

MQSeries® Everyplace 多平台版



# 介绍

版本 1.2



MQSeries® Everyplace 多平台版



# 介绍

版本 1.2

## 请注意

在使用本文档和它支持的产品之前，请确保已阅读第69页的『附录. 通告』中的一般信息。

### 许可证警告

MQSeries Everyplace 版本 1.2 是一个可以让用户编写 MQSeries Everyplace 应用程序并创建运行环境的工具箱。

在生产环境中部署本产品或使用它的应用程序之前，请确保有必需的许可证。

要在指定的服务器平台上使用 MQSeries Everyplace（非代码开发和测试目的），必须获得生产单位使用授权（在“资质证明”文档中记录，按照已发布的生产单位和定价组表，可以支持 MQSeries Everyplace 的使用）才可以得到许可在每台机器和每次机器升级时使用程序。

要在指定的客户机平台上使用产品（非代码开发和测试目的），需要设备平台使用授权（在“资质证明”文档中记录，可以支持 MQSeries Everyplace 的使用）。这些许可证没有授予用户使用 MQSeries Everyplace Bridge 的权利，也不允许它们在 IBM 发布的 MQSeries Everyplace 定价组列表中指定的服务器平台上运行，这些许可证也可以通过下面提到的 URL 从 Web 上获得：

有关这些限制的详细信息，请参考 <http://www.ibm.com/software/mqseries>。

### 第三版（2001 年 5 月）

除非在新版本中另有说明，否则这个版本适用于 MQSeries Everyplace 版本 1.2 和所有后续发行版及修订版。

本文档还会继续被新的和改进的信息所更新。如需最新的修订版，请参阅 MQSeries 系列书库的 Web 页面：<http://www.ibm.com/software/mqseries/library/>。

© Copyright International Business Machines Corporation 2000, 2001. All rights reserved.

# 目录

关于本书 . . . . .	v	事件日志 . . . . .	39
本书面向的读者 . . . . .	v	消息传递 . . . . .	39
必备知识 . . . . .	v	异步消息传递. . . . .	39
本书中使用的术语 . . . . .	v	同步消息传递. . . . .	39
		安全性 . . . . .	40
更改摘要 . . . . .	vii	MQSeries Everyplace 本地安全性 . . . . .	41
本版本 (GB84-0269-02) 的更改 . . . . .	vii	MQSeries Everyplace 基于队列的安全性 . . . . .	41
前一版本 (GB84-0269-01) 的更改 . . . . .	vii	消息级安全性. . . . .	42
		注册表 . . . . .	44
第1章 概述 . . . . .	1	MQSeries Everyplace 可认证的实体 . . . . .	44
		专用注册表和凭证 . . . . .	44
第2章 软件环境 . . . . .	3	自动注册 . . . . .	45
支持的平台 . . . . .	3	公用注册表与证书复制. . . . .	45
Java 环境 . . . . .	4	注册表服务的应用程序使用 . . . . .	45
存储量需求 . . . . .	4	缺省迷你证书发行服务. . . . .	45
		安全性接口 . . . . .	46
第3章 MQSeries 系列 . . . . .	7	定制. . . . .	46
MQSeries 主机和分布式产品 . . . . .	8	规则. . . . .	46
MQSeries Everyplace . . . . .	9	连接样式 . . . . .	47
		对等连接 . . . . .	48
第4章 产品需求 . . . . .	13	客户机 / 服务器连接 . . . . .	48
能力 . . . . .	13	多个连接样式. . . . .	49
应用程序 . . . . .	13	类 . . . . .	49
用户要求 . . . . .	14	应用程序装入. . . . .	50
第5章 产品概念 . . . . .	15	第6章 MQSeries Everyplace 和 MQSeries	
介绍 . . . . .	15	网络 . . . . .	51
消息对象 . . . . .	16	与 MQSeries 的接口 . . . . .	51
转储数据格式. . . . .	20	消息转换 . . . . .	57
队列. . . . .	21	功能. . . . .	58
队列管理器 . . . . .	26	兼容性 . . . . .	58
队列管理器配置 . . . . .	29	确保传递 . . . . .	59
队列管理器操作 . . . . .	32		
连接. . . . .	33	第7章 编程接口. . . . .	61
管理. . . . .	34		
管理消息 . . . . .	35	第8章 MQSeries Everyplace 入门 . . . . .	63
选择性管理 . . . . .	36	使用 MQSeries Everyplace . . . . .	64
监控和相关操作 . . . . .	37	获得经验 . . . . .	65
动态通道 . . . . .	37	首先使用 ES02: MQe_Explorer . . . . .	65
适配器 . . . . .	38		
拨号连接管理. . . . .	38	附录. 通告 . . . . .	69
跟踪. . . . .	38		

商标 . . . . .	70	索引 . . . . .	75
词汇表 . . . . .	71	将您的意见发送到 <b>IBM</b> . . . . .	77
文献目录 . . . . .	73		

---

## 关于本书

本书是对 MQSeries Everyplace 多平台版产品（在本书中通常称为 MQSeries Everyplace）的一般介绍。它涵盖了本产品的概念和与其它 MQSeries 产品之间的关系。

关于 MQSeries Everyplace API 及如何使用它来创建 MQSeries Everyplace 应用程序的详细信息，请参阅 *MQSeries Everyplace for Multiplatforms Programming Reference* 和 *MQSeries Everyplace for Multiplatforms Programming Guide*。

有关对 MQSeries Everyplace 多平台版 使用其它编程语言的相关信息，请参阅 *MQSeries Everyplace for Multiplatforms Native Client Information*

有关 MQSeries Everyplace 多平台版安装过程的信息，请参阅 *MQSeries Everyplace 多平台版首先阅读*

本文档还会继续被新的和改进的信息所更新。如需最新的修订版，请参阅 MQSeries 系列书库的 Web 页面：<http://www.ibm.com/software/ts/mqseries/library/>。

---

## 本书面向的读者

本书适用于对在轻型设备如传感器、电话、个人数字助理 (PDA) 和笔记本电脑上使用安全消息传递感兴趣的人，以及需要扩展 MQSeries Everyplace 消息传递网络范围的人。

---

## 必备知识

阅读本文档不需要预备知识，但最好能对一些安全消息传递概念有初步了解。

如果不了解这些概念，阅读以下 MQSeries 书籍将有所帮助：

- *MQSeries An Introduction to Messaging and Queuing*

该书在联机 MQSeries 书库的“书籍”部分以软拷贝形式存在。可从 MQSeries Web 站点获得，URL 地址是：<http://www.ibm.com/software/ts/MQSeries/library/>

---

## 本书中使用的术语

下列术语贯穿本书：

## **MQSeries 系列**

指以下 MQSeries 产品:

- **MQSeries Workflow** 通过使涉及人和应用程序的商业过程自动化来简化整个企业之间的集成。
- **MQSeries Integrator** 是一种功能强大的消息代理软件, 它提供实时和基于智能规则的消息传递以及内容转换和格式化。
- **MQSeries 消息传递**通过商业质量的消息传递提供了从台式机到大型机的任意连接性, 支持超过 35 个平台。

## **MQSeries 消息传递**

指以下消息传递产品组:

- **分布式消息传递:** MQSeries Windows NT、AIX<sup>®</sup>、AS/400<sup>®</sup>、HP-UX 和 Sun Solaris 版和其它平台
- **主机消息传递:** MQSeries OS/390<sup>®</sup> 版
- **普及型消息传递:** MQSeries Everyplace

## **MQSeries**

指以下三个 MQSeries 消息传递产品组:

- 分布式消息传递
- 主机消息传递
- 工作站消息传递

## **MQSeries Everyplace**

指第三个 MQSeries 消息传递产品组, 普及型消息传递。

## **设备平台**

能够将 MQSeries Everyplace 只作为客户机运行的小型计算机。

## **服务器平台**

能够将 MQSeries Everyplace 作为服务器或客户机运行的大小不限的计算机。

## **网关**

运行包括 MQSeries 网桥功能的 MQSeries Everyplace 程序的大小不限的计算机。



---

## 更改摘要

本节描述了对这个版本的 *MQSeries Everyplace* 多平台版介绍的更改。自本书上一版本以来的更改由更改左侧的竖线标记。

---

### 本版本 (GB84-0269-02) 的更改

---

### 前一版本 (GB84-0269-01) 的更改

添加了以下信息:

- 有关在 AIX 和 Solaris 上使用 MQSeries Everyplace 的信息。
- 存储器需求。
- 读者意见表。



---

## 第1章 概述

MQSeries Everyplace 是商业消息传递产品 MQSeries 系列中的一个成员。它的设计不仅满足轻型设备（如传感器、电话、个人数字助理 (PDA) 和笔记本电脑）的消息传递需要，而且还支持移动连接和由于使用脆弱通信网络而产生的需求。它维护标准 MQSeries 服务质量、提供仅一次确保传递，并可以与系列的其它成员交换消息。由于许多 MQSeries Everyplace 应用程序运行于因特网防火墙的保护范围之外，它还提供完善的安全性能力。

轻型设备要求消息传递子系统对其系统资源妥善利用，因此 MQSeries Everyplace 提供适合于其用户集合的定制的功能和界面，它们并不打算提供与该系列其它成员完全相同的能力。另一方面，它的确包括一些特定的功能以支持其特殊的用户类，例如全面的安全性规定、消息对象、同步和异步消息传递、远程队列访问，以及消息推和拉。

MQSeries Everyplace 还设计成与 IBM 普及型计算系列中的其它成员以及 Websphere Application Server 的其它组件进行良好集成。



---

## 第2章 软件环境

---

### 支持的平台

MQSeries Everyplace 只能在某些服务器平台上直接安装。要将程序和 Java 类传送到其它平台上，必须使用合适的下载或文件传送程序（未提供）。

#### 带有安装支持的直接支持的平台

可以在以下平台上使用内置工具安装产品。

- Windows NT<sup>®</sup> v4
- Windows<sup>®</sup> 2000
- Windows 95/98/ME
- AIX 版本 4.3
- Sun Solaris 版本 7 或 8
- Linux Intel Kernel 2.2（使用 zip 文件安装）。
- HP-UX 11.0（使用 zip 文件安装）

#### 没有安装支持的直接支持的平台

支持在以下平台上测试与部署 MQSeries Everyplace，但只支持通过从另一平台进行文件传送执行的安装。

- 在 HP Jornada 设备上运行的 WinCE 2.1（型号 680 或 820）
- 在 Psion 设备上运行的 EPOC 32 位，发行版 5（5MX Pro 或 NetBook）
- 在 Palm V 和 IBM Workpad C3 上运行的 PalmOS，V3.0 或更高版本
- 带有 Java 的 IBM 4690 OS

#### 间接支持的平台

可以使用以下平台，但只有在其 Java 环境与直接支持的平台上的 Java 环境完全兼容时才受到支持。只能在上面列出的一个经过测试的平台上检查问题。

- 运行 Kernel 2.2 的 zSeries 上的 Linux
- iSeries
- OS/2
- EPOC（不在以上列出的设备中）
- WinCE（不在以上列出的设备中）

## 软件环境

- QNX Neutrino
- Pocket PC
- PalmOS (不在以上列出的设备中)
- 任何运行『Java 环境』中列出的 Java 环境的其它平台

---

## Java 环境

需要以下 Java 运行时环境之一:

- IBM Java 运行时 (JVM 1.3 或更新), 包括 Java Micro Edition
- 任何经 Sun Java 认证的 Java (V1.1 或更新)<sup>1</sup>

**注:** 为使服务可用, Java 需要与在以下一种平台上经过测试的 Java 完全兼容

- 运行 Windows CE 操作系统的 HP Jornada 设备 (型号 680 或 820)
- 运行 EPOC 操作系统的 Psion 设备 (5MX Pro 或 NetBook)
- 上面直接支持列表中列出的一种服务器平台

MQSeries 网桥操作需要 MQSeries Java 类。应该检查运行 MQSeries Java 类版本所需的 Java 的级别。

---

## 存储量需求

下表显示了执行 MQSeries Everyplace 安装所需的存储量。

表 1. 执行安装所需的存储量

操作系统	所需的存储量
Windows NT (文件系统 = NTFS)	26Mb
AIX	29Mb
Solaris	27Mb

下表显示了安装后 MQSeries Everyplace 文件所需的存储量。

表 2. MQSeries Everyplace 所需的存储量

操作系统	所需的存储量
Windows NT (文件系统 = NTFS)	9.5Mb
AIX	11Mb

---

1. 如果在支持 JIT (Just In Time) 编译器的 Sun JVM 下运行安装程序, 可能会遇到问题。如果使用 Sun JVM, 我们建议使用以下命令禁用 JIT 编译器: `java -Djava.compiler=NONE install`

表 2. MQSeries Everyplace 所需的存储量 (续)

操作系统	所需的存储量
Solaris	10Mb





## 第3章 MQSeries 系列

MQSeries 系列包括许多产品并提供了一系列能力，如图1所示

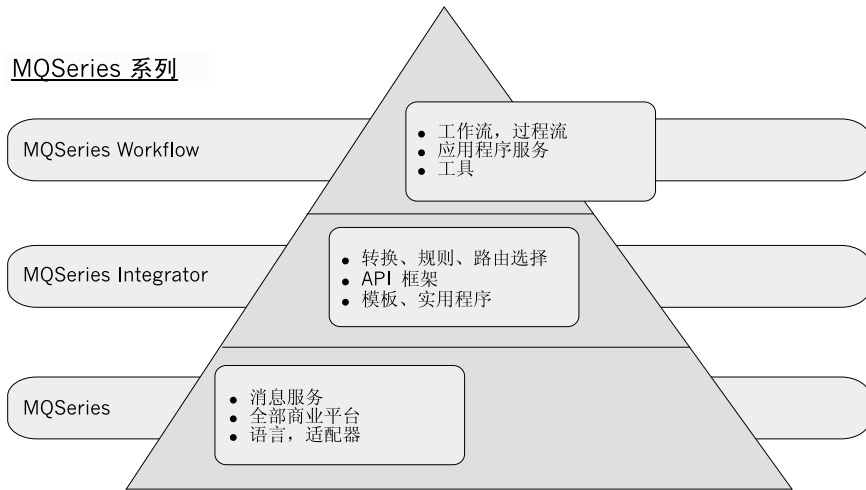


图 1. MQSeries 系列

- **MQSeries Workflow** 通过使涉及人和应用程序的商业过程自动化来简化整个企业的集成。
- **MQSeries Integrator** 是一种功能强大的消息代理软件，它提供实时和基于智能规则的消息传递以及内容转换和格式化。
- **MQSeries 消息传递**通过商业质量的消息传递提供了从台式机到大型机的任意连接性，支持超过 35 个平台。

MQSeries Workflow 和 MQSeries Integrator 产品都利用 MQSeries 消息传递层提供的连接性。

MQSeries 系列的消息传递由 MQSeries 和 MQSeries Everyplace 产品共同提供，每个产品都设计成支持一种或多种硬件服务器平台和 / 或相关操作系统。考虑到平台能力的多样性，这些单独的产品被组织成产品组，以反映一些公共的功能和设计。有三个这样的产品组：

- **分布式消息传递：** MQSeries Windows NT、AIX、AS/400、HP-UX 和 Sun Solaris 版和其它平台
- **主机消息传递：** MQSeries OS/390 版

- 普及型消息传递: MQSeries Everyplace

消息传递本身基于队列管理器，与特定的产品或产品组无关。队列管理器管理可以各自存储消息的队列。应用程序与本地队列管理器进行通信，取出消息或将消息放入队列。如果将消息放入远程队列管理器所拥有的一个远程队列中，消息将通过通道传输到远程队列管理器。这样，消息在到达它们的目的地之前可以经由一个或多个中间队列管理器中继。消息传递的本质是将发送应用程序与接收应用程序分离，必要的话，在中间点对消息进行排队。尽管在细节上有许多差异，但所有的 MQSeries 消息传递产品都与队列管理器、队列、消息和通道的这些相同的基本元素有关。

### MQSeries 主机和分布式产品

MQSeries 主机和分布式消息传递产品用于支持许多不同的网络配置，它们都涉及客户机和服务器，<sup>2</sup> 图2中显示了它们的一些示例。

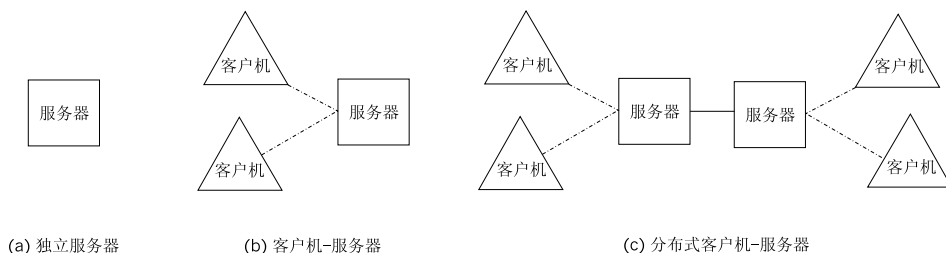


图2. 简单主机和分布式配置

在最简单的情况下，配置一台独立服务器，运行一个队列管理器。一个或多个应用程序在该服务器上运行，经由队列交换消息。另一种配置是客户机/服务器。这种情况下，队列管理器只存在于服务器上，但每个客户机都可以通过客户机通道访问它。客户机通道是一种双向通信链路，其上流动的是独特的 MQSeries 协议，以实现与远程过程调用 (RPC) 相似的某些功能。应用程序可以在客户机上运行，访问服务器队列。客户机/服务器配置的一个优点是，由于依靠服务器队列管理器，客户机消息传递基础结构是轻量型的。缺点是，客户机及其相关服务器同步操作，因此要求客户机通道总是可用的。

分布式客户机/服务器配置显示了一种更为复杂的情况，它涉及到多个服务器。在这些配置中，服务器通过消息通道来交换消息。消息通道是单向的，其协议为安

2. 请注意，这些术语在 MQSeries 主机、分布式和工作站消息传递产品中有非常特定的含义，它们并不总是符合它们更一般的用法。

全、异步地交换消息数据而设计。这些消息通道对于客户机继续处理不必是可用的，但当通信不可用时消息不能在服务器之间流动。

## MQSeries Everyplace

MQSeries Everyplace 通过提供队列管理器，每个队列管理器都启用相应的能力来支持各种各样的网络配置。在这里没有 MQSeries 主机或分布式产品中所具有的客户机或服务器概念。MQSeries Everyplace 队列管理器可以充当传统的客户机或服务器，但实际上，它们每一个都只是可以执行应用程序定义的任务的队列管理器。配置 MQSeries Everyplace 队列管理器时可以带有本地队列也可以不带本地队列就是它的一个例证。在有本地队列的情况下，它可以在本地存储消息，因此可以为应用程序提供异步消息传递，但在没有本地队列的情况下，它只能进行同步消息传递。定制配置的另一个示例是，配置队列管理器时可以带有网桥能力，也可以不带网桥能力。在有网桥的情况下，它可以与 MQSeries 主机或工作站队列管理器交换消息；而在没有网桥的情况下，它只能与其它 MQSeries Everyplace 队列管理器直接通信（尽管它可以通过具有网桥能力网络中的其它队列管理器间接进行通信）。

**注：**MQSeries Integrator (MQSI) 的一个新节点可以在不使用 MQSeries 网桥的情况下直接进行 MQSeries Everyplace 连接。

MQSeries Everyplace 队列管理器使用*动态通道*来交换信息。这些动态通道与该系列中其它成员所使用的*客户机通道*和*消息通道*具有不同的特性。动态通道的一些主要特性有：

- 同时支持*同步*和*异步*消息传递：同步消息传递提供从源应用程序到目标队列直接的传输服务，不需要在源队列管理器上进行排队。异步消息传递是从源队列管理器到目标队列的传输服务，在源队列管理器上可能要进行排队。
- 提供*端到端*服务：通道从源队列管理器到目标队列管理器，可能要通过中间队列管理器。所使用的基本传送协议可能在通道通过这些中间队列管理器时有所改变。
- 支持*压缩*、*加密*和*认证*：通道需要这些安全性特性来保护传输中的数据。
- 支持*对等操作*和*客户机-服务器*操作：定义为对等的通道在功能上是对称的，例如源和目标都可以在通道上启动操作。客户机-服务器通道是请求/响应型的，客户机发出对服务器的请求，服务器响应该请求。（请注意，它不限制消息的流向。消息可以从客户机流到服务器，也可以从服务器流到客户机）。

网桥配置选项可以让 MQSeries Everyplace 队列管理器通过*客户机通道*与 MQSeries 主机和分布式队列管理器进行通信。网桥组件管理一个客户机通道池，这个池可以与一个或多个主机或分布式队列管理器连接。可以在单一网络中配置多个支持网桥的 MQSeries Everyplace 队列管理器，来提供所需的能力、性能和可靠性。

## MQSeries 系列

下面的图表中显示了一些典型但随意的 MQSeries Everyplace 配置。为清楚起见，这些图表只显示了已定义的直接连接。也可以定义利用直接连接的间接连接。带有双向箭头的直线用于表示对等通道，客户机 / 服务器通道由带有指向服务器的箭头的直线表示。客户机可以使用客户机 / 服务器通道来向服务器发送消息，也可以从服务器将指向它们自己的消息拉回。没有箭头的直线表明启用 MQSeries Everyplace 和 MQSeries 之间通信的 MQSeries 客户机通道。



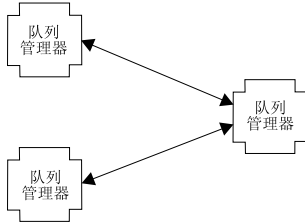
(a) 独立队列管理器

(a) 显示了一个独立的队列管理器，用于支持使用队列来交换数据的一个或多个应用程序。



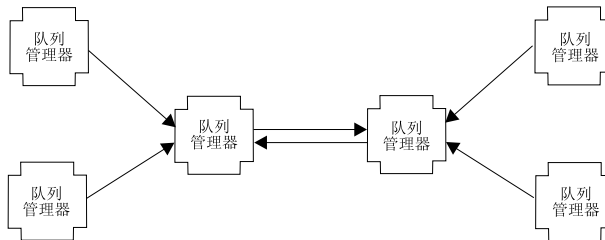
(b) 对等配置

(b) 显示了通过对等通道互连的两个队列管理器。



(b) 对等配置

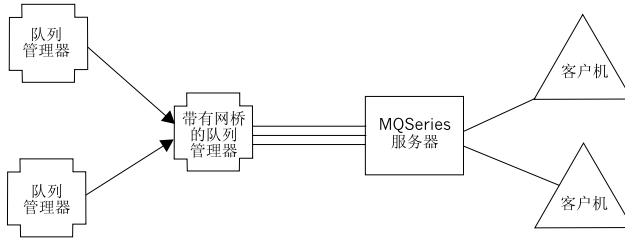
(c) 显示了与第三个队列管理器连接的另一个直接对等通道。它们三个都可以利用基本的直接连接在相互之间交换数据。



(d) 小型网络

(d) 显示了一个小型网络配置，其中的中央服务器队列管理器使用一对直接客户机 / 服务器通道来交换信息。每个客户机队列管理器使用直接客户机 / 服务器通道

来与一个服务器队列管理器连接。



(e) 集成的 MQSeries 系列网络

(e) 显示了一个 MQSeries Everyplace 配置，其中的一个队列管理器已使用网桥选项进行了配置，并且客户机通道池都已指向单一目标 MQSeries 主机 / 分布式服务器。



---

## 第4章 产品需求

本章描述决定 MQSeries Everyplace 设计和实现的需求。

---

### 能力

MQSeries Everyplace 扩展了 MQSeries 系列的消息传递范围:

- 支持低端设备（如 PDA、电话和传感器），允许它们参与 MQSeries 消息传递网络。MQSeries Everyplace 还支持中间设备，例如笔记本电脑、工作站、分布式和主机平台。MQSeries Everyplace 提供相同的服务质量、仅一次确保消息传递，并允许与其它系列成员交换消息。
- 提供轻量型消息传递设施。
- 不论是在存储器中还是在传输中，提供大量的安全性功能来保护消息、队列和相关数据。
- 在网络不稳定或带宽受到严格约束等不利的通信环境中进行有效操作。MQSeries Everyplace 有一个有效的线路协议，可以从通信链路故障中自动恢复。
- 支持移动用户，当设备漫游时允许更改网络连接点。MQSeries Everyplace 还允许在电池资源和网络发生故障或受到约束的情况下对行为加以控制。
- 通过适当配置的防火墙进行操作
- 通过使用户的管理任务最小化，以便设备上的 MQSeries Everyplace 可充分隐藏。这使 MQSeries Everyplace 成为构建具有实用程序风格的应用程序的合适基础。
- 方便地定制和扩展，利用应用程序提供的规则和修改行为的其它类，或对基本对象类进行分类，例如，表示不同的消息类型。

---

### 应用程序

可能的 MQSeries Everyplace 应用程序数目众多而且各不相同，但其中许多应该是为特定用户组开发的定制应用程序。下面的列表提供了一些示例:

- **零售应用程序:** 与主机系统（例如消息代理）进行的帐台交易现金流的输送。
- **消费者应用程序:** 在家里使用 PDA 来进行超市购物、在飞机上收集旅行者的喜好信息、通过移动电话进行金融交易。
- **控制应用程序:** 收集和集成经由人造卫星传送的来自输油管传感器数据，在保证操作员有效的安全性情况下，对装置（如阀）进行远程操作。

## 产品需求

- **移动工作人员:** 拜访的专业人员（保险代理人）、包裹速递公司使用的客户收据的快速打印、快餐店服务员与厨房的信息交换、高尔夫球赛记分、警察使用的移动安全消息发送系统、公用事业工人在经常失去通信的环境中的工作信息、家庭用读表。
- **个人生产力:** 邮件 / 日历复制、数据库复制和笔记本电脑小型化

---

## 用户要求

影响 MQSeries Everyplace 设计的要求包括:

- **管理:** 最少的设置和维护; 支持本地和远程管理; 扩展和定制管理功能以满足特定应用程序需要的能力; 强调自动发现和恢复; 提供可选择使用的独立管理元素。
- **通信:** 一个十分有效的有线协议; 头文件很小; 消息中无强制字段（除了唯一标识外）; 更改数据编码的能力; 压缩、加密和认证支持; 压缩和安全性特性的端对端协商; 方便地通过防火墙的能力; 可插入的通信适配器。
- **兼容性:** MQSeries 服务质量和无缝消息传递交换; 无需应用程序更改而与现有 MQSeries 系统通信的能力; 灵活控制 MQSeries 和 MQSeries Everyplace 之间的消息互换。
- **占用空间:** 对于 Palm 设备, 为 64K 字节。对于 Java 设备, 最小的类文件大小为 100K 字节。
- **功能:** 同步和异步消息传递能力; 访问本地或远程队列中保存的消息; 使用消息中的任何字段进行选择性检索的能力; 有选择地控制队列的后备媒体。
- **规则支持:** 通过规则控制许多方面的行为, 例如, 发送消息的时间、重试通信链接的时间间隔、消息过大的处理方法或当目标队列满时的行为。
- **安全性:** 完全支持安全性、认证和认可; 消息级和队列级安全性; 保护消息传递系统不受安全性攻击; 使用工业标准算法的可插安全性; 与操作系统用户凭证集成的能力; 遵循国家安全性要求的能力, 允许消息跨越国界时更改安全性支持。



## 第5章 产品概念

### 介绍

MQSeries Everyplace 编程模型的基本元素是消息、队列和队列管理器。MQSeries Everyplace 消息是包含应用程序定义的内容的对象。存储时，它们保存在队列中，这种消息可以在 MQSeries Everyplace 网络之间移动。通过指定一对目标队列管理器和队列名，可以将消息指定发往目标队列。应用程序通过放入操作将消息放到队列中，一般通过取出操作来检索它们。队列可以是本地的，也可以是远程的，由队列管理器管理。配置数据存储在注册表中。

MQSeries Everyplace 对象结构可以通过 MQe\_Explorer 管理工具的以下显示来看：

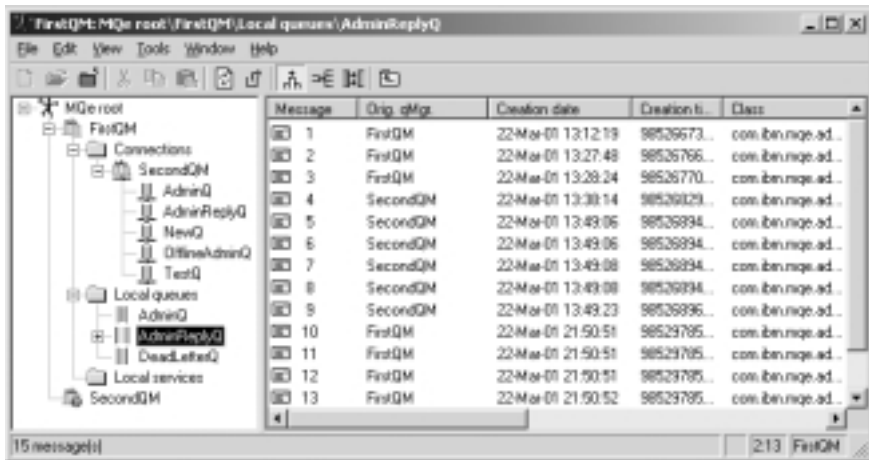


图 3. 由 MQe\_Explorer 显示的 MQSeries Everyplace 对象

在左侧窗格中，对象以树状结构显示。根 MQe root 的下面是两个队列管理器，FirstQM 和 SecondQM。展开 SecondQM 的结构可以看到 Connections、Local queues 和 Local services 的文件夹。展开 Connections 后可以看到从 SecondQM 队列管理器到 FirstQM 队列管理器的连接定义。远程队列管理器 SecondQM 具有五个 FirstQM 队列的定义。展开 SecondQM 的 Local queues 后可以看到两个本地队列，其中一个是在选中的，其内容显示在右侧窗格中。

理解 MQSeries Everyplace 需要对消息、队列、队列管理器和连接的性质有个正确认识。这些概念在以后的章节中进行略述。

## 消息对象

MQSeries Everyplace 消息对象与 MQSeries 消息传递支持的消息有根本的不同。在 MQSeries 中，消息是字节数组，分为消息头和消息主体。需要理解消息头，它包含重要信息，例如应答队列的标识、应答队列管理器、消息标识以及相关标识。消息主体不需要理解。

相反，MQSeries Everyplace 中的消息是从称为字段对象的 MQSeries Everyplace 对象继承来的消息对象。这些消息没有头或消息主体的概念，但有特性和方法。要理解消息对象，首先需要对字段对象的起源有个正确认识。

字段对象在 MQSeries Everyplace 中使用很广泛，它们是字段的累加，其中字段由名称、数据类型和数据本身组成。字段名是长度不限的 ASCII 字符串（除一些保留字符以外）。

字段对象有类型，类型是与编程对象类名称相应的缩写字符串。字段类型可以是：

表 3. 字段对象类型

类型	描述
<b>Ascii</b>	字符串或 ASCII 字符串的动态数组
<b>Boolean</b>	值
<b>Byte</b>	字节值的固定数组或动态数组
<b>Double floating point</b>	双精度浮点值的值、固定数组或动态数组
<b>Fields</b>	对象或字段对象的动态数组（从而支持字段对象的嵌套）
<b>Floating point</b>	浮点值的值、固定数组或动态数组
<b>Integer</b>	整数的（4 字节）值、固定数组或动态数组
<b>Long integer</b>	长整数的（8 字节）值、固定数组或动态数组
<b>Object</b>	值
<b>Short integer</b>	短整数的（2 字节）值、固定数组或动态数组
<b>UNICODE</b>	字符串或 UNICODE 字符串的动态数组

字段对象支持各种方法调用，例如：

- 字段枚举
- 一个或多个构成字段的复制（使用替换选项）
- 字段对象的比较

- 从构成字段取出和放入
- 字段对象的检查（例如，“构成字段是否包含在对象内？”）
- 到字节数组的转储和恢复

转储和恢复方法尤为重要。例如，调用转储方法来提供在链路上进行消息对象传输的数据，而恢复方法可以在接收方重新创建该对象。每个对象都以这种方式负责其自身的传输格式，可以不使用提供的缺省方法来实现定制行为。类似地，这两种方法还用在消息对象从队列进行保存或恢复时。不使用这些方法可以实现完全不同的行为。例如，消息对象可以在传输前的转储时查询数据库来抽取其值。

下表显示了字段对象的某些特性及其构成字段：

表 4. 字段对象及其构成字段特性

特性	存在方式	
	字段对象	字段
关联属性对象	任选	
构成字段	是	
隐藏		是
名称		是
类型	是	是
值		是

两个值得注意的特性是隐藏特性和关联属性对象特性，前者可以出于比较目的而忽略字段，后者提供与字段对象关联属性对象的能力。

属性对象是 MQSeries Everyplace 安全性模型的基础，它允许对内容进行有选择的访问，并对内容加以保护。它们有以下一些重要特性：

表 5. 属性对象特性

特性	描述
认证:	控制访问
加密:	转储对象时保护内容（允许恢复）
压缩:	减少存储量需求（用于传输和 / 或存储目的）
规则:	控制允许的操作

消息对象从字段对象中继承而来，因此包括所有上述能力，包括在消息和组件字段对象级别关联属性对象的能力。另外，消息对象还包括由 MQSeries Everyplace

## 消息对象

生成的 UID（唯一标识）。这个 UID 唯一地标识整个 MQSeries Everyplace 网络中每个个别的消息对象，它从以下内容中构造：

- 发起队列管理器的**名称**（接收到对象时由队列管理器添加）。这个名称必须是全局唯一的。
- 创建消息对象的**时间**（在创建时添加）

因此，除了从字段对象祖先继承来的那些特性以外，消息对象还具有以下特性：

表 6. 消息对象特性

特性	描述
发起队列管理器	首先接收来自应用程序新消息的队列管理器的名称
创建时间	应用程序创建消息对象的时间
UID	MQSeries Everyplace 唯一标识

尽管其它字段几乎肯定会出现，但目的地为另一个 MQSeries Everyplace 队列管理器的消息中不再需要其它信息了。这些附加的字段以几种方式出现：

- 由 MQSeries Everyplace 添加以反映当前状态的字段
- 与特定消息子类相关的字段
- 与消息对象实例相关的定制字段

为了实现消息传递和排队操作，MQSeries Everyplace 将字段添加到消息对象（随后除去它们）。例如，当消息位于队列上时，可以查询消息以找出它是何时放入该队列的。（这可以通过使用将在稍后讨论的队列规则来实现）。当消息在队列管理器之间发送时，可能会添加重新发送字段来表明数据的重新传输正在进行。这里有许多其它示例。

典型的基于应用程序的消息是基本消息对象类的某些子代的实例，因此具有一些额外的字段以满足其自身的需要（例如，发票号码）。在这些附加的字段中，有一些对于许多应用程序是常规和公共的，例如应答队列管理器的名称。认识到这一点后，MQSeries Everyplace 对可能出现在消息中的字段提供了支持。这些字段包括：

表 7. 提供了信息的消息对象字段

字段名	用法
Action	由管理人员使用，表明例如查询、创建、删除等操作
Correlation ID	通常用于使应答与原始消息相关联的字节串
Errors	由管理人员使用，返回错误信息
ExpireTime	过后可以删除消息（即使未被传递）的时间

表 7. 提供了信息的消息对象字段 (续)

字段名	用法
<b>Lock ID</b>	解锁消息所需的密钥
<b>Message ID</b>	消息的唯一标识
<b>Originating queue manager</b>	发送消息的队列管理器的名称
<b>Parameters</b>	由管理人员使用, 传递管理细节
<b>Priority</b>	为消息传输排序所使用的相对优先级
<b>Reason</b>	由管理人员使用, 返回错误信息
<b>Reply-to queue</b>	消息应答应该发往的队列名称
<b>Reply-to queue manager</b>	消息应答应该发往的队列管理器名称
<b>Resend</b>	表明消息是以前消息的重新发送
<b>Return code</b>	由管理人员使用, 返回管理操作的状态
<b>Style</b>	区分命令和请求 / 应答等
<b>Wrap message</b>	包装消息以确保数据保护

在所有情况下都有一个已定义的常数可用, 允许将字段名放在单一字节中。对于某些字段提供了更多信息。例如优先级 (如果存在的话) 影响消息传输的先后顺序、相关标识触发那些字段值的队列索引以供快速检索、到期时触发消息的到期时间等等。

第20页的图4中显示的管理面板说明了消息的结构。请注意, 嵌套字段的使用意味着消息中可以存在复杂的数据层次, 管理工具 (和应用程序) 可以向下探索到自描述结构 (假设它没有通过使用安全性设置禁止)。

当消息对象从字节数组进行序列化或序列化成字节数组来进行传输或队列存储时, 它们使用基本转储和恢复方法。缺省情况下, 为了节省占用空间, 相关的类定义不流动。如果需要, 使用标准序列化方法来覆盖转储和恢复将使类定义流动。

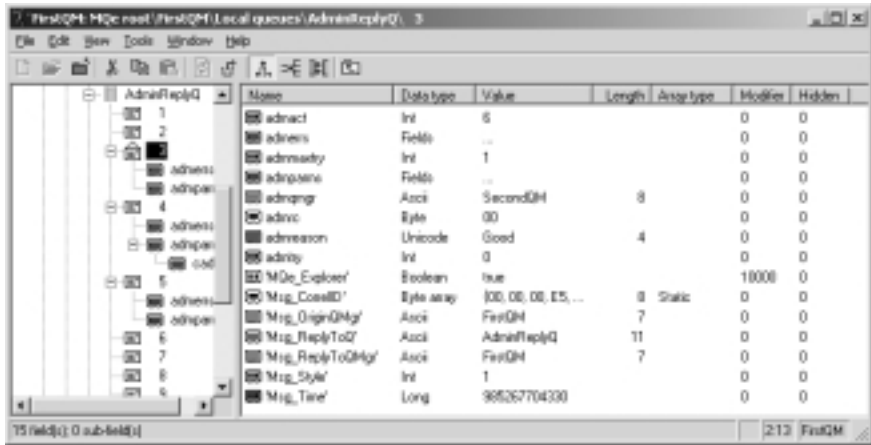


图 4. MQE\_Explorer 显示的消息结构

不过通常类在每个要实例化消息对象的队列管理器上独立可用。如果需要实例化对象（例如，为在传输期间进行中间存储）而知道类文件不可用，可以将消息包装在另一个类中。这通常是缺省消息对象类。这一技术在属性对象用于保护消息时也很有用。这样的属性对象的出现意味着如果没有必需的安全性密钥就无法访问消息内容。不过，包装这样的消息可以让内核消息仍处于完整保护下，而已包装的消息可以自由地转储和恢复。

为了达到有效的消息存储和传输，优化了缺省消息对象转储方法，尽量减小生成的字节串大小。

## 转储数据格式

缺省转储数据格式将字段按如下编码：

```
{长度 标识符 栅栏 {数据}} {长度 标识符 栅栏 {数据}} { ... }
```

其中：

- **数据：** 数据值。除去前导 0 和 F 以压缩整数。布尔值无相关数据字节
- **栅栏：** 一个特殊字节，用于划定标识和任选“数据”项之间的边界。这个字节还表明“数据”项类型
- **标识：** 存放字段名，是一个变长的 ASCII 字节字符串，它以一个结束字节终止。
- **长度：** 表示数据字段的长度。使用可变的字节数，1 到 4。第一个字节的头两位保留，表示长度字段的长度。支持的长度范围在 0 - 1,073,741,823 之间

这会导致非常紧密的数据流。压缩数据可进一步节省空间。对前一个字节流的 XOR 压缩可能会产生好的效果，但因为这些字段的可变性及可更改字段次序这样一个事实，简单一个 XOR 方法并不总能产生所期望的效果。MQSeries Everyplace 包含了一个智能 XOR，在一个字段一个字段的基础上进行操作，这样更能改进压缩。

---

## 队列

队列通常用来存放由应用程序暂挂其要删除的消息对象。如同消息那样，队列也派生自字段对象。由应用程序进行的对队列对象的直接访问通常是不允许的。队列管理器充当应用程序和队列之间的媒介。队列是由名称标识的，名称可以是一个长度不限的 ASCII 字符串<sup>3</sup>，但是，在特定队列管理器内必须是唯一的。MQSeries Everyplace 支持几种不同的队列类型：

### 本地队列

应用程序通过使用本地队列以安全可靠的方式存储消息。存储的消息通过适配器，一个队列一个队列地映射到永久存储器中。MQSeries Everyplace 提供了一系列适配器，其它适配器可以由自己编写，也可以从其它源获得。标准适配器称为 MQeDiskFieldsAdapter，它将队列映射到本地文件系统中，实现确保传递。另一种适配器是 MQeReducedDiskFieldsAdapter，它也将队列映射到文件系统中，但由于操作系统有足够长的时间将其缓冲区清空到物理磁盘子系统，因此它具有更快的性能。而另一种适配器 MQeMemoryFieldsAdapter 将队列映射到内存中。虽然它能提供最快的队列性能，但这种适配器具有一个特性，即消息不能在操作系统或队列管理器重新启动后依然存在。通过创建合适的适配器，消息可以按队列存储在任意地方。存储媒体包括关系数据库或可写 CD。有一些适配器可以利用 IBM 4690 零售商店控制器上镜像文件系统的优势，或利用 DB2 进行队列存储。

本地队列可以联机使用，也可以脱机使用（与网络连接或断开）。队列也可以设置安全性属性，使用的方式与带有属性对象的消息和字段对象的保护非常相象。第40页的『安全性』中讨论了队列安全性。对本地队列上消息的访问总是同步的，这意味着应用程序将在完成放入或取出操作后一直等待 MQSeries Everyplace 返回。

### 远程队列

远程队列是对驻留在远程队列管理器机器上的队列的本地引用。本地引用的名称和目标队列的名称相同，但远程队列定义标识了实际队列的所属队

---

3. 出于互操作性目的，建议遵守 MQSeries 的命名限制，包括将名称的最大长度限制为 48 个字符。长度还会受到所使用的文件系统限制。

列管理器。远程队列还有一些有关访问权的特性，例如访问方式（同步或异步）、所有安全性特征以及传输选项。

MQSeries Everyplace 可以自动建立远程队列。如果尝试访问（例如发送消息）另一队列管理器上的队列，MQSeries Everyplace 将查找远程队列定义。如果有，则使用它，但如果没有，则执行队列发现操作。将发现特征（认证、密码术和压缩）并创建远程队列定义。这样的队列发现取决于目标是否可访问。如果目标不可访问，必须以其它方式提供远程定义。执行队列发现操作时，因为已知队列是同步可用的，MQSeries Everyplace 将访问方式设置成同步。

同步远程队列是一些只有在与特定网络连接时才可访问的队列，这种特殊的网络具有到所属队列管理器的通信路径。如果没有建立网络，那么象放入、取出和浏览这些操作（见第32页的『队列管理器操作』）会导致出现异常。所属队列控制着访问该队列所需的访问许可权和安全性需求。应用程序应该负责处理发送或接收消息时的任何错误或重试，在这种情况下，MQSeries Everyplace 不再对仅一次确保传递负责。

异步远程队列是向远程队列发送消息但无法从远程检索消息的队列。如果建立了网络连接，那么消息将发送到所属队列管理器和队列。但如果没连接网络，消息就存储在本地，直到有一个网络连接为止，然后再传输消息。这可以让应用程序在设备脱机的情况下对队列采取操作。因而这些队列有一个映射到消息存储的适配器，以便在等待传输时，消息可以暂时存储在发送队列管理器上。

### 存储和转发队列

这种类型的队列所存储的消息与一个或多个目标队列管理器目的地相关。它有两个主要用途。第一是在网络中启用消息的中间存储器，这样它们就可以逐步到达它们的目的地（转发角色）。第二个用途是保管等待收集的消息（另见宿主服务器队列）。

存储和转发队列与一组将为之保管消息的队列管理器名相关。它们称为目标队列管理器。发往这些目标队列管理器之一的消息将被替代放在相关的存储和转发队列上。存储和转发队列也可以设置一个转发队列管理器名。如果设置了这个名称，队列就尝试将其所有消息发送到这个指定的队列管理器。如果没有设置该名称，队列将保管消息。

这种类型的队列通常（但不一定）是在服务器或网关上定义。多个存储和转发队列可以存在于一个队列管理器上，但目标名不能重复。存储和转发队列的内容对于应用程序来说不可用。同样，消息发送应用程序也不很清楚在消息传输中存在存储和转发队列和它所扮演的角色。

存储和转发队列上的消息对于应用程序来说不可用。



## 宿主服务器队列

远程队列和存储和转发队列将消息在网络上推送，发送队列启动传输。而宿主服务器队列可以将消息从远程队列拉回。宿主服务器队列定义标识了远程队列管理器上的存储和转发队列。然后宿主服务器队列将所有去住宿主服务器队列本地队列管理器的消息从存储和转发队列上拉回。多个宿主服务器队列定义可以在一个队列管理器上定义，其中每个都与不同的远程存储和转发队列相关。

宿主服务器队列通常位于设备上，并设置成在设备与网络连接时将消息从服务器拉回。当宿主服务器队列将消息从服务器拉回以后，消息被放在相应的目标本地队列中。因此宿主服务器队列本身没有任何应用程序可访问的消息。从网络上的流来说，将消息从服务器取出的拉回方法比服务器推送消息更有效。这是因为宿主服务器队列使用第一条消息的确认作为下一条消息（如果有）的请求，而服务器推送操作需要一个请求/响应来发送消息，需要另一个请求/响应来确认流。当与网络连接时，宿主服务器队列通常设置一个轮询间隔，这样导致队列检查服务器上暂挂的所有消息。这个轮询间隔是一种管理配置选项。宿主服务器队列在让客户机通过客户机-服务器通道接收消息的过程中扮演了重要角色。客户机/服务器连接的性质说明服务器不能启动数据传送。<sup>4</sup>

宿主服务器队列上的消息对于应用程序来说不可用。

## 管理队列

管理队列是在本地或远程对队列管理器（及其相关对象）进行配置所使用的一种机制。发送给管理队列的消息由适当的管理消息类处理，然后，可以任选地将应答发送回起始应用程序。第34页的『管理』中详细讨论了这一主题。

## MQSeries 网桥队列

这是远程队列的一种特殊形式，它的定义位于网关上，目标队列位于MQSeries 队列管理器上。这种形式的队列提供了MQSeries Everyplace 和MQSeries 环境之间的路径。转换程序用于执行所有必需的数据或消息的重新格式化。MQSeries Everyplace 提供了一个基本的转换程序；程序员要定制这个转换程序以满足自身的需求。

MQSeries Everyplace 将数据安全地存储在队列上，针对适配器，确保消息物理地写入媒体而不是简单地由操作系统缓冲。但是，MQSeries Everyplace 并非独立记录对消息和队列的更改。如果要从媒体故障进行恢复，那么必须部署硬件解决方案，如使用RAID 磁盘系统。或者，必须将队列映射到可恢复的存储器，例如某些数据库子系统。

---

4. 另一种方法是使用对等通道，或者将两个队列管理器都配置为具有客户机和服务器能力。

## 队列

MQSeries Everyplace 不要求队列管理器具有已定义的队列。但如果需要，提供了四个系统队列：

- **AdminQ:** 接收管理消息时需要
- **AdminReplyQ:** 可选地用于接收对管理消息的应答
- **DeadLetterQ:** 用于存储不能以其它方式传送的消息
- **SYSTEM.DEFAULT.LOCAL.QUEUE:** 与 MQSeries 服务器上的强制性系统队列共享一个公共名称的队列

下表显示了队列特性。但是请注意，不是所有显示的特性都适用于所有队列类型：

表 8. 队列特性

特性	说明
Admin_Class	队列类
Admin_Name	ASCII 队列名
Queue_Active	表明队列是活动的
Queue_AttRule	控制安全性操作的规则类
Queue_Authenticator	认证程序类
Queue_BridgeName	拥有 MQSeries 网桥名称
Queue_ClientConnection	客户机连接名
Queue_CloseIdle	发送完所有消息后关闭转换程序
Queue_CreationDate	创建队列的日期
Queue_Compressor	压缩程序类
Queue_Cryptor	加密程序类
Queue_CurrentSize	队列上的消息数
Queue_Description	UNICODE 描述
Queue_Expiry	消息的到期时间
Queue_FileDesc	队列的位置和适配器
Queue_MaxMsgSize	队列上允许的最大消息长度
Queue_MaxQSize	允许的最大消息数
Queue_Mode	同步或异步
Queue_MQQMgr	MQSeries 队列管理器代理
Queue_Priority	用于消息的优先级（除非被消息值覆盖）
Queue_QAliasNameList	队列的替代名
Queue_QMgrName	拥有实队列的队列管理器
Queue_QMgrNameList	队列管理器目标

表 8. 队列特性 (续)

特性	说明
Queue_RemoteQName	远程 MQSeries 字段名
Queue_Rule	队列操作的规则类
Queue_QTimeInterval	在处理暂挂消息之前延迟
Queue_TargetRegistry	目标注册表类型
Queue_Transporter	传送程序类
Queue_TransporterXOR	要使用 XOR 压缩的转换程序
Queue_Transformer	转换程序类

管理功能用于创建和删除队列，以及查询或修改其特性。

下面的 MQe\_Explorer 面板显示了描述本地队列特性的四个特性选项卡中的两个。禁用的字段表明该特性对于这类队列不适用。

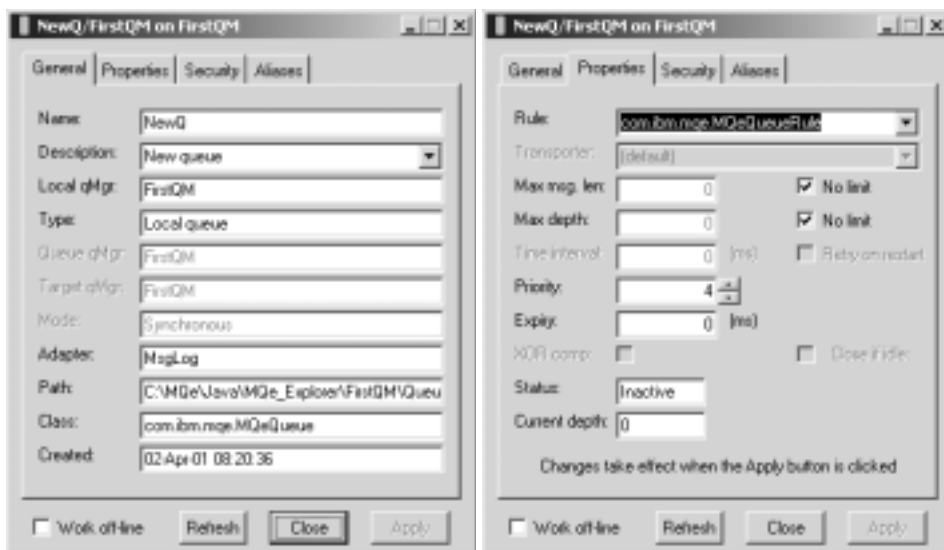


图 5. MQe\_Explorer 显示的本地队列特性

队列不限于作为消息存储使用。划分子类的队列可用在过程控制应用方案中，例如队列对象可直接控制一个阀。属于适当类的消息可以导致阀被打开、流量被更改等等。应用程序不会将消息拖出队列再执行操作，队列对象自己控制操作。例如，其它队列可以更新电子表格或做文本到语音的转换。这一技术的优点是，队列的安全性方面仍保持有效，依然是确保的消息传递。因此，MQSeries Everyplace 仍将确保仅一次消息传递，相关的认证程序和加密程序将保证只有授权的消息发

送方才可发送这样的消息，并且内容在传输过程中是非常安全的。不允许任何应用程序访问该队列，也不需要任何应用程序。

---

## 队列管理器

MQSeries Everyplace 队列管理器为应用程序提供对消息和队列的访问权，并控制所有通道。在 MQSeries Everyplace 版本 1.2 中，无论何时，一台 Java 虚拟机上只能有一个活动的队列管理器。如果一台机器上有多个 JVM，则队列管理器的数目与 JVM 的数目相同。队列管理器由名称标识，该名称必须是全局唯一的<sup>5</sup>，并且是一个长度不限的 ASCII 字符串。<sup>6</sup>

配置具有网桥能力的队列管理器（因此能与 MQSeries 主机和分布式产品交换消息）称为网关。

配置队列管理器可以使用本地排队，也可以不使用本地排队。所有队列管理器都支持同步消息传递操作；有本地排队的队列管理器还支持异步消息传递。

选择使用同步消息传递还是异步消息传递是由发送队列管理器上队列定义的性质所决定的。如果对于目标队列来说存在同步方式远程队列定义，则使用同步传递。如果存在异步方式远程队列定义，则使用异步传递，并带有在消息等待传输时提供本地存储器的定义。如果不存在远程队列定义，但存在为目标队列管理器处理消息的存储和转发队列，则使用异步传递。在这种情况下，存储和转发队列为等待传输的消息提供了本地存储器。如果不存在队列定义，则执行队列发现操作，如果该操作成功，会导致同步消息传递。

不管使用的是同步消息传递还是异步消息传递，MQSeries Everyplace 都会根据可用的连接定义使用直接或间接传输（请参阅第33页的『连接』）。直接传输只涉及两个队列管理器：发送队列管理器和目标队列管理器。间接连接涉及到一系列队列管理器，在这一过程中可能发生协议的更改。在同步消息传递情况下使用间接传输时，消息传递行为与直接传输的情况没有差别。中间队列管理器只是在源和目标之间建立连通性而已。不过，如果在异步消息传递情况下使用间接传输，传输媒介可能在消息从源向目标移动时成为消息的集结待命区。它们是否以这种方式使用取决于在中间队列管理器上，是否以合适的远程队列定义或存储和转发队列的形式定义了适当的中间队列存储器。

---

5. 这一限制不由 MQSeries Everyplace 或 MQSeries 强加，但重复队列管理器名可能会导致消息传递到错误的队列管理器。

6. 出于互操作性目的，建议遵守 MQSeries 的队列管理器命名规则，包括将名称的最大长度限制为 48 个字符。长度还会受到所使用的文件系统的限制。

异步消息传递和同步消息传递有完全不同的特性和结果。使用异步消息传递，应用程序将消息传递给 MQSeries Everyplace，以传递到远程队列。返回到应用程序是立即发生的。如果消息可以立即传递（或移到合适的集结待命区），则发送它，如果不能，则在本地保管。传输重试逻辑由与队列管理器和适当队列相关的规则决定（请参阅第46页的『规则』）。异步传递提供仅一次确保传递服务质量，因为消息已被传递给 MQSeries Everyplace，它已经在负责传递。

使用同步消息传递，应用程序将消息放入 MQSeries Everyplace，以传递给远程队列。MQSeries Everyplace 在与目标队列联系的同时放置消息。传递之后，MQSeries Everyplace 返回到应用程序。返回到应用程序是立即发生的。如果消息无法传递，发送应用程序接收到立即通知。MQSeries 不承担在同步情况下消息传递的责任。

同步和异步远程队列定义可以在 MQSeries Everyplace 网络上自由建立。当网络还包括 MQSeries 消息传递队列管理器（及其相关队列）时，有一些限制很重要。

1. 同步消息传递对于不直接与 MQSeries Everyplace 网关连接的 MQSeries 队列管理器来说不可能使用（因为同步消息传递在 MQSeries 消息通道上不受支持）。为了使这种后果减少到最小，同步传递的定义在这种情况下进行了更改，重新定义成传递给与网关直接连接的 MQSeries 队列服务器。在该队列管理器外，将使用 MQSeries 异步消息传递使消息完成其旅行。
2. 在 MQSeries Everyplace 版本 1.2 中，只可能向 MQSeries 消息传递远程队列定义同步远程队列定义。考虑到上面 (1) 的影响，这意味着异步传递对于与网关直接连接的 MQSeries 队列管理器上的队列来说是不可能的（使用该定义）。不过，如果在这种情况下需要排队，可以通过使用适当的远程队列定义（或存储和转发队列）在 MQSeries Everyplace 网络中通过使用另一个 MQSeries Everyplace 队列管理器来安排。

这样，异步消息传递意味着本地应用程序将消息提供给 MQSeries Everyplace，而 MQSeries Everyplace 要负责将消息从那个本地队列管理器向前传递。这意味着网络和 / 或接收应用程序不必是可用的。发送应用程序不知道实际传递的时间。同步消息传递要求网络是运行的，但发送应用程序知道它已传递到接收应用程序的队列。无论在异步还是同步情况中，接收应用程序都不必是可用的。

MQSeries Everyplace 不象使用其持久和非持久选项对待 MQSeries 消息传递那样为异步消息传递提供仅一次确保传递服务质量的变体。在 MQSeries Everyplace 中，可靠性和性能之间的取舍可以在队列级别，通过选择队列存储适配器来决定，例如：

- MQeDiskFieldsAdapter 确保数据在处理继续之前确实写入磁盘。
- MQeReducedDiskFieldsAdapter 确保数据在处理继续之前传递给操作系统（因此可被检索）
- MQeMemoryFieldsAdapter 将数据保存在内存中

## 队列管理器

在所有情况下都可以实现仅一次确保传递，但恢复则取决于选择的消息存储。性能代价在 SupportPac EP01 中描述。

表9中显示了队列管理器的特性。

表 9. 本地队列管理器特性

特性	描述
Admin_Class	队列管理器类
Admin_Name	队列管理器名
QMgr_ChnlAttrRules	通道属性规则
QMgr_ChnlTimeout	通道超时
QMgr_Description	UNICODE 描述
QMgr_QueueStore	队列的缺省位置和适配器
QMgr_Rules	队列管理器操作的规则类

下面的面板显示了描述队列管理器特性的两个特性选项卡中的一个：



图 6. 由 MQE\_Explorer 显示的队列管理器特性

## 队列管理器配置

队列管理器运行所在的环境是由 MQSeries Everyplace 在装入队列管理器之前建立的。队列管理器自己将配置信息存储在其注册表中（在第44页的『注册表』中详细进行了描述）。队列本身（包括消息）存储在队列存储中。

MQSeries Everyplace 环境可以通过许多方法建立，可以通过调用 API、通过 MQSeries Everyplace 所带的实用程序，或者通过例如 MQe\_Explorer 这样的管理工具。许多工具都捕捉初始化文件中的环境参数，但这完全是可选的。下表显示了一些主要的环境变量，使用的部分名遵照初始化文件中用于表示该信息的约定：

表 10.

部分名	特性	说明
[Alias]	（类别名定义）	别名可以在需要类名的任何地方使用
[ChannelManager]	MaxChannels	允许同时出现的客户机 / 服务器通道的最大数
[Listener]	Network	侦听进入客户机 / 服务器通道连接请求所在的适配器和端口号
	Listen	用于处理产生的连接的适配器
	TimeInterval	客户机 / 服务器通道超时
[MQBridge]	（网桥的初始化参数）	
[MQe_Explorer]	（已保存的寻址信息）	MQe_Explorer 保存的信息描述了如何对该队列管理器创建进入连接
[Permission]	（允许的命令）	允许的通道命令、适配器类和文件描述符映射
[PreLoad]	初始化队列管理器时要装入的类	它提供了装入应用程序类的一种机制
[QueueManager]	Name	队列管理器名

表 10. (续)

部分名	特性	说明
[Registry]	DirName	注册表位置
	LocalRegType	注册表类型 (文件或专用)
	PIN	注册表 PIN (或提示请求)
	CertReqPIN	Certificate PIN (或提示请求)
	KeyRingPassword	密钥环密码 (或提示请求)
	CAIPAddrPort	认证中心 IP 地址

下面的四个图显示了使用 MQe\_Explorer 创建新队列管理器期间的一些重要的选项卡。不假设任何先前的环境，也不需要任何现有的初始化文件。产生的结果是一个正在运行的队列管理器，它的配置数据保存在初始化文件中，因此只需要通过打开该文件便可以重新启动队列管理器。



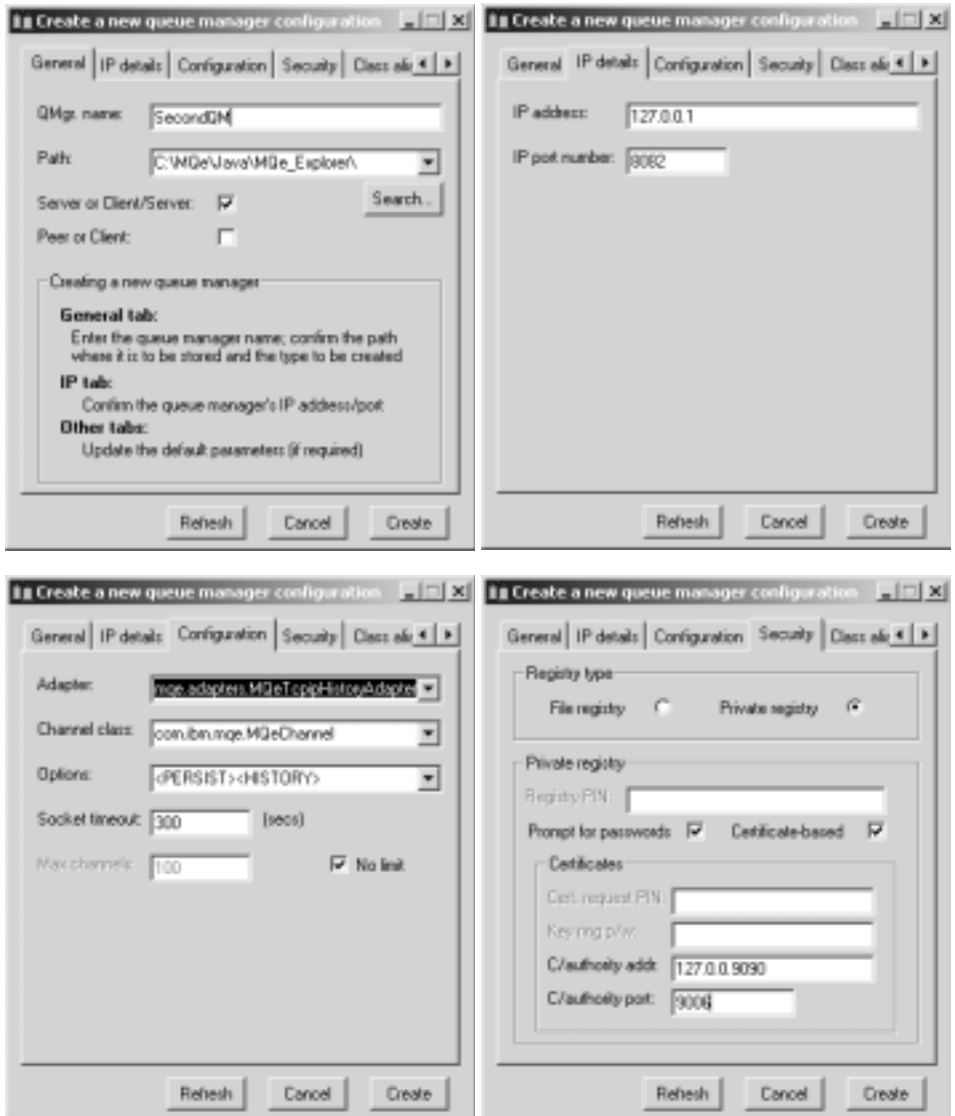


图 7. 使用 MQE\_Explorer 创建队列管理器

创建队列管理器实际上不需要知道队列管理器的 IP 地址。不过，MQE\_Explorer 捕捉到在配置其它与该队列管理器通信的队列管理器中使用的信息。需要端口号来侦听进入客户机 / 服务器连接请求。

通道侦听器需要适配器配置数据；通道管理器需要时间间隔和最大通道数。MQE\_Explorer 象以前那样捕捉其它信息来配置其它队列管理器。

在本例中，已有一个使用基于证书的认证方法配置的一个安全注册表（请参阅第40页的『安全性』）。因为 MQe\_Explorer 会在稍后需要时提示输入 PIN 和密码，现在它们被禁用。

### 队列管理器操作

队列管理器支持消息传递操作并可以管理队列。应用程序通过队列管理器的服务来访问消息，使用的方法有诸如取出、放入、浏览、等待、侦听和删除。许多操作都将过滤器作为它们的参数之一。过滤器是一种用于等同性比较的字段对象，消息中的所有字段都可以用于选择性检索。大多数方法调用还包括一个属性对象，用在消息的编码和译码中。

取出操作破坏性地将消息从队列中除去。根据过滤器所强加的条件，消息按优先级顺序，在其中再按照它们到达队列的时间来检索。因此，如果所有其它信息相同，第一条到达的消息将是第一个被检索的。取出可以作为一个单一步骤使用，也可以作为两步骤操作使用。两步骤的情况是为出现以下情形提供的：当消息从 MQSeries Everyplace 传递到应用程序期间不允许丢失消息。首先发出带有确认标识（其值由应用程序选择）的取出操作。该操作为应用程序取出消息，但并不将它从队列中立即删除，而是将它隐藏在队列上。随后，指定原始消息 UID 的确认操作表明应用程序的取出操作成功，然后删除该消息。如果取出操作失败，则允许恢复该消息。放入操作以类似的方式工作。

通过指定 UID，无需检索就可以从队列中删除消息。

如果需要非破坏性的读取，可以浏览队列以获得消息（可以在过滤器控制之下）。浏览将检索与过滤器匹配的所有消息对象，但仍将它们保留在队列中。还支持锁定下的浏览。它有一个在队列上锁定匹配消息的附加功能。消息可以单独锁定，也可以按由过滤器标识的组锁定，锁定操作返回一个锁定标识。只有在提供了锁定标识后，才可取出或删除锁定的消息。浏览的一个选项允许返回所有消息或只返回 UID。

对于到达队列的消息，应用程序可以等待一段指定的时间。过滤器可以用来标识感兴趣的那部分，而且还可指定一个确认标识。应用程序也可以再次选择通过使用过滤器侦听 MQSeries Everyplace 消息事件。当消息到达队列时通知侦听器。

表11显示了队列允许的消息传递操作

表 11. MQSeries Everyplace 队列上的消息传递操作

	本地队列	远程队列 <sup>1</sup>	
		同步	异步
浏览（±锁定，±过滤）	是	是	

表 11. MQSeries Everyplace 队列上的消息传递操作 (续)

	本地队列	远程队列 <sup>1</sup>	
		同步	异步
删除	是	是	
取出 (±过滤器)	是	是	
侦听 (±过滤器)	是		
放入	是	是	是
等待(±过滤器)	是	是	
<p><b>注:</b></p> <ol style="list-style-type: none"> <li>1. 同步远程等待操作是通过远程队列的轮询实现的，因此实际等待时间是轮询时间的倍数</li> <li>2. <sup>1</sup>MQSeries Everyplace 版本 1.2 所提供的 MQSeries Everyplace MQSeries Bridge 只支持确保 / 非确保放入、非确保取出和非确保浏览（不锁定）。</li> </ol>			

## 连接

拓扑和通过 MQSeries Everyplace 网络的访问是由连接对象定义的。这些定义以本地方式存储在每个队列管理器上。它们是通过提供的标准管理工具创建、修改和毁坏的。

连接对象通常定义对远程队列管理器的访问（因此它们有时也指远程队列管理器定义）。下表给出了一些特性：

表 12. 连接（远程队列管理器）特性

特性	说明
Admin_Name	队列管理器名
Con_Adapter	适配器文件描述符
Con_AdapterOptions	适配器选项（如使用历史）
Con_AdapterParm	适配器使用的 ASCII 数据（如小服务程序名）
Con_Aliases	队列管理器 / 连接的替代名称
Con_Channel	此连接应该使用的通道类型
Con_Description	UNICODE 描述
Queue_QMgrName	定义的拥有者

下面的管理面板显示了两个与连接定义相关的选项卡：

## 连接

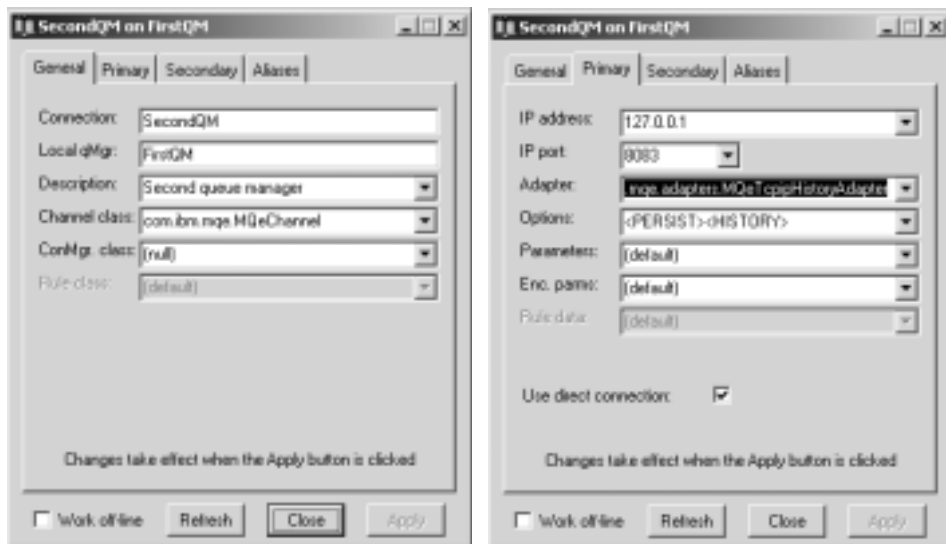


图 8. MQE\_Explorer 显示的连接定义

数据可以以选项、参数和编码参数的形式传递给选中的通信适配器。

可以指定间接连接。在这种情况下，MQSeries Everyplace 把连接路由通过其它队列管理器（可以是链接起来的），途中协议也可能会更改。间接连接在让设备对 MQSeries Everyplace 网络有单个进入点时尤其有用。

连接对象也用于为进入对等通道定义侦听器。

象大多数 MQSeries 对象一样，可以为连接定义别名。本地连接（定义的一种连接，其名称与本地队列管理器的名称匹配）用于为本地队列管理器本身定义别名。

---

## 管理

管理提供了配置和管理 MQSeries Everyplace 资源（如队列和连接）的工具。与消息相关的功能被视为应用程序的责任。管理是通过处理管理消息的产生和接收的接口启用的，并且设计成用同等方式处理本地和远程管理。请求被发送到目标队列管理器的管理队列，如果需要，可以接收应答。任何本地或远程 MQSeries

Everyplace 应用程序都可通过帮助程序方法直接或间接地创建和处理管理消息。管理消息也可以通过 MQe\_Explorer 间接生成<sup>7</sup>，它是为系统管理提供图形用户界面的一种管理工具。

管理队列不知道如何管理个别资源；这一信息被封装在每个资源及其相应的管理消息中。

## 管理消息

管理消息扩展了基本 MQSeries Everyplace 消息对象。表13列出了为 MQSeries Everyplace 资源管理提供的消息类。可以对这些基本管理消息分类，以用于其它对象的管理；例如，可以用 MQeQueueAdminMsg 的子类管理不同类型的队列。MQSeries 的 MQSeries Everyplace 网桥以这种方法使用 MQeAdminMsg 的子类。

表 13. 管理消息类

管理消息类	用途
MQeAdminMsg	用作所有管理消息基础的抽象类
MQeQueueManagerAdminMsg	管理队列管理器
MQeQueueAdminMsg	管理本地队列
MQeRemoteQueueAdminMsg	管理远程队列
MQeAdminQueueAdminMsg	管理管理队列
MQeHomeServerQueueAdminMsg	管理宿主服务器队列
MQeStoreAndForwardQueueAdminMsg	管理存储和转发队列
MQeConnectionAdminMsg	管理队列管理器之间的连接
MQeClientConnectionAdminMsg	管理用于与 MQSeries 的连接网桥客户机连接对象
MQeListenerAdminMsg	管理用于收集来自 MQSeries 的消息的网桥传输队列侦听器对象
MQeBridgeAdminMsg	管理 MQSeries 的网桥
MQeMQBridgesAdminMsg	管理 MQSeries 网桥列表
MQeMQMgrProxyAdminMsg	管理 MQSeries 队列管理器的网桥表示
MQeMQBridgeQueueAdminMsg	管理 MQSeries 网桥队列

管理消息的结构取决于其特定类，即它正在管理的资源的性质以及将对该资源执行的具体操作。然而，通常管理消息的结构如第36页的表14中所示：

7. MQe\_Explorer 不包括在 MQe\_Explorer 中，但可以从万维网上的 MQSeries Everyplace 站点免费下载 (<http://www.ibm.com/software/ts/mqseries/everypace>)。

表 14. 管理消息的一般结构

第 1 级字段	第 2 级及以下字段	用途
Admin_Action		创建、删除、查询等等
Admin_Errors		字段对象父代
	多字段	基于每个错误的详细信息
Admin_MaxAttempts		管理操作应该尝试的最大次数
Admin_Parameters		字段对象父代
	资源	要管理的资源的名称
	多字段	特定于消息类和操作的详细参数数据
Admin_Reason		表明失败原因的文本消息
Msg_ReplyToQ		响应应该发送到的那个队列的名称
Msg_ReplyToQMgr		响应应该发送到的那个队列管理器的名称
Admin_RC		表示结果的数字返回码
Msg_Style		命令或请求 / 应答
Admin_TargetQMgr		拥有目标资源的队列管理器的名称

支持三种风格的管理消息，即表明不需要应答的管理操作的命令（数据报）、需要应答的请求和应答本身。应答是根据原始消息副本构造的，因此，发送方可以添加附加字段，以供接收方使用。

### 选择性管理

对管理的访问可通过管理队列上的认证程序来加以控制。对于本地应用程序，提供的认证程序认为它们都表示同一个本地用户，因此，对于它们的管理或者全部允许或者全部不允许。在任何管理消息流动之前，远程管理应用程序受控于通道上认证程序的调用。这样，可以区分不同的远程用户，并分别启用或禁用。对于任何用户在所有情况下，管理是以其整体启用或禁用的。如果需要更细级别的管理控制，例如要授予某几个管理用户对某些队列的访问权，而不授予对其它队列的访问权，那么就需要进一步编程。一个更高级的认证程序可以跟踪与用户身份相关的许可权，随后可以根据这些许可权来处理管理消息（请参阅第40页的『安全性』）。也可开发一些与队列关联的规则来允许或禁止类似的操作（请参阅第46页的『规则』）。

## 监控和相关操作

管理比对象创建和修改通常包括更多内容。它可以包括监控系统、在队列满的时候通知操作员，或者处理一些错误情况，例如在当对于目标队列来说过大的消息到达时采取适当操作。在 MQSeries Everyplace 中，这些问题都是通过使用规则来处理的，即当对象显著更改其状态时或当出现某些类型的错误状况时调用的类。MQSeries Everyplace 提供了一组缺省规则类，但通常用定制类（见第46页的『规则』）替换它们。

---

## 动态通道

MQSeries Everyplace 通过逻辑链路（称为动态通道）在队列管理器之间通信。这些动态通道支持双向流动，并根据需要由队列管理器建立。异步和同步消息传递都使用相同的通道，而且使用的协议专用于 MQSeries Everyplace。该协议可以通过覆盖消息转储方法来根据消息进行定制。相反，MQSeries 通常将客户机通道用于其同步通信，将一对消息通道用于双向异步消息传递。MQSeries 群集消息通道与 MQSeries Everyplace 动态通道有某些相似的特征，但也存在许多重要差异。

动态通道是两个队列管理器之间的逻辑连接，是为了发送或接收数据而建立的。甚至在同一方之间也可存在多个并行通道。它们有一些特性，例如：认证、密码术、压缩和所使用的传送协议。这些特性是可插入的，（不同的通道上可能使用不同的版本），因此每个通道都有对应以下的其自身的服务质量属性：

- **认证程序：**空或者一个能执行用户或通道认证的**认证程序对象**
- **通道：**提供传送服务的类。
- **压缩程序：**空或者一个能执行数据压缩及解压缩的**压缩程序对象**
- **加密程序：**空或者一个能执行加密和解密的**加密程序对象**
- **目的地：**该通道的目标，例如：SERVER.XYZ.COM

通常仅在设置通道时使用认证程序。压缩程序和加密程序通常在所有的流中使用。

加密程序的最简单的类型是 MQeXorCryptor，它通过对正在发送的数据执行 exclusive-OR 来加密数据。这种加密是不安全的，但它会修改数据，使它不能被查看。相反，MQe3DESCryptor 实现三重 DES。最简单的压缩程序类型是 MQeRleCompressor，它通过用一个计数值替换重复的字符来进行压缩。有关提供的其它认证程序、压缩程序和加密程序，请参阅第41页的表15。

通道的建立使用协议适配器规范来确定用于特定通道的链接和协议。在每个中间节点上，搜索通道定义以解析下一个链接所需要的地址。当没有向前定义存在时，通道终止并且在那时将所有流过的消息传递到队列管理器。使用这种方法，可以利用中间存储和转发队列和远程队列定义。

## 动态通道

通道对应用程序或管理员而言不是直接可见的，它们根据需要由队列管理器建立。通道将队列管理器链接在一起，它们的特性是 MQSeries Everyplace 根据流经的信息进行协商并重新协商。传送程序是使用通道来提供队列层通信的 MQSeries Everyplace 组件。同样，这些传送程序对应用程序员或管理员都是不可见的。

当需要确保的消息传递时，MQSeries Everyplace 将这些消息传递到应用程序一次，且只一次。要实现这个功能，请确保已成功将消息从一个队列管理器传递到另一个队列管理器，并在发送端删除其副本前已得到确认。如果在出现通信故障时还没有接收到确认，可以重新发送消息（仅一次传递并不表示只传输一次），但不重复传递。

---

### 适配器

适配器用来将 MQSeries Everyplace 映射到设备接口。通道使用协议适配器在 HTTP、本机 TCP/IP、UDP 和其它协议上运行。类似地，队列利用字段存储器适配器与存储器子系统（例如内存或文件系统）相接。适配器为 MQSeries Everyplace 提供一种扩展其设备支持并允许版本控制的机制。

文件描述符是一个用来标识、装入和激活适配器的字符串。

---

### 拨号连接管理

设备的拨号联网支持是由设备操作系统处理的。当 MQSeries Everyplace 在一个断开的设备上试图使用网络（例如因为必须发送消息）时，如果网络堆栈是不活动的，则操作系统本身启动远程访问服务 (RAS)。通常，这种方法采取为用户显示提供拨号连接概要的面板的形式。在连接建立之前，操作系统处在控制中。因此，设备用户必须确保相应的拨号连接概要对使用的操作系统是可用的。在 MQSeries Everyplace 版本 1.2 中没有对拨号联网的显式支持。

---

### 跟踪

跟踪是通过运行一个执行跟踪操作的独立程序来启用的。在 MQSeries Everyplace 中嵌入调用以跟踪不同系统和用户的信息、警告和出错情况。应用程序同样可以直接调用跟踪，并可以添加新消息或修改现有的跟踪消息。所提供的样本跟踪程序将允许选定消息的显示、打印和 / 或定向到事件日志中。可以编写有附加能力或设计成以其它方法格式化和传递其输出的其它跟踪程序。

大多数 MQSeries Everyplace 异常将传给应用程序处理，并且应用程序异常处理程序同样可以将这些异常转给跟踪。



---

## 事件日志

MQSeries Everyplace 提供可以用来记录状态（例如队列管理器启动）的事件日志机制和接口。启动记录，缺省情况下将输出写入一个文件，但是这可以截取并指向其它地方。MQSeries Everyplace 事件日志不记录消息数据且不能用来恢复消息或队列。

---

## 消息传递

MQSeries Everyplace 网络是互连的队列管理器，可以包括网关。它们能跨越多个物理网络并在它们之间传递消息。总之，他们用 一个与队列位置无关的编程模型提供对队列的同步和异步访问。

### 异步消息传递

将一条消息以异步方式放入远程队列后，就在逻辑上将这个消息对象，连同它的目的队列管理器和队列名以及压缩程序、认证程序和与消息的目标目的地匹配的加密程序特性一起存放在与队列的本地定义关联的后备存储器中。当对象以一种安全格式保存到持久存储器中（如其目的地队列所定义的那样）时，调用该对象的转储方法。队列管理器控制消息传递。对于下一次中继，它用适当的特性来标识（或建立）到这个队列管理器的通道，然后创建（或重新使用）到目标队列的传送程序。这个传送程序转储对象并发送结果字节串。请注意，目标队列管理器和队列名不是消息流的一部分。

如果合适，消息在通道上加密并压缩。当消息到达目的地队列管理器时，将它解密和解压缩。可以使用对象类的恢复方法创建一个新消息对象，所创建的对象存放在目的地队列中。如果这条消息还没有到达目的地队列管理器，则对它解密并解压缩，然后以适当的向前传送特性将它存放在存储和转发队列。在这两种情况下，如其目的地队列所定义的那样，消息都是以一种安全格式保存在各自的队列中。

异步消息传递的一个特性是将消息经中间中继传到队列管理器，排队以用于向前发送。消息首先按优先级次序然后按时间戳记顺序依次离开中间队列。

### 同步消息传递

同步消息传递类似于上面描述的异步情况，但在中间几次跳跃中，队列管理器是在更低级别参与的，涉及传送程序和通道。使用在每个中间节点的协议规范中定义的适配器来建立端对端通道以标识下一个链接。在最后一个链接的结束处，不存在进一步的相关文件描述符，取出的消息传到队列管理器的更高层以作处理。因此发送节点并不对消息进行排队，而只是将它沿着通道，经过中间中继，然后将它交给目的地队列管理器，以放置在目标队列中。

## 消息传递

到 MQSeries 的链接使用网关上的网桥队列，它将消息转换成 MQSeries 格式。这种机制意味着来自设备的同步 MQSeries Everyplace 样式的消息传递对于 MQSeries 来说是可能的，它的动态通道在网关上终止。从网关以实时方式传递的消息经过一个客户机通道到达 MQSeries 服务器。从那里，其目的地可能需要将它沿着 MQSeries 消息通道进行异步路由。

通过使用类似的方法，假定有一个适当的中间节点，则只有同步消息传递能力的设备才可将消息发送到一个异步 MQSeries Everyplace 队列。

---

## 安全性

MQSeries Everyplace 提供一组集成的安全性功能，能够在消息数据的本地保存和传送时对它加以保护。

MQSeries Everyplace 安全性功能提供三种不同类别的保护：

- 本地安全性 - 消息（和其它）数据的本地保护
- 基于队列的安全性 - 在启动队列管理器 and 目标队列之间的消息保护
- 消息级安全性 - 在启动程序和接收方之间消息的消息级保护

MQSeries Everyplace 本地和消息级安全性不但由 MQSeries Everyplace 在内部使用，而且还可用于 MQSeries Everyplace 应用程序。MQSeries Everyplace 基于队列的安全性是一个内部服务。

所有这三种类别的 MQSeries Everyplace 安全性功能都通过使用一个属性（MQeAttribute 或子代）来保护消息数据。根据类别，可以明确地或隐含地应用这个属性。

每个属性都可以包含以下对象：

- 认证程序
- 加密程序
- 压缩程序
- 密钥
- 目标实体名

根据 MQSeries Everyplace 安全性功能的类别不同，这些对象的使用方法也有所不同，但在所有情况下，当调用附加到这个消息对象的属性时，将应用 MQSeries Everyplace 安全性功能保护。这在调用 MQSeries Everyplace 消息的转储方法时发生（使用属性的 encodeData 方法来进行加密和压缩消息数据等操作时）。在调用

MQSeries Everyplace 消息的恢复方法时（当使用属性的 decodeData 方法来进行解压缩和解密消息数据等操作时），不能实现 MQSeries Everyplace 安全性特性的保护。

MQSeries Everyplace 版本 1.2 所支持的认证、加密和压缩的算法详见表15。

表 15. 认证、加密和压缩支持

功能	算法
认证	WTLS 迷你证书
	确认 Windows NT/2000、AIX 或 Solaris 身份
压缩	LZW
	RLE
加密	3DES
	DES
	MARS
	RC4
	RC6
	XOR

## MQSeries Everyplace 本地安全性

本地安全性在本地保护 MQSeries Everyplace 消息（或 MQeField 或 MQeField 子代）数据。这是通过创建一个具有适当对称加密程序和压缩程序的属性、创建并设置适当的密钥（通过提供密码或密码短语）来实现的。密钥明确与属性相连，而属性又与 MQSeries Everyplace 消息相连。MQSeries Everyplace 提供 MQeLocalSecure 类来帮助设置本地安全性，但在所有情况下，设置适当的属性和管理密码或密码短语密钥是本地安全性用户（MQSeries Everyplace 内部或一个 MQSeries Everyplace 应用程序）的职责。

## MQSeries Everyplace 基于队列的安全性

基于队列的安全性可应用于同步和异步消息。

### 同步基于队列的安全性

使用同步基于队列的安全性可以让应用程序将所有有关消息安全性的考虑事项留给 MQSeries Everyplace。队列有认证、加密和压缩等特性，这些用于确定保护消息流（以及持久存储器）所需的安全性级别。

当发送一条消息时，从本地注册表中检索目标队列的安全性特性。如果没有这些特性，队列管理器尝试从目标队列管理器中发现目标特性并且将它们进行高速缓

## 安全性

存以供以后再次使用。如果到那个队列管理器的通道存在，则使用它；如果不存在，则创建一个新的通道。检索了目标队列属性。

基于所需的服务质量，目标队列管理器的通道属性是动态更改的。这针对已建立的所有规则。通常，规则允许使安全级升级，（例如：从无保护到弱保护，或从弱保护到强保护）。如果不能对通道升级，或认为这个安全级太过分（例如：不需要保护，而可用的通道实现强保护），则需创建新的通道。有通道池存在，它在可能的情况下重用，并根据通信量的需要动态更改特性。在不需要时会自动删除通道。消息总是放在目标队列特性定义的安全性级别的队列中。

认证在通道级发生，使每条消息的开销保持最小。通常，还使用具有对称加密程序的同步基于队列的安全性，因为这可导致快速加密/解密。然而，在这些对称情况下，MQSeries Everyplace 最初使用 RSA 不对称加密，以保护在发送方和接收方处建立共享密钥所需的流。在这以后使用对称加密来保护流动数据的机密性。MQSeries Everyplace 通过在每个通道流中动态更改密钥，使得对数据的密码侵袭更困难。MQSeries Everyplace 还通过在数据发送前生成并向数据添加摘要，并在接收时重新生成和确认它来确保流动的数据的完整性。

### 异步基于队列的安全性

异步消息传递不同于上面所描述的同步情况，它在执行 `putMessage` 时不保证目标队列可访问。在这种情况下，队列管理器不能立即发送消息并将它放在传输队列中；但是可对它进行加密以与目标队列保持一致的特性。当可以发送它时，对它解密，然后发送到一个具有适当特性的通道。因此消息总是受保护的，即使在等候传输时。异步消息传递需要一个远程队列定义来确定目标队列的特性。

在异步情况下，不可能在始发方和目标之间进行认证。如果认证很重要，例如接收方要确定消息的最初发送方（以确定接受还是建立非拒绝）或启动程序来确保该消息只能由指定的接收方处理，则必须使用消息级安全性。

基于队列的安全性可以与消息级安全性同时使用，但不是必需的，因为消息数据已是受保护的。

## 消息级安全性

消息级安全性对启动和接收 MQSeries Everyplace 应用程序之间的消息数据提供保护。

消息级安全性是一个应用层服务，它要求启动 MQSeries Everyplace 应用程序设置消息级属性并在使用 `putMessage` 将消息放入目标队列时提供它。接收应用程序必须设置一个匹配的消息级属性并将其传递到接收队列管理器，以便在应用程序调用 `getMessage` 以从目标队列取出消息时，这个属性是可用的。

和本地安全性一样，消息级安全性利用了消息对象上属性的应用程序。启动应用程序的队列管理器使用转储方法（它使用属性的 `encodeData` 方法保护消息数据）处理 `putMessage`。接收应用程序的队列管理器使用恢复方法（它使用属性的 `decodeData` 方法恢复原始的消息数据）处理应用程序的 `getMessage`。

MQSeries Everyplace 为消息级安全性提供两个可供选择的属性：

### MQeMAttribute

这用于商家对商家通信，其中相互的信任在应用层得到严格的管理并且不需要委托第三方。可使用所有可用的 MQSeries Everyplace 对称加密程序和压缩程序选项。与本地安全性一样，必须在提供给 `putMessage` 或 `getMessage` 之前预置属性的密钥。MQeMAttribute 对消息级保护提供一个简单而有效的方法，启用强加密方法以保护消息的机密性，而不需要任何公用密钥基础架构 (PKI) 开销。

### MQeMTrustAttribute

通过使用数字签名和利用缺省公用密钥基础架构，这个属性提供一种更高级的解决方案。它使用 ISO9796 数字签名 / 确认来使接收应用程序能够证明消息是来自所声称的发送方。使用所提供属性的加密程序来保护消息机密性。SHA1 摘要保证了消息的完整性，RSA 加密 / 解密确保消息只能由指定的接收方恢复。与 MQeMAttribute 一样，能使用所有可用的 MQSeries Everyplace 对称加密程序和压缩程序选项。为大小优化选择，使用的证书是 WTLS 迷你证书。通过 MQSeries Everyplace 缺省基础架构提供了认证（确认签名）和加密 / 解密所必需的信息的相互可用性。

一个典型的用 MQeMTrustAttribute 保护的消息有以下格式：

RSA-enc{SymKey}、SymKey-enc {数据，数据摘要，数字签名}

其中：

<b>RSA-enc:</b>	使用指定接收方的公用密钥进行的 RSA 加密
<b>SymKey</b>	生成的伪随机对称密钥
<b>SymKey-enc</b>	用 SymKey 进行对称加密
<b>数据</b>	消息数据
<b>数据摘要</b>	消息数据摘要
<b>数字签名</b>	启动程序的消息数据数字签名

消息级安全性不依赖于队列级安全性。

### 注册表

注册表主要用于存储与队列管理器相关的信息，每个队列管理器都有一个注册表。每个队列管理器都使用注册表来存放其：

- 队列管理器配置数据
- 队列定义
- 远程队列定义
- 远程队列管理器定义
- 用户数据（包括与配置相关的安全性信息）

通常只限于合法的队列管理器用户来访问注册表，并受 PIN 保护，但对更关心占用空间大小而非安全性的用户有一个可配置选项能够忽略这个限制。

### MQSeries Everyplace 可认证的实体

基于队列的安全性使用基于迷你证书的相互认证，消息级保护使用数字签名引发可认证的实体概念。在相互认证的情况下，通常认为是指两个用户（人）之间的认证，但是一般而言，消息传递没有用户的概念。通常在应用程序层管理这个概念，即，通过消息传递服务的用户。MQSeries Everyplace 有意将认证目标概念从用户抽象为可认证的实体。这不排除可认证实体是人的可能性，但这将是一个选择应用程序的映射。MQSeries Everyplace 在内部将所有或者是迷你证书从属服务的发起者，或者是它的目标的队列管理器定义成可认证实体。另外，MQSeries Everyplace 还将定义成使用基于迷你证书的认证程序的队列定义成可认证实体。因此支持这些服务的队列管理器可以有一个可认证实体、队列管理器，或一组可认证实体、队列管理器和每个使用基于证书的认证程序的队列。

### 专用注册表和凭证

每个可认证实体都需要其自己的凭证才能够发挥作用。这提供了两个难题。首先，如何执行注册以获得凭证，其次，在何处以安全方式管理这些凭证。原先，这些难题比基本的密码技术更难解决。MQSeries Everyplace 提供了用来使可认证实体执行自动注册的缺省服务，启用可认证实体专用凭证安全管理的专用注册表（基本注册表的一个子代），以及一组管理公用凭证的公用注册表（同样是基本注册表的一个子代）。专用注册表向基本注册表提供许多不同质量的安全或密码记号，例如，对于公共对象（例如迷你证书）和专用对象（例如专用密钥），它可以是安全资源库。它提供一种只允许特许用户访问专用对象的机制。它以一种这样的方法对一些服务（例如：数字签名、RSA 解密）提供支持，即专用对象永远没有专用注册表。通过提供公共接口，它隐藏了基本设备支持，当前限于本地文件系统，但是将来非常可能扩展到可移植的记号。

## 自动注册

MQSeries Everyplace 提供支持自动注册的缺省服务。在配置可认证实体时自动触发这些服务，例如当启动队列管理器或定义一个新的队列时。在这两种情况中都触发了注册并在创建新的凭证后将其存储到可认证实体的专用注册表中。自动注册的步骤包括生成新的 RSA 密钥对、保护和保存专用注册表中的专用密钥；以及将公用密钥包装在对缺省迷你证书服务器的新建证书请求中。假设已配置了迷你证书服务器并可以使用它，它在返回可认证实体的新迷你证书的同时返回它自己的迷你证书，它们和受保护的专用密钥一起作为新的凭证存储在可认证实体的专用注册表中。自动注册为可认证实体凭证的建立提供了一种简单的机制，对于消息级保护（MQeMTrustAttribute，参见上面），还需要访问预定接收方的公用密钥（迷你证书）。

## 公用注册表与证书复制

MQSeries Everyplace 提供在 MQSeries Everyplace 组件之间能够共享可认证实体公用凭证（迷你证书）的缺省服务。这些服务是基于 MQeMTrust 消息级安全性的先决条件。MQSeries Everyplace 公用注册表为迷你证书提供公共可访问资源库。这类似于移动电话上的个人电话姓名地址录服务，区别在于它不是电话号码而是联系非常频繁的可认证实体的一组迷你证书。公用注册表在其服务中不是完全被动的。如果通过访问来提供一个它并不拥有的迷你证书，并且配置了一个有效宿主服务器组件，公用注册表将自动尝试从宿主服务器的公用注册表中查取所需的迷你证书。可以用这些服务来提供智能的自动迷你证书复制服务，那就适时促进了正确迷你证书可用性。

## 注册表服务的应用程序使用

在 MQSeries Everyplace 队列管理器设计成利用使用专用和公用注册表服务的长处的时候，不限制对这些服务的访问。MQSeries Everyplace 解决方案可能希望定义和管理它们自己的可认证实体，如用户。然后可以使用专用注册表服务来自动注册和管理新的可认证实体的凭证，并使用公用注册表服务使得公用凭证随处可用。所有已注册的可认证实体都可用作使用 MQeMTrustAttribute 保护的消息级服务的启动程序或接收方。

## 缺省迷你证书发行服务

MQSeries Everyplace 提供一个缺省迷你证书发行服务，可以将它配置成满足专用注册表的自动注册请求。使用 MQSeries Everyplace 提供的工具，解决方案可以设置和管理迷你证书发行服务，以将迷你证书颁发给一组严格控制的实体名。这种发行服务的特性是：

- 管理一组已注册的可认证实体
- 迷你证书发行

- WAP WTLS 迷你证书资源库管理

MQSeries Everyplace 提供的这些工具使迷你证书发行服务管理员能够通过注册其实体名、已注册地址和定义一个一次性证书请求 PIN，来将迷你证书的发行授权给一个给定的实体。这通常在脱机检查确认了请求者的真实性后执行。将证书请求 PIN 告之指定的用户（例如用一个与发行新银行卡时将银行卡告之用户类似的方法）。然后可以配置专用注册表用户（例如 MQSeries Everyplace 应用程序或 MQSeries Everyplace 队列管理器），以在启动时提供这个证书请求 PIN。当专用注册表触发自动注册时，迷你证书发行服务确认生成的新证书请求（基于将提供的实体名和证书请求 PIN 与它们预先注册的值是否匹配），发出这个新迷你证书，并复位已注册的证书请求 PIN，以使其不能重用。所有自动注册新迷你证书的请求都在一个安全通道上处理。

由迷你证书发行服务发出的一组迷你证书保存在发行服务自己的注册表中。如果重新发行一个迷你证书（例如因为到期导致），则将到期的迷你证书归档。

### 安全性接口

提供了一个可由定制安全性管理器实现的可选接口。其方法可以让安全性管理器授权或拒绝与以下操作相关联的请求：

- 添加或除去类的别名
- 定义适配器
- 映射文件描述符
- 处理通道命令

---

## 定制

### 规则

规则是在发生各种状态更改时，用于定制 MQSeries Everyplace 的行为的 Java 类。有必要时提供缺省规则，但是这些规则可由特定于应用程序或安装的规则代替以满足客户需要。支持的规则类型是在触发方式，而不是在执行的任务上有所区别。规则包含逻辑，因此可以执行许多功能。

#### 属性规则

只要试图更改状态，这种规则类就起作用，例如更改：

- 认证程序
- 压缩程序
- 加密程序

规则通常允许或不允许更改。



## MQSeries 网桥规则

这些规则类在 MQSeries Everyplace 到 MQSeries 网桥的代码有状态更改时开始控制。有一个单独的网桥规则类确定以下每种情况:

- 当侦听器无法将来自 MQSeries Everyplace 的消息继续传递到 MQSeries 时如何处理消息。例如因为消息太大或队列不存在。
- 一旦将服务器实例化，网桥管理的对象应当启动
- 当网桥找到 MQSeries 上的同步队列有错误时（用于恢复崩溃的持久存储器）如何处理。缺省规则只显示该问题。
- 如何将 MQSeries Everyplace 消息转换为 MQSeries 消息，反之亦然。在 MQSeries Everyplace 和 MQSeries 消息之间进行消息转换的转换程序不从任何 MQeRule 类中派生，而必须实现 MQeTransformerInterface 接口。除此之外，转换程序的作用和规则一样，并在消息需要格式转换时调用。

## 队列规则

每当有相关队列的状态更改时，这个规则类就起作用，例如:

- 将消息添加到队列中。例如查看是否超过阈值（消息数、消息大小、无效的优先级）
- 指定或更改队列特性
- 打开或关闭队列
- 将删除队列

## 队列管理器规则

每当有队列管理器的状态更改时，这个规则类就起作用，例如:

- 打开队列管理器。例如，启动一个后台计时器线程以允许定时操作发生
- 关闭队列管理器。例如，终止后台计时器线程
- 添加一个新队列

---

## 连接样式

MQSeries Everyplace 可以支持客户机 / 服务器和对等操作。客户机能够启动与服务器的通信。服务器则只能对由客户机启动的请求作出响应。在对等操作中，两个对等方可以启动任一方向的流。这些连接样式需要不同的 MQSeries Everyplace 组件以使其可用和活动。这些组件有:

- **通道侦听器:** 侦听进入的连接请求。
- **通道管理器:** 支持端点间的多个并行通信逻辑管道。
- **队列管理器:** 通过提供消息传递和排队能力支持应用程序。

## 连接样式

表16显示了这些组件和连接样式之间的关系。客户机 / 服务器连接样式描述了 MQSeries Everyplace 能以客户机或服务器任何一种方式运行的情况。小服务程序选项描述了将 MQSeries Everyplace 配置为一个 HTTP 小服务程序（这个 HTTP 服务器本身负责侦听进入的连接请求）的情况。

表 16. 连接样式

	队列管理器	通道管理器	通道侦听器
客户机	是		
客户机 / 服务器	是	是	是
对等	是		
服务器	是	是	是
小服务程序	是	是	

MQSeries Everyplace 应用程序并不直接知道队列管理器所使用的连接样式。不过样式也很重要，因为它影响什么资源对双方是可用的、哪些队列管理器能与其它队列管理器连接、MQSeries Everyplace 的占用空间，以及哪些连接能并存。

### 对等连接

对等通道包括单个通道的通道管理器和通道侦听器的能力。当在两个队列管理器之间创建了一个对等通道后，一个队列管理器必须充当侦听器，另一个充当连接的启动程序。使用对等连接的队列管理器能启动多个与其它队列管理器的对等连接，但它只能响应一个进入的连接请求，然后在响应另一个这种请求前必须等待那个对等通道关闭。在任何一个对等通道上，参与的两个队列管理器都能启动操作，因此，例如，每个队列管理器上的应用程序都能访问另一个队列管理器上的队列。

因为进入连接请求的目标对防火墙来说可能是不可接受的，所以对等通道可能不能通过防火墙使用。

### 客户机 / 服务器连接

用于客户机 / 服务器连接样式的标准通道没有侦听能力，而是依赖于服务器上的独立侦听器，并且服务器需要一个通道管理器来处理多个并行通道。客户机启动连接请求，服务器作出响应。服务器通常可以处理来自客户机的多个进入请求。在一个标准通道上，客户机可以访问服务器上的资源。如果服务器上的应用程序需要对客户机上的资源进行同步访问，则在角色互换时需要另一条通道。不过，因为标准通道本身是双向的，所以从服务器传输队列发往客户机的消息通过它所启动的标准（客户机 / 服务器）通道传递给它。

客户机可以同时是多个服务器的客户机。（请注意，因为通道管理器处理多个人站通道，所以不需要通道管理器支持这种配置。）

客户机 / 服务器连接样式通常适合于通过防火墙的使用，因为进入连接的目标往往标识成可以为防火墙所接受。

### 多个连接样式

单个队列管理器可以启动对等连接，也可以启动客户机 / 服务器连接；可以作为服务器响应，也可以作为对等方响应。在这种情况下，对等通道侦听器和标准通道侦听器必须有不同的端口号。

---

## 类

MQSeries Everyplace 为某些特定的功能提供了一些类，以允许定制 MQSeries Everyplace 的行为，来满足特定应用程序的需要。在某些情况下，将这些类的接口记录成文档，以便于开发其它替代方法。第50页的表17概述了这些可能性。可明确标识这些类，也可以通过使用别名标识它们。

表 17. 类选项

类	提供替代	接口已记录成文档
管理	否	是
认证程序	是	否
通信适配器	是	是
通信样式	是	否
压缩程序	是	否
加密程序	是	否
事件日志	提供样本	是
消息	否	是
队列存储器	是	否
规则	提供缺省类	是
跟踪	提供样本	是

## 应用程序装入

将 MQSeries Everyplace 队列管理器配置成作为客户机（或对等方）操作时，启动应用程序负责将所有其它应用程序装入 JVM。可以将标准 Java 设施用于该目的，也可以使用作为 MQSeries Everyplace 一部分包含的类装入器。因此，多个应用程序可以对同一个 JVM 中的单个队列管理器运行。也可以使用多个 JVM，但是每个都需要自己的队列管理器，并且每个队列管理器都必须有唯一的名称。

将 MQSeries Everyplace 队列管理器配置成服务器时，MQSeries Everyplace 本身就是启动应用程序。MQSeries Everyplace 支持一个预装人类列表，并在装入队列管理器本身之前依次装入这些类。

---

## 第6章 MQSeries Everyplace 和 MQSeries 网络

尽管 MQSeries Everyplace 网络可以单独存在，不需要 MQSeries 服务器或网络，但在实际中，常常使用 MQSeries Everyplace 来补充一个现有的 MQSeries 安装，将它扩展到新的平台与设备上，或提供一些高级能力，例如基于队列或消息的安全性或同步消息传递。从 MQSeries Everyplace 应用程序角度出发，可将 MQSeries 队列与队列管理器看作只是简单附加的远程队列和队列管理器。然而，因为这些队列不可直接通过 MQSeries Everyplace 动态通道和 MQSeries Everyplace 队列管理器访问，而是需要有一个 MQSeries Everyplace 网关，所以存在许多功能限制。网关可以直接或间接地通过 MQSeries 客户机通道将消息发送到多个 MQSeries 队列管理器。如果连接是间接的，则通过 MQSeries 客户机通道将消息传到一个中间 MQSeries 队列管理器，然后继续经过 MQSeries 消息通道到达目标队列管理器。

使用定义的 MQSeries 路由选择（远程队列定义），将目标为 MQSeries Everyplace 的 MQSeries 应用程序中的消息按正常情况发往 MQSeries Everyplace 队列管理器和队列中，这样 MQSeries Everyplace 消息就到达特定的 MQSeries 传输队列。按通常情况，没有为传输队列定义 MQSeries 通道，而是 MQSeries Everyplace 网关从这些队列中拿出消息并确保将它们传递到 MQSeries Everyplace 目的地。要使用的传输队列数（即路由数）是可配置的，并应该设置为反映要传递的消息量。

---

### 与 MQSeries 的接口

MQSeries Everyplace 的体系结构支持 MQSeries Everyplace 与其它消息传递系统之间一个或多个可选网桥类型的概念。

在 MQSeries Everyplace 版本 1.2 中，只支持作为 MQSeries Everyplace 和 MQSeries 网络之间接口的这么一种 *MQSeries* 网桥。这个网桥使用 MQSeries Java 客户机与一个或多个 MQSeries 队列管理器相接，从而允许消息从 MQSeries Everyplace 流到 MQSeries，反之亦然。在当前版本的 MQSeries Everyplace 中，建议每个服务器上都有这样一个网桥，并且每个网桥都与多个 *MQSeries* 队列管理器代理（MQSeries 队列管理器的定义）关联。每个与 MQSeries Everyplace 通信的 MQSeries 队列管理器都需要队列管理器代理定义。每个这样的定义都可以有一个或多个关联的客户机连接服务，其中每个服务代表了与单个 MQSeries 队列管理器的连接。每个都可以使用不同的 MQSeries 服务器连接到队列管理器，并可选择一组不同的特性，如用户出口或端口。

网关可以有許多侦听器，它们使用该网关与 MQSeries 队列管理器连接，并将消息从 MQSeries 检索到 MQSeries Everyplace。对于每个与 MQSeries 队列管理器上

## 与 MQSeries 连接

单个传输队列连接的侦听器，它只使用一个服务来建立连接。每个侦听器将消息从单个 MQSeries 传输队列通过其父网关队列管理器移到 MQSeries Everyplace 网络上的任何地方。因此，单个网关队列管理器可以将多个 MQSeries 消息源集中到 MQSeries Everyplace 网络。

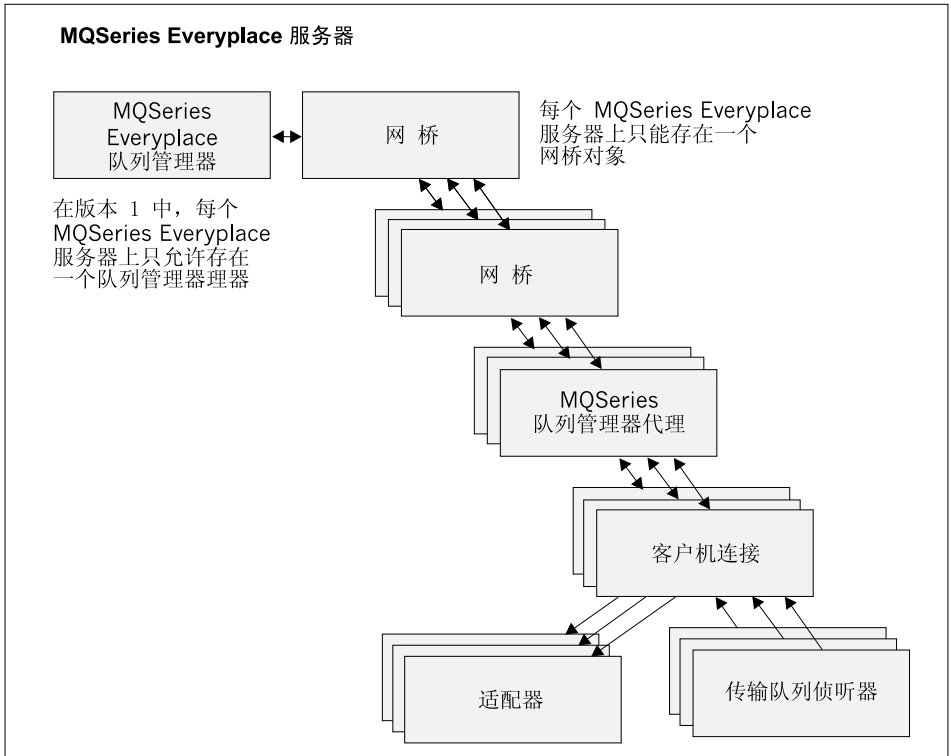


图 9. MQSeries Bridge 对象层次结构

当按另一方向（从 MQSeries Everyplace 到 MQSeries）移动消息时，网关队列管理器配置一个或多个网桥队列对象。每个网桥队列对象都可以与任何队列管理器直接连接，并将其消息发送到目标队列中。按这种方法，网关可以直接或间接地通过单个 MQSeries Everyplace 队列管理器将 MQSeries Everyplace 消息分发到任何 MQSeries 队列管理器中。表 18 显示了网桥对象的特性。

表 18. 网桥对象特性

特性	说明
网桥名称	网桥名称列表
运行状态	状态: 运行或停止

网桥对象和其它网关对象可以独立于 MQSeries Everyplace 队列管理器启动和停止。如果启动（或停止）了这样的网关对象，这个操作还应用于其所有子代（所有网桥、队列管理器代理、客户机连接和传输队列侦听器）。表19显示了网桥对象的特性。

表 19. 网桥特性

特性	说明
类	网桥类
缺省转换程序	如果没有其它的转换程序类与目的地队列关联，则用来将消息从 MQSeries Everyplace 转换到 MQSeries（或反之亦然）的缺省类（规则类）
心跳间隔	对网桥对象执行操作时使用的基本定时单位
名称	网桥对象的名称
运行状态	状态：运行或停止
启动规则类	启动网桥对象时使用的规则类
MQSeries 队列管理器代理子代	这个网桥所拥有的所有队列管理器代理列表

在简单情况下，可以使用缺省转换程序（规则）来处理所有消息转换。另外，可以不考虑这个缺省值，在每个侦听器的基础上（对从 MQSeries 到 MQSeries Everyplace 的消息）设置转换程序。对于更加特定的控制，可以按目标队列，在网关上使用网桥队列定义设置转换规则。这同时应用于 MQSeries Everyplace 和 MQSeries 目标队列。

MQSeries 队列管理器代理保管着特定于单个 MQSeries 队列管理器的特性。代理特性如表20中所示。

表 20. MQSeries 队列管理器代理特性

特性	说明
类	MQSeries 队列管理器代理类
MQSeries 主机名	通过 Java 客户机类用来创建与 MQSeries 队列管理器连接的 IP 主机名。如果不指定，则假定 MQSeries 队列管理器与网桥在同一台机器上并使用 Java 绑定
MQSeries 队列管理器代理名	MQSeries 队列管理器的名称
拥有网桥的名称	拥有这个 MQSeries 队列管理器代理的网桥对象的名称
运行状态	状态：运行或停止
启动规则类	启动 MQSeries 队列管理器对象时使用的规则类

## 与 MQSeries 连接

表 20. MQSeries 队列管理器代理特性 (续)

特性	说明
客户机连接子代	所有由这个代理所拥有的客户机连接对象的列表

网关连接服务定义保存着与 MQSeries 队列管理器连接所需的详细信息。连接特性如表21中所示。

表 21. 客户机连接服务特性

特性	说明
适配器类	用作网关适配器的类
CCSID*	要使用的 MQSeries CCSID 整数值
类	网桥客户机连接服务类
最大连接空闲时间	一个连接在终止前允许空闲的最大时间
MQSeries 密码*	Java 客户机使用的密码
MQSeries 端口*	通过 Java 客户机类用来创建与 MQSeries 队列管理器连接的 IP 端口号。如果不指定, 则假定 MQSeries 队列管理器与网桥在同一台机器上并使用 Java 绑定
MQSeries 接收出口类*	用来匹配在客户机通道另一端使用的接收出口; 该出口有一个关联的字符串以允许将数据传入出口代码中
MQSeries 安全性出口类*	用来匹配在客户机通道另一端使用的安全性出口; 该出口有一个关联的字符串以允许将数据传入出口代码中
MQSeries 发送出口类*	用来匹配在客户机通道另一端使用的发送出口; 这个出口有一个关联的字符串以允许将数据传入出口代码中
MQSeries 用户标识*	由 Java 客户机使用的用户标识
客户机连接服务名	MQSeries 机器上服务器连接通道的名称
所属队列管理器代理的名称	所属队列管理器代理的名称
启动规则类	当启动网桥客户机连接服务对象时使用的规则类
同步队列名	由网桥用于同步目的的 MQSeries 队列名
同步队列清除规则类	当同步队列中找到消息时使用的规则类
运行状态	状态: 运行或停止
拥有网桥的名称	拥有这个客户机连接的网桥对象的名称
MQ XmitQ 侦听器子代	使用这个客户机连接的所有侦听器对象的列表
*这些参数的细节可以在 <i>MQSeries Using Java</i> 文档中找到	



适配器类用于将消息从 MQSeries Everyplace 发送到 MQSeries 并使用同步队列来跟踪处理状态。在恢复时使用其内容以保证确保的消息传递；正常关机后，这个队列为空。只要接收、发送和安全性出口相同，就可以在多个客户机连接之间以及多个网桥定义之间共享它。取决于使用的侦听器特性，这个队列还可用来存储关于从 MQSeries 移到 MQSeries Everyplace 的消息的状态。当在同步队列中找到一条消息时，使用同步队列清除程序规则类表明 MQSeries Everyplace 确认消息失败。

最大连接空闲时间用来控制其 MQSeries 系统的 Java 客户机连接池，该池是由网桥客户机连接服务所维护的。当 MQSeries 连接成为空闲时（由于缺少使用），启动一个计时器，如果计时器在重用这个连接前到期，则废弃这个空闲连接。创建 MQSeries 连接是一个开销很大的操作，而且这个过程要确保这些连接可有效地重用，且无须消耗太多资源。值为 0 表明不应当使用连接池。

表22显示了将消息从 MQSeries 移到 MQSeries Everyplace 的侦听器对象所具有的特性。

表 22. 侦听器特性

特性	说明
类	侦听器类
死信队列名	用来保存无法从 MQSeries 传递到 MQSeries Everyplace 的消息的队列
侦听器状态存储适配器	用于存储状态信息的适配器的类名
侦听器名称	提供消息的 MQSeries XMIT 队列名称
拥有客户机连接服务名	客户机连接服务名
运行状态	状态: 运行或停止
启动规则类	当启动侦听器对象时使用的规则类
转换程序类	用来确定将消息从 MQSeries 转换到 MQSeries Everyplace 的规则类
未传递的消息规则类	当无法将消息从 MQSeries 传递到 MQSeries Everyplace 时用来确定操作的规则类
等待消息的秒数	可用来控制侦听器在异常情况下的性能的高级选项

未传递的消息规则类确定无法将消息从 MQSeries 传递到 MQSeries Everyplace 时所采取的操作。通常将它放到 MQSeries 系统的死信队列中。

为了提供确保的消息传递，侦听器类使用侦听器状态存储适配器来存储状态信息，不是存储在 MQSeries Everyplace 系统上就是在 MQSeries 系统的同步队列中。

## 与 MQSeries 连接

传输队列侦听器可以让 MQSeries 远程队列引用 MQSeries Everyplace 本地队列。也可以创建引用 MQSeries 本地队列的 MQSeries Everyplace 远程队列。这些 MQSeries Everyplace 远程队列定义称为 *MQSeries 网桥队列*，可以使用它们取出、放入和浏览 MQSeries 队列上的消息。

MQSeries 网桥队列定义可以包含以下属性。

表 23. MQSeries 网桥队列特性

特性	说明
别名	队列的替代名
认证程序	必须为空
类	对象类
客户机连接	要使用的客户机连接服务的名称
压缩程序	必须为空
加密程序	必须为空
到期	传递给转换程序
最大消息大小	传递给规则类
方式	必须是同步
MQ 队列管理器代理	应当将消息首先发送到的 MQSeries 队列管理器的名称
MQSeries 网桥	将消息传送到 MQSeries 的网桥名称
名称	为 MQSeries Everyplace 所知的远程 MQSeries 队列的名称
所属队列管理器	拥有定义的队列管理器
优先级	用于消息的优先级（除非被消息值覆盖）
远程 MQSeries 队列名	远程 MQSeries 队列的名称
规则	用于队列操作的规则类
队列管理器目标	拥有队列的 MQSeries 队列管理器
转换程序	将消息从 MQSeries Everyplace 格式转换到 MQSeries 格式的转换程序类的名称
类型	MQSeries 网桥队列

**注：**加密程序、认证程序和压缩程序类定义了一组队列属性，表明传递给这个队列的所有消息的安全性级别。从 MQSeries Everyplace 上消息最初发送时，到将消息传递到 MQSeries 网桥队列时，消息至少在队列这一安全性级别上是受保护的。当 MQSeries 网桥队列将消息传递给 MQSeries 系统时，这些安全性级别不适用，在传送期间使用客户机连接上的安全性发送和接收出口。不确保维护队列安全性级别的检查。

MQSeries 网桥队列只能是同步的。因此异步应用程序在向 MQSeries 网桥队列发送消息时，必须使用 MQSeries Everyplace 存储和转发以及宿主服务器队列的组合，或者使用异步远程队列定义作为一个中间步骤。

利用 MQSeries 网桥队列的应用程序和其它任何 MQSeries Everyplace 远程队列一样，都使用 MQeQueueManager 类的 `putMessage`、`browseMessages` 和 `getMessage` 方法。这些调用中的队列名参数是 MQSeries 网桥队列的名称，队列管理器名参数是 MQSeries 队列管理器的名称。不过，为使这个队列管理器名能被本地 MQSeries Everyplace 服务器接受，必须存在一个带有该 MQSeries 队列名的连接定义，它的所有参数，包括通道名，都为空。

**注：**对 MQSeries 网桥队列使用 `getMessage` 和 `browseMessages` 有一些限制。不可能从指向 MQSeries 远程队列定义的 MQSeries 网桥队列取出或浏览消息。也不可能在 MQSeries 网桥队列取出操作上使用非零的“确认标识”。这意味着对 MQSeries 网桥队列执行的 `getMessage` 操作不提供确保传递。如果需要确保取出操作，应该使用传输队列侦听器来传送 MQSeries 的消息。

MQSeries 网桥 的管理与一般 MQSeries Everyplace 队列管理器管理的处理方法相同 - 都是通过使用管理消息。对被管理对象按需要定义新的消息类。第35页的表13显示了网关管理消息类。

---

## 消息转换

发往 MQSeries 的 MQSeries Everyplace 消息通过网桥传递，并使用缺省转换程序或特定于目标队列的转换程序将它们转换为 MQSeries 格式。定制转换程序提供更多的灵活性，例如使用 MQSeries Everyplace 消息对象类的子类来表示 MQSeries Everyplace 网络上的一个特定消息类型就是一个好的做法。在网关上，无论在字段和相应的 MQSeries 值之间使用什么映射，转换程序都将消息转换成 MQSeries 格式，同时添加特定的数据来表示有效的子类。

从 MQSeries Everyplace 到 MQSeries 的缺省转换程序无法利用子类信息，但设计成对许多情形都有用。它有以下特性：

- **从 MQSeries Everyplace 到 MQSeries 的消息流：**

从 MQSeries Everyplace 到 MQSeries 的缺省转换程序与 MQeMQMMsgObject 类结合使用。这个类是您能在 MQSeries 消息头中找到的所有字段的表示。使用 MQeMQMMsgObject，应用程序可以用 `set()` 方法设置值（例如优先级）。因而，当将 MQeMQMMsgObject（或一个自 MQeMQMMsgObject 类派生的对象）传递到缺省 MQSeries Everyplace 转换程序时，这个缺省转换程序 (MQeBaseTransformer) 从 MQeMSMMsgObject 内获取值，然后在 MQSeries 消息中设置相应的值（例如复制到 MQSeries 消息的优先级值）。

## 与 MQSeries 连接

如果传递的消息不是 MQeMQMsgObject，并且不是从 MQeMQMsgObject 类派生出的，则将整条 MQSeries Everyplace 消息复制到 MQSeries 消息主体（集中的）。设置 MQSeries 消息头中的这个消息格式字段来表明 MQSeries 消息拥有一条 MQSeries Everyplace “集中”格式的消息。

### • MQSeries 到 MQSeries Everyplace 消息流:

使用与反向传送类似的方法为 MQSeries Everyplace 处理 MQSeries 消息。缺省转换程序检查 MQSeries 头的消息类型字段并采取相应操作。

如果 MQSeries 头表明是一条“集中”的 MQSeries Everyplace 消息，则按原始的 MQSeries Everyplace 消息重新构造 MQSeries 消息主体，然后将它发布给 MQSeries Everyplace 网络。

如果这条消息不是“集中”的 MQSeries Everyplace 消息，则抽取 MQSeries 消息头内容，然后将它放到 MQeMQMsgObject 对象中。将 MQSeries 消息主体作为简单的字节字段来处理，然后同样将它放到 MQeMQMsgObject 对象中。然后将 MQeMQMsgObject 发布给 MQSeries Everyplace 网络。

这个 MQeMQMsgObject 类和缺省转换程序的行为意味着:

- 无需更改就可将 MQSeries Everyplace 消息跨越 MQSeries 网络发送到 MQSeries Everyplace 网络。
- 无需更改就可将 MQSeries 消息跨越 MQSeries Everyplace 网络发送到 MQSeries 网络。
- MQSeries Everyplace 应用程序可以驱动任何现有的 MQSeries 应用程序，无需对这个 MQSeries 应用程序作任何更改。

---

## 功能

从一个 MQSeries Everyplace 队列管理器，为同步 MQSeries Everyplace 放入消息传递操作启用 MQSeries 远程队列；所有其它消息传递操作都必须是异步的。

无法将 MQSeries Everyplace 管理消息发送到 MQSeries 队列管理器。AdminQ 在那里不存在，并且这个管理消息格式不同于 MQSeries 使用的格式。

---

## 兼容性

MQSeries Everyplace 网络能独立于 MQSeries 存在，但在许多情况下，同时需要这两个产品以满足应用程序的需求。MQSeries Everyplace 能集成到一个现有的 MQSeries 网络中，关于兼容性，包括以下几个方面:

### 寻址与命名:

- 使用队列管理器 / 队列地址的相同的寻址语义

- ASCII 名称空间的共同使用

#### 应用程序:

MQSeries Everyplace 可以在应用程序不更改的情况下支持现有的 MQSeries 应用程序。

**通道:** MQSeries Everyplace 网关使用 MQSeries 客户机通道。

#### 消息交换与内容:

- 在 MQSeries Everyplace 与 MQSeries 之间的消息交换
- 消息网络不可见（来自 MQSeries Everyplace 或 MQSeries 的消息无需更改就能跨越其它网络）
- 相互支持 MQSeries 消息头中的标识字段
- 仅一次确保消息传递

MQSeries Everyplace 版本 1.2 不支持 MQSeries 的所有功能。除了环境、操作系统和通信考虑之外，以下详细说明一些重要的区别。然而请注意，在 MQSeries Everyplace 内许多应用程序任务都可以通过使用 MQSeries Everyplace 功能的替代方法或通过利用子类来实现，替换所提供的类或利用规则、接口和将其它定制功能构建到产品中。

- 无群集支持。
- 无分发列表支持
- 无分组 / 分段的消息
- 无负载均衡 / 热备用能力
- 无引用消息
- 无报告选项
- 无共享队列支持。
- 无触发
- 无工作单元支持，无 XA 协调

可伸缩性和性能特性不同。

---

## 确保传递

尽管 MQSeries Everyplace 和 MQSeries 都提供确保传递，但它们各自提供的确保级别不同。当将消息从 MQSeries Everyplace 传送到 MQSeries 时，只有在使用 putMessage 和 confirmPutMessage 的组合（参阅第29页的『队列管理器配置』）时才确保消息传送。当将消息从 MQSeries 传送到 MQSeries Everyplace 时，只有将 MQSeries 消息定义为持久时才确保这个传送。



---

## 第7章 编程接口

*MQSeries Everyplace* 系统编程接口 (SPI) 是 MQSeries Everyplace 的编程接口。支持两种语言: Java 和 C:

**Java** 版本提供了对所有 MQSeries Everyplace 功能的访问。详细的类、方法和过程在 *MQSeries Everyplace for Multiplatforms Programming Reference* 中有详细描述; MQSeries Everyplace 编程的示例在 *MQSeries Everyplace for Multiplatforms Programming Guide* 中提供。

对 **Palm** 的 **C** 支持为在 Palm 设备上使用而提供了对一部分 MQSeries Everyplace 功能的访问。这些类和过程的详细信息以及编程指南在 *MQSeries Everyplace Native Client Information* 中提供。





---

## 第8章 MQSeries Everyplace 入门

MQSeries Everyplace 是一系列产品，这些产品共同提供开发、部署和管理 MQSeries Everyplace 消息传递和排队解决方案所需的一些工具。该系列包括：

1. *MQSeries Everyplace* 特许产品（可以通过 IBM 的物理媒体获得，或者从 <http://software.boulder.ibm.com/dl/mqsem/mqsem-p> 进行 Web 下载）。特许产品包括：

- MQSeries Everyplace Java 类
- 帮助程序类
- 应用程序源代码示例
- 实用程序
- 参考手册
- 许可证信息

物理“程序产品”还包括在特定平台上将产品用于非开发（生产性）用途的授权。为在更大型的机器上使用或与 MQSeries 网桥一起使用，还需要购买进一步的容量单元。

2. MQSeries Everyplace SupportPacs（可以从 <http://software.boulder.ibm.com/dl/mqsem/mqsem-p>（同上）或 <http://www.ibm.com/software/mqseries/everyplace> 通过 Web 下载获得）。这些是对特许产品的重要补充，可能包括：

### **EAP1: MQSeries Everyplace - 用于 Palm OS 的设备代码**

在 Palm OS 上进行 MQSeries Everyplace 版本 1.0.1 应用开发所需的 C 编程语言支持。（这些代码还包括在产品 CD 上的文件 eap1.zip 中）

### **EP01: MQSeries Everyplace - 性能报告**

分析 MQSeries Everyplace 在各种客户机平台上的性能

### **ES01: MQSeries Everyplace - 管理工具 (MQExplorer v1.0)**

所有 Java 平台的常规工具，支持 MQSeries Everyplace 队列管理器的简易图形管理

### **ES02: MQSeries Everyplace - 资源管理器 (MQ\_Explorer v1.2)**

专为支持微软 Windows 操作系统系列开发的 MQSeries Everyplace 管理工具

MQSeries Everyplace SupportPacs 中的管理工具在应用开发和展出的所有阶段中都扮演着一个很重要的角色。它们比特许产品所包含的实用程序更为复杂，是开始使用、配置、检查实验网络和管理产品系统的一个重要帮助。

---

## 使用 MQSeries Everyplace

考虑到 MQSeries Everyplace 的广泛用途，该产品与 MQSeries 系列中的其它成员的安装、配置和部署方法并不相同。这里一个基本的概念是，MQSeries Everyplace 要为企业采纳通常有三个阶段：

### 1. 开发和创建原型阶段

在早期的学习、开发和创建原型阶段，安装和使用 MQSeries Everyplace 产品无需付费 - 根据 IBM MQSeries Everyplace 开发许可证条款的规定。MQSeries Everyplace 应用程序是使用 MQSeries Everyplace Java 类和 C 例程的功能开发的。这些应用程序可以以各种方法打包，例如：

- 可以将 MQSeries Everyplace 队列管理器设置成一个守护进程，将一个或多个应用程序安装到同一个 Java 虚拟机中并共享一个公共队列管理器。
- 应用程序嵌套必需的 MQSeries Everyplace 类，使得应用程序可以在没有安装 MQSeries Everyplace 的机器上运行，并将其自己的队列管理器装入到自己的 JVM 中。
- 应用程序使用目标机器上存在的 MQSeries Everyplace 类。

IBM 的支持不包括在开发许可证中。不过，在应用开发等阶段期间的支持是在部署许可证中提供的（见下面）。

### 2. 部署阶段

部署阶段代表着已开发应用程序的展出和使用，因此，按照 IBM MQSeries Everyplace 许可证的条款，使用产品需要容量单元。类只能在 IBM 同意，或者在用户已授权使用它们的地方随应用程序一起分发。否则，用户必须自己提供必需的类。

### 3. 管理阶段

当 MQSeries Everyplace 队列管理器在网络中生效后，接下来就需要一些工具来检查和管理它们。对 MQSeries Everyplace 的支持是在“国际程序许可证协议”的条款规定下提供的。

这一采纳的生命周期是对平台支持级别变体的调整。对于带有容量单元的（和 3 类 SupportPacs）的 MQSeries Everyplace，IBM 区分以下几种情况：

- 支持安装和应用开发的平台：
  - 接受有关安装和 / 或应用开发和使用的报告

- 允许应用程序部署，但不直接支持的平台：
  - 可能需要在支持的平台上重新产生问题报告
- 支持应用程序部署的平台：
  - 接受由应用程序部署所产生的问题报告

---

## 获得经验

开始使用 MQSeries Everyplace 有许多方法。经验建议我们先设置并运行队列管理器，然后再设置并运行一个简单的 MQSeries Everyplace 网络，这是熟悉产品及其概念的一个有效方法。然后编写一个简单的应用程序，这是对产品细节进行深入研究的一个有效的准备工作。在早期阶段，详细了解 MQSeries 系列中的其它成员通常没有什么帮助。稍后，当网桥功能开始变得重要时，对这些成员的理解才是必不可少的。

记住这一策略后，建议新用户采用以下方法：

理解 *MQSeries Everyplace 多平台版介绍*（本书）中出现的概念要点，然后执行以下操作：

1. 如果访问的是运行 Microsoft Windows 操作系统的机器，下载 MQE\_Explorer, SupportPac ES02 (MQE\_Explorer v 1.2)。您不必事先安装特许产品，但如果没有安装，您在开发使用方面会受到受许可证条款的限制。
2. 按照下面『首先使用 ES02: MQE\_Explorer』中给出的特定指令操作。

### 首先使用 ES02: MQE\_Explorer

MQE\_Explorer 提供了一本全面的“用户指南”，描述了配置第一个队列管理器的快速方法。通常将该手册推荐给更广泛的读者，而不只是 MQE\_Explorer 管理员，因为它包括了一些样本脚本，说明了 MQSeries Everyplace 操作的重要示例。SupportPac 包括 MQE\_Explorer 的两个可执行版本：


#### **MQE\_ExplorerX.exe**

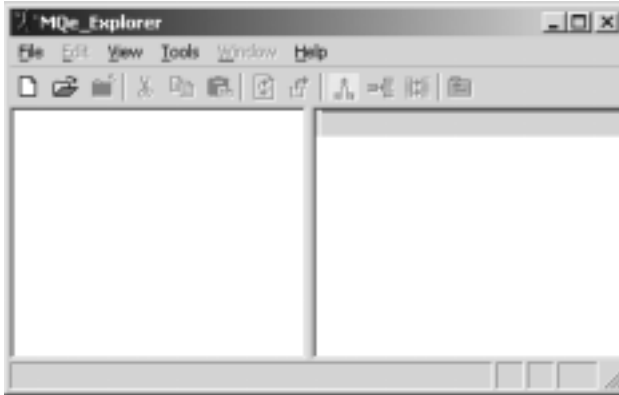
该版本将操作所需的所有 MQSeries Everyplace 类都嵌套在 .exe 文件中。MQE\_ExplorerX.exe 可以在机器上不安装 MQSeries Everyplace 的情况下运行。这对于第一次使用的用户来说非常理想。


#### **MQE\_Explorer.exe**

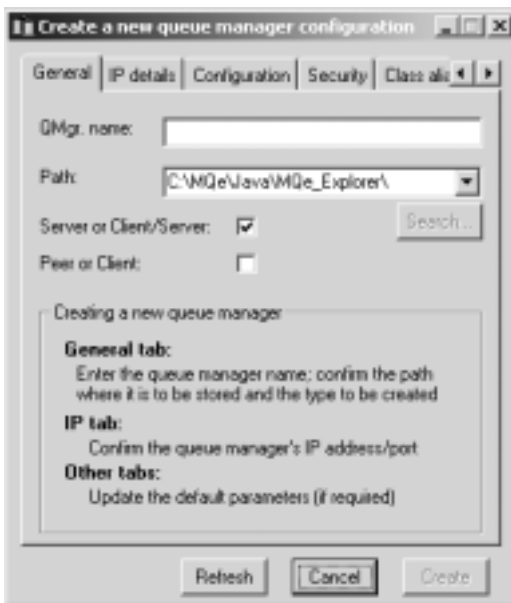
该版本取决于先前安装的 MQSeries Everyplace。它的优势是能够选取最新级别的 MQSeries Everyplace 库，并且非常小。它针对的是开发者和管理人员。

以下经过删节的指令显示了 MQSeries Everyplace 的威力，同时证明了队列管理器的创建非常容易。

1. 双击 MQe\_ExplorerX.exe 图标。  
有消息表明未找到已保存的选项，单击**确定**（这条消息不会再次出现）。显示下面的管理窗口：



2. 单击工具栏上的新建图标。  
这将创建一个新的队列管理器。显示下列窗口：

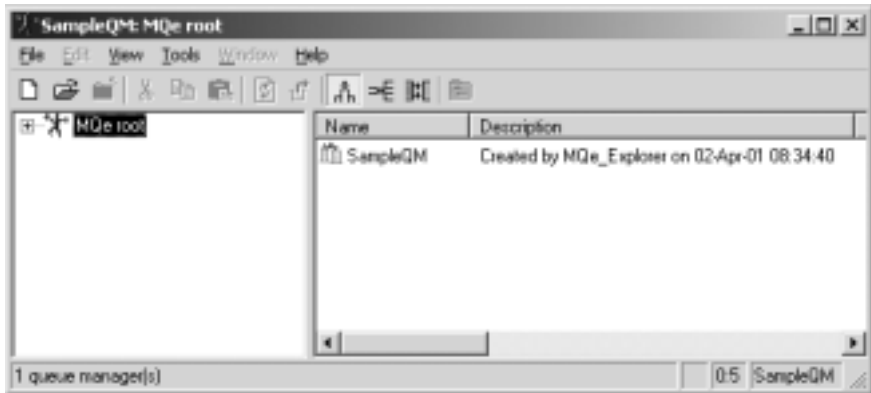


- 3.

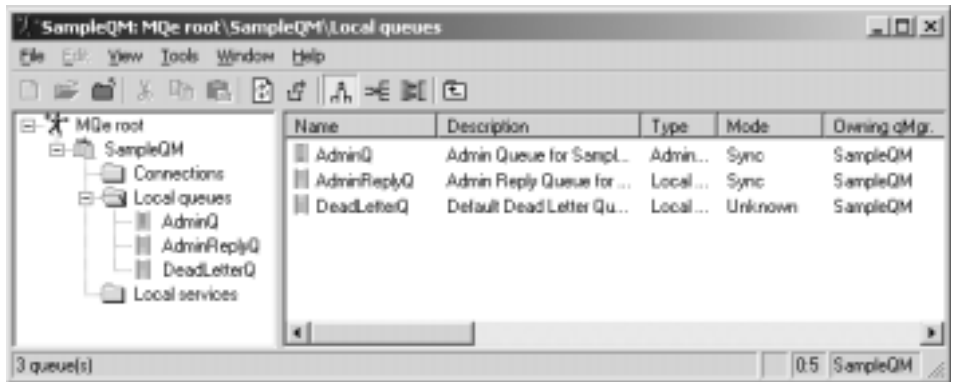
- a. 输入队列管理器名（例如 SampleQM）。
- b. 选择 **IP details** 选项卡



4.
  - a. 输入 IP 地址 - 该地址是正在使用的机器的 IP 地址，但在这一步时最好输入 127.0.0.1（本地主机）。
  - b. 单击**创建**按钮。发出一条消息，标识初始化文件已创建（以后访问该队列管理器时需要该名称）；
  - c. 单击**确定**，显示下面的窗口：



5. 创建了一个服务器队列管理器，它在自身的 JVM 中执行。在端口 8081 上侦听进入客户机 / 服务器通道连接请求。如果将左侧窗格树中的所有 + 符号展开，并重新调整窗口和窗格的大小，可以看到创建了四个队列。



6. 按照 *MQe\_Explorer User Guide* 中的指令操作可以尝试更多操作。可以创建队列、连接、消息，甚至完整的 MQSeries Everyplace 网络。可以将应用程序装入到同一个队列管理器中，并可以和 MQe\_Explorer 同时运行。

---

## 附录. 通告

本文档是专为在美国提供的产品和服务而编写的。IBM 可能不会在其他国家提供在本文档中所述的产品、服务或功能。与您当地的 IBM 代理联系获得有关当前在您所处区域可用的产品和服务的信息。任何对 IBM 的产品、程序或服务的引用并不说明或暗示只能使用 IBM 产品、程序或服务。凡是同等功能的产品、程序或服务，只要不侵犯 IBM 的知识产权，都可以用来代替 IBM 产品、程序或服务。然而，对任何非 IBM 产品、程序或服务运作的评估和验证由用户自行负责。

IBM 可能已经申请或正在申请与本文档有关的各项专利权。提供本文档并不表示允许您使用这些专利。您可以用书面方式将特许查询寄往：

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

下面这段内容不适用于英国或当地法律与本条款不一致的国家：国际商业机器公司以『AS IS』（即“按原样”）方式提供出版物，而本出版物不具有任何形式的担保，无论是明确的还是隐式的，其中包括但不局限于对特定目的之合法性、适销性或适用性的隐式担保。有些国家不允许在特定事务中拒绝明确的或隐式的担保，因此本声明有可能不适用于您的国家。

本文可能会包含技术上的不精确性或印刷错误。此处提到的信息会定期更改；这些更改被合并至本文的新版本中。IBM 可能会在不作通知的情况下，随时对本文中所说明的产品和（或）程序作改进和（或）更改。

本文档中任何涉及非 IBM Web 站点的引用仅为方便而提供，而并不以任何方式为 Web 站点的使用提供担保。Web 站点的资料不是这个 IBM 产品资料的一部分，使用那些 Web 站点请自担风险。

IBM 可能以它认为适当的任何方式使用或散发您提供的信息，而不必对您负担任何责任。

## 通告

希望拥有以下信息的这个程序的许可证持有人：（i）允许在独立创建的程序和其它程序（包括本程序）之间进行信息交换（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

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

依照适当的条款和条件，其中包括在一些情况下需要付费，这些信息或许是可用的。

本文中说明的特许程序和所有用于该特许程序的许可资料由 IBM 根据 IBM 客户协议或 IBM 国际程序特许协议，或任何我们之间的等效协议的条款提供。

---

## 商标

下列术语是国际商业机器公司在美国和 / 或其他国家的商标。

AIX    AS/400    IBM    MQSeries    OS/390

Microsoft、Windows、Windows NT 和 Windows 徽标是微软公司在美国和 / 或其他国家的商标。

Java 和所有基于 Java 的商标及徽标都是 Sun 公司在美国和 / 或其他国家的商标或注册商标。

其他公司、产品及服务名称可能是其他公司的商标或服务标记。



---

## 词汇表

本词汇表描述在本书中使用的术语和不同于日常含义的词汇。在某些情况下，一个定义可能不是那个术语可适用的唯一定义，但它给出了这个词汇在本书中使用的特定含义。

如果没有找到您要查找的术语，请参阅索引或 *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

**应用程序编程接口 (Application Programming Interface, API).** “应用程序编程接口”由允许程序员在其应用程序中使用的函数和变量组成。

**异步消息传递 (asynchronous messaging).** 将消息放入消息队列的一种程序间通信方法。用异步消息传递，发送程序可以继续它自己的处理而无须等待对此消息的应答。对照**同步消息传递(synchronous messaging)**。

**认证程序 (authenticator).** 检查验证消息的发送方和接收方的一个程序。

**网桥 (bridge).** 是一个 MQSeries Everyplace 对象，它允许消息在 MQSeries Everyplace 和其它消息传递系统（包括 MQSeries）间流动。

**通道 (channel).** 见**动态通道(dynamic channel)**和**MQI 通道(MQI channel)**。

**通道管理器 (channel manager).** 是一个 MQSeries Everyplace 对象，它支持端点间逻辑多并行通信管道。

**类 (class).** 类是数据和对数据进行操作的方法的一个封装集合。可以实例化一个类以产生一个对象，这个对象是该类的一个实例。

**客户机 (client).** 在 MQSeries 中，客户机是一个运行时组件，它为本地用户应用程序提供对服务器上的排队服务的访问。

**压缩程序 (compressor).** 是一个程序，它压缩消息以减少发送的数据量。

**加密程序 (cryptor).** 是一个程序，它对消息进行加密以提供传输过程的安全性。

**动态通道 (dynamic channel).** 动态通道连接各 MQSeries Everyplace 设备及传送同步和异步消息，并以双向方式响应消息。

**封装 (encapsulation).** 封装是一个面向对象的编程技术，它使对象数据私有或受保护，并只允许程序员通过方法调用来访问和操作数据。

**网关 (gateway).** MQSeries Everyplace 网关（或服务器）是运行 MQSeries Everyplace 代码（包括通道管理器）的计算机。

**超文本标记语言 (Hypertext Markup Language, HTML).** 用来定义在万维网上显示信息的语言。

**实例 (instance).** 实例是一个对象。当实例化一个类以产生一个对象时，称该对象为类的一个实例。

**接口 (interface).** 接口是一个类，其中只包含抽象的方法，没有实例变量。接口提供一个公共的方法集，它能用几个不同类的子类实现。

**因特网 (Internet).** 因特网是一个共享信息的协作性公共网络。物理上，因特网使用当前所有现存的公共远程通信网络全部资源的一个子集。技术上，将因特网区分为协作性公共网络是因为它使用 TCP/IP（传输控制协议 / 网际协议）协议集。

**Java 开发工具箱 (Java Developers Kit, JDK).** 由 Sun Microsystems 公司为 Java 开发人员提供的一个软件包。它包括 Java 解释器、Java 类和 Java 开发工具：编译器、调试器、反汇编器、小应用程序浏览器、存根文件生成器及文档生成器。

**Java 命名和目录服务 (Java Naming and Directory Service, JNDI).** 在 Java 程序设计语言中指定的 API。它对用 Java 程序设计语言编写的应用程序提供命名和目录功能。

**轻量型目录访问协议 (Lightweight Directory Access Protocol, LDAP).** LDAP 是访问目录服务的客户机 / 服务器协议。

**消息 (message).** 在消息排队应用程序中, 消息是程序间发送的通信。

**消息队列 (message queue).** 见队列 (queue)

**消息排队 (message queuing).** 一种编程技术, 应用程序中的每个程序通过将消息放入队列来与另一个程序进行通信。

**方法 (method).** 方法是函数或过程的面向对象编程术语。

**MQI 通道 (MQI channel).** MQI 通道将 MQSeries 客户机连接到服务器系统的队列管理器上并以双向方式传送 MQI 调用和响应。

**MQSeries.** MQSeries 是一个提供消息排队服务的 IBM 特许程序系列。

**对象 (object).** (1) 在 Java 中, 对象是类的一个实例。类表示一组事物; 对象表示那个组的一个具体成员。(2) 在 MQSeries 中, 对象是队列管理器、队列或通道。

**包 (package).** 在 Java 中, 包是给出一段 Java 代码以访问一组特定类的方法。某个具体包中的 Java 代码可以对包中的所有类及类中所有非私有的方法和字段进行访问。

**个人数字助理 (personal digital assistant, PDA).** 一种口袋大小的个人计算机。

**私有 (private).** 一个私有字段在其自身类外不可见。

**受保护 (protected).** 一个受保护的字段只在其类、子类或这个类所在包的内部可见

**公共 (public).** 公共类或公共接口在任何地方都可见。公共方法或公共变量在可见其类的任何地方都可见。

**队列 (queue).** 队列是一个 MQSeries 对象。消息排队应用程序可将消息放入队列并可从队列中取出消息。

**队列管理器 (queue manager).** 队列管理器是一个系统程序, 它对应用程序提供消息排队服务。

**服务器 (server).** (1) MQSeries Everyplace 服务器是一个配置了 MQSeries Everyplace 通道管理器的设备。(2) MQSeries 服务器是一个队列管理器, 它对运行在远程工作站上的客户机应用程序提供消息排队服务。(3) 通常, 服务器是一个程序, 它响应客户机 / 服务器的特殊两程序信息流模型中的信息请求。(4) 服务器程序在其上运行的计算机。

**小服务程序 (servlet).** 是一个 Java 程序, 它被设计成仅在 Web 服务器上运行。

**子类 (subclass).** 子类是扩展另一个类的类。子类继承其超类的公共和受保护的方法及变量。

**超类 (superclass).** 超类是供其它类扩展的类。超类的公共和受保护的方法及变量对子类可用。

**同步消息传递 (synchronous messaging).** 将消息放置到消息队列中的程序间的通信方法。使用同步消息传递, 发送程序在继续它自身的处理前须等待对此消息的应答。对照异步消息传递 (*asynchronous messaging*)。

**传输控制协议 / 网际协议 (Transmission Control Protocol/Internet Protocol, TCP/IP).** 支持局域网和广域网的对等连接功能的一组通信协议。

**Web.** 见万维网 (World Wide Web)。

**Web 浏览器 (Web browser).** 是一个程序, 它格式化和显示在万维网上分布的信息。

**万维网 (World Wide Web, Web).** 万维网是基于一组公共协议的因特网服务, 它使得特别配置的服务器计算机能用标准的方法通过因特网分发文档。

---

## 文献目录

相关出版物:

- *MQSeries Everyplace* 多平台版首先阅读, GB84-0418
- *MQSeries Everyplace for Multiplatforms Programming Reference*, SC34-5846
- *MQSeries Everyplace for Multiplatforms Programming Guide*, SC34-5845
- *MQSeries Everyplace for Multiplatforms Native Client Information*, GC34-5883
- *MQSeries An Introduction to Messaging and Queuing*, GC33-0805-01
- *MQSeries for Windows NT V5R1 Quick Beginnings*, GC34-5389-00



# 索引

## [ A ]

安全性接口 46  
安全性, 本地 41  
安全性, 基于队列 41  
安全性, 消息级 42  
安全性, MQSeries Everyplace 40

## [ B ]

本地安全性 41  
本地队列 21  
本书面向的读者 v  
必备知识 v  
必需的操作系统 3  
编程接口 61  
拨号连接管理 38

## [ C ]

操作系统, 支持的 3  
操作, 队列管理器 32  
产品概念 15  
存储和转发队列 22

## [ D ]

定制 46  
动态通道 9, 15, 37  
读者 v  
读者范围 v  
对等连接 48  
对等通道 9  
队列管理器 8, 9, 47  
队列管理器操作 32  
队列管理器代理对象 53  
队列管理器规则 47  
队列管理器, MQSeries  
Everyplace 26  
队列规则 47  
队列, 本地 21  
队列, 存储和转发 22

队列, 远程 21  
队列, MQSeries 网桥 23  
队列, MQSeries Everyplace 21  
对象, 消息 16  
对象, MQSeries Everyplace 16  
对 MQSeries 的接口 51  
多个连接样式 49

## [ F ]

法律通告 69  
分布式消息传递 vi, 7  
服务器, MQSeries 8

## [ G ]

概念, 产品 15  
概述 1  
跟踪 MQSeries Everyplace 38  
公用注册表 45  
关于本书 v  
管理消息 35  
管理 MQSeries Everyplace 34  
规则, MQSeries Everyplace 46

## [ H ]

环境, 软件 3

## [ J ]

基于队列的安全性 41  
加密 40  
监控 37  
接口, 安全性 46  
接口, 编程 61

## [ K ]

可认证的实体 44  
客户机通道 15

客户机, MQSeries 8  
客户机 / 服务器连接 48  
客户机 / 服务器通道 9

## [ L ]

类, MQSeries Everyplace 49  
连接样式 47  
连接样式, 多个 49  
连接, 对等 48  
连接, 客户机 / 服务器 48

## [ M ]

迷你证书 44  
迷你证书的发行服务 45  
迷你证书发行服务 45

## [ N ]

能力 13

## [ P ]

配置 46  
配置, 示例 10  
普及型消息传递 vi, 8

## [ Q ]

确保消息传递 59

## [ R ]

软件环境 3

## [ S ]

商标 70  
设备, MQSeries Everyplace 15  
实体, 可认证的 44

- 事件日志 39
- 示例配置 10
- 适配器, MQSeries Everyplace 38
- 属性规则 46
- 术语 v
- 说明 1
- 宿主服务器队列 22
- 宿主服务器, MQSeries Everyplace 22

## [ T ]

- 通道 9
- 通道管理器 47
- 通道侦听器 47
- 通道, 动态 15, 37
- 通道, 客户机 15
- 通告, 法律 69
- 通信 47
- 同步消息传递 39

## [ W ]

- 网关, MQSeries Everyplace 15
- 网络, MQSeries 51
- 网络, MQSeries Everyplace 39, 51
- 网桥对象 52, 53
- 网桥, MQSeries 51

## [ X ]

- 先决条件 3
- 消息传递, 确保 59
- 消息传递, 同步 39
- 消息传递, 异步 39
- 消息传递, MQSeries 7
- 消息对象 16
- 消息级安全性 42
- 消息转换 57
- 消息, 管理 35
- 需求, 用户 14

## [ Y ]

- 压缩 40
- 异步消息传递 39

- 应用程序, 装入 50
- 应用程序, MQSeries Everyplace 13
- 用户要求 14
- 与 MQSeries 的兼容性 58
- 远程队列 21

## [ Z ]

- 侦听器对象 55, 56
- 证书的复制 45
- 证书复制 45
- 支持的操作系统 3
- 主机消息传递 vi, 7
- 注册表 44
- 注册表, 公用 45
- 注册表, 专用 44
- 注册表, MQSeries Everyplace 15
- 专用注册表 44
- 转储数据的格式 20
- 转储数据格式 20
- 转换程序 57
- 装入应用程序 50
- 自动注册 45

## M

- MQeAttribute 43
- MQeMTrustAttribute 43
- MQSeries客户机 8
- MQSeries 服务器 8
- MQSeries 网络 51
- MQSeries 网桥 9, 51
- MQSeries 网桥队列 23
- MQSeries 网桥规则 47
- MQSeries 系列 7
- MQSeries 消息传递 7
- MQSeries Everyplace 安全性 40
- MQSeries Everyplace 的管理 34
- MQSeries Everyplace 队列 21
- MQSeries Everyplace 队列管理器 26
- MQSeries Everyplace 对象 16
- MQSeries Everyplace 规则 46
- MQSeries Everyplace 类 49
- MQSeries Everyplace 设备 15
- MQSeries Everyplace 适配器 38
- MQSeries Everyplace 网关 15
- MQSeries Everyplace 网络 39, 51

- MQSeries Everyplace 应用程序 13
- MQSeries Everyplace 注册表 15, 44
- MQSeries Integrator vi, 7
- MQSeries Workflow vi, 7
- MQSeries, 兼容性 58
- MQSeries, 接口 51

## S

- SPI 61

---

## 将您的意见发送到 IBM

如果您对本书的任何内容有特别推崇或不能苟同的地方，请使用下面列出的一种方法将您的意见发送到 IBM®。

您认为本书中有特别的错误或遗漏，以及在准确度、组织、主题内容或完整性方面有任何意见，请尽量提出。

请仅就本书的信息和信息表示的方法提出您的意见。

**要对 IBM 产品或系统的功能提出意见，请向您的 IBM 代表或 IBM 特许的转销商咨询。**

当您把意见发送到 IBM 后，即授予 IBM 非专有权，IBM 对于您所提供的任何意见，有权利以任何它认为适当的方式使用或分发，而不必对您负任何责任。

您可以用下列任何方法把意见发送到 IBM：

- 邮寄地址是：

User Technologies Department (MP095)  
IBM United Kingdom Laboratories  
Hursley Park  
WINCHESTER,  
Hampshire  
SO21 2JN United Kingdom

- 通过传真：

- 在英国以外，在国际接入代码后使用 44-1962-842327
- 在英国国内，使用 01962-842327

- 通过电子方式，请使用适当的网络标识：

- IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
- IBMLink™: HURSLEY(IDRCF)
- 因特网: idrcf@hursley.ibm.com

不管您使用何种方式，请确保包含：

- 出版物标题和订单号码
- 您的意见所针对的主题
- 您的姓名和地址 / 电话号码 / 传真号码 / 网络标识。









中国印刷

GB84-0269-02

