ファイルに対応する一連の全ログ・ファイルが使用可能である場合に、キュー・マネージャーのデータ・ファイルのみを復元することが可能です。
4. 構成情報ファイルを更新します。

   IBM MQ が復元データの検索にあたって正しい場所を調べられるように、IBM MQ およびキュー・マネージャーの構成ファイルが一貫しているかどうかも確認してください。
5. できあがったディレクトリー構造をチェックして、必要なディレクトリーがすべて 入っていることを確認してください。

   IBM MQ のディレクトリーとサブディレクトリーについて詳しくは、Windows システムでのディレクトリー構造および AIX and Linux システムでのディレクトリーの内容を参照してください。

タスクの結果

データが正しくバックアップされ復元されていれば、キュー・マネージャーは始動します。

Multi キュー・マネージャー構成のバックアップ

キュー・マネージャー構成をバックアップすると、ハードウェア障害が原因でキュー・マネージャー構成とログ・データの両方が失われ、キュー・マネージャーを再始動したりログからリカバリーしたりできない場合に、キュー・マネージャーをその定義から再構築することができます。

このタスクについて

ALW AIX, Linux, and Windows では、**dmpmqcfg** コマンドを使用して IBM MQ キュー・マネージャーの構成をダンプできます。

IBM i IBM i では、MQ 構成のダンプ (**DMPMQMCFG**) コマンドを使用して、キュー・マネージャーの構成オブジェクトと権限をダンプできます。

手順

1. キュー・マネージャーが実行中であることを確認してください。
2. 使用しているプラットフォームに応じて、以下のコマンドのいずれかを使用してキュー・マネージャー構成をバックアップします。

   - ALW AIX, Linux, and Windows の場合: MQ 構成のダンプ・コマンド **dmpmqcfg** を、デフォルトのフォーマット・オプションの (-f mqsc) MQSC、およびすべての属性 (-a) を指定して実行し、標準出力のリダイレクトを使用して、定義をファイルに保存します。 以下に例を示します。

     ```
     dmpmqcfg -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
     ```

   - IBM i IBM i の場合: MQ 構成のダンプ・コマンド (**DMPMQMCFG**) を、デフォルトのフォーマット・オプション OUTPUT(*MQSC) および EXPATTR(*ALL) を指定して実行し、TOFILE および TOMBR を使用して、定義を物理ファイル・メンバーに保存します。 以下に例を示します。

     ```
     DMPMQMCFG MQMNAME(MYQMGR) OUTPUT(*MQSC) EXPATTR(*ALL) TOFILE(QMQMSAMP/QMQSC)
     TOMBR(MYQMGRDEF)
     ```

関連タスク

674 ページの『キュー・マネージャー構成の復元』
まずキュー・マネージャーが稼働していることを確認してから、プラットフォームに適したコマンドを実行することによって、キュー・マネージャーの構成をバックアップから復元することができます。

関連資料

dmpmqcfg (キュー・マネージャー構成のダンプ)
MQ 構成のダンプ (DMPMQMCFG)

Multi キュー・マネージャー構成の復元

まずキュー・マネージャーが稼働していることを確認してから、プラットフォームに適したコマンドを実行することによって、キュー・マネージャーの構成をバックアップから復元することができます。

このタスクについて

ALW AIX, Linux, and Windows では、**runmqsc** コマンドを使用して IBM MQ キュー・マネージャーの構成を復元できます。

IBM i IBM i では、**STRMQMMQSC** コマンドを使用してキュー・マネージャーの構成オブジェクトおよび権限を復元できます。

手順

1. キュー・マネージャーが実行中であることを確認してください。

   データおよびログの損傷が、その他の方法ではリカバリー不能だった場合、キュー・マネージャーが再作成された可能性があることに注意してください。
2. 使用しているプラットフォームに応じて、以下のコマンドのいずれかを使用してキュー・マネージャーの構成を復元します。

   - ALW AIX, Linux, and Windows の場合: キュー・マネージャーに対して **runmqsc** を実行し、標準入力のリダイレクトを使用して、MQ 構成のダンプ (**dmpmqcfg**) コマンドによって生成されたスクリプト・ファイルから定義を復元します (673 ページの『キュー・マネージャー構成のバックアップ』を参照)。 以下に例を示します。

     ```
     runmqsc MYQMGR < /mq/backups/MYQMGR.mqsc
     ```

   - IBM i IBM i の場合: キュー・マネージャーに対して **STRMQMMQSC** を実行し、**SRCMBR** および **SRCFILE** パラメーターを使用して、MQ 構成のダンプ (**DMPMQMCFG**) コマンドによって生成された物

理ファイル・メンバーから定義を復元します (673 ページの『キュー・マネージャー構成のバックアップ』を参照)。 以下に例を示します。

```
STRMQMMQSC MQMNAME(MYQMGR) SRCFILE(QMQMSAMP/QMQSC) SRCMBR(MYQMGR)
```

関連タスク

673 ページの『キュー・マネージャー構成のバックアップ』
キュー・マネージャー構成をバックアップすると、ハードウェア障害が原因でキュー・マネージャー構成とログ・データの両方が失われ、キュー・マネージャーを再始動したりログからリカバリーしたりできない場合に、キュー・マネージャーをその定義から再構築することができます。

関連資料

dmpmqcfg (キュー・マネージャー構成のダンプ)
runmqsc (MQSC コマンドの実行)
MQ 構成のダンプ (DMPMQMCFG)
IBM MQ コマンドの開始 (STRMQMMQSC)

バックアップ・キュー・マネージャーの使用

既存のキュー・マネージャーは、専用のバックアップ・キュー・マネージャーが災害復旧のための有効な方法です

このタスクについて

バックアップ・キュー・マネージャーは、既存のキュー・マネージャーの非アクティブなコピーです。 既存のキュー・マネージャーが重大なハードウェア障害によってリカバリー不能になった場合、バックアップ・キュー・マネージャーをオンラインにして、リカバリー不能のキュー・マネージャーの代わりに使用することができます。

バックアップ・キュー・マネージャーが災害復旧のための有効な方法であり続けるには、既存のキュー・マネージャーのログ・ファイルを定期的にバックアップ・キュー・マネージャーにコピーしておく必要があります。 ログ・ファイルをコピーするために既存のキュー・マネージャーを停止する必要はありませんが、ログ・ファイルをコピーする必要があるのは、キュー・マネージャーがログ・ファイルへの書き込みを完了している場合のみです。特定のログ・ファイルが今後書き込まれないようにして安全にコピーできるようにする方法については、 676 ページの『バックアップ・キュー・マネージャーの更新』 を参照してください。

注 : 既存のキュー・マネージャー・ログは継続的に更新されるため、既存のキュー・マネージャー・ログとバックアップ・キュー・マネージャー・ログにコピーされたログ・データの間には常にわずかな矛盾が存在します。 バックアップ・キュー・マネージャーを定期的に更新することにより、この 2 つのログの間の矛盾を最小に抑えることができます。

バックアップ・キュー・マネージャーをオンラインにする必要がある場合は、必ず始動する前にアクティブ化してください。 バックアップ・キュー・マネージャーをアクティブ化してから始動するという必要条件は、バックアップ・キュー・マネージャーが偶発的に始動されないようにするための予防措置です。 バックアップ・キュー・マネージャーを活動化した後は、更新できなくなります。

重要 : 古いバックアップ・キュー・マネージャーが新しいアクティブ・キュー・マネージャーになると、理由が何であろうと、バックアップ・キュー・マネージャーはなくなります。 これは、実際には非同期複製の 1 つの形式であるため、新しいアクティブ・キュー・マネージャーは、論理的にいつかは古いアクティブ・キュー・マネージャーの後方に存在することになります。 したがって、古いアクティブ・キュー・マネージャーは、新しいアクティブ・キュー・マネージャーのバックアップとして機能しなくなります。

手順

- バックアップ・キュー・マネージャーの使用については、以下のトピックを参照してください。
  - 676 ページの『バックアップ・キュー・マネージャーの作成』
  - 676 ページの『バックアップ・キュー・マネージャーの更新』
  - 677 ページの『バックアップ・キュー・マネージャーの開始』

関連概念

639 ページの『ロギング: メッセージが失われないようにするための機能』

IBM MQ は、キュー・マネージャーが制御する持続データに対する重要な変更をすべてリカバリー・ログに記録します。

バックアップ・キュー・マネージャーの作成

既存のキュー・マネージャーの非アクティブなコピーとしてバックアップ・キュー・マネージャーを作成します。

このタスクについて

重要 : バックアップ・キュー・マネージャーは、リニア・ロギングの使用時にのみ使用できます。

バックアップ・キュー・マネージャーは、以下の条件を満たすことが必要です。

- 既存のキュー・マネージャーと同じ属性 (キュー・マネージャー名、ロギング・タイプ、ログ・ファイル・サイズなど) がある。
- 既存のキュー・マネージャーと同じプラットフォーム上に存在する。
- コード・レベルが既存のキュー・マネージャーと同等以上である。

手順

1. 制御コマンド **crtmqm** を使用して、既存のキュー・マネージャーのバックアップ・キュー・マネージャーを作成します。
2. 672 ページの『キュー・マネージャー・データのバックアップ』の説明に従って、すべての既存のキュー・マネージャーのデータおよびログ・ファイル・ディレクトリー (すべてのサブディレクトリーを含む) をコピーします。
3. バックアップ・キュー・マネージャーのデータおよびログ・ファイル・ディレクトリー (すべてのサブディレクトリーを含む) を、既存のキュー・マネージャーからのコピーで上書きします。
4. 以下の例に示すように、バックアップ・キュー・マネージャー上で **strmqm** 制御コマンドを実行します。

   ```
   strmqm -r BackupQMName
   ```

   このコマンドにより、キュー・マネージャーが IBM MQ 内のバックアップ・キュー・マネージャーであることをフラグで示し、コピーされたすべてのログ・エクステントを再生して、バックアップ・キュー・マネージャーを既存のキュー・マネージャーと一致させます。

関連資料

crtmqm (キュー・マネージャーの作成)
strmqm (キュー・マネージャーの始動)

バックアップ・キュー・マネージャーの更新

バックアップ・キュー・マネージャーが災害復旧のための有効な方法であり続けるには、バックアップ・キュー・マネージャーを定期的に更新しておく必要があります。

このタスクについて

定期的な更新によって、バックアップ・キュー・マネージャー・ログと現在のキュー・マネージャー・ログの間の矛盾を少なくすることができます。 バックアップを行う前に、キュー・マネージャーを停止する必要はありません。

警告 : バックアップ・キュー・マネージャーのログ・ディレクトリー に、不連続なログのセットをコピーした場合、欠落した最初のログが見つかるポイントまでのログが再生されます。

手順

1. バックアップを行うキュー・マネージャーで、以下のスクリプト (MQSC) コマンドを発行します。

```
RESET QMGR TYPE(ADVANCELOG)
```

これにより、現在のログへの書き込みが停止され、キュー・マネージャー・ロギングが次のログ・エクステントに拡張されます。 これにより、現在時刻までに記録されたすべての情報が確実にバックアップされます。

2. バックアップを行うキュー・マネージャーで以下のスクリプト (MQSC) コマンドを発行することにより、(新しい) 現在のアクティブ・ログ・エクステント番号を入手します。

```
DIS QMSTATUS CURRLOG
```

3. 更新されたログ・エクステント・ファイルを、現在のキュー・マネージャー・ログ・ディレクトリーからバックアップ・キュー・マネージャー・ログ・ディレクトリーにコピーします。

   最終更新から 677 ページの『2』 で示されている現在のエクステントまで (ただし、現在のエクステント自体は含まない) のすべてのログ・エクステントをコピーします。 ログ・エクステント・ファイルのみをコピーします(「S..」で始まる)。

4. 以下の例に示すように、バックアップ・キュー・マネージャー上で **strmqm** 制御コマンドを実行します。

```
strmqm -r BackupQMName
```

   これにより、コピーされたすべてのログ・エクステントが再生され、バックアップ・キュー・マネージャーをキュー・マネージャーと一致させます。 再生が完了すると、再始動リカバリーに必要なすべてのログ・エクステント、およびメディア・リカバリーに必要なすべてのログ・エクステントを示すメッセージを受け取ります。

関連資料

RESET QMGR

DISPLAY QMSTATUS

strmqm (キュー・マネージャーの始動)

バックアップ・キュー・マネージャーの開始

リカバリー不能なキュー・マネージャーのバックアップ・キュー・マネージャーを置き換えることができます。

このタスクについて

クラスター内のキュー・マネージャーのバックアップをリストアする場合の詳細については、 367 ページの『クラスター・キュー・マネージャーのリカバリー』 および クラスター化: 可用性、マルチインスタンス、および災害復旧 を参照してください。

リカバリー不能なキュー・マネージャーに専用のバックアップ・キュー・マネージャーが存在すれば、バックアップ・キュー・マネージャーをリカバリー不能なキュー・マネージャーの代わりにアクティブ化することができます。

リカバリー不能のキュー・マネージャーがバックアップ・キュー・マネージャーに置き換えられた場合、リカバリー不能のキュー・マネージャーの一部のキュー・マネージャー・データが失われることがあります。 失われるデータの量は、バックアップ・キュー・マネージャーの最終更新時によって異なります。 最終更新時が現在により近い時点であるほど、失われるキュー・マネージャー・データは少なくなります。

注 : キュー・マネージャー・データおよびログ・ファイルが異なるディレクトリーに保持されている場合でも、それらのディレクトリーのバックアップおよび復元を同時に行ってください。 キュー・マネージャーのデータとログ・ファイルの経過日数が異なっていると、キュー・マネージャーは有効な状態ではなく、おそらく始動しません。 始動する場合でも、データは破損している可能性があります。

手順

1. 以下の例に示すように、**strmqm** 制御コマンドを実行してバックアップ・キュー・マネージャーをアクティブ化します。

```
strmqm -a BackupQMName
```

バックアップ・キュー・マネージャーがアクティブ化されます。 これでアクティブになり、バックアップ・キュー・マネージャーは更新できなくなります。

2. 以下の例に示すように、**strmqm** 制御コマンドを実行してバックアップ・キュー・マネージャーを始動します。

```
strmqm BackupQMName
```

IBM MQ は、これを再始動リカバリーと見なし、バックアップ・キュー・マネージャーのログを使用します。 バックアップ・キュー・マネージャーへの最後の更新時に、再生が行われているため、最後に記録されたチェックポイントからのアクティブ・トランザクションのみがロールバックされます。

3. すべてのチャネルを再開します。
4. できあがったディレクトリー構造をチェックして、必要なディレクトリーがすべて 入っていることを確認してください。

IBM MQ のディレクトリーとサブディレクトリーの詳細については、ファイル・システム・サポートの計画を参照してください。

5. ログ・ファイルだけでなく、ログ制御ファイルもあることを確認してください。 また、IBM MQ が復元データの検索にあたって正しい場所を調べられるように、IBM MQ およびキュー・マネージャーの構成ファイルが一貫しているかどうかも確認してください。

タスクの結果

データが正しくバックアップされ復元されていれば、キュー・マネージャーは始動します。

関連タスク

231 ページの『停止したチャネルの再始動』

チャネルが STOPPED 状態になるときは、手動でチャネルを再始動しなければなりません。

関連資料

strmqm (キュー・マネージャーの始動)

クラスター・エラー・リカバリーの変更 (z/OS 以外のサーバー)

キュー・マネージャーは、問題が解決されるまで、問題の原因となった操作を再実行します。 5 日後になっても問題が解決していない場合、キュー・マネージャーはシャットダウンして、キャッシュがそれ以上古くならないようにします。

キュー・マネージャーは、問題が解決されるまで、問題の原因となった操作を再実行します。 5 日後になっても問題が解決していない場合、キュー・マネージャーはシャットダウンして、キャッシュがそれ以上古くならないようにします。 キャッシュは古くなるにつれて、より多くの問題を引き起こします。 クラスター・エラーに関するこの動作は、 z/OS には適用されません。

クラスター管理のすべての側面は、ローカル・リポジトリー・マネージャー・プロセス `amqrrmfa` によって、キュー・マネージャーのために処理されます。 このプロセスは、クラスター定義がない場合を含め、すべてのキュー・マネージャーで実行されます。

IBM MQ は、リポジトリー・マネージャーを停止して、リポジトリー・マネージャーなしで続行するのではなく、失敗した操作を再実行します。 キュー・マネージャーがリポジトリー・マネージャーの問題を検出すると、2 とおりある一連のアクションのいずれか 1 つを実行します。

1. キュー・マネージャーの操作を阻害しないエラーの場合、キュー・マネージャーはエラー・ログにメッセージを書き込みます。 操作が成功するまで、失敗した操作を 10 分ごとに再実行します。 デフォルトでは、エラーに対処するための時間は 5 日間です。この間に対処できなかった場合、キュー・マネージャーはエラー・ログにメッセージを書き込み、シャットダウンします。 5 日後のシャットダウンは延期できます。
2. キュー・マネージャーの操作を阻害するエラーの場合、キュー・マネージャーはエラー・ログにメッセージを書き込み、直ちにシャットダウンします。

キュー・マネージャーの操作を阻害するエラーとは、キュー・マネージャーがそれまでに診断できなかったエラー、または予測不能な結果をもたらす可能性のあるエラーです。 このタイプのエラーの場合、キュ

ー・マネージャーは通常、結果的に FFST ファイルを作成します。 キュー・マネージャーの操作を阻害するエラーは、IBM MQ のバグによって引き起こされるか、あるいは管理者またはプログラムが IBM MQ プロセスを終了するなどの予期しない操作を行うことによって引き起こされます。

エラー・リカバリーの振る舞いを変更する際のポイントは、矛盾するクラスター定義の数が増えて行く中で、キュー・マネージャーが実行を継続する時間を制限することです。 クラスター定義の矛盾が増えるに連れて、アプリケーションが異常な振る舞いをする機会も増えることになります。

5 日後にキュー・マネージャーをシャットダウンするというデフォルトの選択は、矛盾の数を抑えることと、問題が検出され、解決されるまで、キュー・マネージャーを使用可能な状態に維持することのバランスを取った結果です。

このキュー・マネージャーがシャットダウンされるまでの時間は、問題を修正するか、キュー・マネージャーが予定どおりにシャットダウンされるまでの間、無期限に延長することができます。 5 日間実行を継続することで、長い週末が終わるまでキュー・マネージャーの実行状態を維持し、あらゆる問題に対処する時間を稼ぐことや、キュー・マネージャーが再始動するまでの時間を延長することができます。

修正処置

クラスター・エラー・リカバリーの問題に対する処置は選択できます。 1 番目の選択肢は問題をモニターして修正し、2 番目の選択肢は問題をモニターして問題の修正を延期することです。

1. キュー・マネージャーのエラー・ログで、エラー・メッセージ AMQ9448 および AMQ5008 をモニターし、問題を修正します。

   AMQ9448 は、コマンドの実行後にリポジトリー・マネージャーがエラーを返したことを示します。 このエラーは、10 分ごとのコマンド再試行の開始時にマークを付け、シャットダウンが延期されなければ、最終的には 5 日後にキュー・マネージャーを停止します。
   AMQ5008 は、IBM MQ プロセスの欠落が原因で、キュー・マネージャーが停止したことを示します。 5 日後にリポジトリー・マネージャーが停止した結果、AMQ5008 になります。 リポジトリー・マネージャーが停止すると、キュー・マネージャーも停止します。
2. キュー・マネージャーのエラー・ログで、エラー・メッセージ AMQ9448 をモニターし、問題の修正を延期します。

   SYSTEM.CLUSTER.COMMAND.QUEUE からのメッセージの取得を使用不可にすると、リポジトリー・マネージャーはコマンド実行の試行を止めて、処理を行うことなく無期限に継続します。 ただし、リポジトリー・マネージャーがキューに対して保持しているハンドルはすべて解放されます。 リポジトリー・マネージャーが停止されないため、5 日後になってもキュー・マネージャーは停止されません。
   MQSC コマンドを実行して、SYSTEM.CLUSTER.COMMAND.QUEUE からのメッセージの取得を使用不可にします。
   ```
   ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(DISABLED)
   ```
   SYSTEM.CLUSTER.COMMAND.QUEUE からのメッセージの受信を再開するには、以下の MQSC コマンドを実行します。
   ```
   ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
   ```

特別な考慮事項

IBM MQ で amqrrmfa を停止すると、キュー・マネージャーは停止します。これは、キュー・マネージャーの障害と見なされるためです。 キュー・マネージャーのチューニング・パラメーター TolerateRepositoryFailure を設定しない限り、 amqrrmfa プロセスを停止してはなりません。

例

```
TuningParameters:
        TolerateRepositoryFailure=TRUE
```

図 *86. qm.ini* で *TolerateRepositoryFailure* を *TRUE* に設定する

関連概念

98 ページの『キュー・マネージャー構成ファイル qm.ini』
キュー・マネージャー構成ファイル qm.ini には、特定のキュー・マネージャーに関連する情報が含まれています。 個々のキュー・マネージャーの構成を変更するために使用できる属性によって、IBM MQ の設定がオーバーライドされます。

JMS および Jakarta Messaging リソースの構成

JMS または Jakarta Messaging アプリケーションが、 IBM MQ に接続するために必要なリソースを作成および構成し、メッセージを送受信するための宛先にアクセスする方法の 1 つは、Java Naming and Directory Interface (JNDI) を使用して、JNDI 名前空間と呼ばれるネーミングおよびディレクトリー・サービス内の場所から管理対象オブジェクトを取得する方法です。 JMS アプリケーションで管理対象オブジェクトを JNDI 名前空間から取り出すには、その前にまず管理対象オブジェクトを作成して構成する必要があります。

このタスクについて

JM 3.0 V 9.3.0 V 9.3.0 IBM MQ 9.3.0、、新しいアプリケーションの開発がサポートされている。 Jakarta Messaging 3.0 IBM MQ 9.3.0 既存のアプリケーションについては、 を引き続きサポートする。 JMS 2.0 Jakarta Messaging 3.0 API と JMS 2.0 API の両方を同じアプリケーションで使用することはサポートされていません。 詳細については、 JMS/Jakarta Messaging の IBM MQ クラスの使用を参照してください。

以下のいずれかのツールを使用して、IBM MQ で管理対象オブジェクトを作成して構成できます。

IBM MQ JMS および Jakarta Messaging 管理ツール

IBM MQ JMS 管理ツール、 **JMSAdmin**、および Jakarta Messaging 管理ツール、 **JMS30Admin** は、LDAP、ローカル・ファイル・システム、またはその他の場所に保管される IBM MQ JMS オブジェクトおよび Jakarta Messaging オブジェクトを作成および構成するために使用できるコマンド行ツールです。 JMS および Jakarta Messaging 管理ツールは、 **runmqsc** に似た構文を使用し、スクリプトもサポートします。

管理ツールは、構成ファイルを使用して特定のプロパティーの値を設定します。 サンプル構成ファイルが提供されています。開始する前に、このファイルを管理ツールを使用してご使用のシステムに合わせて編集し、JMS リソースを構成できます。 構成ファイルについて詳しくは、688 ページの『JMSAdmin および JMS30Admin ツールの構成』を参照してください。

JMS 2.0 **IBM MQ Explorer**

JMS 2.0 の場合、 IBM MQ Explorer を使用して、LDAP、ローカル・ファイル・システム、またはその他の場所に保管される JMS 2.0 オブジェクト定義を作成および管理できます。

JM 3.0 V 9.3.0 V 9.3.0 Jakarta Messaging 3.0 の場合、 を使用して JNDI を管理することはできません。 IBM MQ Explorer JNDI 管理は、 **JMSAdmin** の Jakarta Messaging 3.0 変種、 **JMS30Admin** でサポートされている。

WebSphere Application Server にデプロイされた IBM MQ JMS アプリケーションは、アプリケーション・サーバー JNDI リポジトリーから JMS オブジェクトにアクセスする必要があります。 したがって、WebSphere Application Server と IBM MQ の間で JMS メッセージングを使用する場合は、IBM MQ で作成するオブジェクトに対応するオブジェクトを WebSphere Application Server で作成する必要があります。

JM 3.0 V 9.3.0 V 9.3.0 IBM MQ 9.3 は Jakarta Messaging 3.0 をサポートしていますが、WebSphere Application Server には現在同等のサポートはありません。 したがって、 WebSphere Application Server では、 Java Message Service 2.0 リソースを構成します。

IBM MQ Explorer および IBM MQ JMS 管理ツールを使用して、 WebSphere Application Server に保管されている IBM MQ JMS オブジェクトを管理することはできません。 その代わりに、以下のいずれかのツールを使用して、WebSphere Application Server の管理対象オブジェクトを作成して構成できます。

WebSphere Application Server 管理コンソール
WebSphere Application Server 管理コンソールは、 WebSphere Application Server で IBM MQ JMS オブジェクトを管理するために使用できる Web ベースのツールです。

WebSphere Application Server wsadmin スクリプト・クライアント
WebSphere Application Server wsadmin スクリプト・クライアントは、WebSphere Application Server 内の IBM MQ JMS オブジェクトを管理するための特殊なコマンドを提供します。

JMS アプリケーションを使用して WebSphere Application Server 内から IBM MQ キュー・マネージャーのリソースにアクセスする場合は、 IBM MQ classes for JMS のバージョンが含まれている WebSphere Application Server の IBM MQ メッセージング・プロバイダーを使用する必要があります。 WebSphere Application Server に付属の IBM MQ リソース・アダプターは、 IBM MQ メッセージング・プロバイダーとの JMS メッセージングを実行するすべてのアプリケーションで使用されます。 通常、IBM MQ リソース・アダプターは、WebSphere Application Server フィックスパックが適用されると自動的に更新されますが、以前にリソース・アダプターを手動で更新した場合は、保守が正しく適用されるように構成を手動で更新する必要があります。

関連概念
IBM MQ classes for JMS アプリケーションでの接続ファクトリーおよび宛先の作成と構成
関連資料
runmqsc (MQSC コマンドの実行)

JNDI ネーム・スペースでの接続ファクトリーおよび宛先の構成

JMS および Jakarta Messaging アプリケーションは、 Java Naming and Directory Interface (JNDI) を介して、ネーミングおよびディレクトリー・サービス内の管理対象オブジェクトにアクセスします。 JMS または Jakarta Messaging 管理対象オブジェクトは、JNDI 名前空間と呼ばれるネーミングおよびディレクトリー・サービス内の場所に保管されます。 JMS または Jakarta Messaging アプリケーションは、メッセージを送受信するために、管理対象オブジェクトを検索して IBM MQ に接続し、宛先にアクセスすることができます。

このタスクについて

JMS または Jakarta Messaging アプリケーションは、コンテキストを使用して、ネーミングおよびディレクトリー・サービスで JMS または Jakarta Messaging オブジェクトの名前を検索します。

初期コンテキスト
JNDI 名前空間のルートを定義する初期コンテキスト。 ネーミングおよびディレクトリー・サービスのロケーションごとに、初期コンテキストを指定して、JMS または Jakarta Messaging アプリケーションがネーミングおよびディレクトリー・サービスのそのロケーションにある管理対象オブジェクトの名前を解決できる開始点を指定する必要があります。

サブコンテキスト
1 つのコンテキストには 1 つ以上のサブコンテキストを含めることができます。 サブコンテキストは JNDI 名前空間のサブディビジョンであり、接続ファクトリーや宛先などの管理対象オブジェクトやその他のサブコンテキストを含めることができます。 サブコンテキストはそれ自体がオブジェクトではなく、サブコンテキストにあるオブジェクトの命名規則を拡張したものにすぎません。

IBM MQ classes for JMS または IBM MQ classes for Jakarta Messaging アプリケーションが JNDI 名前空間から管理対象オブジェクトを取得できるようにするには、まず管理対象オブジェクトを作成する必要があります。 以下のタイプの JMS または Jakarta Messaging オブジェクトを作成および構成できます。

接続ファクトリー

JMS または Jakarta Messaging 接続ファクトリー・オブジェクトは、接続用の一連の標準構成プロパティーを定義します。 JMS または Jakarta Messaging アプリケーションは、接続ファクトリーを使用して IBM MQ への接続を作成します。 Point-to-Point メッセージ・ドメインとパブリッシュ/サブスクライブ・メッセージ・ドメインという、2 つのメッセージ・ドメインのいずれかに固有の接続ファクトリーを作成できます。

あるいは、JMS 1.1 以降、Point-to-Point およびパブリッシュ/サブスクライブ両方のメッセージングで使用できる、ドメイン非依存型接続ファクトリーを作成することができます。 詳しくは、 JMS および Jakarta Messaging モデルを参照してください。

Destination

JMS または Jakarta Messaging 宛先は、クライアントが作成するメッセージのターゲット、および JMS アプリケーションがコンシュームするメッセージのソースを表すオブジェクトです。 JMS または Jakarta Messaging アプリケーションは、単一の宛先オブジェクトを使用してメッセージの書き込みとメッセージの取得を行うことも、別個の宛先オブジェクトを使用することもできます。 以下の 2 つのタイプの宛先オブジェクトがあります。

- Point-to-Point メッセージングで使用される JMS または Jakarta Messaging キュー宛先
- パブリッシュ/サブスクライブ・メッセージングで使用される JMS または Jakarta Messaging トピック宛先

JMS 2.0 JMS 2.0 の場合、 IBM MQ Explorer または IBM MQ JMS 管理ツール **JMSAdmin** を使用して、コンテキストおよび管理対象オブジェクトを作成できます。

注 : JM 3.0 V 9.3.0 V 9.3.0 Jakarta Messaging 3.0 の場合、を使用して JNDI を管理することはできません。 IBM MQ Explorer JNDI 管理は、 **JMSAdmin** の Jakarta Messaging 3.0 変種、**JMS30Admin** でサポートされている。

以下の図は、IBM MQ JNDI 名前空間に作成された JMS オブジェクトまたは Jakarta Messaging オブジェクトの例を示しています。

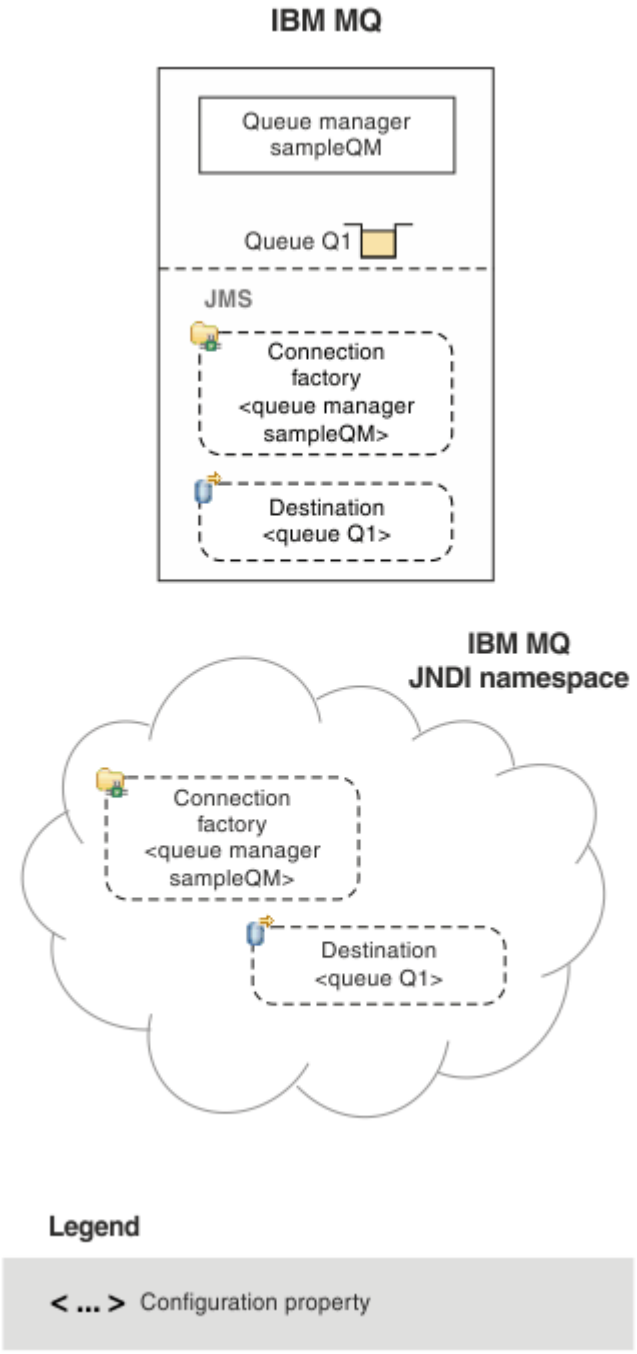


図 *87. IBM MQ* で作成された *JMS* オブジェクトまたは *Jakarta Messaging* オブジェクト

WebSphere Application Server と IBM MQ の間で JMS メッセージングを使用する場合は、IBM MQ との通信に使用する、対応するオブジェクトを WebSphere Application Server 内に作成する必要があります。 これらのオブジェクトのいずれかを WebSphere Application Server で作成すると、以下の図に示すように、それは WebSphere Application Server JNDI 名前空間に保管されます。

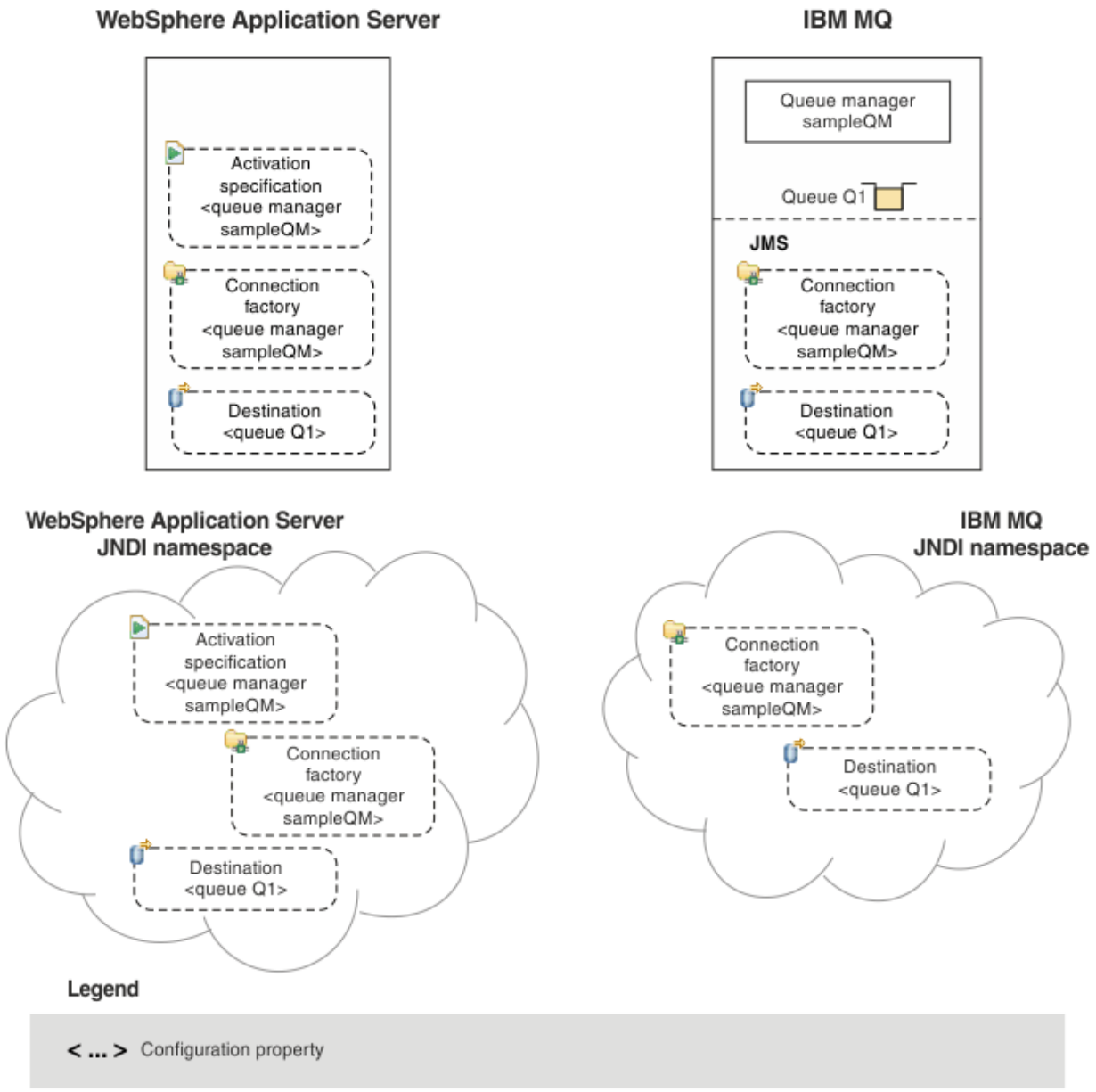


図 *88. WebSphere Application Server* で作成されるオブジェクトと、対応する *IBM MQ* のオブジェクト

JM 3.0 V 9.3.0 V 9.3.0 IBM MQ 9.3 は Jakarta Messaging 3.0 をサポートしていますが、WebSphere Application Server には現在同等のサポートはありません。 したがって、 WebSphere Application Server では、 Java Message Service 2.0 リソースを構成します。

アプリケーションがメッセージ駆動型 Bean (MDB) を使用する場合、接続ファクトリーはアウトバウンド・メッセージにのみ使用され、インバウンド・メッセージはアクティベーション・スペックによって受信されます。 活動化仕様は、Java EE Connector Architecture 1.5 (JCA 1.5) 規格の一部です。 JCA 1.5 には、IBM MQ などの JMS プロバイダーを WebSphere Application Server などの Java EE アプリケーション・サーバーと統合するための標準的な方法が用意されています。 JMS アクティベーション・スペックは、1 つ以上のメッセージ駆動型 Bean (MDB) と関連付けることが可能で、これらの MDB が宛先に到達するメッセージを listen するのに必要な構成を提供します。

WebSphere Application Server 管理コンソールまたは wsadmin スクリプト・コマンドを使用して、必要な JMS リソースを作成および構成できます。

手順

- JMS 2.0
  IBM MQ Explorer を使用して IBM MQ 用に JMS オブジェクトを構成するには、685 ページの『IBM MQ Explorer を使用した JMS 2.0 オブジェクトの構成』を参照してください。
- JMS 2.0
  IBM MQ JMS 管理ツール **JMSAdmin** を使用して IBM MQ の JMS オブジェクトを構成するには、686 ページの『管理ツールを使用した JMS および Jakarta Messaging オブジェクトの構成』を参照してください。
- JM 3.0 V 9.3.0 V 9.3.0
  IBM MQ Jakarta Messaging 管理ツール **JMS30Admin** を使用して IBM MQ の Jakarta Messaging オブジェクトを構成するには、686 ページの『管理ツールを使用した JMS および Jakarta Messaging オブジェクトの構成』を参照してください。
- JMS 2.0
  WebSphere Application Server の JMS オブジェクトを構成するには、696 ページの『WebSphere Application Server での JMS 2.0 リソースの構成』を参照してください。

タスクの結果

IBM MQ classes for JMS または IBM MQ classes for Jakarta Messaging アプリケーションは、JNDI 名前空間から管理対象オブジェクトを取得できます。また、必要に応じて、IBM JMS 拡張または IBM MQ JMS 拡張のいずれかを使用して、1 つ以上のプロパティーを設定または変更できます。

関連タスク

JNDI を使用して JMS アプリケーションで管理対象オブジェクトを取り出す
IBM MQ classes for JMS アプリケーションでの接続ファクトリーおよび宛先の作成と構成

JMS 2.0 IBM MQ Explorer を使用した JMS 2.0 オブジェクトの構成

IBM MQ Explorer のグラフィカル・ユーザー・インターフェースを使用すると、JMS オブジェクトを IBM MQ オブジェクトからしたり、IBM MQ オブジェクトを JMS オブジェクトから作成したり、他の IBM MQ オブジェクトを管理およびモニターしたりできます。

このタスクについて

JMS 2.0 IBM MQ Explorer は、IBM MQ オブジェクトがローカル・コンピューターによってホストされるか、 リモート・システム上にあるかに関係なく、それらのオブジェクトを管理およびモニターできるグラフィカル・ユーザー・インターフェースです。 IBM MQ Explorer は、 Windows および Linux for x86-64 上で稼働します。 サポートされるプラットフォーム (z/OS を含む) で稼働中のキュー・マネージャーにリモートで接続することができるので、コンソールから、メッセージング・バックボーン全体を表示、探索、および変更することができます。

注 : JM 3.0 V 9.3.0 V 9.3.0 Jakarta Messaging 3.0 の場合、を使用して JNDI を管理することはできません。 IBM MQ Explorer JNDI 管理は、 **JMSAdmin** の Jakarta Messaging 3.0 変種、**JMS30Admin** でサポートされている。

IBM MQ Explorer では、すべての接続ファクトリーが適切なコンテキストおよびサブコンテキストの「接続ファクトリー」フォルダーに格納されます。

以下のタイプのタスクは、IBM MQ Explorer を使用して、IBM MQ Explorer 内の既存のオブジェクトからコンテキストにより、または新規オブジェクト作成ウィザード内から実行できます。

- 以下のいずれかの IBM MQ オブジェクトから JMS 接続ファクトリーを作成します。
  - IBM MQ キュー・マネージャー (ローカル・コンピューターまたはリモート・システムのどちらでも)。
  - IBM MQ チャネル。
  - IBM MQ リスナー。

- JMS 接続ファクトリーを使用して、 IBM MQ キュー・マネージャーを IBM MQ Explorer に追加します。
- JMS キューを IBM MQ キューから作成します。
- IBM MQ キューを JMS キューから作成します。
- JMS トピックを IBM MQ トピックから作成します。 これは、IBM MQ オブジェクトまたは動的トピックにすることができます。
- IBM MQ トピックを JMS トピックから作成します。

手順

- IBM MQ Explorer を開始します (まだ実行されていない場合)。

  IBM MQ Explorer が実行されていて、ウェルカム・ページが表示されている場合は、ウェルカム・ページを閉じて IBM MQ オブジェクトの管理を開始します。
- まだ行っていない場合は、JMS オブジェクトが保管される JNDI 名前空間のルートを定義する初期コンテキストを、名前指定およびディレクトリー・サービスに作成します。

  初期コンテキストを IBM MQ Explorer に追加したら、接続ファクトリー・オブジェクト、宛先オブジェクト、およびサブコンテキストを JNDI 名前空間に作成できます。

  ナビゲーター・ビューの JMS 管理対象オブジェクト・フォルダーに、初期コンテキストが表示されます。 JNDI 名前空間の内容がすべて表示されますが、IBM MQ Explorer で編集できるオブジェクトは、そこに格納されている IBM MQ classes for JMS オブジェクトに限られます。 詳しくは、初期コンテキストの追加を参照してください。
- 必要なサブコンテキストおよび JMS 管理対象オブジェクトを作成して構成します。

  詳しくは、JMS 管理対象オブジェクトの作成および構成を参照してください。
- IBM MQ を構成します。

  詳しくは、 IBM MQ Explorer を使用した IBM MQ の構成 を参照してください。

関連概念

IBM MQ Explorer の概要

IBM MQ classes for JMS アプリケーションでの接続ファクトリーおよび宛先の作成と構成

管理ツールを使用した JMS および Jakarta Messaging オブジェクトの構成

IBM MQ には、8 つのタイプの IBM MQ classes for JMS または IBM MQ classes for Jakarta Messaging オブジェクトのプロパティーを定義し、それらを JNDI 名前空間内に保管するために使用できる管理ツールが用意されています。 アプリケーションは、JNDI を使用して、これらの管理対象オブジェクトをネーム・スペースから取り出すことができます。

このタスクについて

JMS 2.0 JMS 2.0 の場合、JNDI 管理は **JMSAdmin** ツールによってサポートされます。

JM 3.0 V 9.3.0 V 9.3.0 Jakarta Messaging 3.0 の場合、JNDI 管理は **JMSAdmin** の Jakarta Messaging 3.0 バリアント (**JMS30Admin**) によってサポートされます。

次の表に、動詞を使用して作成、構成、操作できる管理対象オブジェクトの 8 つのタイプを示します。「キーワード」列には、687 ページの表 36 に示されているコマンドで *TYPE* の代わりに使用できるストリングが表示されます。

表 36. 管理ツールによって処理される *JMS* および *Jakarta Messaging* オブジェクト・タイプ

オブジェクト・タイプ	キーワード	説明
MQConnectionFactory	CF	IBM MQ の JMS ConnectionFactory インターフェースの実装。 これは、Point-to-Point ドメインおよびパブリッシュ/サブスクライブ・ドメインの両方に接続を作成するためのファクトリー・オブジェクトを表します。
MQQueueConnectionFactory	QCF	JMS QueueConnection ファクトリー・インターフェースの IBM MQ 実装。 これは、Point-to-Point ドメインに接続を作成するためのファクトリー・オブジェクトを表します。
MQTopicConnectionFactory	TCF	JMS TopicConnection ファクトリー・インターフェースの IBM MQ 実装。 これは、パブリッシュ/サブスクライブ・ドメインに接続を作成するためのファクトリー・オブジェクトを表します。
MQQueue	Q	JMS キュー・インターフェースの IBM MQ 実装。 これは、Point-to-Point ドメインにおけるメッセージの宛先を表します。
MQTopic	T	JMS トピック・インターフェースの IBM MQ 実装。 これは、パブリッシュ/サブスクライブ・ドメインにおけるメッセージの宛先を表します。
MQXAConnectionFactory 687 ページの『1』	XACF	JMS XAConnectionFactory インターフェースの IBM MQ の実装。 これは、XA バージョンの JMS クラスを使用する Point-to-Point ドメインおよびパブリッシュ/サブスクライブ・ドメインの両方で接続を作成するためのファクトリー・オブジェクトを表します。
MQXAQueueConnection ファクトリー 687 ページの『1』	XAQCF	JMS XAQueueConnection ファクトリー・インターフェースの IBM MQ 実装。 これは、XA バージョンの JMS クラスを使用する Point-to-Point ドメインに接続を作成するためのファクトリー・オブジェクトを表します。
MQXATopicConnection ファクトリー 687 ページの『1』	XATCF	JMS XATopicConnection ファクトリー・インターフェースの IBM MQ 実装。 これは、XA バージョンの JMS クラスを使用するパブリッシュ/サブスクライブ・ドメインに接続を作成するためのファクトリー・オブジェクトを表します。

注 :

1. これらのクラスは、アプリケーション・サーバーのベンダーが使用するために提供されたものです。 アプリケーション・プログラマーには直接的には役立たないと考えられます。

これらのオブジェクトの構成方法について詳しくは、695 ページの『JMS オブジェクトの構成』を参照してください。

このツールを使用する必要のあるプロパティー・タイプと値は、IBM MQ classes for JMS オブジェクトのプロパティーにリストされています。

また、692 ページの『サブコンテキストの構成』の説明のように、このツールを使用して JNDI 内のディレクトリー名前空間のサブコンテキストを操作することもできます。

JMS 2.0 JMS 2.0 以前の場合は、 IBM MQ Explorer を使用して IBM MQ classes for JMS 管理対象オブジェクトを作成および構成することもできます。

JM 3.0 V 9.3.0 V 9.3.0 Jakarta Messaging 3.0 の場合、を使用して JNDI を管理することはできません。 IBM MQ Explorer JNDI 管理は、 **JMSAdmin** の Jakarta Messaging 3.0 変種、 **JMS30Admin** でサポートされている。

関連概念

IBM MQ classes for JMS アプリケーションでの接続ファクトリーおよび宛先の作成と構成

JNDI を使用して JMS アプリケーションで管理対象オブジェクトを取り出す

JMSAdmin および JMS30Admin ツールの構成

IBM MQ JMS および Jakarta Messaging 管理ツールは、構成ファイルを使用して特定のプロパティーの値を設定します。 いずれの場合も、ご使用のシステムに合わせて編集できるサンプル構成ファイルが提供されています。

このタスクについて

JM 3.0 V 9.3.0 V 9.3.0 IBM MQ 9.3.0 のサポートを導入した。 Jakarta Messaging 3.0. JMS 2.0 はまだ完全にサポートされている。

構成ファイルは、キーと値のペアのセットで構成されるプレーンテキストファイルで、等号(=)で区切られています。構成ファイルで定義されている 3 つのプロパティーの値を設定して、管理ツールを構成します。これらの 3 つのプロパティーの例は、以下のとおりです。

```
#Set the service provider
INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
#Set the initial context
PROVIDER_URL=ldap://polaris/o=ibm_us,c=us
#Set the authentication type
SECURITY_AUTHENTICATION=none
```

この例では、行の最初の列のハッシュ記号 (#) は、コメント、または使用されていない行を示します。

IBM MQ では、デフォルトの構成ファイルとして使用されるサンプル構成ファイルが提供されています。サンプル・ファイルの名前は、 `JMSAdmin.config` (JMS 2.0 の場合) または `JMS30Admin.config` (Jakarta Messaging 3.0 の場合) です。 このファイルは、*MQ_JAVA_INSTALL_PATH*`/bin` ディレクトリーにあります。 サンプル・ファイルを編集してシステムに必要な設定を定義するか、独自の構成ファイルを作成することができます。

管理ツールを開始するときに、 690 ページの『JMSAdmin ツールおよび JMS30Admin ツールの開始』で説明されているように、 `-cfg` コマンド行パラメーターを使用して、使用する構成ファイルを指定できます。 ツールの起動時に構成ファイル名を指定しないと、ツールはデフォルトの構成ファイル (`JMSAdmin.config` または `JMS30Admin.config`) をロードしようとします。 最初に現行ディレクトリーでこのファイルを検索し、次に *MQ_JAVA_INSTALL_PATH*`/bin` ディレクトリーでこのファイルを検索します。ここで、 *MQ_JAVA_INSTALL_PATH* は IBM MQ classes for JMS または IBM MQ classes for Jakarta Messaging インストール済み環境へのパスです。

LDAP 環境に保管される JMS オブジェクトまたは Jakarta Messaging オブジェクトの名前は、LDAP 命名規則に準拠している必要があります。 命名規則の 1 つとして、オブジェクトやコンテキストの名前には、cn= (共通名) や ou= (組織単位) といった接頭部を付けなければなりません。 この点、管理ツールでは、接頭部を付けなくてもオブジェクトやコンテキストの名前を参照できるようにすることで、LDAP サービス・プロバイダーの使用を簡単にしています。 ツールは、接頭部が入力されない場合に、自動的に、入力された名前にデフォルトの接頭部を付けます。 LDAP の場合、これは cn= です。 必要に応じてデフォルトの接頭部を変更するには、構成ファイル内の **NAME_PREFIX** プロパティーを設定します。

注 : 場合によっては、LDAP サーバーを Java オブジェクトの保管用に構成する必要があります。 詳しくは、ご使用の LDAP サーバーの資料を参照してください。

手順

1. ツールが使用するサービス・プロバイダーを定義するには、**INITIAL_CONTEXT_FACTORY** プロパティーを構成します。

   このプロパティーでサポートされる値は、以下のとおりです。

   - com.sun.jndi.ldap.LdapCtxFactory (LDAP 用)
   - com.sun.jndi.fscontext.RefFSContextFactory (ファイル・システム・コンテキスト用)
   - z/OS com.ibm.jndi.LDAPCtxFactory は、z/OS でのみサポートされており、LDAP サーバーへのアクセスが提供されます。 ただし、このクラスは com.sun.jndi.ldap.LdapCtxFactory との互換性はありません。 つまり、ある InitialContextFactory を使用して作成されたオブジェクトは、他の InitialContextFactory を使用して読み取ったり変更したりすることはできません。

   管理ツールを使用して、JMSAdmin 構成ファイルまたは JMS30Admin 構成ファイルに定義されている 3 つのパラメーターを使用することにより、他の JNDI コンテキストに接続することもできます。 別の InitialContextFactory を使用するには、次のようにします。

   a) **INITIAL_CONTEXT_FACTORY** プロパティーを、必要とするクラス名に設定します。

   b) **USE_INITIAL_DIR_CONTEXT**、**NAME_PREFIX**、および **NAME_READABILITY_MARKER** プロパティーを使用して、InitialContextFactory の振る舞いを定義します。

   これらのプロパティーの設定については、サンプル構成ファイルのコメントで説明されています。

   サポートされている **INITIAL_CONTEXT_FACTORY** 値のいずれかを使用する場合は、**USE_INITIAL_DIR_CONTEXT**、**NAME_PREFIX**、および **NAME_READABILITY_MARKER** プロパティーを定義する必要はありません。 ただし、それらのプロパティーに値を指定してシステム・デフォルトをオーバーライドすることは可能です。 例えば、オブジェクトが LDAP 環境に格納されている場合、**NAME_PREFIX** プロパティーを必要な接頭部に設定することで、ツールによってオブジェクト名とコンテキスト名に追加されるデフォルトの接頭部を変更することができます。

   3 つの InitialContextFactory プロパティーのうちの 1 つ以上を省略した場合、管理ツールはその他のプロパティーの値に基づいて適切なデフォルトを提供します。

2. **PROVIDER_URL** プロパティーを構成して、セッションの初期コンテキストの URL を定義します。

   この URL は、ツールが実行するすべての JNDI 操作のルートです。 このプロパティーは、次の 2 つの形式でサポートされています。

   - ldap://hostname/contextname
   - file:[drive:]/pathname

   LDAP URL の形式は、LDAP プロバイダーに応じて変わる場合があります。 詳細は、LDAP の資料を参照してください。

3. **SECURITY_AUTHENTICATION** プロパティーを構成して、JNDI がセキュリティー信任状をサービス・プロバイダーに渡すかどうかを定義します。

   このプロパティーは、LDAP サービス・プロバイダーが使用されている場合にのみ使用され、以下の 3 つの値のうちいずれかを指定できます。

   none (匿名認証)
   このパラメーターが none に設定されている場合、JNDI は、セキュリティー信任状を一切サービス・プロバイダーに渡さず、匿名認証 が実行されます。

   simple (単純認証)
   パラメーターが simple に設定された場合は、セキュリティー信任状が JNDI を介して基礎となるサービス・プロバイダーに渡されます。 これらのセキュリティー信任状は、ユーザー識別名 (User DN) とパスワードの形式で与えられます。

CRAM-MD5 (CRAM-MD5 認証メカニズム)
パラメーターが CRAM-MD5 に設定された場合は、セキュリティー信任状が JNDI を介して基礎となるサービス・プロバイダーに渡されます。 これらのセキュリティー信任状は、ユーザー識別名 (User DN) とパスワードの形式で与えられます。

SECURITY_AUTHENTICATION プロパティーに有効な値が指定されない場合、プロパティーのデフォルトは none になります。

セキュリティー資格認定が必要な場合は、ツールの初期設定時にこれらの入力を要求するプロンプトが出されます。 これを避けるには、JMSAdmin 構成ファイルで **PROVIDER_USERDN** プロパティーと **PROVIDER_PASSWORD** プロパティーを設定します。

注 : これらのプロパティーを使用しない場合、入力されたテキスト (パスワードを含む) が画面にもう一度表示されます。 これには、セキュリティー上の事柄が含まれている場合があります。

ツール自体は認証を行わず、認証タスクは LDAP サーバーが代行します。 ディレクトリーの様々な部分に対するアクセス権限のセットアップと保守は、LDAP サーバーの管理者が行う必要があります。 詳細は、LDAP の資料を参照してください。 認証が失敗した場合、ツールは、適切なエラー・メッセージを表示して終了します。

セキュリティーおよび JNDI について詳しくは、Oracle の Java Web サイト (Oracle Technology Network for Java Developers) にある文書を参照してください。

JMSAdmin ツールおよび JMS30Admin ツールの開始

IBM MQ JMS および Jakarta Messaging 管理ツールにはコマンド・ライン・インターフェースがあり、対話式に使用することも、バッチ処理を開始するために使用することもできます。

このタスクについて

対話モードで使用する場合は、管理コマンドを入力するためのコマンド・プロンプトが表示されます。 バッチ・モードを使用する場合は、ツールを開始するコマンドの中に、管理コマンド・スクリプトが入っているファイルの名前が組み込まれています。

手順

対話モード

- このツールを対話モードで開始する場合は、次のコマンドを入力します。

  JMS 2.0

  ```
  JMSAdmin [-t] [-v] [-cfg config_filename]
  ```

  JM 3.0

  ```
  JMS30Admin [-t] [-v] [-cfg config_filename]
  ```

  ここで、

  -t
  トレースを使用可能にします (デフォルトはトレース・オフです)。
  トレース・ファイルは"%MQ_JAVA_DATA_PATH%"\errors(Windows)または/var/mqm/trace(AIX and Linux)で生成されます。トレース・ファイルの名前は、次の形式になります:

  mqjms_*PID*.trc

  ここで、*PID* は JVM のプロセス ID です。

  -v
  詳細出力を生成します (デフォルトは簡潔な出力です)。

-cfg config_filename
代替の構成ファイルの名前を指定する。 このパラメーターを省略すると、デフォルトの構成ファイル JMSAdmin.config (JMS 2.0 の場合) または JMS30Admin.config (Jakarta Messaging 3.0 の場合) が使用されます。 構成ファイルについて詳しくは、688 ページの『JMSAdmin および JMS30Admin ツールの構成』を参照してください。

コマンド・プロンプトが表示されます。これは、管理ツールが管理コマンドを受け入れられる状態になったことを示します。 このプロンプトは、最初に、次のように表示されます。

```
InitCtx>
```

これは、現行コンテキスト (つまり、すべての名前指定およびディレクトリー操作が現在参照している JNDI コンテキスト) が、**PROVIDER_URL** 構成パラメーターに定義された初期コンテキストであることを示します。 このパラメーターの詳細については、688 ページの『JMSAdmin および JMS30Admin ツールの構成』を参照してください。

ディレクトリーのネーム・スペースをトラバースすると、プロンプトにもそれが反映され、プロンプトには常に現行コンテキストが表示されます。

バッチ・モード

- このツールをバッチ・モードで開始する場合は、次のコマンドを入力します。

  JMS 2.0

  ```
  JMSAdmin test.scp
  ```

  JMS 3.0

  ```
  JMS30Admin < test.scp
  ```

  ここで、*test.scp* は管理コマンドを含むスクリプト・ファイルです。 詳しくは、691 ページの『JMSAdmin および JMS30Admin での管理コマンドの使用』を参照してください。 ファイルの最後は、END コマンドでなければなりません。

JMSAdmin および JMS30Admin での管理コマンドの使用

IBM MQ JMS および Jakarta Messaging 管理ツールは、管理動詞とその適切なパラメーターで構成されるコマンドを受け入れます。

このタスクについて

以下の表に、管理ツールを使用してコマンドを入力するときに使用できる管理 verb をリストします。

表 37. 管理動詞

動詞	短縮形	説明
ALTER	ALT (T)	管理対象オブジェクトのプロパティーを少なくとも 1 つ変更する
DEFINE	DEF	管理対象オブジェクトの作成、保管、またはサブコンテキストの作成を行う
DISPLAY	DIS	保管されている 1 つ以上の管理対象オブジェクトのプロパティー、または現行コンテキストの内容を表示する
削除	DEL	1 つ以上の管理対象オブジェクトをネーム・スペースから除去する、あるいは空のサブコンテキストを除去する
CHANGE	CHG	初期コンテキストの下に属する任意のディレクトリー・ネーム・スペースをトラバースし、現行コンテキストを変える (セキュリティーの許可は保留)

表 37. 管理動詞 (続き)		
動詞	**短縮形**	**説明**
COPY	CP	保管されている管理対象オブジェクトのコピーを作成し、それを別名で保管する
MOVE	MV	管理対象オブジェクトの保管場所の名前を変える
END		管理ツールを閉じる

手順

- まだ管理ツールが開始していない場合は、690 ページの『JMSAdmin ツールおよび JMS30Admin ツールの開始』で説明されているようにして開始します。

  コマンド・プロンプトが表示されます。これは、管理ツールが管理コマンドを受け入れられる状態になったことを示します。 このプロンプトは、最初に、次のように表示されます。

  ```
  InitCtx>
  ```

  現在のコンテキストを変更するには、692 ページの『サブコンテキストの構成』で説明されているように、CHANGE 動詞を使用します。
- 以下の形式でコマンドを入力します。

  ```
  verb [param]*
  ```

  ここで、**verb** の部分には、691 ページの表 37 にリストされているいずれかの管理動詞が入ります。 すべての有効なコマンドには、コマンドの先頭に、標準の形式か短縮された形式の動詞が 1 つ含まれています。 動詞の名前は大/小文字を区別しません。
- コマンドを終了するには、Enter を押します。複数のコマンドを一緒に入力する場合は、Enter を押す直前に正符号 (+) を入力します。

  通常、コマンドの終わりには Enter を押します。 ただし、Enter を押す直前に正符号 (+) を入力すると、これをオーバーライドできます。 これによって、下の例に示されているように、複数行に渡ってコマンドを入力することが可能になります。

  ```
  DEFINE Q(BookingsInputQueue) +
  QMGR(QM.POLARIS.TEST) +
  QUEUE(BOOKINGS.INPUT.QUEUE) +
  CCSID(437)
  ```
- 管理ツールを閉じるには、**END** 動詞を使用します。

  この動詞にはパラメーターを指定できません。

サブコンテキストの構成

動詞 **CHANGE**、**DEFINE**、**DISPLAY**、および **DELETE** を使用して、ディレクトリー名前空間のサブコンテキストを構成できます。

このタスクについて

これらの動詞の使用法を、以下の表に示します。

表 38. サブコンテキストの操作に使用されるコマンドの構文と説明	
コマンド構文	**説明**
DEFINE CTX(ctxName)	現行コンテキストに ctxName という名前の子サブコンテキストを作成する処理を試行します。 セキュリティー違反があった場合、そのサブコンテキストがすでに存在している場合、または無効な名前が指定された場合は、コマンドが失敗します。

表 38. サブコンテキストの操作に使用されるコマンドの構文と説明 (続き)	
コマンド構文	**説明**
DISPLAY CTX	現在のコンテキストの内容を表示します。 管理対象オブジェクトには a、サブコンテキストには [D] の注釈が付けられます。 各オブジェクトの Java タイプも表示されます。
DELETE CTX(ctxName)	現行コンテキストから ctxName という名前の子コンテキストを削除する処理を試行します。 そのコンテキストが見つからない場合、コンテキストが空でない場合、またはセキュリティー違反があった場合は、コマンドが失敗します。
CHANGE CTX(ctxName)	現行コンテキストを変更して、ctxName という子のコンテキストを参照するようにします。 ctxName には次の 2 つの特殊値があり、このいずれかを指定できます。 **=UP** 現在のコンテキストの親に移動する **=INIT** 初期コンテキストに直接移動する 指定されたコンテキストが存在しない場合やセキュリティー違反があった場合は、コマンドが失敗します。

LDAP 環境に保管される JMS オブジェクトまたは Jakarta Messaging オブジェクトの名前は、LDAP 命名規則に準拠している必要があります。 命名規則の 1 つとして、オブジェクトやコンテキストの名前には、cn= (共通名) や ou= (組織単位) といった接頭部を付けなければなりません。 この点、管理ツールでは、接頭部を付けなくてもオブジェクトやコンテキストの名前を参照できるようにすることで、LDAP サービス・プロバイダーの使用を簡単にしています。 ツールは、接頭部が入力されない場合に、自動的に、入力された名前にデフォルトの接頭部を付けます。 LDAP の場合、これは cn= です。 必要に応じてデフォルトの接頭部を変更するには、構成ファイル内の **NAME_PREFIX** プロパティーを設定します。 詳しくは、688 ページの『JMSAdmin および JMS30Admin ツールの構成』を参照してください。

注 : 場合によっては、LDAP サーバーを Java オブジェクトの保管用に構成する必要があります。 詳しくは、ご使用の LDAP サーバーの資料を参照してください。

JMS オブジェクトの作成

JMS または Jakarta Messaging 接続ファクトリーおよび宛先オブジェクトを作成し、それらを JNDI 名前空間に保管するには、 DEFINE 動詞を使用します。 作成したオブジェクトを LDAP 環境に保管する場合、その名前は、一定の規則に準拠していなければなりません。 管理ツールを使用すると、デフォルトの接頭部をオブジェクト名に追加することによって、LDAP 命名規則に準拠できます。

このタスクについて

DEFINE 動詞は、指定したタイプ、名前、およびプロパティーを持つ管理対象オブジェクトを作成します。 新しいオブジェクトが現在のコンテキストに格納されます。

LDAP 環境に保管される JMS オブジェクトまたは Jakarta Messaging オブジェクトの名前は、LDAP 命名規則に準拠している必要があります。 命名規則の 1 つとして、オブジェクトやコンテキストの名前には、cn= (共通名) や ou= (組織単位) といった接頭部を付けなければなりません。 この点、管理ツールでは、接頭部を付けなくてもオブジェクトやコンテキストの名前を参照できるようにすることで、LDAP サービス・プロバイダーの使用を簡単にしています。 ツールは、接頭部が入力されない場合に、自動的に、入力された名前にデフォルトの接頭部を付けます。 LDAP の場合、これは cn= です。 必要に応じてデフォルトの接頭部を変更するには、構成ファイル内の **NAME_PREFIX** プロパティーを設定します。 詳しくは、688 ページの『JMSAdmin および JMS30Admin ツールの構成』を参照してください。

注 : 場合によっては、LDAP サーバーを Java オブジェクトの保管用に構成する必要があります。 詳しくは、ご使用の LDAP サーバーの資料を参照してください。

手順

1. まだ管理ツールが開始していない場合は、690 ページの『JMSAdmin ツールおよび JMS30Admin ツールの開始』で説明されているようにして開始します。

   コマンド・プロンプトが表示されます。これは、管理ツールが管理コマンドを受け入れられる状態になったことを示します。
2. コマンド・プロンプトに、新規オブジェクトの作成場所となるコンテキストが表示されていることを確認します。

   管理ツールを開始すると、最初はプロンプトが以下のように表示されます。

   ```
   InitCtx>
   ```

   現在のコンテキストを変更するには、692 ページの『サブコンテキストの構成』で説明されているように、CHANGE 動詞を使用します。
3. 接続ファクトリー、キュー宛先、またはトピック宛先を作成するには、以下のコマンド構文を使用します。

   ```
   DEFINE TYPE (name) [property]*
   ```

   つまり、DEFINE 動詞、管理対象オブジェクト参照 *TYPE* (name)、0 個以上のプロパティー (IBM MQ classes for JMS オブジェクトのプロパティーを参照) の順に入力します。
4. 接続ファクトリー、キュー宛先、またはトピック宛先を作成するには、以下のコマンド構文を使用します。

   ```
   DEFINE TYPE (name) [property]*
   ```
5. 新しく作成されたオブジェクトを表示するには、以下のコマンド構文で DISPLAY 動詞を使用します。

   ```
   DISPLAY TYPE (name)
   ```

例

以下の例では、DEFINE 動詞を使用して、testQueue というキューが初期コンテキストに作成されたことを示しています。 このオブジェクトは LDAP 環境に格納されるため、オブジェクト名 testQueue は接頭部がない状態で入力されますが、LDAP 命名規則に準拠するようにツールが自動的に接頭部を追加します。 コマンド DISPLAY Q(testQueue) をサブミットした場合も同様に、この接頭部が追加されます。

JM 3.0 V 9.3.0 V 9.3.0

```
InitCtx> DEFINE Q(testQueue)

InitCtx> DISPLAY CTX

Contents of InitCtx

a cn=testQueue       com.ibm.mq.jakarta.jms.MQQueue

1 Object(s)
0 Context(s)
1 Binding(s), 1 Administered
```

JMS 2.0

```
InitCtx> DEFINE Q(testQueue)

InitCtx> DISPLAY CTX

Contents of InitCtx

a cn=testQueue       com.ibm.mq.jms.MQQueue

1 Object(s)
0 Context(s)
1 Binding(s), 1 Administered
```

JMS オブジェクトを作成する時のエラー状態の例

オブジェクトの作成時に、いくつかの一般的なエラー状態が発生する場合があります。

このようなエラー状態の例を以下に示します。

CipherSuite にマップされる CipherSpec

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SSLCIPHERSUITE(RC4_MD5_US)
WARNING: Converting CipherSpec RC4_MD5_US to
CipherSuite SSL_RSA_WITH_RC4_128_MD5
```

オブジェクトのプロパティーが無効

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PRIORITY(4)
Unable to create a valid object, please check the parameters supplied
Invalid property for a QCF: PRI
```

プロパティー値のタイプが無効

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) CCSID(english)
Unable to create a valid object, please check the parameters supplied
Invalid value for CCS property: English
```

プロパティーの不調和 - クライアント/バインディング

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) HOSTNAME(polaris.hursley.ibm.com)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: Client-bindings attribute clash
```

プロパティーの不調和 - 出口の初期化

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SECEXITINIT(initStr)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: ExitInit string supplied
without Exit string
```

プロパティー値が有効範囲外

```
InitCtx/cn=Trash> DEFINE Q(testQ) PRIORITY(12)
Unable to create a valid object, please check the parameters supplied
Invalid value for PRI property: 12
```

プロパティーが不明

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PIZZA(ham and mushroom)
Unable to create a valid object, please check the parameters supplied
Unknown property: PIZZA
```

JMS アプリケーションから JNDI 管理対象オブジェクトを検索するときに Windows で発生する可能性があるエラー条件の例を以下に示します。

1. WebSphere JNDI プロバイダーの com.ibm.websphere.naming.WsnInitialContextFactory を使用する場合、サブコンテキストで定義されている管理オブジェクトにアクセスするには、スラッシュ (/) を使う必要があります。例えば、jms/MyQueueName です。 円記号 (¥) を使うと、InvalidNameException がスローされます。
2. Oracle JNDI プロバイダーの com.sun.jndi.fscontext.RefFSContextFactory を使用している場合、サブコンテキストで定義されている管理オブジェクトにアクセスするには、円記号 (¥) を使う必要があります。例えば、ctx1¥¥fred です。 スラッシュ (/) を使うと、NameNotFoundException がスローされます。

JMS オブジェクトの構成

ディレクトリー・ネーム・スペース内の管理対象オブジェクトの操作には、動詞 ALTER、DEFINE、DISPLAY、DELETE、COPY、および MOVE を使用できます。

このタスクについて

696 ページの表 39 に、これらの動詞の使い方を要約します。 *TYPE* の部分は、686 ページの『管理ツールを使用した JMS および Jakarta Messaging オブジェクトの構成』 で説明されている、必要な管理対象オブジェクトを表すキーワードに置き換えてください。

表 *39.* 管理対象オブジェクトの操作に使用されるコマンドの構文と説明

コマンド構文	説明
ALTER *TYPE*(name) [property]*	管理対象オブジェクトのプロパティーを、指定された値に更新する処理を試行します。 セキュリティー違反があった場合、指定されたオブジェクトが見つからない場合、または指定された新しいプロパティーが無効である場合は、コマンドが失敗します。
DEFINE *TYPE* (name) [property]*	指定されたプロパティーを持つタイプ *TYPE* のオブジェクトを作成し、それを name という名前で現行コンテキストに保管する処理を試行します。 セキュリティー違反があった場合、指定された名前が無効であるかその名前のオブジェクトが存在している場合、または指定されたプロパティーが無効である場合は、コマンドが失敗します。
DISPLAY *TYPE* (name)	name という名前で現行コンテキストにバインドされている、タイプ *TYPE* の管理対象オブジェクトのプロパティーを表示します。 オブジェクトが存在しない場合やセキュリティー違反があった場合は、コマンドが失敗します。
DELETE *TYPE*(name)	name という名前を持つタイプ *TYPE* の管理対象オブジェクトを現行コンテキストから除去する処理を試行します。 オブジェクトが存在しない場合やセキュリティー違反があった場合は、コマンドが失敗します。
COPY *TYPE* (nameA) *TYPE* (nameB)	nameA という名前を持つタイプ *TYPE* の管理対象オブジェクトから、nameB という名前のコピーを作成します。 これはすべて現行コンテキストの有効範囲内で行われます。 コピー元のオブジェクトが存在しない場合、nameB という名前のオブジェクトが存在する場合、またはセキュリティー違反があった場合は、コマンドが失敗します。
MOVE *TYPE*(nameA) *TYPE*(nameB)	nameA という名前を持つタイプ *TYPE* の管理対象オブジェクトを、nameB という名前の管理対象オブジェクトに移動 (名前変更) します。 これはすべて現行コンテキストの有効範囲内で行われます。 移動するオブジェクトが存在しない場合、nameB という名前のオブジェクトが存在する場合、またはセキュリティー違反があった場合は、コマンドが失敗します。

JMS 2.0 WebSphere Application Server での JMS 2.0 リソースの構成

WebSphere Application Server で JMS 2.0 リソースを構成するには、管理コンソールまたは wsadmin コマンドのいずれかを使用できます。

始める前に

JM 3.0 V 9.3.0 V 9.3.0 IBM MQ 9.3 は Jakarta Messaging 3.0 をサポートしていますが、WebSphere Application Server には現在同等のサポートはありません。 したがって、 WebSphere Application Server では、 Java Message Service 2.0 リソースを構成します。

このタスクについて

Java Message Service 2.0 アプリケーションは通常、外部で構成されたオブジェクトに依存します。このオブジェクトは、アプリケーションが JMS プロバイダーに接続する方法と、アプリケーションがアクセスする宛先を記述します。 JMS アプリケーションは Java Naming Directory Interface (JNDI) を使用して実行時に以下のタイプのオブジェクトにアクセスします。

- アクティベーション・スペック (Java EE アプリケーション・サーバーで使用)
- 統合接続ファクトリー (JMS 1.1 以降では、ドメイン固有のキュー接続ファクトリーおよびトピック接続ファクトリーよりも、ドメイン非依存 (統合) 接続ファクトリーが優先されます)
- トピック接続ファクトリー (JMS 1.0 アプリケーションで使用)
- キュー接続ファクトリー (JMS 1.0 アプリケーションで使用)
- キュー
- トピック

WebSphere Application Server の IBM MQ メッセージング・プロバイダーを介して、Java Message Service (JMS) メッセージング・アプリケーションは、 JMS メッセージング・リソースの外部プロバイダーとして IBM MQ システムを使用できます。 この方法を使用可能にするには、 WebSphere Application Server の IBM MQ メッセージング・プロバイダーを構成して、 IBM MQ ネットワーク上の任意のキュー・マネージャーに接続するための JMS リソースを定義します。

WebSphere Application Server を使用して、アプリケーション用の IBM MQ リソース(キュー接続ファクトリーなど)を構成したり、JMS 宛先に関連付けられたメッセージやサブスクリプションを管理したりすることができます。 セキュリティーは、IBM MQ を介して管理します。

関連タスク

IBM MQ と WebSphere Application Server の併用

WebSphere Application Server のトピック

IBM MQ メッセージング・プロバイダーを使用した相互運用

IBM MQ メッセージング・プロバイダーによるメッセージングの管理

管理コンソールのパネル名の、コマンド名および IBM MQ 名へのマッピング

JMS 2.0 管理コンソールを使用した JMS 2.0 リソースの構成

WebSphere Application Server 管理コンソールを使用すると、IBM MQ JMS プロバイダーのアクティベーション・スペック、接続ファクトリー、および宛先を構成することができます。

このタスクについて

WebSphere Application Server 管理コンソールを使用すると、以下のリソースを作成、表示、または変更することができます。

- アクティベーション・スペック
- ドメイン非依存型接続ファクトリー (JMS 1.1 以降)
- キュー接続ファクトリー
- トピック接続ファクトリー
- キュー
- トピック

以下のステップでは、管理コンソールを使用して、IBM MQ メッセージング・プロバイダーで使用する JMS リソースを構成する方法の概要を示します。 各手順には、詳細情報を参照できる WebSphere Application Server 製品資料のトピックの名前が含まれています。 IBM Documentation のこれらのトピックへのリンクについては、 関連リンク を参照してください。

混合バージョンの WebSphere Application Server セルでは、すべてのバージョンのノード上の IBM MQ リソースを管理できます。 ただし、一部のプロパティーはすべてのバージョンでは使用できません。 この場合、その特定のノードのプロパティーのみが管理コンソールに表示されます。

手順

IBM MQ メッセージング・プロバイダーで使用するアクティベーション・スペックを作成または構成するには、以下のようにします。

- アクティベーション・スペックを作成するには、IBM MQ JMS リソースの作成ウィザードを使用します。

  ウィザードを使用してアクティベーション・スペックのすべての詳細を指定することもできますし、クライアント・チャネル定義テーブル (CCDT) を使用することによって IBM MQ の接続詳細を指定することもできます。 ウィザードを使用して接続詳細を指定する場合、ホストとポートの情報を別々に入力することもできますし、複数インスタンス・キュー・マネージャーを使用している場合は、接続名のリストの形式でホストとポートの情報を入力することもできます。 詳しくは、「*IBM MQ* メッセージング・プロバイダーのアクティベーション・スペックの作成」を参照してください。
- アクティベーション・スペックの構成プロパティーを表示または変更するには、管理コンソールの IBM MQ メッセージング・プロバイダー接続ファクトリーの設定パネルを使用します。

  これらの構成プロパティーは、関連したキューおよびトピックに対して接続を作成する方法を制御します。 詳しくは、「*IBM MQ* メッセージング・プロバイダーのアクティベーション・スペックの構成」を参照してください。

IBM MQ メッセージング・プロバイダーで使用する統合接続ファクトリー、キュー接続ファクトリー、またはトピック接続ファクトリーを作成または構成するには、以下のようにします。

- 接続ファクトリーを作成するには、作成する接続ファクトリーのタイプを最初に選択し、「IBM MQ JMS リソースの作成」ウィザードを使用して詳細を指定します。
  - JMS アプリケーションが Point-to-Point メッセージングのみを使用する場合、Point-to-Point メッセージング専用の接続を作成する際に使用できる、Point-to-Point メッセージング・ドメイン用のドメイン固有の接続ファクトリーを作成します。
  - JMS アプリケーションがパブリッシュ/サブスクライブ・メッセージングのみを使用する場合、パブリッシュ/サブスクライブ・メッセージング専用の接続を作成する際に使用できる、パブリッシュ/サブスクライブ・メッセージング・ドメイン用のドメイン固有の接続ファクトリーを作成します。
  - JMS 1.1 以降の場合、Point-to-Point メッセージングとパブリッシュ/サブスクライブ・メッセージングの両方で使用できるドメイン非依存型接続ファクトリーを作成します。これにより、アプリケーションは同じトランザクションで Point-to-Point とパブリッシュ/サブスクライブの両方の処理を実行できます。

  ウィザードを使用して接続ファクトリーのすべての詳細を指定するか、クライアント・チャネル定義テーブル (CCDT) を使用することによって IBM MQ の接続詳細を指定するかを選択できます。 ウィザードを使用して接続詳細を指定する場合、ホストとポートの情報を別々に入力することもできますし、複数インスタンス・キュー・マネージャーを使用している場合は、接続名のリストの形式でホストとポートの情報を入力することもできます。 詳しくは、「*IBM MQ* メッセージング・プロバイダーの接続ファクトリーの作成」を参照してください。

接続ファクトリーの構成プロパティーを表示または変更するには、以下のようにします。

- 構成する接続ファクトリーのタイプについては、管理コンソールの接続ファクトリー設定パネルを使用します。

  構成プロパティーは、関連したキューおよびトピックに対して接続を作成する方法を制御します。 詳しくは、「*IBM MQ* メッセージング・プロバイダーのコレクション・ファクトリーの構成」、「*IBM MQ* メッセージング・プロバイダーのキュー・コレクション・ファクトリーの構成」、または「*IBM MQ* メッセージング・プロバイダーのトピック・コレクション・ファクトリーの構成」を参照してください。

IBM MQ メッセージング・プロバイダーとの point-to-point メッセージング用に JMS キュー宛先を構成するには、以下のようにします。

- 管理コンソールの IBM MQ メッセージング・プロバイダー・キュー設定パネルを使用して、以下のタイプのプロパティーを定義します。
  - 一般プロパティー (管理プロパティーおよび IBM MQ キュー・プロパティーを含む)。
  - キューをホストするキュー・マネージャーに接続する方法を指定する接続プロパティー。
  - IBM MQ メッセージング・プロバイダーの宛先に対する接続の動作を制御する拡張プロパティー。
  - キュー宛先用のすべてのカスタム・プロパティー。

詳しくは、「*IBM MQ* メッセージング・プロバイダーのキューの構成」を参照してください。

IBM MQ メッセージング・プロバイダーとのパブリッシュ/サブスクライブ・メッセージング用の JMS トピック宛先を作成または構成するには、以下のようにします。

- IBM MQ メッセージング・プロバイダー・トピック設定パネルを使用して、以下のタイプのプロパティーを定義します。
  - 一般プロパティー (管理プロパティーおよび IBM MQ トピック・プロパティーを含む)。
  - IBM MQ メッセージング・プロバイダーの宛先に対する接続の動作を制御する拡張プロパティー。
  - キュー宛先用のすべてのカスタム・プロパティー。

  詳しくは、「*IBM MQ* メッセージング・プロバイダーのトピックの構成」を参照してください。

関連概念

505 ページの『複数インスタンス・キュー・マネージャー』

複数インスタンス・キュー・マネージャーは、異なる複数のサーバー上で構成されている同じキュー・マネージャーのインスタンスです。 キュー・マネージャーの 1 つのインスタンスはアクティブ・インスタンスとして定義され、もう 1 つのインスタンスはスタンバイ・インスタンスとして定義されます。 アクティブ・インスタンスで障害が発生すると、複数インスタンス・キュー・マネージャーは、スタンバイ・サーバーで自動的に再始動します。

関連タスク

42 ページの『バイナリー形式の CCDT の構成』

クライアント・チャネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャネル定義と認証情報を判別します。 Multiplatforms では、キュー・マネージャーの作成時に、デフォルト設定が組み込まれたバイナリー CCDT が自動的に作成されます。 バイナリー CCDT を更新する場合は、**runmqsc** コマンドを使用します。

432 ページの『パブリッシュ/サブスクライブ・メッセージングの構成』

キュー型パブリッシュ/サブスクライブの開始、停止、および状況の表示を行えます。 ストリームの追加および除去、ブローカー階層に対するキュー・マネージャーの追加および削除も行えます。

WebSphere Application Server のトピック

IBM MQ メッセージング・プロバイダーのアクティベーション・スペック
IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの作成
IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの構成
IBM MQ メッセージング・プロバイダーの接続ファクトリーの作成
IBM MQ メッセージング・プロバイダーの統合接続ファクトリーの構成
IBM MQ メッセージング・プロバイダーのキュー接続ファクトリーの構成
IBM MQ メッセージング・プロバイダーのトピック接続ファクトリーの構成
IBM MQ メッセージング・プロバイダーのキューの構成
IBM MQ メッセージング・プロバイダーのトピックの構成

JMS 2.0 wsadmin スクリプト・コマンドを使用した JMS 2.0 リソースの構成

WebSphere Application Server wsadmin スクリプト・コマンドを使用して、JMS アクティベーション・スペック、接続ファクトリー、キュー、およびトピックに関する情報を作成、変更、削除、または表示できます。 IBM MQ リソース・アダプターの設定を表示および管理することもできます。

このタスクについて

以下のステップでは、 WebSphere Application Server wsadmin コマンドを使用して、 IBM MQ メッセージング・プロバイダーで使用する JMS リソースを構成する方法の概要を示します。 これらのコマンドの使用方法について詳しくは、 WebSphere Application Server 製品資料へのリンクについて「関連リンク」 を参照してください。

コマンドを実行するには、wsadmin スクリプト・クライアントの AdminTask オブジェクトを使用します。

コマンドを使用して新規オブジェクトを作成したり変更したりした後で、その変更をマスター構成に保存します。 例えば、次のコマンドを使用します。

```
AdminConfig.save()
```

使用可能な IBM MQ メッセージング・プロバイダー管理コマンドのリストと各コマンドの要旨を表示するには、wsadmin プロンプトで以下のコマンドを入力します。

```
print AdminTask.help('WMQAdminCommands')
```

指定されたコマンドの概要ヘルプを表示するには、wsadmin プロンプトで以下のコマンドを入力します。

```
print AdminTask.help('command_name')
```

手順

コマンドが発行されるスコープで定義された IBM MQ メッセージング・プロバイダー・リソースをすべてリストするには、以下のコマンドを使用します。

- アクティベーション・スペックをリストするには、**listWMQActivationSpecs** コマンドを使用します。
- 接続ファクトリーをリストするには、**listWMQConnectionFactories** コマンドを使用します。
- キュー・タイプ宛先をリストするには、**listWMQQueues** コマンドを使用します。
- トピック・タイプ宛先をリストするには、**listWMQTopics** コマンドを使用します。

特定の有効範囲で IBM MQ メッセージング・プロバイダーの JMS リソースを作成するには、以下のコマンドを使用します。

- アクティベーション・スペックを作成するには、**createWMQActivationSpec** コマンドを使用します。

  接続の確立に使用するすべてのパラメーターを指定してアクティベーション・スペックを作成することもできますし、クライアント・チャネル定義テーブル (CCDT) を使用して接続先のキュー・マネージャーを見つけるようにアクティベーション・スペックを作成することもできます。
- 接続ファクトリーを作成するには、**createWMQConnectionFactory** コマンドで **-type** パラメーターを使用して、作成する接続ファクトリーのタイプを指定します。
  - JMS アプリケーションが Point-to-Point メッセージングのみを使用する場合、Point-to-Point メッセージング専用の接続を作成する際に使用できる、Point-to-Point メッセージング・ドメイン用のドメイン固有の接続ファクトリーを作成します。
  - JMS アプリケーションがパブリッシュ/サブスクライブ・メッセージングのみを使用する場合、パブリッシュ/サブスクライブ・メッセージング専用の接続を作成する際に使用できる、パブリッシュ/サブスクライブ・メッセージング・ドメイン用のドメイン固有の接続ファクトリーを作成します。
  - JMS 1.1 以降の場合、Point-to-Point メッセージングとパブリッシュ/サブスクライブ・メッセージングの両方で使用できるドメイン非依存型接続ファクトリーを作成します。これにより、アプリケーションは同じトランザクションで Point-to-Point とパブリッシュ/サブスクライブの両方の処理を実行できます。

  デフォルトのタイプはドメイン非依存型接続ファクトリーです。
- キュー・タイプ宛先を作成するには、**createWMQQueue** コマンドを使用します。
- トピック・タイプ宛先を作成するには、**createWMQTopic** コマンドを使用します。

特定の有効範囲で IBM MQ メッセージング・プロバイダーの JMS リソースを変更するには、以下のコマンドを使用します。

- アクティベーション・スペックを変更するには、**modifyWMQActivationSpec** コマンドを使用します。

  アクティベーション・スペックのタイプは変更できません。 例えば、アクティベーション・スペックを作成してすべての構成情報を手動で入力した後で、CCDT を使用するためにそれを変更することはできません。
- 接続ファクトリーを変更するには、**modifyWMQConnectionFactory** コマンドを使用します。

- キュー・タイプ宛先を変更するには、**modifyWMQQueue** コマンドを使用します。
- トピック・タイプ宛先を変更するには、**modifyWMQTopic** コマンドを使用します。

特定の有効範囲で IBM MQ メッセージング・プロバイダーの JMS リソースを削除するには、以下のコマンドを使用します。

- アクティベーション・スペックを削除するには、**deleteWMQActivationSpec** コマンドを使用します。
- 接続ファクトリーを削除するには、**deleteWMQConnectionFactory** コマンドを使用します。
- キュー・タイプ宛先を削除するには、**deleteWMQQueue** コマンドを使用します。
- トピック・タイプ宛先を削除するには、**deleteWMQTopic** コマンドを使用します。

特定の IBM MQ メッセージング・プロバイダー・リソースに関する情報を表示するには、以下のコマンドを使用します。

- 特定のアクティベーション・スペックに関連付けられたすべてのパラメーターとその値を表示するには、**showWMQActivationSpec** コマンドを使用します。
- 特定の接続ファクトリーに関連付けられたすべてのパラメーターとその値を表示するには、**showWMQConnectionFactory** コマンドを使用します。
- 特定のキュー・タイプ宛先に関連付けられたすべてのパラメーターとその値を表示するには、**showWMQQueue** コマンドを使用します。
- 特定のトピック・タイプ宛先に関連付けられたすべてのパラメーターとその値を表示するには、**deleteWMQTopic** コマンドを使用します。

IBM MQ リソース・アダプターまたは IBM MQ メッセージング・プロバイダーの設定を管理するには、以下のコマンドを使用します。

- 特定のスコープでインストールされた IBM MQ リソース・アダプターの設定を管理するには、**manageWMQ** コマンドを使用します。
- **manageWMQ** コマンドで設定可能なすべてのパラメーターとその値を表示するには、**showWMQ** コマンドを使用します。 これらの設定は、IBM MQ リソース・アダプターまたは IBM MQ メッセージング・プロバイダーと関連しています。 **showWMQ** コマンドは、IBM MQ リソース・アダプターで設定されたすべてのカスタム・プロパティーも表示します。

関連概念

505 ページの『複数インスタンス・キュー・マネージャー』
複数インスタンス・キュー・マネージャーは、異なる複数のサーバー上で構成されている同じキュー・マネージャーのインスタンスです。 キュー・マネージャーの 1 つのインスタンスはアクティブ・インスタンスとして定義され、もう 1 つのインスタンスはスタンバイ・インスタンスとして定義されます。 アクティブ・インスタンスで障害が発生すると、複数インスタンス・キュー・マネージャーは、スタンバイ・サーバーで自動的に再始動します。

関連タスク

42 ページの『バイナリー形式の CCDT の構成』
クライアント・チャネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャネル定義と認証情報を判別します。 Multiplatforms では、キュー・マネージャーの作成時に、デフォルト設定が組み込まれたバイナリー CCDT が自動的に作成されます。 バイナリー CCDT を更新する場合は、**runmqsc** コマンドを使用します。

432 ページの『パブリッシュ/サブスクライブ・メッセージングの構成』
キュー型パブリッシュ/サブスクライブの開始、停止、および状況の表示を行えます。 ストリームの追加および除去、ブローカー階層に対するキュー・マネージャーの追加および削除も行えます。

WebSphere Application Server のトピック

createWMQActivationSpec コマンド

createWMQConnectionFactory コマンド

createWMQQueue コマンド

createWMQTopic コマンド

deleteWMQActivationSpec コマンド
deleteWMQConnectionFactory コマンド
deleteWMQQueue コマンド
deleteWMQTopic コマンド
listWMQActivationSpecs コマンド
listWMQConnectionFactories コマンド
listWMQQueues コマンド
listWMQTopics コマンド
modifyWMQActivationSpec コマンド
modifyWMQConnectionFactory コマンド
modifyWMQQueue コマンド
modifyWMQTopic コマンド
showWMQActivationSpec コマンド
showWMQConnectionFactory コマンド
showWMQQueue コマンド
showWMQTopic コマンド
showWMQ コマンド
manageWMQ コマンド

JMS 2.0 JMS 2.0 共用サブスクリプションの使用

WebSphere Application Server traditional 9.0 では、 IBM MQ 9.0 で JMS 2.0 共有サブスクリプションを構成して使用することができます。

このタスクについて

JMS 2.0 の仕様で、1 つのサブスクリプションを 1 つ以上のコンシューマーがオープンできるという共用サブスクリプションの概念が導入されました。 メッセージはそれらのすべてのコンシューマー間で共用されます。 同じキュー・マネージャーに接続する限り、コンシューマーが存在する場所に制限はありません。

共用サブスクリプションは、非共用サブスクリプションと呼ばれるようになったサブスクリプションと同じ意味で、永続サブスクリプションにすることも非永続サブスクリプションにすることもできます。

使用するサブスクリプションをコンシューマーが識別できるように、サブスクリプション名を指定する必要があります。 これは非共用の永続サブスクリプションと似ていますが、共用サブスクリプションが必要な場合は、あらゆるケースでサブスクリプション名が必要になります。 しかし、クライアント ID は、共用の永続サブスクリプションには必要ありません (指定することもできますが必須ではありません)。

共用サブスクリプションはロード・バランシング・メカニズムと見なすことができますが、IBM MQ でも JMS 2.0 仕様でも、メッセージをコンシューマー間に分散させる方法についての規定はありません。

WebSphere Application Server traditional 9.0 では、IBM MQ 9.0 リソース・アダプターが事前にインストールされます。

以下の手順は、WebSphere Application Server traditional 管理コンソールで共用永続サブスクリプションまたは共用非永続サブスクリプションを使用するためのアクティベーション・スペックを構成する方法を示しています。

手順

最初に JNDI でオブジェクトを作成します。

1. トピック宛先を JNDI で通常どおり作成します (697 ページの『管理コンソールを使用した JMS 2.0 リソースの構成』を参照)。
2. アクティベーション・スペックを作成します (697 ページの『管理コンソールを使用した JMS 2.0 リソースの構成』を参照)。

必要なプロパティーを正確に指定してアクティベーション・スペックを作成できます。 永続サブスクリプションを使用する場合は、作成時にそのことを選択して名前を指定できます。 非永続サブスクリプションを使用する場合は、現時点では名前を指定できません。 その代わりに、サブスクリプション名としてカスタム・プロパティーを作成する必要があります。

必要なカスタム・プロパティーを指定して、作成したアクティベーション・スペックを更新します。 次の 2 つのカスタム・プロパティーを指定する必要があります。

- いずれの場合でも、このアクティベーション・スペックが共用サブスクリプションを使用することを示すカスタム・プロパティーを作成する必要があります。
- サブスクリプションを非永続として作成した場合は、サブスクリプション名プロパティーをカスタム・プロパティーとして設定する必要があります。

以下の表に、各カスタム・プロパティーに指定できる有効な値を示します。

プロパティー名	タイプ	有効値
sharedSubscription	ストリング	true、false
subscriptionName	ストリング	ゼロでない長さの Java ストリング

3. 「**アクティベーション・スペック・コレクション (Activation specification collection)**」フォームに表示されたリストから、該当するアクティベーション・スペックを選択します。

   アクティベーション・スペックの詳細が「**IBM MQ メッセージング・プロバイダー・アクティベーション・スペック設定**」フォームに表示されます。
4. 「**IBM MQ メッセージング・プロバイダー・アクティベーション・スペック設定**」フォームで、「**カスタム・プロパティー**」をクリックします。

   「**カスタム・プロパティー**」フォームが表示されます。
5. 非永続サブスクリプションを使用している場合は、subscriptionName カスタム・プロパティーを作成します。

   アクティベーション・スペックの「**カスタム・プロパティー**」パネルで、「**新規**」をクリックし、以下の詳細情報を入力します。

   名前

   カスタム・プロパティーの名前。この場合は subscriptionName です。

   値

   カスタム・プロパティーの値。 「**値**」 フィールドで JNDI 名を使用できます (例: WASSharedSub0ne)。

   タイプ

   カスタム・プロパティーのタイプ。 リストからカスタム・プロパティー・タイプを選択します。この場合 java.lang.String です。
6. 共用永続サブスクリプションと共用非永続サブスクリプションのどちらの場合も、sharedSubscription カスタム・プロパティーを作成します。

   アクティベーション・スペックの「**カスタム・プロパティー**」パネルで、「**新規**」をクリックし、以下の詳細情報を入力します。

   名前

   カスタム・プロパティーの名前。この場合は sharedSubscription です。

   値

   カスタム・プロパティーの値。 アクティベーション・スペックが共用サブスクリプションを使用することを示すために、この値を true に設定します。 後でこのアクティベーション・スペックで共用サブスクリプションの使用を止める場合は、このカスタム・プロパティーの値を false に設定します。

タイプ

カスタム・プロパティーのタイプ。 リストからカスタム・プロパティー・タイプを選択します。この場合 `java.lang.String` です。

7. プロパティーを設定したら、アプリケーション・サーバーを再始動します。

   メッセージを受信するとアクティベーション・スペックのメッセージ駆動型 Bean (MDB) が起動され、それらの MDB の間でのみ、送信されたメッセージが共用されます。

関連概念

複製サブスクリプションおよび共用サブスクリプション

サブスクリプション永続性

関連タスク

インバウンド通信のリソース・アダプターの構成

WebSphere Application Server traditional 9.0 の関連情報

IBM MQ メッセージング・プロバイダーのトピックの構成

IBM MQ メッセージング・プロバイダーのアクティベーション・スペック

IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの作成

IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの構成

IBM MQ メッセージング・プロバイダー JMS リソースのカスタム・プロパティーの構成

JMS 2.0 JMS 2.0 の ConnectionFactory および Destination の Lookup プロパティーの使用

WebSphere Application Server traditional 9.0 では、アクティベーション・スペックで ConnectionFactoryLookup プロパティーと DestinationLookup プロパティーに管理対象オブジェクトの JNDI 名を指定して、他のアクティベーション・スペック・プロパティーより優先的に使用することができます。

このタスクについて

JMS 2.0 仕様では、メッセージ駆動型 Bean (MDB) の起動に使用するアクティベーション・スペックに、2 つの追加のプロパティーが規定されました。 以前は、メッセージング・システムへの接続に必要な詳細情報を入力し、どの宛先からメッセージを取得するのかを定義するためのカスタム・プロパティーを、各ベンダーがアクティベーション・スペックに指定する必要がありました。

標準になった connectionFactoryLookup および destinationLookup プロパティーを使用して、検索して使用する該当オブジェクトの JNDI 名を指定できます。 WebSphere Application Server traditional 9.0 では、IBM MQ 9.0 リソース・アダプターが事前にインストールされます。

以下の手順は、WebSphere Application Server traditional 管理コンソールを使用してこれらの 2 つのプロパティーをカスタマイズして使用する方法を示しています。

手順

最初に JNDI でオブジェクトを作成します。

1. ConnectionFactory を JNDI で通常どおり作成します (697 ページの『管理コンソールを使用した JMS 2.0 リソースの構成』を参照)。
2. 宛先を JNDI で通常どおり作成します (697 ページの『管理コンソールを使用した JMS 2.0 リソースの構成』を参照)。

   宛先オブジェクトには正確な値を指定してください。
3. 必要な値を使用してアクティベーション・スペックを作成します (697 ページの『管理コンソールを使用した JMS 2.0 リソースの構成』を参照)。

   必要なプロパティーを正確に指定してアクティベーション・スペックを作成できます。 ただし、以下の考慮事項に留意してください。

- IBM MQ リソース・アダプターで Java EE の接続ファクトリーと宛先のルックアップ・プロパティーを使用する場合、アクティベーション・スペックの作成時にどのプロパティーが使用されているかはあまり重要ではありません (ActivationSpec の ConnectionFactoryLookup および DestinationLookup プロパティーを参照)。
- 接続ファクトリーまたは宛先でまだ定義されていないプロパティーであっても、アクティベーション・スペックには指定する必要があります。 したがって、接続コンシューマー・プロパティーと追加のプロパティー、および実際に接続が作成されたときに使用する認証情報を定義する必要があります。
- 接続ファクトリーで定義されるプロパティーのうち、ClientID プロパティーには特別な処理があります。 その理由は、一般的なシナリオでは、単一の接続ファクトリーを複数のアクティベーション・スペックで使用するからです。 これによって管理が簡略化されますが、JMS 仕様では固有のクライアント ID が必要なため、ConnectionFactory に設定された値をアクティベーション・スペックでオーバーライドできなければなりません。 ClientID をアクティベーション・スペックで設定しないと、接続ファクトリーの値が使用されます。

WebSphere Application Server 管理コンソールで 2 つの新しいカスタム・プロパティーを指定して、作成したアクティベーション・スペックを更新する (手順 705 ページの『4』 を参照) か、代わりにアノテーションを使用します (手順 705 ページの『5』 を参照)。

4. WebSphere Application Server 管理コンソールでアクティベーション・スペックを更新します。

   これらの 2 つのプロパティーは、アクティベーション・スペックのカスタム・プロパティー・パネルで設定する必要があります。 これらのプロパティーは、アクティベーション・スペックのメイン・パネルにもアクティベーション・スペックの作成ウィザードにも表示されません。

   a)「**アクティベーション・スペック・コレクション (Activation specification collection)**」フォームに表示されたリストから、該当するアクティベーション・スペックを選択します。

   アクティベーション・スペックの詳細が「**IBM MQ メッセージング・プロバイダー・アクティベーション・スペック設定**」フォームに表示されます。

   b)「**IBM MQ メッセージング・プロバイダー・アクティベーション・スペック設定**」フォームで、「**カスタム・プロパティー**」をクリックします。

   「**カスタム・プロパティー**」フォームが表示されます。

   c)「**カスタム・プロパティー**」フォームで、2 つの新しいカスタム・プロパティー (いずれも java.lang.String 型) を作成します。

   いずれの場合も、「**新規**」をクリックしてから、以下のカスタム・プロパティーの詳細情報を入力します。

   名前

   カスタム・プロパティーの名前。connectionFactoryLookup または destinationLookup のいずれかです。

   値

   カスタム・プロパティーの値。 「**値**」 フィールドには、 QuoteCF や QuoteQ などの JNDI 名を使用できます。

   タイプ

   カスタム・プロパティーのタイプ。 リストからカスタム・プロパティー・タイプを選択します。 この場合 java.lang.String です。

   これで、デプロイされた MDB がこれらの値を使用して接続ファクトリーおよび宛先を作成するようになります。 MDB をデプロイするときに、JNDI 値の構成を設定する必要はありません。

5. アクティベーション・スペックの代わりにアノテーションを使用します。

   MDB コード内でアノテーションを使用して値を指定することもできます。 例えば、 JNDI names QuoteCF と QuoteQ を使用すると、コードは次のようになります。

```
@MessageDriven(activationConfig = {
        @ActivationConfigProperty(propertyName =  "destinationType" , propertyValue =
"javax.jms.Topic" ),
        @ActivationConfigProperty(propertyName =  "destinationLookup" , propertyValue =
"QuoteQ" ),
        @ActivationConfigProperty(propertyName =  "connectionFactoryLookup" , propertyValue
```

```
=  "QuoteCF" )}, mappedName =  "LookupMDB" )
        @TransactionAttribute(TransactionAttributeType.REQUIRED)
        @TransactionManagement(TransactionManagementType.CONTAINER)
        publicclass LookupMDB implements MessageListener {
```

関連タスク

インバウンド通信のリソース・アダプターの構成

WebSphere Application Server traditional 9.0 の関連情報

IBM MQ メッセージング・プロバイダーの統合接続ファクトリーの構成

IBM MQ メッセージング・プロバイダーのトピックの構成

IBM MQ メッセージング・プロバイダーのアクティベーション・スペック

IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの作成

IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの構成

IBM MQ メッセージング・プロバイダー JMS リソースのカスタム・プロパティーの構成

最新のリソース・アダプター保守レベルを使用するための WebSphere Application Server の構成

WebSphere Application Server フィックスパックの適用時に IBM MQ リソース・アダプターが使用可能な最新の保守レベルに自動的に更新されるようにするために、各ノードのインストールに適用した WebSphere Application Server フィックスパックに含まれている最新バージョンのリソース・アダプターを使用するように、環境内のすべてのサーバーを構成することができます。

始める前に

重要 :

- JM 3.0 WebSphere Application Server traditional は現在 'Jakarta EE をサポートしていない。IBM MQ リソース・アダプターのサポート・ステートメントを参照してください。
- いずれかのプラットフォームで WebSphere Application Server 8.5 以前を使用している場合、IBM MQ 8.0 以降のリソース・アダプターをアプリケーション・サーバーにインストールしないでください。 IBM MQ 8.0 以降のリソース・アダプターは、JMS 2.0 をサポートするアプリケーション・サーバーにのみデプロイできます。 ただし、WebSphere Application Server 8.5 以前は、JMS 1.1 のみをサポートします。

このタスクについて

現在の構成に以下のいずれかの状況が当てはまり、最新のバージョンの IBM MQ リソース・アダプターを使用するよう環境内のすべてのサーバーを構成する場合は、このタスクを使用します。

- 環境内のアプリケーション・サーバーの JVM ログには、 WebSphere Application Server 7.0.0 Fix Pack 1 以降が適用された後の以下の IBM MQ リソース・アダプターのバージョン情報が示されます。

  ```
  WMSG1703I:RAR 実装バージョン 7.0.0.0-k700-L080820
  ```

- 環境内のどのアプリケーション・サーバーの JVM ログにも次の項目が含まれます。

  ```
  WMSG1625E: 検出できませんでした
  指定されたパス < null> にある IBM MQ メッセージング・プロバイダー・コード
  ```

- これまでは、特定の保守レベルの IBM MQ リソース・アダプターを使用するように 1 つ以上のノードが手動で更新されていましたが、現在は現行の WebSphere Application Server 保守レベルに含まれる最新バージョンのリソース・アダプターを使用するように変更されました。

例で参照されている *profile_root* ディレクトリーは、WebSphere Application Server プロファイルのホーム・ディレクトリーです (例: `C:\Program Files\IBM\WebSphere\AppServer1`)。

ご使用の環境内のすべてのセルおよび単一サーバー・インストール済み環境に対して以下のステップを実行すると、新しい WebSphere Application Server フィックスパックが適用されたときに、サーバーは IBM MQ リソース・アダプターの保守を自動的に受け取ります。

手順

1. アプリケーション・サーバーを始動します。 プロファイルがネットワーク・デプロイメント構成の一部である場合、デプロイメント・マネージャーとすべてのノード・エージェントを始動します。 プロファイルに管理エージェントが含まれる場合は、管理エージェントを始動します。
2. IBM MQ リソース・アダプターの保守レベルを確認します。
   a) コマンド・プロンプト・ウィンドウを開き、 *profile_root*\bin ディレクトリーに移動します。

      例えば、cd C:\Program Files\IBM\WebSphere\AppServer1\bin と入力します。

   b) wsadmin.bat -lang jython と入力して wsadmin ツールを開始し、指示された場合にはユーザー名とパスワードを入力します。

   c) 次のコマンドを入力してから、Return キーを 2 回押します。

```
wmqInfoMBeansUnsplit = AdminControl.queryNames("WebSphere:type=WMQInfo,*")
wmqInfoMBeansSplit = AdminUtilities.convertToList(wmqInfoMBeansUnsplit)
for wmqInfoMBean in wmqInfoMBeansSplit: print wmqInfoMBean; print AdminControl.invoke(wmqInfoMBean,
'getInfo', '')
```

      このコマンドは Jacl でも実行できます。 これを行う方法について詳しくは、 WebSphere Application Server 製品資料の「 使用可能な最新の *IBM MQ* リソース・アダプター保守レベルをサーバーが使用するようにする 」を参照してください。

   d) コマンドの出力表示から WMSG1703I メッセージを探し、リソース・アダプター・レベルを確認します。

      例えば、WebSphere Application Server 7.0.1 Fix Pack 5 の場合、メッセージは次のようになるはずです。

      ```
      WMSG1703I: RAR 実装バージョン 7.0.1.3-k701-103-100812
      ```

      このメッセージは、バージョンがこのフィックスパックにおける正しいリソース・アダプター・レベルである 7.0.1.3-k701-103-100812 であることを示しています。 ただし、代わりに以下のメッセージが表示された場合は、 リソース・アダプターを WebSphere Application Server 7.0.1 Fix Pack 5 用の正しい保守レベルに調整する必要があることを意味します。

      ```
      WMSG1703I: RAR 実装バージョン 7.0.0.0-k700-L080820
      ```

3. 以下の Jython スクリプトを convertWMQRA.py というファイルにコピーし、それをプロファイル・ルート・ディレクトリー (例えば、C:\Program Files\IBM\WebSphere\AppServer1\bin) に保存します。

```
ras = AdminUtilities.convertToList(AdminConfig.list('J2CResourceAdapter'))

for ra in ras :
  desc = AdminConfig.showAttribute(ra, "description")
  if (desc == "WAS 7.0 Built In MQ Resource Adapter") or (desc == "WAS 7.0.0.1 Built In MQ Resource
Adapter"):
    print "Updating archivePath and classpath of " + ra
    AdminConfig.modify(ra, [['archivePath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar"]])
    AdminConfig.unsetAttributes(ra, ['classpath'])
    AdminConfig.modify(ra, [['classpath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar"]])
    AdminConfig.save()
  #end if
#end for
```

   ヒント: ファイルを保存するとき、テキスト・ファイルではなく python ファイルとして保存されていることを確認します。

4. ここで作成した Jython スクリプトを WebSphere Application Server wsadmin ツールを使用して実行します。

   コマンド・プロンプトを開き、WebSphere Application Server ディレクトリーのホーム・ディレクトリーにある\bin ディレクトリーに移動します。例えば、C:\Program Files\IBM\WebSphere\AppServer1\bin ディレクトリーの場合は、以下のコマンドを入力して、Return キーを押します:

   ```
   wsadmin -lang jython -f convertWMQRA.py
   ```

指示された場合にはユーザー名とパスワードを入力します。

注: ネットワーク・デプロイメント構成の一部であるプロファイルに対してスクリプトを実行すると、スクリプトはその構成で更新が必要なプロファイルをすべて更新します。 既存の構成ファイルに不整合がある場合、完全な再同期が必要になる場合があります。

5. ネットワーク・デプロイメント構成で実行する場合、ノード・エージェントが完全に再同期されていることを確認します。 詳しくは、「wsadmin スクリプト・ツールによるノードの同期化」または「ノードの追加、管理、および除去」を参照してください。
6. プロファイル内のすべてのサーバーを停止します。 プロファイルがネットワーク・デプロイメント構成の一部である場合、構成内のすべてのクラスター・メンバーを停止し、構成内のすべてのノード・エージェントを停止し、さらにデプロイメント・マネージャーも停止します。 プロファイルに管理エージェントが含まれる場合は、管理エージェントを停止します。
7. **osgiCfgInit** コマンドを *profile_root*/bin ディレクトリーから実行します。

   osgiCfgInit コマンドは、OSGi ランタイム環境で使用されるクラス・キャッシュをリセットします。プロファイルがネットワーク・デプロイメント構成の一部である場合、構成の一部であるすべてのプロファイルの profile_root/bin ディレクトリーから **osgiCfgInit** コマンドを実行します。
8. プロファイル内のすべてのサーバーを再始動します。 プロファイルがネットワーク・デプロイメント構成の一部である場合、構成内のすべてのクラスター・メンバーを再始動し、構成内のすべてのノード・エージェントを再始動し、さらにデプロイメント・マネージャーも再始動します。 プロファイルに管理エージェントが含まれる場合は、管理エージェントを再始動します。
9. 手順 2 を繰り返して、リソース・アダプターが正しいレベルになったことを確認します。

次のタスク

このトピックで説明されているステップを実行した後も引き続き問題が発生し、 WebSphere Application Server 管理コンソールの「 JMS プロバイダー設定」パネルの「 **リソース・アダプターの更新** 」ボタンを使用して、ご使用の環境内のすべてのノードで IBM MQ リソース・アダプターを更新した場合は、 APAR PM10308 で説明されている問題が発生している可能性があります。

関連概念

IBM MQ リソース・アダプターの使用

WebSphere Application Server 8.5.5 の関連情報

使用可能な最新の IBM MQ リソース・アダプター保守レベルを使用するためのサーバーの構成

wsadmin スクリプト・ツールによるノードの同期化

ノードの追加、管理、および除去

JMS プロバイダー設定

JMS PROVIDERVERSION プロパティーの構成

IBM MQ メッセージング・プロバイダーには、通常モード、制限付き通常モード、マイグレーション・モードという 3 つの操作モードがあります。 JMS **PROVIDERVERSION** プロパティーを設定して、これらのモードのうち JMS アプリケーションがパブリッシュおよびサブスクライブに使用するモードを選択できます。

このタスクについて

IBM MQ メッセージング・プロバイダーの操作モードを選択するときには、主に PROVIDERVERSION 接続ファクトリー・プロパティーを設定して制御できます。 操作モードが指定されていない場合は、操作モードを自動的に選択することもできます。

PROVIDERVERSION プロパティーによって、以下に示す 3 つの IBM MQ メッセージング・プロバイダー操作モードを区別します。

IBM MQ メッセージング・プロバイダーの通常モード

通常モードは、IBM MQ キュー・マネージャーのすべての機能を使用して JMS を実装します。 このモードは、JMS 2.0 の API と機能を使用するように最適化されています。

IBM MQ メッセージング・プロバイダーの制限付き通常モード
制限付き通常モードは、JMS 2.0 の API を使用しますが、新機能 (共有サブスクリプション、遅延送達、および非同期送信) は使用しません。

IBM MQ メッセージング・プロバイダーのマイグレーション・モード
移行モードを使用すると、IBM MQ 8.0 以降のキュー・マネージャーに接続できますが、IBM WebSphere MQ 7.0 以降のキュー・マネージャーの機能 (先読みやストリーミングなど) はどれも使用されません。

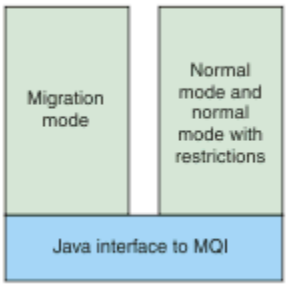


図 *89.* メッセージング・プロバイダー・モード

手順

特定の接続ファクトリーの **PROVIDERVERSION** プロパティーを構成するには、次のようにします。

- IBM MQ Explorer を使用して **PROVIDERVERSION** プロパティーを構成するには、 キュー・マネージャーおよびオブジェクトの構成を参照してください。
- JMS 管理ツールを使用して **PROVIDERVERSION** プロパティーを構成するには、 キュー・マネージャーおよびオブジェクトの構成を参照してください。
- IBM JMS 拡張または IBM MQ JMS 拡張を使用して JMS アプリケーションで **PROVIDERVERSION** プロパティーを構成するには、IBM MQ classes for JMS アプリケーションでの接続ファクトリーおよび宛先の作成と構成を参照してください。

JVM 内のすべての接続ファクトリーの接続ファクトリー・プロバイダー・モード設定を指定変更するには、次のようにします。

- 接続ファクトリー・プロバイダー・モード設定を指定変更するには、`com.ibm.msg.client.wmq.overrideProviderVersion` プロパティーを使用します。

  使用している接続ファクトリーを変更できない場合は、`com.ibm.msg.client.wmq.overrideProviderVersion` プロパティーを使用して、その接続ファクトリーの設定をオーバーライドできます。 この指定変更は JVM 中のすべての接続ファクトリーに適用されますが、実際の接続ファクトリー・オブジェクトは変更されません。

関連概念
JMS プロバイダー・バージョンのトラブルシューティング
関連資料
PROVIDERVERSION
接続ファクトリーのプロパティー
IBM MQ classes for JMS オブジェクトのプロパティー間の依存関係

IBM MQ メッセージング・プロバイダーの操作のモード

JMS アプリケーションがパブリッシュおよびサブスクライブに使用する IBM MQ メッセージング・プロバイダーの操作モードを選択するには、接続ファクトリーの PROVIDERVERSION プロパティーを適切な値に設定します。 場合によっては、PROVIDERVERSION プロパティーを指定なしに設定します。 このようにすると、JMS クライアントではアルゴリズムを使って、使用する操作モードを判別します。

PROVIDERVERSION プロパティー値

接続ファクトリーの **PROVIDERVERSION** プロパティーは、以下のいずれかの値に設定できます。

8 - 通常モード
JMS アプリケーションは、通常モードを使用します。 このモードは、IBM MQ キュー・マネージャーのすべての機能を使用して JMS を実装します。

7 - 制限付き通常モード
JMS アプリケーションは、制限付き通常モードを使用します。 このモードは JMS 2.0 API を使用しますが、共用サブスクリプション、遅延送達、非同期送信などの新機能は使用しません。

6 - 移行モード
JMS アプリケーションは、移行モードを使用します。 移行モードでは、IBM MQ classes for JMS は、IBM WebSphere MQ 6.0 に付属の機能やアルゴリズムに似た機能やアルゴリズムを使用します。

unspecified (デフォルト値)
JMS クライアントでは、アルゴリズムを使って、使用する操作モードを判別します。

PROVIDERVERSION プロパティーに指定する値は、ストリングでなければなりません。 オプション 8、7、または 6 を指定する場合、次のいずれかのフォーマットでこれを行えます。

- V.R.M.F
- V.R.M
- V.R
- V

ここで、V、R、M、および F は、ゼロ以上の整数値です。 追加の R、M、および F の値は任意指定で、細かい制御が必要な場合に使用できます。 例えば、**PROVIDERVERSION** レベル 7 を使用する場合は、**PROVIDERVERSION** = 7、7.0、7.0.0 または 7.0.0.0 を設定できます。

接続ファクトリー・オブジェクトのタイプ

PROVIDERVERSION プロパティーは、以下の接続ファクトリー・オブジェクト・タイプに設定できます。

- MQConnectionFactory
- MQQueueConnectionFactory
- MQTopicConnectionFactory
- MQXAConnectionFactory
- MQXAQueueConnectionFactory
- MQXAQueueConnectionFactory
- MQXAQueueConnectionFactory
- MQXATopicConnectionFactory

これらのさまざまなタイプの接続ファクトリーについて詳しくは、686 ページの『管理ツールを使用した JMS および Jakarta Messaging オブジェクトの構成』 を参照してください。

関連概念
IBM MQ メッセージング・プロバイダー

PROVIDERVERSION 通常モード

通常モードは、IBM MQ キュー・マネージャーのすべての機能を使用して JMS を実装します。 このモードは、JMS 2.0 の API と機能を使用するように最適化されています。

以下のフローチャートは、通常モードの接続を作成できるかどうか判別するために JMS クライアントが実行する検査を示しています。

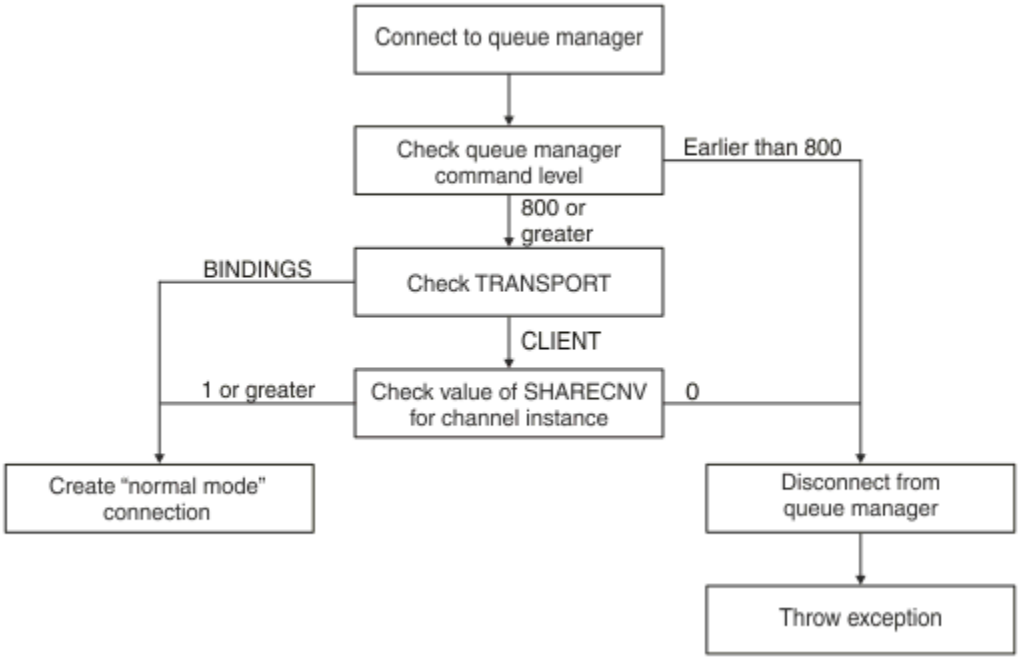


図 *90. PROVIDERVERSION* 通常モード

接続ファクトリー設定で指定されたキュー・マネージャーのコマンド・レベルが 800 以上で、接続ファクトリーの **TRANSPORT** プロパティーが BINDINGS に設定されている場合は、その他のプロパティーを検査することなく、通常モードの接続が作成されます。

接続ファクトリー設定で指定されたキュー・マネージャーのコマンド・レベルが 800 以上で、**TRANSPORT** プロパティーが CLIENT に設定されている場合は、サーバー接続チャネルの **SHARECNV** プロパティーも検査されます。 この検査が必要になるのは、IBM MQ メッセージング・プロバイダー通常モードで共用会話機能を使用するためです。 したがって、通常モードの接続の試みを成功させるには、共用可能な会話の数を制御する **SHARECNV** プロパティーの値が 1 以上でなければなりません。

フローチャートに示されたすべての検査が成功すると、キュー・マネージャーへの通常モードの接続が作成され、JMS 2.0 のすべての API および機能 (つまり、非同期送信、遅延送達、および共有サブスクリプション) を使用できるようになります。

通常モードの接続を作成する試みの失敗は、以下のいずれかの理由によります。

- 接続ファクトリー設定で指定されたキュー・マネージャーのコマンド・レベルが 800 未満である。 この場合、`createConnection` メソッドは、例外 `JMSFMQ0003` で失敗します。
- サーバー接続チャネル上の **SHARECNV** プロパティーは、`0` に設定されます。 プロパティーの値が 1 以上でない場合、`createConnection` メソッドは例外 `JMSCC5007` で失敗します。

関連資料

IBM MQ classes for JMS オブジェクトのプロパティー間の依存関係

DEFINE CHANNEL (SHARECNV プロパティー)

TRANSPORT

***PROVIDERVERSION* 制限付き通常モード**

制限付きの通常モードでは、JMS 2.0API が使用されますが、共有サブスクリプション、遅延送達、非同期送信などの新しい IBM MQ 8.0 またはそれ以降のフィーチャーは使用されません。

以下のフローチャートは、制限付き通常モードの接続を作成できるかどうか判別するために JMS クライアントが実行する検査を示しています。

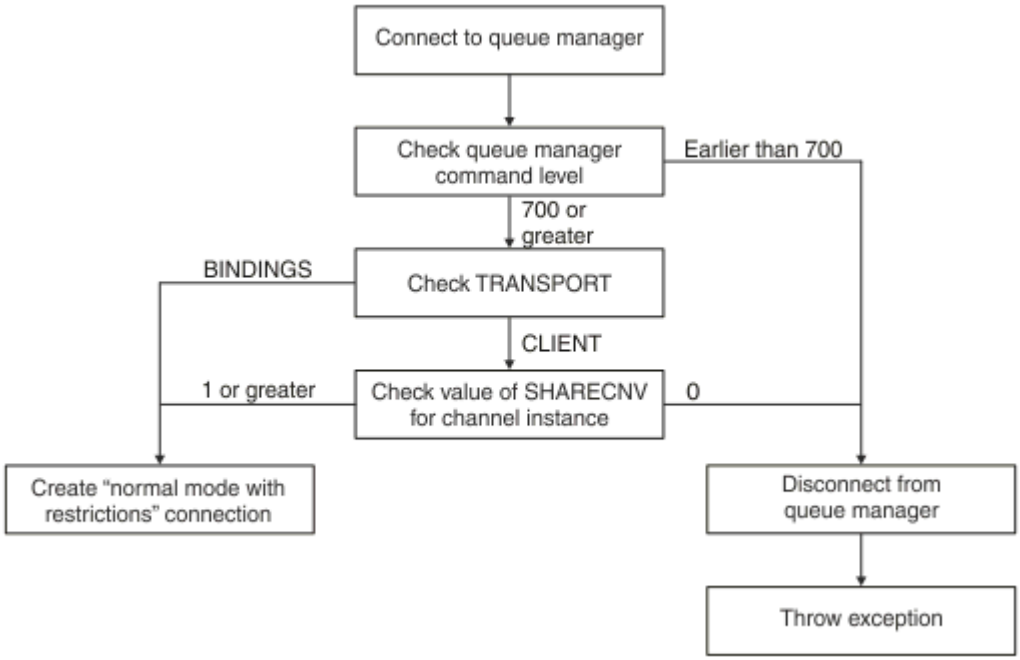


図 *91. PROVIDERVERSION* 制限付き通常モード

接続ファクトリー設定で指定されたキュー・マネージャーのコマンド・レベルが 700 以上で、接続ファクトリーの **TRANSPORT** プロパティーが BINDINGS に設定されている場合は、その他のプロパティーを検査することなく、通常モードの接続が作成されます。

接続ファクトリー設定で指定されたキュー・マネージャーのコマンド・レベルが 700 以上で、**TRANSPORT** プロパティーが CLIENT に設定されている場合は、サーバー接続チャネルの **SHARECNV** プロパティーも検査されます。 この検査が必要になるのは、IBM MQ メッセージング・プロバイダー制限付き通常モードで共用会話機能を使用するためです。 したがって、制限付き通常モードの接続の試みを成功させるには、共用可能な会話の数を制御する **SHARECNV** プロパティーの値が 1 以上でなければなりません。

フローチャートに示されたすべての検査が成功すると、キュー・マネージャーへの制限付き通常モードの接続が作成され、JMS 2.0 API を使用できます。しかし、非同期送信、遅延送達、または共有サブスクリプションの機能は使用できません。

制限付き通常モードの接続を作成する試みの失敗は、以下のいずれかの理由によります。

- 接続ファクトリー設定で指定されたキュー・マネージャーのコマンド・レベルが 700 未満である。 この場合、`createConnection` メソッドは、例外 `JMSFCC5008` で失敗します。
- サーバー接続チャネル上の **SHARECNV** プロパティーは、`0` に設定されます。 プロパティーの値が 1 以上でない場合、`createConnection` メソッドは例外 `JMSCC5007` で失敗します。

関連資料

IBM MQ classes for JMS オブジェクトのプロパティー間の依存関係

DEFINE CHANNEL (SHARECNV プロパティー)

TRANSPORT

PROVIDERVERSION 移行モード

移行モードの場合、IBM MQ classes for JMS は、IBM WebSphere MQ 6.0 に付属の機能やアルゴリズムに似た機能やアルゴリズムを使用します。 それには、キューに入っているパブリッシュ/サブスクライブ、クライアント・サイドに実装される選択項目、非マルチプレクス・チャネル、リスナーの実装に使用するポーリングなどがあります。

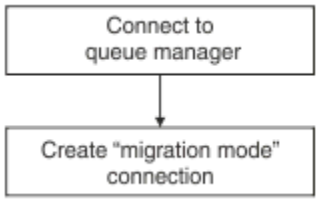


図 92. *PROVIDERVERSION* 移行モード

IBM MQ Enterprise Transport バージョン 6.0 を使用して WebSphere Message Broker 6.0 または WebSphere Message Broker 6.1 に接続する場合は、マイグレーション・モードを使用する必要があります。

移行モードを使用して IBM MQ 8.0 キュー・マネージャーに接続できますが、IBM MQ classes for JMS キュー・マネージャーの新機能 (先読みやストリーミングなど) はいずれも使用されません。 IBM MQ 8.0 以降のクライアントが分散プラットフォーム上の IBM MQ 8.0 以降のキュー・マネージャー z/OS または IBM MQ for z/OS 8.0 以降のキュー・マネージャー に接続している場合、メッセージ選択はクライアント・システムではなくキュー・マネージャーによって行われます。

IBM MQ メッセージング・プロバイダー移行モードが指定されていて、IBM MQ classes for JMS で JMS 2.0 のいずれかの API を使用しようとした場合、API メソッド呼び出しは例外 JMSCC5007 で失敗します。

関連資料

IBM MQ classes for JMS オブジェクトのプロパティー間の依存関係

TRANSPORT

PROVIDERVERSION の指定なし

接続ファクトリーの **PROVIDERVERSION** プロパティーが指定なしの場合、JMS クライアントでは、アルゴリズムを使用して、キュー・マネージャーへの接続に使う操作モードが判別されます。 以前のバージョンの IBM MQ classes for JMS を使用して JNDI 名前空間に作成された接続ファクトリーが、新しいバージョンの IBM MQ classes for JMS で使用されるときには、指定なしの値を取ります。

PROVIDERVERSION プロパティーが指定なしの場合、createConnection メソッドが呼び出されるときにアルゴリズムが使用されます。 このアルゴリズムは、接続ファクトリーの複数のプロパティーを検査して、IBM MQ メッセージング・プロバイダー通常モード、制限付き通常モード、または IBM MQ メッセージング・プロバイダー移行モードのどれが必要かを判別します。 常に通常モードが最初に試行され、次に制限付き通常モードが試行されます。 どちらのタイプの接続も作成できない場合は、JMS クライアントはキュー・マネージャーから切断してから、キュー・マネージャーに再接続し、移行モードの接続を試行します。

BROKERVER、BROKERQMGR、PSMODE、BROKERCONQ プロパティーの検査

図 1 に示されているように、最初に **BROKERVER** プロパティーの値が検査されます。

BROKERVER プロパティーが V1 に設定されている場合、 図 2 に示すように **TRANSPORT** プロパティーが次にチェックされます。 ただし、**BROKERVER** プロパティが V2 に設定されている場合は、図 1 に示す追加のチェックは、**TRANSPORT** プロパティがチェックされる前に実行されます。

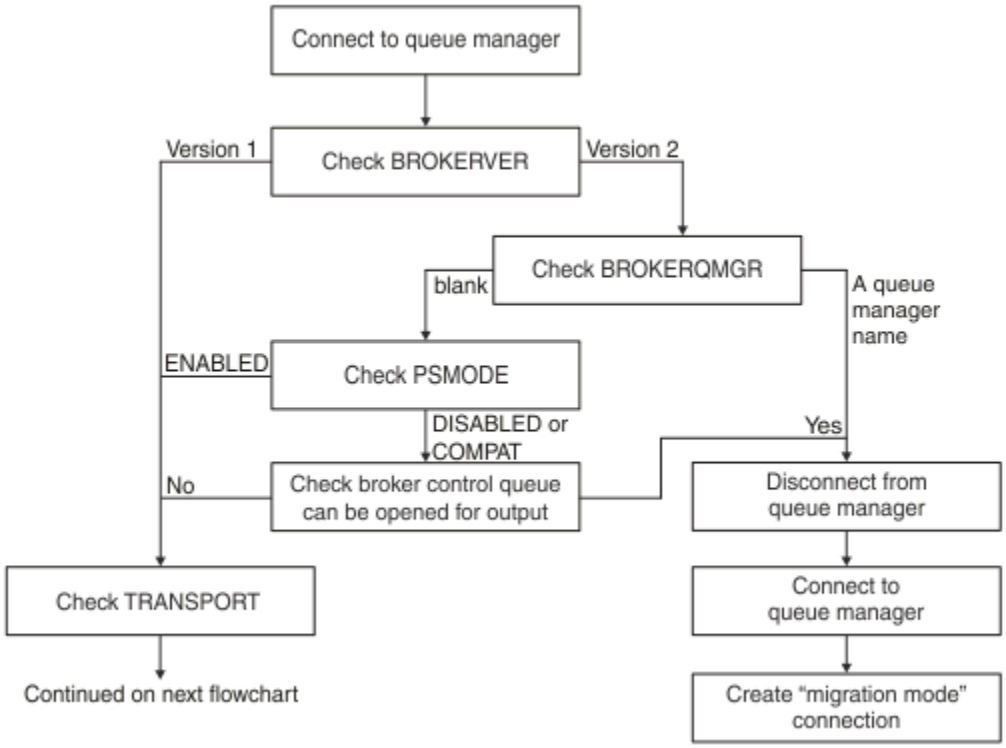


図 *93. PROVIDERVERSION* の指定なし

BROKERVER プロパティーが V2 に設定されている場合、通常モードの接続を行えるようにするには、**BROKERQMGR** プロパティーがブランクでなければなりません。 さらに、キュー・マネージャー上の **PSMODE** 属性が ENABLED に設定されているか、 **BROKERCONQ** プロパティーで指定されたブローカー制御キューが出力用に開くことができないようになっていなければなりません。

通常モード接続でプロパティー値が必要に応じて設定されている場合は、次にチェック・マークを付けると、 図 2 に示すように **TRANSPORT** プロパティーに移動します。

通常モード接続にとって必要なプロパティー値が設定されていない場合は、JMS クライアントはキュー・マネージャーから切断してから再接続して、移行モードの接続を作成します。 これは次のような場合に行われます。

- **BROKERQMGR** プロパティーがブランクで、キュー・マネージャー上の **PSMODE** 属性が COMPAT または DISABLED に設定されており、**BROKERCONQ** プロパティーで指定されたブローカー制御キューを出力用に開くことができる (つまり、出力の MQOPEN が正常に実行される) 場合。
- **BROKERQMGR** プロパティーでキュー名が指定されている場合。

TRANSPORT プロパティーとコマンド・レベルの検査

図 2 には、**TRANSPORT** プロパティーと、キュー・マネージャーのコマンド・レベルを対象に行う検査が示されています。

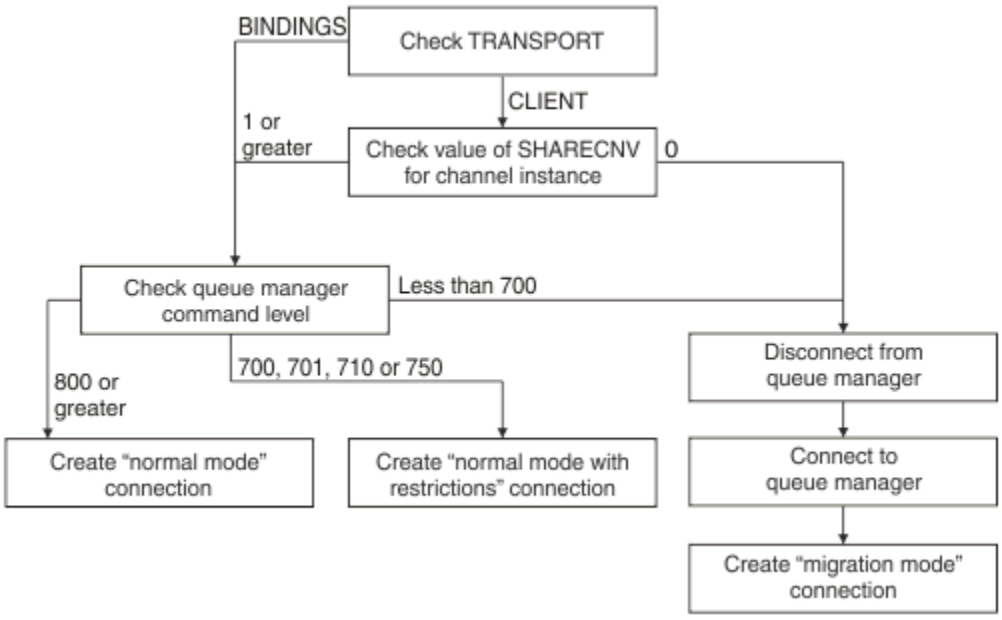


図 *94. PROVIDERVERSION* の指定なし *(続き)*

以下のいずれかの場合に、通常モードの接続が作成されます。

- 接続ファクトリーの **TRANSPORT** プロパティーが BINDINGS に設定され、キュー・マネージャーのコマンド・レベルが 800 以上の場合。
- **TRANSPORT** プロパティーが CLIENT に設定され、サーバー接続チャネル上の **SHARECNV** プロパティーの値が 1 以上で、キュー・マネージャーのコマンド・レベルが 800 以上の場合。

キュー・マネージャーのコマンド・レベルが 750 の場合、キュー・マネージャーへの制限付き通常モード接続が作成されます。

TRANSPORT プロパティーが CLIENT に設定され、サーバー接続チャネル上の **SHARECNV** プロパティーの値が 0 の場合も、移行モードの接続が作成されます。

関連資料

IBM MQ classes for JMS オブジェクトのプロパティー間の依存関係
ALTER QMGR (PSMODE 属性)
BROKERCONQ
BROKERQMGR
BROKERVER
DEFINE CHANNEL (SHARECNV プロパティー)
TRANSPORT

WebSphere Application Server でのプロバイダー・バージョン情報の構成

WebSphere Application Server でプロバイダー・バージョン情報を構成するときに、管理コンソールまたは wsadmin コマンドを使用できます。

手順

WebSphere Application Server で IBM MQ 接続ファクトリーまたはアクティベーション・スペック・オブジェクトのプロバイダー・バージョン情報を構成するには、 WebSphere Application Server 製品資料の詳細情報へのリンクについて、 「関連情報」 を参照してください。

WebSphere Application Server 8.5.5 の関連情報

IBM MQ メッセージング・プロバイダー接続ファクトリー設定

createWMQConnectionFactory コマンド
IBM MQ メッセージング・プロバイダー・アクティベーション・スペック設定
createWMQActivationSpec コマンド
WebSphere Application Server 8.0.0 の関連情報
IBM MQ メッセージング・プロバイダー接続ファクトリー設定
createWMQConnectionFactory コマンド
IBM MQ アクティベーション・スペック設定
createWMQActivationSpec コマンド
WebSphere Application Server 7.0.0 の関連情報
IBM MQ メッセージング・プロバイダー接続ファクトリー設定
createWMQConnectionFactory コマンド
IBM MQ アクティベーション・スペック設定
createWMQActivationSpec コマンド

WebSphere Application Server 永続サブスクリプションの削除

IBM MQ メッセージング・プロバイダーを WebSphere Application Server 7.0 および WebSphere Application Server 8.0 で使用する場合、活動化仕様にバインドされたメッセージ駆動型 Bean アプリケーションによって作成された永続サブスクリプションは削除されません。 永続サブスクリプションは、IBM MQ Explorer または IBM MQ コマンド・ライン・ユーティリティーのいずれかを使用して削除できます。

本タスクについて

永続サブスクリプションを除去するメッセージ駆動型 Bean アプリケーションは、リスナー・ポートまたはアクティベーション・スペックのいずれかを使用するように構成できます。ただし、アプリケーションが、IBM MQ メッセージング・プロバイダー通常モード を使用して IBM MQ に接続する WebSphere Application Server 7.0 または WebSphere Application Server 8.0 インスタンス内で実行されている場合に限ります。

メッセージ駆動型 Bean アプリケーションがリスナー・ポートにバインドされている場合、IBM MQ メッセージング・プロバイダーが、アプリケーションの初回始動時にアプリケーション用の永続サブスクリプションを作成します。 永続サブスクリプションは、メッセージ駆動型 Bean アプリケーションがアプリケーション・サーバーからアンインストールされ、アプリケーション・サーバーが再始動すると削除されます。

活動化仕様にバインドされたメッセージ駆動型 Bean アプリケーションの動作方法は、若干異なります。アプリケーション用の永続サブスクリプションは、アプリケーションの初回始動時に作成されます。 ただし、永続サブスクリプションは、アプリケーションがアンインストールされて、アプリケーション・サーバーが再始動しても削除されません。

これにより、 WebSphere Application Server システムにインストールされなくなったアプリケーションの IBM MQ パブリッシュ/サブスクライブ・エンジンに多数の永続サブスクリプションが残る可能性があります。 このようなサブスクリプションは「オーファン・サブスクリプション」と呼ばれ、パブリッシュ/サブスクライブ・エンジンの実行時にキュー・マネージャー上で問題を生じさせる可能性があります。

トピックに関するメッセージがパブリッシュされると、IBM MQ パブリッシュ/サブスクライブ・エンジンが、そのトピックに登録されている各永続サブスクリプション用にメッセージのコピーを作成し、内部キューに配置します。 その永続サブスクリプションを使用しているアプリケーションは、この内部キューからメッセージを取り出してコンシュームします。

その永続サブスクリプションを使用していたメッセージ駆動型 Bean アプリケーションがもはやインストールされていない場合であっても、そのアプリケーションに対してパブリッシュされたメッセージのコピーは引き続き作成されます。 しかし、これらのメッセージは処理されることがなく、つまり削除されることのない大量のメッセージが内部キューに残される可能性があります。

始める前に

IBM MQ パブリッシュ/サブスクライブ・エンジンに登録されたサブスクリプションには、関連付けられたサブスクリプション名があります。

活動化仕様にバインドされたメッセージ駆動型 Bean 用の WebSphere Application Server IBM MQ メッセージング・プロバイダーによって作成された永続サブスクリプションには、以下のフォーマットのサブスクリプション名が付けられます。

```
JMS:queue manager name:client identifier:subscription name
```

説明

queue manager name
これは、パブリッシュ/サブスクライブ・エンジンが実行されている IBM MQ キュー・マネージャーの名前です。

client identifier
これは、メッセージ駆動型 Bean がバインドされた活動化仕様のクライアント ID プロパティーの値です。

subscription name
これは、メッセージ駆動型 Bean アプリケーションが使用するよう構成されている活動化仕様の、活動化仕様プロパティーのサブスクリプション名の値です。

例えば、キュー・マネージャー testQM に接続するためにセットアップされた活動化仕様があるとします。この活動化仕様のプロパティー・セットは次のようになります。

- クライアント ID = testClientID
- サブスクリプション名 = durableSubscription1

永続サブスクリプションを取得するメッセージ駆動型 Bean がこの活動化仕様にバインドされている場合、キュー・マネージャー testQM の IBM MQ パブリッシュ/サブスクライブ・エンジンで、以下のサブスクリプション名を持つサブスクリプションが作成されます。

- JMS:testQM:testClientID:durableSubscription1

所定のキュー・マネージャー用の IBM MQ パブリッシュ/サブスクライブ・エンジンに登録されているサブスクリプションは、以下の 2 つの方法のいずれかで確認できます。

- 最初のオプションは、MQ エクスプローラーを使用する方法です。 パブリッシュ/サブスクライブ作業に使用されているキュー・マネージャーに MQ エクスプローラーが接続されている場合、ナビゲーション・ペインの IBM WebSphere MQ ->*queue manager name*-> Subscriptions 項目をクリックすると、パブリッシュ/サブスクライブ・エンジンに現在登録されているサブスクライバーのリストを表示できます。
- パブリッシュ/サブスクライブ・エンジンに登録されているサブスクリプションを表示するもう 1 つの方法は、IBM MQ コマンド・ライン・ユーティリティー **runmqsc** を使用して、コマンド **display sub** を実行することです。 これを行うには、コマンド・プロンプトを表示し、 *WebSphere MQ*\bin ディレクトリーに移動して、以下のコマンドを入力して **runmqsc** を開始します。

  ```
  – runmqsc queue manager name
  ```

  runmqsc ユーティリティーが始動したら、以下のコマンドを入力して、**runmqsc** が接続されているキュー・マネージャーで実行しているパブリッシュ/サブスクライブ・エンジンに現在登録されているすべての永続サブスクリプションをリストします。

  ```
  – display sub(*) durable
  ```

パブリッシュ/サブスクライブ・エンジンに登録されている永続サブスクリプションがまだアクティブであるかどうかを調べるには、以下のようにします。

1. パブリッシュ/サブスクライブ・エンジンに登録されている永続サブスクリプションのリストを生成します。
2. 各永続サブスクリプションで、以下を行います。
   - 永続サブスクライバーのサブスクリプション名を調べ、*client identifier* と *subscription name* の値をメモします。
   - このパブリッシュ/サブスクライブ・エンジンに接続されている WebSphere Application Server システムを調べます。 *client identifier* 値に一致するクライアント ID プロパティーと、*subscription name* と

一致するサブスクリプション名プロパティーを持つ活動化仕様が定義されていないかどうかを調べます。

- IBM MQ サブスクリプション名の *client identifier* および *subscription name* フィールドに一致するクライアント ID およびサブスクリプション名プロパティーを持つアクティベーション・スペックが見つからない場合、この永続サブスクリプションを使用するアクティベーション・スペックはありません。その永続サブスクリプションは削除できます。
- 永続サブスクリプション名に一致する活動化仕様が定義されている場合、必要な最終検査は、この活動化仕様を使用しているメッセージ駆動型 Bean アプリケーションがないかどうかを調べることです。 そのためには、次のようにします。
  - 現在確認している永続サブスクリプションを取得している活動化仕様の JNDI 名をメモします。
  - インストールされている各メッセージ駆動型 Bean アプリケーションの WebSphere Application Server 管理コンソールで「構成」ペインを起動します。
  - 「構成」ペイン内のメッセージ駆動型 Bean リスナーのバインディング・リンクをクリックします。
  - メッセージ駆動型 Bean アプリケーションに関する情報を示した表が表示されます。 「バインディング」列で活動化仕様ラジオ・ボタンが選択され、「ターゲット・リソース JNDI 名」フィールドに永続サブスクリプションを取得した活動化仕様の JNDI 名が指定されている場合、そのサブスクリプションはまだ使用されており、削除できません。
  - 活動化仕様を使用しているメッセージ駆動型 Bean アプリケーションが見つからない場合は、その永続サブスクリプションを削除できます。

手順

「オーファン化した」永続サブスクリプションを特定した場合、IBM MQ Explorer か IBM MQ コマンド・ライン・ユーティリティー **runmqsc** のいずれかを使用して削除できます。

「オーファン化した」永続サブスクリプションを IBM MQ Explorer を使用して削除する手順は、以下のとおりです。

1. サブスクリプション用の項目を強調表示します。
2. エントリーを右クリックし、「**削除 ...**」 を選択します。 表示されます。 確認ウィンドウが表示されます。
3. 確認ウィンドウに表示されているサブスクリプション名が正しいことを確認し、「**はい**」をクリックします。

IBM MQ Explorer はパブリッシュ/サブスクライブ・エンジンからサブスクリプションを削除し、それに関連するすべての内部リソース (その永続サブスクリプションが登録されていたトピックに関してパブリッシュされた未処理のメッセージなど) をクリーンアップします。

IBM MQ コマンド・ライン・ユーティリティー **runmqsc** を使用して「孤立した」永続サブスクリプションを削除するには、コマンド **delete sub** を実行する必要があります。

1. コマンド・プロンプト・セッションを開きます。
2. *IBM MQ*\bin ディレクトリーに移動します
3. 以下のコマンドを入力して **runmqsc** を始動します。

```
runmqsc queue manager name
```

4. **runmqsc** ユーティリティーが始動したら、以下を入力します。

```
delete sub(Subscription name)
```

ここで、*Subscription name* は、永続サブスクリプションの次の形式のサブスクリプション名です。

- JMS:*queue manager name*:*client identifier*:*subscription name*

Managed File Transfer の構成

Managed File Transfer のフィーチャーは、インストール後に構成可能です。

IBM MQ 高可用性ソリューションを利用して、 Managed File Transfer 構成の回復力を向上させることができます。 エージェントが複製データ・キュー・マネージャー (RDQM) を使用する場合は、浮動 IP アドレス機能を使用するように構成する必要があります。 これは、エージェントが同じ IP アドレスを使用して、現在実行中の 3 つの RDQM インスタンスのいずれかと通信し、フェイルオーバー時に自動的に再接続することを意味します (RDQM 高可用性 および 浮動 IP アドレスの作成と削除を参照してください)。 複数インスタンス・キュー・マネージャー・ソリューションを使用する場合、アプリケーションは、各インスタンスとの通信に異なる IP アドレスを使用します。これは、フェイルオーバー時にクライアント再接続によって処理されます (複数インスタンス・キュー・マネージャー および チャネルとクライアントの再接続を参照)。

関連概念

Managed File Transfer の使用のヒント

関連タスク

MFT リソースのモニター

ユーザー出口での MFT のカスタマイズ

MQMFTCredentials.xml の構成

Managed File Transfer の保護

MFT で実行するプログラムの指定

トラブルシューティング Managed File Transfer

Managed File Transfer の管理

関連資料

MFT コマンド

MFTagent.properties ファイル

MFT のリカバリーと再始動

MFT 構成オプション (Multiplatforms)

Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティー・ファイルを提供します。 これらのプロパティー・ファイルは、製品のインストール時に定義される構成ディレクトリーにあります。

それぞれがディレクトリーとプロパティー・ファイルのセットを含む構成オプションのセットを複数持つことができます。 コマンド行に別の値を明示的に指定しない限り、すべての Managed File Transfer コマンドについて、これらのプロパティー・ファイルで定義されている値がデフォルトのパラメーターとして使用されます。

使用中のデフォルトの構成オプションのセットを変更するには、**fteChangeDefaultConfigurationOptions** コマンドを使用します。 個々のコマンドに使用している構成オプションのセットを変更するには、任意の Managed File Transfer コマンドで **-p** パラメーターを使用できます。

構成オプションのセットの名前は、調整キュー・マネージャーの名前です。この名前は変更しないことをお勧めします。 構成オプションのセットの名前を変更することは可能ですが、`config` ディレクトリーおよび `logs` ディレクトリーの名前を変更する必要があります。 以下の例では、構成オプションのセットの名前を *coordination_qmgr_name* と表記します。

構成オプションのディレクトリー構造

製品を構成すると、構成ディレクトリー内にディレクトリーとプロパティー・ファイルが以下に示す構造で作成されます。 以下のコマンドを使用すると、ディレクトリーとプロパティー・ファイルを変更するこ

ともできます。**fteSetupCoordination**、**fteSetupCommands**、**fteChangeDefaultConfiguration**、および **fteCreateAgent**。

```
MQ_DATA_PATH/mqft/
    config/
       coordination_qmgr_name/
            coordination.properties
            command.properties
            agents/
               agent_name/
                    agent.properties
                    exits
            loggers/
               logger_name
                    logger.properties
    installations/
       installation_name/
            installation.properties
```

coordination_qmgr_name ディレクトリーは、構成オプションのディレクトリーです。 構成ディレクトリー内に複数の構成オプション・ディレクトリーを置くことができます。 *agent_name* ディレクトリーはエージェント・ディレクトリーです。 このディレクトリーには agent.properties ファイルの他に exits ディレクトリーが含まれており、これはユーザー出口ルーチン、および **fteCreateBridgeAgent** コマンドおよび **fteCreateCDAgent** コマンドで生成されるさまざまな XML ファイルのためのデフォルトの場所です。 構成オプションのセットの agents ディレクトリーには、複数のエージェント・ディレクトリーが存在する場合があります。

プロパティー・ファイル

installation.properties
installation.properties ファイルは、構成オプションのデフォルト・セットの名前を指定します。 この項目は、Managed File Transfer を、使用する構成が含まれる一連の構造化されたディレクトリーおよびプロパティー・ファイルに向けます。 一般に、構成オプションのセット名は、関連付けられている調整キュー・マネージャーの名前です。 installation.properties ファイルについて詳しくは、MFTinstallation.properties ファイルを参照してください。

coordination.properties
coordination.properties ファイルは、調整キュー・マネージャーとの接続の詳細を指定します。 複数の Managed File Transfer インストール済み環境で同じ調整キュー・マネージャーが共用される場合があるため、共用ドライブ上の共通の coordination.properties ファイルへのシンボリック・リンクを使用できます。 coordination.properties ファイルについて詳しくは、MFTcoordination.properties ファイルを参照してください。

command.properties
MFT command.properties ファイルには、ユーザーがコマンドを発行するときに接続先となるコマンド・キュー・マネージャーと、Managed File Transfer がそのキュー・マネージャーと情報のやり取りをするのに必要な情報を指定します。 command.properties ファイルについて詳しくは、MFTcommand.properties ファイルを参照してください。

agent.properties
各 Managed File Transfer Agent には独自のプロパティー・ファイル agent.properties があります。 このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。 agent.properties ファイルには、エージェントの動作を変更するプロパティーを含めることもできます。 agent.properties ファイルについて詳しくは、 MFT agent.properties ファイルを参照してください。

logger.properties
logger.properties ファイルは、ロガーの構成プロパティーを指定します。 logger.properties ファイルについて詳しくは、MFT ロガー構成プロパティーを参照してください。

プロパティー・ファイルとコード・ページ

Java の制限のため、すべての Managed File Transfer プロパティー・ファイルの内容は米国英語のままでなければなりません。 米国英語以外のシステムでプロパティー・ファイルを編集する場合、Unicode エスケープ・シーケンスを使用する必要があります。

関連資料

MFT 用の SSL/TLS プロパティー

MFT Java システム・プロパティー

fteChangeDefaultConfigurationOptions

fteSetupCommands: MFT command.properties ファイルの作成

fteSetupCoordination

fteCreateAgent

z/OS での MFT 構成オプション

z/OS の Managed File Transfer 構成オプションは、分散プラットフォームの場合のオプションと同じです。

マルチプラットフォーム の構成オプションの詳細については、719 ページの『MFT 構成オプション (Multiplatforms)』 を参照してください。

z/OS では、構成の場所が BFG_DATA 環境変数によって定義されます。 BFG_DATA によって参照される z/OS UNIX System Services・ディレクトリーの下に構成がまだ存在しない場合は、MFT コマンド PDSE ライブラリー・データ・セットの BFGCUSTM JCL スクリプトによって、構成を作成するために必要なジョブが生成されます。 次いで、生成されたこれらのジョブを実行すると、構成が作成されます。 アクセス可能な既存のディレクトリーを参照する BFG_DATA に、構成の作成は依存しています。

Multiplatforms と z/OS の両方で使用可能な同じ **fte** コマンドを使用して、構成を作成および保守することもできます。 **fte** コマンドのリストについては、MFT コマンドを参照してください。

関連概念

719 ページの『MFT 構成オプション (Multiplatforms)』
Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティー・ファイルを提供します。 これらのプロパティー・ファイルは、製品のインストール時に定義される構成ディレクトリーにあります。

738 ページの『エージェントの作成』
エージェント固有の PDSE (例えば *user.MFT.AGENT1*) を作成するには、PDSE をコピーする必要があります。 以前のエージェント構成またはロガー構成があれば、その構成から PDSE をコピーします。 初めて構成を作成する場合は、MFT に用意されている PDSE をコピーします。

736 ページの『調整キュー・マネージャーの定義』
Managed File Transfer では、調整キュー・マネージャーとして機能するキュー・マネージャーを作成する必要があります。

関連タスク

z/OS での MQMFTCredentials.xml の構成

739 ページの『MFT の既存のエージェントまたはロガーのコマンド・データ・セットの更新 (z/OS)』
Managed File Transfer コマンド・テンプレート・データ・セットから作成される Managed File Transfer コマンド PDSE ライブラリー・データ・セットを更新できます。

Redistributable Managed File Transfer components のダウンロードおよび構成

Redistributable Managed File Transfer package では、Redistributable Managed File Transfer Agent を既存の IBM MQ インフラストラクチャーに接続するように構成し、IBM MQ をインストールしていないユーザーがファイルを転送できます。 IBM MQ 9.3.0 以降、再配布可能パッケージには Redistributable Managed File Transfer Logger も含まれています。

始める前に

Redistributable Managed File Transfer Agent および Redistributable Managed File Transfer Logger の再配布可能ライセンス条項については、IBM MQ 再配布可能コンポーネントを参照してください。

Redistributable Managed File Transfer package コンポーネントは Managed File Transfer の機能を提供しますが、次の例外があります。

- Redistributable Managed File Transfer Agent の場合、調整キュー・マネージャー、コマンド・キュー・マネージャー、およびエージェント・キュー・マネージャーへのバインディング・モード接続はサポートされないため、クライアント・モード接続を使用する必要があります。 IBM MQ の一部としてインストールされている Managed File Transfer を使用する場合は、コマンドを発行するときにオプションのパラメーター (キュー・マネージャーのホスト、ポート、名前、およびチャネル名) を指定する必要があります。
- V 9.3.0 Redistributable Managed File Transfer Logger は、調整キュー・マネージャーのみへのクライアント・モードの接続に FILE タイプのロガーのみをサポートします。 データベース・ロガーの調整キュー・マネージャーへのクライアント・モード接続はサポートされません。 バインディング・モード接続が必要な場合は、IBM MQ の標準インストール環境を使用する必要があります。
- V 9.3.0 IBM MQ 9.3.0 以降では、 **fteCreateCDAgent.cmd** コマンドは含まれません。 使用可能コマンドの完全なリストについては、インストールされる MFT コマンド・セットを参照してください。
- Managed File Transfer Connect:Direct® は、サポートされていません。
- IBM MQ Explorer は、含まれていません。

Windows Redistributable Managed File Transfer Agent を使用するには、 Microsoft から入手できる Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 ライブラリーをシステムにインストールする必要があります。 The latest supported Visual C++ downloads を参照してください。

V 9.3.0 IBM MQ 9.3.0 以降、 Redistributable Managed File Transfer Logger には Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 ライブラリーも必要になりました。

注 : Advanced Message Security は、Redistributable Managed File Transfer package ではサポートされません。

このタスクについて

オプションで、 Redistributable Managed File Transfer package をダウンロードし、既存の IBM MQ インフラストラクチャーに接続するように Redistributable Managed File Transfer Agent を構成することができます。これにより、ユーザーは、 Managed File Transfer 機能を取得するために IBM MQ をインストールする必要なく、ローカル環境と既存の IBM MQ インフラストラクチャーの間でファイルを転送できます。

V 9.3.0 IBM MQ 9.3.0 以降、Redistributable Managed File Transfer package には Redistributable Managed File Transfer Logger も含まれるようになりました。これによって、ユーザーはファイル・ロガーをセットアップし、クライアント・モードで調整キュー・マネージャーに接続できます。

手順

1. IBM MQ 再配布可能 Managed File Transfer エージェント・パッケージ を Fix Central からダウンロードします。
   a) 使用するオペレーティング・システム用のパッケージを選択します。
   アーカイブまたは .zip ファイルの名前は、ファイルの内容と相当する保守レベルを示しています。ファイル名の形式は次のとおりです。
   - Windows *V.R.M.F*-IBM-MQFA-Redist-Win64
   - Linux *V.R.M.F*-IBM-MQFA-Redist-LinuxX64
   - Linux *V.R.M.F*-IBM-MQFA-Redist-LinuxS390X

- Linux *V.R.M.F*-IBM-MQFA-Redist-LinuxPPC64LE

ここで、*V.R.M.F* はバージョン番号です(例:9.2.0.0 または 9.2.1.0)。

b) パッケージの解凍先ディレクトリーを決定します。以下に例を示します。

- Windows C:\MFTZ
- Linux /home/MFTZ

2. ダウンロードしたパッケージのコンテンツを解凍します。

- Windows Windows で、Windows エクスプローラー・ツールを使用して解凍します。
- Linux Linux で、次のように解凍して untar します。

```
gunzip V.R.M.F-IBM-MQFA-Redist-LinuxX64.tar.gz
```

それから、次のようにします。

```
tar xvf V.R.M.F-IBM-MQFA-Redist-LinuxX64.tar
```

ここで、*V.R.M.F* はバージョン番号です(例:9.3.0.0 または 9.3.1.0)。

以下のディレクトリーが作成されます。

- Windows Linux bin:必要なすべての MFT コマンドが含まれます
- Windows bin64: Windows64 ビット OS サポートに必須ライブラリーが含まれています
- Windows Linux java:IBMJRE および IBM MQ ライブラリーが含まれています
- Windows Linux licenses:ライセンス・ファイルを含みます
- Windows V 9.3.0 META-INF:コード署名情報を持つファイルが含まれています
- Windows Linux mqft:Ant サポートおよびコア MFT 機能のサポートに必要な ant ディレクトリー lib が含まれています
- Windows Linux swtag:マシン上のインストール済み環境を識別するためにライセンス・マネージャーが必要とする swidtag ファイルが含まれます

次のタスク

これで、Managed File Transfer Agent を構成する準備が整いました。 次の手順については、723 ページの『Redistributable Managed File Transfer Agent の初期構成の作成』を参照してください。

V 9.3.0 IBM MQ 9.3.0 以降、Managed File Transfer Logger も構成できるようになりました。 ロガーを構成するための次のステップについては、726 ページの『Redistributable Managed File Transfer Logger の初期構成の作成』を参照してください。

関連資料

Redistributable Managed File Transfer components の構成時に起こる可能性のあるエラー

Windows Linux Redistributable Managed File Transfer Agent の初期構成の作成

Managed File Transfer Agent を、既存の IBM MQ 構成に接続するように構成できます。

始める前に

Redistributable Managed File Transfer Agent・パッケージのコンテンツをダウンロードして解凍します。 詳しくは、721 ページの『Redistributable Managed File Transfer components のダウンロードおよび構成』を参照してください。

このタスクについて

最初に、Redistributable Managed File Transfer Agent が必要とする環境を作成します。 次に、IBM MQ サーバーで実行されているキュー・マネージャーとの接続をセットアップし、その後、エージェントおよびエージェント・キュー・マネージャーを構成してからエージェントを開始して検証します。

V 9.3.0 IBM MQ 9.3.0 以降は、作成した環境は Redistributable Managed File Transfer Logger と共有されます。 詳しくは、726 ページの『Redistributable Managed File Transfer Logger の初期構成の作成』を参照してください。

手順

1. Redistributable Managed File Transfer Agent の環境を作成します。

   fteCreateEnvironment コマンドを実行すると、MFT Agent の構成情報を使用して MFT データ・ディレクトリーが作成されます。 ダウンロードした Redistributable Managed File Transfer Agent コンポーネントを抽出したときに、作成された bin ディレクトリー内にあることを確認します。 以下のコマンドを実行します。

   - Windows

     ```
     fteCreateEnvironment.cmd -d datapath location
     ```

   - Linux

     ```
     . ./fteCreateEnvironment -d datapath location
     ```

   このコマンドには以下のオプション・パラメーターがあります。

   -d
   このパラメーターでは、MFT 構成の作成、保管、保守を行うデータ・パスの場所を指定します。 データの場所を指定せずに **fteCreateEnvironment** を実行すると、 Redistributable Managed File Transfer Agent が解凍された場所にディレクトリー `mftdata` が作成されます。

   注 : 再配布可能エージェントが Windows サービスとして実行される場合、サービスを機能させるには、**BFG_DATA** 環境変数をシステム環境で設定する必要があります。

   -n *installation name*
   このパラメーターでは、IBM MQ のインストール名 (固有の名前) を指定します。

   このパラメーターを使用すると良い状況としては、例えば次のような状況があります。

   - エージェントがクライアント・モードだけでキュー・マネージャーに接続するという既存の構成で、再配布可能パッケージを使用して新機能または新フィーチャーを素早くテストしたい場合 (このパラメーターは、バインディング・モードでキュー・マネージャーに接続するように構成したエージェントには適用されません)。
   - Managed File Transfer の標準インストール環境を Redistributable Managed File Transfer Agent パッケージにマイグレーションして、標準インストール環境で作成した構成と同じ構成を使用したい場合。 これは、標準 Managed File Transfer がインストールされているけれども、別のマシンで実行されているエージェント・キュー・マネージャーに接続するというケースです。

   デフォルトのインストール名変数は **BFG_INSTALLATION_NAME** です。

   fteCreateEnvironment の詳細については、fteCreateEnvironment (Redistributable Managed File Transfer Agent の環境セットアップ) を参照してください。

次のようにデータ・パスの場所を指定して環境変数 *BFG_DATA* を設定することもできます。

```
BFG_DATA=Datapath location
```

エージェントの作成、開始、停止、およびそのほかのコマンドを実行する前に、*BFG_DATA* 変数に正しいデータ・パスの場所が設定されていることを確認してください。

2. IBM MQ 接続をセットアップします。
   a) **fteSetupCoordination** コマンドを使用して調整キュー・マネージャーをセットアップします。

   fteSetupCoordination コマンドは、調整キュー・マネージャーに必要なセットアップ、および詳細な構成に必要なディレクトリーを作成します。 Redistributable Managed File Transfer Agent はクライアント・モードで機能します。バインディング・モードはサポートされないので、エラーを回避するためにこのコマンドには追加のパラメーターを指定する必要があります。

   ```
   fteSetupCoordination -coordinationQMgr PRMFTDEMO2
                -coordinationQMgrHost 9.121.59.233 -coordinationQMgrPort 3002
                -coordinationQMgrChannel SYSTEM.DEF.SVRCONN
   ```

   fteSetupCoordination コマンドの使用に関する詳細と手順については、fteSetupCoordination を参照してください。 調整キュー・マネージャーの構成方法については、765 ページの『MFT 用の調整キュー・マネージャーの構成』を参照してください。

   b) コマンド・キュー・マネージャーを作成し、セットアップします。

   ```
   fteSetupCommands -p PRMFTDEMO2 -connectionQMgrHost 9.121.59.233
                -connectionQMgrPort 3002 -connectionQMgrChannel SYSTEM.DEF.SVRCONN
                -connectionQMgr PRMFTDEMO2 -f
   ```

   fteSetupCommands コマンドの使用に関する詳細と手順については、fteSetupCommands: MFT command.properties ファイルの作成を参照してください。

3. エンドポイントに MFT エージェント定義を作成します。

   ```
   fteCreateAgent -p PRMFTDEMO2 -agentQMgrHost 9.121.59.233
              -agentQMgrPort 3002 -agentQMgrChannel SYSTEM.DEF.SVRCONN
              -agentName AGENT.TRI.BANK -agentQMgr PRMFTDEMO2 -f
   ```

   fteCreateAgent コマンドを使用してエージェントおよびエージェント・キュー・マネージャーを構成する方法について詳しくは、fteCreateAgent を参照してください。

   注 : コマンド出力の一部として表示される MQSC コマンドを使用して、エージェント・キュー・マネージャー上のエージェント・オブジェクトを定義する必要があります。そうしないと、ステップ 725 ページの『4』 の指示が機能しません。

   エージェントごとに、手順 725 ページの『2』 および 725 ページの『3』 で、エージェント・キュー・マネージャーのキューとトピックの定義を作成することになります。

4. エージェントを開始すれば、ファイルを転送するための準備は完了です。

   ```
   fteStartAgent -p PRMFTDEMO2 AGENT.TRI.BANK
   ```

   以下のコマンドを実行して、エージェントの状況を確認できます。

   ```
   fteListAgents
   ```

   fteListAgents コマンドの使用方法について詳しくは、fteListAgents を参照してください。

次のタスク

V 9.3.0 Redistributable Managed File Transfer Logger を構成する場合は、726 ページの『Redistributable Managed File Transfer Logger の初期構成の作成』のステップを実行します。

関連概念

719 ページの『Managed File Transfer の構成』
Managed File Transfer のフィーチャーは、インストール後に構成可能です。

719 ページの『MFT 構成オプション (Multiplatforms)』
Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティー・ファイルを提供します。 これらのプロパティー・ファイルは、製品のインストール時に定義される構成ディレクトリーにあります。

関連資料
fteCreateTransfer: 新規ファイル転送の開始

Windows Linux V 9.3.0 Redistributable Managed File Transfer Logger の初期構成の作成

クライアント・モードで調整キュー・マネージャーに接続するように、FILE タイプ Managed File Transfer Logger を構成できます。

始める前に
Redistributable Managed File Transfer Agent・パッケージのコンテンツをダウンロードして解凍します。 IBM MQ 9.3.0 以降、このパッケージには Redistributable Managed File Transfer Logger も含まれるようになりました。 詳しくは、721 ページの『Redistributable Managed File Transfer components のダウンロードおよび構成』を参照してください。

このタスクについて
Redistributable Managed File Transfer Agent と Redistributable Managed File Transfer Logger は同じ環境を共有します。 この環境が作成され、IBM MQ 接続がセットアップされたら、ロガーを作成して開始できます。

手順
1. ステップ 724 ページの『1』 の説明に従って Redistributable Managed File Transfer Agent および Redistributable Managed File Transfer Logger の共有環境が作成され、 723 ページの『Redistributable Managed File Transfer Agent の初期構成の作成』のステップ 725 ページの『2』 の説明に従って IBM MQ 接続がセットアップされていることを確認します。
   .
2. **fteCreateLogger** コマンドを使用してファイル・ロガーを作成します。
   以下に例を示します。

   ```
   fteCreateLogger FILELOGGER -loggerType FILE -loggerQMgr PRMFTDEMO2
   -loggerQMgrHost 9.121.59.233 -loggerQMgrPort 3003 -loggerQMgrChannel SYSTEM.DEF.SVRCONN
   -fileSize 20MB -fileCount 10 -fileLoggerMode CIRCULAR
   ```

   fteCreateLogger コマンドの使用法について詳しくは、fteCreateLogger を参照してください。
3. **fteStartLogger** コマンドを使用して、ロガーを開始します。
   fteStartLogger コマンドについて詳しくは、fteStartLogger を参照してください。

関連概念
719 ページの『Managed File Transfer の構成』
Managed File Transfer のフィーチャーは、インストール後に構成可能です。

719 ページの『MFT 構成オプション (Multiplatforms)』
Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティー・ファイルを提供します。 これらのプロパティー・ファイルは、製品のインストール時に定義される構成ディレクトリーにあります。

Windows Linux Redistributable Managed File Transfer components のアップグレード

新しい Redistributable Managed File Transfer package をダウンロードすることで、 Redistributable Managed File Transfer components をアップグレードできます。

始める前に

Redistributable Managed File Transfer Agent および Redistributable Managed File Transfer Logger の再配布可能ライセンス条項については、IBM MQ 再配布可能コンポーネントを参照してください。

注 : Advanced Message Security は、Redistributable Managed File Transfer package ではサポートされません。

このタスクについて

Redistributable Managed File Transfer components が既にインストールされている場合は、新しい再配布可能パッケージをダウンロードし、その内容を同じ場所に解凍することで、それらをアップグレードできます。

手順

1. ご使用のオペレーティング・システム用の IBM MQ 再配布可能 Managed File Transfer エージェント・パッケージ を Fix Central からダウンロードします。
2. すべての Managed File Transfer エージェント を停止し、ロガー は実行中の Managed File Transfer コマンドが完了するのを待ちます。
3. ダウンロードした新しい再配布可能パッケージの内容を、 Redistributable Managed File Transfer components が既にインストールされているディレクトリーと同じディレクトリーに解凍して、Redistributable Managed File Transfer components の既存のインストール済み環境のファイルを更新します。

z/OS MFT エージェントまたはロガーのコマンド・データ・セットの作成

特定の調整のために、特定の Managed File Transfer Agent または Managed File Transfer Logger の Managed File Transfer コマンド・テンプレート・データ・セットからコマンドの PDSE データ・セットを作成することができます。

このタスクについて

以下のステップを実行します。

手順

1. MFT コマンド・テンプレート PDSE ライブラリー・データ・セット SCSQFCMD のコピーを作成します。SCSQFCMD は、新しいライブラリー (例えば、*prefix.agent*.JCL) にコピーする必要があります。 更新されたバージョンの SCSQFCMD (BFGCOPY) メンバーを使用して、以下の置換を行うことができます。
   - *++supplied-library++* は、SCSQFCMD PDSE の完全修飾名に置き換えてください。
   - z/OS *++service-library++* を、新規 MFT コマンド PDSE ライブラリー・データ・セットの完全修飾名に置き換えます。 *++service-library++* は、作成されるエージェントまたはロガー・サービス用の出力データ・セットです。
2. 新規 MFT コマンド PDSE ライブラリー・データ・セットについて、メンバー BFGCUSTM を編集します。これは、エージェントまたはロガー用にコマンドをカスタマイズするための JCL スクリプトです。 各変数は *++variable name++* という形式で指定されます。 これを必要な値に置き換える必要があります。各種の JCL 変数の説明については、740 ページの『z/OS JCL 変数』を参照してください。 BFGSTDIN DD ステートメントでは、変数を Variables、Properties、および Environment という 3 つのカテゴリーで定義します。 ステートメントの形式は以下のとおりです。

```
[Variables]
variable1=value1
variable2=value2
....
variableN=valueN
[Properties]
property1=property value1
property2=property value2
```

```
...
propertyN=property valueN
[Environment]
custom_variable1=value1
custom_variable2=value2
....
custom_variableN=valueN
```

Variables は、各コマンドに必要なセットアップと環境変数のセットを定義します。

Properties は、MFT 構成プロパティーの指定変更を定義します。 ご使用の環境に合わせてエージェントまたはロガーをカスタマイズするために、必要に応じて、エージェントおよびロガーのプロパティーを追加できます。 すべてのプロパティーのリストについては、750 ページの『構成プロパティー・ファイル』を参照してください。 この機能は、MFT 構成プロパティー・ファイル (z/OS UNIX System Services・ファイルとして保守される) にアクセスしなくても済むようにするために用意されています。

Environment では、追加で必要なカスタム環境変数を任意に定義します。

3. 新規 MFT コマンド PDSE ライブラリー・データ・セットに対して、ジョブ BFGCUSTM を実行依頼します。 このジョブは、エージェントまたはロガーに適切な JCL コマンドのセットを、PDSE の新規メンバーとして生成します。 コマンドの完全なリストについては、743 ページの『z/OS エージェントおよびロガー・コマンドの JCL スクリプト』を参照してください。

   ジョブ BFGCUSTM は、DISP=OLD の DD ステートメントを含む JCL が入っているライブラリーを更新します。 実行依頼の後にエディターを終了して、ジョブが実行できるようにしなければなりません。

   出力ジョブ・ログを調べて、JCL スクリプトが正常に実行されたことを確認してください。 何らかの障害がある場合には、それらを修正して BFGCUSTM ジョブを再び実行依頼してください。

   さらに BFGCUSTM JCL スクリプトは、z/OS UNIX System Services MFT 構成プロパティー・ファイルがそれぞれ整合するように、必要に応じてそれらのファイルを更新します。 CoordinationQMgr プロパティーによって定義される構成が存在しない場合には、警告メッセージが出力されます。 その場合、生成された BFGCFCR および BFGCMCR のジョブを実行して、構成プロパティー・ファイルを作成する必要があります。 エージェントの編集の場合は BFGAGCR、ロガーの編集の場合は BFGLGCRS を実行する必要があります。 指定された構成が既に存在する場合には、BFTCUSTM JCL スクリプトで定義されているすべてのプロパティーにより、構成が更新されます。

関連概念

721 ページの『z/OS での MFT 構成オプション』
z/OS の Managed File Transfer 構成オプションは、分散プラットフォームの場合のオプションと同じです。

関連タスク

739 ページの『MFT の既存のエージェントまたはロガーのコマンド・データ・セットの更新 (z/OS)』
Managed File Transfer コマンド・テンプレート・データ・セットから作成される Managed File Transfer コマンド PDSE ライブラリー・データ・セットを更新できます。

z/OS Managed File Transfer for z/OS の構成

Managed File Transfer for z/OS では、コンポーネントを正常に動作させるためのカスタマイズが必要です。

このタスクについて

以下の作業が必要です。

1. PDSE メンバーを編集して構成データを指定します
2. 調整キュー・マネージャーを定義します
3. コマンド・キュー・マネージャーを定義します
4. 1 つ以上のエージェントを構成します
5. オプション: データを Db2 に保管するようロガー・タスクを構成します

実行する必要があるタスクの順序については、以下のトピックで詳しく説明されています。

関連概念

729 ページの『MFT 構成の確認』

開始する前に、システムの構成を確認する必要があります。

関連タスク

IBM MQ Advanced for z/OS のインストール

z/OS MFT 構成の確認

開始する前に、システムの構成を確認する必要があります。

Managed File Transfer (MFT) を使用するには、定義した MFT 構成ごとに以下の役割で動作するキュー・マネージャーが 1 つ以上必要です。

- 調整キュー・マネージャー。このキュー・マネージャーは、コーディネーター上のトピックにパブリッシュされた、構成内の各エージェントの状況に関する情報を維持します。
- 1 つ以上のコマンド・キュー・マネージャーまたは接続キュー・マネージャー。このキュー・マネージャーは、MFT コマンドを使用するための IBM MQ ネットワークへの入り口点として機能します。
- 1 つ以上のエージェント・キュー・マネージャー。このキュー・マネージャーは、MFT エージェントと IBM MQ ネットワークの間の通信を提供します。

上記の役割は、それぞれに別々のキュー・マネージャーで実行することも、すべての役割をまとめて 1 つのキュー・マネージャーで実行する (最も単純な構成) こともできます。

既存の MFT 環境に z/OS キュー・マネージャーを追加する場合は、z/OS キュー・マネージャーと構成内の他のキュー・マネージャーの間の接続を定義する必要があります。 これは、手動で定義された伝送キューを使用するか、クラスター化を使用することによって行えます。

各 MFT エージェントは、単一のキュー・マネージャーと通信します。 複数のエージェントが同じキュー・マネージャーと通信する場合は、エージェント・キュー・マネージャーに次の複数のキューをエージェントごとに定義します。

- SYSTEM.FTE.COMMAND.*agent_name*
- SYSTEM.FTE.DATA.*agent_name*
- SYSTEM.FTE.REPLY.*agent_name*
- SYSTEM.FTE.STATE.*agent_name*
- SYSTEM.FTE.EVENT.*agent_name*
- SYSTEM.FTE.AUTHAGT1.*agent_name*
- SYSTEM.FTE.AUTHTRN1.*agent_name*
- SYSTEM.FTE.AUTHOPS1.*agent_name*
- SYSTEM.FTE.AUTHSCH1.*agent_name*
- SYSTEM.FTE.AUTHMON1.*agent_name*
- SYSTEM.FTE.AUTHADM1.*agent_name*

`SYSTEM.FTE.COMMAND.*` などのプロファイルを使用して汎用セキュリティー・プロファイルを定義することも、エージェントごとに特定のプロファイルを定義することもできます。

関連概念

729 ページの『z/OS 用の MFT の構成を開始する前に』

Managed File Transfer (MFT) 構成は、z/OS UNIX System Services (z/OS UNIX) のファイルと PDSE データ・セットを使用します。

関連資料

MFT システム・キューおよびシステム・トピック

z/OS z/OS 用の MFT の構成を開始する前に

Managed File Transfer (MFT) 構成は、z/OS UNIX System Services (z/OS UNIX) のファイルと PDSE データ・セットを使用します。

構成と操作のほとんどは PDSE の JCL を使って行われるため、z/OS UNIX 環境での作業に精通しておく必要があります。

ISPF から OMVS にアクセスすることも、ワークステーションでコマンドを使って Telnet タイプのセッションを使用することもできます (例えば Telnet Putty や SSH)。

ISPF から OMVS を使用する場合、標準 ISPF エディターや、ブラウズ・コマンド **oedit** および **obrowse** を使用できます。

以下の z/OS UNIX コマンドに精通しておく必要があります。

表 *40.* 共通の *z/OS UNIX* コマンド

コマンド	**Function**
chmod xxx path	ファイル・アクセス許可を変更します。
df -k path	ファイル・システムに残っているフリー・スペース量を報告します。 -k はフリー・スペースを KB 単位で報告します。
du -kt path	パスの下のディレクトリーのサイズを報告します。 サイズは KB 単位で報告されます。
find path -name xxx	パスのディレクトリーで、xxxx という名前のファイルを検索します。 xxx には大/小文字の区別があり、*zzz のように指定できます。
ls -ltrd directory	指定されたディレクトリー内のファイルではなく、ディレクトリーに関する情報をリストします。
ls -ltr path	パス内のファイルに関する情報をリストします。
obrowse filename	ファイル名を参照します。
oedit filename	ファイルを OMVS で編集します。

以下の表の項目を確認し、お客様の会社に適した項目を表に記入してください。 これらの値は、メンバー BFGCUSTM を編集する際に必要になります。

表 *41.* メンバー *BFGCUSTM* に必要なパラメーター

名前	**データの例**	**コメント**
ADMIN_JOB1		ジョブ・カード。 ジョブはすべて同じ JCL カードで生成されます。
armELEMENT	ARM が使用されている場合、このエージェントまたはロガーの ARM ポリシーで指定されている ARM ELEMENT 値を使用します。 ARM が使用されていない場合、このパラメーターをブランクに設定します (例、armELEMENT=)。	

表 41. メンバー *BFGCUSTM* に必要なパラメーター (続き)

名前	データの例	コメント
armELEMTYPE	ARM が使用されている場合、ARM ポリシーで指定されている ARM ELEMTYPE を使用します。 例えば、エージェントの場合は armELEMTYPE=SYSBFGAG、ロガーの場合は armELEMTYPE=SYSBFGLG のようになります。 ARM が使用されていない場合、このパラメーターをブランクに設定します (例、armELEMTYPE=)。	
BFG_DATA		必要に応じて記入してください
BFG_GROUP_NAME	MQM	
BFG_JAVA_HOME	/java/java71_bit64_GA/J7.1_64/	
BFG_JVM_PROPERTIES		必要に応じて記入してください
BFG_PROD	/mqm/V9R2M0/mqft	IBM MQ for z/OS UNIX System Services Components ディレクトリーの下の mqft ディレクトリーへの絶対パス。
BFG_WTO	YES	syslog の MFT メッセージを取得します。
CLEAN_AGENT_PROPS	-trs	このパラメーターは、BFGAGCL メンバーが実行されているときにエージェントのクリーンアップに使用するオプションを指定します。 このパラメーターの有効な値について詳しくは、fteCleanAgent: MFT エージェントのクリーンアップを参照してください。
coordinationQMgr	MQPV	必須構成
CREDENTIAL_PATH		マイグレーションで使用します
Db2HLQ	SYS2.Db2V10 (V)	
DB_PROPS_PATH		マイグレーションで使用します
FTE_CONFIG		マイグレーションで使用します
JOBCARD1		これは、長時間実行されているタスク、エージェント、およびロガー用のジョブ・カードです。
LIBRARY	SCEN.FTE.JCL	MFT PDSE の名前。 各エージェントまたはロガー・タスクのコピーが必要です。
MQ_HLQ	IBM MQ データ・セットの高位修飾子。 MQM.V920 など	
MQ_LANG	E	

表 41. メンバー *BFGCUSTM* に必要なパラメーター (続き)		
名前	**データの例**	**コメント**
MQ_PATH	/mqm/V9R2M0	IBM MQ for z/OS UNIX System Services Components のインストール済み環境への絶対ディレクトリー・パス。
名前	AGENT1	
OUTPUT_CLASS	*	
PATH	bin:/usr/bin:/usr/sbin	
productId	ADVANCEDVUE	このパラメーターは、Managed File Transfer 使用率の記録の対象となる製品タイプを設定するために使用されます。 このパラメーターの有効な値について詳しくは、fteSetProductId: z/OS SCRT レコーディング ID の設定を参照してください。
QMGR	MQPV	
SERVICE_TYPE	AGENT または LOGGER	
TMPDIR	/tmp	一時ファイルの読み取り/書き込みアクセスが可能な z/OS UNIX パス。

さらに、以下の変数を確認し、必要に応じて値を指定する必要があります。

- coordinationQMgrHost=
- coordinationQMgrPort=
- coordinationQMgrChannel=
- connectionQMgr=
- connectionQMgrHost=
- connectionQMgrPort=
- connectionQMgrChannel=

これらのプロパティーは、AGENT または LOGGER に共通しています。

注 : クライアント接続にはホスト、ポート、およびチャネルが必要ですが、ローカル・マシン上のバインディング接続ではブランクのままにしておく必要があります。

関連概念

732 ページの『確認する項目』
十分なディスク・スペースがあり、データを保管するためのディレクトリーがあり、必要ファイルが存在することを確認します。

735 ページの『メンバー BFGCUSTM の編集』
ジョブを実行する前に、メンバー BFGCUSTM を編集し、お客様の会社で使用するパラメーターの値を入力する必要があります。

z/OS 確認する項目

十分なディスク・スペースがあり、データを保管するためのディレクトリーがあり、必要ファイルが存在することを確認します。

十分なディスク・スペースの確認

構成固有のファイルを保管しようとしているファイル・システムに十分な使用可能ディスク・スペースが存在することを確認します。

エージェント・トレースが有効な場合、デフォルトで 100 MB のディスク・スペースを使用する場合があります。

構成ファイル自体は小さく、サイズは数 KB ほどです。

2 つのエージェントと 1 つのロガーを使用する計画の場合、少なくとも 300 MB が必要です。 コマンド **df -k** path を使用できます。 path は、インストール済み環境固有のファイルの場所です。 これにより、使用可能スペースと合計スペースが KB で表示されます。

300 MB は 307,200 KB なので、少なくとも 310,000 KB を使用可能にする必要があります。

Managed File Transfer データを保管するためのディレクトリーの作成と確認

Managed File Transfer (MFT) データを保管するためのディレクトリーが必要です。

ファイル・システム `df -k /var` に十分なスペースがあることを確認してください。 このファイル・システムには少なくとも 310,000 KB の使用可能スペースが必要です。

このファイル・システムが作成されていない場合、**mkdir** コマンド (例えば **mkdir /var/mft**) を使用します。

コマンド **ls -ltrd /var/mft** を使用して、ユーザーがこのディレクトリーに対して持つ許可を表示します。

所有者またはグループが正しくない場合は、コマンド **chown owner:group /var/mft** を使用します。

グループの許可が正しくない場合、以下のコマンドを使用して、所有者とグループに読み取り、書き込み、および実行の権限を付与します。 コマンド **chmod 775 /var/mft** は、すべてのユーザーに読み取りと実行の権限を付与します。

ファイルが存在することとそのファイルに対するアクセス権限があることの確認

カスタマイズの際に使用するファイルに対して **ls -ltr** コマンドを使用します。 以下に例を示します。

```
ls -ltrd /java/java71_bit64_GA/J7.1_64/bin
```

この結果は次のようになります。

```
drwxr-xr-x  4 SYSTASK TSOUSER   8192 Nov 15 2013 /java/java71_bit64_GA/J7.1_64/bin
```

`drwxr-xr-x` は、以下のものを意味します。

d
これはディレクトリーです。

rwx
所有者 *SYSTASK* に、ディレクトリーに対する読み取り、書き込み、および実行のアクセス権限があります。

r-x
グループ *TSOUSER* のユーザーは、ディレクトリー内のファイルを読み取りおよび実行できます。

r-x
汎用アクセスです。つまり、どのユーザーもディレクトリー内のファイルを読み取りまたは実行できます。

以下で指定されるファイルを確認してください。

表 42. ユーザーに必要な特定のファイルに対するアクセス権限	
Path	**構成を行うユーザーに必要なアクセス権限**
BFG_JAVA_HOME	読み取りと実行
/tmp	読み取りと書き込み
BFG_PROD	読み取り
BFG_DATA	書き込み
MQ_PATH	読み取り

関連概念

729 ページの『z/OS 用の MFT の構成を開始する前に』
Managed File Transfer (MFT) 構成は、z/OS UNIX System Services (z/OS UNIX) のファイルと PDSE データ・セットを使用します。

734 ページの『z/OS 構成の共通 MFT』
異なる Managed File Transfer 構成の概要

z/OS z/OS 構成の共通 MFT

異なる Managed File Transfer 構成の概要

Managed File Transfer は、キュー・マネージャーに接続したエージェントを使用してデータを転送します。

MFT は、以下のような複数のキュー・マネージャーを使用できます。

- データを転送する 1 つ以上のキュー・マネージャー。
- 要求を実行するコマンド・キュー・マネージャー。 例えば、転送を開始する要求がこのキュー・マネージャーに送信されると、関連したコマンドが MFT エージェントにルーティングされます。
- 作業を管理する調整キュー・マネージャー。

Managed File Transfer (MFT) には、以下の 3 つの共通の構成があります。

1. ローカル接続を使用して 1 つのキュー・マネージャーで 1 つ以上のエージェントを使用する構成。 これは、データ・セットの内容を IBM MQ キューに入れるために使用できます。
2. クライアント・バインディングを使用して 1 つのキュー・マネージャーで分散マシン上の MFT クライアントを使用する構成。
3. チャネルで 2 つのキュー・マネージャーを接続して、各マシンで 1 つ以上のエージェントを使用する構成。 それらのエージェントでは、クライアント・バインディングもローカル・バインディングも可能です。

以下の点に注意してください。

1. MFT は、Java で記述されており、MFT を構成したり操作したりするためのシェル・スクリプトと JCL が組み込まれています。
2. Db2 の状況とアクティビティーをログに記録して、Db2 表に保管できます。
3. MFT を構成する担当者は、z/OS UNIX System Services (z/OS UNIX)について熟知している必要があります。例えば、
   - `/u/userID/myfile.txt2` のような名前のファイルを含むディレクトリー構造
   - 以下のような z/OS UNIX コマンド

     cd (ディレクトリーの変更)
     ls (リスト)
     chmod (ファイル・アクセス許可の変更)
     chown (ファイルの所有権の変更や、ファイルやディレクトリーにアクセスできるグループの変更)
4. z/OS UNIX で MFT を構成および実行するには、以下の製品が必要になります。

- Java(例: /java/java71_bit64_GA/J7.1_64/)
- IBM MQ V920 (例えば、/mqm/V9R2M0)。
- Db2 JDBC ライブラリー (状況およびヒストリーに Db2 を使用したい場合)。例えば、/db2/db2v12/jdbc/lib

調整キュー・マネージャーが必要です。 ただし、同じキュー・マネージャーを使って、エージェントの実行、コマンドの処理、および調整を実行できます。 複数のキュー・マネージャーを使用している場合、コーディネーターとして機能するものを 1 つ選択する必要があります。

IBM MQ 接続の確認

既存の MFT コーディネーター・キュー・マネージャーがある場合、構成を行うキュー・マネージャーと調整キュー・マネージャーおよびコマンド・キュー・マネージャーの間の接続が必要です。

z/OS JCL ライブラリーを作成するための SCSQFCMD のコピー

エージェントおよびロガーごとに JCL ライブラリーを作成する必要があります。 JCL には、エージェントまたはロガーの作成および実行に使用する構成およびジョブが含まれています。

エージェントおよびロガーごとに、BFGCOPY メンバーを編集して実行することにより、 IBM 提供の SCSQFCMD ライブラリーのコピーを作成します。

このライブラリーは、エージェントまたはロガーの構成を定義する場合に使用され、カスタマイズ後には、必要な Managed File Transfer 構成の作成、およびエージェントまたはロガーの作成に使用できるジョブが置かれます。

メンバー BFGCUSTM をこのプロセスの一部として作成します。

注 : z/OS UNIX コマンドに精通している場合、他のプラットフォームで使用するのと同じコマンドを使用して z/OS を構成できます。

関連概念

734 ページの『z/OS 構成の共通 MFT』
異なる Managed File Transfer 構成の概要

735 ページの『メンバー BFGCUSTM の編集』
ジョブを実行する前に、メンバー BFGCUSTM を編集し、お客様の会社で使用するパラメーターの値を入力する必要があります。

z/OS メンバー BFGCUSTM の編集

ジョブを実行する前に、メンバー BFGCUSTM を編集し、お客様の会社で使用するパラメーターの値を入力する必要があります。

特定の値を必要とするパラメーターのリストについては、メンバー BFGCUSTM に必要なパラメーターを参照してください。

さらに、以下の変数を確認し、必要に応じて値を指定する必要があります。

- coordinationQMgrHost=
- coordinationQMgrPort=
- coordinationQMgrChannel=
- connectionQMgr=
- connectionQMgrHost=
- connectionQMgrPort=
- connectionQMgrChannel=

これらのプロパティーは、AGENT または LOGGER に共通しています。

注 : クライアント接続にはホスト、ポート、およびチャネルが必要ですが、ローカル・マシン上のバインディング接続ではブランクのままにしておく必要があります。

これが Managed File Transfer 環境で最初のキュー・マネージャーであり、かつ、この同じキュー・マネージャーを調整、コマンドおよびエージェント実行に使用する場合、この値をローカル・キュー・マネージャーの名前に設定します。

```
coordinationQMgr=MQPV
connectionQMgr=MQPV
```

MQPV は、ローカル・キュー・マネージャーの名前です。

ジョブを実行依頼します。これにより、PDSE が更新され、指定されたパスの下にディレクトリー構造が作成されます。

このジョブは排他的に使用する必要があるため、ジョブの実行中は PSDE の使用を停止する必要があります。

ヒント: ジョブ BFGCUSTM を実行依頼するときは常に、そのジョブによってすべての JCL ファイルが置換されます。 変更するメンバーごとに名前変更する必要があります。

関連概念

729 ページの『z/OS 用の MFT の構成を開始する前に』
Managed File Transfer (MFT) 構成は、z/OS UNIX System Services (z/OS UNIX) のファイルと PDSE データ・セットを使用します。

738 ページの『エージェントの作成』
エージェント固有の PDSE (例えば *user.MFT.AGENT1*) を作成するには、PDSE をコピーする必要があります。 以前のエージェント構成またはロガー構成があれば、その構成から PDSE をコピーします。 初めて構成を作成する場合は、MFT に用意されている PDSE をコピーします。

z/OS 調整キュー・マネージャーの定義

Managed File Transfer では、調整キュー・マネージャーとして機能するキュー・マネージャーを作成する必要があります。

選択した構成に応じて、このキュー・マネージャーは、ローカル MVS システムか、別のマシンに配置されます。 前者の場合、その接続はバインディング接続になり、後者の場合は、クライアント接続になります。

構成ステップを正常に実行した後、PDSE に構成済みメンバーが存在するようになります。

メンバー BFGCFCR は調整キュー・マネージャーを定義します。このジョブは以下のことを行います。

1. Managed File Transfer (MFT) ディレクトリーにディレクトリー構造を作成し、構成ファイルを作成します。
2. CSQUTIL を実行して IBM MQ リソースを定義します。

調整キュー・マネージャーがリモート・マシン上にある場合、このジョブ・ステップは失敗します。

メンバー BCFCFCR は z/OS UNIX System Services にファイルを作成し、MQ 定義を作成します。 このジョブは、以下のことを行います。

1. MFT トピックを作成します
2. MFT キューを作成します
3. *NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)* が *NAMES(SYSTEM.BROKER.DEFAULT.STREAM, SYSTEM.BROKER.ADMIN.STREAM, SYSTEM.FTE)* となるように変更します。
4. *ALTER QMGR PSMODE(ENABLED)* を実行します。

   変更を行う前に *DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)* コマンドが発行されます。 NAMLIST がデフォルトではない場合、名前リストを変更して SYSTEM.FTE を名前リストに追加する必要があります。

メンバー BCFCFCR を独自の接頭部を使って名前変更します (例えば CCPCFCR)。このファイルを再びカスタマイズするとそれが置換されるためです。

資格情報ファイルの名前を挿入して、名前変更されたこのメンバーを編集します。 以下に例を示します。

```
%BFGCMD CMD=fteSetupCoordination +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
```

ジョブを保存し実行依頼します。 ジョブを再実行依頼する必要がある場合、*-f* オプションを追加する必要があります。

このジョブを実行すると、それによって作成される IBM MQ リソースがリストされます。 これらのリソースを保護する必要があります。

```
DEFINE TOPIC('SYSTEM.FTE') TOPICSTR('SYSTEM.FTE') REPLACE
ALTER TOPIC('SYSTEM.FTE') NPMSGDLV(ALLAVAIL) PMSGDLV(ALLAVAIL)
DEFINE QLOCAL(SYSTEM.FTE) LIKE(SYSTEM.BROKER.DEFAULT.STREAM) REPLACE
ALTER QLOCAL(SYSTEM.FTE) DESCR('Stream for MFT Pub/Sub interface')
* Altering namelist: SYSTEM.QPUBSUB.QUEUE.NAMELIST
* Value prior to alteration:
DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)
ALTER NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST) +
NAMES(SYSTEM.BROKER.DEFAULT.STREAM+
,SYSTEM.BROKER.ADMIN.STREAM,SYSTEM.FTE)
* Altering PSMODE. Value prior to alteration:
DISPLAY QMGR PSMODE
ALTER QMGR PSMODE(ENABLED)
```

関連タスク

737 ページの『コマンド・キュー・マネージャーの定義』
同じキュー・マネージャーを調整キュー・マネージャーとコマンド・キュー・マネージャーとして使用することも、新しくコマンド・キュー・マネージャーを作成することもできます。

z/OS コマンド・キュー・マネージャーの定義

同じキュー・マネージャーを調整キュー・マネージャーとコマンド・キュー・マネージャーとして使用することも、新しくコマンド・キュー・マネージャーを作成することもできます。

このタスクについて

コマンド・キュー・マネージャーは必須ですが、同じキュー・マネージャーを調整キュー・マネージャーとコマンド・キュー・マネージャーに使用することができます。 そうしない場合は、新しいコマンド・キュー・マネージャーを作成する必要があります。 これは調整キュー・マネージャーと同じマシンに配置することができますが、そうする必要があるわけではありません。

手順

1. メンバー BFGCMCR を独自の接頭部 (例えば CCPCMCR) を使って名前変更します。
   BFGCMCR は必ず名前変更してください。このファイルを再カスタマイズすると、置き換えられるためです。
2. 資格情報ファイルの名前を挿入して、名前変更されたメンバーを編集します。
   以下に例を示します。

   ```
   %BFGCMD CMD=fteSetupCommands +
   -credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
   ```

3. ジョブを保存し実行依頼します。
   ジョブを再実行依頼する必要がある場合、*-f* オプションを追加する必要があります。

   このキュー・マネージャーは、**ftePingAgent** などのコマンドで使用されます。
4. このメンバーを確認し、実行依頼し、その出力を確認します。

次のタスク

エージェントの作成方法について詳しくは、738 ページの『エージェントの作成』を参照してください。

関連概念

736 ページの『調整キュー・マネージャーの定義』

Managed File Transfer では、調整キュー・マネージャーとして機能するキュー・マネージャーを作成する必要があります。

関連タスク

MQMFTCredentials.xml の構成

関連資料

MFT の資格情報ファイルのフォーマット

z/OS エージェントの作成

エージェント固有の PDSE (例えば *user.MFT.AGENT1*) を作成するには、PDSE をコピーする必要があります。 以前のエージェント構成またはロガー構成があれば、その構成から PDSE をコピーします。 初めて構成を作成する場合は、MFT に用意されている PDSE をコピーします。

メンバー BFGCUSTM を確認し、異なる資格情報ファイルを使用する必要がある場合は、作成します。

内容のほとんどは、735 ページの『メンバー BFGCUSTM の編集』で詳述されているカスタマイズと同じです。

以下の変更を行う必要があります。

- //SYSEXEC DD DSN=SCEN.FTE.JCL.AGENT1
- LIBRARY がエージェント PDSE と一致するようにする
- SERVICE_TYPE=AGENT
- NAME が (PDSE と一致する) エージェント JOBCARD の名前になるようにする
- BFG_JVM_PROPERTIES="-Xmx1024M" を変更する

このジョブを実行依頼します。その際、このジョブはデータ・セットへの排他的アクセスが必要であることを覚えておいてください。

エージェントのジョブの名前の形式はすべて *BFGAG** になります。

メンバー *BFGAGCR* の名前を変更します。 このジョブは Managed File Transfer ディレクトリーのファイルを更新し、CSQUTIL を使用してローカル・キュー・マネージャーのエージェント固有キューを作成します。 資格情報ファイルの名前を指定します(例:-credentialsFile //'SCEN.FTE.JCL.VB(CREDOLD))。 名前を指定しないと、エージェントを開始するためのジョブが資格情報ファイルを使用しなくなります。

出力を確認し、プロセスが正常に実行されたことを確認します。

ヒント: *agent.properties* ファイルのパス名を、ジョブの出力からエージェントの PDSE のメンバーにコピーします。

例えば、/u/userid/fte/wmqmft/mqft/config/MQPA/agents/AGENT1/agent.properties をメンバー AGENT にコピーします。

プロパティー・ファイルを表示し、行/u/userid/fte/wmqmft/mqft/logs/MQPA/agents/AGENT1/logs を追加する必要がある場合に役立ちます。

これはトレース・ファイルの保管場所です。

関連概念

736 ページの『調整キュー・マネージャーの定義』
Managed File Transfer では、調整キュー・マネージャーとして機能するキュー・マネージャーを作成する必要があります。

739 ページの『エージェントの使用』
エージェントが正しく動作していることを確認するためのさまざまなコマンドの使用方法。

関連タスク

737 ページの『コマンド・キュー・マネージャーの定義』
同じキュー・マネージャーを調整キュー・マネージャーとコマンド・キュー・マネージャーとして使用することも、新しくコマンド・キュー・マネージャーを作成することもできます。

z/OS エージェントの使用

エージェントが正しく動作していることを確認するためのさまざまなコマンドの使用方法。

エージェントの 開始

メンバー BFGAGST の名前を変更し、メンバーを確認し、ジョブを実行依頼します。

これが動作している場合、「BFGAG0059I: エージェントは正常に開始されました」というメッセージが表示されます。

アクティブ・エージェントの表示

メンバー BFGAGLI の名前を変更し、メンバーを確認し、調整キュー・マネージャーを使用するジョブを実行依頼します。

接続の問題があれば、それを解決する必要があります。

動作確認のためのエージェントに対する ping

メンバー BFGAGPI の名前を変更し、メンバーを確認し、コマンド・キュー・マネージャーを使用するジョブを実行依頼します。

接続の問題があれば、それを解決する必要があります。

テスト転送の実行

詳しくは、745 ページの『検証転送の実行』を参照してください。

エージェントの停止

メンバー BFGAGSP の名前を変更し、メンバーを確認し、ジョブを実行依頼します。

メンバー BFGAGST を使用してエージェントを再始動します。

関連概念

738 ページの『エージェントの作成』
エージェント固有の PDSE (例えば *user.MFT.AGENT1*) を作成するには、PDSE をコピーする必要があります。 以前のエージェント構成またはロガー構成があれば、その構成から PDSE をコピーします。 初めて構成を作成する場合は、MFT に用意されている PDSE をコピーします。

z/OS MFT の既存のエージェントまたはロガーのコマンド・データ・セットの更新 (z/OS)

Managed File Transfer コマンド・テンプレート・データ・セットから作成される Managed File Transfer コマンド PDSE ライブラリー・データ・セットを更新できます。

手順

1. BFGCUSTM JCL スクリプト・メンバーを編集して、BFGSTDIN DD ステートメント内の変数とプロパティーを更新します。

   以前に定義したプロパティーを除去する場合には、エントリーを除去するのではなく、その値をブランクに設定します。 BFGCUSTM JCL スクリプトの実行時に、指定したプロパティーは、エージェントおよびロガーの実際の z/OS UNIX System Services のプロパティー・ファイルへの更新として適用されます。 あるプロパティーをブランク値に設定することは、そのプロパティーを削除するという指示になります。
2. ジョブ BFGCUSTM を実行依頼します。 このジョブは、エージェントまたはロガーに適切な JCL コマンドのセットを再び生成します。 コマンドの完全なリストについては、743 ページの『z/OS エージェントおよびロガー・コマンドの JCL スクリプト』を参照してください。 出力ジョブ・ログを調べて、JCL

スクリプトが正常に実行されたことを確認してください。 何らかの障害がある場合には、それらを修正して BFGCUSTM ジョブを再び実行依頼してください。

タスクの結果

生成された JCL スクリプトに変更を加えて、独自のロジックを追加することができます。 ただし、BFGCUSTM を再び実行するときには、カスタム・ロジックを上書きすることになる場合があるので、注意してください。

関連概念

721 ページの『z/OS での MFT 構成オプション』
z/OS の Managed File Transfer 構成オプションは、分散プラットフォームの場合のオプションと同じです。

関連タスク

727 ページの『MFT エージェントまたはロガーのコマンド・データ・セットの作成』
特定の調整のために、特定の Managed File Transfer Agent または Managed File Transfer Logger の Managed File Transfer コマンド・テンプレート・データ・セットからコマンドの PDSE データ・セットを作成することができます。

z/OS *z/OS JCL 変数*

置換値、JCL 変数、および構成プロパティーを BFGCUSTM スクリプト内で使用できます。

以下の表に、MFT コマンド PDSE ライブラリー・データ・セット内の BFGCUSTM JCL スクリプトの置換値をリストします。 BFGCUSTM ジョブを実行依頼する前に、これらの置換値を適切な値に置き換える必要があります。

表 *43. BFGCUSTM JCL* スクリプトの置換値

置換変数	**値**
++library++	MFT コマンド PDSE ライブラリーを含むデータ・セットの名前。
++bfg_java_home++	Java インストールの場所。
++mq_path++	IBM MQ for z/OS UNIX System Services Components ディレクトリーへのパス。 例えば、`/mqm/V9R2M0` などです。 これは、MFT のインストールへの絶対パス(例えば、 `/mqm/V9R2M0/mqft`)を指定するために使用されます。

以下の表では、MFT コマンド PDSE ライブラリー・データ・セット内の BFGCUSTM JCL スクリプトの BFGSTDIN DD ステートメントのための環境変数 ([Variables] セクション内) を説明します。 BFGCUSTM ジョブを実行依頼する前に、置換値 (2 つの正符号 ++ で囲まれた値) で指定されたすべての変数を適切な値に置き換える必要があります。

表 *44.* 環境変数

環境変数	**値**
LIBRARY	MFT コマンド PDSE ライブラリーを含むデータ・セットの名前。
TMPDIR	一時ファイル用の z/OS UNIX System Services・ディレクトリー。
BFG_PROD	IBM MQ for z/OS UNIX System Services Components ディレクトリーの下 `mqft` にあるディレクトリーの絶対パス;例:`/mqm/V9R2M0/mqft`。
BFG_DATA	Managed File Transfer for z/OS のデータ・ディレクトリーの場所。これは *DATA_DIR* のパスです。

表 44. 環境変数 (続き)

環境変数	値
BFG_JAVA_HOME	Java インストールの場所。
BFG_JVM_PROPERTIES	オプション。 BFG_JVM_PROPERTIES 環境変数の値を設定します。 これらのプロパティーは、Java 仮想マシンに引き渡されます。
BFG_GROUP_NAME	通常、mqm ファイル・グループは、MFT 構成のデータ・ファイルおよびコマンドに関連付けられます。したがって、mqm グループのメンバーであるすべてのユーザーが MFT 構成へのアクセスと変更を行うことができます。 詳しくは、ファイルのシステム許可 IBM MQ の MFT を参照してください。 z/OS システムの場合、ファイル・グループは z/OS UNIX System Services (z/OS UNIX) ファイル・システム・エンティティーであり、mqm ファイル・グループは必ずしも定義されません。 BFG_GROUP_NAME 環境変数を使用すると、 MFT 構成データ・ファイルの z/OS UNIX ファイル・システム・グループを関連付けることができます。 例えば、z/OS UNIX シェルのプロンプトに次のように入力します。 `export BFG_GROUP_NAME=FTEGB` これは、現行 z/OS UNIX セッション用に今後作成されるすべての構成ファイルに関連付けられるグループ *FTEGB* を定義します。 BFG_GROUP_NAME は、ブランク値に設定することも、削除することもできます。 **注 :** 初めて BFGCUSTM を実行するときに、 MFT 構成を複数のユーザー ID で使用する場合は、必要なすべてのユーザー ID がアクセスできるグループに BFG_GROUP_NAME を設定することが重要です。 BFGCUSTM を再度実行する場合には、 BFG_GROUP_NAME を変更してはなりません (変更する場合は、BFG_DATA によって参照されるディレクトリー内のすべてのファイルおよびディレクトリーの z/OS UNIX グループ・ファイル・アクセス許可も、新しい BFG_GROUP_NAME 設定を反映するように変更する必要があります)。
BFG_WTO	BFG_WTO に YES、ON、または TRUE を設定すると、z/OS ロギングが使用可能になります。 これにより、エージェントのイベント・ログに書き込まれるメッセージが z/OS オペレーター・ログ機能にも書き込まれるかどうかを制御できるため、JCL からエージェントを実行した場合に自動化製品へのアクセスがより簡単になります。 宛先コードは「プログラマー情報」(11) で、記述子コードは「通知」(12) です。
SERVICE_TYPE	MFT コマンド・ライブラリーがエージェント用か、ロガー用かを指定します。 有効な値は AGENT または LOGGER です。

表 44. 環境変数 (続き)

環境変数	値
名前	SERVICE_TYPE 値に対するエージェントまたはロガーの名前。
QMGR	SERVICE_TYPE 値に対するエージェントまたはロガーに関連付けられたローカル・キュー・マネージャーの名前。
OUTPUT_CLASS	SYSOUT データ・セットの出力クラス。 デフォルトは * であり、この場合はジョブ・ステートメントからの MSGCLASS パラメーターと同じ出力クラスを要求します。
MQ_PATH	IBM MQ for z/OS UNIX コンポーネント・ディレクトリーへのパス。
MQ_HLQ	IBM MQ データ・セットの高位修飾子。
MQ_LANG	必要な言語。
DB2_HLQ	オプション。 Db2 データ・セット用の高位修飾子。
JOBCARD1	JCL コマンド・ジョブのヘッダー行 1。
JOBCARD2	JCL コマンド・ジョブのヘッダー行 2。
JOBCARD3	JCL コマンド・ジョブのヘッダー行 3。
ADMIN_JOB1	admin ジョブのヘッダー行 1。
ADMIN_JOB2	admin ジョブのヘッダー行 2。
ADMIN_JOB3	admin ジョブのヘッダー行 3。
FTE_CONFIG	マイグレーションの対象となる既存の MFT 構成。 マイグレーションが必要ない場合には、ブランク値に設定します。
CREDENTIAL_PATH	マイグレーションのための資格情報ファイルへのパス(/u/user1/agent3 など)。 マイグレーション・コマンド **BFGAGMG** および **BFGLGMG** JCL スクリプトでのみ必要です。 マイグレーションが必要ない場合には、ブランク値に設定します。 以下にも注意してください。
DB_PROPS_PATH	マイグレーションの対象となるデータベース・ロガー・プロパティー・ファイルを指定します。 このオプションが必要なのは、プロパティー・ファイルでデフォルトの名前とパス: `config_directory/coordination_qmgr/databaselogger.properties` が使用されていない場合のみです。 マイグレーションが必要ない場合には、ブランク値に設定します。

以下の表では、MFT コマンド PDSE ライブラリー・データ・セット内の BFGCUSTM JCL スクリプト用の BFGSTDIN DD ステートメントのための必須 MFT 構成プロパティーを記述します。 BFGCUSTM ジョブを実行依頼する前に、置換値 (2 つの正符号 ++ で囲まれた値) で指定されたプロパティーを、適切な非ブランク値に置き換える必要があります。 これらのプロパティーは、MFT 構成プロパティーの指定変更を定義します。 ご使用の環境に合わせてエージェントまたはロガーをカスタマイズするために、エージェントおよびロガーのプロパティーを追加することができます。 すべてのプロパティーのリストについては、750 ページの『構成プロパティー・ファイル』を参照してください。

表 45. *BFGSTDIN DD* ステートメント用の必須構成プロパティー

Property	**値**
coordinationQMgr	エージェントまたはロガーが関連付けられる構成のための調整キュー・マネージャーの名前。
coordinationQMgrHost	オプション。 調整キュー・マネージャーが実行されるシステムのホスト名。 このプロパティーの値をブランクのままにした場合、バインディング・モードでの接続が想定されます。
coordinationQMgrPort	オプション。 調整キュー・マネージャーが listen するポート番号。 このパラメーターは、coordinationQMgrHost プロパティーにも非ブランクの値を指定している場合にのみ使用されます。
coordinationQMgrChannel	オプション。 調整キュー・マネージャーへの接続に使用するチャネル。 このパラメーターは、coordinationQMgrHost プロパティーにも非ブランクの値を指定している場合にのみ使用されます。
connectionQMgr	エージェントまたはロガーが関連付けられる構成のためのコマンド・キュー・マネージャーの名前。
connectionQMgrHost	オプション。 コマンド・キュー・マネージャーが実行されるシステムのホスト名。 このプロパティーの値をブランクのままにした場合、バインディング・モードでの接続が想定されます。
connectionQMgrPort	オプション。 コマンド・キュー・マネージャーが listen するポート番号。 このパラメーターは、connectionQMgrHost プロパティーにも非ブランクの値を指定している場合にのみ使用されます。
connectionQMgrChannel	オプション。 コマンド・キュー・マネージャーへの接続に使用するチャネル。 このパラメーターは、connectionQMgrHost プロパティーにも非ブランクの値を指定している場合にのみ使用されます。

z/OS *z/OS* エージェントおよびロガー・コマンドの *JCL* スクリプト

MFT コマンド PDSE ライブラリー・データ・セットにある使用可能な JCL コマンドのセット。

表 46. *MFT* コマンド *PDSE* ライブラリー・データ・セットにある使用可能な *JCL* コマンド

メンバー	**fte コマンド行コマンドの説明**
BFGCOPY	このライブラリーのコピーを作成するジョブ
BFGCUSTM	エージェントまたはロガー用にこのライブラリーをカスタマイズするジョブ
BFGZCFCR	fteSetupCoordination
BFGZCMCR	fteSetupCommands: MFT command.properties ファイル
BFGZAGCR	fteCreateAgent。 SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGLGCRS	fteCreateLogger。 SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。

表 46. *MFT* コマンド *PDSE* ライブラリー・データ・セットにある使用可能な *JCL* コマンド *(*続き*)*

メンバー	fte コマンド行コマンドの説明
BFGZAGST	fteStartAgent。 SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGAGSTP	**fteStartAgent** プロシージャー。 SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGZAGPI	ftePingAgent。 SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGZAGSP	fteStopAgent。 SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGZLGST	fteStartLogger。 SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGLGSTP	**fteStartLogger** プロシージャー。 SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGZLGSP	fteStopLogger。 SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGZAGSH	fteShowAgentDetails. SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGZLGSH	fteShowLoggerDetails. SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGZCFDF	fteChangeDefaultConfigurationOptions
BFGZAGCL	fteCleanAgent。 SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGZAGDE	fteDeleteAgent。 SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGZLGDE	fteDeleteLogger。 SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGZPRSH	fteDisplayVersion
BFGZAGLI	fteListAgents。 SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGZMNLI	fteListMonitors
BFGZSTLI	fteListScheduledTransfers
BFGZTMLI	fteListTemplates
BFGXCROB	**fteObfuscate** サンプル
BFGZRAS	**fteRAS**
BFGZAGTC	fteSetAgentTraceLevel。 SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGZLGTC	fteSetLoggerTraceLevel。 SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGXPRAN	**fteAnt** サンプル

表 46. *MFT* コマンド *PDSE* ライブラリー・データ・セットにある使用可能な *JCL* コマンド *(続き)*	
メンバー	**fte コマンド行コマンドの説明**
BFGXTRCA	**fteCancelTransfer** サンプル
BFGXMNCR	**fteCreateMonitor** サンプル
BFGXTMCR	**fteCreateTemplate** サンプル
BFGXTRCR	**fteCreateTransfer** サンプル
BFGXMNDE	**fteDeleteMonitor** サンプル
BFGXSTDE	**fteDeleteScheduledTransfer** サンプル
BFGXTMDE	**fteDeleteTemplate** サンプル

注:

- MQSC スクリプトを作成するコマンドや削除スクリプトを参照するコマンドの場合、JCL はスクリプトを実行するよう求めてきますが、スクリプトはジョブによって既に実行されています。
- BGCUSTM ジョブを実行すると、BFGZRAS によって BFGRAS メンバーが作成されます。

z/OS 検証転送の実行

製品が正常に動作していることを確認するための転送の実行方法。

メンバー BFGTRCRS の名前を変更し、編集します。

1. %BFGCMD CMD=fteCreateTransfer -h の前に /* を追加します。
2. メンバー内の他のコメントを除去します。
3. -sa および -da で現行エージェント名を指定します。
4. JCL を保存します。
5. JCL を実行依頼します。

この JCL は、コマンド・キュー・マネージャーに接続します。

z/OS ロギング・タスクの構成

ロギング・タスクは、調整キュー・マネージャーと同じイメージで実行する必要があります。 Db2 にログを記録できます。

ロギング・タスクの作成

PDSE をコピーして、ロガー固有の PDSE を作成します。 例えば、 user.MFT.LOGGER などです。

異なる資格情報ファイルを使用する必要がある場合、作成します。 z/OS での MQMFTCredentials.xml の構成を参照してください。

メンバー BFGCUSTM を確認します。 内容のほとんどは、前のカスタマイズと同じです

ただし、以下のことを行う必要があります。

- //SYSEXEC DD DSN=SCEN.FTE.JCL.... を変更する
- エージェント PDSE と一致するように LIBRARY を変更する
- QMGR を調整キュー・マネージャーの名前に変更する
- SERVICE_TYPE=LOGGER を作成する
- NAME を (PDSE と一致する) ロガーの名前になるよう変更する
- JOBCARD を確認し、このジョブ名をエージェントのジョブ名とは異なるものに変更する

- BFG_JVM_PROPERTIES="-Xmx1024M" を確認する

Db2 ロガーを使用している場合、ファイルを作成すると便利です。そうすることにより、Db2 の問題の特定に役立つ Db2 トレースをキャプチャーすることができます。

ファイルの名前は JVM プロパティーで指定されています。 JDBC トレース・プロパティー・ファイルの内容は次のようになります。

```
db2.jcc.traceDirectory=/u/johndoe/fte
db2.jcc.traceFile=jccTrace1
db2.jcc.traceFileAppend=false
# turn on all traces
# db2.jcc.traceLevel=-1
# turn off all traces
db2.jcc.traceLevel=0
```

2 つの JVM プロパティーを設定します。

```
BFG_JVM_PROPERTIES=-Ddb2.jcc.propertiesFile=/u/.../sql.properties
-Ddb2.jcc.ssid=DBCA
```

`/u/.../sql.properties` は Db2 トレース・プロパティー・ファイルの名前で、*DBCA* は Db2 サブシステムの名前です。

このジョブを実行依頼します。その際、このジョブはデータ・セットへの排他的アクセスが必要であることに注意してください。 エージェントのジョブの名前はすべて *BFGLG** のようになります。

ファイルへのロギング

Db2 へのロギングについて詳しくは、747 ページの『ロギング・タスクの作成 (Db2 にロギングする場合)』 を参照してください。

メンバー BFGLGCRS の名前を変更します。 このジョブは Managed File Transfer (MFT) ディレクトリーのファイルを更新し、CSQUTIL を使用してローカル・キュー・マネージャーのエージェント固有キューを作成します。

元のファイルには、コマンドの構文をリストするコマンド `%BFGCMD CMD=fteCreateLogger -h` が含まれています。

ロガー・タスクを作成するには、`%BFGCMD CMD=fteCreateLogger -h` の前に `/*` を置いてこのステートメントをコメント化します。 その際、列 1 は必ずブランクにします。

2 番目のコマンドからコメントを除去し、ステートメントを構成します。 以下に例を示します。

```
%BFGCMD CMD=fteCreateLogger   +
-p MQPH     +
-loggerQMgr MQPH      +
-loggerType FILE        +
-fileLoggerMode circular    +
-fileSize 5MB +
-fileCount 5 +
-p MQPH +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
LOGGER
```

出力を確認して正常に処理されていることを確認します。

ヒント: `logger.properties` ファイルのパス名を、ジョブの出力からエージェントの PDSE のメンバーにコピーします。

例えば、メンバー APATH にコピーします。

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties
```

これは、プロパティー・ファイルを表示する必要がある場合に便利です。

ディレクトリーをこのファイルに追加します。

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/
```

ファイルにログを記録する場合、ログ・ファイルはこのディレクトリーに保管されます (例: LOGGER0-20140522123654897.log)。

トレース・ファイルは、例えば次のようなログ・サブディレクトリーに保管されます。

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

これでロギング・タスクを開始できるようになります。

ロギング・タスクの作成 (Db2 にロギングする場合)

メンバー BFGLGCRS の名前を変更します。

このジョブは MFT ディレクトリーのファイルを更新し、CSQUTIL を使用してローカル・キュー・マネージャーのエージェント固有キューを作成します。

以下を把握しておく必要があります。

表 *47. Db2* の変数

Db2 の名前	**例**
-dbName databaseName	これは、 Db2 サブシステムのメッセージ DSNL004I のロケーション値から取得できます。
- dbDriver filePath	例えば/db2/db2v10/jdbc/classes/db2jcc.jar
-dbLib filePath	例えば/db2/db2v10/jdbc/lib/libdb2jcct2zos_64.so

ファイルを編集します。 元のファイルには、コマンドの構文をリストするコマンド %BFGCMD CMD=fteCreateLogger -h が含まれています。

2 番目のコマンドからコメントを除去し、ステートメントを構成します。 例えば

```
%BFGCMD CMD=fteCreateLogger   +
-p MQPH     +
-loggerQMgr MQPH      +
-loggerType DATABASE      +
-dbType DB2           +
-dbName DSNDBCP         +
-dbDriver /db2/db2v10/jdbc/classes/db2jcc.jar  +
-dbLib /db2/db2v10/jdbc/lib/            +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
LOGGER
```

ロガー・タスクを作成するには、%BFGCMD CMD=fteCreateLogger -h の前に /* を置いてこのステートメントをコメント化します。 その際、列 1 は必ずブランクにします。

ジョブを実行依頼し、出力を確認して正常に処理されていることを確認します。

ヒント: logger.properties ファイルのパス名を、ジョブの出力からエージェントの PDSE のメンバーにコピーします。

例えば、メンバー APATH にコピーします。

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties into member USS
```

これは、プロパティー・ファイルを表示する必要がある場合に便利です。

トレース・ファイルは、例えば次のようなログ・サブディレクトリーに保管されます。

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

Db2 表の作成

Db2 表を作成する必要があります。 定義は z/OS UNIX System Services ファイル `mqft/sql/ftelog_tables_zos.sql` の中にあります。

PDSE にメンバー Db2 を作成します。 このメンバーを編集し、コマンド行で COPY コマンドを使用します。 z/OS UNIX System Services 定義ファイルからコピーします。

サイト固有の要件は大きく異なる場合があるため、このファイルでは表の基本構造と、表が配置される表スペースのみを指定しています。

可能な限り最大の表の行を保持できるページ・サイズを持つバッファー・プールを使用して表が作成されるように、表スペースが SQL スクリプトによって指定されます。 LOB の場所などの属性は指定されないことに注意してください。

データベース管理者は、このファイルのコピーを変更して、パフォーマンスに関連したこれらの属性を定義することもできます。

このファイルでは、デフォルト・スキーマ名として FTELOG、デフォルト表スペース名として FTELOGTS、データベース名として FTELOGDB を使用すると想定します。 必要に応じて、ファイルの先頭にあるコメントで説明されているプロセスに従ってこれらの名前を変更し、既存のデータベースおよびローカル命名規則に一致させることができます。

重要 : ファイルにはコメントがあり、**DSNTINAD** などのバッチ・プログラムはコメントを受け入れないため、**SPUFI** などのオンライン機能を使用してコマンドを実行してください。

詳しくは、「SPUFI を使用した SQL の実行」を参照してください。 さらに、SCSQPROC の CSQ45STB には、 Db2 SELECT コマンドを実行するためにカスタマイズできるサンプル JCL があります。

ロガー・タスクの開始

メンバー BFGLGST の名前を変更し、確認して、実行依頼します。「BFGDB0023I: ロガーは開始アクティビティーを完了し、現在、実行中です」というメッセージが表示されるはずです。

ロガーの操作

ロガーの状況を表示するには、メンバー BFGLGSH の名前を変更し、確認して、実行依頼します。

ロガーを停止するには、メンバー BFGLGSP の名前を変更し、確認して、実行依頼します。

z/OS MFT の環境変数 (z/OS)

z/OS UNIX System Services (z/OS UNIX) 環境から直接、または独自の JCL スクリプトからコマンドを実行している場合には、カスタマイズや構成の後に、Managed File Transfer で提供される構成スクリプトや管理スクリプトを実行するには、その前にいくつかの環境変数を設定する必要があります。 これらの変数は、スクリプトの呼び出し元のユーザーごとおよび環境ごとに設定しなければなりません。

他の製品との競合を避けるために、ホーム・ディレクトリー内に `.wmqfterc` スクリプトを作成することを選択できます。 その後、`.wmqfterc` スクリプトが各 Managed File Transfer スクリプトによって呼び出され、スクリプトを使用して、Managed File Transfer のカスタム環境設定を提供することができます。

さらに、オプションの環境変数 BFG_WTO もあり、この変数は JCL からエージェントを実行する際にオペレーター・ログにメッセージを送信するために設定できます。

表 *48. 必須の z/OS 環境変数*

環境変数	値
BFG_JAVA_HOME	Java インストールの場所。 サポートされている Java のレベルについて詳しくは、IBM MQ のシステム要件を参照してください。

表 48. 必須の z/OS 環境変数 (続き)	
環境変数	**値**
BFG_DATA	Managed File Transfer for z/OS のデータ・ディレクトリーの場所。 これは *DATA_DIR* へのパスです。
STEPLIB	次の IBM MQ データ・セットを含む必要があります。 • SCSQAUTH • SCSQANLE • SCSQLOAD また、z/OS システム上でデータベース・ロガー・コンポーネントを実行する場合は、STEPLIB に、以下の Db2 データ・セットがこの表示順で組み込まれている必要があります。 • SDSNEXIT • SDSNLOD2 • SDSNLOAD

Managed File Transfer の環境変数を正しく構成する `.profile` の例を以下に示します:

```
STEPLIB=MQM.V920.SCSQAUTH:MQM.V920.SCSQANLE:MQM.V920.SCSQLOAD
PATH=/u/fteuser/bin:/u/fteuser/J7.0/bin:/bin:/usr/bin:/u/fteuser/extras/bin:/bin:$PATH
BFG_JAVA_HOME=/u/fteuser/J7.0
BFG_DATA=/u/fteuser/DATA_DIR
export PATH STEPLIB BFG_JAVA_HOME BFG_DATA
```

重要 : `fte*` コマンドを z/OS UNIX 環境から呼び出すときに LIBPATH 環境変数は必要なくなります。このため、既存の `.wmqfterc` スクリプトから削除してください。

また、必要に応じて、以下の環境変数を設定することもできます。

表 49. オプションの z/OS の環境変数

環境変数	値
BFG_WTO	以下のいずれかの値を指定すると、BFG_WTO が使用可能になります。 • YES • ON • TRUE 以下のいずれかの値を指定すると、BFG_WTO が使用不可になります。 これらの値は、大文字と小文字を区別しない。 • NULL • NO • OFF • FALSE z/OS のロギングを使用可能にします。 この環境変数は、デフォルトでは使用不可になります。 エージェントのイベント・ログに書き込まれるメッセージは、z/OS のオペレーター・ログ機能にも書き込まれます。これにより、JCL からエージェントを実行した際に、自動化製品のアクセスがより簡単に行えます。 宛先コードは「プログラマー情報」(11) で、記述子コードは「通知」(12) です。
BFG_GROUP_NAME	mqm ファイル・グループは、通常、Managed File Transfer 構成データ・ファイルおよびコマンドに関連付けられています。 したがって、mqm グループのメンバーであるすべてのユーザーが、Managed File Transfer 構成にアクセスして変更を加えることができます。 詳しくは、IBM MQ の MFT ファイルのシステム許可を参照してください。 z/OS システムで、ファイル・グループは z/OS UNIX ファイル・システム・エンティティーであり、mqm ファイル・グループは必ずしも定義されている必要はありません。 BFG_GROUP_NAME 環境変数を使用して、Managed File Transfer 構成データ・ファイル用に代替の既存の z/OS UNIX ファイル・システム・グループを定義できます。 例えば、z/OS UNIX シェルのプロンプトに次のように入力します。 `export BFG_GROUP_NAME=FTEGB` このようにすると、現在の z/OS UNIX セッションでこれ以後に作成するすべての構成ファイルにグループ FTEGB を関連付ける操作が定義されます。 BFG_GROUP_NAME は、ブランク値に設定することも、削除することもできます。

z/OS 構成プロパティー・ファイル

Managed File Transfer で使用されるプロパティーの要約。

- MFT coordination.properties ファイル
- MFT command.properties ファイル
- MFT agent.properties ファイル
- ロガー構成プロパティー・ファイル

z/OS z/OS 自動リスタート・マネージャー(ARM)の MFT の構成

Managed File Transfer は、ARM を使用可能なアプリケーションです。

始める前に

ARM の使用可能化と、ご使用のシステムに合わせて ARM ポリシーを定義する方法について詳しくは、z/OS 自動再始動管理プログラム (ARM) の使用を参照してください。

MFT DB ロガー機能を使用して、再始動と Db2 データベースへの再接続を自動的に実行する場合は、ARM が唯一の有効な再始動マネージャーになります。

このタスクについて

ARM を使用すると、エージェント/ロガーのプロパティー armELEMTYPE および armELEMENT を設定することにより、エージェントおよびロガーの再始動を構成できます。 プロパティー armELEMTYPE では ARM エレメントのタイプを定義し、プロパティー armELEMENT には ARM を登録するエレメントの名前を指定します。

- エージェントの ELEMTYPE には SYSBFGAG を設定し、armELEMENT にはエージェント名に対応する名前を設定できます。
- ロガーの ELEMTYPE には SYSBFGLG を設定し、armELEMENT にはロガー名に対応する名前を設定できます。

注 : ARM で再始動するように構成したエージェントとロガーは、バッチ・ジョブと開始済みタスクからのみ正常に実行できます。 そのようなエージェントまたはロガーを z/OS UNIX System Services コマンド行から直接開始しようとすると、操作は失敗し、ARM エラーの理由コードが出されます。

例

次に示す再始動ポリシーの例では、エージェント BFGFT7CAG1 をキュー・マネージャー FT7C に依存するものとして定義しています。

```
RESTART_ORDER
   LEVEL(3)
   ELEMENT_TYPE(SYSBFGAG,SYSBFGLG)

RESTART_GROUP(GROUP7C)
   ELEMENT(SYSMQMGRFT7C)
   ELEMENT(BFGFT7CAG1)
       RESTART_ATTEMPTS(3,300)
```

z/OS 例: z/OS での Managed File Transfer エージェント用 JCL の作成

この情報に基づいて、IBM MQ for z/OS でエージェントを作成して開始するための JCL を生成できます。

サンプル・ライブラリーのコピー

以下の手順を実行します。

1. ライブラリーを開いて、ライブラリー SCSQFCMD のコピーを作成します (735 ページの『JCL ライブラリーを作成するための SCSQFCMD のコピー』 を参照)。

   ほとんどのメンバー (BFGX、BFGY、BFGZ で始まる) は、今後エージェントのカスタマイズ JCL を生成する時に使用するテンプレートです。

重要なメンバーは BFGCOPY です。

2. BFGCOPY を開いて、それぞれの値を以下のように置き換えます。

   ++supplied_library++
   プロダクトの一部としてインストールされた SCSQFCMD ライブラリーの名前に置き換えます。

   ++service-library++
   エージェントで使用するライブラリー (ターゲット・ライブラリー) の名前。

3. このジョブを送信すると、新しいライブラリーを使用できるようになります。

BFGCUSTM の編集

以下の手順を実行します。

1. 新しいライブラリーを開きます。BFGCUSTM メンバーを編集できるようになります (735 ページの『メンバー BFGCUSTM の編集』を参照)。
2. そのメンバーで、++ 文字で囲まれているすべてのパラメーターを変更し、それぞれを適切な値に置き換えます。 例えば、次のような行を

   ++mq_path++
   z/OS UNIX System Services (z/OS UNIX) コンポーネント・ディレクトリーへのパス。 例: `/mqm/V9R2M0`。

   注 : この変数には、置き換える必要のある 3 つのインスタンスがあります。

   ++bfg_data++
   IBM MQ Managed File Transfer for z/OS 構成が格納される z/OS UNIX ディレクトリーにします。

   ++service_type++
   AGENT という語にします。

   ++agent_name++
   エージェントの名前にします。

注 :

1. CLEAN_AGENT_PROPS で必要な一部の項目 (`++options++` など) は不要なので、削除します。
2. BFGCUSTM メンバーのすべてのパラメーターとそれぞれの値の説明を載せた詳細なリストについては、729 ページの『z/OS 用の MFT の構成を開始する前に』を参照してください。

BFGCUSTM JCL の実行依頼

以下の手順を実行します。

1. このジョブを送信します。
2. ISPF でライブラリーを終了します。

   この操作が必要なのは、BFGCUSTM ジョブがそのライブラリーを更新する時に、ライブラリーが開いていると更新ができないからです。

3. ジョブが完了したら、ジョブ・ログを確認します。

   新しいメンバーがライブラリー内に作成されたことを示すいくつかのメッセージを確認できます。

   それぞれのメンバーには、エージェントの各タスクを実行するための JCL が含まれています。 これらのメンバーと、対応する IBM MQ Managed File Transfer コマンドのリストについては、 743 ページの『z/OS エージェントおよびロガー・コマンドの JCL スクリプト』 を参照してください。

エージェントを作成する BFGAGCR の実行依頼

新しいメンバー BFGAGCR には、**fteCreateAgent** コマンドを呼び出してエージェントを作成するための JCL が含まれています。

以下の手順を実行します。

1. BFGAGCR メンバーを開きます。

   BFGAGCR に以下の名前が取り込まれているのを確認してください。

   - エージェント
   - エージェント・ キュー・マネージャー
   - MFT トポロジーの調整キュー・マネージャー
2. BFGAGCR メンバーを実行依頼します。

   そのメンバーが実行されると、以下の処理が行われます。

   - エージェントで必要な構成ファイルを作成します。
   - エージェント・キュー・マネージャーに接続し、エージェントに必要なシステム・キューを CSQUTIL で作成します。
   - エージェントを調整キュー・マネージャーに登録します。

BFGAGST の実行依頼によるエージェントの開始

以下の手順を実行します。

1. BFGAGST メンバーを送信します。 エージェントが正しく動作していることを確認するためのさまざまなコマンドについては、エージェントの使用を参照してください。
2. ジョブが完了したら、ジョブ・ログに以下のメッセージが書き込まれていることを確認します。

   ```
   BFGAG0058I: The agent has successfully initialized.
   BFGAG0059I: The agent has been successfully started.
   ```

   エージェントが稼働していて、管理対象転送を実行する準備が整いました。

z/OS MFT エージェントの新規 z/OS LPAR への移動

IBM MQ Managed File Transfer for z/OS エージェントを、同じ調整キュー・マネージャーとコマンド・キュー・マネージャーを持つ同じ IBM MQ Managed File Transfer トポロジー内に保持しながら、そのエージェントをある LPAR から別の LPAR に移動することが必要な場合があります。 これを行うために必要なステップは、マイグレーションされるエージェントがどのように最初に作成されたかによって異なります。

このタスクについて

IBM MQ Managed File Transfer for z/OS エージェントを次のいずれかの方法で移動します。

- エージェントが元々、カスタマイズ・バージョンの SCSQFCMD ライブラリーを使用して作成されていた場合は、そのライブラリーを使用して、新しい LPAR 上にエージェントを再作成します。
- もともとエージェントが z/OS UNIX System Services (z/OS UNIX) コマンドを実行して作成されたものである場合は、コマンドを使用して、新規 LPAR 上にエージェントを再作成します。

注 :

スケジュール済み転送および転送テンプレートは、IBM MQ Managed File Transfer トポロジーの調整キュー・マネージャーに保管されます。 このタスクでは、 調整キュー・マネージャーは移動作業に含まれないものとします。 この場合、移動するエージェントに関連付けられているスケジュール済みの転送および転送テンプレートは、 移動の完了後も既存の調整キュー・マネージャーに残ります。

手順

- カスタマイズ・バージョンの SCSQFCMD ライブラリーを使用して作成されたエージェントを移動します。

  カスタマイズ・バージョンの SCSQFCMD ライブラリーを使用してエージェントを作成した場合は、そのライブラリーを使用して IBM MQ Managed File Transfer for z/OS 環境とエージェント構成を新規 LPAR 上に再作成できます。 これを行うには、以下のステップを実行します。

  1. カスタマイズしたバージョンのライブラリーを元の LPAR から新しい LPAR にコピーします。

2. 新規 LPAR 上のカスタマイズ・バージョンのライブラリーの BFGCUSTM メンバーを編集し、パラメーター値がまだ有効であることを確認します。
3. 新規 LPAR で BFGCUSTM メンバーを実行して、環境の構成およびエージェントの作成に必要なすべての JCL を作成します。
4. BFGCFCR メンバーを実行して、新規 LPAR 上のエージェントが使用する調整キュー・マネージャーを定義し、IBM MQ Managed File Transfer 構成を保管するために必要なディレクトリー構造を作成します。
5. 次に、BFGCMCR メンバーを実行して、新規 LPAR 上のエージェントが使用するコマンド・キュー・マネージャーを定義します。
6. BFGAGCR メンバーを実行して、エージェントとその構成を再作成します。
7. エージェントが使用するシステム・キューが、そのエージェントのキュー・マネージャーに存在することを確認します。

移動するエージェントにリソース・モニターが関連付けられている場合は、新規エージェントでモニターを再作成する必要があります。 これを行うには、以下のステップを実行します。

1. 元の LPAR で BFGMNLI メンバーを実行して、元のエージェントに関連付けられたリソース・モニターの定義を XML ファイルにエクスポートします。
2. リソース・モニター定義を含む XML ファイルを新規 LPAR にコピーします。
3. 新規 LPAR 上の SCSQFCMD ライブラリーの BFGMNCRS メンバーを使用して、XML ファイルに保管されているリソース・モニター定義をインポートします。 これにより、新規エージェントでモニターが作成されます。

- z/OS UNIX でコマンドを実行して作成されたエージェントを移動します。

  もともとエージェントが z/OS UNIX コマンドを実行して作成されたものである場合は、コマンドを使用して、新規 LPAR 上にエージェントを再作成できます。 これを行うには、以下のステップを実行します。

  1. 新規 LPAR で fteSetupCoordination コマンドを実行して、エージェントが使用する調整キュー・マネージャーを定義し、IBM MQ Managed File Transfer 構成を保管するために必要なディレクトリー構造を作成します。
  2. fteSetupCommands コマンドを実行して、新規 LPAR 上のエージェントが使用するコマンド・キュー・マネージャーを定義します。
  3. fteCreateAgent コマンドを実行して、エージェントとその構成を再作成します。
  4. エージェントが使用するシステム・キューが、そのエージェントのキュー・マネージャーに存在することを確認します。

  移動するエージェントにリソース・モニターが関連付けられている場合は、新規エージェントでモニターを再作成する必要があります。 これを行うには、以下のステップを実行します。

  1. 元の LPAR で、 **-ox** パラメーターを指定して fteListMonitors コマンドを実行し、元のエージェントに関連付けられているリソース・モニターの定義を XML ファイルにエクスポートします。
  2. リソース・モニター定義を含む XML ファイルを新規 LPAR にコピーします。
  3. 新しい LPAR で fteCreateMonitor コマンドを実行し、 **-ix** パラメーターを指定して、XML ファイルに保管されているリソース・モニター定義をインポートします。 これにより、新規エージェントでモニターが作成されます。

z/OS V 9.3.0 IBM MQ for z/OS キュー共有グループを使用する MFT インフラストラクチャーの計画

IBM MQ Managed File Transfer (MFT) を使用している場合は、 1 つ以上のエージェント、コマンド、または調整キュー・マネージャーが IBM MQ for z/OS キュー共用グループの一部である場合には、以下の点を考慮する必要があります。

エージェント、コマンド・キュー・マネージャー、および調整キュー・マネージャーの説明については、MFT トポロジーの概要 を参照してください。

エージェント・キュー・マネージャー

通常、MFT Agent は単一のエージェント・キュー・マネージャーに接続し、そのキュー・マネージャーだけが接続可能なローカル・キューを使用します。 エージェントが最初に作成されるときにキュー・マネージャーの名前を指定することで、どのキュー・マネージャーに接続するかがエージェントに通知されます。

IBM MQ for z/OS では、エージェントを作成してからそのキュー・マネージャーの名前をキュー共有グループ (QSG) の名前に置き換えることができます。 つまりエージェントは、ファイル転送を実行するために QSG 内のどの使用可能なキュー・マネージャーにも接続できることを意味します。 エージェントが現在接続しているキュー・マネージャーに障害が発生すると、エージェントはその障害を検出し、QSG 内の代替キュー・マネージャーに再接続します。

IBM MQ 9.2.0 から提供される高可用性エージェント・サポート (777 ページの『Managed File Transfer での高可用性エージェント』を参照) と組み合わせてエージェントを QSG に接続すると、非常に堅固な MFT トポロジーを作成できます。

例えば、次の図では *Agent1* が作成されていて、そのエージェントのキュー・マネージャーは 2 つのキュー・マネージャー *QM1* および *QM2* で構成される QSG です。 エージェント・キューは、結合機能に格納される共有キューとして定義されています。

つまり、エージェントは *LPAR 1* または *LPAR 2* のいずれかで実行でき、*QM1* または *QM2* に接続できます。エージェントが読み書きするファイルとデータ・セットは共有されています。つまり、どちらの LPAR からもアクセスできます。

さらに、エージェントは高可用性エージェントとして構成されています。 この図では、エージェントは *LPAR 1* でアクティブ状態で、エージェントのスタンバイ・インスタンスは *LPAR 2* で実行中です。

このトポロジーは高い回復力を提供します。 *LPAR 1* で実行中のエージェントに障害が発生した場合、キュー・マネージャー *QM1* に障害が発生した場合、または *LPAR 1* に障害が発生した場合は、*LPAR 2* のスタンバイ・インスタンスが障害発生時点からファイル転送処理を引き継いで実行します。

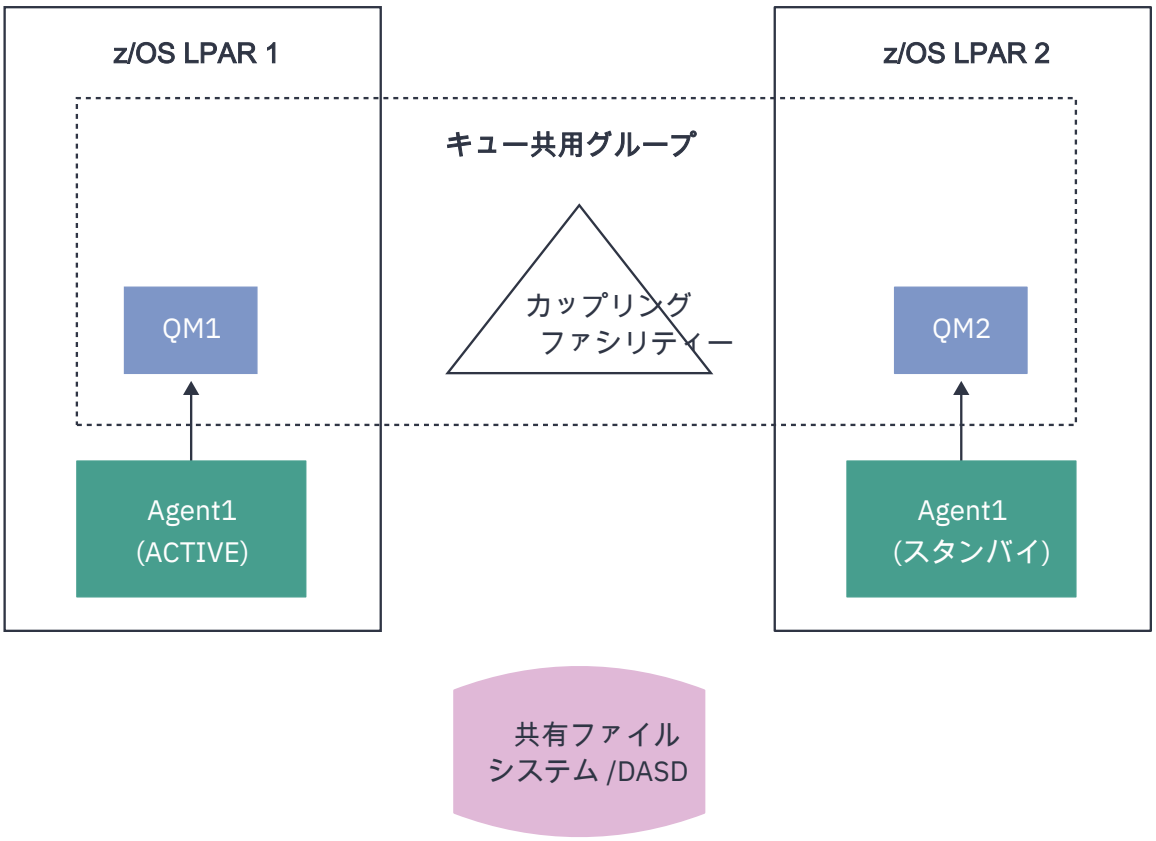


図 *95.* キュー共有グループを使用した高可用性 *MFT* エージェント

エージェント・キュー・マネージャーとして QSG を使用するエージェントの作成

fteCreateAgent コマンドを使用してエージェントを作成します。 その際に、エージェント・キュー・マネージャーに対してキュー共有グループの名前を指定します。 以下に例を示します。

```
fteCreateAgent -agentName Agent1 -agentQMgr QSG1
```

これにより、*Agent1* というエージェントが作成されます。このエージェントは、キュー共有グループ *QSG1* のメンバーである任意のキュー・マネージャーをエージェント・キュー・マネージャーとして使用します。この構成では、エージェントはクロスメモリー (バインディング・モード) 接続を使用してエージェント・キュー・マネージャーに接続します。つまり、エージェントとキュー・マネージャーは同じ LPAR に存在しなければなりません。 これはまさに、上記の図 1 の例と同じです。

fteCreateAgent コマンドを実行すると、エージェント・キュー・マネージャーで必要なキューを作成するための一連の MQSC コマンドが作成されます。

エージェント・キュー・マネージャーが QSG である場合、各キューが共有キューとして作成されるようにこの一連のコマンドを変更する必要があります。 つまり、各キューは QSGDISP(SHARED) と、CFSTRUCT 属性によって指定される適切な結合機能を使用して作成する必要があります。

以下の例は、SYSTEM.FTE.COMMAND.AGENT1 キューを共有キューとして作成するために MQSC コマンドを変更する方法を示しています。 デフォルトのコマンドに対する変更部分は太字で示します。

重要 : エージェントが使用する他のすべてのキューに対しても、同様の変更を行う必要があります。

```
DEFINE QLOCAL(SYSTEM.FTE.COMMAND.AGENT1) +
 QSGDISP(SHARED) +
 CFSTRUCT(MFTSTRUCT) +
 DEFPRTY(0) +
 DEFSOPT(SHARED) +
 GET(ENABLED) +
 INDXTYPE(CORRELID) +
 MAXDEPTH(5000) +
 MAXMSGL(4194304) +
 MSGDLVSQ(PRIORITY) +
 PUT(ENABLED) +
 RETINTVL(999999999) +
 SHARE +
 NOTRIGGER +
 USAGE(NORMAL) +
 REPLACE
```

キュー・マネージャーとして QSG を使用し、クライアントとして接続するエージェントの作成

エージェントは、クライアント・チャネルを使用してそれぞれのエージェント・キュー・マネージャーに接続します。 この方法を使用すると、エージェントは QSG に接続しながら分散プラットフォームで実行できます。 QSG 内のすべてのキュー・マネージャーが IBM MQ Advanced for z/OS Value Unit Edition に対するライセンス交付を受けている場合は、そのエージェントはローカル・キュー・マネージャーを持たない z/OS LPAR からもそれらのキュー・マネージャーに接続できます。

このトポロジーは次の図に示されており、エージェントは QSG の回復力を利用できます。 エージェントが現在接続している QSG 内のキュー・マネージャーに障害が発生した場合、エージェントは QSG の別のメンバーに自動的に再接続して処理を実行します。

エージェントから QSG 内の使用可能なキュー・マネージャーへの接続を配分するには、シスプレックス・ディストリビューターが使用されます。

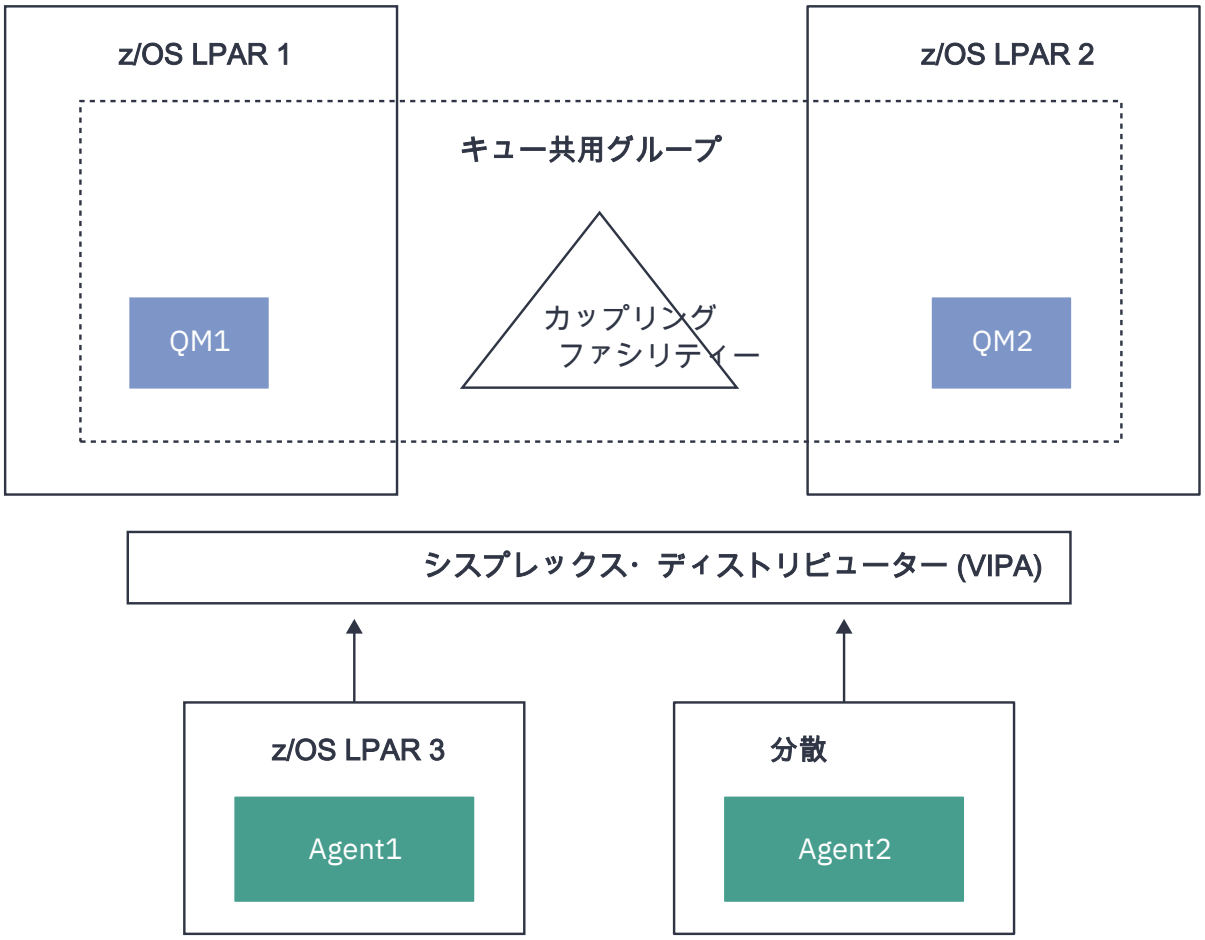


図 96. クライアントとしてキュー共有グループに接続する *MFT* エージェント

このトポロジーを使用するには、QSG 内のキュー・マネージャーごとに、エージェントが使用するように定義されたサーバー接続チャネルを持つ必要があります。 これを行う方法については、61 ページの『キュー共有グループへのクライアントの接続』を参照してください。

エージェントを作成する際には、QSG に対して定義されたチャネルを使用でき、シスプレックス・ディストリビューターを介してそれにアクセスできるようにキュー・マネージャーを構成する必要があります。以下に例を示します。

```
fteCreateAgent -agentName Agent1 -agentQMgr QSG1 -agentQMgrHost vipaAddress
-agentQMgrPort sharedPort -agentQMgrChannel CHANNEL1
```

前述したように、**fteCreateAgent** コマンドを実行することで生成される MQSC コマンドは、QSGDISP(SHARED) と、CFSTRUCT 属性で適切な結合機能を指定するように調整する必要があります。

コマンド・キュー・マネージャー

MFT コマンド・キュー・マネージャーは QSG の一部となることが可能です。 ただし、コマンド・キュー・マネージャーを指定するときに QSG の名前を使用することはできません。固有のキュー・マネージャー名を付ける必要があります。

調整キュー・マネージャー

MFT 調整キュー・マネージャーは QSG の一部となることが可能です。 ただし、コマンド・キュー・マネージャー同様、調整キュー・マネージャーを指定するときに QSG の名前を使用することはできません。固有のキュー・マネージャー名を付ける必要があります。

QSG に接続するためのコマンド

MFT には管理対象エージェント、転送キュー・マネージャー、エージェント・キュー・マネージャー、コマンド・キュー・マネージャー、または調整キュー・マネージャーに対する多くのコマンドが用意されています。 キュー・マネージャーが QSG 内にある場合は、エージェント・キュー・マネージャーに接続するコマンドのみ使用できます。

以下にエージェント・キュー・マネージャーに接続するコマンドのリストを示します。

- **fteCleanAgent**
- **fteCreateAgent**
- **fteCreateBridgeAgent**
- **fteCreateCDAgent**
- **fteDeleteAgent**

これら以外の MFT コマンドを実行するときには、キュー・マネージャーの名前を指定する必要があるので注意してください。

JZOS Java ランチャーで Managed File Transfer for z/OS を使用する

このトピックの説明は、ご使用の IBM MQ for z/OS システムで、エンタープライズ内で Managed File Transfer を使用する代わりの方法として適用できます。

概要

Managed File Transfer for z/OS (MFT)は、標準の z/OS インストール手順を使用します。 MFT コマンドを実行する別の方法は、JCL と JZOS Java ランチャーを使用することです。

詳しくは、『JZOS Batch Launcher and Toolkit』を参照してください。

JCL を正しく処理できない場合は、JZOS でよくある MFT 問題を参照してください。

JCL の例

```
//JOHNDOEA JOB 1,MSGCLASS=H
// JCLLIB ORDER=(SCEN.MFT.JCL)   (1)
// INCLUDE MEMBER=BFGJCL8       (2)
//     DD * (2A)
. ${BFG_PROD}/bin/fteBatch createAgent  (3)
export IBM_JAVA_OPTIONS="${BFG_JAVA_OPTIONS} ${BFG_LANG}"  (4)
export JZOS_MAIN_ARGS="${BFG_MAIN_ARGS}" (4)
//MAINARGS DD *
-agentName MYAGENT  (5)
-f
-agentQMgr MQPD
-p MQPD
/*
```

ここで、

- (1) は、組み込み JCL ステートメントの場所です
- (2) 1) の場所にある指定の JCL メンバーを組み込みます
- (2A) これは //STDENV を拡張します (以下を参照)
- (3) これが、実行されるコマンドです (接頭部 fte は付けない)
- (4) これらの行は必須であり、JZOS の情報をセットアップします
- (5) コマンドへのパラメーター
- BFGJCL8 メンバー (独自の名前を選択できます) が JZOS を呼び出します。 このメンバーには、STEPLIB および MFT の実行に必要なその他の JCL があります。

組み込む必要があるその他の JCL

IBM MQ for z/OS ライブラリーのための JCL と、Db2 ロガーを使用する必要がある場合は Db2 ライブラリーのための JCL を組み込む必要があります。

以下に例を示します。

```
//WMQFTE EXEC PGM=JVMLDM86,REGION=0M PARM='+T'  (1)
//STEPLIB  DD DSN=SYS1.SIEALNKE,DISP=SHR        (2)
//* MQ libraries
//   DD DSN=MQM.V920.SCSQAUTH,DISP=SHR MQ Bindings
//   DD DSN=MQM.V920.SCSQANLE,DISP=SHR MQ Bindings
//   DD DSN=MQM.V920.SCSQLOAD,DISP=SHR MQ Bindings

//* DB2 libraries
//   DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNEXIT.DBCP
//   DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNLOAD
//   DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNLOD2
//SYSOUT  DD SYSOUT=H
//SYSPRINT DD SYSOUT=H
//STDOUT  DD SYSOUT=H
//STDERR  DD SYSOUT=H

//STDENV  DD DSN=SCEN.MFT.JCL(BFGZENV8),DISP=SHR  (3)
```

ここで、

- (1) は、JZOS プログラムの名前です。 SYS1.SIEALNKE でシステム上のバージョンを探します。 追加の診断を行う場合は ,PARM='+T' を追加します。
- (2) これが、JZOS プログラムが含まれているデータ・セットです。
- (3) これは、シェル・スクリプトのメンバー名です。 これは、MFT に必要なパラメーターを定義します。 759 ページの『MFT を定義するためのシェル・スクリプト』を参照してください。

  任意のデータ・セットおよびメンバーを指定できます。 JCL ジョブはこれを拡張するため、ファイルの最後にする必要があります。 758 ページの『JCL の例』 の 2A を参照してください。

MFT を定義するためのシェル・スクリプト

759 ページの『組み込む必要があるその他の JCL』 の例では、メンバー BFGZENV8 が使用されています。これは、JZOS プロファイルに基づきます。

以下を把握しておく必要があります。

- Java がインストールされている場所
- IBM MQ for z/OS Java ライブラリーおよび MFT ライブラリーの場所。
- ユーザー ID が IBM MQ for z/OS 管理者として見なされるためには、特定のグループに属している必要があります。 そのグループの名前が必要です。
- メッセージで英語を使用していない場合は、指定する言語を把握しておく必要があります。

ファイルの例

```
# This is a shell script that configures
# any environment variables for the Java JVM.
# Variables must be exported to be seen by the launcher.
# Use PARM='+T' and set -x to debug environment script problems
set -x
# . /etc/profile
#
# Java configuration (including MQ Java interface)
#
export _BPXK_AUTOCVT="ON"
export JAVA_HOME="/java/java71_bit64_sr3_fp30/J7.1_64/"
export PATH="/bin:${JAVA_HOME}/bin/classic/"
LIBPATH="/lib:/usr/lib:${JAVA_HOME}/bin"
LIBPATH="$LIBPATH:${JAVA_HOME}/bin/classic"
LIBPATH=$LIBPATH:"/mqm/V9R2M0/java/lib/"
export LIBPATH
```

```
export BFG_JAVA_HOME="${JAVA_HOME}"
export BFG_WTO="YES"
export BFG_GROUP_NAME=MQADM
export BFG_PROD="/mqm/V9R2M0/mqft"
export BFG_CONFIG="/u/johndoe/fteconfig"
# export BFG_LANG=" -Duser.language=de "
export BFG_LANG=" "

--------------------
```

ここで、

export _BPXK_AUTOCVT="ON"
Unicode 変換に必要です。

export JAVA_HOME="/java/java71_bit64/J7.1_64/"
これは、 Java ディレクトリーの場所です。 Java のパスの名前を指定します。 このディレクトリーには、bin ディレクトリーおよびその他のディレクトリーが含まれています。

export PATH="/bin:${JAVA_HOME}/bin/classic/"
Java 実行可能ステートメントのパス・ステートメントをセットアップします。

LIBPATH="/lib:/usr/lib:${JAVA_HOME}/bin"
Java 実行可能ステートメントのライブラリー・パスをセットアップします。

LIBPATH="$LIBPATH:${JAVA_HOME}/bin/classic"
LIBPATH ステートメントにさらに Java ライブラリーを追加します。

LIBPATH=$LIBPATH:"/mqm/V9R2M0/java/lib/"
ライブラリー・パスに IBM MQ for z/OS ライブラリーを追加します。 z/OS UNIX System Services 内の IBM MQ for z/OS ライブラリーの名前を指定します。

export LIBPATH
LIBPATH を JZOS で使用できるようにします。

export BFG_JAVA_HOME="${JAVA_HOME}"
BFG_JAVA_HOME に、上記で指定した JAVA_HOME の値を設定します。

export BFG_WTO="YES"
BFG_WTO を YES に設定すると、WTO を使用するジョブ・ログにメッセージが表示されます。

export BFG_GROUP_NAME=MQADM
指定したグループのメンバーであるユーザー ID は、IBM MQ for z/OS 管理者と見なされます。

export BFG_PROD = "/mqm/V9R2M0/mqft"
MFT コードが置かれているパスです。

export BFG_DATA="/u/johndoe/fteconfig"
MFT 構成情報が保管されている場所です。

 # export BFG_LANG=" -Duser.language=de "
言語をドイツ語として定義するステートメントをコメント化したものです。

export BFG_LANG=" "
言語にデフォルト (英語) を指定します。

`/lib/messages/BFGNVMessages_*.properties` の MFT 製品の内容には、使用可能な言語がリストされています。 デフォルトにするには、値をブランクにします。これは、英語が使用されることを意味します。

関連タスク

728 ページの『Managed File Transfer for z/OS の構成』
Managed File Transfer for z/OS では、コンポーネントを正常に動作させるためのカスタマイズが必要です。

Managed File Transfer の計画

IBM i IBM i での MFT の構成

Managed File Transfer のインストール後に、この製品の使用を開始するには、調整キュー・マネージャーおよびエージェントの構成の一部を実行する必要があります。

このタスクについて

インストールの完了後、調整キュー・マネージャーおよびエージェントを使用してファイルを転送できるようにするには、その前に、Managed File Transfer で提供されている、新規調整キュー・マネージャーおよび新規エージェント用の構成スクリプトを実行する必要があります。 その後、作成したエージェントを開始する必要があります。

手順

1. すべての新規調整キュー・マネージャーの場合:調整キュー・マネージャーに *coordination_qmgr_name*.mqsc ファイル内で MQSC コマンドを実行します。 調整キュー・マネージャーがインストール済み環境と同じコンピューター上にない場合、キュー・マネージャーがあるコンピューターに MQSC スクリプト・ファイルをコピーしてから、スクリプトを実行します。

   a) IBM i のコマンド行で、CALL QSHELL コマンドを使用して qshell を開始します。

   b) 次のディレクトリーに移動します: /QIBM/UserData/mqm/mqft/config/*coordination_qmgr_name*

   c) *coordination_qmgr_name* をキュー・マネージャーの名前に置き換えて、以下のコマンドを発行します。

   ```
   /QSYS.LIB/QMQM.LIB/RUNMQSC.PGM coordination_qmgr_name < coordination_qmgr_name.mqsc
   ```

   この代わりに、調整キュー・マネージャーを手動で構成できます。 詳細については、765 ページの『MFT 用の調整キュー・マネージャーの構成』を参照してください。

2. すべての新規エージェントの場合:エージェント・キュー・マネージャーに MQSC コマンドを *agent_name*_create.mqsc ファイルで実行します。

   エージェントのキュー・マネージャーがそのエージェントと同じコンピューター上にない場合、キュー・マネージャーがあるコンピューターに MQSC スクリプト・ファイルをコピーしてから、スクリプトを実行します。

   a) IBM i のコマンド行で、CALL QSHELL コマンドを使用して qshell を開始します。

   b) 次のディレクトリーに移動します: /QIBM/UserData/mqm/mqft/config/*agent_qmgr_name*/agents

   c) *agent_qmgr_name* をエージェント・キュー・マネージャーの名前に置き換え、*agent_name* をエージェントの名前に置き換えて、以下のコマンドを発行します。

   ```
   /QSYS.LIB/QMQM.LIB/RUNMQSC.PGM agent_qmgr_name < agent_name_create.mqsc
   ```

   この代わりに、エージェント・キュー・マネージャーを手動で構成できます。 詳しくは、772 ページの『MFT エージェント・キュー・マネージャーの構成』 を参照してください。

3. インストールの一部として QMFT サブシステムをまだ開始していない場合は、 IBM i コマンド行から、コマンド STRSBS SBSD(QMQMMFT/QMFT) または STRSBS QMQMMFT/QMFT を使用して QMFT サブシステムを開始します。

4. **fteStartAgent** コマンドを使用して、新規エージェントを開始します。

   a) IBM i のコマンド行で、CALL QSHELL コマンドを使用して qshell を開始します。

   b) 次のディレクトリーに移動します: /QIBM/ProdData/mqm/bin

   c) AGENT をエージェントの名前に置き換えて、以下のコマンドを発行します。

   ```
   ./fteStartAgent AGENT
   ```

次のタスク

エージェントがアクセスできるファイル・システムの領域を限定するために、サンドボックスをセットアップすることをお勧めします。 このフィーチャーについての説明は、MFT エージェント・サンドボックスの処理にあります。

関連概念

762 ページの『初めて使用する際の MFT の構成』
Managed File Transfer のエージェントおよびキュー・マネージャーの構成タスクの中には、初回の使用時に一度実行すれば済むものもあります。

初めて使用する際の MFT の構成

Managed File Transfer のエージェントおよびキュー・マネージャーの構成タスクの中には、初回の使用時に一度実行すれば済むものもあります。

関連概念

762 ページの『IBM MQ への接続』
Managed File Transfer に関連する通信を含め、 IBM MQ キュー・マネージャーとのすべてのネットワーク通信には、IBM MQ チャネルが関与します。 IBM MQ チャネルはネットワーク・リンクの一端を表します。チャネルは、メッセージ・チャネルまたは MQI チャネルのいずれかに分類されます。

768 ページの『MFT と連動するマルチインスタンス・キュー・マネージャーの構成』
IBM WebSphere MQ 7.0.1 以降では、マルチ・インスタンス・キュー・マネージャーの作成をサポートしています。 マルチ・インスタンス・キュー・マネージャーはスタンバイ・サーバーで自動的に再始動します。Managed File Transfer は、マルチ・インスタンス・エージェント・キュー・マネージャー、マルチ・インスタンス調整キュー・マネージャー、およびマルチ・インスタンス・コマンド・キュー・マネージャーへの接続をサポートします。

関連タスク

764 ページの『MFT ネットワーク・キュー・マネージャーの構成』
ご使用の Managed File Transfer ネットワークに複数の IBM MQ キュー・マネージャーが含まれている場合、これらの IBM MQ キュー・マネージャーは、互いにリモート側で通信できる必要があります。

772 ページの『MFT エージェント・キュー・マネージャーの構成』
インストール後に、*MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/agents/*agent_name* ディレクトリーで *agent_name*_create.mqsc スクリプトを実行し、エージェント・キュー・マネージャーに必要な構成を実行します。 ただし、この構成を手動で行う場合は、エージェント・キュー・マネージャーに対して以下の手順を実行します。

765 ページの『MFT 用の調整キュー・マネージャーの構成』
fteSetupCoordination コマンドを実行したら、*MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* ディレクトリー内の *coordination_qmgr_name*.mqsc スクリプトを実行して、調整キュー・マネージャーに関する必要な構成を行います。 ただし、この構成を手動で行う場合は、調整キュー・マネージャーに対して以下の手順を実行します。

727 ページの『MFT エージェントまたはロガーのコマンド・データ・セットの作成』
特定の調整のために、特定の Managed File Transfer Agent または Managed File Transfer Logger の Managed File Transfer コマンド・テンプレート・データ・セットからコマンドの PDSE データ・セットを作成することができます。

739 ページの『MFT の既存のエージェントまたはロガーのコマンド・データ・セットの更新 (z/OS)』
Managed File Transfer コマンド・テンプレート・データ・セットから作成される Managed File Transfer コマンド PDSE ライブラリー・データ・セットを更新できます。

関連資料

MFT エージェント・キュー設定

MFT システム・キューおよびシステム・トピック

770 ページの『MFT ログ・メッセージの保持』
Managed File Transfer は、ファイル転送進行情報およびログ情報を調整キュー・マネージャーに送信します。 調整キュー・マネージャーは、この情報を SYSTEM.FTE トピックに一致するサブスクリプションにパブリッシュします。 サブスクリプションがない場合、この情報は保持されません。

IBM MQ への接続

Managed File Transfer に関連する通信を含め、 IBM MQ キュー・マネージャーとのすべてのネットワーク通信には、IBM MQ チャネルが関与します。 IBM MQ チャネルはネットワーク・リンクの一端を表します。チャネルは、メッセージ・チャネルまたは MQI チャネルのいずれかに分類されます。

Managed File Transfer およびチャネル

Managed File Transfer は MQI チャネルを使用して、クライアント・モードでエージェントをそのエージェント・キュー・マネージャーに接続したり、コマンド・アプリケーション (例えば **fteCreateTransfer**) をそのコマンド・キュー・マネージャーや調整キュー・マネージャーに接続したりします。 デフォルト構成では、これらの接続は SYSTEM.DEF.SVRCONN という SVRCONN チャネルを使用して確立されます。これは、デフォルトですべてのキュー・マネージャーに存在します。 これらのデフォルトがあるので、基本 Managed File Transfer インストールでは、どの MQI チャネルも変更不要です。

メッセージ・チャネル・エンドポイントは 6 種類ありますが、このトピックでは送信側と受信側の対についてのみ扱います。 他のチャネルの組み合わせについては、分散キューイング・コンポーネントを参照してください。

必要なメッセージ・パス

IBM MQ メッセージはメッセージ・チャネル経由でのみ移動が可能なため、Managed File Transfer が必要とするすべてのメッセージ・パスでチャネルを使用できるようにする必要があります。 これらのパスは、直接的である必要はありません。メッセージは、必要に応じて、中間キュー・マネージャーを経由することができます。 このトピックでは、直接的な Point-to-Point 通信についてのみ扱います。 これらのオプションの詳細については、リモート・キュー・マネージャーへのアクセス方法を参照してください。

Managed File Transfer で使用される通信パスは、次のとおりです。

エージェントからエージェント
ファイルの転送が行われる 2 つのエージェントは、関連付けられているキュー・マネージャー間の双方向通信を必要とします。 このパスは大量のデータを渡すため、必要に応じて、パスをできるだけ短く、高速で、低コストにすることを検討してください。

エージェントから調整
転送に参加するエージェントからのログ・メッセージは、調整キュー・マネージャーに到達できなければなりません。

コマンドからエージェント
コマンド・アプリケーションまたは IBM MQ Explorer (コマンド・キュー・マネージャーを使用する) が接続するどのキュー・マネージャーも、これらのコマンド・アプリケーションを使用して制御するエージェントのキュー・マネージャーにメッセージを送信できなければなりません。 コマンドによってフィードバック・メッセージを表示できるようにするには、双方向接続を使用します。

詳しくは、企業で使用しているプラットフォーム用の『*IBM MQ* インストール済み環境の検証』を参照してください。

関連概念

768 ページの『MFT と連動するマルチインスタンス・キュー・マネージャーの構成』
IBM WebSphere MQ 7.0.1 以降では、マルチ・インスタンス・キュー・マネージャーの作成をサポートしています。 マルチ・インスタンス・キュー・マネージャーはスタンバイ・サーバーで自動的に再始動します。Managed File Transfer は、マルチ・インスタンス・エージェント・キュー・マネージャー、マルチ・インスタンス調整キュー・マネージャー、およびマルチ・インスタンス・コマンド・キュー・マネージャーへの接続をサポートします。

関連タスク

764 ページの『MFT ネットワーク・キュー・マネージャーの構成』
ご使用の Managed File Transfer ネットワークに複数の IBM MQ キュー・マネージャーが含まれている場合、これらの IBM MQ キュー・マネージャーは、互いにリモート側で通信できる必要があります。

765 ページの『MFT 用の調整キュー・マネージャーの構成』
fteSetupCoordination コマンドを実行したら、*MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* ディレクトリー内の *coordination_qmgr_name*.mqsc スクリプトを実行して、調整キュー・マネージャーに関する必要な構成を行います。 ただし、この構成を手動で行う場合は、調整キュー・マネージャーに対して以下の手順を実行します。

MFT ネットワーク・キュー・マネージャーの構成

ご使用の Managed File Transfer ネットワークに複数の IBM MQ キュー・マネージャーが含まれている場合、これらの IBM MQ キュー・マネージャーは、互いにリモート側で通信できる必要があります。

このタスクについて

キュー・マネージャーの相互通信を可能にするための構成には、次の 2 つの方法があります。

- IBM MQ キュー・マネージャー・クラスターをセットアップする方法。

  IBM MQ キュー・マネージャー・クラスターの情報や構成方法については、『287 ページの『キュー・マネージャー・クラスターの構成』』を参照してください。
- キュー・マネージャー間のチャネルをセットアップする方法。この方法については、以下で説明します。

キュー・マネージャー間のチャネルのセットアップ

キュー・マネージャー間に以下のメッセージ・チャネルをセットアップします。

- エージェント・キュー・マネージャーから調整キュー・マネージャー
- コマンド・キュー・マネージャーからエージェント・キュー・マネージャー
- エージェント・キュー・マネージャーからコマンド・キュー・マネージャー (コマンドによってフィードバック・メッセージを表示できるようにするため)
- コマンド・キュー・マネージャーから調整キュー・マネージャー
- エージェント・キュー・マネージャーから Managed File Transfer のネットワーク内に含まれる他のエージェント・キュー・マネージャー

この通信のセットアップ方法についてさらに情報が必要な場合は、まず『MQSC を使用したリモート IBM MQ オブジェクトの管理』を参照してください。

提案されているサンプルとなるステップのいくつかは、以下のとおりです。

手順

1. IBM MQ のキュー・マネージャー上に、調整キュー・マネージャーと同じ名前で伝送キューを作成する。

   以下の MQSC コマンドを使用できます。

   ```
   DEFINE QLOCAL(coordination-qmgr-name) USAGE(XMITQ)
   ```

2. IBM MQ キュー・マネージャーで、Managed File Transfer の調整キュー・マネージャーへの送信側チャネルを作成する。

   前のステップで作成された伝送キューの名前は、このチャネルの必須パラメーターです。

   Managed File Transfer for IBM MQ 上のエージェントの場合、メッセージはブランク・フォーマットでパブリッシュされます。

   以下の MQSC コマンドを使用できます。

   ```
   DEFINE CHANNEL(channel-name) CHLTYPE(SDR) CONNAME('coordination-qmgr-host(coordination-qmgr-port)')
    XMITQ(coordination-qmgr-name) CONVERT(NO)
   ```

   注 : CONVERT(NO) は、必要なときにのみ設定してください。

3. Managed File Transfer の調整キュー・マネージャーで、IBM MQ キュー・マネージャーへの受信側チャネルを作成する。 この受信側チャネルに、IBM MQ のキュー・マネージャー上の送信側チャネルと同じ名前を付けます。

   以下の MQSC コマンドを使用できます。

   ```
   DEFINE CHANNEL(channel-name) CHLTYPE(RCVR)
   ```

次のタスク

次いで、調整キュー・マネージャーの構成ステップとして、 『765 ページの『MFT 用の調整キュー・マネージャーの構成』』に従います。

関連概念

762 ページの『IBM MQ への接続』
Managed File Transfer に関連する通信を含め、 IBM MQ キュー・マネージャーとのすべてのネットワーク通信には、IBM MQ チャネルが関与します。 IBM MQ チャネルはネットワーク・リンクの一端を表します。チャネルは、メッセージ・チャネルまたは MQI チャネルのいずれかに分類されます。

768 ページの『MFT と連動するマルチインスタンス・キュー・マネージャーの構成』
IBM WebSphere MQ 7.0.1 以降では、マルチ・インスタンス・キュー・マネージャーの作成をサポートしています。 マルチ・インスタンス・キュー・マネージャーはスタンバイ・サーバーで自動的に再始動します。Managed File Transfer は、マルチ・インスタンス・エージェント・キュー・マネージャー、マルチ・インスタンス調整キュー・マネージャー、およびマルチ・インスタンス・コマンド・キュー・マネージャーへの接続をサポートします。

関連タスク

765 ページの『MFT 用の調整キュー・マネージャーの構成』
fteSetupCoordination コマンドを実行したら、*MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* ディレクトリー内の *coordination_qmgr_name*.mqsc スクリプトを実行して、調整キュー・マネージャーに関する必要な構成を行います。 ただし、この構成を手動で行う場合は、調整キュー・マネージャーに対して以下の手順を実行します。

MFT 用の調整キュー・マネージャーの構成

fteSetupCoordination コマンドを実行したら、*MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* ディレクトリー内の *coordination_qmgr_name*.mqsc スクリプトを実行して、調整キュー・マネージャーに関する必要な構成を行います。 ただし、この構成を手動で行う場合は、調整キュー・マネージャーに対して以下の手順を実行します。

このタスクについて

手順

1. SYSTEM.FTE という名前のローカル・キューを作成します。
2. SYSTEM.FTE キューを SYSTEM.QPUBSUB.QUEUE.NAMELIST 名前リストに追加します。
3. SYSTEM.FTE という名前のトピックを、トピック・ストリング SYSTEM.FTE で作成します。
4. SYSTEM.FTE トピックの非永続メッセージ配信 (NPMSGDLV) 属性と永続メッセージ配信 (PMSGDLV) 属性が ALLAVAIL に設定されていることを確認します。
5. 調整キュー・マネージャーのパブリッシュ/サブスクライブ・モード (PSMODE) 属性が ENABLED に設定されていることを確認します。

次のタスク

調整キュー・マネージャーとして構成されているキュー・マネージャーで `strmqm -c` コマンドを実行すると、このコマンドは ステップ 2 で行われた変更を削除します (SYSTEM.FTE キューを SYSTEM.QPUBSUB.QUEUE.NAMELIST 名前リストに追加します)。 これは、`strmqm -c` によって、デフォルトの IBM MQ オブジェクトが再作成され、Managed File Transfer の変更が元に戻されるためです。 したがって、`strmqm -c` を使用してキュー・マネージャーを開始した場合は、以下のいずれかの手順を実行します。

- キュー・マネージャーで *coordination_qmgr_name*.mqsc スクリプトを再度実行します。
- ステップ 2 を繰り返します。

関連概念

762 ページの『IBM MQ への接続』

Managed File Transfer に関連する通信を含め、 IBM MQ キュー・マネージャーとのすべてのネットワーク通信には、 IBM MQ チャネルが関与します。 IBM MQ チャネルはネットワーク・リンクの一端を表します。チャネルは、メッセージ・チャネルまたは MQI チャネルのいずれかに分類されます。

768 ページの『MFT と連動するマルチインスタンス・キュー・マネージャーの構成』
IBM WebSphere MQ 7.0.1 以降では、マルチ・インスタンス・キュー・マネージャーの作成をサポートしています。 マルチ・インスタンス・キュー・マネージャーはスタンバイ・サーバーで自動的に再始動します。Managed File Transfer は、マルチ・インスタンス・エージェント・キュー・マネージャー、マルチ・インスタンス調整キュー・マネージャー、およびマルチ・インスタンス・コマンド・キュー・マネージャーへの接続をサポートします。

関連タスク

764 ページの『MFT ネットワーク・キュー・マネージャーの構成』
ご使用の Managed File Transfer ネットワークに複数の IBM MQ キュー・マネージャーが含まれている場合、これらの IBM MQ キュー・マネージャーは、互いにリモート側で通信できる必要があります。

関連資料

fteSetupCoordination

IBM MQ File Transfer 構造の作成

Managed File Transfer 構造は、同じマシン上のキュー・マネージャーに接続されている単一のエージェントに基づいて構成できます。

このタスクについて

MFT の構成は、エージェントがあるマシン上の IBM MQ DataPath の下のファイル構造に保管されます。

以下のサンプル構成は、SAMPLECOORD (セキュリティー無効) という名前の IBM MQ 8.0 キュー・マネージャー内の MFT と、 SAMPLEAGENT という名前の単一の MFT エージェントを対象としています。

```
+--- config
             +--- SAMPLECOORD
                             +--- command.properties
                             +--- coordination.properties
                             +--- SAMPLECOORD.mqsc
                             +--- agents
                                          +--- SAMPLEAGENT
                                                       +--- agent.properties
                                                       +--- SAMPLEAGENT_create.mqsc
                                                       +--- SAMPLEAGENT_delete.mqsc

+---logs
             +--- SAMPLECOORD
                             +--- agents
                                          +--- SAMPLEAGENT
                                                       +--- logs
```

この例は、キュー・マネージャーのセキュリティーが無効になっていると想定しています。 以下のコマンドは **runmqsc** で実行され、キュー・マネージャーの再始動後にセキュリティーを無効にします。

```
runmqsc queue manager
 alter qmgr CONNAUTH(NONE);
 alter qmgr CHLAUTH(DISABLED);
 end;
```

IBM MQ 8.0 以降の MFT でセキュリティーを有効にした構成の場合、 **CONNAUTH** では、キュー・マネージャーに接続するすべての MFT コマンドでユーザー ID とパスワードの資格情報を提供する必要があります。 各コマンドに追加のパラメーター **-mquserid** および **-mqpassword** を適用するか、MQMFTCredentials.xml ファイルを定義することができます。 以下のサンプル資格情報ファイルでは、キュー・マネージャー SAMPLECOORD に接続する際に、ユーザー ID として fteuser、そのパスワードとして MyPassword を使用するように定義されています。

```
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://wmqfte.ibm.com/MQMFTCredentials MQMFTCredentials.xsd">
```

```
    <tns:qmgr mqPassword="MyPassword" MyUserId="fteuser" name="SAMPELCOORD"/>
</tns:mqmftCredentials>
```

詳しくは、MFT と IBM MQ の接続認証を参照してください。

注:

- MFT の構成ディレクトリーを見つけるには、**fteDisplayVersion -v** コマンドを使用します。
- z/OS ユーザーの場合、 MQMFTCredential.xml ファイルは、可変長レコード・フォーマット (RECFM = V) または不定形式レコード・フォーマット (RECFM = U) の区分データ・セットのメンバーとして見つけることができます。
- セキュリティーを有効にした構成の場合は、以下のステップにパラメーター -F *full_credential_file_path* を追加して、資格情報を関連するキュー・マネージャーに関連付けます。
- MQMFTCredential.xml 内の平文のパスワードは、以下のコマンドを使用して難読化できます:

```
fteObfuscate -f full_file_path_to_MQMFTCredentials.xml
```

手順

1. 調整キュー・マネージャーを作成します。

   調整キュー・マネージャーとは、単一のキュー・マネージャーであり、すべての転送ログおよび状況情報をそのエージェントから受け取るために使用されます。 以下のコマンドを実行します。

   ```
   fteSetupCoordination -coordinationQMgr coordination_qmgr_name
   ```

   これにより、基本的な最上位構成が作成され、*coordination_qmgr_name*.mqsc を呼び出す IBM MQ スクリプト・ファイルが作成されます。

   その後、次の IBM MQ コマンドを実行して構成をキュー・マネージャーにロードする必要があります。

   ```
   runmqsc queue manager name < coordination_qmgr_name.mqsc
   ```

   注: キュー・マネージャーへの TCP クライアント接続では、以下を使用できます。

   ```
   fteSetupCoordination -coordinationQMgr coordination_qmgr_name
   -coordinationQMgrHost coordination_qmgr_host -coordinationQMgrPort coordination_qmgr_port
   -coordinationQMgrChannel coordination_qmgr_channel
   ```

   作成された *coordination_qmgr_name*.mqsc では、調整キュー・マネージャーが実行されているマシンで **runmqsc** コマンドを実行する必要があります。

2. コマンド・キュー・マネージャーを作成します。

   コマンド・キュー・マネージャーとは、単一のキュー・マネージャーであり、これは、IBM MQ インフラストラクチャーが MFT 要求を関連エージェントに経路指定できるよう事前構成されています。 以下のコマンドを実行します。

   ```
   fteSetupCommands -connectionQMgr Command QM Name -p Coordination QM Name
   ```

   これにより、調整ディレクトリーに command.properties ファイルが作成されます。 -p はオプションであり、コマンドがデフォルト調整のためにセットアップされる場合には必要ないことに注意してください。

   注: キュー・マネージャーへの TCP クライアント接続では、以下を使用できます。

   ```
   fteSetupCommands -p coordination_qmgr_name -commandQMgr connection_qmgr_name
   -commandQMgrHost connection_qmgr_host -commandQMgrPort connection_qmgr_port
   -commandQMgrChannel connection_qmgr_channel
   ```

3. エージェントを作成します。

   エージェントとは、ファイルを送受信できるアプリケーションのことです。 以下のコマンドを実行します。

```
fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMgr agent_qmgr_name
```

これにより、調整の下にエージェント構成が作成され、エージェントの構成ディレクトリーに *agent_name*.mqsc を呼び出すための IBM MQ スクリプト・ファイルが作成されます。

次の IBM MQ コマンドを実行して、IBM MQ スクリプト・ファイルをキュー・マネージャーにロードします。

```
runmqsc agent_qmgr_name < agent_name_create.mqsc file
```

注 : キュー・マネージャーへの TCP クライアント接続では、以下を使用できます。

```
fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMgr agent_qmgr_name
-agentQMgrHost agent_qmgr_host -agentQMgrPort agent_qmgr_port -agentQMgrChannel
agent_qmgr_channel
```

4. エージェントを開始します。

   以下のコマンドを実行します。

```
fteStartAgent -p coordination_qmgr_name agentName
```

   エージェントはバックグラウンドで開始され、コマンド・プロンプトが返されます。 エージェントが実行されていることを確認するには、次のコマンドを実行します。

```
fteListAgents -p coordination_qmgr_name
```

   これにより、エージェントの状況が表示されます。 エージェントが正常に実行されている場合、READY 状態にあると報告されます。

タスクの結果

MFT の基本インフラストラクチャーの使用準備が整い、**fteCreateTransfer** コマンドを使用して転送を要求できるようになります。 あるいは、IBM MQ Explorer を使用できる場合は、MFT プラグインを使用して転送を作成およびモニターします。

『ステップ 3: エージェントを作成します』を繰り返して、さらにエージェントを構成に追加できます。 TCP クライアント接続を使用する場合、これらのエージェントを複数の異なるマシンに配置することができます。 異なるマシンに配置する場合には **fteSetupCoordination** コマンドと **fteSetupCommands** コマンドを各マシンで繰り返す必要がありますが、mqsc スクリプトを実行する必要はありません。

より複雑な構成の場合は、調整および各エージェントにそれぞれ別個のキュー・マネージャーを設定することができます。 その場合には、各種キュー・マネージャーを接続する必要があります。

関連概念

MFT エージェントが **fteListAgents** コマンドによってリストされない場合に行う事柄

関連資料

fteSetupCoordination

fteSetupCommands: MFT command.properties ファイルの作成

fteCreateAgent

fteObfuscate: 機密データの暗号化

MFT の資格情報ファイルのフォーマット

MFTagent.properties ファイル

MFT と連動するマルチインスタンス・キュー・マネージャーの構成

IBM WebSphere MQ 7.0.1 以降では、マルチ・インスタンス・キュー・マネージャーの作成をサポートしています。 マルチ・インスタンス・キュー・マネージャーはスタンバイ・サーバーで自動的に再始動します。 Managed File Transfer は、マルチ・インスタンス・エージェント・キュー・マネージャー、マルチ・インスタンス調整キュー・マネージャー、およびマルチ・インスタンス・コマンド・キュー・マネージャーへの接続をサポートします。

マルチ・インスタンス・キュー・マネージャーの構成

重要 : IBM MQ マルチ・インスタンス・キュー・マネージャーの構成については、505 ページの『複数インスタンス・キュー・マネージャー』を参照してください。 Managed File Transfer と連携するように複数インスタンス・キュー・マネージャーを構成する前に、この情報をお読みください。

エージェント・キュー・マネージャーとしてのマルチ・インスタンス・キュー・マネージャーの使用

エージェントがマルチ・インスタンス・キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスの両方に接続できるようにするには、agentQMgrStandby プロパティーをエージェントの agent.properties ファイルに追加します。 agentQMgrStandby プロパティーは、スタンバイ・キュー・マネージャー・インスタンスについてクライアント接続に使用されるホスト名とポート番号を定義します。 プロパティーの値は MQ CONNAME フォーマット、つまり *host_name*(*port_number*) で指定する必要があります。

agentQMgr プロパティーは、マルチ・インスタンス・キュー・マネージャーの名前を指定します。 agentQMgrHost プロパティーは、アクティブ・キュー・マネージャー・インスタンスのホスト名を示し、agentQMgrPort プロパティーは、アクティブ・キュー・マネージャー・インスタンスのポート番号を示します。 エージェントは、マルチ・インスタンス・キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスの両方にクライアント・モードで接続する必要があります。

詳しくは、MFTagent.properties ファイルを参照してください。

この例は、QM_JUPITER というマルチインスタンス・キュー・マネージャーに接続する AGENT1 の agent.properties ファイルの内容を示しています。 QM_JUPITER のアクティブ・インスタンスは、システム host1 上にあり、クライアント接続にポート番号 1414 を使用します。 QM_JUPITER のスタンバイ・インスタンスは、システム host2 上にあり、クライアント接続にポート番号 1414 を使用します。

```
agentName=AGENT1
agentDesc=
agentQMgr=QM_JUPITER
agentQMgrPort=1414
agentQMgrHost=host1
agentQMgrChannel=SYSTEM.DEF.SVRCONN
agentQMgrStandby=host2(1414)
```

調整キュー・マネージャーとしてのマルチ・インスタンス・キュー・マネージャーの使用

マルチ・インスタンス調整キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスの両方の接続を有効にするには、 Managed File Transfer トポロジー内のすべての coordination.properties ファイルに coordinationQMgrStandby プロパティーを追加します。

詳しくは、MFT coordination.properties ファイルを参照してください。

この例は、QM_SATURN という名前のマルチインスタンス調整キュー・マネージャーに接続の詳細を指定する coordination.properties ファイルの内容を示しています。 QM_SATURN のアクティブ・インスタンスは、システム coordination_host1 上にあり、クライアント接続にポート番号 1420 を使用します。 QM_SATURN のスタンバイ・インスタンスは、システム coordination_host2 上にあり、クライアント接続にポート番号 1420 を使用します。

```
coordinationQMgr=QM_SATURN
coordinationQMgrHost=coordination_host1
coordinationQMgrPort=1420
coordinationQMgrChannel=SYSTEM.DEF.SVRCONN
coordinationQMgrStandby=coordination_host2(1420)
```

Managed File Transfer スタンドアロン・ロガーは、そのキュー・マネージャーと常にバインディング・モードで接続している必要があります。 スタンドアロン・ロガーをマルチ・インスタンス調整キュー・マネージャーと共に使用している場合、スタンドアロン・ロガーをバインディング・モードで別のキュー・マネージャーに接続してください。 これを行うステップについては、『804 ページの『MFT スタンドアロン・ロガーの代替構成』』で説明されています。 スタンドアロン・ロガーのキュー・マネージャーと調整キ

ュー・マネージャーの間のチャネルを、マルチ・インスタンス調整キュー・マネージャーの両方のインスタンスのホスト名およびポート番号で定義する必要があります。 この方法については、505 ページの『複数インスタンス・キュー・マネージャー』を参照してください。

IBM MQ Explorer 用の Managed File Transfer プラグインは、クライアント・モードで調整キュー・マネージャーに接続します。 マルチ・インスタンス調整キュー・マネージャーのアクティブ・インスタンスが失敗すると、調整キュー・マネージャーのスタンバイ・インスタンスがアクティブになり、プラグインが再接続します。

Managed File Transfer コマンド **fteList*** および **fteShowAgentDetails** は、調整キュー・マネージャーに直接接続します。 マルチ・インスタンス調整キュー・マネージャーのアクティブ・インスタンスが使用不可になると、これらのコマンドは調整キュー・マネージャーのスタンバイ・インスタンスに接続しようとします。

コマンド・キュー・マネージャーとしてのマルチ・インスタンス・キュー・マネージャーの使用

複数インスタンス・コマンド・キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスの両方の接続を有効にするには、 Managed File Transfer トポロジー内のすべての `command.properties` ファイルに `connectionQMgrStandby` プロパティーを追加します。

詳しくは、MFT command.properties ファイルを参照してください。

この例は、QM_MARS というマルチインスタンス・コマンド・キュー・マネージャーに接続の詳細を指定する `command.properties` ファイルの内容を示しています。 QM_MARS のアクティブ・インスタンスは、システム command_host1 上にあり、クライアント接続にポート番号 1424 を使用します。 QM_MARS のスタンバイ・インスタンスは、システム command_host2 上にあり、クライアント接続にポート番号 1424 を使用します。

```
connectionQMgr=QM_SATURN
connectionQMgrHost=command_host1
connectionQMgrPort=1424
connectionQMgrChannel=SYSTEM.DEF.SVRCONN
connectionQMgrStandby=command_host2(1424)
```

関連概念

762 ページの『IBM MQ への接続』
Managed File Transfer に関連する通信を含め、 IBM MQ キュー・マネージャーとのすべてのネットワーク通信には、IBM MQ チャネルが関与します。 IBM MQ チャネルはネットワーク・リンクの一端を表します。 チャネルは、メッセージ・チャネルまたは MQI チャネルのいずれかに分類されます。

関連タスク

764 ページの『MFT ネットワーク・キュー・マネージャーの構成』
ご使用の Managed File Transfer ネットワークに複数の IBM MQ キュー・マネージャーが含まれている場合、これらの IBM MQ キュー・マネージャーは、互いにリモート側で通信できる必要があります。

765 ページの『MFT 用の調整キュー・マネージャーの構成』
fteSetupCoordination コマンドを実行したら、*MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* ディレクトリー内の *coordination_qmgr_name*.mqsc スクリプトを実行して、調整キュー・マネージャーに関する必要な構成を行います。 ただし、この構成を手動で行う場合は、調整キュー・マネージャーに対して以下の手順を実行します。

MFT ログ・メッセージの保持

Managed File Transfer は、ファイル転送進行情報およびログ情報を調整キュー・マネージャーに送信します。 調整キュー・マネージャーは、この情報を SYSTEM.FTE トピックに一致するサブスクリプションにパブリッシュします。 サブスクリプションがない場合、この情報は保持されません。

情報を確実に保持する方法

転送進行情報またはログ情報が業務上重要な場合、以下のステップのいずれかを実行して、情報が確実に保持されるようにします。

- Managed File Transfer のデータベース・ロガーを使用して、Oracle または Db2 データベースの SYSTEM.FTE/Log トピックにパブリッシュされるメッセージをコピーします。
- サブスクリプションを SYSTEM.FTE トピックに定義する。これにより、IBM MQ キューにパブリケーションが保管されます。 ファイル転送を行う前にこのサブスクリプションを定義して、すべての進行メッセージとログ・メッセージがキューに確実に保持されるようにします。
- メッセージ・キュー・インターフェース (MQI) または IBM MQ JMS を使用して永続サブスクリプションを作成し、そのサブスクリプションに送信されるパブリケーションを処理するアプリケーションを作成する。 このアプリケーションは、すべての進行メッセージとログ・メッセージを確実に受け取るように、ファイルが転送される前に稼働している必要があります。

これらの各手法は、この後のセクションで詳しく説明します。

ログ情報の保存のために IBM MQ Explorer・プラグインに依存することはしないでください。

ログ・メッセージ保存のための Managed File Transfer データベース・ロガーの使用

データベース・ロガーは、Managed File Transfer のオプションのコンポーネントです。このコンポーネントを使用して、分析および監査のために、ログ情報をデータベースにコピーできます。 データベース・ロガーは、調整キュー・マネージャーおよびデータベースをホストするシステムにインストールする、スタンドアロンの Java アプリケーションです。 データベース・ロガーについて詳しくは、782 ページの『MFT ロガーの構成』を参照してください。

IBM MQ Explorer・プラグインを使用した進行メッセージおよびログ・メッセージの保持

IBM MQ Explorer・プラグインのインスタンスが初めて開始されるときに、インスタンスにより永続サブスクリプションが調整キュー・マネージャーに作成されます。 この永続サブスクリプションを使用して、「**転送ログ**」および「**現在の転送進行状況**」ビューに表示される情報が収集されます。

永続サブスクリプションの名前は、サブスクリプションが IBM MQ Explorer MFT プラグインによって作成されたこと、ホスト名、およびユーザーの名前を示すために接頭部が付けられます。 例えば、`MQExplorer_MFT_Plugin_HOST_TJWatson` などです。

この接頭部は、IBM MQ Explorer・プラグインのインスタンスにより実際にはもう使用されていない永続サブスクリプションを管理者が削除する場合に備えて追加されます。

調整キュー・マネージャーで永続サブスクリプションを使用すると、SYSTEM.MANAGED.DURABLE キューにメッセージが蓄積されていく可能性があります。 大容量の Managed File Transfer ネットワークを使用している場合、IBM MQ Explorer プラグインを頻繁に使用するか、または両方を使用すると、このメッセージ・データがローカル・ファイル・システムに入力される可能性があります。

これが発生しないようにするには、 IBM MQ Explorer プラグインが調整キュー・マネージャーに対する非永続サブスクリプションを使用するように指定します。 IBM MQ Explorer で以下の手順を実行してください。

1. 「**ウィンドウ**」 > 「**設定**」 > 「**MQ エクスプローラー**」 > 「**Managed File Transfer**」を選択します。
2. 「**転送ログ・サブスクリプション・タイプ**」リストから「NON_DURABLE」を選択します。

IBM MQ キューへのパブリケーションの保管

IBM MQ キューにログ・メッセージまたは進行メッセージを保管するには、メッセージをこのキューに転送する調整キュー・マネージャー上にサブスクリプションを構成します。 例えば、すべてのログ・メッセージを LOG.QUEUE という名前のキューに転送するには、次の MQSC コマンドを送信します。

```
define sub(MY.SUB) TOPICSTR('Log/#') TOPICOBJ(SYSTEM.FTE) DEST(LOG.QUEUE)WSCHEMA(TOPIC)
```

ログ・メッセージは IBM MQ キューに転送された後、そのキューを使用する IBM MQ アプリケーションによって処理されるまでキューに残ります。

SYSTEM.FTE トピックに対する永続サブスクリプションを管理するアプリケーションの作成

IBM MQ によってサポートされるアプリケーション・プログラミング・インターフェースのいずれかを使用して、SYSTEM.FTE トピックに対する独自の永続サブスクリプションを管理するアプリケーションを作成することができます。 これらのアプリケーションは、IBM MQ キューまたはログ・メッセージを受け取り、ビジネス・ニーズに合わせて、それらを適切に処理できます。

使用可能なアプリケーション・プログラミング・インターフェースについて詳しくは、『アプリケーションの開発』を参照してください。

MFT エージェント・キュー・マネージャーの構成

インストール後に、*MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/agents/*agent_name* ディレクトリーで *agent_name*_create.mqsc スクリプトを実行し、エージェント・キュー・マネージャーに必要な構成を実行します。 ただし、この構成を手動で行う場合は、エージェント・キュー・マネージャーに対して以下の手順を実行します。

手順

1. エージェント操作キューを作成します。
   これらのキューの名前は以下のとおりです。
   - SYSTEM.FTE.COMMAND.*agent_name*
   - SYSTEM.FTE.DATA.*agent_name*
   - SYSTEM.FTE.EVENT.*agent_name*
   - SYSTEM.FTE.REPLY.*agent_name*
   - SYSTEM.FTE.STATE.*agent_name*

   キューのパラメーターとキューの使用方法については、MFT エージェント・キューの設定を参照してください。
2. エージェント権限キューを作成します。
   これらのキューの名前は以下のとおりです。
   - SYSTEM.FTE.AUTHADM1.*agent_name*
   - SYSTEM.FTE.AUTHAGT1.*agent_name*
   - SYSTEM.FTE.AUTHMON1.*agent_name*
   - SYSTEM.FTE.AUTHOPS1.*agent_name*
   - SYSTEM.FTE.AUTHSCH1.*agent_name*
   - SYSTEM.FTE.AUTHTRN1.*agent_name*

   キューのパラメーターとキューの使用方法については、MFT エージェント・キューの設定を参照してください。

次のタスク

プロトコルブリッジエージェントの作成と設定については、fteCreateBridgeAgent（MFT プロトコルブリッジエージェントの作成と設定） および FTPS サーバーのプロトコルブリッジを設定するを参照してください。

関連概念

762 ページの『IBM MQ への接続』
Managed File Transfer に関連する通信を含め、 IBM MQ キュー・マネージャーとのすべてのネットワーク通信には、 IBM MQ チャネルが関与します。 IBM MQ チャネルはネットワーク・リンクの一端を表します。 チャネルは、メッセージ・チャネルまたは MQI チャネルのいずれかに分類されます。

768 ページの『MFT と連動するマルチインスタンス・キュー・マネージャーの構成』

IBM WebSphere MQ 7.0.1 以降では、マルチ・インスタンス・キュー・マネージャーの作成をサポートしています。 マルチ・インスタンス・キュー・マネージャーはスタンバイ・サーバーで自動的に再始動します。 Managed File Transfer は、マルチ・インスタンス・エージェント・キュー・マネージャー、マルチ・インスタンス調整キュー・マネージャー、およびマルチ・インスタンス・コマンド・キュー・マネージャーへの接続をサポートします。

関連タスク

764 ページの『MFT ネットワーク・キュー・マネージャーの構成』
ご使用の Managed File Transfer ネットワークに複数の IBM MQ キュー・マネージャーが含まれている場合、これらの IBM MQ キュー・マネージャーは、互いにリモート側で通信できる必要があります。

765 ページの『MFT 用の調整キュー・マネージャーの構成』
fteSetupCoordination コマンドを実行したら、*MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* ディレクトリー内の *coordination_qmgr_name*.mqsc スクリプトを実行して、調整キュー・マネージャーに関する必要な構成を行います。 ただし、この構成を手動で行う場合は、調整キュー・マネージャーに対して以下の手順を実行します。

関連資料

MFT エージェント・キュー設定
fteSetupCoordination

複数のチャネルに対応した MFT エージェントの構成 (クラスター)

クラスター構成で IBM MQ マルチチャネル・サポートを使用する場合は、まず **agentMultipleChannelsEnabled** プロパティーを true に設定してから、このトピックのステップを実行します。

このタスクについて

クラスターの場合、マルチチャネル・サポートは、宛先エージェントのみのキュー・マネージャーの IBM MQ 定義で使用可能にされます。

762 ページの『初めて使用する際の MFT の構成』にリストされている Managed File Transfer エージェントに必要な標準の IBM MQ 構成ステップに加えて、このトピックのステップを実行する必要があります。

以下の構成例では、**runmqsc** コマンドを使用します。

手順

1. 使用するチャネルごとに、クラスター受信側チャネルを定義します。 例えば、2 つのチャネルを使用する場合、以下のようになります。

   ```
   DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
   DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
   ```

   ここで、

   - *DESTQMGRNAME* は、宛先エージェントのキュー・マネージャーの名前です。
   - *MFTCLUSTER* は IBM MQ クラスターの名前です。

   チャネルには *MFTCLUSTER.DESTMGRNAME*_n という命名規則を使用することが推奨されていますが、この規則は必須ではありません。
2. 各チャネルに対応するキュー・マネージャー別名を定義します。 以下に例を示します。

   ```
   DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
   DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
   ```

   SYSTEM.FTE.*DESTQMGRNAME_n* キュー・マネージャー別名 の命名規則。送信側エージェントがこの形式のキュー・マネージャー別名を検索するためです。 *n* に使用する番号は、1 から開始して、連続している必要があります。 ソース・エージェントのキュー・マネージャーでも使用可能にするために、クラスター全体で定義する必要があります。

ソース・エージェントと宛先エージェントの両方にキュー・マネージャー別名の番号を正しく判別させるには、キュー・マネージャーのデフォルト XMITQ を定義**しないでください**。 デフォルトの XMITQ あるいは送信キューは、**XMITQ** パラメータがどこにも定義されていない場合に使用されるキューマネージャ上のキューです。 詳細は 特定の目的のために使用されるキュー IBM MQ を参照。

関連タスク

774 ページの『複数のチャネルに対応した MFT エージェントの構成 (非クラスター)』

非クラスター構成で IBM MQ マルチチャネル・サポートを使用する場合、最初に agentMultipleChannelsEnabled プロパティーを `true` に設定してから、このトピックのステップを実行してください。

関連資料

MFT `agent.properties` ファイル

複数のチャネルに対応した MFT エージェントの構成 (非クラスター)

非クラスター構成で IBM MQ マルチチャネル・サポートを使用する場合、最初に agentMultipleChannelsEnabled プロパティーを `true` に設定してから、このトピックのステップを実行してください。

このタスクについて

非クラスター構成の場合、マルチチャネル・サポートは、ソース・エージェントと宛先エージェントの両方のキュー・マネージャーの IBM MQ 定義で使用可能にされます。

762 ページの『初めて使用する際の MFT の構成』にリストされている Managed File Transfer エージェントに必要な標準の IBM MQ 構成ステップに加えて、このトピックのステップを実行する必要があります。

以下のステップでは、送信側と受信側のチャネルがソース・キュー・マネージャーと宛先キュー・マネージャー間の通信に使用されていると想定しています。

以下の構成例では、**runmqsc** コマンドを使用します。

手順

1. 宛先エージェントのキュー・マネージャーで、使用するチャネルごとに、受信側チャネルを定義します。 例えば、2 つのチャネルを使用する場合、以下のようになります。

   ```
   DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(RCVR) TRPTYPE(TCP)
   DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(RCVR) TRPTYPE(TCP)
   ```

   ここで、DESTQMGRNAME は、宛先エージェントのキュー・マネージャーの名前です。

   チャネルには TO.DESTMGRNAME_n という命名規則を使用することが推奨されていますが、この規則は必須ではありません。 受信側チャネルの名前は、ソース・エージェントのキュー・マネージャーの対応する送信側チャネルと一致する必要があります。
2. ソース・エージェントのキュー・マネージャーで、使用するチャネルごとに、伝送キューを定義します。 例えば、2 つのチャネルを使用する場合、以下のようになります。

   ```
   DEFINE QLOCAL(DESTQMGRNAME_1) USAGE(XMITQ)
   DEFINE QLOCAL(DESTQMGRNAME_2) USAGE(XMITQ)
   ```

   伝送キューには DESTMGRNAME_n という命名規則を使用することが推奨されていますが、この規則は必須ではありません。 定義する伝送キューは、以下のステップの送信側チャネル定義およびキュー・マネージャー別名定義で参照されます。
3. ソース・エージェントのキュー・マネージャーで、使用するチャネルごとに、送信側チャネルを定義します。 例えば、2 つのチャネルを使用する場合、以下のようになります。

   ```
   DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
   XMITQ(DESTQMGRNAME_1)
   DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
   XMITQ(DESTQMGRNAME_2)
   ```

チャネルには TO.DESTMGRNAME_n という命名規則を使用することが推奨されていますが、この規則は必須ではありません。 送信側チャネルの名前は、宛先エージェントのキュー・マネージャーの対応する受信側チャネルと一致する必要があります。

4. ソース・エージェントのキュー・マネージャーで、各チャネルに対応するキュー・マネージャー別名を定義します。 以下に例を示します。

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_1)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_2)
```

キュー・マネージャー別名には SYSTEM.FTE.DESTQMGRNAME_n という命名規則を使用する必要があります。送信側エージェントがこの形式のキュー・マネージャー別名を検索するためです。 *n* に使用する番号は、1 から開始して、連続している必要があります。

エージェントにキュー・マネージャー別名の番号を正しく判別させるには、キュー・マネージャーのデフォルト XMITQ を定義**しないでください**。 デフォルトの XMITQ あるいは送信キューは、 **XMITQ** パラメータがどこにも定義されていない場合に使用されるキューマネージャ上のキューです。 詳細は 特定の目的のために使用されるキュー IBM MQ を参照。

関連タスク

773 ページの『複数のチャネルに対応した MFT エージェントの構成 (クラスター)』
クラスター構成で IBM MQ マルチチャネル・サポートを使用する場合は、まず **agentMultipleChannelsEnabled** プロパティーを `true` に設定してから、このトピックのステップを実行します。

関連資料

MFT `agent.properties` ファイル

Windows MSCS による MFT エージェントの構成

Managed File Transfer (MFT) エージェント Microsoft Cluster Service (MSCS) セットアップがサポートされるのは、プラットフォームが MFT によってサポートされ、いずれかのバージョンの Windows を実行している場合です。

このタスクについて

このタスクでは、MFT エージェントのフェイルオーバーを実現できる 2 つのシナリオについて説明します。

- シナリオ 1: エージェントを MSCS リソースとして構成する。
- シナリオ 2: エージェント・キュー・マネージャーとエージェントを MSCS リソースとして構成する。

手順

シナリオ 1: エージェントを MSCS リソースとして構成する

- エージェントを MSCS リソースとして構成するには、以下の手順を実行します。
  a) クラスター内の各マシンで Managed File Transfer をローカルにインストールします。
  Managed File Transfer のインストールを参照してください。
  b) クラスター内の 1 次マシン上にエージェントを作成します。
  このエージェントは、CLIENT トランスポートを使用してエージェント・キュー・マネージャーに接続するように構成する必要があります。 キュー・マネージャー上にこのエージェントのためのすべてのオブジェクトを作成してください。 これを行う方法については、エージェントのセットアップを参照してください。
  c) エージェントが Windows サービスとして実行されるように変更し、 Windows サービス・ツールのエージェント・サービスの 「**開始タイプ**」 フィールドを 「手動」に設定して、 Windows の再始動時に自動的に開始されないように構成します。
  詳しくは、Windows サービスとしての MFT エージェントの開始を参照してください。
  d) 二次マシン上で、シナリオ 1 のステップ 775 ページの『2』およびステップ 775 ページの『3』を繰り返します。

これにより、クラスター内のもう一方のマシンにも、ログやプロパティーなどのためのファイル構造が存在するようになります。 手順 775 ページの『2』 のキュー・マネージャー・オブジェクトの作成は行う必要がないので注意してください。

e) 1 次マシンで、エージェントを「汎用サービス」として MSCS の制御対象に追加します。

そのためには、次のようにします。

a. クラスターを右クリックし、「**役割**」->「**リソースの追加**」->「**汎用サービス**」を選択します。

b. Windows サービスのリストから、エージェント・サービスを選択し、「**次へ**」をクリックして構成ウィザードを実行します。

これで、エージェント・サービスが MSCS リソースとして追加されました。 フェイルオーバーが発生した場合は、もう一方のマシン上でエージェント・サービスが開始されます。

シナリオ 2: エージェント・キュー・マネージャーとエージェントを MSCS リソースとして構成する

- エージェント・キュー・マネージャーおよびエージェントを MSCS リソースとして構成するには、以下の手順を実行します。

a) MSCS リソースとして実行するようにエージェント・キュー・マネージャーを構成します。

この方法については、497 ページの『MSCS 制御下にキュー・マネージャーを置く』を参照してください。

b) クラスター内の 1 次マシン上にエージェントを作成します。

このエージェントは、BINDINGS トランスポートを使用してエージェント・キュー・マネージャーに接続するように構成する必要があります。 キュー・マネージャー上にこのエージェントのためのすべてのオブジェクトを作成してください。 これを行う方法については、エージェントのセットアップを参照してください。

c) エージェントが Windows サービスとして実行されるように変更し、 Windows サービス・ツールのエージェント・サービスの 「**開始タイプ**」 フィールドを 「**手動**」に設定して、 Windows の再始動時に自動的に開始されないように構成します。

詳しくは、Windows サービスとしての MFT エージェントの開始を参照してください。

d) (MSCS 制御下にある) エージェント・キュー・マネージャーが 2 次マシンで実行されていることを確認します。

このマシンに作成するエージェントは、BINDINGS トランスポートを使用してキュー・マネージャーに接続するので、エージェントの作成時にキュー・マネージャーが使用可能でなければなりません。

e) 二次マシン上で、シナリオ 2 のステップ 776 ページの『2』およびステップ 776 ページの『3』を繰り返します。

これにより、クラスター内のもう一方のマシンにも、ログやプロパティーなどのためのファイル構造が存在するようになります。 手順 776 ページの『2』 のキュー・マネージャー・オブジェクトの作成は行う必要がないので注意してください。

f) エージェントを「汎用サービス」として MSCS の制御対象に追加します。

そのためには、次のようにします。

a. クラスターを右クリックし、「**役割**」->「**リソースの追加**」->「**汎用サービス**」を選択します。

b. Windows サービスのリストから、エージェント・サービスを選択し、「**次へ**」をクリックして構成ウィザードを実行します。

g) エージェント・サービスのリソース・プロパティーを変更して、依存関係リストにキュー・マネージャー・リソースを追加します。

これにより、エージェントが開始される前にキュー・マネージャー・リソースが開始されるようになります。

h) キュー・マネージャー・リソースをオフラインにしてから、エージェント・リソースをオンラインにします。 キュー・マネージャー・リソースとエージェントの両方が開始されていることを確認してください。

フェイルオーバーが発生した場合は、エージェント・サービスとエージェント・キュー・マネージャーが 2 次マシン上で開始されます。

Managed File Transfer での高可用性エージェント

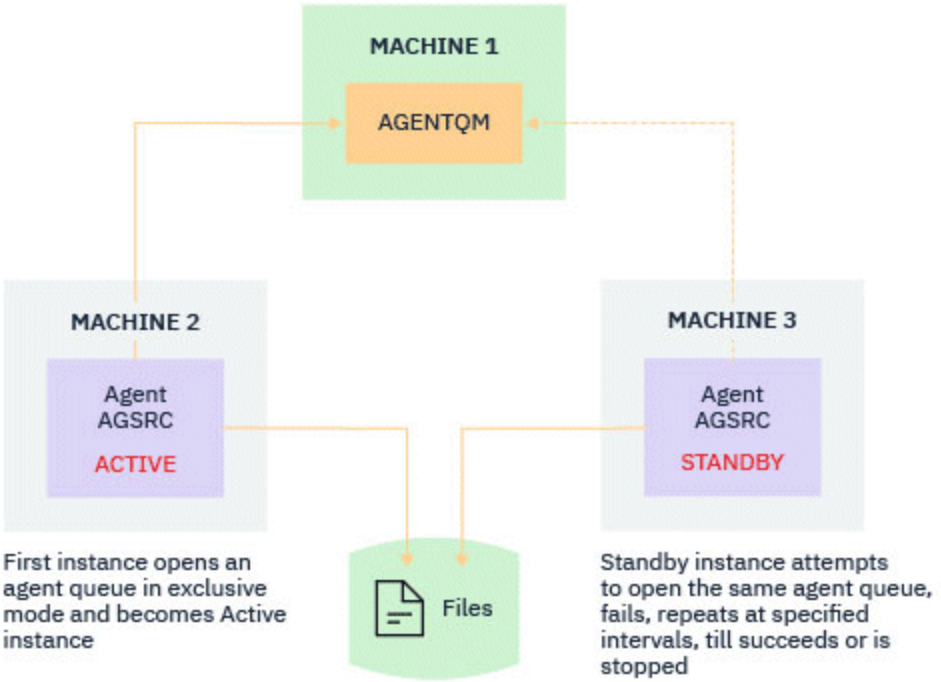
MFT の標準エージェントまたはブリッジ・エージェントを高可用性 (HA) 構成で実行するように構成できます。 HA セットアップには、同じ構成のエージェント・インスタンスのペアが必要です。 あるマシンで一方のインスタンスを実行し、別のマシンでもう一方のインスタンスを実行します。 両方のインスタンスを、同じエージェント・キュー・マネージャーに接続するように構成します。

概要

アクティブ・インスタンス と呼ばれる一方のインスタンスのみがファイル転送を処理します。スタンバイ・インスタンス と呼ばれるもう一方のインスタンスは、一部初期化状態にあり、ファイル転送を処理できません。

アクティブ・インスタンスで障害が発生するか、アクティブ・インスタンスがキュー・マネージャーへの接続を失うと、スタンバイ・インスタンスが 初期化を実行してアクティブになり、ファイル転送の処理を開始します。 アクティブ・インスタンスが最後に確認されたチェックポイントから再開されなかったときに進行中であった未完了の転送。

次の図は、アクティブ・エージェントとスタンバイ・エージェントの一般的な構成を示しています。



注 :

1. 2 台の別々のマシンでエージェントのインスタンスが 1 つずつ実行されます。それらのインスタンスの一方が アクティブ・インスタンスで、もう一方がスタンバイ・インスタンスです。
2. エージェントの各インスタンスは別々のマシンで実行されます。一方のインスタンスはアクティブ・インスタンスで、もう一方のインスタンスはスタンバイ・インスタンスです。
3. エージェントの両方のインスタンス間で、一式の同じエージェント・キューが共有されます。
4. 管理対象転送を実行するには、エージェントの両方のインスタンスが同じ共有ファイル・システムにアクセスする必要があります。

アクティブ/スタンバイのエージェント・インスタンスのメカニズムは、ある共有リソースに対するロックを取得することで機能します。 その共有リソースに対するロックを取得したエージェント・インスタンスがアクティブ・インスタンスになり、 もう一方のインスタンス (ロックを取得できなかったインスタンス) がスタンバイ・インスタンスになります。

この共有リソースとは、新しいキュー SYSTEM.FTE.HA.<agent name> のことです。 このキューは、IBM MQ 9.1.4 以降のエージェントの構成時に自動的に作成されます。

プロセスの仕組み

HA エージェントを作成するには、2 台のマシンで同じ構成パラメーターを使用してエージェントを作成します。 そのためには、agent.properties ファイル内の **highlyAvailable** エージェント・プロパティー を true に設定し、追加の **-x** パラメーターを使用して コマンド **fteCreateAgent** または **fteCreateBridgeAgent** を実行します。

注 :

- 両方の構成が同じエージェント・キュー・マネージャーを指している必要があります。
- 必要なエージェント・キューは、エージェント・キュー・マネージャーで一度だけ作成する必要があります。

-x パラメーターについて詳しくは、 **fteCreateAgent** コマンドを参照してください。 **highlyAvailable** エージェント・プロパティーについて詳しくは、 agent.properties ファイルを参照してください。

注 : fteCreateAgent コマンドまたは **fteCreateBridgeAgent** コマンドのいずれかを実行すると、エージェント・キュー・マネージャーおよび SYSTEM.FTE.HA.*agent name* キューに IBM MQ オブジェクトを作成するために必要なスクリプトを含む MQSC ファイルが作成されます。 この MQSC ファイルは、**-x** パラメーターを指定するかどうかに関係なく作成されます。

高可用性エージェント構成の作成中に、 **fteCreateAgent** コマンドまたは **fteCreateBridgeAgent** コマンドは、 SYSTEM.FTE/Agents/*agent name* トピックにサブスクライブすることによって、他の場所に存在する同じエージェントのインスタンスの存在を検査します。 同じエージェントのインスタンスが検出された場合は、どちらのコマンドでも、必要な構成がファイル・システムに作成されますが、 エージェントの作成が再発行されることはありません。

エージェントが HA モードで始動するときの動作:

1. エージェントは、排他的 GET モードで SYSTEM.FTE.HA.*agent name* キューを開こうとします。
2. エージェントが SYSTEM.FTE.HA.*agent name* キューを正常にオープンすると、これがエージェントのアクティブ・インスタンスになり、さらに起動プロセスが続行されます。
3. 排他的 GET モードで SYSTEM.FTE.HA.*agent name* キューを開こうとするが、MQRC_OBJECT_IN_USE 理由コードで失敗した場合は、他の場所で実行されているエージェントのアクティブ・インスタンスが既に存在していることを意味します。 そのため、このインスタンスはエージェントのスタンバイ・インスタンス になります。

   スタンバイ・インスタンスは、指定された間隔で SYSTEM.FTE.HA.*agent name* キューを開こうとします。 この目的のために、追加のエージェント・プロパティー **standbyPollInterval** が agent.properties ファイル内に用意されています。

   デフォルト値を使用すると、スタンバイ・インスタンスは 5 秒ごとに SYSTEM.FTE.HA.*agent name* キューを開こうとします。 これは、インスタンスが SYSTEM.FTE.HA.*agent name* キューのオープンに成功するか、 または **fteStopAgent** コマンドを使用して停止されるまで繰り返されます。

   IBM MQ 9.2.4 および IBM MQ 9.2.0 Fix Pack 5 以降、すべてのインスタンスが **standbyPollInterval** プロパティーを使用して、インスタンスがエージェント・キュー・マネージャーから切断された場合の再接続の試行間隔を決定することもできます。

複数のスタンバイ・インスタンス

すべてのスタンバイ・インスタンスは、排他的 GET モードで SYSTEM.FTE.HA.*agent name* キューを取得しようとし、アクティブ・インスタンスが失敗した後に正常に実行されたインスタンスがアクティブ・インスタンスになります。

アクティブ・インスタンスは、既知のすべてのスタンバイ・インスタンスの情報を維持し、その情報をエージェント状況のパブリケーションに含めてパブリッシュします。 **fteShowAgentDetails** コマンド、エージェントの GETREST API 応答、および IBM MQ ExplorerMFT プラグインの出力は、すべてのスタンバイ・インスタンスに関する情報を表示します。

詳しくは、**fteShowAgentDetails** コマンドの出力例、およびエージェントの GET REST API の応答を参照してください。

XML 形式のエージェント状況情報の例については、MFT エージェントの状況メッセージを参照してください。

バージョンの要件

アクティブ・エージェントとスタンバイ・エージェントは、IBM MQ 9.1.4 以上でなければなりません。

重要:

- IBM MQ 9.1.4 より前のバージョンの IBM MQ を高可用性モードで構成または開始することはできません。
- アクティブ・インスタンスとスタンバイ・インスタンスの両方で、同じバージョンのコードを実行する必要があります。

アクティブ・インスタンスとスタンバイ・インスタンスのバージョンが検証され、両方のインスタンスが同じバージョンであることが確認されます。 一時動的キューが、インスタンス間の通信に使用されます。 agent.properties ファイルに定義されている 2 つのエージェント・プロパティー **dynamicQueuePrefix** および **modelQueueName** は、一時動的キューの名前を生成します。

Managed File Transfer の高可用性エージェントを使用するために必要な情報

高可用性構成で実行されている標準またはブリッジの MFT エージェントについて知っておく必要があるさまざまなタイプの情報があります。 このような情報には、例えば、エージェントのさまざまな開始方法、ログ・ファイルでのエージェントのインスタンスの特定方法、 エージェントの状況情報などがあります。

エージェントの開始

エージェントのインスタンスが他の場所で非 HA モードで実行されている

HA エージェントとして構成されていないエージェントの別のインスタンスを開始しようとすると、まず、SYSTEM.FTE.HA.*agent name* キューでロックを取得できるかどうかがチェックされます。

他のインスタンスが非 HA モードで開始されたため、SYSTEM.FTE.HA.*agent name* キューのロックはこのインスタンスによって取得されます。 エージェントは初期化を続けますが、 コマンド・キューが別のインスタンスによって排他的に開かれているために、その後の時点で失敗します。

この場合、以下の例に示されているメッセージは、エージェントの output0.log ファイルに記録され、エージェントは 30 秒ごとにコマンド・キューをオープンし続けます:

```
BFGMQ1045I: Agent's system queue 'SYSTEM.FTE.COMMAND.SRC' is configured as either NOSHARE or
DEFSOPT (SHARED)。

BFGAG0035W: The agent received MQI reason code 2042 when trying to open queue
'SYSTEM.FTE.COMMAND.SRC' on the queue manager 'MFTHAQM' with connection name 'localhost(1414)'
and channel 'MFT_HA_CHN'.  The agent will try the operation again every 30 seconds.
```

エージェントのインスタンスが他の場所で HA モードで実行されている

HA エージェントとして構成されていないエージェントの別のインスタンスを開始しようとすると、まず、SYSTEM.FTE.HA.*agent name* キューでロックを取得できるかどうかがチェックされます。

もう一方のインスタンスはアクティブ・インスタンスとして実行されているので、ロックを取得する試行は失敗します。 インスタンスの開始に失敗し、以下のエラー・メッセージがエージェントの output0.log ファイルに記録されます:

```
BFGAG0194E: An instance of this agent is already running elsewhere.
Hence this instance cannot continue and will end.
```

Windows Windows サービスとしてのエージェントの開始

Windows では、エージェントを Windows サービスとして 開始できます。

開始時に、Windows は、MFT エージェントを通常モードまたは HA モードで開始します。 エージェントを HA モードで実行するように構成した場合、そのサービスは、どのインスタンスが先にロックを取得したかに応じて アクティブ・インスタンスまたはスタンバイ・インスタンスとして実行されます。

ログ・ファイルでのエージェントのインスタンス・タイプの特定

情報メッセージは、インスタンスのタイプを示すためにエージェントの output0.log ファイルに書き込まれます。 エージェント・インスタンスがアクティブ・インスタンスとして開始すると、次のメッセージが書き込まれます。

```
BFGAG0193I: The agent has successfully initialized as an active instance.
```

エージェント・インスタンスがスタンバイ・インスタンスとして開始すると、次のメッセージが書き込まれます。

```
BFGAG0193I: The agent has successfully initialized as a standby instance.
```

エージェント状況の更新

同じエージェントの 2 つのインスタンスが実行されるので、両方のインスタンスに関する情報を エージェント状況のパブリッシュに含める必要があります。

アクティブ・インスタンスが、両方のインスタンスの状況をパブリッシュするインスタンスであることに注意してください。

スタンバイ・インスタンス

エージェント状況のパブリッシュ中に、アクティブ・インスタンスはスタンバイ・インスタンスのパブリケーションの経過時間を検査します。

この目的のために、agent.properties ファイルには 2 つの追加プロパティーがあります:

- **standbyStatusExpiry** は、スタンバイ状況メッセージをエージェントのコマンド・キューに書き込むための有効期限です。 エージェントのアクティブ・インスタンスがこの期間内にこのメッセージを処理しない場合、 メッセージは失効します。

  デフォルトでは、**standbyStatusExipry** の値は 30 秒です。 このメッセージは、スタンバイ状況メッセージに対する転送要求の優先順位処理を可能にする、低優先順位、9 メッセージでもあります。
- **standbyStatusPublishInterval** は、スタンバイ・インスタンスがその状態をパブリッシュする頻度を設定します。

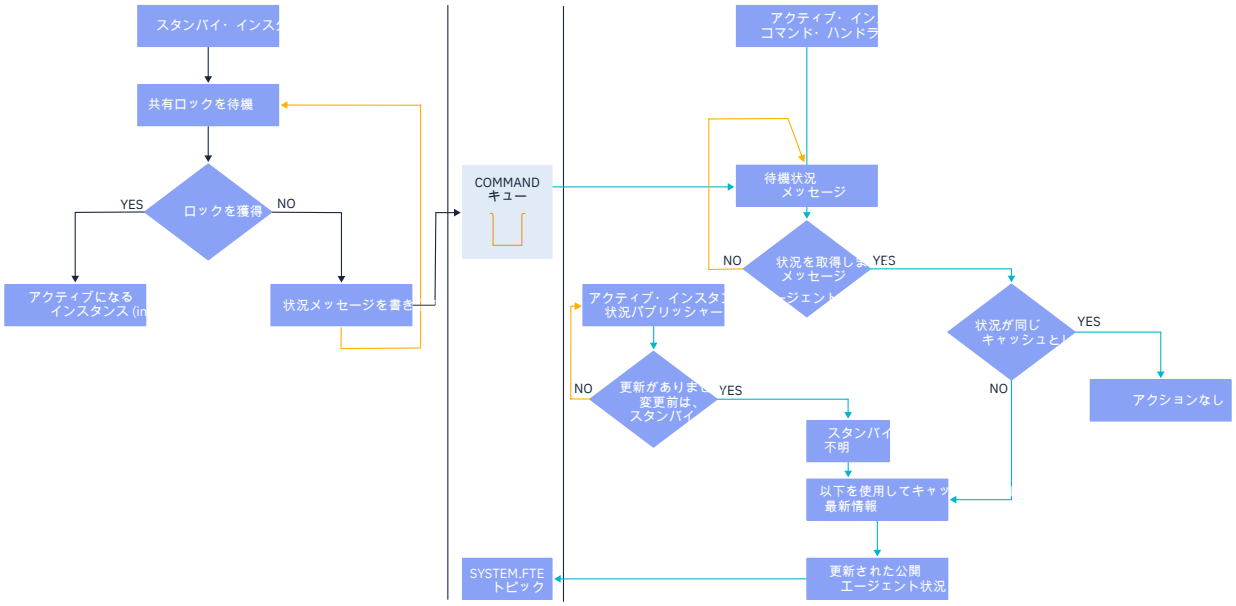
アクティブ・インスタンス

アクティブ・インスタンスは、スタンバイ・インスタンスからの状況の更新を処理するために次のことを実行します。

1. SYSTEM.FTE.COMMAND.<agent name>キューからメッセージを取得し、メッセージ処理をワーカー・スレッドに委任します。
2. ワーカー・スレッドは、メッセージ本文から内容を取得し、スタンバイ・インスタンスの情報を反映してエージェント状況オブジェクトを更新し、 エージェント状況パブリッシャーに、状況をパブリッシュするように通知します。
3. エージェント状況パブリッシャーが状況をパブリッシュします。

   ここで最適化が実行されて、スタンバイの状況情報がキャッシュに入れられることに注意してください。 要求されると、エージェント状況パブリッシャーは新しい状況をキャッシュされている状況と比較し、 相違がある場合にのみパブリッシュします。

次の図は、アクティブ・インスタンスまたはスタンバイ・インスタンスが、エージェントの状況をパブリッシュするためにたどる フローを示しています。



高可用性エージェントのインスタンスの破棄、フェイルオーバー、および保守

高可用性の Managed File Transfer インスタンスはさまざまな状況で破棄されたり、障害が発生したりする可能性があります。その場合は、保守が必要になることがあります。

スタンバイ・インスタンス状況の破棄

アクティブ・インスタンスが転送でビジー状態であるためにスタンバイ・インスタンスの状況メッセージを処理できない状況や、 スタンバイ・インスタンスが障害あるいは何らかの理由で状況メッセージをパブリッシュしていない状況などがあります。

このようなシナリオでは、スタンバイ・インスタンスの存在を認識していたアクティブ・エージェントは、`agent.properties` ファイルの **standbyStatusDiscardTime** プロパティーで指定された値だけ待機してから、スタンバイ・インスタンスをリストから削除します。 このプロパティーのデフォルト値は 600 秒で、**standbyStatusPublishInterval** プロパティーのデフォルト値の 2 倍です。

インスタンスの通常のフェイルオーバー

通常のフェイルオーバーを実行するには、 **-i** オプションを指定して **fteStopAgent** コマンドを使用する必要があります。

これにより、アクティブ・インスタンスが即時に停止されます。 エージェントを停止するときに **-i** オプションを指定しないと、 進行中のすべての転送をアクティブ・インスタンスが完了するまで、エージェントは実行し続けます。 したがって、フェイルオーバーに時間がかかる可能性があります。

未完了の転送は、既知の最後のチェックポイントから再開されます。

他の状況でのインスタンスのフェイルオーバー

アクティブ・インスタンスが正常でない方法で終了した場合、またはマシン全体で障害が発生した場合は、エージェント・キューへの接続が切断され、キュー・マネージャーは、SYSTEM.FTE.HA.<agent name> キューを含め、開かれているすべてのキューおよび接続をクローズします。

このため、スタンバイ・インスタンスは排他的 GET を獲得して、エージェント初期化の残りの処理を実行します。

この場合も、未完了の転送はすべて、既知の最後のチェックポイントから再開されます。

キュー・マネージャーへの接続が切断された場合

クライアント・モード

エージェント・プロセスは、複数のスレッドから構成されます。 デフォルトのスレッド以外に、例えばエージェント状況を一定間隔でパブリッシュするスレッドや、各転送要求を処理して転送が完了すると終了する一連のスレッドなどがあります。

これらのスレッドの多くは、エージェント・キュー・マネージャーに接続して、メッセージの書き込みと読み取りを行います。 これらの接続のいずれかが、ネットワークの問題またはキュー・マネージャーの障害が原因で切断される可能性があります。 接続の切断の問題を検出したスレッドは、メイン・スレッドにリカバリーを開始するように通知してから終了します。

次に、メイン・スレッドは別のスレッドを起動して、キュー・マネージャーへの接続が確立されるのを待ちます。 再接続されると、エージェントに対する排他的 GET を獲得しようとします。 それが成功した場合、エージェントはリカバリーを続行して完了し、アクティブ・インスタンスになります。 排他的 GET の獲得に失敗した場合、インスタンスはスタンバイになります。

バインディング・モード

バインディング・モードで接続する場合、エージェントが接続を失うと、エージェント・プロセスは終了します。 プロセス・コントローラーが、エージェントの再始動を行います。 再始動したエージェントは、そのエージェント自体に対する排他的 GET を獲得しようとします。

エージェントが成功した場合、そのエージェントはアクティブ・インスタンスになり、そうでない場合はスタンバイ・インスタンスになります。

保守レベルのアップグレードの適用

高可用性エージェントに保守を適用するための手順は、複数インスタンス・キュー・マネージャーについて説明している手順と似ています。 詳しくは、Windows 上の複数インスタンス・キュー・マネージャーへの保守レベルのアップデートの適用 、 AIX 上の複数インスタンス・キュー・マネージャーへの保守レベルのアップデートの適用、または Linux 上の複数インスタンス・キュー・マネージャーへの保守レベルのアップデートの適用を参照してください。

保守を適用する前に、保守レベルを適用するマシンで実行されているエージェントを停止する必要があります。 アクティブ・インスタンスを更新する場合、転送を継続するには、アクティブ・インスタンスをスタンバイ・インスタンスにフェイルオーバーする必要があります。

アップグレードが完了したら、エージェント・インスタンスを開始し、現行のアクティブ・インスタンスをアップグレード済みのインスタンスにフェイルオーバーしてから、 スタンバイ・インスタンスをアップグレードする必要があります。

旧バージョンの製品からのエージェントのマイグレーション

IBM MQ 9.1.4 より前のバージョンの IBM MQ からマイグレーションされたエージェントは、非高可用性として実行されます。 旧バージョンからの Managed File Transfer エージェントの移行の手順に従って、それらを高可用性モードで実行できます。

MFT ロガーの構成

Managed File Transfer は、ファイルを転送時に、調整キュー・マネージャーのトピックに対して、処置に関する情報をパブリッシュします。 データベース・ロガーは、Managed File Transfer のオプションのコンポーネントです。このコンポーネントを使用して、分析および監査のために、この情報をデータベースにコピーできます。

ロガーには以下の 3 つのバージョンがあります。

- ALW スタンドアロン・ファイル・ロガー
- スタンドアロン・データベース・ロガー
- Java Platform, Enterprise Edition (Java EE) ロガー

IBM i のロガー

IBM i

Managed File Transfer ロガーは IBM i プラットフォームではサポートされません。

スタンドアロン・ファイル・ロガー

ALW

スタンドアロン・ファイル・ロガーは、2 種類のシステム上で実行される Java プロセスである：

- コーディネーション・キュー・マネージャーをホストするシステム。
- コーディネーション・キュー・マネージャーへの接続性を持つキュー・マネージャーをホストするシステム。

スタンドアロン ファイル ロガーは、 IBM MQ バインディングを使用して、関連するロガー キュー マネージャーに接続します。 ロガーが FILE モードを使用するように設定されている場合、クライアント モードでロガー キュー マネージャに接続することもできます。 スタンドアロンロガーは、 **fteCreateLogger** コマンドを使用して作成します。 Managed File Transfer ロガーには、 `logger.properties` ファイルで指定できる設定プロパティがあります。

たとえば、以下のプロパティを設定して、スタンドアロン ロガーがロガー キュー マネージャへのクライアント接続を使用するように指定できます。

- `wmqfte.queue.manager.host`
- `wmqfte.queue.manager.port`
- `wmqfte.queue.manager.channel`

詳しくは、MFT logger.properties ファイルを参照してください。

Windows スタンドアロン ファイル ロガーを Windows サービスとして実行し、Windows セッションからログオフしてもファイル ロガーが実行され続けるようにすることができます。 詳しくは、784 ページの『MFT スタンドアロン・ファイル・ロガーのインストール』を参照してください。

以下のプラットフォーム上では、スタンドアロン・ファイル・ロガーはサポートされていません。

- z/OS z/OS
- IBM i IBM i

スタンドアロン・データベース・ロガー

スタンドアロンのデータベース・ロガーは、キュー・マネージャーおよびデータベースをホストするシステム上にインストールする、Java アプリケーションです。 スタンドアロン・データベース・ロガーは多くの場合、調整キュー・マネージャーと同じシステムにインストールされますが、調整キュー・マネージャーに接続されているキュー・マネージャーと同じシステムにインストールすることもできます。 スタンドアロン・データベース・ロガーは、IBM MQ バインディングを使用して、関連するキュー・マネージャーに接続し、タイプ 2 またはタイプ 4 の JDBC ドライバーを使用して、Db2 または Oracle データベースに接続します。 スタンドアロン・データベース・ロガーはキュー・マネージャーの XA サポートを使用して、キュー・マネージャーとデータベース両方のグローバル・トランザクションを調整してデータを保護するため、これらのタイプの接続が必要になります。

Windows Windows システムを使用している場合、スタンドアロン・ロガーを Windows サービスとして実行でき、Windows セッションからログオフするときにもロガーを継続して実行するようにできます。 詳しくは、スタンドアロン・データベース・ロガーのトピック 800 ページの『MFT スタンドアロン・データベース・ロガーのインストール』を参照してください。

Java EE データベース・ロガー

Java EE データベース・ロガーは、アプリケーション・サーバーにインストールするための EAR ファイルとして用意されています。 データベースロガーは他のエンタープライズアプリケーションと一緒に管理で

きるため、既存の Java EE アプリケーションサーバー環境を利用できる場合は、スタンドアロンの Java EE データベースロガーを使用するよりも便利です。 IBM MQ サーバーおよびデータベースをホストするシステムとは別のシステムに Java EE データベース・ロガーをインストールすることもできます。 Java EE データベース・ロガーは Db2 および Oracle データベースでの使用がサポートされています。 Java EE データベース・ロガーは、 WebSphere Application Server 7.0 にインストールされている場合、 Oracle Real Application Clusters もサポートします。

ロガーの構成方法については、以下のトピックを参照してください。

- 784 ページの『MFT スタンドアロン・ファイル・ロガーのインストール』
- 800 ページの『MFT スタンドアロン・データベース・ロガーのインストール』
- 805 ページの『MFT 用の Java EE データベース・ロガーのインストール』

関連タスク

802 ページの『リモート・データベースでの MFT の使用』
Managed File Transfer のロガーを使用して、リモート・システムのデータベースと通信できます。

関連資料

MFT ロガーのエラー処理とメッセージ拒否

MFT ロガーの構成プロパティー

ALW MFT スタンドアロン・ファイル・ロガーのインストール

スタンドアロン・ファイル・ロガーは、 IBM MQ バインディング・モードまたはクライアント・モードを使用して調整キュー・マネージャーに接続する必要がある Java プロセスです。 スタンドアロン・ファイル・ロガーを定義するには、**fteCreateLogger** コマンドを使用し、このトピックの手順に従ってください。

このタスクについて

注 : MFT スタンドアロンファイルロガーは、再配布可能な「MFT エージェントインストールイメージ上では構成および実行できません。

スタンドアロン・ファイル・ロガーについて詳しくは、782 ページの『MFT ロガーの構成』を参照してください。 このトピックの手順では、調整キュー・マネージャーに接続するようにロガーを構成します。 ロガーの代替構成については、804 ページの『MFT スタンドアロン・ロガーの代替構成』 を参照してください。

以下のプラットフォーム上では、スタンドアロン・ファイル・ロガーはサポートされていません。

- z/OS z/OS
- IBM i IBM i

手順

1. Managed File Transfer Logger コンポーネントがインストールされていることを確認します。 詳しくは、Managed File Transfer 製品のオプションを参照してください。
2. 調整キュー・マネージャーを指定し -loggerType パラメーターを FILE に設定して **fteCreateLogger** コマンドを実行し、スタンドアロン・ファイル・ロガーを作成します。 詳しくは、fteCreateLogger を参照してください。
3. オプション: カスタム・フォーマットを使用する場合は、**fteCreateLogger** コマンドで作成された XML ファイルを変更できます。 ログ・フォーマット定義は `FileLoggerFormat.xml` ファイルに配置されます。 詳しくは、785 ページの『MFT スタンドアロン・ファイル・ロガー・フォーマット』を参照してください。
4. **fteCreateLogger** コマンドによって提供される MQSC コマンドを調整キュー・マネージャーに対して実行して、ロガー・キューを作成します。

5. ロガー・プロセスを実行するユーザーを識別し、そのユーザーの権限を構成します。 詳細については、799 ページの『MFT スタンドアロン・ファイル・ロガーのユーザー・アクセス権限の構成』を参照してください。
6. オプション: **fteCreateLogger** コマンドを実行したときに作成される logger.properties ファイルを編集すると、スタンドアロン・ファイル・ロガーをさらに構成できます。 このファイルは、キーと値のペアで構成されている Java プロパティー・ファイルです。 logger.properties ファイルは、*MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* ディレクトリーにあります。 使用可能なプロパティーとそれらの効果の詳細については、MFT ロガーの構成プロパティーを参照してください。
7. Windows
   オプション: Windows システムを使用している場合には、スタンドアロン・ファイル・ロガーを Windows サービスとして実行できます。 **-s** パラメーターを指定して、**fteModifyLogger** コマンドを実行します。 詳しくは、fteModifyLogger を参照してください。
8. **fteStartLogger** コマンドを使用してスタンドアロン・ファイル・ロガーを開始します。 詳しくは、fteStartLogger を参照してください。

   前のステップを実行し、**-s** パラメーターを指定した **fteModifyLogger** コマンドを Windows で使用した場合、スタンドアロン・ファイル・ロガーは Windows サービスとして開始されます。
9. ロガー出力を確認します。 スタンドアロン・ファイル・ロガーによって 2 タイプの出力 (ファイル転送監査データおよびロガー診断データ) が生成されます。 ファイル転送監査データは、*MQ_DATA_PATH*/mqft/logs/*coordination_qmgr_name*/loggers/*logger_name*/logs で見つけることができます。 ロガー診断データは、*MQ_DATA_PATH*/mqft/logs/*coordination_qmgr_name*/loggers/*logger_name* で見つけることができます
10. **fteStopLogger** コマンドを使用すると、ロガーを停止できます。 詳しくは、fteStopLogger を参照してください。

タスクの結果

関連タスク

799 ページの『MFT スタンドアロン・ファイル・ロガーのユーザー・アクセス権限の構成』
テスト環境では、通常のユーザー・アカウントに必要な、任意の特権を新たに追加できます。 実稼働環境では、ジョブを実行するために必要な最小限の許可を備えた新しいユーザーを作成することをお勧めします。

関連資料

MFT ロガーの構成プロパティー

fteStartLogger

fteCreateLogger

fteModifyLogger

fteStopLogger

785 ページの『MFT スタンドアロン・ファイル・ロガー・フォーマット』
ファイル・ロガーによって、書き込まれるメッセージ情報のフォーマットは、FileLoggerFormat.xml ファイルで定義することができます。

MFT ロガーの権限

ALW *MFT* スタンドアロン・ファイル・ロガー・フォーマット

ファイル・ロガーによって、書き込まれるメッセージ情報のフォーマットは、FileLoggerFormat.xml ファイルで定義することができます。

ロガーの構成ディレクトリーは、*MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* にあります。 新規ファイル・ロガーを作成すると、そのファイル・ロガーによって使用されるデフォルトの定義セットが含まれたバージョンのファイルが作成されます。 デフォルトのログ・フォーマット定義について詳しくは、MFT スタンドアロン・ファイル・ロガーのデフォルトのログ・フォーマットを参照してください。

独自のカスタムログフォーマットを指定する場合は、FileLoggerFormat.xml ファイルを編集します。

カスタム・ログ・フォーマット定義

ログ・フォーマット定義には、メッセージ・タイプのセットと各メッセージ・タイプのフォーマット定義が含まれています。 メッセージ・タイプのフォーマット定義は、XPATH フォーマットで指定されている挿入項目と各挿入項目を分離する分離文字で構成されます。 挿入項目の配列によって、ログ・ファイルに出力される行中の内容の配列が決まります。 例えば、callStarted メッセージ・タイプの定義は次のようになります。

```
<callStarted>
  <format>
    <inserts>
      <insert type="user"   width="19" ignoreNull="false">/transaction/action/
       @time</insert>
      <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6"  ignoreNull="false">type</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/agent/
       @agent</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/agent/@QMgr</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/job/name</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/transferSet/
       call/command/@type</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/transferSet/
      call/command/@name</insert>
      <insert type="system" width="0"  ignoreNull="true">callArguments</insert>
    </inserts>
    <separator></separator>
  </format>
</callStarted>
```

このフォーマットでは、ログ・ファイルの行が次のように作成されます。

```
2011-11-25T10:53:04;414d5120514d5f67627468696e6b20206466cf4e20004f02;[CSTR];
  AGENT1;AGENT_QM;Managed Call;executable;echo;call test;
```

フォーマット定義では、各情報がログ・ファイルの行中に出現する順序で挿入項目が指定されます。 FileLoggerFormat.xml ファイルの形式を定義する XML スキーマについての詳細は、スタンドアロン・ファイル・ロガー XSD を参照してください。

メッセージ・タイプ

FTE エージェントは、複数の異なるメッセージ・タイプを SYSTEM.FTE/Log サブトピックに書き込みます。 詳しくは、SYSTEM.FTE トピックを参照してください。 ログ・ファイル定義には、次のようなメッセージ・タイプのフォーマット定義を含めることができます。

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
scheduleSubmitTransferSet
transferStarted
transferCancelled
transferComplete
transferDelete
transferProgress
```

MFT ファイルロガーが各メッセージタイプに対して記録できる属性の詳細については、'789 ページの『MFT ファイルロガーが各メッセージタイプに対して記録できる属性』を参照のこと。 属性は、XPATH 挿入として使用し、'MFT ファイル・ロガーの出力をカスタマイズすることができる。

メッセージのフォーマットはそれぞれ異なるものにすることができます。 大部分のメッセージ・タイプでは、SYSTEM.FTE/Log サブトピックからコンシュームされるログ・メッセージごとに 1 行がログ・ファイルに書き込まれます。 この場合、ログ・フォーマット定義内の XPATH アドレスは、メッセージのルートからの相対アドレスで指定する単純なものになります。 この方式で出力が書き込まれるのは、次のメッセージ・タイプの場合です。

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
transferStarted
transferCancelled
transferComplete
transferDelete
```

ログ・メッセージの書き込みに使用される他の方式では、ログ・メッセージ内の 1 つの転送セットの項目を表すのに、複数の行を使用します。 この場合、指定されるフォーマットは、ログ・メッセージ内の転送セットの各項目に適用されます。 転送セット内の各項目に固有の情報を組み込むには、その項目を XPATH のルートとして XPATH を指定する必要があります。 この方式で出力が書き込まれるのは、次のメッセージ・タイプの場合です。

```
scheduleSubmitTransferSet
transferProgress
```

転送セットの項目ごとに出力が 1 行書き込まれます。 転送セット内のすべての項目で固定にする情報については、ログ・メッセージのルートからの相対 XPATH アドレスを使用することができます。 次の transferProgress フォーマット定義の簡単な例では、タイム・スタンプと転送 ID が固定されます。 項目をルートとして相対で指定される情報の内容は、書き込まれる行によって異なります。 この例では、項目ごとにソースと宛先のファイル情報が書き込まれます。

```
<transferProgress>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/transaction/action/
           @time</insert>
          <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="true">status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">source/file |
           source/queue</insert>
          <insert type="user"   width="0"  ignoreNull="false">source/file/@size |
           source/queue/@size</insert>
          <insert type="user"   width="5"  ignoreNull="true">source/@type</insert>
          <insert type="user"   width="6"  ignoreNull="true">source/@disposition</insert>
          <insert type="user"   width="0"  ignoreNull="false">destination/file |
            destination/queue</insert>
          <insert type="user"   width="0"  ignoreNull="false">destination/file/@size |
            destination/queue/@size</insert>
          <insert type="user"   width="5"  ignoreNull="true">destination/@type</insert>
          <insert type="user"   width="9"  ignoreNull="true">destination/@exist</insert>
          <insert type="user"   width="0"  ignoreNull="true">status/supplement</insert>
        </inserts>
        <separator></separator>
      </format>
    </transferProgress>
```

この例では、次のように、このフォーマットで 1 行以上のログ・ファイル・エントリーが作成されます。

```
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPRO];0
  ;/src/test1.file;3575;file;leave ;/dest/test1.file;3575;file;overwrite;;
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPRO];0
  ;/src/test2.file;3575;file;leave ;/dest/test2.file;3575;file;overwrite;;
```

挿入のフォーマット

メッセージ・タイプのフォーマットを定義する際に使用可能な挿入タイプは、user と system の 2 つがあります。 挿入タイプは、insert エレメントの type 属性で定義します。 どちらの挿入タイプも、insert エレメントの **width** 属性および **ignoreNull** 属性を使用してレイアウトをカスタマイズすることができます。 以下に例を示します。

```
<insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
```

この例では、挿入はログメッセージで/transaction/@ID 見つかった情報を取り、ログに書き込む前に 48 文字にトリミングまたはパッドします。 /transaction/@ID の内容がヌルの場合、ignoreNull 属性が false に設定されているため、48 文字になるように埋め込みが行われてからヌル・ストリングが書き込まれます。 ignoreNull が true に設定されている場合には、48 文字になるように埋め込まれた空ストリングが書き込まれます。 width="0"を設定すると、列の幅がトリムされないことを意味します。幅が 0 にトリムされることではない。 この方法で ignoreNull 属性を使用して、予期しないときに、NULL が検出された場合にログを検出することができます。 これは、新しいログ・ファイル定義のデバッグを行う際に役立つ場合があります。

ユーザー定義の挿入

ユーザー定義の挿入には、その挿入で書き込まれる情報の XPATH アドレスが含まれます。 このアドレスは、FTE ログ・メッセージ内の情報の一部を参照します。 ログ・メッセージのフォーマットについて詳しくは、以下を参照してください。

- ファイル転送ログ・メッセージ・フォーマット
- スケジュール済みファイル転送ログ・メッセージ・フォーマット
- MFT モニター・ログ・メッセージ・フォーマット

システム定義の挿入

システム定義の挿入には、ログ・メッセージ内にない情報、または XPATH 言語では定義するのが容易でない情報のいずれかの情報の一部を参照するキーワードが含まれます。

システム定義の挿入では、以下のものがサポートされています。

- type - ログ・メッセージのタイプを、短形式で書き込みます。
- callArguments - 管理対象呼び出しに提供する引数のセットを、スペースで分離したフォーマットで書き込みます。
- transferMetaData - 転送に定義されたメタデータ・エントリーのセットを、*key=value* をコンマで区切った形式で記述します。

次の表では、メッセージ・タイプごとに、システム定義の挿入の「type」値をリストします。

表 *50.* サポートされるメッセージ・タイプとそれらの「*type*」システム挿入の要約

メッセージ・タイプ	「**type**」**システム挿入の値**
callCompleted	[CCOM]
callStarted	[CSTR]
monitorAction	[MACT]
monitorCreate	[MCRT]
monitorFired	[MFIR]
notAuthorized	[AUTH]

表 *50.* サポートされるメッセージ・タイプとそれらの「*type*」システム挿入の要約 *(*続き*)*	
メッセージ・タイプ	**「type」システム挿入の値**
scheduleDelete	[SDEL]
scheduleExpire	[SEXP]
scheduleSkipped	[SSKP]
scheduleSubmitInfo	[SSIN]
scheduleSubmitTransfer	[SSTR]
scheduleSubmitTransferSet	[SSTS]
transferStarted	[TSTR]
transferCancelled	[TCAN]
transferComplete	[TCOM]
transferDelete	[TDEL]
transferProgress	[TPRO]

関連資料
MFT スタンドアロン・ファイル・ロガーのデフォルトのログ・フォーマット
スタンドアロン・ファイル・ロガーのフォーマット XSD
SYSTEM.FTE トピック
ファイル転送ログ・メッセージ・フォーマット
スケジュール済みファイル転送ログ・メッセージ・フォーマット
MFT モニター・ログ・メッセージ・フォーマット

MFT ファイルロガーが各メッセージタイプに対して記録できる属性
Managed File Transfer(MFT) ファイルロガーが各メッセージタイプに対してログを記録できる属性を XPATH 挿入として使用し、'MFT ファイルロガーの出力をカスタマイズすることができます。

メッセージの種類:callCompleted

表 *51. callCompleted* の属性	
属性名	**説明**
/transaction/action/@time	このイベントが発生した日時。
/transaction/@ID	マネージドコールの識別子。
/transaction/status/@resultCode	マネージドコールの結果。
タイプ	短いフォーマットでのログメッセージのタイプ。 このメッセージ・タイプでは、値は CCOM である。
/transaction/agent/@agent	管理対象コールの送信先エージェント。
/transaction/agent/@QMgr	エージェント・キュー・マネージャー。
/トランザクション/ジョブ/名前	ジョブ名

表 51. *callCompleted* の属性 (続き)	
属性名	**説明**
/transaction/transferSet/call/command/@type	コマンドタイプ（実行可能、anscript、jcl）。
/transaction/transferSet/call/command/@name	マネージドコールで実行されるコマンド名。
callArguments	コマンドに渡される引数。
/transaction/transferSet/call/callResult/@outcome	コマンド実行の結果（成功、失敗、エラー）。
/transaction/transferSet/call/callResult/result/error	エラーの詳細情報。

メッセージタイプ：callStarted

表 52. *callStarted* の属性	
属性名	**説明**
/transaction/action/@time	このイベントが発生した日時。
/transaction/@ID	マネージドコールの識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。このメッセージ・タイプでは、値は CSTR である。
/transaction/agent/@agent	管理対象コールの送信先エージェント。
/transaction/agent/@QMgr	エージェント・キュー・マネージャー。
/トランザクション/ジョブ/名前	ジョブ名。
/transaction/transferSet/call/command/@type	コマンドタイプ（実行可能、anscript、jcl）。
/transaction/transferSet/call/command/@name	実行するコマンドの名前。
callArguments	コマンドに渡される引数のリスト。

メッセージタイプ:monitorAction

表 53. *monitorAction* の属性	
属性名	**説明**
/monitorLog/action/@time	このイベントが発生した日時。
/monitorLog/@referenceId	リソースモニタの識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。このメッセージ・タイプでは、値は MACT である。
/monitorLog/status/@resultCode	リソースモニターのアクションの結果。
/monitorLog/@monitorName	リソースモニターの名前。
/monitorLog/monitorAgent/@agent	リソースモニターが実行されているエージェントの名前。
/monitorLog/monitorAgent/@QMgr	エージェント・キュー・マネージャー。
/monitorLog/action	リソースモニタアクションのタイプ - create、delete、start、stop、triggerSatisfied, triggerNotSatisfied, triggerFail。

メッセージタイプ:monitorCreate

表 *54. monitorCreate* の属性	
属性名	**説明**
/monitorLog/action/@time	このイベントが発生した日時。
/monitorLog/@referenceId	リソースモニタの識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。 このメッセージ・タイプでは、値は MCRT である。
/monitorLog/@monitorName	リソースモニターの名前。
/monitorLog/monitorAgent/@agent	リソースモニターが作成されるエージェントの名前。
/monitorLog/monitorAgent/@QMgr	エージェント・キュー・マネージャの名前。
/monitorLog/action	リソースモニター作成アクション。

メッセージの種類:monitorFired

表 *55. monitorFired* の属性	
属性名	**説明**
/monitorLog/action/@time	このイベントが発生した日時。
/monitorLog/@referenceId	リソースモニタの識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。 このメッセージ・タイプでは、値は MFIR である。
/monitorLog/status/@resultCode	リソースモニターが起動したアクションの結果。
/monitorLog/@monitorName	リソースモニターの名前。
/monitorLog/monitorAgent/@agent	リソースモニターが実行されているエージェントの名前。
/monitorLog/monitorAgent/@QMgr	エージェント・キュー・マネージャの名前。
/monitorLog/action	リソースモニターが起動した。
/monitorLog/references/taskRequest	このリソースモニターが提出した転送タスク要求の詳細。

メッセージタイプ：notAuthorized

表 *56. notAuthorized* の属性	
属性名	**説明**
/notAuthorized/action/@time	このイベントが発生した日時。
/notAuthorized/@ID	イベントの識別子
タイプ	短いフォーマットでのログメッセージのタイプ。 このメッセージタイプでは、値は AUTH である。
/notAuthorized/status/@resultCode	イベントで提供される結果コード。
/notAuthorized/action	許可がありません。
/notAuthorized/authority	アクションを実行するために必要な権限。

表 56. *notAuthorized* の属性 (続き)	
属性名	**説明**
/notAuthorized/originator/userID	リクエストを送信したユーザの ID。
/notAuthorized/originator/mqmdUserID	MQ ユーザー ID。
/notAuthorized/status/supplement	非公認イベントに関する追加情報。

メッセージタイプ:scheduleDelete

表 57. *scheduleDelete* の属性	
属性名	**説明**
/schedulelog/action/@time	このイベントが発生した日時。
/schedulelog/@ID	スケジュールされた転送の識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。このメッセージ・タイプでは、値は SDEL である。
/schedulelog/status/@resultCode	スケジュールされた転送削除アクションの結果。
/schedulelog/sourceAgent/@agent	アクションが提出されたエージェントの名前。
/スケジュールログ/アクション	削除する
/schedulelog/originator/userID	スケジュールされた転送の削除要求を提出したユーザー。
/スケジュールログ/ステータス/補足	予定されていた移籍削除リクエストの結果に関する追加情報

メッセージタイプ:scheduleExpire

表 58. *scheduleExpire* の属性	
属性名	**説明**
/schedulelog/action/@time	このイベントが発生した日時。
/schedulelog/@ID	スケジュールされた転送の識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。このメッセージ・タイプでは、値は SEXP である。
/schedulelog/status/@resultCode	予定されていた期限切れ移籍アクションの結果コード。
/schedulelog/sourceAgent/@agent	アクションが提出されたエージェントの名前。
/スケジュールログ/アクション	期限切れ
/schedulelog/originator/userID	スケジュールされた転送を作成したユーザー ID。
/スケジュールログ/ステータス/補足	予定されていた移籍期限切れ要請の結果に関する追加情報。

メッセージの種類:scheduleSkipped

表 *59. scheduleSkipped* の属性	
属性名	**説明**
/schedulelog/action/@time	このイベントが発生した日時。
/schedulelog/@ID	スケジュールされた転送の識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。 このメッセージ・タイプでは、値は SSKP である。
/schedulelog/status/@resultCode	予定されていた移籍スキップの結果。
/schedulelog/sourceAgent/@agent	アクションが提出されたエージェントの名前。
/スケジュールログ/アクション	スキップされたスケジュール
/schedulelog/originator/userID	スケジュールされた転送を作成したユーザー ID。
/スケジュールログ/ステータス/補足	スケジュールされた転送スキップ要求の結果に関する追加情報。

メッセージの種類:scheduleSubmitInfo

表 *60. scheduleSubmitInfo* の属性	
属性名	**説明**
/schedulelog/action/@time	このイベントが発生した日時。
/schedulelog/@ID	スケジュールされた転送の識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。 このメッセージ・タイプでは、値は SSIN である。
/schedulelog/status/@resultCode	予定されていた移籍の結果。
/schedulelog/sourceAgent/@agent	アクションが提出されたエージェントの名前。
/スケジュールログ/アクション	移籍リクエストを送信する。
/schedulelog/originator/userID	スケジュールされた転送を作成したユーザー ID。
/schedulelog/schedule/submit/@timezone	転送要求の送信に使用されるタイムゾーン。
/schedulelog/schedule/repeat/frequency/ @interval	スケジュールされた転送が発生する間隔。
/schedulelog/schedule/repeat/expireCount	スケジュールされた転送の回数。
/スケジュールログ/ステータス/補足	スケジュールされた転送スキップ要求の結果に関する追加情報。

メッセージタイプ:scheduleSubmitTransfer

表 *61. scheduleSubmitTransfer* の属性	
属性名	**説明**
/schedulelog/action/@time	このイベントが発生した日時。
/schedulelog/@ID	スケジュールされた転送の識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。 このメッセージ・タイプでは、値は SSTR である。

表 61. *scheduleSubmitTransfer* の属性 (続き)	
属性名	**説明**
/schedulelog/sourceAgent/@agent	転送要求の送信元エージェント名。
/schedulelog/sourceAgent/@QMgr	ソースエージェントのキューマネージャの名前。
/schedulelog/destinationAgent/@agent	転送要求の宛先エージェント名。
/schedulelog/destinationAgent/@QMgr	宛先エージェントキューマネージャの名前。

メッセージタイプ:scheduleSubmitTransferSet

表 62. *scheduleSubmitTransferSet* の属性	
属性名	**説明**
/schedulelog/action/@time	このイベントが発生した日時。
/schedulelog/@ID	スケジュールされた転送の識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。 このメッセージ・タイプでは、値は SSTS である。
ソース/ファイル	転送要求のソースファイル名。
ソース/キュー	転送要求のキュー名。 ソースタイプはキューである。
source/@type	ソース・タイプ
source/@disposition	転送完了後、ソースファイルを削除または保持する。
宛先/ファイル	保存先ファイル名。
宛先/キュー	宛先キューの名前。 デスティネーションタイプがキューである。
destination/@type	宛先タイプ
destination/@exist	宛先ファイルが既に存在する場合に取るべきアクション。 上書きまたはエラー。

メッセージの種類:transferStarted

表 63. *transferStarted* の属性	
属性名	**説明**
/transaction/action/@time	このイベントが発生した日時。
/transaction/@ID	転送の識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。 このメッセージ・タイプでは、値は TSTR である。
/transaction/status/@resultCode	結果コード
/transaction/sourceAgent/@agent	ソースエージェントの名前。
/transaction/sourceAgent/@QMgr	ソースエージェントのキューマネージャの名前。
/transaction/sourceAgent/@agentType	エージェントのタイプ - スタンダードまたはブリッジ。
/transaction/destinationAgent/@agent	デスティネーション・エージェントの名前。

表 63. *transferStarted* の属性 (続き)	
属性名	**説明**
/transaction/destinationAgent/@QMgr	宛先エージェントキューマネージャの名前。
/transaction/originator/userID	転送要求を送信したユーザー ID。
/トランザクション/ジョブ/名前	ジョブ名
/transaction/scheduleLog/@ID	この転送要求のトリガーとなったスケジュール転送の識別子。
transferMetaData	転送要求に関連するメタデータ。

メッセージの種類:transferCancelled

表 64. *transferCancelled* の属性	
属性名	**説明**
/transaction/action/@time	このイベントが発生した日時。
/transaction/@ID	転送の識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。このメッセージ・タイプでは、値は TCAN である。
/transaction/status/@resultCode	結果コード
/transaction/sourceAgent/@agent	ソースエージェントの名前。
/transaction/sourceAgent/@QMgr	ソースエージェントのキューマネージャの名前。
/transaction/sourceAgent/@agentType	エージェントのタイプ - スタンダードまたはブリッジ。
/transaction/destinationAgent/@agent	デスティネーション・エージェントの名前。
/transaction/destinationAgent/@QMgr	宛先エージェントキューマネージャの名前。
/transaction/destinationAgent/@agentType	宛先エージェントのタイプ - 標準またはブリッジ。
/transaction/originator/userID	転送要求を送信したユーザー ID。
/トランザクション/ジョブ/名前	ジョブ名。
/トランザクション/ステータス/補足	イベントに関する追加情報

メッセージの種類:transferComplete

表 65. *transferComplete* の属性	
属性名	**説明**
/transaction/action/@time	このイベントが発生した日時。
/transaction/@ID	転送の識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。このメッセージ・タイプでは、値は TCOM である。
/transaction/status/@resultCode	結果コード
/transaction/sourceAgent/@agent	ソースエージェントの名前。
/transaction/sourceAgent/@QMgr	ソースエージェントのキューマネージャの名前。

表 65. *transferComplete* の属性 (続き)	
属性名	**説明**
/transaction/sourceAgent/@agentType	ソースエージェントのタイプ。
/transaction/destinationAgent/@agent	デスティネーション・エージェントの名前。
/transaction/destinationAgent/@QMgr	宛先エージェントキューマネージャの名前。
/transaction/destinationAgent/@agentType	デスティネーション・エージェントのタイプ。
/transaction/originator/userID	転送要求を送信したユーザー ID。
/トランザクション/ジョブ/名前	ジョブ名。
/トランザクション/ステータス/補足	移籍完了に関する追加情報。
transferMetaData	転送要求に関連するメタデータ。

メッセージタイプ:transferDelete

表 66. *transferDelete* の属性	
属性名	**説明**
/transaction/action/@time	このイベントが発生した日時。
/transaction/@ID	転送の識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。 このメッセージ・タイプでは、値は TDEL である。
/transaction/status/@resultCode	結果コード
/transaction/sourceAgent/@agent	ソースエージェントの名前。
/transaction/sourceAgent/@QMgr	ソースエージェントのキューマネージャの名前。
/transaction/sourceAgent/@agentType	ソースエージェントのタイプ - スタンダードまたはブリッジ。
/transaction/destinationAgent/@agent	デスティネーション・エージェントの名前。
/transaction/destinationAgent/@QMgr	宛先エージェントキューマネージャの名前。
/transaction/destinationAgent/@agentType	宛先エージェントのタイプ - 標準またはブリッジ。
/transaction/originator/userID	転送要求を送信したユーザー ID。
/トランザクション/ジョブ/名前	ジョブ名。
/トランザクション/ステータス/補足	イベントの追加情報

メッセージタイプ:transferProgress

表 67. *transferProgress* の属性	
属性名	**説明**
/transaction/action/@time	このイベントが発生した日時。
/transaction/@ID	転送の識別子。
タイプ	短いフォーマットでのログメッセージのタイプ。 このメッセージ・タイプでは、値は TPRO である。
status/@resultCode	このアイテムの移籍の結果コード。

表 67. *transferProgress* の属性 (続き)

属性名	説明
ソース/ファイル	ソースファイルの名前。
ソース/キュー	ソースキューの名前。
source/file/@size	ソースファイルのサイズ。
source/queue/@size	メッセージのサイズ。 常に "null "とする。
source/@type	ソース・タイプ
source/@disposition	転送完了後に実行するアクション - ソース・ファイルを削除または残す。
source/file/@alias	Connect:Direct ブリッジを使用して'Connect:Direct ノードから'Managed File Transfer エージェントに転送する場合の'Connect:Direct 転送名。
source/queue/@alias	ソース・タイプがキューである場合のエイリアス・キュー名。
source/@correlationBoolean1	Connect:Direct ブリッジを使用した「Connect:Direct ノードから「Managed File Transfer エージェントへの転送が、ユーザー定義の Connect:Direct プロセスを呼び出すかどうかを示すフラグ。
source/@correlationNum1	Connect:Direct ブリッジを使用して'Connect:Direct ノードから'Managed File Transfer エージェントに転送する場合の'Connect:Direct プロセス識別子。
source/@correlationString1	Connect:Direct ブリッジを使用して 'Connect:Direct ノードから 'Managed File Transfer エージェントに転送する場合の 'Connect:Direct プロセスの名前。
宛先/ファイル	保存先ファイル名。
宛先/キュー	宛先キューの名前。
destination/file/@size	保存先のファイルサイズ
destination/queue/@size	宛先メッセージサイズ - 常に "null "として設定される。
destination/@type	宛先タイプ。
destination/@exist	宛先が存在する場合に実行するアクション - オーバーライトまたはエラー。
destination/file/@alias	Connect:Direct ブリッジを使用して'Managed File Transfer エージェントから'Connect:Direct ノードに転送する場合の'Connect:Direct 転送名。
destination/queue/@alias	宛先の別名キュー名。 宛先タイプがキューの場合のみ有効。
destination/file/@truncateRecords	LRECL データセット属性より長い宛先レコードを切り捨てる。 宛先タイプがデータセットの場合のみ有効。

表 67. *transferProgress* の属性 (続き)	
属性名	**説明**
destination/@correlationBoolean1	Connect:Direct ブリッジを使用した 'Managed File Transfer エージェントから 'Connect:Direct ノードへの転送が、ユーザー定義の 'Connect:Direct プロセスを呼び出すかどうかを示すフラグ。
destination/@correlationNum1	Connect:Direct ブリッジを使用して 'Managed File Transfer エージェントから 'Connect:Direct ノードへ転送する場合の 'Connect:Direct プロセス識別子。
destination/@correlationString1	Connect:Direct ブリッジを使用して 'Managed File Transfer エージェントから 'Connect:Direct ノードに転送する場合の 'Connect:Direct プロセスの名前。
ステータス/補足	イベントに関する追加情報

ALW *MFT* スタンドアロン・ファイル・ロガーからのメッセージ・タイプの除外

特定のメッセージ・タイプをファイル・ロガーの出力から除外するために、空のメッセージ・タイプ・エレメントを使用できます。

例

例えば、次のフォーマット定義では、ファイル・ロガーから `transferProgress` メッセージが出力されないようにします。

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
 xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferProgress></transferProgress>
  </messageTypes>
</logFormatDefinition>
```

ALW *MFT* スタンドアロン・ファイル・ロガーのカスタム・フォーマットの定義

ログ・フォーマット定義内でカスタム・メッセージ・タイプのサブセットを定義して、ログ・ファイル・フォーマットのカスタマイズに必要な構成を減らすことができます。

このタスクについて

`FileLoggerFormat.xml` ファイルに `messageTypes` エレメントが含まれていない場合、そのメッセージ・タイプのフォーマットにはデフォルトのフォーマットが使用されます。 デフォルトと異なる場合にのみフォーマットを指定してください。

例

この例では、フォーマット定義によって、`transferStarted` メッセージ・タイプのデフォルトのフォーマットが、転送を開始したユーザーのみを出力するこの削減されたバージョンに置き換えられます。 他のすべてのメッセージ・タイプは、このログ・フォーマット定義には含まれていないため、デフォルトのフォーマットを使用します。

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
 xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferStarted>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/transaction/action/
           @time</insert>
          <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
```

```
          <insert type="user"   width="0"  ignoreNull="true">/transaction/originator/
           userID</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </transferStarted>
  </messageTypes>
</logFormatDefinition>
```

関連資料
MFT スタンドアロン・ファイル・ロガーのデフォルトのログ・フォーマット
スタンドアロン・ファイル・ロガーのフォーマット XSD

ALW *MFT* スタンドアロン・ファイル・ロガーの重複メッセージの削減
スタンドアロン・ファイル・ロガーのログでは、重複ログ・メッセージが発生する場合があります。 logger.properties ファイルを使用することにより、スタンドアロン・ファイル・ロガーを調整し、重複の数を減らすことができます。

ファイル・ロガー・ログの重複メッセージ

障害が発生した場合、IBM MQ にコミットされている SYSTEM.FTE/Log# トピックからログ・メッセージをコンシュームせずに、スタンドアロン・ファイル・ロガーのログにログ・メッセージが書き込まれる場合があります。 これが発生した場合、スタンドアロン・ファイル・ロガーが再開するときに同じメッセージが再度取得され、それがログ・ファイルに再度書き込まれます。 ログ・ファイルを手動で見る場合でも自動的に処理する場合でも、このような重複が発生したときの対処法を計画してください。 重複の検出を支援する機能として、スタンドアロン・ファイル・ロガーが開始するときに以下のメッセージがログ・ファイルに出力されます。

```
BFGDB0054I: The file logger has successfully started
```

前のインスタンスが失敗する前に読み取られた最終メッセージはスタンドアロン・ファイル・ロガーの開始時に処理されるため、重複が発生するとすれば常にそのときです。 新規インスタンスがいつ開始したかが分かれば、重複発生の可能性とそれらに対処する必要があるかどうかを予測できます。

重複の数の削減

スタンドアロン・ファイル・ロガーは、パフォーマンス向上のため、処理するログ・メッセージをグループ化して各トランザクションにまとめます。 このバッチ・サイズは、障害時に発生する可能性のある重複メッセージの最大数になります。 重複の数を減らすには、logger.properties ファイル内の以下のプロパティーを調整します：

```
wmqfte.max.transaction.messages
```

例えば、この値を 1 に設定すると、重複メッセージの最大数が 1 に削減されます。 この値を変更すると、スタンドアロン・ファイル・ロガーのパフォーマンスに影響を与えるので、システムに悪影響を及ぼさないように徹底的なテストを行う必要があることに注意してください。

logger.properties ファイルは、*MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* ディレクトリーにあります。 使用可能なプロパティーとそれらの効果の詳細については、MFT ロガーの構成プロパティーを参照してください。

ALW ***MFT* スタンドアロン・ファイル・ロガーのユーザー・アクセス権限の構成**
テスト環境では、通常のユーザー・アカウントに必要な、任意の特権を新たに追加できます。 実稼働環境では、ジョブを実行するために必要な最小限の許可を備えた新しいユーザーを作成することをお勧めします。

このタスクについて

スタンドアロン・ファイル・ロガーと IBM MQ を単一のシステム上にインストールする必要があります。 次のようにして、ユーザーの許可を構成します。

手順

1. Managed File Transfer のインストールの一部としてインストールされたファイルの読み取り許可、および必要に応じて実行許可が、ユーザーに確実に付与されているようにしてください。
2. ユーザーが、構成ディレクトリー内の `logs` ディレクトリー内の任意のファイルを作成および書き込み許可を持っていることを確認してください。 このディレクトリーは、イベント・ログで使用され、必要に応じて診断トレース・ファイルおよび初期障害データ・キャプチャー機能 (FFDC) ファイルでも使用されます。
3. ユーザーが、独自のグループに所属していて、かつ調整キュー・マネージャーの広範囲の許可を備えたグループに所属していないことを確実にしてください。 ユーザーは、mqm グループに所属してはなりません。 特定のプラットフォームでは、 staff グループに、キュー・マネージャーアクセス権限も自動的に付与されます。スタンドアロン・ファイル・ロガー・ユーザーは staff グループに入れないでください。 IBM MQ Explorer を使用すると、キュー・マネージャー自体およびキュー・マネージャー内のオブジェクトの権限レコードを表示できます。 オブジェクトを右クリックして、「**オブジェクト権限**」> 「**権限レコードの管理**」をクリックします。 コマンド行で、コマンド dspmqaut (表示権限)または dmpmqaut (ダンプ権限)を使用することができます。
4. IBM MQ Explorer の 「**権限レコードの管理**」 ウィンドウまたは setmqaut (権限の付与または取り消し) コマンドを使用して、ユーザー自身のグループの権限を追加します (AIX の場合、 IBM MQ 権限はグループにのみ関連付けられ、個々のユーザーには関連付けられません)。 必要な権限は以下のとおりです。
   - キュー・マネージャーに対する CONNECT と INQUIRE (IBM MQ Java ライブラリーを操作するには、INQUIRE 権限が必要です)。
   - SYSTEM.FTE トピックに対する SUBSCRIBE 権限。
   - SYSTEM.FTE.LOG.RJCT.*logger_name* キューに対する PUT 権限。
   - SYSTEM.FTE.LOG.CMD.*logger_name* キューに対する GET 権限。

   リジェクト・キュー名とコマンド・キュー名は、デフォルトの名前です。 スタンドアロン・ファイル・ロガー・キューの構成時に別のキュー名を選択した場合は、それらのキュー名に対する権限を追加してください。

MFT スタンドアロン・データベース・ロガーのインストール

以下の手順に従って、スタンドアロン・データベース・ロガーをインストールし、構成します。

このタスクについて

重要 : Managed File Transfer ロガーは IBM i プラットフォームではサポートされません。

スタンドアロン・データベース・ロガーについて詳しくは、782 ページの『MFT ロガーの構成』を参照してください。

注 : データベース内の同じスキーマに対して一度に複数のデータベース・ロガー (スタンドアロンまたは Java EE) を実行することはできません。 実行しようとすると、転送ログ・データをデータベースに書き込もうとしたときに、競合が起きます。

手順

1. 使用するデータベースの資料に従って、データベース・ソフトウェアをインストールします。

   使用するデータベースで、JDBC サポートがオプションのコンポーネントである場合には、ユーザーがこのコンポーネントをインストールする必要があります。
2. **-loggerType** パラメーターを DATABASE に設定して **fteCreateLogger** コマンドを実行し、スタンドアロン・データベース・ロガーを作成します。 詳しくは、fteCreateLogger を参照してください。

   デフォルトのスキーマ名は FTELOG です。 FTELOG 以外のスキーマ名を使用する場合は、次のステップに進む前に、ご使用のデータベース(`ftelog_tables_db2.sql` または `ftelog_tables_oracle.sql`)に適した用意した SQL ファイルを編集して、スキーマ名を反映させる必要があります。 詳しくは、MFT ロガーの構成プロパティーの wmqfte.database.schema を参照してください。

3. データベースのツールを使用して、必要なデータベース表を作成します。

   Multi マルチプラットフォームでは、ファイル `ftelog_tables_db2.sql` および `ftelog_tables_oracle.sql` には、表を作成するために実行できる SQL コマンドが含まれています。

   z/OS z/OS では、実行する必要があるファイルは、使用している Db2 for z/OS のバージョンによって異なります。

   - Db2 for z/OS 9.0 とそれ以前の場合は、ファイル `ftelog_tables_zos.sql` を実行して表を作成します。 このファイルは、転送されるファイルのサイズと各転送に関連付けられた表 ID を示すフィールドに INTEGER データ型を使用して表を作成します。
   - Db2 for z/OS 9.1 の場合は、ファイル `ftelog_tables_zos_bigint.sql` を実行して表を作成します。 このファイルは、転送されるファイルのサイズと各転送に関連付けられた表 ID を示すフィールドに BIGINT データ型を使用して表を作成します。

4. Oracle を使用している場合は、`listener.ora` ファイルに *`listener_name`* `USE_SID_AS_SERVICE_`*`listener_name`*=on を追加する。 その後、リスナーを再起動する。

   このパラメータは、データベース接続が Oracle システム識別子（SID）をデータベースのサービス名として扱うことを指定します。 このパラメーターが必要な理由は、MFT データベース・ロガーは SID だけを使用して Oracle データベースに接続しますが、それ以降の Oracle データベース（バージョン 12c 以降）ではサービス名を使用するためです。 詳しくは、Oracle のバージョンのドキュメント listener.ora ファイル OracleNet Listener Parameters のなど）を参照してください。 `listener.ora` を変更しないと、ロガーを実行したときに以下のメッセージが表示される場合があります：

   ```
   ORA-12505, TNS:listener does not currently know of SID given in connect descriptor
   ```

5. **fteCreateLogger** コマンドによって提供される MQSC コマンドをロガー・コマンド・キュー・マネージャーに対して実行し、ロガー・キューを作成します。 スタンドアロン・データベース・ロガーは、調整キュー・マネージャー上の 2 つのキューを使用します。 1 つ目のキューは、スタンドアロン・データベース・ロガーの操作を制御するメッセージが置かれるコマンド・キューです。 このコマンド・キューのデフォルトの名前は、SYSTEM.FTE.LOG.CMD.*logger_name* です。 2 つ目のキューはリジェクト・キューです。 スタンドアロン・データベース・ロガーはログ・メッセージを廃棄するということがないため、処理できないメッセージを検出すると、調査 (および可能な場合は再処理) のために、そのメッセージをリジェクト・キューに置きます。 キュー・マネージャーの送達不能キューをこの目的で使用することはお勧めできません。これは、リジェクトされたメッセージには DLH ヘッダーが含まれていないため、 およびリジェクトされたメッセージは、他の理由で送達不能キューに書き込まれたメッセージと組み合わせてはならないためです。 リジェクト・キューのデフォルトの名前は、SYSTEM.FTE.LOG.RJCT.*logger_name* です。 これらの 2 つのキューは、**fteCreateLogger** コマンドによって生成される MQSC スクリプト・ファイルで定義されます。
6. ユーザーの選択と許可の構成
7. オプション: ステップ 800 ページの『2』 で **fteCreateLogger** コマンドを実行したときに作成された `logger.properties` ファイルを編集すると、スタンドアロン・データベース・ロガーをさらに構成できます。 このファイルは、キーと値のペアで構成されている Java プロパティー・ファイルです。 `logger.properties` ファイルは、*`MQ_DATA_PATH`*`/mqft/config/`*`coordination_qmgr_name`*`/loggers/`*`logger_name`* ディレクトリーにあります。 使用可能なプロパティーとそれらの効果の詳細については、MFT ロガーの構成プロパティーを参照してください。
8. Windows

   オプション: Windows システムを使用している場合には、スタンドアロン・データベース・ロガーを Windows サービスとして実行できます。 **-s** パラメーターを指定して、**fteModifyLogger** コマンドを実行します。 詳しくは、fteModifyLogger を参照してください。
9. オプション: 使用しているデータベースが Oracle である場合、または Db2 データベースにリモート接続している場合は、ロガーがデータベース・サーバーへの認証に使用するユーザー名とパスワードを指定する必要があります。 ユーザー名とパスワードは、`MQMFTCredentials.xsd` スキーマによって、定義されたフォーマットに準拠する資格情報ファイルで指定されます。 詳しくは、MFT の資格情報ファイルのフォーマットを参照してください。 資格情報ファイルを作成したら、

logger.properties ファイルで wmqfte.database.credentials.file プロパティーを使用して、資格情報ファイルの場所を指定する必要があります。

10. **fteStartLogger** コマンドを使用して、スタンドアロン・データベース・ロガーを開始します。 デフォルトでは、スタンドアロン・データベース・ロガーはバックグラウンドで実行され、出力を logs ディレクトリー内のファイルに格納します。 スタンドアロン・データベース・ロガーをフォアグラウンドで実行して、ログ・ファイルに加えてコンソールにも出力を生成する場合には、**fteStartLogger** コマンドに **-F** パラメーターを追加します。

    前のステップを実行し、**-s** パラメーターを指定した **fteModifyLogger** コマンドを Windows で使用した場合、スタンドアロン・データベース・ロガーは Windows サービスとして開始されます。

関連タスク

803 ページの『MFT スタンドアロン・データベース・ロガーのユーザー・アクセス権限の構成』
テスト環境では、通常のユーザー・アカウントに必要な、任意の特権を新たに追加できます。 実稼働環境では、ジョブを実行するために必要な最小限の許可を備えた新しいユーザーを作成することをお勧めします。

関連資料

MFT ロガーの構成プロパティー

fteStartLogger

fteModifyLogger

MFT ロガーの権限

リモート・データベースでの *MFT* の使用

Managed File Transfer のロガーを使用して、リモート・システムのデータベースと通信できます。

このタスクについて

Managed File Transfer のインストール先のマシンとは別のマシンにデータベースがインストールされている場合は、以下の手順を実行します。 それぞれの手順は、特に注記がない限り、Db2 と Oracle の両方に当てはまります。

手順

1. Managed File Transfer のインストール先のシステムにデータベース・クライアントをインストールします。
2. ローカル・データベース・クライアント構成にリモート・データベース・サーバーを追加します。 Managed File Transfer と IBM MQ からデータベースに正しくアクセスするために、この構成を更新する必要があります。
3. 新しいプロパティーを logger.properties ファイルに指定し、資格情報ファイル **wmfte.database.credentials.file** を使用してデータベースに接続すします。

    注 : Deprecated 以前のバージョンの Managed File Transfer では、プロパティー **wmqfte.oracle.user** または **wmqfte.database.user**、および **wmqfte.oracle.password** または **wmqfte.database.password** が使用されていました。 それらのプロパティーは、非推奨になっています。 代わりに **wmfte.database.credentials.file** を使用してください。

4. **Oracle のみ:**データベースへのリモート接続を許可するには、調整キュー・マネージャー qm.ini ファイル内の XAResourceManager スタンザを以下のように変更します(データベース名、ユーザー名、およびユーザー・パスワードを、ユーザー独自の情報と一致するように変更します):

    Oracle_XA+Acc=P/ftelog/qgw783jhT+SesTm=35+DB=FTEAUDIT1**+SqlNet=FTEAUDIT1**+threads=false、変更は太字で強調表示されます。
5. **Oracle のみ: wmqfte.oracle.host** プロパティーと **wmqfte.oracle.port** プロパティーを使用して、logger.properties ファイル内のホストとポートを指定します。 ホストとポートのデフォルト値でもローカル・データベース・クライアントを操作できるので、これまで操作していたのがローカル・データベースであれば、これらの値を設定していなかった可能性があります。

関連資料
MFT ロガーの構成プロパティー

MFT スタンドアロン・データベース・ロガーのユーザー・アクセス権限の構成

テスト環境では、通常のユーザー・アカウントに必要な、任意の特権を新たに追加できます。 実稼働環境では、ジョブを実行するために必要な最小限の許可を備えた新しいユーザーを作成することをお勧めします。

このタスクについて

スタンドアロン・データベース・ロガーを実行するために必要なユーザー・アカウントの数とタイプは、使用するシステムの数によって異なります。 スタンドアロン・データベース・ロガー、IBM MQ、およびデータベースを 1 つのシステムにインストールすることも、2 つのシステムにインストールすることも可能です。 スタンドアロン・データベース・ロガーは IBM MQ と同じシステム上になければなりません。 それらのコンポーネントをインストールするためのトポロジーを以下に示します。

スタンドアロン・データベース・ロガー、IBM MQ、およびデータベースをすべて同じシステムに配置する場合
3 つのコンポーネントすべてで使用する 1 つのオペレーティング・システム・ユーザーを定義できます。 これは、スタンドアロン・データベース・ロガーに適した構成です。 スタンドアロン・データベース・ロガーは、バインディング・モードを使用して IBM MQ に接続し、ネイティブ接続を使用してデータベースに接続します。

スタンドアロン・データベース・ロガーと IBM MQ を 1 つのシステムに配置し、データベースを別のシステムに配置する場合
この構成では、2 つのユーザーを作成します。つまり、スタンドアロン・データベース・ロガーを実行するシステムのオペレーティング・システム・ユーザーと、データベース・サーバーのデータベースに対するリモート・アクセスを実行するオペレーティング・システム・ユーザーです。 これは、リモート・データベースを使用するスタンドアロン・データベース・ロガーに適した構成です。 スタンドアロン・データベース・ロガーは、バインディング・モードを使用して IBM MQ に接続し、クライアント接続を使用してデータベースにアクセスします。

以下の説明では、例として `ftelog` というユーザーを想定しますが、任意のユーザー名を使用できます。 次のようにして、ユーザーの許可を構成します。

手順

1. Managed File Transfer Remote Tools and Documentation のインストールの一部としてインストールされたファイルの読み取り許可、および必要に応じて実行許可が、ユーザーに確実に付与されているようにしてください。
2. ユーザーが `logs` ディレクトリー内の任意のファイル(構成ディレクトリー内)の作成および書き込みを行う権限を持っていることを確認してください。 このディレクトリーは、イベント・ログで使用され、必要に応じて診断トレース・ファイルおよび FFDC ファイルでも使用されます。
3. ユーザーが、独自のグループに所属していて、かつ調整キュー・マネージャーの広範囲の許可を備えたグループに所属していないことを確実にしてください。 ユーザーは、mqm グループに所属してはなりません。 特定のプラットフォームでは、 staff グループに、キュー・マネージャーアクセス権限も自動的に付与されます。スタンドアロン・データベース・ロガー・ユーザーは staff グループに入れないでください。 IBM MQ Explorer を使用すると、キュー・マネージャー自体およびキュー・マネージャー内のオブジェクトの権限レコードを表示できます。 オブジェクトを右クリックして、「**オブジェクト権限**」**>**「**権限レコードの管理**」をクリックします。 コマンド行で、コマンド dspmqaut (表示権限)または dmpmqaut (ダンプ権限)を使用することができます。
4. IBM MQ Explorer の 「**権限レコードの管理**」 ウィンドウまたは setmqaut (権限の付与または取り消し) コマンドを使用して、ユーザー自身のグループの権限を追加します (AIX の場合、 IBM MQ 権限はグループにのみ関連付けられ、個々のユーザーには関連付けられません)。 必要な権限は以下のとおりです。
   - キュー・マネージャーに対する CONNECT と INQUIRE (IBM MQ Java ライブラリーを操作するには、INQUIRE 権限が必要です)。
   - SYSTEM.FTE トピックに対する SUBSCRIBE 権限。

- SYSTEM.FTE.LOG.RJCT.*logger_name* キューに対する PUT 権限。
- SYSTEM.FTE.LOG.CMD.*logger_name* キューに対する GET 権限。

リジェクト・キュー名とコマンド・キュー名は、デフォルトの名前です。 スタンドアロン・データベース・ロガー・キューの構成時に別のキュー名を選択した場合は、それらのキュー名に対する権限を追加してください。

5. 使用中のデータベースに該当する固有のユーザー構成を実行します。
   - データベースが Db2 の場合は、以下の手順を実行します。

     Db2 でデータベース・ユーザーを管理するメカニズムはいくつかあります。 ここでの説明は、オペレーティング・システム・ユーザーに基づいたデフォルト・スキームに適用されます。

     – `ftelog` ユーザーが Db2 管理グループ (例: db2iadm1、db2fadm1、dasadm1) に所属していない状態にしてください
     – このユーザーに、データベースに接続する許可、および『ステップ 2: 必要なデータベース表の作成』で作成した表に対する select、insert、および update の許可を付与します
   - データベースが Oracle の場合は、以下の手順を実行します。
     – `ftelog` ユーザーが Oracle 管理グループ (例: Windows での ora_dba や AIX and Linux での dba) に所属していない状態にしてください
     – このユーザーに、データベースに接続する許可、および『ステップ 2: 必要なデータベース表の作成』で作成した表に対する select、insert、および update の許可を付与します

MFT スタンドアロン・ロガーの代替構成

通常、Managed File Transfer スタンドアロン・ロガーは、ファイルかデータベース・タイプかにかかわらず、調整キュー・マネージャーと同じシステム上にあり、IBM MQ バインディング・モードで調整キュー・マネージャーに接続されます。 ただし、調整キュー・マネージャーに接続されているキュー・マネージャーと同じシステムにインストールすることもできます。 スタンドアロン・ロガーは、そのスタンドアロン・ロガーが自動的に作成するサブスクリプションを使用して、メッセージを受信します。 これは、インストール指示に記載されている構成です。

しかし、サイト固有の考慮事項がある場合は、wmqfte.message.source.type プロパティーによって制御される別の 2 つの方法でメッセージを受信するようにスタンドアロン・ロガーを構成できます。 このプロパティーについては、MFT 用のロガー構成プロパティーで説明されています。

管理サブスクリプション

デフォルトでは、スタンドアロン・ロガーは、デフォルトの永続サブスクリプション・オプションおよび管理対象サブスクリプションを使用して、SYSTEM.FTE/Log/# トピックへの独自のサブスクリプションを作成します (つまり、キュー・マネージャーは、アプリケーションにメッセージが渡されるまでそれらの保持に使用するバッキング・キューを制御します)。 サブスクリプションまたはキューで他のオプションが必要になった場合は、代わりに自分でサブスクリプションを作成して必要なオプションを設定し、そのサブスクリプションを使用するようスタンドアロン・ロガーを構成できます。 作成したサブスクリプションを使用するために、スタンドアロン・ロガーの許可を追加することを忘れないでください。

この構成を使用する例として、2 つのワイルドカード・サブスクリプションを使用することによるログ・スペースの分割が挙げられます。FINANCE で始まる名前を持つエージェントから一方のデータベースにログを送信し、ACCOUNTING で始まるエージェントから他方のデータベースにログを送信します。 このタイプの構成では、2 つのスタンドアロン・ロガー・インスタンスが必要とな散ります。それぞれのインスタンスは、必要なサブスクリプションと独自のコマンド・キューとリジェクト・キューを参照する独自の `logger.properties` ファイルがあります。

名前が ACCOUNTING で始まるエージェントからのみログ・メッセージを収集するには、SYSTEM.FTE/Log/ACCOUNTING* というトピック・ストリングで調整キュー・マネージャー上にサブスクリプション・オブジェクトを作成します。 「**ワイルドカード使用**」値を「**文字レベル・ワイルドカード**」に設定します。 また、ロガーの `logger.properties` ファイルにエントリーを追加する必要もあります。 例えば、これら

の設定を使用して、ACCOUNTING.LOGS というサブスクリプション・オブジェクトを作成する場合は、logger.properties ファイルに以下のエントリーを追加します:

```
wmqfte.message.source.type=administrative subscription
wmqfte.message.source.name=ACCOUNTING.LOGS
```

スタンドアロン・ロガーは、トピック・ストリングが SYSTEM.FTE/Log/ でのみ始まるログ・メッセージを扱います。 制限がより多いトピック・ストリングを指定することはできますが、制限がより少ないストリングを指定することはできません。 制限がより少ないストリングを間違って指定した場合、SYSTEM.FTE/Log/ 以外のトピック・ストリングに関連するすべてのパブリケーションはリジェクト・キューに移動し、スタンドアロン・ロガーはエラー・メッセージ BFGDB0002E を生成します。 このエラー・メッセージは、スタンドアロン・ロガー構成に問題があることを暗黙に示します。

キュー

典型的なトポロジーでは、スタンドアロン・ロガーは調整キュー・マネージャーと同じシステム上で実行されます。 これが不可能な場合は、サブスクリプション宛先として別のキュー・マネージャー上のキューを使用して調整キュー・マネージャー上にサブスクリプションを作成します (リモート・キュー定義を使用するか、サブスクリプションの DESTQMGR プロパティーを使用して行います)。 そうすれば、2 番目のキュー・マネージャーをホストしているシステムでロガーを実行して、キューからメッセージを読み取ることができるようになります。 トランザクションの保全性を保証するには、スタンドアロン・ロガーは常にバインディング・モードでそのキュー・マネージャーと接続している必要があります。 リジェクト・キューとコマンド・キューは、スタンドアロン・ロガーの接続先となる同じキュー・マネージャー上で定義する必要があります。 キュー・マネージャーは、IBM WebSphere MQ 7.5 以降でなければなりません。

例えば、サブスクリプションによって、USER.QUEUE キューに配置されているログ・メッセージを収集するには、以下の項目を logger.properties ファイルに追加します:

```
wmqfte.message.source.type=queue
wmqfte.message.source.name=USER.QUEUE
```

MFT 用の Java EE データベース・ロガーのインストール

以下の手順を実行して、Managed File Transfer 用の JEE データベース・ロガーをインストールして構成します。

このタスクについて

Java EE データベース・ロガーについて詳しくは、782 ページの『MFT ロガーの構成』のトピックを参照してください。

注 : Java EE データベース・ロガーとスタンドアロン・ロガーは、それぞれデータベースの別個のインスタンスを使用している場合を除き、同時に実行することができません。

手順

1. Java EE データベース・ロガーをインストールする前に、環境を準備する必要があります。 806 ページの『MFT 用の Java EE データベース・ロガーのインストール準備』のトピックの指示に従ってください。
2. Java EE データベース・ロガーを Java Platform, Enterprise Edition (Java EE) または Jakarta EE 準拠のアプリケーション・サーバーにインストールします。

   手順については、809 ページの『WebSphere Application Server traditional 9.0 を使用した MFT 用 Java EE データベース・ロガーのインストール』を参照してください。

関連タスク

806 ページの『MFT 用の Java EE データベース・ロガーのインストール準備』
Java EE データベース・ロガーをインストールする前に、以下の手順に従って Managed File Transfer 環境を準備します。

809 ページの『WebSphere Application Server traditional 9.0 を使用した MFT 用 Java EE データベース・ロガーのインストール』
WebSphere Application Server traditional 9.0 を使用して Managed File Transfer 用の Java Platform, Enterprise Edition (Java EE) データベース・ロガーをインストールおよび構成するには、以下の手順に従います。

813 ページの『MFT 用の Java EE データベース・ロガーのユーザー・アクセス権限の構成』
Managed File TransferJava Platform, Enterprise Edition (Java EE)データベース・ロガーを構成する場合、IBM MQ、データベース、およびオペレーティング・システムにアクセスするためのユーザー・アカウントが必要です。 必要なオペレーティング・システム・ユーザーの数は、それらのコンポーネントのホストとして使用するシステムの数によって異なります。

815 ページの『スタンドアロン・データベース・ロガーから MFT 用 Java EE データベース・ロガーへのマイグレーション』
スタンドアロン・データベース・ロガーから Java EE データベース・ロガーにマイグレーションできます。スタンドアロン・データベース・ロガーを停止して、JEE データベース・ロガーをインストールする必要があります。 ログ・エントリーが失われたり重複したりしないようにするには、 SYSTEM.FTE トピックを停止し、 Java EE データベース・ロガーをインストールした後に再始動します。 マイグレーション前にデータベースをバックアップしてください。

関連資料
MFT ロガーの権限

MFT 用の *Java EE* データベース・ロガーのインストール準備
Java EE データベース・ロガーをインストールする前に、以下の手順に従って Managed File Transfer 環境を準備します。

このタスクについて
Java EE データベース・ロガーについて詳しくは、782 ページの『MFT ロガーの構成』のトピックを参照してください。

手順
1. 使用するデータベースの資料に従って、データベース・ソフトウェアをインストールします。
   使用するデータベースで、JDBC サポートがオプションのコンポーネントである場合には、ユーザーがこのコンポーネントをインストールする必要があります。
2. データベースに用意されているツールを使用してデータベースを作成します。 データベースのテーブル・スペース・サイズとバッファー・プール・ページ・サイズは少なくとも 8K である必要があります。
   デフォルトのスキーマ名は FTELOG です。 FTELOG 以外のスキーマ名を使用する場合は、用意されている SQL ファイル(使用中のデータベースにに応じて、)を編集して、`ftelog_tables_db2.sql` または `ftelog_tables_oracle.sql`。次のステップに進む前にそれを反映させる必要があります。

   注: `ftelog_tables_db2.sql` ファイルと `ftelog_tables_oracle.sql` ファイルはファイル・パスにあります `<MQ-installation-path>/mqft/sql`
3. データベースのツールを使用して、必要なデータベース表を作成します。

   Multi マルチプラットフォームでは、ファイル `ftelog_tables_db2.sql` および `ftelog_tables_oracle.sql` には、表を作成するために実行できる SQL コマンドが含まれています。

   z/OS z/OS では、実行する必要があるファイルは、使用している Db2 for z/OS のバージョンによって異なります。

   - Db2 for z/OS 9.0 とそれ以前の場合は、ファイル `ftelog_tables_zos.sql` を実行して表を作成します。 このファイルは、転送されるファイルのサイズと各転送に関連付けられた表 ID を示すフィールドに INTEGER データ型を使用して表を作成します。
   - Db2 for z/OS 9.1 の場合は、ファイル `ftelog_tables_zos_bigint.sql` を実行して表を作成します。 このファイルは、転送されるファイルのサイズと各転送に関連付けられた表 ID を示すフィールドに BIGINT データ型を使用して表を作成します。

4. Oracle を使用している場合は、listener.ora ファイルに *listener_name* USE_SID_AS_SERVICE_*listener_name*=on を追加する。 その後、リスナーを再起動する。

   このパラメータは、データベース接続が Oracle システム識別子（SID）をデータベースのサービス名として扱うことを指定します。 このパラメーターが必要な理由は、MFT データベース・ロガーは SID だけを使用して Oracle データベースに接続しますが、それ以降の Oracle データベース（バージョン 12c 以降）ではサービス名を使用するためです。 詳しくは、Oracle のバージョンのドキュメント listener.ora ファイル OracleNet Listener Parameters のなど）を参照してください。
   listener.ora を変更しないと、ロガーを実行したときに以下のメッセージが表示される場合があります：

   ```
   ORA-12505, TNS:listener does not currently know of SID given in connect descriptor
   ```

5. FTELOG 以外のスキーマ名に変更した場合は、EAR ファイルでスキーマ名を変更する必要があります。 詳しくは、807 ページの『MFT 用の Java EE データベース・ロガーでのスキーマ名の変更』を参照してください。
6. IBM MQ でリジェクト・キューを作成します。

   ロガーはログ・メッセージを廃棄するということがないため、処理できないメッセージを検出すると、調査 (および可能な場合は再処理) のために、そのメッセージをリジェクト・キューに置きます。 キュー・マネージャーの送達不能キューをこの目的で使用しないでください。これは、リジェクトされたメッセージには DLH ヘッダーが含まれていないため、 およびリジェクトされたメッセージは、他の理由で送達不能キューに書き込まれたメッセージと組み合わせてはならないためです。
   fteCreateLogger コマンドは、リジェクト・キューを作成します。 このリジェクト・キューのデフォルトの名前は、SYSTEM.FTE.LOG.RJCT.*logger_name* です。
7. 813 ページの『MFT 用の Java EE データベース・ロガーのユーザー・アクセス権限の構成』のトピックに記載されている手順に従ってください。

次のタスク

Java EE データベース・ロガーを Java EE または Jakarta EE 準拠のアプリケーション・サーバーにインストールします。 809 ページの『WebSphere Application Server traditional 9.0 を使用した MFT 用 Java EE データベース・ロガーのインストール』の指示に従ってください。

MFT 用の *Java EE* データベース・ロガーでのスキーマ名の変更

Java Platform, Enterprise Edition (Java EE) データベース・ロガーでは、デフォルト以外のスキーマ名を持ったデータベースを使用できます。 Java EE データベース・ロガーの EAR ファイルでスキーマ名を変更する必要があります。

このタスクについて

Java EE データベース・ロガーで使用するスキーマ名を変更するには、以下の手順を実行します。

手順

1. 以下のコマンドを使用して、その EAR ファイルから JPA の JAR ファイルを抽出します。

   ```
   jar -xvf ear_file lib/jpa_file
   ```

   ここで、

   - *ear_file* は、また com.ibm.wmqfte.databaselogger.jee.oracle.earDb2Oracle を使用しているかによって、com.ibm.wmqfte.databaselogger.jee.ear 異なります。
   - *jpa_file* は、また com.ibm.wmqfte.web.jpa.oracle.jarDb2Oracle を使用しているかによって、com.ibm.wmqfte.web.jpa.jar 異なります。
2. 以下のコマンドを使用して、JPA JAR ファイルから persistence.xml ファイルを抽出します:

   ```
   jar -xvf lib/jpa_file META_INF/persistence.xml
   ```

   ここで、

- *jpa_file* は、また com.ibm.wmqfte.web.jpa.oracle.jarDb2Oracle を使用しているかによって、com.ibm.wmqfte.web.jpa.jar 異なります。

3. persistence.xml ファイルを編集して、以下の行を変更します:

```
<property name="openjpa.jdbc.Schema" value="schema_name" />
```

説明:

- *schema_name* は、使用するスキーマ名です。

4. 以下のコマンドを使用して、変更した persistence.xml ファイルを使用して、JPA JAR を更新します:

```
jar -uvf lib/jpa_file META_INF/persistence.xml
```

ここで、

- *jpa_file* は、また com.ibm.wmqfte.web.jpa.oracle.jarDb2Oracle を使用しているかによって、com.ibm.wmqfte.web.jpa.jar 異なります。

5. 以下のコマンドを使用して、変更後の JPA の JAR ファイルで EAR ファイルを更新します。

```
jar -uvf ear_file lib/jpa_file
```

ここで、

- *ear_file* は、また com.ibm.wmqfte.databaselogger.jee.oracle.earDb2Oracle を使用しているかによって、com.ibm.wmqfte.databaselogger.jee.ear 異なります。
- *jpa_file* は、また com.ibm.wmqfte.web.jpa.oracle.jarDb2Oracle を使用しているかによって、com.ibm.wmqfte.web.jpa.jar 異なります。

次のタスク

変更後の EAR ファイルを使用して、Java EE データベース・ロガーをインストールします。

関連タスク

809 ページの『WebSphere Application Server traditional 9.0 を使用した MFT 用 Java EE データベース・ロガーのインストール』

WebSphere Application Server traditional 9.0 を使用して Managed File Transfer 用の Java Platform, Enterprise Edition (Java EE) データベース・ロガーをインストールおよび構成するには、以下の手順に従います。

WebSphere Application Server traditional 9.0 でのネイティブ・ライブラリー・パスの設定

Java Platform, Enterprise Edition (Java EE) データベース・ロガー・アプリケーションを WebSphere Application Server traditional 9.0 にデプロイし、アプリケーションと IBM MQ の間でバインディング・モード接続を使用する場合は、システム上の IBM MQ ネイティブ・ライブラリーの場所を使用して IBM MQ メッセージング・プロバイダーを構成する必要があります。

このタスクについて

アプリケーション・サーバーにネイティブ・ライブラリー・パスを設定しない場合、WebSphere Application Server traditional 9.0 のシステム・アウト・ログで以下のエラー・メッセージを受け取る可能性があります。

```
A connection could not be made to WebSphere MQ for the following reason:
  CC=2;RC=2495;AMQ8568: The native JNI library 'mqjbnd' was not found. [3=mqjbnd]
```

WebSphere Application Server traditional 9.0 管理コンソールを使用して、以下のステップを実行します。

手順

1. ナビゲーション・ペインで、「**リソース**」 > 「**JMS**」 > 「**JMS プロバイダー**」を展開します。

2. 接続ファクトリーまたはバインディング・モード接続を作成するアクティベーション・スペック用に、適切な有効範囲の IBM MQ メッセージング・プロバイダーを選択します。

   注: Server 有効範囲のネイティブ・パス情報は、 より高い有効範囲のネイティブ・パス情報よりも優先して使用され、 Node 有効範囲のネイティブ・パス情報は Cell 有効範囲のネイティブ・パス情報よりも優先して使用されます。
3. 「一般プロパティー」の下で、「**ネイティブ・ライブラリー・パス**」フィールドに、IBM MQ ネイティブ・ライブラリーが含まれるディレクトリーの絶対パス名を入力します。

   例えば、Linux /opt/mqm/java/lib に入力します。 ディレクトリー名を 1 つのみ入力します。
4. 「**OK**」をクリックします。

   パスを設定したら、変更を有効にするために、変更内容をマスター構成に保存する必要があります。
5. アプリケーション・サーバーを再始動して、構成をリフレッシュします。
6. 必須: アプリケーション・サーバーをもう一度再始動して、 ライブラリーをロードします。

関連タスク

809 ページの『WebSphere Application Server traditional 9.0 を使用した MFT 用 Java EE データベース・ロガーのインストール』

WebSphere Application Server traditional 9.0 を使用して Managed File Transfer 用の Java Platform, Enterprise Edition (Java EE) データベース・ロガーをインストールおよび構成するには、以下の手順に従います。

WebSphere Application Server traditional 9.0 を使用した MFT 用 Java EE データベース・ロガーのインストール

WebSphere Application Server traditional 9.0 を使用して Managed File Transfer 用の Java Platform, Enterprise Edition (Java EE) データベース・ロガーをインストールおよび構成するには、以下の手順に従います。

始める前に

JEE データベース・ロガー・アプリケーションをインストールする前に、806 ページの『MFT 用の Java EE データベース・ロガーのインストール準備』 および 808 ページの『WebSphere Application Server traditional 9.0 でのネイティブ・ライブラリー・パスの設定』のトピックに記載された手順に従ってください。

このタスクについて

Java EE データベース・ロガーについて詳しくは、782 ページの『MFT ロガーの構成』を参照してください。

手順

1. XA JDBC プロバイダーをセットアップします。
   a) WebSphere Application Server traditional 9.0 管理コンソールのナビゲーションから、「**リソース**」 > **JDBC** > **JDBC プロバイダー**」 を選択します。
   b)「**新規作成**」をクリックし、コンソール・ウィザードを使用して JDBC プロバイダーを作成します。
   c) ウィザードのステップ 1 で、「**データベース・タイプ**」リストから使用中のデータベースを選択し、「**プロバイダー・タイプ**」リストから関連するプロバイダー・タイプを選択します。 「**実装タイプ**」リストから、「**XA データ・ソース**」を選択します。 「**次へ**」をクリックします。

   z/OS db2jcc_license_cisuz.jar への参照を除去することも、db2jcc.jar を db2jcc4.jar に変更することもできます。つまり、最新バージョンの Db2 に付属している jar ファイルのバージョン、またはローカル・バージョンを変更する必要があります。
   d) ウィザードのステップ 2 で、必要なデータベース JAR ファイルのディレクトリー・ロケーションが正しく設定されていることを確認します。 **次へ** をクリックします。
   e) 要約ページで「**終了**」をクリックして、JDBC プロバイダーを作成します。
2. 認証別名を作成します。 データ・ソースの別名 1 つと IBM MQ の別名 1 つを作成します。

a) WebSphere Application Server traditional 9.0 管理コンソールのナビゲーションから 「**セキュリティー**」 > 「**グローバル・セキュリティー**」 を選択します。

b)「**認証**」の見出しの下で、「**Java 認証・承認サービス (JAAS)**」を展開します。

c)「**J2C 認証データ**」をクリックします。 認証別名のページが開きます。

d) データ・ソースの認証別名を作成します。

i)「**新規**」をクリックします。

ii)「**別名**」、「**ユーザー ID**」、「**パスワード**」、および「**説明**」の詳細情報を入力します。 「**ユーザー ID**」フィールドと「**パスワード**」フィールドに入力する詳細情報は、 データベース・ユーザーを作成した時に入力した詳細情報と一致していなければなりません。 詳しくは、813 ページの『MFT 用の Java EE データベース・ロガーのユーザー・アクセス権限の構成』を参照してください。

iii)「**OK**」をクリックします。

e) IBM MQ の認証別名を作成します。

i)「**新規**」をクリックします。

ii)「**別名**」、「**ユーザー ID**」、「**パスワード**」、および「**説明**」の詳細情報を入力します。 「**ユーザー ID**」フィールドと「**パスワード**」フィールドで入力する詳細情報は、IBM MQ のインストール済み環境のユーザーとパスワード設定に一致していなければなりません。

iii)「**OK**」をクリックします。

3. 次のようにしてデータ・ソースを作成します。

a) WebSphere Application Server traditional 9.0 管理コンソール・ナビゲーションから 「**リソース**」 > **JDBC** > 「**データ・ソース**」 を選択します。

b)「**有効範囲**」ドロップダウン・リストを選択し、有効範囲を適切な値に変更します。 例えば、`Node=yourNode, Server=yourServer` などです。

c)「**新規作成**」をクリックし、コンソール・ウィザードを使用してデータ・ソースを作成します。

d) ウィザードのステップ 1 で、「**データソース名**」フィールドに `wmqfte-database` を「**JNDI 名**」フィールドに `jdbc/wmqfte-database` を入力します。 「**次へ**」をクリックします。

e) ウィザードのステップ 2 で、「**既存 JDBC プロバイダーを選択**」ドロップダウン・リストを使用して、前のステップで作成した JDBC プロバイダーを選択します。 「**次へ**」をクリックします。

f) **Db2:** ウィザードのステップ 3 で、「**ドライバー・タイプ**」フィールドに 4 と入力します。

g) **Db2:** 「**データベース名**」、「**サーバー名**」、および「**ポート番号**」フィールドに詳細を入力して、「**次へ**」をクリックします。

Oracle: 「**URL**」フィールドに接続 URL を入力し、「**データ・ストアのヘルパー・クラス名**」フィールドでデータ・ストアの正しいヘルパーを選択します。

Oracle RAC: Oracle Real Application Cluster に接続する場合、接続 URL には、データベースの使用可能なすべてのインスタンスに接続するために必要なホスト情報が組み込まれている必要があります。

h) ウィザードのステップ 4 で、ステップ 2d で定義したデータ・ソースの認証別名の名前を「**XA リカバリーの認証別名**」リストから選択します。 「**コンポーネント管理認証別名**」リストと「**コンテナー管理認証別名**」リストから同じ名前を選択してください。

i) 要約ページで「**終了**」をクリックして、データ・ソースを作成します。

4. オプション: データ・ソースの構成を検証します。

a) WebSphere Application Server traditional 9.0 管理コンソール・ナビゲーションから 「**リソース**」 > **JDBC** > 「**データ・ソース**」 を選択します。

b)「**テスト接続**」ボタンをクリックします。

5. トピックを作成します。

a) WebSphere Application Server traditional 9.0 の管理コンソールのナビゲーションから、「**リソース**」 >「**JMS**」 >「**トピック**」をクリックします。

b)「**有効範囲**」ドロップダウン・リストを選択し、有効範囲を適切な値に変更します。 例えば、`Node=yourNode, Server=yourServer` などです。

c)「**新規**」をクリックします。

d)「**IBM MQ メッセージング・プロバイダー**」をクリックします。

e) トピックのプロパティー・ページの「**管理**」パネルで、「**Name**」フィールドおよび「**JNDI name**」フィールドに固有の値を選択します。これらのフィールドは、後で構成内で参照します。

f)「**IBM MQ トピック**」パネルの「**トピック名**」フィールドに `SYSTEM.FTE/Log/#` と入力します。

6. 次のようにしてアクティベーション・スペックを作成します。

a) WebSphere Application Server traditional 9.0 の管理コンソールのナビゲーションから、「**リソース**」>「**JMS**」>「**アクティベーション・スペック**」をクリックします。

b)「**有効範囲**」ドロップダウン・リストを選択し、有効範囲を適切な値に変更します。 例えば、`Node=yourNode, Server=yourServer` などです。

c)「**新規**」をクリックします。

d)「**IBM MQ メッセージング・プロバイダー**」をクリックします。

e) ウィザードのステップ 1 で、「**名前**」フィールドと「**JNDI 名**」フィールドに固有値を選択します。これらの値は、構成の後の段階で再び参照することになります。

f) ステップ 1.1 で、ステップ 5 でセットアップしたトピックの JNDI 名を「**宛先 JNDI 名**」フィールドに入力します。

g)「**宛先タイプ**」リストから、「**トピック**」を選択します。

h) ウィザードのステップ 1.2 で、「**永続サブスクリプション**」を選択します。 「**サブスクリプション名**」フィールドに `SYSTEM.FTE.DATABASELOGGER.AUTO` と入力します。

i) ウィザードのステップ 2 で、「**このウィザードに必要な情報をすべて入力**」を選択します。

j) ステップ 2.1 で、「**キュー・マネージャーまたはキュー共用グループ名**」フィールドにキュー・マネージャーの名前を入力します。

k) ステップ 2.2 で、「**トランスポート**」リストから対象のトランスポート方式を選択します。 「**バインディング**」を選択すると、他の情報が不要になります。 「**クライアント**」または「**バインディングとクライアント**」を選択した場合は、「**ホスト名**」、「**ポート**」、および「**サーバー接続チャネル**」の詳細情報を入力してください。

l) オプション: キュー・マネージャーの有無を確認するには、「**テスト接続**」をクリックします。 ただし、ステップ 6n で認証別名の参照を済ませるまでは、`NOT_AUTHORIZED` を受信することになるはずです。

m) **保存** をクリックします。

n) 作成したアクティベーション・スペックの名前をクリックします。 「**構成**」 タブの 「**一般プロパティー**」 セクションで、 「**拡張**」 パネルまでスクロールダウンし、 IBM MQ 接続を識別する固有の名前を 「**クライアント ID**」 フィールドに入力します。 このステップを完了する必要があります。完了しない場合、IBM MQ は `JMSCC0101` エラー・コードを出して、接続を拒否します。

o) トランスポート方式として「**クライアント**」を選択した場合は、「**セキュリティー設定**」パネルにスクロールダウンして、ステップ 8 で定義した認証別名を「**認証別名**」リストから選択します。

p)「**適用**」をクリックします。

q)「**構成**」タブの「**追加プロパティー**」セクションで、「**拡張プロパティー**」をクリックします。 「**拡張プロパティー**」パネルの「**接続コンシューマー**」セクションで、「**最大サーバー・セッション**」フィールドに 1 を入力します。

注 : 先に進む前に、このステップを確実に実行しておきます。 そうしないと、ロガーが正しく作動しない可能性があります。

r)「**構成**」タブの「**追加プロパティー**」セクションで、「**拡張プロパティー**」をクリックします。 「**メッセージ送信失敗時はエンドポイントを停止する**」の値を最小値の 1 に設定します。

_numberOfFailedAttemptsBeforeReject プロパティーの値が 1 より大きい値に設定されている場合 (詳しくは、 9j を参照)、 「**メッセージ配信が失敗した場合にエンドポイントを停止する**」

を少なくとも **_numberOfFailedAttemptsBeforeReject** プロパティーの値に設定します。これにより、処理できないメッセージ (誤った形式の転送ログ・メッセージ) を受け取った場合でも、エンドポイントは停止されません。 詳しくは、 MFT ロガーのエラー処理とリジェクトを参照してください。

7. キュー接続ファクトリーを作成します。
   a) WebSphere Application Server traditional 9.0 の管理コンソールのナビゲーションから、「**リソース**」>「**JMS**」>「**キュー接続ファクトリー**」をクリックします。
   b)「**有効範囲**」ドロップダウン・リストを選択し、有効範囲を適切な値に変更します。 例えば、Node=yourNode, Server=yourServer などです。
   c)「**新規**」をクリックします。
   d)「**IBM MQ メッセージング・プロバイダー**」をクリックします。
   e) ウィザードのステップ 1 で、「**名前**」フィールドと「**JNDI 名**」フィールドに固有値を選択します。これらの値は、構成の後の段階で再び参照することになります。
   f) ステップ 2 で、「**このウィザードに必要な情報をすべて入力**」を選択します。
   g) ステップ 2.1 で、「**キュー・マネージャーまたはキュー共用グループ名**」フィールドにキュー・マネージャーの名前を入力します。
   h) ステップ 2.2 で、「**トランスポート**」リストから対象のトランスポート方式を選択します。 「**バインディング**」を選択すると、他の情報が不要になります。 「**クライアント**」または「**バインディングとクライアント**」を選択した場合は、「**ホスト名**」、「**ポート**」、および「**サーバー接続チャネル**」の詳細情報を入力してください。
   i) オプション: キュー・マネージャーの有無を確認するには、「**テスト接続**」をクリックします。 ただし、ステップ 7h で認証別名の参照を済ませるまでは、NOT_AUTHORIZED を受信することになるはずです。
   j) トランスポート方式として「**クライアント**」または「**バインディングとクライアント**」を選択した場合は、作成したキュー接続ファクトリーの名前をクリックします。 「**構成**」タブの「**セキュリティー設定**」パネルにスクロールダウンして、ステップ 2e で定義した認証別名を「**XA リカバリーの認証別名**」リストと「**コンテナー管理認証別名**」リストから選択します。
8. WebSphere Application Server でリジェクト・キューを作成します。
   a) WebSphere Application Server traditional 9.0 の管理コンソールのナビゲーションから、「**リソース**」>「**JMS**」>「**キュー**」をクリックします。
   b)「**有効範囲**」ドロップダウン・リストを選択し、有効範囲を適切な値に変更します。 例えば、Node=yourNode, Server=yourServer などです。
   c)「**新規**」をクリックします。
   d)「**IBM MQ メッセージング・プロバイダー**」をクリックします。
   e)「**名前**」フィールドと「**JNDI 名**」フィールドに固有値を選択します。これらの値は、構成の後の段階で再び参照することになります。
   f)「**キュー名**」フィールドに、 SYSTEM.FTE.LOG.RJCT.*logger_name* と入力します。 このキューを調整キュー・マネージャー上に作成したことを確認します。
   g)「**キュー・マネージャー名**」フィールドにキュー・マネージャーの名前を入力します。
   h)「**OK**」をクリックします。
9. JEE データベース・ロガー・アプリケーションをインストールします。
   a) WebSphere Application Server traditional 9.0 の管理コンソールから、「**アプリケーション**」 >「**新規アプリケーション**」を選択します。
   b)「**有効範囲**」ドロップダウン・リストを選択し、有効範囲を適切な値に変更します。 例えば、Node=yourNode, Server=yourServer などです。
   c) オプション・リストから、「**新規エンタープライズ・アプリケーション**」を選択します。
   d) **アプリケーション・インストールの準備** のページで、Managed File Transfer Service インストールの *MQ_INSTALLATION_PATH*/mqft/web ディレクトリーから com.ibm.wmqfte.databaselogger.jee.ear ファイルまたは

com.ibm.wmqfte.databaselogger.jee.oracle.ear ファイルを選択し、「**次**」をクリックします。

e) 次の画面で、「**詳細**」を選択して、すべてのインストール・オプションおよびパラメーターを表示し、「**次へ**」をクリックします。

f) ウィザードのステップ 1 から 4 で、「**次へ**」をクリックして、デフォルト値を受け入れます。

g) ウィザードのステップ 5 の「**メッセージ駆動型 Bean のリスナーをバインド**」で、「**リスナー・バインディング**」セクションにスクロールします。 「**アクティベーション・スペック**」をクリックします。

以下のフィールドに必要な値を入力します。

ターゲット・リソース JNDI 名
ステップ 6e でアクティベーション・スペックを作成したときに指定した JNDI 名。

宛先 JNDI 名
ステップ 5e でトピックを作成したときに指定した JNDI 名。

次へ をクリックします。

h) ウィザードのステップ 6 の「**リソース参照をリソースにマップ**」で、「**ターゲット・リソース JNDI 名**」フィールドに詳細情報を入力します。 この名前は、ステップ 7c でリジェクト・キュー接続ファクトリーに指定した JNDI 名です。 **次へ** をクリックします。

i) ウィザードのステップ 7 の「**リソース環境エントリー参照をリソースにマップ**」で、「**ターゲット・リソース JNDI 名**」フィールドに詳細情報を入力します。 この名前は、ステップ 8d で作成したリジェクト・キューの JNDI 名です。 **次へ** をクリックします。

j) ウィザードのステップ 8 で、**EJB モジュールの環境エントリーをマップ**のデフォルト値である 1 を受け入れます。 「**次へ**」をクリックします。

Oracle RAC: Oracle Real Application Cluster に接続する場合は、_numberOfFailedtsBeforeReject プロパティーの値を**少なくとも** 2 に設定する必要があります。 このプロパティーは、障害が発生した後にロガーが監査メッセージの処理を試行する回数を決定します。 データベースのフェイルオーバーでは、少なくとも 1 つの障害が発生する可能性があります。 リジェクト・キューへのメッセージの不要な移動を回避するには、この値を大きくして、再度試行されるようにします。これにより、新しいデータベース・インスタンスに接続されるため、通常は成功します。 それでもまだ、データベース・インスタンスのフェイルオーバー中にメッセージがリジェクト・キューに移動されることが、テストでわかった場合には、この値をさらに大きくします。インスタンス間の切り替えを行うタイミングにより、同じメッセージに対して複数の障害が引き起こされる場合があるからです。 ただし、この値を大きくすると、データベースのフェイルオーバーだけでなく、すべての障害事例 (例えば、誤った形式のメッセージ) に影響します。そのため、値を大きくする際には、不要な再試行が行われないように注意してください。

k) ウィザードのステップ 9 の「**モジュールのメタデータ**」で、「**次へ**」をクリックします。

l) ウィザードのステップ 10 の「**要約**」で、「**終了**」をクリックします。

10. これで、WebSphere Application Server traditional 9.0 管理コンソールからアプリケーションを開始できるようになりました。

a) コンソール・ナビゲーションから「**アプリケーション**」>「**アプリケーション・タイプ**」>「**WebSphere エンタープライズ・アプリケーション**」と選択します。

b) コレクション表で**ロガー**・エンタープライズ・アプリケーションの チェック・ボックスを選択し、「**開始**」をクリックします。

MFT 用の *Java EE* データベース・ロガーのユーザー・アクセス権限の構成

Managed File TransferJava Platform, Enterprise Edition (Java EE)データベース・ロガーを構成する場合、IBM MQ、データベース、およびオペレーティング・システムにアクセスするためのユーザー・アカウントが必要です。 必要なオペレーティング・システム・ユーザーの数は、それらのコンポーネントのホストとして使用するシステムの数によって異なります。

このタスクについて

Java EE データベース・ロガーを実行するために必要なユーザー・アカウントの数とタイプは、使用するシステムの数によって異なります。 以下の 3 つの環境にアクセスするためにユーザー・アカウントが必要です。

- ローカル・オペレーティング・システム
- IBM MQ
- データベース

JEE データベース・ロガー、IBM MQ、およびデータベースは、1 つのシステムにインストールすることも、複数のシステムに分けてインストールすることも可能です。 それらのコンポーネントをインストールするためのトポロジーの例を以下に示します。

Java EE データベース・ロガー、IBM MQ、およびデータベースをすべて同じシステムに配置する場合
3 つのコンポーネントすべてで使用する 1 つのオペレーティング・システム・ユーザーを定義できます。 ロガーは、バインディング・モードを使用して IBM MQ に接続し、ネイティブ接続を使用してデータベースに接続します。

Java EE データベース・ロガーと IBM MQ を 1 つのシステムに配置し、別のシステムにデータベースを配置する場合
この構成では、2 つのユーザーを作成します。つまり、ロガーを実行するシステムのオペレーティング・システム・ユーザーと、データベース・サーバーのデータベースに対するリモート・アクセスを実行するオペレーティング・システム・ユーザーです。 ロガーは、バインディング・モードを使用して IBM MQ に接続し、クライアント接続を使用してデータベースにアクセスします。

Java EE データベース・ロガー、IBM MQ、データベースをそれぞれ別々のシステムに配置する場合
この構成では、3 つのユーザーを作成します。つまり、アプリケーション・サーバーを開始するオペレーティング・システム・ユーザーと、使用するキューとトピックにアクセスする IBM MQ ユーザーと、データベース表にアクセスしてデータを挿入するデータベース・サーバー・ユーザーです。 ロガーは、クライアント・モードを使用して IBM MQ にアクセスし、クライアント接続を使用してデータベースにアクセスします。

例として、これ以降の説明では、ユーザーの名前が `ftelog` であることを前提としていますが、新規または既存の任意のユーザー名を使用できます。 次のようにして、ユーザーの許可を構成します。

手順

1. オペレーティング・システムのユーザーが、独自のグループに所属していて、かつ調整キュー・マネージャーの広範囲の許可を備えたグループに所属していないことを確実にしてください。 ユーザーは、mqm グループに所属してはなりません。 特定のプラットフォームでは、staff グループに、キュー・マネージャーアクセス権限も自動的に付与されます。ロガー・ユーザーは staff グループに入れないでください。 IBM MQ Explorer を使用すると、キュー・マネージャー自体およびキュー・マネージャー内のオブジェクトの権限レコードを表示できます。 オブジェクトを右クリックして、「**オブジェクト権限**」 **>**「**権限レコードの管理**」をクリックします。 コマンド行で、コマンド dspmqaut (表示権限)または dmpmqaut (ダンプ権限)を使用することができます。
2. IBM MQ Explorer の 「**権限レコードの管理**」 ウィンドウまたは setmqaut (権限の付与または取り消し) コマンドを使用して、 IBM MQ ユーザー自身のグループの権限を追加します (AIX の場合、 IBM MQ 権限はグループにのみ関連付けられ、個々のユーザーには関連付けられません)。 必要な権限は以下のとおりです。

   - キュー・マネージャーに対する CONNECT と INQUIRE (IBM MQ Java ライブラリーを操作するには、INQUIRE 権限が必要です)。
   - SYSTEM.FTE トピックに対する SUBSCRIBE 権限。
   - SYSTEM.FTE.LOG.RJCT.*logger_name* キューに対する PUT 権限。

   リジェクト・キュー名とコマンド・キュー名は、デフォルトの名前です。 ロガー・キューの構成時に別のキュー名を選択した場合は、それらのキュー名に対する権限を追加してください。
3. 使用中のデータベースに該当する固有のデータベース・ユーザー構成を実行します。

   - データベースが Db2 の場合は、以下の手順を実行します。

注 : Db2 でデータベース・ユーザーを管理するメカニズムはいくつかあります。 ここでの説明は、オペレーティング・システム・ユーザーに基づいたデフォルト・スキームに適用されます。

- `ftelog` ユーザーがどの Db2 管理グループ (db2iadm1、db2fadm1、dasadm1 など) にも属していないことを確認します。
- このユーザーに、データベースに接続する許可、および『ステップ 2: 必要なデータベース表の作成』で作成した表に対する select、insert、および update の許可を付与します。

- データベースが Oracle の場合は、以下の手順を実行します。
  - `ftelog` ユーザーがどの Oracle 管理グループにも属していないことを確認します (例えば、Windows の場合は ora_dba、AIX and Linux の場合は dba)。
  - このユーザーに、データベースに接続する許可、および『ステップ 2: 必要なデータベース表の作成』で作成した表に対する select、insert、および update の許可を付与します。

スタンドアロン・データベース・ロガーから MFT 用 Java EE データベース・ロガーへのマイグレーション

スタンドアロン・データベース・ロガーから Java EE データベース・ロガーにマイグレーションできます。スタンドアロン・データベース・ロガーを停止して、JEE データベース・ロガーをインストールする必要があります。 ログ・エントリーが失われたり重複したりしないようにするには、 SYSTEM.FTE トピックを停止し、 Java EE データベース・ロガーをインストールした後に再始動します。 マイグレーション前にデータベースをバックアップしてください。

このタスクについて

手順

1. データベースを停止する前に、調整キュー・マネージャーに対して次の MQSC コマンドを実行します。
   `ALTER QM PSMODE(COMPAT)`
   これによって SYSTEM.FTE/Log トピックへのメッセージのパブリッシュが停止します。 ロガーがそのサブスクリプション上のメッセージをすべて処理するまで待ちます。 デフォルトでは、このサブスクリプションは SYSTEM.FTE.LOGGER.AUTO という名前です。
2. **fteStopLogger** コマンドを使用して、データベース・ロガーを停止します。
3. データベース・ソフトウェアに付属のツールを使用して、データベースをバックアップします。
4. スタンドアロンのデータベース・ロガーに属するサブスクリプションを削除します。
   デフォルトでは、このサブスクリプションは SYSTEM.FTE.LOGGER.AUTO という名前です。
5. 使用中のデータベース・スキーマが以前のバージョンの場合、スキーマを順番に以降のレベルにマイグレーションする必要があります。 例えば、データベース・スキーマを V7.0.1 から V7.0.4 にマイグレーションする場合は、まず V7.0.1 から V7.0.2 にマイグレーションしてから、V7.0.2 から V7.0.3 に、さらに V7.0.3 から V7.0.4 にマイグレーションする必要があります。 データベース・スキーマを、バージョン *old* からバーション *new* で移行する。ここで、*old* および *new* は、スキーマ・バージョンを記述する変数です。移行する必要があるスキーマのバージョンごとに、以下のいずれかのアクションを実行して行います:
   - z/OS データベースが Db2 on z/OS であり、V7.0.2 スキーマと V7.0.3 スキーマの間、または V7.0.3 スキーマと V7.0.4 スキーマの間でマイグレーションする場合は、新しいデータベース・スキーマを作成し、既存のデータをそのスキーマにコピーする必要があります。 詳しくは、Db2 資料を参照してください。
   - データベースが Db2 でない場合や、8K を超えるページ・サイズでデータベースを作成していた場合は、以下の手順を実行して、他のバージョンの場合と同じ要領でスキーマをマイグレーションできます。
   - その他の環境でデータベース表をマイグレーションする場合は、以下の手順を実行します。

a. ご使用のデータベース・プラットフォームに適した、名前にストリング *old-new* が含まれているファイルを選択します。 このファイルは、Remote Tools and Documentation インストール済み環境の *MQ_INSTALLATION_PATH*/mqft/sql ディレクトリーにあります。

b. 初期スキーマに対して変更を行った場合は、マイグレーション・ファイルを調べて、ファイルと変更済みデータベースとの互換性が確保されるようにしてください。

c. ご使用のデータベースに対して SQL ファイルを実行します。

6. Java EE データベース・ロガー EAR ファイルをインストールします。
7. Java EE データベース・ロガーをデプロイします。 詳しくは、805 ページの『MFT 用の Java EE データベース・ロガーのインストール』を参照してください。
8. 調整キュー・マネージャーに対して次の MQSC コマンドを実行します。`ALTER QMGR PSMODE(ENABLED)`

   これによってメッセージが SYSTEM.FTE/Log トピックにパブリッシュされるようになります。

タスクの結果

Connect:Direct ブリッジの構成

Managed File Transfer ネットワークと Connect:Direct ネットワークの間でファイルを転送するように Connect:Direct ブリッジを構成します。 Connect:Direct ブリッジのコンポーネントは、Connect:Direct ノードと、そのノードと通信するための専用の Managed File Transfer エージェントです。 このエージェントのことを Connect:Direct ブリッジ・エージェントといいます。

始める前に

Connect:Direct ブリッジを構成するエージェントとノードは、同じシステムに存在しているか、共用 NFS マウントなどによって同じファイル・システムにアクセスできる状態になっている必要があります。 このファイル・システムは、Connect:Direct ブリッジに関係するファイル転送中に、**cdTmpDir** パラメーターで定義されたディレクトリーにファイルを一時的に保管するために使用されます。 Connect:Direct ブリッジ・エージェントと Connect:Direct ブリッジ・ノードでは、同じパス名を使用してこのディレクトリーを指定できる必要があります。 例えば、エージェントとノードが別個の Windows システムにある場合、共有ファイル・システムをマウントするためにそれらのシステムで同じドライブ名が使用されている必要があります。 以下の構成を使用すると、エージェントとノードで同じパス名を使用できます。

- エージェントとノードが、Windows または Linux for x86-64 のいずれかを実行する同じシステム上にある
- エージェントが Linux for x86-64 上にあり、ノードが AIX 上にある
- エージェントが Windows システム上にあり、ノードがそれとは別の Windows システム上にある

以下の構成を使用すると、エージェントとノードで同じパス名を使用できません。

- エージェントが Linux for x86-64 上にあり、ノードが Windows 上にある
- エージェントが Windows 上にあり、ノードが UNIX 上にある

Connect:Direct ブリッジのインストールを計画する際には、これらの制約事項を考慮してください。

Connect:Direct ブリッジでサポートされるオペレーティング・システム・バージョンの詳細については、System Requirements for IBM MQ の Web ページを参照してください。

このタスクについて

Connect:Direct ブリッジ・エージェントは、Connect:Direct ノードとの通信専用の Managed File Transfer エージェントです。

デフォルトでは、Connect:Direct ブリッジ・エージェントは、TCP/IP プロトコルを使用して Connect:Direct ノードに接続します。 Connect:Direct ブリッジ・エージェントと Connect:Direct ノードの間にセキュア接続を確立する場合は、SSL プロトコルまたは TLS プロトコルを使用できます。

手順

1. Connect:Direct ブリッジ・エージェントおよびノードのオペレーティング・システムを選択します。
   a) ブリッジエージェント Connect:Direct をインストールするために、x86-64 で Windows どちら Linux かが稼働しているシステムを選択します。
   b) Connect:Direct for Windows または Connect:Direct for UNIX でサポートされるオペレーティング・システムを選択して、 Connect:Direct ブリッジ・ノードをインストールします。
2. Connect:Direct ノードを選択および構成する。
   以下の手順を実行する前に、Connect:Direct ノードをインストールしておく必要があります。
   a) Managed File Transfer エージェントが通信するために使用する Connect:Direct ノードを選択します。
   b) 選択した Connect:Direct ノードのネットワーク・マップを確認します。 Windows オペレーティング・システム上で実行中のリモート・ノードの項目がネットワーク・マップに含まれている場合は、それらのノードが Windows で実行されていることが、それらの項目で指定されていることを確認します。

      Windows

      Connect:Direct ブリッジ用に選択した Connect:Direct ノードが Windows 上で実行されている場合は、Connect:Direct リクエスターを使用して、ネットワーク・マップを編集します。 Windows で実行されているすべてのリモート・ノードの 「**オペレーティング・システム**」 フィールドが 「**Windows**」に設定されていることを確認します。
3. Connect:Direct ブリッジ・エージェントを作成および構成する。
   a) **fteCreateCDAgent** コマンドを使用して Connect:Direct ブリッジ・エージェントを作成します。
      - **cdNode** パラメーターには値を指定する必要があります。 このパラメーターでは、エージェントが Connect:Direct ブリッジの一部になる Connect:Direct ノードに対して使用する名前を指定します。 前のセクションで選択した Connect:Direct ノードの名前を使用します。
      - **cdNodeHost** パラメーターおよび **cdNodePort** パラメーターの値を指定します。これらは、エージェントが通信する Connect:Direct ノードを定義します。

        cdNodeHost パラメーターに値を指定しないと、ローカル・システムのホスト名または IP アドレスが使用されます。 **cdNodePort** パラメーターに値を指定しないと、値 1363 が使用されます。
      - オプションで、fteCreateAgent の情報を使用し、**cdTmpDir** パラメーターに値を指定する必要があるかどうかを決定します。
   b) Managed File Transfer で使用するユーザー資格情報を、Connect:Direct ノード上のユーザー資格情報にマップします。 以下のいずれかの方法を使用して資格情報をマップできます。
      - `ConnectDirectCredentials.xml` ファイルを作成して、資格情報マッピング情報を定義します。 詳しくは、818 ページの『ConnectDirectCredentials.xml ファイルを使用した Connect:Direct の資格情報のマップ』を参照してください。
      - 使用する Connect:Direct ブリッジ用に資格情報マッピングを実行するユーザー出口を作成する。 詳しくは、821 ページの『出口クラスを使用した Connect:Direct の資格情報のマップ』を参照してください。
4. `ConnectDirectNodeProperties.xml` ファイルを構成して、リモート Connect:Direct ノードに関する情報を組み込みます。
   以下の手順を実行する前に、Connect:Direct ブリッジ・エージェントを作成しておく必要があります。

   Connect:Direct ブリッジ・エージェント構成ディレクトリー内のテンプレート `ConnectDirectNodeProperties.xml` を編集します。 情報を定義する Connect:Direct ノードまたはノード・グループごとに、以下の手順を実行します。
   a) nodeProperties エレメントの内部に、node エレメントを作成します。
   b) node 要素に name 属性を追加する。 1 つ以上のリモート Connect:Direct ノードの名前に合致するパターンとして、この属性の値を指定します。

c) オプション: node 要素に pattern 属性を追加し、name 属性内の値がどのような種類のパターンであるかを指定します。 有効な値は、`regex` と `wildcard` です。 デフォルト・オプションは `wildcard` です。

d) node 要素に、name 属性で指定されたリモート Connect:Direct ノードが動作するオペレーティング・システムを指定する type 属性を追加する。

有効な値は以下のとおりです。

- `Windows` - ノードは Windows で実行されています
- `UNIX` - ノードは AIX and Linux で実行されています
- z/OS `z/OS`、`zos`、`os/390`、`os390` - ノードは z/OS で実行されています

この属性の値には、大/小文字の区別がありません。 Connect:Direct ブリッジは、その他のオペレーティング・システムのリモート・ノードへの転送には対応していません。

詳しくは、Connect:Direct ノードのプロパティー・ファイルのフォーマットを参照してください。

5. Connect:Direct ブリッジ・エージェントと Connect:Direct ノードの間のセキュア接続を構成する。 その方法を示す例については、Connect:Direct ブリッジ・エージェントと Connect:Direct ノードの間の SSL または TLS の構成を参照してください。

関連タスク

Connect:Direct ブリッジのトラブルシューティング

Connect:Direct ブリッジ・エージェントと Connect:Direct ノードの間の SSL または TLS の構成

Connect:Direct ノードへのファイルの転送

Connect:Direct ノードからのファイルの転送

z/OS Connect:Direct ノードからの複数ファイルの転送

関連資料

Connect:Direct ブリッジ

Connect:Direct の資格情報のマップ

Connect:Direct ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用するか、独自のユーザー出口を作成して、Managed File Transfer のユーザー資格情報を Connect:Direct ノードのユーザー資格情報にマップします。 Managed File Transfer には、ユーザー資格情報マッピングを実行するサンプルのユーザー出口が用意されています。

関連タスク

818 ページの『ConnectDirectCredentials.xml ファイルを使用した Connect:Direct の資格情報のマップ』
Connect:Direct ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用して、Managed File Transfer のユーザー資格情報を Connect:Direct ノードのユーザー資格情報にマップします。 Managed File Transfer で提供される XML ファイルを編集して、ユーザーの資格情報を組み込むことができます。

821 ページの『出口クラスを使用した Connect:Direct の資格情報のマップ』
Connect:Direct ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用しない場合は、独自のユーザー出口を作成して、Managed File Transfer 内のユーザー資格情報を Connect:Direct ノード上のユーザー資格情報にマップできます。 資格情報マッピングの独自のユーザー出口を構成すると、デフォルトの資格情報マッピング機能が使用不可になります。

関連資料

CDCredentialExit.java インターフェース

Connect:Direct 資格情報ファイルのフォーマット

ConnectDirectCredentials.xml ファイルを使用した *Connect:Direct* の資格情報のマップ

Connect:Direct ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用して、Managed File Transfer のユーザー資格情報を Connect:Direct ノードのユーザー資格情報にマップします。 Managed File Transfer で提供される XML ファイルを編集して、ユーザーの資格情報を組み込むことができます。

このタスクについて

fteCreateCDAgent コマンドを使用して Connect:Direct ブリッジ・エージェントを作成した後、ConnectDirectCredentials.xml ファイルを手動で作成する必要があります。 Connect:Direct ブリッジ・エージェントを使用するには、まずこのファイルを編集して、ホスト、ユーザー、および資格情報を含める必要があります。 詳しくは、Connect:Direct の資格情報ファイルのフォーマットを参照してください。 デフォルトでは、このファイルは現行ユーザーのホーム・ディレクトリー(例えば、/home/fteuser/ConnectDirectCredentials.xml)からロードされます。 別の場所を使用するには、ConnectDirectNodeProperties.xml ファイル内の <credentialsFile> エレメントで指定します。

手順

1. エレメント <tns:pnode name="Connect:Direct node host" pattern="wildcard"> の name 属性に、 Connect:Direct ブリッジ・エージェントの接続先の Connect:Direct ノードの名前の値が含まれていることを確認します。 この値は、**fteCreateCDAgent -cdNode** パラメーターに指定した値と同じ値でなければなりません。

   pattern 属性の値は、wildcard または regex のいずれかです。 この属性を指定しない場合、デフォルトは wildcard です。

2. <tns:pnode>の子エレメントとして、ユーザー ID および資格情報をファイルに挿入します。

   以下の<tns:user>エレメントの 1 つ以上のインスタンスをファイルに挿入することができます:

   ```
   <tns:user name="name"
             pattern="pattern"
             ignorecase="ignorecase"
             cdUserId="cdUserId"
             cdPassword="cdPassword"
             pnodeUserId="pnodeUserId"
             pnodePassword="pnodePassword">
   </tns:user>
   ```

   ここで、

   - *name* は、MFT 転送要求に関連付けられた MQMD ユーザー ID と一致するパターンです。
   - *pattern* は、name 属性に指定されたパターンがワイルドカード表現であるか、Java 正規表現であるかを指定します。 pattern 属性の値は、wildcard または regex のいずれかです。 この属性を指定しない場合、デフォルトは wildcard です。
   - *ignorecase* は、name 属性で指定されたパターンを大/小文字の区別があるものとして扱うかどうかを指定します。 この属性を指定しない場合、デフォルトは true です。
   - *cdUserId* は、 <tns:pnode> エレメントの name 属性で指定された Connect:Direct ノードに接続するために Connect:Direct ブリッジ・エージェントが使用するユーザー ID です。 可能な場合は、*cdUserId* に Connect:Direct 管理者ユーザー ID を指定してください。 *cdUserId* に Connect:Direct 管理者を指定できない場合は、使用するユーザー ID が Connect:Direct ブリッジ・ノードにおいて以下の機能権限を持つようにしてください。

     – Windows ノードの場合、以下の権限を設定します。 この例では、読みやすくするために復帰文字を使用してフォーマットしています。

     ```
     View Processes in the  value: yes
      TCQ
     Issue the copy         value: yes
      receive, copy send,
      run job, and run task
      Process statements
     Issue the submit       value: yes
      Process statement
     Monitor, submit,       value: all
      change,and delete all
      Processes
     Access Process         value: all
      statistics
     Use the trace tool or  value: yes
      issue traceon and
     ```

```
 traceoff commands
Override Process       value: yes
 options such as file
 attributes and remote
 node ID
```

– AIX ノードまたは Linux ノードの場合、userfile.cfg ファイル内の以下のパラメーターを設定します:

```
pstmt.copy             value: y
pstmt.upload           value: y
pstmt.download         value: y
pstmt.runjob           value: y
pstmt.runtask          value: y
cmd.submit             value: y
pstmt.submit           value: y
cmd.chgproc            value: y
cmd.delproc            value: y
cmd.flsproc            value: y
cmd.selproc            value: a
cmd.selstats           value: a
cmd.trace              value: y
snode.ovrd             value: y
```

- *cdPassword* は、cdUserId 属性で指定されたユーザー ID に関連付けられているパスワードです。
- オプションで、pnodeUserId 属性を指定できます。 この属性の値は、 Connect:Direct プロセスを実行依頼するために <tns:pnode> エレメントの name 属性によって指定される Connect:Direct ノードによって使用されるユーザー ID です。 pnodeUserId 属性を指定しない場合、Connect:Direct ノードは、cdUserId 属性で指定されたユーザー ID を使用して Connect:Direct プロセスを実行依頼します。
- オプションで、属性 pnodePassword を指定できます。 この属性の値は、pnodeUserId 属性で指定されたユーザー ID に関連付けられているパスワードです。

MQMD ユーザー ID に一致するユーザー・エレメントがない場合、転送は失敗します。

3. オプション: 1 つ以上の <tns:snode> エレメントを、<tns:user> エレメントの子エレメントとして含めることができます。 <tns:snode> エレメントでは、Connect:Direct ブリッジの一部である Connect:Direct ノードで使用する資格情報を指定します。 この資格情報は、Connect:Direct ブリッジ・ノードが、ファイル転送のソースまたは宛先である Connect:Direct ノードに接続するときに使用するユーザー ID とパスワードです。

   ファイルに以下のエレメントを 1 つ以上挿入します。

```
<tns:snode name="name"
           pattern="pattern"
           userId="userId"
           password="password"/>
```

   ここで、

   - *name* は、ファイル転送の転送元または転送先である Connect:Direct ノードの名前とマッチングさせるパターンです。
   - *pattern* は、name 属性に指定されたパターンがワイルドカード表現であるか、Java 正規表現であるかを指定します。 pattern 属性の値は、wildcard または regex のいずれかです。 この属性を指定しない場合、デフォルトは wildcard です。
   - *userId* は、 <tns:pnode> エレメントの name 属性で指定された Connect:Direct ノードが、<tns:snode>の name 属性で指定されたパターンに一致する Connect:Direct ノードに接続するために使用するユーザー ID です。
   - *password* は、userId 属性で指定されたユーザー ID に関連付けられているパスワードです。

   ファイル転送の二次ノードと一致する <tns:snode>エレメントがない場合は、転送が失敗することはありません。 転送が開始され、snode で使用するユーザー ID およびパスワードは指定されません。

タスクの結果

ユーザー名または Connect:Direct ノード名をパターン・マッチで検索するとき、Connect:Direct ブリッジ・エージェントは、ファイルの先頭から末尾に向かって検索します。 最初に見つかった一致が使用されます。

関連タスク

816 ページの『Connect:Direct ブリッジの構成』

Managed File Transfer ネットワークと Connect:Direct ネットワークの間でファイルを転送するように Connect:Direct ブリッジを構成します。 Connect:Direct ブリッジのコンポーネントは、Connect:Direct ノードと、そのノードと通信するための専用の Managed File Transfer エージェントです。 このエージェントのことを Connect:Direct ブリッジ・エージェントといいます。

関連資料

Connect:Direct 資格情報ファイルのフォーマット

fteCreateCDAgent: Connect:Direct ブリッジ・エージェントの作成

出口クラスを使用した *Connect:Direct* の資格情報のマップ

Connect:Direct ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用しない場合は、独自のユーザー出口を作成して、Managed File Transfer 内のユーザー資格情報を Connect:Direct ノード上のユーザー資格情報にマップできます。 資格情報マッピングの独自のユーザー出口を構成すると、デフォルトの資格情報マッピング機能が使用不可になります。

このタスクについて

Connect:Direct の資格情報のマッピングのために作成するユーザー出口では、`com.ibm.wmqfte.exitroutine.api.ConnectDirectCredentialExit` インターフェースを実装する必要があります。 詳しくは、CDCredentialExit.java インターフェースを参照してください。

IBM MQ Console および REST API の構成

IBM MQ Console と REST API をホストする mqweb サーバーには、デフォルトの構成が準備されています。どちらのコンポーネントを使用する場合も、ユーザーにログインを許可するために、セキュリティーを構成するなどの複数の構成タスクを実行する必要があります。 このトピックでは、使用可能なすべての構成オプションについて説明します。

手順

- 822 ページの『mqweb サーバーの基本構成』
- 827 ページの『セキュリティーの構成』
- 827 ページの『HTTP ホスト名の構成』
- 828 ページの『HTTP および HTTPS ポートの構成』
- 830 ページの『応答タイムアウトの構成』
- 831 ページの『自動始動の構成』
- 832 ページの『ロギングの構成』
- 835 ページの『LTPA トークンの構成』
- 837 ページの『IBM MQ Console のリモート・キュー・マネージャー接続の動作の構成』
- 839 ページの『administrative REST API ゲートウェイの構成』
- 840 ページの『messaging REST API の構成』
- 846 ページの『REST API for MFT の構成』
- 852 ページの『mqweb サーバーの JVM のチューニング』
- 853 ページの『IBM MQ Console および REST API のインストール・コンポーネントのファイル構造』

mqweb サーバーの基本構成

REST API または IBM MQ Console の使用を開始するには、その前に正しいコンポーネントをインストールし、REST API または IBM MQ Console をホストする mqweb サーバーを構成する必要があります。

このタスクについて

このタスクの手順では、REST API および IBM MQ Console を素早く開始するための mqweb サーバーの基本構成に焦点を当てています。 セキュリティーを構成する手順では、 基本的なユーザー・レジストリーをセットアップする方法の概要を示していますが、ユーザーと役割を構成する方法には他のオプションも存在します。 mqweb サーバーのセキュリティーの構成について詳しくは、 IBM MQ Console および REST API セキュリティーを参照してください。

注 : この手順を実行するには、`mqwebuser.xml` ファイルにアクセスできる必要があります。

- z/OS z/OS では、`mqwebuser.xml` ファイルへの書き込み権限を持つユーザーでなければなりません。
- Multi 他のすべてのオペレーティング・システムでは、`mqwebuser.xml` ファイルにアクセスするには特権ユーザーでなければなりません。
- Linux V 9.3.5 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、 IBM MQ Web Server データ・ディレクトリー内の `mqwebuser.xml` ファイルに対する書き込み権限が必要です。

手順

1. IBM MQ Console および REST API コンポーネントをインストールします:
   - AIX AIX で、`mqm.web.rte` ファイルセットをインストールします。 AIX にファイル・セットをインストールする方法について詳しくは、AIX でのインストール・タスクを参照してください。
   - IBM i IBM i の場合は、WEB コンポーネントをインストールします。 この機能を使用するには、 5724L26 IBM MQ Java Messaging and Web Services、および 5770JV1 Java SE 8 前提条件もインストールする必要があります。 IBM i にフィーチャーをインストールする方法について詳しくは、IBM i でのインストール・タスクを参照してください。
   - Linux Linux で、`MQSeriesWeb` コンポーネントをインストールします。 Linux にコンポーネントをインストールする方法について詳しくは、Linux でのインストール・タスクを参照してください。

     V 9.3.5 IBM MQ 9.3.5 以降、 Linux 上のスタンドアロン IBM MQ Web Server インストール済み環境で mqweb サーバーを実行することもできます。 IBM MQ Web Server のインストールについて詳しくは、 スタンドアロン IBM MQ Web Server のインストールを参照してください。
   - Windows Windows で、`Web Administration` フィールドをインストールします。 Windows にフィーチャーをインストールする方法について詳しくは、Windows でのインストール・タスクを参照してください。
   - z/OS `IBM MQ for z/OS UNIX System Services Web Components` フィールドをインストールします。 z/OS にコンポーネントおよびフィーチャーをインストールする方法について詳しくは、z/OS でのインストール・タスクを参照してください。
2. IBM MQ Console および REST API をホストする mqweb サーバーを作成します。
   - z/OS z/OS では、 **crtmqweb** スクリプトを実行します。

     このスクリプトにより、mqweb サーバーの構成ファイルおよびログ・ファイルを格納する WebSphere Liberty ユーザー・ディレクトリーが作成されます。 **crtmqweb** スクリプトの実行について詳しくは、977 ページの『mqweb サーバーの作成』を参照してください。

- Linux V 9.3.5 スタンドアロンの IBM MQ Web Server インストール済み環境では、826 ページの『スタンドアロン IBM MQ Web Server の構成』の手順に従います。
- それ以外のすべての環境では、mqweb サーバーを作成するためにアクションを実行する必要はありません。

3. z/OS
   z/OS で、mqweb サーバーを始動するためのカタログ式プロシージャーを作成します。
   詳細については、979 ページの『mqweb サーバー用のプロシージャーの作成』を参照してください。
4. 既存の構成ファイル mqwebuser.xml を、基本セキュリティーを提供するように構成されている基本レジストリー・サンプル・ファイルに置き換えます。 basic_registry.xml ファイルを *MQ_INSTALLATION_PATH*/web/mq/samp/configuration ディレクトリーからご使用のシステムの適切なディレクトリーにコピーし、ファイル名を mqwebuser.xml に変更します。
   - IBM MQ インストール済み環境で、このファイルを以下のディレクトリーにコピーします。
     - Linux AIX AIX and Linux 上: /var/mqm/web/installations/*installationName*/servers/mqweb
     - Windows Windows 上: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb

       ここで、*MQ_DATA_PATH* は、IBM MQ データ・パスであり、このパスは IBM MQ のインストール中に選択されたデータ・パスです。 デフォルトでは、このパスは C:\ProgramData\IBM\MQ です。
     - z/OS z/OS 上: *WLP_user_directory*/servers/mqweb

       ここで、*WLP_user_directory* は、mqweb サーバー定義を作成するために **crtmqweb** スクリプトを実行したときに指定したディレクトリーです。
   - Linux V 9.3.5 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

     ここで、*MQ_OVERRIDE_DATA_PATH* は、**MQ_OVERRIDE_DATA_PATH** 環境変数が指す IBM MQ Web Server データ・ディレクトリーです。

   basic_registry.xml サンプル・ファイルは、以下の 4 人のユーザーを構成します。

   mqadmin
   MQWebAdmin の役割のメンバーである管理ユーザー。

   mqreader
   MQWebAdminRO 役割のメンバーである、読み取り専用の管理ユーザー。

   mftadmin
   MFTWebAdmin の役割のメンバーである管理ユーザー。

   mftreader
   MFTWebAdminRO 役割のメンバーである、読み取り専用の管理ユーザー。

   すべてのユーザーは、 MQWebUser 役割のメンバーでもあります。

   利用可能なロールの詳細については、IBM MQ Console の上のロールおよび REST API でのロール
5. オプション: mqwebuser.xml ファイルを編集して、ユーザーとグループをさらに追加します。 それらのユーザーとグループに、REST API または IBM MQ Console の使用が許可される適切な役割を割り当てます。 デフォルトで定義されているユーザーのパスワードを変更し、新規パスワードをエンコードすることもできます。 詳しくは、ユーザーおよび役割の構成を参照してください。

   注:

- z/OS z/OS では、MQWebUser ロールにユーザーを追加する場合、mqweb 開始タスクのユーザー ID に、MQWebUser ロールを持つユーザー ID への代替ユーザー・アクセス権限も付与する必要があります。 以下に例を示します。

```
RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
```

- z/OS Multi messaging REST API の使用を開始するためのステップを完了するには、mqwebuser.xml ファイルにユーザーを追加する必要があります。 このユーザーは、ご使用のシステムの既存の IBM MQ ユーザーの 1 人と同じ名前でなければなりません。 XML ファイル内の他のユーザーと同じフォーマットに従って、XML ファイル内の行 <user name="mftreader" password="mftreader"/> の後にユーザー ID とパスワードを追加します。

6. mqweb サーバー構成を指すように環境を設定します。

- z/OS z/OS で、以下のコマンドを入力して、変数が mqweb サーバー構成を指すように、WLP_USER_DIR 環境変数を設定します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、 *WLP_user_directory* は、 **crtmqweb** コマンドに渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳細については、977 ページの『mqweb サーバーの作成』を参照してください。

- Linux V 9.3.5 スタンドアロンの IBM MQ Web Server インストール済み環境では、**MQ_OVERRIDE_DATA_PATH** 環境変数を IBM MQ Web Server データ・ディレクトリーに設定します。

例えば、 IBM MQ Web Server データ・ディレクトリーとして /var/mqweb を使用することを選択した場合は、次のコマンドを発行します。

```
export MQ_OVERRIDE_DATA_PATH=/var/mqweb
```

- 他のすべての環境では、環境を設定するためにアクションを実行する必要はありません。

7. デフォルトでは、REST API および IBM MQ Console は mqweb サーバーと同じホストからしか使用できません。 mqweb サーバーへのリモート接続を有効にするには、次のコマンドを入力します。

```
setmqweb properties -k httpHost -v hostname
```

hostName は IP アドレス、ドメイン名サフィックス付きのドメイン・ネーム・サーバー (DNS) ホスト名、または IBM MQ がインストールされているサーバーの DNS ホスト名を示します。 使用可能なすべてのネットワーク・インターフェースを指定するには、次の例に示すように、二重引用符で囲んだアスタリスク * を使用します。

```
setmqweb properties -k httpHost -v "*"
```

8. オプション: デフォルトでは、administrative REST API for MFT は有効になっていません。 このフィーチャーを使用する場合は、これを有効にして、調整キュー・マネージャーを構成する必要があります。

a) MFT 以下のコマンドを入力し、administrative REST APIfor を有効にします:

```
setmqweb properties -k mqRestMftEnabled -v true
```

b) 以下のコマンドを入力して、どのキュー・マネージャーを調整キュー・マネージャーにするかを構成します。

```
setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
```

ここで、*qmgrName* は、調整キュー・マネージャーの名前です。

c) POST 呼び出しを有効にするには、以下のコマンドを入力して、どのキュー・マネージャーをコマンド・キュー・マネージャーにするかを構成します。

```
setmqweb properties -k mqRestMftCommandQmgr -v qmgrName
```

ここで、*qmgrName* は、コマンド・キュー・マネージャーの名前です。

9. REST API および IBM MQ Console をサポートする mqweb サーバーを開始します。

- ALW AIX, Linux, and Windows で、特権ユーザーとして、以下のコマンドを入力します。

```
strmqweb
```

- IBM i IBM i で、特権ユーザーとして Qshell で次のコマンドを入力します。

```
/QIBM/ProdData/mqm/bin/strmqweb
```

- z/OS z/OS で、979 ページの『mqweb サーバー用のプロシージャーの作成』で作成したプロシージャーを開始します。

  mqweb サーバーが正常に始動したことを示すために、以下のメッセージが STDOUT DD に発行されます。

```
[AUDIT   ] MQWB2019I: MQ Console level: 9.2.4 - V924-CD924-L211028
[AUDIT   ] MQWB0023I: MQ REST API level: 9.2.4 - V924-CD924-L211028
[AUDIT   ] CWWKZ0001I: Application com.ibm.mq.rest started in 1.763 seconds.
[AUDIT   ] CWWKZ0001I: Application com.ibm.mq.console started in 2.615 seconds.
[AUDIT   ] CWWKF0011I: The mqweb server is ready to run a smarter planet. The mqweb
server started in 10.016 seconds.
```

z/OS で mqweb サーバー開始タスクを停止するか、**endmqweb** コマンドを使用することにより、いつでも mqweb サーバーを停止できます。 ただし、mqweb サーバーが稼働していない場合は、REST API や IBM MQ Console を使用できません。

10. z/OS

オプション: z/OS では、IBM MQ Console および REST API の始動時に発行される MQWB2019I および MQWB0023I メッセージをシステム自動化製品がトラップできるようにするには、これらのメッセージを MVS コンソールに書き込むように mqweb サーバーを構成します。 MQWB2019I メッセージと MQWB0023I メッセージを MVS コンソールに書き込むように mqweb サーバーを構成するには、ステップ 823 ページの『4』 で作成した `mqwebuser.xml` ファイルを編集し、以下の行をファイルに追加します。

```
<zosLogging enableLogToMVS="true" wtoMessage="MQWB2019I,MQWB0023I"/>
```

mqweb サーバーでの z/OS ロギングの構成についての詳細情報は、z/OS ロギング (zosLogging) を参照してください。

次のタスク

1. HTTP 接続の使用可能化やポート番号の変更など、mqweb サーバー設定の構成を行います。 詳細については、821 ページの『IBM MQ Console および REST API の構成』を参照してください。
2. オプションで、REST API を構成します。
   a. REST API のクロス・オリジン・リソース共有を構成します。 デフォルトでは、REST API と同じドメイン上でホストされていない Web リソースからは REST API にアクセスできません。 つまり、クロス・オリジン要求が有効になりません。 指定した URL からのクロス・オリジン要求を許可するようクロス・オリジン・リソース共有 (CORS) を構成することができます。 詳しくは、REST API に対する CORS の構成 を参照してください。
   b. MFT 用に REST API を構成します。 詳しくは、846 ページの『REST API for MFT の構成』を参照してください。
3. REST API または IBM MQ Console を使用します。

- administrative REST API の概要
- messaging REST API の概要
- IBM MQ Console の概要

Linux V 9.3.5 スタンドアロン IBM MQ Web Server の構成

IBM MQ 9.3.5 以降では、 IBM MQ Console および REST API をホストする mqweb サーバーをスタンドアロンの IBM MQ Web Server インストール済み環境で実行できます。

始める前に

スタンドアロン IBM MQ Web Server は、 Linux でのみ使用可能です。

mqweb サーバーを構成する前に、 スタンドアロン IBM MQ Web Server のインストールの手順に従って、IBM MQ Web Server をインストールする必要があります。

このタスクについて

スタンドアロンの IBM MQ Web Server インストール済み環境で実行される新しい mqweb サーバーを作成して構成するには、このタスクの手順に従います。 この手順を繰り返すことで、スタンドアロン IBM MQ Web Server インストール済み環境で複数の mqweb サーバーを実行するように構成できます。

手順

1. IBM MQ Web Server データ・ディレクトリーを作成します。

   データ・ディレクトリーは、 IBM MQ Console および REST API を実行する mqweb サーバーの構成ファイルとログ・ファイルを保管するために使用されます。 IBM MQ Web Server データ・ディレクトリーとして選択した任意のディレクトリーを使用できます。

   mqweb サーバーの始動に使用するユーザー ID には、データ・ディレクトリーに対する読み取りおよび書き込み権限が付与されている必要があります。
2. **MQ_OVERRIDE_DATA_PATH** 環境変数を、ステップ 826 ページの『1』で作成したデータ・ディレクトリーに設定します。
   例えば、 IBM MQ Web Server データ・ディレクトリーとして /var/mqweb を使用することを選択した場合は、次のコマンドを発行します。

   ```
   export MQ_OVERRIDE_DATA_PATH=/var/mqweb
   ```
3. **setmqenv** コマンドを使用して、 IBM MQ 環境をセットアップします。

   IBM MQ Web Server インストール・ディレクトリーの bin ディレクトリーに移動して、以下のコマンドを発行します。

   ```
   . ./setmqenv -s
   ```
4. **crtmqdir** コマンドを使用して、データ・ディレクトリー内に IBM MQ ディレクトリーおよびファイルを作成します。 作成されるファイルには、mqweb サーバーのテンプレート定義が含まれます。

   以下のコマンドを発行します。

   ```
   crtmqdir -s -f
   ```
5. オプション: この mqweb サーバーが、スタンドアロン IBM MQ Web Server のこのインストールで実行するために最初に作成したものである場合は、 **mqlicense** コマンドを使用して、 IBM MQ ライセンスを確認し、同意します。

   このコマンドは、IBM MQ Web Server インストール・ディレクトリーに対する書き込み権限を持つユーザーとして実行する必要があります。

   例えば、 IBM MQ ライセンスを表示するには、次のコマンドを発行します。

   ```
   mqlicense
   ```

詳しくは、 mqlicense を参照してください。

6. オプション: 新しく構成されたスタンドアロン IBM MQ Web Server インストール済み環境で実行するために既存の mqweb サーバーをマイグレーションするには、以下の手順を実行します。

   a. 既存の mqweb サーバー構成をバックアップします。

   b. ファイルを *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST ディレクトリーにリストアします。ここで、 *MQ_OVERRIDE_DATA_PATH* は、ステップ 826 ページの『1』 で作成した IBM MQ Web Server データ・ディレクトリーです。

   詳細については、856 ページの『mqweb サーバーの構成のバックアップとリストア』 を参照してください。

   注 : IBM MQ Console および REST API の一部の機能は、スタンドアロンの IBM MQ Web Server インストール済み環境では使用できません。 mqweb サーバーを IBM MQ インストール済み環境からスタンドアロン IBM MQ Web Server インストール済み環境にマイグレーションする場合、マイグレーション後にこれらの機能を使用することはできません。 スタンドアロン IBM MQ Web Server インストール済み環境で適用される制約事項について詳しくは、 IBM MQ Console および REST API を参照してください。

次のタスク

822 ページの『mqweb サーバーの基本構成』 で説明されている手順に従って、mqweb サーバーを構成します。

セキュリティーの構成

mqwebuser.xml ファイルを編集することにより、IBM MQ Console および REST API のセキュリティーを構成することができます。 基本ユーザー・レジストリー、LDAP レジストリー、または WebSphere Liberty が備える他のレジストリー・タイプを構成することで、ユーザーを構成し、認証することができます。 そして、ユーザーとグループに役割を割り当てることで、それらのユーザーに権限を与えることができます。

このタスクについて

IBM MQ Console、および REST API のためのセキュリティーを構成するには、ユーザーとグループを構成する必要があります。 そして、それらのユーザーとグループに、IBM MQ Console を使用するための権限、または REST API、あるいはその両方を与えることができます。 ユーザーおよびグループの構成、およびユーザーの認証と許可について詳しくは、IBM MQ Console および REST API セキュリティーを参照してください。

ユーザーが IBM MQ Console で認証されると、LTPA トークンが生成されます。 このトークンにより、ユーザーは、トークンの有効期限が切れるまで再認証することなく IBM MQ Console を使用できます。

REST API でトークン・ベースの認証を使用する場合、HTTP POST メソッドで /login REST API リソースを使用してユーザーがログインする際に異なる LTPA トークンが生成されます。 このトークンの有効期限が切れるタイミングや、このトークンを HTTP 接続と HTTPS 接続の両方に使用できるかどうかを構成できます。 詳しくは、 835 ページの『LTPA トークンの構成』 を参照してください。

手順

- IBM MQ Console および REST API セキュリティー
- 835 ページの『LTPA トークンの構成』

HTTP ホスト名の構成

デフォルトでは、IBM MQ Console および REST API をホストする mqweb サーバーは、ローカル接続のみを許可するように構成されます。 つまり、IBM MQ Console と REST API は、IBM MQ Console と REST API がインストールされているシステム上でのみ利用できます。 **setmqweb** コマンドを使用することによって、リモート接続を許可するようにホスト名を構成できます。

始める前に

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、mqwebuser.xml ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。
- V 9.3.5 Linux mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、IBM MQ Web Server データ・ディレクトリー内の mqwebuser.xml ファイルに対する書き込み権限が必要です。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、crtmqweb に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

重要 : V 9.3.5 Linux

スタンドアロン IBM MQ Web Server インストール済み環境で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、**MQ_OVERRIDE_DATA_PATH** 環境変数を IBM MQ Web Server データ・ディレクトリーに設定する必要があります。

手順

- HTTP ホスト名の現在の構成を表示するには、以下のコマンドを使用します。

  ```
  dspmqweb properties -a
  ```

  httpHost フィールドには、HTTP ホスト名が表示されます。
- 以下のコマンドを使用して、HTTP ホスト名を設定します。

  ```
  setmqweb properties -k httpHost -v hostName
  ```

  hostName は IP アドレス、ドメイン名サフィックス付きのドメイン・ネーム・サーバー (DNS) ホスト名、または IBM MQ がインストールされているサーバーの DNS ホスト名を示します。 使用可能なすべてのネットワーク・インターフェースを指定するには、アスタリスクを二重引用符で囲んで使用します。 ローカル接続のみを許可するには、値 localhost を使用します。
- 以下のコマンドを使用して、HTTP ホスト名を設定解除します。

  ```
  setmqweb properties -k httpHost -d
  ```

HTTP および HTTPS ポートの構成

デフォルトでは、IBM MQ Console と REST API をホストする mqweb サーバーは HTTPS ポート 9443 を使用します。 HTTP 接続に関連付けられたポートは無効になります。 HTTP ポートを有効にしたり、別の HTTPS ポートを構成したり、HTTP や HTTPS ポートを無効にしたりできます。 **setmqweb** コマンドを使用して、ポートを構成できます。

始める前に

HTTP ポートを有効にすると、トークン・ベースの認証を使用している場合に、HTTP 接続と HTTPS 接続の両方で同じ LTPA トークンを使用可能にする必要があります。 詳細については、835 ページの『LTPA トークンの構成』を参照してください。

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、`mqwebuser.xml` ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。
- V 9.3.5 Linux mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、IBM MQ Web Server データ・ディレクトリー内の `mqwebuser.xml` ファイルに対する書き込み権限が必要です。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、`crtmqweb` に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

重要 : V 9.3.5 Linux

スタンドアロン IBM MQ Web Server インストール済み環境で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、**MQ_OVERRIDE_DATA_PATH** 環境変数を IBM MQ Web Server データ・ディレクトリーに設定する必要があります。

重要 : デフォルトでは、mqweb サーバーでは、すべての要求に対して LTPA トークンを保護する必要があります。 mqweb サーバーが LTPA トークンの保護を必要とするように構成されている場合、HTTP ポートに接続するときに以下のアクションを実行することはできません。

- IBM MQ Console にログインします。
- REST API でトークン・ベースの認証を使用します。

HTTP 要求で LTPA トークンを使用できるようにするには、 **secureLTPA** プロパティーの値を `false` に設定します。 詳しくは、835 ページの『LTPA トークンの構成』を参照してください。

手順

- HTTP および HTTPS ポートの現在の構成を表示するには、以下のコマンドを使用します。

  ```
  dspmqweb properties -a
  ```

  `httpPort` フィールドは HTTP ポートを示し、`httpsPort` フィールドは HTTPS ポートを示します。
- 以下のコマンドを使用して、HTTP ポートを有効にするか構成します。
  - 以下のコマンドを使用して、HTTP ポートを有効にするか設定します。

    ```
    setmqweb properties -k httpPort -v portNumber
    ```

ここで、*portNumber* は HTTP 接続に使用するポートを示します。 -1 の値を使用すると、ポートを無効にすることができます。

- 以下のコマンドを使用して、HTTP ポート値をデフォルト値の -1 にリセットします。

  ```
  setmqweb properties -k httpPort -d
  ```

- HTTPS ポートを構成します。
  - 以下のコマンドを使用して、HTTPS ポート番号を設定します。

    ```
    setmqweb properties -k httpsPort -v portNumber
    ```

    ここで、*portNumber* は HTTPS 接続に使用するポートを示します。 -1 の値を使用すると、ポートを無効にすることができます。
  - 以下のコマンドを使用して、HTTPS ポート番号をデフォルト値の 9443 にリセットします。

    ```
    setmqweb properties -k httpsPort -d
    ```

応答タイムアウトの構成

デフォルトでは、IBM MQ Console および REST API は、30 秒以内にクライアントに応答を送り返さないとタイムアウトになります。 **setmqweb** コマンドを使用して、別のタイムアウト値を使用するように IBM MQ Console および REST API を構成できます。

始める前に

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、mqwebuser.xml ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。
- V 9.3.5 Linux mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、 IBM MQ Web Server データ・ディレクトリー内の mqwebuser.xml ファイルに対する書き込み権限が必要です。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、crtmqweb に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

重要 : V 9.3.5 Linux

スタンドアロン IBM MQ Web Server インストール済み環境で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、**MQ_OVERRIDE_DATA_PATH** 環境変数を IBM MQ Web Server データ・ディレクトリーに設定する必要があります。

手順

- 要求タイムアウトの現在の構成を表示するには、以下のコマンドを使用します。

  ```
  dspmqweb properties -a
  ```

  mqRestRequestTimeout フィールドは、応答タイムアウトの現行値を示します。 詳しくは、dspmqweb プロパティーを参照してください。
- 以下のコマンドを使用して、要求タイムアウトを設定します。

  ```
  setmqweb properties -k mqRestRequestTimeout -v timeout
  ```

  timeout はタイムアウトになるまでの時間 (秒単位) です。
- 以下のコマンドを使用して、要求タイムアウトをデフォルト値の 30 秒にリセットします。

  ```
  setmqweb properties -k mqRestRequestTimeout -d
  ```

自動始動の構成

デフォルトでは、mqweb サーバーを始動すると、IBM MQ Console が自動的に始動します。 **setmqweb** コマンドを使用して、 IBM MQ Console と REST API を自動的に開始するかどうかを構成できます。

始める前に

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、mqwebuser.xml ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。
- V 9.3.5 Linux mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、 IBM MQ Web Server データ・ディレクトリー内の mqwebuser.xml ファイルに対する書き込み権限が必要です。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、crtmqweb に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

重要 : V 9.3.5 Linux

スタンドアロン IBM MQ Web Server インストール済み環境で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、**MQ_OVERRIDE_DATA_PATH** 環境変数を IBM MQ Web Server データ・ディレクトリーに設定する必要があります。

手順

- 以下のコマンドを使用して、自動始動の現在の構成を表示します。

  ```
  dspmqweb properties -a
  ```

mqRestAutostart フィールドは REST API が自動的に開始されるかどうかを示し、mqConsoleAutostart フィールドは IBM MQ Console が自動的に開始されるかどうかを示します。

- 以下のコマンドを使用して、IBM MQ Console を自動的に開始するかどうかを構成します。

  ```
  setmqweb properties -k mqConsoleAutostart -v start
  ```

  ここで、 *start* は、 IBM MQ Console を自動的に開始する場合は値 `true` 、それ以外の場合は `false` です。
- 以下のコマンドを使用して、REST API を自動的に開始するかどうかを構成します。

  ```
  setmqweb properties -k mqRestAutostart -v start
  ```

  ここで、 *start* は、 REST API を自動的に開始する場合は値 `true` 、それ以外の場合は `false` です。

ロギングの構成

ロギング・レベル、最大ログ・ファイル・サイズ、および IBM MQ Console と REST API をホストする mqweb サーバーで使用されるログ・ファイルの最大数を構成できます。 **setmqweb** コマンドを使用して、ロギングを構成できます。

始める前に

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、`mqwebuser.xml` ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。
- V 9.3.5 Linux mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、IBM MQ Web Server データ・ディレクトリー内の `mqwebuser.xml` ファイルに対する書き込み権限が必要です。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、`crtmqweb` に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

重要 : V 9.3.5 Linux

スタンドアロン IBM MQ Web Server インストール済み環境で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、**MQ_OVERRIDE_DATA_PATH** 環境変数を IBM MQ Web Server データ・ディレクトリーに設定する必要があります。

このタスクについて

mqweb サーバーは、ログ・メッセージとトレースを以下のログ・ファイルに書き込みます。

console.log と messages.log
これらのファイルには、 IBM MQ Console、 REST API、およびこれらのコンポーネントを実行する mqweb サーバーによって発行されるメッセージが含まれています。

trace.log
このファイルには、IBM MQ Console および REST API のトレースが含まれています。 トレースが有効になっている場合にのみ、トレースがこのファイルに書き込まれます。

mqweb サーバーのログ・ファイルは、次のいずれかのディレクトリーにあります。

- IBM MQ インストール済み環境の場合:
  - Linux AIX AIX または Linux の場合: `/var/mqm/web/installations/installationName/servers/mqweb/logs`
  - Windows Windows の場合: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb\logs。ここで、*MQ_DATA_PATH* は IBM MQ データ・パスです。 このパスは、IBM MQ のインストール時に選択されたデータ・パスです。 デフォルトでは、このパスは C:\ProgramData\IBM\MQ です。
  - z/OS z/OS 上: *WLP_user_directory*/servers/mqweb/logs

    ここで、*WLP_user_directory* は、mqweb サーバー定義を作成するために **crtmqweb** スクリプトを実行したときに指定したディレクトリーです。
- Linux V 9.3.5 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb/logs

  ここで、 *MQ_OVERRIDE_DATA_PATH* は、 **MQ_OVERRIDE_DATA_PATH** 環境変数が指す IBM MQ Web Server データ・ディレクトリーです。

mqweb サーバーで実行されるメッセージング REST API コードのメッセージング・トレース・ファイルは、以下のいずれかのディレクトリーにあります。

- IBM MQ インストール済み環境の場合:
  - Linux AIX AIX または Linux の場合: `/var/mqm/web/installations/installationName/servers/mqweb`
  - Windows Windows の場合: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb。ここで、*MQ_DATA_PATH* は IBM MQ データ・パスです。 このパスは、IBM MQ のインストール時に選択されたデータ・パスです。 デフォルトでは、このパスは C:\ProgramData\IBM\MQ です。
  - z/OS z/OS 上: *WLP_user_directory*/servers/mqweb

    ここで、*WLP_user_directory* は、mqweb サーバー定義を作成するために **crtmqweb** スクリプトを実行したときに指定したディレクトリーです。
- Linux V 9.3.5 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

  ここで、 *MQ_OVERRIDE_DATA_PATH* は、 **MQ_OVERRIDE_DATA_PATH** 環境変数が指す IBM MQ Web Server データ・ディレクトリーです。

以下のそれぞれで、トレースを有効にする方法について詳しくは、該当する資料を参照してください。

- REST API の場合は REST API のトレース
- IBM MQ Console の場合は IBM MQ Console のトレース

手順

- 以下のコマンドを使用して、REST API ロギングの現在の構成を表示します。

  ```
  dspmqweb properties -a
  ```

  - maxTraceFileSize フィールドには、最大ログ・ファイル・サイズが表示されます。
  - maxTraceFiles フィールドには、ログ・ファイルの最大数が表示されます。
  - traceSpec フィールドは、使用されているトレースのレベルを示します。
  - maxMsgTraceFileSize フィールドには、メッセージング・トレース・ファイルの最大サイズが表示されます。
  - 「maxMsgTraceFiles」フィールドには、メッセージング・トレース・ファイルの最大数が表示されます。
- messages.log ファイルおよび trace.log ファイルの最大サイズを構成します。
  - 以下のコマンドを使用して、ログ・ファイルの最大サイズを設定します。

    ```
    setmqweb properties -k maxTraceFileSize -v size
    ```

    ここで、*size* は各ログ・ファイルの上限サイズを MB で示します。
  - 以下のコマンドを使用して、ログ・ファイルの最大サイズをデフォルト値の 20 MB にリセットします。

    ```
    setmqweb properties -k maxTraceFileSize -d
    ```
- messages.log ファイルと trace.log ファイルの最大数を構成します。
  - 以下のコマンドを使用して、各ログ・ファイルの最大数を設定します。

    ```
    setmqweb properties -k maxTraceFiles -v max
    ```

    ここで、 *max* はファイルの最大数を示します。
  - 次のコマンドを使用して、各ログ・ファイルの最大数をデフォルト値の 2 にリセットします。

    ```
    setmqweb properties -k maxTraceFiles -d
    ```
- 最大メッセージング・トレース・ファイル・サイズを構成します。
  - 以下のコマンドを使用して、最大メッセージング・トレース・ファイル・サイズを設定します。

    ```
    setmqweb properties -k maxMsgTraceFileSize -v size
    ```

    ここで、*size* は、各メッセージング・トレース・ファイルが到達できるサイズ (MB 単位) を指定します。
  - 以下のコマンドを使用して、最大メッセージング・トレース・ファイル・サイズをデフォルト値の 200 MB にリセットします。

    ```
    setmqweb properties -k maxMsgTraceFileSize -d
    ```
- 使用するメッセージング・トレース・ファイルの最大数を構成します。
  - 以下のコマンドを使用して、メッセージング・トレースに使用するファイルの最大数を設定します。

    ```
    setmqweb properties -k maxMsgTraceFiles -v max
    ```

    ここで、 *max* はファイルの最大数を示します。
  - 以下のコマンドを使用して、メッセージング・トレースに使用するファイルの最大数をデフォルト値の 5 にリセットします。

    ```
    setmqweb properties -k maxMsgTraceFiles -d
    ```
- mqweb サーバーが書き込むトレースのレベルを構成します。
  - 以下のコマンドを使用して、使用するトレース仕様を設定します。

    ```
    setmqweb properties -k traceSpec -v level
    ```

    ここで、 *level* は、 835 ページの表 68 にリストされているいずれかの値です。 この表では、ロギング・レベルの概要を、詳細度の高い順に示します。 あるロギング・レベルを有効にすると、それより前のすべてのレベルも有効になります。 例えば、***=warning** ロギング・レベルを有効にすると、***=severe** および ***=fatal** ロギング・レベルも有効になります。

この値は、 IBM サポートから要求された場合に変更してください。

- 以下のコマンドを使用して、使用するトレース仕様をデフォルト値の ***=info** にリセットします。

  ```
  setmqweb properties -k traceSpec -d
  ```

表 68. 有効なロギング・レベル

値	適用されるロギング・レベル
*=off	ロギングはオフ。
*=fatal	タスクが続行できず、コンポーネント、アプリケーション、およびサーバーが機能できない。
*=severe	タスクが続行できない。ただし、コンポーネント、アプリケーション、およびサーバーはまだ機能できる。 このレベルは、リカバリー不能エラーが差し迫っていることを示すこともあります。
* = 警告	潜在的なエラー、または差し迫ったエラー。 このレベルは、障害の進行 (例えば、潜在的なリソース・リーク) を示すこともあります。
*=audit	サーバー状態またはリソースに影響を与える重大なイベント。
*=info	タスクの全体的な進行状況を大まかに示す一般情報
*=config	構成の変更または状況
*=detail	サブタスクの進行の詳細を示す一般情報
*=fine	トレース情報 - 一般トレース + メソッドの開始、終了、戻り値
*=finer	トレース情報 - 詳細トレース
*=finest	トレース情報 - より詳細なトレース。問題をデバッグするために必要な詳細がすべて含まれる
* = すべて	すべてのイベントをログに記録する

LTPA トークンの構成

LTPA トークンを使用すると、ユーザーは mqweb サーバーへの要求のたびにユーザー名とパスワードの資格情報を入力する必要がなくなります。 **setmqweb** コマンドを使用して、LTPA トークン Cookie の名前、LTPA 認証トークンの有効期限間隔、HTTP 接続で LTPA トークンを使用できるかどうかを構成できます。

始める前に

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、`mqwebuser.xml` ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。
- V 9.3.5 Linux mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、 IBM MQ Web Server データ・ディレクトリー内の `mqwebuser.xml` ファイルに対する書き込み権限が必要です。

注 : IBM MQ Console と REST API を使用したトークン認証の両方を使用している場合、有効期限間隔が共有されます。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、crtmqweb に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

重要 : V 9.3.5 Linux

スタンドアロン IBM MQ Web Server インストール済み環境で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、**MQ_OVERRIDE_DATA_PATH** 環境変数を IBM MQ Web Server データ・ディレクトリーに設定する必要があります。

このタスクについて

ユーザーが IBM MQ Console にログインすると、LTPA トークンが生成されます。 REST API でトークン・ベースの認証を使用する場合、HTTP POST メソッドで /login REST API リソースを使用してユーザーがログインする際に LTPA トークンが生成されます。 このトークンは、Cookie で返されます。 ユーザーの認証にトークンが使用されます。そのトークンの有効期限が切れるまでは、ユーザーはユーザー ID とパスワードを使用して再びログインする必要はありません。 デフォルトの有効期間は 120 分です。

LTPA トークンが含まれる Cookie の名前は、プラットフォームによって異なります。

- MQ Appliance IBM MQ Appliance では、LTPA トークンは LtpaToken2 です。 この値は変更できません。
- z/OS ALW 他のすべてのプラットフォームでは、LTPA トークンが含まれる Cookie の名前のデフォルトは、先頭が LtpaToken2 で、mqweb サーバーの再始動時に変更される可能性がある接尾部が付きます。 このように Cookie 名がランダム化されているので、複数の mqweb サーバーを同一のシステム上で実行できます。 しかし、Cookie 名を一定の値にしておく場合は、**setmqweb** コマンドを使用して Cookie の名前を指定することができます。

z/OS IBM i ALW HTTP ポートと HTTPS ポートの両方を有効にする場合、HTTPS 要求に対して発行される LTPA トークンを、HTTP 要求で再利用できます。 デフォルトではこの動作は無効ですが、**setmqweb** コマンドを使用してこの動作を有効にすることができます。

手順

- 以下のコマンドを使用して、LTPA トークンの現在の有効期限、LTPA トークン Cookie の名前、HTTP 要求で LTPA トークンを使用できるかどうかを表示します。

  ```
  dspmqweb properties -a
  ```

  - ltpaCookieName フィールドに LTPA トークン Cookie の名前が表示されます。 Cookie 名を設定していない場合、このプロパティーの値は LtpaToken2_${env.MQWEB_LTPA_SUFFIX} on AIX, Linux, and Windows 、または z/OS 上の LtpaToken2_${httpsPort} 、 です。 mqweb サーバーは、この接頭部 LtpaToken2_ の後の変数を使用して、固有の Cookie 名を生成します。 この変数は設定できませんが、ltpaCookieName は任意の値に変更できます。
  - ltpaExpiration フィールドには、LTPA トークンの有効期限時刻が示されます。
  - HTTP 要求で LTPA トークンを使用できる場合、secureLtpa フィールドは false に設定されます。

- LTPA トークンの有効期限を構成します。
  - 以下のコマンドを入力して、LTPA トークンの有効期限を設定します。

    ```
    setmqweb properties -k ltpaExpiration -v time
    ```

    ここで、*time* は、LTPA トークンの有効期限が切れてユーザーがログアウトされるまでの時間を分数で指定します。
  - 以下のコマンドを入力して、LTPA トークンの有効期限をデフォルト値の 120 分にリセットします。

    ```
    setmqweb properties -k ltpaExpiration -d
    ```
- z/OS ALW

  LTPA トークン Cookie 名を構成します。
  - 以下のコマンドを入力して、LTPA トークン Cookie 名を設定します。

    ```
    setmqweb properties -k ltpaCookieName -v name
    ```

    ここで、*name* は LTPA トークン Cookie の固有の名前を指定します。
  - 以下のコマンドを入力して、LTPA トークン Cookie 名をデフォルトにリセットします。デフォルトでは、接頭部 `LtpaToken2_` の後にランダム文字が付きます。

    ```
    setmqweb properties -k ltpaCookieName -d
    ```
- z/OS ALW

  以下のコマンドを入力して、HTTP 接続で LTPA トークンを使用できるかどうかを構成します。

  ```
  setmqweb properties -k secureLtpa -v secure
  ```

  ここで、*secure* は LTPA トークンを安全でない HTTP 接続と安全な HTTPS 接続の両方で使用できるかどうかを指定します。 `false` の値を設定すると、HTTP 接続と HTTPS 接続の両方で同じ LTPA トークンを使用できるようになります。

V 9.3.0 IBM MQ Console のリモート・キュー・マネージャー接続の動作の構成

IBM MQ Console の使用時に、リモート・キュー・マネージャーへの接続を作成できます。 つまり、IBM MQ Console を実行する mqweb サーバーと同じインストール済み環境に含まれないキュー・マネージャーに接続することができます。 リモート・キュー・マネージャー接続の動作を制御するために設定できる複数の構成オプションがあります。

始める前に

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、`mqwebuser.xml` ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。
- V 9.3.5 Linux mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、IBM MQ Web Server データ・ディレクトリー内の `mqwebuser.xml` ファイルに対する書き込み権限が必要です。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、crtmqweb に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

重要： V 9.3.5 Linux

スタンドアロン IBM MQ Web Server インストール済み環境で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、**MQ_OVERRIDE_DATA_PATH** 環境変数を IBM MQ Web Server データ・ディレクトリーに設定する必要があります。

このタスクについて

以下の構成オプションを設定できます。

- リモート・キュー・マネージャー接続を許可するかどうか。
- IBM MQ Console を使用して接続を追加できるか、またはコマンド行でのみ接続を追加できるか。
- リモート・キュー・マネージャー接続が許可されているときに、ローカル・キュー・マネージャーが IBM MQ Console に表示されるかどうか。
- IBM MQ Console の開始時、または接続障害が発生したときに、リモート・キュー・マネージャー接続が自動的に確立されるかどうか。
- IBM MQ Console に表示されるリモート・キュー・マネージャーのリストを最新表示する時間間隔。

手順

- 現在のリモート・キュー・マネージャー接続の構成設定を表示するには、以下のコマンドを入力します。

  ```
  dspmqweb properties -a
  ```

  - mqConsoleRemoteSupportEnabled フィールドは、リモート・キュー・マネージャー接続が許可されるかどうかを示します。
  - mqConsoleRemoteUIAdmin フィールドは、IBM MQ Console を使用してリモート・キュー・マネージャー接続を追加できるかどうかを示します。
  - mqConsoleRemoteAllowLocal フィールドは、ローカル・キュー・マネージャーが表示されるかどうかを示します。
  - mqConsoleRemotePollTime フィールドは、リモート・キュー・マネージャーのリストをリフレッシュする間隔を秒数で示します。
- IBM MQ Console を使用したリモート・キュー・マネージャー接続を防止または許可するには、以下のコマンドを入力します。

  ```
  setmqweb properties -k mqConsoleRemoteSupportEnabled -v true or false
  ```

  ここで、 *true* はリモート・キュー・マネージャー接続を許可し、 false はリモート・キュー・マネージャー接続を禁止します。

  注： Linux V 9.3.5 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境で実行されている場合、 **mqConsoleRemoteSupportEnabled** プロパティーは無効です。スタンドアロン IBM MQ Web Server は、リモート・キュー・マネージャーへの接続のみをサポートします。
- IBM MQ Console を使用したリモート・キュー・マネージャー接続の追加を防止または許可したり、コマンド行でしかリモート・キュー・マネージャー接続を追加できないようにしたりするには、以下のコマンドを入力します。

  ```
  setmqweb properties -k mqConsoleRemoteUIAdmin -v true or false
  ```

ここで、 true では、 IBM MQ Console およびコマンド行を使用してリモート・キュー・マネージャー接続を追加できます。また、 false では、コマンド行で **setmqweb remote** コマンドを使用することによってのみリモート・キュー・マネージャー接続を追加できます。

- リモート・キュー・マネージャー接続が許可されているときに、IBM MQ Console でローカル・キュー・マネージャーの表示を防止または許可するには、以下のコマンドを入力します。

  ```
  setmqweb properties -k mqConsoleRemoteAllowLocal -v true or false
  ```

  ここで、 true はローカル・キュー・マネージャーの表示を許可し、 false はローカル・キュー・マネージャーの表示を非表示にします。

  注 : Linux V 9.3.5 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境で実行されている場合、 **mqConsoleRemoteAllowLocal** プロパティーは無効です。 スタンドアロン IBM MQ Web Server は、リモート・キュー・マネージャーへの接続のみをサポートします。
- IBM MQ Console に表示されるリモート・キュー・マネージャーのリストを最新表示する時間間隔を設定するには、以下のコマンドを入力します。

  ```
  setmqweb properties -k mqConsoleRemotePollTime -v seconds
  ```

  ここで、*seconds* には、リモート・キュー・マネージャーのリストを最新表示する間隔 (秒数) の整数値を設定します。

関連資料

setmqweb

dspmqweb

administrative REST API ゲートウェイの構成

administrative REST API ゲートウェイを有効にすると、ゲートウェイ・キュー・マネージャーを使用して REST API によるリモート管理を実行できます。 デフォルトのゲートウェイ・キュー・マネージャーとして使用されるキュー・マネージャーを構成することも、 **setmqweb** コマンドを使用して administrative REST API ゲートウェイを無効にすることでリモート管理を防止することもできます。

始める前に

注 : Linux V 9.3.5 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境で実行される場合、このタスクは適用されません。 administrative REST API は、スタンドアロンの IBM MQ Web Server インストール済み環境では使用できません。

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、mqwebuser.xml ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、crtmqweb に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

このタスクについて

mqweb サーバーが IBM MQ インストール済み環境で実行されている場合、 administrative REST API ゲートウェイはデフォルトで有効になっています。

デフォルトのゲートウェイ・キュー・マネージャーは、以下の両方の記述が当てはまる場合に使用されます。

- REST 要求の `ibm-mq-rest-gateway-qmgr` ヘッダーでキュー・マネージャーが指定されていない。
- REST API リソース URL で指定されたキュー・マネージャーがローカル・キュー・マネージャーではない。

REST API を使用したリモート管理の詳細については、REST API によるリモート管理を参照してください。

手順

- administrative REST API ゲートウェイの現在の構成を表示するには、以下のコマンドを使用します。

  ```
  dspmqweb properties -a
  ```

  `mqRestGatewayEnabled` フィールドにゲートウェイが有効かどうかが表示され、`mqRestGatewayQmgr` フィールドにデフォルトのゲートウェイ・キュー・マネージャーの名前が表示されます。
- 以下のコマンドを使用して、administrative REST API ゲートウェイを有効にするかどうかを構成します。

  ```
  setmqweb properties -k mqRestGatewayEnabled -v enabled
  ```

  ここで、*enabled* は、値 **true** (administrative REST API ゲートウェイを有効にする場合) または **false** (それ以外の場合) です。
- どのキュー・マネージャーがデフォルトのゲートウェイ・キュー・マネージャーとして使用されるかを構成します。
  - 以下のコマンドを使用して、デフォルトのゲートウェイ・キュー・マネージャーを設定します。

    ```
    setmqweb properties -k mqRestGatewayQmgr -v qmgrName
    ```

    ここで、*qmgrName* は mqweb サーバーと同じインストールのキュー・マネージャーの名前です。
  - 以下のコマンドを使用して、デフォルトのゲートウェイ・キュー・マネージャーを設定解除します。

    ```
    setmqweb properties -k mqRestGatewayQmgr -d
    ```

messaging REST API の構成

messaging REST API は、いくつかの方法で構成できます。 messaging REST API 機能を有効にするか無効にするかを選択できます。 messaging REST API によって使用できるプール接続の最大数、およびすべての接続が使用中の場合の messaging REST API の動作を選択できます。 messaging REST API を使用してメッセージを送信、受信、参照、またはパブリッシュするときに、許可に使用するユーザー・コンテキストを選択することもできます。

手順

- 841 ページの『messaging REST API の有効化』
- 842 ページの『messaging REST API の接続プーリングの構成』
- V 9.3.2
  845 ページの『messaging REST API での許可に使用されるユーザー・コンテキストの構成』

messaging REST API の有効化

setmqweb コマンドを使用して、 messaging REST API を有効にするかどうかを構成できます。 デフォルトでは messaging REST API は使用可能です。

始める前に

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、mqwebuser.xml ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。
- V 9.3.5 Linux mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、 IBM MQ Web Server データ・ディレクトリー内の mqwebuser.xml ファイルに対する書き込み権限が必要です。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、crtmqweb に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

手順

- 以下のコマンドを使用して、 messaging REST API の現在の構成を表示します。

  ```
  dspmqweb properties -a
  ```

  mqRestMessagingEnabled フィールドは、messaging REST API が有効かどうかを示します。 値が True の場合、 messaging REST API が有効になります。
- 以下のコマンドを使用して、 messaging REST API を有効にします。

  ```
  setmqweb properties -k mqRestMessagingEnabled -v true
  ```
- 以下のコマンドを使用して、 messaging REST API を無効にします。

  ```
  setmqweb properties -k mqRestMessagingEnabled -v false
  ```

関連タスク

842 ページの『messaging REST API の接続プーリングの構成』
messaging REST API で使用できるプール接続の最大数、およびすべての接続が使用中の場合の messaging REST API の動作を構成することができます。

845 ページの『messaging REST API での許可に使用されるユーザー・コンテキストの構成』
V 9.3.2 messaging REST API を使用してメッセージを送信、受信、参照、またはパブリッシュするときに、許可に使用するユーザー・コンテキストを構成できます。 つまり、 messaging REST API にログインしているユーザーを許可に使用するか、mqweb サーバーを開始したユーザーを許可に使用するかを選択できます。

843 ページの『messaging REST API の接続モードの構成』
ローカル・キュー・マネージャーまたはリモート・キュー・マネージャーのいずれかに接続するように messaging REST API を構成できます。

messaging REST API の接続プーリングの構成

messaging REST API で使用できるプール接続の最大数、およびすべての接続が使用中の場合の messaging REST API の動作を構成することができます。

始める前に

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、`mqwebuser.xml` ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。
- V 9.3.5 Linux mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、 IBM MQ Web Server データ・ディレクトリー内の `mqwebuser.xml` ファイルに対する書き込み権限が必要です。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、`crtmqweb` に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

このタスクについて

messaging REST API のパフォーマンスを最適化するために、IBM MQ キュー・マネージャーへの接続はプールされます。 つまり、各 REST 要求は、独自の接続を作成、使用、および削除する代わりに、接続プールからの接続を使用します。 デフォルトでは、キュー・マネージャー・プールごとに 20 個の接続が使用可能であり、すべての接続が使用中の場合に要求を処理するための 3 つのオプションから選択できます。

- messaging REST API は、要求に使用する新しい非プール接続を作成できます。 この動作がデフォルトの動作です。
- messaging REST API がエラーを返します。
- messaging REST API が、プール内の接続が使用可能になるのを待機します。 待機時間は無制限です。

プール内の接続の最大数や、すべての接続が使用中になった場合の messaging REST API のデフォルト動作を変更する場合は、**setmqweb properties** コマンドを使用します。

手順

- 以下のコマンドを使用して、現在の構成を表示します。

```
dspmqweb properties -a
```

- `mqRestMessagingFullPoolBehavior` フィールドは、プール内のすべての接続が使用中の場合の messaging REST API の動作を示します。 値が `block` の場合、 messaging REST API は接続が使用可能になるまで待機する必要があります。 値が `error` の場合、 messaging REST API はエラーを返す必要があります。 値が `overflow` の場合、 messaging REST API は使用する非プール接続を作成し、使用後に接続を破棄する必要があります。
- `mqRestMessagingMaxPoolSize` フィールドには、接続プールの最大サイズが表示されます。

- 以下のコマンドを使用して、プール内のすべての接続が使用中になった場合の messaging REST API の動作を構成します。

```
setmqweb properties -k mqRestMessagingFullPoolBehavior -v action
```

action で、実行するアクションを指定します。 *action* には、次のいずれかの値を指定できます。

block
プール内のすべての接続が使用中の場合は、接続が使用可能になるまで待ちます。

エラー
プール内のすべての接続が使用中の場合は、エラーを返します。

overflow
プール内のすべての接続が使用中の場合は、使用する非プール接続を作成し、使用後にその接続を破棄します。

- 以下のコマンドを使用して、各キュー・マネージャー・プールの最大接続プール・サイズを構成します。

```
setmqweb properties -k mqRestMessagingMaxPoolSize -v size
```

size でプールのサイズを指定します。

注 : *mqRestMessagingMaxPoolSize* に大きな値が設定されていて、多数のキュー・マネージャーが接続されている場合は、mqweb サーバー・ヒープの最大サイズを増やすことを検討してください。 詳しくは、 mqweb サーバー JVM の調整 を参照してください。

関連タスク

841 ページの『messaging REST API の有効化』
setmqweb コマンドを使用して、 messaging REST API を有効にするかどうかを構成できます。 デフォルトでは messaging REST API は使用可能です。

845 ページの『messaging REST API での許可に使用されるユーザー・コンテキストの構成』
V 9.3.2 messaging REST API を使用してメッセージを送信、受信、参照、またはパブリッシュするときに、許可に使用するユーザー・コンテキストを構成できます。 つまり、 messaging REST API にログインしているユーザーを許可に使用するか、mqweb サーバーを開始したユーザーを許可に使用するかを選択できます。

843 ページの『messaging REST API の接続モードの構成』
ローカル・キュー・マネージャーまたはリモート・キュー・マネージャーのいずれかに接続するように messaging REST API を構成できます。

V 9.3.3 messaging REST API の接続モードの構成

ローカル・キュー・マネージャーまたはリモート・キュー・マネージャーのいずれかに接続するように messaging REST API を構成できます。

始める前に

注 : Linux V 9.3.5 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境で実行される場合、このタスクは適用されません。 スタンドアロン IBM MQ Web Server は、リモート・キュー・マネージャーへの接続のみをサポートします。

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、mqwebuser.xml ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、crtmqweb に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

このタスクについて

messaging REST API のデフォルト接続モードは、mqweb サーバーを実行するインストールのタイプによって異なります。

- IBM MQ インストール済み環境では、デフォルトで、 messaging REST API は mqweb サーバーと同じインストール済み環境内のローカル・キュー・マネージャーにのみ接続します。 接続構成を表示および変更するには、このタスクのステップを実行します。
- Linux V 9.3.5 スタンドアロンの IBM MQ Web Server インストール済み環境では、messaging REST API はリモート・キュー・マネージャーへの接続のみをサポートします。 接続構成を表示または変更できません。

手順

- 以下のコマンドを使用して、 messaging REST API の現在の構成を表示します。

  ```
  dspmqweb properties -a
  ```

  「mqRestMessagingConnectionMode」フィールドには、現行接続モードが表示されます。 値が local の場合、 messaging REST API は、mqweb サーバーと同じインストール済み環境のキュー・マネージャーにのみ接続できます。 値が remote の場合、 messaging REST API はリモート・キュー・マネージャーに接続できます。
- 以下のコマンドを使用して、mqweb サーバーと同じインストール済み環境にあるキュー・マネージャーにのみ messaging REST API が接続できるように mqweb サーバーを構成します。

  ```
  setmqweb properties -k mqRestMessagingConnectionMode -v local
  endmqweb
  strmqweb
  ```
- 以下のコマンドを使用して、 messaging REST API がリモート・キュー・マネージャーに接続できるように mqweb サーバーを構成します。

  ```
  setmqweb properties -k mqRestMessagingConnectionMode -v remote
  endmqweb
  strmqweb
  ```

次のタスク

messaging REST API がリモート・キュー・マネージャーに接続できるように mqweb サーバーを構成する場合は、接続先のキュー・マネージャーごとに接続情報を指定する必要があります。 接続情報を指定する

方法について詳しくは、 messaging REST API で使用するリモート・キュー・マネージャーのセットアップを参照してください。

関連タスク

841 ページの『messaging REST API の有効化』

setmqweb コマンドを使用して、 messaging REST API を有効にするかどうかを構成できます。 デフォルトでは messaging REST API は使用可能です。

842 ページの『messaging REST API の接続プーリングの構成』

messaging REST API で使用できるプール接続の最大数、およびすべての接続が使用中の場合の messaging REST API の動作を構成することができます。

845 ページの『messaging REST API での許可に使用されるユーザー・コンテキストの構成』

V 9.3.2 messaging REST API を使用してメッセージを送信、受信、参照、またはパブリッシュするときに、許可に使用するユーザー・コンテキストを構成できます。 つまり、 messaging REST API にログインしているユーザーを許可に使用するか、mqweb サーバーを開始したユーザーを許可に使用するかを選択できます。

V 9.3.2 messaging REST API での許可に使用されるユーザー・コンテキストの構成

V 9.3.2 messaging REST API を使用してメッセージを送信、受信、参照、またはパブリッシュするときに、許可に使用するユーザー・コンテキストを構成できます。 つまり、 messaging REST API にログインしているユーザーを許可に使用するか、mqweb サーバーを開始したユーザーを許可に使用するかを選択できます。

始める前に

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、`mqwebuser.xml` ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。
- V 9.3.5 Linux mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、 IBM MQ Web Server データ・ディレクトリー内の `mqwebuser.xml` ファイルに対する書き込み権限が必要です。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、`crtmqweb` に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

このタスクについて

- 使用されるユーザー ID が、 messaging REST API にログインしているユーザー ID である場合、**MQMD.UserIdentifier** は、REST API にログインしているユーザー ID に設定されます。 **MQMD.AppIdentityData** は、REST API にログインしているユーザー ID に設定されます。

- 使用されるユーザー ID が mqweb サーバーを開始したユーザー ID である場合、**MQMD.UserIdentifier** はブランクのままになります。 **MQMD.AppIdentityData** は、REST API にログインしているユーザー ID に設定されます。

IBM MQ メッセージのメッセージ記述子部分について詳しくは、 MQMD を参照してください。

手順

- 以下のコマンドを使用して、 messaging REST API の現在の構成を表示します。

  ```
  dspmqweb properties -a
  ```

  mqRestMessagingAdoptWebUserContext フィールドは、メッセージを送信、パブリッシュ、受信、または参照するときに許可に使用されるユーザー ID を示します。 値が True の場合、messaging REST API にログインしているユーザーが許可に使用されます。 値が False の場合、mqweb サーバーを開始したユーザーが許可に使用されます。
- 以下のコマンドを使用して、messaging REST API にログインしているユーザーのユーザー ID を許可に使用するように messaging REST API を構成します。

  ```
  setmqweb properties -k mqRestMessagingAdoptWebUserContext -v true
  ```

  mqRestMessagingAdoptWebUserContext が true に設定されている場合、**MQMD.UserIdentifier** は、REST API にログインしているユーザー ID に設定されます。**MQMD.AppIdentityData** は、REST API にログインしているユーザー ID に設定されます。
- 以下のコマンドを使用して、mqweb サーバーを開始したユーザーのユーザー ID を使用するように messaging REST API を構成します。

  ```
  setmqweb properties -k mqRestMessagingAdoptWebUserContext -v false
  ```

  mqRestMessagingAdoptWebUserContext が false に設定されている場合、**MQMD.UserIdentifier** はブランクのままになります。 **MQMD.AppIdentityData** は、REST API にログインしているユーザー ID に設定されます。

関連タスク

841 ページの『messaging REST API の有効化』
setmqweb コマンドを使用して、 messaging REST API を有効にするかどうかを構成できます。 デフォルトでは messaging REST API は使用可能です。

842 ページの『messaging REST API の接続プーリングの構成』
messaging REST API で使用できるプール接続の最大数、およびすべての接続が使用中の場合の messaging REST API の動作を構成することができます。

843 ページの『messaging REST API の接続モードの構成』
ローカル・キュー・マネージャーまたはリモート・キュー・マネージャーのいずれかに接続するように messaging REST API を構成できます。

REST API for MFT の構成

デフォルトでは、REST API for MFT は有効になっていません。 **setmqweb properties** コマンドを使用して、 REST API for MFT を有効にするかどうか、調整キュー・マネージャーを設定するかどうか、コマンド・キュー・マネージャーを設定するかどうか、および MFT 再接続タイムアウトを指定するかどうかを構成できます。

手順

- 847 ページの『MFT での REST API の有効化』
- 848 ページの『REST API for MFT 用の調整キュー・マネージャーの構成』
- 849 ページの『REST API for MFT 用のコマンド・キュー・マネージャーの構成』
- 850 ページの『MFT タイムアウト値の REST API の構成』

MFT での REST API の有効化

REST API for MFT を使用するには、まず REST API for MFT を有効にする必要があります。 **setmqweb** コマンドを使用して、 REST API for MFT を有効にするかどうかを構成できます。 デフォルトでは、REST API for MFT は有効になっていません。

始める前に

注 : Linux V 9.3.5 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境で実行される場合、このタスクは適用されません。 REST API for MFT は、スタンドアロンの IBM MQ Web Server インストール済み環境では使用できません。

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、`mqwebuser.xml` ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、`crtmqweb` に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

手順

1. 以下のコマンドを使用して、REST API for MFT の現在の構成を表示します。

   ```
   dspmqweb properties -a
   ```

   `mqRestMftEnabled` フィールドは、REST API for MFT が有効であるかどうかを示します。 値は、MFT の REST API が有効な場合は `True` 、それ以外の場合は `False` です。
2. 以下のいずれかのコマンドを使用して、 MFT の REST API を有効または無効にします。
   - 以下のコマンドを使用して、 MFT の REST API を有効にします。

     ```
     setmqweb properties -k mqRestMftEnabled -v true
     ```
   - 以下のコマンドを使用して、 MFT の REST API を無効にします。

     ```
     setmqweb properties -k mqRestMftEnabled -v false
     ```
3. 以下のコマンドを入力して、mqweb サーバーを再始動します。

   ```
   endmqweb
   strmqweb
   ```

次のタスク

MFT に対して REST API を有効にした場合、 MFT に対して REST API を使用する前に調整キュー・マネージャー名を設定する必要があります。 調整キュー・マネージャーの設定方法について詳しくは、 848 ページの『REST API for MFT 用の調整キュー・マネージャーの構成』を参照してください。

REST API for MFT 用の調整キュー・マネージャーの構成

REST API for MFT を使用する前に、 MFT トランザクションの調整キュー・マネージャーとして機能するようにキュー・マネージャーを構成する必要があります。 **setmqweb** コマンドを使用して、どのキュー・マネージャーを調整キュー・マネージャーにするかを設定できます。

始める前に

注 : Linux V 9.3.5 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境で実行される場合、このタスクは適用されません。 REST API for MFT は、スタンドアロンの IBM MQ Web Server インストール済み環境では使用できません。

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、`mqwebuser.xml` ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、`crtmqweb` に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

手順

1. 以下のコマンドを使用して、REST API for MFT の現在の構成を表示します。

   ```
   dspmqweb properties -a
   ```

   `mqRestMftCoordinationQmgr` フィールドに調整キュー・マネージャーの名前が表示されます。
2. 以下のコマンドを使用して、調整キュー・マネージャーを構成します。

   ```
   setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
   ```

   ここで、*qmgrName* は、調整キュー・マネージャーの名前です。 調整キュー・マネージャーは、mqweb サーバーが稼働しているマシンに配置する必要があります。 デフォルトでは、このキュー・マネージャー名はブランクです。 値を設定しないと、REST API for MFT は機能しません。
3. 以下のコマンドを入力して、mqweb サーバーを再始動します。

   ```
   endmqweb
   strmqweb
   ```

次のタスク

- REST API for MFT が有効になっていることを確認します。 詳しくは、847 ページの『MFT での REST API の有効化』を参照してください。
- REST API for MFT を使用して作成要求を実行依頼する場合は、コマンド・キュー・マネージャー名を設定する必要があります。 例えば、 **create transfer** などの REST API コマンドを使用する場合は、コマンド・キュー・マネージャー名を設定する必要があります。 詳しくは、849 ページの『REST API for MFT 用のコマンド・キュー・マネージャーの構成』を参照してください。
- REST API for MFT のタイムアウト値を構成できます。 デフォルトのタイムアウトは 30 分です。 詳しくは、850 ページの『MFT タイムアウト値の REST API の構成』を参照してください。
- REST API for MFT を使用するには、ユーザーが mqweb サーバーで認証され、1 つ以上の MFTWebAdmin 役割または MFTWebAdminRO 役割のメンバーでなければなりません。 ユーザーの構成について詳しくは、 REST API のユーザーおよび役割の構成を参照してください。

REST API for MFT 用のコマンド・キュー・マネージャーの構成

REST API for MFT を使用して作成要求を実行依頼する前に、コマンド・キュー・マネージャー名を設定する必要があります。 例えば、 **create transfer** リソースを使用するには、コマンド・キュー・マネージャー名を設定する必要があります。 コマンド・キュー・マネージャー名は、 **setmqweb** コマンドを使用して設定できます。

始める前に

注 : Linux V 9.3.5 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境で実行される場合、このタスクは適用されません。 REST API for MFT は、スタンドアロンの IBM MQ Web Server インストール済み環境では使用できません。

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、mqwebuser.xml ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、crtmqweb に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

手順

1. 以下のコマンドを使用して、REST API for MFT の現在の構成を表示します。

```
dspmqweb properties -a
```

   mqRestMftCommandQmgr フィールドにコマンド・キュー・マネージャーの名前が表示されます。
2. 以下のコマンドを使用して、コマンド・キュー・マネージャーを構成します。

```
setmqweb properties -k mqRestMftCommandQmgr -v qmgrName
```

ここで、*qmgrName* は、コマンド・キュー・マネージャーの名前です。 コマンド・キュー・マネージャーは、mqweb サーバーが稼働しているマシンに配置する必要があります。 デフォルトでは、このキュー・マネージャー名はブランクです。 値が設定されていない場合、作成コマンドの MFT の REST API は機能しません。

3. 以下のコマンドを入力して、mqweb サーバーを再始動します。

```
endmqweb
strmqweb
```

次のタスク

- REST API for MFT が有効になっていることを確認します。 詳しくは、847 ページの『MFT での REST API の有効化』を参照してください。
- 調整キュー・マネージャーが設定されていることを確認します。 詳細については、848 ページの『REST API for MFT 用の調整キュー・マネージャーの構成』を参照してください。
- REST API for MFT のタイムアウト値を構成できます。 デフォルトのタイムアウトは 30 分です。 詳しくは、850 ページの『MFT タイムアウト値の REST API の構成』を参照してください。
- REST API for MFT を使用するには、ユーザーが mqweb サーバーで認証され、1 つ以上の `MFTWebAdmin` 役割または `MFTWebAdminRO` 役割のメンバーでなければなりません。 ユーザーの構成について詳しくは、 REST API のユーザーおよび役割の構成を参照してください。

MFT タイムアウト値の REST API の構成

接続が切断された後に REST API for MFT が調整キュー・マネージャーへの接続の試行を停止するまでの時間の長さ (分単位) を構成できます。 デフォルトのタイムアウトは 30 分です。 このタイムアウトは、**setmqweb** コマンドを使用して構成できます。

始める前に

注 : Linux V 9.3.5 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境で実行される場合、このタスクは適用されません。 REST API for MFT は、スタンドアロンの IBM MQ Web Server インストール済み環境では使用できません。

このタスクを実行するユーザーは、**dspmqweb** コマンドと **setmqweb** コマンドを使用できる特定の特権を持つユーザーでなければなりません。

- z/OS z/OS では、**dspmqweb** コマンドと **setmqweb** コマンドの実行権限と、`mqwebuser.xml` ファイルに対する書き込み権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。

重要 : z/OS

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、WLP_USER_DIR 環境変数が mqweb サーバー構成を指すように設定する必要があります。

WLP_USER_DIR 環境変数を設定するには、次のコマンドを発行します。

```
export WLP_USER_DIR=WLP_user_directory
```

ここで、*WLP_user_directory* は、`crtmqweb` に渡されるディレクトリーの名前です。 以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、mqweb サーバーの作成 を参照してください。

このタスクについて

MFT の REST API のタイムアウトを構成できます。

REST API for MFT は、調整キュー・マネージャーへの接続が切断された直後に接続の再確立を試行します。この試行が失敗した場合、タイムアウトが経過するまで、各再接続試行の間隔は 5 分になります。 そのため、0 から 5 の間の値を設定すると、再接続は 1 回だけ試行されます。

再接続がタイムアウトになった後の最初の再接続の試行は、いずれかの REST API for MFT のリソースが呼び出される時に行われます。 その再接続の試行が失敗すると、再接続がタイムアウトになるまで、MFT が再び 5 分ごとに再接続を試行するようになります。

手順

1. 以下のコマンドを使用して、REST API for MFT の現在の構成を表示します。

   ```
   dspmqweb properties -a
   ```

   `mqRestMftReconnectTimeoutInMinutes` フィールドには、MFT Transfer Rest サービスが調整キュー・マネージャーへの接続の試行を停止するまでの再接続タイムアウト値が示されます。
2. REST API for MFT が調整キュー・マネージャーへの接続の試行を停止するまでのタイムアウト (分単位) を構成します。
   - タイムアウトをデフォルト値の 30 分にリセットします。

     ```
     setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -d
     ```
   - タイムアウトを設定します。

     ```
     setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -v time
     ```

     ここで、*time* はタイムアウトが発生するまでの時間 (分単位) を指定します。

     `0-5` の間の値を設定すると、REST API for MFT は、調整キュー・マネージャーへの再接続を 1 回だけ試みます。 接続に失敗すると、REST API が呼び出されるまで接続の再確立は試行されません。

     この値を `-1` に設定すると、REST API for MFT は、接続が成功するまで再接続を試みます。
3. 以下のコマンドを入力して、mqweb サーバーを再始動します。

   ```
   endmqweb
   strmqweb
   ```

次のタスク

- REST API for MFT が有効になっていることを確認します。 詳しくは、847 ページの『MFT での REST API の有効化』を参照してください。
- 調整キュー・マネージャーが設定されていることを確認します。 詳細については、848 ページの『REST API for MFT 用の調整キュー・マネージャーの構成』を参照してください。
- REST API for MFT を使用して作成要求を実行依頼する場合は、コマンド・キュー・マネージャー名を設定する必要があります。 例えば、 **create transfer** などの REST API コマンドを使用する場合は、コマンド・キュー・マネージャー名を設定する必要があります。 詳しくは、849 ページの『REST API for MFT 用のコマンド・キュー・マネージャーの構成』を参照してください。
- REST API for MFT を使用するには、ユーザーが mqweb サーバーで認証され、1 つ以上の `MFTWebAdmin` 役割または `MFTWebAdminRO` 役割のメンバーでなければなりません。 ユーザーの構成について詳しくは、 REST API のユーザーおよび役割の構成を参照してください。

mqweb サーバーの JVM のチューニング

デフォルトでは、mqweb サーバーの Java Virtual Machine (JVM) は、ヒープの最小サイズと最大サイズ、クラス・キャッシュのサイズなどの構成パラメーターに対して、プラットフォーム固有のデフォルトを使用します。

このタスクについて

パフォーマンスの向上や問題の解決のために、デフォルト値の変更が必要な場合があります。 例えば、mqweb サーバーによって `java.lang.OutOfMemoryError` がスローされる場合は、ヒープの最大サイズを増やす必要があります。 多数のキュー・オブジェクトをロードしようとする場合は、ヒープのサイズも増やす必要があります。

IBM MQ Console でのダッシュボード構成情報の表示に問題がある場合は、構成のファイル・エンコードを決定する変数を設定する必要があります。 デフォルト値は、`jvm.options` ファイルで変更できます。

手順

1. `jvm.options` ファイルを開きます。

   `jvm.options` ファイルは、次のいずれかのディレクトリーにあります。

   - IBM MQ インストール済み環境の場合:
     - Linux AIX AIX または Linux の場合: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb`
     - Windows Windows の場合: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb`。ここで、*MQ_DATA_PATH* は IBM MQ データ・パスです。 このパスは、 IBM MQ のインストール時に選択されたデータ・パスです。 デフォルトでは、このパスは `C:\ProgramData\IBM\MQ` です。
     - IBM i IBM i 上: *MQ_DATA_PATH*`/web/installations/Installation1/`
     - z/OS z/OS 上: *WLP_user_directory*`/servers/mqweb`

       ここで、*WLP_user_directory* は、mqweb サーバー定義を作成するために **crtmqweb** スクリプトを実行したときに指定したディレクトリーです。
   - Linux V 9.3.5 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

     ここで、 *MQ_OVERRIDE_DATA_PATH* は、 **MQ_OVERRIDE_DATA_PATH** 環境変数が指す IBM MQ Web Server データ・ディレクトリーです。
2. オプション: 次の行をファイルに追加して、最大ヒープ・サイズを設定します。

   ```
   -Xmx MaxSizem
   ```

   ここで、*MaxSize* は、ヒープの最大サイズを指定します (MB 単位)。
   例えば、以下の行は、最大ヒープ・サイズを 1GB に設定します。

   ```
   -Xmx1024m
   ```
3. オプション: 次の行をファイルに追加して、最小ヒープ・サイズを設定します。

   ```
   -Xms MinSizem
   ```

   ここで、*MinSize* は、ヒープの最小サイズを指定します (MB 単位)。 最小ヒープ・サイズをデフォルトから増やすと、mqweb サーバーの始動にかかる時間を削減できます。
   例えば、次の行は、最小ヒープ・サイズを 512MB に設定します。

   ```
   -Xms512m
   ```

4. オプション: 次の行をファイルに追加して、クラス・キャッシュのサイズを設定します。

   ```
   -XscmxSizem
   ```

   ここで、*Size* は、クラス・キャッシュのサイズを指定します (MB 単位)。
   例えば、次の行は、クラス・キャッシュ・サイズを 100MB に設定します。

   ```
   -Xscmx100m
   ```

   Java 共有クラス・キャッシュは、ロード済みクラスや Ahead-Of-Time (AOT) コンパイル済みコードなどのデータを保管するために使用します。

   クラス・キャッシュによって、mqweb サーバーの始動にかかる時間が大幅に削減されます。 mqweb サーバーが初めて始動するときに、クラス・キャッシュは作成され、サーバーの始動にかなりの時間がかかる場合があります。 以降のサーバーの再始動は、クラスを共有クラス・キャッシュからロードできるため、より高速になります。

   クラス・キャッシュのサイズをデフォルトから増やすと、mqweb サーバーの始動にかかる時間を削減できます。

   z/OS 異なる z/OS システムで mqweb サーバーが始動するときに、クラス・キャッシュは再作成されます。 したがって、シスプレックス内の異なる z/OS システムで mqweb サーバーを始動すると、同じシステムでサーバーを再始動する場合よりも著しく長い時間がかかることがあります。

   この値の変更は、クラス・キャッシュが作成される場合にのみ有効になることに注意してください。 mqweb サーバーが初めて始動するとき、または Java クラス・キャッシュ・ユーティリティーを使用してクラス・キャッシュが破棄された後に、クラス・キャッシュは作成されます。
5. 必須: ファイルに以下の行が含まれていることを確認してください。これらの行は、REST API がデータを処理するときに使用されるファイル・エンコードと、 IBM MQ Console 内のユーザー・ダッシュボード構成情報を指定するためのものです。

   ```
   -Dfile.encoding=UTF-8
   -Ddefault.client.encoding=UTF-8
   ```

6. mqweb サーバーを再始動します。

   z/OS z/OS では、mqweb サーバー開始タスクを停止してから、再始動します。

   Multi その他のすべてのプラットフォームでは、コマンド行で次のコマンドを入力します。

   ```
   endmqweb
   strmqweb
   ```

IBM MQ Console および REST API のインストール・コンポーネントのファイル構造

IBM MQ Console および REST API のインストール・コンポーネントには、2 組のディレクトリー構造が関連付けられています。 一方のディレクトリー構造には、編集可能なファイルが含まれています。 もう一方のディレクトリー構造には、編集不可のファイルが含まれています。

編集可能なファイル

ユーザーが編集可能なファイルは、IBM MQ Console および REST API のインストール・コンポーネントの初期インストールの一部として配置されます。 これらは編集可能なファイルであるため、保守を適用するときに変更されることはありません。

ユーザーが編集可能なファイルの場所は、オペレーティング・システム およびインストールされている製品によって異なります。

- IBM MQ インストール済み環境では、ユーザー編集可能ファイルは以下のいずれかのディレクトリーにあります。

- Linux AIX AIX または Linux の場合: `/var/mqm/web/installations/installationName`
- Windows Windows の場合: *MQ_DATA_PATH*\web\installations*installationName*。ここで、 *MQ_DATA_PATH* は IBM MQ データ・パスです。 このパスは、 IBM MQ のインストール時に選択されたデータ・パスです。 デフォルトでは、このパスは C:\ProgramData\IBM\MQ です。
- z/OS z/OS の場合: mqweb サーバー定義を作成するために **crtmqweb** スクリプトを実行したときに指定したディレクトリー。

- Linux V 9.3.5 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST

  ここで、 *MQ_OVERRIDE_DATA_PATH* は、 **MQ_OVERRIDE_DATA_PATH** 環境変数が指す IBM MQ Web Server データ・ディレクトリーです。

この最上位ディレクトリーの下に、以下のディレクトリーとファイルがあります。

ディレクトリーおよびファイル	説明
angular.persistence/	IBM MQ Console ダッシュボード構成が保管されるディレクトリー。
servers/	WebSphere Liberty サーバー・ディレクトリー。
servers/mqweb	mqweb サーバーのディレクトリー構造が含まれるディレクトリー。
servers/mqweb/logs	mqweb サーバーのログが含まれるディレクトリー。
servers/mqweb/logs/console.log	基本的なサーバー状況および操作メッセージのログ。
servers/mqweb/logs/ffdc	初期障害データ・キャプチャー機能 (FFDC) の出力ディレクトリー。
servers/mqweb/logs/messages.log	IBM MQ Console および REST API を含む、mqweb サーバーのランタイム・メッセージのログ。 古いメッセージは、 messages_*timestamp*.log という名前のファイルに保管されます。
servers/mqweb/logs/trace.log	IBM MQ Console および REST API を含む、mqweb サーバーのトレースのログ。 古いトレースは、 trace_*timestamp*.log という名前のファイルに保管されます。 これらのファイルは、トレースが有効になっている場合にのみ存在します。
servers/mqweb/logs/state	サーバー固有の状態。
servers/mqweb/server.xml	メインのサーバー構成ファイル。 このファイルは読み取り専用です。 mqwebuser.xml ファイルを編集して、デフォルト構成をオーバーライドします。
servers/mqweb/mqwebuser.xml	IBM MQ Console および REST API の構成ファイル。 このファイルで構成した設定は、デフォルトの構成をオーバーライドします。 このファイルを編集するには、特権ユーザーである必要があります。

ディレクトリーおよびファイル	説明
servers/mqweb/resources	鍵ストアなどのさまざまなサーバー・リソースを含むディレクトリー。
servers/mqweb/workarea	サーバーの稼働中に作成されるディレクトリー。このディレクトリーは、サーバーが最初に実行された後に作成されます。

編集不可のファイル

編集不可のファイルは、IBM MQ Console および REST API のインストール・コンポーネントの初期インストールの一部として配置されます。 これらのファイルは、保守を適用すると更新されます。

編集不可ファイルの場所は、オペレーティング・システム およびインストールされている製品によって異なります。

- IBM MQ インストール済み環境では、編集不可のファイルは以下のいずれかのディレクトリーにあります。
  - ALW で AIX, Linux, and Windows: *MQ_INSTALLATION_PATH*/web
  - IBM i で IBM i: *MQ_INSTALLATION_PATH*/web
  - z/OS z/OS 上: *installation_directory*/web/

    installation_directory は、IBM MQ for z/OS UNIX System Services Components インストール・パスです。
- Linux V 9.3.5 スタンドアロン IBM MQ Web Server インストール済み環境では、 IBM MQ Web Server インストール・ファイルが解凍されたディレクトリー。

この場所には、以下のディレクトリー構造とファイルがあります。

ディレクトリーおよびファイル	説明
bin/	WebSphere Liberty コマンドを含むディレクトリー。 このディレクトリーのスクリプトを実行するには、特権ユーザーである必要があります。
mq/	さまざまな IBM MQ リソースが含まれているディレクトリー構造。
mq/apps/	IBM MQ Console および REST API アプリケーションが含まれているディレクトリー。
mq/etc/	
mq/etc/mqweb.xml	mqweb サーバーの読み取り専用の構成ファイル。 mqwebuser.xml ファイルを編集して、構成を変更します。
mq/libs	IBM MQ Console および REST API で使用される共用ライブラリーが含まれているディレクトリー。
mq/samp	サンプルが含まれているディレクトリー。
mq/samp/configuration	mqwebuser.xml ファイルにコピーできるサンプル構成ファイルが含まれているディレクトリー。

mqweb サーバーの構成のバックアップとリストア

mqweb サーバー構成をバックアップし、それを同じ場所または別の場所にリストアすることができます。

始める前に

mqweb サーバー構成をリストアするには、その前に、mqweb サーバーをリストアするシステムに IBM MQ またはスタンドアロンの IBM MQ Web Server をインストールする必要があります。 スタンドアロン IBM MQ Web Server インストール済み環境では、 826 ページの『スタンドアロン IBM MQ Web Server の構成』の手順に従って mqweb サーバーを作成する必要があります。

このタスクについて

mqweb サーバー構成をバックアップおよびリストアするには、このタスクの手順に従います。 mqweb サーバーを別の場所にリストアする場合は、mqweb サーバー構成を更新して、ファイルへの参照が正しいことを確認する必要があります。

V 9.3.5 また、この手順を使用して、 IBM MQ インストール済み環境で現在実行されている mqweb サーバーを、スタンドアロンの IBM MQ Web Server インストール済み環境で実行するようにマイグレーションすることもできます。

手順

1. mqweb サーバー構成をバックアップするには、mqweb サーバー構成が含まれているディレクトリー内のすべてのファイルをバックアップ・ロケーションにコピーします。
   - IBM MQ インストール済み環境で、以下のディレクトリーの内容をコピーします。
     - Linux AIX AIX または Linux の場合: `/var/mqm/web/installations/`*`installationName`*
     - Windows Windows の場合: *MQ_DATA_PATH*`\web\installations\`*`installationName`*。 ここで、 *MQ_DATA_PATH* は IBM MQ データ・パスです。 このパスは、 IBM MQ のインストール時に選択されたデータ・パスです。 デフォルトでは、このパスは `C:\ProgramData\IBM\MQ` です。
     - z/OS z/OS の場合: mqweb サーバー定義を作成するために **crtmqweb** スクリプトを実行したときに指定した WebSphere Liberty ユーザー・ディレクトリー。
   - Linux V 9.3.5 スタンドアロンの IBM MQ Web Server インストール済み環境では、*MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST` ディレクトリーの内容をコピーします。ここで、*MQ_OVERRIDE_DATA_PATH* は、**MQ_OVERRIDE_DATA_PATH** 環境変数が指す IBM MQ Web Server データ・ディレクトリーです。
2. mqweb サーバー構成を復元するには、mqweb サーバー構成が含まれているディレクトリーの内容を、ステップ 856 ページの『1』 でコピーしたファイルで置き換えます。
   - IBM MQ インストール済み環境で、以下のディレクトリーの内容を置き換えます。
     - Linux AIX AIX または Linux の場合: `/var/mqm/web/installations/`*`installationName`*
     - Windows Windows の場合: *MQ_DATA_PATH*`\web\installations\`*`installationName`*。 ここで、 *MQ_DATA_PATH* は IBM MQ データ・パスです。 このパスは、 IBM MQ のインストール時に選択されたデータ・パスです。 デフォルトでは、このパスは `C:\ProgramData\IBM\MQ` です。
     - z/OS z/OS の場合: mqweb サーバー定義を作成するために **crtmqweb** スクリプトを実行したときに指定した WebSphere Liberty ユーザー・ディレクトリー。
   - Linux V 9.3.5 スタンドアロンの IBM MQ Web Server インストール済み環境では、*MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST` ディレクトリーの内容を置き換

えます。ここで、*MQ_OVERRIDE_DATA_PATH* は、**MQ_OVERRIDE_DATA_PATH** 環境変数が指す IBM MQ Web Server データ・ディレクトリーです。

3. mqweb サーバーのユーザー ID がファイルにアクセスできるように、ステップ 856 ページの『2』 で復元したファイルの所有権を設定します。
4. mqweb サーバー構成を別の場所に復元した場合は、前の mqweb サーバー構成ディレクトリー内のファイルを参照する mqweb サーバー構成内のプロパティーの値を変更します。
   a) **setmqweb** コマンドまたは **dspmqweb** コマンドを発行する前に、mqweb サーバー構成を指すように環境を設定してください。
   - z/OS z/OS では、以下のコマンドを入力して、**WLP_USER_DIR** 環境変数が mqweb サーバー構成を指すように設定します。

     ```
     export WLP_USER_DIR=WLP_user_directory
     ```

     ここで、 *WLP_user_directory* は、 **crtmqweb** コマンドに渡されるディレクトリーの名前です。

     詳しくは、mqweb サーバーの作成 を参照してください。
   - Linux V 9.3.5 スタンドアロンの IBM MQ Web Server インストール済み環境では、**MQ_OVERRIDE_DATA_PATH** 環境変数を IBM MQ Web Server データ・ディレクトリーに設定します。
   - 他のすべての環境では、環境を設定するためにアクションを実行する必要はありません。

   b) ユーザーが変更したすべての構成可能な mqweb サーバー・プロパティーの値を表示します。 以下のコマンドを発行します。

   ```
   dspmqweb properties -u
   ```

   c) **remoteKeyfile** プロパティーが表示されている場合は、そのプロパティーの値を確認してください。
   プロパティーの値が以前の mqweb サーバー構成ディレクトリー内のファイル・パスを参照している場合は、新しい mqweb サーバー構成ディレクトリー内のファイル・パスを参照するように値を変更します。 以下のコマンドを発行して、 **remoteKeyfile** プロパティーの値を変更します。

   ```
   setmqweb properties -k remoteKeyfile -v path_to_keyfile
   ```

   d) mqweb サーバーのリモート・キュー・マネージャー構成を表示します。 以下のコマンドを発行します。

   ```
   dspmqweb remote -a
   ```

   e) 以下のいずれかのプロパティーが表示されている場合は、そのプロパティーの値を確認してください。
   - **globalTrustStorePath**
   - **globalKeyStorePath**
   - **ccdtURL**
   - **keyStorePath**
   - **trustStorePath**

   前の mqweb サーバー構成ディレクトリー内のファイル・パスを参照するプロパティーの値を、新しい mqweb サーバー構成ディレクトリー内のファイル・パスを参照するように変更します。
   setmqweb remote コマンドを発行して、各プロパティーの値を変更します。 例えば、固有の名前 remote-QM1 を持つリモート・キュー・マネージャーの **keyStorePath** プロパティーの値を変更するには、次のコマンドを発行します。

   ```
   setmqweb remote -uniqueName remote-QM1 -keyStorePath new_keystore_path
   ```

詳しくは、setmqweb remote (mqweb サーバーのリモート・キュー・マネージャー構成の設定)を参照してください。

Windows Linux MQ Adv. MQ Adv. VUE MQ Adv. z/OS Linux または Windows プラットフォームでの Aspera gateway 接続の定義

IBM Aspera faspio Gateway は、IBM MQ のネットワーク・スループットを大幅に向上させることができる高速 TCP/IP トンネルを提供します。 資格のあるプラットフォームで実行されているキュー・マネージャーは、 Aspera gateway を介して接続できます。 ゲートウェイ自体は、 Red Hat または Ubuntu Linux、あるいは Windows にデプロイされます。

このタスクについて

Aspera gateway を使用すると、キュー・マネージャーのチャネルのパフォーマンスを向上させることができます。 ネットワークの待ち時間が長い場合やパケットを損失する傾向がある場合に特に効果的です。一般には、別々のデータ・センターにあるキュー・マネージャー間の接続を高速化するために使用されます。

注 : パケットの損失が発生しない高速ネットワークでは、Aspera gateway を使用するとパフォーマンスが低下するので、Aspera gateway 接続を定義する前と後でネットワーク・パフォーマンスを検査することが重要です。

IP ネットワーク接続の両端で Aspera gateway を定義してから、TCP/IP を使用してキュー・マネージャー・チャネルを各ゲートウェイに接続します。 キュー・マネージャーは、そのキュー・マネージャーで使用する Aspera gateway と同じマシン上で稼働していなくてもかまいません。また、複数のキュー・マネージャーで同じゲートウェイを使用できます。

Aspera gateway を使用するには、以下のライセンスのうち 1 つ以上が必要です。

- IBM MQ Advanced for Multiplatforms
- IBM MQ Appliance
- IBM MQ Advanced for z/OS VUE
- V 9.3.4 LTS IBM MQ Advanced for z/OS(Long Term Support または Continuous Delivery from IBM MQ 9.3.4 のいずれか)

Aspera gateway は、以下のいずれかのプラットフォームにデプロイできます。

- Linux for x86-64
- Linux on Power® Systems - Little Endian
- Linux for IBM Z®
- Windows - Windows でのプラットフォーム・サポートについての詳細情報は、IBM Aspera faspio Gateway の資料を参照してください。

ゲートウェイ用に別途ライセンスが交付されているのではない限り、Aspera gateway の使用は IBM MQ メッセージに限定されます。

Aspera gateway を使用するキュー・マネージャーは、サポートされている任意のプラットフォーム上で実行できます。 サポートされるプラットフォームの完全なリストについては、 製品資料で使用されるアイコンを参照してください。

キュー・マネージャーが、そのキュー・マネージャーで使用する Aspera gateway と同じマシン上にない場合は、キュー・マネージャーと Aspera gateway の間に高速ネットワーク接続があることを確認してください。

`toml` ファイルを使用して、ゲートウェイが使用するインバウンド・ポートとアウトバウンド・ポートを定義するゲートウェイ定義を作成します。 サンプルの `toml` ファイルは、Aspera gateway と一緒に出荷されます。 アウトバウンド・ゲートウェイ定義では、ローカル・キュー・マネージャーからゲートウェイへの接続と、ローカル・ゲートウェイからリモート・ゲートウェイへの接続を定義します。 インバウンド・ゲートウェイ定義では、リモート・ゲートウェイからローカル・ゲートウェイへの接続と、ローカル・ゲートウェイからローカル・キュー・マネージャーへの接続を定義します。

以下の手順を、稼働させるための基本的なガイドとして利用できます。 詳細については、IBM Aspera faspio Gateway の資料を参照してください。

手順

1. Aspera gateway のインストール・イメージを取得します。

   Multi マルチプラットフォームの場合、パスポート・アドバンテージから Aspera gateway をダウンロードします。 ダウンロードには、 "IBM Aspera faspio Continuous Delivery Release for IBM MQ V9.3 Multiplatform Multilingual eAssembly"というラベルが付いています。 Continuous Delivery(CD)イメージとして提供されるのは、この領域の変化のペースが原因で、CD リリースの頻度で更新が必要になることを意味します。そして、IBM MQ Advanced for Multiplatforms または IBM MQ Appliance の資格を持つ任意の IBM MQ システムにインストールすることもできます。 このファイルをダウンロードするには、eAssembly, にアクセスし、IBM MQ 9.3 のダウンロードに進み、必要なリリースのタブをクリックします。 eAssembly には、ゲートウェイを使用できるすべてのプラットフォームのインストール・イメージが含まれています。 V 9.3.0 V 9.3.0 eAssembly には、ライセンス・ファイルを含む `ibm-faspio-license.zip` ファイルも含まれています。

   V 9.3.4 LTS MQ Adv. VUE MQ Adv. z/OS IBM MQ システムに IBM MQ Advanced for z/OS VUE ライセンス、または IBM MQ Advanced for z/OS ライセンス (Long Term Support または IBM MQ 9.3.4 の Continuous Delivery) がある場合は、SMP/E インストールの一部である Connector Pack コンポーネントから Aspera gateway を取得します。

   MQ Adv. VUE MQ Adv. z/OS IBM MQ Advanced for z/OS VUE および IBM MQ Advanced for z/OS のファイルは以下のとおりです。

表 69. プラットフォームおよび *IBM MQ* バージョンごとのファイル名と *faspio* バージョン番号

プラットフォーム	ファイル名	faspio バージョン番号
Linux for x86-64	V 9.3.0 M0559EN.zip	1.3.0
	V 9.3.2 M090HEN.zip	1.3.1
	V 9.3.3 M0B2XEN.zip	1.3.2
	V 9.3.4 M0C5LEN.zip	1.3.3
	V 9.3.0 M0LDNEN.zip	1.3.5 860 ページの『1.b』
Linux on Power Systems - Little Endian	V 9.3.0 M055BEN.zip	1.3.0
	V 9.3.2 M090JEN.zip	1.3.1
	V 9.3.3 M0B2YEN.zip	1.3.2
	V 9.3.4 M0C5MEN.zip	1.3.3
	V 9.3.0 M0LDPEN.zip	1.3.5 860 ページの『1.b』

表 69. プラットフォームおよび *IBM MQ* バージョンごとのファイル名と *faspio* バージョン番号 (続き)		
プラットフォーム	**ファイル名**	**faspio バージョン番号**
Linux for IBM Z	V 9.3.0 M055CEN.zip	1.3.0
	V 9.3.2 M090KEN.zip	1.3.1
	V 9.3.3 M0B2ZEN.zip	1.3.2
	V 9.3.4 M0C5NEN.zip	1.3.3
	V 9.3.0 M0LDQEN.zip	1.3.5 860 ページの『1.b』
Windows	V 9.3.0 M055DEN.zip	1.3.0
	V 9.3.2 M090LEN.zip	1.3.1
	V 9.3.3 M0B30EN.zip	1.3.2
	V 9.3.4 M0C5PEN.zip	1.3.3
	V 9.3.0 M0LDREN.zip	1.3.5 860 ページの『1.b』

注:

a. Aspera gateway は z/OS 上ではネイティブに実行できません。

b. V 9.3.0 fasp.io バージョン は APAR により で提供されています。 1.3.5 PH63366 IBM MQ 9.3.0

V 9.3.0 MQ Adv. VUE V 9.3.0 MQ Adv. z/OS インストール・イメージに加えて、`fasp` ディレクトリーには、ライセンス・ファイルを含む `M05QKEN.zip` が含まれています。

2. ゲートウェイを実行する 2 台のマシンに Aspera gateway のインストール・イメージをコピーしてから、ゲートウェイを解凍してインストールします。

   V 9.3.0 V 9.3.0 `ibm-faspio-license.zip` (Multiplatforms) または `M05QKEN.zip` (z/OS) に含まれているライセンス・ファイルを使用します。 詳しくは、 IBM Aspera faspio Gateway の資料を参照してください。

   - Linux Linux へのインストール
   - Windows Windows へのインストール

3. 各ゲートウェイを構成して保護します。

   V 9.3.0 V 9.3.0 詳しくは、IBM Aspera faspio Gateway の資料を参照してください。

   - ゲートウェイ構成ファイルの構成
   - ゲートウェイの保護

4. ネットワーク接続の両端で、ローカル・ゲートウェイが listen しているポートに接続するようにチャネル定義を変更します。
5. 各ゲートウェイ・サービスを開始します。

   V 9.3.0 V 9.3.0 詳しくは、IBM Aspera faspio Gateway の資料を参照してください。

- Linux Linux での起動
- Windows Windows での起動

6. チャネルを再始動します。

   キュー・マネージャーが Aspera gateway 接続を介して通信するようになります。

例

この例では、Linux を実行する 2 つのマシンで Aspera gateway 接続を定義します。 構成は以下のとおりです。

- ローカル・ゲートウェイ・マシンの IP アドレスは 9.20.193.107 です。 リモート・ゲートウェイ・マシンの IP アドレスは 9.20.192.115 です。
- ローカル・キュー・マネージャーは、IP アドレスが 9.20.121.5 のマシン上で稼働しています。 リモート・キュー・マネージャーは、IP アドレスが 9.20.121.25 のマシン上で稼働しています。 どちらのキュー・マネージャーも、ポート 1414 で listen しています。
- ローカル・キュー・マネージャーのキュー・マネージャー・チャネルは、**conname** 9.20.193.107 (1500) を使用してローカル Aspera gateway に接続するように変更されます。 リモート・キュー・マネージャーのキュー・マネージャー・チャネルは、**conname** 9.20.192.115 (1500) を使用してリモート Aspera gateway に接続するように変更されます。
- V 9.3.0 V 9.3.0 IBM Aspera faspio Gateway 1.2 以降、TLS はデフォルトで有効になっています。 ゲートウェイを使用して TLS を構成する場合は、 IBM Aspera faspio Gateway 資料の「ゲートウェイの保護」を参照してください。

1. ローカル・ゲートウェイ・マシンに Aspera gateway 接続を定義します。
   - Aspera gateway をインストールします:
     - Linux Linux では、以下のコマンドを使用します。

       ```
       rpm -ivh ibm-faspio-gateway-<version>.x86_64.rpm
       ```

   - インストールによって、作成されたディレクトリー内の `gateway.toml` ファイルを変更します:

     このファイルを編集してローカル・ゲートウェイの定義を設定します。

     ```
     [[bridge]]
         name = "Outbound"
         [bridge.local]
                protocol = "tcp"
                    host = "9.20.193.107"
                    port = 1500
         tls_enabled = false

         [bridge.forward]
                protocol = "fasp"
                    host = "9.20.192.115"
                    port = 1600
         tls_enabled = false

     [[bridge]]
         name = "Inbound"
         [bridge.local]
                protocol = "fasp"
                    host = "9.20.193.107"
                    port = 1600
         tls_enabled = false

         [bridge.forward]
                protocol = "tcp"
                    host = "9.20.121.5"
                    port = 1414
         tls_enabled = false
     ```

   - V 9.3.0 V 9.3.0 `aspera-license` ファイルを `ibm-faspio-license.zip` (Multiplatforms) または `M05QKEN.zip` (z/OS) から `/usr/local/etc/faspio/`にコピーします。

2. リモート・ゲートウェイ・マシンで上記の手順を繰り返して Aspera gateway 接続を定義します。
   - インストールによって作成されたディレクトリー内の `gateway.toml` ファイルを変更します。 このファイルを編集してリモート・ゲートウェイの定義を設定します。

```
[[bridge]]
    name = "Outbound"
    [bridge.local]
           protocol = "tcp"
               host = "9.20.193.107"
               port = 1500
    tls_enabled = false

    [bridge.forward]
           protocol = "fasp"
               host = "9.20.192.115"
               port = 1600
    tls_enabled = false

[[bridge]]
    name = "Inbound"
    [bridge.local]
           protocol = "fasp"
               host = "9.20.193.107"
               port = 1600
    tls_enabled = false

    [bridge.forward]
           protocol = "tcp"
               host = "9.20.121.5"
               port = 1414
    tls_enabled = false
```

   - V 9.3.0 V 9.3.0 `aspera-license` ファイルを `ibm-faspio-license.zip` (Multiplatforms) または `M05QKEN.zip` (z/OS) から `/usr/local/etc/faspio/`にコピーします。
3. 接続の両端で、ローカル・ゲートウェイが listen しているポートに接続するようにチャネル定義を変更します。
   - **conname** 9.20.193.107 (1500) を使用してローカル Aspera gateway に接続するように、ローカル・キュー・マネージャーのキュー・マネージャー・チャネルを変更します。
   - **conname** 9.20.192.115 (1500) を使用してリモート Aspera gateway に接続するように、リモート・キュー・マネージャーのキュー・マネージャー・チャネルを変更します。
4. ローカル・ゲートウェイ・マシンで以下のコマンドを実行して、ローカル・ゲートウェイを開始します。
   - Linux

```
sudo systemctl start faspio-gateway
```

5. リモート・ゲートウェイ・マシンで以下のコマンドを実行して、リモート・ゲートウェイを開始します。
   - Linux

```
sudo systemctl start faspio-gateway
```

6. チャネルを再始動します。

次のタスク

Aspera gateway は、受信したデータを一切解釈せずに転送します。 つまり、ゲートウェイ接続で TLS ハンドシェークが認識されないので、Aspera gateway を使用するキュー・マネージャー・チャネル間に TLS を構成できることになります。 これはまた、サポートされるすべての IBM MQ プラットフォーム上のキュー・マネージャーで Aspera gateway を使用できることも意味します。

ゲートウェイで複数インスタンス・キュー・マネージャーを使用する場合には、キュー・マネージャーのインスタンスごとにゲートウェイ定義を構成します。

注 : Aspera gateway は、キュー・マネージャー・チャネルでのみテストされています。 クライアント・チャネルではテストされていません。 これは、通常、クライアント・アプリケーションは高速ネットワークを介してローカル・データ・センター内のキュー・マネージャーに接続するのに対して、 Aspera gateway では低速ネットワークを介してリモート・キュー・マネージャーに接続することが想定されているためです。

関連概念

Aspera gateway ロードマップ

関連資料

16 ページの『使用する通信タイプ』

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

IBM Aspera faspio Gateway 資料

Multi IBM Cloud Private 計量サービスで使用するための IBM MQ の構成

キュー・マネージャーの始動情報と使用情報を報告および表示するために、 IBM Cloud Private 計量サービスで使用する IBM MQ を構成します。

始める前に

IBM Cloud Private サービスを使用するように IBM MQ キュー・マネージャーを構成する前に、 IBM Cloud アカウントを持っている必要があります。 アカウントを作成するには、IBM Cloud に登録を参照してください。

このタスクについて

IBM Cloud Private 計測サービスを使用することにより、オンプレミスの IBM 製品を IBM Cloud Private のサービス・インスタンスに接続し、組織内のすべての登録済み製品を一つのダッシュボードに表示することができます。

AIX、Linux、および Windows のキュー・マネージャーを構成して計量サービス・インスタンスに接続し、開始情報および使用量情報を表示できます。 ただし、Linux コンテナー環境以外のプラットフォームでは、時間単位コンテナー・ベースの価格設定ライセンスに基づいてデータを使用することはできません。

デフォルトの毎時ライセンス・メトリックではなく毎月の VPC ライセンス・タイプの使用データを記録するには、環境変数 `AMQ_LICENSING_METRIC=VPCMonthlyPeak` を設定します。 これにより、キュー・マネージャーは、毎時のコンテナー・ベースのライセンスに関連するデータをアップロードするデフォルト動作ではなく、毎月の VPC ライセンス・タイプに関連するデータをアップロードします。

`qm.ini` ファイルで、`ReportingService` スタンザに以下の属性を使用します。

APIKeyFile
計量サービス・インスタンスの **APIKey** 値を持つテキスト・ファイルの場所。

CapacityReporting
以下の形式でエラー・ログ・メッセージを定期的に AMQERR ログに書き込みます。

```
4/22/2020 01:44:29 PM - Process(1274.1) User(bld-adm) Program(amqmgr0)
Host(8b3b83f2bc7d) Installation(Docker)
VRMF(9.2.0.0)
Time(2020-04-22T13:44:29.295Z)
ArithInsert1(300)
CommentInsert1(8.5)
CommentInsert2(IBM MQ Advanced)
```

CapacityReporting 属性で作成された情報は、メッセージ AMQ5064 内に挿入されます。このメッセージを参照すると、自社で使用している IBM MQ の使用量に関する理解を深めることができます。

AMQ5064
このキュー・マネージャーは、300 秒間実行されています。 現在は 8.5 コアで実行中です。 ライセンス・タイプは IBM MQ Advanced です。

重大度
0 : 情報

説明
これは、使用量のトラッキングに関する情報メッセージです。

応答
なし。

LicensingGroup
キュー・マネージャーが属する請求グループ。 計量サービスが生成するレポートでデータがグループ化される方法に影響を与えます。

ServiceURL
IBM Cloud Private サービス・アドレス。

ServiceProxy
計量サービスが実行されているネットワークにキュー・マネージャーが直接アクセスできない場合に使用できる HTTP プロキシーの URL およびポート。

製品がインストールされているホスト、使用している製品バージョン、および実行されているプラットフォームを表示できます。 製品ごとに表示される使用メトリックの概要から、ワークロードの大きさに関する概要が得られます。 IBM MQ では、どのキュー・マネージャーの使用頻度が高く、どのキュー・マネージャーのワークロードが低いかを知ることができます。

計量サービスのインスタンスに接続するようキュー・マネージャーが構成されると、以下の情報が IBM Cloud Private に報告されます。

- IBM MQ キュー・マネージャー名
- IBM MQ キュー・マネージャー ID
- IBM MQ インストール・ルート・ディレクトリー
- インストール済み IBM MQ コンポーネント (名前とバージョン)
- ホスト名
- ホスト・オペレーティング・システム名
- ホスト・オペレーティング・システム・バージョン
- IBM MQ キュー・マネージャーについての仮想プロセッサー・コア (VPC) の使用量情報

キュー・マネージャーの VPC 使用メトリックは計量サービス・インスタンス・ダッシュボードでモニターできます。

手順

- IBM Cloud Private で、計量サービス・インスタンスで使用するようキュー・マネージャーを構成します。
- HTTP プロキシーを介した IBM Cloud Private 計量サービスに接続します。
- IBM Cloud Private 計量サービスへの接続のトラブルシューティングを行います。

関連資料
仮想プロセッサー・コアの価格設定メトリック (VPC)

Multi IBM Cloud Private の計量サービス・インスタンスで使用するためのキュー・マネージャーの構成

キュー・マネージャーのセキュリティーと IBM Cloud 登録情報をセットアップした後、既に作成されている計量サービス・インスタンスに接続します。

このタスクについて

IBM Cloud Private 計量サービス・インスタンス・ダッシュボードには、セキュリティーおよび IBM Cloud Private 登録情報が含まれるよう構成されているキュー・マネージャーのデータのみが表示されます。

手順

1. ICP の文書に示された、サービス ID を作成する手順に従います。
   IBM Cloud Private CLI を使用したサービス ID の作成。
2. ICP の文書に示された、API 鍵を作成する手順に従います。
   API 鍵管理 API。
3. ICP クラスターから TLS 証明書をダウンロードします。
   証明書をダウンロードした場所をメモします。 ステップ 866 ページの『9』 で、ダウンロードした証明書をキュー・マネージャーの鍵ストアに追加することができます。
4. テキスト・ファイル `apikeyfile.txt` を作成し、前のタスクでコピーした **API key** 値を追加します。
   `apikeyfile.txt` の場所をメモしておくと、ステップ 8 でそのパスを含めることができます。 このファイルは、キュー・マネージャー・ユーザー(AIX and Linux システムでは「*'mqm*」)によって読み取り可能である必要があります。 このファイルには、JSON ペイロードではなく、**API key** 自体のみが含まれている必要があります (例: `d9c11b45-4dda-4de4-c0b2-2e4e1004dc64`)。
5. キュー・マネージャー (例えば *QM1*) を作成します。
   詳細については、Multiplatforms でのキュー・マネージャーの作成と管理を参照してください。
6. キュー・マネージャー *QM1* を開始します。
   詳しくは、 キュー・マネージャーの開始を参照してください。
7. IBM MQ コマンドを実行する前に、忘れずに IBM MQ コマンド行環境をセットアップしておいてください。
   setmqenv コマンドを実行します。

   AIX On AIX:

   ```
   . /usr/mqm/bin/setmqenv -s
   ```

   Linux On Linux:

   ```
   . /opt/mqm/bin/setmqenv -s
   ```

   Windows On Windows:

   ```
   "C:\Program Files\IBM\MQ\bin\setmqenv.cmd" -n installation name
   ```

8. キュー・マネージャー *QM1* の SSL トラストストアを作成します。

   AIX トラストストアの作成を開始します。 AIX の場合:

   ```
   runmqckm -keydb -create -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
   -expire 30 -stash
   ```

   Linux On Linux:

   ```
   runmqckm -keydb -create -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
   -expire 30 -stash
   ```

   Windows On Windows:

   ```
   runmqckm -keydb -create -db "MQ data directory\qmgrs\QM1\ssl\key.kdb" -pw password -type
   cms -expire 30 -stash
   ```

9. 手順 865 ページの『3』 でダウンロードしたデジタル証明書をキュー・マネージャーのトラストストアに追加します。

   AIX On AIX:

   ```
   runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
   -label RootCA
   -file Download_location/RootCA.crt -format ascii -trust enable

   runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
   -label ServerCert
   -file Download_location/CERT.crt -format ascii -trust enable
   ```

   Linux On Linux:

   ```
   runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
   -label RootCA
   -file Download_location/RootCA.crt -format ascii -trust enable

   runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
   -label ServerCert
   -file Download_location/CERT.crt -format ascii -trust enable
   ```

   Windows On Windows:

   ```
   runmqckm -cert -add -db "MQ data directory\qmgrs\QM1\ssl\key.kdb" -pw password -type cms
   -label RootCA
   -file "Download_location\RootCA.crt" -format ascii -trust enable

   runmqckm -cert -add -db "C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl\key.kdb" -pw password -type
   cms -label ServerCert
   -file "Download_location\CERT.crt" -format ascii -trust enable
   ```

10. キュー・マネージャーの qm.ini ファイルへの apikeyfile パスを指定して、新規 ReportingService スタンザを追加します。

    ```
    ReportingService:
       APIKeyFile=APIKey file location/apikeyfile.txt
    ```

11. **API host** 値を qm.ini ファイルに追加します。

    ReportingService スタンザ・セクションには、 apikeyfile 値と **API host** (**ServiceURL**) 値へのパスが含まれるようになりました。

    ```
    ReportingService:
       APIKeyFile=APIKey file location/apikeyfile.txt
       ServiceURL=https://productinsights-api.ng.bluemix.net
    ```

    qm.ini ファイルを保存して終了します。

12. 変更内容を有効にするには、キュー・マネージャーを再始動します。

    キュー・マネージャーのプロセス **amqzmur0** がネットワークにアクセスするために権限を付与するよう求められる可能性があります。 キュー・マネージャーが計量サービスに接続できるようにするには、アクセス権限が必要です。

13. 計量サービス・インスタンスのキュー・マネージャー *QM1* に関する情報を表示します。

    レポート状況がアクティブになっている場合、指定された統合ノードのすべての統合サーバーに関する開始情報と使用情報が計量サービスに報告されます。 使用情報は、15 分ごとに更新されます。

14. オプション: ReportingService スタンザをキュー・マネージャーの qm.ini ファイルから削除し、キュー・マネージャーを再始動して、キュー・マネージャーによる計量サービスへの報告を停止します。

15. オプション: キュー・マネージャーが開始情報と使用情報を計量サービスに報告できない場合、キュー・マネージャーのログ・ファイルにある診断情報を確認します。

    AIX 用に修正

AIX On AIX:

```
/var/mqm/qmgrs/QM1/errors/AMQERR0*.log
```

Linux On Linux:

```
/var/mqm/qmgrs/QM1/errors/AMQERR0*.log
```

Windows On Windows:

```
C:\ProgramData\IBM\MQ\errors\AMQERR0*.log
```

タスクの結果

計量サービス・インスタンスの作成が完了し、インスタンスに接続するようキュー・マネージャーを構成しました。 計量サービス・インスタンスのダッシュボードでキュー・マネージャーに関する情報を参照することができます。

Multi HTTP プロキシーを介した IBM Cloud Private 計量サービスへの接続

キュー・マネージャーが ICP クラスターに直接アクセスできないシステムで実行されている場合は、組織が提供する HTTP プロキシーを使用して、IBM Cloud Private の計量サービス・インスタンスに接続できます。

始める前に

セキュリティーを構成し、 **API key** およびサービスの URL をキュー・マネージャーの `qm.ini` ファイルに追加しました。

このタスクについて

このタスクを使用して、組織によって提供される HTTP プロキシーを介して IBM Cloud Private の 計量サービス ・インスタンスに接続するようにキュー・マネージャーを構成します。

手順

- サービス・プロキシー属性を `qm.ini` ファイルの IBM Cloud Private 登録スタンザに追加します。

  ServiceProxy 属性は、以下のように設定できます。

  - http:// 接頭部と、オプションでポートを含む URL。 ポートを指定しない場合は、*1080* が使用されます。

    ```
    ReportingService:
       ServiceProxy=http://myorgproxy.net:1080
    ```

  注 : **ServiceProxy** パラメーターに有効な http:// URL を設定する必要があります。 その他のプロキシー・プロトコル (例えば、HTTPS や SOCKS) はサポートされません。
- キュー・マネージャーを再始動して変更を有効にします。

Multi 計量サービスへの接続のトラブルシューティング

キュー・マネージャーを計量サービス・インスタンスに接続する場合に発生する可能性があるエラーについてのトラブルシューティング・アドバイス。

キュー・マネージャーが、構成済みの計量サービスに登録できないか、またはサービスに使用メトリックをアップロードできません

キュー・マネージャーにネットワークへのアクセス権があることを確認します。 API 鍵ファイルの **APIKey** 値が間違っています。 IBM Global Security Kit (GSKit) コンポーネントがインストールされていることを確認してください。

無効な qm.ini スタンザ

無効な qm.ini スタンザが見つかりました。 詳しい情報についてはエラー・ログを確認してください。

無効な HTTP サービス・プロキシー・パラメーター

キュー・マネージャー ReportingService スタンザの **ServiceProxy** 属性の値が正しく構成されていません。 キュー・マネージャーがサービスに登録されません。 **ServiceProxy** パラメーターに有効な http:// URL を設定する必要があります。 その他のプロキシー・プロトコル (例えば、HTTPS や SOCKS) はサポートされません。

Deprecated Linux Salesforce プッシュ・トピックおよびプラットフォーム・イベントで使用するための IBM MQ の構成

この情報を使用して、 IBM MQ Bridge to Salesforce を構成してから実行することにより、 Salesforce および IBM MQ ネットワークへのセキュリティーと接続をセットアップします。

始める前に

注 : Deprecated IBM MQ Bridge to Salesforce は、2022 年 11 月 22 日以降のすべてのリリースで非推奨になりました (US Announcement letter 222-341 を参照)。 Salesforce 接続は、 IBM App Connect を使用して、または IBM Cloud Pak® for Integration で使用可能な App Connect 機能を使用して実現できます。

- IBM MQ Bridge to Salesforce は Linux x86-64(64 ビット)の場合に使用可能です。 IBM WebSphere MQ 6.0 以前で実行されているキュー・マネージャーに接続する場合、このブリッジはサポートされていません。
- IBM MQ 9.2.0 以降、キュー・マネージャーは、適切に構成されている複数のブリッジ・インスタンスをサポートできます。 詳しくは、874 ページの『IBM MQ Bridge to Salesforce の追加の構成オプション』を参照してください。
- **MQSeriesSFBridge** パッケージをインストールします。 詳しくは、Linux への IBM MQ サーバーのインストールと Linux システム用の IBM MQrpm コンポーネントを参照してください。

このタスクについて

Salesforce は、クラウド・ベースのカスタマー・リレーションシップ・マネジメント・プラットフォームです。 Salesforce を使用して顧客データと対話を管理している場合、IBM MQ Bridge to Salesforce を使用して Salesforce プッシュ・トピックおよびプラットフォーム・イベントにサブスクライブできます。 これらは、その後 IBM MQ キュー・マネージャーにパブリッシュできます。 そのキュー・マネージャーに接続されているアプリケーションは、プッシュ・トピックおよびプラットフォーム・イベント・データを有用な仕方でコンシュームできます。 このブリッジを使用して、Salesforce 内のプラットフォーム・イベントのイベント・メッセージを作成することもできます。

IBM MQ Bridge to Salesforce の概要については、図 1 のダイアグラムを参照してください。

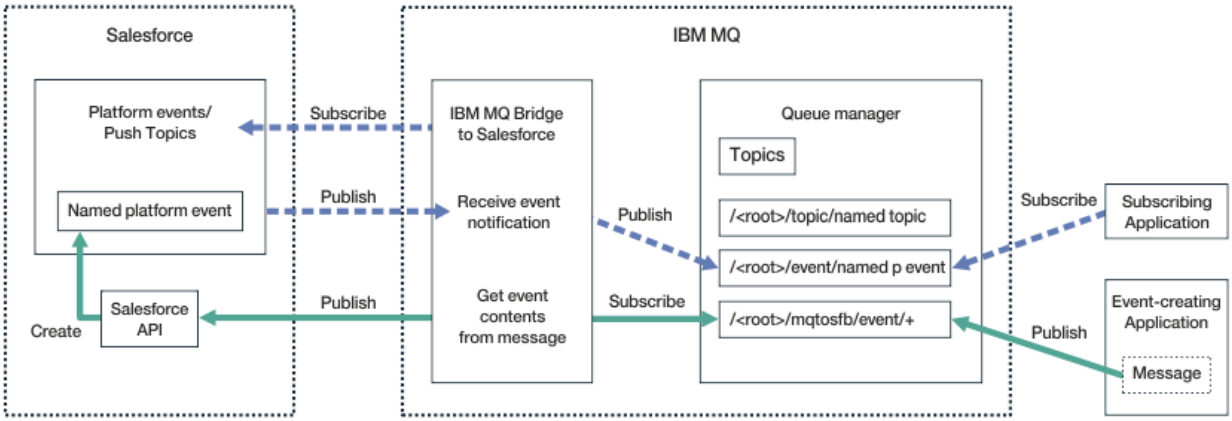


図 *97. IBM MQ Bridge to Salesforce*

プッシュ・トピックは、Salesforce のレコードに対する変更の通知を受信するために、Force.com ストリーミング API を使用するように定義する照会です。 プッシュ・トピックの構成と Streaming API の使用方法について詳しくは、ストリーミング API の概要 および PushTopic を使った作業 を参照してください。

プラットフォームイベントはカスタマイズ可能なイベントメッセージであり、Force.com プラットフォームが生成または消費するイベントデータを決定するために定義できます。 プラットフォーム・イベントと Salesforce イベントとの違いについて詳しくは、Enterprise messaging platform events および What is the difference between the Salesforce events を参照してください。

- プッシュ・トピックとプラットフォーム・イベントのサブスクライブの構成を作成するには、870 ページの『IBM MQ Bridge to Salesforce の構成』を参照してください。
- Salesforce プラットフォーム・イベントのイベント・メッセージを作成するための構成を作成するには、877 ページの『Salesforce プラットフォーム・イベントのイベント・メッセージの作成』を参照してください。

ブリッジからのデータをモニターするには、IBM MQ Console を使用する方法と、**amqsrua** コマンドで **-p** パラメーターを使用する方法の 2 つの方法があります。 以下のように、ブリッジ全体の状況について一連のデータがパブリッシュされます。

- インターバル内で処理されたプッシュ・トピック・メッセージの合計(STATUS/PUSHTOPIC ツリーの下)。
- この間隔で表示されるプッシュ・トピックの数。
- インターバル内で処理されたプラットフォーム・イベントの合計(STATUS/PLATFORM ツリーの下)。
- この間隔で表示されるプラットフォーム・イベントの数。
- インターバル内で(STATUS/MQPE ツリーの下で)処理された IBM MQ 作成されたプラットフォーム・イベントの総数。
- この間隔で表示される IBM MQ 作成のプラットフォーム・イベントの固有数。
- この間隔で表示される IBM MQ 作成のプラットフォーム・イベントのパブリケーションの失敗数。

構成された Salesforce トピックごとに、さらにメッセージがパブリッシュされます。 IBM MQ トピックでは、以下のように、オブジェクト名に完全 Salesforce トピック名と/event または/topic を使用します:

- ある間隔で処理されるメッセージの数。

ブリッジ・データをモニターするように IBM MQ Console を構成する場合は、IBM MQ Bridge to Salesforce の構成のステップ 9 と 10 を参照してください。 **amqsrua** コマンドの使用方法については、IBM MQ Bridge to Salesforce のモニターを参照してください。

以下のタスクのステップに従い、IBM MQ Bridge to Salesforce を構成して実行します。

手順

1. IBM MQ Bridge to Salesforce を構成します。

2. Salesforce プラットフォーム・イベントのイベント・メッセージを作成します。
3. IBM MQ Bridge to Salesforce を実行します。

関連タスク
IBM MQ Bridge to Salesforce のトレース
関連資料
runmqsfb (IBM MQ Bridge to Salesforce の実行)

Deprecated Linux IBM MQ Bridge to Salesforce の構成

IBM MQ を構成し、IBM MQ Bridge to Salesforce パラメーターを入力して、構成ファイルを作成し、Salesforce のプッシュ・トピックおよびプラットフォーム・イベントを IBM MQ キュー・マネージャーに接続することができます。

始める前に

注 : Deprecated IBM MQ Bridge to Salesforce は、2022 年 11 月 22 日以降のすべてのリリースで非推奨になりました (US Announcement letter 222-431 を参照)。 Salesforce 接続は、 IBM App Connect を使用するか、 IBM Cloud Pak for Integration で使用可能な App Connect 機能を使用して実現できます。

この作業を開始する前に、x86-64Linux プラットフォームでのインストールで `MQSeriesSFBridge` パッケージが IBM MQ インストールされていることを確認してください。

詳しくは、Linux への IBM MQ サーバーのインストールと Linux システム用の IBM MQrpm コンポーネントを参照してください。

このタスクについて

このタスクでは、IBM MQ Bridge to Salesforce 構成ファイルを作成し、Salesforce および IBM MQ に正常に接続するために必要な最小限のセットアップを使用して、Salesforce のプッシュ・トピックおよびプラットフォーム・イベントにサブスクライブできるようにします。 すべてのパラメーターの意味とオプションについて詳しくは、runmqsfb コマンドを参照してください。 独自のセキュリティー要件を考慮し、デプロイメントに適したパラメーターをカスタマイズする必要があります。

Salesforce プラットフォーム・イベントのイベント・メッセージを作成するための構成を作成するには、877 ページの『Salesforce プラットフォーム・イベントのイベント・メッセージの作成』を参照してください。

Salesforce プッシュ・トピックとプラットフォーム・イベントのサブスクライブ

IBM MQ Bridge to Salesforce は、Salesforce と IBM MQ の両方への接続を確立すると、Salesforce プッシュ・トピックおよびプラットフォーム・イベントに対するサブスクリプションを作成します。 ブリッジによるサブスクライブが必要なプッシュ・トピックやプラットフォーム・イベントの名前は、接続する前に、構成ファイルに組み込むか、コマンド行に追加する必要があります。

構成属性の 1 つは IBM MQ トピック・ツリーのルートであり、イベントはこのルートの下にパブリッシュされます。 ブリッジはこのルートにアクセスし、完全な Salesforce トピック名、例えば、`/MQ/SF/ROOT/topic/EscalatedCases` を追加します。 IBM MQ に接続しているモニター・トピックおよびアプリケーションは、`/event/NewCustomer__e` の下の`/topic/EscalatedCases` およびプラットフォーム・イベントの下でプッシュ・トピックを探すことができます。

パブリッシュされたメッセージには、制御情報と、要求されたデータ・フィールドを含むデータ構造が含まれています。 プッシュ・トピックの場合、データ構造は **sobject** で、プラットフォーム・イベントの場合、この構造は **payload** です。 ブリッジでは、Salesforce に定義されていないトピックやイベントにサブスクライブすることはできません。 ブリッジでトピックへのサブスクライブを試みたときにエラーが発生すると、ブリッジは停止します。

トピック・オブジェクトは、IBM MQ で定義する必要はありませんが、ツリー内の最も近い親エレメントに基づいて、適切な権限が存在している必要があります。 リパブリッシュされたメッセージには、デフォルトで、元のメッセージからの関連するデータ構造のみが含まれます。 制御情報は削除されます。 プラットフォーム・イベントの場合、パブリケーションはペイロード構造になっています。 構成パラメーターの

「**ブリッジ・プログラムの動作 (Behaviour of bridge program)**」セットの **Publish control data with the payload** 構成オプションにより、制御データを含むメッセージ全体の再パブリッシュが可能になります。 詳しくは、構成パラメーターを参照してください。

各プッシュ・トピックおよびプラットフォーム・イベントには、 Salesforce からの公開時に関連付けられた *ReplayID* があります。 *ReplayID* を使用すると、サーバーに接続する際にパブリケーションの開始点を要求できます。 Salesforce では、履歴を最大 24 時間維持できるので、最新のプッシュ・トピックおよびプラットフォーム・イベントが生成された時刻にブリッジが開始しなかった場合でも、ブリッジでそれらは失われません。 ブリッジでは、以下の 2 つのサービスの品質がサポートされます。

最大 1 回
ブリッジでは、再始動に *ReplayId* は使用されません。 ブリッジの再始動後、新規に生成されたプッシュ・トピックとプラットフォーム・イベントのみが処理されます。 欠落しているパブリケーションのみを処理するように、アプリケーションを準備する必要があります。 *ReplayId* はまだブリッジによって追跡され、キューにハード化されているので、他のサービス品質でブリッジを再始動し、現在の状態を把握することができます。

最低 1 回
ブリッジは、*ReplayId* を追跡し、キューにハード化します。 ブリッジの再始動時には、永続 *ReplayId* を使用して、サーバーからのパブリケーションの開始点が要求されます。 ギャップが 24 時間以内であった場合、それより古いパブリケーションが送信されます。 トピックの *ReplayId* は、すべてのメッセージでハード化される訳ではありません。 定期的な間隔で永続メッセージ内に書き込まれ、ブリッジのシャットダウン時にも書き込まれます。 アプリケーションで、重複パブリケーションを確認する準備ができている必要があります。

ReplayId は、新規に定義されたキューにメッセージとして書き込まれます。 ブリッジの開始前に、このキュー (**SYSTEM.SALESFORCE.SYNCQ**) を定義しておく必要があります。 **SYSTEM.SALESFORCE.SYNCQ** が存在しない場合、ブリッジは、サービスの品質モードに関係なく続行されません。 関連する属性を持つキューを作成するための MQSC スクリプトが用意されています。 DEFSOPT(EXCL) NOSHARE オプションを使用してキューを構成し、ブリッジ・プログラムの 1 つのインスタンスだけが **SYSTEM.SALESFORCE.SYNCQ** キューを更新できるようにする必要があります。

プラットフォーム・イベントのイベント・メッセージを作成するための構成を作成するには、877 ページの『Salesforce プラットフォーム・イベントのイベント・メッセージの作成』を参照してください。

手順

1. キュー・マネージャーを作成して開始します。
   a) キュー・マネージャー (例えば、SQM1) を作成します。
   ```
   crtmqm SQM1
   ```
   b) キュー・マネージャーを開始します。
   ```
   strmqm SQM1
   ```
2. **注 :** 既存のログインおよびセキュリティーの Salesforce 資格情報および自己署名証明書を使用するには、ステップにスキップしてください 871 ページの『3』。
   オプション: ご使用の Salesforce アカウント用のセキュリティー・トークンを作成します。
   a) Salesforce アカウントにログインします。
   b) ヘルプ項目 Salesforce help: Reset your security token のステップに従って、セキュリティー・トークンを作成またはリセットします。
3. CA 署名のセキュリティー証明書を Salesforce に作成します。
   a)「**Force.com ホーム**」ページの「**管理**」メニューから「**セキュリティのコントロール**」を選択し、「**証明書と鍵の管理**」を選択します。
   「**証明書と鍵の管理**」ページが開きます。
   b) **CA 署名済み証明書**をクリックする。
   「**証明書**」ページが開きます。

c)「**Label (ラベル)**」フィールドに証明書の名前を入力し、Tab キーを押してから「**保存**」をクリックします。

証明書と鍵の詳細情報が表示されます。

d)「**Back to list: Certificates and keys (一覧に戻る: 証明書と鍵の管理)**」をクリックします。

e)「**キーストアにエクスポート**」をクリックします。

f) 鍵ストアのパスワードを入力し、「**エクスポート**」をクリックします。

g) エクスポートした鍵ストアをローカル・ファイル・システムに保存します。

4. IBM 鍵管理 GUI を使用して、Salesforce からエクスポートした鍵ストアを開き、署名者証明書を追加します。

a) **strmqikm** コマンドを実行して、IBM 鍵管理 GUI を開きます。

詳しくは、runmqckm、runmqakm、および strmqikm を使用したデジタル証明書の管理を参照してください。

b)「**鍵データベース・ファイルを開く**」をクリックし、Salesforce 鍵ストアのロケーションを参照します。

c)「**開く**」をクリックして、「**鍵データベース・タイプ**」から「**JKS**」が選択されていることを確認し、「**OK**」をクリックします。

d) ステップ 3f で鍵ストア用に作成したパスワードを入力してから、「**OK**」をクリックします。

e)「**鍵データベースの内容**」オプションから「**署名者証明書**」を選択します。

f)「**挿入**」をクリックします。

g)「**CA 証明書の追加**」リストから「**Verisign Inc.**」チェック・ボックスを選択し、「**OK**」をクリックします。

5. オプション: Salesforce アカウントで IBM MQ Bridge to Salesforce のアプリ接続を作成して、OAuth コンシューマー鍵と秘密情報を生成します。

実稼働環境で IBM MQ Bridge to Salesforce を使用する場合は、「**コンシューマー鍵**」 コードと 「**コンシューマー秘密情報**」 コードが必要です。

a)「**作成**」を選択し、「**Force.com ホーム**」ページの「**ビルド**」メニューから「**アプリケーション**」を選択します。

「アプリケーション」ページが開きます。

b)「**接続アプリケーション**」セクションから「**新規**」をクリックします。

「**新規接続アプリケーション**」ページが開きます。

c) ご使用の IBM MQ Bridge to Salesforce の名前 (例えば、**MQBridgeToSalesforce**) を「**接続アプリケーション名**」に入力します。

d)「**API Name (API 名)**」を入力します。

タブで次のフィールドに移動すると、「**接続アプリケーション名**」が「**API Name (API 名)**」名前フィールドにコピーされます。

e)「**Contact Email (連絡先 E メール)**」を入力します。

f)「**API (OAuth 設定の有効化)**」セクションで「**OAuth 設定の有効化**」オプションを選択します。

そのセクションの他のオプションが表示されます。

g)「**コールバック URL**」 (例えば、`https://www.ibm.com`) を入力します。

h)「**選択した OAuth 範囲**」サブセクションの「**利用可能な OAuth 範囲**」リストから「**フルアクセス (full)**」オプションを選択し、「**追加**」をクリックして、「**選択した OAuth 範囲**」リストに全アクセス権限を追加します。

i) **保存** をクリックします。

j)「**次へ進む**」をクリックします。

k)「**コンシューマ鍵**」と「**コンシューマの秘密**」のコードを書き留めます。

6. キュー・マネージャー上に必要な同期キューを作成します。

```
cat /opt/mqm/mqsf/samp/mqsfbSyncQ.mqsc | runmqsc SQM1
```

同期キューによって、アプリケーション間のイベント状態が維持されたり、キュー・マネージャーが再始動されたりします。 キュー上にあると予想されるメッセージは 1 つだけなので、キュー項目数は少なくて済みます。 このキューに対して一度に実行できるブリッジのインスタンスは 1 つだけなので、デフォルト・オプションは排他的アクセス用に設定されます。

7. IBM MQ、Salesforce、および IBM MQ Bridge to Salesforce の動作に対する接続とセキュリティーのパラメーターを使用して、構成ファイルを作成します。

```
runmqsfb -o new_config.cfg
```

既存の値が大括弧内に表示されます。 Enter を押すと、既存の値を受け入れます。 Space を押してから Enter を押すと、値をクリアします。 値を入力してから Enter を押すと、新規の値を追加します。

a) 以下のようにして、キュー・マネージャー SQM1 への接続のための値を入力します。

接続に必要な最小限の値は、キュー・マネージャー名、IBM MQ ベース・トピック・ルート、およびチャネル名です。

```
Connection to Queue Manager
---------------------------
Queue Manager or JNDI CF   : []SQM1
MQ Base Topic              : []/sf
MQ Channel                 : []A channel you have defined or for example
SYSTEM.DEF.SVRCONN
MQ Conname                 : []
MQ Publication Error Queue : [SYSTEM.SALESFORCE.ERRORQ]
MQ CCDT URL                : []
JNDI implementation class  : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL          : []
MQ Userid                  : []
MQ Password                : []
```

注 : ローカル接続を行う場合、チャネル名は必要ありません。 構成ファイルにキュー・マネージャー名とベース・トピックを指定する必要はありません。 これらは、後でブリッジを実行するときにコマンド行に含めることができます。

b) 以下のようにして、Salesforce への接続のための値を入力します。

接続に必要な最小限の値は、Salesforce のユーザー ID、パスワード、セキュリティー・トークン、およびログイン・エンドポイントです。 実稼働環境では、OAuth セキュリティーのコンシューマー鍵と秘密情報を追加できます。

```
Connection to Salesforce
------------------------
Salesforce Userid (reqd)   : []salesforce_login_email
Salesforce Password (reqd) : []salesforce_login_password
Security Token (reqd)      : []Security_Token
Login Endpoint             : [https://login.salesforce.com]
Consumer ID                : []
Consumer Secret Key        : []
```

c) 以下のようにして、TLS 接続のための証明書ストアの値を入力します。

TLS 接続に必要な最小限の値は、TLS 証明書の鍵ストアのパスと、鍵ストアのパスワードです。 トラストストアのパスやパスワードが指定されていない場合は、鍵ストアとパスワードのパラメーターがトラストストアとパスワードに使用されます。 IBM MQ キュー・マネージャー接続に TLS を使用している場合は、同じ鍵ストアを使用できます。

```
Certificate stores for TLS connections
--------------------------------------
Personal keystore for TLS certificates : []path_to_keystore, for example: /var/mqm/qmgrs/
SQM1/ssl/key.jks
Keystore password          : []keystore_password
Trusted store for signer certificates : []
Trusted store password     : []
Use TLS for MQ connection  : [N]
```

d) 以下のようにして、IBM MQ Bridge to Salesforce の動作を構成するための値を入力します。

いずれの値も変更したり、指定したりする必要はありませんが、プッシュ・トピックやプラットフォーム・イベント名が分かっている場合は、ここで追加してください。 また、ブリッジを実行する準備ができたときに、後でコマンド行に追加することもできます。 構成ファイルまたはコマンド行にログ・ファイルを指定する必要があります。

```
Behaviour of bridge program
---------------------------
PushTopic Names           : []
Platform Event Names      : []
MQ Monitoring Frequency   : [30]
At-least-once delivery? (Y/N) : [Y]
Subscribe to MQ publications for platform events? (Y/N) : [N]
Publish control data with the payload? (Y/N) : [N]
Delay before starting to process events : [0]
Runtime logfile for copy of stdout/stderr : []
```

8. オプション: IBM MQ サービスを作成して、プログラムの実行を制御します。 サンプルの `mqsfbService.mqsc` ファイルを編集して、新しく作成された構成ファイルを指すようにし、その他の変更をコマンド・パラメーターに対して行います。

   ```
   cat modified mqsfbService.mqsc | runmqsc SQM1
   ```

9. オプション: IBM MQ コンソールの概要の説明に従って、IBM MQ Console をセットアップします。
10. オプション: IBM MQ Bridge to Salesforce が root 化されていないユーザーとして実行されるように構成します。

    root 化されていないユーザーとして IBM MQ Bridge to Salesforce を実行できるようにするには (例えば、 *root* 化されていないコンテナー内で)、ブリッジ・プロセスを実行するユーザーの読み取り/書き込みアクセス権限を確保するために、 Java `userRoot` および `systemRoot` ディレクトリーを正しく設定する必要があります。 これを行うには、以下の JVM プロパティーを設定します。

    ```
    export MQSFB_EXTRA_JAVA_OPTIONS="-
    Djava.util.prefs.userRoot=directory_with_read_write_access"
    ```

    ```
    export MQSFB_EXTRA_JAVA_OPTIONS="-
    Djava.util.prefs.systemRoot=directory_with_read_write_access"
    ```

タスクの結果

IBM MQ Bridge to Salesforce が Salesforce プッシュ・トピックおよびプラットフォーム・イベントをサブスクライブし、それらを IBM MQ ネットワークにパブリッシュするために使用する構成ファイルを作成しました。

次のタスク

883 ページの『IBM MQ Bridge to Salesforce の実行』のステップに従います。

関連タスク

IBM MQ Bridge to Salesforce のトレース

IBM MQ Bridge to Salesforce のモニター

関連資料

runmqsfb (IBM MQ Bridge to Salesforce の実行)

Deprecated Linux IBM MQ Bridge to Salesforce の追加の構成オプション

このオプションは IBM MQ 9.2.0、「インバウンド」（IBM MQ アプリケーションに公開される、Salesforce から生成されたイベント）と「アウトバウンド」（IBM MQ アプリケーションに送信されるイベントを公開する Salesforce）の作業を扱う。2 つの主要なクラスの追加トポロジーを可能にします。 さらに、トレースとロギングの処理方法も変更されています。

IBM MQ 9.1.0 IBM MQ Bridge to Salesforce からの変更点

IBM MQ 9.2.0 以降、デフォルトでは、ログ・ファイルの循環が可能になったという点以外に、IBM MQ 9.1.0 のブリッジからの動作変更はありません。 詳しくは、875 ページの『ログの循環』 を参照してください。

主な変更は、キュー・マネージャーで複数のブリッジ・インスタンスをサポートできるようになったことです。 この機能とその他の追加トポロジーを有効にするために、いくらかの構成変更を手動で行う必要があります。

追加の構成オプションの詳細については、runmqfsb を参照してください。改訂後の構成情報の例については、876 ページの『IBM MQ Bridge to Salesforce の構成出力例』 を参照してください。

インバウンド処理の分離

ブリッジの複数のインスタンスは、Salesforce から IBM MQ へのインバウンド作業を処理できますが、これらは Salesforce プッシュ・トピックおよびイベントの独立したセットで処理する必要があります。 そうしないと、IBM MQ アプリケーションよってイベントが繰り返し生成される可能性があります。イベントの複製を停止するクロス・ブリッジ・プロトコルがないからです。 各インスタンスがそれぞれの独自の構成可能同期キューを使用して **ReplayId** を保持します。

こうした設定が便利なのは、以下のような場合です。

- Salesforce のトピックごとに別々のセキュリティー許可がある場合。 ブリッジ・インスタンスごとに、Salesforce にアクセスするための別々の資格情報を設定できます。
- Salesforce からのワークロードが大きすぎて 1 つのブリッジでは処理できない、という可能性がある場合。 トピックを分割して、A から M を 1 つのブリッジ、N から Z をもう 1 つのブリッジ、といった具合に調整できます。

アウトバウンド処理の共有

ブリッジの複数インスタンスで、IBM MQ から Salesforce に送信されるアウトバウンド処理に対応できます。 1 つのブリッジ・インスタンスで障害が発生しても、同じキュー・マネージャーの同じトピックにサブスクライブしている他のインスタンスがパブリケーションの処理を続行できます。

注 : そのために、IBM MQ のトピック構成を変更する必要はありません。

インスタンス同士の協力関係をセットアップする時に、Salesforce からのインバウンド処理に最大で 1 つのインスタンスが対応するようにしなければなりません。そのインスタンスには、同期キューへの排他的アクセスが必要だからです。

こうした設定が便利なのは、以下の点に不安がある場合です。

- IBM MQ からのワークロード。 Salesforce への要求は同期型なので、ブリッジは、1 つのメッセージを処理している間に新しい処理に対応することができません。 コンシューマーを複数用意すれば、そのような状況を緩和できます。
- 可用性のアーキテクチャー。 例えば、今は複数のインスタンスをそれぞれ別々のデータ・センターで実行することが可能で、フェイルオーバーと災害復旧のオプションが改善されています。 IBM MQ クライアントとして実行することで、ブリッジをキュー・マネージャーの場所から分離することもできます。

トレースとデバッグの相互作用

IBM MQ 9.2.0 以降も、デバッグ・フラグは IBM MQ 9.1.0 と同様に機能し続けます。 *-d1* を指定すると、ブリッジのデバッグ情報が得られ、*-d2* を指定すると、前提条件コンポーネントのデバッグ・ロギングがオンになります。 ただし、ブリッジの開始時に IBM MQ トレースを有効すると、*-d2* レベルのレポートが自動的にオンになります。

ログの循環

IBM MQ 9.2.0 以降、ログ・ファイルのデフォルトの動作では、3 つのログ・ファイルを使用できます。それぞれのサイズは 2 MB です。 追加の構成プロパティーを使用して、この値をオーバーライドすることも

できます。 ログ・ファイルの既存の構成属性やコマンド・ライン・パラメーターがログの基底名として取り込まれ、索引が追加されます。

構成するログ・ファイルは、以下のようになります。

- ファイル・タイプがない場合は、ファイル名の末尾に索引が追加されます。

  ログ・ファイルを abc に設定すると、abc.0、abc.1 という具合にログが生成されます。
- ファイル・タイプがある場合は、ファイル・タイプの前に索引が挿入されます。

  ログ・ファイルを abc.log に設定すると、abc.0.log、abc.1.log という具合にログが生成されます。

注 :

1. ブリッジは任意のユーザー許可を使用して、実行できるため、特定のディレクトリー(例えば、/var/mqm/qmgrs/<qm>/errors など)をログに強制することはできません。
2. 今後も *stdout* と *stderr* のストリームに同じ情報が書き込まれます。
3. 個々のログ・ファイルを再び開くと、基本的な構成情報が再び出力されます。 その情報はいつでも使用できます。プログラムの開始時にだけ出力されるということではありません。

ログの保存

IBM MQ 9.2.0 トポロジーが導入されたので、1 つのキュー・マネージャーに対してブリッジの複数インスタンスを実行する可能性が高くなりました。

インスタンス同士が干渉し合ったりブリッジの前回の実行が上書きされたりするのを防止するために、.0 のログがすでに存在している場合は、ブリッジが開始されません。

ブリッジの開始前にログの以前のコピーを削除したりファイル名にタイム・スタンプのようなものを追加したりする開始手順が必要になります。

関連タスク

868 ページの『Salesforce プッシュ・トピックおよびプラットフォーム・イベントで使用するための IBM MQ の構成』
この情報を使用して、 IBM MQ Bridge to Salesforce を構成してから実行することにより、 Salesforce および IBM MQ ネットワークへのセキュリティーと接続をセットアップします。

IBM MQ Bridge to Salesforce のトレース

関連資料

runmqfsb

Deprecated Linux IBM MQ Bridge to Salesforce の構成出力例

構成出力の例です。IBM MQ 9.1.0 IBM MQ Bridge to Salesforce からの変更点を示しています。

```
IBM MQ Bridge to Salesforce
5724-H72 (C) Copyright IBM Corp. 2017, 2025.
Level : <<unknown>>

Enter new values for the configuration attributes. The
current settings are shown.
Press ENTER to accept current values; use SPACE+ENTER
to clear values.

Connection to Queue Manager
---------------------------
Queue Manager or JNDI CF   : [V9000_A]
MQ Base Topic              : [/sf]
MQ Channel                 : []
MQ Conname                 : []
MQ Publication Error Queue : [SYSTEM.SALESFORCE.DEADQ]
MQ Replay Status Queue     : [SYSTEM.SALESFORCE.SYNCQ]
MQ CCDT URL                : []
JNDI implementation class  : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL          : []
MQ Userid                  : []
```

```
MQ Password                : []

Connection to Salesforce
------------------------
Salesforce Userid (reqd)   : [johndoe@<yourenterprise>.com]
Salesforce Password (reqd) : [********]
Security Token             : [********]
Login Endpoint             : [https://login.salesforce.com]
Consumer Key               : [3MVG9HxRZv05HarQhSy89qSKYNr1gDcv1wE3zN5kyFAa4Wxt]
Consumer Secret            : [********]

Certificate stores for TLS connections
--------------------------------------
Personal keystore for TLS certificates : [/var/mqm/ssl/key.jks]
Keystore password          : [********]
Trusted store for signer certificates : []
Trusted store password     : []
Use TLS for MQ connection  : [N]

Event processing
----------------
PushTopic Names            : []
Platform Event Names       : []
At-least-once delivery for Salesforce events? (Y/N) : [N]
At-least-once delivery for MQ publications? (Y/N) : [N]
Subscribe to MQ publications for platform events? (Y/N) : [Y]
Publish control data with the payload? (Y/N) : [Y]
Treat unknown Salesforce topic as warning (Y/N) : [N]

Behaviour of bridge program
---------------------------
Bridge unique identifier   : []
MQ Monitoring Frequency    : [30]
Delay before starting to process events : [0]
Continue to retry after maximum reconnection attempts (Y/N) : [N]
Runtime logfile for copy of stdout/stderr : [/tmp/runmqsfb.log]
Number of logfiles         : [3]
Maximum size of each logfile : [2097152]
Done.
```

関連資料

runmqfsb

Deprecated Linux Salesforce プラットフォーム・イベントのイベント・メッセージの作成

IBM MQ を構成し、IBM MQ Bridge to Salesforce parameters を入力して構成ファイルを作成し、ブリッジを使用して Salesforce プラットフォーム・イベントのイベント・メッセージを作成できます。

始める前に

- x86-64 Linux プラットフォーム上の IBM MQ インストール済み環境に **MQSeriesSFBridge** パッケージをインストールしました。

このタスクについて

このタスクでは、IBM MQ Bridge to Salesforce コンフィギュレーションファイルを作成し、Salesforce プラットフォームイベント用のイベントメッセージを作成できるよう、正常に Salesforce と IBM MQ 接続するために必要な最小限のセットアップを行います。 すべてのパラメーターの意味とオプションについて詳しくは、runmqsfb コマンドを参照してください。 独自のセキュリティー要件を考慮し、デプロイメントに適したパラメーターをカスタマイズする必要があります。

プッシュ・トピックとプラットフォーム・イベントのサブスクライブの構成を作成するには、870 ページの『IBM MQ Bridge to Salesforce の構成』を参照してください。

Salesforce プラットフォーム・イベントのイベント・メッセージの作成

IBM MQ アプリケーションを使用して、キュー・マネージャー・トピック/*root*/mqtosfb/event/+上に書き込まれるメッセージを作成できます。 ブリッジはトピックをサブスクライブし、メッセージからコンテンツを取得し、それを使用して Salesforce プラットフォーム・イベントのイベント・メッセージをパブ

リッシュします。 プラットフォーム・イベントについて詳しくは、 Salesforce 開発者資料の「 プラットフォーム・イベントによるカスタム通知の配信 」を参照してください。

ブリッジを有効にしてイベント・メッセージを作成するには、プッシュ・トピックとプラットフォーム・イベントのサブスクライブに使用された属性に加えて 2 つの属性を指定する必要があります。

- **キュー・マネージャーへの接続**のブリッジ構成属性に **MQ Publication Error Queue** の名前を作成して追加します。
- **ブリッジ・プログラムの動作**を定義するためのブリッジ構成属性で、**Subscribe to MQ publications for platform events** オプションを 「*Y*」に設定します。

ブリッジを使用してそのプラットフォーム・イベントのイベント・メッセージを作成できるようにするには、まず、Salesforce でプラットフォーム・イベントを作成し、コンテンツ・フィールドを定義する必要があります。 プラットフォーム・イベント名およびそのコンテンツによって、ブリッジによって処理される IBM MQ メッセージをどのようにフォーマット設定する必要があるかが決定されます。 例えば、Salesforce プラットフォーム・イベント **Object name** が *MQPlatformEvent1* であり、2 つのカスタム定義フィールドが **API name** *MyText__c* および *Name__c* のテキスト・フィールドである場合、*/root/*mqtosfb/event/MQPlatformEvent1__e トピックにパブリッシュされる IBM MQ メッセージは、以下のように正しい形式の JSON でなければなりません。

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

メッセージは、IBM MQ Bridge to Salesforce が MQFMT_STRING 形式のメッセージ本体として認識できるようにフォーマット設定する必要があります。

ステップ 880 ページの『7』 を参照して、Salesforce でプラットフォーム・イベントを作成するか、イベント・メッセージを作成するプラットフォーム・イベントが既にある場合はこのステップをスキップします。 IBM MQ メッセージを、Salesforce プラットフォーム・イベントで設定されているフィールドと一致するようにフォーマット設定する必要があります。 Salesforce プラットフォーム・イベント内のフィールドは、オプションまたは必須として指定できます。 詳しくは、 Salesforce 開発者資料の「 プラットフォーム・イベント・フィールド 」を参照してください。

ブリッジが実行されている場合、ブリッジは指定された IBM MQ トピックをサブスクライブします。

- ブリッジ構成で **At-most-once** サービス品質を指定すると、ブリッジが作成するサブスクリプションは非永続になります。 ブリッジが実行されていないときに IBM MQ アプリケーションによって作成されたパブリケーションは処理されません。
- ブリッジ構成で **At-least-once** サービス品質を指定すると、ブリッジが作成するサブスクリプションは永続的になります。 これは、ブリッジが実行中でないときに IBM MQ アプリケーションによって作成されたパブリケーションをブリッジが処理できることを意味します。 永続サブスクリプションには、既知のサブスクリプションおよびクライアント ID が必要です。 ブリッジは、サブスクリプション名として *D_SUB_RUNMQSFB* を使用し、クライアント ID として *runmqsfb_1* を使用します。

ブリッジが Salesforce プッシュ・トピックおよびプラットフォーム・イベントのサブスクライブのために使用され、イベント・メッセージの作成のためには使用されない場合、構成が変更されると、ブリッジは永続サブスクリプションの削除を試行し、サブスクリプションが孤立します。

ブリッジが作成した永続サブスクリプションを削除するには、以下のようにします。

IBM MQ Explorer を使用する。
ブリッジが使用しているキュー・マネージャーの**サブスクリプション・フォルダー**を開き、トピック・ストリングが /sf/mqtosfb/event+である*:D_SUB_RUNMQSFB* で終了するサブスクリプション名を探します。 サブスクリプション名を右クリックして、「削除」をクリックします。 サブスクリプションが使用中であることを示すエラーが表示された場合は、ブリッジが実行中である可能性があります。 ブリッジを停止し、サブスクリプションの削除を再試行してください。

runmqsc を使用した、サブスクリプションの検索と削除。
runmqsc インターフェースを開始し、DISPLAY SUB (*) を実行します。 *:D_SUB_RUNMQSFB* で終わるサブスクリプション名 **SUB** を探します。 delete サブコマンドを発行し、削除するサブスクリプションの **SUBID** を含めます。 例: DELETE SUB SUBID(414D5120514D312020202020202020205C589459987E8620)

ブリッジを停止し、その後 At-most-once サービス品質を使用して開始します。

At-least-once サービス品質 `At-least-once delivery? (Y/N) :[Y]` を使用してブリッジを開始した場合、作成されるサブスクリプションは永続的です。 サブスクリプションを削除するには、構成ファイルでサービス品質を `At-least-once delivery? (Y/N) :[N]` に変更し、ブリッジを再始動します。 永続サブスクリプションが削除され、非永続サブスクリプションが作成されます。

手順

1. キュー・マネージャーを作成して開始します。
   a) キュー・マネージャー (例えば、PEQM1) を作成します。

   ```
   crtmqm PEQM1
   ```

   b) キュー・マネージャーを開始します。

   ```
   strmqm PEQM1
   ```

2. **注 :** 既存のログインとセキュリティーの Salesforce 資格情報、および自己署名証明書を使用する場合は、ステップ 4 にスキップします。

   オプション: ご使用の Salesforce アカウント用のセキュリティー・トークンを作成します。

   a) Salesforce アカウントにログインします。

   b) ヘルプ項目 Salesforce help: Reset your security token のステップに従って、セキュリティー・トークンを作成またはリセットします。

3. Salesforce で自己署名セキュリティー証明書を作成します。

   a)「**Force.com ホーム**」ページの「**管理**」メニューから「**セキュリティのコントロール**」を選択し、「**証明書と鍵の管理**」を選択します。

   「**証明書と鍵の管理**」ページが開きます。

   b)「**自己署名証明書の作成**」をクリックします。

   「**証明書**」ページが開きます。

   c)「**Label (ラベル)**」フィールドに証明書の名前を入力し、Tab キーを押してから「**保存**」をクリックします。

   証明書と鍵の詳細情報が表示されます。

   d)「**Back to list: Certificates and keys (一覧に戻る: 証明書と鍵の管理)**」をクリックします。

   e)「**キーストアにエクスポート**」をクリックします。

   f) 鍵ストアのパスワードを入力し、「**エクスポート**」をクリックします。

   g) エクスポートした鍵ストアをローカル・ファイル・システムに保存します。

4. IBM 鍵管理 GUI を使用して、Salesforce からエクスポートした鍵ストアを開き、署名者証明書を追加します。

   a) **strmqikm** コマンドを実行して、IBM 鍵管理 GUI を開きます。 詳しくは、runmqckm、runmqakm、および strmqikm を使用したデジタル証明書の管理を参照してください。

   b)「**鍵データベース・ファイルを開く**」をクリックし、Salesforce 鍵ストアのロケーションを参照します。

   c)「**開く**」をクリックして、「**鍵データベース・タイプ**」から「**JKS**」が選択されていることを確認し、「**OK**」をクリックします。

   d) ステップ 3f で鍵ストア用に作成したパスワードを入力してから、「**OK**」をクリックします。

   e)「**鍵データベースの内容**」オプションから「**署名者証明書**」を選択します。

   f)「**挿入**」をクリックします。

   g)「**CA 証明書の追加**」リストから「**Verisign Inc.**」チェック・ボックスを選択し、「**OK**」をクリックします。

5. オプション: Salesforce アカウントで IBM MQ Bridge to Salesforce のアプリ接続を作成して、OAuth コンシューマー鍵と秘密情報を生成します。

   実稼働環境で IBM MQ Bridge to Salesforce を使用する場合は、「**コンシューマー鍵**」 コードと 「**コンシューマー秘密情報**」 コードが必要です。

   a)「**作成**」を選択し、「**Force.com ホーム**」ページの「**ビルド**」メニューから「**アプリケーション**」を選択します。

   「**アプリケーション**」ページが開きます。

   b)「**接続アプリケーション**」セクションから「**新規**」をクリックします。

   「**新規接続アプリケーション**」ページが開きます。

   c) ご使用の IBM MQ Bridge to Salesforce の名前 (例えば、**MQBridgeToSalesforce**) を「**接続アプリケーション名**」に入力します。

   d)「**API Name (API 名)**」を入力します。

   タブで次のフィールドに移動すると、「**接続アプリケーション名**」が「**API Name (API 名)**」名前フィールドにコピーされます。

   e)「**Contact Email (連絡先 E メール)**」を入力します。

   f)「**API (OAuth 設定の有効化)**」セクションで「**OAuth 設定の有効化**」オプションを選択します。

   そのセクションの他のオプションが表示されます。

   g)「**コールバック URL**」 (例えば、`https://www.ibm.com`) を入力します。

   h)「**選択した OAuth 範囲**」サブセクションの「**利用可能な OAuth 範囲**」リストから「**フルアクセス (full)**」オプションを選択し、「**追加**」をクリックして、「**選択した OAuth 範囲**」リストに全アクセス権限を追加します。

   i) **保存** をクリックします。

   j)「**次へ進む**」をクリックします。

   k)「**コンシューマ鍵**」と「**コンシューマの秘密**」のコードを書き留めます。

6. 必要な同期キューおよびエラー・キューをキュー・マネージャー上に作成します。

   ```
   cat /opt/mqm/mqsf/samp/mqsfbSyncQ.mqsc | runmqsc PEQM1
   ```

   同期キューによって、アプリケーション間のイベント状態が維持されたり、キュー・マネージャーが再始動されたりします。 キュー上にあると予想されるメッセージは 1 つだけなので、キュー項目数は少なくて済みます。 このキューに対して一度に実行できるブリッジのインスタンスは 1 つだけなので、デフォルト・オプションは排他的アクセス用に設定されます。 ブリッジを使用してプラットフォーム・イベントのイベント・メッセージを作成できるようにするには、まず、エラー・キューを作成する必要があります。 エラー・キューは、Salesforce で正常に処理できないメッセージに使用されます。 ステップ 881 ページの『8.a』 に示すように、エラー・キュー名をブリッジ構成パラメーター・セクション **Connection to Queue Manager** に追加する必要があります。

7. オプション: ご使用の Salesforce アカウントでプラットフォーム・イベント・オブジェクトを作成します。

   a)「**Force.com ホーム (Force.com Home)**」ページの「**開発 (Develop)**」メニューから「**プラットフォーム・イベント (Platform Events)**」を選択し、「**新規プラットフォーム・イベント (New Platform Event)**」をクリックします。

   「**新規プラットフォーム・イベント (New Platform Event)**」ページが開きます。

   b)「**ラベル (Label)**」フィールドおよび「**複数ラベル (Plural Label)**」フィールドに入力します。

   c)「**保存**」をクリックします。

   「**プラットフォーム・イベント定義の詳細 (Platform Event Definition Detail)**」ページが開きます。

   d) **Custom Fields & Relationships** を定義します。

   例えば、ラベルが *MyText* および *Name* の 2 つのフィールドを追加し、「**データ・タイプ (Data Type)**」 のフィールド長をそれぞれ *Text(64)* および *Text(32)* に設定する場合があります。

   プラットフォーム・イベントを作成し、それに対して **Custom Fields and Relationships** を定義しました。 ブリッジで処理するメッセージを書き込むことができる IBM MQ トピックとして、プラ

ットフォーム・イベントの プラットフォーム・オブジェクト名 または *API* 名 を使用します。 例えば、**AMQSPUBA** サンプルを使用して、以下の JSON 形式のメッセージを /sf/mqtosfb/event/*Salesforce Platform Object Name*/*API name* トピックに追加できます。

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

AMQSPUBA サンプルを実行して、ブリッジの開始後にメッセージを作成できます。 *MQ installation location*/samp/bin ディレクトリーから、以下のコマンドを発行します:

```
./amqspub /sf/mqtosfb/event/Salesforce Platform Object Name/API name PEQM1
```

プロンプトが表示されたら、JSON 形式のメッセージを入力します。

8. IBM MQ、Salesforce、および IBM MQ Bridge to Salesforce の動作に対する接続とセキュリティーのパラメーターを使用して、構成ファイルを作成します。

```
runmqsfb -o new_config.cfg
```

既存の値が大括弧内に表示されます。 Enter を押すと、既存の値を受け入れます。 Space を押してから Enter を押すと、値をクリアします。 値を入力してから Enter を押すと、新規の値を追加します。

a) 以下のようにして、キュー・マネージャー PEQM1 への接続のための値を入力します。

接続に必要な最小値は、キュー・マネージャー名、IBM MQ ベース・トピック・ルート、エラー・キュー名、およびチャネル名です。

```
Connection to Queue Manager
---------------------------
Queue Manager or JNDI CF   : []PEQM1
MQ Base Topic              : []/sf
MQ Channel                 : []A channel you have defined or for example
SYSTEM.DEF.SVRCONN
MQ Conname                 : []
MQ Publication Error Queue : [SYSTEM.SALESFORCE.ERRORQ]
MQ CCDT URL                : []
JNDI implementation class  : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL          : []
MQ Userid                  : []
MQ Password                : []
```

注: ローカルに接続している場合は、チャネル名は必要ありません。 構成ファイルにキュー・マネージャー名とベース・トピックを指定する必要はありません。 これらは、後でブリッジを実行するときにコマンド行に含めることができます。

b) 以下のようにして、Salesforce への接続のための値を入力します。

接続に必要な最小限の値は、Salesforce のユーザー ID、パスワード、セキュリティー・トークン、およびログイン・エンドポイントです。 実稼働環境では、OAuth セキュリティーのコンシューマー鍵と秘密情報を追加できます。

```
Connection to Salesforce
------------------------
Salesforce Userid (reqd)   : []salesforce_login_email
Salesforce Password (reqd) : []salesforce_login_password
Security Token (reqd)      : []Security_Token
Login Endpoint             : [https://login.salesforce.com]
Consumer ID                : []
Consumer Secret Key        : []
```

c) 以下のようにして、TLS 接続のための証明書ストアの値を入力します。

TLS 接続に必要な最小限の値は、TLS 証明書の鍵ストアのパスと、鍵ストアのパスワードです。 トラストストアのパスやパスワードが指定されていない場合は、鍵ストアとパスワードのパラメーターがトラストストアとパスワードに使用されます。 IBM MQ キュー・マネージャー接続に TLS を使用している場合は、同じ鍵ストアを使用できます。

```
Certificate stores for TLS connections
--------------------------------------
Personal keystore for TLS certificates : []path_to_keystore, for example: /var/mqm/qmgrs/
PEQM1/ssl/key.jks
```

```
Keystore password          : []keystore_password
Trusted store for signer certificates : []
Trusted store password     : []
Use TLS for MQ connection  : [N]
```

d) 以下のようにして、IBM MQ Bridge to Salesforce の動作を構成するための値を入力します。

ブリッジを使用してイベント・メッセージを作成するには、**Subscribe to MQ publications for platform events** オプションをデフォルトの「*N*」から「*Y*」に変更する必要があります。また、構成ファイルまたはコマンド行で、ログ・ファイルを指定する必要もあります。

```
Behaviour of bridge program
---------------------------
PushTopic Names            : []
Platform Event Names       : []
MQ Monitoring Frequency    : [30]
At-least-once delivery? (Y/N) : [Y]
Subscribe to MQ publications for platform events? (Y/N) : [Y]
Publish control data with the payload? (Y/N) : [N]
Delay before starting to process events : [0]
Runtime logfile for copy of stdout/stderr : []
```

9. オプション: IBM MQ サービスを作成して、プログラムの実行を制御します。 サンプルの `mqsfbService.mqsc` ファイルを編集して、新しく作成された構成ファイルを指すようにし、その他の変更をコマンド・パラメーターに対して行います。

```
cat modified mqsfbService.mqsc | runmqsc PEQM1
```

10. オプション: IBM MQ コンソールの概要の説明に従って、IBM MQ Console をセットアップします。
11. オプション: IBM MQ Console インスタンスにウィジェットを追加して構成し、Salesforce データを表示します。
    a)「**ウィジェットの追加**」をクリックします。

    新しいウィジェットが開きます。

    b)「**グラフ**」を選択します。

    c) 新規ウィジェットのタイトル・バーにある「**ウィジェットの構成**」アイコンをクリックします。

    d) オプション:「**ウィジェット・タイトル**」を入力します。

    e)「**モニターするリソース (Resource to monitor)**」の「**ソース**」ドロップダウン・メニューから「**Salesforce ブリッジ**」を選択します。

    f)「**リソース・クラス**」ドロップダウン・メニューから「**ブリッジ状況 (Bridge Status)**」を選択します。

    g)「**リソース・タイプ**」ドロップダウン・メニューから「**MQ 作成のプラットフォーム・イベント (MQ-created Platform Events)**」を選択します。

    h)「**リソース・エレメント**」ドロップダウン・メニューから、「**MQ 作成のプラットフォーム・イベントの合計 (Total MQ-created Platform Events)**」を選択します。

    i) **保存** をクリックします。

    IBM MQ 作成のプラットフォーム・イベントの総数を表示するように IBM MQ Console を構成しました。 ブリッジが実行中で、`/sf/mqtosfb/event/`*Salesforce Platform Object Name/API name* トピックへのメッセージの書き込みを開始すると、このウィジェットには、ブリッジが作成したメッセージ・イベントの合計数が表示されます。

Deprecated IBM MQ Bridge to Salesforce のメッセージの形式およびエラー・メッセージ

IBM MQ Bridge to Salesforce によって処理されるメッセージのフォーマット設定に関する情報。

アプリケーションは、特定のキュー・マネージャーのトピック(例えば、`/`*root*`/mqtosfb/event/MQPlatformEvent1__e`)にメッセージを書き込みます。 ブリッジはトピックをサブスクライブし、メッセージからコンテンツを取得し、それを使用して Salesforce プラットフォーム・イベントのイベント・メッセージをパブリッシュします。

ブリッジを使用してそのプラットフォーム・イベントのイベント・メッセージを作成できるようにするには、まず、Salesforce でプラットフォーム・イベントを作成し、コンテンツ・フィールドを定義する必要があります。 プラットフォーム・イベント名およびそのコンテンツによって、ブリッジによって処理される IBM MQ メッセージをどのようにフォーマット設定する必要があるかが決定されます。 例えば、Salesforce プラットフォーム・イベント **Object name** が *MQPlatformEvent1* であり、2 つのカスタム定義フィールドが **API name** *MyText__c* および *Name__c* のテキスト・フィールドである場合、*/root/*mqtosfb/event/MQPlatformEvent1__e トピックにパブリッシュされる IBM MQ メッセージは、以下のように正しい形式の JSON でなければなりません。

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

ブリッジによって消費または生成されるメッセージは、JSON 形式のテキスト (MQSTR) メッセージです。 入力メッセージは単純な JSON であり、プログラムでは文字列の連結を使用して、入力メッセージを生成できます。

エラー・メッセージ

例えば、メッセージがテキスト形式でない場合、または Salesforce によって、例えば、プラットフォーム・イベント名が存在しない場合に、ブリッジによってエラーが検出されることがあります。 入力メッセージの処理中にエラーが発生した場合、メッセージは、エラーを記述するプロパティーとともにブリッジ・エラー・キューに移動されます。 エラーは、ブリッジの *stderr* ストリームにも書き込まれます。

Salesforce によって生成されるエラーは、JSON です。 以下に、形式が無効なメッセージによって発生するいくつかのエラーを示します。

プラットフォーム・イベントの内容が正しくありません。状況 400 テキスト

```
[{"message":"No such column 'Name__c' on sobject of type
MQPlatformEvent2__e","errorCode":"INVALID_FIELD"}
```

プラットフォーム・イベント名が無効です。状況 404 テキスト

```
{"errorCode":"NOT_FOUND","message":"The requested resource does not exist"}
```

JSON が正しくありません。状況 400 テキスト

```
{"errorCode":"NOT_FOUND","message":"The requested resource does not exist"}
```

メッセージが JSON ではありません。状況 400 テキスト

```
[{"message":
  "Unexpected character ('h' (code 104)): expected a valid value (number, String, array,
object, 'true', 'false' or 'null') at [line:1, column:2]",
  "errorCode":"JSON_PARSER_ERROR"}
```

テキスト・メッセージではありません (Salesforce に送信されません)。

```
Error: Publication on topic ' /sf/mqtosfb/event/MQPlatformEvent1' does not contain a text
formatted message
```

Deprecated Linux IBM MQ Bridge to Salesforce の実行

IBM MQ Bridge to Salesforce を実行して、Salesforce および IBM MQ に接続します。 接続されると、ブリッジによって、Salesforce トピックに対するサブスクリプションが作成され、IBM MQ トピックにメッセージがリパブリッシュされます。 ブリッジは、Salesforce プラットフォーム・イベントのイベント・メッセージも作成できます。

始める前に

以下のタスクの構成ステップを完了しておきます。

- 870 ページの『IBM MQ Bridge to Salesforce の構成』

- 877 ページの『Salesforce プラットフォーム・イベントのイベント・メッセージの作成』

.

このタスクについて

前のステップで作成した構成ファイルを使用して、IBM MQ Bridge to Salesforce を実行します。 構成ファイルにすべての必須パラメーターを組み込んではいない場合、コマンド行に必ずそれらを組み込んでください。

手順

1. サブスクライブ先の Salesforce のプッシュ・トピックまたはプラットフォーム・イベント、あるいはイベント・メッセージを作成するプラットフォーム・イベントを定義します。
2. IBM MQ Bridge to Salesforce を開始して、Salesforce とキュー・マネージャーに接続します。 ブリッジを実行して Salesforce イベントをサブスクライブする場合は、ステップ 1 で定義したプッシュ・トピックまたはプラットフォーム・イベントの名前を組み込みます。

   ```
   runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName
   ```

   ブリッジが接続されると、次のようなメッセージが返されます。

   - ブリッジを使用して Salesforce プッシュ・トピックおよびプラットフォーム・イベントをサブスクライブする場合:

     ```
     Successful connection to queue manager QM1
     Warning: Subscribing to MQ-created platform events is not enabled.
     Successful login to Salesforce at https://eu11.salesforce.com
     Ready to process events.
     ```

   - ブリッジを使用して、Salesforce プラットフォーム・イベントのイベント・メッセージを作成する場合:

     ```
     Successful connection to queue manager QM1
     Successful login to Salesforce at https://eu11.salesforce.com
     Successful subscription to '/sf/mqtosfb/event/+' for MQ-created platform events
     Ready to process events.
     ```

3. オプション: ブリッジの実行後に、接続が正常に行われなかったことを示すメッセージが返された場合は、キュー・マネージャーと Salesforce への接続のトラブルシューティングを行います。
   a) デバッグ・オプション 1 を指定して、デバッグ・モードでコマンドを発行します。

      ```
      runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName -d 1
      ```

      ブリッジによって接続のセットアップがステップスルーされ、処理メッセージが簡潔モードで表示されます。
   b) デバッグ・オプション 2 を指定して、デバッグ・モードでコマンドを発行します。

      ```
      runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName -d 2
      ```

      ブリッジによって接続のセットアップがステップスルーされ、処理メッセージが冗長モードで表示されます。 全出力がログ・ファイルに書き込まれます。
4. Salesforce インターフェースを使用してイベントを生成し、データベース内のレコードを変更します。
5. IBM MQ Console に移動し、前のタスクで構成したウィジェットに、プッシュ・トピックに対する変更が表示されていることを確認します。

次のタスク

MQSFB_EXTRA_JAVA_OPTIONS 変数を使用して、例えば、IBM MQ トレースを有効化する JVM プロパティーを渡します。 詳しくは、IBM MQ Bridge to Salesforce のトレースを参照してください。

関連タスク

IBM MQ Bridge to Salesforce のモニター

関連資料

runmqsfb (IBM MQ Bridge to Salesforce の実行)

Deprecated Linux z/OS MQ Adv. ブロックチェーンで使用するための IBM MQ の構成

IBM MQ Bridge to blockchain をセットアップして実行すると、 IBM MQ Advanced キュー・マネージャーまたは IBM MQ Advanced for z/OS Value Unit Edition キュー・マネージャーと IBM Blockchain を安全に接続できます。 IBM MQ Advanced キュー・マネージャーまたは IBM MQ Advanced for z/OS VUE キュー・マネージャーに接続するメッセージ・アプリケーションを使用して、 このブリッジでブロックチェーンに非同期で接続し、ブロックチェーンのリソースの状態を調べて更新することができます。

始める前に

注 :

- Deprecated IBM MQ Bridge to blockchain は、2022 年 11 月 22 日以降のすべてのリリースで非推奨になりました (US Announcement letter 222-341 を参照)。 Blockchain 接続は、 IBM App Connect を使用して、または IBM Cloud Pak for Integration で使用可能な App Connect 機能を介して実現できます。
- V 9.3.2 Removed Continuous Delivery の場合、 IBM MQ Bridge to blockchain は IBM MQ 9.3.2 で製品から削除されています。
- LTS IBM は、今後のフィックスパックで Long Term Support リリースから機能を削除する予定です。 この変更の影響を受けるアプリケーションがある場合は、 IBM サポートにお問い合わせください。

  重要 : Hyperledger Composer 上に構築された IBM MQ Bridge to blockchain はサポートされなくなりました。

  Hyperledger Fabric 上に構築された IBM MQ Bridge to blockchain を使用するには、 IBM MQ 9.1.4 以降を実行する必要があります。
- IBM MQ Bridge to blockchain は、以下のキュー・マネージャーにのみ接続できます。
  - Linux IBM MQ Advanced、または
  - z/OS IBM MQ Advanced for z/OS VUE
- キュー・マネージャーは、ブリッジと同等以上のコマンド・レベルでなければなりません (IBM MQ 9.3.0 など)。
- IBM MQ Bridge to blockchain は、Hyperledger Fabric 1.4 アーキテクチャーに基づく ブロックチェーン・ネットワークで使用できます。

このタスクについて

ブロックチェーンとは、分散型の共有デジタル台帳であり、ネットワーク上のピアの間で合意されたトランザクションを表すブロックがチェーン状に連なったもので構成されます。 チェーン内のそれぞれのブロックは前のブロックにリンクしているため、最終的には最初のトランザクションにリンクしています。

IBM Blockchain は、Hyperledger Fabric 上に構築され、Docker でローカルに開発するか、IBM Cloud のコンテナー・クラスターで開発できます。 また、実稼働環境で IBM Blockchain ネットワークをアクティブ化して使用し、セキュリティー、プライバシー、およびパフォーマンスのレベルの高いビジネス・ネットワークを構築し、管理することもできます。 詳細については、IBM Blockchain Platform を参照してください。

Hyperledger Fabric は、 Hyperledger Project のメンバー (初期コード・コントリビューターとして IBM を含む) によって共同で開発された、オープン・ソースのエンタープライズ・ブロックチェーン・フレームワ

ークです。 Hyperledger Project、 または Hyperledger は、業界横断のブロックチェーン・テクノロジーを推進するための Linux Foundation オープン・ソースのグローバルなコラボレーション・イニシアチブです。詳細については、IBM Blockchain、Hyperledger Projects、および Hyperledger Fabric を参照してください。

IBM MQ Advanced または IBM MQ Advanced for z/OS VUE、および IBM Blockchain を既に使用している場合は、IBM MQ Bridge to blockchain を使用して、単純な照会や更新を送信し、 ブロックチェーン・ネットワークからの応答を受信できます。 そのようにして、オンプレミスの IBM ソフトウェアをクラウドのブロックチェーン・サービスに統合できます。

Figure 1 でのブリッジ・オペレーティング・プロセスの概要については、以下を参照してください。 ユーザー・アプリケーションは、JSON 形式のメッセージを、IBM MQ Advanced キュー・マネージャーまたは IBM MQ Advanced for z/OS VUE 上の入力/リクエストキューに入れます。 ブリッジはキュー・マネージャーに接続し、入力/要求キューからメッセージを取得し、JSON が正しくフォーマット設定されていることを確認してから、照会または更新をブロックチェーンに発行します。 ブロックチェーンによって返されるデータは、ブリッジによって解析され、元の IBM MQ 要求メッセージに定義されているように応答キューに入れられます。 ユーザー・アプリケーションは、キュー・マネージャーに接続して、応答キューから応答メッセージを取得し、その情報を使用できます。

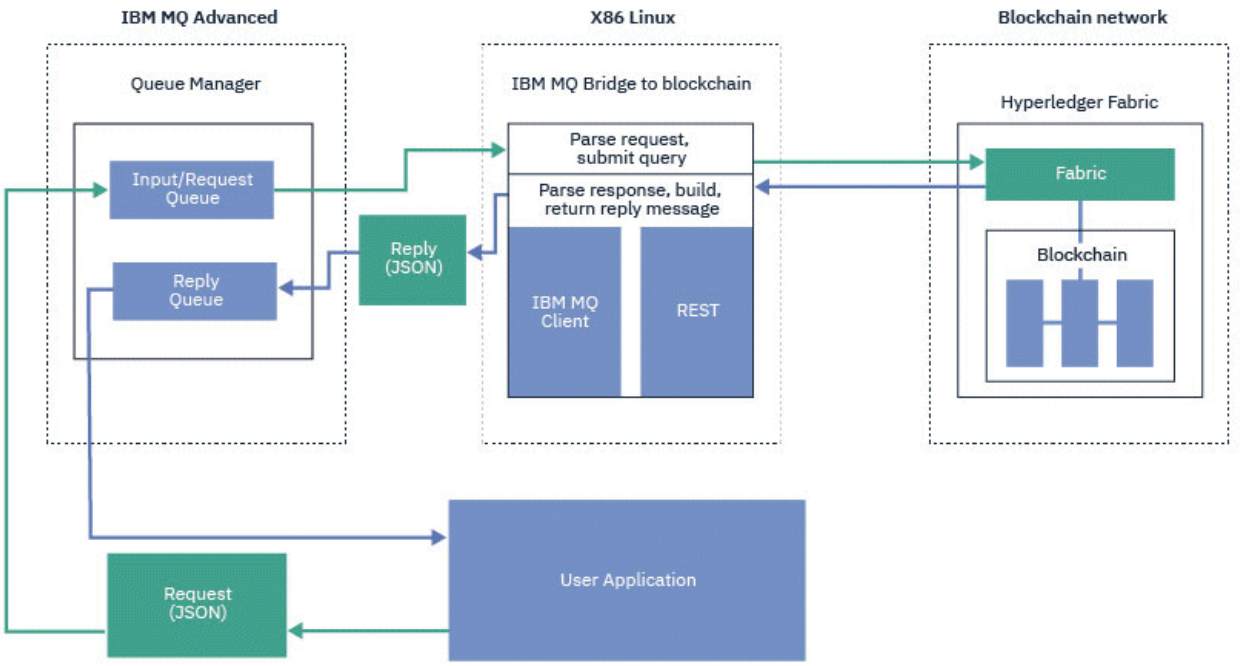


図 *98. IBM MQ Bridge to blockchain*

参加者またはピアとしてブロックチェーン・ネットワークに接続するように IBM MQ Bridge to blockchain を構成できます。 ブリッジの実行中に、メッセージング・アプリケーションからブリッジに、リソースの状態を照会または更新して結果を応答としてメッセージング・アプリケーションに返すチェーンコード・ルーチンを実行するように要求します。

手順

1. キュー・マネージャーを作成して開始するか、または IBM MQ Bridge to blockchain で使用する既存のキュー・マネージャーを開始します。

   キュー・マネージャーの作成:

   ```
   crtmqm adv_qmgr_name
   ```

   キュー・マネージャーの開始:

   ```
   strmqm adv_qmgr_name
   ```

2. **DefineQ.mqsc** スクリプトで定義されているブリッジ用のキューを作成します。

   以下に使用される指定されたデフォルト・キューには、サンプル・ブリッジ・キュー定義が用意されています。

   - ユーザー資格情報、例えば、SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE
   - ブリッジへのメッセージ入力、例えば APPL1.BLOCKCHAIN.INPUT.QUEUE
   - ブロック・チェーンからの応答、例えば、APPL1.BLOCKCHAIN.REPLY.QUEUE

   /opt/mqm/mqbc/samp ディレクトリーから、以下のコマンドを発行します:

   ```
   runmqsc adv_qmgr_name < ./DefineQ.mqsc
   ```

   アプリケーションごとに同じ入力キューを使用できますが、アプリケーションごとに 1 つずつ、複数の応答キューを指定することができます。 定義済み応答キューを使用する必要はありません。 応答に動的キューを使用する場合は、そのセキュリティー構成を考慮する必要があります。

タスクの結果

ブリッジが IBM MQ およびブロックチェーン・ネットワークからのメッセージを処理するために必要なキューを作成しました。

次のタスク

IBM MQ Advanced または IBM MQ Advanced for z/OS VUE、キュー・マネージャー情報、およびブロックチェーン・ネットワークの資格情報を使用して、IBM MQ Bridge to blockchain の構成ファイルを作成します。

Deprecated IBM MQ Bridge to blockchain の構成ファイルの作成

キューマネージャーと IBM Blockchain ブロックチェーン・ネットワークのパラメータを入力し、ネットワークに IBM MQ 接続するための設定ファイル IBM MQ Bridge to blockchain を作成します。

始める前に

- ブロックチェーン・ネットワークを作成し、構成しておきます。
- ブロックチェーン・ネットワークからの資格情報ファイルを入手します。
- x86Linux 環境に、IBM MQ Bridge to blockchain をインストールしました。

  詳しくは、Linux への IBM MQ サーバーのインストールと Linux システム用の IBM MQrpm コンポーネントを参照してください。
- IBM MQ Advanced キュー・マネージャーを開始しておきます。

このタスクについて

このタスクでは、IBM MQ Bridge to blockchain 構成ファイルを作成し、IBM Blockchain および IBM MQ ネットワークに正常に接続するために必要な最小限のセットアップについて説明します。

ブリッジを使用して、Hyperledger Fabric 1.4 architecture を基盤にしたブロックチェーン・ネットワークに接続できます。 ブリッジを使用するには、ブロックチェーン・ネットワークからの構成情報が必要です。このタスクの各ステップで、2 つの異なる構成済みブロックチェーン・ネットワークに基づく構成の詳細の例を示します。

- Docker で実行される Hyperledger Fabric ネットワーク。 詳細については、Getting started with Hyperledger Fabric、Writing your first application、および 889 ページの『Hyperledger Fabric ネットワーク資格情報ファイルの例』を参照してください。
- IBM Cloud の Kubernetes クラスター内で稼働する Hyperledger Fabric ネットワーク。 詳細については、Develop in a cloud sandbox on IBM Blockchain Platform を参照してください。

手順

1. ブリッジを実行して構成ファイルを作成します。
   ブロックチェーン・ネットワークの資格情報ファイルおよび IBM MQ Advanced キュー・マネージャーのパラメーターが必要です。

   ```
   runmqbcb -o config_file_name.cfg
   ```

   以下の例に示すように、既存の値は大括弧に囲まれて表示されます。 既存の値を受け入れる場合は、Enter キーを押します。値をクリアする場合は、Space キーを押してから Enter キーを押します。新しい値を追加する場合は、大括弧で囲んで入力してから Enter キーを押します。 値 (ピアなど) のリストをコンマで区切ることも、新しい行にそれぞれの値を入力して区切ることもできます。 ブランク行はリストの終わりを示します。

   注 : 既存の値は編集できません。 それらを保持、置換、またはクリアすることはできます。
2. IBM MQ Advanced キュー・マネージャーへの接続の値を入力します。
   接続に必要な最小値は、キュー・マネージャー名、ブリッジ入力の名前、およびユーザーが定義したアイデンティティー・キューの名前です。 リモート・キュー・マネージャーに接続する場合は、**MQ Channel** と **MQ Conname** (キュー・マネージャーが実行されているホスト・アドレスとポート) も必要です。 ステップ 888 ページの『4』 で IBM MQ に接続するために TLS を使用するには、JNDI または CCDT を使用し、それに応じて **MQ CCDT URL** または **JNDI implementation class** および **JNDI provider URL** を指定する必要があります。
3. 13.ネットワークの Hyperledger Fabric サーバー資格情報を入力します。
   予想される内容の例を次のコードに示します。

   ```
   Fabric Server
   -------------
   Network configuration file          : []connection-tls.json
   Wallet                              : []
   User Name                           : []User1
   Certificate                         : []<path_to_user_certificate>
   Private Key                         : []<path_to_private_key>/private_key.pem
   Organisation                        : []Org1MSP
   ```

4. TLS 接続の証明書ストア値を入力します。
   この領域は、何もない場合はブランクのままにします。

   ```
   Certificate stores for MQ TLS connections
   -----------------------------------------
   Personal keystore                   : []
   Keystore password                   : []
   Trusted store for signer certs      : []
   Trusted store password              : []
   ```

5. ブリッジのログを書き込むログ・ファイルのパスを入力します。

   ```
   Behavior of bridge program
   --------------------------
   Runtime logfile for copy of stdout/stderr : []bridgelog.log
   Number of logfiles                        : [3]
   Maximum size of each logfile (bytes) : [2097152]
   ```

   重要 : 以前は、ピア、順序付けプログラム、および認証局に関する詳細は、 このブリッジのセットアップに保管されていました。 ただし、この情報は ネットワーク構成ファイルに保管されるようになりました。このファイルは、セットアップの Hyperledger Fabric サーバー・セクションにリンクされています。

タスクの結果

IBM MQ Bridge to blockchain が IBM Blockchain ネットワークおよび IBM MQ Advanced キュー・マネージャーに接続するために使用する構成ファイルを作成しました。

次のタスク

892 ページの『IBM MQ Bridge to blockchain の実行』のステップに従います。

Deprecated Hyperledger Fabric ネットワーク資格情報ファイルの例

Docker で動作しているローカルにインスタンス化された Hyperledger Fabric ブロックチェーンネットワークの .yml ファイルの内容で、IBM MQ Bridge to blockchain が使用することができます。

IBM MQ Bridge to blockchain に接続できるものは次のとおりです。

- Linux IBM MQ Advanced、または
- z/OS IBM MQ Advanced for z/OS VUE

のキュー・マネージャーだけです。

Hyperledger Fabric の使用を開始するチュートリアルを学習し、で何が起きているのかを理解し、Hyperledger Fabric のサンプルのいずれかを使用して、ネットワークを起動した後は、/blockchain/fabric-samples/basic-network フォルダー内に以下の構成ファイルをインストールしておく必要があります。

ブロックチェーン・ネットワークに接続する場合は、887 ページの『IBM MQ Bridge to blockchain の構成ファイルの作成』の際にこのファイルの構成詳細を使用する必要があります。

```
{
  "name": "basic-network",
  "version": "1.0.0",
  "client": {
    "organization": "Org1",
    "connection": {
      "timeout": {
        "peer": {
          "endorser": "300"
        },
        "orderer": "300"
      }
    }
  },
  "channels": {
    "mychannel": {
      "orderers": [
        "orderer.example.com"
      ],
      "peers": {
        "peer0.org1.example.com": {
          "endorsingPeer": true,
          "chaincodeQuery": true,
          "ledgerQuery": true,
          "eventSource": true
        },
        "peer0.org2.example.com": {
          "endorsingPeer": true,
          "chaincodeQuery": false,
          "ledgerQuery": true,
          "eventSource": false
        }
      }
    }
  },
  "organizations": {
    "Org1": {
      "mspid": "Org1MSP",
      "peers": [
        "peer0.org1.example.com"
      ],
      "certificateAuthorities": [
        "ca-org1"
      ],
      "adminPrivateKeyPEM": {
        "path": "$<path_to_private_key>/admin_private_key"
      },
      "signedCertPEM": {
        "path": "<path_to_org_signed_cert>/Admin@org1.example.com-cert.pem"
      }
    },
```

```
    "Org2": {
      "mspid": "Org2MSP",
      "peers": [
        "peer0.org2.example.com"
      ],
      "certificateAuthorities": [
        "ca-org2"
      ]
    }
  },
  "orderers": {
    "orderer.example.com": {
      "url": "grpcs://localhost:7050",
      "mspid": "OrdererMSP",
      "grpcOptions": {
        "ssl-target-name-override": "orderer.example.com",
        "hostnameOverride": "orderer.example.com"
      },
      "tlsCACerts": {
        "path": "<path_to_orderer_cert>/ca.crt"
      },
      "adminPrivateKeyPEM": {
        "path": <path_to_orderers_private_key>/<private_key>"
      },
      "signedCertPEM": {
        "path": "<path_to_orderer_signed_cert>/Admin@example.com-cert.pem"
      }
    }
  },
  "peers": {
    "peer0.org1.example.com": {
      "url": "grpcs://localhost:7051",
      "grpcOptions": {
        "ssl-target-name-override": "peer0.org1.example.com",
        "hostnameOverride": "peer0.org1.example.com",
        "request-timeout": 120001
      },
      "tlsCACerts": {
        "path": <path_to_peer_cert>/ca.crt"
      }
    },
    "peer0.org2.example.com": {
      "url": "grpcs://localhost:9051",
      "grpcOptions": {
        "ssl-target-name-override": "peer0.org2.example.com",
        "hostnameOverride": "peer0.org2.example.com",
        "request-timeout": 120001
      },
      "tlsCACerts": {
        "path": "<path_to_peer_cert>/ca.crt"
      }
    }
  },
  "certificateAuthorities": {
    "ca-org1": {
      "url": "https://localhost:7054",
      "grpcOptions": {
        "verify": true
      },
      "tlsCACerts": {
        "path": "<path_to_ca_cert>/ca.org1.example.com-cert.pem"
      },
      "registrar": [
        {
          "enrollId": "admin",
          "enrollSecret": "adminpw"
        }
      ]
    },
    "ca-org2": {
      "url": "https://localhost:8054",
      "grpcOptions": {
        "verify": true
      },
      "tlsCACerts": {
        "path": "<path_to_ca_cert>/ca.org2.example.com-cert.pem"
      },
      "registrar": [
        {
          "enrollId": "admin",
          "enrollSecret": "adminpw"
        }
```

```
      ]
    }
  }
}
```

Deprecated Linux IBM MQ 9.2.0 からの IBM MQ Bridge to blockchain のメッセージ・フォーマット

IBM MQ Bridge to blockchain によって送受信されるメッセージのフォーマット設定に関する情報。

アプリケーションは、IBM MQ Bridge to blockchain が Hyperledger Fabric サーバーを駆動して、 ブロックチェーンに保持されている情報を処理するように要求します。 アプリケーションでは、ブリッジ要求キューに要求メッセージを配置することによって、これを行います。 要求の結果は、ブリッジによって応答メッセージの形式にフォーマット設定されます。 ブリッジでは、応答メッセージの宛先として、要求メッセージの MQMD の **ReplyToQ** および **ReplyToQMgr** フィールドに格納されている情報を使用します。

要求メッセージと応答メッセージは、JSON 形式のテキスト (MQSTR) メッセージであり、4 つのエレメントを含んでいます。

要求メッセージの形式

要求メッセージには以下の属性が含まれます。

operation
文字列 - 大/小文字を区別しない

更新の場合は `submit`、照会の場合は `evaluate`

ネットワーク
ストリング- Hyperledger Fabric では `channel` と呼ばれることもあります。

contract
文字列 - 呼び出すスマート・コントラクトまたはチェーンコード・パッケージ

args
配列 - 通常は文字列の配列ですが、一部のエレメントはネストされた JSON オブジェクトにすることができます。

contract への実引数 (メソッド名を含む)。

以下に例を示します。

```
{
  "operation" : "Evaluate",
  "network"   : "mychannel",
  "contract"  : "marbles0",
  "args"      : [ "readMarble" , "marble1" ]
}
```

注 : ブリッジは、これらのエレメントの存在と、メッセージが有効な JSON であることは確認しますが、内容の検証を実行しません。 ブリッジは、Hyperledger Fabric に依存して要求の処理またはエラーを返す処理を行います。

応答メッセージの形式

応答メッセージでは、その相関 ID がインバウンド・メッセージのメッセージ ID に設定されています。 ユーザー定義のプロパティーが要求メッセージから応答メッセージにコピーされます。 応答のユーザー ID は、発信元のユーザー ID に設定されます。

statusCode は HTTP 状況コードです。 IBM MQ またはブリッジからのエラーの場合は、該当する **statusCode** が使用されます。

statusType は、*SUCCESS* または *FAILURE* のストリングです。

要求が成功すると、応答メッセージ内の「**data**」エレメントに、呼び出された Hyperledger Composer REST API からの応答が入ります。

正常な処理の例:

```
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "data": [
      {
         "$class": "org.example.trading",
         "firstName": "John",
         "lastName": "Doe",
         "tradeId": "Trader1"
      },
      {
         "$class": "org.example.trading",
         "firstName": "Jane",
         "lastName": "Doe",
         "tradeId": "Trader2"
      }
   ]
}
```

エラー応答の中にあるフィールドはすべて同じです。つまり、ブリッジ自体によって生成された応答、Hyperledger Composer REST サーバーの呼び出しによって生成された応答、ブロックチェーンの呼び出しによって生成された応答、チェーンコード呼び出しによって生成された応答のいずれにおいても同じです。以下に例を示します。

- 誤った JSON 入力メッセージ

```
{
   "statusCode": 400,
   "statusType": "FAILURE",
   "message": "[AMQBC021E] Error: Cannot parse input message or there are
    missing fields in the message. Missing fields appear to be: "method"."
}
```

- Hyperledger Composer REST サーバーでの処理に失敗した要求

```
{
   "statusCode": 500,
   "statusType": "FAILURE",
   "message": "Error trying to invoke business network. Error: No valid responses
    from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
    error (status: 500, message: Error: Failed to add object with ID 'Trader1'
    as the object already exists)"
}
```

アプリケーションでは、**statusType** 文字列またはデータ・フィールドの存在を確認することによって、要求が成功したか失敗したかがわかります。 入力メッセージの処理でエラーが発生し、ブリッジがブロックチェーンに送信しない場合、ブリッジから返される値は MQRC 値で、通常、**MQRC_FORMAT_ERROR** です。

Deprecated IBM MQ Bridge to blockchain の実行

IBM MQ Bridge to blockchain を実行して、IBM Blockchain および IBM MQ に接続します。 接続すると、ブリッジで一連の処理を行う準備が整います。つまり、要求メッセージを処理して Hyperledger Composer ブロックチェーン・ネットワークに送信し、その応答を受信して処理できるようになります。

このタスクについて

前のステップで作成した構成ファイルを使用して、IBM MQ Bridge to blockchain を実行します。

手順

1. ブリッジで使用する IBM MQ Advanced キュー・マネージャーを始動します。
2. IBM MQ Bridge to blockchain を開始して、Hyperledger ComposerREST サーバーに接続し、IBM MQ Advanced キュー・マネージャーに接続します。

   ブリッジ・コマンドを実行します。

```
runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
```

ブリッジが接続されると、以下のような出力が返されます。

```
2018-05-17 14:28:16.866 BST IBM MQ Bridge to Blockchain
5724-H72 (C) Copyright IBM Corp. 2017, 2025.

2018-05-17 14:28:19.331 BST Ready to process input messages.
```

3. オプション: ブリッジの実行後に返されたメッセージが、接続が成功しなかったことを示している場合、IBM MQ Advanced キュー・マネージャーおよびブロックチェーン・ネットワークへの接続をトラブルシューティングします。
   a) デバッグ・オプション 1 を指定して、デバッグ・モードでコマンドを発行します。

   ```
   runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
   -d 1
   ```

   ブリッジによって接続のセットアップがステップスルーされ、処理メッセージが簡潔モードで表示されます。
   b) デバッグ・オプション 2 を指定して、デバッグ・モードでコマンドを発行します。

   ```
   runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
   -d 2
   ```

   ブリッジによって接続のセットアップがステップスルーされ、処理メッセージが冗長モードで表示されます。 全出力がログ・ファイルに書き込まれます。

タスクの結果

IBM MQ Bridge to blockchain を開始し、ご使用の Hyperledger ComposerREST サーバーを使用して、キュー・マネージャーおよびブロック・チェーン・ネットワークに接続しました。

次のタスク

- 895 ページの『z/OS での IBM MQ Bridge to blockchain クライアント・サンプルの実行』のステップに従って、照会または更新メッセージをフォーマット設定し、ブロックチェーン・ネットワークに送信します。
- *MQBCB_EXTRA_JAVA_OPTIONS* 変数を使用して、例えば、IBM MQ トレースを有効化する JVM プロパティーを渡します。 詳しくは、IBM MQ Bridge to blockchain のトレースを参照してください。

Deprecated z/OS z/OS 上の IBM MQ 9.2.0 より前の IBM MQ Bridge to blockchain のメッセージ・フォーマット

IBM MQ Bridge to blockchain によって送受信されるメッセージのフォーマット設定に関する情報。

重要 : メッセージの形式の既存の形式は廃止されました。 IBM MQ 9.2.0 以降、Hyperledger Fabric ネットワークがある場合は、891 ページの『IBM MQ 9.2.0 からの IBM MQ Bridge to blockchain のメッセージ・フォーマット』 で説明しているメッセージの形式を使用してください。

アプリケーションは、ブロックチェーンに保持されている情報に対してアクションを実行するために、IBM MQ Bridge to blockchain が Hyperledger Composer 定義の REST API を駆動することを要求します。 アプリケーションでは、ブリッジ要求キューに要求メッセージを配置することによって、これを行います。 REST 要求の結果は、ブリッジによって応答メッセージの形式にフォーマット設定されます。 ブリッジでは、応答メッセージの宛先として、要求メッセージの MQMD の **ReplyToQ** および **ReplyToQMgr** フィールドに格納されている情報を使用します。

要求メッセージと応答メッセージは、JSON 形式のテキスト (MQSTR) メッセージです。

要求メッセージの形式

要求メッセージには以下の 3 つの属性が含まれます。

メソッド (method)
POST、DELETE、GET などの Hyperledger Composer REST API の呼び出しに使用される REST 動詞

path
Hyperledger Composer REST API へのパス。 これは、基本サーバーの URL に追加されます。 パスの先頭を「`api/`」にする必要があります。

body
メソッド固有の内容。 多くの場合、JSON 構造になります。

次の例では、POST メソッドでパス `api/Trader` を使用して、新規 Trader オブジェクトを作成します。 本文では、Traders クラス (ユーザーの Hyperledger Composer モデルで定義されているもの) を指定し、ブロックチェーン・ネットワーク内で新規 Trader オブジェクトを作成するために必要な追加の値も指定します。

```
{ "method": "POST",
  "path": "api/Trader",
  "body": {
    "$class" : "org.example.trading",
    "tradeId" : "Trader2",
    "firstName": "Jane",
    "lastName" : "Doe"
```

応答メッセージの形式

応答メッセージでは、その相関 ID がインバウンド・メッセージのメッセージ ID に設定されています。 ユーザー定義のプロパティーが要求メッセージから応答メッセージにコピーされます。 応答のユーザー ID は、発信元のユーザー ID に設定されます。

statusCode は HTTP 状況コードです。 IBM MQ またはブリッジからのエラーの場合は、該当する **statusCode** が使用されます。

statusType は、*SUCCESS* または *FAILURE* のストリングです。

要求が成功すると、応答メッセージ内の「**data**」エレメントに、呼び出された Hyperledger Composer REST API からの応答が入ります。

正常な処理の例:

```
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "data": [
      {
         "$class": "org.example.trading",
         "firstName": "John",
         "lastName": "Doe",
         "tradeId": "Trader1"
      },
      {
         "$class": "org.example.trading",
         "firstName": "Jane",
         "lastName": "Doe",
         "tradeId": "Trader2"
      }
   ]
}
```

エラー応答の中にあるフィールドはすべて同じです。つまり、ブリッジ自体によって生成された応答、Hyperledger Composer REST サーバーの呼び出しによって生成された応答、ブロックチェーンの呼び出しによって生成された応答、チェーンコード呼び出しによって生成された応答のいずれにおいても同じです。以下に例を示します。

- 誤った JSON 入力メッセージ

```
{
   "statusCode": 400,
   "statusType": "FAILURE",
   "message": "[AMQBC021E] Error: Cannot parse input message or there are
    missing fields in the message. Missing fields appear to be: "method"."
```

```
}
```

- Hyperledger Composer REST サーバーでの処理に失敗した要求

```
{
   "statusCode": 500,
   "statusType": "FAILURE",
   "message": "Error trying to invoke business network. Error: No valid responses
    from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
    error (status: 500, message: Error: Failed to add object with ID 'Trader1'
    as the object already exists)"
}
```

アプリケーションでは、**statusType** 文字列またはデータ・フィールドの存在を確認することによって、要求が成功したか失敗したかがわかります。 入力メッセージの処理でエラーが発生し、ブリッジがブロックチェーンに送信しない場合、ブリッジから返される値は MQRC 値で、通常、**MQRC_FORMAT_ERROR** です。

Deprecated z/OS z/OS での IBM MQ Bridge to blockchain クライアント・サンプルの実行

IBM MQ Bridge to blockchain に付属の JMS クライアント・サンプルを使用して、ブロックチェーン・ブリッジが検査している入力キューにメッセージを書き込み、受信した応答を確認することができます。 このサンプルは、Hyperledger ComposerTrader ネットワークの例と統合する IBM MQ Bridge to blockchain の使用に基づいています。

始める前に

詳しくは、/trade_network を参照してください

IBM MQ Bridge to blockchain が稼働しており、IBM MQ Advanced キュー・マネージャーまたは IBM MQ Advanced for z/OS VUE およびブロック・チェーン・ネットワークに接続されています。

このタスクについて

IBM MQ Bridge to blockchain の samp ディレクトリーにある JMS サンプル・アプリケーション (ComposerBCBSamp.java)を検索します。

例えば:<MQ_INSTALL_ROOT>/mqbc/samp/ComposerBCBSamp.java の場合、<MQ_INSTALL_ROOT>は次のようになります:

- Linux IBM MQ がインストールされているディレクトリー
- z/OS IBM MQ の z/OS UNIX コンポーネントがインストールされている z/OS UNIX System Services ディレクトリー

.

手順

1. クライアント・サンプルの Java ソース・ファイルを編集します。

   サンプル内の指示に従って、ご使用の IBM MQ 環境およびブロックチェーン・ネットワークに合わせて構成します。

   サンプルの以下のコードは、ブリッジに送信する 3 つの JSON 要求メッセージを定義しています。

   a. まず、既存の 'commodity' を削除します。

   b. 次に、新しい「commodity」、「owner」、および関連する値を作成するには、以下のようにします。

c. 前の 2 つの要求メッセージに続いて、最後に「commodity」に関する新規情報を表示します

```
private static JSONObject[] createMessageBodies() {
    JSONObject[] msgs = new JSONObject[3]; // This method creates 3 messages
    JSONObject m, m2;
    String commodityName = "BC";

    // Clean out the commodity in case it's already there. If
    // it's not there, there will be an error returned from Composer.
    m = new JSONObject();
    m.put("method", "DELETE");
    m.put("path", "api/Commodity/" + commodityName);
    msgs[0] = m;

    // To add the item to the table, the
    // operation looks like this:
    //
    // { "method": "POST",
    //   "path": "api/Commodity",
    //   "body" : {
    //     "$class": "org.example.trading.Commodity",
    //     "tradingSymbol" : "BC",
    //     "description" : "BC",
    //     "mainExchange" : "HERE",
    //     "owner" : "Me",
    //     "quantity" : 100
    //   }
    // }
    // You can see this structure in the API Explorer
    m = new JSONObject();
    m.put("method", "POST");
    m.put("path", "api/Commodity");
    m2 = new JSONObject();
    m2.put("$class", " org.example.trading.Commodity");
    m2.put("tradingSymbol", commodityName);
    m2.put("description", "Blockchain Sample Description");
    m2.put("mainExchange", "My Exchange");
    m2.put("owner", "Me");
    m2.put("quantity", 100);
    m.put("body", m2);
    msgs[1] = m;

    // And list all items that have been created
    m = new JSONObject();
    m.put("method", "GET");
    m.put("path", "api/Commodity");
    msgs[2] = m;

    return msgs;
  }
```

2. サンプルをコンパイルします。
   ブリッジ・ディレクトリーに出荷される IBM MQ クライアント・クラスおよび JSON4J.jar ファイルを指します。

```
javac -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar
ComposerBCBClient.java
```

3. コンパイルされたクラスを実行します。

```
java -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar:.
ComposerBCBClient
```

```
Starting Simple MQ Blockchain Bridge Client
Starting the connection.
Sent message:
  {"method":"DELETE"," path ":"api\/Commodity\/BC"}
Response text:
{
   "statusCode": 204,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": ""
}
SUCCESS
```

```
Sent message:
{"body":
{"$class":"org.example.trading.Commodity","owner":"Me","quantity":100,"description":"Blockcha
in Sample Description","mainExchange":"My
Exchange","tradingSymbol":"BC"},"operation":"POST","url":"Commodity"}
Response text:
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": {
      "$class": "org.example.trading.Commodity",
      "description": "Blockchain Sample Description",
      "mainExchange": "My Exchange",
      "owner": "Me",
      "quantity": 100,
      "tradingSymbol": "BC"
   }
}
SUCCESS
Sent message:
  {"method":"GET","path":"api\/Commodity"}
Response text:
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": [
      {
         "$class": "org.example.trading.Commodity",
         "description": "Blockchain Sample Description",
         "mainExchange": "My Exchange",
         "owner": "resource:org.example.trading.Trader#Me",
         "quantity": 100,
         "tradingSymbol": "BC"
      }
   ]
}
SUCCESS
```

message フィールドには、メッセージが正常に処理された場合は「OK」が設定され、要求が失敗した場合は失敗の理由に関する情報が入ります。

応答の待機中にクライアントがタイムアウト・エラーを受け取った場合は、ブリッジが実行されていることを確認します。

Deprecated Linux IBM MQ Bridge to blockchain の追加の構成オプション

IBM MQ 9.2.0 以降、 IBM MQ Bridge to blockchain でのトレースおよびロギングの動作方法が変更されました。

IBM MQ 9.1.0 IBM MQ Bridge to blockchain からの変更点

デフォルトでは、ログ・ファイルの循環が可能になったという点以外に、IBM MQ 9.1.0 のブリッジからの動作変更はありません。 詳しくは、897 ページの『ログの循環』を参照してください。

トレースとデバッグの相互作用

IBM MQ 9.2.0 以降も、デバッグ・フラグは IBM MQ 9.1.0 と同様に機能し続けます。 *-d1* を指定すると、ブリッジのデバッグ情報が得られ、*-d2* を指定すると、前提条件コンポーネントのデバッグ・ロギングがオンになります。 ただし、ブリッジの開始時に IBM MQ トレースを有効すると、*-d2* レベルのレポートが自動的にオンになります。

ログの循環

IBM MQ 9.2.0 以降、ログ・ファイルのデフォルトの動作では、3 つのログ・ファイルを使用できます。それぞれのサイズは 2 MB です。 追加の構成プロパティーを使用して、この値をオーバーライドすることもできます。 ログ・ファイルの既存の構成属性やコマンド・ライン・パラメーターがログの基底名として取り込まれ、索引が追加されます。

構成するログ・ファイルは、以下のようになります。

- ファイル・タイプがない場合は、ファイル名の末尾に索引が追加されます。

  ログ・ファイルを abc に設定すると、abc.0、abc.1 という具合にログが生成されます。
- ファイル・タイプがある場合は、ファイル・タイプの前に索引が挿入されます。

  ログ・ファイルを abc.log に設定すると、abc.0.log、abc.1.log という具合にログが生成されます。

注 :

1. ブリッジは任意のユーザー許可を使用して、実行できるため、特定のディレクトリー(例えば、/var/mqm/qmgrs/<qm>/errors など)をログに強制することはできません。
2. 今後も *stdout* と *stderr* のストリームに同じ情報が書き込まれます。
3. 個々のログ・ファイルを再び開くと、基本的な構成情報が再び出力されます。 その情報はいつでも使用できます。プログラムの開始時にだけ出力されるということではありません。

Deprecated z/OS MQ Adv. VUE ブロックチェーンで使用するための IBM MQ Advanced for z/OS VUE の構成

IBM MQ Bridge to blockchain をセットアップして実行し、 z/OS キュー・マネージャーと IBM Blockchain で IBM MQ を安全に接続します。 IBM MQ Advanced for z/OS VUE キュー・マネージャーに接続されているメッセージ・アプリケーションを使用すると、ブリッジを使用して、ブロックチェーンのリソースの状態に非同期で接続し、この状態を検索および更新できます。

始める前に

注 :

- Deprecated IBM MQ Bridge to blockchain は、2022 年 11 月 22 日以降のすべてのリリースで非推奨になりました (US Announcement letter 222-341 を参照)。 Blockchain コネクティビティは、 または、 で利用可能な の機能によって実現できる。 IBM App Connect IBM Cloud Pak for Integration App Connect
- V 9.3.2 Removed Continuous Delivery の場合、 IBM MQ Bridge to blockchain は IBM MQ 9.3.2 で製品から削除されています。
- IBM MQ Bridge to blockchain は、IBM MQ Advanced for z/OS Value Unit Edition 9.1.0 の Connector Pack の一部として提供されています。 同等以上のコマンド・レベルで実行されている IBM MQ Advanced for z/OS VUE キュー・マネージャーに接続できます。
- IBM MQ Bridge to blockchain は、Hyperledger Composer をベースとしたブロックチェーン・ネットワーク Hyperledger Fabric で使用するためにサポートされています。
- IBM MQ Bridge to blockchain は、z/OS UNIX System Services 環境にインストールする必要があり、IBM の Java runtime environment のバージョン 8 が必要です。

このタスクについて

ブロックチェーンとは、分散型の共有デジタル台帳であり、ネットワーク上のピアの間で合意されたトランザクションを表すブロックがチェーン状に連なったもので構成されます。 チェーン内のそれぞれのブロックは前のブロックにリンクしているため、最終的には最初のトランザクションにリンクしています。

IBM Blockchain は、Hyperledger Fabric と Hyperledger Composer 上に構築されます。 それを使った開発は、Docker でローカルに行うこともできますし、IBM Cloud のコンテナー・クラスターで行うこともできます。 また、実稼働環境で IBM Blockchain ネットワークをアクティブ化して使用し、セキュリティー、プライバシー、およびパフォーマンスのレベルの高いビジネス・ネットワークを構築し、管理することもできます。 詳細については、IBM Blockchain Platform を参照してください。

Hyperledger Fabric および Hyperledger Composer は、初期コード・コントリビューターとして IBM を含め、 Hyperledger Project のメンバーによって共同で開発された、オープン・ソースのエンタープライズ・ブロックチェーン・フレームワークです。 Hyperledger Project、 または Hyperledger は、業界横断のブロ

ックチェーン・テクノロジーを推進するための Linux Foundation オープン・ソースのグローバルなコラボレーション・イニシアチブです。 詳しくは、IBM Blockchain、Hyperledger Projects、Hyperledger Fabric、および Hyperledger Composer を参照してください。

IBM MQ Advanced for z/OS VUE と IBM Blockchain を既に使用している場合は、IBM MQ Bridge to blockchain を使用して Hyperledger Composer REST インターフェースを介して Hyperledger Composer ビジネス・モデルを駆動できるので、ブロックチェーンの状態の更新や照会を行ったり、ブロックチェーン・ネットワークから応答を受信したりできます。 このようにして、オンプレミスの IBM ソフトウェアを、クラウド・ブロックチェーン・サービスやローカルで管理されたオンプレミス・ソリューションと統合することができる。

ブリッジ操作プロセスの概要は、Figure 1 に記載されています。 ユーザー・アプリケーションは、JSON 形式のメッセージを z/OS キュー・マネージャー上の入力/要求キューに入れます。 Hyperledger Composer REST サーバーを使用して、ブリッジは、キュー・マネージャーに接続し、入力/要求キューからメッセージを取得し、JSON が正しくフォーマット設定されていることを確認してから、REST 要求をブロックチェーンに発行します。 ブロックチェーンによって返されるデータは、ブリッジによって解析され、元の IBM MQ 要求メッセージに定義されているように応答キューに入れられます。 ユーザー・アプリケーションは、キュー・マネージャーに接続して、応答キューから応答メッセージを取得し、その情報を使用できます。

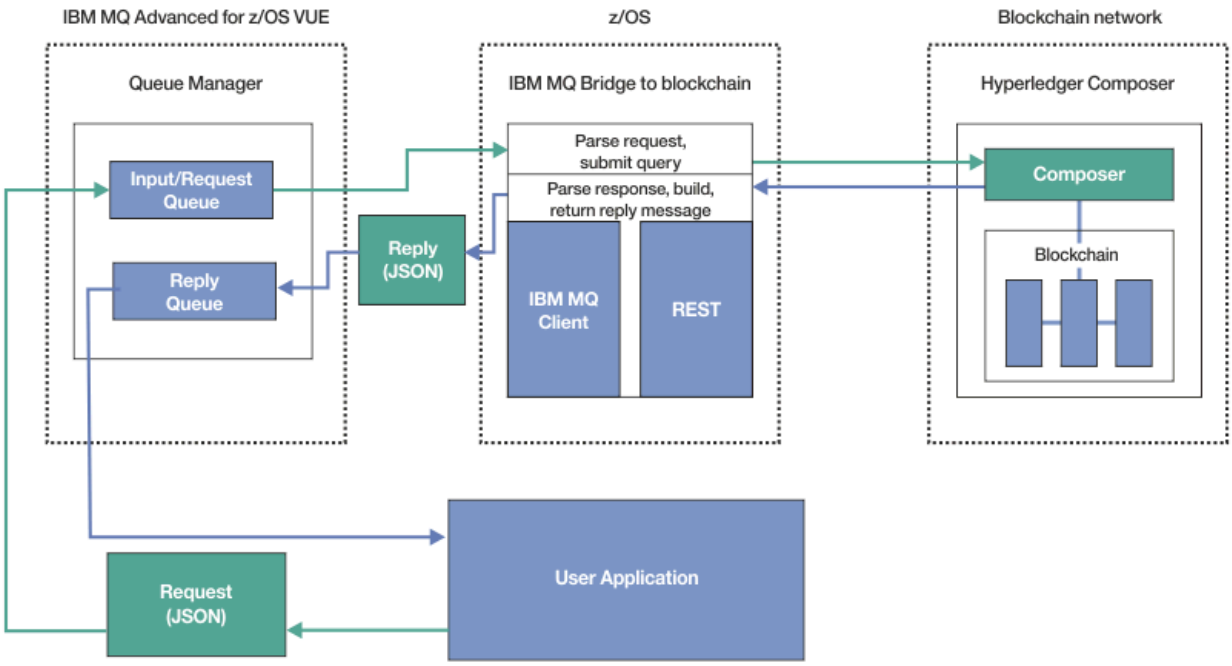


図 *99. IBM MQ Bridge to blockchain*

基礎となる Hyperledger Fabric 層に直接接続するのではなく、 Hyperledger Composer REST サーバーに接続するように IBM MQ Bridge to blockchain を構成する必要があります。 ブリッジの実行中、メッセージング・アプリケーションは、ユーザー定義のビジネス・ネットワーク・モデルに基づいて、ブリッジに Hyperledger Composer REST API を駆動するように要求します。この API はいくつかの基礎的なチェーンコード・ルーチンを駆動して、リソースの状態の照会または更新を行ったり、Hyperledger Composer REST サーバーを使用して結果を応答としてメッセージング・アプリケーションに返したりすることができます。

手順

`thlqual.SCSQPROC(CSQ4BCBQ)`でサンプル JCL をカスタマイズして、実行依頼することにより、ブリッジ用のキューを作成します。

以下に使用される指定されたデフォルト・キューには、サンプル・ブリッジ・キュー定義が用意されています。

- ブリッジの入力メッセージ:`SYSTEM.BLOCKCHAIN.INPUT.QUEUE`、および `APPL1.BLOCKCHAIN.INPUT.QUEUE`

- ブロック・チェーンからの返信:APPL1.BLOCKCHAIN.REPLY.QUEUE

アプリケーションごとに同じ入力キューを使用できますが、アプリケーションごとに 1 つずつ、複数の応答キューを指定することができます。 定義済み応答キューを使用する必要はありません。 応答に動的キューを使用する場合は、そのセキュリティー構成を考慮する必要があります。

タスクの結果

ブリッジが IBM MQ およびブロックチェーン・ネットワークからのメッセージを処理するために必要なキューを作成しました。

次のタスク

キュー・マネージャーの情報とブロックチェーン・ネットワークからの資格情報を使用して、IBM MQ Bridge to blockchain の構成ファイルを作成します。

Deprecated z/OS MQ Adv. VUE z/OS での IBM MQ Bridge to blockchain の構成ファイルの作成

キューマネージャーと IBM Blockchain ブロックチェーン・ネットワークのパラメータを入力し、ネットワークに IBM MQ 接続するための設定ファイル IBM MQ Bridge to blockchain を作成します。

始める前に

注:

- Deprecated IBM MQ Bridge to blockchain は、2022 年 11 月 22 日以降のすべてのリリースで非推奨になりました (US Announcement letter 222-341 を参照)。 Blockchain コネクティビティは、 または、 で利用可能な の機能によって実現できる。 IBM App Connect IBM Cloud Pak for Integration App Connect
- V 9.3.2 Removed Continuous Delivery の場合、 IBM MQ Bridge to blockchain は IBM MQ 9.3.2 で製品から削除されています。
- LTS IBM は、今後のフィックスパックで Long Term Support リリースから機能を削除する予定です。 この変更の影響を受けるアプリケーションがある場合は、 IBM サポートにお問い合わせください。
- Hyperledger Composer ブロックチェーン・ネットワークを作成し、構成しておきます。
- IBM MQ Bridge to blockchain が z/OS 環境にインストールされている。
- IBM MQ Advanced for z/OS VUE キュー・マネージャーを開始しておきます。

このタスクについて

このタスクでは、IBM MQ Bridge to blockchain 構成ファイルを作成し、IBM Blockchain および IBM MQ ネットワークに正常に接続するために必要な最小限のセットアップについて説明します。

ブリッジを使用して、Hyperledger Composer を基盤にしたブロックチェーン・ネットワークに接続できます。 ブリッジを使用するには、ブロックチェーン・ネットワークからの構成情報が必要です。 このタスクの各ステップで、2 つの異なる構成済みブロックチェーン・ネットワークに基づく構成の詳細の例を示します。

- Docker で実行される Hyperledger Composer ネットワーク。 詳しくは、Installing Hyperledger Composer と、Generating a REST API を参照してください。
- IBM Cloud の Kubernetes クラスター内で稼働する Hyperledger Composer ネットワーク。 詳細については、Develop in a cloud sandbox on IBM Blockchain Platform を参照してください。

手順

1. 構成ファイルを作成するには、z/OS UNIX System Services(z/OS UNIX)環境でブリッジを実行します。

Hyperledger Composer のセキュリティー情報と IBM MQ Advanced for z/OS VUE キュー・マネージャーの両方のパラメーターが必要です。

IBM MQ がインストールされている z/OS UNIX 内のロケーションの mqbc/bin ディレクトリーからブリッジ・スクリプトを実行します。

```
./runmqbcb -o config_file_name.cfg
```

以下の例に示すように、既存の値は大括弧に囲まれて表示されます。 既存の値を受け入れる場合は、Enter キーを押します。値をクリアする場合は、Space キーを押してから Enter キーを押します。新しい値を追加する場合は、大括弧で囲んで入力してから Enter キーを押します。 値 (ピアなど) のリストをコンマで区切ることも、新しい行にそれぞれの値を入力して区切ることもできます。 ブランク行はリストの終わりを示します。

注: 既存の値は編集できません。 それらを保持、置換、またはクリアすることはできます。

2. IBM MQ Advanced for z/OS VUE キュー・マネージャーへの接続の値を入力します。

   接続に必要な最小値は、キュー・マネージャー名とユーザーが定義したブリッジ入力キューの名前です。 リモート IBM MQ Advanced for z/OS VUE キュー・マネージャーへの接続の場合、**MQ Channel** と **MQ Conname** (キュー・マネージャーが実行されているホスト・アドレスとポート) も必要です。

   ステップ 901 ページの『5』 で IBM MQ に接続するために TLS を使用するには、JNDI または CCDT を使用し、それに応じて **MQ CCDT URL** または **JNDI implementation class** および **JNDI provider URL** を指定する必要があります。

   注: 値がオーバーラップしている場合、**MQ CCDT** または **JNDI** の値が構成ファイルより優先されます。

```
Connection to Queue Manager
---------------------------
Queue Manager                    : [z/OS_ADV_VUE_qmgr_name]
Bridge Input Queue               : [APPL1.BLOCKCHAIN.INPUT.QUEUE]
MQ Channel                       : []
MQ Conname                       : []
MQ CCDT URL                      : []
JNDI implementation class        : []
JNDI provider URL                : []
MQ Userid                        : []
MQ Password                      : []
```

3. ブロックチェーン・ネットワークに関連付けられている Hyperledger Composer REST サーバーの資格情報を入力します (構成されている場合)。

   以下の例では、 **passport-ldapauth NodeJS** モジュールを使用して、LDAP 資格情報ストアで Hyperledger Composer REST サーバーが構成されています。 **passport-*** モジュールについては、この方法でユーザーとパスワードのスタイルの基本的な資格情報を提供する任意のものを使用できることに注意してください。 詳しくは、Enabling authentication for the REST server を参照してください。

```
User Identification
-------------------------------
Userid                          : []admin
Password                        : []******
API path for Login              : auth/ldap
```

4. Hyperledger Composer REST サーバーのアドレスを入力します。

   この属性にプロトコル (http または https) は不要で、ポート番号は必須であることに注意してください。 HTTP または HTTPS のいずれのプロトコルが使用されるかは、REST サーバーのセキュリティー構成に応じて異なります。 REST サーバーに証明書と秘密鍵のペアが指定されている場合は、HTTPS が使用されます。 HTTPS が使用されます。 その他の場合は、HTTP が使用されます。 証明書と秘密鍵のペアの指定方法については、ステップ 901 ページの『5』 を参照してください。

```
REST Server
---------------------------
Address for Composer REST server    : [composer-rest-server-ip-address:3000]
```

5. TLS 接続の証明書ストア値を入力します。

   ブリッジは、キュー・マネージャーに接続している IBM MQ JMS クライアントとして機能します。つまり、TLS セキュリティーを使用して、その他すべての IBM MQ JMS クライアントと同様に安全に接続す

るようにブリッジを構成できます。 TLS 接続の構成の詳細は、ステップ 901 ページの『2』 で JNDI または CCDT 情報を指定した後にのみ公開されます。

証明書ストアは、Hyperledger Composer で、IBM MQ Advanced for z/OS VUE キュー・マネージャーに対して使用されます。 証明書ストアを指定すると、ブリッジは常に HTTPS を使用して Hyperledger REST サーバーへの接続を試みます。 しかし、以下のオプションを使用して、IBM MQ 接続には TLS を無効にする一方で、Hyperledger Composer には引き続き TLS を使用することができます。

```
Certificate stores for TLS connections
--------------------------------------
Personal keystore                    : []
Keystore password                    : []
Trusted store for signer certs       : []
Trusted store password               : []
Use TLS for MQ connection            : [N]
Timeout for Blockchain operations    : [12]
```

詳しくは、Securing the REST server using HTTPS and TLS を参照してください。

6. オプション: IBM MQ Bridge to blockchain のログ・ファイルの場所を入力します。
   構成ファイルまたはコマンド行でログ・ファイルの名前と場所を指定できます。

```
Behavior of bridge program
--------------------------
Runtime logfile for copy of stdout/stderr : [/var/mqm/errors/runmqbcb.log]
Done.
```

タスクの結果

IBM MQ Bridge to blockchain が IBM Blockchain ネットワークおよび IBM MQ Advanced for z/OS VUE キュー・マネージャーに接続するために使用する構成ファイルを作成しました。

次のタスク

903 ページの『z/OS での IBM MQ Bridge to blockchain の実行』のステップに従います。

Deprecated z/OS MQ Adv. VUE z/OS 上の IBM MQ Bridge to blockchain の IBM MQ セキュリティー構成

IBM MQ Bridge to blockchain を使用した IBM MQ セキュリティーのセットアップに関する考慮事項。

注 :

- Deprecated IBM MQ Bridge to blockchain は、2022 年 11 月 22 日以降のすべてのリリースで非推奨になりました (US Announcement letter 222-341 を参照)。 Blockchain コネクティビティは、 または、 で利用可能な の機能によって実現できる。 IBM App Connect IBM Cloud Pak for Integration App Connect
- V 9.3.2 Removed Continuous Delivery の場合、 IBM MQ Bridge to blockchain は IBM MQ 9.3.2 で製品から削除されています。

以下の例では、必要なキューへのアクセス権限を IBM MQ Bridge to blockchain に付与するために使用できる RACF 定義を示します。 この定義は、ブリッジが MQBCBUSR ユーザー ID で実行されていることを前提としています。

さらに、キュー・マネージャーに接続するためのアクセス権限を、以下の方法で IBM MQ Bridge to blockchain に付与する必要もあります。

- バインディング・モードを使用して直接。バッチ接続のための接続セキュリティー・プロファイルを参照してください。
- CHINIT でクライアント・モードを使用。クライアント MQI 要求を参照してください。

IBM MQ Bridge to blockchain 要求キューに関する権限

以下の RACF コマンドを発行して、デフォルトの SYSTEM.BLOCKCHAIN.INPUT.QUEUE 要求キューからメッセージを受信するためのアクセス権限を MQBCBUSR ユーザー ID に付与します。

```
RDEFINE MQQUEUE SYSTEM.BLOCKCHAIN.INPUT.QUEUE UACC(NONE)
PERMIT SYSTEM.BLOCKCHAIN.INPUT.QUEUE CLASS(MQQUEUE) ID(MQBCBUSR) ACCESS(UPDATE)
```

IBM MQ Bridge to blockchain 応答キューに関する権限

以下の RACF コマンドを発行して、APPL1.BLOCKCHAIN.REPLY.QUEUE にメッセージを送信するためのアクセス権限を MQBCBUSR ユーザー ID に付与します。 このキュー名は、要求メッセージ上のキュー名に対する応答で指定されます。

```
RDEFINE MQQUEUE APPL1.BLOCKCHAIN.REPLY.QUEUE UACC(NONE)
PERMIT APPL1.BLOCKCHAIN.REPLY.QUEUE CLASS(MQQUEUE) ID(MQBCBUSR) ACCESS(UPDATE)
PERMIT CONTEXT.APPL1.BLOCKCHAIN.REPLY.QUEUE CLASS(MQADMIN) ID(MQBCBUSR) ACCESS(UPDATE)
```

関連概念

キュー・セキュリティーのためのプロファイル

関連タスク

895 ページの『z/OS での IBM MQ Bridge to blockchain クライアント・サンプルの実行』

IBM MQ Bridge to blockchain に付属の JMS クライアント・サンプルを使用して、ブロックチェーン・ブリッジが検査している入力キューにメッセージを書き込み、受信した応答を確認することができます。 このサンプルは、Hyperledger ComposerTrader ネットワークの例と統合する IBM MQ Bridge to blockchain の使用に基づいています。

関連資料

API リソース・セキュリティーのアクセス権に関する早見表

Deprecated z/OS MQ Adv. VUE z/OS での IBM MQ Bridge to blockchain の実行

IBM MQ Bridge to blockchain を実行して、IBM Blockchain および IBM MQ に接続します。 接続すると、ブリッジで一連の処理を行う準備が整います。つまり、要求メッセージを処理して Hyperledger Composer ブロックチェーン・ネットワークに送信し、その応答を受信して処理できるようになります。

始める前に

注 :

- Deprecated IBM MQ Bridge to blockchain は、2022 年 11 月 22 日以降のすべてのリリースで非推奨になりました (US Announcement letter 222-341 を参照)。 Blockchain コネクティビティは、 または、 で利用可能な の機能によって実現できる。 IBM App Connect IBM Cloud Pak for Integration App Connect
- V 9.3.2 Removed Continuous Delivery の場合、 IBM MQ Bridge to blockchain は IBM MQ 9.3.2 で製品から削除されています。

このタスクについて

前のステップで作成した構成ファイルを使用して、IBM MQ Bridge to blockchain を実行します。

手順

1. ブリッジで使用する IBM MQ Advanced for z/OS VUE キュー・マネージャーを始動します。
2. IBM MQ Bridge to blockchain を開始して、ブロックチェーン・ネットワークと IBM MQ Advanced for z/OS VUE キュー・マネージャーに接続します。

   以下のいずれかを実行します。

   a) ブリッジがインストールされている z/OS UNIXIBM MQ 場所の `mqbc/bin` ディレクトリから、z/OS UNIX System Services（z/OS UNIX）で直接実行してください。

   ```
   ./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
   ```

   または

b) b. `thlqual.SCSQPROC(CSQ4BCB)`で提供されているサンプル JCL を使用して、z/OS システム上でブリッジを実行します。
ご使用の環境に固有の更新をいくつか JCL に対して行う必要があります。

- ++THLQUAL++ を、IBM MQ ターゲット・ライブラリー・データ・セットの高位修飾子に置き換えます。
- ++LANGLETTER++ を、メッセージを表示する言語の文字に置き換えます。
- ++PATHPREFIX++ を z/OS UNIX コンポーネントのインストール・パスに置き換えます。
- ++CONFIGFILE++ を、z/OS UNIX から `runmqbc -o <file>` コマンドを使用して作成した構成ファイルのパスに置き換えます。
- Java8 以降で動作する 64bitJava の仮想マシン(JVM)の場所に置き換え ++JAVAHOME++てください。

ブリッジが接続されると、以下のような出力が返されます。

```
2018-05-17 14:28:16.866 BST IBM MQ Bridge to Blockchain
5724-H72 (C) Copyright IBM Corp. 2017, 2025.

2018-05-17 14:28:19.331 BST Ready to process input messages.
```

3. オプション: ブリッジの実行後に、接続が成功しなかったことを示すメッセージが返された場合は、IBM MQ Advanced for z/OS VUE キュー・マネージャーとブロックチェーン・ネットワークへの接続のトラブルシューティングを行います。
   a) デバッグ・オプション 1 を指定して、デバッグ・モードでコマンドを発行します。

   ```
   ./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/
   logFile.log -d 1
   ```

   ブリッジによって接続のセットアップがステップスルーされ、処理メッセージが簡潔モードで表示されます。
   b) デバッグ・オプション 2 を指定して、デバッグ・モードでコマンドを発行します。

   ```
   ./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/
   logFile.log -d 2
   ```

   ブリッジによって接続のセットアップがステップスルーされ、処理メッセージが冗長モードで表示されます。 全出力がログ・ファイルに書き込まれます。
   オプションで、「`-d 0`」を「`-d 1`」または「`-d 2`」に変更することにより、JCL 内でデバッグ・モード・オプションを指定することもできます。

タスクの結果

IBM MQ Bridge to blockchain が開始され、キュー・マネージャーとブロックチェーン・ネットワークに接続されます。

次のタスク

- 895 ページの『z/OS での IBM MQ Bridge to blockchain クライアント・サンプルの実行』のステップに従って、照会または更新メッセージをフォーマット設定し、ブロックチェーン・ネットワークに送信します。
- *MQBCB_EXTRA_JAVA_OPTIONS* 変数を使用して、例えば、IBM MQ トレースを有効化する JVM プロパティーを渡します。 詳しくは、IBM MQ Bridge to blockchain のトレースを参照してください。

Deprecated z/OS z/OS 上の IBM MQ 9.2.0 より前の IBM MQ Bridge to blockchain のメッセージ・フォーマット

IBM MQ Bridge to blockchain によって送受信されるメッセージのフォーマット設定に関する情報。

重要 : メッセージの形式の既存の形式は廃止されました。 IBM MQ 9.2.0 以降、Hyperledger Fabric ネットワークがある場合は、891 ページの『IBM MQ 9.2.0 からの IBM MQ Bridge to blockchain のメッセージ・フォーマット』 で説明しているメッセージの形式を使用してください。

アプリケーションは、ブロックチェーンに保持されている情報に対してアクションを実行するために、IBM MQ Bridge to blockchain が Hyperledger Composer 定義の REST API を駆動することを要求します。 アプリケーションでは、ブリッジ要求キューに要求メッセージを配置することによって、これを行います。 REST 要求の結果は、ブリッジによって応答メッセージの形式にフォーマット設定されます。 ブリッジでは、応答メッセージの宛先として、要求メッセージの MQMD の **ReplyToQ** および **ReplyToQMgr** フィールドに格納されている情報を使用します。

要求メッセージと応答メッセージは、JSON 形式のテキスト (MQSTR) メッセージです。

要求メッセージの形式

要求メッセージには以下の 3 つの属性が含まれます。

メソッド (method)
POST、DELETE、GET などの Hyperledger Composer REST API の呼び出しに使用される REST 動詞

path
Hyperledger Composer REST API へのパス。 これは、基本サーバーの URL に追加されます。 パスの先頭を「`api/`」にする必要があります。

body
メソッド固有の内容。 多くの場合、JSON 構造になります。

次の例では、POST メソッドでパス `api/Trader` を使用して、新規 Trader オブジェクトを作成します。 本文では、Traders クラス (ユーザーの Hyperledger Composer モデルで定義されているもの) を指定し、ブロックチェーン・ネットワーク内で新規 Trader オブジェクトを作成するために必要な追加の値も指定します。

```
{ "method": "POST",
  "path": "api/Trader",
  "body": {
    "$class" : "org.example.trading",
    "tradeId" : "Trader2",
    "firstName": "Jane",
    "lastName" : "Doe"
```

応答メッセージの形式

応答メッセージでは、その相関 ID がインバウンド・メッセージのメッセージ ID に設定されています。 ユーザー定義のプロパティーが要求メッセージから応答メッセージにコピーされます。 応答のユーザー ID は、発信元のユーザー ID に設定されます。

statusCode は HTTP 状況コードです。 IBM MQ またはブリッジからのエラーの場合は、該当する **statusCode** が使用されます。

statusType は、*SUCCESS* または *FAILURE* のストリングです。

要求が成功すると、応答メッセージ内の「**data**」エレメントに、呼び出された Hyperledger Composer REST API からの応答が入ります。

正常な処理の例:

```
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "data": [
      {
         "$class": "org.example.trading",
         "firstName": "John",
         "lastName": "Doe",
         "tradeId": "Trader1"
      },
      {
         "$class": "org.example.trading",
         "firstName": "Jane",
```

```
            "lastName": "Doe",
            "tradeId": "Trader2"
        }
    ]
}
```

エラー応答の中にあるフィールドはすべて同じです。つまり、ブリッジ自体によって生成された応答、Hyperledger Composer REST サーバーの呼び出しによって生成された応答、ブロックチェーンの呼び出しによって生成された応答、チェーンコード呼び出しによって生成された応答のいずれにおいても同じです。以下に例を示します。

- 誤った JSON 入力メッセージ

```
{
   "statusCode": 400,
   "statusType": "FAILURE",
   "message": "[AMQBC021E] Error: Cannot parse input message or there are
    missing fields in the message. Missing fields appear to be: "method"."
}
```

- Hyperledger Composer REST サーバーでの処理に失敗した要求

```
{
   "statusCode": 500,
   "statusType": "FAILURE",
   "message": "Error trying to invoke business network. Error: No valid responses
    from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
    error (status: 500, message: Error: Failed to add object with ID 'Trader1'
    as the object already exists)"
}
```

アプリケーションでは、**statusType** 文字列またはデータ・フィールドの存在を確認することによって、要求が成功したか失敗したかがわかります。 入力メッセージの処理でエラーが発生し、ブリッジがブロックチェーンに送信しない場合、ブリッジから返される値は MQRC 値で、通常、**MQRC_FORMAT_ERROR** です。

Deprecated z/OS z/OS での IBM MQ Bridge to blockchain クライアント・サンプルの実行

IBM MQ Bridge to blockchain に付属の JMS クライアント・サンプルを使用して、ブロックチェーン・ブリッジが検査している入力キューにメッセージを書き込み、受信した応答を確認することができます。 このサンプルは、Hyperledger ComposerTrader ネットワークの例と統合する IBM MQ Bridge to blockchain の使用に基づいています。

始める前に

詳しくは、/trade_network を参照してください

IBM MQ Bridge to blockchain が稼働しており、IBM MQ Advanced キュー・マネージャーまたは IBM MQ Advanced for z/OS VUE およびブロック・チェーン・ネットワークに接続されています。

このタスクについて

IBM MQ Bridge to blockchain の samp ディレクトリーにある JMS サンプル・アプリケーション (ComposerBCBSamp.java)を検索します。

例えば:<MQ_INSTALL_ROOT>/mqbc/samp/ComposerBCBSamp.java の場合、<MQ_INSTALL_ROOT>は次のようになります:

- Linux IBM MQ がインストールされているディレクトリー
- z/OS IBM MQ の z/OS UNIX コンポーネントがインストールされている z/OS UNIX System Services ディレクトリー

.

手順

1. クライアント・サンプルの Java ソース・ファイルを編集します。

   サンプル内の指示に従って、ご使用の IBM MQ 環境およびブロックチェーン・ネットワークに合わせて構成します。

   サンプルの以下のコードは、ブリッジに送信する 3 つの JSON 要求メッセージを定義しています。

   a. まず、既存の 'commodity' を削除します。

   b. 次に、新しい「commodity」、「owner」、および関連する値を作成するには、以下のようにします。

   c. 前の 2 つの要求メッセージに続いて、最後に「commodity」に関する新規情報を表示します

```
private static JSONObject[] createMessageBodies() {
    JSONObject[] msgs = new JSONObject[3]; // This method creates 3 messages
    JSONObject m, m2;
    String commodityName = "BC";

    // Clean out the commodity in case it's already there. If
    // it's not there, there will be an error returned from Composer.
    m = new JSONObject();
    m.put("method", "DELETE");
    m.put("path", "api/Commodity/" + commodityName);
    msgs[0] = m;

    // To add the item to the table, the
    // operation looks like this:
    //
    // { "method": "POST",
    //   "path": "api/Commodity",
    //   "body" : {
    //     "$class": "org.example.trading.Commodity",
    //     "tradingSymbol" : "BC",
    //     "description" : "BC",
    //     "mainExchange" : "HERE",
    //     "owner" : "Me",
    //     "quantity" : 100
    //   }
    // }
    // You can see this structure in the API Explorer
    m = new JSONObject();
    m.put("method", "POST");
    m.put("path", "api/Commodity");
    m2 = new JSONObject();
    m2.put("$class", " org.example.trading.Commodity");
    m2.put("tradingSymbol", commodityName);
    m2.put("description", "Blockchain Sample Description");
    m2.put("mainExchange", "My Exchange");
    m2.put("owner", "Me");
    m2.put("quantity", 100);
    m.put("body", m2);
    msgs[1] = m;

    // And list all items that have been created
    m = new JSONObject();
    m.put("method", "GET");
    m.put("path", "api/Commodity");
    msgs[2] = m;

    return msgs;
  }
```

2. サンプルをコンパイルします。

   ブリッジ・ディレクトリーに出荷される IBM MQ クライアント・クラスおよび JSON4J.jar ファイルを指します。

```
javac -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar
ComposerBCBClient.java
```

3. コンパイルされたクラスを実行します。

```
java -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar:.
ComposerBCBClient
```

```
Starting Simple MQ Blockchain Bridge Client
Starting the connection.
Sent message:
  {"method":"DELETE"," path ":"api\/Commodity\/BC"}
Response text:
{
   "statusCode": 204,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": ""
}
SUCCESS
Sent message:
{"body":
{"$class":"org.example.trading.Commodity","owner":"Me","quantity":100,"description":"Blockcha
in Sample Description","mainExchange":"My
Exchange","tradingSymbol":"BC"},"operation":"POST","url":"Commodity"}
Response text:
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": {
      "$class": "org.example.trading.Commodity",
      "description": "Blockchain Sample Description",
      "mainExchange": "My Exchange",
      "owner": "Me",
      "quantity": 100,
      "tradingSymbol": "BC"
   }
}
SUCCESS
Sent message:
  {"method":"GET","path":"api\/Commodity"}
Response text:
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": [
      {
         "$class": "org.example.trading.Commodity",
         "description": "Blockchain Sample Description",
         "mainExchange": "My Exchange",
         "owner": "resource:org.example.trading.Trader#Me",
         "quantity": 100,
         "tradingSymbol": "BC"
      }
   ]
}
SUCCESS
```

message フィールドには、メッセージが正常に処理された場合は「OK」が設定され、要求が失敗した場合は失敗の理由に関する情報が入ります。

応答の待機中にクライアントがタイムアウト・エラーを受け取った場合は、ブリッジが実行されていることを確認します。

z/OS でのキュー・マネージャーの構成

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

始める前に

IBM MQ for z/OS を構成する前に、以下をお読みください。

- IBM MQ for z/OS の概念
- z/OS での IBM MQ 環境の計画

このタスクについて

IBM MQ をインストールした後、それを使用できるようにするためには、多くの作業を行う必要があります。

手順

- IBM MQ for z/OS でキュー・マネージャーを構成する方法については、以下のサブトピックを参照してください。

関連概念

z/OS IBM MQ for z/OS で MQSC コマンドおよび PCF コマンドを発行できるソース

関連タスク

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

セキュリティー

190 ページの『分散キューイングの構成』
このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

15 ページの『クライアントとサーバー間の接続の構成』
IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャネルを定義します。

z/OS IBM MQ for z/OS の管理

計画

関連資料

z/OS IBM MQ for z/OS ユーティリティーの使用

z/OS z/OS でのキュー・マネージャーのカスタマイズ準備

このトピックでは、インストール可能な機能に関する詳細、各国語機能、およびテストに関する詳細を使用してキュー・マネージャーをカスタマイズし、セキュリティーをセットアップすることについて説明しています。

カスタマイズの準備

プログラム・ディレクトリーには、IBM MQ インストール・テープの内容、IBM MQ のプログラムおよびサービス・レベル情報がリストされ、システム修正変更プログラム拡張機能(SMP/E)を使用して、IBM MQ for z/OS をインストールする方法について説明します。プログラム・ディレクトリのダウンロードリンクについては、 IBM MQ for z/OS Program Directory PDF files を参照。

IBM MQ をインストールした後、それを使用できるようにするためには、多くの作業を行う必要があります。 それらの作業の説明については、以下の項を参照してください。

- 914 ページの『IBM MQ for z/OS の設定』
- 980 ページの『z/OS でのキュー・マネージャーのテスト』
- z/OS でのセキュリティーのセットアップ

旧バージョンの IBM MQ for z/OS から移行する場合、大部分のカスタマイズ・タスクについては実行する必要がありません。 実行しなければならないタスクの詳細については、保守とマイグレーションを参照してください。

IBM MQ for z/OS のインストール可能な機能

IBM MQ for z/OS には、以下の機能が含まれます。

基本機能
この機能は必須です。これには、以下の機能を含む、主な機能のすべてが含まれています。

- 管理およびユーティリティー
- IBM MQ アプリケーション・プログラミング・インターフェース(または C++)を使用した、CICS、IMS、およびバッチ・タイプのアプリケーションのサポート
- 分散キューイング機能 (TCP/IP および APPC 通信の両方をサポート)

各国語機能
エラー・メッセージとパネルは、サポートされるすべての各国語で表現されています。 各国語は言語を示す文字で表します。 各国語と言語を示す文字の対応は、次のとおりです。

C
中国語 (簡体字)

E
U.S. 英語(大/小文字混合)

F
フランス語

K
日本語

U
U.S. 英語 (大文字)

米国英語 (大/小文字混合) オプションをインストールする必要があります。 また、1 つ以上の他の言語もインストールできます。 (他の言語のインストール・プロセスには、米国英語 (大/小文字混合) を使用しない場合であっても、米国英語 (大/小文字混合) がインストールされている必要があります。)

IBM MQ for z/OS UNIX System Services Components
このフィーチャーはオプションです。 Jakarta Messaging 3.0 または Java Message Service 2.0 を使用して IBM MQ for z/OS に接続する Java アプリケーションをビルドして実行する場合は、この機能を選択します。

IBM MQ for z/OS UNIX System Services Components のインストールについては、「IBM MQ for z/OS Program Directory PDF files 」を参照してください。

IBM MQ for z/OS UNIX System Services Web Components
このフィーチャーはオプションです。

IBM MQ Console または REST API を使用する場合は、このフィーチャーを選択します。

このフィーチャーをインストールするには、IBM MQ for z/OS UNIX System Services Components フィーチャーをインストールする必要があります。

IBM MQ for z/OS Managed File Transfer
このフィーチャーはオプションであり、IBM MQ Advanced for z/OS、IBM MQ for z/OS Value Unit Edition (VUE)、または IBM MQ for z/OSManaged File Transfer のライセンスを持っている場合にのみ、インストールする必要があります。

IBM MQ for z/OS の Managed File Transfer 機能を使用する場合は、このフィーチャーを選択します。

このフィーチャーをインストールするには、IBM MQ for z/OS UNIX System Services Components フィーチャーをインストールする必要があります。

インストール後に存在するライブラリー

IBM MQ には、複数のロード・ライブラリーが添付されています。 911 ページの表 70 は、IBM MQ をインストールした後に存在する可能性があるライブラリーを示しています。

表 70. インストール後に存在する *IBM MQ* ライブラリー

名前	説明
thlqual.SCSQANLC	IBM MQ の中国語 (簡体字) 版のロード・モジュールが入っています。
thlqual.SCSQANLE	米国のロード・モジュールを含む。 英語(大/小文字混合)の IBM MQ バージョン。
thlqual.SCSQANLF	IBM MQ のフランス語版のロード・モジュールが入っています。
thlqual.SCSQANLK	IBM MQ の日本語版のロード・モジュールが入っています。
thlqual.SCSQANLU	米国のロード・モジュールを含む。 IBM MQ の英語(大文字)のバージョン。
thlqual.SCSQASMS	アセンブラー・サンプル・プログラムのソースが入っています。
thlqual.SCSQAUTH	すべての IBM MQ 製品ロード・モジュールのメイン・リポジトリー。これには、デフォルト・パラメーター・モジュール CSQZPARM も含まれます。 このライブラリーは、APF 許可が必要で、PDS-E 形式でなければなりません。
thlqual.SCSQCICS	CICS DFHRPL 連結に入れる必要のある追加のロード・モジュールが入っています。 このライブラリーは、APF 許可が必要で、PDS-E 形式でなければなりません。
thlqual.SCSQCLST	サンプル・プログラムで使用する CLIST が入っています。
thlqual.SCSQCOBC	サンプル・プログラムに必要なサンプル集などの COBOL サンプル集が入っています。
thlqual.SCSQCOBS	COBOL サンプル・プログラムのソースが入っています。
thlqual.SCSQCPPS	C++ サンプル・プログラムのソースが入っています。
thlqual.SCSQC37S	C サンプル・プログラムのソースが入っています。
thlqual.SCSQC370	サンプル・プログラムに必要なヘッダーを含む、C ヘッダーが入っています。
thlqual.SCSQDEFS	共有キューイング用の C++ および Db2 DBRM のサイド定義が入っています。
thlqual.SCSQEXEC	IBM MQ 操作と操作パネルを使用している場合に SYSEXEC または SYSPROC 連結に組み込む REXX 実行可能ファイルが入っています。
thlqual.SCSQFCMD	Managed File Transfer タスクを作成および実行するジョブのテンプレートが入っています。
thlqual.SCSQHPPS	C++ 用のヘッダー・ファイルが入っています。
thlqual.SCSQINST	インストール・ジョブ用の JCL が入っています。
thlqual.SCSQLINK	早期コード・ライブラリー。 システム初期プログラム・ロード (IPL) 時にロードされるロード・モジュールが入っています。 ライブラリーは、APF 許可が必要です。
thlqual.SCSQLOAD	ロード・ライブラリー。 APF 以外のコード、ユーザー出口、ユーティリティー、サンプル、インストール検査プログラム、およびアダプター・スタブのためのロード・モジュールが入っています。 このライブラリーは、APF 許可の必要はなく、リンク・リスト内にある必要もありません。 このライブラリーは、PDS-E 形式でなければなりません。

表 70. インストール後に存在する *IBM MQ* ライブラリー (続き)

名前	説明
thlqual.SCSQMACS	サンプル・マクロ、製品マクロ、およびシステム・パラメーター・マクロを含むアセンブラー・マクロが入っています。
thlqual.SCSQMAPS	サンプル・プログラムに使用する CICS マップ・セットが入っています。
thlqual.SCSQMSGC	IBM MQ 操作パネルと制御パネルに中国語 (簡体字) 機能を使用している場合に ISPMLIB 連結に組み込む ISPF メッセージが入っています。
thlqual.SCSQMSGE	米国を使用している場合は、ISPMLIB 連結に組み込まれる ISPF メッセージを含む IBM MQ 操作および制御パネルの英語(大/小文字混合)の言語機能。
thlqual.SCSQMSGF	IBM MQ 操作パネルと制御パネルにフランス語機能を使用している場合に ISPMLIB 連結に組み込む ISPF メッセージが入っています。
thlqual.SCSQMSGK	IBM MQ 操作パネルと制御パネルに日本語機能を使用している場合に ISPMLIB 連結に組み込む ISPF メッセージが入っています。
thlqual.SCSQMSGU	米国を使用している場合は、ISPMLIB 連結に組み込まれる ISPF メッセージを含む IBM MQ 操作および制御パネルの英語(大文字)の言語機能。
thlqual.SCSQMVR1	分散キューイングのロード・モジュールが入っています。 このライブラリーは、APF 許可が必要で、PDS-E 形式でなければなりません。
thlqual.SCSQPLIC	PL/I 組み込みファイルが入っています。
thlqual.SCSQPLIS	PL/I サンプル・プログラムのソースが入っています。
thlqual.SCSQPNLA	ISPPLIB 連結に組み込む、ダンプ・フォーマッター用の IPCS パネルが入っています。 また、IBM MQ サンプル・プログラム用のパネルも入っています。
thlqual.SCSQPNLC	IBM MQ 操作パネルと制御パネルに中国語 (簡体字) 機能を使用している場合に ISPPLIB 連結に組み込む ISPF パネルが入っています。
thlqual.SCSQPNLE	米国を使用している場合、ISPPLIB 連結に組み込まれる ISPF パネルが入っています。 IBM MQ 操作および制御パネルの英語(大/小文字混合)の言語機能。
thlqual.SCSQPNLF	IBM MQ 操作パネルと制御パネルにフランス語機能を使用している場合に ISPPLIB 連結に組み込む ISPF パネルが入っています。
thlqual.SCSQPNLK	IBM MQ 操作パネルと制御パネルに日本語機能を使用している場合に ISPPLIB 連結に組み込む ISPF パネルが入っています。
thlqual.SCSQPNLU	米国を使用している場合、ISPPLIB 連結に組み込まれる ISPF パネルが入っています。 IBM MQ 操作および制御パネルの英語(大文字)の言語機能。
thlqual.SCSQPROC	サンプル JCL とデフォルトのシステム初期設定データ・セットが入っています。
thlqual.SCSQSNLC	IBM MQ モジュールの中国語 (簡体字) 版の専用機能 (例えば、早期コード) に必要なロード・モジュールが入っています。

表 70. インストール後に存在する IBM MQ ライブラリー (続き)	
名前	**説明**
thlqual.SCSQSNLE	米国のロード・モジュールを含む。 特殊な目的関数(例えば、初期コード)に必要な IBM MQ モジュールの英語(大/小文字混合)のバージョン。
thlqual.SCSQSNLF	IBM MQ モジュールのフランス語版の専用機能 (例えば、早期コード) に必要なロード・モジュールが入っています。
thlqual.SCSQSNLK	IBM MQ モジュールの日本語版の専用機能 (例えば、早期コード) に必要なロード・モジュールが入っています。
thlqual.SCSQSNLU	米国のロード・モジュールを含む。 特殊目的関数(例えば、初期コード)に必要な IBM MQ モジュールの英語(大文字)のバージョン。
thlqual.SCSQTBLC	IBM MQ 操作パネルと制御パネルに中国語 (簡体字) 機能を使用している場合に ISPTLIB 連結に組み込む ISPF 表が入っています。
thlqual.SCSQTBLE	米国を使用している場合は、ISPTLIB 連結に組み込まれる ISPF テーブルが入っています IBM MQ 操作および制御パネルの英語(大/小文字混合)の言語機能。
thlqual.SCSQTBLF	IBM MQ 操作パネルと制御パネルにフランス語機能を使用している場合に ISPTLIB 連結に組み込む ISPF 表が入っています。
thlqual.SCSQTBLK	IBM MQ 操作パネルと制御パネルに日本語機能を使用している場合に ISPTLIB 連結に組み込む ISPF 表が入っています。
thlqual.SCSQTBLU	米国を使用している場合は、ISPTLIB 連結に組み込まれる ISPF テーブルが入っています IBM MQ 操作および制御パネルの英語(大文字)の言語機能。

注 : これらのライブラリーの変更、またはカスタマイズをしないでください。 変更する場合はライブラリーをコピーし、そのコピーで変更を行ってください。

関連概念

IBM MQ for z/OS の概念

1019 ページの『IBM MQ と IMS の使用』
IBM MQ-IMS アダプター、および IBM MQ-IMS ブリッジは、IBM MQ と IMS との対話を可能にする二つのコンポーネントです。

1027 ページの『IBM MQ と CICS の使用』
CICS で IBM MQ を使用するには、IBM MQ CICS アダプターと、オプションで IBM MQ CICS bridge コンポーネントを構成する必要があります。

1029 ページの『IMS における OTMA 出口の使用』
IBM MQ for z/OS で IMS Open Transaction Manager Access 出口を使用する場合は、このトピックを使用してください。

関連タスク

988 ページの『z/OS での他のキュー・マネージャーとの通信のセットアップ』
このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

IBM MQ for z/OS の管理

関連資料

1027 ページの『言語環境プログラムまたは z/OS 呼び出し可能サービスのアップグレードおよび保守の適用』
必要なアクションは、CALLLIBS または LINK のどちらを使用するか、および使用する SMP/E のバージョンによって異なります。

z/OS IBM MQ for z/OS の設定

このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

キュー・マネージャーを構成する最善の方法は、以下のステップをこの順序で実行することです。

1. 基本キュー・マネージャーを構成します。
2. キュー・マネージャーからキュー・マネージャーへの通信、およびリモート・クライアント・アプリケーション通信を実行するチャネル・イニシエーターを構成します。
3. メッセージを暗号化または保護する場合は、Advanced Message Security for z/OS を構成します。
4. IBM MQ を使用してファイルを転送する場合は、Managed File Transfer for z/OS を構成します。
5. 管理用またはメッセージング用の REST API、あるいは IBM MQ Console を使用して Web ブラウザーから IBM MQ を管理する場合は、mqweb サーバーを構成します。

このトピックでは、IBM MQ を正常にインストールした後のセットアップのさまざまな段階を、順を追って示します。 インストール・プロセスは「プログラム・ディレクトリー」で説明されています プログラム・ディレクトリのダウンロードリンクについては、 IBM MQ for z/OS Program Directory PDF files を参照。

IBM MQ では、カスタマイズに役立つようにサンプルが提供されています。 サンプル・データ・セットのメンバーは、CSQ4 の 4 文字で始まる名前になっています。 また、このメンバーは thlqual.SCSQPROC ライブラリーに格納されています。

このトピックに記載したカスタマイズ作業を実行する前に、さまざまな構成オプションについて検討する必要があります。これらのオプションは、IBM MQ for z/OS のパフォーマンスおよびリソース要件に影響します。 例えば、使用するグローバリゼーション・ライブラリーを決定する必要があります。

カスタマイズ・ステップの一部を自動化する場合は、1033 ページの『IBMz/OSMF を使用して自動化します IBM MQ』を参照してください。

構成オプション

これらのオプションについての詳細は、計画中 z/OS を参照してください。

このセクションの各作業の説明で、以下の点を示しています。

- 作業が IBM MQ のセットアップ・プロセスの一部であるか。 つまり、z/OS システムで IBM MQ をカスタマイズするときに、このタスクを 1 回実行します。 (並列シスプレックスでは、シスプレックス内のそれぞれの z/OS システムごとに作業を実行し、それぞれの z/OS システムがまったく等しくセットアップされていることを確認する必要があります。)
- 作業がキュー・マネージャーの追加作業の一部であるか。 つまり、キュー・マネージャーの追加時にキュー・マネージャーごとに作業を 1 回実行するかどうかということです。

さまざまな z/OS システム・パラメーターを変更するコマンドを使用し、推奨どおりに928 ページの『SYS1.PARMLIB メンバーを更新する』 を実行する場合は、どのタスクでも z/OS システムの IPL を実行する必要はありません。

操作を単純化し、問題判別を支援するために、シスプレックス内の z/OS システムをすべて、まったく等しくセットアップし、緊急の際に、どのシステムでも迅速にキュー・マネージャーを作成できるようにします。

メンテナンスを容易にするために、ご使用の IBM MQ ライブラリーを参照する別名を定義することを検討してください。詳しくは、IBM MQ ライブラリーを参照する別名の使用を参照してください。

関連概念

IBM MQ for z/OS の概念

1019 ページの『IBM MQ と IMS の使用』
IBM MQ-IMS アダプター、および IBM MQ-IMS ブリッジは、IBM MQ と IMS との対話を可能にする二つのコンポーネントです。

1027 ページの『IBM MQ と CICS の使用』

CICS で IBM MQ を使用するには、IBM MQ CICS アダプターと、オプションで IBM MQ CICS bridge コンポーネントを構成する必要があります。

1029 ページの『IMS における OTMA 出口の使用』
IBM MQ for z/OS で IMS Open Transaction Manager Access 出口を使用する場合は、このトピックを使用してください。

関連タスク

988 ページの『z/OS での他のキュー・マネージャーとの通信のセットアップ』
このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

IBM MQ for z/OS の管理

関連資料

1027 ページの『言語環境プログラムまたは z/OS 呼び出し可能サービスのアップグレードおよび保守の適用』
必要なアクションは、CALLLIBS または LINK のどちらを使用するか、および使用する SMP/E のバージョンによって異なります。

z/OS IBM MQ 用の z/OS システムの構成

これらのトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

z/OS *z/OS* システム・パラメーターを識別する

作業の一部には z/OS システム・パラメーターの更新が含まれます。 システム IPL が実行されたとき、どのシステム・パラメーターが指定されたかを知っておく必要があります。

- このタスクは、 *IBM MQ* を実行する *z/OS* システムごとに *1* 回実行する必要があります。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。

SYS1.PARMLIB (IEASYSpp) には、SYS1.PARMLIB の他のメンバーを指すパラメーターのリストが入っています (ここで、pp は、システムの IPL を実行するために使用された z/OS システム・パラメーター・リストを表します)。

検索する必要がある項目は、次のとおりです。

915 ページの『IBM MQ ロード・ライブラリーの APF 許可を行う』 の場合:
PROG=xx または APF=aa は許可プログラム機能 (APF) で許可されたライブラリー・リストを指します (メンバー PROGxx または IEFAPFaa)。

917 ページの『z/OS リンク・リストおよび LPA を更新する』 の場合:
LNK=kk はリンク・リストを指します (メンバー LNKLSTkk)。 LPA=mm は LPA リストを指します (メンバー LPALSTmm)。

921 ページの『z/OS プログラム特性表を更新する』 の場合:
SCH=xx はプログラム特性表 (PPT) を指します (メンバー SCHEDxx)。

921 ページの『IBM MQ サブシステムを z/OS に定義する』 の場合:
SSN=ss は定義されたサブシステム・リストを指します (メンバー IEFSSNss)。

z/OS *IBM MQ* ロード・ライブラリーの *APF* 許可を行う

各種ライブラリーの APF 許可を行います。 ロード・モジュールの中にはすでに許可されているものもあります。

- このタスクは、 *IBM MQ* を実行する *z/OS* システムごとに *1* 回実行する必要があります。
- キュー共用グループを使用している場合は、シスプレックス内の各 *z/OS* システムで *IBM MQ* の設定が同一であることを確認する必要があります。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。
- ライブラリー・ルックアサイド *(LLA)* の使用*:*

– *IBM MQ* の使用方法によっては、ライブラリーからモジュールをロードする際に高入出力 *(IO)* になる場合があります。 この *IO* は、オペレーティング・システムの *LLA* 機能を使用することによって軽減できます。
– この高 *IO* は、以下の場合に発生する可能性があります。
  - *MQCONN/MQDISC* 率の高いアプリケーション *(*例えば、*WLM* ストアード・プロシージャー内*)*。
  - チャネル出口のロード。 開始と停止を頻繁に行うチャネルがあり、チャネル出口を使用する場合。

– *SYS1.PARMLIB* 内のメンバー *CSVLLAxx* は *LLA* セットアップを指定します。 ライブラリー名を *LIBRARIES* ステートメントに含めるということは、プログラム・コピーは常に *VLF (*仮想ルックアサイド機能*)* から取られるので、使用率が高い場合に通常は *I/O* を必要としないということを意味します。

  FREEZE ステートメントに含めるということは、関係のある *DD* ステートメント連結ディレクトリーを取得する *(*しばしばプログラム・ロード自体よりも *I/O* が大きくなることがある*)* ための *I/O* が発生しないということを意味します。

  オペレーティング・システム・コマンドを使用します。*" F LLA, REFRESH "* これらのライブラリーのいずれかが変更された後。

IBM MQ ロード・ライブラリーの thlqual.SCSQAUTH と thlqual.SCSQLINK には、APF 許可が必要です。 さらに、各国語機能用のライブラリー (thlqual.SCSQANLx、thlqual.SCSQSNLx) および分散キューイング機能用のライブラリー (thlqual.SCSQMVR1) についても APF 許可が必要です。

ただし、LPA 内のすべてのロード・モジュールの APF 許可は自動的に行われます。 SYS1.PARMLIB メンバーの IEASYSpp に次のステートメントが含まれている場合は、リンク・リストのすべてのメンバーについても同様です。

```
LNKAUTH=LNKLST
```

LNKAUTH を指定しない場合は、LNKAUTH=LNKLST がデフォルト値になります。

LPA またはリンク・リスト (917 ページの『z/OS リンク・リストおよび LPA を更新する』 を参照) に入れるものによっては、ライブラリーを APF リンク・リストに入れる必要はありません。

注 : IBM MQ STEPLIB に入れたすべてのライブラリーを APF 許可する必要があります。 APF 許可されていないライブラリーを STEPLIB に入れると、ライブラリー連結全体の APF 許可が失われます。

APF リストは、SYS1.PARMLIB のメンバーの PROGxx または IEAAPFaa 内にあります。 これらのリストには、APF 許可された z/OS ライブラリーの名前が入っています。 リスト内の項目の順序には意味がありません。 APF リストについては、 APF 許可ライブラリー・リスト を参照してください。

システムのチューニングについて詳しくは、SupportPac MP16 を参照してください。

動的フォーマットで PROGxx メンバーを使用する場合は、変更を有効にするために z/OS コマンド `SETPROG APF,ADD,DSNAME=hlq.SCSQ XXXX,VOLUME= YYYYYY` を発行するだけで済みます。ここで、*XXXX* はライブラリー名によって異なり、 *YYYYY* はボリュームです。 このコマンドを発行しないで、IEAAPFaa メンバーまたは静的形式を使用した場合は、システムで IPL を実行しなければなりません。

APF リストでは、ライブラリーの実際の名前を使用する必要があります。 ライブラリーのデータ・セット別名を使用しようとすると、許可は失敗します。

関連概念

917 ページの『z/OS リンク・リストおよび LPA を更新する』
早期コード・ライブラリーの新しいバージョンで LPA ライブラリーを更新します。 その他のコードは、リンク・リストまたは LPA に入ることができます。

909 ページの『z/OS でのキュー・マネージャーのカスタマイズ準備』
このトピックでは、インストール可能な機能に関する詳細、各国語機能、およびテストに関する詳細を使用してキュー・マネージャーをカスタマイズし、セキュリティーをセットアップすることについて説明しています。

z/OS *z/OS リンク・リストおよび LPA を更新する*

早期コード・ライブラリーの新しいバージョンで LPA ライブラリーを更新します。 その他のコードは、リンク・リストまたは LPA に入ることができます。

- このタスクは、 IBM MQ を実行する z/OS システムごとに 1 回実行する必要があります。
- キュー共有グループを使用している場合は、キュー・マネージャーを IBM MQ 9.3.0 にマイグレーションする前に、QSG 内の各キュー・マネージャーの早期コードを IBM MQ 9.3.0 レベルにリフレッシュする必要があります。

  各 LPAR に最新の初期コードをインストールしてから、マイグレーション前のある時点でキュー・マネージャーを一度に 1 つずつリフレッシュします。 すべてのキュー・マネージャーを同時にマイグレーションする必要はありません。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。 詳しくは、「プログラム・ディレクトリー」を参照してください。プログラム・ディレクトリのダウンロードリンクについては、 IBM MQ for z/OS Program Directory PDF files を参照。

注 : LPA のデータ・セットは、バージョン固有のものです。 システム内で既存の LPA を使用している場合は、使用する LPA を決定するためにシステム管理者に連絡してください。

早期コード

サブシステムとして動作させるためには、IBM MQ ロードモジュールを MVS 追加する IBM MQ 必要があるものもあります。 これらのモジュールは早期コードと呼ばれ、キュー・マネージャーがアクティブでないときにも実行できます。 例えば、IBM MQ コマンド接頭部を使用して、コンソールでオペレーター・コマンドが実行されると、この早期コードが制御を取得し、キュー・マネージャーを開始する必要があるか、または実行中のキュー・マネージャーに要求を受け渡す必要があるかをチェックします。 このコードはリンク・パック域 (LPA) にロードされます。 早期モジュールは 1 セットあり、すべてのキュー・マネージャーに使用され、IBM MQ の最上位にある必要があります。 あるバージョンの IBM MQ の早期コードを、それより前のバージョンの IBM MQ のキュー・マネージャーに使用することはできますが、その逆は不可能です。

早期コードは、以下のロード・モジュールを構成します。

- CSQ3INI と CSQ3EPX はライブラリ thlqual.SCSQLINK にあります
- CSQ3ECMX in library thlqual.SCSQSNL*x*, *x* はあなたの国の言語の文字です：
  - C、中国語
  - E, 米国英語ミックスケース
  - F はフランス語
  - K、日本語の場合
  - U, 米国英語の大文字

IBM MQ は、thlqual.SCSQSNL*x* ライブラリの内容を thlqual.SCSQLINK に移動し、SMP/E に通知するユーザー修正を含んでいます。 このユーザー修正は CSQ8UERL と呼ばれ、どちらか Long Term Support、または Continuous Delivery のためのプログラムディレクトリ *IBM MQ for z/OS* に記述されています。プログラム・ディレクトリのダウンロードリンクについては、 IBM MQ for z/OS Program Directory PDF files を参照。

LPA ライブラリー内の早期コードを更新した場合、その早期コードは、次の z/OSIPL (CLPA オプションを使用) の時点から、IPL 中に SYS1.PARMLIB 内の IEFSSNss メンバーにある定義から追加されたすべてのキュー・マネージャー・サブシステムで使用することができます。

以下のように早期コードを LPA に追加することにより、後で追加されたどの新規キュー・マネージャー・サブシステム (921 ページの『IBM MQ サブシステムを z/OS に定義する』 を参照) に対しても、IPL なしで即時に早期コードを使用可能にできます。

- CSQ8UERL を使用しなかった場合は、以下の z/OS コマンドを実行します。

```
SETPROG LPA,ADD,MODNAME=(CSQ3INI,CSQ3EPX),DSNAME=thlqual.SCSQLINK
SETPROG LPA,ADD,MODNAME=(CSQ3ECMX),DSNAME=thlqual.SCSQSNLx
```

- CSQ8UERL を使用した場合は、次の z/OS コマンドを使用して早期コードを LPA にロードできます。

```
SETPROG LPA,ADD,MASK=*,DSNAME=thlqual.SCSQLINK
```

- Advanced Message Security を使用する場合、以下の z/OS コマンドを発行して LPA に追加モジュールを組み込む必要もあります。

```
SETPROG LPA,ADD,MODNAME=(CSQ0DRTM),DSNAME=thlqual.SCSQLINK
```

保守を適用した場合、または IBM MQ の新しいバージョンまたはリリースでキュー・マネージャーを再始動する予定の場合は、以下のステップを使用して、既存のキュー・マネージャーが早期コードを使用できるようにすることができます。 これらのステップを実行しないキュー・マネージャーは、既に使用しているバージョンの早期コードを引き続き使用します。 LPAR 上のすべてのキュー・マネージャーに対してこれらのステップを実行する必要はありません。ただし、すべてのキュー・マネージャーに特に保守を適用する場合、またはすべてのキュー・マネージャーを IBM MQ の最新バージョンまたはリリースに更新する場合を除きます。

1. z/OS SETPROG コマンドを使用して、早期コードを LPA に追加します (このトピック内の上記を参照)。
2. IBM MQ コマンド STOP QMGR を使用して、キュー・マネージャーを停止します。
3. qmgr.REFRESH.QMGR セキュリティー・プロファイルがセットアップされていることを確認します。 MQSC コマンドとプロファイルとアクセス権のレベルを参照してください。
4. IBM MQ コマンド REFRESH QMGR TYPE(EARLY) を使用して、キュー・マネージャーの早期コードをリフレッシュします。
5. IBM MQ コマンド START QMGR を使用して、キュー・マネージャーを再始動します。

IBM MQ コマンドの STOP QMGR、REFRESH QMGR、および START QMGR については、MQSC コマンドで説明しています。

その他のコード

次のライブラリーの IBM MQ 提供のロード・モジュールはすべて再入可能であり、LPA に置くことができます。

- SCSQAUTH
- SCSQANL*x*、ここで *x* はあなたの国語の言語文字です
- SCSQMVR1

重要 : ただし、ライブラリーを LPA に置く場合は、保守を適用するたびに、変更したモジュールを手動で LPA にコピーする必要があります。 したがって、 IBM MQ ロード・ライブラリーをリンク・リストに入れることをお勧めします。このリストは、保守後に z/OS コマンド MODIFY LLA REFRESH を発行することによって更新できます。

詳しくは、 LNKLST データ・セットの内容の変更 、および 動的 LNKLST 機能の安全かつ適切な使用を参照してください。

特に SCSQAUTH の場合には、これを複数の STEPLIB に含めないようにするために、LPA に置くことをお勧めします。 LPA またはリンクリストには、SCSQANL*x* という 1 つの言語ライブラリだけを配置する必要があります。 リンク・リスト・ライブラリーは、SYS1.PARMLIB の LNKLSTkk メンバーで指定されます。

分散キューイング機能および CICS bridge (キュー・マネージャーそのものではない) からは、言語環境プログラム (LE) ランタイム・ライブラリー SCEERUN へアクセスする必要があります。 これらのいずれかの機能を使用する場合は、SCEERUN をリンク・リストに組み込む必要があります。

V 9.3.2 一部のモジュールは、キュー・マネージャーの始動時に ECSA にロードされます。 ECSA の制約がある環境では、これらのモジュールを代わりに LPA に配置することができる。 詳しくは、 919 ページの『LPA への IBM MQ グローバル・モジュールの配置』 を参照してください。

重要 : LTS この施設を利用するには、 IBM MQ 9.3 、APAR PH52358 を申請する必要がある。

関連概念

921 ページの『z/OS プログラム特性表を更新する』

IBM MQ キュー・マネージャーには、いくつかの追加 PPT 項目が必要です。

z/OS V 9.3.2 *LPA への IBM MQ グローバル・モジュールの配置*

IBM MQ for z/OS キュー・マネージャーは、始動時に、そのロード・モジュール (グローバル・モジュール) の一部を拡張共通サービス域 (ECSA) にロードします。 キュー・マネージャーのシャットダウン時に、ECSA は解放されます。

IBM MQ 9.3 には 19 個のグローバル・モジュールがあり、実行中のキュー・マネージャーごとに約 1.2 MB の ECSA を消費します。

注 : CSQ7GPLM はグローバル・モジュールですが、LPA に追加しないでください。

LPAR ごとに複数のキュー・マネージャーを実行し、ECSA または高い専用制約のために ECSA の使用量を削減する必要がある環境では、グローバル・モジュールを LPA に配置することができます。 IBM MQ のグローバル・モジュールを LPA に配置するプロセスは手動で行う必要があるため、ECSA または高い専用制約に対処する必要がある場合にのみ、この手順を実行してください。

重要 : LTS IBM MQ 9.3 でこの機能を使用するには、APAR PH52358 を適用する必要があります。

キュー・マネージャーが STEPLIB 内でグローバル・モジュールを検出できず、そのモジュールが LPA 内にあることを検出すると、キュー・マネージャーは、モジュールのコピーを ECSA にロードする代わりに、LPA コピーを直接使用します。 あるいは、キュー・マネージャー・コードが通常リンク・リストからロードされる場合は、LPA 内のすべてのグローバル・モジュールが、リンク・リスト内のすべてのグローバル・モジュールよりも優先してロードされます。

z/OS 共通ストレージ・トラッキング機能 (共通ストレージ・トラッキング機能の使用を参照) は、キュー・マネージャーごとに各キュー・マネージャーの MSTR アドレス・スペースの下のストレージを追跡し、グローバル・モジュールによって使用されているスペースの量を検出するために使用できます。

デフォルトでは、グローバル・モジュールは SCSQAUTH ロード・ライブラリーにあります。 キュー・マネージャーの MSTR アドレス・スペースが STEPLIB 連結を介して SCSQAUTH を見つけると、そこからのグローバル・モジュールが LPA 内のいずれかに優先して使用され、ECSA にロードされます。

グローバル・モジュールは以下のとおりです。

CSQ0GPLM, CSQ3AMGP, CSQ3SSGP, CSQ9PREP,
CSQ9SCNB、CSQGGPLM、CSQMCGLM、CSQMGPLM、 CSQRGLM1、
CSQSLD1、CSQVGEPL、CSQVSRX、 CSQWDLD2、 CSQWDLD3、
CSQWVZSA、 CSQWZDG0、CSQWVZPS、CSQWVGTM、CSQZTDDM

重要 :

- IBM MQ のグローバル・モジュールの名前は、異なる IBM MQ バージョン間で一定のままです。 したがって、グローバル・モジュールを LPA にロードする場合は、単一の IBM MQ バージョンからロードする必要があり、同じ IBM MQ バージョンで実行するキュー・マネージャーでのみ使用する必要があります。
- 同じ LPAR 上で複数のバージョンの IBM MQ が実行されている場合、任意の時点で LPA 内にグローバル・モジュールを持つことができるのは、そのうちの 1 つのみです。
- LPA にロードされたグローバル・モジュールを持つ IBM MQ インストール済み環境に保守が適用され、その保守によっていずれかのグローバル・モジュールが更新される場合は、以下の本文で説明されている手順を再度実行する必要があります。

手順

あるバージョンの IBM MQ のグローバル・モジュールを LPA に配置するには、以下のステップを実行します。

1. `thlqual.SCSQAUTH` ロード・ライブラリーとその内容のコピーを作成します (例: `thlqual.LOCAL.SCSQAUTH`)。 このロード・ライブラリーが、外部セキュリティー・マネージャー (ESM) を使用した無許可アクセスから保護されていることを確認してください。
2. `thlqual.LOCAL.SCSQAUTH` ロード・ライブラリーを APF 許可します。 915 ページの『IBM MQ ロード・ライブラリーの APF 許可を行う』を参照してください。

3. thlqual.LOCAL.SCSQAUTH と同じ属性を使用して、新しい thlqual.GLOBAL.SCSQAUTH ロード・ライブラリーを作成します。

   注: このロード・ライブラリーは APF 許可されている必要はありません。 このロード・ライブラリーが ESM を使用した無許可アクセスから保護されていることを確認してください。
4. 19 個のグローバル・モジュールを thlqual.LOCAL.SCSQAUTH から thlqual.GLOBAL.SCSQAUTH にコピーします。
5. thlqual.LOCAL.SCSQAUTH から 19 個のグローバル・モジュールを削除します。
6. 以下のいずれかの方法で、 thlqual.GLOBAL.SCSQAUTH の 19 個のグローバル・モジュールを LPA に配置します。

   a. a. SYS1.PARMLIB の LPALSTxx メンバーへの thlqual.GLOBAL.SCSQAUTH の追加。 その後、CLPA オプションを指定してシステムを IPL し、ライブラリーの内容が PLPA にロードされるようにする必要があります。

   b. b. 以下のコマンドを使用して、モジュールを LPA に動的に追加します。

```
SETPROG
LPA,ADD,MODNAME=(CSQ0GPLM,CSQ3AMGP,CSQ3SSGP,CSQ9PREP,CSQ9SCNB,CSQGGPLM,
CSQMCGLM,CSQMGPLM,CSQRGLM1,CSQSLD1,CSQVGEPL,CSQVSRX,CSQWDLD2,CSQWDLD3,
CSQWVZSA,CSQWZDG0,CSQWVZPS,CSQWVGTM,CSQZTDDM),DSNAME= thlqual.GLOBAL.SCSQAUTH
```

   注: LPALSTxx は、LPA にモジュールを配置するための推奨される長期的な手段です。
7. 以下のコマンドを発行して、モジュールが LPA 内にあることを確認します。

```
D PROG,LPA,MODNAME=CSQMCGLM
```

コマンドの出力は、モジュールが LPA に正常にロードされた場合、モジュールの入り口点とロード・ポイントを示す必要があります。

LPA からグローバル・モジュールを使用する必要があるキュー・マネージャーごとに、通常は以下を配置します。

1. リンク・リスト内の thlqual.SCSQAUTH は、キュー・マネージャーを停止して開始するだけです。 グローバル・モジュールは LPA からロードされ、ローカル・モジュールはリンク・リストからロードされます。
2. MSTR JCL STEPLIB の thlqual.SCSQAUTH で、STEPLIB が thlqual.SCSQAUTH ではなく thlqual.LOCAL.SCSQAUTH を使用するように JCL を変更してください。 キュー・マネージャーを停止してから開始します。グローバル・モジュールは LPA からロードされ、ローカル・モジュールは STEPLIB からロードされます。

CHIN および AMSM JCL は、 IBM MQ アプリケーションと同様に thlqual.SCSQAUTH を引き続き使用できます。

キュー・マネージャーを ECSA へのグローバル・モジュールのロードに戻すには、以下のステップを実行します。

1. キュー・マネージャーの停止
2. LPALSTxx 定義を除去するか、または次のコマンドを使用して、次回の IPL 時にグローバル・モジュールを LPA から除去します。

```
SETPROG LPA,DELETE,MODNAME=(xxx) FORCE=YES
```

3. thlqual.LOCAL.SCSQAUTH がキュー・マネージャーの STEPLIB にある場合は、thlqual.SCSQAUTH に置き換えます。
4. キュー・マネージャーを再始動します。

関連概念

917 ページの『z/OS リンク・リストおよび LPA を更新する』
早期コード・ライブラリーの新しいバージョンで LPA ライブラリーを更新します。 その他のコードは、リンク・リストまたは LPA に入ることができます。

z/OS *z/OS* プログラム特性表を更新する

IBM MQ キュー・マネージャーには、いくつかの追加 PPT 項目が必要です。

- このタスクは、 *IBM MQ* を実行する *z/OS* システムごとに *1* 回実行する必要があります。
- キュー共用グループを使用している場合は、シスプレックス内の各 *z/OS* システムで *IBM MQ* の設定が同一であることを確認する必要があります。
- 旧バージョンから移行する際は、この作業を実行する必要はありません。
- *Advanced Message Security* を必要とする場合には、この作業の *CSQ0DSRV* 部分を実行する必要があります。

必要なすべての PPT 項目を含むサンプルが、thlqual.SCSQPROC(CSQ4SCHD) に用意されています。 必要な項目が PPT に追加されていることを確認してください。これは SYS1.PARMLIB(SCHEDxx) にあります。

z/OS では、CSQYASCP は既にオペレーティング・システムに対して詳細な属性で定義されており、PARMLIB の SCHEDxx メンバーに組み込む必要がなくなりました。

IBM MQ キュー・マネージャー自体によって、スワッピングが制御されます。 しかし、負荷の高い IBM MQ ネットワークで応答時間が重要な場合、CSQXJST PPT エントリを追加して、IBM MQ チャネルイニシエータをスワップ不可にすることが、z/OS システムの他の部分のパフォーマンスに影響を与えるリスクを考慮すると、有利になる場合があります。

Advanced Message Security が必要な場合は、CSQ0DSRV PPT エントリーを追加します。

これらの変更を有効にするには、 z/OS コマンド **SET SCH=xx** を発行します。ここで、 *xx* は PARMLIB の SCHEDxx メンバーの接尾部です。

関連概念

921 ページの『IBM MQ サブシステムを z/OS に定義する』
サブシステム名の表を更新し、コマンド接頭部ストリング用の規則を決めます。

z/OS キュー・マネージャーとチャネル・イニシエーターの構成

これらのトピックを、キュー・マネージャーとチャネル・イニシエーターを構成するためのステップバイステップ・ガイドとしてご利用ください。

z/OS *IBM MQ* サブシステムを *z/OS* に定義する

サブシステム名の表を更新し、コマンド接頭部ストリング用の規則を決めます。

この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。 旧バージョンから移行する際は、この作業を実行する必要はありません。

関連概念

925 ページの『IBM MQ キュー・マネージャー用のプロシージャーを作成する』
各 IBM MQ サブシステムでは、キュー・マネージャーを始動するためのカタログ式プロシージャーが必要です。 独自のプロシージャー・ライブラリーを作成するか、あるいは IBM 提供のプロシージャー・ライブラリーを使用することができます。

z/OS サブシステム名表の更新

IBM MQ サブシステムを定義する場合、項目をサブシステム名表に追加する必要があります。

SYS1.PARMLIB メンバーの IEFSSNss から最初に取り出される z/OS のサブシステム名表には、正式に定義された z/OS サブシステムの定義が入っています。 個々の IBM MQ サブシステムを定義するには、この表に項目を追加する必要があります。それには、SYS1.PARMLIB の IEFSSNss メンバーを変更する方法と、z/OS コマンド SETSSI を使用する方法 (この方法が望ましい) があります。

IBM MQ サブシステムの初期設定は並列処理をサポートしているため、IBM MQ サブシステム定義ステートメントを、z/OS V1.12 以降で使用可能な IEFSSNss 表の BEGINPARALLEL キーワードの上下に追加することができます。

SETSSI コマンドを使用する場合は変更がただちに有効になるので、システムの IPL を実行する必要はありません。 以降の IPL 後も変更が有効になるように、928 ページの『SYS1.PARMLIB メンバーを更新する』の説明に従って SYS1.PARMLIB も更新してください。

IBM MQ サブシステムを動的に定義するための SETSSI コマンドは、次のとおりです。

```
SETSSI ADD,S=ssid,I=CSQ3INI,P='CSQ3EPX,cpf,scope'
```

IEFSSNss 内の対応する情報は、次の 2 つのいずれかの方法で指定できます。

- IEFSSNss 内の IBM MQ サブシステム定義のキーワード・パラメーター形式。 この方法をお勧めします。

```
SUBSYS SUBNAME(ssid) INITRTN(CSQ3INI) INITPARM('CSQ3EPX,cpf,scope')
```

- IBM MQ サブシステム定義の定位置パラメーター形式。

```
ssid,CSQ3INI,'CSQ3EPX,cpf,scope'
```

これら 2 種類の書式を 1 つの IEFSSNss メンバーの中に混在させないでください。 異なる書式が必要な場合は、新しいメンバーの SSN オペランドを IEASYSpp SYS1.PARMLIB メンバーに追加し、書式の種類ごとに別の IEFSSNss メンバーを使用します。 複数の SSN を指定するには、SSN=(aa,bb,...) を使用します。 IEASYSpp に格納されます。

これらの例について以下に説明します。

ssid
サブシステム ID。 長さは最大 4 文字です。 すべての文字は大文字の英数字 (大文字の A ～ Z、0 ～ 9) で、先頭は英字でなければなりません。 キュー・マネージャーはサブシステムと同じ名前になるので、z/OS サブシステム名と IBM MQ オブジェクト名の両方で許可される文字のみを使用できます。

cpf
コマンド接頭部ストリング (CPF については、923 ページの『コマンド接頭部ストリング (CPF) の定義』を参照)。

scope
z/OS シスプレックスで実行している場合に使用されるシステムの有効範囲 (システムの有効範囲については、924 ページの『シスプレックス環境での CPF』を参照)。

922 ページの図 100 は、IEFSSNss ステートメントのいくつかの例を示しています。

```
CSQ1,CSQ3INI,'CSQ3EPX,+mqs1cpf,S'
CSQ2,CSQ3INI,'CSQ3EPX,+mqs2cpf,S'
CSQ3,CSQ3INI,'CSQ3EPX,++,S'
```

図 *100.* サブシステム定義のためのサンプル *IEFSSNss* ステートメント

注 : オブジェクトをサブシステム内に作成したら、サブシステム名を変更したり、あるサブシステムのページ・セットを別のサブシステムの中で使用したりすることができなくなります。 これらの作業を行う場合は、すべてのオブジェクトとメッセージを 1 つのサブシステムからアンロードして、別のシステムに再ロードする必要があります。

922 ページの表 71 は、922 ページの図 100 のステートメントで定義された、サブシステム名とコマンド接頭部ストリング (CPF) の関連を示すいくつかの例を示しています。

表 *71.* サブシステム名と *CPF* の関連

IBM MQ サブシステム名	**CPF**
CSQ1	+mqs1cpf

表 71. サブシステム名と CPF の関連 (続き)	
IBM MQ サブシステム名	**CPF**
CSQ2	+mqs2cpf
CSQ3	++

注 : z/OS コマンド SETSSI の ACTIVATE 機能と DEACTIVATE 機能は、IBM MQ ではサポートされません。

変更の状況を確認するには、SDSF でコマンド `/D SSI,L` を発行します。「ACTIVE」状況で作成された新規サブシステムが表示されます。

z/OS コマンド接頭部ストリング *(CPF)* の定義

IBM MQ の各サブシステム・インスタンスには、そのサブシステムを識別するためのコマンド接頭部ストリングを含めることができます。

CPF には、システム全域に適用される規則を採用し、すべてのサブシステムが対立しないようにします。以下のガイドラインに従ってください。

- CPF を最大 8 文字のストリングとして定義する。
- すでに他のサブシステムで使用されている CPF は、使用しないでください。また、システムに定義されている JES バックスペース文字をストリングの最初の文字として使用しないでください。
- CPF は、924 ページの表 73 に示されている有効な文字セットから選んだ文字を使用して定義してください。
- すでに定義済みのプロセスの省略形である CPF や、コマンド構文と混同する可能性のある CPF は使用しないでください。 例えば、「D」などの CPF は、DISPLAY などの z/OS コマンドと対立します。 これを回避するには、CPF ストリングの最初の文字あるいは単独の文字として (924 ページの表 73 に示されている) 特殊文字のいずれかを使用してください。
- 既存の CPF のサブセットまたはスーパーセットである CPF を定義しないでください。 例については、923 ページの表 72 を参照してください。

表 72. *CPF* のサブセットおよびスーパーセットの規則例		
サブシステム名	**定義された CPF**	**コマンドの宛先**
MQA	!A	MQA
MQB	!B	MQB
MQC1	!C1	MQC1
MQC2	!C2	MQC2
MQB1	!B1	MQB

Commands intended for subsystemMQB1(using CPF!B1)は、サブシステムの CPF が存在するため、サブシステム MQB に経路指定されます!B、部分集合!B1. 例えば、次のコマンドを入力すると、

```
!B1  START  QMGR
```

サブシステム MQB は、次のコマンドを受け取ります。

```
1 START QMGR
```

(この場合、これは処理できません)。

z/OS コマンドの DISPLAY OPDATA を発行すると、存在する接頭部を表示できます。

シスプレックスで実行している場合、z/OS は CPF 登録時にこの種の対立があるかどうかを診断します (CPF の登録については、924 ページの『シスプレックス環境での CPF』 を参照してください)。

924 ページの表 73 は、CPF ストリングを定義する際に使用できる文字を示しています。

表 73. CPF ストリングに有効な文字セット	
文字セット	**内容**
英字	大文字の A ～ Z、小文字の a ～ z
数字	0 ～ 9
国別 (注を参照)	@ $ # (16 進値として表せる文字)
特殊	. □ ( ) * & + - = ¢ < | ! ; % _ ? : >

注 :

システムは、次の国別文字の 16 進表示を認識します。@ (X'7C' として)、$ (X'5B' として) および # (X'7B' として)。 U.S 以外の国の場合: 端末キーボードに表示される U.S 国別文字により、別の 16 進表記が生成され、エラーが発生する可能性があります。 たとえば、一部の国では、$ 文字は X'4A' を生成することがあります。

セミコロン (;) は CPF として有効ですが、ほとんどのシステムではコマンド区切り文字として使用されます。

z/OS シスプレックス環境での *CPF*

このトピックでは、シスプレックス内で CPF を使用する方法について知ることができます。

シスプレックス環境で使用する場合、IBM MQ は CPF を登録します。これにより、シスプレックス内の任意のコンソールからコマンドを入力し、そのコマンドを適切なシステムへ実行のために経路指定できるようになります。 コマンドの応答は、コマンドが出されたコンソールに戻されます。

シスプレックス操作の有効範囲の定義

有効範囲は、IBM MQ がシスプレックス環境で実行されている場合に、IBM MQ サブシステムが実行する CPF 登録の種類を決定するために使用されます。

可能な有効範囲の値は次のとおりです。

M

システムの有効範囲

CPF は、 IBM MQ によってシステム IPL 時に z/OS に登録され、 z/OS システムがアクティブになっている間ずっと登録されたままになります。

IBM MQ コマンドは、ターゲット・サブシステムを実行している z/OS イメージへ接続されたコンソールで入力するか、ROUTE コマンドを使用してコマンドをそのイメージへ送る必要があります。

シスプレックスで実行していない場合は、このオプションを使用してください。

S

シスプレックスの開始有効範囲

CPF は、 IBM MQ サブシステムの始動時に z/OS に登録され、 IBM MQ サブシステムが終了するまでアクティブのままです。

元の START QMGR コマンドをターゲット・システムに送るためには、ROUTE コマンドを使用する必要がありますが、それ以降のすべての IBM MQ コマンドはシスプレックスに接続された任意のコンソールから入力でき、ターゲット・システムに自動的に送られます。

IBM MQ が終了した後、ROUTE コマンドを使用して、それ以降の START コマンドをターゲット IBM MQ サブシステムへ送る必要があります。

X

シスプレックス IPL 有効範囲

CPF は、 IBM MQ によってシステム IPL 時に z/OS に登録され、 z/OS システムがアクティブになっている間ずっと登録されたままになります。

IBM MQ コマンドは、シスプレックスへ接続された任意のコンソールで入力できます。また、このコマンドはターゲット・システムを実行しているイメージへ自動的に送られます。

有効範囲 S の CPF を持っている IBM MQ サブシステムは、シスプレックス内の 1 つ以上の z/OS イメージについて定義できるので、これらのイメージは単一のサブシステム名表を共有できます。 ただし、IBM MQ サブシステムを実行する z/OS イメージに対して、初期 START コマンドが発行 (または経路指定) されていることを確認する必要があります。 このオプションを使用すると、サブシステム名テーブルを変更したり、z/OS システムの IPL を実行したりすることなく、 IBM MQ サブシステムを停止して、シスプレックス内の別の z/OS イメージで再始動することができます。

有効範囲 X の CPF を持っている IBM MQ サブシステムは、シスプレックス内の 1 つの z/OS イメージについてのみ定義できます。 このオプションを使用する場合は、スコープ X の CPF を使用する IBM MQ サブシステムが必要な z/OS イメージごとに、固有のサブシステム名テーブルを定義する必要があります。

z/OS 自動再始動管理プログラム (ARM) を使用していくつかの異なる z/OS イメージでキュー・マネージャーを自動的に再始動する場合は、キュー・マネージャーが再始動される z/OS イメージごとに、キュー・マネージャーをそれぞれ定義する必要があります。 それぞれのキュー・マネージャーは、有効範囲 S の CPF をもつシスプレックス全体に固有な 4 文字のサブシステム名で定義する必要があります。

z/OS *IBM MQ* キュー・マネージャー用のプロシージャーを作成する

各 IBM MQ サブシステムでは、キュー・マネージャーを始動するためのカタログ式プロシージャーが必要です。 独自のプロシージャー・ライブラリーを作成するか、あるいは IBM 提供のプロシージャー・ライブラリーを使用することができます。

- この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。
- 前のバージョンから移行する際に、カタログ式プロシージャーを変更しなければならない場合があります。

サブシステム名表に定義された IBM MQ サブシステムごとに、カタログ式プロシージャーを、キュー・マネージャー開始用のプロシージャー・ライブラリーで作成してください。 IBM 提供のプロシージャー・ライブラリーの名前は SYS1.PROCLIB ですが、インストール先によっては、独自の命名規則を使用している場合があります。

キュー・マネージャー開始タスク・プロシージャーの名前は、サブシステム名に文字 MSTR を連結して作られます。 例えば、サブシステム CSQ1 では、プロシージャー名は CSQ1MSTR になります。 定義したサブシステムごとに、1 つのプロシージャーが必要です。

選択した言語のメッセージを入れたライブラリーを組み込む必要があります。

- thlqual.SCSQSNLE (米国英語、大/小文字混合)
- thlqual.SCSQSNLU (米国英語、大文字)
- thlqual.SCSQSNLK (日本語)
- thlqual.SCSQSNLF (フランス語)
- thlqual.SCSQSNLC (中国語)

この製品資料の多くの例と説明では、CSQ1 という名前のサブシステムを使用していることを前提としています。 インストール検査とテストのために、CSQ1 という名前のサブシステムを最初に作成すると、これらの例を活用しやすくなります。

2 つのサンプル開始タスク・プロシージャーが thlqual.SCSQPROC 内にあります。 メンバー CSQ4MSTR はメッセージの各クラスごとに 1 つのページ・セットを使用し、メンバー CSQ4MSRR はメッセージの主要クラスに複数のページ・セットを使用します。 SYS1.PROCLIB (SYS1.PROCLIB を使用していない場合は、ご使用のプロシージャー・ライブラリー) のメンバー xxxxMSTR (この xxxx は IBM MQ サブシステムの名前) に、これらのプロシージャーの 1 つをコピーしてください。 各 IBM MQ サブシステムごとに、定義したサンプル・プロシージャーを、プロシージャー・ライブラリーのメンバーにコピーします。

メンバーをコピーした後、メンバー内の説明を使用して、これらのメンバーを各サブシステムの要件に合わせて調整できます。 キュー・マネージャーによって使用されるストレージの制限の指定については、ストレージ構成を参照してください。 さらに、JCL で記号パラメーターを使用して、プロシージャーを開始

時に変更できます。 複数の IBM MQ サブシステムがある場合は、将来の保守を単純化するために、プロシージャーの共通部分に JCL 組み込みグループを使用すると有利な場合があります。

キュー共有グループを使用する場合は、STEPLIB 連結に Db2 ランタイム・ターゲット・ライブラリー SDSNLOAD を含め、APF 許可する必要があります。 このライブラリーは、リンク・リストまたは LPA からアクセスできない場合は、STEPLIB 連結でのみ必要です。

注 :

1. JCL で使用するためにブートストラップ・データ・セット (BSDS)、ログ、およびページ・セットの名前をメモしておき、プロセスの後の手順でこれらのセットを定義することができます。
2. サンプル開始タスク・プロシージャー CSQ4MSTR および CSQ4MSRR が更新され、トランスポート・セキュリティー (SSL または TLS プロパティー) を含む QMINI データ・セットを定義するために使用できる CSQMINI DD カードが組み込まれました (ただし、コメント化されたまま)。

   932 ページの『QMINI データ・セット』 を使用して、TLS 1.3 サポートを使用可能または使用不可にしたり、チャネルによって使用される CipherSpecs のカスタム・リストを定義したり、またはその両方を行うことができます。

関連概念

926 ページの『チャネル・イニシエーター用のプロシージャーを作成する』
IBM MQ サブシステムごとに、CSQ4CHIN のコピーを調整します。 他のどの製品を使用しているかに応じて、他のデータ・セットへのアクセスを許可しなければならないことがあります。

z/OS チャネル・イニシエーター用のプロシージャーを作成する

IBM MQ サブシステムごとに、CSQ4CHIN のコピーを調整します。 他のどの製品を使用しているかに応じて、他のデータ・セットへのアクセスを許可しなければならないことがあります。

- この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。
- 前のバージョンから移行する際に、カタログ式プロシージャーを変更しなければならない場合があります。

分散キューイングを使用する IBM MQ サブシステムには、1 つのサブシステムにつき 1 つずつ、チャネル・イニシエーターの開始タスク・プロシージャーを作成する必要があります。

そのためには、次のようにします。

1. サンプルの開始タスク・プロシージャー thlqual.SCSQPROC(CSQ4CHIN) をプロシージャー・ライブラリーにコピーします。 プロシージャーに *xxxx* CHIN という名前を付けます。ここで、 *xxxx* は IBM MQ サブシステムの名前です (例えば、 CSQ1CHIN は、キュー・マネージャー CSQ1 のチャネル・イニシエーター開始タスク・プロシージャーです)。
2. 使用する IBM MQ サブシステムごとに 1 つずつコピーを作成します。
3. サンプル・プロシージャー CSQ4CHIN 内の命令を使用して、要件に合うようにプロシージャーを調整します。 さらに、JCL で記号パラメーターを使用して、プロシージャーを開始時に変更できます。 これについては、IBM MQ for z/OS の管理で開始オプションとともに説明しています。

   分散キューイング・ライブラリー thlqual.SCSQMVR1 を連結します。

   LE ランタイム・ライブラリー SCEERUN へのアクセスが必要です。このライブラリーがリンク・リスト (SYS1.PARMLIB(LNKLSTkk)) にない場合は、STEPLIB DD ステートメントの中で連結してください。

   V 9.3.1 ストレージ構成の情報を使用して MEMLIMIT パラメーターを調整することを検討してください。
4. 外部セキュリティー管理プログラムの下で実行できるようにプロシージャーに許可を与えます。
5. 選択した言語のメッセージを入れたライブラリーを組み込む必要があります。
   - thlqual.SCSQSNLE (米国英語、大/小文字混合)
   - thlqual.SCSQSNLU (米国英語、大文字)
   - thlqual.SCSQSNLK (日本語)
   - thlqual.SCSQSNLF (フランス語)

- thlqual.SCSQSNLC (中国語)

チャネル・イニシエーターは、実行時間の長いアドレス・スペースです。 制限された量の CPU が消費されてから終了しないように、以下のいずれかが必要です。

- z/OS システムにおける開始タスクのデフォルトが、CPU 無制限であること。 これは、TIME=(1440,00) を指定した、JOBCLASS(STC) の JES2 構成ステートメントで実現します。または、
- CSQXJST の EXEC ステートメントに、TIME=1440 または TIME=NOLIMIT パラメーターを明示的に追加する。

チャネル出口を使用する場合は、あとからこのプロシージャーに出口ライブラリー (CSQXLIB) を追加できます。 そのためにはチャネル・イニシエーターを停止して再始動することが必要です。

TLS を使用する場合は、システム TLS ランタイム・ライブラリーへのアクセスが必要です。 このライブラリーは、SIEALNKE と呼ばれます。 ライブラリーは、APF 許可が必要です。

TCP/IP を使用する場合は、チャネル・イニシエーターのアドレス・スペースは、TCP/IP システム・パラメーターが入っている TCPIP.DATA データ・セットにアクセスできるようにする必要があります。 データ・セットをセットアップする方法は、使用している TCP/IP 製品とインターフェースによって異なります。 次の方法があります。

- 環境変数 RESOLVER_CONFIG
- ファイル・システム上の /etc/resolv.conf
- //SYSTCPD DD ステートメント
- //SYSTCPDD DD ステートメント
- *jobname/userid*.TCPIP.DATA
- SYS1.TCPPARMS(TCPDATA)
- *zapname*.TCPIP.DATA

この一部は、開始タスク・プロシージャー JCL に影響を及ぼします。 詳しくは、z/OS 通信サーバー: IP 構成ガイドを参照してください。

関連概念

927 ページの『IBM MQ サブシステムを z/OSWLM サービス・クラスに定義する』
z/OS システムで IBM MQ に適切なパフォーマンス優先順位を与えるには、キュー・マネージャーおよびチャネル・イニシエーターのアドレス・スペースを適切な z/OS ワークロード管理 (WLM) サービス・クラスに割り当てる必要があります。 これを明示的に行わないと、不適切なデフォルトが適用されることがあります。

z/OS *IBM MQ サブシステムを z/OSWLM サービス・クラスに定義する*

z/OS システムで IBM MQ に適切なパフォーマンス優先順位を与えるには、キュー・マネージャーおよびチャネル・イニシエーターのアドレス・スペースを適切な z/OS ワークロード管理 (WLM) サービス・クラスに割り当てる必要があります。 これを明示的に行わないと、不適切なデフォルトが適用されることがあります。

- この作業は、*IBM MQ* キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際は、この作業を実行する必要はありません。

WLM で提供される ISPF ダイアログを使用して、以下のタスクを実行します。

- WLM 結合データ・セットから z/OS WLM ポリシー定義を取り出す。
- このポリシー定義を更新するため、キュー・マネージャーおよびチャネル・イニシエーター開始タスク・プロシージャーの名前を、選択したサービス・クラスに追加する。
- WLM 結合データ・セットに、変更済みのポリシーをインストールする。

この後、次の z/OS コマンドを使用してこのポリシーをアクティブ化します。

```
V WLM,POLICY=policyname,REFRESH
```

パフォーマンス・オプションの設定について詳しくは、 z/OS での IBM MQ 環境の計画 を参照してください。

関連概念

965 ページの『Db2 環境のセットアップ』
キュー共有グループを使用する場合は、いくつかのサンプル・ジョブをカスタマイズして実行することで、必要な Db2 オブジェクトを作成する必要があります。

z/OS *ESM* セキュリティー管理を実施する

キュー・マネージャーとチャネル・イニシエーターのセキュリティー管理を実装します。

- この作業は、*IBM MQ* キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。

外部セキュリティー管理プログラムとして RACF® を使用している場合は、z/OS でのセキュリティーのセットアップに、これらのセキュリティー管理の実施方法についての説明があります。

チャネル・イニシエーターを使用する場合は、次のようにする必要もあります。

- サブシステムが接続セキュリティーを活動状態にしている場合は、接続セキュリティー・プロファイル ssid.CHIN を外部セキュリティー管理プログラムに定義します (詳細は、チャネル・イニシエーターのための接続セキュリティー・プロファイルを参照)。
- Transport Layer Security (TLS) またはソケット・インターフェースを使用する場合は、「*z/OS UNIX System Services* 計画 」資料の説明に従って、チャネル・イニシエーターを実行する権限を持つユーザー ID が z/OS UNIX System Services を使用するように構成されていることを確認してください。
- TLS を使用する場合は、チャネル・イニシエーターの実行に使用するユーザー ID が、ALTER QMGR コマンドの SSLKEYR パラメーターに指定されている鍵リングにアクセスする設定になっていることを確認してください。

キュー・マネージャーを始動する前に、次の方法によって IBM MQ のデータ・セットとシステム・セキュリティーを設定する必要があります。

- キュー・マネージャーの開始タスク・プロシージャーが、外部セキュリティー管理プログラムの下で実行できるよう許可を与える。
- キュー・マネージャーのデータ・セットへのアクセスの許可を与える。
- 必要に応じて、z/OS データ・セットの暗号化を構成します。

  データ・セット暗号化による IBM MQ for z/OS での保存データの機密性 のセクションを参照してください。 for more information.

この方法の詳細は、「z/OS のセキュリティーのインストール作業」を参照してください。

RACF を使用する場合は、RACF STARTED クラスを使用するなら、システムの IPL を実行する必要はありません (開始済みタスク・プロシージャーに関する RACF 権限を参照)。

関連概念

928 ページの『SYS1.PARMLIB メンバーを更新する』
IPL の実行後も変更を有効な状態に保つには、SYS1.PARMLIB のいくつかのメンバーを更新する必要があります。

969 ページの『キュー共有グループの ESM セキュリティー管理の実装』
キュー共有グループ内のすべてのキュー・マネージャーが Db2 とカップリング・ファシリティー・リスト構造にアクセスするためのセキュリティー管理を実装します。

z/OS *SYS1.PARMLIB* メンバーを更新する

IPL の実行後も変更を有効な状態に保つには、SYS1.PARMLIB のいくつかのメンバーを更新する必要があります。

- このタスクは、 *IBM MQ* を実行する *z/OS* システムごとに *1* 回実行する必要があります。
- キュー共用グループを使用している場合は、シスプレックス内の各 *z/OS* システムで *IBM MQ* の設定が同一であることを確認する必要があります。

- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。

SYS1.PARMLIB メンバーを以下のように更新します。

1. メンバー IEFSSNss を更新します。これについては、921 ページの『IBM MQ サブシステムを z/OS に定義する』で説明されています。
2. IPL の実行時に以下のメンバーが使用されるように IEASYSpp を変更します。
   - PROGxx または IEAAPFaa メンバー (915 ページの『IBM MQ ロード・ライブラリーの APF 許可を行う』で使用)
   - LNKLSTkk および LPALSTmm メンバー (917 ページの『z/OS リンク・リストおよび LPA を更新する』で使用)
   - SCHEDxx メンバー (921 ページの『z/OS プログラム特性表を更新する』で使用)
   - IEFSSNss メンバー (921 ページの『IBM MQ サブシステムを z/OS に定義する』で使用)

関連概念

929 ページの『初期設定入力データ・セットをカスタマイズする』
サンプル初期設定入力データ・セットの作業用コピーを作成し、システム要件に合わせて調整します。

z/OS 初期設定入力データ・セットをカスタマイズする

サンプル初期設定入力データ・セットの作業用コピーを作成し、システム要件に合わせて調整します。

- この作業は、*IBM MQ* キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行する必要があります。

IBM MQ キュー・マネージャーはそれぞれ、初期定義を IBM MQ 初期設定入力データ・セット に入っている一連のコマンドから取得します。 これらのデータ・セットは、キュー・マネージャー開始タスク・プロシージャーで定義された DD 名 CSQINP1、 CSQINP2、および CSQINPT によって参照されます。

これらのコマンドに対する応答は、DD 名 CSQOUT1、CSQOUT2、および CSQOUTT で参照される初期設定出力データ・セットに書き込まれます。

元のサンプルを保存するために、各サンプルの作業用コピーを作成します。 次に、これらの作業用コピー内のコマンドを、システムの要件に合わせて調整できます。

複数の IBM MQ サブシステムを使用する場合は、初期設定入力データ・セットの名前の高位修飾子にサブシステム名を組み込むことで、各データ・セットと関連付けられている IBM MQ サブシステムをより簡単に識別できます。

サンプルについて詳しくは、以下のトピックを参照してください。

- 初期設定データ・セットの形式
- CSQINP1 サンプルの使用
- CSQINP2 サンプルの使用
- CSQINPX サンプルの使用
- CSQINPT サンプルの使用

初期設定データ・セットの形式

初期設定入力データ・セットは、区分データ・セット (PDS) メンバーでも順次データ・セットでも構いません。 初期設定入力データ・セットは、一連のデータ・セットを連結できます。 レコード長 80 バイトで定義しますが、次の条件があります。

- 1 桁目から 72 桁目までだけが有効です。 73 桁目から 80 桁目は無視されます。
- 1 桁目にアスタリスク (*) が付いたレコードは注釈として解釈され、無視されます。
- ブランクのレコードは無視されます。
- 各コマンドは、新しいレコードから開始しなければなりません。
- 行末の - は、次レコードの 1 桁目に継続することを意味します。

- 行末の + は、次レコードの最初のブランク以外の桁に継続することを意味します。
- 1 つのコマンドに使用できる最大文字数は、32 762 です。

初期設定出力データ・セットは順次データ・セットです。レコード長は 125、レコード形式は VBA、ブロック・サイズは 629 です。

CSQINP1 サンプルの使用

データ・セット thlqual.SCSQPROC は、バッファー・プールの定義、バッファー・プール関連へのページ・セット、および ALTER SECURITY コマンドを含む 2 つのメンバーを保持します。

メンバー CSQ4INP1 は、メッセージのクラスごとに 1 つのページ・セットを使用します。 これらのメッセージは、次のいくつかのクラスに分かれています。

- システム関連のメッセージ。
- 重要な長期間のメッセージ。
- 短期間のメッセージ。
- 各種メッセージ。

メンバー CSQ4INPR は、メッセージの主要クラスごとに複数のページ・セットを使用し、他のクラスごとに 1 つのページ・セットを使用します。 メッセージの主要クラスは、次のとおりです。

- 重要な長期間のメッセージ。
- 短期間のメッセージ。

適切なサンプルを、キュー・マネージャー開始タスク・プロシージャーの CSQINP1 連結に組み込みます。

注 :

1. IBM MQ は、ゼロから 99 までの範囲で最大 100 個のバッファー・プールをサポートします。 DEFINE BUFFPOOL コマンドは、CSQINP1 初期設定データ・セットのみから発行できます。 サンプル内の定義では、4 つのバッファー・プールを指定しています。
2. キュー・マネージャーで使用される各ページ・セットは、DEFINE PSID コマンドを使用して、CSQINP1 初期設定データ・セットに定義しなければなりません。 ページ・セット定義は、バッファー・プール ID とページ・セットを関連付けます。 バッファー・プールを指定しなかった場合、バッファー・プール・ゼロ (0) がデフォルトとして使用されます。

   ページ・セット・ゼロ (00) は、必ず定義する必要があります。 このページ・セットには、すべてのオブジェクト定義が入ります。 キュー・マネージャーごとに最大で 100 のページ・セットを定義できます。
3. ALTER SECURITY コマンドは、セキュリティー属性の TIMEOUT および INTERVAL を変更するのに使用できます。 CSQ4INP1 では、デフォルトは TIMEOUT が 54、INTERVAL が 12 に定義されています。

バッファー・プールおよびページ・セットの編成については、 ページ・セットおよびバッファー・プールの計画 を参照してください。

キュー・マネージャーの実行中にバッファー・プールおよびページ・セットの定義を動的に変更する場合は、CSQINP1 定義も更新する必要があります。 これらの変更は、バッファー・プール定義に REPLACE 属性が含まれていなければ、IBM MQ のコールド・スタートの場合にのみ保持されます。

CSQINP2 サンプルの使用

この表には、キュー・マネージャー開始タスク・プロシージャーの CSQINP2 連結に組み込むことができる thlqual.SCSQPROC のメンバーと、その機能の説明がリストされています。 命名規則は CSQ4IN* です。ご使用の構成に合わせて CSQ4INY* メンバーを変更する必要があります。 ただし、CSQINS* メンバーは変更しないでください。次のリリースへのマイグレーション時に変更を再適用する必要があるためです。 代わりに、CSQ4INY* メンバーに DEFINE コマンドまたは ALTER コマンドを入れてください。

表 74. メンバーの所属 *thlqual.SCSQPROC*	
メンバー名	**説明**
CSQ4INSG	システム・オブジェクト定義。
CSQ4INSA	チャネル認証のシステム・オブジェクトおよびデフォルト・ルール。
CSQ4INSX	システム・オブジェクト定義。
CSQ4INSS	キュー共有グループを使用する場合は、このメンバーをカスタマイズして組み込む。
CSQ4INSJ	JMS を使用してパブリッシュ/サブスクライブを行う場合は、このメンバーをカスタマイズして組み込む。
CSQ4INSM	Advanced Message Security のためのシステム・オブジェクト定義。
CSQ4INSR	WebSphere Application Server、または IBM MQ のキュー・パブリッシュ/サブスクライブ・デーモンでサポートされるキュー・パブリッシュ/サブスクライブ・インターフェースを使用している場合は、このメンバーをカスタマイズして組み込みます。
CSQ4DISP	オブジェクト定義を表示する CSQINP2 サンプル。
CSQ4INYC	クラスター化定義。
CSQ4INYD	分散キューイング定義。
CSQ4INYG	一般定義。
CSQ4INYR	メッセージの主要クラスに複数のページ・セットを使用するストレージ・クラス定義。
CSQ4INYS	メッセージの各クラスに 1 つのページ・セットを使用するストレージ・クラス定義。

オブジェクトの定義は 1 回のみ必要です。キュー・マネージャーを始動するたびに定義する必要はありません。そのため、これらの定義を毎回 CSQINP2 に含める必要はありません。 それらを毎回含めると、すでに存在しているオブジェクトを定義することになり、次のようなメッセージを受け取ります。

```
CSQM095I +CSQ1 CSQMAQLC QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) ALREADY EXISTS
CSQM090E +CSQ1 CSQMAQLC FAILURE REASON CODE X'00D44003'
CSQ9023E +CSQ1 CSQMAQLC ' DEFINE QLOCAL' ABNORMAL COMPLETION
```

オブジェクトが、このエラーによって損傷することはありません。 CSQINP2 連結の中に SYSTEM 定義データ・セットを残しておきたい場合は、各オブジェクトに対して REPLACE 属性を指定することにより、このエラー・メッセージを回避できます。

CSQINPX サンプルの使用

サンプル thlqual.SCSQPROC(CSQ4INPX) には、チャネル・イニシエーターが開始するたびに実行できる一連のコマンドが含まれています。 これらは一般的に、START LISTENER などのチャネル関連コマンドです。これらのコマンドは、キュー・マネージャーが開始されるときではなく、チャネル・イニシエーターが開始されるときに毎回必要になります。また、入力データ・セット CSQINP1 あるいは CSQINP2 内では許可されません。 このサンプルは使用前にカスタマイズする必要があり、カスタマイズしたあとでチャネル・イニシエーター用の CSQINPX データ・セットに組み込むことができます。

データ・セットに含まれた IBM MQ コマンドは、チャネル・イニシエーターの初期設定の終わりに実行され、出力は CSQOUTX DD ステートメントによって指定されたデータ・セットへ書き込まれます。 出力は、IBM MQ ユーティリティー・プログラム (CSQUTIL) の COMMAND 機能によって作成された出力と同様です。 IBM MQ for z/OS での CSQUTIL ユーティリティーの使用を参照してください。

チャネル・コマンドのみでなく、CSQUTIL から発行できるどの IBM MQ コマンドでも指定できます。 CSQINPX が処理されている間、他のソースからコマンドを入力できます。 すべてのコマンドは、前のコマンドが正常に実行されたかどうかにかかわらず、順次に発行されます。

コマンド応答時間を指定するには、データ・セットの中の最初のコマンドとして疑似コマンドの COMMAND を使用できます。 これは 1 つのオプションのキーワード RESPTIME (*nnn*) を取ります。ここで、*nnn* は、各コマンドに応答する場合の待ち時間 (秒単位) です。 これは 5 ～ 999 の範囲にあります。デフォルト値は 30 です。

IBM MQ が、4 つのコマンドへの応答が長すぎることを検出すると、CSQINPX の処理は停止され、それ以上コマンドは発行されません。 チャネル・イニシエーターは停止されませんが、メッセージ CSQU052E は CSQOUTX データ・セットに書き込まれ、メッセージ CSQU013E がコンソールに送信されます。

IBM MQ が CSQINPX の処理を正常に完了すると、メッセージ CSQU012I がコンソールに送られます。

CSQINPT サンプルの使用

この表には、キュー・マネージャー開始タスク・プロシージャーの CSQINPT 連結に組み込むことができる thlqual.SCSQPROC のメンバーと、その機能の説明がリストされています。

表 *75.* メンバーの所属 *thlqual.SCSQPROC*

メンバー名	説明
CSQ4INST	システム・デフォルト・サブスクリプション定義。
CSQ4INYT	パブリッシュ/サブスクライブ定義。

データ・セットに含まれた IBM MQ コマンドは、パブリッシュ/サブスクライブの初期設定の完了時に実行され、出力は CSQOUTT DD ステートメントによって指定されたデータ・セットに書き込まれます。 出力は、IBM MQ ユーティリティー・プログラム (CSQUTIL) の COMMAND 機能によって作成された出力と同様です。 IBM MQ for z/OS での CSQUTIL ユーティリティーの使用を参照してください。

関連概念

934 ページの『ブートストラップ・データ・セットおよびログ・データ・セットを作成する』
ブートストラップ・データ・セット (BSDS) およびログ・データ・セットを準備するには、提供されたプログラム CSQJU003 を使用します。

z/OS *QMINI* データ・セット
QMINI データ・セットを使用して、キュー・マネージャーの初期化中に読み取られ、処理されるプロパティーを指定することができます。

QMINI データ・セットの特性

QMINI データ・セットは、最大レコード長が 80 バイト (データに 72 バイト、行番号に 8 バイト) の順次データ・セット、または区分データ・セットのメンバーです。

以下の例は、順次 QMINI データ・セットのプロパティーを示しています。 いくつかのプロパティーはご使用の環境に基づいて決まります。

```
Data Set Name . . . . : QM01.QMINI
General Data                          Current Allocation
Management class . .  : STANDARD      Allocated tracks  . : 1
Storage class  . . .  : STANDARD      Allocated extents . : 1
Volume serial . . .   : P5P21E
Device type . . . .   : 3390
Data class . . . . .  : **None**
Organization  . . .   : PS            Current Utilization
Record format . . .   : FB            Used tracks . . . . : 0
Record length . . .   : 80            Used extents  . . . : 0
Block size  . . . .   : 3120
1st extent tracks .   : 1
Secondary tracks  .   : 1             Dates
Data set name type    :               Creation date . . . : 2020/08/11
Data set encryption   : NO            Referenced date . . : ***None***
                                      Expiration date . . : ***None***
SMS Compressible  .   : NO
```

thlqual.SCSQPROC, includes:

- CSQ4QMIN にある QMINI データ・セットのサンプル・コンテンツ。
- 開始タスク・プロシージャー CSQ4MSTR および CSQ4MSRR にある、キュー・マネージャー開始 JCL 内の //CSQMINI DD カードを使用して QMINI データ・セットを指定する方法のサンプル。

注 :

- データ・セットを解析するコードは、各レコードの最初の 72 バイトのみを解析します。
- 行番号は無視されるので、行番号を指定する必要はありません。
- 行がアスタリスク文字 (*) で始まっている場合、その行はコメントとして扱われます。
- QMINI データ・セットのコンテンツは、キュー・マネージャーの開始時に解析されます。 コンテンツが正常に解析されると、メッセージ CSQM578I がキュー・マネージャーのジョブ・ログに発行されます。 解析中にエラーが検出された場合は、CSQM573E などのエラー・メッセージがキュー・マネージャーのジョブ・ログに発行されますが、キュー・マネージャーは始動します。

  エラー・メッセージを確認し、QMINI データ・セットのコンテンツに関する問題をすべて解決してください。

  キュー・マネージャーが QMINI データ・セットを解析できない場合、チャネル・イニシエーターを開始することはできますが、セキュリティー構成の設定が不明であるため、SSL または TLS を使用するように構成されているチャネルを開始することはできません。
- キュー・マネージャーを開始した後にデータ・セットに対する更新を行う場合は、変更を反映させるためにキュー・マネージャーを再始動する必要があります。

TransportSecurity スタンザ

IBM MQ for z/OS 9.2.0 以降、QMINI データ・セットは TransportSecurity スタンザをサポートします。このスタンザは、IBM MQ for Multiplatforms 上の qm.ini ファイル内の SSL スタンザによって提供される機能と同様の機能を提供します。

TransportSecurity スタンザは、以下のプロパティーをサポートします。

AllowTLSV13
キュー・マネージャーが TLS 1.3 CipherSpecs を使用できるかどうか。有効な値は *TRUE/T/YES/Y* または *FALSE/F/NO/N* です。

マイグレーションされたキュー・マネージャーの場合、デフォルトでは TLS 1.3 は使用可能になっていません。 TLS 1.3 を使用可能にするには、TransportSecurity スタンザと **AllowTLSV13**=TRUE を指定して QMINI データ・セットを定義します。

新しく作成されたキュー・マネージャーの場合、TLS 1.3 はデフォルトで有効になっています。

AllowedCipherSpecs
有効になっている CipherSpecs のカスタム・リスト。

このプロパティーについて詳しくは、IBM MQ for z/OS 上での順序付けと有効化が行われた CipherSpec のカスタム・リストの提供を参照してください。

リスト内の重複した CipherSpec 名は無視されます。

V 9.3.0 **OutboundSNI**
TLS 接続の開始時に、サーバー名表示 (SNI) をリモート・システムへのターゲット IBM MQ チャネル名に設定するか、ホスト名に設定するかを指定します。有効な値は、CHANNEL または HOSTNAME です。

宛先チャネルがチャネル・オブジェクト CERTLABL フィールドの証明書ラベルを使用して構成されている場合、CERTLABL をチャネル値に設定する必要があります。 設定が HOSTNAME の接続が CERTLABL 設定のチャネルに対して行われた場合、接続は失敗し、リモート・キュー・マネージャーのエラー・ログに AMQ9673 メッセージが出力されます。

以下の例は、TransportSecurity スタンザを指定する方法を示しています。

```
TransportSecurity:
AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
                   ECDHE_RSA_AES_256_GCM_SHA384
AllowTLSV13=TRUE
```

z/OS ブートストラップ・データ・セットおよびログ・データ・セットを作成する

ブートストラップ・データ・セット (BSDS) およびログ・データ・セットを準備するには、提供されたプログラム CSQJU003 を使用します。

注 :

- この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。
- z/OS データ・セット暗号化を使用して BSDS またはアクティブ・ログ・データ・セットを保護する場合は、このステップでデータ・セットを割り振る前に、このオプションを構成する必要があります。
- 旧バージョンから移行する際は、この作業を実行する必要はありません。
- キュー・マネージャーをマイグレーションし、アクティブ・ログ・データ・セットまたは BSDS の z/OS データ・セット暗号化を追加する場合は、データ・セットを変換する必要があります。
- z/OS データ・セット暗号化の構成、および既存の IBM MQ データ・セットを暗号化するための変換について詳しくは、 データ・セット暗号化による IBM MQ for z/OS での保存データの機密性を参照してください。

単一または重複のロギング環境を作成する CSQJU003 を実行するためのサンプル JCL およびアクセス方式サービス (AMS) 制御ステートメントが、thlqual.SCSQPROC(CSQ4BSDS) に保持されています。 BSDS およびログを作成するため、およびログを事前フォーマットするためには、このジョブをカスタマイズして、実行してください。

重要 : 最新バージョンの CSQ4BSDS を使用するか、JCL を手動で更新して RECORDS(850 60) を使用しなければなりません。

925 ページの『IBM MQ キュー・マネージャー用のプロシージャーを作成する』 で説明している開始タスク・プロシージャー CSQ4MSTR は、次の形式のステートメントの BSDS を参照します。

```
//BSDS1   DD DSN=++HLQ++.BSDS01,DISP=SHR
//BSDS2   DD DSN=++HLQ++.BSDS02,DISP=SHR
```

ログ・データ・セットは、BSDS によって参照されます。

注 :

1. BLKSIZE を、LOGDEF ステップの SYSPRINT DD ステートメントに指定しなければなりません。 この BLKSIZE は 629 でなければなりません。
2. ブートストラップ・データ・セットおよびログ・データ・セットを異なるキュー・マネージャーから識別するのに役立てるために、これらのデータ・セットの高位修飾子にサブシステム名を使用してください。
3. キュー共有グループを使用する場合、SHAREOPTIONS(2 3) を指定して、 ブートストラップ・データ・セットおよびログ・データ・セットを定義する必要があります。

ブートストラップとログ・データ・セット、およびそれらのサイズの計画方法について詳しくは、計画中 z/OS を参照してください。

IBM MQ 8.0 では、8 バイト・ログ RB の機能強化により、ログ相対バイト・アドレスのラガーの説明に従って、キュー・マネージャーの可用性が向上します。 キュー・マネージャーを最初に開始する前にキュー・マネージャーで 8 バイトのログ RBA を有効にするには、ロギング環境の作成後に以下の手順を実行します。

注 : V 9.3.0 IBM MQ 9.3.0 またはそれ以降の 8 バイト・ログ RBA で作成されたキュー・マネージャーについては、既に使用可能になっているため、以下のステップは不要です。

1. **IDCAMS ALTER** を使用して、バージョン 1 形式の BSDS (CSQJU003 プログラムを使用して作成されたもの) を ++HLQ++.V1.BSDS01 のような名前に変更します。

   注 : VSAM クラスターとともに、データおよび索引コンポーネントを必ず名前変更してください。
2. 既に定義されている属性を使用して、新規の BSDS を割り振ります。 これらは、開始時にキュー・マネージャーによって使用されるバージョン 2 形式の BSDS になります。

3. BSDS 変換ユーティリティー (CSQJUCNV) を実行して、バージョン 1 形式の BSDS を新規のバージョン 2 形式の BSDS に変換します。
4. 変換が正常に完了したら、バージョン 1 形式の BSDS を削除します。

注 : キュー・マネージャーがキュー共有グループに含まれる場合、8 バイトのログ RBA を有効にするためには、その前に次のようにしてキュー共有グループ内のすべてのキュー・マネージャーを開始しておく必要があります。

- キュー・マネージャーが IBM MQ 9.0.0 LTS である場合は、**OPMODE**(NEWFUNC,900) または **OPMODE**(*NEWFUNC,800*) を指定して開始しておく必要があります。
- キュー・マネージャーが IBM MQ 9.0.n CD にある場合、 または IBM MQ 9.1.0 LTS 以降では、そのレベルで開始されている必要があります。

関連概念

935 ページの『ページ・セットの定義』
提供されているサンプルの 1 つを使用して、キュー・マネージャーごとにページ・セットを定義します。

z/OS ページ・セットの定義

提供されているサンプルの 1 つを使用して、キュー・マネージャーごとにページ・セットを定義します。

- この作業は、*IBM MQ* キュー・マネージャーごとに繰り返してください。

  z/OS データ・セット暗号化を使用してページ・セットを保護する場合は、 その手順でデータ・セットを割り振る前に、このオプションを構成する必要があります。
- 旧バージョンから移行する際は、この作業を実行する必要はありません。

  キュー・マネージャーを移行 (マイグレーション) し、ページ・セット用の *z/OS* データ・セット暗号化を追加する場合は、そのページ・セットを変換する必要があります。

  「 データ・セット暗号化を使用した IBM MQ for z/OS での保存データの機密性 」のセクションを参照してください。 *z/OS* データ・セット暗号化の構成および既存の *IBM MQ* データ・セットの暗号化への変換について詳しく説明します。

IBM MQ キュー・マネージャーごとに別々のページ・セットを定義します。 thlqual.SCSQPROC(CSQ4PAGE) および thlqual.SCSQPROC(CSQ4PAGR) には、ページ・セットの定義とフォーマットを行うための JCL 制御ステートメントおよび z/OS アクセス方式サービス・プログラム (AMS) 制御ステートメントが含まれています。 メンバー CSQ4PAGE はメッセージの各クラスごとに 1 つのページ・セットを使用し、メンバー CSQ4PAGR はメッセージの主要クラスに複数のページ・セットを使用します。 JCL は、提供されたユーティリティー・プログラム CSQUTIL を実行します。 サンプルを検討し、必要なページ・セットの数と使用するサイズに合わせてそれらをカスタマイズしてください。 ページ・セットおよび適切なサイズの計算方法については、 ページ・セットおよびバッファー・プールの計画 を参照してください。

925 ページの『IBM MQ キュー・マネージャー用のプロシージャーを作成する』で説明している開始タスク・プロシージャー CSQ4MSTR は、次の形式のステートメントでページ・セットを参照します。

```
//CSQP00nn DD DISP=OLD,DSN=xxxxxxxxxx
```

ここで、*nn* は 00 ～ 99 までのページ・セット番号、*xxxxxxxxxx* は、定義するデータ・セットです。

注 :

1. 動的ページ・セット拡張機能を使用する場合は、2 次エクステントを各ページ・セットに定義しておかなければなりません。 thlqual.SCSQPROC(CSQ4PAGE) は、これを行う方法を示しています。
2. キュー・マネージャーごとにページ・セットが識別しやすくなるように、各ページ・セットへ関連付けられるデータ・セットの高位修飾子にサブシステム名を使用してください。
3. ユーティリティー・プログラム CSQUTIL の FORMAT 機能で FORCE オプションを使用する場合は、AMS DEFINE CLUSTER ステートメントに REUSE 属性を追加しなければなりません。

   REUSE について詳しくは、z/OS DEFINE CLUSTER コマンドの オプション・パラメーター のセクションを参照してください。

4. ご使用のページ・セットが 4 GB を超える場合は、ストレージ管理サブシステム (SMS) の EXTENDED ADDRESSABILITY 機能を使用する必要があります。

関連概念

968 ページの『IBM MQ 項目を Db2 表に追加する』
キュー共有グループを使用している場合は、CSQ5PQSG ユーティリティーを実行して、Db2 データ共有グループの IBM MQ 表にキュー共有グループとキュー・マネージャーの項目を追加します。

z/OS システム・パラメーター・モジュールの調整

IBM MQ システム・パラメーター・モジュールは、IBM MQ がその処理で使用するロギング、保存、トレース、および接続の環境を制御します。 デフォルトのモジュールが提供されます。 いくつかのパラメーター (データ・セット名など) は通常、サイト固有であるため、独自のシステム・パラメーター・モジュールを作成する必要があります。

- この作業は、必要に応じて *IBM MQ* キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。 詳しくは、 z/OS での IBM MQ のマイグレーションを参照してください。
- 既存のキュー・マネージャーで *Advanced Message Security for z/OS* を有効にするために必要なのは、938 ページの『CSQ6SYSP の使用』で説明されているように、*SPLCAP* を *YES* に設定するだけです。 このキュー・マネージャーを初めて構成する場合は、この作業全体を実行してください、

システム・パラメーター・モジュールには、次の 4 つのマクロがあります。

マクロ名	目的
CSQ6SYSP	接続およびトレースのパラメーターを指定します。938 ページの『CSQ6SYSP の使用』を参照してください。
CSQ6LOGP	ログの初期設定を制御します。947 ページの『CSQ6LOGP の使用』を参照してください。
CSQ6ARVP	アーカイブの初期設定を制御します。951 ページの『CSQ6ARVP の使用』を参照してください。
CSQ6USGP	使用率レコーディングを制御します。958 ページの『CSQ6USGP の使用』を参照してください。

IBM MQ には、デフォルトのシステム・パラメーター・モジュール CSQZPARM があります。このモジュールは、(PARM パラメーターを指定しないで) START QMGR コマンドを発行すると自動的に呼び出され、IBM MQ のインスタンスを開始します。 CSQZPARM は APF 許可ライブラリー thlqual.SCSQAUTH に入っています。このライブラリーも IBM MQ で提供されています。 これらのパラメーターの値は、IBM MQ の開始時に一連のメッセージとして表示されます。

このコマンドの使用法について詳しくは、START QMGR を参照してください。

独自のシステム・パラメーター・モジュールの作成

必要なシステム・パラメーターが CSQZPARM に入っていない場合、thlqual.SCSQPROC(CSQ4ZPRM) で提供されているサンプル JCL を使用して、独自のシステム・パラメーター・モジュールを作成することができます。

独自のシステム・パラメーター・モジュールを作成するには、次のようにします。

1. JCL サンプルの作業用コピーを作成します。
2. 必要に応じて、コピー内の各マクロに対応するパラメーターを編集します。 マクロ呼び出しからいずれかのパラメーターを取り除くと、実行時にデフォルトの値が自動的に設定されます。
3. ロード・モジュールに付ける名前でプレースホルダー ++NAME++ を置き換えます (CSQZPARM にすることができます)。

4. 使用しているアセンブラーが高水準アセンブラーでない場合は、アセンブラーの要求に従って JCL を変更します。
5. JCL を実行することにより、調整されているバージョンのシステム・パラメーター・マクロをアセンブルおよびリンク・エディットしてロード・モジュールを生成します。 これが、指定した名前を取る新しいシステム・パラメーター・モジュールになります。
6. 生成されたロード・モジュールを APF 許可ユーザー・ライブラリーに書き込みます。
7. ユーザーの READ アクセス権限を APF 許可ユーザー・ライブラリーに追加します。
8. このライブラリーを、IBM MQ キュー・マネージャーの開始タスク・プロシージャー STEPLIB に組み込みます。 このライブラリー名は、STEPLIB 内のライブラリー thlqual.SCSQAUTH より前になければなりません。
9. キュー・マネージャーを起動するときに、新しいシステム・パラメーター・モジュールを呼び出します。 例えば、新しいモジュールが NEWMODS という名前の場合は、次のコマンドを発行します。

```
START QMGR PARM(NEWMODS)
```

10. コマンドが正常に完了したことを確認するため、ジョブ・ログを調べます。 ログに次のようなエントリーが含まれているはずです。

```
CSQ9022I CDL1 CSQYASCP 'START QMGR' NORMAL COMPLETION
```

キュー・マネージャー開始 JCL で、パラメーター・モジュール名を指定することもできます。 詳しくは、MQSC を使用した z/OS でのキュー・マネージャーの開始と停止を参照してください。

注 : モジュールの名前として CSQZPARM を選択している場合は、START QMGR コマンドに PARM パラメーターを指定する必要はありません。

システム・パラメーター・モジュールの微調整

IBM MQ には、3 つのアセンブラー・ソース・モジュールからなるセットもあります。このセットは、既存のシステム・パラメーター・モジュールを微調整するために使用します。 これらのモジュールはライブラリー thlqual.SCSQASMS に入っています。 これらのモジュールは、通常、テスト環境で システム・パラメーター・マクロのデフォルト・パラメーターを変更するために使用します。 各ソース・モジュールは、次のようにそれぞれ異なるシステム・パラメーター・マクロを呼び出します。

アセンブラー・ソース・モジュール...	呼び出されるマクロ...
CSQFSYSP	CSQ6SYSP (接続パラメーターおよびトレース・パラメーター)
CSQJLOGP	CSQ6LOGP (ログの初期設定)
CSQJARVP	CSQ6ARVP (保存の初期設定)

上記のモジュールの使用法は次のとおりです。

1. 各アセンブラー・ソース・モジュールの作業用コピーを、ユーザー・アセンブラー・ライブラリーに作成します。
2. 必要に応じてパラメーターの値を追加または変更して、コピーを編集します。
3. 編集したモジュールのコピーをアセンブルして、ユーザー・オブジェクト・ライブラリーにオブジェクト・モジュールを作成します。
4. これらのオブジェクト・コード・モジュールを既存のシステム・パラメーター・モジュールと共にリンク・エディットし、新しいシステム・パラメーター・モジュールとなるロード・モジュールを作成します。
5. 新しいシステム・パラメーター・モジュールが、ユーザー許可ライブラリーのメンバーとなるようにします。

6. このライブラリーを、キュー・マネージャーの開始タスク・プロシージャー STEPLIB に組み込みます。このライブラリーは、STEPLIB 内のライブラリー thlqual.SCSQAUTH より前になければなりません。
7. 前述と同様に、PARM パラメーターに新しいモジュール名を指定した START QMGR コマンドを発行して、新しいシステム・パラメーター・モジュールを呼び出します。

サンプル usermod は、カスタマイズされたシステム・パラメーターを SMP/E 制御の下で管理する方法を示す、SCSQPROC のメンバー CSQ4UZPR で提供されます。

システム・パラメーターの変更

システム・パラメーターの中には、キュー・マネージャーの実行中に変更できるものもあります。 SET SYSTEM、SET LOG、および SET ARCHIVE コマンドを参照してください。

キュー・マネージャーを始動するたびに SET コマンドが有効になるようにするため、コマンドを初期設定入力データ・セットに書き込んでください。

関連概念

959 ページの『チャネル・イニシエーター・パラメーターを調整する』
ALTER QMGR を使用して、要件に適合するようにチャネル・イニシエーターをカスタマイズします。

z/OS *CSQ6SYSP の使用*

このトピックは、CSQ6SYSP を使用したシステム・パラメーターの設定方法のリファレンスとして使用してください。

CSQ6SYSP のデフォルト・パラメーターと、SET SYSTEM コマンドを使用して各パラメーターを変更できるかどうかについては、938 ページの表 76 に示されています。 これらの値を変更する場合は、該当するパラメーターの詳細記述を参照してください。

表 *76. CSQ6SYSP* パラメーターのデフォルト値

パラメーター	説明	デフォルト値	SET コマンド
V 9.3.0 940 ページの『[MQ 9.3.0 Jul 2021]ACCTIME』	アカウンティング・データを収集する時間間隔 (分および秒単位)。	-1	✓
940 ページの『ACELIM』	ACE ストレージ・プールのサイズ (1 KB ブロック単位)。	0 (制限なし)	✓
941 ページの『CLCACHE』	使用するクラスター・キャッシュのタイプを指定します。	STATIC	-
941 ページの『CMDUSER』	コマンド・セキュリティー検査のためのデフォルト・ユーザー ID。	CSQOPR	-
941 ページの『EXCLMSG』	すべてのログから除外するメッセージのリストを指定します。 このリストにあるメッセージは、z/OS コンソールおよびハードコピー・ログに送られません。 結果として、964 ページの『通知メッセージを抑止する』に記載している方式を使用するより、EXCLMSG パラメーターを使用してメッセージを除外する方が、CPU の観点から見れば効率的です	()	✓
942 ページの『EXITLIM』	各呼び出し中にキュー・マネージャー出口が実行できる最大時間 (秒単位)。	30	-
942 ページの『EXITTCB』	キュー・マネージャー出口の実行に使用する開始済みサーバー・タスクの数。	8	-

表 76. *CSQ6SYSP* パラメーターのデフォルト値 (続き)			
パラメーター	**説明**	**デフォルト値**	**SET コマンド**
942 ページの『LOGLOAD』	1 つのチェックポイントの開始から次のチェックポイントの開始までの間に IBM MQ によって書き込まれるログ・レコードの数。	500 000	✓
942 ページの『MULCCAPT』	従量制ソフトウェア料金 (MULC) で使用されるデータを収集するためのアルゴリズムを制御する「従量制価格設定」プロパティーを決定します。	パラメーターの説明を参照してください。	-
943 ページの『OTMACON』	OTMA 接続パラメーター。	パラメーターの説明を参照してください。	-
943 ページの『QINDXBLD』	キュー・マネージャーの再始動を、すべての索引が再作成されるまで待つか、すべての索引が再作成される前に完了するかを決定します。	WAIT	-
944 ページの『QMCCSID』	キュー・マネージャー用のコード化文字セット ID。	ゼロ	-
944 ページの『QSGDATA』	キュー共有グループ・パラメーター。	パラメーターの説明を参照してください。	-
945 ページの『RESAUDIT』	RESLEVEL 監査パラメーター。	YES	-
945 ページの『ROUTCDE』	特定のコンソールから送信請求されていないメッセージに割り当てられるメッセージ宛先コード。	1	-
945 ページの『SERVICE』	IBM の使用のため予約済み。	0	✓
945 ページの『SMFACCT』	キュー・マネージャーが開始されるときに SMF アカウンティング・データを収集するかどうかを指定します。 クラス 4 チャネル・アカウンティング・データは、チャネル・イニシエーターの開始時にのみ収集される点に注意してください。	NO	-
SMFSTAT	キュー・マネージャーが開始されるときに SMF 統計を収集するかどうかを指定します。 クラス 4 チャネル・イニシエーター統計データは、チャネル・イニシエーターの開始時にのみ収集される点に注意してください。	NO	-
SPLCAP	このキュー・マネージャーのキュー・セキュリティー・ポリシー機能を有効にするかどうかを指定します。 Advanced Message Security for z/OS の場合は、このパラメーターを YES に設定します。	NO	-
STATIME	V 9.3.0 統計の各収集の間の時間 (分と秒)。	30	✓
TRACSTR	トレースを自動的に開始するかどうかを指定します。	NO	-

表 76. *CSQ6SYSP* パラメーターのデフォルト値 (続き)			
パラメーター	**説明**	**デフォルト値**	**SET コマンド**
TRACTBL	グローバル・トレース機能によって使用されるトレース表のサイズ (4 KB ブロック単位)。	99 (396 KB)	✓
WLMTIME	WLM 管理対象キューのキュー索引を走査する時間間隔。	30	-
WLMTIMU	WLMTIME の単位 (分または秒)	MINS	-

V 9.3.0 **ACCTIME**

アカウンティング・データの連続収集の間隔 (分と秒) を指定します。

-1 または 0 から 1440 分の範囲の数値を 'mmmm' の形式で指定するか、0 から 1440 分 0 から 59 秒の範囲の数値を 'mmmm.ss' の形式で指定します。

注:

- 秒のみの間隔を指定する場合は、間隔値に接頭部として 0 の値を付加する必要があります。 可能な最小間隔は 1 秒 (0.01) です。
- 0 の値を指定すると、アカウンティング・データは SMF グローバル記録間隔で収集されます。 詳しくは、システム管理機能の使用を参照してください。
- 値 -1 (デフォルト) を指定すると、アカウンティング・データは、STATIME 値で指定した間隔で収集されます。

以下に例を示します。

'0.30' は、30 秒の間隔を設定します。
'5.30' は、5 分 30 秒の間隔を設定します。
'30' は、30 分の間隔を設定します。

ACELIM

ACE ストレージ・プールの最大サイズを指定します (1 KB ブロック単位)。 この数値は 0 から 999999 の範囲でなければなりません。 デフォルト値である 0 は、システムで使用できるサイズ以外の制約は課されないことを意味します。

ACELIM の値は、適正な範囲を超えた量の ECSA ストレージを使用していると見なされているキュー・マネージャーに対してのみ設定する必要があります。 ACE ストレージ・プールを制限すると、システム内の接続数が制限されるという影響があるため、キュー・マネージャーで使用される ECSA ストレージの量も制限されます。

キュー・マネージャーが制限に達すると、アプリケーションは新しい接続を取得できなくなります。 新しい接続を取得できないと MQCONN の処理が失敗し、RRS によって調整されるアプリケーションは、どの IBM MQ API でも障害が発生する可能性があります。

ACE は、接続のスレッド関連制御ブロックに必要な全体の ECSA の約 12.5% を表します。 したがって、例えば ACELIM=5120 を指定すると、キュー・マネージャーによって割り振られる ECSA の総量 (スレッド関連制御ブロックの場合) の上限は、約 40960K; (5120 x 8) になります。

スレッド関連制御ブロック用にキュー・マネージャーによって割り振られる ECSA の総量の上限を 5120K に設定するには、ACELIM 値を 640 に設定する必要があります。

統計 CLASS(3) のトレースで生成される SMF 115 サブタイプ 5 のレコードを使用して、「ACE/PEB」ストレージ・プールのサイズをモニターできます。これにより、ACELIM に適切な値を設定できます。

キュー・マネージャーが制御ブロックに使用する ECSA ストレージの合計量を、統計 CLASS(2) トレースによって書き込まれる SMF 115 サブタイプ 7 のレコードから取得できます。 使用される ECSA ストレージの合計の量は、QSRSPHBGF および QSRSPHBGV フィールドの合計です。

SMF 115 統計レコードについて詳しくは、『IBM MQ パフォーマンス統計の解釈』を参照してください。

ACELIM の設定は、キュー・マネージャーへのアプリケーション接続の制御手段というよりは、z/OS イメージがキュー・マネージャーの誤動作によって影響を受けないよう保護するための手段として検討する必要があります。

CLCACHE
使用するクラスター・キャッシュのタイプを指定します。

クラスター・キャッシュは、クラスターに関連する情報を保管するために使用されるストレージ域です。

クラスター・キャッシュが静的である場合は、キュー・マネージャーの始動時に割り振られる固定サイズになります。 キャッシュが満杯になると、メッセージ CSQM060E が発行され、より多くのスペースを必要とするアプリケーション要求が MQRC_CLUSTER_RESOURCE_ERROR を受け取ります。

CLCACHE を動的に設定すると、クラスター・キャッシュを必要に応じて拡張できます。 ただし、最初に、インストールされているクラスター・ワークロード出口が動的キャッシュを使用して機能できることを確認する必要があります。

インストール済みのクラスター・ワークロード出口が動的キャッシュ・メッセージ CSQM061E で機能できない場合。

MQXCLWLN は、動的キャッシュと静的キャッシュのどちらが使用されているかに関係なく動作する方法でクラスター・ワークロード出口がクラスター・キャッシュをナビゲートするために提供されています。

新しいキュー・マネージャーの場合は、動的キャッシュをサポートしないクラスター・ワークロード出口を使用する場合を除き、CLCACHE=DYNAMIC を設定します。

既存のキュー・マネージャーが既に静的キャッシュを使用しており、多数の新規キューとキュー・マネージャーが追加されていないクラスター内にある場合は、CLCACHE=STATIC を使用して続行するのが妥当です。

既に静的キャッシュを使用しており、多数の新しいキューまたはキュー・マネージャーが追加されるクラスター内にある既存のキュー・マネージャーの場合は、CLCACHE=DYNAMIC を使用して開始します。

STATIC
クラスター・キャッシュが静的であるときには、そのサイズがキュー・マネージャーの開始時に固定されて、クラスター情報の現在量に、拡張のための余分のスペースを加えたサイズになります。 キュー・マネージャーが活動状態であるときには、サイズを増やすことはできません。 これはデフォルトです。

DYNAMIC
クラスター・キャッシュが動的であるときには、キュー・マネージャーの開始時に割り振られた初期サイズを、 キュー・マネージャーが活動状態であるときに必要に応じて自動的に増やすことができます。

CMDUSER
コマンド・セキュリティー検査に使用されるデフォルトのユーザー ID を指定します。 このユーザー ID は、ESM(例えば、RACF)に対して、定義されている必要があります。1 から 8 文字の英数字で名前を指定してください。 最初の文字は英字でなければなりません。

デフォルトは CSQOPR です。

EXCLMSG
除外するエラー・メッセージのリストを指定します。

このリストは動的であり、SET SYSTEM コマンドを使用して更新されます。

デフォルト値は空のリスト ( ) です。

メッセージは、CSQ 接頭部なし、アクション・コード接尾部 (I-D-E-A) なしで指定されます。 例えば、メッセージ CSQX500I を除外するには、このリストに X500 を追加します。 このリストには、最大 16 のメッセージ ID を含めることができます。

リストに含める対象にできるメッセージは、MSTR または CHIN アドレス・スペースの正常始動後に発行され、かつ、先頭文字が E、H、I、J、L、M、N、P、R、T、V、W、X、Y、2、3、5、9 のいずれかの文字であるものです。

コマンド処理結果として発行されるメッセージ ID をリストに追加できますが、除外されません。 例えば、DISPLAY USAGE PSID(*) コマンドの結果としてメッセージ ID が発行されますが、このメッセージを抑止することはできません。

EXITLIM
キュー・マネージャー出口の呼び出しごとに許可される時間 (秒単位) を指定します。 (このパラメーターはチャネル出口には効果がありません。)

5 以上 9999 以下の範囲の値を指定します。

デフォルトは 30 です。 キュー・マネージャーは実行中の出口を 30 秒ごとにポーリングします。 ポーリングのたびに、EXITLIM で指定されている時間を超えて実行されていた出口は強制終了されます。

EXITTCB
キュー・マネージャー内で出口の実行に使用する開始済みサーバー・タスクの数を指定します。 (このパラメーターはチャネル出口には効果がありません。) キュー・マネージャーが実行する可能性のある (チャネル出口以外の) 出口の最大数と同じか、 またはそれよりも大きな数を指定する必要があります。 そのようにしない場合、それは 6c6 異常終了を出して失敗します。

0 ～ 99 の範囲の値を指定してください。 値 0 は出口を実行できないことを意味します。

デフォルトは 8 です。

LOGLOAD
1 つのチェックポイントの開始から次のチェックポイントの開始までの間に IBM MQ が書き込むログ・レコードの数を指定します。 IBM MQ は、指定した数のレコードが書き込まれた後で、新しいチェックポイントを開始します。

200 から 16 000 000 の範囲の値を指定してください。

デフォルトは 500 000 です。

この値が大きければ大きいほど、IBM MQ のパフォーマンスは向上します。ただし、このパラメーターを大きな値に設定すると、再始動の時間が長くなります。

推奨設定値:

テスト・システム	10 000
実動システム	500 000 実動システムで設定されているデフォルト値を使用すると、チェックポイントの出現が頻繁になりすぎる場合があります。

LOGLOAD の値により、キュー・マネージャー・チェックポイントの頻度が決まります。 値が大きすぎると、チェックポイント間のログに書き込まれるデータの量が多くなり、その結果、障害後のキュー・マネージャーの順方向回復再始動時間が長くなります。 値が小さすぎると、ピーク・ロード中にチェックポイントの出現が頻繁になりすぎ、応答時間およびプロセッサーの使用に悪影響が出ます。

LOGLOAD の初期値として 500 000 を推奨します。 1 KB の持続メッセージが 1 秒あたり 100 メッセージの速度で送信される場合 (つまり、コミット付きの MQPUT 100 回と、コミット付きの MQGET 100 回の場合)、チェックポイント間の間隔は約 5 分です。

注 : ここで述べたことはガイドラインにすぎません。このパラメーターの最適値は、個々のシステムの特性に依存します。

MULCCAPT
従量制ソフトウェア料金 (MULC) で使用されるデータの収集に使用されるアルゴリズムを指定します。

STANDARD
MULC は、IBM MQAPI MQCONN 呼び出しから IBM MQ API MQDISC 呼び出しまでの時間に基づきます。

REFINED
MULC は、IBM MQ API 呼び出しが開始してから IBM MQ API 呼び出しが終了するまでの時間に基づきます。

デフォルトは STANDARD です。

OTMACON
OTMA パラメーター。 このキーワードは、5 つの定位置パラメーターをとります。

OTMACON = (Group,Member,Druexit,Age,Tpipepfx)

グループ
IBM MQ のこの特定インスタンスが属している XCF グループの名前です。

長さは 1 〜 8 文字です。大文字で入力する必要があります。

デフォルトはブランクです。これは、IBM MQ が XCF グループに結合しないように指示します。

メンバー
XCF グループ内における IBM MQ のこの特定インスタンスのメンバー名です。

長さは 1 〜 16 文字です。大文字で入力する必要があります。

デフォルトは 4 文字のキュー・マネージャー名です。

Druexit
IMS によって実行される OTMA 宛先解決ユーザー出口の名前を指定します。

長さは 1 〜 8 文字です。

デフォルトは DFSYDRU0 です。

このパラメーターはオプションですが、IBM MQ によって開始されなかった IMS アプリケーションから IBM MQ がメッセージを受け取る場合は必須になります。 この名前は、IMS システムでコーディングされた宛先解決ユーザー出口に対応している必要があります。 詳しくは、1029 ページの『IMS における OTMA 出口の使用』.を参照してください。

Age
IBM MQ からのユーザー ID が以前に IMS によって検査されたと考えられる時間の長さ (秒単位) を表します。

範囲は 0 〜 2 147 483 647 です。

デフォルトは 2 147 483 647 です。

メインフレーム内のセキュリティー・キャッシュ設定の整合性を保守するために、このパラメーターを ALTER SECURITY コマンドの `interval` パラメーターと共に設定することが勧められています。

Tpipepfx
Tpipe 名に使用される接頭部を表します。

これは、3 文字で構成されます。一番目の文字は A から Z までの範囲です。後続の文字は A から Z、または 0 から 9 です。 デフォルトは CSQ です。

これは IBM MQ が Tpipe を作成するたびに使用されます。名前の残りの部分は IBM MQ によって割り当てられます。 IBM MQ によって作成される Tpipe の名前全体を設定することはできません。

QINDXBLD
キュー・マネージャーの再始動を、すべてのキュー索引が再作成されるまで待つか、すべての索引が再作成される前に完了するかを決定する。

WAIT
キュー・マネージャーの再始動を、すべてのキュー索引作成が完了するまで待ちます。 これは、索引が作成されている間は、通常の IBM MQ API 処理中に遅れが発生するアプリケーションがないことを意味します。アプリケーションがキュー・マネージャーに接続できるようになる前に、すべての索引が作成されるからです。

これはデフォルトです。

NOWAIT
キュー・マネージャーは、すべてのキュー索引作成が完了する前に再始動できます。

QMCCSID
キュー・マネージャーで (したがって、分散キューイングで) 使用されるデフォルトのコード化文字セット ID を指定します。

0 から 65535 の範囲内で値を指定します。 この値は、 各国語で選択した言語のネイティブ z/OS コード・ページとしてリストされている EBCDIC コード・ページを表す必要があります。

デフォルト値の 0 を指定すると、 現在設定されている CCSID が使用されます。設定されているものがない場合は、CCSID 500 を使用します。 つまり、CCSID をゼロ以外の値に明示的に設定している場合、それを QMCCSID をゼロに設定することによってリセットできません。 その後、正しいゼロ以外の CCSID を使用しなければなりません。 QMCCSID がゼロの場合、コマンド DISPLAY QMGR CCSID を発行して、実際に使用されている CCSID が何であるかを調べることができます。

注 : キュー共有グループ内のすべてのキュー・マネージャーは、同じ QMCCSID を使用する必要があります。

QSGDATA
キュー共有グループ・データ。 このキーワードは、5 つの定位置パラメーターをとります。

QSGDATA=(Qsgname,Dsgname,Db2name,Db2serv ,Db2blob)

Qsgname
これは、キュー・マネージャーが属しているキュー共有グループの名前です。

有効な文字は IBM MQ オブジェクトの命名規則を参照してください。 名前には次の条件があります。

- 長さは 1 ～ 4 文字です
- 名前の先頭は数字であってはなりません
- @ で終わってはなりません。

  これは、実装上の理由により、4 文字未満の名前は内部的に @ 記号が埋め込まれるためです。

デフォルトはブランクです。これはキュー・マネージャーがキュー共有グループのメンバーではないことを示します。

Dsgname
これは、キュー・マネージャーが接続している Db2 データ共有グループの名前です。

長さは 1 ～ 8 文字です。大文字で入力する必要があります。

デフォルトはブランクです。これはキュー共有グループを使用しないことを示します。

Db2name
これは、キュー・マネージャーが接続している Db2 サブシステムまたはグループの名前です。

長さは 1 ～ 4 文字です。大文字で入力する必要があります。

デフォルトはブランクです。これはキュー共有グループを使用しないことを示します。

注 : Db2 サブシステム (またはグループ接続) は、 Dsgname で指定された Db2 データ共用グループ内になければならず、すべてのキュー・マネージャーは同じ Db2 データ共用グループを指定しなければなりません。

Db2serv
これは、Db2 にアクセスするために使用するサーバー・タスクの数です。

範囲は 4 ～ 10 です。

デフォルトは 4 です。

Db2blob
これは、バイナリー・ラージ・オブジェクト (BLOB) にアクセスするために使用する Db2 タスクの数です。

範囲は 4 ～ 10 です。

デフォルトは 4 です。

名前パラメーターの 1 つ (つまり、 **Qsgname**、 **Dsgname**、または **Db2name**) を指定する場合は、他の名前の値を入力する必要があります。そうしないと、 IBM MQ は失敗します。

RESAUDIT
接続処理中に実行される RESLEVEL セキュリティー検査のための RACF 監査レコードが作成されるかどうか指定します。

以下のいずれかを指定します。

NO
RESLEVEL 監査は実行されません。

YES
RESLEVEL 監査は実行されます。

デフォルトは「はい」です。

ROUTCDE
メッセージに割り当てられるデフォルトの z/OS メッセージ宛先コードを指定します。このメッセージは、MQSC コマンドへの直接的な応答として送られるものではありません。

以下のいずれかを指定します。

1. 1 ～ 16 の範囲の値 (1 と 16 を含む)。
2. 値のリスト (値をコンマで区切り、括弧で囲む)。 それぞれの値は 1 ～ 16 までの範囲になければなりません (1 と 16 を含む)。

デフォルトは 1 です。

z/OS 宛先コードについて詳しくは、「 *z/OS MVS* システム・メッセージ 」マニュアルのいずれかのボリュームにある「 メッセージの説明 」の「 宛先コード 」を参照してください。

SERVICE
このフィールドは IBM の使用のために予約されています。

SMFACCT
キュー・マネージャーの開始時に IBM MQ がアカウンティング・データを SMF に自動的に送信するかどうかを指定します。

以下のいずれかを指定します。

NO
アカウンティング・データの収集を自動的には開始しません。

YES
デフォルト・クラス 1 に対応するアカウンティング・データの収集を自動的に開始します。

整数
V 9.3.0 アカウンティング・データが 1 から 4 の範囲で自動的に収集されるクラスのリスト。

V 9.3.0 * クラス 1、2、および 3 の SMF アカウンティングを自動的に開始します。

デフォルトは NO です。

SMFSTAT
キュー・マネージャーが開始したとき、SMF 統計を自動的に収集するかどうかを指定します。

以下のいずれかを指定します。

NO
統計の収集を自動的には開始しません。

YES
デフォルト・クラス 1 に対応する統計の収集を自動的に開始します。

整数
V 9.3.0 統計が自動的に収集されるクラスのリスト (1 から 5 の範囲)。

クラス 2 または 3 の統計を収集するには、クラス 1 も指定する必要があります。

V 9.3.0 * クラス 1、2、および 3 の SMF 統計を自動的に開始します。

デフォルトは NO です。

SPLCAP
セキュリティー・ポリシー機能は、キューに対するメッセージの書き込みおよび読み取り時にメッセージへの署名 (または暗号化) を行うかどうかを制御するポリシーを介在させることにより、メッセージ・セキュリティーのレベルを高めます。

SPLCAP を以下のいずれかの値に設定することにより、このキュー・マネージャーのセキュリティー・ポリシー処理が構成されます。

NO
キューのためのメッセージ・セキュリティー・ポリシーを実装する機能は、キュー・マネージャーの初期化時に有効になりません。

YES
キュー・マネージャーの初期化時にメッセージ・セキュリティー機能が有効になります。

キュー・マネージャーは、AMSPROD 属性が AMS、ADVANCED、または ADVANCEDVUE のいずれかに設定されていることを検査します。この場合、キュー・マネージャーは AMS のライセンス交付を受けています。 それ以外の場合は、開始されません。

キュー・マネージャーは、必要な AMS 構成が設定されているかどうかも検査します。 そうでない場合、キュー・マネージャーは始動しません。

キュー・マネージャーが両方とも AMS のライセンス交付を受けており、必要な構成が行われている場合、キュー・マネージャーの初期化中にメッセージ・セキュリティー機能を有効にしてキュー・マネージャーが開始され、AMSM アドレス・スペースが開始されます。

デフォルトは NO です。

STATIME
V 9.3.0 IBM MQ for z/OS 9.3.0 以降では、統計データの連続収集の時間間隔を分と秒で指定します。 ACCTIME が設定されていないか、-1 に設定された場合は、アカウンティング・データの連続収集の時間間隔も指定します。

0 から 1440 分までの範囲の数値を 'mmmm' の形式で指定するか、0 から 1440 分までの範囲の数値と、0 から 59 秒までの範囲の数値を 'mmmm.ss' の形式で指定します。 デフォルトは 30 分です。

注:

- 秒のみの間隔を指定する場合は、間隔値に接頭部として 0 の値を付加する必要があります。 指定可能な最小の間隔は 1 秒 ('0.01') です。
- V 9.3.0 IBM MQ for z/OS 9.3.0 以降、0 の値を指定した場合、統計データが SMF データ収集ブロードキャストで収集されます。 ACCTIME が指定されていないか、-1 の場合、SMF データ収集ブロードキャスト時にアカウンティング・データも収集されます。 詳しくは、システム管理機能の使用を参照してください。
- 値 -1 (デフォルト) を指定すると、アカウンティング・データは、STATIME 値で指定した間隔で収集されます。

TRACSTR
グローバル・トレースを自動的に開始するかどうかを指定します。

以下のいずれかを指定します。

NO
グローバル・トレースを自動的には開始しません。

YES
デフォルト・クラス (クラス 1) に対応するグローバル・トレースを自動的に開始します。

整数
グローバル・トレースが自動的に開始されるクラスのリストです。範囲は 1 ～ 4 です。

*****
すべてのクラスについて、グローバル・トレースを自動的に開始します。

マクロ内にキーワードを指定していない場合、デフォルトは NO です。

注 : 提供されているデフォルトのシステム・パラメーター・ロード・モジュール (CSQZPARM) では TRACSTR=YES です (アセンブラー・モジュール CSQFSYSP で設定されています)。 トレースを自動的に開始させたくない場合は、 ユーザー独自のシステム・パラメーター・モジュールを作成するか、 キュー・マネージャーが開始したあとで STOP TRACE コマンドを実行します。

STOP TRACE コマンドについて詳しくは、STOP TRACE を参照してください。

TRACTBL
グローバル・トレース機能により IBM MQ トレース・レコードが保管されるトレース表のサイズを 4 KB ブロック単位で指定します。

1 ～ 999 の範囲の値を指定してください。

デフォルトは 99 です。 これは 396 KB に相当します。

注 : トレース表のストレージは ECSA に割り振られます。 したがって、この値の選択は慎重に行ってください。

WLMTIME
WLM 管理対象キューの索引を走査する時間間隔 (WLMTIMU の値によって、分単位または秒単位) を指定します。

1 から 9999 の範囲内で値を指定します。

デフォルトは 30 です。

WLMTIMU
WLMTIME パラメーターで使用する時間単位。

次のうちの 1 つを指定します。

MINS
WLMTIME は分数を表します。

SECS
WLMTIME は秒数を表します。

デフォルトは MINS です。

関連資料
947 ページの『CSQ6LOGP の使用』
このトピックは、CSQ6LOGP を使用したロギング・オプションの指定方法の参照として使用します。

951 ページの『CSQ6ARVP の使用』
このトピックは、CSQ6ARVP を使用したアーカイブ環境の指定方法の参照として使用します。

z/OS *CSQ6LOGP* の使用

このトピックは、CSQ6LOGP を使用したロギング・オプションの指定方法の参照として使用します。

CSQ6LOGP を使用して、ロギング・オプションを設定します。

CSQ6LOGP のデフォルト・パラメーター、および ログの設定コマンドを使用して、各パラメーターを変更できるかどうかは、CSQ6LOGP パラメーターのデフォルト値に示されています。 これらの値を変更する必要がある場合は、該当するパラメーターの詳細記述を参照してください。

表 77. *CSQ6LOGP* パラメーターのデフォルト値

パラメーター	説明	デフォルト値	SET コマンド
COMPLOG	ログ圧縮を使用可能にするかどうかを制御します。	NONE	X
DEALLCT	保存テープ装置が割り振り解除される前に未使用状態になっている時間の長さ	ゼロ	X

表 77. *CSQ6LOGP* パラメーターのデフォルト値 (続き)			
パラメーター	**説明**	**デフォルト値**	**SET コマンド**
INBUFF	アクティブ・ログ・データ・セットおよびアーカイブ・ログ・データ・セットの入力バッファー・ストレージのサイズ	60 KB	-
MAXARCH	記録できるアーカイブ・ログ・ボリュームの最大数	500	X
MAXCNOFF	並行して実行できる CSQJOFF7 オフロード・タスクの最大数。	31	-
MAXRTU	読み取りアーカイブ・ログ・テープ・ボリュームに同時に割り振られる専用テープ装置の最大数	2	X
OFFLOAD	保存のオン/オフ	YES (ON)	-
OUTBUFF	アクティブ・ログ・データ・セットおよびアーカイブ・ログ・データ・セットの出力バッファー・ストレージのサイズ	4 000 KB	-
TWOACTV	単一または重複の活動ロギング	YES (重複)	-
TWOARCH	単一または重複の保存ロギング	YES (重複)	-
TWOBSDS	単一または重複の BSDS	YES (重複 BSDS)	-
WRTHRSH	アクティブ・ログ・データ・セットに書き込まれる前に、満杯になる出力バッファーの数	20	X
ZHYWRITE	zHyperWrite フィーチャーを有効にするかどうかを指定します。	NO	X

COMPLOG
ログ圧縮を使用可能にするかどうかを指定します。

次のどちらかを指定します。

NONE
ログ圧縮は使用可能にされません。

RLE
ログ圧縮は、ラン・レングス・エンコードを使用して使用可能にされます。

ANY
キュー・マネージャーは、最高度のログ・レコード圧縮を提供する圧縮アルゴリズムを選択します。 このオプションを指定すると、RLE 圧縮が行われます。

デフォルトは NONE です。

ログ圧縮について詳しくは、ログ圧縮を参照してください。

DEALLCT
保存読み取りテープ装置が割り振り解除される前に未使用状態になることができる時間の長さ (分単位) を指定します。

次の 1 つを指定します。

- 0 ～ 1440 の範囲の時間 (分単位)。
- NOLIMIT

1440 または NOLIMIT を指定すると、テープ装置が割り振り解除されないことを意味します。

デフォルト値は 0 です。

アーカイブ・ログ・データがテープから読みとられる際に、IBM MQ が複数の読み取りアプリケーション用にテープ処理を最適化できるように、十分な大きさの値を設定するようにお勧めします。

INBUFF
回復時にアクティブ・ログおよびアーカイブ・ログを読み込むための入力バッファーのサイズを K バイト単位で指定します。 28 ～ 60 の範囲の 10 進数を使用してください。 指定した数は、4 の倍数に切り上げられます。

デフォルトは 60 KB です。

推奨設定値:

テスト・システム	28 KB
実動システム	60 KB 最良のパフォーマンスでログ読み取りを行えるように、この値を最大に設定します。

MAXARCH
BSDS に記録できる保存ログ・ボリュームの最大数を指定します。 この数を超えると、再び BSDS の開始位置から記録が始まります。

10 ～ 1000 の範囲の 10 進数を使用してください。

デフォルトは 500 です。

推奨設定値:

テスト・システム	500 (デフォルト)
実動システム	1 000 できるだけ多くのログを BSDS に記録できるように、この値を最大に設定します。

ログおよび BSDS について詳しくは、IBM MQ リソースの管理を参照してください。

MAXCNOFF
並行して実行できる CSQJOFF7 オフロード・タスクの数を指定します。

これにより、使用可能なテープ装置をすべて使用してしまうことがないように、キュー・マネージャー (複数可) を調整することができます。

キュー・マネージャーは、CSQJOFF7 オフロード・タスクが完了するまで待機してから、新たに保存データ・セットの割り振りを試みます。

キュー・マネージャーがテープに保存する場合は、同時になされるテープ要求の数が、使用可能なテープ装置の数以上にならないように、このパラメーターを設定してください。 そのようにしないと、システムがハングする可能性があります。

なお、重複保存を使用している場合は、オフロード・タスクごとに両方の保存を実行するため、それに合わせてこのパラメーターを設定する必要があります。 例えば、キュー・マネージャーがテープへの重複保存をしている場合は、値を MAXCNOFF=2 にすると、最大で 2 つのアクティブ・ログを 4 つのテープに同時に保存することができます。

複数のキュー・マネージャーがテープ装置を共有している場合は、キュー・マネージャーごとに MAXCNOFF を適宜設定する必要があります。

デフォルト値は 31 です。

1 ～ 31 の範囲の値を指定してください。

MAXRTU
読み取りアーカイブ・ログ・テープ・ボリュームに同時に割り振られる専用テープ装置の最大数を指定します。

このパラメーターと DEALLCT パラメーターを指定すると、IBM MQ はテープ装置からのアーカイブ・ログの読み取りを最適化することができます。

1 ～ 99 の範囲の値を指定してください。

デフォルトは 2 です。

値は IBM MQ に使用可能なテープ装置の数より、少なくとも 1 つ小さく設定するようにお勧めします。そうしないと、オフロード・プロセスの遅延が起こり、システムのパフォーマンスに影響を与える可能性があります。 アーカイブ・ログ処理中にスループットを最大にするには、このオプションに設定できる最大値を指定します。ただし、オフロード・プロセス用に少なくとも 1 つのテープ装置が必要であることを覚えておいてください。

OFFLOAD
アーカイブのオン/オフを指定します。

次のどちらかを指定します。

YES
保存はオン

NO
保存はオフ

デフォルトは「はい」です。

注意 : テスト環境で作業している場合以外は、保存をオフに**切り替えない**でください。 保存をオフにすると、システムまたはトランザクションの障害が発生した場合に、データを回復できないことがあります。

OUTBUFF
アクティブ・ログ・データ・セットおよびアーカイブ・ログ・データ・セットを書き込むための出力バッファー用に IBM MQ が使用するストレージのサイズの合計を K バイト単位で指定します。 出力バッファーはそれぞれ 4 KB です。

このパラメーターは 128 ～ 4000 の範囲でなければなりません。 指定した数は、4 の倍数に切り上げられます。 40 から 128 までの値は、互換性の理由により受け入れられ、値 128 として扱われます。

デフォルトは 4000 KB です。

推奨設定値:

テスト・システム	400 KB
実動システム	4 000 KB ログ出力バッファーが容量不足にならないように、この値を最大に設定します。

TWOACTV
単一または重複の活動ロギングを指定します。

次のどちらかを指定します。

NO
単一アクティブ・ログ

YES
重複アクティブ・ログ

デフォルトは「はい」です。

単一ロギングおよび重複ロギングの使用方法について詳しくは、IBM MQ リソースの管理を参照してください。

TWOARCH
アクティブ・ログがオフロードされる場合に IBM MQ が作成するアーカイブ・ログの数を指定します。

次のどちらかを指定します。

NO
単一アーカイブ・ログ

YES
重複アーカイブ・ログ

デフォルトは「はい」です。

推奨設定値:

テスト・システム	NO
実動システム	YES (デフォルト)

単一ロギングおよび重複ロギングの使用方法について詳しくは、IBM MQ リソースの管理を参照してください。

TWOBSDS
ブートストラップ・データ・セットの数を指定します。

次のどちらかを指定します。

NO
単一 BSDS

YES
重複 BSDS

デフォルトは「はい」です。

単一ロギングおよび重複ロギングの使用方法について詳しくは、IBM MQ リソースの管理を参照してください。

WRTHRSH
活動ログ・データ・セットに書き込まれる前に満杯になる 4 KB 出力バッファーの数を指定します。

バッファーの数が多いほど、書き込みの回数が少なくなり、IBM MQ のパフォーマンスが向上します。コミット点などのような重要なイベントが発生した場合は、この数に達する前にバッファーが書き込まれることがあります。

1 ～ 256 の範囲のバッファー数を指定します。

デフォルトは 20 です。

ZHYWRITE
有効になっている zHyperWrite を使用してアクティブ・ログへの書き込みが行われるかどうかを指定します。

zHyperWrite によってアクティブ・ログを有効にする方法について詳しくは、zHyperWrite と IBM MQ アクティブ・ログの使用を参照してください。

値は次のいずれかです。

NO
zHyperWrite は無効です。

YES
zHyperWrite は有効です。

関連資料
938 ページの『CSQ6SYSP の使用』
このトピックは、CSQ6SYSP を使用したシステム・パラメーターの設定方法のリファレンスとして使用してください。

951 ページの『CSQ6ARVP の使用』
このトピックは、CSQ6ARVP を使用したアーカイブ環境の指定方法の参照として使用します。

z/OS *CSQ6ARVP* の使用
このトピックは、CSQ6ARVP を使用したアーカイブ環境の指定方法の参照として使用します。

CSQ6ARVP を使用して、保存環境を設定します。

CSQ6ARVP のデフォルト・パラメーターと、SET ARCHIVE コマンドを使用して各パラメーターを変更できるかどうかについては、952 ページの表 78 に示されています。 これらの値を変更する必要がある場合は、該当するパラメーターの詳細記述を参照してください。 ストレージの計画について詳しくは、z/OS でのストレージ要件およびパフォーマンス要件の計画 を参照してください。

表 *78. CSQ6ARVP* パラメーターのデフォルト値

パラメーター	説明	デフォルト値	SET コマンド
ALCUNIT	1 次および 2 次のスペース割り振りが行われる単位	BLK (ブロック数)	X
ARCPFX1	最初のアーカイブ・ログ・データ・セット名の接頭部	CSQARC1	X
ARCPFX2	2 番目のアーカイブ・ログ・データ・セット名の接頭部	CSQARC2	X
ARCRETN	アーカイブ・ログ・データ・セットの保存期間 (日数)	9999	X
ARCWRTC	アーカイブ・ログ・データ・セットに関するオペレーターあてメッセージの宛先コードのリスト	1、3、4	X
ARCWTOR	アーカイブ・ログ・データ・セットの取り付けを試みる前に、オペレーターにメッセージを送って応答を待つかどうか	YES	X
blksize	アーカイブ・ログ・データ・セットのブロック・サイズ	28 672	X
CATALOG	アーカイブ・ログ・データ・セットを ICF にカタログ化するかどうか	NO	X
COMPACT	アーカイブ・ログ・データ・セットを圧縮するかどうか	NO	X
PRIQTY	DASD データ・セットの 1 次スペース割り振り	25 715	X
PROTECT	データ・セットが作成される場合、ESM プロファイルによってアーカイブ・ログ・データ・セットが保護されるかどうか	NO	X
QUIESCE	MODE(QUIESCE) を指定して ARCHIVE LOG を出す場合に静止状態が許可される最大時間 (秒単位)	5	X
SECQTY	DASD データ・セットの 2 次スペース割り振り。 使用される単位については、ALCUNIT パラメーターの項を参照してください。	540	X
TSTAMP	保存データ・セット名にタイム・スタンプを設定するかどうか	NO	X
unit	アーカイブ・ログ・データ・セットの最初のコピーが保管される装置タイプまたは装置名	TAPE	X
UNIT2	アーカイブ・ログ・データ・セットの 2 番目のコピーが保管される装置タイプまたは装置名	ブランク	X

ALCUNIT

1 次および 2 次のスペース割り振りが行われる単位を指定します。

以下のいずれかを指定します。

CYL
シリンダー

TRK
トラック

BLK
ブロック

BLK は装置タイプに依存しないので、BLK を使用することをお勧めします。

デフォルトは BLK です。

保存 DASD ボリュームのフリー・スペースがフラグメント化している可能性がある場合は、 より小さい 1 次エクステントを指定して、2 次エクステントへ拡張できるようにすることをお勧めします。 アクティブ・ログのスペース割り振りについて詳しくは、 ログ・アーカイブ・ストレージの計画を参照してください。

ARCPFX1
最初の保存ログ・データ・セット名の接頭部を指定します。

データ・セット名の設定方法および ARCPFX1 の長さの制限については、TSTAMP パラメーターの項を参照してください。

このパラメーターをブランクにすることはできません。

デフォルトは CSQARC1 です。

この接頭部の付いたアーカイブ・ログを作成する許可を、IBM MQ キュー・マネージャーのアドレス・スペースに関連付けられたユーザー ID に与えることが必要な場合があります。

ARCPFX2
2 番目の保存ログ・データ・セット名の接頭部を指定します。

データ・セット名の設定方法および ARCPFX2 の長さの制限については、TSTAMP パラメーターの項を参照してください。

このパラメーターは、TWOARCH パラメーターに NO を指定した場合でも、ブランクにすることはできません。

デフォルトは CSQARC2 です。

この接頭部の付いたアーカイブ・ログを作成する許可を、IBM MQ キュー・マネージャーのアドレス・スペースに関連付けられたユーザー ID に与えることが必要な場合があります。

ARCRETN
アーカイブ・ログ・データ・セットが作成される場合に使用される保存期間を日数で指定します。

このパラメーターは 0 ～ 9999 の範囲でなければなりません。

デフォルトは 9999 です。

推奨設定値:

テスト・システム	3 テスト・システムでは、長期間のアーカイブ・ログは必要ありません。
実動システム	9 999 (デフォルト) アーカイブ・ログ自動削除を効率的にオフに切り替えできるように、この値を高く設定します。

アーカイブ・ログ・データ・セットの廃棄については、保存ログ・データ・セットの廃棄を参照してください。

ARCWRTC
アーカイブ・ログ・データ・セットに関するオペレーター向けメッセージの z/OS 宛先コードのリストを指定します。 ARCWTOR が NO に設定されている場合は、このフィールドは無視されます。

それぞれ 1 以上 16 以下の範囲の値の宛先コードを、14 個まで指定できます。 少なくとも 1 つのコードを指定する必要があります。 コードとコードの間は、ブランクではなく、コンマで区切ります。

デフォルトは 1、3、4 という値のリストです。

z/OS 宛先コードについて詳しくは、「*z/OS MVS* システム・メッセージ」マニュアルのいずれかのボリュームにある「メッセージの説明」の「宛先コード」を参照してください。

ARCWTOR
アーカイブ・ログ・データ・セットの取り付けを試みる前に、オペレーターにメッセージを送って応答を受信するかどうかを指定します。

その他の IBM MQ ユーザーは、データ・セットがマウントされるまで強制的に待機させられることがありますが、IBM MQ がメッセージへの応答を待機している間は影響を受けません。

次のどちらかを指定します。

YES
装置は、保存ログ・データ・セットが取り付けられるまで、長い時間を必要とします。 例えば、テープ装置がこれに該当します。

NO
装置は長時間の遅延を必要としません。 例えば、DASD がこれに該当します。

デフォルトは「はい」です。

推奨設定値:

テスト・システム	NO
実動システム	YES (デフォルト) これは操作手順に依存します。 テープ・ロボットを使用する場合は、NO のほうが適切です。

BLKSIZE
保存ログ・データ・セットのブロック・サイズを指定します。 指定するブロック・サイズは、UNIT パラメーターで指定する装置タイプと互換性がなければなりません。

このパラメーターは 4 097 から 28 672 の範囲でなければなりません。 指定した値は 4 096 の倍数に切り上げられます。

デフォルトは 28 672 です。

このパラメーターは、提供されている場合、ストレージ管理サブシステム (SMS) データ・クラスのブロック・サイズによって指定変更されます。

アーカイブ・ログ・データ・セットが DASD に書き込まれる場合は、トラックごとに 2 ブロックが許容される最大ブロック・サイズを選択することをお勧めします。 例えば、3390 装置の場合、24 576 のブロック・サイズを使用します。

アーカイブ・ログ・データ・セットをテープに書き込む場合、可能な最大のブロック・サイズを指定すると、アーカイブ・ログの読み取り速度が向上します。 ブロック・サイズとして 28 672 を使用してください。

推奨設定値:

テスト・システム	アーカイブ・ログのために使用するメディアに応じたブロック・サイズの推奨値を使用します。 つまり、ディスクの場合は 24 576、およびテープの場合は 28 672 です。
実動システム	アーカイブ・ログのために使用するメディアに応じたブロック・サイズの推奨値を使用します。 つまり、ディスクの場合は 24 576、およびテープの場合は 28 672 です。

CATALOG
アーカイブ・ログ・データ・セットを 1 次統合カタログ機能 (ICF) カタログにカタログ化するかどうかを指定します。

次のどちらかを指定します。

NO
アーカイブ・ログ・データ・セットはカタログ化されません。

YES
アーカイブ・ログ・データ・セットはカタログ化されます。

デフォルトは NO です。

DASD に割り振られているすべてのアーカイブ・ログ・データ・セットを、カタログ化する必要があります。 CATALOG パラメーターを NO に設定して DASD に保存すると、アーカイブ・ログ・データ・セットが割り振られるたびにメッセージ CSQJ072E が表示され、IBM MQ によってデータ・セットがカタログされます。

推奨設定値:

テスト・システム	YES
実動システム	YES (アーカイブが DASD に割り振られる場合)

COMPACT
保存ログに書き込まれたデータを圧縮するかどうかを指定します。 このオプションは、改良データ記録機能 (IDRC) を備えた 3480 または 3490 装置だけに適用されます。 この機能がオンになっていると、テープ制御装置のハードウェアは通常よりかなり高い密度でデータを書き込むため、1 つのボリュームにより多くのデータを記録することができます。 3480 装置に IDRC 機能または 3490 基本モデル (ただし 3490E は除く) が装備されている場合は、NO を指定します。 データを圧縮する場合は、YES を指定します。

次のどちらかを指定します。

NO
データ・セットを圧縮しません。

YES
データ・セットを圧縮します。

デフォルトは NO です。

YES を指定すると、パフォーマンスに悪影響を与えます。 また、テープに圧縮されたデータは IDRC 機能をサポートする装置を使用しなければ読めないことも念頭に置いてください。 保存テープをリモート回復のために別のサイトに送る必要がある場合は、これも考慮する必要があります。

推奨設定値:

テスト・システム	適用外
実動システム	NO (デフォルト) これは 3480 および 3490 IDR 圧縮だけに適用されます。 YES に設定すると、回復および再始動時のアーカイブ・ログの読み取りパフォーマンスが低下する可能性がありますが、テープへの書き込みには影響しません。

PRIQTY
DASD データ・セットの 1 次スペース割り振りを ALCUNIT の単位で指定します。

値はゼロより大きくなければなりません。

デフォルトは 25 715 です。

ログ・データ・セットまたはそれに対応する BSDS のどちらか大きい方をコピーする場合は、この値で十分です。 必要な値を求めるには、次の手順に従ってください。

1. 934 ページの『ブートストラップ・データ・セットおよびログ・データ・セットを作成する』で説明されているように、割り振られているアクティブ・ログ・レコードの数 (c) を判別します。
2. 次のように、各アーカイブ・ログ・ブロックの中の 4096 バイト・ブロックの数を決定します。

```
d = BLKSIZE / 4096
```

   ここで、BLKSIZE は切り上げた値です。
3. ALCUNIT=BLK の場合:

```
PRIQTY = INT(c / d) + 1
```

   ここで、INT は整数まで端数を切り捨てた値です。

   ALCUNIT=TRK の場合:

```
PRIQTY = INT(c / (d * INT(e/BLKSIZE))) + 1
```

   ここで、e はトラックごとのバイト数 (3390 装置の場合は 56664) であり、INT は整数まで端数を切り捨てた値です。

   ALCUNIT=CYL の場合:

```
PRIQTY = INT(c / (d * INT(e/BLKSIZE) * f)) + 1
```

   ここで、f はシリンダーごとのトラック数 (3390 装置の場合は 15) であり、INT は整数まで端数を切り捨てた値です。

ログ・データ・セットおよびアーカイブ・ログ・データ・セットの大きさについては、934 ページの『ブートストラップ・データ・セットおよびログ・データ・セットを作成する』 および 935 ページの『ページ・セットの定義』 を参照してください。

推奨設定値:

テスト・システム 1 680

活動ログ全体を保持する場合は、以下の値で十分です。

```
10  080 / 6 = 1  680 blocks
```

実動システム テープに保存する場合は適用されません。

保存 DASD ボリュームのフリー・スペースがフラグメント化している可能性がある場合は、 より小さい 1 次エクステントを指定して、2 次エクステントへ拡張できるようにすることをお勧めします。 アクティブ・ログのスペース割り振りについて詳しくは、 ログ・アーカイブ・ストレージの計画を参照してください。

PROTECT
保存ログ・データ・セットが作成される場合、離散 ESM (外部セキュリティー管理プログラム) プロファイルによってそのデータ・セットが保護されるかどうかを指定します。

次のどちらかを指定します。

NO
プロファイルは作成されません。

YES
ログをオフロードする場合、離散データ・セット・プロファイルが作成されます。 YES を指定する場合は、以下の条件を満たしている必要があります。

- ESM 保護は IBM MQ に対して活動状態でなければならない。
- IBM MQ キュー・マネージャー・アドレス・スペースに関連しているユーザー ID に、これらのプロファイルを作成する権限が必要である。
- テープに保存している場合、TAPEVOL クラスは活動状態でなければならない。

そうでない場合、オフロードは失敗します。

デフォルトは NO です。

QUIESCE
MODE (QUIESCE) を指定して ARCHIVE LOG コマンドを発行するときに、静止状態が許される最大時間 (秒数) を指定します。

このパラメーターは 1 ～ 999 の範囲でなければなりません。

デフォルトは 5 です。

SECQTY
DASD データ・セットの 2 次スペース割り振りを ALCUNIT の単位で指定します。 2 次エクステントは、最大 15 回割り振ることができます。ALCUNIT について詳しくは、「IBM z/OS Management Facility プログラミング・ガイド」を参照してください。

パラメーターは 0 より大きくなければなりません。

デフォルトは 540 です。

TSTAMP
保存ログ・データ・セット名にタイム・スタンプを含めるかどうかを指定します。

次のどちらかを指定します。

NO
名前にタイム・スタンプは入りません。 保存ログ・データ・セットの名前は、次のように設定されます。

```
arcpfxi.A nnnnnnn
```

この場合、*arcpfxi* は、ARCPFX1 または ARCPFX2 で指定されているデータ・セット名接頭部です。 *arcpfxi* は、最大 35 文字を保持することができます。

YES
名前にタイム・スタンプを入れます。 保存ログ・データ・セットの名前は、次のように設定されます。

```
arcpfxi.cyyddd.T hhmmsst.A nnnnnnn
```

この場合、*c* は、1999 年までは「D」、2000 年以降は「E」になります。*arcpfxi* は、ARCPFX1 または ARCPFX2 で指定されているデータ・セット名接頭部です。 *arcpfxi* は、最大 19 文字を保持することができます。

EXT
名前にタイム・スタンプを入れます。 保存ログ・データ・セットの名前は、次のように設定されます。

```
arcpfxi.D yyyyddd.T hhmmsst.A nnnnnnn
```

この場合、*arcpfxi* は、ARCPFX1 または ARCPFX2 で指定されているデータ・セット名接頭部です。 *arcpfxi* は、最大 17 文字を保持することができます。

デフォルトは NO です。

UNIT
保存ログ・データ・セットの最初のコピーの保管に使用する装置の装置タイプまたは装置名を指定します。

装置タイプまたは装置名は 1 ～ 8 文字の英数字で指定します。 最初の文字は英字でなければなりません。

このパラメーターをブランクにすることはできません。

デフォルトは TAPE です。

DASD にアーカイブする場合は、総称装置タイプと限定するボリューム範囲を指定することができます。例: UNIT=3390。

DASD に保存する場合は、次の点に注意してください。

- 1 次スペース割り振りは、アクティブ・ログ・データ・セットのすべてのデータを入れるのに十分かどうか。
- アーカイブ・ログ・データ・セットのカタログ・オプション (CATALOG) が YES に設定されているかどうか。
- BLKSIZE に対して適切な値が使用されているかどうか。

テープに保存する場合、IBM MQ は最大 20 ボリュームまで拡張できます。

推奨設定値:

テスト・システム	DASD
実動システム	TAPE

アーカイブ・ログの場所の選択について詳しくは、 ログ・アーカイブ・ストレージの計画を参照してください。

UNIT2
アーカイブ・ログ・データ・セットの 2 番目のコピーの保管に使用する装置の装置タイプまたは装置名を指定します。

装置タイプまたは装置名は 1 ～ 8 文字の英数字で指定します。 最初の文字は英字でなければなりません。 このパラメーターがブランクの場合は、UNIT パラメーターの値が使用されます。

デフォルトはブランクです。

関連資料
938 ページの『CSQ6SYSP の使用』
このトピックは、CSQ6SYSP を使用したシステム・パラメーターの設定方法のリファレンスとして使用してください。

947 ページの『CSQ6LOGP の使用』
このトピックは、CSQ6LOGP を使用したロギング・オプションの指定方法の参照として使用します。

z/OS *CSQ6USGP* の使用
このトピックは、CSQ6USGP を使用したシステム・パラメーターの設定方法のリファレンスとして使用してください。

CSQ6USGP を使用して、製品の使用記録を制御します。

CSQ6USGP のデフォルトのパラメーターは、959 ページの表 79 に示すとおりです。 これらの値を変更する必要がある場合は、該当するパラメーターの詳細記述を参照してください。

重要 : SET SYSTEM コマンドを使用してこれらのパラメーターを変更することはできません。

表 79. CSQ6USGP パラメーターのデフォルト値		
パラメーター	**説明**	**デフォルト値**
QMGRPROD	キュー・マネージャーの使用が記録される製品	ブランク
AMSPROD	Advanced Message Security (AMS) の使用が記録される製品	ブランク

QMGRPROD
キュー・マネージャーの使用が記録される製品を指定します。

以下のいずれかを指定します。

MQ
キュー・マネージャーの使用がスタンドアロン IBM MQ for z/OS 製品として、製品 ID 5655-MQ9 で記録されます。

VUE
キュー・マネージャーの使用がスタンドアロン IBM MQ for z/OS Value Unit Edition (VUE) 製品として、製品 ID 5655-VU9 で記録されます。

ADVANCEDVUE
キュー・マネージャーの使用が IBM MQ Advanced for z/OS Value Unit Edition 製品の一部として、製品 ID 5655-AV1 で記録されます。

AMSPROD
このパラメーターを設定しないと、AMS アドレス・スペースが開始されず、メッセージ CSQY024I が出力されます。

Advanced Message Security の使用が記録される製品を指定します (使用される場合)。

以下のいずれかを指定します。

アームズ
AMS の使用がスタンドアロン Advanced Message Security for z/OS 製品として、製品 ID 5655-AM9 で記録されます。

ADVANCED
AMS の使用が IBM MQ Advanced for z/OS 製品の一部として、製品 ID 5655-AV9 で記録されます。

ADVANCEDVUE
AMS の使用が IBM MQ Advanced for z/OS Value Unit Edition 製品の一部として、製品 ID 5655-AV1 で記録されます。

製品使用記録の詳細については、製品情報のレポートを参照してください。

関連資料
938 ページの『CSQ6SYSP の使用』
このトピックは、CSQ6SYSP を使用したシステム・パラメーターの設定方法のリファレンスとして使用してください。

947 ページの『CSQ6LOGP の使用』
このトピックは、CSQ6LOGP を使用したロギング・オプションの指定方法の参照として使用します。

z/OS チャネル・イニシエーター・パラメーターを調整する

ALTER QMGR を使用して、要件に適合するようにチャネル・イニシエーターをカスタマイズします。

- この作業は、必要に応じて *IBM MQ* キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行する必要があります。

多数のキュー・マネージャー属性によって、分散キューイングの操作方法が制御されます。 MQSC コマンド ALTER QMGR を使用して、これらの属性を設定してください。 初期設定データ・セットのサンプル thlqual.SCSQPROC(CSQ4INYG) には、カスタマイズできる設定が含まれています。 詳しくは、ALTER QMGR を参照してください。

これらのパラメーターの値は、チャネル・イニシエーターを開始するたびに一連のメッセージとして表示されます。

アダプター、ディスパッチャー、およびチャネルの最大数の間の関係

ALTER QMGR パラメーターの CHIADAPS および CHIDISPS は、チャネル・イニシエーターによって使用されるタスク制御ブロック (TCB) の数を定義します。 CHIADAPS (アダプター) TCB は、キュー・マネージャーに対して IBM MQ API 呼び出しを行うために使用されます。 CHIDISPS (ディスパッチャー) TCB は、通信ネットワークに対して呼び出しを行うために使用されます。

ALTER QMGR パラメーターの MAXCHL は、ディスパッチャー TCB へのチャネルの配分に影響を与えます。

CHIDISPS

チャネルが少数の場合には、デフォルト値を使用してください。

各プロセッサーにつきタスクを 1 つにすると、システム・パフォーマンスが最適化されます。 ディスパッチャー・タスクでは CPU の使用量が高いため、原則として、ビジーになるタスクをできるだけ少ない状態にします。そうすることで、スレッドを見つけて開始するまでにかかる時間を短縮します。

チャネル数が 100 を超えるシステムでは、CHIDISPS(20) が適しています。 必要以上のディスパッチャー TCB がある場合に CHIDISPS(20) を使用しても、特に問題が生じることはありません。

指針として、1000 を超える数のチャネルがある場合、50 の現行チャネルごとに 1 つのディスパッチャーを配分してください。 例えば、2000 までのアクティブ・チャネルを処理するには、CHIDISPS(40) を指定します。

TCP/IP を使用している場合、CHIDISPS に 100 よりも大きい値を指定した場合でも、TCP/IP チャネルに使用されるディスパッチャーの最大数は 100 となります。

CHIADAPS

キュー・マネージャーへの各 IBM MQ API 呼び出しは、他のいずれとも独立していて、任意のアダプター TCB に対して行うことができます。 持続メッセージを使用する呼び出しは、ログ入出力のために非持続メッセージの場合よりも時間がかかることがあります。 したがって、多くのチャネルで多数の持続メッセージを処理するチャネル・イニシエーターは、最適なパフォーマンスを得るために、デフォルトの 8 個のアダプター TCB を必要とする場合があります。 特にそうなるのは、作成されたバッチ・サイズが小さい場合 (各バッチ処理の最後にもログ I/O が必要となるため)、およびシン・クライアント・チャネルが使用される場合です。

実稼働環境のための推奨値は、CHIADAPS(30) です。 これより大きな値を指定しても特に利点はなく、必要以上のアダプター TCB がある場合に CHIADAPS(30) を指定しても特に問題が生じることはありません。

MAXCHL

各チャネルはチャネル開始時に特定のディスパッチャー TCB に関連付けられて、 そのチャネルが停止するまでその TCB に関連付けられたままとなります。 多数のチャネルで、それぞれの TCB を共有できます。 MAXCHL を使用すると、使用可能なディスパッチャー TCB にチャネルを展開します。 開始する最初の (MIN((MAXCHL / CHIDISPS) , 10)) チャネルは最初のディスパッチャー TCB に関連付けられて、すべてのディスパッチャー TCB が使用されるまで同様に関連付けられます。

このため、チャネルが少数で、MAXCHL が大きい値の場合には、チャネルはディスパッチャーに対して均等に配分されません。 例えば、CHIDISPS(10) を設定して MAXCHL がデフォルト値の 200 のままの場合に、チャネル数が 50 だけであれば、5 つのディスパッチャーはそれぞれ 10 のチャネルと関連付けられて、5 つのディスパッチャーは使用されません。 実際に使用するチャネルの数が小さな固定数のときには、MAXCHL をその数に設定することをお勧めします。

このキュー・マネージャー・プロパティーを変更する場合は、確実に値が互換であるようにするため、ACTCHL、LU62CHL、および TCPCHL キュー・マネージャー・プロパティーも検討しなければなりません。 これらのプロパティーの完全な説明とその関係については、キュー・マネージャーのパラメーターを参照してください。

チャネル・イニシエーター用の z/OS UNIX System Services 環境のセットアップ

チャネル・イニシエーター (CHINIT) は OMVS スレッドを使用します。 OMVS 構成パラメーターを検討してから、新規 CHINIT を作成するか、ディスパッチャーまたは SSLTASKS の数を変更します。

各 CHINIT は、3 + CHIDISP + SSLTASKS OMVS スレッドを使用します。 これらは、LPAR で使用される OMVS スレッドの総数、CHINIT 開始タスク・ユーザー ID により使用されるスレッドの数に入ります。

D OMVS,L を使用して、現行使用量、最高使用量、MAXPROCSYS のシステムしきい値 (システムが許容するプロセス最大数) を検討できます。

新規 CHINIT を追加するか CHIDISPS または SSLTASKS の値を増やした場合、スレッドの増加を計算して、MAXPROCSYS 値への影響を検討しなければなりません。 **SETOMVS** コマンドを使用して、直接、MAXPROCSYS の変更、または BPXPRCxx parmlib 値の更新 (またはその両方) を行えます。

OMVS パラメーター MAXPROCUSER は、単一 OMVS ユーザー、つまり同じ UID が所有できる OMVS スレッドの数です。 スレッドは、この値に数えられます。 それで、同じ開始タスク・ユーザー ID の 2 つの CHINITS に、それぞれ 10 のディスパッチャーと 3 つの SSLTASKS がある場合、その OMVS uid には 2 *(3 +10 + 3) = 32 スレッドがあることになります。

デフォルト MAXPROCUSER は **D OMVS,O** コマンドを発行して表示できますし、**SETOMVS** コマンドを使用して、直接、MAXPROCUSER の変更、または BPXPRCxx parmlib 値の更新 (またはその両方) を行えます。

この値は、 RACF コマンド **ALTUSER userid OMVS(PROCUSERMAX(nnnn))** または同等のコマンドを使用して、ユーザーごとにオーバーライドできます。

チャネル・イニシエーターを開始するには、次のコマンドを実行します。

```
START CHINIT
```

チャネル・イニシエーターが正常に開始したことを確認するために、xxxxCHIN(ssidCHIN) ジョブ・ログに ICH408I エラーがないことを確認します。

関連概念

961 ページの『バッチ、TSO、および RRS アダプターをセットアップする』
ライブラリーを該当する STEPLIB 連結に追加して、アダプターをアプリケーションで使用できるようにします。 アダプターによって発行されるスナップ・ダンプに対応するには、CSQSNAP DD 名を割り振ります。 CSQBDEFV を使用してアプリケーション・プログラムの移植性を向上させることを検討してください。

関連資料

チャネル・イニシエーター統計データ・レコード

z/OS バッチ、*TSO*、および *RRS* アダプターをセットアップする

ライブラリーを該当する STEPLIB 連結に追加して、アダプターをアプリケーションで使用できるようにします。 アダプターによって発行されるスナップ・ダンプに対応するには、CSQSNAP DD 名を割り振ります。 CSQBDEFV を使用してアプリケーション・プログラムの移植性を向上させることを検討してください。

- この作業は、必要に応じて *IBM MQ* キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。

バッチ接続を使用するバッチおよびその他のアプリケーションでアダプターを使用できるようにするには、バッチ・アプリケーションの STEPLIB 連結に、次の IBM MQ ライブラリーを追加します。

- thlqual.SCSQANL*x*
- thlqual.SCSQAUTH

この場合、*x* は各国語の言語を示す文字です。 (ライブラリーが LPA またはリンク・リストにある場合は、これを行う必要はありません。)

TSO アプリケーションの場合は、TSO ログオン・プロシージャー内でライブラリーを STEPLIB 連結に 追加するか、TSO コマンドの TSOLIB を使用してライブラリーをアクティブ化します。

アダプターは、予期しない IBM MQ エラーを検出すると、DD 名 CSQSNAP に z/OS SNAP ダンプを発行し、アプリケーションに理由コード MQRC_UNEXPECTED_ERROR を発行します。CSQSNAP DD ステートメントがアプリケーション JCL にないか、または CSQSNAP が TSO の下のデータ・セットに割り振られていない場合は、ダンプはとられません。 この場合、アプリケーション JCL に CSQSNAP DD ステートメントを含めるか、または CSQSNAP を TSO の下のデータ・セットに割り振ってから、アプリケーションを再実行することができます。 しかし、いくつかの問題が断続的に発生することがあるため、CSQSNAP ステートメントをアプリケーション JCL に含めるか、または CSQSNAP を TSO ログオン・プロシージャーでデータ・セットに割り振って、問題の発生時に障害の理由を見極めることをお勧めします。

提供されているプログラム CSQBDEFV により、使用しているアプリケーション・プログラムの移植性が向上します。 CSQBDEFV では、接続先のキュー・マネージャーまたはキュー共有グループの名前を指定でき、その名前をアプリケーション・プログラムの MQCONN または MQCONNX 呼び出しの中で指定せずに済みます。 キュー・マネージャーまたはキュー共有グループごとに、新しいバージョンの CSQBDEFV を作成できます。 そのためには、次の手順を実行してください。

1. IBM MQ アセンブラー・プログラム CSQBDEFV を、thlqual.SCSQASMS からユーザー・ライブラリーへコピーします。
2. 提供されたプログラムには、デフォルトのサブシステム名 CSQ1 が設定されます。 テストおよびインストール検査では、この名前をそのまま使用できます。 実動サブシステムの場合は、NAME=CSQ1 を 1 ～ 4 文字の独自のサブシステム名に変更することも、CSQ1 を そのまま使用することもできます。

   キュー共有グループを使用する場合、CSQ1 ではなくキュー共有グループ名を指定することができます。これを行う場合、プログラムはそのグループ内の活動状態のキュー・マネージャーに接続要求を出します。
3. プログラムのアセンブルおよびリンク・エディットを実行し、CSQBDEFV ロード・モジュールを生成します。 アセンブルするときには、ライブラリー thlqual.SCSQMACS を SYSLIB 連結に組み込みます。その場合は、リンク・エディット・パラメーター RENT,AMODE=31,RMODE=ANY を使用してください。 これは thlqual.SCSQPROC(CSQ4DEFV) のサンプル JCL に示されています。 次に、ロード・ライブラリーを thlqual.SCSQAUTH より前に z/OS バッチ、TSO STEPLIB に組み込みます。

関連概念

962 ページの『操作パネルおよび制御パネルをセットアップする』
操作パネルおよび制御パネルをセットアップするには、まず、必要なパネル、EXEC、メッセージ、および表が入ったライブラリーをセットアップする必要があります。 そのためには、パネルに使用する各国語機能を考慮する必要があります。 これを行った後、オプションで IBM MQ 操作パネルおよび制御パネル用のメイン ISPF メニューを更新し、ファンクション・キーの設定を変更することができます。

z/OS 操作パネルおよび制御パネルをセットアップする

操作パネルおよび制御パネルをセットアップするには、まず、必要なパネル、EXEC、メッセージ、および表が入ったライブラリーをセットアップする必要があります。 そのためには、パネルに使用する各国語機能を考慮する必要があります。 これを行った後、オプションで IBM MQ 操作パネルおよび制御パネル用のメイン ISPF メニューを更新し、ファンクション・キーの設定を変更することができます。

- このタスクは、 *IBM MQ* を実行する *z/OS* システムごとに *1* 回実行する必要があります。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。

ライブラリーのセットアップ

以下のステップを実行して、IBM MQ の操作パネルと制御パネルをセットアップします。

1. 連結に含まれるすべてのライブラリーは、 同じ形式 (F、FB、V、VB) であり、かつ同じブロック・サイズであるか、降順ブロック・サイズの順番のどちらかにしてください。 そのようにしないと、これらのパネルを使用しようとしたときに問題が発生することがあります。
2. ライブラリー thlqual.SCSQEXEC を SYSEXEC 連結または SYSPROC 連結に組み込むか、 あるいは、TSO ALTLIB コマンドを使用してライブラリーをアクティブ化します。 このライブラリーは、インストール時に固定ブロック 80 のレコード形式で割り振られており、必要な EXEC が設定されています。

   ライブラリーは SYSEXEC 連結に入れることをお勧めします。 ただし、ライブラリーを SYSPROC 連結に入れる場合は、ライブラリーは 80 バイトのレコード長でなければなりません。

3. thlqual.SCSQAUTH がリンク・リストまたは LPA に入っていない場合は、thlqual.SCSQAUTH および thlqual.SCSQANLx を TSO ログオン・プロシージャー STEPLIB に追加するか、TSO TSOLIB コマンドを使ってアクティブ化します。
4. IBM MQ パネル・ライブラリーを永続的に ISPF ライブラリー・セットアップに追加することも、パネルの使用時に動的にセットアップされるようにすることもできます。 永続的にインストールするには、次の手順を実行する必要があります。

   a. 操作パネルと制御パネルの定義が入っているライブラリーを ISPPLIB 連結に組み込みます。 この名前は thlqual.SCSQPNLx です。この場合、x は各国語の言語を示す文字です。

   b. 必要な表が入っているライブラリーを ISPTLIB 連結に組み込みます。 この名前は thlqual.SCSQTBLx です。 この場合、x は各国語の言語を示す文字です。

   c. 必要なメッセージが入っているライブラリーを ISPMLIB 連結に組み込みます。 この名前は thlqual.SCSQMSGx です。この場合、x は各国語の言語を示す文字です。

   d. 必要なロード・モジュールが入っているライブラリーを ISPLLIB 連結に組み込みます。 このライブラリーの名前は thlqual.SCSQAUTH です。

   後者の選択項目の場合は、z/OS LIBDEF コマンドを使用します。 使用できるさまざまなキーワードへのリンクについては、 例 を参照してください。
5. TSO コマンド処理プログラムのパネルから IBM MQ パネルへのアクセスが可能であるかどうかを検査します。 これは通常、ISPF/PDF 基本オプション・メニューのオプション 6 です。 実行する EXEC の名前は CSQOREXX です。ステップ 4 のように、IBM MQ ライブラリーを ISPF セットアップに永続的に入れたかどうかを指定するパラメーターはありません。 永続的にインストールしていない場合は、次のパラメーターを指定します。

   ```
   CSQOREXX thlqual langletter
   ```

   この場合、`langletter` は、使用する各国語を示す文字です。

   C
   中国語 (簡体字)

   E
   U.S. 英語(大/小文字混合)

   F
   フランス語

   K
   日本語

   U
   U.S. 英語 (大文字)

ISPF メニューの更新

ISPF メインメニューを更新すると、ISPF から IBM MQ 操作パネルおよび制御パネルにアクセスすることができます。 &ZSEL に必要な設定は、次のとおりです。

```
CMD(%CSQOREXX thlqual langletter)
```

`thlqual` および `langletter` については、ステップ 963 ページの『5』 を参照してください。

詳しくは、「*z/OS: ISPF Dialog Developer 's Guide and Reference*」を参照してください。

ファンクション・キーおよびコマンド設定値の更新

通常の ISPF プロシージャーを使用して、パネルで使用されるファンクション・キーおよびコマンド設定値を変更することができます。 アプリケーション ID は CSQO です。

ただし、作成した変更内容を反映するようにヘルプ情報が更新されるわけではないので、この変更はお勧めしません。

関連概念

964 ページの『IBM MQ ダンプ・フォーマット・メンバーを組み込む』
対話式問題管理システム (IPCS) を使用して IBM MQ ダンプをフォーマットできるようにするには、いくつかのシステム・ライブラリーを更新する必要があります。

z/OS *IBM MQ* ダンプ・フォーマット・メンバーを組み込む

対話式問題管理システム (IPCS) を使用して IBM MQ ダンプをフォーマットできるようにするには、いくつかのシステム・ライブラリーを更新する必要があります。

- このタスクは、*IBM MQ* を実行する *z/OS* システムごとに *1* 回実行する必要があります。
- 旧バージョンから移行する際に、この作業を実行する必要があります。

対話式問題管理システム (IPCS) を使用して IBM MQ ダンプをフォーマットできるようにするには、データ・セット thlqual.SCSQPROC(CSQ7IPCS) を SYS1.PARMLIB にコピーします。 このデータ・セットを編集する必要はありません。

IPCS 用の TSO プロシージャーをカスタマイズした場合は、thlqual.SCSQPROC(CSQ7IPCS) を IPCSPARM 定義内の任意のライブラリーにコピーすることができます。 詳しくは、「*z/OS MVS IPCS* ユーザーズ・ガイド」を参照してください。

ライブラリー thlqual.SCSQPNLA も ISPPLIB 連結に取り込む必要があります。

ダンプ・フォーマット・プログラムを TSO セッションまたは IPCS ジョブで使用できるようにするには、ライブラリー thlqual.SCSQAUTH も STEPLIB 連結に組み込むか、 TSO TSOLIB コマンドを使用して thlqual.SCSQAUTH を アクティブ化する必要があります (リンク・リストまたは LPA に既に存在する場合)。

関連概念

964 ページの『通知メッセージを抑止する』
IBM MQ システムでは、多数の通知メッセージが生成されることがあります。 選択したメッセージがコンソールまたはハードコピー・ログに送信されないようにすることができます。

z/OS 通知メッセージを抑止する

IBM MQ システムでは、多数の通知メッセージが生成されることがあります。 選択したメッセージがコンソールまたはハードコピー・ログに送信されないようにすることができます。

- このタスクは、*IBM MQ* を実行する *z/OS* システムごとに *1* 回実行する必要があります。
- 旧バージョンから移行する際は、この作業を実行する必要はありません。

IBM MQ システムが大量に使用されている場合、多数のチャネルが停止して開始している場合は、多数の情報メッセージが z/OS コンソールおよびハードコピー・ログに送信されます。 IBM MQ-IMS ブリッジとバッファー管理プログラムも、大量の通知メッセージを生成します。

必要な場合は、SYS1.PARMLIB の MPFLSTxx メンバーで指定されている z/OS メッセージ処理機能リストを使用して、これらのコンソール・メッセージの一部を抑止することができます。 指定したメッセージは、ハードコピー・ログには書き込まれますが、コンソールには表示されません。

サンプル `thlqual.SCSQPROC(CSQ4MPFL)` は、MPFLSTxx の推奨設定を示しています。 詳しくは、MPFLSTxx (メッセージ処理機能リスト) を参照してください。

ハードコピー・ログの選択した通知メッセージを抑止する場合は、z/OS インストール・システム出口 IEAVMXIT を使用することができます。 必要なメッセージに対して次のビット・スイッチを ON に設定できます。

CTXTRDTM
メッセージを削除します。

このメッセージはコンソールに表示されたり、ハードコピーに記録されることがありません。

CTXTESJL
ジョブ・ログから抑止します。

このメッセージは JES ジョブ・ログに入れられません。

CTXTNWTP
WTP 処理を実行しません。

メッセージは TSO 端末またはバッチ・ジョブのシステム・メッセージ・データ・セットに送られません。

注 :

1. その他のパラメーターについて詳しくは、 MVS インストール・システム出口を参照してください。
2. 提案されている抑止リスト CSQ4MPFL にあるもの以外のメッセージを抑止することは勧められていません。

さらに、パラメーターを追加指定することもできます。

EXCLMSG
すべてのログから除外するメッセージのリストを指定します。

このリストにあるメッセージは、z/OS コンソールおよびハードコピー・ログに送られません。 詳しくは、938 ページの『CSQ6SYSP の使用』の EXCLMSG を参照してください。

関連タスク
980 ページの『z/OS でのキュー・マネージャーのテスト』
キュー・マネージャーのカスタマイズまたは移行を済ませた後、IBM MQ for z/OS に添付されているインストール検査プログラムおよびサンプル・アプリケーションを実行することによって、キュー・マネージャーをテストできます。

z/OS キュー共有グループの構成

高可用性のために共有キューを使用する場合は、キュー共有グループを構成するためのステップバイステップ・ガイドとしてこれらのトピックをご利用ください。

IBM MQ for z/OS システムのセットアップ処理のこの部分のステップを完了した後、936 ページの『システム・パラメーター・モジュールの調整』を行い、キュー共有グループ・データを追加する必要があります。CSQ6SYSP を変更して QSGDATA パラメーターを指定する必要があります。

z/OS *Db2* 環境のセットアップ

キュー共有グループを使用する場合は、いくつかのサンプル・ジョブをカスタマイズして実行することで、必要な Db2 オブジェクトを作成する必要があります。

Db2 環境のセットアップ

いくつかのサンプル・ジョブをカスタマイズして実行することで、必要な Db2 オブジェクトを作成し、バインドする必要があります。

- この作業は、Db2 データ共有グループごとに繰り返してください。
- 旧バージョンから移行する際に、`bind` および `grant` のステップを実行する必要があります。
- キュー共有グループを使用しない場合は、この作業を省略してください。

  後にキュー共有グループを使用する必要が生じた場合は、その時点でこのタスクを実行してください。

IBM MQ には、同等のジョブ・セットが 2 つあります。 接頭部が CSQ45 のジョブは、旧バージョンの IBM MQ との互換性を保持し、Db2 バージョン 11 以前で使用するために提供されています。 Db2 V12 またはそれ以降で新しいデータ共有グループをセットアップする場合は、CSQ4X 接頭辞を持つジョブを使用することを推奨します。これらのジョブは、ダイナミック・サイジングとユニバーサル・テーブル・スペース（UTS）のための、より新しい 'Db2 機能を利用するからです。

次の手順は、新規 Db2 データ共有グループごとに実行する必要があります。 すべてのサンプル JCL は thlqual.SCSQPROC にあります。

1. サンプル JCL CSQ4XCSG を、IBM MQ データベース、表スペース、または表に使用されるストレージ・グループを作成するようにカスタマイズして、それを実行します。
2. サンプル JCL CSQ4XCDB を、この Db2 データ共有グループに接続するすべてのキュー・マネージャーによって使用されるデータベースを作成するようにカスタマイズして、それを実行します。
3. サンプル JCL CSQ4XCTS を、キュー共有グループ用に使用されるキュー・マネージャーおよびチャネル・イニシエーター表を含む表スペースを作成するようにカスタマイズして、それを実行します。
4. サンプル JCL CSQ4XCTB を、15 個の Db2 表と関連索引を作成するようにカスタマイズして、それを実行します。 行名または属性は変更しないでください。
5. サンプル JCL CSQ45BPL を、キュー・マネージャー、ユーティリティー、およびチャネル・イニシエーター用の Db2 計画をバインドするようにカスタマイズして、それを実行します。
6. サンプル JCL CSQ45GEX を、キュー・マネージャー、ユーティリティー、およびチャネル・イニシエーターによって使用されるユーザー ID 用の計画に実行権限を認可するようにカスタマイズして、それを実行します。 キュー・マネージャーおよびチャネル・イニシエーター用のユーザー ID は、開始された作業プロシージャーが実行しているユーザー ID です。 ユーティリティーのユーザー ID は、バッチ・ジョブが実行依頼されるユーザー ID です。

   該当するプランの名前を以下の表に示します。

ユーザー	計画 (LTS)	計画 (CD)
キュー・マネージャー	CSQ5A 930、 CSQ5C 930、 CSQ5D 930、 CSQ5K 930、 CSQ5L 930、 CSQ5M 930、 CSQ5P 930、 CSQ5R 930、 CSQ5S 930、 CSQ5T 930、 CSQ5U 930、 CSQ5W 930	CSQ5A 9X0, CSQ5C 9X0, CSQ5D 9X0, CSQ5K 9X0, CSQ5L 9X0, CSQ5M 9X0, CSQ5P 9X0, CSQ5R 9X0, CSQ5S 9X0, CSQ5T 9X0, CSQ5U 9X0, CSQ5W 9X0
CSQUTIL バッチ・ユーティリティー用の SDEFS 機能	CSQ52 930	CSQ52 9X0
CSQ5PQSG および CSQJUCNV バッチ・ユーティリティー	CSQ5B 930	CSQ5B 9X0
CSQUZAP サービス・ユーティリティー	CSQ5Z 930	CSQ5Z 9X0

Db2 のセットアップ中に障害が発生する場合、以下のジョブをカスタマイズして、実行することができます。

- CSQ45DTB。表および索引を除去します。
- CSQ4XDTS。表スペースを除去します。
- CSQ4XDDB。データベースを除去します。
- CSQ4XDSG。ストレージ・グループを除去します。

注 : Db2 のロックの問題によってこれらのジョブが失敗する場合は、Db2 リソースの競合が原因である可能性があります。特に、システムの負荷が大きい場合は、その可能性が高くなります。 ジョブを後から再実行依頼してください。 システムの負荷を軽減するか、またはシステムが静止状態になったときに、これらのジョブを実行することをお勧めします。

Db2 のセットアップについて詳しくは、 *Db2 For z/OS 12.0.0* の「Db2 Administration」を参照してください。

Db2 表サイズについては、 計画中 z/OS を参照してください。

関連概念

967 ページの『カップリング・ファシリティーのセットアップ』
キュー共有グループを使用する場合、キュー共有グループ (QSG) 内のキュー・マネージャーが使用するカップリング・ファシリティー構造を、IXCMIAPU を使用して、カップリング・ファシリティー・リソース管理 (CFRM) ポリシー・データ・セットに定義します。

z/OS カップリング・ファシリティーのセットアップ

キュー共有グループを使用する場合、キュー共有グループ (QSG) 内のキュー・マネージャーが使用するカップリング・ファシリティー構造を、IXCMIAPU を使用して、カップリング・ファシリティー・リソース管理 (CFRM) ポリシー・データ・セットに定義します。

IXCMIAPU について詳しくは、 管理データ・ユーティリティー を参照してください。

- この作業は、キュー共用グループごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。
- キュー共有グループを使用しない場合は、この作業を省略してください。

  後にキュー共有グループを使用する必要が生じた場合は、その時点でこのタスクを実行してください。

キュー共有グループのすべての構造は、キュー共有グループの名前で始まります。 以下の構造を定義します。

- *qsg-name*CSQ_ADMIN と呼ばれる管理構造。 この構造は IBM MQ そのものに使用され、ユーザー・データは含まれません。
- *qsg-name* CSQSYSAPPL と呼ばれるシステム・アプリケーション構造。 この構造は IBM MQ システム・キューが状態情報を保管するために使用されます。
- 共有キューのメッセージを保持するために使用する 1 つ以上の構造。 これには、長さが 16 文字までの任意の名前を付けることができます。
  - 最初の 4 文字はキュー共有グループ名でなければなりません。 (キュー共有グループ名が 4 文字未満の場合は、4 文字になるまで @ 記号を埋め込む必要があります。)
  - 5 番目の文字は英字で、その後の文字は英字または数字でなければなりません。 名前のこの部分 (キュー共有グループ名以外) は、共有キューまたは CF 構造オブジェクトを定義する際に CFSTRUCT 名に指定するものです。

  共有キューのメッセージを保持するために使用する構造名には英字および数字だけを使用できます。その他の文字 (例えば、_ 文字。これは管理構造の名前で使用されます) は使用できません。

IXCMIAPU 用のサンプル制御ステートメントがデータ・セット thlqual.SCSQPROC(CSQ4CFRM) に入っています。 これらをカスタマイズしてから、カップリング・ファシリティー用の IXCMIAPU ジョブに追加し、それを実行してください。

構造の定義が正常に終了したら、使用する予定の CFRM ポリシーをアクティブ化します。 そのためには、次の z/OS コマンドを発行します。

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME= policy-name
```

CF ストラクチャーとサイズの計画については、カップリング・ファシリティー資源の定義を参照してください。

関連概念

928 ページの『ESM セキュリティー管理を実施する』
キュー・マネージャーとチャネル・イニシエーターのセキュリティー管理を実装します。

z/OS *SMDS* 環境をセットアップする

SMDS を使用して共有キューのメッセージをオフロードする場合は、SMDS オフロード・ストレージ環境をセットアップします。

- *SMDS* にデータをオフロードするように構成するキュー共有グループのキュー・マネージャーおよび構造ごとにこの作業を実行します。
- *SMDS* にデータをオフロードするための追加の構造を後で構成する場合は、その時点で、再度この作業を実行できます。
- キュー共有グループを使用しない場合は、この作業を省略してください。

  後にキュー共有グループを使用する必要が生じた場合は、その時点でこのタスクを実行してください。

SMDS 環境をセットアップする

1. 構造とデータ・セット・スペース所要量を見積もります。 共有メッセージ・データ・セット容量の考慮事項を参照してください。
2. データ・セットを割り振り、事前フォーマットします。 共有メッセージ・データ・セットの作成を参照してください。
3. IBM MQ に CF 構造を定義する場合、必ず CFLEVEL(5) および OFFLOAD(SMDS) を使用して CFSTRUCT を定義します。

関連概念

967 ページの『カップリング・ファシリティーのセットアップ』
キュー共有グループを使用する場合、キュー共有グループ (QSG) 内のキュー・マネージャーが使用するカップリング・ファシリティー構造を、IXCMIAPU を使用して、カップリング・ファシリティー・リソース管理 (CFRM) ポリシー・データ・セットに定義します。

z/OS *IBM MQ* 項目を *Db2* 表に追加する

キュー共有グループを使用している場合は、CSQ5PQSG ユーティリティーを実行して、Db2 データ共有グループの IBM MQ 表にキュー共有グループとキュー・マネージャーの項目を追加します。

- この作業は、*IBM MQ* キュー共有グループごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。
- キュー共有グループを使用しない場合は、この作業を省略してください。

  後にキュー共有グループを使用する必要が生じた場合は、その時点でこのタスクを実行してください。

キュー共有グループごと、およびキュー共有グループのメンバーになる各キュー・マネージャーごとに、CSQ5PQSG を実行します。

以下のアクションを指定された順序で実行します。

1. CSQ5PQSG プログラムの ADD QSG 機能を使用して、キュー共有グループの項目を IBM MQ Db2 表に追加します。 thlqual.SCSQPROC(CSQ45AQS) にサンプルが用意されています。

   Db2 データ共有グループに定義されているそれぞれのキュー共有グループごとに、この機能を 1 回実行します。 キュー共有グループの項目は、そのキュー共有グループを参照するキュー・マネージャーの項目を追加する前に、存在していなければなりません。
2. CSQ5PQSG プログラムの ADD QMGR 機能を使用して、キュー・マネージャーの項目を IBM MQ Db2 表に追加します。 thlqual.SCSQPROC(CSQ45AQM) にサンプルが用意されています。

   キュー共有グループのメンバーになるそれぞれのキュー・マネージャーごとに、この機能を実行します。

   注:

   a. 1 つのキュー・マネージャーに対して、 それをメンバーにできるのは 1 つのキュー共有グループだけです。

   b. キュー共有グループを使用できるようにするには、RRS が実行されていなければなりません。

関連概念

936 ページの『システム・パラメーター・モジュールの調整』
IBM MQ システム・パラメーター・モジュールは、IBM MQ がその処理で使用するロギング、保存、トレース、および接続の環境を制御します。 デフォルトのモジュールが提供されます。 いくつかのパラメーター

(データ・セット名など) は通常、サイト固有であるため、独自のシステム・パラメーター・モジュールを作成する必要があります。

z/OS キュー共有グループの *ESM* セキュリティー管理の実装

キュー共有グループ内のすべてのキュー・マネージャーが Db2 とカップリング・ファシリティー・リスト構造にアクセスするためのセキュリティー管理を実装します。

- キュー共有グループ内の *IBM MQ* キュー・マネージャーごとにこの作業を繰り返してください。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。

キュー・マネージャー、チャネル・イニシエーター、およびユーティリティーに関連したユーザー ID が、接続を確立する各 Db2 サブシステムに対して RRSAF 接続を確立する権限を持っていることを確認してください。 キュー・マネージャーおよびチャネル・イニシエーター用のユーザー ID は、開始された作業プロシージャーが実行しているユーザー ID です。

ユーティリティーのユーザー ID は、バッチ・ジョブが実行依頼されるユーザー ID です。 ユーザー ID が READ アクセスを必要とする RACF プロファイルは、DSNR 資源クラスの `Db2ssid.RRSAF` です。

キュー共有グループ内の各キュー・マネージャーに関連付けられているユーザー ID に、カップリング・ファシリティー・リスト構造に対する適切なアクセス・レベルを付与する必要があります。 RACF クラスは FACILITY です。

以下のユーザー ID には ALTER アクセス権が必要です。

- `IXLSTR.structure-name` プロファイルに対するキュー・マネージャー ID
- CSQ5PQSG を実行するユーザー ID

関連概念

928 ページの『ESM セキュリティー管理を実施する』
キュー・マネージャーとチャネル・イニシエーターのセキュリティー管理を実装します。

z/OS Advanced Message Security for z/OS の構成

これらのトピックは、Advanced Message Security(AMS)を構成するためのステップ・ガイドとして使用してください。

始める前に

AMS の構成を開始する前に、以下のキュー・マネージャー構成ステップが実行されていることを確認してください。

1. 917 ページの『z/OS リンク・リストおよび LPA を更新する』で説明されているように、CSQ0DRTM モジュールを LPA に追加します。
2. 921 ページの『z/OS プログラム特性表を更新する』で説明されているように、 CSQ0DSRV のエントリーを z/OS プログラム・プロパティー・テーブル (PPT) に追加します。
3. 929 ページの『初期設定入力データ・セットをカスタマイズする』で説明されているように、CSQ4INSM メンバーをキュー・マネージャー開始タスク・プロシージャーの CSQINP2 連結に組み込みます。
4. AMSPROD 属性を使用して AMS を使用可能にする。 詳しくは、IBM MQ for z/OS 製品による製品の使用記録を参照してください。

次のタスク

AMS によって保護されるキューのポリシーを構成します。 セキュリティー・ポリシーについては、Advanced Message Security セキュリティー・ポリシーの管理を参照してください。

z/OS での構成例には、AMS 構成の例があります。

z/OS *Advanced Message Security* 用のプロシージャーを作成する

Advanced Message Security (AMS)を使用するように構成される各 IBM MQ サブシステムは、AMS アドレス・スペースを開始するためのカタログ式プロシージャーを必要とします。 独自のプロシージャー・ライブラリーを作成するか、あるいは IBM 提供のプロシージャー・ライブラリーを使用することができます。

手順

1. サンプルの開始タスク・プロシージャー *thlqual*.SCSQPROC(CSQ4AMSM) を SYS1.PROCLIB にコピーするか、SYS1.PROCLIB を使用していない場合はプロシージャー・ライブラリーにコピーします。 このプロシージャーに xxxxAMSM という名前を付けます。ここで、xxxx は IBM MQ サブシステムの名前です。例えば、CSQ1AMSM は、キュー・マネージャー CSQ1 の AMS 開始タスク・プロシージャーに相当します。
2. 使用する IBM MQ サブシステムごとに 1 つずつコピーを作成します。
3. サンプル・プロシージャー CSQ4AMSM 内の命令を使用して、要件に合うようにプロシージャーを調整します。 さらに、JCL で記号パラメーターを使用して、プロシージャーを開始時に変更できます。
4. Language Environment® の _CEE_ENVFILE ファイルを使用して AMS タスクに渡されるパラメーターを検討し、任意に変更します。 サンプル thlqual.SCSQPROC(CSQ40ENV) は、サポートされるパラメーターをリストしています。
5. IBM MQ キュー・マネージャーごとにステップ 1 から 4 を繰り返します。

次のタスク

970 ページの『Advanced Message Security 開始タスク・ユーザー ID をセットアップする』

z/OS *Advanced Message Security* 開始タスク・ユーザー *ID* をセットアップする

Advanced Message Security(AMS)タスクでは、ユーザー ID が z/OS UNIX System Services(z/OS UNIX)プロセスとして認識されるようにする必要があります。

このタスクについて

さらに、タスクが代行するユーザーは、 UNIX UID (ユーザー ID) および GID (グループ ID) の適切な定義も持っている必要があります。これらのユーザーは、 z/OS UNIX System Services ユーザーと呼ばれます。 z/OS UNIX System Services UID および GID の定義について詳しくは、「*z/OS: Security Server RACF* セキュリティー管理者のガイド」を参照してください。

「*z/OS UNIX System Services* 計画 」を参照して、従来の UNIX セキュリティーと z/OS UNIX セキュリティーの間のセキュリティーの違いを確実に理解してください。 これにより、特権 z/OS UNIX System Services プロセスをデプロイおよび実行するためのインストール・システムのセキュリティー・ポリシーに従って、Advanced Message Security タスクを管理することができます。

従来の UNIX セキュリティーと z/OS セキュリティーとの主な違いは、カーネル・サービスが、UNIX レベルと z/OSUNIX レベルの二のレベルの適切な特権をサポートしているという点です。

インストール済み環境のセキュリティー・ポリシーに応じて、Advanced Message Security タスクは、スーパーユーザー権限 (uid(0)) で実行するか、あるいは、RACF FACILITY クラスの BPX.DAEMON および BPX.SERVER プロファイルに対するアクセス権がある RACF ID で実行する必要があります (このタスクに、そのユーザーの RACF ID を付与できなければならないため)。

後者の方式が使用される場合、あるいは BPX.DAEMON または BPX.SERVER プロファイルを既にアクティブ化している場合には、Advanced Message Security タスク・プログラム (thlqual.SCSQAUTH(CSQ0DSRV)) が、RACF プログラム管理ライブラリーに配置されていなければなりません。

注 : このタスクのユーザー ID に関連付けられた鍵リングに Advanced Message Security の受信側証明書がロードされるため、このユーザー ID は注意深く選択してください。 この考慮事項については、z/OS での証明書の使用に説明されています。

以下の手順では、Advanced Message Security 開始タスク・ユーザーのセットアップ方法を示します。 この手順では、RACF コマンドを例として使用します。 別のセキュリティー・マネージャーを使用する場合には、同等のコマンドを使用する必要があります。

注 : このセクションの例は、RACF の STARTED、FACILITY、および SURROGAT クラスの総称プロファイル・コマンド処理と総称プロファイル検査を既にアクティブ化していることを前提としています。 RACF が総称プロファイルを処理する方法について詳しくは、「*z/OS: Security Server RACF コマンド言語解説書*」を参照してください。

手順

1. Advanced Message Security 開始タスク・ユーザーを RACF に定義します。 このセクションの例では、ユーザー ID WMQAMSM を使用します。

   ```
   ADDUSER WMQAMSM NAME('AMS user') OMVS (UID(0)) DFLTGRP(group)
   ```

   インストール済み環境の基準に応じて適切にデフォルトのグループを選択します。

   注 : z/OS UNIX スーパーユーザー権限 (UID(0)) を付与しない場合は、BPX.DAEMON および BPX.SERVER 機能クラス・プロファイルに対するアクセス権を Advanced Message Security ユーザー ID に付与する必要があります。

   ```
   PERMIT BPX.DAEMON CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
   ```

   Advanced Message Security タスク・プログラム(*thlqual*.SCSQAUTH(CSQ0DSRV))は、RACF プログラム制御ライブラリー内になければなりません。

   SCSQAUTH ライブラリーをプログラム管理下に置くには、以下のコマンドを使用します。

   ```
   RALTER PROGRAM * ADDMEM('thlqual.SCSQAUTH'//NOPADCHK) -or-
   RALTER PROGRAM ** ADDMEM('thlqual.SCSQAUTH'//NOPADCHK)
   SETROPTS WHEN(PROGRAM) REFRESH
   ```

   また、 Advanced Message Security タスクによって使用される各国語ライブラリー (*thlqual*.SCSQANLx) のプログラム制御も有効にする必要があります。
2. RACF STARTED クラスがアクティブであるかどうかを判別します。 アクティブでない場合は、RACF STARTED クラスをアクティブ化します。

   ```
   SETROPTS CLASSACT(STARTED)
   ```

3. Advanced Message Security タスクの開始クラス・プロファイルを定義し、ステップ 1 で選択または作成したユーザー ID を指定します:

   ```
   RDEFINE STARTED qmgrAMSM.* STDATA(USER(WMQAMSM))
   ```

   ここで、*qmgr* は、開始タスク名の接頭部です。 例えば、開始タスクの名前を CSQ1AMSM とします。この場合、*qmgr*AMSM.* を CSQ1AMSM.* に置き換えます。

   AMS 開始タスクの名前は *qmgr*AMSM にする必要があります。
4. **SETROPTS** RACF コマンドを使用して、ストレージ内 RACLIST STARTED クラス・プロファイルをリフレッシュします。

   ```
   SETROPTS RACLIST(STARTED) REFRESH
   ```

5. Advanced Message Security タスクは、IBM MQ メッセージの保護処理中に、要求側のホスト・ユーザー ID を一時的に想定します。 このため、要求を行うことができる各ユーザー ID の SURROGAT クラスにプロファイルを定義する必要があります。

   RACF SURROGAT クラスがアクティブである場合は、単一の総称プロファイルを定義すると、Advanced Message Security タスクで任意のユーザーの ID が付与されます。 SURROGAT クラスがアクティブでない場合には、検査が無視されます。 必要な SURROGAT プロファイルについては、「*z/OS UNIX System Services 計画*」で説明しています。

   SURROGAT クラスにプロファイルを定義するには、以下を行います。

   a) RACF SETROPTS コマンドを使用して RACF SURROGAT クラスをアクティブ化します。

```
SETROPTS CLASSACT(SURROGAT)
```

b) RACFSURROGAT クラスの総称プロファイル処理を活動化します:

```
SETROPTS GENERIC(SURROGAT)
```

c) RACF SURROGAT クラスの総称プロファイル・コマンド処理をアクティブ化します。

```
SETROPTS GENCMD(SURROGAT)
```

d) SURROGAT クラスに総称プロファイルを定義します。

```
RDEFINE SURROGAT BPX.SRV.* UACC(NONE)
```

e) 総称 SURROGAT クラス・プロファイルに対するアクセス権を Advanced Message Security ユーザー ID に付与します。

```
PERMIT BPX.SRV.* CLASS(SURROGAT) ID(WMQAMSM) ACCESS(READ)
```

注 : 「*z/OS UNIX System Services* 計画」で説明されているように、特定のユーザーが Advanced Message Security タスクによって処理されるように制限する場合は、より具体的なプロファイルを定義できます。

例えば、BPX.SRV.MQUSER1 というプロファイルは、AMS タスクでユーザー ID MQUSER1 の ID を付与できるかどうかを制御します。

f) Advanced Message Security ユーザー ID を BPX.SERVER 機能に対して許可します(証明書および鍵リングの作成でまだ実行されていない場合):

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
```

g) **SETROPTS** RACF コマンドを使用して、ストレージ内 RACLIST 開始クラス・プロファイルをリフレッシュします。

```
SETROPTS RACLIST(SURROGAT) REFRESH
SETROPTS RACLIST(FACILITY) REFRESH
```

6. Advanced Message Security タスクでは、z/OS System SSL サービスによって提供される機能を使用して、SAF 管理の鍵リングを開きます。 鍵リングの内容にアクセスする、基礎となるシステム許可機能 (SAF) は、RACF か、または同等のセキュリティー・マネージャーによって制御されます。

このサービスは IRRSDL00 (R_datalib) 呼び出し可能サービスです。 この呼び出し可能サービスは、RACF FACILITY クラスに定義される、RACF RACDCERT コマンドの保護に使用するのと同じプロファイルで保護されます。 したがって、Advanced Message Security ユーザー ID には、それらのプロファイルへのアクセス権を、以下のコマンドを使用して付与する必要があります。

a) まだ行っていない場合は、RACDCERT コマンドおよび IRRSDL00 呼び出し可能サービスを保護する RACF FACILITY クラスに、RACF 総称プロファイルを定義します。

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)
SETROPTS RACLIST(FACILITY) REFRESH
```

b) RACF 総称プロファイルに対する権限を、開始タスク・ユーザー ID に付与します。

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(WMQAMSM) ACC(READ)
```

あるいは、以下のように、RDATALIB クラスのデータ・サービス・タスク・ユーザーの鍵リングに対する READ アクセス権を付与することができます。

```
PERMIT WMQASMD.DRQ.AMS.KEYRING.LST CLASS(RDATALIB) ID(WMQAMSM) ACC(READ)
```

7. リソース・セキュリティーを構成します。

a) Advanced Message Security 開始タスク・ユーザーには、バッチ・アプリケーションとしてキュー・マネージャーに接続するための権限が必要です。

キュー・マネージャーの接続セキュリティーが使用可能にされている場合、以下のコマンドを使用して、AMS タスクにキュー・マネージャーに接続する権限を付与します。

```
PERMIT hlq.BATCH CLASS(MQCONN) ID(WMQAMSM) ACC(READ)
```

ここで、*hlq* は、キュー・マネージャー名またはキュー共有グループ名のいずれかです。

詳しくは、バッチ接続のための接続セキュリティー・プロファイルを参照してください。

b) Advanced Message Security 開始タスク・ユーザーには、SYSTEM.PROTECTION.POLICY.QUEUE をブラウズするための権限が必要です。

キュー・マネージャーのキュー・セキュリティーがアクティブである場合、以下のコマンドを使用して、AMS ユーザーにキューにアクセスする権限を付与します。

```
RDEFINE MQQUEUE hlq.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
PERMIT hlq.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE) ID(WMQAMSM) ACCESS(READ)
```

ここで、*hlq* は、キュー・マネージャー名またはキュー共有グループ名のいずれかです。

キュー・マネージャーが大/小文字混合プロファイルを使用している場合、代わりに MXQUEUE クラスにプロファイルを定義します。

CSQ0UTIL ユーティリティーを使用して AMS セキュリティー・ポリシーを管理するには、管理者に、SYSTEM.PROTECTION.POLICY.QUEUE にメッセージを書き込むためのアクセス権限が必要です。これを実行するには、キューを保護しているプロファイルに対する UPDATE アクセス権を付与します。

詳しくは、キュー・セキュリティー用のプロファイルを参照してください。

次のタスク

973 ページの『Advanced Message Security のセキュリティー管理者に RACDCERT 権限を付与する』

z/OS *Advanced Message Security* のセキュリティー管理者に *RACDCERT* 権限を付与する

Advanced Message Security のセキュリティー管理者には、デジタル証明書を作成して管理するために RACDCERT コマンドを使用する権限が必要です。

手順

- この役割のために適切なユーザー ID を特定し、RACDCERT コマンドを使用する権限を付与します。 以下に例を示します。

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(admin) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

ここで、admin は、Advanced Message Security のセキュリティー管理者のユーザー ID です。

次のタスク

973 ページの『Advanced Message Security のためのリソース・アクセス権をユーザーに付与する』

z/OS *Advanced Message Security* のためのリソース・アクセス権をユーザーに付与する

Advanced Message Security のユーザーには、適切なリソース・アクセス権が必要です。

このタスクについて

Advanced Message Security のユーザー、つまり Advanced Message Security によって保護されたメッセージの書き込みと取得を行うユーザーは、以下を必要とします。

- ユーザー ID に関連付けられた OMVS セグメント
- IRR.DIGTCERT.LISTRING または RDATALIB に対するアクセス権

- ICSF クラスの CSFSERV および CSFKEYS プロファイルに対するアクセス権
- SYSTEM.PROTECTION.ERROR.QUEUE への書き込み権限

Advanced Message Security タスクは、クライアント ID を一時的に想定します。つまり、Advanced Message Security によって、保護されているキューへの IBM MQ メッセージの処理中に、タスクは Advanced Message Security ユーザーの z/OS ユーザー ID の代理として機能することになります。

このタスクにユーザーの z/OS ID が付与されるようにするには、クライアントの z/OS ユーザー ID に、ユーザー・プロファイルと関連付けられた OMVS セグメントが定義されていなければなりません。

管理の助けとして、RACF は、RACF ユーザーおよびグループのプロファイルに関連付けできる、デフォルト OMVS セグメントを定義する機能を備えています。 このデフォルトは、z/OS ユーザー ID またはグループのプロファイルに OMVS セグメントが明示的に定義されていない場合に使用されます。 多数のユーザーが Advanced Message Security を使用する予定であるなら、ユーザーごとに明示的に OMVS セグメントを定義する代わりに、このデフォルトの使用を選択することもできます。

「*z/OS: Security Server RACF* セキュリティー管理者のガイド」には、デフォルト OMVS セグメントを定義するための詳細な手順が記載されています。 この資料に概説されている手順を参照して、RACF ユーザーおよびグループのプロファイルにデフォルト OMVS セグメントを定義することが、ご使用のインストール済み環境にとって適切かどうか見極めてください。

手順

1. FACILITY クラスの IRR.DIGTCERT.LISTRING プロファイルに対する READ 権限を以下のように付与します。
   - FACILITY クラスの IRR.DIGTCERT.LISTRING プロファイルに対する READ 権限をすべてのユーザーに付与するには、次のコマンドを発行します。

     ```
     RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(READ)
     ```

   - FACILITY クラスの IRR.DIGTCERT.LISTRING プロファイルに対する READ 権限をユーザーごとに付与するには、次のコマンドを発行します。

     ```
     PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACCESS(READ)
     ```

     ここで、userid は Advanced Message Security ユーザーの名前です。
   - または、RDATALIB クラスを使用して、特定の鍵リングに対するアクセス権限を付与します。 RDATALIB 権限は、IRR.DIGTCERT.LISTRING 権限よりも優先されます。 以下に例を示します。

     ```
     PERMIT user.DRQ.AMS.KEYRING.LST CLASS(RDATALIB) ID(user) ACC(READ)
     ```

2. ICSF 管理の証明書と秘密鍵を使用している場合には、Advanced Message Security ユーザーは、特定のクラス CSFSERV および CSFKEYS のプロファイルに対するアクセス権を必要とします。 このアクセス権について、以下の表に示します。

表 *80.* クラス *CSFSERV* および *CSFKEYS* のプロファイルに対する必須のユーザー・アクセス権		
クラス	**プロファイル**	**アクセス権**
CSFSERV	CSFDSG	READ
CSFSERV	CSFPKE	READ
CSFSERV	CSFPKD	READ
CSFSERV	CSFDSV	READ
CSFKEYS	ICSF PKDS ラベル	READ

3. AMS ポリシーが定義されたキューに対して操作を実行するアプリケーションには、SYSTEM.PROTECTION.ERROR.QUEUE へのメッセージ書き込み権限が必要です。 以下のコマンドを使用して、キューへの書き込みアクセス権限を付与します。

```
RDEFINE MQQUEUE hlq.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
PERMIT hlq.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE) ID(userId) ACCESS(UPDATE)
```

ここで、*hlq* は、キュー・マネージャー名またはキュー共有グループ名のいずれかであり、*userID* は、アプリケーション・ユーザー ID です。

次のタスク

975 ページの『Advanced Message Security の鍵リングを作成する』

z/OS *Advanced Message Security* の鍵リングを作成する

署名および暗号化用に Advanced Message Security(AMS)で使用される証明書は、z/OSSAF 鍵リングに保管されます。 これらの鍵リングおよび証明書は、AMS を使用する前に作成する必要があります。

このタスクについて

Advanced Message Security は、以下の鍵リング内の証明書にアクセスします。

- AMS アドレス・スペース・ユーザーにより所有される単一鍵リング。
- 定義された AMS ポリシーを使用してキューでメッセージを送信または受信する個別ユーザーにより所有される鍵リング。

これらの鍵リングはすべて `drq.ams.keyring` という名前にする必要があります。

AMS によって使用される鍵リングおよび証明書の詳細情報とシナリオ例については、z/OS での証明書の使用を参照してください。

以下の手順に従って、AMS に必要な鍵リングを作成して、証明書を鍵リングに接続します。 AMS アドレス・スペース・ユーザーにより所有される鍵リングは、AMS を開始する前に作成する必要があります。 メッセージを送信または受信するユーザーにより所有される鍵リングはいつでも作成できます。

手順

1. 以下のコマンドを発行して、AMS アドレス・スペース・ユーザーにより所有される鍵リングを作成します。

   ```
   RACDCERT ID(amsUser) ADDRING(drq.ams.keyring)
   ```

   ここで、*amsUser* は、AMS アドレス・スペースのユーザー ID です。
2. AMS により保護されるメッセージを送信または受信するユーザーごとに鍵リングを作成するには、ステップ 1 のコマンドをユーザー ID ごとに発行します。
3. AMS アドレス・スペースのユーザー ID により所有される鍵リングに、ユーザー証明書の発行者の認証局 (CA) 証明書を接続します。 以下のコマンドを発行します。

   ```
   RACDCERT ID(amsUser) CONNECT(CERTAUTH LABEL('caLabel') RING(drq.ams.keyring))
   ```

   ここで、*amsUser* は、AMS アドレス・スペースのユーザー ID であり、*caLabel* は、CA 証明書のラベルです。

   CA として RACF を使用しており、認証局証明書を作成する必要がある場合は、ローカル認証局証明書の定義の例に従ってください。
4. AMS により保護されるキューのメッセージを暗号化するためにプライバシーまたは機密性セキュリティー・ポリシーを使用する場合は、AMS アドレス・スペースのユーザー ID により所有される鍵リングに、メッセージ受信者の証明書を接続します。 以下のコマンドを発行します。

   ```
   RACDCERT ID(amsUser) CONNECT(ID(userId) LABEL('certLabel')
   RING(drq.ams.keyring) USAGE(SITE))
   ```

   ここで、*amsUser* は、AMS アドレス・スペースのユーザー ID であり、*userId* は、メッセージ受信者であり、*certLabel* は、ユーザー証明書のラベルです。

   USAGE(SITE) 属性により、秘密鍵は鍵リング内でアクセスされることを防ぎます。

RACF で独自の証明書を作成する場合は、秘密鍵によるデジタル証明書の作成の例に従って証明書を作成してください。

5. ユーザーにより所有される鍵リングに、AMS により保護されるメッセージを送信または受信する各ユーザーの証明書を接続します。 証明書は鍵リングのデフォルト証明書として接続する必要があります。 以下のコマンドを発行します。

```
RACDCERT ID(userId) CONNECT(ID(userId) LABEL('certLabel')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

ここで、*userId* は、メッセージを送信または受信するユーザーであり、*certLabel* は、ユーザー証明書のラベルです。

注:

a. アプリケーションが出力用にのみキューをオープンし、 AMS 機密性ポリシーによって保護されているキューにメッセージを送信する場合、ステップ 975 ページの『2』 および 976 ページの『5』 は必要ありません。

b. アプリケーションが入力/ブラウズのためにのみキューをオープンし、AMS 保全性ポリシーによって保護されているキューからメッセージを受信する場合は、ステップ 975 ページの『2』 および 976 ページの『5』 は必要ありません。

次のタスク

976 ページの『Advanced Message Security を有効にします』

z/OS *Advanced Message Security* を有効にします

キュー・マネージャーのセキュリティー・ポリシーの機能は、システム・パラメーター・モジュールの SPLCAP パラメーターによって制御されます。

このタスクについて

以下のステップに従って、単一のキュー・マネージャーに対して、Advanced Message Security (AMS)を使用可能にします

この作業では、システム・パラメーター・モジュールを変更する必要があります。 システム・パラメーター・モジュールの作成およびカスタマイズについて詳しくは、936 ページの『システム・パラメーター・モジュールの調整』を参照してください。

手順

1. CSQ6SYSP で **SPLCAP** を YES に設定します。 CSQ6SYSP マクロについての詳細は、938 ページの『CSQ6SYSP の使用』を参照してください。
2. ライセンス資格に応じて、**AMSPROD** を AMS、ADVANCED、または ADVANCEDVUE のいずれかに設定します。 CSQ6USGP マクロについて詳しくは、CSQ6USGP の使用を参照してください。
3. システム・パラメーター・モジュールを再コンパイルします。
4. 更新されたシステム・パラメーター・モジュールを使用してキュー・マネージャーを再始動します。 キュー・マネージャーが始動すると、AMS アドレス・スペースは自動的に開始されます。

z/OS mqweb サーバーの構成

これらのトピックを、mqweb サーバーを構成するためのステップバイステップ・ガイドとしてご利用ください。

関連タスク

821 ページの『IBM MQ Console および REST API の構成』

IBM MQ Console と REST API をホストする mqweb サーバーには、デフォルトの構成が準備されています。 どちらのコンポーネントを使用する場合も、ユーザーにログインを許可するために、セキュリティーを構成するなどの複数の構成タスクを実行する必要があります。 このトピックでは、使用可能なすべての構成オプションについて説明します。

z/OS *mqweb* サーバーの作成

IBM MQ for z/OS UNIX System Services Web Components をインストールして IBM MQ Console、または REST API を使用する場合は、mqweb サーバーを作成してカスタマイズする必要があります。

始める前に

crtmqweb スクリプトを実行して mqweb サーバーを作成する前に、システム上の 64 ビット・バージョンの Java を参照するように JAVA_HOME 環境変数を設定します。

IBM MQ Console および administrative REST API には、SYSTEM.REST.REPLY.QUEUE キュー。 929 ページの『初期設定入力データ・セットをカスタマイズする』の **CSQ4INSG** サンプルを使用して、このキューを作成します。

重要 : mqweb サーバーの始動時に、以下の出力に示すようなエラー・メッセージ CWWKG0014E が表示された場合:

```
     Launching mqweb (MQM MVS/ESA V9 R2.0/wlp...) (en_US)
               ÝAUDIT   ¨ CWWKE0001I: The server mqweb has been
launched.
               ÝWARNING ¨ CWWKF0009W: The server has not been configured to install any
features.
               ÝAUDIT   ¨ CWWKF0011I: The mqweb server is ready to run a smarter planet.
The mqweb server started in 6.348 seconds.
               ÝERROR   ¨ CWWKG0014E: The configuration parser detected an XML syntax
error while parsing the root of the configuration and the referenced configuration
documents.
                                Error: An invalid XML character (Unicode: 0x4c) was found
in the prolog of the document.
                                 File: file:<your filepath>/servers/mqweb/server.xml Line:
1 Column: 1
```

AUTOCVT の z/OS 設定 (ファイルをあるコード・セットから別のコード・セットに自動的に変換する) を確認し、以下のいずれかを実行して、必要に応じて値を調整する必要があります。

USS 端末の場合:

コマンド echo **$_BPXK_AUTOCVT** を発行して、この環境変数の値を表示します。 環境変数が定義されていない場合、値は表示されません。

環境変数を設定するには、 _BPXK 環境変数を参照してください。

システム全体:

「z/OS UNIX System Services (OMVS) の状況の表示」の例 6 は、BPXPRMxx 内のシステム全体の AUTOCVT ステートメントの値を表示する方法を示しています。

環境変数をシステム全体に設定するには、BPXPRMxx で AUTOCVT ステートメントを使用します。

環境変数 _BPXX_AUTOCVT が USS 端末で設定されている場合、BPXPRMxx 内のステートメント AUTOCVT のシステム全体の設定をオーバーライドします。

このタスクについて

- このタスクは、 IBM MQ Console または REST API を実行する z/OS システムごとに 1 回実行します。
- administrative REST API を使用するには、実行中の IBM MQ のバージョンごとに mqweb サーバーが必要です。 例えば、 IBM MQ 9.3.0、 9.2.5 、および 9.2.0 を実行している場合、3 つの異なる mqweb サーバーが必要です。
- 前のバージョンからマイグレーションするときに、サーバー構成のリフレッシュまたは変更が必要になる場合があります。

IBM MQ Console および REST API には、mqweb という単一の WebSphere Liberty サーバーを作成する必要があります。

このサーバーの構成ファイルとログ・ファイルはすべて、Liberty ユーザー・ディレクトリー内に格納されます。

mqweb サーバーは、その実行に使用される製品 ID (PID) を使用して構成する必要があります。 mqweb サーバーの作成時に PID が設定されます。 mqweb サーバーの接続先のローカル・キュー・マネージャーを実行するために使用されるのと同じ PID を使用します。

注 : ローカル・キュー・マネージャーが複数の異なる PID で実行される場合は、mqweb サーバーを実行する ID を 1 つ選択します。

PID およびそれらが z/OS でどのように使用されるかについて詳しくは、 IBM MQ for z/OS 製品での製品使用状況の記録を参照してください。

mqweb サーバーの作成後に、 setmqweb コマンドを使用して、mqweb サーバーを実行する PID を変更することができます。

mqweb サーバーを作成するには、以下のステップを実行します。

手順

1. mqweb サーバーを実行する PID を決定します。
2. Liberty ユーザー・ディレクトリーとして適切な場所を選択します。

   mqweb サーバーを実行するユーザー ID には、このユーザー・ディレクトリーとそのコンテンツへの読み取り権限と書き込み権限が必要です。 このユーザー・ディレクトリーにはログ・ファイルが含まれているため、サーバー構成に加えて、このディレクトリーを別のファイル・システムに作成してください。

   注 : mqweb サーバーが始動するときに、かなりの量のディスク入出力があります。 mqweb サーバーの始動にかかる時間を短縮するには、 IBM MQ インストール z/OS UNIX ・ファイル・システムと Liberty ユーザー・ディレクトリー・ファイル・システムの両方がシスプレックス対応であるか、mqweb サーバーが実行されているシステムにローカルにマウントされていることを確認してください。
3. z/OS UNIX System Services で、以下のコマンドを発行して、現行作業ディレクトリーを `PathPrefix/web/bin` に変更します。

   ```
   cd PathPrefix/web/bin
   ```

   PathPrefix は、IBM MQ for z/OS UNIX System Services Components のインストール・パスです。
4. **crtmqweb** スクリプトを実行して、テンプレート mqweb サーバー定義を含む Liberty ユーザー・ディレクトリーを作成します。

   crtmqweb コマンドの形式は、次のとおりです。

   ```
   crtmqweb user_directory -p pid_value
   ```

   ここで、

   user_directory
   ステップ 978 ページの『2』で決定した Liberty ユーザー・ディレクトリーです。 このパラメーターはオプションです。 このパラメーターを指定しない場合は、デフォルトの Liberty ユーザー・ディレクトリー `/var/mqm/web/installation1` が使用されます。

   pid_value
   mqweb サーバーを実行する PID を示します。 この PID は、ステップ 978 ページの『1』で選択した PID です。 *pid_value* は、以下のいずれかの値です。

   MQ
   mqweb サーバーは PID IBM MQ for z/OS (5655-MQ9) の下で実行されます。

   VUE
   mqweb サーバーは、PID IBM MQ for z/OS Value Unit Edition (VUE) (5655-VU9) の下で実行されます。

   ADVANCEDVUE
   mqweb サーバーは PID IBM MQ Advanced for z/OS VUE (5655-AV1) で実行されます。

例えば、 Liberty ユーザー・ディレクトリーが /usr/mqweb で PID が IBM MQ Advanced for z/OS VUE (5655-AV1) の mqweb サーバーを作成するには、次のコマンドを実行します。

```
./crtmqweb /usr/mqweb -p ADVANCEDVUE
```

5. 以下のコマンドを使用して、 Liberty ユーザー・ディレクトリー内のディレクトリーおよびファイルの所有権を、mqweb サーバーが実行されるユーザー ID およびグループに属するように変更します。

   ```
   chown -R userid:group path
   ```

   このパスに対する書き込み権限をグループに付与するには、次のコマンドを実行します。

   ```
   chmod -R 770 path
   ```

次のタスク

979 ページの『mqweb サーバー用のプロシージャーの作成』

関連タスク

821 ページの『IBM MQ Console および REST API の構成』

IBM MQ Console と REST API をホストする mqweb サーバーには、デフォルトの構成が準備されています。どちらのコンポーネントを使用する場合も、ユーザーにログインを許可するために、セキュリティーを構成するなどの複数の構成タスクを実行する必要があります。 このトピックでは、使用可能なすべての構成オプションについて説明します。

z/OS *mqweb* サーバー用のプロシージャーの作成

IBM MQ for z/OS UNIX System Services Web Components をインストールして IBM MQ Console、または REST API を使用する場合は、mqweb サーバーを開始するためのカタログ式プロシージャーを作成する必要があります。 mqweb サーバーは、 IBM MQ Console および REST API のホストとなる Liberty サーバーです。

- このタスクは、 IBM MQ Console または REST API を実行する z/OS システムごとに 1 回実行する必要があります。
- 実行されている IBM MQ のバージョンごとに mqweb サーバーが必要です。 例えば、IBM MQ for z/OS 9.1.0 のキュー・マネージャーの MQWB0910 という開始タスク、IBM MQ for z/OS 9.0.5 のキュー・マネージャーの MQWB0905 という開始タスクなどがあります。

  z/OS システム上にキュー・マネージャーが 1 つしかない場合は、Liberty サーバーの開始タスクを 1 つ実行し、キュー・マネージャーのマイグレーション時に、その開始タスクが使用するライブラリーを変更します。
- 前のバージョンから移行する際に、カタログ式プロシージャーを変更しなければならない場合があります。

以下の手順を実行して、カタログ式プロシージャーを作成します。

1. サンプルの開始タスク・プロシージャー `thlqual.SCSQPROC(CSQ4WEBS)` をプロシージャー・ライブラリーにコピーします。

   企業の規約に従ってプロシージャーに名前を付けます。

   例えば `MQWB0910` にします。これは、IBM MQ for z/OS 9.1.0 mqweb サーバー用のカタログ式プロシージャーであることを示しています。
2. サンプル・プロシージャー CSQ4WEBS 内の命令を使用して、要件に合うようにプロシージャーを調整します。

   Liberty ユーザー・ディレクトリーは、mqweb サーバー定義を作成するために **crtmqweb** スクリプトを実行したときに指定したディレクトリーです。

   詳細は、977 ページの『mqweb サーバーの作成』を参照してください。

   注 : ファイルのデータが小文字になるように、メンバーの編集時には必ず「**大文字オフ (Caps off)**」を指定してください。

3. 外部セキュリティー管理プログラムの下で実行できるようにプロシージャーに許可を与えます。
4. IBM ワークロード・マネージャー (WLM) を使用して、このアドレス・スペースを分類します。

   mqweb サーバーは IBM MQ アプリケーションで、ユーザーはこのアプリケーションと対話します。 アプリケーションは WLM での重要度を高くする必要はなく、サービス・クラス **STCUSER** が適切です。

次のタスク

822 ページの『mqweb サーバーの基本構成』のステップに従って mqweb サーバーの構成を完了します。

関連タスク

821 ページの『IBM MQ Console および REST API の構成』
IBM MQ Console と REST API をホストする mqweb サーバーには、デフォルトの構成が準備されています。 どちらのコンポーネントを使用する場合も、ユーザーにログインを許可するために、セキュリティーを構成するなどの複数の構成タスクを実行する必要があります。 このトピックでは、使用可能なすべての構成オプションについて説明します。

z/OS z/OS でのキュー・マネージャーのテスト

キュー・マネージャーのカスタマイズまたは移行を済ませた後、IBM MQ for z/OS に添付されているインストール検査プログラムおよびサンプル・アプリケーションを実行することによって、キュー・マネージャーをテストできます。

このタスクについて

IBM MQ for z/OS をインストールおよびカスタマイズした後、提供されているインストール検査プログラム CSQ4IVP1 を使用して、IBM MQ for z/OS が操作可能であることを確認できます。

基本インストール検査プログラム CSQ4IVP1 は、C、COBOL、または CICS サンプルを使用せずに、非共有キューをテストし、ベース IBM MQ を検査します。

基本インストール検査を実行した後、別のキューで CSQ4IVP1 を使用して共有キューをテストし、Db2 とカップリング・ファシリティーが正しくセットアップされていることをテストすることもできます。 分散キューイングが操作可能であることを確認するには、提供されているインストール検査プログラム CSQ4IVPX を使用できます。

CSQ4IVP1 はロード・モジュールとして提供され、Message Queue Interface (MQI) の典型的な使用方法を示すソース・モジュールとして一連のプロシージャー型サンプル・アプリケーションを提供します。 これらのソース・モジュールを使用して、さまざまなプログラミング言語環境をテストすることができます。 提供されたサンプル JCL を使用して、インストール済み環境に適した他のサンプルのうちどれでもコンパイルおよびリンク・エディットできます。

手順

- z/OS でキュー・マネージャーをテストする方法の詳細については、以下のサブトピックを参照してください。
  - 981 ページの『基本インストール検査プログラムの実行』
  - 984 ページの『キュー共有グループのテスト』
  - 985 ページの『分散キューイングのテスト』
  - 988 ページの『IBM MQ for z/OS を使用した C、C++ 、COBOL、PL/I、および CICS プログラムのテスト』

関連概念

IBM MQ for z/OS の概念

関連タスク

z/OS での IBM MQ 環境の計画
908 ページの『z/OS でのキュー・マネージャーの構成』
IBM MQ for z/OS でキュー・マネージャーを構成するには、この指示に従ってください。

z/OS 基本インストール検査プログラムの実行

IBM MQ をインストールおよびカスタマイズした後、提供されているインストール検査プログラム CSQ4IVP1 を使用して、IBM MQ が操作可能であることを確認できます。

基本インストール検査プログラムは、C、COBOL、または CICS サンプルを使用せずに基本 IBM MQ を検査するバッチ・アセンブラー IVP です。

バッチ・アセンブラー IVP は SMP/E でリンク・エディットされ、そのロード・モジュールはライブラリー thlqual.SCSQLOAD に入って出荷されます。

SMP/E APPLY ステップおよびカスタマイズ・ステップの両方を完了した後、バッチ・アセンブラー IVP を実行してください。

詳細については、以下のセクションを参照してください。

- CSQ4IVP1 アプリケーションの概要
- CSQ4IVP1 の実行準備
- CSQ4IVP1 の実行
- CSQ4IVP1 の実行結果の検査

CSQ4IVP1 アプリケーションの概要

CSQ4IVP1 は、使用している IBM MQ サブシステムに接続して、次の基本機能を実行するためのバッチ・アプリケーションです。

- IBM MQ 呼び出しを発行する。
- コマンド・サーバーと通信する。
- トリガーが活動状態であることを検証する。
- 動的キューを生成および削除する。
- メッセージの有効期限処理を検査する。
- メッセージのコミット処理を検査する。

CSQ4IVP1 の実行準備

CSQ4IVP1 を実行する前に、次のことを行ってください。

1. IVP 項目が、キュー・マネージャー始動プログラムの中の CSQINP2 データ・セット連結に入っているかどうか検査します。 この IVP 項目は、メンバー thlqual.SCSQPROC(CSQ4IVPQ) に入って提供されます。 入っていない場合には、thlqual.SCSQPROC(CSQ4IVPQ) で提供された定義を、CSQINP2 連結に追加してください。 キュー・マネージャーが現在実行されている場合、これらの定義を有効にするために、キュー・マネージャーを再始動する必要があります。
2. インストール検査プログラムを実行するために必要なサンプル JCL の CSQ4IVPR は、ライブラリー thlqual.SCSQPROC に入っています。

   CSQ4IVPR JCL をカスタマイズするため、IBM MQ ライブラリーの高位修飾子、使用する各国語、4 文字の IBM MQ キュー・マネージャー名、およびジョブ出力の宛先を指定します。
3. RACF を更新して、IBM MQ セキュリティーが活動状態の場合に CSQ4IVP1 が RACF の資源にアクセスできるようにします。

   IBM MQ のセキュリティーが使用可能になったときに CSQ4IVP1 を実行するには、オブジェクトへのアクセス権限を持っている RACF ユーザー ID が必要です。 リソースを RACF に定義する詳細については、z/OS でのセキュリティーのセットアップを参照してください。 IVP を実行するユーザー ID には、次のようなアクセス権が必要です。

権限	プロファイル	クラス
READ	ssid.DISPLAY.PROCESS	MQCMDS
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.COMMAND.REPLY.MODEL	MQQUEUE
UPDATE	ssid.CSQ4IVP1.**	MQQUEUE
READ	ssid.BATCH	MQCONN

上記の要件は、すべての IBM MQ セキュリティーが活動状態であることを想定しています。 IBM MQ セキュリティーを活動化するための RACF コマンドを 982 ページの図 101 に示します。 この例では、キュー・マネージャー名が CSQ1 であること、およびサンプル CSQ4IVP1 を実行するユーザーのユーザー ID が TS101 であることを前提としています。

```
RDEFINE MQCMDS CSQ1.DISPLAY.PROCESS
PERMIT CSQ1.DISPLAY.PROCESS CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.REPLY.MODEL
PERMIT CSQ1.SYSTEM.COMMAND.REPLY.MODEL CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.CSQ4IVP1.**
PERMIT CSQ1.CSQ4IVP1.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)
```

図 *101. CSQ4IVP1* のためのコマンド *RACF*

CSQ4IVP1 の実行

これらのステップを完了したら、キュー・マネージャーを開始してください。 キュー・マネージャーが既に実行中で、かつ CSQINP2 を変更した場合は、キュー・マネージャーを停止して再始動する必要があります。

IVP はバッチ・ジョブとして実行されます。 ジョブ・カードを、インストール先のジョブ実行依頼の要件に合うようにカスタマイズしてください。

CSQ4IVP1 の実行結果の検査

IVP は、10 のステージに分かれており、各ステージが完了して完了コード・ゼロを出さないと、次のステージを実行することはできません。 IVP は、次のものをリストしたレポートを生成します。

- 接続されているキュー・マネージャーの名前
- 各ステージから戻された完了コードおよび理由コードを示す 1 行のメッセージ
- 1 行の通知メッセージ (該当する場合)

サンプル・レポートは、984 ページの図 102 にあります。

z/OS 完了コードと理由コードの解説については、IBM MQ for z/OS のメッセージ、完了コード、および理由コードを参照してください。

ステージによっては複数の IBM MQ 呼び出しがあるため、障害があった場合は、障害を戻した特定の IBM MQ 呼び出しを示すメッセージが発行されます。 また、あるステージでは、IVP が説明および診断のための情報をコメント欄に書き込みます。

IVP ジョブは、特定のキュー・マネージャー・オブジェクトの排他制御を要求するので、システム全体を通して単一スレッドで実行されます。 ただし、キュー・マネージャーに対して実行できる IVP の回数には、制限がありません。

各ステージで実行される機能は、次のとおりです。

ステージ 1
MQCONN API 呼び出しを発行することにより、キュー・マネージャーに接続します。

ステージ 2
要求メッセージを取り出すために、コマンド・サーバーによって使用されるシステム・コマンド入力キューの名前を判別します。 このキューは、ステージ 5 からの表示要求を受け取ります。

これを行うための呼び出しの順序は、次のとおりです。

1. キュー・マネージャー・オブジェクトをオープンするために、キュー・マネージャー名を指定して MQOPEN 呼び出しを発行します。
2. システム・コマンド入力キューの名前を知るために、MQINQ 呼び出しを発行します。
3. キュー・マネージャーの各種イベント・スイッチを知るために、MQINQ 呼び出しを発行します。
4. キュー・マネージャー・オブジェクトを閉じるために、MQCLOSE 呼び出しを発行します。

このステージが正常に完了すると、システム・コマンド入力キューの名前がコメント欄に表示されます。

ステージ 3
MQOPEN 呼び出しを使用して、開始キューをオープンします。

このキューは、ステージ 5 からのリクエストにコマンドサーバーが返答した結果として、到着するトリガー・メッセージを予期して、このステージでオープンされます。 トリガー基準を満たすために、入力用にキューをオープンする必要があります。

ステージ 4
モデルとして CSQ4IVP1.MODEL キューを使用して、永続動的キューを作成します。 動的キューは、それが作成されたモデルと同じ属性を持ちます。 これは、ステージ 5 でのコマンド・サーバー要求からの応答がこのキューに書き込まれると、ステージ 3 でオープンされた開始キューにトリガー・メッセージが書き込まれることを意味します。

このステージが正常に完了すると、永続動的キューの名前がコメント欄に表示されます。

ステージ 5
コマンド・サーバーのコマンド・キューに対して、MQPUT1 要求を発行します。

タイプが MQMT_REQUEST のメッセージが、プロセス CSQ4IVP1 の表示を要求して、システム・コマンド入力キューに書き込まれます。 このメッセージのメッセージ記述子は、コマンド・サーバーの応答のための応答先キューとしてステージ 4 で作成された永続動的キューを指定しています。

ステージ 6
開始キューから **MQGET** 要求を発行します。 この段階では、ステージ 3 でオープンされた開始キューに対して、1 分間隔で GET WAIT が発行されます。 戻されるメッセージは、応答先キューに書き込まれるコマンド・サーバーの応答メッセージによって、生成されるトリガー・メッセージとなることが期待されます。

ステージ 7
ステージ 4 で作成した永続動的キューを削除します。 キューには依然としてメッセージが含まれているため、MQCO_PURGE_DELETE オプションが使用されます。

ステージ 8

1. 動的キューをオープンします。
2. 有効期限が設定されたメッセージを MQPUT します。
3. メッセージの期限が切れるのを待ちます。
4. 期限切れのメッセージに対して MQGET を試行します。
5. キューを MQCLOSE します。

ステージ 9

1. 動的キューをオープンします。
2. メッセージを MQPUT します。
3. MQCMIT を発行して、現在の作業単位をコミットします。
4. メッセージを MQGET します。
5. MQBACK を発行して、メッセージをバックアウトします。
6. 同じメッセージを MQGET し、バックアウト・カウントが 1 に設定されていることを確認します。
7. MQCLOSE を発行して、キューを閉じます。

ステージ 10

MQDISC を使用して、キュー・マネージャーから切断します。

IVP を実行した後、必要のないオブジェクトはすべて削除できます。

IVP を正常に実行できない場合は、手動でステップを 1 つ 1 つ試行し、どの機能に障害があるかを判別してください。

```
DATE : 2005.035                IBM MQ for z/OS - V6           PAGE : 0001
INSTALLATION VERIFICATION PROGRAM
PARAMETERS ACCEPTED. PROGRAM WILL CONNECT TO : CSQ1
,OBJECT QUALIFER : CSQ4IVP1
INSTALLATION VERIFICATION BEGINS :
STAGE 01 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR BRIDGE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS EXCP FOR CHANNEL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR SSL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR INHIBITED EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR LOCAL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR PERFORMENCE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR REMOTE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR START/STOP EVENTS
STAGE 02 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 SYSTEM.COMMAND.INPUT
STAGE 03 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 04 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9810EFEAC8980
STAGE 05 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 06 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 07 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 08 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9810F0070E645
STAGE 09 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9812BA8706803
STAGE 10 COMPLETE. COMPCODE : 0000 REASON CODE : 0000>>>>>>>>>>   END OF REPORT   <<<<<<<<<<
```

図 *102. CSQ4IVP1* からのサンプル・レポート

z/OS キュー共有グループのテスト

基本インストール検査プログラム CSQ4IVP1 は非共有キューをテストします。

CSQ4IVP1 は、キュー・マネージャーがキュー共有グループのメンバーであるかどうかにかかわらず、使用することができます。 基本 IVP を実行した後、別のキューで CSQ4IVP1 インストール検査プログラムを使用して、共有キューのテストを行うことができます。 また、Db2 とカップリング・ファシリティーが正しく設定されているかどうかをテストします。

キュー共有グループ用の CSQ4IVP1 の実行準備

CSQ4IVP1 を実行する前に、次のことを行ってください。

1. 967 ページの『カップリング・ファシリティーのセットアップ』に説明されているとおり、IVP が使用するカップリング・ファシリティー構造を CFRM ポリシー・データ・セットに追加します。 提供されているサンプルでは、APPLICATION1 という構造を使用しますが、必要に応じて変更することもできます。
2. IVP 項目が、キュー・マネージャー始動プログラムの中の CSQINP2 データ・セット連結に入っているかどうか検査します。 この IVP 項目は、メンバー thlqual.SCSQPROC(CSQ4IVPG) に入って提供されます。 入っていない場合には、thlqual.SCSQPROC(CSQ4IVPG) で提供された定義を、CSQINP2 連結に追加してください。 キュー・マネージャーが現在実行されている場合、これらの定義を有効にするために、キュー・マネージャーを再始動する必要があります。

3. 必要であれば、thlqual.SCSQPROC(CSQ4IVPG) で使用されているカップリング・ファシリティー構造の名前を変更します。
4. キュー共有グループ用のインストール検査プログラムを実行するために必要なサンプル JCL の CSQ4IVPS は、ライブラリー thlqual.SCSQPROC に入っています。

   CSQ4IVPS JCL をカスタマイズするため、IBM MQ ライブラリーの高位修飾子、使用する各国語、4 文字の IBM MQ キュー・マネージャー名、およびジョブ出力の宛先を指定します。
5. RACF を更新して、IBM MQ セキュリティーが活動状態の場合に CSQ4IVP1 が RACF の資源にアクセスできるようにします。

   IBM MQ のセキュリティーが使用可能になったときに CSQ4IVP1 を実行するには、オブジェクトへのアクセス権限を持っている RACF ユーザー ID が必要です。 リソースを RACF に定義する詳細については、z/OS でのセキュリティーのセットアップを参照してください。 IVP を実行するユーザー ID には、基本 IVP を実行するのに必要な権限に加えて、次のようなアクセス権が必要です。

権限	プロファイル	クラス
UPDATE	ssid.CSQ4IVPG.**	MQQUEUE

   上記の要件は、すべての IBM MQ セキュリティーが活動状態であることを想定しています。 IBM MQ セキュリティーを活動化するための RACF コマンドを 985 ページの図 103 に示します。 この例では、キュー・マネージャー名が CSQ1 であること、およびサンプル CSQ4IVP1 を実行するユーザーのユーザー ID が TS101 であることを前提としています。

```
RDEFINE MQQUEUE CSQ1.CSQ4IVPG.**
PERMIT CSQ1.CSQ4IVPG.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)
```

図 *103.* キュー共有グループ用の *CSQ4IVP1* のための *RACF* コマンド

キュー共有グループ用の CSQ4IVP1 の実行

これらのステップを完了したら、キュー・マネージャーを開始してください。 キュー・マネージャーが既に実行中で、かつ CSQINP2 を変更した場合は、キュー・マネージャーを停止して再始動する必要があります。

IVP はバッチ・ジョブとして実行されます。 ジョブ・カードを、インストール先のジョブ実行依頼の要件に合うようにカスタマイズしてください。

キュー共有グループ用の CSQ4IVP1 の実行結果の検査

キュー共有グループ用の IVP は、作成されるキューが CSQIVPG と呼ばれることを除いて、基本 IVP と同じように動作します。 *xx*。 982 ページの『CSQ4IVP1 の実行結果の検査』にある指示に従って、キュー共有グループ用の IVP の実行結果を検査してください。

z/OS 分散キューイングのテスト

提供されたインストール検査プログラム CSQ4IVPX を使用すると、分散キューイングが操作可能になっているかどうかを確認できます。

CSQ4IVPX ジョブの概要

CSQ4IVPX は、チャネル・イニシエーターを開始し、IBM MQ DISPLAY CHINIT コマンドを発行するバッチ・ジョブです。 これは、チャネルやネットワークの定義を設定しなくても、分散キューイングの主要なすべての部分が操作可能になっているかどうかを検査します。

CSQ4IVPX の実行準備

CSQ4IVPX を実行する前に、次のことを行ってください。

1. インストール検査プログラムを実行するために必要なサンプル JCL の CSQ4IVPX は、ライブラリー thlqual.SCSQPROC に入っています。

   CSQ4IVPX JCL をカスタマイズするため、IBM MQ ライブラリーの高位修飾子、使用する各国語、4 文字のキュー・マネージャー名、およびジョブ出力の宛先を指定します。

2. RACF を更新して、IBM MQ セキュリティーが活動状態の場合に CSQ4IVPX が RACF の資源にアクセスできるようにします。 IBM MQ のセキュリティーが使用可能になったときに CSQ4IVPX を実行するには、オブジェクトへのアクセス権限を持っている RACF ユーザー ID が必要です。 リソースを RACF に定義する詳細については、z/OS でのセキュリティーのセットアップを参照してください。 IVP を実行するユーザー ID には、次のようなアクセス権が必要です。

権限	プロファイル	クラス
CONTROL	ssid.START.CHINIT および ssid.STOP.CHINIT	MQCMDS
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.CSQUTIL.*	MQQUEUE
READ	ssid.BATCH	MQCONN
READ	ssid.DISPLAY.CHINIT	MQCMDS

   上記の要件では、接続セキュリティー・プロファイル ssid.CHIN が定義されていて (チャネル・イニシエーターのための接続セキュリティー・プロファイルを参照)、しかもすべての IBM MQ セキュリティーが活動状態になっているものと想定しています。 この作業を行うための RACF コマンドを、987 ページの図 104 に示します。 この例では、次のことを前提としています。

   - キュー・マネージャー名が CSQ1 である
   - サンプルの CSQ4IVPX を実行しているユーザーのユーザー ID が TS101 である
   - チャネル・イニシエーター・アドレス・スペースは、ユーザー ID CSQ1MSTR の下で実行されている

3. RACF を更新して、チャネル・イニシエーター・アドレス・スペースに以下のアクセス権限を許可します。

権限	プロファイル	クラス
READ	ssid.CHIN	MQCONN
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.CHANNEL.INITQ	MQQUEUE
UPDATE	ssid.SYSTEM.CHANNEL.SYNCQ	MQQUEUE
ALTER	ssid.SYSTEM.CLUSTER.COMMAND.QUEUE	MQQUEUE
UPDATE	ssid.SYSTEM.CLUSTER.TRANSMIT.QUEUE	MQQUEUE
ALTER	ssid.SYSTEM.CLUSTER.REPOSITORY.QUEUE	MQQUEUE
CONTROL	ssid.CONTEXT.**	MQADMIN

   この作業を行うための RACF コマンドも、987 ページの図 104 に示します。

```
RDEFINE MQCMDS CSQ1.DISPLAY.DQM
PERMIT CSQ1.DISPLAY.DQM CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQCMDS CSQ1.START.CHINIT
PERMIT CSQ1.START.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQCMDS CSQ1.STOP.CHINIT
PERMIT CSQ1.STOP.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101,CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CSQUTIL.*
PERMIT CSQ1.SYSTEM.CSQUTIL.* CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)

RDEFINE MQCONN CSQ1.CHIN
PERMIT CSQ1.CHIN CLASS(MQCONN) ID(CSQ1MSTR) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.SYNCQ
PERMIT CSQ1.SYSTEM.CHANNEL.SYNCQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.INITQ
PERMIT CSQ1.SYSTEM.CHANNEL.INITQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQADMIN CSQ1.CONTEXT.**
PERMIT CSQ1.CONTEXT.** CLASS(MQADMIN) ID(CSQ1MSTR) ACCESS(CONTROL)
```

図 *104. CSQ4IVPX のための RACF* コマンド

CSQ4IVPX の実行

これらのステップを完了したら、キュー・マネージャーを開始してください。

IVP はバッチ・ジョブとして実行されます。 ジョブ・カードを、インストール先のジョブ実行依頼の要件に合うようにカスタマイズしてください。

CSQ4IVPX の実行結果の検査

CSQ4IVPX は CSQUTIL IBM MQ ユーティリティーを実行して、3 つの MQSC コマンドを発行します。 SYSPRINT 出力データ・セットは、988 ページの図 105 のようにしてください。 ただし、使用するキュー・マネージャーの属性によって詳細は異なります。

- コマンド **(1)** が表示され、それぞれにいくつかのメッセージが続きます。
- 各コマンドの最後のメッセージは次のようになります"CSQ9022I... NORMAL COMPLETION" **(2)**。
- 全体としてのジョブは、戻りコード 0 **(3)** で完了します。

```
CSQU000I CSQUTIL IBM MQ for z/OS - V6
CSQU001I CSQUTIL Queue Manager Utility - 2005-05-09 09:06:48
COMMAND
CSQU127I CSQUTIL Executing COMMAND using input from CSQUCMD data set
CSQU120I CSQUTIL Connecting to queue manager CSQ1
CSQU121I CSQUTIL Connected to queue manager CSQ1
CSQU055I CSQUTIL Target queue manager is CSQ1
START CHINIT
(1)
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQM138I +CSQ1 CSQMSCHI CHANNEL INITIATOR STARTING
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' START CHINIT' NORMAL COMPLETION
(2)
DISPLAY CHINIT
(1)
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMDDQM DISPLAY CHINIT COMMAND ACCEPTED
CSQN205I  COUNT=   12, RETURN=00000000, REASON=00000000
CSQX830I +CSQ1 CSQXRDQM Channel initiator active
CSQX002I +CSQ1 CSQXRDQM Queue sharing group is QSG1
CSQX831I +CSQ1 CSQXRDQM 8 adapter subtasks started, 8 requested
CSQX832I +CSQ1 CSQXRDQM 5 dispatchers started, 5 requested
CSQX833I +CSQ1 CSQXRDQM 0 SSL server subtasks started, 0 requested
CSQX840I +CSQ1 CSQXRDQM 0 channel connections current, maximum 200
CSQX841I +CSQ1 CSQXRDQM 0 channel connections active, maximum 200,
including 0 paused
CSQX842I +CSQ1 CSQXRDQM 0 channel connections starting,
0 stopped, 0 retrying
CSQX836I +CSQ1 Maximum channels - TCP/IP 200, LU 6.2 200
CSQX845I +CSQ1 CSQXRDQM TCP/IP system name is TCPIP
CSQX848I +CSQ1 CSQXRDQM TCP/IP listener INDISP=QMGR not started
CSQX848I +CSQ1 CSQXRDQM TCP/IP listener INDISP=GROUP not started
CSQX849I +CSQ1 CSQXRDQM LU 6.2 listener INDISP=QMGR not started
CSQX849I +CSQ1 CSQXRDQM LU 6.2 listener INDISP=GROUP not started
CSQ9022I +CSQ1 CSQXCRPS ' DISPLAY CHINIT' NORMAL COMPLETION
(2)
STOP CHINIT
(1)
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMTCHI STOP CHINIT COMMAND ACCEPTED
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' STOP CHINIT' NORMAL COMPLETION
(2)
CSQU057I CSQUCMDS 3 commands read
CSQU058I CSQUCMDS 3 commands issued and responses received, 0 failed
CSQU143I CSQUTIL 1 COMMAND statements attempted
CSQU144I CSQUTIL 1 COMMAND statements executed successfully
CSQU148I CSQUTIL Utility completed, return code=0
(3)
```

図 *105. CSQ4IVPX* からの出力の例

z/OS IBM MQ for z/OS を使用した C、C++ 、COBOL、PL/I、および CICS プログラムのテスト

IBM MQ で提供されているサンプル・アプリケーションを使用して、C、C++ 、COBOL、PL/I、または CICS のいずれかをテストすることができます。

IVP (CSQ4IVP1) はロード・モジュールとして提供されますが、ソース・モジュールとしてサンプルを提供します。 これらのソース・モジュールを使用して、さまざまなプログラミング言語環境をテストすることができます。

サンプル・アプリケーションについて詳しくは、IBM MQ for z/OS のサンプル・プログラムを参照してください。

z/OS z/OS での他のキュー・マネージャーとの通信のセットアップ

このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

このタスクについて

分散キューイングの要件定義では、以下の項目の定義が必要です。

- チャネル・イニシエーターのプロシージャーとデータ・セット
- チャネル定義
- キューおよびその他のオブジェクト
- アクセス・セキュリティー

キュー共用グループを使用している場合は、分散キューイングとキュー共用グループを参照してください。

IBM MQ for z/OS を使用して分散キューイングをセットアップする準備をする際に考慮すべき追加の点については、 989 ページの『z/OS での分散キューイングの使用に関する考慮事項』を参照してください。

手順

分散キューイングを使用可能にするには、以下のステップを実行します。

- 分散キューイング機能をカスタマイズし、システムオブジェクトの定義および 909 ページの『z/OS でのキュー・マネージャーのカスタマイズ準備』で説明されているように、必要な IBM MQ オブジェクトを定義します。
- アクセス・セキュリティーを定義する (z/OS で使用するチャネル・イニシエーターのセキュリティーに関する考慮事項を参照)。
- 1009 ページの『z/OS における通信のセットアップ』の説明に従って、通信をセットアップします。

関連概念

914 ページの『IBM MQ for z/OS の設定』
このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

関連タスク

190 ページの『分散キューイングの構成』
このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

z/OS z/OS での分散キューイングの使用に関する考慮事項

z/OS で分散キューイングを使用する準備をする際に考慮すべき点を示します。

キュー共用グループを使用している場合は、分散キューイングとキュー共用グループを参照してください。

オペレーター・メッセージ

チャネル・イニシエーターは、 非同期的に稼働するディスパッチャーをいくつか使用するため、 ログに表示されるオペレーター・メッセージは発生順ではないことがあります。

チャネル操作コマンド

チャネル操作コマンドには、一般に 2 つのステージが関係します。 コマンド構文を検査し、 チャネルの存在が確認されたら、 要求はチャネル・イニシエーターに送信されます。 最初のステージの完了を示すために、メッセージ CSQM134I または CSQM137I がコマンド発行者に送信されます。 チャネル・イニシエーターがコマンドを処理すると、そのコマンドが成功したかどうかを示すメッセージが、メッセージ CSQ9022I または CSQ9023E とともにコマンド発行者に送信されます。 エラー・メッセージが生成される場合には、z/OS コンソールにも送信されます。

ただし、 **DISPLAY CLUSQMGR** を除くすべてのクラスター・コマンドは非同期的に機能します。 オブジェクト属性を変更するコマンドは、 オブジェクトを更新し、要求をチャネル・イニシエーターに送信します。 クラスターを操作するコマンドは、 構文が検査され、要求がチャネル・イニシエーターに送信されます。 どちらの場合も、要求が送信されたことを示すメッセージ CSQM130I がコマンド発行者に送信されます。 このメッセージの後にメッセージ CSQ9022I が続き、要求が送信されたことで、コマンドが正常に完了し

たことを示します。 これは、クラスター要求が正常に完了したことを示すものではありません。 チャネル・イニシエーターに送信された要求は、クラスターの他のメンバーから受信したクラスター要求と共に、非同期で処理されます。 場合によっては、この要求をクラスター全体に送信して、成功したかどうかを判断する必要があります。 エラーがある場合は、チャネル・イニシエーターが稼働しているシステムの z/OS に報告されます。 コマンド発行者には送信されません。

未配布メッセージ・キュー

IBM MQ for z/OS には送達不能ハンドラーが備わっています。 詳しくは、 送達不能キュー・ハンドラー・ユーティリティー (CSQUDLQH)を参照してください。

使用中のキュー

受信側チャネルの MCA は、メッセージが伝送されていないときでも宛先キューをオープンしておくことができます。 この動作により、キューが「使用中」として見えるようになります。

セキュリティーの変更

ユーザー ID に対するセキュリティー・アクセスを変更しても、それがすぐに有効になるわけではありません。 詳しくは、「 z/OS でのチャネル・イニシエーターのセキュリティーに関する考慮事項 」、「 キュー・セキュリティーのプロファイル」、および「 928 ページの『ESM セキュリティー管理を実施する』」を参照してください。

通信の停止 - TCP

何らかの理由で TCP が停止されてから再始動される場合は、TCP ポート上で待機中の IBM MQ for z/OS TCP リスナーは停止されます。

自動チャネル再接続では、チャネル・イニシエーターが TCP/IP が使用不可であることを検知し、TCP/IP の戻り時に TCP/IP リスナーを自動的に再始動することができます。 この自動再始動によって、 TCP/IP の問題を検知してリスナーを手動で再始動するという操作スタッフの責務が軽減されます。 リスナーが動作していない間に、チャネル・イニシエーターを使用して、 LSTRTMR で指定された間隔でリスナーを再試行することもできます。 TCP/IP が戻ってリスナーが正常に再始動するまで、この試みを続けることができます。 LSTRTMR について詳しくは、ALTER QMGR および 分散キューイング・メッセージ (CSQX ...)を参照してください。

通信の停止 - LU6.2

APPC が停止した場合は、リスナーも停止します。 この場合も、リスナーは LSTRTMR 間隔で自動的に再試行するため、APPC が再始動すると、リスナーも再始動することができます。

Db2 に障害が起きた場合、既に実行中であった共有チャネルの実行は継続されますが、以後のチャネル開始要求はすべて失敗します。 Db2 が復元されたら、新規の要求を完了することができます。

z/OS 自動再始動管理 (ARM)

自動再始動管理 (ARM) は、z/OS のリカバリー機能の 1 つです。この機能を利用すると、特定のバッチ・ジョブまたは開始済みタスク (例えば、サブシステム) の可用性を向上させることができます。 これによって、生産性の高い作業をより速やかに 再開することができます。

ARM を使用するには、キュー・マネージャーとチャネル・イニシエーターが自動的に再始動されるように設定する必要があります。 詳しくは、 z/OS 自動リスタート・マネージャー (ARM) の使用を参照してください。

関連概念

914 ページの『IBM MQ for z/OS の設定』

このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

関連タスク
190 ページの『分散キューイングの構成』
このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

z/OS での IBM MQ オブジェクトの定義

z/OS では、いずれかの IBM MQ コマンド入力メソッドを使用して IBM MQ オブジェクトを定義します。

オブジェクトの定義について詳しくは、992 ページの『z/OS 上のチャネルのモニターおよび制御』を参照してください。

伝送キューおよびチャネルのトリガー操作

以下のものを定義します。

- それぞれの送信側メッセージ・チャネルごとに、用途として XMITQ を指定したローカル・キューを 1 つずつ。
- リモート・キュー定義。

  リモート・キュー・オブジェクトは、名前および内容の指定方法に応じて、3 つの個別の用途が与えられます。

  – リモート・キュー定義
  – キュー・マネージャー別名定義
  – 応答先キュー別名定義

  これらの 3 つの方法は、リモート・キュー定義オブジェクトの 3 つの使用方法に示されています。

伝送キューの TRIGDATA フィールドを使用して、指定したチャネルをトリガー操作します。 以下に例を示します。

```
DEFINE QLOCAL(MYXMITQ) USAGE(XMITQ) TRIGGER +
INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(MYCHANNEL)
DEFINE CHL(MYCHANNEL) CHLTYPE(SDR) TRPTYPE(TCP) +
XMITQ(MYXMITQ) CONNAME('9.20.9.30(1555)')
```

システムから提供されるサンプル CSQ4INYD は、必要な定義について別の例を提供しています。

z/OS 共有チャネルの同期キューが定義されている CF 構造への接続が失われるか、同種の問題が発生すると、一時的にチャネルが開始できなくなる可能性があります。 問題が解決した後、トリガー・タイプ FIRST を使用している場合にそのトリガー時にチャネルを開始できない場合、チャネルを手動で開始する必要があります。 問題が解決した後に、トリガーされたチャネルが自動的に開始されるようにする場合、キュー・マネージャーの TRIGINT 属性をデフォルト以外の値に設定することを検討してください。
TRIGINT 属性をデフォルト以外の値に設定すると、伝送キューにメッセージがある間に、チャネル・イニシエーターがチャネルの開始を定期的に再試行するようになります。

同期キュー

DQM には、シーケンス番号および作業論理単位 ID (LUWID) で使用するためのキューが必要です。 SYSTEM.CHANNEL.SYNCQ という名前のキューを利用できるようにする必要があります (計画中 z/OS を参照してください)。 このキューが使用可能になっていないと、チャネル・イニシエーターを開始できません。

このキューは、必ず INDXTYPE(MSGID) を使用して定義してください。 この属性により、このキューに対するアクセス速度が向上します。

チャネル・コマンド・キュー

使用するシステムには必ず SYSTEM.CHANNEL.INITQ という名前のチャネル・コマンド・キューを用意する必要があります。

チャネル・イニシエーターが SYSTEM.CHANNEL.INITQ に問題があることを検出すると、その問題が訂正されるまで、正常に継続することができません。 問題は、次のいずれかが原因だと考えられます。

- キューが満杯である。
- キューは書き込みができない。
- キューが入っているページ・セットが満杯である。
- 当該チャネル・イニシエーターがキューに対して正しいセキュリティー権限をもっていない。

チャネル・イニシエーターの実行中に、キューの定義が GET(DISABLED) に変更されると、チャネル・イニシエーターはそのキューからメッセージを読み取ることができなくなり、終了します。

チャネル・イニシエーターの開始

トリガー操作は、チャネル・イニシエーターによって実行されます。 IBM MQ for z/OS では、イニシエーターは MQSC コマンドの START CHINIT で開始されます。

チャネル・イニシエーターの停止

チャネル・イニシエーターは、キュー・マネージャーを停止すると自動的に停止されます。 キュー・マネージャーではなくチャネル・イニシエーターを停止する必要がある場合は、MQSC コマンド STOP CHINIT を使用できます。

z/OS z/OS 上のチャネルのモニターおよび制御

リモート・キュー・マネージャーにつながるチャネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。

各 z/OS キュー・マネージャーには、DQM プログラム (チャネル・イニシエーター) があります。 ネイティブ z/OS 機能を使用してリモート・キュー・マネージャーへの相互接続を制御します。

z/OS におけるこれらのパネルとコマンドは、操作および制御パネルと MQSC コマンドに組み込まれて実装されます。 これらのパネルとコマンドの 2 つのセットには、編成上の違いはありません。

プログラマブル・コマンド・フォーマット (PCF) コマンドを使用してコマンドを入力することもできます。これらのコマンドの使用については、管理タスクの自動化を参照してください。

このセクションの情報は、分散キューイングでチャネル・イニシエーターが使用されるすべてのケースに適用されます。 キュー共有グループを使用するかグループ内キューイングを使用するかにかかわりなく適用されます。

DQM チャネル制御機能

分散キューの管理モデルの概要については、212 ページの『メッセージの送受信』を参照してください。

チャネル制御機能は、パネル、コマンド、プログラム、同期キュー、チャネル・コマンド・キュー、およびチャネル定義で構成されます。 このトピックでは、チャネル制御機能のコンポーネントについて簡単に説明します。

- チャネル定義は、 z/OS 内の他の IBM MQ オブジェクトと同様に、ページ・セット 0 または Db2 内のオブジェクトとして保持されます。
- 操作および制御パネル、MQSC コマンド、または PCF コマンドを使用する目的は以下のとおりです。
  - チャネル定義の作成、コピー、表示、変更、および削除
  - チャネル・イニシエーターおよびリスナーの開始および停止
  - リンクを再確立できない場合の、チャネルの開始、停止、ping、チャネル・シーケンス番号のリセット、および未確定メッセージの解決
  - チャネルに関する状況情報の表示
  - DQM に関する情報の表示

特に CSQINPX 初期設定の入力データ・セットを使用して、MQSC コマンドを出すことができます。 このセットは、チャネル・イニシエーターを開始するたびに処理することができます。 詳しくは、初期化コマンドを参照してください。

- チャネルの再同期に使われる 2 つのキュー (SYSTEM.CHANNEL.SYNCQ および SYSTEM.QSG.CHANNEL.SYNCQ) があります。 パフォーマンス上の理由から、INDXTYPE(MSGID) を使用してこれらのキューを定義してください。
- チャネル・イニシエーター、チャネル、およびリスナーに対するコマンドを保持するために、チャネル・コマンド・キュー (SYSTEM.CHANNEL.INITQ) が使用されます。
- チャネル制御機能プログラムは、キュー・マネージャーとは異なる独自のアドレス・スペースで実行されます。チャネル制御機能プログラムは、チャネル・イニシエーター、リスナー、MCA、トリガー・モニターおよびコマンド・ハンドラーから構成されます。
- キュー共用グループおよび共用チャネルについては、共用キューおよびキュー共用グループを参照してください。
- グループ内キューイングについては、グループ内キューイングを参照してください

z/OS におけるチャネルの管理

チャネル、チャネル・イニシエーター、およびリスナーを管理する方法についての詳細は、以下の表のリンクを使用します。

表 *81.* チャネル操作

実行される操作	**MQSC コマンド**
チャネルの定義	DEFINE CHANNEL
チャネル定義の変更	ALTER CHANNEL
チャネル定義の表示	DISPLAY CHANNEL
チャネル定義の削除	Delete Channel
チャネル・イニシエーターの開始	START CHINIT
チャネル・イニシエーターの停止	STOP CHINIT
チャネル・イニシエーター情報の表示	DISPLAY CHINIT
チャネル・リスナーの開始	START LISTENER
チャネル・リスナーの停止	STOP LISTENER
チャネルの開始	START CHANNEL
チャネルのテスト	PING CHANNEL
チャネルのメッセージ・シーケンス番号のリセット	RESET CHANNEL
チャネルの未確定メッセージの解決	RESOLVE CHANNEL
チャネルの停止	STOP CHANNEL
チャネル状況の表示	DISPLAY CHSTATUS
クラスター・チャネルの表示	DISPLAY CLUSQMGR

関連概念

994 ページの『パネルおよびコマンドの使用法』
MQSC コマンド、PCF コマンド、または操作および制御パネルを使用して DQM を管理できます。

914 ページの『IBM MQ for z/OS の設定』
このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

1009 ページの『z/OS における通信のセットアップ』
分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。 このセクションでは、接続を定義する方法を説明します。

1014 ページの『キュー共有グループを使用する DQM のための IBM MQ for z/OS の準備』
このセクションの説明を使用して、IBM MQ for z/OS で、キュー共有グループを使用した分散キューイングを構成します。

1018 ページの『キュー共有グループを使用する IBM MQ for z/OS の通信のセットアップ』
分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 この試行を正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。

関連タスク
190 ページの『分散キューイングの構成』
このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

988 ページの『z/OS での他のキュー・マネージャーとの通信のセットアップ』
このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

z/OS パネルおよびコマンドの使用法

MQSC コマンド、PCF コマンド、または操作および制御パネルを使用して DQM を管理できます。

MQSC コマンドについて詳しくは、 MQSC コマンドを使用した IBM MQ の管理を参照してください。 PCF コマンドについては、プログラマブル・コマンド・フォーマットのコマンドによる管理の自動化を参照してください。

初期パネルの使用

操作および制御パネルの呼び出し、ファンクション・キーの使用、およびヘルプの表示を行う方法の概要については、管理 IBM MQ for z/OS を参照してください。

注 : 操作および制御パネルを使用するには、正しいセキュリティー許可が必要です。詳しくは、IBM MQ for z/OS の管理およびサブトピックを参照してください。 994 ページの図 106 は、パネル・セッションを開始したときに表示されるパネルを示しています。 パネルの後のテキストでは、このパネルで実行するアクションを説明します。

```
IBM MQ for z/OS - Main Menu

Complete fields. Then press Enter.

Action . . . . . . . . . . 1   0. List with filter  4. Manage
1. List or Display  5. Perform
2. Define like    6. Start
3. Alter       7. Stop
8. Command
Object type . . . . . . . . CHANNEL    +
Name . . . . . . . . . . . . *
Disposition . . . . . . . . A Q=Qmgr, C=Copy, P=Private, G=Group,
S=Shared, A=All

Connect name . . . . . . . MQ25 - local queue manager or group
Target queue manager . . . MQ25
- connected or remote queue manager for command input
Action queue manager . . . MQ25 - command scope in group
Response wait time . . . . 10  5 - 999 seconds

(C) Copyright IBM Corporation 1993, 2025. All rights reserved.

Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F4=Prompt  F9=SwapNext F10=Messages
F12=Cancel
```

図 *106.* 操作および制御初期パネル

このパネルから以下のことを行うことができます。

- 「**Action (アクション)**」フィールドに適切な番号を入力することによって、実行したいアクションを選択します。
- 作業を進めたいオブジェクト・タイプを指定します。 オブジェクト・タイプが分からない場合は、F4 キーを押してオブジェクト・タイプのリストを表示します。
- 指定されたタイプのオブジェクトのリストを表示します。 「**Name (名前)**」フィールドにアスタリスク (*) を入力し、Enter キーを押して、このサブシステムですでに定義されているオブジェクト (指定したタイプ) のリストを表示します。 次に、1 つまたは複数のオブジェクトを選択して、それらを順次に処理できます。 995 ページの図 107 は、この方法で生成されたチャネルのリストを示しています。
- 処理したいオブジェクトのキュー共有グループにおける属性指定を「**Disposition (属性指定)**」フィールドに指定します。 オブジェクトの保管先とオブジェクトの動作は、属性指定によって決まります。
- 接続先にしたいローカル・キュー・マネージャーまたはキュー共有グループを「**Connect name (接続名)**」フィールドで選びます。 リモート・キュー・マネージャーでコマンドを出したければ、そのリモート・キュー・マネージャーがキュー共有グループのメンバーであるかどうかに応じて、「**Target queue manager (宛先 QMGR)**」フィールドまたは「**Action queue manager (アクション QMGR)**」フィールドを選びます。 リモート・キュー・マネージャーがキュー共有グループのメンバーでなければ、「**Target queue manager (宛先 QMGR)**」フィールドを選びます。 リモート・キュー・マネージャーがキュー共有グループのメンバーであれば、「**Action queue manager (アクション QMGR)**」フィールドを選びます。
- 応答を受信するまでの待ち時間を「**Response wait time (応答待ち時間)**」フィールドで選びます。

```
List Channels - MQ25          Row 1 of 8

Type action codes, then press Enter. Press F11 to display connection status.
1=Display  2=Define like  3=Alter  4=Manage  5=Perform
6=Start   7=Stop

Name         Type     Disposition  Status
<> *           CHANNEL    ALL   MQ25
_  SYSTEM.DEF.CLNTCONN  CLNTCONN   QMGR  MQ25
_  SYSTEM.DEF.CLUSRCVR  CLUSRCVR   QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.CLUSSDR  CLUSSDR    QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.RECEIVER  RECEIVER   QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.REQUESTER REQUESTER   QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.SENDER   SENDER    QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.SERVER   SERVER    QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.SVRCONN  SVRCONN    QMGR  MQ25 INACTIVE
******** End of list ********



Command ===> ____________________________________________________________________
F1=Help   F2=Split   F3=Exit   F4=Filter  F5=Refresh  F7=Bkwd
F8=Fwd    F9=SwapNext F10=Messages F11=Status  F12=Cancel
```

図 *107.* チャネルのリスト

z/OS *z/OS でのチャネルの定義*

z/OS では、MQSC コマンドを使用するか、操作パネルおよび制御パネルを使用して、チャネルを定義できます。

手順

- MQSC コマンドを使用してチャネルを定義するには、 **DEFINE CHANNEL** コマンドを使用します。
- 初期パネルから操作および制御パネルを使用するには、以下のフィールドに入力して Enter キーを押します。

表 82. 操作および制御パネル: 初期パネル・フィールド	
フィールド	**フィールドに入力する値**
アクション	2 (類似定義)
オブジェクト・タイプ	チャネル・タイプ (SENDER など) または CHANNEL
名前	
後処理	新規オブジェクトのロケーション

定義するチャネルに必要な名前と属性に関する情報を入力するためのパネルがいくつか表示されます。 それらは、デフォルトの属性値で初期化されています。 実行キーを押す前に, 必要なものを変更してください。

注 :「 **object type (オブジェクト・タイプ)** 」フィールドに CHANNEL を入力した場合は、最初に「 **Select a Valid Channel Type (有効なチャネル・タイプの選択)** 」パネルが表示されます。

既存のチャネルと同じ属性を持つチャネルを定義する場合は、コピーするチャネルの名前を初期パネルの「 **Name** 」フィールドに入力します。 パネルは、既存オブジェクトの属性で初期化されます。

チャネル属性の詳細は、チャネル属性を参照してください。

注 :

1. ネットワーク内のすべてのチャネルに固有の名前を付けてください。 このネーミングのためには、すべてのチャネルを示すネットワーク・ダイアグラム に示すように、ソースおよびターゲットのキュー・マネージャーの名前をチャネル名に取り入れるのはよい方法です。

次のタスク

チャネルを定義した後、チャネルを保護する必要があります。 詳細については、998 ページの『チャネルの保護』 を参照してください。

z/OS チャネル定義の変更

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネル定義を変更できます。

MQSC コマンドを使用してチャネル定義を変更するには、ALTER CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	**値**
アクション	3 (変更)
オブジェクト・タイプ	チャネル・タイプ (SENDER など) または CHANNEL
名前	CHANNEL.TO.ALTER
後処理	保管されているオブジェクトのロケーション

チャネルの現行の属性に関する情報が入ったパネルがいくつか表示されます。 必要に応じて、無保護フィールドを新しい値で指定変更し、Enter キーを押して、チャネル定義を変更してください。

チャネル属性の詳細は、チャネル属性を参照してください。

z/OS チャネル定義の表示

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネル定義を表示できます。

MQSC コマンドを使用してチャネル定義を表示するには、DISPLAY CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	1 (リストまたは表示)
オブジェクト・タイプ	チャネル・タイプ (SENDER など) または CHANNEL
名前	CHANNEL.TO.DISPLAY
後処理	オブジェクトのロケーション

チャネルの現行の属性に関する情報を示すパネルがいくつか表示されます。

チャネル属性の詳細は、チャネル属性を参照してください。

z/OS チャネル定義の削除

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネル定義を削除できます。

MQSC コマンドを使用してチャネル定義を削除するには、DELETE CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	4 (管理)
オブジェクト・タイプ	チャネル・タイプ (SENDER など) または CHANNEL
名前	CHANNEL.TO.DELETE
後処理	オブジェクトのロケーション

別のパネルが表示されます。 このパネルで機能タイプ 1 を選択します。

Enter キーを押してチャネル定義を削除しようとすると、チャネル定義を削除したいことを確認するためにもう一度 Enter キーを押すよう指示されます。

注 : チャネル定義を削除できるようにするためには、チャネル・イニシエーターを実行させておく必要があります (クライアント接続チャネルを除く)。

z/OS チャネル・イニシエーターに関する情報の表示

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネル・イニシエーターに関する情報を表示できます。

MQSC コマンドを使用して、チャネル・イニシエーターに関する情報を表示するには、DISPLAY CHINIT を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	1 (表示)
オブジェクト・タイプ	SYSTEM
名前	ブランク

別のパネルが表示されます。 このパネルで機能タイプ 1 を選択します。

注 :

1. 多数のチャネルがあると、分散キューイング情報の表示に時間がかかることがあります。

2. 分散キューイングの情報を表示するには、先にチャネル・イニシエーターを実行しておく必要があります。

z/OS チャネルの保護

MQSC コマンドを使用して、または操作パネルと制御パネルを使用して、チャネルを保護できます。

MQSC コマンドを使用してチャネルを保護するには、SET CHLAUTH を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	8

MQSC コマンドを提供できるエディターが表示されます。この場合は、CHLAUTH コマンドを使用してください。998 ページの図 108 を参照してください。 コマンド入力の最後には正符号 (+) が必要です。 PF3 と入力してこのエディターを終了し、コマンドをコマンド・サーバーに実行依頼します。

```
****************************** Top of Data ******************************
000001 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000002 TYPE(SSLPEERMAP) +
000003 SSLPEER('CN="John Smith"') +
000004 MCAUSER('PUBLIC')
****** **************************** Bottom of Data ****************************

Command ===>                          Scroll ===> PAGE
F1=Help   F3=Exit   F4=LineEdit F12=Cancel
```

図 *108.* コマンド入力

次に、コマンドの出力が表示されます。998 ページの図 109 を参照してください

```
****** ***************************** Top of Data ******************************
000001 CSQU000I CSQUTIL IBM MQ for z/OS V7.1.0
000002 CSQU001I CSQUTIL Queue Manager Utility - 2011-04-20 14:42:58
000003 COMMAND TGTQMGR(MQ23) RESPTIME(30)
000004 CSQU127I Executing COMMAND using input from CSQUCMD data set
000005 CSQU120I Connecting to MQ23
000006 CSQU121I Connected to queue manager MQ23
000007 CSQU055I Target queue manager is MQ23
000008 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000009 TYPE(SSLPEERMAP) +
000010 SSLPEER('CN="John Smith"') +
000011 MCAUSER('PUBLIC')
000012 CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000000
000013 CSQ9022I !MQ23 CSQMSCA ' SET CHLAUTH' NORMAL COMPLETION
000014 CSQU057I 1 commands read
000015 CSQU058I 1 commands issued and responses received, 0 failed
000016 CSQU143I 1 COMMAND statements attempted
000017 CSQU144I 1 COMMAND statements executed successfully
000018 CSQU148I CSQUTIL Utility completed, return code=0
Command ===>                          Scroll ===> PAGE
F1=Help   F3=Exit   F5=Rfind   F6=Rchange  F9=SwapNext F12=Cancel
```

図 *109.* コマンド出力

z/OS チャネル・イニシエーターの開始

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネル・イニシエーターを開始できます。

MQSC コマンドを使用してチャネル・イニシエーターを開始するには、START CHINIT を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	6 (開始)
オブジェクト・タイプ	SYSTEM
名前	ブランク

「Start a System Function) (システム機能の開始)」パネルが表示されます。 以下のパネルの後のテキストでは、実行するアクションについて説明します。

```
Start a System Function

Select function type, complete fields, then press Enter to start system
function.

Function type . . . . . _   1. Channel initiator
2. Channel listener
Action queue manager . . . : MQ25

Channel initiator
JCL substitution . . . . . ________________________________________________
------------------------------------------------

Channel listener
Inbound disposition . . . Q G=Group, Q=Qmgr
Transport type . . . . . _ L=LU6.2, T=TCP/IP
LU name (LU6.2) . . . . . ________________
Port number (TCP/IP) . . . 1414
IP address (TCP/IP) . . . ________________________________________________



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

図 *110.* システム機能の開始

「Function type (機能タイプ)」の「1」(チャネル・イニシエーター) を選択して、Enter キーを押します。

z/OS チャネル・イニシエーターの停止

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネル・イニシエーターを停止できます。

MQSC コマンドを使用してチャネル・イニシエーターを停止するには、STOP CHINIT を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	7 (停止)
オブジェクト・タイプ	SYSTEM
名前	ブランク

「Stop a System Function (システム機能の停止)」パネルが表示されます。 パネルの後のテキストでは、このパネルの使用方法を説明します。

```
Stop a System Function

Select function type, complete fields, then press Enter to stop system
function.

Function type . . . . . _   1. Channel initiator
2. Channel listener
Action queue manager . . . : MQ25

Channel initiator
Restart shared channels  Y Y=Yes, N=No

Channel listener
Inbound disposition . . . Q G=Group, Q=Qmgr
Transport type . . . . . _ L=LU6.2, T=TCP/IP

Port number (TCP/IP) . . . _____
IP address (TCP/IP) . . . ________________________________________________



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

図 *111.* 機能制御の停止

「Function type (機能タイプ)」の「1」(チャネル・イニシエーター) を選択して、Enter キーを押します。

チャネル・イニシエーターは、実行中のすべてのチャネルが静止モードで停止するのを待ってから、停止します。

注 : チャネルのなかに実行されているがアクティブになっていない受信側チャネルまたは要求側チャネルがある場合、受信側または送信側のチャネル・イニシエーターに出された停止要求によって、そのチャネルは即座に停止されます。

ただし、次のようなメッセージが表示される場合には、チャネル・イニシエーターは現行のバッチ・メッセージが完了するのを待ってから、そのチャネルを停止します。

z/OS チャネル・リスナーの開始

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネル・リスナーを開始できます。

MQSC コマンドを使用してチャネル・リスナーを開始するには、START LISTENER を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	6 (開始)
オブジェクト・タイプ	SYSTEM
名前	ブランク

システム機能の開始のパネルが表示されます(999 ページの図 110 を参照)。

機能タイプ 2 (チャネル・リスナー) を選択します。 Inbound disposition (到着属性指定) を選択します。 Transport type (トランスポート・タイプ) を選択します。 トランスポート・タイプが L の場合、LU 名を選択します。 トランスポート・タイプが T の場合、ポート番号と (オプションで) IP アドレスを選択します。 Enter キーを押します。

注 : TCP/IP リスナーの場合、ポートと IP アドレスの複数の組み合わせを開始することができます。

z/OS チャネル・リスナーの停止

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネル・リスナーを停止できます。

MQSC コマンドを使用してチャネル・リスナーを停止するには、STOP LISTENER を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	7 (停止)
オブジェクト・タイプ	SYSTEM
名前	ブランク

「システム機能の停止」パネルが表示されます(1000 ページの図 111 を参照)。

機能タイプ 2 (チャネル・リスナー) を選択します。 Inbound disposition (到着属性指定) を選択します。 Transport type (トランスポート・タイプ) を選択します。 トランスポート・タイプが T の場合、ポート番号と (オプションで) IP アドレスを選択します。 Enter キーを押します。

注 : TCP/IP リスナーの場合、ポートと IP アドレスの組み合わせを個別に停止したり すべての組み合わせを停止したりすることができます。

z/OS チャネルの開始

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネルを開始できます。

MQSC コマンドを使用してチャネルを開始するには、START CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	6 (開始)
オブジェクト・タイプ	チャネル・タイプ (SENDER など) または CHANNEL
名前	CHANNEL.TO.USE
後処理	オブジェクトの属性指定

「Start a Channel (チャネルの開始)」パネルが表示されます。 パネルの後のテキストでは、パネルの使用方法を説明します。

```
Start a Channel

Select disposition, then press Enter to start channel.


Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE


Disposition . . . . . P   P=Private on MQ25
S=Shared on MQ25
A=Shared on any queue manager









Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

図 *112.* チャネルの開始

チャネル・インスタンスの属性指定と、どのキュー・マネージャーでそれを開始するかを選択します。

Enter キーを押してチャネルを開始させます。

z/OS 共有チャネルの開始

共用チャネルを開始し、指定されたチャネル・イニシエーター上でそれを保持するには、属性 = S を使用します (START CHANNEL コマンドで、CHLDISP(FIXSHARED) を指定します)。

一度に実行される共有チャネルのインスタンスは 1 つのみです。 2 つ目のチャネル・インスタンスを開始しようとしても失敗します。

この方法でチャネルを開始するときには、以下の規則がそのチャネルに適用されます。

- キュー共有グループ内のどのキュー・マネージャーからでもチャネルを停止することができます。 チャネルの停止要求の発行時点に、開始元のチャネル・イニシエーターが実行されていなかった場合でも停止することができます。 チャネルが停止したとき、同じチャネル・イニシエーターまたは別のチャネル・イニシエーターで属性 = S (CHLDISP(FIXSHARED)) を指定してチャネルを再始動することができます。 属性 = A (CHLDISP(SHARED)) を指定してチャネルを開始することもできます。
- STARTING 状態または RETRY 状態のチャネルは、同じチャネル・イニシエーターまたは別のチャネル・イニシエーターで属性 = S (CHLDISP(FIXSHARED)) を指定して再始動することができます。 属性 = A (CHLDISP(SHARED)) を指定してチャネルを開始することもできます。
- チャネルが非アクティブ状態になるとき、チャネルがトリガーを開始することは適切です。 トリガーを開始する共有チャネルは、常に、共有属性 (CHLDISP(SHARED)) を持っています。
- チャネルが非アクティブ状態になるとき、チャネル・イニシエーターで CHLDISP(FIXSHARED) を指定してチャネルを開始することは適切です。 属性 = A (CHLDISP(SHARED)) を指定してチャネルを開始することもできます。
- チャネルが開始されたチャネル・イニシエーターが SHARED(RESTART)で停止されたとき、またはチャネル・イニシエーターが異常終了したときに、チャネルがキュー共用グループ内の他のアクティブなチャネル・イニシエーターによってリカバリーされない。 チャネルがリカバリーされるのは、チャネルが開始されたチャネル・イニシエーターが次に再始動されたときだけです。 これにより、キュー共有グループ内の他のチャネル・イニシエーターには、失敗したチャネルをリカバリーしようとすることによって生じる作業負荷がかかりません。

z/OS チャネルのテスト

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネルをテストできます。

MQSC コマンドを使用してチャネルをテストするには、PING CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	5 (実行)
オブジェクト・タイプ	SENDER、SERVER、または CHANNEL
名前	CHANNEL.TO.USE
後処理	チャネル・オブジェクトの属性指定

「Perform a Channel Function (チャネル機能の実行)」パネルが表示されます。 パネルの後のテキストでは、パネルの使用方法を説明します。

```
Perform a Channel Function

Select function type, complete fields, then press Enter.


Function type . . . . . _   1. Reset  3. Resolve with commit
2. Ping  4. Resolve with backout

Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE


Disposition . . . . . P   P=Private on MQ25
S=Shared on MQ25
A=Shared on any queue manager

Sequence number for reset . . 1     1 - 999999999
Data length for ping . . . . 16   16 - 32768



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

図 *113.* チャネルのテスト

機能タイプ 2 (Ping) を選択します。

テストの対象のチャネルの属性指定と、どのキュー・マネージャーでそれをテストするかを選択します。

データ長は最初は 16 に設定されます。 必要があればその長さを変更してから Enter キーを押します。

z/OS チャネルのメッセージ・シーケンス番号のリセット

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネルのメッセージ・シーケンス番号をリセットできます。

MQSC コマンドを使用してチャネル・シーケンス番号をリセットするには、RESET CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	5 (実行)
オブジェクト・タイプ	チャネル・タイプ (SENDER など) または CHANNEL

フィールド	値
名前	CHANNEL.TO.USE
後処理	チャネル・オブジェクトの属性指定

「チャネル機能の実行」パネルが表示されます(1003 ページの図 113 を参照)。

機能タイプ 2 (リセット) を選択します。

リセットの対象のチャネルの属性指定と、どのキュー・マネージャーでそれをリセットするかを選択します。

「**sequence number (シーケンス番号)**」フィールドは初めは 1 に設定されています。 必要があればその値を変更してから Enter キーを押します。

z/OS チャネルの未確定メッセージの解決

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネルの未確定メッセージを解決できます。

MQSC コマンドを使用するチャネルの未確定メッセージを解決するには、RESOLVE CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	5 (実行)
オブジェクト・タイプ	SENDER、SERVER、または CHANNEL
名前	CHANNEL.TO.USE
後処理	オブジェクトの属性指定

「チャネル機能の実行」パネルが表示されます(1003 ページの図 113 を参照)。

「Function type (機能タイプ)」の「3」または「4」(コミットまたはバックアウトで解決) を選択します。(詳しくは、231 ページの『未確定チャネルの処理』を参照してください。)

解決対象のチャネルの属性指定と、どのキュー・マネージャーでそれを解決するかを選択します。 Enter キーを押します。

z/OS チャネルの停止

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネルを停止できます。

MQSC コマンドを使用してチャネルを停止するには、STOP CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	7 (停止)
オブジェクト・タイプ	チャネル・タイプ (SENDER など) または CHANNEL
名前	CHANNEL.TO.USE
後処理	オブジェクトの属性指定

「Stop a Channel (チャネルの停止)」パネルが表示されます。 パネルの後のテキストでは、パネルの使用方法を説明します。

```
Stop a Channel

Complete fields, then press Enter to stop channel.


Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE


Disposition . . . . . P   P=Private on MQ25
A=Shared on any queue manager

Stop mode . . . . . 1   1. Quiesce  2. Force
Stop status . . . . . 1   1. Stopped  2. Inactive

Queue manager . . . . . _________________________________________________
Connection name . . . . . _________________________________________________



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

図 *114.* チャネルの停止

停止の対象のチャネルの属性指定と、どのキュー・マネージャーでそれを停止するかを選択します。

必要な停止モードを選択します。

静止
バッチ・サイズの値がまだ上限に達しておらず、伝送キューでは既にメッセージが待ち状態になっていても、現行メッセージが完了し、次にバッチが終了すると、チャネルが停止します。 新しいバッチは開始されません。 このモードはデフォルトです。

強制
チャネルは即時に停止します。 メッセージのバッチ化が進行中である場合は、「未確定」状態の結果が生じることがあります。

停止するチャネルのキュー・マネージャーと接続名を選択します。

必要な状況を選択します。

停止
チャネルは自動的に再開しないので、手動で再開する必要があります。 このモードは、キュー・マネージャーまたは接続名を指定しない場合のデフォルトです。 名前を指定する場合、このモードは許可されません。

非アクティブ
チャネルは必要なときに自動的に再開されます。 このモードは、キュー・マネージャーまたは接続名を指定する場合のデフォルトです。

Enter キーを押してチャネルを停止させます。

詳しくは、229 ページの『チャネルの停止および静止』 を参照してください。 停止したチャネルの再始動に関する情報については、231 ページの『停止したチャネルの再始動』 を参照してください。

注 : 共有チャネルが再試行状態になっていて、開始元のチャネル・イニシエーターが実行されていない場合は、コマンドの入力元のキュー・マネージャーでそのチャネルの停止要求を出します。

z/OS チャネル状況の表示

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネル状況を表示できます。

MQSC コマンドを使用してチャネル状況または一連のチャネルを表示するには、DISPLAY CHSTATUS を使用します。

注 : 多数のチャネルがある場合は、チャネル状況情報の表示にはある程度時間がかかります。

「リスト・チャネル」パネル(995 ページの図 107 を参照)で操作および制御パネルを使用すると、チャネル状況の要約がチャネルごとに次のように表示されます:

INACTIVE	アクティブの接続がありません。
status	アクティブの接続が 1 つあります。
nnn status	複数の現行接続があり、すべてが同じ状況です。
nnn CURRENT	複数の現行接続がありますが、すべてが同じ状況であるわけではありません。
ブランク	IBM MQ は、アクティブの接続数を判断することができません (チャネル・イニシエーターが実行中でないなどの理由)。 **注 :** GROUP の属性指定を指定されたチャネル・オブジェクトには、状況は示されません。

ここで、*nnn* はアクティブの接続数を表し、*status* は次のいずれかです。

INIT	INITIALIZING
BIND	BINDING
START	STARTING
RUN	実行中
STOP	STOPPING または STOPPED
RETRY	RETRYING
REQST	REQUESTING

チャネル状況に関する詳細情報を表示するには、「リスト・チャネル」または「表示」、または「チャネルの変更」パネルで「状況」キー(F11)を押して、「リスト・チャネル-現在の状況」パネルを表示します(1006 ページの図 115 を参照)。

```
List Channels - Current Status - MQ25      Row 1 of 16

Type action codes, then press Enter. Press F11 to display saved status.
1=Display current status

Channel name     Connection name                 State
Start time      Messages Last message time  Type   Disposition
<> *                          CHANNEL  ALL   MQ25

_ RMA0.CIRCUIT.ACL.F  RMA1                        STOP
_  2005-03-21 10.22.36 557735  2005-03-24 09.51.11 SENDER  PRIVATE MQ25
_ RMA0.CIRCUIT.ACL.N  RMA1
_  2005-03-21 10.23.09 378675  2005-03-24 09.51.10 SENDER  PRIVATE MQ25
_ RMA0.CIRCUIT.CL.F  RMA2
_  2005-03-24 01.12.51 45544   2005-03-24 09.51.08 SENDER  PRIVATE MQ25
_ RMA0.CIRCUIT.CL.N  RMA2
_  2005-03-24 01.13.55 45560   2005-03-24 09.51.11 SENDER  PRIVATE MQ25
_ RMA1.CIRCUIT.CL.F  RMA1
_  2005-03-21 10.24.12 360757  2005-03-24 09.51.11 RECEIVER PRIVATE MQ25
_ RMA1.CIRCUIT.CL.N  RMA1
_  2005-03-21 10.23.40 302870  2005-03-24 09.51.09 RECEIVER PRIVATE MQ25
******** End of list ********
Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F4=Filter  F5=Refresh  F7=Bkwd
F8=Fwd    F9=SwapNext F10=Messages F11=Saved  F12=Cancel
```

図 *115.* チャネル接続のリスト

状況の値は次のとおりです。

INIT	INITIALIZING

BIND	BINDING
START	STARTING
RUN	実行中
STOP	STOPPING または STOPPED
RETRY	RETRYING
REQST	REQUESTING
DOUBT	STOPPED および INDOUBT(YES)

詳しくは、 222 ページの『チャネルの状態』 を参照してください。

F11 キーを押して、同様のチャネル接続リストと保管された状況を表示することができます。F11 キーを再度押すと、現行リストに戻ります。 保存された状況は、メッセージの少なくとも 1 つのバッチがチャネルに伝送されるまでは適用されません。

アクション・コード「1」またはスラッシュ (/) を使用して接続を選択し、Enter キーを押します。「Display Channel Connection Current Status (チャネル接続の現在の状況の表示)」パネルが表示されます。

z/OS クラスター・チャネルの表示

MQSC コマンドを使用して、または操作および制御パネルを使用して、クラスター・チャネルを表示できます。

明示的に定義されているかまたは自動定義を使用して定義されているすべてのクラスター・チャネルを表示するには、MQSC コマンド DISPLAY CLUSQMGR を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	**値**
アクション	1 (リストまたは表示)
オブジェクト・タイプ	CLUSCHL
名前	*

図 1008 ページの図 116 のようなパネルが表示されます。このパネルには、各クラスター・チャネルの情報が三行を占め、チャネル、クラスター、およびキュー・マネージャー名が含まれています。 クラスター送信側チャネルについては、全体的な状態が表示されます。

```
List Cluster queue manager Channels - MQ25    Row 1 of 9

Type action codes, then press Enter. Press F11 to display connection status.
1=Display  5=Perform  6=Start  7=Stop

Channel name     Connection name                 State
Type     Cluster name                   Suspended
Cluster queue manager name            Disposition
<>   *                         -    MQ25
_ TO.MQ90.T      HURSLEY.MACH90.COM(1590)
_  CLUSRCVR   VJH01T                      N
_   MQ90                      -    MQ25
_ TO.MQ95.T      HURSLEY.MACH95.COM(1595)             RUN
_  CLUSSDRA   VJH01T                      N
_   MQ95                      -    MQ25
_ TO.MQ96.T      HURSLEY.MACH96.COM(1596)             RUN
_  CLUSSDRB   VJH01T                      N
_   MQ96                      -    MQ25
******** End of list ********


Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F4=Filter  F5=Refresh  F7=Bkwd
F8=Fwd    F9=SwapNext F10=Messages F11=Status  F12=Cancel
```

図 *116.* クラスター・チャネルのリスト

1 つまたは複数のチャネルに関する完全な情報を表示するには、チャネル名の前にアクション・コード「1」を入力して、Enter キーを押します。 機能 (ping、解決、リセットなど) の実行、クラスター・チャネルの開始または停止を行うには、アクション・コード「5」、「6」、または「7」を使用します。

チャネル状況の詳細を表示するには、状況キー (F11) を押します。

z/OS *zEnterpriseData Compression Express* 機能を使用するため *IBM MQ for z/OS* の準備

zEnterprise® Data Compression (zEDC) Express 機能は、最小 z/OS レベル z/OS 2.1 を使用して、IBM zEC12 GA2 以降の IBM Z マシンの特定のモデルで使用可能です。

詳しくは、「zEnterprise Data Compression (zEDC) 」を参照してください。

前提条件

IBM z15 以降では、zEnterprise Data Compression (zEDC) Express 機能は、Integrated Accelerator for zEDC としてオンチップとなる、ハードウェア・システムの PCIe I/O ドロワーのオプション・フィーチャーから移動されました。 この変更により、構成の前提条件が更新され、ご使用のハードウェア・システムに依存するようになります。

IBM z15 以降

ご使用の z/OS のレベルに応じて、以下のいずれかの PTF を適用します。

- z/OS 2.4: UJ00636
- z/OS 2.3: UJ00635
- z/OS 2.2: UJ00638
- z/OS 2.1: UJ00639

z15 以降のシステムでは、ハードウェア要件はありません。 これらのシステムにおける Integrated Accelerator for zEDC ソリューションは組み込みデータ・アクセラレーションを提供するため、別のアダプターは必要ありません。

IBM zEC12 GA2 から IBM z14

ご使用のシステムに以下の要件が必要です。

- zEDC Express® アダプター。ハードウェア・システムの PCIe I/O ドロワーに取り付けられます。
- zEDC ソフトウェア機能 (オプションの有料機能) は、IFAPRDxx parmlib メンバーで使用可能にする必要があります。

手順

IBM zEC12 GA2 から IBM z14

チャネル・イニシエーターのユーザー ID に、RACF FACILITY CLASS の FPZ.ACCELERATOR.COMPRESSION プロファイルに対する READ 権限か、または企業が使用している外部セキュリティー・マネージャー (ESM) におけるそれと同等のものに対する READ 権限が付与されていることを確認します。

重要 : IBM z15 以降では必要ありません。

IBM zEnterprise zEC12 GA2 以降

送信側および受信側の両方で、COMPMSG(ZLIBFAST) を指定してチャネルを構成します。 構成が済むと、zlib 圧縮を使用して、チャネル全体を流れるメッセージの圧縮および圧縮解除が行われます。

圧縮するデータのサイズが最小しきい値を超えると、zEDC で圧縮が実行されます。 しきい値は、使用されている IBM z ハードウェアによって異なります。

- IBM zEC12 GA2 から IBM z14 への最小しきい値 4KB
- IBM z15 以降の最小しきい値は 1KB

しきい値サイズを下回るメッセージの場合、圧縮またはインフレーションはソフトウェアで実行されます。

z/OS z/OS における通信のセットアップ

分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。 このセクションでは、接続を定義する方法を説明します。

DQM は IBM MQ 用のリモート・キューイング機能です。 これはキュー・マネージャーにチャネル制御プログラムを提供し、このプログラムによって、通信リンクに対するインターフェースが形成されます。 これらのリンクは、システム・オペレーターによって制御可能です。 分散キューイング管理によって維持されるチャネル定義は、これらの接続を使用します。

z/OS で使用できる次の 2 つの形式の通信プロトコルから、いずれかを選択します。

- 1010 ページの『z/OS での TCP 接続の定義』
- 1012 ページの『APPC/MVS を使用した z/OS 用の LU6.2 接続の定義』

MQ Adv. CD TCP/IP を使用するメッセージ・チャネルは、ネットワーク・スループットを大幅に向上させることができる高速 TCP/IP トンネルを提供する IBM Aspera faspio Gateway を指すことができます。 資格のあるプラットフォームで実行されているキュー・マネージャーは、Aspera gateway を介して接続できます。 ゲートウェイ自体は、 Red Hat または Ubuntu Linux、あるいは Windows にデプロイされます。 Linux または Windows での Aspera gateway 接続の定義を参照してください。

どのチャネル定義も伝送プロトコル (トランスポート・タイプ) 属性を 1 つだけ指定する必要があります。 1 つのキュー・マネージャーで、複数のプロトコルを通信のために使用できます。

また、Example configuration -IBM MQ for z/OS を参照すると役立つ場合もあります。キュー共用グループを使用している場合は、1018 ページの『キュー共有グループを使用する IBM MQ for z/OS の通信のセットアップ』 を参照してください。

関連概念

994 ページの『パネルおよびコマンドの使用法』
MQSC コマンド、PCF コマンド、または操作および制御パネルを使用して DQM を管理できます。

914 ページの『IBM MQ for z/OS の設定』

このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

992 ページの『z/OS 上のチャネルのモニターおよび制御』
リモート・キュー・マネージャーにつながるチャネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。

1014 ページの『キュー共有グループを使用する DQM のための IBM MQ for z/OS の準備』
このセクションの説明を使用して、IBM MQ for z/OS で、キュー共有グループを使用した分散キューイングを構成します。

1018 ページの『キュー共有グループを使用する IBM MQ for z/OS の通信のセットアップ』
分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 この試行を正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。

関連タスク

190 ページの『分散キューイングの構成』
このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

988 ページの『z/OS での他のキュー・マネージャーとの通信のセットアップ』
このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

z/OS *z/OS での TCP 接続の定義*

TCP 接続を定義するために、構成するいくつかの設定があります。

TCP アドレス・スペースの名前は、TCP システム・パラメーター・データ・セット *tcpip*.TCPIP.DATA 内に指定する必要があります。 データ・セットには、 "TCPIPJOBNAME *TCPIP_proc*" ステートメントを組み込む必要があります。

ファイアウォールを使用している場合、チャネル・イニシエーターからチャネルのアドレス、およびリモート接続からキュー・マネージャーへの allow 接続を構成する必要があります。

通常、ファイアウォールの定義により、送信側の IP アドレスとポートから宛先の IP アドレスとポートが構成されます。

- z/OS イメージには複数のホスト名を含めることができるので、ファイアウォールのソース・アドレスとして複数のホストのアドレスを構成する必要が生じることもあります。

  NETSTAT HOME コマンドを使用して、これらの名前とアドレスを表示できます。
- チャネル・イニシエーターでは異なるポートで複数のリスナーを指定できるため、それらのポートを構成する必要があります。
- キュー共用グループに対して共用ポートを使用する場合、共用ポートの構成も必要です。

チャネル・イニシエーターのアドレス・スペースには、データ・セットに対する読み取り権限が必要です。使用している TCP/IP 製品とインターフェースに応じて、次の方法で TCPIP.DATA にアクセスすることができます。

- 環境変数 RESOLVER_CONFIG
- ファイル・システム上の /etc/resolv.conf
- //SYSTCPD DD ステートメント
- //SYSTCPDD DD ステートメント
- *jobname/userid*.TCPIP.DATA
- SYS1.TCPPARMS(TCPDATA)
- *zapname*.TCPIP.DATA

また、TCP/IP に正しい高位修飾子を指定する場合にも注意が必要です。

ドメイン・ネーム・システム (DNS) サーバーを適切に構成し、名前から IP アドレスへの変換と IP アドレスから名前への変換の両方を実行できるようにする必要があります。

注 : リゾルバーの構成に対する変更のいくつかは、例えば IBM MQ など、それを使用するアプリケーションのリサイクルを必要とします。

詳細については、次の資料を参照してください。

- 基本 TCP/IP システム
- z/OS UNIX System Services.

TCP チャネルはいずれも開始されると、TCP リソースを使用します。 PROFILE.TCPIP 構成データ・セットの次のパラメーターを調整することが必要な場合もあります。

ACBPOOLSIZE
開始された TCP チャネルごとに 1 を加え、さらに 1 を加える。

CCBPOOLSIZE
開始した TCP チャネルごとに 1、DQM ディスパッチャーごとに 1 を加え、さらに 1 を加える。

DATABUFFERPOOLSIZE
開始した TCP チャネルごとに 2 を加え、さらに 1 を加える。

MAXFILEPROC
チャネル・イニシエーター内のそれぞれのディスパッチャーが処理できるチャネルの数を制御する。

このパラメーターは SYSI.PARMLIB の BPXPRMxx メンバーで指定します。 必ず要件を満たす値を指定してください。

チャネル・イニシエーターは、デフォルトでは、TCPNAME キュー・マネージャー属性に指定されているスタックに関連する IP アドレスへのバインドのみを実行できます。 チャネル・イニシエーターがシステム上の追加 TCP/IP スタックを使用して通信するようにするには、TCPSTACK キュー・マネージャー属性を MULTIPLE に変更する必要があります。

MQ Adv. CD TCP/IP を使用するメッセージ・チャネルは、ネットワーク・スループットを大幅に向上させることができる高速 TCP/IP トンネルを提供する IBM Aspera faspio Gateway を指すことができます。 資格のあるプラットフォームで実行されているキュー・マネージャーは、Aspera gateway を介して接続できます。 ゲートウェイ自体は、 Red Hat または Ubuntu Linux、あるいは Windows にデプロイされます。 Linux または Windows での Aspera gateway 接続の定義を参照してください。

関連概念

1011 ページの『送信側』
TCP/IP 接続の送信側には、構成するいくつかの設定があります。

1012 ページの『TCP での受信』
TCP/IP 接続の受信側には、構成するいくつかの設定があります。

1012 ページの『z/OS での TCP リスナー・バックログ・オプションの使用』
TCP/IP で受信するときに、未解決の接続要求の最大数を設定します。 これらの未解決の要求は、リスナーが要求を受け入れるのを TCP/IP ポート上で待っている要求のバックログ と考えることができます。

z/OS 送信側

TCP/IP 接続の送信側には、構成するいくつかの設定があります。

チャネル定義内の接続名 (CONNAME) フィールドには、ホスト名 (例: MVSHUR1) か、またはターゲットの TCP ネットワーク・アドレスを設定する必要があります。 TCP ネットワーク・アドレスは、IPv4 小数点付き 10 進数形式 (例えば、127.0.0.1) または IPv6 16 進数形式 (例えば、2001:DB8:0:0:0:0:0:0) になります。 接続名がホスト名である場合は、ホスト名を TCP ホスト・アドレスに変換するために TCP ネーム・サーバーが必要です。 (この要件は、IBM MQ の機能でなく、TCP の機能です。)

接続の開始側 (送信側、要求側、およびサーバーのチャネル・タイプ) では、例えば、次のようにオプションの接続用ポート番号を指定できます。

接続名
192.0.2.0(1555)

この場合、接続の開始側では、ポート 1555 で listen 中の受信プログラムに接続しようとします。

注 : オプションのポート番号が指定されない場合、デフォルトのポート番号である 1414 が使用されます。

チャネル・イニシエーターはアクティブかつ使用可能な任意の TCP/IP スタックを使用できます。 チャネル・イニシエーターは、デフォルトでは、アウトバウンド・チャネルを TCPNAME キュー・マネージャー属性に指定されている TCP/IP スタックのデフォルト IP アドレスにバインドします。 異なるスタックを使用して接続するには、チャネルの LOCLADDR 属性に、スタックのホスト名または IP アドレスを指定する必要があります。

z/OS *TCP* での受信

TCP/IP 接続の受信側には、構成するいくつかの設定があります。

受信側のチャネル・プログラムは、送信側チャネルからの開始要求に応じて開始されます。 そのためには、リスナー・プログラムを開始して着信ネットワーク要求を検出し、関連のチャネルを開始する必要があります。 このリスナー・プログラムは START LISTENER コマンドか操作および制御パネルを使用して開始してください。

デフォルトでは、次のようになります。

- TCP リスナー・プログラムはポート 1414 を使用し、TCP スタックで使用できるすべてのアドレスで listen します。
- TCP/IP リスナーは、TCPNAME キュー・マネージャー属性に指定されている TCP/IP スタックに関連する IP アドレスへのバインドのみを実行できます。

他のアドレス、または使用可能なすべての TCP スタックのリスナーを開始するには、TCPSTACK キュー・マネージャー属性を「MULTIPLE」に設定します。

START LISTENER コマンドで IPADDR を指定すれば、特定のアドレスまたはホスト名でのみ listen するように TCP リスナー・プログラムを開始することができます。 詳しくは、Listeners を参照してください。

z/OS *z/OS* での *TCP* リスナー・バックログ・オプションの使用

TCP/IP で受信するときに、未解決の接続要求の最大数を設定します。 これらの未解決の要求は、リスナーが要求を受け入れるのを TCP/IP ポート上で待っている要求のバックログ と考えることができます。

z/OS でのデフォルトのリスナー・バックログ値は 10000 です。 バックログがこの値に到達すると、TCP/IP 接続は拒否され、チャネルを開始できません。

MCA チャネルの場合は、チャネルが RETRY 状態になり、後で接続が再試行されます。

クライアント接続の場合は、クライアントは、MQCONN から理由コード MQRC_Q_MGR_NOT_AVAILABLE を受け取り、後で接続を再試行することができます。

関連概念

262 ページの『IBM MQ for Multiplatforms での TCP リスナー・バックログ・オプションの使用』
TCP では、サーバーとクライアントとの間で 3 ウェイ・ハンドシェークが実行されない限り、接続は不完全として扱われます。 これらの接続は、未解決接続要求と呼ばれます。 最大値をこれらの未解決接続要求に設定し、リスナーが要求を受け入れるのを TCP ポート上で待っている要求のバックログと考えることができます。

z/OS *APPC/MVS* を使用した *z/OS* 用の *LU6.2* 接続の定義

LU6.2 接続を定義するために、構成するいくつかの設定があります。

APPC/MVS のセットアップ

チャネル・イニシエーターの各インスタンスは、次の例に示すように、APPC/MVS に対して定義するために使用する LU の名前を SYS1.PARMLIB の APPCPMxx メンバー内に入れておく必要があります。

```
LUADD ACBNAME( luname ) NOSCHED TPDATA(CSQ.APPCTP)
```

luname は、使用する論理装置名です。 NOSCHED は必須です。TPDATA は使用されません。 ASCHPMxx メンバーまたは APPC/MVS TP プロファイル・データ・セットへの追加は必要ありません。

DQM によって使用される接続を定義するために、サイド情報データ・セットを拡張します。 APPC ユーティリティー・プログラム ATBSDFMU によるサイド情報データ・セットの拡張方法については、システムか

ら提供されるサンプル CSQ4SIDE を参照してください。 使用する TPNAME 値の詳細については、以下の表の情報を参照してください。

表 *83.* ローカル *z/OS* システムにおけるリモート・キュー・マネージャー・プラットフォーム用の設定値	
リモート・プラットフォーム	**TPNAME**
z/OS または MVS	リモート・キュー・マネージャーの対応するサイド情報内の TPNAME と同じ。
IBM i	IBM i システムにおいて経路指定項目の比較値と同じ。
AIX and Linux システム	リモート・キュー・マネージャーの対応するサイド情報内の TPNAME と同じ。
Windows	Windows Run Listener コマンドで指定されたとおり、または Windows において TpSetup を使用して定義された呼び出し可能なトランザクション・プログラム。

同一のマシン上に複数のキュー・マネージャーがある場合は、チャネル定義内の TPname をそれぞれ固有にする必要があります。

キュー・マネージャーが、APPC を使用して同一または別の z/OS システム上のキュー・マネージャーと通信している環境では、通信中の LU の VTAM 定義が SECACPT(ALREADYV) を指定しているか、LU 間の接続用の RACF の APPCLU プロファイルが CONVSEC(ALREADYV) を指定していることを確認してください。

基本およびリスナー LU の両方に対して z/OS コマンド VARY ACTIVE を発行してから、インバウンドまたはアウトバウンド通信のいずれかの開始を試みるようにします。

重要 : APPC セットアップに加えて、以下のコマンドを発行する必要があります。

```
ALTER QMGR LUNAME(luname)
```

次に、チャネル・イニシエーターを再始動します。

詳細については、LUNAME を参照してください。

関連概念

1013 ページの『LU 6.2 への接続』
LU 6.2 に接続するために、構成するいくつかの設定があります。

1013 ページの『LU 6.2 での受信』
LU 6.2 で受信するために、構成するいくつかの設定があります。

z/OS *LU 6.2* への接続

LU 6.2 に接続するために、構成するいくつかの設定があります。

チャネル定義内の接続名 (CONNAME) フィールドは、APPC/MVS のサイド情報データ・セット内に指定されているシンボリック宛先名に設定する必要があります。

使用する LU 名は (上述のように APPC/MVS に定義されます) チャネル・イニシエーター・パラメーター内にも指定する必要があります。 これは、リスナーによって受信される LU と同じものに設定します。

チャネル・イニシエーターは、"SECURITY(SAME)"APPC/MVS オプションを使用するので、アウトバウンド伝送に使用されるチャネル・イニシエーター・アドレス・スペースのユーザー ID であり、受信側に提示されます。

z/OS *LU 6.2* での受信

LU 6.2 で受信するために、構成するいくつかの設定があります。

送信側チャネルからの始動要求に応じて、MCA の受信が開始されます。 そのためには、リスナー・プログラムを開始して着信ネットワーク要求を検出し、関連のチャネルを開始する必要があります。 リスナー・プログラムは APPC/MVS サーバーの 1 つです。 これは START LISTENER コマンドまたは操作および制御パネルを使用して開始します。 サイド情報データ・セット内に定義されているシンボリック宛先名を使用して、LU 名を指定する必要があります。 このようにして識別されるローカル LU は、チャネル・イニシエーター・パラメーターに設定された、アウトバウンド伝送に使用されるものと同じでなければなりません。

z/OS キュー共有グループを使用する DQM のための IBM MQ for z/OS の準備

このセクションの説明を使用して、IBM MQ for z/OS で、キュー共有グループを使用した分散キューイングを構成します。

キュー共用グループを使用した構成の例については、構成例 - キュー共用グループを使用する IBM MQ for z/OS を参照してください。 キュー共有グループを使用するメッセージ・チャネルの計画例については、キュー共有グループを使用する z/OS におけるメッセージ・チャネルの計画例を参照してください。

キュー共有グループを使用した分散キューイングを使用可能にするために、以下のコンポーネントを作成および構成する必要があります。

- LU 6.2 および TCP/IP リスナー
- 伝送キューおよびトリガー操作
- メッセージ・チャネル・エージェント
- 同期キュー

通信をセットアップするために必要なコンポーネントを作成した後、1018 ページの『キュー共有グループを使用する IBM MQ for z/OS の通信のセットアップ』を参照してください。

キュー共用グループを使用しているときに、チャネルをモニターおよび制御する方法については、992 ページの『z/OS 上のチャネルのモニターおよび制御』を参照してください。

キュー共有グループの概念および利点については、以下のセクションを参照してください。

サービス・クラス

共有キューとは、さまざまなサービス・クラスを備えた一種のローカル・キューのことです。 共有キュー上のメッセージはカップリング・ファシリティー (CF) に保管されます。そのため、キュー共有グループ内のどのキュー・マネージャーからでもアクセスすることができます。 共有キュー上のメッセージの長さは、100 MB を超えないものでなければなりません。

汎用インターフェース

キュー共有グループには、ネットワークがグループを単一のエンティティーと見なすのに必要な汎用インターフェースがあります。 このように見なされるには、グループ内のどのキュー・マネージャーに接続するのにも使用できる単一の汎用アドレスを設けます。

キュー共有グループのすべてキュー・マネージャーは、その汎用アドレスに論理的に関連したアドレスで、インバウンド・セッション要求を listen します。 詳しくは、1015 ページの『キュー共有グループの LU 6.2 および TCP/IP リスナー』を参照してください。

ロード・バランス・チャネル開始機能

共有伝送キューは、キュー共有グループ内の任意のチャネル・イニシエーターで実行されているアウトバウンド・チャネルを利用することができます。 ロード・バランス・チャネル開始機能によってチャネル・コマンドの宛先が決められます。 所定の通信サブシステムにアクセスできる適切なチャネル・イニシエーターが選ばれます。 例えば、TRPTYPE(LU6.2) と定義されたチャネルを、TCP/IP サブシステムにしかアクセスできないチャネル・イニシエーターで開始することはできません。

どのチャネル・イニシエーターが選択されるかは、チャネル・イニシエーターのチャネル・ロードと余力によって決まります。 チャネル・ロードとは、チャネル・イニシエーターのパラメーターに定義されている最大許容アクティブ・チャネル数に対する実際のアクティブ・チャネル数のパーセントのことです。 余力とは、実際のアクティブ・チャネル数と最大許容数の差のことです。

インバウンド共用チャネルは、1015 ページの『キュー共有グループの LU 6.2 および TCP/IP リスナー』で説明されているように、総称アドレスを使用して、キュー共用グループ間でロード・バランシングすることができます。

共用チャネルの回復

以下の表に、共用チャネルでの障害の種類とそれぞれの種類別の対処法を示します。

障害の種類	障害の内容とその対処
チャネル・イニシエーターの通信サブシステムの障害	通信サブシステムに従属しているチャネルは、チャネル再試行に入ってから、ロード・バランス開始コマンドによって、該当するキュー共用グループ・チャネル・イニシエーターで再始動されます。
チャネル・イニシエーター障害	チャネル・イニシエーターに障害が起きましたが、それに関連したキュー・マネージャーはアクティブのままです。 キュー・マネージャーは障害をモニターし、回復処理を開始します。
キュー・マネージャー障害	キュー・マネージャーに障害が起き、それに関連したチャネル・イニシエーターに障害が起きました。 キュー共用グループ中の他のキュー・マネージャーが、イベントをモニターし、ピア回復を開始します。
共用状況障害	チャネルの状態情報は Db2 に保管されるので、チャネルの状態の変更のときに Db2 への接続が失われると障害が起きます。 実行中のチャネルは、このリソースにアクセスできなくても実行を継続できます。 Db2 へのアクセスで障害が起きると、チャネルは再試行に入ります。

障害を起こしたシステムに代わって共用チャネル回復処理を行う場合、共用チャネル状況を取得するため、回復しようとするシステム上で Db2 への接続が使用可能になっている必要があります。

クライアント・チャネル

個々のキュー・マネージャーにではなく汎用インターフェースに接続されているキュー共有グループ内のメッセージの高い利用性をクライアント接続チャネルで活用することができます。 詳しくは、クライアント接続チャネルを参照してください。

関連概念

共用キューとキュー共用グループ

914 ページの『IBM MQ for z/OS の設定』
このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

1017 ページの『クラスターとキュー共用グループ』
1 つの定義で、クラスターが共有キューを使用可能にすることができます。 そうするためには、共有キューを定義するときにクラスター名を指定します。

1017 ページの『チャネルと直列化』
共有キュー対等リカバリー中に、共有キュー上のメッセージを処理するメッセージ・チャネル・エージェントは、それらのキューへの自身のアクセスを直列化します。

グループ内キューイング

関連タスク

190 ページの『分散キューイングの構成』
このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

988 ページの『z/OS での他のキュー・マネージャーとの通信のセットアップ』
このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

z/OS キュー共有グループの *LU 6.2* および *TCP/IP* リスナー

グループ LU 6.2 および TCP/IP のリスナーは、汎用アドレスに論理接続されているアドレスで listen します。

LU 6.2 リスナーの場合、指定の LUGROUP は、キュー共有グループに関連した VTAM 汎用リソースにマップされます。 このテクノロジーをセットアップする例については、1012 ページの『APPC/MVS を使用した z/OS 用の LU6.2 接続の定義』を参照してください。

TCP/IP リスナーの場合、指定のポートは、以下のいずれかの方法で汎用アドレスに接続することができます。

- IBM Network Dispatcher などのフロントエンド・ルーターの場合、インバウンド接続要求は、ルーターからキュー共有グループのメンバーに転送されます。
- TCP/IP のシスプレックス・ディストリビューターの場合、分散 DVIPA としてセットアップされた特定アドレス上で実行および listen 中の各リスナーに一定の比率で着信要求が割り振られます。 このテクノロジーのセットアップの例は、シスプレックス・ディストリビューターの使用に記載されています。

z/OS キュー共有グループの伝送キューおよびトリガー操作

キュー共有グループから宛先へのメッセージの移動までの間、そのメッセージを保管するのに共有伝送キューが使われます。

これが共有キューであり、これには、キュー共有グループのどのキュー・マネージャーからでもアクセスすることができます。

トリガー

トリガー共有キューは、トリガー条件が満たされた場合に複数のトリガー・メッセージを生成することができます。 トリガー共有キューに関連付けられたキュー共有グループ内のキュー・マネージャーで定義されている各ローカル開始キューごとにメッセージが 1 つずつ生成されます。

分散キューイングの場合、共有伝送キュー・トリガー条件が満たされると、各チャネル・イニシエーターがトリガー・メッセージを受け取ります。 ただし、トリガーの開始を実際に処理するチャネル・イニシエーターは 1 つだけで、残りのプログラムは処理に携わりません。 トリガーされたチャネルはロード・バランス開始 (1014 ページの『キュー共有グループを使用する DQM のための IBM MQ for z/OS の準備』を参照) で開始され、トリガーされてチャネル QSG.TO.QM2 を開始します。 共有伝送キューを作成するには、IBM MQ コマンド (MQSC) を次の例のように使用します。

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') +
USAGE(XMITQ) QSGDISP(SHARED) +
CFSTRUCT(APPLICATION1) INITQ(SYSTEM.CHANNEL.INITQ) +
TRIGGER TRIGDATA(QSG.TO.QM2)
```

注 : 共用キューがトリガー用にセットアップされていて、共用キューをホストするカップリング・ファシリティーへの接続が失われると、トリガー・イベントが生成され、メッセージが開始キューに書き込まれる可能性があります。 これは、トリガー操作のために元の共用キュー・セットアップにメッセージが書き込まれなかった場合でも発生する可能性があります。 これは、 「リスト通知ベクトル」で説明されているように、IXLVECTR マクロによるビットの過剰表示が原因です。

z/OS キュー共有グループのメッセージ・チャネル・エージェント

チャネル・イニシエーターでチャネルを開始できるのは、該当する名前の付いたチャネルのチャネル定義にアクセスできる場合だけです。

メッセージ・チャネル・エージェントは、メッセージの送受信を制御する IBM MQ プログラムです。 メッセージ・チャネル・エージェントは、1 つのキュー・マネージャーから別のキュー・マネージャーにメッセージを移動します。 チャネルの両端に 1 つずつ、メッセージ・チャネル・エージェントが存在します。

チャネル定義は、キュー・マネージャー専用として定義することができますが、共有リポジトリーに保管してどこででも使える (グループ定義) ようにすることもできます。 つまり、グループ定義のチャネルは、キュー共有グループ内のすべてのチャネル・イニシエーターで使えるということです。

注 : グループ定義を専用コピーしたものは、変更または削除することができます。

グループ・チャネル定義を作成するには、IBM MQ コマンド (MQSC) を次の例のように使用します。

```
DEFINE CHL(QSG.TO.QM2) CHLTYPE(SDR) +
TRPTYPE(TCP) CONNAME(QM2.MACH.IBM.COM) +
XMITQ(QM2) QSGDISP(GROUP)
```

```
DEFINE CHL(QM2.TO.QSG) CHLTYPE(RCVR) TRPTYPE(TCP) +
QSGDISP(GROUP)
```

キュー共有グループでの分散キューイングに使用されるメッセージ・チャネル・エージェントには、次のように 2 とおりの見方があります。

インバウンド

インバウンド・チャネルは、グループ・リスナーを通してキュー・マネージャーに接続されていれば共有チャネルになります。 このチャネルは、汎用インターフェースを通してキュー共有グループに接続されてグループ内のキュー・マネージャーにつながるか、あるいは特定のキュー・マネージャーのグループ・ポートまたはグループ・リスナーが使用する LU 名の対象になります。

アウトバウンド

アウトバウンド・チャネルは、共有伝送キューからメッセージを移動する場合は共有チャネルになります。このコマンド例では、伝送キュー QM2 が QSGDISP (SHARED) で定義されているため、送信側チャネル QSG.TO.QM2 は共有チャネルです。

z/OS キュー共有グループの同期キュー

共有チャネルには、SYSTEM.QSG.CHANNEL.SYNCQ という名前の独自の共有同期キューがあります。

この同期キューは、キュー共有グループ内のすべてメンバーにアクセスすることができます。 (専用チャネルは専用同期キューを引き続き使用します。 991 ページの『z/OS での IBM MQ オブジェクトの定義』).を参照してください。 つまり、通信サブシステム、チャネル・イニシエーター、またはキュー・マネージャーに障害が起きた場合、キュー共有グループ内の別のキュー・マネージャーおよびチャネル・イニシエーターのインスタンスからチャネルを再始動できるということです。 詳しくは、1014 ページの『キュー共有グループを使用する DQM のための IBM MQ for z/OS の準備』を参照してください。

キュー共有グループを使用する DQM の場合、SYSTEM.QSG.CHANNEL.SYNCQ という名前の共有キューが使用可能でなければなりません。 グループ・リスナーを正常に始動できるようにするには、このキューが使用可能になっていなければなりません。

このキューが使用可能でないためにグループ・リスナーに障害が起きても、チャネル・イニシエーターのリサイクルなしに、このキューを定義してリスナーを再始動することができます。 非共有チャネルは影響を受けません。

このキューは、必ず INDXTYPE(MSGID) を使用して定義してください。 この定義により、キューのメッセージに対するアクセス速度が向上します。

z/OS クラスターとキュー共用グループ

1 つの定義で、クラスターが共有キューを使用可能にすることができます。 そうするためには、共有キューを定義するときにクラスター名を指定します。

キュー共有グループ内のすべてのキュー・マネージャーを共有キューはホストとして利用しているようにネットワーク内のユーザーには見えます。 (共有キューがキュー共有グループをホストとして利用していることは公示されません。) クライアントは、キュー共有グループのどのメンバーとのセッションを開始しても、同じ共有キューにメッセージを書き込むことができます。

詳細については、287 ページの『キュー・マネージャー・クラスターの構成』を参照してください。

z/OS チャネルと直列化

共有キュー対等リカバリー中に、共有キュー上のメッセージを処理するメッセージ・チャネル・エージェントは、それらのキューへの自身のアクセスを直列化します。

メッセージ・チャネル・エージェントが 1 つ以上の共有キュー上のコミットされていないメッセージを扱っているときに キュー共有グループ内のキュー・マネージャーに障害が起きた場合は、チャネルおよび関連したチャネル・イニシエーターは終了して、キュー・マネージャーの共有キュー対等リカバリーが行われます。

共有キュー対等リカバリーは非同期アクティビティーであるため、対等チャネル・リカバリーでは、共有キュー対等リカバリーが完了する前に、同時にキュー共有グループ内の別の部分のチャネルに対して再始動の試行が行われる可能性があります。 このイベントが起きた場合は、まだリカバリー中のメッセージより先に、コミット済みメッセージが処理される可能性があります。 メッセージがこのように違う順序で処理されないようにするために、共有キュー上のメッセージを処理するメッセージ・チャネル・エージェントは、それらのキューへの自身のアクセスを直列化します。

共有キュー対等リカバリーがまだ進行中のチャネルを開始しようとすると、失敗になります。 リカバリーが進行中であることを示すエラー・メッセージが出されて、チャネルは再試行状態になります。 キュー・マネージャー対等リカバリーが完了すると、チャネルは次の再試行時に再始動されます。

チャネルの解決 (RESOLVE)、PING、削除 (DELETE) を行おうとした場合も、同じ理由で失敗になります。

z/OS キュー共有グループを使用する *IBM MQ for z/OS* の通信のセットアップ

分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。 この試行を正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。

使用できる次の 2 つの形式の通信プロトコルから、いずれかを選択します。

- tcp
- LU 6.2 から APPC/MVS

構成例 - キュー共有グループを使用する IBM MQ for z/OS を参照すると役に立つ場合があります。

z/OS キュー共有グループ用の *TCP* 接続の定義

キュー共有グループ用に TCP 接続を定義するには、送信側および受信側の特定の属性を構成する必要があります。

TCP のセットアップについては、1010 ページの『z/OS での TCP 接続の定義』を参照してください。

送信側

キュー共有グループに接続するためのチャネル定義の接続名(CONNAME)フィールドは、キュー共有グループの汎用インタフェースに設定する必要があります(キュー共有グループ参照)。 詳細については、シスプレックス・ディストリビューターの使用を参照してください。

キュー共有グループを使用した TCP での受信

受信側の共有チャネル・プログラムは、送信側チャネルからの始動要求に応じて開始されます。 そのためには、リスナーを開始して着信ネットワーク要求を検出し、関連のチャネルを開始する必要があります。 このリスナー・プログラムの開始は START LISTENER コマンド、グループの到着属性指定、または操作および制御パネルを使用して行います。

キュー共有グループ内のすべてのグループ・リスナーが同一ポートで listen していなければなりません。 単一の MVS イメージ上で複数のチャネル・イニシエーターを実行する場合は、START LISTENER コマンドで IPADDR を指定すれば、仮想 IP アドレスを定義することができ、特定のアドレスまたはホスト名でのみ listen するように TCP リスナー・プログラムを開始することができます。 (詳細については、START LISTENER を参照してください。)

z/OS *z/OS での LU 6.2* 接続の定義

キュー共有グループ用に LU 6.2 接続を定義するには、送信側および受信側の特定の属性を構成する必要があります。

APPC/MVS のセットアップについては、z/OS における通信のセットアップを参照してください。

APPC/MVS (LU 6.2) への接続

キュー共有グループに接続するためのチャネル定義内の接続名 (CONNAME) フィールドは、APPC/MVS のサイド情報データ・セットで指定されているシンボリック宛先名に設定する必要があります。 このシンボリック宛先で定義されているパートナー LU は、汎用リソース名でなければなりません。 詳細については、汎用リソースの使用によるネットワークへの自身の登録を参照してください。

汎用インターフェースを使用した LU 6.2 での受信

受信側の共有 MCA は、送信側チャネルからの始動要求に応じて開始されます。 そのためには、グループ・リスナー・プログラムを開始して着信ネットワーク要求を検出し、関連のチャネルを開始する必要があります。 リスナー・プログラムは APPC/MVS サーバーの 1 つです。 リスナー・プログラムの開始は START LISTENER コマンド、到着属性指定グループ、または操作および制御パネルを使用して行います。 サイド情報データ・セット内に定義されているシンボリック宛先名を使用して、LU 名を指定する必要があります。 詳細については、汎用リソースの使用によるネットワークへの自身の登録を参照してください。

z/OS IBM MQ と IMS の使用

IBM MQ-IMS アダプター、および IBM MQ-IMS ブリッジは、IBM MQ と IMS との対話を可能にする二つのコンポーネントです。

IBM MQ と IMS が一緒に動作するよう構成するには、以下のタスクを完了する必要があります。

- 1019 ページの『IMS アダプターのセットアップ』
- 1026 ページの『IMS ブリッジのセットアップ』

関連概念

IBM MQ と IMS

1027 ページの『IBM MQ と CICS の使用』
CICS で IBM MQ を使用するには、IBM MQ CICS アダプターと、オプションで IBM MQ CICS bridge コンポーネントを構成する必要があります。

1029 ページの『IMS における OTMA 出口の使用』
IBM MQ for z/OS で IMS Open Transaction Manager Access 出口を使用する場合は、このトピックを使用してください。

IBM MQ for z/OS 上の IMS および IMS ブリッジ・アプリケーション

関連タスク

908 ページの『z/OS でのキュー・マネージャーの構成』
IBM MQ for z/OS でキュー・マネージャーを構成するには、この指示に従ってください。

関連資料

1027 ページの『言語環境プログラムまたは z/OS 呼び出し可能サービスのアップグレードおよび保守の適用』
必要なアクションは、CALLLIBS または LINK のどちらを使用するか、および使用する SMP/E のバージョンによって異なります。

z/OS IMS アダプターのセットアップ

IMS 内で IBM MQ を使用するには、IBM MQ-IMS アダプター(一般的には IMS アダプターと呼ばれます)が必要です。

このトピックでは、IMS アダプターを IMS サブシステムで使用できるようにする方法について説明します。 IMS サブシステムの調整に精通していない場合は、 IMS の資料 を参照してください。

IMS アダプターを IMS アプリケーションで使用できるようにするには、次のステップを実行してください。

1. IMS 外部サブシステム接続機能 (ESAF) を使用して、IMS に対して外部サブシステムとして IBM MQ を定義します。

   1021 ページの『IBM MQ の定義 IMS』 を参照してください。

2. JOBLIB 連結または STEPLIB 連結内の IBM MQ ロード・ライブラリー thlqual.SCSQAUTH を、使用している IMS 制御領域用および IBM MQ に接続した従属領域用の JCL に組み込みます (LPA またはリンク・リストに含まれていない場合)。 また、使用している JOBLIB または STEPLIB が許可済みでない場合は、それを DFSESL 連結内の、IMS モジュールが入っているライブラリー (通常は IMS RESLIB) のあとに組み込みます。

   また、thlqual.SCSQANLx (x は言語を示す文字) も組み込んでください。

   DFSESL が存在する場合、SCSQAUTH および SCSQANLx を連結に含めるか、LNKLIST に追加する必要があります。 JCL の STEPLIB または JOBLIB 連結に追加するだけでは十分ではありません。
3. IBM MQ アセンブラー・プログラム CSQQDEFV を、thlqual.SCSQASMS からユーザー・ライブラリーへコピーします。
4. 提供されるプログラム CSQQDEFV は、IMS 言語インターフェース・トークン (LIT) である MQM1 と共に、デフォルト値であるサブシステム名 CSQ1 を含んでいます。 テストおよびインストール検査では、この名前をそのまま使用できます。

   実動サブシステムでは、NAME=CSQ1 を独自のサブシステム名に変更するか、CSQ1 をそのまま使用します。 必要に応じて、さらにサブシステム定義を追加することもできます。 LITs の詳細については、1024 ページの『IMS アダプターに対する IBM MQ キュー・マネージャーの定義』を参照してください。
5. プログラムをアセンブルし、リンク・エディットして、CSQQDEFV ロード・モジュールを生成します。アセンブリーではライブラリー thlqual.SCSQMACS を SYSLIB 連結に組み込みます (リンク・エディット・パラメーター RENT を使用してください)。 これは thlqual.SCSQPROC(CSQ4DEFV) のサンプル JCL に示されています。
6. JOBLIB 連結または STEPLIB 連結内で作成したモジュール CSQQDEFV が入っているユーザー・ライブラリーを、IBM MQ に接続した従属領域用の JCL に組み込みます。 SCSQAUTH にデフォルトのロード・モジュールがあるので、このライブラリーを SCSQAUTH の前に入れてください。 これを行わないと、IMS からユーザー 3041 異常終了コードを受け取ります。
7. IMS アダプターは、予期しない IBM MQ エラーを検出すると、DD 名 CSQSNAP に対して、z/OSSNAP ダンプを発行し、理由コード MQRC_UNEXPECTED_ERROR をアプリケーションに発行します。 CSQSNAP DD ステートメントが IMS 従属領域 JCL に含まれていなかった場合、ダンプはとられません。 この場合、JCL に CSQSNAP DD ステートメントを含めて、アプリケーションを再実行することができます。 しかし、いくつかの問題が断続的に発生することがあるため、CSQSNAP DD ステートメントを含めて、問題の発生時に障害の理由を見極めることをお勧めします。
8. 動的 IBM MQ 呼び出し(IBM MQ スタブの動的呼び出しでの説明)を使用する場合は、1021 ページの図 117 に示されているように、動的スタブをビルドします。
9. IMS トリガー・モニターを使用する場合は、IMS トリガー・モニター・アプリケーション CSQQTRMN を定義し、PSBGEN と ACBGEN を実行します。 1025 ページの『IMS トリガー・モニターのセットアップ』を参照してください。
10. OPERCMDS クラスのリソースを保護するために RACF を使用している場合は、IBM MQ キュー・マネージャーのアドレス・スペースに関連したユーザー ID が、接続されている可能性のある IMS システムに対して、MODIFY コマンドを発行する権限を持っていることを確認してください。

```
//DYNSTUB EXEC PGM=IEWL,PARM='RENT,REUS,MAP,XREF'
//SYSPRINT DD  SYSOUT=*
//ACSQMOD  DD  DISP=SHR,DSN=thlqual.SCSQLOAD
//IMSLIB   DD  DISP=SHR,DSN=ims.reslib
//SYSLMOD  DD  DISP=SHR,DSN=private.load[1]
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,1)
//SYSLIN   DD  *
  INCLUDE ACSQMOD(CSQQSTUB)
  INCLUDE IMSLIB(DFSLI000)
  ALIAS MQCONN,MQCONNX,MQDISC             MQI entry points
  ALIAS MQGET,MQPUT,MQPUT1                MQI entry points
  ALIAS MQOPEN,MQCLOSE                    MQI entry points
  ALIAS MQBACK,MQCMIT                     MQI entry points
  ALIAS CSQBBAK,CSQBCMT                   MQI entry points
  ALIAS MQINQ,MQSET                       MQI entry points
  ALIAS DFSPLI,PLITDLI                    IMS entry points
  ALIAS DFSCOBOL,CBLTDLI                  IMS entry points
  ALIAS DFSFOR,FORTDLI                    IMS entry points
  ALIAS DFSASM,ASMTDLI                    IMS entry points
  ALIAS DFSPASCL,PASTDLI                  IMS entry points
  ALIAS DFHEI01,DFHEI1                    IMS entry points
  ALIAS DFSAIBLI,AIBTDLI                  IMS entry points
  ALIAS DFSESS,DSNWLI,DSNHLI              IMS entry points
  ALIAS MQCRTMH,MQDLTMH,MQDLTMP           IMS entry points
  ALIAS MQINQMP,MQSETMP,MQMHBUF,MQBUFMH   IMS entry points
  MODE AMODE(31),RMODE(24)                Note RMODE setting
  NAME CSQQDYNS(R)
/*

[1]Specify the name of a library accessible to IMS applications that
want to make dynamic calls to IBM MQ.
```

図 *117.* 動的呼び出しスタブをリンク・エディットする *JCL* のサンプル

関連概念

IBM MQ と IMS

1026 ページの『IMS ブリッジのセットアップ』

IBM MQ-IMS ブリッジは、IBM MQ 対応ではない既存のプログラムおよびトランザクションとの間で、IBM MQ による入出力を可能にするオプションのコンポーネントです。

IBM MQ for z/OS 上の IMS および IMS ブリッジ・アプリケーション

z/OS *IBM MQ の定義 IMS*

IBM MQ は、IMS 制御領域に対して定義し、またその IBM MQ キュー・マネージャーにアクセスする各従属領域に対して定義する必要があります。 このためには、IMS .PROCLIB ライブラリーにサブシステム・メンバー (SSM) を 1 つ作成し、その SSM を適用可能な IMS 領域に知らせなければなりません。

IMS にサブシステム・メンバー項目を配置する。PROCLIB

IMS .PROCLIB 内の 1 つの SSM 項目は、IMS 領域から別のキュー・マネージャーへの 1 つの接続を定義します。

SSM の名前を付けるために、IMS IMSCTRL マクロの IMSID フィールドの値 (1 から 4 文字の英数字) とユーザーが定義した任意の名前 (1 から 4 文字の英数字) とを連結します。

1 つの SSM をすべての IMS 領域で共有するか、特定のメンバーを各領域用に定義できます。 このメンバーには、外部サブシステムへの接続の数だけの項目が入っています。 1 つの項目は、80 文字からなる 1 つのレコードです。

定位置パラメーター

この項目のフィールドは、次のとおりです。

```
SSN,LIT,ESMT,RTT,REO,CRC
```

ここで、

SSN
IBM MQ キュー・マネージャー名を指定します。 このフィールドは必須であり、1 ～ 4 文字が入っていなければなりません。

LIT
IMS に供給する言語インターフェース・トークン (LIT) を指定します。 このフィールドは必須であり、その値は、CSQQDEFV モジュールのフィールドと一致する必要があります。

ESMT
外部サブシステム・モジュール表 (ESMT) を指定します。 この表は、どの接続モジュールを IMS がロードしなければならないかを指定します。 CSQQESMT がこのフィールドの必須の値です。

RTT
このオプションは、IBM MQ ではサポートされません。

REO
IMS アプリケーションが動作しない外部サブシステムを参照する場合、または資源がスレッド作成時に使用可能でない場合に使用される領域エラー・オプション (REO) を指定します。 このフィールドの指定は任意であり、次の 1 文字が含まれます。

R
戻りコードをアプリケーションに渡し、IBM MQ サービスに対する要求が失敗したことを示します。

Q
アプリケーションを異常終了コード U3051 で終了させ、アクティビティーを最後のコミット点までバックアウトし、トランザクションの PSTOP を行い、入力メッセージを再キューイングします。 このオプションは、IMS アプリケーションが動作しない外部サブシステムを参照しようとするか、資源がスレッド作成時に使用可能でない場合にだけ適用されます。

IBM MQ が要求を処理している間、すなわちアダプターが要求を IBM MQ へ渡した後、IBM MQ の問題が起こった場合、IBM MQ の完了コードおよび理由コードがアプリケーションへ戻されます。

A
アプリケーションを異常終了コード U3047 で終了させ、入力メッセージを破棄します。 このオプションは、IMS アプリケーションが動作しない外部サブシステムを参照するか、資源がスレッド作成時に使用可能でない場合にだけ適用されます。

IBM MQ が要求を処理している間、すなわちアダプターが要求を IBM MQ へ渡した後、IBM MQ の問題が起こった場合、IBM MQ の完了コードおよび理由コードがアプリケーションへ戻されます。

CRC
このオプションを指定できますが、IBM MQ では使用されません。

注 : すべての定位置パラメーターについて詳しくは、IMS への外部サブシステムの指定方法を参照してください。

SSM 項目の例を次に示します。

```
CSQ1,MQM1,CSQQESMT,,R,
```

ここで、

CSQ1	IBM MQ で提供されているデフォルトのサブシステム名。 この名前は、ご使用のシステムに合わせて変更できます。
MQM1	CSQQDEFV 内で提供されるデフォルトの LIT。
CSQQESMT	外部サブシステム・モジュール名。 この値は必ず使用してください。
R	REO オプション。

キーワード・パラメーター

IBM MQ パラメーターはキーワード形式で指定できます。 SST パラメーターには、DB2 または MQ のいずれかの値を指定できます。 IMS 14 で、MQ の値のサポートが追加されました。 MQ を使用するとより明確になり、IMS サブシステム・コマンドに SST 値が含まれるようになりましたが、それ以外の大きな影響はありません。 必要に応じて、DB2 の値を引き続き使用できます。 その他のパラメーターについては定位置パラメーターに説明があります。 また、以下の例も参照してください。

```
SST=MQ,SSN=SYS3,LIT=MQM3,ESMT=CSQQESMT
```

ここで、

SYS3	サブシステム名。
MQM3	CSQQDEFV に指定された LIT。
CSQQESMT	外部サブシステム・モジュール名。

SSM EXEC パラメーターの指定

IMS 制御領域の始動プロシージャーに SSM EXEC パラメーターを指定してください。 このパラメーターには、1 ～ 4 文字のサブシステム・メンバー名 (SSM) を指定します。

IMS 制御領域に対して、SSM を指定すると、その制御領域で実行されている従属領域は、IMS で指定された IBM MQ キュー・マネージャーに接続することができます。SSM パラメーターによって指定された PROCLIB メンバー。 IMS.PROCLIB メンバ名は IMSID(IMSID=*xxxx*)に SSM EXEC パラメータで指定された 1～4 文字を連結したものです。 IMS ID は、IMSCTRL 生成マクロの IMSID パラメーターです。

IMS によって、いくつでも必要な数の外部サブシステム接続を定義できます。 さまざまな IBM MQ キュー・マネージャーについて、複数の接続を定義できます。 すべての IBM MQ 接続は、同じ z/OS システム内になければなりません。 従属領域では、従属領域 SSM を指定するか、制御領域用に指定されたものを使用することができます。 従属領域 SSM と制御領域 SSM で、異なる領域エラー・オプション (REO) を指定できます。 1023 ページの表 84 は、使用できる各種の SSM 指定を示しています。

表 *84. SSM* 指定オプション

制御領域用 SSM	従属領域用 SSM	アクション	コメント
いいえ	いいえ	なし	外部サブシステムを接続することはできません。
いいえ	はい	なし	外部サブシステムを接続することはできません。
はい	いいえ	制御領域 SSM を使用します。	この領域内でスケジュールされたアプリケーションは、制御領域 SSM 内で識別される外部サブシステムにアクセスできます。 各接続に対する出口および制御ブロックは、制御領域と従属領域のアドレス・スペースにロードされます。
はい	あり (空)	従属領域については、SSM は使用されません。	この領域内でスケジュールされたアプリケーションは、DL/I データベースにだけアクセスできます。 各接続に対する出口および制御ブロックは、制御領域のアドレス・スペースにロードされます。
はい	あり (空ではない)	従属領域 SSM を制御領域 SSM と照合します。	この領域内でスケジュールされたアプリケーションは、両方の SSM で識別される外部サブシステムにだけアクセスできます。 各接続に対する出口および制御ブロックは、制御領域と従属領域のアドレス・スペースにロードされます。

指定可能な SSM の最大数を制御する特定のパラメーターはありません。

IMS アダプターのプリロード

IMS アダプターは、IMS によってプリロードされると、パフォーマンスが向上する場合があります。 プリロードは、IMS .PROCLIB の DFSMPLxx メンバーによって制御されます。詳しくは、「IMS 管理の手引き: システム」を参照してください。 指定する IBM MQ モジュール名は次のとおりです。

CSQACLST	CSQAMLST	CSQAPRH	CSQAVICM	CSQFSALM	CSQQDEFV
CSQQCONN	CSQQDISC	CSQQTERM	CSQQINIT	CSQQBACK	CSQQCMMT
CSQQESMT	CSQQPREP	CSQQTTHD	CSQQWAIT	CSQQNORM	CSQQSSOF
CSQQSSON	CSQFSTAB	CSQQRESV	CSQQSNOP	CSQQCMND	CSQQCVER
CSQQTMID	CSQQTRGI	CSQQCON2	CSQBPAPI	CSQBCRMH	CSQBAPPL

IBM MQ classes for JMS の使用について詳しくは、IMS での IBM MQ classes for JMS の使用を参照してください。

IMS の現行リリースでは、PDS-E 形式ライブラリーの IBM MQ モジュールをプリロードすることがサポートされるのは、MPP、BMP、IFP、JMP および JBP 領域のみです。 その他のタイプの IMS 領域では、PDS-E ライブラリーからのプリロードがサポートされていません。 その他のタイプの領域でプリロードが必要な場合は、提供されている IBM MQ モジュールを PDS 形式ライブラリーにコピーする必要があります。

z/OS *IMS アダプターに対する IBM MQ キュー・マネージャーの定義*

IBM MQ キュー・マネージャーの名前とそれに対応する言語インターフェース・トークン (LIT) をキュー・マネージャーの定義表に定義する必要があります。

CSQQDEFV ロード・モジュールを作成するには、提供された CSQQDEFX マクロを使用します。 1024 ページの図 118 に、このアセンブラー・マクロの構文を示します。

```
CSQQDEFX TYPE=ENTRY|DEFAULT,NAME=qmgr-name,LIT=token
or
CSQQDEFX TYPE=END
```

図 *118. CSQQDEFX* マクロの構文

パラメーター

TYPE=ENTRY|DEFAULT
TYPE=ENTRY または TYPE=DEFAULT を次のように指定します。

TYPE=ENTRY
IMS アプリケーションで使用可能な IBM MQ キュー・マネージャーを記述する表項目を生成することを指定します。 これが最初の項目である場合、CSQQDEFV CSECT ステートメントを含む表ヘッダーも生成されます。

TYPE=DEFAULT
TYPE=ENTRY の場合と同様です。 指定されるキュー・マネージャーは、MQCONN または MQCONNX でブランクのみからなる名前を指定したときに使用されるデフォルトのキュー・マネージャーになります。 このような項目は、表内に 1 つのみ指定できます。

NAME=*qmgr-name*
MQCONN または **MQCONNX** で指定したキュー・マネージャーの名前を指定します。

LIT= token
IMS がキュー・マネージャーを識別するために使用する言語インターフェース・トークン (LIT) の名前を指定します。

MQCONN または MQCONNX 呼び出しは、*name* 入力パラメーターと *hconn* 出力パラメーターを名前ラベルに関連付けます。 これにより、CSQQDEFV 項目内の LIT が関連付けられます。 *hconn* パラ

メーターを渡す以降の IBM MQ 呼び出しでは、 MQCONN または MQCONNX 呼び出しで識別される CSQQDEFV 項目の LIT を使用して、同じ LIT を持つ IMS SSM PROCLIB メンバーで定義された IBM MQ キュー・マネージャーへの呼び出しを送信します。

要約すると、 MQCONN または MQCONNX 呼び出しの **name** パラメーターは CSQQDEFV の LIT を識別し、SSM メンバーの同じ LIT が IBM MQ キュー・マネージャーを識別します。 (MQCONN 呼び出しについては、MQCONN-キュー・マネージャーの接続を参照してください。 MQCONNX 呼び出しについて詳しくは、MQCONNX - キュー・マネージャーの接続 (拡張)」を参照してください。)

TYPE=END
表が完了したことを指定します。 このパラメーターを省略した場合は、TYPE=ENTRY が指定されたものと見なされます。

CSQQDEFX マクロの使用

1025 ページの図 119 には、キュー・マネージャー定義テーブルの一般的なレイアウトが示されます。

```
CSQQDEFX NAME=subsystem1,LIT=token1
CSQQDEFX NAME=subsystem2,LIT=token2,TYPE=DEFAULT
CSQQDEFX NAME=subsystem3,LIT=token3
...
CSQQDEFX NAME=subsystemN,LIT=tokenN
CSQQDEFX TYPE=END
END
```

図 *119.* キュー・マネージャー定義表のレイアウト

z/OS *IMS* トリガー・モニターのセットアップ

IMS バッチ型プログラムをセットアップして、IBM MQ 開始キューをモニターすることができます。

thlqual.SCSQPROC ライブラリーのモデル CSQQTAPL を使用して、アプリケーションを IMS に定義します (CSQQTRMN のトランザクション定義の例 を参照)。

thlqual.SCSQPROC ライブラリー内のモデル CSQQTPSB (CSQQTRMN 用の PSB 定義の例を参照) を使用して、PSB および ACB を生成します。

```
*  This is the application definition *
*  for the IMS Trigger Monitor BMP    *

APPLCTN PSB=CSQQTRMN,
PGMTYPE=BATCH,
SCHDTYP=PARALLEL
```

図 *120. CSQQTRMN* 用のトランザクション定義の例

```
PCB  TYPE=TP,      ALTPCB for transaction messages
MODIFY=YES,        To "triggered" IMS transaction
PCBNAME=CSQQTRMN
PCB  TYPE=TP,      ALTPCB for diagnostic messages
MODIFY=YES,        To LTERM specified or "MASTER"
PCBNAME=CSQQTRMG,
EXPRESS=YES
PSBGEN LANG=ASSEM,
PSBNAME=CSQQTRMN,  Runs program CSQQTRMN
CMPAT=YES
```

図 *121. CSQQTRMN* 用の *PSB* 定義の例

IMS トリガー・モニターの開始および停止について詳しくは、IMS トリガー・モニターの制御 を参照してください。

z/OS IMS ブリッジのセットアップ

IBM MQ-IMS ブリッジは、IBM MQ 対応ではない既存のプログラムおよびトランザクションとの間で、IBM MQ による入出力を可能にするオプションのコンポーネントです。

トピックでは、IBM MQ-IMS ブリッジをカスタマイズするために必要な作業について説明します。

IBM MQ 用に XCF と OTMA パラメーターを定義します。
このステップは、IBM MQ システム用に XCF グループとメンバー名を定義し、さらに他の OTMA パラメーターを定義します。 IBM MQ と IMS は、同じ XCF グループに属している必要があります。 CSQ6SYSP マクロの OTMACON キーワードを使用して、システム・パラメーター・ロード・モジュール内でこれらのパラメーターを調整します。

詳細については、CSQ6SYSP の使用を参照してください。

XCF と OTMA パラメーターを IMS に定義します。
このステップでは、IMS システム用の XCF グループとメンバー名を定義します。 IMS と IBM MQ は、同じ XCF グループに属している必要があります。

JCL または IMS PROCLIB 内のメンバー DFSPBxxx の IMS パラメーター・リストに、次のパラメーターを追加してください。

OTMA=Y
これにより、IMS の開始時に OTMA が自動的に開始されます。 (これはオプションです。OTMA=N を指定した場合は、IMS コマンド /START OTMA を発行することによって OTMA を開始することもできます。)

GRNAME=
このパラメーターには XCF グループ名を指定します。

これはストレージ・クラス定義の中で指定されたグループ名と同じです (次のステップを参照)。また、CSQ6SYSP マクロの OTMACON キーワードの **Group** パラメーターで指定されたグループ名と同じです。

OTMANM=
このパラメーターには IMS システムの XCF メンバー名を指定します。

これはストレージ・クラス定義の中で指定されたメンバー名と同じです (次のステップを参照)。

IBM MQ に IMS システムの XCF グループとメンバー名を指示します。
これはキューのストレージ・クラスによって指定されます。 IBM MQ-IMS ブリッジを介してメッセージを送る場合、キュー用にストレージ・クラスを定義するときに、これを指定する必要があります。 ストレージ・クラスには、対象の IMS システムの XCF グループとメンバー名を定義する必要があります。 これを行うには、 IBM MQ 操作および制御パネルを使用するか、「プログラマブル・コマンド・フォーマットの概要」で説明されているように IBM MQ コマンドを使用します。

必要なセキュリティーを設定します。
/SECURE OTMA IMS コマンドは、OTMA を介して IMS に接続する **すべての** IBM MQ キュー・マネージャーに適用されるセキュリティーのレベルを決定します。 詳しくは、IBM MQ IMS を使用する場合のセキュリティーの考慮事項 を参照してください。

追加の IMS 接続の同じキュー・マネージャーへの追加

IMS 接続を同じキュー・マネージャーに追加するには、2 番目のストレージ・クラス (STGCLASS) を定義して、新しい IMS を指すようにする必要があります。詳しくは、DEFINE STGCLASS を参照してください。

重要 :

- 1 つのローカル・キューが 2 つのストレージ・クラスを指すことはできません。
- 1 つのストレージ・クラスが 2 つの IMS ブリッジを指すことはできません。
- IBM MQ と IMS は、同じ XCF グループに属している必要があります。 CSQ6SYSP マクロの OTMACON キーワードを使用して、システム・パラメーター・ロード・モジュール内でこれらのパラメーターを調整します。

  詳細については、CSQ6SYSP の使用を参照してください。

関連概念

IBM MQ と IMS

1019 ページの『IMS アダプターのセットアップ』

IMS 内で IBM MQ を使用するには、IBM MQ-IMS アダプター(一般的には IMS アダプターと呼ばれます)が必要です。

IBM MQ for z/OS 上の IMS および IMS ブリッジ・アプリケーション

z/OS IBM MQ と CICS の使用

CICS で IBM MQ を使用するには、IBM MQ CICS アダプターと、オプションで IBM MQ CICS bridge コンポーネントを構成する必要があります。

IBM MQ CICS アダプターおよび IBM MQ CICS bridge コンポーネントの構成について詳しくは、CICS 資料の 「MQ セクションを参照してください。

関連概念

IBM MQ と CICS

1019 ページの『IBM MQ と IMS の使用』

IBM MQ-IMS アダプター、および IBM MQ-IMS ブリッジは、IBM MQ と IMS との対話を可能にする二つのコンポーネントです。

関連資料

1027 ページの『言語環境プログラムまたは z/OS 呼び出し可能サービスのアップグレードおよび保守の適用』

必要なアクションは、CALLLIBS または LINK のどちらを使用するか、および使用する SMP/E のバージョンによって異なります。

z/OS 言語環境プログラムまたは z/OS 呼び出し可能サービスのアップグレードおよび保守の適用

必要なアクションは、CALLLIBS または LINK のどちらを使用するか、および使用する SMP/E のバージョンによって異なります。

次の表には、以下の製品のレベルをアップグレードする場合、または保守を適用する場合に、IBM MQ for z/OS に行う必要がある事項を示しています。

- Language Environment
- z/OS 呼び出し可能サービス (例えば、APPC および RRS)

表 *85.* 保守が適用されたか、製品が新しいリリースにアップグレードされた場合

製品	**CALLLIBS および SMP/E V3R2 以降を使用する場合の処置** **注 : Language Environmentp、および呼び出し可能サービスのために異なるジョブを実行する必要はありません。 1 つのジョブを実行するだけで済みます。**	**LINK を使用する場合の処置**
Language Environment	1. ターゲット・ゾーンに対する境界を SMP/E 上に設定する。 2. SMPCNTL カードに LINK LMODS CALLLIBS を指定する。 また、CHECK、RETRY(YES)、RC など、他のパラメーターも指定できる。 詳しくは、「*z/OS SMP/E コマンド* 」を参照してください。 3. SMP/E ジョブを実行する。	処置は不要。ただし、SMP/E ゾーンが自動再リンクにセットアップされ、CSQ8SLDQ ジョブが実行されている。

表 *85.* 保守が適用されたか、製品が新しいリリースにアップグレードされた場合 *(*続き*)*

製品	**CALLLIBS および SMP/E V3R2 以降を使用する場合の処置** **注 : Language Environmentp、および呼び出し可能サービスのために異なるジョブを実行する必要はありません。 1 つのジョブを実行するだけで済みます。**	**LINK を使用する場合の処置**
呼び出し可能サービス	1. ターゲット・ゾーンに対する境界を SMP/E 上に設定する。 2. SMPCNTL カードに LINK LMODS CALLLIBS を指定する。 また、CHECK、RETRY(YES)、RC など、他のパラメーターも指定できる。 詳しくは、「*z/OS SMP/E* コマンド」を参照してください。 3. SMP/E ジョブを実行する。	処置は不要。ただし、SMP/E ゾーンが自動再リンクにセットアップされ、CSQ8SLDQ ジョブが実行されている。

表 *86.* 新しい *SMP/E* 環境およびライブラリーで、製品のいずれかが新しいリリースに更新された場合

製品	**CALLLIBS および SMP/E V3R2 以降を使用する場合の処置** **注 : Language Environmentp、および呼び出し可能サービスのために異なる 3 つのジョブを実行する必要はありません。 両製品に対して 1 つのジョブを実行するだけで済みます。**	**LINK を使用する場合の処置**
Language Environment	1. SCEELKED および SCEESPC のための DDDEF を新しいライブラリーを指すように変更する。 2. ターゲット・ゾーンに対する境界を SMP/E 上に設定する。 3. SMPCNTL カードに LINK LMODS CALLLIBS を指定する。 また、CHECK、 RETRY(YES)、RC など、他のパラメーターも指定できる。 詳しくは、「*z/OS SMP/E* コマンド」を参照してください。 4. SMP/E ジョブを実行する。	1. IBM MQ for z/OS 宛先ゾーンの中で次の LMOD 項目の XZMOD 副項目を削除する。 CMQXDCST, CMQXRCTL, CMQXSUPR, CSQCBE00, CSQCBE30, CSQCBP00, CSQCBP10, CSQCBR00, CSQUCVX, CSQUDLQH, CSQVXPCB, CSQVXSPT, CSQXDCST, CSQXRCTL, CSQXSUPR, CSQXTCMI, CSQXTCP, CSQXTNSV, CSQ7DRPS, IMQB23IC, IMQB23IM, IMQB23IR, IMQS23IC, IMQS23IM, IMQS23IR 2. IBM MQ ゾーンと言語環境プログラム・ゾーンの間に適切な ZONEINDEX をセットアップする。 3. LINK コマンドの FROMZONE パラメーター上で新しいゾーンを参照するよう CSQ8SLDQ を調整する。 CSQ8SLDQ は SCSQINST ライブラリーで見つけることができる。 4. CSQ8SLDQ を実行する。

表 86. 新しい *SMP/E* 環境およびライブラリーで、製品のいずれかが新しいリリースに更新された場合 (続き)

製品	**CALLLIBS および SMP/E V3R2 以降を使用する場合の処置** **注 : Language Environmentp、および呼び出し可能サービスのために異なる 3 つのジョブを実行する必要はありません。 両製品に対して 1 つのジョブを実行するだけで済みます。**	**LINK を使用する場合の処置**
呼び出し可能サービス	1. CSSLIB のための DDDEF を、新しいライブラリーを指すように変更する。 2. ターゲット・ゾーンに対する境界を SMP/E 上に設定する。 3. SMPCNTL カードに LINK LMODS CALLLIBS を指定する。 また、CHECK、 RETRY(YES)、RC など、他のパラメーターも指定できる。 詳しくは、「*z/OS SMP/E コマンド*」を参照してください。 4. SMP/E ジョブを実行する。	1. IBM MQ for z/OS 宛先ゾーンの中で次の LMOD 項目の XZMOD 副項目を削除する。 CMQXRCTL、CMQXSUPR、 CSQBSRV、CSQILPLM、CSQXJST、CSQXRCTL、 CSQXSUPR、CSQ3AMGP、CSQ3EPX、 CSQ3REPL 2. IBM MQ ゾーンと呼び出し可能サービス・ゾーンとの間に適切な ZONEINDEX をセットアップする。 3. LINK コマンドの FROMZONE パラメーター上で新しいゾーンを参照するよう CSQ8SLDQ を調整する。 CSQ8SLDQ は SCSQINST ライブラリーで見つけることができる。 4. CSQ8SLDQ を実行する。

CALLLIB を使用するときにモジュールを再リンクするジョブの例については、1029 ページの『LINK CALLLIBS ジョブの実行』を参照してください。

z/OS LINK CALLLIBS ジョブの実行

CALLLIB を使用するときにモジュールを再リンクするジョブの例。

次の例は、SMP/E V3R2 システムで CALLLIB を使用するときにモジュールを再リンクするジョブの例です。 JOBCARD を用意し、IBM MQ for z/OS が入っている SMP/E CSI のデータ・セット名を指定する必要があります。

```
//*******************************************************************
//* RUN LINK CALLLIBS.
//*******************************************************************
//CALLLIBS EXEC PGM=GIMSMP,REGION=4096K
//SMPCSI  DD DSN=your.csi
//       DISP=SHR
//SYSPRINT DD SYSOUT=*
//SMPCNTL DD *
SET BDY(TZONE).
LINK LMODS CALLLIBS .
/*
```

図 *122. SMP/E LINK CALLLIBS* ジョブの例

z/OS IMS における OTMA 出口の使用

IBM MQ for z/OS で IMS Open Transaction Manager Access 出口を使用する場合は、このトピックを使用してください。

IMS トランザクションからの出力を IBM MQ へ送る場合、そのトランザクションが IBM MQ の中で生じたものでなければ、1 つ以上の IMS OTMA 出口をコーディングする必要があります。

同様に、OTMA 以外の宛先へ出力を送る場合、トランザクションが IBM MQ の中で生じたものであれば、1 つ以上の IMS OTMA 出口をコーディングする必要があります。

IMS では、以下の出口を使用して、 IMS と IBM MQ の間の処理をカスタマイズできます。

- OTMA 事前経路指定出口
- 宛先解決ユーザー (DRU) 出口

OTMA 出口名

事前経路指定出口の名前は DFSYPRX0 にする必要があります。 DRU 出口の名前は、IMS の既存モジュール名と競合しなければ、どのような名前でも構いません。

宛先解決ユーザー出口名の指定

CSQ6SYSP マクロの OTMACON キーワードにある *Druexit* パラメーターを使用して、IMS によって実行される OTMA DRU 出口の名前を指定することができます。

オブジェクトの識別を単純化するために、DRU0xxxx の命名規則の使用を検討してください。ここで、xxxx は使用中の IBM MQ キュー・マネージャーの名前です。

DRU 出口の名前を OTMACON パラメーターで指定しなければ、デフォルトは DFSYDRU0 になります。詳しくは、 DFSYDRU0 を参照してください。

IMS 宛先の命名規則

IMS プログラムから出力を送る場合、その宛先について命名規則が必要です。 この宛先は、IMS アプリケーションの CHNG 呼び出しに設定されるものであるか、IMS PSB に事前設定されているものです。

OTMA 出口のサンプル・シナリオ

IMS 用の事前経路指定出口および宛先経路指定出口の例については、以下のトピックを使用してください。

- 1030 ページの『事前経路指定出口 DFSYPRX0』
- 1031 ページの『宛先解決ユーザー出口』

識別を単純化するために、OTMA 宛先名を IBM MQ キュー・マネージャー名と同じにすることをお勧めします (例えば、繰り返される IBM MQ キュー・マネージャー名)。 この場合、IBM MQ キュー・マネージャー名が「**VCPE**」であれば、CHNG 呼び出しによって設定される宛先は「**VCPEVCPE**」になります。

関連概念

IBM MQ と IMS

1019 ページの『IBM MQ と IMS の使用』

IBM MQ-IMS アダプター、および IBM MQ-IMS ブリッジは、IBM MQ と IMS との対話を可能にする二つのコンポーネントです。

IBM MQ for z/OS 上の IMS および IMS ブリッジ・アプリケーション

z/OS 事前経路指定出口 DFSYPRX0

このトピックには、IMS における OTMA の事前経路指定出口のサンプルが含まれます。

最初に、事前経路指定出口 DFSYPRX0 をコード化する必要があります。 IMS によってこのルーチンに渡されるパラメーターについては、 OTMA 宛先解決ユーザー出口 (DFSYPRX0 およびその他の OTMAYPRX タイプの出口) を参照してください。

この出口は、メッセージの宛先が既知の OTMA 宛先 (例では VCPEVCPE) に向けられたものかどうかをテストします。 もしそうであれば、メッセージを送っているトランザクションが OTMA で生じたものかどうかを出口で検査する必要があります。 メッセージが OTMA で生じたものであれば、それは OTMA ヘッダーを持っているので、レジスター 15 を 0 に設定して、DFSYPRX0 を終了します。

- メッセージを送っているトランザクションが OTMA の中で生じたものでなければ、クライアント名を有効な OTMA クライアントに設定する必要があります。 これは、メッセージを送る IBM MQ キュー・マネージャーの XCF メンバー名です。 クライアント名を (CSQ6SYSP マクロの OTMACON パラメーターで)

キュー・マネージャー名に設定する必要があります。 これはデフォルトです。 次に、レジスター 15 を 4 に設定して、DFSYPRX0 を終了します。

- メッセージを送っているトランザクションが OTMA で生じたものであり、OTMA 以外の宛先であれば、レジスター 15 を 8 に設定して終了します。
- その他の場合、レジスター 15 は 0 に設定します。

OTMA クライアントの名前を、IMS が認知できない名前に設定すると、アプリケーションの CHNG 呼び出しまたは ISRT 呼び出しは A1 状況コードを戻します。

IMS システムが複数の IBM MQ キュー・マネージャーと通信している場合、ロジックを各 IBM MQ キュー・マネージャーごとに繰り返してください。

1031 ページの図 123 にサンプル・アセンブラー・コードが示されています:

```
TITLE 'DFSYPRX0: OTMA PRE-ROUTING USER EXIT'
DFSYPRX0 CSECT
DFSYPRX0 AMODE 31
DFSYPRX0 RMODE ANY
*
SAVE (14,12),,DFSYPRX0&SYSDATE&SYSTIME
SPACE 2
LR  R12,R15        MODULE ADDRESSABILITY
USING DFSYPRX0,R12
*
L   R2,12(,R1)      R2 -> OTMA PREROUTE PARMS
*
LA  R3,48(,R2)      R3 AT ORIGINAL OTMA CLIENT (IF ANY)
CLC  0(16,R3),=XL16'00'  OTMA ORIG?
BNE  OTMAIN        YES, GO TO THAT CODE
*
NOOTMAIN DS 0H            NOT OTMA INPUT
LA  R5,8(,R2)       R5 IS AT THE DESTINATION NAME
CLC  0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE  EXIT0         NO, NORMAL PROCESSING
*
L   R4,80(,R2)      R4 AT ADDR OF OTMA CLIENT
MVC  0(16,R4),=CL16'VCPE' CLIENT OVERRIDE
B   EXIT4         AND EXIT
*
OTMAIN  DS 0H           OTMA INPUT
LA  R5,8(,R2)       R5 IS AT THE DESTINATION NAME
CLC  0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE  EXIT8         NO, NORMAL PROCESSING
```

```
*
EXIT0  DS 0H
LA  R15,0         RC = 0
B   BYEBYE
*
EXIT4  DS 0H
LA  R15,4         RC = 4
B   BYEBYE
*
EXIT8  DS 0H
LA  R15,8         RC = 8
B   BYEBYE
*
BYEBYE  DS 0H
RETURN (14,12),,RC=(15)   RETURN WITH RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
END
```

図 *123.* アセンブラーで書いた *OTMA* 事前経路指定出口のサンプル

z/OS 宛先解決ユーザー出口

このトピックには、IMS の宛先解決ユーザー出口のサンプルが含まれます。

DFSYPRX0 において、レジスター 15 を 4 に設定した場合、トランザクションのソースが OTMA であり、**かつ** レジスター 15 を 0 に設定すると、DRU 出口が呼び出されます。 この例では、DRU 出口の名前は DRU0VCPE です。

DRU 出口は、宛先が VCPEVCPE であるかどうかを検査します。 その場合には、OTMA ユーザー・データ (OTMA 接頭部の中にある) を次のように設定します。

オフセット
OTMA ユーザー・データ

(decimal)

0
OTMA ユーザー・データの長さ (この例では、334)

2
MQMD

326
様式化する応答

これらのオフセットは、IBM MQ - IMS ブリッジが該当する情報の存在を予想する場所です。

DRU 出口は、できるだけ単純でなければなりません。 したがって、このサンプルでは、特定の IBM MQ キュー・マネージャーに対して、IMS で発生するすべてのメッセージは、同じキュー IBM MQ キューに入れられます。

持続メッセージにする必要がある場合は、IMS で同期化トランザクション・パイプを使用しなければなりません。 このためには、DRU 出口で OUTPUT フラグを設定する必要があります。 詳しくは、IBM MQ の同期化 T パイプの指定 を参照してください。

このキューを処理するために IBM MQ アプリケーションを作成し、MQMD 構造、MQIIH 構造 (もしあれば)、またはユーザー・データからの情報を使用して、各メッセージをその宛先へ送ります。

サンプルのアセンブラー DRU 出口が 1033 ページの図 124 に示されています。

```
TITLE 'DRU0VCPE: OTMA DESTINATION RESOLUTION USER EXIT'
DRU0VCPE CSECT
DRU0VCPE AMODE 31
DRU0VCPE RMODE ANY
*
SAVE (14,12),,DRU0VCPE&SYSDATE&SYSTIME
SPACE 2
LR  R12,R15            MODULE ADDRESSABILITY
USING DRU0VCPE,R12
*
L   R2,12(,R1)           R2 -> OTMA DRU PARMS
*
L   R5,88(,R2)           R5 ADDR OF OTMA USERDATA
LA  R6,2(,R5)           R6 ADDR OF MQMD
USING MQMD,R6            AS A BASE
*
LA  R4,MQMD_LENGTH+10       SET THE OTMA USERDATA LEN
STH  R4,0(,R5)           = LL + MQMD + 8
*                       CLEAR REST OF USERDATA
MVI  0(R6),X'00'          ...NULL FIRST BYTE
MVC  1(255,R6),0(R6)        ...AND PROPAGATE IT
MVC  256(MQMD_LENGTH-256+8,R6),255(R6) ...AND PROPAGATE IT
*
VCPE   DS  0H
CLC  44(16,R2),=CL16'VCPE'     IS DESTINATION VCPE?
BNE  EXIT4             NO, THEN DEST IS NON-OTMA
MVC  MQMD_REPLYTOQ,=CL48'IMS.BRIDGE.UNSOLICITED.QUEUE'
MVC  MQMD_REPLYTOQMGR,=CL48'VCPE'  SET QNAME AND QMGRNAME
MVC  MQMD_FORMAT,MQFMT_IMS     SET MQMD FORMAT NAME
MVC  MQMD_LENGTH(8,R6),MQFMT_IMS_VAR_STRING
*                        SET REPLYTO FORMAT NAME
B   EXIT0
*
EXIT0  DS  0H
LA  R15,0             SET RC TO OTMA PROCESS
B   BYEBYE             AND EXIT
*
EXIT4  DS  0H
LA  R15,4             SET RC TO NON-OTMA
B   BYEBYE             AND EXIT
*
BYEBYE  DS  0H
RETURN (14,12),,RC=(15)       RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
CMQA  EQUONLY=NO
CMQMDA DSECT=YES
SPACE 2
END
```

図 *124.* アセンブラーで書いた *DRU* 出口のサンプル

z/OS IBMz/OSMF を使用して自動化します IBM MQ

IBM z/OS Management Facility(z/OSMF)は、統合されたユーザー・アシスタンスを使用して、タスク指向の Web ブラウザー・ベースのユーザー・インターフェースにシステム管理機能を提供するため、メインフレーム z/OS システムの日常の運用および管理をより簡単に管理できます。

従来のタスクの一部を簡素化し、その他を自動化することにより、z/OSMF は z/OS システム管理のいくつかの領域の単純化に貢献します。

ユーザー提供のポータルから、ボタンをクリックするだけで、リソースをプロビジョンまたはプロビジョン解除することができます。 z/OSMF には、このタスクに役立つ REST API が用意されています。

z/OSMF に付属するサンプルのマーケットプレイス・ポータルを使用して、リソースのプロビジョンおよびプロビジョン解除を行うこともできます。 あるいは、経験豊富なユーザーであれば z/OSMF の Web ユーザー・インターフェース (WUI) を使用することもできます。

このセクションでは、z/OSMF を理解していることを前提としていますが、z/OSMF を理解していない場合は、z/OSMF の概要を読んでください。 また、このセクションには z/OSMF WUI オンライン・ヘルプからもアクセスできます。

以下のような z/OS クラウド構成を理解する必要があります。

- クラウド・プロビジョニング- リソース管理サービス
- ワークロード管理-詳しくは、「 IBM z/OS Management Facility プログラミング・ガイド 」を参照してください。
- 入門- 入門チュートリアル-クラウド を参照してください。

z/OSMF 2.2 で、役割ベースのアクティビティーとタスクが導入されたため、以下のような概念を理解することが重要です。

ドメイン
管理者
承認者
テナント
テンプレート
インスタンス
ワークフロー

などです。

サンプル IBM MQ z/OSMF ワークフローおよび関連ファイルが提供されています。これらを IBM MQ for z/OS UNIX System Services Components フィーチャーの一部としてインストールできます。 フィーチャー、およびディレクトリーとファイルの構造に関するインストール・プロセスについては、IBM MQ for z/OSProgram Directory に記載されています。プログラム・ディレクトリのダウンロードリンクについては、IBM MQ for z/OS Program Directory PDF files を参照。

サンプル・ワークフローは XML で記述されており、IBM MQ キュー・マネージャー、チャネル・イニシエーター、ローカル・キューのプロビジョン (作成) またはプロビジョン解除 (破壊) を自動化する方法、およびプロビジョンされた IBM MQ リソースに対してアクションを実行する方法を示します。 これらのワークフロー内のステップでは、ジョブ (JCL) の実行依頼、REXX 実行ファイルの実行、シェル・スクリプトの処理、または REST API 呼び出しの実行を行います。

サンプルは、z/OSMF を使用して実現できる機能のタイプを示すように設計されています。 リソースのプロビジョンには通常、z/OSMF ワークフローが使用されること、PUT メッセージや GET メッセージのようなアクションは本質的に IBM MQ アプリケーションを使用して行われることが予期されています。

ワークフロー変数プロパティーが設定されている場合 (後続のセクションで説明します) は、提供されたままのサンプル・ワークフローを実行できます。あるいは、必要に応じてカスタマイズすることもできます。追加の機能を実行するために独自のワークフローを作成することもできます。 サンプル・ワークフローを実行する前に、以下を参照してください。

- 1035 ページの『z/OSMF の前提条件』
- 1036 ページの『セキュリティー設定』
- 1039 ページの『制限』

サンプル・ワークフロー・アプリケーションは以下の目的で提供されています。

- 1040 ページの『IBM MQ キュー・マネージャーのプロビジョンまたはプロビジョン解除の自動化、およびプロビジョンされたキュー・マネージャーに対するアクションの実行』
- 1041 ページの『IBM MQ ローカル・キューのプロビジョンまたはプロビジョン解除の自動化、およびプロビジョンされたキューに対するアクションの実行』.

関連概念

914 ページの『IBM MQ for z/OS の設定』
このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

z/OS z/OSMF の前提条件

IBM MQ で IBM z/OS Management Facility (z/OSMF) を実行するために必要な前提条件

IBM MQ for z/OS 9.1.0 で出荷されるワークフローは、z/OSMF の新機能を活用します。この機能は、z/OS2.1 と 2.2 の両方の APAR を介して提供されます。 詳細を、以下のテキストで説明します。

1. IBM z/OS Management Facility 2.2 が正しくインストールされ、構成されている。 セキュリティーを有効にして実行している場合は、z/OSMF に説明があるように、すべてのセキュリティー設定が構成されている必要があります。
2. 以下に対する APAR がインストールされている。

   z/OS 2.1

   - PI71068
   - PI71079
   - PI71082
   - PI71084
   - OA50130

   z/OS 2.2

   - PI70526
   - PI70521
   - PI70527
   - PI67839
   - PI70767
   - PI46315
   - OA49081
   - OA49802
   - OA50130
3. z/OSMF のエンジェル・プロセス (必要な場合) およびサーバー・プロセスが構成されている。
4. z/OS クラウド環境が構成されている (上記で少し触れていますが z/OSMF に説明があります)
5. IBM MQ for z/OS 9.0.1 がインストールされていて、製品ロード・ライブラリーが使用可能である。
6. 以下の IBM MQ キュー・マネージャーのカスタマイズ・タスクが実行済みである。

タスク	説明
1	z/OS システム・パラメーターを識別する
2	IBM MQ ロード・ライブラリーの APF 許可を行う
3	z/OS リンク・リストおよび LPA を更新する
4	z/OS プログラム特性表を更新する

7. サンプル・ワークフローおよび関連ファイルが適切な z/OS UNIX System Services (z/OS UNIX) ディレクトリーにインストールされている。
8. `/tmp` z/OS UNIX ディレクトリーは使用可能です。これは provision.xml のワークフローがディレクトリー内に、一時ファイルを作成する可能性があるためです。 ファイルが作成された場合、通常は使用された後にワークフローによってそのファイルは削除されます。
9. `deprovision.xml` ファイル内に、REXX 実行ファイル CSQ4ZWS1.rexx および CSQ4ZWS2.rexx を呼び出すステップが含まれている。 これらの実行ファイルは、キュー・マネージャーおよびチャネル・イニシエーター・サブシステムが停止するまで待機します。つまり、実行ファイルが z/OS UNIX **SLEEP** コマンドをシステム呼び出しとして起動します。

ご使用の z/OS UNIX 構成によっては、コードされたとおりに **SLEEP** コマンドが機能しないことがあります。 処理中に **SLEEP** コマンドが見つからないことを示すエラーが発生した場合は、実行ファイル CSQ4ZWS1.rexx および CSQ4ZWS2.rexx 内の以下の行を置き換えてみてください。

```
CALL SYSCALLS('ON')             /* Enable z/OS UNIX calls    */
ADDRESS SYSCALL
"SLEEP" 10                           /* Sleep for 10 seconds */
CALL SYSCALLS 'OFF'             /* Disable z/OS UNIX calls    */
```

を次のコマンドに置き換えてください:

```
'sleep' 10
```

次に、Open MVS (OMVS) **env** コマンドを実行して PATH 環境変数の設定値を確認します。 **sleep** コマンドが含まれているディレクトリーが PATH に定義されていることを確認します。 通常、**sleep** コマンドは /bin ディレクトリーにあることに留意してください。

10. z/OSMF が始動していることを確認する。

    z/OSMF のエンジェル・プロセスとサーバー・プロセスの両方が開始されていて、z/OSMF Web ユーザー・インターフェース (WUI) が稼働している必要があります。 詳細については、『Liberty profile: Process types on z/OS』を参照してください。

    REST API を使用してワークフローを起動する場合でも、z/OSMF WUI を開始する必要があります。 z/OSMF WUI は、ワークフローの作成と実行をモニターするのに役立ちます。

関連概念

1033 ページの『IBMz/OSMF を使用して自動化します IBM MQ』
IBM z/OS Management Facility(z/OSMF)は、統合されたユーザー・アシスタンスを使用して、タスク指向の Web ブラウザー・ベースのユーザー・インターフェースにシステム管理機能を提供するため、メインフレーム z/OS システムの日常の運用および管理をより簡単に管理できます。

z/OS セキュリティー設定

z/OSMF の実行に必要なセキュリティー設定です。

以下のユーザー ID 変数プロパティーは、properties ファイルに定義されています。 詳細は、1043 ページの『ワークフローの実行』を参照してください。

ユーザー ID プロパティー	説明
CSQ_USERID	ワークフロー・ステップの実行に使用するユーザー ID。 ただし、選択されたステップ (一般に、昇格された権限レベルが必要なもの) は、下記にリストしている一連の **CSQ_ADMIN_*** のユーザー ID の設定を基に別のユーザー ID で実行されることに注意してください。 使用するユーザー ID は、ワークフローの各ステップの **runAsUser** プロパティーで指定します。
CSQ_ADMIN_APF_USERID	キュー・マネージャーのシステム・パラメーター・モジュールを含むロード・ライブラリーを APF が許可するときに使用するユーザー ID。
CSQ_APF_APPROVAL_ID	データ・セット APF 許可ステップをユーザー CSQ_ADMIN_APF_USERID として実行することをユーザーに許可するために使用する承認 ID 。
CSQ_ADMIN_CONSOLE_USERID	z/OS コンソール・コマンドを発行するステップの実行に使用するユーザー ID。 **重要 :** このユーザー ID は、開始タスク・プロファイル (MVS.START.STC. *) への UPDATE アクセスを許可されている必要があります。 OPERCMDS クラスで 詳しくは、 z/OS 資料の「オペレーター・コマンドの使用の制御 」を参照してください。

ユーザー ID プロパティー	説明
CSQ_CONSOLE_APPROVAL_ID	z/OS コンソール・コマンドを発行するステップをユーザー CSQ_ADMIN_CONSOLE_USERID として実行することをユーザーに許可するために使用する承認 ID。
CSQ_ADMIN_SAF_USERID	SAF コマンドを実行するときに使用するユーザー ID。
CSQ_SAF_APPROVAL_ID	SAF コマンド・ステップをユーザー CSQ_ADMIN_SAF_USERID として実行することをユーザーに許可するために使用する承認 ID。
CSQ_ADMIN_SSI_USERID	SETSSI コマンドを実行して z/OS にプロビジョンされるサブシステムを識別するときに使用するユーザー ID。
CSQ_SSI_APPROVAL_ID	SETSSI コマンド・ステップをユーザー CSQ_ADMIN_SSI_USERID として実行することをユーザーに許可するために使用する承認 ID。

注 : ワークフローのプロビジョンおよびプロビジョン解除の実行に使用するユーザー ID には、以下にリストする十分な権限がなければなりません。

1. キュー・マネージャーのプロビジョンおよびプロビジョン解除のワークフローは、SETPROG コマンドを使用してデータ・セットの APF 許可を行います。 ユーザー ID をプロパティー CSQ_ADMIN_APF_USERID に設定するか、ワークフローの実行に使用するユーザー ID にこのコマンドを実行する権限を付与する必要があります。 これを行うには、以下のコマンドを実行します。

```
PERMIT MVS.SETPROG CLASS(OPERCMDS) ID(value of CSQ_ADMIN_APF_USERID) ACCESS(UPDATE)
```

   注 : SETPROG コマンドは z/OS システムの IPL 後は持続されない可能性があるため、IPL 後は次の SETPROG コマンドを手動で実行する必要が生じることがあります。

```
SETPROG APF,ADD,DSN=value of CSQ_AUTH_LIB_HLQ.value of CSQ_SSID.APF.LOAD,SMS
```

   SETPROG コマンドの詳細については、RACF を使用した APF リストの制御を参照してください。

   さらに、どのライブラリーに APF 許可を付与するかを制御するために FACILITY クラスを有効にした可能性があるため、次のコマンドを実行する必要があります。

```
PERMIT CSVAPF.libname CLASS(FACILITY) ID(value of CSQ_ADMIN_APF_USERID)
       ACCESS(UPDATE)
```

2. キュー・マネージャーのプロビジョン・ワークフローのステップで、SETSSI コマンドを発行して、z/OS に対して IBM MQ サブシステムを識別します。 プロパティー CSQ_ADMIN_SSI_USERID に設定されているユーザー ID が、このコマンドの使用を許可されている必要があります。 これを行うには、以下のコマンドを実行します。

```
PERMIT MVS.SETSSI.ADD CLASS(OPERCMDS) ID(value of CSQ_ADMIN_SSI_USERID)
        ACCESS(CONTROL)
```

   注 : SETSSI コマンドを使用して z/OS に対して識別されたサブシステムは、z/OS システムの IPL 後は持続しません。 そのため、IPL 後は次の SETSSI コマンドを手動で実行しなければならないことがあります。

```
SETSSI  ADD,S='value of CSQ_SSID',I=CSQ3INI,
       P='CSQ3EPX,value of CSQ_CMD_PFX,S'
```

   SETSSI コマンドの詳細については、SETSSI コマンドを参照してください。

3. ワークフローはキュー・マネージャー・コマンドを実行するので、セキュリティーを有効にする予定の場合は、プロパティー CSQ_ADMIN_RACF_USERID に設定されているユーザー ID (またはワークフローの実行に使用されているユーザー ID) に 対し、(使用するクラスに応じて) MQADMIN クラスまたは

MXADMIN クラスの CLAUTH (クライアント認証) 権限を (使用するクラスに応じて) 付与する必要があります。 これにより、このユーザー ID がこれらのクラスに対してセキュリティー・プロファイルを定義できるようになります。 これを行うには、以下のコマンドを実行します。

```
ALTUSR value of CSQ_ADMIN_RACF_USERID CLAUTH(MQADMIN)
```

CLAUTH について詳しくは、 CLAUTH (クラス権限) 属性を参照してください。

4. deprovision.xml ワークフローは z/OS コマンド (例えばジョブの DISPLAY ACTIVE、サブシステムの CANCEL および FORCE) を実行するので、プロパティー CSQ_ADMIN_CONSOLE_USERID に設定するユーザー ID (あるいはワークフローの実行に使用するユーザー ID) は、そのようなコマンドを実行するための適切な権限を持っている必要があります。
5. ソフトウェア・サービス・タスクのテンプレート・テーブルを使用してキュー・マネージャー・インスタンスを要求しているユーザーは、z/OSMF および構成アシスタントにアクセスするための権限 (z/OSMF で定義されている) を持っている必要があります。
6. キュー・マネージャーをプロビジョンするコンシューマーのユーザー ID には、変数 CSQ_PROC_LIB で定義された PROCLIB データ・セットのメンバーを追加および削除する権限が必要です。
7. キューをプロビジョンする前に、キュー・マネージャーをプロビジョンする必要があります。
8. queueLoad.xml ワークフローおよび queueOffload.xml ワークフローを使用するには、使用するデータ・セットをあらかじめ定義しておく必要があります。 また、これらのワークフローの実行に使用されるユーザー ID は、データ・セットに対する UPDATE 権限が付与されている必要があります。
9. キュー・マネージャーの provision.xml ワークフローのステップは、サブシステムのセキュリティーを無効にします。 IBM MQ リソースを保護するための適切なセキュリティー・コマンドを追加することにより、ジョブ csq4znse.jcl を変更して、サブシステム・セキュリティーを使用可能にする。 ただし、コマンドを追加する場合は、deprovision.xml ワークフローによって実行依頼される csq4dse.jcl 内のセキュリティー権限を削除するためのコマンドも追加する必要があります。

   注: このステップは、RACF セキュリティー・コマンドを発行します。 別のセキュリティー製品を使用する場合は、このステップを変更して、そのセキュリティー製品の適切なコマンドを発行してください。

ネットワーク要件

キュー・マネージャーのテンプレートおよびそのテンプレート用のリソースを追加する場合は、「**ネットワーク・リソース・プールの作成 (Create network resource pool)**」をクリックします。 これによって、そのテンプレートのネットワーク・リソースを持つリソース・プールが作成されます。

ネットワーク管理者は、構成アシスタントでそのテンプレートに割り振られるポート数の限度を定義して、このネットワーク・リソース・プール定義を完了する必要があります。

provision.xml ワークフローは、テンプレート・インスタンスごとに、範囲内にポートを割り振り、ポートで listen するリスナーを開始します。

IBM ワークロード・マネージャーによる分類

WLM を使用してキュー・マネージャーおよびチャネル・イニシエーターのアドレス・スペースを分類する場合は、キュー・マネージャーのプロビジョニング用テンプレートを追加する際にこの分類を指定する必要があります。

分類するかどうかは、ファイル workflow_variables.properties に設定されているフラグ **CSQ_DEFINE_MSTR_WLM_RULE** および **CSQ_DEFINE_CHIN_WLM_RULE** によって制御されます。

WLM を使用した分類について詳しくは、「*z/OSMF* 構成ガイド」を参照してください。

関連概念

1035 ページの『z/OSMF の前提条件』
IBM MQ で IBM z/OS Management Facility (z/OSMF) を実行するために必要な前提条件

z/OS 制限

IBM MQ で z/OSMF を使用する場合の制限。

1. `provision.xml` ワークフローは現在、以下の強調表示されたキュー・マネージャーのカスタマイズ・タスクを自動化:

タスク	説明
1	z/OS システム・パラメーターを識別する
2	IBM MQ ロード・ライブラリーを APF 許可する **(provision.xml は一部のライブラリーの APF 許可を行います)**
3	z/OS リンク・リストおよび LPA を更新する
4	z/OS プログラム特性表を更新する
5	**IBM MQ サブシステムを z/OS に定義する**
6	**IBM MQ キュー・マネージャー用のプロシージャーを作成する**
7	**チャネル・イニシエーター用のプロシージャーを作成する**
8	**IBM MQ サブシステムを z/OSWLM サービス・クラスに定義する。**
9	カップリング・ファシリティー・オフロード・ストレージ環境の選択と設定
10	カップリング・ファシリティーのセットアップ
11	ESM セキュリティー管理を実施する
12	SYS1.PARMLIB メンバーを更新する
13	**初期設定入力データ・セットをカスタマイズする**
14	**ブートストラップ・データ・セットおよびログ・データ・セットを作成する**
15	**ページ・セットを定義する**
16	IBM MQ 項目を Db2 データ共有グループに追加する
17	**システム・パラメーター・モジュールを調整する (一部)**
18	**チャネル・イニシエーター・パラメーターを調整する (一部)**
19	バッチ、TSO、および RRS アダプターをセットアップする
20	操作パネルおよび制御パネルをセットアップする
21	IBM MQ ダンプ・フォーマット・メンバーを組み込む
22	通知メッセージを抑止する
23	Advanced Message Security 用にシステム DIAG メンバーを更新する
24	Advanced Message Security 用のプロシージャーを作成する
25	開始タスク・ユーザー Advanced Message Security をセットアップする
26	Advanced Message Security のセキュリティー管理者に RACDCERT 権限を付与する
27	Advanced Message Security のためのリソース・アクセス権をユーザーに付与する

2. 太字のテキストで強調表示されていないカスタマイズ・タスクは、必要な場合に手動で実行する必要があります。

3. サンプルの INP1 メンバー および INP2 メンバーは現在もそのまま使用されます。 必要な場合は、追加のプロパティーを定義して、これらのメンバーによって定義されたリソースを制御できます。
4. properties ファイルにリストされている特定のプロパティーに関連するコメントは、それらのプロパティーの使用に関する制約を示しています。 詳細は、1043 ページの『ワークフローの実行』を参照してください。

関連概念

1036 ページの『セキュリティー設定』
z/OSMF の実行に必要なセキュリティー設定です。

z/OS IBM MQ オブジェクトのプロビジョニングの自動化

キュー・マネージャーおよびローカル・キューのプロビジョニングを自動化するためのサンプルが用意されています。

IBM MQ キュー・マネージャーのプロビジョンまたはプロビジョン解除の自動化、およびプロビジョンされたキュー・マネージャーに対するアクションの実行

以下のキュー・マネージャー固有のサンプル z/OSMF ワークフローが用意されています。

ワークフロー名	説明
provision.xml	IBM MQ for z/OS キュー・マネージャーをプロビジョンする このサンプル・ワークフローは以下のことを行います。 • キュー・マネージャーに必要なシステム・リソースをプロビジョンします。 • チャネル・イニシエーターに必要なシステム・リソースをプロビジョンします。 • キュー・マネージャーを開始します (これにより、チャネル・イニシエーターおよび TCP/IP リスナーも開始します) • サンプル・キュー・マネージャーのインストール検証プログラムを実行します。 環境プロパティーを設定して、異なる特性を持つキュー・マネージャーのプロビジョニングを制御できます。 詳細については、1043 ページの『ワークフローの実行』を参照してください。 **注 :** このワークフローのテンプレートの追加を支援するために、マニフェスト・ファイル(`provision.mf`)が提供されています。 このファイルには追加の情報を含む **qaas_readme.pdf** ファイルへの参照が組み込まれています。テンプレートが追加されると、リンクを介してこのファイルにアクセスできるようになります。
deprovision.xml	IBM MQ for z/OS キュー・マネージャーをプロビジョン解除する このサンプル・ワークフローは以下のことを行います。 • チャネル・イニシエーターを停止します (これにより、TCP/IP リスナーも停止します) また、キュー・マネージャーも停止します。 • サブシステムが停止するのを待機します • すべてのチャネル・イニシエーターおよびキュー・マネージャーのシステム・リソースをプロビジョン解除します。
startQMgr.xml	IBM MQ for z/OS キュー・マネージャーを始動する このサンプル・ワークフローはキュー・マネージャーを開始します (これにより、チャネル・イニシエーターおよび TCP/IP リスナーも開始します)。

ワークフロー名	説明
stopQMgr.xml	IBM MQ for z/OS キュー・マネージャーを停止する このサンプル・ワークフローは、チャネル・イニシエーター (TCP/IP リスナーも停止します) とキュー・マネージャーを停止します。

各ワークフローは、1 つ以上のステップを実行します。 ワークフロー内のコメントは、各ステップで実行される機能について説明しています。 ステップには、データ入力を要求するだけのものもあれば、定義された機能を遂行するために JCL を実行依頼したり、REXX 実行ファイルやシェル・スクリプトを呼び出したり、REST API 呼び出しを実行したりするものもあります。

JCL または REXX 実行ファイルの正確な名前については、各ステップを参照してください。 ワークフローおよび関連する JCL または REXX 実行ファイルは、1 つ以上の変数 XML ファイルで宣言されている変数を参照します。 詳細は、1043 ページの『ワークフロー変数宣言ファイル』を参照してください。

deprovision、**startQMgr**、および **stopQMgr** は、プロビジョンされた IBM MQ for z/OS キュー・マネージャーに対するアクションとして実行できます。

IBM MQ ローカル・キューのプロビジョンまたはプロビジョン解除の自動化、およびプロビジョンされたキューに対するアクションの実行

以下のキュー固有のサンプル z/OSMF ワークフローが用意されています。

ワークフロー名	説明
defineQueue.xml	ローカル・キューを定義する このサンプル・ワークフローでは、z/OSMF ワークフローをどのように使用して、プロパティー設定に基づいた小規模、中規模、大規模のキューを定義できるかが示されています。 **注 :** このワークフローのテンプレートの追加を支援するために、マニフェスト・ファイル(`provision.mf`)が提供されています。 このファイルには追加の情報を含む **qaas_readme.pdf** ファイルへの参照が組み込まれています。 テンプレートが追加されると、リンクを介してこのファイルにアクセスできるようになります。
displayQueue.xml	ローカル・キューの選択した属性を表示する このサンプル・ワークフローは、ローカル・キューの選択した属性を表示します。 属性は、z/OSMF 変数 (変数名についてはワークフロー内のステップを参照してください) で返され、その後に表示されます。 必要な場合は、REST API を使用して変数の内容にアクセスすることができます。 詳細については、Cloud provisioning REST APIs を参照してください。また、z/OSMF ワークフロー・サービスも参照してください。
deleteQueue.xml	ローカル・キューを削除する このサンプル・ワークフローは、指定されたキュー・マネージャーのローカル・キューを削除します。
putQueue.xml	1 つ以上のメッセージをローカル・キューに書き込む。 このサンプル・ワークフローは 1 つ以上のメッセージをローカル・キューに書き込みます。 メッセージ・テキストは指定できますが、複数のメッセージをローカル・キューに同時に書き込むと、同じメッセージ・テキストが使用されます。
getQueue.xml	1 つ以上のメッセージをローカル・キューから読み取る。

ワークフロー名	説明
	このサンプル・ワークフローは 1 つ以上のメッセージをローカル・キューから読み取ります。 メッセージは、z/OSMF 変数 (変数名についてはワークフロー内のステップを参照してください) で返され、その後に表示されます。 必要な場合は、REST API を使用して変数の内容にアクセスすることができます。 詳細については、Cloud provisioning REST APIs を参照してください。また、z/OSMF ワークフロー・サービスも参照してください。
loadQueue.xml	データ・セットからローカル・キューにメッセージをロードする。 このサンプル・ワークフローは、メッセージをデータ・セットからローカル・キューにロードします。 データ・セットのデフォルト名は、プロパティーを設定することによって指定します。 詳細は、1043 ページの『ワークフローの実行』を参照してください。
offloadQueue.xml	メッセージをローカル・キューからデータ・セットにオフロードする。 このサンプル・ワークフローは、メッセージをローカル・キューからデータ・セットにオフロードします。 データ・セットのデフォルト名は、プロパティーを設定することによって指定します。 詳細は、1043 ページの『ワークフローの実行』を参照してください。
clearQueue.xml	ローカル・キューのメッセージを消去する。 このサンプル・ワークフローは、ローカル・キューのすべてのメッセージを消去 (削除) します。

注 :

1. **キューへの書き込み (PutQueue)** アクションを使用すると、いくつかのメッセージ・データを入力し、1 つ以上のメッセージをキューに書き込むことができます。 1 つの要求で複数のメッセージがキューに書き込まれた場合は、同じメッセージ・データが使用されます。
2. loadQueue.xml ワークフローおよび offloadQueue.xml ワークフローでは、SCSQLOAD ライブラリーの実行可能モジュール CSQUDMSG が、別名 QLOAD を使用して呼び出されます。 これは、 IBM MQ for Multiplatforms で使用可能な **dmpmqmsg** ユーティリティーと同等です。 したがって、データ・セットからキューにロードされるメッセージ、およびキューからデータ・セットにロードされるメッセージは、**dmpmqmsg** 形式である必要があります。

   サンプル JCL も、SCSQPROC のメンバー CSQ4QLOD として用意されています。

   loadQueue および offloadQueue アクションを試す最も簡単な方法は、以下のとおりです。

   a. **putQueue** を数回実行して、いくつかのメッセージをキューに書き込みます。
   b. **offloadQueue** を使用して、メッセージをキューからデータ・セットにオフロードします。
   c. 必要であれば、**clearQueue** を実行してキューからすべてのメッセージを除去します。
   d. **loadQueue** を使用して、メッセージをデータ・セットから同じまたは別のキューにロードします。

   dmpmqmsg 形式を確認するには、オフロード要求を発行した後に、データ・セットのコンテンツをブラウズしてください。
3. **displayQueue**、**deleteQueue**、 **putQueue**、**getQueue**、**loadQueue**、 **offloadQueue**、および **clearQueue** を、プロビジョンされた IBM MQ for z/OS ローカル・キューに対するアクションとして実行できます。 アクションおよびアクション・ファイルについて詳しくは、「*z/OSMF プログラミング・ガイド*」を参照してください。
4. デフォルトでは、ワークフローに関連するすべてのアクションが削除されます。 これは、ユーザーがワークフローをクリーンアップする必要性を最小限にするためです。

ただし、ここで問題なのは、アクションが何らかの出力を生成する場合です。 例えば、**displayQueue** アクションと **getQueue** アクションの両方が出力を生成します。

アクションの実行が完了するとすぐに、関連するワークフローが削除されるため、出力は表示されません。 したがって、 z/OS WUI からワークフロー・アクションを駆動する場合は、出力を表示するアクションごとに、 **< workflow>** タグの **cleanAfterComplete** フラグを *false* に設定する必要があります。

例えば、**displayQueue** の出力を表示するには、フラグを以下のように設定します。

```
<action name="displayQueue">
  <workflow cleanAfterComplete="false">
  ...

  ...
  </workflow>
</action>
```

ただし、これは、アクションに関連するワークフローを手動でクリーンアップしなければならないことを意味します。

各サンプル z/OSMF ワークフローは、1 つ以上のステップを実行します。 ワークフロー内のコメントは、各ステップで実行される機能について説明しています。 ステップには、データ入力を要求するだけのもの、JCL を実行依頼するもの、REXX 実行ファイルを呼び出して記述されている機能を実行するものなどがあります。

JCL または REXX 実行ファイルの正確な名前については、各ステップを参照してください。 ワークフローおよび関連する JCL または REXX 実行ファイルは、1 つ以上の 1043 ページの『ワークフロー変数宣言ファイル』 で宣言されている変数を参照します。

関連概念

1039 ページの『制限』
IBM MQ で z/OSMF を使用する場合の制限。

z/OS ワークフローの実行

サンプル z/OSMF ワークフローが参照するファイルと、ワークフローの実行方法についての説明です。

ワークフロー変数宣言ファイル

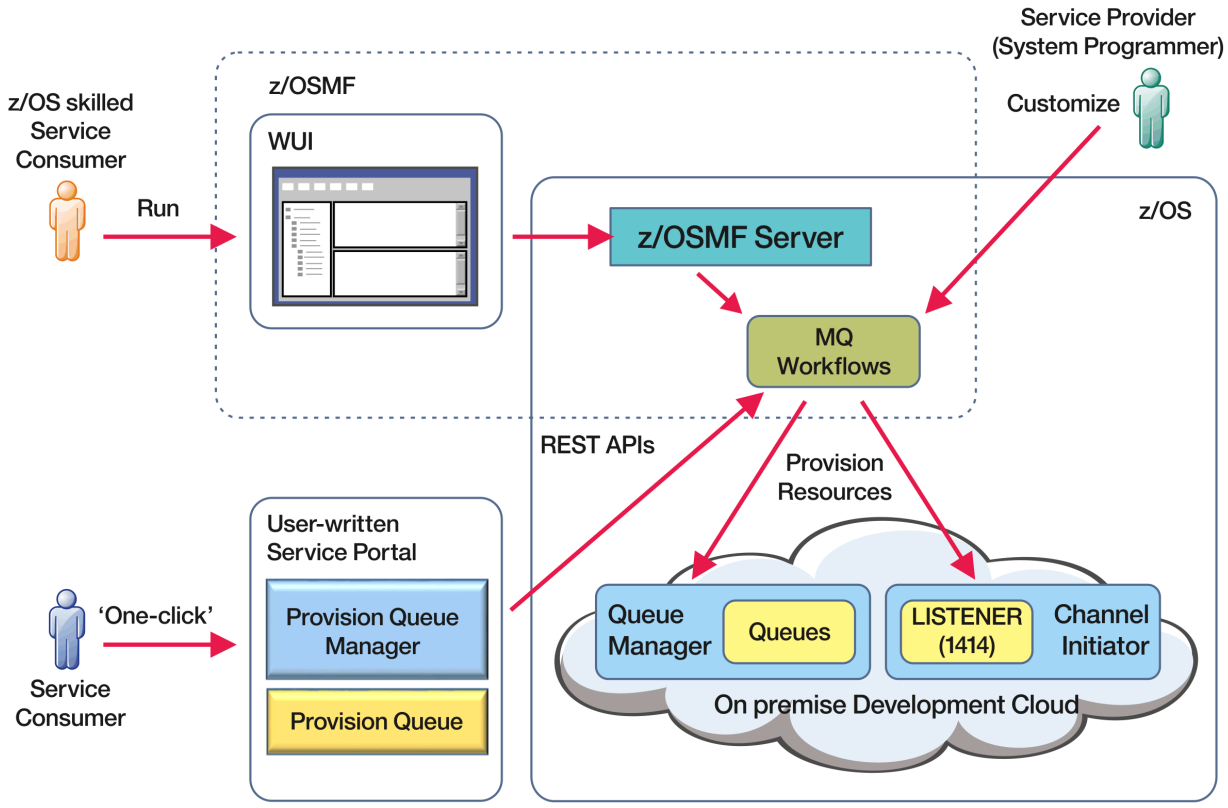
以下のファイルは、サンプル z/OSMF ワークフローおよび関連する JCL ファイルまたは REXX 実行ファイルが参照する変数を宣言します。

ワークフロー変数宣言ファイル名	説明
common_variables.xml	キュー・マネージャー (およびチャネル・イニシエーター) とキュー・ワークフローの両方に共通の変数。
qmgr_variables.xml	キュー・マネージャー (およびチャネル・イニシエーター) ワークフロー固有の変数。
queue_variables.xml	キュー・ワークフロー固有の変数。
tcpip_variables.xml	キュー・マネージャー (およびチャネル・イニシエーター) ワークフロー固有で、TCP/IP リソースの識別に使用される変数。

注 : 変数のデフォルトの可視性は *private* です。 z/OSMF REST API を使用して変数を照会できるように、選択された変数に *public* というマークが付けられています。 しかし、必要に応じて変数の可視性は変更することができます。

ワークフローの実行

図 *125. IBM MQ for z/OS* リソースの「ワンクリック」プロビジョニング



ワークフローを実行する前に、一部のプロパティーを以下のファイルに設定する必要があります。

ワークフロー変数プロパティー・ファイル名	説明
workflow_variables.properties	ワークフロー変数の初期プロパティー。 ファイル内のコメントに各プロパティーの目的が示されています。 • メタ括弧 (< >) 内のプロパティーは、ユーザー固有の値に設定する必要があります。 • 環境プロパティーを設定して、開発 (DEV) 環境、テスト (TEST) 環境、品質保証 (QA) 環境、実動 (PROD) 環境用にキュー・マネージャーをプロビジョンできます。 追加のプロパティー設定で、各環境用にプロビジョンされるキュー・マネージャーの特性を制御します。 例えば、環境タイプごとにアクティブ・ログ数またはページ・セット数を変更することができます。 • その他のプロパティーは IBM MQ デフォルト値に設定されますが、必要に応じてローカルの規則に合わせて変更できます。

通常、プロパティーを設定したら、ワークフローはそのまま実行できます。 ただし、必要であれば、ワークフローをカスタマイズして既存のステップを変更または除去したり、新規ステップを追加したりできます。

ワークフローは、以下のように実行できます。

- z/OSMF WUI から。

  WUI の「クラウド・プロビジョニング」->「ソフトウェア・サービス」から、自動モードまたは手動モードでワークフローを実行できます。 テストの際は手動モードが便利です。 また、両方のモードで、ワークフロー内の各ステップの進行状況をモニターできます。

詳しくは、 クラウド・プロビジョニング・サービス および ワークフローの作成を参照してください。

- z/OSMF REST Workflow Services の使用。

  REST Workflow Services を使用して REST API を介してワークフローを実行できます。 このモードは、ユーザー作成ポータルからのワンクリック操作を作成する際に役立ちます。

  詳細については、Cloud provisioning REST APIs を参照してください。また、z/OSMF ワークフロー・サービスも参照してください。
- z/OSMF に用意されているサンプルのマーケットプレイス・ポータルを使用する。

関連概念

1040 ページの『IBM MQ オブジェクトのプロビジョニングの自動化』

キュー・マネージャーおよびローカル・キューのプロビジョニングを自動化するためのサンプルが用意されています。

z/OS MQ Adv. VUE MFT エージェントからリモート z/OS キュー・マネージャーへの接続を有効にする

場合によっては、 z/OS 上の Managed File Transfer エージェントは、クライアント接続を使用して z/OS 上のリモート・キュー・マネージャーに接続できます。 これにより、 IBM MQ トポロジーが簡素化される可能性があります。

リモート z/OS キュー・マネージャーへのクライアント接続は、以下の場合にサポートされます。

- V 9.3.4 LTS MFT エージェントは、 IBM MQ 9.3.4 以降、または APAR PH56722 が適用された Long Term Support にあり、 IBM MQ Advanced for z/OS VUE または IBM MQ Advanced for z/OS のいずれかの製品 ID (PID) に関連付けられています。
- MFT エージェントは IBM MQ 9.3.0 にあり、 IBM MQ Advanced for z/OS VUE の PID に関連付けられています。

各種 PID については、 IBM MQ 製品 ID およびエクスポート情報 を参照してください。

MFT インストールに関連した PID を設定する方法については、「**fteSetProductId**」を参照してください。

エージェントの実行に使用される PID はエージェント開始時にログに表示されます。

他の PID の下で実行されている z/OS 上の MFT エージェントは、バインディング・モード接続を使用してローカル・キュー・マネージャーにのみ接続できます。

z/OS 上で実行されていないキュー・マネージャーにエージェントが接続しようとすると、 BFGQM1044E メッセージが発行され、エージェントの開始が終了します。

関連タスク

z/OS での MFT エージェントの開始

IBM MQ Internet Pass-Thru の構成

このセクションでは、IBM MQ Internet Pass-Thru (MQIPT) がサポートする各種機能と、それらの構成方法について説明します。

構成ファイル `mqipt.conf` を変更して、MQIPT を構成します。 MQIPT 構成ファイルの構造、および指定できるプロパティーは、IBM MQ Internet Pass-Thru 構成のリファレンスに説明されています。

注: 保管されているパスワードを表示したり、構成を変更したりすることができないように、`mqipt.conf` ファイルが置かれているディレクトリーに対してセキュア・ファイルの許可を設定する必要があります。 1085 ページの『MQIPT に保管されているパスワードの暗号化』の手順に従って、構成ファイルに指定するすべてのパスワードを保護してください。

構成ファイルに対する変更は、MQIPT の開始時またはリフレッシュ時に有効になります。 MQIPT のアクティブ・インスタンスをリフレッシュすると、MQIPT を再始動せずに構成変更が有効になります。 MQIPT

がリフレッシュされると、mqipt.conf 構成ファイルが再読み取りされ、MQIPT によって以下のアクションが実行されます:

- アクティブな経路のうち、非アクティブとしてマークされた経路、および構成ファイル内で指定されなくなった経路は、クローズされ、着信接続を受け入れなくなります。
- 構成ファイル内でアクティブとしてマークされ、現在実行されていないすべての経路が、開始されます。
- アクティブ経路の構成パラメーターに対する変更はすべて適用されます。 可能な場合、これらの変更は、アクティブな接続を中断することなく有効になります。 経路宛先への変更などの一部のパラメーター変更については、変更が適用される前にすべての接続が閉じて、経路が再始動されます。

MQIPT をリフレッシュするには、**mqiptAdmin** コマンドを使用します。 **mqiptAdmin** コマンドを使用した MQIPT の管理について詳しくは、 コマンド行を使用した MQIPT の管理を参照してください。

MQIPT での HTTP のサポート

MQIPT は HTTP トンネリングを サポートします。 MQIPT は、転送するデータ・パケットを HTTP 要求として エンコードするように構成できます。

IBM MQ チャネルは HTTP 要求を受け入れません。 そのため、HTTP リクエストを受け取り、IBM MQ プロトコルパケットに戻すために、二番目 MQIPT のパケットが必要です。 二番目の MQIPT は、受信パケットを宛先キュー・マネージャーに渡す前に、HTTP ヘッダーを除去して、着信パケットを標準の IBM MQ プロトコル・パケットに変換します。

MQIPT の 2 つのインスタンス間で HTTP が使用される場合、HTTP 要求 および応答が流れる TCP/IP 接続が持続し、メッセージ・チャネルの存続期間中はオープンのままになります。 MQIPT は要求/応答ペア間の TCP/IP 接続をクローズしません。

MQIPT の 2 つのインスタンスが HTTP を介して通信する場合、HTTP 要求が 長期間未処理のままとなる可能性があります。 例として、サーバー・サイドが伝送キューで新規メッセージの到着を 待機している場合の、要求側/サーバーのチャネルがあります。 IBM MQ チャネル・プロトコルは、待機側が定期的にハートビート・メッセージをパートナーに送信する必要がある「ハートビート」メカニズムを 提供しています。 デフォルトのチャネル・ハートビート期間は 5 分 です。 MQIPT は HTTP 応答としてこのハートビートを使用します。 一部の ファイアウォールでタイムアウトになる問題を回避するために、このチャネル・ハートビートを使用不可にしたり、 ハートビートに過剰に高い値を設定したりしないでください。

MQIPT は、HTTP プロキシーまたはサーバーで生成される、チャンク化された 形式の HTTP トラフィックを受け入れます。

MQIPT での HTTP の使用例については、HTTP トンネリングの構成を参照してください。

HTTP プロキシー

HTTP プロキシーは、MQIPT の 2 つのインスタンスの間に配置することができます。 HTTP プロキシーは以下の要件を満たす必要があります:

- プロキシーは HTTP1.1 プロトコルをサポートする必要があります。
- MQIPT によって設定される **Connection** ヘッダーまたは **Proxy-Connection** HTTP ヘッダーは、プロキシーによって受け入れられる必要があります。 これにより、MQIPT の 2 つのインスタンス間の接続が、メッセージ・チャネルの存続期間中にオープンされるようになります。
- 永続的な 1 対 1 マッピングは、プロキシーを介して維持する必要があります。 これにより、プロキシーから変換先 MQIPT への TCP/IP 接続が、複数のメッセージ・チャネルにデータを伝送するために使用されることはありません。

いくつかの HTTP プロキシー上で永続接続を管理する方法を構成するためのプロパティーを設定できます。 例えば、持続接続で作成できる要求の最大数を設定することができます。 次のプロパティーを設定する必要があります:

- 永続接続は使用可能にする必要があります。
- プロキシー経由での持続接続の 1 対 1 マッピングを維持するために、複数の HTTP セッションによるプロキシーから MQIPT への TCP/IP 接続の再利用は、使用不可にする必要があります。

- プロキシー要求のタイムアウトは、高い値に設定する必要があります。 例えば、12 時間を選択します。
- 持続接続に対して、実行できる要求の最大数は、高い値に設定する必要があります。 例えば、5000 とします。

MQIPT は HTTP POST 要求を使用して、MQIPT の 2 つのインスタンス間でデータを送信します。 MQIPT 構成で **HTTPProxy** プロパティーを使用して、プロキシーのホスト名が指定されている場合、MQIPT はプロキシーに接続し、HTTP CONNECT メソッドを使用して、プロキシーが変換先 MQIPT にトンネルを確立するように要求します。 これにより、HTTPS 接続はプロキシー内の TLS セッションを終了せずにプロキシーを移動することができます。

ロード・バランサーが MQIPT インスタンス間に配置されている場合は、*MQIPTSessionId* の HTTP Cookie の値を使用して、各セッションのすべての要求が同じ変換先に転送されるようにするために、それを構成する必要があります。

MQIPT の HTTPS

HTTPS は、クライアント接続を発行する MQIPT で **HTTPS** および **SSLClient** 経路プロパティーを有効にすることによって、 HTTP 接続で使用できます。

MQIPT は、ターゲット HTTP プロキシー/サーバーを認証するために使用される トラステッド CA 証明書にアクセスできる必要があります。 **SSLClientCAKeyring** プロパティーは、 トラステッド CA 証明書を含む鍵リング・ファイルを定義するために使用できます。

HTTPS の一般的なセットアップでは、ローカル HTTP プロキシーを使用してファイアウォールを トンネリングし、リモート HTTP サーバー (または別のプロキシー) に接続して、そこからリモート MQIPT に接続します。 接続のサーバー・サイドにある この MQIPT では、接続要求は通常の HTTP 接続として処理されるため、 特定の構成は必要ありません。

MQIPT では、**HTTPProxy** および **HTTPServer** の プロパティーを使用してローカルとリモートのプロキシーが区別されます。 **HTTPProxy** プロパティーが設定された MQIPT 経路はローカル HTTP プロキシーとして認識され、 **HTTPServer** プロパティーが設定された MQIPT 経路はリモート・サーバー (またはプロキシー) です。

HTTPS 接続は通常、 HTTP プロキシー/サーバー上のリスナー・ポート・アドレス 443 に対して行われますが、 **HTTPProxyPort** プロパティーおよび **HTTPServerPort** プロパティーを使用して、このデフォルトをオーバーライドすることができます。

MQIPT での SOCKS サポート

SOCKS プロキシーは、ファイアウォールを介した出口の制御点として使用されるネットワーク・サービスです。 ファイアウォール内で実行している SOCKS 対応アプリケーションは、SOCKS プロキシーを使用してリモート・アプリケーションに接続できます。

MQIPT は、**SocksServer** プロパティーを使用可能にすることによって、SOCKS プロキシーとして機能することができます。これにより、SOCKS 対応の IBM MQ アプリケーションは、MQIPT を介して、リモートの IBM MQ キュー・マネージャーに接続できるようになります。 この機能を使用すると、SOCKS ハンドシェーク・プロセスの間にターゲットの宛先および宛先ポート・アドレスが取得されるため、**Destination** および **DestinationPort** 経路プロパティーが指定変更されます。 これは、IBM MQ クラスター化をサポートするための重要な機能です。

MQIPT は、SOCKS 対応になっていないローカルの IBM MQ アプリケーションに代わる SOCKS クライアントとしても機能します。 これは、SOCKS プロキシー経由のみのアウトバウンド接続を許可するファイアウォールの使用時に有用です。 各 MQIPT 経路を異なる SOCKS プロキシーと通信するように構成できます。

SOCKS を使用する方法の例については、SOCKS プロキシーの構成を参照してください。

MQIPT でのクラスタリング

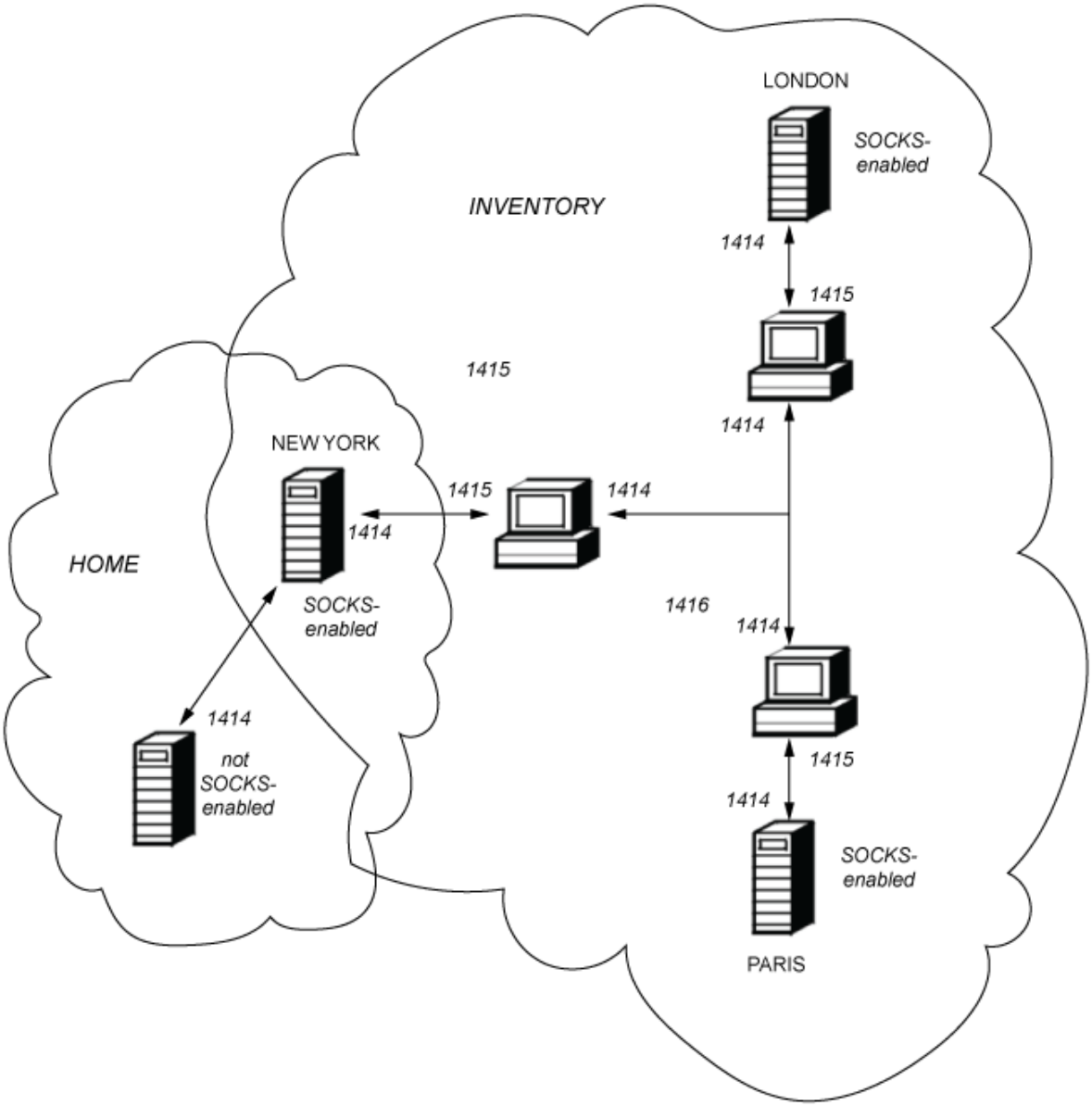
IBM MQ クラスターを MQIPT とともに使用することができます。これを行うには、インターネットに分散しているクラスター内の各キュー・マネージャー を SOCKS 対応にし、MQIPT を SOCKS プロキシーとして機能できるようにします。

以下の図では、NEWYORK および CHICAGO が HOME というクラスター内に存在し、両方がフルリポジトリーを保持しています。 NEWYORK、LONDON および PARIS は INVENTORY という別のクラスター内に存在します。 CHICAGO は SOCKS 対応にする必要はありません。 これは、MQIPT を必要としないクラスター内に存在するためです。

INVENTORY クラスター内の各キュー・マネージャーは、事実上 MQIPT の背後に隠されています。 キュー・マネージャーは SOCKS 対応になっているため、クラスター送信側チャネルが開始されると、SOCKS プロキシーとして 機能している MQIPT を使用して、要求がその宛先に送出されます。 通常、クラスター受信側チャネル上の CONNAME が、ローカル・キュー・マネージャーの特定に使用されますが、 MQIPT とともに使用する場合は、CONNAME はローカル MQIPT およびその着信リスナー・ポートを特定する必要があります。 以下の図では、すべての着信リスナー・ポート・アドレスが 1414 であり、発信リスナー・ポート・アドレスは 1415 です。

SOCKS 対応キュー・マネージャーを実行する方法は 2 つあります。 1 つ目は、キュー・マネージャーが実行しているコンピューター全体を SOCKS 対応にすることです。 2 つ目は、キュー・マネージャーのみを SOCKS 対応にすることです。 どちらの方法を使用する場合でも SOCKS クライアントを構成して、このクライアントが MQIPT を SOCKS プロキシーとして使用した場合のみリモート接続を確立するようにして、また、ユーザー認証を使用不可にする必要があります。 SOCKS サポートを実現する製品は市場に数多く存在します。 SOCKS V5 プロトコルをサポートする製品を選択する必要があります。

クラスター・ネットワークを構成する方法の例については、MQIPT クラスタリング・サポートの構成を参照してください。



MQIPT での SSL/TLS サポート

セキュア・ソケットを使用して、通信のプライバシー、通信の整合性、および認証を保証することができます。

通信のプライバシー
接続をプライベートにすることができます。 クライアントとサーバー間で交換されるデータを暗号化し、送信側と受信側のみがデータの意味を理解することができます。 これは、クレジット・カード番号などのプライベート情報を安全に転送できることを意味します。

通信の整合性
接続が信頼性の高いものになります。 メッセージ・トランスポートには、セキュア・ハッシュ関数に基づいたメッセージ整合性検査が含まれます。

認証
クライアントはサーバーを認証することができ、認証サーバーはクライアントを認証することができます。 これは、情報が確実に意図した相手の間でのみ交換されるようになることを意味します。 認証メカニズムは、デジタル証明書 (X.509v3 証明書) の交換に基づいています。

セキュア・ソケット・プロトコル

MQIPT では、Transport Layer Security (TLS) プロトコルと Secure Sockets Layer (SSL) プロトコルを使用して、セキュアなソケットを利用できます。 2 つのセキュア・ソケット・プロトコルは似ていますが相互運用はされません。 この資料では、特に違いについて説明する場合を除き、SSL および TLS という用語を交換可能な用語として使用しています。

MQIPT は Java runtime environment (JRE) で提供されている、SSL 3.0、TLS 1.0、TLS 1.1、および TLS 1.2 をサポートしています。 V 9.3.0 IBM MQ 9.3.0 以降、MQIPT は TLS 1.3 もサポートします。 リモート・チャネルの IBM MQ CipherSpec によって MQIPT が使用するプロトコルが決まります。

SSL 3.0、TLS 1.0、および TLS 1.1 は安全ではなく、 MQIPT ではデフォルトで無効になっています。 使用不可になっているこれらのプロトコルのいずれかを使用する必要がある場合は、1072 ページの『MQIPT での非推奨プロトコルおよび暗号スイートの使用可能化』の手順に従って再度使用可能にすることができます。

SSL/TLS プロトコルは通信相手の認証にさまざまなデジタル署名アルゴリズムを使用することができます。 SSL/TLS で使用される暗号操作、データの機密性のための暗号化、およびメッセージ整合性のためのセキュア・ハッシュは、クライアントとサーバー間での秘密鍵の共有に依存しています。 SSL/TLS が提供するさまざまな鍵交換メカニズムにより、秘密鍵の共有が可能になります。 SSL/TLS は暗号化およびハッシュのためのさまざまなアルゴリズムを使用することができます。

MQIPT での FIPS モードの有効化

JRE の SSL/TLS 暗号コンポーネントには、FIPS 140-2 標準に準拠するように認証された IBMJCEPlusFIPS セキュリティー・プロバイダーが含まれています。 MQIPT で FIPS 認定の暗号化のみを使用する場合は、 MQIPT の始動時に以下の Java システム・プロパティーを設定して、 IBMJSSE2 プロバイダーで FIPS モードを有効にします。

- `com.ibm.jsse2.usefipsprovider=true`
- V 9.3.0 `com.ibm.jsse2.usefipsProviderName=IBMJCEPlusFIPS`

MQIPT の始動時に、 **MQIPT_JVM_OPTIONS** 環境変数を使用して Java システム・プロパティーを設定できます。 例えば、 Linux では、 MQIPT を開始するコマンドを発行する前に、以下のコマンドを発行して環境変数を設定します。

```
export MQIPT_JVM_OPTIONS="-Dcom.ibm.jsse2.usefipsprovider=true
-Dcom.ibm.jsse2.usefipsProviderName=IBMJCEPlusFIPS"
```

FIPS モードを有効にする方法の詳細については、「FIPS モード IBMJSSE2 で」を参照してください。

SSL/TLS ブリッジング・モード

経路に SSLServer と SSLClient の両方が設定されている場合、MQIPT は 1 つの着信 SSL/TLS セキュア接続を受け入れ、別の MQIPT または宛先キュー・マネージャーへの 2 番目の SSL/TLS セキュア接続を確立します。 IBM MQ チャネル情報は暗号化解除され、これら 2 つの SSL/TLS 接続の間で再暗号化されます。 SSL/TLS ブリッジングは、 *SSL/TLS* 終端プロキシーとも呼ばれます。

IBM MQ は、 MQIPT を使用した SSL/TLS ブリッジングをサポートします。 IBM MQ を使用するその他の SSL/TLS 終端プロキシーでは、 IBM MQ によって送信されるサイズとは異なるサイズの SSL/TLS レコードがプロキシーによって結合または再構成されると、接続が切断されることが観測されています。 これは、キュー・マネージャーが着信 IBM MQ ネットワーク・データ用のメモリーを割り振って管理する方法と、IBM MQ ネットワーク・データを SSL/TLS レコードにパッケージする方法との間の相互作用が原因です。

MQIPT は、 IBM MQ ネットワーク・データを分割したり結合したりすることなく、それらのデータを SSL/TLS レコードにパッケージ化します。 他の SSL/TLS ブリッジが SSL/TLS レコードを正確に保存しない場合、 IBM MQ チャネルがエラー・メッセージを出して失敗する可能性があります。

```
AMQ9638: SSL communications error for channel
AMQ9208: Error on receive from host
```

SSL/TLS プロキシー・モード

MQIPT 経路は、SSL/TLS ブリッジングの代わりに SSL/TLS プロキシー・モードで構成できます。 このモードでは、経路は 2 つの IBM MQ エンドポイント間で SSL/TLS データを転送するだけで、SSL/TLS ハンドシェークには参加せず、デジタル証明書を必要としません。

SSL/TLS プロキシー・モードは、MQIPT を介して通信する IBM MQ チャネルが既に SSL/TLS 通信用に構成されており、ファイアウォールを介した接続のルーティングや、セキュリティー出口を介した許可される接続のセットの制限など、別の目的で MQIPT を使用する場合に使用できます。 SSL/TLS プロキシー・モードで実行されている場合、MQIPT は、新規接続から受信した初期 SSL/TLS パケットが有効であることをチェックしてから、パケットを宛先に転送します。

IBM MQ は、MQIPT またはその他の SSL/TLS プロキシーを使用して SSL/TLS プロキシー・モードをサポートします。

MQIPT に IBM MQ 複数の証明書サポート

IBM MQ 8.0 以降では、チャネル定義の **CERTLABL** 属性を使用して指定されたチャネルごとの証明書ラベルを使用して、同じキュー・マネージャー上で複数の証明書を使用することができます。 キュー・マネージャーへのインバウンド・チャネル (サーバー接続や受信側など) は、キュー・マネージャーからの正しい証明書を提示するために、TLS Server Name Indication (SNI) を使用したチャネル名の検出に依存します。 キュー・マネージャーでの複数の証明書の使用について、詳しくは、IBM MQ に複数の証明書機能を提供を参照してください。

チャネルが MQIPT を介して宛先キュー・マネージャーに接続し、 MQIPT 経路に **SSLServer** と **SSLClient** の両方が設定されている場合、エンドポイント間には 2 つの別個の TLS セッションが存在します。 「IBM MQ 9.3.0 より前のバージョンでは、SNI データはセッション中断を超えて流れません。」 これにより、MQIPT とキュー・マネージャーとの間の TLS 接続のために、宛先キュー・マネージャーでのチャネルごとの証明書が使用されなくなります。 宛先キュー・マネージャーでチャネルごとの証明書を使用するには、MQIPT IBM MQ 9.3.0 より前のバージョンでを通過する TLS 接続の場合、MQIPT 経路は SSL/TLS プロキシー・モードを使用する必要があります。このモードは、SNI 名を含め、すべての TLS 制御フローを無傷で転送します。

V 9.3.0 IBM MQ 9.3.0 から、SSLClientOutboundSNI ルートプロパティを使用して、TLS 接続の SNI を特定の値に設定するか、受信接続で受け取った SNI をルートに通過させるように、MQIPT 設定できます。 宛先キュー・マネージャーでのチャネルごとの証明書を使用できるようにするには、経路は SNI を IBM MQ チャネル名に設定するように構成するか、または経路にインバウンド接続で受信した SNI をパススルーするように構成する必要があります。 MQIPT が SNI を通過するように構成されている場合、MQIPT に接続するキュー・マネージャーまたはクライアントは SNI をチャネル名に設定する必要があります。

MQIPT によって終了または開始される TLS 接続に使用される証明書は、経路ごとに個別に構成できます。 例えば、**SSLServerSiteLabel** および **SSLClientSiteLabel** 経路プロパティーを使用します。

MQIPT でサポートされる CipherSuite

以下の表は、MQIPT でサポートされる CipherSuite と、デフォルトで使用可能になっている CipherSuite を示しています。

デフォルトでは、CipherSuite のサブセットのみが使用可能になっています。 安全でないと見なされるいくつかのアルゴリズムに基づく CipherSuite は、JRE によって使用不可にされています。 潜在的な危険を認識したうえで、これらの CipherSuite のいずれかを使用する必要がある場合は、1072 ページの『MQIPT での非推奨プロトコルおよび暗号スイートの使用可能化』の手順に従って、使用不可になっている CipherSuite のサポートを追加できます。

表 *87. MQIPT* とともに使用できる *CipherSuite*	
CipherSuite	**デフォルトでは使用可能**
TLS 1.3 の CipherSuite	

表 87. *MQIPT* とともに使用できる *CipherSuite* (続き)	
CipherSuite	**デフォルトでは使用可能**
V 9.3.0 V 9.3.0 TLS_AES_128_GCM_SHA256	はい
V 9.3.0 V 9.3.0 TLS_AES_256_GCM_SHA384	はい
V 9.3.0 V 9.3.0 TLS_CHACHA20_POLY1305_SHA256	はい
CipherSuites for SSL 3.0, TLS 1.0, TLS 1.1 and TLS 1.2	
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA	
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5	
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	
SSL_DH_anon_WITH_AES_128_CBC_SHA	
SSL_DH_anon_WITH_AES_128_CBC_SHA256	
SSL_DH_anon_WITH_AES_128_GCM_SHA256	
SSL_DH_anon_WITH_AES_256_CBC_SHA	
SSL_DH_anon_WITH_AES_256_CBC_SHA256	
SSL_DH_anon_WITH_AES_256_GCM_SHA384	
SSL_DH_anon_WITH_DES_CBC_SHA	
SSL_DH_anon_WITH_RC4_128_MD5	
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	
SSL_DHE_DSS_WITH_AES_128_CBC_SHA	はい
SSL_DHE_DSS_WITH_AES_128_CBC_SHA256	はい
SSL_DHE_DSS_WITH_AES_128_GCM_SHA256	はい
SSL_DHE_DSS_WITH_AES_256_CBC_SHA	はい
SSL_DHE_DSS_WITH_AES_256_CBC_SHA256	はい
SSL_DHE_DSS_WITH_AES_256_GCM_SHA384	はい
SSL_DHE_DSS_WITH_DES_CBC_SHA	
SSL_DHE_DSS_WITH_RC4_128_SHA	
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_DHE_RSA_WITH_AES_128_CBC_SHA	はい
SSL_DHE_RSA_WITH_AES_128_CBC_SHA256	はい
SSL_DHE_RSA_WITH_AES_128_GCM_SHA256	はい
SSL_DHE_RSA_WITH_AES_256_CBC_SHA	はい
SSL_DHE_RSA_WITH_AES_256_CBC_SHA256	はい
SSL_DHE_RSA_WITH_AES_256_GCM_SHA384	はい

表 87. *MQIPT* とともに使用できる *CipherSuite (*続き*)*	
CipherSuite	**デフォルトでは使用可能**
SSL_DHE_RSA_WITH_DES_CBC_SHA	
SSL_ECDH_anon_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_anon_WITH_AES_128_CBC_SHA	
SSL_ECDH_anon_WITH_AES_256_CBC_SHA	
SSL_ECDH_anon_WITH_NULL_SHA	
SSL_ECDH_anon_WITH_RC4_128_SHA	
SSL_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA （注） 1055 ページの『1』	
SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 （注：） 1055 ページの『1』	
SSL_ECDH_ECDSA_WITH_AES_128_GCM_SHA256 （注：） 1055 ページの『1』	
SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA （注：） 1055 ページの『1』	
SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 （注） 1055 ページの『1』	
SSL_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 （注：） 1055 ページの『1』	
SSL_ECDH_ECDSA_WITH_NULL_SHA	
SSL_ECDH_ECDSA_WITH_RC4_128_SHA	
SSL_ECDH_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_RSA_WITH_AES_128_CBC_SHA （注：） 1055 ページの『1』	
SSL_ECDH_RSA_WITH_AES_128_CBC_SHA256 （注） 1055 ページの『1』	
SSL_ECDH_RSA_WITH_AES_128_GCM_SHA256 （注） 1055 ページの『1』	
SSL_ECDH_RSA_WITH_AES_256_CBC_SHA （注：） 1055 ページの『1』	
SSL_ECDH_RSA_WITH_AES_256_CBC_SHA384 （注：） 1055 ページの『1』	
SSL_ECDH_RSA_WITH_AES_256_GCM_SHA384 （注：） 1055 ページの『1』	
SSL_ECDH_RSA_WITH_NULL_SHA	
SSL_ECDH_RSA_WITH_RC4_128_SHA	
SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	はい
SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	はい
SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	はい
SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	はい
SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	はい
SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	はい
SSL_ECDHE_ECDSA_WITH_NULL_SHA	
SSL_ECDHE_ECDSA_WITH_RC4_128_SHA	
SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	

表 87. *MQIPT* とともに使用できる *CipherSuite (*続き*)*

CipherSuite	**デフォルトでは使用可能**
SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA	はい
SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256	はい
SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256	はい
SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA	はい
SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384	はい
SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384	はい
SSL_ECDHE_RSA_WITH_NULL_SHA	
SSL_ECDHE_RSA_WITH_RC4_128_SHA	
SSL_KRB5_EXPORT_WITH_DES_CBC_40_MD5	
SSL_KRB5_EXPORT_WITH_DES_CBC_40_SHA	
SSL_KRB5_EXPORT_WITH_RC4_40_MD5	
SSL_KRB5_EXPORT_WITH_RC4_40_SHA	
SSL_KRB5_WITH_3DES_EDE_CBC_MD5	
SSL_KRB5_WITH_3DES_EDE_CBC_SHA	
SSL_KRB5_WITH_DES_CBC_MD5	
SSL_KRB5_WITH_DES_CBC_SHA	
SSL_KRB5_WITH_RC4_128_MD5	
SSL_KRB5_WITH_RC4_128_SHA	
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	
SSL_RSA_EXPORT_WITH_RC4_40_MD5	
SSL_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_RSA_WITH_AES_128_CBC_SHA	はい
SSL_RSA_WITH_AES_128_CBC_SHA256	はい
SSL_RSA_WITH_AES_128_GCM_SHA256	はい
SSL_RSA_WITH_AES_256_CBC_SHA	はい
SSL_RSA_WITH_AES_256_CBC_SHA256	はい
SSL_RSA_WITH_AES_256_GCM_SHA384	はい
SSL_RSA_WITH_DES_CBC_SHA	
SSL_RSA_WITH_NULL_MD5	
SSL_RSA_WITH_NULL_SHA	
SSL_RSA_WITH_NULL_SHA256	
SSL_RSA_WITH_RC4_128_MD5	はい
SSL_RSA_WITH_RC4_128_SHA	

表 87. *MQIPT* とともに使用できる *CipherSuite* (続き)	
CipherSuite	**デフォルトでは使用可能**
V 9.3.0 TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	
V 9.3.0 TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	
V 9.3.0 TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	

注 :

1. V 9.3.0.30 CipherSuite は、 MQIPT ではデフォルトでは有効になっていません。

CipherSpec および MQIPTCipherSuites

以下の表は、IBM MQ でサポートされる CipherSpec と MQIPT でサポートされる CipherSuite の関係を示しています。

また、この表は、各 CipherSpec が使用することを IBM MQ が想定しているプロトコル・バージョンも示しています。

IBM MQ CipherSpec は、使用する暗号化アルゴリズムとセキュア・ソケット・プロトコル・バージョンの両方を一意的に決定します。 一部の IBM MQ CipherSpec はプロトコル・バージョンが異なるだけであるため、CipherSuite のみを構成するのでは十分ではありません。 SSL/TLS ハンドシェークによって、両側でサポートされる最も高いセキュア・ソケット・プロトコル・バージョンがネゴシエーションされて、相互に使用可能になっている暗号のセットから CipherSuite が選択されます。

例えば、`SSLClientCipherSuites=SSL_RSA_WITH_3DES_EDE_CBC_SHA` を使用する SSLClient 経路は、TLS_RSA_WITH_3DES_EDE_CBC_SHA (TLS 1.0) または TRIPLE_DES_SHA_US (SSL 3.0) のいずれかをリモート・キュー・マネージャーとネゴシエーションできます。 実際には、TLS 1.2 を介したこの CipherSuite をネゴシエーションすることができますが、IBM MQ は TLS 1.2 を介したこの CipherSuite をサポートしていません。 このような理由から、SSLClient 経路はキュー・マネージャーで AMQ9616 エラーまたは AMQ9631 エラーの原因となる可能性が特に高くなります。

SSLClient 経路でこのようなエラーを回避するには、**SSLClientProtocols** 経路プロパティーを目的の CipherSpec の適切な値に設定します。 場合によっては、**SSLServerProtocols** 経路プロパティーを使用してサーバー・サイド・プロトコル・セットを制限する必要が生じることもあります。 表に示されているプロトコル・バージョンを使用して、これらの経路プロパティーの正しい設定を決定します。

この問題は特に、SSLClient 経路の以下の CipherSuite および CipherSpec に影響を与えます。

- SSL_RSA_WITH_3DES_EDE_CBC_SHA。これは以下に対応しています。
  - SSL 3.0: MQ CipherSpec TRIPLE_DES_SHA_US
  - TLS 1.0: MQ CipherSpec TLS_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_DES_CBC_SHA。これは以下に対応しています。
  - SSL 3.0: MQ CipherSpec DES_SHA_EXPORT
  - TLS 1.0: MQ CipherSpec TLS_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_RC4_128_SHA。これは以下に対応しています。
  - SSL 3.0: MQ CipherSpec RC4_SHA_US
  - TLS 1.2: MQ CipherSpec TLS_RSA_WITH_RC4_128_SHA256

単一の MQIPT SSLClient 経路を使用して異なる CipherSpec を使用する複数の IBM MQ チャネルをトンネリングするには、すべてのチャネルに、相互に同じセキュア・ソケット・プロトコル・バージョンを使用する CipherSpec があり、**SSLClientProtocols** をこの単一のプロトコル・バージョンを使用するように設定してます。

IBM MQ CipherSpec の詳細については、CipherSpec の有効化を参照してください。

IBM MQ CipherSpec	MQIPT CipherSuite	プロトコル・バージョン
DES_SHA_EXPORT	SSL_RSA_WITH_DES_CBC_SHA	SSLv3
DES_SHA_EXPORT1024	N/A	N/A
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	TLSv1.2
ECDHE_ECDSA_AES_128_CBC_SHA256	SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	TLSv1.2
ECDHE_ECDSA_AES_128_GCM_SHA256	SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	TLSv1.2
ECDHE_ECDSA_AES_256_CBC_SHA384	SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	TLSv1.2
ECDHE_ECDSA_AES_256_GCM_SHA384	SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLSv1.2
ECDHE_ECDSA_NULL_SHA256	SSL_ECDHE_ECDSA_WITH_NULL_SHA	TLSv1.2
ECDHE_ECDSA_RC4_128_SHA256	SSL_ECDHE_ECDSA_WITH_RC4_128_SHA	TLSv1.2
ECDHE_RSA_3DES_EDE_CBC_SHA256	SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1.2
ECDHE_RSA_AES_128_CBC_SHA256	SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLSv1.2
ECDHE_RSA_AES_128_GCM_SHA256	SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256	TLSv1.2
ECDHE_RSA_AES_256_CBC_SHA384	SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384	TLSv1.2
ECDHE_RSA_AES_256_GCM_SHA384	SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384	TLSv1.2
ECDHE_RSA_NULL_SHA256	SSL_ECDHE_RSA_WITH_NULL_SHA	TLSv1.2
ECDHE_RSA_RC4_128_SHA256	SSL_ECDHE_RSA_WITH_RC4_128_SHA	TLSv1.2
NULL_MD5	SSL_RSA_WITH_NULL_MD5	SSLv3
NULL_SHA	SSL_RSA_WITH_NULL_SHA	SSLv3
RC2_MD5_EXPORT	N/A	N/A
RC4_56_SHA_EXPORT1024	N/A	N/A
RC4_MD5_EXPORT	SSL_RSA_EXPORT_WITH_RC4_40_MD5	SSLv3
RC4_MD5_US	SSL_RSA_WITH_RC4_128_MD5	SSLv3
RC4_SHA_US	SSL_RSA_WITH_RC4_128_SHA	SSLv3
V 9.3.0 V 9.3.0 TLS_AES_128_GCM_SHA256	TLS_AES_128_GCM_SHA256	TLSv1.3
V 9.3.0 V 9.3.0 TLS_AES_256_GCM_SHA384	TLS_AES_256_GCM_SHA384	TLSv1.3
V 9.3.0 V 9.3.0 TLS_CHACHA20_POLY1305_SHA256	TLS_CHACHA20_POLY1305_SHA256	TLSv1.3
TLS_RSA_WITH_3DES_EDE_CBC_SHA	SSL_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA	SSL_RSA_WITH_AES_128_CBC_SHA	TLSv1

IBM MQ CipherSpec	MQIPT CipherSuite	プロトコル・バージョン
TLS_RSA_WITH_AES_128_CBC_SHA256	SSL_RSA_WITH_AES_128_CBC_SHA256	TLSv1.2
TLS_RSA_WITH_AES_128_GCM_SHA256	SSL_RSA_WITH_AES_128_GCM_SHA256	TLSv1.2
TLS_RSA_WITH_AES_256_CBC_SHA	SSL_RSA_WITH_AES_256_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_256_CBC_SHA256	SSL_RSA_WITH_AES_256_CBC_SHA256	TLSv1.2
TLS_RSA_WITH_AES_256_GCM_SHA384	SSL_RSA_WITH_AES_256_GCM_SHA384	TLSv1.2
TLS_RSA_WITH_DES_CBC_SHA	SSL_RSA_WITH_DES_CBC_SHA	TLSv1
TLS_RSA_WITH_NULL_NULL	N/A	N/A
TLS_RSA_WITH_NULL_SHA256	SSL_RSA_WITH_NULL_SHA256	TLSv1.2
TLS_RSA_WITH_RC4_128_SHA256	SSL_RSA_WITH_RC4_128_SHA	TLSv1.2
TRIPLE_DES_SHA_US	SSL_RSA_WITH_3DES_EDE_CBC_SHA	SSLv3

MQIPT での SSL/TLS ハンドシェーク

SSL/TLS のハンドシェーク・プロセスは、CipherSuites の認証とネゴシエーションが実行されるときに、SSL/TLS のクライアントとサーバー間における初期接続要求中に行われます。

匿名 CipherSuites を除き、サポートされるすべての SSL/TLS CipherSuites では(1049 ページの『MQIPT での SSL/TLS サポート』を参照)、サーバー認証が必要であり、クライアント認証が許可されます。クライアント認証を要求するようにサーバーを構成できます。 匿名 CipherSuites では、リモート・ピアの身元が保証されないため、これを使用することは推奨されません。 自分の知らないところで、中間者攻撃により匿名 SSL/TLS 接続がインターセプトされる可能性があります。 匿名 CipherSuites は、データがインターセプトされるリスクを受け入れることが可能な場合にのみ、信頼できる内部ネットワークでのみ使用するようにしてください。

SSL/TLS の通信ピア認証は、公開鍵暗号および X.509v3 デジタル証明書に基づいています。 SSL/TLS プロトコルで認証を行う必要があるサイトでは、秘密鍵とデジタル証明書 (この証明書には対応する公開鍵と、サイトの身元に関する情報が含まれます)、および証明書の有効期間が必要となります。 証明書は認証局によって署名され、このような証明書は署名者証明書と呼ばれます。 証明書に 1 つ以上の署名者証明書が続いて、証明書チェーンが構成されます。 証明書チェーンの特徴として、最初の証明書 (サイト証明書) から開始して、チェーン内の各証明書の署名が、次の署名者証明書に含まれる公開鍵を使用して検証可能である点を挙げることができます。

サーバー認証が必要なセキュア接続が確立されると、サーバーは、その身元を証明するために証明書チェーンをクライアントに送信します。 SSL/TLS クライアントは、サーバーを認証可能な場合にのみ、そのサーバーとの接続の確立を続行し、例えばサーバーのサイト証明書の署名を検証します。 署名を検証するために、SSL/TLS クライアントは、サーバーのサイト自体を信頼するか、またはサーバーが提供する証明書チェーン内の少なくとも 1 つの署名者を信頼する必要があります。 信頼できるサイトおよび署名者の証明書は、この検証を実行するためにクライアント・サイドで維持する必要があります。

SSL/TLS クライアントはサーバーの証明書チェーンを検査します。その際、サイト証明書から開始します。クライアントは、以下の点についてサイト証明書の署名が有効であるかどうかを検討します。

- サイト証明書が信頼できるサイトのリポジトリーまたは署名者証明書に含まれている
- チェーン内の署名者証明書が、信頼できる署名者証明書のリポジトリーに基づいて検証可能である

2 番目の場合、SSL/TLS クライアントは、信頼できる署名者証明書からサーバーのサイト証明書まで、証明書チェーンが正しく署名されているかどうかをチェックします。 また、このプロセスに関連する各証明書についても、形式や有効期間が検査されます。 これらのチェックのいずれかが失敗すると、サーバーへの接続は拒否されます。 クライアントは、サーバーの証明書を検証した後、SSL/TLS プロトコルの次のステップで、その証明書に埋め込まれている公開鍵を使用します。 SSL/TLS 接続は、サーバーが対応する秘密鍵を実際に保有している場合にのみ確立できます。

クライアント認証も同じ手順に従います。SSL/TLS サーバーでクライアント認証が必要な場合、クライアントはその身元を証明するためにサーバーに証明書チェーンを送信します。 サーバーは、信頼できるサイトおよび署名者証明書のリポジトリーに基づいてチェーンを検証します。 サーバーは、クライアントの証明書を検証した後、SSL/TLS プロトコルの次のステップで、その証明書に埋め込まれている公開鍵を使用します。 SSL/TLS 接続は、クライアントが対応する秘密鍵を実際に保有している場合にのみ確立できます。

最新バージョンの TLS プロトコルは、高度なセキュリティー通信を実現します (SSL および古い TLS プロトコルは安全でないと見なされています)。 ただし、プロトコルはアプリケーションで指定される情報に基づいて動作します。 その情報ベースも安全に管理されている場合にのみ、安全な通信という目標が達成されます。 例えば、信頼できるサイトおよび署名者証明書のリポジトリーのセキュリティーが侵害されると、安全でない通信パートナーとセキュア接続を確立することになる場合があります。

SSL/TLS の MQIPT 実装

SSL 3.0、TLS 1.0、TLS 1.1、TLS 1.2 および TLS 1.3 は、 X509.V3 証明書を含む鍵リング・ファイル (ファイル・タイプは `.p12` または `.pfx`) に保管される Public Key Cryptography Standards (PKCS) #12 トークンを使用して実装されます。 MQIPT では、PKCS#11 Cryptographic Token Interface 標準に対応した暗号ハードウェア鍵ストアを使用することもできます。 MQIPT では IBM Java Secure Socket Extension (JSSE) パッケージを使用します。

MQIPT は、接続がどちらの側から開始されるかによって、SSL/TLS クライアントまたは SSL/TLS サーバーとして機能できます。 クライアントは接続を開始して、サーバーは接続要求を受け入れます。 MQIPT 経路はクライアントとサーバーの両方として機能できます。 この場合、通常は、SSL/TLS プロキシー・モード機能を使用することでパフォーマンスが向上します。

MQIPT が SSL/TLS プロキシー・モード用に構成されている場合、2 つのエンドポイント間で SSL/TLS データのみを転送し、SSL/TLS ハンドシェークに参加しないため、デジタル証明書は必要ありません。

IBM MQ 9.3.0 より前のバージョンでは、MQIPT は、インバウンド TLS 接続で受信した TLS Server Name Indication (SNI) データをアウトバウンド TLS 接続に渡しません。 つまり、**CERTLABL** チャネル属性を使用して指定されたチャネルごとの証明書を、MQIPT と宛先キュー・マネージャーの間の TLS 接続に使用できません。 宛先キュー・マネージャーでチャネルごとの証明書を使用するには、MQIPT IBM MQ 9.3.0 より前のバージョンでを通過する TLS 接続の場合、MQIPT 経路は SSL/TLS プロキシー・モードを使用する必要があります。このモードは、SNI 名を含め、すべての TLS 制御フローを無傷で転送します。 V 9.3.0 IBM MQ 9.3.0 以降、MQIPT は、TLS 接続の SNI を特定の値に設定するか、インバウンド接続で受信した SNI を経路にパススルーするように構成できます。 MQIPT でキュー・マネージャーで複数の証明書を使用する方法について詳しくは、1051 ページの『MQIPT に IBM MQ 複数の証明書サポート』を参照してください。

各 MQIPT 経路は、独自の SSL/TLS プロパティー・セットで個別に構成できます。 詳しくは、MQIPT 経路のプロパティーを参照してください。

MQIPT での鍵リング・パスワードの暗号化

mqiptPW コマンドを使用して、鍵リング・ファイルを開くため、または MQIPT が使用する暗号ハードウェアにアクセスするために使用するパスワードを暗号化します。 暗号化されたパスワードは、**SSLClientKeyRingPW**、**SSLClientCAKeyRingPW**、**SSLServerKeyRingPW**、**SSLServerCAKeyRingPW**、および **SSLCommandPortKeyRingPW** のいずれかのプロパティーで使用されます。 このトピックでは、MQIPT で使用する鍵リング・パスワードの正しい保管方法について説明します。

mqiptkeyman (iKeyman) stash ファイル機能は、MQIPT ではサポートされません。 stash ファイルを使用する代わりに、**mqiptPW** コマンドを使用して、暗号化されたパスワードを保管する必要があります。

IBM MQ 9.1.5 より前のバージョンでは、MQIPT で使用する鍵リング・パスワードはいずれかの **SSL*KeyRingPW** プロパティーによって、参照されるファイルに保管されています。

IBM MQ 9.1.5 からは、MQIPT で使用する鍵リング・パスワードを **mqiptPW** コマンドを使用して暗号化し、**SSL*KeyRingPW** プロパティーの値を、暗号化されたパスワードに設定します。 MQIPT は、IBM MQ 9.1.5 よりも前に作成された構成との互換性のために、プロパティー値における暗号化されたパスワードとファイル名を区別できます。

Deprecated IBM MQ 9.1.5 より前の MQIPT バージョンで使用可能な鍵ストア・パスワードを暗号化する方式は推奨されませんが、引き続き使用することができます。 鍵リング・パスワードの保護を向上させるには、最新の保護方式を使用して、以前に暗号化された鍵リング・パスワードを再暗号化します。

MQIPT で使用する鍵リング・パスワードを暗号化するには、1085 ページの『MQIPT に保管されているパスワードの暗号化』の手順に従ってください。

MQIPT インストール・ディレクトリーの samples/ssl サブディレクトリーに提供されているいずれかのサンプル鍵リング・ファイルを開くには、パスワード mqiptSample を使用する必要があります。

MQIPT での鍵リング・ファイルからの証明書の選択

複数の個人証明書を同じ鍵リング・ファイルまたは暗号ハードウェア・トークンに保管できます。 **SSLClientSite*** プロパティーをクライアント・サイドで使用して、認証のためにサーバーに送信する証明書を選択するか、**SSLServerSite*** プロパティーをサーバー・サイドで使用して、認証のためにクライアントに送信する証明書を選択することができます。

これらのプロパティーを使用すれば、証明書をその識別名 (DN) に基づいて選択することができます。 また、**SSLServerSiteLabel** と **SSLClientSiteLabel** プロパティーで証明書ラベルを使用して、証明書を選択することもできます。

TLS コマンド・ポートによって使用されるサーバー証明書を選択するには、**SSLCommandPortSiteLabel** プロパティーを使用して、証明書のラベル名を指定します。

MQIPT での信頼設定

鍵リングには個人証明書が含まれています。この個人証明書には署名者証明書または署名者証明書チェーンが含まれています。

MQIPT は、以下の 2 種類の鍵リングを使用します。

認証局 (CA) の鍵リング

この鍵リングにはトラステッド CA 証明書が含まれています。この証明書は、リモート・ピアに属している証明書を検証するために使用されます。 これらの CA 証明書は、リモート・ピアが信頼できるかどうかを判断するのに役立ちます。 MQIPT は、CA 証明書を保管するために、PKCS #12 形式の鍵リング・ファイルと、PKCS #11 インターフェースに対応した暗号ハードウェア鍵ストアの両方をサポートします。 MQIPT CA 鍵リング・ファイルは **SSLClientCAKeyRing** と **SSLServerCAKeyRing** 経路プロパティーで識別されます。 CA 証明書にアクセスするために暗号ハードウェアを使用できるようにするには、 **SSLClientCAKeyRingUseCryptoHardware** プロパティーと **SSLServerCAKeyRingUseCryptoHardware** プロパティーを設定します。

SSL/TLS クライアント・サイドの CA 鍵リングには、サーバーから送信された証明書の認証に使用するトラステッド CA 証明書のリストが含まれていなければなりません。 クライアント認証を行うように SSL サーバー経路を構成する場合は、SSL/TLS サーバー・サイドの CA 鍵リングに、クライアントから送信された証明書の認証に使用するトラステッド CA 証明書のリストが含まれていなければなりません。

個人証明書の鍵リング

この鍵リングには、MQIPT がリモート・ピアに対して自身を識別するために使用する個人証明書が含まれています。 自己署名証明書を生成するか、CA 署名済み証明書を要求する場合は、個人証明書の鍵リングを使用して行う必要があります。 MQIPT は、個人証明書を保管するために、PKCS #12 形式の鍵リング・ファイルと、PKCS #11 インターフェースに対応した暗号ハードウェア鍵ストアの両方をサポートします。 MQIPT では、個人証明書の鍵リング・ファイルは **SSLClientKeyRing** と **SSLServerKeyRing** 経路プロパティーで識別されます。 個人証明書にアクセスするために暗号ハードウェアを使用できるようにするには、 **SSLClientKeyRingUseCryptoHardware** プロパティーと **SSLServerKeyRingUseCryptoHardware** プロパティーを設定します。

SSL/TLS サーバー・サイドの鍵リングには、MQIPT サーバーの個人証明書が含まれていなければなりません。 SSL クライアント経路でクライアント認証が必要な場合は、SSL/TLS クライアント・サイドの鍵リングにクライアントの個人証明書が含まれていなければなりません。

クライアント認証が必要な場合は、サーバー・サイドで **SSLServerAskClientAuth** プロパティーを使用可能にする必要があります。 クライアント・サイドの鍵リングに、クライアントの個人証明書が含まれていなければなりません。 **SSLServerCAKeyRing** プロパティーによって識別される、サーバー・サイドの MQIPT 鍵リングには、クライアントの認証に使用されるトラステッド CA 証明書のリストが含まれている必要があります。

経路に CA 鍵リングを構成しない場合、MQIPT は、代わりに個人証明書の鍵リング (構成されている場合) で CA 証明書を検索します。 例えば、**SSLServerCAKeyRing** に値が設定されていない場合、MQIPT は、**SSLServerKeyRing** に指定されている鍵リングで CA 証明書を検索します。

トラステッド CA によって署名された証明書を使用する代わりに、自己署名証明書を使用できます。 自己署名証明書の例は、`samples/ssl` サブディレクトリで提供される `MQIPTsslSample.pfx` サンプルキーリングファイルで見ることができます。 サンプル PKCS #12 鍵リング・ファイルを開くには、パスワード `mqiptSample` を使用する必要があります。

自己署名証明書は、CA に料金を支払って証明書を取得せずに SSL/TLS 接続を確認する必要があるテスト・シナリオで役立ちます。 ただし、実稼働環境では、自己署名証明書は使用しないでください。 CA 署名証明書を作成するには、鍵リング・ファイルの作成を参照してください。

MQIPT に付属の **mqiptkeyman** というユーティリティーを使用して、デジタル証明書と鍵ストアを管理できます。 インストール手順と詳細情報については、1063 ページの『MQIPT の mqiptKeyman および mqiptKeycmd』を参照してください。

オペレーティング・システムのセキュリティー機能を使用して、すべての鍵リング・ファイルとパスワード・ファイルを保護し、無許可アクセスを防ぐ必要があります。

MQIPT での SSL/TLS のテスト

SSL/TLS 接続のテストに役立つ例を示します。

さまざまなシナリオについては、IBM MQ Internet Pass-Thru の使用を開始するを参照してください。 特に、以下のタスクを参照してください。

- SSL/TLS サーバーの認証
- SSL/TLS クライアントの認証
- SSL/TLS プロキシー・モードでの MQIPT の実行
- セキュリティー・マネージャーを使用した SSL/TLS プロキシー・モードでの MQIPT の実行

SSL/TLS 構成が正しく機能することをテストする場合は、自己署名証明書を使用できます。 自己署名証明書は、認証局 (CA) に料金を支払って証明書を取得せずに SSL/TLS 接続を確認できるテスト・シナリオで役立ちます。 詳細はテスト認証の作成を参照してください。

自己署名証明書の例は、`samples/ssl` サブディレクトリで提供される `MQIPTsslSample.pfx` サンプルキーリングファイルで見ることができます。 サンプルの PKCS #12 鍵リング・ファイルを開くには、パスワード `mqiptSample` を使用する必要があります。 サンプル証明書は、テスト時に参考用に提供されます。 ただし、サンプル証明書の秘密鍵はすべての MQIPT ユーザーに認識されます。 これは、秘密鍵が安全ではないため、テスト環境でのみ使用する必要があることを意味します。

自己署名証明書がサンプル証明書かどうかに関係なく、実稼働環境で使用することはできません。 代わりに、トラステッド CA から CA 署名済み証明書を取得してください。 CA 署名証明書を作成するには、鍵リング・ファイルの作成を参照してください。

証明書を作成または要求する場合は、セキュリティー・ニーズに合った鍵タイプ、鍵サイズおよびデジタル署名アルゴリズムを考慮する必要があります。 詳細については、1065 ページの『MQIPT に対するデジタル証明書の考慮事項』を参照してください。

多くのサード・パーティー・サプライヤーから提供される証明書や証明書管理テクノロジーを利用できます。

MQIPT の SSL/TLS エラー・メッセージ

ハンドシェーク障害は、JSSE 例外の形式で MQIPT 接続ログに記録されます。

詳細については、1088 ページの『MQIPT の接続ログ』を参照してください。 以下の表では、さまざまな例外、考えられる原因および障害を解決するための対応するアクションについて説明します。

証明書の例外は、通常、接続のリモート・エンドの証明書に関連します。

エラーが IBM MQ クライアントまたはキュー・マネージャーの証明書に関連している場合、キー・リング・ファイル という用語には、リモート・パートナーの IBM MQ キー・リポジトリーが含まれます。

MQIPT では、CA 証明書は CA 鍵リング・ファイルに保管されます。このファイルは **SSLClientCAKeyRing** および **SSLServerCAKeyRing** 経路プロパティーで識別されます。 CA 鍵リングの経路プロパティーが設定されていない場合は、代わりに、(**SSLClientKeyRing** または **SSLServerKeyRing** のいずれかのプロパティーで指定されている) 対応する個人鍵リング・ファイルで CA 証明書が検索されます。

例外	原因	アクション
CertificateException	証明書は CA 鍵リングに存在しない CA によって署名されているため、信頼されません。	CA 鍵リング・ファイルに必要な CA 証明書がすべて存在することを確認します。 MQIPT に用意されている IBM 鍵管理ツールを使用して、欠落している CA 証明書を追加します。その際、信頼できるソースから各 CA 証明書のコピーを取得するように注意してください。
CertificateExpiredException	1. 証明書の有効期限が切れています。 **notAfter** 日を過ぎています。 2. システム・クロックが正しく設定されていません。	1. 新しい証明書を取得して、それを鍵リング・ファイルに挿入します。 証明書が認証局に属している場合は、新しい証明書を CA 鍵リング・ファイルに配置します。 2. UTC システム・クロックが正しい時刻に設定されていることを確認します。
CertificateNotYetValidException	1. 証明書の使用が早すぎます。 **notBefore** 日にはまだ達していません。 2. システム・クロックが正しく設定されていません。	1. 証明書が正しく生成され、署名されていることを確認します。 組織で独自の CA を運営している場合は、CA の UTC システム・クロックが正しくない可能性があります。 2. UTC システム・クロックが正しい時刻に設定されていることを確認します。
CertificateParsingException	1. 証明書に無効な DER データが含まれています。 2. 証明書でサポートされていない DER 機能が使用されています。	証明書が正しく生成されており、MQIPT に付属の IBM 鍵管理ツールで表示できることを確認してください。 証明書拡張が少ない新しい証明書の取得を検討してください。
CertificateRevokedException	証明書取り消しの確認が使用可能になっており、取り消す証明書が見つかりました。	該当する証明書は信頼できません。 交換用の証明書を取得して、鍵リング・ファイルに新しい証明書とその秘密鍵が存在することを確認します。

例外	原因	アクション
CertPathBuilderException	証明書チェーンが認識された認証局によって署名されていません。	1. CA 署名済み証明書を使用している場合は、CA 鍵リング・ファイルに、root CA および中間 CA 証明書がすべて存在することを確認します。 2. 自己署名証明書を使用している場合は、リモート証明書の公開部分のコピーを抽出しており、それを CA 鍵リング・ファイルに追加していることを確認します。 実稼働環境では自己署名証明書を使用しないようにしてください。
CertStoreException KeyStoreException	以下のいずれかの理由で、鍵リングからの証明書の読み取り時にエラーが発生しました。 1. 鍵リング・ファイルが損傷している。 2. 鍵リング・ファイルが欠落している。 3. 保管されているパスワードと鍵リング・ファイルのパスワードが一致していない。 4. 暗号ハードウェアを使用するように経路が構成されている場合に、MQIPT が暗号ハードウェアに接続できなかった。	1. 鍵リング・ファイルが読み取れることと、IBM 鍵管理ツールですべての証明書を表示できることを確認します。 2. すべての鍵リングの経路プロパティーが正しいファイル名を指していることを確認します。 3. 保管されている鍵リング・ファイルのパスワードが正しいことを確認します。 **mqiptPW** ツールを使用して正しいパスワードを保管します。 4. 暗号ハードウェアを使用するように経路が構成されている場合は、以下を確認してください。 • Java セキュリティー・プロパティー・ファイルに、IBMPKCS11Impl セキュリティー・プロバイダーがインストールされていることが指定されている。 • Java セキュリティー・プロパティー・ファイルに、IBMPKCS11Impl セキュリティー・プロバイダーの初期化で使用される構成ファイルの完全修飾名が含まれている。 • IBMPKCS11Impl セキュリティー・プロバイダーの初期化に使用される構成ファイルが有効である。
SSLException: 使用可能になっている SSL 暗号スイートに対応する使用可能な証明書または鍵がない。	使用している CipherSuite に対して正しいタイプの鍵を持つ個人証明書が必要です。 例えば、名前が SSL_ECDH_ECDSA_ で始まる CipherSuite には、楕円曲線公開鍵を持つ証明書が必要です。 最もよく使用される CipherSuite には、RSA 公開鍵を持つ証明書が必要です。	IBM 鍵管理ツールを使用して、鍵リング・ファイルを開きます。 「個人証明書」ビューで、各証明書を順番に選択して、表示します。 「**詳細の表示**」をクリックして、「サブジェクト公開鍵」セクションにナビゲートして公開鍵タイプを表示します。 次に、MQIPT **SSLClientCipherSuites** および **SSLServerCipherSuites** 経路プロパティーで、適切な CipherSuite が使用可能になっていることを確認します。

例外	原因	アクション
SSLException: 共通の暗号スイートがない SSLHandshakeException: 共通の暗号スイートがない (No cipher suites in common)	接続の両側で使用可能な CipherSuite のセット間にオーバーラップがないため、ハンドシェークで CipherSuite の同意に失敗しました。 特に、アウトバウンド IBM MQ 接続では単一の暗号のみが使用可能になるため、SSLServer MQIPT 経路では特にこのエラーが発生しやすくなります。 このエラーは、以下の 3 つの条件がすべて真の場合にも発生することがあります。 • 経路に CipherSuite が指定されていない • 経路に構成された鍵リングに適切なサイト証明書が見つからない • 匿名 CipherSuites が使用不可である	MQIPT **SSLClientCipherSuites** および **SSLServerCipherSuites** 経路プロパティーで使用可能な CipherSuite のリストを確認します。 追加の CipherSuite を使用可能にすることを検討してください。 各 IBM MQ チャネルの CipherSpec 値で使用可能にする正しい CipherSuite を判別するために提供されている表を参照してください。 ルート・プロパティー上に CipherSuite が指定されていない場合は、正しいキー・リング・ファイルを参照して、キー・リング・ルートのプロパティーおよびキー・リングに MQIPT が使用できる個人証明書が含まれていることを確認してください。 暗号ハードウェアを使用するように経路が構成されている場合は、IBMPKCS11Impl セキュリティー・プロバイダーの初期化に使用される構成ファイル内の **tokenlabel** 属性に、正しい暗号デバイス・トークン・ラベルを指定していることを確認してください。

MQIPT の mqiptKeyman および mqiptKeycmd

mqiptKeyman (iKeyman) は、IBM MQ ユーザーには既知の証明書および鍵管理アプリケーションです。 **mqiptKeyman** および **mqiptKeycmd** コマンドは、IBM MQ Internet Pass-Thru で使用される鍵リング・ファイルの対称鍵、非対称鍵、デジタル証明書、および証明書要求を管理するために使用できます。 また、鍵リング・ファイル自体の管理にも使用できます。

mqiptKeyman コマンドおよび **mqiptKeycmd** コマンドは、鍵リング・ファイルを指すために 鍵データベース という用語を使用します。これらの用語は同義です。

iKeyman はグラフィカル・ユーザー・インターフェース (GUI) とコマンド行インターフェース (CLI) の 2 つのモードで実行できます。 **mqiptKeyman** コマンドを使用して GUI を開始し、**mqiptKeycmd** コマンドを使用して CLI を実行します。

IBM MQ で証明書を管理するための同等のコマンドは、**strmqikm** (GUI を開始) と **runmqckm** (CLI を実行) です。 IBM MQ コマンドについては、**runmqckm**、**runmqakm**、および **strmqikm** を使用したデジタル証明書の管理で説明しています。

注 : V 9.3.4 Deprecated **mqiptKeycmd** および **mqiptKeyman** ツールは、IBM MQ 9.3.4 から非推奨になりました。 詳細については、**runmqckm**、**runmqakm**、**strmqikm** を使用したデジタル証明書の管理を参照してください。

MQIPT に必要な鍵リング・ファイル・フォーマット

MQIPT で使用するために鍵リング・ファイルを作成する場合は、PKCS#12 ファイル・フォーマットを使用する必要があります。

- UI では、鍵リング・ファイルの作成時に「**鍵データベース・タイプ**」フィールドで PKCS#12 を選択します。
- CLI では、 `mqiptKeycmd -keydb -create` コマンドに `-type pkcs12` パラメーターを含めます。

MQIPT は、PKCS #11 インターフェースに対応した暗号ハードウェアに保管された証明書にアクセスすることもできます。 また、インターフェースを使用して、PKCS #11 ハードウェアにある証明書を管理することもできます。 詳しくは、1073 ページの『MQIPT での PKCS #11 暗号ハードウェアの使用』を参照してください。

MQIPT 用の鍵リング・パスワードの暗号化

鍵リング・ファイルの作成後に、MQIPT がファイルにアクセスするために使用できる形式で鍵リング・パスワードを暗号化する必要があります。 この情報については1058 ページの『MQIPT での鍵リング・パスワードの暗号化』を参照してください。

stash ファイル機能は MQIPT でサポートされないことに注意してください。 stash ファイルを使用する代わりに、**mqiptPW** コマンドを使用して鍵リング・パスワードを暗号化する必要があります。

コマンド行の例

CLI は、IBM MQ **runmqckm** コマンドと同じ構文を使用します。 以下の例に示すように、必要なパラメーターを **mqiptKeycmd** に追加してください。

- PKCS#12 ファイルを作成する場合:

```
mqiptKeycmd -keydb -create -db key.p12 -pw password -type pkcs12
```

- テスト目的の自己署名個人証明書を作成する場合:

```
mqiptKeycmd -cert -create -db key.p12 -pw password -type pkcs12
-label mqipt -dn "CN=Test Certificate,OU=Sales,O=Example,C=US"
-sig_alg SHA256WithRSA -size 2048
```

  このコマンドは、2048 ビット RSA 公開鍵、および RSA と SHA-256 ハッシュ・アルゴリズムを使用したデジタル署名付きのデジタル証明書を作成します。 証明書を作成する場合は、組織のセキュリティー・ニーズに適した公開鍵暗号化アルゴリズム、鍵サイズ、およびデジタル署名アルゴリズムを慎重に選択してください。 詳しくは、1065 ページの『MQIPT に対するデジタル証明書の考慮事項』を参照してください。

  この例では、テスト目的に適した自己署名証明書を使用しています。 ただし、実稼働環境では、代わりに認証局署名済み証明書を使用する必要があります。

  MQIPT v2.0 およびこれより古いバージョンは SHA-2 デジタル署名をサポートしないことに注意してください。したがって、この証明書は以前の MQIPT リリースへのセキュア・ソケット接続の確立には適していません。SHA1WithRSA などの古い署名アルゴリズムが必要となります。

- 実動目的の CA 署名済み証明書の認証要求を作成する場合:

```
mqiptKeycmd -certreq -create -db key.p12 -pw password -type pkcs12 -file cert.req
-label mqipt -dn "CN=Test Certificate,OU=Sales,O=Example,C=US"
-sig_alg SHA256WithRSA -size 2048
```

  このコマンドは、2048 ビット RSA 公開鍵、および RSA と SHA-256 ハッシュ・アルゴリズムを使用したデジタル署名を持つデジタル証明書要求を作成します。 証明書を作成する場合は、組織のセキュリティー・ニーズに適した公開鍵暗号化アルゴリズム、鍵サイズ、およびデジタル署名アルゴリズムを慎重に選択してください。 詳しくは、1065 ページの『MQIPT に対するデジタル証明書の考慮事項』を参照してください。

- CA 署名付きの個人証明書ファイル `cert.crt` をキー・リング・ファイルに受信するには:

```
mqiptKeycmd -cert -receive -db key.p12 -pw password -type pkcs12 -file cert.crt
```

  個人証明書に署名した CA の CA 証明書が必ず CA 鍵リング・ファイル内に存在するようにする必要があります。例:

```
mqiptKeycmd -cert -add -db key.p12 -pw password -type pkcs12 -file ca.crt -label rootCA
```

MQIPT に対するデジタル証明書の考慮事項

考慮すべきポイントには、証明書の鍵サイズ、適切な証明書のデジタル署名アルゴリズムとデジタル証明書の選択、および CipherSuite compatibilityDigital 証明書と CipherSuite 互換性が含まれます。

MQIPT に対する証明書鍵サイズの考慮事項

公開鍵のサイズは、組織のセキュリティー・ポリシーおよび使用している暗号化アルゴリズムによって異なります。 一般的には、鍵サイズが大きいほど安全になります。 以下の表は、使用する必要がある最小鍵サイズを示しています。

アルゴリズム	最小鍵サイズ (ビット)
楕円曲線	256
RSA	2048

証明書または認証要求の作成時に、証明書の鍵サイズを指定します。

- **mqiptKeycmd** CLI コマンドを使用する場合は、**-size** パラメーターで鍵サイズを指定します。
- **mqiptKeyman** GUI を使用する場合は、「証明書作成」ウィンドウの「**鍵サイズ**」フィールドで鍵サイズを指定します。

適切な証明書デジタル署名アルゴリズムの選択

デジタル証明書の偽造を防ぐために、強力なデジタル署名アルゴリズムを使用することが重要です。 証明書を作成または要求する場合、適切なアルゴリズムを慎重に選択してください。

MD5 または SHA-1 に基づいた古いデジタル署名アルゴリズム の使用は避けるべきです。これらのアルゴリズムは最新の使用法では十分にセキュアではなくなりました。 可能な場合は、SHA-256 と RSA の組み合わせ (SHA256WithRSA) などの新しい SHA-2 ベースのデジタル署名アルゴリズムを使用してください。

バージョン 2.1 より古いバージョンの MQIPT は SHA-2 デジタル署名をサポートしないため、前の MQIPT リリースとの相互運用性を確保するために SHA1WithRSA デジタル署名アルゴリズムを使用してください。 ただし、古いバージョンの MQIPT のアップグレードを計画し、MD5 および SHA-1 デジタル署名アルゴリズムの使用を段階的に中止する必要があります。

- **mqiptKeycmd** CLI コマンドを使用する場合は、**-sig_alg** パラメーターでデジタル署名アルゴリズムを指定します。
- **mqiptKeyman** GUI を使用する場合は、「証明書作成」ウィンドウの「**署名アルゴリズム**」フィールドでデジタル署名アルゴリズムを指定します。

MQIPT におけるデジタル証明書と CipherSuite の互換性

すべての CipherSuites がすべてのデジタル証明書と共に使用できるわけではありません。 さまざまなタイプの CipherSuite があり、CipherSuite 名の接頭部でグループ化されています。 CipherSuite のタイプごとに、使用できるデジタル証明書のタイプに関する制限があります。 これらの制限は、すべての MQIPT SSL/TLS 接続に適用されますが、楕円曲線暗号のユーザーにとっては、特に重要です。 セキュア・ソケット・ハンドシェークを実行するときに、MQIPT は自動的に個人証明書を選択して、ネゴシエーションされる CipherSuite に適切である自身の身元を証明します。 多くの場合、MQIPT は自動的にリモート・ピアと相互に作動します。 ただし、特定のシナリオでは、特定の MQIPTCipherSuite を使用して、リモート IBM MQ システムと相互運用することが必要な場合があります。 MQIPT に付属の **mqiptKeyman** アプリケーションは、DSA および RSA 公開鍵のみを使用して証明書および認証要求を作成できます。 さらに、IBM MQ **runmqakm** ユーティリティーは、楕円曲線公開鍵を使用した証明書および認証要求の作成が可能です。 認証局に問い合わせて、他のタイプの証明書の作成について助言を受けてください。

使用するデジタル証明書のタイプは、使用している CipherSuite のタイプにより異なります。

- SSL_ECDH_ECDSA_ および SSL_ECDHE_ECDSA_ で始まる名前の CipherSuites では、楕円曲線公開鍵を持つデジタル証明書が必要です。

- 名前に *anon* が含まれる CipherSuites は匿名です。これらは、リモート・ピアを特定するためにデジタル証明書を必要としません。 このような CipherSuites は、認証の代替手段が使用されるネットワーク内の証明書ライフサイクル管理のオーバーヘッドを回避できますが、一般的にはこれらは認証がないため、使用を避けてください。
- その他の CipherSuites では、RSA 公開鍵を持つデジタル証明書が必要です。

注 : **mqiptKeyman** および **mqiptKeycmd** ツールは、楕円曲線公開鍵を持つ証明書または認証要求を作成できません。 この目的のために、 IBM MQ に用意されている **runmqakm** コマンドを使用できます。 **runmqakm** コマンドについては、**runmqckm**、**runmqakm**、および **strmqikm** を使用したデジタル証明書の管理で説明しています。

MQIPT の証明書出口

証明書出口の目的は、MQIPT が受け取る SSL/TLS ピア証明書を検証することです。

MQIPT の経路を、新しい接続を作成するときは SSL/TLS クライアントとして機能するように構成し、接続要求を受け取るときは SSL/TLS サーバーとして機能するように構成できます。 SSL/TLS のハンドシェーク・プロセス中に、SSL/TLS クライアントはサーバーからピア証明書を受け取り、サーバーを認証するためにその証明書が使用されます。 また、SSL/TLS サーバーもクライアントからピア証明書を受け取り、クライアントを認証するためにその証明書が使用されます。

証明書出口は、MQIPT がピア証明書を受け取るときに呼び出され、さらに検証を実行できるようになります。 出口によって検出される例外は MQIPT によって検出され、接続要求が終了します。 したがって、出口によってすべての例外が検出され、MQIPT に適切な戻りコードが渡されることが推奨されます。

証明書出口を表示するためのサンプルが提供されています。詳細については、証明書出口を使用し、SSL/TLS サーバーの認証を参照してください。

注 : MQIPT は単一の Java Virtual Machine で実行されます。このため、ユーザー定義の証明書出口は、以下のように MQIPT の通常の動作に悪影響を及ぼす可能性があります。

- システム・リソースへの影響
- ボトルネックの発生
- パフォーマンスの低下

証明書出口を実稼働環境に実装する場合は、その影響を十分にテストする必要があります。

MQIPT の com.ibm.mq.ipt.exit.CertificateExit クラス

com.ibm.mq.ipt.exit.CertificateExit クラスは抽象クラスであり、 SSLExitName プロパティーで定義されたクラスによって実装する必要があります。

このクラスには、必要に応じてオプションで指定変更可能な出口および一部の public メソッドを実行するためのデフォルトの実装が含まれます。 サポートされるメソッドの完全なリストは以下のとおりです。

方法

public int init(IPTTrace)

init メソッドは、出口が MQIPT によってロードされるときに MQIPT によって呼び出されます。このメソッドは、検証プロセス中に使用されるデータのロードなど、出口の初期化を実行するために実装できます。 デフォルトの実装では何も実行されません。

public int refresh(IPTTrace)

refresh メソッドは、検証プロセス中に使用されるディスク・データの再ロードなど、データのリフレッシュを行うために実装されます。 このメソッドは、MQIPT 管理者がリフレッシュ・コマンドを発行したときに呼び出されます。 デフォルトの実装では何も実行されません。

public void close(IPTTrace)

close メソッドは、経路が停止するとき、または MQIPT が クローズするときにハウスキーピングを行うために実装されます。 デフォルトの実装では何も実行されません。

public CertificateExitResponse validate(IPTTrace)

validate メソッドは、ピア証明書の検証を行うために呼び出されます。 MQIPT に情報を戻すために戻りオブジェクト (戻りコードや、接続ログに追加可能な テキストなど) を使用できます。 デフォルトの実装では、CertificateExitResponse.OK で CertificateExitResponse が 戻されます。

プロパティーの取得のためにサポートされるメソッドは、以下のとおりです。

public int getListenerPort()
ListenerPort プロパティーに定義されている、経路リスナー・ポートを取得します。

public String getDestination()
Destination プロパティーに定義されている、宛先アドレスを取得します。

public int getDestinationPort()
DestinationPort プロパティーに定義されている、宛先リスナー・ポートのアドレスを取得します。

public String getClientIPAddress()
接続要求を出したクライアントの IP アドレスを取得します。

public int getClientPortAddress()
接続要求を出したクライアントによって使用されるポート・アドレスを取得します。

public boolean isSSLClient()
出口が SSL/TLS クライアントまたは SSL/TLS サーバーの いずれとして呼び出されるかを判別するために使用されます。 true を戻す場合、出口は接続の クライアント・サイドで、サーバーから取得した証明書を検証します。 false を戻す場合、出口は接続の サーバー・サイドで、クライアントによって送信された証明書を検証します。 経路が SSL/TLS サーバーと SSL/TLS クライアントの 両方として機能することは有効であり、トラフィックの暗号化解除および再暗号化を行います。 この場合、単一の出口クラスが存在しますが、 クラスの一部のインスタンスはクライアントとして呼び出され、一部はサーバーとして呼び出されます。 isSSLClient を使用して、特定のインスタンスの状態を判別できます。

public int getConnThreadID()
接続要求を処理しているワーカー・スレッドの ID を取得するために使用されます。 これはデバッグに有用です。

public String getChannelName()
接続要求で使用される IBM MQ のチャネル名 を取得します。 これは、着信要求で SSL/TLS が 使用されておらず、MQIPT が SSL/TLS クライアントとして 機能している場合にのみ使用できます。

public String getQMName()
接続要求で使用される IBM MQ キュー・マネージャーの 名前を取得します。 これは、クライアント要求で SSL/TLS が 使用されておらず、MQIPT が SSL/TLS クライアントとして 機能している場合にのみ使用できます。

public boolean getTimedout()
タイムアウトが有効期限切れであるかどうかを判別するために出口で使用されます。

public IPTCertificate getCertificate()
検証が必要な SSL/TLS 証明書を取得します。

public String getExitData()
SSLExitData プロパティーで定義されている、出口データを取得します。

public String getExitName()
SSLExitName プロパティーで定義されている、出口名を取得します。

MQIPT の com.ibm.mq.ipt.exit.CertificateExitResponse クラス

このクラスは、証明書が検証された後に、MQIPT に 情報を戻すために使用します。

コンストラクター

public CertificateExitResponse(*int rc*, string message)

このコンストラクターは、戻りコードおよび一部のメッセージ・テキスト を戻すために使用できます。 可能な理由コードは以下のとおりです。

- ExitRc.OK

- ExitRc.VALIDATE_ERROR
- ExitRc.VALIDATE_REJECTED

public CertificateExitResponse(*int rc*)

このコンストラクターは、メッセージ・テキストなしで戻りコード を戻すために使用できます。 可能な理由コードは以下のとおりです。

- ExitRc.OK
- ExitRc.VALIDATE_ERROR
- ExitRc.VALIDATE_REJECTED

public CertificateExitResponse()

このコンストラクターは、メッセージ・テキストなしで戻りコード ExitRc.OK を 戻すために使用できます。

方法

public String getVersion()

このメソッドは、このクラスのバージョンを戻します。

public String toString

このメソッドは、「Reason code: 4, Message: Failed CRL check」など、 応答のストリング表現を戻します。

MQIPT の com.ibm.mq.ipt.exit.IPTCertificate クラス

このクラスには、検証対象の SSL/TLS 証明書が含まれます。

方法

public int getVersion()

このメソッドは、このクラスのバージョンを戻します。

public byte [] getDerEncoding()

このメソッドは、X.509 証明書の ASN.1/DER エンコードを戻すか、エラーがある場合は NULL を戻します。

public byte [] getPemEncoding()

このメソッドは、X.509 証明書の PEM (BASE64) エンコードを戻すか、エラーがある場合は NULL を戻します。

public String getLabel()

このメソッドは証明書ラベルを戻すか、エラーがある場合は NULL を戻します。

public String getName()

このメソッドは、証明書の識別名を戻すか、使用できない場合は NULL を戻します。 以下に例を示します。

```
CN=Test Queue Manager,OU=Sales,O=Example,L=London,C=GB
```

public String getIssuerName()

このメソッドは、証明書の発行者の識別名を戻すか、使用できない場合は NULL を戻します。 以下に例を示します。

```
CN=Certificate Authority,OU=Security,O=Example,L=New York,C=US
```

public IPTCertificate getSigner()

このメソッドは署名者証明書を戻すか、使用できない場合は NULL を戻します。 自己署名証明書の場合は、それ自体への参照を戻します。

public String toString()

このメソッドは、証明書のストリング表現を戻します。

MQIPT の com.ibm.mq.ipt.exit.IPTTrace クラス

MQIPT トレース関数は、メソッドへの入り口およびメソッドからの出口で使用できる入り口および出口呼び出しを提供します。 また、有用な情報をトレースするためのさまざまなデータ呼び出しもあります。

方法

public void entry(String *fid*)

この *fid* は、クラスやメソッド名など、呼び出し場所を特定するために使用されます。

このメソッドは、制御フローがメソッドに入るポイントを記録するのに適したレベルのインデントを使用してトレース出力ファイルに入り口を書き込みます。 この呼び出しはオプションですが、使用する場合は、対応する「exit(String)」への呼び出しを同じメソッド内で使用する必要もあります。

public void exit(String *fid*)

この *fid* は、クラスやメソッド名など、呼び出し場所を特定するために使用されます。

このメソッドは、制御フローがメソッドから出るポイントを記録するのに適したレベルのインデントを使用してトレース出力ファイルに出口を書き込みます。 このメソッドが使用されるのは、同じメソッド内で「entry(String)」への呼び出しが既に使用されている場合のみです。

public void exit(String *fid*, int *rc*)

この *fid* は、クラスやメソッド名など、呼び出し場所を特定するために使用され、*rc* はメソッドからの数値の戻りコードです。 この trace メソッドは、整数を返すメソッドからの出口を記録するために使用する必要があります。

このメソッドは、制御フローがメソッドから出るポイントと、そのメソッドからの数値の戻りコードを記録するのに適したレベルのインデントを使用してトレース出力ファイルに出口を書き込みます。 このメソッドが使用されるのは、同じメソッド内で「entry(String)」への呼び出しが既に使用されている場合のみです。

public void exit(String *fid*, boolean *rc*)

この *fid* は、クラスやメソッド名など、呼び出し場所を特定するために使用され、*rc* はメソッドからのブール値の戻りコードです。 この trace メソッドは、ブール値を返すメソッドからの出口を記録するために使用する必要があります。

このメソッドは、制御フローがメソッドから出るポイントと、そのメソッドからのブール値の戻りコードを記録するのに適したレベルのインデントを使用してトレース出力ファイルに出口を書き込みます。 このメソッドが使用されるのは、同じメソッド内で「entry(String)」への呼び出しが既に使用されている場合のみです。

public void data(String *fid*, String data)

この *fid* は、クラスやメソッド名など、呼び出し場所を特定するために使用されます。

このメソッドは、トレース出力ファイルにいくつかのストリング・データを書き込みます。

public void data(String *fid*, int data)

この *fid* は、クラスやメソッド名など、呼び出し場所を特定するために使用されます。

このメソッドは、トレース出力ファイルにいくつかの整数データを書き込みます。

public void data(String *fid*, byte[])

この *fid* は、クラスやメソッド名など、呼び出し場所を特定するために使用されます。

このメソッドは、トレース出力ファイルにいくつかのバイナリー・データを書き込みます。

サンプル・トレース

出口での問題の診断を容易にするために、MQIPT と同じトレース機能を使用できます。また、独自のトレース機能を実装することもできます。 MQIPT トレース機能を使用する場合は、メソッドへの入り口とメソッドからの出口で入り口と出口呼び出しをそれぞれ使用できます。 また、以下の例に示されている役立つ情報をトレースするためのさまざまなデータ呼び出しがあります。

```
/**
* This method is called to initialize the exit (for example, for
* loading validation information) and place itself in a ready
* state to validate connection requests.
*/
 public int init(IPTTrace t) {
 final String fid = "MyExit.init";

 // Trace entry into this method
 t.entry(fid);

 // Trace useful information
 t.data(fid, "Starting exit - MQIPT version " + getVersion());

 // Perform initialization and load any data
 t.data(fid, "Ready for work");

 // Trace exit from this method
 t.exit(fid);

 return ExitRc.OK;
}
```

このメソッドは、以下の例に示されている形式でトレースを生成します。

```
16:36:48.625    14   5000-1s    ------{ ConnectionThread.setCertificateExit()
16:36:48.625    14   5000-1s     Creating instance of certificate exit
16:36:48.625    14   5000-1s     Calling init() of certificate exit
16:36:48.625    14   5000-1s    -------{ MyExit.init()
16:36:48.625    14   5000-1s     Starting exit - MQIPT version 2.1.0.0
16:36:48.625    14   5000-1s    Ready for work
16:36:48.625    14   5000-1s    -------} MyExit.init() rc=0
16:36:48.625    14   5000-1s    ------} ConnectionThread.setCertificateExit() rc=0
```

MQIPT の証明書出口戻りコード

さまざまな状況で証明書出口を呼び出すときに MQIPT が認識する戻りコード

以下の戻りコードは、以下の状態での証明書出口の呼び出し時に MQIPT によって認識されます。

戻りコード	説明	init	Validate	リフレッシュ
ExitRc.OK	要求が正常に完了しました。	yes	yes	yes
ExitRc.INIT_ERROR	初期化要求が失敗し、経路が使用不可になります。	yes		
ExitRc.REFRESH_ERROR	リフレッシュ要求が失敗し、経路が使用不可になります。			yes
ExitRc.VALIDATE_ERROR	検証プロセスが失敗し、接続要求が拒否されました。		yes	
ExitRc.VALIDATE_REJECTED	検証要求と接続要求が拒否されました。		yes	

MQIPT での LDAP と CRL

MQIPT では、デジタル証明書に対する証明書取り消しリスト (CRL) 認証を実行するための Lightweight Directory Access Protocol (LDAP) サーバーの使用がサポートされます。

LDAP サポートは、 IBM MQ と MQIPT の両方に同じ LDAP サーバーを使用できるため、 IBM MQ の場合と同様の方法で実装されています。

SSL/TLS ハンドシェーク時に、通信するパートナーはデジタル証明書を使用して互いに認証します。 認証には、受信された証明書が引き続き信頼できるかどうかの検査が組み込まれる場合があります。 認証局 (CA) は、次の理由を含め、さまざまな理由で証明書を取り消します。

- 所有者が別の組織に移動した。
- 秘密鍵が秘密でなくなった。

CA は、証明書取り消しリスト (CRL) で、取り消された個人用証明書を公開します。 取り消された CA 証明書は、権限取り消しリスト (ARL) で公開されます。 これ以降の CRL の参照は ARL にも適用されることに注意してください。

IBM MQ での LDAP サーバーの使用、および CRL と ARL の管理について詳しくは、証明書取り消しリストおよび権限取り消しリストの取り扱いを参照してください。

MQIPT では、各経路に対して最大 2 つの LDAP サーバーをサポートできます。 最初の LDAP サーバーはメイン・サーバーとして扱われ、2 番目の LDAP サーバーはバックアップとして保持されます。 2 番目のサーバーが使用されるのは、メイン・サーバーに到達できない場合のみです。 バックアップ・サーバーはメイン・サーバーのミラー・イメージでなければなりません。

LDAP サーバーに保管されている情報へのアクセスは、LDAP ユーザー ID とパスワード・プロパティーを使用して、ユーザー ID とパスワードで保護できます。 LDAP サーバー・パスワードは、 IBM MQ 9.1.5 から MQIPT 構成で暗号化できます。 MQIPT によって使用されるパスワードの暗号化について詳しくは、1085 ページの『MQIPT に保管されているパスワードの暗号化』を参照してください。

MQIPT が鍵リング・ファイルから PKCS#12 トークンをロードする際に、CA 証明書で CRL の妥当性が確認されます。 CA 証明書に CRL が添付されている場合は、有効期限が切れていないかどうかが確認され、切れている場合は、LDAP サーバーから新しい CRL が取得されます。 取得されたすべての CRL が現行のトークンにロードされ、その CA 証明書に添付されます。

メイン LDAP サーバーへのクエリーの送信時に指定された CA と一致するエントリーがない場合は、その CA の CRL がないと見なされ、バックアップ・サーバーは使用されません。 ただし、メイン LDAP サーバーに到達できないか、一定の時間フレーム内に何も返されない場合は、バックアップ・サーバーが使用されます。 バックアップ・サーバーでエラーが発生した場合は、クライアント接続が終了します。 このアクションは、**LDAPIgnoreErrors** プロパティーを `true` に設定することで指定変更できます。

MQIPT が取得した CRL はすべてキャッシュで保持され、その経路でのすべての接続で共有されます。 キャッシュされた CRL の有効期限が切れた場合、CRL はキャッシュから除去され、LDAP サーバーから新しい CRL が取得されます。 新しい CRL が使用不可の場合、接続は引き続き拒否されます。

LDAP サーバーから取得した CRL の有効期限も確認され、警告メッセージが表示されます (MQCPW001)。 有効期限切れの CRL は引き続きシステムにロードされますが、その CRL を参照するすべての接続要求は拒否されます。 LDAP サーバーの有効期限切れの CRL は現行のものと置き換える必要があります。

LDAPCacheTimeout プロパティーを使用して、CRL キャッシュがクリアされる頻度を制御できます。 デフォルト値は 1 日です。 この値を `0` に設定すると、経路が再始動するまでキャッシュ・エントリーはクリアされません。

有効期限切れの CRL は鍵リング・ファイル内または LDAP サーバー上に保管できます。 新しい CRL が発行されていない場合、以降の接続要求は拒否されます。 有効期限切れの CRL は **IgnoreExpiredCRLs** プロパティーを使用可能にすることで無視できます。

注 : LDAPIgnoreErrors プロパティーまたは **IgnoreExpiredCRLs** プロパティーを使用可能にすると、取り消された証明書が SSL/TLS 接続で使用される可能性があります。

MQIPT での複数値証明書の識別名 OU プロパティー

証明書の識別名内の複数の組織単位 (OU) 値を突き合わせることができます。

以下の経路プロパティーでは、複数の OU 値の突き合わせがサポートされるようになりました。

- **SSLClientDN_OU**

- **SSLClientSiteDN_OU**
- **SSLServerDN_OU**
- **SSLServerSiteDN_OU**

複数の OU 値を突き合わせるには、 経路プロパティー値でコンマを分離文字として使用します。 以下に例を示します。

```
SSLClientDN_OU=Sales, Europe
```

この 場合、OU=Sales と OU=Europe の両方を持つ証明書と一致します。 OU 値は、IBM MQ SSLPEER フィルター内の複数の OU 値と同じ順序で突き合わされます。

[route] セクションに、同じ経路プロパティーを 2 回以上指定しないでください。 複数の OU 値を正しく突き合わせるには、前の例に示しているように、プロパティーを 1 回指定します。 同じ mqipt.conf セクションに同じ属性を二回以上入力すると、最後の値が有効になります。 例えば、以下のエントリーでは、2 行目によって最初の行が 指定変更されるため、Europe のみが一致することになります。

```
SSLClientDN_OU=Sales
SSLClientDN_OU=Europe
```

OU 値内のリテラルのコンマと突き合わせる必要がある場合は、コンマの直前にエスケープ文字として円記号 (¥) を挿入します。 以下に例を示します。

```
SSLClientDN_OU=Sales\, Europe
```

これは、単一値 OU=Sales, Europe と一致します。 コンマが直後に 続かない円記号は、リテラルの円記号と一致します。

以前のリリースの MQIPT からアップグレードしており、OU 値内の コンマと一致する機能に依存している場合は、以前の動作を維持するために、円記号エスケープ文字 を OU 経路プロパティーに挿入する必要があります。

Deprecated MQIPT での非推奨プロトコルおよび暗号スイートの使用可能化

デフォルトでは、非セキュアと見なされるセキュア・ソケット・プロトコルおよび暗号スイートは、MQIPT に付属の Java runtime environment (JRE) では無効になっています。 これらの非推奨のプロトコルおよび暗号スイートを使用するには、その前に有効にする必要があります。

このタスクについて

潜在的な危険を認識したうえで、セキュアでないと見なされているプロトコルまたは暗号スイートのいずれかを MQIPT で使用する必要がある場合は、以下の手順に従って、使用する必要があるプロトコルまたは暗号スイートを有効にします。

注 : 推奨されないプロトコルおよび暗号スイートは、TLS コマンド・ポートでは使用できません。

手順

1. *mqipt_path*/java/jre/lib/security ディレクトリー内にある java.security ファイルを編集します。*mqipt_path* は MQIPT がインストールされている場所です。
2. jdk.tls.disabledAlgorithms プロパティーの無効にするアルゴリズムのリストから対応する項目を削除して、プロトコルまたはアルゴリズムのサポートを JRE に追加します。
   - プロトコルのサポートを追加するには、無効にするアルゴリズムのリストからそのプロトコルを削除します。 例えば、TLS 1.0 のサポートを追加するには、リストから TLSv1 を削除します。
   - 暗号スイートのサポートを追加するには、無効にするアルゴリズムのリストから対応するアルゴリズムを削除します。 例えば、SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA 暗号スイートのサポートを追加するには、リストから 3DES_EDE_CBC および DESede を削除します。

3. JRE で SSL 3.0 を有効にするには、システム・プロパティー com.ibm.jsse2.disableSSLv3=false も設定する必要があります。

   mqipt コマンドを使用してコマンド行から MQIPT を開始する場合は、環境変数 **MQIPT_JVM_OPTIONS** を使用してプロパティーを設定できます。 以下に例を示します。

   ```
   set MQIPT_JVM_OPTIONS=-Dcom.ibm.jsse2.disableSSLv3=false
   ```

   Windows MQIPT が Windows サービスとしてインストールされている場合は、Windows レジストリー内の HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MQInternetPassThru キーの下にストリング値を定義することによって、このプロパティーを設定できます。 値には、以下の属性が必要です:

   名前
   MqiptJvmOptions

   バリューデータ
   -Dcom.ibm.jsse2.disableSSLv3=false
4. MQIPT 経路で SSL 3.0、TLS 1.0、または TLS 1.1 を有効にするには、対応するプロトコルを **SSLServerProtocols** または **SSLClientProtocols** 経路プロパティーに追加します。
5. JRE プロパティーに対する変更を有効にするために、MQIPT を再始動します。

MQIPT での PKCS #11 暗号ハードウェアの使用

MQIPT は、PKCS #11 インターフェースに対応した暗号ハードウェアに保管されたデジタル証明書にアクセスできます。

始める前に

暗号ハードウェアを使用するように MQIPT を構成する前に、暗号カード、カード・ドライバー、 および関連するサポート・ソフトウェアがインストールされ、正しく機能していることを確認してください。

MQIPT での PKCS #11 暗号ハードウェアのサポートは、IBM Java PKCS11 Cryptographic Provider (IBMPKCS11Impl プロバイダー) によって提供されます。 IBMPKCS11Impl プロバイダー、および Java 8 でサポートされる暗号カードのリストについて詳しくは、IBM PKCS11 Cryptographic Provider を参照してください。

このタスクについて

MQIPT でアクセスする個人証明書および CA 証明書は、暗号ハードウェアの鍵ストアに保管できます。 ただし、通常、PKCS#11 デバイスには大量の署名者証明書を保管するための十分なスペースがないため、CA 証明書にはファイル・ベースの別の鍵ストアを使用することをお勧めします。

以下の手順に従って、暗号ハードウェアの鍵ストア内の証明書を使用するように MQIPT を構成します。

注 : MQIPT による暗号化ハードウェアの使用は、IBM MQ Advanced 機能です。 この機能を使用するには、MQIPT 経路を使用して接続されているローカル・キュー・マネージャーにも、 IBM MQ Advanced、 IBM MQ Appliance、 IBM MQ Advanced for z/OS VUE、または IBM MQ Advanced for z/OS のライセンスが必要です。

手順

1. IBMPKCS11Impl プロバイダーの初期化時に使用される構成ファイルを作成します。

   IBMPKCS11Impl プロバイダーでサポートされる各ハードウェア暗号カードのサンプル構成ファイルをダウンロードし、システムに合わせてサンプルを構成します。 これらのサンプルは、 IBM Documentation for Java の 構成ファイルのトピックからダウンロードできます。

   構成ファイルはテキスト・ファイルであり、少なくとも以下の属性が含まれている必要があります。

   名前
   プロバイダー・インスタンスの名前の接尾部。

library
暗号ハードウェアに付属の PKCS #11 ライブラリーの完全修飾名。

tokenlabel
PKCS #11 暗号デバイスのトークン・ラベル。

例えば、構成ファイルには以下のようなエントリーが含まれます。

```
name = IPTPKCS11Provider
library = /usr/lib64/pkcs11/PKCS11_API.so
tokenlabel = icatoken
```

2. MQIPT インストール・ディレクトリーの java/jre/lib/security サブディレクトリーにある Java セキュリティー・プロパティー・ファイル(java.security)を編集します。
   a) ファイル内にまだない場合は IBMPKCS11Impl セキュリティー・プロバイダーを追加します。
   例えば、次の行を追加します。

   ```
   security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
   ```

   b) 構成ファイルの完全修飾名をプロバイダー名の後に追加します。
   例えば、ステップ 1073 ページの『1』で作成した構成ファイルが /opt/mqipt/pkcs11.cfg と呼ばれる場合、このパスをセキュリティー・プロバイダーと同じ行に追加する必要があります:

   ```
   security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl /opt/mqipt/
   pkcs11.cfg
   ```

3. CA 証明書に鍵リング・ファイルを使用する場合は、暗号ハードウェアに CA 証明書を保管する代わりに、PKCS #12 形式の CA 鍵リング・ファイルを作成します。
   CA 鍵リング・ファイルは、**mqiptKeyman** グラフィカル・ユーザー・インターフェース (GUI) または **mqiptKeycmd** コマンド・ライン・インターフェース (CLI) のどちらでも作成できます。
   - CLI を使用するには、次のコマンドを入力します。

     ```
     mqiptKeycmd -keydb -create -db filename -pw password -type pkcs12
     ```

     filename は作成する鍵リング・ファイルの名前、 *password* は鍵リングのパスワードです。
   - GUI を使用するには、以下の手順に従います。
     a. コマンド **mqiptKeyman** を入力して、GUI を開始します。
     b.「**鍵データベース・ファイル**」>「**オープン**」をクリックします。
     c.「**鍵データベース・タイプ**」をクリックし、**PKCS11Config** を選択します。
     d.「**OK**」をクリックします。「Open Cryptographic Token (暗号トークンのオープン)」ウィンドウが開きます。
     e. 証明書の保管に使用する暗号デバイス・トークン・ラベルを選択します。
     f.「**暗号トークンのパスワード**」フィールドに、暗号ハードウェアへのアクセスに必要なパスワードを入力します。
     g. 新しい CA 鍵リング・ファイルを作成するには、「**新規 2 次鍵データベース・ファイルの作成**」を選択します。
     h.「**鍵データベース・タイプ**」をクリックし、**PKCS12** を選択します。
     i.「**ファイル名**」フィールドに、CA 鍵リングのファイル名を入力します。
     j.「**ロケーション**」フィールドに、CA 鍵リング・ファイルの絶対パスを入力します。
     k.「**OK**」をクリックします。「Password Prompt (パスワード・プロンプト)」ウィンドウが開きます。
     l.「**パスワード**」フィールドに CA 鍵リングのパスワードを入力し、「**パスワードの確認**」フィールドにそのパスワードをもう一度入力します。
     m.「**OK**」をクリックします。
4. **mqiptKeycmd** または **mqiptKeyman** を使用して、暗号ハードウェアの個人証明書を要求します。

- CLI を使用するには、次のコマンドを入力します。

```
mqiptKeycmd -certreq -create -crypto module_name -tokenlabel hardware_token
            -pw password -label label -size key_size
            -sig_alg algorithm -dn distinguished_name -file filename
```

ここで、

-crypto *module_name*
暗号化ハードウェアに用意されている PKCS #11 ライブラリーの完全修飾名を指定します。

-tokenlabel *token_label*
PKCS #11 暗号デバイス・トークン・ラベルを指定します。

-pw *password*
暗号化ハードウェアへアクセスするためのパスワードを指定します。

-label *label*
証明書ラベルを指定します。

-size *key_size*
鍵のサイズを指定します。 値は、512、1024、2048、または 4096 のいずれかです。

-sig_alg *algorithm*
項目の鍵ペアの作成に使用される非対称署名アルゴリズムを指定します。 値は、MD2_WITH_RSA、MD2WithRSA、 MD5_WITH_RSA、MD5WithRSA、SHA1WithDSA、SHA1WithECDSA、SHA1WithRSA、SHA2/ECDSA、SHA224WithECDSA、SHA256_WITH_RSA、SHA256WithECDSA、 SHA256WithRSA、SHA2WithECDSA、SHA3/ECDSA、SHA384_WITH_RSA、SHA384WithECDSA、 SHA384WithRSA、SHA3WithECDSA、SHA5/ECDSA、SHA512_WITH_RSA、SHA512WithECDSA、 SHA512WithRSA、SHA5WithECDSA、SHA_WITH_DSA、SHA_WITH_RSA、または SHAWithDSA のいずれかです。 デフォルト値は SHA256WithRSA です。

-dn *distinguished_name*
二重引用符で囲んだ X.500 識別名を指定します。

-file *filename*
認証要求のファイル名を指定します。

- GUI を使用するには、以下の手順に従います。
  a.「**Create (作成)**」メニューから、「**New Certificate Request (新規認証要求)**」をクリックする。
  b.「**鍵ラベル**」フィールドに証明書ラベルを入力します。
  c. 必要な「**鍵サイズ**」と「**署名アルゴリズム**」を選択します。
  d.「**共通名**」および「**組織**」に値を入力し、「**国**」を選択します。 残りのオプション・フィールドでは、デフォルト値を受け入れるか、別の値を入力または選択します。
  e.「**認証要求を保管するファイル名の入力**」 フィールドで、デフォルトの certreq.arm を受け入れるか、絶対パスを指定して新しい値を入力します。
  f.「**OK**」をクリックします。
  g.「**Personal Certificate Requests (個人用証明書の要求)**」 リストに、作成した新しい個人用証明書要求のラベルが表示されます。 証明書要求は、選択したファイルに保管されます。

5. CA から個人証明書が送信されたら、CA 証明書を暗号鍵ストアまたは CA 鍵リング・ファイルに追加します (まだ存在しない場合)。

- CLI を使用して CA 証明書を CA 鍵リング・ファイルに追加するには、以下のコマンドを入力します。

```
mqiptKeycmd -cert -add -db filename -pw password -type pkcs12
            -label label -file cert_filename
```

filename は CA 鍵リング・ファイルの名前、 *password* は CA 鍵リングのパスワード、*label* は証明書に付加されたラベル、 *cert_filename* は CA 証明書が含まれているファイルの名前です。

- CLI を使用して CA 証明書を暗号ハードウェアに追加するには、以下のコマンドを入力します。

```
mqiptKeycmd -cert -add -crypto module_name -tokenlabel hardware_token
            -pw password -label label -file cert_filename
```

module_name は 暗号ハードウェアに付属の PKCS #11 ライブラリーの完全修飾名、 *hardware_token* は PKCS #11 暗号デバイスのトークン・ラベル、 *password* は暗号ハードウェアにアクセスするためのパスワード、 *label* は証明書に付加されたラベル、 *cert_filename* は CA 証明書が含まれているファイルの名前です。

- GUI を使用するには、以下の手順に従います。
  a. 「**Key database content (鍵データベースの内容)**」フィールドで、「**Signer Certificates (署名者証明書)**」を選択する。
  b. **追加** をクリックします。 「Add CA's Certificate from a File (ファイルからの CA の証明書の追加)」ウィンドウが開きます。
  c. 証明書が保管されている証明書ファイルの名前と位置を入力するか、「**Browse (ブラウズ)**」 をクリックして名前と位置を選択する。
  d. 「**OK**」をクリックします。 「Enter a Label (ラベルの入力)」ウィンドウが開きます。
  e. 「Enter a Label (ラベルの入力)」ウィンドウに、証明書の名前を入力する。
  f. 「**OK**」をクリックします。 証明書が鍵データベースに追加されます。

6. CA から提供された個人証明書を暗号ハードウェアの鍵ストアに受け取ります。
   - CLI を使用するには、次のコマンドを入力します。

```
mqiptKeycmd -cert -receive -file filename -crypto module_name
            -tokenlabel hardware_token -pw password
```

filename は 受け取る証明書が含まれているファイルの名前、*module_name* は暗号ハードウェアに付属の PKCS #11 ライブラリーの完全修飾名、 *hardware_token* は PKCS #11 暗号デバイスのトークン・ラベル、 *password* は暗号ハードウェアにアクセスするためのパスワードです。

CA 証明書が暗号ハードウェアではなく CA 鍵リングに保管されている場合は、**mqiptKeycmd** コマンドでは個人証明書を暗号鍵ストアに受け取るときに CA 鍵リングにアクセスできないため、 証明書チェーンを検証できないという警告を受け取ります。

   - GUI を使用するには、以下の手順に従います。
     a. 「**Receive (受信)**」をクリックする。 「Receive Certificate from a File (ファイルからの証明書の受信)」ウィンドウが開きます。
     b. 新しい個人用証明書の証明書ファイルの名前と位置を入力するか、「**Browse (ブラウズ)**」をクリックして名前と位置を選択する。
     c. 「**OK**」をクリックします。 「**Personal Certificates (個人用証明書)**」フィールドに、追加した新しい個人用証明書のラベルが表示されます。

7. **mqiptPW** コマンドを使用して、暗号ハードウェアにアクセスするためのパスワードを暗号化します。

   以下のコマンドを入力します。

```
mqiptPW -sf encryption_key_file
```

ここで、*encryption_key_file* は、ご使用の MQIPT インストールのパスワード暗号鍵を含むファイルの名前です。 MQIPT インストール済み環境でデフォルトのパスワード暗号鍵を使用している場合は、**-sf** パラメーターを指定する必要はありません。 プロンプトが表示されたら、暗号化ハードウェアにアクセスするためのパスワードを入力します。

鍵ストア・パスワードの暗号化について詳しくは、1058 ページの『MQIPT での鍵リング・パスワードの暗号化』を参照してください。

8. ステップ 1074 ページの『3』 で CA 鍵リング・ファイルを作成した場合は、ステップ 1076 ページの『7』 の指示に従って、CA 鍵リング・ファイルのパスワードを暗号化します。
9. `mqipt.conf` 構成ファイルを編集します。

a) **EnableAdvancedCapabilities** グローバル・プロパティーを真に設定して、この IBM MQ Advanced 機能を使用するための適切な資格があることを確認します。

b) 1 つ以上の **SSLServerKeyRingUseCryptoHardware**、**SSLServerCAKeyRingUseCryptoHardware**、**SSLServerKeyRingUseCryptoHardware**、または **SSLServerKeyRingUseCryptoHardware** プロパティーを真に設定して、経路上で暗号ハードウェア鍵ストアを使用できるようにします。

経路で暗号ハードウェアを使用できるようにするプロパティーについて詳しくは、MQIPT 経路プロパティーを参照してください。

IBM MQ 9.2.0 以降、**SSLCommandPortKeyRingUseCryptoHardware** プロパティーを真に設定することにより、TLS コマンド・ポートで暗号化ハードウェアを使用することもできます。

c) CA 証明書に鍵リング・ファイルを使用する場合は、1 つ以上の **SSLServerCAKeyRing** または **SSLServerCAKeyRing** プロパティーを設定して、CA 鍵リングの場所を指定します。

サイト証明書に暗号ハードウェアを使用するように経路を構成した場合に、CA 鍵リング・ファイルを指定しないと、暗号ハードウェアの鍵ストアが CA 鍵ストアとして使用されます。

d) **SSLServerKeyRingPW**、**SSLServerCAKeyRingPW**、**SSLClientKeyRingPW**、**SSLClientCAKeyRingPW**、または **SSLCommandPortKeyRingPW** プロパティーを使用して、暗号化ハードウェアおよび CA 鍵リングにアクセスするための暗号化されたパスワードを指定します。

mqiptPW コマンドによって出力された暗号化パスワードに **SSL*KeyRingPW** プロパティーの値を設定します。

e) 暗号ハードウェアに複数の個人証明書が含まれている場合は、認証のために SSL /TLS サーバーまたはクライアントに送信するために、MQIPT が選択すべき証明書を指定します。

SSL/TLS クライアント経路の 1 つ以上の **SSLClientSite*** プロパティー、または SSL/TLS サーバー経路の 1 つ以上の **SSLServerSite*** プロパティーを設定することで、選択する証明書を指定できます。

SSLCommandPortSiteLabel プロパティーを使用して証明書ラベル名を指定することにより、TLS コマンド・ポートで使用する証明書を指定できます。

鍵リングから証明書を選択する方法について詳しくは、1059 ページの『MQIPT での鍵リング・ファイルからの証明書の選択』を参照してください。 鍵リングから証明書を選択するためのプロパティーについては、MQIPT 経路プロパティーで説明しています。

例えば、TLS サーバー経路のサイト証明書には暗号ハードウェアの鍵ストアを使用し、同じ経路の CA 証明書の保管には鍵リング・ファイルを使用するには、以下のプロパティーを経路定義に追加します。

```
SSLServerKeyRingUseCryptoHardware=true
SSLServerKeyRingPW=<mqiptPW>1!gORdM4wft5d1rCgNMDEGag==!dZxhgQD2A8Ea0yeqawQvPg==
SSLServerCAKeyRing=/opt/mqipt/ssl/ca.pfx
SSLServerCAKeyRingPW=<mqiptPW>1!3Vdrpiu6kMwn0sWRCVgT5g==!LHltGLEg3OFvN8+02Re0YA==
SSLServerSiteLabel=mqiptsite
```

10. MQIPT の再始動。

Java security manager (MQIPT)

Java security manager を MQIPT 機能とともに使用して、より高いレベルのセキュリティーを提供することができます。

注 : Deprecated MQIPT での Java security manager の使用は、 Java の将来のリリースで削除するために Java security manager が非推奨になったため、非推奨になりました。

MQIPT は、 `java.lang.SecurityManager` クラスで定義されているデフォルトの Java security manager を使用します。 MQIPT の Java security manager 機能は、グローバル・プロパティー **SecurityManager** を使用して有効または無効にすることができます。 詳しくは、MQIPT のグローバル・プロパティーを参照してください。

Java security manager は、次の 2 つのデフォルト・ポリシー・ファイルを使用します。

- *$MQIPT_PATH*/java/jre/lib/security/java.policy (ここで、*$MQIPT_PATH* は MQIPT のインストール先ディレクトリーです) という名前のグローバル・システム・ポリシー・ファイルは、ホスト上の仮想マシンのすべてのインスタンスで使用されます。
- .java.policy というユーザー固有のポリシー・ファイルは、ユーザーのホーム・ディレクトリーにあります。

追加の MQIPT ポリシー・ファイルを使用することもできます。 前述のデフォルト・ポリシー・ファイルではなく、MQIPT ポリシー・ファイルを使用する必要があります。 詳しくは、 MQIPT グローバル・プロパティー の **SecurityManagerPolicy** を参照してください。

ポリシー・ファイルの構文は非常に複雑です。それをテキスト・エディターを使用して変更できますが、通常は、Java で提供されるポリシー・ツール・ユーティリティーを使用して変更するほうが簡単です。 ポリシー・ツール・ユーティリティーは、*$MQIPT_PATH*/java/jre/bin ディレクトリーにあり、Java の資料内で完全に文書化されています。

を実行する MQIPT ために定する必要のある権限を示すために、サンプルのポリシーファイル (mqiptSample.policy)が MQIPT 提供されています。

サンプル・ポリシー・ファイルはご使用の構成に合わせて編集する必要があります。 特に、 mqipt.conf 構成ファイルを含む MQIPT ホーム・ディレクトリーは、 MQIPT インストール・ディレクトリーとは異なる場合があるため、セキュリティー・ポリシーで **FilePermission** 項目を構成する際には、正しいディレクトリーを指定するように注意してください。

以下のエントリーを変更する必要があります。

- errors ディレクトリーに対する読み取りおよび書き込み権限を付与する **java.io.FilePermission** エントリー。 このエントリーのファイル・パスは MQIPT ホーム・ディレクトリーを指していなければなりません。ここに、errors ディレクトリーが配置されるためです。 MQIPT は、 FFST Failure Data Capture ファイル (AMQ*.FDC) およびトレース・ファイル (AMQ*.TRC*) を errors ディレクトリーに作成します。 MQIPT が errors ディレクトリーにトレース・ファイルおよび FFST ファイルを作成する許可を持っていることを確認し、トラブルシューティングが可能であることを確認する必要があります。
- logs ディレクトリーに対する読み取りおよび書き込み権限を付与する **java.io.FilePermission** エントリー。 このエントリーのファイル・パスは MQIPT ホーム・ディレクトリーを指していなければなりません。ここに、logs ディレクトリーが配置されるためです。 **ConnectionLog** グローバル・プロパティーが有効になっている場合、 MQIPT は接続ログ・ファイル (mqipt*.log) を logs ディレクトリーに作成します。
- MQIPT インストール・ディレクトリー内のすべてのディレクトリー (bin、 exits、 lib、および ssl ディレクトリーなど) に対するアクセス権限を read および execute に付与する **java.io.FilePermission** エントリー。 これらのエントリーのファイル・パスは、MQIPT インストール・ディレクトリーを指すように変更する必要があります。 これらのエントリーの一部は、不要な場合は省略できます。
- **java.net.SocketPermission** エントリーは、各 listen MQIPT 経路への接続を制御するために変更する必要があります。 各 MQIPT 経路のリスナー・ポートとリスナー・アドレスには、 listen 許可と accept 許可が必要です。
- **java.net.SocketPermission** エントリーは、各 MQIPT 経路外の接続を制御するために変更する必要があります。 MQIPT 経路の接続先の経路宛先、プロキシー・サーバー、または LDAP サーバーには、connect 許可が必要です。 IP アドレスの代わりにホスト名を使用して宛先を指定する場合は、解決権限が必要です。

MQIPT が証明書終了またはセキュリティ終了を使うように設定されている場合は、以下のエントリも追加する必要があります：

- **java.io.FilePermission** エントリで、read または execute アクセスを MQIPT 終了クラスに許可します。 出口で必要な場合は、追加権限の付与が必要になることがあります。
- createClassLoader 権限を与える **java.lang.RuntimePermission** エントリです。 このエントリーは、MQIPT に終了クラスをロードする許可を与えます。

- V 9.3.1 closeClassLoader 権限を与えるための **java.lang.RuntimePermission** エントリです。

ご使用の構成によっては、以下のエントリーの追加も必要な場合があります。

- mqipt.conf 構成ファイル、または mqipt.conf を含む MQIPT ホーム・ディレクトリーに対する読み取り権限を付与するための **java.io.FilePermission** エントリー。
- セキュリティー・ポリシー・ファイル自体に対する読み取り権限を付与する **java.io.FilePermission** エントリー。 これは、MQIPT リフレッシュによりセキュリティー・ポリシー・ファイルが再度読み取られる場合に役立ちます。
- **java.io.FilePermission** エントリで、SSL/TLS キーリング・ファイルとキーリング・パスワード・ファイルへの読み取りアクセスを許可します。 これは、**SSLClient** または **SSLServer** プロパティーが使用可能になっている経路を使用する場合、または TLS コマンド・ポートが構成されている場合にのみ必要です。

注 : Windows **java.io.FilePermission** エントリーでは、パスに 2 つの円記号 (¥¥) を使用する必要があります。 1 つの円記号はエスケープ文字として使用されるためです。

サンプル・ファイルは、MQIPT が C:\Program Files\IBM\MQ Internet Pass-Thru の Windows システムにインストールされていることを前提としています。 また、MQIPT のホーム・ディレクトリー (mqipt.conf ファイルの場所)が MQIPT のインストール・ディレクトリーと同じであることを前提としています。

MQIPT を別の場所にインストールした場合は、 MQIPT インストール・ディレクトリーを参照するように **codeBase** 定義内のディレクトリーを変更する必要があります。 正しい接頭部 (file:/) と正しいファイル接尾部 (/lib/com.ibm.mq.ipt.jar) を含めるように注意してください。 AIX and Linux システムでは、 MQIPT が /opt/mqipt にインストールされていると想定すると、標準的な **codeBase** URL は file:/opt/mqipt/lib/com.ibm.mq.ipt.jar になります。

権限は通常、3 つの属性で定義されます。 ソケット接続を制御する場合の値は次のとおりです。

クラス権限
java.net.SocketPermission

制御する名前
これは hostname:port という形式で構成されます。この場合、名前の各コンポーネントはワイルドカードで指定できます。 ホスト名はドメイン名または IP アドレスにすることができます。 ホスト名の左端の位置は、アスタリスク(*)で指定することができます。例えば、harry.company1.com は、以下のストリングのそれぞれによって突き合わせされます:

- harry
- harry.company1.com
- *.company1.com
- *
- 198.51.100.123 (これが harry.company1.com の IP アドレスであることを前提とした場合)

名前のポート・コンポーネントは、単一ポート・アドレスまたはポート・アドレスの範囲として指定できます。以下に例を示します。

1414
ポート 1414 のみ

1414-
1414 以上のすべてのポート・アドレス

-1414
1414 以下のすべてのポート・アドレス

1-1414
1 と 1414 の間 (両端を含む) のすべてのポート・アドレス

許可されるアクション
java.net.SocketPermission で使用されるアクションは次のとおりです。

accept
指定されたターゲットからの接続の受け入れを許可します

connect
指定されたターゲットへの接続を許可します

listen
アプリケーションによる指定されたポート (複数可) での接続要求の listen を許可します

解決
ドメイン・ネームを IP アドレスに解決するために DNS の使用を許可します

Java security manager の制御は、 java.security.manager システム・プロパティーと java.security.policy Java システム・プロパティーを使用して行うこともできますが、 MQIPT の制御には **SecurityManager** プロパティーと **SecurityManagerPolicy** プロパティーを使用することをお勧めします。

トレース・レコードおよび FFST レコードに診断情報を組み込むには、MQIPT は特定の MQIPT システム・プロパティーおよび環境変数にアクセスする必要があります。 Java セキュリティー・ポリシーには以下のプロパティーを常に含める必要があります。

```
permission java.util.PropertyPermission "java.home", "read";
permission java.util.PropertyPermission "java.version", "read";
permission java.util.PropertyPermission "java.runtime.version", "read";
permission java.util.PropertyPermission "java.vm.info", "read";
permission java.util.PropertyPermission "java.vm.vendor", "read";
permission java.util.PropertyPermission "os.arch", "read";
permission java.util.PropertyPermission "os.name", "read";
permission java.util.PropertyPermission "os.version", "read";
permission java.lang.RuntimePermission "getenv.MQIPT_PATH";
permission java.lang.RuntimePermission "getStackTrace";
permission javax.management.MBeanServerPermission "createMBeanServer";
permission javax.management.MBeanPermission "com.ibm.mq.ipt.IPTManager#-
[com.ibm.mq.ipt:type=IPTManager]", "registerMBean";
permission javax.management.MBeanPermission "com.ibm.mq.ipt.IPTManager#-
[com.ibm.mq.ipt:type=IPTManager]", "unregisterMBean";
permission javax.management.MBeanTrustPermission "register";
```

これらのプロパティーをすべて含めないと、MQIPT が正しく動作しなくなり、問題診断が正しく行われなくなります。

MQIPT のセキュリティー出口

Destination 経路プロパティーで定義されている、ターゲット宛先へのアクセスを制御する場合は、セキュリティー出口を使用します。 セキュリティー出口は、MQIPT が、ターゲット宛先に接続する前に、クライアントから接続要求を受信するポイントで呼び出されます。

初期接続プロパティーに基づいて、セキュリティー出口は接続の完了を許可するかどうかを決定します。

経路の開始時に、セキュリティー出口が呼び出され、接続要求を処理するための初期化と自身の準備が行われます。 ユーザー・データをロードして、このデータに迅速かつ容易にアクセスできるように準備することで接続要求の処理に要する時間を最小限に抑えるために、初期化プロセスを使用する必要があります。

各経路は独自のセキュリティー出口を持つことができます。

- **SecurityExit** プロパティーは、ユーザー定義のセキュリティー出口を使用可能/使用不可にするために使用されます。
- **SecurityExitName** プロパティーは、ユーザー定義のセキュリティー出口のクラス名を定義するために使用されます。
- **SecurityExitPath** プロパティーは、クラス・ファイルを含むディレクトリー名を定義するために使用されます。 このプロパティーが設定されていない場合は、exits サブディレクトリーにクラス・ファイルがあると想定されます。 **SecurityExitPath** では、ユーザー定義のセキュリティー出口を含む JAR ファイルの名前を定義することもできます。

- **SecurityExitTimeout** プロパティーは、接続要求の検証時にセキュリティー出口からの応答を待機する時間を決定するために MQIPT によって使用されます。

セキュリティー出口プロパティーについて詳しくは、MQIPT 経路のプロパティーを参照してください。

MQIPT は SecurityExit クラスを使用して、ユーザー定義のセキュリティー出口を呼び出します。 このクラスはユーザー定義のセキュリティー出口により拡張される必要があり、必要な機能を提供するためにそのメソッドの大部分を指定変更する必要があります。 SecurityExitResponse オブジェクトは MQIPT にデータを戻すために使用され、このデータは接続要求を受け入れるか拒否するかを決定するために MQIPT によって使用されます。 SecurityExitResponse オブジェクトには、セキュリティー出口プロパティーで定義された経路を指定変更するために使用される、新しい宛先と宛先ポート・アドレスを含めることもできます。

セキュリティー出口の実装方法を示す 3 つのサンプル・セキュリティー出口が用意されています。

- SampleSecurityExit は、IBM MQ チャネルの名前に基づいて IBM MQ キュー・マネージャーへのアクセスを制御する方法を示します。 これは、「MQIPT」というストリングで始まるチャネル名の接続のみを許可します。 詳しくは、セキュリティー出口の使用を参照してください。
- SampleRoutingExit は、定義されている IBM MQ サーバー (各サーバーが同じ名前と同じ属性のキュー・マネージャーをホストする) のプールへのクライアント接続要求の動的ルーティングを許可します。 サンプルには、サーバー名のリストを含む構成ファイルが含まれています。 詳しくは、セキュリティー出口を使用した IBM MQ キュー・マネージャー・サーバーへのクライアント接続要求のルーティングを参照してください。
- SampleOneRouteExit は、接続要求で使用される IBM MQ チャネル名から派生する IBM MQ キュー・マネージャーへの動的ルーティングを許可します。 サンプルには、キュー・マネージャー名からサーバー名へのマップを含む構成ファイルが含まれています。 詳しくは、クライアント接続要求の動的ルーティングを参照してください。

注 : MQIPT は単一の JVM で実行されるため、ユーザー定義のセキュリティー出口は、以下のいずれかの方法で MQIPT の通常の操作に悪影響を与える可能性があります。

- システム・リソースへの影響
- ボトルネックの発生
- パフォーマンスの低下

セキュリティー出口を実稼働環境に実装する前に、その効果を詳細にテストする必要があります。

MQIPT の com.ibm.mq.ipt.exit.SecurityExit クラス

このクラスとその public メソッドをユーザー定義のセキュリティー出口で拡張して、いくつかの共通データへのアクセス権を取得し、いくつかの MQIPT の初期化の実行を許可する必要があります。

各メソッドが MQIPT によって呼び出される前に、使用するメソッドのいくつかのプロパティーを使用できます。 これらの値は、このクラスに定義されている適切な get メソッドを使用して取得できます。

方法

public int init(IPTTrace)

使用可能なプロパティーは以下のとおりです。

- リスナー・ポート
- destination
- 宛先ポート
- バージョン

init メソッドは、経路の開始時に MQIPT によって呼び出されます。 このメソッドからの戻り時に、セキュリティー出口は接続要求の検証準備ができていなければなりません。 有効な戻りコードは ExitRc.OK または ExitRc.INIT_ERROR です。

public int refresh(IPTTrace)
使用可能なプロパティーは以下のとおりです。

- リスナー・ポート
- destination
- 宛先ポート

`refresh` メソッドは、MQIPT 構成がリフレッシュされるときに、MQIPT によって呼び出されます。 このアクションは、通常、プロパティーが構成ファイルで変更された場合に実行されます。 MQIPT は、すべてのプロパティーを構成ファイルから再ロードし、どのプロパティーが変更されたか、および経路を再始動する必要があるかどうかを判別します。

このメソッドでは、使用する外部データ (つまり、**init** メソッドによってロードされるデータ) の再ロードを実行する必要があります。 有効な戻りコードは `ExitRc.OK` または `ExitRc.REFRESH_ERROR` です。

public void close(IPTTrace)
使用可能なプロパティーは以下のとおりです。

- リスナー・ポート
- destination
- 宛先ポート

`close` メソッドは、停止時に MQIPT によって呼び出されます。 このメソッドは、出口がその操作中に取得したシステム・リソースを解放します。 MQIPT は、シャットダウンする前にこのメソッドが完了するまで待機します。

このメソッドは、セキュリティー出口が以前に使用可能にされたが、構成ファイルでは現在使用不可にされている場合にも呼び出されます。

public SecurityExitResponse validate(IPTTrace)
使用可能なプロパティーは以下のとおりです。

- リスナー・ポート
- destination
- 宛先ポート
- timeout
- クライアント IP アドレス
- クライアント・ポート・アドレス
- チャネル名
- キュー・マネージャー名

`validate` メソッドは、検証する接続要求の受信時に MQIPT によって呼び出されます。
SSLProxyMode プロパティーが使用可能な場合、チャネル名とキュー・マネージャー名は使用できません。この機能は TLS データのトンネリングにのみ使用されるため、通常は初期データ・フローから取得されるデータは読み取れません。

セキュリティー出口は、以下の情報を含む、`SecurityExitResponse` オブジェクトを返す必要があります。

- 理由コード (要設定)
- 新規宛先アドレス (オプション)
- 新規宛先リスナー・ポート・アドレス (オプション)
- メッセージ (オプション)

理由コードは、接続が MQIPT によって受け入れられるか、拒否されるかを判別します。 `newDestination` および `newDestinationPort` フィールドをオプションで設定して、新規ターゲット・キュー・マネージャーを定義することができます。 これらのプロパティーを設定しない場合は、構成ファイルに定義されて

いる経路の **Destination** および **DestinationPort** プロパティーが使用されます。 接続ログ・ファイル項目には、返されたメッセージが付加されます。

MQIPT 構成プロパティーの値を取得するために、以下のメソッドがサポートされています。

public int getListenerPort()
ListenerPort プロパティーに定義されている、経路リスナー・ポートを取得します。

public String getDestination()
Destination プロパティーに定義されている、宛先アドレスを取得します。

public int getDestinationPort()
DestinationPort プロパティーに定義されている、宛先リスナー・ポートのアドレスを取得します。

public String getClientIPAddress()
接続要求を出したクライアントの IP アドレスを取得します。

public int getClientPortAddress()
接続要求を出したクライアントによって使用されるポート・アドレスを取得します。

public int getTimeout()
タイムアウト値を取得します。 MQIPT は、**SecurityExitTimeout** プロパティーでの定義に従って、セキュリティー出口が要求を検証するまで待機します。

public int getConnThreadID()
デバッグに役立つ、接続要求を処理する接続スレッド ID を取得します。

public String getChannelName()
接続要求で使用される IBM MQ チャネル名を取得します。

public String getQMName()
接続要求で使用される IBM MQ キュー・マネージャー名を取得します。

public boolean getTimedout()
タイムアウトが期限切れになったかどうかを判断するためにセキュリティー出口で使用できます。

MQIPT の com.ibm.mq.ipt.exit.SecurityExitResponse クラス

このクラスは、ユーザー定義のセキュリティー出口から MQIPT に応答を戻すために使用され、接続要求を受け入れるか拒否するかを判断するために使用されます。

このタイプのオブジェクトは validate メソッドでのみ作成されます (1081 ページの『MQIPT の com.ibm.mq.ipt.exit.SecurityExit クラス』を参照)。 これらのオブジェクトを作成するための便利なコンストラクターがあり、プロパティーごとにメソッドがあります。 詳細については、サンプルのセキュリティー出口を参照してください。

デフォルトの SecurityExitResponse オブジェクトを作成すると、接続要求は拒否されます。

コンストラクター

- **public SecurityExitResponse (String dest, int destPort, int rc, String msg)**

  ここで、

  - `dest` は新規ターゲット宛先です。
  - `destPort` は新規宛先ポート・アドレスです。
  - `rc` は理由コードです。
  - `msg` は、接続ログ項目に追加されるメッセージです。
- **public SecurityExitResponse (String dest, int destPort, int rc)**
- **public SecurityExitResponse (int rc, String msg)**
- **public SecurityExitResponse (int rc)**

方法

public void setDestination(String dest)
接続要求の新規宛先アドレスを設定します。

public void setDestinationPort(int port) throws IPTException
接続要求の新規宛先リスナー・ポート・アドレスを設定します。無効なポート・アドレスの IPTException をスローします。

public void setMessage(String msg)
接続ログ・レコードにメッセージを追加します。

public void setReasonCode(int rc)
接続要求の理由コードを設定します。

MQIPT のセキュリティー出口戻りコード

いくつかの異なる状況でセキュリティー出口を呼び出すときに MQIPT が認識する戻りコード。

以下に示す状況でセキュリティー出口を呼び出すと、MQIPT によって 以下の戻りコードが認識されます。

戻りコード	説明	init	Validate	リフレッシュ
ExitRc.OK	要求が正常に完了しました。	yes	yes	yes
ExitRc.INIT_ERROR	初期化要求が失敗し、経路が使用不可になります。	yes		
ExitRc.REFRESH_ERROR	リフレッシュ要求が失敗しました。			yes
ExitRc.NOT_AUTHORIZED	検証プロセスが失敗し、接続要求が拒否されました。		yes	
ExitRc.DISABLE_SSL	検証要求が成功し、ターゲットへの接続で SSL/TLS は使用されません。		yes	

MQIPT のポート番号制御

MQIPT を使用することで、発信接続時に使用されるローカル・ポート番号の範囲を制限できます。

初期ローカル・ポート番号を指定するには、経路に **OutgoingPort** プロパティーを設定し、使用されるポートの数を指定するには、**MaxConnectionThreads** を設定します。 例えば、**OutgoingPort** を 1600 に、**MaxConnectionThreads** を 20 に設定した場合、その経路のローカル・ポート番号の範囲は 1600 から 1619 になります。

経路間でポート番号が競合していないことを確認するのは、MQIPT 管理者の責任です。

OutgoingPort が定義されていない場合、デフォルト値の 0 は、システムに割り振られたポート番号が各接続で使用されることを意味します。

HTTP を使用すると、発信ポートの数は HTTP を使用していないときの 2 倍になります。 前述の例では、経路で HTTP を使用した場合、数の範囲は 1600 から 1639 になります。

詳しくは、ポート番号の割り振りを参照してください。

マルチホーム・システム

マルチホーム・システムの使用時に、**LocalAddress** プロパティーを使用して発信接続がバインドされる IP アドレスを指定することができます。 このプロパティーではホスト名はサポートされません。

MQIPT に保管されているパスワードの暗号化

MQIPT 構成には、さまざまなリソースにアクセスするためのパスワードと、コマンド・ポートを使用して MQIPT にアクセスするためのパスワードが含まれている場合があります。 IBM MQ 9.2.0 からは、これらのパスワードはすべて暗号化されることによって保護されます。

このタスクについて

IBM MQ 9.2.0 より前のバージョンでは、 MQIPT が鍵リング、または暗号ハードウェア鍵ストアにアクセスするために使用するパスワードのみを暗号化できます。 暗号化されたパスワードは、いずれかの **SSL*KeyRingPW** プロパティーで参照されるファイルに保管されます。 LDAP サーバー用の他のパスワードおよび MQIPT アクセス・パスワードは、`mqipt.conf` 構成ファイル内にプレーン・テキストで保管されます。

IBM MQ 9.2.0 からは、MQIPT で使用するために保管されているすべてのパスワードは、**mqiptPW** コマンドによってパスワードを暗号化して保護する必要があります。 暗号化されたパスワードは、`mqipt.conf` 構成ファイル内にプロパティー値として保管されます。 MQIPT は、暗号化されたパスワード、プレーン・テキスト・パスワード、およびプロパティー値のファイル名を区別できます。 MQIPT で使用するために保管されているすべてのパスワードは、最も安全な保護方式として暗号化する必要があります。

Deprecated IBM MQ 9.2.0 より前の MQIPT で使用されていた鍵ストア・パスワードの暗号化方式は非推奨になりましたが、IBM MQ 9.2.0 より前のバージョンで使用可能であった構成プロパティーには引き続き使用できます。 鍵リング・パスワードの保護を向上させるには、最新の保護方式を使用して、以前に暗号化された鍵リング・パスワードを再暗号化します。

注: `mqipt.conf` 構成ファイルの **SSLCommandPortKeyRingPW** プロパティー、および **mqiptAdmin** プロパティー・ファイルの **SSLClientCAKeyRingPW** プロパティーは、パスワード・ファイルを参照できません。 これらのプロパティーの値は、**mqiptPW** コマンドによって出力された暗号化パスワード・ストリングに設定する必要があります。

MQIPT 構成にプレーン・テキストまたは脆弱な保護パスワードがある場合は、MQIPT の開始時または経路の開始時に警告メッセージが表示されます。

最新の保護方式を使用して MQIPT で使用するために保管されるパスワードを暗号化するには、以下の手順を使用します。 IBM MQ 9.2.0 より前の MQIPT で鍵リング・パスワードを暗号化するには、1086 ページの『IBM MQ 9.2.0 での MQIPT 前の鍵リング・パスワードの暗号化』の手順に従ってください。

手順

1. オプション: パスワード暗号鍵を含むファイルがまだない場合は、このファイルを作成します。

   MQIPT は暗号鍵を使用してパスワードを暗号化します。 ファイルに独自の暗号鍵を指定できます。 このファイルには少なくとも 1 つの文字と 1 行のみのテキストが含まれている必要があります。

   MQIPT のインスタンスのために保管されるすべてのパスワードの暗号化と復号に、同じパスワード暗号鍵が使用されます。 そのため、必要なパスワード暗号鍵ファイルは、MQIPT インストール環境ごとに 1 つだけです。

   MQIPT 構成でパスワードの暗号化に使用した暗号鍵とは異なるパスワード暗号鍵を使用して、**mqiptAdmin** プロパティー・ファイルに保管されているパスワードを暗号化することができます。

   MQIPT を自動的に開始されるサービスとして実行することを計画している場合は、デフォルト名の `mqipt_cred.key` を使用してパスワード暗号鍵ファイルを作成し、それを MQIPT ホーム・ディレクトリーに配置する必要があります。

   パスワード暗号鍵の指定は必須ではありませんが、指定するとセキュリティーが向上します。 独自の暗号鍵を指定しない場合は、デフォルトの暗号鍵が使用されます。

   注: 必ず、無許可のユーザーが暗号鍵を読み取れないように、パスワード暗号鍵のファイルに適切なファイル許可を設定する必要があります。 パスワード暗号鍵の読み取り権限を必要とするのは、**mqiptPW** コマンドを実行するユーザーと、 MQIPT を実行するユーザーだけです。
2. **mqiptPW** コマンドを使用してパスワードを暗号化します。

mqiptPW コマンドの構文は、mqiptPW (encrypt stored password) で説明されています。

ステップ 1085 ページの『1』でパスワード暗号鍵ファイルを作成した場合は、 **-sf** パラメーターを使用して **mqiptPW** にファイル名を指定します。 例えば、**-sf** パラメーターで指定されたファイル内の暗号鍵を使用してパスワードを暗号化するには、以下のコマンドを発行します。

```
mqiptPW -sf /opt/mqipt/mqipt_password.key
```

3. プロンプトが表示されたら、暗号化するパスワードを入力します。
   暗号化されたパスワードは、**mqiptPW** によって出力されます。
4. mqipt.conf 構成ファイル、または **mqiptAdmin** プロパティー・ファイルの適切なプロパティーに暗号化されたパスワードをコピーします。
   例えば、以下の行は、MQIPT アクセス・パスワードの暗号化パスワードを指定します。

```
AccessPW=<mqiptPW>1!QL+2Jvj/tigKKlD7Nz8Oqw==!AMDBefOUrmPf5i1OuqV5MA==
```

5. MQIPT を開始します。 デフォルト名以外の名前を持つパスワード暗号鍵ファイルを手順 1085 ページの『1』 で作成した場合は、MQIPT の開始時に暗号鍵ファイルの名前を指定します。
   MQIPT の始動時に **-sf** パラメーターを使用して、パスワード暗号鍵ファイルの名前を指定できます。 例えば、**-sf** パラメーターで指定されたファイル内の暗号鍵を使用して MQIPT を開始するには、以下のコマンドを発行します。

```
mqipt /opt/mqipt -sf /opt/mqipt/mqipt_password.key
```

   MQIPT の開始時にパスワード暗号鍵ファイル名を指定するその他の方式については、Specifying the password encryption key を参照してください。

   mqiptAdmin プロパティー・ファイルの **PasswordProtectionKeyFile** プロパティーを使用して、**mqiptAdmin** コマンドのパスワード暗号鍵ファイルの名前を指定できます。

IBM MQ 9.2.0 での MQIPT 前の鍵リング・パスワードの暗号化

IBM MQ 9.2.0 より前では、 MQIPT によって使用される鍵リングへのアクセスに使用される暗号化されたパスワードは、ファイルに保管されます。

このタスクについて

このタスクの手順に従って、 IBM MQ 9.2.0 より前に MQIPT が使用する鍵リング・パスワードを暗号化します。 IBM MQ 9.2.0 for Long Term Support の MQIPT 以降では、 1085 ページの『MQIPT に保管されているパスワードの暗号化』で説明されている、よりセキュアな保護方式を使用してください。

手順

1. **mqiptPW** コマンドを使用して鍵リング・パスワードを暗号化します。
   以下のコマンドを入力して、パスワードを暗号化します。

```
mqiptPW password filename
```

   説明:

   パスワード
   鍵リングにアクセスするために必要な平文パスワードです。

   filename
   作成されるパスワード・ファイルの名前です。

   mqiptPW コマンドの構文は、mqiptPW (encrypt stored password) で説明されています。
2. 手順 1086 ページの『1』 で作成された暗号化パスワードを含むファイルの名前に適切な経路プロパティーを設定します。

例えば、MQIPTTLS サーバー証明書が含まれているキー・リングのパスワード・ファイルを指定するには、mqipt.conf 構成ファイルに以下の行を追加します:

```
SSLServerKeyRingPW=filename
```

MQIPT のその他のセキュリティーに関する考慮事項

MQIPT には、設計者が安全なソリューションを構築する際に役立つ追加機能がいくつかあります。

- 内部ネットワーク内の多くのクライアントが発信接続を試行している場合、これらのクライアントはすべて、ファイアウォールの内側にある MQIPT を通過できます。 そのため、ファイアウォール管理者は MQIPT コンピューターのみへの外部アクセス権を付与する必要があります。
- MQIPT は、構成ファイルに明示的に構成されているキュー・マネージャーにのみ接続できます。ただし、MQIPT が SOCKS プロキシーとして機能しているか、セキュリティー出口を使用している場合を除きます。
- MQIPT は、送受信するメッセージが有効であり、IBM MQ プロトコルに準拠していることを検証します。このようにすることで、MQIPT が、IBM MQ プロトコルの外側のセキュリティー・アタックに使用されるのを防ぐことができます。 MQIPT が SSL/TLS プロキシーとして機能している場合、すべての IBM MQ データおよびプロトコルが暗号化されている場合、MQIPT は、初期 SSL/TLS ハンドシェークのみを保証することができます。 この状態では、Java security manager を使用します。
- MQIPT は、チャネル出口で独自のエンドツーエンド・セキュリティー・プロトコルを実行できるようにします。
- MaxConnectionThreads プロパティーを設定することで、着信接続の総数を制限できます。 これは、サービス妨害攻撃からぜい弱な内部キュー・マネージャーを保護するのに役立ちます。

構成ファイル

MQIPT 構成ファイル mqipt.conf は、MQIPT へのリモート管理アクセスを制御する **AccessPW** パスワードなどの機密情報が含まれている可能性があるため、無許可ユーザーによって読み取られないように保護する必要があります。 1085 ページの『MQIPT に保管されているパスワードの暗号化』の手順に従って、構成ファイルに指定するすべてのパスワードを保護してください。 また、mqipt.conf が無許可変更に対して保護されていることを確認します。 MQIPT を実行するユーザー・アカウントのみがファイルを読み取りまたは更新できるように、 mqipt.conf のオペレーティング・システム・ファイル許可を設定します。

コマンド・ポート

MQIPT コマンド・ポートは、 **mqiptAdmin** コマンドによってネットワークを介して MQIPT のリモート・インスタンスに発行された管理コマンドを受け入れます。

IBM MQ 9.2.0 以降、MQIPT には、保護されていないコマンド・ポート 1 つと、TLS で保護されたコマンド・ポート 1 つを構成することができます。 非セキュアのコマンド・ポートへの接続は暗号化されません。

注 : 保護されていないコマンド・ポートにネットワークを介して送信されるデータは、MQIPT アクセス・パスワードを含め、ネットワーク上の他のユーザーに見られる可能性があります。

保護されていないコマンド・ポートまたは TLS コマンド・ポートを使用可能にする前に、そのコマンド・ポートを使用可能にする必要があるかどうかを検討し、MQIPT のリモート管理を許可することのリスクを評価する必要があります。 IBM MQ 9.2.0 以降、 **mqiptAdmin** コマンドは、コマンド・ポートを使用せずに **mqiptAdmin** コマンドと同じユーザーの下で実行されている MQIPT のローカル・インスタンスを管理できます。 したがって、MQIPT のローカル・インスタンスを管理するためにコマンド・ポートを使用可能にする必要がない場合もあります。

保護されていないコマンド・ポートまたは TLS コマンド・ポートを使用可能にする場合は、そのコマンド・ポートに対する無許可アクセスを防ぐ必要があります。 例えば、コマンド・ポートへのアクセスを保護する場合、以下の点を考慮する必要があります。

- ファイアウォールを使用して、MQIPT コマンド・ポートに接続できる一連のコンピューターを制限すること。

- **AccessPW** プロパティーおよび **RemoteCommandAuthentication** プロパティーを使用して、コマンド・ポートでの認証を有効にすること。 コマンド・ポート認証を使用可能にすることについて詳しくは、コマンド・ポート認証を参照してください。
- **RemoteShutdown** プロパティーによるリモート・シャットダウンを無効にすることを検討してください。
- **CommandPortListenerAddress** プロパティーおよび **SSLCommandPortListenerAddress** プロパティーを使用して、特定のネットワーク・インターフェースで listen するようにコマンド・ポートを構成することを考慮してください。

mqiptAdmin コマンドを使用した MQIPT の管理について詳しくは、コマンド・ラインによる MQIPT の管理を参照してください。

MQIPT の接続ログ

MQIPT では接続ログ機能が提供され、ここにすべての成功接続および失敗接続の試行のリストが含まれます。

エントリーは、MQIPT 経路で受信または作成された各接続に対して、また MQIPT によって受信された各管理コマンドに対して、接続ログに書き込まれます。 接続ログは、**ConnectionLog** および **MaxLogFileSize** プロパティーを使用して制御されます。 詳しくは、MQIPT のグローバル・プロパティーを参照してください。

MQIPT が開始されると、そのたびに新しい接続ログが作成されます。 識別できるように、ファイル名には以下のように現在のタイム・スタンプが含まれます。

```
mqiptYYYYMMDDHHmmSS.log
```

説明:

YYYY は年
MM は月
DD は日
HH は時
mm は分
SS は秒

接続ログが **MaxLogFileSize** プロパティーで決められた最大サイズに達すると、バックアップ・ファイル mqipt001.log が作成されます。 バックアップ・ファイルが最大 2 つまで保持されます(mqipt001.log および mqipt002.log)。

接続ログ内のエントリーは、接続要求の各部分を表しています。 MQIPT により受信された接続要求、およびその結果として MQIPT が宛先アドレスに対して確立する新規接続が 2 つのログ・エントリーとして表示され、さらに各接続の終了時に 2 つのエントリーが続きます。

成功した接続要求の接続ログを以下に示します:

```
Wed May 15 13:13:51 BST 2013 conn accept 127.0.0.1(3842) 127.0.0.1(5000) OK 5000-0
Wed May 15 13:13:51 BST 2013 conn conn   127.0.0.1(3843) localhost(3500) OK 5000-0
Wed May 15 13:13:52 BST 2013 conn close  127.0.0.1(3842) 127.0.0.1(5000) OK 5000-0
Wed May 15 13:13:52 BST 2013 conn close  127.0.0.1(3843) localhost(3500) OK 5000-0
```

失敗した接続要求の接続ログを以下に示します:

```
Wed May 15 14:56:40 BST 2013 conn accept 127.0.0.1(4138) 127.0.0.1(7000) OK    7000-0
Wed May 15 14:56:40 BST 2013 conn close  127.0.0.1(4138) 127.0.0.1(7000) ERROR 7000-0
 Unrecognized SSL handshake request '54'
```

接続ログのエントリー

それぞれの接続ログ・エントリーには、以下の情報が含まれます。

- エントリーが作成された時刻。
- 項目のタイプ。 値は、次の値のうちのいずれかです。

  admin
  管理コマンド

  conn
  経路接続
- 発生したイベント。 値は、次の値のうちのいずれかです。

  accept
  接続要求を受信しました

  close
  接続がクローズされました

  conn
  経路宛先への接続要求

  dspipt
  MQIPT 表示コマンドを受信しました

  nodata
  呼び出し元から受け取ったデータはありません

  ping
  ping 要求を受信しました

  状況
  状況表示コマンドを受信しました

  refr
  リフレッシュ・コマンドを受信しました

  stop
  停止コマンドを受信しました
- 送信元のネットワーク・アドレスとポート番号。 コマンド・ポートを使用せずにローカルで発行された管理コマンドについて、値 LOCAL が表示されます。
- 宛先のネットワーク・アドレスとポート番号。 コマンド・ポートを使用せずにローカルで発行された管理コマンドに対して、この値は表示されません。
- 完了コード。 この値は、OK または ERROR のいずれかです。
- MQIPT スレッド ID。
- オプションのエラー・メッセージ。

コンテナーを使用した IBM MQ Internet Pass-Thru の構成

コンテナーで IBM MQ Internet Pass-Thru (MQIPT) を実行することができます。 コンテナー・イメージによって使用されるベース・イメージは、サポートされている Linux オペレーティング・システムを使用する必要があります。

手順

- MQIPT Docker イメージのサンプルは、GitHub リポジトリー mq-container にあります。 コンテナーをビルドして実行するには、IBM MQ Internet Pass-Thru on Docker の説明に従ってください。

次のタスク

docker ps コマンドを使用して実行中のコンテナーを表示できます。 Docker コンテナーで実行されている MQIPT のコンソール出力を表示するには、**docker logs ${CONTAINER_ID}** コマンドを使用します。

V 9.3.0 ストリーミング・キューの構成

ストリーミング・キュー機能を使用すると、あるキューに書き込まれた各メッセージの重複コピーを、2 番目のキューに送信できます。 ストリーミング・キューの構成は、キュー単位で行われます。

ローカル・キューおよびモデル・キューには、ストリーミング・キューに関連する以下の 2 つの新しい属性があります。

STREAMQ
これは、ストリーム・メッセージの送信先のキューの名前です。 **STREAMQ** 属性は、別のキューの名前に設定する必要があります。

他のキューにメッセージをストリーミングするためのキューの構成に関するいくつかの制限事項があります。また、ストリーム・メッセージの宛先として設定できるキューに関する制限事項もあります。メッセージ・ストリーミング制限については、ストリーム・キュー制限を参照してください。

STRMQOS
これは、ストリーム・メッセージを送信するときに使用するサービス品質です。

STRMQOS 属性には、以下のいずれかの値を設定できます。

BESTEF
ベスト・エフォート。これはデフォルト値です。

キュー・マネージャーは、各メッセージのコピーを、**STREAMQ** 属性で指定されたキューに送信しようとします。 ストリーム・メッセージの送信中に問題が発生しても、元のメッセージの送信には影響しません。

MUSTDUP
キュー・マネージャーは、各メッセージのコピーをストリーミング・キューに送信しようとします。

ストリームされたメッセージの配信に問題がある場合、元のメッセージはそのキューに配信されず、アプリケーションは該当する理由コードとともに MQCC_FAILED を受け取ります。

詳細については、ALTER キュー、DEFINE キュー、DISPLAY QUEUE MQSC コマンド、およびキューの変更、コピー、作成、応答(反応) PCF コマンドを参照してください。

メッセージごとに複数のコピーが必要な場合は、**STREAMQ** 属性を、ターゲットが IBM MQ トピックを参照する IBM MQ 別名キューの名前を参照するように構成できます。 メッセージが元のキューに書き込まれると、メッセージのコピーが、指定されたトピックにパブリッシュされます。

各サブスクリプションがメッセージのコピーを受信するたびに、トピック・オブジェクトに対する API サブスクリプションまたは管理サブスクリプションがあることを確認する必要があります。 サブスクライバーに送信されるメッセージは、他のパブリッシュ/サブスクライブ・メッセージと同じルールに従います。例えば、各メッセージは新規メッセージ ID を持ち、MQMD のコンテキスト・フィールドは元のメッセージのコンテキスト・フィールドとは異なります。 ストリーム・メッセージとストリーム・メッセージの間の類似点と相違点について詳しくは、ストリーミングされたメッセージを参照してください。

例

ベスト・エフォートの例

以下の例では、ローカル・キュー ORDERS.QUEUE は、ストリーム・メッセージを 2 番目のキュー ANALYTICS.QUEUE。 BESTEF サービス品質は、ストリーム・メッセージを ANALYTICS.QUEUE(例えば、ANALYTICS.QUEUE がいっぱいであるため、元のメッセージを ORDERS.QUEUE。

このような構成を使用して、ストリーム・メッセージを分析することによって受信中のオーダーに対する分析を実行しながら、元のメッセージをオーダー・キューに入れて処理することができます。 ストリーミング・キュー機能の利点は、ビジネスによって満たされている実際のオーダーに影響を与えずに、ANALYTICS.QUEUE 上のストリーム・メッセージを処理待機状態にしておくことができる点です。

```
DEFINE QLOCAL(ANALYTICS.QUEUE)

ALTER QLOCAL(ORDERS.QUEUE) STRMQOS(BESTEF) STREAMQ(ANALYTICS.QUEUE)
```

注 : この例では、 **STRMQOS** が BESTEF に設定されていますが、BESTEF がデフォルトのサービス品質であるため、この属性は **ALTER** コマンドから除外することができます。

必ず複製 (Must duplicate) の例

以下の例では、すべてのメッセージのストリーム・コピーを別のローカル・キュー AUDIT.QUEUE に書き込むようにローカル・キュー PAYMENTS.QUEUE を変更します。 支払キューに入れられるすべてのメッセージが監査キューにストリーミングされることが重要です。そのため、MUSTDUP サービス品質が使用されます。

ストリーム・メッセージをそのキューに送信するときに問題が発生すると、元のメッセージも送信されず、アプリケーションは適切な完了コードと理由コードを受け取ります。 アプリケーションは、単一のキューのみが関係している場合と同じ方法で、書き込みを再試行する必要があります。

```
DEFINE QLOCAL(AUDIT.QUEUE)

ALTER QLOCAL(PAYMENTS.QUEUE) STRMQOS(MUSTDUP) STREAMQ(AUDIT.QUEUE)
```

注 :

1. 元のキューを変更するときに、ストリーミング・キューが存在している必要はありません。 ただし、MUSTDUP のサービス品質を使用しているため、ストリーミング・キューを定義するまでは、元のキューにメッセージを書き込もうとしても失敗することに注意してください。
2. トピック・オブジェクトのターゲットを持つキュー別名を使用するときに、サブスクライバーが存在しない場合は、ストリーム・メッセージの送信は引き続き成功と見なされ、元のメッセージはそのキューに送信されます。
3. ストリーム・メッセージをそのキューに送信できない場合、キュー・マネージャーはそのメッセージを送達不能キューに送信しようとはしません。 ただし、ストリーム・メッセージがリモート・キューに送信される場合、メッセージがチャネル経由で別のキュー・マネージャーに移動すると、既存の送達不能ルールに従ってそのメッセージを送達不能キューに送信できます。

ストリーミング・キューの構成

ストリーミング・キューに対して追加の構成を実行する必要はありません。 ストリーミング・キューという名前が付けられたすべてのキューからメッセージが受信されます。 ただし、ストリーミング・キューに構成されている属性値を考慮することが推奨される場合があります。

例えば、元のキューの最大深度が 100,000 で、ストリーミング・キューの最大深度が 5000 しかない場合、STRMQOS が BESTEF に設定されていると、ストリーム・メッセージが失われることがあります。また、STRMQOS が MUSTDUP に設定されている場合、元のキューにスペースが十分に残されている場合でも、MQRC_Q_FULL エラーにより書き込み時に失敗することがあります。

元のキューの構成方法に基づいて、適切な値を設定するための変更が必要になる可能性があるストリーミング・キューの属性を検討してください。

関連概念

ストリーミング・キュー

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

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

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

〒 103-8510
東京都中央区日本橋箱崎町 19 番 21 号
日本アイ・ビー・エム株式会社
日本アイ・ビー・エム株式会社
法務・知的財産
U.S.A.

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

Intellectual Property Licensing
Legal and Intellectual Property Law
〒 103-8510
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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

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

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

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

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

東京都中央区日本橋箱崎町 19 番 21 号
日本アイ・ビー・エム株式会社
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

© Copyright IBM Corp. 2007 年, 2025

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

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

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

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

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

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

著作権使用許諾:

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

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

プログラミング・インターフェース情報

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

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

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

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

商標

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

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

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

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

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

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

部品番号:

(1P) P/N: