

7.5

IBM WebSphere MQ の概要

IBM

注記

本書および本書で紹介する製品をご使用になる前に、[227 ページの『特記事項』](#)に記載されている情報をお読みください。

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

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

© Copyright International Business Machines Corporation 2007 年, 2024.

目次

IBM WebSphere MQ 製品情報.....	5
IBM WebSphere MQ の概要.....	6
IBM WebSphere MQ ライセンス情報.....	9
IBM WebSphere MQ Telemetry の概要.....	11
IBM WebSphere MQ Version 7.5 の情報ロードマップ.....	14
IBM Documentation オフラインアプリケーションの IBM WebSphere MQ Version 7.5.....	17
IBM WebSphere MQ 7.5 PDF 資料.....	17
IBM WebSphere MQ バージョン 7.5 の新機能.....	18
IBM WebSphere MQ client for HP Integrity NonStop Server.....	20
IBM WebSphere MQ バージョン 7.5 の変更点.....	24
IBM WebSphere MQ Version 7.5 フィックスパックの変更点.....	24
以前のバージョンの新機能および変更点.....	32
以前の IBM WebSphere MQ 資料と現在の資料の構造の対応.....	33
スタートアップ・ガイド (AIX).....	33
スタートアップ・ガイド (HP-UX).....	33
スタートアップ・ガイド (Linux).....	34
スタートアップ・ガイド (Solaris).....	34
スタートアップ・ガイド (Windows).....	34
アプリケーション・プログラミング・ガイド.....	35
アプリケーション・プログラミング・リファレンス.....	35
クライアント.....	35
定数.....	36
相互通信.....	36
メッセージおよびコード.....	37
マイグレーション.....	37
モニター.....	37
プログラマブル・コマンド・フォーマットおよび管理インターフェース.....	37
Publish/Subscribe ユーザーズ・ガイド.....	37
キュー・マネージャー・クラスター.....	38
スクリプト (MQSC) コマンド・リファレンス.....	38
機密保護.....	38
システム管理ガイド.....	38
.NET の使用.....	39
C++ の使用.....	39
Java の使用.....	39
Web サービス.....	39
コンポーネント・オブジェクト・モデル・インターフェースの使用.....	39
IBM WebSphere MQ Version 7.5、IBM i および z/OS.....	39
特記事項.....	39
技術概要.....	42
メッセージ・キューイングの概要.....	42
相互通信の概念.....	50
IBM WebSphere MQ Telemetry.....	68
オブジェクトの管理.....	105
IBM WebSphere MQ Multicast.....	129
機密保護.....	130
クライアントおよびサーバー.....	131
トランザクションの管理とサポート.....	138
キュー・マネージャーの機能の拡張.....	140
IBM WebSphere MQ client for HP Integrity NonStop Server の技術概要.....	141
シナリオ.....	142
IBM WebSphere MQ 7.5 の概要.....	142

スクリプトを使用する基本的なファイル転送.....	149
基本的なファイル転送の詳細.....	158
スクリプトを使用する 2 台のコンピューターでのファイル転送.....	163
2 台のコンピューターでのファイル転送の詳細.....	170
ファイル転送管理への監査機能の追加.....	178
用語集.....	181
A.....	181
B.....	184
C.....	185
D.....	190
E.....	192
F.....	193
G.....	194
H.....	195
I.....	195
J.....	197
K.....	198
L.....	198
M.....	200
N.....	204
O.....	205
P.....	207
Q.....	210
R.....	210
S.....	213
T.....	218
U.....	221
V.....	222
W.....	222
X.....	223
アクセシビリティ.....	223
Windows でのアクセシビリティ.....	224
特記事項.....	227
プログラミング・インターフェース情報.....	228
商標.....	228

IBM WebSphere MQ について

このセクションでは、IBM WebSphere MQ の使用を始めるにあたり役立つ概要情報を提供します。

関連タスク

[WebSphere MQ アーキテクチャの設計](#)

IBM WebSphere MQ

IBM WebSphere MQ Version 7.5 製品資料へようこそ。ここでは、MQ 環境を作成および保守するために実行する必要があるタスクの実行方法について詳しく説明します。また、この資料には、製品と、製品を使用してビジネス問題を解決する方法を理解するのに役立つ概念情報も記載されています。



IBM WebSphere MQ は、堅固かつセキュアで信頼性の高いメッセージング・ミドルウェアです。この製品は、メッセージおよびキューを使用して、アプリケーション、システム、サービス、およびファイル間での情報交換をサポートします。これにより、アプリケーションやビジネス・データを複数のプラットフォームでシンプルかつ迅速に統合することができます。

関連概念

[IBM WebSphere MQ 製品情報](#)

関連タスク

計画

[マイグレーションおよびアップグレード](#)

[インストール](#)

[機密保護](#)

[構成](#)

[の管理](#)

[アプリケーションの開発](#)

[モニターおよびパフォーマンス](#)

[トラブルシューティングとサポート](#)

[WebSphere MQ Managed File Transfer](#)

[WebSphere MQ エクスプローラー](#)

[MQ GUI ウィザードのヘルプ](#)

[WebSphere MQ Internet Pass-Thru](#)

[Message Service Client for .NET](#)

[WebSphere MQ Hypervisor Edition](#)

[モバイル・メッセージングおよび M2M](#)

関連資料

[参照](#)

関連情報

[IBM MQ on Cloud](#)

IBM WebSphere MQ の概要

IBM WebSphere MQ を利用すると、複数のアプリケーションが、さまざまな時点で、多種多様なコンピューティング環境で通信できるようになります。

IBM WebSphere MQ とは何ですか？

- IBM WebSphere MQ は、アプリケーション用のメッセージングです。さまざまなコンポーネントのネットワークを介してメッセージを送信します。アプリケーションは、IBM WebSphere MQ に接続して、メッセージを送受信します。IBM WebSphere MQ は、メッセージを転送する際に検出する、さまざまなプロセッサ、オペレーティング・システム、サブシステム、通信プロトコルに対応します。接続またはプロセッサが一時的に使用できない場合、IBM WebSphere MQ はメッセージをキューに書き込み、接続がオンラインに復帰したときにそのメッセージを転送します。
- アプリケーションは、プログラミング・インターフェース、および IBM WebSphere MQ に接続するプログラミング言語を選択できます。
- IBM WebSphere MQ は、*Point-to-Point*、パブリッシュ/サブスクライブ、およびファイル転送の操作モードを備えた、メッセージングおよびキューイングのミドルウェアです。アプリケーションは、マルチキャストで複数のサブスクライバーにメッセージをパブリッシュできます。

メッセージング

プログラムは、相互に直接呼び出すのではなく、メッセージ形式で相互にデータをやりとりすることによって通信を行います。

キューイング

メッセージがキューに置かれるため、プログラムは異なる速度や時間、異なる場所で、またプログラム間で直接接続せずに、相互に独立した形で実行できます。

Point-to-point

アプリケーションは、メッセージをキューまたはキューのリストに送信します。送信側は、宛先の名前を知っている必要がありますが、存在する場所については知る必要がありません。

publish/subscribe

アプリケーションは、チームでプレイされるゲームの結果など、トピック上のメッセージをパブリッシュします。IBM WebSphere MQ は、メッセージのコピーを、結果トピックにサブスクライブするアプリケーションに送信します。それらのアプリケーションは、チームによってプレイされるゲームの結果としてメッセージを受信します。パブリッシャーは、サブスクライバーの名前、またはその存在する場所を知りません。

マルチキャスト

マルチキャストは、多数のサブスクライバーに拡張可能な、パブリッシュ/サブスクライブ・メッセージングの効率の良い形式です。マルチキャストは、パブリケーションのコピーを送信する働きを各サブスクライバー (IBM WebSphere MQ からネットワークまで) に伝達します。パブリケーションのパスがパブリッシャーとサブスクライバーの間でいったん確立されると、IBM WebSphere MQ はパブリケーションの転送に関与しなくなります。

ファイル転送

ファイルは、メッセージという形で転送されます。IBM WebSphere MQ File Transfer Edition は、ファイル転送を管理し、転送の自動化を設定したり、結果をログに記録したりします。ファイル転送を他のファイル転送システム、IBM WebSphere MQ メッセージング、および Web と統合できます。

Telemetry (遠隔測定)

IBM WebSphere MQ Telemetry は、デバイス用のメッセージングです。IBM WebSphere MQ は、デバイスとアプリケーション・メッセージングを共に結び付けます。これにより、インターネット、アプリケーション、サービス、および意思決定者は、機能の備わったデバイスのネットワークと結ばれます。IBM WebSphere MQ テレメトリーには、ネットワークを介して多数のデバイスを接続する効率的なメッセージング・プロトコルがあります。デバイスに組み込むことができるように、メッセージング・プロトコルがパブリッシュされます。そのプロトコルのパブリッシュされたプログラミング・インターフェースの 1 つを使用して、デバイス・プログラムを開発することもできます。

利点

- IBM WebSphere MQ は、アプリケーション間で、ネットワーク経由でデータを送受信します。
- メッセージ・デリバリーは確実であり、アプリケーションからは分離されています。IBM WebSphere MQ はメッセージをトランザクションとして交換するため、確実です。また、アプリケーションは送信したメッセージが無事に送達されたことを検査する必要がないため、分離されています。
- SSL/TLS を使用して、キュー・マネージャー間のメッセージ送達を保護できます。
- Advanced Message Security (AMS) を使用して、1つのアプリケーションによって書き込まれ、別のアプリケーションによって取得されるメッセージの暗号化および署名を行えます。
- アプリケーション・プログラマーには通信プログラミングの知識は必要ありません。

使用方法

- IBM WebSphere MQ Explorer GUI を使用するか、コマンド・ウィンドウまたはアプリケーションからコマンドを実行して、IBM WebSphere MQ を作成および管理します。
- プログラミング・インターフェースの1つを呼び出してメッセージを送受信するように、アプリケーションをプログラムします。プログラミング・インターフェースは、複数の言語で提供され、標準的な JMS プログラミング・インターフェース、および Windows Communication Foundation のクラスを組み込みます。
- HTTP プロトコルを使用してブラウザから IBM WebSphere MQ メッセージを送受信します。

動作方法

- 管理者は、コマンドを使用してキュー・マネージャーを作成および開始します。その後は、オペレーティング・システムが起動するときに、通常、キュー・マネージャーは自動的に開始します。アプリケーションおよびその他のキュー・マネージャーは、メッセージを送受信するために、そのキュー・マネージャーに接続できます。
- アプリケーションまたは管理者は、キューまたはトピックを作成します。キューおよびトピックは、キュー・マネージャーによって所有および保管されるオブジェクトです。
- アプリケーションが別のアプリケーションにデータを転送する場合は、データをメッセージに書き込みます。そのアプリケーションは、メッセージをキューに書き込むか、またはメッセージをトピックにパブリッシュします。主に以下の3つの方法で、メッセージを取得できます。
 - 同じキュー・マネージャーに接続された Point-to-Point アプリケーションは、同じキューからメッセージを取得します。

例えば、アプリケーションは、一時データまたは永続データを保管する方法として、メッセージをキューに書き込みます。2つ目の例として、異なるプロセスで実行している別のアプリケーションとデータを共有するアプリケーションがあります。
 - 別のキュー・マネージャーに接続された Point-to-Point アプリケーションは、異なるキューから同じメッセージを取得します。

アプリケーションは、キューでメッセージを交換することにより、相互に通信します。IBM WebSphere MQ の主な使用法は、メッセージを送信または交換することです。1つのアプリケーションが1台のコンピューター上でキューにメッセージを書き込み、別のアプリケーションが異なるコンピューター上で別のキューから同じメッセージを受信します。2台のコンピューター上にあるキュー・マネージャーは連携して、最初のキューから2番目のキューにメッセージを転送します。これらのアプリケーションは相互に通信せず、キュー・マネージャーが通信します。
 - キュー・マネージャーに接続されたサブスクライバー・アプリケーションは、共通トピックでメッセージを取得します。

パブリッシャー・アプリケーションは、メッセージを作成し、そのメッセージを1台のコンピューター上にあるトピックにパブリッシュします。任意の数のサブスクライバー・アプリケーションが、異なるコンピューター上の同じトピックにサブスクライブします。IBM WebSphere MQ サブスクライバーが接続されているキュー・マネージャーに属するキューにパブリケーションを送達します。サブスクライバーは、メッセージをそれらのキューから取得します。

- MQ チャネルは、1つのキュー・マネージャーをネットワーク上の別のキュー・マネージャーに接続します。ユーザーは自分で MQ チャネルを作成できます。または、必要に応じて、キュー・マネージャーのクラスター内のキュー・マネージャーによって、MQ チャネルが作成されます。
- 1つのキュー・マネージャー上に、多数のキューやトピックを設定できます。
- 1つのコンピューター上に複数のキュー・マネージャーを持つことができます。
- アプリケーションは、キュー・マネージャーと同じコンピューター上で実行することも、別のコンピューター上で実行することもできます。アプリケーションを同じコンピューター上で実行する場合、それは IBM WebSphere MQ サーバー・アプリケーションになります。アプリケーションを別のコンピューター上で実行する場合、それは IBM WebSphere MQ クライアント・アプリケーションになります。IBM WebSphere MQ クライアントであっても、サーバーであっても、アプリケーションにほとんど違いはありません。クライアント/サーバー・アプリケーションを IBM WebSphere MQ クライアントまたはサーバーで作成できます。

IBM WebSphere MQ に付属するツールやリソースの概要

- コマンド・ラインから実行する制御コマンド。制御コマンドを使用して、キュー・マネージャーの作成、開始、および停止を実行します。また、制御コマンドを使用して、IBM WebSphere MQ の管理および問題判別プログラムを実行できます。
- インタープリターによって実行される IBM WebSphere MQ スクリプト・コマンド (MQSC)。これらのコマンドを使用して、キューおよびトピックの作成、および IBM WebSphere MQ の構成や管理を実行します。ファイルでコマンドを編集し、そのファイルを **runmqsc** プログラムに渡すと、それらのコマンドが解釈されます。また、別のキュー・マネージャーを管理する別のコンピューターにコマンドを送信する、1つのキュー・マネージャー上でインタープリターを実行することもできます。
- IBM WebSphere MQ を管理するために、独自のアプリケーションで呼び出すプログラマブル・コマンド・フォーマット (PCF) コマンド。PCF コマンドには、スクリプト・コマンドと同じ機能がありますが、より簡単にプログラムできます。
- サンプル・プログラム。
- Windows、および Linux® x86 と x86-64 プラットフォームでは、以下のユーティリティーを実行できます。
 - IBM WebSphere MQ Explorer。このエクスプローラーは、スクリプト・コマンドと同じ管理タスクを実行しますが、より簡単に対話式に使用できます。
 - *Postcard* アプリケーション。メッセージングのデモを行って、インストールを検証します。
 - チュートリアル。

関連概念

[18 ページの『IBM WebSphere MQ Version 7.5 の新機能』](#)

IBM WebSphere MQ Version 7.5 の主な新機能について学習します。

[WebSphere MQ Multicast](#)

IBM WebSphere MQ Multicast は、待ち時間が短く、ファンアウトが大きい、高信頼性マルチキャスト・メッセージングです。

[WebSphere MQ Telemetry](#)

世間一般や企業、行政機関の間で、私たちが生活し、働いている環境との連携を、IBM WebSphere MQ Telemetry を使ってもっとスマートなものにしたいという願望がますます強くなっています。IBM WebSphere MQ Telemetry は、あらゆる種類の装置をインターネットや企業に接続し、スマート・デバイス用アプリケーションの作成コストを削減します。

[メッセージングおよびキューイングの技術概要](#)

WebSphere MQ 製品では、整合性のあるアプリケーション・プログラミング・インターフェースを使用して、性質の異なるコンポーネント (プロセッサ、オペレーティング・システム、サブシステム、および通信プロトコル) のネットワーク内でプログラムが相互に通信できるようにしています。

[WebSphere MQ クライアントおよびサーバーの技術概要](#)

IBM WebSphere MQ がアプリケーションのためにクライアント/サーバー構成をどのようにサポートしているかについて、概要を説明します。

[キュー・マネージャー通信の技術概要](#)

WebSphere MQ において、相互通信とは、あるキュー・マネージャーから別のキュー・マネージャーへメッセージが送信されることを意味します。受信キュー・マネージャーは、同じマシン上に置くことも別のマシン上に置くこともでき、近くにあっても地球の裏側にあってもメッセージを受信できます。このプログラムは、ローカル・キュー・マネージャーと同じプラットフォーム、または WebSphere MQ がサポートするいずれかのプラットフォームで実行することができます。これを分散環境と呼びます。WebSphere MQ は、このような分散環境での通信を分散キューイング管理 (DQM) によって処理します。

関連タスク

[WebSphere MQ Advanced Messages Security \(AMS\)](#)

[WebSphere MQ Managed File Transfer](#)

IBM WebSphere MQ ライセンス情報

購入可能な IBM WebSphere MQ 関連製品、および購入した各製品によってインストールする資格が付与される製品。

注意: このライセンス・ガイドには、購入したライセンス内で IBM からライセンス交付を受けたプログラムをデプロイする際に役立つ補足情報が記載されています。お客様の使用許諾契約書 (IBM プログラムのご使用条件 (IPLA) またはそれに相当するもの)、およびその取引文書 (IBM WebSphere MQ 製品オファリングのライセンス情報を含む) は、お客様と IBM との間で締結される、プログラムの使用に関する唯一かつ完全な契約書です。

購入可能な IBM WebSphere MQ 関連製品

分散プラットフォーム

分散プラットフォーム上の IBM WebSphere MQ の場合、製品オファリングには、別個に購入できる 11 個の課金対象のコンポーネントが含まれています。

5724-H72 IBM IBM WebSphere MQ

IBM IBM WebSphere MQ (サーバー)

IBM IBM WebSphere MQ テレメトリー

IBM IBM WebSphere MQ 拡張メッセージ・セキュリティー

IBM IBM WebSphere MQ アイドル・スタンバイ

IBM IBM WebSphere MQ Advanced Message Security アイドル・スタンバイ

IBM IBM WebSphere MQ 拡張

IBM IBM WebSphere MQ 拡張アイドル・スタンバイ

IBM IBM WebSphere MQ Advanced for Developers

IBM IBM WebSphere MQ Managed File Transfer サービス

IBM IBM WebSphere MQ Managed File Transfer サービス・アイドル・スタンバイ

IBM IBM WebSphere MQ Managed File Transfer 管理対象エンドポイント

社内で自分がインストールする資格のある製品

分散プラットフォーム上の IBM WebSphere MQ について、以下のコンポーネントは、IBM WebSphere MQ インストーラーがインストールできるコンポーネントに直接マップされています。そのため、購入した製品とインストールできる製品のマッピングは簡単に行えます。

重要: IBM WebSphere MQ インストール・メディアにはすべてのコンポーネントが含まれていますが、ライセンスを購入したサブセットのみをインストールする必要があります。

5724-H72 IBM IBM WebSphere MQ

IBM WebSphere (サーバー)

以下が含まれます。

ClientDevelopment Kit (SDK)

IBM グローバル・セキュリティー・キット (UNIX)
IBM WebSphere MQ エクスプローラー
Java .NET メッセージングおよび Web サービス
サンプル・プログラム
サーバー/ランタイム
UNIX マニュアル・ページ

IBM IBM WebSphere MQ テレメトリー

以下が含まれます。

Telemetry サービス

IBM IBM WebSphere MQ 拡張メッセージ・セキュリティー

以下が含まれます。

Advanced Message Security

IBM IBM WebSphere MQ Managed File Transfer サービス

以下が含まれます。

IBM WebSphere MQ Managed File Transfer ロガー
IBM WebSphere MQ Managed File Transfer サービス
IBM WebSphere MQ Managed File Transfer ツール

IBM WebSphere IBM WebSphere MQ Managed File Transfer 管理対象エンドポイント

以下が含まれます。

IBM WebSphere MQ Managed File Transfer エージェント
IBM WebSphere MQ Managed File Transfer ロガー
IBM WebSphere MQ Managed File Transfer ツール

IBM IBM WebSphere MQ Advanced について

IBM WebSphere MQ Advanced は、ライセンスの購入を簡単にするために導入されています。企業は 1 回の支払いで、複数の IBM WebSphere MQ コンポーネントのライセンスを得ることができます。

IBM WebSphere MQ Advanced には、以下のパーツが含まれます。

5724-H72 IBM IBM WebSphere MQ

IBM IBM WebSphere MQ 拡張

IBM IBM WebSphere MQ Advanced for Developers

分散プラットフォームで IBM IBM WebSphere MQ を使用する場合は、IBM WebSphere MQ Advanced を 100 プロセッサ・バリュー・ユニット (PVU) 購入すると、次をインストールするための企業ライセンスが与えられます。

- IBM IBM WebSphere MQ (サーバー) の 100 PVU、および
- IBM IBM WebSphere MQ Advanced Message Security の 100 PVU、および
- IBM IBM WebSphere MQ Managed File Transfer Service の 100 PVU、および
- IBM IBM WebSphere MQ Telemetry を無制限でインストールできます。

さらに、必要に応じてエンタープライズ規模で IBM のバージョンのミックス・アンド・マッチが可能です。そのため、100 PVU の IBM IBM WebSphere MQ (サーバー) ライセンスを、このコンポーネントの IBM WebSphere MQ 7.1 バージョンの 50 PVU と IBM WebSphere MQ 7.5 バージョンの 50 PVU に分割することができます。

IBM IBM WebSphere MQ Advanced for Developers は、開発目的での使用に限り、IBM IBM WebSphere MQ Advanced および IBM IBM WebSphere MQ Managed File Transfer Managed Endpoint のすべてのコンポーネントに対するライセンスを提供します。



重要: IBM ライセンスはどの製品が開発目的として見なされるかを定義します。

Idle Standby パーツとは

Idle Standby パーツは、パッシブ・システムに IBM WebSphere MQ がインストールされ、使用可能にされているが、アクティブなキュー・マネージャーの構成やアクティビティに合わせて最新の状態を維持する以外にシステムが実行する IBM WebSphere MQ 処理やアクティビティは何もないという、高可用性環境に対応するために導入されました。この場合、よりお手頃な料金が適用できる場合があります。

注:

1. IBM WebSphere MQ 複数インスタンス・キュー・マネージャー機能の使用には、Idle Standby ライセンスも必要になります。
2. IBM WebSphere MQ Telemetry コンポーネントの Idle Standby パーツはありません。同じ IBM WebSphere MQ Telemetry パーツをアクティブおよびパッシブ・システム用に購入する必要があります。ただし、IBM WebSphere MQ Advanced Idle Standby を所有している場合は、これに含まれているので購入不要です。
3. エンドポイントはサーバー環境の一部ではないため、IBM WebSphere Managed File Transfer Managed Endpoint の Idle Standby パーツもありません。

関連概念

[24 ページの『IBM WebSphere MQ Version 7.5 の変更点』](#)

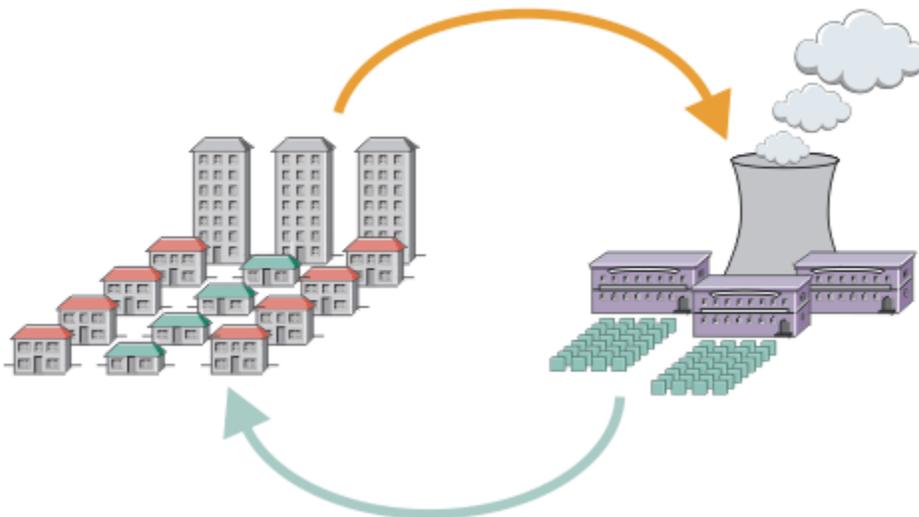
[42 ページの『IBM WebSphere MQ 技術概説』](#)

IBM WebSphere MQ を使用して、アプリケーションに接続し、組織全体での情報の分散を管理します。

IBM WebSphere MQ Telemetry の概要

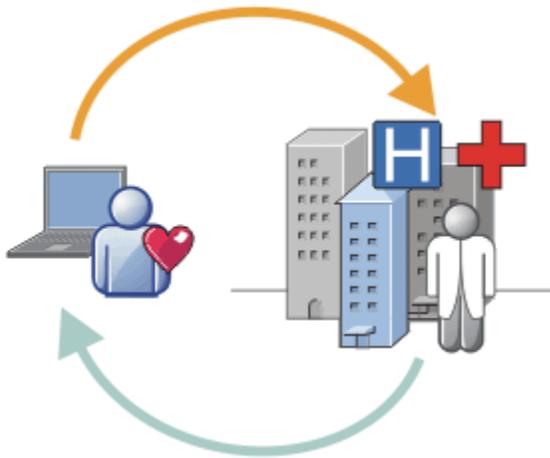
世間一般や企業、行政機関の間で、私たちが生活し、働いている環境との連携を、IBM WebSphere MQ Telemetry を使ってもっとスマートなものにしたいという願望がますます強くなっています。IBM WebSphere MQ Telemetry は、あらゆる種類の装置をインターネットや企業に接続し、スマート・デバイス用アプリケーションの作成コストを削減します。

以下の各図は、IBM WebSphere MQ Telemetry の代表的な使用例を示したものです。



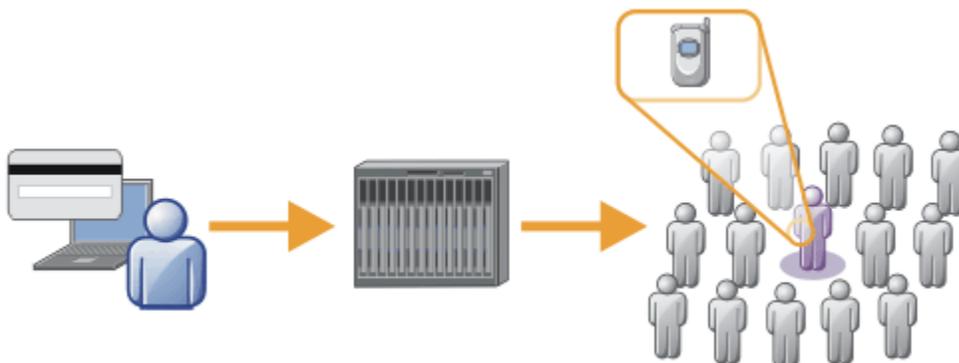
- エネルギー使用量データを含んだ MQTT メッセージがサービス・プロバイダーに送られます。
- 遠隔測定アプリケーションは、エネルギー使用量データの分析に基づいた制御コマンドを送信します。
- 詳しくは、[73 ページの『遠隔測定シナリオ: 家庭エネルギーのモニターと制御』](#) を参照してください。

図 1. スマート電力計量



- 遠隔測定アプリケーションは、医療データを病院と医師に送信します。
- 医療データの分析に基づいて、MQTT メッセージの警報やフィードバックが返されます。
- 詳しくは、[72 ページの『遠隔測定シナリオ: 在宅患者モニター』](#)を参照してください。

図 2. スマート・ヘルスケア・モニター



- シンプルなカード・トランザクションが銀行のサーバーに送られます。
- IBM WebSphere MQ Telemetry は、数千名の中から一人の個人を特定して、その顧客のカードが使用されているという警報を返します。
- IBM WebSphere MQ Telemetry は、最もシンプルな入力情報に基づいてその個人を特定します。

図 3. Telemetry: 大勢の中の一人

WebSphere MQ Telemetry とは

- これは IBM WebSphere MQ のフィーチャーの 1 つで、IBM WebSphere MQ によって提供される汎用メッセージング・バックボーンを広範なリモート・センサー、アクチュエーター、およびテレメトリー装置に拡張します。IBM WebSphere MQ Telemetry は、IBM WebSphere MQ を拡張して、インテリジェントなエンタープライズ・アプリケーション、サービス、および意思決定者をインストールされたデバイスのネットワークと相互接続できるようにします。
- WebSphere MQ Telemetry の 2 つの中核部分は以下のとおりです。
 1. IBM WebSphere MQ サーバーの内部で実行される IBM WebSphere MQ Telemetry サービス。
 2. アプリケーションと共にデバイスに配布される IBM WebSphere MQ Telemetry クライアント。

利点

- MQ Telemetry は、MQ Telemetry Transport (MQTT) を使用して、アプリケーションと IBM WebSphere MQ キュー・マネージャーとの間でデータを送受信します。
- MQTT は、幅広い種類のデバイス用に MQTT 実装環境を作成できるようにするオープン・メッセージング・トランスポートです。
- MQTT クライアントは、リソースが制限されている可能性のある、フットプリントの小さいデバイスで実行できます。
- MQTT は、処理能力が低い可能性があるネットワークや、データ送信の費用が高額なネットワーク、または不安定なネットワークで効率的に機能します。
- メッセージ・デリバリーは確実であり、アプリケーションからは分離されています。
- アプリケーション・プログラマーには通信プログラミングの知識は必要ありません。
- メッセージを、他のメッセージング・アプリケーションと交換することができます。他の遠隔測定アプリケーション、MQI、JMS またはエンタープライズ・メッセージング・アプリケーションなどです。

使用方法

- IBM WebSphere MQ Explorer とその関連ツールを使用して、MQ の WebSphere MQ Telemetry 機能を管理します。
- アプリケーションの MQTT クライアントを使用して、キュー・マネージャーへの接続、メッセージのプブリッシュおよびサブスクライブを行います。
- アプリケーションを実行するデバイスに、MQTT クライアントと共にアプリケーションを配布します。

動作方法

- MQ Telemetry (MQXR) サービスは、IBM WebSphere MQ キュー・マネージャーを MQTT サーバーに変換します。
- MQTT サーバーは MQTT メッセージ・トランスポートを理解し、MQTT クライアントとの間でメッセージの送受信ができます。
- MQ Telemetry は、MQTT メッセージ・トランスポートを実装した複数の Telemetry クライアントと共に出荷されます。これらは多くの場合 MQTT クライアントと呼ばれます。
- 基本的な Telemetry クライアントは標準の MQ クライアントのように動作しますが、より幅広い種類のプラットフォームおよびネットワークで稼働できます。
- Advanced Telemetry Client は、膨大な数の MQTT クライアントを単一のキュー・マネージャーに接続するためのネットワーク・コンセントレーターの役割を果たすことができます。短時間のネットワーク障害時にメッセージをバッファに入れる手段を欠く小型装置のためのストア・アンド・フォワードも提供します。
- IBM WebSphere MQ Telemetry daemon for devices は、IBM WebSphere MQ Telemetry の一部である Advanced Telemetry クライアントです。詳細については、[99 ページの『Telemetry デーモン \(デバイス用\)』](#)を参照してください。
- MQTT は、プブリッシュ/サブスクライブ・プロトコルです。
 - MQTT クライアント・アプリケーションは MQTT サーバーにメッセージをプブリッシュできます。
 - IBM WebSphere MQ キュー・マネージャーが MQTT サーバーとして機能する場合、キュー・マネージャーに接続する他のアプリケーションは、MQTT クライアントにサブスクライブして、MQTT クライアントからメッセージを受信することができます。
 - MQTT クライアントは、MQ キュー・マネージャーに接続するアプリケーションにより送信されたメッセージにサブスクライブできます。
 - キュー・マネージャーは、プブリッシュするアプリケーションからサブスクライブするアプリケーションへメッセージを配布するルーターとして動作します。
 - メッセージは、異なるタイプのクライアント・アプリケーション間で配布できます。例えば、Telemetry クライアントと JMS クライアント間などです。

IBM WebSphere MQ Telemetry は、WebSphere Message Broker のバージョン 7 で廃止され、Windows、Linux、および AIX® で実行される SCADA ノードを置き換えます。「[Migration of telemetry applications from using WebSphere Message Broker version 6 to use IBM WebSphere MQ Telemetry and WebSphere Message Broker version 7.0](#)」には、WebSphere Message Broker V6 の SCADA ノードを使用してアプリケーションをマイグレーションする際に役立つ情報が記載されています。WebSphere Message Broker バージョン 7 を使用するテレメトリー・アプリケーションは、MQTT クライアントに共通のトピックにサブスクライブします。これらのアプリケーションは MQInput ノードを使用して MQTT クライアントからパブリケーションを受信し、パブリケーション・ノードを使用して MQTT クライアントにパブリッシュします。

関連概念

[70 ページの『モニターと制御に関するテレメトリーの概念とシナリオ』](#)

テレメトリーは、自動化されたセンシング、データの測定、およびリモート装置の制御を行うものです。特に重要なのは、装置から中央制御点へのデータ伝送です。遠隔測定には、構成情報と制御情報を装置に送信することも含まれます。

関連タスク

[WebSphere MQ Telemetry のインストール](#)

[WebSphere MQ Telemetry の管理](#)

[WebSphere Message Broker v6 を使用している遠隔測定アプリケーションを WebSphere MQ Telemetry および WebSphere Message Broker v7 を使用するようにマイグレーションする](#)

[WebSphere MQ Telemetry のアプリケーションの開発](#)

[WebSphere MQ Telemetry のトラブルシューティング](#)

関連資料

[WebSphere MQ Telemetry リファレンス](#)

関連情報

[68 ページの『IBM WebSphere MQ Telemetry』](#)

IBM WebSphere MQ Version 7.5 の情報ロードマップ

この情報ロードマップには、さまざまな IBM WebSphere MQ Version 7.5 リソースへのリンクが含まれています。

このロードマップはさまざまなソースからの情報を 1 つにまとめ、IBM WebSphere MQ の特定の分野に関する詳細情報を見つけやすくします。ロードマップ内の各セクションへのリンクをクリックし、どのようなリソースが入手可能であるかを確認してください。

- [製品概要](#)
- [技術概要](#)
- [シナリオ](#)
- [計画](#)
- [マイグレーションおよびアップグレード](#)
- [インストール](#)
- [セキュリティー](#)
- [構成](#)
- [管理](#)
- [アプリケーションの開発](#)
- [モニターおよびパフォーマンス](#)
- [トラブルシューティングおよびサポート](#)
- [参照情報](#)

表 1. IBM WebSphere MQ 情報ロードマップの表

カテゴリー	情報源
製品概要	<p>IBM WebSphere MQ 全体の目的、機能、新機能に関する概要。</p> <p>❖ 5 ページの『IBM WebSphere MQ について』 IBM WebSphere MQ Version 7.5 の使用を開始するにあたって役に立つ初歩的信息。製品の概要、このリリースの新機能および変更点についての概要など。</p> <p>IBM WebSphere MQ Version 7.5</p> <p>この IBM Redbooks パブリケーションでは、IBM WebSphere MQ Version 7.5 で作成された中核となる機能拡張、および理解しておく必要がある概念について説明されます。</p> <p>IBM WebSphere MQ 製品 Web ページ リソースや追加情報へのリンクを含む製品 Web ページ。</p> <p>IBM WebSphere MQ システム要件 IBM WebSphere MQ のさまざまなリリースに関するシステム要件へのリンクを含む Web ページ。</p> <p>17 ページの『IBM Documentation オフラインアプリケーションの IBM WebSphere MQ Version 7.5』 IBM WebSphere MQ Version 7.5 メッセージングの資料は、ローカルにインストールしたオフライン・バージョンの IBM Documentation にダウンロードできます。</p>
技術概要	<p>❖ 42 ページの『IBM WebSphere MQ 技術概説』</p> <p>IBM WebSphere MQ Version 7.5 で提供されるメッセージ・キューイングおよびその他の機能についての情報を得ることができます。</p>
シナリオ	<p>各シナリオにより、相当数のタスクの集合を概観できるので、主要な製品フィーチャーを構成するのに役立ちます。このシナリオには、関心のある分野についてさらに理解を深めるうえで役立つ他のコンテンツへの有用なリンクが含まれています。</p> <p>❖ 142 ページの『シナリオ』 「準備作業」シナリオでは、IBM WebSphere MQ を開始する方法について説明します。IBM WebSphere MQ を以前に使用したことがなく、すぐに使用を開始する場合に、このシナリオを使用します。その他にも役に立つシナリオが用意されているので、対応するタスクの手順に従って製品機能を構成したり使用したりできます。</p> <p>❖ WebSphere Application Server の IBM WebSphere MQ への接続</p> <p>さまざまなシナリオで WebSphere Application Server を IBM WebSphere MQ に接続するために必要な主なタスクを実行するのに役立つ情報が含まれます。</p> <p>❖ WebSphere Application Server Liberty プロファイルの IBM WebSphere MQ</p> <p>さまざまなシナリオで WebSphere Application Server Liberty プロファイルを IBM WebSphere MQ に接続するために必要な主なタスクを実行するのに役立つ情報が含まれます。</p> <p>❖ IBM MessageSight の IBM WebSphere MQ および WebSphere Application Server</p> <p>さまざまなシナリオで IBM MessageSight を IBM WebSphere MQ および WebSphere Application Server に接続するために必要な主要なタスクについて説明します。</p>

表 1. IBM WebSphere MQ 情報ロードマップの表 (続き)

カテゴリー	情報源
計画	<p>❖ 計画</p> <p>IBM WebSphere MQ 環境を計画する際、単一および複数キュー・マネージャーのアーキテクチャーについて、また Point-to-Point およびパブリッシュ/サブスクライブのメッセージング・スタイルについて IBM WebSphere MQ が提供するサポートを考慮します。また、リソース要件、およびロギングやバックアップの機能の使用方法を計画します。</p>
マイグレーションおよびアップグレード	<p>❖ マイグレーションおよびアップグレード</p> <p>キュー・マネージャーを、新しいコード・レベルで実行するようにマイグレーションするには、まず IBM WebSphere MQ をアップグレードして、新しいコード・レベルをインストールする必要があります。アップグレードが成功したことを検証したら、キュー・マネージャー、およびキュー・マネージャーに関連付けられているすべてのアプリケーションとリソースを移行します。このプロセスを開始する前に、このマイグレーション・ガイドの情報に基づいて、移行計画を作成します。保守を適用する場合、移行は不要です。ただし、新しいレベルの IBM WebSphere MQ コードを使用してアプリケーションをテストする必要があります。</p> <p>❖ WebSphere / IBM MQ 移行ガイド</p> <p>このガイドには、古いバージョンから新しいバージョンに移行する処理を計画する際に役立つ情報が記載されています。Web ブラウザーでガイドを表示したり、PDF ファイルとしてダウンロードしたりできます。</p>
インストール	<p>❖ インストールおよびアンインストール</p> <p>インストールの準備、製品のインストール、およびインストールの検証を行ううえで役立つ情報。また、製品のアンインストールに役立つ情報もあります。</p>
セキュリティ	<p>❖ 安全</p> <p>識別と認証、許可、監査、機密性、データ保全性など IBM WebSphere MQ インストール済み環境で考慮すべきセキュリティの側面。</p>
構成	<p>❖ 構成</p> <p>1 台以上のコンピューター上に 1 つ以上のキュー・マネージャーを作成し、ビジネス・データを含むメッセージを処理するため、そのキュー・マネージャーおよび開発、テスト、実動システム上の関連するリソースを構成します。</p>
管理	<p>❖ 管理 IBM WebSphere MQ</p> <p>キュー・マネージャーおよび関連付けられたリソースを管理します。</p>
アプリケーションの開発	<p>❖ アプリケーションの開発</p> <p>メッセージを送受信したり、キュー・マネージャーと関連リソースを管理したりするためのアプリケーションを開発します。IBM WebSphere MQ では、プロシージャー型言語で記述されたアプリケーション、およびオブジェクト指向言語とフレームワークで記述されたアプリケーションがサポートされます。</p>
モニターおよびパフォーマンス	<p>❖ モニターおよびパフォーマンス</p> <p>キュー・マネージャー・ネットワークのパフォーマンスを向上させるのに役立つモニター情報とガイダンス、およびキュー・マネージャー・ネットワークのパフォーマンス・チューニングのヒント。</p>

表 1. IBM WebSphere MQ 情報ロードマップの表 (続き)

カテゴリー	情報源
トラブルシューティングおよびサポート	<p>❖ トラブルシューティングとサポート キュー・マネージャー・ネットワークや IBM WebSphere MQ アプリケーションに関する問題を診断して解決するための手法。</p> <p>IBM Support Assistant Web ページ IBM Support Assistant (ISA) は、サポート関連情報へのアクセスやトラブルシューティング・ツールを提供することによって、IBM ソフトウェア製品に関する質問や問題の解決を支援します。</p> <p>IBM サポート・ポータル Web ページ IBM WebSphere MQ の IBM サポート・ポータル。</p>
参照情報	<p>❖ 参照 構成、管理、アプリケーション開発、テレメトリー、セキュリティー、モニタリング、トラブルシューティングとサポート、および診断メッセージについての参照情報。</p>

IBM Documentation オフライン アプリケーションの IBM WebSphere MQ Version 7.5

インターネットにアクセスできないエアギャップ環境の場合は、**ダーク・ショップ・アプリ「IBM Documentation オフライン」**を使用して、IBM WebSphere MQ Version 7.5 製品資料のダウンロードを表示します。

IBM Documentation オフラインには2つのコンポーネントがあります。

- **IBM Documentation オフライン アプリ**。これは、ローカルにインストール可能なオフライン・バージョンの IBM Documentation です。
- **IBM Documentation オフライン アプリにインストールするドキュメンテーション・パッケージ**。これらのパッケージには、IBM Documentation でオンラインで公開されているものと同じ資料が含まれています。

アプリケーションおよび IBM WebSphere MQ Version 7.5 資料パッケージをダウンロードするには、IBM Documentation にログインする必要があります。詳しくは、[IBM Documentation オフライン](#)を参照してください。

IBM WebSphere MQ Version 7.5 PDF 資料

IBM WebSphere MQ Version 7.5 製品資料は、一連の PDF ファイルとしてダウンロードできます。

IBM WebSphere MQ Version 7.5 PDF ファイルには、IBM Documentation で公開されている HTML バージョンと同じ情報が含まれています。以下の表のリンクをクリックして、それらをダウンロードすることができます。

注：また、ローカルにインストールしたオフライン・バージョンの IBM Documentation に、HTML 形式の IBM WebSphere MQ Version 7.5 製品資料をダウンロードすることもできます。詳しくは、[17 ページの『IBM Documentation オフライン アプリケーションの IBM WebSphere MQ Version 7.5』](#)を参照してください。

PDF ファイルの名前とダウンロード・リンク	IBM Documentation 内の対応するセクション
mq75.overview.pdf	製品情報 IBM WebSphere MQ
mq75.plan.pdf	計画

表 2. 製品資料の PDF ファイルと、その IBM Documentation のセクションとの対応 (続き)

PDF ファイルの名前とダウンロード・リンク	IBM Documentation 内の対応するセクション
mq75.migrate.pdf	マイグレーションおよびアップグレード
mq75.install.pdf	インストール
mq75.secure.pdf	機密保護
mq75.configure.pdf	構成
mq75.administer.pdf	の管理
mq75.develop.pdf	アプリケーションの開発
mq75.monitor.pdf	モニターおよびパフォーマンス
mq75.troubleshoot.pdf	トラブルシューティングとサポート
mq75.reference.pdf	参照
mq75.refcon.pdf	構成に関する参照情報
mq75.refadmin.pdf	管理に関する参照情報
mq75.refdev.pdf	アプリケーションの開発に関する参照情報
mq75.mft.pdf	IBM WebSphere MQ Managed File Transfer
mq75.explorer.pdf	IBM WebSphere MQ Explorer
mq75.guiwizard.pdf	MQ GUI ウィザードのヘルプ
mq75.xms.pdf	Message Service Client for .NET
mq75.hve.pdf	WebSphere MQ Hypervisor Edition
mq75.mmtc.pdf	モバイル・メッセージングおよび M2M
MQ Migration Guide.pdf	WebSphere / IBM MQ マイグレーション・ガイド

注: PDF ファイルは、PDF ファイル間のリンクが正しく機能するように同じフォルダーに入っていないとばなりません。

IBM WebSphere MQ Version 7.5 の新機能

IBM WebSphere MQ Version 7.5 の主な新機能について学習します。

- [19 ページの『IBM WebSphere MQ ファイル転送管理』](#) .
- [19 ページの『IBM WebSphere MQ Advanced Message Security』](#) .
- [19 ページの『メッセージ・チャンネル・エージェント \(MCA\) インターセプト』](#) .
- [19 ページの『複数のクラスター伝送キュー』](#) .
- [19 ページの『拡張トランザクション機能がコア・クライアントの一部になった』](#) .
- [19 ページの『アプリケーション名の設定によるキュー・マネージャーへの接続の識別』](#) .
- [20 ページの『証明書妥当性検査ポリシー』](#) .
- [20 ページの『トランザクションの可視性の向上』](#) .
- [20 ページの『シナリオ』](#) .
- [20 ページの『IBM WebSphere MQ Explorer』](#) .

IBM WebSphere MQ ファイル転送管理

IBM WebSphere MQ Managed File Transfer は、IBM WebSphere MQ を使用して、キュー・マネージャー間でファイルを転送します。その範囲をキュー・マネージャーを持たないワークステーションやサーバーに拡張できます。ファイル転送エージェント Apache Ant を使用することにより、また、IBM Sterling Commerce®:Direct、Web ゲートウェイ、および SFTP、FTP、または FTPS のプロトコル・サーバーと統合することにより、その範囲を拡張できます。

IBM WebSphere MQ Managed File Transfer を使用すると、ファイル転送の自動化、制御、保護、および監査を行うことができます。[IBM WebSphere MQ Managed File Transfer の概要](#)を参照してください。

IBM WebSphere MQ Advanced Message Security

IBM WebSphere MQ Advanced Message Security (AMS) は別個にインストールされるコンポーネントであり、別途課金されます。AMS は、IBM WebSphere MQ ネットワークを流れる機密データを高い水準で保護します。AMS を利用するために、既存のアプリケーションを変更する必要はありません。[IBM WebSphere MQ Advanced Message Security](#) を参照してください。

メッセージ・チャネル・エージェント (MCA) インターセプト

MCA インターセプト機能により、Advanced Message Security のライセンス交付を受けたインストール済みの IBM WebSphere MQ の下で実行されるキュー・マネージャーは、サーバー接続チャンネルに適用するポリシーを選択的に使用可能にすることができます。MCA インターセプトを使用することで、IBM WebSphere MQ AMS の外部にあるクライアントは、引き続きキュー・マネージャーに接続し、メッセージを暗号化および復号できます。[メッセージ・チャネル・エージェント \(MCA\) インターセプト](#)を参照してください。

複数のクラスター伝送キュー

新しいキュー・マネージャー属性 **DEFCLXQ** を変更して、クラスター送信側チャンネルごとに別々のクラスター伝送キューを割り当てることができます。各クラスター送信側チャンネルによって転送されるメッセージは、独立したクラスター伝送キューに入れられます。[クラスター伝送キューとクラスター送信側チャンネル](#)を参照してください。新しいキュー属性 **CLCHNAME** を設定して、手動でクラスター伝送キューを構成することもできます。どのクラスター送信側チャンネルがどの伝送キューを共有し、どれが独立した伝送キューを持ち、どれがクラスター伝送キュー (複数可) を使用するのかを決めることができます。[クラスター化: クラスター伝送キューの構成方法の計画](#)を参照してください。このように変更することは、クラスター化されたキュー・マネージャー間でのメッセージ転送を管理するシステム管理者にとって役立ちます。

拡張トランザクション機能がコア・クライアントの一部になった

現在、拡張トランザクション機能は、IBM WebSphere MQ コア・クライアントに組み込まれています。拡張トランザクション・クライアントのライセンスを別途購入したり、別個の拡張トランザクション・クライアント・コンポーネントをインストールしたりする必要はありません。[135 ページの『拡張トランザクション・クライアントの概要』](#)を参照してください。

アプリケーション名の設定によるキュー・マネージャーへの接続の識別

アプリケーションに名前を設定して、アプリケーションからキュー・マネージャーへの接続を識別することができます。**DISPLAY CONN** コマンドを使用して、アプリケーション名を表示します。名前は **APPLTAG** フィールドで返されます。この名前は、IBM WebSphere MQ Explorer の「**アプリケーション接続**」ウィンドウにも表示できます。このフィールドの名前は **App name** です。[WebSphere MQ classes for Java 用の WebSphere MQ 環境のセットアップ](#)を参照してください。z/OS®を除くすべてのプラットフォームで、アプリケーション接続の名前を設定できます。

証明書妥当性検査ポリシー

UNIX, Linux, and Windows では、証明書チェーンの妥当性検査が RFC 5280 業界セキュリティ標準にどの程度厳密に準拠しているかを指定できます。 [WebSphere MQ の証明書妥当性検査ポリシー](#) を参照してください。

トランザクションの可視性の向上

`dspmqrtn` コマンドに、非同期ロールバックが発生したときにより詳細な情報を提供する **-a** および **-q** という 2 つの新規パラメーターが追加されました。 2 つの新規メッセージ [AMQ7486](#) および [AMQ7487](#) によって、ロールバックされるトランザクション、およびトランザクションが接続に関連付けられているかどうかに関する情報が提供されます。

シナリオ

142 ページの『シナリオ』では、IBM WebSphere MQ Version 7.5 の新機能を迅速かつ容易に使用および結合する方法が示されます。 このシナリオには、関心のある分野についてさらに理解を深めるうえで役立つ製品資料のコンテンツへの有用なリンクが含まれています。

IBM WebSphere MQ Explorer

IBM WebSphere MQ Explorer Version 7.5 の新機能として、IBM WebSphere MQ Managed File Transfer と IBM WebSphere MQ Advanced Message Security の統合、複数バージョン・サポートの改善、オーバーヘッドの軽減などが行われています。

IBM WebSphere MQ Explorer の新機能について詳しくは、[WebSphere MQ Explorer の新機能と変更点を参照してください](#)。

関連概念

24 ページの『[IBM WebSphere MQ Version 7.5 の変更点](#)』

24 ページの『[IBM WebSphere MQ Version 7.5 フィックスパックの変更点](#)』

このセクションでは、Version 7.5 フィックスパックで機能およびリソースに加えられた変更点について説明します。

HP Integrity NonStop Server 用の IBM WebSphere MQ クライアント

IBM WebSphere MQ は、HP Integrity NonStop Server プラットフォーム用のクライアントをサポートするようになりました。

概要

HP Integrity NonStop Server プラットフォーム用のクライアントを含む IBM WebSphere MQ クライアントの概要については、[132 ページの『IBM WebSphere MQMQI クライアントの概要』](#)を参照してください。

HP Integrity NonStop Server プラットフォーム用の IBM WebSphere MQ クライアントの技術的な概要については、[141 ページの『IBM WebSphere MQ Client for HP Integrity NonStop Server 技術概要』](#)を参照してください。

IBM WebSphere MQ client for HP Integrity NonStop Server でサポートされる環境と機能については、[141 ページの『IBM WebSphere MQ クライアント \(HP Integrity NonStop Server 用\) でサポートされる環境と機能』](#)を参照してください。

計画

IBM WebSphere MQ Client for HP Integrity NonStop Server 環境を計画する際のヘルプについては、[HP Integrity NonStop Server での IBM WebSphere MQ クライアント環境の計画](#)を参照してください。

インストール

IBM WebSphere MQ client for HP Integrity NonStop Server のインストールに関するヘルプ。

- インストール内容の選択については、[HP Integrity NonStop Server 用の IBM WebSphere MQ クライアント・コンポーネント](#)を参照してください。
- インストールの計画。 [HP Integrity NonStop Server でのインストールの計画](#)を参照してください。
 - [ファイル・システム](#)
- ハードウェア/ソフトウェア要件。 [HP Integrity NonStop Server でのハードウェア要件とソフトウェア要件](#)を参照してください。
- 適切なソフトウェアが存在することの確認。 [システム・ソフトウェア前提条件の確認](#)を参照してください。
- システムの準備。 [HP Integrity NonStop Server でのユーザーとグループのセットアップ](#)を参照してください。
- クライアントのインストール。 [HP Integrity NonStop Server システムでの IBM WebSphere MQ クライアントのインストール](#)を参照してください。
- インストールの検証。 [クライアントのインストールの検証](#)を参照してください。
- アンインストールについては、 [HP Integrity NonStop Server での IBM WebSphere MQ のアンインストール](#)を参照してください。

HP Integrity NonStop Server クライアントのコマンド

以下のコマンドは、HP Integrity NonStop Server OSS および Guardian 環境の IBM WebSphere MQ クライアントに適用されます。

- [dspmqver](#)
- [endmqtrc](#)
- [mqrc](#) (MQ 戻りコード)
- [runmqras](#)
- [runmqtmc](#)
- [strmqtrc](#)

以下のコマンドは、HP Integrity NonStop Server OSS 環境の IBM WebSphere MQ クライアントに適用されます。

- [dspmqtrc](#)

新しい製品 ID である MQNC が [DISPLAY CHSTATUS](#) コマンドの製品 ID の値表に追加されています。

機密保護

IBM WebSphere MQ Client for HP Integrity NonStop Server 環境を保護するには、以下を参照してください。

- IBM WebSphere MQ client for HP Integrity NonStop Server をキュー・マネージャーに識別させる方法に関する情報が、[クライアント・アプリケーションの認証の計画](#)に追加されています。
- [HP Integrity NonStop Server でのセキュリティーのセットアップ](#)
 - [OpenSSL](#)
 - [エントロピー・デーモン](#)
- [IBM WebSphere MQ での SSL および TLS のサポート](#)
- [HP Integrity NonStop Server での SSL または TLS の取り扱い](#)
 - [証明書管理](#)
 - [個人証明書ストア](#)

- [証明書トラストストア](#)
- [パスフレーズ stash ファイル](#)
- [証明書取り消しリスト・ファイル](#)

トランザクション管理機能

トランザクション管理機能 (TMF) について詳しくは、以下のセクションおよびトピックを参照してください。

- [IBM WebSphere MQ での HP Integrity NonStop Server クライアント環境の計画](#)
 - [HP Integrity NonStop Server 環境の準備](#)
 - [IBM WebSphere MQ と HP NonStop TMF](#)
 - [HP NonStop TMF の使用](#)
 - [グローバル作業単位の使用](#)
 - [長期実行トランザクションの回避](#)
 - [事前に構成した非アクティブ間隔が経過した後にグローバル作業単位が有効期限切れとなるキュー・マネージャー構成に関する情報が、グローバル作業単位の有効期限切れに追加されています。](#)
- [HP Integrity NonStop Server の構成](#)
 - [Gateway プロセスの概要](#)
 - [Gateway を Pathway のもとで動作するように構成する](#)
 - [TMF および TMF/Gateway のスタンザ](#)
 - [クライアント初期設定ファイルの構成](#)
 - [チャンネルへのアクセス権付与](#)
- [管理 HP Integrity NonStop Server](#)
 - [TMF/Gateway を Pathway から手動で開始する](#)
 - [TMF/Gateway を Pathway から停止する](#)
- [IBM WebSphere MQ Client for HP Integrity NonStop Server のトラブルシューティング](#)

アプリケーションの開発

HP Integrity NonStop Server プラットフォームでの IBM WebSphere MQ クライアント用アプリケーションの開発については、以下を参照してください。

- [HP Integrity NonStop Server でのアプリケーションの構築](#)
 - [OSS および Guardian ヘッダーとパブリック・ライブラリー](#)
 - [HP Integrity NonStop Server での C プログラムの作成](#)
 - [COBOL プログラムの作成](#)
 - [pTAL プログラムの作成](#)
- [pTAL によるコーディング](#)
- [IBM WebSphere MQ Client for HP Integrity NonStop Server 用の JMS プログラムの作成。](#)

新規メッセージ

HP Integrity NonStop Server 上の IBM WebSphere MQ クライアントの新規メッセージは以下のとおりです。

- AMQ5000-5999: インストール可能サービス
 - [AMQ5370](#)
 - [AMQ5371](#)

- [AMQ5372](#)
- [AMQ5373](#)
- [AMQ5374](#)
- [AMQ5375](#)
- [AMQ5376](#)
- [AMQ5377](#)
- [AMQ5378](#)
- [AMQ5379](#)
- [AMQ5380](#)
- [AMQ5390](#)
- [AMQ5391](#)
- [AMQ5392](#)
- [AMQ5393](#)
- [AMQ5394](#)
- [AMQ5395](#)
- [AMQ5396](#)
- [AMQ5397](#)
- [AMQ5398](#)
- [AMQ5399](#)
- AMQ9000-9999: リモート
 - [AMQ9816](#)
 - [AMQ9817](#)
 - [AMQ9818](#)
 - [AMQ9819](#)
 - [AMQ9820](#)
 - [AMQ9821](#)
 - [AMQ9823](#)
 - [AMQ9824](#)

変更された API 理由コード

以下の既存の API 理由コードに HP Integrity NonStop Server が含まれるようになりました。

- [2354 \(0932\) \(RC2354\): MQRC_UOW_ENLISTMENT_ERROR](#)
- [2355 \(0933\) \(RC2355\): MQRC_UOW_MIX_NOT_SUPPORTED](#)
- [2072 \(0818\) \(RC2072\): MQRC_SYNCPOINT_NOT_AVAILABLE](#)
- [2003 \(07D3\) \(RC2003\): MQRC_BACKED_OUT](#)

サンプル

サンプル・プログラムで示される手法については、[Samples for IBM WebSphere MQ client for HP Integrity NonStop Server](#) を参照してください。

トラブルシューティングとサポート

HP Integrity NonStop Server 上の IBM WebSphere MQ クライアントのトラブルシューティングおよびサポート情報については、以下のトピックを参照してください。

- [IBM WebSphere MQ Client for HP Integrity NonStop Server のトラブルシューティング](#)
- [HP Integrity NonStop Server のエラー・ログ](#)
- [HP Integrity NonStop Server でのトレースの使用](#)
- [FFST \(2\): IBM WebSphere MQ \(HP Integrity NonStop Server の場合\)](#)

IBM WebSphere MQ Version 7.5 の変更点

キュー・マネージャーを IBM WebSphere MQ Version 7.5 にアップグレードする前に、変更のリストを注意深く確認してください。Version 7.5 へのシステムの移行を開始する前に、既存のアプリケーション、スクリプト、およびプロシージャの変更を計画する必要があるかどうかを判断してください。

以下のリンクは、製品資料の『マイグレーションおよびアップグレード』のセクション内の情報に対応しています。新機能と、既存のアプリケーション、管理プロシージャ、および管理スクリプトに影響しない変更は、ここにはリストされていません。18 ページの『IBM WebSphere MQ Version 7.5 の新機能』を参照してください。

[バージョン、リリース、および保守レベル別の変更点のリスト](#)

- [バージョン 7.1 からバージョン 7.5 での変更点](#)

関連概念

18 ページの『IBM WebSphere MQ Version 7.5 の新機能』

IBM WebSphere MQ Version 7.5 の主な新機能について学習します。

24 ページの『IBM WebSphere MQ Version 7.5 フィックスパックの変更点』

このセクションでは、Version 7.5 フィックスパックで機能およびリソースに加えられた変更点について説明します。

IBM WebSphere MQ Version 7.5 フィックスパックの変更点

このセクションでは、Version 7.5 フィックスパックで機能およびリソースに加えられた変更点について説明します。

- **V7.5.0.1** 25 ページの『Version 7.5.0, Fix Pack 1: MQTT over WebSockets のサポート』
- **V7.5.0.4** 26 ページの『Version 7.5.0, Fix Pack 4: AMQ_DISABLE_CLIENT_AMS を使用してクライアントで IBM WebSphere MQ Advanced Message Security を無効にします。』
- **V7.5.0.5** 26 ページの『Version 7.5.0, Fix Pack 5: mqclient.ini ファイルを使用して、クライアントで IBM WebSphere MQ Advanced Message Security を無効にします。』
- **Windows V7.5.0.6** 27 ページの『Version 7.5.0, Fix Pack 6: 管理対象 .NET クライアント・アプリケーションのユーザー ID とパスワード』
- **V7.5.0.6** 27 ページの『Version 7.5.0, Fix Pack 6: 非推奨の CipherSpecs』
- **V7.5.0.6** 27 ページの『Version 7.5.0, Fix Pack 6: IBM WebSphere MQ Managed File Transfer の保守容易性の機能拡張』
- **V7.5.0.6** 27 ページの『Version 7.5.0, Fix Pack 6: IBM WebSphere MQ Managed File Transfer の新しいエージェント・プロパティ failTransferOnFirstFailure』
- **V7.5.0.7** 28 ページの『Version 7.5.0, Fix Pack 7: 非推奨の CipherSpecs』
- **Windows V7.5.0.7** 28 ページの『Version 7.5.0, Fix Pack 7: Oracle 12c データベースへの JDBC 接続のサポート』
- **Windows V7.5.0.7** 28 ページの『Version 7.5.0, Fix Pack 7: MQCNO_CLIENT_BINDING および MQCNO_LOCAL_BINDING での .NET の使用に関する制限の削除』
- **V7.5.0.8** 28 ページの『Version 7.5.0, Fix Pack 8: 配布リスト内のトピック別名キューの使用に関する制限』

- **V7.5.0.8** 28 ページの『Version 7.5.0, Fix Pack 8: GSKit バージョンの更新』
- **V7.5.0.8** 29 ページの『Version 7.5.0, Fix Pack 8: 非推奨の CipherSpecs』
- **V7.5.0.8** 29 ページの『Version 7.5.0, Fix Pack 8: JmsConstants インターフェースに追加された新規規定数 JMS IBM SUBSCRIPTION_USER_DATA』
- **V7.5.0.8** 29 ページの『Version 7.5.0, Fix Pack 8: JMS 例外リスナーの更新』
- **V7.5.0.8** 29 ページの『Version 7.5.0, Fix Pack 8: IBM WebSphere MQ classes for JMS ObjectMessage 内のクラス名許可リスティングのサポート』
- **V7.5.0.8** 30 ページの『Version 7.5.0, Fix Pack 8: IBM WebSphere MQ Managed File Transfer の新しいエージェント・プロパティ additionalWildcardSandboxChecking』
- **V7.5.0.8** 30 ページの『Version 7.5.0, Fix Pack 8: IBM WebSphere MQ Managed File Transfer fteCleanAgent コマンドの動作の変更』
- **V7.5.0.9** 30 ページの『Version 7.5.0, Fix Pack 9: 新しい環境変数 AMQ_SSL_LDAP_SERVER_VERSION』
- **V7.5.0.9** 30 ページの『Version 7.5.0, Fix Pack 9: IBM WebSphere MQ classes for JMS のサンプル構成ファイル』
- **V7.5.0.9** 30 ページの『Version 7.5.0, Fix Pack 9: dmpmqcfg コマンドの変更点』
- **V7.5.0.9** 31 ページの『Version 7.5.0, Fix Pack 9: strmqm コマンドの変更点』
- **V7.5.0.9** 31 ページの『runmqras ユーティリティの機能拡張』
- **V7.5.0.9** 32 ページの『Version 7.5.0, Fix Pack 9: メッセージからファイルへの転送のために復元される転送 ID とグループ ID 属性値の比較検査』
- **V7.5.0.9** 31 ページの『Version 7.5.0, Fix Pack 9: 活動化仕様の targetClientMatching プロパティのためのリソース・アダプターの更新』
- **V7.5.0.9** 31 ページの『Version 7.5.0, Fix Pack 9: runmqras コマンドの新しい -caseno パラメーター』
- **V7.5.0.9** 31 ページの『Version 7.5.0, Fix Pack 9: キュー・マネージャーのクラスター化コンポーネントで使用される内部 MQ コマンド・メッセージ内の破損した文字データに関するエラー処理の変更』
- **V7.5.0.9** **Windows** 32 ページの『Version 7.5.0, Fix Pack 9: Windows での Microsoft Visual Studio 2013 C/C++ ランタイムの追加』
- **V7.5.0.9** 32 ページの『Version 7.5.0, Fix Pack 9: メッセージからファイルへの転送のために復元される転送 ID とグループ ID 属性値の比較検査』
- **V7.5.0.9** 32 ページの『Version 7.5.0, Fix Pack 9: メッセージからファイルへの転送のために復元される転送 ID とグループ ID 属性値の比較検査』
- **V7.5.0.9** 32 ページの『Version 7.5.0, Fix Pack 9 および APAR IT26482: IBM WebSphere MQ classes for JMS がクラスター・キューの BackoutThreshold および BackoutRequeueQName を照会するために必要な権限の変更』

Version 7.5.0, Fix Pack 1: MQTT over WebSockets のサポート

V7.5.0.1

新しい通信プロトコル・パラメーター (PROTOCOL) が MQTT チャネル定義 (DEFINE CHANNEL (MQTT)) に追加されました。

- パラメーターが MQTTV3 に設定されている場合、チャンネルは、MQ Telemetry Transfer プロトコルのバージョン 3 を使用するクライアントからの接続のみを受け入れます。IBM WebSphere MQ Version 7.5.0、フィックスパック 1 より前のバージョンでは、このプロトコルだけがサポートされていました。
- パラメーターが HTTP に設定されている場合、チャンネルはページの HTTP 要求、または IBM WebSphere MQ Telemetry への WebSockets 接続のみを受け入れます。
- パラメーターが MQTTV3, HTTP に設定されている場合、チャンネルは、どちらのプロトコルを使用するクライアントからの接続も受け入れます。これは、IBM WebSphere MQ Version 7.5.0、フィックスパック 1 以降のバージョンで作成された新規 MQTT チャンネルのデフォルトの動作です。

詳しくは、[SSL を介した JavaScript 用の MQTT メッセージング・クライアントと WebSockets の接続を参照してください](#)。

クライアントが SSL を使用して MQTT チャンネルに接続する場合、パラメーター SSLCAUTH は、IBM WebSphere MQ がクライアントからの証明書が必要とするかどうかを決定します ([DEFINE CHANNEL \(MQTT\)](#) を参照)。IBM WebSphere MQ Version 7.5.0、フィックスパック 1 より前では、MQTT チャンネルの場合、このパラメーターは REQUIRED または OPTIONAL のいずれかです。

- REQUIRED の場合、IBM WebSphere MQ はクライアントからの証明書を要求し、クライアントは有効な証明書を提供する必要があります。
- OPTIONAL の場合、IBM WebSphere MQ がクライアントからの証明書を要求しますが、クライアントはそれを提供する必要がありません。クライアント接続は、クライアントが有効な証明書を提供した場合、またはクライアントが証明書を提供しない場合に許可されます。クライアントが無効な証明書を提供した場合にのみ、クライアント接続が不許可になります。

IBM WebSphere MQ Version 7.5.0、フィックスパック 1 以降では、MQTT チャンネルの場合、パラメーター SSLCAUTH を NEVER に設定できます。NEVER は、IBM WebSphere MQ がクライアントからの証明書を要求しないことを意味します。新しい値は、JavaScript の MQTT メッセージング・クライアントを使用するクライアントのサポートの一部として追加されました。これは、クライアント証明書の要求をプロトコル・エラーとして扱う一部の Web ブラウザーの動作に対応しています。

Version 7.5.0, Fix Pack 4: AMQ_DISABLE_CLIENT_AMS を使用してクライアントで IBM WebSphere MQ Advanced Message Security を無効にします。

V 7.5.0.4

Version 7.5.0, Fix Pack 4 以降では、環境変数 AMQ_DISABLE_CLIENT_AMS を使用して、IBM WebSphere MQ classes for JMS および IBM WebSphere MQ classes for Java クライアント内の IBM WebSphere MQ Advanced Message Security (AMS) を無効にし、旧バージョンの製品で実行されているキュー・マネージャーに接続する際のエラーを防止します。あるいは、Version 7.5.0, Fix Pack 4 以降、Java システム・プロパティー com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS を使用して、クライアントで AMS を無効にすることもできます。

詳しくは、[クライアントでの IBM WebSphere MQ Advanced Message Security の無効化を参照してください](#)。

Version 7.5.0, Fix Pack 5: mqclient.ini ファイルを使用して、クライアントで IBM WebSphere MQ Advanced Message Security を無効にします。

V 7.5.0.5

Version 7.5.0, Fix Pack 4 以降、構成変数 AMQ_DISABLE_CLIENT_AMS を設定することにより、クライアントで IBM WebSphere MQ Advanced Message Security (AMS) を無効にして、旧バージョンの製品で実行されているキュー・マネージャーに接続するときにエラーが発生しないようにすることができます。Version 7.5.0, Fix Pack 5 以降、mqclient.ini ファイルの **Security** スタンザの下にある DisableClientAMS プロパティーを使用して、AMS を無効にすることもできます。詳しくは、[クライアントでの IBM WebSphere MQ Advanced Message Security の無効化を参照してください](#)。

Version 7.5.0, Fix Pack 6: 管理対象 .NET クライアント・アプリケーションのユーザー ID とパスワード

Windows V 7.5.0.6

IBM WebSphere MQ Version 7.5.0、フィックスパック 6 以降、管理対象 .NET クライアント・アプリケーションで指定されたユーザー ID とパスワードは、クライアント・セキュリティー出口に渡される IBM WebSphere MQ .NET MQChannelDefinition クラスで設定されます。詳しくは、『[IBM WebSphere MQ .NET でのチャンネル出口の使用](#)』を参照してください。

Version 7.5.0, Fix Pack 6: 非推奨の CipherSpecs

IBM WebSphere MQ Version 7.5.0、フィックスパック 6 以降、以下の CipherSpec が非推奨になりました。

- DES_SHA_EXPORT
 - ▶ Windows ▶ Linux ▶ UNIX DES_SHA_EXPORT1024
 - ▶ Windows ▶ Linux ▶ UNIX FIPS_WITH_DES_CBC_SHA
- NULL_MD5
- NULL_SHA
- TLS_RSA_WITH_DES_CBC_SHA
- ECDHE_ECDSA_NULL_SHA256
 - ▶ Windows ▶ Linux ▶ UNIX ECDHE_RSA_NULL_SHA256
 - ▶ Windows ▶ Linux ▶ UNIX TLS_RSA_WITH_NULL_NULL
- TLS_RSA_WITH_NULL_SHA256

詳しくは、[非推奨 CipherSpecs](#) を参照してください。

Version 7.5.0, Fix Pack 6: IBM WebSphere MQ Managed File Transfer の保守容易性の機能拡張

V 7.5.0.6

IBM WebSphere MQ Version 7.5.0、フィックスパック 6 以降では、IBM WebSphere MQ Managed File Transfer で以下の点に変更されています。

- `installation.properties` ファイルの `commandMessagePriority` プロパティーのデフォルト値が 8 に変更されました。詳しくは、[installation.properties ファイル](#) を参照してください。
- `agent.properties` ファイル内の `logTransferRecovery` プロパティーのデフォルト値が TRUE に変更されました。詳しくは、[agent.properties ファイル](#) を参照してください。
- `fteShowAgentDetails` コマンドで `-d` パラメーターを使用するときに、`installation.properties` ファイル内の `enableFunctionalFixPack` プロパティーを設定する必要がなくなりました。詳しくは、[fteShowAgentDetails](#) を参照してください。
- エージェントでリカバリー不能エラーが発生すると、First Failure Data Capture (FDC) が生成されます。

Version 7.5.0, Fix Pack 6: IBM WebSphere MQ Managed File Transfer の新しいエージェント・プロパティー `failTransferOnFirstFailure`

V 7.5.0.6 distributed

IBM WebSphere MQ Version 7.5.0、フィックスパック 6 以降では、`agent.properties` ファイル内の `failTransferOnFirstFailure` プロパティーを設定することにより、管理対象転送の転送項目が失敗すると直ちにその管理対象転送が失敗するように、エージェントを構成できます。詳しくは、[agent.properties ファイル](#) を参照してください。

Version 7.5.0, Fix Pack 7: 非推奨の CipherSpecs

IBM WebSphere MQ Version 7.5.0、フィックスパック 7 以降、以下の CipherSpec が非推奨になりました。

- RC2_MD5_EXPORT
- RC4_MD5_EXPORT
- RC4_MD5_US
- RC4_SHA_US
-    RC4_56_SHA_EXPORT1024
-    ECDHE_ECDSA_RC4_128_SHA256
-    ECDHE_RSA_RC4_128_SHA256
-    TLS_RSA_WITH_RC4_128_SHA256

詳しくは、[非推奨 CipherSpecs](#) を参照してください。

Version 7.5.0, Fix Pack 7: Oracle 12c データベースへの JDBC 接続のサポート



IBM WebSphere MQ Version 7.5.0、フィックスパック 7 以降、Oracle 12c データベースへの JDBC 接続をサポートするために、新規ファイル `jdbcora12.d11` が IBM WebSphere MQ Windows サーバー・インストール・イメージに付属しています ([Windows での JTA/JDBC 調整の構成](#)を参照)。

Version 7.5.0, Fix Pack 7: MQCNO_CLIENT_BINDING および MQCNO_LOCAL_BINDING での .NET の使用に関する制限の削除



IBM WebSphere MQ Version 7.5.0、フィックスパック 7 以降、Microsoft Windows Communication Foundation (WCF) 用の IBM WebSphere MQ カスタム・チャンネルが更新され、クライアントのみのインストール済み環境から実行する場合に正しいクライアント接続構成が使用されるようになりました。詳しくは、『[MQCONNX 呼び出しを使用したキュー・マネージャーへの接続](#)』を参照してください。

Version 7.5.0, Fix Pack 8: 配布リスト内のトピック別名キューの使用に関する制限



配布リストでは、トピック・オブジェクトを指す別名キューの使用はサポートされていません。Version 7.5.0, Fix Pack 8 以降では、別名キューが配布リスト内のトピック・オブジェクトを参照していると、IBM WebSphere MQ から `MQRC_ALIAS_BASE_Q_TYPE_ERROR` が返されます。

Version 7.5.0, Fix Pack 8: GSKit バージョンの更新



GSKit バージョンが更新されました。新しいバージョンの GSKit では、鍵データベースのパスワードを隠しておくための `.sth` ファイルを生成する時に使用する `stash` ファイル・フォーマットが変更されています。このバージョンの GSKit で生成される `stash` ファイルは、以前のバージョンの GSKit で読み取ることはできません。Version 7.5.0, Fix Pack 8 以降で生成される `stash` ファイルが使用しているアプリケーションおよびその他の IBM WebSphere MQ インストール済み環境と互換性を持つようにするために、互換性のあるバージョンの GSKit が含まれる IBM WebSphere MQ のバージョンに更新する必要があります。IBM WebSphere MQ Version 7.5 の場合、これは Version 7.5.0, Fix Pack 8 です。

使用しているアプリケーションやその IBM WebSphere MQ インストール環境を更新できない場合は、旧バージョンとの間に互換性がある `stash` ファイル・フォーマットを要求できます。-`stash` または -`stashpw` オプションを指定して `runmqakm` または `runmqckm` コマンドを使用する場合は、-`v1stash` コマンド・ライン・パラメーターを含めます。以前のバージョンと互換性のある `stash` ファイルを生成する際に iKeyman GUI を使用することはできません。

Version 7.5.0, Fix Pack 8: 非推奨の CipherSpecs

V7.5.0.8

Version 7.5.0, Fix Pack 8 以降では、以下の CipherSpec が非推奨になりました。

-    FIPS_WITH_3DES_EDE_CBC_SHA
- TRIPLE_DES_SHA_US
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
-    ECDHE_ECDSA_3DES_EDE_CBC_SHA256
-    ECDHE_RSA_3DES_EDE_CBC_SHA256

詳しくは、[非推奨 CipherSpecs](#) を参照してください。

Version 7.5.0, Fix Pack 8: JmsConstants インターフェースに追加された新規定数 JMS_IBM_SUBSCRIPTION_USER_DATA

V7.5.0.8

Version 7.5.0, Fix Pack 8 以降、IBM WebSphere MQ classes for JMS が更新され、MQPS フォルダを持つ RFH2 ヘッダーが含まれたメッセージがキューからコンシュームされる際に、Sud キーに関連付けられた値が存在する場合、IBM WebSphere MQ classes for JMS アプリケーションに返される JMS メッセージ・オブジェクトにその値が String プロパティとして追加されるようになりました。アプリケーションがこのプロパティをメッセージから取得できるようにするために、新しい定数

JMS_IBM_SUBSCRIPTION_USER_DATA が JmsConstants インターフェースに追加されました。この新しいプロパティは、サブスクリプション・ユーザー・データを取得するために、`javax.jms.Message.getStringProperty(java.lang.String)` メソッドで使用できます。詳しくは、[ユーザー・サブスクリプション・データの取得および DEFINE SUB](#) を参照してください。

Version 7.5.0, Fix Pack 8: JMS 例外リスナーの更新

V7.5.0.8

IBM WebSphere MQ Version 7.5.0、フィックスパック 8 以降、JMS MessageListener および JMS ExceptionListener を構成する現行 JMS アプリケーションの動作を維持するために、IBM WebSphere MQ classes for JMS for Version 7.5 の ASYNC_EXCEPTIONS JMS ConnectionFactory プロパティのデフォルト値が ASYNC_EXCEPTIONS_CONNECTIONBROKEN に変更されました。結果として、デフォルトでは、接続切断のエラー・コードに対応する例外だけがアプリケーションの JMS ExceptionListener に送信されます。メッセージを同期的または非同期的にコンシュームすると、接続切断エラーが例外リスナーに送信されます。

IBM WebSphere MQ classes for JMS for Version 7.5 も更新され、非同期メッセージ・コンシューマーへのメッセージ配信中に発生する非接続切断エラーに関連する JMS 例外が、アプリケーションによって使用される JMS ConnectionFactory の ASYNC_EXCEPTIONS プロパティの値が ASYNC_EXCEPTIONS_ALL に設定されている場合に、登録済みの ExceptionListener に配信されるようになりました。

詳しくは、『[JMS: バージョン 7.5 の例外リスナーの変更点](#)』および『[IBM WebSphere MQ classes for JMS の例外](#)』を参照してください。

Version 7.5.0, Fix Pack 8: IBM WebSphere MQ classes for JMS ObjectMessage 内の クラス名許可リスティングのサポート

V7.5.0.8

IBM WebSphere MQ Version 7.5.0、フィックスパック 8 以降、IBM WebSphere MQ classes for JMS は、JMS ObjectMessage インターフェースの実装におけるクラスの許可リスト登録をサポートします。許可リストには、`ObjectMessage.setObject()` でシリアライズできる Java クラスと、`ObjectMessage.getObject()` でデシリアライズできる Java クラスを定義します。

詳しくは、[JMS でのクラス名の許可リスティング ObjectMessage](#) および [IBM WebSphere MQ classes for JMS アプリケーションの実行](#)を参照してください。

Version 7.5.0, Fix Pack 8: IBM WebSphere MQ Managed File Transfer の新しいエージェント・プロパティ `additionalWildcardSandboxChecking`

▼7.5.0.8

IBM WebSphere MQ Version 7.5.0、フィックスパック 8 以降、エージェントによるファイル転送の宛先/送信元の場所を制限するために、エージェントにユーザー・サンドボックスまたはエージェント・サンドボックスを構成した場合に、`additionalWildcardSandboxChecking` プロパティを `true` に設定して、そのエージェントのワイルドカード転送に対する追加検査を行うことを指定できます。詳しくは、『[ワイルドカード転送の追加検査](#)』および『[agent.properties ファイル](#)』を参照してください。

Version 7.5.0, Fix Pack 8: IBM WebSphere MQ Managed File Transfer `fteCleanAgent` コマンドの動作の変更

▼7.5.0.8

IBM WebSphere MQ Version 7.5.0、フィックスパック 8 以降、`fteCleanAgent` コマンドは、エージェント名を指定するだけでなく、適切なパラメーターを渡して、クリアする IBM WebSphere MQ Managed File Transfer 状態を指定しなければならないように更新されました。この動作の変更により、`fteCleanAgent` で、指定したエージェントの進行中の転送および保留中の転送、リソース・モニター定義、およびスケジュールされた転送定義がすべてデフォルトでクリアされることがなくなりました。

必要に応じて、`command.properties` ファイルの新しい `failCleanAgentWithNoArguments` プロパティを `false` に設定して、`fteCleanAgent` の以前の動作に戻すことができます。

詳しくは、『[fteCleanAgent \(Managed File Transfer Agent のクリーンアップ\)](#)』および『[command.properties ファイル](#)』を参照してください。

Version 7.5.0, Fix Pack 9: 新しい環境変数 `AMQ_SSL_LDAP_SERVER_VERSION`

▼7.5.0.9

Version 7.5.0, Fix Pack 9 以降、環境変数 `AMQ_SSL_LDAP_SERVER_VERSION` を設定できるようになりました。この環境変数により、より高いバージョンの LDAP 証明書取り消しリスト (CRL) サーバーに対して証明書が検証されるときに、IBM WebSphere MQ が必要な Lightweight Directory Access Protocol (LDAP) バージョン (LDAP v2 または LDAP v3) を GSKit に渡すことができます。詳しくは、『[環境変数](#)』を参照してください。

Version 7.5.0, Fix Pack 9: IBM WebSphere MQ classes for JMS のサンプル構成ファイル

▼7.5.0.9

Version 7.5.0, Fix Pack 9 以降、サンプル構成ファイル `mqjava.config` が IBM WebSphere MQ classes for JMS インストール・ディレクトリーの `bin` サブディレクトリーに提供されています。詳しくは、『[IBM MQ classes for Java 構成ファイル](#)』を参照してください。

Version 7.5.0, Fix Pack 9: `dmpmqcfg` コマンドの変更点

▼7.5.0.9

Version 7.5.0, Fix Pack 9 以降、コマンド行スイッチ `-w` が `dmpmqcfg` に追加されました。このスイッチを使用すると、`dmpmqcfg` がコマンド・サーバーからの応答メッセージを待つ時間を秒単位で増やすことができます。

詳しくは、『[dmpmqcfg](#)』を参照してください。

Version 7.5.0, Fix Pack 9: strmqm コマンドの変更点

V 7.5.0.9

Version 7.5.0, Fix Pack 9 以降の場合、**strmqm** コマンドによって、キュー・マネージャーを完全に開始する前に `qm.ini` ファイル内の CHANNELS スタンザと SSL スタンザの構文が検査されます。これにより、**strmqm** は、`qm.ini` ファイルにエラーが含まれていることを検出した場合に、問題を容易に判別して迅速に修正できるようになります。詳しくは、[strmqm](#) を参照してください。

runmqras ユーティリティの機能拡張

V 7.5.0.9

Version 7.5.0, Fix Pack 9 以降、**runmqras** ユーティリティには、以下の機能拡張がなされています。

- **AIX** **Solaris** **Linux** 環境変数情報がデフォルトで取得されます。
- **distributed** キュー・マネージャー・データ・ディレクトリーのリストがデフォルトで取得されます。
- **runmqras** コマンドに次の 2 つのセクションが追加されています。
 - **Linux** **UNIX** IBM WebSphere MQ プロセス・リソース使用情報を収集するための leak セクション。
 - **distributed** **fteRas** コマンドによって取得されるデータを取り込む mft セクション。

詳しくは、[runmqras](#) を参照してください。

Version 7.5.0, Fix Pack 9: 活動化仕様の targetClientMatching プロパティーのためのリソース・アダプターの更新

V 7.5.0.9

IBM WebSphere MQ JCA リソース・アダプター (MQ-RA) が更新され、アクティベーション・スペックに対して **targetClientMatching** プロパティーを構成できるようになりました。プロパティーを構成することで、要求メッセージに MQRFH2 ヘッダーが含まれないときに応答メッセージに MQRFH2 ヘッダーを含めることができます。これは、応答メッセージでアプリケーションが定義するメッセージ・プロパティーがメッセージの送信時に組み込まれることを意味します。詳しくは、[活動化仕様の targetClientMatching プロパティーの構成](#)を参照してください。

Version 7.5.0, Fix Pack 9: runmqras コマンドの新しい -caseno パラメーター

V 7.5.0.9

-caseno パラメーターは **-pmrno** とほぼ同等ですが、有効な Salesforce ケース番号を指定できる点が異なります。**-caseno** と **-pmrno** はともにオプション・パラメーターですが、両方一緒に指定することはできません。詳しくは、[runmqras \(IBM MQ 診断情報の収集\)](#) を参照してください。

Version 7.5.0, Fix Pack 9: キュー・マネージャーのクラスター化コンポーネントで使用する内部 MQ コマンド・メッセージ内の破損した文字データに関するエラー処理の変更

V 7.5.0.9

Version 7.5.0, Fix Pack 9 以降、キュー・マネージャーのクラスターリング・コンポーネントによって使用される内部 MQ コマンド・メッセージ内の破損した文字データに関連するエラーをキュー・マネージャーが処理する方法が変更されました。変換不能な文字が含まれている可能性のあるフィールドがブランクになり、変換が再試行されます。変換が成功すると、ブランクの説明とともに更新が保管され、エラー・メッセージ AMQ6174 と AMQ9880 が書き込まれて、管理者に通知されます。メッセージが引き続き変換不能である場合は、それがデッド・レター・キューに移され、エラー・メッセージが書き込まれます。

Version 7.5.0, Fix Pack 9: Windows での Microsoft Visual Studio 2013 C/C++ ランタイムの追加

V 7.5.0.9

Windows

Version 7.5.0, Fix Pack 9 以降、IBM WebSphere MQ の一部の要素は Microsoft Visual Studio 2013 C/C++ (VS2013) ランタイムを必要とします。これらのランタイムは、IBM WebSphere MQ Version 7.5.0、フィックスパック 9 をフレッシュ・インストールするとインストールされます。

VS2013 ランタイムがマシン上に存在しない場合、新しいクライアントをインストールするか、クライアントを 7.5.0.9 保守レベルに更新するか、7.5.0.9 フィックスパックを適用すると、ランタイムがインストールされます。ほとんどの場合、このプロセスにユーザーのアクションは不要ですが、サーバー・フィックスパックの適用時にインストールが失敗した場合に備えて、ランタイムの手動インストールを指定するために必要な環境変数が用意されています。詳しくは、『[サーバー・フィックスパックでの Microsoft Visual Studio 2013 C/C++ ランタイムのインストール](#)』を参照してください。

Version 7.5.0, Fix Pack 9: メッセージからファイルへの転送のために復元される転送 ID とグループ ID 属性値の比較検査

V 7.5.0.9

Version 7.5.0, Fix Pack 9 以降、IBM WebSphere MQ Managed File Transfer が更新され、転送要求 XML ペイロード内の転送 ID と groupId 属性の値の比較検査 (以前に Version 7.5.0, Fix Pack 6 で APAR IT18213 によって削除されたもの) が復元されるようになりました。比較した 2 つの ID が等しい場合、ソース・エージェントはその ID を、メッセージからファイルへの転送のための入力キューに対して行われる 1 回目の MQGET の試行で、メッセージ ID のマッチ・オプション (グループ ID のマッチ・オプションと対照) として使用します。詳しくは、『[メッセージからファイルへのデータ転送](#)』を参照してください。

Version 7.5.0, Fix Pack 9 および APAR IT26482: IBM WebSphere MQ classes for JMS がクラスター・キューの BackoutThreshold および BackoutQueueName を照会するために必要な権限の変更

V 7.5.0.9

Version 7.5.0, Fix Pack 9 および APAR IT26482 の暫定修正の場合、IBM WebSphere MQ classes for JMS が更新され、クラスター・キューの **BackoutThreshold** および **BackoutQueueName** を照会するために必要なのは照会アクセスのみになるようになりました。その他のすべてのバージョンでは、参照と取得のアクセス権限も必要です。詳細については、[IBM MQ classes for JMS でのポイズン・メッセージの処理](#)を参照してください。

関連概念

18 ページの『[IBM WebSphere MQ Version 7.5 の新機能](#)』

IBM WebSphere MQ Version 7.5 の主な新機能について学習します。

24 ページの『[IBM WebSphere MQ Version 7.5 の変更点](#)』

関連情報

[IBM WebSphere MQ のシステム要件](#)

[IBM MQ、WebSphere MQ、および MQSeries の製品 README の Web ページ](#)

[WebSphere MQ の推奨フィックス](#)

[WebSphere MQ の計画されたメンテナンス・リリースの日付](#)

以前のバージョンの新機能および変更点

このセクションでは、IBM WebSphere MQ Version 7.5 より前の製品バージョンにおける新機能、およびこれらのバージョンで行われた機能およびリソースの変更 (固定化、非推奨、削除など) に関する情報へのリンクを提供します。

以前のバージョンの製品における新機能および変更内容については、そのバージョンの製品資料の該当するセクションを参照してください。

IBM WebSphere MQ Version 7.1

- [IBM WebSphere MQ バージョン 7.1 の新機能](#)
- [IBM WebSphere MQ バージョン 7.1 の変更点](#)
- [IBM WebSphere MQ バージョン 7.1 フィックスパックの変更点](#)

IBM WebSphere MQ Version 7.0.1 以前

IBM Documentation の外部で資料が提供されている旧バージョンの製品については、[旧バージョンの WebSphere MQ の Documentation](#) を参照してください。

関連概念

18 ページの『[IBM WebSphere MQ Version 7.5 の新機能](#)』

IBM WebSphere MQ Version 7.5 の主な新機能について学習します。

24 ページの『[IBM WebSphere MQ Version 7.5 の変更点](#)』

24 ページの『[IBM WebSphere MQ Version 7.5 フィックスパックの変更点](#)』

このセクションでは、Version 7.5 フィックスパックで機能およびリソースに加えられた変更点について説明します。

[WebSphere MQ マイグレーションの概要](#)

以前の IBM WebSphere MQ 資料と現在の資料の構造の対応

IBM Documentation の情報は、一連の汎用アクティビティに従って構造化されています。例えば、インストール、開発、管理、構成、保護などの操作があります。以前の資料（「MQ ブック」）の構造は少し異なっていました。このセクションでは、これらの以前の IBM WebSphere MQ 資料と現在の情報構造との間のマッピングについて説明します。

スタートアップ・ガイド (AIX)

このセクションでは、古い「スタートアップ・ガイド (AIX)」ブックから新しい製品資料の構造へのマッピングを示します。

- [AIX: 新しいバージョンへのマイグレーションの計画](#)
- [要件のチェック](#)
- [システムの準備](#)
- [AIX での IBM WebSphere MQ サーバーのインストール](#)
- [サーバー・インストールの検査](#)
- [AIX システムでの IBM WebSphere MQ クライアントのインストール](#)
- [クライアントのインストールの検査](#)
- [AIX: IBM WebSphere MQ Version 7.5 での保守レベル・アップグレードの適用](#)
- [AIX での IBM WebSphere MQ のアンインストール](#)

スタートアップ・ガイド (HP-UX)

このセクションでは、古い「スタートアップ・ガイド (HP-UX)」ブックから新しい製品資料の構造へのマッピングを示します。

- [HP-UX: 新しいバージョンへのマイグレーションの計画](#)
- [要件のチェック](#)
- [システムの準備](#)
- [HP-UX での IBM WebSphere MQ サーバーのインストール](#)
- [サーバー・インストールの検査](#)
- [HP-UX システムでの IBM WebSphere MQ クライアントのインストール](#)

- [クライアントのインストールの検査](#)
- [HP-UX: IBM WebSphere MQ Version 7.5 での保守レベル・アップデートの適用](#)
- [HP-UX での IBM WebSphere MQ のアンインストール](#)

Linux のクイック・スタート

このセクションでは、古い「Linux のクイック・スタート」ブックから現在の製品資料の構造へのマッピングを示します。

- [Linux: IBM WebSphere MQ Version 7.0.1 から IBM WebSphere MQ Version 7.5 からのマイグレーションの計画](#)
- [要件のチェック](#)
- [システムの準備](#)
- [Linux での IBM WebSphere MQ サーバーのインストール](#)
- [サーバー・インストールの検査](#)
- [Linux での WebSphere MQ クライアントのインストール](#)
- [クライアントのインストールの検査](#)
- [Linux: IBM WebSphere MQ Version 7.5 での保守レベル・アップデートの適用](#)
- [Linux での IBM WebSphere MQ のアンインストール](#)

スタートアップ・ガイド (Solaris)

このセクションでは、古い「スタートアップ・ガイド (Solaris)」ブックから新しい製品資料の構造へのマッピングを示します。

- [Solaris: 新しいバージョンへのマイグレーションの計画](#)
- [要件のチェック](#)
- [システムの準備](#)
- [Solaris での IBM WebSphere MQ サーバーのインストール](#)
- [サーバー・インストールの検査](#)
- [Solaris での IBM WebSphere MQ クライアントのインストール](#)
- [クライアントのインストールの検査](#)
- [Solaris: IBM WebSphere MQ Version 7.5 での保守レベル・アップデートの適用](#)
- [Solaris での IBM WebSphere MQ のアンインストール](#)

スタートアップ・ガイド (Windows)

このセクションでは、古い「スタートアップ・ガイド (Windows)」ブックから新しい製品資料の構造へのマッピングを示します。

- [Windows: 新しいバージョンへのマイグレーションの計画](#)
- [要件のチェック](#)
- [システムの準備](#)
- [Windows での IBM WebSphere MQ サーバーのインストール](#)
- [サーバー・インストールの検査](#)
- [Windows システムでの IBM WebSphere MQ クライアントのインストール](#)
- [クライアントのインストールの検査](#)
- [Windows: IBM WebSphere MQ Version 7.5 での保守レベル・アップグレードの適用](#)
- [Solaris での IBM WebSphere MQ のアンインストール](#)

アプリケーション・プログラミング・ガイド

このセクションでは、古い「アプリケーション・プログラミング・ガイド」ブックから新しい製品資料の構造へのマッピングを示します。

- [アプリケーション開発の概念](#)
- [IBM WebSphere MQ アプリケーションの設計](#)
- [キューイング・アプリケーションの作成](#)
- [WebSphere MQ プログラムのサンプル](#)
- [C 言語の例](#)
- [COBOL の例](#)
- [System/390® アセンブラー言語の例 System/390 アセンブラー言語の例](#)
- [WebSphere MQ データ定義ファイル](#)
- [64 ビット・プラットフォームでのコーディング標準](#)

アプリケーション・プログラミング・リファレンス

このセクションでは、古い「アプリケーション・プログラミング・リファレンス」ブックから新しい製品資料の構造へのマッピングを示します。

- [MQI で使用されるデータ・タイプ](#)
- [関数呼び出し](#)
- [オブジェクトの属性](#)
- [戻りコード](#)
- [MQI オプションの妥当性検査に関する規則](#)
- [レポート・オプションおよびメッセージ・フラグ](#)
- [レポート・オプションおよびメッセージ・フラグ](#)
- [データ変換](#)
- [MQRFH2 エlementとして指定されるプロパティ](#)
- [コード・ページ変換](#)

クライアント

このセクションでは、古い「クライアント」ブックから新しい製品資料の構造へのマッピングを示します。

- [132 ページの『IBM WebSphere MQMQI クライアントの概要』](#)
- [134 ページの『IBM WebSphere MQ クライアントのプラットフォームのサポート』](#)
- [IBM WebSphere MQ クライアントのインストール](#)
- [サーバーとクライアント間の接続の構成](#)
- [拡張ランザクション・クライアントの構成](#)
- [クライアントのインストールの検査](#)
- [WebSphere MQ MQI クライアント・セキュリティーのセットアップ](#)
- [118 ページの『チャンネル』](#)
- [MQI チャンネルの定義](#)
- [異なるプラットフォーム上でのサーバー接続およびクライアント接続の定義の作成](#)
- [サーバー側でのサーバー接続およびクライアント接続の定義の作成](#)
- [MQI チャンネル用のチャンネル出口プログラム](#)
- [キュー共用グループへのクライアントの接続](#)
- [構成ファイルを使用したクライアントの構成](#)

- [WebSphere MQ 環境変数の使用](#)
- [クライアント・アプリケーションでのメッセージ・キュー・インターフェース \(MQI\) の使用](#)
- [WebSphere MQ MQI クライアント用のアプリケーションの作成](#)
- [WebSphere MQ MQI クライアント環境でのアプリケーションの実行](#)
- [CICS® および Tuxedo アプリケーションの準備と実行](#)
- [Microsoft Transaction Server アプリケーションの準備と実行](#)
- [WebSphere MQ JMS アプリケーションの準備と実行](#)
- [IBM WebSphere MQ MQI クライアントの問題の解決](#)
- [リポジトリからの、接続前出口を使用した接続定義の参照](#)

定数

このセクションでは、以前の「定数」ブックと新しい製品資料の構造の対応を示します。

- [WebSphere MQ コピー・ファイル、ヘッダー・ファイル、インクルード・ファイル、およびモジュール・ファイル WebSphere MQ コピー・ファイル、ヘッダー・ファイル、インクルード・ファイル、およびモジュール・ファイル](#)
- [定数](#)

相互通信

このセクションでは、以前の「相互通信」ブックと新しい製品資料の構造の対応を示します。

概要

- [50 ページの『相互通信の概念』](#)
- [分散キューイングを使用したアプリケーションの接続](#)
- [ネットワークおよびネットワーク計画](#)
- [WebSphere MQ 分散メッセージング技法](#)
- [分散キュー管理の概要](#)
- [チャンネルの属性](#)
- [構成情報の例](#)

WebSphere MQ for Windows および UNIX プラットフォームでの分散キュー管理

- [Windows、UNIX and Linux プラットフォームにおけるチャンネルのモニターと制御](#)
- [伝送キューの作成](#)
- [チャンネルのトリガー操作](#)
- [チャンネル・プログラム](#)
- [リモート・メッセージングのセキュリティー](#)
- [分散キュー管理に関するその他の考慮事項](#)
- [Windows における通信のセットアップ](#)
- [構成例 - IBM WebSphere MQ for Windows](#)
- [構成例 - IBM WebSphere MQ for AIX](#)
- [構成例 - IBM WebSphere MQ for HP-UX](#)
- [構成例 - IBM WebSphere MQ for Solaris](#)
- [構成例 - IBM WebSphere MQ for Linux](#)
- [分散プラットフォームにおけるメッセージ・チャンネルの計画例](#)

相互通信に関する追加の考慮事項

- [メッセージング・チャンネルのためのチャンネル出口プログラム](#)
- [チャンネル出口呼び出しおよびデータ構造体](#)
- [キュー名の解決](#)

メッセージおよびコード

このセクションでは、古い「メッセージおよびコード」ブックから新しい製品資料の構造へのマッピングを示します。

- [診断メッセージ: AMQ4000-9999](#)
- [API 完了コードと理由コード](#)
- [PCF 理由コード](#)
- [Secure Sockets Layer \(SSL\) および Transport Layer Security \(TLS\) の戻りコード](#)
- [WCF カスタム・チャンネル例外](#)

マイグレーション

Version 7.5 の移行セクションの構造は、Version 7.1 と同じです。

Version 7.5 への移行にはトピックが追加され、Version 7.1 への移行では削除されました。Version 7.1 以前のリリースへのマイグレーションについては、[特定のマイグレーション・パスに関するトピックの入手先](#)を参照してください。

モニター

このセクションでは、古い「モニター」ブックから新しい製品資料の構造へのマッピングを示します。

- [イベント・モニター](#)
- [メッセージ・モニター](#)
- [アカウンティング・メッセージと統計メッセージ](#)
- [リアルタイム・モニター](#)
- [構造データ型](#)
- [イベント・データのオブジェクト属性](#)

プログラマブル・コマンド・フォーマットおよび管理インターフェース

このセクションでは、古い「プログラマブル・コマンド・フォーマットおよび管理インターフェース」ブックから新しい製品資料の構造へのマッピングを示します。

- [プログラマブル・コマンド・フォーマットの紹介](#)
- [WebSphere MQ 管理インターフェース \(MQAI\) の紹介](#)

Publish/Subscribe ユーザーズ・ガイド

このセクションは、古い Publish/Subscribe ユーザーズ・ガイドから新しい製品資料の構造へのマッピングを提供します。

- [WebSphere MQ パブリッシュ/サブスクライブ・メッセージングの紹介](#)
- [分散パブリッシュ/サブスクライブ](#)
- [パブリッシュ/サブスクライブ・アプリケーションの作成](#)
- [パブリッシュ/サブスクライブのセキュリティー](#)
- [パブリッシュ/サブスクライブの Version 6.0 からの移行](#)
- [WebSphere Event Broker および WebSphere Message Broker のパブリッシュ/サブスクライブ・ブローカーの移行](#)

キュー・マネージャー・クラスター

このセクションは、古いキュー・マネージャー・クラスターから新しい製品資料の構造へのマッピングを提供します。

- [64 ページの『クラスターが機能する仕組み』](#)
- [キュー・マネージャー・クラスターの構成](#)
- [IBM WebSphere MQ クラスターの管理](#)
- [クラスターとの間のルーティング・メッセージ](#)
- [クラスターによるワークロードの管理](#)
- [クラスターのセキュリティーの確保](#)
- [MQI とクラスターの処理](#)
- [WebSphere MQ クラスター・コマンド](#)
- [キュー・マネージャー・クラスターに関する問題の解決](#)

スクリプト (MQSC) コマンド・リファレンス

このセクションでは、古い「スクリプト (MQSC) コマンド・リファレンス」ブックから新しい製品資料の構造へのマッピングを示します。

- [スクリプト \(MQSC\) コマンド](#)
- [総称値および特別な意味を持つ文字](#)
- [コマンド・スクリプトの作成](#)
- [123 ページの『IBM WebSphere MQ オブジェクトの命名規則』](#)
- [構文図](#)
- [MQSC コマンド](#)

機密保護

このセクションでは、以前の「セキュリティー」ブックと新しい製品資料の構造の対応を示します。

- [セキュリティー](#)

システム管理ガイド

このセクションでは、古い「システム管理ガイド」ブックから新しい製品資料の構造へのマッピングを示します。

- [42 ページの『IBM WebSphere MQ 技術概説』](#)
- [管理 IBM WebSphere MQ](#)
- [ローカル WebSphere MQ オブジェクトの管理](#)
- [IBM WebSphere MQ Explorer を使用した管理](#)
- [WebSphere MQ Taskbar アプリケーションの使用 \(Windows のみ\)](#)
- [WebSphere MQ 制御コマンド](#)

構成および管理

- [IBM WebSphere MQ およびキュー・マネージャー構成情報の変更](#)
- [ファイル・システム・サポートの計画](#)
- [Windows、UNIX and Linux システムでのセキュリティーのセットアップ](#)
- [トランザクション・サポート](#)
- [WebSphere MQ 送達不能キュー・ハンドラーを使用した、未配布メッセージの処理](#)
- [可用性、リカバリー、および再始動](#)

- [トラブルシューティングとサポート](#)
- [WebSphere MQ と UNIX System V IPC リソース](#)
- [WebSphere MQ および UNIX のプロセス優先順位](#)
- [ユーザー出口、API 出口、および WebSphere MQ インストール可能サービス](#)

.NET の使用

このセクションでは、古い「.Net の使用」ブックから新しい製品資料の構造へのマッピングを示します。

- [.NET の使用](#)
- [WebSphere MQ .NET プログラムの作成およびデプロイ](#)
- [WebSphere MQ .NET のクラスとインターフェース](#)
- [Microsoft Windows Communication Foundation \(WCF\) 用の IBM WebSphere MQ カスタム・チャンネル](#)

C++ の使用

このセクションでは、古い「C++ の使用」ブックから新しい製品資料の構造へのマッピングを示します。

- [C++ の使用](#)
- [WebSphere MQ C++ クラス](#)

Java の使用

このセクションでは、古い「Java の使用」ブックから新しい製品資料の構造へのマッピングを示します。

- [WebSphere MQ classes for Java または WebSphere MQ classes for JMS を使用する必要がありますか?](#)
- [WebSphere MQ classes for Java の使用](#)
- [WebSphere MQ classes for JMS](#)
- [WebSphere MQ classes for JMS の使用](#)
- [WebSphere MQ classes for Java](#)

Web サービス

このセクションでは、以前の Web サービスと新しい製品資料の構造の対応を示します。

- [WebSphere MQ transport for SOAP](#)
- [WebSphere MQ Bridge for HTTP](#)

コンポーネント・オブジェクト・モデル・インターフェースの使用

このセクションでは、古い「コンポーネント・オブジェクト・モデル・インターフェースの使用」ブックから新しい製品資料の構造へのマッピングを示します。

- [コンポーネント・オブジェクト・モデル・インターフェース \(WebSphere MQ Automation Classes for ActiveX\) の使用](#)

IBM WebSphere MQ Version 7.5、IBM i、および z/OS

IBM i および z/OS では、IBM WebSphere MQ Version 7.5 を使用できません。

これらのプラットフォームは、新しいバージョンの製品で使用できます。

IBM MQ for IBM i および z/OS の最新バージョンについては、[IBM MQ Web サイト](#)を参照してください。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

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

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

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

日本アイ・ビー・エム株式会社

法務・知的財産

U.S.A.

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

Intellectual Property Licensing

Legal and Intellectual Property Law

日本 IBM

:NONE.

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

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

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

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

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

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

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

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

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

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

IBM の将来の方向性および指針に関するすべての記述は、予告なく変更または撤回される場合があります。これらは目標および目的を提示するものにすぎません。

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

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物にも、次のように、著作権表示を入れていただく必要があります。© (お客様の会社名) (西暦年)。このコードの一部は、IBM Corp. のサンプル・プログラム。© Copyright IBM Corp. 1993, 2024. All rights reserved.

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

商標

IBM、IBM ロゴ、ibm.com[®]、Passport Advantage[®]、WebSphere、MQSeries[®]、および z/OS は、世界の多くの国で登録された International Business Machines Corporation の商標です。世界中の多くの国で登録されています。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<https://www.ibm.com/legal/copytrade.shtml> の「Copyright and trademark information」をご覧ください。

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

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

Adobe、Adobe ロゴ、PostScript、および PostScript ロゴは、Adobe System sIncorporated の米国およびその他の国における登録商標または商標です。

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

ここに含まれる Oracle Outside In Technology は、制限付きご使用条件の適用を受け、このアプリケーションとともに使用する場合のみ使用できます。

Intel は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

登録商標 Linux は、世界中の商標の所有者である Linus Torvalds の排他的ライセンシーである Linux Foundation からのサブライセンスに従って使用されます。

IBM WebSphere MQ 技術概説

IBM WebSphere MQ を使用して、アプリケーションに接続し、組織全体での情報の分散を管理します。

IBM WebSphere MQ により、複数のプログラムは、一貫性のあるアプリケーション・プログラミング・インターフェースを使用して、異なるコンポーネント (プロセッサ、オペレーティング・システム、サブシステム、および通信プロトコル) のネットワーク間で相互に通信することができます。このインターフェースを使用して設計および作成されるアプリケーションは、メッセージ・キューイング・アプリケーションとして知られています。

以下のサブトピックを使用して、IBM WebSphere MQ で提供されるメッセージ・キューイングおよび他の機能についての情報を得ることができます。

関連概念

[6 ページの『IBM WebSphere MQ の概要』](#)

IBM WebSphere MQ を利用すると、複数のアプリケーションが、さまざまな時点で、多種多様なコンピューティング環境で通信できるようになります。

関連タスク

[WebSphere MQ アーキテクチャーの設計](#)

[WebSphere MQ Managed File Transfer](#)

関連資料

[44 ページの『メッセージ・キューイングの主な機能とメリット』](#)

ここでは、メッセージ・キューイングの主な機能とメリットを中心に説明します。メッセージ・キューイングのセキュリティやデータ保全性などについて説明します。

メッセージ・キューイングの概要

WebSphere MQ 製品では、整合性のあるアプリケーション・プログラミング・インターフェースを使用して、性質の異なるコンポーネント (プロセッサ、オペレーティング・システム、サブシステム、および通信プロトコル) のネットワーク内でプログラムが相互に通信できるようにしています。

メッセージングとキューイングのスタイルをとるため、このインターフェースを使用して設計、作成されるアプリケーションをメッセージ・キューイング・アプリケーションといいます。

メッセージング	プログラムは、相互に直接呼び出すのではなく、メッセージ形式で相互にデータをやりとりすることによって通信を行います。
キューイング	プログラムが異なるスピードおよび時間、異なる場所で、相互に論理的な結合をせずに独立した形で実行できるように、メッセージがストレージ内のキューに入れられます。

メッセージ・キューイングはデータ処理の分野で長年用いられてきました。現在では、電子メールの分野で広く使われています。キューイングがない場合、電子メッセージをリモート側で送信するには、経路上のすべてのノードがメッセージを転送でき、受信局をログオンさせて、メッセージを送信しようとしていることを常に認識させることが必要です。キューイング・システムでは、メッセージはシステムがそれを転送できるようになるまで中間ノードに格納されます。最終の宛先では、受信局が読み取り可能になるまで、メッセージは電子メールボックスに格納されます。

しかし、今日では多くの複雑な業務トランザクションがキューイングを使用しないで処理されています。大規模ネットワークでは、システムが膨大な数の接続を使用可能な状態にしていなければなりません。システムの一部で問題が発生すると、システムの多くの部分が使用できなくなります。

メッセージ・キューイングは、プログラム用の電子メールと考えることができます。メッセージ・キューイング環境では、アプリケーション・スイートの各部分を構成するそれぞれのプログラムが、特定の要求に応答する形で、明確に定義された自己完結型の機能を実行します。他のプログラムと通信するには、プログラムは事前定義キューにメッセージを書き込む必要があります。他のプログラムは、そのメッセージをキューから取り出し、そこに入っている要求や情報を処理します。このため、メッセージ・キューイングはプログラム間通信の 1 つのスタイルと言えます。

キューイングとは、アプリケーションがメッセージの処理が行えるようになるまでメッセージを保留する機能のことです。キューイングによって次のことが可能になります。

- 通信コードを記述することなく、プログラム間で通信を行うことができる (これらのプログラムはそれぞれ異なる環境で実行中でも構わない)。
- プログラムがメッセージを処理する順序を選択することができる。
- メッセージの数がしきい値を超えると、複数のプログラムが1つのキューをサービスするようにして、システム上の負荷のバランスをとることができる。
- 基本システムが使用不可のときは、代替システムがキューをサービスするようにして、アプリケーションの使用可能度を向上させることができる。

メッセージ・キューとは

メッセージ・キューは、単にキューと呼ばれており、メッセージを送信することができる名前付きの宛先です。メッセージは、それらのキューを取り扱うプログラムによって取り出されるまで、キューに蓄積されます。

キューは、キュー・マネージャー内にあり、キュー・マネージャーによって管理されます ([46 ページの『メッセージ・キューイングの用語』](#)を参照)。キューの物理的な特性は、キュー・マネージャーが実行されているオペレーティング・システムに依存します。キューはコンピューターのストレージ内の揮発性バッファ領域にあっても、あるいはディスクなどの永続記憶装置上のデータ・セットにあっても構いません。キューの物理的な管理は、キュー・マネージャーの役目であり、関連するアプリケーション・プログラムには明らかにされません。

プログラムは、キュー・マネージャーの外部サービスを通じてのみキューにアクセスします。外部サービスは、キューのオープン、キューへのメッセージの書き込み、キューからのメッセージの読み取り、およびキューのクローズを行うことができます。また、キューの属性を設定したり、照会したりすることもできます。

さまざまなスタイルのメッセージ・キューイング

Point-to-Point

1つのメッセージがキューに入れられ、1つのアプリケーションがそのメッセージを受け取ります。

Point-to-Point メッセージングでは、送信側アプリケーションが受信側アプリケーションにメッセージを送信する前に、送信側アプリケーションが受信側アプリケーションについての情報を認識している必要があります。例えば、送信側アプリケーションは、情報を送信するキューの名前を認識していて、キュー・マネージャー名を指定しなければならない場合があります。

パブリッシュ/サブスクライブ

パブリッシング・アプリケーションによってパブリッシュされた各メッセージのコピーは、各インタレスト・アプリケーションに送信されます。インタレスト・アプリケーションは多数ある場合も、1つある場合も、まったくない場合もあります。パブリッシュ/サブスクライブでは、インタレスト・アプリケーションはサブスクライバーと呼ばれ、メッセージはサブスクリプションによって識別されるキューに入れられます。

パブリッシュ/サブスクライブ・メッセージングを使用すると、情報の提供者とその情報の利用者を分離することができます。送信側アプリケーションと受信側のアプリケーションは、情報を送受信するために、お互いについて何も知る必要がありません。パブリッシュ/サブスクライブ・メッセージングについて詳しくは、[WebSphere MQ パブリッシュ/サブスクライブ・メッセージングの概要](#)を参照してください。

アプリケーションの設計担当者および開発者にとってのメッセージ・キューイングの利点

WebSphere MQ により、アプリケーション・プログラムは、メッセージ・キューイングを使用してメッセージ・ドリブン処理に加わることができます。適切なメッセージ・キューイング・ソフトウェア製品を使用すれば、プラットフォームが異なっても、アプリケーション・プログラム相互間で通信することができます。例えば、HP-UX アプリケーションと z/OS アプリケーションは、WebSphere MQ for HP-UX お

よび WebSphere MQ for z/OS を介して通信することができます。アプリケーションは、基礎となる通信機構からは保護されています。その他に、メッセージ・キューイングには次のような利点があります。

- 多くのアプリケーション間で共用できる小規模のプログラム群を使用して、アプリケーションを設計できる。
- これらの構築ブロックを再使用することによって、新しいアプリケーションを速やかに構築できる。
- メッセージ・キューイング技法を用いるため作成されたアプリケーションは、キュー・マネージャーの作業の仕方が変更されても影響を受けない。
- 通信プロトコルを使用する必要がない。キュー・マネージャーが通信のすべての局面を処理します。
- メッセージを受信するプログラムは、メッセージが送信されてくる時に実行中である必要はない。これらのメッセージはキューに保存されます。

設計担当者はアプリケーションのコストを下げるすることができます。なぜなら、開発の速度が速くなり、開発担当者の必要人数が少なくてすみ、しかもメッセージ・キューイングを使わないアプリケーションの場合よりもプログラミング・スキルの要求レベルが低いからです。

WebSphere MQ は、アプリケーションの実行場所にかかわらず、メッセージ・キュー・インターフェース (または MQI) と呼ばれる共通アプリケーション・プログラミング・インターフェースを実装します。このため、アプリケーション・プログラムを、あるプラットフォームから別のプラットフォームに移植するのが容易になります。

MQI の詳細については、[Message Queue Interface の概要](#)を参照してください。

メッセージ・キューイングの主な機能とメリット

ここでは、メッセージ・キューイングの主な機能とメリットを中心に説明します。メッセージ・キューイングのセキュリティーやデータ保全性などについて説明します。

メッセージ・キューイング方法を用いるアプリケーションの主な機能は、次のとおりです。

- プログラム間に直接接続がない。
- プログラム間の通信は時間に依存しないようにできる。
- 作業は小規模な、自己完結型のプログラムによって行える。
- 通信をイベント主導にすることができる。
- アプリケーションがメッセージに優先順位を割り当てることができる。
- セキュリティー。
- データ保全性。
- リカバリー・サポート。

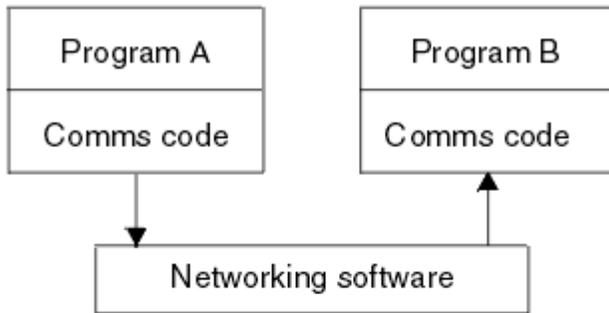
プログラム間に直接の接続がない

メッセージ・キューイングは、間接的なプログラム間通信の技法です。これは、プログラムが相互に通信し合うアプリケーション内で用いることができます。1つのプログラムがメッセージを (キュー・マネージャーが所有する) キューに書き込んで、次に別のプログラムがそのメッセージをキューから読み取ることによって通信が行われます。

プログラムは、他のプログラムによってキューに書き込まれたメッセージを読み取ることができます。他のプログラムは、受信プログラムとして同じキュー・マネージャーに接続することも、または別のキュー・マネージャーに接続することもできます。この別のキュー・マネージャーは別のシステム、異なったコンピューター・システム、またはたとえ異なった企業にあっても構いません。

メッセージ・キューを用いて通信するプログラム間には、物理的な接続はありません。一方のプログラムが、キュー・マネージャーが所有するキューにメッセージを送信し、もう一方のプログラムがキューからメッセージを取り出します ([45 ページの図 4](#) を参照)。

Traditional communication between programs



Communication by message queuing

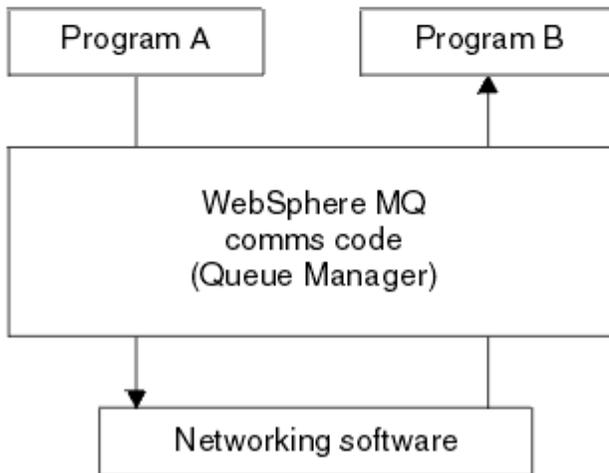


図 4. 従来の通信方式と比較したメッセージ・キューイング

電子メールの場合と同様に、トランザクションの一部である個々のメッセージは、蓄積交換方式に基づいてネットワーク内を移動します。基盤。ノード間のリンクに障害が起こった場合は、そのリンクが復旧するか、オペレーターまたはプログラムがそのメッセージの宛先を変更するまで、メッセージは保留されます。

メッセージがキューからキューへ移動する機構はプログラムには表れません。そのため、プログラムはより単純化されます。

時間に依存しない通信

他のプログラムに作業を要求したプログラムは、要求に対する応答を待つ必要はありません。これらのプログラムは他の作業を行うことができ、応答の処理をそれが到着したとき、または後で行うことができます。メッセージング・アプリケーションを作成するときは、プログラムがいつメッセージを送信するか、または相手がいつメッセージを受信できるかを知る (あるいは関与する) 必要はありません。メッセージが失われることはありません。メッセージは、相手がそれを処理できるようになるまで、キュー・マネージャーによって保存されます。メッセージは、プログラムが削除するまでキューにとどまっています。これは、送信と受信のアプリケーション・プログラムは分離されていることを意味します。送信側は、メッセージの受信に関する受信側の応答を待たずに処理を継続できます。ターゲット・アプリケーションが、メッセージの送信時に、実行中である必要もありません。ターゲット・アプリケーションは、始動後にメッセージを取り出すことができます。

小規模のプログラム

メッセージ・キューイングには、小規模の自己完結型のプログラムを使用できるという利点があります。単一の大規模なプログラムに作業のすべての部分を順次に行わせる代わりに、その作業をいくつかの小規模で独立したプログラムに分散させることができます。要求側プログラムは、個別の各プログラムにメッセージを送信して、それぞれの機能を実行するよう要求します。各プログラムは、機能を完了すると、1つまたは複数のメッセージとしてその結果を戻します。

メッセージ・ドリブン型の処理

トリガー操作と呼ばれるメカニズムを使用して、メッセージがキューに到着したときに自動的にアプリケーションを開始できます。必要に応じて、メッセージの処理が完了したときに、アプリケーションを停止させることもできます。

イベント主導型の処理

プログラムを、キューの状態に応じて制御できます。例えば、メッセージがキューに到着した直後にプログラムを開始するようしたり、またはある優先順位より上あるいはあらゆる優先順位のメッセージが、例えば 10 個キューに入るまではプログラムを開始しないようにすることもできます。

メッセージ優先順位

プログラムは、メッセージをキューに書き込むときに優先順位を割り当てることができます。これにより、新しいメッセージが追加されるキュー内の位置が決まります。

プログラムがキューからメッセージを取得する方法は、キューに入っているメッセージの順番に取得するか、特定のメッセージを取得するかのいずれかです。(プログラムが以前に送信した要求に対する応答を探している場合は、特定のメッセージを取得する必要があることがあります。)

セキュリティ

許可検査は各リソースごとに実行され、WebSphere MQ 管理担当者によって設定され、保守されているテーブルが使用されます。

- WebSphere MQ for z/OS 上の Security Server (旧称 RACF®) またはその他の外部セキュリティ・マネージャーを使用します。
- WebSphere MQ (UNIX システム、Linux システム、Windows システム、および IBM i) では、オブジェクト権限マネージャー (OAM) と呼ばれるセキュリティ・マネージャーがインストール可能サービスとして提供されます。デフォルトでは、OAM は活動状態です。

データ保全性

データ保全性は、作業単位によって提供されます。各 MQGET や MQPUT では、作業単位の開始と終了の同期は、オプションとして完全にサポートされており、作業単位の結果をコミットしたり、ロールバックしたりすることができます。同期点のサポートは、そのアプリケーションに選択された同期点調整の形式によって、WebSphere MQ の外側または内側のどちらかで動作します。

リカバリー・サポート

リカバリーを可能にするため、WebSphere MQ の持続的な更新はすべてログに記録されます。リカバリーが必要な場合、すべてのパーシスタント・メッセージがリストアされ、すべての未完了トランザクションがロールバックされ、すべての同期点コミットおよびバックアウトは制御中の同期点マネージャーが通常ハンドルの方法で処理されます。持続メッセージについては、[メッセージ持続性](#)を参照してください。

注: WebSphere MQ クライアントおよびサーバーを考慮する際、新しいプラットフォームで追加の WebSphere MQ MQI クライアントをサポートするためにサーバー・アプリケーションを変更する必要はありません。同様に、WebSphere MQ MQI クライアントは、変更しなくても、追加タイプのサーバーで機能することができます。

メッセージ・キューイングの用語

ここでは、メッセージ・キューイングで使われている一部の用語について説明します。

次の方法があります。

- [メッセージ](#)
- [メッセージ記述子](#)
- [キュー](#)
- [キュー・マネージャー](#)
- [チャネル](#)
- [メッセージ・チャネル・エージェント](#)
- [クラスター](#)
- [WebSphere MQ MQI クライアント](#)

- [Point-to-Point](#)
- [パブリッシュ/サブスクライブ](#)
- [トピック](#)
- [サブスクリプション](#)

メッセージ

メッセージ・キューイングでのメッセージとは、1つのプログラムから別のプログラムに向けて送られるデータの集合です。[IBM WebSphere MQ メッセージ](#)を参照してください。メッセージ・タイプの詳細については、[メッセージのタイプ](#)を参照してください。

メッセージ記述子

IBM WebSphere MQ メッセージは制御情報とアプリケーション・データから構成されています。制御情報は、メッセージ記述子構造体 (MQMD) で定義され、次のような項目で構成されています。

- メッセージのタイプ
- メッセージの ID
- メッセージ送達の優先順位

アプリケーション・データの構造と内容は、IBM WebSphere MQ ではなく、関連するプログラムによって決定されます。

キュー

メッセージの送信先になる指定された宛先です。メッセージは、それらのキューを取り扱うプログラムによって取り出されるまで、キューに蓄積されます。

キュー・マネージャー

キュー・マネージャーは、アプリケーションに対してキューイング・サービスを提供するシステム・プログラムです。

これは、アプリケーション・プログラミング・インターフェースを提供し、それによってプログラムはメッセージのキューへの書き込みおよびキューからのメッセージの読み取りができます。キュー・マネージャーは管理者が新しいキューを生成したり、既存のキューの属性を変更したり、キュー・マネージャーの操作を制御したりできる付加的な機能も提供します。

IBM WebSphere MQ メッセージ・キューイング・サービスをシステムで利用する場合、キュー・マネージャーを実行する必要があります。1つのシステムに複数のキュー・マネージャーを稼働させることができます(例えば、テスト・システムと実働システムを区別するため)。各キュー・マネージャーは、アプリケーションに対して、接続ハンドル (Hconn) で識別される。

種々のアプリケーションがキュー・マネージャーのサービスを同時に使用でき、しかもこれらのアプリケーションは互いにまったく無関係でも構いません。プログラムがキュー・マネージャーのサービスを利用するためには、そのキュー・マネージャーとの接続を確立する必要があります。

アプリケーションが他のキュー・マネージャーに接続しているアプリケーションにメッセージを送信するためには、そのキュー・マネージャーはこれらのキュー・マネージャー間で通信できなければなりません。IBM WebSphere MQ は、蓄積交換プロトコルを実装することにより、このようなアプリケーションの間で安全にメッセージが送達されるようにしています。

チャンネル

チャンネルとは、あるキュー・マネージャーと別のキュー・マネージャーを結ぶ通信パスを提供するオブジェクトのことです。あるキュー・マネージャーから別のキュー・マネージャーにメッセージを移すために、分散キューイングでチャンネルが使用され、それによってアプリケーションは基礎をなす通信プロトコルから遮蔽されます。キュー・マネージャーは、同一のプラットフォーム上に存在する場合もあれば、異なるプラットフォーム上に存在する場合もあります。

MSG チャンネル・エージェント

メッセージ・チャンネル・エージェントによって、あるキュー・マネージャーから他のキュー・マネージャーへメッセージが移動します。

参照は、レポート・メッセージを処理するときに作成されます。アプリケーションを設計する際に、これらの参照を考慮する必要があります。詳細については、独自のメッセージ・チャンネル・エージェントの作成を参照してください。

クラスター

クラスターとは、何らかの方法で論理的に関連付けられているキュー・マネージャーのネットワークです。クラスター化は、IBM WebSphere MQ Version 7.0 以降のバージョンのキュー・マネージャーで実行可能です。

クラスター化せずに分散キューイングを使用する IBM WebSphere MQ のネットワークでは、キュー・マネージャーはすべて独立しています。あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信する必要がある場合、送信元のキュー・マネージャーには、伝送キューとリモート・キュー・マネージャーへのチャンネルが定義されていなければなりません。

クラスターの使用には、システム管理の単純化、および可用性とワークロードの平衡化の向上という 2 つの異なる目的があります。

設定したクラスターが最小のものであっても、システム管理の単純化は実現されます。1 つのクラスターを構成しているキュー・マネージャーは多くの定義を必要としないので、間違った定義を作成する状況は少なくなります。

クラスターリングの詳細については、64 ページの『クラスターが機能する仕組み』を参照してください。

IBM WebSphere MQ MQI client

IBM WebSphere MQ MQI クライアントは、独立してインストール可能な IBM WebSphere MQ のコンポーネントです。MQI クライアントを使用することにより、通信プロトコルを使って IBM WebSphere MQ アプリケーションを実行し、他のプラットフォーム上の 1 つ以上の Message Queue Interface (MQI) サーバーと対話して、それらのキュー・マネージャーに接続することができます。

IBM WebSphere MQ MQI client コンポーネントをインストールおよび使用方法の詳細については、IBM WebSphere MQ MQI client のインストールおよびサーバーとクライアント間の接続の構成を参照してください。

Point-to-Point メッセージング

Point-to-Point メッセージングでは、各メッセージは、これを生成する 1 つのアプリケーションから、これを消費する 1 つのアプリケーションへ移動します。メッセージは、生成側のアプリケーションがメッセージをキューに入れることで転送され、消費側のアプリケーションはそのキューからメッセージを取得します。

パブリッシュ/サブスクライブ・メッセージング

パブリッシュ/サブスクライブ・メッセージングでは、パブリッシング・アプリケーションによってパブリッシュされた各メッセージのコピーは、各インタレスト・アプリケーションに送信されます。インタレスト・アプリケーションは多数ある場合も、1 つある場合も、まったくない場合もあります。パブリッシュ/サブスクライブでは、インタレスト・アプリケーションはサブスクライバーと呼ばれ、メッセージはサブスクリプションによって識別されるキューに入れられます。パブリッシュ/サブスクライブについて詳しくは、IBM WebSphere MQ パブリッシュ/サブスクライブ・メッセージングの概要を参照してください。

トピック

トピックは、パブリッシュ/サブスクライブ・メッセージでパブリッシュされる情報の主題を示す文字列です。

トピックは、パブリッシュ/サブスクライブ・システムでメッセージを正常に送達するうえで、重要な役割を果たします。パブリッシャーは、各メッセージに特定の宛先アドレスを含める代わりに、各メッセージにトピックを割り当てます。キュー・マネージャーは、トピックと、そのトピックをサブスクライブしているサブスクライバーのリストとを突き合わせ、それらのサブスクライバーの各々にメッセージを送達します。

サブスクリプション

パブリッシュ/サブスクライブ・アプリケーションは、特定のトピックに関して、インタレストをメッセージに登録することができます。これを行ったアプリケーションはサブスクライバーと呼ばれ、マッチング・メッセージが処理のためにキューに入れられる方法を、用語サブスクリプションが定義します。

サブスクリプションには、サブスクライバーの ID、パブリケーションを配置する宛先キューの ID についての情報が含まれます。また、パブリケーションを宛先キューに配置する方法についての情報も含まれます。

メッセージとキュー

メッセージとキューは、メッセージ・キューイング・システムの基本構成要素です。

メッセージとは

メッセージを使用するアプリケーションにとって、メッセージは意味のあるバイト・ストリングです。メッセージは、ある 1 つのアプリケーションから別のアプリケーションに (または、同じアプリケーションの異なる部分の間で) 情報を転送するために使用されます。伝達に関するアプリケーションは、同じプラットフォーム上で実行されていても、別のプラットフォーム上で実行されていてもかまいません。

IBM WebSphere MQ メッセージには、次の 2 つのパーツがあります。

- アプリケーション・データ。アプリケーション・データの内容と構造は、そのデータを使用するアプリケーション・プログラムによって定義されます。
- メッセージ記述子。メッセージ・ディスクリプターは、メッセージを識別し、メッセージのタイプおよび送信側アプリケーションによってメッセージに割り当てられた優先順位などの追加の制御情報を含んでいます。

メッセージ記述子の形式は、IBM WebSphere MQ によって定義されます。メッセージ記述子の詳細な説明については、[MQMD - メッセージ記述子](#)を参照してください。

メッセージ長

デフォルトの最大メッセージ長は 4 MB ですが、この最大長を 100 MB まで増やすことができます (1 MB は 1 048 576 バイトです)。ただし、実際には、次のものによってメッセージ長は制限されます。

- 受信側のキュー用に定義した最大メッセージ長
- キュー・マネージャー用に定義した最大メッセージ長
- キューによって定義された最大メッセージ長
- 送信側のアプリケーションまたは受信側のアプリケーションで定義した最大メッセージ長
- メッセージ用として使用可能なストレージの量

1 つのアプリケーションが必要とするすべての情報を送るには、複数のメッセージが必要な場合があります。

アプリケーションによるキューの送信/受信方法

アプリケーションでは、**MQI 呼び出し**を使用してメッセージを送受信します。

例えば、キューにメッセージを書き込むために、アプリケーションでは以下の処理が行われます。

1. MQOPEN 呼び出しを発行して必要なキューをオープンする。
2. MQPUT 呼び出しを発行してキューにメッセージを書き込む。

別のアプリケーションがそのキューからメッセージを取り出すときには、MQI MQGET 呼び出しを発行します。

MQI 呼び出しの詳細については、[MQI 呼び出し](#)を参照してください。

キューとは

キューとは、メッセージを保管するためのデータ構造体です。

キューはすべて、キュー・マネージャーに属しています。キュー・マネージャーは、所有するキューを管理し、受信したすべてのメッセージを適切なキューに格納します。メッセージは、アプリケーション・プログラムまたはキュー・マネージャーによって、通常の操作の一部としてキューに書き込まれます。

事前定義キューと動的キュー

キューは、その作成方法によって次のような特徴があります。

- **事前定義キュー**は、管理者が該当する MQSC または PCF コマンドを使用して作成します。事前定義キューは、永続キューであって、それらを使用するアプリケーションとは無関係に存在し、IBM WebSphere MQ が再始動しても存続します。
- **動的キュー**が作成されるのは、アプリケーションがモデル・キューの名前を指定して MQOPEN 要求を出した場合です。作成されたキューは、モデル・キューというテンプレート・キュー定義に基づきます。MQSC コマンド DEFINE QMODEL を使用してモデル・キューを作成することができます。モデル・キューの属性 (例えば、キューに保管できるメッセージの最大数) は、そのモデル・キューから作成される動的キューが継承します。

モデル・キューは、作成される動的キューが永続キューになるか一時キューになるかを指定する属性を持っています。永続キューは、アプリケーションやキュー・マネージャーが再始動しても存続しますが、一時キューは、再始動すると失われます。

キューからのメッセージの取り出し

MQSeries では、アプリケーションに適切な許可が与えられている場合、次の取り出しアルゴリズムに従ってキューからメッセージを取り出すことができます。

- 先入れ先出し法 (first-in-first-out (FIFO))。
- メッセージ記述子に定義されたメッセージ優先順位。同じ優先順位を持つメッセージは、FIFO 順に取り出されます。
- 特定のメッセージについてのプログラム要求。

アプリケーションからの MQGET 要求によって、使用される方式が決まります。

相互通信の概念

WebSphere MQ において、相互通信とは、あるキュー・マネージャーから別のキュー・マネージャーへメッセージが送信されることを意味します。受信キュー・マネージャーは、同じマシン上に置くことも別のマシン上に置くこともでき、近くにあっても地球の裏側にあってもメッセージを受信できます。このプログラムは、ローカル・キュー・マネージャーと同じプラットフォーム、または WebSphere MQ がサポートするいずれかのプラットフォームで実行することができます。これを分散環境と呼びます。WebSphere MQ は、このような分散環境での通信を分散キューイング管理 (DQM) によって処理します。

ローカル・キュー・マネージャーは、発信元キュー・マネージャーとも呼ばれ、リモート・キューはターゲット・キュー・マネージャーまたはパートナー・キュー・マネージャーとも呼ばれます。

分散キューイングの機能の仕方

分散キューイングは、あるキュー・マネージャーから別のキュー・マネージャーへのメッセージの送信を可能にします。受信キュー・マネージャーは、同じマシン上に置くこともリモート・マシン上に置くこともできます。キュー・マネージャー、キュー、チャンネルおよび関連する定義が、クラスタリング (論理的に関連付けられたキュー・マネージャーのネットワーク) とともに略述されています。

51 ページの図 5 は、分散キューイングのコンポーネントの概要を示しています。

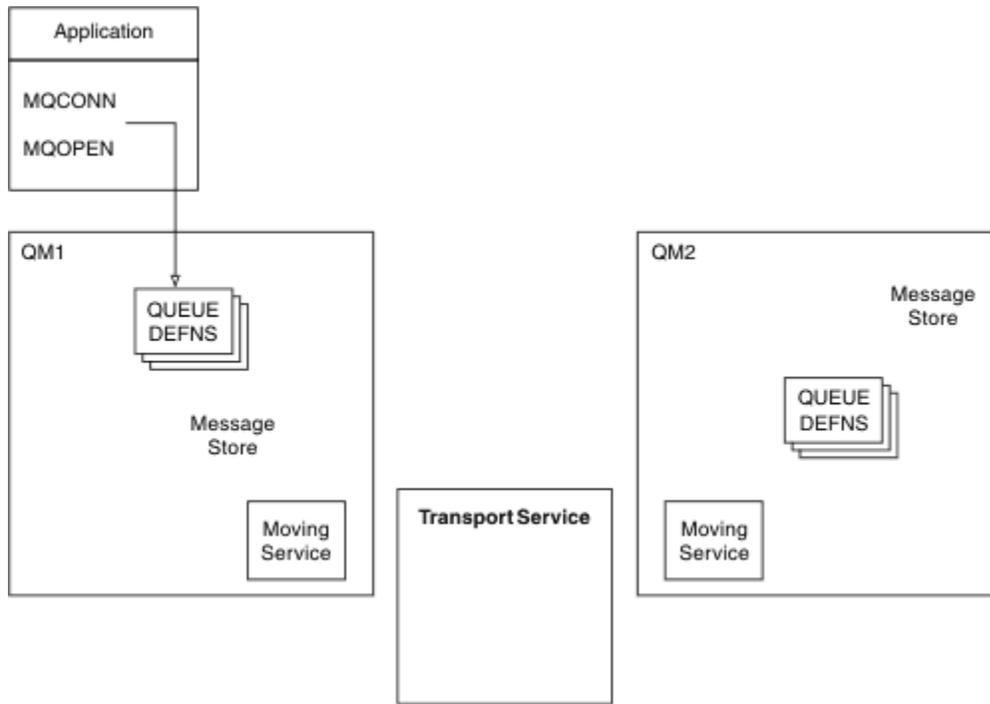


図 5. 分散キューイングのコンポーネントの概要

1. アプリケーションは、MQCONN 呼び出しを使用してキュー・マネージャーに接続します。
2. 次に MQOPEN 呼び出しを使用してキューをオープンします。これによって、アプリケーションはメッセージを書き込むことができます。
3. キュー・マネージャーには、キューに書き込めるメッセージの最大数などの情報を指定した定義が、キューごとに備えられています。また、リモート・キュー・マネージャーに配置されているキューの定義が入っている場合もあります。
4. リモート・システムでキューに対してメッセージが指定されると、それらをリモート・キュー・マネージャーに送る準備ができるまで、ローカル・キュー・マネージャーはメッセージ・ストアにそれらのメッセージを保持します。これは、アプリケーションには影響ありません。
5. キュー・マネージャーにはそれぞれ移動サービス・コンポーネントと呼ばれる通信ソフトウェアが備えられています。このソフトウェアを介してキュー・マネージャーは他のキュー・マネージャーと通信することができます。
6. トランスポート・サービスは、キュー・マネージャーが関与しないサービスであり、次のいずれかになります(プラットフォームにより異なります)。
 - SNA APPC
 - 伝送制御プロトコル/インターネット・プロトコル (TCP/IP) (Transmission Control Protocol/Internet Protocol (TCP/IP))
 - NetBIOS
 - SPX

分散キューイングのコンポーネントについて

WebSphere MQ アプリケーションは、メッセージをローカル・キュー (アプリケーションの接続先のキュー・マネージャーのキュー) に書き込むことができます。

キュー・マネージャーには各キューの定義が入っています。また、他のキュー・マネージャーのキューについての定義が入っている場合もあります。これらは、リモート・キュー定義と呼ばれます。WebSphere MQ アプリケーションは、このようなりモート・キュー宛でのメッセージを書き込むこともできます。

リモート・キュー・マネージャーに対してメッセージが指定されると、それらをリモート・キュー・マネージャーに送る準備ができるまで、ローカル・キュー・マネージャーは伝送キューにそれらのメッセージ

を保管します。伝送キューは特殊なタイプのローカル・キューであり、メッセージは、リモート・キュー・マネージャーに正常に伝送され保管されるまでここに保管されます。

メッセージの送受信を処理するソフトウェアは、メッセージ・チャンネル・エージェント (MCA) と呼ばれます。

メッセージは、チャンネルを通してキュー・マネージャー同士でやりとりされます。チャンネルは、2つのキュー・マネージャーをつなぐ一方向の通信リンクです。チャンネルを使用して、リモート・キュー・マネージャーの任意の数のキュー宛てのメッセージを送ることができます。

メッセージの送信に必要なコンポーネント

メッセージがリモート・キュー・マネージャーに送信される場合、ローカル・キュー・マネージャーには伝送キューとチャンネルに関する定義が必要です。

チャンネルの両側には、例えば送信側や受信側として定義される独立した定義が備えられています。簡単なチャンネルは、ローカル・キュー・マネージャーの送信側チャンネル定義とリモート・キュー・マネージャーの受信側チャンネル定義から構成されます。これらの2つの定義は、同じ名前をもち、共に1つのチャンネルを構成している必要があります。

また、チャンネルの両側に、メッセージ・チャンネル・エージェント (MCA) が1つずつあります。

それぞれのキュー・マネージャーには送達不能キュー (未配布メッセージ・キューとも呼びます) が入っていないければなりません。メッセージを宛先に送達できない場合、メッセージはこのキューに書き込まれます。

52 ページの図 6 は、キュー・マネージャー、伝送キュー、チャンネル、および MCA の関係を示しています。

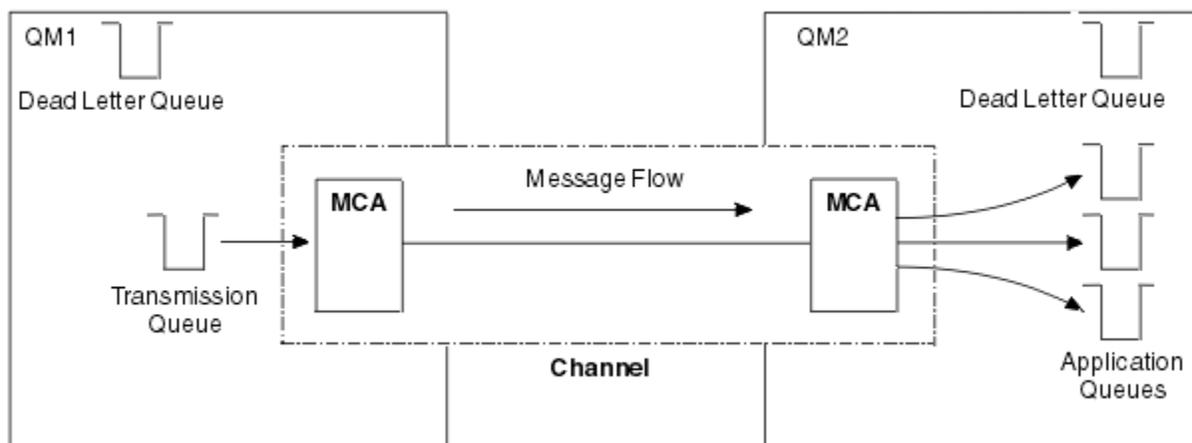


図 6. メッセージの送信

メッセージが戻るために必要なコンポーネント

リモート・キュー・マネージャーからアプリケーションにメッセージが戻される必要がある場合、別のチャンネルを定義して、53 ページの図 7 に示すようにキュー・マネージャー間で反対方向に実行する必要があります。

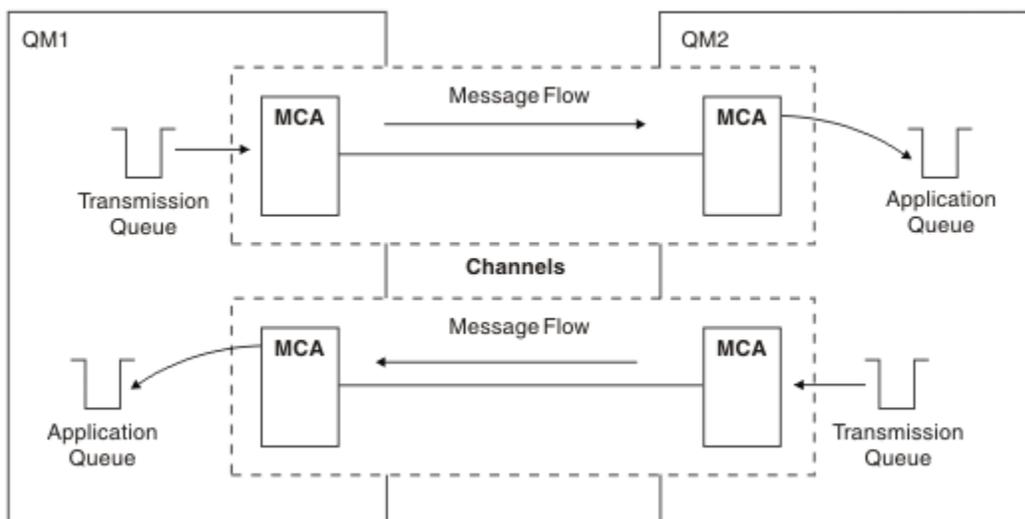


図 7. 両方向のメッセージ送信

分散キュー管理の詳細については、[分散キュー管理の概要](#)を参照してください。

クラスターのコンポーネント

手動で定義されているチャンネルを介して相互接続されている従来の WebSphere MQ ネットワークに代わるものとして、クラスターを使用できます。

クラスターとは、何らかの形態で論理的に関連付けられたキュー・マネージャーのネットワークです。複数のキュー・マネージャーを1つのクラスターにグループ化すると、各クラスターがホストになっているキューをクラスター内の別のキュー・マネージャーから使用可能にすることができます。必要なインフラストラクチャーが整っていれば、すべてのキュー・マネージャーは各宛先の明示的なチャンネル定義、リモート・キュー定義、または伝送キューがなくても同じクラスター内の別のキュー・マネージャーへメッセージを送信できます。クラスター内のすべてのキュー・マネージャーは単一の伝送キューを備えており、その伝送キューによってクラスター内の別のキュー・マネージャーへメッセージが伝送されます。それぞれのキュー・マネージャーは、ただ1つのクラスター受信側チャンネルと1つのクラスター送信側チャンネルを定義する必要があります。追加のチャンネルはクラスターによって自動的に管理されます。

54 ページの図 8 は、CLUSTER というクラスターのコンポーネントを示しています。

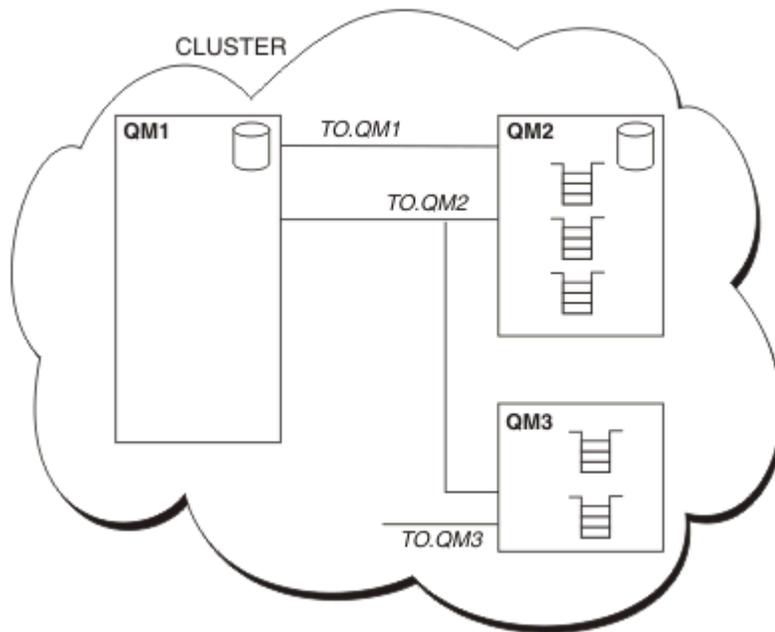


図 8. 複数のキュー・マネージャーで構成されるクラスター

- CLUSTER には、QM1、QM2、および QM3 という 3 つのキュー・マネージャーが備えられています。
- QM1 と QM2 は、クラスター内にあるキュー・マネージャーとキューに関する情報の全リポジトリのホストです。
- QM2 と QM3 は、いくつかのクラスター・キュー (つまり、このクラスター内にある別のキュー・マネージャーからアクセス可能なキュー) のホストです。
- それぞれのキュー・マネージャーは、メッセージを受信する TO.qmgr というクラスター受信側チャンネルを 1 つずつ備えています。
- また、それぞれのキュー・マネージャーは、リポジトリ・キュー・マネージャーの 1 つに情報を送信するクラスター送信側チャンネルも 1 つずつ備えています。
- QM1 と QM3 は、QM2 のリポジトリへ送信し、QM2 は QM1 のリポジトリへ送信します。

分散キューイングと同様に、いずれかのキュー・マネージャーのキューへメッセージを置くには、MQPUT 呼び出しを使用します。ローカル・キューからメッセージを取り出すには、MQGET 呼び出しを使用します。

クラスターについての詳細は、[38 ページの『キュー・マネージャー・クラスター』](#)を参照してください。

関連概念

[55 ページの『分散キューイング・コンポーネント』](#)
相互通信を使用可能にするのに必要なオブジェクトがあります。

[57 ページの『送達不能キュー』](#)
送達不能キュー (または未配布メッセージ・キュー) は、正しい宛先にメッセージを送信できない場合にそのメッセージが送信されるキューです。

[58 ページの『リモート・キュー定義』](#)
リモート・キュー定義とは、別のキュー・マネージャーに所有されるキューの定義です。

[58 ページの『リモート・キュー・マネージャーへのアクセス方法』](#)
それぞれの発信元キュー・マネージャーとターゲット・キュー・マネージャーをつなぐチャンネルは常に 1 つとは限りません。2 つをつなぐいくつかの別の方法として、マルチ・ホップ、チャンネルの共用、個別のチャンネルおよびクラスタリングの使用などがあります。

[60 ページの『アドレッシング情報』](#)

アプリケーションがリモート・キュー・マネージャーに定義されているメッセージを書き込むとき、それらを伝送キューに入れる前に、ローカル・キュー・マネージャーは伝送ヘッダーを追加します。このヘッダーには、宛先キューとキュー・マネージャーの名前、すなわちアドレッシング情報が収められています。

61 ページの『別名について』

別名を使用すれば、メッセージ・サービスの質が向上します。キュー・マネージャーの別名を使用すると、システム管理者はアプリケーションを変更しないでターゲット・キュー・マネージャーの名前を変更することができます。また、システム管理者は、ターゲット・キュー・マネージャーへの経路を変更したり、多数の他のキュー・マネージャーを介する(マルチ・ホップ)経路を設定したりできます。応答キューに別名が付けられるため、サービスが使いやすくなります。

61 ページの『キュー・マネージャー別名定義』

メッセージを書き込むためにキューをオープンするアプリケーションがキュー名、およびキュー・マネージャー名を指定するとき、キュー・マネージャー別名定義が適用されます。

63 ページの『応答先キュー別名の定義』

応答先キューの別名定義は、メッセージ記述子の応答情報の代替名を指定します。この利点は、アプリケーションを変更しないでキューまたはキュー・マネージャーの名前を変更できることです。

分散キューイング・コンポーネント

相互通信を使用可能にするのに必要なオブジェクトがあります。

分散キューイングのコンポーネントは次のとおりです。

- メッセージ・チャンネル
- メッセージ・チャンネル・エージェント
- 伝送キュー
- チャンネル・イニシエーターとチャンネル・リスナー
- チャンネル出口プログラム

メッセージ・チャンネルは、あるキュー・マネージャーから別のキュー・マネージャーへメッセージを送るチャンネルです。メッセージ・チャンネルと MQI チャンネルを混同しないようにしてください。MQI チャンネルには、サーバー接続 (SVRCONN) とクライアント接続 (CLNTCONN) の2つのチャンネルがあります。詳しくは、[MQI チャンネル](#)を参照してください。

メッセージ・チャンネルのそれぞれの側の定義は、次のタイプのいずれかです。

- 送信側 (SDR)
- 受信側 (RCVR)
- サーバー (SVR)
- 要求側 (RQSTR)
- クラスター送信側 (CLUSSDR)
- クラスター受信側 (CLUSRCVR)

メッセージ・チャンネルは、一方の側で定義されたタイプと、他方の側の互換性のあるタイプを使用して定義されます。可能な組み合わせは以下のとおりです。

- 送信側と受信側
- 要求側とサーバー
- 要求側と送信側(コールバック)
- サーバーと受信側
- クラスター送信側とクラスター受信側

送受信チャンネル作成の詳しい手順は、[チャンネルの定義](#)に記載されています (z/OS 以外)。送受信チャンネルのセットアップに必要なパラメーターの例は、ご使用のプラットフォーム用の[構成情報の例](#)を参照してください。チャンネルの定義(タイプを問わず)に必要なパラメーターについては、[DEFINE CHANNEL](#)を参照してください。

送信側 - 受信側チャンネル

あるシステムで送信側がチャンネルを開始すると、他のシステムにメッセージを送信できるようになります。送信側は、チャンネルのもう一方の側の受信側に開始を要求します。送信側は伝送キューから受信側にメッセージを送信します。受信側はメッセージを宛先キューに書き込みます。56 ページの図 9 これを示します。

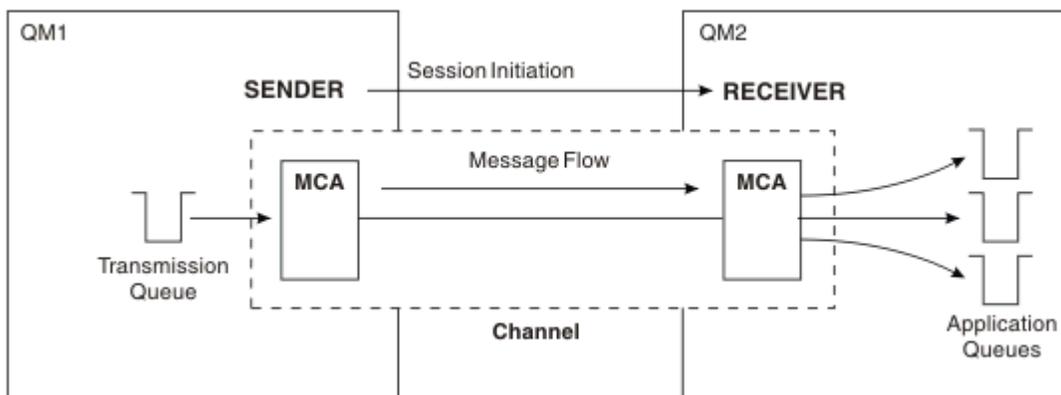


図 9. 送信側チャンネルと受信側チャンネル

要求側 - サーバー・チャンネル

あるシステムの要求側がチャンネルを開始すると別のシステムからのメッセージを受信できるようになります。要求側は、チャンネルのもう一方の側のサーバーに開始を要求します。サーバーは、そのチャンネル定義に定義された伝送キューから要求側にメッセージを送信します。

サーバー・チャンネルは、通信を開始して要求側にメッセージを送信することもできます。これは完全修飾サーバー、すなわちチャンネル定義で指定されたパートナーの接続名を持つサーバー・チャンネルにのみ適用されます。要求側が完全修飾サーバーを開始することができますが、完全修飾サーバーが要求側との通信を開始することもできます。

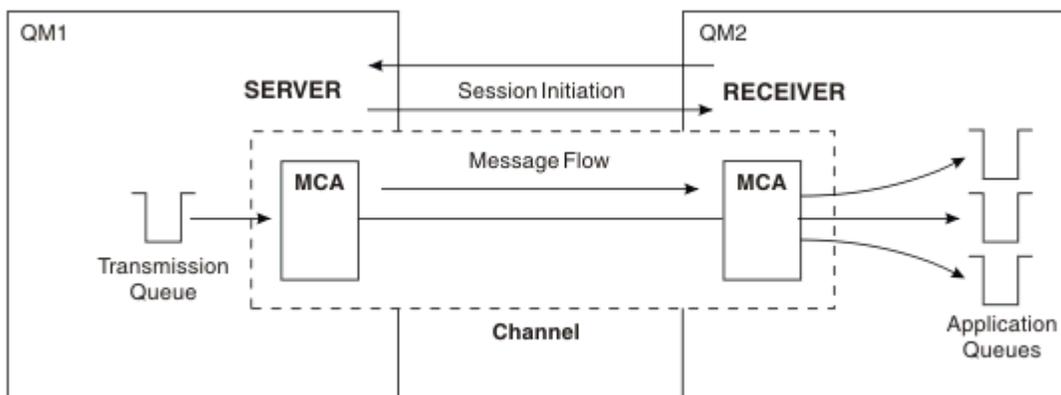


図 10. 要求側チャンネルとサーバー・チャンネル

要求側 - 送信側チャンネル

要求側はチャンネルを開始し、送信側は呼び出しを終了します。次に、送信側はチャンネル定義に入っている情報によって通信を再開します (コールバックと呼ばれます)。その後、伝送キューから要求側にメッセージを送ります。

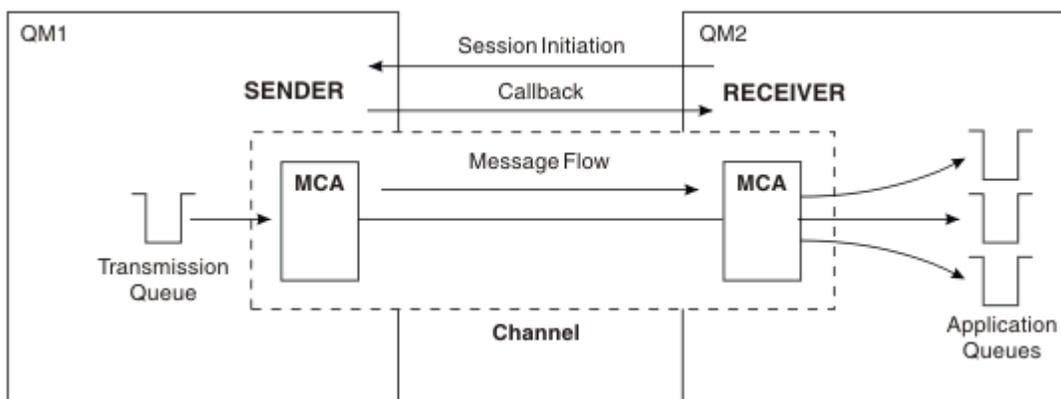


図 11. 要求側チャンネルと送信側チャンネル

サーバー・チャンネルと受信側チャンネル

これは、送信側チャンネルと受信側チャンネルに似ていますが、完全修飾サーバー、すなわちチャンネル定義で指定されたパートナーの接続名をもつサーバー・チャンネルにだけ適用されます。チャンネルは、リンクのサーバー側で開始する必要があります。次の図は、56 ページの図 9 に似ています。

クラスター送信側チャンネル

クラスター内では、それぞれのキュー・マネージャーは全リポジトリ・キュー・マネージャーの 1 つに情報を送信するクラスター送信側チャンネルを 1 つずつ備えています。キュー・マネージャーは、クラスター送信側チャンネル上の別のキュー・マネージャーへメッセージを送信することもできます。

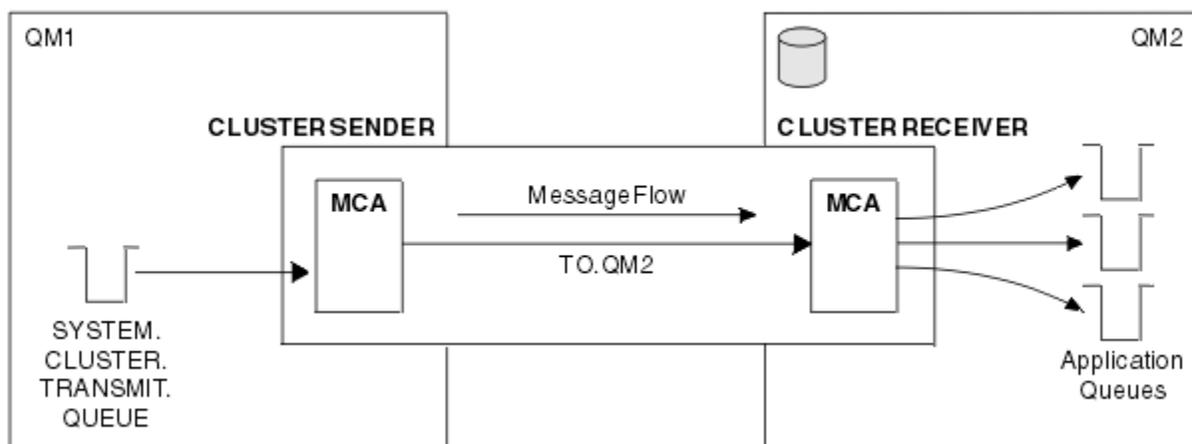


図 12. クラスター送信側チャンネル

クラスター受信側チャンネル

クラスター内の各キュー・マネージャーは、そのクラスターに関する情報とメッセージを受信するクラスター受信側チャンネルを 1 つずつ備えています。次の図は、57 ページの図 12 に似ています。

送達不能キュー

送達不能キュー (または未配布メッセージ・キュー) は、正しい宛先にメッセージを送信できない場合にそのメッセージが送信されるキューです。

メッセージを宛先キューに書き込みできないとき、メッセージはこのキューに書き込まれます。例えば、キューが存在しなかったり、満杯であるためです。送達不能キューは、チャンネルの送信側のデータ変換エラーの場合にも使用されます。

キュー・マネージャーごとに送達不能キューを定義することを考慮してください。定義しない場合、MCA はメッセージを書き込むことができず、伝送キューに残ったままとなり、チャンネルは停止してしまいます。

また、高速の非持続メッセージ ([Fast, nonpersistent messages](#) を参照) を送達できず、ターゲット・システム上に送達不能キューが存在しない場合、これらのメッセージは破棄されます。

ただし、送達不能キューを使用すると、メッセージが送達される順序に影響するので、これらを使用しなくてもかまいません。

USEDLQ チャンネル属性を使用して、メッセージが配信できない場合に送達不能キューを使用するかどうかを判別できます。キュー・マネージャーのいくつかの機能が送達不能キューを使用する一方で、他の機能がそれを使用しないように、この属性を構成できます。さまざまな MQSC コマンドでの USEDLQ チャンネル属性の使用については、[DEFINE CHANNEL](#)、[DISPLAY CHANNEL](#)、[ALTER CHANNEL](#)、および [DISPLAY CLUSQMGR](#) を参照してください。

リモート・キュー定義

リモート・キュー定義とは、別のキュー・マネージャーに所有されるキューの定義です。

アプリケーションはローカル・キューからしかメッセージを取り出せませんが、ローカル・キューまたはリモート・キューのどちらにもメッセージを書き込むことができます。したがって、キュー・マネージャーには、ローカル・キューごとの定義だけでなくリモート・キュー定義も備えていることがあります。リモート・キュー定義の利点は、リモート・キューやリモート・キュー・マネージャーの名前、または伝送キューの名前を指定しなくても、アプリケーションがリモート・キューにメッセージを書き込めることです。リモート・キュー定義を使用すると、ロケーションに依存しなくて済みます。

リモート・キュー定義は、他にも使用できますが、それについては後述します。

リモート・キュー・マネージャーへのアクセス方法

それぞれの発信元キュー・マネージャーとターゲット・キュー・マネージャーをつなぐチャンネルは常に 1 つとは限りません。2 つをつなぐいくつかの別の方法として、マルチ・ホップ、チャンネルの共用、個別のチャンネルおよびクラスターリングの使用などがあります。

マルチ・ホップ

ソース・キュー・マネージャーとターゲット・キュー・マネージャーをつなぐ直接の通信リンクがない場合、ターゲット・キュー・マネージャーへの経路の途中で 1 つまたは複数の中間キュー・マネージャーを介して伝送することができます。これは、マルチ・ホップと呼ばれます。

中間キュー・マネージャーでは、すべてのキュー・マネージャー同士、および伝送キュー同士をつなぐチャンネルを定義する必要があります。これについては、[59 ページの図 13](#) に示してあります。

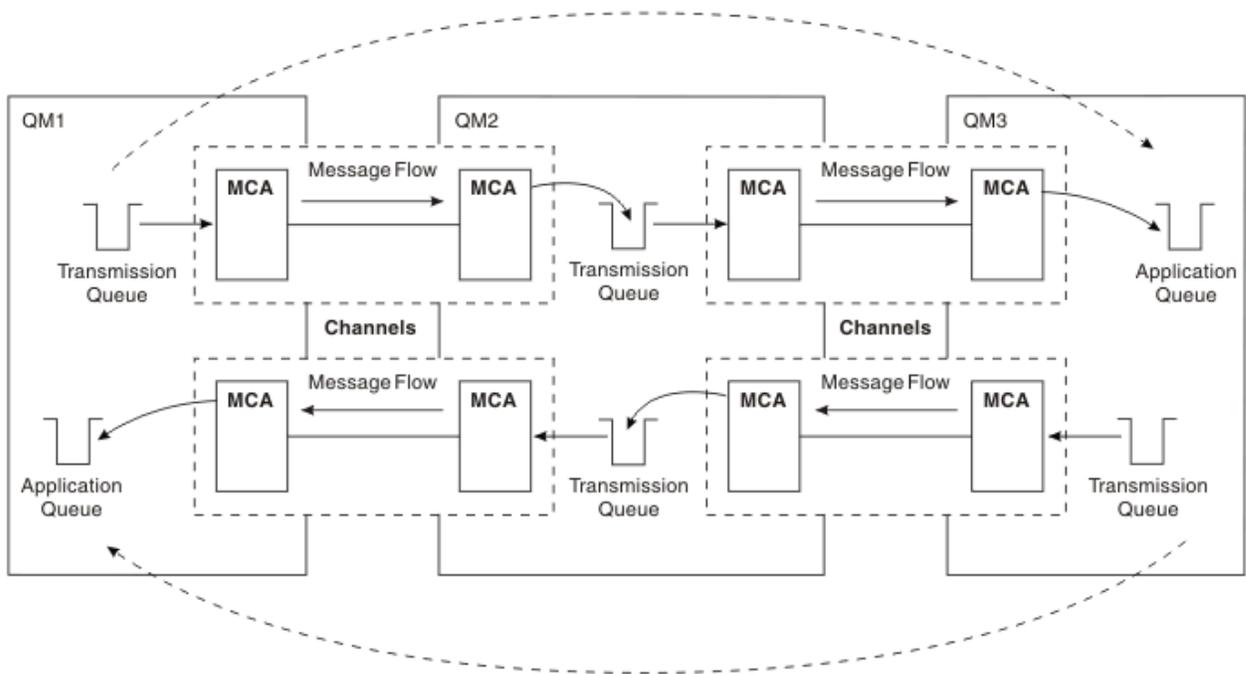


図 13. 中間キュー・マネージャーを介した伝送

チャンネルの共有

アプリケーションを設計する場合は、キュー名と共にリモート・キュー・マネージャー名をアプリケーションに指定させるか、各リモート・キューごとにリモート・キュー定義を作成するかを選択できます。この定義には、リモート・キュー・マネージャー名、キュー名、および伝送キューの名前が入っています。どちらの方法を使用しても、同じリモート・ローケーションのキューをアドレッシングするすべてのアプリケーションからのすべてのメッセージは、同じ伝送キューを介して送られます。これについては、[59 ページの図 14](#) に示してあります。

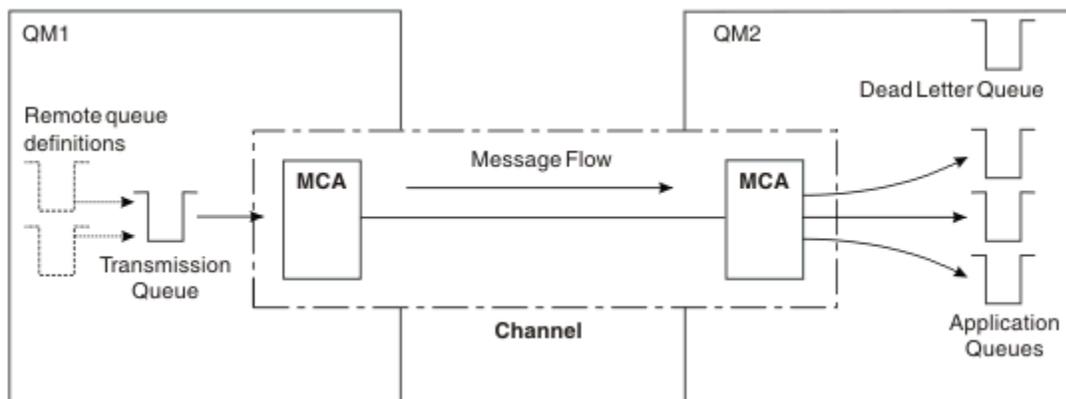


図 14. 伝送キューの共有

[59 ページの図 14](#) は、複数のアプリケーションから同じチャンネルを使用できる複数のリモート・キューへのメッセージの伝達を表しています。

個別のチャンネルの使用

2つのキュー・マネージャー間で異なるタイプのメッセージを送信する場合、その2つのキュー・マネージャー間に複数のチャンネルを定義することができます。セキュリティのため、あるいは配送速度を犠牲にしてもメッセージ・トラフィックを一括させるために、代替チャンネルが必要な場合があります。

2番目のチャンネルをセットアップするには、別のチャンネルと伝送キューを定義し、ロケーションと伝送キュー名を指定するリモート・キュー定義を作成する必要があります。これらを定義した後、アプリケーションではどちらのチャンネルも使用できますが、メッセージは依然として同じターゲット・キューに送達されます。これについては、60ページの図15に示してあります。

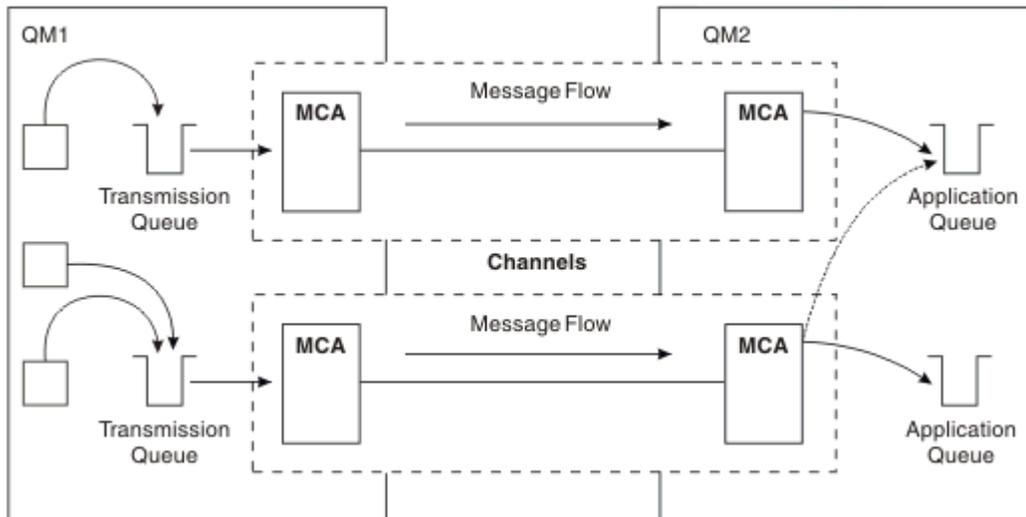


図 15. 複数チャンネルの使用方法

伝送キューを指定するのにリモート・キュー定義を使用する場合、アプリケーションではロケーション(つまり、ターゲット・キュー・マネージャー)を指定してはなりません。指定した場合、キュー・マネージャーはリモート・キュー定義を使用しません。リモート・キュー定義を使用すると、ロケーションに依存しなくて済みます。アプリケーションは、このキューのロケーションを知らなくても、メッセージを論理キューに書き込むことができます。したがって、アプリケーションを変更しなくても、物理キューを変更することができます。

クラスターの使用

クラスター内のすべてのキュー・マネージャーは、クラスター受信側チャンネルを定義します。別のキュー・マネージャーは、そのキュー・マネージャーへメッセージを送信するときに自動的にクラスター送信側チャンネルを定義します。例えば、あるキューのインスタンスがクラスター内に2つ以上存在する場合、そのキューのホストとなるどのキュー・マネージャーに対してもクラスター送信側チャンネルを定義できます。WebSphere MQでは、メッセージの経路を指定するのに使えるキュー・マネージャーを選択するため、ラウンドロビン・ルーチンを使用する作業負荷管理アルゴリズムが使用されます。詳細については、128ページの『クラスター』を参照してください。

アドレッシング情報

アプリケーションがリモート・キュー・マネージャーに定義されているメッセージを書き込むとき、それらを伝送キューに入れる前に、ローカル・キュー・マネージャーは伝送ヘッダーを追加します。このヘッダーには、宛先キューとキュー・マネージャーの名前、すなわちアドレッシング情報が収められています。

単一キュー・マネージャー環境では、宛先キューのアドレスは、メッセージを書き込むキューをアプリケーションがオープンしたときに設定されます。宛先キューは同じキュー・マネージャー上にあるため、アドレッシングのための情報は必要ありません。

分散環境では、キュー・マネージャーは、宛先キュー名だけでなくキューのロケーション(すなわち、キュー・マネージャー名)、およびその離れた場所への経路(すなわち、伝送キュー)を知る必要があります。このアドレッシング情報は、伝送ヘッダーに含まれています。受信チャンネルは、伝送ヘッダーを取り外して、その中の情報を使用して宛先キューの場所を見つけます。

リモート・キュー定義を使用すると、宛先キュー・マネージャーの名前をアプリケーションが指定する必要がなくなります。この定義は、リモート・キューの名前、メッセージが送られるリモート・キュー・マネージャーの名前、およびメッセージの移送に使用される伝送キューの名前を指定します。

別名について

別名を使用すれば、メッセージ・サービスの質が向上します。キュー・マネージャーの別名を使用すると、システム管理者はアプリケーションを変更しないでターゲット・キュー・マネージャーの名前を変更することができます。また、システム管理者は、ターゲット・キュー・マネージャーへの経路を変更したり、多数の他のキュー・マネージャーを介する(マルチ・ホップ)経路を設定したりできます。応答キューに別名が付けられるため、サービスが使いやすくなります。

キュー・マネージャーの別名および応答キューの別名は、ブランクの RNAME の入ったリモート・キュー定義を使用して作成されます。これらの定義は、実際のキューを定義するものではありません。これらの定義は、物理キュー名、キュー・マネージャー名、および伝送キューを解決するためにキュー・マネージャーが使用します。

別名定義はブランクの RNAME を備えているという特徴があります。

キュー名の解決

キュー名の解決は、キューがオープンされるごとにすべてのキュー・マネージャーにおいて生じます。その目的は、ターゲット・キュー、ターゲット・キュー・マネージャー(ローカルの場合もあります)、およびそのキュー・マネージャーへの経路(ヌルの場合もあります)の指定です。解決された名前は、キュー・マネージャー名、キュー名、および伝送キュー(キュー・マネージャーがリモートの場合)の3つの部分から構成されます。

リモート・キュー定義がある場合、別名定義は参照されません。アプリケーションから提供されたキュー名は、ターゲット・キューの名前、リモート・キュー・マネージャー、およびリモート・キュー定義に指定される伝送キューに解決されます。キュー名の解決についての詳細は、[キュー名の解決](#)を参照してください。

リモート・キュー定義がなく、キュー・マネージャーが指定されているか、または名前サービスによって解決されている場合、キュー・マネージャーは、システムに提供されたキュー・マネージャー名と一致するキュー・マネージャー別名定義があるかどうかを探します。その定義がある場合は、その中の情報を使用して、ターゲット・キュー・マネージャーの名前へのキュー・マネージャーを解決します。キュー・マネージャー別名定義は、ターゲット・キュー・マネージャーへの伝送キューを決定するために使用することもできます。

解決されたキュー名がローカル・キューではない場合、アプリケーションが伝送キューに入れる各メッセージの伝送ヘッダーには、キュー・マネージャー名とキュー名の両方が収められます。

リモート・キュー定義またはキュー・マネージャー別名定義によって変更されない限り、通常、使用される伝送キューには、解決されたキュー・マネージャーの名前が付けられます。そのような伝送キューを定義していなくても、デフォルト伝送キューを定義していれば、そのキューが使われます。

z/OS で実行されるキュー・マネージャーの名前は 4 文字までに制限されています。

キュー・マネージャー別名定義

メッセージを書き込むためにキューをオープンするアプリケーションがキュー名、およびキュー・マネージャー名を指定するとき、キュー・マネージャー別名定義が適用されます。

キュー・マネージャー別名の定義には、次の3つの使用方法があります。

- メッセージを送信し、キュー・マネージャー名を再マップするとき
- メッセージを送信し、伝送キューを変更または指定するとき
- メッセージを受信し、ローカル・キュー・マネージャーがこれらのメッセージの宛先を指しているかどうか判定するとき

アウトバウンド・メッセージ - キュー・マネージャー名の再マップ

キュー・マネージャー別名定義は、MQOPEN 呼び出しに指定されるキュー・マネージャー名を再マップするために使用できます。例えば、MQOPEN 呼び出しは、THISQ のキュー名と YOURQM のキュー・マネー

ジャー名を指定します。ローカル・キュー・マネージャーには、次の例のようなキュー・マネージャー別名定義があります。

```
DEFINE QREMOTE (YOURQM) RQMNAME(REALQM)
```

これは、アプリケーションがキュー・マネージャー YOURQM にメッセージを書き込むときに、使用される実際のキュー・マネージャーが REALQMであることを示しています。ローカル・キュー・マネージャーが REALQM の場合、メッセージはローカル・キューの、キュー THISQ に書き込まれます。ローカル・キュー・マネージャーが REALQM ではない場合、メッセージは REALQM という伝送キューにルーティングされます。キュー・マネージャーは、伝送ヘッダーを変更して YOURQM の代わりに REALQM を指定します。

アウトバウンド・メッセージ - 伝送キューの変更および指定

62 ページの図 16 は、キュー・マネージャー QM3 でのキュー名を示す伝送ヘッダーを持ったメッセージが、キュー・マネージャー QM1 に到着するシナリオを示しています。このシナリオでは、QM3 へは QM2 を介するマルチ・ホップを使用して到達できます。

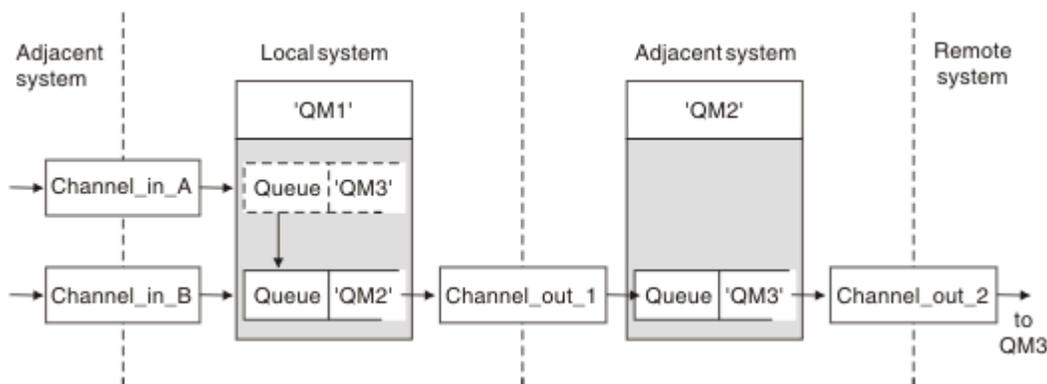


図 16. キュー・マネージャー別名

QM3 へのすべてのメッセージは、キュー・マネージャー別名を使用して QM1 で取り込まれます。キュー・マネージャーの別名は QM3 であり、これには、伝送キュー QM2 を介した定義 QM3 が含まれています。定義は、以下の例のようになります。

```
DEFINE QREMOTE (QM3) RNAME(' ') RQMNAME(QM3) XMITQ(QM2)
```

キュー・マネージャーはメッセージを伝送キュー QM2 に書き込みますが、宛先キュー・マネージャーの名前 QM3 が変わらないため、伝送キューのヘッダーを変更しません。

QM2 でのキュー名の入った伝送ヘッダーを示すメッセージが QM1 に着信すると、すべて QM2 伝送キューにも書き込まれます。このようにして、宛先の異なるメッセージが共通の伝送キューに収集され、該当の隣接システムに送られて、さらに宛先に転送されます。

インバウンド・メッセージ - 宛先の決定

受信側 MCA は、伝送ヘッダーで参照されるキューをオープンします。参照されるキュー・マネージャーと同じ名前前のキュー・マネージャー別名定義が存在する場合、伝送ヘッダーに受信されるキュー・マネージャー名は、その定義からの RQMNAME と置き換わります。

このプロセスには、次の 2 つの使用法があります。

- 別のキュー・マネージャーにメッセージを送信する
- キュー・マネージャー名をローカル・キュー・マネージャーと同じ名前に変更する

応答先キュー別名の定義

応答先キューの別名定義は、メッセージ記述子の応答情報の代替名を指定します。この利点は、アプリケーションを変更しないでキューまたはキュー・マネージャーの名前を変更できることです。

キュー名の解決

アプリケーションは、メッセージに回答するとき、受信したメッセージのメッセージ記述子内のデータを使用して、応答先のキューの名前を確かめます。送信側アプリケーションは、応答が送信される場所を示し、この情報をそのメッセージに付加します。この概念は、アプリケーション設計時に調整する必要があります。

キュー名の解決は、メッセージがキューに書き込まれる前にアプリケーションの送信側で行われます。この例は、キュー名の解決の一般的ではない使用方法です。キューがオープンされていないときにネーム・レゾリューションが一度に行われる唯一の状況です。キュー名の解決は、メッセージの送信先のリモート・アプリケーションと対話する前に生じます。

キュー・マネージャー別名を使用するキュー名の解決

通常、アプリケーションは、応答先キューを指定し、応答先キュー・マネージャー名を空白にしておきます。キュー・マネージャーは、書き込み時にその固有の名前を書き込みます。この方式が唯一首尾よくいかないのは、例えば、伝送キュー QM1 を使用するデフォルト返送チャンネルの代わりに、伝送キュー QM1_relief を使うチャンネルなどの、代替チャンネルを応答で使用したい場合です。この状況では、伝送キューのヘッダーに指定されるキュー・マネージャー名は「実際の」キュー・マネージャー名と一致しませんが、キュー・マネージャーの別名定義を使用して再定義されます。代替経路で応答を返すには、応答先キュー別名定義を使用して、応答先キュー・データも同様にマップする必要があります。

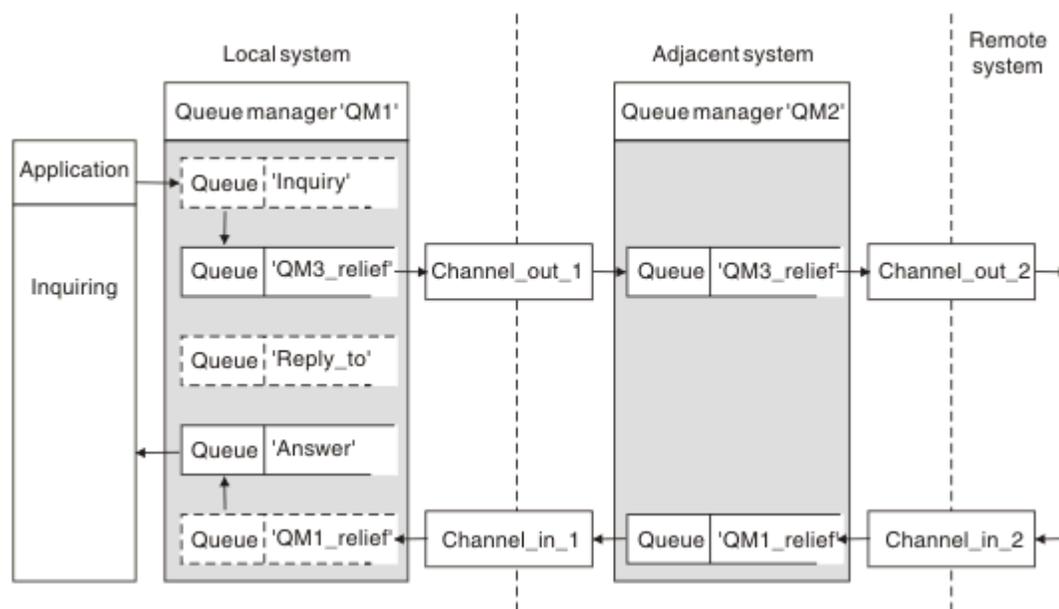


図 17. 応答ロケーションの変更に使用される応答先キューの別名

63 ページの図 17 の例の説明:

1. アプリケーションは、MQPUT 呼び出しを使用し、メッセージ記述子に以下のような情報を指定して、メッセージを書き込みます。

```
ReplyToQ='Reply_to'  
ReplyToQMgr=''
```

ReplyToQMgr は、使用される応答先キュー別名用で、空白にしておく必要があります。

2. Answer という名前と QM1_relief というキュー・マネージャー名の付いた、応答先キュー別名定義 Reply_to を作成します。

```
DEFINE QREMOTE ('Reply_to') RNAME ('Answer')
RQMNAME ('QM1_relief')
```

3. メッセージは、ReplyToQ='Answer' および ReplyToQMgr='QM1_relief' を示すメッセージ記述子付きで送信されます。
4. アプリケーションの仕様には、応答が Reply_to ではなくキュー Answer になければならないという情報が必要です。

応答に備えるには、以下のように定義をして、並列戻りチャンネルを作成する必要があります。

- QM2 で、伝送キュー定義 QM1_relief

```
DEFINE QLOCAL ('QM1_relief') USAGE(XMITQ)
```

- QM1 で、キュー・マネージャー別名 QM1_relief

```
DEFINE QREMOTE ('QM1_relief') RNAME() RQMNAME(QM1)
```

このキュー・マネージャー別名は、並列戻りチャンネルのチェーンを終了させ、QM1 へのメッセージを取り込みます。

実際の運用が将来の場合でも、アプリケーションが最初から別名を使用することを確認してください。現在は、これは応答先キューへの通常のキュー別名ですが、あとでキュー・マネージャー別名に変更することができます。

応答先キュー名

応答先キューの命名には注意が必要です。アプリケーションが応答先キュー名をメッセージに組み込むのは、応答の送り先となるキューを指定できるようにするためです。この名前を使用して応答先キュー別名定義を作成すると、実際の応答先キュー（すなわち、ローカル・キュー定義）と同じ名前を使用できなくなります。したがって、応答先キュー別名定義には、キュー・マネージャー名の他に新しいキュー名を入れる必要があり、また、この別のキューに応答が入ることを示す情報がアプリケーションの仕様に入っていないければなりません。

こうして、アプリケーションは、元のメッセージを書き込むときに応答先キューとして指定したキューとは別のキューからメッセージを取り出す必要があります。

クラスターが機能する仕組み

クラスターとは何か、およびその働きについて説明します。

クラスターとは、何らかの形態で論理的に関連付けられたキュー・マネージャーのネットワークです。クラスターを構成する各キュー・マネージャーは、物理的に遠く離れた場所に置くことができます。例えば、各キュー・マネージャーが世界各国に展開されているチェーン店の各支店を表すようにし、それらを異なる国に物理的に配置することができます。企業内のクラスターにはそれぞれ固有の名前を付ける必要があります。

通常、クラスターは、何らかの概念に基づいて論理的に関連付けられ、特定のデータまたはアプリケーションを共用する必要があるキュー・マネージャーで構成されています。例えば、社内の各部署につき1つのキュー・マネージャーを設定して、各部署ごとのデータおよびアプリケーションを管理している場合、これらすべてのキュー・マネージャーを1つのクラスターとしてまとめると、各キュー・マネージャーから給与計算アプリケーションにデータを提供できるようになります。あるいは、チェーン店の各支店につき1つのキュー・マネージャーを設定して、各支店ごとに在庫量などの情報を管理している場合は、これらのキュー・マネージャーを1つのクラスターとしてまとめると、各キュー・マネージャーが同じ販売・購入アプリケーションにアクセスできるようになります。本社のキュー・マネージャーで販売・購入アプリケーションを集中管理できます。

クラスターをセットアップすると、チャンネル定義やリモート・キュー定義をさらに定義しなくても、そのクラスター内のキュー・マネージャー間で相互に通信できるようになります。

キュー・マネージャーで構成される既存のネットワークを変換してクラスターに入れるか、または、新しいネットワークのセットアップの一部としてクラスターを確立することができます。

IBM WebSphere MQ クライアントは、他のキュー・マネージャーに接続できるのと同じように、クラスターを構成しているキュー・マネージャーに接続できます。

クラスターはワークロード管理用に使用することもできます。詳細については、[クラスターによるワークロードの管理](#)を参照してください。

クラスター内でメッセージが経路指定される方法

これまで IBM WebSphere MQ および分散キューイング機能を使用してこられた方は、クラスターを、信頼できるシステム管理者によって管理されているキュー・マネージャーのネットワークのようなものであると考えてください。クラスター・キューを定義すると、システム管理者は対応するリモート・キューの定義を必要に応じて他のキュー・マネージャーに対して自動的に作成します。

IBM WebSphere MQ のクラスター内の各キュー・マネージャーには伝送キューが1つ設定されているので、伝送キューを定義する必要はありません。この単一の伝送キューにより、クラスター内のその他のキュー・マネージャーにメッセージを送信することができます。伝送キューを1つしか使用できないという制限はありません。キュー・マネージャーは複数のクラスターを使用して、クラスター内の各キュー・マネージャーにメッセージを別々に送信できます。通常、1つのキュー・マネージャーは1つのクラスター伝送キューを使用します。キュー・マネージャー属性 DEFCLXQ を変更して、キュー・マネージャーがクラスター内のキュー・マネージャーごとに異なるクラスター伝送キューを使用するように設定することもできます。クラスター伝送キューを手動で定義することもできます。

クラスターを構成するキュー・マネージャーはすべて、上述のような方法で処理を実行します。各キュー・マネージャーは、各キュー・マネージャー自体に関する情報とホストしているキューに関する情報を送信し、同じクラスター内にあるその他のキュー・マネージャーに関する情報を受信します。

受信した情報はリポジトリに格納されます。大部分のキュー・マネージャーには、必要な情報だけが格納されています。必要な情報とは、通信相手となるキュー・マネージャーおよび関連するキューについての情報です。各キュー・マネージャーは情報を部分リポジトリに保持します。一部の指定されたキュー・マネージャーには、クラスター内にあるすべてのキュー・マネージャーに関する全ての情報が入っている完全リポジトリが格納されています。

クラスターの一部になるために、キュー・マネージャーには、クラスター送信側チャンネルおよびクラスター受信側チャンネルの2つのチャンネルが必要です。

クラスター送信側チャンネルとは、送信側チャンネルに似た通信チャンネルです。キュー・マネージャーに手動で1つのクラスター送信側チャンネルを作成し、既にクラスターのメンバーである完全リポジトリに接続する必要があります。

クラスター受信側チャンネルとは、受信側チャンネルに似た通信チャンネルです。クラスター受信側チャンネルを1つ手動で作成する必要があります。このチャンネルはキュー・マネージャーの仕組みとして機能し、クラスター通信を受信します。

このキュー・マネージャーとクラスターの他のメンバーとの通信に必要な他のすべてのチャンネルは自動的に作成されます。

クラスターをサポートするプラットフォーム上にあるキュー・マネージャーは、クラスター構成にしなくても構いません。そのような場合には、クラスターを使用せずに、分散キューイング機能を引き続き使用することもできますし、分散キューイング機能とクラスターを合わせて使用することもできます。

クラスターの例

66 ページの図 18 に CLSTR1 という名前のクラスターのコンポーネントを示しています。

- このクラスターには、QM1、QM2、QM3 という3つのキュー・マネージャーがあります。

- QM1 および QM2 では、クラスター内のすべてのキュー・マネージャーとクラスター関連オブジェクトに関する情報のリポジトリをホストしています。このようなキュー・マネージャーを完全リポジトリ・キュー・マネージャーといいます。リポジトリは、図の中で陰影の付いた円柱で示されています。
- QM2 および QM3 では、このクラスター内にあるその他のキュー・マネージャーからアクセスできるいくつかのキューをホストしています。このクラスター内にあるその他のキュー・マネージャーからアクセスできるキューをクラスター・キューといいます。図の中で陰影の付いたキューの部分でクラスター・キューを表します。クラスター・キューへはクラスター内のどこからでもアクセスできます。IBM WebSphere MQ クラスタリング・コードにより、これらのキューのリモート・キュー定義が、その定義を参照するすべてのキュー・マネージャーに必ず作成されるようになります。

分散キューイングの場合と同様に、アプリケーションは MQPUT 呼び出しを使用してクラスター内の任意のキュー・マネージャーにあるクラスター・キューにメッセージを書き込みます。アプリケーションは MQGET 呼び出しを使用して、キューが存在するキュー・マネージャーのみにあるクラスター・キューからメッセージを取り出します。

- 各キュー・マネージャーには、メッセージを受信できる *cluster-name.queue-manager* と呼ばれるチャンネルの受信側に対して、手動で作成された定義があります。受信側のキュー・マネージャーでは、*cluster-name.queue-manager* はクラスター受信側チャンネルです。クラスター受信側チャンネルは分散キューイングで使用されている受信側チャンネルに似ており、キュー・マネージャーのメッセージを受信します。さらに、クラスターについての情報も受け取ります。

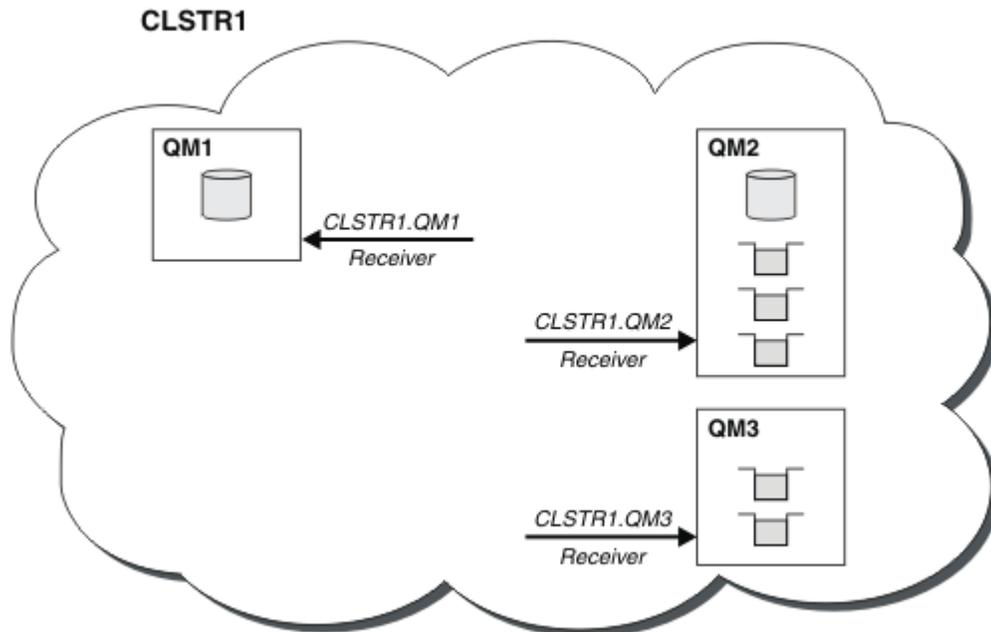


図 18. 複数のキュー・マネージャーで構成されるクラスター

- 67 ページの図 19 では、各キュー・マネージャーに、チャンネルの送信側であることを示す定義もあります。送信側のチャンネルは、いずれかの完全リポジトリ・キュー・マネージャーのクラスター受信側チャンネルに接続しています。送信側キュー・マネージャーでは、*cluster-name.queue-manager* はクラスター送信側チャンネルです。QM1 および QM3 のクラスター送信側チャンネルは CLSTR1.QM2 に接続されています。点線「2」を参照してください。

QM2 のクラスター送信側チャンネルは CLSTR1.QM1 に接続されています。点線「3」を参照してください。クラスター送信側チャンネルは、分散キューイングで使用されている送信側チャンネルに似ており、受信側キュー・マネージャーにメッセージを送信します。さらに、クラスターについての情報も送信します。

クラスター受信側チャンネルとクラスター送信側チャンネルの両方の定義が終わると、自動的にチャンネルが起動します。

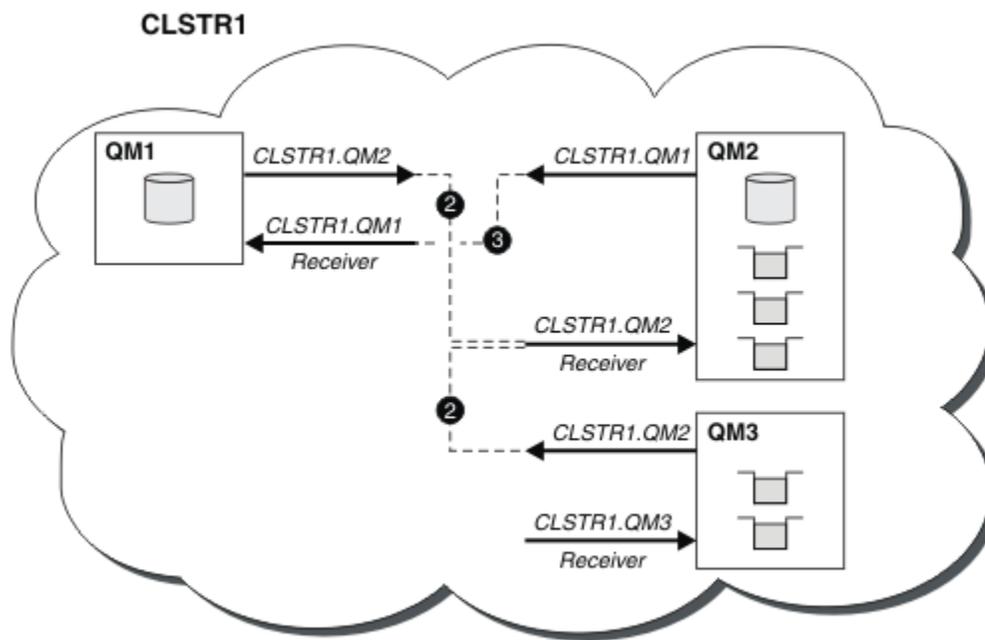


図 19. 送信側チャンネルのある複数のキュー・マネージャーで構成されるクラスター

クラスター機能の仕組み

ローカル・キュー・マネージャーにクラスター送信側チャンネルを定義すると、キュー・マネージャーがいずれかの完全リポジトリ・キュー・マネージャーで認識されます。これにより、その完全リポジトリ・キュー・マネージャーは完全リポジトリの情報を更新します。そして、元のキュー・マネージャー宛てのクラスター送信側チャンネルを自動的に作成して、そのキュー・マネージャーにクラスターの情報を送信します。このように、キュー・マネージャーとクラスターは相互に認識できます。

以降に、66 ページの図 18 に示すクラスターを再び例にとって説明します。例えば、キュー・マネージャー QM3 に接続されているアプリケーションから QM2 のキューにメッセージを送信するとします。QM3 が初めてそれらのキューにアクセスする際には、完全リポジトリを参照して、キューを見つけることができます。この場合の完全リポジトリは QM2 で、このリポジトリへは送信側チャンネル CLSTR1.QM2 を使用してアクセスします。リポジトリからの情報を使用して、それらのキュー用にリモート定義を自動的に作成することができます。キューが QM1 にある場合でも、QM2 が完全リポジトリであるため、このメカニズムは引き続き機能します。完全リポジトリは、クラスター内のすべてのオブジェクトの完全なレコードを保持しています。後者の事例では、QM3 は自動的に QM1 のクラスター受信側チャンネルに対応するクラスター送信側チャンネルを作成することもでき、これら 2 つの間で直接通信が可能になります。

68 ページの図 20 は同じクラスターを示していますが、ここでは、自動的に作成された 2 つのクラスター送信側チャンネルが追加されています。クラスター送信側チャンネルは、クラスター受信側チャンネル CLSTR1.QM3 につながっている 2 本の破線で示されています。また、この図に示されているクラスター伝送キュー SYSTEM.CLUSTER.TRANSMIT.QUEUE は、QM1 がメッセージを送信するときに使用するキューです。クラスター伝送キューは、クラスター内のどのキュー・マネージャーにもあります。このキューにより、各キュー・マネージャーは同じクラスター内のその他のキュー・マネージャーにメッセージを送信することができます。

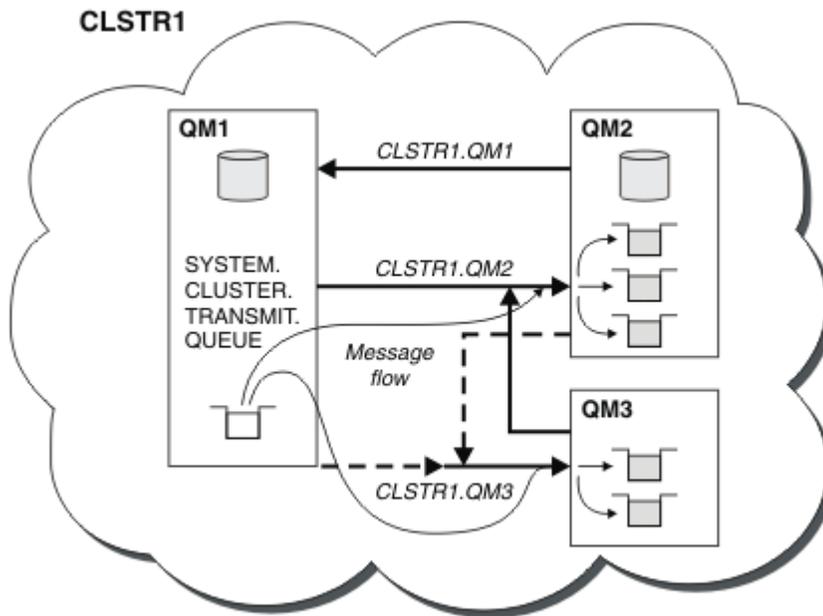


図 20. 複数のキュー・マネージャーで構成されるクラスター (自動定義チャンネル付き)

注: 他の図は、手動で定義を作成する、チャンネルの受信側のみを示しています。送信側は、必要な場合に自動的に定義されることが多いので、省略しています。多くのクラスター送信側チャンネルの自動定義は、クラスターの機能と効率に影響を及ぼす非常に重要なものです。

関連概念

クラスター

キュー・マネージャーは、クラスターにまとめることができます。キュー・マネージャーを1つのクラスターとしてまとめると、クラスターを構成するキュー・マネージャーがホスト管理しているキューは、そのクラスター内の他のキュー・マネージャーも使用することができます。同じクラスター内にあるキュー・マネージャー間では、標準の分散キューイングで必要なオブジェクト定義の多くを使用せずに、メッセージを送信することができます。

クラスター化と分散キューイングとの比較

クラスターのコンポーネント

関連タスク

キュー・マネージャー・クラスターの構成

新規クラスターのセットアップ

WebSphere MQ クラスターの管理

IBM WebSphere MQ Telemetry

IBM WebSphere MQ Telemetry は、キュー・マネージャーの一部である遠隔測定 (MQXR) サービス、テレメトリー・クライアント (自分で記述することも、提供されているクライアントの1つを使用することもできる)、コマンド行インターフェース、およびエクスプローラー形式の管理インターフェースで構成されます。テレメトリーとは、幅広い種類のリモート・デバイスからデータを収集し、それらのデバイスを管理することです。IBM WebSphere MQ Telemetry を利用すると、データの収集と、デバイスの制御を Web アプリケーションに統合できます。

MQTT サポートは以前は、WebSphere MQ Telemetry が別個のフィーチャーであった WebSphere Message Broker または WebSphere MQ バージョン 7.0.1 で使用可能でした。WebSphere MQ Telemetry は WebSphere MQ Version 7.1 以降ではコンポーネントとなったため、アップグレード時には実質的に WebSphere MQ Telemetry バージョン 7.0.1 のアンインストールと WebSphere MQ バージョン 7.1 のインストールが行われます。WebSphere MQ Telemetry は、メインの製品と一緒にインストールすることも、バージョン 7.1 以降がインストールされた後にインストールすることもできます。マイグレーションにつ

いては、[Version 7.0.1 から Version 7.5 からの IBM WebSphere MQ Telemetry のマイグレーション](#)、または [WebSphere Message Broker バージョン 6 から IBM WebSphere MQ Telemetry および WebSphere Message Broker バージョン 7.0 を使用するためのテレメトリー・アプリケーションのマイグレーション](#)を参照してください。

IBM WebSphere MQ Telemetry には、以下のコンポーネントが含まれています。

遠隔測定チャンネル

MQTT クライアントと IBM WebSphere MQ との接続を管理するには、遠隔測定チャンネルを使用します。テレメトリー・チャンネルは、`SYSTEM.MQTT.TRANSMIT.QUEUE` などの新しい IBM WebSphere MQ オブジェクトを使用して IBM WebSphere MQ と対話します。

遠隔測定 (MQXR) サービス

MQTT クライアントは、`SYSTEM.MQXR.SERVICE` 遠隔測定サービスを使用して遠隔測定チャンネルに接続します。

IBM WebSphere MQ Explorer サポート IBM WebSphere MQ Telemetry

IBM WebSphere MQ Telemetry は、IBM WebSphere MQ Explorer を使用して管理できます。

クライアント Software Development Kit (SDK)

クライアント SDK には、以下の 4 つの部分が含まれます。

1. Java SE および Java ME 用の MQTT v3 クライアント・ライブラリー。この Java ライブラリーは、Java SE または Java ME をサポートする装置用の Java クライアントを作成する場合に使用します。
2. C 用の MQTT v3 ライブラリー C ライブラリーを使用して、いくつかのプラットフォーム用の C クライアントを作成します。
3. IBM WebSphere MQ Telemetry daemon for devices。C で作成された、数多くのプラットフォームで稼働する拡張クライアントです。
4. MQTT v3 プロトコル。MQTT v3 プロトコルは公開プロトコルであり、再使用ライセンス付きです。さまざまなプラットフォームと言語で使用される MQTT クライアントを作成するには、このプロトコルを使用するとともに、MQTT クライアント実装を参照してください。

Documentation

IBM WebSphere MQ Telemetry 資料は、Version 7.1 の標準 IBM WebSphere MQ 製品資料に含まれています。Java クライアントおよび C クライアント用の SDK 資料は、Javadoc および HTML の形式で製品資料に用意されています。

遠隔測定の概念

何をすべきかを定める際は、自分を取り巻く環境から情報を収集するものです。例えば消費者の立場では、どの食料を買うかを定める前に、蓄えとして何があるかを確認します。乗り継ぎを予約する前に、今出発すると移動にどれだけ時間がかかるかを知りたいでしょう。医者に診てもらおうかどうかを決める前に、自分の症状を確認します。バスを待つかどうかを決める前に、バスの到着予定時刻を確認します。こうした意思決定のための情報は、メーターと装置から、または紙に書かれた言葉または画面から、あるいは自分自身から、直接もたらされます。どこにしようと、いつやらなければならないとしても、情報を収集してまとめ、分析し、それに基づいて行動します。

情報源が広く分散していたりアクセスできなかつたりする場合は、最も正確な情報を収集することは難しくなり、コストもかかるようになります。加える変更が多い場合や変更することが難しい場合は、変更がなされないか、あまり効果的でないときに変更が行われます。

広く分散した装置からの情報収集とそれらの装置の制御のコストが、デジタル・テクノロジーを搭載した装置をインターネットに接続することによって大幅に削減されるとしたら、どうなるでしょう。インターネットと企業のリソースを使って情報を分析することができます。情報で裏付けられた判断をし、それに基づいて行動する機会が増えます。

テクノロジー・トレンドと環境面、経済面のプレッシャーが、以下のような変化をもたらしています。

1. 標準化と低価格デジタル・プロセッサへの接続により、センサーとアクチュエーターの接続と制御のコストが削減されています。

2. 装置を接続するのにインターネットとインターネット・テクノロジーがますます使われています。国によっては、インターネット・アプリケーションへの接続数の点で、携帯電話がパーソナル・コンピューターを上回っています。他の装置も確実に後を追っています。
3. インターネットとインターネット・テクノロジーは、アプリケーションによるデータの取得を格段に容易にします。データに簡単にアクセスできるので、センサーからのデータをさらに多くのソリューションで役立つ情報に変えるためのデータ分析論が使われるようになっていきます。
4. リソースを賢く使うことは、多くの場合に、炭酸ガス放出とコストを削減するための、より手っ取り早く安上がりな方法です。別の方法としては、新しいリソースを見つけるか、既存のリソースを利用するための新しいテクノロジーを開発することになりますが、この場合は長期的ソリューションになる可能性があります。短期間のうちに新しいテクノロジーを開発したり新しいリソースを見つけたりするのは、たいていは既存のソリューションを改良するよりもリスクがあり、時間とコストもかかります。

例

このような傾向が、環境とインテリジェントに連携する新しい機会をいかに生み出すかを、1つの例が示しています。

「海上における人命の安全のための国際条約」(SOLAS)は、多くの船舶に船舶自動識別装置(AIS)を配備することを義務付けています。AISを義務付けられているのは、300トンを超える商船と、客船です。AISは本来、沿岸運航を対象とした衝突回避システムです。管海官庁による沿岸水域の監視と管理に使われます。

世界中の熱心な人たちが、低価格AIS追跡局を配備して、沿岸運航情報をインターネット上に流しています。その一方で、AISからの情報をインターネットからの他の情報と結合するアプリケーションを作成している熱心な人たちもいます。結果はWebサイトに置かれ、TwitterやSMSを使って公開されます。

あるアプリケーションでは、サウサンプトン近辺のAIS局からの情報が、船舶所有権および地理情報と結合されます。このアプリケーションは、フェリーの到着とTwitterへの逸脱に関する情報をフィードします。サウサンプトンとワイト島を結ぶフェリーを使用している定期利用者は、TwitterやSMSを使ってニュース・フィードをサブスクライブしています。フェリーが遅れている場合は、定期便の到着時間が予定されている時間より遅く、出発を遅らせ、フェリーに乗船することができます。

さらに別の例については、[70ページの『モニターと制御に関するテレメトリー概念とシナリオ』](#)を参照してください。

関連タスク

[WebSphere MQ Telemetry のインストール](#)

[WebSphere MQ Telemetry の管理](#)

[WebSphere Message Broker v6 を使用している遠隔測定アプリケーションを WebSphere MQ Telemetry および WebSphere Message Broker v7 を使用するようにマイグレーションする](#)

[バージョン 7.0.1 からバージョン 7.5 への WebSphere MQ Telemetry の移行](#)

[WebSphere MQ Telemetry のアプリケーションの開発](#)

[WebSphere MQ Telemetry のトラブルシューティング](#)

関連資料

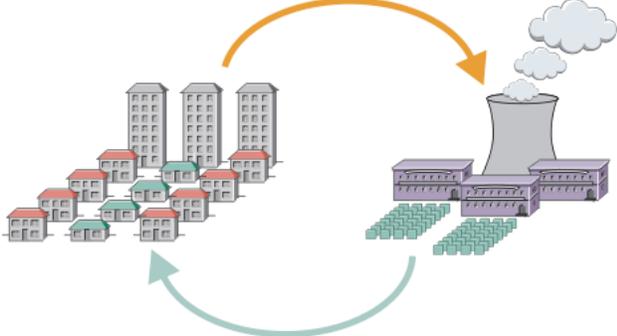
[WebSphere MQ Telemetry リファレンス](#)

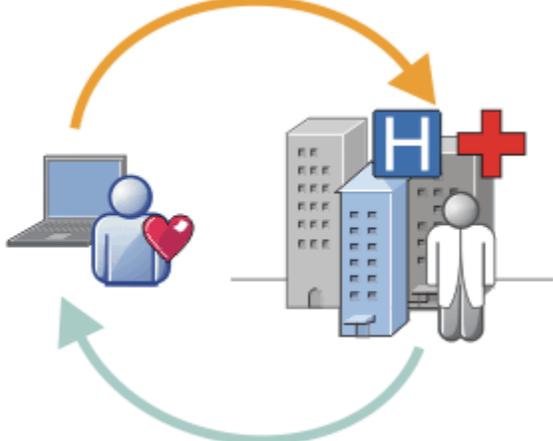
モニターと制御に関するテレメトリー概念とシナリオ

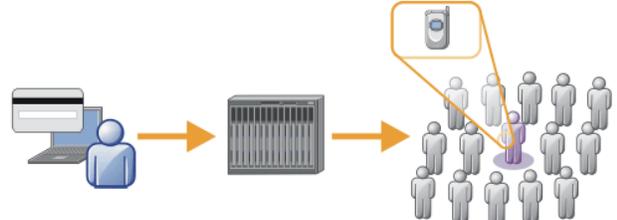
テレメトリーは、自動化されたセンシング、データの測定、およびリモート装置の制御を行うものです。特に重要なのは、装置から中央制御点へのデータ伝送です。遠隔測定には、構成情報と制御情報を装置に送信することも含まれます。

IBM WebSphere MQ Telemetry は、MQTT プロトコルを使用して小型装置を接続し、それらの装置を IBM WebSphere MQ を使用して他のアプリケーションに接続します。IBM WebSphere MQ Telemetry デバイスとインターネットの間のギャップを埋め、「スマート・ソリューション」の構築を容易にします。スマート・ソリューションは、装置をモニターして制御するアプリケーションのために、インターネットおよびエンタープライズ・アプリケーションで使用可能な情報の宝庫の錠を開けます。

以下の各図は、IBM WebSphere MQ Telemetry の代表的な使用例を示したものです。

Telemetry: スマート電力	
	<ul style="list-style-type: none"> • エネルギー使用量データを含んだ MQTT メッセージがサービス・プロバイダーに送られます。 • IBM WebSphere MQ Telemetry は、エネルギー使用量データの分析に基づいて制御コマンドを送ります。 • 詳しくは、「73 ページの『遠隔測定シナリオ: 家庭エネルギーのモニターと制御』」を参照してください。

Telemetry: スマート医療サービス	
<ul style="list-style-type: none"> • IBM WebSphere MQ Telemetry は、医療データを病院 & Doctor に送信します。 • 医療データの分析に基づいて、MQTT 警報やフィードバックが送られます。 • 詳しくは、「72 ページの『遠隔測定シナリオ: 在宅患者モニター』」を参照してください。 	

Telemetry: 大勢の中の一人	
	<ul style="list-style-type: none"> • シンプルなカード・トランザクションが銀行のサーバーに送られます。 • IBM WebSphere MQ Telemetry は、数千名の中から一人の個人を特定して、その顧客のカードが使用されているという警報を返します。 • IBM WebSphere MQ Telemetry は、最もシンプルな入力情報に基づいてその個人を特定します。

以下のシナリオは実例に基づいたものであり、遠隔測定のいくつかの使用法と、遠隔測定テクノロジーで解決されるはずである一般的な問題のいくつかを示しています。

関連概念

72 ページの『遠隔測定シナリオ: 在宅患者モニター』

心臓病患者ケア・システムに関する IBM と医療機関のコラボレーションでは、埋め込み除細動器が病院とコミュニケーションを取ります。患者と埋め込みデバイスに関するデータが、RF 遠隔測定を使用して患者の自宅内の MQTT 装置に転送されます。

73 ページの『遠隔測定シナリオ: 家庭エネルギーのモニターと制御』

75 ページの『遠隔測定シナリオ: Radio Frequency Identification (RFID)』

75 ページの『遠隔測定シナリオ: 環境センシング』

環境センシングでは、遠隔測定を使用して、河川の水位や質、大気汚染物質などの環境データに関する情報を収集します。

76 ページの『遠隔測定シナリオ: モバイル・アプリケーション』

モバイル・アプリケーションとは、ワイヤレス・デバイス上で動作するアプリケーションのことです。デバイスは、汎用アプリケーション・プラットフォームかカスタム・デバイスのどちらかです。

遠隔測定シナリオ: 在宅患者モニター

心臓病患者ケア・システムに関する IBM と医療機関のコラボレーションでは、埋め込み除細動器が病院とコミュニケーションを取ります。患者と埋め込みデバイスに関するデータが、RF 遠隔測定を使用して患者の自宅内の MQTT 装置に転送されます。

転送は通常、ベッドのそばに置かれた送信機に対して毎晩行われます。送信機は電話システムを介してデータを病院に安全に転送し、病院でデータが分析されます。

このシステムにより、患者が医師に診てもら回数が少なくなります。患者またはデバイスが要注意であることが検出され、緊急の場合には待機医師に通報されます。

IBM と医療機関のこのコラボレーションは、さまざまな遠隔測定シナリオに共通の以下の特性を持っています。

不可視性

電源と電話回線をつなぐことと一日の一部の時間に装置のすぐそばにいること以外に、装置はユーザー介入を必要としません。確実に操作でき、使い方も簡単です。

患者が装置をセットアップしなくてもよいように、装置サプライヤーが装置を事前構成します。患者は装置のプラグを差し込むだけでよいのです。患者による構成がないので、装置の操作が単純になり、装置が誤って構成される可能性が低くなります。

MQTT クライアントが装置の一部として組み込まれます。装置デベロッパーは装置に MQTT クライアント実装を組み込み、デベロッパーまたはサプライヤーが事前構成の一部として MQTT クライアントを構成します。

MQTT クライアントは、Java SE and Java ME jar ファイルとして出荷されます。開発者はこのファイルを Java アプリケーションに組み込みます。このシナリオのような非 Java 環境の場合、装置デベロッパーは MQTT の公開フォーマットと公開プロトコルを使用して、異なる言語でクライアントを実装できます。また、開発者は、Windows、Linux、および ARM プラットフォーム用の共有ライブラリーとして出荷される C クライアントの 1 つを使用することもできます。

不均等な接続性

除細動器と病院のコミュニケーションは、不均等なネットワーク特性を持ちます。患者からデータを収集することとそのデータを病院に送信することにつながるさまざまな問題を解決するために、2つの異なるネットワークが使用されます。患者と MQTT 装置の間には、近距離低出力 RF ネットワークが使用されます。送信機は、低帯域幅電話回線を介する VPN TCP/IP 接続を使用して病院に接続します。

何とか工夫してすべての装置をインターネット・プロトコル・ネットワークに直接接続することは、多くの場合、実際的ではありません。ハブによって接続された2つのネットワークを使用するのが、一般的なソリューションです。MQTT 装置は単純なハブであり、患者からの情報を保管し、それを病院に転送します。

機密保護

医師は患者データの確実性を信用できなければならず、患者は自分のデータのプライバシーが尊重されることを望んでいます。

VPN または SSL を使用して接続を暗号化するだけで十分なシナリオもあります。一方、保管されたデータもセキュリティー保護しておくことが望ましいシナリオもあります。

遠隔測定装置がセキュリティー保護されていないこともあります。例えば、共用住居の中に置かれる可能性があります。装置のユーザーは、正しい患者からのデータであることを確認するために認証されなければなりません。装置自体は SSL を使用してサーバーに対して認証することができ、またそのサーバーを装置に対して認証できます。

装置とキュー・マネージャーの間の遠隔測定チャンネルは、ユーザー認証のための JAAS と通信暗号化のための SSL、および装置認証をサポートします。パブリケーションに対するアクセス権限は、WebSphere MQ のオブジェクト権限マネージャーによって制御されます。

ユーザーを認証するために使用される ID は、共通患者 ID などの異なる ID にマップできます。共通 ID を使用すると、WebSphere MQ でパブリケーション・トピックに対する許可を構成するのが簡単になります。

接続性

MQTT 装置と病院の間の接続にはダイヤルアップが使用され、300 ボーという低帯域幅で動作します。

300 ボーで効果的に作動するように、MQTT プロトコルでは、TCP/IP ヘッダーのほかにさらにメッセージに追加されるのは数バイトだけです。

MQTT プロトコルには、待ち時間が少ない単一伝送の「応答不要送信」メッセージングが用意されています。応答時間よりも送達が保証されることの方が重要である場合は、「最低 1 回」および「正確に 1 回」の送達を保証する複数伝送も使用できます。送達を保証するために、メッセージは正常に送達されるまで装置で保管されます。装置がワイヤレス接続される場合に、保証送達は特に有用です。

拡張容易性

遠隔測定装置は通常、数万から数百万まで、大量に配備されます。

多数の装置をシステムに接続すると、大きな要求がソリューションに突きつけられます。装置とそのソフトウェアのコストなどのビジネス上の要求と、ライセンスや装置、ユーザーの管理という管理上の要求があります。技術上の要求としては、ネットワークおよびサーバーの負荷があります。

接続を開く際に、オープン接続を維持するよりも多くのサーバー・リソースを使用します。しかし、このような電話回線を使用するシナリオでは、接続の費用は、必要な時間しか接続が開かれていないことが重要になります。データ転送は、大体はバッチ処理される性質のものです。就寝時間前の突然の接続ピークを回避するために、接続のスケジュールを夜間に分散させることができます。

クライアントに関しては、必要とするクライアント構成が最低限のものであることが、クライアントの拡張容易性に寄与します。MQTT クライアントは装置に組み込まれます。構成ステップや MQTT クライアント・ライセンス承諾ステップを患者への装置の配備に組み込まなければならないという要件はありません。

サーバーに関しては、WebSphere MQ Telemetry の初期ターゲットは、キュー・マネージャー当たり 50,000 のオープン接続です。

接続は WebSphere MQ エクスプローラーを使用して管理されます。エクスプローラーは、表示する接続をフィルターに掛けて、管理しやすい数に絞り込みます。ID をクライアントに割り振るためのスキームを適切に選択すれば、地理に基づいて、または患者の名前のアルファベット順に、接続をフィルターに掛けることができます。

遠隔測定シナリオ: 家庭エネルギーのモニターと制御

スマート・メーターは、従来型メーターよりも詳細にエネルギー使用量について収集します。

スマート・メーターはしばしば、家庭内の個々の電気製品をモニターおよび制御するために、ローカル遠隔測定ネットワークと結合されます。ある距離を置いてモニターと制御を行うために、リモート接続されるものもあります。

リモート接続のセットアップを行う場合は、個人または電力事業者(つまり中央制御点)が行うこととなります。リモート制御点は電力使用量を読み取り、使用量データを提供できます。また、継続料金設定や気象情報など、使用量に影響するデータを提供できます。全体発電効率を上げるために、負荷を制限することもできます。

スマート・メーターは、広く配備され始めています。例えば英国政府は、2020 年までに英国のすべての家庭にスマート・メーターを配備することについて協議しています。

家庭メーターのシナリオは、以下のようないくつかの共通特性を持ちます。

不可視性

ユーザーがメーターを利用してエネルギー節約に踏み込むつもりでない限り、メーターはユーザー介入を必要としません。メーターが個々の電気製品へのエネルギー供給の信頼性を落としてはなりません。

MQTT クライアントはメーターと一体で配備されたソフトウェアに組み込むことができ、別個のインストールや構成を必要としません。

不均等な接続性

電気製品とスマート・メーターの間の通信は、メーターとリモート接続ポイントの間とは異なる接続標準を必要とします。

スマート・メーターから電気製品への接続は可用性が高くなければならず、Home Area Network のネットワーク標準に準拠していなければなりません。

リモート・ネットワークでは、さまざまな物理接続が使用されるでしょう。その中には、セルラーのように、伝送コストが高く、途切れることがあるものもあります。MQTT v3 仕様は、リモート接続、およびローカル・アダプターとスマート・メーターの間の接続を対象としています。

電源コンセントと電気製品の間の接続とメーターでは、Zigbee のような Home Area Network が使用されます。センサー・ネットワーク用 MQTT (MQTT-S) は、Zigbee などの低帯域幅ネットワーク・プロトコルと連動するように設計されています。WebSphere MQ Telemetry は、MQTT-S を直接はサポートしていません。MQTT-S を MQTT v3 に接続するためのゲートウェイが必要です。

在宅患者モニターのように、家庭エネルギーのモニターと制御のソリューションも複数のネットワークを必要とし、これらのネットワークはスマート・メーターをハブとして使用して接続されます。

機密保護

スマート・メーターに関連するいくつかのセキュリティー問題があります。トランザクションの否認防止、開始された制御アクションの許可、電力使用量データのプライバシーが問題として挙げられます。

プライバシーを確保するために、メーターとリモート制御点の間で MQTT により転送されるデータを、SSL を使用して暗号化できます。制御アクションの許可を確かなものにするために、メーターとリモート制御点の間の MQTT 接続を、SSL を使用して相互に認証できます。

接続性

リモート・ネットワークの物理的性質は、大きく異なる可能性があります。既存のブロードバンド接続を使用する場合もあれば、通話コストが高く使用可能状態が断続的なモバイル・ネットワークを使用する場合もあります。高コストで断続的な接続 MQTT は、効率的で信頼性の高いプロトコルです。[72 ページの『遠隔測定シナリオ: 在宅患者モニター』](#)を参照してください。

拡張容易性

電力会社 (つまり中央制御点) は、最終的に数千万台のスマート・メーターを配備することを計画しています。当初の配備ごとのメーター台数は、数万台から数十万台の規模です。この数は、MQTT の初期ターゲットであるキュー・マネージャー当たりのオープン・クライアント接続数 50,000 に匹敵します。

家庭エネルギーのモニターと制御のアーキテクチャーの重大な側面は、スマート・メーターをネットワーク・コンセントレーターとして使用する点です。電気製品用アダプターは、それぞれが別個のセンサーです。それらを MQTT を使用するローカル・ハブに接続することにより、ハブは中央制御点との単一 TCP/IP セッションにデータ・フローを集信することができます。また、セッション障害を乗り越えられるように、しばらくの間メッセージを保管できます。

家庭エネルギーのシナリオでは、2つの理由により、リモート接続を開いたままにしておく必要があります。第1に、接続を開くための時間が、要求を送信することと比較して長くなるためです。短い間隔で「負荷制限」"要求を送信するために接続をいくつも開いては、時間がかかりすぎてしまいます。第2に、電力会社から負荷制限要求を受信するためには、まずクライアントが接続を開かなければなりません。MQTT の場合、接続は常にクライアントによって開始されます。電力会社から負荷制限要求を受信するには、接続を開いたままにしておく必要があります。

接続を開く速度が重要である場合や、時間が重要な要求をサーバーが開始する場合、解決策としては通常、オープン接続を多数維持しておきます。

遠隔測定シナリオ: Radio Frequency Identification (RFID)

RFIDとは、対象をワイヤレスで識別および追跡するために組み込み RFID タグを使用することです。RFID タグは、RFID リーダーの照準に入っていれば、最大で数メートルの距離から読み取ることができます。パッシブ・タグは RFID リーダーによってアクティブ化されます。アクティブ・タグは、外部からの活動化がなくても送信します。アクティブ・タグには給電部がなければなりません。パッシブ・タグにも、距離を広げるために給電部を組み込むことができます。

RFID は多くのアプリケーションで使用されており、シナリオのタイプは実にさまざまです。RFID のシナリオ、在宅患者モニターのシナリオ、家庭エネルギーのモニターおよび制御のシナリオには、類似点と相違点がいくつかあります。

不可視性

多くのシナリオでは、RFID リーダーは大量に配備されるため、ユーザー介入なしで動作しなければなりません。リーダーには、中央制御点と通信するための組み込み MQTT クライアントが搭載されます。

例えば、流通倉庫では、リーダーはモーション・センサーを使用してパレットを検出します。パレットに積まれたアイテムの RFID タグをアクティブ化し、中央アプリケーションにデータと要求を送信します。データは、在庫の場所を更新するために使用されます。要求は、特定のベイに移動するなどの、パレットに対する次のアクションを制御します。航空会社と空港のバゲージ (荷物) システムでは、RFID がこのように使用されています。

一部の RFID シナリオでは、リーダーに Java ME などの標準コンピューティング環境があります。このような場合は、製造後に別個の構成ステップで MQTT クライアントが配備される場合があります。

不均等な接続性

RFID リーダーが MQTT クライアント搭載のローカル制御装置とは別になっている場合もあれば、各リーダーに MQTT クライアントが組み込まれている場合もあります。通常は、地理的要因や通信上の要因から、トポロジーを選択することになります。

機密保護

RFID タグの添付におけるセキュリティ上の懸念は、プライバシーと認証性です。RFID タグは人目に付かないため、モニターやスプーフ、改ざんがひそかに行われるおそれがあります。

RFID のセキュリティ問題のソリューションにより、新しい RFID ソリューションを配備する機会が増えています。RFID タグとローカル・リーダーにセキュリティ上の脆弱性があるものの、中央情報処理を使用することが、さまざまな脅威に対抗するためのアプローチになります。例えば、タグの改ざんは、デリバリーとディスパッチの在庫基準を動的に相互に関連付けることにより、検出される可能性があります。

接続性

RFID アプリケーションには通常、RFID リーダーと即時照会から集められた情報のバッチ型ストア・アンド・フォワードが含まれます。流通倉庫シナリオでは、RFID リーダーは常に接続されています。タグが読み取られると、リーダーに関する情報と共にパブリッシュされます。倉庫アプリケーションが、リーダーに応答をパブリッシュして戻します。

倉庫アプリケーションでは、ネットワークは一般的に信頼性が高いので、即時要求ではパフォーマンス面で待ち時間の少ない「応答不要送信」メッセージが使用されることがあります。データのバッチ型ストア・アンド・フォワードでは、データ脱落に伴う管理コストを最小にするために、「正確に 1 回」メッセージングが使用されることがあります。

拡張容易性

RFID アプリケーションが 1 秒か 2 秒程度の即時応答を必要とする場合、RFID リーダーは接続されたままではなりません。

遠隔測定シナリオ: 環境センシング

環境センシングでは、遠隔測定を使用して、河川の水位や質、大気汚染物質などの環境データに関する情報を収集します。

センサーはしばしば、有線通信への到達経路のない遠隔地に置かれます。無線帯域幅はコストがかかり、信頼性が低い場合があります。通常は、地理上の小区域内のいくつかの環境センサーが、安全な場所に設

置されたローカル・モニター装置に接続されます。このローカル接続は、有線の場合もあれば、無線の場合もあります。

不可視性

センサー・デバイスは、中央モニター装置よりも、その場所に行きにくく、低電力駆動であり、多数配備される傾向があります。センサーが「ダム」(無言)の場合もあり、ローカル・モニター装置には、センサー・データを変換して保管するためのアダプターが組み込まれます。モニター・デバイスには、Java SE または ME をサポートする汎用コンピューターが組み込まれている可能性があります。不可視性は、MQTT クライアントを構成するときの大きな要件にはならないでしょう。

不均等な接続性

センサーの能力と、リモート接続のコストおよび帯域幅は通常、中央サーバーに接続されたローカル・モニター・ハブに依存することになります。

機密保護

このソリューションが軍や防衛のシナリオで使用されない限り、セキュリティーは大きな要件ではありません。

接続性

多くの場合、連続してモニターすることも、データを直ちに使用できることも、必須ではありません。洪水水位警報などの例外データは、直ちに転送されなければなりません。センサー・データは、接続コストと通信コストを削減するためにローカル・モニターで集約され、予定接続を使用して転送されます。例外データは、モニターで検出されると直ちに転送されます。

拡張容易性

センサーはローカル・ハブの周囲に集められ、センサー・データはパケットに集約されて、スケジュールに従って送信されます。これらの要素は両方とも、直接接続されたセンサーを使用した場合に中央サーバーにかかる負荷を軽減します。

遠隔測定シナリオ: モバイル・アプリケーション

モバイル・アプリケーションとは、ワイヤレス・デバイス上で動作するアプリケーションのことです。デバイスは、汎用アプリケーション・プラットフォームかカスタム・デバイスのどちらかです。

汎用プラットフォームとしては、電話や携帯情報端末などのハンドヘルド・デバイスと、ノートブック・コンピューターなどのポータブル・デバイスがあります。カスタム・デバイスでは、特定のアプリケーションに合わせた特殊目的のハードウェアが使用されます。「受領署名」小荷物配達を記録するためのデバイスは、カスタム・モバイル・デバイスの 1 例です。カスタム・モバイル・デバイス上のアプリケーションは、多くの場合、汎用ソフトウェア・プラットフォーム上で作成されます。

不可視性

カスタム・モバイル・アプリケーションのデプロイメントは管理されるので、MQTT クライアント・アプリケーションの構成を含めることができます。不可視性は、MQTT クライアントを構成するときの大きな要件にはならないでしょう。

不均等な接続性

これまでのシナリオのローカル・ハブ・トポロジーとは異なり、モバイル・クライアントはリモートで接続します。クライアント・アプリケーション層は、中央ハブでアプリケーションに直接接続します。

機密保護

物理的セキュリティーがほとんどないため、モバイル・デバイスとモバイル・ユーザーは認証されなければなりません。デバイスの ID の確認には SSL、ユーザーの認証には JAAS が使用されます。

接続性

モバイル・アプリケーションは、無線可能区域に依存する場合は、オフラインで作動できなければならず、また中断接続に効率的に対処できなければなりません。こうした環境では、接続されたままであることが目標とはいえ、アプリケーションはメッセージのストア・アンド・フォワードを実行できなければなりません。メッセージは注文や配達確認であることが多く、重要なビジネス価値を持っています。メッセージのストア・アンド・フォワードを確実に行う必要があります。

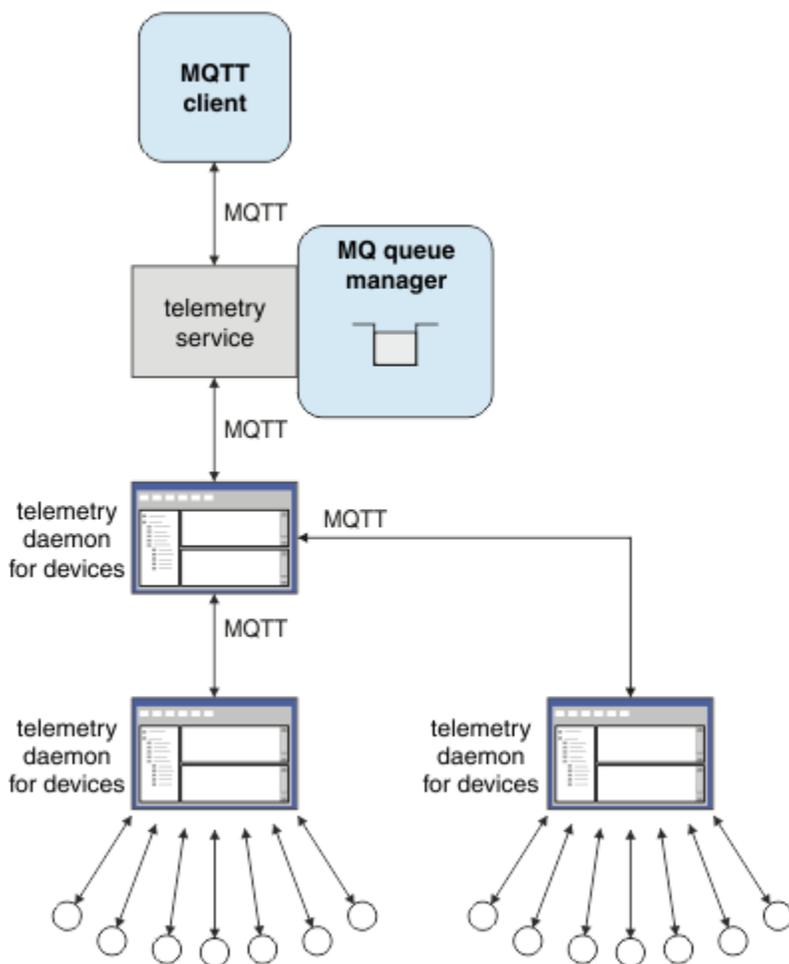
拡張容易性

拡張容易性は大きな課題ではありません。アプリケーション・クライアントの数は、カスタム・モバイル・アプリケーションのシナリオの場合、数千あるいは数万を超えることはないように思われます。

キュー・マネージャーへの遠隔測定装置の接続

遠隔測定装置は、MQTT v3 クライアントを使用してキュー・マネージャーに接続しています。MQTT v3 クライアントは、遠隔測定 (MQXR) サービスと呼ばれる TCP/IP リスナーに、TCP/IP を使用して接続します。

遠隔測定サービスに遠隔測定装置を直接接続する代わりに、装置を WebSphere MQ Telemetry デモン (デバイス用) に接続できます。デーモン自体が MQTT v3 クライアントとなります。装置の接続をプールして、遠隔測定 (MQXR) サービスへの単一接続を行います。複数のデーモンを階層構造で接続して、重大度による複数の順序で IBM WebSphere MQ に間接的に接続する装置の数を増やすことができます。



MQTT クライアントは、`MqttClient.connect` メソッドを使用して TCP/IP 接続を開始します。IBM WebSphere MQ クライアントのように、MQTT クライアントは、メッセージを送受信するにはキュー・マネージャーに接続されなければなりません。接続は、遠隔測定 (MQXR) サービスと呼ばれる TCP/IP リスナー (IBM WebSphere MQ Telemetry と共にインストールされる) を使用してサーバーで行われます。各キュー・マネージャーは、最大1つの遠隔測定 (MQXR) サービスを実行します。

遠隔測定 (MQXR) サービスは、各クライアントによって `MqttClient.connect` メソッドで設定されたりリモート・ソケット・アドレスを使用して、接続を遠隔測定チャンネルに割り振ります。ソケット・アドレスは、TCP/IP ホスト名とポート番号の組み合わせです。同じリモート・ソケット・アドレスを使用する複数のクライアントが、遠隔測定 (MQXR) サービスによって同一遠隔測定チャンネルに接続されます。

サーバー上に複数のキュー・マネージャーがある場合は、キュー・マネージャー間で遠隔測定チャンネルを分割します。これらのキュー・マネージャー間で、リモート・ソケット・アドレスを割り振ります。各遠

隔測定チャンネルに、固有のリモート・ソケット・アドレスを定義します。2つの遠隔測定チャンネルが同じソケット・アドレスを使用してはなりません。

複数のキュー・マネージャーの遠隔測定チャンネルに同じリモート・ソケット・アドレスを構成した場合は、最初に接続する遠隔測定チャンネルが成功します。同じアドレスで接続する後続のチャンネルは失敗し、**first-failure data capture (FDC)** ファイルを作成します。

サーバー上に複数のネットワーク・アダプターがある場合は、遠隔測定チャンネル間でリモート・ソケット・アドレスを分割します。特定のソケット・アドレスが1つの遠隔測定チャンネルのみに構成されている限り、ソケット・アドレスの割り振りは完全に任意となります。

IBM WebSphere MQ エクスプローラーのための WebSphere MQ Telemetry 補足で提供されるウィザードを使用して、MQTT クライアントを接続するように IBM WebSphere MQ を構成します。あるいは、[Telemetry 対応キュー・マネージャーの構成 \(Linux および AIX\) および Windows 上のテレメトリー用キュー・マネージャーの構成](#)の説明に従って、手動で遠隔測定を構成します。

関連資料

[MQXR プロパティ](#)

Telemetry 接続プロトコル

WebSphere MQ Telemetry は、TCP/IP IPv4 および IPv6 と SSL をサポートします。

遠隔測定 (MQXR) サービス

遠隔測定 (MQXR) サービスは TCP/IP リスナーであり、IBM WebSphere MQ サービスとして管理されます。このサービスは、IBM WebSphere MQ Explorer のウィザード、または **runmqsc** コマンドを使用して作成します。

IBM WebSphere MQ Telemetry (MQXR) サービスは、**SYSTEM.MQXR.SERVICE** と呼ばれます。

IBM WebSphere MQ Telemetry の補足 IBM WebSphere MQ Explorer に用意されている **Telemetry sample configuration** ウィザードは、テレメトリー・サービスとサンプル・テレメトリー・チャンネルを作成します。[IBM WebSphere MQ Explorer を使用した IBM WebSphere MQ Telemetry のインストールの検査](#)を参照してください。コマンド行からサンプル構成を作成します。[コマンド行を使用した IBM WebSphere MQ Telemetry のインストールの検査](#)を参照してください。

遠隔測定 (MQXR) サービスは、キュー・マネージャーと同時に自動的に開始および停止します。IBM WebSphere MQ Explorer でサービス・フォルダーを使用して、サービスを制御します。サービスを表示するには、エクスプローラーが表示から **SYSTEM** オブジェクトを除外するのを停止するためのアイコンをクリックする必要があります。

[installMQXRService_unix.mqsc](#) は、AIX および Linux でサービスを手動で作成する方法の例を示しています。[installMQXRService_win.mqsc](#) は、Windows でサービスを手動で作成する方法を示しています。

遠隔測定チャンネル

遠隔測定チャンネルを作成して、Java 認証・承認サービス (JAAS) または SSL 認証などのさまざまなプロパティを持つ接続を作成したり、クライアントのグループを管理したりします。

IBM WebSphere MQ Explorer の IBM WebSphere MQ Telemetry 補足資料で提供されている **New Telemetry Channel** ウィザードを使用して、テレメトリー・チャンネルを作成します。このウィザードを使用して、特定の TCP/IP ポートで MQTT クライアントからの接続を受け入れるようにチャンネルを構成します。Version 7.1 以降、コマンド行プログラム **runmqsc** を使用して IBM WebSphere MQ Telemetry を構成できます。

さまざまなポートについて複数の遠隔測定チャンネルを作成し、クライアントをグループに分けることによって、多数のクライアント接続を管理しやすくします。遠隔測定チャンネルごとに別々の名前を持ちます。

セキュリティ属性が異なる遠隔測定チャンネルを構成して、異なるタイプの接続を作成することができます。複数のチャンネルを作成して、さまざまな TCP/IP アドレスでクライアント接続を受け入れるようにします。SSL を使用して、メッセージを暗号化し、遠隔測定チャンネルとクライアントを認証するようにします ([MQTT クライアントおよびテレメトリー・チャンネルの SSL 構成](#)を参照)。ユーザー ID を指定して、

WebSphere MQ オブジェクトへのアクセスの許可を単純化します。JAAS 構成を指定して、MQTT ユーザーを JAAS で認証します。[MQTT クライアントの識別、許可、および認証を参照してください](#)。

MQTT プロトコル

MQ Telemetry Transport (MQTT) v3 プロトコルは、低帯域幅または高コスト接続を使用する小型デバイス間のメッセージ交換用に設計されており、メッセージを確実に送信することを目的としています。TCP/IP を使用します。

MQTT プロトコルは公開されています ([MQ Telemetry Transport のフォーマットおよびプロトコルを参照](#))。プロトコルのバージョン 3 は、パブリッシュ/サブスクライブを使用し、「応答不要送信」、「最低 1 回」、「正確に 1 回」の 3 種類のサービス品質をサポートします。

プロトコル・ヘッダーのサイズが小さいことと、メッセージ・ペイロードがバイト配列であることから、小さいメッセージになります。ヘッダーは、2 バイト固定ヘッダーと、最大 12 バイトの追加可変ヘッダーから成ります。プロトコルは 12 バイトの可変ヘッダーを使用してサブスクライブと接続を行い、ほとんどのパブリケーションには、わずか 2 バイトの可変ヘッダーを使用します。

3 種類のサービス品質がサポートされているので、待ち時間の少なさと信頼性の間でトレードオフが可能です (MQTT クライアントによって提供されるサービス品質を参照)。「応答不要送信」は持続的デバイス・ストレージを使用せず、パブリケーションの送受信に伝送を 1 回だけ行います。「最低 1 回」と「正確に 1 回」は、プロトコル状態を維持して確認応答があるまでメッセージを保存するための持続的ストレージをデバイス上に必要とします。

このプロトコルは、他の製品で使用される MQTT プロトコルのファミリーの 1 つです。

MQTT クライアント

MQTT クライアント・アプリは、遠隔測定装置からの情報収集、サーバーへの接続、およびサーバーへの情報のパブリッシュを担います。さらに、トピックにサブスクライブする、パブリケーションを受信する、遠隔測定装置を制御するという処理も行えます。

IBM WebSphere MQ クライアント・アプリケーションと異なり、MQTT クライアント・アプリは IBM WebSphere MQ アプリケーションではありません。MQTT クライアントは接続先のキュー・マネージャーを指定しません。特定の IBM WebSphere MQ プログラミング・インターフェースを使用しなければならないという制限を受けません。代わりに、MQTT クライアントに MQTT version 3 プロトコルが実装されます。MQTT プロトコルとのインターフェースを提供する独自のクライアント・ライブラリーを、選択したプログラミング言語で、選択したプラットフォーム上に作成することができます。[MQ Telemetry Transport のフォーマットおよびプロトコルを参照してください](#)。

MQTT クライアント・アプリを簡単に作成できるように、さまざまなプラットフォームを対象とする MQTT プロトコルをカプセル化した C、Java、および JavaScript クライアント・ライブラリーを使用します。MQTT クライアント・ライブラリーのクライアント API 資料へのリンクについては、[MQTT クライアント・プログラミング・リファレンス](#)を参照してください。これらのライブラリーを MQTT アプリに取り込むと、完全に機能する MQTT クライアントを、15 行ほどの短いコードで書くことができます。[PubSync.java](#) を参照してください。

`com.ibm.micro.client.mqttv3.jar` JAR ファイルの 2 つのコピーがインストールされています。一方のコピーには、ファイル名の一部としてバージョン番号が含まれています。例えば、`com.ibm.micro.client.mqttv3_3.0.2.0-20100723.jar` のようになっています。バージョン付きのコピーを OSGi アプリケーションで使用します。2 つの JAR ファイルの内容は同じです。

MQTT クライアント・アプリは、遠隔測定チャンネルとの接続の開始を常に担っています。接続されると、MQTT クライアント・アプリが IBM WebSphere MQ アプリケーションのどちらかがメッセージの交換を開始できます。

MQTT クライアント・アプリと IBM WebSphere MQ アプリケーションは、同じトピック・セットにパブリッシュとサブスクライブを行います。IBM WebSphere MQ アプリケーションは、最初にサブスクリプションを作成することなく、MQTT クライアント・アプリケーションにメッセージを直接送信することもできます。[メッセージを MQTT クライアントに送信するように分散キューイングを構成を参照してください](#)。

MQTT クライアント・アプリは、遠隔測定チャンネルを使用して IBM WebSphere MQ に接続されます。テレメトリー・チャンネルは、MQTT と IBM WebSphere MQ が使用するさまざまなタイプのメッセージどうしの

橋渡しとして機能します。MQTT クライアント・アプリのためにキュー・マネージャーにパブリケーションとサブスクリプションを作成します。遠隔測定チャネルは、キュー・マネージャーから MQTT クライアント・アプリへ、MQTT クライアント・アプリのサブスクリプションと一致するパブリケーションを送信します。

MQTT クライアントへのメッセージの送信

WebSphere MQ アプリケーションは、クライアントによって作成されたサブスクリプションにパブリッシュするか、あるいはメッセージを直接送信して、MQTT v3 クライアントにメッセージを送信できます。MQTT クライアントは、他のクライアントからサブスクライブされたトピックにパブリッシュすることで、互いにメッセージを送信できます。

MQTT クライアントがパブリケーションにサブスクライブする (このパブリケーションは WebSphere MQ から受信)

WebSphere MQ から MQTT クライアントにパブリケーションを送信するには、[82 ページの『IBM WebSphere MQ Explorer から MQTT クライアント・ユーティリティーへのメッセージのパブリッシュ』](#)の作業を実行します。

MQTT v3 クライアントがメッセージを受信する標準的な方法は、次のように、クライアントがトピックまたはトピックのセットへのサブスクリプションを作成することです。[81 ページの図 21](#) のコード・スニペットの例では、MQTT クライアントは、トピック・ストリング "MQTT Examples" を使用してサブスクライブを行います。[81 ページの図 22](#) の WebSphere MQ C アプリケーションは、トピック・ストリング "MQTT Examples" を使用してトピックへのパブリッシュを行います。[81 ページの図 23](#) のコード・スニペットでは、MQTT クライアントは、コールバック・メソッド `messageArrived` でパブリケーションを受け取ります。

MQTT クライアントからのサブスクリプションにตอบสนองしてパブリケーションを送信するように WebSphere MQ を構成する方法については、[MQTT クライアント・サブスクリプションへの応答としてのメッセージのパブリッシュ](#) を参照してください。

WebSphere MQ アプリケーションが MQTT クライアントにメッセージを直接送信する

WebSphere MQ から MQTT クライアントにメッセージを直接送信するには、[87 ページの『IBM WebSphere MQ Explorer を使用した MQTT クライアントへのメッセージの送信』](#)の作業を実行します。

この方法で MQTT クライアントに送信されるメッセージは、非送信請求メッセージと呼ばれます。MQTT v3 クライアントは、トピック名のセットを持つパブリケーションとして非送信請求メッセージを受け取ります。遠隔測定 (MQXR) サービスは、トピック名をリモートのキュー名に設定します。

デバイス用の WebSphere MQ デーモンに非送信請求メッセージを送信することはできません。デーモンは、非送信請求メッセージを受信すると、シャットダウンする場合があります。MQTT v3 クライアントは、非送信請求メッセージを別の MQTT v3 クライアントに送信することも、WebSphere MQ キューに送信することもできません。

MQTT クライアントにメッセージを直接送信するように WebSphere MQ を構成する方法については、[クライアントへのメッセージの直接送信](#) を参照してください。

MQTT クライアントがメッセージをパブリッシュする

MQTT v3 クライアントは、別の MQTT v3 クライアントが受信するメッセージをパブリッシュすることはできますが、非送信請求メッセージを送信することはできません。コード・スニペット [82 ページの図 24](#) は、Java で作成された MQTT v3 クライアントがメッセージをパブリッシュする方法を示しています。

ある特定の MQTT v3 クライアントにメッセージを送信する場合、各クライアントが独自の `ClientIdentifier` へのサブスクリプションを作成するのが標準的なパターンです。トピック・ストリングとして `ClientIdentifier` を使用して、ある MQTT クライアントから別の MQTT クライアントにメッセージをパブリッシュするには、[88 ページの『特定の MQTT v3 クライアントへのメッセージのパブリッシュ』](#)の作業を実行します。

コード・スニペットの例

81 ページの図 21 のコード・スニペットは、Java で作成された MQTT クライアントがサブスクリプションを作成する方法を示しています。これには、そのサブスクリプションに対するパブリケーションを受信するためのコールバック・メソッド、`messageArrived` も必要です。このコード・スニペットは、[Java を使用した MQ Telemetry Transport のサブスクライバーの作成のタスク](#)から抜き出したものです。

```
String    clientId = String.format("%-23.23s",
                                System.getProperty("user.name") + " " +
                                (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient client = new MqttClient("localhost", clientId);
String topicString = "MQTT Examples";
int       QoS = 1;
client.subscribe(topicString, QoS);
```

図 21. MQTT v3 クライアント・サブスクライバー

81 ページの図 22 のコード・スニペットは、C で書かれた WebSphere MQ アプリケーションがパブリケーションを送信する方法を示しています。このコード・スニペットは、[可変トピックへのパブリッシャーの作成のタスク](#)から抜き出したものです。

```
/* Define and set variables to defaults */
/* Omitted lines declaring variables */
char * topicName = ""
char * topicString = "MQTT Examples"
char * publication = "Hello world!";
do {
    MQCONN(qMgrName, &Hconn, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    td.ObjectType = MQOT_TOPIC; /* Object is a topic */
    td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
    strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
    td.ObjectString.VSPtr = topicString;
    td.ObjectString.VSLength = (MQLONG)strlen(topicString);
    MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF_QUIESCING, &Hobj, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    pmo.Options = MQPMO_FAIL_IF_QUIESCING | MQPMO_RETAIN;
    MQPUT(Hconn, Hobj, &md, &pmo, (MQLONG)strlen(publication)+1, publication, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQCLOSE(Hconn, &Hobj, MQCO_NONE, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQDISC(&Hconn, &CompCode, &Reason);
} while (0);
```

図 22. WebSphere MQ パブリッシャー

パブリケーションが着信すると、MQTT クライアントは MQTT アプリケーション・クライアントの `MqttCallback` クラスの `messageArrived` メソッドを呼び出します。このコード・スニペットは、[Java を使用した MQ Telemetry Transport のサブスクライバーの作成のタスク](#)から抜き出したものです。

```
public class CallBack implements MqttCallback {
    public void messageArrived(MqttTopic topic, MqttMessage message) {
        try {
            System.out.println("Message arrived: \"" + message.toString()
                               + "\" on topic \"" + topic.toString() + "\"");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
// ... Other callback methods
}
```

図 23. `messageArrived` メソッド

次の 82 ページの図 24 は、81 ページの図 21 で作成されたサブスクリプションにメッセージをパブリッシュする MQTT v3 を示しています。このコード・スニペットは、Java を使用した最初の MQ Telemetry Transport パブリッシャー・アプリケーションの作成のタスクから抜き出したものです。

```
String      address = "localhost";
String      clientId = String.format("%-23.23s",
    System.getProperty("user.name") + "-" +
    (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient  client = new MqttClient(address, clientId);
String      topicString = "MQTT Examples";
MqttTopic   topic = client.getTopic(Example.topicString);
String      publication = "Hello world";
MqttMessage message = new MqttMessage(publication.getBytes());
MqttDeliveryToken token = topic.publish(message);
```

図 24. MQTT v3 クライアント・パブリッシャー

IBM WebSphere MQ Explorer から MQTT クライアント・ユーティリティへのメッセージのパブリッシュ

この作業のステップに従って、IBM WebSphere MQ Explorer を使用してメッセージをパブリッシュし、MQTT クライアント・ユーティリティでそのメッセージにサブスクライブします。追加作業では、デフォルトの伝送キューを SYSTEM.MQTT.TRANSMIT.QUEUE に設定するのではなく、キュー・マネージャー別名を構成する方法を示します。

始める前に

この作業は、ユーザーが IBM WebSphere MQ および IBM WebSphere MQ Explorer に精通しており、IBM WebSphere MQ および IBM WebSphere MQ Telemetry 機能がインストールされていることを前提としています。

この作業用にキュー・マネージャー・リソースを作成するユーザーには、そのための十分な権限が必要です。デモンストレーションのために、IBM WebSphere MQ Explorer のユーザー ID は、mqm グループのメンバーとします。

このタスクについて

この作業では、IBM WebSphere MQ 内にトピックを作成し、MQTT クライアント・ユーティリティを使用してそのトピックにサブスクライブします。IBM WebSphere MQ Explorer を使用してトピックへのパブリッシュを行うと、MQTT クライアントがパブリケーションを受信します。

手順

以下の作業のいずれかを実行します。

- WebSphere MQ Telemetry はインストールされているが、まだ始動していない場合。83 ページの『作業の開始時に遠隔測定 (MQXR) サービスが定義されていない場合』の作業を実行します。
- 以前は WebSphere MQ Telemetry を実行していたが、新しいキュー・マネージャーを使用してデモンストレーションを行う場合。83 ページの『作業の開始時に遠隔測定 (MQXR) サービスが定義されていない場合』の作業を実行します。
- テレメトリー・リソースが定義されていない既存のキュー・マネージャーを使用して、この作業を実行する場合。「サンプル構成の定義」ウィザードを実行しない場合。
 - a. 以下の作業のいずれかを実行して、テレメトリーをセットアップします。
 - Telemetry 対応キュー・マネージャーの構成 (Linux および AIX)
 - Telemetry 対応キュー・マネージャーの構成 (Windows)
 - b. 84 ページの『作業の開始時に遠隔測定 (MQXR) サービスが実行されている場合』の作業を実行します。

- テレメトリー・リソースが既に定義されている既存のキュー・マネージャーを使用してこの作業を行う場合には、84 ページの『作業の開始時に遠隔測定 (MQXR) サービスが実行されている場合』の作業を実行します。

次のタスク

87 ページの『IBM WebSphere MQ Explorer を使用した MQTT クライアントへのメッセージの送信』を実行して、クライアント・ユーティリティにメッセージを直接送信します。

関連タスク

IBM WebSphere MQ Explorer を使用した MQTT クライアントへのメッセージの送信

IBM WebSphere MQ Explorer を使用して IBM WebSphere MQ キューにメッセージを書き込むことにより、MQTT クライアント・ユーティリティにメッセージを送信します。この作業では、MQTT クライアントにメッセージを直接送信するように、リモート・キュー定義を構成する方法を示します。

特定の MQTT v3 クライアントへのメッセージのパブリッシュ

トピック名として ClientIdentifier を、パブリッシュ/サブスクライブ・ブローカーとして IBM WebSphere MQ を使用して、ある MQTT v3 クライアントから別の MQTT v3 クライアントへメッセージをパブリッシュします。パブリッシュ/サブスクライブ・ブローカーとしてデバイス用の WebSphere MQ Telemetry デモンを使用して、作業を繰り返します。

作業の開始時に遠隔測定 (MQXR) サービスが定義されていない場合

キュー・マネージャーを作成し、「**サンプル構成の定義**」を実行して、キュー・マネージャー用にサンプルのテレメトリー・リソースを定義します。IBM WebSphere MQ Explorer を使用してメッセージをパブリッシュし、MQTT クライアント・ユーティリティでそのメッセージにサブスクライブします。

このタスクについて

「**サンプル構成の定義**」を使用してサンプルのテレメトリー・リソースをセットアップすると、このウィザードによってゲスト・ユーザー ID のアクセス権が設定されます。この方法でゲスト・ユーザー ID に権限を与えることについては、注意深く検討してください。Windows の場合は guest、Linux の場合は nobody に、トピック・ツリーのルートへのパブリッシュとサブスクライブ、および SYSTEM.MQTT.TRANSMIT.QUEUE へのメッセージの書き込みを行う権限が付与されます。

またこのウィザードは、デフォルトの伝送キューを SYSTEM.MQTT.TRANSMIT.QUEUE に設定します。これを行うと、既存のキュー・マネージャー上で実行されているアプリケーションに干渉する可能性があります。難しい作業になりますが、デフォルトの伝送キューを使用せずにテレメトリーを構成することも可能です。これを行う場合は、85 ページの『キュー・マネージャー別名の使用』の作業を実行してください。この作業では、既存のデフォルト伝送キューに干渉する可能性を回避するように、キュー・マネージャーを作成します。

手順

1. IBM WebSphere MQ Explorer を使用して、新しいキュー・マネージャーを作成および開始します。
 - a) Queue Managers フォルダー > 「新規」 > 「キュー・マネージャー ...」を右クリックします。キュー・マネージャーの名前を入力し、> 「完了」を選択します。

キュー・マネージャーの名前を、例えば MQTQMGR のように作成します。
2. 遠隔測定 (MQXR) サービスを作成および開始し、サンプルの遠隔測定チャンネルを作成します。
 - a) Queue Managers\QmgrName\Telemetry フォルダーを開きます。
 - b) 「**サンプル構成の定義...**」 > 「完了」をクリックします。

「**MQTT クライアント・ユーティリティを起動**」チェック・ボックスは、チェック・マークを付けたままにしておきます。
3. MQTT クライアント・ユーティリティを使用して、「MQTT Example」へのサブスクリプションを作成します。
 - a) 「**接続**」をクリックします。

「**クライアント・ヒストリー**」には、Connected イベントが記録されます。

- b) 「サブスクリプション\トピック」 フィールド> 「サブスクリाइブ」に MQTT Example と入力します。
- 「クライアント・ヒストリー」には、Subscribed イベントが記録されます。
4. WebSphere MQ に MQTTExampleTopic を作成します。
- a) WebSphere MQ Explorer> 「新規」 > 「トピック」で Queue Managers\QmgrName\Topics フォルダーを右クリックします。
- b) 「名前」に MQTTExampleTopic と入力し、> 「次へ」を選択します。
- c) 「トピック・ストリング」 > 「終了」として MQTT Example と入力します。
- d) 「OK」をクリックして、確認応答ウィンドウを閉じます。
5. IBM WebSphere MQ Explorer を使用して、Hello World! をトピック MQTT Example にパブリッシュします。
- a) Queue Managers\QmgrName\Topics 内の IBM WebSphere MQ Explorer フォルダーをクリックします。
- b) MQTTExampleTopic を右クリックし、> 「パブリケーションのテスト...」を選択します。
- c) 「メッセージ・データ」フィールドに Hello World! と入力し、「メッセージのパブリッシュ」 > 「MQTT クライアント・ユーティリティ (MQTT Client Utility)」ウィンドウに切り替えます。
- 「クライアント・ヒストリー」には、Received イベントが記録されます。

作業の開始時に遠隔測定 (MQXR) サービスが実行されている場合、テレメトリー・チャンネルおよびトピックを作成します。トピックおよびテレメトリー伝送キューを使用する権限をユーザーに与えます。IBM WebSphere MQ Explorer を使用してメッセージをパブリッシュし、MQTT クライアント・ユーティリティでそのメッセージにサブスクリाइブします。

始める前に

この作業バージョンでは、キュー・マネージャー *QmgrName* が定義および実行されています。遠隔測定 (MQXR) サービスは定義および実行されています。遠隔測定 (MQXR) サービスは、手動で作成されたか、あるいは「サンプル構成の定義」ウィザードを実行して作成された可能性があります。

このタスクについて

この作業では、既存のキュー・マネージャーを構成して、MQTT クライアント・ユーティリティにパブリケーションを送信します。

この作業のステップ 84 ページの『1』では、デフォルトの伝送キューを SYSTEM.MQTT.TRANSMIT.QUEUE に設定します。これを行うと、既存のキュー・マネージャー上で実行されているアプリケーションに干渉する可能性があります。難しい作業になりますが、デフォルトの伝送キューを使用せずにテレメトリーを構成することも可能です。これを行う場合は、85 ページの『キュー・マネージャー別名の使用』の作業を実行してください。

手順

1. SYSTEM.MQTT.TRANSMIT.QUEUE をデフォルトの伝送キューとして設定します。
 - a) Queue Managers\QmgrName folder > 「プロパティ...」を右クリック
 - b) ナビゲーターで「通信」をクリックします。
 - c) 「選択...」をクリックし、> SYSTEM.MQTT.TRANSMIT.QUEUE を選択して、> 「OK」 > 「OK」を選択します。
2. MQTT クライアント・ユーティリティを WebSphere MQ に接続するためのテレメトリー・チャンネル MQTTExampleChannel を作成し、MQTT クライアント・ユーティリティを開始します。
 - a) WebSphere MQ エクスプローラー> 「新規」 > 「テレメトリー・チャンネル ...」で Queue Managers\QmgrName\Telemetry\Channels フォルダーを右クリックします。

- b) 「チャンネル名」フィールドに MQTTExampleChannel と入力し、> 「次へ」> 「次へ」を選択します。
 - c) クライアント許可パネルの「固定ユーザー ID」を MQTTExample へのパブリッシュおよびサブスクライブを行うユーザー ID に変更し、> 「次へ」を選択します。
 - d) 「クライアント・ユーティリティーの起動」にチェック・マークを付けたままにします。「終了」
3. MQTT クライアント・ユーティリティーを使用して、「MQTT Example」へのサブスクリプションを作成します。
- a) 「接続」をクリックします。
「クライアント・ヒストリー」には、Connected イベントが記録されます。
 - b) 「サブスクリプション\トピック」フィールド> 「サブスクライブ」に MQTT Example と入力します。
「クライアント・ヒストリー」には、Subscribed イベントが記録されます。
4. WebSphere MQ に MQTTExampleTopic を作成します。
- a) WebSphere MQ Explorer> 「新規」> 「トピック」で Queue Managers\QmgrName\Topics フォルダを右クリックします。
 - b) 「名前」に MQTTExampleTopic と入力し、> 「次へ」を選択します。
 - c) 「トピック・ストリング」> 「終了」として MQTT Example と入力します。
 - d) 「OK」をクリックして、確認応答ウィンドウを閉じます。
5. mqm グループに入っていないユーザーが MQTTExample トピックへのパブリッシュおよびサブスクライブを行うようにするには、以下のようにします。
- a) トピック MQTTExampleTopic へのパブリッシュおよびサブスクライブを行うための権限をユーザーに与えます。

```
setmqaut -m qMgrName -t topic -n MQTTExampleTopic -p User ID -all +pub +sub
```

- b) メッセージを SYSTEM.MQTT.TRANSMIT.QUEUE に入れる権限をユーザーに与えます。

```
setmqaut -m qMgrName -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p User ID -all +put
```

6. IBM WebSphere MQ Explorer を使用して、Hello World! をトピック MQTT Example にパブリッシュします。
- a) Queue Managers\QmgrName\Topics 内の IBM WebSphere MQ Explorer フォルダをクリックします。
 - b) MQTTExampleTopic を右クリックし、> 「パブリケーションのテスト...」を選択します。
 - c) 「メッセージ・データ」フィールドに Hello World! と入力し、「メッセージのパブリッシュ」> 「MQTT クライアント・ユーティリティー (MQTT Client Utility)」ウィンドウに切り替えます。
「クライアント・ヒストリー」には、Received イベントが記録されます。

キュー・マネージャー別名の使用

デフォルト伝送キューを SYSTEM.MQTT.TRANSMIT.QUEUE に設定せずに、IBM WebSphere MQ Explorer を使用して MQTT クライアント・ユーティリティーにメッセージをパブリッシュします。

この作業は前の作業の続きであり、キュー・マネージャー別名を使用することで、デフォルト伝送キューを SYSTEM.MQTT.TRANSMIT.QUEUE に設定することを回避します。

始める前に

83 ページの『作業の開始時に遠隔測定 (MQXR) サービスが定義されていない場合』または 84 ページの『作業の開始時に遠隔測定 (MQXR) サービスが実行されている場合』のいずれかの作業を実行します。

このタスクについて

MQTT クライアントがサブスクリプションを作成すると、IBM WebSphere MQ は `ClientIdentifier` をリモート・キュー・マネージャー名として使用して、応答を送信します。この作業では、`ClientIdentifier` には `MyClient` が使用されます。

`MyClient` という名前の伝送キューまたはキュー・マネージャー別名がない場合、応答はデフォルト伝送キューに入れられます。デフォルト伝送キューを `SYSTEM.MQTT.TRANSMIT.QUEUE` に設定すると、MQTT クライアントは応答を受け取ります。

キュー・マネージャー別名を使用することで、デフォルト伝送キューを `SYSTEM.MQTT.TRANSMIT.QUEUE` に設定することを回避できます。キュー・マネージャー別名は、`ClientIdentifier` ごとにセットアップする必要があります。通常は、多数のクライアントが存在しており、キュー・マネージャー別名を使用するのは現実的ではありません。`ClientIdentifier` が予測不能な場合も多く、この方法でテレメトリを構成することはできません。

それでもなお、デフォルトの伝送キューを `SYSTEM.MQTT.TRANSMIT.QUEUE` 以外に構成しなければならない状況は存在します。手順のステップでは、デフォルトの伝送キューを `SYSTEM.MQTT.TRANSMIT.QUEUE` に設定する代わりに、キュー・マネージャー別名を構成します。

手順

- デフォルトの伝送キューとしての `SYSTEM.MQTT.TRANSMIT.QUEUE` を削除します。
 - Queue Managers\`QmgrName` folder > 「プロパティ...」 を右クリック
 - ナビゲーターで「通信」をクリックします。
 - 「デフォルト伝送キュー」フィールドから `SYSTEM.MQTT.TRANSMIT.QUEUE` を削除し、> 「OK」を選択します。
- MQTT クライアント・ユーティリティーを使用して、サブスクリプションを作成できなくなったことを確認します。
 - 「接続」をクリックします。
「クライアント・ヒストリー」には、Connected イベントが記録されます。
 - 「サブスクリプション\トピック」フィールド> 「サブスクリाइブ」に `MQTT Example` と入力します。
「クライアント・ヒストリー」には、Subscribe failed イベントと Connection lost イベントが記録されます。
- `ClientIdentifier` のキュー・マネージャー別名、`MyClient` を作成します。
 - Queue Managers\`QmgrName`\Queues フォルダー> 「新規」 > 「リモート・キュー定義」を右クリックします。
 - 定義に `MyClient` という名前を付け、> 「次へ」を選択します。
 - 「リモート・キュー・マネージャー」フィールドに `MyClient` と入力します。
 - 「伝送キュー」フィールドに `SYSTEM.MQTT.TRANSMIT.QUEUE` と入力し、> 「完了」を選択します。
- MQTT クライアント・ユーティリティーに再度接続します。
 - 「クライアント ID」が `MyClient` に設定されていることを確認します。
 - 接続
「クライアント・ヒストリー」には、Connected イベントが記録されます。
- MQTT クライアント・ユーティリティーを使用して、「MQTT Example」へのサブスクリプションを作成します。
 - 「接続」をクリックします。
「クライアント・ヒストリー」には、Connected イベントが記録されます。

- b) 「サブスクリプション\トピック」 フィールド> 「サブスクリाइブ」に MQTT Example と入力します。
- 「クライアント・ヒストリー」には、Subscribed イベントが記録されます。
6. IBM WebSphere MQ Explorer を使用して、Hello World! をトピック MQTT Example にパブリッシュします。
- a) Queue Managers\QmgrName\Topics 内の IBM WebSphere MQ Explorer フォルダをクリックします。
- b) MQTTEExampleTopic を右クリックし、> 「パブリケーションのテスト...」を選択します。
- c) 「メッセージ・データ」 フィールドに Hello World! と入力し、「メッセージのパブリッシュ」> 「MQTT クライアント・ユーティリティ (MQTT Client Utility)」 ウィンドウに切り替えます。
- 「クライアント・ヒストリー」には、Received イベントが記録されます。

IBM WebSphere MQ Explorer を使用した MQTT クライアントへのメッセージの送信

IBM WebSphere MQ Explorer を使用して IBM WebSphere MQ キューにメッセージを書き込むことにより、MQTT クライアント・ユーティリティにメッセージを送信します。この作業では、MQTT クライアントにメッセージを直接送信するように、リモート・キュー定義を構成する方法を示します。

始める前に

82 ページの『[IBM WebSphere MQ Explorer から MQTT クライアント・ユーティリティへのメッセージのパブリッシュ](#)』のタスクを実行します。MQTT クライアント・ユーティリティは、接続したままにします。

このタスクについて

この作業では、トピックへのパブリッシュを使用するのではなく、キューを使用して MQTT クライアントへメッセージを送信する方法について説明します。クライアントでのサブスクリプションの作成は行いません。この作業のステップ [87 ページの『2』](#)では、以前のサブスクリプションが削除されていることを確認します。

手順

1. MQTT クライアント・ユーティリティへの接続を切断してから再接続することで、既存のサブスクリプションをすべて廃棄します。

デフォルトが変更されていなければ、MQTT クライアント・ユーティリティは新規セッションで接続するため、サブスクリプションは廃棄されます。[クリーン・セッション](#)を参照してください。

作業を容易にするために、MQTT クライアント・ユーティリティによって作成された生成 ClientIdentifier を使用するのではなく、独自の ClientIdentifier を入力してください。

a) 「切断」をクリックして、テレメトリー・チャンネルから MQTT クライアント・ユーティリティを切断します。

「クライアント履歴」には、Disconnected イベントが記録されます。

b) 「クライアント ID」を MyClient に変更します。

c) 「接続」をクリックします。

「クライアント履歴」には、Connected イベントが記録されます。
2. MQTT クライアント・ユーティリティが MQTTEExampleTopic へのパブリケーションを受信しなくなったことを確認します。

a) Queue Managers\QmgrName\Topics 内の IBM WebSphere MQ Explorer フォルダをクリックします。

b) MQTTEExampleTopic を右クリックし、> 「パブリケーションのテスト...」を選択します。

- c) 「メッセージ・データ」 フィールドに Hello World! と入力し、「メッセージのパブリッシュ」 > 「MQTT クライアント・ユーティリティ (MQTT Client Utility)」 ウィンドウに切り替えます。
「クライアント・ヒストリー」にはイベントは何も記録されません。
3. クライアントのリモート・キュー定義を作成します。
- リモート・キュー定義で、リモート・キュー・マネージャー名として、ClientIdentifier である MyClient を設定します。リモート・キュー名には任意の名前を使用できます。リモート・キュー名は、トピック名として MQTT クライアントに渡されます。
- a) Queue Managers\QmgrName\Queues フォルダー > 「新規」 > 「リモート・キュー定義」を右クリックします。
- b) 定義に MyClientRemoteQueue という名前を付け、> 「次へ」を選択します。
- c) 「リモート・キュー」 フィールドに MQTTExampleQueue と入力します。
- d) 「リモート・キュー・マネージャー」 フィールドに MyClient と入力します。
- e) 「伝送キュー」 フィールドに SYSTEM.MQTT.TRANSMIT.QUEUE と入力し、> 「完了」を選択します。
4. テスト・メッセージを MyClientRemoteQueue に入れます。
- a) **MyClientRemoteQueue** を右クリックし、> 「テスト・メッセージの書き込み...」を選択します。
- b) 「メッセージ・データ」 フィールドに Hello queue! と入力し、「メッセージの書き込み」 > 「閉じる」を選択します。
- 「クライアント・ヒストリー」には、Received イベントが記録されます。
5. デフォルトの伝送キューとしての SYSTEM.MQTT.TRANSMIT.QUEUE を削除します。
- a) Queue Managers\QmgrName folder > 「プロパティ...」を右クリック
- b) ナビゲーターで「通信」をクリックします。
- c) 「デフォルト伝送キュー」フィールドから SYSTEM.MQTT.TRANSMIT.QUEUE を削除し、> 「OK」を選択します。
6. ステップ 88 ページの『4』をやり直します。
- MyClientRemoteQueue は、明示的に伝送キューの名前を付けるリモート・キュー定義です。MyClient にメッセージを送信するために、デフォルトの伝送キューを定義する必要はありません。

次のタスク

もはやデフォルト伝送キューは SYSTEM.MQTT.TRANSMIT.QUEUE に設定されていないため、MQTT クライアント・ユーティリティは、ClientIdentifier である MyClient にキュー・マネージャー別名が定義されない限り、新しいサブスクリプションを作成できません。デフォルト伝送キューを SYSTEM.MQTT.TRANSMIT.QUEUE に復元します。

関連タスク

IBM WebSphere MQ Explorer から MQTT クライアント・ユーティリティへのメッセージのパブリッシュ
この作業のステップに従って、IBM WebSphere MQ Explorer を使用してメッセージをパブリッシュし、MQTT クライアント・ユーティリティでそのメッセージにサブスクライブします。追加作業では、デフォルトの伝送キューを SYSTEM.MQTT.TRANSMIT.QUEUE に設定するのではなく、キュー・マネージャー別名を構成する方法を示します。

特定の MQTT v3 クライアントへのメッセージのパブリッシュ

トピック名として ClientIdentifier を、パブリッシュ/サブスクライブ・ブローカーとして IBM WebSphere MQ を使用して、ある MQTT v3 クライアントから別の MQTT v3 クライアントへメッセージをパブリッシュします。パブリッシュ/サブスクライブ・ブローカーとしてデバイス用の WebSphere MQ Telemetry デモンを使用して、作業を繰り返します。

特定の MQTT v3 クライアントへのメッセージのパブリッシュ

トピック名として ClientIdentifier を、パブリッシュ/サブスクライブ・ブローカーとして IBM WebSphere MQ を使用して、ある MQTT v3 クライアントから別の MQTT v3 クライアントへメッセージを

パブリッシュします。パブリッシュ/サブスクライブ・ブローカーとしてデバイス用の WebSphere MQ Telemetry デーモンを使用して、作業を繰り返します。

始める前に

82 ページの『IBM WebSphere MQ Explorer から MQTT クライアント・ユーティリティーへのメッセージのパブリッシュ』のタスクを実行します。MQTT クライアント・ユーティリティーは、接続したままにします。

このタスクについて

この作業では、以下の 2 つについて説明します。

1. 1 つの MQTT クライアントでトピックへのサブスクライブを行い、別の MQTT クライアントからのパブリケーションを受け取ります。
2. トピック・ストリングとして ClientIdentifier を使用して、「Point-to-Point」サブスクリプションをセットアップします。

追加作業の 90 ページの『パブリッシュ/サブスクライブ・ブローカーとしてデバイス用の WebSphere MQ Telemetry デーモンを使用する』では、パブリッシュ/サブスクライブ・ブローカーとして、WebSphere MQ ではなくデバイス用の WebSphere MQ Telemetry デーモンを使用します。

手順

1. MQTT クライアント・ユーティリティーへの接続を切断してから再接続することで、既存のサブスクリプションをすべて廃棄します。

デフォルトが変更されていなければ、MQTT クライアント・ユーティリティーは新規セッションで接続するため、サブスクリプションは廃棄されます。[クリーン・セッション](#)を参照してください。

作業を容易にするために、MQTT クライアント・ユーティリティーによって作成された生成 ClientIdentifier を使用するのではなく、独自の ClientIdentifier を入力してください。

- a) 「**切断**」をクリックして、テレメトリー・チャンネルから MQTT クライアント・ユーティリティーを切断します。

「**クライアント履歴**」には、Disconnected イベントが記録されます。

- b) 「**クライアント ID**」を MyClient に変更します。
- c) 「**接続**」をクリックします。

「**クライアント履歴**」には、Connected イベントが記録されます。

2. トピック MyClient へのサブスクリプションを作成します。

MyClient は、このクライアントの ClientIdentifier です。

- a) 「**サブスクリプション\トピック**」フィールド> 「**サブスクライブ**」に MyClient と入力します。

「**クライアント・ヒストリー**」には、Subscribed イベントが記録されます。

3. 別の MQTT クライアント・ユーティリティーを開始します。

- a) Queue Managers*QmgrName*\Telemetry\channels フォルダーを開きます。
- b) **PlainText** チャンネル> 「**MQTT クライアント・ユーティリティーの実行 ...**」を右クリックします。
- c) 「**接続**」をクリックします。

「**クライアント履歴**」には、Connected イベントが記録されます。

4. トピック MyClient に Hello MyClient! をパブリッシュします。

- a) ClientIdentifier、MyClient を指定して実行する MQTT クライアント・ユーティリティーから、サブスクリプション・トピック MyClient をコピーします。
- b) MyClient を、各 MQTT クライアント・ユーティリティー・インスタンスの 「**パブリケーション\トピック**」 フィールドに貼り付けます。

- c) 「パブリケーション・メッセージ」 フィールドに Hello MyClient! と入力します。
- d) 両方のインスタンスで「パブリッシュ」をクリックします。

タスクの結果

MQTT クライアント・ユーティリティの「クライアント・ヒストリー」に ClientIdentifier, MyClient を指定すると、2つの「受信済み」イベントと1つの「パブリッシュ済み」イベントが記録されます。もう一方の MQTT クライアント・ユーティリティ・インスタンスには、1つの「パブリッシュ済み」イベントが記録されます。

「受信済み」 イベントが1つしか表示されない場合は、以下の考えられる原因を確認してください。

1. キュー・マネージャーのデフォルト伝送キューが SYSTEM.MQTT.TRANSMIT.QUEUE に設定されている。
2. 他の演習を実行する際に、MyClient を参照するキュー・マネージャー別名またはリモート・キュー定義を作成しましたか？構成上の問題が発生した場合は、MyClient を参照するリソース(キュー・マネージャー別名や伝送キューなど)を削除します。クライアント・ユーティリティへの接続を切断し、遠隔測定 (MQXR) サービスを停止して再開します。

次のタスク

次に、90 ページの『パブリッシュ/サブスクライブ・ブローカーとしてデバイス用の WebSphere MQ Telemetry デモンを使用する』の作業を実行します。MQTT クライアント・ユーティリティは、テレメトリー・チャンネルではなく、デバイス用の WebSphere MQ Telemetry デモンに接続します。

関連タスク

IBM WebSphere MQ Explorer から MQTT クライアント・ユーティリティへのメッセージのパブリッシュ
この作業のステップに従って、IBM WebSphere MQ Explorer を使用してメッセージをパブリッシュし、MQTT クライアント・ユーティリティでそのメッセージにサブスクライブします。追加作業では、デフォルトの伝送キューを SYSTEM.MQTT.TRANSMIT.QUEUE に設定するのではなく、キュー・マネージャー別名を構成する方法を示します。

IBM WebSphere MQ Explorer を使用した MQTT クライアントへのメッセージの送信

IBM WebSphere MQ Explorer を使用して IBM WebSphere MQ キューにメッセージを書き込むことにより、MQTT クライアント・ユーティリティにメッセージを送信します。この作業では、MQTT クライアントにメッセージを直接送信するように、リモート・キュー定義を構成する方法を示します。

パブリッシュ/サブスクライブ・ブローカーとしてデバイス用の WebSphere MQ Telemetry デモンを使用する

WebSphere MQ の代わりに、デバイス用の WebSphere MQ Telemetry デモンを、パブリッシュ/サブスクライブ・ブローカーとして使用します。MQTT クライアント・ユーティリティの1つのインスタンスでその ClientIdentifier をトピック・ストリングに使用してサブスクライブし、別のインスタンスでメッセージをパブリッシュして最初のインスタンスに送信します。

始める前に

デモンをインストールします(まだインストールしていない場合)。

検証は実行しないでください。検証にはポート 1883 が使用されますが、このポートは既に PlainText テレメトリー・チャンネルによって使用されています。

このタスクについて

この作業では、デフォルト以外の TCP/IP ポートを使用して、MQTT クライアント・ユーティリティを、デバイス用の WebSphere MQ Telemetry デモンに接続します。片方のクライアントがその ClientIdentifier をトピック・ストリングとして使用してサブスクライブし、もう一方のクライアントは、前の作業とまったく同様に、その ClientIdentifier にパブリッシュします。手順を参照してください。

注: この作業に関する文書は、Windows でのデーモンの実行用に作成されています。Linux でデーモンを実行するには、amqtdd のパスと許可を変更します。

手順

1. デバイス用の WebSphere MQ Telemetry デーモンが含まれているディレクトリーで、コマンド・ウィンドウを開きます。

Windows のディレクトリー・パスは、*WebSphere MQ installation directory\mqxr\SDK\advanced\DeviceDaemon\windows_ia32* です。

2. 別の TCP/IP ポートでデーモンを実行します。
 - a) デーモンと同じディレクトリーに、amqtdd.cfg というファイルを作成します。
 - b) このファイルに、そのデーモン用に別のデフォルト・ポートを構成するための行を追加します。

```
port 1884
```

- c) ファイルを保存します。

3. デーモンを開始します。

```
amqtdd
```

デーモンは、そのコンソール・ログをコマンド・ウィンドウに書き込みます。

```
20100712 123133.857 CWNAN9999I IBM WebSphere MQ Telemetry daemon for devices
20100712 123133.857 CWNAN9997I Licensed Materials - Property of IBM
20100712 123133.857 CWNAN9996I Copyright IBM Corp. 2007, 2024. All Rights Reserved
20100712 123133.857 CWNAN9995I US Government Users Restricted Rights ...
20100712 123133.857 CWNAN0049I Configuration file name is .\amqtdd.cfg
20100712 123133.873 CWNAN0054I Features included: bridge
20100712 123134.060 CWNAN0014I MQTT protocol starting, listening on port 1884
```

4. MQTT クライアント・ユーティリティーのインスタンスを開始します。

MQTT クライアント・ユーティリティーは、テレメトリー・チャンネルからのみ開始します。これにより、このデーモンに接続できるようになります。また、IBM WebSphere MQ SupportPac、IA92 をインストールすることもできます。SupportPac は、[IA92: WBI Brokers-Java implementation of WebSphere MQ Telemetry transport](#) から入手できます。

- a) Queue Managers*QmgrName*\Telemetry\channels フォルダーを開きます。
- b) **PlainText** チャンネル> 「**MQTT クライアント・ユーティリティーの実行 ...**」 を右クリックします。
- c) 「ポート」を 1884 に変更します。
- d) 「クライアント ID」を MyClient に変更します。
- e) 「接続」 をクリックします。

「クライアント履歴」には、Connected イベントが記録されます。

5. トピック MyClient へのサブスクリプションを作成します。

MyClient は、このクライアントの ClientIdentifier です。

- a) 「サブスクリプション\トピック」 フィールド> 「サブスクライブ」に MyClient と入力します。
- 「クライアント・ヒストリー」には、Subscribed イベントが記録されます。

6. 別の MQTT クライアント・ユーティリティーを開始します。

- a) Queue Managers*QmgrName*\Telemetry\channels フォルダーを開きます。
- b) **PlainText** チャンネル> 「**MQTT クライアント・ユーティリティーの実行 ...**」 を右クリックします。
- c) 「ポート」を 1884 に変更します。
- d) 「接続」 をクリックします。

「クライアント履歴」には、Connected イベントが記録されます。

7. トピック MyClient に Hello MyClient! をパブリッシュします。

- a) ClientIdentifier、MyClient を指定して実行する MQTT クライアント・ユーティリティーから、サブスクリプション・トピック MyClient をコピーします。
- b) MyClient を、各 MQTT クライアント・ユーティリティー・インスタンスの「パブリケーション\トピック」フィールドに貼り付けます。
- c) 「パブリケーション・メッセージ」フィールドに Hello MyClient! と入力します。
- d) 両方のインスタンスで「パブリッシュ」をクリックします。

タスクの結果

MQTT クライアント・ユーティリティーの「クライアント・ヒストリー」に ClientIdentifier, MyClient を指定すると、2つの「受信済み」イベントと1つの「パブリッシュ済み」イベントが記録されます。もう一方の MQTT クライアント・ユーティリティー・インスタンスには、1つの「パブリッシュ済み」イベントが記録されます。

デバイス用の WebSphere MQ Telemetry デーモンとの接続イベントおよび切断イベントを、コマンド・ウィンドウでモニターすることも可能です。

MQTT クライアントから IBM WebSphere MQ アプリケーションへのメッセージの送信

IBM WebSphere MQ アプリケーションは、トピックにサブスクライブすることにより、MQTT v3 クライアントからメッセージを受信できます。MQTT クライアントは、遠隔測定チャンネルを使用して IBM WebSphere MQ に接続し、同じトピックにパブリッシュすることによって IBM WebSphere MQ アプリケーションにメッセージを送信します。

92 ページの『[MQTT クライアントから IBM WebSphere MQ Explorer へのメッセージのパブリッシュ](#)』のタスクを実行して、MQTT クライアントから IBM WebSphere MQ で定義されているサブスクリプションにパブリケーションを送信する方法を学習してください。

トピックがクラスター化されている場合、またはパブリッシュ/サブスクライブ階層を使用して分散されている場合は、サブスクリプションを、MQTT クライアントが接続されているキュー・マネージャーとは異なるキュー・マネージャー上に置くことができます。

MQTT クライアントから IBM WebSphere MQ Explorer へのメッセージのパブリッシュ

WebSphere MQ エクスプローラーを使用して、トピックへのサブスクリプションを作成し、WebSphere MQTT クライアント・ユーティリティーを使用して、トピックにパブリッシュします。

始める前に

82 ページの『[IBM WebSphere MQ Explorer から MQTT クライアント・ユーティリティーへのメッセージのパブリッシュ](#)』のタスクを実行します。MQTT クライアント・ユーティリティーは、接続したままにします。

このタスクについて

このタスクでは、MQTT クライアントによるメッセージのパブリッシュ、および WebSphere MQ エクスプローラーで作成した管理対象外の永続サブスクリプションを使用したパブリケーションの受信について説明します。

手順

1. トピック・ストリング MQTT Example に対する永続サブスクリプションを作成します。次のいずれかの手順を実行します。
 - 結果に記載されているコマンド・スクリプトを実行します。
 - WebSphere MQ エクスプローラーを使用して、以下のステップを実行し、キューおよびサブスクリプションを作成します。

- a) WebSphere MQ エクスプローラー> 「新規」 > 「ローカル・キュー ...」 で Queue Managers*QmgrName*\Queues フォルダを右クリックします。
 - b) キュー名として MQTTExampleQueue と入力し、「終了」をクリックします。
 - c) WebSphere MQ Explorer> 「新規」 > 「サブスクリプション ...」 で Queue Managers*QmgrName*\Subscriptions フォルダを右クリックします。
 - d) キュー名として MQTTExampleSubscription と入力し、「次へ」をクリックします。
 - e) 「選択...」 > 「MQTTExampleTopic」 > 「OK」をクリックします。
トピック MQTTExampleTopic は、82 ページの『IBM WebSphere MQ Explorer から MQTT クライアント・ユーティリティへのメッセージのパブリッシュ』のステップ 84 ページの『4』で既に作成されています。
 - f) 宛先名として MQTTExampleQueue と入力し、「終了」をクリックします。
2. 任意指定のステップとして、mqm 権限を持たずに、異なるユーザーが使用できるようにキューをセットアップします。
- mqm 以外の少ない権限を有するユーザーのために設定をセットアップしている場合、put と get 権限を MQTTExampleQueue に付与すべきです。トピックと伝送キューへのアクセス権限は 82 ページの『IBM WebSphere MQ Explorer から MQTT クライアント・ユーティリティへのメッセージのパブリッシュ』で構成されています。
- a) 以下のようにして、ユーザーにキュー MQTTExampleQueue の書き込みと取得を認可します。
- ```
setmqaut -m qMgrName -t queue -n MQTTExampleQueue -p User ID -all +put +get
```
3. MQTT クライアント・ユーティリティを使用して、Hello WebSphere MQ! をトピック MQTT Example にパブリッシュします。
- MQTT クライアント・ユーティリティを接続したままにしている場合は、**PlainText** チャネル> 「MQTT クライアント・ユーティリティの実行 ...」 > 「接続」を右クリックします。
- a) 「パブリケーション&#xa5;トピック」フィールドに MQTT Example と入力します。
  - b) 「パブリケーション\メッセージ」フィールド> 「パブリッシュ」に Hello WebSphere MQ! と入力します。
4. Queue Managers\*QmgrName*\Queues フォルダを開き、MQTTExampleQueue を見つけます。
- 「現行キュー項目数」フィールド値は 1 です。
5. MQTTExampleQueue > 「メッセージの参照 ...」 を右クリックします。資料を調べてください。

## IBM WebSphere MQ Telemetry デーモン (デバイス用) と IBM WebSphere MQ の間でメッセージを転送します。

このタスクを実行して、コマンドを IBM WebSphere MQ Telemetry デーモン (デバイス用) に送信する方法を学習します。コマンドを書くことによって、IBM WebSphere MQ からデーモンへ、また、デーモンから IBM WebSphere MQ へメッセージを転送するブリッジが作成されます。

### 始める前に

タスクの 88 ページの『特定の MQTT v3 クライアントへのメッセージのパブリッシュ』と 90 ページの『パブリッシュ/サブスクライブ・ブローカーとしてデバイス用の WebSphere MQ Telemetry デーモンを使用する』を実行して、MQTT クライアント・ユーティリティの使用についてよく理解しておきます。これらのタスクが完了しても、MQTT クライアント・ユーティリティのインスタンスの 1 つをデバイス用の Telemetry デーモンに接続したままにしておいてください。別のインスタンスは遠隔測定チャンネルに接続したままにしておきます。

このタスクは、アドレス 127.0.0.1 のポート 1883 を listen するテレメトリー・サービスに対してチャンネルが定義されていることを前提としています。同様に、デフォルトのデーモン・リスナーは、アドレス 127.0.0.1 のポート 1884 を listen するように構成されます。デーモン amqtd と同じディレクトリー

に保管されているファイル `amqtdc.cfg` 内の単一行で、デフォルトのデーモン・リスナー・ポートを構成します。

```
port 1884
```

## このタスクについて

このタスクでは、実行中のデーモンを更新して、WebSphere MQ 遠隔測定 (MQXR) サービスへの接続ブリッジを作成し、デーモンとメッセージを交換します。

**ヒント:** 更新ファイル `amqtdc.upd` は、使用後にデーモンによって削除されます。作成したコマンドを後で使用するために保持する場合は、コマンドを別のファイルに作成してから `amqtdc.upd` に転送する方法があります。

## 手順

1. MQTT クライアント・ユーティリティーの 2 つのインスタンスが実行されていることを確認します。1 つはポート 1884 のデーモンに接続されており、もう 1 つはポート 1883 で実行されている遠隔測定チャンネルに接続されています。
2. デーモンの `amqtdc` と同じディレクトリーに `amqtdc.upd` ファイルを作成して、ファイルに以下のコマンドを含めます。

```
connection daemon1
address 127.0.0.1:1883
topic # in import/ export/
topic # out export/ import/
try_private false
```

- ブリッジは `daemon1` と呼ばれ、ソケット・アドレス `127.0.0.1:1883` で実行中の遠隔測定 (MQXR) サービス用に構成されたチャンネルに接続されています。 `try_private` コマンドはオプションで、デフォルトは `true` です。この行がない場合、ブリッジは最初に、WebSphere MQ Telemetry デーモン (デバイス用) が認識する専用プロトコルを使用して、接続を試行します。コマンドに `try_private false` を含めるとこのステップが実行されないため、正常な接続を完了させる時間が短縮されます。
- 行 `topic # in import/ export/` は、キュー・マネージャーで作成されたトピック・ストリング `export/#` に一致するすべてのトピックにサブスクライブするように `daemon1` に指示します。一致するパブリケーションが、トピック・ストリングの先頭を `export/` から `import/` に変更して、キュー・マネージャーからデーモンに転送されます。行 `topic # out export/ import/` は、ローカル・デーモンでサブスクリプションを作成します。ブリッジは、デーモンで作成されたトピック・ストリング `export/#` に一致するすべてのトピックにサブスクライブします。パブリケーションは、トピック・ストリングの先頭を `export/` から `import/` に変更して、デーモンからキュー・マネージャーに転送されます。

94 ページの図 25 は結果として生成されるコンソール・ログを示しています。

```
CWNAN0124I Starting bridge connection daemon1
CWNAN0133I Bridge connection daemon1 to 127.0.0.1:1883 now established
```

図 25. 接続ブリッジ開始からのコンソール・ログ

3. MQTT クライアント・ユーティリティーの各インスタンスで、「サブスクリプション/トピック:」入力フィールド> 「サブスクライブ」に `import/#` と入力します。
4. MQTT クライアント・ユーティリティーの各インスタンスで、「パブリケーション/トピック:」入力フィールドに `export/#` と入力します。
  - a) ポート 1883 に接続されている MQTT クライアント・ユーティリティーで、テレメトリー・チャンネルの「パブリケーション/メッセージ:」入力フィールド> 「パブリッシュ」に `From the queue manager` と入力します。

b) ポート 1884 に接続された MQTT クライアント・ユーティリティー (テレメトリー・デーモン) で、「パブリケーション/メッセージ:」 入力フィールド> 「パブリッシュ」に From the daemon と入力します。

各 MQTT クライアント・ユーティリティーのクライアント・履歴に、ブローカー間で転送されたパブリケーションが表示されます。

## MQTT パブリッシュ/サブスクライブ・アプリケーション

トピック・ベースのパブリッシュ/サブスクライブを使用して MQTT アプリケーションを作成します。

MQTT クライアントが接続されると、クライアントとサーバーの間でどちらかの方向にパブリケーションがフローされます。クライアントで情報がパブリッシュされる場合、パブリケーションはクライアントから送信されます。クライアントによって作成されたサブスクリプションと一致するトピックにメッセージがパブリッシュされた場合、パブリケーションはクライアントで受信されます。

WebSphere MQ パブリッシュ/サブスクライブ・ブローカーは、MQTT クライアントによって作成されたトピックとサブスクリプションを管理します。MQTT クライアントによって作成されたトピックは、WebSphere MQ アプリケーションによって作成されたトピックと同じトピック・スペースを共有します。

MQTT クライアント・サブスクリプションのトピック・ストリングと一致するパブリケーションは、リモート・キュー・マネージャー名がクライアントの `ClientIdentifier` に設定された `SYSTEM.MQTT.TRANSMIT.QUEUE` に配置されます。遠隔測定 (MQXR) サービスは、サブスクリプションを作成したクライアントにパブリケーションを転送します。その場合、クライアントを識別するリモート・キュー・マネージャー名として設定されている `ClientIdentifier` が使用されます。

通常は、`SYSTEM.MQTT.TRANSMIT.QUEUE` をデフォルトの伝送キューとして定義する必要があります。デフォルト伝送キューを使用しないように MQTT を構成することは可能ですが、煩雑です。[MQTT クライアントにメッセージを送信するように分散キューイングを構成する](#) を参照してください。

MQTT クライアントは持続セッションを作成できます (100 ページの『MQTT のステートレス・セッションとステートフル・セッション』を参照)。持続セッションで作成されたサブスクリプションは永続的です。持続セッションを持つクライアント宛に到着したパブリケーションは、`SYSTEM.MQTT.TRANSMIT.QUEUE` に保管され、クライアントが再接続した時点でクライアントに転送されます。

MQTT クライアントは、保存パブリケーションにパブリッシュとサブスクライブを行うこともできます (保存パブリケーションおよび MQTT クライアントを参照)。保存パブリケーション・トピックのサブスクライバーは、トピックへの最新のパブリケーションを受信します。サブスクライバーは、サブスクリプションの作成時、または以前のセッションへの再接続時に保存パブリケーションを受信します。

### 関連タスク

[Java を使用した最初の MQ Telemetry Transport パブリッシャー・アプリケーションの作成](#)

[Java を使用した MQ Telemetry Transport のサブスクライバーの作成](#)

## 遠隔測定アプリケーション

WebSphere MQ または WebSphere Message Broker メッセージ・フローを使用して、遠隔測定アプリケーションを作成します。

JMS や MQI などの WebSphere MQ プログラミング・インターフェースを使用して、WebSphere MQ の遠隔測定アプリケーションをプログラムします。

遠隔測定 (MQXR) サービスは、MQTT v3 メッセージと WebSphere MQ メッセージの間の変換を行います。MQTT クライアントのためにサブスクリプションとパブリケーションを作成し、パブリケーションを MQTT クライアントに転送します。パブリケーションは MQTT v3 メッセージのペイロードです。ペイロードは、メッセージ・ヘッダーと `json-bytes` フォーマットのバイト配列から成ります。遠隔測定サーバーは、MQTT v3 メッセージのヘッダーと WebSphere MQ メッセージのヘッダーをマップします (96 ページの『WebSphere MQ Telemetry とキュー・マネージャーの統合』を参照)。

`Publication`、`MQInput`、および `JMSInput` ノードを使用して、WebSphere Message Broker と MQTT クライアントの間でパブリケーションを送受信します。

メッセージ・フローを使用して、遠隔測定を Web サイトと統合 (HTTP を使用) できるほか、他のアプリケーションとも統合 (WebSphere MQ および WebSphere Adapter を使用) できます。

WebSphere MQ Telemetry は、WebSphere Message Broker バージョン 7 の SCADA ノードを置き換えます。SCADAInput ノードおよび SCADAOutput ノードを使用するバージョン 6 WebSphere Message Broker メッセージ・フローをバージョン 7 にマイグレーションする方法については、[WebSphere Message Broker バージョン 6 を使用する場合の遠隔測定アプリケーションのマイグレーション IBM WebSphere MQ Telemetry および WebSphere Message Broker バージョン 7.0](#) を参照してください。

## WebSphere MQ Telemetry とキュー・マネージャーの統合

MQTT クライアントは、パブリッシュ/サブスクライブ・アプリケーションとして WebSphere MQ と統合されます。新しいトピックを作成するか、既存のトピックを使用して、WebSphere MQ 内のトピックに対して、パブリッシュとサブスクライブのどちらも行えます。自身のサブスクリプションのトピックにパブリッシュする MQTT クライアント (自身を含む) や他の WebSphere MQ アプリケーションの結果として、WebSphere MQ からパブリケーションを受信します。パブリケーションの属性を決定するために規則が適用されます。

トピック、パブリケーション、サブスクリプション、メッセージに関連付けられる属性のうち、WebSphere MQ で定義された属性の多くはサポートされません。96 ページの『MQTT クライアントから WebSphere MQ パブリッシュ/サブスクライブ・ブローカーへ』と 98 ページの『WebSphere MQ から MQTT クライアントへ』は、パブリケーションの属性がどのように設定されるかを示しています。設定は、パブリケーションが WebSphere MQ パブリッシュ/サブスクライブ・ブローカーに向かうものであるか、そこからのものであるかによって異なります。

WebSphere MQ 内のパブリッシュ/サブスクライブ・トピックは、管理可能トピック・オブジェクトに関連付けられます。MQTT クライアントによって作成されるトピック間に違いはありません。MQTT クライアントがパブリケーションのトピック・ストリングを作成すると、WebSphere MQ パブリッシュ/サブスクライブ・ブローカーはそれを管理可能トピック・オブジェクトに関連付けます。ブローカーはパブリケーションに含まれるトピック・ストリングを、最も近い管理可能トピック・オブジェクトの親にマップします。このマッピングは、WebSphere MQ アプリケーションの場合と同じです。ユーザーが作成したトピックがない場合、パブリケーション・トピックは SYSTEM.BASE.TOPIC にマップされます。パブリケーションに適用される属性は、トピック・オブジェクトに由来します。

WebSphere MQ アプリケーションまたは管理者がサブスクリプションを作成する際に、そのサブスクリプションに名前が付けられます。WebSphere MQ エクスプローラーを使用するか、`runmqsc` または PCF コマンドを使用して、サブスクリプションをリストします。すべての MQTT クライアントのサブスクリプションの名前が示されます。`ClientIdentifier:Topic name` という形式の名前が付けられます。

## MQTT クライアントから WebSphere MQ パブリッシュ/サブスクライブ・ブローカーへ

MQTT クライアントが WebSphere MQ にパブリケーションを送信しました。遠隔測定 (MQXR) サービスが、パブリケーションを WebSphere MQ メッセージに変換します。WebSphere MQ メッセージは、以下の 3 つの部分で構成されます。

1. MQMD
2. RFH2
3. メッセージ

MQMD の各プロパティは、96 ページの表 3 に示すものを除いて、それぞれのデフォルト値に設定されます。

| MQMD フィールド | タイプ     | 値                 |
|------------|---------|-------------------|
| Foformat   | MQCHAR8 | MQFMT_RF_HEADER_2 |

表 3. MQMD (続き)

| MQMD フィールド            | タイプ      | 値                                                                                                                            |
|-----------------------|----------|------------------------------------------------------------------------------------------------------------------------------|
| <b>UserIdentifier</b> | MQCHAR12 | 以下のいずれかに設定されます。<br>MqttClient.ClientIdentifier<br>MqttConnectOptions.UserName<br>WebSphere MQ 管理者によって設定された遠隔測定チャンネルのユーザー ID。 |
| <b>Priority</b>       | MQLONG   | MQPRI_PRIORITY_AS_Q_DEF (WebSphere MQ のデフォルト。デフォルトが 4 の JMS とは異なります。)                                                        |
| <b>Persistence</b>    | MQLONG   | QoS=0→MQPER_NOT_PERSISTENT<br>QoS=1→MQPER_PERSISTENT<br>QoS=2→MQPER_PERSISTENT                                               |

RFH2 ヘッダーには、JMS メッセージのタイプを定義するための <msd> フォルダーは含まれていません。遠隔測定 (MQXR) サービスによって、WebSphere MQ メッセージがデフォルトの JMS メッセージとして作成されます。jms-bytes メッセージがデフォルトの JMS メッセージ・タイプです。アプリケーションからは、メッセージ・プロパティとして追加されたヘッダー情報にアクセス可能です ([メッセージ・プロパティ](#)を参照)。

RFH2 の値は、[97 ページの表 4](#) に示すように設定されます。Format プロパティは RFH2 固定ヘッダーに設定され、他の値は RFH2 フォルダーに設定されます。

表 4. RFH2

| RFH2 プロパティ       | タイプ/フォルダー     | ヘッダー                                                                           |
|------------------|---------------|--------------------------------------------------------------------------------|
| 形式 (Format)      | MQCHAR8       | MQFMT_NONE                                                                     |
| ClientIdentifier | mqtt/clientId | 長さが 1...23 バイトの MqttClient.ClientIdentifier をコピーします。                           |
| QoS              | mqtt/qos      | 着信 MQTT メッセージから QoS をコピーします。                                                   |
| メッセージ ID         | mqtt/msgid    | QoS が 1 または 2 の場合に、着信 MQTT メッセージからメッセージ ID をコピーします。                            |
| MQIsRetained     | mqs/Ret       | 元の MQTT パブリケーションが RETAIN プロパティ・セットが設定されて送信された場合に設定し、メッセージは保存パブリケーションとして受信されます。 |
| MQTopicString    | mqs/Top       | MQTT メッセージがパブリッシュされたトピック。                                                      |

MQTT パブリケーションに含まれるペイロードは、WebSphere MQ メッセージの内容に次のようにマップされます。

表 5. メッセージの内容

| メッセージの内容 | タイプ             | 目次                                       |
|----------|-----------------|------------------------------------------|
| BUFFER   | MQBYTE <i>n</i> | 着信 MQTT メッセージに含まれるバイトのコピー。長さがゼロの場合もあります。 |

## WebSphere MQ から MQTT クライアントへ

クライアントがパブリケーション・トピックにサブスクライブしました。WebSphere MQ アプリケーションがトピックにパブリッシュした結果、WebSphere MQ パブリッシュ/サブスクライブ・ブローカーによってパブリケーションが MQTT サブスクライバーに送信されました。あるいは、WebSphere MQ アプリケーションが非送信請求メッセージを直接 MQTT クライアントに送信しました。98 ページの表 6 は、MQTT クライアントに送信されるメッセージで固定メッセージ・ヘッダーがどのように設定されるかを示しています。WebSphere MQ メッセージ・ヘッダーの他のデータ、または他のヘッダーは破棄されます。WebSphere MQ メッセージのメッセージ・データは、変更されることなく、MQTT メッセージのメッセージ・ペイロードとして送信されます。MQTT メッセージは、遠隔測定 (MQXR) サービスによって MQTT クライアントに送信されます。

| MQTT フィールド    | タイプ  | 値                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DUP</b>    | ブール値 | QoS = 1 または 2 であり、前の伝送でこのクライアントにメッセージが送信されており、一定の時間が経過してもメッセージの確認応答がない場合に設定されます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>QoS</b>    | int  | <p>WebSphere MQ のパブリッシュ/サブスクライブ・ブローカーからの発信パブリケーションで QoS の値がどのように設定されるかは、着信パブリケーションによって決まります。それは、着信パブリケーションが MQTT クライアントから送信されたものであるか WebSphere MQ アプリケーションから送信されたものであるかによって異なります。</p> <p><b>MQTT</b><br/>着信パブリケーションでの QoS とサブスクライバーによって要求された QoS のうち、低い方の値。</p> <p><b>WebSphere MQ</b><br/>着信パブリケーションから派生した QoS の低い方の値</p> <p style="padding-left: 40px;">MQPER_NOT_PERSISTENT→QoS=0<br/>MQPER_PERSISTENT→QoS=2</p> <p>およびサブスクライバーにより要求された QoS。サブスクリプションなしでクライアントにメッセージが送信される場合、QoS はデフォルトで 2 に設定されます。クライアントはこの値を、異なる QoS を指定して DEFAULT.QoS にサブスクライブすることによって変更できます。</p> |
| <b>RETAIN</b> | ブール値 | 着信パブリケーションに保存プロパティ・セットがある場合に設定されます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

98 ページの表 7 は、MQTT クライアントに送信される MQTT メッセージで可変メッセージ・ヘッダーがどのように設定されるかを示しています。

| MQTT フィールド        | タイプ | 値                                                                                   |
|-------------------|-----|-------------------------------------------------------------------------------------|
| <b>Topic name</b> | 文字列 | パブリッシュされたメッセージに含まれるトピック・ストリング。                                                      |
| <b>Message ID</b> | 文字列 | パブリケーションが SYSTEM.MQTT.TRANSMIT.QUEUE に入れられたときのパブリケーションの MQMD.MsgId プロパティの最後の 2 バイト。 |

表 7. MQTT 可変ヘッダーにおけるプロパティ (続き)

| MQTT フィールド | タイプ    | 値                                                          |
|------------|--------|------------------------------------------------------------|
| Payload    | byte[] | 着信パブリケーションからパブリッシュ/サブスクライブ・ブローカーへのバイトの直接コピー。長さがゼロの場合もあります。 |

## Telemetry デーモン (デバイス用)

WebSphere MQ Telemetry デーモン (デバイス用) は、拡張 MQTT V3 クライアント・アプリケーションです。これは、他の MQTT クライアントからのメッセージをストア・アンド・フォワードするために使用します。MQTT クライアントのように WebSphere MQ に接続しますが、他の MQTT クライアントを接続することもできます。他の遠隔測定デーモンにも接続できます。

以下の 4 つの基本的な目的に役立ちます。

### 1. パブリッシュ/サブスクライブ・ネットワーク内のローカル MQTT クライアントを相互に接続する。

例えば、デバイスのセンサーとアクチュエーターを別々の MQTT クライアントとしてデーモンに接続することもできます。センサーは自身のゲージ測定値をパブリッシュします。アクチュエーターはその測定値にサブスクライブして、値に基づいて自身の動作を変更します。測定値はローカルに作用します。

### 2. キュー・マネージャーにパブリッシュするサブスクリプションとデバイスにパブリッシュするメッセージをフィルターに掛ける。

前の例では、デーモンが測定値をパブリッシュするトピックに WebSphere Message Broker メッセージ・フローがサブスクライブできます。フローは Web ページを更新し、デバイスの状態を示します。

デーモンは、アクチュエーターが作成したサブスクリプションをキュー・マネージャーに転送することもできます。WebSphere Message Broker フローは、アクチュエーターにサービスを提供している MQTT クライアントがサブスクライブしたトピックに、メッセージをパブリッシュします。MQTT クライアントは、デバイスの設定を変更します。

メッセージ・フローは、WebSphere Message Broker HTTPInput ノードを使用して、Web ページから開始することもできます。

### 3. 複数の MQTT クライアントをまとめて、遠隔測定サーバーとの 1 つの接続にする。

各デバイスを単独で遠隔測定サーバーに接続するのではなく、デーモンは単一 TCP/IP 接続でアプリケーションとサブスクリプションを転送します。デーモンを使用することにより、遠隔測定 (MQXR) サービスが管理する TCP/IP 接続の数が少なくなります。

個々の MQTT クライアントがデーモンに接続します。個々のクライアントはキュー・マネージャーには見えません。キュー・マネージャーに接続するすべてのクライアントのために、デーモンはキュー・マネージャーへの 1 つの接続を設けます。

### 4. デバイスとキュー・マネージャーの間でメッセージのストア・アンド・フォワードを行う。

デーモンは、キュー・マネージャーへの接続の短期接続障害から遠隔測定デバイスを保護する役割を担います。

例えば、デバイスが「応答不要送信」メッセージングのみサポートしているとします。キュー・マネージャーへの接続が断続的にしか使用できない場合や不確実である場合、デバイスには、情報を予測どおりに、つまり確実に転送する手段がありません。

解決策の 1 つは、常に使用可能なローカル接続を使用して、デバイスをデーモンに接続することです。デーモンは、キュー・マネージャーとの間を流れるメッセージを自身のバッファ・メモリーに入れます。不確実な接続であっても、キュー・マネージャーとの間でメッセージを高信頼サービス品質で送信できます。

**注:** デーモンには、"inflight" メッセージ用の永続ストレージがありません。メッセージはバッファ・メモリーに入れられます。

## MQTT のステートレス・セッションとステートフル・セッション

MQTT クライアントは、キュー・マネージャーを使用してステートフル・セッションを作成できます。ステートフル MQTT クライアントが切断した場合、キュー・マネージャーは、クライアントによって作成されたサブスクリプションと未完了メッセージを維持します。クライアントが再接続されると、未完了メッセージを解決します。配信用のキューに入れられているすべてのメッセージを送信し、切断中にそのサブスクリプションに対してパブリッシュされたすべてのメッセージを受信します。

MQTT クライアントは、遠隔測定チャンネルに接続すると、新しいセッションを開始するか、前のセッションを再開します。新しいセッションには、確認応答のない未解決メッセージも、サブスクリプションも、配信を待機しているパブリケーションもありません。クライアントは接続するときに、クリーン・セッションから開始するか、既存のセッションを再開するかを指定します ([クリーン・セッション](#)を参照)。

クライアントが既存のセッションを再開すると、接続の切断はなかったかのようにセッションが続行されます。配信を待っているパブリケーションはクライアントに送信され、コミットされていないメッセージの転送はすべて完了されます。クライアントが持続セッションで遠隔測定 (MQXR) サービスから切断しても、クライアントが作成したサブスクリプションは残ります。サブスクリプションに対応するパブリケーションが、クライアントが再接続した時点でクライアントに送信されます。クライアントが前のセッションを再開せずに再接続した場合、パブリケーションは遠隔測定 (MQXR) サービスによって破棄されます。

セッション状態情報は、キュー・マネージャーによって `SYSTEM.MQTT.PERSISTENT.STATE` キューに保存されます。

WebSphere MQ 管理者は、セッションを切断してページすることができます。

## MQTT クライアントが接続されていないとき

クライアントが接続されていないときも、キュー・マネージャーはクライアントのためにパブリケーションの受信を続行できます。こうしたパブリケーションは、クライアントが再接続した時点でクライアントに転送されます。クライアントが予期せず切断した場合、キュー・マネージャーはクライアントに代わって、この「遺言」をパブリッシュします。

クライアントが予期せず切断されたときに通知を受け取るようにしたい場合は、遺言パブリケーションを登録できます。遺言パブリケーションを参照してください。この「遺言」は、クライアントの要求なしにクライアントへの接続が切断されていることを遠隔測定 (MQXR) サービスが検出した場合に、遠隔測定サービスによって送信されます。

クライアントはいつでも保存パブリケーションをパブリッシュできます。保存パブリケーションと MQTT クライアントを参照してください。トピックへの新規サブスクリプションは、トピックに関連付けられた保存パブリケーションを送信するよう要求できます。「遺言」を保存パブリケーションとして作成すると、それを使用してクライアントの状況をモニターすることができます。

例えば、クライアントは接続時に保存パブリケーションをパブリッシュして、自身が使用可能であることを広告します。同時に、自身が使用不可であることを通告するための「遺言」保存パブリケーションも作成します。さらに、計画的な切断を行う直前に、自身が保存パブリケーションとして使用不可であることをパブリッシュします。クライアントが使用可能かどうかを知るには、その保存パブリケーションのトピックにサブスクライブすることになります。常に、3つのパブリケーションのうちの1つを受信することになります。

クライアントがその切断中にパブリッシュされたメッセージを受信するようにする場合は、クライアントを前のセッションに再接続します ([100 ページの『MQTT のステートレス・セッションとステートフル・セッション』](#)を参照)。前のセッションのサブスクリプションは、削除されるかクライアントがクリーン・セッションを作成するまではアクティブです。

## MQTT クライアントと WebSphere MQ アプリケーションの疎結合

MQTT クライアントと WebSphere MQ アプリケーション間のパブリケーションのフローは疎結合されています。パブリケーションは、MQTT クライアントまたは WebSphere MQ アプリケーションのいずれかから、順不同で発信されます。パブリッシャーとサブスクライバーは疎結合です。これらは、パブリケーションとサブスクリプションを介して間接的に対話します。WebSphere MQ アプリケーションから MQTT クライアントにメッセージを直接送信することもできます。

MQTT クライアントと WebSphere MQ アプリケーションは、以下の 2 つの理由により、疎結合していると言えます。

1. パブリッシャーとサブスクライバーは、パブリケーションおよびサブスクリプションのトピックとの関連により疎結合しています。パブリッシャーとサブスクライバーは、通常、パブリケーションまたはサブスクリプションの他のソースのアドレスまたは ID を認識しません。
2. MQTT クライアントは、パブリケーションのパブリッシュ、サブスクライブ、受信、および送達確認の処理を、別々のスレッドで行います。

MQTT クライアント・アプリケーションは、パブリケーションの送達を待機しません。アプリケーションは、メッセージを MQTT クライアントに渡してから、自身のスレッドの実行を続けます。アプリケーションとパブリケーションの送達の同期を取るのに送達トークンが使用されます ([送達トークン](#)を参照)。

MQTT クライアントにメッセージを渡した後、アプリケーションには送達トークンを待つという選択肢があります。クライアントは、待機するのではなく、パブリケーションが WebSphere MQ に送達されると呼び出されるコールバック・メソッドを定義できます。このコールバック・メソッドは、送達トークンを無視することもできます。

メソッドに関連付けられたサービス品質によって、送達トークンはコールバック・メソッドに直ちに返されるか、場合によってはかなり時間がたってから返されます。送達トークンは、クライアントが切断して再接続した後でも返されることがあります。サービス品質が「応答不要送信」である場合、送達トークンは直ちに返されます。これ以外の 2 つの場合は、サブスクライバーにパブリケーションが送信されたという確認応答をクライアントが受信したときのみ、送達トークンが返されます。

クライアント・サブスクリプションの結果として MQTT クライアントに送信されたパブリケーションは、`messageArrived` コールバック・メソッドに送達されます。`messageArrived` は、メインアプリケーションとは別スレッドで動作します。

## MQTT クライアントへのメッセージの直接送信

2 つの方法のいずれかを使用して、特定の MQTT クライアントにメッセージを送信することができます。

1. WebSphere MQ アプリケーションは、サブスクリプションなしで MQTT クライアントにメッセージを直接送信できます。[クライアントへのメッセージの直接送信](#)を参照してください。
2. 別の方法は、`ClientIdentifier` 命名規則を使用します。すべての MQTT サブスクライバーに、それぞれの固有の `ClientIdentifier` をトピックとして使用して、サブスクリプションを作成させます。`ClientIdentifier` にパブリッシュします。トピック `ClientIdentifier` にサブスクライブしたクライアントに、パブリケーションが送信されます。この手法を使用すれば、パブリケーションを特定の MQTT サブスクライバーに送信できます。

## WebSphere MQ Telemetry におけるセキュリティー

遠隔測定装置をセキュリティー保護することが重要になる場合があります。装置がポータブルであることと、綿密に管理できない場所で使用されることが予想されるからです。VPN を使用して、MQTT 装置から遠隔測定 (MQXR) サービスへの接続をセキュリティー保護できます。WebSphere MQ Telemetry では、これとは別の 2 つのセキュリティー・メカニズム (SSL と JAAS) を使用できます。

SSL は主に、装置と遠隔測定チャネルの間の通信を暗号化するため、および正しいサーバーに装置が接続しようとしていることを認証するために使用されます (SSL を使用する遠隔測定チャネルの認証を参照)。SSL を使用して、クライアント装置がサーバーに接続することを許可されているかどうかを検査することもできます (SSL を使用する MQTT クライアントの認証を参照)。

JAAS は主に、装置のユーザーがサーバー・アプリケーションを使用することを許可されているかどうかを検査するために使用されます ([パスワードを使用する MQTT クライアントの認証](#)を参照)。JAAS を LDAP と一緒に使用することにより、シングル・サインオン・ディレクトリーを使用してパスワードを検査できます。

SSL と JAAS を併用することにより、2 因子認証を行えます。SSL によって使用される暗号を FIPS 標準に適合する暗号に制限できます。

少なくとも何万というユーザーがいれば、個人のセキュリティー・プロファイルを用意することは必ずしも現実的とは限りません。そしてまた、個別ユーザーが WebSphere MQ オブジェクトにアクセスするのを

プロファイルを使用して許可するのも、必ずしも現実的ではありません。代わりに、トピックへのパブリケーションとサブスクリプションを許可するためのクラス、およびクライアントにパブリケーションを送信するためのクラスに、ユーザーをグループ化します。

各遠隔測定チャンネルを構成して、クライアントをクライアント共通ユーザー ID にマップします。特定のチャンネルで接続するすべてのクライアントに、共通ユーザー ID を使用します ([MQTT クライアントの ID および許可を参照](#))。

ユーザーのグループを許可することが、各個人の認証を阻害するわけではありません。各個別ユーザーをクライアントまたはサーバーでそれぞれのユーザー名とパスワードで認証した後、サーバーで共通ユーザー ID を使用して許可することができます。

## WebSphere MQ Telemetry におけるグローバル化

MQTT v3 プロトコルでは、メッセージ・ペイロードはバイト配列としてエンコードされます。一般に、テキストを扱うアプリケーションは、UTF-8 でメッセージ・ペイロードを作成します。遠隔測定チャンネルは、メッセージ・ペイロードを UTF-8 であると示しますが、コード・ページ変換はまったく行いません。パブリケーション・トピックのストリングは UTF-8 でなければなりません。

アプリケーションは、英字データを正しいコード・ページに変換し、数値データを正しい数値エンコード方式に変換する必要があります。

MQTT Java クライアントには、便利な `MqttMessage.toString` メソッドがあります。このメソッドは、メッセージ・ペイロードを、ローカル・プラットフォームのデフォルト文字セット (一般に UTF-8) でエンコードされているものとして扱います。ペイロードを Java ストリングに変換します。Java には、ストリング・メソッド `getBytes` があります。このメソッドは、ローカル・プラットフォームのデフォルト文字セットを使用してエンコードされたバイト配列にストリングを変換します。同じデフォルト文字セットを持つプラットフォーム間で、メッセージ・ペイロード内のテキストを交換する 2 つの MQTT Java プログラムは、UTF-8 では非常に簡単かつ効率的に機能します。

いずれかのプラットフォームのデフォルト文字セットが UTF-8 でない場合、アプリケーションはメッセージの交換のためのきまりを確立しなければなりません。例えば、パブリッシャーは、`getBytes("UTF8")` メソッドを使用して、ストリングから UTF-8 への変換を指定します。メッセージのテキストを受け取るには、サブスクライバーはメッセージが UTF-8 文字セットでエンコードされているものと見なします。

遠隔測定 (MQXR) サービスは、MQTT クライアントのメッセージからのすべての着信パブリケーションのエンコード方式を、UTF-8 であると示します。MQMD.CodedCharSetId が UTF-8 に、RFH2.CodedCharSetId が MQCCSI\_INHERIT に、それぞれ設定されます ([96 ページの『WebSphere MQ Telemetry とキュー・マネージャーの統合』を参照](#))。パブリケーションのフォーマットが MQFMT\_NONE に設定されるので、チャンネルによる、つまり MQGET による変換は行えません。

## WebSphere MQ Telemetry のパフォーマンスと拡張容易性

多数のクライアントを管理するときや WebSphere MQ Telemetry の拡張容易性を高めるときは、以下の要因を考慮に入れてください。

### キャパシティー・プランニング

WebSphere MQ Telemetry のパフォーマンス・レポートについては、「[WebSphere MQ Family-Performance Reports](#)」の「WebSphere MQ Telemetry パフォーマンス評価レポート」を選択してください。

### 接続

接続に関係するコストには、以下の要素が含まれます。

- プロセッサ使用率とプロセッサ時間の点から見た接続そのもののセットアップ・コスト。
- ネットワーク・コスト。
- 開いたままの接続を使用しないときの使用メモリー。

クライアントが接続されたままのときは、さらに負荷が発生します。接続が開いたままの場合、TCP/IP フローと MQTT メッセージではネットワークを使用して、まだ接続されているかどうかを確認されます。加えて、開いたままのクライアント接続ごとに、サーバーでメモリーが使用されます。

メッセージを毎分複数回の頻度で送信する場合は、新たに接続を開始するときのコストを回避するために、接続を開いたままにしておきます。メッセージを 10 分から 15 分ごとに 1 回未満の頻度で送信する場合は、接続を開いたままにしておくときのコストを回避するために、切断することを検討してください。SSL 接続は、セットアップにより多くの費用がかかるため、使用されない状態で長時間開いたままにしておいてもかまいません。

また、クライアントの能力も考慮に入れてください。クライアントにストア・アンド・フォワード機能がある場合は、メッセージをバッチにまとめて、バッチの送信と送信の間は接続を切断することができます。ただし、クライアントが切断されると、クライアントはサーバーからのメッセージを受信できなくなります。したがって、判断にはアプリケーションの目的が関係します。

システムのクライアントが 1 つで、それが多数のメッセージを送信する場合は (例えばファイル転送)、メッセージごとにサーバーの応答を待つことはしないでください。代わりに、すべてのメッセージを送信し、最後にそれらがすべて受信されたことを検査します。あるいは、[サービス品質 \(QoS\)](#) を使用します。

QoS をメッセージによって変更することができます。QoS 0 を使用する重要でないメッセージと、QoS 2 を使用する重要なメッセージを送達することができます。メッセージのスループットは、QoS が 2 の場合よりも、QoS が 0 の 2 倍程度になることがあります。

## 命名規則

多数のクライアントに対応するアプリケーションを設計する場合は、実効的な命名規則を実施してください。各クライアントを正しい `ClientIdentifier` にマップするために、`ClientIdentifier` を意味のある名前にしてください。適切な命名規則により、管理者が実行中のクライアントを判別しやすくなります。命名規則は、管理者が WebSphere MQ エクスプローラーで多数のクライアントのリストをフィルタリングする際に役立ち、また、問題の判別にも役立ちます ([クライアント ID](#) を参照)。

## スループット

トピック名の長さは、ネットワーク上を流れるバイト数に影響します。パブリッシュまたはサブスクライブ時に、メッセージに含まれるバイト数が重要になることがあります。したがって、トピック名の文字数を制限してください。MQTT クライアントがトピックをサブスクライブすると、WebSphere MQ はそれに次の形式の名前を付けます。

```
ClientIdentifier:TopicName
```

MQTT クライアントのサブスクリプションをすべて表示するには、WebSphere MQ MQSC **DISPLAY** コマンドを次のように使用します。

```
DISPLAY SUB('ClientID1:*')
```

## MQTT クライアントが使用する WebSphere MQ のリソースの定義

MQTT クライアントは、IBM WebSphere MQ リモート・キュー・マネージャーに接続します。WebSphere MQ アプリケーションが MQTT クライアントにメッセージを送信するには、2 つの基本的な方法があります。それは、デフォルト伝送キューを `SYSTEM.MQTT.TRANSMIT.QUEUE` に設定する方法と、キュー・マネージャーの別名を使用する方法です。MQTT クライアントが多数ある場合は、キュー・マネージャーのデフォルト伝送キューを定義します。デフォルトの伝送キュー設定を使用すると、管理作業が簡略化されます。[MQTT クライアントにメッセージを送信するように分散キューイングを構成する](#) を参照してください。

## サブスクリプションを回避することによる拡張容易性の向上

MQTT V3 クライアントがトピックにサブスクライブすると、遠隔測定 (MQXR) サービスによってサブスクリプションが WebSphere MQ で作成されます。サブスクリプションにより、クライアントのパブリケーシ

ョンが SYSTEM.MQTT.TRANSMIT.QUEUE に送信されます。各パブリケーションの伝送ヘッダーのリモート・キュー・マネージャー名は、サブスクリプションを作成した MQTT クライアントの ClientIdentifier に設定されます。多数のクライアントがそれぞれ独自のサブスクリプションを行う場合は、WebSphere MQ パブリッシュ/サブスクライブのクラスターまたは階層の全体にわたって、多数のプロキシー・サブスクリプションが維持されることになります。パブリッシュ/サブスクライブの代わりに、Point-to-Point ベースのソリューションを使用する方法については、[クライアントへのメッセージの直接送信](#)を参照してください。

## 多数のクライアントの管理

同時に接続される多数のクライアントをサポートするには、JVM パラメーターの **-Xms** と **-Xmx** を設定して、遠隔測定 (MQXR) サービスで使用可能なメモリーを増やします。以下のステップに従ってください。

1. テレメトリー・サービスの構成ディレクトリーで `java.properties` ファイルを見つけます。  
[Windows でのテレメトリー \(MQXR\) サービスの構成ディレクトリー](#) または [Linux でのテレメトリー・サービスの構成ディレクトリー](#) を参照してください。
2. ファイルの指示に従います。同時に接続されるクライアントの数が 50,000 の場合は、1 GB のヒープで十分です。

```
Heap sizing options - uncomment the following lines to set the heap to 1G
#-Xmx1024m
#-Xms1024m
```

3. `java.properties` ファイル内でテレメトリー (MQXR) サービスを実行している JVM に渡すために、他のコマンド行引数を追加します。[テレメトリー \(MQXR\) サービスへの JVM パラメーターの引き渡し](#) を参照してください。

Linux 上のオープン・ファイル記述子の数を増やすには、以下の行を `/etc/security/limits.conf/` に追加し、再度ログインしてください。

```
@mqm soft nofile 65000
@mqm hard nofile 65000
```

ソケットごとに1つのファイル記述子を必要とします。遠隔測定サービスでさらにファイル記述子がいくつか必要になるため、この数は、必要なオープン・ソケットの数よりも多くする必要があります。

キュー・マネージャーは、非永続サブスクリプションごとに1つのオブジェクト処理を使用します。多数のアクティブな非永続サブスクリプションをサポートするには、キュー・マネージャーで使用できるアクティブ・ハンドルの最大数を増やします。例えば、以下のようになります。

```
echo ALTER QMGR MAXHANDS(99999999) | runmqsc qMgrName
```

### 図 26. Windows でのハンドルの最大数の変更

```
echo "ALTER QMGR MAXHANDS(99999999)" | runmqsc qMgrName
```

### 図 27. Linux でのハンドルの最大数の変更

## その他の考慮事項

システム要件を計画する際は、システムの再始動に要する時間の長さを考慮に入れてください。計画したダウン時間が、処理待ちのキューに入れられるメッセージの数に影響する可能性があります。許容時間内にメッセージを正常に処理できるように、システムを構成してください。ディスク・ストレージ、メモリー、および処理能力を確認してください。一部のクライアント・アプリケーションでは、クライアントの再接続時にメッセージを破棄できる場合があります。メッセージを破棄するには、クライアント接続パラメーターで `CleanSession` を設定します ([クリーン・セッション](#)を参照)。あるいは、MQTT クライアントでベスト・エフォートのサービス品質 `0` を使用して、パブリッシュおよびサブスクライブを行います ([サービス品質](#)を参照)。WebSphere MQ からのメッセージの送信時には、`non-persistent` メッセージを使用

します。これらのサービス品質を使用したメッセージは、システムまたは接続の再始動時には回復されません。

## IBM WebSphere MQ Telemetry がサポートする装置

MQTT クライアントは、センサーとアクチュエーターからハンドヘルド・デバイス、車両システムまで、さまざまな装置の上で実行できます。

MQTT クライアントは小さいので、小容量メモリーや低処理能力に制約される装置上で動作します。MQTT プロトコルは信頼性が高くヘッダーが小さいので、低帯域幅、高コスト、および使用可能状態が断続的であることに制約されるネットワークに適しています。

IBM WebSphere MQ Telemetry には、次の 3 つのクライアントが用意されており、それらすべてにおいて MQTT v3 プロトコルが実装されています。

- 最小の CLDC (Connected Limited Device Configuration) /MIDP (Mobile Information Device Profile) から CDC (Connected Device Configuration) /Foundation、J2SE (Java Platform, Standard Edition)、および J2EE (Java Platform, Enterprise Edition) のすべてのバリエーションの Java 上で実行できる Java クライアント。IBM jclRM カスタマイズ・クラス・ライブラリーもサポートされています。
- Windows システムおよび Linux システム用の事前構築されたネイティブ・クライアントとともに C リファレンス実装。この C リファレンス実装に基づいて、広範囲の装置やプラットフォームに MQTT を移植できます。
- C で作成され、任意の適切な Linux または Windows プラットフォームで実行できる、拡張クライアント IBM WebSphere MQ Telemetry デーモン (デバイス用)。

Intel 上の一部の Windows システム (Windows XP、RedHat、Ubuntu など)、および ARM プラットフォーム上の一部の Linux システム (Eurotech Viper など) は、C クライアントを実行する Linux のバージョンを実装していますが、IBM はプラットフォームのサービス・サポートを提供していません。IBM サポート・センターに連絡を取る場合は、サポートされているプラットフォーム上でクライアントの問題を再現しておく必要があります。

Java ME プラットフォームは通常、アクチュエーター、センサー、携帯電話、その他の組み込みデバイスなどの小型デバイスで使用されます。Java SE プラットフォームは通常、デスクトップ・コンピューターやサーバーなどの上位の組み込みデバイスにインストールされます。

注: [EclipsePaho](#) プロジェクトと [MQTT.org](#) には、プログラミング言語の範囲の最新のテレメトリー・クライアントとサンプルの無料ダウンロードがあります。

### 関連概念

99 ページの『[Telemetry デーモン \(デバイス用\)](#)』

WebSphere MQ Telemetry デーモン (デバイス用) は、拡張 MQTT V3 クライアント・アプリケーションです。これは、他の MQTT クライアントからのメッセージをストア・アンド・フォワードするために使用します。MQTT クライアントのように WebSphere MQ に接続しますが、他の MQTT クライアントを接続することもできます。他の遠隔測定デーモンにも接続できます。

## オブジェクトの管理

キュー・マネージャーは、IBM WebSphere MQ オブジェクトのプロパティ、つまり属性を定義します。それらの属性の値は、WebSphere MQ がこれらのオブジェクトを処理する方法に影響します。アプリケーションから、Message Queue Interface (MQI) を使用してオブジェクトを制御します。プログラムからアドレッシングされる時、オブジェクトは MQ オブジェクト記述子 (MQOD) によって識別されます。

WebSphere MQ コマンドを使用してオブジェクトの定義、変更、削除などのオブジェクト管理操作を行う場合、キュー・マネージャーは、ユーザーがそれらの操作を実行するのに必要なレベルの権限を持っているかどうかを検査します。同様に、アプリケーションが MQOPEN 呼び出しを使用してオブジェクトをオープンするとき、キュー・マネージャーは、そのオブジェクトへのアクセスを許可する前に、アプリケーションが必要なレベルの権限を持っているかどうかを検査します。検査は、オープンされているオブジェクトの名前に対して行われます。

オブジェクトの管理には、以下のタスクが含まれます。

- キュー・マネージャーの始動および停止

- アプリケーション用のオブジェクト (特にキュー) の作成
- チャンネルを使用して、他の (リモート) システムにあるキュー・マネージャーへの通信パスを作成。
- キュー・マネージャーのクラスターを作成することによって、管理プロセス全体を簡易化し、ワークロードのバランスをとる。

WebSphere MQ オブジェクトの作成と管理方法の概要については、[127 ページの『オブジェクトの管理』](#)を参照してください。

オブジェクト管理タスクについて詳しくは、以下のサブトピックを参照してください。

## オブジェクト

管理タスクの多くには、さまざまな種類の IBM WebSphere MQ オブジェクト の操作が関係します。

IBM WebSphere MQ オブジェクトの命名については、[123 ページの『IBM WebSphere MQ オブジェクトの命名』](#)を参照してください。

キュー・マネージャー上に作成されるデフォルトのオブジェクトについては、[128 ページの『システム・デフォルト・オブジェクト』](#)を参照してください。

異なるタイプの IBM WebSphere MQ オブジェクトについての詳細は、以下のサブトピックを参照してください。

### 関連概念

[42 ページの『メッセージ・キューイングの概要』](#)

WebSphere MQ 製品では、整合性のあるアプリケーション・プログラミング・インターフェースを使用して、性質の異なるコンポーネント (プロセッサ、オペレーティング・システム、サブシステム、および通信プロトコル) のネットワーク内でプログラムが相互に通信できるようにしています。

[128 ページの『オブジェクトの属性』](#)

オブジェクトのプロパティは、その属性によって定義されます。属性の一部はユーザーが指定できますが、他の属性は表示のみ可能です。

### 関連資料

[MQSC コマンド](#)

## キュー

WebSphere MQ のキューとキュー属性の紹介。

WebSphere MQ キュー は名前付きオブジェクトで、アプリケーションはそこにメッセージを書き込んだり、そこからメッセージを読み取ったりできます。

メッセージはキューに入れられるので、メッセージを書き込んだアプリケーションは、そのメッセージに対する応答を期待する場合でも、応答を待っている間に他の作業を自由に行うことができます。アプリケーションは Message Queue Interface (MQI) を用いてキューにアクセスします。MQI については、[Message Queue Interface の概要](#)で説明します。

メッセージをキューに書き込むには、その前にキューが作成されている必要があります。キューはキュー・マネージャーによって所有され、そのキュー・マネージャーは多数のキューを所有できます。ただし、各キューはそのキュー・マネージャー内で固有の名前を持っていないければなりません。

キューはキュー・マネージャーを通じて保守されます。ほとんどの場合、各キューはそのキュー・マネージャーによって物理的に管理されますが、このことはアプリケーション・プログラムからは認識されません。WebSphere MQ for z/OS 共有キューは、キュー共有グループ内の任意のキュー・マネージャーによって管理できます。

キューを作成するには、WebSphere MQ コマンド (MQSC)、PCF コマンド、または WebSphere MQ for z/OS 操作および制御パネルなどのプラットフォーム固有のインターフェースを使用します。

一時的なジョブ用のローカル・キューを独自のアプリケーションから動的に作成できます。例えば、応答先キュー (アプリケーションが終了したら必要なくなる) を作成できます。詳しくは、[112 ページの『動的キューとモデル・キュー』](#)を参照してください。

キューを使用する前に、そのキューで何を行いたいかを指定して、キューをオープンする必要があります。例えば、以下の目的でキューをオープンすることができます。

- メッセージのブラウズのみ (取り出しは行わない)
- メッセージの取り出し (他のプログラムとの共用アクセスか、または排他的アクセスで)
- キューへのメッセージの書き込み
- キューの属性の照会
- キューの属性の設定

キューを開くときに指定できるオプションの完全なリストについては、[MQOPEN - オブジェクトのオープン](#)を参照してください。

## キューの属性

キューの属性には、そのキューが定義される時に指定され、後からは変更できないものがあります (例えば、キューのタイプ)。キューのその他の属性は、次のいずれかの方法で変更可能な属性にグループ分けできます。

- キューの処理中にキュー・マネージャーによって (例えば、キューの現在のサイズ)
- コマンドによってのみ (例えば、キューのテキスト記述)
- アプリケーションが MQSET 呼び出しを使用して (例えば、そのキューに書き込み操作ができるかどうか)

MQINQ 呼び出しを用いてすべての属性の値を知ることができます。

複数のキュー・タイプに共通な属性として次のものがあります。

### **QName**

キューの名前

### **QType**

キューのタイプ

### **QDesc**

キューのテキスト記述

### **InhibitGet**

プログラムがそのキューからメッセージを読み取ることができるかどうか (ただし、リモート・キューからはメッセージを読み取ることはできません)

### **InhibitPut**

プログラムがそのキューにメッセージを書き込むことができるかどうか

### **DefPriority**

キューに書き込まれたメッセージのデフォルト優先順位

### **DefPersistence**

キューに書き込まれたメッセージのデフォルト持続性

### **Scope (z/OS ではサポートされていません)**

このキューに対するエントリーが名前サービスでも存在するか制御します。

これらの属性の詳細な説明については、[キューの属性](#)を参照してください。

## 関連概念

### [109 ページの『リモート・キュー』](#)

プログラムが接続されているキュー・マネージャーとは別のキュー・マネージャーが所有するキューは、そのプログラムにとってリモートです。

### [110 ページの『別名キュー』](#)

別名キューとは、別のキューまたはトピックにアクセスするために使用できる WebSphere MQ オブジェクトです。これは、複数のプログラムがその同じキューを別の名前でもアクセスして、作業できることを意味しています。

### [114 ページの『キューの定義』](#)

IBM WebSphere MQ にキューを定義するには、MQSC コマンドの DEFINE または PCF Create Queue コマンドを使います。

#### [114 ページの『IBM WebSphere MQ によって使用されるキュー』](#)

IBM WebSphere MQ は、一部のローカル・キューを、その操作に関連する特定の目的のために使用します。

### 関連タスク

[アプリケーションの開発に関する参照情報](#)

### 関連資料

#### [MQSC コマンド](#)

#### [108 ページの『ローカル・キュー』](#)

ローカル・キューのタイプは、伝送、開始、送達不能、コマンド、デフォルト、チャンネル、およびイベント・キューです。

#### [110 ページの『共用キューとクラスター・キュー』](#)

ここでは、共用キューとクラスター・キューの用語を定義し、説明します。また両者の比較について説明します。

#### [112 ページの『動的キューとモデル・キュー』](#)

この情報は、動的キュー、一時および永続の動的キューのプロパティ、動的キューの使用方法、動的キューを使用する際の考慮事項、およびモデル・キューについて詳しく説明するものです。

### ローカル・キュー

ローカル・キューのタイプは、伝送、開始、送達不能、コマンド、デフォルト、チャンネル、およびイベント・キューです。

キューは、プログラムが接続されているキュー・マネージャーに所有される場合、プログラムではローカルと認知されます。メッセージは、ローカル・キューから取得し、ローカル・キューに入れることができます。

キュー定義オブジェクトは、キューに入る物理メッセージと同様、そのキューの定義情報を保持します。

各キュー・マネージャーは、ある特別な目的のために用いられる次のようなローカル・キューを持つことができます。

### 伝送キュー

アプリケーションがメッセージをリモート・キューに送信するとき、ローカル・キュー・マネージャーはそのメッセージを、伝送キューと呼ばれる特別のローカル・キューに保管します。

メッセージ・チャンネル・エージェントは、伝送キューと関連付けられたチャンネル・プログラムで、次の宛先にメッセージを送達します。次の宛先とは、メッセージ・チャンネルが接続されるキュー・マネージャーです。それはメッセージの最終宛先と同じキュー・マネージャーである必要はありません。メッセージは、次の宛先に送達されると伝送キューから削除されます。メッセージは、最終宛先までの経路上でいくつものキュー・マネージャーを通過しなければならない場合があります。経路上の各キュー・マネージャーには、伝送キューを定義する必要があります。各伝送キューは、次の宛先に伝送されるのを待っているメッセージを保持します。通常の伝送キューは、メッセージの最終的な宛先が異なる場合でも、次の宛先へのメッセージを保持します。クラスター伝送キューは、複数の宛先へのメッセージを保持します。各メッセージの `correlID` は、次の宛先に転送するためにメッセージが入れられるチャンネルを識別します。

キュー・マネージャーには、複数の伝送キューを定義できます。同じ宛先に対して、それぞれが異なるサービスのクラスに用いられる伝送キューをいくつか定義する場合があります。例えば、同じ宛先に送る小規模メッセージと大規模メッセージに異なる伝送キューを作成したい場合があります。そうすれば異なるメッセージ・チャンネルを使用してメッセージを転送できるため、大規模メッセージが小規模メッセージを遅らせてしまうことがありません。z/OS 以外のプラットフォームでは、メッセージはデフォルトで単一クラスター伝送キュー `SYSTEM.CLUSTER.TRANSMIT.QUEUE` に送られます。他のプラットフォームでのオプションとして、デフォルトを変更し、異なるクラスター・キュー・マネージャーから異なるクラスター伝送キューに行くようメッセージ・トラフィックを分離することができます。キュー・マネージャー属性 `DEFCLXQ` を `CHANNEL` に設定すると、各クラスター送信側チャンネルは別々のクラスター伝送キューを作成します。他のプラットフォームのもう 1 つのオプションは、クラスター送信側チャンネルが使用するクラスター伝送キューを手動で定義することです。

伝送キューは、メッセージ・チャンネル・エージェントをトリガーしてメッセージを送信することができます。[トリガーによる WebSphere MQ アプリケーションの開始](#)を参照してください。

## 開始キュー

開始キューは、トリガー・イベントがアプリケーション・キューで起きたときに、キュー・マネージャーがトリガー・メッセージを書き込むローカル・キューです。

トリガー・イベントとは、プログラムにキューの処理を開始させることを目的とするイベントのことです。例えば、到着メッセージが 10 件を超えるというイベントなどがあります。トリガー操作の詳細については、[トリガーによる WebSphere MQ アプリケーションの開始](#)を参照してください。

## 送達不能 (未配布メッセージ) キュー

送達不能 (未配布メッセージ) キューは、キュー・マネージャーが送達不能なメッセージを書き込むローカル・キューです。

キュー・マネージャーは、メッセージを送達不能キューに書き込むときに、メッセージにヘッダーを追加します。ヘッダー情報には、キュー・マネージャーがそのメッセージを送達不能キューに書き込んだ理由が入っています。また、元のメッセージの宛先、キュー・マネージャーがそのメッセージを送達不能キューに書き込んだ日時なども入っています。

また、アプリケーションも送達不能のメッセージ用にキューを使用できます。詳細については、[送達不能 \(未配布メッセージ\) キューの使用](#)を参照してください。

## システム・コマンド・キュー

システム・コマンド・キューは、適正な許可を持つアプリケーションが WebSphere MQ コマンドを送信できるキューです。これらのキューはプラットフォームでサポートされている PCF、MQSC および CL コマンドを受信し、キュー・マネージャーがそれらを処理できるようにします。

## システム・デフォルト・キュー

システム・デフォルト・キューには、ご使用のシステム用のキューの初期定義が格納されています。キュー定義を作成すると、キュー・マネージャーは該当のシステム・デフォルト・キューからその定義をコピーします。キュー定義の作成は、動的キューの作成とは異なります。動的キューの定義は、動的キューのテンプレートとして選択したモデル・キューに基づきます。

## イベント・キュー

イベント・キューはイベント・メッセージを保持します。これらのメッセージは、キュー・マネージャーまたはチャンネルによって報告されます。

## リモート・キュー

プログラムが接続されているキュー・マネージャーとは別のキュー・マネージャーが所有するキューは、そのプログラムにとってリモートです。

通信リンクが確立されている場合、プログラムはリモート・キューにメッセージを送信できます。ただし、プログラムはリモート・キューからメッセージを読み取ることはできません。

キュー定義オブジェクトはリモート・キューを定義する際に作成され、ローカル・キュー・マネージャーがメッセージの送信先のキューを見つけるために必要な情報だけを保持します。このオブジェクトは、リモート・キューのローカル定義と呼ばれます。リモート・キューの属性は、すべてそれを所有するキュー・マネージャーによって保持されますが、これは、リモート・キューが、そのキュー・マネージャーにとってはローカル・キューになるためです。

リモート・キューをオープンするときは、キューを識別するために次のどちらかを指定しなければなりません。

- リモート・キューを定義するローカル定義の名前。

リモート・キューのローカル定義を作成するには、DEFINE QREMOTE コマンドを使用します。WebSphere MQ for IBM i では、CRTMQMQ コマンドを使用します。

アプリケーションの視点からは、これはローカル・キューのオープンと同じです。アプリケーションは、キューがローカルかリモートかを認識する必要はありません。

- リモート・キュー・マネージャーの名前と、そのリモート・キュー・マネージャーに認識されているキューの名前。

107 ページの『[キューの属性](#)』で説明した共通属性のほかに、リモート・キューのローカル定義には3つの属性があります。これらは、*RemoteQName* (キューを所有するキュー・マネージャーが認識する名前)、*RemoteQMgrName* (所有するキュー・マネージャーの名前)、および *XmitQName* (メッセージを他のキュー・マネージャーに転送するときに使用されるローカル伝送キューの名前) です。これらの属性の詳細な説明については、[キューの属性](#)を参照してください。

リモート・キューのローカル定義に対し MQINQ 呼び出しを使用する場合、キュー・マネージャーが戻すのは、リモート・キュー名、リモート・キュー・マネージャー名、および伝送キュー名のローカル定義の属性だけで、リモート・システム内で一致するローカル・キューの属性は戻しません。

[伝送キュー](#)も参照してください。

### 別名キュー

別名キューとは、別のキューまたはトピックにアクセスするために使用できる WebSphere MQ オブジェクトです。これは、複数のプログラムがその同じキューを別の名前でもアクセスして、作業できることを意味しています。

別名を解決した結果として得られるキュー (基本キューと呼ばれる)は、ローカル・キュー、リモート・キューのローカル定義、または共用キュー (WebSphere MQ for z/OS でのみ使用可能なローカル・キューのタイプ) のいずれかです。これは、プラットフォームでサポートされている事前定義キュー、または動的キューのどちらかです。

別名を解決した結果がトピックになることもあります。現在はキューにメッセージを書き込むアプリケーションがある場合、キューの名前をトピックの別名にすれば、そのアプリケーションはトピックへのパブリッシュを行うようになります。アプリケーション・コードを変更する必要はありません。

注: 別名は、ローカルに定義された別の別名キューに解決できません。

別名キューの使用例は、システム管理者が、基本キューの名前 (つまり、別名が解決された結果のキュー名) と別名キューの名前に異なったアクセス許可を与える場合です。これは、プログラムまたはユーザーが、基本キューではなく、別名キューを使用するように許可されることを意味します。

または、別名の書き込み操作は禁止し、基本キューの書き込み操作は可能になるように許可を設定できます。

あるアプリケーションでは、別名キューの使用により、システム管理者がアプリケーションを変更しなくても、別名キュー・オブジェクトの定義を容易に変更できるようになります。

プログラムが別名を使用しようとする時、WebSphere MQ は、別名に対する許可検査を行います。ただし、プログラムに、その別名が解決された結果の名前にアクセスする許可が与えられているかどうかは検査しません。したがって、プログラムには別名キューの名前にアクセスする許可は与えられますが、解決された結果のキュー名にアクセスする許可は与えられません。

106 ページの『[キュー](#)』で説明されている一般的なキュー属性に加えて、別名キューには *BaseQName* 属性があります。これは、別名が解決されたときの基本キューの名前です。この属性の詳細については、[BaseQName \(MQCHAR48\)](#) を参照してください。

別名キューの *InhibitGet* 属性と *InhibitPut* 属性 (106 ページの『[キュー](#)』を参照) は、別名に属します。例えば、別名キュー名 ALIAS1 が基本キュー名 BASE に解決される場合は、ALIAS1 に対する禁止事項は ALIAS1 だけを対象にし、BASE は禁止されません。ただし、BASE に対する禁止事項は ALIAS1 にも影響を及ぼします。

*DefPriority* 属性と *DefPersistence* 属性も別名に属します。したがって、例えば、同じ基本キューの異なる別名に異なるデフォルト優先順位を割り当てることができます。さらに、別名を使用するアプリケーションを変更しなくても、これらの優先順位を変えることもできます。

### 共用キューとクラスター・キュー

ここでは、共用キューとクラスター・キューの用語を定義し、説明します。また両者の比較について説明します。

## 共有キュー

共有キューとは、キュー共用グループ内にある1つ以上のキュー・マネージャーがアクセスできるメッセージを持つ、一種のローカル・キューです。**共有キューは WebSphere MQ for z/OS でのみ使用可能です。**(これは、同一のキュー・マネージャーを使用して複数のアプリケーションによって共有されるキューとは異なります。)共有キューは Coupling Facility (CF) によって保持され、キュー共用グループ内のどのキュー・マネージャーからでもアクセスできます。キュー共用グループ内の共有キューはそれぞれ、そのグループ内で固有の名前を持っていないければなりません。

## クラスター・キュー

クラスター・キューとは、クラスター・キュー・マネージャーでホストされ、同じクラスター内の別のキュー・マネージャーで使用できるキューです。

クラスター・キュー・マネージャーは、キューを使用可能にするクラスターの名前を指定して、そのキューのローカル・キュー定義を作成します。この定義は、クラスター内の他のキュー・マネージャーにキューを知らせます。クラスター内の他のキュー・マネージャーは、対応するリモート・キュー定義がなくても、クラスター・キューにメッセージを書き込むことができます。また、クラスター・キューを複数のクラスターに通知することもできます。詳細については、[クラスターおよびキュー・マネージャー・クラスターの構成](#)を参照してください。

## 共有キューとクラスター・キューの比較

ここでは、共有キューおよびクラスター・キューを比較し、ご使用のシステムにどちらが適切かを判断できるようにすることを目的としています。

## ムーバーのコスト

クラスター・キューでは、メッセージはムーバーによって送信されるので、アプリケーションのコストに加えてムーバーのコストを考慮に入れます。チャンネルがメッセージを取得し、書き込むため、ネットワークでコストが発生します。共有キューにはこれらのコストがありません。そのため、キュー共用グループ内のキュー・マネージャー間でメッセージを移動するとき、共有キューでは、クラスター・キューよりも処理能力の使用が少なくなります。

## メッセージの可用性

クラスター・キューの場合、キューに書き込みを行うと、メッセージは、ご使用のキュー・マネージャーへのアクティブなチャンネルが接続されているキュー・マネージャーの1つへ送信されます。リモート・キュー・マネージャーでは、メッセージを処理するために使用するアプリケーションが機能していない場合、メッセージは処理されず、アプリケーションが始動するまで待機します。同様に、キュー・マネージャーがシャットダウンすると、そのキュー・マネージャー上のメッセージは、キュー・マネージャーが再始動するまで使用可能になりません。これらの事例から、共有キューを使用する場合に比べてメッセージの可用性が低いことが分かります。

共有キューを使用している場合、キュー共用グループ内のどのアプリケーションも、送信されたメッセージを取得することができます。キュー共用グループで1つのキュー・マネージャーをシャットダウンした場合、メッセージは他のキュー・マネージャーで使用することができるので、クラスター・キューを使用する場合に比べてメッセージの可用性が高くなります。

## キャパシティ

カップリング・ファシリティは、ディスクよりも高価です。それで、ローカル・キューに1,000,000メッセージを格納するためのコストは、同じ数のメッセージを十分な容量のカップリング・ファシリティに格納するよりも低くなります。

## 他のキュー・マネージャーへの送信

共用キュー・メッセージは、キュー共用グループでのみ使用可能です。キュー共用グループの外側でキュー・マネージャーを使用する場合、ムーバーを使用する必要があります。複数のリモートの分散キュー・マネージャー間でワークロード・バランスをとるために、クラスタリングを使用することができます。

## ワークロード・バランシング

クラスタリングを使用して、チャンネルおよびキュー・マネージャーに対して、送信されたメッセージを取得する比率に重みづけをすることができます。例えば、1つのキュー・マネージャーにメッセージの60%を送信し、別のキュー・マネージャーにメッセージの40%を送信することができます。この事例は、リモート・キュー・マネージャーのプロセス処理能力に左右されません。最初のキュー・マネージャーのシステムは過負荷になる可能性があり、2番目のキュー・マネージャーのシステムはアイドル状態になる可能性があります。しかし、メッセージの多くは、最初のキュー・マネージャーに依然として送信されます。

共用キューを使用すると、2つの CICS システムがメッセージを取得できます。1つのシステムが過負荷になると、もう一方のシステムが大部分の作業負荷を引き継ぎます。

### 動的キューとモデル・キュー

この情報は、動的キュー、一時および永続の動的キューのプロパティ、動的キューの使用法、動的キューを使用する際の考慮事項、およびモデル・キューについて詳しく説明するものです。

アプリケーション・プログラムがモデル・キューをオープンするために MQOPEN 呼び出しを出すと、キュー・マネージャーは動的に、そのモデル・キューと同じ属性をもつローカル・キューのインスタンスを作成します。モデル・キューの *DefinitionType* フィールドの値に応じて、キュー・マネージャーは一時動的キューまたは永続動的キューのいずれかを作成します ([動的キューの作成](#)を参照)。

### 一時動的キューの特性

一時動的キューは次のような特性を持っています。

- キュー共用グループ内のキュー・マネージャーがアクセスできる共用キューではない。  
キュー共用グループが使用可能なのは、WebSphere MQ for z/OS のみです。
- 非持続メッセージだけを保持する。
- リカバリー不能である。
- キュー・マネージャーの始動時に削除される。
- キューを作成した MQOPEN 呼び出しを発行したアプリケーションがキューをクローズしたときか、そのアプリケーションが終了したときに、削除される。
  - コミットされたメッセージがキューにある場合には削除される。
  - この時に、キューに対して未解決で、コミットされない MQGET、MQPUT、または MQPUT1 呼び出しがある場合は、キューは論理的に削除されたものとしてマーク付けされる。さらに、このキューは、(これらの呼び出しがコミットされたあとに) クローズ処理の一部として、またはアプリケーションが終了した時に、物理的に削除される。
  - この時 (作成中のアプリケーションまたは他のアプリケーションが) キューを使用している場合、そのキューは物理的に削除されたものとしてマーク付けされ、それを使用する最後のアプリケーションによってクローズされたときに、物理的に削除される。
  - 論理的に削除されたキューに (クローズ以外の目的で) アクセスしようとすると、MQRC\_Q\_DELETED の理由コードで失敗する。
  - MQCO\_NONE、MQCO\_DELETE、および MQCO\_DELETE\_PURGE は、キューを作成した MQOPEN 呼び出しに対応する MQCLOSE 呼び出しで指定すると、すべて MQCO\_NONE として処理される。

### 永続動的キューの特性

永続動的キューは、次のような特性を持っています。

- 持続メッセージまたは非持続メッセージを保持する。
- システム障害が生じた場合にリカバリー可能である。

- アプリケーション (必ずしも、キューを作成した MQOPEN 呼び出しを発行したアプリケーションでなくてもよい) が、MQCO\_DELETE または MQCO\_DELETE\_PURGE オプションを使用してキューを正常にクローズしたときに、削除される。
  - MQCO\_DELETE オプションを指定したクローズ要求は、キュー上にメッセージ (コミットされた、またはコミットされない) がまだ残っている場合は、失敗する。MQCO\_DELETE\_PURGE オプションを指定したクローズ要求は、コミットされたメッセージがキュー上にある場合でも、正常に実行される (メッセージはクローズの一部として削除される)。ただし、キューに対して未解決で、コミットされない MQGET、MQPUT、または MQPUT1 呼び出しがある場合は、失敗する。
  - 削除要求が成功しても (作成中の、または他のアプリケーションによって)、キューが使用中の場合は、そのキューは、論理的に削除されたものとしてマーク付けされ、それを使用する最後のアプリケーションによってクローズされた時に、物理的に削除される。
- キューを削除する権限のないアプリケーションがキューをクローズした場合は、キューをクローズしたアプリケーションがキューを作成した MQOPEN 呼び出しを発行しない限り、キューは削除されません。許可検査は、対応する MQOPEN 呼び出しの妥当性検査に使用されたユーザー ID (MQOO\_ALTERNATE\_USER\_AUTHORITY が指定されていた場合は、代替ユーザー ID) に対して行われる。
- 通常のキューと同様に削除できる。

## 動的キューの使用方法

次のような場合に動的キューを使用できます。

- 処理を終了した後、キューを保持する必要がないアプリケーション
- メッセージに対する応答を別のアプリケーションに処理させるアプリケーション。このようなアプリケーションは、モデル・キューをオープンして応答先キューを動的に作成できます。例えば、クライアント・アプリケーションは次のようにすることができます。
  1. 動的キューを作成する。
  2. 要求メッセージのメッセージ記述子構造体の *ReplyToQ* フィールドにその名前を指定する。
  3. サーバーによって処理されるキューにその要求を入れる。

これで、サーバーは応答メッセージを応答先キューに入れることができるようになります。最後に、クライアントはその応答を処理し、削除オプションを使用して応答先キューをクローズできます。

## 動的キューを使用する際の考慮事項

動的キューを使用するときは次の点を考慮してください。

- クライアント/サーバー・モデルでは、各クライアントが各自の動的応答先キューを作成して使用しなければならない。動的応答先キューが複数のクライアントで共用されている場合は、そのキューに対して未解決でコミットされない活動があったり、そのキューが他のクライアントによって使用されていたりして、応答先キューの削除が遅れることがあります。さらに、キューは論理的に削除されたというマークが付けられたために、後続の (MQCLOSE 以外の) API 要求でアクセス不能になることもあります。
- 使用しているアプリケーションの環境により、動的キューをアプリケーション間で共用しなければならない場合は、必ず、そのキューに対するすべての活動がコミットされてから (削除オプションで)、そのキューがクローズされるようにする。これは、最後のユーザーが行う必要があります。これによって、キューの削除が遅れないようになり、論理的に削除されたものとしてマーク付けされたためにキューがアクセス不能になっている期間が最小限になります。

## モデル・キュー

モデル・キューはキュー定義のテンプレートで、動的キューを作成する際に使用します。

WebSphere MQ プログラムからローカル・キューを動的に作成し、キュー属性のテンプレートとして使用したいモデル・キューを命名できます。この時点で、新規のキューのいくつかの属性を変更することができます。ただし、*DefinitionType* を変更することはできません。例えば、永続キューが必要な場合には、定義タイプが永続的に設定されたモデル・キューを選択してください。ある会話型のアプリケーションで

は、動的キューを使用して照会に対する応答を入れておくことができます。なぜなら、多くの場合、応答を処理し終わったらそれらのキューを保持する必要がなくなるからです。

MQOPEN 呼び出しのオブジェクト記述子 (MQOD) に、モデル・キューの名前を指定します。キュー・マネージャーは、指定されたモデル・キューの属性を使用して、ローカル・キューを動的に作成します。

動的キューの名前を (完全名で) 指定するか、名前の語幹 (例えば、ABC) を指定して、キュー・マネージャーが固有の部分にこれを追加するようにできます。または、キュー・マネージャーがユーザーの代わりに完全な固有の名を割り当てるようにすることもできます。キュー・マネージャーが名前を割り当てる場合、それを MQOD 構造体に書き込みます。

モデル・キューに直接 MQPUT1 呼び出しを発行することはできませんが、モデル・キューをオープンしたときに作成された動的キューに対して MQPUT1 を発行することは可能です。

モデル・キューに対して MQSET や MQINQ を発行することはできません。MQOO\_INQUIRE か MQOO\_SET を使用してモデル・キューを開くと、それ以降の MQINQ 呼び出しや MQSET 呼び出しは、動的に作成されたキューに対して行われます。

モデル・キューの属性は、ローカル・キューの属性のサブセットです。詳細については、[キューの属性](#)を参照してください。

### キューの定義

IBM WebSphere MQ にキューを定義するには、MQSC コマンドの DEFINE または PCF Create Queue コマンドを使います。

これらのコマンドは、キューのタイプおよびキューの属性を指定します。例えば、ローカル・キュー・オブジェクトは、アプリケーションがそのキューを MQI 呼び出しで参照したときに何が発生するかを指定する属性を持っています。属性には、次のものがあります。

- アプリケーションがメッセージをキューから取り出せるかどうか (読み取り (GET) 可能)
- アプリケーションがメッセージをキューに書き込めるかどうか (書き込み (PUT) 可能)
- キューへのアクセスが、1つのアプリケーション専用になるか、または複数のアプリケーションで共有されるか
- 同時にキューに保管できるメッセージの最大数 (キューの最大サイズ)
- キューに書き込むことのできる最大メッセージ長

キュー・オブジェクトの定義について詳しくは、[スクリプト \(MQSC\) コマンド](#)を参照してください。

### IBM WebSphere MQ によって使用されるキュー

IBM WebSphere MQ は、一部のローカル・キューを、その操作に関連する特定の目的のために使用します。

これらのキューは、IBM WebSphere MQ によって使用される前に、必ず定義する必要があります。

### 開始キュー

開始キューは、トリガー操作に使用するキューです。キュー・マネージャーは、トリガー・イベントが発生すると、開始キューにトリガー・メッセージを書き込みます。トリガー・イベントとは、キュー・マネージャーによって検出される条件の論理的組み合わせのことです。例えば、キュー上のメッセージの数が、あらかじめ定義されたキューのサイズに達したときにトリガー・イベントが生成される場合があります。このイベントが発生すると、キュー・マネージャーは指定の開始キューにトリガー・メッセージを入れることとなります。このトリガー・メッセージは、開始キューをモニターする特殊アプリケーションであるトリガー・モニターによって取り出されます。次に、トリガー・モニターは、トリガー・メッセージに指定されているアプリケーション・プログラムを開始します。

キュー・マネージャーでこのトリガー操作を使用する場合は、少なくとも1つの開始キューを、そのキュー・マネージャー用に定義する必要があります。[トリガー操作のためのオブジェクトの管理](#)、`runmqtrm`、およびトリガーによる [WebSphere MQ アプリケーションの開始](#)を参照してください。

### 伝送キュー

伝送キューは、リモート・キュー・マネージャー宛てのメッセージを一時的に保管するキューです。ローカル・キュー・マネージャーがメッセージを直接送信する各リモート・キュー・マネージャーごとに、少なくとも1つの伝送キューを定義する必要があります。これらのキューは、リモート管理にも使用されます ([ローカル・キュー・マネージャーからのリモート管理](#)を参照)。分散キューイングでの伝送キューの使用については、[WebSphere MQ 分散メッセージング技法](#)を参照してください。

各キュー・マネージャーには、1つのデフォルト伝送キューがあります。クラスター外のキュー・マネージャーがリモート・キューにメッセージを書き込む場合、デフォルトでは、デフォルト伝送キューが使用されます。宛先のキュー・マネージャーと同名の伝送キューがあれば、その伝送キューにメッセージが入れられます。キュー・マネージャーの別名定義で、**RQMNAME** パラメーターが宛先キュー・マネージャーと一致し、**XMITQ** パラメーターが指定されている場合は、**XMITQ** により指定された伝送キューにメッセージが入れられます。**XMITQ** パラメーターがない場合は、メッセージで指定されたローカル・キューにメッセージが入れられます。

### クラスター伝送キュー

クラスター内のそれぞれのキュー・マネージャーには、**SYSTEM.CLUSTER.TRANSMIT.QUEUE** というクラスター伝送キューと、**SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE** というモデル・クラスター伝送キューがあります。これらのキューの定義は、キュー・マネージャーを定義するときデフォルトとして作成されます。キュー・マネージャーの属性、**DEFCLXQ** が **CHANNEL** に設定された場合、作成された各クラスター送信側チャンネルに、永続動的クラスター伝送キューが自動的に作成されます。キューは **SYSTEM.CLUSTER.TRANSMIT.ChannelName** と呼ばれます。クラスター伝送キューを手動で定義することもできます。

クラスター内のキュー・マネージャーは、同じクラスター内の他のキュー・マネージャーに、これらのキュー上のメッセージを送ります。

名前の解決時には、クラスター伝送キューがデフォルト伝送キューより優先され、特定のクラスター伝送キューが **SYSTEM.CLUSTER.TRANSMIT.QUEUE** より優先されます。

### 送達不能キュー

送達不能 (未配布メッセージ) キューは、正しい宛先に渡すことができないメッセージを保管します。例えば、宛先キューが満杯である場合には、メッセージは転送されません。システムに提供された送達不能キューは、**SYSTEM.DEAD.LETTER.QUEUE** と呼ばれます。

分散キューイングでは、関係する各キュー・マネージャーごとに1つの送達不能キューを定義してください。

### コマンド・キュー

コマンド・キュー **SYSTEM.ADMIN.COMMAND.QUEUE** は、適切な許可を与えられたアプリケーションが処理対象の MQSC コマンドを送信する先のローカル・キューです。次に、これらのコマンドは、コマンド・サーバーと呼ばれる IBM WebSphere MQ コンポーネントによって取り出されます。コマンド・サーバーは、それらのコマンドを検査し、正しいものをキュー・マネージャーの処理用に渡し、該当する応答先キューに応答を戻します。

キュー・マネージャーを作成すると、各キュー・マネージャーごとにコマンド・キューが自動的に作成されます。

### 応答先キュー

あるアプリケーションが要求メッセージを送信した場合、そのメッセージを受信するアプリケーションは、送信側のアプリケーションに応答メッセージを戻すことができます。この応答メッセージは、応答先キューと呼ばれるキューに書き込まれます。このキューは、通常は送信側のアプリケーションのローカル・キューです。応答先キューの名前は、送信側のアプリケーションによってメッセージ記述子の一部として指定されます。

### イベント・キュー

観測イベントは、MQI アプリケーションとは無関係にキュー・マネージャーをモニターするときに使用できます。

観測イベントが発生すると、キュー・マネージャーはイベント・キューにイベント・メッセージを書き込みます。書き込まれたこのメッセージは、モニター・アプリケーションにより読み取られます。このアプリケーションは、イベントが問題を提示すると、管理者に通知したり何らかの矯正処置を開始します。

**注:** トリガー・イベントは観測イベントとは異なります。トリガー・イベントは、同じ条件では発生せず、イベント・メッセージが生成されません。

観測イベントの詳細については、[観測イベント](#)を参照してください。

## IBM WebSphere MQ キュー・マネージャー

キュー・マネージャー およびキュー・マネージャーがアプリケーションに提供するキューイング・サービスの概要です。

プログラムは、キュー・マネージャーのサービスを使用する前に、そのキュー・マネージャーへ接続していなければなりません。この接続は、プログラムが明示的に (MQCONN または MQCONNX 呼び出しを使用する) 行うこともできますし、暗示的に行われることもあります (これはプログラムを実行するプラットフォームおよび環境に依存します)。

キュー・マネージャーは、アプリケーションにキューイング・サービスを提供し、そのキューを管理します。キュー・マネージャーは、次の操作を保証します。

- オブジェクトの属性は、受け取ったコマンドに応じて変更されます。
- 該当する条件が満たされたときに、トリガー・イベントや観測イベントなどの特殊イベントが生成されます。
- メッセージが、MQPUT 呼び出しを行ったアプリケーションの要求により、正しいキューに書き込まれます。正しいキューに入れられなかった場合には、アプリケーションに通知され、該当する理由コードが戻されます。

それぞれのキューは、1つのキュー・マネージャーに属しており、そのキュー・マネージャーに対してローカル・キューであるといえます。アプリケーションが接続されているキュー・マネージャーは、そのアプリケーションに対してローカル・キュー・マネージャーであるといえます。アプリケーションのローカル・キュー・マネージャーに属しているキューは、そのアプリケーションのためのローカル・キューです。

リモート・キューとは、別のキュー・マネージャーに属しているキューのことです。リモート・キュー・マネージャーとは、ローカル・キュー・マネージャー以外の任意のキュー・マネージャーのことです。リモート・キュー・マネージャーは、ネットワーク内のリモート・マシン上にある場合と、ローカル・キュー・マネージャーと同じマシン上にある場合があります。WebSphere MQ では、1つのマシン上で複数のキュー・マネージャーを使用することができます。

キュー・マネージャー・オブジェクトは、一部の MQI 呼び出しで使用することができます。例えば、キュー・マネージャー・オブジェクトの属性について、MQI 呼び出しの MQINQ を使用して問い合わせることができます。

### キュー・マネージャーの属性

各キュー・マネージャーに関連付けられているのは、その特性を定義する一連の属性 (または特性) です。キュー・マネージャーの属性のいくつかは、作成される時に固定されます。他のものについては、WebSphere MQ コマンドを使用して変更できます。すべての属性の値は、Secure Sockets Layer (SSL) 暗号化で使用されるものを除いて、MQINQ 呼び出しを使って問い合わせることができます。

固定されている属性は、次のとおりです。

- キュー・マネージャーの名前
- キュー・マネージャーが稼働するプラットフォーム (例えば、Windows)
- キュー・マネージャーがサポートするシステム制御コマンドのレベル
- キュー・マネージャーが処理するメッセージに割り当てられる最高優先順位
- プログラムが WebSphere MQ コマンドを送信できるキューの名前
- キュー・マネージャーが処理できるメッセージの最大長
- プログラムがメッセージを書き込んだり読み取ったりするときに、キュー・マネージャーが同期点機能をサポートするかどうか

変更可能な属性は、次のとおりです。

- キュー・マネージャーのテキスト記述子
- キュー・マネージャーが MQI 呼び出しを処理するときに文字ストリングに使用する文字セットの ID
- トリガー・メッセージの数を制限するためにキュー・マネージャーが使用する時間間隔
- キュー・マネージャーの送達不能 (未配布メッセージ) キュー

- キュー・マネージャーのデフォルトの伝送キューの名前
- 任意の接続のためのオープン・ハンドルの最大数
- イベント報告の各種カテゴリーの設定または解除
- 1つの作業単位内のコミットされていないメッセージの最大数

## キュー・マネージャーとワークロード管理

1つのキューに対して複数の定義が作成されている場合は、いくつかのキュー・マネージャーからなる1つのクラスターを設定することができます(例えば、クラスター内のキュー・マネージャーは、それぞれが他のキュー・マネージャーの複製として機能できます)。特定のキューに対するメッセージは、そのキューのインスタンスのホストになっているすべてのキュー・マネージャーで処理できます。ワークロード管理アルゴリズムは、どのキュー・マネージャーがメッセージを処理するか決定し、キュー・マネージャー間でワークロードを分散します。詳細については、[クラスター・ワークロード管理アルゴリズム](#)を参照してください。

## プロセス定義

プロセス定義オブジェクトを使用すると、キュー・マネージャーが使用するアプリケーションの属性を定義することによって、オペレーターによる介入がなくてもアプリケーションを開始することができます。

プロセス定義オブジェクトは、IBM WebSphere MQ キュー・マネージャー上のトリガー・イベントに応答して開始されるアプリケーションを定義します。プロセス定義の属性には、アプリケーション ID、アプリケーション・タイプ、およびアプリケーション特有のデータがあります。詳細については、[114 ページの『IBM WebSphere MQ によって使用されるキュー』](#)の「開始キュー」の項を参照してください。

アプリケーションをオペレーターの手動介入なしで開始させる(これについては、[トリガーによる WebSphere MQ アプリケーションの開始](#)を参照)には、アプリケーションの属性をキュー・マネージャーに通知する必要があります。これらの属性はプロセス定義オブジェクトで定義されます。

`ProcessName` 属性は、オブジェクトの作成時に固定されます。その他の属性は、IBM WebSphere MQ コマンドを使用して変更できます。

すべての属性の値について問い合わせるには、[MQINQ - オブジェクト属性の照会](#)を使用します。

プロセス定義の属性の詳細な説明については、[プロセス定義の属性](#)を参照してください。

## 名前リスト

名前リストは、クラスター名、キュー名、または認証情報オブジェクト名のリストが含まれた WebSphere MQ オブジェクトです。クラスターでは、このリストを使用して、キュー・マネージャーがリポジトリを保持しているクラスターのリストを識別することができます。

名前リストは、WebSphere MQ オブジェクトの1つで、他の WebSphere MQ オブジェクトのリストが格納されています。通常、名前リストはトリガー・モニターなどのアプリケーションによって、キュー・グループの識別に使用されます。名前リストを使用した場合の利点は、アプリケーションとは別個に管理できるということです。更新時に、その名前リストを使用しているアプリケーションを停止する必要はありません。また、あるアプリケーションで障害が起こった場合でも、名前リストには影響はなく、他のアプリケーションは引き続きその名前リストを使用できます。

名前リストを、キュー・マネージャーのクラスター用に使用して、複数の WebSphere MQ オブジェクトが参照するクラスターのリストを管理することもできます。

名前リストの定義や変更を行うには、MQSC コマンドを使用します。

プログラムは MQI を使用することにより、これらの名前リストにどのキューが登録されているかを検知できます。名前リストの編成は、アプリケーション設計担当者およびシステム管理者が行います。

名前リストの属性の詳細な説明については、[名前リストの属性](#)を参照してください。

## 認証情報オブジェクト

キュー・マネージャー認証情報オブジェクトの概要と詳細情報へのリンクです。

キュー・マネージャー認証情報オブジェクトは、Secure Sockets Layer (SSL) および Transport Layer Security (TLS) の WebSphere MQ サポートの一部を構成しています。このオブジェクトには、取り消された証明書の検査に必要な定義があります。認証局は、信頼できなくなった証明書を取り消します。

このセクションでは、認証情報オブジェクトに関連して、`setmqaut`、`dspmqaout`、`dmpmqaut`、`rcrimqobj`、`rcdmqimg`、および `dspmqls` の各コマンドの使用法について説明します。SSL と TLS の概要、および認証情報オブジェクトの使用については、[WebSphere MQ support for SSL and TLS](#) を参照してください。

SSL および TLS について詳しくは、[Secure Sockets Layer \(SSL\) および Transport Layer Security \(TLS\) の概念](#) を参照してください。

認証情報オブジェクトは、証明書取り消し検査を実行するために必要な定義を提供します。

認証情報オブジェクトの属性の詳細な情報については、[認証情報オブジェクト](#) を参照してください。

## 通信情報オブジェクト

IBM WebSphere MQ Multicast は、待ち時間が短く、ファンアウトが大きい、高信頼性マルチキャスト・メッセージングです。マルチキャスト送信を使用するには、通信情報 (COMMINFO) オブジェクトが必要です。

COMMINFO オブジェクトは、マルチキャスト伝送に関連付けられた属性が含まれる IBM WebSphere MQ オブジェクトです。これらの属性について詳しくは、[DEFINE COMMINFO](#) を参照してください。COMMINFO オブジェクトの作成について詳しくは、[マルチキャストの概要](#) を参照してください。

### 関連概念

[129 ページの『IBM WebSphere MQ マルチキャスト』](#)

IBM WebSphere MQ Multicast は、待ち時間が短く、ファンアウトが大きい、高信頼性マルチキャスト・メッセージングです。

## チャネル

チャネルとは、分散キュー・マネージャーによって使用される通信リンクです。

チャネルとは、あるキュー・マネージャーと別のキュー・マネージャーを結ぶ通信パスを提供するオブジェクトのことです。あるキュー・マネージャーから別のキュー・マネージャーにメッセージを移すために、分散キューイングでチャネルが使用され、それによってアプリケーションは基礎をなす通信プロトコルから遮蔽されます。キュー・マネージャーは、同一のプラットフォーム上に存在する場合もあれば、異なるプラットフォーム上に存在する場合もあります。

キュー・マネージャーが相互に通信できるようにするには、メッセージを送信するキュー・マネージャーに 1 つのチャネル・オブジェクトを定義し、メッセージを受信するキュー・マネージャーに別の補完的なチャネル・オブジェクトを定義する必要があります。

WebSphere MQ のチャネルには、次の 2 つのカテゴリがあります。

- **メッセージ・チャネル。** 単一方向のもので、あるキュー・マネージャーから別のキュー・マネージャーへメッセージを転送します。詳しくは、[チャネル出口呼び出しおよびデータ構造体](#) を参照してください。
- **MQI チャネル。** 双方向のもので、WebSphere MQ MQI クライアントからキュー・マネージャーに MQI 呼び出しを転送し、キュー・マネージャーから WebSphere MQ クライアントに応答を転送します。詳しくは、[119 ページの『チャネルについて』](#) を参照してください。

### 関連概念

[50 ページの『相互通信の概念』](#)

WebSphere MQ において、相互通信とは、あるキュー・マネージャーから別のキュー・マネージャーへメッセージが送信されることを意味します。受信キュー・マネージャーは、同じマシン上に置くことも別のマシン上に置くこともでき、近くにあっても地球の裏側にあってもメッセージを受信できます。このプログラムは、ローカル・キュー・マネージャーと同じプラットフォーム、または WebSphere MQ がサポートするいずれかのプラットフォームで実行することができます。これを分散環境と呼びます。WebSphere MQ は、このような分散環境での通信を分散キューイング管理 (DQM) によって処理します。

### 関連タスク

[リモート WebSphere MQ オブジェクトの管理](#)

## 関連資料

[チャンネル出口呼び出しおよびデータ構造体](#)

[121 ページの『通信』](#)

WebSphere MQ MQI クライアントは、サーバーとの通信に MQI チャンネルを使用します。

チャンネルについて

チャンネルは、WebSphere MQ MQI クライアントと WebSphere MQ サーバーとの、あるいは 2 つの WebSphere MQ サーバー間の論理通信リンクです。

1 つのチャンネルには、2 つの定義があり、それらは接続の両端に 1 つずつあります。接続の両端には、同じチャンネル名を使用する必要があり、使用するチャンネル・タイプには互換性がなければなりません。

WebSphere MQ のチャンネルは、次に示すように 2 つのカテゴリーに分かれます (これらのカテゴリー内に異なるチャンネル・タイプがあります)。

## 関連概念

[119 ページの『メッセージ・チャンネル』](#)

メッセージ・チャンネルは、単一方向のリンクです。これは、メッセージ・チャンネル・エージェント (MCA) を使用して 2 つのキュー・マネージャーを接続します。

[119 ページの『MQI チャンネル』](#)

MQI チャンネルは、WebSphere MQ MQI クライアントをサーバー・マシン上のキュー・マネージャーに接続し、WebSphere MQ MQI クライアント・アプリケーションから MQCONN または MQCONNX 呼び出しを発行すると確立されます。

[120 ページの『チャンネルの停止』](#)

WebSphere MQ では、サーバー接続のチャンネルに STOP CHANNEL コマンドを発行すると、クライアント接続のチャンネルを停止するために使用するメソッドを選択できます。

メッセージ・チャンネル

メッセージ・チャンネルは、単一方向のリンクです。これは、メッセージ・チャンネル・エージェント (MCA) を使用して 2 つのキュー・マネージャーを接続します。

メッセージ・チャンネルの目的は、1 つのキュー・マネージャーから別のキュー・マネージャーにメッセージを移すことです。メッセージ・チャンネルは、クライアント・サーバー環境で必須ではありません。

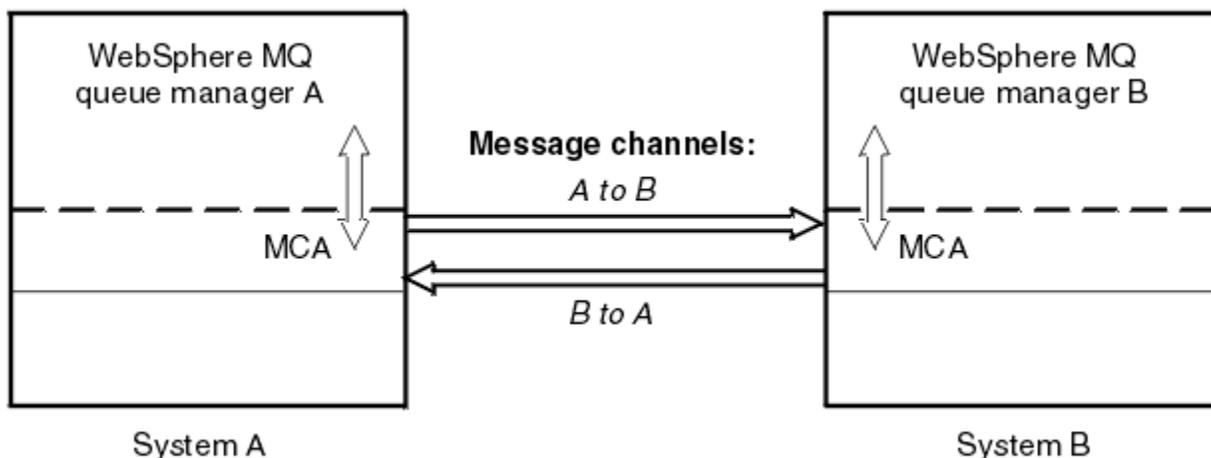


図 28. 2 つのキュー・マネージャー間のメッセージ・チャンネル

MQI チャンネル

MQI チャンネルは、WebSphere MQ MQI クライアントをサーバー・マシン上のキュー・マネージャーに接続し、WebSphere MQ MQI クライアント・アプリケーションから MQCONN または MQCONNX 呼び出しを発行すると確立されます。

これは両方向のリンクであり、メッセージ・データが入った MQPUT 呼び出しや、メッセージ・データが戻される結果となる MQGET 呼び出しなど、MQI 呼び出しおよび応答だけを転送するのに使用されます。チャンネル定義の作成と使用方法は 2 つあります ([MQI チャンネルの定義](#)を参照してください)。

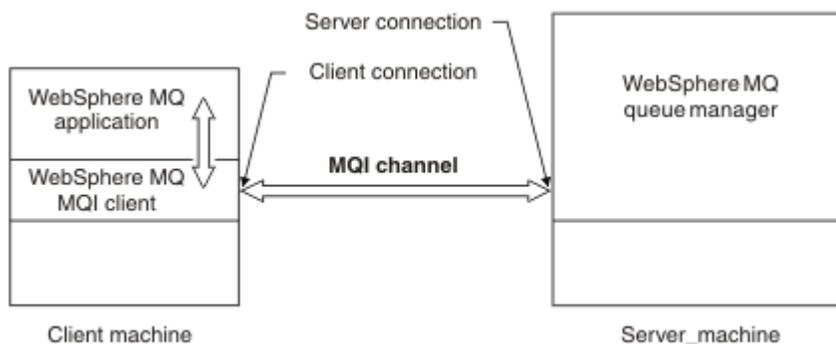


図 29. MQI チャンネル上のクライアント接続およびサーバー接続

MQI チャンネルを使用して、単一のキュー・マネージャー、あるいはキュー共有グループの一部となっているキュー・マネージャーにクライアントを接続することができます (キュー共有グループへのクライアントの接続を参照)。

MQI チャンネル定義のチャンネル・タイプには 2 種類あります。これらのチャンネル・タイプでは、両方向 MQI チャンネルを定義します。

#### クライアント接続チャンネル

このタイプは、WebSphere MQ MQI クライアント用です。

#### サーバー接続チャンネル

このタイプは、キュー・マネージャーを稼働するサーバー用です。WebSphere MQ MQI クライアント環境で実行される WebSphere MQ アプリケーションは、このキュー・マネージャーと通信します。

#### チャンネルの停止

WebSphere MQ では、サーバー接続のチャンネルに STOP CHANNEL コマンドを発行すると、クライアント接続のチャンネルを停止するために使用するメソッドを選択できます。

つまり、MQGET Wait 呼び出しを発行するクライアント・チャンネルを制御でき、チャンネルを停止する方法とタイミングを決定できます。

STOP CHANNEL コマンドは以下の 3 つのモードで発行できます。これらは、チャンネルを停止する方法を示しています。

#### 静止

現在のメッセージが処理されてからチャンネルを停止します。

共有会話が有効になっていると、WebSphere MQ MQI クライアントはその停止要求をタイミングよく認識します。このタイミングは、ネットワークのスピードに依存します。その後の WebSphere MQ への呼び出し発行の結果により、クライアント・アプリケーションはその停止要求を認識します。

#### 強制

即時にチャンネルを停止します。

#### 終了

即時にチャンネルを停止します。チャンネルがプロセスとして実行されている場合は、チャンネルのプロセスが終了します。スレッドとして実行されている場合は、スレッドが終了します。

これは段階的なプロセスです。終了モードが使用される場合、まず静止モード、次に強制モードでサーバー接続チャンネルの停止が試行され、さらに必要であれば、終了モードが使用されます。終了のさまざまな段階で、クライアントがさまざまな戻りコードを受け取る場合があります。プロセスまたはスレッドが終了されると、クライアントは通信エラーを受け取ります。

アプリケーションに戻される戻りコードは、発行される MQI 呼び出し、および発行される STOP CHANNEL コマンドによって異なります。クライアントは、MQRC\_CONNECTION\_QUIESCING または MQRC\_CONNECTION\_BROKEN のどちらかの戻りコードを受け取ります。

MQRC\_CONNECTION\_QUIESCING を検出したクライアントは、現在のトランザクションを完了させ、終了させようと試みます。これは MQRC\_CONNECTION\_BROKEN ではできません。素早くトランザクションを完了、終了できなかったクライアントは、数秒後に CONNECTION\_BROKEN を受信します。

MODE(FORCE) または MODE(TERMINATE) を使った STOP CHANNEL コマンドは、MODE(QUIESCE) を使った場合よりも、CONNECTION\_BROKEN になる可能性が高くなります。

## 通信

WebSphere MQ MQI クライアントは、サーバーとの通信に MQI チャンネルを使用します。

チャンネル定義は、WebSphere MQ MQI クライアントとサーバーの接続の両端で作成する必要があります。チャンネル定義の作成方法については、[MQI チャンネルの定義](#)で説明されています。

次のテーブルに使用可能な伝送プロトコルを示します。

| クライアント・プラットフォーム     | LU 6.2          | TCP/IP | NetBIOS | SPX |
|---------------------|-----------------|--------|---------|-----|
| UNIX and Linux システム | はい <sup>1</sup> | Yes    |         |     |
| Windows             | Yes             | Yes    | Yes     | Yes |

注:

- LU6.2 は、Linux (POWER® プラットフォーム)、Linux (x86-64 プラットフォーム)、Linux (zSeries s390x プラットフォーム)、または Solaris (x86-64 プラットフォーム) ではサポートされません。

伝送プロトコル - WebSphere MQ MQI クライアントとサーバーのプラットフォームの組み合わせに、これらの伝送プロトコルを使用して実現できる WebSphere MQ MQI クライアントおよびサーバーのプラットフォームの組み合わせが示されています。

WebSphere MQ MQI クライアント上の WebSphere MQ アプリケーションは、キュー・マネージャーがローカルの場合と同様に、すべての MQI 呼び出しを使用できます。MQCONN または MQCONNX は、選択したキュー・マネージャーに WebSphere MQ アプリケーションを関連付け、接続ハンドルを作成します。接続ハンドルを使用するその他の呼び出しは、その後、接続されたキュー・マネージャーに処理されます。WebSphere MQ MQI クライアント通信は、クライアントとサーバーの間にアクティブな接続を必要としません。対照的に、キュー・マネージャー間の通信は、接続にも時間にも依存しません。

伝送プロトコルはチャンネル定義を使用することで指定され、アプリケーションに影響しません。例えば、Windows アプリケーションは、TCP/IP を介してあるキュー・マネージャーと接続し、NetBIOS を介して別のキュー・マネージャーと接続することができます。

## パフォーマンス上の考慮事項

使用する伝送プロトコルが、WebSphere MQ クライアント/サーバー・システムのパフォーマンスに影響することがあります。低速電話回線によるダイヤルアップ・サポートの場合は、WebSphere MQ チャンネル圧縮を使用するとよい場合があります。

## クライアント接続チャンネル

クライアント接続チャンネル・オブジェクトの紹介と、詳細情報へのリンク。

クライアント接続チャンネルとは、WebSphere MQ MQI クライアントからキュー・マネージャーへの通信パスを提供するオブジェクトのことです。クライアント接続チャンネルは、キュー・マネージャーとクライアントとの間でメッセージを移動するために、分散キューイングで使用されます。チャンネルは、基礎をなす通信プロトコルからアプリケーションを保護します。クライアントは、キュー・マネージャーと同一のプラットフォーム上に存在する場合もあれば、異なるプラットフォーム上に存在する場合もあります。

クライアント接続チャンネルとその用法については、36 ページの『相互通信』を参照してください。

## リスナー

リスナーは、他のキュー・マネージャーまたはクライアント・アプリケーションからネットワーク要求を受け取るプロセスで、関連付けられたチャンネルを始動します。

リスナーは、他のキュー・マネージャーまたはクライアント・アプリケーションからネットワーク要求を受け取るプロセスで、関連付けられたチャンネルを始動します。リスナー・プロセスは、`runmqclsr` 制御コマンドを使用して開始できます。

リスナー・オブジェクトは、キュー・マネージャーの有効範囲内からリスナー・プロセスの開始と停止を管理することができるようにする、WebSphere MQ オブジェクトです。リスナー・オブジェクトの属性を定義することにより、次の操作を実行できます。

- リスナー・プロセスを構成する。
- キュー・マネージャーが開始および停止したときに、リスナー・プロセスも自動的に開始および停止させるかどうかを指定する。

リスナー・オブジェクトは、**WebSphere MQ for z/OS** ではサポートされていません。

## サービス

サービス・オブジェクトは、キュー・マネージャーが開始または停止したときに実行するプログラムを定義するための方法です。

プログラムは次のいずれかのタイプになります。

### サーバー

サーバーは、SERVTYPE パラメーターが SERVER に指定されているサービス・オブジェクトです。サーバー・サービス・オブジェクトは、指定したキュー・マネージャーの開始時に実行されるプログラムの定義です。サーバー・プロセスの1つのインスタンスだけを並行して実行できます。実行中は、サーバー・プロセスの状況を MQSC コマンド `DISPLAY SVSTATUS` を使用してモニターできます。通常、サーバー・サービス・オブジェクトは、送達不能ハンドラーまたはトリガー・モニターなどのプログラムの定義ですが、実行可能なプログラムは WebSphere MQ によって提供されるプログラムに限定されません。また、サーバー・サービス・オブジェクトを定義して、指定されたキュー・マネージャーがシャットダウンしてプログラムが終了したときに実行されるコマンドを含めることができます。

### コマンド

コマンドは、SERVTYPE パラメーターが COMMAND に指定されているサービス・オブジェクトです。コマンド・サービス・オブジェクトは、指定されたキュー・マネージャーが開始または停止したときに実行されるプログラムの定義です。コマンド・プロセスの複数のインスタンスを並行して実行できます。コマンド・サービス・オブジェクトは、プログラムが実行されるとキュー・マネージャーがプログラムをモニターしなくなる点で、サーバー・サービス・オブジェクトと異なっています。通常、コマンド・サービス・オブジェクトは存続期間の短いプログラムの定義で、1つ以上の他のタスクを開始するなどの特定のタスクを実行します。

## 関連概念

### [サービスの取り扱い](#)

## トピック・オブジェクト

トピック・オブジェクトは、特定の非デフォルト属性をトピックに割り当てることができるようにする WebSphere MQ オブジェクトです。

トピックは、特定のトピック・ストリングにパブリッシュまたはサブスクライブするアプリケーションによって定義されます。トピック・ストリングでは、スラッシュ記号 (/) でトピックを分離することにより、トピックの階層を指定できます。階層はトピック・ツリーで視覚化されます。例えば、アプリケーションがトピック・ストリング `/Sport/American Football` および `/Sport/Soccer` にパブリッシュすると、`American Football` と `Soccer` の2つの子を持つ親ノード `Sport` を持つトピック・ツリーが作成されます。

各トピックは、トピック・ツリー内にある最初の親管理ノードから属性を継承します。特定のトピック・ツリーに管理トピック・ノードが存在しない場合、全トピックは基本トピック・オブジェクトである `SYSTEM.BASE.TOPIC` から属性を継承します。

トピック・ツリー内の任意のノードでトピック・オブジェクトを作成できます。これは、トピック・オブジェクトの `TOPICSTR` 属性でそのノードのトピック・ストリングを指定することにより可能となります。また、管理トピック・ノードの他の属性も定義することができます。これらの属性について詳しくは、[MQSC コマンド](#)、または [管理タスクの自動化](#) を参照してください。デフォルトでは、各トピック・オブジェクトは直近の親管理トピック・ノードから属性を継承します。

さらに、トピック・オブジェクトを使用することにより、アプリケーション開発者からトピック・ツリー全体を隠すことも可能です。トピック /Sport/American Football に対して FOOTBALL.US という名前のトピック・オブジェクトが作成されている場合、アプリケーションにおいて、ストリング /Sport/American Football の代わりに FOOTBALL.US という名前のオブジェクトにパブリッシュまたはサブスクライブすることで、同じ結果を得ることができます。

トピック・オブジェクトのトピック・ストリング中に文字 #、+、/、または \* を入力する場合、その文字はストリング中では通常文字として扱われ、トピック・オブジェクトが関連付けられているトピック・ストリングの一部であると見なされます。

トピック・オブジェクトについて詳しくは、[Introduction to WebSphere MQ publish/subscribe messaging](#) を参照してください。

## IBM WebSphere MQ オブジェクトの命名

WebSphere MQ オブジェクトに適用される命名規則は、オブジェクトごとに異なります。また、IBM WebSphere MQ で使用するマシンの名前およびユーザー ID は、いくつかの命名規則に従います。

キュー・マネージャーの各インスタンスは、その名前で見分けられます。この名前は、相互接続されたキュー・マネージャーのネットワーク内で固有である必要があります。固有になっていると、あるキュー・マネージャーは、所定のメッセージを送るべきターゲットのキュー・マネージャーを明確に識別することができます。

他のタイプのオブジェクトの場合、各オブジェクトにはそれぞれ関連付けられた名前があり、各オブジェクトはその名前で参照できます。これらの名前は、1つのキュー・マネージャーおよびオブジェクト・タイプ内において固有のものである必要があります。例えば、同じ名前のキューとプロセスを持つことはできますが、同じ名前の2つのキューを持つことはできません。

WebSphere MQ では、名前には最大 48 文字まで使用することができます。ただし、チャンネルは例外で、最大 20 文字まで使用できます。IBM WebSphere MQ オブジェクトの命名規則の詳細については、[123 ページの『IBM WebSphere MQ オブジェクトの命名規則』](#) を参照してください。

IBM WebSphere MQ で使用するマシンの名前およびユーザー ID は、以下のいくつかの命名規則にも従います。

- マシン名にスペースが含まれていないことを確認します。IBM WebSphere MQ スペースを含むマシン名はサポートされません。名前にスペースが含まれているマシンに IBM WebSphere MQ をインストールした場合は、キュー・マネージャーを作成できなくなります。
- IBM WebSphere MQ 権限のためのユーザー ID およびグループの名前は、20 文字以内にする必要があります (スペースは使用できません)。
- WebSphere MQ for Windows サーバーでは、Windows クライアントが @ 文字を含むユーザー ID (例: abc@d) の下で実行されている場合、そのクライアントの接続はサポートされません。

### 関連概念

[126 ページの『IBM WebSphere MQ のファイル名についての理解』](#)

WebSphere MQ のキュー・マネージャー、キュー、プロセス定義、名前リスト、チャンネル、クライアント接続チャンネル、リスナー、サービス、および認証情報オブジェクトは、それぞれファイルで表されます。これらのオブジェクト名は必ずしも有効なファイル名ではないので、キュー・マネージャーは、必要に応じてそのオブジェクト名を有効なファイル名に変換します。

### 関連資料

[123 ページの『IBM WebSphere MQ オブジェクトの命名規則』](#)

IBM WebSphere MQ オブジェクト名には最大長があり、大/小文字が区別されます。すべてのオブジェクト・タイプにすべての文字がサポートされているわけではなく、多くのオブジェクトには名前の固有性に関する規則があります。

## IBM WebSphere MQ オブジェクトの命名規則

IBM WebSphere MQ オブジェクト名には最大長があり、大/小文字が区別されます。すべてのオブジェクト・タイプにすべての文字がサポートされているわけではなく、多くのオブジェクトには名前の固有性に関する規則があります。

IBM WebSphere MQ オブジェクトには多種多様なタイプがあり、オブジェクトはタイプごとに別個のオブジェクト名前空間に存在するので、タイプの異なる複数オブジェクトが同じ名前を持つことができます。例えば、ローカル・キューと送信側チャンネルとが同じ名前であってもかまいません。しかし、あるオブジェクトが同じ名前空間にある別のオブジェクトと同じ名前を持つことはできません。例えば、ローカル・キューはモデル・キューと同じ名前を持つことができず、送信側チャンネルは受信側チャンネルと同じ名前を持つことができません。

以下の IBM WebSphere MQ オブジェクトは、別個のオブジェクト名前空間に存在します。

- 認証情報
- チャンネル
- クライアント・チャンネル
- リスナー
- 名前リスト
- プロセス
- キュー
- サービス
- ストレージ・クラス
- サブスクリプション
- トピック

## オブジェクト名の文字長

一般に、IBM WebSphere MQ オブジェクト名の最大長は 48 文字です。この規則は、次のオブジェクトに当てはまります。

- 認証情報
- クラスタ
- リスナー
- 名前リスト
- プロセス定義
- キュー
- キュー・マネージャー
- サービス
- サブスクリプション
- トピック

以下の制限があります。

1. チャンネル・オブジェクト名とクライアント接続チャンネル名の最大長は 20 文字です。チャンネルについて詳しくは、[チャンネルの定義](#)を参照してください。
2. トピック・ストリングに使用できるのは、最大 10240 バイトです。すべての IBM WebSphere MQ オブジェクト名は大/小文字を区別します。
3. ストレージ・クラス名の最大長は 8 文字です。
4. CF 構造名の最大長は 12 文字です。

## オブジェクト名の文字

IBM WebSphere MQ オブジェクト名に有効な文字は、次のとおりです。

| 文字          | 制限対象機能                                                                                                                                                                                    |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 大文字の A から Z | • なし                                                                                                                                                                                      |
| 小文字の a から z | <ul style="list-style-type: none"> <li>• MQSC スクリプトでは、小文字を含む名前を単一引用符で囲む必要があります。これにより、小文字が大文字に変換されなくなります。</li> <li>• EBCDIC カタカナを使用するシステムは、オブジェクト名に小文字の a から z の文字を使用できません。</li> </ul>     |
| 数字 0 から 9   | • なし                                                                                                                                                                                      |
| ピリオド (.)    | • なし                                                                                                                                                                                      |
| 下線 (_)      | •  なし                                                                                                   |
| 順方向斜線 (/)   | <ul style="list-style-type: none"> <li>•  Windows システムでは、キュー・マネージャー名の最初の文字をスラッシュにすることはできません。</li> </ul> |
| パーセント記号 (%) | •  なし                                                                                                   |

オブジェクト名の文字に関しては、いくつかの一般的な規則もあります。

1. 先行ブランクや組み込みブランクは使用できません。
2. 各国語文字は使用不可です。
3. フィールド長全体に長さが満たない名前は、右側にブランクが埋め込まれる場合があります。キュー・マネージャーから返されるすべてのショート・ネームには、必ず右側にブランクが埋め込まれます。

## キュー名

キューの名前には次の 2 つの部分があります。

- キュー・マネージャーの名前。
- そのキュー・マネージャーによって認識されているキューのローカル名

キュー名の各部分の長さは 48 文字です。

ローカル・キューを参照する場合は、キュー・マネージャーの名前を (ブランク文字に置き換えたり、先行ヌル文字を使用したりして) 省略できます。しかし、IBM WebSphere MQ によってプログラムに返されるすべてのキュー名には、キュー・マネージャーの名前が含まれます。

リモート・キューを参照する場合、プログラムは完全なキュー名にキュー・マネージャーの名前を含めるか、またはリモート・キューのローカル定義が存在する必要があります。

アプリケーションがキュー名を使用する場合、その名前はローカル・キューの名前 (またはその別名) またはリモート・キューのローカル定義の名前いずれかにすることができますが、アプリケーションがキューからメッセージを受け取る必要がない場合 (キューがローカルでなければならない場合) は、アプリケーションがそのいずれであるかを認識する必要はありません。アプリケーションがキュー・オブジェクトをオープンすると、MQOPEN 呼び出しはネーム・レゾリューション機能を実行して、それ以降の操作をどのキューに対して実行するかを判別します。ここで大切な点は、キュー・マネージャーのネットワーク内で特定の場所に定義されている特定のキューに対する依存関係がアプリケーションに組み込まれてはいないということです。したがって、システム管理者がネットワーク内でキューを再配置し、その定義を変更しても、そのキューを使用するアプリケーションを変更する必要はありません。

## 予約オブジェクト名

"SYSTEM."で始まるオブジェクト名はキュー・マネージャーによって定義されたオブジェクト用に予約済みです。Alter、Define、および Replace コマンドを使用して、ご使用のインストール環境に合わせてこれらのオブジェクト定義を変更できます。IBM WebSphere MQ 用に定義されている名前は、キュー名にすべてリストされています。

### 関連タスク

[インストール名の選択](#)

## IBM WebSphere MQ のファイル名についての理解

WebSphere MQ のキュー・マネージャー、キュー、プロセス定義、名前リスト、チャネル、クライアント接続チャネル、リスナー、サービス、および認証情報オブジェクトは、それぞれファイルで表されます。これらのオブジェクト名は必ずしも有効なファイル名ではないので、キュー・マネージャーは、必要に応じてそのオブジェクト名を有効なファイル名に変換します。

キュー・マネージャー・ディレクトリーへのデフォルトのパスは、次のものから作られます。

- 接頭部。次のように、WebSphere MQ 構成情報に定義されています。
  - Windows 32 ビット・システムでは、デフォルトの接頭部は C:\Program Files\IBM\WebSphere MQ です。Windows 64 ビット・システムの場合、デフォルトの接頭部は C:\Program Files\IBM\WebSphere MQ (x86)\ です。これは、mq.ini 構成ファイルの DefaultPrefix スタンザで構成されます。
  - UNIX and Linux システムでは、デフォルトの接頭部は /var/mqm です。これは、mq.ini 構成ファイルの DefaultPrefix スタンザで構成されます。

使用可能な場合は、IBM WebSphere MQ エクスプローラーの WebSphere MQ プロパティ・ページを使用して接頭部を変更できます。使用できない場合は、mq.ini 構成ファイルを手動で編集します。

- キュー・マネージャー名は、有効なディレクトリー名に変換されます。例えば、次のとおりです。

```
queue.manager
```

このキュー・マネージャーは、次のように表されます。

```
queue!manager
```

この処理を、名前変換と呼びます。

WebSphere MQ では、キュー・マネージャーに 48 文字までの名前を付けることができます。

例えば、キュー・マネージャーに次の名前を付けることができます。

```
QUEUE.MANAGER.ACCOUNTING.SERVICES
```

しかし、各キュー・マネージャーはファイルで表されるため、ファイル名の最大長および名前に使用できる文字に制限があります。そのため、オブジェクトを表すファイルの名前は、ファイル・システムの要件に合うように自動的に変換されます。

キュー・マネージャー名の変換の規則を以下に示します。

1. 個々の文字が次のように変換されます。
  - 開始。先へ!
  - / が & に変換される
2. 名前が正しくない場合は次のようになります。
  - a. 8 文字に切り捨てられる。
  - b. 3 文字の数値接尾部が付加される。

例えば、デフォルト接頭部であるとする、`queue.manager` という名前のキュー・マネージャーは次のようになります。

- NTFS または FAT32 のある WebSphere MQ for Windows では、キュー・マネージャー名は次のようになります。

```
c:\Program Files\IBM\WebSphere MQ\mqgrs\queue!manager
```

- FAT を備えた WebSphere MQ for Windows では、キュー・マネージャー名は次のようになります。

```
c:\Program Files\IBM\WebSphere MQ\mqgrs\queue!ma
```

- WebSphere MQ for UNIX and Linux システムでは、キュー・マネージャー名は次のようになります。

```
/var/mqm/mqgrs/queue!manager
```

変換アルゴリズムでは、大文字小文字の区別のないファイル・システム上で、大文字小文字の違いのみの名前を区別することもできます。

## オブジェクト名の変換

オブジェクト名は、必ずしも有効なファイル・システム名になっていません。オブジェクト名を変換することが必要な場合があります。使用される変換方法は、キュー・マネージャー名の場合とは異なります。つまり、1つのマシンにつきキュー・マネージャー名はわずかしかありませんが、各キュー・マネージャーごとに相当数の他のオブジェクトが存在する可能性があるためです。キュー、プロセス定義、名前リスト、チャンネル、クライアント接続チャンネル、リスナー、サービス、および認証情報オブジェクトは、ファイル・システム内に表示されます。

変換処理で新しい名前が生成された場合、元のオブジェクト名との関係は簡単には分かりません。`dspmqls` コマンドを使用して、本当のオブジェクト名と変換されたオブジェクト名との間での変換を行うことができます。

## オブジェクトの管理

オブジェクトの作成、変更、表示、削除の方法の概要です。

詳しくは、[106 ページの『オブジェクト』](#)を参照してください。

動的キューの場合を除き、これらのオブジェクト (動的キューを除く) は、処理の前にキュー・マネージャーに定義されていなければなりません。

オブジェクトの定義と管理には、次のものを使用します。

- [プログラマブル・コマンド・フォーマット・リファレンス](#) および [管理タスクの自動化](#) で説明されている PCF コマンド
- [MQSC コマンド](#) で説明されている MQSC コマンド
- WebSphere MQ エクスプローラー (Windows、UNIX、および Linux for Intel システムのみ)

次のメソッドを使用してオブジェクトを管理することもできます。

- キーボードから入力する制御コマンド。 [制御コマンド](#) を参照してください。
- プログラムからの IBM WebSphere MQ 管理インターフェース (MQAI) の呼び出し。 [WebSphere MQ 管理インターフェース \(MQAI\)](#) を参照してください。
- **Windows** IBM WebSphere MQ for Windows のみ。
  - プログラムからの MQAI コンポーネント・オブジェクト・モデル (COM) 呼び出し
  - Windows デフォルト構成アプリケーション

またオブジェクト属性の表示や変更あるいはオブジェクトの削除もできます。

Windows、UNIX and Linux システム上の WebSphere MQ コマンドのシーケンスの場合、MQSC 機能を使用して、ファイルに保持されている一連のコマンドを実行できます。

## オブジェクトの属性

オブジェクトのプロパティは、その属性によって定義されます。属性の一部はユーザーが指定できますが、他の属性は表示のみ可能です。

例えば、キューが収容できる最大メッセージ長は、*MaxMsgLength* 属性によって定義されます。この属性は、キューの作成時に指定できます。*DefinitionType* 属性は、キューが作成された方法を指定します。この属性は表示のみできます。

WebSphere MQ では、属性を参照する方法としては、次の 2 とおりの方法があります。

- 属性の PCF 名 (例えば、*MaxMsgLength*) を使用する方法
- 属性の MQSC コマンド名 (例えば、MAXMSGL) を使用する方法

本ガイドは主に MQSC コマンドを使って属性を指定する方法について説明しているため、ほとんどの属性を PCF 名ではなく、MQSC コマンド名で指定しています。

## クラスター

キュー・マネージャーは、クラスターにまとめることができます。キュー・マネージャーを 1 つのクラスターとしてまとめると、クラスターを構成するキュー・マネージャーがホスト管理しているキューは、そのクラスター内の他のキュー・マネージャーも使用することができます。同じクラスター内にあるキュー・マネージャー間では、標準の分散キューイングで必要なオブジェクト定義の多くを使用せずに、メッセージを送信することができます。

分散キューイングを使用する従来の WebSphere MQ ネットワークでは、すべてのキュー・マネージャーは独立しています。この場合、あるキュー・マネージャーから別のキュー・マネージャーへメッセージを送信するには、伝送キュー、リモートのキュー・マネージャーへのチャネル、およびメッセージの宛先になるすべてのリモート・キューが定義されていなければなりません。

クラスターは、単一のネットワークを介してキュー・マネージャー間の直接の通信が可能になるように設定されたキュー・マネージャーの集合体です。この場合、伝送キュー、チャネル、およびリモート・キューの定義の必要はありません。

クラスター内の個々のキュー・マネージャーには、同じクラスター内の他のキュー・マネージャーにメッセージを送送するための、単一の伝送キューがあります。

### 関連概念

#### [クラスターの設計](#)

クラスターとは何か、およびその働きについて説明します。

### 関連タスク

#### [キュー・マネージャー・クラスターの構成](#)

#### [新規クラスターのセットアップ](#)

## システム・デフォルト・オブジェクト

システム・デフォルト・オブジェクトの紹介と、詳細情報へのリンク。

システム・デフォルト・オブジェクトとは、キュー・マネージャーの作成時に各キュー・マネージャーごとに自動的に作成される 1 組のオブジェクト定義のことです。ご使用のシステムのアプリケーションで使用するために、これらのオブジェクト定義はすべてコピーしたり、修正することができます。

デフォルト・オブジェクト名には、語幹である SYSTEM が付いています。例えば、デフォルト・ローカル・キューは SYSTEM.DEFAULT.LOCAL.QUEUE であり、デフォルト受信チャネルは SYSTEM.DEF.RECEIVER です。これらのオブジェクトの名前を変更することはできません。これらの名前を持つデフォルト・オブジェクトは必須です。

オブジェクトを定義する際に、明示的に指定されなかった属性は該当するデフォルト・オブジェクトからコピーされます。例えば、ローカル・キューを定義する場合、指定しなかった属性は、デフォルト・キュー SYSTEM.DEFAULT.LOCAL.QUEUE から取られます。

システム・デフォルトについての詳細は、[システムおよびデフォルト・オブジェクト](#)を参照してください。

## IBM WebSphere MQ マルチキャスト

IBM WebSphere MQ Multicast は、待ち時間が短く、ファンアウトが大きい、高信頼性マルチキャスト・メッセージングです。

マルチキャストは、パフォーマンスに悪影響を与えずに非常に多数のサブスクライバーに拡張することが可能な、効率の良いパブリッシュ/サブスクライブ・メッセージング方式です。WebSphere MQ は、肯定応答、否定応答、およびシーケンス番号を使用して、短い待ち時間で大きなファンアウトを実現した、信頼できるマルチキャスト・メッセージングです。

WebSphere MQ Multicast の公平な配信機能により、同時送達に近い通信が可能であり、特定の受信者だけが有利になることはありません。WebSphere MQ Multicast はネットワークを使用してメッセージを配信するため、データをファンアウトするためにパブリッシュ/サブスクライブ・エンジンは必要ありません。トピックがグループ・アドレスにマップされたら、パブリッシャーとサブスクライバーはピアツーピア・モードで作動できるため、キュー・マネージャーは必要ありません。これにより、キュー・マネージャー・サーバーの負荷が軽減され、キュー・マネージャー・サーバーが潜在的な障害点となることはなくなります。

### 初期マルチキャストの概念

通信情報 (COMMINFO) オブジェクトを使用して、WebSphere MQ Multicast を既存のシステムやアプリケーションに簡単に統合できます。2つの TOPIC オブジェクト・フィールドを使用して、マルチキャスト・トラフィックをサポートしたり無視したりするように既存の TOPIC オブジェクトを短時間で構成できます。

### マルチキャストに必要なオブジェクト

以下の情報は、WebSphere MQ Multicast に必要な2つのオブジェクトの概要です。

#### COMMINFO オブジェクト

COMMINFO オブジェクトには、マルチキャスト伝送に関連付けられた属性が含まれます。COMMINFO オブジェクト・パラメーターの詳細については、[DEFINE COMMINFO](#)を参照してください。

必ず設定しなければならない COMMINFO フィールドは、COMMINFO オブジェクトの名前だけです。設定後、この名前はトピックに対して COMMINFO オブジェクトを識別するために使用されます。COMMINFO オブジェクトの **GRPADDR** フィールドを調べて、値が有効なマルチキャスト・グループ・アドレスであることを確認しなければなりません。

#### TOPIC オブジェクト

トピックとは、パブリッシュ/サブスクライブ・メッセージでパブリッシュされる情報のサブジェクトのことで、トピックを定義するには TOPIC オブジェクトを作成します。TOPIC オブジェクト・パラメーターの詳細については、[DEFINE TOPIC](#)を参照してください。

TOPIC オブジェクト・パラメーター **COMMINFO** および **MCAST** の値を変更すると、既存のトピックをマルチキャストで使用できます。

- **COMMINFO** パラメーターは、マルチキャスト通信情報オブジェクトの名前を指定します。
- **MCAST** パラメーターは、トピック・ツリー内のこの位置でマルチキャストを許容するかどうかを指定します。デフォルトでは、**MCAST** は **ASPARENT** に設定されます。これは、トピックのマルチキャスト属性が親から継承されることを意味します。**MCAST** を **ENABLED** に設定すると、このノードでマルチキャスト・トラフィックが許可されます。

## マルチキャスト・ネットワークとトピック

以下の情報は、様々なタイプのサブスクリプションとトピック定義を持つサブスクリプションがどうなるかに関する概要です。これらの例はすべて、TOPIC オブジェクトの **COMMINFO** パラメーターが有効な COMMINFO オブジェクトの名前に設定されていることを前提にしています。

### マルチキャスト有効に設定されているトピック

トピック・ストリング **MCAST** パラメーターが **ENABLED** に設定されていると、以下の場合以外は、マルチキャスト可能なクライアントからのサブスクリプションが許可され、マルチキャスト・サブスクリプションが行われます。

- マルチキャスト可能なクライアントからの永続サブスクリプションである。
- マルチキャスト可能なクライアントからの非管理対象サブスクリプションである。
- マルチキャスト不可能なクライアントからのサブスクリプションである。

これらの場合は、非マルチキャスト・サブスクリプションが行われ、サブスクリプションが通常のパブリッシュ/サブスクライブにダウングレードされます。

### マルチキャスト無効に設定されているトピック

トピック・ストリング **MCAST** パラメーターが **DISABLED** に設定されていると、常に非マルチキャスト・サブスクリプションが行われ、サブスクリプションが通常のパブリッシュ/サブスクライブにダウングレードされます。

### マルチキャスト専用設定されているトピック

トピック・ストリング **MCAST** パラメーターが **ONLY** に設定されていると、以下の場合以外は、マルチキャスト可能なクライアントからのサブスクリプションが許可され、マルチキャスト・サブスクリプションが行われます。

- 永続サブスクリプションです。永続サブスクリプションは拒否され、理由コード [2436 \(0984\) \(RC2436\): MQRC\\_DURABILITY\\_NOT\\_ALLOWED](#) が出力されます。
- これは非管理サブスクリプションです。管理対象外サブスクリプションは拒否され、理由コード [2046 \(07FE\) \(RC2046\): MQRC\\_OPTIONS\\_ERROR](#) が出力されます。
- これは、非マルチキャスト対応クライアントからのサブスクリプションです。これらのサブスクリプションは拒否され、理由コード [2560 \(0A00\) \(RC2560\): MQRC\\_MULTICAST\\_ONLY](#) が出力されます。
- これは、ローカルにバインドされたアプリケーションからのサブスクリプションです。これらのサブスクリプションは拒否され、理由コードが [2560 \(0A00\) \(RC2560\): MQRC\\_MULTICAST\\_ONLY](#) が出力されます。

## 機密保護

IBM WebSphere MQ では、セキュリティーを提供するいくつかの方法があります。許可サービス・インターフェース、ユーザー作成またはサード・パーティーのチャンネル出口、Secure Sockets Layer (SSL) を使用したチャンネル・セキュリティー、チャンネル認証レコード、およびメッセージ・セキュリティーです。

### 許可サービス・インターフェース

MQI 呼び出しの使用、コマンドの発行、およびオブジェクトへのアクセスは、**オブジェクト権限マネージャー (OAM)** によって提供されます。OAM はデフォルトでは使用可能です。IBM WebSphere MQ エンティティーへのアクセスは、IBM WebSphere MQ ユーザー・グループおよび OAM を通して制御されます。管理者はコマンド行インターフェースを使用して、必要に応じて許可を与えたり取り消したりすることができます。

許可サービス・コンポーネントの作成については、[Windows、UNIX and Linux システムでのセキュリティーのセットアップ](#)を参照してください。

### ユーザー作成またはサード・パーティーのチャンネル出口

チャンネルでは、ユーザー作成出口またはサード・パーティー出口を使用できます。詳細については、[メッセージング・チャンネルのためのチャンネル出口プログラム](#)を参照してください。

## SSL を使用したチャンネル・セキュリティ

Secure Sockets Layer (SSL) プロトコルは、業界標準のチャンネル・セキュリティを提供し、盗聴、改ざん、偽名の使用に対して保護します。

SSL は、公開鍵とシンメトリック手法を使用して、メッセージの機密性と保全性、および相互認証を提供します。

SSL の詳細情報を含めた IBM WebSphere MQ のセキュリティに関する総合的なレビューについては、[セキュリティ](#)を参照してください。このセクションで説明したコマンドのポインターを含めた SSL の概要については、[暗号セキュリティ・プロトコル: SSL および TLS](#)を参照してください。

## チャンネル認証レコード

チャンネル認証レコードを使用して、チャンネル・レベルで接続システムに許可されているアクセスに対して正確な制御を行います。詳しくは、[チャンネル認証レコード](#)を参照してください。

## メッセージ・セキュリティ

Advanced Message Security (別個にインストールおよびライセンス交付される IBM WebSphere MQ のコンポーネント) を使用し、IBM WebSphere MQ を使用して送受信されるメッセージに対する暗号保護を提供します。[WebSphere MQ Advanced Message Security](#) を参照してください。

### 関連タスク

[機密保護](#)

[セキュリティ要件の計画](#)

## クライアントおよびサーバー

IBM WebSphere MQ がアプリケーションのためにクライアント/サーバー構成をどのようにサポートしているかについて、概要を説明します。

IBM WebSphere MQ MQI クライアントとは、あるシステム上で稼働しているアプリケーションが、別のシステム上で稼働しているキュー・マネージャーに MQI 呼び出しを発行できるようにするコンポーネントのことです。呼び出しからの出力はクライアントに返送され、さらにクライアントからアプリケーションに戻されます。

IBM WebSphere MQ サーバーとは、キューイング・サービスを 1 つ以上のクライアントに提供するキュー・マネージャーのことです。キューなどのすべての IBM WebSphere MQ オブジェクトは、キュー・マネージャーのマシン上 (IBM WebSphere MQ サーバー・マシン) にもみ存在し、クライアント上には存在しません。IBM WebSphere MQ サーバーは、ローカルの IBM WebSphere MQ アプリケーションもサポートすることができます。

IBM WebSphere MQ サーバーと通常のキュー・マネージャーとの相違点は、サーバーには、各クライアントとの専用通信リンクが備わっているという点です。クライアントとサーバーのためにチャンネルを作成する方法の詳細については、[分散キューイングを使用したアプリケーションの接続](#)を参照してください。

クライアント一般については、[132 ページの『IBM WebSphere MQMQI クライアントの概要』](#)を参照してください。

## クライアント/サーバー環境での IBM WebSphere MQ アプリケーション

サーバーにリンクされていれば、クライアント IBM WebSphere MQ アプリケーションは、ローカル・アプリケーションと同じやり方でたいの MQI 呼び出しを発行することができます。クライアント・アプリケーションは、MQCONN 呼び出しを発行して、指定のキュー・マネージャーに接続します。接続要求から戻された接続ハンドルを指定した追加の MQI 呼び出しは、このキュー・マネージャーによって処理されます。

ユーザーのアプリケーションを、該当するクライアント・ライブラリーにリンクする必要があります。[WebSphere MQ MQI クライアント用のアプリケーションの作成](#)を参照してください。

## 関連概念

138 ページの『トランザクションの管理とサポート』

トランザクション管理、および WebSphere MQ がトランザクションをサポートをする方法に関する概要です。

140 ページの『キュー・マネージャーの機能の拡張』

ユーザー出口、API 出口、またはインストール可能サービスを使用して、キュー・マネージャーの機能を拡張できます。

## IBM WebSphere MQMQI クライアントの概要

WebSphere MQ MQI クライアントは、キュー・マネージャーが実行されていないシステム上にインストールできる、IBM WebSphere MQ 製品のコンポーネントです。

IBM WebSphere MQ MQI クライアントを使用すると、そのクライアントと同じシステムで実行されているアプリケーションは、別のシステムで実行されているキュー・マネージャーに接続できます。アプリケーションは、そのキュー・マネージャーに対して MQI 呼び出しを発行できます。このようなアプリケーションは、WebSphere MQ MQI クライアント・アプリケーションと呼ばれ、キュー・マネージャーはサーバー・キュー・マネージャーと呼ばれます。

IBM WebSphere MQ MQI クライアント・アプリケーションとサーバー・キュー・マネージャーは、MQI チャンネルを使用して相互に通信します。MQI チャンネルは、クライアント・アプリケーションが、キュー・マネージャーに接続するための **MQCONN** または **MQCONNX** 呼び出しを発行した時に開始されます。また、MQI チャンネルは、クライアント・アプリケーションが、キュー・マネージャーとの接続を解除する **MQDISC** 呼び出しを発行した時に終了されます。MQI 呼び出しの入力パラメーターは、MQI チャンネル上で 1 つの方向に流れ、出力パラメーターは、それと反対の方向に流れます。

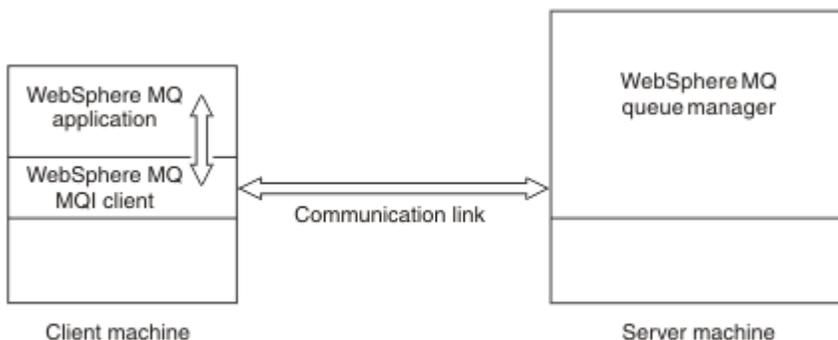


図 30. クライアントとサーバー間のリンク

次のプラットフォームを使用できます。組み合わせは、使用している IBM WebSphere MQ 製品によって異なります。これについては、134 ページの『IBM WebSphere MQ クライアントのプラットフォームのサポート』で説明しています。

### IBM WebSphere MQ MQI クライアント

UNIX and Linux  
Windows

HP Integrity NonStop Server

### IBM WebSphere MQ Server

UNIX and Linux  
Windows

HP Integrity NonStop Server

MQI は、クライアント・プラットフォーム上で実行されているアプリケーションで使用できます。キューおよびその他の IBM WebSphere MQ オブジェクトは、サーバーにインストールしたキュー・マネージャーに保持されます。

IBM WebSphere MQ MQI クライアント環境で実行するアプリケーションは、まず、関連するクライアント・ライブラリーにリンクする必要があります。アプリケーションが MQI 呼び出しを発行すると、IBM

WebSphere MQ MQI クライアントは要求をキュー・マネージャーに送信します。キュー・マネージャーでは要求が処理され、そこから応答が IBM WebSphere MQ MQI クライアントに返送されます。

アプリケーションと IBM WebSphere MQ MQI クライアントとの間のリンクは、実行時に動的に確立されません。

IBM WebSphere MQ classes for .NET、 IBM WebSphere MQ classes for Java または IBM WebSphere MQ classes for Java Message Service (JMS) を使用して、クライアント・アプリケーションを開発することもできます。Java および JMS クライアントは、UNIX、Linux および Windows プラットフォームで使用できます。ここでは、Java および JMS の使用については説明しません。IBM WebSphere MQ classes for Java および IBM WebSphere MQ classes for JMS をインストール、構成、および使用方法については、[WebSphere MQ classes for Java の使用](#) および [WebSphere MQ classes for JMS の使用](#) を参照してください。

## 関連概念

[133 ページの『IBM WebSphere MQ クライアントを使用する理由』](#)

IBM WebSphere MQ クライアントを使用することは、IBM WebSphere MQ メッセージングおよびキューイングを効率的に実装する方法です。

[134 ページの『IBM WebSphere MQ MQI クライアントをセットアップするにはどうすればよいですか?』](#)  
クライアントを設定するには、次の指示に従ってください。

[135 ページの『拡張ランザクション・クライアントの概要』](#)

WebSphere MQ 拡張ランザクション・クライアントは、外部ランザクション・マネージャーの制御の元で、別のリソース・マネージャーが管理するリソースを更新できます。

[137 ページの『クライアントとサーバーの接続方法』](#)

クライアントは MQCONN または MQCONNX を使ってサーバーに接続し、チャンネルを通して通信します。

## IBM WebSphere MQ クライアントを使用する理由

IBM WebSphere MQ クライアントを使用することは、IBM WebSphere MQ メッセージングおよびキューイングを効率的に実装する方法です。

MQI とキュー・マネージャーを別々のマシン上 (物理マシン、仮想マシンのどちらでも) で実行するようにアプリケーションを作成することもできます。MQI とキュー・マネージャーを別のマシンで実行すると、次のような利点があります。

- クライアント・マシン上に WebSphere MQ を完全に実装する必要がなくなります。
- クライアント・システム側のハードウェア要件が減少します。
- システム管理要件が減少します。
- クライアントで実行する WebSphere MQ アプリケーションをさまざまなシステム上の複数のキュー・マネージャーに接続できます。
- 異なる伝送プロトコルを使用する代替チャンネルが使用できます。

## 関連資料

[133 ページの『IBM WebSphere MQ MQI クライアント上で実行されるアプリケーション』](#)  
クライアント環境では MQI が完全にサポートされています。

[134 ページの『IBM WebSphere MQ クライアントのプラットフォームのサポート』](#)

すべてのサーバー・プラットフォーム上の IBM WebSphere MQ は、UNIX または Linux システム、および Windows 上の IBM WebSphere MQ MQI クライアントからのクライアント接続を受け入れます。

*IBM WebSphere MQ MQI* クライアント上で実行されるアプリケーション

クライアント環境では MQI が完全にサポートされています。

これにより、IBM WebSphere MQ MQI クライアント上のアプリケーションを MQI ライブラリーではなく MQIC ライブラリーにリンクすることにより、ほとんどすべての WebSphere MQ アプリケーションを IBM WebSphere MQ MQI クライアント・システム上で実行するように構成することができます。ただし、次の例外があります。

- 信号付きの MQGET

- 他のリソース管理プログラムとの同期点の調整を必要とするアプリケーションでは、拡張トランザクション・クライアントを使用する必要があります。

非持続メッセージングのパフォーマンスを良くするために先読みを使用可能にした場合、すべての MQGET オプションが使用可能な訳ではありません。次の表は、使用できるオプションとそれらを MQGET 呼び出しの間に変更できるかどうかを示します。

|                   | 先読みが有効になっている場合に使用でき、MQGET 呼び出し間で変更できる                                                                                                                                         | 先読みの有効時に使用可能だが、MQGET 呼び出し間で変更不可 <sup>1</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                       | 先読みが有効になっている場合に使用できない MQGET オプション <sup>2</sup>                                                                                               |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| MQGET MD 値        | MsgId <sup>3</sup><br>CorrelId <sup>3</sup>                                                                                                                                   | Encoding<br>CodedCharSetId                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                              |
| MQGET MQGMO オプション | MQGMO_WAIT<br>MQGMO_NO_WAIT<br>MQGMO_FAIL_IF QUIESCING<br>MQGMO_BROWSE_FIRST <sup>4</sup><br>MQGMO_BROWSE_NEXT <sup>4</sup><br>MQGMO_BROWSE_MESSAGE_UNDER_CURSOR <sup>4</sup> | MQGMO_SYNCPOINT_IF_PERSISTENT<br>MQGMO_NO_SYNCPOINT<br>MQGMO_ACCEPT_TRUNCATED_MSG<br>MQGMO_CONVERT<br>MQGMO_LOGICAL_ORDER<br>MQGMO_COMPLETE_MSG<br>MQGMO_ALL_MSGS_AVAILABLE<br>MQGMO_ALL_SEGMENTS_AVAILABLE<br>MQGMO_MARK_BROWSE_HANDLE<br>MQGMO_MARK_BROWSE_CO_OP<br>MQGMO_UNMARK_BROWSE_CO_OP<br>MQGMO_UNMARK_BROWSE_HANDLE<br>MQGMO_UNMARKED_BROWSE_MSG<br>MQGMO_PROPERTIES_FORCE_MQRFH2<br>MQGMO_NO_PROPERTIES<br>MQGMO_PROPERTIES_IN_HANDLE<br>MQGMO_PROPERTIES_COMPATIBILITY | MQGMO_SET_SIGNAL<br>MQGMO_SYNCPOINT<br>MQGMO_MARK_SKIP<br>MQGMO_BACKOUT<br>MQGMO_MSG_UNDER_CURSOR <sup>4</sup><br>MQGMO_LOCK<br>MQGMO_UNLOCK |
| MQGMO 値           |                                                                                                                                                                               | MsgHandle                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                              |

1. これらのオプションが MQGET 呼び出し間で変更された場合、MQRC\_OPTIONS\_CHANGED 理由コードが戻されます。
2. これらのオプションが最初の MQGET 呼び出しで指定されると、先読みは使用不可になります。これらのオプションを後続の MQGET 呼び出しで指定すると、理由コード MQRC\_OPTIONS\_ERROR が戻されます。
3. クライアント・アプリケーション側で以下の点に留意する必要があります。すなわち、MsgId および CorrelId の値が MQGET 呼び出し間で変更された場合、変更前の値によるメッセージがクライアントに送信済みの可能性があり、コンSUME (または自動的にパージ) されるまでクライアントの先読みバッファ内に残るということです。
4. 最初の MQGET 呼び出しは、先読みが有効である場合にメッセージをキューからブラウズするか取得するかを決定します。アプリケーションがブラウズと取得の組み合わせを使用しようとすると、MQRC\_OPTIONS\_CHANGED 理由コードが戻されます。
5. MQGMO\_MSG\_UNDER\_CURSOR は先読みでは使用できません。先読みが有効な場合、メッセージのブラウズまたは取得が可能ですが、ブラウズと取得の組み合わせは指定できません。

IBM WebSphere MQ MQI クライアント上で実行されているアプリケーションは、複数のキュー・マネージャーに同時に接続することも、MQCONN 呼び出しまたは MQCONNX 呼び出しでアスタリスク (\*) 付きのキュー・マネージャー名を使用することもできます (IBM WebSphere MQ MQI クライアント・アプリケーションのキュー・マネージャーへの接続の例を参照してください)。

#### IBM WebSphere MQ クライアントのプラットフォームのサポート

すべてのサーバー・プラットフォーム上の IBM WebSphere MQ は、UNIX または Linux システム、および Windows 上の IBM WebSphere MQ MQI クライアントからのクライアント接続を受け入れます。

基本製品およびサーバーとしてインストールされた WebSphere MQ は、以下のプラットフォーム上の IBM WebSphere MQ MQI クライアントからの接続を受け入れることができます。

- HP Integrity NonStop Server
- UNIX and Linux システム
- Windows

クライアント接続によって、コード化文字セット ID (CCSID) および通信プロトコルが異なる場合があります。

#### IBM WebSphere MQ MQI クライアントをセットアップするにはどうすればよいですか?

クライアントを設定するには、次の指示に従ってください。

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

1. IBM WebSphere MQ MQI クライアント用に適切なプラットフォームがあるかどうか、ハードウェアおよびソフトウェアが要件を満たしているかどうかをチェックします。プラットフォームのサポートについては、[134 ページの『IBM WebSphere MQ クライアントのプラットフォームのサポート』](#)で説明されています。
2. クライアント・ワークステーション上に IBM WebSphere MQ をインストールする方法を決定し、使用するクライアント・プラットフォームとサーバー・プラットフォームの組み合わせに応じた指示に従います。インストールについては、[IBM WebSphere MQ クライアントのインストール](#)で説明されています。
3. 通信リンクが設定され、接続されていることを確認します。通信リンクの構成については、[サーバーとクライアント間の接続の構成](#)で説明されています。
4. インストールが正しく行われたことを確認します。インストールの検証については、[クライアントのインストールの検証](#)で説明されています。
5. IBM WebSphere MQ MQI クライアントのインストールを確認した後、クライアントを保護する必要があるかどうかを検討します。クライアントのセキュリティーについては、[IBM WebSphere MQ MQI クライアント・セキュリティーのセットアップ](#)で説明されています。
6. クライアント上で実行する IBM WebSphere MQ アプリケーションに必要な IBM WebSphere MQ MQI クライアントとサーバー間のチャンネルをセットアップします。チャンネルの設定については、[MQI チャンネルの定義](#)で説明されています。SSL を使用している場合はその他の考慮事項がいくつかあります。これらの考慮事項については、[MQI チャンネルで SSL を使用するように指定する](#)で説明されています。IBM WebSphere MQ MQI クライアント構成ファイルまたは IBM WebSphere MQ 環境変数を使用してチャンネルを設定することが必要な場合があります。IBM WebSphere MQ 環境変数については、[IBM WebSphere MQ 環境変数の使用](#)で説明されています。
7. IBM WebSphere MQ アプリケーションについては、[アプリケーションの開発](#)で説明されています。
8. IBM WebSphere MQ MQI クライアント環境でアプリケーションの設計、ビルド、実行を行う際に、キュー・マネージャー環境のいくつかの相違点について考慮できます。これらの相違点については、以下を参照してください。
  - [クライアント・アプリケーションでのメッセージ・キュー・インターフェース \(MQI\) の使用](#)
  - [WebSphere MQ MQI クライアント用のアプリケーションの作成](#)
  - [キュー・マネージャーへの IBM WebSphere MQ MQI クライアント・アプリケーションの接続](#)
  - [IBM WebSphere MQ MQI クライアントの問題の解決](#)

## 拡張トランザクション・クライアントの概要

WebSphere MQ 拡張トランザクション・クライアントは、外部トランザクション・マネージャーの制御の元で、別のリソース・マネージャーが管理するリソースを更新できます。

トランザクション管理の概念についてよくご存じでない場合は、[138 ページの『トランザクションの管理とサポート』](#)を参照してください。

XA トランザクション・クライアントは、WebSphere MQ の一部として提供されるようになったことに注意してください。

クライアント・アプリケーションは、接続先のキュー・マネージャーによって管理される作業単位に加わることができます。その作業単位内で、クライアント・アプリケーションは、キュー・マネージャーが所有するキューにメッセージを書き込んだり、キューからメッセージを取得したりすることができます。次に、クライアント・アプリケーションは、**MQCMIT** 呼び出しを使用して作業単位をコミットするか、**MQBACK** 呼び出しを使用して作業単位をバックアウトすることができます。しかし、クライアント・アプリケーションは同じ作業単位内で、別のリソース・マネージャーのリソース、例えば DB2® データベースの表を更新することはできません。WebSphere MQ 拡張トランザクション・クライアントを使用すると、この制限がなくなります。

WebSphere MQ 拡張トランザクション・クライアントは、いくつかの追加機能を持つ IBM WebSphere MQ MQI client です。この機能を使用して、クライアント・アプリケーションは、同一の作業単位内で次のタスクを実行できます。

- 接続先のキュー・マネージャーが所有するキューにメッセージを書き込み、そのキューからメッセージを取得する
- WebSphere MQ キュー・マネージャー以外のリソース・マネージャーのリソースを更新する

この作業単位は、クライアント・アプリケーションと同じシステム上で実行されている外部トランザクション・マネージャーによって管理されなければなりません。クライアント・アプリケーションの接続先のキュー・マネージャーが作業単位を管理することはできません。つまり、キュー・マネージャーは、リソース・マネージャーの役目だけをする事ができ、トランザクション・マネージャーの役目をする事はできません。また、クライアント・アプリケーションは、作業単位のコミットまたはバックアウトを行うのに、外部トランザクション・マネージャーが提供するアプリケーション・プログラミング・インターフェース (API) しか使用できません。したがって、クライアント・アプリケーションは、MQI 呼び出し **MQBEGIN**、**MQCMIT**、および **MQBACK** を使用できません。

外部トランザクション・マネージャーは、キュー・マネージャーに接続されるクライアント・アプリケーションによって使用されるのと同じ MQI チャンネルを使用して、リソース・マネージャーとしてのキュー・マネージャーと情報を交換できます。しかし、障害後のリカバリー状態では、アプリケーションが実行していないので、トランザクション・マネージャーは、専用の MQI チャンネルを使用して、障害の時点でキュー・マネージャーが参加していた未完了の作業単位をリカバリーすることができます。

このセクションでは、拡張トランザクション機能を持たない WebSphere MQ MQI クライアントは、*WebSphere MQ* ベース・クライアントと呼ばれます。したがって、WebSphere MQ 拡張トランザクション・クライアントは、拡張トランザクション機能が付加された WebSphere MQ ベース・クライアントであると見なすことができます。

## 関連資料

136 ページの『[拡張トランザクション・クライアントのプラットフォームのサポート](#)』

IBM WebSphere MQ 拡張トランザクション・クライアントは、z/OS を除いてベース・クライアントをサポートするすべてのプラットフォームで利用可能です。

拡張トランザクション・クライアントのプラットフォームのサポート

IBM WebSphere MQ 拡張トランザクション・クライアントは、z/OS を除いてベース・クライアントをサポートするすべてのプラットフォームで利用可能です。

拡張トランザクション・クライアントを使用するクライアント・アプリケーションは、次の IBM WebSphere MQ Version 7.5 製品のキュー・マネージャーのみに接続できます。

- IBM WebSphere MQ ( AIX の場合)
- IBM WebSphere MQ ( HP-UX の場合)
- IBM WebSphere MQ ( HP Integrity NonStop Server の場合)
- IBM WebSphere MQ ( Linux の場合)
- IBM WebSphere MQ ( Solaris の場合)
- IBM WebSphere MQ ( Windows の場合)

z/OS で稼働する拡張トランザクション・クライアントはありませんが、拡張トランザクション・クライアントを使用しているクライアント・アプリケーションが z/OS で稼働するキュー・マネージャーに接続することはできます。

各プラットフォームでは、拡張トランザクション・クライアントのハードウェアおよびソフトウェアの要件は、IBM WebSphere MQ ベース・クライアントの要件と同じです。プログラム言語は、IBM WebSphere MQ ベース・クライアントおよびご使用のトランザクション・マネージャーによってサポートされている場合は、拡張トランザクション・クライアントによってもサポートされます。

各プラットフォームの外部トランザクション・マネージャーは、次の Web ページにリストされています。

| 拡張トランザクション・クライアント・プラットフォーム | Web ページ                                         |
|----------------------------|-------------------------------------------------|
| AIX                        | <a href="#">WebSphere MQ V7.5 の最小要件 - AIX</a>   |
| HP-UX                      | <a href="#">WebSphere MQ V7.5 の最小要件 - HP-UX</a> |

|                             |                                                                             |
|-----------------------------|-----------------------------------------------------------------------------|
| 拡張トランザクション・クライアント・プラットフォーム  | Web ページ                                                                     |
| HP Integrity NonStop Server | <a href="#">IBM WebSphere MQ での HP Integrity NonStop Server クライアント環境の計画</a> |
| Linux                       | WebSphere MQ V7.5 の最小要件 - Linux                                             |
| Solaris                     | WebSphere MQ V7.5 の最小要件 - Solaris                                           |
| Windows                     | WebSphere MQ V7.5 の最小要件 - Windows                                           |

## クライアントとサーバーの接続方法

クライアントは MQCONN または MQCONNX を使ってサーバーに接続し、チャンネルを通して通信します。

IBM WebSphere MQ クライアント環境で実行されるアプリケーションは、クライアント・マシンとサーバー・マシン間の接続をアクティブに保つ必要があります。

接続は、アプリケーションが MQCONN 呼び出しまたは MQCONNX 呼び出しを発行することによって、確立されます。クライアントとサーバーは、MQI チャンネルを介して通信します。ただし、共用会話の使用時には、それぞれの会話どうしが MQI チャンネル・インスタンスを共有します。その呼び出しが成功すると、アプリケーションが MQDISC 呼び出しを発行するまで、MQI チャンネル・インスタンスまたは会話は接続されたままになります。このことは、アプリケーションが接続する必要があるどのキュー・マネージャーにも該当します。

### 関連概念

[137 ページの『同一マシン上のクライアントとキュー・マネージャー』](#)

キュー・マネージャーがインストールされているマシンでも、WebSphere MQ MQI クライアント環境でアプリケーションを実行することができます。

[137 ページの『異なるプラットフォーム上のクライアント』](#)

次に WebSphere MQ MQI クライアント/サーバー・システムのもう 1 つの例を示します。この例では、サーバー・マシンは、別々のプラットフォーム上にある 3 つの WebSphere MQ MQI クライアントと通信します。

[138 ページの『異なるバージョンのクライアントおよびサーバー・ソフトウェアの使用』](#)

旧バージョンの IBM WebSphere MQ 製品を使用する場合は、クライアントの CCSID からのコード変換がサーバーでサポートされることを確認してください。

#### 同一マシン上のクライアントとキュー・マネージャー

キュー・マネージャーがインストールされているマシンでも、WebSphere MQ MQI クライアント環境でアプリケーションを実行することができます。

この場合、キュー・マネージャーのライブラリー、クライアントのライブラリーのどちらにでも、リンクすることができます。ただし、クライアントのライブラリーにリンクする場合でも、チャンネル接続を定義する必要があることに注意してください。キュー・マネージャーがインストールされているマシンで、クライアント環境を実現できると、アプリケーションの開発段階で役に立ちます。他のマシンに依存することなく、開発者自身のマシン上でプログラムをテストすることができ、独立した WebSphere MQ MQI クライアント環境に移動したときに、これまでどおり順調に稼働させることができます。

#### 異なるプラットフォーム上のクライアント

次に WebSphere MQ MQI クライアント/サーバー・システムのもう 1 つの例を示します。この例では、サーバー・マシンは、別々のプラットフォーム上にある 3 つの WebSphere MQ MQI クライアントと通信します。

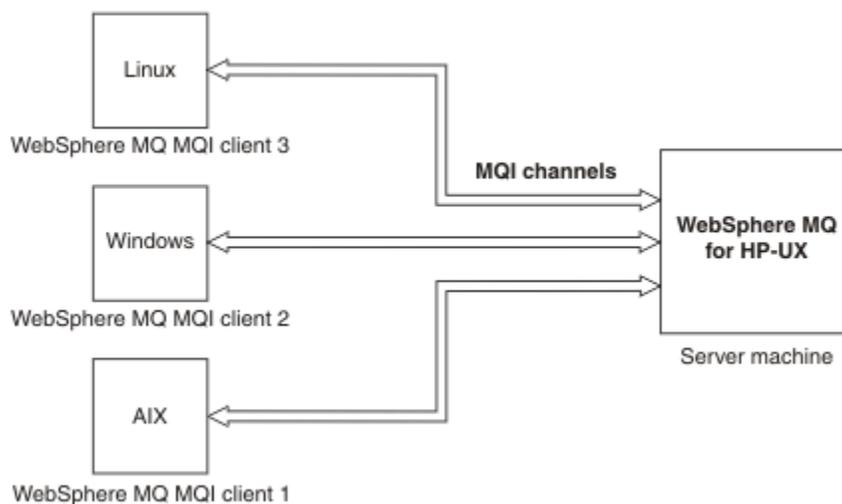


図 31. 異なるプラットフォーム上のクライアントに接続された WebSphere MQ サーバー

他のもっと複雑な環境も可能です。例えば、1つの WebSphere MQ クライアントが複数のキュー・マネージャーに接続する環境や、キュー共有グループの一部として接続されている任意の数のキュー・マネージャーに接続する環境も実現できます。

#### 異なるバージョンのクライアントおよびサーバー・ソフトウェアの使用

旧バージョンの IBM WebSphere MQ 製品を使用する場合は、クライアントの CCSID からのコード変換がサーバーでサポートされることを確認してください。

IBM WebSphere MQ クライアントは、サポートされるすべてのバージョンのキュー・マネージャーに接続できます。旧バージョンのキュー・マネージャーに接続する場合は、クライアント上の IBM WebSphere MQ アプリケーションで、製品のより新しいバージョンの機能および構造体は使用できません。

IBM WebSphere MQ キュー・マネージャーは、相互にサポートされている最も高いプロトコル・レベルまでネゴシエーションすることにより、異なるバージョンのクライアントとそれ自体と通信することができます。これは、古いクライアントが新しいキュー・マネージャー・レベルで使用される可能性があることを意味します。問題診断を容易にし、IBM によるサポートを有効にするために、クライアントとサーバーの両方が、現在サポートされている IBM WebSphere MQ のバージョンであることをお勧めします。

詳しくは、[使用するプログラミング言語の決定](#)でサポートされているプログラミング言語を参照してください。

## トランザクションの管理とサポート

トランザクション管理、および WebSphere MQ がトランザクションをサポートをする方法に関する概要です。

リソース・マネージャーは、アプリケーションによるアクセスと更新が可能なリソースを所有し管理する、コンピューター・サブシステムです。例えば、次のリソース・マネージャーがあります。

- WebSphere MQ キュー・マネージャー、キューがリソース
- DB2 データベース、テーブルがリソース

アプリケーションが1つ以上のリソース・マネージャーのリソースを更新する場合、所定の更新がすべて1つのグループとして正常に必ず完了するようにする、またはすべて完了しないようにすることが、ビジネス上必要になる場合があります。こうした必要が生じる理由は、これらの更新の中で正常に完了しているものと、完了していないものがあると、ビジネス・データが不整合のままになるからです。

このように管理されるリソースの更新は、作業単位、またはトランザクション内で行われます。アプリケーション・プログラムは、更新のセットを作業単位にグループ化できます。

1つの作業単位の中で、アプリケーションは、リソース・マネージャーのリソースを更新する要求をリソース・マネージャーに出します。アプリケーションが、すべての更新をコミットする要求を出すと、この作

業単位は終了します。更新がコミットされるまで、それらの更新は、同じリソースにアクセスする他のアプリケーションから見えません。または、アプリケーションは、なんらかの理由で作業単位を完了できないと判断した場合、その時点までに要求したすべての更新をバックアウトする要求を出すことができます。この場合、これらの更新はいずれも、他のアプリケーションから見えません。作業単位としてグループ化される更新は相互に論理的に関連しているため、データ保全性を維持できるように正常に処理されなければなりません。ある更新が正常に行われても、別の更新が失敗すれば、データ保全性は失われます。

作業単位が正常に完了したことを、コミットされたといいます。コミットされると、作業単位内のすべての更新内容が永久的になり、これ以降は取り消し不可能になります。しかし、作業単位が失敗した場合は、すべての更新がバックアウトされます。このような、データの保全性を維持しながら作業単位をコミットまたはバックアウトするプロセスのことを同期点調整といいます。

作業単位内のすべての更新がコミットされるか、バックアウトされる時点は、同期点と呼ばれます。作業単位内の更新は、同期点制御内で行われます。アプリケーションが、同期点制御外にある更新を要求する場合、リソース・マネージャーは、進行中の作業単位があっても、その更新をただちにコミットします。この更新を後でバックアウトすることはできません。

作業単位を管理するコンピューター・サブシステムは、トランザクション・マネージャー、または同期点コーディネーターと呼ばれます。

ローカル作業単位とは、WebSphere MQ キュー・マネージャーのリソースのみが更新対象のリソースとなる作業単位をいいます。この場合、キュー・マネージャー自体が単一フェーズ・コミットによって同期点を調整します。

グローバル作業単位では、XA 準拠データベースなどの他のリソース・マネージャーにより管理されているリソースも更新されます。この場合、グローバル作業単位に対しては必ず2フェーズ・コミットが実行され、キュー・マネージャー自体によって調整されるか、あるいはIBM TXSeries® や BEA Tuxedo などのXA 準拠トランザクション・マネージャーによって外部から調整されます。

トランザクション・マネージャーは、作業単位内のリソースの更新がすべて正常に完了したか、完了しなかったかを確認します。アプリケーションが、作業単位のコミットまたはバックアウトを要求するのは、トランザクション・マネージャーに対してです。例えば、トランザクション・マネージャーの例としてCICS や WebSphere Application Server がありますが、どちらも他の機能も持っています。

一部のリソース・マネージャーは、独自のトランザクション管理機能を備えています。例えば、WebSphere MQ キュー・マネージャーは、独自のリソースの更新およびDB2 表の更新を伴う作業単位を管理できます。キュー・マネージャーは、この機能を実行するのに別個のトランザクション・マネージャーを必要としません。ただし、ユーザーの要求であれば、トランザクション・マネージャーを使用することはできます。別個のトランザクション・マネージャーを使用する場合、外部トランザクション・マネージャーと呼ばれます。

外部トランザクション・マネージャーが作業単位を管理するには、トランザクション・マネージャーと、作業単位に加わっているすべてのリソース・マネージャーとの間に、標準インターフェースが存在している必要があります。このインターフェースにより、トランザクション・マネージャーとリソース・マネージャーは互いに情報を交換することが可能になります。これらのインターフェースの1つがXA インターフェースです。XA インターフェースは、複数のトランザクション・マネージャーとリソース・マネージャーによってサポートされる標準インターフェースです。XA インターフェースは、The Open Group によって *Distributed Transaction Processing: The XA Specification* の中で公開されています。

複数のリソース・マネージャーが1つの作業単位に加わっている場合、トランザクション・マネージャーは、システム障害の場合であっても、2フェーズ・コミット・プロトコルを使用して、その作業単位内のすべての更新が正常に完了するか、完了しないかを確認する必要があります。アプリケーションが、作業単位をコミットする要求をトランザクション・マネージャーに出す場合、トランザクション・マネージャーは次のことを行います。

### フェーズ1(コミットの準備)

トランザクション・マネージャーは、作業単位に加わっている各リソース・マネージャーに対して、そのリソースの対象の更新についてのすべての情報が、リカバリー可能な状態であることを確認するように求めます。リソース・マネージャーは通常これを確認するために、情報をログに書き込み、その情報がハード・ディスクに書き込まれるようにします。トランザクション・マネージャーが、そのリソースの対象の更新についての情報が、リカバリー可能な状態であるという通知を各リソース・マネージャーから受け取ると、フェーズ1が完了します。

## フェーズ 2 (コミットの実行)

フェーズ 1 が完了すると、トランザクション・マネージャーは、作業単位をコミットするという、取り消すことができない決定を行います。トランザクション・マネージャーは、その作業単位に加わっている各リソース・マネージャーに対して、そのリソースの更新をコミットするように求めます。リソース・マネージャーは、この要求を受け取ると、更新をコミットする必要があります。この段階で更新をバックアウトすることはできません。トランザクション・マネージャーが、そのリソースの更新をコミットしたという通知を各リソース・マネージャーから受け取ると、フェーズ 2 が完了します。

XA インターフェースは、2 フェーズ・コミット・プロトコルを使用します。

詳細については、[トランザクション・サポート](#)を参照してください。

WebSphere MQ は、Microsoft Transaction Server (COM+) のサポートも提供します。[Microsoft Transaction Server \(COM+\) の使用](#)では、COM+ サポートを利用するために WebSphere MQ をセットアップする方法について説明します。

## キュー・マネージャーの機能の拡張

ユーザー出口、API 出口、またはインストール可能サービスを使用して、キュー・マネージャーの機能を拡張できます。

### ユーザー出口

ユーザー出口は、独自のコードをキュー・マネージャー機能に挿入するメカニズムを提供します。サポートされるユーザー出口は、次のとおりです。

#### チャンネル出口

この出口では、チャンネルの動作方法を変更できます。チャンネル出口については、[メッセージング・チャンネルのためのチャンネル出口プログラム](#)で説明されています。

#### データ変換出口

この出口では、アプリケーション・プログラムに書き込んでデータの形式を変換するための部分ソース・コードを作成できます。データ変換出口については、[データ変換出口の作成](#)で説明されています。

#### クラスター・ワークロード出口

この出口により実行できる機能は、出口のプロバイダーにより定義されます。呼び出し定義の情報は、[MQ CLUSTER WORKLOAD EXIT - 呼び出しの説明](#)にあります。

## API 出口

API 出口を使用すると、MQPUT および MQGET などの WebSphere MQ API 呼び出しの動作を変更するコードを作成して、これらの呼び出しの直前または直後に、そのコードを挿入することができます。挿入は自動的に行われます。キュー・マネージャーは登録されているポイントで出口コードを駆動します。API 出口の詳細は、API 出口の使用/作成方法を参照してください。

## インストール可能サービス

インストール可能サービスには、複数のエントリー・ポイントを持つ形式化されたインターフェース (API) があります。

インストール可能サービスは、サービス・コンポーネントと呼ばれるもので実現されます。WebSphere MQ で提供されているコンポーネントを使用することも、必要な機能を実行するコンポーネントを独自に作成することもできます。

現在、次のようなインストール可能サービスが用意されています。

### 許可サービス

これにより、独自のセキュリティ機能を構築することができます。

このサービスを実装するデフォルト・サービス・コンポーネントは、オブジェクト権限マネージャー (OAM) です。デフォルトでは、OAM はアクティブになっています。つまり、OAM を構成するための作業は一切必要ありません。許可サービス・インターフェースを使用して別のコンポーネントを作成

し、OAM を置き換えたり OAM を補強したりできます。OAM の詳細については、[Windows、UNIX and Linux システムでのセキュリティーのセットアップ](#)を参照してください。

#### ネーム・サービス

これにより、アプリケーションはリモート・キューをローカル・キューであるかのように認識することができ、キューを共用できます。

所有するネーム・サービス・コンポーネントを書き込むことができます。これは、例えば、IBM WebSphere MQ でネーム・サービスを使用する予定の場合に行う必要が生じることがあります。ネーム・サービスを使用するには、ユーザー作成によるコンポーネント、または別のソフトウェア・ベンダーによって提供されたコンポーネントのどちらかが必要です。デフォルトでは、ネーム・サービスは使用できない状態になっています。

## IBM WebSphere MQ Client for HP Integrity NonStop Server 技術概要

HP Integrity NonStop Server オペレーティング・システムの技術概要。

### HP Integrity NonStop Server SupportPac 用 IBM WebSphere MQ クライアント

HP Integrity NonStop Server 用の IBM WebSphere MQ クライアントは、SupportPac MAT1 でリリースされています。

### HP Integrity NonStop Server オペレーティング・システムの技術概要

HP Integrity NonStop Server は、複数のハードウェアまたはソフトウェアの障害があっても、計画されているダウン時間や計画されていないダウン時間を伴わない、最も高い可用性を実現するために設計されたオペレーティング・システムです。このオペレーティング・システムは直線的にスケラブルです。例えば、ハードウェアを 20 パーセント追加すれば、使用可能なパフォーマンスが 20 パーセント増えます。データ保全性を維持するため、このオペレーティング・システムには独自のトランザクション・マネージャーとトランザクション・ファイル・システムがあります。

HP Integrity NonStop Server オペレーティング・システムが使用されるのは、通常、以下の用途です。

- 金融機関。ATM ネットワーク、オンライン・バンキング・サポート、信用照会、証券取引、商取引、銀行間取引など。
- 製造業。Web ストアのバックエンド、在庫、プロセス制御など。
- 通信業。交換、救急、その他のネットワーク・サービスなど。

### IBM WebSphere MQ クライアント (HP Integrity NonStop Server 用) でサポートされる環境と機能

IBM WebSphere MQ クライアント (HP Integrity NonStop Server プラットフォーム用) について詳しく説明し、サポートされるクライアント API と環境、および HP Integrity NonStop Server システム固有のクライアント機能を示します。

#### サポートされるクライアント API および環境

IBM WebSphere MQ クライアント (HP Integrity NonStop Server 用) は、以下の実行環境をサポートします。

|       | OSS | Guardian |
|-------|-----|----------|
| C     | ✓   | ✓        |
| JMS   | ✓   |          |
| COBOL | ✓   | ✓        |
| pTAL  | ✓   | ✓        |

## 機能の要約

一部のクライアント機能は、ホスト・オペレーティング・システムに固有です。以下の要約は、IBM WebSphere MQ クライアント (HP Integrity NonStop Server 用) に固有のクライアント機能を示しています。

- C (ネイティブ)、PTAL、COBOL (ネイティブ)
  - ネットワーク・プロトコル: TCP (IPv4 および IPv6)
  - トランスポート・タイプ: クライアントのみ
  - トランスポート・セキュリティー: SSL/TLS
  - トランザクション・サポート: Transaction Management Facility (TMF) によって調整される 2 フェーズ・コミット (IBM WebSphere MQ Version 7.1 以降のキュー・マネージャーへの接続が必要)
  - アドレッシング・モード: 32 ビット
- Java Message Service (JMS)
  - ネットワーク・プロトコル: TCP (IPv4 および IPv6)
  - トランスポート・タイプ: クライアントのみ (Bindings、Direct、および Direct HTTP はサポートされません)
  - トランスポート・セキュリティー: SSL/TLS
  - トランザクション・サポート: 単一フェーズ・コミット
  - 実行: スタンドアロン (アプリケーション支援機能 (ASF) および J2EE コネクタ・アーキテクチャ (JCA) はサポートされません)
  - 出口: Java 言語のみ (他の言語で作成されたネイティブ出口はサポートされません)
  - IBM WebSphere MQ ヘッダーと PCF: 以下のクラスはサポートされていません:  
com.ibm.mq.headers.\* および com.ibm.mq.pcf.\*

## シナリオ

このセクションでは、WebSphere MQ バージョン 7.5 の新機能を使用および結合する方法を示すシナリオについて情報を提供します。これらのシナリオには、対象となる分野についてさらに理解を深める上で役立つインフォメーション・センターのコンテンツへの有用なリンクが含まれています。

使用可能なシナリオについては、以下のサブトピックに説明があります。

### IBM WebSphere MQ Version 7.5 はじめに

このシナリオでは、Windows プラットフォーム上で IBM WebSphere MQ Version 7.5 を開始する方法について説明します。IBM WebSphere MQ をまだ使用したことがなく、すぐに使用を開始する場合に、このシナリオを使用します。

このシナリオには、以下のセクションがあります。このシナリオに示されているように、グラフィカル・インターフェースまたはコマンド行インターフェースのいずれかを使用して、これらの手順を実行できます。

### 基本概念と重要な用語

『IBM WebSphere MQ Version 7.5 の概要』のシナリオを使用する前に理解しておく必要のある、基本概念および重要な用語について説明します。

### 基本概念

IBM WebSphere MQ によって、アプリケーションは、キューに対するメッセージの読み取り/書き込みを実行できます。メッセージを読み取るアプリケーションは、メッセージを書き込むアプリケーションから独立しています。2つのアプリケーションを同時に実行する必要はありません。メッセージを読み取るアプリケーションが使用不可である場合、アプリケーションがメッセージを読み取るまで、そのメッセージは IBM WebSphere MQ キューで待機します。

## 重要な用語

以下に、メッセージ・キューイングに関する重要な用語のリストを示します。  
メッセージ・キューイングに関する重要な用語。

| 用語         | 説明                                                                                                                                                                    |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| キュー・マネージャー | キュー・マネージャーは、所有するキューを管理し、受信したすべてのメッセージを適切なキューに格納します。                                                                                                                   |
| メッセージ      | メッセージを使用するアプリケーションにとって、メッセージは意味のあるバイト・ストリングです。メッセージは、ある1つのアプリケーション・プログラムから別のアプリケーションに情報を転送するために使用されます。アプリケーションは、同じコンピューターで実行することも、異なるコンピューターで実行することもできます。             |
| ローカル・キュー   | ローカル・キューとは、メッセージを保管するためのデータ構造体です。このキューは、通常キューであることも、伝送キューであることも可能です。通常キューは、キュー・マネージャーからメッセージを直接読み取るアプリケーションによって読み取られるメッセージを保持します。伝送キューは、別のキュー・マネージャーに転送中のメッセージを保持します。 |
| リモート・キュー   | リモート・キューは、別のキュー・マネージャーにメッセージをアドレッシングするために使用されます。                                                                                                                      |
| チャンネル      | チャンネルは、キュー・マネージャー間でメッセージを送受信するために使用されます。                                                                                                                              |
| リスナー       | リスナーは、他のキュー・マネージャーまたはクライアント・アプリケーションからネットワーク要求を受け取るプロセスで、関連付けられたチャンネルを始動します。                                                                                          |

## QM1 という名前のキュー・マネージャーの作成

コマンド行インターフェースまたは WebSphere MQ エクスプローラーのいずれかを使用して、「IBM WebSphere MQ Version 7.5 の概要」シナリオで使用する QM1 というキュー・マネージャーを作成します。キュー・マネージャーは、WebSphere MQ メッセージング・ネットワークのメイン・コンポーネントです。

### 始める前に

IBM WebSphere MQ Version 7.5 がインストールされている必要があります。インストールされていない場合、その方法について、[インストールおよびアンインストール](#)を参照してください。

### このタスクについて

この例では、すべての名前が大文字になっています。IBM WebSphere MQ の名前には大/小文字の区別があるため、すべての名前は大文字で入力する必要があります。

### コマンド行インターフェースを使用したキュー・マネージャーの作成

コマンド行インターフェースを使用してキュー・マネージャーを作成および開始するには、以下のステップを実行します。

### 手順

1. 以下のコマンドを入力して、QM1 という名前でキュー・マネージャーを作成します。

```
crtmqm QM1
```

システムによってキュー・マネージャーが作成されると、以下の出力が表示されます。

```
C:\>crtmqm QM1
WebSphere MQ queue manager created.
Creating or replacing default objects for QM1.
Default objects statistics : 61 created. 0 replaced. 0 failed.
```

```
Completing setup.
Setup completed.
```

キュー・マネージャーが作成され、停止しています。キュー・マネージャーを管理し、そのキューのメッセージの読み取り/書き込みを実行するには、キュー・マネージャーを開始する必要があります。

2. 次のコマンドを入力して、キュー・マネージャーを開始します。

```
strmqm QM1
```

キュー・マネージャーが正常に開始されると、次の出力が表示されます。

```
C:\>strmqm QM1
WebSphere MQ queue manager 'QM1' starting.
5 log records accessed on queue manager 'QM1' during the log replay phase.
Log replay for queue manager 'QM1' complete.
Transaction manager state recovered for queue manager 'QM1'.
WebSphere MQ queue manager 'QM1' started.
```

キュー・マネージャーが開始されます。

## 次のタスク

キューの作成については、144 ページの『[LQ1 という名前のキューの作成](#)』を参照してください。

### WebSphere MQ エクスプローラーを使用したキュー・マネージャーの作成

WebSphere MQ エクスプローラーを使用してキュー・マネージャーを作成および開始するには、以下のステップを実行します。

#### 手順

1. WebSphere MQ エクスプローラーを起動します。
2. **Navigator** ビューで、「キュー・マネージャー」フォルダーを右クリックし、「新規」>「キュー・マネージャー」をクリックします。「キュー・マネージャーの作成」ウィザードが開始します。
3. 「キュー・マネージャー名」フィールドに、QM1 と入力します。
4. **Make this the default queue manager** チェック・ボックスを選択します。
5. 「送達不能キュー」フィールドに、SYSTEM.DEAD.LETTER.QUEUE と入力します。  
これは、キュー・マネージャーを作成するときに自動的に作成される送達不能キューの名前です。
6. 他のフィールドを空のままにして「完了」をクリックするか、このボタンが使用不可になっている場合は「次へ」をクリックします。  
ポート番号が既存のキュー・マネージャー（デフォルト構成の一部として作成されたキュー・マネージャーなど）と競合する場合は、「完了」ボタンが使用不可になります。ウィザードで先に進んでデフォルトのポート番号を変更する必要があります。
7. 「次へ」をクリックした場合は、デフォルトを受け入れて、各ページで「次へ」をクリックします。ウィザードの最後のページに達すると、「完了」ボタンが使用可能になります。指定のポート番号を 1415 などに変更し、「完了」をクリックします。

WebSphere MQ で「キュー・マネージャーの作成」ダイアログ・ウィンドウが表示され、キュー・マネージャーが作成および開始されます。

## 次のタスク

キューの作成については、144 ページの『[LQ1 という名前のキューの作成](#)』を参照してください。

### LQ1 という名前のキューの作成

『WebSphere IBM WebSphere MQ Version 7.5 の概要』シナリオで使用するために、コマンド行インターフェースまたは WebSphere MQ エクスプローラーを使用してキューを作成します。キューは、メッセージを保管するために使用されるデータ構造体であり、IBM WebSphere MQ キュー・マネージャーのオブジェクトです。

## このタスクについて

IBM WebSphere MQ オブジェクトを作成する 3 つの方法があります。

- コマンド行。
- IBM WebSphere MQ Explorer.
- プログラマブル・インターフェースの使用。

このタスクでは、コマンド行または IBM WebSphere MQ Explorer を使用して IBM WebSphere MQ オブジェクトを作成できます。

## コマンド行インターフェースを使用したキューの作成

コマンド行インターフェースには、IBM WebSphere MQ Script Commands (MQSC) と呼ばれるスクリプト言語があります。スクリプト・ツール **runmqsc** を使用し、キュー・マネージャーに対してスクリプトを実行します。コマンド行インターフェースを使用してキューを作成および開始するには、以下のステップを実行します。

## 手順

1. 次のコマンドを入力して、スクリプト・ツールを開始します。

```
runmqsc QM1
```

スクリプト・ツールを開始すると、次の出力が表示されます。

```
C:\>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
```

これで、ツールは MQSC コマンドを受け入れることができます。

2. 以下の MQSC コマンドを入力して、LQ1 という名前のローカル・キューを作成します。

```
define qlocal(LQ1)
```

キューが作成されると、以下の出力が表示されます。

```
define qlocal(LQ1)
 2 : define qlocal(LQ1)
AMQ8006: WebSphere MQ queue created.
```

3. 次の MQSC コマンドを入力して、スクリプト・ツールを停止します。

```
end
```

スクリプト・ツールが終了する時、以下の出力が表示されます。

```
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.

C:\>
```

## 次のタスク

これで、メッセージをキューに書き込む準備が整いました。メッセージをキューに置く場合は、[146 ページの『キュー LQ1 へのメッセージの書き込み』](#)を参照してください。

## WebSphere MQ エクスプローラーを使用したキューの作成

WebSphere MQ エクスプローラーを使用してキューを作成および開始するには、以下のステップを実行します。

## 手順

1. 「ナビゲーター」ビューで、**Queue Managers** フォルダーを展開します。
2. キュー・マネージャー「**QM1**」を展開します。
3. 「キュー」フォルダーを右クリックし、「新規」>「ローカル・キュー...」をクリックします。「新規ローカル・キュー」ウィザードが起動します。
4. 名前フィールドに、LQ1 と入力します。
5. 「完了 (Finish)」をクリックします。

「コンテンツ」ビューに、新規のキュー LQ1 が表示されます。キューが「コンテンツ」ビューに表示されない場合は、「コンテンツ」ビューの上部にある「リフレッシュ」ボタンをクリックします。

## 次のタスク

これで、メッセージをキューに書き込む準備が整いました。メッセージをキューに置く場合は、[146 ページの『キュー LQ1 へのメッセージの書き込み』](#)を参照してください。

## キュー LQ1 へのメッセージの書き込み

「IBM WebSphere MQ Version 7.5 の概要」シナリオで使用するために、コマンド行インターフェースまたは IBM WebSphere MQ Explorer のいずれかを使用して、キュー LQ1 にメッセージを書き込みます。

## このタスクについて

IBM WebSphere MQ には、amqsput という名前のサンプル・アプリケーションが付属しています。このアプリケーションは、事前定義されたキューにメッセージを書き込みます。

## コマンド行インターフェースを使用したキューへのメッセージの書き込み

コマンド行インターフェースを使用してキューにメッセージを書き込むには、以下のステップを実行します。

## 手順

1. 以下のコマンドを入力することにより、amqsput サンプル・アプリケーションを使用して、メッセージをキュー LQ1 に書き込みます。

```
amqsput LQ1 QM1
```

サンプル・アプリケーションを開始すると、次の出力が表示されます。

```
C:\>amqsput LQ1 QM1
Sample AMQSPUT0 start
target queue is LQ1
```

2. Hello World と入力して、Enter キーを押します。「Hello World」というテキストを格納したメッセージを QM1 という名前のキュー・マネージャーによって管理されるキュー LQ1 に入れました。
3. amqsput を終了するには、Enter キーを押します。以下の出力が表示されます。

```
C:\>amqsput LQ1 QM1
Sample AMQSPUT0 start
target queue is LQ1
Hello World

Sample AMQSPUT0 end
```

## 次のタスク

キューからメッセージを読み取る場合は、[147 ページの『キュー LQ1 からのメッセージの取得』](#)を参照してください。

## IBM WebSphere MQ Explorer を使用したキューへのメッセージの書き込み

IBM WebSphere MQ Explorer を使用してキューにメッセージを書き込むには、以下のステップを実行します。

### 手順

1. 「ナビゲーター」ビューで、**Queue Managers** フォルダを展開します。
2. 作成したキュー・マネージャー QM1 を展開します。
3. 「キュー」フォルダをクリックします。キュー・マネージャーのキューが「コンテンツ」ビューにリストされます。
4. 「コンテンツ」ビューでローカル・キュー LQ1 を右クリックし、「テスト・メッセージの書き込み...」をクリックします。  
「テスト・メッセージの書き込み」ダイアログが開きます。
5. 「メッセージ・データ」フィールドに何らかのテキスト (Hello World など) を入力し、「メッセージの書き込み」をクリックします。  
「メッセージ・データ」フィールドがクリアされ、メッセージがキューに書き込まれます。
6. 「クローズ」をクリックします。  
「コンテンツ」ビューで、LQ1 「現在のキュー項目数」の値が 1 になったことに注意してください。現行キュー項目数の列が表示されていない場合、コンテンツ・ビューの右側にスクロールする必要があります。

### 次のタスク

キューからメッセージを読み取る場合は、[147 ページの『キュー LQ1 からのメッセージの取得』](#)を参照してください。

## キュー LQ1 からのメッセージの取得

コマンド・ライン・インターフェースまたは IBM WebSphere MQ Explorer のいずれかを使用して、「IBM WebSphere MQ Version 7.5 の概要」シナリオで使用するために、キュー LQ1 からメッセージを取得します。

### このタスクについて

IBM WebSphere MQ には、amqsget という名前のサンプル・アプリケーションが付属しています。このアプリケーションは、キューからメッセージを読み取ります。

### コマンド行インターフェースを使用したキューからのメッセージの読み取り

コマンド行インターフェースを使用してキューからメッセージを読み取るには、以下のステップを実行します。

### 手順

以下のコマンドを入力することにより、amqsget サンプル・アプリケーションを使用して、キュー LQ1 のメッセージを読み取ります。

```
amqsget LQ1 QM1
```

サンプル・アプリケーションを開始すると、次の出力が表示されます。

```
C:\>amqsget LQ1 QM1
Sample AMQSGET0 start
message <Hello World>
no more messages
Sample AMQSGET0 end
```

メッセージを読み取った 30 秒後に、amqsget アプリケーションは終了します。

## 次のタスク

キューイング・アプリケーションの作成、キュー・マネージャーへの接続と切断、パブリッシュ/サブスクライブ、およびオブジェクトのオープン/クローズについて学習するには、[キューイング・アプリケーションの作成](#)を参照してください。

### IBM WebSphere MQ Explorer を使用したキューからのメッセージの読み取り

IBM WebSphere MQ Explorer を使用してキューからメッセージを読み取るには、以下のステップを実行します。

#### 手順

1. 「ナビゲーター」ビューで「キュー・マネージャー」フォルダーを展開し、次に QM1 を展開します。
2. 「キュー」フォルダーをクリックします。
3. 「コンテンツ」ビューで QM1 を右クリックし、「メッセージのブラウズ...」をクリックします。「メッセージ・ブラウザー」が開き、現在 QM1 に入っているメッセージのリストが表示されます。
4. 最後のメッセージをダブルクリックして、プロパティ・ダイアログを開きます。  
プロパティ・ダイアログの「データ」ページの「メッセージ・データ」フィールドに、人間が読める形式でメッセージの内容が表示されます。

## 次のタスク

キューイング・アプリケーションの作成、キュー・マネージャーへの接続と切断、パブリッシュ/サブスクライブ、およびオブジェクトのオープン/クローズについて学習するには、[キューイング・アプリケーションの作成](#)を参照してください。

## 次のタスク

『IBM WebSphere MQ Version 7.5 の概要』のシナリオを完了した後に実行する内容。

IBM WebSphere MQ では、特定の WebSphere 製品オフリングのスキルを獲得するためのパスを定義することで、ユーザーに役立つ役割ベースのトレーニング・パスを利用できます。IBM WebSphere MQ には、2つのトレーニング・パスがあります。

#### • アプリケーション開発者

このユーザーは、キュー・マネージャーを使用するアプリケーションの作成を担当します。このシナリオでは、amqsput および amqsget のアプリケーションを書き込みます。

#### • システム管理者

このユーザーは、キュー・マネージャーとそのオブジェクトの作成を担当します。通常、このシナリオで網羅されるのと似たタスクを実行します。

IBM WebSphere MQ のトレーニング・パスの詳細については、<https://www.ibm.com/software/websphere/education/paths/> を参照してください。

IBM WebSphere MQ のコースの完全なリストを確認するには、<https://www.ibm.com/software/websphere/education/curriculum/appint/wmq/> を参照してください。

IBM WebSphere MQ での到達したスキル・レベルを示す認定プログラムを使用できます。詳しくは、<https://www.ibm.com/certify/index> を参照してください。

他のユーザーとコラボレーションできます。例については、以下を参照してください。

- IBM WebSphere MQ プロフェッショナルの E メール・ベースのコミュニティについては、<https://listserv.meduniwien.ac.at/archives/mqser-l.html> を参照してください。
- IBM WebSphere MQ 製品ファミリーに特化したディスカッション・フォーラムについては、<https://www.mqseries.net/> を参照してください。
- さまざまな IBM メッセージング製品の開発者による IBM Developer ブログについては、<https://www.ibm.com/developerworks/blogs/page/messaging/> を参照してください。

- IBM 提供の IBM WebSphere MQ 公式フォーラムについては、<https://www.ibm.com/developerworks/forums/forum.jspa?forumid=280> を参照してください。
- stackoverflow.com の IBM WebSphere MQ のタグが付けられた質疑応答については、<https://stackoverflow.com/questions/tagged/websphere-mq> を参照してください。

この製品資料には、その他にも参照できるトピックがあります。以下のセクションを参照できます。

- 管理 IBM WebSphere MQ

IBM WebSphere MQ には、使用できる制御コマンドがあります。このシナリオでは、**crtmqm** および **strmqm** の 2 つのコマンドを使用します。このセクションでは、メッセージ・キューイングに関する優れた概要も提供されます。

- MQSC リファレンス

このシナリオでは、`define qlocal('LQ1')` コマンドを使用して、LQ1 という名前のローカル・キューを定義します。このコマンドは、MQSC コマンドです。IBM WebSphere MQ システム管理者は、これらのコマンドを使用してキュー・マネージャーを管理します。このセクションでは、コマンドの概要と使用方法について説明した後、コマンドの詳細についてアルファベット順に説明しています。

- キュー・マネージャー・クラスターの構成

このセクションでは、クラスターとして知られる仮想グループ内のキュー・マネージャーを編成、使用、および管理する方法について説明します。クラスタリングによって、クラスター内の各キュー・マネージャーは確実に同じクラスター内の他のすべてのキュー・マネージャーを認識します。また、クラスタリングによって、複雑なキュー・マネージャー・ネットワークの管理がより簡単になります。

Product Connectivity Scenarios 製品資料は、さまざまなシナリオにおいて WebSphere Application Server を WebSphere MQ に接続するのに必要な主要なタスクを実行するのを支援します。各シナリオには、ビジネス・コンテキストにおけるソリューションを実装するための指示が含まれています。これにより、他の情報リソースを使用しなくても学習を進めることができます。[https://www.ibm.com/docs/prodconn.1.0.0/com.ibm.prodconn.doc/infocenter\\_homepage/ic\\_home.htm](https://www.ibm.com/docs/prodconn.1.0.0/com.ibm.prodconn.doc/infocenter_homepage/ic_home.htm)

## スクリプトを使用する基本的なファイル転送

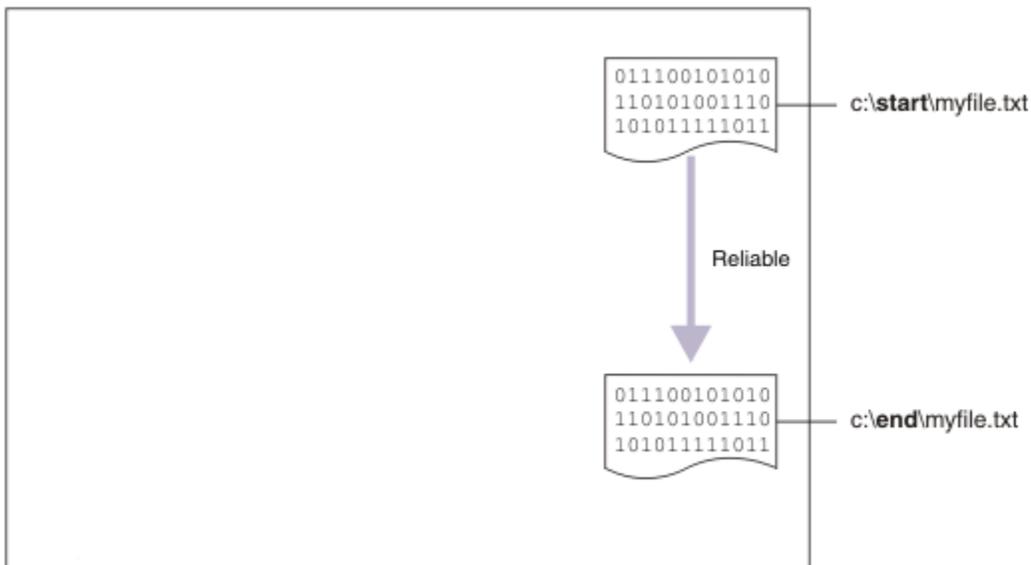
IBM WebSphere MQ Version 7.5 を使用して、さまざまな方法でファイルを転送できます。このセクションのトピックを参照して、このシナリオで網羅されている内容、ビジネスがこのシナリオに従うべき理由、関連するユーザー役割、およびこのシナリオで提案されるソリューションの概要を理解できます。

ファイル転送をコントロールし、信頼性を高め、監査機能を付けることは、企業にとって基本的な要件になる場合があります。IBM WebSphere MQ Version 7.5 統合メッセージング・プラットフォームの一部として Managed File Transfer 機能を提供します。Managed File Transfer 機能を使用すると、基本的なファイル転送によって、またはメッセージングへの本格的な参加者として、メッセージング・インフラストラクチャーへのファイルのシームレスな統合が可能になります。この機能の詳細については、[WebSphere MQ Managed File Transfer の概要](#)を参照してください。

最も単純な IBM WebSphere MQ メッセージング・トポロジーの中にファイルを統合する基本的な方法を理解するうえで、このシナリオが役立ちます。その目的に合わせて、ファイルを 1 つの場所から別の場所へ移動するように設計された基本的な IBM WebSphere MQ のシナリオを順番に説明します。この初期シナリオは 1 台のコンピューターに制限されていますが、環境を構成する経験を積むことができるので、この後のシナリオを実行するための重要な土台になります。この後のシナリオでは、IBM WebSphere MQ を使用してネットワーク経由でファイルを転送する方法をデモンストレーションした後、Managed File Transfer コンポーネントを実際のビジネス問題に適用する方法を紹介していきます。

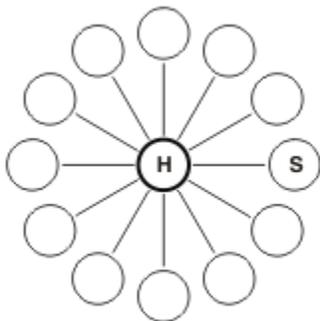
IBM WebSphere MQ の基本について理解しておく必要があります。具体的には、キュー・マネージャーの概念と、**runmqsc** や IBM WebSphere MQ Explorer などのコマンドを使用することによる IBM WebSphere MQ の基本的な構成および管理について理解しておく必要があります。

このシナリオでは、1 台のコンピューター上で IBM WebSphere MQ を使用して 1 つの場所から別の場所へのファイル転送を開始して追跡する方法を調べます。また、IBM WebSphere MQ Version 7.5 で信頼性の高い Managed File Transfer 機能をインストールし、構成し、利用する面で経験を積むことができます。



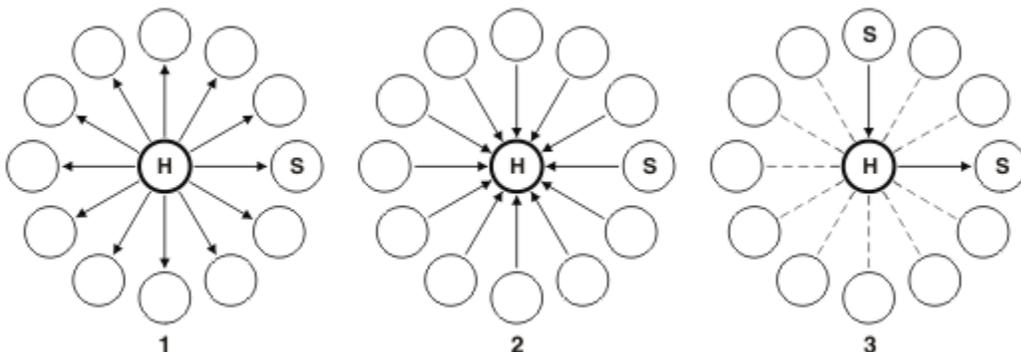
### ファイル転送トポロジーの例

このシナリオとこれ以降のシナリオは、ハブ・アンド・スポーク・トポロジーに基づくものです。次の図は、ハブ (H) と複数のスポーク (S) で構成される概念上のハブ・アンド・スポーク・トポロジーを示しています。



このようなトポロジーは多くのビジネス・シナリオのアプリケーションで目にします。例えば、次のような必要がある場合です。

1. 中央のHQ (ハブ) から多数の外側のノード (スポーク) にファイルを送信します。
2. 多数のノード (スポーク) から1箇所 (ハブ) にファイルを順番に集めます。
3. 1つのスポークから別のスポークにファイルを転送します。



現実のトポロジーはこの例とは異なることがありますが、このシナリオで紹介する原則とコマンドを応用すれば、どんな種類の IBM WebSphere MQ ネットワークにも対応するよう、迅速に拡張できます。

## ソリューションの計画

1 台のコンピューターとの間のスクリプトを使用したファイル転送を計画します。セキュリティー・モデルと前提条件とライセンスの理解、IBM WebSphere MQ のインストールと構成、およびユーザーとグループの準備が含まれます。

ポイント A からポイント B にファイルを転送するには、1 台のコンピューター上にハブ・アンド・スポーク・トポロジーを作成します。このトポロジーは、IBM WebSphere MQ キュー・マネージャーというハブと、各ファイル転送エージェントという 2 つのスポークで構成されます。

ファイル転送エージェントとは、コンピューター上で実行され、1 つのエージェントから他のエージェントにファイルを転送する Java プロセスを指します。このシナリオでは、これらのファイル転送エージェントを使用するファイル転送を定義し、次の 2 つのメカニズムでサンプル・ファイルを 1 つの場所から別の場所に移動します。

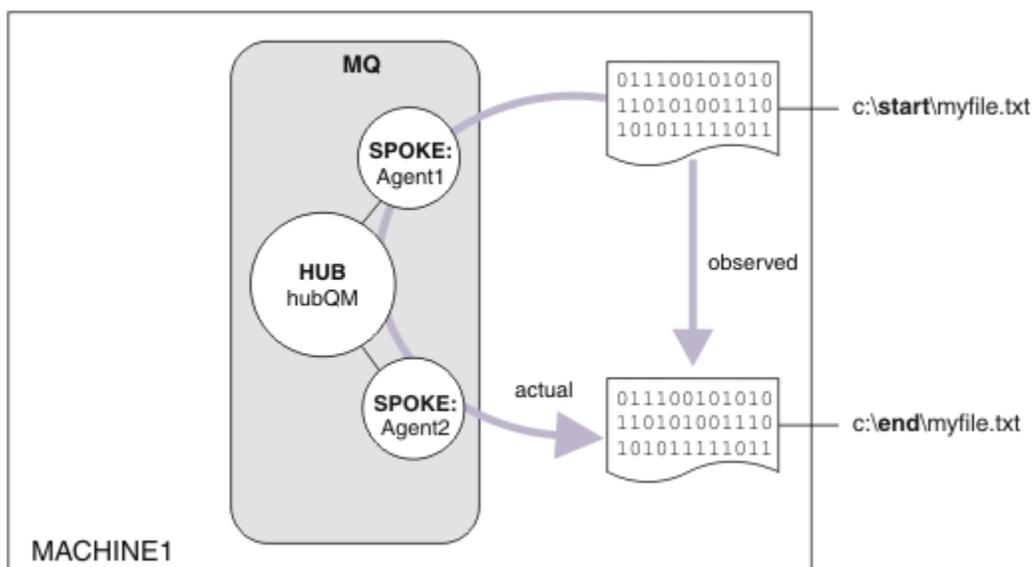
1. グラフィカルな IBM WebSphere MQ エクスプローラーを使用する方法
2. コマンド・ラインを使用する方法

このシナリオでは、Windows システムを使用していることを想定します。UNIX システムでは、該当するパスおよびコマンドに置き換えることと、関係するすべてのディレクトリーに対して読み取り権限と書き込み権限を持っていることが必要になります。また、このシナリオでは、ユーザーがキュー・マネージャーの動作について基本的に理解していることも前提としています。

あるコンピューターとの間でファイルを転送するには、そのコンピューター上でファイル転送エージェントを実行しておく必要があります。すべてのエージェントは、IBM WebSphere MQ キュー・マネージャーに接続し、IBM WebSphere MQ を使用して他のエージェントと通信します。詳細については、[WebSphere MQ Managed File Transfer トポロジーの概要](#)を参照してください。

IBM WebSphere MQ Version 7.5 にはサンプル・スクリプト createHub および addSpoke が付属していて、ファイル転送トポロジーを素早く簡単に構築するために役立ちます。これらのスクリプトを使用すると、このトポロジーをハブから外側へと作成できます。このスクリプトはサンプルなので、ご自分の要件に合わせて変更できます。スクリプトを変更する場合は、スクリプトをまず独自のユーザー・ディレクトリーにコピーしてください。

サンプル・スクリプト createHub および addSpoke は、トポロジーを構成するために使用します。



転送が開始されると、ファイルが 1 つの場所から別の場所にコピーされることが分かります。このシンプルなシナリオは、ソース・ファイルと宛先ファイルの両方を管理する 1 つのファイル転送エージェントを使用して達成することも可能です。しかし、複数コンピューターのトポロジーに進む前の準備として、敢

えて2つのファイル転送エージェントを使用する例を紹介し、このセットアップがどのように機能するかを説明します。

## セキュリティ・モデルの理解

createHub および addSpoke スクリプトによって構成されるファイル転送トポロジーのセキュリティ特性は次のとおりです。

- IBM WebSphere MQ へのアクセスは、次の3つの役割に区分されています。
  1. IBM WebSphere MQ 管理者。IBM WebSphere MQ を構成し、createHub および addSpoke スクリプトを実行します。
  2. FTAGENTS。エージェントを開始および停止し、エージェントと対話します。
  3. FTUSERS。ファイル転送を開始します。

アクセス権限の構成について詳しくは、[WebSphere MQ Managed File Transfer に固有のリソースに関する権限](#)を参照してください。

- エージェントからのすべての着信接続は、ハブ上の単一のユーザー FTAGENT にマップされます。
- 個々のエージェントの認証には、セキュリティ度の低い IP ベースの認証が使用されています。

理解しやすくする目的で、このスクリプトのセキュリティ・モデルは完全には強化されていません。これらのスクリプトを実稼働環境で使用する場合は、使用するトポロジーのニーズと、セキュリティ上の脅威について事前に理解しておく必要があります。そのため、次のような潜在的な脆弱性と推奨事項を考慮に入れるかどうかを検討してください。

- どのユーザーも他のユーザーとして偽名を使用できます。ファイル転送リソースのオブジェクト・アクセス・モデルの細分度を細かくすることを検討してください。
- どのエージェントも他のエージェントとして偽名を使用できます。より強力な認証 (例えば、TLS/SSL) を使用し、ファイル転送リソースのオブジェクト・アクセス・モデルの細分度を細かくすることを検討してください。
- ファイル・システムと IBM WebSphere MQ との間のインターフェースでセキュリティが考慮されていません。ファイルのサンドボックスを実装することを検討し、エージェントの構成ファイルのアクセス権が持つ影響を理解してください。サンドボックスの詳細については、[サンドボックス](#)を参照してください。
- エージェントとオペレーティング・システムとの間のインターフェースが記述されていません。ファイル・サンドボックスの実装を検討してください。詳細については、[サンドボックス](#)を参照してください。

セキュリティ、IBM WebSphere MQ、およびファイル転送の詳細については、[次に行うこと](#)を参照してください。

## 前提条件およびライセンス

以下の項目が必要になります。

- IBM WebSphere MQ のハードウェアおよびオペレーティング・システムの前条件を満たすテスト・コンピューター。詳しくは、IBM WebSphere MQ または IBM WebSphere MQ データの既存のインストール済み環境がない <https://www.ibm.com/support/docview.wss?uid=swg27006467> を参照してください。
- IBM WebSphere MQ Version 7.5 試用版は <https://www.ibm.com/developerworks/downloads/ws/wmq/> からダウンロードできます。

## コンピューターの準備

テスト・コンピューターが IBM WebSphere MQ Version 7.5 のインストール要件を満たしていることを確認してください。[要件のチェック](#)を参照してください。

ご使用のコンピューターがインストール用に適切に準備されていることを確認します。[コマンド行を使用したローカル・インストールの検査](#)を参照してください。

## IBM WebSphere MQ Version 7.5 のインストール

IBM WebSphere MQ Version 7.5 サーバーを、次のコンポーネントと一緒にインストールします: サーバー、IBM WebSphere MQ エクスプローラー、IBM WebSphere MQ Advanced Managed File Transfer エージェント、および IBM WebSphere MQ Managed File Transfer コマンド・ライン・ツール。詳細については、[インストール内容の選択](#)を参照してください。

IBM WebSphere MQ をどのように管理するかを決定します。IBM WebSphere MQ の管理には、次のような方法があります。

- **setmqenv** コマンドを使用して適切な環境をセットアップします。詳細については、[setmqenv](#) を参照してください。
- 完全修飾の IBM WebSphere MQ 管理コマンドを呼び出します。

このシナリオでは、以前に IBM WebSphere MQ または IBM WebSphere MQ File Transfer Edition をインストールしたことがない、クリーンなコンピュータを使用していることを想定しています。そうでない場合は、共存することがサポートされているかどうか、またはインストール・メカニズムを調整して環境を構成できるかどうかを調べておく必要があります。詳しくは、『[複数のインストール](#)』を参照してください。

## ユーザーおよびグループの作成

このセキュリティー・モデルは、以下のユーザーとグループが作成されていることを前提としています。

ユーザー:

- mqmAdmin

IBM WebSphere MQ 管理者。mqm グループのメンバーであるか、または Windows 上では Administrators グループのメンバーです。このユーザーを作成し、それを mqm グループのメンバーにする必要があります。

- ftuser

このユーザーを作成し、それを FTUSERS グループのメンバーにする必要があります。キュー・マネージャーに対して管理レベル・セキュリティーの操作を実行することがないようにするため、このユーザーは mqm グループに追加しないでください。

- ftagent

このユーザーを作成し、それを FTAGENTS グループのメンバーにする必要があります。キュー・マネージャーに対して管理レベル・セキュリティーの操作を実行することがないようにするため、このユーザーは mqm グループに追加しないでください。

Groups

- mqm

IBM WebSphere MQ インストールの一部として自動的に作成されます。このグループのメンバーは IBM WebSphere MQ とそのリソースを管理できます。

- FTUSERS

このグループは必ず作成してください。このグループのメンバーは、ファイル転送を開始できます。

- FTAGENTS

このグループは必ず作成してください。このグループのメンバーは、ネットワーク内でファイル転送を処理するエンドポイントとなるファイル転送エージェントを開始および停止できます。

## ファイル転送用の IBM WebSphere MQ の構成

スクリプト・シナリオを使用して基本的なファイル転送のトポロジーを作成するには、サンプル・スクリプト createHub および addSpoke を使用して、ファイル転送用に IBM WebSphere MQ を構成します。

## 手順

1. リモート・エージェント (後のシナリオで定義します) の接続先にできる空きポートを識別します (例: 1414)。
2. ファイル転送ネットワークのハブとして機能するキュー・マネージャーにふさわしい名前を決めます (例: hubQM)。
3. ユーザー mqmAdmin として、IBM WebSphere MQ bin ディレクトリー < MQ\_INSTALL\_ROOT >/bin から以下を行います。
  - a) ハブを作成します。

次のコマンドを入力します。

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\createHub hubQmgr=hubQM hubPort=1414
```

- b) 最初のエージェント・スポークを追加し (キュー・マネージャーに対してローカルなので BINDINGS)、このエージェントに AGENT1 という名前を付けます。

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\addSpoke agentName=AGENT1
hubQmgr=hubQM connectionMode=BINDINGS
```

- c) 2 番目のエージェント・スポークを追加し (キュー・マネージャーに対してローカルなので BINDINGS)、このエージェントに AGENT2 という名前を付けます。

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\addSpoke agentName=AGENT2
hubQmgr=hubQM connectionMode=BINDINGS
```

4. ユーザー ftagent として、以下の操作を行います。

- a) 次のコマンドを入力して AGENT1 を開始します。

```
fteStartAgent -p hubQM AGENT1
```

なお、複数のハブを構成した場合は、コマンドの -p hubQM の部分で正しいものを指定してください。ハブが 1 つの場合には、その必要はありません。

- b) 次のコマンドを入力して AGENT2 を開始します。

```
fteStartAgent -p hubQM AGENT2
```

- c) エージェントのリストを表示して、正常に稼働していることを確認します。

```
fteListAgents -p hubQM
```

次の出力が表示されます。

| Agent Name: | Queue Manager Name: | Status: |
|-------------|---------------------|---------|
| AGENT1      | hubQM               | READY   |
| AGENT2      | hubQM               | READY   |

## ソリューションの実装

このシナリオでのソリューションの実装には、IBM WebSphere MQ エクスプローラーを使用して、ファイル転送を定義して開始する操作が含まれます。転送プロセスをモニターし、ファイル・システムを調べることによって動作が正常かどうかを確認できます。

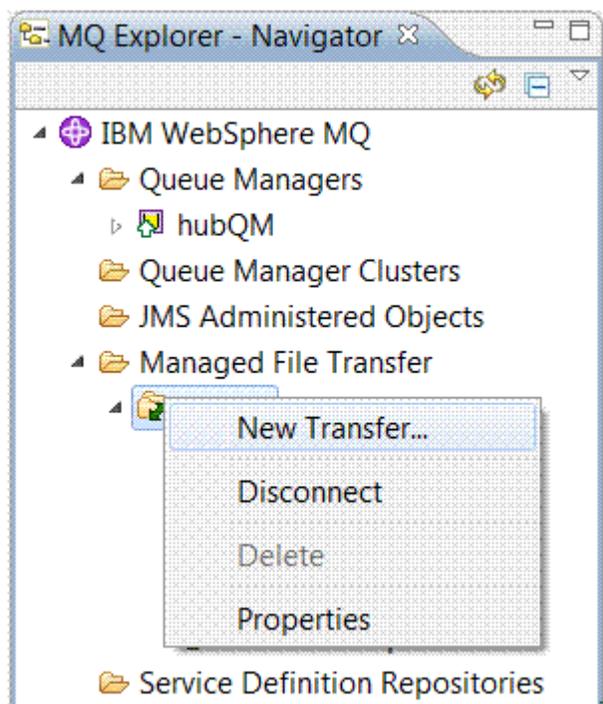
### このタスクについて

IBM WebSphere MQ エクスプローラーを非管理ユーザー ftuser 用に構成しないで済むようにするため、シナリオのこの部分はユーザー mqmAdmin として実行します。この機能を有効にする場合は、[セキュリティ](#)を参照してください。

## 手順

1. ソース・ファイルとターゲット・ディレクトリーを識別します。

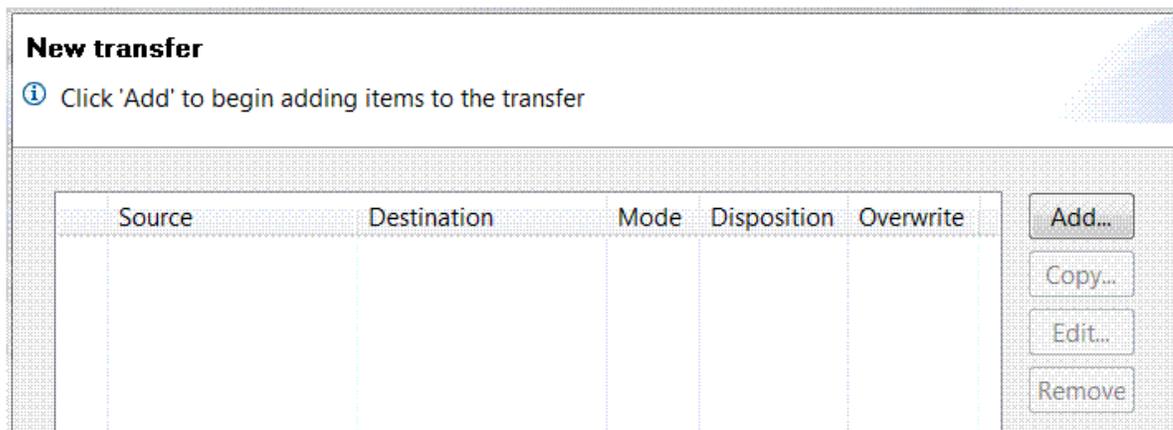
- a) 転送するサンプル・ファイルを作成します (例: C:\start\myfile.txt)。
- b) このファイルの転送先となる既存のディレクトリーを識別します (例: C:\end\ )。
2. 管理者 (この例では mqmAdmin) として IBM WebSphere MQ エクスプローラーを開始します。「スタート」メニュー (または同等の機能) からプログラムを開始するか、コマンド **MQExplorer** を実行します。詳細については、[WebSphere MQ エクスプローラーの起動](#)を参照してください。
3. 「ファイル転送管理」セクションで、hubQM という名前の構成を右クリックし、「接続」を選択します。
4. IBM WebSphere MQ ナビゲーターの「ファイル転送管理」セクションで、hubQM を右クリックし、「新規転送」を選択して「新規転送」ウィザードを開始します。



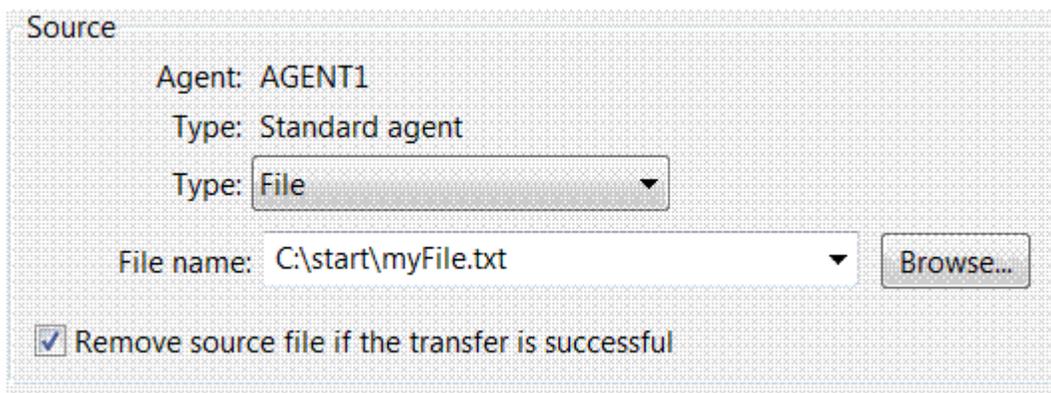
5. メニューを使用して、ソース・エージェントとして AGENT1 を、宛先エージェントとして AGENT2 を選択します。

|                   |          |
|-------------------|----------|
| Source agent      |          |
| Name:             | AGENT1   |
| Type:             | Standard |
| Destination agent |          |
| Name:             | AGENT2   |
| Type:             | Standard |

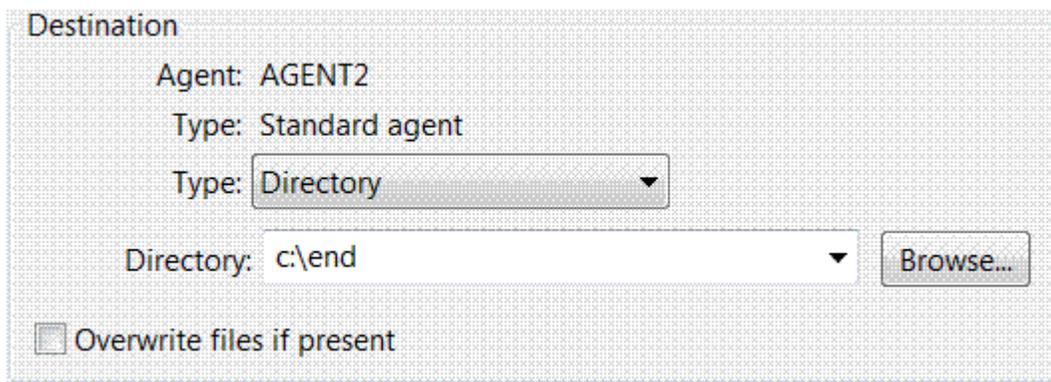
6. **次へ** をクリックします。



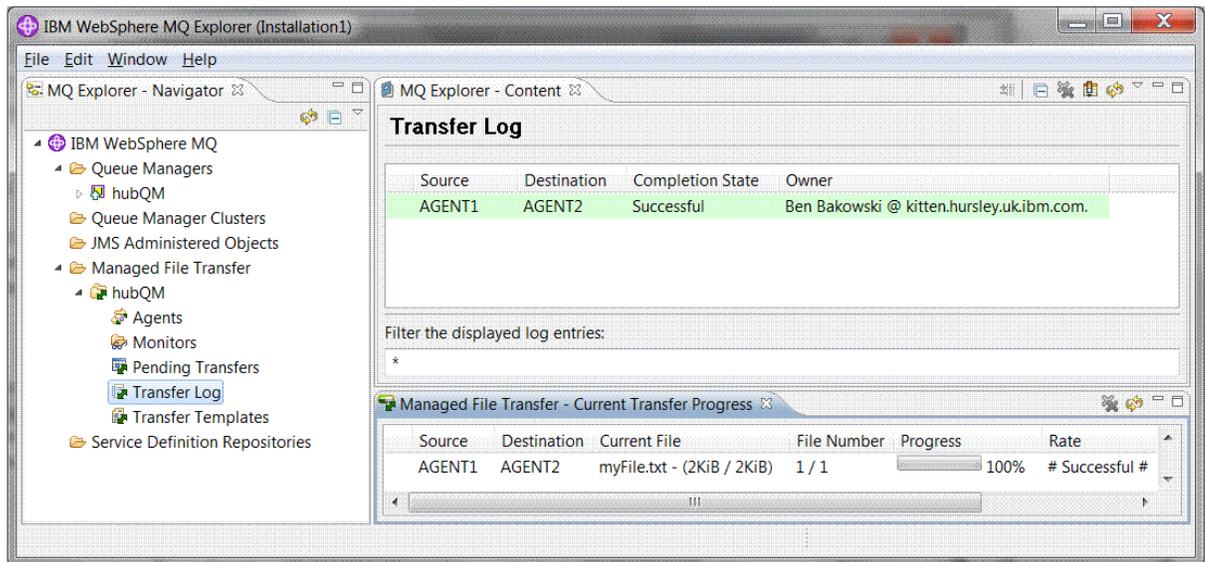
7. 「追加...」をクリックし、AGENT1 から AGENT2 に転送するファイルの選択を開始します。
8. 「ソース」フレームで、「参照」をクリックし、先ほど識別したファイル (例えば、C:\start\myFile.txt) を参照します。「転送が正常に行われた場合はソース・ファイルを削除」チェック・ボックスにチェック・マークを付けてあることを確認してください。



9. 「宛先」フレームで、「タイプ」として Directory を選択し、先ほど識別した宛先ディレクトリー (例えば、C:\end\)を入力します。



10. 「OK」をクリックしてから、「完了」をクリックして、転送を開始します。
11. IBM WebSphere MQ エクスプローラーの下部にある「現在の転送進行状況」タブで、転送の進行状況をモニターできます。



12. ファイル・システムを手動で検査して、新規ファイルが存在することを確認することもできます (例: C:\end\myfile.txt)。

## コマンド・ラインを使用してファイルを転送する

このセクションでは、ファイル転送の柔軟性を示す例として、コマンド・ラインを使用して転送を開始する方法を説明します。このシナリオの範囲外ではありますが、これらの原則および Ant スクリプティング・テクノロジーに基づいて、より強力なファイル転送シナリオの定義および実装を行えます。

### 手順

1. 転送したファイルを元の場所に返します。
2. ユーザー `ftuser` として `fteCreateTransfer` コマンドを入力し、`C:\start\myfile.txt` から `C:\end\myfile.txt` へのファイル転送を開始します。

```
fteCreateTransfer -sa AGENT1 -sm hubQM -sd delete -da AGENT2 -dm hubQM -w -dd C:\end\
C:\start\myfile.txt
```

- `-sa AGENT1` は、ソース・エージェント (つまり、ファイルの転送元のエージェント) を `AGENT1` に定義します。
- `-sm hubQM` は、ソース・エージェント `AGENT1` の接続先のキュー・マネージャーを定義します。
- `-sd delete` は、転送が正常に完了した後にソース・ファイルを削除することを指定します。
- `-da AGENT2` は、宛先エージェント (つまり、ファイルの転送先のエージェント) を `AGENT2` に定義します。
- `-w` は、コマンドの正常終了を確認するまで待機することを `fteCreateTransfer` コマンドに要求します。
- `-dd C:\end\` は、宛先ディレクトリーを `C:\end\` として定義します。
- `C:\start\myfile.txt` は、転送するファイルを定義します。

詳細については、[fteCreateTransfer \(新規ファイル転送の作成\)](#) を参照してください。

3. IBM WebSphere MQ エクスプローラーのファイル転送ログを調べるか、手動でファイル・システムを調べることによって、ファイル転送が成功したことを確認します。

### 次のタスク

外部のメディアを通じて、ファイル転送機能に関係したさらに多くのフィーチャーについて調べたいと思われるかもしれません。以下を参照してください。

- [163 ページの『スクリプトを使用する 2 台のコンピューターでのファイル転送』](#)

- ご使用の環境の機密保護機能をさらに高める。要件によっては、このシナリオで使用したものとは異なるアクセス・モデルが必要になることがあります。この分野でのベスト・プラクティスの詳細については、[Securing WebSphere MQ File Transfer Edition V7](#) を参照してください。

## 基本的なファイル転送の詳細

IBM WebSphere MQ Version 7.5 を使用して、さまざまな方法でファイルを転送できます。このセクションのトピックでは、このシナリオで説明されている内容、ビジネスがこのシナリオに従う理由、関連するユーザー・ロール、およびこのシナリオで提案されているソリューションの概要について説明します。

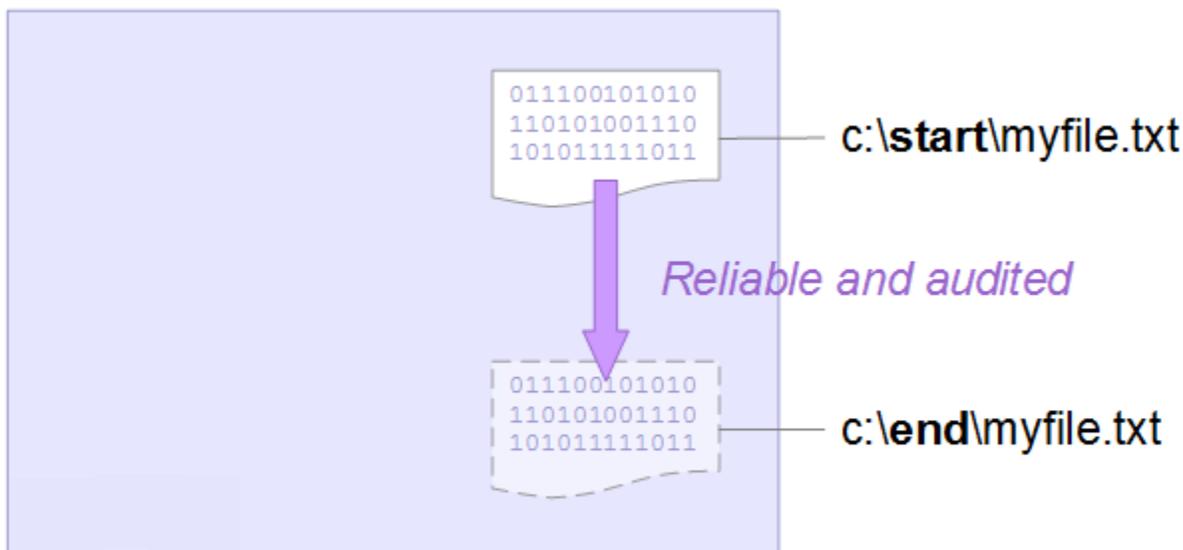
ファイル転送をコントロールし、信頼性を高め、監査機能を付けることは、企業にとって基本的な要件になる場合があります。IBM WebSphere MQ Version 7.5 統合メッセージング・プラットフォームの一部として Managed File Transfer 機能を提供します。Managed File Transfer 機能を使用すると、基本的なファイル転送によって、またはメッセージングへの本格的な参加者として、メッセージング・インフラストラクチャへのファイルのシームレスな統合が可能になります。

最も単純な IBM WebSphere MQ メッセージング・トポロジーの中にファイルを統合する基本的な方法を理解するうえで、このシナリオが役立ちます。ファイルを 1 つの場所から別の場所へ移動することを目的として設計された、この基本的な IBM WebSphere MQ シナリオを学習してください。この初期シナリオは 1 台のコンピューターに制限されていますが、環境を構成する経験を積むことができるので、この後のシナリオを実行するための重要な土台になります。このシナリオでは、IBM WebSphere MQ を使用してネットワーク上でファイルを転送する方法を示します。次に、Managed File Transfer コンポーネントを使用して実際の業務上の問題に取り組む方法を示します。

IBM WebSphere MQ の基本について理解しておく必要があります。具体的には、キュー・マネージャーの概念と、**runmqsc** や IBM WebSphere MQ Explorer などのコマンドを使用することによる IBM WebSphere MQ の基本的な構成および管理について理解しておく必要があります。

### 概要

IBM WebSphere MQ を使用して、単一コンピューター上での 1 つの場所から別の場所へのファイル転送を開始および追跡できます。このシナリオをとおして、IBM WebSphere MQ Version 7.5 での Managed File Transfer 機能のインストール、構成、および使用を体験できます。また、この機能を使用して、ファイル転送に関する実際の業務上の問題への取り組みを開始する方法を理解することもできます。



ファイル転送機能の計画に関する詳細については、[WebSphere MQ Managed File Transfer の概要](#)を参照してください。

## ソリューションの計画

基本的なファイル転送のシナリオにおける、1台のコンピューターとのファイル転送、ファイル転送エージェント、前提条件、ライセンス、コンピューターの準備、およびファイル転送のための IBM WebSphere MQ のインストールと構成。

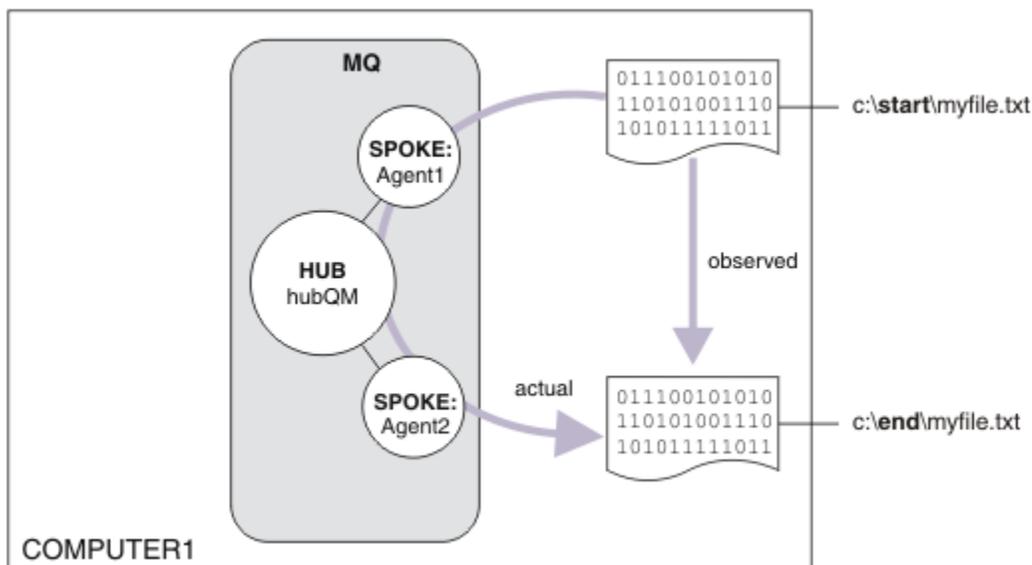
ポイント A からポイント B にファイルを転送するには、IBM WebSphere MQ キュー・マネージャーと 2 つのファイル転送エージェントを定義します。読者は既にキュー・マネージャーについて理解しているはずですが、ファイル転送エージェントという用語は初めて目にするかもしれません。ファイル転送エージェントとは、コンピューター上で実行され、1つのエージェントから他のエージェントにファイルを転送する Java プロセスを指します。このシナリオでは、これらのファイル転送エージェントを使用するファイル転送を定義し、次の 2 つのメカニズムでサンプル・ファイルを 1 つの場所から別の場所に移動します。

1. グラフィカルな IBM WebSphere MQ エクスプローラーを使用する方法
2. コマンド・ラインを使用する方法

ウォークスルーは、Windows システムがあることを前提としています。UNIX システムでは、該当するパスおよびコマンドに置き換えることと、関係するすべてのディレクトリーに対して読み取り権限と書き込み権限を持っていることが必要になります。また、キュー・マネージャーが実行する事柄についても基本的に理解していることを前提としています。

あるコンピューターとの間でファイルを転送するには、そのコンピューター上でファイル転送エージェントを実行しておく必要があります。すべてのエージェントは、IBM WebSphere MQ キュー・マネージャーに接続し、IBM WebSphere MQ を使用して他のエージェントと通信します。詳細については、[WebSphere MQ Managed File Transfer トポロジーの概要](#)を参照してください。

このシナリオでは、ソースおよび宛先の両方のファイルを管理する 1 つのファイル転送エージェントを使用できます。しかし、この例では、通常は複数のネットワークにまたがる実際の状況でファイル転送が機能する仕方をより現実的に理解できるように、2 つのファイル転送エージェントを使用します。



この図には、IBM WebSphere MQ を使用して、基本的なファイル転送を実行する方法が示されています。1 つの場所から別の場所にコピーされるファイルが監視されます。

## 前提条件およびライセンス

以下の項目が必要になります。

- IBM WebSphere MQ のハードウェアおよびオペレーティング・システムの前提条件を満たすテスト・コンピューター。詳しくは、IBM WebSphere MQ または IBM WebSphere MQ データの既存のインストール済み環境がない状態で、<https://www.ibm.com/support/docview.wss?uid=swg27006467> を参照してください。

- IBM WebSphere MQ Version 7.5 試用版は <https://www.ibm.com/developerworks/downloads/ws/wmq/> からダウンロードできます。

## コンピューターの準備

テスト・コンピューターが IBM WebSphere MQ Version 7.5 のインストール要件を満たしていることを確認します。要件の確認を参照してください。

ご使用のコンピューターが、インストールするために適切に準備されていることを確認してください。コマンド・ラインを使用したローカル・インストールの検査を参照してください。

## WebSphere MQ をファイル転送用に構成する

1. キュー・マネージャー hubQM を作成します。

コマンド `crtmqm hubQM` を入力します。詳しくは、[crtmqm](#) を参照してください。

2. キュー・マネージャー hubQM を開始します。

コマンド `strmqm hubQM` を入力します。詳細については、[strmqm](#) を参照してください。

3. キュー・マネージャー hubQM を構成し、ファイル転送を調整します。

- a. 次のコマンドを入力して、IBM WebSphere MQ のプロパティ・ファイルおよび調整キュー・マネージャーのディレクトリーを作成します。

```
fteSetupCoordination -coordinationQMgr hubQM
```

このコマンドによって、IBM WebSphere MQ のプロパティ・ファイルおよび調整キュー・マネージャーのディレクトリーが作成されます。この場合、hubQM は、監査およびファイル転送の情報をブロードキャストする調整キュー・マネージャーとして機能します。**fteSetupCoordination** コマンドを使用できない場合、デフォルト設定を使用して WebSphere MQ をインストールしたことを意味します。追加のパッケージ (例えば Managed File Transfer) をインストールすると、この問題が修正されます。

詳細については、[IBM WebSphere MQ Managed File Transfer トポロジーの概要および fteSetupCoordination \(調整セットアップの詳細\)](#) を参照してください。

- b. 次のコマンドを入力して、調整キュー・マネージャーとして機能するように hubQM を構成します。

```
runmqsc hubQM < <filepath from previous statement>
```

- c. ファイル転送コマンドを処理するキュー・マネージャー (この場合、hubQM) を定義します。

```
fteSetupCommands -connectionQMgr hubQM
```

詳細については、[fteSetupCommands \(command.properties ファイルの作成\)](#) を参照してください。

4. 最初のファイル転送エージェント AGENT1 を作成します。

- a. 次のコマンドを入力して、エージェントが接続するキュー・マネージャー (この場合、hubQM) に対して実行する必要がある MQSC スクリプトが組み込まれた、ファイル転送エージェント AGENT1 を作成します。

```
fteCreateAgent -agentName AGENT1 -agentQMgr hubQM
```

詳細については、[fteCreateAgent \(WebSphere MQ Managed File Transfer エージェントの作成\)](#) を参照してください。

- b. 作成したエージェントを処理するように hubQM を構成します。

```
runmqsc hubQM < <location of AGENT1_create.mqsc>
```

このファイルの場所は、IBM WebSphere MQ をインストールした場所によって異なります。

5. 2つ目のファイル転送エージェント AGENT2 を作成します。

```
fteCreateAgent -agentName AGENT2 -agentQMgr hubQM
runmqsc hubQM < <location of AGENT2_create.mqsc>
```

6. AGENT1 を開始します。

```
fteStartAgent AGENT1
```

詳細については、[fteStartAgent \(WebSphere MQ Managed File Transfer エージェントの開始\)](#) を参照してください。

7. AGENT2 を開始します。

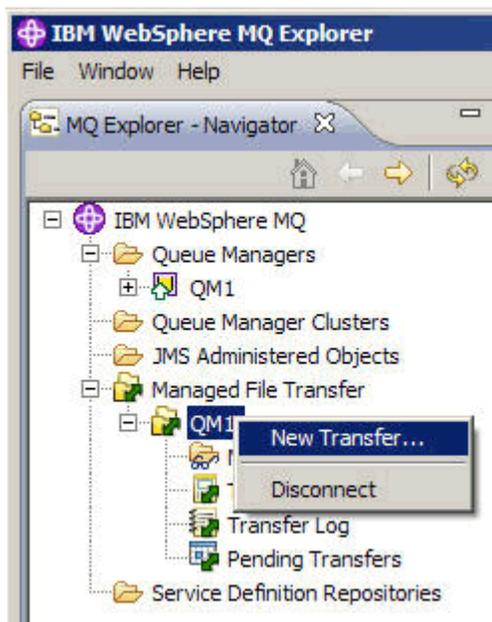
```
fteStartAgent AGENT2
```

## ソリューションの実装

このシナリオでソリューションを実装するには、IBM WebSphere MQ Explorer を使用してファイル転送を定義し、開始する必要があります。転送プロセスをモニターし、ファイル・システムを調べることによって動作が正常かどうかを確認できます。

### 手順

1. ソース・ファイルとターゲット・ディレクトリーを識別します。
  - a) 転送するサンプル・ファイルを作成します (例: C:\start\myfile.txt)。
  - b) このファイルの転送先となる既存のディレクトリーを識別します (例: C:\end\ )。
2. IBM WebSphere MQ Explorer を開始します。「スタート」メニューから (または同等の操作で) プログラムを開始するか、またはコマンド **MQExplorer** を実行します。詳しくは、[IBM WebSphere MQ Explorer の起動](#) を参照してください。
3. IBM WebSphere MQ Explorer ナビゲーターで「**ファイル転送管理**」をクリックし、**QM** を右クリックして「**新規転送**」を選択し、「新規転送」ウィザードを開始します。



4. 「**From**」セクションで、ソース・エージェントとして **AGENT1** を選択します。

From:

Agent: AGENT1

Type: File

File:

Include subdirectories

5. 先ほど作成したファイルへのパス (例えば、C:\start\myfile.txt) を入力します。

From:

Agent: AGENT1

Type: File

File: C:\Users\Ben Bakowski\Desktop\FTETEST\START\myDemoTransfer.txt

Include subdirectories

6. 「宛先」セクションで、「AGENT2」を宛先エージェントとして選択します。

7. 先ほど確認した宛先ディレクトリー (例えば、C:\end\) を入力します。

To:

Agent: AGENT2

Type: File

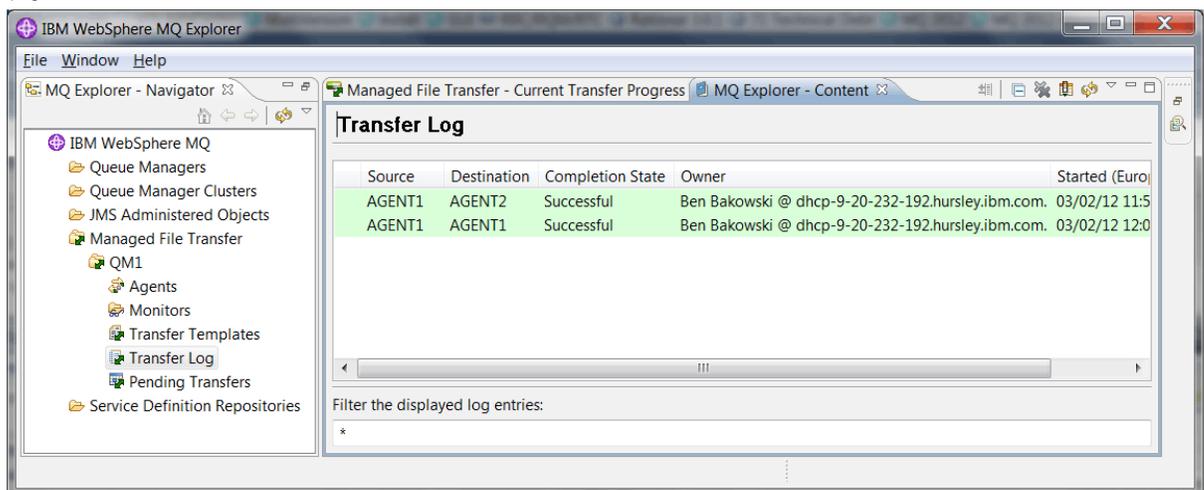
Directory: C:\Users\Ben Bakowski\Desktop\FTETEST\END

File name: myDemoTransfer.txt

Overwrite files on the destination file system that have the same name

8. 「すぐに終了」をクリックして、転送を開始します。

9. IBM WebSphere MQ Explorer の「現在の転送進行状況」タブで、転送の進行状況をモニターできます。



10. ファイル・システムを手動で検査して、新規ファイルが存在することを確認することもできます (例: C:\end\myfile.txt)。

## コマンド・ラインを使用してファイルを転送する

転送を開始する方法としてコマンド・ラインを使用すると、ファイル転送の柔軟性を例示することができます。このシナリオの範囲外ではありますが、これらの原則および Ant スクリプティング・テクノロジーに基づいて、より強力なファイル転送シナリオの定義および実装を行えます。

### 手順

1. 以前のデモンストレーションで転送したファイル (C:\end\myfile.txt など) を削除します。
2. **fteCreateTransfer** コマンドを使用して、C:\start\myfile.txt から C:\end\myfile.txt へのファイル転送を開始します。

```
fteCreateTransfer -sa AGENT1 -sm hubQM -da AGENT2 -dm hubQM -w -dd C:\end\
C:\start\myfile.txt
```

- -sa AGENT1 は、ソース・エージェント (つまり、ファイルの転送元のエージェント) を AGENT1 に定義します。
- -sm hubQM は、ソース・エージェント AGENT1 の接続先のキュー・マネージャーを定義します。
- -da AGENT2 は、宛先エージェント (つまり、ファイルの転送先のエージェント) を AGENT2 に定義します。
- -w は、コマンドの正常終了を確認するまで待機することを **fteCreateTransfer** コマンドに要求します。
- -dd C:\end\ は、宛先ディレクトリーを C:\end\ として定義します。
- C:\start\myfile.txt は、転送するファイルを定義します。

詳細については、[fteCreateTransfer \(新規ファイル転送の作成\)](#) を参照してください。

3. IBM WebSphere MQ Explorer のファイル転送ログを調べるか、手動でファイル・システムを調べることによって、ファイル転送が成功したことを確認します。

### 次のタスク

外部のメディアを通じて、ファイル転送機能に関係したさらに多くのフィーチャーについて調べたいと思われるかもしれません。以下を参照してください。

- [163 ページの『スクリプトを使用する 2 台のコンピューターでのファイル転送』](#)
- ご使用の環境の機密保護機能をさらに高める。要件によっては、このシナリオで使用したのものとは異なるアクセス・モデルが必要になることがあります。この分野でのベスト・プラクティスに関する情報は、[https://www.ibm.com/developerworks/websphere/library/techarticles/0902\\_wyatt/0902\\_wyatt.html](https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html) を参照してください。

## スクリプトを使用する 2 台のコンピューターでのファイル転送

スクリプトのシナリオを使用する基本的なファイル転送を拡張し、複数コンピューターで構成される IBM WebSphere MQ メッセージング・トポロジーにファイル転送を統合します。

ここまでで、Managed File Transfer 機能のシンプルなデモンストレーション ([149 ページの『スクリプトを使用する基本的なファイル転送』](#)を参照) を構築しましたので、ファイル転送管理に関する基本的な原則に慣れてきたはずです。コンピューターが 1 台の場合、この機能にはほとんどメリットがないことが分かります。そこで次に、このシナリオを拡張して、複数コンピューターによる IBM WebSphere MQ メッセージング・トポロジーにファイル転送を統合する方法を調べていきましょう。

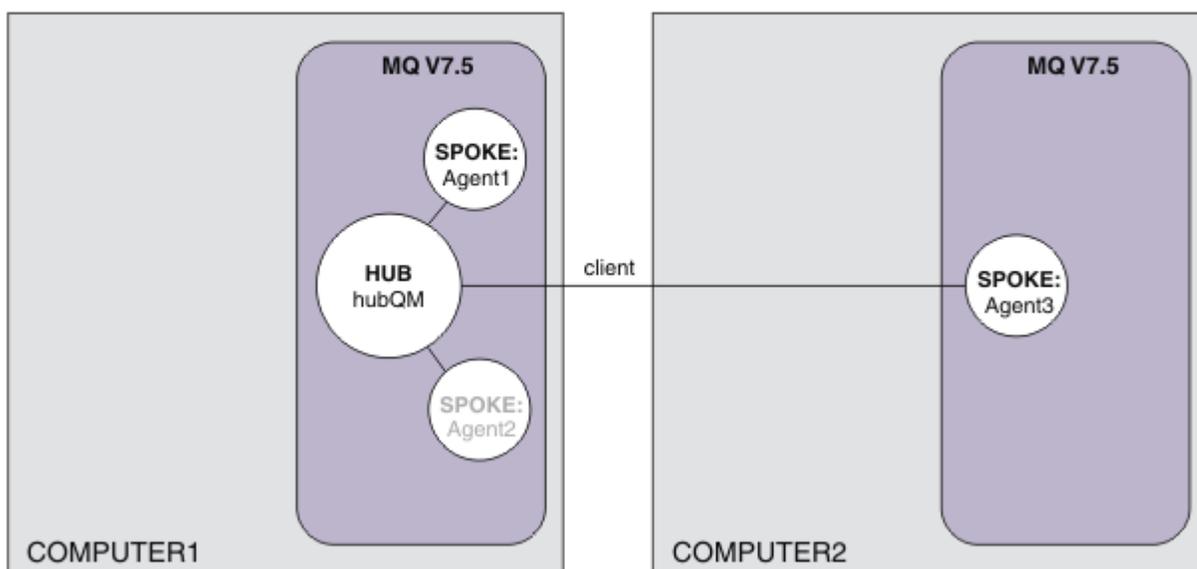
ここで初めて、基礎となる IBM WebSphere MQ テクノロジーが提供するシステム、つまり信頼性が高く、1 回ずつに限りファイルが送達されるシステムのメリットを実感してください。これを実現するために、ファイル転送に参加する 2 番目のコンピューターを組み込んでトポロジーを拡張します。このシナリオでは、別個に利用可能な Managed File Transfer エージェントをインストールして構成します。こうして、複数コンピューターによるファイル転送トポロジーに適用されるセキュリティー上の考慮点について少しずつ理解を深めていくことができます。このシナリオの最後では、1 つのコンピューターから次のコンピューターへのファイル転送をデモンストレーションします。これは、監査機能を追加するという次の

シナリオに向けて強力な土台を据えることにつながり、なぜこれをファイル転送管理機能と呼ぶのかを理解できます。

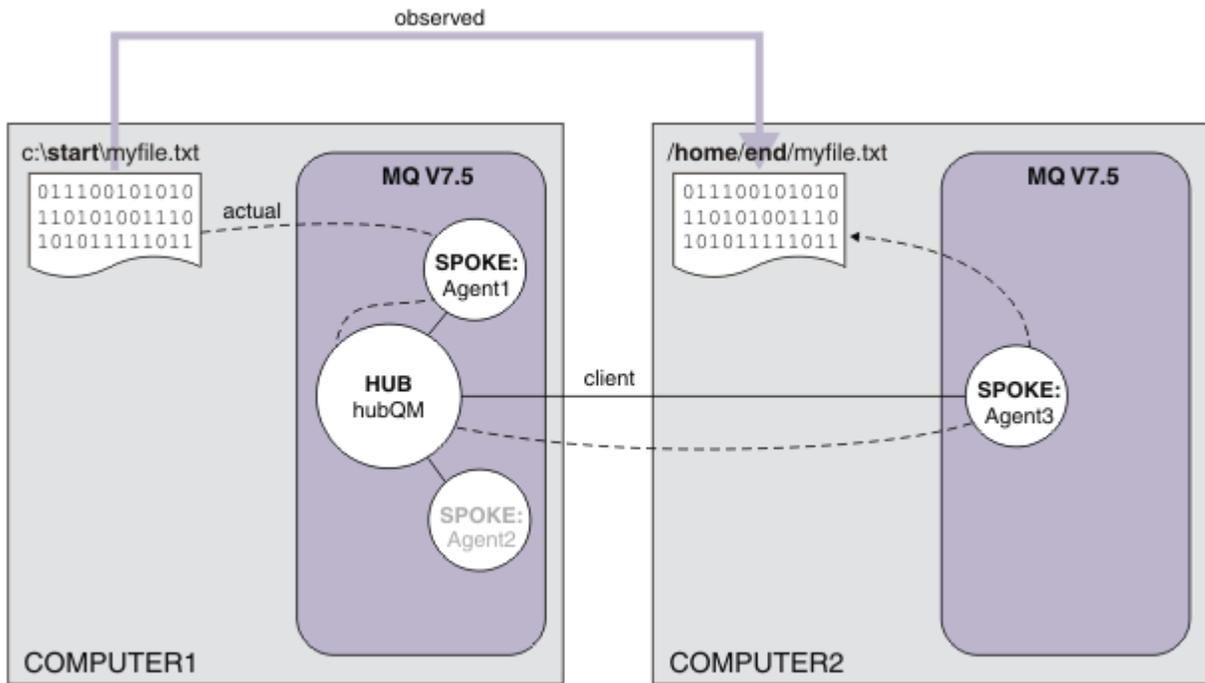
前提条件は、[149 ページの『スクリプトを使用する基本的なファイル転送』](#)の作業を完了していることです。

## 概要

このシナリオでは、[149 ページの『スクリプトを使用する基本的なファイル転送』](#)シナリオで構成されたハブ・スポーク・トポロジーが拡張され、2 台目のコンピューターであるコンピューター 2 が組み込まれます。コンピューター 2 には、Managed File Transfer エージェントとコマンド・ライン・ツールがインストールされています。この 2 番目のコンピューターには、ローカル IBM WebSphere MQ サーバーのインストールの前提条件はありません。このようなモデルは、ハブ・アンド・スポーク・ソリューションでよく利用される形態です。つまり、複数のファイル転送管理エージェントが中央の IBM WebSphere MQ キュー・マネージャーとクライアント接続上で通信を行い、各スポークには IBM WebSphere MQ サーバーのインストールとそれに関連するライセンスが不要であるというモデルです。他のトポロジーもサポートされており、ニーズに合わせてトポロジーを構築できますが、それぞれのトポロジーに応じて追加のライセンスや資格が必要になります。コンピューター 1 では AGENT1 だけを使用して転送を処理するために、AGENT2 は廃止します。



この図は、2 台のコンピューターにまたがるハブ・アンド・スポーク・トポロジーを示しています。AGENT1 と AGENT2 は (ローカル) ハブへのバインディングを使用するように構成されているのに対して、AGENT3 はクライアント接続を介して接続しています。AGENT2 は、このシナリオでは特に必要ないので、使用不可にします。AGENT1 は、コンピューター 1 上のすべてのファイル転送アクティビティを処理します。セキュリティー・モデルとその制限について理解しておいてください。詳しくは、[149 ページの『スクリプトを使用する基本的なファイル転送』](#)シナリオを参照してください。構成が終わると、このトポロジーを使用してコンピューター 1 からコンピューター 2 にファイルを転送できるようになります。



この図は、デモンストレーションするファイル転送経路を示しています。この場合も、基礎となるファイル転送は、信頼性の高い、アクティブな IBM WebSphere MQ 接続上で実行されます。この例では、コンピューター 1 が Windows コンピューターで、コンピューター 2 が Linux コンピューターであるとしてます。別のプラットフォームやアーキテクチャーを使用する場合は、サポートされるプラットフォームの完全なリストについて <https://www.ibm.com/support/docview.wss?uid=swg27006467#7.1> を参照してください。

## ソリューションの計画

1 台のコンピューターとのファイル転送、ファイル転送エージェント、前提条件、ライセンス、コンピューターの準備、およびスクリプト・シナリオを使用した 2 台のコンピューターのファイル転送のための IBM WebSphere MQ の構成を説明します。

## 前提条件

以下の項目が必要になります。

- コンピューター 1、149 ページの『スクリプトを使用する基本的なファイル転送』からの作業用構成。
- コンピューター 2、IBM WebSphere MQ Version 7.5 のハードウェアおよびオペレーティング・システムの前提条件を満たす第 2 のテスト・コンピューター。詳しくは、<https://www.ibm.com/support/docview.wss?uid=swg27006467> を参照してください。
- IBM WebSphere MQ Version 7.5 試用版は <https://www.ibm.com/developerworks/downloads/ws/wmq/> からダウンロードできます。
- コンピューター 1 とコンピューター 2 の IP アドレスについての知識 (nnn.nnn.nnn.nnn の形式)。

## IBM WebSphere MQ Version 7.5 のインストール

IBM WebSphere MQ Version 7.5 サーバーを、次のコンポーネントと一緒にインストールします: サーバー、IBM WebSphere MQ エクスプローラー、IBM WebSphere MQ Advanced Managed File Transfer エージェント、および IBM WebSphere MQ Managed File Transfer コマンド・ライン・ツール。詳細については、[インストール内容の選択](#)を参照してください。

IBM WebSphere MQ をどのように管理するかを決定します。IBM WebSphere MQ の管理には、次のような方法があります。

- **setmqenv** コマンドを使用して適切な環境をセットアップします。詳細については、[setmqenv](#) を参照してください。
- 完全修飾の IBM WebSphere MQ 管理コマンドを呼び出します。

## ユーザーおよびグループの準備

このセキュリティー・モデルは、次に挙げるグループおよびユーザーがあることを想定しています。

### グループ

- mqm

IBM WebSphere MQ インストールの一部として自動的に作成されます。このグループのメンバーは IBM WebSphere MQ とそのリソースを管理できます。

- FTUSERS

このグループは必ず作成してください。このグループのメンバーは、ファイル転送を開始できます。

- FTAGENTS

このグループは必ず作成してください。このグループのメンバーは、ネットワーク内でファイル転送を処理するエンドポイントとなるファイル転送エージェントを開始および停止できます。

### ユーザー:

- mqmAdmin

IBM WebSphere MQ 管理者。mqm グループのメンバーであるか、または Windows 上では Administrators グループのメンバーです。

- ftuser

このユーザーを作成し、それを FTUSERS グループのメンバーにする必要があります。このユーザーは mqm グループに追加しないようにしてください。キュー・マネージャーに対して管理セキュリティーの操作を実行するという可能性をなくすためです。

- ftagent

このユーザーを作成し、それを FTAGENTS グループのメンバーにする必要があります。このユーザーは mqm グループに追加しないようにしてください。キュー・マネージャーに対して管理セキュリティーの操作を実行するという可能性をなくすためです。

## コンピューター 2 上のスポークとして新しいエージェント AGENT3 を追加する

スクリプトを使用した IBM WebSphere MQ 2 コンピューター間ファイル転送シナリオのために、拡張トポロジーをサポートするよう、コンピューター 2 を準備します。

### このタスクについて

コンピューター 1 から再度 addSpoke サンプル・スクリプトを使用し、クライアント接続を通じて接続するエージェント・スポークを定義します。**addSpoke** コマンドの終了時には、コンピューター 2 上の IBM WebSphere MQ のインストール・システムから実行する一連のコマンドが示されます。

### 手順

1. コンピューター 1 で、ユーザー mqmAdmin として、IBM WebSphere MQ bin ディレクトリー <MQ\_INSTALL\_ROOT>\bin から **addspoke** コマンドを実行します。

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\addSpoke agentName=AGENT3
hubQmgr=hubQM connectionMode=CLIENT agentIPAddress=<IP address of computer 2>
hubIPAddress=<IP address of computer 1> hubPort=1414
```

2. コンピューター 2 上でユーザー mqmAdmin として、コンピューター 1 からの **addSpoke** コマンドからのコマンド出力を実行します。例えば、

- a) `fteSetupCoordination -coordinationQMgr hubQM-coordinationQMgrHost <IP address of computer 1> -coordinationQMgrPort 1414 -coordinationQMgrChannel FTE.USER.SVRCONN -f`
- b) `fteSetupCommands -p hubQM -connectionQMgr hubQM -connectionQMgrHost <IP address of computer 1> -connectionQMgrPort 1414 -connectionQMgrChannel FTE.USER.SVRCONN -f`
- c) `fteCreateAgent -p hubQM -agentName AGENT3 -agentQMgr hubQM -agentQMgrHost <IP address of computer 1> -agentQMgrPort 1414 -agentQMgrChannel FTE.AGENT.SVRCONN -f`

なお、生成される MQSC スクリプトを実行する必要はありません。

3. コンピューター 2 上でユーザー `ftagent` として `AGENT3` を始動します。

```
fteStartAgent -p hubQM AGENT3
```

4. コンピューター 2 上でユーザー `ftagent` として、3 つのエージェントが使用可能になっていることを確認します (必要ならコンピューター 1 上でユーザー `ftagent` としてエージェントを開始します)。

```
fteListAgents -p hubQM
```

次の出力が表示されます。

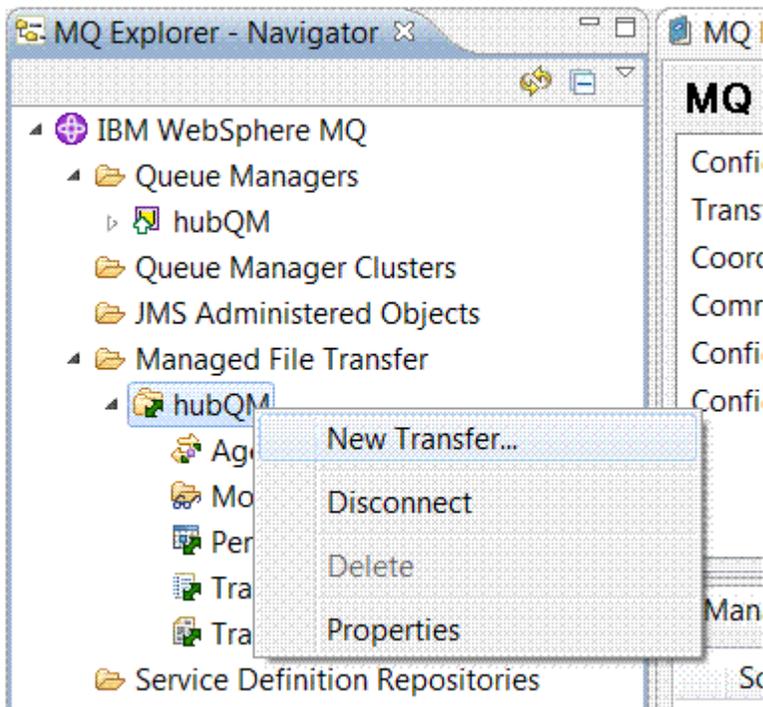
| Agent Name: | Queue Manager Name: | Status: |
|-------------|---------------------|---------|
| AGENT1      | hubQM               | READY   |
| AGENT2      | hubQM               | READY   |
| AGENT2      | hubQM               | READY   |

## ソリューションの実装

ファイルの転送に IBM WebSphere MQ を使用することにより、スクリプト・シナリオを使用して、2 コンピューター間ファイル転送のデモンストレーションを開始します。転送の進行状況をモニターし、ファイル・システムを調査することによって、それが正常に実行されたかどうかを確認します。

## 手順

1. ソース・ファイルとターゲット・ディレクトリーを識別します。
  - a) 転送するサンプル・ファイルをコンピューター 1 上に作成します (例: `C:\start\myfile.txt`)。
  - b) コンピューター 2 上で、このファイルの転送先となる既存のディレクトリーを決めます (例: `C:\end\`)。ユーザー `ftagent` に、そのディレクトリーへの書き込み権限が付与されていることを確認してください。
2. コンピューター 1 でユーザー `mqmAdmin` として IBM WebSphere MQ エクスプローラーを開始します。前のシナリオと同様に、これは IBM WebSphere MQ 管理者として実行します。これは、IBM WebSphere MQ エクスプローラーの構成ではなく、ファイル転送に関するシナリオに注目するためです。「スタート」メニューから (または同等の操作で) プログラムを開始するか、またはコマンド **MQExplorer** を実行します。詳細については、[WebSphere MQ エクスプローラーの起動](#)を参照してください。
3. IBM WebSphere MQ ナビゲーターの「**ファイル転送管理**」セクションで、`hubQM` を右クリックし、「**新規転送**」を選択して「**新規転送**」ウィザードを開始します。



4. メニューを使用して、ソース・エージェントとして AGENT1 を選択し、宛先エージェントとして AGENT3 を選択します。

5. 「次へ」をクリックし、「追加 ...」をクリックします。ソースの場合は、転送するファイルへのパスを入力します(例: C:\start\myfile.txt)。「転送が正常に行われた場合はソース・ファイルを削除」チェック・ボックスにチェック・マークが付いていることを確認します。

6. 宛先ディレクトリーを入力します。このシナリオの場合、この宛先は Linux プラットフォーム上にあるため、それにふさわしい表記として /home/end/ を使用します。

Destination

Agent: AGENT3

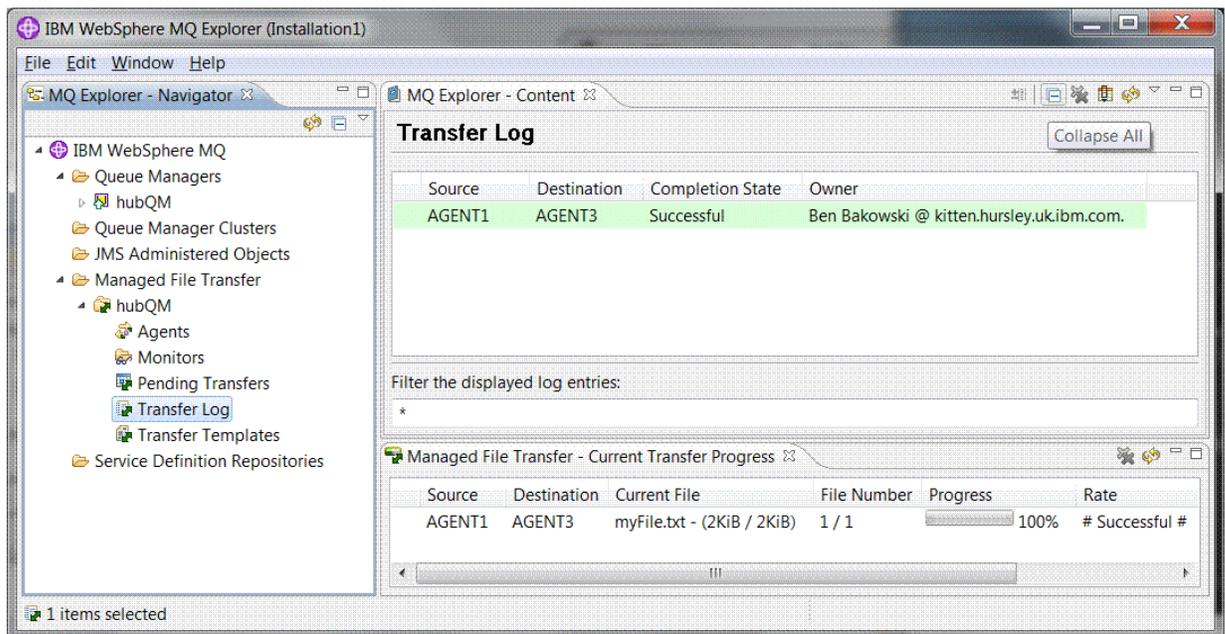
Type: Standard agent

Type:

Directory:

Overwrite files if present

7. 「OK」をクリックしてから、「完了」をクリックして、転送を開始します。
8. IBM WebSphere MQ エクスプローラーの「現在の転送進行状況」タブで転送の進行状況をモニターできます。



9. ファイル・システムを手動で調べることで、新規ファイル (/home/end/myfile.txt など) が存在していることを確認することもできます。

## コマンド・ラインを使用してファイルを転送する

転送を開始する方法としてコマンド・ラインを使用すると、ファイル転送の柔軟性を例示することができます。このシナリオの範囲外ではありますが、これらの原則および Ant スクリプティング・テクノロジーに基づいて、より強力なファイル転送シナリオの定義および実装を行えます。

### 手順

1. 前のデモンストレーションで転送したファイル (例: /home/end/myfile.txt) を削除し、その元の場所にそのファイルを再作成します。
2. ftuser を使用する場合は、**fteCreateTransfer** コマンドを使用して、コンピューター 1 上の C:\start\myfile.txt からコンピューター 2 上の /home/end/myfile.txt へのファイル転送を開始します。

```
fteCreateTransfer -sa AGENT1 -sm hubQM -da AGENT3 -dm hubQM -w -dd "/home/end/"
"C:\start\myfile.txt"
```

3. IBM WebSphere MQ エクスプローラーのファイル転送ログを調べるか、手動でファイル・システムを調べることによって、ファイル転送が成功したことを確認します。

## 次のタスク

IBM WebSphere MQ セキュリティーについて詳しくは、[セキュリティー](#)を参照してください。

外部のメディアを通じて、ファイル転送機能に関係したさらに多くのフィーチャーについて調べたいと思われるかもしれません。

- ファイル転送管理の管理という側面を提供する監査機能の追加について詳しくは、178 ページの『[ファイル転送管理への監査機能の追加](#)』を参照してください。
- 以下の事項に関する詳しい情報は、
  - トリガー: 新しいファイルを検出した時点で移動する。
  - トリガー: 1つのファイルを検出した時点で複数のファイルの転送を開始するように構成する。
  - Apache Ant を使用して転送のスクリプトを記述する。

[https://www.ibm.com/developerworks/websphere/library/techarticles/1003\\_phillips/1003\\_phillips.html](https://www.ibm.com/developerworks/websphere/library/techarticles/1003_phillips/1003_phillips.html) を参照してください。

- 要件によっては、このシナリオで使用したものと異なるアクセス・モデルが必要になることがあります。使用環境の機密保護機能を高めるためのベスト・プラクティスの詳細については、[https://www.ibm.com/developerworks/websphere/library/techarticles/0902\\_wyatt/0902\\_wyatt.html](https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html) を参照してください。

## 2 台のコンピューターでのファイル転送の詳細

基本的なファイル転送のシナリオを拡張し、ファイル転送を複数コンピューターの IBM WebSphere MQ メッセージング・トポロジーに統合します。

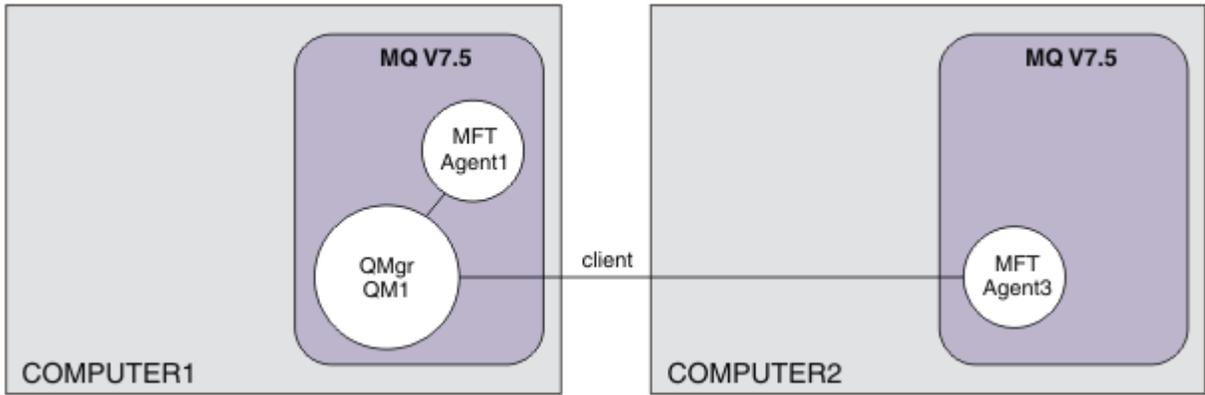
ここまでで、Managed File Transfer 機能のシンプルなデモンストレーション (158 ページの『[基本的なファイル転送の詳細](#)』を参照) を完了しましたので、ファイル転送管理に関する基本的な原則に慣れてきたはずです。コンピューターが 1 台の場合、この機能にはほとんどメリットがないことが分かります。そこで次に、このシナリオを拡張して、複数コンピューターによる IBM WebSphere MQ メッセージング・トポロジーにファイル転送を統合する方法を調べていきましょう。

ここで初めて、基礎となる IBM WebSphere MQ テクノロジーが提供するシステム、つまり信頼性が高く、1 回ずつに限りファイルが送達されるシステムのメリットを実感してください。これを実現するために、ファイル転送に参加する 2 番目のコンピューターを組み込んでトポロジーを拡張します。このシナリオでは、別個に利用可能な Managed File Transfer エージェントをインストールして構成します。こうして、複数コンピューターによるファイル転送トポロジーに適用されるセキュリティー上の考慮点について少しずつ理解を深めていくことができます。このシナリオの最後では、1つのコンピューターから次のコンピューターへのファイル転送をデモンストレーションします。これは、監査機能を追加するという次のシナリオに向けて強力な土台を据えることにつながり、なぜこれをファイル転送管理機能と呼ぶのかを理解できます。

前提条件は、158 ページの『[基本的なファイル転送の詳細](#)』の作業を完了していることです。

### 概要

このシナリオでは、基本的なファイル転送のシナリオで構成した、既存の Windows コンピューターを引き続き使用します。この最初のコンピューターのファイル転送エンドポイントとして単一のエージェントを使用するため、Agent2 を無効にします。Managed File Transfer エージェントを 2 台目のコンピューターにインストールします。この 2 台目のコンピューターには、ローカル IBM WebSphere MQ サーバーをインストールしておく必要はありません。このようなモデルは、ハブ・アンド・スポーク・ソリューションでよく利用される形態です。つまり、複数のファイル転送管理エージェントが中央の IBM WebSphere MQ キュー・マネージャーとクライアント接続上で通信を行い、各スポークには IBM WebSphere MQ サーバーのインストールとそれに関連するライセンスが不要であるというモデルです。他のトポロジーもサポートされており、ニーズに合わせてトポロジーを構築できますが、それぞれのトポロジーに応じて追加のライセンスや資格が必要になります。

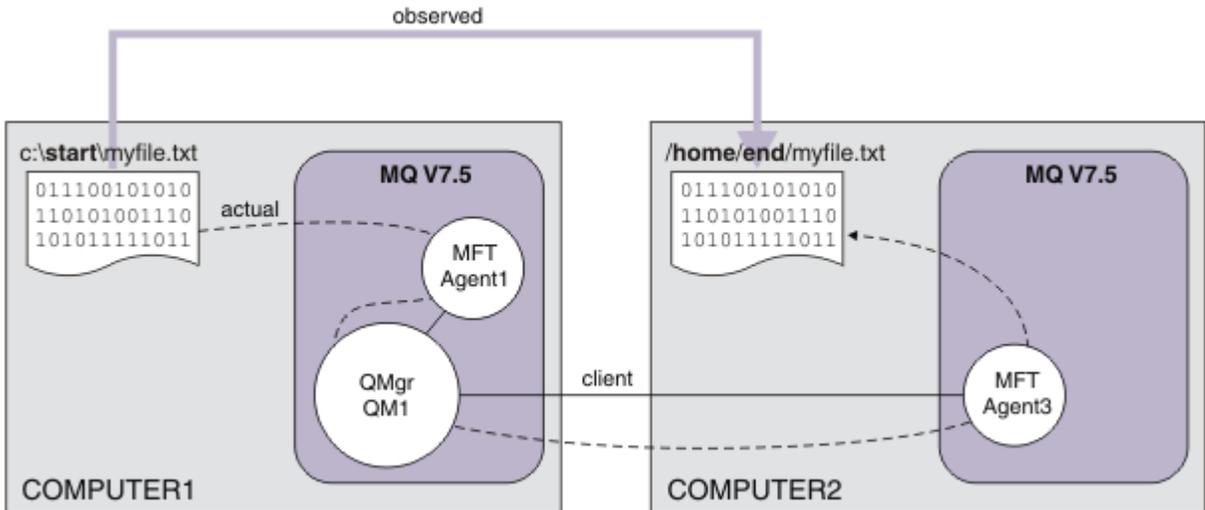


ファイル転送ネットワークでは、監査情報およびファイル転送情報をブロードキャストするためにネットワーク上の1点として機能する、単一のキュー・マネージャーを選択します。以前の158ページの『基本的なファイル転送の詳細』シナリオでは、単一のキュー・マネージャーが調整キュー・マネージャーとして暗黙的に使用されていました。このシナリオでは、コンピューター1でキュー・マネージャーQM1を調整キュー・マネージャーとして引き続き使用します。また、コンピューター2の構成の一部が、このキュー・マネージャーを使用するように環境を設定します。

このトポロジーでは、コンピューター2からファイル転送を開始する機能が必要ないため、オプションのManaged File Transfer コマンド・ライン・ツールのコンポーネントをこの2台目のコンピューターにインストールしないでください。

基本的なセキュリティー機能が多少は考慮されますが、ファイル転送トポロジーは適切と思われるレベルでは保護されないことに注意する必要があります。IBM WebSphere MQ の保護、特にファイル転送については、[https://www.ibm.com/developerworks/websphere/library/techarticles/0902\\_wyatt/0902\\_wyatt.html](https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html) を参照してください。

2台目のコンピューターにインストールするときに、実際のエージェント Agent3 を作成し、コンピューター1からコンピューター2にファイル移動するソリューションを実際に行います。



基礎となるファイル転送は、信頼性の高い IBM WebSphere MQ 接続を介して実行されます。続くタスクでは、これらの接続を構成する方法、および適切に保護する方法を示します。

この例では、コンピューター1はWindowsコンピューターであり、コンピューター2はLinuxコンピューターであるとして。別のプラットフォームやアーキテクチャーを使用する場合は、サポートされるプラットフォームの完全なリストについて <https://www.ibm.com/support/docview.wss?uid=swg27006467#7.1> を参照してください。

## ソリューションの計画

2台のコンピューターによるファイル転送ソリューションの計画。これには、適切なインフラストラクチャーについての説明、および作成する必要のあるグループとユーザーについての説明も含まれます。

### 始める前に

以下の項目が必要になります。

- 基本的なファイル転送のシナリオからの作業用構成。詳細については、[158 ページの『基本的なファイル転送の詳細』](#)を参照してください。
- IBM WebSphere MQ Version 7.5 試用版は <https://www.ibm.com/developerworks/downloads/ws/wmq/> からダウンロードできます。
- IBM WebSphere MQ Version 7.5 に関するハードウェアおよびオペレーティング・システムの前提条件を満たす 2 台目のテスト・コンピューター。詳しくは、<https://www.ibm.com/support/docview.wss?uid=swg27006467> を参照してください。

このシナリオでは、2 台目のコンピューターは、クライアント接続チャンネル上でキュー・マネージャーと対話します。チャンネル認証レコードを使用することによって、以下の点が保証されます。

- 追加の Managed File Transfer エージェントをホストする新規コンピューターから発信されるものとして、着信接続が認証されます。
- 着信要求が、Managed File Transfer リソースへの適切なアクセス権限が付与されているユーザーにマップされます。

サンドボックス (ファイル転送がファイル・システムの特定の領域 (例えば、IBM WebSphere MQ 構成ファイル) を操作することを制限する手法) は、考慮されません。

### 手順

両方のコンピューターで、以下を実行します。

1. グループ FTEUSERS および FTEAGENTS を作成します。詳細については、ご使用のオペレーティング・システムの手順を参照してください。
2. ユーザー fteuser を作成し、そのユーザーを FTEAGENTS グループに追加します。
3. ユーザー fteagent を作成し、そのユーザーを FTEUSERS グループに追加します。

ユーザー fteuser はファイル転送を開始し、ユーザー fteagent はエージェント・プロセスの開始および停止を実行します。これらのユーザーは、mqm (または、Windows 管理者) グループのメンバーではないため、迷惑な管理攻撃に対してトポロジーを強化するのに役立ちます。エージェント・ベースで特定のユーザー (例えば、fteagent1、fteagent2) を定義することにより、よりきめの細かいセキュリティ・アクセスを実現できます。

## 拡張されたトポロジーをサポートするためのコンピューター 1 の変更

IBM WebSphere MQ の 2 台のコンピューターによるファイル転送のシナリオ用に拡張されたトポロジーをサポートするようにコンピューター 1 を準備します。

### このタスクについて

このタスクでは必要がないため、Agent2 を削除します。次に、リスナーを作成して開始し、コンピューター 2 に作成したエージェントからクライアント接続を受け入れます。このようにして、2 台のコンピューターにまたがる単純な IBM WebSphere MQ ネットワークを構築します。このシナリオでは、コンピューター 1 が Windows オペレーティング・システムを実行していることを想定しています。異なるプラットフォームを使用してこのシナリオを実行する場合、プラットフォーム固有の適切なコマンドを代わりに使用してください。

## 手順

1. エージェント Agent2 を停止します。

```
fteStopAgent AGENT2
```

**fteStopAgent** コマンドの詳細については、[fteStopAgent \(WebSphere MQ Managed File Transfer エージェントの停止\)](#) を参照してください。

2. エージェント Agent2 を削除します。

```
fteDeleteAgent AGENT2
runmqsc QM1 < <output>
```

**fteDeleteAgent** コマンドの詳細については、[fteDeleteAgent \(WebSphere MQ Managed File Transfer エージェントの削除\)](#) を参照してください。

コンピューター 2 で構成される新規ファイル転送エージェントが調整キュー・マネージャー QM1 と対話できるように、IBM WebSphere MQ のセキュリティーを構成します。この新規エージェントは、既存の SYSTEM.DEF.SVRCONN チャンネルを介して QM1 に接続します。ユーザーによってセキュリティー・ニーズが異なる場合があります。このトポロジーの強化に関する詳細については、[次の作業を参照](#)してください。

3. QM1 の MQSC インターフェースを開始します。

```
runmqsc QM1
```

4. ユーザーおよびエージェントからの着信要求を処理するために、2つのチャンネルを作成します。

```
DEFINE CHANNEL(FTE.USER.SVRCONN) CHLTYPE(SVRCONN)
DEFINE CHANNEL(FTE.AGENT.SVRCONN) CHLTYPE(SVRCONN)
```

5. チャンネル認証レコードを作成し、作成したユーザーを割り当てることにより、コンピューター 2 から QM1 への接続を許可します。

```
SET CHLAUTH('FTE.USER.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('<IP address of computer2>')
USERSRC (MAP) MCAUSER('fteuser' DESCR('Rule to allow file transfer users to communicate'))
ACTION(ADD)

SET CHLAUTH('FTE.AGENT.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('<IP address of computer2>')
USERSRC(MAP) MCAUSER('fteagent') DESCR('Rule to allow file transfer agent processes to
communicate') ACTION(ADD)
```

詳しくは、[チャンネル認証レコード](#)を参照してください。

このシナリオの目的は、トポロジーのロックダウンや強化ではなく、基本的なファイル転送を説明することにあります。実装されるセキュリティー・モデルは、このデモンストレーションをサポートしますが、ユーザー独自のセキュリティー脅威を理解し、必要に応じて適切な処置を取る必要があります。考慮すべきオプションについては、[次の作業を参照](#)してください。

6. IBM WebSphere MQ とのネットワーク通信に使用できるフリー・ポートを指定します。このフリー・ポート (例えば、1414) 使用するようリスナー LISTENER1 を定義します。

```
DEFINE LISTENER(LISTENER1) TRPTYPE(TCP) CONTROL(QMGR) PORT(1414)
```

7. リスナー LISTENER1 を開始します。

```
START LISTENER(LISTENER1)
```

8. QM1 の MQSC インターフェースを停止します。

```
end
```

9. FTEAGENTS グループと FTEUSERS グループに、コンピューター 2 で作成するエージェント AGENT3 のファイル転送アクションを実行するための IBM WebSphere MQ オブジェクトへの適切なアクセス権限

があることを確認します。独自のセキュリティー要件に合わせて、この構成を調整することができます。

- a. `setmqaut -m QM1 -t qmgr -g FTEAGENTS +connect +inq`
- b. `setmqaut -m QM1 -t qmgr -g FTEUSERS +connect`
- c. `setmqaut -m QM1 -n "SYSTEM.FTE" -t q -g FTEAGENTS +get +put`
- d. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT1" -t q -g FTEUSERS +put`
- e. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT1" -t q -g FTEAGENTS +setid +get +put`
- f. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT3" -t q -g FTEUSERS +put`
- g. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT3" -t q -g FTEAGENTS +setid +get +put`
- h. `setmqaut -m QM1 -n "SYSTEM.FTE.DATA.AGENT1" -t q -g FTEAGENTS +get +put`
- i. `setmqaut -m QM1 -n "SYSTEM.FTE.DATA.AGENT3" -t q -g FTEAGENTS +get +put`
- j. `setmqaut -m QM1 -n "SYSTEM.FTE.EVENT.AGENT1" -t q -g FTEAGENTS +get +put`
- k. `setmqaut -m QM1 -n "SYSTEM.FTE.EVENT.AGENT3" -t q -g FTEAGENTS +get +put`
- l. `setmqaut -m QM1 -n "SYSTEM.FTE.REPLY.AGENT1" -t q -g FTEAGENTS +get +put`
- m. `setmqaut -m QM1 -n "SYSTEM.FTE.REPLY.AGENT3" -t q -g FTEAGENTS +get +put`
- n. `setmqaut -m QM1 -n "SYSTEM.FTE.STATE.AGENT1" -t q -g FTEAGENTS +get +put +inq`
- o. `setmqaut -m QM1 -n "SYSTEM.FTE.STATE.AGENT3" -t q -g FTEAGENTS +get +put +inq`
- p. `setmqaut -m QM1 -n "SYSTEM.FTE" -t topic -g FTEUSERS +sub`
- q. `setmqaut -m QM1 -n "SYSTEM.FTE" -t topic -g FTEAGENTS +pub +sub`
- r. `setmqaut -m QM1 -n "SYSTEM.DEFAULT.MODEL.QUEUE" -t q -g FTEUSERS +dsp +browse +get +put`
- s. `setmqaut -m QM1 -n "SYSTEM.DEFAULT.MODEL.QUEUE" -t q -g FTEAGENTS +dsp +browse +get +put`

**setmqaut** コマンドの詳細については、[setmqaut](#) を参照してください。

グループへの権限付与の詳細については、[WebSphere MQ Managed File Transfer に固有のリソースのグループ権限](#)を参照してください。

## ファイル転送を行うためのコンピューター 2 の準備

IBM WebSphere MQ 2 台のコンピューターによるファイル転送シナリオのファイル転送のためのコンピューター 2 の準備について説明します。

### このタスクについて

このタスクは、コンピューター 2 が Linux オペレーティング・システムを実行していることを前提としています。異なるプラットフォームを使用してこのシナリオを実行する場合は、該当するコマンドをそのプラットフォーム固有の適切なコマンドで置き換える必要があります。

### 手順

1. IBM WebSphere MQ を各コンピューターにインストールします。その際、Managed File Transfer Agent コンポーネントと、その他の該当する前提条件コンポーネント (例えば、Managed File Transfer コマンド・ライン・ツール) も一緒にインストールします。詳細については、[インストール内容の選択](#)を参照してください。

このステップでは、IBM WebSphere MQ または IBM WebSphere MQ File Transfer Edition の以前のインストール済み環境がインストールされていないクリーンなコンピューターを使用していることを前提としています。そうでない場合は、共存することがサポートされていてインストール・メカニズムを調整できるかどうか、または環境を適切に構成できるかどうかを調べておく必要があります。詳細については、[複数のインストール](#)を参照してください。

Managed File Transfer コマンド・ライン・ツール・コンポーネントをインストールする際には、コンピューター 1 だけでなくコンピューター 2 から転送を定義して開始できるように設定できます。そのための手順をこのシナリオで説明します。

- mqm グループのユーザーとして、コンピューター 1 の QM1 を調整マネージャーに使用するファイル転送を構成します。次のコマンドを入力します。

```
fteSetupCoordination -coordinationQMgr QM1 -coordinationQMgrHost <computer1_hostname>
-coordinationQMgrPort 1414 -coordinationQMgrChannel FTE.USER.SVRCONN
```

生成される MQSC スクリプトを実行する必要はありません。コンピューター 1 を構成したときに実行したからです。

```
fteSetupCommands -connectionQMgr QM1 -connectionQMgrHost <computer1_hostname>
-connectionQMgrPort 1414 -connectionQMgrChannel FTE.USER.SVRCONN
```

詳細については、次の章を参照してください。

- [WebSphere MQ Managed File Transfer トポロジーの概要](#)
  - [fteSetupCoordination \(調整の詳細のセットアップ\)](#)
  - [fteSetupCommands \(command.properties ファイルの作成\)](#)
- QM1 に登録されたエージェントをリストして、クライアント接続の構成が正しいことを確認します。次のコマンドを入力します。

```
ftelistAgents
```

以下の出力が表示されます。

| Agent Name: | Queue Manager Name: | Status: |
|-------------|---------------------|---------|
| AGENT1      | QM1                 | Ready   |

- 2 番目のファイル転送エージェント AGENT3 を作成します。

```
fteCreateAgent -agentName AGENT3 -agentQMgr QM1 -agentQMgrHost <computer1_hostname>
-agentQMgrPort 1414 -agentQMgrChannel FTE.AGENT.SVRCONN
```

コンピューター 1 に切り替えて、次のコマンドを入力します。

```
runmqsc QM1 < <AGENT3_create.mqsc>
```

- コンピューター 2 に切り替えて、ユーザー fteagent として新しいファイル転送エージェント AGENT3 を始動します。

```
fteStartAgent AGENT3
```

- オプション: コンピューター 1 に切り替え、最初のシナリオで使用した特権 IBM WebSphere MQ 管理者ユーザーではなく、fteagent ユーザーとして AGENT1 を再始動します。
- クライアント接続の構成が正しいことを確認するために、QM1 に登録されたエージェントをリストします。

```
ftelistAgents
```

以下の出力が表示されます。

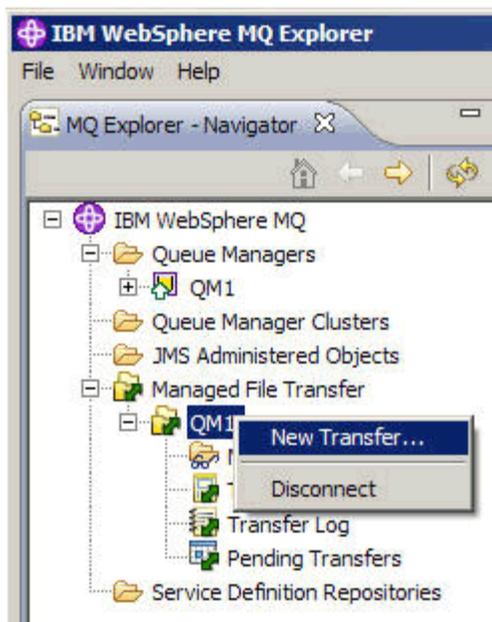
| Agent Name: | Queue Manager Name: | Status: |
|-------------|---------------------|---------|
| AGENT1      | QM1                 | READY   |
| AGENT3      | QM1                 | READY   |

## ソリューションの実装

このシナリオでソリューションを実装するには、IBM WebSphere MQ エクスプローラーを使用してファイル転送を定義および開始する必要があります。転送プロセスをモニターし、ファイル・システムを調べることによって動作が正常かどうかを確認できます。

### 手順

1. ソース・ファイルとターゲット・ディレクトリーを識別します。
  - a) コンピューター 1 上に、転送の対象となるサンプル・ファイルを作成します。例:  
C:\start\myfile.txt
  - b) このファイルの転送先にするコンピューター 2 上の既存のディレクトリーを識別します (例: /home/end/)。エージェントを始動したユーザーがこのディレクトリーに対して書き込み権限を持っていることを確認してください。
2. コンピューター 1 で IBM WebSphere MQ エクスプローラーを開始します。「スタート」メニューから (または同等の操作で) プログラムを開始するか、またはコマンド **MQExplorer** を実行します。詳細については、[IBM WebSphere MQ エクスプローラーの起動](#)を参照してください。
3. IBM WebSphere MQ エクスプローラー・ナビゲーターで「**ファイル転送管理**」をクリックし、**QM** を右クリックして「**新規転送**」を選択し、「新規転送」ウィザードを開始します。



4. 「**From**」セクションで、ソース・エージェントとして **AGENT1** を選択します。

From:

|                                                 |        |
|-------------------------------------------------|--------|
| Agent:                                          | AGENT1 |
| Type:                                           | File   |
| File:                                           |        |
| <input type="checkbox"/> Include subdirectories |        |

5. 先ほど作成したファイルへのパス (例えば、C:\start\myfile.txt) を入力します。

From:

Agent: AGENT1

Type: File

File: C:\Users\Ben Bakowski\Desktop\FTETEST\START\myDemoTransfer.txt

Include subdirectories

6. 「宛先」セクションで、「AGENT2」を宛先エージェントとして選択します。
7. 以前に識別した宛先ディレクトリーを入力します (例: /home/end/).

To:

Agent: AGENT2

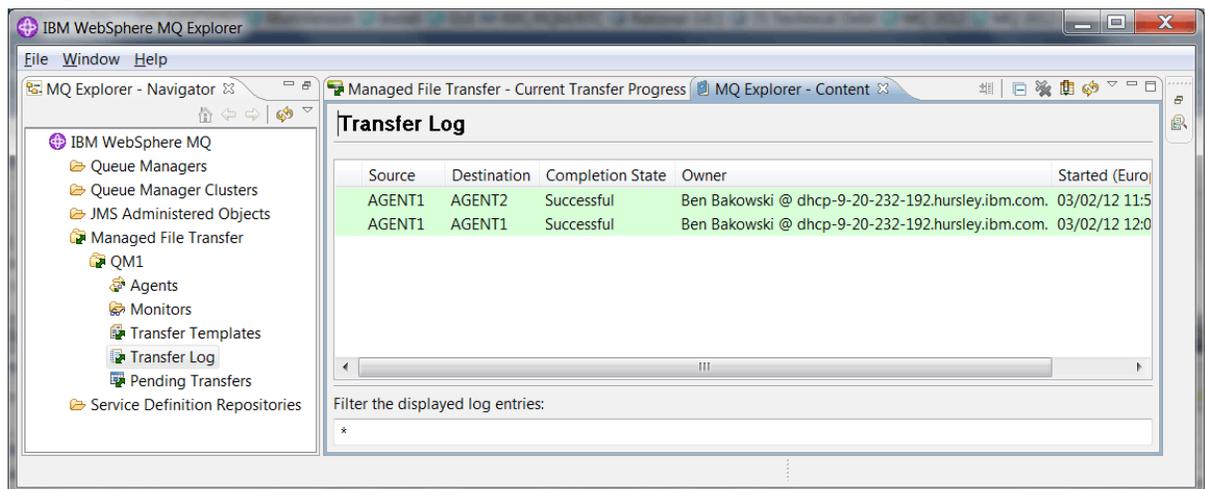
Type: File

Directory: /home/ben/FTETEST/END/

File name: myDemoTransfer.txt

Overwrite files on the destination file system that have the same name

8. 「すぐに終了」をクリックして、転送を開始します。
9. IBM WebSphere MQ エクスプローラーの「現在の転送進行状況」タブで、転送の進行状況をモニターできます。



10. 別の方法として、コンピューター 2 のファイル・システムを手動で調べて、新しいファイル (例: /home/end/myfile.txt) の存在を確認することもできます。

## コマンド・ラインを使用してファイルを転送する

このセクションでは、ファイル転送を開始する操作の柔軟性を示す例として、コマンド・ラインを使用して転送を開始する方法を説明します。このシナリオの範囲外ではありますが、これらの原則および Ant スクリプティング・テクノロジーに基づいて、より強力なファイル転送シナリオの定義および実装を行えます。

## 手順

1. 前のデモンストレーションで転送したファイル (例: /home/end/myfile.txt) を削除します。

2. ユーザー `fteuser` として `fteCreateTransfer` コマンドを入力し、コンピューター 1 上の `C:\start\myfile.txt` からコンピューター 2 上の `/home/end/myfile.txt` へのファイル転送を開始します。

```
fteCreateTransfer -sa AGENT1 -sm QM1 -da AGENT3 -dm QM1 -w -dd "/home/end/"
"C:\start\myfile.txt"
```

3. IBM WebSphere MQ エクスプローラーのファイル転送ログを調べるか、手動でファイル・システムを調べることによって、ファイル転送が成功したことを確認します。

## 次のタスク

IBM WebSphere MQ セキュリティーについて詳しくは、[セキュリティ](#) を参照してください。

外部のメディアを通じて、ファイル転送機能に関係したさらに多くのフィーチャーについて調べたいと思われるかもしれません。

- ファイル転送管理の管理という側面を提供する監査機能の追加について詳しくは、[178 ページの『ファイル転送管理への監査機能の追加』](#) を参照してください。
- 以下の事項に関する詳しい情報は、
  - トリガー: 新しいファイルを検出した時点で移動する。
  - トリガー: 1 つのファイルを検出した時点で複数のファイルの転送を開始するように構成する。
  - Apache Ant を使用して転送のスクリプトを記述する。

[https://www.ibm.com/developerworks/websphere/library/techarticles/1003\\_phillips/1003\\_phillips.html](https://www.ibm.com/developerworks/websphere/library/techarticles/1003_phillips/1003_phillips.html) を参照してください。

- 要件によっては、このシナリオで使用したのものとは異なるアクセス・モデルが必要になることがあります。使用環境の機密保護機能を高めるためのベスト・プラクティスの詳細については、[https://www.ibm.com/developerworks/websphere/library/techarticles/0902\\_wyatt/0902\\_wyatt.html](https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html) を参照してください。

## ファイル転送管理への監査機能の追加

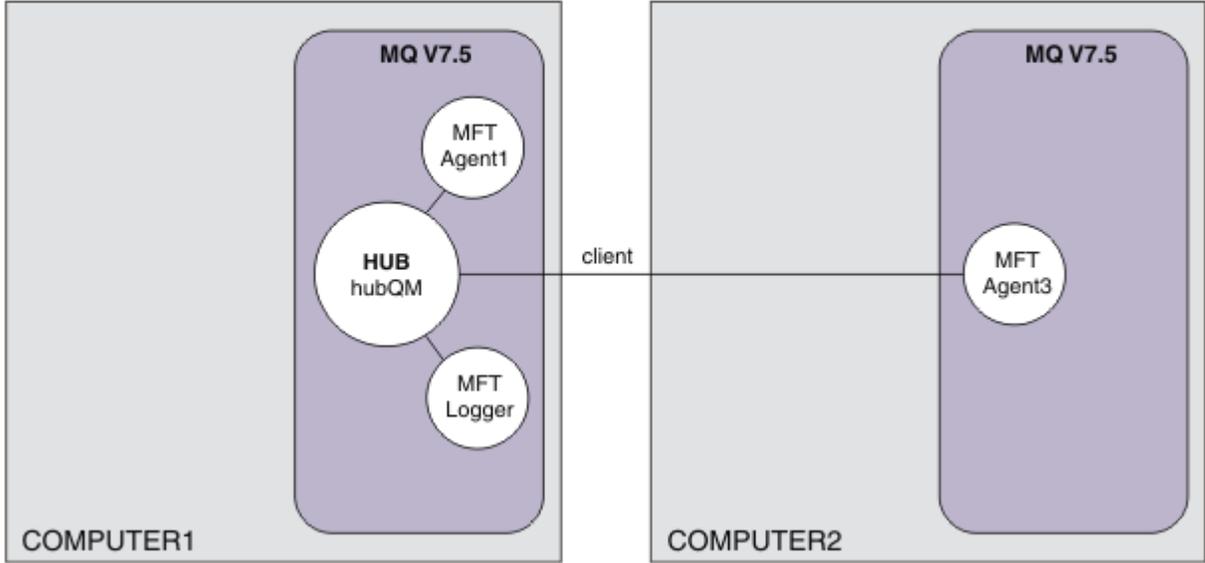
このシナリオを使用すると、ロガーを構成し、その機能を使用してどのように監査証跡を提供できるかを理解できます。

複数コンピューター間のファイル転送管理機能のデモンストレーションは、前述の 2 つのシナリオ、[158 ページの『基本的なファイル転送の詳細』](#) および [170 ページの『2 台のコンピューターでのファイル転送の詳細』](#) で作成しましたので、読者はファイル転送トポロジーの構成作業に習熟しているものと想定します。ここでは、IBM WebSphere MQ Version 7.5 がどのようにファイル転送をログに記録し、監査するかについて説明し、ファイル転送管理のうち管理という側面を紹介します。

前提条件は、2 番目のファイル転送管理シナリオを [170 ページの『2 台のコンピューターでのファイル転送の詳細』](#) シナリオの指示に従って完了していることです。

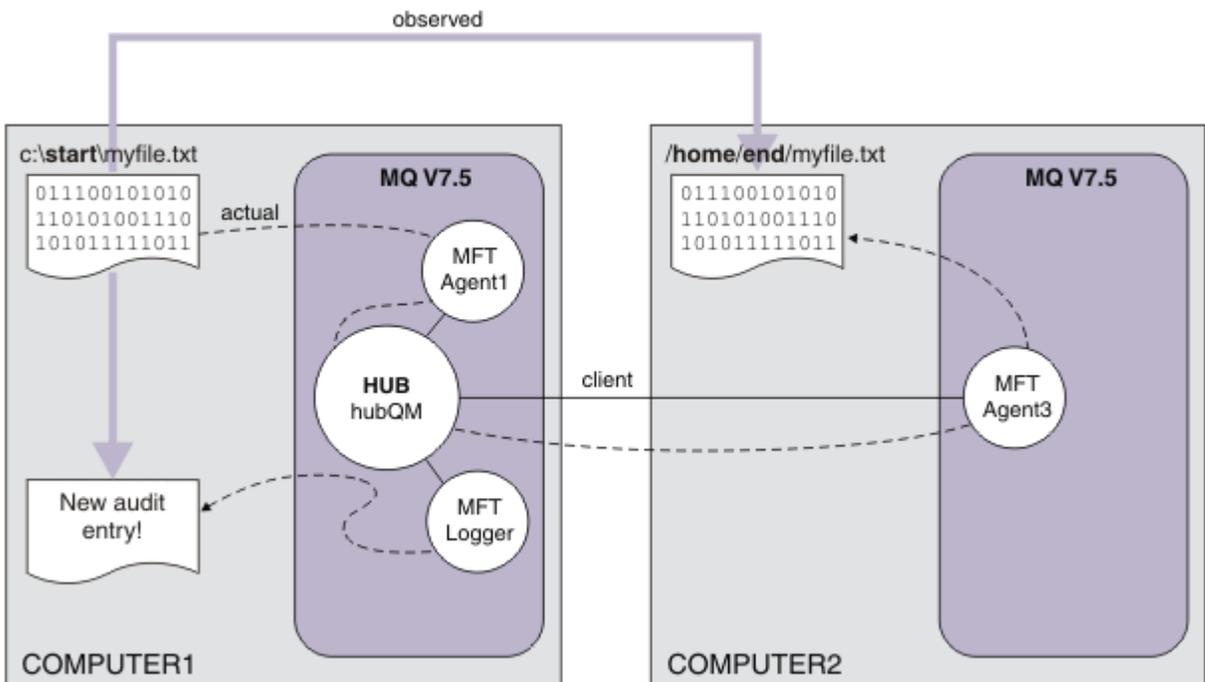
### 概要

このシナリオでは、[170 ページの『2 台のコンピューターでのファイル転送の詳細』](#) で構成した既存の Windows トポロジーおよび Linux トポロジーに進み、ファイル・ロギング機能を有効にします。ファイルのロギング機能を使用するために他のコンポーネントや製品をインストールする必要はないため、このシナリオを実行するために、2 台のコンピューターによるファイル転送シナリオの場合のライセンス交付や資格を変更する必要はありません。



また、データベース・ロガーを実装することもできます。データベース・ロガーは、例えばスケラビリティやフェイルオーバーといった面で実稼働環境に、より適しています。しかし、このシナリオをシンプルにするためと、他の製品をインストールする必要なしで済ませるために、完全にサポートされたファイル・ロガーを使用することにします。それ以上のセキュリティー上の側面は考慮に入れません。この機能により監査情報が提供されるので、それに関するセキュリティーの強化を検討することもできます。

このシナリオでは、ファイル転送が開始され、その詳細がログ・ファイルに取り込まれます。



## ソリューションの計画

ファイル転送管理のシナリオに監査機能を追加するために IBM WebSphere MQ に必要となる前提条件とライセンス要件について説明します。

このシナリオでは、ファイル転送を監査するためにロガーを作成します。ロガーはエージェント・プロセスと類似性の高い機能を提供するので、ロガーの開始と停止には既存の `ftagent` ユーザーを使用できます。ロガーを管理するために独自のユーザーまたはグループを作成しても構いません。

## 前提条件およびライセンス

170 ページの『[2 台のコンピューターでのファイル転送の詳細](#)』のシナリオからの作業用構成が必要です。

## ソリューションの実装

ファイル転送管理のシナリオに監査機能を追加するために、コンピューター 1 の IBM WebSphere MQ のファイル・ロガーの構成を修正します。

## 手順

1. ロガー・プロセスの実行時に `ftagent` グループが IBM WebSphere MQ オブジェクトに対して適切なアクセス権限を持っていることを確認します。次のコマンドを入力します。

```
setmqaut -m hubQM -n "SYSTEM.FTE.LOG.RJCT.MYFILELOGGER" -t q -g FTAGENTS +put
setmqaut -m hubQM -n "SYSTEM.FTE.LOG.CMD.MYFILELOGGER" -t q -g FTAGENTS +get
```

詳細については、[データベース・ロガーの権限](#)を参照してください。

2. IBM WebSphere MQ 管理者として、`hubQM` をロガー・キュー・マネージャーに使用して、ファイル・ロガーを作成します。

```
fteCreateLogger -loggerType FILE -loggerQMGr hubQM -fileLoggerMode LINEAR -fileSize 5MB
myFileLogger
runmqsc hubQM < <MYFILELOGGER_create.mqsc>
```

詳細については、[fteCreateLogger \(WebSphere MQ Managed File Transfer ロガーの作成\)](#)を参照してください。実稼働環境では、データベース・ロガーの使用を検討することもできます。

3. ユーザー `ftagent` として、ロガーを開始します。

```
fteStartLogger MYFILELOGGER
```

4. ロガーが開始したことを確認します。

ロガーが開始したことを確認するには、ファイル・システムを調べます。上記のコマンドを使用してファイル・ロガーを構成した後、ログは `<MQ INSTALL>/mqft/logs/hubQM/loggers/MYFILELOGGER/logs` にあります。 `output0.log` にメッセージ `"BFGDB0023I: The logger has completed startup activities and is now running."` が含まれていることを確認します。

5. 前のデモンストレーションで転送したファイル (例: `/home/end/myfile.txt`) を削除します。
6. ユーザー `ftuser` として、コンピューター 1 から `fteCreateTransfer` (新規ファイル転送の作成) コマンドを使用して、コンピューター 1 上のファイル `C:\start\myfile.txt` をコンピューター 2 上の `/home/end/myfile.txt` に転送する処理を開始します。次のコマンドを入力します。

```
fteCreateTransfer -sa AGENT1 -sm hubQM -da AGENT3 -dm hubQM -w -dd "/home/end/"
"C:\start\myfile.txt"
```

7. ロガーがこの転送をキャプチャーしたことを確認し、ログ・エントリーの内容を解釈します。

- a. ファイルを開く `<MQ INSTALL>/mqft/logs/hubQM/loggers/MYFILELOGGER/MYFILELOGGER-XXXXXXXXX.log`
- b. ログ・エントリーに、先ほど開始した転送が記録されており、ファイルのソースと宛先の場所、日付、時刻、リクエスター ID が含まれています。以下に例を示します。

```
2012-03-23T16:42:21;414d5120514d312020202020202020207a556b4f2000aa03;[TSTR]; ;
AGENT1;hubQM;STANDARD;AGENT3;hubQM;User;;;com.ibm.wmqfte.SourceAgent=AGENT1,
com.ibm.wmqfte.DestinationAgent=AGENT3, com.ibm.wmqfte.MqmdUser=User,
com.ibm.wmqfte.OriginatingUser=User, com.ibm.wmqfte.OriginatingHost=
```

```
dhcp-9-10-123-123.hursley.ibm.com., com.ibm.wmqfte.TransferId=
414d5120514d3120202020202020202020202020207a556b4f2000aa03, com.ibm.wmqfte.Priority=0;

2012-03-23T16:42:21;414d5120514d31202020202020202020202020207a556b4f2000aa03;[TPRO];0 ;
C:\start\myfile.txt;51447;file;leave ;;;;;/home/end/myfile.txt;51447;file;
;;;;;;

2012-03-23T16:42:21;414d5120514d31202020202020202020202020207a556b4f2000aa03;[TCOM];0 ;
AGENT1;hubQM;STANDARD;AGENT3;hubQM;STANDARD;User;;BFGRP0032I: The file transfer
request has successfully completed.;com.ibm.wmqfte.SourceAgent=AGENT1,
com.ibm.wmqfte.DestinationAgent=AGENT3, com.ibm.wmqfte.MqmdUser=User,
com.ibm.wmqfte.OriginatingUser=User, com.ibm.wmqfte.OriginatingHost=
dhcp-9-20-123-123.hursley.ibm.com.,
com.ibm.wmqfte.TransferId=414d5120514d31202020202020202020202020207a556b4f2000aa03,
com.ibm.wmqfte.Priority=0;
```

## 用語集

この用語集には、IBM WebSphere MQ の用語と定義が含まれています。

この用語集では以下の相互参照が使用されています。

- 「を参照してください」と示されている場合は、当該用語から、優先的に使用される同義語を参照します。または、頭字語や略語から、定義されている完全な形式の用語を参照します。
- 「も参照」と示されている場合は、関連用語や対義語を参照します。

[181 ページの『A』](#) [184 ページの『B』](#) [185 ページの『C』](#) [190 ページの『D』](#) [192 ページの『E』](#) [193 ページの『F』](#) [194 ページの『G』](#) [195 ページの『H』](#) [195 ページの『I』](#) [197 ページの『J』](#) [198 ページの『K』](#) [198 ページの『L』](#) [200 ページの『M』](#) [204 ページの『N』](#) [205 ページの『O』](#) [207 ページの『P』](#) [210 ページの『Q』](#) [210 ページの『R』](#) [213 ページの『S』](#) [218 ページの『T』](#) [221 ページの『U』](#) [222 ページの『V』](#) [222 ページの『W』](#) [223 ページの『X』](#)

## A

### 異常終了理由コード (abend reason code)

z/OS オペレーティング・システムで稼働するプログラムの問題を一意的に特定する、4 バイトの 16 進コード。

### 抽象クラス (abstract class)

オブジェクト指向プログラミングにおいて、1つの概念を表すクラスのこと。これから派生するクラスは、その概念の実装を表す。抽象クラスからはオブジェクトを構成できません。つまり、インスタンスを生成することはできません。 [親クラス \(parent class\)](#) も参照してください。

### アクセス制御 (access control)

コンピューター・セキュリティーで、許可されているコンピューター・システムのリソースのみにユーザーがアクセスできることを保証するプロセス。

### アクセス制御リスト (ACL) (access control list (ACL))

コンピューター・セキュリティーにおいて、あるオブジェクトに関して、そのオブジェクトにアクセスできるすべてのサブジェクト、およびそれらのアクセス権限を識別するリスト。

### アカウントビリティ (accountability)

個人のアクションに対して負っている責任の質。

### ACL

[アクセス制御リスト \(access control list\)](#) を参照してください。

### アクティブ・ログ (active log)

リカバリー・イベントが発生した際にそれを記録する場所となる、固定サイズのデータ・セット。アクティブ・ログがフルの場合、そのアクティブ・ログのコンテンツはアーカイブ・ログにコピーされる。

### アクティブ・キュー・マネージャー・インスタンス (active queue manager instance)

要求を処理している、実行中のマルチ・インスタンス・キュー・マネージャーのインスタンス。複数インスタンス・キュー・マネージャーのアクティブ・インスタンスは1つしかありません。

## アダプター

2つの異なるソフトウェア・コンポーネントが相互通信するための仲介ソフトウェア・コンポーネント。

## アドレス・スペース (ASID) (address space (ASID))

コンピューター・プログラムまたはプロセスで使用可能なアドレスの範囲。アドレス・スペースは、物理ストレージ、仮想ストレージ、またはその両方を指す場合がある。[接続アドレス・スペース \(allied address space\)](#)、[バッファ・プール \(buffer pool\)](#) も参照してください。

## 管理バッグ (administration bag)

WebSphere MQ 管理インターフェース (MQAI) で、WebSphere MQ を管理するために作成されるもので、データ項目の順序変更、リストの作成、およびメッセージ内のセクター検査が可能な、データ・バッグの一種。

## 管理トピック・オブジェクト (administrative topic object)

ユーザーがデフォルト以外の特定の属性をトピックに割り当てられるようにするオブジェクト。

## 管理者コマンド (administrator command)

キュー、プロセス、名前リストなどの WebSphere MQ オブジェクトを管理するために使用されるコマンド。

## 拡張プログラム間通信機能 (APPC) (Advanced Program-to-Program Communication (APPC))

相互接続されたシステム間での通信、およびプログラムの処理の共有を可能にする SNA LU 6.2 プロトコルのインプリメンテーション。

## 拡張テレメトリー・クライアント (advanced telemetry client)

[テレメトリー拡張クライアント \(telemetry advanced client\)](#) を参照してください。

## アフィニティー (affinity)

互いに何らかの関係または依存性を持つオブジェクト間の関連。

## アラート (alert)

イベントまたは発生しそうなイベントを信号通知するメッセージまたはその他の標識。

## アラート・モニター (alert monitor)

WebSphere MQ for z/OS で、WebSphere MQ for z/OS への接続要求の結果として発生する予定外のイベントを処理するための CICS アダプターのコンポーネント。

## 別名キュー

WebSphere MQ オブジェクトの 1 つで、ローカル・キュー・マネージャーに対して定義されている基本キューまたはトピックの別名を名前として持つもの。アプリケーションまたはキュー・マネージャーが別名キューを使用するときには、別名が解決され、要求した操作が関連する基本オブジェクトに対して実行されます。

## 別名キュー・オブジェクト (alias queue object)

WebSphere MQ のオブジェクトで、その名前が、ローカル・キュー・マネージャーに定義したベース・キューの別名であるもの。アプリケーションまたはキュー・マネージャーが別名キューを使用するときには、別名が解決され、要求した操作が関連する基本キューで実行されます。

## 接続アドレス・スペース (allied address space)

WebSphere MQ for z/OS に接続する z/OS アドレス・スペース。

## アライ (ally)

[接続アドレス・スペース \(allied address space\)](#) を参照してください。

## 代替ユーザー権限 (alternate user authority)

セキュリティ検査のために別のユーザー ID に提供されるユーザー ID の能力。アプリケーションでは、WebSphere MQ オブジェクトをオープンするときに、MQOPEN、MQPUT1 または MQSUB 呼び出しにユーザー ID を指定することができます。それにより、キュー・マネージャーが権限検査において、アプリケーションに関連付けられたユーザー ID ではなく、このユーザー ID を使用するようになります。

## 代替ユーザー・セキュリティ (alternate user security)

z/OS で、アプリケーションが WebSphere MQ オブジェクトをオープンするときに、代替ユーザー権限を要求すると実行される権限チェック。

## APAR

[プログラム診断依頼書 \(authorized program analysis report\)](#) を参照してください。

## APF

[許可プログラム機能 \(authorized program facility\)](#) を参照してください。

## API 交差出口 (API-crossing exit)

API 出口と類似する概念の、ユーザーが書いたプログラム。WebSphere MQ for z/OS 上の CICS アプリケーションでのみサポートされます。

## API 出口 (API exit)

MQI 呼び出しの機能をモニターまたは変更するためにユーザーが書いたプログラム。アプリケーションから出される各 MQI 呼び出しに対して、API 出口は、キュー・マネージャーが呼び出しの処理を開始する前、およびキュー・マネージャーが呼び出しの処理を完了した後に起動されます。API 出口は、MQI 呼び出し上のすべてのパラメーターを調査および変更することができます。

## APPC

[拡張プログラム間通信機能 \(Advanced Program-to-Program Communication\)](#) を参照してください。

## アプリケーション定義フォーマット (application-defined format)

ユーザー・アプリケーションがその意味を定義する、メッセージ内のアプリケーション・データ。[ビルトイン・フォーマット \(built-in format\)](#) も参照してください。

## アプリケーション環境 (application environment)

ソフトウェアおよびそれをサポートするサーバーまたはネットワークのインフラストラクチャーを含む環境。

## アプリケーション・レベル・セキュリティー (application level security)

アプリケーションによる MQI 呼び出しの発行時に呼び出されるセキュリティー・サービス。

## アプリケーション・ログ (application log)

Windows システムでは、意味を持つアプリケーション・イベントを記録するログ。

## アプリケーション・キュー (application queue)

トリガーが設定されていて、そのトリガー条件が満たされた場合にトリガー・メッセージを書き込む必要があるローカル・キュー。

## 保存ログ (archive log)

ストレージ・デバイス上のデータ・セット。アクティブ・ログがそのサイズの上限に達したときに、各アクティブ・ログ・データ・セットの内容が WebSphere MQ によってここにコピーされます。[リカバリー・ログ \(recovery log\)](#) も参照してください。

## ARM

[自動再始動マネージャー \(automatic restart manager\)](#) を参照してください。

## ASID

[アドレス・スペース \(address space\)](#) を参照してください。

## 非対称鍵暗号化方式 (asymmetric key cryptography)

暗号化方式のシステム。誰でも知っている公開鍵と、メッセージの受信者と送信者しか知らない秘密鍵の 2 つのキーを使うものです。[対称鍵暗号化方式 \(symmetric key cryptography\)](#) も参照してください。

## 非同期消費 (asynchronous consumption)

アプリケーションが一連のキューからメッセージを消費できるようにする、MQI 呼び出しのセットを使用した処理。メッセージは、アプリケーションによって識別されたコードのユニットを呼び出すことによりアプリケーションに送信され、メッセージまたはメッセージを表すトークンのいずれかを引き渡します。

## 非同期メッセージング (asynchronous messaging)

プログラム間の通信方法。プログラムがメッセージをメッセージ・キューに置き、そのメッセージに対する応答を待たないで、独自に処理を続けます。[同期メッセージング \(synchronous messaging\)](#) も参照してください。

## 非同期書き込み (asynchronous put)

アプリケーションがキュー・マネージャーからの応答を待たずに行う、メッセージの書き込み。

## 属性

1. オブジェクト指向プログラミングで、他のどのプロパティーからも明確に区別できるオブジェクトまたはクラスのプロパティー。属性は、状況情報を記述することが頻繁にあります。

2. エンティティを記述するエンティティの特性または特徴。例えば、従業員の電話番号は、従業員の属性の1つ。 [エンティティ \(entity\)](#) も参照してください。

## 認証

コンピューター・システムのユーザーが本人であることを証明するセキュリティー・サービス。このサービスを実現するための共通メカニズムは、パスワードおよびデジタル署名です。

## 認証情報オブジェクト (authentication information object)

Secure Sockets Layer (SSL) セキュリティーのサポートとして、LDAP サーバーを使用した証明書取り消しリスト (CRL) を検査する場合に必要な定義を提供するオブジェクト。

## 権限チェック (authority check)

[許可検査 \(authorization check\)](#) を参照してください。

## 許可

ユーザー、システム、またはプロセスに、オブジェクト、リソース、または機能への完全なアクセス権限または制限付きのアクセス権限を付与するプロセス。

## 許可検査 (authorization check)

例えば、管理者が WebSphere MQ を管理するためのコマンドを出した時、またはアプリケーションがキュー・マネージャーに接続しようとした時など、ユーザーまたはアプリケーションがシステム・リソースにアクセスしようとした時に実行されるセキュリティー検査。

## 許可ファイル (authorization file)

1つのオブジェクト、複数のオブジェクトのクラス、または複数のオブジェクトのすべてのクラスのセキュリティー定義を指定するファイル。

## 許可サービス (authorization service)

UNIX および Linux システム上の WebSphere MQ、および WebSphere MQ for Windows において、コマンドまたは呼び出しに関連付けられたユーザー ID のコマンドおよび MQI 呼び出しの権限検査を提供するサービス。

## プログラム診断依頼書 (APAR) (authorized program analysis report (APAR))

IBM 提供のプログラムのサポート対象リリースにおける問題点に対する修正の要求。

## 許可プログラム機能 (APF) (authorized program facility (APF))

z/OS 環境において、制限付き機能の使用を許可されているプログラムを識別できるようにする機能。

## 自動再始動マネージャー (ARM) (automatic restart manager (ARM))

z/OS のリカバリー機能の1つ。バッチ・ジョブおよび開始タスク、またはそれらが実行されるシステムが予期せずに終了した後、そのバッチ・ジョブおよび開始タスクを自動的に再始動できる。

## B

### バックアウト

現在の作業単位中にリソースに対して行われたすべての変更を元に戻す操作。 [コミット \(commit\)](#) も参照してください。

### バッグ

[データ・バッグ \(data bag\)](#) を参照してください。

### バー (bar)

z/OS のメモリー制限。64 ビット・システムの場合、2 GB に設定されています。バーは、2 ギガバイト・アドレスより下のストレージと 2 ギガバイト・アドレスより上のストレージを分けます。バーより上に位置するエリアはデータ用です。プログラムがバーより上で実行されることはありません。

### 基本マッピング・サポート (BMS) (Basic Mapping support (BMS))

CICS とアプリケーション・プログラム間のインターフェース。多様な端末によって使われる制御文字に関係なく、入力および出力表示データをフォーマットし、複数ページの出力メッセージを経路指定します。

### 動作 (behavior)

オブジェクト指向プログラミングで、メソッド中に組み込まれた機能性。

### BMS

[基本マッピング・サポート \(basic mapping support\)](#) を参照してください。

## Booch 手法 (Booch methodology)

ユーザーがオブジェクト指向パラダイムを使ってシステムを設計できるようにする、オブジェクト指向の手法。

## ブートストラップ・データ・セット (BSDS) (bootstrap data set (BSDS))

WebSphere MQ for z/OS にとって既知のすべてのアクティブ・ログ・データ・セットおよびアーカイブ・ログ・データ・セットのインベントリ、および最近のすべての WebSphere MQ for z/OS アクティビティのラップアラウンド・インベントリが含まれる VSAM データ・セット。BSDS は、WebSphere MQ for z/OS サブシステムを再始動するために必要です。

## ブラウズ (browse)

メッセージ・キューイングにおいて、メッセージをキューから削除せずにコピーすること。[読み取り \(get\)](#)、[書き込み \(put\)](#) も参照してください。

## ブラウズ・カーソル (browse cursor)

メッセージ・キューイングにおいて、次の順番のメッセージを識別するためにキューをブラウズする際に使用される標識。

## BSDS

[ブートストラップ・データ・セット \(bootstrap data set\)](#) を参照してください。

## バッファ・プール

データ・ページが読み込まれるメモリーの領域。ここでそれらのデータ・ページが変更され、処理中は保持される。[アドレス・スペース \(address space\)](#) も参照してください。

## ビルトイン・フォーマット (built-in format)

キュー・マネージャーがその意味を定義する、メッセージ内のアプリケーション・データ。[アプリケーション定義のフォーマット \(application-defined format\)](#) も参照してください。

# C

## CA

[認証局 \(certificate authority\)](#) を参照してください。

## CAF

[クライアント接続機構 \(client attachment feature\)](#) を参照してください。

## コールバック (callback)

メッセージ・コンシューマーまたはイベント・ハンドラー・ルーチン。

## CCDT

[クライアント・チャンネル定義テーブル \(client channel definition table\)](#) を参照してください。

## 相互相関関数

[チャンネル制御機能 \(channel control function\)](#) を参照してください。

## CCSID

[コード化文字セット識別子 \(coded character set identifier\)](#) を参照してください。

## CDF

[チャンネル定義ファイル \(channel definition file\)](#) を参照してください。

## 認証局 (CA) (certificate authority (CA))

証明書署名要求に応じてデジタル証明書を発行する、信頼できるサード・パーティーの組織または会社。認証局は、固有の証明書を付与された個人の ID を検査する。[Secure Sockets Layer](#) も参照。

## 証明書チェーン (certificate chain)

個人証明書からチェーンの最上位のルート証明書まで、暗号化で相互に関係する証明書の階層。

## 証明書の有効期限 (certificate expiration)

デジタル証明書には、証明書が有効である日付範囲があります。有効な日付範囲外では、証明書は「有効期限切れ」と言われます。

## 証明書取り消しリスト (CRL) (certificate revocation list (CRL))

予定された満了日より前に取り消された認証のリスト。証明書取り消しリストは認証局によって保守され、Secure Sockets Layer (SSL) ハンドシェイク中に、関与する証明書が取り消されていないか確認するために使用される。

## 証明書ストア (certificate store)

キー・リポジトリの Windows での名称。

## 証明書署名要求 (CSR) (certificate signing request (CSR))

ユーティリティまたは組織の公開鍵およびサブジェクト識別名が含まれる要求。CA にそのユーティリティのデジタル署名を発行してもらうため、CA に送信します。

## CF

[カップリング・ファシリティ \(coupling facility\)](#) を参照してください。

## CFSTRUCT

キュー・マネージャーがカップリング・ファシリティ・リスト構造を使うことを記述するために使われる WebSphere MQ オブジェクト。

## channel

2つのキュー・マネージャー間 (メッセージ・チャンネル) またはクライアントとキュー・マネージャー間 (MQI チャンネル) の通信リンクを定義する WebSphere MQ オブジェクト。 [メッセージ・チャンネル \(message channel\)](#)、 [MQI チャンネル \(MQI channel\)](#) も参照してください。

## チャンネル・コールバック (channel callback)

正しいマシンへのチャンネル接続が確立されていることを確認するメカニズム。チャンネル・コールバックでは、送信側チャンネルは送信側の定義を使用して、元の要求側チャンネルをコールバックします。

## チャンネル制御機能 (CCF) (channel control function (CCF))

チャンネルのセットアップと制御ができるように、オペレーター・パネル・インターフェースと一緒にメッセージを送信キューから通信リンクに、また通信リンクからローカル・キューに移動するプログラム。

## チャンネル定義ファイル (CDF) (channel definition file (CDF))

伝送キューを通信リンクに関連付ける通信チャンネル定義を含むファイル。

## チャンネル・イベント (channel event)

チャンネル・インスタンスの開始または定義など、チャンネル操作中に検出された状況を報告するイベント。チャンネル・イベントは、チャンネルの両端のキュー・マネージャーで生成されます。

## チャンネル出口プログラム (channel exit program)

メッセージ・チャンネル・エージェント (MCA) の処理シーケンス内で定義された幾つかの場所の 1 つから呼び出される、ユーザーが書いたプログラム。

## チャンネル・イニシエーター

起動キューをモニターし、トリガー条件が発生したら送信側チャンネルを開始する WebSphere MQ 分散キューイングのコンポーネント。

## チャンネル・リスナー (channel listener)

ネットワークをモニターし、スタートアップ要求が発生したら受信側チャンネルを開始する WebSphere MQ 分散キューイングのコンポーネント。

## チェックポイント (checkpoint)

中断した場合にプログラムを再始動できるように、チェックを行ったり、データの記録を行ったりする、プログラム内の場所。

## CI

[制御インターバル \(control interval\)](#) を参照してください。

## CipherSpec

認証が完了した後、SSL メッセージに適用される暗号化アルゴリズムとハッシュ機能の組み合わせ。

## 暗号スイート (cipher suite)

保護されたデータ交換に使用される、認証、鍵交換アルゴリズム、および Secure Sockets Layer (SSL) 暗号仕様の組み合わせ。

## 暗号文 (ciphertext)

暗号化されたデータ。暗号文 (ciphertext) は、鍵を使ってプレーン・テキスト (plaintext) に変換 (復号) しないと読めません。 [平文 \(cleartext\)](#) も参照してください。

## 循環ロギング (circular logging)

UNIX および Linux システム上の WebSphere MQ、および WebSphere MQ for Windows において、すべての再始動データをログ・ファイルのリングに保持するプロセス。 [リニア・ロギング \(linear logging\)](#) も参照してください。

## CL

[コマンド言語 \(Command Language\)](#) を参照してください。

### クラス

オブジェクト指向設計またはプログラミングで、共通の定義および共通のプロパティ、操作、および動作を持つオブジェクトの作成に使用できるモデルまたはテンプレート。オブジェクトはクラスのインスタンスになります。

### クラス階層 (class hierarchy)

単一の継承を共有するクラス間の関係。

### クラス・ライブラリー (class library)

オブジェクト指向プログラミングにおいて、プログラマーがアプリケーション開発時に指定して使用することができる、事前に作成されたクラスまたはコード化されたテンプレートの集合。

### 平文 (cleartext)

ネットワーク上で読み取り可能な形式で送信される文字ストリング。圧縮の目的でエンコードされることはあるが、簡単にデコードできます。[暗号文 \(ciphertext\)](#) も参照してください。

### クライアント

ローカル・ユーザー・アプリケーションに、サーバーのキューイング・サービスへのアクセスを提供する実行時コンポーネント。アプリケーションが使用するキューは、サーバーに置かれています。[WebSphere MQ MQI クライアント \(WebSphere MQ MQI client\)](#)、[WebSphere MQ Java クライアント \(WebSphere MQ Java client\)](#)、[WebSphere MQ の完全に管理された .NET クライアント \(WebSphere MQ fully-managed .NET client\)](#) も参照してください。

### クライアント・アプリケーション (client application)

ワークステーション上で実行されるアプリケーションで、サーバー上のキューイング・サービスにアプリケーション・アクセスを提供するためにクライアントにリンクされているもの。

### クライアント接続機構 (CAF) (Client Attachment feature (CAF))

z/OS へのクライアントの接続をサポートするオプション。

### クライアント・チャンネル定義テーブル (CCDT) (client channel definition table (CCDT))

1つ以上のクライアント接続チャンネル定義を含むファイル。

### クライアント接続チャンネル・タイプ (client-connection channel type)

WebSphere MQ クライアントに関連付けられた、MQI チャンネル定義のタイプ。[サーバー接続チャンネル・タイプ \(server-connection channel type\)](#) も参照してください。

## CLUSRCVR

[クラスター受信側チャンネル \(cluster-receiver channel\)](#) を参照してください。

## CLUSSDR

[クラスター送信側チャンネル \(cluster-sender channel\)](#) を参照してください。

### クラスター

WebSphere MQ で、1つ以上のコンピューター上に存在する複数のキュー・マネージャーで構成されたグループ。自動的に相互接続し、キュー・マネージャー間にキューおよびトピックを通知して、ロード・balancingと冗長化を行います。

### クラスター・キュー

クラスター・キュー・マネージャーによってホストされ、クラスター内の任意のキュー・マネージャーに接続されたアプリケーションから書き込まれるメッセージのターゲットとして定義されるローカル・キュー。メッセージを取得するすべてのアプリケーションが、ローカルで接続されている必要があります。

### クラスター・キュー・マネージャー (cluster queue manager)

クラスターのメンバーであるキュー・マネージャー。キュー・マネージャーは、複数のクラスターのメンバーにすることができます。

### クラスター受信側チャンネル (CLUSRCVR) (cluster-receiver channel (CLUSRCVR))

クラスター・キュー・マネージャーが、クラスター内の別のキュー・マネージャーからメッセージを受信したり、リポジトリ・キュー・マネージャーからクラスター情報を受信することのできるチャンネル。

### クラスター送信側チャンネル (CLUSDR) (cluster-sender channel (CLUSDR))

クラスター・キュー・マネージャーが、クラスター内の別のキュー・マネージャーへメッセージを送信したり、リポジトリ・キュー・マネージャーへクラスター情報を送信することのできるチャンネル。

### クラスター・トピック (cluster topic)

クラスター・キュー・マネージャーで定義されている管理トピックで、クラスター内の他のキュー・マネージャーも使用できる。

### クラスター伝送キュー (cluster transmission queue)

キュー・マネージャーが同じクラスター内の別のキュー・マネージャーに送信するすべてのメッセージを保持する伝送キュー。このキューは、SYSTEM.CLUSTER.TRANSMIT.QUEUE と呼ばれます。

### CMS 鍵データベース (CMS key database)

CMS 鍵データベースは、Windows システム、UNIX システム、Linux、およびそれらのプラットフォームのクライアントによってサポートされるデータベースのフォーマットです。 .kdb で終わるファイルは CMS 形式です。 .kdb ファイルには、証明書および鍵が格納されます。

### コード化文字セット ID (CCSID) (coded character set identifier (CCSID))

エンコード・スキーム ID、文字セット ID、コード・ページ ID、およびその他の情報の特定のセットを含んでいる 16 ビット数。コード化されたグラフィック文字の表現を一意的に識別する。

### 共存 (coexistence)

同じコンピューターで複数の異なるバージョンの WebSphere MQ が機能できること。

### command

アクションを開始したりサービスを開始したりするのに使用するステートメント。コマンドは、コマンド名の省略形とそのパラメーター、およびフラグ (該当する場合) で構成される。

### コマンド・バッグ (command bag)

MQAI で、WebSphere MQ オブジェクトを管理するために作成されるバッグの一種。ただし、メッセージ内でデータ項目の順序を変更したり、リストを作成したりすることはできません。

### コマンド・イベント (command event)

MQSC または PCF コマンドが正常に実行されたという通知。

### コマンド言語 (CL) (Command Language (CL))

WebSphere MQ for iSeries において、コマンド行で、または制御言語プログラムを作成することで、コマンドを実行するために使用できる言語。

### コマンド接頭部 (command prefix)

1. 1 文字から 8 文字のコマンド ID。このコマンド接頭部により、コマンドが、z/OS ではなく、アプリケーションまたはサブシステムに属していると識別されます。
2. WebSphere MQ for z/OS で、WebSphere MQ for z/OS のコマンドの送信先となり、WebSphere MQ for z/OS のオペレーター・メッセージの送信元となるキュー・マネージャーを特定する文字ストリング。

### コマンド・サーバー (command server)

システム・コマンド入力キューからコマンドを読み取って検査し、正しいコマンドをコマンド・プロセッサへ渡す WebSphere MQ コンポーネント。

### コミット (commit)

現行のリカバリー単位 (UR) または作業単位 (UOW) で行われたすべての変更を適用すること。この操作の完了後に、新しい UR または UOW を開始できる。

### 共通名 (CN) (common name (CN))

通常、証明書の所有者に関連付けられた名前を表す、X.509 証明書の識別名 (DN) 属性のコンポーネント。人物の場合、通常 CN は実際の名前になります。Web サーバーの場合、CN はサーバーの完全修飾されたホスト名およびドメイン・ネームになります。WebSphere MQ の場合、このフィールドに関する特定の要件はありませんが、多くの管理者はキュー・マネージャーの名前を使用します。

識別名 (Distinguished Name) も参照。

### 完了コード

メッセージ・キュー・インターフェース (MQI) 呼び出しがどのように終了したかを示す戻りコード。

## 機密性 (confidentiality)

機密情報を無認可の開示から保護するセキュリティー・サービス。このサービスを実現するための一般的な方法は、暗号化です。

## 構成イベント (configuration event)

オブジェクトの属性についての通知。この通知は、オブジェクトが作成、変更、削除されたときに生成されます。また、明示的な要求によっても生成されます。

## 接続アフィニティー (connection affinity)

複数接続が使用可能な場合に、クライアント・アプリケーションがキュー・マネージャーに接続するのに使用するクライアント・チャンネル定義を指定する、チャンネル属性。

## 接続ファクトリー (connection factory)

Java EE コンポーネントがリソースにアクセスすることを可能にする接続を生成する一連の構成値。接続ファクトリーは、アプリケーションからエンタープライズ情報システム (EIS) へのオンデマンド接続を提供し、アプリケーション・サーバーが分散トランザクションで EIS を登録できるようにする。

## 接続ハンドル

プログラムが、接続先のキュー・マネージャーにアクセスするために使う識別子またはトークン。

## コンストラクター (constructor)

オブジェクト指向プログラミングで、オブジェクトを初期化するために使用される特殊なメソッド。

## コンシューム (consume)

キューからメッセージを除去し、そのコンテンツを呼び出し側のアプリケーションに戻すこと。

## コンシューマー (consumer)

メッセージを受信し、処理するアプリケーション。 [メッセージ・コンシューマー \(message consumer\)](#) も参照してください。

## コンテキスト・セキュリティー (context security)

z/OS で、アプリケーションがキューをオープンし、キューに置くメッセージ内にコンテキストを設定するか、受信したメッセージからのコンテキストをキューに置くメッセージへ渡すように指定したときに実施される権限チェック。

## 制御コマンド (control command)

UNIX および Linux システム上の WebSphere MQ、および WebSphere MQ for Windows で、オペレーティング・システムのコマンド行から対話式に入力できるコマンド。制御コマンドを実行するには、WebSphere MQ 製品がインストールされていることのみが必要で、特別なユーティリティーまたはプログラムは必要ありません。

## 制御インターバル (CI) (control interval (CI))

VSAM がレコードを保管したり分散フリー・スペースを作成する、直接アクセス・ストレージ中の固定長の領域。制御インターバルは、VSAM と直接アクセス・ストレージ間で転送される情報の単位です。制御インターバルには、必ず整数個の物理レコードが含まれます。

## 制御されたシャットダウン (controlled shutdown)

[静的シャットダウン \(quiesced shutdown\)](#) を参照してください。

## 相関 ID

関連メッセージを識別する手段を提供する、メッセージ内のフィールド。相関 ID は、要求メッセージを対応する応答メッセージと突き合わせるためなどに使用される。

## カップリング・ファシリティ (CF) (coupling facility (CF))

シスプレックスで高速なキャッシング、リスト処理、およびロック機能を提供する特殊なロジカル・パーティション。

## CPF

[コマンド接頭部 \(command prefix\)](#) を参照してください。

## CR (証明書要求) (certificate request)

[証明書署名要求 \(certificate signing request\)](#) の同義語。

## CRL

[認証取り消しリスト \(certificate revocation list\)](#) を参照してください。

## システム間カップリング・ファシリティ (XCF) (cross-system coupling facility (XCF))

シスプレックス内で実行される認可プログラム間の連携をサポートする機能を提供するコンポーネント。

## 暗号化方式 (cryptography)

情報を、暗号文 (ciphertext) と呼ばれる、読み取り不能のフォーマットに変換 (暗号化) することによって保護すること。メッセージを暗号解除 (または復号) してプレーン・テキストにすることができるのは、秘密鍵の所有者のみです。

## D

### DAE

[ダンプ分析重複回避機能 \(dump analysis and elimination\)](#) を参照してください。

### デーモン (daemon)

無人で稼働し、ネットワーク制御などの連続的で定期的な機能を実行するプログラム。

### データ・バッグ (data bag)

MQAI がキュー・マネージャーの管理で使用するオブジェクト・プロパティのコンテナ。データ・バッグには、ユーザー (ユーザー・データ用)、管理 (前提オプションを使用した管理用)、およびコマンド (前提オプションを使用しない管理用) の 3 種類があります。

### データ変換インターフェース (DCI) (data-conversion interface (DCI))

異なるマシン・エンコーディングおよび CCSID 間でアプリケーション・データを変換するためのプログラムを、カスタマーまたはベンダーが作成する際に準拠しなければならない WebSphere MQ インターフェース。WebSphere MQ フレームワークの一部です。

### データ変換サービス (data-conversion service)

アプリケーション・データを他のプラットフォーム上のアプリケーションが必要とする文字セットおよびエンコーディングに変換するサービス。

### データグラム (datagram)

アプリケーションはメッセージを送信するが、応答を必要としない、非同期メッセージングの形式。 [要求/応答 \(request/reply\)](#) も参照してください。

### データ保全性 (data integrity)

データが無許可で変更されたり、改ざんされたりしていないかどうかを検出するセキュリティー・サービス。このサービスは、データが変更されたかどうかを検出するだけです。変更されていた場合、もとのデータはリストアされません。

### データ項目 (data item)

MQAI では、データ・バッグに含まれる項目。数字、文字列、ユーザーまたはシステム項目のいずれでもかまいません。

### DCE

[分散コンピューティング環境 \(Distributed Computing Environment\)](#) を参照してください。

### DCE プリンシパル (DCE principal)

分散コンピューティング環境を使うユーザー ID。

### DCI

[データ変換インターフェース \(data-conversion interface\)](#) を参照してください。

### DCM

[Digital Certificate Manager \(DCM\)](#) を参照してください。

### 送達不能キュー (DLQ) (dead-letter queue (DLQ))

キュー・マネージャーまたはアプリケーションが、正しい宛先へ配達できないメッセージを送信するキュー。

### 送達不能キュー・ハンドラー (dead-letter queue handler)

送達不能キュー (DLQ) をモニターし、キューにあるメッセージをユーザー作成の規則テーブルに従って処理するユーティリティー。WebSphere MQ には、送達不能キュー・ハンドラーのサンプルが用意されています。

### 復号 (decryption)

秘密フォーマットに暗号化されているデータをデコードする処理。復号には秘密鍵またはパスワードが必要です。

### デフォルト・オブジェクト (default object)

すべての属性が定義されているオブジェクト (例えば、キュー) の定義。ユーザーがオブジェクトを定義したときにそのオブジェクトに指定可能な属性すべてを指定しなかった場合、キュー・マネージャーは、指定されていない属性の代わりにデフォルト属性を使用します。

### 据え置き接続 (deferred connection)

CICS サブシステムが、開始される前の WebSphere MQ for z/OS に接続しようとした時にアクティブになる保留イベント。

### 派生 (derivation)

オブジェクト指向プログラミングで、あるクラスから別のクラスへの細分化または拡張。

### 宛先 (destination)

1. JMS で、メッセージの送受信の場所と方法を指定するオブジェクト。
2. メッセージの送信先のエンドポイント (キューまたはトピックなど)。

### Diffie-Hellman 鍵交換 (Diffie-Hellman key exchange)

安全でないチャネルで共有秘密鍵を安全に確立するために使用するパブリック鍵交換アルゴリズム。

### デジタル証明書 (digital certificate)

個人、システム、サーバー、企業、またはその他のエンティティを識別し、公開鍵をそのエンティティに関連付けるために使用される電子文書。デジタル証明書は認証局によって発行され、デジタル署名されます。

### Digital Certificate Manager (DCM)

IBM i システムで、デジタル証明書を管理し、iSeries サーバー上のセキュア・アプリケーションで使用するためのメソッド。デジタル証明書マネージャーは、認証局 (CA) またはその他のサード・パーティーからのデジタル証明書を要求および処理します。

### デジタル署名 (digital signature)

秘密鍵で暗号化され、メッセージまたはオブジェクトに付加される情報で、メッセージまたはオブジェクトの認証性および安全性を受信者に対して保証します。デジタル署名は、秘密鍵または共有秘密対称鍵を所有する、またはそのアクセス権限を持つエンティティによって、メッセージまたはオブジェクトが署名されたことを証明します。

### 切断 (disconnect)

アプリケーションとキュー・マネージャーの間の接続を切ること。

### 識別名 (DN) (distinguished name (DN))

デジタル証明書のエンティティを一意的に識別する名前と値のペア (「CN = 個人名」、「C = 国名」など) のセット。識別名は、特定の認証局の名前空間内のみで一意であることに注意してください。異なる認証局によって同一の識別名で証明書が発行される可能性は十分に考えられます。そのため、鍵リポジトリには、トラステッド・ルート CA 証明書が可能な限り少なくなるように、可能であれば 1 つだけにします。認証局 (certificate authority)、[デジタル証明書 \(digital certificate\)](#)、[X509](#) も参照。

### 分散アプリケーション (distributed application)

メッセージ・キューイングにおいて、それぞれが異なるキュー・マネージャーに接続できる一方で、全体として単一のアプリケーションを構成するアプリケーション・プログラムの集合。

### 分散コンピューティング環境 (DCE) (Distributed Computing Environment (DCE))

ネットワーク・コンピューティングにおいて、異機種混合のオペレーティング・システムとネットワークの環境で分散アプリケーションの作成、使用、および保守をサポートするサービスとツールのセット。

### 分散キュー管理 (distributed queue management)

メッセージ・キューイングにおいて、他システム上のキュー・マネージャーに対してメッセージ・チャネルのセットアップおよび管理すること。

### 配布リスト (distribution list)

単一のステートメントでメッセージの送付先にできるキューのリスト。

### DLQ

[送達不能キュー \(dead-letter queue\)](#) を参照してください。

### DN

[識別名 \(distinguished name\)](#) を参照してください。

### 重複ロギング (dual logging)

WebSphere MQ for z/OS アクティビティを記録する方式。すべての変更内容が2つのデータ・セットに記録されるため、再始動が必要な時に片方のデータ・セットが読み取り不能でも、別のデータ・セットを使うことができます。[単一ロギング \(single logging\)](#) も参照してください。

### 重複モード (dual mode)

[重複ロギング \(dual logging\)](#) を参照してください。

### ダンプ分析重複回避機能 (DAE) (dump analysis and elimination (DAE))

以前に書き込まれたダンプを重複するため、必要ではない SVC ダンプおよび ABEND SYSUDUMP ダンプを抑制する z/OS サービス。

### 永続サブスクリプション (durable subscription)

サブスクライブしているアプリケーションのキュー・マネージャーへの接続がクローズされたときに保存されるサブスクリプション。サブスクライブ・アプリケーションが切断されても、永続サブスクリプションはそのままそこに残り、パブリケーションの配信は継続されます。アプリケーションが再接続したときは、固有のサブスクリプション名を指定することで同じサブスクリプションを使用できます。[非永続サブスクリプション \(nondurable subscription\)](#) も参照してください。

### 動的キュー (dynamic queue)

プログラムがモデル・キュー・オブジェクトをオープンする時に作成されるローカル・キュー。

## E

### 盗聴 (eavesdropping)

情報は損なわれなくても、プライバシーが危険に晒される通信セキュリティの侵害。[偽名の使用 \(impersonation\)](#)、[改ざん \(tampering\)](#) も参照してください。

### Eclipse

独立系ソフトウェア・ベンダー (ISV) やその他のツール・デベロッパーに対して、プラグ・コンパチブルなアプリケーション開発ツールを開発するための標準プラットフォームを提供する、オープン・ソース・イニシアチブ。

### カプセル化 (encapsulation)

オブジェクト指向プログラミングで、オブジェクト、関数、またはクラスの固有の詳細をクライアント・プログラムから隠蔽するために使用される技法。

### 暗号化 (encryption)

コンピューター・セキュリティにおいて、読むことができない形式にデータを変換して、元のデータを取得できないように、あるいは暗号化解除プロセスを使用した場合のみ元のデータを取得できるようにするプロセス。

### エンキュー (enqueue)

メッセージやアイテムをキューに置くこと。

### エンティティ (entity)

RACF などのセキュリティ・サービスに対して定義されるユーザー、グループ、あるいはリソース。

### 環境変数

オペレーティング・システムや他のプログラムがどのように稼働するかを指定したり、オペレーティング・システムが認識する装置を指定したりする変数。

### ESM

[外部セキュリティ・マネージャー \(external security manager\)](#) を参照してください。

### ESTAE

[拡張タスク異常終了指定出口 \(extended specify task abnormal exit\)](#) を参照してください。

### イベント・データ (event data)

イベント・メッセージ内で、イベントについての情報(キュー・マネージャー名、イベントを発生させたアプリケーションなど)を含んでいる部分。[イベント・ヘッダー \(event header\)](#) も参照。

### イベント・ヘッダー (event header)

イベント・メッセージ内で、イベントの理由コードのイベント・タイプを識別するメッセージ・データの部分。[イベント・データ \(event data\)](#) も参照。

### イベント・メッセージ (event message)

WebSphere MQ システム・ネットワーク内の観測イベントの発信元に関する情報 (イベントのカテゴリ、イベントを発生させたアプリケーション名、キュー・マネージャー統計など) を含むメッセージ。

### イベント・キュー (event queue)

キュー・マネージャーがイベントを検出した後、イベント・メッセージを書き込むキュー。イベントのカテゴリ (キュー・マネージャー、パフォーマンス、構成、計測、またはチャンネル・イベント) ごとに独自のイベント・キューがあります。

### イベント・ビューアー (Event Viewer)

ログ・ファイルを検査および管理するために Windows システムが提供するツール。

### 例外リスナー (exception listener)

アプリケーションによって登録することができるクラスのインスタンス。このリスナー用に `onException()` メソッドが呼び出されて、アプリケーションに JMS 例外が非同期に渡されます。

### 排他メソッド (exclusive method)

オブジェクト指向プログラミングで、ポリモアフィズムを提示しない、特定の効果を持つメソッド。

### 拡張タスク異常終了指定出口 (ESTAE) (extended specify task abnormal exit (ESTAE))

リカバリー機能を提供する z/OS マクロ。処理や、異常終了の診断、再試行アドレスの指定を行うために、ユーザー指定の出口ルーチンに制御を渡します。

### 外部セキュリティー・マネージャー (ESM) (external security manager (ESM))

ユーザーまたはリソースに対するセキュリティー検査を実行するセキュリティー製品。RACF は ESM の一例です。

## F

### フェイルオーバー

ソフトウェア、ハードウェア、またはネットワークの障害が発生した場合に、冗長システムまたは待機システムに自動的に切り替わること。

### FAP

[フォーマットおよびプロトコル \(Formats and Protocols\)](#) を参照してください。

### FFDC

[First Failure Data Capture](#) を参照してください。

### FFST

[First Failure Support Technology](#) を参照してください。

### FFST ファイル (FFST file)

[First Failure Support Technology ファイル \(First Failure Support Technology file\)](#) を参照してください。

### FIFO

「[先入れ先出し法 \(first-in first-out\)](#)」を参照。

### FIPS

米国連邦情報処理標準 (Federal Information Processing Standards)

### First Failure Data Capture (FFDC)

- エラーを識別し、それらのエラーに関する情報を収集してログに記録し、影響を受けるランタイム・ソフトウェアに制御を戻すための、問題診断援助機能。
- FFST アーキテクチャーの IBM i 実装で、問題の認識、診断データの指定域ダンプ、症状ストリングの生成、および問題ログの入力が行える。

### First Failure Support Technology (FFST)

防御的プログラミング手法によるエラー検出のための単一方法を定義する IBM アーキテクチャー。これらの手法によって、事前の対策を講じた (必要になるまでは受動状態) 問題認識、およびソフトウェア問題のデバッグに必要な診断出力の説明が可能になる。

### First Failure Support Technology ファイル (FFST ファイル) (First Failure Support Technology file (FFST file))

ソフトウェアの問題の検出および診断に使用される情報が格納されたファイル。WebSphere MQ では、FFST ファイルのファイル・タイプは FDC です。

### 先入れ先出し法 (FIFO) (first-in first-out (FIFO))

キューに最も長い間入っていた項目が、次に検索される項目となるような仕方で検索が行われる、キューイングの技法。

### 強制シャットダウン (forced shutdown)

CICS アダプターのシャットダウンのタイプで、現在アクティブなタスクの状況に関係なく、ただちに WebSphere MQ for z/OS からシャットダウンするもの。静的シャットダウン (quiesced shutdown) も参照してください。

### 形式

メッセージ・キューイングで、メッセージ内のアプリケーション・データの性質を識別するために使われる用語。

### フォーマットおよびプロトコル (FAP) (formats and Protocols (FAP))

メッセージ・キューイングにおいて、キュー・マネージャー間で相互に通信する方法や、クライアントがサーバー・キュー・マネージャーと通信する方法の定義。

### フレームワーク (Framework)

WebSphere MQ で、カスタマーまたはベンダーが、WebSphere MQ 製品が提供する機能の一部を拡張または置換するためのプログラムを作成できるようにするプログラミング・インターフェースの集合。このようなインターフェースとしては、データ変換インターフェース (DCI)、メッセージ・チャンネル・インターフェース (MCI)、ネーム・サービス・インターフェース (NSI)、セキュリティ・イネープリング・インターフェース (SEI)、トリガー・モニター・インターフェース (TMI) があります。

### フレンド・クラス (friend class)

すべてのメンバー関数が他のクラスのプライベート・メンバーおよび保護されたメンバーへのアクセス権限を付与されているクラス。フレンド・クラスは、他のクラスの宣言の中で指定され、クラスの接頭部としてキーワード friend を使用します。

### FRR

[機能リカバリー・ルーチン \(functional recovery routine\)](#) を参照してください。

### 完全リポジトリ (full repository)

クラスター内のすべてのキュー・マネージャーについての完全な情報セット。この情報セットは、リポジトリまたは完全リポジトリと呼ばれ、通常はクラスター内の 2 つのキュー・マネージャーに保管されています。[部分リポジトリ \(partial repository\)](#) も参照してください。

### 関数

呼び出されて、評価され、呼び出し側ステートメントに値を戻すことができる、名前付きのステートメント・グループ。

### 機能リカバリー・ルーチン (FRR) (functional recovery routine (FRR))

プログラムが中断された時にリカバリー・ルーチンが制御を獲得できるようにする、z/OS リカバリーおよび終了マネージャー。

## G

### ゲートウェイ・キュー・マネージャー (gateway queue manager)

アプリケーションからクラスター内の他のキュー・マネージャーへメッセージを送付するのに使用する、クラスター・キュー・マネージャー。

### 汎用トレース機能 (GTF) (generalized trace facility (GTF))

I/O 割り込み、SVC 割り込み、プログラム割り込み、外部割り込みなどの重要なシステム・イベントを記録する z/OS サービス・プログラム。

### 汎用セキュリティ・サービス API (Generic Security Services API)

[汎用セキュリティ・サービス・アプリケーション・プログラミング・インターフェース \(Generic Security Services application programming interface\)](#) を参照してください。

### 汎用セキュリティ・サービス・アプリケーション・プログラミング・インターフェース (Generic Security Services API, GSS API) (Generic Security Services application programming interface (Generic Security Services API, GSS API))

セキュリティ・サービスにアクセスするための共通アプリケーション・プログラミング・インターフェース (API)。

## get

メッセージ・キューイングで、MQGET 呼び出しを使用してメッセージをキューから削除し、そのコンテナを呼び出し側のアプリケーションに戻すこと。 [ブラウズ \(browse\)](#)、[書き込み \(put\)](#) も参照してください。

## グローバル定義オブジェクト (globally defined object)

z/OS で、その定義が共用リポジトリに保管されているオブジェクト。このオブジェクトは、キュー・マネージャー・グループ内のすべてのキュー・マネージャーが使うことができます。 [ローカル定義のオブジェクト \(locally defined object\)](#) も参照してください。

## グローバル・トレース (global trace)

WebSphere MQ for z/OS サブシステム全体からトレース・データが送られてくる WebSphere MQ for z/OS トレース・オプション。

## グローバル・トランザクション (global transaction)

分散トランザクション環境で 1 つ以上のリソース・マネージャーによって実行され、外部トランザクション・マネージャーによって調整されるリカバリー可能な作業単位。

## GSS API

[汎用セキュリティ・サービス・アプリケーション・プログラミング・インターフェース \(Generic Security Services application programming interface\)](#) を参照してください。

## GTF

[汎用トレース機能 \(generalized trace facility\)](#) を参照してください。

# H

## ハンドシェイク (handshake)

Secure Sockets Layer (SSL) セッションの開始時に行われるメッセージ交換。これにより、公開鍵の手法を用いてクライアントがサーバーを認証できるようになり (場合によっては、サーバーもクライアントを認証する)、クライアントとサーバーが協力して暗号化、復号化、および改ざん検出のための対称鍵を作成できるようになる。

## ハード化メッセージ (hardened message)

補助 (ディスク) ストレージに書き込まれたメッセージ。システム障害が起きても消えないメッセージです。

## ヘッダー

[メッセージ・ヘッダー \(message header\)](#) を参照してください。

## ハートビート (heartbeat)

エンティティがまだアクティブであることを通知するために別のエンティティに送信する信号。

## ハートビート・フロー (heartbeat flow)

送信するメッセージがない場合に、送信側のメッセージ・チャンネル・エージェント (MCA) から受信側の MCA に渡されるパルス。パルスは受信側の MCA を非ブロック化します。それ以外の場合は、メッセージが到着または切断間隔が満了するまで、受信側の MCA は待ち状態のままになります。

## ハートビート間隔 (heartbeat interval)

ハートビート・フロー間の経過時間を秒単位で示すもの。

## 階層 (hierarchy)

パブリッシュ/サブスクライブ・メッセージング・トポロジーにおいて、親キュー・マネージャーに接続されたローカル・キュー・マネージャー。

## HTTP

[Hypertext Transfer Protocol](#) を参照してください。

## Hypertext Transfer Protocol (HTTP)

Web 上のハイパーテキスト文書および XML 文書を転送および表示するために使用されるインターネット・プロトコル。

# I

## ID コンテキスト (identity context)

キューに最初にメッセージを書き込むアプリケーションのユーザーを識別する情報。

## 識別

コンピューター・システムの各ユーザーを一意的に識別することのできるセキュリティー・サービス。このサービスを実現するための一般的な方法は、各ユーザーにユーザー ID を割り当てることです。

## identity コンテキスト (identity context)

キューにメッセージを最初書き込んだアプリケーションのユーザーを識別する情報。

## IFCID

[観測機能コンポーネント ID \(instrumentation facility component identifier\)](#) を参照してください。

## ILE

[Integrated Language Environment](#) を参照してください。

## 即時シャットダウン (immediate shutdown)

WebSphere MQ で、アプリケーションの切断を待たないキュー・マネージャーのシャットダウン。現在のメッセージ・キュー・インターフェース (MQI) 呼び出しは完了できますが、即時シャットダウンが要求された後の新規 MQI 呼び出しはすべて失敗します。[優先シャットダウン \(preemptive shutdown\)](#)、[静的シャットダウン \(quiesced shutdown\)](#) も参照してください。

## 偽名の使用 (impersonation)

情報が本来の受信者ではなくそのふりをした他人に渡されたり、または情報が他人のふりをした誰かによって送信されることによる通信セキュリティーの侵害。「[盗聴 \(eavesdropping\)](#)」、「[改ざん \(tampering\)](#)」も参照。

## インバウンド・チャネル (inbound channel)

他のキュー・マネージャーからメッセージを受信するチャネル。

## 組み込みフォーマット (in-built format)

[ビルトイン・フォーマット \(built-in format\)](#) を参照してください。

## 索引 (index)

WebSphere MQ 管理インターフェース (MQAI) で、データ項目を参照する方法。

## 未確定の回復単位

同期点が要求されたがまだ確認されていない、リカバリー単位の状況。

## 未完了 (inflight)

コミット・プロセスの準備フェーズを完了していない、リソースまたはリカバリー単位の状況。

## 継承 (inheritance)

既存のクラスを他のクラスを作成するための基礎として使用するオブジェクト指向プログラミング技法。継承によって、より一般化されたエレメントの構造および動作が、より特殊化されたエレメントに組み込まれる。

## 初期化入力データ・セット (initialization input data set)

WebSphere MQ for z/OS が始動時に使用するデータ・セット。

## 開始キュー

キュー・マネージャーがトリガー・メッセージを入れるローカル・キュー。

## イニシエーター (initiator)

分散キューイングにおいて、別のシステムにネットワーク接続を要求するプログラム。「[応答側 \(responder\)](#)」も参照してください。

## 入力パラメーター (input parameter)

呼び出しを行う時にユーザーが情報を入力する、MQI 呼び出しのパラメーター。

## 挿入順序 (insertion order)

WebSphere MQ 管理インターフェース (MQAI) で、データ項目がデータ・バッグに書き込まれる順序。

## インストール可能サービス (installable service)

UNIX および Linux システム上の WebSphere MQ、および WebSphere MQ for Windows では、独立したコンポーネントとして提供される追加機能。各コンポーネントをインストールするかどうかは任意選択で、お客様作成のコンポーネント、またはサードパーティのコンポーネントを代わりに使用することができます。

## インスタンス (instance)

あるクラスに属するオブジェクトの特定のオカレンス。「[オブジェクト \(object\)](#)」も参照。

### インスタンス・データ (instance data)

オブジェクト指向プログラミングで、オブジェクトに関連付けられている状態の情報。

### 観測イベント (instrumentation event)

WebSphere MQ システムのネットワークで、キュー・マネージャーのリソース定義、パフォーマンス条件、およびチャネル条件をモニターする方法。

### 観測機能コンポーネント ID (IFCID) (instrumentation facility component identifier (IFCID))

Db2 for z/OS で、イベントのトレース記録を指定し、識別する値。START TRACE および MODIFY TRACE コマンドのパラメーターとして、対応するイベントをトレースするよう指定します。

### Integrated Language Environment (ILE)

ILE 準拠のすべての高水準言語に対して、共通のランタイム環境とランタイム・バインド可能アプリケーション・プログラム・インターフェース (API) を提供する構成およびインターフェースのセット。

### 対話式問題制御システム (IPCS) (Interactive Problem Control System (IPCS))

MVS and z/OS のコンポーネントで、オンライン問題管理、対話式問題診断、ディスク常駐の異常終了ダンプのオンライン・デバッグ、問題のトラッキング、および問題報告を可能にする。

### 対話式システム生産性向上機能 (ISPF) (Interactive System Productivity Facility (ISPF))

フル・スクリーン・エディターおよびダイアログ・マネージャーとして機能する IBM ライセンス・プログラム。ISPF はアプリケーション・プログラムを作成するために使われ、標準的な画面パネルおよびアプリケーション・プログラマーと端末ユーザー間の対話式ダイアログを生成するための手段を提供します。

### 中間証明書 (Intermediate certificate)

ルート証明書ではない署名者証明書。

### インターフェース

オブジェクト指向プログラミングにおける動作の抽象モデルであり、関数またはメソッドの集合。

### インターネット・プロトコル (IP) (internet Protocol (IP))

単一ネットワークまたは相互接続ネットワークを通してデータを送信するプロトコル。このプロトコルは、より上位のプロトコル層と物理的なネットワーク間の仲介をします。「[伝送制御プロトコル \(Transmission Control Protocol\)](#)」も参照。

### プロセス間通信 (IPC) (interprocess communication (IPC))

プログラム間でメッセージを送信し合うためのプロセス。ソケット、セマフォ、信号、および内部メッセージ・キューは、プロセス間通信の一般的なメソッドである。「[クライアント \(client\)](#)」も参照。

### システム間通信 (ISC) (intersystem communication (ISC))

他のコンピューター・システムからの通信に関してインバウンドおよびアウトバウンド・サポートを提供する CICS 機能。

### IP

[インターネット・プロトコル \(Internet Protocol\)](#) を参照してください。

### IPC

[プロセス間通信 \(interprocess communication\)](#) を参照してください。

### IPCS

[対話式問題制御システム \(Interactive Problem Control System\)](#) を参照してください。

### ISC

[システム間通信 \(intersystem communication\)](#) を参照してください。

### ISPF

[対話式システム生産性向上機能 \(Interactive System Productivity Facility\)](#) を参照してください。

## J

### JAAS

「[Java 認証・承認サービス \(Java Authentication and Authorization Service\)](#)」を参照してください。

### Java 認証・承認サービス (JAAS) (Java Authentication and Authorization Service (JAAS))

Java EE テクノロジーにおいて、セキュリティー・ベースのオペレーションを実行するための標準 API。サービスは、JAAS を介して、ユーザーを認証および承認し、基礎となるテクノロジーからアプリケーションを独立させておくことを可能にする。

## Java Message Service (JMS)

メッセージ処理のための Java 言語機能を提供する、アプリケーション・プログラミング・インターフェース。「[メッセージ・キュー・インターフェース \(Message Queue Interface\)](#)」も参照してください。

## Java ランタイム環境 (Java runtime environment) (JRE)

標準的 Java プラットフォームを構成する中核の実行可能プログラムおよびファイルを含む Java Developer Kit のサブセット。JRE には、Java 仮想マシン (JVM)、コア・クラス、およびサポート・ファイルが含まれています。

## JMS

[Java Message Service](#) を参照してください。

## JMSAdmin

管理者が JMS オブジェクトのプロパティを定義し、それらを JNDI 名前空間内に保管できるようにする管理ツール。

## ジャーナル (journal)

OS/400 のフィーチャーで、WebSphere MQ for iSeries がローカル・オブジェクトの更新を制御するのに使用する。各キュー・マネージャー・ライブラリーに、そのキュー・マネージャーのジャーナルが含まれます。

## JRE

「[Java ランタイム環境 \(Java runtime environment\)](#)」を参照してください。

# K

## キープアライブ (keepalive)

ソケットがまだ正しく機能しているかどうかを判別するために、事前定義した間隔でネットワーク上に小さいパケットを送信する TCP/IP メカニズム。

## Kerberos

対称鍵暗号方式に基づくネットワーク認証プロトコル。Kerberos は、ネットワークにログオンするユーザーに対し、それぞれ「チケット」と呼ばれる固有キーを割り当てる。チケットは、ネットワークを介して送信されるメッセージに埋め込まれている。メッセージの受信者は、チケットを使って送信者を認証します。

## キー認証 (key authentication)

[認証 \(authentication\)](#) を参照してください。

## 鍵リポジトリ (key repository)

デジタル証明書とそれに関連付けられる鍵のストアの総称。鍵リポジトリの種類には、Certificate Management System (CMS)、Java Keystore (JKS)、Java Cryptography Extension Keystore (JCEKS)、Public Key Cryptography Standard 12 (PKCS12) 鍵ストア、RACF 鍵リングなどがあります。鍵リポジトリ・タイプを区別することが重要になる場合、文書は固有の名前で鍵リポジトリ・タイプを参照します。複数の鍵ストア・タイプを適用可能なコンテキストでは、総称鍵リポジトリが使用されません。

## 鍵リング

コンピューター・セキュリティーにおいて、公開鍵、秘密鍵、トラステッド・ルート、および証明書を含むファイル。

## 鍵ストア (key store)

秘密鍵および対応する個人証明書を格納する場所。[トラスト・ストア \(trust store\)](#)も参照。

# L

## 遺言 (last will and testament)

モニターを使用するクライアントにより登録されるオブジェクトで、クライアントが予期しない形で終了した場合にモニターによって使用される。

## LDAP

[Lightweight Directory Access Protocol](#) を参照してください。

## Lightweight Directory Access Protocol (LDAP)

TCP/IP を使用して、X.500 モデルをサポートするディレクトリーにアクセスできるようにするためのオープン・プロトコル。このプロトコルは、より複雑な X.500 Directory Access Protocol (DAP) のリソ

ース要件には対応しません。例えば LDAP は、インターネットまたはイントラネットのディレクトリー内のユーザー、組織、およびその他のリソースを探し出すのに使用できます。

### リニア・ロギング (linear logging)

UNIX および Linux システム上の WebSphere MQ、および WebSphere MQ for Windows において、再始動データを一連のファイルに保持するプロセス。必要に応じて、新しいファイルがシーケンスに追加されます。データが書き込まれるスペースは再利用されません。「[循環ロギング \(circular logging\)](#)」も参照してください。

### リンク・レベル・セキュリティー (link level security)

メッセージ・チャネル・エージェント (MCA)、通信サブシステム、またはこれらの組み合わせから直接的または間接的に起動されるセキュリティー・サービス。

### リスナー

到着する要求を検出して関係するチャネルを始動するプログラム。

### リモート・キューのローカル定義 (local definition of a remote queue)

ローカル・キュー・マネージャーに属する WebSphere MQ オブジェクト。別のキュー・マネージャーが所有するキューの属性を定義します。また、キュー・マネージャーの別名および応答先キューの別名としても使用されます。

### ロケール (locale)

言語や地域を識別し、フォーマットの規則 (照合、大/小文字変換、文字種別、メッセージに使用する言語、日時の表記、数値表現など) を決定する設定。

### ローカル定義オブジェクト (locally defined object)

z/OS で、その定義がゼロに設定されたページ上に保管されているオブジェクト。オブジェクトを定義したキュー・マネージャーだけが、その定義にアクセスすることができます。「[グローバル定義オブジェクト \(globally defined object\)](#)」も参照してください。

### ローカル・キュー

ローカル・キュー・マネージャーに属するキュー。ローカル・キューには、処理を待っているメッセージのリストが含まれています。「[リモート・キュー \(remote queue\)](#)」も参照してください。

### ローカル・キュー・マネージャー (local queue manager)

プログラムの接続先であり、そのプログラムに対してメッセージ・キューイングのサービスを提供するキュー・マネージャー。「[リモート・キュー・マネージャー \(remote queue manager\)](#)」も参照してください。

### log

WebSphere MQ で、障害が起きた場合に回復できるように、キュー・マネージャーが受信、転送、および配達したメッセージを記録するファイル。

### ログ制御ファイル (log control file)

UNIX および Linux システム上の WebSphere MQ、および WebSphere MQ for Windows において、ログ・ファイルの使用をモニターするために必要な情報 (例えば、ログ・ファイルのサイズと場所、次に使用可能なファイルの名前など) を含むファイル。

### ログ・ファイル (log file)

UNIX 上の WebSphere MQ、Linux システム、および WebSphere MQ for Windows では、キュー・マネージャーによって制御されるデータに対するすべての重要な変更が記録されるファイル。最初のログ・ファイルがいっぱいになると、WebSphere MQ は第 2 のログ・ファイルを割り当てます。

### 論理単位 (LU) (logical unit (LU))

ユーザーまたはアプリケーション・プログラムが他のユーザーまたはアプリケーション・プログラムと通信するために SNA ネットワークにアクセスする際に経由するアクセス・ポイント。

### 論理装置タイプ 6.2 (LU 6.2) (logical unit 6.2 (LU 6.2))

SNA 論理装置で、分散処理環境におけるプログラム間の汎用通信をサポートする。

### 論理作業単位の識別子 (LUWID) (logical unit of work identifier (LUWID))

ネットワーク内のスレッドを一意的に識別する名前。この名前は、完全修飾論理装置のネットワーク名、作業論理単位のインスタンス番号、および作業論理単位のシーケンス番号で構成される。

### ログ・レコード (log record)

ログ・ファイルで単一ユニットとして扱われるデータのセット。

## ログ・レコード・シーケンス番号 (LRSN) (log record sequence number (LRSN))

ログ・レコードの固有 ID で、データ共有メンバーと関連付けられている。Db2 for z/OS はデータ共有環境でのリカバリーに LRSN を使用する。

## LRSN

[ログ・レコード・シーケンス番号 \(log record sequence number\)](#) を参照してください。

## LU

[論理単位 \(logical unit\)](#) を参照してください。

## LU 6.2

「[論理装置タイプ 6.2 \(logical unit 6.2\)](#)」を参照してください。

## LU6.2 会話 (LU 6.2 conversation)

SNA で、2 つのトランザクション・プログラム間において相互通信を可能にする LU 6.2 セッションを介した論理接続。

## LU6.2 会話レベル・セキュリティー (LU 6.2 conversation level security)

SNA で、パートナーのトランザクション・プログラムが会話を始めたトランザクション・プログラムを認証できるようにする、会話レベルのセキュリティー・プロトコル。LU 6.2 会話レベル・セキュリティーは、エンド・ユーザー検査とも呼ばれます。

## LU 6.2 セッション (LU 6.2 session)

SNA で、タイプ 6.2 の 2 つの論理単位 (LU) 間のセッション。

## LU 名

VTAM がネットワーク内のノードを参照するのに使用する名前。

## LUWID

[論理作業単位の識別子 \(logical unit of work identifier\)](#) を参照してください。

# M

## 管理宛先 (managed destination)

管理サブスクリプションを使用することを選択したアプリケーション用に、パブリッシュされたメッセージを送信する宛先として、キュー・マネージャーによって提供されるキュー。「[管理サブスクリプション \(managed subscription\)](#)」も参照してください。

## 管理ハンドル (managed handle)

キュー・マネージャーが、サブスクリプションに送信されたメッセージの保管を管理するように指定されているときに MQSUB 呼び出しによって戻される識別子。

## 管理サブスクリプション (managed subscription)

アプリケーションが特定のキューの使用を要求しないため、キュー・マネージャーがパブリケーションを受信するためにサブスクリャイバー・キューを作成するサブスクリプション。「[管理宛先 \(managed destination\)](#)」も参照してください。

## マーシャル (marshalling)

[シリアライゼーション \(serialization\)](#) を参照してください。

## MCA

[メッセージ・チャネル・エージェント \(message channel agent\)](#) を参照してください。

## MCI

[メッセージ・チャネル・インターフェース \(message channel interface\)](#) を参照してください。

## メディア・イメージ (media image)

UNIX および Linux システム上の WebSphere MQ、および WebSphere MQ for Windows で、オブジェクトのイメージを含むログ・レコードのシーケンス。オブジェクトはこのイメージから再作成することができます。

## メッセージ

1. 1 人の個人または 1 つのプログラムから別の個人またはプログラムへ送られる通信。
2. システム・プログラミングで、端末オペレーターまたはシステム管理者に宛てられた情報。

### メッセージ・アフィニティー (message affinity)

2つのアプリケーション間で交換される会話型メッセージ間の関係で、メッセージが特定のキュー・マネージャーによって、または特定の順序で処理される必要があるもの。

### メッセージ・チャンネル (message channel)

分散メッセージ・キューイングで、メッセージをあるキュー・マネージャーから別のキュー・マネージャーに移動するメカニズム。メッセージ・チャンネルは、2つのメッセージ・チャンネル・エージェント (一方は送信側で他方は受信側) および1つの通信リンクにより構成されます。「[チャンネル \(channel\)](#)」も参照してください。

### メッセージ・チャンネル・エージェント (MCA) (message channel agent (MCA))

準備されたメッセージを、伝送キューから通信リンクに、あるいは通信リンクからターゲット・キューに伝送するプログラム。「[メッセージ・キュー・インターフェース \(Message Queue Interface\)](#)」も参照してください。

### メッセージ・チャンネル・インターフェース (MCI) (message channel interface (MCI))

WebSphere MQ キュー・マネージャー間、および別のメッセージング・システム間でメッセージを転送する、顧客作成またはベンダー作成のプログラムが準拠する必要のある WebSphere MQ インターフェース。WebSphere MQ フレームワークの一部です。「[メッセージ・キュー・インターフェース \(Message Queue Interface\)](#)」も参照してください。

### メッセージ・コンシューマー (message consumer)

1. メッセージを取得して処理するプログラムまたは機能。[コンシューマー \(consumer\)](#) も参照してください。
2. JMS において、宛先からメッセージを受信するためにセッション内で作成されるオブジェクト。

### メッセージ・コンテキスト (message context)

メッセージ記述子内のフィールドに保管されている、メッセージの発信元についての情報。コンテキスト情報には、origin と identity という2つのカテゴリがあります。

### メッセージ記述子

WebSphere MQ メッセージの一部として運ばれる、メッセージ形式と表示形式を記述した制御情報。メッセージ記述子の形式は、MQMD 構造によって定義されます。

### メッセージ出口 (message exit)

メッセージの内容を変更するのに使用されるチャンネル出口プログラム的一种。メッセージ出口は通常はチャンネルの各先端にひとつずつのペアで稼働します。チャンネルの送信側終端では、メッセージ・チャンネル・エージェント (MCA) が伝送キューからメッセージを受け取った後にメッセージ出口が呼び出されます。チャンネルの受信側終端では、メッセージ・チャンネル・エージェント (MCA) がターゲット・キューにメッセージを書き込む前にメッセージ出口が呼び出されます。

### メッセージ・フロー制御 (message flow control)

キュー・マネージャー間のメッセージ・ルートをセットアップおよび維持する分散キュー管理タスク。

### メッセージ・フォーマット・サービス (MFS) (Message Format Service (MFS))

IMS 編集機能の1つで、この機能を利用すると、アプリケーション・プログラムは、装置に依存するようなデータではなく、単純な論理メッセージが処理できるようになり、これによってアプリケーション開発プロセスを簡素化することができる。

### メッセージ・グループ (message group)

関連するメッセージの論理グループ。この関係はメッセージを書き込んでいるアプリケーションによって定義されて、プロデューサーとコンシューマーの両方がグループ化を順守している場合は、必ず書き込まれた順序で取り出されるようになる。

### メッセージ・ハンドル (message handle)

メッセージへの参照。ハンドルは、メッセージのメッセージ・プロパティへのアクセス取得に使用できます。

### メッセージ・ヘッダー (message header)

固有のメッセージ ID、メッセージの送信側と受信側、メッセージ優先順位、およびメッセージのタイプなどの制御情報を含むメッセージ部分。

### メッセージ入力記述子 (MID) (message input descriptor (MID))

アプリケーション・プログラムによって提示されるデータの形式を記述するメッセージ・フォーマット・サービス (MFS) 制御ブロック。「[メッセージ出力記述子 \(message output descriptor\)](#)」も参照してください。

### メッセージ・リスナー (message listener)

非同期メッセージ・コンシューマーとして動作するオブジェクト。

### メッセージ出力記述子 (MOD) (message output descriptor (MOD))

アプリケーション・プログラムによって生成される出力データの形式が記述されているメッセージ形式サービス (MFS) 制御ブロック。「[メッセージ入力記述子 \(message input descriptor\)](#)」も参照してください。

### メッセージ優先順位 (message priority)

WebSphere MQ で、キュー上のメッセージが検索される順序およびトリガー・イベントが生成されるかどうかに影響を与えるメッセージ属性。

### メッセージ・プロデューサー (message producer)

JMS においては、セッションにより作成され、宛先にメッセージを送信するために使用されるオブジェクト。

### メッセージ・プロパティ (message property)

名前と値のペア形式の、メッセージに関連したデータ。メッセージ・プロパティは、パブリケーションをフィルターに掛けたり、キューからメッセージを選択的に取得する場合にメッセージ・セレクターとして使用できる。メッセージ・プロパティを使用すると、メッセージ本体を変更することなく、処理に関する業務データや状態情報を組み込むことができる。

### メッセージ・キュー・インターフェース (MQI) (Message Queue Interface (MQI))

WebSphere MQ キュー・マネージャーが提供するプログラミング・インターフェース。プログラミング・インターフェースにより、アプリケーション・プログラムはメッセージ・キューイング・サービスにアクセスすることができます。[Java Message Service](#)、[メッセージ・チャンネル・エージェント \(message channel agent\)](#)、[メッセージ・チャンネル・インターフェース \(message channel interface\)](#) も参照してください。

### メッセージ・キュー管理 (MQM) (message queue management (MQM))

WebSphere MQ for HP Integrity NonStop Server で、PCF コマンド形式と制御コマンドを使用する方法を提供することにより、キュー・マネージャー、キュー、およびチャンネルを管理する機能。

### メッセージ・キューイング (message queuing)

アプリケーション内の各プログラムが、他のプログラムとの通信を、キューにメッセージを入れることで行うプログラミング技法。

### メッセージ再試行 (message-retry)

MCA がメッセージを書き込むことができない場合に使用可能なオプション。MCA は、事前定義された時間だけ待機し、その後でメッセージの書込みを再試行します。

### メッセージ・セグメント (message segment)

大きすぎるため、アプリケーションやキュー・マネージャーが処理できないメッセージの幾つかのセグメントの 1 つ。

### メッセージ・セレクター (message selector)

アプリケーション・プログラミングにおいて、アプリケーションが使用する可変長ストリングで、その選択ストリングで示す構造化照会言語 (SQL) の照会を満たすプロパティのメッセージのみにインタレストを登録する。メッセージ・セレクターの構文は、SQL92 条件式構文のサブセットに基づきます。

### メッセージ・シーケンス番号付け (message sequence numbering)

通信リンク上での伝送中に、メッセージに固有の番号を与えるプログラミング技法。これにより、すべてのメッセージが受信されたかどうかの検査、メッセージを元の順序でキュー内に配置すること、および重複メッセージを廃棄するための受信処理が可能になります。

### メッセージ・トークン (message token)

アクティブ・キュー・マネージャー内のメッセージの固有 ID。

### メソッド (method)

オブジェクト指向設計またはプログラミングで、操作によって指定される動作を実装するソフトウェア。

## MFS

[メッセージ・フォーマット・サービス \(Message Format Service\)](#) を参照してください。

## MGAS

「[大部分がグローバルなアドレス・スペース \(mostly global address space\)](#)」を参照してください。

## Microsoft Cluster Server (MSCS)

Windows を実行している複数のコンピューターを MSCS クラスターにグループ化することで高可用性を提供するテクノロジー。クラスター内のコンピューターの 1 つに問題が起きた場合、MSCS は中断されたアプリケーションを通常の方法でシャットダウンし、その状態データをクラスター内の別のコンピューターに転送して、そこでアプリケーションを再始動します。

## Microsoft Transaction Server (MTS)

Windows ユーザーが中間層サーバーでビジネス・ロジック・アプリケーションを実行できるようにする機能。MTS は、短い独立したビジネス・ロジックの塊であるアクティビティーに作業を分解します。

## MID

[メッセージ入力記述子 \(message input descriptor\)](#) を参照してください。

## MOD

[メッセージ出力記述子 \(message output descriptor\)](#) を参照してください。

## モデル・キュー・オブジェクト (model queue object)

プログラムが動的キューを作成するときにテンプレートとして動作するキュー属性のセット。

## 大部分がグローバルなアドレス・スペース (MGAS) (mostly global address space (MGAS))

柔軟な仮想アドレス・スペース・モデル。アドレス・スペースのほとんどを共有アプリケーションのために保持する、HP-UX などのシステムで使用される。これにより、多くのデータを共有するプロセスのパフォーマンスが高められる場合がある。「[大部分がプライベートなアドレス・スペース \(mostly private address space\)](#)」も参照してください。

## 大部分がプライベートなアドレス・スペース (MPAS) (mostly private address space (MPAS))

柔軟な仮想アドレス・スペース・モデル。より大きなアドレス・スペース・ブロックをプロセスに割り当てることができ、HP-UX などのシステムで使用されます。これにより、大きなデータ・スペースを必要とするプロセスのパフォーマンスが高められる場合がある。「[大部分がグローバルなアドレス・スペース \(mostly global address space\)](#)」も参照してください。

## MPAS

「[大部分がプライベートなアドレス・スペース \(mostly private address space\)](#)」を参照してください。

## MQAI

「[WebSphere MQ Administration Interface](#)」を参照。

## MQI

[メッセージ・キュー・インターフェース \(Message Queue Interface\)](#) を参照してください。

## MQI チャンネル (MQI channel)

WebSphere MQ クライアントと、サーバー・システム上のキュー・マネージャーの間の接続。MQI チャンネルは、MQI 呼び出しと応答のみを双方向で転送します。「[チャンネル \(channel\)](#)」も参照してください。

## MQM

[メッセージ・キュー管理 \(message queue management\)](#) を参照してください。

## MQSC

[WebSphere MQ スクリプト・コマンド \(WebSphere MQ script commands\)](#) を参照してください。

## MQSeries

WebSphere MQ の以前の名前。

## MQ Telemetry Transport

MQ Telemetry Transport (MQTT) は、多数の装置 (サーボ、アクチュエーター、スマートフォン、車両、家庭、医療機器、リモート・センサー、制御装置など) に接続するために TCP/IP を介してフローする、オープンで単純なパブリッシュ/サブスクライブ・プロトコルです。MQTT は、ネットワークが帯域幅によって制約される環境、または装置がメモリーやプロセッサによって制約される可能性がある環境などで動作するように設計されています。

## MQTT

[MQ Telemetry Transport](#) を参照してください。

## MQTT クライアント (MQTT client)

MQTT クライアント・アプリケーションは、WebSphere MQ Telemetry チャンネルなどの、MQTT 対応のサーバーに接続します。パブリッシュされるプロトコルを使用するために独自のクライアントを作成することも、WebSphere MQ Telemetry のインストールに付属するクライアントの 1 つを使用することもできます。標準的なクライアントは、テレメトリー・デバイスからの情報収集、およびサーバーへの情報のパブリッシュを行います。また、トピックへのサブスクライブ、メッセージの受信、およびこの情報を使用したテレメトリー・デバイスの制御も行うことができます。一部のクライアントは、WebSphere MQ Telemetry と共に提供されます。テレメトリー・クライアント (Telemetry clients) およびテレメトリー拡張クライアント (Telemetry advanced clients) も参照してください。

## MQTT サーバー (MQTT server)

MQTT サーバーは、MQTT プロトコルのサーバー側の処理を行います。通常は、多数の MQTT クライアントからの同時接続を許可し、MQTT クライアントにメッセージを配布するためのハブを提供します。遠隔測定 (MQXR) サービスを行う WebSphere MQ キュー・マネージャーは、MQTT サーバーです。

## MSCS

[Microsoft Cluster Server](#) を参照してください。

## MTS

[Microsoft Transaction Server](#) を参照してください。

## マルチ・ホップ (multi-hop)

送信元キュー・マネージャーとターゲット・キュー・マネージャー間に直接の通信リンクがない場合、ひとつまたは複数の中間キュー・マネージャーをパス・スルーすること。

## 複数インスタンス・キュー・マネージャー (multi-instance queue manager)

キュー・マネージャー・データの使用を他のキュー・マネージャー・インスタンスと共有するよう構成されたキュー・マネージャー。実行中の複数インスタンス・キュー・マネージャーの 1 つのインスタンスがアクティブとなり、他のインスタンスはスタンバイ状態となって、アクティブ・インスタンスからの引き継ぎに備えます。「[単一インスタンス・キュー・マネージャー \(single instance queue manager\)](#)」も参照してください。

# N

## 名前リスト

キュー名などのオブジェクト名のリストを含む WebSphere MQ オブジェクト。

## ネーム・サービス

UNIX および Linux システム上の WebSphere MQ、および WebSphere MQ for Windows で、指定されたキューをどのキュー・マネージャーが所有するかを決定する機能。

## ネーム・サービス・インターフェース (NSI) (name service interface (NSI))

キュー名の所有者を解決する顧客作成またはベンダー作成のプログラムが準拠する必要がある WebSphere MQ インターフェース。WebSphere MQ フレームワークの一部です。

## 名前変換 (name transformation)

UNIX および Linux システム上の WebSphere MQ、および WebSphere MQ for Windows において、キュー・マネージャー名が使用中のシステムに対して固有で有効になるように変更する内部プロセス。外部から見れば、キュー・マネージャー名は変更されません。

## ネストされたバッグ (nested bag)

WebSphere MQ 管理インターフェース (MQAI) で、別のデータ・バッグに挿入されるシステム・バッグ。

## ネスティング (nesting)

WebSphere MQ 管理インターフェース (MQAI) で、WebSphere MQ から戻された情報をグループ化する手段。

## NetBIOS (Network Basic Input/Output System)

ローカル・エリア・ネットワーク上で使用することによりメッセージ、プリント・サーバー、およびファイル・サーバー機能を提供する、ネットワークおよびパーソナル・コンピューターへの標準インターフェース。NetBIOS を使用するアプリケーション・プログラムは、LAN のデータ・リンク制御 (DLC) プロトコルの詳細を処理する必要がない。

## Network Basic Input/Output System

[NetBIOS](#) を参照してください。

## New Technology File System (NTFS)

Windows オペレーティング環境におけるネイティブ・ファイル・システムの1つ。

### ノード

Microsoft Cluster Server (MSCS) における、クラスター内の各コンピューター。

### 非永続サブスクリプション (nondurable subscription)

サブスクライブしているアプリケーションのキュー・マネージャーへの接続がオープンしている間のみ存在するサブスクリプション。サブスクライブ・アプリケーションが、意図的に、あるいは接続の損失により、キュー・マネージャーから切断されると、サブスクリプションは除去されます。[永続サブスクリプション \(durable subscription\)](#) も参照してください。

### 非持続メッセージ (nonpersistent message)

キュー・マネージャーを再始動すると消えるメッセージ。[持続メッセージ \(persistent message\)](#) も参照してください。

### NSI

[名前サービス・インターフェース \(name service interface\)](#) を参照してください。

### NTFS

[New Technology File System](#) を参照してください。

### NUL

[ヌル文字 \(null character\)](#) を参照してください。

### ヌル文字 (null character (NUL))

表示または印刷される文字のない状態を表す、X'00' の値を持つ制御文字。

## O

### OAM

[オブジェクト権限マネージャー \(object authority manager\)](#) を参照してください。

### object

1. WebSphere MQ における、キュー・マネージャー、キュー、プロセス定義、チャンネル、名前リスト、認証情報オブジェクト、管理トピック・オブジェクト、リスナー、サービス・オブジェクト、または、(z/OS のみ) CF 構造オブジェクトまたはストレージ・クラス。
2. オブジェクト指向設計またはプログラミングで、クラスを具体的に実現したもの (インスタンス) であり、データとそのデータに関連付けられた操作で構成される。オブジェクトには、クラスによって定義されたインスタンス・データが含まれますが、そのデータに関連付けられた操作はクラスが所有します。

### オブジェクト権限マネージャー (OAM) (object authority manager (OAM))

UNIX および Linux システム上の WebSphere MQ、WebSphere MQ for IBM i、および WebSphere MQ for Windows では、コマンドおよびオブジェクト管理のためのデフォルトの許可サービス。OAM は、顧客提供のセキュリティー・サービスに置き換えることも、組み合わせて実行することもできます。

### オブジェクト記述子 (object descriptor)

特定の WebSphere MQ オブジェクトを識別するデータ構造。オブジェクト記述子には、オブジェクトの名前およびオブジェクト・タイプが含まれます。

### オブジェクト・ハンドル (object handle)

プログラムが作業対象の WebSphere MQ オブジェクトにアクセスするために使う識別子またはトークン。

### オブジェクト指向プログラミング (object-oriented programming)

データの抽象化および継承という概念をベースとするプログラミング・アプローチ。プロシージャー型プログラミング技法と違い、オブジェクト指向プログラミングは、何をどのように達成するかではなく、どのデータ・オブジェクトに問題があり、どのように扱うべきかということに重点を置きます。

### OCSP

オンライン証明書状況プロトコル (Online Certificate Status Protocol)。証明書が取り消されているかどうかを確認する方法。

### オフロード (offloading)

WebSphere MQ for z/OS で、キュー・マネージャーのアクティブ・ログが保存ログに移動される自動的なプロセス。

### 単方向認証 (one way authentication)

この認証方法では、キュー・マネージャーが証明書をクライアントに提示しますが、クライアントからキュー・マネージャーへの認証の確認が行われません。

### オープン (open)

キューまたはトピックなどのオブジェクトへのアクセスを確立すること。

### オープン・システム間相互通信 (OSI) (open systems interconnection (OSI))

国際標準化機構 (ISO) の情報の交換に関する規格に従った、オープン・システムの相互接続。

### オープン・トランザクション・マネージャー・アクセス (OTMA) (Open Transaction Manager Access (OTMA))

MVS シスプレックス環境で、トランザクションに基づくコネクションレスのクライアント/サーバー・プロトコルを実装する IMS コンポーネント。このプロトコルのドメインは、z/OS システム間カップリング・ファシリティ (XCF) のドメインに制限されている。OTMA はクライアントをサーバーに接続し、クライアントが高いパフォーマンスを維持しながら大規模なネットワーク (または多数のセッション) をサポートできるようにします。

### OPM

オリジナル・プログラム・モデル (original program model) を参照してください。

### オリジナル・プログラム・モデル (OPM) (original program model (OPM))

Integrated Language Environment (ILE) モデルが導入される以前の、ソース・コードをコンパイルして高水準言語プログラムを作成するための機能セット。

### OSGi 協力会社 (OSGi Alliance)

音声、データ、およびマルチメディアの無線および有線ネットワークの管理用オープン・スタンダードの概要を定める仕様書を作成する、IBM を含む 20 社を超える会社のコンソーシアム。

### OSI

オープン・システム間相互通信 (open systems interconnection) を参照してください。

### OSI ディレクトリー標準 (OSI directory standard)

情報モデル、名前空間、機能モデル、認証フレームワークを含む包括的なディレクトリー・サービスを定義する、X.500 として知られる規格。また、X.500 はクライアントがディレクトリーにアクセスするために使用するディレクトリー・アクセス・プロトコル (DAP) も定義します。Lightweight Directory Access Protocol (LDAP) は、ディレクトリー・クライアントから X.500 アクセスの負荷をいくらか取り除いて、より広範なマシンおよびアプリケーションがディレクトリーを使用できるようにします。

### OTMA

オープン・トランザクション・マネージャー・アクセス (Open Transaction Manager Access) を参照してください。

### アウトバウンド・チャネル (outbound channel)

伝送キューからメッセージを取り出して別のキュー・マネージャーへ送信するチャネル。

### 出力ログ・バッファー (output log-buffer)

WebSphere MQ for z/OS で、アーカイブ・ログに書き込まれる前のリカバリー・ログ・レコードを保持するバッファー。

### 出力パラメーター (output parameter)

MQI 呼び出しのパラメーター。呼び出しが完了または失敗した場合にキュー・マネージャーがこのパラメーターに情報を戻します。

### オーバーローディング (overloading)

オブジェクト指向プログラミングにおいて、演算子やメソッドがコンテキストによって異なる意味を持つようにする機能。例えば、C++ では、関数やほとんどの標準的な演算子がクラス・タイプと一緒に使用されたときに、ユーザーはその関数と演算子を再定義できる。メソッドの名前あるいは演算子が同じままでも、メソッドのパラメーターがタイプ、数値、あるいはその両方で異なる。この違いをまとめて関数のシグニチャーあるいは演算子のシグニチャーと呼び、それぞれのシグニチャーには別々の実装が必要である。

## P

### ページ・セット

WebSphere MQ for z/OS がメイン・ストレージ内のバッファから永続的なバックアップ・ストレージ (DASD) ヘデータ (キューやメッセージなど) を移動する時に使う VSAM データ・セット。

### 親クラス (parent class)

他のクラスがインスタンス・メソッド、属性、およびインスタンス変数を継承する元のクラス。「[抽象クラス \(abstract class\)](#)」も参照してください。

### 部分リポジトリ (partial repository)

クラスター内のキュー・マネージャーについての情報の部分的なセット。部分リポジトリは、完全リポジトリを持たないすべてのクラスター・キュー・マネージャーによって維持されます。「[完全リポジトリ \(full repository\)](#)」も参照してください。

### パートナー・キュー・マネージャー (partner queue manager)

「[リモート・キュー・マネージャー \(remote queue manager\)](#)」を参照してください。

### PasTicket

RACF セキュア・サインオンにおいて、ワークステーションまたは他のクライアントが、RACF パスワードをネットワークを介して送信せずにホストにサインオンするために使用できる、動的に生成されるランダムで一度限りのパスワード置換。

### PCF

[プログラム式コマンド形式 \(programmable command format\)](#) を参照してください。

### 保留イベント (pending event)

CICS アダプターからの接続要求の結果として起きる予定外のイベント。

### 回復機能委任 (percolation)

エラー回復において、事前に定めた経路に沿って、あるリカバリー・ルーチンから、さらに高いレベルのリカバリー・ルーチンに進むこと。

### パフォーマンス・イベント (performance event)

限界条件が起きたことを示すイベントのカテゴリ。

### パフォーマンス・トレース (performance trace)

トレース・データがパフォーマンス分析およびチューニングに使用される、WebSphere MQ トレース・オプション。

### 永続動的キュー (permanent dynamic queue)

削除が明示的に要求された場合に限り、クローズ時に削除される動的なキュー。永続動的キューはキュー・マネージャーが失敗したときに回復されるので、持続メッセージを入れることができます。[一時動的キュー \(temporary dynamic queue\)](#) も参照してください。

### 持続メッセージ (persistent message)

キュー・マネージャーが再始動されても消えないメッセージ。[非持続メッセージ \(nonpersistent message\)](#) も参照してください。

### 個人証明書 (personal certificate)

ユーザーが所有する対応する秘密鍵の証明書。キュー・マネージャーまたはアプリケーションに関連付けられます。

### PGM

[Pragmatic General Multicast \(PGM\)](#) を参照してください。

### PID

[プロセス ID \(process ID\)](#) を参照してください。

### ping

応答を受信することを期待して、Internet Control Message Protocol (ICMP) エコー要求パケットをゲートウェイ、ルーター、またはホストに送信するコマンド。

### PKCS

Public Key Cryptography Standard。暗号化に関する標準の集合。以下の種類があります。

- 7 はメッセージ用です。
- 11 はハードウェア・セキュリティー・モジュール用です。

- 12 は鍵りポジトリーで使用されるファイル・フォーマット用です。

## PKI

[公開鍵インフラストラクチャー \(public key infrastructure\)](#) を参照してください。

## プレーン・テキスト (plain text)

[「平文 \(cleartext\)」](#) を参照してください。

## リカバリー・ポイント (point of recovery)

WebSphere MQ for z/OS で、WebSphere MQ for z/OS ページ・セットのバックアップ・コピーのセット、およびこれらのページ・セットを回復するために必要な関連ログ・データ・セット。これらのバックアップ・コピーは、ページ・セットが失われた (ページ・セットの I/O エラーなど) 場合に使用できる再始動点を提供します。

## 有害メッセージ (poison message)

キューにある正しくないフォーマットのメッセージで、受信したアプリケーションが処理できないもの。メッセージが入力キューに繰り返し配信されて、アプリケーションによって繰り返しバックアウトされる場合があります。

## ポリモアフィズム (polymorphism)

メソッドを、それが実装されたクラスに応じてさまざまな方法で実行できる、オブジェクト指向プログラミングの特性。ポリモアフィズムを使用すると、サブクラスは、親クラスのメソッドに影響を与えずに、継承されたメソッドを指定変更できます。また、ポリモアフィズムを使用すれば、クライアントは単一インターフェースからオブジェクトの複数の実装にアクセスできます。

## Pragmatic General Multicast (PGM)

信頼性の高いマルチキャスト・トランスポート・プロトコルで、複数の受信者に信頼性の高いパケットのシーケンスを同時に提供します。

## 優先シャットダウン (preemptive shutdown)

WebSphere MQ で、アプリケーションの切断を待たない、または現在の MQI 呼び出しの完了を待たないキュー・マネージャーのシャットダウン。「[即時シャットダウン \(immediate shutdown\)](#)」、「[静止シャットダウン \(quiesced shutdown\)](#)」も参照。

## 優先コンピューター (preferred computer)

Microsoft Cluster Server の制御下で実行されるアプリケーションが使用する 1 次コンピューター。別のコンピューターへのフェイルオーバー後、MCS は優先コンピューターが修復されるまでモニターし、優先コンピューターが正常に再稼働すると、すぐにアプリケーションを優先コンピューターに戻します。

## プリンシパル

別のエンティティーと安全に通信できるエンティティー。プリンシパルはそれに関連したセキュリティ・コンテキストで識別され、セキュリティ・コンテキストはそのアクセス権限を定義する。

## プライベート定義オブジェクト (privately defined object)

[ローカル定義オブジェクト \(locally defined object\)](#) を参照してください。

## プライベート・メソッドおよびインスタンス・データ (private methods and instance data)

オブジェクト指向プログラミングで、同じクラスの実装のみがアクセス可能なメソッドおよびインスタンス・データ。

## プロセス定義オブジェクト (process definition object)

WebSphere MQ アプリケーションの定義を含む WebSphere MQ オブジェクト。例えば、キュー・マネージャーはトリガー・メッセージを用いた作業を行うときにこの定義を使用します。

## プロセス ID (PID) (process ID (PID))

プロセスを表す固有の ID。プロセス ID は正整数であって、そのプロセスの存続期間が終了するまで再使用されません。

## プロデューサー (producer)

メッセージを作成し、送信するアプリケーション。「[パブリッシャー \(publisher\)](#)」、「[メッセージ・プロデューサー \(message producer\)](#)」も参照してください。

## プログラム式コマンド形式 (PCF) (programmable command format (PCF))

WebSphere MQ メッセージの一種。このメッセージが使用されるのは、ユーザー管理アプリケーションが指定のキュー・マネージャーのシステム・コマンド入力キューに PCF コマンドを書き込む場合や、ユーザー管理アプリケーションが指定のキュー・マネージャーから PCF コマンドの結果を取得する場合

合、キュー・マネージャーがイベントが発生したことの通知として使用する場合があります。 [WebSphere MQ スクリプト・コマンド \(MQSC\) \(WebSphere MQ script commands\)](#) も参照してください。

### **プログラム一時修正 (PTF) (program temporary fix (PTF))**

System i 製品、System p 製品、および System z 製品において、個別または複数の修正を含むパッケージ。ライセンスを持つすべてのお客様に提供される。PTF は、問題を解決して、場合によっては機能拡張を提供することもある。

### **property**

オブジェクトの特性で、そのオブジェクトについて記述するもの。プロパティは変更または修正できます。プロパティは、オブジェクト名前、タイプ、値、振る舞いなどの事項を記述できる。

### **保護されたメソッドおよびインスタンス・データ (protected methods and instance data)**

オブジェクト指向プログラミングで、同じクラスか派生したクラスの実装、またはフレンド・クラスからのみアクセス可能なメソッドおよびインスタンス・データ。

### **プロキシ・サブスクリプション (proxy subscription)**

プロキシ・サブスクリプションは、あるキュー・マネージャーによってパブリッシュされるトピックに関する、別のキュー・マネージャーによって行われるサブスクリプションです。プロキシ・サブスクリプションは、サブスクリプションによってサブスクライブされている各トピック・ストリングのキュー・マネージャーの間で流れます。ユーザーはプロキシ・サブスクリプションを明示的に作成しません。キュー・マネージャーが代わりにそれを行います。

### **PTF**

[プログラム一時修正 \(program temporary fix\)](#) を参照してください。

### **公開鍵 (public key)**

すべてのユーザーに公開される鍵。通常、この鍵は、公開鍵の所有者を指定するデジタル証明書に組み込まれます。

### **公開鍵暗号化方式 (public key cryptography)**

すべてのユーザーに公開される公開鍵と、メッセージの受信者のみに公開される秘密鍵の2つのキーを使用する暗号化システム。公開鍵と秘密鍵は、ある鍵で暗号化されたものは対応する秘密鍵でしか復号できない、という関係にあります。

### **公開鍵インフラストラクチャー (PKI) (public key infrastructure (PKI))**

ネットワーク・トランザクションに關与する各関係者の妥当性を検証および認証する、デジタル証明書、認証局、およびその他の登録局からなるシステム。

### **共用メソッドおよびインスタンス・データ (public methods and instance data)**

オブジェクト指向プログラミングで、すべてのクラスがアクセス可能なメソッドおよびインスタンス・データ。

### **publish**

パブリッシュ/サブスクライブ・システムにおいて、特定のトピックに関する情報をキュー・マネージャーに対し提供すること。

### **パブリッシャー (publisher)**

パブリッシュ/サブスクライブ・システム内のブローカーに対して、指定されたトピックに関する情報を使用可能にするアプリケーション。

### **パブリッシュ/サブスクライブ**

パブリッシュ・アプリケーションによって提供される情報が、その種の情報に関心があることを表明しているすべてのサブスクライブ側アプリケーションに、インフラストラクチャーによって配信されるメッセージング対話のタイプの1つ。

### **パブリッシュ/サブスクライブ・クラスター (publish/subscribe cluster)**

完全に相互接続されたキュー・マネージャーのセットで、パブリッシュ/サブスクライブ・アプリケーションのマルチ・キュー・マネージャー・ネットワークの一部を形成するもの。

### **put**

メッセージ・キューイングにおいて、MQPUT または MQPUT1 呼び出しを使用してキューにメッセージを置くこと。「[ブラウズ \(browse\)](#)」、「[読み取り \(get\)](#)」も参照してください。

## Q

### キュー

メッセージ・キューイング・アプリケーションの場合に、メッセージを保持するオブジェクト。キューは、キュー・マネージャーによって所有および管理される。

### キュー・インデックス (queue index)

WebSphere MQ for z/OS において、キューの MQGET 操作の速度を増すのに使用できるメッセージ ID のリスト、または関連 ID のリスト。

### キュー・マネージャー

アプリケーションにキューイング・サービスを提供するメッセージ・キューイング・システムのコンポーネント。

### キュー・マネージャー・イベント (queue manager event)

キュー・マネージャーが使用するリソースに関連してエラー状態が発生したことを示すイベント。例えばキューが利用できないか、またはキュー・マネージャーで重大な変更が行われた。例えば、キュー・マネージャーが停止または開始したなど。

### キュー・マネージャー・グループ (queue manager group)

クライアント・チャンネル定義テーブル (CCDT) において、サーバーへの接続が確立されたときにクライアントが接続を試行する、キュー・マネージャーのグループ。

### キュー・マネージャー・レベル・セキュリティー

WebSphere MQ for z/OS で、キュー・マネージャーに固有の RACF プロファイルを使って実施される許可検査。

### キュー・マネージャー・セット (queue manager set)

WebSphere MQ エクスプローラーのキュー・マネージャーのグループ化。これにより、ユーザーは、そのグループ内のキュー・マネージャーすべてに対してアクションを実行できます。

### キュー共有グループ

WebSphere MQ for z/OS で、共有リポジトリに保管されているオブジェクト定義の単一セット、およびカップリング・ファシリティに保管されている共有キューの単一セットにアクセスすることができ、同じシスプレックス内のキュー・マネージャー・グループ。「[共用キュー \(shared queue\)](#)」も参照。

### キュー共有グループ・レベル・セキュリティー (queue-sharing group level security)

WebSphere MQ for z/OS で、キュー共有グループ内のすべてのキュー・マネージャーが共有する RACF プロファイルを使って実施される許可検査。

### 静止

アクティブな操作が正常に完了した後で、プロセスを終了またはシステムをシャットダウンすること。

### 静的シャットダウン (quiesced shutdown)

1. CICS アダプターのシャットダウンのタイプの 1 つ。現在アクティブなタスクすべてが完了してからアダプターが WebSphere MQ から切断されます。「[強制シャットダウン \(forced shutdown\)](#)」も参照。
2. WebSphere MQ で、接続しているすべてのアプリケーションが切断できるようにするキュー・マネージャーのシャットダウン。即時シャットダウン (immediate shutdown)、優先シャットダウン (preemptive shutdown) も参照してください。

### 静止

WebSphere MQ で、停止される前のキュー・マネージャーの状態。この状態では、プログラムは処理を終了することはできますが、新しくプログラムを開始することはできません。

### クォーラム・ディスク (quorum disk)

クラスターのリカバリー・ログを保管し、サーバーが稼働中かどうかを判別するために、Microsoft Cluster Server が排他的にアクセスするディスク。一時点で 1 つのサーバーだけがクォーラム・ディスクを所有することができます。クラスター内のサーバーは、所有権を折衝することができます。

## R

### RACF

[リソース・アクセス管理機能 \(RACF\)](#) を参照してください。

## RAID

新磁気ディスク制御機構 (Redundant Array of Independent Disks) を参照してください。

## RBA

相対バイト・アドレス (relative byte address) を参照してください。

## RC

「戻りコード (return code)」を参照してください。

## 先読み (read ahead)

アプリケーションによって要求される前にクライアントにメッセージが送信されるようにできるオプション。

## 理由コード (reason code)

メッセージ・キュー・インターフェース (MQI) 呼び出しの失敗または部分的な成功の理由を記述する戻りコード。

## 受信出口 (receive exit)

メッセージ・チャンネル・エージェント (MCA) が受信通信の後に制御を再獲得し、通信接続からデータ単位を受信した直後に呼び出される、チャンネル出口プログラムの一種。「送信出口 (send exit)」も参照してください。

## 受信側チャンネル (receiver channel)

メッセージ・キューイングにおいて、送信側チャンネルに応答するチャンネル。通信リンクからメッセージを読み取り、ローカル・キューに書き込みます。

## リカバリー・ログ (recovery log)

WebSphere MQ for z/OS で、メッセージ、キュー、および WebSphere MQ サブシステムを回復させるために必要な情報を含むデータ・セット。保存ログ (archive log) も参照してください。

## 回復終了マネージャー (RTM) (recovery termination manager (RTM))

強制終了された機能に関連付けられているリカバリー・ルーチンへ制御を渡すことにより、タスクのすべての正常終了と異常終了を処理するプログラム。

## 新磁気ディスク制御機構 (RAID) (Redundant Array of Independent Disks (RAID))

ホストに対しては1つ以上の論理ディスク・ドライブのイメージを示すような、2つまたはそれ以上の物理ディスク・ドライブの集合。物理装置に障害が発生した場合に、データの冗長性により他のディスク・ドライブからのデータの読み取りまたは再生成ができます。

## 参照メッセージ (reference message)

転送されるデータの一部を参照するメッセージ。参照メッセージはメッセージ出口プログラムによって処理されます。このプログラムは、メッセージにデータを添付したりメッセージからデータを切り離すことにより、データをキュー上に保管する必要なしに転送できるようにします。

## レジストリー (registry)

ユーザー、システム、およびソフトウェアのためのアクセス情報および構成情報を格納しているリポジトリー。

## レジストリー・エディター (Registry Editor)

Windows で、ユーザーがレジストリーを編集できるようにするプログラム項目。

## レジストリー・ハイブ (registry hive)

Windows システムで、レジストリーに保管されているデータの構造。

## 相対バイト・アドレス (RBA) (relative byte address (RBA))

データ・レコードまたは制御インターバルに関する、所属データ・セットまたはファイルに割り振られたストレージ・スペースの最初からのオフセット。

## 信頼できるマルチキャスト・メッセージング (RMM) (reliable multicast messaging (RMM))

高スループットと低レイテンシーのトランスポート・ファブリックで、1対多のデータ配信または多対多データ交換のために、メッセージ中心のミドルウェア・パブリッシュ/サブスクライブの方式で設計されています。RMMはIPマルチキャスト・インフラストラクチャーを使用して、スケーラブルなリソースの節約とタイムリーな情報分散を保証します。

## リモート・キュー

リモート・キュー・マネージャーに属するキュー。プログラムはリモート・キューにメッセージを書き込むことはできますが、リモート・キューからメッセージを読み取ることはできません。「ローカル・キュー (local queue)」も参照してください。

### リモート・キュー・マネージャー

あるプログラムと同じシステム上で実行されていても、そのプログラムの接続先になっていないキュー・マネージャー。「ローカル・キュー・マネージャー ([local queue manager](#))」も参照してください。

### リモート・キュー・オブジェクト (remote queue object)

ローカル・キュー・マネージャーに属する WebSphere MQ オブジェクト。このオブジェクトは、別のキュー・マネージャーが所有するキューの属性を定義します。また、キュー・マネージャーの別名および応答先キューの別名としても使用されます。

### リモート・キューイング (remote queuing)

メッセージ・キューイングにおいて、アプリケーションが、他のキュー・マネージャーに属しているキューにメッセージを書き込むことができるようにするサービスの提供。

### 応答メッセージ (reply message)

要求メッセージへの応答に使用されるメッセージのタイプ。「レポート・メッセージ ([report message](#))」、「要求メッセージ ([request message](#))」も参照してください。

### 応答先キュー

MQPUT 呼び出しを出したプログラムが応答メッセージまたはレポート・メッセージの送信先として希望するキューの名前。

### 報告メッセージ (report message)

別のメッセージに関する情報を与えるメッセージのタイプ。レポート・メッセージは、メッセージが送達された、その宛先に到着した、その期限が切れた、何らかの理由でメッセージを処理できなかったなどということを示します。「応答メッセージ ([reply message](#))」、「要求メッセージ ([request message](#))」も参照してください。

### リポジトリ (repository)

クラスターのメンバーであるキュー・マネージャーについての情報の集合。この情報には、キュー・マネージャーの名前、ロケーション、チャンネル、キュー・マネージャーがホストするキューなどが含まれます。

### リポジトリ・キュー・マネージャー (repository queue manager)

クラスターについての情報の完全リポジトリを持つキュー・マネージャー。

### 要求側チャンネル (requester channel)

メッセージ・キューイングで、サーバー・チャンネルの操作を開始するためにローカルに始動することができるチャンネル。[サーバー・チャンネル \(server channel\)](#) も参照。

### 要求メッセージ (request message)

別のプログラムからの応答を要求するために使用されるメッセージのタイプ。[応答メッセージ \(reply message\)](#)、[レポート・メッセージ \(report message\)](#) も参照してください。

### 要求/応答 (request/reply)

別のアプリケーションに応答を要求するために要求メッセージを使用する、メッセージング・アプリケーションのタイプ。「[データグラム \(datagram\)](#)」も参照してください。

## RESLEVEL

WebSphere MQ for z/OS で、API リソースのセキュリティ用にチェックされるユーザー ID の数を制御するオプション。

### 解決パス (resolution path)

アプリケーションが、MQOPEN 呼び出しに対する入力として別名またはリモート・キューを指定した場合にオープンされる一組のキュー。

## resource

ジョブ、タスク、または実行中のプログラムが必要とするコンピューター・システムまたはオペレーティング・システムの機能。リソースには、メイン・ストレージ、入出力装置、処理装置、データ・セット、ファイル、ライブラリー、フォルダー、アプリケーション・サーバー、制御プログラム、処理プログラムなどがある。

### リソース・アクセス管理機能 (RACF) (Resource Access Control Facility (RACF))

アクセス制御を提供する IBM ライセンス・プログラム。このアクセス制御は、システムに対してユーザーを識別し、システムのユーザーを認証し、保護されたリソースに対するアクセスを許可し、システムに入ろうとする無許可の試行をログに記録し、保護リソースへのアクセスをログに記録することによって行われる。

### リソース・アダプター (resource adapter)

アプリケーション・サーバーで実行している JMS アプリケーションおよびメッセージ駆動型 Bean を WebSphere MQ キュー・マネージャーのリソースにアクセスできるようにする、Java Enterprise Edition Connector Architecture のインプリメンテーション。

### リソース・マネージャー (resource manager)

メモリー・バッファーやデータ・セットなどの共有リソースへのアクセスを管理、制御するアプリケーション、プログラム、またはトランザクション。WebSphere MQ、CICS、および IMS はリソース・マネージャーです。

### リソース・リカバリー・サービス (RRS) (Resource Recovery Services (RRS))

同期点マネージャーを使用して参加するリソース・マネージャー間での変更を調整する z/OS のコンポーネント。

### 応答側 (responder)

分散キューイングにおいて、他のシステムからのネットワーク接続要求に応答するプログラム。「[イニシエーター \(initiator\)](#)」も参照してください。

### 再同期 (resynch)

WebSphere MQ で、メッセージ転送を再始動しないでチャンネルを始動し、未確定の状態のメッセージすべてを解決するように指示するオプション。

### 戻りコード (RC) (return code (RC))

プログラムの処理結果を示すためにプログラムによって返される値。完了コードおよび理由コードは戻りコードの一例である。

### 送信側へ戻す (return-to-sender)

MCA がメッセージを配達することができない場合に使用可能なオプション。MCA は、発信元にメッセージを送り返すことができます。

### Rivest-Shamir-Adleman アルゴリズム (RSA) (Rivest-Shamir-Adleman algorithm (RSA))

RSA Data Security, Inc が開発した公開鍵暗号化技術。IBM における SSL の実装に使用されます。

### RMM

[信頼できるマルチキャスト・メッセージング](#)を参照してください。

### ロールバック (rollback)

[バックアウト \(backout\)](#) を参照してください。

### ルート証明書 (root certificate)

チェーンにおける最上位の証明書。この証明書が自己署名証明書である場合、他の証明書の署名にのみ使用されます。[自己署名証明書 \(self-signed certificate\)](#)も参照。

### RRS

[リソース・リカバリー・サービス \(Resource Recovery Services\)](#) を参照してください。

### RSA

[Rivest-Shamir-Adleman アルゴリズム \(Rivest-Shamir-Adleman algorithm\)](#) を参照してください。

### RTM

[回復終了マネージャー \(recovery termination manager\)](#) を参照してください。

### 規則テーブル (rules table)

送達不能キュー・ハンドラーが送達不能キュー (DLQ) 上のメッセージに適用する 1 つ以上の規則が入った制御ファイル。

## S

### スケーラブル・パラレル 2 (SP2) ((Scalable Parallel 2 (SP2))

IBM のパラレル UNIX システム。高速ネットワーク上で効率的なパラレル AIX システム。

### SDK

[ソフトウェア開発キット \(software development kit\)](#)を参照してください。

### SDWA

[システム診断作業域 \(system diagnostic work area\)](#) を参照してください。

### SECMEC

[セキュリティ・メカニズム \(security mechanism\)](#) を参照してください。

### セキュア・ソケット・レイヤー (SSL)

通信のプライバシーを確保するセキュリティー・プロトコル。SSL を使用すれば、盗聴、改ざん、およびメッセージ偽造を防止するよう設計された方法で、クライアント/サーバー・アプリケーションは通信することができる。「[認証局 \(certificate authority\)](#)」も参照。

### セキュリティー・イネーブリング・インターフェース (SEI) (security enabling interface (SEI))

許可の検査、ユーザー識別の提供、または許可を実施する顧客作成またはベンダー作成のプログラムが準拠する必要のある WebSphere MQ インターフェース。WebSphere MQ フレームワークの一部です。

### セキュリティー出口

チャネルの始動時に、最初のデータ・ネゴシエーションが完了した後すぐに呼び出されるチャネル出口プログラム。セキュリティー出口は、通常はペアで動作し、メッセージ・チャネルと MQI チャネルの両方で呼び出すことができます。セキュリティー出口の主な目的は、チャネルの両端にあるメッセージ・チャネル・エージェント (MCA) が、互いを認証できるようにすることです。

### セキュリティー ID (SID) (security identifier (SID))

Windows システムで、ユーザー ID を補完して、ユーザーが定義されている Windows セキュリティー・アカウント・マネージャー・データベース上の完全なユーザー・アカウント詳細を識別します。

### セキュリティー・メカニズム (SECMEC) (security mechanism (SECMEC))

セキュリティー・サービスを実現するために使われるテクニカル・ツールまたはテクニック。メカニズムは、特定のサービスを提供するために、単独で実行されることも、他のメカニズムと一緒に実行されることもあります。セキュリティー・メカニズムの例としては、アクセス制御リスト、暗号化、およびデジタル署名があります。

### セキュリティー・メッセージ (security message)

チャネルの両端で呼び出されるセキュリティー出口が、相互に通信するために送信するメッセージの 1 つ。セキュリティー・メッセージのフォーマットは定義されていないため、ユーザーが決定します。

### セキュリティー・サービス (security service)

コンピューター・システムのリソースを保護するためのサービス。セキュリティー・サービスの例として、アクセス制御などがあります。

### セキュリティー・サポート・プロバイダー・インターフェース (SSI) (Security Support Provider Interface (SSI))

ネットワーク・アプリケーションが認証済み接続を確立し、この接続を介してデータを安全に交換するために、幾つかのセキュリティー・サポート・プロバイダー (SSP) の 1 つを呼び出す方法。Windows システム上で使用可能です。

### 自己署名証明書 (self-signed certificate)

証明書のデジタル署名は、その証明書の公開鍵に対応した秘密鍵を使用して生成されます。

### segmentation

キュー・マネージャー、キュー、あるいは、アプリケーションで扱うには大きすぎるメッセージを、複数の小さい物理メッセージに分割すること。これは、受信側のキュー・マネージャーあるいはアプリケーションで再組み立てされます。

### SEI

[セキュリティー・イネーブリング・インターフェース \(security enabling interface\)](#) を参照してください。

### セレクター

データ項目の ID。WebSphere MQ Administration Interface (MQAI) では、ユーザー・セレクターおよびシステム・セレクターの 2 つのタイプのセレクターがある。

### セマフォ (semaphore)

UNIX および Linux システムにおいて、シグナルの機能を拡張する、2 つのプロセス間の一般的な通信方式。

### 送信側チャネル (sender channel)

メッセージ・キューイングで、転送を開始し、伝送キューからメッセージを除去したり、通信リンクを介してメッセージを受信側または要求側チャネルに移動したりするチャネル。

### 送信出口 (send exit)

メッセージ・チャネル・エージェント (MCA) が送信通信を発行してデータ単位を通信接続経由で送信する直前に呼び出される、チャネル出口プログラム的一种。「[受信出口 \(receive exit\)](#)」も参照してください。

## Sequenced Packet Exchange (SPX) プロトコル (Sequenced Packet Exchange protocol (SPX))

ネットワーク上の2つのノード間でコネクション・オリエンテッドのサービスを提供し、主にクライアント/サーバー・アプリケーションによって使用される、セッション指向のネットワーク・プロトコル。このプロトコルは Internet Packet Exchange (IPX) プロトコルを利用しているため、フロー制御やエラー・リカバリーが提供されるとともに、物理ネットワークの信頼性が保証されます。

## シーケンス番号折り返し値 (sequence number wrap value)

WebSphere MQ で、通信リンクの両端が現在のメッセージ・シーケンス番号を同時にリセットすることを保証するメソッド。シーケンス番号を付けてメッセージを送信することにより、受信チャネルは、メッセージを保管する際にメッセージの順序を確実に再確立することができます。

## シリアルライゼーション (serialization)

オブジェクト指向プログラミングで、プログラム・メモリーから通信メディアに順番にデータを書き込むこと。

## サーバー

1. リモート・ワークステーションで稼働するクライアント・アプリケーションにキュー・サービスを提供するキュー・マネージャー。
2. 他のソフトウェア・プログラムまたは他のコンピューターにサービスを提供する、ソフトウェア・プログラムまたはコンピューター。「[クライアント \(client\)](#)」も参照。

## サーバー・チャネル (server channel)

メッセージ・キューイングで、要求側チャネルに回答し、伝送キューからメッセージを除去したり、通信リンクを介してメッセージを要求側チャネルに移動したりするチャネル。「[要求側チャネル \(requester channel\)](#)」も参照してください。

## サーバー接続チャネル・タイプ (server-connection channel type)

キュー・マネージャーを実行するサーバーに関連付けられた MQI チャネル定義のタイプ。「[クライアント接続チャネル・タイプ \(client-connection channel type\)](#)」も参照してください。

## サービス間隔 (service interval)

サービス・インターバル・イベントの条件が満たされたかどうかを判別するときに、キュー・マネージャーが書き込みまたは読み取りと、それに続く読み取りとの間の経過時間を比較する対象となる時間間隔。キューのサービス・インターバルはキュー属性によって指定されます。

## サービス・インターバル・イベント (service interval event)

サービス・インターバルに関連したイベント。

## サービス・オブジェクト (service object)

キュー・マネージャーの開始時に追加のプロセスを開始し、キュー・マネージャーの停止時にそのプロセスを停止することができるオブジェクト。

## セッション (session)

ネットワーク上にある2つの端末、ソフトウェア・プログラム、またはデバイスがセッション中に通信してデータを交換できるようにする、これら2つのエレメント間の論理または仮想接続。

## セッション ID (session ID)

WebSphere MQ for z/OS で、伝送キューからリンクへメッセージを移動する際にメッセージ・チャネル・エージェントが使用する通信リンクを定義する CICS 固有の ID。

## セッション・レベル認証 (session-level authentication)

システム・ネットワーク体系 (Systems Network Architecture (SNA)) で、2つの論理単位 (LU) がセッションの活動中にお互いを認証できるようにする、セッション・レベルのセキュリティー・プロトコル。セッション・レベル認証は、LU-LU 検査とも呼ばれます。

## セッション・レベル暗号化 (session-level cryptography)

システム・ネットワーク体系 (Systems Network Architecture (SNA)) で、2つの論理単位 (LU) の間のセッション上を流れるデータを暗号化および復号する方法。

## 共有インバウンド・チャネル (shared inbound channel)

WebSphere MQ for z/OS で、リスナーがグループ・ポートを使って開始したチャネル。共有チャネルのチャネル定義は、ゼロのページ・セット (プライベート) または共有リポジトリ (グローバル) のどちらにも保管することができます。

### 共有アウトバウンド・チャンネル (shared outbound channel)

WebSphere MQ for z/OS で、共有伝送キューからメッセージを移動するチャンネル。共有チャンネルのチャンネル定義は、ゼロのページ・セット (プライベート) または共有リポジトリ (グローバル) のどちらにも保管することができます。

### 共有キュー (shared queue)

WebSphere MQ for z/OS のローカル・キューのタイプ。キュー上のメッセージはカップリング・ファシリティに保管され、キュー共有グループ内の 1 つ以上のキュー・マネージャーからアクセスすることができます。キューの定義は、共用リポジトリに保管されています。「[キュー共有グループ \(queue-sharing group\)](#)」も参照してください。

### 共有リポジトリ (shared repository)

WebSphere MQ for z/OS で、グローバルに定義されたオブジェクト定義を保持するために使われる、共有 Db2 データベース。

### 共用、会話の (sharing conversations)

チャンネル・インスタンスを共有する複数の会話用の機能、あるいは、チャンネル・インスタンスを共有する会話そのもの。

### シェル (shell)

ユーザーとオペレーティング・システム間のソフトウェア・インターフェース。シェルは一般に次の 2 つのカテゴリに分けられる。オペレーティング・システムへのコマンド・ラインインターフェースを提供するコマンド・ラインシェル、およびグラフィカル・ユーザー・インターフェース (GUI) を提供するグラフィカル・シェル。

### SID

「[セキュリティ ID \(security identifier\)](#)」を参照してください。

### 信号 (signal)

プロセスがシステム内で発生するイベントを通知される、またはそのイベントによって影響を受けるメカニズム。そのようなイベントの例には、ハードウェア例外やプロセスによる特定のアクションがある。

### 信号機能 (signaling)

WebSphere MQ for z/OS および WebSphere MQ for Windows で、予期されたメッセージがキューに届いた時に、オペレーティング・システムがプログラムに通知することができる機能。

### 署名 (signature)

メソッドと関連付けられたタイプの集合。署名には、任意で、戻り値のタイプと共に、各メソッドの引数の数、順序、およびタイプが含まれる。

### 署名者証明書 (signer certificate)

暗号化または署名に使用される証明書。

### 単一インスタンス・キュー・マネージャー (single instance queue manager)

複数インスタンスを持たないキュー・マネージャー。[複数インスタンス・キュー・マネージャー](#) も参照してください。

### 単一ロギング (single logging)

WebSphere MQ for z/OS アクティビティの記録方法で、各変更がひとつのデータ・セットにのみ記録されるもの。[重複ロギング \(dual logging\)](#) も参照してください。

### 単一フェーズ・バックアウト (single-phase backout)

進行中のアクションを終了させず、そのアクション中のすべての変更を取り消すメソッド。

### 単一フェーズ・コミット (single-phase commit)

プログラムが、別のリソース・マネージャーに制御されているリソースに対して行った更新と調整せずに、コミットメント・リソースに対する更新をコミットする方式。

### SIT

[システム初期化テーブル \(system initialization table\)](#) を参照してください。

### SMF

[システム管理機能 \(System Management Facilities\)](#) を参照してください。

### SNA

[システム・ネットワーク体系 \(Systems Network Architecture\)](#) を参照してください。

## ソフトウェア開発キット (SDK) (software development kit (SDK))

特定のコンピューター言語で、または特定のオペレーティング環境用にソフトウェアを開発するためのツール、API、およびドキュメンテーションのセット。

## ソース・キュー・マネージャー (source queue manager)

[ローカル・キュー・マネージャー \(local queue manager\)](#) を参照してください。

## SP2

[スケーラブル・パラレル 2 \(Scalable Parallel 2\)](#) を参照してください。

## SPX

[Sequenced Packet Exchange プロトコル \(Sequenced Packet Exchange protocol\)](#) を参照してください。

## SSI

[セキュリティ・サポート・プロバイダー・インターフェース \(Security Support Provider Interface\)](#) を参照してください。

## SSL

[「Secure Sockets Layer」](#) を参照。

## SSLPeer

発行者の値は、リモート個人証明書の識別名を表します。

## SSL または TLS クライアント (SSL or TLS client)

接続の開始側。キュー・マネージャーからのアウトバウンド・チャンネルの1つが、SSL または TLS クライアントにもなります。

## スタンバイ・キュー・マネージャー・インスタンス (standby queue manager instance)

実行中の複数インスタンス・キュー・マネージャーのインスタンスで、アクティブなインスタンスから引き継ぐ準備をしているもの。複数インスタンス・キュー・マネージャーのスタンバイ・インスタンスは1つ以上あります。

## スタンザ

ファイル内のグループになった行。合わせて共通の機能を持つか、システムの一部を定義する。スタンザは通常ブランク行またはコロンの分離され、各スタンザは名前を持つ。

## スター接続通信ネットワーク (star-connected communications network)

すべてのノードが中央ノードに接続するネットワーク。

## ストレージ・クラス

WebSphere MQ for z/OS で、特定のキュー用のメッセージを保管するページ・セット。キューが定義される時に、ストレージ・クラスが指定されます。

## 蓄積交換 (store and forward)

パケット、メッセージ、またはフレームが、その宛先に再送される前に、データ・ネットワーク内に一時的に保管されること。

## ストリーミング (streaming)

オブジェクト指向プログラミングにおける、クラス情報およびオブジェクト・データのシリアライゼーション。

## サブスクライブ

トピックに関する情報を要求すること。

## サブシステム (subsystem)

z/OS で、1つまたは複数の機能を実行するが、要求されるまでは何も行わないサービス・プロバイダー。例えば、WebSphere MQ for z/OS の各キュー・マネージャー、あるいは Db2 for z/OS データベース管理システムのインスタンスは、z/OS サブシステムです。

## スーパーバイザー呼び出し (SVC) (supervisor call (SVC))

実行中のプログラムを中断して制御をスーパーバイザーに渡し、特定のサービスを実行する命令。

## SVC

[スーパーバイザー呼び出し \(supervisor call\)](#) を参照してください。

## 切り替え (switchover)

アクティブなマルチ・インスタンス・キュー・マネージャーのインスタンスからスタンバイ・インスタンスへ変更すること。切り替えは、オペレーターにより意図的にアクティブなマルチ・インスタンスのキュー・マネージャー・インスタンスを停止することで起きます。

### スイッチ・プロファイル (switch profile)

WebSphere MQ for z/OS で、WebSphere MQ の始動時またはリフレッシュ・セキュリティー・コマンドが出された場合に使用される RACF プロファイル。WebSphere MQ が検出したスイッチ・プロファイルは、指定されたリソースの検査をオフにします。

### 対称鍵暗号化方式 (symmetric key cryptography)

メッセージの送信側と受信側が、単一の共通な秘密鍵を共有し、それを使用してメッセージを暗号化および復号する暗号化システム。このシステムでは、認証が提供されません。「[非対称鍵暗号化方式 \(asymmetric key cryptography\)](#)」も参照してください。

### 症状ストリング (symptom string)

IBM ソフトウェア・サポート・データベースの検索用に設計された、構造化形式で表示される診断情報。

### 同期メッセージング (synchronous messaging)

プログラム間の通信方法。プログラムがメッセージをメッセージ・キューに置き、独自の処理を続ける前に、そのメッセージに対する応答を待ちます。「[非同期メッセージング \(asynchronous messaging\)](#)」も参照してください。

### 同期点 (sync point)

保護リソースが整合しているトランザクションの処理時のポイント。

### シスプレックス (sysplex)

マルチシステム対応の特定のハードウェア・コンポーネントおよびソフトウェア・サービスを通じて相互に通信する一連の z/OS システム。

### システム・バッグ (system bag)

MQAI によって作成されるデータ・バッグの一種。

### システム制御コマンド (system control commands)

バッファ・プール、ストレージ・クラス、ページ・セットなど、プラットフォームに特定のエンティティの操作に使用されるコマンド。

### システム診断作業域 (SDWA) (system diagnostic work area (SDWA))

z/OS 環境で、SYS1.LOGREC エントリに記録される、プログラムまたはハードウェアのエラーについて記述したデータ。

### システム初期化テーブル (SIT) (system initialization table (SIT))

CICS が始動時に使うパラメーターを含む表。

### システム項目 (system item)

MQAI によって作成されるデータ項目の一種。

### システム管理機能 (SMF) (System Management Facilities (SMF))

システムおよびジョブに関連するさまざまな情報を収集および記録する z/OS のコンポーネント。

### システム・セレクター (system selector)

WebSphere MQ 管理インターフェース (MQAI) で、作成時にデータ・バッグに組み込まれるシステム項目 ID。

### システム・ネットワーク体系 (SNA) (Systems Network Architecture (SNA))

ネットワークの構成と運用を制御することによって情報を送信するための論理構造、形式、プロトコル、および操作シーケンスの説明。

## T

### 改ざん (tampering)

情報が転送中に変更または置換されてから宛先に送信される、通信セキュリティーの侵害。「[盗聴 \(eavesdropping\)](#)」、「[偽名 \(impersonation\)](#)」も参照。

### ターゲット・ライブラリー高レベル修飾子 (thlqual) (target library high-level qualifier (thlqual))

z/OS ターゲット・データ・セット名の高レベルな修飾子。

### ターゲット・キュー・マネージャー (target queue manager)

「[リモート・キュー・マネージャー \(remote queue manager\)](#)」を参照してください。

## タスク制御ブロック (TCB) (task control block (TCB))

サブシステムに接続するアドレス・スペース内のタスクについての情報を通信するために使用される z/OS 制御ブロック。

## タスク・スイッチング (task switching)

幾つかのタスク間での I/O オペレーションと処理がオーバーラップすること。

## TCB

[タスク制御ブロック \(task control block\)](#) を参照してください。

## TCP

[伝送制御プロトコル \(Transmission Control Protocol\)](#) を参照してください。

## TCP/IP

[伝送制御プロトコル/インターネット・プロトコル \(Transmission Control Protocol/Internet Protocol\)](#) を参照してください。

## 技術情報 (technote)

単一トピックに関する短い文書。

## テレメトリー・チャンネル (telemetry channel)

テレメトリー・チャンネルは、WebSphere MQ 上のキュー・マネージャーと MQTT クライアントの間の通信リンクです。各チャンネルに 1 つ以上のテレメトリー・デバイスが接続される可能性があります。

## テレメトリー拡張クライアント (telemetry advance client)

拡張テレメトリー・クライアントは、メインの WebSphere MQ インストール済み環境の mqxr サブフォルダーにインストールされます。それらの占有スペースは小さく、MQTT サーバーは複数の MQTT クライアントからの接続を許可して、WebSphere MQ へのアップリンクおよびブリッジを提供します。拡張クライアントは、アップリンク接続が切れた場合に、そのクライアントの代わりに、メッセージを開始することができます。

## テレメトリー・クライアント (telemetry client)

テレメトリー・クライアントは、メインの WebSphere MQ インストール済み環境の mqxr サブフォルダーにインストールされる MQTT クライアントです。テレメトリー・クライアントは、MQTT プロトコルを使用して MQ に接続します。

## 遠隔測定 (MQXR) サービス (telemetry (MQXR) service)

MQTT プロトコルのサーバー側の処理を行う MQ サービス (MWTT サーバーを参照)。遠隔測定 (MQXR) サービスは、遠隔測定チャンネルをホストします。

## 一時動的キュー (temporary dynamic queue)

クローズ時に削除される動的キュー。一時動的キューは、キュー・マネージャーで障害が発生した場合に回復されないため、非持続メッセージしか入れることができません。「[永続動的キュー \(permanent dynamic queue\)](#)」も参照してください。

## テラスペース (teraspace)

プロセスに専用のストレージを提供する、1 テラバイトの一時ストレージ域。

## 終了通知 (termination notification)

CICS サブシステムが WebSphere MQ for z/OS に接続した時にアクティブになる保留イベント。

## thlqual

[ターゲット・ライブラリー高レベル修飾子 \(target library high-level qualifier\)](#) を参照してください。

## スレッド

プロセスを制御しているコンピューター命令のストリーム。一部のオペレーティング・システムでは、スレッドはプロセス内の操作の最小単位である。いくつかのスレッドを同時に実行して、異なるジョブを実行させることができます。

## TID

[トランザクションの識別子 \(transaction identifier\)](#) を参照してください。

## 時間非依存メッセージング (time-independent messaging)

[非同期メッセージング \(asynchronous messaging\)](#) を参照してください。

## TLS

Transport Layer Security - SSL の後継。

## TMF

[トランザクション・マネージャー機能 \(Transaction Manager Facility\)](#) を参照してください。

## TMI

[トリガー・モニター・インターフェース \(trigger monitor interface\)](#) を参照してください。

## TP

[トランザクション・プログラム \(transaction program\)](#) を参照してください。

### トレース

コンピューター・プログラムまたはトランザクションの処理の記録。トレースにより収集された情報を使用して、問題やパフォーマンスを査定できます。

### トランザクション ID

[トランザクションの識別子 \(transaction identifier\)](#) を参照してください。

### トランザクション識別子 (TID、トランザクション ID、XID) (transaction identifier (TID, transaction ID, XID))

トランザクションに割り当てられて、そのトランザクションに関連するアクションを識別するために使用される、固有の名前。

### トランザクション・マネージャー (transaction manager)

グローバル・トランザクションを管理し、トランザクションをコミットするかロールバックするかの決定を調整することにより、リソース・マネージャーのアクティビティーを調整するソフトウェア・ユニット。

### トランザクション・マネージャー機能 (Transaction Manager Facility (TMF))

IBM WebSphere MQ for HP Integrity NonStop Server で、ビジネス・トランザクションおよびデータベースの完全性を保護するサブシステム。多くの場合、NonStop Transaction Manager/MP と同じ意味で使用されます。

### トランザクション・プログラム (transaction program (TP))

SNA ネットワークでトランザクションを処理するプログラム。

### 伝送制御プロトコル (TCP) (Transmission Control Protocol (TCP))

インターネット、および Internet Engineering Task Force (IETF) のインターネットワーク・プロトコル標準に準拠するネットワークで使用される通信プロトコル。TCP は、パケット交換通信ネットワークとそのようなネットワークで相互接続されたシステムで、信頼できるホスト間プロトコルを提供する。「[インターネット・プロトコル \(Internet Protocol\)](#)」も参照。

### 伝送制御プロトコル/インターネット・プロトコル (TCP/IP) (Transmission Control Protocol/Internet Protocol (TCP/IP))

業界標準の独占されていない通信プロトコルのセットのことで、異なる種類の相互接続ネットワークにおいて、アプリケーション間の信頼性のあるエンドツーエンド接続を提供する。

### 伝送プログラム (transmission program)

[メッセージ・チャンネル・エージェント \(message channel agent\)](#) を参照してください。

### 伝送 キュー

リモート・キュー・マネージャーを宛先とする準備済みメッセージが一時的に保管されるローカル・キュー。

### トリガー・キュー (triggered queue)

トリガーが設定されていて、そのトリガー条件が満たされた場合にトリガー・メッセージを書き込む必要があるローカル・キュー。

### トリガー・イベント (trigger event)

キュー・マネージャーに、開始キュー上にトリガー・メッセージを作成させるイベント。例えば、キューに届いているメッセージなど。

### トリガー操作 (triggering)

WebSphere MQ で、キュー上で事前定義された条件が満たされた場合、キュー・マネージャーに自動的にアプリケーションを開始させる機能。

### トリガー・メッセージ

トリガー・モニターを開始するプログラムについての情報が含まれるメッセージ。

### トリガー・モニター (trigger monitor)

ひとつまたは複数の開始キューにサービスを提供する、常に実行されているアプリケーション。トリガー・メッセージが開始キューに到達すると、トリガー・モニターがそのメッセージを検索します。

トリガー・モニターはトリガー・メッセージ内の情報を使って、トリガー・イベントが発生したキューにサービスを提供する処理を開始します。

### **トリガー・モニター・インターフェース (TMI) (trigger monitor interface (TMI))**

顧客作成またはベンダー作成のトリガー・モニター・プログラムが準拠する必要がある WebSphere MQ インターフェース。WebSphere MQ フレームワークの一部です。

### **トラストストア (trust store)**

リモート・システムから証明書を検証するために、CA 証明書が配置される場所。[鍵ストア \(key store\)](#) も参照。

### **双方向認証 (two way authentication)**

この認証方法では、キュー・マネージャーとクライアントが相互に証明書を提示します。相互認証とも呼ばれます。

### **2 フェーズ・コミット (two-phase commit)**

リカバリー可能なリソースと外部サブシステムがコミットされる 2 ステップのプロセス。最初のステップでは、データベース・マネージャー・サブシステムがポーリングされ、コミットの準備ができてることが確認されます。すべてのサブシステムが肯定応答を返した場合、データベース・マネージャーはコミットするようにそれらに指示を出します。

### **タイプ**

データの内部形式を指定し、データの使用方法を決定する特性。

## **U**

### **UDP**

[ユーザー・データグラム・プロトコル \(User Datagram Protocol\)](#) を参照してください。

### **無許可アクセス (unauthorized access)**

コンピューター内のリソースに、許可無くアクセスすること。

### **未配布メッセージ・キュー (undelivered message queue)**

[送達不能キュー \(dead-letter queue\)](#) を参照してください。

### **undo/redo レコード (undo/redo record)**

リカバリーで使用されるログ・レコード。レコードの redo 部分には、WebSphere MQ オブジェクトに対して実行される変更が記述されます。undo の部分には、作業がコミットされない場合に、変更をどのようにバックアウトするかが記述されます。

### **リカバリー単位 (unit of recovery)**

例えば Db2 for z/OS のような、単独のリソース・マネージャー内のリカバリー可能なオペレーション・シーケンス。「[作業単位 \(UOW\) \(unit of work\)](#)」も参照してください。

### **作業単位 (UOW) (unit of work (UOW))**

2 つの整合点間でアプリケーションが実施した、回復可能なオペレーションのシーケンス。作業単位は、トランザクションの開始時、またはユーザーが要求した同期点で開始します。また、作業単位はユーザーが要求した同期点、またはトランザクションの終了時に終わります。

### **UOW**

「[作業単位 \(UOW\) \(unit of work\)](#)」を参照してください。

### **ユーザー・バッグ (user bag)**

MQAI において、ユーザーが作成するデータ・バッグの一種。

### **ユーザー・データグラム・プロトコル (User Datagram Protocol (UDP))**

信頼性が低いコネクションレス・データグラム・サービスを提供するインターネット・プロトコル。このプロトコルは、あるマシン上のアプリケーション・プログラムまたはプロセスが、別のマシン上のアプリケーション・プログラムまたはプロセスにデータグラムを送信できるようにします。

### **ユーザー項目 (user item)**

MQAI において、ユーザーが作成するデータ項目の一種。

### **ユーザー・セレクター (user selector)**

WebSphere MQ 管理インターフェース (MQAI) で、データ項目を識別するために、そのデータ項目とともにデータ・バッグに書き込まれる ID。WebSphere MQ では、WebSphere MQ オブジェクトに関して事前定義されたユーザー・セレクターが提供されます。

## ユーザー・トークン (user token) (UTOKEN)

ユーザーのセキュリティー特性をカプセル化または表現する RACF セキュリティー・トークン。RACF は、システムの各ユーザーに対して UTOKEN を割り当てます。

## ユーティリティー

WebSphere MQ で、システム・オペレーターまたはシステム管理者に WebSphere MQ コマンド以上の機能を提供するプログラムのセット。

## UTOKEN

[「ユーザー・トークン \(user token\)」](#) を参照してください。

## V

### 値

データ項目の内容。整数、ストリング、または別のデータ・バッグのハンドルのいずれかになります。

### 仮想メソッド (virtual method)

オブジェクト指向プログラミングで、ポリモアフィズムを呈するメソッド。

## W

### WebSphere MQ

メッセージ・キューイング・サービスを提供する IBM ライセンス・プログラムのファミリー。

### WebSphere MQ 管理インターフェース (MQAI)

データ・バッグを使用して WebSphere MQ キュー・マネージャー上で管理タスクを実行するプログラミング・インターフェース。データ・バッグを使用すると、ユーザーは、WebSphere MQ オブジェクトのプロパティー (またはパラメーター) を処理することができます。

### WebSphere MQ classes for .NET

.NET プログラミング・フレームワークを使って作成されたプログラムが、WebSphere MQ に WebSphere MQ クライアントとして接続したり、または WebSphere MQ サーバーに直接接続したりできるようにするクラスのセット。

### C++ 用の WebSphere MQ クラス (WebSphere MQ classes for C++)

WebSphere MQ Message Queue Interface (MQI) を C++ プログラミング言語でカプセル化するクラスのセット。

### WebSphere MQ classes for Java

WebSphere MQ Message Queue Interface (MQI) を Java プログラミング言語でカプセル化するクラスのセット。

### WebSphere MQ の完全に管理された .NET クライアント (WebSphere MQ fully-managed .NET client)

キュー・マネージャー全体をインストールしなくても、システムにインストール可能な WebSphere MQ 製品の一部。WebSphere MQ .NET クライアントは、完全に管理された .NET アプリケーションにより使用され、サーバー・システム上のキュー・マネージャーと通信します。完全に管理されていない .NET アプリケーションは、WebSphere MQ MQI クライアントを使用します。「[クライアント \(client\)](#)」、「[WebSphere MQ MQI クライアント \(WebSphere MQ MQI client\)](#)」、「[WebSphere MQ Java クライアント \(WebSphere MQ Java client\)](#)」も参照してください。

### WebSphere MQ Java クライアント (WebSphere MQ Java client)

キュー・マネージャー全体をインストールしなくても、システムにインストール可能な WebSphere MQ 製品の一部。WebSphere MQ Java クライアントは、Java アプリケーション (WebSphere MQ classes for Java と WebSphere MQ classes for JMS の両方) により使用され、サーバー・システム上のキュー・マネージャーと通信します。「[クライアント \(client\)](#)」、「[WebSphere MQ MQI クライアント \(WebSphere MQ MQI client\)](#)」、「[WebSphere MQ の完全に管理された .NET クライアント \(WebSphere MQ fully-managed .NET client\)](#)」も参照してください。

### WebSphere MQ MQI クライアント

キュー・マネージャー全体をインストールしなくても、システムにインストール可能な WebSphere MQ 製品の一部。WebSphere MQ MQI クライアントは、アプリケーションからの MQI 呼び出しを受け付け、サーバー・システム上のキュー・マネージャーと通信します。「[クライアント \(client\)](#)」、「[WebSphere MQ Java クライアント \(WebSphere MQ Java client\)](#)」、「[WebSphere MQ の完全に管理された .NET クライアント \(WebSphere MQ fully-managed .NET client\)](#)」も参照してください。

### WebSphere MQ スクリプト・コマンド (MQSC) (WebSphere MQ script commands (MQSC))

WebSphere MQ オブジェクトを操作するために使われる、すべてのプラットフォームで同じ形式の、人間が読むことのできるコマンド。「[プログラム式コマンド形式 \(PCF\) \(programmable command format\)](#)」も参照してください。

### WebSphere MQ サーバー

ひとつまたは複数のクライアントに対してキューイング・サービスを提供するキュー・マネージャー。キューなど、すべての WebSphere MQ オブジェクトはキュー・マネージャー・システム上にのみ、つまり MQI サーバー・マシン上にのみ存在します。サーバーは、通常のローカル MQI アプリケーションもサポートすることができます。

### WebSphere MQ Telemetry

WebSphere MQ Telemetry は、さまざまな装置プラットフォーム上で動作するスマート・デバイスに組み込める小規模クライアント・ライブラリーを提供します。クライアントを使用して作成されたアプリケーションは、MQ Telemetry Transport (MQTT) と WebSphere MQ Telemetry (MQXR) サービスを使用して、WebSphere MQ によるメッセージの確実なパブリッシュとサブスクライブを行います。テレメトリーをインストールするために WebSphere MQ カスタム・インストール・オプションが選択されている場合には、1) 遠隔測定 (MQXR) サービス、2) テレメトリー・クライアント、3) テレメトリー拡張クライアントがインストールされます。

### WebSphere MQ Telemetry デーモン (デバイス用)

WebSphere MQ Telemetry デーモン (デバイス用) は、拡張 MQTT V3 クライアントです。これは、組み込みシステム用に設計された、占有スペースの非常に小さな MQTT サーバーです。

### Windows NT Challenge/Response

Windows NT システムを含むネットワーク、およびスタンドアロン・システム上で使用される認証プロトコル。

### ワイヤー・タッピング (wiretapping)

通信に使われるワイヤーまたは別のタイプのコンダクターに付随して流れている情報にアクセスする行為。ワイヤー・タッピングの目的は、情報に対する無許可のアクセスを、検出されないように獲得することです。

## X

### X509

PKI に関する国際電気通信連合の標準。公開鍵証明書および公開鍵暗号方式の形式を指定します。

### XCF

[システム間カップリング・ファシリティ \(cross-system coupling facility\)](#) を参照してください。

### XID

[トランザクションの識別子 \(transaction identifier\)](#) を参照してください。

### X/Open XA

X/Open 分散トランザクション処理の XA インターフェース。分散トランザクション通信に提案されている標準です。この標準は、トランザクション内の共有リソースにアクセスするリソース・マネージャー間と、トランザクションをモニターおよび解決するトランザクション・サービス間に、双方向のインターフェースを指定します。

## IBM WebSphere MQ のアクセシビリティ機能

アクセシビリティ機能は、運動機能または視覚などに障害をお持ちの方が情報技術コンテンツを快適に使用できるようにサポートします。

### アクセシビリティ機能

IBM WebSphere MQ の主なアクセシビリティ機能は次のとおりです。

- キーボードのみによる操作
- スクリーン・リーダーを使用する操作

IBM WebSphere MQ は、最新の W3C Standard [WAI-ARIA 1.0 \(https://www.w3.org/TR/wai-aria/\)](https://www.w3.org/TR/wai-aria/) を使用して、[US Section 508 \(https://www.access-board.gov/guidelines-and-standards/communications-and-it/\)](https://www.access-board.gov/guidelines-and-standards/communications-and-it/)

about-the-section-508-standards/section-508-standards)に確実に準拠します。および [Web Content Accessibility Guidelines \(WCAG\) 2.0 \(https://www.w3.org/TR/WCAG20/\)](#)。アクセシビリティ機能を利用するには、この製品でサポートされる最新の Web ブラウザーと組み合わせて、スクリーン・リーダーの最新リリースを使用します。

IBM Documentation の IBM WebSphere MQ オンライン製品情報は、アクセシビリティに対応しています。IBM Documentation のアクセシビリティ機能については、<https://www.ibm.com/docs/about/releasenotes.html> を参照してください。

## キーボードによるナビゲーション

この製品では、標準のナビゲーション・キーを使用します。

## インターフェース情報

完全にアクセシビリティに対応した IBM WebSphere MQ の使用方法は、コマンド・ラインインターフェースを使用する方法です。コマンドの使用法の詳細については、[IBM WebSphere MQ 制御コマンドの使用法および MQSC コマンドを使用した管理](#)を参照してください。

Windows では、アクセシビリティに対応した IBM WebSphere MQ のインストール方法は、非対話式のインストールを使用する方法です。詳細については、[msiexec を使用した拡張インストール](#)を参照してください。

IBM WebSphere MQ のユーザー・インターフェースには、毎秒 2 回から 55 回フラッシュするコンテンツは含まれません。

IBM WebSphere MQ の Web ユーザー・インターフェースは、コンテンツを適切にレンダリングし、使いやすさを実現するために、カスケーディング・スタイル・シートに依存していません。ただし、製品資料では、カスケーディング・スタイル・シートが使用されています。IBM WebSphere MQ には、ハイコントラスト・モードを含む、低視力のユーザーがユーザーのシステム表示設定を使用するための同等の方法が用意されています。デバイスまたはブラウザの設定を使用して、フォント・サイズを制御できます。

## アクセシビリティの関連情報

標準の IBM ヘルプ・デスクとサポート Web サイトに加え、IBM は、聴覚が不自由なお客様が営業やサポート・サービスにアクセスするために使用できる TTY 電話サービスを立ち上げました。

TTY サービス  
800-IBM-3383 (800-426-3383)  
(北アメリカ)

## IBM とアクセシビリティ

IBM のアクセシビリティに対する取り組みについては、[IBM Accessibility\(www.ibm.com/able\)](http://www.ibm.com/able) を参照してください。

## Windows でのアクセシビリティ

IBM WebSphere MQ ユーザー・インターフェースでは、特殊キーを使用することはありませんが、その代わりにコンテキスト・メニュー、ダイアログ、およびボタンといったダイアログ制御などの項目のアクセラレーター・キーについては、Windows のユーザー・インターフェースのガイドラインに従ってください。アクセラレーター・キーには、通常の方法でアクセスします。詳細については、Windows ヘルプを参照してください (Windows ヘルプ索引で「キーボード」を検索します。アクセシビリティ・フィーチャーについては、「アクセシビリティ」を検索します)。

## アクセシビリティのための特殊フィーチャー

IBM WebSphere MQ のユーザー・インターフェースのいくつかは、通常は表示されていますが、アクセシビリティ・フィーチャーをアクティブにすると、以下のように違ってきます。

- ハイコントラスト・モード

このモードでは、Launchpad、IBM WebSphere MQ 準備ウィザード、Postcard、およびデフォルト構成はすべて、見やすくまた読みやすくするために、バックグラウンドのビットマップを隠し、システム・テキスト・カラーを使用します。

- スクリーン・リーダー・モード

スクリーン・リーダーがアクティブな場合、IBM WebSphere MQ 準備ウィザード、デフォルト構成、および Postcard は、バックグラウンド・ビットマップ、引き起こされる効果、シャドー・ボックス、およびその他のスクリーン・リーダーを混乱させてしまう可能性のあるものを隠すことによって、外観を単純化します。

- エクスプローラー・オブジェクトの状況

IBM WebSphere MQ のエクスプローラー・コンポーネントは、アイコンを使用してキュー・マネージャーなどのオブジェクトの状況を表示します。スクリーン・リーダーでは、これらのアイコンを解釈できないため、アイコンのテキスト記述を表示するオプションがあります。このオプションを選択するには、エクスプローラー内から **ウィンドウ** > **設定** > **WebSphere MQ エクスプローラー** をクリックし、「**オブジェクト名の後にオブジェクトの状況を表示**」を選択します。



## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

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

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

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

日本アイ・ビー・エム株式会社

法務・知的財産

U.S.A.

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

Intellectual Property Licensing

Legal and Intellectual Property Law

〒 103-8510

103-8510

東京 103-8510、日本

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

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

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

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

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

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N

Rochester, MN 55901

U.S.A.

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

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

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

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

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

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

著作権使用許諾:

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

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

## プログラミング・インターフェース情報

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

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

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

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

## 商標

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

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

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

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

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

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







部品番号:

(1P) P/N: