



WebSphere MQ Everyplace

版本 2.0.2

目录

欢迎使用 MQe	1	使用的机器	22
MQe 概述	1	使用的代码库	22
代码库	3	MQe 开发周期	22
2.0.2 新增功能	4	积累 MQe 经验	23
使用本信息中心	4	结合 MQ 使用 MQe	24
什么是 MQe?	6	进一步信息	26
MQe 介绍	6	声明和商标	27
MQ 系列中的 MQe	6	声明	27
MQe 的用途	7	商标	28
MQe 的工作原理	8	词汇表	28
MQe SupportPac	20		
实施规划	21	索引	33
许可证	21		

欢迎使用 MQe

本产品的全名是 WebSphere® MQ Everyplace® V2.0.1, 更常用的名称为 WebSphere MQ Everyplace V2.0.1。

在本文档中, 本产品一般简称为 *MQe*。

MQe 用于在传感器、电话、个人数字助理 (PDA)、膝上型计算机和台式计算机等轻型设备上实现安全消息传递。

编程接口

MQe 的应用程序编程接口在本文档中称为 *MQe API*。MQe API 对 Java™ 和 C 编程语言的支持如下:

Java 提供对 V2 所有 MQe 功能的访问。

有三种版本的 C 支持:

本机 C 代码库

提供对 MQe 功能子集的访问, 主要限制是只能使用设备队列管理器, 因此它只能发送消息, 不能接收消息。

C 绑定

您可以使用 C 绑定, 直至本机 C 代码库提供完整功能。应用程序调用绑定中的 C API, 调用传递到 Java 类并由 MQe 实施功能。该绑定是针对 MQe V1 编写的, 但仍提供对 MQe V2 中几乎所有 Java 功能的访问。

要获取更多关于代码库的详细信息, 请参阅第 3 页的『代码库』。

另见: 第 28 页的『商标』。

MQe 概述

MQe 有什么用途?

- 传感器、电话、个人数字助理 (PDA)、膝上型计算机和台式计算机等轻型设备上的安全消息传递

什么是消息传递?

- 消息传递是一种软件 (如 MQ 和 MQe 中包含的软件), 它们在应用程序之间、通过网络发送和接收数据。消息传递十分可靠并与应用程序保持独立, 应用程序员不必具备具体的通信编程知识。
- 当某个应用程序要将数据传送到另一个应用程序, 只需将数据放入消息中, 然后将消息放入队列即可。
- 队列归队列管理器所有并由它运行。
- 可以将另一个应用程序 (或同一应用程序的另一部分) 配置为执行以下某项任务:
 - 该应用程序可以从同一队列检索那些消息。
 - 可以配置队列管理器, 使它通过网络上的某个连接将队列上的消息发送到另一台计算机上的远程队列管理器上的队列, 另一个应用程序再从该队列检索这些消息。
 - 目标应用程序可以根据需要跨网络拉出消息。
- 一个队列管理器上可以存在多个队列。
- 在 MQe 上, 每个 JVM 或进程只能有一个队列管理器。

什么是 MQe?

- MQe 是一个各种平台都支持的工具箱:
- API 提供 Java 或 C 版本。
- 产品功能以 Java 类和 C .DLL 形式提供。

MQe 的工作原理是什么?

- 以一组信息的形式创建队列管理器，这组信息描述了队列管理器的原始基本配置。
- 该信息保存在 MQe 注册表中（注：在 Windows® 系统上，这不是 Windows 注册表）。
- 在设备上，应用程序现在即可启动队列管理器，它将伴随应用程序一直运行下去。当应用程序停止时，队列管理器也随之停止。
- 当队列管理器处于运行状态时，可以通过向它发送 MQe 管理消息（这些消息同时更新注册表中的信息），对它进行各种配置和重新配置。通常可以使用管理工具 MQe_Script 和 MQe_Explorer 生成这些消息。
- 您可以在服务器上编写一个运行队列管理器的应用程序。该应用程序随后可以不间断运行，使客户机队列管理器能根据需要发送消息。在 Windows 系统上，可以将队列管理器应用程序作为服务运行，这样每次启动计算机时即可运行它。

如何使用 MQe?

- 您必须自行编写应用程序以使用 MQe。
- 在 PC 上创建 MQe 设备队列管理器配置并编写 MQe 应用程序。
- 只要将（应用程序和 MQe）所需的文件和组件下载到设备上，并在设备上运行。
- 在设备以外的计算机上运行 MQe 服务器队列管理器，使设备能与之连接并向它发送消息。
- 如果要与 MQ 通信，请在（设备以外的计算机上）运行 MQe 网关队列管理器，它将作为媒介 - 从设备（或服务器）接收 MQe 消息，将它们转换为 MQ 兼容的消息，然后与 MQ 交换消息。
- 可以创建队列管理器，并使用可下载的 SupportPac: MQe_Script（可编写脚本的命令）或 MQe_Explorer（GUI）远程管理 MQe 网络。
- 您可以尝试将这些工具用于 MQe，而不必编写应用程序。

有什么特殊功能?

- 可以利用适配器将 MQe 映射到设备接口。例如：
 - （连接每一端使用的）通道利用协议适配器在 HTTP、本机 TCP/IP、UDP 和其他协议上运行。
 - 队列利用字段存储适配器与内存或文件系统等存储子系统相接。
- 可以在三个级别实现安全性:

本地安全性

在本地级别保护与消息相关的数据。

消息级安全性

保护启动和接收 MQe 应用程序之间的消息。

基于队列的安全性

保护启动队列管理器和目标队列之间的消息。

- 可以使用规则定制某些主 MQe 组件的行为。
- 可以使用跟踪和事件记录帮助您调试应用程序。

另见:

『代码库』

第 20 页的『MQe SupportPac』

代码库

概述

MQe 应用程序编程接口 (API) 是 MQe 的编程接口。MQe API 对 Java 和 C 编程语言的支持如下:

Java 版本

提供对 V2 所有 MQe 功能的访问。Java Programming Reference 中描述了详细的类、方法和过程。本信息中心提供多个 MQe 编程示例。

有三种版本的 C 支持:

本机 C 代码库

提供对 MQe 功能子集的访问, 主要限制是只能使用设备队列管理器。其他限制如下 (另见下表):

- 不支持存储转发队列或网桥队列
- 只支持 HTTP 适配器
- 只支持 RLE 压缩程序
- 只支持 RC4 加密程序
- 只支持 *MAttribute* 和本地安全功能

C Programming Reference 中描述了详细的方法和过程。本信息中心为 C API 提供多个 MQe 编程示例。

C 绑定

您可以使用 C 绑定, 直至本机 C 代码库提供完整功能。应用程序调用绑定中的 C API, 调用传递到 Java 类并由 MQe 实施功能。该绑定是针对 MQe V1 编写的, 但仍提供对 MQe V2 中几乎所有 Java 功能的访问。C Programming Reference 中描述了详细的方法和过程。C Bindings Programming Reference 中提供 C 绑定的 MQe 编程示例。

队列管理器的类型

本文档及下表中使用以下队列管理器描述, 必须将它们区分对待:

设备队列管理器

没有侦听器组件和网桥组件的队列管理器。因此, 它只能发送消息, 不能接收消息。

服务器队列管理器

可以添加侦听器的队列管理器。有了侦听器, 服务器队列管理器可以接收和发送消息。

网关队列管理器

可以添加侦听器和网桥的队列管理器。有了侦听器, 网关队列管理器可以接收和发送消息, 而有了网桥, 它可以与 MQ 进行通信。

选项表

选项	Java	本机 C	C 绑定
操作系统	带有 Java 2 的任何操作系统（从 Java V1.2 开始）	PocketPC2002、PocketPC2003 和 Windows（从 MQe V2.0.1.5 开始）；	Windows 32 位
队列管理器	任何	仅设备	任何
MQ 的网关（带有网桥和侦听器的队列管理器）	是	否	是
存储转发队列，网桥队列	是	否	是
适配器	全部	仅 HTTP	全部
压缩程序	全部	仅 RLE	全部
加密程序	全部	RC4	全部
安全功能	全部	<i>MAttribute</i> 并且仅限本地	全部
将消息添加到跟踪	是	否	是
事件记录	是	否	是
专用注册表和凭证	是	否	是
属性规则	是	否	是
网桥规则	是	否	是
定制类	全部	否	部分
应用程序装入	是	否	是

2.0.2 新增功能

WebSphere MQ Everyplace V2.0.2 发行版中的更改

- 已从工具箱安装中除去所有文档并将它们放到 Web 上。
- 已将带有超大字符集支持的 UTF-16 添加到 C 本机代码库中。
- C 本机代码库中添加了 WinCE .NET 4.2 支持。请注意，现使用 .NET 环境提供的平台调用服务（Platform Invocation Services）提供对 WinCE .NET 4.2 上 WebSphere MQ Everyplace C DLL 的支持。
- 已添加迁移部分。本部分包含有关如何使用随 MQe 一起提供的新类的信息，这些新类使用户能通过网关队列管理器向 WebSphere Message Broker 发送、发布和预订消息。

使用本信息中心

辅助功能

本页的以下部分提供了将 Eclipse 用于 MQe 文档的特定技巧。

要获取更多关于 Eclipse 中辅助功能的一般信息，请参阅 Eclipse 帮助系统的辅助功能的相关帮助。

键盘技巧

- 使用功能键 F6 在窗格之间移动，如果使用基于 Mozilla 1.2 或更新版本的浏览器，则使用 Ctrl-F6。
- 在任何窗格或工具栏中，使用 Tab 键在图标或对象之间移动。

- 要返回先前查看过的主题，请使用退格键。要前进和后退，请分别使用“Alt-向右箭头”和“Alt-向左箭头”。
- 使用链接或搜索查找主题后，通过单击主题工具栏中的“以目录显示”图标在目录中找到相应的主题（使用键盘上的功能键 F6 在窗格之间移动，如果使用基于 Mozilla 1.2 或更新版本的浏览器，则使用 Ctrl-F6），然后在工具栏中使用 Tab 键选择相应的图标。
- 要打开上下文菜单，请使用 Shift-F10 或特殊的上下文菜单键（如果键盘上存在此键）。

窗口窗格

- 导航窗格在左侧。它为以下内容提供选项卡：
 - 内容
 - 搜索结果
- 主题窗格是您现在阅读的文字所在的窗格，它显示了当前的帮助主题。单击突出显示的链接即可直接转至其他主题（使用键盘上的 Tab 键在链接之间移动焦点，按 Enter 键打开焦点所在链接）。单击主题并按 **Ctrl-F** 在页面中查找单词（使用键盘上的功能键 F6 在窗格之间移动，如果使用基于 Mozilla 1.2 或更新版本的浏览器，则使用 Ctrl-F6）。使用键盘上的向上箭头和向下箭头键滚动页面。
- 该“主题窗格”顶部的工具栏提供以下操作的图标：后退、前进、以目录显示（适用于搜索后）和打印（分别使用键盘上的“Alt-向左箭头”、“Alt-向右箭头”、工具栏中的图标和 Ctrl-P；使用功能键 F6 在窗格之间移动，如果使用基于 Mozilla 1.2 或更新版本的浏览器，则使用 Ctrl-F6，然后在工具栏中使用 Tab 键选择相应的图标）。
- “导航窗格”上方的工具栏提供了搜索工具。

实用功能

- 使用“搜索”工具栏查找主题后，通过单击主题工具栏中的以目录显示图标在目录中找到相应的主题（使用键盘上的功能键 F6 在窗格之间移动，如果使用基于 Mozilla 1.2 或更新版本的浏览器，则使用 Ctrl-F6），然后在工具栏中使用 Tab 键选择相应的图标。
- 单击“主题窗格”（使用键盘上的功能键 F6 在窗格之间移动，如果使用基于 Mozilla 1.2 或更新版本的浏览器，则使用 Ctrl-F6），然后按 **Ctrl-F** 在页面中查找单词。
- 右键单击目录中的某项并选择属性打开一个对话框，选择并复制可供日后参考的主题的 URL（例如，可以将它添加到浏览器书签列表中或复制到电子邮件中）。

打印

- 单击工具栏中的打印图标，或单击相应的主题并按 Ctrl-P（使用键盘上的功能键 F6 在窗格之间移动，如果使用基于 Mozilla 1.2 或更新版本的浏览器，则使用 Ctrl-F6）。

字体大小

- 使用浏览器自身的功能控制字体大小。

搜索技巧

使用导航窗格上方的搜索工具栏在整个库中查找主题。请注意以下问题：

- 使用“高级搜索”限制在库中进行搜索的范围。
- 搜索数字时，对于某些版本的 Eclipse 帮助浏览器，必须用双引号括起数字（例如：“**2.0.1**”），否则无法进行搜索。
- 对于不完整的单词使用 *，例如：**admin*** 将找到 *administration*、*administer* 等。
- 搜索多个单词（以空格分隔），例如：**queue manager** 将找到带有单词 *queue* 和 *manager* 的主题。
- 搜索短语（用双引号括起），例如：“**queue manager**” 只会找到带有完整短语 *queue manager* 的主题。

- 当您查看主题时，将突出显示找到的单词。
- 要在显示的主题中查找单词，请单击主题窗格（使用键盘上的功能键 F6 在窗格之间移动，如果使用基于 Mozilla 1.2 或更新版本的浏览器，则使用 Ctrl-F6），然后按 **Ctrl-F**。

什么是 MQe?

MQe 介绍

MQe 是商业消息传递产品的 WebSphere MQ 系列的成员之一，也是 WebSphere Everyplace 系列的成员之一。您使用它来编写自己的应用程序，用于交换包含数据的消息，提供一次性可靠传递。MQe 的设计确保了它能与 WebSphere MQ 系列的其他成员紧密集成。

MQe 的设计旨在满足传感器、电话、个人数字助理 (PDA)、膝上型计算机和台式机等轻型设备的消息传递需求。它支持移动式环境并且适合在公用网络中使用，支持从弱通信网络的使用中引起的要求。由于许多 MQe 应用程序运行于因特网防火墙的保护范围之外，所以它还提供安全能力。

理解安全消息传递的概念对于了解本产品和文档将有所帮助。

如果您不了解这些概念，则阅读 WebSphere MQ 书籍 *An Introduction to Messaging and Queuing*, GC33-0805 可能会有所帮助。联机 WebSphere MQ 库的“书籍”部分以软拷贝形式提供该书。您可以从以下 WebSphere MQ Web 页面获得它：<http://www.ibm.com/software/integration/websphere/library/>。

MQ 系列中的 MQe

基本消息传递

消息传递基于队列和队列管理器，与特定产品或产品组无关。队列管理器管理可存储消息的队列。应用程序与本地队列管理器通信，并获取消息或将消息放入队列。如果将消息放入远程队列（队列归另一个队列管理器所有），则消息通过连接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跳跃通过一个或多个中间队列管理器。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。

MQ 和 MQe 提供 MQ 系列消息传递。两者都设计为支持一个或多个硬件服务器平台和多数关联的操作系统。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，以反映公共的功能和设计：

分布式消息传递

WebSphere MQ for Windows NT[®]、Windows 2000、AIX[®]、iSeries[™]、HP-UX、Solaris 及其他平台

主机消息传递

WebSphere MQ for z/OS[®]

普及消息传递

MQe for Windows、AIX、Solaris、Linux[®] 和 HP-UX

要获取更多详细信息，请参阅第 8 页的『MQe 的工作原理』。

MQe

MQe 支持各种网络配置。这里不存在 MQ 主机或分布式产品中的客户机或服务器概念。反之，您可以将 MQe 队列管理器配置为客户机或服务器，使它们能执行应用程序定义的任务。

定制配置的示例是您可以向 MQe 提供与 MQ 主机队列管理器交换消息的能力。要执行此操作，请配置一个具有网桥能力的 MQe 队列管理器。如果没有网桥，MQe 队列管理器只能与其他 MQe 队列管理器直接通信。但是，通过具有网桥能力的网络中的其他队列管理器，它可以间接通信。

要获取更多详细信息，请参阅第 8 页的『MQe 的工作原理』。

MQe 如何扩展 MQ 系列

MQe 通过以下方式扩展 MQ 系列的消息传递作用域：

- 支持 PDA、电话和传感器等低端设备。MQe 还支持膝上型计算机、工作站、分布式和主机平台等中间设备。MQe 提供一次性可靠消息传递，并允许与其他系列成员交换消息。
- 提供轻量级消息传递工具。
- 不论是在存储器中还是在传输中，提供大量的安全功能来保护消息、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。MQe 有一个有效的有线通信协议，并且可以在通信链路发生故障时自动恢复。
- 支持移动用户，当设备漫游时允许更改网络连接点。MQe 还允许在电池资源和网络受到约束的情况下控制行为。
- 通过适当配置的防火墙进行操作。
- 最小化用户的管理任务。这使 MQe 成为构建具有实用程序风格的应用程序的合适基础。
- 通过使用应用程序提供的规则，可轻松实现定制和扩展。

MQe 不支持所有 MQ 功能。除了要考虑环境、操作系统和通信因素之外，以下是一些更重要的区别：

- 无集群支持
- 无分发列表支持
- 无分组或分段的消息
- 无负载平衡 / 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
- 不同的可伸缩性和性能特征

但在 MQe 中，许多应用程序任务都可以通过使用 MQe 功能的替代方法，或通过利用划分子类、替换提供的类或利用规则、接口以及构建到产品中的其他定制功能来实现。

MQe 的用途

MQe 支持移动、脆弱的通信网络。由于 MQe 是针对轻型设备的，所以它只使用少量系统资源。它提供多种定制功能和接口，并不准备提供与 MQ 系列的其他成员相同的能力。它还包含一些独特的功能以支持其特殊的用户类，例如全面的安全性供应、消息、同步和异步消息传递、远程队列访问以及消息推和拉。

方案 and 应用程序

有各种类型的 MQe 应用程序，其中许多应用程序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输送

用户应用程序

- 使用 PDA 在家进行超市购物
- 收集航班上旅客的首选项
- 通过移动电话进行金融交易

控制应用程序

- 收集并集成经由人造卫星传输的来自输油管传感器的数据
- 在保证操作员有效的安全性情况下，对装置（例如：阀）进行远程操作

移动劳动力

- 拜访专业人员，例如保险代理人
- 快速发布快递公司的客户收据的证明
- 厨房与服务员之间的信息交换
- 高尔夫联赛得分记录
- 警察机关的安全移动系统消息传递
- 频繁丢失通信的情况下，打杂人员的工作信息
- 家庭仪表阅读

个人生产力

- 邮件和日历复制
- 数据库复制
- 下载到膝上型计算机

MQe 的工作原理

MQe 编程模型的基本元素是消息、队列和队列管理器。

- MQe 消息包含应用程序定义的内容。消息存储在队列中并且可以通过 MQe 网络移动。您可以通过指定目标队列管理器和队列名，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上，并且通常通过获取操作来检索它们。
- 队列可以是本地或远程的，并且由队列管理器管理。
- 注册表存储配置数据。

请阅读本部分的其余主题以了解更多内容。

消息

消息是一个应用程序要发送给另一个应用程序的数据集合。MQe 消息与 MQ 消息传递支持的那些消息有所不同：

- 在 MQ 中，消息是字节数组，分为消息头和消息体。MQ 创建消息头，其中包含应答队列的标识、应答队列管理器、消息标识以及相关标识等重要信息。消息体包含仅对应用程序有用的数据。
- MQe 中的消息没有消息头或消息体的概念。它们属于 *MQeFields* 类型，由名称、数据类型和数据本身组成。消息名称是无限长度的 ASCII 字符串，但不包含以下任何字符：

```
{ } [ ] # ( ) : ; , ' " =
```

第 9 页的表 1 描述了不同的数据类型：

表 1. 数据类型

类型	描述
ASCII	字符串或不变量 ASCII 字符串的动态数组，其中不包含以下任何字符 { } [] # () : ; , ' " =
Boolean	真或假值
Byte	字节值的固定数组或动态数组
Double floating point	双精度浮点值的值、固定数组或动态数组
Fields	对象或字段对象的动态数组（从而支持字段对象的嵌套）
Floating point	浮点值的值、固定数组或动态数组
Integer	整型的 4 字节值、固定数组或动态数组
Long integer	长整型的 8 字节值、固定数组或动态数组
Short integer	短整型的 2 字节值、固定数组或动态数组
Unicode	字符串或 Unicode 字符串的动态数组

另外，消息包含由 MQe 生成的 UID（唯一标识）。这个 UID 唯一地标识整个 MQe 网络中各个单独的消息对象，它是根据以下内容构造而成的：

发起队列管理器

这是发起队列管理器的名称，它必须是唯一的。它在接收到消息时由队列管理器添加。由于它是 ASCII，所以每个字符为一字节长。

创建时间

这是消息的时间戳记。因此，在 Java 中，这是本地系统上创建消息的时间，而在 C 中，这是本地系统上将字段项添加到队列的时间。然后字段项成为消息。

虽然几乎肯定存在其他属性，但是发往另一个 MQe 队列管理器的消息不需要任何其他信息。其他属性可以：

- 反映当前状态
- 与特定消息子类关联
- 允许您定制消息。

注：在 C 代码库中，字段项仅在到达队列时才变为消息。

MQe 将属性相关信息添加到消息（随后除去它），从而实现消息传递和排队操作。在队列管理器之间发送消息时，您可以添加重新发送信息以表明正在重新传输该数据。

消息也可以有属性。属性是 MQe 安全模型或不可缺的，它允许对内容进行选择性访问并保护内容。它们有下列属性：

表 2. 属性对象属性

属性	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（用于传输和存储）
规则（不适用于 C 代码库）	控制允许的操作

要获取更多关于表 2 中属性的信息，请参阅第 16 页的『安全性』。

队列用于存放消息。应用程序不会直接访问队列，但通过队列管理器使用队列。

队列由名称标识，并且名称可以是无限长度的 ASCII 字符串，不包含以下任何字符：

```
{ } [ ] # ( ) : ; , ' " =
```

但是，队列名必须在特定队列管理器中唯一。为了实现与 MQ 的互操作性，我们建议您同时遵循 MQ 命名限制（包括 48 个字符的最大名称长度限制）。名称长度可能还受您所使用文件系统的限制。MQe 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（不包含硬件故障或设备丢失）。本地队列属于特定队列管理器。这可以是独立队列管理器或连接到网络的队列管理器。

远程队列

远程队列是对驻留在 MQe 网络中另一个队列管理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实际队列自己的队列管理器。远程队列还有一些有关访问权、安全特征和传输选项的属性。它们的访问方式可以是同步或异步的。同步远程队列没有存储器，消息会立即传递到远程队列管理器。异步队列将消息存储在本地存储器中，然后使用后台线程将消息发送到远程队列管理器。

存储转发队列

存储转发队列代表一个或多个队列管理器存储消息，直到它们准备好接收消息。C 代码中没有此类队列。存储转发队列有两种主要用法：

1. 在网络中启用消息的中间存储器，这样它们就可以继续到它们的目的地（转发角色）。
2. 存放等待由宿主服务器队列收集的消息。

通过在服务器或网关上定义此类队列（但这不是必需的）。存储转发队列可以存放许多目标队列管理器的消息，或者有一个用于每个目标队列管理器的存储转发队列。

宿主服务器队列

当远程队列以及存储转发队列将消息在网络上推送，并且发送队列启动传输时，宿主服务器队列将消息从远程队列中拉出。消息永不寻址到宿主服务器队列。

宿主服务器队列定义标识远程队列管理器上的存储转发队列。然后宿主服务器队列从存储转发队列中拉出去往其本地队列管理器的消息。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转发队列关联。

管理队列

管理队列是一种接受管理消息的本地队列。管理消息包含由应用程序内部处理的指令，这些指令与 MQe 的特定元素相关。每个管理操作可以选择将管理应答消息发送回发起应用程序。这些应答消息通知您管理操作成功还是失败。这样，使用管理队列允许一个队列管理器上的元素能同步或异步控制第二个队列管理器的配置。按管理消息到达管理队列的顺序处理它们。要获取进一步的信息，请参阅关于第 12 页的『管理』的部分。

MQ 网桥队列

网桥队列是一种特殊形式的远程队列，它描述了 MQ 远程队列管理器上的队列。网桥队列放入它们引用的 MQ 队列或从中获取。仅在 Java 中，它使用转换程序在 MQe 和 MQ 系统之间交换每条消息时，执行任何必需的数据或消息重新格式化。您只可在网关队列管理器上创建网桥队列。

MQe 将数据安全地存储在队列上，确保将消息实际写入介质而不只是由操作系统存储。但是，MQe 不会独立记录对消息和队列的更改。因此，要从介质故障中恢复过来，您需要部署硬件解决方案，例如，RAID 磁盘系统。或者，将队列映射到可恢复的存储器，例如，数据库子系统。MQe 客户机运行于轻型系统上，但服务器队列管理器往往需要不间断运行，此时需要实现故障转移。

MQe 有四种常用系统队列:

管理队列

接收管理消息

死信队列

存储不能以其他方式传送的消息

管理应答队列

接收对管理消息的应答

SYSTEM.DEFAULT.LOCAL.QUEUE

与 MQ 服务器上的强制性系统队列共享一个公共名称。

队列管理器

MQe 队列管理器允许 MQe 支持各种网络配置。它提供:

- 对 MQe 应用程序的消息传递及排队网络的中央访问点
- 可选的客户端排队
- 连接控制
- 可选的管理功能
- 一次性可靠消息传递
- 从故障条件中自动恢复
- 可定制基于规则的行为

在 MQe 中, 一个 Java 虚拟机 (JVM) 或一个本机应用程序进程上在任意时刻只能有一个队列管理器处于活动状态。要使一台机器上有多个队列管理器, 您需要多个 JVM 或多个本机应用程序进程。

队列管理器由全局唯一的名称和不限长度的 ASCII 字符串标识, 不包含以下任何字符:

```
{ } [ ] # ( ) : ; , ' " =
```

MQe 或 MQ 没有这一限制, 但重复的队列管理器名称可能导致消息传递到错误的队列管理器。为了实现互操作性, 我们建议您将最大名称长度限制为 48 个字符。您所使用的文件系统也可能限制名称长度。

您可以使用或不使用本地排队来配置队列管理器。所有队列管理器支持同步消息传递操作。使用本地排队的队列管理器还支持异步消息传递。异步消息传递和同步消息传递具有完全不同的特征和结果:

同步消息传递

使用同步消息传递, 应用程序将消息放入 MQe 以传递到远程队列。MQe 同时联系目标队列并传递消息。传递后, MQe 立即返回到应用程序。如果无法传递消息, 发送应用程序会立即接收到通知。MQe 对于同步情况 (不可靠消息传递) 下的消息传递不承担任何责任。

异步消息传递

使用异步消息传递, 应用程序将消息放入 MQe 以传递到远程队列。MQe 立即返回到应用程序。如果可以立即传递消息或将它移到合适的登台区, 则发送它。否则, 将它存储在本地。异步传递提供了一次性可靠传递, 因为消息已传递到 MQe 并且它承担传递责任 (可靠消息传递)。

队列管理器配置: 在装入队列管理器之前, MQe 队列管理器需要一个应用程序以创建所需的环境。这意味着启动队列管理器的应用程序必须能访问特定信息, 随后才能装入队列管理器。例如, 如果为 Java 中的通信适配器设置数据包大小, 则必须在装入队列管理器之前, 设置所需的 Java 属性。应用程序所需的两部分信息分别是 MQe 注册表的位置以及队列存储的位置。注册表 (请勿与 Windows 注册表混淆) 中存放属于队列管理器的所有对象的定义, 如队列和连接定义。它使队列管理器在装入时能创建正确的对象。队列存储中存放队列并允许队列管理器存放本地队列上的消息, 这些消息在停止和启动队列管理器之间保持持久。

您可以使用 API、与 MQe 一起提供的实用程序或 MQe_Explorer 等管理工具来配置 MQe 环境。这些方法可捕捉初始化文件中的环境参数，但这是可选的。

可以使用 MQ 网桥能力来配置队列管理器。这称为网关，它在 Java 中与 MQ 主机和分布式产品交换消息。C 代码库仅使用设备队列管理器。

队列管理器操作： 队列管理器支持消息传递操作和管理队列。应用程序通过队列管理器的服务访问消息，使用的方法如下：

Get 该操作从队列中除去消息。

Put 该操作将消息放在队列中。

Delete 通过指定 UID，您可以从队列中删除消息，而不使用 get 操作。

Browse

您可以使用过滤器浏览消息的队列（请参阅以下内容）。浏览将检索与过滤器匹配的所有消息，但仍将它们保留在队列中。MQe 还支持在锁定情况下浏览。这允许您锁定匹配的消息。

Wait 在 Java 中，应用程序可以在指定的时间内等待消息到达队列。这不适用于 C 代码库。

Listen 在 Java 中，应用程序可以使用可选的过滤器侦听 MQe 消息事件。但是，为了执行该操作，您必须将侦听器添加到队列。当消息到达队列时，将通知侦听器。这不适用于 C 代码库。

许多此类操作都将 **filter** 作为它们的参数之一。过滤器检查元素是否相等并且消息的任何部分都可用于选择性检索。多数方法调用还包含消息编码和解码中使用的属性。

管理

MQe 接口负责管理消息的生成和接收，从而实现管理。应用程序负责与消息相关的功能，而管理则提供配置和管理 MQe 资源（如队列和连接）的工具。

请求发送到目标队列管理器的管理队列，并且可以根据需要接收应答。任何本地或远程 MQe 应用程序可以通过助手方法直接或间接创建和处理管理消息。

C 代码库提供一个管理器，它允许您执行部分管理操作。但是，只能对本地队列管理器管理的资源执行这些操作。

管理队列本身不能执行个别资源的管理。相关信息包含在每个资源及其相应的管理消息中。

管理消息： 创建队列管理器后，通过将管理消息发送到目标队列管理器管理队列来配置队列管理器。无法管理没有管理队列的队列管理器。使用管理消息潜在的意图是本地和远程管理都以同样的方式执行。

管理消息创建并发送到要管理的队列管理器的管理队列。您可应用基于队列的安全性属性以控制访问。管理消息包含请求的详细信息，表明是否需要响应，并且包含标识目标队列管理器和队列的地址。因此，MQe 具有以下样式的管理消息：

- 表明不需要答复的管理操作的命令
- 需要答复的请求
- 从原始消息的副本中构造的答复消息

发送方可添加其他字段以由接收方使用。管理队列本身遵照消息执行。管理消息可查询、创建、删除或更新对象。对于对象的子集，它们可执行其他功能，例如，停止和启动。

还可以通过 MQe_Explorer 间接生成管理消息，这个管理工具为系统管理提供了一个图形用户界面。MQe_Explorer 未包含在 MQe 中，但可以作为 SupportPac™ 免费下载。

选择性管理: 管理队列上的认证程序可控制对于管理的访问。提供的认证程序认为本地应用程序表示相同的本地用户，因此，启用或阻止所有应用程序的管理。

在任何管理消息流控制远程管理应用程序之前，在连接上启动认证程序。这将不同的远程应用程序区分开，然后启用或阻止每个远程应用程序的管理。在所有情况下，完全启用管理或是完全阻止管理。

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。要获取关于认证的更多信息，请参阅第 16 页的『安全性』。您也可以使用与队列关联的规则以类似方式启用或阻止操作。要获取关于规则的更多信息，请参阅第 18 页的『定制规则』。

监视和相关操作: 管理不仅仅涉及创建和修改元素。它可以包含监视系统、在队列满的时候通知操作员，或者处理一些错误情况，例如，在对于目标队列来说过大的消息到达时采取适当操作。每当元素显著更改其状态或出现某些类型的错误情况时，MQe 使用规则来处理这些问题。MQe 提供缺省规则实施，用户可以根据需要定制它。要获取关于该操作的更多信息，请参阅第 18 页的『定制规则』。

连接

连接一起提供队列管理器与信息以建立与远程队列管理器的通信链接。然后队列管理器使用连接交换信息。连接定义本地存储在每个队列管理器上。

注：C 代码库只是一个设备队列管理器。

连接的一些主要功能是：

支持同步和异步消息传递

同步消息传递提供直接从源应用程序到目标队列的传输服务，而不在源队列管理器上排队。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。

终端对终端服务供应

连接从源队列管理器到目的地队列管理器，可能要运行通过中间队列管理器。所使用的潜在传送协议可在连接传递通过这些中间队列管理器时有所更改。多个连接可以链接起来以形成终端对终端连接。

支持压缩、加密和认证

连接具有这些安全特征以保护传输中的数据。

支持客户机 / 服务器操作

客户机 / 服务器连接是请求 / 响应。客户机向服务器发出请求，服务器响应该请求。注意，这并不限制消息流。消息可以从客户机流向服务器，反之亦可。

以下图显示一些典型的 MQe 配置。为清楚起见，这些图只显示了已定义的直接连接。您也可定义利用直接连接的间接连接。在这些图中，带箭头并指向服务器的线表示客户机 / 服务器连接。客户机可以使用客户机 / 服务器连接向服务器发送消息，也可以从该服务器拉出发给它们自己的消息。不带箭头的线表明在 MQe 与 MQ 之间启用通信的 MQ 客户机通道。



图 1. 独立 MQe 队列管理器

第 13 页的图 1 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。

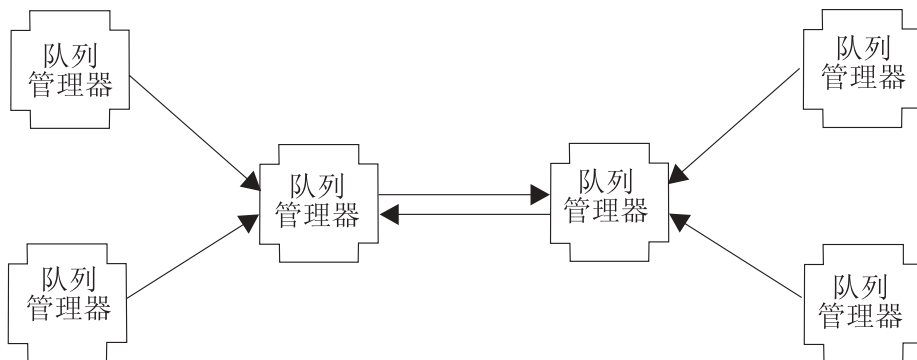


图 2. 小型网络配置

图 2 显示一个小型网络配置，其中的中央服务器队列管理器使用一对客户机 / 服务器直接连接来交换信息。每个客户机队列管理器使用一个客户机 / 服务器直接连接来链接到某个服务器队列管理器。

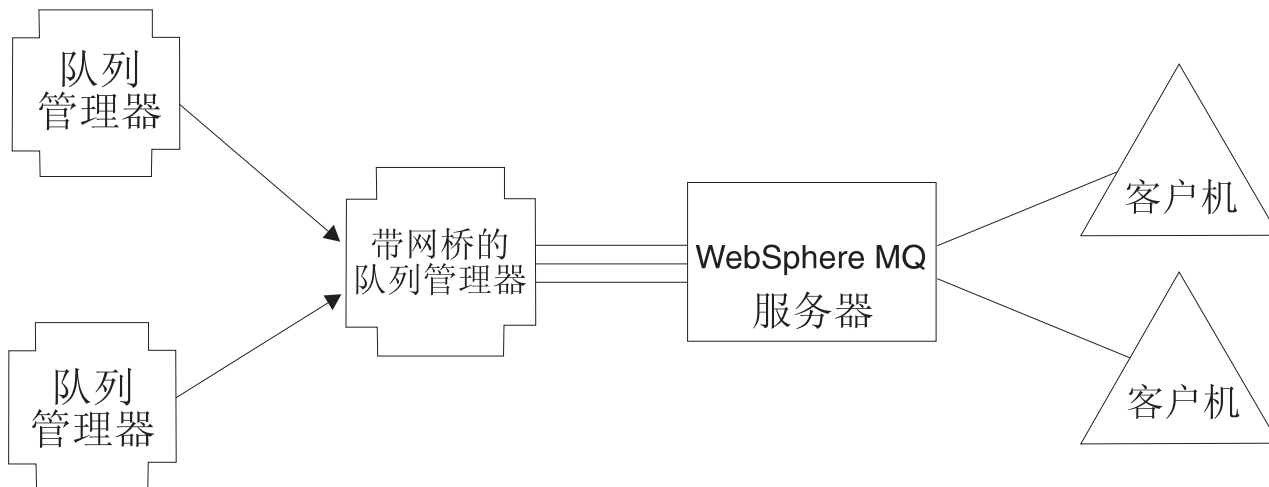


图 3. 集成型 MQ 系列网络

图 3（图 3）显示一个 MQe 配置，其中某个队列管理器已使用网桥选项进行配置，并且客户机通道池已指向单个目标 MQ 主机 / 分布式服务器。

由于连接通常定义对远程队列管理器的访问，所以它们有时也指远程队列管理器定义。

您也可指定间接连接。在这种情况下，MQe 通过其他队列管理器传递连接（可以链接起来），途中协议可能会更改。间接连接在使设备拥有 MQe 网络的单个进入点方面尤其有用。

对于多数 MQ 元素，您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。

连接支持双向流动，并根据需要由队列管理器建立。异步和同步消息传递使用相同的连接，而且使用的协议对于 MQe 是唯一的。

连接定义确定用于特定连接的链接和协议。在每个中间节点上，那时任何流过的消息传递到队列管理器。队列管理器将根据它具有的资源处理消息。因此消息可放在队列上，这种队列可以是本地队列、远程队列或存储转发队列。放在远程队列上的消息将根据远程队列的类型继续它们的旅程。同步远程队列将立即向前移动消息。异步远程队列将在移动它们之前存储它们的消息。

连接对应用程序或管理员而言不是直接可视的，它们根据需要由队列管理器建立。连接将队列管理器链接在一起，它们的特征则由 MQe 根据流过的信息进行更改。传送程序是利用连接提供队列级别通信的 MQe 组件。同样，这些传送程序对应用程序或管理员都是不可视的。

当需要可靠消息传递时，MQe 将消息一次性传递到应用程序。要实现这个功能，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，并在传输端删除其副本前已得到确认。万一出现通信故障，如果未接收到确认，则可以重新传输消息，因为一次性传递并不意味着一次性传输。但是，不传递重复消息。

连接样式： MQe 支持客户机 / 服务器操作。客户机可以启动与服务器的通信。服务器只能对客户机启动的请求做出响应。这些组件有：

侦听器 侦听进入的连接请求。

队列管理器

通过提供消息传递和排队能力支持应用程序。

表 3. 连接样式

	队列管理器	侦听器
客户机	是	否
客户机 / 服务器	是	是
服务器	是	是
Servlet	是	否

表 3 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 MQe 能以客户机或服务器方式运行的情况。Servlet 选项描述了将 MQe 配置为 HTTP Servlet 的情况，其中 HTTP 服务器本身负责侦听进入的连接请求。

MQe 应用程序不会直接发现队列管理器使用的连接样式。但是，样式也很重要，因为它影响着各方可用的资源、能与其他队列管理器连接的队列管理器、MQe 使用的内存量以及可以并存的连接。

适配器

适配器用于将 MQe 映射到设备接口。例如：

- 通道利用协议适配器在 HTTP、本机 TCP/IP、UDP 和其他协议上运行。
- 队列利用字段存储适配器与内存或文件系统等存储子系统相接。

适配器为 MQe 提供一种扩展其设备支持并允许版本控制的机制。

注：与 MQe Java 代码库不同，C 代码库仅使用 HTTP 适配器。

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。

当断开连接的设备上的 MQe 试图使用网络（例如：因为必须发送消息）而网络堆栈不活动时，操作系统将自己启动远程访问服务（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所使用的操作系统是可用的。MQe 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。

MQe 中嵌入调用以跟踪信息、警告和出错情况。

应用程序还可以直接调用跟踪，并且仅使用 MQe Java 代码库添加新消息。

由于发布了跟踪处理程序必须实施的接口，解决方案可以实施该接口来收集 MQe 和应用程序跟踪、使之交错并将输出直接发送到可以收集它的位置。一些跟踪处理程序作为产品代码的一部分提供。

并且，由于多数 MQe 异常将传递到应用程序进行处理，应用程序异常处理程序还可以将这些异常传递到跟踪。

事件日志

这不适用于 C 代码库。

MQe 提供可用于记录状态的事件日志机制和接口。例如，如果无法传递异步远程队列上的消息，将记录警告消息。

缺省情况下，记录写入 `system.out`，您也可以拦截记录内容并将它指向其他位置。

MQe 事件日志不记录消息数据，并且不能用于恢复消息或队列。

安全性

MQe 提供一组集成的安全功能，能够在本地存放和传送消息数据时保护它们。

MQe 提供三种不同类别的安全性：

本地安全性

在本地级别保护与消息相关的数据

消息级安全性

保护启动和接收 MQe 应用程序之间的消息

基于队列的安全性

保护启动的队列管理器 and 目标队列之间的消息

本地和消息级安全性由 MQe 内部使用，同时还可供 MQe 应用程序使用。基于 MQe 队列的安全性是一项内部服务。

所有这三类 MQe 安全功能通过使用 `MQeAttribute` 等属性来保护消息数据。根据类别，可以在外部或内部应用该属性。

每个属性都可以包含下列内容：

认证程序

提供其他控制以阻止未授权的用户对本地数据的访问

加密程序

控制必需的保护强度

压缩程序

优化受保护数据的大小

密钥 通过请求密码来控制访问

目标实体名称

请求目标队列名

根据 MQe 安全性类别，这些元素的使用方法有所不同，但在所有情况下，当调用附加到消息的属性时，将应用 MQe 安全功能进行保护。

注册表： 注册表是与队列管理器相关信息的主要存储。每个队列管理器至少具有一个注册表。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
- 连接定义
- 用户数据（包括与配置相关的安全性信息）

使用适配器（通常是 MQeDiskFields 适配器）存储注册表。

专用注册表和凭证： 本部分不适用于 C 代码库。

由于每个实体需要认证其自己的凭证，我们需要知道：

1. 如何执行注册以获取凭证
2. 在何处以安全的方法管理凭证

专用注册表实现了实体专用凭证的安全管理，并由公用注册表管理公用凭证集。

专用注册表提供一个带有安全或密码令牌的基本注册表。例如，它可以是迷你证书等公用元素、专用密钥等专用元素的安全存储库。

专用注册表只允许授权的用户访问专用元素。通常，只有合法的队列管理器用户可以使用 PIN 访问注册表。但是，如果您不是非常关心安全性问题，则配置选项允许您忽略它。

该专用注册表以专用对象从不保留专用注册表这样一种方法对一些（例如：数字签名、RSA 解密）服务提供支持。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。

自动注册： MQe 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实体的专用注册表中。因此，自动注册为建立消息级保护的凭证提供一个简单机制。

自动注册步骤包含：

1. 生成新的 RSA 密钥对
2. 保护和保存专用注册表中的专用密钥
3. 将新证书请求中的公用密钥封装到缺省迷你证书服务器中

假设迷你证书服务器已配置并且可用，它将实体的新迷你证书及其自己的证书一同返回。这些服务器和受保护的专用密钥一起作为新的凭证存储在实体的专用注册表中。

公用注册表与证书复制： MQe 提供的缺省服务使 MQe 组件能共享迷你证书。MQe 公用注册表为迷你证书提供可公开访问的存储库。这类似于移动电话上的个人电话簿服务，区别在于它不是电话号码而是联系最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没有的迷你证书，并且使用一个有效的宿主服务器组件进行配置，公用注册表将自动尝试从宿主服务器的公用注册表中访存所需的迷你证书。这些服务可用于提供智能化的迷你证书自动复制服务，可以在适当的时候提供相应的迷你证书。

注册表服务的应用程序使用： MQe 队列管理器充分利用使用专用及公用注册表服务的优势，但不限制对这些服务的访问。MQe 解决方案可以定义和管理它们自己的实体（如用户）。然后您可以使用专用注册表服务自动注册和管理新实体的凭证，并使用公用注册表服务使得公用凭证随处可用。

缺省迷你证书发行服务： SupportPac MQe Server Support 包含一个 WTLS 迷你证书服务器，可以单独免费下载。

该软件包为 WTLS 证书提供一个证书发行服务。您可以在该证书发行服务器上配置队列管理器和队列实体，以提供缺省迷你证书发行服务，该服务使专用注册表自动注册请求满足 WTLS 证书的发行。您可以使用 MQe 证书发行服务设置和管理迷你证书发行服务，将迷你证书颁发给一组严格控制的实体名称。这种发行服务的特征是：

- 管理一组已注册的可认证实体
- 迷你证书发行
- WAP WTLS 迷你证书存储库管理

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

- 添加和除去类别名
- 定义适配器
- 映射文件描述符
- 处理连接命令

定制规则

规则允许用户定制某些主 MQe 组件的行为。

MQe 根据需要提供缺省规则，但您可以将这些缺省规则替换为应用程序或安装特定的规则，以满足客户要求。

支持的规则类型是在触发方式以及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： 属性规则仅适用于 Java 代码库。

只要试图更改状态，就将控制这种规则类，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。

MQ 网桥规则： MQ 网桥规则仅适用于 Java 代码库。

当 MQe 到 MQ 的网桥代码状态更改时，这些规则类将接管控制。有一个单独的网桥规则类用于确定以下每种情况：

- 当侦听器无法将来自 MQ 的消息传递到 MQe（例如，因为消息太大或队列不存在）时，如何处理消息

- 一旦将服务器实例化，网桥管理的队列应在什么状态下中启动
- 当网桥发现 MQ 同步队列，即用于崩溃恢复的持久存储（缺省规则仅显示问题）出错时，如何处理
- 如何使用转换程序将 MQe 消息转换为 MQ 消息以及反之

队列规则: 在队列生命周期的关键点调用该规则类，例如以下情况:

- 将消息添加到队列中，以确定是否超出阈值（即超出消息数或消息大小）。
- 打开或关闭队列。
- 从队列管理器中除去队列。这不适用于本机 C 代码库。
- 队列上的消息已超出队列或它自己的到期间隔。

队列管理器规则: 在队列管理器生命周期的关键点调用该规则类，例如以下情况:

- 例如，打开队列管理器，启动后台计时器线程以允许定时操作发生。
- 例如，关闭队列管理器以终止后台计时器线程。
- 触发队列管理器的暂挂消息的传输。

类

本部分不适用于 MQe 本机 C 代码库。

MQe 为某些功能提供了一些类，这些功能允许您定制 MQe 行为以满足特定应用程序需求。在某些情况下，将这些类的接口记录成文档，以便于开发其他替代方法。下表概述了这些可能性。可以明确标识这些类，也可以使用别名标识它们。

注: C 绑定 API 中未提供部分类。要获取受支持类的明确列表，请参阅 *Java Programming Reference* 和 *C Programming Reference*。

MQe 为许多这些类自动提供别名。*Java Programming Reference* 中的 `com.ibm.mqe.MQe.alias` 记录了这些内容。

表 4. 类选项

类	已提供备用项	已记录接口	MQe 程序包	如何实施
管理	否	是		
认证程序	是	否	<code>com.ibm.mqe.attributes</code>	扩展 <code>com.ibm.mqe.MQeAuthenticator</code>
通信适配器	是	是	<code>com.ibm.mqe.adapters</code>	扩展 <code>com.ibm.mqe.adapters.MQeCommunicationsAdapter</code>
通信样式	是	否		
压缩程序	是	否	<code>com.ibm.mqe.attributes</code>	扩展 <code>com.ibm.mqe.MQeCompressor</code>
加密程序	是	否	<code>com.ibm.mqe.attributes</code>	扩展 <code>com.ibm.mqe.MQeCryptor</code>
事件日志	已提供样本	是		实施 <code>com.ibm.mqe.MQeEventLogInterface</code>
消息	否	是	<code>com.ibm.mqe</code>	扩展 <code>com.ibm.mqe.MQeMsgObject</code>
队列存储	是	否		通常应使用别名 <code>MsgLog</code> ；定义的缺省值。
规则	已提供缺省类	是		扩展 <code>com.ibm.mqe.MQeRule</code>
存储适配器	是	是	<code>com.ibm.mqe.adapters</code>	扩展 <code>com.ibm.mqe.adapters.MQeAdapter</code>
跟踪	已提供样本	是	<code>com.ibm.mqe.trace</code>	

应用程序装入:

注：本部分不适用于 C 代码库。

当装入 MQe 队列管理器时，启动应用程序必须将任何其他应用程序装入 JVM。

标准 Java 工具可用于该目的，也可以使用作为 MQe 的一部分提供的类装入器。

因此：

- 多个应用程序可以对相同 JVM 中的单个队列管理器运行。
- 您也可以使用多个 JVM，但每个 JVM 都必须有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。

MQe SupportPac

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

1. **MQe 许可产品**，可以通过 IBM® 提供的物理介质获得或从以下站点进行 Web 下载：

<http://www.ibm.com/software/integration/wmqe/>

许可产品包括：

- MQe Java 类
- 帮助程序类
- MQe C 绑定文件和本机 C 代码库
- 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”还包括在某些平台上将产品用于非开发用途的授权。要在大型机器上使用或与 MQ 网桥一起使用，还需要另行购买容量单位。

2. **MQe SupportPac**，可以从以下站点进行 Web 下载：

<http://www.ibm.com/software/integration/support/supportpacs/>

或

<http://www.ibm.com/software/integration/wmqe/>

MQe SupportPac 中的管理工具在应用程序开发和实施的所有阶段中都扮演着重要角色。它们比许可产品所包含的实用程序更为复杂，并且是入门、配置、检查实验网络和管理产品系统的一个基础辅助工具。

EA01: WebSphere MQ Everyplace - XML 转换实用程序

该软件可以将 MQeFields 对象转换为 XML 表示法，反只亦可。

EP02: WebSphere MQ Everyplace - DB2® 适配器用户指南

将消息持久存储扩展到 DB2 数据库中。

ES06: WebSphere MQ Everyplace 服务器支持

捆绑了 MQe_Explorer、MQe_Script、MQe_MinicertServer 和 MQe_Service，它们为 MQe 提供了管理和安全功能。

MS0B: MQSeries® Java classes for PCF

提供 PCF 消息支持的 Java 代码。有关使用方法，请参阅第 21 页的『MS0B - MQSeries Java classes for PCF』。

MS0B – MQSeries Java classes for PCF

PCF 消息是 MQ 队列管理器使用的管理消息。该 SupportPac 包含提供 PCF 消息支持的 Java 代码。

如果下载并安装它，然后将 `com.ibm.mq.pcf.jar` 文件放入类路径环境变量中，您即可访问 Java 类，这些类可以动态操作 MQ 资源。将 PCF 消息与 MQe 管理消息结合使用时，您可以对网桥资源以及 MQe 队列管理器上的相应资源进行全面的程序化配置。`examples.mqbridge.administration.programming.AdminHelperMQ` 类包含的示例代码与 `examples.mqbridge.administration.programming.MQAgent` 结合使用，演示了如何实现该操作。该示例代码已添加到 `examples.awt.AwtMQeServer` 程序，这样选择视图 → 连接本地 MQ 缺省队列管理器将：

- 确保存在网桥对象，并根据需要进行创建。
- 从缺省 MQ 队列管理器查询属性。
- 尝试将该队列管理器连接到当前正在运行的 MQe 队列管理器。
- 确保存在代表缺省 MQ 队列管理器的代理对象，并根据需要进行创建。
- 确保存在 MQe 客户机连接以及相应的 MQ 服务器连接通道，并根据需要创建这些资源。
- 确保 MQ 队列管理器上存在同步队列。
- 确保 MQ 上存在传送队列，并根据需要创建传送队列。
- 确保当前 MQe 队列管理器的配置中存在匹配的 MQ 传送队列侦听器，并根据需要进行创建。
- 确保已启动所有网桥资源。
- 确保 MQ 队列管理器上存在测试队列，并根据需要进行创建。
- 确保存在匹配的 MQe 网桥队列，它引用该测试队列。
- 向测试队列发送测试 MQeMQMsgObject 以确保配置可用。
- 从测试队列获取测试 MQeMQMsgObject 以确保配置可用。

实施规划

许可证

部署 MQe 应用程序所需的许可证

MQe 是一个允许用户编写 MQe 应用程序并创建此类应用程序的运行环境的工具箱。在部署本产品或使用它的应用程序之前，请确保您有必需的许可证。

1. 在**服务器**上使用“程序”的许可证定价基于**处理器许可证单位**。在服务器上使用“程序”的每个副本，需要为包含在运行“程序”副本的服务器中的每个处理器或对称多处理器获取一个**处理器许可证单位**。根据“程序”运行于销售点（即零售）设备还是另一种类型的计算机上，需要不同类型的**处理器许可证单位**和**设备使用授权**。在零售设备上使用“程序”需要**零售服务器许可证**，而在其他（非零售）设备上使用“程序”则需要**网络服务器许可证**。
2. 在单独的**客户机设备**上使用“程序”需要附加的**设备使用授权**，以下 3. 中描述的网络服务器许可证中包含的除外。
3. 每个**网络服务器许可证**包含不超过一百（100）台客户机设备受限使用“程序”的授权，前提是所有此类副本都用在与服务器副本相同的经济企业或组织中。

要获取这些限制的详细信息，请参阅 <http://www.ibm.com/software/integration/wmqe/>。

要在指定的客户机平台上使用产品（代码开发和测试以外的用途）需要**设备平台使用授权**（记录在权利证明文档中并且可用于支持 MQe 使用）。这些许可证未授予用户使用 MQe 网桥的权利，也不允许在 IBM 发布的 MQe 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以下 URL 从 Web 上获得。

要获取这些限制的详细信息，请参阅 <http://www.ibm.com/software/integration/mqfamily/>。

使用的机器

用于开发和部署应用程序的机器

您需要：

- 一台 PC 用于编写和编译应用程序。

建议使用 Windows 系统，因为随后即可运行任何 MQe SupportPac，特别是开发时很实用的 MQe Explorer。

这台计算机必须能访问因特网，以下载 MQe、SupportPac 和文档等。

- 至少一台要进行部署的计算机或设备（用于测试）。
- 将设备连接到开发 PC 的接口设备和电缆。

使用的代码库

MQe 应用程序编程接口（API）是 MQe 的编程接口。支持两种语言：Java 和 C。

Java 版本提供对所有 MQe 功能的访问。Java Programming Reference 中描述了详细的类、方法和过程。本信息中心提供多个 MQe 编程示例。

有三种版本的 C 支持：

本机 C 代码库提供对 MQe 功能主要子集的访问。因为 C 代码库只是一个设备队列管理器，所以它：

- 不支持存储转发队列或网桥队列
- 只支持 HTTP 适配器
- 只支持 RLE 压缩程序
- 只支持 RC4 加密程序
- 只支持 *MAttribute* 和本地安全功能

C Programming Reference 中描述了详细的方法和过程。本信息中心提供 C 绑定的 MQe 编程示例。

您可以使用 **C 绑定**，直至本机 C 代码库提供完整功能。它们提供对 MQe 功能主要子集的访问。C Programming Reference 中描述了详细的方法和过程。C Bindings Programming Reference 中提供 C 绑定的 MQe 编程示例。

MQe 开发周期

考虑到 MQe 的广泛用途，该产品与 MQ 系列中其他成员的安装、配置和部署方法有所不同。采用 MQe 分三个阶段：

1. 开发和创建原型阶段

根据 MQe 开发许可证条款规定，可免费安装和使用 MQe。MQe 应用程序是使用 JMS API 或 MQe Java 和 C API 提供的功能开发的。创建和使用 MQe 队列管理器时，请注意以下问题。这些问题可能影响到应用程序实施方式。

- 在一个 JVM 或一个 C 进程中，任意时刻只能使用一个 MQe 队列管理器。MQe 中会进行检查，如果某个应用程序尝试启动多个队列管理器，将抛出错误。如果多个应用程序需要使用一个队列管

理器，它们必须驻留在同一 JVM 中。在一个 JVM 中运行多个应用程序时，这些应用程序可以使用 `MQeQueueManager.getDefaultQueueManager()` 方法来检查是否已经有队列管理器处于运行状态。一台机器上可以使用任意数量的应用程序和队列管理器，但必须注意期望性能和系统资源情况。

- 一个消息存储上只应启动一个 MQe 队列管理器。使用图形用户界面时，比较容易启动一个应用程序的多个实例。因此，可能启动相同消息存储上运行的同一队列管理器的多个实例。这可能导致不可预测的结果，尤其是使用异步排队时。

C C API 使您能访问与客户机队列管理器的角色相当的功能。可以通过平台调用服务使用 .NET 环境，以调用到 MQe C DLL。您必须将所需的 DLL 下载到设备上。

在创建原型阶段中，强烈建议在生产过程中要使用的网络上，使用生产级数据运行测试。这使您能对性能做出正确的期望并对 MQe 中的缺省通信设置做出评估。

开发许可证不包括 IBM 的支持。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（请参阅以下内容）。

2. 部署阶段

部署阶段是指如何使用开发的应用程序，因此根据 MQe 许可证的条款规定，使用产品需要容量单位。Java 类和 C API 只能在获得 IBM 同意或用户已经获得使用授权的情况下随应用程序一起分发。否则，在 Java 中，用户必须自行定制所需的类，并在 C 中将 MQe 复制到设备。

3. 管理阶段

当 MQe 队列管理器在网络中变为活动后，接下来就需要一些工具对它们进行检查和管理。根据“国际程序许可协议”的条款规定对 MQe 提供支持。

支持级别

这一采纳的生命周期解释平台支持级别的变化。对于带有容量单位和 3 类 SupportPac 的 MQe，IBM 区分以下几种情况：

- 支持安装和应用开发的平台：
 - 接受关于安装、应用程序开发和使用的问题报告
- 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再现问题报告
- 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

积累 MQe 经验

MQe 有多种入门方法。

- 启动并运行队列管理器，然后设置一个简单的 MQe 网络，这是熟悉产品及其概念的一个有效方法。
- 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，了解 MQ 系列的其他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，建议新用户理解文档的这一介绍部分中的概念要点。

如果您可以访问运行 Windows 操作系统的机器，请下载 SupportPac ES06 - MQe 服务器支持（其中包括 MQe_Explorer）并按照给定的说明开始使用 MQe。您不必事先安装许可产品，但如果没有安装，您将受到许可证条款的限制。

结合 MQ 使用 MQe

介绍

尽管 MQe 网络可以独立存在，而不需要 MQ 服务器或网络，但实际经常使用 MQe 补充现有的 MQ 安装。

这将 MQe 扩展到新的平台和设备上并提供一些高级能力，例如基于队列或消息的安全性或同步消息传递。

从 MQe 应用程序的角度而言，MQ 队列和队列管理器扮演了附加远程队列和队列管理器的角色。然而，因为这些队列不是通过 MQe 连接和 MQe 队列管理器直接访问的，而是需要一个 MQe 网关，所以存在一些功能限制。

网关可以直接或间接地通过 MQ 客户机通道将消息发送到多个 MQ 队列管理器。如果连接是间接的，则通过 MQ 客户机通道将消息传递到一个中间 MQ 队列管理器，然后继续通过 MQ 消息通道到达目标队列管理器。

MQ 的网关（网桥）

本部分不适用于 C 代码库。

MQe 支持 MQ 网桥，该网桥充当 MQe 与 MQ 网络之间的接口。

该网桥使用 MQ Java 客户机与一个或多个 MQ 队列管理器相接，从而使消息能从 MQe 流到 MQ，反之亦然。

在当前版本的 MQe 中：

- 建议每个服务器使用一个这样的网桥
- 每个网桥与多个 MQ 队列管理器代理（MQ 队列管理器的定义）关联
- 每个与 MQe 通信的 MQ 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，它们各自代表与单个 MQ 队列管理器的连接
- 其中每个都可以使用与队列管理器的不同 MQ 服务器连接，并可以选择一组不同的属性（如用户出口或端口）

消息转换

发往 MQ 的 MQe 消息通过网桥传递，并使用缺省转换程序或目标队列特定的转换程序将消息转换为 MQ 格式。定制转换程序提供更高的灵活性，例如使用 MQe 消息类的子类来表示 MQe 网络上的一个特定消息类型就是一个好方法。在网关上，转换程序可以使用字段与 MQ 值之间相应的映射并添加特定数据以表示子类的重要性，将消息转换为 MQ 格式。

从 MQe 到 MQ 的缺省转换程序无法利用子类信息，但它已设计为适用于各种情况。它有以下特征：

- 从 MQe 到 MQ 的消息流：

从 MQe 到 MQ 的缺省转换程序与 *MQeMQMsgObject* 类结合使用。该类表示 MQ 消息头中可以找到的所有字段。

使用 *MQeMQMsgObject*，您的应用程序可以用 *set()* 方法设置值。因此，将 *MQeMQMsgObject* 或从它派生的对象通过缺省 MQe 转换程序（即 *MQeBaseTransformer*）传递时，*MQeBaseTransformer* 从 *MQeMQMsgObject* 内部获取值，并设置 MQ 消息中相应的值（例如，复制到 MQ 消息的优先级值）。

如果传递的消息不是 *MQeMQMsgObject* 并且不是从 *MQeMQMsgObject* 类派生而来的，则将整个 MQe 消息复制到 MQ 消息体。这称为集中。设置 MQ 消息头中的消息格式字段来表明 MQ 消息中存放了一条 MQe 集中格式的消息。

• 从 MQ 到 MQe 的消息流:

MQe 的 MQ 消息与反方向传递的消息采用类似的处理方式。缺省转换程序检查 MQ 头的消息类型字段并采取相应操作。

如果 MQ 头表明集中 MQe 消息, 则按原始 MQe 消息重新构造 MQ 消息体, 然后将它发布到 MQe 网络。

如果消息不是集中 MQe 消息, 则抽取 MQ 消息头内容并将它放入 MQeMQMsgObject。MQ 消息体作为简单的字节字段处理, 然后同样将它放入 MQeMQMsgObject 中。然后将 MQeMQMsgObject 发布到 MQe 网络。

这个 MQeMQMsgObject 类和缺省转换程序的行为意味着:

- MQe 消息可以通过 MQ 网络传递到 MQe 网络, 而不发生变化。
- MQ 消息可以通过 MQe 网络传递到 MQ 网络, 而不发生变化。
- MQe 应用程序可以驱动任何现有的 MQ 应用程序, 而不更改 MQ 应用程序。

功能

从 MQe 队列管理器中, 为同步 MQe 放入消息传递操作启用 MQ 远程队列。

所有其他消息传递操作必须是异步的。

MQe 管理消息不能发送到 MQ 队列管理器。那里没有管理队列, 并且管理消息格式不同于 MQ 使用的格式。

兼容性

MQe 网络能独立于 MQ 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需求。MQe 可以集成到现有的 MQ 网络中, 其兼容性包括以下概述的几个方面:

寻址与命名:

- 使用队列管理器或队列地址的相同寻址语义
- ASCII 名称空间的共同使用

应用程序:

- MQe 能支持现有的 MQ 应用程序, 而不更改它们。

连接:

- MQe 网关使用 MQ 客户机通道。

消息交换与内容:

- 在 MQe 与 MQ 之间交换消息
- 消息网络不可视性 (来自 MQe 或 MQ 的消息可以穿过其他网络, 而不更改)
- 相互支持 MQ 消息头中的标识字段
- 一次性可靠消息传递

MQe 不支持所有 MQ 功能。除了要考虑环境、操作系统和通信因素之外, 以下是一些更重要的区别:

- 无集群支持
- 无分发列表支持
- 无分组或分段的消息
- 无负载平衡 / 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
- 不同的可伸缩性和性能特征

但在 MQe 中，许多应用程序任务都可以通过使用 MQe 功能的替代方法，或通过利用划分子类、替换提供的类或利用规则、接口以及构建到产品中的其他定制功能来实现。

可靠传递

尽管 MQe 和 MQ 都提供可靠传递，但它们分别提供不同级别的可靠性。

- 当消息从 MQe 传递到 MQ 时，只有当结合使用 *putMessage* 和 *confirmPutMessage* 时，才能确保消息的传送。
- 当消息从 MQ 传递到 MQe 时，只有当 MQ 消息定义为持久时，才能确保消息的传送。

进一步信息

MQ 的相关信息

以下是相关的 MQ 出版物，可能会对您有所帮助：

WebSphere MQ: An Introduction to Messaging and Queuing (GC33-0805)

本书简要描述了什么是 MQ、其工作原理以及它如何解决一些典型的互操作性问题。

WebSphere MQ: 快速入门系列

MQ 支持的每个平台都有相应的 MQ 快速入门书籍。这些书籍包含 MQ 的平台特定规划和安装信息。

Web 站点

MQe 主页位于：

<http://www.ibm.com/software/integration/wmq/>

通过从该主页进行链接，您可以：

- 查找更多关于 MQe 功能和优点的信息
- 获得有关培训和证书的信息
- 访问 PDF 和 HTML 格式的 MQe 手册
- 下载最新升级和试用代码。

您可以在以下页面上选择产品 *WebSphere MQ Everyplace* 下载 MQe SupportPac:

<http://www.ibm.com/software/integration/support/supportpacs/>

您可能还希望访问 MQ 的主页，它位于：

<http://www.ibm.com/software/integration/wmq/>

MQ 系列的主页位于：

<http://www.ibm.com/software/integration/mqfamily/>

您可以通过以下站点访问 MQ 系列产品的书籍库：

<http://www.ibm.com/software/integration/websphere/library/books/>

已翻译的文档

MQe 介绍书籍已翻译为各种非英语语言。您可以从以下 MQ 库 Web 站点下载这些已翻译的文档：
<http://www.ibm.com/software/integration/websphere/library/>。

新闻组

这些新闻组都位于 news.software.ibm.com 上，还将放置在许多其他公用新闻服务器上。

MQe:

- ibm.software.websphere.mqeveryplace

MQ:

- ibm.software.websphere.mq
- ibm.software.websphere.mq.administration
- ibm.software.websphere.mq.programming

其他相关内容:

- ibm.software.websphere.mqintegrator
- ibm.software.websphere.studio
- ibm.software.websphere.studio.various

MQe 证书

您可以获得 MQe 的培训和证书。要获取更多详细信息，请从以下站点着手：

<http://www.ibm.com/software/integration/websphere/education/>

声明和商标

声明

本信息是为在美国提供的产品和服务编写的。IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

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

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：International Business Machines Corporation “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本出版物的新版本中。IBM 可以随时对本出版物中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（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

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本信息中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

商标

下列术语是 International Business Machines Corporation 在美国和 / 或其他国家或地区的商标。

AIX、Everyplace、IBM、IBMLink、iSeries、MQSeries、SupportPac、WebSphere、z/OS 和 zSeries

Microsoft、Windows、Windows NT 和 Windows 徽标是 Microsoft Corporation 在美国和 / 或其他国家或地区的商标。

Java 和所有基于 Java 的商标和徽标是 Sun Microsystems, Inc. 在美国和 / 或其他国家和地区的商标或注册商标。

Linux 是 Linus Torvalds 在美国和 / 或其他国家或地区的商标。

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以及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。

如果找不到相应的术语，请尝试软拷贝搜索、参阅硬拷贝索引或 *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

C D F G J K L Q R S T W X Y Z j m s w

[C]

超类 (superclass)：超类是供其他类扩展的类。超类的公用和受保护的方法及变量对子类是可用的。

超文本标记语言 (Hypertext Markup Language, HTML)：用于定义要显示在万维网上的信息的语言。

程序包 (package)：在 Java 中，程序包是使一段 Java 代码能访问一组特定类的方法。作为特定程序包的一部分的 Java 代码可以访问程序包中的所有类以及类中的所有非专用方法和字段。

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。

[D]

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。

队列 (queue)：队列是一种 MQ 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。

对象 (object)：(1) 在 Java 中，对象是类的一个实例。类模型化一组事物；对象模型化该组的一个具体成员。(2) 在 MQ 中，对象是队列管理器、队列或通道。

[F]

方法 (method)：函数或过程的面向对象的编程术语。

封装 (encapsulation)：一种面向对象的编程技术，它使对象的数据成为专用或受到保护，并且只允许程序员通过方法调用访问和操作数据。

服务器队列管理器 (server queue manager)：有了侦听器，队列管理器可以接收和发送消息。请参阅 MQe 队列管理器。

服务器队列管理器 (server queue manager)：在 MQe 上：可以添加侦听器的队列管理器。有了侦听器，服务器队列管理器可以接收和发送消息。

服务器平台 (server platform)：一台不限大小、能以服务器或客户机形式运行 MQe 的计算机。

服务器 (server)：

1. MQe 服务器是一个配置了 MQe 侦听器的设备，它在客户机 / 服务器设置中对信息请求做出响应。
2. MQ 服务器是一个队列管理器，它为运行于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，它对客户机 / 服务器的特定两程序信息流模型中的信息请求做出响应。
4. 服务器程序运行所在的计算机。

[G]

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

公用 (public)：公用类或公用接口在任何地方都是可视的。公用方法或变量在其类可视的任何地方都是可视的。

[J]

加密程序 (cryptor)：一个程序，它对消息进行加密以在传输过程中提供安全性。

接口 (interface)：其中只包含抽象的方法而没有实例变量的类。接口提供一个公共的方法集，它可由多个不同类的子类实现。

[K]

客户机 (**client**)： 在 MQ 中，客户机是一个运行时组件，它允许本地用户应用程序向服务器发送消息。

[L]

类 (**class**)： 一个数据以及数据操作方法的封装集合。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。

连接 (**connection**)： 它链接 MQe 设备并以双向方式传送同步和异步消息及响应。

[Q]

轻量级目录访问协议 (**Lightweight Directory Access Protocol, LDAP**)： 访问目录服务的客户机 / 服务器协议。

[R]

认证程序 (**authenticator**)： 一个验证消息的发送方和接收方的程序。

[S]

设备队列管理器 (**device queue manager**)： 在 MQe 上：没有侦听器组件和网桥组件的队列管理器。因此，它只能发送消息，不能接收消息。

设备队列管理器 (**device queue manager**)： 请参阅 MQe 队列管理器。

设备平台 (**device platform**)： 一种小型计算机，它只能以客户机方式运行 MQe，即仅使用设备队列管理器。

实例 (**instance**)： 一个对象。当实例化一个类以产生一个对象时，该对象是类的一个实例。

受保护 (**protected**)： 一个受保护的字段只在其类、子类或这个类所在程序包的内部是可视的。

[T]

通道管理器 (**channel manager**)： 一个 MQe 对象，它支持端点之间的逻辑多并行通信管道。

通道 (**channel**)： 请参阅 *动态通道 (dynamic channel)* 和 *MQI 通道 (MQI channel)*。

同步消息传递 (**synchronous messaging**)： 程序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该消息的应答。对照 *异步消息传递 (asynchronous messaging)*。

[W]

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。

网关队列管理器 (**gateway queue manager**)： 在 MQe 上：可以添加侦听器和网桥的队列管理器。有了侦听器，网关队列管理器可以接收和发送消息，而有了网桥，它可以与 MQ 进行通信。

网关队列管理器 (**gateway queue manager**)： 带有侦听器和网桥的队列管理器。请参阅 MQe 队列管理器。

网关 (**gateway**)： 一台不限大小、运行 MQe 网关队列管理器的计算机，它包含 MQ 网桥功能。请参阅 MQe 队列管理器。

网桥 (**bridge**)： 一个可以添加到 MQe 队列管理器，使它能与 MQ 进行通信的组件。请参阅 MQe 队列管理器。

[X]

消息队列 (message queue)： 请参阅**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。

消息 (message)： 消息排队应用程序中程序间发送的通信。

[Y]

压缩程序 (compressor)： 一个程序，它通过压缩消息来减少发送的数据量。

异步消息传递 (asynchronous messaging)： 一种程序之间的通信方法，它将消息放入消息队列中。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该消息的应答。对照同步消息传递 (synchronous messaging)。

因特网 (internet)： 一个共享信息的协作性公用网络。物理上，因特网使用当前所有现有的公用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用网络是因为它使用称为 **TCP/IP** (传输控制协议/网际协议) 的协议集。

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。

远程队列管理器 (remote queue manager)： 该术语与远程队列定义一起使用。它描述了拥有本地队列的远程队列管理器，该本地队列是远程队列定义的目标。

[Z]

注册表 (registry)： 存储队列管理器配置信息。

专用 (private)： 一个专用字段在其自身类外是不可视的。

转换程序 (transformer)： 执行数据或消息重新格式化的一段代码。

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

j

Java 开发工具箱 (Java Development Kit, JDK)： Sun Microsystems 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、反汇编程序、Applet 查看器、存根文件生成器和文档生成器。

Java 命名和目录服务 (Java Naming and Directory Service, JNDI)： Java 编程语言中指定的一种 API。它为用 Java 编程语言编写的应用程序提供命名和目录功能。

m

MQ 和 MQ 系列 (MQ and MQ family)： 请参阅 **WebSphere MQ**，它包括以下产品：

- **WebSphere MQ Workflow** 通过对涉及人员和应用程序的业务流程实现自动化，简化了整个企业的集成。
- **WebSphere MQ Integrator** 是一个消息代理软件，它提供实时、智能化、基于规则的消息传递以及内容转换和格式化。
- **WebSphere MQ Messaging** 通过商业质量的消息传递，提供从台式机到大型机的任意连接性，支持超过 35 个平台。

MQ 网桥 (MQ bridge)： 带有网关队列管理器的计算机，它可以与 MQ 通信。请参阅 **MQe 队列管理器**。

MQ 消息传递 (MQ Messaging)： 请参阅以下 **WebSphere MQ** 消息传递产品组：

- **分布式消息传递**： MQ for Windows NT、Windows 2000、AIX、iSeries、HP-UX、Solaris 及其他平台

- 主机消息传递: MQ for z/OS
- 普及消息传递: MQe

MQe: 请参阅 **WebSphere MQ Everyplace**, MQ 普及消息传递产品组。

MQI 通道 (MQI channel): 将 MQ 客户机连接到服务器系统上的队列管理器, 并以双向方式传送 MQI 调用和响应。

S

Servlet: 一种 Java 程序, 它设计为只能在 Web 服务器上运行。

W

Web: 请参阅万维网 (World Wide Web)。

Web 浏览器 (Web browser): 一个程序, 它格式化和显示在万维网上发布的信息。

索引

[A]

安全接口 18
安全性, MQe 16

[B]

本地队列 10
编程接口 1, 3
拨号连接管理 15

[C]

操作, 队列管理器 12
词汇表 28
存储转发队列 10

[D]

队列管理器操作 12
队列管理器规则 19
队列管理器, MQe 11
队列管理器 (queue manager) 11, 15
队列规则 19
队列, 本地 10
队列, 存储转发 10
队列, 远程 10
队列, MQ 网桥 10
队列, MQe 9

[F]

法律声明 27

[G]

概述 6
跟踪 MQe 16
公用注册表 17
管理消息 12
规则, 定制 18
规则, MQe 18

[H]

欢迎使用 1

[J]

加密 16
监视 13
接口, 安全 18
接口, 编程 1, 3

[K]

可靠消息传递 26
客户机通道 8

[L]

类, MQe 19
连接 8, 13
连接管理器 15
连接样式 15
连接, 动态 8

[M]

迷你证书的发行服务 18
迷你证书发行服务 18

[P]

配置 18
配置, 示例 13

[S]

商标 28
设备, MQe 8
声明, 法律 27
使用 MQe 管理 12
事件日志 16
示例配置 13
适配器, MQe 15
属性规则 18
术语 28
宿主服务器队列 10

[T]

通道, 客户机 8
通道 (channel) 13
通信 15

[W]

网桥, MQ 24

[X]

消息 8
消息传递, 可靠 26
消息转换 24
消息, 管理 12

[Y]

压缩 16
应用程序, 装入 19
应用程序, MQe 7
与 MQ 的兼容性 25
远程队列 10

[Z]

侦听器 15
证书的复制 17
证书复制 17
注册表 17
注册表, 公用 17
注册表, 专用 17
注册表, MQe 8
专用注册表 17
转换程序 24
装入应用程序 19
自动注册 17

A

API 1, 3

M

MQ 的接口 24
MQ 网桥队列 10
MQ 网桥规则 18
MQ 网桥 (MQ bridge) 24
MQ 系列 6
MQ 消息传递 6
MQ PCF 消息 21
MQ, 兼容性 25
MQ, 接口 24
MQe 8
MQe 安全性 16

MQe 队列 9
MQe 队列管理器 11
MQe 管理 12
MQe 规则 18

MQe 简介 6
MQe 介绍 6
MQe 类 19
MQe 设备 8

MQe 适配器 15
MQe 应用程序 7
MQe 注册表 8, 17