

MQSeries® Everyplace



紹介

バージョン 1

MQSeries® Everyplace



紹介

バージョン 1

ご注意

この情報およびサポートされている製品をお使いになる前に、63ページの『付録. 特記事項』にある一般情報をお読みください。

ライセンスについての警告

MQSeries Everyplace バージョン 1 ツールキットにより、開発者は MQSeries Everyplace アプリケーションを作成し、それを実行するための環境を作成することができます。

ツールキットのご購入の際のライセンス条件により、それを使用できる環境が決まります。

MQSeries Everyplace をデバイス (クライアント) として使用するために購入された場合は、その MQSeries Everyplace を使用して **MQSeries Everyplace** チャンネル・マネージャー または **MQSeries Everyplace** チャンネル・リスナー を作成することはできません。

MQSeries Everyplace チャンネル・マネージャー または **MQSeries Everyplace** チャンネル・リスナー の存在により、ゲートウェイ (サーバー) 環境が定義されますが、それには、ゲートウェイ・ライセンスが必要です。

この版は、MQSeries Everyplace バージョン 1.0、および新版において特に断りのない限り、それ以降のすべてのリリースとモディフィケーションに適用されます。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。(URL は、変更になる場合があります)

原 典 :	GC34-5843-00 MQSeries Everyplace Introduction Version 1
発 行 :	日本アイ・ピー・エム株式会社
担 当 :	ナショナル・ランゲージ・サポート

第1刷 2000.6

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 2000. All rights reserved.

Translation: © Copyright IBM Japan 2000

目次

図	v
表	vii
本書について	ix
本書の対象読者	ix
前提条件となる知識	ix
本書で使用する用語	ix
第1章 概要	1
バージョン 1.0 に関する注意事項	1
第2章 前提条件	3
第3章 MQSeries ファミリー	5
第4章 要件	11
機能	11
アプリケーション	11
顧客の要件	12
第5章 製品の概念	13
紹介	13
メッセージ・オブジェクト	14
ダンプ・データ・フォーマット	18
キュー	18
キュー・マネージャー	23
キュー・マネージャー操作	25
管理	26
管理メッセージ	27
選択的な管理	28
モニターと関連処置	29
動的チャネル	29
アダプター	31
ダイヤルアップ接続の管理	31
トレース	31
イベント・ログ	32
MQSeries Everyplace ネットワーク	32
構成とスケラビリティ	32
非同期メッセージ送達	34
同期メッセージ送達	35
セキュリティ	35
MQSeries Everyplace のローカル・セキュリティ	37
MQSeries Everyplace キュー・ベースのセキュリティ	37

メッセージ・レベルのセキュリティー	38
レジストリー	40
MQSeries Everyplace の認証可能なエンティティー	40
私用レジストリーと信任状	41
自動登録	41
公開レジストリーと認証の複写	41
レジストリー・サービスのアプリケーションの使用	42
デフォルトのミニ認証発行サービス	42
セキュリティー・インターフェース	43
構成とカスタマイズ	43
ルール	43
接続スタイル	45
対等通信接続	46
クライアント / サーバー接続	46
複数接続スタイル	47
クラス	47
アプリケーションのロード	47
第6章 MQSeries Everyplace ネットワークと MQSeries ネットワーク	49
MQSeries へのインターフェース	49
メッセージの変換	56
機能	57
互換性	57
確実な送達	58
第7章 アプリケーションとユーティリティー	59
Postcard	59
MQSeries Everyplace Explorer	59
第8章 プログラミング・インターフェース	61
付録. 特記事項	63
商標	64
用語集	65
参考文献	69
索引	71



1.	MQSeries ファミリー	5
2.	単純なホストおよび分散構成	6
3.	典型的なワークステーション構成	7
4.	典型的なデバイス構成	9
5.	簡単な MQSeries Everyplace ネットワーク	33
6.	スター型 MQSeries Everyplace ネットワーク	34
7.	MQSeries ブリッジのオブジェクト階層	50

一 表

1.	バージョン 1 でサポートされるソフトウェア環境	3
2.	MQSeries Everyplace および MQSeries のエレメント	9
3.	フィールド・オブジェクトおよびその構成フィールドのプロパティ	15
4.	属性オブジェクトのプロパティ	16
5.	メッセージ・オブジェクトのプロパティ	16
6.	用意されているメッセージ・オブジェクト・フィールド	17
7.	キューのプロパティ	21
8.	ローカル・キュー・マネージャーのプロパティ	24
9.	接続 (リモート・キュー・マネージャー) のプロパティ	24
10.	MQSeries Everyplace キューにおけるメッセージング操作	26
11.	管理メッセージ・クラス	27
12.	管理メッセージの一般構造	28
13.	認証、暗号化、および圧縮のサポート	36
14.	接続スタイル	45
15.	クラスのオプション	47
16.	ブリッジ・オブジェクトの特性	50
17.	ブリッジの特性	51
18.	MQSeries キュー・マネージャー・プロキシの特性	51
19.	クライアント接続サービスの特性	52
20.	リスナーの特性	53
21.	MQSeries リモート・キューの特性	54

本書について

本書は、MQSeries Everyplace の入門情報を扱っています。製品の概念と、この製品と他の MQSeries 製品の関連について説明します。

MQSeries Everyplace API についての詳細と、この API を使って MQSeries Everyplace アプリケーションを作成する方法については、*MQSeries Everyplace プログラミング・リファレンス* および *MQSeries Everyplace プログラミング・ガイド* を参照してください。

本書の対象読者

本書は、センサー、電話、携帯情報端末 (PDA)、ラップトップ・コンピューターなどの軽量デバイスで、安全なメッセージングを行うことに関心のあるユーザーを対象としています。

前提条件となる知識

記載されている情報を読むにあたり、前提条件となる知識は特にありませんが、安全なメッセージングの概念の基本を理解していると役立ちます。

この点についてご理解いただく上で、次の MQSeries 資料が参考になります。

- *MQSeries An Introduction to Messaging and Queuing*
- *MQSeries (Windows NT® 版) インストールの手引き V5.1*

これらの資料は、オンラインの MQSeries ライブラリーの『Book』セクションから、ソフトコピーの形で利用できます。MQSeries の Web サイト (URL アドレス <http://www.ibm.com/software/ts/MQSeries/library/>) から、この資料を利用することもできます。

本書で使用する用語

本書では、以下の用語が使われています。

MQSeries ファミリー

5ページの『第3章 MQSeries ファミリー』で説明されている MQSeries 製品の集まりを指します。

MQSeries メッセージング

5ページの『第3章 MQSeries ファミリー』で説明されている 4 つのメッセージング製品グループを指します。

MQSeries

以下の 3 つの MQSeries メッセージング製品グループを指します。

- 分散メッセージング
- ホスト・メッセージング
- ワークステーション・メッセージング

MQSeries Everyplace

4 番目の MQSeries メッセージング製品グループ (パーベイシブ・メッセージング) を指します。

デバイス

MQSeries Everyplace プログラムを実行するものの、**MQSeries Everyplace** チャネル・マネージャー、または **MQSeries Everyplace** チャネル・リスナー・オブジェクトがインストールされていない、任意のサイズのコンピューター。

注: ライセンス交付の関係上、デバイス は、*MQSeries Everyplace* クライアント と同義になります。

ゲートウェイ

MQSeries Everyplace プログラムを実行し、かつ **MQSeries Everyplace** チャネル・マネージャー、または **MQSeries Everyplace** チャネル・リスナー・オブジェクトがインストールされている、任意のサイズのコンピューター。

注: ライセンス交付の関係上、ゲートウェイ は、*MQSeries Everyplace* サーバー と同義になります。

第1章 概要

MQSeries Everyplace は、業務用の品質を持ったメッセージ送信プロダクトである、MQSeries ファミリーの一員です。センサー、電話、PDA (携帯情報端末)、およびラップトップ・コンピューターなどの軽量デバイスのメッセージ送信の必要を満たすだけでなく、モバイル接続の必要、および不安定な通信ネットワークの使用に伴って生じる要件も満たすように設計されています。MQSeries で標準の、確実に一回だけの送達を実行するとともに、他のファミリーのメンバーとメッセージを交換することもできます。MQSeries Everyplace アプリケーションの多くは、インターネット・ファイアウォールの保護の外部で実行されるため、この製品は優れたセキュリティー機能も提供します。

軽量デバイスでは、メッセージ・サブシステムが使用するシステム・リソースを節約する必要があります。したがって、MQSeries Everyplace は、システムのフットプリントとプロトコルの効率を考慮して最適化されています。そのため、MQSeries ファミリーの他のメンバーと同一のメッセージ送信機能が提供されているわけではありませんが、相互間で直接に操作することは可能です。MQSeries Everyplace は、モバイル操作、ローミング、ローカルおよびリモート・メッセージ・アクセス、信頼性の不十分なネットワークを介した場合のメッセージ送信のセキュリティーとサポートの各機能を、広範囲に備えています。

MQSeries Everyplace は、IBM® パーベイシブ・コンピューティング・ファミリーのメンバーです。したがって、他の IBM パーベイシブおよびワイヤレス製品と統合して機能するように設計されています。

バージョン 1.0 に関する注意事項

- MQSeries Everyplace バージョン 1.0 はツールキットであり、MQSeries Everyplace アプリケーションを作成し、それを実行するための環境を作成することができます。
- このリリースでは、パーベイシブ・デバイスに MQSeries Everyplace を配置するのは、アプリケーションおよびソリューション提供者の責任で行っていただきます。

第2章 前提条件

表1 は、MQSeries Everyplace バージョン 1 を実行するのに使用できるソフトウェア環境を示しています。¹

表1. バージョン 1 でサポートされるソフトウェア環境

	オペレーティング・システム
デバイス	EPOC
	Palm OS
	Windows CE
	Windows [®] 95 Windows 98 Windows NT v4 Windows 2000
ゲートウェイ	Windows NT v4 Windows 2000
注: 1. バージョン 1.0 は、Java をサポートするすべてのプラットフォームで使用できるよう、Java で提供されています。 2. リモート・キューへの同期アクセス機能だけを備えた機能限定クライアントが、Palm OS 専用の C コードベースで用意されています。 3. Java 1.1 の、プラットフォームで利用可能な最新レベルをお勧めします。テスト済みの Java のレベルの詳細については、MQSeries Everyplace Web サイト (www.ibm.com/software/mqseries/everyplace) を参照してください。	

1. MQSeries Everyplace デバイス・コードは、Java[®] を実行できるデバイスであればどれも実行できますが、表1 にリストされているオペレーティング・システムでしかテストされていません。

第3章 MQSeries ファミリー

MQSeries ファミリーには、広範囲の機能を提供する、多くの製品が含まれます。図1に示されているとおりです。

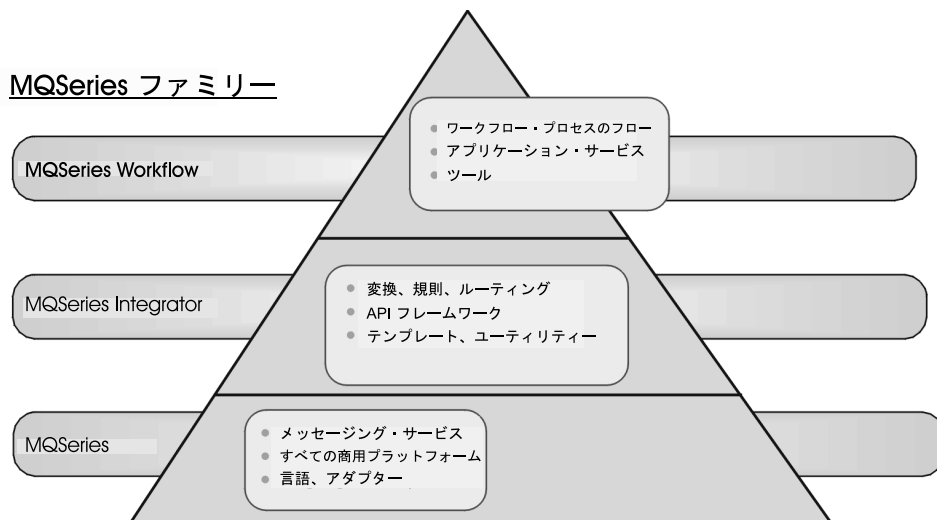


図1. MQSeries ファミリー

- **MQSeries Workflow** は、人間とアプリケーションを含めたビジネス・プロセスを自動化することで、企業の全体的な統合を容易に行います。
- **MQSeries Integrator** は、強力なメッセージ・ブローカー・ソフトウェアで、これによりリアルタイムでインテリジェントな、ルールに基づくメッセージ・ルーティング、ならびに、内容の変換とフォーマットが可能になります。
- **MQSeries Massaging** は、デスクトップからメインフレームまで 35 を超えるプラットフォームをサポートしており、ANY-TO-ANY で業務用品質のメッセージングを可能にします。

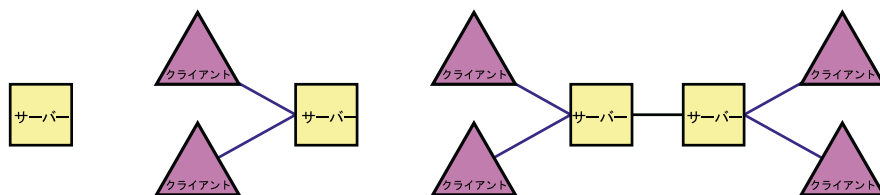
MQSeries Workflow と MQSeries Integrator 製品は、両方とも、MQSeries メッセージング層が提供する接続を利用します。

MQSeries ファミリーのメッセージングは、MQSeries (MQS) 製品と MQSeries Everyplace 製品の両方によって提供されています。それぞれ 1 つ以上のハードウェア・サーバー・プラットフォームと、それに関連するオペレーティング・システム (またはそのどちらか) をサポートするよう設計されています。これらの個々の製品は、さまざまなプラットフォームに対応でき、共通する機能と設計に基づき、いくつかの製品グループにまとめられています。4 つの製品グループがあります。

- 分散メッセージング: MQSeries (Windows NT 版、 AIX[®] 版、 AS/400[®] 版、 HP-UX 版、 Sun Solaris 版、 および他のプラットフォーム用)
- ホスト・メッセージング: MQSeries (OS/390[®] 版)
- ワークステーション・メッセージング: MQSeries (Windows 3.1、 95、 98 版)
- パーベイスブ・メッセージング: MQSeries Everyplace

メッセージング自体は、特定の製品または製品グループに関係なく、キュー・マネージャーを基にしています。キュー・マネージャーは、メッセージをそれぞれ保管できるキューを管理します。アプリケーションはローカル・キュー・マネージャーと通信し、メッセージを入手したり、キューに入れたりします。メッセージは、リモート・キューに入れられると、そのキューを所有するリモート・キュー・マネージャーに、チャンネルを介して伝送されます。このように、メッセージは、1 つ以上の中間キュー・マネージャーを経由して宛先に到達することがあります。メッセージングの基本は、必要であれば途中でメッセージをキューに入れ、送信アプリケーションと受信アプリケーションとを分離することです。すべての MQSeries メッセージング製品は、詳細においては多数の相違点があるものの、キュー・マネージャー、キュー、メッセージおよびチャンネルの同じ基本的なエレメントに関連しています。

MQSeries ホスト・メッセージング製品と分散メッセージング製品は、多数の異なるネットワーク構成をサポートするのに使用されます。これらのネットワークにははすべて、クライアントとサーバー² が含まれます。いくつかの例を、図2 に示します。



(a) スタンドアロン・サーバー

(b) クライアント / サーバー

(c) 分岐クライアント / サーバー

図2. 単純なホストおよび分散構成

最も単純なケースでは、スタンドアロン・サーバーが構成され、そこでキュー・マネージャーが実行されます。そのサーバー上で、1 つ以上のアプリケーションが実行され、キューを介してメッセージを交換します。それに代わる構成は、クライアント/サーバーです。ここではキュー・マネージャーはサーバー上にのみ存在しますが、それぞれのクライアントはクライアント・チャンネルを介してそれにアクセスします。クライアント・チャンネルとは、固有の MQSeries プロトコル (リモート・プロシーチャー呼び出し (RPC) に似たものを実現する) を送信する、両方向の通信リンクです。アプリケーション

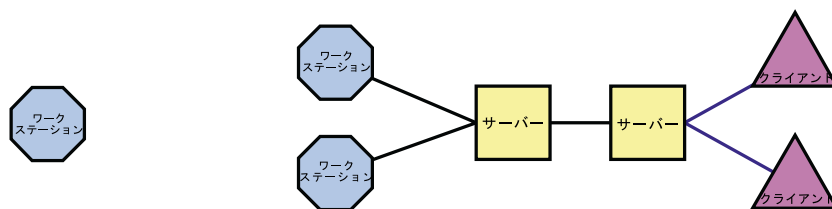
2. これらの用語は、MQSeries では非常に特殊な意味を持ち、一般的な意味と一致するとは限らないことに注意してください。本書では、これらの用語を、常に MQSeries における意味で使用します。

ンは、サーバーのキューにアクセスしながら、クライアントで実行することができます。クライアント / サーバー構成の 1 つの利点は、クライアント・メッセージングのインフラストラクチャーがサーバー・キュー・マネージャーに依存しており、軽量であるということです。欠点は、クライアントとそれに関連するサーバーが同期を取りながら動作するので、クライアント・チャンネルが常に使用可能な状態でなければならないことです。

分散クライアント/サーバー構成は、複数のサーバーが関係する、より複雑なケースです。これらの構成では、サーバーは、メッセージ・チャンネルを介してメッセージを交換します。メッセージ・チャンネルは単一方向に流れます。これは、安全で非同期のメッセージ・データ交換のために設計されたプロトコルを使用します。メッセージ・チャンネルが使用可能でなくても、クライアントは処理を続行できます。ただし、通信が使用不可の場合、サーバーとの間でメッセージをやり取りすることはできません。

MQSeries ワークステーション・メッセージング製品は、これらの構成オプションのサブセットを提供します。サーバーの代わりに、キュー・マネージャーを持つワークステーションをサポートします。しかし、クライアントの接続はサポートしません。しかし、ワークステーション同士が接続することはできますし、MQSeries メッセージ・チャンネルを介してサーバーに接続することもできます。したがって、ワークステーションは一般的に簡易なサーバーと見なされたり、非同期機能が必要な場面でクライアントの代わりに使用されたりします。

図3 には、典型的なワークステーション構成が 2 つ示されています。(b) では、ワークステーション・アプリケーションを、サーバーおよびクライアントから独立して実行することができます。



(a) スタンドアロン・ワークステーション

(b) 分散クライアント / ワークステーション / サーバー

図3. 典型的なワークステーション構成

パーベシブ・メッセージング製品である MQSeries Everyplace は、デバイスとゲートウェイを提供することによって構成をサポートします。

MQSeries Everyplace デバイスは、チャンネル・マネージャーなしで MQSeries Everyplace コードを実行するコンピューターです。つまり、デバイスが一度に通信できるデバイスまたはゲートウェイは、1 つに制限されているということです。MQSeries Everyplace

デバイスは、非常に小さなデバイス（石油パイプラインのセンサーなど）から、大きなデバイス（電話、携帯情報端末（PDA）、またはラップトップ・コンピューターなど）、デスクトップ・マシンおよびワークステーションに至るまでをカバーできます。このようなデバイス・コンピューターをパーベイシブ・デバイスということがあります。この名称は、サイズや機能の点で制限があるような印象を与えますが、実際には、製品にそのような制限はありません。

ゲートウェイとは、MQSeries Everyplace コードを実行するコンピューターで、MQSeries Everyplace チャンネル・マネージャー、またはMQSeries Everyplace チャンネル・リスナーが構成されているものを言います。これは、デバイス・コードのすべての機能に加えて、複数のデバイス・ゲートウェイと並行して通信するための機能を提供します。また、ゲートウェイは、MQSeries Everyplace ネットワークとMQSeries ネットワーク間で、メッセージを交換するためのメカニズムを提供します。

おおまかに言って、デバイスは、クライアントとサーバーの属性の多くを結合します。デバイスは、完全なキューイング機能を持つように構成することができ、こうして非同期操作を可能にすることができます。また、デバイスはリモートに保持されるキューにアクセスすることもできます。これは、サーバー・キューへのクライアント・アクセスにいくらか似た機能です。サーバーとは異なり、デバイスはクライアントを接続することができません。デバイスは、対等通信メッセージング機能を介して、直接相互に通信できます。また、デバイスはチャンネルを介して通信できます。ただし、これらのチャンネルはMQSeries Everyplace に独特のものであり、MQSeries クライアント・チャンネルやMQSeries メッセージ・チャンネルと区別するために、動的チャンネルと呼ばれます。動的チャンネルは両方向で、同期メッセージングと非同期メッセージングを含め、MQSeries Everyplace が提供する全機能をサポートします。

ゲートウェイは、デバイスと通信するために、必ずMQSeries Everyplace 動的チャンネルをサポートします。また、サーバーと通信するために、MQSeries クライアント・チャンネルをサポートします。サーバー同様、ゲートウェイにはキュー・マネージャーがあり、したがってローカル・メッセージング・アプリケーションをサポートできます。

典型的なパーベイシブ構成を9ページの図4に示します。

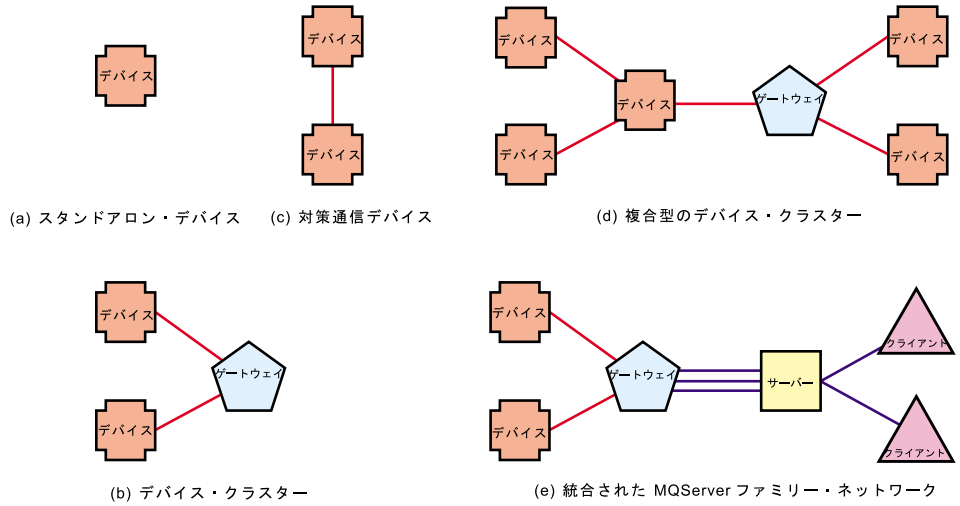


図 4. 典型的なデバイス構成

図4 の (b) では、ゲートウェイはデバイス群をまとめて接続するのに使用されます。ゲートウェイの特徴は、複数の着信接続要求を同時に扱えることです。一方、デバイスはその種の要求を一度に 1 つしか扱えません。発信要求については、ゲートウェイもデバイスも、複数同時に扱うことができます。構成 (b) で、ゲートウェイの代わりにデバイスを使用すると、2 つの端末デバイスはこの中間デバイスに順番で接続しなければなりません。一方、中間デバイスから端末デバイスへの接続は同時に行えます。(d) では、デバイス群をリンクするためにゲートウェイとデバイスの両方が使用されています。(e) では、デバイス群からなるネットワークを MQSeries サーバーに接続するために、ゲートウェイが使用されています。この構成においては、メッセージは、すべての構成要素、デバイス、ゲートウェイ、サーバー、ワークステーション、およびクライアントの間を行き来できます。これらのコンポーネントの最も重要な特性を表2 に示します。

表 2. MQSeries Everyplace および MQSeries のエレメント

コンポーネント	特性	提供元
デバイス (MQSeries Everyplace)	動的チャンネルを介してアプリケーションに確実なメッセージングを行う 同期ローカル・アクセスとリモート・キュー・アクセスの両方が可能 リモート・キューへの非同期送達が可能 着信要求は一度に 1 つしか扱えない	パーベシブ

表2. MQSeries Everyplace および MQSeries のエレメント (続き)

コンポーネント	特性	提供元
クライアント (MQS)	<p>ローカル・アプリケーションに確実なメッセージングを行う</p> <p>サーバーへの同期クライアント・チャンネル接続が必要</p> <p>キューへの同期アクセスが可能 (キューは、接続されたサーバーになければならない)</p> <p>接続されたサーバーを介して、リモート・キューへの非同期送達が可能</p>	分散ホスト
ゲートウェイ (MQSeries Everyplace)	<p>動的チャンネルを介してアプリケーションに確実なメッセージングを行う</p> <p>同期ローカル・アクセスとリモート・キュー・アクセスの両方が可能</p> <p>リモート・キューへの非同期送達が可能</p> <p>一度に複数の着信要求を扱える</p> <p>クライアント・チャンネルを介して複数のMQSeries サーバーとの接続をサポート</p>	パーペイシブ
サーバー (MQS)	<p>メッセージ・チャンネルを介してアプリケーションに確実なメッセージングを行う</p> <p>同期ローカル・キュー・アクセスが可能</p> <p>リモート・キューへの非同期送達が可能</p> <p>クライアント・チャンネルを介して複数のMQSeries クライアントとの接続をサポート</p>	分散ホスト
ワークステーション (MQS)	<p>メッセージ・チャンネルを介してアプリケーションに確実なメッセージングを行う</p> <p>同期ローカル・キュー・アクセスが可能</p> <p>リモート・キューへの非同期送達が可能</p>	ワークステーション

第4章 要件

この章では、MQSeries Everyplace の設計とインプリメンテーションの要件を説明します。

機能

MQSeries Everyplace は、次のように MQSeries ファミリーのメッセージング範囲を拡張します。

- PDA、電話、およびセンサーなどのローエンド・デバイスをサポートする。これにより、それらのデバイスが MQSeries メッセージング・ネットワークに参加できるようになります。また、ラップトップ、ワークステーション、および特定の分散プラットフォームなどの中間デバイスもサポートします。MQSeries Everyplace は、同じ質のサービス、つまり確実に一回だけのメッセージ送達を提供します。ファミリーの他のメンバーとのメッセージ交換も可能です。
- メッセージ、キュー、および関連データがストレージ内にあるか伝送中かにかかわらず、それらを保護するための広範囲に渡るセキュリティー機能を備える。
- 悪条件の通信環境（不安定なネットワーク、または帯域幅の制限が厳しい場合）でも効率的に動作する。効率的なワイヤー・プロトコル、および通信リンク障害からの自動回復機能があります。
- モバイル・ユーザーをサポートする。デバイスが移動するにつれて、ネットワーク接続ポイントを変更できます。バッテリー・リソースやネットワークに障害または制限がある場合の動作を制御することもできます。
- 適切に構成されたファイアウォールを介して動作する。
- ユーザーの管理タスクを最低限のものにする。これにより、デバイス上に MQSeries Everyplace があることを事実上意識しないですみます。MQSeries Everyplace を基盤にして、その上にユーティリティー型のアプリケーションを構築することができます。
- カスタマイズと拡張が容易。アプリケーション提供のルール、および動作を変更する他のクラスを使用したり、基本オブジェクト・クラスをサブクラス化（たとえば、別のメッセージ・タイプを表すように）したりすることができます。

アプリケーション

選択肢が多岐に渡るので、作成できる MQSeries Everyplace アプリケーションには限界がありません。特定のユーザー・グループ用に開発するカスタム・アプリケーションだけでも、相当の数に及ぶものと考えられます。以下に、考え得る例をいくつか挙げます。

- **消費者アプリケーション:** PDA で家にいながらにしてスーパーマーケットで買い物、航空会社の乗客の好みの判断、携帯電話を使った金融取引
- **制御アプリケーション:** 石油パイプライン・センサーから人工衛星経由で伝送されるデータの収集と統合、バルブなどの装置のリモート・オペレーション (オペレーターの操作の妥当性を確認するセキュリティー機能付き)
- **業務用モバイル機器:** 訪問担当者 (保険外交員)、宅配業者で顧客の受領書を迅速に発行する、ファースト・フードのウェイターがキッチンと連絡をとる、ゴルフ・トーナメントでスコアを付ける、警察用の移動セキュリティー・システムのメッセージング・システム、連絡が途絶えがちな状況にいる機械保守作業員への作業情報、家庭用メーターの読み取り。
- **個人生産性:** メール / カレンダーの複写、データベースの複写、ラップトップのダウンサイジング。

顧客の要件

MQSeries Everyplace の設計に影響する要件には、次のものが含まれます。

- **管理:** 最小限のセットアップとメンテナンス、ローカルおよびリモート両方の管理のサポート、特定のアプリケーションの必要に合わせて管理機能を拡張およびカスタマイズする能力、自動ディスカバリーおよびリカバリーの重視、選択的に使用できる独立した管理エレメントの提供。
- **通信:** 非常に効率的なワイヤー・プロトコル、最小限のヘッダー、メッセージ中に強制フィールドがないこと (固有 ID を除く)、データ・エンコードを変更する機能、圧縮、暗号化、および認証のサポート、圧縮とセキュリティー特性の終端間の折衝、ファイアウォールを容易にパススルーする機能、交換可能な通信アダプター。
- **互換性:** MQSeries と同じ質のサービスおよび直接に行うメッセージング、アプリケーションを変更せずに既存の MQSeries システムと通信する機能、MQSeries と MQSeries Everyplace の間のメッセージ交換の柔軟な制御。
- **フットプリント:** 最小構成のデバイスの場合、実質 100K バイト未満。
- **機能:** 同期および非同期メッセージング機能、ローカルまたはリモート・キューのどちらかで保持されているメッセージへのアクセス、メッセージ内のフィールドをどれでも選択的な検索に使用可能、選択的な検索用のメッセージにあるフィールドを使用する機能、キュー用の支援メディアの選択的制御。
- **ルールのサポート:** ルールを介してさまざまな動作を制御 (たとえば、メッセージ送信の時刻、通信リンクの再試行の頻度、メッセージが大きすぎる場合の処理、ターゲット・キューがいっぱいの場合の動作など)。
- **セキュリティー:** セキュリティー、認証、否認なしの完全サポート、メッセージ・レベルとキュー・レベルのセキュリティー、セキュリティー攻撃からのメッセージ・システムの保護、業界標準アルゴリズムを使用したプラグ可能なセキュリティー、オペレーティング・システム・ユーザー信任を統合する機能、国内のセキュリティー要件に準拠して、国境を越えるメッセージのセキュリティー・サポートを変える機能。

第5章 製品の概念

紹介

MQSeries Everyplace プログラミング・モデルの基本的なエレメントは、メッセージ、キュー、およびキュー・マネージャーです。MQSeries Everyplace メッセージは、アプリケーションによって定義される内容を含むオブジェクトです。保管時にはメッセージはキューに保持され、そのようなメッセージは MQSeries Everyplace ネットワークを介して移動することができます。メッセージは、ターゲット・キュー・マネージャーとキュー名の対を指定することによって、ターゲット・キューに送られます。アプリケーションは、put 操作を使ってキューにメッセージを入れ、通常は get 操作を使用してそれらを取り出します。キューにはローカルなものトリモートなものがあり、どちらもキュー・マネージャーによって管理されます。デバイスとゲートウェイの両方が、レジストリーに構成データを保管します。

デバイス上のアプリケーションは、デバイス上で使用可能な API か機能をどれでも利用できます。MQSeries Everyplace プログラミング・インターフェースだけに限定されてはいません。動的チャンネルを介して、MQSeries Everyplace は他の MQSeries Everyplace デバイスおよび MQSeries Everyplace ゲートウェイ (またはそのどちらか) に接続できます。

ゲートウェイのアプリケーションも、MQSeries Everyplace プログラミング・インターフェースだけでなく、ゲートウェイ上で使用可能な API か機能をどれでも利用できます。ゲートウェイは、動的チャンネルを介して他のゲートウェイ、および MQSeries Everyplace デバイス (またはそのどちらか) に接続できます。MQSeries クライアント・チャンネルを使用すると、ゲートウェイは 1 つ以上の MQSeries サーバーに接続できます (しかし、他の MQSeries Everyplace ゲートウェイには接続できません)。MQSeries Everyplace と MQSeries の両方を 1 つのマシンに共存させることができます (ただし、MQSeries はなくてもかまいません)。

完全な機能を備えたデバイスとゲートウェイは、以下の点を除いて同一の機能を持ちます。

- ゲートウェイは同時に複数の着信要求 (他のデバイスおよびゲートウェイ、またはそのどちらか一方からの) を処理できる
- ゲートウェイは同時に複数の MQSeries サーバーと情報交換できる

動的チャンネルは、次のネットワーク接続をサポートします。

- ダイアルイン接続
- 永続接続 (従来型の LAN、専用回線、赤外線または無線 LAN など)

通信プロトコルは、アダプターのセットによってインプリメントされます。サポートされるプロトコルごとに、アダプターが 1 セットずつ必要です。これによって、必要に応

じて新しいプロトコルを追加することが可能になります。さらに、ある環境のメモリー・フットプリントを、特定の構成に合わせて調整することができるようになります。

キューは、別のアダプターのセットを介して、ストレージ・メディアに個別にマップされます。したがって、選択したアダプターに応じて、キューは、ファイル・システムまたはメモリーに保管できます。

MQSeries Everyplace プログラミング・インターフェースは、キューの位置に関係なくアプリケーションを作成できるように設計されています。したがって、ローカル・キューにアクセスするために設計されたプログラムは、変更を加えることなしに、リモート・キュー・マネージャーから実行できなければなりません (有効なセキュリティ考慮事項を満たす必要があります。また、特定の MQSeries Everyplace 操作はリモート・キューではサポートされていません)。この独立性には、管理機能の使用が含まれます。

メッセージ・オブジェクト

MQSeries Everyplace メッセージ・オブジェクトは、MQSeries がサポートするメッセージとは基本的に異なります。MQSeries では、メッセージはバイト配列で、メッセージ・ヘッダーとメッセージ本体に分割されています。MQSeries は、メッセージ・ヘッダーを理解します。この部分には、キューへの応答やキュー・マネージャーへの応答の識別、メッセージ ID、および相関 ID などの重要な情報が入っています。MQSeries は、メッセージ本体を理解しません。

対照的に、MQSeries Everyplace のメッセージは、フィールド・オブジェクト と呼ばれる MQSeries Everyplace オブジェクトを継承するメッセージ・オブジェクトです。メッセージは、ヘッダーやメッセージ本体の概念を持たない、真のオブジェクトです。基本フィールド・オブジェクトを理解しないと、メッセージ・オブジェクトの本質を明確に理解できません。これらのフィールド・オブジェクトは、MQSeries Everyplace で広範囲に使用される、フィールドの集合です。各フィールドは、名前、データ・タイプ、およびデータそのものから成ります。フィールド名は、長さが無制限の ASCII 文字ストリング (複数の予約文字を除く) です。

フィールド・タイプは次のとおりです。

- **ASCII** ストリングまたは ASCII ストリングの動的配列
- **ブール** 値
- **バイト** (固定配列)、またはバイト値の動的配列
- **二重浮動小数点** (固定配列)、または二重浮動小数点値の動的配列
- **フィールド・オブジェクト**または**フィールド・オブジェクトの動的配列** (したがってフィールド・オブジェクトのネストはサポートされる)
- **浮動小数点値** (固定配列)、または浮動小数点値の動的配列
- **整数** (4 バイト、固定配列)、または整数の動的配列
- **長整数** (8 バイト、固定配列)、または長整数の動的配列

- **短整数** (2 バイト、固定配列)、または短整数の動的配列
- **UNICODE** ストリングまたは **UNICODE** ストリングの動的配列

フィールド・オブジェクトには**タイプ**があり、タイプはプログラミング・オブジェクト・クラス名に対応します。このオブジェクト・クラスの子孫はメッセージ・オブジェクトとしてアプリケーション・プログラムに使用され、そのタイプは、必要な時に **MQSeries Everyplace** が正しいオブジェクト・クラスのインスタンスを生成するために使用されます (たとえば、メッセージ・オブジェクトがチャンネルを介して送信された後など)。

フィールド・オブジェクトには、多くのメソッドが備わっています。たとえばフィールドを列挙したり、その存在を検査したりできます。同様に、フィールド・オブジェクト同士の同等性を比較できます。また、フィールド項目をバイト配列にダンプしたり、それを復元したりする機能があります。これは、たとえば、リンクを介してオブジェクトを転送するためにデータを準備したり、転送後にオブジェクトを復元したりするために使用します。ダンプおよび復元メソッドはオーバーライドできます。これは、フィールド・オブジェクトが、自分自身を別の方法でシリアル化できるようにするためです。たとえば、転送時にフィールドの内容をデータベースで照会する場合などです。表3 に、フィールド・オブジェクトとその構成フィールド (またはそのどちらか) のプロパティをリストします。

表3. フィールド・オブジェクトおよびその構成フィールドのプロパティ

プロパティ	存在	
	フィールド・オブジェクト	フィールド
関連属性オブジェクト	オプション	
構成フィールド	あり	
Hidden		あり
Name		あり
Type	あり	あり
Value		あり

フィールドの **Hidden** プロパティによって、比較操作の目的でフィールドを無視することができます。

属性オブジェクトには、認証、暗号化および圧縮を実行するメカニズムが入っており、フィールド・オブジェクトと関連付けることができます。

- **認証:**アクセスの制御
- **圧縮:**ストレージ要件の削減 (伝送およびストレージ、またはそのどちらか)
- **暗号化:**オブジェクトをダンプする時に内容を保護 (および復元を許可)

属性オブジェクトは **MQSeries Everyplace** セキュリティー・モデルにとって基本的なものであり、これによって、内容への選択的なアクセス、支援ストレージ上のデータの保

護、および他のさまざまな使用法が可能になります。表4 では、属性オブジェクトのプロパティをリストします。Rule 値は (存在する場合)、許可される操作を制御します。

表4. 属性オブジェクトのプロパティ

プロパティ	存在
Cryptor	オプション (状況によっては必須)
Authenticator	オプション
Compressor	オプション
Rule	オプション
Type	オプション

メッセージ・オブジェクトは、フィールド・オブジェクトから派生し、MQSeries Everyplace が生成する UID (固有 ID) を含みます。この UID は、メッセージ・オブジェクトを固有に識別します。次のものから構成されています。

- 発信元のキュー・マネージャーの**名前** (オブジェクトの受信時に、キュー・マネージャーが追加します)。この名前は、グローバルに固有でなければなりません。
- メッセージ・オブジェクトが作成された**時間** (作成時に追加されます)。

メッセージ・オブジェクトには、フィールド・オブジェクトとして継承するプロパティに加え、表5 でリストされる基本的なプロパティがあります。

表5. メッセージ・オブジェクトのプロパティ

プロパティ	説明
Msg_OriginQMgr	メッセージを送信するキュー・マネージャーの名前
Msg_Time	メッセージ・オブジェクトがアプリケーションによって作成された時間

これらの 2 つのプロパティは、メッセージ・オブジェクトの固有 ID (UID) を構成します。

MQSeries Everyplace キュー・マネージャーあてのメッセージには、他の情報は必要ありません。もっとも、メッセージ情報の内容を伝えるための、他のフィールドが含まれるのが普通です。一般的には、メッセージは基本メッセージ・オブジェクト・クラスの子孫なので、その目的に合った追加のフィールドがあります。これらの追加フィールドのうちいくつかは、幅広いアプリケーションで共通に使用されます (キュー・マネージャーへの応答など)。したがって MQSeries Everyplace は、それらのフィールドをある程度サポートします。

表6 では、サポートされるフィールドをリストします。

表6. 用意されているメッセージ・オブジェクト・フィールド

フィールド名	使用法
Msg_CorrelID	通常は、応答と元のメッセージと関連させるのに使用されるバイト・ストリング
Msg_ExpireTime	メッセージを削除できるようになる時間 (未送達の場合も含む)
Msg_LockID	メッセージのロックを解除するのに必要なキー
Msg_MsgID	元のメッセージとの関連のためにアプリケーションが使用する
Msg_Priority	メッセージの優先順位
Msg_ReplyToQ	メッセージ応答の宛先となるキューの名前
Msg_ReplyToQMgr	メッセージ応答の宛先となるキュー・マネージャーの名前
Msg_Resend	以前のメッセージの再送メッセージであることを示す標識
Msg_Style	要求 / 応答などからコマンドを区別する

どの場合についても、定義された定数を使って、フィールド名を単一バイトで伝えることが可能です。フィールドによっては、さらに拡張された機能が備わっています。たとえば、優先順位 (存在する場合) はメッセージが伝送される順序に影響します。相関 ID は、迅速に検索を行えるよう、キューでそのフィールド値に基づいて索引付けを開始します。また、有効期限はメッセージの期限切れ処理を開始します。

メッセージ・オブジェクトには、たくさんのメソッドが定義されています。たとえば、メッセージ UID、発信元のキュー・マネージャー名、およびオブジェクト作成時刻を抽出する機能などです。他にも役立つメソッドがフィールド・オブジェクト・クラスから継承されます。たとえば、フィールド値の `get` や `put` を行うさまざまなメソッドがあります。特に興味深いものとして `dump` メソッドがあります。これは、オブジェクト・データをバイト・ストリングにダンプするのに使用されます。MQSeries Everyplace は、メッセージを永続ストレージに保管する場合や、動的チャンネルを介して伝送する場合に、このメソッドを呼び出します。この方法によって、メッセージ・オブジェクト自体が自分のデータ値の外部表現を決める責任を持つこととなります。そして、これは多くの方法で活用できます。たとえば、オブジェクトは単に構成フィールドの値をダンプすることもできますし、代わりにデータベースを照会することを選択することもできます。これを補完する `restore` メソッドは、ダンプされたフォーマットからオブジェクトを再作成する時に、同様の制御を行うことを可能にします。メッセージ・オブジェクトに属性オブジェクトが付加されている場合、属性のデータ・エンコーダー、暗号化機能、および圧縮機能はダンプ時に起動します。同様に、デコーダー、暗号解除機能、解凍機能は復元時に起動します。

MQSeries Everyplace がメッセージ・オブジェクトを送信する場合、ワイヤー上のフットプリントを削減するため、関連したクラス・ファイルは送信しません。したがって、メ

メッセージ・オブジェクトのインスタンスが生成される各キュー・マネージャーで、適切なメッセージ・クラスが使用可能でなければなりません。

デフォルトのメッセージ・オブジェクトの `dump` メソッドは、効率的なメッセージ・ストレージと伝送を可能にするため、生成されるバイト・ストリングのサイズが最小になるよう最適化されています。

ダンプ・データ・フォーマット

デフォルトのダンプ・データ・フォーマットは、次のようにフィールドをエンコードします。

```
{Length Identifier Fence {Data}} {Length Identifier Fence {Data}} { ... }
```

ここで、

- *Data*: データ値。整数は、先行する 0 と F を除去した形で圧縮されます。ブール値には関連データ・バイトはありません。
- *Fence*: ID とオプションのデータ項目間の境界を区切る、特殊バイト。このバイトも、Data 項目タイプを示します。
- *Identifier*: フィールド名を、可変長の ASCII ストリングのバイト (エンド・バイトで終了) に保持します。
- *Length*: データ・フィールドの長さを示します。1 ~ 4 の間の可変数のバイトが使用されます。最初のバイトの最初の 2 ビットは、長さフィールドの長さを示すために予約されています。範囲 0 ~ 1,073,741,823 の長さがサポートされます。

この結果、データ・ストリームが効率的に圧縮されます。データを圧縮すると、さらに節約が可能です。前のバイト・ストリームの XOR 圧縮がよい結果を生むと考えられる場合もありますが、これらのフィールドが可変的な性質を持つこと、およびフィールドの順序が変更可能であることのため、単純な XOR が常に予想する効果をもたらすとは限りません。MQSeries Everyplace には、フィールドごとに動作し、大幅に圧縮を改善する可能性のある、インテリジェントな XOR が含まれています。

キュー

キューは、通常、アプリケーション・プログラムによる除去を免れているメッセージ・オブジェクトを保持するのに使用されます。メッセージと同様、キューもフィールド・オブジェクトから派生します。アプリケーションがキュー・オブジェクトに直接アクセスすることは許可されていません。³ その代わりに、キュー・マネージャーがアプリケー

3. 直接アクセスは、キュー・ルールの内部では許可されています。

ション・プログラムとキューの間の仲介を行います。キューは名前によって識別されま
す。名前は長さが無制限の ASCII 文字ストリングですが、⁴ 特定のキュー・マネージャ
ー内で固有でなければなりません。

MQSeries Everywhere は、数多くの異なるキュー・タイプをサポートしています。

ローカル・キュー

ローカル・キューは、アプリケーションがメッセージを安全に保護された方法
で保管するために使用されます。ローカル・キューにはメッセージ・ストアが
あります。メッセージ・ストアは、アダプター・クラス (通常はディスク・ア
ダプター・クラス) を介してアクセスされます。しかしながら、MQSeries
Everyplace には、メモリー・アダプター・クラスも用意されています。これ
は、高速なアクセスができるよう、メッセージをメモリーに保持するものです
(システム破壊時にメッセージを失う危険性が伴います)。適切なアダプターを
作成することによって、メッセージは、キュー単位でどこにでも保管できます
(リレーショナル・データベース、書き込み可能 CD など)。ローカル・キュー
は、オンラインでもオフラインでも、つまりネットワークに接続されているか
どうかに関係なく使用できます。アクセスとセキュリティは、キューが責任
を持ちます。ネットワークへの接続時に、リモート・キュー・マネージャーか
ら使用の権限を受けることができるので、他の人がキューとメッセージをやり
取りすることが可能になります。ローカル・キュー・アクセスは常に同期をと
って行われます。

リモート・キュー

リモート・キューはローカル環境には存在しません。ただし、キューを所有す
るキュー・マネージャーと実際のキューを識別する定義は、ローカルに存在し
ます。リモート・キューは、同期的にも非同期的にもアクセスできます。リモ
ート・キューの定義がローカルに保持されている場合、アクセスのモードはこ
の定義に基づいて決まります。定義がない場合、キュー・ディスカバリーが発
生し、その結果特性が見つかり、アクセスのモードは強制的に同期になりま
す。

同期 キューとは、所有するキュー・マネージャーへのパスを持つネットワー
クに接続されている場合にのみアクセスされるキューです。ネットワークが確立
されていない場合、put、get、およびブラウズなどの操作 (26ページの表10 を
参照) によって、例外が発生します。所有するキューは、キューにアクセスす
るために必要なアクセス許可、およびセキュリティ要件を制御します。メッ
セージの送受信時に、エラーや再試行を扱うのはアプリケーションの責任であ
り、この場合、MQSeries Everywhere が確実に一回だけの送達の実行を持つ必要
はなくなります。

4. インターオペラビリティのため、MQSeries の命名の制限に従うことをお勧めします。これには、名前の最大長が 48 文字という規
則も含まれます。この長さは、使用するファイル・システムの制限を受けることもあります。

非同期 キューは、put はできても検索できないキューです。ネットワーク接続が確立されている場合、メッセージは所有するキュー・マネージャーおよびキューに送信されます。しかし、ネットワークが接続されていない場合は、メッセージはネットワーク接続が確立されるまでローカルに保管され、その後伝送されます。これにより、デバイスがオフラインの時に、アプリケーションがキューを操作できるようになります。しかし、これは、このタイプのキューが、メッセージを一時保管するためにメッセージ・ストアを持つ必要があるということではありません。

ストア・アンド・フォワード・キュー

このタイプのキューは、次のキュー・マネージャー（必ずしも所有するキュー・マネージャーとは限らない）に転送されるまで、メッセージを保管します。このタイプのキューは通常（絶対ではない）ゲートウェイで定義され、デバイスがネットワークに接続された時にメッセージを収集することになります。ストア・アンド・フォワード・キューは、多数のターゲット・キュー・マネージャーへのメッセージを保持することもありますし、ターゲット・キュー・マネージャーごとに 1 つのストア・アンド・フォワード・キューを設ける場合もあります。送信側は、切断されている可能性のある受信側にメッセージを送信したい場合でも、メッセージをその受信側のキュー・マネージャーまたはキューにあてて送ります。すると、中間サーバーが、受信側が接続されていないことを検出し、ローカル・メッセージ・ストアにメッセージを保管します。送信側アプリケーションは、ターゲット・キューが接続されていても切断されていても、メッセージを送信するために何らかの変更を加える必要はありません。

ホーム・サーバー・キュー

このタイプのキューは、通常、デバイス上に存在し（たびたび接続されることが前提となる）、ホーム・サーバーと呼ばれるキュー・マネージャー上のストア・アンド・フォワード・キューを指します。ホーム・サーバー・キューは、デバイスがネットワークに接続されると、ホーム・サーバーからメッセージを取り出します。サーバーからメッセージを取り出すと、キューは `putMessage`、および `confirmputMessage` メソッド呼び出し（25ページの『キュー・マネージャー操作』を参照）を使用して、そのメッセージをローカル・キューに渡します。この時点で、メッセージを正しいローカル・キューに入れることは、キュー・マネージャーの責任になります。サーバーからメッセージを入手する `pull` メソッドは、ネットワークを介した流れの観点からすると、サーバーがメッセージを `push` するよりも効率的です。これは、ホーム・サーバー・キューが、最初のメッセージへの肯定応答を次のメッセージ（もしあれば）への要求としても使用するのに対し、サーバーの `push` には、メッセージを送信するための要求 / 応答、および確認の流れのための 2 番目の要求 / 応答が必要だからです。ホーム・サーバー・キューには通常、ネットワークの接続中、サーバー上の保留メッセージの有無をチェックさせる、ポーリング間隔が設定されています。ポーリング間隔は、管理構成オプションの 1 つです。

管理キュー

このタイプのキューは、MQSeries Everyplace 管理メッセージを受け取ります。オプション管理メッセージ応答キューも、MQSeries Everyplace システムが送信する管理メッセージへの応答を受け取るのに使用されます。管理キューは、管理を実行する方法は理解していませんが、管理の詳細をカプセル化したメッセージを処理します。

MQSeries ブリッジ・キュー

これは特殊な形式のリモート・キューです。定義がゲートウェイ上にあり、ターゲット・キューがMQSeries キュー・マネージャー上にあります。この形式のキューは、MQSeries Everyplace 環境とMQSeries 環境の間のパスとなります。必要なデータまたはメッセージの再フォーマット設定を実行する、変換機能が使用されます。MQSeries Everyplace には、非常に基本的な変換機能が用意されています。プログラマーは、この変換機能を自分の要件に合うようにカスタマイズする必要があります。

MQSeries Everyplace はデータを安全にキューに保管します。これは、メッセージがオペレーティング・システムによって単にバッファーに入れられるということではなく、メディアに物理的に書き込まれるということです。しかし、MQSeries Everyplace は独立して変更をメッセージおよびキューにログ記録するわけではありません。メディア障害からの回復が必要である場合、RAID ディスク・システムの使用など、ハードウェア上の解決策も配備する必要があります。あるいは、キューを回復可能なストレージ (特定のデータベース・サブシステムなど) にマップしなければなりません。

MQSeries Everyplace では、キュー・マネージャーは、定義されたキューがなくてもかまいません。しかし、必要であれば4つのシステム・キューを使用できます。

- **AdminQ:** 管理メッセージの受信に必要
- **AdminReplyQ:** 管理メッセージへの応答を受信するため、オプションで使用される
- **DeadLetterQ:** 他の方法では送達できないメッセージを保管するのに使用
- **SYSTEM.DEFAULT.LOCAL.QUEUE:** MQSeries サーバー上の必須システム・キューと共通の名前を共用するキュー

キューのプロパティが表7に示されています。ただし、リストされているすべてのプロパティがすべてのキュー・タイプに適用されるとは限らないことに注意してください。

表7. キューのプロパティ

プロパティ	説明
Admin_Class	キュー・クラス
Admin_Name	ASCII キュー名
Queue_Active	キューがアクティブであることを示す
Queue_AttRule	セキュリティの運用を制御するルール・クラス
Queue_Authenticator	Authenticator クラス

表7. キューのプロパティ (続き)

プロパティ	説明
Queue_BridgeName	所有する MQ ブリッジ名
Queue_ClientConnection	クライアント接続名
Queue_CreationDate	キューが作成された日付
Queue_Compressor	Compressor クラス
Queue_Cryptor	Cryptor クラス
Queue_CurrentSize	キュー上のメッセージ数
Queue_Description	Unicode 記述
Queue_Expiry	メッセージの有効期限
Queue_FileDesc	キューが保管されている位置
Queue_MaxMsgSize	キューで許可されているメッセージの最大長
Queue_MaxQSize	許可されている最大メッセージ数
Queue_Mode	同期または非同期
Queue_MQMgr	MQSeries キュー・マネージャー・プロキシー
Queue_Priority	メッセージに関して使用される優先順位 (メッセージ値でオーバーライドされない場合)
Queue_QAliasNameList	キューの代替名
Queue_QMgrName	実際のキューを所有するキュー・マネージャー
Queue_RemoteQName	リモート MQSeries フィールド名
Queue_Rule	キュー操作のルール・クラス
Queue_TargetRegistry	ターゲット・レジストリー・タイプ
Queue_Transporter	Transporter クラス
Queue_Transformer	Transformer クラス

管理機能は、キューの作成および削除、およびそのプロパティの調査または変更のために使用されます。

キューは、メッセージ・ストアとしての使用に限定されてはいません。サブクラス・キューは、プロセス制御アプリケーションのシナリオで使用できます。たとえば、キュー・オブジェクトによって直接バルブを制御することができます。正しいタイプのメッセージによって、バルブを開いたり、流出量を変えたりすることができます。アプリケーションが、キューからメッセージを pull したり、処置を実行したりするのではなく、キュー・オブジェクト自体が処置を制御します。たとえば、他のキューはスプレッドシートを更新したり、テキストの話法の切り替えを行ったりします。この技法の利点は、キューのセキュリティ面が適所にあって有効であり、さらに確実なメッセージングが行われることです。したがって、MQSeries Everyplace は一回限りの確実なメッセージ送信を行い、関連する認証機能と暗号化機能は、権限を持つメッセージ送信者のみができるようなメッセージを送信できることを保証し、転送中、メッセージの内容は確実に保護

されることとなります。どのアプリケーションもキューへのアクセスを許可されません。実際、アクセスはまったく必要ありません。

キュー・マネージャー

MQSeries Everyplace キュー・マネージャーは、アプリケーションがメッセージとキューにアクセスできるようにするとともに、チャンネルがあればそれを制御します。MQSeries Everyplace バージョン 1.0 では、単一の Java 仮想計マシン (JVM) 上で一度にアクティブになれるキュー・マネージャーは 1 つだけです。複数の JVM がマシン上にある場合、JVM と同じ数のキュー・マネージャーが存在できます。キュー・マネージャーは名前によって識別されます。この名前はグローバルに固有でなければならず、⁵ 長さが無制限の ASCII 文字ストリングでなければなりません。⁶ キュー・マネージャーは、ローカル・キューの有無に関係なく構成できます。すべてのキュー・マネージャーは同期メッセージ操作をサポートします。ローカル・キューを持つキュー・マネージャーは、非同期メッセージ送達もサポートします。

非同期メッセージ送達と同期メッセージ送達は、特性と結果が大いに異なります。

非同期メッセージ送達⁷

非同期メッセージ送達では、アプリケーションは、リモート・キューへの送達のために MQSeries Everyplace にメッセージを渡します。put 操作が行われると、即座にアプリケーションに戻ります。MQSeries Everyplace は、送達できるようになるまで、一時的にメッセージをローカルに保持します。送達は、MQSeries Everyplace の責任で行われます。この操作モードでは、**確実に一回だけの送達**が行われます。詳細は、34ページの『非同期メッセージ送達』を参照してください。

同期メッセージ送達:

同期メッセージングは、以下のキューに対して行われます。

- MQSeries Everyplace ネットワークを介してルーティングされた、MQSeries Everyplace キュー・マネージャー上のターゲット・キュー
- MQSeries Everyplace ゲートウェイに直接接続された、MQSeries キュー・マネージャー上のターゲット・キュー
- 間接的に接続された MQSeries サーバー上のターゲット・キュー

同期メッセージングでは、アプリケーションは、送達のためにメッセージを MQSeries Everyplace に put します。MQSeries Everyplace は、同期を取って

5. MQSeries Everyplace または MQSeries がこの制限を強制施行することはありませんが、キュー・マネージャー名が重複すると、メッセージが間違ったキュー・マネージャーに送達される可能性があります。

6. インターオペラビリティのため、MQSeries キュー・マネージャーの命名規則に従うことをお勧めします。これには、名前の最大長を 48 文字に制限することが含まれます。この長さは、使用するファイル・システムの制限を受けることもあります。

7. MQSeries Everyplace は、MQSeries が提供する持続性モードと非持続性モードを区別せず、持続性モードのみをサポートします。しかし、必要であれば、キュー支援ストレージの選択により、パフォーマンスと回復性のバランスを取ることができます。

ターゲット・リモート・キューに接続し、メッセージをキューに入れます。送達後、MQSeries Everyplace はアプリケーションに戻ります。

リモート・キュー・マネージャーとの接続には、中間デバイスとゲートウェイ (またはそのどちらか) を介した、MQSeries Everyplace のルーティングが関係しています。詳細は、35ページの『同期メッセージ送達』を参照してください。

したがって、非同期メッセージ送達とは、ローカル・アプリケーションは MQSeries Everyplace にメッセージを渡し、そのローカル・キュー・マネージャーから先の送達は、MQSeries Everyplace の責任で行われるという意味です。つまり、ネットワークと受信側アプリケーション (またはそのどちらか) が使用可能である必要はありません。実際に送達が行われる時間は、送信側アプリケーションには通知されません。同期メッセージ送達では、ネットワークが稼動していることが必要ですが、送信側アプリケーションには、メッセージが受信側アプリケーションのキューに送達されたことが通知されます。受信側アプリケーションは、非同期の場合も同期の場合も、使用可能である必要はありません。

ローカル・キュー・マネージャーのプロパティは、キューのローカル管理を反映します。また、接続する必要があるリモート・キュー・マネージャーごとに、接続定義が必要です。したがって、接続定義は、リモート・キュー・マネージャー定義 と呼ばれることもあります。これらの定義には、キュー・マネージャー間の直接通信に必要な情報 (ネットワーク・アドレスを含む) がすべて含まれる場合もありますし、単に通信が中間キュー・マネージャーを介する間接的なものであることを示すだけである場合もあります。後者の場合、必要なのはネクスト・ホップのキュー・マネージャーの名前だけです。

キュー・マネージャーのプロパティを、表8 および 表9 で示します。

表8. ローカル・キュー・マネージャーのプロパティ

プロパティ	説明
QMgr_ChnlAttrRules	チャンネル属性ルール
QMgr_ChnlTimeout	チャンネル・タイムアウト
Admin_Class	キュー・マネージャー・クラス
QMgr_Description	Unicode 記述
Admin_Name	キュー・マネージャー名
QMgr_Rules	キュー・マネージャー操作のルール・クラス

表9. 接続 (リモート・キュー・マネージャー) のプロパティ

プロパティ	説明
Con_Adapter	アダプター・ファイル記述子
Con_AdapterOptions	アダプター・オプション (ヒストリーの使用など)

表9. 接続 (リモート・キュー・マネージャー) のプロパティ (続き)

プロパティ	説明
Con_AdapterParm	アダプターが使用する ASCII データ (サブレット名など)
Con_Aliases	キュー・マネージャーの代替名
Con_Channel	この接続が使用するチャンネルのタイプ
Con_Description	Unicode 記述
Queue_QMgrName	定義の所有者
Admin_Name	キュー・マネージャー名

1 つの接続定義で複数のアダプターがサポートされています。

キュー・マネージャー操作

キュー・マネージャーは、メッセージング操作をサポートし、オプションでキューの管理も行います。アプリケーションは、キュー・マネージャーで実行される操作を介して、メッセージにアクセスします。フィルターが指定されない限り、キューで最初に使用可能なメッセージが取り出されます。フィルターは、等価なものを探すために突き合わされるフィールド・オブジェクトです。メッセージ内のどのフィールドでも、選択的な検索のために使用できます。 `get` 操作は、すべてのメッセージ送信操作および検索操作と同様、メッセージのエンコードとデコードに使用するための属性オブジェクトを、オプションで受け取ることができます。

MQSeries Everyplace では、MQSeries と同様、`get` は通常破壊的な操作です。確実なメッセージングが MQSeries Everyplace とアプリケーションの間で必要な場合、`get` の後に確認メソッドのシーケンスを使用する必要があります。まず、確認 ID (アプリケーションがその値を選択する) を指定して `get` が発行されます。この操作によってメッセージが入手されますが、メッセージはすぐに削除されず、キューに隠されます。続いて、元のメッセージ UID が指定されて確認操作が行われます。これはアプリケーションのために `get` が正常に実行されたことを示すものとなります。メッセージが削除されるのは、その後です。`get` が失敗すると、メッセージは回復されます。`put` 操作も同じように動作します。

UID を指定することにより、メッセージを検索しないでキューから削除 することができます。

破壊的でない読み取りが必要な場合、メッセージを探して、キューがブラウズ されます (オプションで、フィルターを使ってブラウズすることも可能です)。ブラウズによって、フィルターと一致するすべてのメッセージ・オブジェクトが取り出されますが、それらはキューに残されたままです。ロックされた状態でブラウズ もサポートされています。これには、キュー上の一致するメッセージをロックする追加機能があります。メッセージは個別に、またはフィルターによって識別されるグループ単位にロックされます。ロック操作はロック ID を戻します。ロックされたメッセージは、ロック ID が提

供された場合のみ入手または削除されます。ブラウザに関するオプションを指定することによって、メッセージ全体を戻すことも、UID だけを戻すこともできます。

アプリケーションは、メッセージがキューに届くのを、指定された時間だけ待機 することができます。オプションで、フィルターを使って関係のあるメッセージを識別することができます。また、確認 ID を指定することもできます。別の方法として、アプリケーションは MQSeries Everyplace メッセージ・イベントを listen することができます。この場合もフィルターをオプションで使用できます。リスナーは、メッセージがキューに届いたときに通知を受けます。

キューは、表10 で示されているように、メッセージング操作に使用できます。

表 10. MQSeries Everyplace キューにおけるメッセージング操作

	ローカル・キュー	リモート・キュー ¹	
		同期	非同期
ブラウザ (±ロック、±フィルター)	あり	あり	
削除	あり	あり	
Get (±フィルター)	あり	あり	
Listen (±フィルター)	あり		
Put	あり	あり	あり
待機 (±フィルター)	あり	あり	

注:

1. 同期リモート待機操作は、リモート・キューのポーリングを介して実現されるので、実際の待ち時間はポーリング時間の倍数になります。
2. ¹MQSeries Everyplace バージョン 1.0 に提供される MQSeries Everyplace MQSeries Bridge は、'PUT' 操作だけをサポートします。

キュー・マネージャーは、オプションで、開始直後にアプリケーション (クラス) をロードすることができます。同様に、シャットダウン時にアプリケーションを終了することができます。キュー・マネージャーは、状況またはエラーを反映するイベントを生じます。デフォルトでは、これらのイベントはイベント・ログに表示されます。

管理

管理は、キューおよび接続などの MQSeries Everyplace リソースを構成および管理するための機能を提供します。メッセージ関連の機能は、アプリケーションの担当と見なされます。管理は、管理メッセージの生成と受信を扱うインターフェースを介して使用可能にされ、ローカル管理とリモート管理が同じ方法で処理されるように設計されています。要求は、ターゲット・キュー・マネージャーの管理キューに送信されます。必要な場合、応答を受信できます。ローカルまたはリモート MQSeries Everyplace アプリケーション・プログラムはどれも、ヘルパー・メソッドを介して、直接的または間接的に管

理メッセージを作成および処理できます。また、管理メッセージは MQSeries Everyplace Explorer⁸ (システム管理用のグラフィカル・ユーザー・インターフェースを提供する管理ツール) を介して、間接的に生成することもできます。

管理キューは、個々のリソースの管理を実行する方法を理解していません。この知識は、各リソースおよびそれに対応する管理メッセージにカプセル化されます。

管理メッセージ

管理メッセージは、基礎となる MQSeries Everyplace メッセージ・オブジェクトを拡張します。表11 は、MQSeries Everyplace リソースの管理用に提供されているメッセージ・クラスのリストです。これらの基本管理メッセージをサブクラス化して、他のオブジェクトの管理のために提供できます。たとえば、MQeQueueAdminMsg のサブクラスを使用して、異なるタイプのキューを管理できます。MQSeries への MQSeries Everyplace ブリッジは、このように MQAdminMsg のサブクラスを使用します。

表 11. 管理メッセージ・クラス

管理メッセージ・クラス	使用法
MQAdminMsg	すべての管理メッセージの基礎として使用される抽象クラス
MQeQueueManagerAdminMsg	キュー・マネージャーの管理
MQeQueueAdminMsg	ローカル・キューの管理
MQeRemoteQueueAdminMsg	リモート・キューの管理
MQAdminQueueAdminMsg	管理キューの管理
MQeHomeServerQueueAdminMsg	ホーム・サーバー・キューの管理
MQeStoreAndForwardQueueAdminMsg	ストア・アンド・フォワード・キューの管理
MQeConnectionAdminMsg	キュー・マネージャー間の接続の管理
MQeClientConnectionAdminMsg	MQS への接続に使用される、ブリッジ・クライアント接続オブジェクトの管理
MQeListenerAdminMsg	MQS からメッセージを収集するのに使用される、ブリッジ伝送キュー・リスナー・オブジェクトの管理
MQeBridgeAdminMsg	MQS へのブリッジの管理
MQeMQBridgesAdminMsg	MQ ブリッジのリストの管理
MQeMQMgrProxyAdminMsg	MQSeries キュー・マネージャーのブリッジの表現の管理
MQeMQBridgeQueueAdminMsg	MQ ブリッジ・キューの管理

8. MQSeries Everyplace Explorer は、バージョン 1.0 には含まれていませんが、ワールド・ワイド・ウェブ (WWW) の MQSeries ソフトウェア・ダウンロード・サイト (<http://www-4.ibm.com/software/ts/MQSeries/>) からダウンロードできます。

管理メッセージの構造は、その特定のクラス（つまり、管理するリソースの性質と、そのリソースに対して実行する操作の詳細）に応じて異なります。しかし、一般的には、管理メッセージは表12 で示されているような構造です。

表 12. 管理メッセージの一般構造

レベル 1 フィールド	レベル 2 以下のフィールド	使用法
Admin_Action		作成、削除、調査など
Admin_Errors		フィールド・オブジェクトの親
	複数のフィールド	エラーごとの詳細情報
Admin_MaxAttempts		管理処置の最大試行回数
Admin_Parameters		フィールド・オブジェクトの親
	リソース	管理されるリソースの名前
	複数フィールド	メッセージ・クラスと処置に特定の、詳細パラメーター・データ
Admin_Reason		失敗の理由を示すテキスト・メッセージ
Msg_ReplyToQ		応答の送信先のキューの名前
Msg_ReplyToQMgr		応答の送信先のキュー・マネージャーの名前
Admin_RC		結果を示す数値戻りコード
Msg_Style		コマンドまたは要求 / 応答
Admin_TargetQMgr		ターゲット・リソースを所有するキュー・マネージャーの名前

3 つのスタイルの管理メッセージがサポートされています。つまり、応答を必要としない管理処置を示すコマンド（データグラム）、応答を必要とする要求、および応答そのものです。応答は、オリジナル・メッセージのコピーから構成されます。したがって、付加的なフィールドは、受信側が使用できるように、送信側が追加します。

基本管理メッセージのサポートに加え、最も一般的な管理操作に、メッセージ構造と応答の変換処理をカプセル化するヘルパー・クラスも用意されています。これらのクラスは、オプションでユーザー・ダイアログを提供することができます。これにより、簡単な管理ツールを構築するのに、クラスを有用に用いることができるようになります。⁹

選択的な管理

管理へのアクセスは、管理キュー上にある認証機能を介して制御できます。ローカル・アプリケーションの場合、提供される認証機能は、それらすべての同じローカル・ユー

9. これらのクラスはバージョン 1.0 には含まれていませんが、ワールド・ワイド・ウェブ (WWW) の MQSeries ソフトウェア・ダウンロード・サイト (<http://www-4.ibm.com/software/ts/MQSeries/>) からダウンロードできます。

ザーを表すものと見なすため、それらすべてに管理を許可したり、拒否したりします。リモート管理アプリケーションは、管理メッセージの送信に先立って、チャンネル上の認証機能の起動によって制御されます。したがって、異なるリモート・ユーザーが区別でき、個別に使用可能にしたり、使用不可にしたりすることができます。どのユーザーの場合でも、管理は全体として使用可能にされたり使用不可にされたりします。さらに高度なレベルの管理制御が必要な場合、たとえば特定の管理ユーザーが、あるキューにはアクセスを許可され、他のキューには許可されないような場合、追加のプログラミングが必要です。より洗練された認証機能がユーザー識別に関連する許可を追跡し、続いて管理メッセージをこれらの許可ごとに処理することができます (セキュリティーのセクションを参照)。キューに関連したルールも、同様の方法で処置を許可したり拒否したりするために活用できます (43ページの『ルール』を参照)。

モニターと関連処置

管理はしばしば、オブジェクトの作成および変更以上に、たとえばシステムの状態のモニターやエラー状況の処理などに関係しています。たとえば、キューがまもなくいっぱいになることをオペレーターに通知したり、ターゲット・キューにとって大きすぎるメッセージが届いた場合に、適切な処置をとったりすることです。これらの局面は、ルールの使用によって MQSeries Everyplace で処理されます。ルールとは、オブジェクトの状況が大幅に変化したり、特定のタイプのエラー状況が発生したりした場合に呼び出されるクラスです。ルール・クラスのデフォルトのセットは MQSeries Everyplace に備わっていますが、通常はカスタム・クラスに置き換えられます (43ページの『ルール』を参照)。

動的チャンネル

MQSeries Everyplace は、デバイスとゲートウェイ (またはそのどちらか) のキュー・マネージャー間の通信を、動的チャンネルと呼ばれる論理リンクを介して行います。これらは、両方向フローをサポートし、必要に応じてキュー・マネージャーにより確立されます。非同期メッセージングも同期メッセージングも同じチャンネルを使用します。使用されるプロトコルは MQSeries Everyplace に対して固有なものです。対照的に、MQSeries は通常、同期トラフィックにはクライアント・チャンネルを使用し、両方向の非同期メッセージングにはメッセージ・チャンネルの対を使用します。MQSeries クラスター・メッセージ・チャンネルには、MQSeries Everyplace 動的チャンネルと類似した特性がありますが、相違点も多数あります。

動的チャンネルは 2 つのキュー・マネージャー間の論理接続で、データの送受信の目的で確立されます。同じパーティー間であっても、複数のチャンネルが並行して存在する場合があります。これらのチャンネルでは、認証、暗号、圧縮、移送プロトコルなどの特性が使用されます。これらの特性はプラグ可能で(異なるチャンネルでは、異なるバージョンが使用されることもある)、その結果各チャンネルごとに次のような独自のサービス品質属性があります。

- **認証機能:** NULL か、ユーザーまたはチャンネル認証を実行できる**認証機能** のどちらか
- **チャンネル:** 移送サービスを提供するクラス
- **圧縮機能:** NULL か、データ圧縮および解凍を実行できる**圧縮機能** のどちらか
- **暗号化機能:** NULL か、暗号化および暗号化解除を実行できる**暗号化機能** のどちらか
- **宛先:** このチャンネルのターゲット (SERVER.XYZ.COM など)

認証機能は、通常チャンネルの設定時のみに使用されます。圧縮機能と暗号化機能は、通常すべてのフローで使用されます。

最も単純なタイプの暗号化機能は MQeXorCryptor で、データの排他論理和を実行することにより、送信するデータを暗号化します。この暗号化は安全なものではありませんが、データを表示できなくします。対照的に、MQe3DESCryptor は triple-DES をインプリメントする最も単純なタイプの圧縮機能は MQeRleCompressor で、反復文字をカウントで置換することによってデータを圧縮します。他にも認証機能、圧縮機能、および暗号化機能が用意されています。36ページの表13 を参照してください。

チャンネル確立は、プロトコル・アダプター仕様を使用して、特定のチャンネルに使用されるリンクとプロトコルを判別します。各中間ノードでチャンネル定義が検索され、次のリンクに必要なアドレッシングが解決されます。前方向への定義がなくなった地点でチャンネルは終了します。その地点でメッセージの流れがあれば、そのメッセージはキュー・マネージャーに渡されます。

チャンネルは、アプリケーションまたは管理者から直接見ることはできません。必要に応じてキュー・マネージャーにより確立されます。チャンネルは、キュー・マネージャーをまとめてリンクします。その特性は、流される情報に従って、MQSeries Everyplace により交渉および再交渉されます。送達機能は、チャンネルを活用してキュー・レベルの通信を提供する MQSeries Everyplace コンポーネントです。送達機能も、アプリケーション・プログラマーや管理者には見えません。

確実なメッセージングが必要な場合、MQSeries Everyplace はメッセージをアプリケーションに一度限り送達します。これは、伝送の終了時にコピーを削除する前に、メッセージが正常にあるキュー・マネージャーから別のキュー・マネージャーに渡されたことと、肯定応答を確認することによって実現されます。通信障害が生じた場合、肯定応答が受け取れないと、メッセージは場合によっては再送されますが (一度限りの送達は、一度限りの伝送ということを暗に示しているわけではありません)、重複メッセージは送達されません。

アダプター

アダプターは、MQSeries Everyplace をデバイス接続インターフェースにマップするのに使用されます。チャンネルは、プロトコル・アダプターを活用して、HTTP、ネイティブ TCP/IP、および他のプロトコルを実行します。同様に、キューはフィールド・ストレージ・アダプターをインターフェースとして、メモリーまたはファイル・システムなどのストレージ・サブシステムに接続します。アダプターは MQSeries Everyplace に、デバイス・サポートを拡張するとともに、バージョン管理を行うためのメカニズムを提供します。

ファイル記述子は、アダプターを識別し、ロードし、アクティブにするのに使用される文字列です。

ダイヤルアップ接続の管理

デバイスのダイヤルアップ・ネットワーク・サポートは、デバイスのオペレーティング・システムの役割です。接続されていないデバイス上の MQSeries Everyplace で、たとえばメッセージの送信などで、ネットワークを使用する場合、ネットワーク・スタックがアクティブでないと、オペレーティング・システムそのものがリモート・アクセス・サービス (RAS) を開始します。一般的に、この場合には、パネルが表示されて、ダイヤルアップ接続プロファイルが示されます。接続を確立するまで、オペレーティング・システムによって制御されます。したがって、デバイスのユーザーは、使用するオペレーティング・システムに適したダイヤルアップ接続プロファイルを使えるようにする必要があります。ですから、MQSeries Everyplace のデバイスのインプリメンテーションには、ダイヤルアップ・ネットワークのための特別なサポートは必要ありません。

トレース

トレースは、トレース・アクションを行う独立したプログラムを実行することにより、利用することができます。MQSeries Everyplace には、システムおよびユーザーに応じて、通知状態、警告状態、およびエラー状態をトレースするための呼び出しが組み込まれています。アプリケーション側は、トレースを直接に呼び出すことにより、新しいメッセージを追加するか既存のトレース・メッセージを変更できます。提供されているサンプル・トレース・プログラムでは、選択したメッセージを表示したり、印刷したり、イベント・ログへ送ることができます。他のトレース・プログラムを作成するときには、別の機能を付加したり、出力を別の方法で整形したり送付するように設計することもできます。

ほとんどの MQSeries Everyplace 例外は、アプリケーションに渡されて処理されます。そして、これらはアプリケーション例外ハンドラーでトレースに送ることができます。

イベント・ログ

MQSeries Everyplace には、(たとえば、キュー・マネージャーを開始したときの) 状況をログに記録するときに使う、イベント・ログ・メカニズムおよびインターフェースが備えられています。ログの記録を開始して、デフォルトでファイルへ書き込むことができますが、その記録を代行受信して別のものに書き込むこともできます。MQSeries Everyplace イベント・ログは、メッセージ・データをログに記録しないので、メッセージまたはキューの回復に使うことはできません。

MQSeries Everyplace ネットワーク

MQSeries Everyplace ネットワークは、接続されたデバイスとゲートウェイで構成されます。これは、複数の物理ネットワークや、その間でメッセージをやり取りすることが関係する場合があります。一般には、キューの位置に左右されないプログラミング・モデルにより、キューへの同期および非同期アクセスが行

構成とスケーラビリティ

基本的な MQSeries Everyplace ネットワーク接続形態の例を、33ページの図5 に示します。そのため、それぞれの構成は同期通信機能と非同期通信機能の両方を備えているものとします。

一番簡単なケースは、スタンドアロンのデバイスが、(a) で示されているような、ローカル・キューを経由した同期のアプリケーション間通信をサポートするものです。それ以上の相関関係があるのが、ケース (b) であり、ここでは対等ネットワークが示されています。この場合、一方のデバイスがもう一方からの接続に応じられるように、両方のデバイスで同じ通信プロトコルを用いてやり取りし、少なくとも片方のデバイスに listen 機能が構成されている必要があります。当然のことながら、この簡単なケースでは、両方が同じネットワーク上で使用可能な場合に限り、通信が可能です。非同期メッセージングを使うと、デバイスが接続されていないときに、アプリケーションを実行することができますが、同期メッセージングでは、デバイスが実際に接続されている場合にだけそれが可能です。

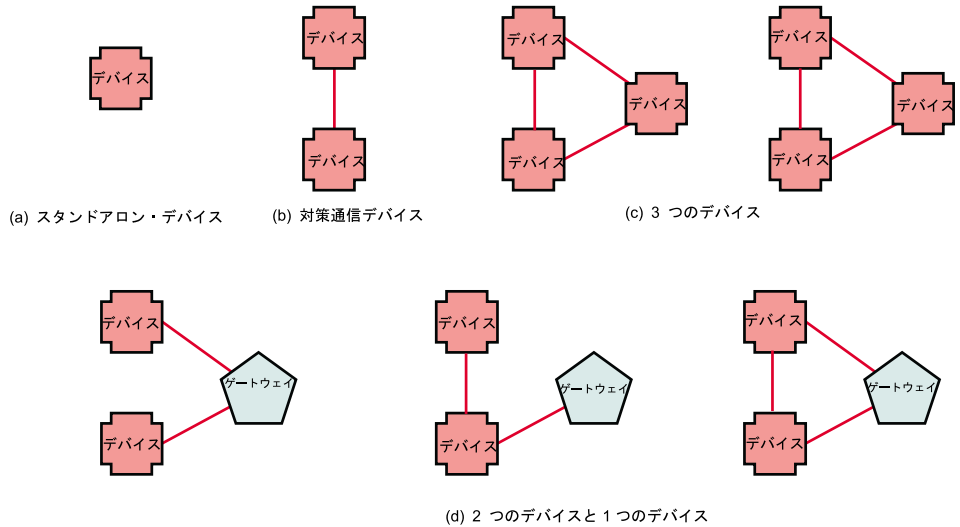


図5. 簡単な MQSeries Everyplace ネットワーク

3つのデバイスを使うと、さらに多くのことが可能です。いずれかのデバイスがリンク・デバイスの役割をすることもできますし、3つすべてを相互接続させることもできます。対等通信の形式に基づいて相互にメッセージを交換するよう構成されている場合、上記の考慮事項が当てはまります。3番目のデバイスを経由して通信するのであれば、さらに柔軟な構成が可能になります。2つの通信デバイスは同じ通信プロトコルでやり取りする必要はありません。それぞれは、ブリッジとなるデバイスに通じる通信プロトコルを使えばよいわけです。さらに、非同期による接続を行うと、送信側と受信側が、ネットワークへ同時に接続しなくてもメッセージを流せます（ただし、リンク・デバイスが使用可能なときに、それぞれが別々にネットワークに接続することが必要）。同期通信を行う場合には、3つのデバイスすべてが同時に使用可能になっていなければなりません。しばしば切断されるデバイスの場合、何らかの3番目のノードを経由した構成が非常に適しています。ただし、中間のデバイスはたいてい使用可能になっている必要があります。

実際には、図5の(d)に示されている3つの例のように、デバイスはゲートウェイによってリンクされるのが普通です。リンク・ノードとしてゲートウェイを使う方が望ましいのは、ゲートウェイが、複数の並行した着信接続要求をサポートできるためです。示されている最初の構成は最も一般的なもので、2番目と3番目の構成は、異常が生じた場合に適したものです。3番目の構成では、特定のリモート・キュー・マネージャーに向けて、1つの経路だけしか構成されていないので、2つの経路があったとしても、1つしか選択されないことに注意してください。

大規模ネットワークでは、たくさんのゲートウェイを使うことができ、それぞれのゲートウェイでたくさんのデバイスをサポートします。ゲートウェイは任意の方法で相互接続できますが、完全な相互接続が定義されている場合、デバイス間の経路に3つ以上の

ゲートウェイが関係することはありません。図6には、大規模ネットワークの例が示されています。

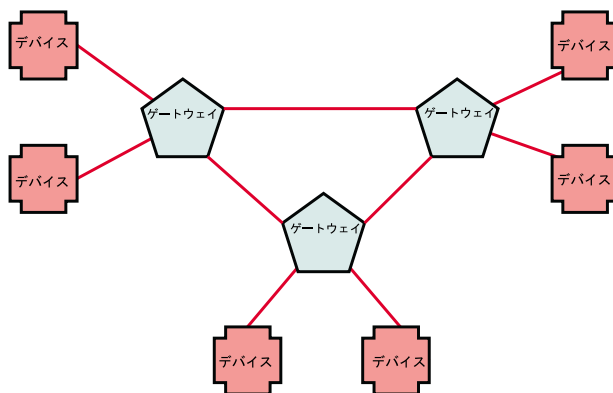


図6. スター型 MQSeries Everyplace ネットワーク

非同期メッセージ送達

メッセージを非同期でリモート・キューに書き込む場合、メッセージ・オブジェクトは、論理的には支援記憶装置に置かれます。この支援記憶装置については、宛先キュー・マネージャーやキュー名が示されたそのキューのローカル定義、そしてメッセージの宛先と一致する圧縮、認証、暗号化特性と関連付けられています。オブジェクトを安全な形式で永続的ストレージへ保管するときには、宛先キューで定義されているように、オブジェクトの `dump` メソッドが呼び出されます。キュー・マネージャーは、メッセージの送達を制御します。ネクスト・ホップのキュー・マネージャーへの適切な特性を持つチャンネルが識別 (または設定) され、宛先キューへの送達機能が作成 (または再利用) されます。この送達機能は、オブジェクトをダンプし、結果のバイト・ストリングを伝送します。宛先キュー・マネージャーとキュー名は、そのメッセージの流れの一部ではないことに注意してください。

適切であれば、メッセージはチャンネルの中で暗号化され圧縮されます。メッセージが宛先キュー・マネージャーに到達したら、暗号化解除されて解凍されます。新しいメッセージ・オブジェクトは、そのオブジェクト・クラスの `restore` メソッドで作成されます。生成されたオブジェクトは、宛先キューに置かれます。メッセージが宛先キュー・マネージャーに到達していない場合、暗号化解除されて解凍された後、前方伝送のための適切な特性を持つストア・アンド・フォワード・キューに置かれます。どちらの場合も、それぞれの宛先キューによって定義されているように、該当するキューに安全な形式で置かれます。

非同期メッセージ送達の特徴は、メッセージが中間ホップのキュー・マネージャーに渡され、前方伝送のためにキューに入れられることです。メッセージは、まず優先順位に従って中間キューから取り出され、続いてタイム・スタンプ順に取り出されます。

同期メッセージ送達

同期のメッセージ送達は、前述の非同期の場合と似ていますが、中間ホップでキュー・マネージャーが関係する比率ははるかに少なく、送達機能とチャンネルの関係する割合が多くなります。チャンネルは、各中間ノードのプロトコル仕様で定義されたアダプターを使用して、終端間で確立され、次のリンクを識別します。関係のあるファイル記述子が存在しない最後のリンクの終端では、メッセージは、高位層のキュー・マネージャーに渡されて処理されます。したがって、送信側ノードはメッセージをキューに入れず、中間ホップ経由でチャンネルを通じて送ります。そしてそのメッセージを宛先キュー・マネージャーに渡し、宛先キューに入れます。

MQSeries へのリンクでは、ゲートウェイでブリッジ・キューを使います。ここで、メッセージは MQSeries 形式に変換されます。このメカニズムでは、ゲートウェイで終了する動的チャンネルを使うことにより、あるデバイスから MQSeries への MQSeries Everyplace スタイルの同期メッセージングが可能です。メッセージは、クライアント・チャンネルを経由して、ゲートウェイから MQSeries サーバーヘリアルタイムで送達されます。サーバーからは、MQSeries メッセージ・チャンネルを利用し、宛先へ非同期で送達することも可能です。

適切なメディアが使用可能であれば、同期メッセージングだけが可能なデバイスでも、同じやり方でメッセージを MQSeries Everyplace キューへ非同期に送ることができます。

セキュリティ

MQSeries Everyplace には、統合された一群のセキュリティ機能が備えられており、メッセージ・データを、ローカルに保管するときにも転送するときにも、保護することができます。

MQSeries Everyplace セキュリティ機能に備えられている保護は、次の 3 つのカテゴリに分けることができます。

- ローカル・セキュリティ - メッセージ (およびその他の) データのローカルな保護
- キュー・ベースのセキュリティ - 開始するキュー・マネージャーと宛先キューとの間でのメッセージの保護
- メッセージ・レベルのセキュリティ - 開始側と宛先との間でのメッセージのメッセージ・レベルの保護

MQSeries Everyplace のローカルおよびメッセージ・レベルのセキュリティーは、MQSeries Everyplace によって内部的に使われますが、MQSeries Everyplace アプリケーションでも使えます。MQSeries Everyplace キュー・ベースのセキュリティーは、内部サービスです。

MQSeries Everyplace セキュリティー機能の 3 つのカテゴリーはすべて、属性 (MQeAttribute またはそれより下のもの) を使うことにより、メッセージ・データを保護します。属性は、カテゴリーに応じて、明示的か暗黙的に適用されます。

各属性には、以下のオブジェクトが含まれます。

- 認証機能
- 暗号化機能
- 圧縮機能
- キー
- 宛先エンティティ名

これらのオブジェクトの使われ方は、MQSeries Everyplace セキュリティー機能のカテゴリーに応じて異なりますが、どの場合でも、メッセージ・オブジェクトに付与された属性を呼び出すと、MQSeries Everyplace セキュリティー機能による保護が行われます。これは、MQSeries Everyplace メッセージの 'dump' メソッドを呼び出したときに生じます (たとえば、属性の 'encodeData' メソッドを使ってメッセージ・データを暗号化して圧縮する場合)。MQSeries Everyplace メッセージの 'restore' メソッドが呼び出されると、MQSeries Everyplace セキュリティー機能による保護解除が行われます (たとえば、属性の 'decodeData' メソッドを使ってメッセージ・データを圧縮解除して暗号化解除する場合)。

認証、暗号化、および圧縮のために MQSeries Everyplace バージョン 1.0 でサポートされているアルゴリズムについては、表13 に示されています。

表 13. 認証、暗号化、および圧縮のサポート

機能	アルゴリズム
認証	Mini Certificate ベース (WAP フォーラム WTLS Mini Certificate による)
	検証 Windows NT/2000 による識別
圧縮	LZW
	RLE
暗号化	Triple-DES
	DES
	MARS
	RC4
	RC6
	XOR

MQSeries Everyplace のローカル・セキュリティ

ローカル・セキュリティは、MQSeries Everyplace メッセージ (または MQeFields か、MQeFields より下のもの) データを保護します。これを実現するためには、適切で対称的な暗号化機能と圧縮機能を使って属性を作成し、(パスワードあるいはパスワードを入力することにより) 適切な「鍵」を作成して設定して、その鍵を属性へ明示的に付加し、その属性を MQSeries Everyplace メッセージに付加します。MQSeries Everyplace には、ローカル・セキュリティの設定を援助する MQeLocalSecure クラスが備えられていますが、どの場合にも、適切な属性を設定して、パスワードまたはパスワードのキーを管理することは、ローカル・セキュリティ・ユーザー (MQSeries Everyplace 内部または MQSeries Everyplace アプリケーション) の役割です。

MQSeries Everyplace キュー・ベースのセキュリティ

キュー・ベースのセキュリティは、同期および非同期メッセージに適用できます。

同期でのキュー・ベースのセキュリティ

同期でのキュー・ベースのセキュリティを使うことにより、アプリケーション側は、メッセージ・セキュリティについての考慮事項をすべて MQSeries Everyplace に任せることができます。キューには、認証、暗号化、および圧縮の特性があり、これらを使うことにより、メッセージの流れを保護するのに必要なセキュリティのレベルを判別します (永続的ストレージのセキュリティのレベルも)。

メッセージを送信する場合、宛先キューのセキュリティ特性は、ローカル・レジストリーから検索されます。そこに存在しなければ、キュー・マネージャーは、宛先キュー・マネージャーで宛先特性を見つけようとし、それ以降に再利用するときのために、キャッシュに入れておきます。そのキュー・マネージャーへのチャンネルがあれば使われますし、なければ新しいチャンネルが作成されます。宛先キューの属性が検索されます。

必要なサービス品質に基づき、宛先キュー・マネージャーへのチャンネル属性は、動的に変わります。この場合、確立されたルールがあれば、それに従います。一般に、ルールによってセキュリティのレベルでのアップグレードが可能になります (たとえば、保護なしから弱い保護、あるいは弱い保護から強力な保護へのアップグレードなど)。チャンネルをアップグレードできないか、セキュリティ・レベルが過度に思える場合 (たとえば、保護が必要ないのに使用可能なチャンネルで強力な保護を行っているなど)、新しいチャンネルが作成されます。チャンネルのプールが存在し、可能な場合に再利用されますが、トラフィックの要求に応じて、特性が動的に変わります。チャンネルは、必要ない場合には、自動的に破棄されます。メッセージは、常に宛先キューの特性で定義されたセキュリティ・レベルで、キューに入れられます。

認証はチャンネル・レベルで行われるので、メッセージ当たりのオーバーヘッドは最小に抑えられます。さらに、同期でのキュー・ベースのセキュリティは、一般的に、対称的な暗号化機能と共に使われます。そのようにすると、暗号機能が高速になります。し

かし、そのような対称的なケースでは、MQSeries Everyplace はまず RSA 非対称暗号化機能を使い、送信側と受信側での共用鍵を設定するときに必要なデータの流れを保護します。その後で対称的な暗号化機能を使い、流されるデータの秘密を保護します。

MQSeries Everyplace では、チャンネルの流れごとに鍵が動的に変わるので、このデータの暗号攻撃はさらに困難になります。さらに MQSeries Everyplace では、送信前にダイジェストを生成してデータへ付加し、受け取ったらすぐにそのダイジェストを再生成および妥当性検査することにより、流されるデータの整合性を確認します。

非同期でのキュー・ベースのセキュリティ

非同期メッセージングは、`putMessage` の実行時に宛先キューへのアクセスが必ずしも保証されないという点で、前述の同期メッセージングと異なっています。この場合、キュー・マネージャーはメッセージをすぐに送信できず、伝送キューへ入れることとなります。このときには、宛先キューの特性に応じて暗号化されます。伝送できるようになると、暗号化解除されて、適切な特性を備えたチャンネルを通じて送信されます。したがって、メッセージは常に保護されており、伝送を待機しているときも例外ではありません。非同期メッセージングには、リモート・キュー定義が必要です。これがないと、宛先キューの特性を判別できません。

非同期のケースでは、発信元と宛先との間の認証は行えません。認証が重要な場面では（たとえば、受信側にとっては、メッセージの発信元を判別し、受け入れるかどうかを決定する、あるいは拒否しないことを設定する場面、そして、発信元にとっては、メッセージが意図した宛先だけに扱われていることを確認する場面）、メッセージ・レベルのセキュリティを使う必要があります。

キュー・ベースのセキュリティとメッセージ・レベルのセキュリティを同時に使うことは可能ですが、メッセージ・データはすでに保護されているので、その必要性はありません。

メッセージ・レベルのセキュリティ

メッセージ・レベルのセキュリティでは、発信側と受信側の MQSeries Everyplace アプリケーション間で、メッセージ・データを保護します。

メッセージ・レベルのセキュリティは、アプリケーション層サービスであり、発信側の MQSeries Everyplace アプリケーションでメッセージ・レベルの属性を設定し、`putMessage` を使ってメッセージを宛先キューに入れるときに、その属性を使う必要があります。受信側アプリケーションでは、アプリケーションが `getMessage` を起動し、宛先キューからメッセージを取得するときにその属性を使えるように、一致するメッセージ・レベルの属性を設定し、受信側のキュー・マネージャーへ渡さなければなりません。

ローカル・セキュリティの場合と同様に、メッセージ・レベルのセキュリティでは、メッセージ・オブジェクトの属性のアプリケーションを活用します。発信側のアプリケーションのキュー・マネージャーは、`'dump'` メソッドで `putMessage` を処理しま

す。そして、属性の 'encodeData' メソッドでメッセージ・データを保護します。受信側のアプリケーションのキュー・マネージャーは、'restore' メソッドでアプリケーションの getMessage を処理します。そして、属性の 'decodeData' メソッドで元のメッセージ・データを復元します。

MQSeries Everyplace には、メッセージ・レベルのセキュリティのために、以下の 2 つの代替属性が備えられています。

MQeMAttribute

これは B to B (企業間) 通信で使われます。ここでは、相互の信用がアプリケーション層で厳しく管理されているので、信用された第三者による介入は必要ありません。利用できる MQSeries Everyplace のすべての対称的な暗号化機能および圧縮機能は、どれでも選択して使うことができます。ローカル・セキュリティの場合と同様で、属性の鍵が putMessage または getMessage へ渡される前に、その鍵を事前設定しておく必要があります。MQeAttribute には、メッセージ・レベルの保護のために、簡単かつ強力なメソッドが備えられています。これにより、公開鍵 PKI のオーバーヘッドを感じることなく、強力な暗号化機能を使ってメッセージの秘密を保護することができます。

MQeMTrustAttribute

この属性には、デジタル署名を使い、デフォルトの公開鍵 PKI を活用した、さらに高機能なソリューションが用意されています。これは、ISO9796 デジタル署名 / 妥当性検査を使い、受信側アプリケーションで、示された送信側からメッセージが着信したことを証明できるようにします。提供された属性の暗号化機能を使い、メッセージの秘密を保護します。SHA1 ダイジェストはメッセージの整合性を保ち、RSA 暗号機能は、意図した宛先によってのみメッセージを復元できるようにします。MQeMAttribute の場合にそうであったように、利用できる MQSeries Everyplace のすべての対称的な暗号化機能および圧縮機能は、どれでも選択して使うことができます。サイズを最適化するために、使われる認証は、WAP フォーラム WTLS の仕様で提案されている、WTLS Certificate に基づいたミニ認証です。MQSeries Everyplace のデフォルトのインフラストラクチャーでは、認証 (シグニチャーの妥当性検査) に必要な情報と、暗号化 / 暗号解除に必要な情報は、相互に使うことができます。

一般的な MQeMTrustAttribute 保護メッセージは、次のような形式になります。

RSA-enc{SymKey}, SymKey-enc {Data, DataDigest, DataSignature}

それぞれは以下のような意味になります。

RSA-enc:	意図した受信側の公開鍵によって RSA 暗号化されている
SymKey	疑似乱数による対称キーが生成されている
SymKey-enc	SymKey で対称的に暗号化されている
Data	メッセージ・データ

DataDigest メッセージ・データのダイジェスト

DigSignature メッセージ・データに対する発信側のデジタル署名

メッセージ・レベルのセキュリティは、キュー・レベルのセキュリティには従属していません。

レジストリー

レジストリーは、キュー・マネージャーに関連した情報のための、基本的なストレージです。これは、キュー・マネージャーごとに存在します。それぞれのキュー・マネージャーは、レジストリーを使って、以下のものを保管します。

- キュー・マネージャーの構成データ
- キュー定義
- リモート・キュー定義
- リモート・キュー・マネージャー定義
- ユーザー・データ (構成に応じたセキュリティ情報を含む)

レジストリーへのアクセスは、通常は、正当なキュー・マネージャーのユーザーに限定されていて、PIN に基づいて保護されていますが、構成可能なオプションを使うと、セキュリティよりもフットプリントのサイズが重要なユーザーは、これを迂回できます。

MQSeries Everyplace の認証可能なエンティティー

相互の認証に基づくミニ認証を使うキュー・ベースのセキュリティと、デジタル署名を使うメッセージ・レベルの保護では、「認証可能なエンティティー」という概念が提示されています。相互の認証というと、一般に 2 人のユーザー (複数のユーザー) 間での認証を思い浮かべますが、実際は、メッセージングにはユーザーという概念はありません。普通は、この概念はアプリケーション・レベルで、すなわち、メッセージング・サービスのユーザーによって管理されます。MQSeries Everyplace では、ユーザーから「認証可能なエンティティー」に至るまで、「認証の宛先」という概念を慎重に抽象化しています。このことは、認証可能なエンティティーが人間である可能性を除外するものではありませんが、アプリケーションによって選択されて割り当てられることとなります。内部的には、MQSeries Everyplace はすべてのキュー・マネージャーを定義します。これらは、認証可能なエンティティーとして、発信側の可能性もありますし、ミニ認証に従属するサービスの宛先の可能性もあります。さらに、MQSeries Everyplace では、認証機能に基づいてミニ認証を使うよう定義されたキューを、認証可能なエンティティーとしても定義します。したがって、これらのサービスをサポートするキュー・マネージャーは、1 つの認証可能なエンティティー (キュー・マネージャー) か、一群の認証可能なエンティティー (キュー・マネージャーと、認証に記された認証機能を使うそれぞれのキュー) を持つことができます。

私用レジストリーと信任状

各認証可能なエンティティーに、それぞれの信任状があると便利です。その場合、2 つのことに取り組む必要があります。1 つ目は、信用状を入手するためにどのように登録を実行するかということ、そして 2 つ目は、その信任状を安全な方法でどこに管理するかということです。これらの問題は、基礎となる暗号化技法よりも解決が難しいものでした。MQSeries Everyplace には、認証可能なエンティティーで自動登録を実行できるようにするデフォルトのサービス、認証可能なエンティティーの私用信用状を安全に管理できるようにする私用レジストリー (基本レジストリーの下位)、そして、一群の公開信用状を管理する公開レジストリー (やはり基本レジストリーの下位) が備えられています。私用レジストリーは、基本レジストリーに対して、いろいろな品質の保護または暗号化トークンを渡します。このようなトークンは、たとえば、ミニ認証のような公開オブジェクトの保護リポジトリーであったり、秘密鍵のような私用オブジェクトの保護リポジトリーであったりします。許可ユーザーだけが私用オブジェクトへアクセスできるメカニズムになっています。さらに、私用オブジェクトが私用レジストリーから抜けないようにするサービス (たとえば、デジタル署名、RSA 暗号化解除など) をサポートしています。共通のインターフェースが提供されると、基本となるデバイスのサポートが意識されることはなくなります。したがって、現在はローカル・ファイル・システムに限定されているサポートも、将来的に拡張されて移植可能なトークンに割り当てることができます。

自動登録

MQSeries Everyplace には、自動登録をサポートするデフォルトのサービスがあります。これらのサービスは、認証可能なエンティティーが構成されると、自動的に起動します。たとえば、キュー・マネージャーの開始時や、新しいキューの定義時などです。どちらのケースでも、登録が起動し、新しい信任状が作成されて、認証可能なエンティティーの私用レジストリーに格納されます。自動登録のステップには、新しい RSA キーのペアを生成すること、秘密鍵を私用レジストリーで保護し保管すること、そして、デフォルトのミニ認証サーバーに対する「新しい認証」要求に、公開鍵をパッケージすることが含まれます。ミニ認証サーバーが構成されて使用可能であるとすると、そのミニ認証と共に、保護された秘密鍵が組み込まれた認証可能なエンティティーの新しいミニ認証が戻されます。それらのミニ認証は、新しい信用状として、認証可能なエンティティーの私用レジストリーに格納されます。自動登録は、認証可能なエンティティーの信用状を設定する簡単なメカニズムですが、メッセージ・レベルの保護 (MqeMTrustAttribute、前述の説明を参照) の場合は、意図した受信側の公開鍵 (ミニ認証) へアクセスする必要もあります。

公開レジストリーと認証の複写

MQSeries Everyplace には、MQSeries Everyplace コンポーネント間で、認証可能なエンティティーの公開認証 (ミニ認証) を共用できるようにする、デフォルトのサービスがあります。これらは、MqeMTrust ベースのメッセージ・レベルのセキュリティーの前提条件になります。MQSeries Everyplace 公開レジストリーは、ミニ認証へ公的にアクセスできるリポジトリーになっています。これは、携帯電話の個人電話番号登録簿サービ

スに似ています。異なる点は、電話番号の代わりに、もっとも頻繁に接触する認証可能なエンティティのミニ認証群が使われる点です。ただし、公開レジストリーは、サービスの中で完全に受動的なわけではありません。アクセスされた後に、保持していないミニ認証を要求される場合で、有効なホーム・サーバー・コンポーネントで構成されている場合には、公開レジストリーは、自動的に、ホーム・サーバーの公開レジストリーから、要求されたミニ認証を入手しようとします。このようなサービスを利用して、自動化されたインテリジェントなミニ認証複写サービスが提供されます。これにより、必要なときに必要なミニ認証を使うことができます。

レジストリー・サービスのアプリケーションの使用

MQSeries Everyplace キュー・マネージャーは、私用および公開レジストリー・サービスを最大限に活用するよう設計されていますが、これらのサービスへのアクセスは制限されているわけではありません。MQSeries Everyplace ソリューションでは、それぞれの認証可能なエンティティ、たとえば、ユーザーを定義して管理することができます。私用レジストリー・サービスは、新しい認証可能なエンティティの信用状を自動登録して管理するときに、そして公開レジストリー・サービスは、必要な場合に、公開信用状を使用できるようにするときに使えます。登録された認証可能なエンティティはすべて、MQeMTrustAttribute で保護されたメッセージ・レベルのサービスの発信側としても、また受信側としても使うことができます。

デフォルトのミニ認証発行サービス

MQSeries Everyplace には、デフォルトのミニ認証発行サービスがあります。これは、私用レジストリー自動登録要求を満たすように構成できます。MQSeries Everyplace に付属するツールを使うことにより、特定のソリューションで、ミニ認証発行サービスを設定して管理し、注意深く制御された一群のエンティティ名にミニ認証を発行できます。この発行サービスには、以下のような特性があります。

- 登録された認証可能なエンティティ群を管理する
- ミニ認証を発行する (WAP WTLS ミニ認証に基づくミニ認証)
- ミニ認証のリポジトリを管理する

MQSeries Everyplace に付属するツールを使うことにより、ミニ認証発行サービスの管理者は、エンティティ名と登録アドレスを登録し、一時使用の認証要求 PIN を定義することにより、指定されたエンティティへのミニ認証発行を許可することができます。これは、通常は、オフライン検査で要求側の認証性が妥当性検査されてから行われます。認証要求 PIN は、意図したユーザーにポストされます (たとえば、新しいキャッシュ・カードが発行されるときに、キャッシュ・カード PIN がポストされるのと同様の方法)。次に私用レジストリーのユーザー (たとえば、MQSeries Everyplace アプリケーションや MQSeries Everyplace キュー・マネージャー) を構成し、起動時にこの認証要求 PIN を送るようにすることができます。私用レジストリーが自動登録を起動すると、ミニ認証発行サービスでは、生じる新しい認証要求を (示されるエンティティ名と、値が事前登録された認証要求 PIN との突き合わせに基づき) 妥当性検査し、新しい

ミニ認証を発行し、登録された認証要求 PIN をリセットして再利用できないようにします。自動登録される新しいミニ認証要求はすべて、安全なチャネル経由で処理されません。

ミニ認証発行サービスで発行される一連のミニ認証は、発行サービスのレジストリーに保管されます。ミニ認証が再発行されると (たとえば、有効期限切れなどで)、有効期限が切れたミニ認証は保存されます。

セキュリティ・インターフェース

オプションのインターフェースが用意されており、カスタムのセキュリティ・マネージャーでインプリメントすることができます。この方式を使うと、セキュリティ・マネージャーは、以下と関連した要求を許可したり拒否できます。

- クラス別名の追加または除去
- アダプターの定義
- ファイル記述子の割り当て
- チャネル・コマンドの処理

構成とカスタマイズ

ルール

ルールは、いろいろな状態変更が生じるときに、MQSeries Everyplace の動作をカスタマイズするために使われる Java クラスです。必要な個所にデフォルト・ルールが用意されていますが、顧客の要件を満たすために、アプリケーションまたはインストール固有のルールで置き換えられることがあります。サポートされるルール・タイプは、その機能ではなく、起動される方法によって異なってきます。ルールには論理が含まれているため、広範囲な機能を実行することができます。

属性のルール

このルール・クラスには、状態の変更が試みられるとき (たとえば、以下の変更) に、制御権が与えられます。

- 認証機能
- 圧縮機能
- 暗号機能

変更は、ルールによって、許可されたりされなかったりすることが一般的です。

MQSeries ブリッジのルール

このルール・クラスには、MQSeries ブリッジ・コードへの MQSeries Everyplace に状態の変更がある場合に、制御権が与えられます。以下のそれぞれを決定するために、個別のブリッジ・ルール・クラスが準備されています。

- リスナーがメッセージを MQSeries へ送達できない場合で、メッセージがその MQSeries から送られてくる場合に、そのメッセージをどうするか。たとえば、メッセージが大きすぎるか、キューが存在しない可能性があります。
- サーバーをインスタンス化した後で、ブリッジで管理されたオブジェクトをどの状態で開始するか。
- ブリッジが MQSeries 上の Sync Q (クラッシュ・リカバリーに使う永続的ストレージ) に、何らかのエラーを検出した場合にどうするか。デフォルト・ルールは、問題を表示するだけです。
- MQSeries Everyplace メッセージを MQ メッセージに変換する方法 (およびその逆)。MQSeries Everyplace および MQ メッセージ間でメッセージ変換を行う変換機能は、MQeRule クラスから派生することはありません。その代わりに、MQeTransformerInterface インターフェースをインプリメントする必要があります。このこととは別に、変換機能はルールのような機能を持ち、メッセージで形式変換が必要なときに呼び出されます。

RAS ダイアラーのルール

このルール・クラスには、RAS ダイアラーで状態変更が行われるときに、制御権が与えられます。たとえば、以下のような場合があります。

- 呼び出した番号で接続されない場合
- エラーしきい値を超過する場合
- ダイヤルしてみても、時刻に基づく特定タイプの接続しか使われない場合。たとえば、オフピークのときにしか、電話が使えないなど。

キューのルール

このルール・クラスには、関連するキューの状態変更が行われるときに、制御権が与えられます。たとえば、以下のような場合があります。

- メッセージをキューに追加する場合。たとえば、しきい値 (メッセージ数、メッセージのサイズ、無効な優先順位) を超過していないかを確認する場合があります。
- キューの特性を割り当てたり変更する場合
- キューを開いたり閉じたり場合
- キューを削除する場合

キュー・マネージャーのルール

このルール・クラスには、キュー・マネージャーの状態変更が行われるときに、制御権が与えられます。たとえば、以下のような場合があります。

- キュー・マネージャーが開く場合。たとえば、バックグラウンド・タイマー・スレッドの実行を開始し、時間を指定したアクションが生じるようにする場合があります。
- キュー・マネージャーが閉じる場合。たとえば、バックグラウンド・タイマー・スレッドを終了する場合があります。
- 新しいキューを追加する場合

接続スタイル

MQSeries Everyplace では、クライアント / サーバー¹⁰ または対等通信 (あるいはその両方) による運用がサポートされています。クライアント は、サーバーとの通信を開始できますが、一方で、サーバー は、クライアントによって開始された要求に応じることができるだけです。対等通信 での運用では、2 つの対等機能がどちらの方向でも流れを開始できます。このような接続スタイルでは、MQSeries Everyplace の使用可能でアクティブなコンポーネントが別途必要です。関係するコンポーネントは、以下のとおりです。

- **チャンネル・リスナー:** 着信接続要求を listen します。
- **チャンネル・マネージャー:** エンドポイント間での、論理的な複数の並行通信パイプをサポートします。
- **キュー・マネージャー:** メッセージングおよびキュー機能を提供することにより、アプリケーションをサポートします。

表14 には、それぞれのコンポーネントと、接続スタイルとの相関関係が示されています。クライアント / サーバーの接続スタイルは、MQSeries Everyplace をクライアント・モードでもサーバー・モードでも運用できる状態を示しています。サブレット・オプションは、MQSeries Everyplace が HTTP サーブレットとして構成され、HTTP サーバーそのものは着信接続要求を listen する状態を示しています。

表 14. 接続スタイル

	キュー・マネージャー	チャンネル・マネージャー	チャンネル・リスナー
クライアント	はい		
クライアント / サーバー	はい	はい	はい
対等機能	はい		
サーバー	はい	はい	はい
サブレット	はい	はい	

MQSeries Everyplace チャンネル・マネージャーを使うのか、あるいは MQSeries Everyplace チャンネル・リスナーを使うのかによって、ライセンスが異なり、MQSeries Everyplace インスタンスがゲートウェイになるのかどうかが決まります。

MQSeries Everyplace アプリケーションは、キュー・マネージャーによって使われる接続スタイルを直接に認識するわけではありません。しかし、接続している集団でどのリソースが使えるのか、他のキュー・マネージャーに接続できるのはどのキュー・マネー

10. ここでは、「クライアント」と「サーバー」という用語は、一般的な使われ方であり、MQSeries での意味とは異なります。

ャーか、デバイスまたはゲートウェイでの MQSeries Everyplace フットプリントは何か、そして、どの接続を並行して存在させられるのかに影響を与えることなので重要です。

対等通信接続

対等通信チャンネルでは、1 つのチャンネルについて、チャンネル・マネージャーとチャンネル・リスナーの機能が備えられています。2 つのキュー・マネージャー間で対等通信チャンネルを作成すると、片方のキュー・マネージャーはリスナーとして機能し、もう一方は、接続の開始側として機能します。対等通信接続されたキュー・マネージャーは、他のキュー・マネージャーに対する複数の対等通信接続を開始できますが、応答できるのは、1 つの着信接続要求だけです。したがって、別の要求に応答する前に、その対等通信チャンネルが閉じるのを待つ必要があります。任意の 1 つの対等通信チャンネルを使うことにより、2 つの参加キュー・マネージャーは、どちらもアクションを開始できます。たとえば、それぞれのキュー・マネージャーのアプリケーションは、相手方のキューにアクセスすることができます。

着信接続要求の宛先がファイアウォールに受け入れられない可能性があるため、ファイアウォール経由で対等通信チャンネルを使えないことがあります。

クライアント / サーバー接続

クライアント / サーバー接続スタイルで使われる標準チャンネルには、listen 機能がなく、サーバーの独立した listen 機能に依存しているため、サーバー側に、複数の並行チャンネルを扱うチャンネル・マネージャーが必要です。クライアントは接続要求を開始し、サーバーはそれに応答します。一般的に、1 つのサーバーで、クライアントからの複数の着信要求を処理できます。クライアントは、標準チャンネルを経由して、サーバーのリソースにアクセスします。サーバーのアプリケーションで、クライアントのリソースへ同期アクセスする必要がある場合、役割が逆転した 2 番目のチャンネルが必要になります。ただし、標準チャンネルそのものは両方向なので、相手方サーバーの伝送キューからクライアントへ宛てたメッセージは、クライアントが開始した標準 (クライアント / サーバー) チャンネル経由で、クライアントへ送られます。

クライアントは、複数のサーバーに同時に対応するクライアントになることがあります。チャンネル・マネージャーは複数のインバウンド・チャンネルを扱うので、この構成をサポートするためのチャンネル・マネージャーは必要ないことに注意してください。

着信接続の宛先は、通常はファイアウォールに受け入れられるものとして識別されるため、クライアント / サーバー接続スタイルは、一般に、ファイアウォールを経由して使用する場合に適しています。

複数接続スタイル

1 つのキュー・マネージャーには、対等通信かクライアント / サーバー接続のどちらかを開始する機能、そして、サーバーか対等機能のどちらかで応答する機能があります。この場合、対等機能チャンネル・リスナーと標準チャンネル・リスナーには、異なるポート番号が割り振られます。

クラス

MQSeries Everyplace では、特定の機能向けにクラスを選択し、MQSeries Everyplace の動作をカスタマイズして、特定のアプリケーション要件を満たすように設定できます。クラスによっては、別の方法を開発できるように、クラスへのインターフェースについて言及されています。表15 には、発展性についてまとめられています。クラスは、明示的に識別することもできますし、別名で識別することもできます。

表 15. クラスのオプション

クラス	別の方法の提供	インターフェースについての言及
administration	なし	あり
Authenticators	あり	なし
Communications adapter	あり	あり
Communications style	あり	なし
Compressors	あり	なし
Cryptors	あり	なし
Event log	サンプルを提供	あり
Messages	なし	あり
Queue storage	あり	なし
Rules	デフォルト・クラスを提供	あり
Trace	サンプルを提供	あり

アプリケーションのロード

MQSeries Everyplace が、クライアント (または対等機能) として運用されるよう構成される場合、開始アプリケーションには、他のアプリケーションを JVM へロードする役割があります。このために、標準の Java 機能を使うか、MQSeries Everyplace の一部として組み込まれているクラス・ローダーを使うことができます。したがって、同じ JVM にある 1 つのキュー・マネージャーに対して、複数のアプリケーションを実行することができます。別の方法として、複数の JVM を使うこともできますが、この場合、それぞれが各自のキュー・マネージャーを持ち、それぞれが固有な名前を持つこととなります。

MQSeries Everyplace がサーバーとして構成される場合、MQSeries Everyplace そのものが、開始アプリケーションになります。MQSeries Everyplace ではプリロード・クラスのリストがサポートされており、キュー・マネージャーそのものがロードされる前に、これらのクラスが順番にロードされます。

第6章 MQSeries Everyplace ネットワークと MQSeries ネットワーク

MQSeries Everyplace ネットワークは、MQSeries サーバーまたはネットワークがなくても、スタンドアロンで存在することのできるものですが、実際には、MQSeries Everyplace は、既存の MQSeries インストール・システムを補うときに使われることがあり、その範囲を新しいプラットフォームやデバイスに拡張したり、キューまたはメッセージ・ベースのセキュリティや、同期メッセージングなどの拡張機能を提供するものとなります。MQSeries Everyplace アプリケーションの全体像を見ると、MQSeries キューおよびキュー・マネージャーは、単なる追加のリモート・キューおよびキュー・マネージャーと見なされることがあります。しかし、これらのキューは MQSeries Everyplace 動的チャンネルや MQSeries Everyplace キュー・マネージャー経由で直接にアクセスできるものではなく、MQSeries Everyplace ゲートウェイを関係させることが必要であるため、機能的な制限がたくさんあります。ゲートウェイは、MQSeries クライアント・チャンネルを経由して、メッセージを複数の MQSeries キュー・マネージャーへ直接または間接的に送信できます。接続が間接的である場合、メッセージは MQSeries クライアント・チャンネルを移動して、中間の MQSeries キュー・マネージャーへ到達し、そこから MQSeries メッセージ・チャンネルを移動して、宛先キュー・マネージャーへ到達します。

MQSeries Everyplace へあてた MQSeries アプリケーションからのメッセージは、一般に MQSeries Everyplace キュー・マネージャーおよびキューに向けられます。その場合、MQSeries の経路が定義され (リモート・キュー・マネージャー定義)、MQSeries Everyplace メッセージが特定の MQSeries 伝送キューに到達するようにされます。MQSeries チャンネルは、一般的に、伝送キュー向けに定義されていません。その代わり、MQSeries Everyplace ゲートウェイは、メッセージをこれらのキューから取り出し、確実に MQSeries Everyplace 宛先へ送られるようにします。使用する伝送キューの数 (すなわち経路数) は構成可能であり、送達するメッセージのボリュームを反映するよう設定するようにします。

MQSeries へのインターフェース

MQSeries Everyplace のアーキテクチャーでは、MQSeries Everyplace と他のメッセージ・システムとの間での、1 つ以上の任意指定ブリッジという概念がサポートされています。

MQSeries Everyplace バージョン 1.0 では、サポートされているブリッジは 1 つだけです。すなわち、MQSeries Everyplace と MQSeries ネットワークとの間のインターフェースとなる *MQSeries* ブリッジです。このブリッジでは、1 つ以上の MQSeries キュー・マネージャーのインターフェースとして、MQSeries Java クライアントを使います。これにより、メッセージを MQSeries Everyplace から MQSeries へ流したり、その逆が可能になります。MQSeries Everyplace バージョン 1 では、ゲートウェイごとに 1 つのブリッジを使い、それぞれを複数の *MQSeries* キュー・マネージャー・プロキシ

(すなわち、MQSeries キュー・マネージャーの定義) に関連付けることをお勧めします。キュー・マネージャー・プロキシ定義は、MQSeries Everyplace と通信する MQSeries キュー・マネージャーごとに必要です。これらの定義のそれぞれには、1 つ以上のクライアント接続サービスを関連付けることができます。それぞれのサービスは、1 つの MQSeries キュー・マネージャーへの接続を表します。これらの定義ごとに、キュー・マネージャーに対する異なる MQSeries サーバー接続を使ったり、ユーザー出口やポートなど、異なるプロパティの設定を使うこともできます。

1 つのゲートウェイ・クライアント接続サービスで、たくさんの リスナー を扱うことができます。その場合、ゲートウェイ・サービスを使って MQSeries キュー・マネージャーへ接続し、MQSeries から MQSeries Everyplace へメッセージを取り出します。1 つのリスナーは、接続を確立するときに 1 つのサービスだけを使うので、それぞれのリスナーが、MQSeries キュー・マネージャーにある 1 つの伝送キューに接続することになります。それぞれのリスナーは、親ゲートウェイ・キュー・マネージャー経由で、メッセージを 1 つの MQSeries 伝送キューから MQSeries Everyplace ネットワークの任意の位置に移動します。したがって、1 つのゲートウェイ・キュー・マネージャーを使い、複数の MQSeries メッセージ・ソースを MQSeries Everyplace ネットワークに集約できます。

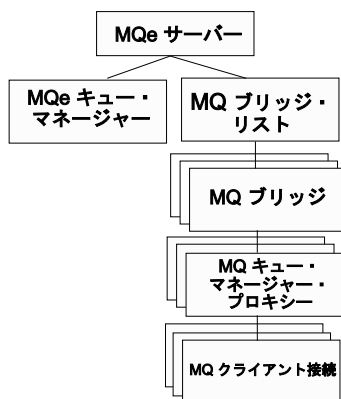


図7. MQSeries ブリッジのオブジェクト階層

メッセージを反対の方向に移動する場合 (MQSeries Everyplace から MQSeries)、ゲートウェイ・キュー・マネージャーは、1 つ以上のブリッジ・キュー オブジェクトを構成します。それぞれのブリッジ・キュー オブジェクトは、任意のキュー・マネージャーへ直接に接続し、メッセージを宛先キューに送信することができます。このように、1 つのゲートウェイを使うことにより、1 つの MQSeries Everyplace キュー・マネージャーを経由し、直接あるいは間接的に、MQSeries Everyplace メッセージを任意の MQSeries キュー・マネージャーへ送ることができます。ブリッジ・オブジェクトには、51ページの表16 に示されている特性があります。

表 16. ブリッジ・オブジェクトの特性

特性	説明
ブリッジ名	ブリッジのリスト
実行状態	状況: 実行中または停止状態

ブリッジ・オブジェクト、そして他のゲートウェイ・オブジェクトは、MQSeries Everyplace キュー・マネージャーには関係なく、開始および停止できます。そのようにしてゲートウェイ・オブジェクトを開始 (または停止) した場合、そのアクションは、そのすべての子 (すべてのブリッジ、キュー・マネージャー・プロキシ、クライアント接続、そして伝送キュー・リスナー) にも影響します。ブリッジ・オブジェクトには、表17 に示されている特性があります。

表 17. ブリッジの特性

特性	説明
クラス	ブリッジのクラス
デフォルト変換機能	他の変換機能クラスが宛先キューに関連付けられていない場合に、メッセージを MQSeries Everyplace から MQSeries へ (またはその逆) 変換するときに使われるデフォルトのクラス (ルール・クラス)。
ハートビート間隔	ブリッジ・オブジェクトに対してアクションを実行するときに使う、基本的な時間単位。
名前	ブリッジ・オブジェクトの名前
実行状態	状況: 実行中または停止状態
開始ルール・クラス	ブリッジ・オブジェクトの開始時に使われるルール・クラス
MQSeries キュー・マネージャー・プロキシの子	このブリッジによって所有されるすべてのキュー・マネージャー・プロキシのリスト

簡単なケースでは、すべてのメッセージ変換を処理するときに、デフォルトの変換機能 (ルール) を使うことができます。さらに、変換機能をリスナーごとに設定して (MQSeries から MQSeries Everyplace のメッセージの場合)、このデフォルト設定をオーバーライドすることができます。さらに個別に制御できるようにするために、ゲートウェイのブリッジ・キュー定義を使い、変換機能ルールを宛先キューごとに設定することも可能です。このことは、MQSeries Everyplace と MQSeries 宛先キューの両方に当てはまります。

MQSeries キュー・マネージャー・プロキシには、1 つの MQSeries キュー・マネージャーに固有の特性があります。プロキシの特性については、表18 に示されています。

表 18. MQSeries キュー・マネージャー・プロキシの特性

特性	説明
クラス	MQSeries キュー・マネージャー・プロキシのクラス

表 18. MQSeries キュー・マネージャー・プロキシの特性 (続き)

特性	説明
MQSeries ホスト名	Java クライアント・クラス経由で MQSeries キュー・マネージャーへの接続を作成するときを使う IP ホスト名。これを指定しない場合、MQSeries キュー・マネージャーは、ブリッジと同じマシンにあるものと見なされ、Java によるバインディングが使われる。
MQSeries キュー・マネージャー・プロキシ名	MQSeries キュー・マネージャーの名前
所有ブリッジの名前	この MQSeries キュー・マネージャー・プロキシを所有するブリッジ・オブジェクトの名前
実行状態	状況: 実行中または停止状態
開始ルール・クラス	MQSeries キュー・マネージャー・オブジェクトの開始時に使われるルール・クラス
クライアント接続の子	このプロキシによって所有される全クライアント接続オブジェクトのリスト

ブリッジ・クライアント接続サービスの定義には、MQSeries キュー・マネージャーへの接続するときに必要な詳細情報が記されています。接続の特性については、表19 に示されています。

表 19. クライアント接続サービスの特性

特性	説明
アダプター・クラス	ゲートウェイ・アダプターとして使うクラス
CCSID*	使用する整数 MQSeries CCSID 値
クラス	ブリッジ・クライアント接続サービスのクラス
最大接続アイドル時間	接続を終了せずにアイドルにしておける最大時間
MQSeries パスワード*	Java クライアントで使われるパスワード
MQSeries ポート*	Java クライアント・クラス経由で MQSeries キュー・マネージャーへの接続を作成するときを使う IP ポート番号。これを指定しない場合、MQSeries キュー・マネージャーは、ブリッジと同じマシンにあるものと見なされ、Java によるバインディングが使われる。
MQSeries 受信出口クラス*	クライアント・チャネルの他方の終端で使われる受信出口を突き合わせるときに使う。この出口には関連付けられたストリングがあり、データを出口コードに渡すことができる。
MQSeries セキュリティー出口クラス*	クライアント・チャネルの他方の終端で使われるセキュリティー出口を突き合わせるときに使う。この出口には関連付けられたストリングがあり、データを出口コードに渡すことができる。
MQSeries 送信出口クラス*	クライアント・チャネルの他方の終端で使われる送信出口を突き合わせるときに使う。この出口には関連付けられたストリングがあり、データを出口コードに渡すことができる。
MQSeries ユーザー ID*	Java クライアントで使われるユーザー ID

表 19. クライアント接続サービスの特性 (続き)

特性	説明
クライアント接続サービス名	MQSeries マシンでのサーバー接続チャンネルの名前
所有キュー・マネージャー・プロキシ名	所有するキュー・マネージャー・プロキシの名前
開始ルール・クラス	ブリッジ・クライアント接続サービス・オブジェクトの開始時に使われるルール・クラス
同期キュー名	同期をとるためにブリッジで使われる MQSeries キューの名前
同期キュー除去機能ルール・クラス	メッセージが同期キューにある場合に使われるルール・クラス
実行状態	状況: 実行中または停止状態
所有ブリッジの名前	このクライアント接続を所有するブリッジ・オブジェクトの名前
MQ XmitQ リスナーの子	このクライアント接続を使う全リスナー・オブジェクトのリスト
*これらのパラメーターの詳細については、 <i>MQSeries Using Java</i> の資料に掲載されています。	

アダプター・クラス は、MQSeries Everyplace から MQSeries へメッセージを送信するときに使われ、同期キュー は、この処理の状況を追跡するときに使われます。その内容は、リカバリー状態のときに、メッセージングを確実にを行うために使われます。通常シャットダウンの後には、キューは空になります。受信、送信、およびセキュリティー出口が同じであれば、複数のクライアント接続で、そして複数のブリッジ定義でこのキューを共有することができます。さらに、このキューは、使用中のリスナー特性に応じて、MQSeries から MQSeries Everyplace へ移動するメッセージの状態を保管するときにも使えます。同期キュー除去機能ルール・クラス は、メッセージが同期キューにあり、MQSeries Everyplace の障害が示されている場合に使われます。

最大接続アイドル時間は、MQSeries システムへのブリッジ・クライアント接続サービスで保守されている、Java クライアント接続のプールを制御するときに使われます。MQSeries 接続が使用されずアイドル状態になると、タイマーが開始され、接続が再利用される前にタイマーが切れると、そのアイドル接続は廃棄されます。MQSeries 接続を作成する操作は高くつくので、このプロセスを使うことにより、余分なりソースを使わずに、接続を効果的に再利用できるようになります。値がゼロの場合、接続プールを使ってはならないということです。

メッセージを MQSeries から MQSeries Everyplace へ移動するリスナー・オブジェクトには、表 20 に示されている特性があります。

表 20. リスナーの特性

特性	説明
クラス	リスナーのクラス

表 20. リスナーの特性 (続き)

特性	説明
送達不能キュー名	MQSeries から MQSeries Everyplace へ送達できないメッセージを保留しておくときに使うキュー
リスナー状態保管アダプター	状態情報を保管するときに使うアダプターのクラス名
リスナー名	メッセージを提供する MQSeries XMIT キューの名前
所有クライアント接続サービス名	クライアント接続サービス名
実行状態	状況: 実行中または停止状態
開始ルール・クラス	リスナー・オブジェクトの開始時に使われるルール・クラス
変換機能クラス	MQSeries メッセージの MQSeries Everyplace への変換を決定するときに使われるルール・クラス
未配布メッセージ・ルール・クラス	MQSeries から MQSeries Everyplace へのメッセージが送達されない場合のアクションを決定するルール・クラス
メッセージ待機秒数	例外環境でリスナーのパフォーマンスを制御するときに使える拡張オプション

未配布メッセージ・ルール・クラス は、MQSeries から MQSeries Everyplace へのメッセージを送達できない場合に、行われるアクションを決定します。これは、一般に MQSeries システムの送達不能キュー に置かれます。

メッセージを確実に送達するために、リスナー・クラスでは、リスナー状態保管アダプター を使い、MQSeries Everyplace システムか、MQSeries システムの同期キューに、状況情報を保管します。

ブリッジの構成を完成させるためには、リモート・キュー・マネージャーと、リモート・キュー定義の両方が必要です。リモート MQSeries Everyplace キュー・マネージャーのリモート・キュー・マネージャー定義は、標準の MQSeries Everyplace の決まりに従います。リモート MQSeries キュー・マネージャーの定義では、チャンネル定義がヌルに設定されています。このことは、通常の MQSeries Everyplace 動的チャンネルは使われないということを示しています。その代わりに、前述のように、特定の接続が MQSeries キュー・マネージャーに対して定義されます。

MQSeries Everyplace キューのリモート・キュー定義は、標準の決まりに従います。しかし、MQSeries キューの場合は、MQSeries Everyplace キューで使った定義とは著しく異なります。表21 には、MQSeries リモート・キューの特性が示されています。

表 21. MQSeries リモート・キューの特性

特性	説明
別名	キューの代替名
認証機能	ヌルにする必要がある

表 21. MQSeries リモート・キューの特性 (続き)

特性	説明
クラス	オブジェクト・クラス
クライアント接続	使うクライアント接続サービスの名前
圧縮機能	ヌルにする必要がある
暗号機能	ヌルにする必要がある
期限切れ	変換機能へ渡される
最大メッセージ・サイズ	ルール・クラスへ渡される
モード	同期にする必要がある
MQ キュー・マネージャー・プロキシ	メッセージを最初に送信する先の MQSeries キュー・マネージャーの名前
MQSeries ブリッジ	メッセージを MQSeries へ送るブリッジの名前
名前	リモート MQSeries キューが MQSeries Everyplace に認識される時の名前
所有キュー・マネージャー	定義を所有するキュー・マネージャー
優先順位	メッセージで使う優先順位 (メッセージ値でオーバーライドされない場合)
リモート MQSeries キュー名	リモート MQSeries キューの名前
ルール	キューの操作で使われるルール・クラス
キュー・マネージャー宛先	キューを所有する MQSeries キュー・マネージャー
変換機能	メッセージを MQSeries Everyplace 形式から MQSeries 形式に変換する変換機能クラスの名前
タイプ	MQSeries ブリッジ・キュー

暗号化機能、認証機能、および圧縮機能 クラスでは、一連のキュー属性が定義されています。これらの属性には、メッセージをこのキューに渡すときのセキュリティのレベルが示されています。メッセージは、MQSeries Everyplace 上でメッセージが最初に送信されることから、そのメッセージが MQSeries ブリッジ・キューへ渡される時まで、少なくともキュー・レベルのセキュリティで保護されます。MQSeries ブリッジ・キューがメッセージを MQSeries システムへ渡すとき、そして、その転送時にクライアント接続でセキュリティ送信および受信出口が使われる場合には、このセキュリティ・レベルは関係なくなります。キュー・レベルのセキュリティが保たれていることについては、検査されません。

MQSeries ブリッジ・キューは同期のみの使用になります。したがって、非同期アプリケーションでは、MQSeries Everyplace ストア・アンド・フォワード / ホーム・サーバー・キュー経由、あるいは、非同期リモート・キュー定義経由で、メッセージをこれらのキューに送信することになります。

ゲートウェイの管理は、通常の MQSeries Everyplace キュー・マネージャーの管理と同じ方法で行われます。すなわち、管理メッセージを使います。管理下のオブジェクトに

合わせて、メッセージの新しいクラスが定義されます。27ページの表11には、ゲートウェイ管理メッセージ・クラスが示されています。

メッセージの変換

MQSeries にあてた MQSeries Everyplace メッセージは、デフォルトの変換機能、または宛先キューに固有な変換機能のいずれかを使うことにより、ブリッジを移動して MQSeries 形式に変換されます。カスタム変換機能は非常に柔軟性があります。たとえば、MQSeries Everyplace ネットワークを経由する特定タイプのメッセージを表すときに、MQSeries Everyplace メッセージ・オブジェクト・クラスのサブクラスを使うことができます。ゲートウェイでは、変換機能は、フィールドと MQSeries 値の適切な割り当てを使い、メッセージを MQSeries 形式に変換すると共に、特定のデータを追加することにより、サブクラスの意味を表します。

MQSeries Everyplace から MQSeries へのデフォルトの変換機能は、サブクラスの情報 を最大限に活用することはできませんが、広範囲の状況に対処できるように設計されています。以下のような特性があります。

- **MQSeries Everyplace から MQS へのメッセージの流れの場合**

MQSeries Everyplace から MQSeries へのデフォルトの変換機能は、MQeMQMsgObject クラスと連結されて機能します。このクラスは、MQ メッセージ・ヘッダーにあるすべてのフィールドを表すものです。MQeMQMsgObject を使うと、使用しているアプリケーションでは、set() メソッドを使って値 (たとえば、優先順位) を設定できます。したがって、MQeMQMsgObject (または MQeMQMsgObject クラスから派生したオブジェクト) を、デフォルトの MQSeries Everyplace 変換機能で処理する場合、デフォルトの変換機能 (MQeBaseTransformer) は、MQeMMsgObject の中から値を入手して、対応する値を MQSeries メッセージにセットします (たとえば、優先順位の値を MQSeries メッセージにコピーする)。

渡されるメッセージが MQeMQMsgObject ではなく、MQeMQMsgObject クラスから派生したものでもない場合、MQSeries Everyplace メッセージ全体が MQSeries メッセージ本体にコピーされます (集約)。MQSeries メッセージ・ヘッダーのメッセージ形式のフィールドでは、MQSeries メッセージがメッセージを MQSeries Everyplace 「集約」形式で保持するよう設定されます。

- **MQSeries から MQSeries Everyplace へのメッセージの流れの場合**

MQSeries Everyplace へ向けた MQSeries メッセージは、他の方向へ移動するメッセージと同じように処理されます。デフォルトの変換機能は、MQSeries ヘッダーのメッセージ・タイプのフィールドを調べ、その設定に従って動作します。

MQSeries ヘッダーが「集約された」MQSeries Everyplace メッセージを示している場合、MQSeries メッセージ本文は元の MQSeries Everyplace メッセージとして再構成され、MQSeries Everyplace ネットワークにポストされます。

メッセージが「集約された」MQSeries Everyplace メッセージでない場合、MQSeries メッセージ・ヘッダーの内容が取り出されて、MQeMQMsgObject オブジェクトに入

られます。MQSeries メッセージ本体は、簡単なバイト・フィールドとして扱われ、MQeMQMsgObject オブジェクトにも入れられます。その後、MQeMQMsgObject が MQSeries Everyplace ネットワークにポストされます。

この MQeMQMsgObject クラスとデフォルトの変換機能の動作では、以下のことが実現されます。

- MQSeries Everyplace メッセージは、変更を加えなくても、MQSeries ネットワーク経由で MQSeries Everyplace ネットワークへ移動できる。
- MQSeries メッセージは、変更を加えなくても、MQSeries Everyplace ネットワーク経由で MQSeries ネットワークへ移動できる。
- MQSeries Everyplace アプリケーションは、MQSeries アプリケーションを変更しなくても、希望する既存の MQSeries アプリケーションを制御できる。

機能

MQSeries リモート・キューは、同期 MQSeries Everyplace によるメッセージング操作の場合に、MQSeries Everyplace キュー・マネージャーから使用できるようになります。他のメッセージング操作はすべて、非同期でなければなりません。

MQSeries Everyplace 管理メッセージは、MQSeries キュー・マネージャーへ送信できません。そこには AdminQ が存在しませんし、管理メッセージの形式が、MQSeries で使われる形式と異なるためです。

互換性

MQSeries Everyplace ネットワークは、MQSeries とは別に存在させられますが、ほとんどの場合、アプリケーションの要件を満たすため、これら 2 つの製品は同時に必要になります。MQSeries Everyplace を既存の MQSeries ネットワークに組み込むことも可能ですが、以下にまとめられている互換性が必要です。

- **アドレッシングと命名について:**
 - キュー・マネージャーとキュー・アドレスを使うときに同一のアドレッシング・セマンティクスであること
 - ASCII ネーム・スペースを共通に使用すること
- **アプリケーションについて:** MQSeries Everyplace は、アプリケーションを変更することなく、既存の MQSeries アプリケーションをサポートできます。
- **チャンネルについて:** MQSeries Everyplace ゲートウェイでは、MQSeries クライアント・チャンネルを使います。
- **メッセージングと内容について:**
 - MQSeries Everyplace と MQS との間のメッセージングであること
 - メッセージ・ネットワークが見えないこと (MQSeries Everyplace か MQSeries からのメッセージが、変更されることなく他のネットワークを経由できる)

- MQSeries メッセージ・ヘッダーで識別されたフィールドが相互にサポートされること
- 1 度だけ確実にメッセージが送達されること
- サンプル・アプリケーションについて: MQSeries Postcard と MQSeries Everyplace Postcard ¹¹ アプリケーションのインターオペラビリティがあること

MQSeries Everyplace バージョン 1 では、MQSeries のすべての機能をサポートしているわけではありません。環境、オペレーティング・システム、そして通信の考慮事項の他に、さらに重大な相違点のいくつかを、以下に示します。ここで、MQSeries Everyplace 内では、MQSeries Everyplace 機能を使った別の方法を使って、多くのアプリケーション作業を行えることに注意してください。すなわち、サブクラスを利用する、提供されたクラスを置き換える、またはルールを利用する、製品に組み込まれたインターフェースや、他のカスタマイズ機能を使うことなどです。

- クラスタ化はサポートされていません
- 配布先リストはサポートされていません
- メッセージのグループ化 / セグメント化はされません
- ロード・バランシング / ウォーム・スタンドバイ機能はありません
- 参照メッセージはありません
- レポート・オプションはありません
- 共用キューはサポートされていません
- トリガーはありません
- 作業単位のサポート、XA 調整はありません

スケラビリティとパフォーマンスの特性は異なります。

確実な送達

MQSeries Everyplace と MQSeries のどちらでも、確実に送達されますが、確実性のレベルはそれぞれ異なります。メッセージが MQSeries Everyplace から MQSeries へ移動するときは、putMessage と confirmPutMessage の組み合わせが使われていれば、確実にメッセージの転送だけが行われます (25ページの『キュー・マネージャー操作』を参照)。メッセージが MQSeries から MQSeries Everyplace へ移動するときは、MQSeries メッセージが永続的と定義されている場合に限り、確実に転送が行われます。

11. このアプリケーションは、バージョン 1.0 には含まれていませんが、WWW の MQSeries ソフトウェア・ダウンロード・サイトで入手できます (<http://www-4.ibm.com/software/ts/MQSeries/>)。

第7章 アプリケーションとユーティリティー

注: これらのアプリケーションおよびユーティリティーは、バージョン 1.0 には含まれていませんが、WWW の MQSeries ソフトウェア・ダウンロード・サイトで入手できます (<http://www-4.ibm.com/software/ts/MQSeries/>)。

Postcard

Postcard は、スタンドアロン MQSeries Everyplace ネットワークの運用、または、MQSeries Everyplace および MQSeries ネットワークの相互運用を妥当性検査するときに使える、MQSeries Everyplace アプリケーションです。Postcard は、リモート・キュー・マネージャーのユーザーヘテキスト・メッセージを送信できるようにする、Java アプリケーションです。このアプリケーションは、Postcard と相互運用されるものです。たとえば、Postcard が MQSeries 宛先へ送信されることにより、Postcard が受信されますし、その逆もあります。

また、C バージョンの MQSeries Everyplace Postcard アプリケーションもあり、PalmOS 上で実行したり、Java バージョンと相互運用もできます。

MQSeries Everyplace Explorer

MQSeries Everyplace Explorer は、Java で作成された管理ツールです。これを使うと、ローカルおよびリモート・キュー・マネージャー、キュー、およびメッセージを構成したり利用できるようになります。このツールでは、標準の Windows グラフィカル・ユーザー・インターフェースを表示するため、Microsoft® ベースのクラスが使われています。そのため、Windows 以外のプラットフォームでは実行できません。しかし、管理メッセージの送受信専用で運用されるため、MQSeries Everyplace キュー・マネージャーすべての管理に使えます。ここで、MQSeries Everyplace ネットワークが 2 つのペイン・ビューに表示されます。左側のペインにはオブジェクトがツリー表示で、右側のペインにはオブジェクトの詳細がリスト表示で示されます。

MQSeries Everyplace Explorer には、以下の機能があります。

- キュー・マネージャーの特性を表示して変更する
- 接続を作成、削除、または変更し、その特性を表示する
- キューを作成、削除、または変更し、その特性または内容 (またはその両方) を表示する
- メッセージをブラウズするか削除し、その特性を表示し、フィールドを調べる
- テスト・メッセージを送信する
- MQSeries への MQSeries Everyplace ブリッジを構成する

MQSeries Everyplace Explorer では、一般的に、すでに構成されたキュー・マネージャーを使い、実行する他のクラスをロードできます。そのようなキュー・マネージャーが存在しなければ、ユーザー選択の特性を指定したキュー・マネージャーが作成されます。

1 つのマシン上で、ツールの複数のコピーを実行することができます。その場合、各コピーはそれぞれの JVM で実行されることとなります。このような設定を使うことにより、MQSeries Everyplace ネットワークをシミュレーションしたり、MQSeries Everyplace ネットワークおよび運用を調べて実演することができます。

第8章 プログラミング・インターフェース

MQSeries Everyplace システム・プログラム・インターフェース (SPI) は、MQSeries Everyplace に対するプログラミング・インターフェースです。オペレーティング・システムに応じて、2 種類のインプリメントが可能です。Java バージョンでは、すべての MQSeries Everyplace 機能へアクセスできます。MQSeries Everyplace バージョン 1.0 の C インターフェースでは、サブセットへのアクセスだけが可能です。クラス、メソッド、プロシージャの詳細は、*MQSeries Everyplace* プログラミング・リファレンスで説明されています。MQSeries Everyplace プログラミングの例は、*MQSeries Everyplace* プログラミング・ガイド で説明されています。

付録. 特記事項

本書において、日本では発表されていない IBM 製品 (機械またはプログラム)、プログラミングまたはサービスについて言及または説明する場合があります。しかし、このことは、弊社がこのような IBM 製品、プログラミングまたはサービスを、日本で発表する意図があることを必ずしも示すものではありません。本書で、IBM ライセンス・プログラムまたは他の IBM 製品に言及している部分があっても、このことは当該プログラムまたは製品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。これらのプログラムまたは製品に代えて、IBM の知的所有権を侵害することのない機能的に同等な他社のプログラム、製品またはサービスを使用することができます。ただし、IBM によって明示的に指定されたものを除き、これらのプログラムまたは製品に関連する稼働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 および他社は、本書で説明する主題に関する特許権 (特許出願を含む)、商標権、または著作権を所有している場合があります。本書は、これらの特許権、商標権、および著作権について、本書で明示されている場合を除き、実施権、使用权等を許諾することを意味するものではありません。実施権、使用权等の許諾については、下記の宛先に、書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31
AP 事業所
IBM World Trade Asia Corporation
Intellectual Property Law & Licensing

IBM は、本書を “現状のまま” 提供するものであり、明示または黙示にかかわらず、非侵害性、商業的な使用可能性、または特定の目的に対する適合性に関する黙示の保証を含み、かつそれには限定されない、いかなる保証も行いません。国によっては特定の商取引における明示または黙示の保証の放棄は認められていません。したがってこの記述はお客様には適用されない場合もあります。

本書の情報は定期的に変更されており、それらの変更は本書の新しい版で取り入れられる予定です。IBM は本書に記載された製品およびプログラムを予告なく改良または変更する場合があります。

IBM はこの Web サイトよりアクセスできるその他の Web サイトに関していかなる保証もしません。お客様が IBM 以外の Web サイトにアクセスされた場合、これらの Web サイトは、IBM から独立して運営されており、IBM は、当該 Web サイトの内容に関していかなる責任も負わないことをご了承ください。さらに、IBM 以外の Web サイトにリンクがはられていることにより IBM が推奨するものでもなく、当該 Web サイトの内容もしくは使用について責任を負うものではありません。

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。

IBM Corporation
Software Interoperability Coordinator
3605 Highway 52 N
Rochester, MN 55901-7829
U.S.A

本プログラムに関する上記の情報は、適切な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。

本書において解説されているライセンス・プログラムおよびそのライセンス・プログラム資料は、「IBM プログラム使用契約書」および「IBM プログラムのご使用条件」の契約条件に基づいて弊社から提供されるものです。

商標

次のものは、IBM Corporation の米国およびその他の国における商標です。

AIX AS/400 IBM MQSeries OS/390

Microsoft、Windows、Windows NT、および Windows のロゴは 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 Sun Microsystems, Inc. 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。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標または登録商標です。

用語集

この用語集では、本書で使われる用語や、通常の意味とは違う意味で使われる言葉を説明します。場合によっては、ある定義は 1 つの用語だけに使われるものではないことがあります。しかし、その言葉が本書で使われるときの特定の意味を説明します。

探している用語がなければ、索引か *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994 を参照してください。

アプリケーション・プログラミング・インターフェース (Application Programming Interface (API)). アプリケーション・プログラミング・インターフェースは、プログラマーがアプリケーションで使うことのできる関数と変数で構成される。

非同期メッセージング (asynchronous messaging). プログラムがメッセージをメッセージ・キューに入れるときの、プログラム間での通信方式。非同期メッセージングでは、送信側プログラムは、メッセージに対する応答を待機することなく処理を進める。同期メッセージング (*synchronous messaging*) と対比。

認証プログラム (authenticator). メッセージの送信側と受信側を検査するプログラム。

ブリッジ (bridge). MQSeries Everyplace と他のメッセージ・システム (MQSeries を含む) との間で、メッセージを流せるようにする MQSeries Everyplace オブジェクト。

チャンネル (channel). 動的チャンネル (*dynamic channel*) および MQI チャンネル (*MQI channel*) を参照。

チャンネル・マネージャー (channel manager). エンドポイント間での論理的な複数の並行通信パイプをサポートする MQSeries Everyplace オブジェクト。

クラス (class). クラスとは、データおよびそのデータ上で運用するメソッドの集合をカプセル化したものである。特定のクラスをインスタンス化することにより、クラスのインスタンスであるオブジェクトを作成することができる。

クライアント (client). MQSeries では、クライアントとは実行時コンポーネントのことであり、ローカル・ユーザー・アプリケーションのために、サーバー上のキュー・サービスへアクセスできる。

圧縮プログラム (compressor). メッセージを圧縮して、伝送するデータのボリュームを削減するプログラム。

暗号化プログラム (cryptor). メッセージを暗号化して、伝送時のセキュリティを保つプログラム。

動的チャンネル (dynamic channel). 動的チャンネルは、MQSeries Everyplace デバイスを接続し、同期メッセージと非同期メッセージを転送し、両方向の方式で応答する。

カプセル化 (encapsulation). カプセル化は、オブジェクト指向のプログラミング手法であり、オブジェクトのデータをプライベートにするか保護する。プログラマーは、メソッドを呼び出すことによつてのみ、データをアクセスしたり処理できる。

ゲートウェイ (gateway). MQSeries Everyplace ゲートウェイ (またはサーバー) は、チャンネル・マネージャーを含む、MQSeries Everyplace コードを実行するコンピューターのことである。

ハイパーテキスト・マークアップ言語 (Hypertext Markup Language (HTML)). WWW で表示される情報を定義するときに使われる言語。

インスタンス (instance). インスタンスはオブジェクトの 1 つ。クラスがインスタンス化されてオブジェクトが作成される場合、そのオブジェクトはそのクラスのインスタンスといえる。

インターフェース (interface). インターフェースとは、抽象メソッドだけを含み、インスタンス変数を含まないクラスのことである。1 つのインターフェースには一連の共通メソッドが備えられており、たくさんある異なるクラスのサブクラスによってインプリメントできる。

インターネット (Internet). インターネットは、情報を共有するための協同公衆ネットワークである。物理的には、インターネットでは、現存する公衆通信ネットワークすべてのリソース全体の部分集合を使用する。技術的には、インターネットを協同公衆ネットワークとして区別している要素は、TCP/IP (伝送制御プロトコル / インターネット・プロトコル) というプロトコル群の使用である。

Java 開発者キット (Java Developers Kit (JDK)). Java 開発者のために Sun Microsystems 社によって配布されるソフトウェアのパッケージ。ここには、Java インタープリター、Java クラス、そして Java 開発ツール (コンパイラー、デバッガー、逆アセンブラー、アプレット・ビューアー、スタブ・ファイル生成プログラム、および文書生成プログラム) が含まれる。

Java Naming and Directory Service (JNDI). Java プログラム言語で指定される API。Java プログラム言語で作成されたアプリケーションに、命名機能とディレクトリー機能を付加する。

Lightweight Directory Access Protocol (LDAP). LDAP は、ディレクトリー・サービスにアクセスするための、クライアント / サーバー・プロトコルである。

メッセージ (message). メッセージ・キューイング・アプリケーションでは、メッセージとは、プログラム間で送信される通信のことである。

メッセージ・キュー (message queue). キュー (queue) を参照。

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、メッセージをキューに入れることにより、他のプログラムと通信するというプログラミング手法。

メソッド (method). メソッドは、関数またはプロシージャのための、オブジェクト指向プログラミング用語である。

MQI チャネル (MQI channel). MQI チャネルは、MQSeries クライアントをサーバー・システムのキュー・マネージャーに接続し、MQI 呼び出しを転送して、両方向の方式で応答する。

MQSeries. MQSeries は、IBM ライセンス・プログラムのファミリーの 1 つで、メッセージ・キューイング・サービスを提供する。

オブジェクト (object). (1) Java では、オブジェクトとはクラスのインスタンスのことである。クラスとは、物事のグループをモデル化したものである。オブジェクトとは、そのグループの特定のメンバーをモデル化したものである。(2) MQSeries では、オブジェクトとはキュー・マネージャー、キュー、またはチャネルのことである。

パッケージ (package). Java におけるパッケージとは、Java コードの一部を使うことにより、特定クラス群にアクセスできるようにする方式のことである。特定のパッケージの一部となっている Java コードでは、そのパッケージのすべてのクラス、そしてクラスのすべてのプライベート・メソッドおよびフィールドにアクセスできる。

携帯情報端末 (personal digital assistant (PDA)). ポケット・サイズのパーソナル・コンピューター。

プライベート (private). プライベート・フィールドは、クラスの外側からは見えない。

保護 (protected). 保護フィールドは、クラス内、サブクラス内、そして、クラスが一部となっているパッケージ内からのみ見ることができる。

公開 (public). 公開クラスまたはインターフェースは、どこからでも見える。公開メソッドまたは変数は、クラスが見えるところではどこからでも見える。

キュー (queue). キューとは MQSeries オブジェクトのことである。メッセージ・キュー・アプリケーションでは、メッセージをキューに入れ、キューからメッセージを取り出すことができる。

キュー・マネージャー (queue manager). キュー・マネージャーとは、メッセージ・キューイング・サービスをアプリケーションに提供するシステム・プログラムである。

サーバー (server). (1) MQSeries Everyplace サーバーは、MQSeries Everyplace チャンネル・マネージャーが構成されているデバイスである。(2) MQSeries サーバーは、リモート・ワークステーション上で稼動するクライアント・アプリケーションに、メッセージ・キューイング・サービスを提供するキュー・マネージャーである。(3) さらに一般的には、サーバーとは、クライアント / サーバーでの特定の 2 プログラム間情報フロー・モデルにおいて、情報の要求に応答するプログラムのことである。(4) サーバー・プログラムを実行するコンピューター。

サーブレット (servlet). Web サーバー上でのみ稼働するように設計された Java プログラム。

サブクラス (subclass). サブクラスとは、他のクラスから拡張されるクラスのことである。サブクラスは、そのスーパークラスの公開メソッドと保護メソッド、および公開変数と保護変数を継承する。

スーパークラス (superclass). スーパークラスとは、他のクラスに拡張されるクラスのことである。スーパークラスの公開メソッドと保護メソッド

ド、および公開変数と保護変数については、サブクラスでも使うことができる。

同期メッセージング (synchronous messaging). プログラムがメッセージをメッセージ・キューに入れるときの、プログラム間での通信方式。同期メッセージングでは、送信側プログラムは、処理を再開する前に、メッセージへの応答を待機する。非同期メッセージング (asynchronous messaging) と対比。

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 (TCP/IP)). ローカル・エリア・ネットワークでも、広域ネットワークの場合でも、対等通信接続機能をサポートする一連の通信プロトコル。

ウェブ (Web). ワールド・ワイド・ウェブ (World Wide Web) を参照。

Web ブラウザー (Web browser). WWW で配布される情報を形式設定して表示するプログラム。

ワールド・ワイド・ウェブ (World Wide Web). ワールド・ワイド・ウェブ (WWW) は、一連の共通プロトコルに基づくインターネット・サービスのことである。これにより、特別に構成されたサーバー・コンピューターを使って、インターネットを経由して規格化された方法で文書を配布できる。

参照文献

関連資料:

- *MQSeries Everyplace* 最初にお読みください, (GC88-8656-00)
- *MQSeries Everyplace* プログラミング・リファレンス, (SC88-8655-00)
- *MQSeries Everyplace* プログラミング・ガイド, (SC88-8654-00)
- *MQSeries An Introduction to Messaging and Queuing*, (GC33-0805-01)
- *MQSeries (Windows NT®版)* インストールの手引き, (GD88-7162-00)

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アダプター, MQSeries Everyplace 31
圧縮 35
アプリケーション 59
アプリケーション, MQSeries Everyplace 11
アプリケーション, Postcard 59
アプリケーションのロード 47
暗号化 35
イベント・ログ 32
インターフェース, セキュリティー 43
インターフェース, プログラミング 61
インターフェース, MQSeries への 49
エンティティー, 認証可能な 40
オブジェクト, メッセージ 14
オブジェクト, MQSeries Everyplace 14
オペレーティング・システム, サポートされる 3

[カ行]

概念, 製品 13
概要 1
確実なメッセージ送達 58
カスタマイズ 43
管理, MQSeries Everyplace での 26
管理メッセージ 27
管理ユーティリティー 59
機能 11
キュー, ストア・アンド・フォワード 20
キュー, リモート 19
キュー, ローカル 19
キュー, MQSeries Everyplace 18
キュー, MQSeries ブリッジ 20, 21
キューのルール 44
キュー・ベースのセキュリティー 37
キュー・マネージャー 6, 45
キュー・マネージャー, MQSeries Everyplace 23
キュー・マネージャー操作 25
キュー・マネージャーのルール 44
キュー・マネージャー・プロキシ・オブジェクト 51

クライアント / サーバー接続 46
クライアント, MQSeries 6
クライアント・チャンネル 7, 13
クラス, MQSeries Everyplace 47
ゲートウェイ, MQSeries Everyplace 7, 13
公開レジストリー 41
構成 43
構成, MQSeries Everyplace 32
互換性, MQSeries との 57
顧客の要件 12

[サ行]

サーバー, MQSeries 6
サポートされるオペレーティング・システム 3
自動登録 41
商標 64
私用レジストリー 41
スケーラビリティ 32
ストア・アンド・フォワード・キュー 20
製品の概念 13
セキュリティー, キュー・ベースの 37
セキュリティー, メッセージ・レベルの 38
セキュリティー, ローカル 37
セキュリティー, MQSeries Everyplace 35
セキュリティー・インターフェース 43
接続, クライアント / サーバー 46
接続, 対等通信 46
接続スタイル 45
接続スタイル, 複数 47
説明 1
前提条件 3
前提条件となる知識 ix
操作, キュー・マネージャー 25
属性のルール 43

[タ行]

対象 ix
対等通信接続 46
ダイヤルアップ接続の管理 31
ダンプ・データ・フォーマット 18
チャンネル, クライアント 7, 13
チャンネル, 動的 7, 13, 29

チャンネル・マネージャー 45
チャンネル・リスナー 45
通信 45
デバイス、MQSeries Everyplace 7, 13
同期メッセージング 23, 35
動的チャンネル 7, 13, 29
読者 ix
特記事項 63
トレース、MQSeries Everyplace の 31

[ナ行]

認証可能なエンティティ 40
認証の複写 41
ネットワーク、MQS 49
ネットワーク、MQSeries Everyplace 32, 49

[ハ行]

パーベイス・メッセージング 6
発行サービス、ミニ認証 42
必要なオペレーティング・システム 3
非同期メッセージング 23, 34
フォーマット、ダンプ・データの 18
複写、認証の 41
複数接続スタイル 47
ブリッジ、MQS 49
ブリッジ・オブジェクト 50, 51
プログラミング・インターフェース 61
分散メッセージング 6
変換機能 56
ホーム・サーバー、MQSeries Everyplace 20
ホーム・サーバー・キュー 20
ホスト・メッセージング 6
本書について ix
本書の対象読者 ix

[マ行]

ミニ認証 40
ミニ認証発行サービス 42
メッセージ、管理 27
メッセージ送達、確実な 58
メッセージの変換 56
メッセージング、同期 23, 35
メッセージング、非同期 23, 34
メッセージング、MQSeries 5

メッセージ・オブジェクト 14
メッセージ・レベルのセキュリティ 38
モニター 29

[ヤ行]

ユーティリティ 59
ユーティリティ、MQSeries Everyplace Explorer 59
要件、顧客 12
用語 ix

[ラ行]

リスナー・オブジェクト 53, 54
リモート・キュー 19
ルール、MQSeries Everyplace 43
レジストリー 40
レジストリー、公開 41
レジストリー、私用 41
レジストリー、MQSeries Everyplace 13
ローカル・キュー 19
ローカル・セキュリティ 37
ロード、アプリケーションの 47

[ワ行]

ワークステーション・メッセージング 6

M

MQeAttribute 39
MQeMTrustAttribute 39
MQSeries Everyplace Explorer ユーティリティ 59
MQSeries Everyplace アダプター 31
MQSeries Everyplace アプリケーション 11, 59
MQSeries Everyplace オブジェクト 14
MQSeries Everyplace 管理 26
MQSeries Everyplace 管理ユーティリティ 59
MQSeries Everyplace キュー 18
MQSeries Everyplace キュー・マネージャー 23
MQSeries Everyplace クラス 47
MQSeries Everyplace ゲートウェイ 7, 13
MQSeries Everyplace 構成 32
MQSeries Everyplace デバイス 7, 13
MQSeries Everyplace ネットワーク 32, 49
MQSeries Everyplace のセキュリティ 35

MQSeries Everyplace ユーティリティー 59
MQSeries Everyplace ルール 43
MQSeries Everyplace レジストリー 13, 40
MQSeries Integrator 5
MQSeries Workflow 5
MQSeries クライアント 6
MQSeries サーバー 6
MQSeries との互換性 57
MQSeries ネットワーク 49
MQSeries ファミリー 5
MQSeries ブリッジ 49
MQSeries ブリッジのルール 43
MQSeries ブリッジ・キュー 20, 21
MQSeries へのインターフェース 49
MQSeries メッセージング 5

P

Postcard アプリケーション 59

R

RAS ダイアラーのルール 44

S

SPI 61



部品番号: CT84GJA

Printed in Japan

GC88-8653-00



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12

CT84GJA



Spine information:



MQSeries® Everyplace

MQSeries Everyplace 紹介

バージョン 1