

MQSeries® Everyplace for Multiplatforms



紹介

バージョン 1.2

MQSeries® Everyplace for Multiplatforms



紹介

バージョン 1.2

ご注意

本書の情報およびそれによってサポートされる製品を使用する前に、81ページの『付録. 特記事項』に記載す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 1.2 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。

この製品、または本製品を使用するアプリケーションを実稼働環境に配置する前に、必要なライセンスをもっていることをご確認ください。

指定したサーバー・プラットフォームで MQSeries Everyplace を (コード開発とテストの目的以外で) 使用する場合、各マシンおよびマシン・アップグレードで本プログラムを使用するライセンスを受けるために、*capacity-unit 使用許可* (「ライセンス証書」文書に記されていて、発行されている *capacity unit table* および *pricing group table* に従って MQSeries Everyplace の使用をサポートするために有効なもの) を入手する必要があります。

指定したクライアント・プラットフォームで本製品を (コード開発およびテストの目的以外で) 使用するには、*device platform 使用許可* (「ライセンス証書」文書に記されていて、MQSeries Everyplace の使用をサポートするために有効なもの) が必要です。これらのライセンスは、MQSeries Everyplace Bridge を使用したり、IBM によって発行された MQSeries Everyplace *pricing group* リスト (下記の URL によって Web でも使用可能) に指定されているサーバー・プラットフォームで実行したりする資格をユーザーに与えるものではありません。

これらの制約事項の詳細については、<http://www.ibm.com/software/mqseries> を参照してください。

この版は、MQSeries Everyplace バージョン 1.2、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。

この文書は、新しい情報および改訂される情報によって、継続的に更新されます。最新版については、<http://www.ibm.com/software/mqseries/library/> にある MQSeries ファミリー・ライブラリーの Web ページを参照してください。

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： GC34-5843-02
MQSeries® Everyplace for Multiplatforms
Introduction
バージョン 1.2

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 2001.5

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, 2001. All rights reserved.

Translation: © Copyright IBM Japan 2001

目次

本書について	vii
本書の対象読者	vii
前提条件となる知識	vii
本書で使用する用語	viii
変更の要約	ix
この版 (GC88-8653-02) での変更点	ix
前版 (GC88-8653-01) での変更点	ix
第1章 概要	1
第2章 ソフトウェア環境	3
サポートされるプラットフォーム	3
Java 環境	4
ストレージ要件	4
第3章 MQSeries ファミリー	7
MQSeries ホスト製品と分散製品	8
MQSeries Everyplace	9
第4章 製品の要件	13
機能	13
アプリケーション	13
顧客の要件	14
第5章 製品の概念	17
紹介	17
メッセージ・オブジェクト	18
ダンプ・データ・フォーマット	23
キュー	24
キュー・マネージャー	30
キュー・マネージャー構成	34
キュー・マネージャー操作	37
接続	38
管理	40
管理メッセージ	40
選択的な管理	42
モニターおよび関連処置	42
動的チャネル	43
アダプター	44
ダイヤルアップ接続の管理	44
トレース	45
イベント・ログ	45
メッセージ送達	45

	非同期メッセージ送達	45
	同期メッセージ送達	46
	セキュリティ	47
	MQSeries Everyplace のローカル・セキュリティ	48
	MQSeries Everyplace キュー・ベースのセキュリティ	48
	メッセージ・レベルのセキュリティ	50
	レジストリー	51
	MQSeries Everyplace の認証可能なエンティティ	51
	専用レジストリーと信任状	52
	自動登録	52
	公開レジストリーと認証の複製	53
	レジストリー・サービスのアプリケーションの使用	53
	デフォルトのミニ認証発行サービス	53
	セキュリティ・インターフェース	54
	カスタマイズ	54
	ルール	54
	接続スタイル	56
	対等通信接続	57
	クライアント / サーバー接続	57
	複数接続スタイル	57
	クラス	58
	アプリケーションのロード	58
	第6章 MQSeries Everyplace ネットワークと MQSeries ネットワーク	59
	MQSeries へのインターフェース	59
	メッセージの変換	67
	機能	69
	互換性	69
	確実な送達	70
	第7章 プログラミング・インターフェース	71
	第8章 MQSeries Everyplace を開始するには	73
	MQSeries Everyplace の使用	74
	経験の紹介	75
	ES02 の最初の使用: MQe_Explorer	75
	付録. 特記事項	81
	商標	82
	用語集	83
	参考文献	87
	索引	89

本書について

本書は、MQSeries Everyplace for Multiplatforms 製品 (本書では通常 MQSeries Everyplace と記載されている) 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の詳細については、「*MQSeries Everyplace for Multiplatforms プログラミング・リファレンス*」および「*MQSeries Everyplace for Multiplatforms プログラミング・ガイド*」を参照してください。

MQSeries Everyplace for Multiplatforms での他のプログラム言語の使用に関する情報は、「*MQSeries Everyplace for Multiplatforms Native Client Information*」を参照してください。

MQSeries Everyplace for Multiplatforms のインストール手順については、「*MQSeries Everyplace for Multiplatforms 最初にお読みください*」を参照してください。

この文書は、新しい情報および改訂される情報によって、継続的に更新されます。最新版については、<http://www.ibm.com/software/ts/mqseries/library/> にある MQSeries ファミリー・ライブラリーの Web ページ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興味のあるユーザー、および MQSeries Everyplace メッセージング・ネットワークの有効範囲を拡張する必要が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*

この資料は、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

以下の MQSeries 製品を指します。

- **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設定が可能になります。
- **MQSeries Messaging** 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 メッセージング

以下の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:** MQSeries for Windows NT、AIX[®]、AS/400[®]、HP-UX、Sun Solaris、およびその他のプラットフォーム
- **ホスト・メッセージング:** MQSeries for OS/390[®]
- **パーベイスブ・メッセージング:** MQSeries Everyplace

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

3 番目の MQSeries メッセージング製品グループ、パーベイスブ・メッセージングを指します。

デバイス・プラットフォーム

MQSeries Everyplace をクライアントとしてのみ実行することのできる小さなコンピューター。

サーバー・プラットフォーム

MQSeries Everyplace をサーバーまたはクライアントとして実行することのできる、任意のサイズのコンピューター。

ゲートウェイ

MQSeries-Bridge 機能を組み込む MQSeries Everyplace プログラムを実行する、任意のサイズのコンピューター。

変更の要約

このセクションでは、「MQSeries *Everyplace for Multiplatforms* 紹介」のこの版における変更点を説明します。本書の直前の版からの変更点は、変更部分の左側にある縦線によって示されています。

この版 (GC88-8653-02) での変更点

前版 (GC88-8653-01) での変更点

以下の情報が追加されています。

- MQSeries *Everyplace* を AIX および Solaris 上で使用するための情報。
- ストレージ要件。
- ご意見記入用紙。

第1章 概要

MQSeries Everyplace は、ビジネス・メッセージング製品である、MQSeries ファミリーのメンバーです。センサー、電話、PDA (携帯情報端末)、およびラップトップ・コンピュータなどの軽量デバイスのメッセージングの必要を満たすだけでなく、移動性、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。一回だけの確実な送達を実行して、標準の MQSeries サービス品質を保つとともに、他のファミリーのメンバーとメッセージを交換します。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、メッセージング・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、カスタマー・セットに応じて、調整済みの機能とインターフェースを提供しており、ファミリーの他のメンバーと全く同じ能力を提供することを目的とはしていません。一方、包括的なセキュリティーの提供、メッセージ・オブジェクト、同期および非同期のメッセージング、リモート・キュー・アクセス、およびメッセージの push と pull など、ユーザーの特定クラスをサポートするために、固有の機能は組み込んでいます。

また MQSeries Everyplace は、IBM パーベイシブ・コンピューティング・ファミリーの他のメンバーや、Websphere Application Server の他のコンポーネントとうまく統合されるように設計されています。

第2章 ソフトウェア環境

サポートされるプラットフォーム

MQSeries Everyplace は、特定のサーバー・プラットフォームに直接、インストール可能です。プログラムおよび Java クラスを他のプラットフォームに転送するには、適切なダウンロードまたはファイル転送プログラム（提供されていません）を使用する必要があります。

インストール・サポートがある、直接サポートされるプラットフォーム

以下は、組み込みツールを使用して製品をインストールできるプラットフォームです。

- Windows NT[®] v4
- Windows[®] 2000
- Windows 95/98/ME
- AIX バージョン 4.3
- Sun Solaris バージョン 7 または 8
- Linux Intel Kernel 2.2 (ZIP ファイルを使用してインストールされます)。
- HP-UX 11.0 (ZIP ファイルを使用してインストールされます)。

インストール・サポートがない、直接サポートされるプラットフォーム

以下は、MQSeries Everyplace のテストおよび配置のためにサポートされるプラットフォームですが、別のプラットフォームからのファイル転送によるインストールだけしかサポートされません。

- HP Jornada デバイス (モデル 680 または 820) で実行される WinCE 2.1
- Psion デバイス (5MX Pro または NetBook) で実行される EPOC 32 ビット、リリース 5
- Palm V および IBM Workpad C3 で実行される Palm OS V3.0 またはそれ以降
- Java を備えた IBM 4690 OS

間接的にサポートされるプラットフォーム

以下のプラットフォームを使用することはできますが、それらの Java 環境が、直接サポートされるプラットフォーム上の Java 環境と完全に互換性をもつ場合にのみサポートされます。問題は、上記にリストしたテスト済みのプラットフォーム上でのみ調べることができます。

- カーネル 2.2 を実行する zSeries 上の Linux
- iSeries
- OS/2

ソフトウェア環境

- EPOC (上記にリストしたデバイス以外のデバイス上)
- WinCE (上記にリストしたデバイス以外のデバイス上)
- QNX Neutrino
- Pocket PC
- Palm OS (上記にリストしたデバイス以外のデバイス上)
- 『Java 環境』にリストされているいずれかの Java 環境を実行するその他のプラットフォーム

Java 環境

以下のいずれかの Java ランタイム環境が必要です。

- Java Micro Edition を含む、IBM Java ランタイム (JVM 1.3 またはそれ以降)
- Sun Java に認可された任意の Java (V1.1 またはそれ以降)¹

注: サービスが使用可能になるためには、Java は、以下のいずれかのプラットフォームでテストされたものと完全に互換性がなければなりません。

- Windows CE オペレーティング・システムを実行する HP Jornada デバイス (モデル 680 または 820)
- EPOC オペレーティング・システムを実行する Psion デバイス (5MX Pro または NetBook)
- 上にリストした直接サポート・サーバー・プラットフォームのいずれか

MQSeries-Bridge 操作には、MQSeries Classes for Java が必要です。MQSeries Classes for Java のバージョンを実行するのに必要な Java のレベルをチェックする必要があります。

ストレージ要件

次の表は、MQSeries Everyplace のインストールを実行するために必要なストレージを示しています。

表 1. インストールを実行するために必要なストレージ

オペレーティング・システム	必要なストレージ
Windows NT (ファイル・システム = NTFS)	26Mb
AIX	29Mb
Solaris	27Mb

1. JIT (Just In Time) コンパイラーを使用可能にして Sun の JVM のもとでインストーラーを実行すると、問題が発生することがあります。Sun JVM を使用する場合は、次のコマンドを使用して、JIT コンパイラーを使用不可にすることをお勧めします。コマンドは、次のとおりです: `java -Djava.compiler=NONE install`

次の表は、インストール後に MQSeries Everyplace ファイル用として必要なストレージを示しています。

表 2. MQSeries Everyplace 用に必要なストレージ

オペレーティング・システム	必要なストレージ
Windows NT (ファイル・システム = NTFS)	9.5Mb
AIX	11Mb
Solaris	10Mb

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ています。図1 に示されているとおりです。

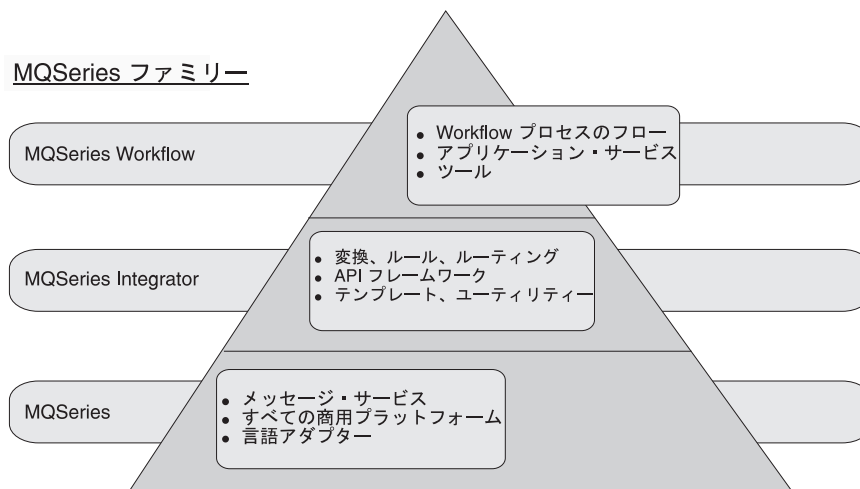


図1. MQSeries ファミリー

- **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・ソフトウェアであり、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設定が可能になります。
- **MQSeries Messaging** 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 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。3 つの製品グループがあります。

MQSeries ファミリー

- 分散メッセージング: MQSeries for Windows NT、AIX、AS/400、HP-UX、Sun Solaris、およびその他のプラットフォーム
- ホスト・メッセージング: MQSeries for OS/390
- 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（リモート・キュー・マネージャーによって所有されているもの）に入れられると、そのリモート・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 つまたは複数の中間キュー・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、必要であれば中間点でメッセージをキューに入れ、送信アプリケーションを受信アプリケーションから分離することです。すべてのMQSeries メッセージング製品は、詳細においては相違点が数多くあるものの、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 ホスト製品と分散製品

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構成にはすべて、クライアントとサーバー² が関係します。その例は、図2 に例示してあります。

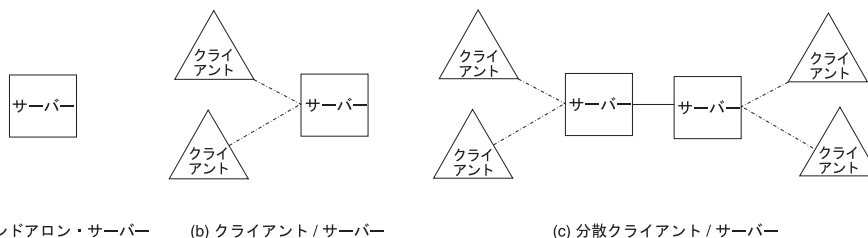


図2. 単純なホスト構成および分散構成

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 つまたは複数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。この構成では、キュー・マネージャーはサーバー上にのみ存在し、それぞれのクライアントは、クライアント・チャンネル を介してキュー・マネージャーにアク

2. これらの用語は、MQSeries ホスト、分散、およびワークステーションのメッセージング製品では非常に特殊な意味をもっており、一般的な用法と常に対応するものとは限らないことに注意してください。

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・プロシージャー呼び出し (RPC) に似たものをインプリメントする) を送信する、両方向の通信リンクです。アプリケーションは、サーバーのキューにアクセスし、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 Everyplace

MQSeries Everyplace は、キュー・マネージャーの提供によって、幅広い多様なネットワーク構成をサポートしています。それぞれのキュー・マネージャーで、適切な機能が使用可能です。MQSeries のホスト製品や分散製品におけるような、クライアントまたはサーバーという概念はありません。MQSeries Everyplace キュー・マネージャーは、従来のクライアントまたはサーバーとして動作できますが、実際には単に、アプリケーション定義のタスクを実行するように使用可能にされたキュー・マネージャーです。このことの例証として、MQSeries Everyplace キュー・マネージャーは、ローカル・キューがあっても、ローカル・キューがなくても構成することができます。ローカル・キューがある場合、メッセージをローカルに保管するので、アプリケーション非同期メッセージングが提供されますが、ローカル・キューがない場合は、同期メッセージングに制限されます。キュー・マネージャーを、ブリッジ機能を付けて構成することも、ブリッジ機能なしで構成することもできるというのが、調整された構成を示すもう 1 つの例です。ブリッジがある場合は、MQSeries ホストまたはワークステーションのキュー・マネージャーとメッセージを交換する能力があり、ブリッジがない場合は、他の MQSeries Everyplace キュー・マネージャーと直接、通信することしかできません (ただし、ネットワーク内のブリッジ能力をもつ他のキュー・マネージャーを介して間接的に通信することはできません)。

注: MQSeries Integrator (MQSI) 用の新規ノードを使用すると、MQSeries-Bridge を使用することなく、MQSeries Everyplace の直接的な接続が可能です。

MQSeries Everyplace キュー・マネージャーは動的チャンネル を使用して、情報を交換します。これらの動的チャンネルには、ファミリーの他のメンバーで使用されるクライアント・チャンネル および メッセージ・チャンネル とは異なる特性があります。動的チャンネルの主要な機能は以下のとおりです。

- 同期 および非同期 の両方のメッセージングのサポート。同期メッセージングは、ソース・キュー・マネージャーでのキューイングなしに、ソース・アプリケーションからターゲット・キューに直接、伝送サービスを提供します。非同期メッセージングは、ソース・キュー・マネージャーでのキューイングを伴う、ソース・キュー・マネージャーからターゲット・キューへの伝送サービスです。
- 終端間サービス の提供。チャンネルは、中間キュー・マネージャーを経由して、ソース・キュー・マネージャーから宛先キュー・マネージャーに進みます。使用される基礎のトランスポート・プロトコルは、チャンネルがこれらの中間メディアを移動するときに変わることがあります。
- 圧縮、暗号化、および認証 のサポート。チャンネルには、転送中のデータを保護するためにこれらのセキュリティー特性があります。
- 対等通信 操作およびクライアント / サーバー 操作のサポート。対等通信として定義されたチャンネルには、ソース (送信元) またはターゲット (宛先) のいずれもがチャンネル上の操作を開始できるという意味で、機能上、対称性があります。クライアント / サーバー・チャンネルは要求 / 応答であり、クライアントはサーバーの要求を行い、サーバーはその要求に応答します。(これによってメッセージの流れが制限されるわけはありません。メッセージは、クライアントからサーバーに、またサーバーからクライアントに流れることができます)。

ブリッジ構成オプションを使用すると、MQSeries Everyplace キュー・マネージャーは、クライアント・チャンネル を介して MQSeries ホストおよび分散のキュー・マネージャーと通信できます。ブリッジ・コンポーネントは、1 つまたは複数のホストまたは分散のキュー・マネージャーに接続できる、クライアント・チャンネルのプールを管理します。ブリッジが使用可能な複数の MQSeries Everyplace キュー・マネージャーを単一ネットワークで構成して、必要な機能、パフォーマンス、および信頼性を提供することができます。

次の図には、一般的な、ただし任意の、MQSeries Everyplace 構成を示しています。明解な図解にするため、これらの図には、定義済みの直接接続だけを示しています。直接接続を活用する間接接続を定義することもできます。対等通信チャンネルを表すには、両方向の矢印ヘッドをもつ線を使用しており、クライアント / サーバー・チャンネルは、サーバーを指す矢印をもつ線で表しています。クライアントはクライアント / サーバー・チャンネルを使用して、サーバーへのメッセージの送信と、そのサーバーからの自分あてに送られたメッセージの pull の両方を行うことができます。矢印のない線は、MQSeries Everyplace と MQSeries 間の通信を使用可能にする MQSeries クライアント・チャンネルを示しています。



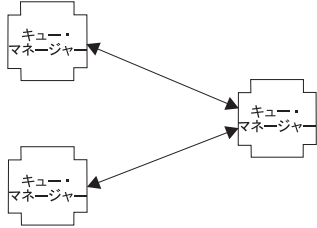
(a) スタンドアロン・キュー・マネージャー

(a) キューを使用してデータを交換する 1 つまたは複数のアプリケーションをサポートするために使用される、スタンドアロン・キュー・マネージャーを示しています。



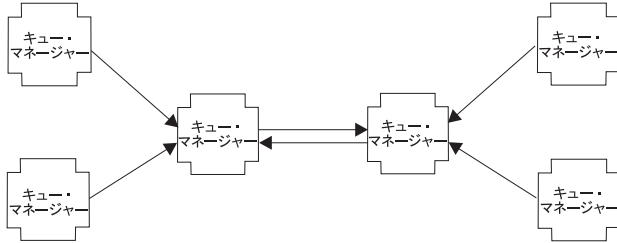
(b) 対等通信構成

(b) 対等通信チャンネルで相互接続される 2 つのキュー・マネージャーを示しています。



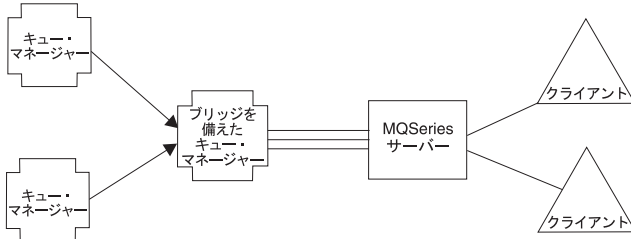
(c) 対等通信構成

(c) 3 番目のキュー・マネージャーへの 2 番目の直接の対等通信チャンネルを示しています。これら 3 つはすべて、基礎の直接接続を使用して、互いにデータを交換できます。



(d) 小規模ネットワーク

(d) 中央サーバーのキュー・マネージャーが、直接クライアント / サーバー・チャンネルの対を使用して情報を交換する、小規模なネットワーク構成を示しています。クライアント・キュー・マネージャーはそれぞれ、直接クライアント / サーバー・チャンネルを使用して、サーバー・キュー・マネージャーの 1 つと接続します。



(e) 統合された MQSeries ファミリー・ネットワーク

MQSeries ファミリー

(e) キュー・マネージャーの 1 つがブリッジ・オプションで構成されていて、クライアント・チャンネルのプールがすべて、単一のターゲット MQSeries ホスト / 分散サーバーに送信されている、MQSeries Everyplace 構成を示しています。

第4章 製品の要件

この章では、MQSeries Everyplace の設計とインプリメンテーションを形成している要件を説明します。

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。MQSeries Everyplace は、ラップトップ、ワークステーション、分散およびホストの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、確実に一回だけのメッセージ送達により、同一のサービス品質を提供し、他のファミリー・メンバーとのメッセージ交換も可能です。
- 軽量のメッセージング機能を提供する。
- 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わた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。MQSeries Everyplace には、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リカバリー機能があります。
- 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。MQSeries Everyplace では、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。これにより、MQSeries Everyplace が、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ための適切なベースとなります。
- 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、および振る舞い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

選択肢が多岐にわたるので、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が、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。

製品の要件

- **小売アプリケーション:** メッセージ・ブローカーなど、ホスト・システムへの現金取り引きの少量ずつの送り込み。
- **消費者アプリケーション:** PDA を使用して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調理場と連絡をとる、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。
- **個人の生産性:** メール / カレンダーの複製、データベースの複製、ラップトップのダウンサイジング。

顧客の要件

MQSeries Everyplace の設計に影響した要件には、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 と同じサービス品質および直接に行うメッセージング交換、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:** Palm デバイス用、64K バイト。Java デバイスの場合、100K バイトの最小クラス・ファイル・サイズ。
- **機能:** 同期メッセージングおよび非同期メッセージング機能、ローカル・キューまたはリモート・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な機能、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振る舞いを制御 (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振る舞いなど)。
- **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ン

グ・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・ユーザー信任と統合する機能、国内のセキュリティー要件に準拠して、セキュリティー・サポートを、国境を越えるメッセージとして変更する機能。

製品の要件

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とり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。構成データはレジストリーに保管されます。

MQSeries Everyplace オブジェクト構造は、MQe_Explorer 管理ツールから次のディスプレイで見ることができます。

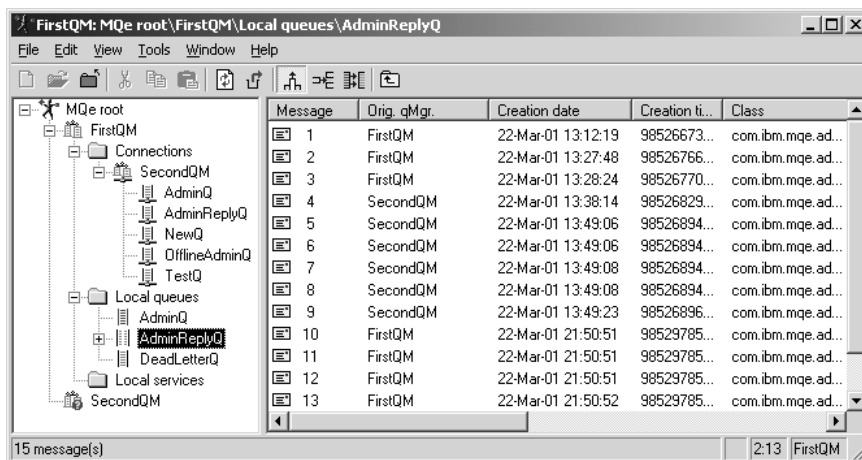


図 3. MQe_Explorer によって表示される MQSeries Everyplace オブジェクト

左側の画面区画には、オブジェクトがツリー構造で表示されています。ルート MQe root の下には、2 つのキュー・マネージャー、ExampleQM と OfficeQM があります。OfficeQM の構造を展開すると、「接続 (Connections)」、「ローカル・キュー (Local queues)」、および「ローカル・サービス (Local services)」のフォルダーが現れます。「接続 (Connections)」を展開すると、OfficeQM キュー・マネージャーから ExampleQM キュー・マネージャーへの接続定義が現れます。リモート・キュー・マネージャー OfficeQM には、ExampleQM のキューの 2 つ、AdminQ と AdminReplyQ の定義がありま

す。OfficeQM の「ローカル・キュー (Local queues)」を展開すると、5 つのローカル・キューが現れます。そのうち 1 つは選択済みであり、その内容が右側の画面区画に表示されています。

MQSeries Everyplace を理解するには、メッセージ、キュー、キュー・マネージャー、および接続の性質についての知識が必要です。これらの概念については以下のセクションで概説します。

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 メッセージングでサポートされ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。メッセージ・ヘッダーは MQSeries によって理解されます。この部分には、キューへの応答の識別やキュー・マネージャーへの応答の識別、メッセージ ID、および関連 ID などの重要な情報が入っています。メッセージ本体は理解されません。

これと対照的に、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。これらのメッセージには、ヘッダーやメッセージ本体の概念はありませんが、プロパティとメソッドをもっています。メッセージ・オブジェクトを理解するには、最初に祖先のフィールド・オブジェクトについての正しい知識が必要です。

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・オブジェクトにはタイプがあります。タイプはプログラミング・オブジェクト・クラス名に対応する、省略されたストリングです。フィールド・タイプは次のとおりです。

表 3. フィールド・オブジェクト・タイプ

タイプ	説明
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	値、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がサポートされる)

表3. フィールド・オブジェクト・タイプ (続き)

タイプ	説明
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) 値、固定配列、または整数の動的配列
長整数	(8 バイト) 値、固定配列、または長整数の動的配列
オブジェクト	値
短整数	(2 バイト) 値、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・オブジェクトは、以下のような各種のメソッド呼び出しをサポートします。

- フィールドの列挙
- 構成フィールド (1 つまたは複数) を (置換オプションによって) コピーする
- フィールド・オブジェクトを比較する
- 構成フィールドとの間での put および get
- フィールド・オブジェクトを検査する (たとえば、'そのオブジェクト内に構成フィールドが含まれているか?')
- バイト配列との間でのダンプおよび復元を行う

dump メソッドと restore メソッドは特に重要です。たとえば、dump メソッドは、リンク上でのメッセージ・オブジェクトの伝送用のデータを提供するために呼び出されます。restore メソッドを使用すると、受信側でそのオブジェクトの再作成が可能です。このように、各オブジェクトはそれぞれ独自の伝送形式を受け持ちます。また、提供されているデフォルト・メソッドをオーバーライドして、カスタムの振る舞いをインプリメントすることができます。同様に、メッセージ・オブジェクトが保管されたり、キューから復元されるときも、これらの 2 つのメソッドを使用します。これらのメソッドをオーバーライドすると、別の振る舞いをインプリメントすることができます。たとえば、メッセージ・オブジェクトは、ダンプ時にデータベースを照会して、伝送前にその値を取り出すことができます。

次の表に、フィールド・オブジェクトとそれらの構成フィールドのプロパティを示します。

メッセージ・オブジェクト

表4. 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 (複数可)	あり	
Hidden		あり
Name		あり
Type	あり	あり
Value		あり

注意を要する 2 つのプロパティがあります。比較の目的のためにフィールドを無視させることのできる *hidden* プロパティと、フィールド・オブジェクトに 属性オブジェクトを関連付ける 能力です。

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、および内容の保護が可能になります。以下の重要なプロパティがあります。

表5. 属性オブジェクトのプロパティ

プロパティ	説明
Authentication:	アクセスを制御する
Encryption:	オブジェクトをダンプするときに内容を保護 (および復元を許可) する
Compression:	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の場合)
Rule:	許可された操作を制御する

メッセージ・オブジェクトは、フィールド・オブジェクトから継承されるので、メッセージおよびコンポーネントのフィールド・オブジェクト・レベルに属性オブジェクトを関連付ける能力を含め、上記の能力すべてが組み込まれます。さらにメッセージ・オブジェクトは、MQSeries Everyplace が生成する UID (固有 ID) を含みます。この UID は、MQSeries Everyplace ネットワーク全体のそれぞれ個々のメッセージ・オブジェクトを一意的に識別し、以下のものから構成されます。

- 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)。

したがってメッセージ・オブジェクトには、祖先のフィールド・オブジェクトから継承されたものに加えて、以下のプロパティがあります。

表6. メッセージ・オブジェクトのプロパティ

プロパティ	説明
発信元のキュー・マネージャー	アプリケーションから新規メッセージを最初に受信したキュー・マネージャーの名前
作成時刻	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
UID	MQSeries Everyplace 固有 ID

別の MQSeries Everyplace キュー・マネージャーあてのメッセージには、その他の情報は必要ありませんが、他のフィールドが含まれるのが普通です。これらの追加フィールドは、以下のようなさまざまな方法で生じます。

- 現行の状況を反映するように MQSeries Everyplace が追加したフィールド
- 特定のメッセージ・サブクラスに関連したフィールド
- メッセージ・オブジェクト・インスタンスに関連したカスタム・フィールド

MQSeries Everyplace は、メッセージング操作とキューイング操作をインプリメントするために、メッセージ・オブジェクトにフィールドを追加します (そして後でそれらを除去します)。たとえば、メッセージがキューにあるとき、メッセージを照会して、それがいつそのキューに put されたかを検出することが可能です。(これは、後述するキュー・ルールを使用して行うことができます)。キュー・マネージャー間でメッセージが送信されているときは、データの再送を実行中であることを示すため、再送フィールドを追加できます。そのほかにも多数の例があります。

一般的なアプリケーション・ベースのメッセージは、基本メッセージ・オブジェクト・クラスの子孫のインスタンスなので、その目的に合った追加のフィールドをもちます (たとえば、送り状番号)。これらの追加フィールドのうち一部は、汎用フィールドであり、多数のアプリケーションに共通です。たとえば、応答先キュー・マネージャーの名前などです。MQSeries Everyplace はこれに応じて、メッセージに存在する可能性があるフィールドのためのサポートを用意しています。これらのフィールドは以下のとおりです。

表7. 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
アクション (Action)	質問、作成、削除などのアクションを指示するために管理で使用される
相関 ID (Correlation ID)	通常は、応答を元のメッセージと相関させるのに使用されるバイト・ストリング
エラー (Errors)	エラー情報を戻すために管理で使用される
有効期限 (Expire Time)	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)
ロック ID (Lock ID)	メッセージをアンロックするのに必要なキー

表7. 用意されているメッセージ・オブジェクト・フィールド (続き)

フィールド名	使用法
メッセージ ID (Message ID)	メッセージの固有 ID
発信元のキュー・マネージャー (Originating queue manager)	メッセージを送信したキュー・マネージャーの名前
パラメーター (Parameters)	管理詳細を渡すために管理で使用される
優先順位 (Priority)	メッセージ伝送を順序付ける相対優先順位
理由 (Reason)	エラー情報を戻すために管理で使用される
応答先キュー (Reply-to queue)	メッセージ応答の宛先となるキューの名前
応答先キュー・マネージャー (Reply-to queue manager)	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
再送 (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
戻りコード (Return code)	管理操作の状況を戻すために管理で使用される
スタイル (Style)	要求 / 応答などからコマンドを区別する
ラップ・メッセージ (Wrap message)	データ保護を確保するためにラップされたメッセージ

どの場合についても、定義済みの定数を使って、フィールド名を単一バイトで伝えることが可能です。一部のフィールドについては、さらに拡張された機能が用意されています。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。

23ページの図4 に示す管理パネルは、メッセージの構造を表しています。ネストされたフィールドの使用は、メッセージ内に複雑なデータ階層が存在する可能性があること、および、管理ツール (およびアプリケーション) が、自己記述構造にドリルダウンできること (これが、セキュリティ設定値によって使用禁止されていない場合) を意味することに注意してください。

メッセージ・オブジェクトは、伝送またはキュー・ストレージの目的のためにバイト配列との間でそれらが逐次化されるとき、基礎の dump メソッドと restore メソッドを使用します。デフォルトでは、フットプリントで保管するために、これは、関連したクラス定義をフローしません。クラス定義が必要であれば、dump および restore を標準逐次化メソッドで上書きすると、クラス定義がフローします。

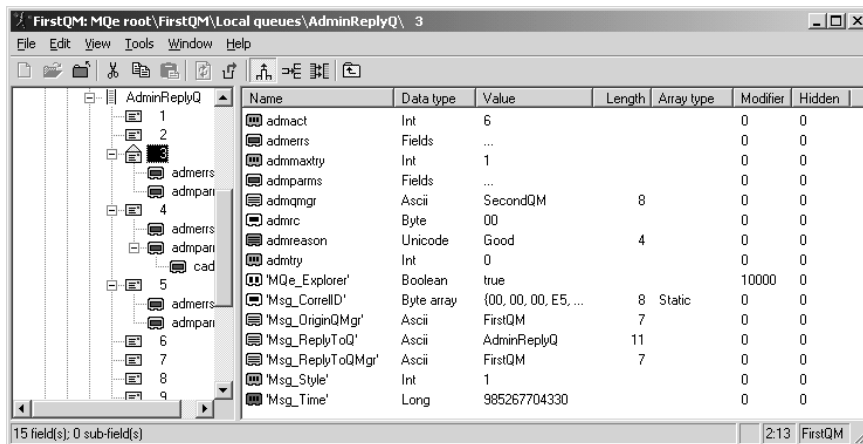


図 4. Mqe_Explorer で表示されるメッセージ構造

ただし、一般に、メッセージ・オブジェクトのインスタスが生成されるそれぞれのキュー・マネージャーで、クラスは独立して使用可能にされます。オブジェクトのインスタスを生成することが（たとえば、伝送中の中間記憶装置について）必要で、かつクラス・ファイルが利用不能であることが認識されている場合には、メッセージを別のクラスにラップすることができます。これは一般に、デフォルトのメッセージ・オブジェクト・クラスです。この手法は、メッセージを保護するために属性オブジェクトを使用した場合にも有用です。このような属性オブジェクトが存在することの意味は、必要なセキュリティー・キーなしではメッセージ内容にアクセスできないということです。ただし、このようなメッセージのラッピングにより、カーネル・メッセージを完全に保護されたままにでき、一方で、ラップされたメッセージを自由にダンプおよび復元することができます。

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}} { ... }
```

それぞれ以下のような意味になります。

- *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。プ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。

メッセージ・オブジェクト

- *Identifier*: フィールド名を、可変長の ASCII スtring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。0 ~ 1,073,741,823 の範囲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もつこと、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。キュー・オブジェクトへのアプリケーションによる直接アクセスは通常は許可されません。その代わりに、キュー・マネージャーがアプリケーション・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せん。名前は長さが無制限の ASCII 文字ストリングですが、³ 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 Everyplace 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。保管されたメッセージは、キューごとに、アダプターを介して永続記憶域にマップされます。MQSeries Everyplace には一定範囲のアダプターが用意されていますが、他のアダプターを作成することもできますし、他のアダプターが他のソースから使用可能な場合もあります。標準アダプターは、MQeDiskFieldsAdapter と呼ばれており、キューをローカル・ファイル・システムにマップし、確実な送達をインプリメントします。もう 1 つのアダプター MQeReducedDiskFieldsAdapter も、キューをファイル・システムにマップしますが、パフォーマンスの高速化、バッファを空にして物理ディスク・サブシステムへの書き込みが行われるほど長時間存続するオペレーティング・システムの信頼性の向上が優先されます。しかし、別のアダプター MQeMemoryFieldsAdapter はキューをメモリーにマップします。このアダプターは、最速のキュー・パフォーマンスを提供しますが、オペレーティング・システムやキュー・マネージャーが再始動された後、メッセージが存在しなくなるという特性をもっています。適切なアダプターを作成することで、メッセー

3. 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 文字という制限も含まれます。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。

ジは、キュー単位でどこにでも保管することができます。ストレージ・メディアの例としては、リレーショナル・データベース、書き込み可能 CD があります。IBM 4690 小売ストア・コントローラーでミラーリングされたファイル・システムを利用するアダプターや、キュー・ストレージに DB2 を活用するアダプターがあります。

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ても、接続されていなくても）使用できます。また、属性オブジェクトによるメッセージ・オブジェクトおよびフィールド・オブジェクトの保護に非常によく似た方法で、キューにセキュリティー属性を設定することもできます。キューのセキュリティーについては、47ページの『セキュリティー』で説明します。ローカル・キューのメッセージへのアクセスは、常に同期に行われます。これは、MQSeries Everyplace が、put または get 操作の完了後に戻るまで、アプリケーションが待機することを意味します。

リモート・キュー

リモート・キューは、リモート・キュー・マネージャーにあるキューへのローカル参照です。ローカル参照はターゲット・キューと同じ名前をもちますが、リモート・キュー定義は、実際のキューを所有するキュー・マネージャーを識別します。リモート・キューにはアクセスに関するプロパティーもあります。たとえば、アクセスのモード（同期または非同期）、セキュリティー特性、および伝送オプションなどです。

MQSeries Everyplace は、リモート・キューを自動的に確立することができます。別のキュー・マネージャーにあるキューへのアクセスが試みられる（たとえば、キューへメッセージを送信するため）と、MQSeries Everyplace はリモート・キュー定義を探します。定義があればそれが使用されますが、定義がない場合は、キュー・ディスカバリーが行われます。特性（認証、暗号化、および圧縮）が発見され、リモート・キュー定義が作成されます。このようなキュー・ディスカバリーは、アクセス可能になっているターゲットに依存して行われます。ターゲットがアクセス可能でない場合、他のなんらかの方法でリモート定義を提供する必要があります。キュー・ディスカバリーが発生すると、キューが同期に使用可能なことが分かっているため、MQSeries Everyplace はアクセス・モードを同期に設定します。

同期 リモート・キューとは、所有するキュー・マネージャーへの通信パスをもつネットワークに接続されている場合にのみアクセスすることができるキューです。ネットワークが確立されていない場合、put、get、および browse などの操作（37ページの『キュー・マネージャー操作』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ー要件を制御します。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 Everyplace が確実に一回だけの送達の責任をもつ必要はなくなります。

非同期 リモート・キューとは、リモート・キューにメッセージを送信できるものの、リモート側でメッセージを検索することが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。これによって、アプリケーションが、デバイスがオフラインのときにキューを操作できるようになります。したがって、これらのキューは、伝送を待ちながら、送信側キュー・マネージャーで一時的にメッセージを保管できるように、メッセージ・ストアにマップされるアダプターをもちます。

ストア・アンド・フォワード・キュー

このタイプのキューは、1 つまたは複数のターゲット・キュー・マネージャー宛先に関連したメッセージを保管します。これには 2 つの主な使用方法があります。最初の用法は、ネットワークでのメッセージの中間ストレージを使用可能にして、それらのメッセージが段階的に宛先に進めるようにします (フォワード役割)。2 番目の用法は、収集待ちのメッセージを保留することです (ホーム・サーバー・キューも参照)。

ストア・アンド・フォワード・キューは、メッセージを保持するキュー・マネージャー名のセットに関連付けられます。これらは、ターゲット・キュー・マネージャーと呼ばれます。これらのターゲット・キュー・マネージャーの 1 つに送られるメッセージは、関係のあるストア・アンド・フォワード・キューに置かれます。ストア・アンド・フォワード・キューは、オプションで、転送側キュー・マネージャー名が設定されることもあります。この名前が設定されている場合、キューはそのすべてのメッセージを、その名前前のキュー・マネージャーに送信しようとします。その名前が設定されていない場合は、キューはただメッセージを保持します。

このタイプのキューは通常 (絶対ではない)、サーバーまたはゲートウェイで定義されます。1 つのキュー・マネージャーに複数のストア・アンド・フォワード・キューが存在することは可能ですが、ターゲット名が重複してはなりません。ストア・アンド・フォワード・キューの内容は、アプリケーション・プログラムには利用不能です。同様に、メッセージ送信アプリケーションは、メッセージ伝送でのストア・アンド・フォワード・キューの存在や役割についても知りません。

ストア・アンド・フォワード・キューのメッセージはアプリケーションには利用不能です。

ホーム・サーバー・キュー

リモート・キューおよびストア・アンド・フォワード・キューでは、ネットワーク経由でメッセージ、つまり、伝送を開始する送信側キューを push します。しかし、ホーム・サーバー・キューでは、メッセージをリモート・キューから pull できます。ホーム・サーバー・キュー定義は、リモート・キュー・マ

ネージャー上のストア・アンド・フォワード・キューを識別します。次にホーム・サーバー・キューは、ストア・アンド・フォワード・キューから、ホーム・サーバー・キューのローカル・キュー・マネージャーを宛先とするメッセージを pull します。1 つのキュー・マネージャーで複数のホーム・サーバー・キュー定義を定義できます。それぞれの定義は、異なるリモート・ストア・アンド・フォワード・キューに関連付けられます。

ホーム・サーバー・キューは通常、デバイスに常駐し、デバイスがネットワークに接続するときは常に、サーバーからメッセージを pull するようにセットアップされます。ホーム・サーバー・キューがサーバーからメッセージを pull すると、メッセージは、正しいターゲット・ローカル・キューに入れられます。そのため、ホーム・サーバー・キューそのものが、アプリケーションがアクセス可能なメッセージをもつことはありません。ネットワークでの流れという観点からすると、サーバーからメッセージを入手する pull メソッドの方が、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に、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。このポーリング間隔は、管理構成オプションの 1 つです。ホーム・サーバー・キューには、クライアント / サーバー・チャンネルでクライアントがメッセージを受信できるようにするうえで、重要な役割があります。クライアント / サーバー接続の性質は、サーバーがデータ転送を開始できないということです。⁴

ホーム・サーバー・キュー上のメッセージはアプリケーションには利用不能です。

管理キュー

管理キューは、キュー・マネージャー（およびそれらの関連オブジェクト）が、ローカルまたはリモート側で構成されるときに使われるメカニズムです。管理キューに送信されたメッセージは、関係のある管理メッセージ・クラスによって処理され、その後、オプションで、応答が発信アプリケーションに送られます。このトピックについては、40ページの『管理』でさらに詳細に説明します。

MQSeries-Bridge 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ために、

4. 代わりの方法は、対等通信チャンネルを使用するか、両方のキュー・マネージャーを、クライアント機能とサーバー機能を両方とも付けて構成することです。

キュー

変換機能が使用されます。MQSeries Everyplace には、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、アダプターによって、メッセージがオペレーティング・システムにより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 が単独で、メッセージおよびキューに変更をログ記録するわけではありません。メディア障害からのリカバリーが必要である場合、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、リカバリー可能なストレージ (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・マネージャーに、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、提供されている以下の 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を次の表に示します。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 8. 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	セキュリティ操作を制御するルール・クラス
Queue_Authenticator	Authenticator クラス
Queue_BridgeName	所有する MQSeries-Bridge 名
Queue_ClientConnection	クライアント接続名
Queue_CloseIdle	すべてのメッセージが伝送されると、Transporter をクローズする
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

表 8. キューのプロパティ (続き)

プロパティ	説明
Queue_Expiry	メッセージの有効期限
Queue_FileDesc	キューの位置およびアダプター
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)
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_QMgrNameList	キュー・マネージャー・ターゲット
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_QTimerInterval	保留メッセージの処理前の遅延
Queue_TargetRegistry	ターゲット・レジストリー・タイプ
Queue_Transporter	Transporter クラス
Queue_TransporterXOR	XOR 圧縮を使用する Transporter
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。

次の MQe_Explorer パネルは、ローカル・キューのプロパティを記述する 4 つのプロパティ・タブのうち、2 つを示しています。使用不可のフィールドは、特定のプロパティがそのクラスのキューに関係がないことを表します。

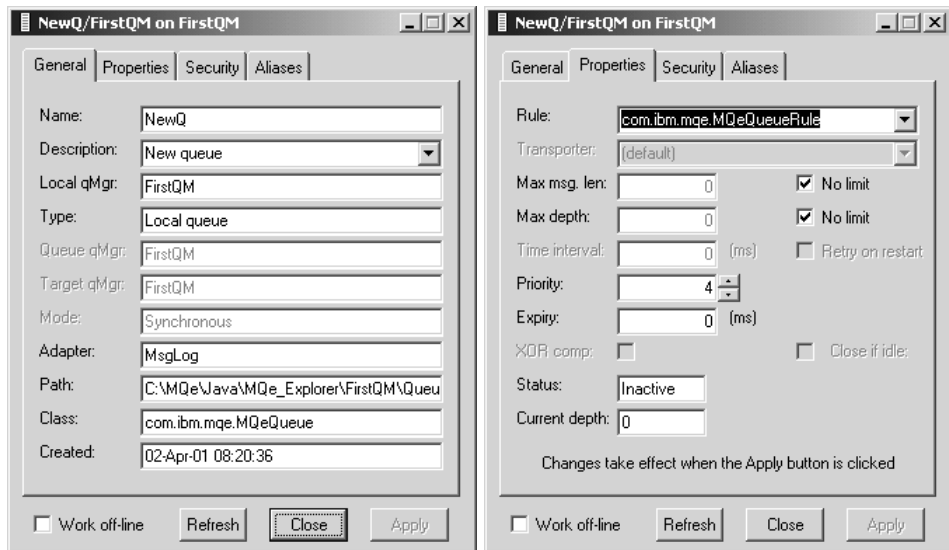


図 5. MQE_Explorer で表示されるローカル・キューのプロパティ

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なりません。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。正しいクラス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、アクション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アクションを制御します。たとえば、他のキューがスプレッドシートを更新したり、テキストから話法へ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ー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、関連するオーセンティケーターと暗号化機能は、許可を受けたメッセージ送信者のみがそのようなメッセージを送信できることを保証し、転送中、メッセージの内容が確実に保護されることになります。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 1.2 では、1 つの Java 仮想計算機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

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

ブリッジ能力をもつように構成されているため、MQSeries ホスト製品および分散製品とメッセージを交換することのできるキュー・マネージャーは、ゲートウェイと呼ばれています。

キュー・マネージャーは、ローカル・キューイングを持つ構成することも、ローカル・キューイングを持たない構成することができます。すべてのキュー・マネージャーは同期メッセージング操作をサポートします。ローカル・キューイングがあるキュー・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。

同期メッセージ送達を使用するか、非同期メッセージ送達を使用するかは、送信側のキュー・マネージャー上のキュー定義の性質によって判別されます。ターゲット・キューに対する同期モードのリモート・キュー定義が存在する場合は、同期送達を使用されます。非同期モードのリモート・キュー定義が存在する場合は、メッセージが伝送を待つ間にローカル・ストレージを提供する定義によって、非同期送達を使用されます。リモート・キュー定義は存在しないものの、そのターゲット・キュー・マネージャー用のメッセージを扱うストア・アンド・フォワード・キューが存在する場合は、非同期送達を使用されます。この場合、ストア・アンド・フォワード・キューは、伝送待ち中のメッセージにローカル・ストレージを用意します。キュー定義が存在しないときは、キュー・ディスカバリーが行われ、ディスカバリーが成功すると、同期メッセージングが使用されます。

同期メッセージングが使用されるか、非同期メッセージングが使用されるかは関係なく、MQSeries Everyplace は、使用可能な接続定義に応じて、直接伝送または間接伝送を使用できます (38ページの『接続』を参照)。直接伝送では、2 つのキュー・マネージャー、すなわち、送信キュー・マネージャーとターゲット・キュー・マネージャーだけが関係します。間接接続では、一連のキュー・マネージャーが関係し、伝送途上でプロトコルの変更がある可能性があります。同期メッセージングで間接伝送が使用されるとき、メッセージングの振る舞いは、直接伝送のケースと変わりません。中間キュー・マネージャーは単に、ソース (送信元) とターゲット (宛先) の間の接続を確立するだけのものです。ただし、非同期メッセージングで間接伝送が使用される場合、伝送メディアは、メッセージがソースからターゲットに移動するときに、メッセージのステージング・ポストになります。そのように使用されるかどうかは、適切な中間キュー・ストレージが、適切なリモート・キュー定義またはストア・アンド・フォワード・キューの形式で、中間キュー・マネージャーに定義済みかどうかに応じて決定されます。

5. MQSeries Everyplace または MQSeries がこの制限を強制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。

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。即座にアプリケーションに戻ります。メッセージを即時に送達できる (または、適切なステージング・ポストに移動することができる) 場合は送信され、できない場合は、ローカルに保持されます。伝送再試行ログは、キュー・マネージャーおよび関係のあるキューに関連したルールによって定義されます (54ページの『ルール』を参照)。非同期送達では、メッセージが MQSeries Everyplace に渡され、それが送達の責任を担うことになっているので、1 回限りの確実な送達サービス品質が提供されます。

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 put します。MQSeries Everyplace は、同期を取ってターゲット・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。即座にアプリケーションに戻ります。メッセージを送達できない場合、送信側のアプリケーションが即時通知を受信します。MQSeries は、同期のケースでは、メッセージ送達の責任を想定しません。

同期と非同期のリモート・キュー定義は、MQSeries Everyplace ネットワーク上で自由に確立することができます。ネットワークに MQSeries メッセージング・キュー・マネージャー (およびそれらに関連したキュー) も含まれるときは、以下のいくつかの制約事項が重要となります。

1. 同期メッセージングは、MQSeries Everyplace ゲートウェイに直接接続されていない MQSeries キュー・マネージャーに対しては可能ではありません (同期メッセージングは、MQSeries メッセージ・チャンネル経由ではサポートされないため)。このことの重大性を最小限にするために、このケースでは同期送達の定義が変更され、ゲートウェイに直接接続された MQSeries キュー・マネージャーへの送達として、再定義されます。このキュー・マネージャーのほかに、MQSeries 非同期メッセージングを使用して、メッセージがその移送を完了できるようにします。
2. MQSeries Everyplace バージョン 1.2 では、MQSeries メッセージング・リモート・キューに対して同期リモート・キュー定義を定義することのみ可能です。これは、上記の (1) の影響を考慮して、ゲートウェイに直接接続された MQSeries キュー・マネージャー上のキューに対する非同期送達が (この定義を使用して) 可能でないことを意味します。ただし、このケースでキューイングが必要な場合は、MQSeries Everyplace ネットワーク内で、2 番目の MQSeries Everyplace キュー・マネージャーを使用することで調整できます (そこで、適切なリモート・キュー定義 (またはストア・アンド・フォワード・キュー) を使用します)。

したがって、非同期メッセージ送達とは、ローカル・アプリケーションが 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、ネットワークが稼働していることが必要ですが、送信側アプリケーション

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。

MQSeries Everyplace では、非同期メッセージングのための 1 回限りの確実な送達サー
ビス品質について、MQSeries メッセージングにおいて持続オプションと非持続オプション
によって行われるような、選択肢は提案されていません。MQSeries Everyplace におい
ては、キュー・ストア・アダプター の選択項目によって、キュー・レベルでの信頼性と
パフォーマンスのトレードオフを行うことができます。その例を以下に示します。

- MQeDiskFieldsAdapter を使用すると、データが安全にディスクに書き込まれてから処
理が続行されます。
- MQeReducedDiskFieldsAdapter を使用すると、データがオペレーティング・システム
に渡されて (そして検索できるようになって) から処理が続行されます。
- MQeMemoryFieldsAdapter は、データをメモリーに保管します。

いずれの場合も、1 回限りの確実な送達がインプリメントされますが、リカバリーは、
選択されたメッセージ・ストアに応じて行われます。パフォーマンスのトレードオフに
ついては、SupportPac EP01 に記述されています。

キュー・マネージャーのプロパティを表9 に示します。

表9. 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
Admin_Class	キュー・マネージャー・クラス
Admin_Name	キュー・マネージャー名
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
QMgr_Description	UNICODE 記述
QMgr_QueueStore	キューのデフォルトの位置およびアダプター
QMgr_Rules	キュー・マネージャー操作のルール・クラス

次のパネルは、キュー・マネージャーのプロパティを記述する 2 つのプロパティ・
タブのうち、1 つを示しています。

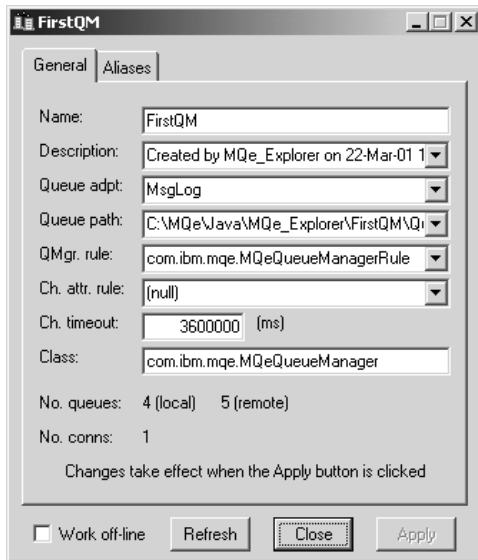


図 6. MQE_Explorer によって表示されるキュー・マネージャーのプロパティ

キュー・マネージャー構成

キュー・マネージャーは、キュー・マネージャーがロードされる前に、MQSeries Everyplace によって確立された環境で実行されます。キュー・マネージャーそのものは、その構成情報をそのレジストリー（詳細は、51ページの『レジストリー』で説明します）に保管します。キュー（メッセージを含んでいる）そのものは、キュー・ストアに保管されます。

MQSeries Everyplace 環境は多くの方法で確立することができます。すなわち、API による呼び出しによって、MQSeries Everyplace と一緒に配布されているユーティリティーによって、または MQE_Explorer などの管理ツールによって、確立できます。これらのツールの多くは、初期設定ファイルの環境パラメーターを取り込みますが、これは完全にオプションです。環境に関する主要パラメーターを次の表に示しています。使用するセクション名は、この情報を初期設定ファイルで表すために使用される規則に従っています。

表 10.

セクション名	プロパティ	説明
[Alias]	(クラス別名定義)	クラス名が必要な場所どこでも別名を使用できる
[ChannelManager]	MaxChannels	許可される同時のクライアント / サーバー・チャネルの最大数

表 10. (続き)

セクション名	プロパティ	説明
[Listener]	Network	着信のクライアント / サーバー・チャンネル接続要求について listen するアダプターおよびポート番号
	Listen	結果として得られた接続を処理するために使用されるアダプター
	TimeInterval	クライアント / サーバー・チャンネルのタイムアウト
[MQBridge]	(ブリッジの初期設定パラメーター)	
[MQe_Explorer]	(保管されたアドレッシング情報)	MQe_Explorer は、このキュー・マネージャーへの着信接続の方法を記述する情報を保管する
[Permission]	(許可されたコマンド)	許可されたチャンネル・コマンド、アダプター・クラス、およびファイル記述子マップ
[PreLoad]	キュー・マネージャーが初期設定されるときにロードされるクラス	これは、アプリケーション・クラスのロード用の 1 つのメカニズムを提供する
[QueueManager]	Name	キュー・マネージャーの名前
[Registry]	DirName	レジストリー位置
	LocalRegType	レジストリーのタイプ (ファイルまたは専用)
	PIN	レジストリー PIN (またはプロンプト要求)
	CertReqPIN	認証 PIN (またはプロンプト要求)
	KeyRingPassword	鍵リング・パスワード (またはプロンプト要求)
	CAIPAddrPort	認証局 IP アドレス

以下の 4 つの図では、MQe_Explorer を使用して新規のキュー・マネージャーを作成するときの主要タブを表示しています。前提となる事前の環境はありません。また既存の初期設定ファイルも必要ありません。この結果は、構成データを初期設定ファイルに保管した、実行キュー・マネージャーとなります。初期設定ファイルに構成データが保管されるので、キュー・マネージャーはそのファイルをオープンするだけで再始動することができます。

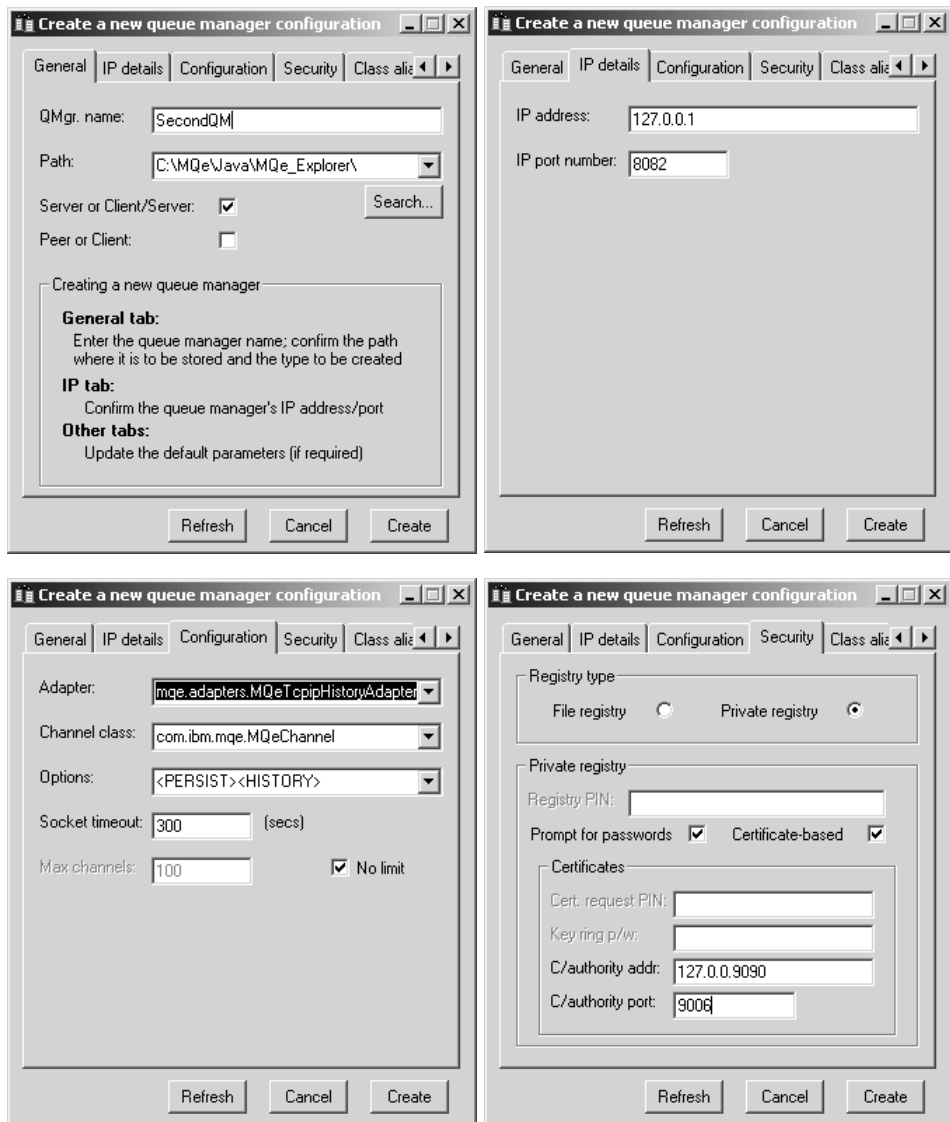


図7. MQE_Explorer によるキュー・マネージャーの作成

キュー・マネージャーの作成では実際には、そのキュー・マネージャーの IP アドレスが認識されている必要はありません。ただし、MQE_Explorer は、このキュー・マネージャーと話す他のキュー・マネージャーを構成するために使用するための情報を取り込みます。着信のクライアント / サーバー接続要求を listen するためには、ポート番号が必要です。

チャンネル・リスナーのためにアダプター構成データが必要です (たとえば、チャンネル・マネージャーの時間間隔や最大チャンネルなど)。MQe_Explorer は、他のキュー・マネージャーを構成することができるように、以前と同様、その他の情報を取り込みます。

この例では、証明書をベースとする認証によって保護レジストリーが構成されました (47ページの『セキュリティー』を参照)。PIN とパスワードは使用不可ですが、これは、それらが必要になったときに後で MQe_Explorer が PIN とパスワードのプロンプトを出すからです。

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、次のようなメソッドを使用してキュー・マネージャーのサービスを介してメッセージにアクセスします。すなわち、*get*、*put*、*browse*、*wait*、*listen*、および *delete* メソッドが使用されます。これらの操作の多くでは、パラメーターの 1 つとして、フィルターを指定します。フィルターは、等価なものを探すために突き合わせるフィールド・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。また、ほとんどのメソッド呼び出しには、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も組み込まれます。

get 操作は、キューからメッセージを破壊的に除去します。フィルターによって課せられた条件によって、メッセージは優先順位で検索され、そしてその中で、キューへの到着時間順に検索されます。そのため、他のすべてのことが等しければ、到着した最初のメッセージが、検索される最初のメッセージとなります。*get* は、単一ステップとしても、2 つのステップ操作としても、使用可能です。2 ステップ操作は、MQSeries Everywhere からアプリケーションへメッセージが渡されるときに、メッセージを失う可能性がないことが重要となるような場合に使用するものです。最初に、確認 ID (その値はアプリケーションによって選択される) を指定して、*get* が発行されます。この操作により、アプリケーションのメッセージが取得されますが、キューからそのメッセージをただちに削除する代わりに、キュー上に隠されます。続いて、元のメッセージ UID が指定された確認操作では、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します。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、メッセージはリカバーされます。*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制御して、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・オブジェクトが検索されますが、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の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、個別にロックされることも、フィルターによって識別されるグルー

キュー・マネージャー

ブ単位でロックされることもあります。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。ブラウズ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、フィルターを使用して、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11 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 11. 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
ブラウズ (±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インプリメントされるので、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 1.2 に提供されている MQSeries Everyplace MQSeries Bridge は、 確実 / 不確実な put、不確実な get、および (ロックなしの) 不確実なブラウズだけしかサポートしません。

接続

MQSeries Everyplace ネットワークの接続形態およびそれを介したアクセスは、接続オブジェクトで定義されます。これらの定義は、各キュー・マネージャーでローカルに保管されます。これらの作成、変更、および破棄は、用意されている標準の管理機能を使って行います。

接続オブジェクトは一般に、リモート・キュー・マネージャーへのアクセスを定義します (したがって、リモート・キュー・マネージャー定義と呼ばれることがあります)。次の表にそれらのプロパティを示しています。

表 12. 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
Admin_Name	キュー・マネージャー名
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 (使用履歴など)
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 / 接続の代替名
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

以下の管理パネルは、接続定義に関連したタブの 2 つを示しています。

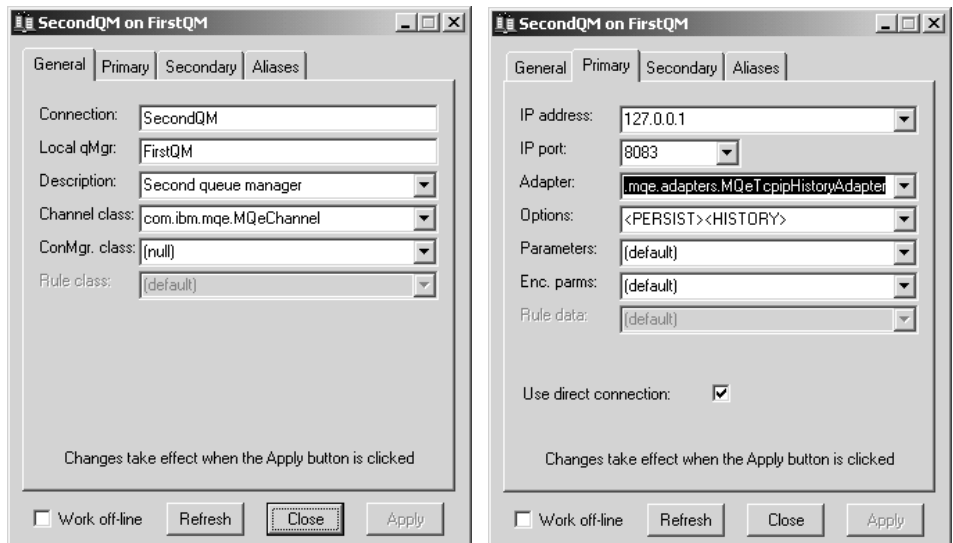


図 8. MQ Explorer で表示される接続定義

データは、オプション、パラメーター、およびエンコードされたパラメーターの形式で、選択した通信アダプターに渡すことができます。

間接接続を指定することができます。この場合には、MQSeries Everyplace は、他のキュー・マネージャー (チェーンできます) 経由で接続をルーティングします。また、プロトコルは途中で変わることがあります。間接接続は、MQSeries Everyplace ネットワークに対する単一のエントリー・ポイントをデバイスがもつことができるようにするとき、特に有用です。

接続

接続オブジェクトは、着信対等通信チャネルのリスナーを定義するときにも使用します。

ほとんどの MQSeries オブジェクトに関して、接続に別名を定義できます。ローカル接続 (ローカル・キュー・マネージャーの名前と一致する名前をもつ接続として定義されたもの) は、ローカル・キュー・マネージャーそのものの別名を定義するために使用されます。

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 MQe_Explorer⁷、つまり、システム管理のためのグラフィカル・ユーザー・インターフェースを提供する管理ツールを使用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13 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 ブリッジでは、このように MQeAdminMsg のサブクラスを使用します。

表 13. 管理メッセージ・クラス

管理メッセージ・クラス	使用法
MQeAdminMsg	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

7. MQe_Explorer は、バージョン 1.2 には組み込まれていませんが、ワールド・ワイド・ウェブ (WWW) の MQSeries Everyplace サイト (<http://www.ibm.com/software/ts/mqseries/everypace>) から、無料でダウンロードできます。

表 13. 管理メッセージ・クラス (続き)

管理メッセージ・クラス	使用法
MQeAdminQueueAdminMsg	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	MQSeries 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	MQSeries 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	MQSeries へのブリッジの管理
MQeMQBridgesAdminMsg	MQSeries-Bridge のリストの管理
MQeMQQMGrProxyAdminMsg	MQSeries キュー・マネージャーのブリッジ表現の管理
MQeMQBridgeQueueAdminMsg	MQSeries-Bridge キューの管理

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、一般的に管理メッセージは、表14 で示されているような構造です。

表 14. 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		管理アクションの最大試行回数
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・クラスおよびアクションに特有の、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

管理

表 14. 管理メッセージの一般構造 (続き)

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_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アクションを示すコマンド (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。

選択的な管理

管理へのアクセスは、管理キュー上にあるオーセンティケーター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、提供されているオーセンティケーターは、それらすべてが同じローカル・ユー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、チャンネル上のオーセンティケーター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・ユーザーを区別することができ、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。より洗練されたオーセンティケーターでは、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 (47ページの『セキュリティ』を参照)。キューに関連したルールを活用して、同様の方法でアクションを許可したり拒否したりできます (54ページの『ルール』を参照)。

モニターおよび関連処置

管理には、オブジェクトの作成や変更以上の広範囲の作業が含まれることがあります。たとえば、システムをモニターしたり、キューがいっぱいになったときに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、適切な処置をとるなど、エラー状態を処理したりする作業です。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 (54ページの『ルール』を参照)。

動的チャネル

MQSeries Everyplace は、キュー・マネージャー間の通信を、動的チャ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。メッセージ・ダンプ・メソッドをオーバーライドすることで、メッセージごとにこのプロトコルをカスタマイズすることができます。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・チャネルの対を使用します。MQSeries クラスター・メッセージ・チャネル には、MQSeries Everyplace 動的チャネルと類似した特性がありますが、重要な相違も多数あります。

動的チャネルは 2 つのキュー・マネージャー間の論理接続であり、データの送受信の目的で確立されます。同じ通話者間であっても、複数のチャネルが並行して存在する場合があります。これらのチャネルには、認証、暗号化、圧縮、使用されるトランスポート・プロトコルなどの特性があります。これらの特性はプラグ可能であり (異なるチャネルでは、異なるバージョンが使用されることもあります)、そのため、各チャ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。

- **オーセンティケーター:** NULL か、ユーザーまたはチャネル認証を実行できるオーセンティケーター・オブジェクトのどちらか
- **チャネル:** 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 オブジェクトのどちらか
- **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 オブジェクトのどちらか
- **宛先:** このチャネルのターゲット (たとえば SERVER.XYZ.COM など)

オーセンティケーターは通常、チャネルのセットアップ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 MQeXorCrytor 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。この暗号化機能は安全ではありませんが、データを変更して表示不可にします。対照的に、MQe3DESCrytor は triple-DES をインプリメントします。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。他にもオーセンティケーター、圧縮機能、および暗号化機能が用意されています。48ページの表15 を参照してください。

チャネル確立は、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・

動的チャネル

マネージャーに渡されます。このようにして、中間のストア・アンド・フォワード・キューおよびリモート・キュー定義を活用することができます。

チャネルは、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、あるキュー・マネージャーから別のキュー・マネージャーに正常に渡されたことと、肯定応答を確認することによって実現されます。通信障害が発生した場合、肯定応答が受け取られないと、メッセージは場合によっては再送されますが（一回限りの送達は、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・インターフェースにマップするのに使用されます。チャネルは、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で実行されます。同様に、キューはフィールド・ストレージ・アダプターを活用して、メモリーまたはファイル・システムなどのストレージ・サブシステムにインターフェースします。アダプターは MQSeries Everyplace に対し、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・ネットワーキング・サポートは、デバイスのオペレーティング・システムによって処理されます。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しようとする場合、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、ユーザーに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。接続が確立されるまでは、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、オペレーティング・システムが使用するのに適したダイ

ヤルアップ接続プロファイルを使用可能にしておく必要があります。MQSeries Everyplace バージョン 1.2 には、ダイヤルアップ・ネットワーキングのための明示的サポートはありません。

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。アプリケーションは、トレースを直接に呼び出すこともでき、また、新しいメッセージを追加したり、既存のトレース・メッセージを変更したりできます。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、別の機能を付加したり、出力を別の方法でフォーマットしたり送達したり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もできます。

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。ロギングを開始して、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をリカバリーするために使うことはできません。

メッセージ送達

MQSeries Everyplace ネットワークは、接続されたキュー・マネージャーであり、ゲートウェイを含むこともあります。それらは、複数の物理ネットワークにまたがることや、その間でメッセージをやり取り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われます。

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、その宛先キュー・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ターゲット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る、支援記憶装置に置かれます。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、その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

メッセージ送達

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もつチャンネルが識別 (または確立) され、ターゲット・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。ターゲット・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もつストア・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、ターゲット・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 サーバーへ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が必要になります。

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非同期 MQSeries Everyplace キューへ送ることができます。

セキュリティー

MQSeries Everyplace には、統合された一群のセキュリティー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 セキュリティー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・セキュリティー - 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ー - 開始するキュー・マネージャーとターゲット・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ー - 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。

- オーセンティケーター
- 暗号機能
- 圧縮機能
- キー
- ターゲット・エンティティー名

これらのオブジェクトの使用法は、MQSeries Everyplace セキュリティー機能のカテゴリに応じて異なりますが、どの場合でも、メッセージ・オブジェクトに付加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行われます (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 1.2 でサポートされているアルゴリズムについては、48ページの表15 に示しています。

表 15. 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	WTLS ミニ認証
	検証 Windows NT/2000、AIX、または Solaris 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・セキュリティー

ローカル・セキュリティーは、MQSeries Everyplace メッセージ (または MQeFields か、MQeFields より下のもの) データをローカルに保護します。これを実現するために、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フレーズを入力することにより) 適切な鍵 を作成および設定して、その鍵を属性に明示的に付加します。その属性が MQSeries Everyplace メッセージに付加されます。MQSeries Everyplace には、ローカル・セキュリティーのセットアップを援助する MQeLocalSecure クラスが用意されていますが、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フレーズの鍵を管理することは、ローカル・セキュリティー・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ー

キュー・ベースのセキュリティーは、同期メッセージ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ー

同期でのキュー・ベースのセキュリティーを使うことにより、アプリケーション側は、メッセージ・セキュリティー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ーのレベルを判別します (および永続的ストレージのセキュリティーのレベルも判別します)。

メッセージを送信する場合、ターゲット・キューのセキュリティー特性は、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、ターゲット・キュー・マネージャーでターゲット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。ターゲット・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、ターゲット・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ー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、セキュリティー・レベルが過度に思える場合 (たとえば、保護が必要な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、必要でない場合は、自動的に破棄されます。メッセージは、常にターゲット・キューの特性で定義されたセキュリティー・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ーは、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、流されるデータの機密性を保護します。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、このデータへ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ー

非同期メッセージングは、putMessage の実行時にターゲット・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になります。このときには、そのターゲット・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ではリモート・キュー定義が必要です。これがないと、ターゲット・キューの特性を判別できません。

非同期のケースでは、発信元とターゲット (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する場面 (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ため)、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ることを確認する場面に、メッセージ・レベルのセキュリティーを使用する必要があります。

セキュリティー

キュー・ベースのセキュリティーをメッセージ・レベルのセキュリティーと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、その必要はありません。

メッセージ・レベルのセキュリティー

メッセージ・レベルのセキュリティーでは、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ーは、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ターゲット・キューに入れるときに、その属性を提供する必要があります。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、ターゲット・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・セキュリティーの場合と同様に、メッセージ・レベルのセキュリティーでは、メッセージ・オブジェクトで属性のアプリケーションを活用します。発信側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` メソッドを使用してメッセージ・データを保護します。受信側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・データをリカバリします。

MQSeries Everyplace には、メッセージ・レベルのセキュリティーのために、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・セキュリティーの場合と同様に、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。MQeMAttribute 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、強度暗号化機能を使ってメッセージの機密性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、メッセージの機密性を保護します。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、WTLS 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 / 暗号化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鍵が生成されている
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、基本的なストアであり、各キュー・マネージャーごとに 1 つずつ存在します。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、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

ミニ認証に基づく相互の認証を使うキュー・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

セキュリティ

されています。相互の認証というと、2人のユーザー（複数のユーザー）間での認証を通常は思い浮かべますが、一般に、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に至るまで、認証のターゲット 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が人間である可能性を除外するものではありませんが、これは、アプリケーションによって選択されたマッピングになります。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。キュー・マネージャーは、認証可能なエンティティとして、発信側の可能性もありますし、ミニ認証に依存するサービスのターゲットである可能性もあります。さらに、MQSeries Everyplace では、ミニ認証に基づくオーセンティケーター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（キュー・マネージャー）か、一群の認証可能なエンティティ（キュー・マネージャーと、認証に記されたオーセンティケーターを使うそれぞれのキュー）をもつことができます。

専用レジストリーと信任状

各認証可能なエンティティに、それぞれの信任状があると便利です。その場合、以下の2つのことに取り組む必要があります。1つ目は、信任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 では、認証可能なエンティティが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を用意しています。認証可能なエンティティの専用信任状を安全に管理できるようにする専用レジストリー（基本レジストリーの下部）と、一群の公開信任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部）が備えられています。専用レジストリーは、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、秘密鍵のような専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専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、専用オブジェクトが専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の提供によって、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マップすることができます。

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。どちらの場合にも、登録が起動され、新しい信任状が作成されて、認証可能なエンティティ

ィーの専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 鍵のペアを生成すること、秘密鍵を専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、新しい信任状として、認証可能なエンティティの専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の信任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製

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の公開信任状 (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、最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、そのサービスにおいて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製サービスが提供されます。これにより、必要と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、専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。専用レジストリー・サービスは、新しい認証可能なエンティティの信任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、公開信任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 を使用して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。これを、専用レジストリー自動登録要求を満たすように構成することができます。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

セキュリティー

行サービスをセットアップして管理し、注意深く制御された一群のエンティティー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ー群を管理する
- ミニ認証の発行
- WAP WTLS ミニ認証のリポジトリ管理

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、エンティティー名と登録アドレスを登録し、一回限り使用する認証要求 PIN を定義することにより、指定されたエンティティー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通知されます (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 が通知されるのと同様の方法)。次に専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、セットアップ時にこの認証要求 PIN を提供するように構成することができます。専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 (示されたエンティティー名と、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す。

ミニ認証発行サービスで発行される一連のミニ認証は、発行サービス独自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、有効期限切れなどの理由で)、有効期限が切れたミニ認証が保存されます。

セキュリティー・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ー・マネージャーによってインプリメントすることができます。この方式を使うと、セキュリティー・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
- ファイル記述子のマッピング
- チャネル・コマンドの処理

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 の振る舞い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ることもできます。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。

- オーセンティケーター
- 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-Bridge のルール

このルール・クラスには、MQSeries-Bridge 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 Everyplace 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。
- サーバーのインスタンスを生成した後に、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 キュー (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。デフォルトのルールは、問題を表示するだけです。
- MQSeries Everyplace メッセージを MQSeries メッセージに変換する方法 (およびその逆)。MQSeries Everyplace メッセージと MQSeries メッセージ間でメッセージ変換を行う変換機能は、MQeRule クラスから派生しません。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に動作し、メッセージで形式変換が必要なときに呼び出されます。

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 / サーバーおよび対等通信の操作をサポートできます。クライアント は、サーバーとの通信を開始することができます。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。対等通信の操作では、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イングの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16 には、それぞれのコンポーネントと、接続スタイルとの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・モードでも操作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 16. 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	あり		
クライアント / サーバー	あり	あり	あり
対等機能	あり		
サーバー	あり	あり	あり
サブレット	あり	あり	

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・マネージ

ヤーか、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ので、スタイルが重要となります。

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き、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なければなりません。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 つの着信接続要求に対してだけです。したがって、別のそのような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。

着信接続要求のターゲット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、サーバーにある独立した 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。通常、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が必要な場合、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あてたメッセージは、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でないことに注意してください。)

着信接続のターゲットは、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、対等通信接続か、またはクライアント / サーバー接続のどちらかを開始する機能、また、サーバーまたは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・リスナーは、異なるポート番号をもたなければなりません。

クラス

MQSeries Everyplace では、特定の機能向けのクラスの選択が用意されており、特定のアプリケーション要件を満たすように、MQSeries Everyplace の振る舞いをカスタマイズすることができます。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しています。表17 には、そのような発展性についてまとめています。クラスは、明示的に識別することもできますし、別名を使用して識別することもできます。

表 17. 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 キュー・マネージャーが、クライアント (または対等機能) として操作されるように構成されている場合、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、それぞれに独自のキュー・マネージャーが必要であり、それぞれが固有な名前をもつ必要があります。

MQSeries Everyplace キュー・マネージャーがサーバーとして構成される場合は、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・システムを補うため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観点からは、MQSeries キューおよびキュー・マネージャーを、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すことができます。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、ターゲット・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、通常と同じく 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 のルーティングが定義され (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。その代わりに、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必要があります。

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 と他のメッセージング・システムとの間で、1 つまたは複数のオプション・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 1.2 では、このようなブリッジ・タイプは 1 つだけサポートされています。すなわち、MQSeries Everyplace ネットワークと MQSeries ネットワークとの間のインターフェースとなる *MQSeries-Bridge* です。このブリッジでは、1 つまたは複数の MQSeries キュー・マネージャーへ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、その逆に流すことが可能になります。MQSeries Everyplace の現行バージョンでは、サーバーごとに 1 つのブリッジを使用し、それぞれ

MQSeries への接続

を複数の *MQSeries* キュー・マネージャー・プロキシー (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・プロキシー定義は、*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 つまたは複数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* サーバー接続を使うことができます。またオプションで、ユーザー出口やポートなど、異なるプロパティ・セットを使うこともできます。

1 つのゲートウェイが、多数のリスナーを扱うことができます。そのような場合、そのゲートウェイを使って *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。

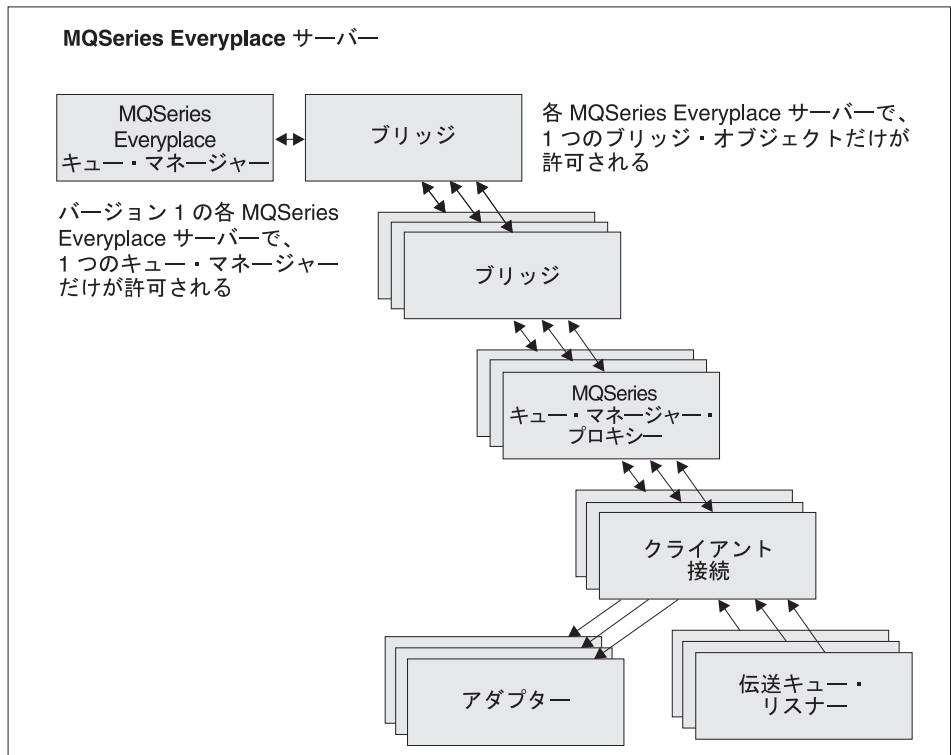


図9. MQSeries Bridge オブジェクト階層

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 つまたは複数のブリッジ・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、メッセージをターゲット・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た MQSeries Everyplace メッセージを、直接または間接的に、任意の MQSeries キュー・マネージャーへディスパッチすることができます。ブリッジ・オブジェクトには、表18 に示されているプロパティがあります。

表 18. ブリッジ・オブジェクトのプロパティ

プロパティ	説明
Bridgename (ブリッジ名)	ブリッジ名のリスト
Run state (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、およびその他のゲートウェイ・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。そのようにゲートウェイ・オブジェクトが開始 (または停止) された場合、そのアクションは、そ

MQSeries への接続

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) にも適用されます。ブリッジ・オブジェクトには、表19 に示されているプロパティがあります。

表 19. ブリッジのプロパティ

プロパティ	説明
Class (クラス)	ブリッジ・クラス
Default transformer (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 (またはその逆へ) 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)。
Heartbeat interval (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、基本的なタイミング単位。
Name (名前)	ブリッジ・オブジェクトの名前
Run state (実行状態)	状況: 実行中または停止状態
Startup rule class (始動ルール・クラス)	ブリッジ・オブジェクトの始動時に使われるルール・クラス
MQSeries Queue Manager Proxy Children (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もできます。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、変換機能ルールをターゲット・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 の両方のターゲット・キューに当てはまります。

MQSeries キュー・マネージャー・プロキシは、1 つの MQSeries キュー・マネージャーに固有のプロパティを保持します。プロキシのプロパティについては、表20 に示されています。

表 20. MQSeries キュー・マネージャー・プロキシのプロパティ

プロパティ	説明
Class (クラス)	MQSeries キュー・マネージャー・プロキシのクラス
MQSeries host name (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に使う 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。

表 20. MQSeries キュー・マネージャー・プロキシのプロパティ (続き)

プロパティ	説明
MQSeries queue manager proxy name (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
Name of owning bridge (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
Run state (実行状態)	状況: 実行中または停止状態
Startup rule class (始動ルール・クラス)	MQSeries キュー・マネージャー・オブジェクトの始動時に使われるルール・クラス
Client Connection Children (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すべての Client Connection オブジェクトのリスト

ゲートウェイ接続サービス定義には、MQSeries キュー・マネージャーへ接続するときに必要な詳細情報が保持されています。接続のプロパティについては、表21 に示されています。

表 21. クライアント接続サービスのプロパティ

プロパティ	説明
Adapter class (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
Class (クラス)	Bridge クライアント接続サービスのクラス
Max connection idle time (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 password* (MQSeries パスワード)	Java クライアントで使われるパスワード
MQSeries port* (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に使う 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 receive exit class* (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、データを出口コードに渡すことができるように、関連付けられたストリングがある。
MQSeries security exit class* (MQSeries セキュリティ出口クラス)	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出口を突き合わせるときに使う。この出口には、データを出口コードに渡すことができるように、関連付けられたストリングがある。
MQSeries send exit class* (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、データを出口コードに渡すことができるように、関連付けられたストリングがある。

表 21. クライアント接続サービスのプロパティ (続き)

プロパティ	説明
MQSeries user ID* (MQSeries ユーザー ID)	Java クライアントで使われるユーザー ID
Client connection service name (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
Name of owning queue manager proxy (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
Startup rule class (始動ルール・クラス)	ブリッジ・クライアント接続サービス・オブジェクトの始動時に使われるルール・クラス
Sync queue name (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
Sync queue purger rules class (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
Run state (実行状態)	状況: 実行中または停止状態
Name of owning Bridge (所有 Bridge の名前)	このクライアント接続を所有する Bridge オブジェクトの名前
MQ XmitQ Listener Children (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、「MQSeries Using Java」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の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用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・プロパティに応じて、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、メッセージを確認する 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、接続プールを使つ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22 に示されているプロパティがあります。

表 22. リスナーのプロパティ

プロパティ	説明
Class (クラス)	リスナーのクラス
Dead letter queue name (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
Listener state store adapter (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
Listener name (リスナー名)	メッセージを提供する MQSeries XMIT キューの名前
Owning client connection service name (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
Run state (実行状態)	状況: 実行中または停止状態
Startup rule class (始動ルール・クラス)	リスナー・オブジェクトの始動時に使われるルール・クラス
Transformer class (Transformer 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
Undelivered message rule class (未配布メッセージ・ルール・クラス)	MQSeries から MQSeries Everyplace へのメッセージを送達できない場合のアクションを決定するルール・クラス
Seconds wait for message (メッセージ待機秒数)	例外的状況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、状態情報を保管します。

伝送キュー・リスナーでは、MQSeries リモート・キューが MQSeries Everyplace ローカル・キューを参照することができます。また、MQSeries ローカル・キューを参照する MQSeries Everyplace リモート・キューを作成することもできます。これらの MQSeries Everyplace リモート・キュー定義は、MQSeries-Bridge キュー と呼ばれています。これらの定義を使用して、MQSeries キューのメッセージを get、put、および browse できます。

MQSeries への接続

MQSeries-Bridge キュー定義には、以下の属性を含むことができます。

表 23. MQSeries-Bridge キューのプロパティー

プロパティー	説明
Alias names (別名)	キューの代替名
Authenticator (オーセンティケーター)	ヌルにする必要がある
Class (クラス)	オブジェクト・クラス
Client connection (クライアント接続)	使用するクライアント接続サービスの名前
Compressor (圧縮機能)	ヌルにする必要がある
Cryptor (暗号機能)	ヌルにする必要がある
Expiry (期限切れ)	変換機能へ渡される
Maximum message size (最大メッセージ・サイズ)	ルール・クラスへ渡される
Mode (モード)	同期にする必要がある
MQ queue manager proxy (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 bridge (MQSeries-Bridge)	メッセージを MQSeries へ送るブリッジの名前
Name (名前)	リモート MQSeries キューが MQSeries Everyplace に認識されている名前
Owning queue manager (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
Priority (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)
Remote MQSeries queue name (リモート MQSeries キュー名)	リモート MQSeries キューの名前
Rule (ルール)	キューの操作で使われるルール・クラス
Queue manager target (キュー・マネージャー・ ターゲット)	キューを所有する MQSeries キュー・マネージャー
Transformer (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
Type (タイプ)	MQSeries-Bridge キュー

注: 暗号化機能、オーセンティケーター、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-Bridge キ

キューへ渡されるときまで、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-Bridge キューがメッセージを MQSeries システムへ渡すときは、これらのセキュリティ・レベルは適用されません。この転送時には、クライアント接続上のセキュリティ送信出口および受信出口が使われます。キュー・レベルのセキュリティが保たれているかどうかについては、検査されません。

MQSeries-Bridge キューは同期だけであるため、非同期アプリケーションは、MQSeries-Bridge キューにメッセージを送信するときの中間ステップとして、MQSeries Everyplace ストア・アンド・フォワード・キューとホーム・サーバー・キューの組み合わせか、または非同期リモート・キュー定義を使用することが必要です。

アプリケーションは、その他の MQSeries Everyplace リモート・キューのように、MQQueueManager クラスの `putMessage`、`browseMessages`、および `getMessage` メソッドを使用して、MQSeries-Bridge キューを利用します。これらの呼び出しでのキュー名パラメーターは、MQSeries-Bridge キューの名前であり、キュー・マネージャー名パラメーターは、MQSeries キュー・マネージャーの名前です。ただし、このキュー・マネージャー名がローカル MQSeries Everyplace サーバーで受け入れられるためには、チャンネル名を含め、すべてのパラメーターにヌルを指定した、この MQSeries キュー・マネージャー名をもつ接続定義が存在しなければなりません。

注: `getMessage` および `browseMessages` を MQSeries-Bridge キューで使用する場合には、制約事項がいくつかあります。MQSeries リモート・キュー定義を指す MQSeries-Bridge キューからメッセージを `get` または `browse` することはできません。また、MQSeries-Bridge キューの `get` でゼロ以外の Confirm ID を使用することもできません。つまり、MQSeries-Bridge キューでの `getMessage` 操作では、確実な送達を提供されません。`get` 操作を確実にする必要があるときは、伝送キュー・リスナーを使用して、MQSeries からメッセージを転送することが必要です。

MQSeries-Bridge 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。40ページの表13 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、またはターゲット・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 値の適切なマッピングを使用して、メッセージを MQSeries 形式に変換するとともに、サブクラスの意味を表すために特定のデータを追加することができます。

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利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 MQSeries 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、MQSeries 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 使用すると、ご使用のアプリケーションで、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、MQeMQMsgObject 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します)。

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 メッセージの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 メッセージ本体は元の MQSeries Everyplace メッセージとして再構成され、それが次に 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 オブジェクトに入れます。MQSeries メッセージ本体は、単純なバイト・フィールドとして扱われ、これも MQeMQMsgObject オブジェクトに入れます。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振る舞いは、以下のことを意味します。

- 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 put 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 管理メッセージを、MQSeries キュー・マネージャーへ送信することはできません。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 とは別に存在させることができますが、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 つの製品が同時に必要になります。MQSeries Everyplace は、以下にまとめられている性質を含む互換性をもつ、既存の MQSeries ネットワークに組み込むことが可能です。

アドレッシングと命名について:

- キュー・マネージャー / 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:

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る。

チャネル:

MQSeries Everyplace ゲートウェイで、MQSeries クライアント・チャネルを使用する。

メッセージング交換と内容について:

- MQSeries Everyplace と MQSeries との間のメッセージの交換であること
- メッセージ・ネットワークが見えないこと (MQSeries Everyplace または 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

MQSeries Everyplace バージョン 1.2 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、そして通信に関する考慮事項のほかに、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、

MQSeries への接続

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。

- クラスター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 / セグメント化はありません
- ロード・バランシング / ウォーム・スタン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

スケー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る場合にのみ、確実にメッセージの転送が行われます (34ページの『キュー・マネージャー構成』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。

第7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。Java 言語と C 言語の 2 つの言語がサポートされます。

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。クラス、メソッド、およびプロシージャの詳細は、「*MQSeries Everyplace for Multiplatforms* プログラミング・リファレンス」に説明されています。MQSeries Everyplace プログラミングの例は、「*MQSeries Everyplace for Multiplatforms* プログラミング・ガイド」に記載されています。

Palm 用の C サポートは、Palm デバイスで使用するための MQSeries Everyplace 機能のサブセットへのアクセスを提供しています。これらのクラスおよびプロシージャの詳細と、プログラミングのガイドは、「*MQSeries Everyplace Native Client Information*」で説明されています。

第8章 MQSeries Everyplace を開始するには

MQSeries Everyplace は、MQSeries Everyplace メッセージングおよびキューイングのソリューションを開発、配置、および管理するために必要なツールを、トータルに提供する製品のファミリーです。このファミリーは以下のものから構成されています。

1. *MQSeries Everyplace* ライセンス・プロダクト (IBM から物理メディアとして提供されるか、または <http://software.boulder.ibm.com/dl/mqsem/mqsem-p> からの Web ダウンロードとして、使用可能です)。このライセンス・プロダクトには以下のものが組み込まれています。

- MQSeries Everyplace Java クラス
- ヘルパー・クラス
- アプリケーション・ソース・コード例
- ユーティリティー
- 解説書
- ライセンス情報

物理的なプログラム・プロダクトには、特定のプラットフォームでの開発以外 (実動) の使用のためにこの製品を使用するための証書も含まれています。大規模なマシンでの使用や、MQSeries-Bridge で使用する場合は、さらに capacity units を購入することが必要です。

2. MQSeries Everyplace SupportPacs (<http://software.boulder.ibm.com/dl/mqsem/mqsem-p> (上記)、または <http://www.ibm.com/software/mqseries/everyplace> から、Web ダウンロードとして使用可能)。これらは、このライセンス・プロダクトの欠くことのできない補足であり、たとえば、以下を組み込んでいます。

EAP1: MQSeries Everyplace - Palm OS 用のデバイス・コード

Palm OS 上での MQSeries Everyplace バージョン 1.0.1 アプリケーション開発のための、C プログラミング言語サポート。(このコードは、製品 CD のファイル eap1.zip にも含まれています。)

EP01: MQSeries Everyplace - パフォーマンス・レポート

各種のクライアント・プラットフォームでの MQSeries Everyplace パフォーマンスを分析しています。

ES01: MQSeries Everyplace - 管理ツール (MQExplorer v1.0)

MQSeries Everyplace キュー・マネージャーについて簡便なグラフィカル管理を使用可能にする、すべての Java プラットフォーム用の汎用ツール

ES02: MQSeries Everyplace - エクスプローラー (MQExplorer v1.2)

Microsoft Windows のオペレーティング・システムをサポートするために特別に開発された MQSeries Everyplace 管理ツール

MQSeries Everyplace を開始するには

MQSeries Everyplace SupportPacs 中の管理ツールは、アプリケーション開発とロールアウトのすべての段階で重要な役割を果たします。これらの管理ツールは、このライセンス・プロダクトに組み込まれているユーティリティよりも高性能なものであり、入門、構成、パイロット・ネットワークの検査、および実動システムを管理する際に、必須の援助機能となります。

MQSeries Everyplace の使用

MQSeries Everyplace には幅広い使用法が与えられているので、この製品のインストール、構成、配置は、MQSeries ファミリーの他のメンバーと同じ方法では行なわれません。ここで基礎となる概念は、企業による MQSeries Everyplace の採用において、一般に以下の 3 つの段階があるということです。

1. 開発およびプロトタイピングの段階

初期の確認、開発、およびプロトタイピングの段階では、MQSeries Everyplace 製品は、インストールに使用可能であり、無料で使用することができます (ただし、IBM MQSeries Everyplace 開発ライセンスの条項に従います)。MQSeries Everyplace アプリケーションは、MQSeries Everyplace Java クラスおよび C ルーチンの機能を使用して、開発されます。これらのアプリケーションは多様な方法でパッケージすることができます。たとえば、次のとおりです。

- MQSeries Everyplace キュー・マネージャーは、同じ Java 仮想計算機にランチされ、共通キュー・マネージャーを共用する、1 つまたは複数のアプリケーションをもつデーモンとして、セットアップすることができます。
- このアプリケーションには、MQSeries Everyplace をインストールしていないマシンで、独自のキュー・マネージャーを各 JVM にランチして、アプリケーションが実行される、必須の MQSeries Everyplace クラスが組み込まれています。
- このアプリケーションは、ターゲット・マシンに存在する MQSeries Everyplace クラスを使用します。

開発ライセンスには、IBM からのサポートは含まれていません。ただし、アプリケーション開発中とそれ以降のサポートは、配置ライセンスに提供されています (下記を参照)。

2. 配置段階

配置段階は、開発したアプリケーションのロールアウトと使用を表すので、この製品を使用するには、IBM MQSeries Everyplace ライセンスの条件のもとに、capacity units が必要になります。クラスは、IBM からの同意を受けて、アプリケーションとともに配布されます。あるいは、すでにクラスの使用資格をもっているユーザーに配布されます。それ以外の場合は、必要なクラスをユーザー自身で用意することが必要です。

3. 管理段階

続いて、ネットワーク内で MQSeries Everyplace キュー・マネージャーがアク

タイプになると、キュー・マネージャーを検査および管理するためのツールが必要になります。MQSeries Everyplace 用のサポートは、「IBM プログラムのご使用条件」の条項のもとに提供されます。

このような採用のライフ・サイクルがあることが、各プラットフォームでのサポート・レベルの多様性の理由付けとなります。capacity units (および Category 3 SupportPacs) をもつ MQSeries Everyplace の場合、IBM では以下のような区別をしています。

- インストールおよびアプリケーション開発がサポートされるプラットフォーム
 - インストールまたはアプリケーション開発・使用 (または、その両方) に関する問題報告書が受け入れられる
- アプリケーション配置が許可されるが、直接にはサポートされないプラットフォーム
 - 問題報告書を、サポートされるプラットフォームで複製する必要がある
- アプリケーション配置がサポートされるプラットフォーム
 - アプリケーション配置の結果得られた問題報告書が受け入れられる

経験の紹介

MQSeries Everyplace を開始する方法は数多くあります。この製品とその概念に精通するための実践的な方法は、キュー・マネージャーを立ち上げて実行し、その後に簡単な MQSeries Everyplace ネットワークを体験してみることでということが経験上分かっています。その後、簡単なアプリケーションを作成します。それが、製品の詳細を深く知るための賢明な準備となります。初期の段階では、このような方法は一般に、MQSeries ファミリーの他のメンバーを検討するうえでは役立ちません。後の段階になって、ブリッジの機能が興味の対象となると、この知識が重要になります。

このような戦略に留意して、新規ユーザーは以下のアプローチを採用するようお勧めします。

「MQSeries Everyplace for Multiplatforms 入門」(本書) に提示されている概念の基本を理解してから、以下を実行します。

1. Microsoft Windows オペレーティング・システムを実行するマシンにアクセスできれば、MQe_Explorer、SupportPac ES02 (MQe_Explorer v 1.2) をダウンロードします。事前にこのライセンス・プロダクトをインストールしておく必要はありませんが、インストールしないと、ライセンスの条項によって、開発使用に制限されます。
2. 『ES02 の最初の使用: MQe_Explorer』に記載されている特定の指示に従います。

ES02 の最初の使用: MQe_Explorer

MQe_Explorer と一緒に、最初のキュー・マネージャーを構成する迅速な方法を説明している包括的な「ユーザー・ガイド」が配布されます。この資料は一般に、MQe_Explorer 管理者だけでなく、より広い範囲のユーザーにお勧めします。この資料には、MQSeries

MQSeries Everyplace を開始するには

Everyplace 操作の重要な例を分かりやすく示すいくつかのサンプル・スクリプトが含まれているので、役立ちます。SupportPac には、2 つの実行可能な MQe_Explorer バージョンが組み込まれています。

MQe_ExplorerX.exe

このバージョンは、.exe ファイル内に、操作に必要なすべての MQSeries Everyplace クラスを組み込んでいます。MQe_ExplorerX.exe は、マシンに MQSeries Everyplace をインストールしていない状態で、実行されます。これは、初めてのユーザーには理想的です。

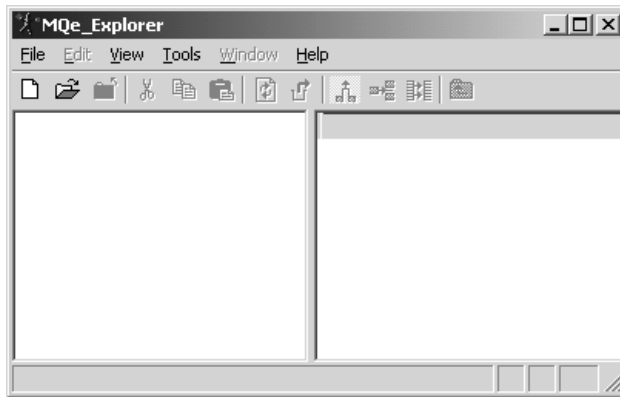
MQe_Explorer.exe


このバージョンは、事前にインストールされた MQSeries Everyplace に依存します。この利点は、最新レベルの MQSeries Everyplace ライブラリーを取り込むことと、サイズが小さいことです。このバージョンは、開発者と管理者向けです。

キュー・マネージャーを容易に作成できることの例として、以下で MQSeries Everyplace の能力を簡単に説明します。

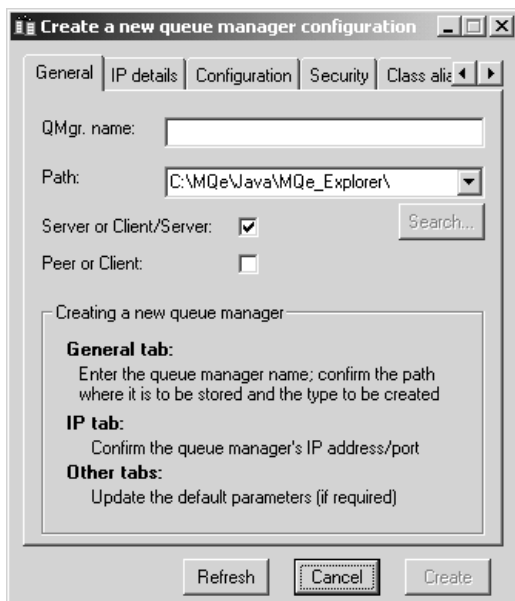
1. MQe_ExplorerX.exe アイコン  をダブルクリックします。

保管されているオプションが検出されなかったことを表すメッセージが表示されます。**OK** をクリックします (このメッセージは、もう現れません)。以下の管理ウィンドウが表示されます。

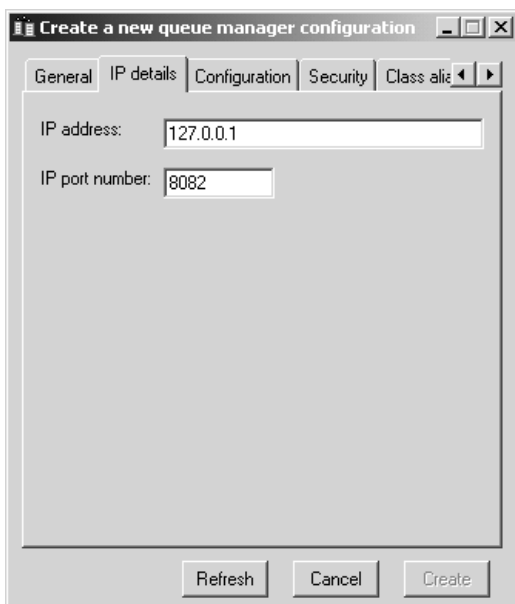


2. ツールバーの新しいアイコン  をクリックします。

これで新規キュー・マネージャーが作成されます。以下のウィンドウが表示されます。

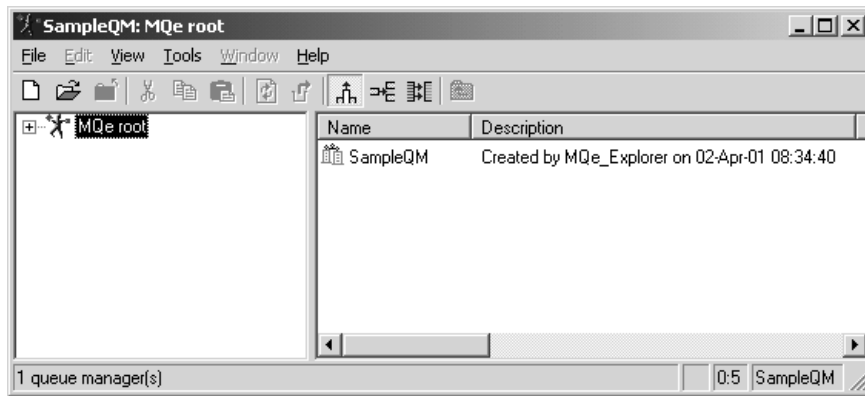


3.
 - a. キュー・マネージャー名を入力します (たとえば、SampleQM)。
 - b. 「IP 明細 (IP details)」タブを選択します。

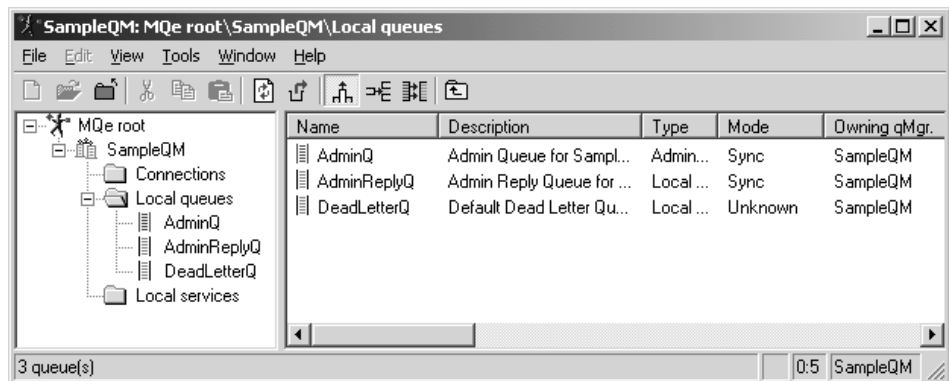


MQSeries Everyplace を開始するには

4.
 - a. IP アドレスを入力します。このアドレスは、使用するマシンの IP アドレスですが、この段階で入力する有効な値は、127.0.0.1 (ローカル・ホスト) です。
 - b. 「作成 (Create)」 ボタンをクリックします。作成された初期設定ファイル (この名前は、このキュー・マネージャーに将来アクセスするために必要です) を識別するメッセージが表示されます。
 - c. **OK** をクリックすると、以下のウィンドウが表示されます。



5. サーバー・キュー・マネージャーが作成され、自身の JVM で実行されます。ポート 8081 で、着信クライアント / サーバー・チャンネル接続要求を listen しています。左側の画面区画のツリーで + 記号がすべて展開され、ウィンドウと画面区画がサイズ変更されると、4 つのキューが作成されるのが分かります。



6. さらに体験してみたい場合は、「MQE_Explorer User Guide」の指示に従ってください。キュー、接続、メッセージを作成できるだけでなく、MQSeries Everyplace ネット

| トワークを完成することもできます。この同じキュー・マネージャーにアプリケーションをロードすることができます。それらのアプリケーションを MQe_Explorer と並行して実行できます。

付録. 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あり、米国以外の国においては本書で述べる製品、サービス、またはプログラムを提供しない場合があります。日本で利用可能な製品、サービス、およびフィーチャーについては、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。IBM 製品、プログラム、またはサービスに代えて、IBM の有効な知的所有権またはその他の法的に保護された権利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 によって明示的に指定されたものを除き、他社の製品と組み合わせた場合の操作の評価と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

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。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、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。

本書に対して、周期的に変更が行われ、これらの変更は、文書の次版に組み込まれます。IBM は、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。

特記事項

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 United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire
England
SO21 2JN

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。

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 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い場合は、本書の索引か、「*IBM Dictionary of Computing*」、New York: McGraw-Hill, 1994 を参照してください。

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。

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。

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。

インスタンス (instance). インスタンスはオブジェクトの 1 つ。クラスのインスタンスが生成されて、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。

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、多数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。

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

オーセンティケーター (authenticator). メッセージの送信側と受信側を検査するプログラム。

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。

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ってのみ、データをアクセスしたり処理できる。

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・キューイング・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。

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。

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。特定のクラスのインスタンスを生成することにより、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。

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。

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のことである。

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。

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働するクライアント・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。

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、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。

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。

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。

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。

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。

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。

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。

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。

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。

ブリッジ (bridge). MQSeries Everyplace と他のメッセージング・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。

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。

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。

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。

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。

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。

MQI チャンネル (MQI channel). MQI チャンネルは、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。

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。

参照文献

関連資料:

- *MQSeries Everyplace for Multiplatforms* 最初にお読みください, GC88-8656
- *MQSeries Everyplace for Multiplatforms* プログラミング・リファレンス, SC88-8655-02
- *MQSeries Everyplace for Multiplatforms* プログラミング・ガイド, SC88-8654
- | • *MQSeries Everyplace for Multiplatforms Native Client Information*, GC34-5883
- | • *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 44
圧縮 47
アプリケーション、MQSeries Everyplace 13
アプリケーションのロード 58
暗号化 47
イベント・ログ 45
インターフェース、セキュリティ 54
インターフェース、プログラミング 71
インターフェース、MQSeries への 59
エンティティ、認証可能な 51
オブジェクト、メッセージ 18
オブジェクト、MQSeries Everyplace 18
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 17
概要 1
確実なメッセージ送達 70
カスタマイズ 54
環境、ソフトウェア 3
管理、MQSeries Everyplace での 40
管理メッセージ 40
機能 13
キュー、ストア・アンド・フォワード 26
キュー、リモート 25
キュー、ローカル 24
キュー、MQSeries Everyplace 24
キュー、MQSeries-Bridge 27
キューのルール 55
キュー・ベースのセキュリティ 48
キュー・マネージャー 8, 9, 56
キュー・マネージャー、MQSeries Everyplace 30
キュー・マネージャー操作 37
キュー・マネージャーのルール 56
キュー・マネージャー・プロキシ・オブジェクト 62
クライアント / サーバー接続 57
クライアント / サーバー・チャンネル 9

クライアント、MQSeries 8
クライアント・チャンネル 17
クラス、MQSeries Everyplace 58
ゲートウェイ、MQSeries Everyplace 17
公開レジストリー 53
構成 54
構成、例 10
構成例 10
互換性、MQSeries との 69
顧客の要件 14

[サ行]

サーバー、MQSeries 8
サポートされるオペレーティング・システム 3
自動登録 52
商標 82
ストア・アンド・フォワード・キュー 26
製品の概念 17
セキュリティ、キュー・ベースの 48
セキュリティ、メッセージ・レベルの 50
セキュリティ、ローカル 48
セキュリティ、MQSeries Everyplace 47
セキュリティ・インターフェース 54
接続、クライアント / サーバー 57
接続、対等通信 57
接続スタイル 56
接続スタイル、複数 57
説明 1
前提条件 3
前提条件となる知識 vii
専用レジストリー 52
操作、キュー・マネージャー 37
属性のルール 55
ソフトウェア環境 3

[タ行]

対象 vii
対等通信接続 57
対等通信チャンネル 9
ダイヤルアップ接続の管理 44
ダンプ・データ・フォーマット 23
チャンネル 9

チャンネル、クライアント 17
チャンネル、動的 17, 43
チャンネル・マネージャー 56
チャンネル・リスナー 56
通信 56
デバイス、MQSeries Everyplace 17
同期メッセージング 46
動的チャンネル 9, 17, 43
読者 vii
特記事項 81
トレース、MQSeries Everyplace の 45

[ナ行]

認証可能なエンティティ 51
認証の複製 53
ネットワーク、MQSeries 59
ネットワーク、MQSeries Everyplace 45, 59

[ハ行]

パーベイスブ・メッセージング viii, 8
発行サービス、ミニ認証 53
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 45
フォーマット、ダンプ・データの 23
複数接続スタイル 57
複製、認証の 53
ブリッジ、MQSeries 59
ブリッジ・オブジェクト 61, 62
プログラミング・インターフェース 71
分散メッセージング viii, 8
変換機能 67
ホーム・サーバー、MQSeries Everyplace 26
ホーム・サーバー・キュー 26
ホスト・メッセージング viii, 8
本書について vii
本書の対象読者 vii

[マ行]

ミニ認証 51
ミニ認証発行サービス 53
メッセージ、管理 40
メッセージ送達、確実な 70
メッセージの変換 67
メッセージング、同期 46

メッセージング、非同期 45
メッセージング、MQSeries 7
メッセージ・オブジェクト 18
メッセージ・レベルのセキュリティ 50
モニター 42

[ヤ行]

要件、顧客 14
用語 viii

[ラ行]

リスナー・オブジェクト 65, 66
リモート・キュー 25
ルール、MQSeries Everyplace 54
レジストリー 51
レジストリー、公開 53
レジストリー、専用 52
レジストリー、MQSeries Everyplace 17
ローカル・キュー 24
ローカル・セキュリティ 48
ロード、アプリケーションの 58

M

MQeAttribute 50
MQeMTrustAttribute 50
MQSeries Everyplace アダプター 44
MQSeries Everyplace アプリケーション 13
MQSeries Everyplace オブジェクト 18
MQSeries Everyplace 管理 40
MQSeries Everyplace キュー 24
MQSeries Everyplace キュー・マネージャー 30
MQSeries Everyplace クラス 58
MQSeries Everyplace ゲートウェイ 17
MQSeries Everyplace デバイス 17
MQSeries Everyplace ネットワーク 45, 59
MQSeries Everyplace のセキュリティ 47
MQSeries Everyplace ルール 54
MQSeries Everyplace レジストリー 17, 51
MQSeries Integrator viii, 7
MQSeries Workflow viii, 7
MQSeries クライアント 8
MQSeries サーバー 8
MQSeries との互換性 69
MQSeries ネットワーク 59

MQSeries ファミリー 7
MQSeries へのインターフェース 59
MQSeries メッセージング 7
MQSeries-Bridge 9, 59
MQSeries-Bridge キュー 27
MQSeries-Bridge のルール 55

S

SPI 71



Printed in Japan

GC88-8653-02



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