



Sun Java™ System

Message Queue 3

管理指南

2005Q1

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

文件号码 819-2219

版权所有 © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 保留所有权利。

对于本文中介绍的产品，Sun Microsystems, Inc. 对其所涉及的技术拥有相关的知识产权。需特别指出的是（但不仅限于此），这些知识产权可能包括在 <http://www.sun.com/patents> 中列出的一项或多项美国专利，以及在美国和其他国家 / 地区申请的一项或多项其他专利或待批专利。

美国政府权利 - 商业软件。政府用户应遵循 Sun Microsystems, Inc. 的标准许可协议，以及 FAR（Federal Acquisition Regulations，即“联邦政府采购法规”）的适用条款及其补充条款。必须依据许可证条款使用。本发行版可能包含由第三方开发的内容。

Sun、Sun Microsystems、Sun 徽标、Java、Solaris、Sun[tm] ONE、JDK、Java Naming and Directory Interface、JavaMail、JavaHelp 和 Javadoc 是 Sun Microsystems, Inc. 在美国和其他国家 / 地区的商标或注册商标。

所有 SPARC 商标的使用均已获得许可，它们是 SPARC International, Inc. 在美国和其他国家 / 地区的商标或注册商标。标有 SPARC 商标的产品均基于由 Sun Microsystems, Inc. 开发的体系结构。

UNIX 是 X/Open Company, Ltd. 在美国和其他国家 / 地区独家许可的注册商标。

本产品受美国出口控制法制约，并应遵守其他国家 / 地区的进出口法律。严禁将本产品直接或间接地用于核设施、导弹、生化武器或海上核设施，也不能直接或间接地出口给核设施、导弹、生化武器或海上核设施的最终用户。严禁出口或转口到美国禁运的国家 / 地区以及美国禁止出口清单中所包含的实体，包括但不限于被禁止的个人以及特别指定的国家 / 地区的公民。

目录

| | |
|--|-----------|
| 图 | 13 |
| 表 | 15 |
| 过程 | 19 |
| 前言 | 21 |
| 目标读者 | 22 |
| 阅读本书之前 | 22 |
| 本书的结构 | 22 |
| 本书所使用的约定 | 24 |
| 文本约定 | 24 |
| 目录变量约定 | 24 |
| 相关文档 | 27 |
| Message Queue 文档集 | 27 |
| 联机帮助 | 28 |
| JavaDoc | 28 |
| 示例客户机应用程序 | 28 |
| Java 消息服务 (JMS) 规范 | 28 |
| 第三方 Web 站点 | 29 |
| Sun 欢迎您提出意见 | 29 |
| | |
| 第 I 部分 Message Queue 管理简介 | 31 |
| | |
| 第 1 章 管理任务和工具 | 33 |
| 开发环境中的管理任务 | 34 |
| 生产环境中的管理任务 | 34 |
| 设置操作 | 34 |
| 维护操作 | 35 |
| 管理工具 | 36 |

| | |
|---------------|----|
| 命令行实用程序 | 36 |
| 管理控制台 | 38 |

| | |
|---------------------------|-----------|
| 第 2 章 管理快速入门 | 39 |
| 准备工作 | 40 |
| 启动管理控制台 | 40 |
| 获得帮助 | 42 |
| 启动代理 | 43 |
| 添加代理 | 43 |
| 连接到代理 | 45 |
| 查看连接服务 | 45 |
| 将物理目标添加到代理 | 47 |
| 管理物理目标 | 48 |
| 获得主题的相关信息 | 50 |
| 使用对象存储库 | 51 |
| 添加对象存储库 | 51 |
| 检查对象存储库属性 | 53 |
| 连接到对象存储库 | 54 |
| 添加连接工厂受管理对象 | 54 |
| 添加目标对象 | 56 |
| 查看受管理对象的属性 | 57 |
| 更新控制台信息 | 58 |
| 运行样例应用程序 | 58 |

第 II 部分 管理任务 **61**

| | |
|-----------------------------------|-----------|
| 第 3 章 启动代理和客户机 | 63 |
| 准备系统资源 | 64 |
| 同步系统时钟 | 64 |
| 设置文件描述符限制 (Solaris 或 Linux) | 64 |
| 以交互方式启动代理 | 64 |
| 自动启动代理 | 66 |
| 在 Solaris 和 Linux 上自动启动 | 66 |
| 在 Windows 中自动启动 | 66 |
| 启动 Message Queue 客户机 | 68 |
| 删除代理实例 | 69 |
| 第 4 章 配置代理 | 71 |
| 关于可配置的代理组件 | 72 |
| 连接服务 | 73 |
| 消息路由器 | 76 |

| | |
|-------------------------|-----------|
| 持久性管理器 | 80 |
| 安全性管理器 | 83 |
| 监视服务 | 86 |
| 关于配置文件 | 90 |
| 实例配置文件 | 90 |
| 合并属性值 | 91 |
| 属性命名语法 | 92 |
| 编辑实例配置文件 | 92 |
| 在命令行中输入配置选项 | 92 |
| 设置持久性存储 | 93 |
| 配置文件系统存储 | 93 |
| 配置 JDBC 存储 | 94 |
| 确保持久性数据的安全 | 97 |
| 内置（基于文件的）持久性存储 | 97 |
| 插入的 (JDBC) 持久性存储 | 98 |
| 第 5 章 管理代理 | 99 |
| 前提条件 | 100 |
| 使用 imqcmd 命令实用程序 | 100 |
| 指定用户名和密码 | 100 |
| 指定代理名和端口 | 101 |
| 示例 | 101 |
| 显示帮助 | 102 |
| 显示产品版本 | 102 |
| 显示代理信息 | 102 |
| 更新代理属性 | 103 |
| 暂停和恢复代理 | 104 |
| 暂停代理 | 104 |
| 恢复代理 | 105 |
| 关闭并重新启动代理 | 105 |
| 显示代理度量 | 106 |
| 管理连接服务 | 107 |
| 列出连接服务 | 107 |
| 显示连接服务信息 | 108 |
| 更新连接服务属性 | 109 |
| 显示连接服务度量 | 109 |
| 暂停和恢复连接服务 | 110 |
| 获得有关连接的信息 | 111 |
| 管理长期订阅 | 112 |
| 管理事务 | 113 |

| | |
|----------------------------|------------|
| 第 6 章 管理物理目标 | 117 |
| 使用 imqcmd 命令实用程序 | 118 |
| 子命令 | 118 |
| 创建物理目标 | 119 |
| 列出物理目标 | 120 |
| 显示有关物理目标的信息 | 121 |
| 更新物理目标属性 | 122 |
| 暂停和恢复物理目标 | 122 |
| 清除物理目标 | 123 |
| 销毁物理目标 | 124 |
| 压缩物理目标 | 124 |
| 配置停用消息队列的使用 | 126 |
| 配置停用消息队列的使用 | 126 |
| 配置和管理停用消息队列 | 127 |
| 启用停用消息日志记录 | 128 |
| | |
| 第 7 章 管理安全性 | 129 |
| 验证用户 | 130 |
| 使用平面文件用户系统信息库 | 130 |
| 使用 LDAP 服务器管理用户系统信息库 | 135 |
| 授权用户：访问控制属性文件 | 138 |
| 创建访问控制属性文件 | 139 |
| 访问规则语法 | 139 |
| 权限的计算方式 | 140 |
| 用于连接服务的访问控制 | 141 |
| 对物理目标的访问控制 | 143 |
| 对自动创建的物理目标的访问控制 | 144 |
| 使用基于 SSL 的服务 | 144 |
| TCP/IP 的安全连接服务 | 145 |
| 配置自签名证书的使用 | 145 |
| 配置签名证书的使用 | 150 |
| 使用密码文件 | 153 |
| 安全性问题 | 154 |
| 密码文件内容 | 154 |
| 创建审计日志 | 154 |
| | |
| 第 8 章 管理受管理对象 | 157 |
| 关于对象存储库 | 157 |
| LDAP 服务器对象存储库 | 158 |
| 文件系统对象存储库 | 159 |
| 关于受管理对象的属性 | 160 |
| 连接工厂属性 | 160 |
| 客户机标识 | 163 |

| | |
|-------------------------------|------------|
| 目标受管理对象的属性 | 166 |
| 使用对象管理器实用程序 (imqobjmgr) | 167 |
| 所需的信息 | 167 |
| 使用命令文件 | 168 |
| 添加和删除受管理对象 | 170 |
| 添加连接工厂 | 170 |
| 添加主题或队列 | 171 |
| 删除受管理对象 | 173 |
| 列出受管理对象 | 174 |
| 获取有关单个对象的信息 | 174 |
| 更新受管理对象 | 175 |
| 第 9 章 使用代理群集 | 177 |
| 群集配置属性 | 178 |
| 为单个代理设置群集属性 | 178 |
| 使用群集配置文件 | 179 |
| 管理群集 | 179 |
| 连接代理 | 179 |
| 将代理添加到群集 | 180 |
| 从群集中删除代理 | 181 |
| 主代理 | 182 |
| 管理配置更改记录 | 182 |
| 主代理不可用时 | 183 |
| 第 10 章 监视消息服务器 | 185 |
| 监视工具简介 | 185 |
| 配置和使用代理日志记录 | 187 |
| 默认日志记录配置 | 187 |
| 日志消息格式 | 188 |
| 更改记录程序配置 | 188 |
| 以交互方式显示度量 | 192 |
| imqcmd metrics | 192 |
| 使用 metrics 子命令显示度量数据 | 194 |
| 度量输出: imqcmd metrics | 194 |
| imqcmd query | 196 |
| 编写应用程序来监视代理 | 196 |
| 设置基于消息的监视 | 197 |
| 安全性和访问注意事项 | 198 |
| 度量输出: 度量消息 | 199 |
| 第 11 章 分析和调整消息服务 | 201 |
| 关于性能 | 201 |

| | |
|----------------------------|------------|
| 性能调整过程 | 201 |
| 性能方面 | 202 |
| 基准检验 | 202 |
| 基线使用模式 | 203 |
| 影响性能的因素 | 204 |
| 影响性能的应用程序设计因素 | 206 |
| 影响性能的消息服务因素 | 212 |
| 调整配置以提高性能 | 216 |
| 系统调整 | 216 |
| 代理调整 | 220 |
| 客户机运行时消息流调整 | 221 |
| 第 12 章 问题疑难解答 | 225 |
| 客户机无法建立连接 | 226 |
| 连接的吞吐量太慢 | 230 |
| 客户机无法创建消息生产方 | 232 |
| 消息的生成过程发生延迟或减慢 | 233 |
| 消息堆积 | 235 |
| 消息服务器吞吐量呈间歇性 | 239 |
| 消息无法到达使用方 | 240 |
| 停用消息队列包含消息 | 244 |

第 III 部分 参考 251

| | |
|--------------------------|------------|
| 第 13 章 命令参考 | 253 |
| 命令行语法 | 254 |
| 输入命令的规则 | 254 |
| 命令行示例 | 254 |
| 通用命令选项 | 255 |
| imqbrokerd | 255 |
| 语法 | 255 |
| 命令选项 | 256 |
| 另请参见 | 259 |
| imqcmd | 260 |
| 语法 | 260 |
| 子命令 | 260 |
| 命令选项 | 266 |
| 另请参见 | 268 |
| imqobjmgr | 269 |
| 语法 | 269 |
| 子命令 | 269 |

| | |
|--------------------------------|------------|
| 命令选项 | 270 |
| 另请参见 | 271 |
| imqdbmgr | 271 |
| 语法 | 271 |
| 子命令 | 272 |
| 命令选项 | 272 |
| 另请参见 | 273 |
| imqusermgr | 273 |
| 语法 | 273 |
| 子命令 | 273 |
| 命令选项 | 274 |
| 另请参见 | 275 |
| imqsvcadmin | 275 |
| 语法 | 275 |
| 子命令 | 275 |
| 命令选项 | 275 |
| 另请参见 | 276 |
| imqkeytool | 277 |
| 语法 | 277 |
| 另请参见 | 277 |
| | |
| 第 14 章 代理属性参考 | 279 |
| 按字母顺序排列的属性列表 | 279 |
| 连接服务属性 | 283 |
| 消息路由器属性 | 285 |
| 持久性管理器属性 | 287 |
| 基于文件的持久性 | 288 |
| 基于 JDBC 的持久性 | 289 |
| 安全性管理器属性 | 291 |
| 监视和日志记录属性 | 295 |
| 群集配置属性 | 298 |
| | |
| 第 15 章 物理目标属性参考 | 301 |
| | |
| 第 16 章 受管理对象的属性参考 | 305 |
| 目标属性 | 305 |
| 连接工厂属性 | 305 |
| 连接处理 | 306 |
| 客户机标识 | 309 |
| 消息标题覆盖 | 310 |
| 可靠性和流控制 | 311 |
| 队列浏览器行为和服务器会话 | 312 |

| | |
|-------------------|-----|
| JMS 定义的属性支持 | 313 |
| SOAP 端点属性 | 313 |

| | |
|---|------------|
| 第 17 章 JMS 资源适配器属性参考 | 315 |
| ResourceAdapter JavaBean | 315 |
| ManagedConnectionFactory JavaBean | 317 |
| ActivationSpec JavaBean | 318 |

| | |
|--------------------------|------------|
| 第 18 章 度量参考 | 321 |
| JVM 度量 | 321 |
| 代理范围度量 | 322 |
| 连接服务度量 | 324 |
| 目标度量 | 325 |

第 IV 部分 附录 **329**

| | |
|---|------------|
| 附录 A Message Queue 数据在特定操作系统中的位置 | 331 |
| Solaris | 331 |
| Linux | 333 |
| Windows | 334 |

| | |
|--|------------|
| 附录 B Message Queue 接口的稳定性 | 337 |
|--|------------|

| | |
|---|------------|
| 附录 C HTTP/HTTPS 支持 | 341 |
| HTTP/HTTPS 支持体系结构 | 342 |
| 实现 HTTP 支持 | 343 |
| 步骤 1: 在 Web 服务器上部署 HTTP 隧道 Servlet | 343 |
| 步骤 2: 配置 httpjms 连接服务 | 344 |
| 步骤 3: 配置 HTTP 连接 | 346 |
| 示例 1: 在 Sun Java System Web Server 上部署 HTTP 隧道 Servlet | 347 |
| 示例 2: 在 Sun Java System Application Server 7.0 上部署 HTTP 隧道 Servlet | 350 |
| 实现 HTTPS 支持 | 352 |
| 步骤 1: 为 HTTPS 隧道 Servlet 生成自签名证书 | 352 |
| 步骤 2: 在 Web 服务器上部署 HTTPS 隧道 Servlet | 353 |
| 步骤 3: 配置 httpsjms 连接服务 | 355 |
| 步骤 4: 配置 HTTPS 连接 | 356 |
| 示例 3: 在 Sun Java System Web Server 上部署 HTTPS 隧道 Servlet | 358 |
| 示例 4: 在 Sun Java System Application Server 7.0 上部署 HTTPS 隧道 Servlet | 362 |
| 疑难解答 | 364 |
| 服务器或代理故障 | 364 |

| | |
|------------------------------|------------|
| 客户机无法通过隧道 Servlet 进行连接 | 364 |
| 词汇表 | 367 |
| 索引 | 369 |



| | | |
|--------|---|-----|
| 图 1-1 | 本地和远程管理实用程序 | 37 |
| 图 4-1 | 代理服务组件 | 72 |
| 图 4-2 | 连接服务支持 | 74 |
| 图 4-3 | 持久性管理器支持 | 80 |
| 图 4-4 | 安全性管理器支持 | 84 |
| 图 4-5 | 监视服务支持 | 87 |
| 图 4-6 | 代理配置文件 | 91 |
| 图 11-1 | 通过 Message Queue 服务传送消息 | 205 |
| 图 11-2 | 传送模式的性能影响 | 208 |
| 图 11-3 | 订阅类型的性能影响 | 210 |
| 图 11-4 | 消息大小的性能影响 | 211 |
| 图 11-5 | 传输协议速度 | 213 |
| 图 11-6 | 传输协议的性能影响 | 214 |
| 图 11-7 | 更改大小为 1K (1024 字节) 的包的 inbufsz 的结果 | 219 |
| 图 11-8 | 更改大小为 1K (1024 字节) 的包的 outbufsz 的结果 | 219 |
| 图 12-1 | QBrowser 窗口 | 242 |
| 图 12-2 | QBrowser 消息的详细信息 | 243 |
| 图 C-1 | HTTP/HTTPS 支持体系结构 | 342 |

表

| | | |
|--------|-------------------------------|-----|
| 表 1 | 本书内容 | 22 |
| 表 2 | 文档约定 | 24 |
| 表 3 | Message Queue 目录变量 | 25 |
| 表 4 | Message Queue 文档集 | 27 |
| 表 4-1 | 主要代理服务组件及功能 | 73 |
| 表 4-2 | 代理支持的连接服务 | 73 |
| 表 4-3 | 度量主题目标 | 88 |
| 表 5-1 | 代理支持的连接服务 | 107 |
| 表 5-2 | 由 imqcmd 更新的连接服务属性 | 109 |
| 表 6-1 | imqcmd 命令实用程序的物理目标子命令 | 118 |
| 表 6-2 | 物理目标磁盘占用度量 | 125 |
| 表 6-3 | 标准物理目标属性的停用消息队列处理 | 127 |
| 表 7-1 | 用户系统信息库中的初始条目 | 131 |
| 表 7-2 | imqusermgr 选项 | 132 |
| 表 7-3 | 访问规则的语法元素 | 140 |
| 表 7-4 | 物理目标访问控制规则的元素 | 143 |
| 表 7-5 | 自签名证书所需的标识名信息 | 146 |
| 表 7-6 | 使用密码的命令 | 153 |
| 表 7-7 | 密码文件中的密码 | 154 |
| 表 8-1 | LDAP 对象存储库属性 | 158 |
| 表 8-2 | 文件系统对象存储库属性 | 159 |
| 表 8-3 | 命名约定示例 | 171 |
| 表 10-1 | 度量监视工具的益处和限制 | 186 |
| 表 10-2 | 日志记录级别 | 187 |
| 表 10-3 | imqbrokerd 记录程序选项及相应的属性 | 189 |
| 表 10-4 | imqcmd metrics 子命令语法 | 192 |
| 表 10-5 | imqcmd metrics 子命令选项 | 193 |

| | | |
|---------|------------------------------|-----|
| 表 10-6 | imqcmd query 子命令语法 | 196 |
| 表 10-7 | 度量主题目标 | 197 |
| 表 11-1 | 高可靠性和高性能方案比较 | 207 |
| 表 13-1 | 通用 Message Queue 命令行选项 | 255 |
| 表 13-2 | imqbrokerd 选项 | 256 |
| 表 13-3 | imqcmd 子命令 | 260 |
| 表 13-4 | 用于管理代理的 imqcmd 子命令 | 262 |
| 表 13-5 | 用于管理目标的 imqcmd 子命令 | 263 |
| 表 13-6 | 用于管理连接服务的 imqcmd 子命令 | 264 |
| 表 13-7 | 用于管理连接服务的 imqcmd 子命令 | 265 |
| 表 13-8 | 用于管理长期订阅的 imqcmd 子命令 | 266 |
| 表 13-9 | 用于管理事务的 imqcmd 子命令 | 266 |
| 表 13-10 | imqcmd 选项 | 267 |
| 表 13-11 | imqobjmgr 子命令 | 269 |
| 表 13-12 | imqobjmgr 选项 | 270 |
| 表 13-13 | imqdbmgr 子命令 | 272 |
| 表 13-14 | imqdbmgr 选项 | 272 |
| 表 13-15 | imqusermgr 子命令 | 274 |
| 表 13-16 | imqusermgr 选项 | 274 |
| 表 13-17 | imqsvcadm 子命令 | 275 |
| 表 13-18 | imqsvcadm 选项 | 275 |
| 表 14-1 | 代理实例配置属性 | 280 |
| 表 14-2 | 连接服务属性 | 283 |
| 表 14-3 | 消息路由器属性 | 285 |
| 表 14-4 | 自动创建配置属性 | 286 |
| 表 14-5 | 必需的持久性管理器属性 | 287 |
| 表 14-6 | 基于文件的持久性的属性 | 288 |
| 表 14-7 | 基于 JDBC 的持久性的属性 | 289 |
| 表 14-8 | 安全性管理器属性 | 291 |
| 表 14-9 | 密钥库属性 | 295 |
| 表 14-10 | 监视服务属性 | 295 |
| 表 14-11 | 群集配置属性 | 298 |
| 表 15-1 | 物理目标属性 | 301 |
| 表 16-1 | 目标受管理对象的属性 | 305 |
| 表 16-2 | 连接工厂属性: 连接处理 | 306 |
| 表 16-3 | imqAddressList 属性的寻址方案 | 308 |
| 表 16-4 | 消息服务器地址示例 | 309 |

| | | |
|---------|--|-----|
| 表 16-5 | 连接工厂属性：客户机标识 | 309 |
| 表 16-6 | 连接工厂属性：消息标题覆盖 | 310 |
| 表 16-7 | 连接工厂属性：可靠性和流控制 | 311 |
| 表 16-8 | 连接工厂属性：队列浏览器行为 | 312 |
| 表 16-9 | 连接工厂属性：JMS 定义的属性支持 | 313 |
| 表 16-10 | SOAP 端点属性 | 314 |
| 表 17-1 | 资源适配器属性 | 316 |
| 表 17-2 | 受管理连接工厂属性 | 317 |
| 表 17-3 | 激活规范属性 | 318 |
| 表 18-1 | JVM 度量 | 321 |
| 表 18-2 | 代理范围度量 | 322 |
| 表 18-3 | 连接服务度量 | 324 |
| 表 18-4 | 目标度量 | 326 |
| 表 A-1 | Message Queue 数据在 Solaris 平台上的位置 | 331 |
| 表 A-2 | Message Queue 数据在 Linux 平台上的位置 | 333 |
| 表 A-3 | Message Queue 数据在 Windows 平台上的位置 | 334 |
| 表 B-1 | 接口稳定性分类方案 | 337 |
| 表 B-2 | Message Queue 接口的稳定性 | 338 |
| 表 C-1 | httpjms 连接服务属性 | 345 |
| 表 C-2 | 用于部署 HTTP 隧道 Servlet Jar 文件的 Servlet 参数 | 348 |
| 表 C-3 | httpsjms 连接服务属性 | 355 |
| 表 C-4 | 用于部署 HTTPS 隧道 Servlet Jar 文件的 Servlet 参数 | 359 |

过程

| | |
|-------------------------------------|-----|
| 显示管理控制台帮助信息 | 42 |
| 将代理添加到管理控制台 | 43 |
| 连接到代理 | 45 |
| 查看可用连接服务 | 45 |
| 将队列目标添加到代理 | 47 |
| 查看物理目标的属性 | 48 |
| 清除物理目标中的消息 | 49 |
| 删除目标 | 50 |
| 添加文件系统对象存储库 | 51 |
| 显示对象存储库属性 | 53 |
| 连接到对象存储库 | 54 |
| 将连接工厂添加到对象存储库 | 54 |
| 将目标添加到对象存储库 | 56 |
| 查看或更新目标对象的属性 | 57 |
| 运行 HelloWorldMessageJNDI 应用程序 | 58 |
| 查看记录的服务错误事件 | 67 |
| 基本传送机制 | 76 |
| 插入 JDBC 可访问的数据存储库 | 94 |
| 创建物理目标 | 120 |
| 回收未使用的物理目标磁盘空间 | 126 |
| 编辑配置文件以使用 LDAP 服务器 | 136 |
| 设置管理用户 | 137 |
| 设置基于 SSL 的连接服务 | 145 |
| 重新生成关键字对 | 147 |
| 在代理中启用基于 SSL 的服务 | 148 |
| 获取签名证书 | 150 |
| 安装签名证书 | 151 |

| | |
|---|-----|
| 配置 Java 客户机运行时 | 152 |
| 使用群集配置文件在群集中添加新代理 | 180 |
| 不使用群集配置文件向群集中添加新代理 | 181 |
| 使用命令行从群集中删除代理 | 181 |
| 使用群集配置文件从群集中删除代理 | 181 |
| 备份配置更改记录 | 182 |
| 恢复配置更改记录 | 182 |
| 更改代理的记录程序配置 | 188 |
| 要使用日志文件来报告度量信息，请执行下列操作： | 190 |
| 要使用 <code>metrics</code> 子命令，请执行下列操作： | 194 |
| 要设置基于消息的监视，请执行下列操作： | 197 |
| 实现 HTTP 支持 | 343 |
| 激活 <code>httpjms</code> 连接服务 | 344 |
| 添加隧道 Servlet | 348 |
| 配置隧道 Servlet 的虚拟路径 (Servlet URL) | 348 |
| 在 Web 服务器启动时装入隧道 Servlet | 349 |
| 禁用服务器访问日志 | 349 |
| 将 HTTP 隧道 Servlet 部署为 WAR 文件 | 349 |
| 在 Application Server 7.0 环境下部署 HTTP 隧道 Servlet | 350 |
| 修改 Application Server 的 <code>server.policy</code> 文件 | 351 |
| 实现 HTTPS 支持 | 352 |
| 激活 <code>httpsjms</code> 连接服务 | 355 |
| 配置 JSSE | 356 |
| 添加隧道 Servlet | 359 |
| 配置隧道 Servlet 的虚拟路径 (Servlet URL) | 360 |
| 在 Web 服务器启动时装入隧道 Servlet | 360 |
| 禁用服务器访问日志 | 360 |
| 修改 HTTPS 隧道 Servlet WAR 文件 | 361 |
| 将 HTTPS 隧道 Servlet 部署为 WAR 文件 | 361 |
| 在 Application Server 7.0 环境下部署 HTTPS 隧道 Servlet | 363 |
| 修改 Application Server 的 <code>server.policy</code> 文件 | 364 |

前言

Sun Java™ System Message Queue 管理指南提供了管理 Message Queue 消息传送系统所需的信息。

本书介绍了 Sun Java System Message Queue 3 2005Q1 (Message Queue 3.6)。

本前言包含以下章节：

- 第 22 页 “目标读者”
- 第 22 页 “阅读本书之前”
- 第 22 页 “本书的结构”
- 第 24 页 “本书所使用的约定”
- 第 27 页 “相关文档”
- 第 29 页 “第三方 Web 站点”
- 第 29 页 “Sun 欢迎您提出意见”

目标读者

本指南的目标读者为需要执行 Message Queue 管理任务的系统管理员以及应用程序开发者。

Message Queue 管理员负责设置和管理 Message Queue 消息传送系统，尤其是该系统的核心 Message Queue 消息服务器。

阅读本书之前

必须阅读 Message Queue 技术概述，以熟悉 Java 消息规范的 Message Queue 实现、Message Queue 服务的组件以及开发、部署和管理 Message Queue 应用程序的基本过程。

本书的结构

下表简要介绍了本手册的内容：

表 1 本书内容

| 部分 / 章节 | 说明 |
|------------------------------------|---|
| 第 I 部分 “Message Queue 管理简介” | |
| 第 1 章 “管理任务和工具” | 介绍 Message Queue 管理任务和工具。 |
| 第 2 章 “管理快速入门” | 提供一个实用教程，借助此教程，您将了解管理控制台的操作方法。 |
| 第 II 部分 “管理任务” | |
| 第 3 章 “启动代理和客户机” | 介绍如何启动 Message Queue 代理和客户机。 |
| 第 4 章 “配置代理” | 介绍如何设置和读取配置属性，并简要介绍了代理的可配置部分。此外，还介绍了如何设置文件或数据库以执行持久性功能。 |
| 第 5 章 “管理代理” | 介绍代理管理任务。 |
| 第 6 章 “管理物理目标” | 介绍与主题和队列相关的管理任务。 |
| 第 7 章 “管理安全性” | 说明与安全相关的任务，例如管理密码文件、验证、授权和加密。 |

表 1 本书内容（续）

| 部分 / 章节 | 说明 |
|------------------------------------|--|
| 第 8 章 “管理受管理对象” | 介绍对象存储库，并说明如何执行与目标受管理对象和连接工厂受管理对象相关的任务。 |
| 第 9 章 “使用代理群集” | 介绍如何设置和管理 Message Queue 代理群集。 |
| 第 10 章 “监视消息服务器” | 介绍如何设置和使用 Message Queue 监视工具。 |
| 第 11 章 “分析和调整消息服务” | 说明用于分析消息服务器性能的技术，并介绍如何调整消息服务器以优化其性能。 |
| 第 12 章 “问题疑难解答” | 提供有关如何确定 Message Queue 常见问题的原因以及解决这些问题所采取的措施的建议。 |
| 第 III 部分 “参考” | |
| 第 13 章 “命令参考” | 提供 Message Queue 命令实用程序的语法和说明。 |
| 第 14 章 “代理属性参考” | 列出并说明可用来配置代理的属性。 |
| 第 15 章 “物理目标属性参考” | 列出并说明可用来配置主题和队列的属性。 |
| 第 16 章 “受管理对象的属性参考” | 列出并说明可用来配置目标受管理对象和连接工厂受管理对象的属性。 |
| 第 17 章 “JMS 资源适配器属性参考” | 列出并说明可用来配置 Message Queue 资源适配器（用于应用服务器）的属性。 |
| 第 18 章 “度量参考” | 列出并说明由 Message Queue 代理生成的度量。 |
| 第 IV 部分 “附录” | |
| 附录 A “Message Queue 数据在特定操作系统中的位置” | 列出 Message Queue 文件在每个支持的平台中的位置。 |
| 附录 B “Message Queue 接口的稳定性” | 说明不同 Message Queue 接口的稳定性。 |
| 附录 C “HTTP/HTTPS 支持” | 介绍如何为 Message Queue 通信设置 HTTP 的使用。 |

本书所使用的约定

本节介绍本文档中使用的有关约定。

文本约定

表 2 文档约定

| 格式 | 说明 |
|------------------------|--|
| <i>italics</i> 斜体文本 | 斜体文本代表占位符。斜体文本可用相应的子句或值替换。斜体文本还用来表示保留未译的新词或术语以及要强调的词。 |
| monospace 等宽文本 | 等宽文本表示实例代码、在命令行输入的命令、目录、文件、路径名、错误消息文本、类名称、方法名称（包括签名中的所有元素）、软件包名称、保留字和 URL。 |
| [] | 方括号用来表示命令行语法语句中的可选值。 |
| 新词术语强调 | 新词或术语以及要强调的词。 |
| 《书名》 | 书名。 |
| 全部大写文本 | 全部大写的文本表示文件系统类型（GIF、TXT、HTML 等）、环境变量（IMQ_HOME）或首字母缩略词（Message Queue、JSP）。 |
| 键名 + 键名 | 用加号连接的一组同时按下的键： Ctrl+A 表示同时按下这两个键。 |
| 键名 - 键名 | 用连字符连接的一组依次按下的键： Esc-S 表示先按 Esc 键，然后释放它，再按 S 键。 |

目录变量约定

Message Queue 使用三种目录变量；其设置因平台而异。表 3 介绍了这些变量并概述了如何在 Solaris™、Windows 和 Linux 平台上使用这些变量。

表 3 Message Queue 目录变量

| 变量 | 说明 |
|-------------|--|
| IMQ_HOME | <p>Message Queue 文档中通常使用此变量，表示 Message Queue 基目录（根安装目录）：</p> <ul style="list-style-type: none"> 在 Solaris 和 Linux 上，不存在 Message Queue 根安装目录。因此，Message Queue 文档中不使用 IMQ_HOME 表示 Solaris 平台上的文件位置。 在 Solaris 和 Windows 上，对于 Sun Java System Application Server，Message Queue 根安装目录为 Application Server 基目录下的 /imq。 在 Windows 上，Message Queue 根安装目录由 Message Queue 安装程序设置（默认情况下为 C:\Program Files\Sun\MessageQueue3）。 |
| IMQ_VARHOME | <p>这是 /var 目录，其中存储了 Message Queue 临时或动态创建的配置和数据文件。可设置为指向任何目录的环境变量。</p> <ul style="list-style-type: none"> 在 Solaris 上，IMQ_VARHOME 默认为 /var/imq 目录。 在 Solaris 上，对于 Sun Java System Application Server 测试版，IMQ_VARHOME 默认为 IMQ_HOME/var 目录。 在 Windows 上，IMQ_VARHOME 默认为 IMQ_HOME\var 目录。 在 Windows 上，对于 Sun Java System Application Server，IMQ_VARHOME 默认为 IMQ_HOME\var 目录。 在 Linux 上，IMQ_VARHOME 默认为 /var/opt/sun/mq 目录。 |

表 3 Message Queue 目录变量 (续)

| 变量 | 说明 |
|---------------------|---|
| IMQ_JAVAHOME | <p>这是一个环境变量，指向 Message Queue 可执行文件所需的 Java™ 运行时 (JRE) 的位置：</p> <ul style="list-style-type: none"> • 在 Solaris 上，IMQ_JAVAHOME 按下列顺序查找 Java 运行时，但用户可以选择性地将该值设置为所需的 JRE 所在的位置。 Solaris 8 或 9: /usr/jdk/entsys-j2se /usr/jdk/jdk1.5.* /usr/jdk/j2sdk1.5.* /usr/j2se Solaris 10: /usr/jdk/entsys-j2se /usr/java /usr/j2se • 在 Linux 上，Message Queue 先按下列顺序查找 Java 运行时，但用户可以选择性地将 IMQ_JAVAHOME 值设置为所需的 JRE 所在的位置。 /usr/jdk/entsys-j2se /usr/java/jre1.5.* /usr/java/jdk1.5.* /usr/java/jre1.4.2* /usr/java/j2sdk1.4.2* • 在 Windows 上，IMQ_JAVAHOME 默认为 IMQ_HOME\jre，但用户可以选择性地将该值设置为所需的 JRE 所在的位置。 |

在本指南中，显示 IMQ_HOME、IMQ_VARHOME 和 IMQ_JAVAHOME 时，不使用特定平台的环境变量表示法或语法（例如，在 UNIX® 平台上为 \$IMQ_HOME）。路径名通常采用 UNIX 目录分隔符表示法 (/)。

相关文档

除本指南外，Message Queue 还提供了其他文档资源。

Message Queue 文档集

表 4 按照通常的使用顺序列出了 Message Queue 文档集中的文档。

表 4 Message Queue 文档集

| 文档 | 读者 | 说明 |
|--|---------|---|
| Message Queue Installation Guide | 开发者和管理员 | 介绍如何在 Solaris、Linux 和 Windows 平台上安装 Message Queue 软件。 |
| Message Queue 发行说明 | 开发者和管理员 | 包含对新功能、限制、已知错误以及技术说明的介绍。 |
| Message Queue 技术概述 | 开发者和管理员 | 介绍了 Message Queue 的概念、功能和组件。 |
| Message Queue 管理指南 | 管理员和开发者 | 提供使用 Message Queue 管理工具执行管理任务时所需的背景和信息。 |
| Message Queue Developer's Guide for Java Clients | 开发者 | 提供了有关如何开发 Java 客户机程序的信息，该程序使用 JMS 和 SOAP/JAXM 规范的 Message Queue 实现。 |
| Message Queue Developer's Guide for C Clients | 开发者 | 提供了有关如何开发 C 客户机程序的信息，该程序使用 Message Queue 消息服务的 C 接口 (C-API)。 |

联机帮助

Message Queue 包含用于执行 Message Queue 消息服务的管理任务的命令行实用程序。要访问这些实用程序的联机帮助，请参见第 13 章“命令参考”。

Message Queue 还包含图形用户界面 (GUI) 管理工具，即管理控制台 (imqadmin)。管理控制台包含上下文有关联机帮助。

JavaDoc

在以下位置提供了 JMS 和 Message Queue API 文档 (JavaDoc 格式)：

| 平台 | 位置 |
|---------|-----------------------------------|
| Solaris | /usr/share/javadoc/imq/index.html |
| Linux | /opt/sun/mq/javadoc/index.html/ |
| Windows | IMQ_HOME/javadoc/index.html |

此文档可以在任何 HTML 浏览器中浏览，如 Netscape 或 Internet Explorer。它包括标准 JMS API 文档以及 Message Queue 受管理对象的 Message Queue 特定 API（请参见 Message Queue Developer's Guide for Java Clients 的第 3 章），这些对消息传送应用程序的开发者很有帮助。

示例客户机应用程序

许多示例应用程序提供了样例客户机应用程序代码，这些示例应用程序所在的目录取决于平台（请参见附录 A“Message Queue 数据在特定操作系统中的位置”）。

请参见位于该目录及其每个子目录中的 README 文件。

Java 消息服务 (JMS) 规范

可以在以下位置查找 JMS 规范：

<http://java.sun.com/products/jms/docs.html>

规范包含样例客户机代码。

第三方 Web 站点

本文档中所引用的第三方 URL 提供了附加的相关信息。

注 Sun 对本文档中提及的第三方 Web 站点的有效性概不负责。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他材料，Sun 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的或连带产生的实际或名义损坏或损失，Sun 概不负责，也不承担任何责任。

Sun 欢迎您提出意见

Sun 致力于提高其文档的质量，并十分乐意收到您的意见和建议。

要分享您的意见，请登陆 <http://docs.sun.com>，然后单击“发送意见”。在联机表单中，请提供文档的标题和文件号码。文件号码是一个 7 位或 9 位的数字，可以在书的标题页或文档顶部找到。例如，本书的标题是《Sun Java System Message Queue 3 2005Q1 管理指南》，文件号码是 819-2219。

当您提供意见和建议时，可能需要在表单中提供文档英文版本的标题和文件号码。本文档英文版本的文件号码和标题是：819-0066，《Sun Java System Message Queue 3 2005Q1 Administration Guide》。

Sun 欢迎您提出意见

Message Queue 管理简介

第 1 章 管理任务和工具

第 2 章 管理快速入门

管理任务和工具

Sun Java™ System Message Queue 管理由若干任务以及用于执行这些任务的若干工具组成。

本章首先概述了管理任务，然后介绍了管理工具，其中着重介绍命令行管理实用程序的通用特性。本章包含以下小节：

- 第 34 页 “开发环境中的管理任务”
- 第 34 页 “生产环境中的管理任务”
- 第 36 页 “管理工具”

开发环境中的管理任务

在开发环境中，工作的重点是 Message Queue 客户机应用程序编程，编程人员通常管理自身的系统。Message Queue 消息服务器主要用于测试。在开发环境中，强调的是灵活性，管理通常包含以下惯例：

- 做最少的管理工作，主要是为开发者启动代理以便用于测试。
- 使用基于文件的内置持久性、基于文件的用户系统信息库和文件系统存储。通常，使用这些简单的配置进行开发测试就已经足够了。
- 在多代理测试中，不使用主代理。
- 使用自动创建的目标，而不使用管理员创建的目标。
- 在客户机代码中实例化受管理对象，而不是由管理员来实例化。

生产环境中的管理任务

在生产环境中，由于必须在其中可靠地部署和运行应用程序，所以管理更为重要。执行的管理任务取决于消息传送系统的复杂性和它必须支持的应用程序的复杂性。通常，这些任务可分为设置操作和维护操作。

设置操作

通常，必须至少执行以下某些（如果不是全部）设置操作：

- **管理员安全性**（保护管理工具的使用）：
 - 授权：允许特定的个人或组访问管理连接服务以及使用来自停用消息队列的消息（请参见第 141 页“用于连接服务的访问控制”和第 143 页“对物理目标的访问控制”）。
 - 如果要使用默认管理用户 (admin) 和基于文件的用户系统信息库，请更改用户密码（请参见第 135 页“更改默认的管理员密码”）。
 - 如果要授权某个组，应确保每个管理员都属于该组。
- 基于文件的用户系统信息库

基于文件的用户系统信息库只有一个管理员组 (admin)。如果要创建新的管理用户，请确保新用户属于 admin 组。

- **LDAP 用户系统信息库**

在 LDAP 服务器中创建组，或使用现有组。确保要授予管理权限的用户是该组成员，然后授权该组成员使其可以管理连接。

有关详细信息，请参见第 135 页“使用 LDAP 服务器管理用户系统信息库”。

- **一般安全性**（请参见第 7 章“管理安全性”）：
 - 验证：创建进入基于文件的用户系统信息库的入口或将代理配置为使用现有 LDAP 用户系统信息库。
(至少需要设置保护管理功能的密码。)
 - 授权：修改访问控制属性文件中的访问设置。
 - 加密：设置基于 SSL 的连接服务（请参见第 144 页“使用基于 SSL 的服务”）。
- **受管理对象**（请参见第 8 章“管理受管理对象”）：
 - 配置或设置 LDAP 对象存储库。
 - 创建连接工厂和目标受管理对象。
- **代理群集**（请参见第 9 章“使用代理群集”）：
 - 创建中心配置文件。
 - 使用主代理。
- **持久性**：确定希望代理使用插入的持久性，还是使用内置的持久性，并设置所需的存储（请参见第 93 页“设置持久性存储”）。
- **内存管理**：设置目标属性，以便消息数和为消息分配的内存容量与可用的代理内存资源相适合（请参见第 301 页表 15-1）。

维护操作

在生产环境中，需要对 Message Queue 消息服务器资源进行严密的监视和控制。应用程序性能、可靠性和安全性是优先考虑的因素，并且必须使用 Message Queue 管理工具执行如下所述的若干任务：

- **应用程序管理**
 - 通过设置 `imq.autocreate.queue` 和 `imq.autocreate.topic` 属性的值，禁用代理的自动创建功能（请参见第 285 页“消息路由器属性”）。

- 代表应用程序创建物理目标（请参见第 117 页第 6 章“管理物理目标”）。
- 设置对目标的用户访问权限（请参见第 138 页“授权用户：访问控制属性文件”）。
- 监视和管理目标（请参见第 112 页“管理长期订阅”）。
- 监视和管理长期订阅（请参见第 112 页“管理长期订阅”）。
- 监视和管理事务（请参见第 113 页“管理事务”）。
- **代理管理和调整**
 - 使用代理度量来调整和重新配置代理（请参见第 201 页第 11 章“分析和调整消息服务”）。
 - 管理代理内存资源（请参见第 201 页第 11 章“分析和调整消息服务”）。
 - 将代理添加到群集中以平衡负荷（请参见第 9 章“使用代理群集”）。
 - 恢复出现故障的代理（请参见第 64 页“以交互方式启动代理”）。
- **管理应用程序**
 - 根据需要创建其他连接工厂和目标受管理对象（请参见第 170 页“添加和删除受管理对象”）。
 - 调整连接工厂的属性值，确保 Java 客户机应用程序的行为正确无误（请参见第 8 章“管理受管理对象”）。

管理工具

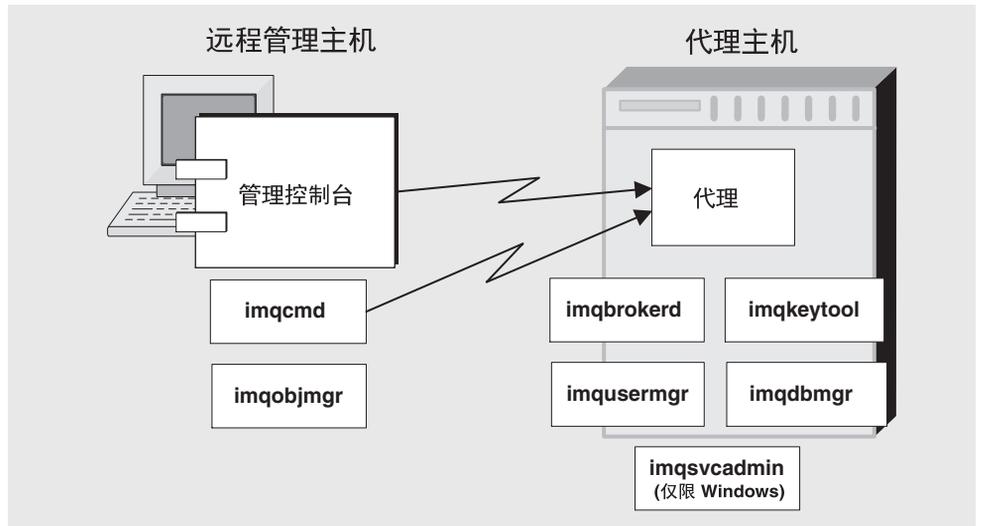
Message Queue 管理工具分为两类：

- 命令行实用程序
- 图形管理控制台 (imqadmin)

命令行实用程序

本节介绍用于执行 Message Queue 管理任务的命令行实用程序。使用 Message Queue 实用程序启动和管理代理并执行其他更加具体的管理任务。

图 1-1 本地和远程管理实用程序



所有 Message Queue 实用程序均可通过命令行接口 (CLI) 来访问。本章后面将介绍实用程序命令共用的通用格式、语约定和选项。有关使用命令行实用程序的参考信息，请参见第 13 章“命令参考”。

代理 (imqbrokerd) 使用代理实用程序启动代理。可以使用 imqbrokerd 命令的选项来指定是否应在群集中连接代理，并指定代理在启动时使用的其他配置信息。

命令 (imqcmd) 启动了代理后，您可以使用命令行实用程序创建、更新和删除物理目标；控制代理及其连接服务；管理代理的资源。

对象管理器 (imqobjmgr) 使用对象管理器实用程序在可通过 JNDI 访问的对象存储库中添加、列出、更新和删除受管理对象。通过将 JMS 客户机与 JMS 提供者特有的命名和配置格式分离，受管理对象使 JMS 客户机可以与提供者无关。

用户管理器 (imqusermgr) 使用用户管理器实用程序来填充用于验证和授权用户的基于文件的用户系统信息库。

密钥工具 (imqkeytool) 使用密钥工具实用程序来生成用于 SSL 验证的自签名证书。

数据库管理器 (imqdbmgr) 使用数据库管理器实用程序来创建和管理用于持久性存储器的 JDBC 兼容数据库。

服务管理器 (imqsvcadmin) 使用服务管理器实用程序可以将代理作为 Windows 服务来进行安装、查询和删除。

管理控制台

管理控制台结合了两个命令行实用程序的某些功能：命令行实用程序 (imqcmd) 和对象管理器实用程序 (imqobjmgr)。

可以使用管理控制台和这两个命令行实用程序来远程管理代理以及管理 Message Queue 受管理对象。其他命令行实用程序 (imqusermgr、imqdbmgr 和 imqkeytool) 必须在其关联代理的同一主机上运行，如第 37 页图 1-1 所示。

可以从联机帮助中获得有关管理控制台的信息。“[命令行实用程序](#)”介绍了通常用于执行特殊任务的命令行实用程序。

可以使用管理控制台执行以下任务：

- 连接到代理并对其进行管理。
- 在代理上创建和管理物理目标。
- 连接到对象存储库。
- 向对象存储库添加受管理对象并对其进行管理。

某些任务不能使用管理控制台来执行，包括启动代理、创建代理群集、配置更加具体的代理属性和物理目标以及管理用户数据库。

第 2 章“[管理快速入门](#)”提供了一个简明扼要的实用练习，帮助您熟悉管理控制台，并说明了如何使用它完成基本任务。

管理快速入门

本快速入门重点介绍基本管理任务，即使用具有图形界面的管理控制台来管理 Message Queue 代理和对象存储库。按照本章中的说明，您将学习如何进行以下操作：

- 启动代理。
- 连接到代理并使用管理控制台对其进行管理。
- 在代理上创建物理目标。
- 创建对象存储库，并使用管理控制台连接到对象存储库。
- 在对象存储库中添加一个目标对象并查看其属性。

本快速入门建立了运行简单的 JMS 兼容应用程序 HelloWorldMessageJNDI 所需的物理目标和受管理对象。该应用程序位于示例应用程序目录（Solaris 和 Windows 平台上为 demo，Linux 平台上为 examples；请参见附录 A “Message Queue 数据在特定操作系统中的位置”）的 helloworld 子目录中。在本快速入门的最后一部分，您将运行此应用程序。

本快速入门主要用来指导您通过使用管理控制台来执行基本管理任务。即使了解了本快速入门，您还是需要仔细阅读和参考本文档。

某些 Message Queue 管理任务无法使用管理控制台来完成。您必须使用命令行实用程序按如下所示的过程来执行此类任务：

- 配置某些物理目标属性
- 创建代理群集
- 管理用户数据库

有关如何完成这些任务的详细信息，请参见第 6 章 “管理物理目标”、第 9 章 “使用代理群集” 和第 7 章 “管理安全性”

准备工作

在开始之前，必须先安装 Message Queue 产品。有关详细信息，请参见 Message Queue 安装指南。请注意，本章以 Windows 为中心，并为 UNIX® 用户添加了说明。

在本章中，选择“项目 1” > “项目 2” > “项目 3”表示，您应该下拉名为“项目 1”的菜单，从该菜单选择“项目 2”，然后从“项目 2”提供的选择中选择“项目 3”。

启动管理控制台

要启动管理控制台，请使用以下方法之一：

- 在 Windows 上，选择“开始” > “程序” > "Sun Microsystems" > "Sun Java System Message Queue 3.6" > “管理”。

- 在 Solaris 上，输入下列命令：

```
/usr/bin/imqadmin
```

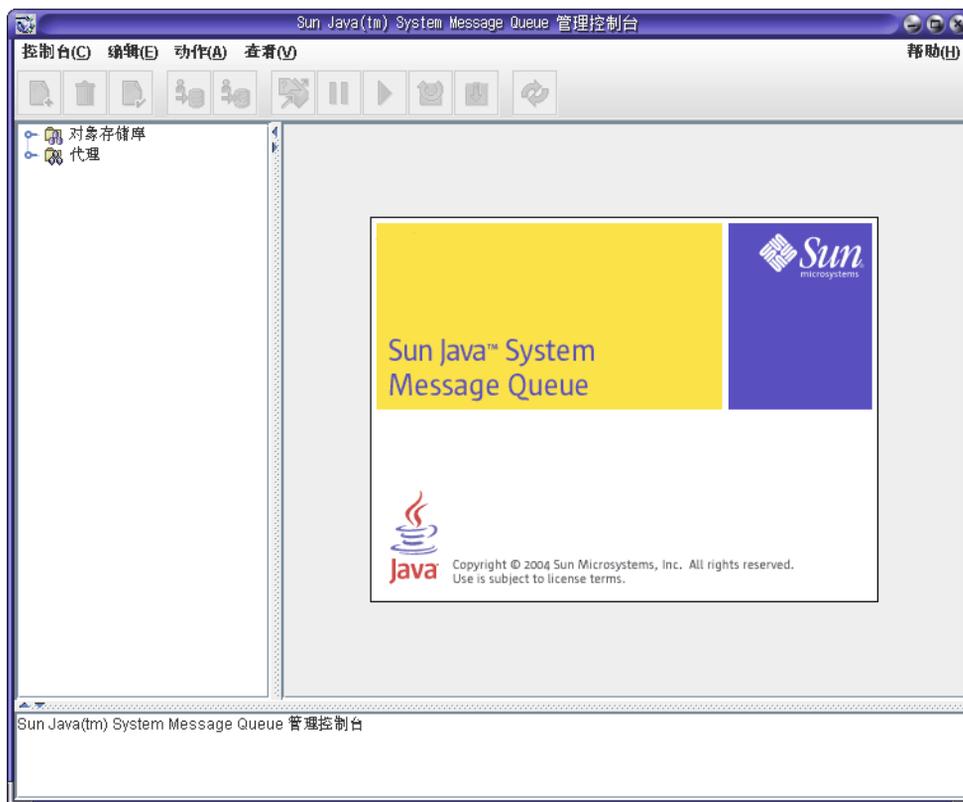
- 在 Linux 上，输入下列命令：

```
/opt/sun/mq/bin/imqadmin
```

显示控制台窗口之前，您可能需要等待几秒钟。

请花几秒钟仔细观察一下控制台窗口。

控制台包括位于顶部的菜单栏、菜单栏下面的工具栏、左侧的浏览窗格、右侧的结果窗格（现在显示标识 Sun Java System Message Queue 产品的图形）和底部的状态窗格。



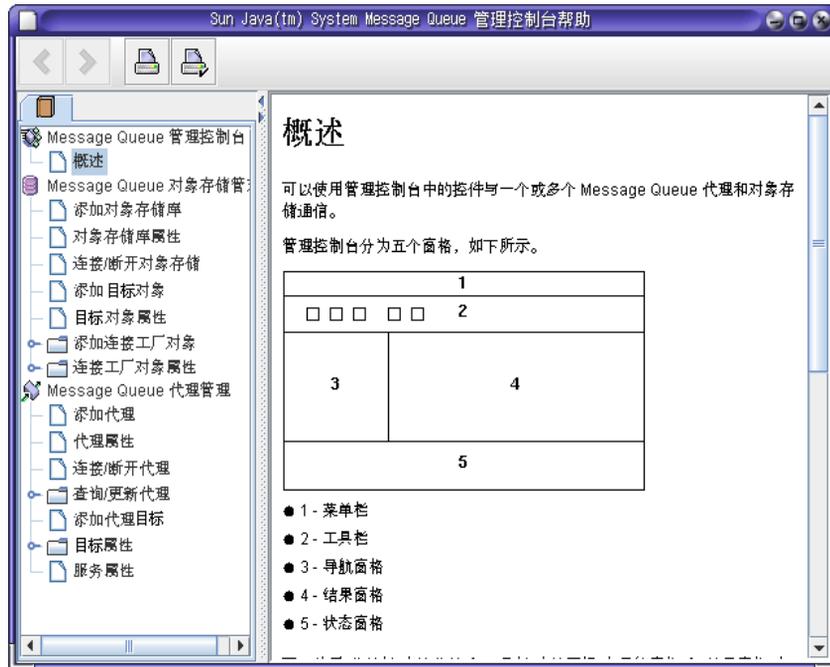
本章不可能提供全部的信息，因此让我们首先了解如何获得管理控制台的帮助信息。

获得帮助

在菜单栏的最右侧找到“帮助”菜单。

► 显示管理控制台帮助信息

1. 下拉“帮助”菜单并选择“概述”。将显示一个帮助窗口。



请注意帮助信息是如何组织的。左侧的浏览窗格显示目录，右侧的结果窗格显示在浏览窗格中选择的任何项的内容。

观察“帮助”窗口的结果窗格。它显示管理控制台的概貌，并标识了每个控制台窗格的用途。

2. 观察“帮助”窗口的浏览窗格。它使用以下三个区域来组织主题：概述、对象存储库管理和代理管理。其中每个区域都包含文件和文件夹。每个文件夹提供的是包含多个选项卡的对话框的帮助，每个文件则提供简单对话框或选项卡的帮助。

您的第一个控制台管理任务（请参见第 43 页“添加代理”）将是通过控制台创建对所管理的代理的引用。但是，启动之前请查阅联机帮助以获得有关信息。

3. 单击“帮助”窗口的浏览窗格中的“添加代理”项。

您会发现结果窗格发生了变化。它现在包含一些文本，说明添加代理的含义并介绍“添加代理”对话框中每个字段的使用。字段名称以粗体文本显示。

4. 仔细阅读帮助文本。
5. 关闭“帮助”窗口。

启动代理

使用管理控制台无法启动代理。相反，请使用下列方法之一：

- 在 Windows 上，选择“开始” > “程序” > "Sun Microsystems" > "Sun Java System Message Queue 3.6" > “消息代理”。
- 在 Solaris 上，输入下列命令：

```
/usr/bin/imbrokerd
```

- 在 Linux 上，输入下列命令：

```
/opt/sun/mq/bin/imbrokerd
```

如果使用 Windows “开始”菜单，将会出现命令窗口。然后出现命令响应，通过显示如下所示的各行内容来表明代理已准备就绪：

```
Loading persistent data...  
Broker "imbroker@stan:7676 ready.
```

现在让我们把注意力转回到“管理控制台”窗口。您现在可以将代理添加到控制台并连接到代理。

在管理控制台中添加对代理的引用之前不必启动代理，但是必须启动代理才能连接到它。

添加代理

添加代理在管理控制台中创建了一个对该代理的引用。添加代理之后，您就可以连接到它。

► 将代理添加到管理控制台

1. 在浏览窗格的“代理”上单击右键并选择“添加代理”。
2. 在“代理标签”字段中输入 MyBroker。

这样便提供了一个在管理控制台中标识代理的标签。



注意，在对话框中已指定默认主机名 (localhost) 和默认主端口 (7676)。这些值是以后必须指定的值，即在您配置客户机用来建立与此代理的连接的连接工厂时需要指定这些值。

将“密码”字段保留为空。如果在连接时指定密码，则密码将会更安全。

- 单击“确定”以添加代理。

观察浏览窗格。“代理”下方应已列出刚才添加的代理。代理图标上的红色 X 表示该代理当前未连接到控制台。



- 在 "MyBroker" 上单击右键，并从弹出式菜单中选择“属性”。
将显示代理属性对话框。可以使用此对话框更新添加代理时指定的任意属性。
- 单击“取消”关闭对话框。

连接到代理

► 连接到代理

1. 在 "MyBroker" 上单击右键并选择 “连接到代理”。

将显示一个对话框，要求输入用户名和密码。



默认情况下，管理控制台可以作为用户 admin 并使用密码 admin 连接到代理。在本练习中，请使用默认值。在实际环境中，您应尽快设置安全的用户名和密码。有关详细信息，请参见第 130 页 “验证用户”。

2. 在 “密码” 字段中输入 admin。

指定用户名 admin 并提供正确密码将使您以管理员权限连接到代理。

3. 单击 “确定” 连接到代理。

连接到代理后，可以从 “操作” 菜单中选择各个菜单项，以获得有关代理的信息、暂停和恢复代理、关闭和重新启动代理以及从代理断开连接。

查看连接服务

从代理提供的连接服务及其支持的物理目标可以区分代理。

► 查看可用连接服务

1. 在浏览窗格中选择 “服务”。

在结果窗格中列出了可用服务。对于每项服务，均提供了其名称、端口号和状态。



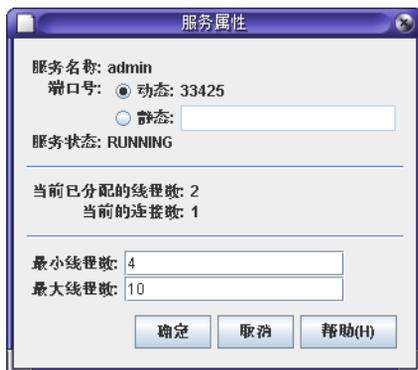
2. 单击结果窗格中的 "jms" 服务，以选中它。

3. 下拉 “操作” 菜单并注意高亮显示的项。

您可以选择暂停 jms 服务或查看和更新其属性。

4. 从 “操作” 菜单中选择 “属性”。

请注意，在 “服务属性” 对话框中，您可以分配服务的静态端口号并可以更改分配给此服务的最小和最大线程数。



5. 单击 “确定” 或 “取消” 关闭 “属性” 对话框。

6. 在结果窗格中选择 "admin" 服务。

7. 下拉 “操作” 菜单。

您会注意到无法暂停此服务，暂停项已被禁用。admin 服务是指向代理的管理员链接。如果您暂停该服务，将再也无法访问代理。

8. 选择 “操作” > “属性” 查看 admin 服务的属性。

9. 完成时请单击“确定”或“取消”。

将物理目标添加到代理

默认情况下，为代理启用物理目标自动创建功能。自动创建功能使代理可以动态创建物理目标。

在开发环境中，您不必为了测试客户机代码而明确地创建物理目标。

在生产设置中，建议您明确地创建物理目标。这样，您（管理员）就可以完全掌握代理上使用的物理目标。

现在在代理中添加一个物理目标。请注意分配给目标的名称，您以后会在创建对应此物理目标的受管理对象时用到此名称。

► 将队列目标添加到代理

1. 在 "MyBroker" 的“目标”节点上单击右键并选择“添加代理目标”。

将显示以下对话框：



添加代理目标

目标名称:

目标类型: 队列
 主题

最大消息数:
 无限制

消息的最大总计大小:
 无限制
 字节

每条消息的最大大小:
 无限制
 字节

最大生产者数:
 无限制
 100

最大活动用户数:
 无限制
 1

最大备份用户数:
 无限制
 0

确定 重置为默认值 取消 帮助(H)

2. 在“目标名称”字段中输入 `MyQueueDest`。
3. 选择“队列”单选按钮（如果尚未选中此按钮）。
4. 单击“确定”添加物理目标。

现在，物理目标显示在结果窗格中。

管理物理目标

在代理上添加一个物理目标后，您可以按照稍后的步骤说明来执行以下任意一项任务：

- 查看和更新物理目标的属性
- 清除物理目标上的消息
- 删除物理目标

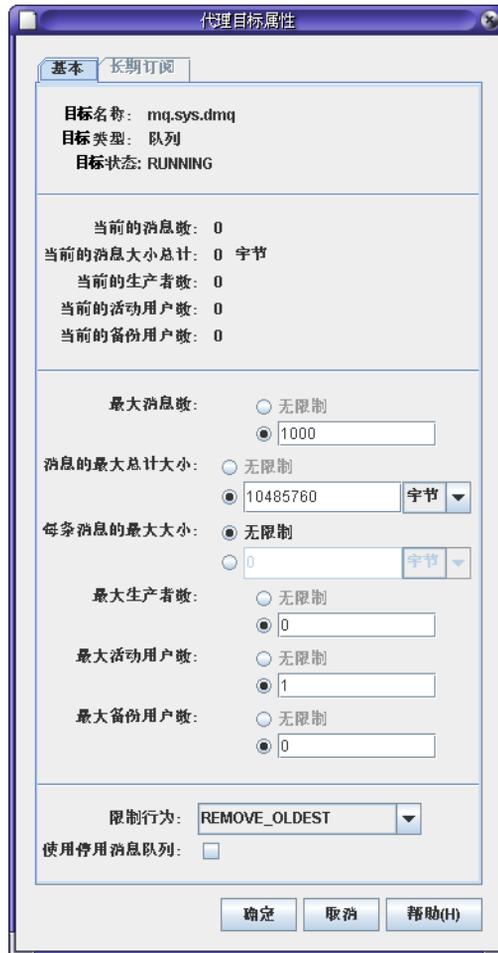
► 查看物理目标的属性

1. 选择 "MyBroker" 的“目标”节点。

结果面格中出现两个物理目标 `MyQueueDest` 和 `mq.sys.dmq`。`mq.sys.dmq` 目标是系统创建的队列，其中存储过期和被拒绝的代理消息。暂时，先忽略此停用消息队列。

2. 在结果窗格中选择 "MyQueueDest"。
3. 选择“操作” > “属性”。

将显示以下对话框：



注意，对话框显示有关队列以及您可更改的某些属性的当前状态信息。

4. 单击“取消”关闭对话框。

► 清除物理目标中的消息

1. 在结果窗格中选择物理目标。
2. 选择“操作” > “清除消息”。

将显示一个确认对话框。

清除消息将删除消息，留下一个空目标。

► 删除目标

1. 在结果窗格中选择物理目标。
2. 选择“编辑” > “删除”。

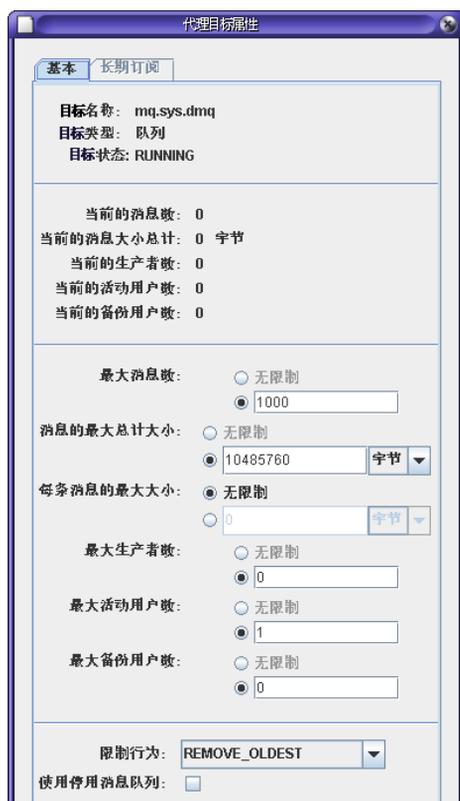
将显示一个确认对话框。

注 请不要删除 MyQueueDest 队列目标。

删除物理目标操作将清除该目标上的消息并删除目标。

获得主题的相关信息

代理主题目标属性对话框包含一个列出长期订阅的相关信息的附加选项卡。对于队列，该选项卡处于禁用状态。



使用此对话框可以执行以下操作：

- 清除长期订阅，从而删除与长期订阅关联的所有消息
- 删除长期订阅，从而清除与长期订阅关联的所有消息并删除该长期订阅

使用对象存储库

对象存储库用于存储 Message Queue 受管理对象。这些受管理对象封装有关对象（由客户机应用程序使用）的 Message Queue 特有实现和配置信息。对象存储库可以是 LDAP 目录服务器或文件系统存储（文件系统目录中的目录）。

可以在客户机代码中实例化和配置受管理对象。但是，管理员最好在客户机应用程序能够使用 JNDI 进行访问的对象存储库中创建、配置和存储这些对象。这将使客户机代码与提供者无关。

您不能使用管理控制台来创建对象存储库。必须按照以下章节中的说明提前创建。

添加对象存储库

添加对象存储库将在管理控制台中创建一个对现有对象存储库的引用。即使退出并重新启动控制台，此引用也将保留。

► 添加文件系统对象存储库

1. 如果在您的 C 驱动器上没有名为 Temp 的文件夹，那么就请创建这个文件夹。

本章中使用的样例应用程序假定对象存储库是 C 驱动器上一个名为 Temp 的文件夹。文件系统对象存储库通常可以是任何驱动器上的任何目录。

非 Windows 用户：可以使用现有的 /tmp 目录（应该已经存在）。

2. 在“对象存储库”上单击右键并选择“添加对象存储库”。

将显示以下对话框：



3. 在名为“对象存储库标签”的字段中输入 MyObjectStore。

这仅是在管理控制台中提供一个对象存储库显示的标签。

在以下步骤中, 必须输入 JNDI 名称 / 值对。JMS 兼容应用程序使用这些名称 / 值对查找受管理对象。

4. 从“名称”下拉列表中, 选择 java.naming.factory.initial。

此属性使您可以指定希望使用哪个 JNDI 服务提供商。例如, 文件系统服务提供商或 LDAP 服务提供商。

5. 在“值”字段中, 输入以下内容

```
com.sun.jndi.fscontext.RefFSContextFactory
```

这表示您将使用文件系统存储。(对于 LDAP 存储, 将指定 com.sun.jndi.ldap.LdapCtxFactory。)

在生产环境中, 您可能希望将 LDAP 目录服务器用作对象存储库。有关设置服务器和进行 JNDI 查找的信息, 请参见第 158 页“LDAP 服务器对象存储库”。

6. 单击“添加”按钮。

您会注意到现在属性概要窗格中列出了属性及其值。

7. 从“名称”下拉列表中, 选择 java.naming.provider.url。

此属性使您可以指定对象存储库的确切位置。对于文件系统类型的对象存储库, 这将是现有目录的名称。

8. 在“值”字段中, 输入以下内容

file:///C:/Temp

(在 Solaris 和 Linux 操作系统上为 file:///tmp)

9. 单击“添加”按钮。

您会注意到现在属性概要窗格中列出了属性及其值。如果正在使用 LDAP 服务器，还可能必须要指定验证信息。这对文件系统存储不是必需的。

10. 单击“确定”添加对象存储库。

11. 如果未在浏览窗格中选中节点 "MyObjectStore"，那么现在请选中它。

现在，管理控制台的显示如下所示：



浏览窗格中列出了该对象存储库，结果窗格中则列出了对象存储库的内容“目标”和“连接工厂”。我们尚未将任何受管理对象添加到对象存储库，如结果窗格的“计数”列中所示。

浏览窗格中的对象存储库图标上绘制了一个红色的 X。这表示它处于断开连接的状态。必须先连接到对象存储库，然后才能使用它。

检查对象存储库属性

当管理控制台从对象存储库断开连接时，可以查看和更改对象存储库的某些属性。

► 显示对象存储库属性

1. 在浏览窗格的 "MyObjectStore" 上单击右键。
2. 从弹出式菜单中选择“属性”。

将显示一个对话框，其中显示所有在添加对象存储库时指定的属性。您可以更改这些属性中的任意属性，并单击“确定”更新旧信息。

3. 单击“确定”或“取消”关闭对话框。

连接到对象存储库

必须先连接到对象存储库，然后才能将对象添加其中。

► 连接到对象存储库

1. 在浏览窗格的 "MyObjectStore" 上单击右键。
2. 从弹出式菜单中选择“连接到对象存储库”。

您会注意到对象存储库图标上的红色 X 不见了。现在可以将对象、连接工厂和目标添加到对象存储库。

添加连接工厂受管理对象

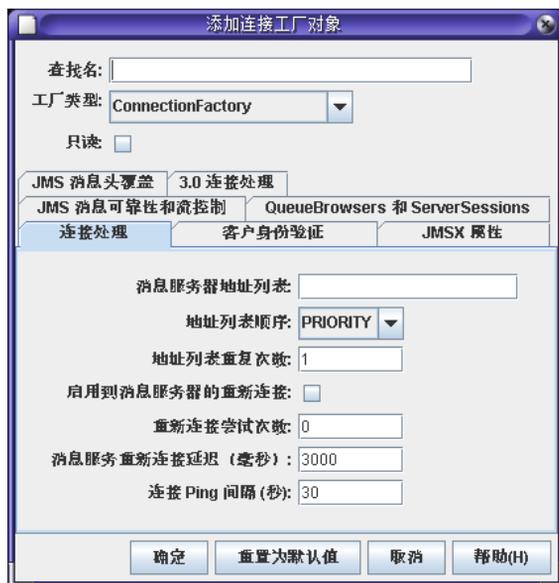
可以使用管理控制台来创建和配置连接工厂。客户机代码使用连接工厂连接到代理。通过配置连接工厂，可以控制使用连接工厂创建的连接的行为。

有关配置连接工厂的信息，请参见联机帮助和 *Message Queue Developer's Guide for Java Clients*。

注 管理控制台仅列出和显示了 *Message Queue* 受管理对象。如果对象存储库中包含与要添加的受管理对象具有相同查找名称的非 *Message Queue* 对象，则您将在尝试执行添加操作时收到一条错误消息。

► 将连接工厂添加到对象存储库

1. 如果仍未连接，请连接到 *MyObjectStore*（请参见第 54 页“连接到对象存储库”）
2. 在“连接工厂”节点上单击右键，并选择“添加连接工厂对象”。
将显示“添加连接工厂对象”对话框。



3. 在“查找名”字段中输入名称 "MyQueueConnectionFactory"。
这是客户机代码查找连接工厂时使用的名称，如 HelloWorldMessageJNDI.java 中的以下代码行所示：

```
qcf=(javax.jms.QueueConnectionFactory)
      ctx.lookup("MyQueueConnectionFactory")
```
4. 从下拉菜单中选择 "QueueConnectionFactory" 以指定连接工厂的类型。
5. 单击“连接处理”选项卡。
6. 您通常会在“消息服务器地址列表”字段中输入客户机将连接的代理的地址。此字段的一个示例如下所示：

```
mq://localhost:7676/jms
```

您不需要输入值，因为默认情况下已经将连接工厂配置为连接在端口 7676 的本地主机上运行的代理，这是快速入门示例所期望的配置。
7. 逐个单击此对话框的各个选项卡，查看您可以配置的连接工厂的信息类型。使用“添加连接工厂对象”对话框右下角的“帮助”按钮来获得有关各个选项卡的信息。目前请勿更改任何默认值。
8. 单击“确定”创建队列连接工厂。
9. 观察结果窗格。列出了新创建的连接工厂的查找名称和类型。

添加目标对象

目标受管理对象与代理中的物理目标相关联，并且指向这些目标。目标受管理对象使客户机可以查找并找到物理目标，而与提供者特定的目标名称和配置无关。

当客户机发送消息时，它将查找或实例化目标受管理对象，并在 JMS API 的 `send()` 方法中引用它。然后，代理负责将消息传送给与该受管理对象相关联的物理目标，如下所示：

- 如果已创建与该受管理对象相关联的物理目标，则代理会将消息传送给该物理目标。
- 如果尚未创建物理目标，并且物理目标的自动创建功能已启用，代理将自动创建物理目标并将消息传送给该目标。
- 如果尚未创建物理目标，并且物理目标的自动创建功能已被禁用，代理将无法创建物理目标，也就无法传送消息。

在本快速入门的下一部分中，您将添加一个与前面添加的物理目标相对应的受管理对象。

► 将目标添加到对象存储库

1. 在浏览窗格中的“目标”节点（在“MyObjectStore”节点下）上单击右键。
2. 选择“添加目标对象”。

管理控制台将显示一个“添加目标对象”对话框，用于指定有关对象的信息。



3. 在“查找名”字段中输入“MyQueue”。

JNDI 查找调用使用该查找名称来查找对象。在样例应用程序中，此调用如下所示：

```
queue=(javax.jms.Queue) ctx.lookup("MyQueue");
```

4. 对于“目标类型”，请选择“队列”单选按钮。
5. 在“目标名称”字段中输入 MyQueueDest。
这是您在代理上添加物理目标时指定的名称（请参见第 47 页“将物理目标添加到代理”）。
6. 单击“确定”。
7. 在浏览窗格中选择“目标”，并注意在结果窗格中如何显示有关刚才添加的队列目标受管理对象的信息。



查看受管理对象的属性

要查看或更新受管理对象的属性，请在浏览窗格中选择“目标”或“连接工厂”，然后在结果窗格中选择特定对象，并选择“操作” > “属性”。

► 查看或更新目标对象的属性

1. 在浏览窗格中选择 "MyObjectStore" 的“目标”节点。
2. 在结果窗格中选择 "MyQueue"。
3. 选择“操作” > “属性”以查看“目标对象属性”对话框。

请注意，您仅能更改目标名称和说明的值。要更改查找名称，就必须先删除对象，然后添加一个命名为所需查找名称的新队列受管理对象。

4. 单击“取消”关闭对话框。



更新控制台信息

不管使用对象存储库还是使用代理，都可以通过选择“查看” > “刷新”来更新任何元素或元素组的显示。

运行样例应用程序

样例应用程序 HelloWorldMessageJNDI 是为与本快速入门配套使用而提供的。它使用的是您所创建的物理目标和受管理对象：

- 名为 MyQueueDest 的队列物理目标
- JNDI 查找名称分别为 MyQueueConnectionFactory 和 MyQueue 的队列连接工厂受管理对象和队列受管理对象

代码创建简单的队列发件人和收件人，并发送和接收一条 "Hello World" 消息。

► 运行 HelloWorldMessageJNDI 应用程序

1. 使包含 HelloWorldmessageJNDI 应用程序的目录成为当前目录，例如：

```
cd IMQ_HOME\demo\helloworld\helloworldmessagejndi (Windows)
```

```
cd /usr/demo/imq/helloworld/helloworldmessagejndi (Solaris)
```

```
cd /opt/sun/mq/examples/helloworld/helloworldmessagejndi (Linux)
```

您应查看 HelloWorldMessageJNDI.class 文件是否存在。（如果要更改该应用程序，必须使用 Message Queue Developer's Guide for C Clients 的快速入门教程中用于编译客户机应用程序的指令对其重新编译。）设置 CLASSPATH 变量以便包括含有 HelloWorldMessageJNDI.class 文件及 Message Queue 产品附带的以下 jar 文件的当前目录：jms.jar、imq.jar 和 fscontext.jar。有关设置 CLASSPATH 的说明，请参见 Message Queue Developer's Guide for Java Clients。

JNDI jar 文件 (jndi.jar) 是随 JDK 1.4 一起提供的。如果您使用的是该 JDK，则不必将 jndi.jar 添加到 CLASSPATH 设置。如果您使用的是 JDK 的早期版本，则必须将 jndi.jar 包含在 CLASSPATH 中。有关其他信息，请参见 Message Queue Developer's Guide for Java Clients。

2. 运行应用程序之前，请打开源文件 HelloWorldMessageJNDI.java 并仔细阅读该源文件。虽然这个源文件很简短，但是其注释非常详细，您可以很清楚地了解该应用程序是如何使用您所创建的受管理对象和目标。
3. 通过执行以下命令之一来运行 HelloWorldMessageJNDI 应用程序：

```
java HelloWorldMessageJNDI （在 Windows 操作系统上）
```

```
% java HelloWorldMessageJNDI file:///tmp （在 Solaris 和 Linux 操作系统上）
```

如果应用程序成功运行，则会看到以下输出：

```
java HelloWorldMessageJNDI
正在使用 file:///C:/Temp for Context.PROVIDER_URL

正在以此查找名称查找队列连接工厂对象: MyQueueConnectionFactory
找到队列连接工厂对象。
正在以此查找名称查找队列对象: MyQueue
找到队列对象。

正在建立与代理的连接。
已建立与代理的连接。

正在向队列发布消息: MyQueueDest
已接收以下消息: Hello World
```

运行样例应用程序

管理任务

- 第 3 章 启动代理和客户机
- 第 4 章 配置代理
- 第 5 章 管理代理
- 第 6 章 管理物理目标
- 第 7 章 管理安全性
- 第 8 章 管理受管理对象
- 第 9 章 使用代理群集
- 第 10 章 监视消息服务器
- 第 11 章 分析和调整消息服务
- 第 12 章 问题疑难解答

启动代理和客户机

安装 Sun Java™ System Message Queue 后执行某些预备步骤后，即可启动代理和客户机。

本章包含以下小节：

- [第 64 页 “准备系统资源”](#)
- [第 64 页 “以交互方式启动代理”](#)
- [第 66 页 “自动启动代理”](#)
- [第 68 页 “启动 Message Queue 客户机”](#)
- [第 69 页 “删除代理实例”](#)

代理实例的配置由一组配置文件和使用 `imqbrokerd` 命令传送的选项共同控制，这些选项会覆盖配置文件中相应的属性。有关代理配置的信息，请参见 [第 71 页第 4 章 “配置代理”](#)。

准备系统资源

在启动代理之前，需要执行两项系统级别的任务：同步系统时钟，并在 Solaris 或 Linux 上设置文件描述符限制。以下各节将介绍这些任务。

同步系统时钟

在启动任何代理或客户机之前，同步将要与 Message Queue 系统进行交互的所有主机的时钟至关重要。如果使用消息失效期 (TimeToLive)，则同步尤为重要。来自未同步时钟的时间戳可能会使 TimeToLive 功能无法按预期方式工作，并且可能会阻止消息传送。同步对于代理群集同样至关重要。

对系统进行配置以运行时间同步协议，如简单网络时间协议 (Simple Network Time Protocol, SNTP)。通常，Solaris 和 Linux 系统中的 xntpd 守护程序以及 Windows 系统中的 W32Time 时间服务支持时间同步。有关配置此服务的信息，请参见操作系统文档。

在代理运行后，要避免向后设置系统时钟。

设置文件描述符限制（Solaris 或 Linux）

在 Solaris 和 Linux 平台上，客户机或代理在其中运行的 shell 对进程可以使用的文件描述符的数量进行了不严格的限制。在 Message Queue 系统中，客户机创建的每个连接，或代理接受的每个连接都使用其中一种文件描述符。每个具有持久性消息的物理目标也使用文件描述符。

因此，连接数由这些因素限制。如果不更改文件描述符限制，Solaris 上代理或客户机运行的连接不能超过 256 个，Linux 上不能超过 1024 个。（连接限制实际上要少一些，因为使用文件描述符可保持持久。）

要更改文件描述符限制，请参见 ulimit 手册页。此限制需在要执行客户机或代理的每个 shell 中更改。

以交互方式启动代理

可以使用 imqbrokerd 命令，从命令行中以交互方式启动代理。（或者，也可以在 Windows 上从“开始”菜单启动代理。）不能使用管理控制台 (imqadmin) 或命令行实用程序 (imqcmd) 启动代理；代理必须已经运行，才能使用这些工具。

在 Solaris 和 Linux 平台中，代理实例必须始终由最初启动此实例的用户启动。第一次启动代理实例时，Message Queue 使用该用户的 `umask` 来设置包含配置信息和持久性数据的代理实例目录的权限。每个代理实例都有其自身的一组配置属性和基于文件的消息存储。

默认情况下，代理实例具有实例名 `imqbroker`。要在命令行中使用此名称和默认配置启动代理，只需使用命令

```
imqbrokerd
```

该命令启动本地计算机上名为 `imqbroker` 的代理实例，且端口映射器的默认端口是 7676。

要指定非默认实例名，请使用 `imqbrokerd` 命令的 `-name` 选项。以下命令启动实例名为 `myBroker` 的代理：

```
imqbrokerd -name myBroker
```

在 `imqbrokerd` 命令行中，还可以使用其他选项来控制各个方面的代理操作。下面的示例使用 `-tty` 选项向命令窗口发送错误和警告（标准输出）：

```
imqbrokerd -name myBroker -tty
```

也可以在命令行中使用 `-D` 选项来覆盖在代理的实例配置文件 (`config.properties`) 中指定的属性值。下面的示例设置了 `imq.jms.max_threads` 属性，将 `jms` 连接服务可用的最大线程数提高到 2000：

```
imqbrokerd -name myBroker -Dimq.jms.max_threads=2000
```

有关 `imqbrokerd` 命令的语法、子命令和选项的完整信息，请参见第 13 章“命令参考”。要了解此信息的简要概述，请输入命令

```
imqbrokerd -help
```

注 如果您拥有 Sun Java System Message Queue Platform Edition 许可证，则可使用 `imqbrokerd` 命令的 `-license` 选项来激活试用企业版许可证，它允许您试用企业版功能 90 天。指定 `try` 作为许可证名称：

```
imqbrokerd -license try
```

每次启动代理时都必须使用此选项，否则，代理将默认为标准平台版许可证。

自动启动代理

您可以将代理设置为在系统启动时自动启动，而不是在命令行中显式启动它。具体操作方法取决于运行代理的平台（Solaris、Linux 或 Windows）。

在 Solaris 和 Linux 上自动启动

在 Solaris 和 Linux 系统中，Message Queue 安装期间将允许自动启动的脚本放在 `/etc/rc*` 目录树中。要允许使用这些脚本，您必须按如下所示编辑配置文件 `/etc/imq/imqbrokerd.conf` (Solaris) 或 `/etc/opt/sun/mq/imqbrokerd.conf` (Linux)：

- 要在系统启动时自动启动代理，请将 `AUTOSTART` 属性设置为 `YES`。
- 要在退出异常后自动重新启动代理，请将 `RESTART` 属性设置为 `YES`。
- 要设置代理的启动命令行参数，请为 `ARGS` 属性指定一个或多个值。

在 Windows 中自动启动

要在 Windows 系统启动时自动启动代理，必须将代理定义为 Windows 服务。在 Windows 系统上，可以在安装 Message Queue 时将代理作为服务进行安装。安装后，可以使用服务管理器实用程序 `imqsvcadm` 执行下列操作：

- 将代理作为 Windows 服务添加。
- 确定代理服务的启动选项。
- 删除作为 Windows 服务运行的代理。

有关 `imqsvcadm` 命令的语法、子命令和选项的参考信息，请参见第 13 章“命令参考”

将代理作为 Windows 服务安装意味着它将在系统启动时启动，并在后台运行直到系统关闭。因此，不要使用 `imqbrokerd` 命令启动代理，除非您希望启动其他实例。

要将启动选项传递给代理，请使用 `imqsvcadm` 命令的 `-args` 参数。其工作方式与 `imqbrokerd` 命令的 `-D` 选项相同，如第 64 页“以交互方式启动代理”中所述。照常使用 `imqcmd` 命令来控制代理操作。

将代理作为 Windows 服务运行时，任务管理器将该代理作为两个可执行进程列出。

- 本地 Windows 服务包装程序 `imqbrokersvc.exe`

- 正在运行代理的 Java 运行时

系统只能有一个作为 Windows 服务运行的代理。

重新配置代理服务

重新配置 Windows 服务的顺序如下：

1. 停止服务。
2. 删除服务。
3. 添加服务，并使用 `-args` 选项指定不同的代理启动选项，或使用 `-vmargs` 选项指定不同的 Java 版本参数。

使用替代 Java 运行时

可以使用 `-javahome` 或 `-jrehome` 选项指定替代 Java 运行时的位置。（也可以在 Windows 服务控制面板的“启动参数”字段中指定这些选项。）

“启动参数”字段将反斜杠 (\) 视为转义符，因此，将它作为路径分界符时必须键入两次，例如：

```
-javahome d:\\jdk1.3。
```

显示代理服务启动选项

要确定代理服务的启动选项，请使用 `imqsvcadm` 命令的 `query` 选项。

```
imqsvcadm query

Service iMQ_Broker is installed.
Display Name:iMQ_Broker
Start Type:Manual
Binary location:c:\Program Files\Sun Microsystems\
                Message Queue 3.5\bin\imqbrokersvc
JavaHome:c:\j2sdk1.4.0
Broker Args:-passfile d:\imqpassfile
```

服务启动问题疑难解答

如果试图启动服务时出现错误，您可以查看记录的错误事件。

► 查看记录的服务错误事件

1. 启动事件查看器。

2. 在“日志” > “应用程序”下查找。
3. 选择“查看” > “刷新”，以查看所有的错误事件。

删除作为 Windows 服务运行的代理

要删除作为服务运行的代理，请执行以下操作之一：

- 使用命令。先使用 `imqcmd shutdown bkr` 命令关闭代理，然后使用 `imqsvcadm remove` 命令删除服务。
- 对于 Windows 服务，可以使用控制面板中的管理工具。在不同版本的 Windows 中，此功能所在的位置不同。

完成后，重新启动计算机。

启动 Message Queue 客户机

启动客户机应用程序之前，请向应用程序开发者了解有关如何设置系统的信息。如果要启动 Java 客户机应用程序，必须设置 `CLASSPATH` 变量并确保已安装了正确的 Jar 文件。Message Queue Developer's Guide for Java Clients 包含有关系统设置常规步骤的信息，但开发者可能会提供额外信息。

要启动 Java 客户机应用程序，请使用下列命令行格式：

```
java clientAppName
```

要启动 C 客户机应用程序，请使用应用程序开发者提供的格式。

应用程序开发者或应用程序文档应提供有关应用程序设置的属性值的信息。您可能希望覆盖应用程序设置的某些属性。可以通过在命令行中指定这些属性来完成此操作。

您可能还希望在命令行中使用 JNDI 查找来查找其连接工厂的任何 Java 客户机指定属性。如果该查找返回的连接工厂比应用程序旧，则该连接工厂可能缺少对较新属性的支持。在这种情况下，Message Queue 将这些属性设置为默认值。通过在命令行中指定属性，可以将其设置为非默认值。

要在命令行中提供属性值，请对 Java 应用程序使用下列命令行语法：

```
java [[-Dattribute=value ]...] clientAppName
```

attribute 的值必须是连接工厂受管理对象属性，如第 16 章“受管理对象的属性参考”中所述；如果该值中包含空格，请在命令行的 *attribute=value* 部分的两端添加引号。

下列示例启动了客户机应用程序 MyMQClient。该应用程序连接的代理位于 OtherHost 主机（端口为 7677）中，它将覆盖应用程序所设置的任何主机名和端口。

```
java -DimqAddressList=mq://OherHost:7677/jms MyMQClient
```

在某些情况下，无法使用命令行来指定属性值。管理员可以将受管理对象设置为只允许读取访问，或者应用程序开发者也可以通过对客户机进行编码来实现此目的。与应用程序开发者进行沟通非常必要，这有助于了解启动客户机程序的最佳途径。

删除代理实例

本节包含有关在 Solaris 或 Linux 中删除代理实例的信息。有关删除 Windows 服务的信息，请参见第 68 页“删除作为 Windows 服务运行的代理”。

要删除代理实例，请使用 imqbrokerd 命令及 -remove 选项。用于删除代理实例的命令格式如下：

```
imqbrokerd [options..] -remove instance
```

例如，如果代理名为 myBroker，则此命令如下：

```
imqbrokerd -name myBroker -remove instance
```

此命令删除指定代理的整个实例目录。

有关可用来删除代理的选项列表，请参见第 253 页“命令参考”中的 imqbrokerd 参考信息。

在 Solaris 或 Linux 中，如果将代理设置为在系统启动时自动启动，请编辑配置文件 /etc/imq/imqbrokerd.conf (Solaris) 或 /etc/opt/sun/mq/imqbrokerd.conf (Linux)，并将 AUTOSTART 属性设置为 NO。

删除代理实例

配置代理

在代理实例启动时，其配置由一组配置文件和传送到 `imqbrokerd` 命令的选项管理。本章说明了配置文件和命令行选项如何交互以配置代理实例，介绍了每个代理组件的功能并列出了其配置属性，另外还说明了如何设置配置。

本章包含以下小节：

- 第 72 页 “关于可配置的代理组件”
- 第 90 页 “关于配置文件”
- 第 92 页 “编辑实例配置文件”
- 第 92 页 “在命令行中输入配置选项”
- 第 93 页 “设置持久性存储”
- 第 97 页 “确保持久性数据的安全”

有关配置属性的完整参考信息，请参见第 14 章 “代理属性参考”

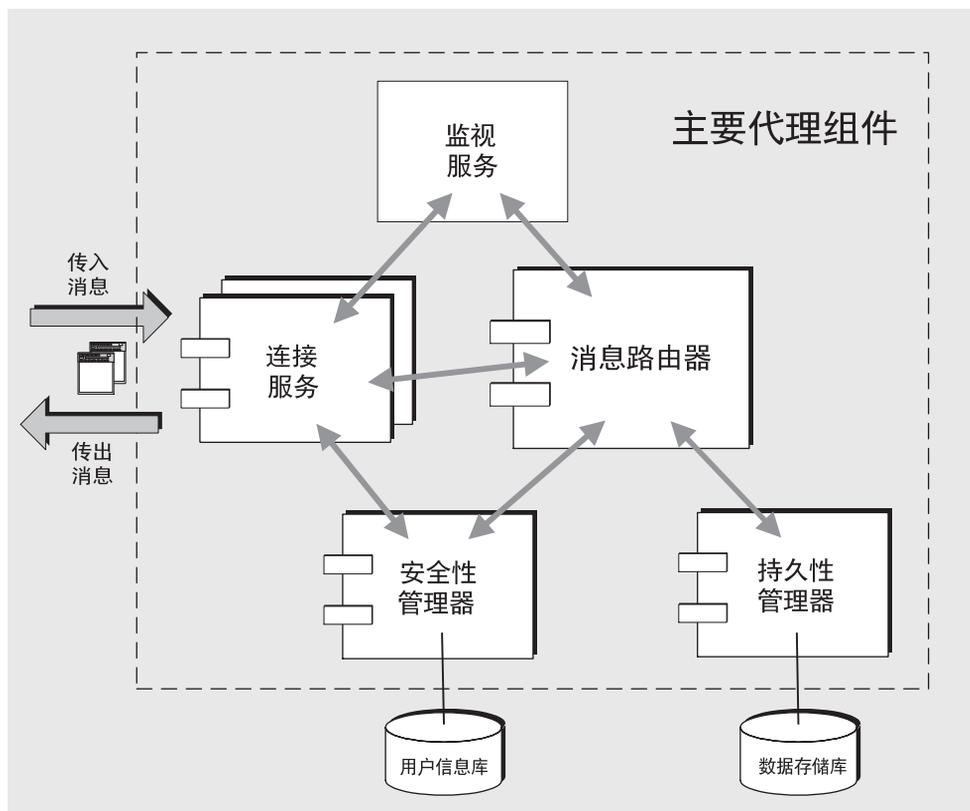
关于可配置的代理组件

在 Message Queue 消息传送系统中，消息传送（从生产方客户机到目标，然后再从目标到一个或多个使用方客户机）由代理或以串连模式工作的代理实例群集来执行。

要执行消息传送，代理必须建立与客户机之间的通信通道、执行验证和授权、适当的路由消息、确保可靠传送并提供用于监视系统性能的数据。

为了实现其功能，代理使用了大量的内部组件，每个组件都在传送过程中担当着各自不同的角色。图 4-1 中说明了这些代理组件。

图 4-1 代理服务组件



消息路由器组件执行关键的消息路由和传送服务，而其他组件则提供重要的支持服务。表 4-1 简要说明了各个组件。

表 4-1 主要代理服务组件及功能

| 组件 | 说明 / 功能 | 属性说明 ... |
|--------|---|---------------------|
| 连接服务 | 管理代理和客户机之间的物理连接，用于传输外来和外出消息。 | 第 283 页 “连接服务属性” |
| 消息路由器 | 管理消息的路由和传送，这里的消息包括 JMS 消息以及在 Message Queue 消息传送系统中用于支持 JMS 消息传送的控制消息。 | 第 285 页 “消息路由器属性” |
| 持久性管理器 | 管理向持久性存储器写入数据和从持久性存储器中检索数据。 | 第 287 页 “持久性管理器属性” |
| 安全性管理器 | 为请求连接到代理的用户提供验证服务，并为通过验证的用户提供授权服务（访问控制）。 | 第 291 页 “安全性管理器属性” |
| 监视服务 | 生成可写入管理员用于监视和管理代理的很多输出通道的度量和诊断信息。 | 第 295 页 “监视和日志记录属性” |

可以根据负荷条件、应用程序的复杂性等因素，配置这些组件以优化代理的性能。以下小节详细介绍了每个组件执行的功能以及可以设置以影响其行为的属性。

连接服务

Message Queue 代理支持与 Message Queue 应用程序客户机和 Message Queue 管理客户机的通信。各项连接服务是由其服务类型和协议类型指定的，如下所示：

- 服务类型指定服务提供的是 JMS 消息传送 (NORMAL) 服务，还是 Message Queue 管理 (ADMIN) 服务
- 协议类型指定支持服务的底层传输协议层。

表 4-2 列出了可从 Message Queue 代理获得的连接服务：

表 4-2 代理支持的连接服务

| 服务名称 | 服务类型 | 协议类型 |
|----------------|--------|--------------------|
| jms | NORMAL | tcp |
| ssljms (企业版) | NORMAL | tls (其安全性基于 SSL) |
| httpjms (企业版) | NORMAL | http |
| httpsjms (企业版) | NORMAL | https (其安全性基于 SSL) |

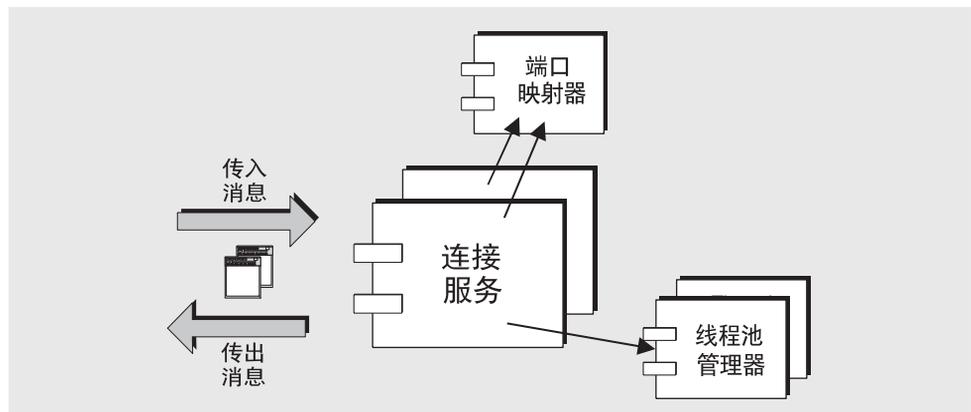
表 4-2 代理支持的连接服务（续）

| 服务名称 | 服务类型 | 协议类型 |
|----------|-------|-----------------|
| admin | ADMIN | tcp |
| ssladmin | ADMIN | tls（其安全性基于 SSL） |

可以将代理配置为运行以上任一或全部连接服务。每个连接服务仅在由代理的主机名和端口号指定的特定端口上可用。默认情况下，启用 `jms` 和 `admin` 服务。

Message Queue 可动态地将某项连接服务映射到某一端口号，或者也可以显式地分配一个端口。每项服务都具有一个线程池管理器，并向通用端口映射器服务注册它本身，如图 4-2 所示。

图 4-2 连接服务支持



以下部分介绍连接服务与端口映射器及线程池管理器之间的关系。

端口映射器

Message Queue 提供了一个端口映射器，用于将端口分配到连接服务。端口映射器位于标准端口号 7676 上。当客户机与代理建立连接时，它先与端口映射器通信，请求获得已指定连接服务的端口号。

`jms`、`ssljms`、`admin` 和 `ssladmin` 连接服务的端口号可以是动态的，也可以是静态的。默认情况下，连接服务在启动时动态地配置其端口。或者，也可以为该服务指定一个静态端口，但一般不建议使用静态端口号。静态端口号通常只用于一些特殊情况，例如经过防火墙的连接。

分别使用第 345 页表 C-1 和第 355 页表 C-3（这两个表格位于附录 C “HTTP/HTTPS 支持”）中说明的属性来配置 `httpjms` 和 `httpsjms` 服务。

线程池管理器

每个连接服务都是多线程的，从而支持多个连接。这些连接所需的线程在线程池中进行维护，而线程池又由线程池管理器组件来管理。

通过配置线程池管理器，可以设置线程池中维护的线程的最大和最小数量。当连接需要线程时，这些线程将被添加到线程池中。当线程数量超过最小值时，系统将关闭变为空闲状态的线程，直到达到最小阈值，以节省内存资源。此值应该足够大，这样就不必不断创建新线程了。如果连接负荷较重，线程数量可能会增加，直到达到线程池的最大数量。此后，连接必须一直等待，直到某个线程可用。

线程池中的线程可专用于单个连接（专用模型），也可根据需要分配到多个连接（共享模型）。

专用模型 每个到代理的连接需要两个专用线程：一个用于处理连接收到的消息，另一个用于处理连接发出的消息。这样，连接的数量就被限制为线程池中最大线程数量的一半，但却提供了较高的性能。

共享模型（企业版） 只要发送或接收消息，连接就会由共享线程处理。由于每个连接不需要专用线程，此模型增加了连接服务可支持的连接数量，因而，也增加了代理可支持的连接数量。但这同时带来了共享线程相关的某些性能开销。线程池管理器使用一组分配器线程来监视连接活动并根据需要将连接分配到线程。可通过限制每个此类分配器线程所监视的连接的数量使此活动带来的性能开销最小化。

安全性

每个连接服务均支持特定的验证和授权（访问控制）功能（请参见第 83 页“安全性管理器”）。

连接服务属性

以下是与连接服务相关的可配置属性：

- `imq.service.activelist`。在代理启动时将启动的连接服务的列表。
- `imq.hostname`。指定当有多个主机可用（例如，一台计算机中安装了多个网络接口卡）时所有连接服务将绑定到的主机。
- `imq.portmapper.port`。指定代理的主端口，即端口映射器所在的端口。
- `imq.portmapper.hostname`。指定当有多个主机可用时端口映射器将绑定到的主机。

- `imq.portmapper.backlog` 的值。指定端口映射器在拒绝请求之前可处理的并发请求的最大数量。该属性设置了可存储在操作系统待办事项中等待端口映射器处理的请求的数量。
- `imq.service_name.protocol_type.port`。仅适用于 `jms`、`ssljms`、`admin` 和 `ssladmin` 服务，指定命名的连接服务的端口号。
- `imq.service_name.protocol_type.hostname`。仅适用于 `jms`、`ssljms`、`admin` 和 `ssladmin` 服务，指定当有多个主机可用时命名的连接服务将绑定到的主机。
- `imq.service_name.min_threads`。指定线程数量，达到该数量后，线程池中将维护此数量的线程供命名的连接服务使用。
- `imq.service_name.max_threads`。指定线程数量，达到该数量后，新的线程将不会被添加到线程池中供命名的连接服务使用。
- `imq.service_name.threadpool_model`。指定线程是专用于特定的连接，还是根据命名的连接服务的需要由多个连接共享。
- `imq.shared.connectionMonitor_limit`。仅适用于共享线程池模型，指定分配器线程可监视的连接的最大数量。

有关这些属性的完整说明，请参见第 283 页表 14-2。

消息路由器

使用支持的连接服务在客户机与代理之间建立起连接后，即可进行消息的路由和传送。

基本传送机制

广义来讲，代理处理的消息可分为两类：

- 由生产方客户机发出和发往使用方客户机的 **JMS** 有效负荷消息
- 客户机发送或接收的用于支持 **JMS** 消息传送的控制消息

如果收到的消息为 **JMS** 消息，代理会根据其目标是队列还是主题，将其路由到使用方客户机：

- 如果目标为主题，**JMS** 消息将立即被路由到此主题的所有活动订户。如果长期订户处于非活动状态，消息路由器会一直保留消息，直到该订户变为活动状态，再传送消息。
- 如果目标为队列，**JMS** 消息将被放置在相应的队列中，并在到达队列前端时被传送到相应的使用方。消息到达队列前端的顺序取决于它们进入队列的顺序以及它们的优先级。

消息路由器将消息传送到其所有目标使用方后，将立即从内存中删除此消息。如果消息是持久性的，则消息路由器将其从代理的持久性数据存储库中删除。

可靠传送：确认和事务

如果增加了可靠传送要求，则刚才讨论的传送机制就会复杂得多。可靠传送涉及两个方面：

- 确保消息成功传送到代理以及从代理传送出去
- 确保在实际传送消息前，代理没有丢失消息或传送信息

为了确保消息成功传送到代理及从代理传送出去，`Message Queue` 使用了大量响应控制消息。

例如，当生产方向目标发送 `JMS` 消息（有效负荷消息）时，代理给出接收到 `JMS` 消息的响应。（默认情况下，只有当生产方将 `JMS` 消息指定为持久性消息时，`Message Queue` 才会进行上述操作。）生产方客户机使用代理响应来确保对目标的传送成功。

同样，当代理将 `JMS` 消息传送到使用方时，使用方客户机会发回一个确认，说明它已接收并处理了此消息。客户机可以在创建会话对象时指定发送这些确认的自动程度和频率，但消息路由器在收到每个消息使用方（它向其传送消息，例如某个主题的每个订户）的确认之前，不会从内存中删除 `JMS` 消息。

如果某个主题有多个长期订阅，消息路由器会将每条 `JMS` 消息保留在该目标中，以便在每个长期订户变为活动使用方时将消息传送出去。

消息路由器会在接收客户机确认时进行记录，并且只有在接收到所有确认后才会删除 `JMS` 消息，除非在这之前 `JMS` 消息已过期。

另外，消息路由器会将代理响应发送回客户机，确认已接收到客户机确认。使用方客户机使用代理响应来确保代理不会多次传送 `JMS` 消息。如果代理无法接收客户机确认，就可能会出现这种情况。

如果代理没有接收到客户机确认并再次传送了 `JMS` 消息，此消息将被标注“重新传送”标志。通常，代理在以下情况下将重新传送 `JMS` 消息：

- 在代理收到客户机确认前关闭客户机连接，且随后打开一个新的连接。
- 客户机应用程序恢复某一会话。
- 客户机应用程序恢复回滚的事务。

例如，如果队列的消息使用方在确认消息之前脱机，随后另一位使用方向队列进行了注册，则代理会向该新使用方重新传送未确认的消息。

上述客户机确认和代理响应同样适用于编组为事务的 JMS 消息传送。这种情况下，这些过程在事务级别以及各个 JMS 消息发送或接收级别运行。当提交事务时，将自动发送代理响应。

代理会跟踪事务，允许提交它们或在出现故障时回滚。该事务管理也支持本地事务（较大的分布式事务的一部分）。代理会一直跟踪这些事务的状态，直到它们被提交。当代理启动时，它会检查所有未提交的事务，在默认情况下，代理回滚未处于 PREPARED 状态的所有事务。如果设置了 `imq.transaction.autorollback` 属性，代理还会回滚处于 PREPARED 状态的事务。

可靠传送：持久性

可靠传送的另一个方面是确保在实际传送消息前，代理没有丢失消息或要传送的信息。通常，消息将始终保留在内存中，直到被传送或过期。但是，如果代理出现故障，这些消息就会丢失。

如果生产方客户机指定某一消息是持久性的，则消息路由器会将该消息传送到持久性管理器。持久性管理器将该消息存储到数据库或文件系统中（请参见第 80 页“持久性管理器”），以便在代理出现故障时能够恢复该消息。

管理内存资源和消息流

代理的性能和稳定性取决于可用的系统资源以及资源（如内存）的使用效率。特别是，当消息的生成大大快于使用时，消息路由器可能无法承受大量的消息，可能会用光其所有内存资源。为防止这种情况，消息路由器使用了三级内存保护以在资源不足时保持操作系统运转：

单个目标的消息限制 可以设置物理目标属性，指定消息数量和消息所使用的总内存限制（请参见第 15 章“物理目标属性参考”）。也可以指定达到限制时消息路由器的行为。四种限制行为是：

- 减慢消息生产方 (FLOW_CONTROL)
- 丢弃内存中最旧的消息 (REMOVE_OLDEST)
- 根据消息的生存期丢弃内存中优先级最低的消息 (REMOVE_LOW_PRIORITY)
- 拒绝最新的消息 (REJECT_NEWEST)

系统范围消息限制 系统范围消息限制构成了第二道防线。可指定应用于系统中所有目标的系统范围限制：消息总数和所有消息占用的内存（请参见第 285 页表 14-3）。如果达到了任何系统范围消息限制，消息路由器将拒绝新消息。

系统内存阈值 系统内存阈值是第三道防线。可指定可用系统内存的阈值，代理可利用阈值采取更为进一步的操作以防内存过载。采取的操作取决于内存资源的状态，如下所示：

- green（有大量的可用内存）
- yellow（代理内存正在降低）
- orange（代理内存低）
- red（代理内存不足）

随着代理的内存状态由 green 变为 yellow，再变为 orange，最后变为 red，代理所采取的措施也会越来越严格，操作类型如下：

- 将消息从活动内存交换到持久性存储器中（请参见第 80 页“持久性管理器”）。
- 限制非持久性消息的生产方，最终会停止进入代理的消息流。持久性消息流自动受以下要求的限制：代理确认每一条消息。

这些措施都会降低性能。

如果达到系统内存阈值，则目标消息限制和系统范围消息限制都太小。在某些情况下，阈值无法及时捕获所有潜在的内存过载。因此，不应依赖此功能来控制内存资源，而是分别配置目标，共同优化内存资源。

消息路由器属性

以下是用于管理内存资源的系统范围限制和系统内存阈值：

- `imq.destination.DMQ.truncateBody`。指定停用消息队列只包括消息的标题和属性数据。废弃消息主体内容。
- `imq.message.expiration.interval`。指定过期消息的回收频率（以秒为单位）。
- `imq.system.max_count`。指定代理保留的消息的最大数量。
- `imq.system.max_size`。指定代理保留的消息总大小的最大值。
- `imq.message.max_size`。指定消息主体大小的最大值。
- `imq.resource_state.threshold`。指定将触发各个内存资源状态的内存占用百分比。
- `imq.resource_state.count`。指定当触发各个内存资源状态时允许同时处理的一批外来消息的最大数量。
- `imq.transaction.autorollback`。指定启动代理时，是否自动回滚处于 PREPARED 状态的分布式事务。

有关这些属性的完整说明，请参见第 285 页表 14-3。

持久性管理器

对于发生故障后待恢复的代理，需要重新创建其消息传送操作的状态。这需要它将所有持久性消息以及基本的路由和传送信息保存到数据存储库中。持久性管理器组件用于管理此信息的写入和检索。

为了恢复出现故障的代理，不仅需要恢复未传送的消息，代理还必须完成以下操作：

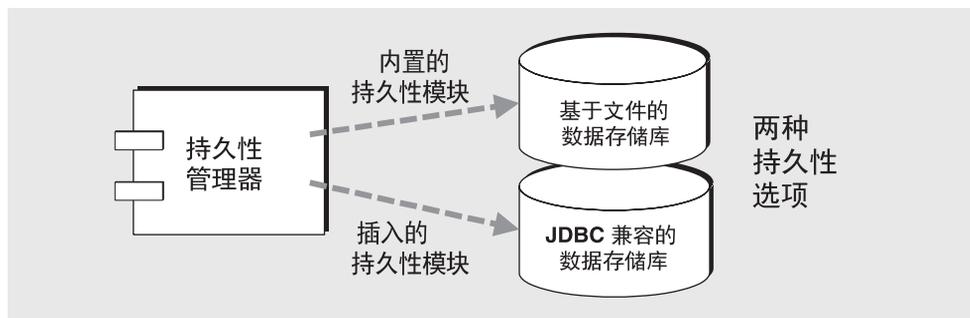
- 重新创建目标
- 恢复每个主题的长期订阅列表
- 恢复每条消息的确认列表
- 重新生成所有已提交事务的状态

持久性管理器用于管理所有这些状态信息的存储和检索。

当代理重新启动时，它将重新创建目标和长期订阅，恢复持久性消息和所有事务的状态，并重新创建未传送消息的路由表。然后代理才可以恢复消息传送。

Message Queue 既支持内置的持久性模块，也支持插入的持久性模块（请参见图 4-3）。内置的持久性是基于文件的数据存储库。插入的持久性使用 Java 数据库连接 (JDBC™) 接口，并需要 JDBC 数据存储库。通常，内置的持久性比插入的持久性速度更快，但某些用户更希望获得使用 JDBC 兼容的数据库系统的冗余和管理功能。

图 4-3 持久性管理器支持



内置的持久性

默认的 Message Queue 持久性存储器解决方案是基于文件的数据存储库。该解决方案使用单个文件来存储持久性数据，如消息、目标、长期订阅和事务等。

基于文件的数据存储库位于由与数据存储库相关联的代理实例的名称 (*instanceName*) 标识的目录内（请参见附录 A “Message Queue 数据在特定操作系统中的位置”）：

```
.../instances/instanceName/fs350/
```

可构建基于文件的数据存储库，以便持久性消息所在的目标将其保存在目录中。大多数消息存储在由大小可变的记录组成的单个文件中。

为减少添加和删除消息带来的碎片，可压缩大小可变的记录文件（请参见第 124 页“压缩物理目标”）。另外，内置的持久性管理器可在其各自的文件中存储大小超过配置阈值 (`imq.persist.file.message.max_record_size`) 的消息，而不是存储在大小可变的记录文件中。对于这些单个文件，可维护文件池以便重新使用这些文件。当不再需要某一消息文件时，并不会将其删除。而是将该消息文件添加到其目标目录下的空闲文件池中，用于存储新消息。

可配置目标文件池中文件的最大数量

(`imq.persist.file.destination.message.filepool.limit`)。也可以指定文件池中空闲文件所占的百分比，达到此百分比后将通过清空来清除文件池，而不是仅标记这些空闲文件供重复使用 (`imq.persist.file.message.filepool.cleanratio`)。随着清除文件的百分比增加，磁盘空间量逐渐减小，而维护文件池所需的开销却逐渐增加。

可以指定关闭时是否清除标记的文件 (`imq.persist.file.message.cleanup`)。如果清除这些文件，它们所占用的磁盘空间就会减少，但代理将需要更长的时间来进行关闭操作。

所有其他持久性数据（目标、长期订阅和事务）都存储在单独文件中：所有目标在一个文件中，所有长期订阅在另一个文件中，依此类推。

要最大限度地提高可靠性，可以使用 `imq.persist.file.sync.enabled` 属性指定持久性操作应使内存中状态与物理存储设备同步。这有助于消除因系统崩溃而导致的数据丢失，但代价是性能的下降。如果在 Sun Cluster 环境中运行 Message Queue，必须将群集中所有节点的该属性设置为 `true`。

因为数据存储库可能包含敏感或专用消息，所以应保护

`instances/instanceName/fs350/` 目录以防止未经授权的访问。有关说明，请参见第 97 页“确保持久性数据的安全”。

插入的持久性

您可以代理进行设置，以访问可通过 JDBC 驱动程序访问的任何数据存储库。这将设计到设置多个与 JDBC 相关的代理配置属性以及使用数据库管理器实用程序 (imqdbmgr) 来创建具有相应架构的数据存储库。第 93 页“设置持久性存储”中详细说明了上述过程和相关的配置属性。

持久性管理器属性

该属性指定所使用的持久性的类型：

- `imq.persist.store`。指定代理是使用内置的、基于文件的 (file) 持久性，还是使用插入的 JDBC 兼容 (jdbc) 持久性。

以下属性适合内置持久性：

- `imq.persist.file.sync.enabled`。指定持久性操作是否使内存中状态与物理存储设备同步。
- `imq.persist.file.message.max_record_size`。指定将要添加到消息存储文件中的消息大小的最大值。
- `imq.persist.file.destination.message.filepool.limit`。指定目标文件池中可供重复使用的空闲文件的最大数量。
- `imq.persist.file.message.filepool.cleanratio`。指定保持清除状态（清空）的目标文件池中空闲文件的百分比。
- `imq.persist.file.message.cleanup`。指定代理在关闭时是否清除目标文件池中的空闲文件。

有关这些属性的完整说明，请参见第 288 页表 14-6。

以下属性适合基于 JDBC 的持久性：

- `imq.persist.jdbc.brokerid`。指定附加到数据库（由多个代理实例使用）中表名后面的代理实例标识符。
- `imq.persist.jdbc.driver`。指定连接到数据库的 JDBC 驱动程序的 Java 类名。
- `imq.persist.jdbc.opendburl`。指定打开现有数据库连接的数据库 URL。
- `imq.persist.jdbc.createdburl`。指定打开创建数据库连接的数据库 URL。
- `imq.persist.jdbc.closedburl`。指定关闭代理时关闭当前数据库连接的数据库 URL。
- `imq.persist.jdbc.user`。指定打开数据库连接的用户名（如果需要）。

- `imq.persist.jdbc.needpassword`。指定数据库是否要求输入为访问代理的密码。
- `imq.persist.jdbc.password`。指定用于打开数据库连接的密码（如果需要）。
- `imq.persist.jdbc.table.IMQSV35`。用于创建版本表的 SQL 命令。
- `imq.persist.jdbc.table.IMQCCREC35`。用于创建配置更改记录表的 SQL 命令。
- `imq.persist.jdbc.table.IMQDEST35`。用于创建目标表的 SQL 命令。
- `imq.persist.jdbc.table.IMQINT35`。用于创建 **Interest** 表的 SQL 命令。
- `imq.persist.jdbc.table.IMQMSG35`。用于创建消息表的 SQL 命令。
- `imq.persist.jdbc.table.IMQPROPS35`。用于创建属性表的 SQL 命令。
- `imq.persist.jdbc.table.IMQILIST35`。用于创建 **Interest** 状态表的 SQL 命令。
- `imq.persist.jdbc.table.IMQTXN35`。用于创建事务表的 SQL 命令。
- `imq.persist.jdbc.table.IMQTACK35`。用于创建事务确认表的 SQL 命令。

有关这些属性的完整说明，请参见第 289 页表 14-7。

安全性管理器

Message Queue 提供验证和授权（访问控制）功能，同时也支持加密功能。

验证和授权功能取决于用户系统信息库（请参见第 84 页图 4-4）：包含消息传送系统的用户信息（用户名、密码和组成员资格）的文件、目录或数据库。用户名和密码用于在请求连接至代理时对用户进行验证。用户名和组成员资格（与访问控制文件配套使用）用于授权操作，例如为目标生成消息或使用目标的消息。

Message Queue 管理员可填充 Message Queue 提供的用户系统信息库（请参见第 130 页“使用平面文件用户系统信息库”），也可以将原有的 LDAP 用户系统信息库插入到安全性管理器组件中（请参见第 135 页“使用 LDAP 服务器管理用户系统信息库”）。

验证

Message Queue 安全性支持基于密码的验证。当客户机请求连接到代理时，该客户机必须提交用户名和密码。

安全性管理器将把客户机提交的用户名和密码与用户系统信息库中存储的用户名和密码进行比较。密码在从客户机传送到代理的过程中，将使用 Base 64 编码或消息摘要 (MD5) 进行编码。要想获得更安全的传送，请参见第 85 页“加密”。可以分别配置每个连接服务使用的编码的类型，也可以基于代理范围设置编码方式。

第 85 页“安全性管理器属性”中列出了所有安全管理器属性，并在第 85 页“安全性管理器属性”中进行了详细说明。

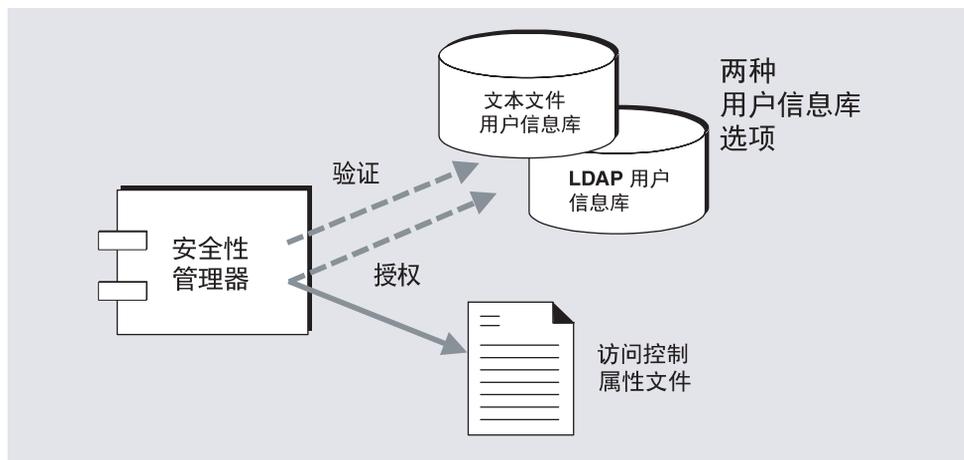
授权

客户机应用程序的用户通过验证后，即可以获得授权来执行各种 Message Queue 相关的活动。安全性管理器既支持基于用户的访问控制，也支持基于组的访问控制。根据用户在用户系统信息库中分配到的用户名或组的不同，该用户所拥有的执行特定 Message Queue 操作的权限也不同。这些访问控制在访问控制属性文件中指定（请参见图 4-4）。

当用户尝试执行某个操作时，安全性管理器将检查该用户的用户名和组成员资格（通过用户系统信息库）是否属于被指定允许访问该操作的用户名或组成员资格（在访问控制属性文件中）。访问控制属性文件指定了执行以下操作的权限：

- 建立与代理的连接
- 访问目标：创建任意给定目标或所有目标的使用方、生产方或队列浏览器
- 自动创建目标

图 4-4 安全性管理器支持



默认的控制访问属性文件只明确引用一个组：管理员（请参见第 133 页“组”）。管理员组中的用户拥有管理服务连接权限。管理服务使用户可以执行管理功能，如创建目标以及监视和控制代理。默认情况下，您定义的其他任何组中的用户均无法获得管理服务连接。

作为 Message Queue 管理员，可以在用户系统信息库中定义组并使用户与这些组关联（虽然在平面文件用户系统信息库中不完全支持组）。

通过编辑访问控制属性文件，可以按照用户和组的不同目的（生成和使用消息，或浏览队列目标中的消息）指定对目标的权限。可以分别指定各个目标或指定所有目标仅能够被特定用户或组访问。如果将代理配置为允许自动创建目标，可编辑访问控制属性文件来控制代理可自动为其创建目标的用户和组。

第 85 页“安全性管理器属性”中列出了所有安全管理器属性，并在第 85 页“安全性管理器属性”中进行了详细说明。

加密

要对客户机和代理之间发送的消息进行加密，需要使用基于安全套接字层 (SSL) 标准的连接服务。SSL 通过在启用 SSL 的代理与启用 SSL 的客户机之间建立加密连接来提供连接级别的安全性。

为了使用 Message Queue 基于 SSL 的连接服务，需要使用密钥工具实用程序 (imqkeytool) 生成专用密钥 / 公用密钥对。此实用程序将公用密钥嵌入自签名的证书中，并将其放入 Message Queue 密钥库中。Message Queue 密钥库本身设有密码保护，要解除锁定，必须在启动时提供密钥库的密码。请参见第 144 页“使用基于 SSL 的服务”。

解除密钥库的锁定后，代理即可将证书传送给所有请求连接的客户机。然后，客户机使用此证书建立与代理的加密连接。

下一节中列出了所有安全管理器属性，并在第 85 页“安全性管理器属性”中进行了详细说明。

安全性管理器属性

以下是用于验证、授权、加密以及其他安全性通信的可配置属性。

- `imq.authentication.type`。指定传送密码时应使用 Base 64 编码 (basic) 还是 MD5 摘要 (digest)。
- `imq.service_name.authentication.type`。指定传送密码时应使用 Base 64 编码 (basic) 还是 MD5 摘要 (digest)。
- `imq.authentication.basic.user_repository`。对于 Base 64 编码，指定用于验证的用户系统信息库的类型（基于文件或 LDAP）。

- `imq.authentication.client.response.timeout`。指定系统等待客户机响应来自代理的验证请求的时间（以秒为单位）。
- `imq.accesscontrol.enabled`。表明系统是否要检查已验证的用户是否如访问控制属性文件所指定，拥有使用连接服务或执行与特定目标有关的特定 Message Queue 操作的权限。
- `imq.service_name.accesscontrol.enabled`。为命名的连接服务设置访问控制 (`true/false`)，此设置将覆盖代理范围的设置。
- `imq.accesscontrol.file.filename`。为代理实例支持的所有连接服务指定访问控制属性文件的名称。
- `imq.service_name.accesscontrol.file.filename`。为代理实例的命名连接服务指定访问控制属性文件的名称。
- `imq.passfile.enabled`。指定是否在文件中指定用于安全性通信的用户密码（适用于 SSL、LDAP 和 JDBC™）。
- `imq.passfile.dirpath`。指定包含密码文件的目录的路径。
- `imq.passfile.name`。指定密码文件的名称。
- `imq.keystore.property_name`。适用于基于 SSL 的服务：指定与 SSL 密钥库相关的安全性属性。请参见第 295 页表 14-9。

有关这些属性的完整说明，请参见第 291 页表 14-8。

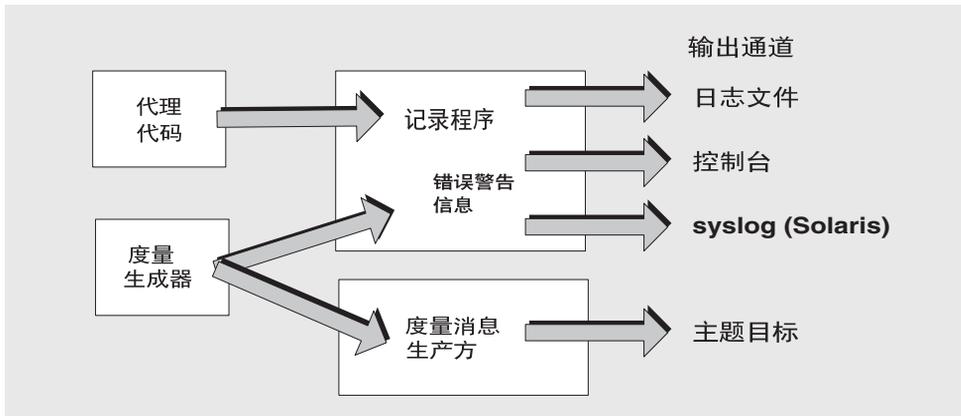
监视服务

代理包含大量用于监视和诊断其操作的组件。包括：

- 生成数据的组件（记录事件的代理代码和度量生成器）
- 通过多种输出通道记录输出信息的记录程序组件（请参见“记录程序”）
- 将包含度量信息的 JMS 消息发送到主题目标供 JMS 监视客户机使用的消息生产方。

图 4-5 中是通用方案的图解。

图 4-5 监视服务支持



度量生成器

度量生成器提供有关代理活动的信息，如流入流出代理的消息、代理内存中的消息数量及其使用的内存量、打开的连接数量和正在使用的线程数量。

可以打开或关闭度量数据的生成，并指定生成度量报告的频率。

记录程序

Message Queue 记录程序记录代理代码和度量生成器生成的信息，并将该信息写入多个输出通道中：标准输出（控制台）、日志文件以及 Solaris™ 操作系统上的 syslog 守护程序进程。

您可以指定记录程序所收集的信息类型以及写到每个输出通道的类型。

例如，可指定记录程序级别以确定记录程序收集的信息类型：错误 (ERROR)；错误和警告 (WARNING)；或错误、警告和信息 (INFO)。

可以分别指定写入到每个输出通道的记录程序的种类集。例如，如果将记录程序级别设置为 INFO，则可以指定只将错误和警告写入控制台，只将信息（度量数据）写入日志文件。

如果使用日志文件，可以指定何时关闭日志文件并将输出转移到新文件。创建新的转移日志文件时，将保留最新的 9 个日志文件的归档文件。

有关配置记录程序的信息，请参见第 187 页“配置和使用代理日志记录”。有关配置和使用 Solaris syslog 的信息，请参见 syslog(1M)、syslog.conf(4) 和 syslog(3C) 手册页。

度量消息生产方（企业版）

消息生产方组件按一定的时间间隔接收度量生成器组件的信息。并将该信息写入消息，然后将其发送到度量主题目标。而度量消息发送到的目标取决于它所包含的信息类型。

共有五个度量主题目标，它们的名称以及传送到各个目标的度量消息的类型显示在表 4-3 中。

表 4-3 度量主题目标

| 主题目标名称 | 度量消息类型 |
|--|----------------|
| mq.metrics.broker | 代理度量 |
| mq.metrics.jvm | Java 虚拟机度量 |
| mq.metrics.destination_list | 目标及其类型的列表 |
| mq.metrics.destination.queue. <i>monitoredDestinationName</i> | 具有指定名称的队列的目标度量 |
| mq.metrics.destination.topic. <i>monitoredDestinationName</i> | 具有指定名称的主题的目标度量 |

订阅这些度量主题目标的 Message Queue 客户机使用目标中的消息，并处理这些度量信息。例如，客户机可订阅 mq.metrics.broker 目标以接收和处理诸如代理中消息总数的信息。

度量消息生产方是一个内部 Message Queue 客户机，可创建包含与度量数据相对应的名称值对的消息（类型为 MapMessage）。仅当相应的度量主题目标有一个或多个订户时，才生成这些消息。

度量消息生产方生成的消息属于 MapMessage 类型。它们由许多名称 / 值对组成，取决于它们所包含的度量类型。每个名称 / 值对与度量数量及其值相对应。

例如，代理度量消息包含流入和流出代理的消息数量的值、这些消息的大小以及当前在内存中的消息数量和大小，等等。有关每种类型的度量消息中报告的度量数量的详细信息，请参见 Message Queue Developer's Guide for Java Clients。该手册介绍了如何向 Message Queue 客户机写入使用度量消息。

除了包含在度量消息主体中的度量信息之外，每条消息的标题还包含提供以下信息的属性：

- 消息类型
- 发送消息的代理的主机、端口和地址

- 度量示例被取样的时间

以下属性对于处理不同类型（或来自不同代理）的度量消息的 Message Queue 客户机应用程序非常有用。

监视服务属性

以下是用于通过代理设置信息的生成、记录和度量消息生成的可配置属性。

- `imq.metrics.enabled`。指定是否要将度量信息写入记录程序。
- `imq.metrics.interval`。如果启用了度量日志记录，指定将度量信息写入记录程序的时间间隔（以秒为单位）。
- `imq.log.level`。指定记录程序级别：可以写入到输出通道的输出的种类。
- `imq.log.file.output`。指定写入到日志文件的日志记录信息的种类。
- `imq.log.file.dirpath`。指定包含日志文件的目录的路径。
- `imq.log.file.filename`。指定日志文件的名称。
- `imq.log.file.rolloverbytes`。指定日志文件的大小（以字节为单位），达到该大小后输出将转移到新的日志文件。
- `imq.log.file.rolloversecs`。指定日志文件的生存期（以秒为单位），达到该值后输出将转移到新的日志文件。
- `imq.log.console.output`。指定将哪些种类的日志记录信息写入到控制台。
- `imq.log.console.stream`。指定是否将控制台输出写入到标准输出 (OUT) 或标准错误 (ERR)。
- `imq.log.syslog.facility`。（仅对于 Solaris）指定 Message Queue 代理应将 syslog 记录为哪种系统日志工具。
- `imq.log.syslog.logpid`。（仅对于 Solaris）指定是否将代理进程 ID 与消息一起记录。
- `imq.log.syslog.logconsole`。（仅对于 Solaris）指定如果无法将消息发送到 syslog，是否将其写入到系统控制台。
- `imq.log.syslog.identity`。（仅对于 Solaris）此属性指定的标识字符串应添加到每个记录到 syslog 的消息的前面。
- `imq.log.syslog.output`。（仅对于 Solaris）指定将哪些种类的日志记录信息写入到 `syslogd(1M)`。
- `imq.log.timezone`。指定日志时间戳的时区。
- `imq.metrics.topic.enabled`。指定是否启用度量消息生成。

- `img.metrics.topic.interval`。指定生成度量消息的时间间隔（以秒为单位）。
- `img.metrics.topic.persist`。指定度量消息是否为持久性的。
- `img.metrics.topic.timetolive`。指定生成度量消息（发送到度量主题目标）的生命周期（以秒为单位）。
- `img.destination.logDeadMsgs`。指定代理是否在每次废弃一条停用消息或将停用消息放入停用消息队列中时，都在日志中写入一条消息。

有关这些属性的完整参考信息，请参见第 295 页表 14-10。

关于配置文件

代理配置文件用于配置代理。附录 A “Message Queue 数据在特定操作系统中的位置” 中列出了这些配置文件在操作系统中所在的目录。

该目录存储以下文件：

- **启动时装入的默认配置文件**。文件名是 `default.properties`。该文件不可编辑。您可以读取该文件来确定默认设置以及查找要更改的属性的确切名称。
- **安装配置文件**，其中包含安装 Message Queue 时指定的所有属性。文件名是 `install.properties`。安装完成后该文件不可编辑。

实例配置文件

第一次运行代理时，将创建一个实例配置文件。使用实例配置文件为该代理实例指定配置属性。

实例配置文件存储在一个目录中，该目录用与此配置文件相关联的代理实例的名称 (*instanceName*) 标识：

```
.../instances/instanceName/props/config.properties
```

有关 `instances` 目录的位置，请参见附录 A “Message Queue 数据在特定操作系统中的位置”。

注 `.../instances/instanceName` 目录和实例配置文件由创建相应代理实例的用户拥有。代理实例必须始终由同一用户重新启动。

实例配置文件由代理实例维护。使用管理工具更改配置时，此文件也将被修改。您还可以手动编辑实例配置文件来更改配置（请参见第 92 页“编辑实例配置文件”）。要完成此操作，您必须是 `.../instances/instanceName` 目录的拥有者或以超级用户身份登录，更改目录特权。

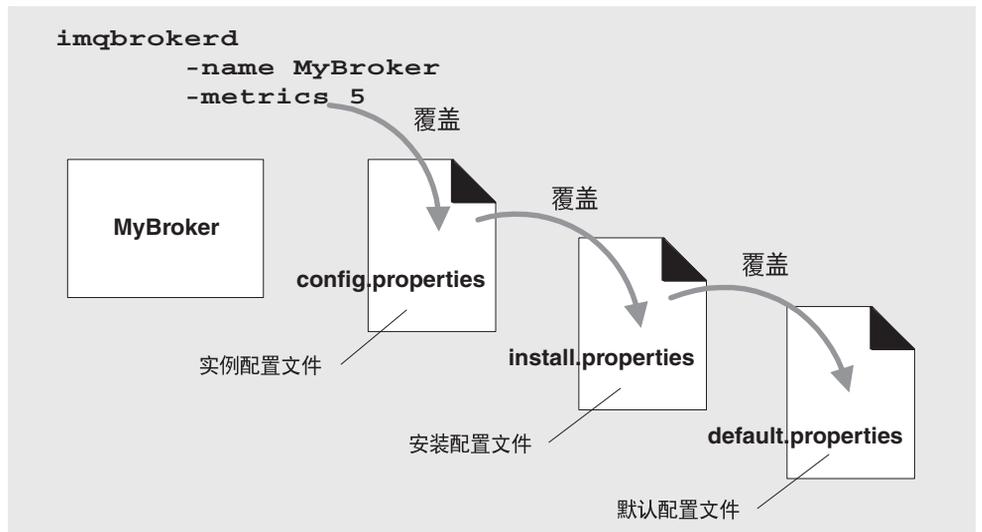
如果连接群集中的代理实例，则可能还需要使用群集配置文件来指定群集配置信息。有关详细信息，请参见第 298 页“群集配置属性”。

合并属性值

启动时，代理会合并不同配置文件中的属性值。它将使用安装配置文件和实例配置文件中的值来覆盖默认配置文件中指定的值。

您可以用 `imqbrokerd` 命令选项来覆盖得到的值。图 4-6 所示为以上方案的图解。

图 4-6 代理配置文件



属性命名语法

配置文件中的所有 Message Queue 属性定义均使用以下命名语法：

```
propertyName=value [ [, value1] ...]
```

例如，下面的条目指定：在拒绝其他消息前，代理可以在内存和持久性存储器中最多保留 50,000 条消息：

```
imq.system.max_count=50000
```

下面的条目指定：将每天（86400 秒）创建一个新日志文件：

```
imq.log.file.rolloversecs=86400
```

第 279 页第 14 章“代理属性参考”中列出了代理配置属性及其默认值。

编辑实例配置文件

首次运行代理实例时，系统将自动创建一个 config.properties 文件。您可以编辑该实例配置文件，以自定义相应代理实例的行为和资源使用。

代理实例仅在启动时读取 config.properties 文件。要对 config.properties 文件执行永久性更改，可执行以下任一操作：

- 使用管理工具。关于可以使用 imqcmd 设置的属性的信息，请参见第 280 页表 14-1。
- 在代理实例关闭时编辑 config.properties 文件，然后重新启动实例。（在 Solaris 和 Linux 操作系统中，只有第一个启动代理实例的用户具有编辑 config.properties 文件的权限。）

表 14-1 按字母顺序列出了代理配置属性及其默认值。关于各个属性的含义和用法的更多信息，请参考指定的交叉参考的小节。

在命令行中输入配置选项

在启动代理时（或之后），可以在命令行中输入代理配置选项。

启动时，请使用 imqbrokerd 命令启动代理实例。通过使用该命令的 -D 选项，可以指定任何代理配置属性及其值。如果使用 imqsvcadm 命令将代理作为 Windows 服务启动，请使用 -args 选项指定启动配置属性。

代理实例运行时，也可以设置某些代理属性。要修改正在运行的代理的配置，请使用 `imqcmd update bkr` 命令。

有关启动配置的详细信息，请参见第 3 章“启动代理和客户机”，尤其是第 64 页“以交互方式启动代理”中的示例。

有关修改正在运行的代理的配置的信息，请参见第 5 章“管理代理”和第 14 章“代理属性参考”

设置持久性存储

Message Queue 代理包含一个持久性管理器组件，该组件用于管理持久性信息的写入和检索。默认情况下，持久性管理器配置为访问基于文件的内置数据存储库，但是可以重新对其进行配置，以插入任何可通过 JDBC 兼容的驱动程序进行访问的数据存储库。

Message Queue 数据存储库中包含有关事务、消息、长期订阅和物理目标的信息。它还包含与确认相关的消息状态的信息。

本章介绍了如何设置代理以使用持久性存储。包括下列主题：

- 第 93 页“配置文件系统存储”
- 第 94 页“配置 JDBC 存储”
- 第 97 页“确保持久性数据的安全”

配置文件系统存储

创建代理实例时，将自动创建文件系统数据存储库。此存储位于该代理的实例目录中。该位置因操作系统而异；有关持久性存储具体位置的信息，请参见附录 A“Message Queue 数据在特定操作系统中的位置”。

默认情况下，Message Queue 执行非同步的磁盘写入操作。操作系统可以缓冲这些操作以提供良好的性能。但是，如果在两次写入操作之间出现意外的系统故障，则消息可能会丢失。要提高可靠性，可使 Message Queue 执行同步的磁盘写入操作。但要注意，此选项会导致性能降低。要指定同步的磁盘写入，请设置代理属性 `imq.persist.file.sync`。有关该属性的详细信息，请参见第 288 页表 14-6。

启动代理实例时，可以使用 `imqbrokerd -reset` 选项清除文件系统存储。有关该选项及其子选项的详细信息，请参见第 256 页表 13-2。

配置 JDBC 存储

要配置代理以使用基于 JDBC 的持久性，请在代理实例配置文件中设置与 JDBC 相关的属性，并创建相应的数据库模式。Message Queue 数据库管理器实用程序 (imqdbmgr) 使用 JDBC 驱动程序和代理配置属性来创建和管理数据库。

本章中介绍的步骤以 Java 2 Platform Enterprise Edition (J2EE) SDK 附带的 PointBase DBMS 为例进行说明。1.4 版本可以从 java.sun.com 下载。该示例使用 PointBase 的嵌入式版本（而不是客户机 / 服务器版本）。这些步骤以 PointBase 示例中的路径名和属性名为例进行说明。它们标有“示例：”字样。

Oracle 和 PointBase 的示例配置已可用。要查找这些示例文件，请参见附录 A “Message Queue 数据在特定操作系统中的位置”。在列出操作系统信息的表中，查找“示例应用程序和配置”的位置。

另外，PointBase 嵌入式版本、PointBase 服务器版本和 Oracle 的示例作为实例配置文件 config.properties 中的注释值提供。

插入 JDBC 可访问的数据存储库

插入 JDBC 可访问的数据存储库只需几个步骤。

► 插入 JDBC 可访问的数据存储库

1. 在代理的配置文件中设置 JDBC 相关属性。

请参见第 289 页“基于 JDBC 的持久性”中列出的属性文档。

2. 在以下路径中放置一个 JDBC 驱动程序 jar 文件的副本或符号链接：

`/usr/share/lib/imq/ext/ (Solaris)`

`/opt/sun/mq/share/lib/ (Linux)`

`IMQ_VARHOME\lib\ext (Windows)`

副本示例 (Solaris):

```
% cp j2eeSDK_install_directory/pointbase/lib/pointbase.jar
/usr/share/lib/imq/ext
```

符号链接示例 (Solaris):

```
% ln -s j2eeSDK_install_directory/lib/pointbase/pointbase.jar
/usr/share/lib/imq/ext
```

3. 创建 Message Queue 持久性所需的数据库模式。

使用 `imqdbmgr create all` 命令（对于嵌入式数据库）或者 `imqdbmgr create tbl` 命令（对于外部数据库）。请参见第 97 页“数据库管理器实用程序 (`imqdbmgr`)”。

示例：

- a. 转到 `imqdbmgr` 所在的目录。

```
cd /usr/bin (Solaris)
```

```
cd /opt/sun/mq/bin (Linux)
```

```
cd IMQ_HOME/bin (Windows)
```

- b. 输入 `imqdbmgr` 命令。

```
imqdbmgr create all
```

注 如果使用嵌入式数据库，最好在以下目录中创建它：

```
.../instances/instanceName/dbstore/databatbaseName。
```

如果嵌入式数据库没有设置用户名和密码保护，则可能设置了文件系统权限保护。要确保代理能够对数据库进行读写操作，运行该代理的用户应该是使用 `imqdbmgr` 命令创建该嵌入式数据库的同一个用户（请参见第 97 页“数据库管理器实用程序 (`imqdbmgr`)”）。

JDBC 相关代理属性

代理的实例配置文件位于一个由代理实例名称所标识的目录中，而该实例名称与此配置文件是相关联的（请参见附录 A “Message Queue 数据在特定操作系统中的位置”）：

```
.../instances/instanceName/props/config.properties
```

如果该文件尚不存在，必须使用 `-name instanceName` 选项启动代理，以便 Message Queue 创建该文件。

第 289 页“基于 JDBC 的持久性”列出了插入 JDBC 可访问的数据存储库时需要设置的配置属性。在本节内容末尾对这些属性进行了汇总。在使用插入持久性的每个代理实例的实例配置文件 (`config.properties`) 中设置这些属性。

利用实例配置属性可以自定义用于创建 Message Queue 数据库架构的 SQL 代码：有一个可配置的属性可指定用于创建每个数据库表的 SQL 代码。要正确指定插入的数据库使用的数据类型，必须使用这些属性。

由于数据库供应商之间在具体的 SQL 语法方面存在不兼容问题，请务必检查数据库供应商提供的相应文档并对第 289 页表 14-7 中的属性进行相应调整。例如，对于 PointBase 数据库，可能需要调整 IMQMSG35 表中 MSG 列所允许的最大长度（请参见 `imq.persist.jdbc.table.IMQMSG35` 属性）。

所有代理配置属性的值都可以使用 `-D` 命令行选项进行设置。如果某个数据库要求设置某些特定的数据库属性，也可以在启动代理 (`imqbrokerd`) 时使用 `-D` 命令行选项进行设置，或者使用数据库管理器实用程序 (`imqdbmgr`) 进行设置。

示例：

对于 PointBase 嵌入式数据库示例，可以使用 `-D` 命令行选项定义 PointBase 系统目录，而不是在数据库连接 URL 中指定数据库的绝对路径：

```
-Ddatabase.home=IMQ_VARHOME/instances/instanceName/dbstore
```

在这种情况下，可以指定 URL 以创建数据库，如下所示：

```
imq.persist.jdbc.createdburl=jdbc:pointbase:embedded:dbName;new
```

可以指定 URL 以打开数据库，如下所示：

```
imq.persist.jdbc.opendburl=jdbc:pointbase:embedded:dbName
```

下面是与 JDBC 相关的属性的汇总：

- `imq.persist.store`。指定基于文件或基于 JDBC 的数据存储库。
- `imq.persist.jdbc.brokerid`。指定附加到数据库表名后面的代理实例标识符，以使其具有唯一性。
- `imq.persist.jdbc.driver`。指定连接到数据库的 JDBC 驱动程序的 Java 类名。
- `imq.persist.jdbc.opendburl`。指定打开现有数据库连接的数据库 URL。
- `imq.persist.jdbc.createdburl`。指定打开创建数据库连接的数据库 URL。
- `imq.persist.jdbc.closedburl`。指定关闭代理时关闭当前数据库连接的数据库 URL。
- `imq.persist.jdbc.user`。指定打开数据库连接的用户名（如果需要）。
- `imq.persist.jdbc.needpassword`。指定数据库是否要求输入为访问代理的密码。
- `imq.persist.jdbc.password`。指定用于打开数据库连接的密码（如果需要）。
- `imq.persist.jdbc.table.IMQSV35`。用于创建版本表的 SQL 命令。
- `imq.persist.jdbc.table.IMQCCREC35`。用于创建配置更改记录表的 SQL 命令。

- `imq.persist.jdbc.table.IMQDEST35`。用于创建目标表的 SQL 命令。
- `imq.persist.jdbc.table.IMQINT35`。用于创建 **Interest** 表的 SQL 命令。
- `imq.persist.jdbc.table.IMQMSG35`。用于创建消息表的 SQL 命令。
- `imq.persist.jdbc.table.IMQPROPS35`。用于创建属性表的 SQL 命令。
- `imq.persist.jdbc.table.IMQILIST35`。用于创建 **Interest** 状态表的 SQL 命令。
- `imq.persist.jdbc.table.IMQTXN35`。用于创建事务表的 SQL 命令。
- `imq.persist.jdbc.table.IMQTACK35`。用于创建事务确认表的 SQL 命令。

有关这些属性的完整参考信息，请参见第 14 章“代理属性参考”

数据库管理器实用程序 (imqdbmgr)

Message Queue 为设置持久性所需的模式提供了数据库管理器实用程序 (imqdbmgr)。如果 Message Queue 数据库表被损坏或者您希望使用不同的数据库作为数据存储库，也可以使用该实用程序删除此数据库表。

有关 imqdbmgr 命令的语法、子命令和选项的参考信息，请参见第 13 章“命令参考”。

确保持久性数据的安全

持久存储可以包含与其他信息一起临时存储的消息文件。由于这些消息可能包含专用信息，所以保护数据存储库以防止未授权的访问至关重要。本节介绍如何保护内置文件存储或 JDBC 存储中的数据。

内置（基于文件的）持久性存储

使用内置持久性的代理将持久性数据写入平面文件数据存储库，其位置因操作系统而异（请参见附录 A“Message Queue 数据在特定操作系统中的位置”）：

```
.../instances/instanceName/fs350/
```

其中，`instanceName` 为标识代理实例的名称。

第一次启动代理实例时，将创建 `instanceName/filestore/` 目录。保护此目录的过程取决于运行代理的操作系统。

Solaris 和 Linux `IMQ_VARHOME/instances/instanceName/filestore/` 目录的访问权限取决于启动代理实例的用户的 `umask`。因此，可以通过适当设置 `umask` 来限制获取启动代理实例并读取其持久性文件的权限。或者，管理员（超级用户）可以通过将 `IMQ_VARHOME/instances` 目录上的权限设为 700 来保护持久性数据。

Windows 可以通过您使用的 Windows 操作系统提供的机制来设置 `IMQ_VARHOME/instances/instanceName/filestore/` 目录的访问权限。这通常涉及打开目录的属性对话框。

插入的 (JDBC) 持久性存储

使用插入的持久性的代理向 JDBC 兼容数据库写入持久性数据。

对于由数据库服务器管理的数据库（例如，Oracle 数据库），建议创建一个用户名和密码来访问 Message Queue 数据库表（名称以“IMQ”开头的表）。如果数据库不允许保护单个表，请创建一个仅由 Message Queue 代理使用的专用数据库。请参见数据库供应商提供的文档，以了解如何创建用户名 / 密码访问。

代理打开数据库连接所需的用户名和密码可以作为代理配置属性提供。但是，启动代理时将它们作为命令行选项更为安全（请参见 Message Queue 管理指南的附录 A “设置插入的持久性”）。

对于代理通过数据库的 JDBC™ 驱动程序直接访问的嵌入式数据库，通常通过在要存储持久性数据的目录上设置文件权限来提供安全性，如“[内置（基于文件的）持久性存储](#)”中所述。但是，为确保代理和 `imqdbmgr` 实用程序均可读写数据库，两者应由同一用户运行。

管理代理

本章介绍如何执行与管理代理及其服务相关的基本任务。本章包含以下小节：

- 第 100 页 “前提条件”
- 第 100 页 “使用 `imqcmd` 命令实用程序”
- 第 102 页 “显示帮助”
- 第 102 页 “显示产品版本”
- 第 102 页 “显示代理信息”
- 第 103 页 “更新代理属性”
- 第 104 页 “暂停和恢复代理”
- 第 105 页 “关闭并重新启动代理”
- 第 106 页 “显示代理度量”
- 第 107 页 “管理连接服务”
- 第 111 页 “获得有关连接的信息”
- 第 112 页 “管理长期订阅”
- 第 113 页 “管理事务”

本章并未涵盖与管理代理相关的全部主题。其他较大的主题将在后面的单独章节中进行介绍：

- 管理代理上的物理目标。有关如何创建、显示、更新和销毁物理目标，以及如何使用停用消息队列等主题的信息，请参见第 6 章 “管理物理目标”
- 设置代理的安全性。有关用户验证、访问控制、加密、密码文件和审计日志等主题的信息，请参见第 7 章 “管理安全性”

前提条件

使用 `imqcmd` 和 `imqusermgr` 命令来管理代理。在管理代理前，必须执行以下操作：

- 使用 `imqbrokerd` 命令启动代理。在代理运行前，不能使用其他命令。
- 确定要设置 Message Queue 管理用户，还是使用默认帐户。要使用管理命令，必须指定用户名和密码。

安装 Message Queue 时，系统会安装默认的用户系统信息库，这是一个平面文件。系统信息库发行时包括两个默认条目：管理员用户和 `guest` 用户。如果要测试 Message Queue，可使用默认用户名和密码 (`admin/admin`) 运行 `imqcmd` 实用程序。

如果要设置生产系统，则必须为管理用户设置验证和授权。有关设置基于文件的用户系统信息库或配置 LDAP 目录服务器的使用的信息，请参见第 7 章“管理安全性”。在生产环境中，使用非默认用户名和密码是一项比较安全的做法。

- 如果要使用安全的代理连接，请在目标代理实例中设置并启用 `ssladmin` 服务。有关详细信息，请参见第 144 页“使用基于 SSL 的服务”。

使用 imqcmd 命令实用程序

`imqcmd` 命令实用程序可用于管理代理及其服务。

有关 `imqcmd` 命令的语法、子命令和选项的参考信息，请参见第 253 页第 13 章“命令参考”。在单独章节第 301 页第 15 章“物理目标属性参考”中，介绍了管理物理目标时使用的参考信息。

指定用户名和密码

因为每个 `imqcmd` 命令均通过用户系统信息库进行验证，因此要求提供用户名和密码。唯一的特例如下：

- 使用 `-h` 或 `-H` 选项显示帮助命令的命令。
- 使用 `-v` 选项显示产品版本的命令。

指定用户名

使用 `-u` 选项指定管理用户名。如果省略了用户名，该命令会提示您输入用户名。

为使本章中的示例便于阅读，我们将默认用户名 `admin` 显示为 `-u` 选项的参数。在生产环境中，应该使用自定义的用户名。

指定密码

按以下方法之一指定密码：

- 创建密码文件 (`passfile`) 并在该文件中输入密码。在命令行中，使用 `-passfile` 选项提供密码文件的名称。
- 让命令提示您输入密码。这种指定密码的方法最安全，除非有人看见了输入的内容。

在 `Message Queue` 以前的版本中，可以使用 `-p` 选项在命令行中指定密码。不赞成使用此选项，在今后的版本中会将其删除。

指定代理名和端口

`imqcmd` 的默认代理是本地主机上运行的代理，默认端口是 `7676`。

如果对远程主机和 / 或非默认端口上运行的代理执行命令，必须使用 `-b` 选项指定代理的主机和端口。

示例

本节中的示例说明了 `imqcmd` 的用法。

第一个示例列出了在 `localhost 7676` 端口上运行的代理的属性，因此不需要使用 `-b` 选项。该命令使用默认管理用户名 (`admin`) 并省略了密码，因此命令会提示您输入密码。

```
imqcmd query bkr -u admin
```

以下示例将列出在 `myserver 1564` 端口上运行的代理的属性。用户名是 `aladdin`。该命令要求更新用户系统信息库，以便将用户名 `aladdin` 分配给 `admin` 组。

```
imqcmd query bkr -b myserver:1564 -u aladdin
```

以下示例将列出在 `localhost 7676` 端口上运行的代理的属性。命令的初始超时设置为 `20` 秒，超时后的重试次数设置为 `7`。用户密码在名为 `myPassfile` 的密码文件中，该文件位于调用命令时的当前目录中。

```
imqcmd query bkr -u admin -passfile myPassfile -rtm 20 -rtr 7
```

对于安全的代理连接，这些示例可能会包括 `-secure` 选项。`-secure` 选项使 `imqcmd` 使用 `ssladmin` 服务（如果已配置并启动了该服务）。

显示帮助

要显示有关 `imqcmd` 命令实用程序的帮助，请使用 `-h` 或 `-H` 选项，而不要使用子命令。您无法获取有关特定子命令的帮助。

例如，以下命令显示有关 `imqcmd` 的帮助：

```
imqcmd -H
```

如果输入的命令行除了包括子命令或其他选项外，还包括 `-h` 或 `-H` 选项，则该命令实用程序只处理 `-h` 或 `-H` 选项。命令行中的所有其他项均被忽略。

显示产品版本

要显示 Message Queue 产品版本，请使用 `-v` 选项。例如：

```
imqcmd -v
```

如果输入的命令行除了包括子命令或其他选项外，还包括 `-v` 选项，则该命令实用程序只处理 `-v` 选项。命令行中的所有其他项均被忽略。

显示代理信息

要查询并显示某个代理的信息，请使用 `query bkr` 子命令。

下面是 `query bkr` 子命令的语法：

```
imqcmd query bkr -b hostName:port
```

该子命令列出默认代理或指定主机和端口上的代理的当前属性设置。它还列出与指定代理连接且正在运行的代理（在多代理群集中）。

例如：

```
imqcmd query bkr -u admin
```

提示输入密码后，此命令将产生类似如下内容的输出：

| | | |
|------------------|-----------|-----|
| Version | | 3.6 |
| 实例名 | imgbroker | |
| 主端口 | 7676 | |
| 系统中当前的消息数 | 0 | |
| 系统中当前的消息大小 (字节) | 0 | |
| 停用消息队列中的当前消息数量 | 0 | |
| 停用消息队列中的当前消息字节总数 | 0 | |
| 记录停用消息 | true | |
| 截断停用消息队列中的消息主体 | false | |
| 系统中的最大消息数 | 无限制 (-1) | |
| 系统中的最大消息大小 (字节) | 无限制 (-1) | |
| 最大消息大小 | 70m | |
| 自动创建队列 | true | |
| 自动创建主题 | true | |
| 自动创建队列的最大活动使用方数 | 1 | |
| 自动创建队列的最大备份使用方数 | 0 | |
| 群集代理列表 (处于活动状态) | | |
| 群集代理列表 (处于已配置状态) | | |
| 群集主代理 | | |
| 群集 URL | | |
| 日志等级 | INFO | |
| 日志转移间隔 (秒) | 604800 | |
| 日志转移大小 (字节) | 无限制 (-1) | |

更新代理属性

可以使用 `update bkr` 子命令更新以下代理属性:

- `img.autocreate.queue`
- `img.autocreate.topic`
- `img.autocreate.queue.maxNumActiveConsumers`
- `img.autocreate.queue.maxNumBackupConsumers`
- `img.cluster.url`
- `img.destination.DMQ.truncateBody`
- `img.destination.logDeadMsgs`
- `img.log.level`

- `imq.log.file.rolloversecs`
- `imq.log.file.rolloverbytes`
- `imq.system.max_count`
- `imq.system.max_size`
- `imq.message.max_size`
- `imq.portmapper.port`

下面是 `update bkr` 子命令的语法:

```
imqcmd update bkr [-b hostName:port] -o attribute=value [-o attribute=value1]...
```

此子命令更改默认代理或指定主机和端口的代理的指定属性。

第 14 章“代理属性参考”中介绍了这些属性

例如，以下命令可关闭队列目标的自动创建:

```
imqcmd update bkr -o "imq.autocreate.queue=false" -u admin
```

暂停和恢复代理

启动代理后，可以使用 `imqcmd` 子命令控制其状态。

暂停代理

暂停代理将暂停代理的连接服务线程，从而使代理停止侦听连接端口。其结果是代理将无法接受新连接、接收消息、发送消息。

但是，暂停代理不会暂停 `admin` 连接服务，允许执行控制代理消息流所需的管理任务。例如，如果某个特殊的物理目标受到消息攻击，您就可以暂停该代理，然后执行下列操作以期解决该问题:

- 跟踪消息来源
- 限制物理目标的大小
- 销毁物理目标。

暂停代理也不会暂停 `cluster` 连接服务。但是，群集中的消息传送依赖于群集中不同代理所执行的传送功能。

下面是 `pause bkr` 子命令的语法:

```
imqcmd pause bkr [-b hostName:port]
```

此命令暂停默认代理或指定主机和端口的代理。

以下命令暂停在 myhost 1588 端口上运行的代理。

```
imqcmd pause bkr -b myhost:1588 -u admin
```

也可以暂停单个连接服务和单个物理目标。有关详细信息，请参见第 110 页“暂停和恢复连接服务”和第 122 页“暂停和恢复物理目标”。

恢复代理

恢复代理将重新激活代理的服务线程，使代理恢复侦听端口。

下面是 `resume bkr` 子命令的语法：

```
imqcmd resume bkr [-b hostName:port]
```

此子命令恢复默认代理或指定主机和端口的代理。

以下命令将恢复在 localhost 7676 端口上运行的代理。

```
imqcmd resume bkr -u admin
```

关闭并重新启动代理

关闭代理将正常终止代理进程。代理停止接收新的连接和消息，完成现有消息的传送后将终止代理进程。

下面是 `shutdown bkr` 子命令的语法：

```
imqcmd shutdown bkr [-b hostName:port]
```

此子命令关闭默认代理或指定主机和端口的代理。

以下命令将关闭在 ctrlsrv 1572 端口上运行的代理：

```
imqcmd shutdown bkr -b ctrlsrv:1572 -u admin
```

您可以关闭并重新启动代理。下面是 `restart bkr` 子命令的语法：

```
imqcmd restart bkr [-b hostName:port]
```

该子命令关闭并重新启动默认代理或指定主机和端口的代理，（使用首次启动该代理时指定的选项）要选择其他选项，请关闭并重新启动该代理以指定所需的选项。

以下命令将重新启动在 localhost 7676 端口上运行的代理：

```
imqcmd restart bkr -u admin
```

显示代理度量

要显示有关代理的度量信息，请使用 `metrics bkr` 子命令。

下面是 `metrics bkr` 子命令的语法：

```
imqcmd metrics bkr [-b hostName:port]
                  [-m metricType] [-int interval] [-msp numSamples]
```

此子命令显示默认代理或指定主机和端口的代理的代理度量。

可以使用 `-m` 选项指定要显示的以下度量类型之一：

- **ttl** 显示流入和流出代理的消息和包的度量（默认度量类型）。
- **rts** 显示消息和包流入和流出代理的速率（每秒）的度量。
- **cxn** 显示连接、虚拟内存堆和线程。

使用 `-int` 选项指定显示度量的时间间隔（以秒为单位）。默认值为 5 秒。

使用 `-msp` 选项指定在输出中显示的样例数量。默认值为无限数（无穷大）。

例如，要获得消息在 10 秒间隔内流入和流出代理的速率，请使用以下命令：

```
imqcmd metrics bkr -m rts -int 10 -u admin
```

此命令语句的输出如下：

| 消息 / 秒 | | 消息字节 / 秒 | | 包 / 秒 | | 包字节 / 秒 | |
|--------|----|----------|------|-------|----|---------|------|
| 流入 | 流出 | 流入 | 流出 | 流入 | 流出 | 流入 | 流出 |
| 0 | 0 | 27 | 56 | 0 | 0 | 38 | 66 |
| 10 | 0 | 7365 | 56 | 10 | 10 | 7457 | 1132 |
| 0 | 0 | 27 | 56 | 0 | 0 | 38 | 73 |
| 0 | 10 | 27 | 7402 | 10 | 20 | 1400 | 8459 |
| 0 | 0 | 27 | 56 | 0 | 0 | 38 | 73 |

有关使用 `imqcmd` 报告代理度量的详细说明，请参见第 322 页“代理范围度量”。

管理连接服务

命令行实用程序包含的子命令可用于执行以下连接服务管理任务：

- 列出连接服务
- 显示连接服务信息
- 更新连接服务属性
- 显示连接服务度量
- 暂停和恢复连接服务

代理支持与应用程序客户机和管理客户机的连接。表 5-1 显示了 Message Queue 代理当前可用的连接服务。“服务名称”一栏中的值用于为 `-n` 选项指定服务名称。如该表所示，每项服务都与它所使用服务类型 `NORMAL`（用于应用程序客户机）或 `ADMIN`（用于管理客户机）和底层传输协议相关联。

表 5-1 代理支持的连接服务

| 服务名称 | 服务类型 | 协议类型 |
|---------------|--------|-------------------|
| jms | NORMAL | tcp |
| ssljms（企业版） | NORMAL | tls（其安全性基于 SSL） |
| httpjms（企业版） | NORMAL | http |
| httpsjms（企业版） | NORMAL | https（其安全性基于 SSL） |
| admin | ADMIN | tcp |
| ssladmin（企业版） | ADMIN | tls（其安全性基于 SSL） |

列出连接服务

要列出代理中可用的连接服务，请使用 `list svc` 子命令。

下面是 `list svc` 子命令的语法：

```
imqcmd list svc [-b hostName:port]
```

此子命令列出默认代理或指定主机和端口的代理中的所有连接服务。

在命令行中使用该子命令，如下所示：

```
imqcmd list svc [-b hostName:portNumber] -u admin
```

例如，以下命令列出在主机 myServer（位于 6565 端口）上运行的代理所支持的服务。

```
imqcmd list svc -b MyServer:6565 -u admin
```

以下命令列出在 localhost 7676 端口上运行的代理上的所有服务：

```
imqcmd list svc -u admin
```

该命令的输出信息如下：

| 服务名称 | 端口号 | 服务状态 |
|----------|------------|------|
| admin | 41844 (动态) | 运行中 |
| httpjms | - | 未知 |
| httpsjms | - | 未知 |
| jms | 41843 (动态) | 运行中 |
| ssladmin | 动态 | 未知 |
| ssljms | 动态 | 未知 |

显示连接服务信息

要查询并显示某项服务的信息，请使用 `query` 子命令。

下面是 `query svc` 子命令的语法：

```
imqcmd query svc -n serviceName [-b hostName:port]
```

有关在默认代理或指定主机和端口上运行的代理的指定服务的子命令信息。

例如：

```
imqcmd query svc -n jms -u admin
```

提示输入密码后，此命令将产生类似如下内容的输出：

| | |
|-----------|------------|
| 服务名称 | jms |
| 服务状态 | 运行中 |
| 端口号 | 60920 (动态) |
| 当前已分配的线程数 | 0 |
| 当前的连接数 | 0 |
| 最小线程数 | 10 |
| 最大线程数 | 1000 |

更新连接服务属性

可以使用 `update` 子命令更改表 5-2 中列出的一个或多个服务属性的值。

表 5-2 由 `imqcmd` 更新的连接服务属性

| 属性 | 说明 |
|-------------------------|---|
| <code>port</code> | 为要更新的服务指定的端口（不适用于 <code>httpjms</code> 和 <code>httpsjms</code> ）。0 值表示端口由端口映射器动态分配。 |
| <code>minThreads</code> | 为服务指定的最小线程数。 |
| <code>maxThreads</code> | 为服务指定的最大线程数。 |

下面是 `update` 子命令的语法：

```
imqcmd update svc -n serviceName [-b hostName:port]
-o attribute=value [-o attribute=value1]...
```

此子命令更新在默认代理或指定主机和端口上运行的代理中的指定服务的指定属性。有关服务属性的说明，请参见第 283 页“连接服务属性”。

以下命令将指定给 `jms` 服务的最小线程数更改为 20。

```
imqcmd update svc -n jms -o "minThreads=20" -u admin
```

显示连接服务度量

要显示有关某个服务的度量信息，请使用 `metrics` 子命令。

下面是 `metrics` 子命令的语法：

```
imqcmd metrics svc -n serviceName [-b hostName:port] [-m metricType]
[-int interval] [-msp numSamples]
```

此子命令显示默认代理或指定主机和端口上的代理中指定服务的度量。

使用 `-m` 选项指定要显示的度量类型：

- **ttl** 显示消息和包按指定连接服务方式流入和流出代理的度量。（默认度量类型）
- **rts** 显示消息和包按指定连接服务方式流入和流出代理的速率（每秒）度量。
- **cxn** 显示连接、虚拟内存堆和线程。

使用 `-int` 选项指定显示度量的时间间隔（以秒为单位）。默认值为 5 秒。

使用 `-msp` 选项指定在输出中显示的样例数量。默认值为无限数（无穷大）。

例如，要获得 `jms` 连接服务处理的消息和包的累计总数，请使用以下命令：

```
imqcmd metrics svc -n jms -m ttl -u admin
```

提示输入密码后，此命令将产生类似如下内容的输出：

| 消息 / 秒 | | 消息字节 / 秒 | | 包 / 秒 | | 包字节 / 秒 | |
|--------|-----|----------|-------|-------|-----|---------|--------|
| 流入 | 流出 | 流入 | 流出 | 流入 | 流出 | 流入 | 流出 |
| 164 | 100 | 120704 | 73600 | 282 | 383 | 135967 | 102127 |
| 657 | 100 | 483552 | 73600 | 775 | 876 | 498815 | 149948 |

有关使用 `imqcmd` 报告连接服务度量的详细说明，请参见第 324 页“[连接服务度量](#)”。

暂停和恢复连接服务

要暂停除管理服务之外的任何其他服务（不能暂停管理服务），请使用 `pause svc` 和 `resume svc` 子命令。

下面是 `pause svc` 子命令的语法：

```
imqcmd pause svc -n serviceName [-b hostName:port]
```

此子命令暂停在默认代理或指定主机和端口上运行的代理中的指定服务。不能暂停管理服务。

使用如下所示的命令行：

```
imqcmd pause svc -n serviceName -u admin
```

暂停服务影响如下：

- 代理将停止接受暂停服务的新客户机连接。如果 Message Queue 客户机尝试打开新的连接，将出现异常。
- 暂停服务的所有现有连接将保持活动状态，但是代理将暂停这些连接上的所有消息处理，直到服务恢复。（例如，如果客户机尝试发送消息，将禁止使用 `send()` 方法，直到服务恢复。）

- 会保留代理已经接收的任何消息的消息传送状态。（例如，服务恢复之后，事务不会中断，消息传送将恢复。）

要恢复服务，请使用 `resume svc` 子命令。

下面是 `resume svc` 子命令的语法：

```
imqcmd resume svc -n serviceName [-b hostName:port]
```

此子命令恢复在默认代理或指定主机和端口上运行的代理中的指定服务。

使用如下所示的命令行：

```
imqcmd resume svc -n serviceName -u admin
```

获得有关连接的信息

命令实用程序包含的子命令允许您列出并获得有关连接的信息。

`list cxn` 子命令列出指定服务名称的全部连接。下面是 `list cxn` 子命令的语法：

```
imqcmd list cxn [-svn serviceName] [-b hostName:port]
```

此子命令列出默认代理或指定主机和端口上的代理中指定服务名称的所有连接。如果未指定服务名称，将列出所有连接。

例如：

```
imqcmd list cxn -u admin
```

提示输入密码后，此命令将产生类似如下内容的输出：

| 正在列出指定的代理上的所有连接： | | | | | |
|---------------------|-------|-------|-----|-----|-----------|
| ----- | | | | | |
| 主机 | 主端口 | | | | |
| ----- | | | | | |
| localhost | 7676 | | | | |
| ----- | | | | | |
| 连接 ID | 用户 | 服务 | 生产方 | 使用方 | 主机 |
| ----- | | | | | |
| 1964412264455443200 | guest | jms | 0 | 1 | 127.0.0.1 |
| 1964412264493829311 | admin | admin | 1 | 1 | 127.0.0.1 |
| ----- | | | | | |
| 成功列出连接。 | | | | | |

要查询并显示某项连接服务的信息，请使用 `query` 子命令。

```
query cxn -n connectionID [-b hostName:port]
```

此子命令显示默认代理或指定主机和端口上的代理中的指定连接的信息。

例如：

```
imqcmd query cxn -n 421085509902214374 -u admin
```

提示输入密码后，此命令将产生类似如下内容的输出：

| | |
|--------|--------------------|
| 连接 ID | 421085509902214374 |
| 用户 | guest |
| 服务 | jms |
| 生产方 | 0 |
| 使用方 | 1 |
| 主机 | 111.22.333.444 |
| 端口 | 60953 |
| 客户机 ID | |
| 客户机平台 | |

管理长期订阅

要管理代理的长期订阅，需要使用 `imqcmd` 子命令。长期订阅是指某台客户机长期注册订阅某个主题；长期订阅有唯一标识，要求代理保留该订阅的消息，即使消息使用方处于非活动状态。通常，代理只能在消息已过期的情况下删除为长期订户保留的消息。

要列出指定物理目标的长期订阅，请使用 `list dur` 子命令。下面是 `list dur` 子命令的语法：

```
imqcmd list dur -d destName
```

例如，以下命令使用本地主机默认端口上的代理，列出了 `SPQuotes` 主题的所有长期订阅：

```
imqcmd list dur -d SPQuotes
```

对于某个主题下的每个长期订阅，`list dur` 子命令将返回长期订阅的名称、用户的客户机 ID、该主题下排队消息的数量以及长期订阅的状态（活动 / 非活动）。例如：

| 名称 | 客户机 ID | 消息数量 | 长期订户状态 |
|-----------|------------|------|----------|
| ----- | | | |
| myDurable | myClientID | 1 | INACTIVE |

可以使用 `list dur` 子命令返回的信息标识希望销毁或清除其消息的长期订阅。

`destroy dur` 子命令销毁带有指定客户机标识符的指定长期订阅。下面是 `destroy dur` 子命令的语法：

```
imqcmd destroy dur -n subscrName -c client_id
```

使用订阅名称和客户机 ID 来标识订阅。例如：

```
imqcmd destroy dur -n myDurable -c myClientID
```

`purge dur` 子命令清除带有指定客户机标识符的指定长期订阅的所有消息。下面是 `purge dur` 子命令的语法：

```
imqcmd purge dur -n subscrName -c client_id
```

管理事务

客户机应用程序启动的所有事务都由代理进行跟踪。这些事务可以由分布式事务（XA 资源）管理器管理的简单 `Message Queue` 事务或分布式事务。

每个事务都有一个 `Message Queue` 事务 ID，这是一个 64 位数字，唯一标识代理上的事务。分布式事务也有一个分布式事务 ID (XID)，长度为 128 字节，由分布式事务管理器指定。`Message Queue` 负责维护 `Message Queue` 事务 ID 与 XID 之间的关联。

对于分布式事务来说，失败的事务可能会保持 `PREPARED` 状态，而不会提交。因此，作为管理员需要监视并回滚或提交那些处于 `PREPARED` 状态的事务。

要列出代理跟踪的所有事务，请使用 `list txn` 命令。下面是 `list tx` 子命令的语法

```
imqcmd list txn
```

例如，以下命令列出某个代理中的所有事务。

```
imqcmd list txn
```

对于每个事务，`list` 子命令将返回事务 ID、状态、用户名、消息或确认数量以及创建时间。例如：

```
-----
```

| 事务 ID | 状态 | 用户名 | 消息数 / 确认数 | 创建时间 |
|-------|-------|-------|-----------|-------|
| ----- | ----- | ----- | ----- | ----- |

```
事务 ID 状态 用户名 消息数 / 确认数 创建时间
```

```
-----
```

| | | | | |
|----------------|----------|-------|-----|---------------------|
| 64248349708800 | PREPARED | guest | 4/0 | 1/30/02 10:08:31 AM |
| 64248371287808 | PREPARED | guest | 0/4 | 1/30/02 10:09:55 AM |

该命令显示了代理中的所有事务，包括本地事务和分布式事务。只能提交或回滚处于 PREPARED 状态的事务。只有当您知道该事务由于失败而处于 PREPARED 状态，而且分布式事务管理器当前没有提交该事务时才可以这样做。

例如，如果将代理的 `auto-rollback` 属性设置为 `false`（请参见第 285 页表 14-3），则必须在启动代理时手动提交或回滚处于 PREPARED 状态的事务。

`list` 子命令还显示事务中生成和确认的消息数量（#Msgs/#Acks）。提交事务之前不会传送消息，也不会处理确认。

`query` 子命令可以显示相同的信息以及许多其他值：客户机 ID、连接标识和分布式事务 ID (XID)。下面是 `query txn` 子命令的语法：

```
imqcmd query txn -n transaction_id
```

例如，以下示例生成的输出如下所示：

```
imqcmd query txn -n 64248349708800
```

下面是该命令生成的输出：

| | |
|--------|--|
| 客户机 ID | |
| 连接 | guest@192.18.116.219:62209->jms:62195 |
| 创建时间 | 1/30/02 10:08:31 AM |
| 确认数 | 0 |
| 消息数 | 4 |
| 状态 | PREPARED |
| 事务 ID | 64248349708800 |
| 用户名 | guest |
| XID | 6469706F6C7369646577696E6465723130313234313431313030373230 |

`commit` 和 `rollback` 子命令可用于提交和回滚分布式事务。正如前文所述，只能提交或回滚处于 PREPARED 状态的事务。

下面是 `commit` 子命令的语法：

```
imqcmd commit txn -n transaction_id
```

例如：

```
imqcmd commit txn -n 64248349708800
```

也可以配置代理，使它在启动时自动回滚处于 PREPARED 状态的事务。

下面是 rollback 子命令的语法：

```
imqcmd rollback txn -n transaction_id
```

有关详细信息，请参见第 285 页表 14-3 中的 `imq.transaction.autorollback` 属性。

管理物理目标

Message Queue 消息通过代理上的物理目标路由到其使用方客户机。代理管理与物理目标关联的内存和持久性存储器并设置其行为。

在群集中，您在某个代理上创建一个物理目标，然后群集会将该物理目标传播到所有代理。应用程序客户机可以订阅某一主题，或者使用来自群集中任何代理上的队列的主题，因为这些代理会协同工作将消息路由到整个群集。但是，只有最初生成消息的代理管理该消息的持久性和确认。

本章说明如何执行以下任务：

- 第 118 页 “使用 `imqcmd` 命令实用程序”
- 第 119 页 “创建物理目标”
- 第 120 页 “列出物理目标”
- 第 121 页 “显示有关物理目标的信息”
- 第 122 页 “更新物理目标属性”
- 第 122 页 “暂停和恢复物理目标”
- 第 123 页 “清除物理目标”
- 第 124 页 “销毁物理目标”
- 第 124 页 “压缩物理目标”
- 第 126 页 “配置停用消息队列的使用”

表 13-5 中提供了用于管理物理目标和完成上述任务的 `imqcmd` 子命令的完整参考信息。

注 每当与物理目标交互时，客户机应用程序就会使用 Destination 对象。为了实现提供者无关性和可移植性，客户机通常使用管理员创建的目标对象，它们被称为目标受管理对象。可以配置受管理对象以用于客户机应用程序，请参见第 8 章“管理受管理对象”

使用 imqcmd 命令实用程序

imqcmd 命令实用程序使您可以管理物理目标。imqcmd 命令的语法与使用它管理其他代理服务时的语法一样。

有关 imqcmd、其子命令和选项的完整参考信息，请参见第 253 页第 13 章“命令参考”。

子命令

表 6-1 列出了 imqcmd 子命令，本章中描述了其用法。有关这些子命令的参考信息，请参见第 262 页“物理目标管理子命令”。

表 6-1 imqcmd 命令实用程序的物理目标子命令

| 子命令和参数 | 说明 |
|-------------|-----------------------------|
| compact dst | 压缩一个或多个物理目标的内置的、基于文件的数据存储库。 |
| create dst | 创建物理目标。 |
| destroy dst | 销毁物理目标。 |
| list dst | 列出代理中的物理目标。 |
| metrics dst | 显示物理目标度量。 |
| pause dst | 暂停代理中的一个或多个物理目标。 |
| purge dst | 清除物理目标中的所有消息，但不销毁该物理目标。 |
| query dst | 查询并显示物理目标中的信息。 |
| resume dst | 恢复代理中的一个或多个暂停的物理目标。 |
| update dst | 更新目标属性。 |

创建物理目标

要创建物理目标，请使用 `imqcmd create` 子命令。下面是 `create` 子命令的语法：

```
create dst -t destType -n destName [-o property=value] [-o property=value1]...
```

创建物理目标时，请指定以下内容：

- 物理目标类型，`t`（主题）或 `q`（队列）。
- 物理目标名称。命名规则如下：
 - 名称必须仅包含字母数字字符。不能包含空格。
 - 名称可以以字母字符、下划线字符（`_`）或美元符号（`$`）开头。名称不能以字符串“`mq`”开头。
- 物理目标属性的任何非默认值。

也可以在更新物理目标时设置属性。

许多物理目标属性都管理代理的内存资源和消息流。例如，可以指定能发送到物理目标的生产方数量，它们可以发送的消息数量和大小，以及达到物理目标限制时代理应做出的响应。这些限制与代理配置属性控制的代理范围限制相似。

以下属性既适用于队列目标，也适用于主题目标：

- `maxNumMsgs`。指定物理目标中允许的未使用消息的最大数量。
- `maxTotalMsgBytes`。指定物理目标中允许的未使用消息内存的最大总量（以字节为单位）。
- `limitBehavior`。指定当达到内存限制的阈值时代理响应的方式。
- `maxBytesPerMsg`。指定物理目标中允许的任何单个消息大小的最大值（以字节为单位）。
- `maxNumProducers`。指定物理目标的生产方的最大数量。
- `consumerFlowLimit`。指定在一批中传送给使用方的消息的最大数量。
- `isLocalOnly`。仅适用于代理群集。指定物理目标不能在其他代理上复制，因而限制只将消息传送到本地使用方（连接到创建物理目标的代理的使用方）。
- `useDMQ`。指定是废弃物理目标的停用消息，还是将其放在停用消息队列中。

以下属性只适用于队列目标：

- `maxNumActiveConsumers`。指定从队列目标的负荷平衡传送中可以处于活动状态的最大使用方数。)

- `maxNumBackupConsumers`。在出现从队列目标的负荷平衡传送错误时，指定代替这些活动使用方的最大备份使用方数量。
- `localDeliveryPreferred`。仅应用于代理群集中的负荷平衡队列传送。指定仅当在本地代理中没有使用方时才将消息传送到远程的使用方。

有关物理目标属性的完整参考信息，请参见第 301 页第 15 章“物理目标属性参考”。

对于自动创建的目标，可在代理的实例配置文件中设置默认属性值。有关自动创建属性的参考信息位于第 286 页“表 14-4”中。

► 创建物理目标

- 要创建队列目标，请输入如下命令：

```
imqcmd create dst -n myQueue -t q -o "maxNumActiveConsumers=5"
```

- 要创建主题目标，请输入如下命令：

```
imqcmd create dst -n myTopic -t t -o "maxBytesPerMsg=5000"
```

列出物理目标

可以获得有关物理目标的当前属性值、物理目标相关的生产方和使用方的数量、以及消息传送度量（如物理目标中消息的数量和大小）的信息。

要查找希望获取其信息的物理目标，请列出代理中的所有物理目标。要执行此操作，请使用 `list dst` 子命令。下面是 `list dst` 子命令的语法：

```
list dst [-t destType] [-tmp]
```

该命令列出指定类型的物理目标。目标类型 (-t) 选项的值可以是 `q` (队列) 或 `t` (主题)。

如果省略目标类型，则会列出所有类型的物理目标。

`list dst` 子命令可选择性地指定要列出的目标类型或选择性地包含临时目标（使用 `-tmp` 选项）。临时目标由客户机创建，通常用于接收发送到其他客户机的消息回复。

例如，要获得在 `myHost 4545` 端口上运行的代理中的所有物理目标列表，请输入以下命令：

```
imqcmd list dst -b myHost:4545
```

除任何其他物理目标外，始终会显示停用消息队列 `mq.sys.dmq`，除非指定目标类型 `t` 只包括主题。

显示有关物理目标的信息

要获得有关物理目标当前属性值的信息，请使用 `query dst` 子命令。下面是 `query dst` 子命令的语法：

```
query dst -t destType -n destName
```

此命令列出有关指定类型和名称的目标的信息。例如：

```
imqcmd query dst -t q -n XQueue -u admin
```

此命令将产生类似如下内容的输出：

```

-----
目标名称      目标类型
-----
XQueue                队列

在由以下所指定的代理上：

-----
主机            主端口
-----
localhost       7676

目标名称                XQueue
目标类型                队列
目标状态                运行中
以管理方式创建          true

当前的消息数            0
当前的消息大小（字节）  0
当前的使用方数          0
当前的活动使用方数      0
当前的备份使用方数      0

最大消息数              无限制（-1）
最大消息大小（字节）    无限制（-1）
每条消息的最大大小（字节） 无限制（-1）
最大生产方数            100
最大活动使用方数        1
最大备份使用方数        0

限制行为                REJECT_NEWEST
使用方流限制            1000
是本地目标              false
首选本地传递            false
使用停用消息队列        true

```

输出同时还显示与目标相关联的生产方和使用方的数量。对于队列目标，该数量包括活动使用方和备份使用方。

可以使用 `update dst` 子命令更改一个或多个属性的值（请参见第 122 页“更新物理目标属性”）。

更新物理目标属性

可以使用 `update dst` 子命令更改物理目标的属性，并使用 `-o` 选项指定要更新的属性。下面是 `update dst` 子命令的语法：

```
update dst -t destType -n destName -o property=value [-o property=value1]...
```

此命令更新指定目标上指定属性的值。属性名可以是表 15-1 中介绍的任何属性。

可以多次使用 `-o` 选项来更新多个属性。例如，以下命令将 `maxBytesPerMsg` 属性更改为 1000，同时将 `MaxNumMsgs` 属性更改为 2000：

```
imqcmd update dst -t q -n myQueue -o "maxBytesPerMsg=1000"
-o "maxNumMsgs=2000" -u admin
```

有关可以更新的属性列表，请参见第 15 章“物理目标属性参考”。

不能使用 `update dst` 子命令更新物理目标的类型或更新 `isLocalOnly` 属性。

注 停用消息队列是专用的物理目标，其属性与其他目标的属性稍有不同。有关详细信息，请参见第 126 页“配置停用消息队列的使用”。

暂停和恢复物理目标

可暂停物理目标以控制从生产方到目标的消息传送，或从目标到使用方的消息传送，或者二者同时控制。特别是，可暂停到目标的消息流，有助于防止在消息生成明显快于使用时，目标所具有的过多消息。

要暂停传送到或传送出物理目标的消息，请使用 `pause dst` 子命令。下面是 `pause dst` 子命令的语法：

```
pause dst [-t destType -n destName] [-pst pauseType]
```

对于指定类型和名称的目标，此子命令暂停将消息传送给使用方 (`-pst CONSUMERS`)，或暂停从生产方传送消息 (`-pst PRODUCERS`)，或二者同时暂停 (`-pst ALL`)。如果未指定目标类型和名称，则暂停所有物理目标。默认值为 `ALL`。

示例：

```
imqcmd pause dst -n myQueue -t q -pst PRODUCERS -u admin
```

```
imqcmd pause dst -n myTopic -t t -pst CONSUMERS -u admin
```

要恢复向暂停目标的传送，请使用 `resume dst` 子命令。下面是 `resume dst` 子命令的语法：

```
resume dst [-t destType -n destName]
```

此子命令恢复向指定类型和名称的暂停目标的消息传送。如果未指定目标类型和名称，则恢复所有目标。

示例：

```
imqcmd resume dst -n myQueue -t q
```

在代理群集中，物理目标实例位于群集的各个代理中。必须逐一暂停各个目标。

清除物理目标

可以清除某个物理目标上当前排队的所有消息。清除物理目标意味着目标上排队的所有消息都将被删除。

当堆积的消息占用了大量系统资源时，可能需要清除这些消息。在某个队列没有注册使用方客户机，但仍在接收大量消息时可能会发生这种情况。如果某个主题的长期订户始终处于非活动状态也可能发生这种情况。在上述两种情况下，都没有必要保留消息。

要清除物理目标中的消息，请使用 `purge dst` 子命令。下面是 `purge dst` 子命令的语法：

```
purge dst -t destType -n destName
```

此子命令清除指定类型和名称的物理目标中的消息。

示例：

```
imqcmd purge dst -n myQueue -t q -u admin
```

```
imqcmd purge dst -n myTopic -t t -u admin
```

如果关闭代理后不希望在重新启动代理时传送过时消息，请使用 `-reset messages` 选项清除过时消息；例如：

```
imqbrokerd -reset messages -u admin
```

这样可以避免重新启动代理后清除目标的麻烦。

在代理群集中，物理目标实例位于群集的各个代理中。必须分别清除每个目标。

销毁物理目标

要销毁物理目标，请使用 `destroy dst` 子命令。下面是 `destroy dst` 子命令的语法：

```
destroy dst -t destType -n destName
```

该子命令销毁指定类型和名称的物理目标。

示例：

```
imqcmd destroy dst -t q -n myQueue -u admin
```

销毁物理目标将清除该目标中的所有消息并将该目标从代理上删除，此操作是不可逆的。

不能销毁停用消息队列。

压缩物理目标

如果使用内置的、基于文件的数据存储库（相对于插入的 JDBC 兼容数据存储库）作为消息的持久性存储，可以监视磁盘的利用率并在需要时压缩磁盘。

可构建基于文件的消息存储，以便根据消息所在的物理目标将其存储在目录中。在每个物理目标的目录中，大多数消息存储在一个文件中，该文件由大小可变的记录组成，即大小可变的记录文件。（为减少文件碎片，大小超过可配置的阈值的消息将存储在其自身的单独文件中。）

由于各种大小的消息可保持持久，并随后从大小可变的记录文件中删除，因而可能会在文件中出现漏洞，即文件中的空闲记录无法重新使用。

要管理未使用的空闲记录，命令行实用程序可包含用于监视每个物理目标的磁盘占用的子命令，以及在磁盘占用降低时用于回收空闲磁盘空间的子命令。

监视物理目标的磁盘占用

要监视物理目标的磁盘占用情况，请使用如下所示的命令：

```
imqcmd metrics dst -t q -n myQueue -m disk -u admin
```

此命令将产生类似如下内容的输出：

```
-----
保留的      已用的      占用率
-----
806400      804096      99
```

| | | |
|---------|---------|-----|
| 1793024 | 1793024 | 100 |
| 2544640 | 2518272 | 98 |

子命令输出中的各列具有以下含义：

表 6-2 物理目标磁盘占用度量

| 度量 | 说明 |
|-----|--|
| 保留的 | 所有记录使用的磁盘空间（以字节为单位），其中包括保存活动消息的记录以及等待重新使用的空闲记录 |
| 已用的 | 保存活动消息的记录使用的磁盘空间（以字节为单位） |
| 占用率 | 保留的磁盘空间除已用的磁盘空间所得的系数。比率越高，用于保存活动消息的磁盘空间就越多。 |

回收未使用的物理目标磁盘空间

磁盘占用模式取决于使用特定物理目标的消息发送应用程序的特征。根据流入和流出物理目标的消息的相对数量，以及消息的相对大小，保留的磁盘空间可能会随时间而增加。

如果消息生产率大于消息消费率，通常应该重新使用空闲记录，且占用率应偏高。但是，如果消息生产率等于或小于消息消费率，则占用率将较低。

总之，请尽量使保留的磁盘空间稳定并使磁盘占用率保持较高水平。通常，如果系统达到稳定状态，其中保留的磁盘空间量较为稳定，磁盘占用率较高（大于 75%），则不必回收未使用的磁盘空间。如果系统达到稳定状态，而占用率较低（低于 50%），可压缩磁盘以回收空闲记录占用的磁盘空间。

可以使用 `compact dst` 子命令压缩数据存储库。下面是 `compact dst` 子命令的语法：

```
compact dst [-t destType -n destName]
```

此子命令为指定类型和名称的物理目标压缩内置的、基于文件的数据存储库。如果未指定目标类型和名称，则会压缩所有目标。在压缩之前必须暂停物理目标。

如果保留的磁盘空间随时间持续增加，请通过设置目标内存限制属性和限制行为来重新配置目标的内存管理（请参见第 301 页“表 15-1”）。

► 回收未使用的物理目标磁盘空间

1. 暂停目标。

```
imqcmd pause dst -t q -n myQueue -u admin
```

2. 压缩磁盘。

```
imqcmd compact dst -t q -n myQueue -u admin
```

3. 恢复物理目标。

```
imqcmd resume dst -t q -n myQueue -u admin
```

如果未指定目标类型和名称，则会为所有物理目标执行此操作。

配置停用消息队列的使用

停用消息队列 `mq.sys.dmq` 是系统创建的物理目标，它保留了代理的停用消息及其他物理目标。停用消息队列是一种工具，用于监视、调整系统效率以及故障排除。有关术语“停用消息”的定义以及停用消息队列的更详细介绍，请参见 [Message Queue 技术概述](#)。

代理在启动时会自动创建停用消息队列。如果代理无法处理消息或者消息的有效期到期，它就会将消息放到该队列中。另外，其他物理目标也可以使用停用消息队列来保留废弃的消息。停用消息队列的使用提供了有利于排除系统故障的信息。

配置停用消息队列的使用

默认情况下，将物理目标配置为使用停用消息队列。可以通过设置物理目标属性 `useDMQ` 来禁止或允许物理目标使用停用消息队列。

在下面的示例中，创建了一个称为 `myDist` 的队列，它在默认情况下使用停用消息队列：

```
imqcmd create dst -n -myDist -t q
```

以下示例禁止上述队列使用停用消息队列：

```
imqcmd update dst -n myDist -t q -o useDMQ=false
```

通过设置 `imq.autocreate.destination.useDMQ` 代理属性，可允许或禁止代理中所有自动创建的物理目标使用停用消息队列。

配置和管理停用消息队列

imqcmd 命令实用程序用于管理停用消息队列。管理停用消息队列与管理其他队列相似，但有一些不同之处。例如，由于停用消息队列是系统创建的，因而不能创建、暂停或销毁该队列。

停用消息队列属性

配置停用消息队列与配置其他队列相似，但某些物理目标属性不应用默认值或具有不同的默认值。表 6-3 列出了停用消息队列以独特方式处理的队列属性。

表 6-3 标准物理目标属性的停用消息队列处理

| 属性 | 由停用消息队列执行的独特处理 |
|------------------------|---|
| limitBehavior | 对于停用消息队列，默认值是 REMOVE_OLDEST。对于其他队列，默认值是 REJECT_NEWEST。停用消息队列不支持流控制。 |
| localDeliveryPreferred | 不适用于停用消息队列。 |
| maxNumMsgs | 对于停用消息队列，默认值是 1000。对于其他队列，默认值是 -1（无限制）。 |
| maxNumProducers | 不适用于停用消息队列。 |
| maxTotalMsgBytes | 对于停用消息队列，默认值是 10 MB。对于其他队列，默认值是 -1（无限制）。 |
| isLocalOnly | 在代理群集中，停用消息队列始终是本地物理目标，并且将此属性永久性地设置为 true。但是，如果本地代理将这些消息标记为停用，则本地代理的停用消息队列可包含由群集中其他代理的客户机生成的消息。 |

消息内容

代理可将一条完整消息放到停用消息队列中，也可以废弃消息主体内容，只保留标题和属性数据。默认情况下，停用消息队列存储完整消息。

如果要减小队列的大小并且不打算恢复停用消息，请考虑废弃主体内容。

要废弃主体内容，只保留标题和属性数据，请将

imq.destination.DMQ.truncateBody 代理属性设置为 true，如下面的示例所示：

```
imqcmd update bkr -o imq.destination.DMQ.truncateBody=true
```

启用停用消息日志记录

除标准队列监视和日志记录选项外，还可以记录已被代理归类为停用的消息。

如果启用了停用消息日志记录，则代理会记录以下类型的事件：

- 代理将消息移动到停用消息队列中。
- 代理废弃来自停用消息队列以及来自不使用停用消息队列的任何物理目标的消息。
- 物理目标达到其限制。

默认情况下，禁用停用消息日志记录。以下示例启用停用消息日志记录：

```
imqcmd update bkr -o imq.destination.logDeadMsgs=true
```

停用消息日志记录适用于所有使用停用消息队列的物理目标。不能为单个物理目标启用或禁用日志记录。

管理安全性

作为管理员，您可以配置用来验证用户的用户系统信息库；定义访问控制；配置用来加密客户机 - 代理通信的安全套接字层 (SSL) 连接服务；以及设置代理启动时使用的密码文件。

本章包含以下小节：

- [第 130 页 “验证用户”](#)
- [第 138 页 “授权用户：访问控制属性文件”](#)
- [第 144 页 “使用基于 SSL 的服务”](#)
- [第 153 页 “使用密码文件”](#)
- [第 154 页 “创建审计日志”](#)

验证用户

您负责维护用户列表、用户组及其用户系统信息库中的用户密码。各个代理实例可以使用不同的用户系统信息库。本节将说明如何创建、填充和管理系统信息库。

当用户尝试连接到代理时，代理将通过检查所提供的名称和密码对用户进行验证。如果名称和密码与代理特定的用户系统信息库（配置每个代理时要查阅）中的名称和密码相匹配，则允许该用户与代理建立连接。

系统信息库可以是下列类型之一：

- **Message Queue** 发行时附带的平面文件系统信息库

此类型的用户系统信息库非常易于使用。您可以使用用户管理器实用程序 (`imqusermgr`) 填充和管理系统信息库。要启用验证，您需要用每个用户名、密码和用户组名称填充用户系统信息库。

有关设置和管理用户系统信息库的详细信息，请参见 [“使用平面文件用户系统信息库”](#)

- **LDAP** 服务器

这可以是现有的或新的 LDAP 目录服务器，这种服务器使用 LDAP v2 或 v3 协议。它并不象平面文件系统信息库那样易于使用，但它的可伸缩性更好，因此更适用于生产环境。

如果您目前使用的是 LDAP 用户系统信息库，请使用 LDAP 供应商提供的工具来填充和管理该用户系统信息库。有关详细信息，请参见第 135 页 [“使用 LDAP 服务器管理用户系统信息库”](#)。

使用平面文件用户系统信息库

Message Queue 提供了一个平面文件用户系统信息库，还提供了一个命令行工具 Message Queue 用户管理器 (`imqusermgr`)，您可以使用它填充和管理平面文件用户系统信息库。以下各节介绍平面文件用户系统信息库以及如何使用 Message Queue 用户管理器实用程序 (`imqusermgr`) 来填充和管理该系统信息库。

创建用户系统信息库

平面文件用户系统信息库是特定于实例的。默认的用户系统信息库（名为 `passwd`）是为启动的每个代理实例创建的。此用户系统信息库所在的目录由与该系统信息库相关联的代理实例的名称标识（请参见附录 A [“Message Queue 数据在特定操作系统中的位置”](#)）：

```
...instances/instanceName/etc/passwd
```

系统信息库创建时附带了两个条目（行），如下面的表 7-1 所示。

表 7-1 用户系统信息库中的初始条目

| 用户名 | 密码 | 组 | 状态 |
|-------|-------|-----------|--------|
| admin | admin | admin | 处于活动状态 |
| guest | guest | anonymous | 处于活动状态 |

使用这些初始条目时，Message Queue 代理安装后即可使用，不需要管理员进行干预。对于要使用的 Message Queue 代理，无需设置初始用户 / 密码。

使用初始 guest 用户条目时，客户机可以使用默认的 guest 用户名和密码连接到代理实例（例如，用于测试目的）。

使用初始 admin 用户条目时，您可以通过 imqcmd 命令，使用默认的 admin 用户名和密码管理代理实例。您应更新此初始条目以更改密码（请参见第 135 页“更改默认的管理员密码”）。

以下各节说明如何填充和管理平面文件用户系统信息库。

用户管理器实用程序 (imqusermgr)

此用户管理器实用程序 (imqusermgr) 允许您编辑或填充平面文件用户系统信息库。本节介绍用户管理器实用程序。下文说明如何使用 imqusermgr 子命令完成特定任务。

有关 imqusermgr 命令的完整参考信息，请参见第 13 章“命令参考”

使用 imqusermgr 之前，请谨记以下内容：

- 如果代理特定的用户系统信息库不存在，您必须启动相应的代理实例来创建此系统信息库。
- imqusermgr 命令必须在已经安装了代理的主机上运行。
- 您必须具有写入系统信息库的适当权限：即在 Solaris 和 Linux 上，您的身份必须是超级用户或首次创建代理实例的用户。

注 以下各节以默认代理实例作为示例。

子命令

`imqusermgr` 命令包含子命令 `add`、`delete`、`list` 和 `update`。

add 子命令 `add` 子命令将用户和关联的密码添加到指定的（或默认的）代理实例系统信息库中，并有选择地指定用户所属的组。子命令语法如下所示：

```
add [-i instanceName] -u userName -p passwd [-g group] [-s]
```

delete 子命令 `delete` 子命令从指定的（或默认的）代理实例系统信息库中删除指定用户。子命令语法如下所示：

```
delete [-i instanceName] -u userName [-s] [-f]
```

list 子命令 `list` 子命令显示指定的（或默认的）代理实例系统信息库中有关指定用户或所有用户的信息。子命令语法如下所示：

```
list [-i instanceName] [-u userName]
```

update 子命令 `update` 子命令更新指定的（或默认的）代理实例系统信息库中指定用户的密码和 / 或状态。子命令语法如下所示：

```
update [-i instanceName] -u userName -p passwd [-a state] [-s] [-f]
```

```
update [-i instanceName] -u userName -a state [-p passwd] [-s] [-f]
```

命令选项

表 7-2 列出了 `imqusermgr` 命令的选项。

表 7-2 `imqusermgr` 选项

| 选项 | 说明 |
|------------------------|---|
| -a <i>active_state</i> | 指定用户是否处于活动状态 (true/false)。true 表示处于活动状态。这是默认值。 |
| -f | 执行操作，无需用户确认。 |
| -h | 显示使用帮助。不执行命令行上的其他选项。 |
| -i <i>instanceName</i> | 指定此命令要应用到的代理实例用户系统信息库。如果未指定，则假定为默认的实例名称 <code>imqbroker</code> 。 |
| -p <i>passwd</i> | 指定用户密码。 |
| -g <i>group</i> | 指定用户组。有效值包括 <code>admin</code> 、 <code>user</code> 和 <code>anonymous</code> 。 |
| -s | 设置静音模式。 |

表 7-2 imqusermgr 选项 (续)

| 选项 | 说明 |
|--------------------|----------------------|
| -u <i>userName</i> | 指定用户名。 |
| -v | 显示版本信息。不执行命令行上的其他选项。 |

组

在代理实例的用户系统信息库中添加用户条目时，您可以指定以下三个预定义组之一：`admin`、`user` 或 `anonymous`。如果不指定任何组，则默认属于 `user` 组。

- **admin 组。** 用于代理管理员。默认情况下，指定到该组中的用户可以配置和管理代理。您可以为 `admin` 组指定多个用户。
- **user 组。** 用于普通（非管理）Message Queue 客户机用户。多数客户机用户都位于 `user` 组。默认情况下，该组中的用户可以生成发往所有主题和队列的消息，使用来自所有主题和队列的消息，以及浏览任意队列中的消息。
- **anonymous 组。** 用于那些不想使用代理已知用户名（可能是因为客户机应用程序不知道要使用的实际用户名）的 Message Queue 客户机。此帐户类似于大多数 FTP 服务器中的匿名帐户。每次只能为 `anonymous` 组分配一个用户。与 `user` 组相比，您应限定此组的访问权限，或者在部署时从该组中删除用户。

要更改某个用户所属的组，您必须删除该用户的条目，然后再添加一个用户条目并为其指定一个新组。

您不能重命名或删除这些系统创建的组，也不能创建新组。但是，您可以指定访问规则，定义该组的成员可以执行的操作。有关详细信息，请参见第 138 页“[授权用户：访问控制属性文件](#)”。

用户状态

向系统信息库中添加用户时，用户的默认状态是活动的。要使用户处于非活动状态，您必须使用 `update` 命令。例如，以下命令将使用户 `JoeD` 处于非活动状态：

```
imqusermgr update -u JoeD -a false
```

处于非活动状态的用户条目将保留在系统信息库中，但不能打开新连接。当某个用户处于非活动状态时，如果您试图添加具有相同名称的另一个用户，操作将失败。必须删除处于非活动状态的用户条目，或者更改新用户的名称，或者为新用户指定不同的名称。这样可以防止添加重复的用户名。

用户名和密码的格式

用户名和密码必须遵循以下原则：

- 用户名不能包含星号 (*)、逗号 (,)、冒号 (:)、新行符或回车符。
- 用户名或密码必须至少包含一个字符。
- 如果用户名或密码包含空格，必须将整个用户名或密码用引号引起来。
- 除了命令 `shell` 限定的命令行中最多可输入的字符数之外，密码或用户名的长度没有限制。

填充和管理用户系统信息库

要在系统信息库中添加用户，可以使用 `add` 子命令。例如，以下命令将向默认代理实例用户系统信息库添加用户名为 `Katharine`，密码为 `sesame` 的用户。

```
imqusermgr add -u Katharine -p sesame -g user
```

要从系统信息库中删除用户，可以使用 `delete` 子命令。例如，以下命令将删除用户 `Bob`：

```
imqusermgr delete -u Bob
```

要更改用户的密码或状态，可以使用 `update` 子命令。例如，以下命令将 `Katharine` 的密码更改为 `aladdin`：

```
imqusermgr update -u Katharine -p aladdin
```

要列出一个或所有用户的相关信息，可以使用 `list` 命令。以下命令将显示用户名为 `isa` 的相关信息：

```
imqusermgr list -u isa
```

```
% imqusermgr list -u isa

代理实例的用户系统信息库: imqbroker
-----
用户名          组    活动状态
-----
isa              admin  true
```

以下命令将列出所有用户的相关信息：

```
imqusermgr list
```

```
% imqusermgr list

代理实例的用户系统信息库: imqbroker
-----
用户名  组          活动状态
-----
admin   admin       true
guest   anonymous   true
isa     admin       true
testuser1  user       true
testuser2  user       true
testuser3  user       true
testuser4  user       false
testuser5  user       false
```

更改默认的管理员密码

为安全起见，应该将 admin 的默认密码更改为只有您自己知道的密码。为此，需要使用 imqusermgr 工具。

以下命令将 mybroker 代理实例的默认管理员密码由 admin 更改为 grandpoobah。

```
imqusermgr update mybroker -u admin -p grandpoobah
```

要快速确认此更改是否已生效，可以在代理实例运行时运行任何命令行工具。例如，以下命令将提示您输入密码：

```
imqcmd list svc mybroker -u admin
```

输入新密码 (grandpoobah) 将会被接受；而输入旧密码则会失败。

更改密码后，使用任何 Message Queue 管理工具（包含管理控制台）时都应该提供新密码。

使用 LDAP 服务器管理用户系统信息库

要使用 LDAP 服务器来管理用户系统信息库，请执行下列任务：

- 编辑实例配置文件

- 为管理员设置访问控制

编辑实例配置文件

要使代理使用目录服务器，请设置代理实例配置文件 `config.properties` 中某些属性的值。这些属性使代理实例可在 LDAP 服务器中查询有关用户和组的信息。只要用户试图连接到代理实例或执行某些信息传送操作，代理就会查询 LDAP 服务器。

实例配置文件位于代理实例目录下的某个目录中。路径的格式如下：

```
.../instances/instanceName/props/config.properties
```

有关特定操作系统中实例目录位置的信息，请参见附录 A “[Message Queue 数据在特定操作系统中的位置](#)”

► 编辑配置文件以使用 LDAP 服务器

1. 通过设置以下属性，指定您正在使用 LDAP 用户系统信息库：

```
imq.authentication.basic.user_repository=ldap
```

2. 设置 `imq.authentication.type` 属性，确定是以 Base64 编码的形式 (basic) 还是以 MD5 摘要的形式 (digest) 将密码从客户机传送给代理。使用 LDAP 目录服务器管理用户系统信息库时，必须将验证类型设置为 basic。例如，

```
imq.authentication.type=basic
```

3. 还必须设置控制 LDAP 访问的代理属性。这些属性存储在代理的实例配置文件中。本节后面部分说明并概述了这些属性。

Message Queue 使用 JNDI API 与 LDAP 目录服务器进行通信。有关这些属性的语法及其所引用术语的详细信息，请参见 JNDI 文档。Message Queue 使用 Sun JNDI LDAP 提供商并使用简单验证。

Message Queue 支持 LDAP 验证故障转移：可以为要尝试进行的验证指定 LDAP 目录服务器列表（请参见 `imq.user.repos.ldap.server` 属性的参考信息）。

请参见代理的 `config.properties` 文件，了解如何设置 LDAP 用户系统信息库相关属性的示例。

4. 必要时还需要编辑访问控制属性文件中的用户 / 组和规则。有关使用访问控制属性文件的详细信息，请参见第 138 页 “[授权用户：访问控制属性文件](#)”。
5. 如果您希望代理在连接验证和组搜索时通过 SSL 与 LDAP 目录服务器进行通信，您需要在 LDAP 服务器中激活 SSL，然后在代理配置文件中设置以下属性：
 - 指定 LDAP 服务器进行 SSL 通信时所使用的端口。例如：

```
imq.user_repository.ldap.server=myhost:7878
```

- 将代理属性 `imq.user_repository.ldap.ssl.enabled` 设置为 `true`。

这些是与 LDAP 相关的属性：

- `imq.user_repository.ldap.server`。LDAP 服务器的 *host:port*
- `imq.user_repository.ldap.principal`。代理用来绑定到目录服务器以进行搜索时使用的标识名。
- `imq.user_repository.ldap.password`。与代理使用的标识名关联的密码。
- `imq.user_repository.ldap.base`。用户条目的目录库。
- `imq.user_repository.ldap.uidattr`。提供商特定的属性标识符，其值用于唯一标识一个用户。例如：`uid`、`cn`。
- `imq.user_repository.ldap.usrfilter`。与用户配合使用的 JNDI 搜索过滤器。
- `imq.user_repository.ldap.grpsearch`。指定是否启用组搜索的布尔值。
- `imq.user_repository.ldap.grpbase`。组条目的目录库。
- `imq.user_repository.ldap.gidattr`。提供商特定的属性标识符，其值为组名。
- `imq.user_repository.ldap.memattr`。组条目中的属性标识符，其值是组成员的标识名。
- `imq.user_repository.ldap.grpfilter`。与组配合使用的 JNDI 搜索过滤器。
- `imq.user_repository.ldap.timeout`。指定搜索的时间限制（以秒为单位）的一个整数。
- `imq.user_repository.ldap.ssl.enabled`。指定代理在与 LDAP 服务器通信时是否应使用 SSL 协议的布尔值。

有关这些属性的完整参考信息，请参见第 291 页“安全性管理器属性”。

为管理员设置访问控制

要创建管理用户，请使用访问控制属性文件指定能够创建 ADMIN 连接的用户和组。必须在 LDAP 目录中预定义这些用户和组。

能够创建 ADMIN 连接的任何用户或组都可以执行管理命令。

► 设置管理用户

1. 通过将代理属性 `imq.accesscontrol.enabled` 设置为 `true`（这是默认值）来允许使用访问控制文件。

`imq.accesscontrol.enabled` 属性允许使用访问控制文件。

2. 打开访问控制文件 `accesscontrol.properties`。该文件的位置列于附录 A “Message Queue 数据在特定操作系统中的位置”

该文件包含一个如下所示的条目：

```
service connection access control
#####
connection.NORMAL.allow.user=*
connection.ADMIN.allow.group=admin
```

列出的条目是示例。请注意，`admin` 组存在于基于文件的用户系统信息库中，而非默认存在于 LDAP 目录中。您必须将在 LDAP 目录中定义的组的名称替换为要为其授予 Message Queue 管理员权限的组的名称。

3. 要为用户授予 Message Queue 管理员权限，请按如下所示输入用户名：

```
connection.ADMIN.allow.user=userName [, userName2, ..]
```

4. 要为组授予 Message Queue 管理员权限，请按如下所示输入组名称：

```
connection.ADMIN.allow.group=groupName [, groupName2, ..]
```

授权用户：访问控制属性文件

访问控制属性文件（ACL 文件）包含一些规则，用来指定用户和用户组可以执行的操作。您可以编辑 ACL 文件，将操作限定到某些用户和组。各个代理实例可以使用不同的 ACL 文件。

当客户机应用程序执行以下操作之一时，代理将对其 ACL 文件进行检查：

- 创建连接
- 创建生产方
- 创建使用方
- 浏览队列

代理将检查 ACL 文件以确定是授权生成请求的用户，还是用户所属的组来执行该操作。

如果对 ACL 文件进行编辑，新的设置在下一次代理检查该文件以验证授权时生效。编辑完文件后，不需要重新启动代理。

不管用户信息是放置在平面文件用户系统信息库中（请参见第 130 页“使用平面文件用户系统信息库”）还是放置在 LDAP 用户系统信息库（请参见第 135 页“使用 LDAP 服务器管理用户系统信息库”）中，均使用 ACL 文件。

创建访问控制属性文件

ACL 文件是特定于实例的。每次启动代理实例时，都会在该实例目录中创建一个名为 `accesscontrol.properties` 的默认文件。该文件的路径格式如下（请参见附录 A“Message Queue 数据在特定操作系统中的位置”）：

```
.../instances/brokerInstanceName/etc/accesscontrol.properties
```

ACL 文件的格式与 Java 属性文件类似。首先定义文件的版本，然后指定访问控制规则，规则分为三部分：

- 连接访问控制
- 物理目标访问控制
- 物理目标自动创建访问控制

`version` 属性定义 ACL 属性文件的版本，不能更改此条目。

```
version=JMQFileAccessControlModel/100
```

下面介绍指定访问控制的 ACL 文件的三个组成部分，然后说明访问规则的基本语法并介绍如何计算权限。

访问规则语法

在 ACL 属性文件中，访问控制用于定义特定用户或组对受保护的资源（如物理目标和连接服务）具有哪些访问权限。访问控制由一个规则或一组规则组成，每个规则都由一个 Java 属性表示：

这些规则的基本语法如下：

```
resourceType.resourceVariant.operation.access.principalType = principals
```

表 7-3 介绍了语法规则的元素。

表 7-3 访问规则的语法元素

| 元素 | 说明 |
|------------------------|---|
| <i>resourceType</i> | 下列选项之一：connection、queue 或 topic。 |
| <i>resourceVariant</i> | <i>resourceType</i> 指定的类型的一个实例。例如，myQueue。通配符 (*) 可用于表示所有连接服务类型或所有物理目标。 |
| <i>operation</i> | 其值取决于所设置访问规则的类型。 |
| <i>access</i> | 下列选项之一：allow 或 deny。 |
| <i>principalType</i> | 下列选项之一：user 或 group。有关详细信息，请参见第 133 页“组”。 |
| <i>principals</i> | 可能具有规则左侧指定了访问权限的人。如果 <i>principalType</i> 是 user，则可能是单个用户或以逗号分隔的用户列表；如果 <i>principalType</i> 是 group，则可能是单个组或以逗号分隔的组列表。通配符 (*) 可用于表示所有用户或所有组。 |

下面是一些访问规则实例：

- 以下规则表示所有用户都可以向名为 q1 的队列发送消息。
queue.q1.produce.allow.user=*
- 以下规则表示任何用户都可以向任何队列发送消息。
queue.*.produce.allow.user=*

注 要指定非 ASCII 用户、组或目标名称，请使用 Unicode 转义符 (\uXXXX) 表示法。如果在您编辑并保存的 ACL 文件中，这些名称采用了非 ASCII 编码，则可以通过 Java native2ascii 工具将此文件转换为 ASCII。有关详细信息，请访问 <http://java.sun.com/j2se/1.4/docs/guide/intl/faq.html>

权限的计算方式

当文件中存在多个访问规则时，将按如下方式计算权限：

- 指定的访问规则将覆盖一般访问规则。应用以下两条规则之后，所有用户都可以向所有队列发送消息，但是 Bob 不能向 tq1 发送消息。

```
queue.*.produce.allow.user=*
```

```
queue.tq1.produce.deny.user=Bob
```

- 指定给显式 **principal** 的访问权限将覆盖指定给 *** principal** 的访问权限。以下规则将拒绝 Bob 向 tq1 发送消息，但允许其他人发送。

```
queue.tq1.produce.allow.user=*
```

```
queue.tq1.produce.deny.user=Bob
```

- 用户的 *** principal** 规则将覆盖组的对应 *** principal** 规则。例如，以下两条规则允许所有授权用户向 tq1 发送消息。

```
queue.tq1.produce.allow.user=*
```

```
queue.tq1.produce.deny.group=*
```

- 授予用户的访问权限将覆盖授予用户所属组的访问权限。在以下示例中，即使 Bob 是 User 的成员，他也不能生成发送到 tq1 的消息。User 的其他所有成员都可以生成此类消息。

```
queue.tq1.produce.allow.group=User
```

```
queue.tq1.produce.deny.user=Bob
```

- 任何未通过访问规则显式授予的访问权限均默认为被拒绝。例如，如果 ACL 文件不包含任何访问规则，则所有用户的所有操作都将被拒绝。
- 如果同时允许和拒绝同一个用户或组的访问权限，则访问权限将相互抵消。例如，以下两条规则将使 Bob 无法浏览 q1：

```
queue.q1.browse.allow.user=Bob
```

```
queue.q1.browse.deny.user=Bob
```

以下两条规则阻止 User 组使用 q5 中的消息。

```
queue.q5.consume.allow.group=User
```

```
queue.q5.consume.deny.group=User
```

- 如果多条规则等号左侧的内容都相同，那么只有最后一条规则起作用。

用于连接服务的访问控制

ACL 属性文件中的连接访问控制部分包含代理连接服务的访问控制规则。连接访问控制规则的语法如下：

```
connection.resourceVariant.access.principalType = principals
```

为 *resourceVariant* 定义了两个值：NORMAL 和 ADMIN。这些预定义的值是您唯一能够授予访问权限的连接服务类型。

默认的 ACL 属性文件授予所有用户访问 NORMAL 连接服务的权限，并授予 admin 组中的用户访问 ADMIN 连接服务的权限：

```
connection.NORMAL.allow.user=*  
connection.ADMIN.allow.group=admin
```

如果使用的是基于文件的用户系统信息库，则由 imqusermgr 创建默认组 admin。如果使用的是 LDAP 用户系统信息库，则可通过执行以下操作之一来使用默认 ACL 属性文件：

- 在 LDAP 目录中定义一个名为 admin 的组。
- 使用在 LDAP 目录中定义的一个或多个组的名称来替换 ACL 属性文件中的名称 admin。

您可以对连接访问权限加以限定。例如，以下规则将拒绝 Bob 访问 NORMAL，但允许其他人访问：

```
connection.NORMAL.deny.user=Bob  
connection.NORMAL.allow.user=*
```

可以使用星号 (*) 指定所有授权用户或组。

使用 ACL 属性文件来授权访问 ADMIN 连接的方式与使用基于文件的用户系统信息库和 LDAP 用户系统信息库不同，如下所示：

- **基于文件的用户系统信息库**
 - 如果禁用访问控制，admin 组中的用户将具有 ADMIN 连接权限。
 - 如果启用访问控制，请编辑 ACL 文件。显式授予用户或组访问 ADMIN 连接服务的权限。
- **LDAP 用户系统信息库。**如果您使用的是 LDAP 用户系统信息库，请执行下列所有操作：
 - 启用访问控制。
 - 编辑 ACL 文件并提供可以建立 ADMIN 连接的用户或组的名称。指定在 LDAP 目录服务器中定义的任一用户或组。

对物理目标的访问控制

访问控制属性文件的目标访问控制部分包含基于物理目标的访问控制规则。这些规则决定谁（用户 / 组）可以在哪里（物理目标）执行什么（操作）。这些规则控制的访问类型包括向队列发送消息、向主题发布消息、从队列接收消息、订阅主题以及浏览队列中的消息。

默认情况下，任何用户或组都能够对任何物理目标进行任意类型的访问。您可以添加更多特定的目标访问规则或编辑默认的规则。本节下面介绍物理目标访问规则的语法，必须理解该语法才能编写自己的规则。

目标规则的语法如下：

```
resourceType.resourceVariant.operation.access.principalType = principals
```

表 7-4 列出了这些元素：

表 7-4 物理目标访问控制规则的元素

| 组件 | 说明 |
|------------------------|---|
| <i>resourceType</i> | 可以是 <code>queue</code> 或 <code>topic</code> 。 |
| <i>resourceVariant</i> | 某个物理目标名或所有物理目标 (*)，星号表示所有队列或所有主题。 |
| <i>operation</i> | 可以是 <code>produce</code> 、 <code>consume</code> 或 <code>browse</code> 。 |
| <i>access</i> | 可以是 <code>allow</code> 或 <code>deny</code> 。 |
| <i>principalType</i> | 可以是 <code>user</code> 或 <code>group</code> 。 |

可以将访问权限授予一个或多个用户和 / 或一个或多个组。

以下示例说明了不同类型的物理目标访问控制规则：

- 允许所有用户向任意 `queue` 目标发送消息。
`queue.*.produce.allow.user=*`
- 拒绝 `user` 组的任何成员订阅 `Admissions` 主题。
`topic.Admissions.consume.deny.group=user`

对自动创建的物理目标的访问控制

ACL 属性文件的最后一部分访问规则指定代理将为哪些用户和组自动创建物理目标。

当用户在尚不存在的物理目标上创建生产方或使用方时，如果已启用代理的自动创建属性，则代理将会创建该目标。

默认情况下，任何用户或组都有权让代理为其自动创建一个物理目标。此特权由以下规则指定：

```
queue.create.allow.user=*
topic.create.allow.user=*
```

您可以编辑 ACL 文件以限制此类访问权限。

物理目标自动创建访问规则的一般语法如下：

```
resourceType.create.access.principalType = principals
```

其中 *resourceType* 为 `queue` 或 `topic`。

例如，以下规则将允许代理为除 **Snoopy** 之外的每个用户自动创建 `topic` 目标。

```
topic.create.allow.user=*
topic.create.deny.user=Snoopy
```

请注意，物理目标自动创建规则的效果必须与物理目标访问规则的效果一致。例如，如果您 1) 更改目标访问规则，禁止任何用户向目标发送消息；但是 2) 启用了目标的自动创建，那么如果目标不存在，代理将创建一个物理目标，但不会向该目标发送任何消息。

使用基于 SSL 的服务

基于安全套接字层 (SSL) 标准的连接服务发送在客户机与代理之间发送的加密消息。本节将说明如何设置基于 SSL 的连接服务。

Message Queue 支持以下基于安全套接字层 (SSL) 标准的连接服务：

- `ssljms`、`ssladmin` 和 `cluster` 均通过 TCP/IP 使用。
- `httpsjms` 通过 HTTP 使用。

这些连接服务允许对在客户机和代理之间发送的消息进行加密。Message Queue 支持基于自签名服务器证书或签名证书的 SSL 加密。

为了使用基于 SSL 的连接服务，需要使用密钥工具实用程序 (`imqkeytool`) 生成专用密钥 / 公用密钥对。此实用程序将公用密钥嵌入自签名证书，此证书将传递给请求连接代理的客户机，客户机需要使用该证书建立加密连接。

尽管 Message Queue 基于 SSL 的连接服务在概念上很相似，但它们的设置方法却不尽相同。

本节后面部分将介绍如何通过 TCP/IP 建立安全连接。

基于 SSL 的连接服务（适用于通过 HTTP 连接的用户）(`httpsjms`)，使客户机和代理可以通过 HTTPS 隧道 Servlet 方式建立安全连接。有关通过 HTTP 建立安全连接的信息，请参见第 341 页附录 C “HTTP/HTTPS 支持”。

TCP/IP 的安全连接服务

以下基于 SSL 的连接服务通过 TCP/IP 提供直接、安全的连接：

- `ssljms` 服务可通过安全的加密连接在客户机与代理之间传送消息。
- `ssladmin` 服务将在 Message Queue 命令实用程序 (`imqcmd`) 与代理之间创建一个安全的加密连接。不支持管理控制台 (`imqadmin`) 安全连接。
- `cluster` 服务可通过安全的加密连接在群集中的代理之间传送消息并提供交叉代理通信（请参见第 180 页“代理间的安全连接”）。

配置自签名证书的使用

本节介绍如何使用自签名证书来设置基于 SSL 的服务。

如果需要更严密的验证，您可以使用由证书颁发机构检验的签名证书。先按照本节中的步骤进行操作，然后转至第 150 页“配置签名证书的使用”执行其他步骤。

► 设置基于 SSL 的连接服务

1. 生成自签名证书。
2. 在代理中启用 `ssljms`、`ssladmin` 或 `cluster` 连接服务。
3. 启动代理。
4. 配置并运行客户机（仅适用于 `ssljms` 连接服务）。

设置 `ssljms` 和 `ssladmin` 连接服务的过程基本相同，不同之处在于步骤 4，配置并运行客户机。

下文详细介绍了每个步骤。

步骤 1：生成自签名证书

Message Queue 带有自签名证书的 SSL 支持用于保护所传输数据的安全，假定客户机正在与已知且可信任的服务器进行通信。

运行 `imqkeytool` 命令，为代理生成自签名证书。在 UNIX® 系统上，可能需要以超级用户 (root) 身份运行 `imqkeytool` 命令，以获得创建密钥库所需的权限。

可以对 `ssljms`、`ssladmin` 或 `cluster` 连接服务使用相同的证书。

在命令提示符下输入以下内容：

```
imqkeytool -broker
```

实用程序会提示您输入密钥库密码。

```
Generating keystore for the broker ...
Enter keystore password:
```

然后，实用程序将提示信息，要求标识拥有该证书的代理。您提供的信息将构成 X.500 标识名。下表列出了一些提示，对其进行描述，并为每个提示提供了一个示例。值区分大小写，并且可以包含空格。

表 7-5 自签名证书所需的标识名信息

| 提示 | 说明 | 示例 |
|----------------------|---|------------------|
| 您的姓名是什么？ | X.500 commonName (CN) 。输入正在运行代理的服务器的全限定名称。 | myhost.sun.com |
| 您所在部门的名称是什么？ | X.500 organizationalUnit (OU) 。输入部门或分部的名称。 | purchasing |
| 您的工作单位的名称是什么？ | X.500 organizationName (ON) 。大型工作单位的名称，如公司或政府机构。 | My Company, Inc. |
| 您所在城市或地区的名称是什么？ | X.500 localityName (L) 。 | San Francisco |
| 您所在国家或省市的名称是什么？ | X.500 stateName (ST) 。输入国家或省市的全称，不要使用缩写。 | California |
| 此单位的两字母国家 / 地区代码是什么？ | X.500 country (C) 。 | US |

当您输入信息后，`imqkeytool` 将显示该信息以进行确认。例如：

Is CN=mqserver.sun.com, OU=purchasing, O=My Company, Inc., L=San Francisco, ST=California, C=US correct?

要重新输入值，请接受默认值或输入 no；要接受当前值并继续，请输入 yes。确认后，imqkeytool 将生成关键字对而暂停其他操作。

接下来，imqkeytool 要求您输入密码以锁定特定关键字对（关键字密码）。按回车键响应此提示，以使用与关键字密码和密钥库密码相同的密码。

注 请记住您提供的密码。在启动代理时，您必须提供此密码以使代理打开密钥库。可以将密钥库密码存储在密码文件中（请参见第 153 页“使用密码文件”）。

运行 imqkeytool 命令以运行 JDK keytool 实用程序，生成自签名证书并将其放置 Message Queue 的密钥库中。密钥库所在的目录因操作系统而异，如附录 A “Message Queue 数据在特定操作系统中的位置” 中所示。

密钥库格式与 JDK1.2 keytool 实用程序支持的格式相同。

这些是 Message Queue 密钥库的可配置属性。

- imq.keystore.file.dirpath。适用于基于 SSL 的服务：指定包含密钥库文件的目录的路径。有关默认值的信息，请参见附录 A “Message Queue 数据在特定操作系统中的位置”。
- imq.keystore.file.name。适用于基于 SSL 的服务：指定密钥库文件的名称。
- imq.keystore.password。适用于基于 SSL 的服务：指定密钥库密码。

您可能需要重新生成关键字对以解决某些问题，例如：

- 您忘记了密钥库密码。
- 启动代理时，基于 SSL 的服务初始化失败而出现异常
java.security.UnrecoverableKeyException:Cannot recover key。

出现异常的原因可能是您提供的密码与您在第 146 页“步骤 1：生成自签名证书”中生成自签名证书时的密钥库密码不同。

► 重新生成关键字对

1. 删除代理的密钥库，它的位置如附录 A “Message Queue 数据在特定操作系统中的位置” 中所示。
2. 再次运行 imqkeytool，生成第 146 页“步骤 1：生成自签名证书”中描述的关键字对。

步骤 2: 在代理中启用基于 SSL 的服务

要在代理中启用基于 SSL 的服务，您需要将 `ssljms`（或 `ssladmin`）添加到 `imq.service.activelist` 属性中。

注 基于 SSL 的 cluster 连接服务是使用 `imq.cluster.transport` 属性启用的，而不是使用 `imq.service.activelist` 属性启用的。请参见第 180 页“代理间的安全连接”。

► 在代理中启用基于 SSL 的服务

1. 打开代理的实例配置文件。

实例配置文件位于一个目录中，该目录用与此配置文件相关联的代理实例的名称 (`instanceName`) 标识（请参见附录 A “Message Queue 数据在特定操作系统中的位置”）：

```
.../instances/instanceName/props/config.properties
```

2. 为 `imq.service.activelist` 属性添加一个条目（如果此条目尚不存在），并将基于 SSL 的服务包括在列表中。

默认情况下，此属性包括 `jms` 和 `admin` 连接服务。您需要添加 `ssljms` 或 `ssladmin` 连接服务或两个服务都添加（取决于您要激活的服务）：

```
imq.service.activelist=jms,admin,ssljms,ssladmin
```

步骤 3: 启动代理

启动代理并提供密钥库密码。可以通过以下方法之一提供密码：

- 让代理启动时提示您输入密码：

```
imqbrokerd
Please enter Keystore password:mypassword
```

- 将密码放在密码文件中，如第 153 页“使用密码文件”中所述。将密码放在密码文件中并设置属性 `imq.passfile.enabled=true` 后，请执行以下操作之一：

- 将密码文件的位置传递给 `imqbrokerd` 命令：

```
imqbrokerd -passfile /tmp/mypassfile
```

- 启动代理时不选择 `-passfile` 选项，但使用以下两个代理配置属性指定密码文件的位置：

```
imq.passfile.dirpath=/tmp
```

```
imq.passfile.name=mypassfile
```

启用带有 SSL 的代理或客户机时，您可能会注意到它在几秒钟内使用了大量的 CPU 资源。这是因为 Message Queue 使用 JSSE（Java 安全套接扩展）来实现 SSL。JSSE 使用 `java.security.SecureRandom()` 生成随机数。此方法需要大量时间生成初始随机数初始化向量，这就是您看到 CPU 使用增加的原因。生成初始化向量后，CPU 的使用将降到正常水平。

步骤 4：配置并运行基于 SSL 的客户机

最后，配置客户机以使用安全连接服务。有两种通过 TCP/IP 的安全连接方案：

- 使用 `ssljms` 的应用程序客户机
- 使用 `ssladmin` 的 Message Queue 管理客户机（如 `imqcmd`）

下面分别介绍这两种客户机。

使用 `ssljms` 的应用程序客户机

您必须确保客户机的类路径中有所需的安全套接扩展 (JSSE) Jar 文件，还需要告知该文件使用 `ssljms` 连接服务。

1. 如果您的客户机不是使用 J2SDK1.4（它具有内置的 JSSE 和 JNDI 支持），请确保客户机的类路径中有以下 Jar 文件：

```
jsse.jar、jnet.jar、jcert.jar 和 jndi.jar
```

2. 请确保客户机的类路径中有以下 Message Queue Jar 文件：

```
imq.jar 和 jms.jar
```

3. 启动客户机并连接到代理的 `ssljms` 服务。执行上述操作的方法之一是输入如下命令：

```
java -DmqConnectionType=TLS clientAppName
```

设置 `mqConnectionType`，通知连接使用 SSL。

有关在客户机应用程序中使用 `ssljms` 连接服务的详细信息，请参见 Message Queue Developer's Guide for Java Clients 中有关使用受管理对象的章节。

使用 `ssladmin` 的管理客户机 (`imqcmd`)

您可以通过在使用 `imqcmd` 时包含 `-secure` 选项来建立一个安全的管理连接。例如：

```
imqcmd list svc -b hostName:port -u adminName -secure
```

其中，*adminName* 是 Message Queue 用户系统信息库中的有效条目，该命令将提示您输入密码。（如果使用的是平面文件系统信息库，请参见第 135 页“更改默认的管理员密码”）。

列出连接服务是查看 `ssladmin` 服务是否正在运行的一种方法，您可以成功建立安全管理连接，如以下输出所示：

列出指定的代理上的所有服务：

| | | |
|-----------|------------|------|
| 主机 | 主端口 | |
| localhost | 7676 | |
| 服务名称 | 端口号 | 服务状态 |
| admin | 33984 (动态) | 运行中 |
| httpjms | - | 未知 |
| httpsjms | - | 未知 |
| jms | 33983 (动态) | 运行中 |
| ssladmin | 35988 (动态) | 运行中 |
| ssljms | 动态 | 未知 |

成功列出服务。

配置签名证书的使用

签名证书可以提供比自签名证书更严密的服务器验证。要实现签名证书，请将其安装到密钥库中，然后对 Message Queue 客户机进行配置，以使其在建立与 `imqbrokerd` 的 SSL 连接时请求签名证书。

您只能在客户机与代理之间实现签名证书，而不能在群集中的多个代理之间实现。

后面的说明假定您已经执行了第 145 页“配置自签名证书的使用”中所述的步骤。在按照这些说明进行操作时，访问 <http://java.sun.com> 中有关 J2SE keytool 和 X.509 证书的信息可能会有所帮助。

步骤 1：获取和安装签名证书

► 获取签名证书

1. 使用 J2SE keytool 为刚刚生成的自签名证书生成一个证书签名请求 (CSR)。

以下是一个示例：

```
keytool -certreq -keyalg RSA -alias imq -file certreq.csr
        -keystore /etc/imq/keystore -storepass myStorePassword
```

CSR 现在将证书封装到文件 certreq.csr 中。

2. 通过以下方法之一生成或请求签名证书：
 - 由众所周知的证书颁发机构 (CA) 签署证书，如 Thawte 或 Verisign。有关此过程的详细信息，请参见 CA 文档。
 - 使用 SSL 签名软件包亲自对证书进行签名。

最终的签名证书是一个 ASCII 字符序列。如果从 CA 收到签名证书，它可能是电子邮件附件或消息文本。

3. 在收到签名证书后，请将其保存到文件中。

以下说明使用示例名 broker.cer 来表示代理证书。

► 安装签名证书

1. 检查 `$JAVA_HOME/lib/security/cacerts` 以查明 J2SE 是否默认支持您的 CA，如下所示：

```
keytool -v -list -keystore $JAVA_HOME/lib/security/cacerts
```

此命令列出系统密钥库中的根 CA。

如果您的 CA 已列出，请跳过下一步。

2. 如果 J2SE 中不支持您的 CA，请将证书颁发机构的根证书导入到 imqbrokerd 密钥库中。

以下是一个示例：

```
keytool -import -alias ca -file ca.cer -noprompt -trustcacerts
        -keystore /etc/imq/keystore -storepass myStorePassword
```

ca.cer 值是从 CA 获取的 CA 根证书。

如果您使用的是 CA 测试证书，则可能需要导入测试 CA 根证书。您的 CA 应提供有关如何获取测试 CA 根副本的说明。

3. 将签名证书导入到密钥库中以替换原来的自签名证书。

例如：

```
keytool -import -alias imq -file broker.cer -noprompt -trustcacerts
        -keystore /etc/imq/keystore -storepass myStorePassword
```

broker.cer 值是包含从 CA 收到的签名证书的文件。

imqbrokerd 密钥库现在包含一个用于 SSL 连接的签名证书。

步骤 2: 配置客户机运行时以请求签名证书

► 配置 Java 客户机运行时

默认情况下, Message Queue 客户机运行时信任 imqbrokerd 并接受提供给它的任何证书。现在, 您必须将客户机运行时配置为请求签名证书, 并确保客户机信任对该证书进行签名的 CA。

1. 要对客户机进行配置以使其从 imqbrokerd 请求有效的签名证书, 请将客户机的 ConnectionFactory 对象的 imqSSLIsHostTrusted 属性设置为 false。
2. 尝试建立到 imqbrokerd 的 SSL 连接, 如第 149 页“步骤 4: 配置并运行基于 SSL 的客户机”中所述。

如果代理的证书是由众所周知的 CA 签名的, 则连接可能会成功, 您可以跳过下一步。如果连接由于证书验证错误而失败, 请执行下一步。

3. 在客户机的信任库 (truststore) 中安装签名 CA 的根证书, 如下面各节所述。

有三个选项可用于配置具有信任库的客户机:

- 将根 CA 安装到默认系统 cacerts 文件中。
- 将根 CA 安装到备用系统文件 jssecacerts 中。这是推荐选项。
- 将根 CA 安装到任一密钥库文件中, 并将客户机配置为使用该文件作为其信任库。

下面几部分包含有关如何使用这些选项来安装 Verisign Test Root CA 的示例。根 CA 包含在名为 testrootca.cer 的文件中。这些示例假定 J2SE 安装在 /usr/j2se 中。

安装到默认的系统 cacerts 文件中

本示例将根 CA 安装到文件 \$JAVA_HOME/usr/jre/lib/security/cacerts 中。

```
keytool -import -keystore /usr/j2se/jre/lib/security/cacerts
        -alias VerisignTestCA -file testrootca.cer -noprompt
        -trustcacerts -storepass myStorePassword
```

默认情况下, 客户机搜索此密钥库, 因此无需对客户机进行进一步的配置。

安装到 jssecacerts 中

本示例将根 CA 安装到文件 \$JAVA_HOME/usr/jre/lib/security/jssecacerts 中。

```
keytool -import -keystore /usr/j2se/jre/lib/security/jssecacerts
        -alias VerisignTestCA -file testrootca.cer -noprompt
        -trustcacerts -storepass myStorePassword
```

默认情况下, 客户机搜索此密钥库, 因此无需对客户机进行进一步的配置。

安装到其他文件中

本示例将根 CA 安装到文件 `/home/smith/.keystore` 中。

```
keytool -import -keystore /home/smith/.keystore
        -alias VerisignTestCA -file testrootca.cer -noprompt
        -trustcacerts -storepass myStorePassword
```

默认情况下，客户机不搜索此密钥库，因此您必须向客户机提供信任库的位置。为此，请在客户机运行后设置 Java 系统属性 `javax.net.ssl.trustStore`。例如：

```
javax.net.ssl.trustStore=/home/smith/.keystore
```

使用密码文件

多种类型的命令都需要密码。在表 7-6 中，第一列列出了需要密码的命令，第二列列出了需要密码的原因。

表 7-6 使用密码的命令

| 命令 | 用途 | 密码的用途 |
|-------------------------|------------|---------------------------------------|
| <code>imqbrokerd</code> | 启动代理 | 访问插入的持久性数据存储库、SSL 证书密钥库或 LDAP 用户系统信息库 |
| <code>imqcmd</code> | 管理代理 | 验证被授权使用此命令的管理用户 |
| <code>imqdbmgr</code> | 管理插入的数据存储库 | 访问数据存储库 |

可以在密码文件 (*passfile*) 中指定这些密码，并使用 `-passfile` 选项指定该文件的名称。下面是 `-passfile` 选项的格式：

```
imqbrokerd -passfile myPassfile
```

注 在以前的版本中，您可以使用 `-p`、`-password`、`-dbpassword` 和 `-ldappassword` 选项在命令行指定密码。不赞成使用这些选项，在今后的版本中会将其删除。在这一版本中，命令行中的这些选项值将取代密码文件中相关联的值。

安全性问题

以交互方式指定密码以响应提示是最安全的密码指定方法，除非他人也可以看到您的监视器。您也可以通过命令行指定密码文件。但是，以非交互方式使用命令时，必须使用密码文件。

密码文件是未加密的，因此，您必须设置其权限以防止对其进行未授权的访问。设置权限以限制可以查看该文件的用户，但为启动代理的用户提供读取访问权限。

密码文件内容

密码文件是包含一组属性和值的纯文本文件。每个值都是由某一命令使用的密码。

表 7-7 列出了密码文件可能包含的密码：

表 7-7 密码文件中的密码

| 密码 | 受影响的命令 | 说明 |
|-----------------------------------|-----------------------|--|
| imq.imqcmd.password | imqcmd | 为 imqcmd 命令行指定管理员密码。每个命令都需要验证此密码。 |
| imq.keystore.password | imqbrokerd | 为基于 SSL 的服务指定键存储密码。 |
| imq.persist.jdbc.password | imqbrokerd imdbmgr | 指定必要时用于打开数据库连接的密码。 |
| imq.user_repository.ldap.password | imqbrokerd | 指定与代理的标识名（用于绑定到已配置的 LDAP 用户系统信息库）相关联的密码。 |

样例密码文件是 Message Queue 产品的一部分。有关样例文件的位置信息，请参见附录 A “Message Queue 数据在特定操作系统中的位置”。

创建审计日志

Message Queue 仅在企业版中支持审计日志。启用审计日志后，Message Queue 将为下列类型的事件生成记录：

- 启动、关闭、重新启动以及删除代理实例
- 用户验证和授权

- 重置持久性存储
- 创建、清除以及破坏物理目标
- 以管理方式破坏长期订户

要将审计记录记录到 Message Queue 代理日志文件中，请将 `imq.audit.enabled` 代理属性设置为 `true`。日志中的所有审计记录都包含关键字 `AUDIT`。

有关 `imq.audit.enabled` 属性的参考信息，请参见第 291 页“安全性管理器属性”。

创建审计日志

管理受管理对象

使用受管理对象可以开发供其他 JMS 提供者使用的客户机应用程序。受管理对象封装提供者特定的配置和命名信息。

通常，Message Queue 管理员为客户机应用程序创建受管理对象，以便在获取代理连接时使用。客户机应用程序使用连接将消息发送到物理目标，并从物理目标接收消息。

本章说明如何使用对象管理器实用程序 (imqobjmgr) 执行上述任务。由于这些任务涉及了解您正在使用的对象存储库以及您正在创建的受管理对象的属性，本章在说明如何使用 imqobjmgr 管理受管理对象之前介绍了这两个主题的背景。

本章包含以下小节：

- [第 157 页 “关于对象存储库”](#)
- [第 160 页 “关于受管理对象的属性”](#)
- [第 167 页 “使用对象管理器实用程序 \(imqobjmgr\)”](#)
- [第 170 页 “添加和删除受管理对象”](#)
- [第 174 页 “列出受管理对象”](#)
- [第 174 页 “获取有关单个对象的信息”](#)
- [第 175 页 “更新受管理对象”](#)

关于对象存储库

受管理对象放置在便于访问的对象存储库中，客户机应用程序可以通过 JNDI 查找访问这些对象。可以使用两种类型的对象存储库：标准 LDAP 目录服务器和文件系统对象存储库。

LDAP 服务器对象存储库

对于生产型消息传送系统，建议使用 LDAP 服务器对象存储库。很多供应商都提供 LDAP 实现，而且 LDAP 实现可以在分布式系统中使用。LDAP 服务器还提供对生产环境特别有用的安全功能。

Message Queue 管理工具可以管理 LDAP 服务器上的对象存储库。不过，您可能需要按照 LDAP 服务器文档中的规定先配置 LDAP 服务器以存储 Java 对象并执行 JNDI 查找。

将 LDAP 服务器用作对象存储库时，需要指定表 8-1 中显示的属性。这些属性分为以下几类：

- **初始上下文：**对于 LDAP 服务器对象存储库，该属性是固定的。
- **位置：**按照 LDAP 服务器中的设置指定用于存储受管理对象的 URL 和目录路径。特别是，必须检查指定的路径是否存在。
- **安全信息：**取决于 LDAP 提供者。应参见 LDAP 实现附带的文档，以决定是所有的操作都需要安全信息，还是只有更改存储数据的操作需要安全信息。

表 8-1 LDAP 对象存储库属性

| 属性 | 说明 |
|----------------------------------|---|
| java.naming.factory.initial | LDAP 服务器上 JNDI 查找的初始上下文 com.sun.jndi.ldap.LdapCtxFactory |
| java.naming.provider.url | LDAP 服务器 URL 和目录路径信息。例如： ldap://mydomain.com:389/ou=mqobjs,o=myapp 其中受管理对象存储库在 /myapp/mqobjs 目录中。 |
| java.naming.security.principal | 负责验证 LDAP 服务器呼叫者的人员的身份。此条目的格式取决于验证方案。例如： uid=fooUser, ou=People, o=mq 如果不指定此属性，性能将由 LDAP 服务提供商决定。 |
| java.naming.security.credentials | 负责验证 LDAP 服务器呼叫者的人员的凭证。此属性的值取决于验证方案：它可能是散列密码、纯文本密码、密钥、证书等。例如： fooPasswd 如果不指定此属性，性能将由 LDAP 服务提供商决定。 |

表 8-1 LDAP 对象存储库属性 (续)

| 属性 | 说明 |
|-------------------------------------|---|
| java.naming.security.authentication | <p>要使用的安全级别。其值是以下关键字之一：none、simple 或 strong。</p> <p>例如，如果指定 simple，缺少任何 principal 或 credential 值时，系统都会提示您。这使您可以一种更安全的方式提供身份信息。</p> <p>如果不指定此属性，性能将由 LDAP 服务提供商决定。</p> |

文件系统对象存储库

Message Queue 还支持文件系统对象存储库实现。文件系统对象存储库尚未进行完整测试，因此不建议用于生产系统。它的优势就是在开发环境中非常易于使用。您必须在本地文件系统中创建一个目录，而非设置一台 LDAP 服务器，。

不过，文件系统存储不能用作在多个计算机节点上配置的客户机的集中式对象存储库，除非这些客户机可以访问该对象存储库所在的目录。此外，任何可以访问这个目录的用户都可以使用 Message Queue 管理工具创建和管理受管理对象。

使用文件系统对象存储库时，需要指定表 8-2 中显示的属性。这些属性分为以下几类：

- **初始上下文：**对于文件系统对象存储库，该属性的值是固定的。
- **位置：**该属性的值指定用于存储受管理对象的目录路径。目录必须存在且必须拥有 Message Queue 管理工具用户以及将访问该存储的客户机应用程序用户的相应访问权限。

表 8-2 文件系统对象存储库属性

| 属性 | 说明 |
|-----------------------------|---|
| java.naming.factory.initial | 文件系统对象存储库上 JNDI 查找的初始上下文： com.sun.jndi.fscontext. RefFSContextFactory |
| java.naming.provider.url | 目录路径信息。例如： file:///C:/myapp/mqobjs |

关于受管理对象的属性

Message Queue 受管理对象有两种基本类型：

- 连接工厂受管理对象由客户机应用程序用来创建与代理之间的连接。
- 目标受管理对象由客户机应用程序用来标识生产方向其发送消息或使用方从中检索消息的目标。

Message Queue 为创建和管理受管理对象提供了两个管理工具：命令行对象管理器实用程序 (`imqobjmgr`) 和 GUI 管理控制台。本章只介绍如何使用命令行。

受管理对象的属性是使用属性 - 值对指定的。

连接工厂属性

连接工厂配置将传送到连接工厂代表客户机应用程序所创建的所有连接。可以对连接进行配置以定义发送或接收消息的参与方，指定客户机运行时处理消息流的方式，以及自动为通过连接而发送的所有消息设置某些信息。

共有两种类型的连接工厂对象：

- `ConnectionFactory` 支持一般的消息传送和非分布式事务。
- `XAConnectionFactory` 支持分布式事务。

`ConnectionFactory` 和 `XAConnectionFactory` 对象共享同一属性集。

管理员或应用程序可以创建和配置连接工厂对象（用于建立原型或测试）。您可以使用 `imqobjmgr` 工具或管理控制台来设置连接工厂属性。

本节在以下部分中介绍连接工厂属性，各部分内容是按属性所影响的行为而编排的：

- [第 161 页 “连接处理”](#)
- [第 163 页 “客户机标识”](#)
- [第 164 页 “可靠性和流控制”](#)
- [第 165 页 “队列浏览器行为和服务器会话”](#)
- [第 166 页 “JMS 消息标题覆盖”](#)
- [第 165 页 “JMS 定义的属性支持”](#)

您最关心的属性是 `imqAddressList`，用它来指定客户机将与之建立连接的代理。[第 170 页 “添加连接工厂”](#) 说明了在对象存储库中添加连接工厂受管理对象时如何指定这些属性。

有关连接工厂属性的参考信息，请参见第 16 章“受管理对象的属性参考”和适用于 Message Queue 类 `com.sun.messaging.ConnectionConfiguration` 的 JavaDoc API 文档。

连接处理

可以使用连接处理属性指定要连接的消息服务器地址，如果请求重新连接，则指定尝试重新连接的次数以及尝试的时间间隔。

客户机连接到的消息服务器的地址为所指定的 `imqAddressList` 属性值。消息服务器地址包含代理主机名、端口号和连接服务类型。

端口号可以是代理端口映射器所在的端口，也可以是某一特定连接服务所在的端口。如果指定了端口映射器端口，则端口映射器会动态为连接分配端口号。有关指定消息服务器地址的完整信息，请参见第 307 页“`imqAddressList` 属性值的语法”。

自动重新连接

在单个代理环境或多代理群集环境中，可以将连接处理属性设置为使客户机在连接失败时自动与代理重新连接。也可以配置重新连接过程。

重新连接功能提供连接故障转移，但不提供数据故障转移：客户机重新连接到其他代理实例时，出现故障或断开的代理所保存的持久性消息和其他状态信息可能会丢失。

如果启用了自动重新连接，则连接失败时，Message Queue 会使临时目标保持持久，因为客户机可能会重新连接并再次访问它们。在客户机重新连接并使用这些目标后，代理会将其删除。

重新连接的处理方式取决于客户机是连接到单个代理，还是连接到群集中的某个代理上。以下部分介绍了其中的每一种可能性。

重新连接到单个代理 要在连接失败时使客户机自动重新连接到代理，请设置以下连接工厂属性：

- `imqReconnectEnabled`。启用自动重新连接行为。
- `imqReconnectAttempts`。指定客户机运行时尝试重新连接客户机的次数。
- `imqReconnectInterval`。指定客户机运行时两次尝试重新连接客户机之间所等待的时间。

有关这些属性的完整参考信息，请参见第 306 页“连接处理”。

重新连接到群集中的代理 在多代理群集环境中，如果为 `imqAddressList` 属性指定了多个地址，则自动重新连接会重复访问代理列表。列表中的所有代理必须均安装了 Message Queue Enterprise Edition。

如果客户机与列表中第一个地址的连接失败，则客户机运行时会尝试将该客户机重新连接到列表中的其他代理。如果该尝试失败，客户机运行时会连接列表中的其他代理，直到能重新连接客户机为止。

如果所有尝试都失败，客户机运行时会在指定的次数内循环访问列表，直到找到可用代理或者没有找到代理。`imqAddressListBehavior` 属性设置确定选择进行重新连接的代理是按地址列表中的地址顺序选择的，还是在列表中随机选择的。

要使客户机能够重新连接到群集中的代理，请使用以下属性：

- `imqReconnectEnabled`。启用自动重新连接行为。
- `imqReconnectAttempts`。指定在传送到下一个代理地址之前尝试各个代理地址的次数。
- `imqReconnectInterval`。指定两次尝试之间等待的时间。
- `imqAddressListIterations`。指定重复访问列表的次数。
- `imqAddressListBehavior`。指定连接尝试是按地址列表中地址的顺序，还是按随机顺序。

有关这些属性的完整参考信息，请参见第 306 页“连接处理”。

连接 Ping

`imqPingInterval` 属性指定客户机运行时对代理执行 Ping 操作的频率。通过定期测试连接，客户机运行时可尽早检测到失败的连接。如果 Ping 操作失败，客户机运行时会向客户机应用程序的异常侦听器对象抛出异常。如果应用程序没有异常侦听器，则应用程序下次尝试使用连接会失败。

对于等待接收消息而不发送消息的使用方客户机应用程序而言，Ping 的使用尤其重要。否则，此类应用程序不知道连接何时失败。生成不常见消息的客户机也能利用该功能，因为它能够在需要发送消息前处理失败的连接。

默认情况下，将 Ping 的时间间隔设置为 30 秒。值 -1 表示禁用 Ping 操作。

对断开连接的响应因操作系统而异。例如，在某些操作系统中，Ping 立即报告连接失败。其他操作系统可能会继续尝试建立代理连接，连续缓冲 Ping，直到 Ping 成功或缓冲区溢出为止。

有关 `imqPingInterval` 属性的完整参考信息，请参见第 306 页“连接处理”。

客户机标识

Message Queue 定义了一组连接工厂属性以支持客户机验证和唯一客户机 ID 设置（是长期订户所必需的）。

必须对尝试连接到代理的客户机进行验证。如果客户机在创建连接时未指定用户名或密码，则会出现以下情况之一：

- 如果未设置连接工厂属性 `imqDefaultUsername` 和 `imqDefaultPassword`，客户机运行时会将值 `guest/guest` 传递给代理，代理使用这些值来验证客户机。

用户系统信息库发行时包括条目 `guest/guest`，因此客户机将会获得连接。

- 如果设置了连接工厂属性 `imqDefaultUsername` 和 `imqDefaultPassword`，客户机运行时会将这些值传递给代理，代理使用这些值来验证客户机。

如果用户系统信息库中包含用户 / 密码对，则客户机就会获得连接。

此方案允许任何用户获得连接，这对于开发和测试非常方便。在生产系统中，只应为添加到用户系统信息库中的用户授予连接访问权限。

除了对请求连接的客户机进行代理验证外，JMS 规范还要求连接在必须保留客户机状态时提供唯一的客户机标识符。Message Queue 使用客户机 ID 来跟踪其长期订户。如果长期订户处于非活动状态，代理会为该订户保留消息，并在该订户再次处于活动状态时将消息传送给它。代理通过订户的客户机 ID 对其进行标识。

您可通过管理方式设置 `ClientID`，或者也可以通过客户机以编程方式对其进行设置。如果多个客户机从同一个连接工厂对象获取连接，则请设置该连接工厂的 `ClientID`。随后 Message Queue 可为从该工厂获取的每个连接提供唯一的 `ClientID`。

要确保唯一的 `ClientID` 值，请使用以下格式设置 `imqConfiguredClientID` 属性：

```
imqConfiguredClientID=${u}string
```

`${u}` 必须是该属性值的前四个字符。如果未遇到 "u"，在创建连接时会出现 JMS 异常。

`string` 的值是要与该连接工厂生成的连接相关联的任意值，例如 `Xconn`。在用户验证阶段，Message Queue 使用 `u:userName` 替代 `u`。例如，如果与连接关联的用户是 `Athena`，为连接指定的字符串是 `${u}Xconn`，则 `ClientID` 就是 `u:AthenaXconn`。

此方案确保连接工厂生成的每个连接（虽然它们在其他方面都是相同的）具有唯一的 `ClientID`。

在以下情况下，该方案不适用：如果两个客户机使用默认用户名（例如 `guest`）获得连接，则每个客户机的 `ClientID` 包含相同的组成部分 `{u}`。在运行时，第一个请求连接的客户机将获得连接；而第二个请求连接的客户机不会获得连接，因为 MQ 无法创建具有非唯一 `ClientID` 的连接。

可以设置 `imqDisableSetClientID` 属性，以禁止使用连接工厂的客户机通过编程方式更改已配置的客户机 ID。

除非应用程序代码使用 `setClientID()` 方法，否则必须为长期订阅设置 `imqConfiguredClientID` 属性。

下面概括介绍了影响客户机标识的属性：

- `imqDefaultUsername`。如果客户机创建连接时未指定用户名，此属性指定将用于验证代理的默认用户名。
- `imqDefaultPassword`。如果客户机在创建连接时未指定密码，此属性指定将用于验证代理的默认密码。
- `imqConfiguredClientID`。指定以管理方式配置的客户机 ID 的值。
- `imqDisableSetClientID`。指定使用连接工厂的客户机能否通过编程方式更改客户机 ID。

有关这些属性的完整参考信息，请参见第 309 页“客户机标识”。

可靠性和流控制

客户机收发的消息以及 Message Queue 使用的控制消息通过同一客户机 - 代理连接进行传送。因此，如果控制消息（例如代理确认）的传送被 JMS 消息的传送阻挡，则可能会发生延迟。

您可以设置允许您管理与客户机消息流相对的控制消息流的连接工厂属性。控制这两种类型的消息流需要考虑可靠性和吞吐量之间的折衷。有关如何使用这些属性来管理流控制和可靠性的讨论，请参见第 221 页“客户机运行时消息流调整”。

以下属性影响客户机消息流和控制消息流：

- `imqAckTimeout`。指定客户机运行时等待任何代理响应的最长时间（以毫秒为单位）。
- `imqConnectionFlowCount`。指定测量批中 JMS 消息的数量。
- `imqConnectionFlowLimitEnabled`。限制连接级别的消息流。
- `imqConnectionFlowLimit`。指定可以通过连接传送并在客户机运行时中缓冲（等待被使用）的消息数限制。

- `imqConsumerFlowLimit`。指定每个使用方可以通过连接传送并在客户机运行时中缓冲（等待被使用）的消息数限制。
- `imqConsumerFlowThreshold`。指定（以 `imqConsumerFlowLimit` 百分比的形式表示）每个使用方在客户机运行时中缓冲的消息数，低于此数值时将恢复使用方的消息传送。

有关这些属性的完整参考信息，请参见第 311 页“可靠性和流控制”。

队列浏览器行为和服务器会话

以下属性影响客户机队列浏览：

- `imqQueueBrowserMaxMessagesPerRetrieve`。指定客户机在浏览队列目标内容时一次检索的最大消息数。
- `imqQueueBrowserRetrieveTimeout`。指定客户机在浏览队列目标内容时等待检索消息的时间。
- `imqLoadMaxToServerSession`。对于 JMS 应用服务器工具，指定 `Message Queue Connection Consumer` 是将 `maxMessages` 数量的消息加载到 `ServerSession` 会话中，还是一次加载一条消息。

有关这些属性的完整参考信息，请参见第 312 页“队列浏览器行为和服务器会话”。

JMS 定义的属性支持

可以使用连接工厂属性自动在连接生成的消息上设置 JMS 定义的属性。JMS 属性是在 JMS 规范中定义的，位于 <http://java.sun.com/products/jms/docs.html>。

可以使用以下属性来设置 JMS 定义的属性：

- `imqSetJMSXUserID`。对于生成的消息，指定 `Message Queue` 是否设置了 JMS 定义的属性 `JMSXUserID`（发送消息的用户标识）。
- `imqSetJMSXAppID`。对于生成的消息，指定 `Message Queue` 是否设置 JMS 定义的属性 `JMSXAppID`（发送消息的应用程序标识）。
- `imqSetJMSXProducerTXID`。对于生成的消息，指定 `Message Queue` 是否设置 JMS 定义的属性 `JMSXProducerTXID`（生成消息的事务的标识）。
- `imqSetJMSXConsumerTXID`。对于使用的消息，指定 `Message Queue` 是否应该设置 JMS 定义的属性 `JMSXConsumerTXID`（使用消息的事务的标识）。
- `imqSetJMSXRcvTimestamp`。对于使用的消息，指定 `Message Queue` 是否应该设置 JMS 定义的属性 `JMSXRcvTimestamp`（将消息发送到使用方的时间）。

有关这些属性的完整参考信息，请参见第 313 页“JMS 定义的属性支持”。

JMS 消息标题覆盖

通过设置连接工厂的属性，可以覆盖 JMS 消息标题字段，这些字段指定了消息的持久性、生命周期以及优先级。这些设置用于由连接（从连接工厂获取的）所生成的所有消息。

可以覆盖以下 JMS 字段中的值：

- JMSDeliveryMode（消息持久性 / 非持久性）
- JMSExpiration（消息生命周期）
- JMSPriority（消息优先级——0 到 9 之间的整数）

有关上述字段的详细信息，请参见 JMS 规范，位于 <http://java.sun.com/products/jms/docs.html>。

因为覆盖消息标题可能会妨碍应用程序要求，所以只有在咨询了应用程序用户或设计者后才能使用此功能。

以下列表包含与消息覆盖有关的连接工厂属性。其中的大部分属性是成对的。在每一对属性中，第一个属性指定是否可以覆盖指定的标题字段，第二个属性指定覆盖值。

- `imqOverrideJMSDeliveryMode` 和 `imqJMSDeliveryMode`。第一个属性指定是否可以覆盖客户机设置的 `JMSDeliveryMode` 字段，第二个属性指定其覆盖值。
- `imqOverrideJMSExpiration` 和 `imqJMSExpiration`。第一个属性指定是否可以覆盖客户机设置的 `JMSExpiration` 字段，第二个属性指定其覆盖值。
- `imqOverrideJMSPriority` 和 `imqJMSPriority`。第一个属性指定是否可以覆盖客户机设置的 `JMSPriority` 字段，第二个属性指定其覆盖值。
- `imqOverrideJMSHeadersToTemporaryDestinations`。指定覆盖是否适用于临时目标。

有关这些属性的完整参考信息，请参见第 310 页“消息标题覆盖”。

目标受管理对象的属性

第 305 页表 16-1 列出了标识物理主题或队列目标的目标受管理对象的属性。第 171 页“添加主题或队列”一节说明了如何在向对象存储库中添加目标受管理对象时指定这些属性。

最重要的属性是 `imqDestinationName`。这是您为与主题或队列受管理对象相对应的物理目标指定的名称。您也可以为目标提供说明，帮助您将它与您创建以支持更多应用程序的其他目标区分开。

有关详细信息，请参见适用于 Message Queue 类 `com.sun.messaging.DestinationConfiguration` 的 JavaDoc API 文档。

使用对象管理器实用程序 (imqobjmgr)

对象管理器实用程序允许您创建和管理 Message Queue 受管理对象。使用此工具，您可以执行以下任务：

- 在对象存储库中添加或删除受管理对象。
- 列出现有的受管理对象。
- 查询和显示受管理对象的相关信息。
- 修改对象存储库中现有的受管理对象。

有关 `imqobjmgr` 命令的语法、子命令和选项的参考信息，请参见第 13 章“命令参考”。下一节介绍了使用任一 `imqobjmgr` 子命令时需要提供的信息。

所需的信息

执行大多数与受管理对象相关的任务时，必须指定以下信息作为 `imqobjmgr` 子命令的选项：

- **受管理对象的类型**

第 269 页表 13-11 显示了允许的类型。

- **受管理对象的 JNDI 查找名称：**

这是在客户机代码中表示对象存储库中的受管理对象（使用 JNDI）时使用的逻辑名。

- **受管理对象的属性**（`add` 和 `update` 子命令特别需要）：

- 对于目标：代理上的物理目标名称。这是使用 `imqcmd create dst` 子命令的 `-n` 选项指定的名称。如果不指定名称，将使用默认名称 `Untitled_Destination_Object`。
- 对于连接工厂：最常用的属性是地址列表 (`imqAddressList`)，用于指定客户机将试图连接的消息服务器地址（一个或多个）。如果不指定此信息，则使用本地主机和默认端口号 (7676)，这意味着客户机将试图与本地主机端口 7676 上的某个代理建立连接。第 170 页“添加连接工厂”一节说明了如何指定对象属性。

有关其他属性的详细信息，请参见第 160 页“连接工厂属性”。

- **对象存储库属性**

此信息取决于您使用的是文件系统存储还是 LDAP 服务器，但必须包括以下属性：

- JNDI 实现的类型（初始上下文属性）。例如，文件系统或 LDAP。
- 受管理对象在对象存储库中的位置（提供者 URL 属性），即它所在的“文件夹”。
- 访问对象存储库所需的用户名、密码和授权类型（如果有）。

有关对象存储库属性的详细信息，请参见第 158 页“LDAP 服务器对象存储库”和第 159 页“文件系统对象存储库”。

使用命令文件

imqobjmgr 命令允许您指定命令文件的名称，该文件使用 Java 属性文件语法表示全部或部分 imqobjmgr 子命令子句。

在对象管理器实用程序 (imqobjmgr) 中使用命令文件对于指定对象存储库属性特别有用，因为对象存储库属性可能在 imqobjmgr 的多次调用中都相同，并且通常需要多次键入。使用命令文件还可以避免一种情况，即避免超出命令行所允许的最大字符数。

imqobjmgr 命令文件的一般语法如下（版本属性反映命令文件的版本，而非 Message Queue 产品的版本，它不是一个命令行选项，它的值必须设置为 2.0）：

```
version=2.0
cmdtype=[ add | delete | list | query | update ]
obj.type=[ q | t | qf | tf | cf | xqf | xtf | xcf | e ]
obj.lookupName=lookup name
obj.attrs.objAttrName1=value1
obj.attrs.objAttrName2=value2
obj.attrs.objAttrNameN=valueN
...
objstore.attrs.objStoreAttrName1=value1
objstore.attrs.objStoreAttrName2=value2
objstore.attrs.objStoreAttrNameN=valueN
...
```

作为如何使用命令文件的一个示例，请考虑使用以下 imqobjmgr 命令：

```
imgobjmgr add
-t qf
-l "cn=myQCF"
-o "imgAddressList=mq://foo:777/jms"
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=img"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=img"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

此命令可以包含在一个文件中（例如 MyCmdFile 文件），其内容如下：

```
version=2.0
cmdtype=add
obj.type=qf
obj.lookupName=cn=myQCF
obj.attrs.imgAddressList=mq://foo:777/jms
objstore.attrs.java.naming.factory.initial=\
    com.sun.jndi.ldap.LdapCtxFactory
objstore.attrs.java.naming.provider.url=\
    ldap://mydomain.com:389/o=img
objstore.attrs.java.naming.security.principal=\
    uid=fooUser, ou=People, o=img
objstore.attrs.java.naming.security.credentials=fooPasswd
objstore.attrs.java.naming.security.authentication=simple
```

然后可以使用 -i 选项将该文件传递给对象管理器实用程序 (imgobjmgr)：

```
imgobjmgr -i MyCmdFile
```

也可以使用命令文件指定一些选项，而使用命令行指定其他选项。这样即可使用命令文件指定在实用程序的多次调用中都相同的那部分命令子句。例如，以下命令可以指定添加连接工厂受管理对象所需的所有选项，但不包括指定受管理对象存储库位置的选项。

```
imgobjmgr add
-t qf
```

```
-l "cn=myQCF"  
-o "imqAddressList=mq://foo:777/jms"  
-i MyCmdFile
```

在本例中，文件 MyCmdFile 将包含以下定义：

```
version=2.0  
objstore.attrs.java.naming.factory.initial=\  
    com.sun.jndi.ldap.LdapCtxFactory  
objstore.attrs.java.naming.provider.url=\  
    ldap://mydomain.com:389/o=imq  
objstore.attrs.java.naming.security.principal=\  
    uid=fooUser, ou=People, o=imq  
objstore.attrs.java.naming.security.credentials=fooPasswd  
objstore.attrs.java.naming.security.authentication=simple
```

可以从以下位置找到命令文件的其他示例：

```
/usr/demo/imq/imqobjmgr (Solaris)  
/opt/sun/mq/examples/imqobjmgr (Linux)  
IMQ_HOME/demo/imqobjmgr (Windows)
```

添加和删除受管理对象

本节说明如何在对象存储库中添加连接工厂和主题或队列目标的受管理对象。

注 对象管理器实用程序 (imqobjmgr) 只列出和显示了 Message Queue 受管理对象。如果对象存储库中包含与希望添加的受管理对象具有相同查找名称的非 Message Queue 对象，则您将在尝试执行添加操作时收到一条错误消息。

添加连接工厂

要使客户机应用程序获得与代理的连接，需要添加一个用来表示客户机应用程序所需连接类型的受管理对象：主题连接工厂或队列连接工厂

要添加队列连接工厂，请使用如下所示的命令：

```
imqobjmgr add
-t qf
-l "cn=myQCF"
-o "imqAddressList=mq://myHost:7272/jms"
-j "java.naming.factoryinitial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

上述命令创建一个查找名称为 `cn=myQCF` 的受管理对象，它连接到在 `myHost` 上运行的代理并在端口 7272 上执行侦听。受管理对象存储库在 LDAP 服务器上。通过指定一个命令文件作为 `imqobjmgr` 命令的参数，也可以实现此目的。有关详细信息，请参见第 168 页“使用命令文件”。

注 **命名约定：**如果使用 LDAP 服务器存储受管理对象，必须像上例那样指定一个前缀为 `"cn="` 的查找名称 (`cn=myQCF`)。可以通过 `-l` 选项指定查找名称。如果正在使用文件系统对象存储库，则不必使用 `cn` 前缀，不过，请不要使用带有 `"/` 的查找名称。请参见表 8-3。

表 8-3 命名约定示例

| 对象存储库类型 | 好的名称 | 禁用名称 |
|----------|-----------------------|--------------------------------|
| LDAP 服务器 | <code>cn=myQCF</code> | <code>myQCF</code> |
| 文件系统 | <code>myTopic</code> | <code>myObjects/myTopic</code> |

添加主题或队列

要使客户机应用程序能够访问代理上的物理目标，可以将标识这些目标的受管理对象添加到对象存储库中。

最好先创建物理目标，然后再将对应的受管理对象添加到对象存储库中。使用命令行实用程序 (imqcmd) 在代理上创建物理目标，这些目标由对象存储库中的目标受管理对象标识。有关创建物理目标的信息，请参见第 111 页“获得有关连接的信息”。

以下命令添加了一个标识主题目标的受管理对象，这个主题目标的查找名称为 myTopic，物理目标名称为 TestTopic。受管理对象存储库在 LDAP 服务器上。

```
imqobjmgr add
-t t
-l "cn=myTopic"
-o "imqDestinationName=TestTopic"
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

这是同一个命令，只不过受管理对象存储库在 Solaris 文件系统中：

```
imqobjmgr add
-t t
-l "cn=myTopic"
-o "imqDestinationName=TestTopic"
-j "java.naming.factory.initial=
    com.sun.jndi.fscontext.RefFSContextFactory"
-j "java.naming.provider.url=
    file:///home/foo/imq_admin_objects"
```

以 LDAP 服务器为例，可以使用命令文件 MyCmdFile 指定子命令子句。该文件将包含以下内容：

```
version=2.0
cmdtype=add
obj.type=t
obj.lookupName=cn=myTopic
obj.attrs.imqDestinationName=TestTopic
objstore.attrs.java.naming.factory.initial=
    com.sun.jndi.fscontext.RefFSContextFactory
```

```
objstore.attrs.java.naming.provider.url=
    file:///home/foo/img_admin_objects
objstore.attrs.java.naming.security.principal=
    uid=fooUser, ou=People, o=img
objstore.attrs.java.naming.security.credentials=fooPasswd
objstore.attrs.java.naming.security.authentication=simple
```

使用 `-i` 选项将该文件传递给 `imgobjmgr` 命令：

```
imgobjmgr -i MyCmdFile
```

注 如果使用 LDAP 服务器存储受管理对象，必须像上例那样指定一个前缀为 "cn=" 的查找名称。可以通过 `-l` 选项指定查找名称。如果使用文件系统对象存储库，则无需使用该前缀。

添加队列对象与此基本相同，不同之处在于要为 `-t` 选项指定 `q`。

删除受管理对象

可以使用 `delete` 子命令删除受管理对象。必须指定对象的查找名称、类型和地址。

以下命令删除了查找名称为 `cn=myTopic` 且存储在 LDAP 服务器上的主题的受管理对象。

```
imgobjmgr delete
-t t
-l "cn=myTopic"
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=img"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=img"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

列出受管理对象

使用 `list` 子命令可以获得所有受管理对象的列表，或者某个特定类型的所有受管理对象的列表。以下样例代码假定受管理对象存储库在 LDAP 服务器上。

以下命令将列出所有对象。

```
imgobjmgr list
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=imgq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imgq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

以下命令将列出类型为 `queue` 的所有对象。

```
imgobjmgr list
-t q
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=imgq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imgq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

获取有关单个对象的信息

使用 `query` 子命令可以获得有关一个受管理对象的信息。必须指定对象的查找名称，以及包含受管理对象的对象存储库的属性（如初始上下文和地址）。

在以下示例中，使用 `query` 子命令显示查找名称为 `cn=myTopic` 的对象的相关信息。

```
imgobjmgr query
-l "cn=myTopic"
-j "java.naming.factory.initial=
```

```
com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

更新受管理对象

可以使用 `update` 命令修改受管理对象的属性。必须指定对象的查找名称和地址。可以使用 `-o` 选项修改属性值。

此命令将改变表示主题连接工厂的受管理对象的属性：

```
imgobjmgr update
-t tf
-l "cn=MyTCF"
-o imgReconnectAttempts=3
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

更新受管理对象

使用代理群集

Message Queue Enterprise Edition 支持使用代理群集：一组协同工作为客户机提供消息传送服务的代理。通过使用群集，消息服务器可将客户机连接分布在多个代理上，从而调整对消息流量的操作。有关群集及其工作原理的概述，请参见 Message Queue 技术概述。

本章介绍如何管理代理群集、将其与代理连接以及对其进行配置。它包含以下各节：

- [第 178 页 “群集配置属性”](#)
- [第 179 页 “管理群集”](#)
- [第 182 页 “主代理”](#)

群集配置属性

可通过为群集中的每个成员代理指定群集配置属性来定义群集。可为群集中的每个代理单独设置这些属性，但是，通常将这些属性收集到所有代理参考的中心群集配置文件中会更方便一些。这样，可防止设置出现不一致的情况，确保群集中的所有代理都共享相同、一致的配置信息。

第 298 页表 14-11 中详细介绍了群集配置属性。其中包括以下内容：

- `imq.cluster.brokerlist` 给出了属于群集的所有代理的主机名和端口号。
- `imq.cluster.masterbroker` 指定跟踪状态变化的主代理（如果有）。
- `imq.cluster.url` 指定群集配置文件（如果有）的位置。
- `imq.cluster.hostname` 给出 `cluster` 连接服务（用于群集中代理之间的内部通信）的主机名或 IP 地址。如果有多个可用主机，该设置可能比较有用：例如，如果一台计算机中安装了多个网络接口卡。
- `imq.cluster.port` 给出 `cluster` 连接服务的端口号。
- `imq.cluster.transport` 指定 `cluster` 连接服务所使用的传输协议，例如 `tcp` 或 `ssl`。

可为每个代理单独设置 `hostname` 和 `port` 属性，但群集中所有代理的 `brokerlist`、`masterbroker`、`url` 和 `transport` 的值必须相同。

以下部分介绍了如何设置代理的群集配置属性，可以对群集中的每个代理单独进行设置，也可以使用群集配置文件集中进行设置。

为单个代理设置群集属性

可以在代理的实例配置文件（或在启动代理时的命令行）中设置其群集配置属性。例如，要创建由 `host1` 端口 9876 上的代理、`host2` 端口 5000 上的代理以及 `ctrlhost` 默认端口 (7676) 上的代理组成的群集，应在这三个代理的实例配置文件中包含以下属性：

```
imq.cluster.brokerlist=host1:9876,host2:5000,ctrlhost
```

请注意，如果需要更改群集配置，此方法要求更新群集中每个代理的实例配置文件。

使用群集配置文件

为保持一致和易于维护，建议将所有共享群集配置属性收集到一个群集配置文件中，而不是分别为每个代理设置这些属性。在此方法中，每个代理的实例配置文件必须将 `imq.cluster.url` 属性设置为指向该群集配置文件的位置：例如，

```
imq.cluster.url=file:/home/cluster.properties
```

然后群集配置文件定义群集中所有代理的共享配置属性，例如要连接的代理列表 (`imq.cluster.brokerlist`)、用于 `cluster` 连接服务的传输协议 (`imq.cluster.transport`) 以及可选的主代理地址 (`imq.cluster.masterbroker`)。以下代码定义了与前一实例中所定义的相同的群集，它将在 `ctrlhost` 上运行的代理作为主代理：

```
imq.cluster.brokerlist=host1:9876,host2:5000,ctrlhost
imq.cluster.masterbroker=ctrlhost
```

管理群集

本节介绍如何将一组代理连接成一个群集，在现有群集中添加新代理，以及从群集中删除代理。

连接代理

将代理连接到群集一般有两种方法：从命令行（使用 `-cluster` 选项）或通过群集配置文件中设置 `imq.cluster.brokerlist` 属性。无论使用哪种方法，启动的每个代理每隔五秒钟就尝试与其他代理连接；在主代理（如果配置了主代理）启动后，连接就会成功。如果群集中的某个代理在主代理之前启动，它将保持暂停状态，拒绝客户机连接，直到主代理启动为止；暂停的代理将自动进入正常运行状态。

在启动群集中的每个代理时，可以使用 `imqbrokerd` 命令的 `-cluster` 选项指定群集中代理的完整列表，而不使用群集配置文件。例如，以下命令启动一个新代理，并将其连接到在 `host1` 默认端口 (7676) 上运行的代理、`host2` 端口 5000 上运行的代理以及默认主机 (`localhost`) 端口 9876 上运行的代理：

```
imqbrokerd -cluster host1,host2:5000,:9876
```

有一种更适用于生产系统的替代方法，即创建一个群集配置文件，它使用 `imq.cluster.brokerlist` 属性来指定要连接的代理列表。然后群集中的每个代理必须将其自身的 `imq.cluster.url` 属性设置为指向该群集配置文件。

Linux 前提条件：设置 IP 地址

在 Linux 系统中，将代理连接为群集时有一个特殊的前提条件。某些 Linux 安装程序自动将 localhost 条目设置为网络回送 IP 地址 (127.0.0.1)。您必须设置系统的 IP 地址，以便为群集中的所有代理设置正确的地址。

对于加入群集的所有 Linux 系统，选中 /etc/hosts 文件作为群集设置的一部分。如果系统使用静态 IP 地址，请编辑 /etc/hosts 文件以指定 localhost 的正确地址。如果地址是使用域名服务 (DNS) 注册的，请编辑 /etc/nsswitch.conf 文件以更改条目顺序，使系统在查阅本地 hosts 文件前执行 DNS 查找。/etc/nsswitch.conf 文件中的读取顺序如下：

```
hosts:dns files
```

代理间的安全连接

如果要在群集中的代理之间传送安全、加密的消息，请配置 cluster 连接服务以使用基于 SSL 的传输协议。对于群集中的各个代理，请设置基于 SSL 的连接服务，如第 144 页“使用基于 SSL 的服务”中所述。然后，将每个代理的 `imq.cluster.transport` 属性设置为 `ssl`。可以在群集配置文件中设置，也可以为每个代理单独进行设置。

将代理添加到群集

在群集中添加新代理的步骤取决于该群集是否使用群集配置文件。

► 使用群集配置文件在群集中添加新代理

1. 将新代理添加到群集配置文件中的 `imq.cluster.brokerlist` 属性。
2. 对群集中的各个代理执行以下命令：

```
imqcmd reload cls
```

该命令强制每个代理重新加载群集配置，确保群集中代理的所有持久性信息都是最新的。

3. (可选) 在代理的 `config.properties` 文件中将 `imq.cluster.url` 属性的值设置为指向群集配置文件。
4. 启动新代理。

如果未执行步骤 3，请使用 `imqbrokerd` 命令行中的 `-D` 选项设置 `imq.cluster.url` 的值。

► 不使用群集配置文件向群集中添加新代理

通过编辑 `config.properties` 文件或者使用 `imqbrokerd` 命令行中的 `-D` 选项设置以下属性的值：

- `imq.cluster.brokerlist`
- `imq.cluster.masterbroker`（如有必要）
- `imq.cluster.transport`（如果使用安全的 `cluster` 连接服务）

从群集中删除代理

从群集中删除代理所使用的方法取决于最初是通过命令行创建群集，还是通过中央群集配置文件创建群集的。

使用命令行删除代理

如果在命令行中使用 `imqbrokerd` 命令将代理连接为群集，则必须停止每个代理，然后重新启动它们，以便在命令行中指定一组新的群集成员。过程如下所示：

► 使用命令行从群集中删除代理

1. 使用 `imqcmd` 命令停止群集中的每个代理。
2. 请使用 `imqbrokerd` 命令的 `-cluster` 选项仅指定那些要保留的代理，来重新启动要在群集中保留的代理。

例如，假设最初创建的群集包括代理 A、B 和 C，而且这三个代理是用以下命令启动的：

```
imqbrokerd -cluster A,B,C
```

要从群集中删除代理 A，请使用以下命令重新启动代理 B 和 C：

```
imqbrokerd -cluster B,C
```

使用群集配置文件删除代理

如果最初创建群集时，在中央群集配置文件中使用 `imq.cluster.brokerlist` 属性指定其成员代理，则不必停止这些代理以删除其中一个代理。相反，只需编辑配置文件来排除要删除的代理，强制群集中的其他代理成员重新加载群集配置，并重新配置被排除的代理，以使它不再指向同一群集配置文件。过程如下：

► 使用群集配置文件从群集中删除代理

1. 编辑群集配置文件，从为 `imq.cluster.brokerlist` 属性指定的列表中删除被排除的代理。

2. 对保留在群集中的各个代理执行以下命令：

```
imqcmd reload cls
```

该命令强制代理重新加载群集配置。
3. 停止从群集中删除的代理。
4. 请编辑该代理的 `config.properties` 文件，删除 `imq.cluster.url` 属性的值或为其指定其他值。

主代理

群集可选择包含一个主代理，它维护配置更改记录以跟踪群集持久性状态所发生的任何变化。主代理是使用 `imq.cluster.masterbroker` 配置属性标识的。该属性位于群集配置文件或者单个代理的实例配置文件中。

配置更改记录包含与群集关联的持久性实体的相关更改信息，例如长期订阅和管理员创建的物理目标。群集中的所有代理在启动时都会咨询主代理，以便更新与这些持久性实体相关的信息。如果主代理出现故障，就不会实现上述同步；有关详细信息，请参见第 183 页“主代理不可用时”。

管理配置更改记录

因为配置更改记录中包含的信息很重要，所以要定期进行备份以便在出现故障时能够恢复，这一点非常重要。虽然从备份恢复会丢失群集持久性状态自备份以后发生的任何更改，但经常进行备份可以将信息丢失的可能性降到最低。备份和恢复操作也有利于压缩和优化配置更改记录中的更改历史，因为该记录会随时间迅速增加。

► 备份配置更改记录

使用 `imqbrokerd` 命令的 `-backup` 选项指定备份文件的名称。例如：

```
imqbrokerd -backup mybackuplog
```

► 恢复配置更改记录

1. 关闭群集中的所有代理。
2. 使用以下命令从备份文件中恢复主代理的配置更改记录

```
imqbrokerd -restore mybackuplog
```

3. 如果为主代理分配了新的名称或端口号，请在群集配置文件中相应地更新 `imq.cluster.brokerlist` 和 `imq.cluster.masterbroker` 属性。

4. 重新启动群集中的所有代理。

主代理不可用时

因为群集中的所有代理都需要利用主代理来执行持久性操作，在主代理不可用时，群集中任一代理的以下 `imgcmd` 子命令将返回一条错误信息：

- `create dst`
- `destroy dst`
- `update dst`
- `destroy dur`

自动创建的物理目标和临时目标均不受影响。

在没有主代理时，任何尝试创建长期订户或取消长期订阅的客户机应用程序都会收到一条错误信息。但是，客户机可以成功地指定现有长期订阅，并与其进行交互。

主代理

监视消息服务器

本章介绍可用来监视消息服务器的工具，以及如何获得度量数据。本章包含以下小节：

- 第 185 页 “监视工具简介”
- 第 187 页 “配置和使用代理日志记录”
- 第 192 页 “以交互方式显示度量”
- 第 196 页 “编写应用程序来监视代理”

有关特定度量的参考信息可从第 18 章 “度量参考” 获得。

监视工具简介

Message Queue 信息有三种监视接口：日志文件、交互命令和可获得度量的客户机 API。每一种接口都有各自的优点和缺点，如下所示：

- 日志文件提供长期的度量数据记录，但不易于解析。
- 命令使您可以根据需要快速地调整示例信息，但不能查看历史信息或通过编程方式操纵数据。
- 客户机 API 使您可以提取信息、对其进行处理、操纵数据、提供图形或发送警报。但是，要使用它，您必须编写自定义应用程序以捕获及分析数据。

表 10-1 比较了各种不同的工具。

表 10-1 度量监视工具的益处和限制

| 度量 监视工具 | 益处 | 限制 |
|----------------|--|---|
| imqcmd metrics | 远程监视 便于抽样检查 在命令选项中设置报告的时间间隔，可以随时更改 易于选择感兴趣的特定数据 数据以简易的表格格式提供 | 单条命令无法获取所有数据 难于以编程方式分析数据 不能创建历史记录 难于看到历史趋势 |
| 日志文件 | 定期采样 创建历史记录 | 需要配置代理属性；必须关闭并重新启动代理才能生效 仅限本地监视 数据格式非常难于读取或解析；没有解析工具 报告时间间隔不能随时更改；所有度量数据也是如此 在数据的选择上不能提供灵活性 仅限代理度量；目标和连接服务度量不包括在内 如果时间间隔过短，性能可能会受影响 |
| 客户机 API | 远程监视 易于选择感兴趣的特定数据 数据可以通过编程方式分析，并能够以任何格式提供 | 需要配置代理属性；必须关闭并重新启动代理才能生效 需要编写您自己的度量监视客户机 报告时间间隔不能随时更改；所有度量数据也是如此 |

除了表中给出的不同外，每种工具收集到的由代理生成的度量信息的子集也稍有不同。有关每种监视工具收集的度量数据的信息，请参见第 321 页第 18 章“度量参考”。

配置和使用代理日志记录

Message Queue 记录程序获取代理代码、调试器和度量生成器等生成的信息，并将该信息写到多个输出通道：标准输出（控制台）、日志文件以及 Solaris™ 操作系统上的 syslog 守护程序进程。

您可以指定记录程序所收集的信息类型以及写到每个输出通道的类型。特别地，您可以指定希望将度量信息写入到日志文件。

本节介绍了代理的默认日志记录配置，并说明了如何将日志信息重定向到替代输出通道，如何更改日志文件转移条件，以及如何将度量数据发送到日志文件。

默认日志记录配置

代理被自动配置为将日志输出保存到一组滚动的日志文件中。这些日志文件位于由关联代理的实例名称标识的目录中（请参见附录 A “Message Queue 数据在特定操作系统中的位置”）：

```
.../instances/instanceName/log/
```

日志文件是纯文本文件。它们的名称如下（从最早生成到最近生成）：

```
log.txt
log_1.txt
log_2.txt
...
log_9.txt
```

默认情况下，日志文件每星期转移一次；系统维护九个备份文件。

- 要更改保存日志文件的目录，请将 `imq.log.file.dirpath` 属性设置为所需的路径。
- 要将日志文件的根名称由 `log` 改为其他名称，请设置 `imq.log.file.filename` 属性。

代理支持三种日志级别：ERROR、WARNING 和 INFO。表 10-2 说明了每一种级别。

表 10-2 日志记录级别

| 级别 | 说明 |
|---------|-------------------|
| ERROR | 指出可能导致系统故障的问题的消息。 |
| WARNING | 需要注意但不会导致系统故障的警报。 |
| INFO | 度量及其他信息性消息的报告。 |

设置日志记录级别后，将收集所有高于和等于该级别的消息。默认的日志级别是 INFO，因此，默认情况下会记录所有 ERROR、WARNING 和 INFO 消息。

日志消息格式

记录的消息由时间戳、消息代码和消息本身构成。信息的大小取决于所设置的日志级别。以下是一个 INFO 消息的示例。

```
[13/Sep/2000:16:13:36 PDT] B1004 Starting the broker service using tcp [
25374,100] with min threads 50 and max threads of 500
```

要更改时间戳时区，请参见有关 `imq.log.timezone` 属性的信息，[第 295 页表 14-10](#) 中介绍了该信息。

更改记录程序配置

[第 295 页表 14-10](#) 中介绍了与日志相关的属性。

► 更改代理的记录程序配置

1. 设置日志级别。
2. 设置一个和多个日志记录种类的输出通道（文件、控制台或两者）。
3. 如果将输出记录到文件中，请为该文件配置转移条件。

可以通过设置记录程序属性完成上述步骤。可以使用以下两种方法之一来进行设置：

- 启动代理之前，在该代理的 `config.properties` 文件中更改或添加记录程序属性。
- 在启动代理的 `imqbrokerd` 命令中指定记录程序命令行选项。也可以使用代理选项 `-D` 来更改记录程序属性（或任何代理属性）。

在命令行上传送的选项将覆盖代理实例配置文件中指定的属性。[表 10-3](#) 列出了影响日志记录的 `imqbrokerd` 选项。

表 10-3 imqbrokerd 记录程序选项及相应的属性

| imqbrokerd 选项 | 说明 |
|--------------------------------|--|
| <code>-metrics interval</code> | 指定度量信息写入记录程序的时间间隔（以秒为单位）。 |
| <code>-loglevel level</code> | 将日志级别设置为以下三项之一：ERROR、WARNING 和 INFO。 |
| <code>-silent</code> | 停止向控制台记录日志信息。 |
| <code>-tty</code> | 将所有消息发送到控制台。默认情况下，只显示 WARNING 和 ERROR 级别的消息。 |

以下小节介绍了如何更改默认配置，以执行以下操作：

- 更改输出通道（日志消息的目标）
- 更改转移条件

更改输出通道

默认情况下，错误和警告消息除了记录到日志文件中以外，还会显示在终端上。（在 Solaris 中，错误消息还会写入到系统的 syslog 守护程序中。）

可以用以下方式更改日志消息的输出通道：

- 要在屏幕上显示所有的日志种类（对于给定的级别）输出，请在 `imqbrokerd` 命令中使用 `-tty` 选项。
- 要阻止在屏幕上显示日志输出，请在 `imqbrokerd` 命令中使用 `-silent` 选项。
- 使用 `imq.log.file.output` 属性指定将哪些种类的日志记录信息写入到日志文件。例如，


```
imq.log.file.output=ERROR
```
- 使用 `imq.log.console.output` 属性指定将哪些种类的日志记录信息写入到控制台。例如，


```
imq.log.console.output=INFO
```
- 在 Solaris 中，使用 `imq.log.syslog.output` 属性指定将哪些种类的日志记录信息写入到 Solaris syslog 中。例如，


```
imq.log.syslog.output=NONE
```

注 在更改记录程序输出通道前，必须确保日志记录级别已设置为支持映射到输出通道的信息。例如，如果将日志级别设置为 `ERROR`，然后将 `imq.log.console.output` 属性设置为 `WARNING`，那么将不会记录任何消息，因为没有启用对 `WARNING` 消息的日志记录。

更改日志文件转移条件

转移日志文件有两个条件：时间和大小。默认是使用时间条件，每七天转移一次文件。

- 要更改时间间隔，需要更改 `imq.log.file.rolloversecs` 属性。例如，以下属性定义将时间间隔更改为十天：

```
imq.log.file.rolloversecs=864000
```

- 要将转移条件更改为取决于文件大小，需要设置 `imq.log.file.rolloverbytes` 属性。例如，以下定义将在文件达到 500,000 字节的限制时将代理定向到转移文件。

```
imq.log.file.rolloverbytes=500000
```

如果同时设置与时间相关和与大小相关的转移属性，首先达到的限制将触发转移。如前所述，代理最多维护九个转移文件。

可以在代理运行时设置或更改日志文件转移属性。要设置这些属性，请使用 `imqcmd update bkr` 命令。

将度量数据发送到日志文件

本节讲述使用代理日志文件报告度量信息的过程。有关配置记录程序的一般信息，请参见第 187 页“配置和使用代理日志记录”。

► 要使用日志文件来报告度量信息，请执行下列操作：

1. 配置代理的度量生成功能：

- a. 确认 `imq.metrics.enabled=true`

默认情况下，用于日志记录的度量生成功能会打开。

- b. 将度量生成时间间隔设置为比较合适的秒数。

```
imq.metrics.interval=interval
```

此值可以在 `config.properties` 文件中设置，或者在启动代理时使用 `-metrics interval` 命令行选项进行设置。

2. 确认记录程序是否收集度量信息：

```
imq.log.level=INFO
```

这是默认值。此值可以在 `config.properties` 文件中设置，或者在启动代理时使用 `-loglevel level` 命令行选项进行设置。

3. 确认记录程序是否已设置为将度量信息写入日志文件：

```
imq.log.file.output=INFO
```

这是默认值。它可以在 `config.properties` 文件中设置。

4. 启动代理。

下面显示了输出到日志文件的样例代理度量：

```
[21/Jul/2004:11:21:18 PDT]
连接: 0      JVM 堆: 8323072 字节 (7226576 可用) 线程: 0 (14-1010)
  流入: 0 个消息 (0 个字节) 0 个包 (0 个字节)
  流出: 0 个消息 (0 个字节) 0 个包 (0 个字节)
  流入速率: 0 个消息 / 秒 (0 个字节 / 秒) 0 个包 / 秒 (0 个字节 / 秒)
  流出速率: 0 个消息 / 秒 (0 个字节 / 秒) 0 个包 / 秒 (0 个字节 / 秒)
```

有关度量数据的参考信息，请参见第 18 章“度量参考”。

记录停用消息

可以通过启用代理的停用消息日志记录来监视物理目标。无论是否使用停用消息队列，都可以记录停用消息。

如果启用了停用消息日志记录，代理会记录以下类型的事件：

- 物理目标超出其大小的最大值。
- 代理由于如下原因从物理目标中删除消息：
 - 到达目标大小限制。
 - 消息的有效期到期。
 - 消息太大。
 - 代理试图处理消息时出现错误。

如果使用了停用消息队列，日志记录还会包括以下类型的事件：

- 代理将消息移动到停用消息队列中。

- 代理从停用消息队列中删除某条消息并将其废弃。

默认情况下，禁用停用消息日志记录。要启用它，请设置代理属性 `imq.destination.logDeadMsgs`。

以交互方式显示度量

Message Queue 代理可以报告以下类型的度量：

- **Java 虚拟机 (JVM) 度量。**有关 JVM 堆大小的信息。
- **代理范围度量。**有关代理中存储的消息、流入或流出代理的消息以及内存使用的信息。按消息数和字节数跟踪消息。
- **连接服务度量。**有关连接和连接线程资源的信息，以及有关特定连接服务的消息流的信息。
- **目标度量。**有关流入和流出特定物理目标的消息的信息、有关物理目标的使用方的消息以及有关内存和磁盘空间使用情况的信息。

`imqcmd` 命令可以为整体代理、单个连接服务以及单个物理目标获取度量信息。要获得度量数据，通常应该使用 `imqcmd` 的 `metrics` 子命令。度量数据将按照您指定的时间间隔或者次数写到控制台屏幕上。

也可以使用 `query` 子命令查看也包含配置信息的类似数据。有关详细信息，请参见第 196 页“[imqcmd query](#)”。

imqcmd metrics

`imqcmd metrics` 的语法和选项分别如表 10-4 和表 10-5 所示。

表 10-4 `imqcmd metrics` 子命令语法

| 子命令语法 | 提供的度量数据 |
|---|----------------------------------|
| <pre>metrics bkr [-b hostName:port] [-m metricType] [-int interval] [-msp numSamples]</pre> | 显示默认代理的代理度量，或显示指定主机和端口上的代理的代理度量。 |
| 或 | |

表 10-4 imqcmd metrics 子命令语法（续）

| 子命令语法 | 提供的度量数据 |
|---|-----------------------------|
| <pre>metrics svc -n <i>serviceName</i> [-b <i>hostName:port</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</pre> | 显示默认代理或指定主机和端口上的代理中指定服务的度量。 |
| 或 | |
| <pre>metrics dst -t <i>destType</i> -n <i>destName</i> [-b <i>hostName:port</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</pre> | 显示指定类型和名称的物理目标的度量信息。 |

表 10-5 imqcmd metrics 子命令选项

| 子命令选项 | 说明 |
|-------------------------|--|
| -b <i>hostName:port</i> | 指定度量数据所报告的代理的主机名和端口。默认为 localhost:7676 |
| -int <i>interval</i> | 指定显示度量的时间间隔（以秒为单位）。默认值为 5 秒。 |
| -m <i>metricType</i> | 指定要显示的度量类型。 ttl 显示流入和流出代理、服务或目标的消息和包的度量（默认度量类型） rts 显示消息和包流入和流出代理、连接服务或目标的速率（每秒）的度量 cxn 显示连接、虚拟内存堆和线程（仅适用于代理和连接服务） con 显示与使用方相关的度量（仅适用于目标） disk 显示磁盘使用情况度量（仅适用于目标） |
| -msp <i>numSamples</i> | 指定输出中显示的样例数。默认值为无限数（无穷大）。 |
| -n <i>destName</i> | 指定要报告度量数据的物理目标（如果有）的名称。没有默认值。 |
| -n <i>serviceName</i> | 指定要报告度量数据的连接服务（如果有）。没有默认值。 |

表 10-5 imqcmd metrics 子命令选项（续）

| 子命令选项 | 说明 |
|-------------------|--------------------------------------|
| -t <i>destTyp</i> | 指定要报告度量数据的物理目标（如果有）的类型（队列或主题）。没有默认值。 |

使用 metrics 子命令显示度量数据

本节讲述使用 metrics 子命令报告度量信息的过程。

► **要使用 metrics 子命令，请执行下列操作：**

1. 启动度量信息需要的代理。

请参见第 64 页“以交互方式启动代理”。

2. 发出相应的 imqcmd metrics 子命令和选项，如表 10-4 和表 10-5 中所示。

度量输出：imqcmd metrics

本节包含 imqcmd metrics 子命令的输出示例。这些示例显示了代理范围、连接服务和物理目标度量。

代理范围度量

要获取在 10 秒的时间间隔内流入和流出代理的消息和包的速率，请使用 metrics bkr 子命令：

```
imqcmd metrics bkr -m rts -int 10 -u admin
```

此命令产生的输出类似如下内容（请参见第 322 页表 18-2 中的数据说明）：

| 消息 / 秒 | | 消息字节 / 秒 | | 包 / 秒 | | 包字节 / 秒 | |
|--------|----|----------|------|-------|----|---------|------|
| 流入 | 流出 | 流入 | 流出 | 流入 | 流出 | 流入 | 流出 |
| 0 | 0 | 27 | 56 | 0 | 0 | 38 | 66 |
| 10 | 0 | 7365 | 56 | 10 | 10 | 7457 | 1132 |
| 0 | 0 | 27 | 56 | 0 | 0 | 38 | 73 |
| 0 | 10 | 27 | 7402 | 10 | 20 | 1400 | 8459 |
| 0 | 0 | 27 | 56 | 0 | 0 | 38 | 73 |

连接服务度量

要获取 `jms` 连接服务所处理的消息和包的累计总数，请使用 `metrics svc` 子命令：

```
imqcmd metrics svc -n jms -m ttl -u admin
```

此命令产生的输出类似如下内容（请参见第 324 页表 18-3 中的数据说明）：

```
-----
      消息           消息字节       包       包字节
      流入  流出   流入  流出   流入  流出   流入  流出
-----
    164  100 120704 73600  282  383 135967 102127
    657  100 483552 73600  775  876 498815 149948
-----
```

物理目标度量

要获取有关物理目标的度量信息，请使用 `metrics dst` 子命令：

```
imqcmd metrics dst -t q -n XQueue -m ttl -u admin
```

此命令产生的输出类似如下内容（请参见第 326 页表 18-4 中的数据说明）：

```
-----
      消息           消息字节       消息计数       消息总字节数 (k)       最大
      流入  流出   流入  流出   当前  峰值  平均值   当前  峰值  平均值   消息 (k)
-----
    200  200 147200 147200     0   200   0     0   143   71     0
    300  200 220800 147200   100  200  10     71  143   64     0
    300  300 220800 220800     0   200   0     0   143   59     0
-----
```

要获取有关物理目标的使用方的信息，请使用下面的 `metrics dst` 子命令：

```
imqcmd metrics dst -t q -n SimpleQueue -m con -u admin
```

此命令产生的输出类似如下内容（请参见第 326 页表 18-4 中的数据说明）：

```
-----
      活动使用方           备份使用方           消息计数
      当前  峰值  平均值   当前  峰值  平均值   当前  峰值  平均值
-----
    1     1     0     0     0     0     944 1000  525
-----
```

imqcmd query

imqcmd query 命令的语法和选项如表 10-6 中所示，并带有该命令所提供的度量数据的说明。

表 10-6 imqcmd query 子命令语法

| 子命令语法 | 提供的度量数据 |
|--|---|
| <pre>query bkr [-b <i>hostName:port</i>]</pre> | 有关当前存储在代理内存和持久性存储中的消息个数和消息字节数目的信息（请参见第 102 页“显示代理信息”） |
| 或 | |
| <pre>query svc -n <i>serviceName</i> [-b <i>hostName:port</i>]</pre> | 有关指定的连接服务的当前已分配线程数和连接数的信息（请参见第 108 页“显示连接服务信息”） |
| 或 | |
| <pre>query dst -t <i>destType</i> -n <i>destName</i> [-b <i>hostName:port</i>]</pre> | 有关指定目标的当前生产方的数目、活动和备份使用方的数目以及存储在内存和持久性存储中的消息数和消息字节数的信息（请参见第 121 页“显示有关物理目标的信息”） |

注 由于 imqcmd query 提供的度量数据有限，第 321 页第 18 章“度量参考”的表中未提供此工具。

编写应用程序来监视代理

Message Queue 提供度量的监视功能，通过该功能，代理可以将度量数据写入到 JMS 消息，然后根据消息中包含的度量信息类型将它们发送到众多度量主题目标中的某一个目标。

您可以通过编写具有下列功能的客户机应用程序来访问此度量信息：订阅度量主题目标、使用这些目标中的消息以及处理消息中包含的度量信息。

共有五个度量主题目标，它们的名称以及传送到各个目标的度量消息的类型显示在表 10-7 中。

表 10-7 度量主题目标

| 主题名称 | 度量消息类型 |
|--|----------------|
| mq.metrics.broker | 代理度量 |
| mq.metrics.jvm | Java 虚拟机度量 |
| mq.metrics.destination_list | 目标及其类型列表 |
| mq.metrics.destination.queue. <i>monitoredDestinationName</i> | 具有指定名称的队列的目标度量 |
| mq.metrics.destination.topic. <i>monitoredDestinationName</i> | 具有指定名称的主题的目标度量 |

设置基于消息的监视

本节讲述使用基于消息的监视功能来收集度量信息的过程。该过程包括客户机开发和管理任务这两方面。

► **要设置基于消息的监视，请执行下列操作：**

1. 编写度量监视客户机。

有关以编程方式编写订阅度量主题目标、使用度量消息并从这些消息中提取度量数据的客户机的说明，请参见 *Message Queue Developer's Guide for Java Clients*。

2. 通过在 `config.properties` 文件中设置代理属性来配置代理的度量消息生产方：

a. 启用度量消息生成。

设置 `imq.metrics.topic.enabled=true`

默认值为 `true`。

b. 设置生成度量消息的时间间隔（以秒为单位）。

设置 `imq.metrics.topic.interval=interval`

默认值为 60 秒。

c. 指定是否需要让度量消息成为持久性的消息（即，当代理发生故障时它们是否还能保留）。

设置 `imq.metrics.topic.persist`

默认值为 `false`。

- d. 指定在删除度量消息之前它们能在各自的目标中保留的时间。

设置 `mq.metrics.topic.timetolive`

默认值为 300 秒。

3. 设置需要对度量主题目标设置的任何访问控制。

请参见下面的“[安全性和访问注意事项](#)”中的讨论。

4. 启动度量监视客户机。

当使用方订阅度量主题时，系统就会自动创建度量主题目标。创建度量主题后，代理的度量消息生产方就会开始向度量主题发送度量消息。

安全性和访问注意事项

出于以下两个原因，需要限制对度量主题目标的访问：

- 度量数据可能包含有关代理及其资源的敏感信息
- 对度量主题目标的过多订阅可能会增大代理开销，从而给性能带来负面影响

出于这些方面的考虑，建议限制对度量主题目标的访问。

监视客户机的验证和授权控制与其他任何客户机都相同。只有保持在 `Message Queue` 用户系统信息库中的用户才允许连接代理。

您可以通过某个访问控制属性文件，限制对特定度量主题目标的访问，从而提供其他保护，如第 138 页“[授权用户：访问控制属性文件](#)”中所述。

例如，`accesscontrol.properties` 文件中的下列条目将拒绝除 `user1` 和 `user2` 之外的其他所有用户对 `mq.metrics.broker` 度量主题的访问。

```
topic.mq.metrics.broker.consume.deny.user=*
topic.mq.metrics.broker.consume.allow.user=user1,user2
```

下列条目将仅允许用户 `user3` 监视主题 `t1`。

```
topic.mq.metrics.destination.topic.t1.consume.deny.user=*
topic.mq.metrics.destination.topic.t1.consume.allow.user=user3
```

根据度量数据的敏感度不同，您也可以使用加密连接将度量监视客户机连接至代理。有关使用加密连接的信息，请参见第 144 页“使用基于 SSL 的服务”。

度量输出：度量消息

使用基于消息的监视 API 获得的度量数据输出是您编写的度量监视客户机的一个函数。您只会受到代理中度量生成器提供的数据的限制。有关此数据的完整列表，请参见第 321 页“度量参考”。

编写应用程序来监视代理

分析和调整消息服务

本章包括大量有关如何分析和调整 Message Queue 服务以优化消息传送应用程序性能的主题。包括下列主题：

- [第 201 页 “关于性能”](#)
- [第 204 页 “影响性能的因素”](#)
- [第 216 页 “调整配置以提高性能”](#)

关于性能

本节提供有关性能调整的一些背景信息。

性能调整过程

您的消息传送应用程序的性能取决于该应用程序与 Message Queue 服务之间的交互。因此，若要获得最佳性能，则需要应用程序开发者和管理员的共同努力。

优化性能的过程从应用程序设计开始，一直持续到部署应用程序之后对消息服务的调整阶段。性能调整过程包括下列阶段：

- 定义应用程序的性能要求
- 设计的应用程序要考虑到影响性能的因素（特别是可靠性和性能之间的折衷）
- 建立性能衡量基线
- 调整或重新配置消息服务以优化性能。

上述过程常常需要反复进行。在应用程序的部署过程中，Message Queue 管理员应该评估消息服务器是否适用于应用程序的总体性能要求。如果基准检验测试符合这些要求，管理员就可以按本章中所述调整系统。但是，如果基准检验测试不符合性能要求，则可能需要重新设计应用程序或者修改部署体系结构。

性能方面

通常，性能是对消息服务将消息从生产方传送到使用方时的速度和效率的一种衡量。但是，取决于您的需要，可能会有几个不同的性能方面对您特别重要。

连接负荷 系统所能支持的消息生产方、消息使用方或并行连接的数量。

消息吞吐量 每秒钟能通过消息传送系统抽取的消息数或消息字节数。

延迟 特定消息从消息生产方传送到消息使用方所需的时间。

稳定性 消息服务的总体可用性，或者当负荷较重或发生故障时性能的下降程度。

效率 消息的传送效率，这是一种与使用的计算资源相关的消息吞吐量的衡量。

这些不同的性能方面总而言之是相辅相成的。如果消息吞吐量很高，则意味着消息在消息服务器上作为待办事项累积的可能性就会变小，因此延迟时间也应该较低（单条消息可以很快地传送）。但是，延迟可能取决于许多因素：通信链接的速度、消息服务器的处理速度和客户机的处理速度，以及其他众多方面。

在任何情况下，性能都有几个不同方面。哪些方面对您最重要？这通常要取决于特定应用程序的要求。

基准检验

基准检验是为消息传送应用程序创建测试套件，并针对此测试套件衡量消息吞吐量或其他性能方面的一种过程。

例如，您可以创建这样一个测试套件：一定数量的生产方客户机使用一定数量的连接、会话和消息生成器，将标准大小的持久性或非持久性消息按照某个特定速率发送至队列或主题（这完全取决于消息传送应用程序的设计）。同样，测试套件中还包括一定数量的使用方客户机，它们使用一定数量的连接、会话和特定类型的消息使用者（通过特定的确认模式来使用测试套件的物理目标中的消息）。

使用标准的测试套件可以衡量消息的生成和使用之间所花的时间或者消息的平均吞吐速率，您还可以监视系统以观察连接线程使用情况、消息存储数据、消息流数据以及其他相关度量。这样就可以提高消息的生成速率、消息生产方的数量或者其他变量，直到性能受到负面影响为止。可能达到的最大吞吐量就是消息服务配置的基准。

使用此基准可以修改测试套件的某些特征。控制所有可能影响性能的因素时请小心（请参见第 206 页“影响性能的应用程序设计因素”），您可以记录这些因素导致基准发生了哪些变化。例如，您可以将连接数或消息大小增加五倍或十倍，并记下对性能产生的影响。

相反，您也可以将基于应用程序的因素保持不变，而以某种控制方式更改代理配置（例如，更改连接属性、线程池属性、JVM 内存限制、限制行为、内置与插入的持久性等），并记下这些更改如何影响性能。

当您需要通过调整消息服务来增加所部署的应用程序的性能时，这种应用程序基准检验可以提供非常有价值的信息。基准检验可用于更准确地预测所作的更改或一系列更改产生的影响。

通常，基准检验应该在受控制的测试环境下运行，并且运行足够长的时间，以使消息服务能够稳定。（性能在启动时会受到实时编译的负面影响，因为要将 Java 代码转换为机器代码。）

基线使用模式

部署并运行消息传送应用程序后，建立基线使用模式是很重要的。您需要知道何时发生峰值需求，并且需要能够量化该需求。例如，需求通常会随最终用户的数量、活动级别、一天当中的时间，或者以上所有这些因素而发生波动。

要建立基线使用模式，您需要在一段较长的时间内监视消息服务器，来查看如下数据：

- 连接数
- 在代理（或特定物理目标）中存储的消息数
- 流入和流出代理（或特定物理目标）的消息
- 活动使用方数

还可以使用度量数据中提供的平均值和峰值。

将这些基线度量数据与设计时的期望值进行比较是非常重要的。通过执行此操作，可以检查客户机代码是否运转正常：例如，检查连接是否未保持打开或者已使用的消息是否仍然保留在未确认状态。这些编码错误会消耗消息服务器资源，并可能极大地影响性能。

基线使用模式有助于确定如何调整系统以优化性能。例如：

- 如果某个物理目标的使用频率明显高于其他目标，那么您可能需要对该物理目标设置比其他目标更高的内存限制，或者相应地调整限制行为。

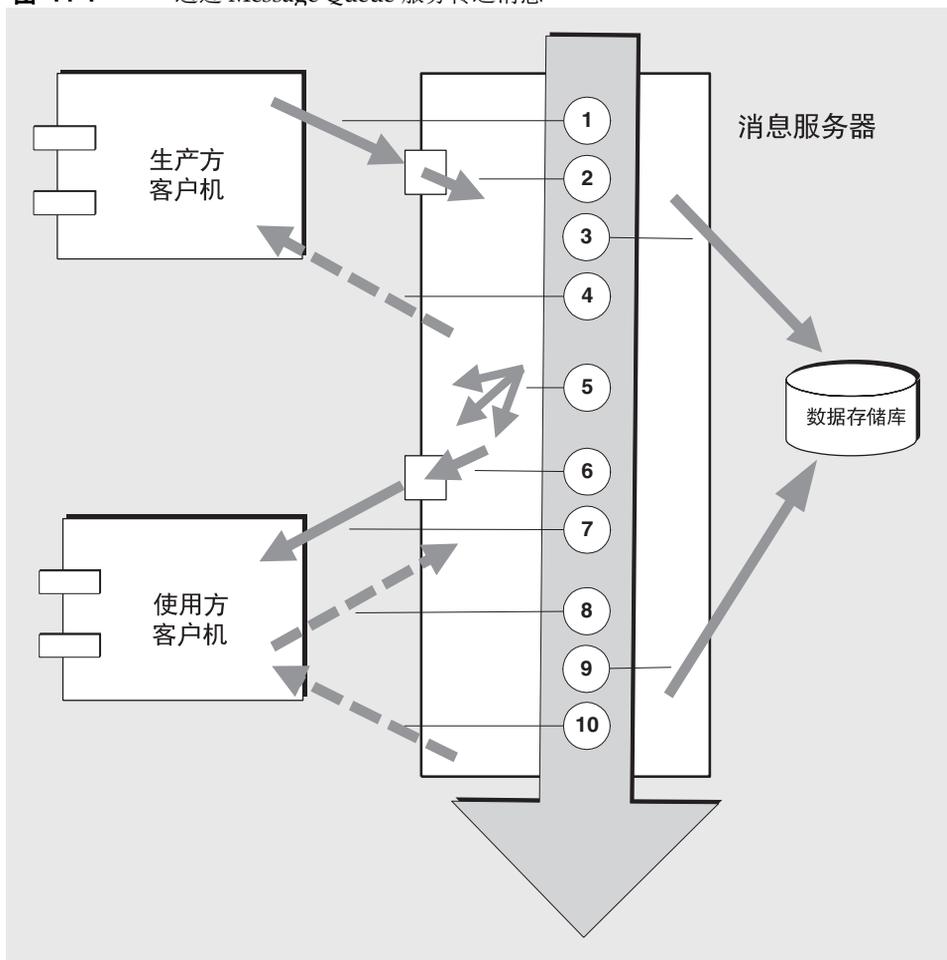
- 如果需要的连接数明显大于允许的最大线程池大小，那么可能需要增大线程池的大小，或者采用共享线程模型。
- 如果峰值消息流远远大于平均消息流，这可能影响您在内存不足时采用的限制行为。

通常，您对使用模式了解得越多，就能越好地将系统调整为这些模式。并计划将来的需要。

影响性能的因素

消息延迟和消息吞吐量是两个主要的性能指示器，它们通常取决于典型消息在完成消息传送过程中的各个步骤时所需的时间。下面步骤显示了如何以持久、可靠的方式传送消息。这些步骤将在插图之后讲述。

图 11-1 通过 Message Queue 服务传送消息



1. 消息从生产者客户机传送到消息服务器
2. 消息服务器读取消息
3. 消息被放置到持久性存储器当中（出于可靠性的考虑）
4. 消息服务器确认收到消息（出于可靠性的考虑）
5. 消息服务器确定消息的路由
6. 消息服务器写出消息
7. 消息从消息服务器传送到使用者客户机

8. 使用方客户机确认收到消息（出于可靠性的考虑）
9. 消息服务器处理客户机确认（出于可靠性的考虑）
10. 消息服务器确定已经处理客户机确认

由于这些步骤是连续的，所以任何步骤都可能成为消息从生产方客户机到使用方客户机的传送过程的瓶颈。这些步骤中的大多数都取决于消息传送系统的物理特征：网络带宽、计算机处理速度和消息服务器体系结构等等。但是，有一些步骤还取决于消息传送应用程序的特征和该应用程序要求的可靠性级别。

以下各小节讨论应用程序设计因素和消息传送系统因素这二者对性能的影响。尽管应用程序设计和消息传送系统因素在消息的传送过程中交互紧密，但它们彼此是独立的。

影响性能的应用程序设计因素

应用程序的设计决策对消息传送的总体性能有很大影响。

对性能影响最大的主要是那些影响消息传送的可靠性因素。包括下列因素：

- 传送模式（持久性 / 非持久性消息）
- 使用事务
- 确认模式
- 长期与非长期订阅

其他影响性能的应用程序设计因素有：

- 使用选择器（消息过滤）
- 消息大小
- 消息主体类型

接下来的几节讲述这些因素当中的每一个对消息传送性能所产生的影响。通常，性能和可靠性应该折衷。增加可靠性的因素可能会导致性能降低。

表 11-1 显示各个应用程序设计因素大体上如何影响消息传送性能。该表显示了两个方案（一个高可靠性、低性能方案和一个高性能、低可靠性方案）和分别相应于这两个方案的应用程序设计因素选项。在这两种极端情况之间，有很多可以同时影响可靠性和性能的选项和折衷。

表 11-1 高可靠性和高性能方案比较

| 应用程序设计因素 | 高可靠性 低性能方案 | 高性能 低可靠性方案 |
|------------|--|-----------------------|
| 传送模式 | 持久性消息 | 非持久性消息 |
| 使用事务 | 事务会话 | 非事务 |
| 确认模式 | AUTO_ACKNOWLEDGE CLIENT_ACKNOWLEDGE | 或 DUPS_OK_ACKNOWLEDGE |
| 长期 / 非长期订阅 | 长期订阅 | 非长期订阅 |
| 使用选择器 | 消息过滤 | 无消息过滤 |
| 消息大小 | 大量的小消息 | 少量的大消息 |
| 消息主体类型 | 复杂主体类型 | 简单主体类型 |

注 在接下来的图中，性能数据是在一个使用基于文件的持久性双 CPU、1002 Mhz Solaris 8 系统上生成的。性能测试首先预热 Message Queue 代理，使实时编译器优化系统，并启动持久性数据库。

代理预热后，会创建单个生产方和单个使用方，并在 30 秒钟内生成消息。系统将记录使用方接收所有生成的消息所需的时间，并计算吞吐率（每秒收到的消息数）。对于表 11-1 中所示的不同应用程序设计因素组合，此情形会重复出现。

传送模式（持久性 / 非持久性消息）

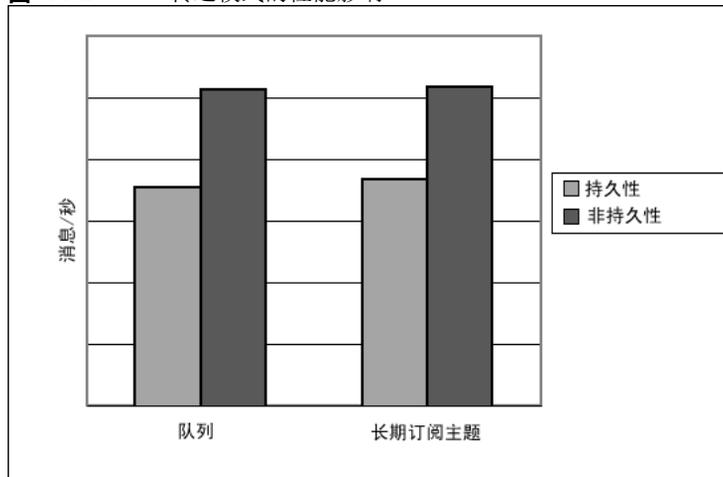
持久性消息能够在消息服务器发生故障的情况下保证消息的传送。代理会把消息存储在持久性存储中，直到所有预期的使用方都确认已使用消息为止。

通过代理处理持久性消息比处理非持久性消息要慢，理由如下：

- 代理必须可靠地存储持久性消息，以使其即使在代理出现故障时也不会丢失。
- 代理必须确认它所收到的每一条持久性消息。如果生成消息的方法返回时没有异常，则对代理的传送即已确保成功。
- 根据客户机确认模式的不同，代理可能需要确认使用方客户机对持久性消息的确认。

持久性和非持久性模式的性能之间可以有很大的差别。图 11-2 比较了在两种可靠的传送情况下持久性和非持久性消息的吞吐量；同时将 10k 大小的消息传送至队列和长期订阅的主题。两种情况下都使用 AUTO_ACKNOWLEDGE 确认模式。

图 11-2 传送模式的性能影响



使用事务

事务是一种保证，保证在事务会话中生成和使用的消息将作为一个单元进行处理或不进行处理（回滚）。

Message Queue 支持本地事务和分布式事务。

消息在事务会话中生成或确认时比在非事务会话中要慢，理由如下：

- 其他信息必须随每个生成的消息存储。
- 在某些情况下，事务中通常不该存储的消息也存储了（例如，传送至没有订阅的主题目标的持久性消息通常应该删除，但在事务开始时，却没有关于订阅方面的信息）。
- 提交事务时，该事务中有关消息的使用和确认的信息必须存储并处理。

确认模式

确保可靠传送 JMS 消息的一种机制是，客户机确认使用了由 Message Queue 消息服务器传送给它的消息。

如果在客户机尚未确认消息之前就关闭了会话，或者如果消息服务器在处理确认之前发生了故障，代理将重新传送该消息，并设置一个 JMSRedelivered 标志。

对于非事务会话，客户机可以选择三种确认模式之一，这三种模式有其各自的性能特征：

- `AUTO_ACKNOWLEDGE`。使用方处理消息后，系统会自动确认该消息。此模式可确保在提供商发生故障后至少重新传送一次消息。
- `CLIENT_ACKNOWLEDGE`。应用程序控制确认消息的时间点。该会话中自上次确认以来处理的所有消息都将被确认。如果消息服务器在处理一组确认时发生故障，则该组中的一个或多个消息将被重新传送。
- `DUPS_OK_ACKNOWLEDGE`。此模式指示系统以一种惰性方式确认消息。在提供商发生故障后，可以重新传送多条消息。

（`CLIENT_ACKNOWLEDGE` 模式的用法与事务的用法类似，不同之处在于：它不能保证当提供商在处理过程中发生故障时，所有确认都将一起处理）。

确认模式影响性能的原因如下：

- `AUTO_ACKNOWLEDGE` 和 `CLIENT_ACKNOWLEDGE` 模式需要在代理和客户机之间有额外的控制消息。附加的控制消息会添加额外的处理开销，并干扰 JMS 有效负荷消息，从而导致处理延迟。
- 在 `AUTO_ACKNOWLEDGE` 和 `CLIENT_ACKNOWLEDGE` 模式中，客户机必须等到代理确认它已处理客户机的确认后，才能使用其他消息。（这种代理确认可确保代理不会无意地重新传送这些消息。）
- `Message Queue` 持久性存储必须采用使用方收到的所有持久性消息的确认信息进行更新，因而降低了性能。

长期与非长期订阅

主题目标的订户可以归为两类，即长期订阅和非长期订阅的订户。

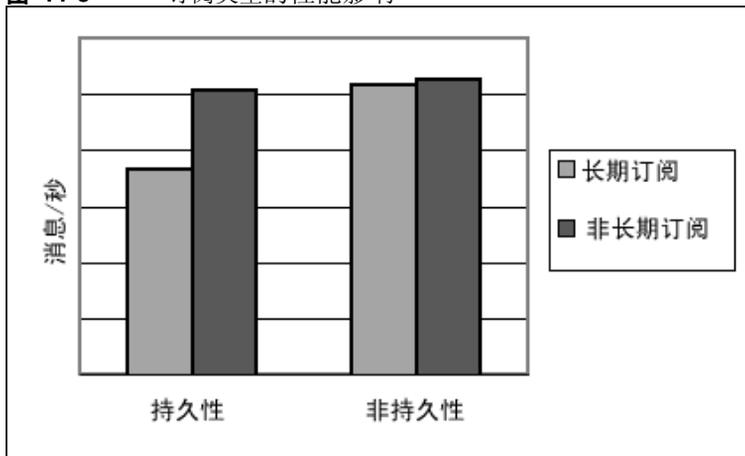
长期订阅的可靠性较高，但吞吐较慢，原因如下：

- `Message Queue` 消息服务器必须永久地存储指定给每个长期订阅的消息的列表，因此如果消息服务器发生故障，该列表只有在恢复之后才可用。
- 长期订阅的持久性消息是永久存储的，因此如果消息服务器发生故障，消息仍可以在恢复后传送（只要相应的使用方处于活动状态）。与此相反，非长期订阅的持久性消息不是永久存储的（如果消息服务器发生故障，相应的使用方的连接就会断开，消息不再被传送）。

图 11-3 比较了两种情况下的长期订阅和非长期订阅主题目标的吞吐量：10K 大小的持久性和非持久性消息。两种情况下都使用 `AUTO_ACKNOWLEDGE` 确认模式。

从图 11-3 中可以看到只有在持久性消息情况下，使用长期订阅的性能影响才会很明显，出现这种影响是因为持久性消息只有在长期订阅时才会永久存储，如上文所述。

图 11-3 订阅类型的性能影响



使用选择器（消息过滤）

应用程序的开发者常常需要将消息组的目标定为特定使用方。实现此愿望的方法有两种：将每组消息分别指向唯一的物理目标；或者是仅使用单个物理目标，并为每个使用方注册一个或多个选择器。

选择器是一种只请求特定消息的字符串，这些消息的属性值应该与传送到特定使用方的字符串匹配。例如，选择器 `NumberOfOrders > 1` 仅传送 `NumberOfOrders` 属性值为 2 或更大值的消息。

将使用方注册到选择器会降低性能（与使用多个物理目标相比），因为会需要进行额外的处理，以处理每个消息。如果使用选择器，则必须对它进行解析，以使它与将来的消息匹配。另外，路由每个消息时，都必须检索该消息的属性，并与选择器相比较。但是，使用选择器为消息传送应用程序提供了更大的灵活性。

消息大小

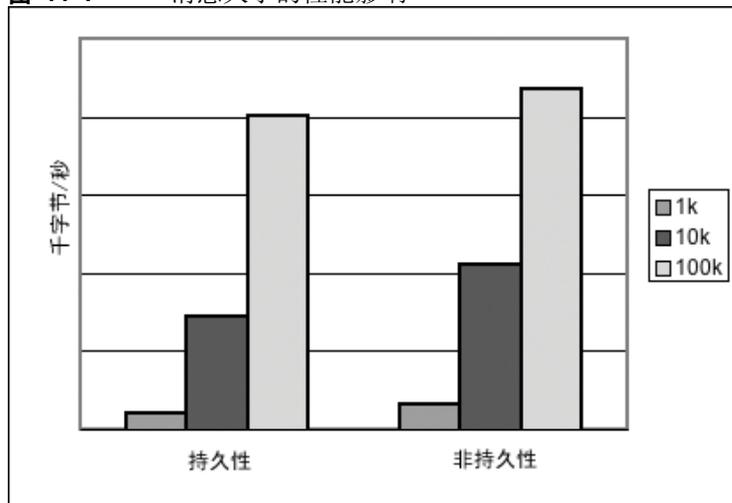
消息大小会影响性能是因为从生产方客户机到代理以及从代理到使用方客户机之间必须传递更多的数据，并且对于持久性消息，必须存储更大的消息。

但是，通过将较小的消息成批地变为一个消息，对各个消息的路由和处理就可以变得尽可能地简单，从而获得总体性能的提高。此时，就会丢失有关各个消息的状态的信息。

图 11-4 比较了在两种情况下，1k、10k 和 100k 大小的消息的吞吐量（KB/秒）：持久性和非持久性消息。每种情况都将消息发送至队列目标，并使用 `AUTO_ACKNOWLEDGE` 确认模式。

图 11-4 显示在这两种情况下，传送较大消息的开销都比传送较小消息时要少。您还可以看到在 1k 和 10k 大小的消息上出现的情况：非持久性消息较持久性消息接近 50% 的性能提高在 100k 大小的消息上并未得到体现，可能是因为在这种情况下，网络带宽已经成为消息吞吐量的瓶颈。

图 11-4 消息大小的性能影响



消息主体类型

JMS 支持五种消息主体类型，按照复杂性顺序大致显示如下：

- `BytesMessage`: 按照应用程序所确定的格式包含一组字节
- `TextMessage`: 一种简单的 `java.lang.String`
- `StreamMessage`: 包含 Java 基元值流
- `MapMessage`: 包含一组名 - 值对
- `ObjectMessage`: 包含 Java 序列化对象

尽管通常情况下消息类型是由应用程序的需要所决定的，但较复杂的类型（`MapMessage` 和 `ObjectMessage`）会增加性能成本——对数据进行序列化和反序列化的成本。性能成本取决于数据的简单或复杂程度。

影响性能的消息服务因素

消息传送应用程序的性能不但受应用程序设计的影响，而且还受执行消息路由和传送的消息服务的影响。

以下各节讨论影响性能的各个消息服务因素。了解这些因素的影响对于评估消息服务以及诊断并解决在部署的应用程序中可能发生的性能瓶颈来说至关重要。

Message Queue 服务中影响性能的最重要的因素有：

- 硬件
- 操作系统
- Java 虚拟机 (JVM)
- 连接
- 代理限制和行为
- 消息服务器体系结构
- 数据存储库性能
- 客户机运行时配置

以下各节讲述这些因素中的每一个对消息传送性能所产生的影响。

硬件

对于 Message Queue 消息服务器和客户机应用程序而言，CPU 处理速度和可用内存都是消息服务性能的主要决定因素。大多数软件限制都可以通过增加处理能力来消除，而添加内存则可以同时提高处理速度和容量。但是，仅通过升级硬件来克服瓶颈通常过于昂贵。

操作系统

由于不同操作系统的效率不同，所以即使硬件平台相同，性能也会各不相同。例如，操作系统使用的线程模型会对消息服务器可以支持的并行连接数产生重要影响。在所有硬件都相同的情况下，Solaris 通常比 Linux 快，而后者通常又比 Windows 快。

Java 虚拟机 (JVM)

消息服务器是受主机 JVM 支持并运行在其中的一种 Java 进程。因此，JVM 处理是决定消息服务器路由和传送消息的速度和效率的重要因素。

特别是 JVM 的内存资源管理至关重要。必须为 JVM 分配足够的内存以满足不断增大的内存负荷。另外，JVM 将定期地释放未使用的内存，而这种内存释放会延迟消息的处理。JVM 内存堆越大，内存释放过程中可能遇到的延迟就越长。

连接

客户机和代理之间的连接数目和速度可能影响消息服务器可以处理的消息数以及消息的传送速度。

消息服务器连接限制

对消息服务器的所有访问都是通过连接的方式进行的。对并行连接数的任何限制都会影响可以同时使用消息服务器的生产方或使用方客户机的数目。

消息服务器的连接数通常由可用的线程数限制。Message Queue 使用了线程池管理器，您可以将其配置为支持专用线程模型或共享线程模型（请参见第 75 页“线程池管理器”）。

专用线程模型的速度非常快，因为每个连接都有专用的线程，但是连接数目受可用线程数的限制（每个连接都有一个输入线程和一个输出线程）。共享线程模型对连接数不加任何限制，但是在大量连接之间共享线程时会有明显的开销和吞吐量延迟，特别是当这些连接都很繁忙时。

传输协议

Message Queue 软件允许客户机使用各种低级别的传输协议与消息服务器进行通信。Message Queue 支持第 73 页“连接服务”中所述的连接服务（及相应协议）。

协议的选择基于应用程序的要求（加密、通过防火墙访问等），但是所作的选择会影响总体性能。

图 11-5 传输协议速度

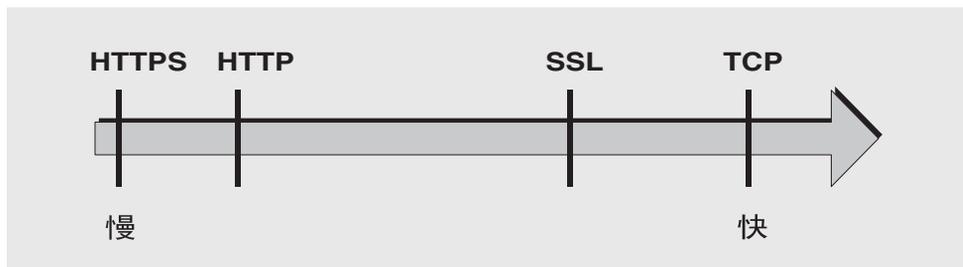


图 11-5 反映不同协议技术的性能特征：

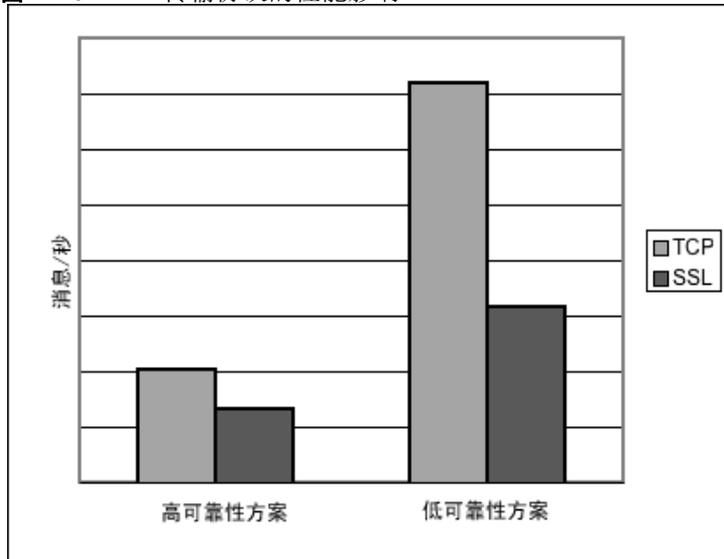
- TCP 提供与代理通信的最快方法。

- SSL 在收发消息时比 TCP 要慢 50% 到 70%（对于持久性消息是 50%，对于非持久性消息则接近 70%）。另外，SSL 在建立初始连接时比较慢（可能需要数秒钟），因为客户机和代理（在使用 HTTPS 时为 Web 服务器）需要建立专用密钥，以供加密数据以进行传输时使用。性能的下降源自于加密和解密每个低级别 TCP 包时所需的额外处理。

图 11-6 比较了两种情况下的 TCP 和 SSL 吞吐量：一个高可靠性方案（将 1k 大小的持久性消息发送至长期订阅主题目标，并使用 AUTO_ACKNOWLEDGE 确认模式）和一个高性能方案（将 1k 大小的非持久性消息发送至非长期订阅主题目标，并使用 DUPES_OK_ACKNOWLEDGE 确认模式）。

图 11-6 显示了在高可靠性情况下产生较小影响的协议。这可能是因为在高可靠性情况下所需的持久性开销在限制吞吐量方面是比协议速度更重要的因素。

图 11-6 传输协议的性能影响



- HTTP 比 TCP 或 SSL 都要慢。它使用 servlet，该 servlet 在 Web 服务器上作为客户机和代理之间的代理服务器运行。性能开销涉及封装 HTTP 请求中的包，并涉及消息要经过两个跃点（客户机到 servlet，servlet 到代理）后才到达代理这一要求。
- HTTPS 比 HTTP 要慢是因为需要额外的开销以加密客户机和 servlet 之间以及 servlet 和代理之间的包。

消息服务器体系结构

Message Queue 消息服务器可以作为单个代理实现，也可以作为多个互相连接的代理实例（即代理群集）实现。

随着连接到代理的客户机数量以及所传送的消息数量的不断增加，代理最终将超出资源限制（例如文件描述符、线程和内存限制）。适应不断增加的负荷的方法之一是将更多的代理实例添加到 Message Queue 消息服务器，从而将客户机连接以及消息路由和传送分布到多个代理。

通常，如果客户机（特别是消息生产方客户机）均匀地分布在群集中，则这种调整最为有效。由于在群集的代理之间传送消息会涉及到开销，因此连接数有限或消息传送速率有限的群集，其性能可能会比单个代理要低。

您也可以使用代理群集来优化网络带宽。例如，您可能希望在群集内的一组远程代理之间使用低速的长途网络链路，而在客户机与其各自的代理实例之间使用高速链接进行连接。

有关群集的详细信息，请参见第 9 章“使用代理群集”。

代理限制和行为

消息服务器可能需要处理的消息吞吐量是消息服务器所支持的消息传送应用程序的使用模式的一项功能。但是，消息服务器在资源方面有限：内存、CPU 周期等。因此，在消息服务器变得无响应或不稳定的位置，它可能会崩溃。

Message Queue 消息服务器具有管理内存资源并防止代理用尽内存的内置机制。这些机制包括可配置的消息数限制，或代理可以拥有的消息字节数或其各自的物理目标，以及当达到物理目标限制时可以实施的一组行为。

通过仔细地监视和调整，这些可配置机制可以用于平衡消息的内流和外流，以使系统过载不会发生。尽管这些机制会造成开销并限制消息的吞吐量，但它们可以维护操作的完成性。

数据存储库性能

Message Queue 既支持内置的持久性，也支持插入的持久性。内置的持久性是基于文件的数据存储库。插入的持久性使用 Java 数据库连接 (JDBC™) 接口，并需要 JDBC 兼容的数据存储库。

内置的持久性明显比插入的持久性要快，但是 JDBC 兼容的数据库系统可以提供应用程序需要的冗余、安全性和管理功能。

如果是内置持久性，您可以通过指定让持久性操作将内存中状态与数据存储库同步，从而将可靠性增至最大。这有助于消除因系统崩溃而导致的数据丢失，但代价是性能的下降。

客户机运行时配置

Message Queue 客户机运行时可提供客户机应用程序及其与 Message Queue 消息服务的接口。它支持客户机向物理目标发送消息以及接收来自这些目标的消息所需的所有操作。客户机运行时是可配置的（通过设置连接工厂属性值），使您可以设置通常能够提高性能和消息吞吐量的属性和行为。

例如，Message Queue 客户机运行时支持下列可配置行为：

- 连接流测量 (`imqConnectionFlowCount`)，此行为有助于防止由于 JMS 消息和 Message Queue 控制消息通过同一连接时所形成的流引起的拥塞。
- 连接流限制 (`imqConnectionFlowLimit`)，此行为有助于通过限制等待使用的、可以通过连接传送到客户机运行时的消息数，来避免客户机的资源限制。
- 使用方流限制 (`imqConsumerFlowLimit`)，此行为有助于改进在多用户队列传送情况下用户之间的负荷平衡（因此向任何一个用户发送的消息数都是均衡的），并有助于阻止连接上的任何一个用户影响连接上的其他用户。此属性可以限制每个使用方等待使用的、可以通过连接传送到客户机运行时的消息数。还可以将该属性配置为队列目标属性 (`consumerFlowLimit`)。

有关这些行为和用来配置这些行为的属性的详细信息，请参见第 221 页“客户机运行时消息流调整”。

调整配置以提高性能

系统调整

以下各节讲述您可以对操作系统、JVM 和通信协议所作的调整。

Solaris 调整：CPU 使用、分页 / 交换 / 磁盘 I/O

有关对操作系统的调整，请参见系统文档。

Java 虚拟机调整

默认情况下，代理使用大小为 192MB 的 JVM 堆。通常，这对于较大的消息负荷来说太小，应该增大。

当代理快要耗尽 Java 对象使用的 JVM 堆空间时，它将使用各种技术（如流控制和消息交换）来释放内存。在极端情况下，代理甚至关闭客户机连接以释放内存并减少消息内流。所以最好将最大 JVM 堆空间设置得足够大，以避免这种情况。

但是，与系统的物理内存相比，如果最大 Java 堆空间设置过大，代理将继续增大 Java 堆空间，直至整个系统耗尽内存。这会导致性能的降低、不可预计的代理崩溃和 / 或影响系统中运行的其他应用程序和服务的行为。通常，需要有足够的物理内存以便操作系统和其他应用程序在计算机上运行。

总的说来，好的方法是：估算正常和峰值系统内存容量，并配置 Java 堆大小，使其足以提供良好性能，但同时不应过大，以免引起系统内存问题。

要更改代理的最小和最大堆大小，请在启动代理时使用 `-vmargs` 命令行选项。例如：

```
/usr/bin/imqbrokerd -vmargs "-Xms256m -Xmx1024m"
```

此命令会将启动 Java 堆大小设置为 256MB，将最大 Java 堆大小设置为 1GB。

- 在 Solaris 或 Linux 上，如果通过 `/etc/rc*`（即 `/etc/init.d/imq`）启动代理，请在 `/etc/imq/imqbrokerd.conf` (Solaris) 或 `/etc/opt/sun/mq/imqbrokerd.conf` (Linux) 文件中指定代理的命令行参数。有关详细信息，请参见该文件中的注释。
- 在 Windows 上，如果将代理作为 Window 服务启动，请使用 `imqsvcadm install` 命令的 `-vmargs` 选项指定 JVM 参数。请参见第 13 章“命令参考”中的“[imqsvcadm](#)”

在任何情况下，请通过检查代理的日志文件或通过使用 `imqcmd metrics bkr -m cxn` 命令验证设置。

调整传输协议

选择了符合应用程序需要的协议后，基于该协议进行其他调整可以提高性能。

协议的性能可以使用下面三个代理属性进行修改：

- `imq.protocol.protocol_type.nodelay`
- `imq.protocol.protocol_type.inbufsz`
- `imq.protocol.protocol_type.outbufsz`

对于 TCP 和 SSL 协议，这些属性会影响客户机和代理之间的消息传送速度。对于 HTTP 和 HTTPS 协议，这些属性会影响 Message Queue 隧道 servlet（在 Web 服务器上运行）和代理之间的消息传送速度。对于 HTTP/HTTPS 协议，还有其他可以影响性能的属性（请参见第 219 页“[HTTP/HTTPS 调整](#)”）。

协议调整属性将在以下各节中讲述。

nodelay

`nodelay` 属性影响给定协议的 Nagle 算法（TCP/IP 上的 TCP_NODELAY 套接字级选项的值）。Nagle 算法用于提高使用慢速连接（例如广域网 (WAN)）的系统上的 TCP 性能。

如果使用了此算法，TCP 将通过把多个数据捆绑为较大的包来尝试防止将多个小块数据发送到远程系统。如果写入套接字中的数据没有填满需要的缓冲区大小，协议将延迟发送包，直到缓冲区被填满，或者已经过了特定的延迟时间为止。填满了缓冲区或者发生了超时时，包将发送。

对于大多数消息传送应用程序，如果包发送过程中没有延迟（Nagle 算法未启用），则性能是最佳的。这是因为客户机和代理之间的大多数交互都是请求 / 响应交互：客户机向代理发送包，并等待响应。例如，典型的交互包括：

- 创建连接
- 创建生产方或使用方
- 发送持久性消息（代理确认收到消息）
- 在 `AUTO_ACKNOWLEDGE` 或 `CLIENT_ACKNOWLEDGE` 会话中发送客户机确认（代理确认对客户机确认的处理）

对于这些交互，大多数包都比缓冲区大小要小。这意味着如果使用 Nagle 算法，代理会在向使用方发送响应之前延迟几毫秒。

但是，在连接较慢以及不需要代理响应的情况下，Nagle 算法可以提高性能。例如，当客户机发送非持久性消息或者当客户机确认未被代理确认（`DUPS_OK_ACKNOWLEDGE` 会话）时，就属于这样的情况。

inbufsz/outbufsz

`inbufsz` 属性用于在读取来自套接字的数据的输入流上设置缓冲区大小。同样，`outbufsz` 用于设置代理用来将数据写入套接字的输出流的缓冲区大小。

通常，这两个参数都应该设置为比收发的平均包要稍大的值。一个很好的经验是将这些属性值设为平均包的大小再加上 1k（舍入为最接近的 k 值）。

例如，如果代理正在接收主体大小为 1k 的包，则该包的总体大小（消息主体 + 标题 + 属性）约为 1200 字节。大小为 2k（2048 字节）的 `inbufsz` 可以提供合理的性能。

增大 `inbufsz` 或 `outbufsz`（使其大于该值）可以稍微提高性能，但这样会增加每个连接所需的内存。

图 11-7 显示对大小为 1K 的包更改 `inbufsz` 的结果。

图 11-7 更改大小为 1K（1024 字节）的包的 `inbufsz` 的结果

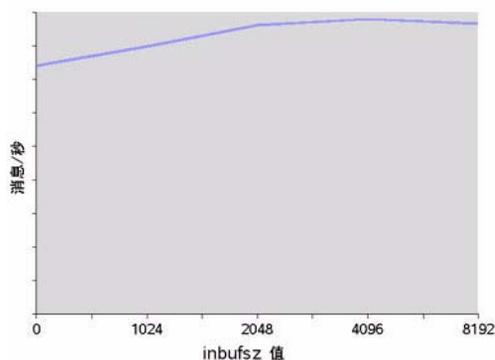


图 11-8 显示对大小为 1K 的包更改 `outbufsz` 的结果。

图 11-8 更改大小为 1K（1024 字节）的包的 `outbufsz` 的结果



HTTP/HTTPS 调整

除了前面两节讨论的一般属性之外，HTTP/HTTPS 的性能还受到客户机向作为 Message Queue 隧道 servlet 宿主的 Web 服务器发出 HTTP 请求的速度的限制。

Web 服务器可能需要优化，以处理单个套接字上的多个请求。在 JDK 1.4 版及更高版本中，对 Web 服务器的 HTTP 连接会一直保持（对 Web 服务器的套接字保持打开），以使 Web 服务器在处理多个 HTTP 请求时使用的资源最小化。如果使用 JDK 1.4 版的客户机应用程序的性能比运行早期 JDK 版本的同一应用程序要低，则可能需要调整 Web 服务器的保持配置参数以提高性能。

除了这样的 Web 服务器调整之外，您还可以调整客户机轮询 Web 服务器的频率。HTTP 是一种基于请求的协议。这意味着使用基于 HTTP 的协议的客户机需要定期地检查 Web 服务器，以查看是否有消息在等待。imq.httpjms.http.pullPeriod 代理属性（以及相应的 imq.httpsjms.https.pullPeriod 属性）可指定 Message Queue 客户机运行时轮询 Web 服务器的频率。

如果 pullPeriod 值为 -1（默认值），客户机运行时将在前一个请求返回后立即轮询服务器，从而将各个客户机的性能最大化。结果，每个客户机连接都会在 Web 服务器中独占一个请求线程，这样可能会耗费 Web 服务器资源。

如果 pullPeriod 值为正数，客户机运行时将定期向 Web 服务器发送请求，以查看是否有挂起的数据。在这种情况下，客户机不会独占 Web 服务器中的请求线程。因此，如果有大量的客户机在使用 Web 服务器，您可以通过将 pullPeriod 设为正值来节约 Web 服务器资源。

调整基于文件的持久性存储

有关调整基于文件的持久性存储的信息，请参见第 80 页“持久性管理器”。

代理调整

以下各节讲述为了提高性能而可以对代理属性作出的调整。

内存管理：增大代理在负荷下的稳定性

内存管理可以在逐个目标的级别上配置，也可以在系统范围级别（集中地对于所有目标）上配置。

使用物理目标限制

有关物理目标限制的信息，请参见第 6 章“管理物理目标”。

使用系统范围限制

如果消息生产方的数目超过消息使用方的数目，则消息可能在代理中堆积。代理包含限制生产方和在内存过小的情况下将消息交换出活动内存的机制，但最好还是对代理可以保持的消息数和消息字节总数进行严格限制。

可以通过设置 imq.system.max_count 和 imq.system.max_size 代理属性来控制这些限制。

例如

```
imq.system.max_count=5000
```

上面定义的值表示代理最多仅能保留 5000 条未传送 / 未确认的消息。如果发送了其他消息，它们将被代理拒绝。如果消息是持久性的，当生产方尝试发送该消息时，会收到一个异常。如果消息是非持久性的，代理将自行废弃该消息。

若要让非持久性消息像持久性消息那样返回异常，请对客户机使用的连接工厂对象设置下列属性：

```
imqAckOnProduce = true
```

上面的设置会降低向代理发送非持久性消息的性能（客户机会等待回复后才发送下一条消息），但通常这是可以接受的，因为内流到代理的消息通常不是系统瓶颈。

如果在发送消息的过程中返回了异常，客户机应该暂停一会，然后再次重试发送。

多使用方队列性能

多队列使用方处理队列目标中消息的效率取决于下列可配置的队列目标属性：

- 活动使用方数 (maxNumActiveConsumers)
- 可以在一批中传送给使用方的消息的最大数量 (consumerFlowLimit)

要达到优化的消息吞吐量，必须有足够数量的活动使用方以适应队列的消息生产方的速率，并且队列中的消息必须以最大化使用速率的方式路由和传送给活动使用方。Sun Java System Message Queue 技术概述中介绍了在多个使用方之间平衡消息传送的一般机制。

如果消息在队列中堆积，这可能是因为没有足够的活动使用方来处理消息负荷。也可能是每批传送给使用方的消息太多，导致消息在使用方堆积。例如，如果每批的大小 (consumerFlowLimit) 太大，某个使用方就可能会收到一个队列中的所有消息，而其他活动使用方则一个也没收到。如果使用方速度特别快，这也不会成为问题。

但是如果使用方相对较慢，而您又希望将消息均匀地分布给它们，则需要将每批的大小减小。每批的大小越小，将消息传送到使用方所需的开销就越多。但是对于较慢的使用方，使用较小的批大小通常却能获得网络性能的提升。

客户机运行时消息流调整

本节讨论影响性能的流控制行为（请参见第 216 页“客户机运行时配置”）。这些行为可配置为连接工厂受管理对象的属性。有关设置连接工厂属性的信息，请参见第 8 章“管理受管理对象”

消息流测量

客户机收发的消息（JMS 消息）以及 Message Queue 控制消息通过同一客户机 n 代理连接进行传递。如果控制消息（例如代理确认）的传送被 JMS 消息阻挡，则会发生延迟。为防止此类拥塞，Message Queue 会测量通过连接的 JMS 消息流。

JMS 消息是成批的（由 `imqConnectionFactoryCount` 属性指定），因此只会传送一组 JMS 消息。传送完一批后，将暂停 JMS 消息的传送，而只传送暂挂的控制消息。当另一批 JMS 消息传送后，接着又传送暂挂的控制消息，这样循环往复。

如果客户机正在执行需要代理作出大量响应的操作，`imqConnectionFactoryCount` 的值应该保持较低；例如，当客户机正在使用 `CLIENT_ACKNOWLEDGE` 或 `AUTO_ACKNOWLEDGE` 模式、持久性消息、事务、队列浏览器时，或者当客户机正在添加或删除使用方时。从另一方面说，如果客户机仅在使用 `DUPS_OK_ACKNOWLEDGE` 模式的连接上有简单的使用方，则可以增大 `imqConnectionFactoryCount` 而不会降低性能。

消息流限制

在遇到本地资源（例如内存）限制之前，存在 Message Queue 客户机运行时可以处理的 JMS 消息数限制。如果达到了此限制，性能将受影响。因此，Message Queue 可以让您限制每个使用方（或每个连接）能够通过连接传送和能够在客户机运行时中缓冲以等待使用的消息数。

基于使用方的限制

当传送到客户机运行时的 JMS 消息数超过任意使用方的 `imqConsumerFlowLimit` 值时，将停止传送该使用方的消息。仅当该使用方的未使用消息数下降到低于 `imqConsumerFlowThreshold` 设置的值时才会恢复。

下例说明了如何使用这些限制：以主题使用方的默认设置为例

```
imqConsumerFlowLimit=1000
```

```
imqConsumerFlowThreshold=50
```

创建使用方后，代理将向此使用方传送第一批 1000 条消息（前提是有这么多），中间不会暂停。发送 1000 条消息后，代理将停止传送，除非客户机运行时要求更多消息。客户机运行将保留这些消息，直到应用程序处理它们为止。在要求代理发送下一批消息之前，客户机运行时允许应用程序使用至少 50% (`imqConsumerFlowThreshold`) 的消息缓冲区容量（即 500 条消息）。

在同等情况下，如果阈值为 10%，则客户机运行时等待应用程序使用至少 900 条消息，然后才会要求发送下一批消息。

下一批消息的大小按如下计算：

$$\text{imqConsumerFlowLimit} - (\text{缓冲区中当前暂挂的消息数})$$

因此，如果 `imqConsumerFlowThreshold` 为 50%，则下一批的大小会在 500 和 1000 之间波动，这取决于应用程序处理消息的速度。

如果 `imqConsumerFlowThreshold` 设置得过高（接近 100%），代理就会发送较小的分批消息，这样会降低消息的吞吐量。如果该值过低（接近 0%），则客户机可以在代理传送下一组消息之前处理完剩余的缓冲消息，从而导致消息吞吐量的降低。通常，除非您有特别的性能或可靠性考虑，否则不需要更改 `imqConsumerFlowThreshold` 属性的默认值。

基于使用方的流控制（特别是 `imqConsumerFlowLimit`）是管理客户机运行时中内存的最好方法。通常，根据客户机应用程序的不同，您应该知道在任意连接上需要支持的使用方数、消息的大小以及可用于客户机运行时的内存总量。

基于连接的限制

但是在某些客户机应用程序中，使用方数量不是确定的，这取决于最终用户所作的选择。在这些情况下，您仍可以使用连接级流限制来管理内存。

连接级流控制可以限制针对一个连接上的所有使用方而缓冲的总消息数。如果此数目超过了 `imqConnectionFlowLimit`，则通过该连接进行的消息传送将停止，直到该总数降到连接限制以下为止。（只有在将 `imqConnectionFlowLimitEnabled` 属性设置为 `true` 时，`imqConnectionFlowLimit` 才会启用）。

在会话中暂挂的消息数是使用该会话的消息使用方数量以及每个使用方的消息负荷的函数。如果客户机在生成或使用消息时表现出延迟，您通常可以通过下列操作提高性能：重新设计应用程序，以便在更大数量的会话之间分布消息生产方和使用方，或者在更大数量的连接之间分布会话。

调整配置以提高性能

问题疑难解答

本章介绍如何了解及解决以下问题：

- 第 226 页 “客户机无法建立连接”
- 第 230 页 “连接的吞吐量太慢”
- 第 232 页 “客户机无法创建消息生产方”
- 第 233 页 “消息的生成过程发生延迟或减慢”
- 第 235 页 “消息堆积”
- 第 239 页 “消息服务器吞吐量呈间歇性”
- 第 240 页 “消息无法到达使用方”
- 第 244 页 “停用消息队列包含消息”

出现问题时，检查所安装 Message Queue 软件的版本号会很有帮助。使用版本号来确保目前正在使用的文档版本与软件版本相匹配。向 Sun 报告问题时，也需要用到版本号。要检查版本号，请执行以下命令：

```
imqcmd -v
```

客户机无法建立连接

此问题的症状如下：

- 客户机无法建立新连接。
- 客户机无法对失败的连接进行自动重新连接。

本节详细说明了以下可能的原因：

- 客户机应用程序不是闭合连接，导致连接数超出资源限制
- 代理未运行或者网络连接有问题
- 连接服务处于非活动状态或者已暂停
- 连接所要求的线程数不够
- Solaris 或 Linux 操作系统上连接数所需的文件描述符不足
- TCP 待办事项限制了可以同时建立的新连接请求数目
- 操作系统限制了并行连接数
- 对用户的验证或授权失败

客户机应用程序不是闭合连接，导致连接数超出资源限制

确认这是否就是问题的起因

列出代理的所有连接：

```
imqcmd list cxn
```

输出结果将列出所有的连接以及发起每个连接的主机，从而展示特定客户机的特有的打开连接数。

解决此问题

重写有问题的客户机，以关闭未使用的连接。

代理未运行或者网络连接有问题

确认这是否就是问题的起因

- 远程登录到代理的主端口（例如，默认端口为 7676），并验证代理是否以端口映射器输出结果作为响应。
- 验证代理进程是否正在主机上运行。

解决此问题

- 启动代理。
- 修复网络连接问题。

连接服务处于非活动状态或者已暂停

确认这是否就是问题的起因

检查所有连接服务的状态：

```
imqcmd list svc
```

如果某个连接服务的状态显示为 `unknown`（未知）或 `paused`（暂停），客户机将无法使用该服务建立连接。

解决此问题

- 如果连接服务的状态显示为 `unknown`，它将从活动服务列表 (`imq.service.active`) 中消失。如果是基于 SSL 的服务，那么还可能是因为服务未被正确配置，导致代理在代理日志中生成下面的条目：`ERROR [B3009]: 无法启动服务 ssljms: [B4001]: 无法打开 ssljms 服务的协议 tls ...`，后面带有关于引起此异常的解释。
要正确配置 SSL 服务，请参见第 144 页“使用基于 SSL 的服务”。
- 如果连接服务的状态显示为 `paused`，可以恢复该服务（请参见第 110 页“暂停和恢复连接服务”）。

连接所要求的线程数不够

确认这是否就是问题的起因

在代理日志中检查下面的条目：

```
WARNING [B3004]: 没有可用于处理服务上新连接的线程 ... 正在关闭新连接。
```

此外，请检查连接服务上的连接数以及当前使用的线程数（使用以下格式之一）：

```
imqcmd query svc -n serviceName
```

```
imqcmd metrics svc -n serviceName -m cxn
```

每个连接都需要两个线程：一个用于外来消息，另一个用于外出消息（请参见第 75 页“线程池管理器”）。

解决此问题

- 如果使用专用线程池模型 (`imq.service_name.threadpool_model=dedicated`)，则最大连接数是该线程池中的最大线程数的一半。因此，要增加连接数，请增加线程池的大小 (`imq.service_name.max_threads`) 或切换到共享线程池模型。
- 如果使用共享线程池模型 (`imq.service_name.threadpool_model=shared`)，则最大连接数是下面两个属性的乘积的一半。连接监视限制 (`imq.service_name.connectionMonitor_limit`) 和最大线程数 (`imq.service_name.max_threads`)。因此，要增加连接数，可以增加线程池的大小或增大连接监视限制。
- 最终，可支持的连接数（或连接上的吞吐量）将达到输出 / 输出限制。这种情况下，可以使用多代理群集在群集内的代理实例之间分布连接。

Solaris 或 Linux 操作系统上连接数所需的文件描述符不足

有关此问题的详细信息，请参见第 64 页“设置文件描述符限制（Solaris 或 Linux）”。

确认这是否就是问题的起因

在代理日志中检查类似下面的条目：打开的文件太多。

解决此问题

增大文件描述符限制，如 `ulimit` 手册页中所述。

TCP 待办事项限制了可以同时建立的新连接请求数目

TCP 待办事项设置可以存储在系统待办事项 (`imq.portmapper.backlog`) 中的同时连接请求数限制后，端口映射器才会拒绝额外的请求。（在 Windows 操作系统上，有一种硬编码的待办事项限制：Windows 台式机限制为 5，而 Windows 服务器限制为 200。）

出于待办事项限制而拒绝请求通常是一种由于同时连接请求数过多而导致的瞬态现象。

确认这是否就是问题的起因

检查代理日志。首先，检查代理是否在接受某些连接的同时拒绝其他连接。其次，检查说明拒绝连接原因的消息。如果找到此类消息，则说明问题可能不是由 TCP 待办事项引起的，因为代理不记录由于 TCP 待办事项而引起的连接拒绝。

如果记录了一些成功连接，但未记录任何连接拒绝，则问题可能是由 TCP 待办事项引起的。

解决此问题

下列方法可用于解决 TCP 待办事项限制：

- 对客户机进行编程，使其在较短的时间间隔后重试所尝试的连接（此方法之所以生效通常是由于此问题的瞬态性）。
- 提高 `imq.portmapper.backlog` 的值。
- 检查客户机是否过于频繁地反复关闭和打开连接。

操作系统限制了并行连接数

Windows 操作系统许可证对支持的并行远程连接数进行了限制。

确认这是否就是问题的起因

检查是否有可用于连接的足够线程（使用 `imqcmd query svc`），并检查您的 Windows 许可协议的条款。如果您可以从本地客户机建立连接，但不能从远程客户机建立连接，则操作系统的限制可能就是问题的起因。

解决此问题

- 升级 Windows 许可证，以允许更多的连接。
- 通过设置多代理群集将连接分布在大量代理实例之间。

对用户的验证或授权失败

验证可能因为密码错误而失败，原因是在用户系统信息库中没有该用户的条目，或者该用户没有对连接服务的访问权限。

确认这是否就是问题的起因

检查代理日志中的条目，判断是否有禁用错误消息。此消息指明存在验证错误，但不会指明该错误的原因。

- 如果使用的是基于文件的用户系统信息库，请输入下面的命令：

```
imqusermgr list -i instanceName -u userName
```
- 如果输出结果显示的是用户，说明可能提交了错误的密码。如果输出显示下列错误，则说明用户系统信息库中不包含任何条目：

```
ERROR [B3048]: 密码文件中不存在用户，
```
- 如果使用的是 LDAP 服务器用户系统信息库，请使用相应的工具检查是否存在该用户的条目。
- 检查访问控制文件以查看是否对连接服务有访问限制。

解决此问题

- 如果在用户系统信息库中没有该用户的条目，则将该用户添加到用户系统信息库中（请参见第 134 页“填充和管理用户系统信息库”）。
- 如果使用了错误的密码，请提供正确的密码。
- 如果访问控制属性设置不当，则编辑访问控制属性文件以授予连接服务权限（请参见第 141 页“用于连接服务的访问控制”）。

连接的吞吐量太慢

此问题的症状如下：

- 消息的吞吐量与预期不符。
- 支持的代理连接数受到限制不是如第 226 页“客户机无法建立连接”中所述的原因，而是出于消息的输入 / 输出速率。

本节详细说明了以下可能的原因：

- [网络连接或 WAN 太慢](#)
- [与 TCP 相比，连接服务协议本身就慢](#)
- [连接服务协议未优化调整](#)
- 消息太大，以致于占用了太多的带宽
- [使连接吞吐量变慢的可能原因也就是消息传送过程中某个步骤的瓶颈](#)

网络连接或 WAN 太慢

确认这是否就是问题的起因

Ping 网络，查看返回 ping 需要多长时间，然后咨询网络管理员。另外还可以使用本地客户机发送并接收消息，并将传送时间与远程客户机（使用网络链路）的传送时间相比。

解决此问题

如果连接太慢，则升级网络链路。

与 TCP 相比，连接服务协议本身就慢

例如，基于 SSL 或基于 HTTP 的协议要比 TCP 慢（请参见第 213 页图 11-5）。

确认这是否就是问题的起因

如果您使用的是基于 SSL 或基于 HTTP 的协议，请尝试使用 TCP，然后比较传送时间。

解决此问题

应用程序的要求通常会限定要使用的协议，因此您可以做的事情就很少了，无非是按第 217 页“调整传输协议”中所述尝试调整协议而已。

连接服务协议未优化调整

确认这是否就是问题的起因

尝试调整协议，并查看是否发生了变化。

解决此问题

尝试按第 217 页“调整传输协议”中所述调整协议。

消息太大，以致于占用了太多的带宽

确认这是否就是问题的起因

尝试使用较小的消息运行基准检验。

解决此问题

- 请应用程序开发者对应用程序进行修改以使用消息压缩功能，请参见 Message Queue Developer's Guide for Java Clients。
- 将消息作为要发送的数据的通知来使用，而使用其他协议移动数据。

使连接吞吐量变慢的可能原因也就是消息传送过程中某个步骤的瓶颈

确认这是否就是问题的起因

如果上述的每一条看来都不是造成连接吞吐量变慢的原因，请参见第 205 页图 11-1 以了解其他可能的瓶颈，并检查与下列问题相关的症状：

- 第 233 页“消息的生成过程发生延迟或减慢”
- 第 235 页“消息堆积”
- 第 239 页“消息服务器吞吐量呈间歇性”

解决此问题

遵循前面有关问题疑难解答的各节中提供的问题解决方案指导。

客户机无法创建消息生产方

此问题的症状如下：

- 无法为物理目标创建消息生产方，客户机收到异常。

本节详细说明了以下可能的原因：

- [物理目标被配置为仅允许有限数目的生产方](#)
- [由于访问控制属性文件中的设置，用户未获得创建消息生产方的授权](#)

物理目标被配置为仅允许有限数目的生产方

限制某个物理目标所支持的生产方 (`maxNumProducers`) 数目是避免消息在该物理目标上堆积的方法之一。

确认这是否就是问题的起因

检查物理目标（请参见第 121 页“显示有关物理目标的信息”）：

```
imqcmd query dst
```

输出结果将显示当前的生产方数目以及 `maxNumProducers` 的值。如果这两个值相同，说明生产方的数目已达到所配置的限制。如果新的生产方被代理拒绝，代理将返回“`ResourceAllocationException [C4088]: 到达 JMS 目标限制`”消息，且在代理日志中生成如下条目：`[B4183]: 无法将生产方添加到目标。`

解决此问题

增大 `maxNumProducers` 属性的值（请参见第 122 页“更新物理目标属性”）。

由于访问控制属性文件中的设置，用户未获得创建消息生产方的授权

确认这是否就是问题的起因

如果新的生产方被代理拒绝，代理将返回以下消息：

```
JMSSecurityException [C4076]: 客户机没有在目标上创建生产方的权限。
```

代理还在代理日志中记录下列条目：

```
[B2041]: 目标上的生产方被拒绝 和 [B4051]: 禁用 guest。
```

解决此问题

更改访问控制属性，允许用户生成消息（请参见第 143 页“对物理目标的访问控制”）。

消息的生成过程发生延迟或减慢

此问题的症状如下：

- 发送持久性消息时，`send()` 方法并没有返回，且客户机发生阻塞。
- 发送持久性消息时，客户机收到异常。
- 生产方客户机速度变慢。

本节详细说明了以下可能的原因：

- 消息服务器上堆满了待办事项，从而做出使消息生产方减慢的响应
- 代理无法将持久性消息保存到数据存储库中
- 代理的确认超时太短
- 生产方客户机遇到了 JVM 限制

消息服务器上堆满了待办事项，从而做出使消息生产方减慢的响应

堆满待办事项的服务器将消息堆积在代理内存中。

当物理目标内存中的消息数或消息字节数达到配置的限制时，代理会尝试根据指定的限制行为节约内存资源。下列限制行为会使消息生产方变慢：

- `FLOW_CONTROL`：代理不会立即确认收到持久性消息（这样就会阻塞生产方客户机）。
- `REJECT_NEWEST`：代理将拒绝新的持久性消息。

同样，如果代理范围的内存（对于所有物理目标）中消息数或消息字节数达到配置的限制，代理将尝试通过拒绝最新的消息来节约内存资源。

另外，如果达到了系统内存限制（由于物理目标或代理范围限制设置不正确），代理将采取越来越严格的操作来防止内存过载。这些操作包括限制消息生产方。

确认这是否就是问题的起因

如果某个消息因为达到了配置的消息限制而被代理拒绝，代理将返回以下消息：

```
JMSEException [C4036]: 服务器出错
```

代理还在代理日志中记录下列条目：

```
WARNING [B2011]: 存储来自 IMQconn 的 JMS 消息失败
```

该消息后面接有一条表明已达到限制的消息。如果消息限制位于物理目标上，则代理将生成类似下面的条目：

```
[B4120]: 无法在目标 destName 上存储消息，因为会超出 maxNumMsgs 的容量。
```

如果消息限制是代理范围的，则代理将生成类似下面的条目：

[B4024]：已经超出目前系统中的最大消息数，正在拒绝消息。

通常，您可以在发生拒绝之前按如下方式检查消息限制情况：

- 查询物理目标和代理，并检查其配置的消息限制设置。
- 使用相应的 `imqcmd` 命令，监视当前物理目标或代理（作为一个整体）中的消息数或消息字节数。有关可以监视的度量以及用来获取它们的命令的信息，请参见第 18 章“度量参考”。

解决此问题

有很多方法可以解决由于消息变得堆积而导致生产方变慢的问题。

- 修改物理目标（或代理范围）上的消息限制，请小心不要超出内存资源。

通常，应该根据每个目标来管理内存，这样才不会达到代理范围的消息限制。有关详细信息，请参见第 220 页“代理调整”。

- 将目标上的限制行为更改为当达到消息限制时并不减慢消息的生成，而是在内存中废弃消息。

例如，您可以指定 `REMOVE_OLDEST` 和 `REMOVE_LOW_PRIORITY` 限制行为，这些行为可以删除在内存中堆积的消息（请参见第 301 页表 15-1）。

代理无法将持久性消息保存到数据存储库中

如果代理无法访问数据存储库或者将持久性消息写入数据存储库，则生产方客户机将阻塞。如果达到了如前所述的目标或代理范围消息限制，也将发生此情况。

确认这是否就是问题的起因

如果代理无法写入数据存储库，它将在代理日志中生成下列条目之一：[B2011]：存储来自 `connectionID` 的 JMS 消息失败... 或 [B4004]：无法持续消息 `messageID`...

解决此问题

- 如果是内置持久性，则尝试增大基于文件的数据存储库的磁盘空间。
- 如果是 JDBC 兼容的数据存储库，则检查插入的持久性是否正确配置（请参见第 4 章“配置代理”）。如果是这样，请向数据库管理员咨询，以解决其他数据库问题。

代理的确认超时太短

如果连接太慢或消息服务器反应迟缓（由于 CPU 使用率太高或者内存资源不足导致），代理用于确认收到持久性消息的时间比连接工厂的 `imqAckTimeout` 属性的值允许的时间要长。

确认这是否就是问题的起因

如果超出了 `imqAckTimeout` 值，代理将返回以下消息：

```
JMSEException [C4000]: 包确认失败。
```

解决此问题

更改 `imqAckTimeout` 连接工厂属性的值（请参见第 160 页“连接工厂属性”）。

生产方客户机遇到了 JVM 限制

确认这是否就是问题的起因

- 查明客户机应用程序是否收到了“内存不足”错误。
- 使用诸如 `freeMemory()`、`MaxMemory()` 和 `totalMemory()` 等运行时方法检查 JVM 堆中的可用内存。

解决此问题

调整 JVM（请参见第 216 页“Java 虚拟机调整”）。

消息堆积

此问题的症状如下：

- 代理（或特定目标）中的消息数目或消息字节数随着时间稳定增加。

要查看消息是否在堆积，请检查代理中的消息数目或消息字节数如何随时间而改变，并比较已配置的限制。首先检查配置的限制：

```
imqcmd query bkr
```

（注：`imqcmd metrics bkr` 子命令不会显示此信息）。

然后检查每个目标中的消息堆积情况。

```
imqcmd list dst
```

要检查消息是否已超出配置的目标或代理范围限制，请在代理日志中检查如下条目：`WARNING [B2011]: 存储来自 ... 的 JMS 消息失败`。此条目后面接有另一个阐明限制已超出的条目。

- 消息的生成发生了延迟，或者生成的消息被代理拒绝。
- 消息到达使用方的时间过长。

本节详细说明了以下可能的原因：

- 主题目标上有非活动的长期订阅
- 队列中可以使用消息的使用方太少
- 消息使用方的处理速度太慢，跟不上消息生产方的速度
- 客户机的确认处理过程减慢了消息的使用
- 代理无法适应生成的消息
- 客户机代码缺陷：使用方不确认消息

主题目标上有非活动的长期订阅

如果长期订阅是非活动的，则消息会存储在目标中，直到相应的使用方变为活动状态且能够使用这些消息为止。

确认这是否就是问题的起因

检查每个主题目标上的长期订阅的状态：

```
imqcmd list dur -d destName
```

解决此问题

可以采用下列任意操作：

- 清除所有存在问题的长期订阅的消息（请参见第 112 页“管理长期订阅”）。
- 指定主题的消息限制以及限制行为属性（请参见第 301 页表 15-1）。例如，您可以指定 REMOVE_OLDEST 和 REMOVE_LOW_PRIORITY 限制行为，这些行为可以删除在内存中堆积的消息。
- 清除来自相应目标的所有消息（请参见第 123 页“清除物理目标”）。
- 限制消息可以在内存中保留的时间。您可以重写生产方客户机以便对每个消息都设置一个有效期值。您可以通过设置 `imqOverrideJMSEExpiration` 和 `imqJMSEExpiration` 连接工厂属性（请参见第 310 页“消息标题覆盖”）来覆盖共享一个连接的所有生产方的这些设置。

队列中可以使用消息的使用方太少

如果消息可以传送到活动使用方太少，队列目标可能会随着消息的堆积而变得堆满了待办事项。只要有下列任何原因，都会发生此情况：

- 目标的活动使用方太少。
- 使用方客户机建立连接失败。
- 活动使用方没有使用与队列中的消息匹配的选择器。

确认这是否就是问题的起因

要确定使用方不可用的原因，请检查目标上活动使用方的数目：

```
imqcmd metrics dst -n destName -t q -m con
```

解决此问题

您可以根据使用方不可用的原因采取下列任意操作：

- 通过启动其他使用方客户机来为队列创建更多的活动使用方。
- 调整 `imq.consumerFlowLimit` 代理属性以优化对多个使用方的队列传送（请参见第 221 页“多使用方队列性能”）。
- 指定队列的消息限制以及限制行为属性（请参见第 301 页表 15-1）。例如，您可以指定 `REMOVE_OLDEST` 和 `REMOVE_LOW_PRIORITY` 限制行为，这些行为可以删除在内存中堆积的消息。
- 清除来自相应目标的所有消息（请参见第 123 页“清除物理目标”）。
- 限制消息可以在内存中保留的时间。您可以重写生产方客户机以对每个消息都设置一个有效期值，也可以通过设置 `imqOverrideJMSEExpiration` 和 `imqJMSEExpiration` 连接工厂属性（请参见第 310 页“消息标题覆盖”）来覆盖共享一个连接的所有生产方的这些设置。

消息使用方的处理速度太慢，跟不上消息生产方的速度

在这种情况下，主题的订阅者或队列的接收者使用消息的速度要比生产方发送消息的速度慢。会有一个或多个目标因为这种不平衡而堆满了消息。

确认这是否就是问题的起因

检查消息流入和流出代理的速率：

```
imqcmd metrics bkr -m rts
```

然后检查每个单独目标的流速：

```
imqcmd metrics bkr -t destType -n destName -m rts
```

解决此问题

- 优化使用方客户机代码。
- 对于队列目标，增大活动使用方的数目（请参见第 221 页“多使用方队列性能”）。

客户机的确认处理过程减慢了消息的使用

有两个因素影响对客户机确认过程的处理：

- 在处理客户机确认的过程中会消耗大量的代理资源。结果，如果使用方客户机会一直阻塞到代理对客户机确认进行确认时为止，则对于这样的确认模式，消息的使用会变慢。
- JMS 有效负荷消息和 Message Queue 控制消息（例如客户机确认）共享同一连接。结果，控制消息可能会被 JMS 有效负荷消息阻挡，从而使消息的使用变慢。

确认这是否就是问题的起因

- 检查与包流相关的消息流。如果每秒包数与消息数的不成比例，则客户机确认可能有问题。
- 检查客户机是否收到下列消息：

JMSEException [C4000]: 包确认失败。

解决此问题

- 修改客户机使用的确认模式，例如切换到 DUPES_OK_ACKNOWLEDGE 或 CLIENT_ACKNOWLEDGE。
- 如果使用 CLIENT_ACKNOWLEDGE 或事务会话，则将更多数目的消息组合到一个确认中。
- 调整使用方和连接流控制参数（请参见第 221 页“客户机运行时消息流调整”）。

代理无法适应生成的消息

在这种情况下，消息流入代理的速度比代理可以将它们路由并发送到使用方的速度要快。代理的迟缓可能由下列任一或全部内容的限制所导致：CPU、网络套接字读 / 写操作、磁盘读 / 写操作、内存分页、持久性存储库或 JVM 内存限制。

确认这是否就是问题的起因

检查有无其他原因导致此问题。

解决此问题

- 升级计算机或数据存储库的速度。
- 使用代理群集将负荷分布到多个代理实例之间。

客户机代码缺陷：使用方不确认消息

消息会保留在目标中，直到消息所发送到的所有使用方都进行了确认为止。如果客户机没有确认已使用的消息，该消息会在目标中堆积，而不会删除。

例如，客户机代码可能会存在下列缺陷：

- 使用 CLIENT_ACKNOWLEDGEacknowledgment 或事务会话的使用方可能没有定期调用 Session.acknowledge() 或 Session.commit()。

- 使用 `AUTO_ACKNOWLEDGE` 会话的使用方可能因为某种原因而挂起。

确认这是否就是问题的起因

首先检查本节中列出的所有其他可能原因。其次，使用以下命令列出目标：

```
imqcmd list dst
```

请注意 `UnAcked` 标题下列出的消息数目是否与目标中的消息数目相同。`UnAcked` 标题下的消息已发送到使用方但未得到确认。如果此数目与消息总数相同，则说明代理已发送所有消息，正在等待确认。

解决此问题

请求应用程序开发者帮助调试此问题。

消息服务器吞吐量呈间歇性

此问题的症状如下：

- 消息的吞吐量间歇性地下降，然后又恢复正常性能。

本节详细说明了以下可能的原因：

- [代理的内存资源非常低](#)
- [正在发生 JVM 内存回收（垃圾收集）](#)
- [JVM 正在使用实时编译器来提高性能](#)

代理的内存资源非常低

由于目标和代理的限制设置得不正确，代理采取了越来越严格的措施以防止内存过载，这样就导致代理变得非常迟缓，直到消息的堆积得到清除为止。

确认这是否就是问题的起因

在代理日志中检查内存小的情况（`[B1089]`：内存很低，代理正在尝试释放资源），该情况后文会接有一个描述新内存状态和已用内存总量的条目。

另外请检查 JVM 堆中的可用内存：

```
imqcmd metrics bkr -m cxn
```

当总 JVM 内存接近 JVM 内存最大值时，可用内存就会很小。

解决此问题

- 调整 JVM（请参见第 216 页“[Java 虚拟机调整](#)”）。

- 增大系统交换空间。

正在发生 JVM 内存回收（垃圾收集）

内存回收会定期清扫整个系统，以释放内存。发生此操作时，所有的线程都会阻塞。要释放的内存量以及 JVM 堆的大小越大，因内存回收而导致的延迟就越长。

确认这是否就是问题的起因

监视计算机上的 CPU 使用。发生内存回收时，CPU 使用会下降。

另外，使用下列命令行选项启动代理：

```
-vmargs -verbose:gc
```

其标准输出指明发生内存回收的时间。

解决此问题

在多个 CPU 的计算机中，将内存回收设置为并行发生：

```
-XX:+UseParallelGC=true
```

JVM 正在使用实时编译器来提高性能

确认这是否就是问题的起因

检查有无其他原因导致此问题。

解决此问题

让系统运行一段时间，性能应该会有所改善。

消息无法到达使用方

此问题的症状如下：

- 使用方未收到生产方发送的消息。

本节详细说明了以下可能的原因：

- [限制行为导致消息在代理上被删除](#)
- [消息超时值即将到期](#)
- [时钟不同步](#)
- [使用方客户机未能启动在某个连接上的消息传送](#)

限制行为导致消息在代理上被删除

如果目标内存中的消息数目或消息字节数达到了配置限制，代理将尝试节约内存资源。当达到限制时，代理将采取下列三个可配置的行为，从而导致消息丢失：

- REMOVE_OLDEST: 删除最旧的消息
- REMOVE_LOW_PRIORITY: 删除根据消息时间排列时优先级最低的消息
- REJECT_NEWEST: 拒绝新的持久性消息

如果代理内存中的消息数目或消息字节数达到配置的限制，代理将尝试通过拒绝最新的消息来节约内存资源。

确认这是否就是问题的起因

检查停用消息队列，如第 244 页“停用消息队列包含消息”中所述。特别是，可以使用第 245 页“消息的数目或者其大小超过目标限制”中的说明。查找 REMOVE_OLDEST 或 REMOVE_LOW_PRIORITY 原因。

解决此问题

增加目标限制。例如：

```
imqcmd update dst -n MyDest -o maxNumMsgs=1000
```

消息超时值即将到期

代理将删除超时值已过期的消息。如果目标上完全堆满了消息，有效期值过短的消息将被删除。

确认这是否就是问题的起因

检查停用消息队列以查看消息是否超时。

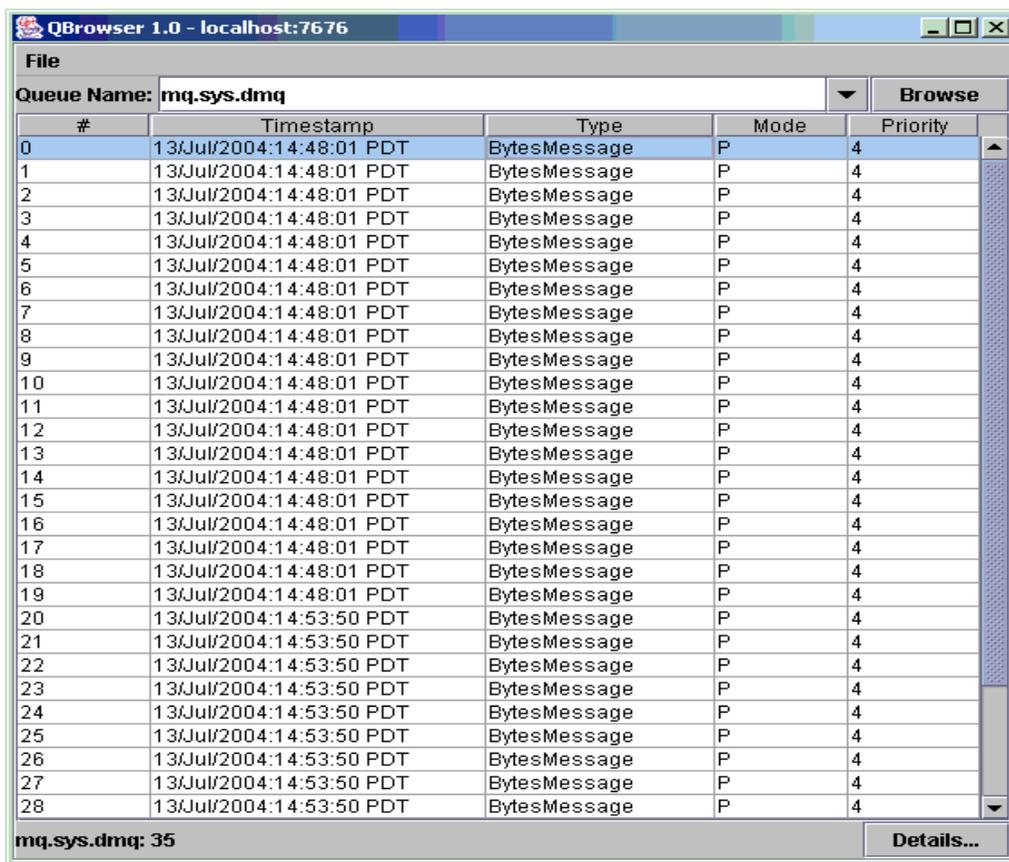
使用 QBrower 演示应用程序来查看 DMQ 内容。QBrower 演示程序位于操作系统特定的位置；要了解该位置，请参见附录 A “Message Queue 数据在特定操作系统中的位置”及查看“示例应用程序和位置”表。

下面是 Windows 中的一个调用示例：

```
cd \MessageQueue3\demo\applications\qbrowser java QBrower
```

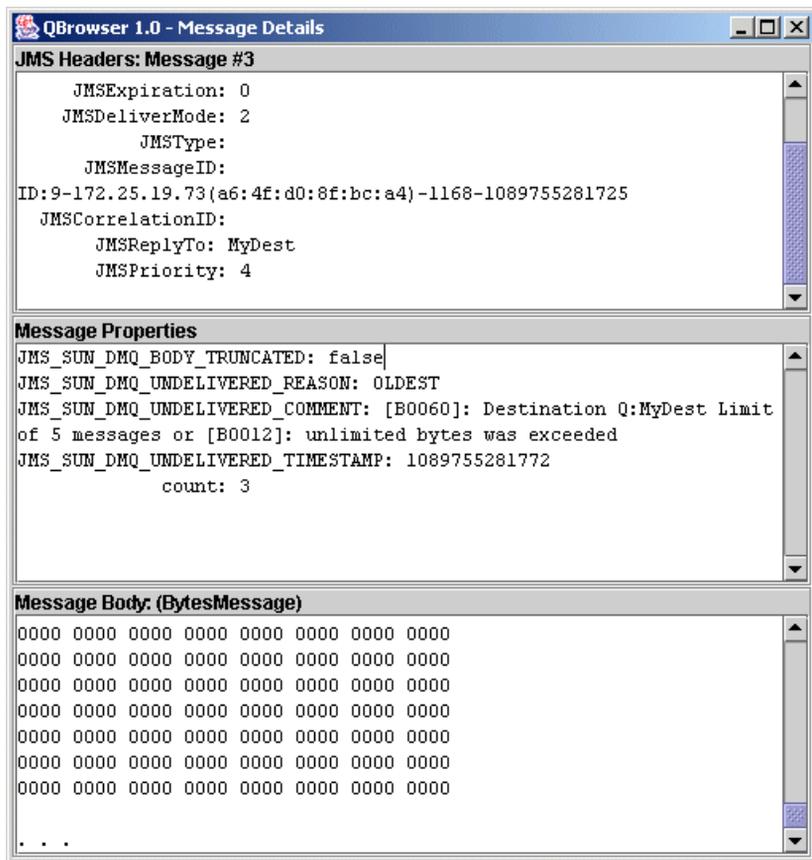
QBrower 主窗口出现后，选择队列名称 mq.sys.dmq 之后单击“浏览”。将会出现如下所示的列表。

图 12-1 QBrowser 窗口



双击消息可显示该消息的详细信息。

图 12-2 QBrowser 消息的详细信息



请注意消息的 `JMS_SUN_DMQ_UNDELIVERED_REASON` 属性的值是否为 `EXPIRED`。

解决此问题

联系应用程序开发者并请他们提高有效期值。

时钟不同步

如果时钟之间不同步，则代理对消息生命周期的计算可能有错误，从而导致消息超过它们的到期时间而被删除。

确认这是否就是问题的起因

在代理日志文件中，查找下列任一消息：B2102、B2103、B2104。这些消息均报告检测到可能存在时钟脉冲相位差。

解决此问题

检查您是否正在运行时间同步程序，如第 64 页“准备系统资源”中所述。

使用方客户端未能启动在某个连接上的消息传送

除非客户端代码建立了连接，并在该连接上启动了消息传送，否则消息将无法传送。

确认这是否是问题的起因

检查客户端代码是否能建立连接并启动消息传送。

解决此问题

重写客户端代码，以建立连接并启动消息传送。

停用消息队列包含消息

此问题的症状如下：

- 列出目标后，发现停用消息队列包含消息。例如，执行如下所示的命令。

```
imqcmd list dst
```

在提供用户名和密码后，将显示类似于以下内容的输出：

```
列出指定的代理上的所有目标：
-----
主机 主端口
-----
localhost      7676
-----
   名称      类型      状态      生产方      使用方      消息总数      未确认      平均大小
-----
MyDest      Queue     RUNNING   0           0           5             0           1177.0
mq.sys.dmq  Queue     RUNNING   0           0           35            0           1422.0
成功列出目标。
```

在本示例中，停用消息队列 `mq.sys.dmq` 包含 35 条消息。

本节详细说明了以下可能的原因：

- 消息的数目或者其大小超过目标限制
- 代理时钟和生产方时钟不同步
- 消息超时前，使用方未接收到消息

- 相对使用方数目而言，生产方太多
- 生产方比使用方的速度快
- 使用方太慢
- 客户机不提交消息
- 长期使用方处于不活动状态
- 发生意外的代理错误

消息的数目或者其大小超过目标限制

确认这是否就是问题的起因

使用 QBrower 演示应用程序来查看停用消息队列的内容。QBrower 演示程序位于操作系统特定的位置；要了解该位置，请参见附录 A “[Message Queue 数据在特定操作系统中的位置](#)” 及查看 “[示例应用程序和位置](#)” 表。

下面是 Windows 中的一个调用示例：

```
cd \MessageQueue3\demo\applications\qbrower java QBrower
```

QBrower 主窗口出现后，选择队列名称 mq.sys.dmq 之后单击 “浏览”。将会出现如第 242 页图 12-1 中所示的列表。

双击消息可显示该消息的详细信息。将会出现如第 243 页图 12-2 中所示的窗口。

请注意下列消息属性的值：

- JMS_SUN_DMQ_UNDELIVERED_REASON
- JMS_SUN_DMQ_UNDELIVERED_COMMENT
- JMS_SUN_DMQ_UNDELIVERED_TIMESTAMP

请注意 JMS 标题下面的 JMSDestination 值，以确定将要停用其消息的目标。

解决此问题

增加目标限制。例如：

```
imqcmd update dst -n MyDest -o maxNumMsgs=1000
```

代理时钟和生产方时钟不同步

要确认这是否就是问题的起因，请执行下列操作：

使用 QBrower 应用程序来查看停用消息队列中各消息的详细信息。检查 JMS_SUN_DMQ_UNDELIVERED_REASON 的值，查找原因为 EXPIRED 的消息。

在代理日志文件中，查找下列任一消息：B2102、B2103、B2104。这些消息均报告检测到可能存在时钟脉冲相位差。

解决此问题

检查您是否正在运行时间同步程序，如第 64 页“准备系统资源”中所述。

消息超时前，使用方未接收到消息

确认这是否就是问题的起因

使用 `QBrowser` 应用程序来查看停用消息队列中各消息的详细信息。检查 `JMS_SUN_DMQ_UNDELIVERED_REASON` 的值，查找原因为 `EXPIRED` 的消息。

检查目标中是否有任何使用方。例如：

```
imqcmd query dst -t q -n MyDest
```

检查列出的“当前活动使用方数”值。如果有活动使用方，则下面的两项之一为真：

- 使用方的连接暂停。
- 相对使用方执行速度而言，消息超时时间太短。

解决此问题

请求应用程序开发者提高消息的有效期值。

相对使用方数目而言，生产方太多

确认这是否就是问题的起因

使用 `QBrowser` 应用程序来查看停用消息队列中各消息的详细信息。检查 `JMS_SUN_DMQ_UNDELIVERED_REASON` 的值。

如果原因是 `REMOVE_OLDEST` 或 `REMOVE_LOW_PRIORITY`，请使用 `imqcmd query dst` 命令来检查目标中的生产方和使用方的数目。如果生产方的数目超过使用方的数目，生产率可能会远远超出消费率。

解决此问题

添加更多的使用方客户机，或将目标设置为使用 `FLOW_CONTROL` 限制行为。`FLOW_CONTROL` 限制行为使用消费率来控制生产率。

使用如下示例所示的命令来启动流控制行为：

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

生产方比使用方的速度快

确认这是否就是问题的起因

要确定较慢的使用方是否会导致生产方速度降低，请将目标限制行为设置为 `FLOW_CONTROL`。`FLOW_CONTROL` 限制行为使用消费率来控制生产率。

使用如下示例所示的命令来启动流控制行为：

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

通过执行如下示例所示的命令，使用度量来检查目标的输入和输出：

```
imqcmd metrics dst -n myDst -t q -m rts
```

在度量输出中，检查下列值：

- `Msgs/sec Out`

此值显示代理每秒删除多少条消息。在所有使用方确认收到消息后，代理将删除这些消息，因此，该度量反映了消费率。

- `Msgs/sec In`

此值显示代理每秒从生产方接收多少条消息。此度量反映了生产率。

由于流控制使生产与消费协调一致，因此请注意生产是否减慢或停止。如果速率减慢或停止，则说明生产方和使用方的处理速度不一致。

也可以使用 `imqcmd list dst` 命令，检查未确认的 (`UnAcked`) 发送消息的数目。如果未确认的消息数目小于目标大小，则表明目标尚有额外的容量，并受到客户机流控制的抑制。

解决此问题

如果生产率始终高于消费率，可以考虑规则地使用流控制以使系统保持协调一致。

此外，使用后面各节的内容，考虑并尝试消除下列可能的因素：

- 使用方太慢
- 客户机不提交消息
- 使用方未能确认消息
- 长期使用方处于不活动状态
- 发生意外的代理错误

使用方太慢

确认这是否就是问题的起因

使用度量来确定生产和消费的速度，如第 247 页“生产方比使用方的速度快”中所述。

解决此问题

尝试下列一项或多项：

- 将目标设置为使用 FLOW_CONTROL 限制行为。使用如下所示的命令：

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

使用流控制将生产率降低到消费率，防止消息在代理中堆积。生产方应用程序保留消息，直到目标可以及时处理它们，从而降低了过期风险。

- 向应用程序开发者了解生产方是以稳定的速率发送消息，还是周期性成组传送。如果应用程序发送成组消息，请按照下一项中的说明增加目标限制。
- 根据消息数目和 / 或字节数目增加目标限制。

要更改目标中的消息数目，请输入具有下列格式的命令：

```
imqcmd update dst -n destName -t {q/t} -o maxNumMsgs=number
```

要更改目标的大小，请输入具有下列格式的命令：

```
imqcmd update dst -n destName -t {q/t} -o maxTotalMsgBytes=number
```

请注意，增加限制会增加代理使用的内存数目。如果限制过高，代理可能会耗尽内存，从而无法处理消息。

- 考虑您是否可以在高生产负荷级别期间接受消息丢失。

客户机不提交消息

确认这是否就是问题的起因

与应用程序开发者进行确认以查明应用程序是否使用事务。如果应用程序使用事务，则按如下所示列出活动事务：

```
imqcmd list txn
```

下面是一个命令输出示例：

| 事务 ID | 状态 | 用户名 | 消息数 / 确认数 | 创建时间 |
|---------------------|---------|-------|-----------|---------------------|
| 6800151593984248832 | STARTED | guest | 3/2 | 7/19/04 11:03:08 AM |

请注意消息的数目和确认的数目。

如果消息的数目很高，则生产方可能正在发送个别的消息，而未能提交事务。代理在收到提交之前，无法路由并传送该事务的消息。

如果确认的数目很高，则使用方可能正在发送个别消息的确认，而未能提交事务。代理在收到提交之前，无法删除该事务的确认。

解决此问题

联系应用程序开发者来解决编码错误。

使用方未能确认消息

确认这是否就是问题的起因

联系应用程序开发者以确定应用程序使用的是基于系统的确认还是基于客户机的确认。如果应用程序使用基于系统的确认，则跳过本节。

如果应用程序使用基于客户机的确认（CLIENT_ACKNOWLEDGE 类型），请先减少客户机中存储的消息数目。使用如下所示的命令：

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=1
```

其次，确定代理缓冲消息的原因是由于使用方速度慢，还是由于使用方处理消息的速度很快而未对其进行确认。

使用下列命令列出目标：

```
imqcmd list dst
```

在提供用户名和密码后，将显示类似于以下内容的输出：

| 列出指定的代理上的所有目标： | | | | | | | |
|----------------|-------|---------|-----|-----|------|-----|--------|
| 主机 | 主端口 | | | | | | |
| localhost | 7676 | | | | | | |
| 名称 | 类型 | 状态 | 生产方 | 使用方 | 消息总数 | 未确认 | 平均大小 |
| MyDest | Queue | RUNNING | 0 | 0 | 5 | 200 | 1177.0 |
| mq.sys.dmq | Queue | RUNNING | 0 | 0 | 35 | 0 | 1422.0 |

成功列出目标。

未确认数值表示代理已发送且正在等待确认的消息数。如果未确认数值较高或不断增加，则说明代理正在发送消息，因此未等待速度较慢的使用方。还说明使用方未确认消息。

解决此问题

联系应用程序开发者来解决编码错误。

长期使用方处于不活动状态

确认这是否就是问题的起因

使用下列命令格式查看主题的长期订户：

```
imqcmd list dur -d topicName
```

解决此问题

- 使用 `imqcmd purge dur` 命令清除长期使用方。
- 重新启动使用方应用程序。

发生意外的代理错误

确认这是否就是问题的起因

使用 QBrowser 对消息进行检查，如第 247 页“生产方比使用方的速度快”中所述。

如果 `JMS_SUN_DMQ_UNDELIVERED_REASON` 的值是 `ERROR`，则说明代理发生错误。

解决此问题

- 检查代理日志文件来查找相关错误。
- 联系 Sun 技术支持以报告代理问题。

参考

- 第 13 章 命令参考
- 第 14 章 代理属性参考
- 第 15 章 物理目标属性参考
- 第 16 章 受管理对象的属性参考
- 第 17 章 JMS 资源适配器属性参考
- 第 18 章 度量参考

命令参考

本章包含的一个小节介绍了通用命令行语法，并提供了各个 Message Queue 命令的参考信息。本章包含以下小节：

- 第 254 页 “命令行语法”
- 第 255 页 “imqbrokerd”
- 第 260 页 “imqcmd”
- 第 269 页 “imqobjmgr”
- 第 271 页 “imqdbmgr”
- 第 273 页 “imqusermgr”
- 第 275 页 “imqsvcadm”
- 第 277 页 “imqkeytool”

命令行语法

Message Queue 命令行实用程序是 shell 命令。实用程序的名称是命令，其子命令或选项是传递给该命令的参数。因此，没有、也不需要用于启动或退出实用程序的命令。

所有的命令行实用程序均共享以下命令语法：

```
Utility_Name [subcommand] [argument] [[-option_name [-option_argument]]...]
```

Utility_Name 指定 Message Queue 实用程序的名称，例如，`imqcmd`、`imqobjmgr` 和 `imqusermgr` 等。

输入命令的规则

输入命令时要遵循的一些通用规则：

- 在子命令之后指定选项（如果此实用程序还接受变量类型的操作数，则还包括变量）。
- 如果选项值包含空格，请将整个值置于引号中。将属性 - 值对置于引号中通常是最安全的做法。
- 如果在命令行中指定了 `-v`（版本）或 `-h/-H`（帮助）选项，所有此命令行中的其他选项均不执行。
- 用空格分隔子命令、变量、选项和选项变量。

命令行示例

以下是不包含子命令子句的一个命令行示例。此命令启动默认代理。

```
imqbrokerd
```

以下命令更复杂一些。此命令用于销毁类型为 `queue`、名为 `myQueue` 的目标。验证是基于用户 `admin` 执行的；此命令将提示用户输入用户密码。`-f` 选项指定将没有确认；`-s` 选项指定在静音模式下执行此命令。

```
imqcmd destroy dst -t q -n myQueue -u admin -f -s
```

通用命令选项

表 13-1 介绍了所有 Message Queue 管理实用程序的通用选项。在命令行中，必须在子命令之后指定这些选项。选项的输入顺序是任意的。

表 13-1 通用 Message Queue 命令行选项

| 选项 | 说明 |
|-----------------------|---|
| -h | 显示指定实用程序的使用帮助。 |
| -H | 显示详细的使用帮助，包括属性列表和示例（只支持 imqcmd 和 imqobjmgr）。 |
| -s | 打开静音模式：不显示输出。对于 imqbrokerd，指定为 -silent。 |
| -v | 显示版本信息。 |
| -f | 执行给定操作，而不提示用户进行确认。 |
| -pre | （仅用于 imqobjmgr）打开预览模式，允许用户查看命令行中其他部分的执行效果而无需实际执行此命令。此项功能有助于检查默认属性值。 |
| -javahome <i>path</i> | 指定要使用的替代 Java 2 兼容运行时（默认情况下使用系统上的运行时或 Message Queue 附带的运行时）。 |

imqbrokerd

imqbrokerd 命令将启动一个代理。命令行选项覆盖代理配置文件中的值，但仅对当前代理会话有效。

语法

```
imqbrokerd [[ -Dproperty=value]...]
  [ -backup fileName]
  [ -cluster "[broker1] [[,broker2]...]"
  [ -dbuser userName]
  [ -force]
  [ -h|-help]
  [ -javahome path]
  [ -license licenseName]
  [ -loglevel level]
  [ -metrics interval]
```

```

[ -name instanceName]
[ -passfile fileName]
[ -port number]
[ -remove instance]
[ -reset data]
[ -restore fileName]
[ -shared]
[ -silent|-s] [ -tty]
[ -upgrade-store-nobackup]
[ -version]
[ -vmargs arg1 [[arg2]...]]

```

命令选项

表 13-2 说明了 `imqbrokerd` 命令的选项以及受到各个选项影响的配置属性（如果存在）。

表 13-2 `imqbrokerd` 选项

| 选项 | 所影响的属性 | 说明 |
|---|--|--|
| <code>-backup fileName</code> | 不影响任何属性。 | 仅适用于代理群集。将主代理的配置更改记录备份到指定的文件。请参见第 182 页“管理配置更改记录”。 |
| <code>-cluster "broker1 [[,broker2]...]"</code> | 用要连接的代理的列表覆盖 <code>imq.cluster.brokerlist</code> 。 | 仅适用于代理群集。连接至指定的主机和端口上的所有代理。该列表将与 <code>imq.cluster.brokerlist</code> 属性中的列表合并。如果没有指定 <code>host</code> 的值，将使用 <code>localhost</code> 。如果没有指定 <code>port</code> 的值，将使用值 7676。关于如何使用该选项连接多个代理的详细信息，请参见第 177 页“使用代理群集”。 |
| 其中 <code>broker</code> 可以是 | | |
| <ul style="list-style-type: none"> • <code>host</code> • <code>:port</code> • <code>host:port</code> | | |
| <code>-dbpassword password</code> | 使用指定密码覆盖 <code>imq.persist.jdbc.password</code> | 指定插入的 JDBC 兼容数据仓库的密码。不赞成使用此选项，在今后的版本中会将其删除。使用下列替代方法之一： <ul style="list-style-type: none"> • 在命令行中省略密码，以使命令提示您输入密码。 • 使用 <code>-passfile</code> 选项指定包含数据库密码的文件。 |
| <code>-dbuser userName</code> | 使用指定用户名覆盖 <code>imq.persist.jdbc.user</code> | 指定插入的 JDBC 兼容数据仓库的用户名。请参见第 93 页“设置持久性存储”。 |

表 13-2 imqbrokerd 选项 (续)

| 选项 | 所影响的属性 | 说明 |
|-------------------------------------|---|---|
| <code>-Dproperty=value</code> | 设置系统属性。覆盖实例配置文件中的相关属性值。 | 将指定属性设置为指定值。有关代理配置属性的信息，请参见第 14 章“代理属性参考”。 注意： 请仔细检查使用 <code>-D</code> 选项设置的属性的拼写和格式。如果传送了不正确的值，系统不会向您发出警告， Message Queue 将无法对这些属性进行设置。 |
| <code>-force</code> | 不影响任何属性。 | 执行操作，而无需用户确认。此选项仅用于 <code>-remove instance</code> 和 <code>-upgrade-store-nobackup</code> 选项，通常要求确认。 |
| <code>-h -help</code> | 不影响任何属性。 | 显示帮助。不执行命令行上的其他选项。 |
| <code>-javahome path</code> | 不影响任何属性。 | 指定替代 Java 2 兼容 JDK 的路径。默认使用捆绑的运行库。 |
| <code>-ldappassword password</code> | 使用指定密码覆盖 <code>imq.user_repository.ldap.password</code> | 指定访问 LDAP 用户系统信息库的密码。不赞成使用此选项，在今后的版本中会将其删除。使用下列替代方法之一： <ul style="list-style-type: none"> 在命令行中省略密码，以使命令提示您输入密码。 使用 <code>-passfile</code> 选项指定包含 LDAP 密码的文件。 |
| <code>-license [licenseName]</code> | 不影响任何属性。 | 指定装入一个与 Message Queue 产品版本的默认许可证不同的许可证。如果不指定许可证名称，该选项将列出系统中安装的所有许可证。根据所安装的 Message Queue 版本的不同， <code>licenseName</code> 的取值也不同： <code>pe</code> （平台版，只具有基本功能）、 <code>try</code> （平台版的 90 天试用企业功能）和 <code>unl</code> （企业版）。 |
| <code>-loglevel level</code> | 使用指定级别覆盖 <code>imq.broker.log.level</code> 。 | 将日志记录级别指定为下列四项之一： <code>NONE</code> 、 <code>ERROR</code> 、 <code>WARNING</code> 或 <code>INFO</code> 。默认值是 <code>INFO</code> 。 |
| <code>-metrics interval</code> | 使用指定秒数覆盖 <code>imq.metrics.interval</code> 。 | 指定将代理度量写入记录程序的时间间隔（以秒为单位）。 |
| <code>-name instanceName</code> | 将 <code>imq.instanceName</code> 设置为指定名称。 | 指定该代理的实例名称，并使用相应的实例配置文件。如果不指定代理名称，实例名将设置为 <code>imqbroker</code> 。 注： 如果在同一台主机上运行了多个代理实例，每个实例必须具有唯一的名称。 |

表 13-2 imqbrokerd 选项 (续)

| 选项 | 所影响的属性 | 说明 |
|---|--|---|
| -passfile <i>fileName</i> | 覆盖 <code>imq.passfile.enabled</code> 并将其设置为 <code>true</code> 。使用包含该文件的路径覆盖 <code>imq.passfile.dirpath</code> 。使用该文件的名称覆盖 <code>imq.passfile.name</code> 。 | 指定文件名，从该文件中可以读取 <code>imqcmd</code> 命令实用程序、SSL 密钥库、LDAP 用户系统信息库、JDBC 兼容数据库或它们的任意组合的密码。有关详细信息，请参见第 153 页“使用密码文件”。 |
| -password <i>keypassword</i> | 使用指定密码覆盖 <code>imq.keystore.password</code> 。 | 指定 SSL 证书密钥库的密码。不赞成使用此选项，在今后的版本中会将其删除。使用下列替代方法之一： <ul style="list-style-type: none"> 在命令行中省略密码，以使命令提示您输入密码。 使用 <code>-passfile</code> 选项指定包含 SSL 证书密钥库密码的文件。 |
| -port <i>number</i> | 使用指定号码覆盖 <code>imq.portmapper.port</code> 。 | 指定代理的端口映射器的端口号。默认情况下，将其设置为 7676。要在同一台服务器上运行两个代理实例，每个代理的端口映射器必须拥有不同的端口号。Message Queue 客户机使用此端口号连接到代理实例。 |
| -remove instance | 不影响任何属性。 | 导致代理实例被删除：删除与实例相关联的实例配置文件、日志文件、持久性存储与其他文件和目录。除非指定 <code>-force</code> 选项，否则需要用户确认。 |
| -reset store messages durables props | 不影响任何属性。 | 根据给定变量，重置代理实例的数据存储库（或数据存储库的子集）或配置属性。 重置数据存储库将清除所有的持久性数据，包括持久性消息、长期订阅和事务信息。这使您可以启动一个完全清空的代理实例。也可以仅清除所有的持久性消息或仅清除所有的长期订阅。（如果不希望下次重新启动代理实例时重置持久性存储，请勿使用 <code>-reset</code> 选项重新启动代理实例。） 重置代理的属性，用一个空文件代替现有的实例配置文件 (<code>config.properties</code>)：所有的属性将使用默认值。 |
| -restore <i>fileName</i> | 不影响任何属性。 | 仅适用于代理群集。使用指定备份文件替换主代理的配置更改记录。该文件必须是原来用 <code>-backup</code> 选项创建的那个备份文件。请参见第 182 页“管理配置更改记录”。 |

表 13-2 imqbrokerd 选项 (续)

| 选项 | 所影响的属性 | 说明 |
|--|--|---|
| -shared | 覆盖 imq.jms.threadpool_model 并将其设置为 shared。 | 指定使用共享线程池模型来实现 jms 连接服务。在该模型中, 连接之间共享线程, 以增加代理实例支持的连接数量。 |
| -silent -s | 覆盖 imq.log.console.output 并将其设置为 NONE。 | 停止向控制台记录日志信息。 |
| -tty | 覆盖 imq.log.console.output 并将其设置为 ALL | 指定在控制台上显示所有消息。默认情况下, 只显示 WARNING 和 ERROR 级别的消息。 |
| -upgrade-store-nobackup | 不影响任何属性。 | 指定从不兼容版本升级到 Message Queue 3.5 或 Message Queue 3.5 SPx 将自动删除原有的数据存储库。有关其他详细信息, 请参见 Message Queue Installation Guide 。 |
| -version | 不影响任何属性。 | 显示所安装产品的版本号。 |
| -vmargs <i>arg1</i> [<i>arg2</i>]... | 不影响任何属性。 | 指定传送到 Java VM 的变量。变量之间以空格分隔。如要传送多个变量, 或是变量内包含空格, 请使用闭合的引号。例如: imqbrokerd -tty -vmargs "-Xmx128m -Xincgc" 只能在命令行中传递这些参数。config.props 文件中没有相关联的配置属性。 |

另请参见

有关使用 imqbrokerd 及命令示例的详细信息, 请参见第 64 页“以交互方式启动代理”。

imqcmd

imqcmd 命令实用程序使您可以管理代理及其服务。

语法

```
imqcmd subcommand argument [options]
imqcmd -h|H
imqcmd -v
```

子命令

必须始终将子命令与 imqcmd 配合使用，除非您希望显示帮助或显示产品版本。
表 13-3 列出了 imqcmd 子命令，并指定了该子命令的参考信息所在的位置。

表 13-3 imqcmd 子命令

| 子命令和参数 | 说明 | 参考 |
|-------------|-------------------------|---------------------|
| commit txn | 提交事务。 | 第 266 页 “事务管理子命令” |
| destroy dur | 销毁长期订阅。 | 第 266 页 “长期订阅子命令” |
| list cxn | 列出代理的连接。 | 第 265 页 “连接子命令” |
| list dur | 列出对主题的长期订阅。 | 第 266 页 “长期订阅子命令” |
| list svc | 列出代理中的服务。 | 第 264 页 “连接服务管理子命令” |
| list txn | 列出代理中的事务。 | 第 266 页 “事务管理子命令” |
| metrics bkr | 显示代理度量。 | 第 261 页 “代理管理子命令” |
| metrics svc | 显示服务度量。 | 第 264 页 “连接服务管理子命令” |
| pause bkr | 暂停代理中的所有服务。 | 第 261 页 “代理管理子命令” |
| pause svc | 暂停代理中的单项服务。 | 第 264 页 “连接服务管理子命令” |
| purge dur | 清除长期订阅中的所有消息，但不销毁该长期订阅。 | 第 266 页 “长期订阅子命令” |
| query bkr | 查询并显示代理中的信息。 | 第 261 页 “代理管理子命令” |
| query cxn | 查询并显示连接中的信息。 | 第 265 页 “连接子命令” |

表 13-3 imqcmd 子命令 (续)

| 子命令和参数 | 说明 | 参考 |
|--------------|----------------|---------------------|
| query svc | 查询并显示服务中的信息。 | 第 264 页 “连接服务管理子命令” |
| query txn | 查询并显示事务中的信息。 | 第 266 页 “事务管理子命令” |
| reload cls | 重新装入代理群集配置。 | 第 261 页 “代理管理子命令” |
| restart bkr | 重新启动当前运行的代理实例。 | 第 261 页 “代理管理子命令” |
| resume bkr | 恢复代理中的所有服务。 | 第 261 页 “代理管理子命令” |
| resume svc | 恢复一项服务。 | 第 264 页 “连接服务管理子命令” |
| rollback txn | 回滚某项事务。 | 第 266 页 “事务管理子命令” |
| shutdown bkr | 关闭代理实例。 | 第 261 页 “代理管理子命令” |
| update bkr | 更新代理属性。 | 第 261 页 “代理管理子命令” |
| update svc | 更新服务属性。 | 第 264 页 “连接服务管理子命令” |

imqcmd 命令实用程序也包含用于代理中物理目标的子命令。第 6 章 “管理物理目标” 中介绍了目标子命令。

以下各节按功能列出了 imqcmd 子命令。

代理管理子命令

表 13-4 列出了用于管理代理的 imqcmd 子命令。如果未指定主机名和端口, 则会假定为默认值 (localhost:7676)。

表 13-4 用于管理代理的 imqcmd 子命令

| 子命令语法 | 说明 |
|---|--|
| <pre>metrics bkr [-b <i>hostName:port</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</pre> | <p>显示默认代理的代理度量，或显示指定主机和端口上的代理的代理度量。</p> <p>使用 <code>-m</code> 选项指定要显示的度量类型：</p> <p>ttl 显示消息和包流入和流出代理的度量。（默认度量类型）</p> <p>rts 显示消息和包流入和流出代理的速率（每秒）的度量。</p> <p>cxn 显示连接、虚拟内存堆和线程。</p> <p>使用 <code>-int</code> 选项指定显示度量的时间间隔（以秒为单位）。默认值为 5 秒。</p> <p>使用 <code>-msp</code> 选项指定在输出中显示的样例数量。默认值为无限数（无穷大）。</p> |
| <pre>pause bkr [-b <i>hostName:port</i>]</pre> | <p>暂停默认代理或指定主机和端口上的代理。请参见第 104 页“暂停和恢复代理”。</p> |
| <pre>query bkr -b <i>hostName:port</i></pre> | <p>列出默认代理或指定主机和端口上的代理的当前属性设置。还将列出与指定代理连接且正在运行的代理（在多代理群集中）。</p> |
| <pre>reload cls</pre> | <p>仅适用于代理群集。强制群集中的所有代理重新装入 <code>imq.cluster.brokerlist</code> 属性并更新群集信息。有关详细信息，请参见第 180 页“将代理添加到群集”。</p> |
| <pre>restart bkr [-b <i>hostName:port</i>]</pre> | <p>使用代理启动时指定的选项，关闭并重新启动默认代理或指定主机和端口上的代理。</p> |
| <pre>resume bkr [-b <i>hostName:port</i>]</pre> | <p>恢复默认代理或指定主机和端口上的代理。</p> |
| <pre>shutdown bkr [-b <i>hostName:port</i>]</pre> | <p>关闭默认代理或指定主机和端口上的代理。</p> |
| <pre>update bkr [-b <i>hostName:port</i>] -o <i>attribute=value</i> [-o <i>attribute=value1</i>]...</pre> | <p>更改默认代理或指定主机和端口上的代理的指定属性。</p> |

物理目标管理子命令

表 13-5 列出了用于管理物理目标的 imqcmd 子命令。如果未指定主机名和端口，则会假定为默认值 (localhost:7676)。

表 13-5 用于管理目标的 imqcmd 子命令

| 子命令语法 | 说明 |
|---|--|
| <code>compact dst [-t <i>destType</i> -n <i>destName</i>]</code> | 为指定类型和名称的目标压缩内置的、基于文件的数据存储库。如果未指定目标类型和名称，则会压缩所有目标。在压缩之前必须暂停目标。 |
| <code>create dst -t <i>destType</i> -n <i>destName</i> [-o <i>attribute=value</i>] [-o <i>attribute=value1</i>]...</code> | 创建指定类型、名称和属性的目标。目标名称必须只包含字母数字字符（不包括空格），可以以字母字符或 "_" 和 "\$" 字符开头。它不能以字符串 "mq." 开头。 如果群集的主代理暂时不可用，则不能在群集中执行此操作。 |
| <code>destroy dst -t <i>destType</i> -n <i>destName</i></code> | 销毁指定类型和名称的目标。不能销毁系统创建的目标，如停用消息队列。 如果群集的主代理暂时不可用，则不能在群集中执行此操作。 |
| <code>list dst [-t <i>destType</i>] [-tmp]</code> | 列出指定类型的所有目标，同时包含列出临时目标的选项。 类型参数可有两个值： <i>destType</i> = q（队列） <i>destType</i> = t（主题） 如果未指定类型，将列出所有类型的所有目标。 |
| <code>metrics dst -t <i>destType</i> -n <i>destName</i> [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</code> | 显示指定类型和名称的目标的度量信息。 使用 -m 选项指定要显示的度量类型： ttl 显示消息和包流入和流出目标以及占用内存的度量。（默认度量类型） rts 显示消息和包（每秒）流入和流出目标的速率度量及其他比率信息。 con 显示使用方相关度量。 dsk 显示磁盘使用情况度量。 使用 -int 选项指定显示度量的时间间隔（以秒为单位）。默认值为 5 秒。 使用 -msp 选项指定在输出中显示的样例数量。默认值为无限数（无穷大）。 |

表 13-5 用于管理目标的 imqcmd 子命令（续）

| 子命令语法 | 说明 |
|---|--|
| <pre>pause dst [-t destType -n destName] [-pst pauseType]</pre> | 对于指定类型和名称的目标，暂停将消息传送给使用方 (-pst CONSUMERS)，或暂停从生产方传送消息 (-pst PRODUCERS)，或二者同时暂停 (-pst ALL)。如果未指定目标类型和名称，则暂停所有目标。默认值为 ALL。 |
| <pre>purge dst -t destType -n destName</pre> | 清除指定类型和名称的目标中的消息。 |
| <pre>query dst -t destType -n destName</pre> | 列出指定类型和名称的目标的相关信息。 |
| <pre>resume dst [-t destType -n destName]</pre> | 恢复指定类型和名称的暂停目标中的消息传送。如果未指定目标类型和名称，则恢复所有目标。 |
| <pre>update dst -t destType -n destName -o attribute=value [-o attribute=value1]...</pre> | 更新指定目标上指定属性的值。 属性名称可以是表 15-1 中所描述的任何属性，除非该目标是停用消息队列 mq.sys.dmq。 |

连接服务管理子命令

表 13-6 列出了用于管理连接服务的 imqcmd 子命令。如果未指定主机名和端口，则会假定为默认值 (localhost:7676)。

表 13-6 用于管理连接服务的 imqcmd 子命令

| 子命令语法 | 说明 |
|--|-----------------------------|
| <pre>list svc [-b hostName:port]</pre> | 列出默认代理或指定主机和端口上的代理中的所有连接服务。 |

表 13-6 用于管理连接服务的 imqcmd 子命令（续）

| 子命令语法 | 说明 |
|---|--|
| <pre>metrics svc -n <i>serviceName</i> [-b <i>hostName:port</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</pre> | <p>显示默认代理或指定主机和端口上的代理中指定服务的度量。</p> <p>使用 <code>-m</code> 选项指定要显示的度量类型：</p> <p>ttl 显示消息和包按指定服务方式流入和流出代理的度量。（默认度量类型）</p> <p>rts 显示消息和包按指定连接服务方式流入和流出代理的速率（每秒）度量。</p> <p>cxn 显示连接、虚拟内存堆和线程。</p> <p>使用 <code>-int</code> 选项指定显示度量的时间间隔（以秒为单位）。默认值为 5 秒。</p> <p>使用 <code>-msp</code> 选项指定在输出中显示的样例数量。默认值为无限数（无穷大）。</p> |
| <pre>pause svc -n <i>serviceName</i> [-b <i>hostName:port</i>]</pre> | <p>暂停运行在默认代理或指定主机和端口上的代理中的指定服务。不能暂停管理服务。</p> |
| <pre>query svc -n <i>serviceName</i> [-b <i>hostName:port</i>]</pre> | <p>显示运行在默认代理或指定主机和端口上的代理中的指定服务的信息。</p> |
| <pre>resume svc -n <i>serviceName</i> [-b <i>hostName:port</i>]</pre> | <p>恢复运行在默认代理或指定主机和端口上的代理中的指定服务。</p> |
| <pre>update svc -n <i>serviceName</i> [-b <i>hostName:port</i>] [-o <i>attribute=value</i>] [-o <i>attribute=value1</i>]...</pre> | <p>更新运行在默认代理或指定主机和端口上的代理中的指定服务的指定属性。有关服务属性的说明，请参见第 283 页“连接服务属性”。</p> |

连接子命令

表 13-7 列出了应用于连接的 imqcmd 子命令。如果没有指定主机名和端口，则会假定为 localhost 和 7676。

表 13-7 用于管理连接服务的 imqcmd 子命令

| 子命令语法 | 说明 |
|---|--|
| <pre>list cxn [-svn <i>serviceName</i>] [-b <i>hostName:port</i>]</pre> | <p>列出默认代理或指定主机和端口上的代理中指定服务名称的所有连接。如果未指定服务名称，将列出所有连接。</p> |
| <pre>query cxn -n <i>connectionID</i> [-b <i>hostName:port</i>]</pre> | <p>显示默认代理或指定主机和端口上的代理中的指定连接的信息。</p> |

长期订阅子命令

表 13-8 概述了 imqcmd 长期订阅子命令。如果未指定主机名和端口，则会假定默认值 (localhost:7676)。

表 13-8 用于管理长期订阅的 imqcmd 子命令

| 子命令 | 说明 |
|---|---|
| <code>list dur -d <i>destName</i></code> | 列出指定目标的所有长期订阅。 |
| <code>destroy dur -n <i>subscrName</i> -c <i>client_id</i></code> | 销毁指定客户机标识符的指定长期订阅。 如果群集的主代理暂时不可用，则不能在群集中执行此操作。 |
| <code>purge dur -n <i>subscrName</i> -c <i>client_id</i></code> | 清除指定客户机标识符的指定长期订阅的所有消息。 |

事务管理子命令

表 13-9 概述了 imqcmd 事务子命令。如果未指定主机名和端口，则会假定默认值 (localhost:7676)。

表 13-9 用于管理事务的 imqcmd 子命令

| 子命令 | 说明 |
|--|--------------|
| <code>list txn</code> | 列出代理跟踪的所有事务。 |
| <code>query txn -n <i>transaction_id</i></code> | 列出指定事务的相关信息。 |
| <code>commit txn -n <i>transaction_id</i></code> | 提交指定的事务。 |
| <code>rollback txn -n <i>transaction_id</i></code> | 回滚指定的事务。 |

命令选项

表 13-10 列出了 imqcmd 命令的选项。

表 13-10 imqcmd 选项

| 选项 | 说明 |
|---------------------------|---|
| -b <i>hostName:port</i> | 指定代理的主机名及其端口号。默认值为 localhost:7676。 要仅指定端口，请使用：-b :7878 要仅指定主机名，请使用：-b somehost |
| -c <i>clientID</i> | 指定订阅某个主题的长期订户的 ID。请参见第 112 页“管理长期订阅”。 |
| -d <i>destinationName</i> | 指定主题名。与 list dur 和 destroy dur 子命令一起使用。请参见第 112 页“管理长期订阅”。 |
| -f | 执行操作，而无需用户确认。 |
| -h | 显示使用帮助。不执行命令行上的其他选项。 此选项不需要用户名和密码。 |
| -H | 显示使用帮助、属性列表和示例。不执行命令行上的其他选项。 此选项不需要用户名和密码。 |
| -int <i>interval</i> | 指定 metrics bkr、metrics dst 和 metrics svc 子命令显示度量输出的时间间隔，以秒为单位。 |
| -javahome <i>path</i> | 指定要使用的替代 Java 2 兼容运行时（默认情况下使用系统上的运行时或 Message Queue 附带的运行时）。 |
| -m <i>metricType</i> | 指定要显示的度量类型。将此选项与 metrics dst、metrics svc 或 metrics bkr 子命令一起使用。metricType 的值取决于为目标、服务还是代理生成度量。 |
| -msp <i>numSamples</i> | 指定 metrics bkr、metrics dst 和 metrics svc 子命令在其度量输出中显示的度量样例的数量。 |
| -n <i>argumentName</i> | 指定子命令参数的名称。根据子命令的不同，它可能是服务名称、物理目标、长期订阅、连接 ID 或事务 ID。 |
| -o <i>attribute=value</i> | 指定属性值。根据子命令参数的不同，它可能是代理属性（请参见第 100 页“使用 imqcmd 命令实用程序”）、服务属性（请参见第 107 页“管理连接服务”）或目标属性（请参见第 112 页“管理长期订阅”）。 |
| -p <i>password</i> | 指定您（管理员）的密码。不赞成使用此选项，在今后的版本中不再对其提供支持。使用下列替代方法之一： <ul style="list-style-type: none"> 在命令行中省略密码，以使命令提示您输入密码。 使用 -passfile 选项指定包含管理员密码的文件。 |

表 13-10 imqcmd 选项 (续)

| 选项 | 说明 |
|-------------------------|---|
| -passfile <i>path</i> | 指定包含用户执行命令所需密码的文件的的路径。有关详细信息，请参见第 153 页“使用密码文件”。 |
| -pst <i>pauseType</i> | 指定在暂停目标时是暂停生产方、使用方，还是二者都暂停。请参见第 112 页“管理长期订阅”。 |
| -rtm <i>timeout</i> | 指定 imqcmd 子命令的初始（重试）超时周期（以秒为单位）。超时是 imqcmd 子命令在向代理发出请求之后等待的时间长度。该子命令随后的每次重试将使用初始超时周期的倍数作为超时值。默认值：10 |
| -rtr <i>numRetries</i> | 指定 imqcmd 子命令首次超时之后尝试重试的次数。默认值：5 |
| -s | 静音模式。不显示任何输出。 |
| -secure | 使用 ssladmin 连接服务指定代理的安全管理连接（请参见第 149 页“步骤 4：配置并运行基于 SSL 的客户机”）。如果省略此选项，连接将是不安全的。 |
| -svn <i>serviceName</i> | 指定列出其连接的服务。请参见第 111 页“获得有关连接的信息”。 |
| -t <i>destType</i> | 指定目标类型：t（主题）或 q（队列）。请参见第 112 页“管理长期订阅”。 |
| -tmp | 显示临时目标。请参见第 263 页表 13-5。 |
| -u <i>userName</i> | 指定您（管理员）的名称。如果省略此值，系统会提示您输入。 |
| -v | 显示版本信息。不执行命令行上的其他选项。 此选项不需要用户名和密码。 |

另请参见

有关使用 imqcmd 及命令示例的详细信息，请参见第 5 章“管理代理”和第 6 章“管理物理目标”。

imqobjmgr

对象管理器实用程序 imqobjmgr 创建和管理 Message Queue 受管理对象。

语法

```
imqobjmgr subcommand [options]  
imqobjmgr -h|H  
imqobjmgr -v
```

子命令

对象管理器实用程序 (imqobjmgr) 包含表 13-3 中列出的子命令：

表 13-11 imqobjmgr 子命令

| 子命令 | 说明 |
|--------|-------------------|
| add | 在对象存储库中添加受管理对象。 |
| delete | 从对象存储库中删除受管理对象。 |
| list | 列出对象存储库中的受管理对象。 |
| query | 显示指定的受管理对象的相关信息。 |
| update | 修改对象存储库中的现有受管理对象。 |

命令选项

表 13-12 列出了 `imqobjmgr` 命令的选项。有关如何使用这些选项的论述，请参见基于任务的后续各节。

表 13-12 `imqobjmgr` 选项

| 选项 | 说明 |
|---------------------------------|---|
| <code>-f</code> | 执行操作，而无需用户确认。 |
| <code>-h</code> | 显示使用帮助。不执行命令行上的其他选项。 |
| <code>-H</code> | 显示使用帮助、属性列表和示例。不执行命令行上的其他选项。 |
| <code>-i fileName</code> | 指定命令文件的名称，该文件包含所有或部分子命令子句，指定了对象类型、查找名称、对象属性、对象存储库属性或其他选项。通常用于重复性信息，例如对象存储库属性。 |
| <code>-j attribute=value</code> | 指定标识和访问 JNDI 对象存储库所需的属性。请参见第 157 页“关于对象存储库”。 |
| <code>-javahome path</code> | 指定要使用的替代 Java 2 兼容运行时（默认情况下使用系统上的运行时或 Message Queue 附带的运行时）。 |
| <code>-l lookupName</code> | 指定受管理对象的 JNDI 查找名称。此名称在对象存储库的上下文中必须唯一。 |
| <code>-o attribute=value</code> | 指定受管理对象的属性。请参见第 305 页第 16 章“受管理对象的属性参考”。 |
| <code>-pre</code> | 预览模式。指出在不执行命令的情况下将执行的操作。 |
| <code>-r read-only_state</code> | 指定某个受管理对象是否为只读对象。 <code>true</code> 值表示受管理对象为只读对象。客户不能修改只读受管理对象的属性。默认情况下，只读状态设置为 <code>false</code> 。 |
| <code>-s</code> | 静音模式。不显示任何输出。 |

表 13-12 imqobjmgr 选项 (续)

| 选项 | 说明 |
|----------------------|---|
| -t <i>objectType</i> | 指定 Message Queue 受管理对象的类型: q = 队列 t = 主题 cf = 连接工厂 qf = 队列连接工厂 tf = 主题连接工厂 xcf = XA 连接工厂 (分布式事务) xqf = XA 队列连接工厂 (分布式事务) xtf = XA 主题连接工厂 (分布式事务) e = SOAP 端点 (此受管理对象类型用于支持 SOAP 消息, 如 Message Queue Developer's Guide for Java Clients 中所述。) |
| -v | 显示版本信息。不执行命令行上的其他选项。 |

另请参见

有关 imqobjmgr 及命令示例的详细信息, 请参见第 8 章 “管理受管理对象”。

imqdbmgr

数据库管理器实用程序 (imqdbmgr) 将会设置持久性所需的模式。也可以使用 imqdbmgr 命令来删除 Message Queue 已损坏的数据库表或更改数据存储库。

语法

```
imqdbmgr subcommand argument [options]
imqdbmgr -h|-help
imqdbmgr -v|-version
```

子命令

表 13-13 列出了数据库管理器实用程序 (imqdbmgr) 包含的子命令：

表 13-13 imqdbmgr 子命令

| 子命令和参数 | 说明 |
|---------------|--|
| create all | 创建一个新的数据库和 Message Queue 持久性存储模式。此命令用于嵌入式数据库系统，使用时需指定 <code>imq.persist.jdbc.createdburl</code> 属性。 |
| create tbl | 在现有数据库系统中创建 Message Queue 持久性存储模式。此命令用于外部数据库系统。 |
| delete tbl | 删除当前持久性存储数据库中的现有 Message Queue 数据库表。 |
| delete oldtbl | 删除早期版本的持久性存储数据库中的所有 Message Queue 数据库表。将持久性存储自动迁移至 Message Queue 当前版本后使用。 |
| recreate tbl | 删除当前持久性存储数据库中的现有 Message Queue 数据库表，然后重新创建 Message Queue 持久性存储模式。 |
| reset lck | 对锁进行重置，以便其他进程可以使用该持久性存储数据库。 |

命令选项

表 13-14 列出了 imqdbmgr 命令的选项。

表 13-14 imqdbmgr 选项

| 选项 | 说明 |
|------------------|--|
| -Dproperty=value | 将指定属性设置为指定值。 |
| -b instanceName | 指定代理实例名称并使用对应的实例配置文件。 |
| -h | 显示使用帮助。不执行命令行上的其他选项。 |
| -p password | 指定数据库密码。不赞成使用此选项，在今后的版本中不再对其提供支持。使用下列替代方法之一： <ul style="list-style-type: none"> 在命令行中省略密码，以使命令提示您输入密码。 使用 <code>-passfile</code> 选项指定包含数据库密码的文件。 |
| -passfile path | 指定包含数据库密码的文件的完整路径。有关详细信息，请参见第 153 页“使用密码文件”。 |

表 13-14 imqdbmgr 选项

| 选项 | 说明 |
|----------------|----------------------|
| -u <i>name</i> | 指定数据库用户名。 |
| -v | 显示版本信息。不执行命令行上的其他选项。 |

另请参见

有关设置持久性存储的详细信息，请参见第 93 页“设置持久性存储”。

imqusermgr

此用户管理器实用程序 (imqusermgr) 允许您编辑或填充平面文件用户系统信息库。使用 imqusermgr 之前，请谨记以下内容：

- 如果代理特定的用户系统信息库不存在，您必须启动相应的代理实例来创建此系统信息库。
- imqusermgr 命令需要运行在已经安装了代理的主机上。
- 您需要具有写入系统信息库的相应权限：即在 Solaris 和 Linux 上，您的身份必须是超级用户或首次创建代理实例的用户。

语法

```
imqusermgr subcommand [options]
imqusermgr -h
imqusermgr -v
```

子命令

表 13-15 列出了 imqusermgr 子命令，本章中描述了其用法。

表 13-15 imqusermgr 子命令

| 子命令 | 说明 |
|--|--|
| add [-i <i>instanceName</i>] -u <i>userName</i> -p <i>passwd</i> [-g <i>group</i>] [-s] | 将用户和关联的密码添加到指定的（或默认的）代理实例系统信息库中，并有选择地指定用户所属的组。 |
| delete [-i <i>instanceName</i>] -u <i>userName</i> [-s] [-f] | 从指定的（或默认的）代理实例系统信息库中删除指定用户。 |
| list [-i <i>instanceName</i>] [-u <i>userName</i>] | 显示指定的（或默认的）代理实例系统信息库中有关指定用户或所有用户的信息。 |
| update [-i <i>instanceName</i>] -u <i>userName</i> -p <i>passwd</i> [-a <i>state</i>] [-s] [-f] | 更新指定的（或默认的）代理实例系统信息库中指定用户的密码和 / 或状态。 |
| update [-i <i>instanceName</i>] -u <i>userName</i> -a <i>state</i> [-p <i>passwd</i>] [-s] [-f] | |

命令选项

表 13-16 列出了 imqusermgr 命令的选项。

表 13-16 imqusermgr 选项

| 选项 | 说明 |
|------------------------|--|
| -a <i>active_state</i> | 指定用户是否处于活动状态 (true/false)。true 表示处于活动状态。这是默认值。 |
| -f | 执行操作，无需用户确认。 |
| -h | 显示使用帮助。不执行命令行上的其他选项。 |
| -i <i>instanceName</i> | 指定此命令要应用到的代理实例用户系统信息库。如果未指定，则假定为默认的实例名称 imqbroker。 |
| -p <i>passwd</i> | 指定用户密码。 |
| -g <i>group</i> | 指定用户组。有效值包括 admin、user 和 anonymous。 |
| -s | 设置静音模式。 |
| -u <i>userName</i> | 指定用户名。 |
| -v | 显示版本信息。不执行命令行上的其他选项。 |

另请参见

有关设置和管理平面文件用户系统信息库及 `imqusermgr` 命令示例的详细信息，请参见第 130 页“使用平面文件用户系统信息库”。

imqsvcadmin

服务管理 (`imqsvcadmin`) 实用程序将代理安装为 Windows 服务。

语法

```
imqsvcadmin subcommand [options]
```

```
imqsvcadmin -h
```

子命令

Message Queue 服务管理器实用程序 (`imqsvcadmin`) 包含表 13-17 中列出的子命令。

表 13-17 imqsvcadmin 子命令

| 子命令 | 说明 |
|----------------------|---|
| <code>install</code> | 安装服务并特定启动选项。 |
| <code>query</code> | 显示 <code>imqsvcadmin</code> 命令的启动选项。启动选项包括服务的启动方式（手动或自动）、服务的位置、Java 运行时的位置以及启动时传递给代理的参数值。 |
| <code>remove</code> | 删除服务。 |

命令选项

表 13-18 列出了 `imqsvcadmin` 命令的选项。

表 13-18 imqsvcadmin 选项

| 选项 | 说明 |
|-----------------|----------------------|
| <code>-h</code> | 显示使用帮助。不执行命令行上的其他选项。 |

表 13-18 imqsvcadmin 选项 (续)

| 选项 | 说明 |
|--|--|
| -javahome <i>path</i> | 指定要使用的替代 Java 2 兼容运行时 (默认使用系统上的运行时或 Message Queue 附带的运行时) 的路径。 示例: <code>imqsvcadmin -install -javahome d:\jdk1.4</code> |
| -jrehome <i>path</i> | 指定替代 Java 2 兼容 JRE 的路径。 示例: <code>imqsvcadmin -install -jrehome d:\jre\1.4</code> |
| -vmargs <i>arg</i> [[<i>arg</i>]...] | 指定传递给正在运行代理服务的 Java VM 的其他参数。(也可以在 Windows 服务控制面板的“启动参数”字段中指定这些参数。) 示例: <code>-vmargs "-Xms16m -Xmx128m"</code> |
| -args <i>arg</i> [[<i>arg</i>]?] | 指定传递给代理服务的其他命令行参数。有关 <code>imqbrokerd</code> 选项的说明, 请参见第 255 页“ <code>imqbrokerd</code> ”。 (也可以在 Windows 服务控制面板的“启动参数”字段中指定这些参数。) 例如, <code>imqsvcadmin -install -args "-passfile d:\imqpassfile"</code> |

使用 `-javahome`、`-vmargs` 和 `-args` 选项指定的信息存储在 Window 注册表项 `JREHome`、`JVMArgs` 和 `ServiceArgs` 下, 其路径位于:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
\Services\iMQ_Broker\Parameters
```

另请参见

有关作为 Windows 服务运行 Message Queue 的详细信息, 请参见第 66 页“在 Windows 中自动启动”。

imqkeytool

(`imqusermgr` 命令为代理生成自签名证书。可以对 `ssljms`、`ssladmin` 或 `cluster` 连接服务使用相同的证书。在 UNIX 系统中，您可能需要以超级用户 (`root`) 帐户来运行 `imqkeytool`。

语法

```
imqkeytool -broker
```

另请参见

有关设置安全连接的详细信息，请参见第 144 页 “使用基于 SSL 的服务”。

imqkeytool

代理属性参考

本章列出并说明了代理配置属性。第一节是按字母顺序列出的所有代理属性的列表以及包含完整说明的小节参考。所有其他小节均按功能对代理属性进行分组，并提供这些属性的完整说明。

本章包含以下小节：

- 第 279 页 “按字母顺序排列的属性列表”
- 第 283 页 “连接服务属性”
- 第 285 页 “消息路由器属性”
- 第 287 页 “持久性管理器属性”
- 第 291 页 “安全性管理器属性”
- 第 295 页 “监视和日志记录属性”
- 第 298 页 “群集配置属性”

在说明表格中，将可以使用 `imqcmd update bkr` 命令进行设置的属性做了标记。

按字母顺序排列的属性列表

表 14-1 是一个按字母顺序排列的代理实例属性列表。可以使用它确定任何属性的类别，然后使用类别说明来查找位于本章其他地方的完整属性说明。

在本表格中，左侧列按字母顺序列出了各个属性。右侧列显示了该属性所属的类别，并提供了到相应小节的交叉参考。

表 14-1 代理实例配置属性

| 属性名称 | 参考 |
|--|---------------------|
| <code>imq.accesscontrol.enabled</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.accesscontrol.file.filename</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.audit.enabled</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.authentication.basic.user_repository</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.authentication.client.response.timeout</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.authentication.type</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.autocreate.destination.isLocalOnly</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.destination.limitBehavior</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.destination.maxBytesPerMsg</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.destination.maxNumMsgs</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.destination.maxNumProducers</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.destination.maxTotalMsgBytes</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.destination.useDMQ</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.queue</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.queue.consumerFlowLimit</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.queue.localDeliveryPreferred</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.queue.maxNumActiveConsumers</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.queue.maxNumBackupConsumers</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.topic</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.autocreate.topic.consumerFlowLimit</code> | 第 286 页 “自动创建配置属性” |
| <code>imq.cluster.<i>property_name</i></code> | 第 298 页 “群集配置属性” |
| <code>imq.destination.DMQ.truncateBody</code> | 第 285 页 “消息路由器属性” |
| <code>imq.destination.logDeadMsgs</code> | 第 295 页 “监视和日志记录属性” |
| <code>imq.hostname</code> | 第 283 页 “连接服务属性” |
| <code>imq.httpjms.http.<i>property_name</i></code> | 第 355 页表 C-3 |
| <code>imq.httpsjms.https.<i>property_name</i></code> | 第 355 页表 C-3 |
| <code>imq.imqcmd.password</code> | 第 291 页 “安全性管理器属性” |

表 14-1 代理实例配置属性 (续)

| 属性名称 | 参考 |
|--|-----------------------|
| <code>imq.keystore.property_name</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.log.console.output</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.console.stream</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.file.dirpath</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.file.filename</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.file.output</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.file.rolloverbytes</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.file.rolloversecs</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.level</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.syslog.facility</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.syslog.identity</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.syslog.logconsole</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.syslog.logpid</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.syslog.output</code> | 第 295 页 “监视服务属性” |
| <code>imq.log.timezone</code> | 第 295 页 “监视服务属性” |
| <code>imq.message.expiration.interval</code> | 第 285 页 “消息路由器属性” |
| <code>imq.message.max_size</code> | 第 285 页 “消息路由器属性” |
| <code>imq.metrics.enabled</code> | 第 285 页 “消息路由器属性” |
| <code>imq.metrics.interval</code> | 第 285 页 “消息路由器属性” |
| <code>imq.metrics.topic.enabled</code> | 第 285 页 “消息路由器属性” |
| <code>imq.metrics.topic.interval</code> | 第 285 页 “消息路由器属性” |
| <code>imq.metrics.topic.persist</code> | 第 285 页 “消息路由器属性” |
| <code>imq.metrics.topic.timetolive</code> | 第 295 页 “监视服务属性” |
| <code>imq.passfile.dirpath</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.passfile.enabled</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.passfile.name</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.persist.file.destination.message.filepool.limit</code> | 第 288 页 “基于文件的持久性的属性” |

表 14-1 代理实例配置属性 (续)

| 属性名称 | 参考 |
|---|-----------------------|
| <code>imq.persist.file.message.cleanup</code> | 第 285 页 “消息路由器属性” |
| <code>imq.persist.file.message.filepool.cleanratio</code> | 第 285 页 “消息路由器属性” |
| <code>imq.persist.file.message.max_record_size</code> | 第 285 页 “消息路由器属性” |
| <code>imq.persist.file.sync.enabled</code> | 第 288 页 “基于文件的持久性的属性” |
| <code>imq.persist.jdbc.property_name</code> | 第 287 页 “持久性管理器属性” |
| <code>imq.persist.store</code> | 第 285 页 “消息路由器属性” |
| <code>imq.ping.interval</code> | 第 283 页 “连接服务属性” |
| <code>imq.portmapper.backlog</code> | 第 283 页 “连接服务属性” |
| <code>imq.portmapper.hostname</code> | 第 283 页 “连接服务属性” |
| <code>imq.portmapper.port</code> | 第 283 页 “连接服务属性” |
| <code>imq.resource_state.count</code> | 第 285 页 “消息路由器属性” |
| <code>imq.resource_state.threshold</code> | 第 285 页 “消息路由器属性” |
| <code>imq.service.activelist</code> | 第 283 页 “连接服务属性” |
| <code>imq.service_name.accesscontrol.enabled</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.service_name.accesscontrol.file.filename</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.service_name.authentication.type</code> | 第 291 页 “安全性管理器属性” |
| <code>imq.service_name.max_threads</code> | 第 283 页 “连接服务属性” |
| <code>imq.service_name.min_threads</code> | 第 283 页 “连接服务属性” |
| <code>imq.service_name.protocol_type.hostname</code> | 第 283 页 “连接服务属性” |
| <code>imq.service_name.protocol_type.port</code> | 第 283 页 “连接服务属性” |
| <code>imq.service_name.threadpool_model</code> | 第 283 页 “连接服务属性” |
| <code>imq.shared.connectionMonitor_limit</code> | 第 283 页 “连接服务属性” |
| <code>imq.system.max_count</code> | 第 285 页 “消息路由器属性” |
| <code>imq.system.max_size</code> | 第 285 页 “消息路由器属性” |
| <code>imq.transaction.autorollback</code> | 第 285 页 “消息路由器属性” |
| <code>imq.user_repository.ldap.property_name</code> | 第 291 页 “安全性管理器属性” |

连接服务属性

表 14-2 列出了连接服务属性。第一列列出了属性名称。对于每个属性名称，第二列对该属性进行了说明；第三列指定了其类型；而第四列则给出了其默认值。

表 14-2 连接服务属性

| 属性名称 | 说明 | 类型 | 默认值 |
|--|---|-----|---------------------------------------|
| <code>imq.service.activelist</code> | 按名称列出的启动代理时将变为活动状态的连接服务列表，服务之间以逗号分隔。支持的服务包括： <code>jms</code> 、 <code>ssljms</code> 、 <code>httpjms</code> 、 <code>httpsjms</code> 、 <code>admin</code> 和 <code>ssladmin</code> 。 | 列表 | <code>jms</code> 、 <code>admin</code> |
| <code>imq.ping.interval</code> | 代理通过连接连续尝试 ping Message Queue 客户机运行时的时间周期（以秒为单位）。 | 整数 | 120 |
| <code>imq.hostname</code> | 当有多个主机可用（例如，一台计算机中安装了多个网络接口卡）时所有连接服务将绑定到的主机（主机名或 IP 地址）。 | 字符串 | 全部可用 IP 地址 |
| <code>imq.portmapper.port¹</code> | 代理的主端口，即端口映射器所在的端口。如果主机上运行多个代理实例，则必须为每个实例分配唯一的端口映射器端口。 | 整数 | 7676 |
| <code>imq.portmapper.hostname</code> | 当有多个主机可用（例如，一台计算机中安装了多个网络接口卡）时端口映射器将绑定到的主机（主机名或 IP 地址）。 | 字符串 | 从 <code>imq.hostname</code> 继承 |
| <code>imq.portmapper.backlog</code> | 端口映射器在拒绝请求之前可处理的并发请求的最大数量。该属性设置了可存储在操作系统待办事项中等待端口映射器处理的请求的数量。 | 整数 | 50 |
| <code>imq.service_name.protocol_type².port</code> | 仅适用于 <code>jms</code> 、 <code>ssljms</code> 、 <code>admin</code> 和 <code>ssladmin</code> 服务，命名的连接服务的端口号。 要配置 <code>httpjms</code> 和 <code>httpsjms</code> 连接服务，请参见附录 C “HTTP/HTTPS 支持” | 整数 | 0（零） 端口是由端口映射器动态分配的。 |
| <code>imq.service_name.protocol_type².hostname</code> | 仅适用于 <code>jms</code> 、 <code>ssljms</code> 、 <code>admin</code> 和 <code>ssladmin</code> 服务，当有多个主机可用（例如，一台计算机中安装了多个网络接口卡）时命名的连接服务将绑定到的主机（主机名或 IP 地址）。 | 字符串 | 从 <code>imq.hostname</code> 继承 |

表 14-2 连接服务属性 (续)

| 属性名称 | 说明 | 类型 | 默认值 |
|--|--|-----|---|
| imq.service_name. min_threads | 线程数量, 达到该数量后, 线程池中将维护此数量的线程供命名的连接服务使用。 默认值随连接服务的不同而有所变化。 | 整数 | 10 (jms) 10 (ssljms) 10 (httpjms) 10 (httpsjms) 4 (admin) 4 (ssladmin) |
| imq.service_name. max_threads | 线程数量, 达到该数量后, 新的线程将不会被添加到线程池中供命名的连接服务使用。该数量必须大于零, 并且必须大于 min_threads 的值。 默认值随连接服务的不同而有所变化。 | 整数 | 1000 (jms) 500 (ssljms) 500 (httpjms) 500 (httpsjms) 10 (admin) 10 (ssladmin) |
| imq.service_name. threadpool_model | 一个字符串, 指定线程是专用于特定的连接 (dedicated), 还是根据命名的连接服务的需要由多个连接共享 (shared)。共享模型 (线程池管理) 将增加代理支持的连接数量, 但只能用于 jms 和 admin 连接服务。 默认值随连接服务的不同而有所变化。 | 字符串 | dedicated (jms) dedicated (ssljms) dedicated (httpjms) dedicated (httpsjms) dedicated (admin) dedicated (ssladmin) |
| imq.shared. connectionMonitor_limit | 仅适用于共享线程池模型, 分配器线程可监视的连接的最大数量。(系统会分配足够多的分配器线程来监视所有连接。) 此值越小, 系统向线程分配活动连接的速度就越快。值 -1 表示没有限制。 默认值随操作系统的不同而有所变化。 | 整数 | 512 (Solaris 和 Linux) 64 (Windows) |

1. 此属性可用于 imqcmd update bkr 命令。
2. protocol_type 为表 4-2 中指定的协议类型。

消息路由器属性

表 14-3 列出了消息路由器属性。第一列列出了属性名称。对于每个属性名称，第二列对该属性进行了说明；第三列指定了其类型；而第四列则给出了其默认值。

第 286 页表 14-4 中列出了用来配置消息服务器自动创建目标功能的自动创建属性。

表 14-3 消息路由器属性

| 属性名称 | 说明 | 类型 | 默认值 |
|---|---|--------------------|--|
| imq.destination. DMQ.truncateBody ¹ | 一个布尔值，指定在将消息存储到停用消息队列中之前代理是否删除消息主体。当值为 true 时，代理只保存消息标题和属性数据。当值为 false 时，代理保存标题和主体。 | 布尔值 | false |
| imq.message. expiration.interval | 发生过期消息回收的时间间隔（以秒为单位）。 | 整数 | 60 |
| imq.system.max_count ¹ | 代理保留的消息的最大数量。附加消息将被拒绝。值 -1 表示没有限制。 | 整数 | -1 |
| imq.system.max_size ¹ | 代理保留的消息总大小的最大值（以字节、千字节或兆字节为单位）。附加消息将被拒绝。值 -1 表示没有限制。 | 字节字符串 ² | -1 |
| imq.message.max_size ¹ | 允许的消息主体大小的最大值（以字节、千字节或兆字节为单位）。任何大于此值的消息将被拒绝。值 -1 表示没有限制。 | 字节字符串 ² | 70m |
| imq.resource_state. threshold | 将触发各个内存资源状态的内存占用百分比。资源状态的值可以是 green、yellow、orange 和 red。 | 整数 (百分比) | 0 (green) 80 (yellow) 90 (orange) 98 (red) |
| imq.resource_state.count | 在检查系统内存查看是否达到新的内存阈值之前，允许同时处理的一批外来消息的最大数量。当系统内存愈加不足时，此限制可限制消息的生产方。 | 整数 (百分比) | 5000 (green) 500 (yellow) 50 (orange) 0 (red) |
| imq.transaction. autorollback | 一个布尔值，指定当启动代理时，是否自动回滚处于 PREPARED 状态的分布式事务。如果为 false，则必须使用 imqcmd 手动提交或回滚事务（请参见第 113 页“管理事务”）。 | 布尔值 | false |

1. 此属性可用于 imqcmd update bkr 命令。

2. 能够以字节、千字节和兆字节的形式来表示作为**字节字符串**键入的值：例如：1000 表示 1000 字节；7500b 表示 7500 字节；77k 表示 77 千字节（77 x 1024 = 78848 字节）；17m 表示 17 兆字节（17 x 1024 x 1024 = 17825792 字节）

表 14-4 列出了自动创建目标时代理使用的属性。

表 14-4 自动创建配置属性

| 属性名称 | 说明 | 类型 | 默认值 |
|--|--|--------------------|------------------|
| <code>imq.autocreate.destination.isLocalOnly</code> | (仅适用于代理群集。) 一个布尔值, 指定目标不能在其他代理上复制, 因而将消息传送限制为本地使用方 (连接到创建目标的代理的使用方)。在创建目标之后此属性无法更新。 | 布尔值 | false |
| <code>imq.autocreate.destination.limitBehavior</code> | <p>一个字符串, 指定当达到内存限制的阈值时代理响应的方式。允许的值为:</p> <ul style="list-style-type: none"> FLOW_CONTROL——减慢生产方。 REMOVE_OLDEST——丢弃最旧的消息。 REMOVE_LOW_PRIORITY——根据消息的生存期丢弃优先级最低的消息。 REJECT_NEWEST——拒绝最新的消息。生产方客户机仅收到拒绝持久性消息的异常。要将此限制行为用于非持久性消息, 请设置 <code>imqAckOnProduce</code> 连接工厂属性。 <p>如果将此属性设置为 REMOVE_OLDEST 或 REMOVE_LOW_PRIORITY, 并将 <code>imq.autocreate.destination.useDMQ</code> 设置为 true, 代理将把超出限制的消息移动到停用消息队列中。</p> | 字符串 | REJECT NEWEST |
| <code>imq.autocreate.destination.maxBytesPerMsg</code> | 自动创建的目标中允许的任何单个消息大小的最大值 (以字节为单位)。值 -1 表示消息大小是无限的。 | 字节字符串 ² | 10k |
| <code>imq.autocreate.destination.maxNumMsgs</code> | 自动创建的目标中允许的未使用消息的最大数量。值 -1 表示数量是无限的。 | 整数 | 100,000 |
| <code>imq.autocreate.destination.maxNumProducers</code> | 目标允许的生产方的最大数量。当达到此限制时, 将无法创建新的生产方。值 -1 表示生产方数量是无限的。 | 整数 | 100 |
| <code>imq.autocreate.destination.maxTotalMsgBytes</code> | 目标中允许未使用消息占用的内存的最大总量 (以字节为单位)。值 -1 表示内存是无限的。 | 字节字符串 ² | 10m |
| <code>imq.autocreate.destination.useDMQ</code> | 一个布尔值, 指定代理是否将自动创建目标的停用消息移动到停用消息队列中。 | 布尔值 | true |
| <code>imq.autocreate.queue¹</code> | 一个布尔值, 指定是否允许代理自动创建队列目标。 | 布尔值 | true |

表 14-4 自动创建配置属性（续）

| 属性名称 | 说明 | 类型 | 默认值 |
|---|--|-----|-------|
| imq.autocreate.queue.consumerFlowLimit | 将要在一批中传送给使用方的消息的最大数量。在负荷平衡队列传送中，为负荷平衡开始之前路由至活动使用方的队列消息的初始数量。可以使用为相应连接中的目标使用方设置的较小值来覆盖此限制。值为 -1 表示不限制数量。 | 整数 | 1000 |
| imq.autocreate.queue.localDeliveryPreferred | （仅应用于代理群集中的负荷平衡队列传送。）一个布尔值，指定仅当在本地代理中没有使用方时才将消息传送到远程的使用方。要求未将自动创建的目标限制为仅本地传送 (<code>isLocalOnly = false</code>)。 | 布尔值 | false |
| imq.autocreate.queue.maxNumActiveConsumers | 在从自动创建的队列目标的负荷平衡传送中可以处于活动状态的最大使用方数。值为 -1 表示不限制数量。 | 整数 | 1 |
| imq.autocreate.queue.maxNumBackupConsumers | 在从自动创建的队列目标的负荷平衡传送期间出现任何错误时，可以代替这些活动使用方的最大备份使用方的数量。值为 -1 表示不限制数量。 | 整数 | 0（零） |
| imq.autocreate.topic | 一个布尔值，指定是否允许代理自动创建主题目标。 | 布尔值 | true |
| imq.autocreate.topic.consumerFlowLimit | 将要在一批中传送给使用方的消息的最大数量。值为 -1 表示不限制数量。 | 整数 | 1000 |

1. 此属性可用于 `imqcmd update bkr`。

2. 能够以字节、千字节和兆字节的形式来表示作为**字节字符串**键入的值：例如：1000 表示 1000 字节；7500b 表示 7500 字节；77k 表示 77 千字节（ $77 \times 1024 = 78848$ 字节）；17m 表示 17 兆字节（ $17 \times 1024 \times 1024 = 17825792$ 字节）

持久性管理器属性

要配置代理的持久性功能，您必须指定 `imq.persist.store` 的值或接受默认值。

表 14-5 必需的持久性管理器属性

| 属性名称 | 说明 | 类型 | 默认值 |
|-------------------|---|-----|------|
| imq.persist.store | 一个字符串，指定代理是使用内置的、基于文件的持久性，还是使用插入的、JDBC 兼容的持久性。 该值必须是 <code>file</code> 或 <code>jdbc</code> 。 | 字符串 | file |

在下面的小节中，介绍了支持基于文件的持久性和基于 JDBC 的持久性的属性。

基于文件的持久性

表 14-6 列出了支持基于文件的持久性的属性。第一列列出了属性名称。对于每个属性名称，第二列对该属性进行了说明；第三列指定了其类型；而第四列则给出了其默认值。

表 14-6 基于文件的持久性的属性

| 属性名称 | 说明 | 类型 | 默认值 |
|---|---|--------------------|-------|
| imq.persist.file.sync.enabled | <p>一个布尔值，指定持久性操作是否使内存中状态与物理存储设备同步。如果将此属性设置为 true，则可以避免由于系统崩溃而导致的数据丢失，但持久性操作的性能会有所降低。</p> <p>如果您正在对 Message Queue 运行 Sun Cluster 和 Sun Cluster Data Service，则请将所有群集节点上的代理的该属性设置为 true。</p> | 布尔值 | false |
| imq.persist.file.message.max_record_size | 适用于内置的、基于文件的持久性，将添加到消息存储文件中的消息（而不是存储在单独文件中的消息）大小的最大值。 | 字节字符串 ¹ | 1m |
| imq.persist.file.destination.message.filepool.limit | 适用于内置的、基于文件的持久性，目标文件池中可供重复使用的空闲文件的最大数量。该数量越大，代理处理持久性数据的速度越快。超过该值的空闲文件将被删除。代理将根据需要创建文件或删除超过此限制的文件。 | 整数 | 100 |
| imq.persist.file.message.filepool.cleanratio | 适用于内置的、基于文件的持久性，保持清除状态（清空）的目标文件池中空闲文件的百分比。该值越大，在操作中清除文件所需的开销越大，但文件池所需的磁盘空间越小。 | 整数 | 0（零） |
| imq.persist.file.message.cleanup | 适用于内置的、基于文件的持久性，一个指定代理在关闭时是否清除目标文件池中的空闲文件的布尔值。如果值为 false，则关闭代理的速度会加快，但文件存储需要的磁盘空间会更多。 | 布尔值 | false |

1. 能够以字节、千字节和兆字节的形式来表示作为字节字符串键入的值。示例：1000 表示 1000 字节；7500b 表示 7500 字节；77k 表示 77 千字节（77 x 1024 = 78848 字节）；17m 表示 17 兆字节（17 x 1024 x 1024 = 17825792 字节）

基于 JDBC 的持久性

表 14-7 包含支持基于 JDBC 的持久性的属性。该表列出了各个属性，对其进行了说明，然后给出如何进行配置以用于 PointBase 产品的示例。

表 14-7 基于 JDBC 的持久性的属性

| 属性名称 | 说明 | 示例 |
|---|---|---|
| <code>imq.persist.store</code> | 一个字符串，指定基于文件或基于 JDBC 的数据存储库。 | <code>jdbc</code> |
| <code>imq.persist.jdbc.brokerid</code> | <p>(可选) 附加到数据库表名称中的代理实例标识符。当多个代理实例使用同一个数据库作为持久性数据存储库时，标识符可使代理实例保持唯一。</p> <p>嵌入式数据库只存储一个代理实例的数据，因此通常不需要此属性。</p> <p>标识符必须是字母数字字符串，其长度不能超过数据库所允许的最大表名长度减去 12 后的长度。</p> | (PointBase 嵌入式版本不需要) |
| <code>imq.persist.jdbc.driver</code> | 连接到数据库的 JDBC 驱动程序的 Java 类名。 | <code>com.pointbase.jdbc.jdbcUniversalDriver</code> |
| <code>imq.persist.jdbc.opendburl</code> | 打开现有数据库连接的数据库 URL。 | <code>jdbc:pointbase:embedded:dbName;database.home=../instances/instanceName/dbstore</code> |
| <code>imq.persist.jdbc.createdburl</code> | <p>(可选) 打开创建数据库连接的数据库 URL。</p> <p>仅当使用 <code>imqdbmgr</code> 创建数据库时才指定此属性。</p> | <code>jdbc:pointbase:embedded:dbName;new,database.home=../instances/instanceName/dbstore</code> |
| <code>imq.persist.jdbc.closedburl</code> | (可选) 当代理关闭时关闭当前数据库连接的数据库 URL。 | PointBase 不需要 |
| <code>imq.persist.jdbc.user</code> | <p>(可选) 用于打开数据库连接的用户名 (如果需要)。出于安全性考虑，可以指定值，而不使用命令行选项 <code>imqbrokerd -dbuser</code> 和 <code>imqdbmgr -u</code></p> | |

表 14-7 基于 JDBC 的持久性的属性 (续)

| 属性名称 | 说明 | 示例 |
|-----------------------------------|---|--|
| imq.persist.jdbc.needpassword | <p>(可选) 一个布尔值, 指定数据库是否要求输入代理访问的密码。值 true 表示密码是必需的。</p> <p>如果设置此选项, imqbrokerd 和 imqdbmgr 命令将提示输入密码, 除非使用 -passfile 选项指定包含该密码的文件。</p> | |
| imq.persist.jdbc.password | <p>(可选) 用于打开数据库连接的密码 (如果需要)。</p> <p>仅在密码文件中指定此属性。</p> | |
| imq.persist.jdbc.table.IMQSV35 | 用于创建版本表的 SQL 命令。 | <pre>CREATE TABLE \${name} (STOREVERSION INTEGER NOT NULL, BROKERID VARCHAR(100))</pre> |
| imq.persist.jdbc.table.IMQCCREC35 | 用于创建配置更改记录表的 SQL 命令。 | <pre>CREATE TABLE \${name} (RECORDTIME BIGINT NOT NULL, RECORD BLOB(10k))</pre> |
| imq.persist.jdbc.table.IMQDEST35 | 用于创建目标表的 SQL 命令。 | <pre>CREATE TABLE \${name} (DID VARCHAR(100) NOT NULL, DEST BLOB(10k), primary key(DID))</pre> |
| imq.persist.jdbc.table.IMQINT35 | 用于创建 Interest 表的 SQL 命令。 | <pre>CREATE TABLE \${name} (CUID BIGINT NOT NULL, INTEREST BLOB(10k), primary key(CUID))</pre> |
| imq.persist.jdbc.table.IMQMSG35 | <p>用于创建消息表的 SQL 命令。</p> <p>MSG 列的默认最大长度为 1 Megabyte (1m)。如果希望消息超过该长度, 请对长度进行相应的设置。如果已经创建了这些表, 则需要重新创建它们以进行更改。</p> | <pre>CREATE TABLE \${name} (MID VARCHAR(100) NOT NULL, DID VARCHAR(100), MSGSIZE BIGINT, MSG BLOB(1m), primary key(MID))</pre> |

表 14-7 基于 JDBC 的持久性的属性（续）

| 属性名称 | 说明 | 示例 |
|---------------------------------------|----------------------------|---|
| imq.persist.jdbc.table. IMQPROPS35 | 用于创建属性表的 SQL 命令。 | CREATE TABLE \${name} (PROPNAME VARCHAR(100) NOT NULL, PROPVALUE BLOB(10k), primary key(PROPNAME)) |
| imq.persist.jdbc.table. IMQILIST35 | 用于创建 Interest 状态表的 SQL 命令。 | CREATE TABLE \${name} (MID VARCHAR(100) NOT NULL, CUID BIGINT, DID VARCHAR(100), STATE INTEGER, primary key(MID, CUID)) |
| imq.persist.jdbc.table.IMQTXN35 | 用于创建事务表的 SQL 命令。 | CREATE TABLE \${name} (TUID BIGINT NOT NULL, STATE INTEGER, TSTATEOBJ BLOB(10K), primary key(TUID)) |
| imq.persist.jdbc.table. IMQTACK35 | 用于创建事务确认表的 SQL 命令。 | CREATE TABLE \${name} (TUID BIGINT NOT NULL, TXNACK BLOB(10k)) |

安全性管理器属性

表 14-8 列出了安全性管理器属性。第一列列出了属性名称。对于每个属性名称，第二列对该属性进行了说明；第三列指定了其类型；而第四列则给出了其默认值。

如果您正在使用 SSL，请参考在下面的表 14-9 中列出的密钥库配置属性。

表 14-8 安全性管理器属性

| 属性名称 | 说明 | 类型 | 默认值 |
|-------------------------------|---|-----|------|
| imq.accesscontrol. enabled | 一个布尔值，指定是否为代理支持的所有连接服务设置访问控制。表明系统是否要检查已验证的用户是否如访问控制属性文件所指定，拥有使用连接服务或执行与特定目标有关的特定 Message Queue 操作的权限。 | 布尔值 | true |

表 14-8 安全性管理器属性（续）

| 属性名称 | 说明 | 类型 | 默认值 |
|--|---|-----|--|
| imq.accesscontrol.file.filename | 代理实例支持的所有连接服务的访问控制属性文件的名称。文件名称指定访问控制目录的相对文件路径（请参见附录 A “ Message Queue 数据在特定操作系统中的位置 ”）。 | 字符串 | accesscontrol.properties |
| imq.audit.enabled | 一个布尔值，指定是否启动代理日志文件的审计日志（仅限企业版）。 | 布尔值 | false |
| imq.authentication.basic.user_repository | 对于 Base 64 编码，一个指定用于验证的用户系统信息库类型的字符串：基于文件 (file) 或 LDAP (ldap)。 | 字符串 | file |
| imq.authentication.client.response.timeout | 系统等待客户机响应来自代理的验证请求的时间间隔（以秒为单位）。 | 整数 | 180 |
| imq.authentication.type | 一个字符串，指定传送密码时应使用 Base 64 编码 (basic) 还是 MD5 摘要 (digest)。设置代理支持的所有连接服务的编码。 | 字符串 | digest |
| imq.imqcmd.password | 管理用户的密码。在用户使用某个命令执行操作前，imqcmd 命令实用程序使用此密码对其进行验证。 | 字符串 | 无 |
| imq.keystore.property_name | 一个字符串，指定与 SSL 密钥库相关的安全性属性（适用于基于 SSL 的服务）。请参见第 295 页表 14-9。 | 字符串 | 无 |
| imq.passfile.dirpath | 包含密码文件的目录的路径（取决于操作系统）。 | 字符串 | 请参见附录 A |
| imq.passfile.enabled | 一个布尔值，指定是否在密码文件中指定用于安全性通信的用户密码（适用于 SSL 、 LDAP 和 JDBC™ ） | 布尔值 | false |
| imq.passfile.name | 密码文件的名称。 | 字符串 | 密码文件 |
| imq.service_name.accesscontrol.enabled | 一个布尔值，指定是否为命名连接服务设置访问控制以覆盖代理范围的设置。表明系统是否要检查已验证的用户是否如访问控制属性文件所指定，拥有使用连接服务或执行与特定目标有关的特定 Message Queue 操作的权限。 | 布尔值 | 从系统范围的属性 imq.accesscontrol.enabled 继承。 |

表 14-8 安全性管理器属性（续）

| 属性名称 | 说明 | 类型 | 默认值 |
|--|---|-----|-------|
| imq.service_name.accesscontrol.file.filename | 代理实例的命名连接服务的访问控制属性文件的名称。文件名称指定访问控制目录的相对文件路径（请参见附录 A “Message Queue 数据在特定操作系统中的位置”）。 默认值是从系统范围的属性 imq.accesscontrol.file.filename 继承的。 | 字符串 | 请参见说明 |
| imq.service_name.authentication.type | 一个字符串，指定传送密码时应使用 Base 64 编码 (basic) 还是 MD5 摘要 (digest)。设置命名的连接服务的编码，此设置将覆盖任意代理范围的设置。 默认值是从系统范围的属性 imq.authentication.type 继承的。 | 字符串 | 请参见说明 |
| imq.user_repository.ldap.base | 用户条目的目录库。 | 字符串 | 无 |
| imq.user_repository.ldap.gidattr | 提供商特定的属性标识符，其值为组名。 | 字符串 | 无 |
| imq.user_repository.ldap.grpbase | 组条目的目录库。 | 字符串 | 无 |
| imq.user_repository.ldap.grpfilter | JNDI 搜索过滤器（以逻辑表达式的形式表示搜索查询）。通过为组指定搜索过滤器，代理可以缩小搜索范围，从而使搜索更有效。有关详细信息，请参见位于以下位置的 JNDI 教程： http://java.sun.com/products/jndi/tutorial 此属性并不是必须设置的。 | 字符串 | 无 |
| imq.user_repository.ldap.grpsearch | 一个布尔值，指定是否启用组搜索。请参见 LDAP 提供商提供的文档，决定您是否可以将用户与组相关联。 请注意，Message Queue 不支持嵌套组。 | 布尔值 | false |
| imq.user_repository.ldap.memattr | 组条目中的属性标识符，其值是组成员的独特名称。 | 字符串 | 无 |
| imq.user_repository.ldap.password | 与代理使用的独特名称关联的密码。 仅在密码文件中指定此属性。 如果目录服务器允许匿名搜索，则无需输入密码。 | 字符串 | 无 |

表 14-8 安全性管理器属性（续）

| 属性名称 | 说明 | 类型 | 默认值 |
|--|--|-----|-------|
| imq.user_repository. ldap.principal | 代理用来绑定到目录服务器以进行搜索时使用的独特名称。如果目录服务器允许匿名搜索，则无需为此属性指定值。 | 字符串 | 无 |
| imq.user_repository. ldap.property_name | 待提供 | 待提供 | 待提供 |
| imq.user_repository. ldap.server | LDAP 服务器的 <i>host:port</i> ，其中 <i>host</i> 指定运行目录服务器的主机的全限定 DNS 名称， <i>port</i> 指定目录服务器用于通信的端口号。 要指定故障转移服务器的列表，请使用以下语法： <i>host1:port1 ldap://host2:port2</i> <i>ldap://host3:port3...</i> 列表中的条目以空格分隔。请注意，每个故障转移服务器地址均以 <i>ldap://</i> 开头。 即使您使用 SSL 并将属性 <i>imq.user_repository.ldap.ssl.enabled</i> 设置为 <i>true</i> ，也请使用此格式。您无需在地址中指定 “ <i>ldaps</i> ”。 | 字符串 | 无 |
| imq.user_repository. ldap.ssl.enabled | 一个布尔值，指定代理在与 LDAP 服务器通信时是否应使用 SSL 协议。 | 布尔值 | false |
| imq.user_repository. ldap.timeout | 用于搜索的时间限制（以秒为单位）。 | 整数 | 280 |
| imq.user_repository. ldap.uidattr | 提供商特定的属性标识符，其值用于唯一标识一个用户。例如：uid、cn 等。 | 字符串 | 无 |
| imq.user_repository. ldap.usrfilter | JNDI 搜索过滤器（以逻辑表达式的形式表示搜索查询）。通过为用户指定搜索过滤器，代理可以缩小搜索范围，从而使搜索更有效。有关详细信息，请参见位于以下位置的 JNDI 教程： http://java.sun.com/products/jndi/tutorial 。 此属性并不是必须设置的。 | 字符串 | 无 |

表 14-9 列出了 Message Queue 密钥库的可配置属性。请在 SSL 中使用这些属性。

表 14-9 密钥库属性

| 属性名称 | 说明 | 类型 | 默认值 |
|---------------------------|---|-----|-----|
| imq.keystore.file.dirpath | 适用于基于 SSL 的服务，包含密钥库文件的目录的路径。默认值：请参见附录 A “Message Queue 数据在特定操作系统中的位置” | 字符串 | 无 |
| imq.keystore.file.name | 适用于基于 SSL 的服务：密钥库文件的名称。 | 字符串 | 密钥库 |
| imq.keystore.password | 适用于基于 SSL 的服务：密钥库密码。 仅在密码文件中指定此属性。 | 字符串 | 无 |

监视和日志记录属性

表 14-10 列出了与监视和日志记录相关的属性。第一列列出了属性名称。对于每个属性名称，第二列对该属性进行了说明；第三列指定了其类型；而第四列则给出了其默认值。

表 14-10 监视服务属性

| 属性名称 | 说明 | 类型 | 默认值 |
|--|--|-----|---------------|
| imq.destination.logDeadMsgs ¹ | 一个布尔值，指定代理是否记录下列类型的事件： <ul style="list-style-type: none"> 目标已满，已达到其容量的最大值或最大消息数。 代理由于管理命令或传送确认之外的原因废弃消息。 代理将消息移动到停用消息队列中。 | 布尔值 | false |
| imq.log.console.output | 一个字符串，指定要写入到控制台的日志记录信息类别。可以是下列值之一： <ul style="list-style-type: none"> ALL NONE 以下的一个或多个值，以竖线 () 分隔：ERROR、WARNING 和 INFO。分别指定每种日志消息类别；所有消息类别均不包括其他类别。 | 字符串 | ERROR WARNING |

表 14-10 监视服务属性 (续)

| 属性名称 | 说明 | 类型 | 默认值 |
|---|---|-----|----------------|
| imq.log.console.stream | 一个字符串, 指定是否将控制台输出写入到标准输出 (OUT) 或标准错误 (ERR)。 | 字符串 | ERR |
| imq.log.file.dirpath | 包含日志文件的目录的路径 (取决于操作系统)。 | 字符串 | 请参见附录 A |
| imq.log.file.filename | 日志文件的名称。 | 字符串 | log.txt |
| imq.log.file.output | 要写入到控制台的日志记录信息的类别。可以是下列值之一: <ul style="list-style-type: none"> • ALL • NONE • 以下的一个或多个值, 以竖线 () 分隔: ERROR、WARNING 和 INFO。分别指定每种日志消息类别; 所有消息类别均不包括其他类别。 | 字符串 | ALL |
| imq.log.file.rolloverbytes ¹ | 日志文件的大小 (以字节为单位), 达到该值后输出将转移到新的日志文件。值 -1 将禁用基于文件大小的转移。 | 整数 | -1 |
| imq.log.file.rolloversecs ¹ | 日志文件的生存期 (以秒为单位), 达到该值后输出将转移到新的日志文件。值 -1 将禁用基于文件生存期的转移。 | 整数 | 604800 (一周) |
| imq.log.level ¹ | 一个指定记录程序级别的字符串: 可以写入到输出通道的输出的种类。包括指定的种类以及所有更高级别的种类。值分别为 (由高至低): ERROR 、 WARNING 和 INFO 。 | 字符串 | INFO |
| imq.log.syslog.facility | (仅对于 Solaris) 一个字符串, 指定 Message Queue 代理应将 syslog 记录为哪种系统日志工具。属性值镜像 syslog(3C) 手册页中列出的值。适用于 Message Queue 的值包括: LOG_USER 、 LOG_DAEMON 和 LOG_LOCAL0 到 LOG_LOCAL7 。 | 字符串 | LOG_DAEMON |
| imq.log.syslog.identity | (仅对于 Solaris) 应添加到每条记录到 syslog 中的消息前面的标识字符串。 默认值是 <code>imqbrokerd_\${imq.instanceName}</code> | 字符串 | 请参见说明 |
| imq.log.syslog.logconsole | (仅对于 Solaris) 一个布尔值, 指定如果无法将消息发送到 syslog , 是否将其写入到系统控制台。 | 布尔值 | false |

表 14-10 监视服务属性 (续)

| 属性名称 | 说明 | 类型 | 默认值 |
|------------------------------|---|-----|-------|
| imq.log.syslog.logpid | (仅对于 Solaris) 一个布尔值 (true/false), 指定是否将代理进程 ID 与消息一起记录。 | 布尔值 | true |
| imq.log.syslog.output | (仅对于 Solaris) 一个字符串, 指定写入到 syslogd(1M) 的日志记录信息类别。可以是下列值之一: <ul style="list-style-type: none"> • ALL • NONE • 以下的一个或多个值, 以竖线 () 分隔: ERROR、WARNING 和 INFO。分别指定每种日志消息类别; 所有消息类别均不包括其他类别。 | 字符串 | ERROR |
| imq.log.timezone | 一个表示日志时间戳时区的字符串。标识符与 java.util.TimeZone.getTimeZone() 使用的一致。例如: GMT、America/LosAngeles、Europe/Rome 和 Asia/Tokyo。 | 字符串 | 当地时区 |
| imq.metrics.enabled | 一个布尔值, 指定是否要将度量信息写入记录程序。不影响度量消息的生成 (请参见 imq.metrics.topic.enabled)。 | 布尔值 | true |
| imq.metrics.interval | 如果启用了度量日志记录 (imq.metrics.enabled=true), 将度量信息写入记录程序的时间间隔 (以秒为单位)。不影响度量消息的生成时间间隔 (请参见 imq.metrics.topic.interval)。 值 -1 表示从不。 | 整数 | -1 |
| imq.metrics.topic.enabled | 一个布尔值, 指定是否启用度量消息生成。如果为 false, 则试图订阅度量主题目标时会引发客户端异常。 | 布尔值 | true |
| imq.metrics.topic.interval | 生成度量消息 (发送到度量主题目标) 的时间间隔 (以秒为单位)。 | 整数 | 60 |
| imq.metrics.topic.persist | 一个布尔值, 指定度量消息是否为持久性消息。 | 布尔值 | false |
| imq.metrics.topic.timetolive | 度量消息 (发送到度量主题目标) 的生命周期 (以秒为单位)。 | 整数 | 300 |

1. 此属性可用于 `imqcmd update bkr`。

群集配置属性

表 14-11 概述了与代理群集相关的配置属性。

表 14-11 群集配置属性

| 属性名称 | 说明 | 类型 | 默认值 |
|--|---|-----|---|
| <code>imq.cluster.brokerlist</code> | 以逗号分隔的 <i>host:port</i> 条目列表，它用来标识群集中的所有代理，其中 <i>host</i> 是代理的主机名， <i>port</i> 是其端口映射器端口号。 示例： <code>host1:3000,host2:8000,ctrlhost</code> 群集中的所有代理必须具有相同的值。 | 字符串 | 无 |
| <code>imq.cluster.masterbroker</code> | 群集的主代理（如果有）的主机名和端口号。 该值的形式为 <i>host:port</i> ，其中 <i>host</i> 是主代理的主机名， <i>port</i> 是其端口映射器端口号。 示例： <code>ctrlhost:7676</code> 群集中的所有代理必须具有相同的值。 | 字符串 | 无 |
| <code>imq.cluster.url¹</code> | 群集配置文件（如果有）的 URL。 示例： <code>http://webserver/imq/cluster.properties</code> (对于 Web 服务器上的文件) <code>file:/net/mfsserver/imq/cluster.properties</code> (对于共享驱动器上的文件) 群集中的所有代理必须具有相同的值。 | 字符串 | 无 |
| <code>imq.cluster.hostname</code> | 当有多个可用主机时， <code>cluster</code> 连接服务所绑定的主机名或 IP 地址（用于群集中代理之间的内部通信）：例如，如果一台计算机中安装了多个网络接口卡。 可以单独地为群集中的每个代理指定。 | 字符串 | 从 <code>imq.hostname</code> 值继承（请参见第 283 页表 14-2） |

表 14-11 群集配置属性 (续)

| 属性名称 | 说明 | 类型 | 默认值 |
|-----------------------|---|-----|-----------|
| imq.cluster.port | cluster 连接服务的端口号。 可以单独地为群集中的每个代理指定。 | 整数 | 0 (动态分配。) |
| imq.cluster.transport | cluster 连接服务使用的网络传输协议。为了在代理之间传送安全、加密的消息, 请将此属性设置为 ssl。 群集中的所有代理必须具有相同的值。 | 字符串 | tcp |

1. 此属性可用于 imqcmd update bkr。

物理目标属性参考

本章介绍可为各种类型物理目标设置的属性。可在创建或更新物理目标时设置属性值。

对于自动创建的目标，可以在代理的实例配置文件中设置默认值（请参见第 286 页表 14-4）。

表 15-1 物理目标属性

| 属性 | 目标类型 | 默认值 | 说明 |
|-------------------------------|------|-------------|--|
| maxNumMsgs ¹ | 队列主题 | -1 (无限制) | 目标中允许的未使用消息的最大数量。 对于停用消息队列，默认值是 1000。 |
| maxTotalMsgBytes ¹ | 队列主题 | -1 (无限制) | 目标中未使用消息允许占用的内存的最大总量（以字节为单位）。 对于停用消息队列，默认值是 10 兆字节。 |

表 15-1 物理目标属性（续）

| 属性 | 目标类型 | 默认值 | 说明 |
|-----------------------|------|---------------|--|
| limitBehavior | 队列主题 | REJECT_NEWEST | <p>一个字符串，指定当达到内存限制的阈值时代理响应的方式。允许的值为：</p> <p>FLOW_CONTROL——减慢生产方。</p> <p>REMOVE_OLDEST——丢弃最旧的消息。</p> <p>REMOVE_LOW_PRIORITY——根据消息存在的时间丢弃优先级最低的消息（生产方客户机不会收到消息删除的通知）。</p> <p>REJECT_NEWEST——拒绝最新的消息。生产方客户机仅收到拒绝持久性消息的异常。要将此限制行为用于非持久性消息，请设置 <code>imqAckOnProduce</code> 连接工厂属性。</p> <p>如果将此属性设置为 <code>REMOVE_OLDEST</code> 或 <code>REMOVE_LOW_PRIORITY</code>，并将目标属性 <code>useDMQ</code> 设置为 <code>true</code>，代理将把超出限制的消息移动到停用消息队列中。</p> <p>与其他目标不同，停用消息队列自身具有默认限制行为 <code>REMOVE_OLDEST</code>，因此无法设置为 <code>FLOW_CONTROL</code> 行为。</p> |
| maxBytesPerMsg | 队列主题 | -1 (无限制) | <p>目标中允许的任何单个消息大小的最大值（以字节为单位）。生产方客户机会获得拒绝持久性消息的异常，但是不会收到拒绝非持久性消息的通知，除非设置了 <code>ackOnProduce</code> 属性。</p> |
| maxNumProducers | 队列主题 | -1 (无限制) | <p>目标允许的生产方的最大数量。当达到此限制时，将无法创建新的生产方。</p> <p>不能为停用消息队列设置此属性。</p> |
| maxNumActiveConsumers | 仅限队列 | 1 | <p>从队列目标的负荷平衡传送中可以处于活动状态的最大使用方数。值为 -1 表示不限制数量。</p> <p>平台版将此值限制为 2。</p> |
| maxNumBackupConsumers | 仅限队列 | 0 | <p>在出现从队列目标的负荷平衡传送错误时，代替这些活动使用方的最大备份使用方数量。值为 -1 表示不限制数量。</p> <p>平台版将此值限制为 1（一）。</p> |

表 15-1 物理目标属性（续）

| 属性 | 目标类型 | 默认值 | 说明 |
|------------------------|------|----------------------|---|
| consumerFlowLimit | 队列主题 | 主题: 1000 队列: 1000 | <p>将要在一批中传送给使用方的消息的最大数量。在负荷平衡队列传送中，为负荷平衡开始之前路由至活动使用方的队列消息的初始数量。</p> <p>目标使用方可以通过在连接中指定一个较小的值来覆盖此限制。值为 -1 表示不限制数量。</p> |
| localDeliveryPreferred | 仅限队列 | false | <p>一个布尔值（适用于代理群集中的负荷平衡队列传送），指定是否仅当本地代理中没有使用方时才将消息传送到远程使用方。要求目标不限于仅本地传送 (<code>isLocalOnly = false</code>)。</p> <p>此属性不适用于停用消息队列。</p> |
| isLocalOnly | 队列主题 | false | <p>一个布尔值（适用于队列群集中的目标），指定是否将目标限制为仅本地传送。如果为 <code>true</code>，则不能在其他代理上复制目标，并限制只将消息传送到本地使用方（连接到创建目标的代理的使用方）。在创建目标后，无法更改此属性。</p> <p>此属性不适用于停用消息队列。</p> |
| useDMQ | 队列主题 | true | <p>一个布尔值，指定是否应将停用消息发送到停用消息队列中，而不是将其废弃。</p> <p>此属性不适用于停用消息队列。</p> |

1. 在群集环境中，此属性将应用于群集中每个目标实例中，而不是应用到群集中的所有实例。

受管理对象的属性参考

本章提供了有关受管理对象属性的参考信息。其中包含以下各节：

- 第 305 页 “目标属性”
- 第 305 页 “连接工厂属性”
- 第 313 页 “SOAP 端点属性”

目标属性

表 16-1 列出了用于配置目标受管理对象的各种属性。

表 16-1 目标受管理对象的属性

| 属性名称 | 说明 | 类型 | 默认值 |
|---------------------------|----------|------------------|-----------------------------|
| imqDestinationDescription | 目标对象的说明。 | 字符串 | 无 |
| imqDestinationName | 物理目标的名称。 | 字符串 ¹ | Untitled_Destination_Object |

1. 目标名称只能包含字母数字字符（不能包含空格），且必须以字母字符或者字符 "_" 和 / 或 "\$" 开头。

连接工厂属性

本节包含有关用于配置连接工厂受管理对象的属性的参考信息。属性可划分为以下类别：

- 第 306 页 “连接处理”
- 第 309 页 “客户机标识”
- 第 310 页 “消息标题覆盖”

- 第 311 页 “可靠性和流控制”
- 第 312 页 “队列浏览器行为和服务器会话”
- 第 313 页 “JMS 定义的属性支持”

表 16-2 是连接工厂受管理对象属性的索引。第一列按字母顺序列出了各个属性；第二列列出了其类别；第三列是描述各属性的表格的交叉参考。

连接处理

表 16-2 列出了用于连接处理的连接工厂属性。

表 16-2 连接工厂属性：连接处理

| 属性名称 | 说明 | 类型 | 默认值 |
|--------------------------|--|-----|--|
| imqAddressList | 以逗号分隔的列表，其中包含一个或多个消息服务器地址。共有多种寻址方案，具体取决于您要使用的连接服务和端口分配方法。 有关如何指定地址列表的信息以及说明列表条目的示例，请参见第 307 页“imqAddressList 属性值的语法”。 | 字符串 | 现有的 Message Queue 3.0 地址（如果有），如果没有，则使用第 308 页表 16-3 中的第一个条目。 |
| imqAddressListBehavior | 一个字符串，用来指定连接尝试是采用 imqAddressList 属性中的地址顺序 (PRIORITY)，还是采用随机顺序 (RANDOM)。如果多个客户机使用同一连接工厂尝试连接，则可以使用随机顺序，以防止所有客户机都连接到同一地址。 | 字符串 | PRIORITY |
| imqAddressListIterations | 在尝试建立或重新建立连接时，客户机运行时重复访问 imqAddressList 的次数。值 -1 表示尝试次数是无限的。 | 整数 | 5 |
| imqPingInterval | 客户机运行时测试应用程序与代理之间连接情况的频率（以秒为单位）。 当值为 -1 或 0（零）时，将禁止客户机运行时定期测试连接。 | 整数 | 30 |
| imqReconnectEnabled | 一个布尔值，用来指定当连接断开时，客户机运行时是否应尝试重新连接到消息服务器（或 imqAddressList 中的地址列表）。 | 布尔值 | false |

表 16-2 连接工厂属性：连接处理（续）

| 属性名称 | 说明 | 类型 | 默认值 |
|-----------------------------------|---|-----|-------------------|
| <code>imqReconnectAttempts</code> | 在客户机运行时转为尝试列表中的下一个地址之前，对 <code>imqAddressList</code> 中的每个地址尝试连接（或重新连接）的次数。当值为 -1 时，表明尝试重新连接的次数是无限的（客户机运行时将不断尝试连接到第一个地址，直到连接成功为止）。 | 整数 | 0（零） |
| <code>imqReconnectInterval</code> | 两次重新连接尝试的时间间隔（以毫秒为单位）。此值适用于对 <code>imqAddressList</code> 中每个地址的尝试，也适用于对列表中连续地址的尝试。如果此值过小，代理将没有足够的恢复时间。如果此值过大，则重新连接可能出现无法接受的延迟。 | 长型 | 3000（毫秒） |
| <code>imqSSLIsHostTrusted</code> | 一个布尔值，用来指定客户机是否可以接受代理的自签名证书。要使用证书颁发机构的签名证书，请将此值设置为 <code>false</code> 。 | 布尔值 | <code>true</code> |

imqAddressList 属性值的语法

`imqbrokerlist` 值中的每个地址对应于客户机运行时可以连接的一个代理实例。

对于每个连接服务，请指定不同的代理地址。其语法通常如下所示：

```
scheme://address_syntax
```

要在列表中添加地址，请先添加一个逗号，然后再添加其他地址。列表可以包含任意数量的条目，其格式如下：

```
scheme://address_syntax,scheme://address_syntax...
```

变量 `scheme` 用来指定您使用的是下列哪些寻址类型，如表 16-3 中所述：`mq`、`mqtcp`、`mqssl`、`http` 或 `https`。变量 `address_syntax` 表示特定方案的代理地址。表 16-3 列出了寻址方案。第一列包含寻址方案的名称；第二列显示与该名称相关的连接服务；第三列是说明；第四列提供要使用的语法。

表 16-3 imqAddressList 属性的寻址方案

| 方案 | 连接服务 | 说明 | 语法 |
|-------|--------------|---|---|
| mq | ms ssljms | 提供用于 jms 或 ssljms 服务的动态端口分配。 指定端口映射器主机和端口。端口映射器动态地分配用于连接的端口。 | <i>[hostName][:port][/serviceName]</i> 对于 jms 连接服务, 将应用以下默认值: <i>hostName = localhost</i> <i>port = 7676</i> <i>serviceName = jms</i> 对于 ssljms 连接服务, 没有默认值。您必须指定所有变量。 |
| mqtcp | jms | 指定端口号并使用 jms 连接服务。 Message Queue 客户机运行时通过建立到指定主机和端口的 tcp 连接来建立连接。 | <i>hostName:port/jms</i> |
| mqssl | ssljms | 指定端口号并使用 ssljms 连接服务。 Message Queue 客户机运行时通过建立到指定主机和端口的安全 ssl 连接来建立连接。 | <i>port/ssljms</i> |
| http | httpjms | 使用 httpjms 连接服务。 客户机运行时建立到指定 URL 的 Message Queue 隧道 Servlet 的 HTTP 连接。必须将代理配置为访问 HTTP 隧道 Servlet。 | <i>http://hostName:port/contextRoot/tunnel</i> 如果多个代理实例使用同一隧道 Servlet, 则下面是连接到特定代理实例 (而非随机选择的实例) 的语法: <i>http://hostName:port/contextRoot/tunnel?ServerName=hostName:instanceName</i> |
| https | httpsjms | 使用 httpsjms 连接服务。 Message Queue 客户机运行时建立到指定 Message Queue 隧道 Servlet URL 的安全 HTTPS 连接。必须将代理配置为访问 HTTPS 隧道 Servlet。 | <i>https://hostName:port/contextRoot/tunnel</i> 如果多个代理实例使用同一隧道 Servlet, 则下面是连接到特定代理实例 (而非随机选择的实例) 的语法: <i>https://hostName:port/contextRoot/tunnel?ServerName=hostName:instanceName</i> |

表 16-4 包含有关寻址格式的示例。第一列是连接服务的名称。第二列指定示例中的主机是本地主机、未指定的主机、指定的主机还是不适用。第三列指定示例中的端口是指定的、未指定的还是不适用；第四列是示例。

表 16-4 消息服务器地址示例

| 连接服务 | 代理主机 | 端口 | 示例地址 |
|----------|-------|----------------|--|
| 未指定 | 本地主机 | 未指定 | 默认值 (mq://localhost:7676/jms) |
| 未指定 | 指定的主机 | 未指定 | myBkrHost (mq://myBkrHost:7676/jms) |
| 未指定 | 未指定 | 指定了端口映射器 端口 | 1012 (mq://localhost:1012/jms) |
| ssljms | 本地主机 | 未指定端口映射器 端口 | mq://localhost:7676/ssljms |
| ssljms | 指定的主机 | 端口映射器端口 | mq://myBkrHost:7676/ssljms |
| ssljms | 指定的主机 | 指定了端口映射器 端口 | mq://myBkrHost:1012/ssljms |
| jms | 本地主机 | 指定了服务端口 | mqtcp://localhost:1032/jms |
| ssljms | 指定的主机 | 指定了服务端口 | mqssl://myBkrHost:1034/ssljms |
| httpjms | 不适用 | 不适用 | http://webservr1:8085/imq/tunnel |
| httpsjms | 不适用 | 不适用 | https://webservr2:8090/imq/tunnel |

客户机标识

表 16-5 列出了用于客户机标识的连接工厂属性。

表 16-5 连接工厂属性：客户机标识

| 属性名称 | 说明 | 类型 | 默认值 |
|-----------------------|-----------------|-----|-------|
| imqDefaultUsername | 用于验证代理的默认用户名。 | 字符串 | guest |
| imqDefaultPassword | 用于验证代理的默认密码。 | 字符串 | guest |
| imqConfiguredClientID | 以管理方式配置的客户机 ID。 | 字符串 | null |

表 16-5 连接工厂属性：客户机标识（续）

| 属性名称 | 说明 | 类型 | 默认值 |
|-----------------------|---|-----|-------|
| imqDisableSetClientID | 一个布尔值，用来指定是否阻止客户机使用 JMS API 中的 setClientID() 方法来更改客户机 ID。 | 布尔值 | false |

消息标题覆盖

表 16-6 列出了用于覆盖 JMS 消息标题字段的连接工厂属性。

表 16-6 连接工厂属性：消息标题覆盖

| 属性名称 | 说明 | 类型 | 默认值 |
|--|--|-----|------------|
| imqOverrideJMSDeliveryMode | 一个布尔值，用来指定能否覆盖客户设置的 JMSDeliveryMode 字段。 | 布尔值 | false |
| imqJMSDeliveryMode | JMSDeliveryMode 的覆盖值。值为 1（非永久性）和 2（永久性）。 | 整数 | 2 |
| imqOverrideJMSExpiration | 一个布尔值，用来指定能否覆盖客户设置的 JMSExpiration 字段。 | 布尔值 | false |
| imqJMSExpiration | JMSExpiration 的覆盖值（以毫秒为单位）。 | 长型 | 0 (不过期) |
| imqOverrideJMSPriority | 一个布尔值，用来指定能否覆盖客户设置的 JMSPriority 字段。 | 布尔值 | false |
| imqJMSPriority | JMSPriority 的覆盖值（0 到 9 的整数）。 | 整数 | 4（标准） |
| imqOverrideJMSHeadersToTemporaryDestinations | 一个布尔值，用来指定覆盖是否适用于临时目标。 | 布尔值 | false |

可靠性和流控制

表 16-7 列出了用于配置可靠性和流控制的连接工厂属性。

表 16-7 连接工厂属性：可靠性和流控制

| 属性名称 | 说明 | 类型 | 默认值 |
|-------------------------------|---|-----|-------|
| imqAckTimeout | <p>客户机运行时在抛出异常前等待代理响应的最长时间（以毫秒为单位）。当值为 0 时表示无超时，客户机运行时将一直等待。</p> <p>在某些情况下，此值可能过低而导致客户机运行时超时。例如，当代理使用 LDAP 用户系统信息库和安全 (SSL) 连接验证每个用户时，第一次验证所需的时间可能会超过 30 秒。</p> | 字符串 | 0 |
| imqAckOnProduce | <p>一个字符串，用来指定代理如何响应来自生产方客户机的消息。</p> <p>如果将此属性设置为 true，代理将响应从生产方客户机接收到的所有 JMS 消息（持久性和非持久性消息）。在等待这些响应期间，生产方客户机线程将会阻塞。</p> <p>如果未指定此属性，代理仅响应持久性消息。在等待这些响应期间，生产方客户机线程将会阻塞。</p> | 字符串 | 未指定 |
| imqConnectionFlowCount | <p>测量批中 JMS 消息的数量。当此数目的 JMS 消息传送到客户机运行时，传送将被暂时挂起，允许传送任何已阻塞的控制消息。当收到客户机运行时通知时，有效负荷消息将继续进行传送，直到再次达到计数为止。</p> <p>如果将计数设置为 0，则对测量批中 JMS 消息的数目没有限制。非零设置允许客户机运行时测量消息流，以使 Message Queue 控制消息不被繁忙的 JMS 消息传送阻塞。</p> | 整数 | 100 |
| imqConnectionFlowLimitEnabled | <p>一个布尔值，用来指定是否使用 imqConnectionFlowLimit 值来限制连接级别的消息流。</p> | 布尔值 | false |
| imqConnectionFlowLimit | <p>可通过连接传送并在客户机运行时中进行缓冲（等待被使用）的最大消息数。但请注意，除非启用 imqConnectionFlowIsLimited，否则不会检查此限制。</p> <p>当传送到客户机运行时的 JMS 消息数（按照 imqConnectionFlowCount 控制的流量测量）超过此限制时，消息传送将停止。仅当未使用的消息数目低于使用此属性设置的值时，才会恢复消息传送。</p> <p>此限制防止用很长时间处理消息的使用方客户机必须处理可能导致它内存不足的大量挂起消息。</p> | 整数 | 1000 |

表 16-7 连接工厂属性：可靠性和流控制（续）

| 属性名称 | 说明 | 类型 | 默认值 |
|--------------------------|---|----|-----|
| imqConsumerFlowLimit | <p>每个使用方可通过连接传送并在客户机运行时中进行缓冲（等待被使用）的最大消息数。此限制用来改进在多使用方队列传送情况下使用方之间的负荷平衡（向任何一个使用方发送的消息数都是均衡的）。可以使用在代理端为队列的 <code>consumerFlowLimit</code> 属性设置的较小值覆盖此限制（请参见 <code>Message Queue</code> 管理指南中有关目标属性的信息）。</p> <p>此限制还有助于防止连接上的任一使用方迫使其他使用方断开连接。</p> <p>当传送到客户机运行时的 JMS 消息数超过任何使用方的此限制时，将停止传送该使用方的消息。仅当该使用方的未使用消息数下降到低于 <code>imqConsumerFlowThreshold</code> 设置的值时才会恢复。</p> <p>（请注意，如果为一个连接上所有使用方缓冲的消息总数超过 <code>imqConnectionFlowLimit</code>，则将停止通过此连接传送消息，直到消息总数降到连接限制以下。）</p> | 整数 | 100 |
| imqConsumerFlowThreshold | <p>每个使用方在客户机运行时中缓冲的消息数（以 <code>imqConsumerFlowLimit</code> 百分比的形式表示），低于此数值时将恢复使用方的消息传送。</p> | 整数 | 50 |

队列浏览器行为和服务器会话

表 16-8 描述影响客户机队列浏览的属性。

表 16-8 连接工厂属性：队列浏览器行为

| 属性名称 | 说明 | 类型 | 默认值 |
|---------------------------------------|--|----|-------------------|
| imqQueueBrowserMaxMessagesPerRetrieve | 客户机运行时在浏览队列目标的内容时一次检索的最大消息数。 | 整数 | 1000 |
| imqQueueBrowserRetrieveTimeout | 客户机运行时抛出异常之前在浏览队列目标内容时等待检索消息的最大时间值（以毫秒为单位）。 | 长型 | 60000 |
| imqLoadMaxToServerSession | 适用于 JMS 应用服务器工具，它是一个布尔值，用来指定 <code>Message Queue</code> 连接使用方是否应将 <code>maxMessages</code> 数量的消息加载到 <code>ServerSession</code> 会话中。如果为 <code>false</code> ，则客户机一次只加载一条消息。 | | <code>true</code> |

JMS 定义的属性支持

JMS 定义的属性是 JMS 保留的名称，JMS 提供者可以选择提供支持以增强客户机编程功能。表 16-9 描述了 Message Queue 所支持的 JMS 定义的属性。

表 16-9 连接工厂属性：JMS 定义的属性支持

| 属性名称 | 说明 | 类型 | 默认值 |
|------------------------|--|-----|-------|
| imqSetJMSXUserID | 一个布尔值，用来指定是否要在生成的消息上设置 JMS 定义的属性 JMSXUserID（发送消息的用户标识）。 | 布尔值 | false |
| imqSetJMSXAppID | 一个布尔值，用来指定是否要在生成的消息上设置 JMS 定义的属性 JMSXAppID（发送消息的应用程序标识）。 | 布尔值 | false |
| imqSetJMSXProducerTXID | 一个布尔值，用来指定是否要在生成的消息上设置 JMS 定义的属性 JMSXProducerTXID（生成此消息的事务的事务标识符）。 | 布尔值 | false |
| imqSetJMSXConsumerTXID | 一个布尔值，用来指定是否要在使用的消息上设置 JMS 定义的属性 JMSXConsumerTXID（使用此消息的事务的事务标识符）。 | 布尔值 | false |
| imqSetJMSXRcvTimestamp | 一个布尔值，用来指定是否要在使用的消息上设置 JMS 定义的属性 JMSXRcvTimestamp（消息传送到使用方所需的时间）。 | 布尔值 | false |

SOAP 端点属性

表 16-10 列出了用于为使用 SOAP 的应用程序配置端点 URL 的属性。有关使用 SOAP 的应用程序的信息，请参见 Message Queue Developer's Guide for Java Clients。

表 16-10 SOAP 端点属性

| 属性名称 | 说明 | 类型 | 默认值 |
|------------------------|---|-----|--------------------------|
| imqSOAPEndpointList | <p>由单个或多个以空格分隔的 URL 组成的列表，这些 URL 表示将消息发送到的 SOAP 端点。</p> <p>如果指定多个 URL，则将消息广播到列表中的所有 URL。每个 URL 都应与一个 Servlet 相关联，该 Servlet 可以接收和处理 SOAP 消息。</p> <p>示例： http://www.myServlet/ http://www.myServlet2/</p> | 字符串 | |
| imqEndpointName | <p>SOAP 端点的名称。</p> <p>示例：MyTopicEndpoint</p> | 字符串 | Untitled_Endpoint_Object |
| imqEndpointDescription | <p>SOAP 端点说明。</p> <p>示例： "imqEndpointDescription=my endpoints for broadcast"</p> | 字符串 | 端点对象的说明 |

JMS 资源适配器属性参考

利用 Message Queue JMS 资源适配器 (JMS RA)，可以通过标准 J2EE 连接器体系结构 (JCA) 来集成 Sun Java System Message Queue 和任何 J2EE 1.4 应用服务器。将 Message Queue JMS 资源适配器插入应用服务器后，在该应用服务器中部署的应用程序即可使用 Message Queue 来发送和接收 JMS 消息。

Message Queue JMS 资源适配器通过以下三个 JavaBean 组件来公开其配置属性：

- ResourceAdapter 配置从总体上影响资源适配器的行为。
- ManagedConnectionFactory 配置影响资源适配器创建的用于消息驱动 Bean (MDB) 的连接。
- ActivationSpec 配置影响消息端点，这些消息端点在它们与消息系统的交互中表示消息驱动 Bean (MDB)。

要设置这些组件的属性值，可使用应用服务器提供的用于配置和部署资源适配器及用于部署 MDB 的工具。

本章列出并说明了 Message Queue JMS 资源适配器的配置属性。它包含以下各节：

- [第 315 页 “ResourceAdapter JavaBean”](#)
- [第 317 页 “ManagedConnectionFactory JavaBean”](#)
- [第 318 页 “ActivationSpec JavaBean”](#)

ResourceAdapter JavaBean

ResourceAdapter 配置可用于配置默认 JMS 资源适配器行为。[表 17-1](#) 列出并说明了用来配置此 JavaBean 的属性。每个必需的属性以脚注形式标出。

表 17-1 资源适配器属性

| 名称 | 说明 | 默认值 |
|--------------------------|--|-----------------------------|
| addressList ¹ | <p>资源适配器建立的到 Message Queue 服务的连接，它是使用消息服务地址格式指定的。</p> <p>资源适配器提供默认值。</p> <p>属性名称 <code>addressList</code> 是 Sun Java System Message Queue 特有的，但与标准属性 <code>connectionURL</code> 具有相同的含义。Sun Java System Message Queue 提供了这两个属性名。必须设置 <code>connectionURL</code> 或 <code>ddressList</code>；它们是等效的。</p> | mq://localhost:7676 /jms |
| addressListBehavior | <p>一个字符串，指定资源适配器连接到 Message Queue 服务的方式。值为 <code>PRIORITY</code> 或 <code>RANDOM</code>。</p> <p><code>PRIORITY</code> 连接通过选择地址列表 (<code>addressList</code>) 中指定的第一个代理来选择 Message Queue 代理。</p> <p><code>RANDOM</code> 连接从地址列表中随机选择 Message Queue 代理。</p> <p>连接中断后进行重新连接时，<code>PRIORITY</code> 和 <code>RANDOM</code> 的作用是相同的。重新连接尝试将从连接中断的代理开始。如果尝试失败，资源适配器将按顺序处理活动地址列表中的条目。</p> | PRIORITY |
| addressListIterations | <p>重复访问地址列表的次数。此值适用于初始连接以及后来的重新连接尝试。</p> | 1 |
| connectionURL | <p>资源适配器建立的到 Message Queue 服务的连接，它是使用消息服务地址格式指定的。</p> <p>等效于 <code>addressList</code> 属性；请参见上面的说明以了解更详细的信息。</p> | mq://localhost:7676 /jms |
| userName ¹ | <p>资源适配器连接到 Message Queue 服务时使用的默认用户名。</p> <p>资源适配器提供默认值。</p> | guest |
| password ¹ | <p>资源适配器连接到 Message Queue 服务时使用的默认密码。</p> <p>资源适配器提供默认值。</p> | guest |
| reconnectAttempts | <p>尝试重新连接到地址列表中的单个条目的次数。在将 <code>reconnectEnabled</code> 设置为 <code>true</code> 时，将使用此属性。</p> | 6 |

表 17-1 资源适配器属性（续）

| 名称 | 说明 | 默认值 |
|-------------------|--|-------|
| reconnectEnabled | 一个布尔值，指定在连接中断后是否尝试重新连接。 重新连接尝试的行为是由 reconnectInterval 和 reconnectAttempts 值管理的。 | false |
| reconnectInterval | 两次重新连接尝试的间隔时间（以毫秒为单位）。在将 reconnectEnabled 设置为 true 时，将使用此属性。 | 30000 |

1. 此属性是必需的。

ManagedConnectionFactory JavaBean

受管理连接工厂提供并定义资源适配器为消息驱动 Bean 提供的连接。对于所设置的属性，如果 ResourceAdapter JavaBean 具有类似属性，则该设置将取代为 ResourceAdapter Bean 指定的类似值。

表 17-2 列出并说明了 Message Queue 资源适配器提供的受管理连接工厂的可配置属性。

表 17-2 受管理连接工厂属性

| 名称 | 说明 | 默认值 |
|---------------------|--|----------|
| addressList | 一个来自此受管理连接工厂的连接列表。 此属性格式遵循消息服务 addressList，如第 316 页表 17-1 中所述。如果未设置此值，连接将使用为 ResourceAdapter JavaBean 指定的 addressList 值（如该表中所述）。 | 无 |
| addressListBehavior | 一个字符串，指定资源适配器连接到 Message Queue 服务的方式。值为 PRIORITY 或 RANDOM。 PRIORITY 连接通过选择地址列表 (addressList) 中指定的第一个代理来选择 Message Queue 代理。 RANDOM 连接从地址列表中随机选择 Message Queue 代理。 连接中断后进行重新连接时，PRIORITY 和 RANDOM 的作用是相同的。重新连接尝试将从连接中断的代理开始。如果该尝试失败，连接尝试按顺序处理活动地址列表中的条目。 | PRIORITY |

表 17-2 受管理连接工厂属性（续）

| 名称 | 说明 | 默认值 |
|-----------------------|--|-------|
| addressListIterations | 循环访问地址列表的次数。此值适用于初始连接以及后来的重新连接尝试。 | 1 |
| clientID | 用于来自此受管理连接工厂的连接的客户机标识符。 | 无 |
| password | （可选）用于连接的密码。 如果未设置此值，连接将使用为 ResourceAdapter JavaBean 指定的密码（如第 316 页表 17-1 中所述）。 | guest |
| reconnectAttempts | 尝试重新连接到地址列表中的单个条目的次数。 | 6 |
| reconnectEnabled | 一个布尔值，指定在连接中断后是尝试重新连接，还是进行新的连接尝试。 重新连接尝试是由 reconnectInterval 和 reconnectAttempts 属性管理的。 | false |
| reconnectInterval | 在尝试重新连接到 Message Queue 服务之前等待的最短时间间隔（以毫秒为单位）。 | 30000 |
| userName | （可选）用于连接的用户名。 如果未设置此值，连接将使用为 ResourceAdapter JavaBean 指定的用户名（如第 316 页表 17-1 中所述）。 | guest |

ActivationSpec JavaBean

当指示资源适配器激活消息端点并将消息端点与消息驱动 Bean 相关联时，应用服务器将使用 ActivationSpec JavaBean 属性。

表 17-3 列出并说明了消息端点激活规范的可配置属性。该表列出了 Message Queue 资源适配器的特有属性以及企业 JavaBean 2.1 标准或 J2EE 连接器体系结构 (J2EE CA) 1.5 标准特有属性。

表 17-3 激活规范属性

| 名称 | 说明 | 默认值 |
|-----------------|--|------------------|
| acknowledgeMode | （可选）用于使用方的 JMS 会话确认模式。 这是标准的 EJB 2.1 和 J2EE CA 1.5 属性。 该值可以是 Auto-acknowledge 或 Dups-ok-acknowledge。 | Auto-acknowledge |

表 17-3 激活规范属性 (续)

| 名称 | 说明 | 默认值 |
|-------------------------------------|--|---|
| addressList | <p>(可选) 资源适配器代表消息端点所建立的连接的规范。</p> <p>此属性是 Message Queue JMS 资源适配器的特定属性。</p> <p>有效值必须符合消息服务连接地址语法。</p> | 从 ResourceAdapter JavaBean 配置中的 addressList 继承。 |
| clientId | <p>为此使用方创建的 JMS 连接所使用的 JMS 客户机 ID。</p> <p>如果将 subscriptionDurability 属性设置为 Durable, 则必须设置此属性。</p> <p>这是标准的 EJB 2.1 和 J2EE CA 1.5 属性。</p> | 无 |
| customAcknowledgeMode | <p>一个字符串, 指定 MDB 消息使用的模式。</p> <p>此属性的有效值是 No_acknowledge 或 null。</p> <p>只能将 No_acknowledge 模式用于非事务、非长期的主题订阅。如果对事务订阅或长期订阅使用此设置, 订阅激活将失败。</p> | 无 |
| destination | <p>MDB 所使用消息的目标的名称。</p> <p>这是必需的属性。它是标准的 EJB 2.1 和 J2EE CA1.5 属性。</p> <p>对于 Message Queue 目标受管理对象, 必须将该值设置为 destinationName 属性的值。</p> | 无 |
| destinationType | <p>destination 属性指定的目标类型。有效值是 javax.jms.Queue 或 javax.jms.Topic。</p> <p>这是必需的属性。它是标准的 EJB 2.1 和 J2EE CA1.5 属性。</p> | 无 |
| endpointExceptionRedeliveryAttempts | <p>消息传送期间当 MDB 抛出异常时, 向 MDB 重新传送消息的次数。</p> | 6 |
| messageSelector | <p>(可选) 用于对传送到使用方的消息进行过滤的 JMS 消息选择器。此值的类型是字符串。</p> <p>这是标准的 EJB 2.1 和 J2EE CA 1.5 属性。</p> | 无 |

表 17-3 激活规范属性（续）

| 名称 | 说明 | 默认值 |
|----------------------------|--|-------------------------|
| sendUndeliverableMsgsToDMQ | <p>一个布尔值，指定当 MDB 抛出运行时异常且重新传送尝试次数超出 <code>endpointExceptionRedeliveryAttempts</code> 值时，是否将消息置于停用消息队列中。</p> <p>如果为 <code>false</code>，Message Queue 代理将尝试将消息重新传送到任何有效的使用方，包括相同的 MDB。</p> | <code>true</code> |
| subscriptionDurability | <p>一个字符串，指定某一主题目标的使用方是长期的还是非长期的。该值可以是 <code>NonDurable</code> 或 <code>Durable</code>。</p> <p>对于非长期订阅，此属性是可选的；而对于长期订阅，则是必需的。如果将此值设置为 <code>Durable</code>，则还必须设置属性 <code>clientID</code> 和 <code>subscriptionName</code>。</p> <p>这是标准的 EJB 2.1 和 J2EE CA1.5 属性，只有将 <code>destinationType</code> 属性设置为 <code>avax.jms.Topic</code> 时才有效。</p> | <code>NonDurable</code> |
| subscriptionName | <p>一个字符串，用于对长期订阅进行命名。</p> <p>如果将 <code>subscriptionDurability</code> 属性设置为 <code>Durable</code>，则必须设置此属性。</p> <p>这是标准的 EJB 2.1 和 J2EE CA 1.5 属性。</p> | 无 |

度量参考

本章列出并说明了由 Message Queue 产品生成的度量。本章包含以下小节：

- 第 321 页 “JVM 度量”
- 第 322 页 “代理范围度量”
- 第 324 页 “连接服务度量”
- 第 325 页 “目标度量”

JVM 度量

表 18-1 列出并说明了代理为代理进程 JVM 堆生成的度量数据。对于每个度量，该表都显示了提供该度量的度量监视工具。

表 18-1 JVM 度量

| 度量数量 | 说明 | imqcmd metrics bkr (metricType) | 日志文件 | 度量消息 (度量主题) ² |
|------------|------------------|---------------------------------------|----------------|-----------------------------|
| JVM 堆：可用内存 | 可用于 JVM 堆的内存量 | 是 (cxn) | 是 | 是 (...jvm) |
| JVM 堆：总内存 | 当前 JVM 堆大小 | 是 (cxn) | 是 | 是 (...jvm) |
| JVM 堆：最大内存 | JVM 堆可以达到大小的最大值。 | 否 | 是 ¹ | 是 (...jvm) |

1. 仅在代理启动时显示。

2. 有关度量主题目标的名称，请参见第 197 页表 10-7。

代理范围度量

表 18-2 列出并说明了代理所报告的有关代理范围内的度量信息的数据。它还显示了哪些数据可以使用不同的度量监视工具获得。

表 18-2 代理范围度量

| 度量数量 | 说明 | imqcmd metrics bkr (metricType) | 日志文件 | 度量消息 (度量主题) ¹ |
|---------------|--|---------------------------------------|------|-----------------------------|
| 连接数据 | | | | |
| 连接数 | 当前对代理打开的连接数 | 是 (cxn) | 是 | 是 (...broker) |
| 线程数 | 所有连接服务当前使用的线程总数 | 是 (cxn) | 是 | 否 |
| 最小线程数 | 特定的线程数，达到该数量后，线程池中 将维护此数量的线程供连接服务使用 | 是 (cxn) | 是 | 否 |
| 最大线程数 | 特定的线程数，达到该数量后，线程池中 将不会添加新的线程以供连接服务使用 | 是 (cxn) | 是 | 否 |
| 存储消息数据 | | | | |
| 消息数 | 当前存储在代理内存和持久性存储中的 JMS 消息数 | 否 使用 query bkr | 否 | 是 (...broker) |
| 消息字节总数 | 当前存储在代理内存和持久性存储中的 JMS 消息字节数 | 否 使用 query bkr | 否 | 是 (...broker) |
| 消息流数据 | | | | |
| 流入消息数 | 自从代理上次启动以来流入该代理的 JMS 消息数 | 是 (ttl) | 是 | 是 (...broker) |
| 流入消息字节 | 自从代理上次启动以来流入该代理的 JMS 消息字节数 | 是 (ttl) | 是 | 是 (...broker) |
| 流入包数 | 自从代理上次启动以来流入该代理的 包（包括 JMS 消息和控制消息）数 | 是 (ttl) | 是 | 是 (...broker) |
| 流入包字节 | 自从代理上次启动以来流入该代理的 包（包括 JMS 消息和控制消息）字节 数 | 是 (ttl) | 是 | 是 (...broker) |

表 18-2 代理范围度量（续）

| 度量数量 | 说明 | imqcmd metrics bkr (metricType) | 日志文件 | 度量消息 (度量主题) ¹ |
|-------------|--------------------------------------|---------------------------------------|------|-----------------------------|
| 流出消息数 | 自从代理上次启动以来流出该代理的 JMS 消息数 | 是 (ttl) | 是 | 是 (...broker) |
| 流出消息字节 | 自从代理上次启动以来流出该代理的 JMS 消息字节数 | 是 (ttl) | 是 | 是 (...broker) |
| 流出包数 | 自从代理上次启动以来流出该代理的包（包括 JMS 消息和控制消息）数 | 是 (ttl) | 是 | 是 (...broker) |
| 流出包字节 | 自从代理上次启动以来流出该代理的包（包括 JMS 消息和控制消息）字节数 | 是 (ttl) | 是 | 是 (...broker) |
| 消息流入速率 | JMS 消息流入代理的当前速率 | 是 (rts) | 是 | 否 |
| 消息字节流入速率 | JMS 消息字节流入代理的当前速率 | 是 (rts) | 是 | 否 |
| 包流入速率 | 包（包括 JMS 消息和控制消息）流入代理的当前速率 | 是 (rts) | 是 | 否 |
| 包字节流入速率 | 包（包括 JMS 消息和控制消息）字节流入代理的当前速率 | 是 (rts) | 是 | 否 |
| 消息流出速率 | JMS 消息流出代理的当前速率 | 是 (rts) | 是 | 否 |
| 消息字节流出速率 | JMS 消息字节流出代理的当前速率 | 是 (rts) | 是 | 否 |
| 包流出速率 | 包（包括 JMS 消息和控制消息）流出代理的当前速率 | 是 (rts) | 是 | 否 |
| 包字节流出速率 | 包（包括 JMS 消息和控制消息）字节流出代理的当前速率 | 是 (rts) | 是 | 否 |
| 目标数据 | | | | |
| 目标数 | 代理中的物理目标数 | 否 | 否 | 是 (...broker) |

1. 有关度量主题目标的名称，请参见第 197 页表 10-7。

连接服务度量

表 18-3 列出并说明了代理为各个连接服务所报告的度量数据。它还显示了哪些数据可以使用不同的度量监视工具获得。

表 18-3 连接服务度量

| 度量数量 | 说明 | imqcmd metrics svc (metricType) | 日志文件 | 度量消息 (度量主题) |
|--------------|---|---------------------------------------|------|----------------|
| 连接数据 | | | | |
| 连接数 | 当前打开的连接数 | 是 (cxn) 还有 query svc | 否 | 否 |
| 线程数 | 当前使用的线程数 | 是 (cxn) 还有 query svc | 否 | 否 |
| 最小线程数 | 特定的供所有连接服务使用的总计线程数，达到该数量后，线程池中将维护此数量的线程供连接服务使用 | 是 (cxn) | 否 | 否 |
| 最大线程数 | 特定的供所有连接服务使用的总计线程数，达到该数量后，线程池中将不会添加新的线程以供连接服务使用 | 是 (cxn) | 否 | 否 |
| 消息流数据 | | | | |
| 流入消息数 | 自从代理上次启动以来流入连接服务的 JMS 消息数 | 是 (ttl) | 否 | 否 |
| 流入消息字节 | 自从代理上次启动以来流入连接服务的 JMS 消息字节数 | 是 (ttl) | 否 | 否 |
| 流入包数 | 自从代理上次启动以来流入连接服务的包（包括 JMS 消息和控制消息）数 | 是 (ttl) | 否 | 否 |
| 流入包字节 | 自从代理上次启动以来流入连接服务的包（包括 JMS 消息和控制消息）字节数 | 是 (ttl) | 否 | 否 |
| 流出消息数 | 自从代理上次启动以来流出连接服务的 JMS 消息数 | 是 (ttl) | 否 | 否 |
| 流出消息字节 | 自从代理上次启动以来流出连接服务的 JMS 消息字节数 | 是 (ttl) | 否 | 否 |

表 18-3 连接服务度量（续）

| 度量数量 | 说明 | imqcmd metrics svc (metricType) | 日志文件 | 度量消息 (度量主题) |
|----------|---------------------------------------|---------------------------------------|------|----------------|
| 流出包数 | 自从代理上次启动以来流出连接服务的包（包括 JMS 消息和控制消息）数 | 是 (ttl) | 否 | 否 |
| 流出包字节 | 自从代理上次启动以来流出连接服务的包（包括 JMS 消息和控制消息）字节数 | 是 (ttl) | 否 | 否 |
| 消息流入速率 | JMS 消息通过连接服务流入代理的当前速率。 | 是 (rts) | 否 | 否 |
| 消息字节流入速率 | JMS 消息字节流入连接服务的当前速率 | 是 (rts) | 否 | 否 |
| 包流入速率 | 包（包括 JMS 消息和控制消息）流入连接服务的当前速率 | 是 (rts) | 否 | 否 |
| 包字节流入速率 | 包（包括 JMS 消息和控制消息）字节流入连接服务的当前速率 | 是 (rts) | 否 | 否 |
| 消息流出速率 | JMS 消息流出连接服务的当前速率 | 是 (rts) | 否 | 否 |
| 消息字节流出速率 | JMS 消息字节流出连接服务的当前速率 | 是 (rts) | 否 | 否 |
| 包流出速率 | 包（包括 JMS 消息和控制消息）流出连接服务的当前速率 | 是 (rts) | 否 | 否 |
| 包字节流出速率 | 包（包括 JMS 消息和控制消息）字节流出连接服务的当前速率 | 是 (rts) | 否 | 否 |

目标度量

表 18-4 列出并说明了代理为各个目标所报告的度量数据。它还显示了哪些数据可以使用不同的度量监视工具获得。

表 18-4 目标度量

| 度量数量 | 说明 | imqcmd metrics dst (metricType) | 日志文件 | 度量消息 (度量主题) ¹ |
|---------------|--|---|----------------|-----------------------------|
| 使用方数据 | | | | |
| 使用方数 | 当前的使用方数。对于主题，此值包括非长期订阅、活动的长期订阅以及非活动的长期订阅。对于队列，此值包括活动使用方和备份使用方。 | 是 (con) | 否 | 是 (...destName) |
| 平均使用方数 | 自从代理上次启动以来的平均使用方数 | 是 (con) | 否 | 是 (...destName) |
| 最大使用方数 | 自从代理上次启动以来的最大使用方数 | 是 (con) | 否 | 是 (...destName) |
| 活动使用方数 | 当前活动使用方数 | 是 (con) | 否 | 是 (...destName) |
| 平均活动使用方数 | 自从代理上次启动以来的平均活动使用方数 | 是 (con) | 否 | 是 (...destName) |
| 最大活动使用方数 | 自从代理上次启动以来的最大活动使用方数 | 是 (con) | 否 | 是 (...destName) |
| 备份使用方数 | 当前备份使用方数（仅适用于队列） | 是 (con) | 否 | 是 (...destName) |
| 平均备份使用方数 | 自从代理上次启动以来的平均备份使用方数（仅适用于队列） | 是 (con) | 否 | 是 (...destName) |
| 最大备份使用方数 | 自从代理上次启动以来的最大备份使用方数（仅适用于队列） | 是 (con) | 否 | 是 (...destName) |
| 存储消息数据 | | | | |
| 消息数 | 当前存储在目标内存和持久性存储中的 JMS 消息数 | 是 (con) (ttl) (rts) 还有 dst | 否 query | 是 (...destName) |
| 平均消息数 | 自从代理上次启动以来存储在目标内存和持久性存储中的平均 JMS 消息数 | 是 (con) (ttl) (rts) | 否 | 是 (...destName) |

表 18-4 目标度量（续）

| 度量数量 | 说明 | imqcmd metrics dst (metricType) | 日志文件 | 度量消息 (度量主题) ¹ |
|--------------|---------------------------------------|---------------------------------------|------------|-----------------------------|
| 最大消息数 | 自从代理上次启动以来存储在目标内存和持久性存储中的最大 JMS 消息数 | 是 (con) (ttl) (rts) | 否 | 是 (...destName) |
| 消息字节总数 | 当前存储在目标内存和持久性存储中的 JMS 消息字节数 | 是 (ttl) (rts) 还有 dst | 否 query | 是 (...destName) |
| 平均消息字节总数 | 自从代理上次启动以来存储在目标内存和持久性存储中的平均 JMS 消息字节数 | 是 (ttl) (rts) | 否 | 是 (...destName) |
| 最大消息字节总数 | 自从代理上次启动以来存储在目标内存和持久性存储中的最大 JMS 消息字节数 | 是 (ttl) (rts) | 否 | 是 (...destName) |
| 最大消息字节 | 自从代理上次启动以来目标接收到的单个消息中 JMS 消息的最大字节数 | 是 (ttl) (rts) | 否 | 是 (...destName) |
| 消息流数据 | | | | |
| 流入消息数 | 自从代理上次启动以来流入此目标的 JMS 消息数 | 是 (ttl) | 否 | 是 (...destName) |
| 流入消息字节 | 自从代理上次启动以来流入此目标的 JMS 消息字节数 | 是 (ttl) | 否 | 是 (...destName) |
| 流出消息数 | 自从代理上次启动以来流出此目标的 JMS 消息数 | 是 (ttl) | 否 | 是 (...destName) |
| 流出消息字节 | 自从代理上次启动以来流出此目标的 JMS 消息字节数 | 是 (ttl) | 否 | 是 (...destName) |
| 消息流入速率 | JMS 消息流入目标的当前速率 | 是 (rts) | 否 | 否 |
| 消息流出速率 | JMS 消息流出目标的当前速率 | 是 (rts) | 否 | 否 |
| 消息字节流入速率 | JMS 消息字节流入目标的当前速率 | 是 (rts) | 否 | 否 |

表 18-4 目标度量（续）

| 度量数量 | 说明 | imqcmd metrics dst (metricType) | 日志文件 | 度量消息 (度量主题) ¹ |
|-----------------|--|---------------------------------------|------|-----------------------------|
| 消息字节流出速率 | JMS 消息字节流出目标的当前速率 | 是 (rts) | 否 | 否 |
| 磁盘使用情况数据 | | | | |
| 保留的磁盘 | 基于目标文件的存储中所有消息记录（活动的和空闲的）使用的磁盘空间（以字节为单位） | 是 (dsk) | 否 | 是 (...destName) |
| 已用的磁盘 | 基于目标文件的存储中活动消息记录使用的磁盘空间（以字节为单位） | 是 (dsk) | 否 | 是 (...destName) |
| 磁盘占用率 | 已用磁盘空间与保留磁盘空间的比率。比率越高，用于容纳活动消息时所占用的磁盘空间就越多 | 是 (dsk) | 否 | 是 (...destName) |

1. 有关度量主题目标的名称，请参见第 197 页表 10-7。

附录

- 附录 A Message Queue 数据在特定操作系统中的位置
- 附录 B Message Queue 接口的稳定性
- 附录 C HTTP/HTTPS 支持

Message Queue 数据在特定操作系统中的位置

Sun Java System 在不同的操作系统中，Message Queue 数据的存储位置也不同，如下章节所示。

本附录提供了以下操作系统中不同类型的 Message Queue 数据的位置：

- 第 331 页 “Solaris”
- 第 333 页 “Linux”
- 第 334 页 “Windows”

在下面的各表中，*instanceName* 标识了数据关联的代理实例的名称。

Solaris

表 A-1 中显示了 Message Queue 数据在 Solaris 操作系统上的位置。

如果在安装了 Sun Java System Application Server 独立版的 Solaris 中使用 Message Queue，目录结构与第 334 页 “Windows” 中所述的结构类似。

表 A-1 Message Queue 数据在 Solaris 平台上的位置

| 数据类型 | 在 Solaris 平台上的位置 |
|----------------------|--|
| 代理实例配置属性 | <code>/var/imq/instances/<i>instanceName</i>/props/config.properties</code> |
| 代理配置文件模板 | <code>/usr/share/lib/imq/props/broker/</code> |
| 持久性存储（消息、目标、长期订阅、事务） | <code>/var/imq/instances/<i>instanceName</i>/fs350/</code> 或 JDBC 可访问的数据存储库 |

表 A-1 Message Queue 数据在 Solaris 平台上的位置 (续)

| 数据类型 | 在 Solaris 平台上的位置 |
|--|---|
| 代理实例日志文件目录 (默认位置) | <code>/var/imq/instances/instanceName/log/</code> |
| 受管理对象 (对象存储库) | 您选择的本地目录 或 LDAP 服务器 |
| 安全性: 用户系统信息库 | <code>/var/imq/instances/instanceName/etc/passwd</code> 或 LDAP 服务器 |
| 安全性: 访问控制文件 (默认位置) | <code>/var/imq/instances/instanceName/etc/accesscontrol.properties</code> |
| 安全性: 密码文件目录 (默认位置) | <code>/var/imq/instances/instanceName/etc/</code> |
| 安全性: 密码文件示例 | <code>/etc/imq/passfile.sample</code> |
| 安全性: 代理的密钥库文件位置 | <code>/etc/imq/</code> |
| JavaDoc API 文档 | <code>/usr/share/javadoc/imq/index.html</code> |
| 示例应用程序和配置 | <code>/usr/demo/imq/</code> |
| Java 归档 (.jar) 文件、Web 归档 (.war) 文件和资源适配器归档 (.rar) 文件 | <code>/usr/share/lib/</code> |

Linux

表 A-2 中显示了 Message Queue 数据在 Linux 操作系统中的位置。

表 A-2 Message Queue 数据在 Linux 平台上的位置

| 数据类型 | 在 Windows 平台上的位置 |
|--|--|
| 代理实例配置属性 | <code>/var/opt/sun/mq/instances/instanceName/props/config.properties</code> |
| 代理配置文件模板 | <code>/opt/sun/mq/private/share/lib/props/</code> |
| 持久性存储（消息、目标、长期订阅、事务） | <code>/var/opt/sun/mq/instances/instanceName/fs350/</code> 或 JDBC 可访问的数据存储库 |
| 代理实例日志文件目录（默认位置） | <code>/var/opt/sun/mq/instances/instanceName/log/</code> |
| 受管理对象（对象存储库） | 您选择的本地目录或 LDAP 服务器 |
| 安全性：用户系统信息库 | <code>/var/opt/sun/mq/instances/instanceName/etc/passwd</code> 或 LDAP 服务器 |
| 安全性：访问控制文件（默认位置） | <code>/var/opt/sun/mq/instances/instanceName/etc/accesscontrol.properties</code> |
| 安全性：密码文件目录（默认位置） | <code>/var/opt/sun/mq/instances/instanceName/etc/</code> |
| 安全性：密码文件示例 | <code>/etc/opt/sun/mq/passfile.sample</code> |
| 安全性：代理的密钥库文件位置 | <code>/etc/opt/sun/mq/</code> |
| JavaDoc API 文档 | <code>/opt/sun/mq/javadoc/index.html</code> |
| 示例应用程序和配置 | <code>/opt/sun/mq/examples/</code> |
| Java 归档 (.jar) 文件、Web 归档 (.war) 文件和资源适配器归档 (.rar) 文件 | <code>/opt/sun/mq/share/lib/</code> |
| 共享库 (.so) 文件 | <code>/opt/sun/mq/lib/</code> |

Windows

表 A-3 中显示了 Message Queue 数据在 Windows 操作系统中的位置。

该表还显示了当 Message Queue 与 Sun Java System Application Server 独立版捆绑在一起时 Message Queue 数据在 Solaris 中的位置。该版本 Application Server 既不是与 Solaris 捆绑在一起，也不是与 Sun Java Enterprise System 捆绑在一起。可以使用表 A-3 中的路径名，但要将斜杠字符的方向由 Windows 中的反斜杠 (\) 更改为 Solaris 中的正斜杠 (/)。有关详细信息，请参见第 25 页“表 3”中的 IMQ_HOME 和 IMQ_VARHOME 定义。

表 A-3 Message Queue 数据在 Windows 平台上的位置

| 数据类型 | 在 Windows 平台上的位置 |
|----------------------|--|
| 代理实例配置属性 | IMQ_VARHOME\instances\ <i>instanceName</i> \props\ config.properties |
| 代理配置文件模板 | IMQ_HOME\lib\props\broker\ |
| 持久性存储（消息、目标、长期订阅、事务） | IMQ_VARHOME\instances\ <i>instanceName</i> \fs350\ 或 JDBC 可访问的数据存储库 |
| 代理实例日志文件目录（默认位置） | IMQ_VARHOME\instances\ <i>instanceName</i> \log\ |
| 受管理对象 （对象存储库） | 您选择的本地目录 或 LDAP 服务器 |
| 安全性：用户系统信息库 | IMQ_VARHOME\instances\ <i>instanceName</i> \etc\ passwd 或 LDAP 服务器 |
| 安全性：访问控制文件（默认） | IMQ_VARHOME\instances\ <i>instanceName</i> \ etc\accesscontrol.properties |
| 安全性：密码文件目录 （默认位置） | IMQ_HOME\etc\ |
| 安全性：密码文件示例 | IMQ_HOME\etc\passfile.sample |
| 安全性：代理的密钥库文件位置 | IMQ_HOME\etc\ 置 |

表 A-3 Message Queue 数据在 Windows 平台上的位置 (续)

| 数据类型 | 在 Windows 平台上的位置 |
|--|-----------------------------|
| JavaDoc API 文档 | IMQ_HOME\javadoc\index.html |
| 示例应用程序和配置 | IMQ_HOME\demo\ |
| Java 归档(.jar)文件、Web 归档(.war)文件和资源适配器归档(.rar)文件 | IMQ_HOME\lib\ |

Message Queue 接口的稳定性

Sun Java System Message Queue 使用的许多接口可以帮助管理员自动完成任务。本附录根据这些接口的稳定性对其进行分类。接口越稳定，在本产品的后续版本中对其进行更改的可能性就越小。

本附录中未列出的接口都是专用的，而非供用户使用的。

表 B-1 说明了稳定性分类方案。

表 B-1 接口稳定性分类方案

| 分类 | 说明 |
|-----|--|
| 专用 | 用户无法直接使用。在任何发行版中都可能对其进行改进或将其删除。 |
| 开发中 | 可供用户使用。可能会在主版本（例如 3.0 和 4.0）或次要版本（例如 3.1 和 3.2）中做不兼容更改。所有更改都将慎重逐步地进行。虽然我们会尽量保证所有更改都是兼容的，但不能保证做到这一点。 |
| 稳定 | 可供用户使用。只在主版本（例如 3.0 和 4.0）中做不兼容更改。 |
| 标准 | 可供用户使用。这些接口根据官方认可的标准进行定义，由标准组织控制。很少对这些接口做不兼容更改。 |
| 不稳定 | 可供用户使用。可能会在主版本（例如 3.0 和 4.0）或次要版本（例如 3.1 和 3.2）中做不兼容更改。用户需注意，在以后的发行版中可能会删除这些接口，也可能以一种不兼容的方式更改这些接口。建议用户不要在不稳定的接口上创建显式相关性。 |

表 B-2 列出了这些接口及其分类。

表 B-2 Message Queue 接口的稳定性

| 接口 | 分类 |
|---|-----|
| 命令行接口 | |
| imqbrokerd 命令行接口 | 开发中 |
| imqadmin 命令行接口 | 不稳定 |
| imqcmd 命令行接口 | 开发中 |
| imqdbmgr 命令行接口 | 不稳定 |
| imqkeytool 命令行接口 | 开发中 |
| imqobjmgr 命令行接口 | 开发中 |
| imqusermgr 命令行接口 | 不稳定 |
| imqbrokerd、imqadmin、imqcmd、imqdbmgr、imqkeytool、imqobjmgr 和 imqusermgr 的输出 | 不稳定 |
| 命令 | |
| imqobjmgr 命令文件 | 开发中 |
| imqbrokerd 命令 | 稳定 |
| imqadmin 命令 | 不稳定 |
| imqcmd 命令 | 稳定 |
| imqdbmgr 命令 | 不稳定 |
| imqkeytool 命令 | 稳定 |
| imqobjmgr 命令 | 稳定 |
| imqusermgr 命令 | 不稳定 |
| API | |
| JMS API (javax.jms) | 标准 |
| JAXM API (javax.xml) | 标准 |
| C-API | 开发中 |
| C-API 环境变量 | 不稳定 |
| 基于消息的监视 API | 开发中 |
| 受管理对象 API (com.sun.messaging) | 开发中 |
| JAR 文件和 WAR 文件 | |
| imq.jar 位置和名称 | 稳定 |

表 B-2 Message Queue 接口的稳定性 (续)

| 接口 | 分类 |
|--|-----|
| jms.jar 位置和名称 | 开发中 |
| imqbroker.jar 位置和名称 | 专用 |
| imqutil.jar 位置和名称 | 专用 |
| imqadmin.jar 位置和名称 | 专用 |
| imqservlet.jar 位置和名称 | 开发中 |
| imqhttp.war 位置和名称 | 开发中 |
| imqhttps.war 位置和名称 | 开发中 |
| imqjmsra.rar 位置和名称 | 开发中 |
| imqxm.jar 位置和名称 | 开发中 |
| jaxm-api.jar 位置和名称 | 开发中 |
| saaj-api.jar 位置和名称 | 开发中 |
| saaj-impl.jar 位置和名称 | 开发中 |
| activation.jar 位置和名称 | 开发中 |
| mail.jar 位置和名称 | 开发中 |
| dom4j.jar 位置和名称 | 专用 |
| fscontext.jar 位置和名称 | 不稳定 |
| 文件 | |
| 代理日志文件的位置和内容格式 | 不稳定 |
| 密码文件 | 不稳定 |
| accesscontrol.properties 文件 | 不稳定 |
| 系统目标 | |
| mq.sys.dmq 目标 | 稳定 |
| mq.metrics.* 目标 | 开发中 |
| 配置属性 | |
| Message Queue JMS 资源适配器配置属性 | 开发中 |
| Message Queue JMS 资源适配器 JavaBean 和 ActivationSpec 配置属性 | 开发中 |

表 B-2 Message Queue 接口的稳定性 (续)

| 接口 | 分类 |
|--|-----|
| 消息属性和格式 | |
| 停用消息队列消息属性 JMSXDeliveryCount | 标准 |
| 停用消息队列消息属性 JMS_SUN_* | 开发中 |
| Message Queue 客户机消息属性: JMS_SUN_* | 开发中 |
| 用于度量或监视消息的 JMS 消息格式 | 开发中 |
| 杂项 | |
| Message Queue JMS 资源适配器软件包 com.sun.messaging.jms.ra | 开发中 |
| 用于存储持久性消息的 JDBC 模式 | 开发中 |

HTTP/HTTPS 支持

Message Queue Enterprise Edition 支持 Java 客户机通过 HTTP 或安全 HTTP (HTTPS) 传输与代理通信，而不是通过直接 TCP 连接。HTTP/HTTPS 支持不适用于 C 客户机。

本附录介绍了实现此支持所使用的体系结构，并说明了如何设置客户机，使之能够使用基于 HTTP 的连接进行 Message Queue 消息传送。其中包含以下各节：

- 第 342 页 “HTTP/HTTPS 支持体系结构”
- 第 343 页 “实现 HTTP 支持”
- 第 352 页 “实现 HTTPS 支持”
- 第 364 页 “疑难解答”

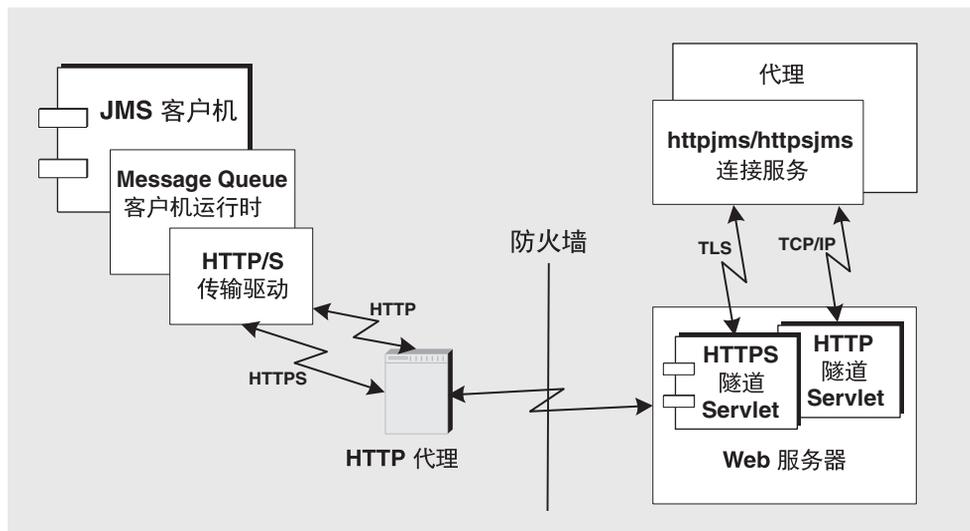
HTTP/HTTPS 支持体系结构

可以通过 HTTP/HTTPS 连接进行 Message Queue 消息传送。因为 HTTP/HTTPS 连接通常可以穿过防火墙，因此可以通过防火墙将客户机应用程序与代理隔开。

第 342 页图 C-1 列出了提供 HTTP/HTTPS 支持所需的主要组件。

- 在客户端，HTTP 或 HTTPS 传输驱动程序将 Message Queue 消息封装到 HTTP 请求中，并确保将这些请求以正确的顺序发送给 Web 服务器。
- 必要时，客户机可以使用 HTTP 代理服务器与代理进行通信。可以在启动客户机时使用命令行选项指定代理地址。有关详细信息，请参见第 347 页“使用 HTTP 代理”。
- 将 JMS 消息转发给代理之前，需要将 HTTP 或 HTTPS 隧道 Servlet（均随 Message Queue 一起提供）装入 Web 服务器，并使用它们从客户机 HTTP 请求中提取 JMS 消息。HTTP/HTTPS 隧道 Servlet 还将代理消息发送回客户机，以响应客户机发出的 HTTP 请求。可以使用一个 HTTP/HTTPS 隧道 Servlet 访问多个代理。

图 C-1 HTTP/HTTPS 支持体系结构



- 在代理端，httpjms 或 httpsjms 连接服务对来自相应隧道 Servlet 的消息进行展开和分离。

- 如果 Web 服务器失败并重新启动，所有连接都将恢复且不会对客户机产生影响。如果代理失败并重新启动，将引发异常，而客户机必须重新建立连接。如果 Web 服务器和代理同时失败（这种情况不太可能发生）而代理没有重新启动，Web 服务器将恢复客户机连接并等待代理恢复连接而不会通知客户机。要避免这种情况，请务必重新启动代理。

从图 C-1 可以看出，HTTP 和 HTTPS 支持所需的体系结构非常相似。主要区别在于，HTTPS（httpsjms 连接服务）中的隧道 Servlet 与客户机应用程序和代理之间的连接都是安全的。

与代理之间的安全连接是通过支持 SSL 的隧道 Servlet（Message Queue 的 HTTPS 隧道 Servlet）提供的，该 Servlet 会向请求连接的任何代理发送自签名证书。代理使用该证书建立与 HTTPS 隧道 Servlet 的加密连接。建立此连接后，客户机应用程序和 Web 服务器可以协商建立客户机应用程序和隧道 Servlet 之间的安全连接。

实现 HTTP 支持

以下各节介绍了实现 HTTP 支持所需执行的步骤。

► 实现 HTTP 支持

1. 在 Web 服务器上部署 HTTP 隧道 Servlet。
2. 配置代理的 httpjms 连接服务并启动代理。
3. 配置 HTTP 连接。

步骤 1：在 Web 服务器上部署 HTTP 隧道 Servlet

在 Web 服务器上部署 HTTP 隧道 Servlet 的常见方法有两种：

- 将其部署为支持 Servlet 2.1 或更早版本的 Web 服务器 Jar 文件
- 将其部署为支持 Servlet 2.2 或更高版本的 Web 服务器 Web 归档 (WAR) 文件

部署为 Jar 文件

部署 Message Queue 隧道 Servlet 分为三个步骤，首先要使主机 Web 服务器可以访问相应的 Jar 文件，然后要对该 Web 服务器进行配置，使之在启动时装入该 Servlet，最后需要指定该 Servlet URL 的上下文根分区。

隧道 Servlet Jar 文件 (imgServlet.jar) 包含 HTTP 隧道 Servlet 所需的所有类，该文件所在的目录因操作系统而异（请参见附录 A “[Message Queue 数据在特定操作系统中的位置](#)”）。

可以使用任何支持 Servlet 2.x 的 Web 服务器装入此 Servlet。Servlet 类名为：

```
com.sun.messaging.jmq.transport.  
httpunnel.servlet.HttpTunnelServlet
```

Web 服务器必须能够访问 imgServlet.jar 文件。如果计划在不同的主机上运行 Web 服务器和代理，应该在 Web 服务器能够访问的位置放置 imgServlet.jar 文件的一个副本。

还需要配置 Web 服务器以在启动时装入此 Servlet，可能需要指定该 Servlet URL 的上下文根分区（请参见第 347 页 “[示例 1: 在 Sun Java System Web Server 上部署 HTTP 隧道 Servlet](#)”）。

同时建议您禁用 Web 服务器的访问日志记录功能以提高性能。

部署为 Web 归档文件

要将 HTTP 隧道 Servlet 部署为 WAR 文件，需要使用 Web 服务器提供的部署机制。HTTP 隧道 Servlet WAR 文件 (imghttp.war) 所在的目录中包含 .jar、.war 和 .rar 文件，具体情况取决于您的操作系统（请参见附录 A “[Message Queue 数据在特定操作系统中的位置](#)”）。

WAR 文件包含一个部署描述符，该描述符包含 Web 服务器装入和运行 Servlet 所需的基本配置信息。根据 Web 服务器的不同，可能还需要指定该 Servlet URL 的上下文根分区（请参见第 350 页 “[示例 2: 在 Sun Java System Application Server 7.0 上部署 HTTP 隧道 Servlet](#)”）。

步骤 2：配置 httpjms 连接服务

默认情况下，未为代理激活 HTTP 支持，因此您需要重新配置代理以激活 httpjms 连接服务。重新配置后，可以按照第 64 页 “[以交互方式启动代理](#)” 中介绍的步骤启动代理。

► 激活 httpjms 连接服务

1. 打开代理的实例配置文件。

实例配置文件存储在一个目录中，该目录用与此配置文件相关联的代理实例的名称 (*instanceName*) 标识（请参见附录 A “[Message Queue 数据在特定操作系统中的位置](#)”）：

```
.../instances/instanceName/props/config.properties
```

2. 将 `httpjms` 值添加到 `imq.service.activelist` 属性中：

```
imq.service.activelist=jms,admin,httpjms
```

启动时，代理将在其主机上查找运行的 Web 服务器和 HTTP 隧道 Servlet。但是要访问远程隧道 Servlet，需要重新配置 `servletHost` 和 `servletPort` 连接服务属性。

还可以重新配置 `pullPeriod` 属性以提高性能。第 345 页表 C-1 详细介绍了 `httpjms` 连接服务配置属性。

表 C-1 httpjms 连接服务属性

| 属性名称 | 说明 |
|---|---|
| <code>imq.httpjms.http.servletHost</code> | 必要时可以更改此值，以指定运行 HTTP 隧道 Servlet 的主机的名称（主机名或 IP 地址）。（可以是远程主机或本地主机上的特定主机名。）默认值： <code>localhost</code> |
| <code>imq.httpjms.http.servletPort</code> | 需要更改此值，以指定代理用于访问 HTTP 隧道 Servlet 的端口号。（如果更改了 Web 服务器上的默认端口，则必须对此属性做相应的更改。）默认值： <code>7675</code> |
| <code>imq.httpjms.http.pullPeriod</code> | 指定客户机运行时从代理提取消息的 HTTP 请求的时间间隔（以秒为单位）。（请注意，该属性在代理上设置并传播到客户机运行时。）如果值为零或为负数，客户机将始终使一个 HTTP 请求处于待处理状态，这样可以随时尽快地提取消息。如果客户机的数量过多，会大量消耗 Web 服务器资源，可能导致服务器停止响应。在这种情况下，应将 <code>pullPeriod</code> 属性设置为正秒数。此属性设置客户机 HTTP 传输驱动程序在发出下一个提取请求之前等待的时间。将此属性设置为正值将以牺牲客户机响应时间为代价来保持 Web 服务器资源的可用性。默认值： <code>-1</code> |
| <code>imq.httpjms.http.connectionTimeout</code> | 指定客户机运行时在引发异常前等待 HTTP 隧道 Servlet 响应的的时间（以秒为单位）。（注意，该属性在代理上设置并传播到客户机运行时。）该属性还指定代理在与 HTTP 隧道 Servlet 进行了通信后等待断开的时间。在这种情况下可能会超时，因为代理和隧道 Servlet 不知道客户机正在访问的 HTTP Servlet 是否已经异常终止。默认值： <code>60</code> |

步骤 3: 配置 HTTP 连接

客户机应用程序必须使用正确配置的连接工厂管理对象，建立与代理之间的 HTTP 连接。本节介绍 HTTP 连接配置问题。

配置连接工厂

要实现 HTTP 支持，需要将连接工厂的 `imqAddressList` 属性设置为 HTTP 隧道 Servlet URL。HTTP 隧道 Servlet URL 的一般语法如下：

```
http://hostName:port/contextRoot/tunnel
```

其中，`hostName:port` 是作为 HTTP 隧道 Servlet 宿主的 Web 服务器的名称和端口，`contextRoot` 是在该 Web 服务器上部署隧道 Servlet 时设置的路径。

通常有关连接工厂属性，特别是有关 `imqAddressList` 属性的详细信息，请参见 *Message Queue Developer's Guide for Java Clients*。

可以按照以下方法之一设置连接工厂属性：

- 在创建连接工厂管理对象（请参见第 170 页“添加连接工厂”）的 `imqobjmgr` 命令中使用 `-o` 选项，或在使用管理控制台 (`imqadmin`) 创建连接工厂管理对象时设置属性。
- 在启动客户机的命令中使用 `-D` 选项（请参见 *Message Queue Developer's Guide for Java Clients*）。
- 通过编程方式在客户机代码中创建连接工厂（请参见 *Message Queue Developer's Guide for Java Clients*）之后，使用 API 调用设置其属性。

使用一个 Servlet 访问多个代理

即使正在运行多个代理，也无需配置多个 Web 服务器和多个 Servlet 实例。可以在并行运行的多个代理之间共享一个 Web 服务器和一个 HTTP 隧道 Servlet 实例。如果多个代理实例共享一个隧道 Servlet，则必须配置 `imqAddressList` 连接工厂属性，如下所示：

```
http://hostName:port/contextRoot/tunnel?ServerName=bkrHostName:instanceName
```

其中，`bkrHostName` 是代理实例主机名，`instanceName` 是您希望客户机访问的特定代理实例的名称。

要查看是否输入了正确的 `bkrHostName` 和 `bkrHostName` 字符串，可以通过浏览器访问 Servlet URL，生成 HTTP 隧道 Servlet 的状态报告。状态报告将列出 Servlet 正在访问的所有代理：

```

HTTP 隧道 servlet 就绪。
Servlet 开始时间: Thu May 30 01:08:18 PDT 2002
接受端口上来自代理的 TCP 连接: 7675
可用代理总数 = 2
代理列表:
  jpgserv:broker2
  cochin:broker1

```

使用 HTTP 代理

如果使用 HTTP 代理访问 HTTP 隧道 Servlet:

- 将 `http.proxyHost` 系统属性设置为代理服务器主机名。
- 将 `http.proxyPort` 系统属性设置为代理服务器端口号。

可以通过在启动客户机应用程序的命令中使用 `-D` 选项来设置这些属性。

示例 1: 在 Sun Java System Web Server 上部署 HTTP 隧道 Servlet

本节说明如何在 Sun Java System Web Server 上将 HTTP 隧道 Servlet 部署为 Jar 文件和 WAR 文件。使用的方法取决于 Sun Java System Web Server 的版本: 如果 Sun ONE Web Server 不支持 Servlet 2.2 或更高版本, 则不能处理 WAR 文件部署。

部署为 Jar 文件

以下说明适用于使用基于浏览器的管理 GUI 在 Sun Java System Web Server 6.1 上部署 HTTP 隧道 Servlet 的情况。此过程包含以下通用步骤:

1. 添加 Servlet
2. 配置 Servlet 虚拟路径
3. 装入 Servlet
4. 禁用 Servlet 访问日志

在以下小节中, 对这些步骤进行了说明。通过使用 Web 浏览器访问 Servlet URL 可以验证是否成功部署了 HTTP 隧道 Servlet。它应该显示状态信息。

添加 Servlet

► 添加隧道 Servlet

1. 选择 "Servlets" 选项卡。
2. 选择 “配置 Servlet 属性”。
3. 在 “Servlet 名称” 字段中指定隧道 Servlet 的名称。
4. 将 “Servlet 代码” (类名) 字段设置为以下值：
`com.sun.messaging.jmq.transport.httptunnel.servlet.HttpTunnelServlet`
5. 在 “Servlet 类路径” 字段中输入 `imqservlet.jar` 的完整路径。例如：
`/usr/share/lib/imq/imqservlet.jar (Solaris)`
`/opt/sun/mq/share/lib/imqservlet.jar (Linux)`
`IMQ_HOME/lib/imqservlet.jar (Windows)`
6. 在 “Servlet 参数” 字段中输入可选参数，如表 C-2 所示：

表 C-2 用于部署 HTTP 隧道 Servlet Jar 文件的 Servlet 参数

| 参数 | 默认值 | 参考 |
|--------------------------|------|-----------------|
| <code>servletHost</code> | 所有主机 | 请参见第 345 页表 C-1 |
| <code>servletPort</code> | 7675 | 请参见第 345 页表 C-1 |

如果使用两个参数，请使用逗号分隔：

```
servletPort=portNumber, servletHost=...
```

`servletHost` 和 `servletPort` 参数仅应用于 Web 服务器和代理之间的通信，仅当默认值存在问题时才需要进行设置。但是，在这种情况下，还必须相应地设置代理配置属性（请参见第 345 页表 C-1），例如：

```
imq.httpjms.http.servletPort
```

配置 Servlet 虚拟路径 (Servlet URL)

► 配置隧道 Servlet 的虚拟路径 (Servlet URL)

1. 选择 "Servlets" 选项卡。
2. 选择 “配置 Servlet 虚拟路径转换”。
3. 设置 “虚拟路径” 字段。

“虚拟路径”是隧道 Servlet URL 的 `/contextRoot/tunnel` 分区：

```
http://hostName:port/contextRoot/tunnel
```

例如，如果将 `contextRoot` 设置为 `img`，则“虚拟路径”字段将为：

```
/img/tunnel
```

4. 设置“Servlet 名称”字段的值，使之与第 348 页“添加 Servlet”中的步骤 3 中设置的值相同。

装入 Servlet

► 在 Web 服务器启动时装入隧道 Servlet

1. 选择“Servlets”选项卡。
2. 选择“配置全局属性”。
3. 在“启动 Servlets”字段中，输入与第 348 页“添加 Servlet”中的步骤 3 中的值相同的 Servlet 名称值。

禁用服务器访问日志

不是必须禁用服务器访问日志，但禁用服务器访问日志可以获得更佳的性能。

► 禁用服务器访问日志

1. 选择“状态”选项卡。
2. 选择“日志参考页”。
3. 使用日志客户机访问控制禁用日志记录。

部署为 WAR 文件

以下是对 Sun Java System Web Server 6.0 Service Pack 2 部署的说明。通过使用 Web 浏览器访问 Servlet URL，可以验证是否成功部署了 HTTP 隧道 Servlet。它应该显示状态信息。

► 将 HTTP 隧道 Servlet 部署为 WAR 文件

1. 在基于浏览器的管理 GUI 中，选择“虚拟 Server 类”选项卡并选择“管理类”。
2. 选择相应的虚拟服务器类名（例如，`defaultClass`）并单击“管理”按钮。
3. 选择“管理虚拟服务器”。
4. 选择相应的虚拟服务器名称并单击“管理”按钮。

5. 选择 “Web 应用程序” 选项卡。
6. 单击 “部署 Web 应用程序”。
7. 为 “WAR 文件打开” 和 “WAR 文件路径” 字段选择相应的值，以指向 `imghttp.war` 文件，该文件所在的目录因操作系统而异（请参见附录 A “[Message Queue 数据在特定操作系统中的位置](#)”）。
8. 在 “应用程序 URI” 字段中输入路径。

“应用程序 URI” 字段值为隧道 Servlet URL 的 `/contextRoot` 分区：

```
http://hostName:port/contextRoot/tunnel
```

例如，如果将 `contextRoot` 设置为 `img`，则 “应用程序 URI” 字段将为：

```
/img
```

9. 输入要在其中部署 Servlet 的安装目录路径（通常位于 Sun Java System Web Server 的安装根目录下）。
10. 单击 “确定”。
11. 重新启动 Web 服务器实例。

Servlet 现在即被部署到以下位置：

```
http://hostName:port/contextRoot/tunnel
```

客户机现在即可使用此 URL 连接至使用 HTTP 连接的消息服务。

示例 2：在 Sun Java System Application Server 7.0 上部署 HTTP 隧道 Servlet

本节说明如何在 Sun Java System Application Server 7.0 上将 HTTP 隧道 Servlet 部署为 WAR 文件。

需要执行两个步骤：

- 使用 Application Server 7.0 部署工具部署 HTTP 隧道 Servlet
- 修改应用服务器实例的 `server.policy` 文件

使用部署工具

► 在 Application Server 7.0 环境下部署 HTTP 隧道 Servlet

1. 在基于 Web 的管理 GUI 中，选择

“应用服务器” > “实例” > “服务器 1” > “应用程序” > “Web 应用程序”。

2. 单击“部署”按钮。
3. 在“文件路径”文本字段中，输入 HTTP 隧道 Servlet WAR 文件 (imqhttp.war) 的位置。

imqhttp.war 文件的位置因操作系统而异（请参见附录 A “Message Queue 数据在特定操作系统中的位置”）

4. 单击“确定”。
5. 在下一个屏幕上，设置“上下文根”文本字段的值。

“上下文根”字段值为隧道 Servlet URL 的 `/contextRoot` 分区：

```
http://hostName:port/contextRoot/tunnel
```

例如，可以将“上下文根”字段设置为 `/imq`。

6. 单击“确定”。

下一个屏幕显示隧道 Servlet 已成功部署，默认情况下处于启用状态，在这种情况下，它位于以下位置：

```
/var/opt/SUNWappserver7/domains/domain1/server1/applications/  
j2ee-modules/imqhttp_1
```

Servlet 现在即被部署到以下位置：

```
http://hostName:port/contextRoot/tunnel
```

客户机现在即可使用此 URL 连接至使用 HTTP 连接的消息服务。

修改 server.policy 文件

Application Server 7.0 实施了一套默认的安全策略，除非经过修改，否则它们将阻止 HTTP 隧道 Servlet 与 Message Queue 代理进行连接。

每个应用服务器实例都有一个包含其安全策略或规则的文件。例如，Solaris 上 server1 实例的此文件的位置为：

```
/var/opt/SUNWappserver7/domains/domain1/server1/config/server.policy
```

要使隧道 Servlet 与 Message Queue 代理进行连接，此文件中还必须有另外一个条目。

► 修改 Application Server 的 server.policy 文件

1. 打开 server.policy 文件。

2. 添加以下条目:

```
grant codeBase
"file:/var/opt/SUNWappserver7/domains/domain1/server1/
    applications/j2ee-modules/imqhttp_1/-
{
    permission java.net.SocketPermission "*",
        "connect, accept, resolve";
};
```

实现 HTTPS 支持

以下各节介绍了实现 HTTPS 支持所需执行的步骤。所需操作与第 343 页“实现 HTTP 支持”中的操作类似，只是增加了生成和访问 SSL 证书的步骤。

► 实现 HTTPS 支持

1. 为 HTTPS 隧道 Servlet 生成自签名证书。
2. 在 Web 服务器上部署 HTTPS 隧道 Servlet。
3. 配置代理的 httpsjms 连接服务并启动代理。
4. 配置 HTTPS 连接。

下文详细介绍了每个步骤。

步骤 1: 为 HTTPS 隧道 Servlet 生成自签名证书

Message Queue 的 SSL 支持用于保护所传输数据的安全，假定客户机正在与已知且可信任的服务器进行通信。因此，仅使用自签名的服务器证书实现 SSL。在 httpsjms 连接服务体系结构中，HTTPS 隧道 Servlet 充当代理和应用程序客户机的服务器。

运行 imqkeytool 实用程序，为隧道 Servlet 生成自签名证书。在命令提示符下输入以下内容：

```
imqkeytool -servlet keystore_location
```

命令行实用程序会提示您提供所需的信息。（在 Unix 操作系统上，可能需要以超级用户（root）身份运行 imqkeytool 命令，以获得创建密钥库所需的权限。）

首先，`imqkeytool` 提示您输入密钥库密码，然后提示您输入组织信息，最后提示您进行确认。收到确认后，它将在生成密钥对时暂停。然后它要求您输入密码以锁定特定密钥对（关键字密码），此时请按回车键响应此提示：这将使关键字密码与密钥库密码相同。

注 请记住您输入的密码，稍后必须将此密码提供给隧道 Servlet 使其能够打开密钥库。

运行 `imqkeytool` 命令以运行 JDK `keytool` 实用程序，生成自签名证书并将其放置在 Message Queue 的密钥库文件中，该文件的位置由 `keystore_location` 参数指定。（密钥库格式与 JDK1.2 `keytool` 支持的密钥库格式相同。）

注 HTTPS 隧道 Servlet 必须可以访问该密钥库。确保将 `keystore_location` 中已生成的密钥库移动 / 复制到 HTTPS 隧道 Servlet 可以访问的位置（请参见第 353 页“步骤 2：在 Web 服务器上部署 HTTPS 隧道 Servlet”）。

步骤 2：在 Web 服务器上部署 HTTPS 隧道 Servlet

在 Web 服务器上部署 HTTPS 隧道 Servlet 的常见方法有两种：

- 将其部署为支持 Servlet 2.1 或更早版本的 Web 服务器 Jar 文件
- 将其部署为支持 Servlet 2.2 或更高版本的 Web 服务器 Web 归档 (WAR) 文件

在任一种情况下，都应确保激活 Web 服务器的加密功能，确保客户机与代理之间的端到端通信是安全的。

部署为 Jar 文件

部署 Message Queue 隧道 Servlet 分为三个步骤，首先要使主机 Web 服务器可以访问相应的 Jar 文件，然后要对该 Web 服务器进行配置，使之在启动时装入该 Servlet，最后需要指定该 Servlet URL 的上下文根分区。

隧道 Servlet Jar 文件 (`imqservlet.jar`) 包含 HTTPS 隧道 Servlet 所需的所有类，该文件所在的目录因操作系统而异（请参见附录 A “Message Queue 数据在特定操作系统中的位置”）。

可以使用任何支持 Servlet 2.x 的 Web 服务器装入此 Servlet。Servlet 类名为：

```
com.sun.messaging.jmq.transport.  
httpunnel.servlet.HttpsTunnelServlet
```

Web 服务器必须能够访问 `imgqservlet.jar` 文件。如果计划在不同的主机上运行 Web 服务器和代理，应该在 Web 服务器能够访问的位置放置 `imgqservlet.jar` 文件的一个副本。

还需要配置 Web 服务器以在启动时装入此 Servlet，可能需要指定该 Servlet URL 的上下文根分区（请参见第 358 页“[示例 3: 在 Sun Java System Web Server 上部署 HTTPS 隧道 Servlet](#)”）。

确保 JSSE Jar 文件位于 Web 服务器上运行 Servlets 的类路径中。请参见 Web 服务器文档，了解如何执行此操作。

配置 Web 服务器有一个重要的方面，即指定 HTTPS 隧道 Servlet 用来与代理建立安全连接所需的自签名证书的位置和密码。必须将您在第 352 页“[步骤 1: 为 HTTPS 隧道 Servlet 生成自签名证书](#)”中创建的密钥库放置在 HTTPS 隧道 Servlet 可以访问的位置。

同时建议您禁用 Web 服务器的访问日志记录功能以提高性能。

部署为 Web 归档文件

要将 HTTPS 隧道 Servlet 部署为 WAR 文件，需要使用 Web 服务器提供的部署机制。HTTPS 隧道 Servlet WAR 文件 (`imghttps.war`) 所在的目录因操作系统而异（请参见附录 A“[Message Queue 数据在特定操作系统中的位置](#)”）。

WAR 文件包含一个部署描述符，该描述符包含 Web 服务器装入和运行 Servlet 所需的基本配置信息。根据 Web 服务器的不同，可能还需要指定该 Servlet URL 的上下文根分区（请参见第 362 页“[示例 4: 在 Sun Java System Application Server 7.0 上部署 HTTPS 隧道 Servlet](#)”）。

但是，`imghttps.war` 文件的部署描述符无法知道您放置隧道 Servlet 所需的密钥库的位置（请参见第 352 页“[步骤 1: 为 HTTPS 隧道 Servlet 生成自签名证书](#)”）。因此您需要在部署 `imghttps.war` 文件之前编辑隧道 Servlet 部署描述符（一个 XML 文件），以指定密钥库的位置。

步骤 3: 配置 httpsjms 连接服务

默认情况下, 未为代理激活 HTTPS 支持, 因此您需要重新配置代理以激活 httpsjms 连接服务。重新配置后, 可以按照第 64 页“以交互方式启动代理”中介绍的步骤启动代理。

► 激活 httpsjms 连接服务

1. 打开代理的实例配置文件。

实例配置文件存储在一个目录中, 该目录用与此配置文件相关联的代理实例的名称 (*instanceName*) 标识 (请参见附录 A “Message Queue 数据在特定操作系统中的位置”):

```
.../instances/instanceName/props/config.properties
```

2. 将 httpsjms 值添加到 `imq.service.activelist` 属性中:

```
imq.service.activelist=jms,admin,httpsjms
```

启动时, 代理将在其主机上查找运行的 Web 服务器和 HTTPS 隧道 Servlet。但是要访问远程隧道 Servlet, 需要重新配置 `servletHost` 和 `servletPort` 连接服务属性。

还可以重新配置 `pullPeriod` 属性以提高性能。表 C-3 详细介绍了 httpsjms 连接服务配置属性。

表 C-3 httpsjms 连接服务属性

| 属性名称 | 说明 |
|---|---|
| <code>imq.httpsjms.https.servletHost</code> | 必要时可以更改此值, 以指定运行 HTTPS 隧道 Servlet 的主机的名称 (主机名或 IP 地址)。(可以是远程主机或本地主机上的特定主机名。)默认值: localhost |
| <code>imq.httpsjms.https.servletPort</code> | 需要更改此值, 以指定代理用于访问 HTTPS 隧道 Servlet 的端口号。(如果更改了 Web 服务器上的默认端口, 则必须对此属性做相应的更改。)默认值: 7674 |
| <code>imq.httpsjms.https.pullPeriod</code> | 指定每一台客户机创建从代理提取消息的 HTTP 请求的时间间隔 (以秒为单位)。(请注意, 该属性在代理上设置并传播到客户机运行时。)如果值为零或为负数, 客户机将始终使一个 HTTP 请求处于待处理状态, 这样可以随时尽快地提取消息。如果客户机的数量过多, 会大量消耗 Web 服务器资源, 可能导致服务器停止响应。在这种情况下, 应将 <code>pullPeriod</code> 属性设置为正秒数。此属性设置客户机 HTTP 传输驱动程序在发出下一个提取请求之前等待的时间。将此属性设置为正值将以牺牲客户机响应时间为代价来保持 Web 服务器资源的可用性。默认值: -1 |

表 C-3 httpsjms 连接服务属性 (续)

| 属性名称 | 说明 |
|--------------------------------------|---|
| imq.httpsjms.https.connectionTimeout | 指定客户机运行时在引发异常前等待 HTTPS 隧道 Servlet 响应的时间（以秒为单位）（请注意，该属性在代理上设置并传播到客户机运行时。）该属性还指定代理在与 HTTPS 隧道 Servlet 进行了通信后等待断开的时间。在这种情况下可能会超时，因为代理和隧道 Servlet 不知道客户机正在访问的 HTTPS Servlet 是否已经异常终止。默认值：60 |

步骤 4：配置 HTTPS 连接

客户机应用程序必须使用正确配置的连接工厂管理对象，建立与代理之间的 HTTPS 连接。

但是，客户机还必须能够访问 Java 安全套接扩展 (JSSE) 提供的 SSL 库，并且必须具有一个根证书。SSL 库是随 JDK 1.4 一起提供的。如果您使用的是早期版本的 JDK，请参见“配置 JSSE”；否则，请转至“输入根证书”。

解决上述问题后，可以继续配置 HTTPS 连接。

配置 JSSE

► 配置 JSSE

1. 将 JSSE Jar 文件复制到 JRE_HOME/lib/ext 目录中。

包括 jsse.jar、jnet.jar 和 jcert.jar

2. 通过添加以下内容来静态添加 JSSE 安全服务供应商：

```
security.provider.n=com.sun.net.ssl.internal.ssl.Provider
```

将这些内容添加到 JRE_HOME/lib/security/java.security 文件中，（其中 *n* 是安全服务供应商软件包的下一个可用优先编号）。

3. 如果使用的不是 JDK1.4，则需要在启动客户机应用程序的命令中使用 -D 选项，以设置下列 JSSE 属性：

```
java.protocol.handler.pkgs=com.sun.net.ssl.internal.www.protocol
```

输入根证书

如果默认情况下签署您的 Web 服务器证书的 CA 的根用户证书不在信任数据库中，或者使用的是专用 Web 服务器证书，则必须将该证书添加到信任数据库中。如果出现上述情况，请执行以下操作；否则，请转至“配置连接工厂”。

如果证书保存在 *cert_file* 中，而您的密钥库为 *trust_store_file*，请运行以下命令：

```
JRE_HOME/bin/keytool -import -trustcacerts
-alias alias_for_certificate -file cert_file
-keystore trust_store_file
```

显示以下问题时请选择 YES: Trust this certificate?

还需要在启动客户机应用程序的命令中使用 -D 选项，以指定下列 JSSE 属性：

```
javax.net.ssl.trustStore=trust_store_file
javax.net.ssl.trustStorePassword=trust_store_passwd
```

配置连接工厂

要实现 HTTPS 支持，需要将连接工厂的 *imqAddressList* 属性设置为 HTTPS 隧道 Servlet URL。HTTPS 隧道 Servlet URL 的一般语法如下：

```
https://hostName:port/contextRoot/tunnel
```

其中，*hostName:port* 作为 HTTPS 隧道 Servlet 宿主的 Web 服务器的名称和端口，*contextRoot* 是在该 Web 服务器上部署隧道 Servlet 时设置的路径。

通常有关连接工厂属性，特别是有关 *imqAddressList* 属性的详细信息，请参见 *Message Queue Developer's Guide for Java Clients*。

可以按照以下方法之一设置连接工厂属性：

- 在创建连接工厂管理对象（请参见第 170 页“添加连接工厂”）的 *imqobjmgr* 命令中使用 -o 选项，或在使用管理控制台 (*imqadmin*) 创建连接工厂管理对象时设置属性。
- 在启动客户机应用程序的命令中使用 -D 选项（请参见 *Message Queue Developer's Guide for Java Clients*）。
- 通过编程方式在客户机应用程序代码中创建连接工厂（请参见 *Message Queue Developer's Guide for Java Clients*）之后，使用 API 调用设置其属性。

使用一个 Servlet 访问多个代理

即使正在运行多个代理，也无需配置多个 Web 服务器和多个 Servlet 实例。可以在并行运行的多个代理之间共享一个 Web 服务器和一个 HTTPS 隧道 Servlet 实例。如果多个代理实例共享一个隧道 Servlet，则必须配置 *imqAddressList* 连接工厂属性，如下所示：

```
https://hostName:port/contextRoot/tunnel?ServerName=bkrHostName:instanceName
```

其中, *bkrHostName* 是代理实例主机名, *instanceName* 是您希望客户机访问的特定代理实例的名称。

要查看是否输入了正确的 *bkrHostName* 和 *instanceName* 字符串, 可以通过浏览器访问 Servlet URL, 生成 HTTPS 隧道 Servlet 的状态报告。状态报告将列出 Servlet 正在访问的所有代理:

```
HTTPS 隧道 servlet 就绪。
Servlet 开始时间: Thu May 30 01:08:18 PDT 2002
接受端口上来自代理的安全连接: 7674
可用代理总数 = 2
代理列表:
  jpgserv:broker2
  cochin:broker1
```

使用 HTTP 代理

如果使用 HTTP 代理访问 HTTPS 隧道 Servlet:

- 将 `http.proxyHost` 系统属性设置为代理服务器主机名。
- 将 `http.proxyPort` 系统属性设置为代理服务器端口号。

可以通过在启动客户机应用程序的命令中使用 `-D` 选项来设置这些属性。

示例 3: 在 Sun Java System Web Server 上部署 HTTPS 隧道 Servlet

本节说明如何在 Sun Java System Web Server 上将 HTTPS 隧道 Servlet 部署为 Jar 文件和 WAR 文件。使用的方法取决于 Sun Java System Web Server 的版本: 如果 Sun ONE Web Server 不支持 Servlet 2.2 或更高版本, 则不能处理 WAR 文件部署。

部署为 Jar 文件

以下说明适用于使用基于浏览器的管理 GUI 在 Sun Java System Web Server 6.1 上部署 HTTP 隧道 Servlet 的情况。此过程包含以下通用步骤:

1. 添加 Servlet
2. 配置 Servlet 虚拟路径

3. 装入 Servlet
4. 禁用 Servlet 访问日志

在以下小节中，对这些步骤进行了说明。通过使用 Web 浏览器访问 Servlet URL 可以验证是否成功部署了 HTTPS 隧道 Servlet。它应该显示状态信息。

添加 Servlet

► 添加隧道 Servlet

1. 选择 "Servlets" 选项卡。
2. 选择 “配置 Servlet 属性”。
3. 在 “Servlet 名称” 字段中指定隧道 Servlet 的名称。
4. 将 “Servlet 代码”（类名）字段设置为以下值：


```
com.sun.messaging.jmq.transport.  
httpunnel.servlet.HttpsTunnelServlet
```
5. 在 “Servlet 类路径” 字段中输入 imqervlet.jar 的完整路径。例如：


```
/usr/share/lib/imq/imqervlet.jar (Solaris)  
  
/opt/sun/mq/share/lib/imqervlet.jar (Linux)  
  
IMQ_HOME/lib/imqervlet.jar (Windows)
```
6. 在 “Servlet 参数” 字段中输入必需参数和可选参数，如表 C-4 所示。

表 C-4 用于部署 HTTPS 隧道 Servlet Jar 文件的 Servlet 参数

| 参数 | 默认值 | 是否必需? |
|------------------|------|-------|
| keystoreLocation | 无 | 是 |
| keystorePassword | 无 | 是 |
| servletHost | 所有主机 | 否 |
| servletPort | 7674 | 否 |

参数之间以逗号分隔。例如：

```
keystoreLocation=keystore_location, keystorePassword=keystore_password,  
servletPort=portnumber
```

`servletHost` 和 `servletPort` 参数仅应用于 Web 服务器和代理之间的通信，仅当默认值存在问题时才需要进行设置。但是，在这种情况下，还必须相应地设置代理配置属性（请参见第 355 页表 C-3）。例如：

```
imq.httpsjms.https.servletPort
```

配置 Servlet 虚拟路径 (Servlet URL)

► 配置隧道 Servlet 的虚拟路径 (Servlet URL)

1. 选择 "Servlets" 选项卡。
2. 选择 “配置 Servlet 虚拟路径转换”。
3. 设置 “虚拟路径” 字段。

“虚拟路径” 是隧道 Servlet URL 的 `/contextRoot/tunnel` 分区：

```
https://hostName:port/contextRoot/tunnel
```

例如，如果将 `contextRoot` 设置为 `imq`，则 “虚拟路径” 字段将为：

```
/imq/tunnel
```

4. 设置 “Servlet 名称” 字段的值，使之与第 359 页 “添加 Servlet” 中的步骤 3 中设置的值相同。

装入 Servlet

► 在 Web 服务器启动时装入隧道 Servlet

1. 选择 "Servlets" 选项卡。
2. 选择 “配置全局属性”。
3. 在 “开启 Servlets” 字段中，输入与第 359 页 “添加 Servlet” 中的步骤 3 中的值相同的 Servlet 名称值。

禁用服务器访问日志

不是必须禁用服务器访问日志，但禁用服务器访问日志可以获得更佳的性能。

► 禁用服务器访问日志

1. 选择 “状态” 选项卡。
2. 选择 “日志参考页”。
3. 使用日志客户机访问控制禁用日志记录。

部署为 WAR 文件

以下是对 Sun Java System Web Server 6.0 Service Pack 2 部署的说明。通过使用 Web 浏览器访问 Servlet URL，可以验证是否成功部署了 HTTPS 隧道 Servlet。它应该显示状态信息。

部署 HTTPS 隧道 Servlet 之前，请确保 JSSE Jar 文件位于 Web 服务器的类路径中。要达到此目的，最简单的方法是将 jsse.jar、jnet.jar 和 jcert.jar 文件复制到 IWS60_TOPDIR/bin/https/jre/lib/ext 中。

另外，部署 HTTPS 隧道 Servlet 之前，必须修改部署描述符，使其指向放置密钥库文件的位置，并指定密钥库密码。

► 修改 HTTPS 隧道 Servlet WAR 文件

1. 将 WAR 文件复制到临时目录中。

```
cp /usr/share/lib/imq/imqhttps.war /tmp (Solaris)
cp /opt/sun/mq/share/lib/imqhttps.war /tmp (Linux)
cp IMQ_HOME/lib/imqhttps.war /tmp (Windows)
```

2. 使临时目录成为当前目录。

```
$ cd /tmp
```

3. 提取 WAR 文件的内容。

```
$ jar xvf imqhttps.war
```

4. 列出 WAR 文件的部署描述符。

```
$ ls -l WEB-INF/web.xml
```

5. 编辑 web.xml 文件，以提供正确的 keystoreLocation 和 keystorePassword 参数值（如有必要，还包括 servletPort 和 servletHost 参数）。

6. 重新装入 WAR 文件的内容。

```
$ jar uvf imqhttps.war WEB-INF/web.xml
```

现在即可使用这个修改后的 imqhttps.war 文件部署 HTTPS 隧道 Servlet 了。（如果担心会泄漏密钥库密码，可以使用文件系统权限限制对 imqhttps.war 文件的访问。）

► 将 HTTPS 隧道 Servlet 部署为 WAR 文件

1. 在基于浏览器的管理 GUI 中，选择“虚拟服务器类”选项卡。单击“管理类”。
2. 选择相应的虚拟服务器类名（例如，defaultClass）并单击“管理”按钮。

3. 选择“管理虚拟服务器”。
4. 选择相应的虚拟服务器名称并单击“管理”按钮。
5. 选择“Web 应用程序”选项卡。
6. 单击“部署 Web 应用程序”。
7. 为“WAR 文件打开”和“WAR 文件路径”字段选择相应的值，以指向修改后的 `imqhttps.war` 文件（请参见第 361 页“修改 HTTPS 隧道 Servlet WAR 文件”）。
8. 在“应用程序 URI”字段中输入路径。

“应用程序 URI”字段值为隧道 Servlet URL 的 `/contextRoot` 分区：

```
https://hostName:port/contextRoot/tunnel
```

例如，如果将 `contextRoot` 设置为 `imq`，则“应用程序 URI”字段将为：

```
/imq
```

9. 输入要在其中部署 Servlet 的安装目录路径（通常位于 Sun Java System Web Server 的安装根目录下）。
10. 单击“确定”。
11. 重新启动 Web 服务器实例。

Servlet 现在即被部署到以下位置：

```
https://hostName:port/imq/tunnel
```

客户机现在即可使用此 URL 连接至使用安全 HTTPS 连接的消息服务。

示例 4：在 Sun Java System Application Server 7.0 上部署 HTTPS 隧道 Servlet

本节说明如何在 Sun Java System Application Server 7.0 上将 HTTPS 隧道 Servlet 部署为 WAR 文件。

需要执行两个步骤：

- 使用 Application Server 7.0 部署工具部署 HTTPS 隧道 Servlet
- 修改应用服务器实例的 `server.policy` 文件

使用部署工具

► 在 Application Server 7.0 环境下部署 HTTPS 隧道 Servlet

1. 在基于 Web 的管理 GUI 中，选择
“应用服务器” > “实例” > “服务器 1” > “应用程序” > “Web 应用程序”。

2. 单击“部署”按钮。

3. 在“文件路径：”文本字段中，输入 HTTPS 隧道 Servlet WAR 文件 (imqhttps.war) 的位置。

imqhttps.war 文件的位置因操作系统而异（请参见附录 A “Message Queue 数据在特定操作系统中的位置”）

4. 单击“确定”。

5. 在下一个屏幕上，设置“上下文根”文本字段的值。

“上下文根”字段值为隧道 Servlet URL 的 `/contextRoot` 分区：

```
https://hostName:port/contextRoot/tunnel
```

例如，可以将“上下文根”字段设置为：

```
/imq
```

6. 单击“确定”。

下一个屏幕显示隧道 Servlet 已成功部署，默认情况下处于启用状态，在这种情况下，它位于以下位置：

```
/var/opt/SUNWappserver7/domains/domain1/server1/applications/  
j2ee-modules/imqhttps_1
```

Servlet 现在即被部署到以下位置：

```
https://hostName:port/contextRoot/tunnel
```

客户机现在即可使用此 URL 连接至使用 HTTPS 连接的消息服务。

修改 server.policy 文件

Application Server 7.0 实施了一套默认的安全策略，除非经过修改，否则它们将阻止 HTTPS 隧道 Servlet 与 Message Queue 代理进行连接。

每个应用服务器实例都有一个包含其安全策略或规则的文件。例如，Solaris 上 server1 实例的此文件的位置为：

```
/var/opt/SUNWappserver7/domains/domain1/server1/config/server.policy
```

要使隧道 Servlet 与 Message Queue 代理进行连接，此文件中还必须有另外一个条目。

► 修改 Application Server 的 server.policy 文件

1. 打开 server.policy 文件。
2. 添加以下条目：

```
grant codeBase
"file:/var/opt/SUNWappserver7/domains/domain1/server1/
    applications/j2ee-modules/imqhttps_1/-"
{
    permission java.net.SocketPermission "*",
        "connect,accept,resolve";
};
```

疑难解答

本节介绍了 HTTP 或 HTTPS 连接可能会出现的问题，并提供如何解决这些问题的指导。

服务器或代理故障

如果 Web 服务器出现故障并重新启动，所有连接都将恢复且不会对客户机产生影响。然而，如果代理出现故障并重新启动，将引发异常，而客户机必须重新建立连接。

如果 Web 服务器和代理同时出现故障，而代理没有重新启动，则 Web 服务器将恢复客户机连接并继续等待代理恢复连接而不会通知客户机。要避免这种情况，请务必重新启动代理。

客户机无法通过隧道 Servlet 进行连接

如果 HTTPS 客户机无法通过隧道 Servlet 与代理建立连接，请执行以下操作：

1. 启动 Servlet 和代理。
2. 使用浏览器通过 HTTPS 隧道 Servlet URL 手动访问 Servlet。

3. 使用以下管理命令暂停并恢复连接:

```
imqcmd pause svc -n httpsjms -u admin  
imqcmd resume svc -n httpsjms -u admin
```

服务恢复后，HTTPS 客户机应该能够通过隧道 Servlet 与代理建立连接。

词汇表

有关 Message Queue 术语的信息，请参见 Message Queue 技术概述中的词汇表。有关 Sun Java System 产品套件中使用的术语的完整列表，请参见 Java Enterprise System 词汇表 (<http://docs.sun.com/doc/819-1935>)。

A

- acknowledgeMode 激活规范属性 318
- ActivationSpec JavaBean 318
- addressList 激活规范属性 319
- addressList 受管理连接工厂属性 317
- addressList 资源适配器属性 316, 317
- addressListBehavior 受管理连接工厂属性 317
- addressListBehavior 资源适配器属性 316
- addressListIterations 受管理连接工厂属性 318
- addressListIterations 资源适配器属性 316
- ADMIN 服务类型 73
- admin 连接服务 74, 107
- admin 用户 131, 135, 137
- admin 组 133
- anonymous 组 133
- API 文档 332, 333, 335
- AUTOSTART 属性 66
- 安全套接字层标准, **请参见** SSL
- 安全性
 - 对象存储库 158
 - 管理器, **请参见** 安全性管理器
 - 加密, **请参见** 加密
 - 授权, **请参见** 授权
 - 验证, **请参见** 验证
- 安全性管理器
 - 关于 83
 - 属性 291
 - 作为代理组件 73

B

- 版本 255
- 帮助 (命令行) 255

C

- clientId 激活规范属性 319, 320
- clientID 受管理连接工厂属性 318
- config.properties 文件 92, 180, 181, 182
- connectionURL 资源适配器属性 316
- customAcknowledgeMode 激活规范属性 319
- 操作系统
 - 调整 Solaris 性能 216
 - 性能影响 212
- 插入的持久性
 - 关于 82
 - 设置 94
 - 性能调整 220
- 查询
 - 代理 102
 - 连接服务 108, 112, 265
- 长期订阅
 - 管理 112
 - id 267
 - 列出 112, 266
 - 清除消息 266
 - 销毁 113, 266

D

- 性能影响 209
- 持久性
 - 安全 97
 - 插入的, **请参见**插入的持久性
 - 持久性管理器, **请参见**持久性管理器
 - JDBC, **请参见**JDBC 持久性
 - 内置的 81
 - 数据存储库 **请参见**数据存储库
 - 选项 (图) 80
- 持久性管理器
 - 插入的持久性, 和 93
 - 关于 80
 - 数据存储库, **请参见**数据存储库
 - 属性 288
 - 作为代理组件 73
- 重新连接, 自动 **请参见**自动重新连接
- 重新启动代理 105, 262
- 重新传送标志 77
- 传输协议
 - 相对速度 213
 - 协议类型, **请参见**协议类型
 - 性能调整 217
 - 性能影响 213
- 传送模式
 - 性能影响 207
- 磁盘空间
 - 回收 125
 - 物理目标占用 124

D

- default.properties 文件 90
- destination 激活规范属性 319
- destinationType 激活规范属性 319, 320
- 代理
 - 安全性管理器, **请参见**安全性管理器
 - 查询 102
 - 持久性管理器, **请参见**持久性管理器
 - 从故障中恢复 80
 - 度量, **请参见**代理度量

- 访问控制, **请参见**授权
- 服务 (图) 72
- 更新属性 103
- 关闭 105
- 管理 99
- HTTP 支持 343
- httpjms 连接服务属性 345
- HTTPS 支持 352
- httpsjms 连接服务属性 355
- 互连, **请参见**代理群集
- 恢复 104, 105, 262
- 监视, **请参见**代理监视服务
- 连接 179
- 连接服务, **请参见**连接服务
- 列出连接服务 107
- 内存管理 78, 119, 215
- 配置文件, **请参见**配置文件
- 启动所需的权限 65
- 群集, **请参见**代理群集
- 日志记录, **请参见**记录程序
- 删除 69
- 实例名称 257
- 实例配置属性 92
- 时钟同步 64
- 属性 (参考) 279
- 停用消息队列 126
- 显示属性 102
- 限制行为 78, 215
- 消息流控制, **请参见**消息流控制
- 消息路由, **请参见**消息路由器
- 消息容量 79, 104, 282, 285
- 与 SSL 一起启动 148
- 暂停 104, 105, 262
- 重新启动 80, 105, 262
- 自动创建物理目标属性 286
- 自动重新启动 66
- 组件及功能 (表) 72
- 作为 Windows 服务运行 66
- 代理 Windows 服务的启动参数 67
- 代理标准
 - 记录程序属性 297
- 代理度量

- 报告间隔, 记录程序 257
 - 度量数量 (表) 322
 - 度量消息 88
 - 记录程序属性 89, 190
 - 使用 imqcmd 106, 194, 196
 - 使用代理日志文件 191
 - 使用基于消息的监视 197
 - 代理故障和安全连接 364
 - 代理监视服务
 - 关于 86
 - 属性 295
 - 代理群集
 - 安全的交叉代理连接 180
 - 连接代理 179
 - 配置更改记录 182
 - 配置属性 178, 298
 - 配置文件 178, 179, 298
 - 使用原因 215
 - 体系结构 215
 - 添加代理到 180
 - 物理目标的复制 119
 - 性能影响 215
 - 选项以指定 256
 - 暂停物理目标 123
 - 代理响应
 - 客户机等待周期 164, 311
 - 生产时 311
 - 度量
 - 关于 87
 - 数据, **请参见**度量数据
 - 消息, **请参见**度量消息
 - 主题目标 88, 197
 - 度量监视工具
 - 比较 185
 - 基于消息的监视 API 196
 - Message Queue 命令行实用程序 (imqcmd) 192
 - Message Queue 日志文件 190
 - 度量数据
 - 代理, **请参见**代理度量
 - 连接服务, **请参见**连接服务度量
 - 使用 imqcmd metrics 194
 - 使用代理日志文件 190
 - 使用基于消息的监视 API 197
 - 物理目标, **请参见**物理目标度量
 - 度量消息
 - 关于 88, 196
 - 类型 88, 197
 - 内容 88
 - 端口映射器
 - 端口分配 258
 - 关于 74
 - 队列
 - 添加受管理对象 172
 - 自动创建 280, 286
 - 对象存储库
 - 关于 157
 - LDAP 服务器 158
 - LDAP 服务器属性 158
 - 位置 332, 333, 334
 - 文件系统存储 159
 - 文件系统存储属性 159
 - 对象存储库的位置 158, 159
- ## E
- endpointExceptionRedeliveryAttempts 激活规范属性 319, 320
 - /etc/hosts 文件 (Linux) 180
- ## F
- 防火墙 342
 - 访问规则 140
 - 访问控制文件
 - 版本 139
 - 访问规则 140
 - 格式 139
 - 使用 139
 - 位置 332, 333, 334
 - 分布式事务

G

- XA 资源管理器 113
- 覆盖
 - 消息标题 166
 - 在命令行中 68
- 负荷平衡的队列传送
 - 属性 287
 - 性能调整 221
- 负荷平衡队列传送
 - 属性 119, 302
- 服务 (Windows)
 - Java 运行时 67
 - 启动参数 67
 - 启动疑难解答 67
 - 删除代理 68
 - 运行代理作为 66
 - 重新配置 67
- 服务类型
 - ADMIN 73
 - NORMAL 73
- 服务器故障和安全连接 364

G

- guest 用户 131
- 更新
 - 代理 103
 - 连接服务 109, 112, 265
- 工具, 管理, **请参见** 管理工具
- 关闭代理 105, 262
 - 作为 Windows 服务 68
- 关键字对
 - 生成 147
 - 重新生成 147
- 管理工具 36
 - 管理控制台 38
 - 命令行实用程序 36
- 管理控制台
 - 快速入门 39
 - 启动 40
- 管理任务
 - 开发环境 34
 - 生产环境 34
 - 管理员密码 135

H

- hosts 文件 (Linux) 180
- HTTP
 - 代理 342
 - 连接服务, **请参见** httpjms 连接服务
 - 支持体系结构 342
 - 传输驱动程序 342
- HTTP 连接
 - 多个代理 346
 - 请求时间间隔 345
 - 隧道 Servlet, **请参见** HTTP 隧道 Servlet
 - 支持 342
- HTTP 隧道 Servlet
 - 部署 343
 - 关于 342
- httpjms 连接服务
 - 关于 73, 107
 - 配置 344
 - 设置 343
- HTTPS
 - 连接服务, **请参见** httpsjms 连接服务
 - 支持体系结构 342
- HTTPS 连接
 - 多个代理 357
 - 请求时间间隔 355
 - 隧道 Servlet, **请参见** HTTPS 隧道 Servlet
 - 支持 342
- HTTPS 隧道 Servlet
 - 部署 353
 - 关于 342
- httpsjms 连接服务
 - 关于 73, 107
 - 配置 355
 - 设置 352
- 环境变量, **请参见** 目录变量

恢复

代理 104, 105, 262

连接服务 110, 111, 265

物理目标 123

回送地址 180

I

imq.accesscontrol.enabled 属性 86, 280, 291

imq.accesscontrol.file.filename 属性 86, 280, 292

imq.audit.enabled 属性 280, 292

imq.authentication.basic.user_repository 属性 85, 280, 292

imq.authentication.client.response.timeout 属性 86, 280, 292

imq.authentication.type 属性 85, 280, 292

imq.autocreate.destination.isLocalOnly 属性 280, 286

imq.autocreate.destination.limitBehavior 属性 280, 286

imq.autocreate.destination.maxBytesPerMsg 属性 280, 286

imq.autocreate.destination.maxCount 属性 280, 286

imq.autocreate.destination.maxNumMsgs 属性 286

imq.autocreate.destination.maxNumProducers 属性 280, 286

imq.autocreate.destination.maxTotalMsgBytes 属性 280, 286

imq.autocreate.destination.useDMQ 属性 126, 280

imq.autocreate.queue 属性 103, 280, 286

imq.autocreate.queue.consumerFlowLimit 属性 280, 287

imq.autocreate.queue.localDeliveryPreferred 属性 280, 287

imq.autocreate.queue.maxNumActiveConsumers 属性 103, 280, 287

imq.autocreate.queue.maxNumBackupConsumers 属性 103, 280, 287

imq.autocreate.topic 属性 103, 280, 287

imq.cluster.brokerlist 属性 178, 179, 180, 181, 298

imq.cluster.masterbroker 属性 178, 181, 182, 298

imq.cluster.port 属性 178, 299

imq.cluster.property_name 属性 280

imq.cluster.transport 属性 178, 180, 181, 299

imq.cluster.url 属性 103, 178, 179, 180, 182, 298

imq.destination.DMQ.truncateBody 属性 79, 103, 280, 285

imq.destination.logDeadMsgs 属性 90, 103, 280, 295

imq.hostname 属性 75, 280, 283

imq.httpjms.http.connectionTimeout 属性 345

imq.httpjms.http.property_name 属性 280

imq.httpjms.http.pullPeriod 属性 345

imq.httpjms.http.servletHost 属性 345

imq.httpjms.http.servletPort 属性 345

imq.httpsjms.https.connectionTimeout 属性 356

imq.httpsjms.https.property_name 属性 280

imq.httpsjms.https.pullPeriod 属性 355

imq.httpsjms.https.servletHost 属性 355

imq.httpsjms.https.servletPort 属性 355

imq.imqcmd.password 属性 280, 292

imq.keystore.file.dirpath 属性 147, 295

imq.keystore.file.name 属性 147, 295

imq.keystore.password 属性 147, 154, 295

imq.keystore.property_name 属性 86

imq.keystore.property_name 属性 281, 292

imq.log.console.output 属性 89, 281, 295

imq.log.console.stream 属性 89, 281, 296

imq.log.file.dirpath 属性 89, 281, 296

imq.log.file.filename 属性 89, 296

imq.log.file.name 属性 281

imq.log.file.output 属性 89, 281, 296

imq.log.file.rolloverbytes 属性 89, 104, 281, 296

imq.log.file.rolloversecs 属性 89, 104, 281, 296

imq.log.level 属性 89, 103, 281, 296

imq.log.syslog.facility 属性 89, 281, 296

imq.log.syslog.identity 属性 89, 281, 296

imq.log.syslog.logconsole 属性 89, 281, 296

- imq.log.syslog.logpid 属性 89, 281, 297
- imq.log.syslog.output 属性 89, 281, 297
- imq.log.timezone 属性 89, 281, 297
- imq.message.expiration.interval 属性 79, 281, 285
- imq.message.max_size 属性 79, 104, 281, 285
- imq.metrics.enabled 属性 89, 281, 297
- imq.metrics.interval 属性 89, 281, 297
- imq.metrics.topic.enabled 属性 89, 281, 297
- imq.metrics.topic.interval 属性 90, 281, 297
- imq.metrics.topic.persist 属性 90, 281, 297
- imq.metrics.topic.timetolive 属性 90, 281, 297
- imq.passfile.dirpath 属性 86, 281, 292
- imq.passfile.enabled 属性 86, 281, 292
- imq.passfile.name 属性 86, 281, 292
- imq.persist.file.destination.message.filepool.limit 属性 81, 82, 281, 288
- imq.persist.file.message.cleanup 属性 81, 82, 282, 288
- imq.persist.file.message.filepool.cleanratio 属性 82, 282, 288
- imq.persist.file.message.max_record_size 属性 82, 282, 288
- imq.persist.file.message.vrfile.max_record_size 属性 81
- imq.persist.file.sync 属性 93
- imq.persist.file.sync.enabled 属性 81, 82, 282, 288
Sun Cluster 要求 288
- imq.persist.jdbc.brokerid 属性 82, 96, 289
- imq.persist.jdbc.closedburl 属性 82, 96, 289
- imq.persist.jdbc.createdburl 属性 82, 96, 289
- imq.persist.jdbc.driver 属性 82, 96, 289
- imq.persist.jdbc.needpassword 属性 83, 96, 290
- imq.persist.jdbc.opendburl 属性 82, 96, 289
- imq.persist.jdbc.password 属性 83, 96, 154, 290
- imq.persist.jdbc.property_name 属性 282
- imq.persist.jdbc.table.IMQCCREC35 属性 83, 96, 290
- imq.persist.jdbc.table.IMQDEST35 属性 83, 97, 290
- imq.persist.jdbc.table.IMQINT35 属性 83, 97, 290
- imq.persist.jdbc.table.IMQLIST35 属性 83, 97, 291
- imq.persist.jdbc.table.IMQMSG35 属性 83, 97, 290
- imq.persist.jdbc.table.IMQPROPS35 属性 83, 97, 291
- imq.persist.jdbc.table.IMQSV35 属性 83, 96, 290
- imq.persist.jdbc.table.IMQTACK35 属性 83, 97, 291
- imq.persist.jdbc.table.IMQTXN35 属性 83, 97, 291
- imq.persist.jdbc.user 属性 82, 96, 289
- imq.persist.store 属性 82, 96, 282, 287, 289
- imq.ping.interval 属性 282, 283
- imq.portmapper.backlog 属性 76, 282, 283
- imq.portmapper.hostname 属性 75, 282, 283
- imq.portmapper.port 属性 75, 104, 282, 283
- imq.protocol.protocol_type inbufsz 217
- imq.protocol.protocol_type nodelay 217
- imq.protocol.protocol_type outbufsz 217
- imq.resource_state.count 属性 79, 282, 285
- imq.resource_state.threshold 属性 79, 282, 285
- imq.service.activelist 属性 75, 282, 283
- imq.service_name.accesscontrol.enabled 属性 86, 282, 292
- imq.service_name.accesscontrol.file.filename 属性 86, 282, 293
- imq.service_name.authentication.type 属性 85, 282, 293
- imq.service_name.max_threads 属性 76, 282, 284
- imq.service_name.min_threads 属性 76, 282, 284
- imq.service_name.protocol_type.hostname 属性 76, 178, 282, 283, 298
- imq.service_name.protocol_type.port 属性 76, 282, 283
- imq.service_name.threadpool_model 属性 76, 282, 284
- imq.shared.connectionMonitor_limit 属性 76, 282, 284
- imq.system.max_count 属性 79, 104, 282, 285
- imq.system.max_size 属性 79, 104, 282, 285
- imq.transaction.autorollback 属性 78, 79, 115, 282, 285
- imq.user_repository.ldap.base 属性 137, 293
- imq.user_repository.ldap.gidattr 属性 137, 293
- imq.user_repository.ldap.grpbase 属性 137, 293
- imq.user_repository.ldap.grpfiltler 属性 137, 293

imq.user_repository.ldap.grpsearch 属性 137, 293
imq.user_repository.ldap.memattr 属性 137, 293
imq.user_repository.ldap.password 属性 137, 154, 293
imq.user_repository.ldap.principal 属性 137, 294
imq.user_repository.ldap.property_name 属性 282, 294
imq.user_repository.ldap.server 属性 137, 294
imq.user_repository.ldap.ssl.enabled 属性 137, 294
imq.user_repository.ldap.timeout 属性 137, 294
imq.user_repository.ldap.uidattr 属性 137, 294
imq.user_repository.ldap.usrfilter 属性 137, 294
IMQ_HOME 目录变量 25
IMQ_JAVAHOME 目录变量 26
IMQ_VARHOME 目录变量 25
imqAckOnProduce 属性 311
imqAckTimeout 属性 164, 311
imqAddressList 属性 306
imqAddressListBehavior 属性 306
imqAddressListIterations 属性 306
imqbrokerd 命令 64
 备份配置更改记录 182
 参考 255
 从群集中删除代理 181
 关于 37
 恢复配置更改记录 182
 将参数传递给 92
 连接代理 179
 配置文件 (Solaris、Linux) 66, 69
 清除数据存储库 93, 123
 删除代理 69
 设置日志记录属性 188
 向群集中添加代理 180, 181
 选项 256
 语法 255
 在密码文件中 153
imqbrokerd.conf 文件 66, 69
imqcmd 命令
 安全连接到代理 149, 268
 参考 260
 长期订阅子命令 112
 度量监视 192
 关于 37
 取决于主代理 183
 事务管理 113
 物理目标管理 117
 物理目标子命令 (表) 118
 选项 266
 语法 260
 在密码文件中 153
imqConfiguredClientID 属性 164, 309
imqConnectionFlowCount 属性 164, 311
imqConnectionFlowLimit 属性 164, 311
imqConnectionFlowLimitEnabled 属性 164, 311
imqConsumerFlowLimit 属性 165, 312
imqConsumerFlowThreshold 属性 165, 312
imqdbmgr 命令
 参考 271
 关于 37
 选项 272
 语法 271
 在密码文件中 153
 子命令 272
imqDefaultPassword 属性 164, 309
imqDefaultUsername 属性 164, 309
imqDestinationDescription 属性 305
imqDestinationName 属性 305
imqDisableSetClientID 属性 164, 310
imqFlowControlLimit 属性 165, 312
imqJMSDeliveryMode 属性 166, 310
imqJMSExpiration 属性 166, 310
imqJMSPriority 属性 166, 310
imqkeytool 命令
 参考 277
 关于 37
 命令语法 146, 352
 使用 146, 352
imqLoadMaxToServerSession 属性 165, 312
imqobjmgr 命令
 参考 269

J

- 关于 37
- 选项 270
- 语法 269
- 子命令 269
- imqOverrideJMSDeliveryMode 属性 166, 310
- imqOverrideJMSExpiration 属性 166, 310
- imqOverrideJMSHeadersToTemporaryDestinations 属性 166, 310
- imqOverrideJMSPriority 属性 166, 310
- imqQueueBrowserMax MessagesPerRetrieve 属性 165, 312
- imqQueueBrowserRetrieveTimeout 属性 165, 312
- imqReconnectAttempts 属性 307
- imqReconnectEnabled 属性 306
- imqReconnectInterval 属性 307
- imqSetJMSXAppID 属性 165, 313
- imqSetJMSXConsumerTXID 属性 165, 313
- imqSetJMSXProducerTXID 属性 165, 313
- imqSetJMSXRcvTimestamp 属性 165, 313
- imqSetJMSXUserID 属性 165, 313
- imqSSLIsHostTrusted 属性 307
- imqsvcadmin 命令
 - 参考 275
 - 关于 37
 - 选项 275
 - 语法 275
 - 子命令 275
- imqusermgr 命令
 - 参考 273
 - 关于 37
 - 密码 133
 - 使用 131
 - 选项 273, 274
 - 用户名 133
 - 语法 273
 - 子命令 273
- install.properties 文件 90

J

- J2EE 连接器体系结构 (JCA) 315, 318
- Java 虚拟机, **请参见** JVM
- Java 运行时 255
 - 对于 Windows 服务 67
- java.naming.factory.initial 属性 158, 159
- java.naming.provider.url 属性 158, 159
- java.naming.security.authentication 属性 159
- java.naming.security.credentials 属性 158
- java.naming.security.principal 属性 158
- javahome 选项 67, 255
- JCA (J2EE 连接器体系结构) 315, 318
- JDBC 支持
 - 关于 82
 - 配置 93
 - 驱动程序 82, 93, 96, 289
 - 设置 94
- JDK
 - 指定路径 257, 267, 270, 276
- 记录程序
 - 种类 187
 - 度量信息 89, 297
 - 更改配置 188
 - 关于 87
 - 级别 89, 187, 257, 296
 - 设置属性 188
 - 输出通道 87, 187, 189
 - 消息格式 188
 - 写入到控制台 89, 259, 295
 - 重定向日志消息 190
 - 转移条件 190
 - 作为代理组件 73
- JMS 规范 28
- jms 连接服务 73, 107
- JMSDeliveryMode 消息标题字段 166
- JMSExpiration 消息标题字段 166
- JMSPriority 消息标题字段 166
- JNDI
 - 查找 51, 167
 - 查找名称 167, 171

- 初始上下文 158, 159
- 地址（提供者 URL） 158, 159
- 对象存储库 37, 157
- 对象存储库属性 158, 168
- jrehome 选项 67
- JVM
 - 度量, **请参见** JVM 度量
 - 性能调整 216
 - 性能影响 212
- JVM 度量
 - 度量数量 321
 - 使用 imqcmd metrics 193
 - 使用代理日志文件 191
 - 使用基于消息的监视 197
- 基于 SSL 的连接服务
 - 启动 148
 - 设置 144, 145
- 基于文件的持久性 81
 - 另请参见**持久性管理器, 基于文件的持久性
- 基准检验, 性能 202
- 加密
 - 关于 85
 - 基于 SSL 的服务, 和 144
 - 密钥工具, 和 85
- 简单网络时间协议 64
- 监视, **请参见**性能监视
- 教程 39

K

- 开发环境管理任务 34
- 客户机
 - 启动 68
 - 时钟同步 64
- 客户机标识符 (ClientID) 163
 - 销毁长期订阅 113
- 客户机应用程序
 - 示例 28, 332, 333, 335
 - 影响性能的因素 206

- 客户机运行时
 - 配置 216
 - 消息流调整 222
- 可靠传送 164
 - 性能折衷 206
- 控制消息 76

L

- LDAP 服务器
 - 对象存储库属性 158
 - 验证故障转移 136
 - 用户系统信息库访问 136
 - 作为用户系统信息库 135
- 连接
 - 查询 112, 265
 - 服务器或代理故障 364
 - 故障转移, **请参见**自动重新连接
 - 列出 111, 265
 - 受文件描述符限制的限制 64
 - 性能影响 213
 - 自动重新连接, **请参见**自动重新连接
- 连接代理 179
- 连接服务
 - admin 74, 107
 - 查询 108, 112, 265
 - 度量数据, **请参见**连接服务度量
 - 端口映射器, **请参见**端口映射器
 - 访问控制 86, 291
 - 服务类型 73
 - 更新 109, 112, 265
 - 关于 73
 - HTTP, **请参见** HTTP 连接
 - httpjms 73, 107
 - HTTPS, **请参见** HTTPS 连接
 - httpsjms 73, 107
 - 恢复 110, 111, 265
 - jms 73, 107
 - 基于 SSL 147
 - 连接类型 73
 - 启动时激活 283

M

- 群集 145, 180
 - ssladmin, [请参见 ssladmin 连接服务](#)
 - ssljms, [请参见 ssljms 连接服务](#)
 - 属性 109, 283
 - 线程池管理器 75
 - 线程分配 109
 - 显示属性 108
 - 相关命令 264
 - 暂停 110, 265
 - 连接服务度量
 - 度量数量 324
 - 使用 `imqcmd metrics` 109, 195
 - 使用 `imqcmd query` 196
 - 连接工厂受管理对象
 - 队列浏览器行为属性 165, 312
 - 覆盖消息标题字段 166
 - JMS 属性支持属性 165, 313
 - 客户机身份验证属性 163
 - 可靠性和流控制属性 164
 - 连接处理属性 161
 - 属性 160
 - 添加 170
 - 应用服务器支持属性 165, 312
 - 临时物理目标 120
 - 流控制, [请参见消息流控制](#)
 - 路由, [请参见消息路由器](#)
- ## M
- ManagedConnectionFactory JavaBean 317
 - MDB, [请参见消息驱动 Bean](#)
 - messageSelector 激活规范属性 319
 - 密码
 - 编码 292
 - 管理员 135
 - JDBC 154
 - LDAP 154
 - 密码文件, [请参见密码文件](#)
 - 命名约定 133
 - 默认 164, 309
 - SSL 密钥库 147, 154, 258
 - 密码文件
 - 代理配置属性 86, 292
 - 命令行选项 258
 - 使用 153
 - 位置 154, 332, 333, 334
 - 密码文件, [请参见密码文件](#)
 - 密钥工具 85
 - 密钥库
 - 文件 147, 295, 353
 - 属性 295
 - 命令文件 168
 - 命令行实用程序
 - 帮助 255
 - 关于 36
 - 基本语法 254
 - `imqbrokerd`, [请参见](#), `imqbrokerd` 命令
 - `imqcmd`, [请参见](#), `imqcmd` 命令
 - `imqdbmgr` [请参见](#), `imqdbmgr` 命令
 - `imqkeytool`, [请参见](#), `imqkeytool` 命令
 - `imqobjmgr`, [请参见](#), `imqobjmgr` 命令
 - `imqsvcadmin`, [请参见](#), `imqsvcadmin` 命令
 - `imqusermgr`, [请参见](#), `imqusermgr` 命令
 - 通用选项 255
 - 显示版本 255
 - 命令行语法 254
 - 命令选项 255
 - 作为配置覆盖 68
 - 目标度量
 - 度量数量 325
 - 使用 `imqcmd metrics` 193, 195, 263
 - 使用 `imqcmd query` 196
 - 使用基于消息的监视 197
 - 目标受管理对象
 - 属性 166
 - 目录变量
 - IMQ_HOME 25
 - IMQ_JAVAHOME 26
 - IMQ_VARHOME 25

N

- NORMAL 服务类型 73
- nsswitch.conf 文件 (Linux) 180
- 内存管理
 - 对于代理 78
 - 使用物理目标属性 119
 - 性能调整 220
 - 阈值 78
- 内置的持久性 81

O

- Oracle 94, 98

P

- password 受管理连接工厂属性 318
- password 资源适配器属性 316
- PointBase 94
- 属性
 - 物理目标, **请参见**物理目标, 属性
- 配置更改记录 182
 - 备份 182
 - 恢复 182
- 配置文件 90
 - 安装 90
 - 编辑 92
 - 代理 (图) 91
 - 模板 331, 333, 334
 - 模板位置 331, 333, 334
 - 默认 90
 - 群集 178, 179, 298
 - 实例 90, 178, 331, 333, 334
 - 位置 331, 333, 334
- 瓶颈, 性能 206

Q

- 启动
 - 基于 SSL 的连接服务 148
 - 客户机 68
- 清除, 来自物理目标的消息 123
- 权限
 - 访问控制属性文件 84, 139
 - 管理服务 85
 - 计算 140
 - Message Queue 操作 84
 - 密码文件 154
 - 密钥库 352
 - 嵌入式数据库 95
 - 数据存储库 81
 - 用户系统信息库 131, 273
- 确认
 - 客户机 77
 - 事务和 78
 - 传送, 对象 77
- 群集连接服务 145, 180
 - 端口号 178, 299
 - 网络传输 178, 179, 299
 - 主机名或 IP 地址 178, 298
- 群集目录查找 (Linux) 180
- 群集配置属性 178, 298
- 群集配置文件 178, 179, 298
- 群集, 请参见代理群集

R

- reconnectAttempts 受管理连接工厂属性 318
- reconnectAttempts 资源适配器属性 316, 317
- reconnectEnabled 受管理连接工厂属性 318
- reconnectEnabled 资源适配器属性 316, 317
- reconnectInterval 受管理连接工厂属性 318
- reconnectInterval 资源适配器属性 317
- reset messages 选项 123
- ResourceAdapter JavaBean 315
- RESTART 属性 66

日志记录, [请参见](#)记录程序

日志文件

默认位置 [332, 333, 334](#)

转移标准 [89, 296](#)

S

sendUndeliverableMsgsToDMQ 激活规范属性 [320](#)

SNTP [64](#)

SSL

关于 [85](#)

加密, 和 [144](#)

连接服务, [请参见](#)基于 SSL 的连接服务

启用 [148](#)

通过 TCP/IP [145](#)

ssladmin 连接服务

关于 [74, 107](#)

设置 [145](#)

ssljms 连接服务

关于 [73, 107](#)

设置 [145](#)

subscriptionDurability 激活规范属性 [319, 320](#)

subscriptionName 激活规范属性 [320](#)

Sun Cluster

配置 [288](#)

同步属性和 [81](#)

syslog [87, 189](#)

删除

代理 [69](#)

代理实例 [69](#)

物理目标 [124](#)

删除目标 [124](#)

审计日志 [154](#)

生产方

目标限制于 [286, 302](#)

物理目标限制 [119](#)

生产环境

管理任务 [34](#)

设置 [34](#)

维护 [35](#)

时间同步服务 [64](#)

实例目录

和基于文件的数据存储库 [93](#)

和实例配置文件 [136](#)

删除 [69](#)

实例配置文件, [请参见](#)配置文件

示例应用程序 [28, 332, 333, 335](#)

事务

管理 [113](#)

回滚 [113, 266](#)

确认和 [78](#)

提交 [114, 115, 266](#)

信息 [266](#)

性能影响 [208](#)

使用帮助 [255](#)

时钟同步 [64](#)

受管理对象

查询 [174](#)

查找名称 [270](#)

队列, [请参见](#)队列

对象存储库, [参见](#)对象存储库

更新 [175](#)

列出 [174](#)

删除 [173](#)

属性 (参考) [305](#)

所需的信息 [167](#)

XA 连接工厂, [请参见](#)连接工厂受管理对象

主题, [请参见](#)主题

授权

管理 [138](#)

关于 [84](#)

用户组 [85](#)

[请参见](#)访问控制文件

数据存储库

关于 [80](#)

JDBC 可访问 [82](#)

内容 [93](#)

配置 [93](#)

位置 [331, 333, 334](#)

平面文件 [81](#)

性能影响 [215](#)

压缩 [81](#)

与磁盘同步 93

重置 258

属性

安全性 291

持久性 288

代理监视服务 295

代理实例配置 92

httpjms 连接服务 345

httpsjms 连接服务 355

记录程序 295

连接服务 283

密钥库 295

内存管理 119, 285

群集配置 298

消息路由器 285

与 JDBC 相关 95, 289

语法 92

自动创建 286

隧道 Servlet 连接 364

所有命令的语法 254

T

TCP 73, 107

TimeToLive 功能

时钟同步和 64

TLS 73, 107

停用消息

日志记录 90

另请参见停用消息队列

停用消息队列

maxNumMsgs 值 127

maxTotalMsgBytes 值 127

配置 126

日志记录 90, 128

限制行为 127

消息截断 79

停用消息队列中的截断 79

同步

内存到磁盘 81, 93

时钟 64

U

ulimit 命令 64

update dst 子命令

限定 122

username 受管理连接工厂属性 318

userName 资源适配器属性 316

W

W32Time 服务 64

Windows 服务, **请参见**服务 (Windows)

文件描述符限制 64

连接限制和 64

文件同步

imq.persist.file.sync.enabled 选项 288

与 Sun Cluster 288

物理目标

创建 119

磁盘占用 124

度量, **请参见**物理目标度量

更新属性 122, 264

管理 117

恢复 123, 264

回收磁盘空间 125

获得有关信息 121, 264

类型 120, 263

列出 120, 263

临时 120

清除消息来自 123, 264

使用停用消息队列 126

属性 301

属性值 121

停用消息队列 126

显示属性值 121

限制范围于群集中 119, 286, 303

- 限制行为 78, 119, 302
- 销毁 124, 263
- 信息 121
- 压缩 124
- 压缩基于文件的数据存储库 125, 263
- 暂停 122, 264
- 传送批量消息 119, 287, 303
- 自动创建 144
- 物理目标属性 301

X

- xntpd 守护程序 64
- 系统时钟同步 64
- 线程池管理器
 - 共享线程 75
 - 关于 75
 - 专用线程 75
- 显示产品版本 255
- 限制行为
 - 代理 78
 - 物理目标 78, 119, 302
- 销毁物理目标 124
- 消息
 - 持久性 78, 80
 - 从物理目标中清除 123, 264
 - 大小, 和性能 210
 - 代理限制 79, 104, 282, 285
 - 度量 88
 - 度量消息, **请参见**度量消息
 - 过期回收 79, 285
 - 可靠传送 164
 - 流控制, **请参见**消息流控制
 - 路由和传送 76
 - 目标限制于 301
 - 确认 77
 - 碎片 81
 - 吞吐量性能 202
 - 物理目标限制 119
 - 延迟 202
 - 暂停流 122
 - 重新传送 77
 - 主体类型和性能 211
- 消息标题覆盖 166
- 消息服务器体系结构 215
- 消息服务性能 212
- 消息流控制
 - 测量 222
 - 代理 78, 119
 - 属性 164
 - 限制 222
 - 性能调整 222
 - 性能影响 216
- 消息路由器
 - 关于 76
 - 属性 285
 - 作为代理组件 73
- 消息驱动 Bean
 - 资源适配器配置 315, 318
- 消息碎片 81
- 写入操作 (适用于基于文件的存储) 93
- 协议, **请参见**传输协议
- 协议类型
 - HTTP 73, 107
 - TCP 73, 107
 - TLS 73, 107
- 性能
 - 调整, **请参见**性能调整
 - 关于 201
 - 衡量 202
 - 基线模式 203
 - 基准检验 202
 - 监视, **请参见**性能监视
 - 可靠性折衷 206
 - 瓶颈 206
 - 疑难解答 225
 - 影响因素, **请参见**性能因素
 - 优化, **请参见**性能调整
 - 指示器 202
- 性能调整
 - 代理调整 220
 - 过程概述 201

- 客户机运行时调整 221
- 系统调整 216
- 性能监视
 - 度量数据, [请参见](#)度量数据
 - 工具, [请参见](#)度量监视工具 185
- 性能因素
 - 操作系统 212
 - 长期订阅 209
 - 代理限制行为 215
 - JVM 212
 - 连接 213
 - 确认模式 209
 - 事务 208
 - 数据存储库 215
 - 文件同步 288
 - 消息大小 210
 - 消息服务器体系结构 215
 - 消息流控制 216
 - 消息主体类型 211
 - 选择器 210
 - 硬件 212
 - 传输协议 213
 - 传送模式 207
- 许可证
 - 启动选项 257
- 选择器
 - 关于 210
 - 性能影响 210

Y

- 压缩
 - 基于文件的数据存储库 81
 - 物理目标 124
- 验证
 - 管理 130
 - 关于 83
- 疑难解答 225
 - Windows 服务启动 67
- 硬件, 性能影响 212

- 应用程序, [请参见](#)客户机应用程序
- 用户名 164, 309
 - 格式 133
 - 默认 131
- 用户系统信息库
 - 初始条目 131
 - 管理 134
 - 关于 83
 - LDAP 135
 - LDAP 服务器 136
 - 平台相关性 131, 273
 - 属性 85
 - 填充 134
 - 位置 332, 333, 334
 - 平面文件 130
 - 用户状态 133
 - 用户组 133
- 用户组 133
 - 关于 84
 - 默认 85
 - 删除指定 133
 - 预定义 133
- 优先级 (配置属性的) 91
- 阈值
 - 内存 78

Z

- 暂停
 - 代理 104, 105, 262
 - 连接服务 110, 265
 - 物理目标 122, 264
- 占用率 125
- 证书 146, 352
- 主代理
 - 不可用 183
 - 配置更改记录 182
 - 指定 178, 179
- 主题
 - 添加受管理对象 172

Z

- 自动创建 [280, 286](#)
- 自动创建物理目标
 - 访问控制 [144](#)
 - 禁用 [35](#)
 - 配置 [85](#)
 - 属性（表） [286](#)
- 自动重新连接功能
 - 属性 [162](#)
- 自签名证书 [146, 352](#)
- 资源适配器 [315](#)
 - 重新连接 [316, 317, 318](#)