WebSphere MQ internet pass-thru V1.3

S152-0311-01

注意！

在使用此信息及其支持的产品之前，请确保阅读第 165 页的『声明』中的一般信息。

第四版（2003 年 3 月）

本版本适用于 WebSphere MQ internet pass-thru V1.3（程序号 5639-L92）及所有后续发行版和修订版，直到在新版本中另有声明为止。

目录

图

前言

什么是 internet pass-thru?

WebSphere MQ internet pass-thru 以前称为 MQSeries internet pass-thru。从现在开始本书中将把 MQSeries 称为 WebSphere MQ。请注意，不是所有的 MQSeries 手册都将立刻更名为 WebSphere MQ，注意有时会同时提到 MQSeries 和 WebSphere MQ。

IBM® WebSphere MQ internet pass-thru:

- 是一个 WebSphere MQ 基本产品扩展，它可用于实现因特网上远程站点间消息传递的解决方案
- 通过对 HTTP 内部协议进行隧道处理或充当代理，使 WebSphere MQ 通道协议进出防火墙更为简单且更易于管理
- 作为可以接收和转发 WebSphere MQ 消息流的独立服务运行。它运行所在的系统不一定要主管 WebSphere MQ 队列管理器
- 使用 WebSphere MQ 帮助您提供商家对商家交易
- 使现有未更改的 WebSphere MQ 应用程序能通过防火墙使用
- 对多个队列管理器的访问提供单点控制
- 允许所有数据加密
- 记录所有连接尝试

为了方便起见，本书中通常将 WebSphere MQ internet pass-thru 称为 “MQIPT”。

本书针对的读者

本书是为系统设计者、WebSphere MQ 技术管理员，以及防火墙和网络管理员编写的。

理解本文档所需了解的内容

您需要充分理解:

- WebSphere MQ 队列管理器和消息通道的管理，如*《WebSphere MQ 系统管理指南》*和 *WebSphere MQ Intercommunication* 中所描述的
- 防火墙的实现方法
- 因特网协议路由选择 / 联网
- 用于负载平衡和增强可用性的 IBM Network Dispatcher
- IBM WebSphere® Application Server

先决条件

本发行版的 internet pass-thru 在下列平台上运行:

- Windows NT® V4.0 带 Service pack 6
- Windows® 2000
- Windows XP
- Sun Solaris

- AIX® V5.1
- HP-UX 11
- Linux

MQIPT 服务器需要 J2SE V1.4.0 运行时（JRE）。需要 SDK V1.4.0 以创建安全性出口。

唯一支持的网络协议是 TCP/IP。

管理客户机帮助需要使用 Netscape 浏览器。

可访问性信息

管理客户机 GUI 构建时充分考虑了可访问性。在该 GUI 中，您可以使用键盘上的同等功能键来直接执行所有可用的功能，而无须使用鼠标。您可以通过按标准方式使用 tab、shift-tab、ctrl-tab 和光标键来浏览屏幕。通过先选择按钮然后按 Enter 键可与鼠标按键达到同样的效果。

可以通过 tab 和光标键的组合，或者使用可用于所有选项的加速键来到达菜单选项。例如，可以通过首先选择 alt-f，然后选择 alt-q（“文件”-> “退出”）来关闭 GUI。一旦到达菜单项，就可以使用 Enter 键激活该菜单项。

您可以使用光标键浏览树。特别是，右光标键和左光标键可用于打开或关闭 MQIPT 节点，从而显示或隐藏路由。

可以使用空格键来更改所选复选框的状态。可以使用 Enter 键选择字段进行编辑。

感观

理想情况下，GUI 应采用环境的感观。由于这并不是一定发生，因此您可以提供一个配置文件来定制 GUI 的感观以使其适合您的需要。这一名为“custom.properties”的配置文件应放置在 `bin` 子目录中。

使用此配置文件配置下列内容:

- 前景色 – 文本的颜色
- 背景色
- 文本字体
- 文本样式 – 正常体、粗体、斜体或粗斜体

提供了一个样本配置文件“customSample.properties”，其中包含显示如何更改该文件的注释。建议您将此文件复制到 `bin/custom.properties` 并进行必需的更改。

更改总结

本部分描述本修订版的 WebSphere MQ internet pass-thru 中的更改。在更改的左边以竖线标记自前一修订版以来的更改。

本修订版的更改（S152-0311-01）

本版本的 WebSphere MQ internet pass-thru 的增强包括:

- 用于控制客户机连接请求的安全性出口
- 用于 CRL 和 ARL 的 LDAP 支持
- 对密钥环密码的加密
- 在密钥环中选择证书
- 新的 AES 密码套件
- 通用 UNIX® 磁盘映象
- 对路由重新启动操作的控制
- AIX 和 HP-UX 平台现在支持 Java™ 1.4

第三版的更改（S152-0311-00）

本版本的 WebSphere MQ internet pass-thru 中的增强包括以下几个方面:

- 控制外出端口地址分配
- 示例配置
- 改进的 SSL 跟踪
- Java 安全性管理器
- 用于管理 SSL 证书和密钥环文件的 KeyMan 实用程序
- Linux 支持，包含 WebSphere MQ 消息的服务质量
- 在 Windows 平台上可用的 NLS 安装映象
- 属性名现在不区分大小写
- Servlet 版本
- Socks 客户机和服务器支持
- SSL 代理方式
- 支持多主机系统
- 管理客户机的信号灯状态
- WebSphere MQ 群集支持

第二版的更改

本版本的 WebSphere MQ internet pass-thru 中的增强包括以下几个方面:

- 增加 AIX、HP-UX 和 Windows 2000 作为用于 MQIPT 的平台
- 增加 HTTP 代理支持
- 增加安全套接字层（SSL）支持
- MQIPT 通过 SOCKS 代理与另一个外部 MQIPT 或 MQSeries® 服务器进行通信的能力
- 使用管理客户机 GUI 使管理一个或多个 MQIPT 更容易
- 增加对 IBM Network Dispatcher 的支持
- 对跟踪的少许增强
- 对 mqiptAdmin 命令的少许增强

第 1 章 WebSphere MQ internet pass-thru 介绍

WebSphere MQ internet pass-thru 是基本 WebSphere MQ 产品的扩展。MQIPT 作为一个独立的服务运行，它可以在两个 WebSphere MQ 队列管理器之间或者 WebSphere MQ 客户机和 WebSphere MQ 队列管理器之间接收和转发 WebSphere MQ 消息流。MQIPT 使位于不同物理网络中的客户机和服务器能够相互连接。

在两个 WebSphere MQ 队列管理器之间或者 WebSphere MQ 客户机和 WebSphere MQ 队列管理器之间的通信路径中可以放置一个或多个 MQIPT。MQIPT 使两个 WebSphere MQ 系统能够不依靠这两个系统间的直接 TCP/IP 连接而进行消息交换。如果防火墙配置禁止这两个系统间的直接 TCP/IP 连接，那么该属性将十分有用。

MQIPT 在一个或多个 TCP/IP 端口上侦听进入连接，这些连接用于传送普通的 WebSphere MQ 消息、HTTP 中隧道化的 WebSphere MQ 消息或者使用安全套接字层（SSL）加密的 WebSphere MQ 消息。它可处理多个并发连接。

生成初始 TCP/IP 连接请求的 WebSphere MQ 通道称为“调用程序”，它正试图连接的通道称为“响应程序”，最终试图联络的队列管理器称为“目的地队列管理器”。

MQIPT 的预期用途有:

- MQIPT 可用作为通道集中器，这样对于防火墙来说，到达或来自多个独立主机的通道都好象来自或到达此 MQIPT 主机。这使得定义和管理防火墙过滤规则更为方便。

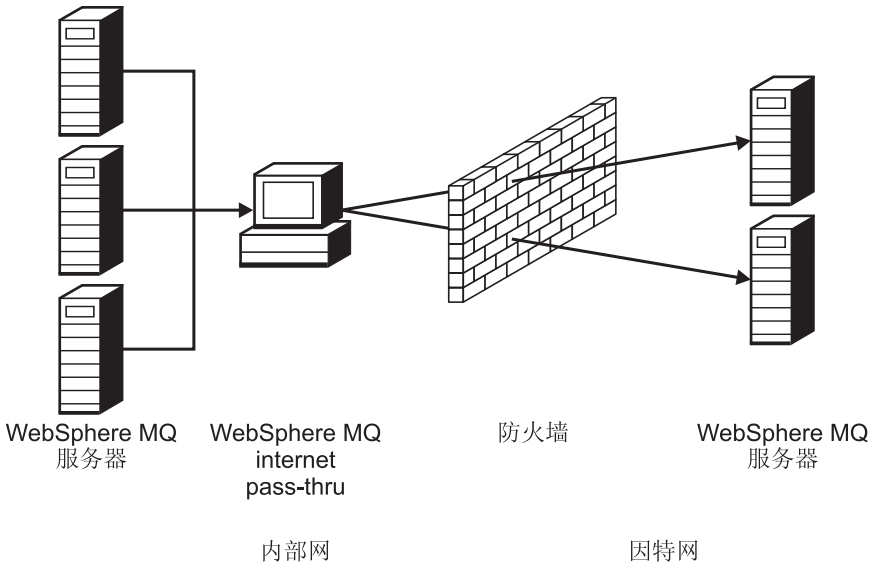

图 1. MQIPT 作为通道集中器的示例

- 如果 MQIPT 位于防火墙的“非保护区”（DMZ）中，在一台具有已知可信网际协议（IP）地址的机器上，MQIPT 可用于侦听进入 WebSphere MQ 通道连接，随后它可以将这些连接转发到可信内部网；内层防火墙必须允许这台可信机器建立入站连

接。在本配置中，MQIPT 不允许外部访问请求看到可信内部网中机器的真实 IP 地址。因此，MQIPT 提供单点访问。

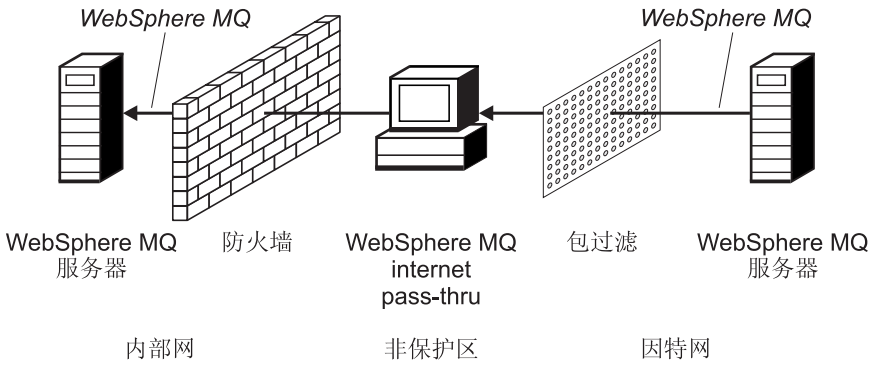

图 2. “非保护区”和 MQIPT 的示例

- 如果两个 MQIPT 都部署为内嵌，则它们可使用 HTTP 或 SSL 通信。HTTP 隧道功能使请求能够穿过防火墙发送（通过使用现有的 HTTP 代理）。第一个 MQIPT 将 WebSphere MQ 协议插入 HTTP，第二个 MQIPT 则从其 HTTP 包装器抽取 WebSphere MQ 协议，并将其转发到目的地队列管理器。

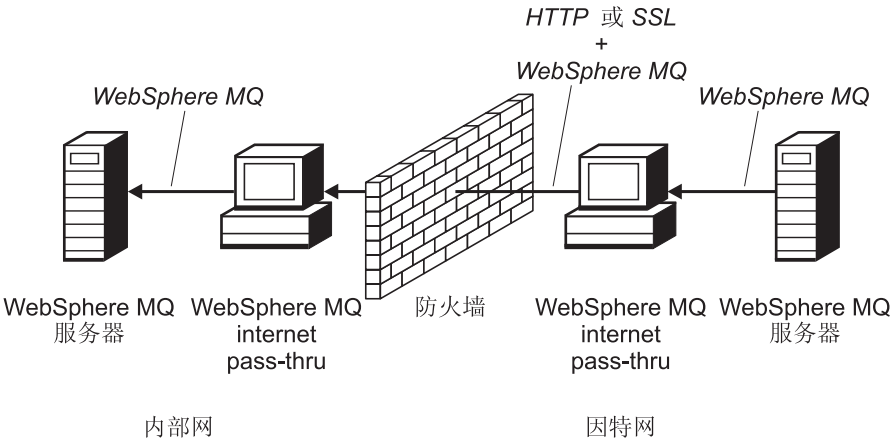

图 3. MQIPT 和 HTTP 隧道的示例

- 类似地，在穿过防火墙传送请求之前，可以先加密这些请求。第一个 MQIPT 加密数据，第二个 MQIPT 则使用 SSL 解密数据，然后将这些数据发送到目的地队列管理器。

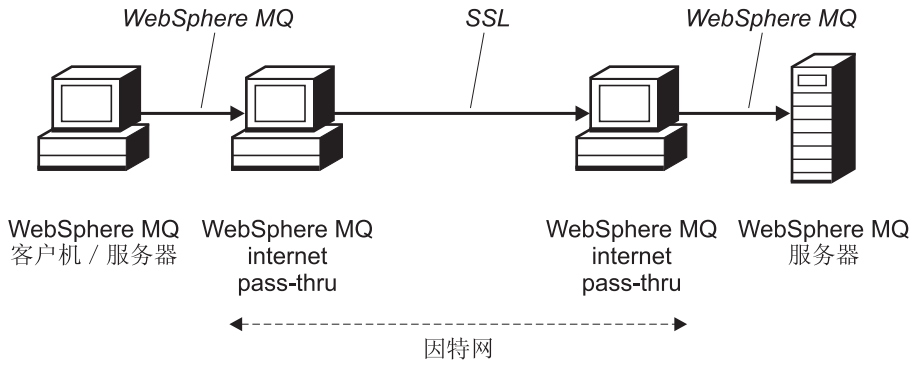

图 4. MQIPT 和 SSL 的示例

当 MQIPT 从源向其目标转发数据时，它在内存中保存数据。磁盘上不保存数据（除由操作系统调页到磁盘的内存外）。MQIPT 唯一明确访问磁盘是在读取其配置文件以及写日志和跟踪记录的时候。

可以通过一个或多个 MQIPT 组成各种 WebSphere MQ 通道类型。通信路径中存在的 MQIPT 对所连接 WebSphere MQ 组件的功能属性没有影响，但这可能对消息传送的性能略有影响。

MQIPT 可与 WebSphere MQ Publish/Subscribe 或 WebSphere MQ Integrator 消息代理结合使用。

第 4 页的图 5 显示了 WebSphere MQ 拓扑结构中所有可能的 MQIPT 配置。在本图中，注意“出站连接”侧的防火墙上的 HTTP 代理、SOCKS 代理 和 MQIPT 机器代表因特网上的多台机器链接在一起的可能性。例如，在到达目标之前，MQIPT 机器可以通过一个或多个 SOCKS 或 HTTP 代理机器或者另外的 MQIPT 机器进行通信。

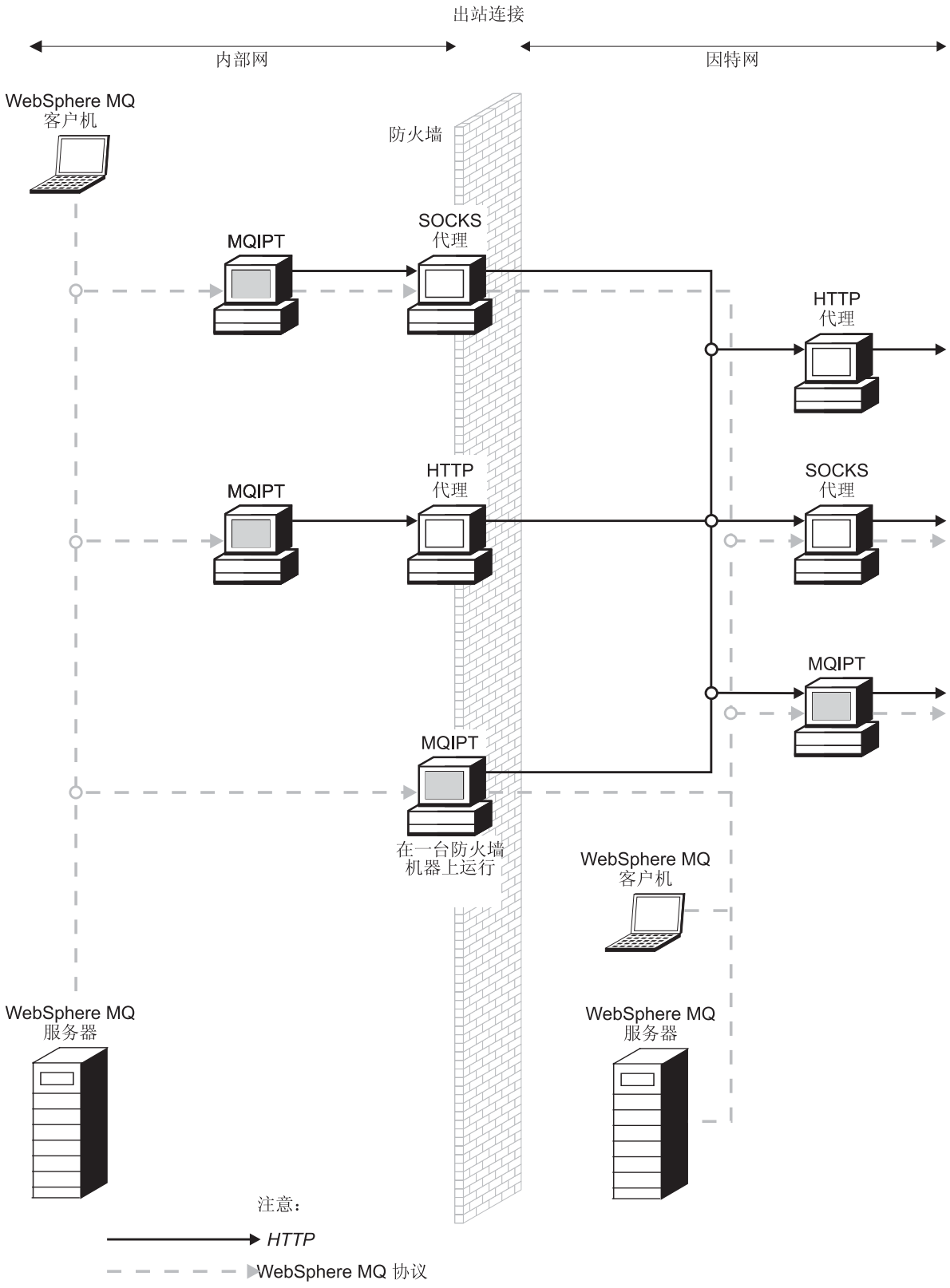

图 5. 显示可能的 MQIPT 配置的 WebSphere MQ 拓扑结构 (1/2)

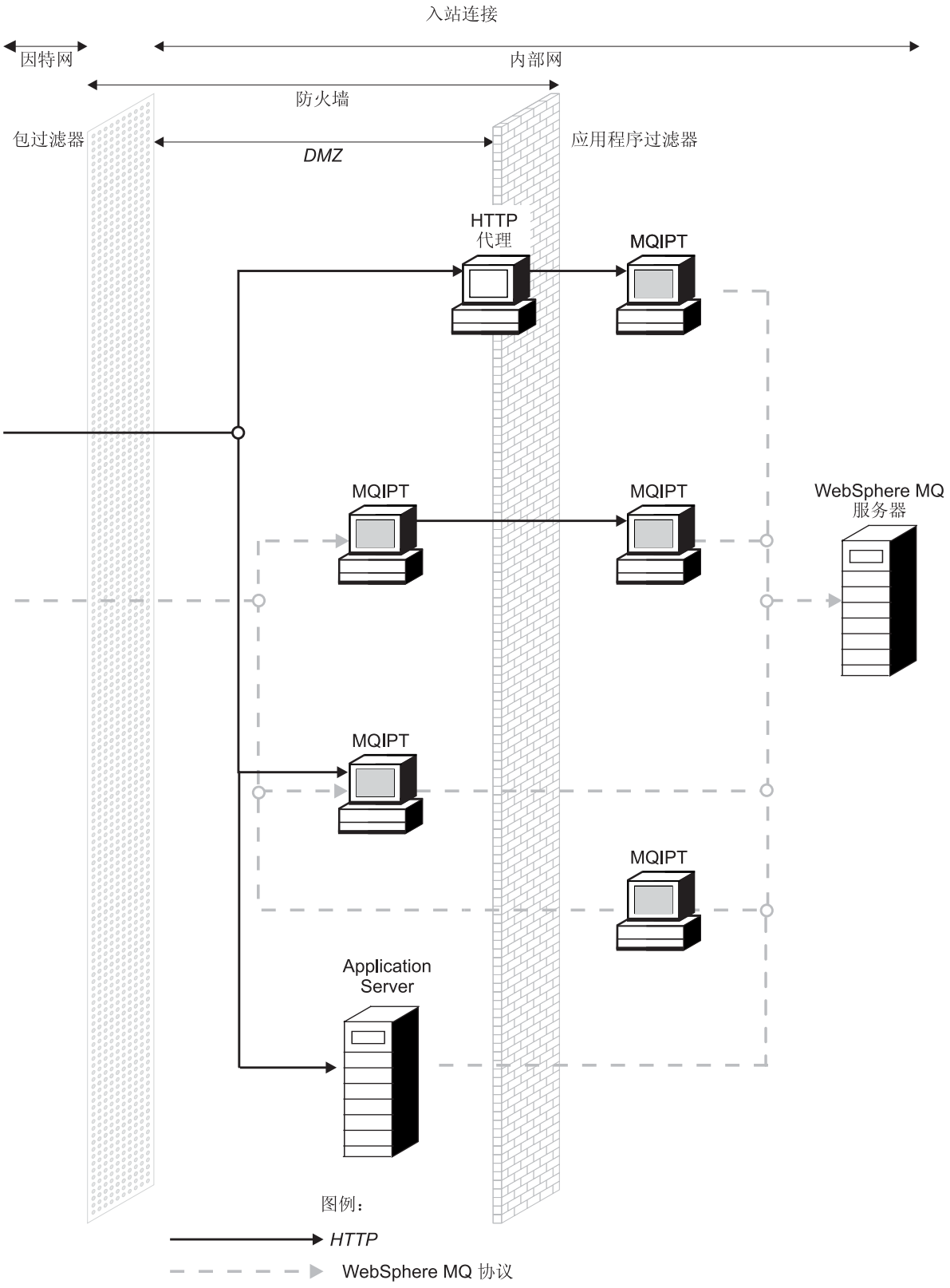

图 5. 显示可能的 MQIPT 配置的 WebSphere MQ 拓扑结构 (2/2)

第 2 章 internet pass-thru 如何工作

本章概述 internet pass-thru 的工作方式。

概述 internet pass-thru 如何工作

在最简单的配置中，MQIPT 充当 WebSphere MQ 协议转发器。它在 TCP/IP 端口上侦听，并接受来自 WebSphere MQ 通道的连接请求。如果接收到一个完好的请求，MQIPT 建立自己与目标 WebSphere MQ 队列管理器间的进一步 TCP/IP 连接。然后它将所有从其进入连接接收到的协议包传递到目的地队列管理器，并在原始进入连接上从目的地队列管理器返回协议包。

不包含对 WebSphere MQ 协议（客户机／服务器或队列管理器到队列管理器）的更改 – 因为两端都不直接知道中介物的存在 – 因此不需要 WebSphere MQ 客户机或服务器代码的新版本。

要使用 MQIPT，必须配置调用程序通道以使用 MQIPT 的主机名和端口，而不是目的地队列管理器的主机名和端口。它使用 WebSphere MQ 通道的 CONNAME 属性进行定义。MQIPT 读取进入数据并简单地将它传递到目的地队列管理器。其它配置字段（如，客户机／服务器通道中的用户标识和密码）类似地传递到目的地队列管理器。

MQIPT 可用于允许访问一个或多个目的地队列管理器。要使其工作，必须要有一个机制能告诉 MQIPT 连接到哪一个队列管理器，因此 MQIPT 使用进入 TCP/IP 端口号来确定连接到哪一个队列管理器，在下一个段落中描述了这一点。

要允许访问多个目的地队列管理器，MQIPT 可以配置为在多个 TCP/IP 端口上侦听。每个侦听端口通过一个 MQIPT “路由” 映射到目的地队列管理器。MQIPT 管理员可以至多定义 100 个这样的路由，这些路由使侦听 TCP/IP 端口与目的地队列管理器的主机名和端口相关联。这意味着原始通道中看不见目的地队列管理器的主机名（IP 地址）。每个路由可以处理其侦听端口与目标间的多个连接，每个连接独立操作。

MQIPT 使用称为 mqipt.conf 的配置文件，该文件包含所有路由和它们的关联属性的定义。请参阅第 69 页的第 19 章，『管理和配置 internet pass-thru』以获取有关此文件的更多信息。

MQIPT 启动时，它将启动配置文件中的每个路由。显示每个路由的状态的消息写到系统控制台。当为路由出现消息 MQCPI078 时，说明该路由用于接受连接请求准备就绪。

支持的通道配置

支持所有 WebSphere MQ 通道类型，但是配置限定于 TCP/IP 连接。对于 WebSphere MQ 客户机或队列管理器，MQIPT 好象是它是目的地队列管理器。在通道配置中需要目标主机和端口号的地方，指定 MQIPT 的主机名和侦听器端口号。

客户机／服务器通道

MQIPT 侦听进入客户机连接请求，然后转发它们（使用 HTTP 隧道、SSL 或者作为标准 WebSphere MQ 协议包）。如果 MQIPT 使用的是 HTTP 隧道或 SSL，它使用到第二个 MQIPT 的连接转发它们。如果不是使用 HTTP 隧道，它使用到看上

去象目的地队列管理器（尽管它可能是按顺序下来的下一个 MQIPT）的连接转发它们。一旦目的地队列管理器接受了客户机连接，将在客户机和服务器之间中继包。

群集发送方 / 接收方通道

如果 MQIPT 接收到来自群集发送方通道的进入请求，它假设队列管理器已经 socks 化且在 SOCKS 握手过程期间会获得真实目标地址。它使用与客户机连接通道完全相同的方式转发请求到下一个 MQIPT 或者目的地队列管理器。这也包含自动定义的群集发送方通道。

发送方 / 接收方

如果 MQIPT 接收到来自发送方通道的进入请求，它使用与客户机连接通道完全相同的方式转发请求到下一个 MQIPT 或者目的地队列管理器。目的地队列管理器验证进入连接，并在适当的情况下启动接收方通道。将中继所有发送方和接收方通道之间的通信（包含安全性流量）。

请求方 / 服务器

该组合的处理方法与上面类型的处理方法相同。由目的地队列管理器上的服务器通道来验证连接请求。

请求方 / 发送方

如果两个队列管理器之间不允许建立直接连接，但是它们都可以连接到 MQIPT 并接受来自它的连接，可使用“回调”配置。

服务器 / 请求方和服务器 / 接收方

MQIPT 象处理发送方 / 接收方配置一样处理它们

第 3 章 HTTP 支持

可以配置 MQIPT，以便把它转发的数据包编码为 HTTP 请求。MQIPT 支持分块或不分块的 HTTP 隧道。

因为现今的 WebSphere MQ 通道不接受 HTTP 请求，所以需要第二个 MQIPT 接收 HTTP 请求，并将它们转换回正常的 WebSphere MQ 协议包。第二个 MQIPT 除去 HTTP 头，以便使进入包在传递到目的地队列管理器之前转换回标准 WebSphere MQ 协议包中。

当使用不分块的 HTTP 隧道时，每个 HTTP 请求的 HTTP 应答将发送回第一个 MQIPT。此应答可以是来自目的地队列管理器响应或伪确认。如果两个 WebSphere MQ 系统的其中一个必须发送一连串连续的 WebSphere MQ 协议包（在传送一条大的消息时会发生这种情况），则使用几个 HTTP 请求／应答对来传送数据。为了这样做，MQIPT 插入附加的请求或应答流。

当使用分块的 HTTP 隧道时，HTTP 头仅包含第一个包。中间和最后的包有分块头。此排列从第二个 MQIPT 除去对伪确认的等待，因此提供的性能比没有分块的 HTTP 隧道所提供的性能稍好一些。

在两个 MQIPT 之间使用 HTTP 时，HTTP 请求和应答流动所在的 TCP/IP 连接是持久的，并且为消息通道的生命期保留为打开。MQIPT 不关闭请求／应答对之间的 TCP/IP 连接。

如果两个 MQIPT 通过 HTTP 通信，HTTP 请求可能会在一段持续时间内停留在未完成的阶段。当服务器端正在等待新消息到达其传输队列时，一个示例在请求者／服务器通道内。WebSphere MQ 通道协议提供一个“心跳”机制，此机制周期性地要求等待结束以将心跳消息发送给它的伙伴（缺省通道心跳周期为 5 分钟），并且 MQIPT 使用此心跳作为 HTTP 应答。不要禁用此通道心跳或将其值设置地过高，以避免引起一些防火墙中的超时问题。

有些 HTTP 代理具有它们自己的属性以用于控制持久连接（例如，对一个持久连接可发出的请求数）。HTTP 代理还必须支持 HTTP 1.1 协议。当使用 IBM WebSphere Caching Proxy 时，应复位下列属性:

- MaxPersistenceRequest 设置为高值（例如，5000）
- PersistentTimeout 设置为高值（例如，12 小时）
- ProxyPersistence 设置为 on

请参阅第 100 页的『HTTP 代理配置』以获取使用 HTTP 的示例。

HTTPS

通过启用发出客户机连接的 MQIPT 上的 HTTPS 和 SSLClient 路由属性，可在 HTTP 连接上使用 HTTPS。MQIPT 必须能访问将用于认证目标 HTTP 代理／服务器的可信 CA 证书。SSLClientCAKeyring 属性可用于定义包含可信 CA 证书的密钥环文件。

HTTPS 的公共设置将使用本地 HTTP 代理建立穿过防火墙的隧道，然后连接到远程 HTTP 服务器（或另一个代理），该服务器接着连接到远程 MQIPT。该连接的服务器端上的 MQIPT 不需要任何特定配置，因为连接请求作为常规 HTTP 连接来对待。

MQIPT 使用 HTTPProxy 和 HTTPServer 属性来分辨本地和远程代理。HTTPProxy 看作为本地 HTTP 代理，HTTPServer 看作为远程服务器（或代理）。

通常 HTTPS 连接是在 HTTP 代理／服务器上的侦听器端口地址 443 上生成，但可使用 HTTPProxyPort 和 HTTPServerPort 覆盖这个缺省值。请参阅第 115 页的『HTTPS 配置』以获取使用 HTTPS 的示例。

Servlet

现在有一个 MQIPT 的 servlet 版本（称为 MQIPTServlet），它可以在 Application Server 上部署为非分布式应用程序。它的工作方式类似于普通 MQIPT，只是工作起来好象只有一个路由。启动 WebSphere MQ 通道的进入连接请求由 MQIPTServlet 的实例处理，且每个实例维护一个到目标队列管理器的持久连接。后继的数据由同一通道通过使用第一个连接请求期间创建的会话标识来维护。

可在 web 子目录中找到一个称为 MQIPTServlet.war 的 Web 应用程序归档文件。必须把这个 war 文件导入／部署到您的 Application Server。如果在导入这个 servlet 时，您需要指定上下文名称，则需要覆盖缺省 UriName 属性以包含新的上下文名称。请参阅第 88 页的『UriName』以获取更多信息

通过设置 web.xml 文件中的属性来完成 MQIPTServlet 的配置。web.xml 可在应用程序服务器的 WEB-INF 子目录中找到。只有现有 MQIPT 属性的子集是适用于 MQIPTServlet 的。下列属性可同 MQIPTServlet 一起使用:

- ClientAccess
- ConnectionLog
- MaxLogFileSize
- QMgrAccess
- Trace

连接日志和跟踪文件写入使用新的称为 LogDir 属性定义的目录中。建议您在启动 MQIPTServlet 之前，先定义这个属性。

要控制 MQIPTServlet 使用的资源数量，可能需要更改某些 Application Server 属性。每个 Application Server 有它自己管理配置数据的方法，通常这是通过使用 GUI、web 界面或编辑配置文件来完成的。要考虑更改的属性是最大活动会话数目或 Application Server 中的 servlet 的实例数目。 这将控制客户机连接的数目，它类似于 MQIPT 中使用的 MaxConnectionThreads 属性。

其它可能需要更改的属性与超时值相关，是否支持持久连接以及在一个持久连接上允许多少个请求。由于 MQIPTServlet 依赖于到目标队列管理器的持久连接，因此必须启用这个属性。其它属性可能需要增加，但这取决于它们的缺省值和正在使用的 WebSphere MQ 连接的类型。WebSphere MQ 客户机连接通常是短暂活动的，因此使用缺省值是相当安全的。队列管理器到队列管理器连接可以持续不确定的时间长度，在这种情况下，建议适当增加某些超时值和在一个持久连接上允许的请求数目。

web.xml 文件中还定义了一个会话超时属性，其缺省值为 30 分钟。这个属性可用于控制客户机的休止状态，如果在指定的时间内没有检测到会话活动，将关闭这个会话。

客户机和 MQIPTServlet 之间的链接中必须至少有一个 MQIPT。必须在连接到 MQIPTServlet 的 MQIPT 中启用 ServletClient 属性，HTTPServer 属性可直接指向 Application Server 或指向为 Application Server 服务的 HTTP 服务器。

要测试 MQIPTServlet 是否已成功启动，您可以启动 web 浏览器，并输入类似如下的 URL 名称:

```
http://localhost:80/MQIPTServlet
```

将在浏览器中看到肯定响应。

MQIPTServlet 已与 IBM WebSphere Application Server 5.0（带和不带 IBM HTTP Server）、Tomcat 3.3 和 Tomcat 4.0 一起测试了。MQIPTServlet 不需要 Java 1.4，它将使用 Application Server 实现的 Java 级别。

请参阅第 112 页的『配置 MQIPT Servlet』以获取如何使用 servlet 的示例。

第 4 章 Socks 支持

Socks 代理是一个用作通过防火墙的受控出口点的网络服务。在防火墙内运行的启用 Socks 的应用程序可使用 Socks 代理连接到远程应用程序。

通过启用 SocksServer 属性，MQIPT 可以充当 Socks 代理，从而允许启用 Socks 的 WMQ 应用程序通过 MQIPT 连接到远程 WMQ 队列管理器。当使用此功能时，在 Socks 握手过程期间获取了目标目的地和目的地端口地址，因此将覆盖 Destination 和 DestinationPort 路由属性。这是支持 WMQ 群集的关键功能。请参阅下面内容以获取更多信息。

MQIPT 也可以代表未启用 Socks 的本地 WMQ 应用程序充当 Socks 客户机。当使用只允许通过 Socks 代理的出站连接的防火墙时，这是有用的。每个 MQIPT 路由可配置为与不同的 Socks 代理进行通信。

请参阅第 108 页的『配置 SOCKS 代理』以获取如何使用 SOCKS 的示例。

群集

可通过 socks 化跨越因特网的群集中的每个队列管理器和启用 MQIPT 充当 SOCKS 代理，使 WebSphere MQ 群集能与 MQIPT 一同使用。由于有很多方法可以将队列管理器配置为群集，下面的说明是基于 *WebSphere MQ Queue Manager Clusters*，SC34-6061 中描述的任务。下图是从称为 “Adding a new queue manager to a cluster” 的任务中的定义扩展而来的。NEWYORK 和 CHICAGO 在称为 HOME 的群集中，它们都包含完整的资源库。NEWYORK、LONDON 和 PARIS 在另一个称为 INVENTORY 的群集中。注意，不需要 socks 化 CHICAGO（因为它在不需要 MQIPT 的群集中）。

INVENTORY 群集中的每个队列管理器被有效地 “隐藏” 在 MQIPT 后面。由于队列管理器已经 socks 化，当群集发送方通道启动时，请求发送到它的目的地（使用 MQIPT 充当 SOCKS 代理）。通常，群集接收方通道上的 CONNAME 用于标识本地队列管理器。但当与 MQIPT 一起使用时，CONNAME 必须标识本地 MQIPT 及其进入侦听器端口。在下图中，所有进入侦听器端口地址为 1414，外出侦听器端口地址为 1415。

有两种方法运行 socks 化的队列管理器。第一种是 socks 化整个运行队列管理器的机器。第二种是只 socks 化队列管理器。不管使用哪种方法，您必须配置 SOCKS 客户机，使之只使用 MQIPT 作为 SOCKS 代理建立远程连接并禁用用户认证。市场中有许多产品可以做到 SOCKS 支持。您必须选择一个产品以支持 SOCKS V5 协议。

请参阅第 118 页的『配置 MQIPT 群集支持』以获取如何配置群集网络的示例。

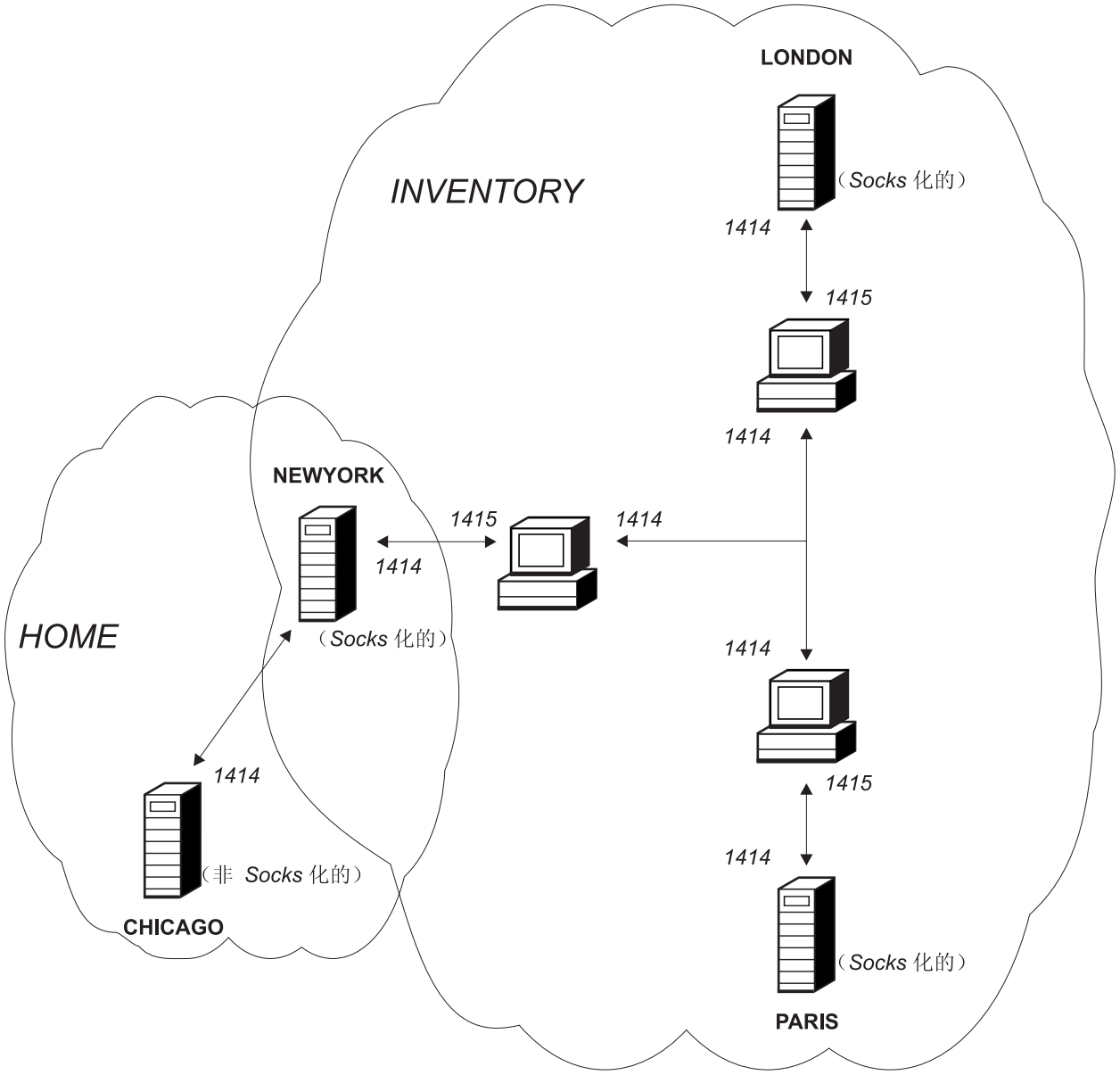

图 6. MQIPT 群集支持

第 5 章 SSL 概述和支持

SSL 协议在不安全的通信通道上提供连接安全性，并确保:

通信保密

可以通过加密将在客户机和服务器间交换的数据使此连接保密（例如，只有它们可以理解数据）。这使保密信息（例如，信用卡号）能安全传送。

通信完整性

连接可靠。消息传送包含基于安全散列函数的消息完整性检查。

认证 客户机可以认证服务器，并且已认证的服务器可以认证客户机。这意味着保证只在预定的两方之间交换信息。认证机制基于数字证书（X.509v3 证书）的交换。

SSL 协议可对通信各方的认证使用不同的数字签名算法。SSL 中使用的加密操作、数据保密加密和消息完整性的安全散列法依赖于客户机和服务器之间的密钥共享。SSL 提供各种允许密钥共享的密钥交换机制。 SSL 可以对加密和散列法使用各种算法。支持各种密码算法；通过使用 SSL 密码套件，指定。支持这些密码套件:

| SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
| SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
| SSL_DH_anon_WITH_AES_128_CBC_SHA
| SSL_DH_anon_WITH_AES_256_CBC_SHA
| SSL_DH_anon_WITH_DES_CBC_SHA
| SSL_DH_anon_WITH_RC4_40_MD5
| SSL_DH_anon_WITH_RC4_128_MD5
| SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
| SSL_DHE_DSS_WITH_3DES__EDE_CBC_SHA
| SSL_DHE_DSS_WITH_AES_128_CBC_SHA
| SSL_DHE_DSS_WITH_AES_256_CBC_SHA
| SSL_DHE_DSS_WITH_DES_CBC_SHA
| SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
| SSL_DHE_RSA_WITH_3DES__EDE_CBC_SHA
| SSL_DHE_RSA_WITH_AES_128_CBC_SHA
| SSL_DHE_RSA_WITH_AES_256_CBC_SHA
| SSL_DHE_RSA_WITH_DES_CBC_SHA
| SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
| SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5#
| SSL_RSA_EXPORT_WITH_RC4_40_MD5
| SSL_RSA_WITH_3DES_EDE_CBC_SHA
| SSL_RSA_WITH_AES_128_CBC_SHA
| SSL_RSA_WITH_AES_256_CBC_SHA
| SSL_RSA_WITH_DES_CBC_SHA
| SSL_RSA_WITH_NULL_MD5
| SSL_RSA_WITH_NULL_SHA
| SSL_RSA_WITH_RC4_128_MD5
| SSL_RSA_WITH_RC4_128_SHA

SSL 握手

当执行密码套件的认证和协商时，在 SSL 客户机和服务器之间的初始连接请求期间发生 SSL 握手进程。

上面列出的具有无名密码套件异常的所有 SSL 密码套件需要服务器认证并允许客户机认证：可以将服务器配置为需要客户机认证。SSL 中的通信对等认证基于公用密钥密码术和 X.509v3 数字证书。应该在 SSL 协议中认证的站点需要专用密钥和包含相应公用密钥以及站点身份信息和证书有效时间的数字证书。证书由认证中心签署，具有这种权限的证书称为签署者证书。一个证书后跟一个或多个签署者证书构成证书链。证书链以从第一个证书（站点证书）开始的事实为特征，可以使用包含在下一个签署者证书中的公用密钥验证链中每一个证书的签名。

当正在建立需要服务器认证的安全连接时，服务器将证书链发送到客户机以证明其身份。 SSL 客户机只有在可以认证服务器（例如，验证服务器站点证书的签名）时，才继续建立到服务器的连接。为了验证该签名，SSL 客户机需相信服务器站点本身，或者至少要相信服务器提供的证书链中的其中一个签署者。必须在客户机端维护可信的站点证书和签署者，以便执行此验证。

SSL 客户机从站点证书开始检查服务器的证书链，并且如果站点证书在信任站点或签署者证书库中，或如果可以根据其信任签署者证书库确认链中的签署者证书，则 SSL 认为站点证书的签名有效。在后一种情况下，SSL 客户机检查是否的确从信任签署者证书到服务器的站点证书正确签署证书链。还检查包含在此进程中的每个证书的格式正确性及有效日期。如果其中任何一种检查失败，则拒绝连接到服务器。验证服务器证书后，客户机在 SSL 协议接下来的步骤中使用嵌入到该证书中的公用密钥。只有服务器确实具有相应的专用密钥，才能建立 SSL 连接。

客户机认证遵循相同的过程：如果 SSL 服务器需要客户机认证，则客户机将证书链发送到服务器以证明其身份，并且服务器根据其信任站点和签署者证书库验证该链。验证客户机证书后，服务器在 SSL 协议接下来的步骤中使用嵌入到该证书中的公用密钥。只有客户机确实具有相应的专用密钥，才能建立 SSL 连接。

SSL 协议本身提供非常高的通信安全性。但是，协议基于应用程序提供的信息操作。只有安全维护该信息基础，才能完成安全通信的整个目的。例如，如果信任站点和签署者证书库被泄露，您可能建立了一个到非常不安全的通信伙伴的安全连接。

WebSphere MQ internet pass-thru 和 SSL

SSL V3.0 实现了使用存储在包含 X509.V3 证书的密钥环文件（文件类型为 .p12 或 .pfx）中的公用密钥密码术标准（PKCS）#12 令牌。密钥环文件也可以包含证书撤销列表（CRL）和权限撤销列表（ARL）。WebSphere MQ internet pass-thru 使用 IBM Secure Socket Lite（SSLite）软件包。

WebSphere MQ internet pass-thru 可以充当 SSL 客户机或 SSL 服务器，这取决于在哪一端启动连接。客户机启动连接，服务器接受连接请求。WebSphere MQ internet pass-thru 路由可同时充当客户机和服务器，尽管出于性能方面的考虑，本实例中建议使用 SSL 代理方式功能。每个 WebSphere MQ internet pass-thru 路由可以使用其自己的 SSL 属性集独立地进行配置。请参阅第 78 页的『route 节参考信息』以获取更多详细信息。

信任设置

密钥环文件包含个人证书，该证书包含签署者证书或者签署者证书链。要在建立连接时启用认证，证书需要信任设置。信任可分为两个级别:

信任为对等
意味着只信任此证书，但是不信任此证书签署的任何证书。

信任为认证中心（CA）
意味着可以信任此证书签署的所有证书。

SSLServerKeyRing 属性标识的 SSL 服务器端密钥环文件应该包含它的个人证书。

SSLClientCAKeyRing 属性标识的 SSL 客户机端密钥环文件应该包含将用于认证从服务器发送的证书的可信 CA 证书列表。

如果还需要客户机认证，则必须在服务器端启用 SSLServerAskClientAuth 属性，且 SSLClientKeyRing 属性标识的客户机端上的密钥环文件应该包含它的个人证书。SSLServerCAKeyRing 属性标识的服务器端密钥环文件应该包含将用于认证客户机的可信 CA 证书的列表。

作为使用可信 CA 签署的证书的替代方法，您可以使用自签署证书。这些证书的示例可在随 MQIPT 一起提供的样本密钥环文件（这些文件在 ssl 子目录中）: sslSample.pfx 和 sslCAdefault.pfx 中找到。

要打开这些密钥环文件中存储的任何一个 PKCS#12 令牌，必须使用密码 `mqiptV1.3`。

可在 ssl 子目录中找到一个称为 KeyMan 的实用程序，使用它可以管理 SSL 证书和密钥环文件。请参阅第 21 页的『KeyMan』以获取安装说明和进一步的信息。

您必须使用操作系统的安全性功能保护密匙环和密码文件，以防止对它们进行未经授权的访问。

测试 SSL

第 91 页的第 20 章，『internet pass-thru 入门』描述了可用于测试 SSL 连接的任务。

许多供应商的可提供证书和证书管理技术，它们包括:

- RSA Security (www.rsasecurity.com)
- Entrust Technologies (www.entrust.com)
- Verisign (www.verisign.com)

SSL 错误消息

如果在 SSL 方法调用之一中使用了无效的参数值，或者向 SSL 协议提供了错误的数据，将会在 SSLRuntimeException 中看到下列错误代码。

表 1. SSLRuntimeException 错误消息

标识	描述
1	方法用法错误，或者一个或多个输入参数超出范围
2	无法处理提供的数据

表 1. SSLRuntimeException 错误消息 (续)

3	无法验证提供的数据的签名
10	签署者证书的主题名称不匹配证书的签发者名称
11	不支持的证书类型
12	在有效期之前使用证书
13	证书到期
14	无法验证证书签署
15	无法使用证书
20	所有客户机提议的密码套件不被服务器支持。
21	所有客户机提议的压缩方法不被服务器支持。
22	没有可用的证书
23	不支持的算法或格式类型
24	拒绝过时的信息
25	取消了一个证书
26	一组 CRL 是不完整的（缺少一些增量 CRL）
27	要认证的名称已经存在
28	要认证的公用密钥已经存在
29	有些序列号或者密钥（证书、CRL）是错误的
30	授权失败

如果终止执行 SSL 握手协议，则会抛出一个 SSLException。

表 2. SSLException 错误消息

标识	描述
3	SSLContext 中定义的连接超时到期，且没有从对等接收到响应
4	对等在 SSL 握手期间放弃了连接，且没有给出进一步的错误指示
10	接收到意外的消息
20	接收到带有损坏记录 MAC 的消息
30	解压缩故障
40	握手故障
41	对等没有发送证书
42	接收损坏的证书
43	接收到不支持的证书
44	接收到撤消的证书
45	接收到到期的证书
46	接收到未知的证书
47	检测到非法参数

LDAP 和 CRL

WebSphere internet pass-thru 支持使用轻量级目录访问协议（LDAP）服务器在数字证书上执行证书撤销列表（CRL）认证。LDAP 支持是以类似于在基本 WebSphere MQ 中实现 LDAP 支持的方法实现的，因为相同的 LDAP 服务器可能同时用于 WebSphere MQ 和 MQIPT。有关将 LDAP 服务器与 WebSphere MQ 一起使用的进一步信息可在书籍 “WebSphere MQ Security Version 5.3” SC34-6079-01 中的 chapter 15 中找到。下面包含这本书籍中的摘录以供参考。

在 SSL 握手期间，通信伙伴使用数字证书进行相互认证。认证可以包含对接收到的证书作是否仍是可信的检查。认证中心（CA）可因为各种原因撤销证书，它们包括:

- 所有者移动到了不同的组织
- 专用密钥不再是保密的

CA 在证书撤销列表（CRL）中发布已撤销的个人证书。已撤销的 CA 证书已经在权限撤销列表（ARL）中发布。本章中对 CRL 的后继引用也应用到 ARL。

市场中有多个私有的 LDAP 目录服务器。WebSphere internet pass-thru 已经与 IBM Directory Server 一起经过使用: 请参阅 http://www.ibm.com/software/network/directory/server。可在与已安装的产品一起提供的文档中找到有关安装和维护 LDAP 服务器的说明。

有关管理 CRL 和 ARL 的进一步信息可在书籍 “WebSphere MQ Security Version 5.3” SC34-6079-01 中找到。

MQIPT 在每个路由上可以最多支持两台 LDAP 服务器。第一台 LDAP 服务器作为主服务器对待，第二台 LDAP 服务器看作为备份服务器，仅当无法联系到主服务器时才使用备份服务器。备份服务器应该是主服务器的镜像映象。

对 LDAP 服务器上存储的信息的访问可能受用户标识和密码保护。如果是这种情况，可使用 LDAP*Userid 和 LDAP*Password 属性。

当 MQIPT 从密钥环文件装入 PKCS#12 令牌时，将检查任何 CA 证书的 CRL 有效性。如果 CA 证书有连接的 CRL，将检查它是否已到期，如果已到期，将从 LDAP 服务器检索更新的 CRL。检索得到的任何 CRL 将装入当前令牌并连接到它的 CA 证书。更新后的令牌可以保存在密钥环文件中（请参阅第 78 页的『route 节参考信息』中的LDAPSaveCRL 属性）。

当向主 LDAP 服务器发送查询时，如果没有匹配给定 CA 的条目，则假设该 CA 没有 CRL。将不使用备份服务器。但是，如果无法联系到主 LDAP 服务器或者没有在给定的时间帧内返回，则将使用备份服务器。来自备份服务器的任何错误将导致客户机连接终止。可通过将属性 LDAPIgnoreErrors 设置为 true，覆盖此操作。

注意

如果您启用 LDAPIgnoreErrors 属性，可使用已撤销的证书来生成 SSL 连接。

LDAP 客户机模型基于 “com.sun.jndi.ldap.LdapCtxFactory” 实现。MQIPT 检索到的任何 CRL 将保持在高速缓存中，且被该路由上的所有连接共享。

如果高速缓存的 CRL 已到期，将从高速缓存中除去 CRL，并从 LDAP 服务器检索新的 CRL。如果新的 CRL 不可用，则仍将拒绝此连接。

从 LDAP 服务器检索的 CRL 还要检查是否显示到期和警告系统控制台消息（MQCPW001）。到期的 CRL 仍将装入系统，引用该 CRL 的任何连接请求将被拒绝。应该使用当前的 CRL 替换 LDAP 服务器中的到期 CRL。

LDAPCacheTimeout 属性可用于控制每隔多久清除一次 CRL 高速缓存。缺省值为 1 天。将这个值设置为 0 表明高速缓存条目将在路由重新启动后再清除。

到期的 CRL 可存储在密钥环文件或 LDAP 服务器中。如果没有发出新的 CRL，将拒绝进一步的连接请求。您可以通过启用 IgnoreExpiredCRLs 属性忽略到期的 CRL。

注意

如果您启用 IgnoreExpiredCRLs 属性，可使用已撤销的证书来生成 SSL 连接。

高级加密标准

高级加密标准（AES）将成为新的联邦信息处理标准（FIPS）发布，它指定供美国政府组织使用的、用于保护敏感（非机密）信息的密码算法。国家标准和技术局（NIST）也预见除美国政府之外，AES 将广泛地被非官方组织、公共机构和个人（某些情况下，在美国之外）使用。

从密钥环文件选择证书

在同一密钥环文件中可存储多个个人证书，因此可在客户机端使用 SSLClientSite* 属性来选择发送到服务器用于认证的证书，可在服务器端使用 SSLServerSite* 属性来选择发送到客户机用于认证的证书。

使用这些属性时，可以根据证书的专有名称（DN）来选择证书。或者，可通过使用 SSLServerSiteLabel 和 SSLClientSiteLabel 属性使用证书标号来选择证书。

加密密钥环密码

可使用 mqiptPW 脚本加密用于打开密钥环文件的密码。加密后的密码存储在文件中，它可供以下任何属性使用：
SSLClientKeyRingPW、SSLClientCAKeyRingPW、SSLServerKeyRingPW 和 SSLServerCAKeyRingPW。

命令格式:

```
mqiptPW <password> <file name> <-replace>
```

其中

password
是打开给定密钥环文件所需的明文密码。

file name
是要创建的密码文件的名称

replace

覆盖 <file name>（如果存在）所需的选项

密码可以包含空格字符（ “ ” ），但整个密码字符串必须包含在引号中，以使这样能被接受。密码的长度或格式没有限制。

注: 从先前级别的 WebSphere Internet pass-thru 迁移的用户需要使用一份经过加密的密码文件副本替换当前的包含明文密码的密码文件。

使用密钥管理实用程序（例如，KeyMan）时，您必须使用密码 mqiptV1.3 打开任何一个样本密钥环文件。

KeyMan

WebSphere Internet pass-thru 现在提供了一个称为 KeyMan 的独立实用程序，以允许管理 SSL 证书和密钥环文件。可在 ssl 子目录中找到一个包含 KeyMan 的 zip 文件。要安装 KeyMan，解压缩此文件到一个临时目录，然后按 README.txt 文件中的说明进行操作。KeyMan 具有许多功能，但本节涉及的范围只限于创建测试证书和管理包含 PKCS#12 令牌的密钥环文件。

KeyMan 是用于公用密钥基础结构（PKI）的客户机端的管理工具。KeyMan 管理密钥、证书、证书撤销列表（CRL）和用于存储和检索它们的各自的库。支持完整的证书生命周期和处理用户证书期间涉及的过程。

KeyMan 管理包含密钥、证书和撤销列表的集合的库。库称为令牌。令牌包括特定应用程序（例如，WebSphere Internet pass-thru）的信任设置。通常，令牌包含专用密钥和用于其它站点认证用户的相关证书链。另外，令牌还包含信任通信伙伴的证书和认证中心（CA）。

支持的令牌类型

KeyMan 支持许多种不同类型的令牌。令牌是包含密钥、证书、CRL 和信任设置的库。有些令牌只能存储这些物件类型的子集。

PKCS#7 令牌

包含一组证书（可选地，包含关联的 CRL）。无法在这种类型的库中存储密钥。这种库不需要认证。证书和 CRL 受签名的保护。然而，对手可以更改存储在特定 PKCS#7 令牌中的物件集合。当使用有些上下文定义期望的物件集合时，使用此类型的令牌。

PKCS#12 令牌

包含专用密钥、证书和关联的 CRL。内容受用户 passphrase 保护。可使用不同强度的算法来保护公用物件（证书、CRL）和专用物件（密钥）。

PKCS#11（CryptoKi）库

PKCS#11 定义到加密令牌的接口。这些令牌可以存储密钥和证书。不支持存储 CRL。对令牌的访问受个人标识号（PIN）的保护。您必须指定 KeyMan 使用的令牌特定 PKCS#11 DLL 来访问令牌。

KeyMan 支持 PKCS#11 V2.01 和 2.10 DLL。

PKCS#7 和 PKCS#12 是软令牌，可从不同的媒体（例如，文件、URI 和剪贴板）检索。

KeyMan 具有特殊的能力，它可以使用未知格式的数据构造 PKCS#7 令牌。它扫描数据以查找 X.509 证书和 CRL，然后从检测到的证书和 CRL 构造 PKCS#7 令牌。如果您有包含证书或 CRL 的电子邮件，您可在 KeyMan 中打开电子邮件文件夹，KeyMan 会尝试抽取 X.509 物件。当然，数据无法存储回原来的格式。抽取的数据可以使用 PKCS#7 格式存储到一个文件。

支持的标准数据格式

KeyMan 支持许多种标准数据格式。下面是描述它们的含义和用法的内容:

PKCS#7

这种数据格式是证书和 CRL 的集合。PKCS#7 描述的证书和 CRL 集合是不受保护的。然而，每个独立的证书和 CRL 是受签名保护的。当使用有些上下文定义期望的证书和 CRL 集合时，使用 PKCS#7。在 Windows 系统上 PKCS#7 文件的标准文件后缀是 .p7r 和 .p7b。

PKCS#10

PKCS#10 定义证书请求消息。它包含公用密钥和有关请求者的 X.500 名称的信息。消息使用相应的专用密钥签署。PKCS#10 消息可以使用二进制格式和 ASCII 保护格式来生成。消息必须提交到认证中心（CA）。

PKCS#12

PKCS#12 供浏览器和 Web 服务器使用，以导入和导出专用密钥和关联的证书。KeyMan 可以读写这些 PKCS#12 文件。然而这些程序只能理解非常特定的 PKCS#12 概要文件。KeyMan 可以生成更常用的 PKCS#12 文件。KeyMan 可在单个 PKCS#12 文件中存储专用密钥、证书、CRL 和相应的信任设置的集合。PKCS#12 文件受 passphrase 保护。通常，PKCS#12 令牌包含特定应用程序的信任策略。在使用 IBM BlueZ SSLite 的情况下，密钥和关联的证书链将用于客户机／服务器认证。取决于各自的信任设置，其它证书代表可信的 CA 或者可信的服务器。在 Windows 系统上 PKCS#12 文件的标准文件后缀是 .p12 和 .pfx。

SPKAC

SignedPublicKeyAndChallenge（SPKAC）是从 CA 请求证书的数据格式。每次使用 HTML 标记 <keygen> 时，Netscape 会生成这种特定的格式。它包含签署的公用密钥和提问。 KeyMan 可以使用二进制和 Base64 格式生成这种数据格式。

X.509 V3 证书

KeyMan 可以读取二进制格式或者 ASCII 保护所包装的 X.509 V3 证书。KeyMan 可以打开或者导入这些文件。从令牌写这两种格式的单个证书是可能的（**证书详细信息 -> 保存图标**）。在 Windows 系统上 X.509 证书文件的标准文件后缀是 .crt、.cer 和 .der。

X.509 V2 证书撤销列表（CRL）

KeyMan 可以读取二进制格式或者 ASCII 保护所包装的 X.509 V2 CRL。无法打开单个 CRL。KeyMan 只可导入 CRL 到已经包含关联的 CA 证书的令牌。写二进制或者 ASCII 保护格式的单个 CRL 是可能的（**证书详细信息 -> CRL 详细信息 -> 保存图标**）。在 Windows 系统上 X.509 CRL 文件的标准文件后缀是 .crl。

KeyMan FAQ

对于密码术和相关术语的一般问题，请参阅 RSA Laboratories 和它们的 “Frequently Asked Questions About Today’s Cryptography”。下列 FAQ 讨论与 KeyMan 相关的问题。

KeyMan 可以读取 Netscape 或 Internet Explorer 生成的 PKCS#12 文件吗?

KeyMan 可以读取 Netscape 或 Internet Explorer 生成的 PKCS#12 文件，但您要提供您所知道的保护内容的密码。

KeyMan 可以创建 Netscape 或 Internet Explorer 可以读取的 PKCS#12 文件吗?

PKCS#12 标准提供了很大的自由以选择算法和排列内容。浏览器只接受所有可能选项中非常特定的概要文件。KeyMan 可以创建 Netscape 或 Internet Explorer 可以读取的 PKCS#12 文件。由于 KeyMan 允许您对 PKCS#12 进行更多的操作，您可以创建这些浏览器无法理解的文件。浏览器的公共概要文件看上去类似这样：公用 / 专用加密（参见**菜单选项 -> PKCS#12 设置**）分别应该是 “RC2（40 比特）” / “DES（168 比特）”。PKCS#12 令牌中应该有且只有一个专用证书。

什么是专用证书?

如果 KeyMan 检测到匹配的密钥和证书，KeyMan 会把它们组合起来成为一张专用证书。这意味着，对于任何专用证书您也相应地拥有与之对应的专用密钥。如果您导入证书到令牌，KeyMan 检查是否有匹配的专用密钥，并自动把密钥和导入的证书组合起来成为一张专用证书。如果发生此操作，KeyMan 会弹出一个对话框通知您。

什么是 CA 或对等证书?

证书包含在令牌建立信任中。它们定义您可以信任谁。信任意味着什么和证书的精确评估取决于使用令牌的应用程序。在 KeyMan 中您可以为证书设置两种类型的信任：CA 和对等。如果您信任一张证书为 CA，则隐含地信任此 CA 直接或间接签署的任何证书。如果您设置信任级别为 “对等”，您只信任此特定的证书。信任不扩展到 “对等” 证书签署的证书。

那些既不是专用，也不是 CA 或者对等证书的证书是什么?

KeyMan 试图存储每张专用证书到根证书为止的完整链。这些证书不必是可信的，因此不会出现在 CA 或者对等证书中。如果您选择密钥环 “所有证书物件”，则可以找到这些证书。不可信的证书没有图标。

什么是令牌?

令牌是密钥、证书和 CRL 的集合。令牌存储在某些媒体（例如，文件、URL、某个硬件）中。不同类型的令牌具有不同的能力：软件令牌、硬件令牌、不受保护的令牌和受密码或 PIN 保护的令牌。

什么是密钥环?

令牌由一组密钥环组成。特定的密钥环标识一组特定的物件（例如，相同信任级别的证书、您拥有专用密钥的证书或者没有匹配证书的密钥）。

第 6 章 服务质量

服务质量（QoS）

IBM WebSphere Edge Server 通过使用 Linux 平台上的事务服务质量插件提供带宽管理解决方案。事务服务质量（TQoS）是指向网络用户提供的全面服务（以元素（例如吞吐量和延迟）观点来看）。可以把属性设置为确保与所有外出数据关联的服务质量通过一个连接发送。这使策略管理员能定义标识与特定服务器相关的流量的规则和此流量的唯一区别服务控制的策略操作。例如，安装可以定义一个策略。该策略指定给予支持某些货物的销售的服务器流量相关的外出流量以优先考虑（相反，不优先考虑与支持客户机浏览的服务器流量相关的外出流量）。另外，TQoS 也允许管理员收集相应策略的性能数据以监视策略是否实现了他们所要的服务级别目的（如连接吞吐量、延迟和丢失率这样的重要度量指标）。MQIPT 只需要安装并运行策略代理（pagent）以实现服务质量（QoS）。

TQoS 策略在策略配置文件（pagent.conf）中定义或者使用 LDAP 服务器定义。TQoS pagent 可以访问策略配置文件，也可以转至 LDAP 服务器，或者同时从两者检索 TQoS 策略条目。*IBM Edge Server Administration Guide* 给出了有关 pagent 的更多信息，可在以下 URL 找到它:

```
http://www.ibm.com/software/webservers/edgeserver/library.html
```

您可以到这个站点在线查看 HTML 或者下载 PDF 版本。这两种格式的帮助中您都可以搜索 TQoS。

可从与下载 MQIPT 的相同位置下载带有安装和管理说明的 TQoS 代码。请参阅 WebSphere MQ family SupportPacs 站点，地址为 http://www.ibm.com/webspheremq/supportpacs，单击 Category 3 - Product Extensions。

随 MQIPT 一起提供了一个称为 libmqiptqos.so 的虚拟库（您可在 MQIPT lib 子目录中找到）。这使 MQIPT 无需安装 TQoS pagent 即可在 Linux 平台上运行。安装 TQoS 后，您可能需要使用 TQoS 使用的库替换这个虚拟库。可在 MQIPT bin 子目录中找到一个称为 mqiptQoS 的脚本以帮助完成此任务。使用下列命令来重命名虚拟库和定义到实 TQoS 运行时库的软链接:

```
mqiptQoS -install
```

使用 mqiptQoS -remove 将撤消以上操作。

MQIPT 只需要安装并运行 pagent 以实现服务质量（QoS）。通过使用 MQIPT，可在路由上为各个方向的数据流设置应用程序优先级，因此这将影响到所有使用该路由的通道。此优先级是使用 MQIPT 属性 QosToCaller 和 QosToDest（请参阅第 78 页的『route 节参考信息』以获取更多信息）定义的，此处使用的值必须与 pagent.conf 控制文件中的 ApplicationPriority 策略定义相匹配。如果 pagent 没有找到匹配策略，则不会为数据指定任何优先级。对策略的任何更改要等到 pagent 重新启动之后才会反映到 MQIPT。请参阅第 105 页的『配置服务质量（QoS）』以获取有关策略定义的更多信息。

第 7 章 Network Dispatcher

Network Dispatcher 支持

MQIPT 可同 IBM Network Dispatcher 一起使用以提供增强的可用性和许多服务器之间的负载平衡（通过使用定制顾问程序）。本节假设您熟悉 Network Dispatcher 和定制顾问程序。

MQIPT 提供了两个顾问程序；可在 `lib` 子目录中找到它们。按照 *Network Dispatcher User's Guide*（GC31-8496）中的说明安装定制顾问程序。图 7 显示一个 MQIPT 使用 Network Dispatcher 来监控端口地址 1414 的示例。注意，每个 MQIPT 必须有相同的配置文件。

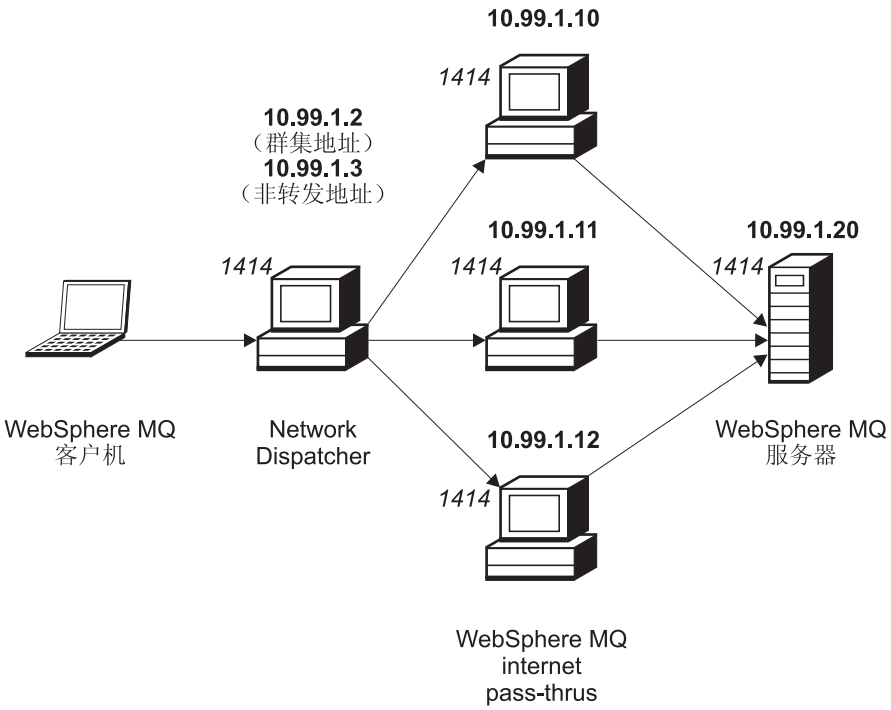

图 7. Network Dispatcher 与 MQIPT 一同使用

按照 *Network Dispatcher User's Guide* 第五章中的说明配置调度器组件定义端口 1414 和负载平衡服务器机器。您可以使用管理客户机的菜单选项或者 "ndcontrol" 行方式命令。例如:

```
ndcontrol port add 10.99.1.2 : 1414
ndcontrol server add 10.99.1.2 : 1414 : 10.99.1.10
ndcontrol server add 10.99.1.2 : 1414 : 10.99.1.11
ndcontrol server add 10.99.1.2 : 1414 : 10.99.1.12
```

MQIPT 配置文件中的路由定义看上去类似这样:

```
[route]
ListenerPort=1414
Destination=10.99.1.20
DestinationPort=1414
NDAdvisor=true
```

您只可从命令行启动（和停止）定制顾问程序。例如:

```
ndcontrol advisor start mqipt_normal 1414
```

此命令以“普通”方式启动 MQIPT 顾问程序，在这种方式下基本顾问程序自己计时以计算每个 MQIPT 的权值因子。要在“替换”方式下使用 MQIPT 顾问程序，添加下面这行到 MQIPT 路由定义:

```
NDAdvisorReplaceMode=true
```

您还必须启动 mqipt_replace 定制顾问程序而不是 mqipt_normal 定制顾问程序。例如:

```
ndcontrol advisor start mqipt_replace 1414
```

当使用顾问程序来监控 SSL 侦听器端口（即，它的 mqipt.conf 配置文件中 SSLServer=true），则必须在 Network Dispatcher 的工作目录中放置一个“触发器”文件。“触发器”文件具有特定的名称，它与要监控的路由相关。例如，如果路由 1414 的 SSLServer=true，则必须在 c:\winnt\system32 目录中放置一个称为 mqipt1414.ssl 的文件（在 Windows NT 上）。参见 mqipt1414Sample.ssl 文件以获取更多信息。

第 8 章 Java 安全性管理器和安全性出口

Java 安全性管理器

Java 安全性管理器的支持原本是为使用 SSL 代理方式功能来管理套接字连接控制而实现，但是它也可以与任何其它 MQIPT 功能一同使用以提供进一步级别的安全性。

MQIPT 使用 java.lang.SecurityManager 类中定义的缺省 Java 安全性管理器。可使用全局属性 SecurityManager 启用或禁用 MQIPT 中的 Java 安全性管理器功能部件，请参阅第 77 页的『Global 节参考信息』以获取更多信息。

Java 安全性管理器使用两个缺省策略文件。称为 $JREHOME/lib/security/java.policy 的全局系统策略文件（其中 $JREHOME 是包含您的 Java 运行时环境的目录）供主机上的虚拟机的所有实例使用。第二个称为 .java.policy 的用户特定策略文件可能存在于用户主目录。也可能使用附加的 MQIPT 策略文件，请参阅第 77 页的『Global 节参考信息』以获取更多信息。要使用附加的策略文件，确保全局系统策略文件（java.security）中的 policy.allowSystemProperty 属性设置为 true。

策略文件的语法是相当复杂的，尽管可以使用文本编辑器修改它，仍然建议使用 Java 提供的 policytool 实用程序做任何修改。policytool 实用程序可在 $JREHOME/bin 目录中找到，且在 Java 文档中有完整的记录。

MQIPT 提供了一个样本策略文件（mqiptSample.policy），其中表明了运行 MQIPT 需要设置哪些许可权。只需要添加／更改／删除 java.net.SocketPermission 条目以匹配您自己特定的要求，用来控制哪些人可以连接到 MQIPT，以及 MQIPT可以连接到哪些人。样本文件假设 MQIPT 安装在缺省主目录中，例如 c:\Program Files\IBM\Websphere MQ internet pass-thru\。如果您把 MQIPT 安装到另外的位置，需要相应修改 codeBase 和 java.io.FilePermission 定义。

通常使用三个属性定义许可权并控制套接字连接，它们的值是:

类许可权

java.net.SocketPermission

要控制的名称

它的组成格式是 hostname:port，其中每个名称的组件可使用通配符指定。主机名可以是域名或者是 IP 地址。主机名的最左边可指定为星号。例如，harry.company1.com 可被下面每个字符串匹配:

- harry
- harry.company1.com
- *.company1.com
- *
- 123.456.789（假设这是 harry.company1.com 的 IP 地址）

名称的端口组件可指定为单个端口地址或者端口地址范围，例如:

1414 仅端口 1414

1414- 所有大于或等于 1414 的端口地址

-1414 所有小于或等于 1414 的端口地址

1-1414

所有 1 到 1414（包括 1 和 1414）之间的端口地址

允许的操作

`java.net.SocketPermission` 使用的操作有:

- 接受: 允许许可权接受来自指定目标的连接
- 连接: 允许许可权连接到指定目标
- 侦听: 允许许可权在指定的端口或者用于连接请求的端口上侦听。
- 解析: 这允许许可权使用 DNS 名称服务把域名解析为 IP 地址

也可以使用 `java.security.manager` 和 `java.security.policy` Java 系统属性来控制 Java 安全性管理器，但是建议您使用 SecurityManager 和 SecurityManagerPolicy 属性控制 MQIPT。

安全性出口

注意

MQIPT 在单个 JVM 中运行，因此用户定义的安全性出口可能在以下方面危害 MQIPT 的正常操作:

- 影响系统资源
- 生成瓶颈
- 降低性能

在生产环境中实现安全性出口之前，应该广泛地测试您的安全性出口所造成的影响。

安全性出口的用途是用于控制对 Destination 路由属性定义的目标目的地的访问。在 MQIPT 生成到目标目的地的连接之前，当 MQIPT 从客户机接收到连接请求时，将调用安全性出口。安全性出口可根据初始连接属性确定是否允许完成连接。

当路由启动时，将调用安全性出口对其进行初始化并使它自己处理连接请求准备就绪。初始化过程应该用于装入任何用户数据，并准备此数据以用于快速和方便的访问，从而使处理连接请求的时间最短。

每个路由可以有它自己的安全性出口。SecurityExit 属性用于启用／禁用用户定义的安全性出口。SecurityExitName 属性用于定义用户定义的安全性出口的类名。SecurityExitPath 属性用于定义包含类文件的目录名。如果没有设置这个属性，则假设类文件将在 exits 子目录中找到。SecurityExitPath 也可以定义包含用户定义的安全性出口的 jar 文件的名称。最后，MQIPT 使用 SecurityExitTimeout 属性来确定验证连接请求时它等待来自安全性出口的响应的时间。

已创建了称为 SecurityExit 的新类以允许 MQIPT 调用用户定义的安全性出口。必须使用用户定义的安全性出口来扩展新类，且应该覆盖它的大部分方法以提供所需的功

能。SecurityExitResponse 对象用于将数据传递回 MQIPT，MQIPT 使用该数据来确定应该接受还是拒绝连接请求。SecurityExitResponse 还可以包含新的目的地和目的地端口地址，以用于覆盖路由定义的属性。

提供了三个样本安全性出口以演示如何实现安全性出口。第一个样本（称为 SampleSecurityExit）演示了如何根据 WMQ 通道的名称来控制对 WebSphere MQ 队列管理器的访问。它只允许通道名以字符串 “MQIPT” 开头的连接。请参阅第 135 页的『安全性出口』以获取更多信息。

第二个样本（称为 SampleRoutingExit）允许将客户机连接请求动态路由到已定义的 WebSphere MQ 服务器的池，每台服务器主管具有相同名称和相同属性的 QM。样本包含一个配置文件，该配置文件包含服务器名列表。请参阅第 137 页的『路由安全性出口』以获取更多信息。

第三个样本（称为 SampleOneRouteExit）允许动态路由到从连接请求中使用的 WMQ 通道名称派生出来的 WMQ QM。样本包含一个配置文件，该配置文件包含 QM 名称到服务器名的映射。请参阅第 140 页的『动态一个路由出口』以获取更多信息。

com.ibm.mq.ipt.SecurityExit 类

必须使用用户定义的安全性出口扩展这个类及其公用方法以获取对某些公用数据的访问，并允许进行某些 MQIPT 初始化。在 MQIPT 调用每个方法之前，有些属性将变为可用以供方法使用。它们的值可通过使用这个类中定义的相应的 get 方法而检索得到。请参阅下面内容以获取支持的方法的完整列表。

方法

init

```
public void init () throws IPTException
```

下列属性是可用的:

- 侦听器端口
- 目的地
- 目的地端口
- 版本

路由启动时，MQIPT 将调用 init 方法。当从这个方法返回时，安全性出口必须是验证连接请求准备就绪。这个方法中如果抛出任何异常，路由将无法启动。

refresh

```
public void refresh () throws IPTException
```

下列属性是可用的:

- 侦听器端口
- 目的地
- 目的地端口

当 MQIPT 管理客户机要求 MQIPT 进行自身刷新时，MQIPT 将调用 refresh 方法。通常在配置文件中属性发生更改后会调用此操作。MQIPT 将从配置文件

装入所有属性，并确定哪些属性发生了更改，以及路由是否需要立即重新启动，还是可以等到 MQIPT 下次重新启动。

这个方法应该执行重新装入它使用的任何外部数据（即，执行 init 方法期间所装入的数据）。这个方法中如果抛出任何异常，将导致路由被禁用。

close

```
public void close ()
```

下列属性是可用的:

- 侦听器端口
- 目的地
- 目的地端口

当 MQIPT 管理客户机要求 MQIPT 停止时，MQIPT 将调用 close() 方法。它应该释放在其操作期间获取的任何系统资源。MQIPT 将等到这个方法完成后再关闭。

如果启用了安全性出口，则也将调用这个方法，但现在它在配置文件中是禁用的。

validate

```
public SecurityExitResponse validate ()
```

下列属性是可用的:

- 侦听器端口
- 目的地
- 目的地端口
- 超时
- 客户机 IP 地址
- 客户机端口地址
- 通道名称
- 队列管理器名称

当 MQIPT 接收到要验证的连接请求时，MQIPT 将调用 validate 方法。如果启用了 SSLProxyMode 属性，通道名称和队列管理器名称将不可用，因为这个功能仅用于隧道传输 SSL 数据，因此，通常从初始数据流获取的数据将是无法读取的。队列管理器名称对于 WMQ 客户机连接将不可用，因为此信息在到目标队列管理器的连接建立之后才可用。

安全性出口必须返回 SecurityExitResponse 对象，该对象包含以下信息:

- 原因码（必须设置）
- 新的目的地地址（可选）
- 新的目的地侦听器端口地址（可选）
- 消息（可选）

原因码将确定 MQIPT 将接受还是拒绝连接。可选地可以设置 newDestination 和 newDestinationPort 字段以定义新的目标（QM）。如果您不设置这些属性，将使用配置文件中定义的路由 Destination 和 DestinationPort 属性。任何消息将附加到连接日志文件条目。

支持的用于获取属性的方法:

public int getListenerPort()
检索路由侦听器端口 – 如 ListenerPort 属性定义的那样

public String getDestination()
检索目的地地址 – 如 Destination 属性定义的那样

public int getDestinationPort()
检索目的地侦听器端口地址 – 如 DestinationPort 属性定义的那样

public String getClientIPAddress()
检索生成连接请求的客户机的 IP 地址

public int getClientPortAddress()
检索生成连接请求的客户机使用的端口地址

public int getTimeout()
检索超时值。MQIPT 将等待安全性出口验证请求 – 如 SecurityExitTimeout 属性定义的那样

public int getConnThreadID()
检索处理连接请求的连接线程标识，该标识有助于进行调试

public String getChannelName()
检索连接请求中使用的 WMQ 通道名称

public String getQMName()
检索连接请求中使用的 WMQ 队列管理器名称

public boolean getTimedout()
可被安全性出口使用以确定超时是否到期

com.ibm.mq.ipt.SecurityExitResponse 类

这个类将用于将响应从用户定义的安全性出口传送回 MQIPT，并将用于确定是接受还是拒绝连接请求。此类型的对象仅在 validate 方法（请参阅上述内容）中创建。有方便的构造函数可用于创建这些对象，每个属性有几个设置方法。请参阅样本安全性出口以获取更多信息。

创建缺省 SecurityExitResponse 对象将拒绝连接请求。

支持的构造函数:

public SecurityExitResponse (String dest, int destPort, int rc, String msg) throws IPTException

其中:

- `dest` 是新的目标目的地
- `destPort` 是新的目的地端口地址
- `rc` 是原因码
- `msg` 是将添加到连接日志条目的消息

public SecurityExitResponse (String dest, int destPort, int rc) throws IPTException

public SecurityExitResponse (int rc, String msg) throws IPTException

public SecurityExitResponse (int rc) throws IPTException

支持的用于设置属性值的方法:

public void setDestination(String dest)
为连接请求设置新的目的地地址

public void setDestinationPort(int port) throws IPTException
为连接请求设置新的目的地侦听器端口地址 – 对于无效的端口地址，抛出 IPTException

public void setMessage(String msg)
添加消息到连接日志记录

public void setReasonCode(int rc) throws IPTException
为连接请求设置原因码 – 对于未知值，抛出 IPTException

有效的原因码:

- `SecurityExitResponse.OK = 0`
- `SecurityExitResponse.NOT_AUTHORIZED = 1`
- `SecurityExitResponse.NOT_READY = 2`

跟踪

为了帮助诊断用户定义的安全性出口中的任何问题，您可以启用跟踪工具（它类似于 MQIPT 使用的跟踪工具）。将路由 Trace 属性的值设置为 1-5 将在 errors 子目录中创建跟踪文件。跟踪文件的名称与安全性出口的名称相同。

可能有安全性出口的多个实例在同时运行，因此可使用线程标识符来标识跟踪文件中的各个条目。

安全性出口启动时，MQIPT 将执行跟踪功能的初始化；所有您必须要做的是选择您要跟踪什么信息。样本用户出口中有许多跟踪示例。

跟踪的最小需求是 `entry` 调用、`exit` 调用和您要跟踪的数据。例如:

```
<a_method>
{
  SecurityExit.rastlRoute.entry(RASITraceEvent.TYPE_ENTRY_EXIT,
                                this,
                                "method_name");

.
.
.
  <code>

.
.
.
  SecurityExit.rastlRoute.trace(RASITraceEvent.TYPE_MISC_DATA,
                                this,
                                "data");

.
.
.
  <code>
```

```
                    :
                      SecurityExit.rastlRoute.exit(RASITraceEvent.TYPE_ENTRY_EXIT,
                                                   this,
                                                   "method_name");
                    }
```

第 9 章 端口地址控制

端口地址控制

使用 MQIPT 时，可以通过设置路由上的 OutgoingPort 属性来限制生成外出连接时它使用的本地端口地址的范围。通过使用 MaxConnectionThreads 值来计算本地端口地址的范围。例如，如果 OutgoingPort 设置为 1600，MaxConnectionThreads 设置为 20，则对于该路由，本地端口地址的范围将是 1600-1619。MQIPT 管理员负责确保路由中的端口地址没有冲突。如果没有定义 OutgoingPort，将使用缺省值 0，它表示将对每个连接使用系统分配的端口地址。

请参阅示例第 124 页的『分配端口地址』以获取更多信息。

多主机系统

当使用多主机系统时，您可以通过使用 LocalAddress 属性指定将外出连接绑定到哪个 IP 地址。该属性中不支持主机名。

第 10 章 其它安全性注意事项

其它安全性注意事项

如果您选择不使用 SSL，MQIPT 允许通道安全性流，这样 WebSphere MQ 通道出口可用于提供整个通道从头至尾的安全性。

MQIPT 有几项附加的功能，用于帮助设计者构建安全解决方案:

- 如果内部网络中有许多客户机都试图建立外出连接，它们可以都通过位于防火墙内的 MQIPT 建立外出连接。防火墙管理员只授权给 MQIPT 机器以外部访问权限。
- MQIPT 只可连接到在其配置文件中已明确配置的队列管理器（除非 MQIPT 正充当 SOCKS 代理或正在使用安全性出口）。
- MQIPT 验证它接收到的消息、发送是否是有效的以及是否遵从 WebSphere MQ 协议。这有助于防止将 MQIPT 用于安全性攻击（使用 WebSphere MQ 之外的协议）。如果 MQIPT 充当 SSL 代理，当所有 WebSphere MQ 数据和协议都是加密的时候，MQIPT 只能确保初始 SSL 握手。在这种情况下，建议您使用 Java 安全性管理器，请参阅第 29 页的『Java 安全性管理器』。
- 它允许通道出口运行它们自己的端到端安全性协议。
- MQIPT 允许您通过设置 `MaxConnectionThreads` 属性限制进入连接总数。这有助于保护易受攻击的内部队列管理器免受拒绝服务攻击。

您必须保护 MQIPT 的配置文件 `mqipt.conf`，因为此文件控制对内部主机的访问。而且您必须阻止对命令端口（如果已启用）未授权的访问，因为这样的访问使外部人员能关闭 MQIPT。

第 11 章 杂项功能

正常终止和故障条件

当 MQIPT 检测到 WebSphere MQ 通道关闭（正常或异常），它会传播通道关闭。如果管理员通过 MQIPT 关闭路由，则所有经过此路由的通道关闭。

MQIPT 提供了一个可选的空闲超时机制。如果 MQIPT 检测到一个通道的空闲时间超时，它将对这两个有问题的连接执行立即关机。

通道两端的两个 WebSphere MQ 系统观察这些异常终止条件（可能是网络失败或者是它们的伙伴终止了通道）。如果 MQIPT 不在使用中，则有问题的通道可以按它们的意愿重新启动和恢复（如果故障发生在协议不确定期间）。

消息安全

使用快速、非持久性 WebSphere MQ 消息时，如果 MQIPT 路由失败或重新启动，而这时正在传送 WebSphere MQ 消息，则该消息可能丢失。重新启动路由前，确保所有使用 MQIPT 路由的 WebSphere MQ 通道处于非活动状态。

请参阅 *MQSeries Intercommunication*，SC33-1872 以获取有关 WebSphere MQ 消息和通道的更多信息。

连接日志

MQIPT 提供了一个连接日志设施，它包含了成功的和不成功的连接尝试的列表。可使用 `ConnectionLog` 和 `MaxLogFileSize` 属性控制它。请参阅第 77 页的『Global 节参考信息』以获取更多信息。

每次启动 MQIPT 时，会创建一个新的连接日志。为了便于识别，文件名包含当前时间戳记，例如:

```
mqiptYYYYMMDDHHmmSS.log
```

其中

- `YYYY` 是年
- `MM` 是月
- `DD` 是日
- `HH` 是小时
- `mm` 是分钟
- `SS` 是秒

出于审计目的，将永不擦除这些日志文件。MQIPT 管理员负责管理这些文件和在不再需要它们的时候删除它们。

第 12 章 从先前版本升级

要将 MQIPT 从 V1.2 升级到 V1.3，请执行以下这些步骤:

1. 复制配置文件 mqipt.conf 和 client.conf。mqipt.conf 可在 MQIPT 主目录中找到，client.conf 可在 bin 子目录中找到。
2. 通过运行以下命令来停止 MQIPT:

   ```
   mqiptAdmin -stop
   ```

3. 如果您已将 MQIPT 安装为一个服务，则您必须在卸载 MQIPT 之前除去它:

   ```
   mqiptService -remove
   ```

4. 运行 MQIPT 的卸载程序。
5. 安装 MQIPT V1.3 后，将保存过的配置文件复制回它们原来的位置。
6. 建议您使用 MQIPT 管理 GUI 来管理对 MQIPT 的更改。来自 V1.2 的配置文件与 GUI 兼容。

有些实现需要您自己的组织控制下的本地 MQIPT 服务和您的客户机组织控制下的远程 MQIPT 服务。在这种情况下，同时迁移这两个 MQIPT 服务非常困难，但这不是 MQIPT 的问题。除非另有声明，旧版本的 MQIPT 兼容最新版本的 MQIPT。这使 MQIPT 的迁移过程更容易。

也可以在不先卸载 MQIPT 的情况下，升级 MQIPT 的核心。运行 MQIPT 所需的所有类存储在 MQipt.jar 文件中；您可以在另一台机器上安装最新版本的 MQIPT，并将安装的 MQipt.jar 文件复制到您正在使用的系统。对于运行管理 GUI 所需的类，也同样是这样。这些类包含在 guiadmin.jar 文件中。

新的配置选项

下列是在 V1.3 中新增的属性:

- IgnoreExpiredCRLs
- LDAP
- LDAPCacheTimeout
- LDAPIgnoreErrors
- LDAPSaveCRL
- LDAPServer1
- LDAPServer1Password
- LDAPServer1Port
- LDAPServer1Timeout
- LDAPServer1Userid
- LDAPServer2
- LDAPServer2Password
- LDAPServer2Port
- LDAPServer2Timeout

- LDAPServer2Userid
- RouteRestart
- SecurityExit
- SecurityExitName
- SecurityExitPath
- SecurityExitTimeout
- SSLClientSiteDN_C
- SSLClientSiteDN_CN
- SSLClientSiteDN_L
- SSLClientSiteDN_O
- SSLClientSiteDN_OU
- SSLClientSiteDN_ST
- SSLClientSiteLabel
- SSLServerSiteDN_C
- SSLServerSiteDN_CN
- SSLServerSiteDN_L
- SSLServerSiteDN_O
- SSLServerSiteDN_OU
- SSLServerSiteDN_ST
- SSLServerSiteLabel

关于所有这些属性的参考信息，请参阅第 73 页的『配置参考信息』。

第 13 章 在 Windows 上安装 internet pass-thru

本章描述如何在 Windows NT、Windows 2000 或 Windows XP 系统上安装 MQIPT:

- 『下载和安装文件』
- 第 46 页的『设置 internet pass-thru』
- 第 46 页的『从命令行启动 internet pass-thru』
- 第 47 页的『从命令行启动管理客户机』
- 第 47 页的『使用 Windows 服务控制程序』
- 第 48 页的『卸载作为 Windows 服务的 internet pass-thru』
- 第 48 页的『卸载 internet pass-thru』

下载和安装文件

MQIPT（MS81，类别 3 SupportPac™）可从 WebSphere MQ SupportPac Web 页面下载，地址如下:

`http://www.ibm.com/webspheremq/supportpacs`

按照说明进行下载。

打开命令提示符，并解包 `ms81_nt.zip` 到临时目录。运行`setup.exe`，并按照联机说明进行操作。

MQIPT 必须由具有管理员权限的用户安装。

MQIPT 包含下表中显示的文件，以及用于管理客户机 GUI 的文件，这些文件作为一个可单独安装的功能部件提供，它们将在下一张表中显示。

文件	用途
Readme.txt	没有包含在出版物中的最新消息
mqiptSample.conf	样本配置文件
ssl\sslSample.pfx	测试密钥环文件
ssl\sslSample.pwd	测试密钥环文件的密码文件
ssl\sslCAdefault.pfx	样本认证中心（CA）密钥环文件
ssl\sslCAdefault.pwd	样本 CA 密钥环文件的密码文件
ssl\KeyMan.zip	KeyMan 实用程序
exits\ SampleOneRouteExit.java	样本安全性出口
exits\ SampleOneRouteExit.conf	SampleOneRouteExit 的配置文件
exits\SampleRoutingExit.java	样本安全性出口
exits\SampleRoutingExit.conf	SampleRoutingExit 的配置文件
exits\SampleSecurityExit.java	样本安全性出口
lib\MQipt.jar	包含运行时、类和属性文件
lib\ADV_mqipt_normal. class	“normal” 节点的 Network Dispatcher 顾问程序
lib\ADV_mqipt_replace. class	“replace” 节点的 Network Dispatcher 顾问程序

文件	用途
lib\mqipt1414Sample.ssl	Network Dispatcher 顾问程序的样本触发器文件
bin\mqipt.bat	从命令行运行 MQIPT 的快捷方式
bin\mqiptAdmin.bat	停止 MQIPT 和刷新文件信息的快捷方式
bin\mqiptPW.bat	加密用于打开密钥环文件的密码
bin\mqiptservice.exe	用于添加或除去 MQIPT 或者从 Windows 服务控制管理器中添加或除去 MQIPT
bin\mqiptVersion.bat	显示 MQIPT 的版本号
web\MQIPTServlet.war	用于 servlet 版本的 Web 归档文件。
doc\<lang>\html\ <filename>.zip	HTML 格式的 *internet pass-thru* 手册的主文件。请参阅第 167 页的『文献目录』以获取有关软拷贝文档的更多信息。

与管理客户机 GUI 功能部件相关的文件有:

文件	用途
lib\guiadmin.jar	包含运行时、类和属性文件
bin\mqiptGui.bat	从命令行运行管理客户机的快捷方式
bin\customSample. 属性	用于定制外观（因此也定制管理客户机的可访问性）的样本文件

安装程序将系统 CLASSPATH 环境变量更新为 MQipt.jar 和 guiadmin.jar 文件的位置。

设置 internet pass-thru

第一次启动 MQIPT 之前，请将样本配置文件 mqiptSample.conf 复制为 mqipt.conf。请参阅第 69 页的第 19 章，『管理和配置 internet pass-thru』以获取进一步的信息。

从命令行启动 internet pass-thru

打开命令提示符，更改目录至 bin 目录并运行 mqipt。例如:

```
c:
cd \mqipt\bin
mqipt ..
```

您也可以从 Windows “开始” -> “程序” 菜单启动 MQIPT。

不使用任何选项运行 mqipt 脚本时，将对配置文件（mqipt.conf）使用缺省位置 “.”。要指定不同的位置:

```
mqipt <directory name>
```

消息将出现在显示 MQIPT 状态的控制台中。如果发生错误，请参阅第 145 页的『问题确定』。下列消息是 MQIPT 成功启动的消息示例:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在从 c:\mqipt\mqipt.conf 读取配置信息
MQCPI008 正在端口 1881 上侦听控制命令
MQCPI011 路径 c:\mqipt\logs 将用于存储日志文件
```

```
MQCPI006 路由 1418 已经启动并将转发消息到:
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....正在使用 MQ 协议
MQCPI078 路由 1418 用于连接请求准备就绪
MQCPI006 路由 1415 已经启动并将转发消息到:
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....正在使用 MQ 协议
MQCPI036 ....使用下列属性启用 SSL 客户机端:
MQCPI031 ......密码套件 <null>
MQCPI032 ......密钥环文件 c:\mqipt\KeyMan.pfx
MQCPI038 ......专有名称 CN=*Doe O=IBM OU=* L=* ST=* C=*
MQCPI078 路由 1415 用于连接请求准备就绪
```

第一次调用 MQIPT 时，将在 mqipt 主目录下自动创建下列子目录:

- 保存连接日志的 “logs” 目录
- 写入任何 First Failure Support Technology™（FFST™）和跟踪记录的 “errors” 目录

从命令行启动管理客户机

打开命令提示符，更改目录至 bin 目录并运行 mqiptGui。例如:

```
c:
cd \mqipt\bin
mqiptGui
```

为了使管理客户机能使用 SOCKS 代理穿过防火墙连接到外面的 MQIPT，请指定主机名或地址及端口号:

```
mqiptGui <socksHostName> <socksPort>
```

缺省 socksPort 为 1080。

管理客户机主窗口中出现的消息显示了管理客户机的状态。

使用 Windows 服务控制程序

提供了一个独立的服务控制程序 mqiptservice.exe，该程序能够管理 MQIPT 并将其作为 Windows 服务来启动。

mqiptservice.exe 使用下列命令行参数:

mqiptservice -install *path*

安装和注册服务，这样它在 Windows “服务” 面板上出现为手工服务。转至 “服务” 面板并将设置更改为 “自动” 以使 MQIPT 在系统启动时自动启动。安装此服务之后您必须重新引导 Windows。必须提供 path 参数，它是包含 mqipt.conf 配置文件的目录的全限定路径。如果此名称包含空格，请在路径名两边加上引号。

mqiptservice -remove

除去服务，使该服务从 “服务” 面板中消失。

mqiptservice ?

显示列出有效参数的美国英语帮助消息。

在同一命令中同时指定 install 和 remove 将引起错误。

Windows 内部调用 mqiptservice 程序时不使用参数。如果您从命令行调用它时不指定任何参数，程序将超时并返回一个错误。

当 MQIPT 服务启动时，所有活动的 MQIPT 路由都将启动。当它停止时，所有路由也跟着立即关闭。

注: 系统 PATH 环境变量必须包含 JNI 运行时库的位置。`jvm.dll` 文件可在 JDK 的 `client` 子目录中找到。

卸载作为 Windows 服务的 internet pass-thru

要将 MQIPT 作为服务卸载，您必须首先在 Windows “服务” 面板中停止它。然后打开命令提示符，转至 MQIPT 的 `bin` 子目录并输入:

```
mqiptservice -remove
```

卸载 internet pass-thru

从系统中卸载 MQIPT 之前，请按如上所述将它作为 Windows 服务除去。然后从 Windows “开始” 菜单运行卸载过程。

第 14 章 在 Sun Solaris 上安装 internet pass-thru

本章描述如何在 Sun Solaris 系统上安装 MQIPT:

下载和安装文件

MQIPT 可从 WebSphere MQ SupportPac Web 页面下载，地址如下:

```
http://www.ibm.com/webspheremq/supportpacs
```

按照说明进行下载。

以 root 用户身份登录，解压并解包 ms81_sol.tar.Z 到一个临时目录。运行 pkgadd 命令，如本示例中所示:

```
login root
cd /tmp
uncompress -fv ms81_sol.tar.Z
tar xvf ms81_sol.tar
pkgadd -d . mqipt
```

本示例假设 ms81_sol.tar.Z 位于 /tmp 目录中。

MQIPT 包含下表中显示的文件，其中包括用于管理客户机 GUI 的文件。

文件	用途
Readme.txt	没有包含在出版物中的最新消息
mqiptSample.conf	样本配置文件
ssl/sslSample.pfx	测试密钥环文件
ssl/sslSample.pwd	测试密钥环文件的密码文件
ssl/sslCAdefault.pfx	样本认证中心（CA）密钥环文件
ssl/sslCAdefault.pwd	样本 CA 密钥环文件的密码文件
ssl/KeyMan.zip	KeyMan 实用程序
exits/ SampleOneRouteExit.java	样本安全性出口
exits/ SampleOneRouteExit.conf	SampleOneRouteExit 的配置文件
exits/SampleRoutingExit.java	样本安全性出口
exits/SampleRoutingExit.conf	SampleRoutingExit 的配置文件
exits/SampleSecurityExit.java	样本安全性出口
lib/MQipt.jar	包含运行时、类和属性文件
lib/ADV_mqipt_normal. class	“normal” 节点的 Network Dispatcher 顾问程序

文件	用途
lib/ADV_mqipt_replace. class	“replace” 节点的 Network Dispatcher 顾问程序
lib/mqipt1414Sample.ssl	Network Dispatcher 顾问程序的样本触发器文件
bin/mqipt	从命令行运行 MQIPT 的快捷方式
bin/mqiptAdmin	停止 MQIPT 和刷新文件信息的快捷方式
bin/mqiptPW	加密用于打开密钥环文件的密码
bin/mqiptVersion	显示 MQIPT 的版本号
bin/mqiptService	用于安装 MQIPT 以使其能在系统启动时自动启动。
bin/mqiptEnv	定义 mqipt.jar 文件的位置，它只供其它脚本使用。
web/MQIPTServlet.war	用于 servlet 版本的 Web 归档文件。
doc/<lang>/html/ <filename>.zip	HTML 格式的 *internet pass-thru* 手册的主文件。请参阅第 167 页的『文献目录』以获取有关软拷贝文档的更多信息。
lib/guiadmin.jar	包含用于管理客户机 GUI 的运行时、类和属性文件
bin/mqiptGui	从命令行运行管理客户机 GUI 的快捷方式
bin/customSample. 属性	用于定制外观（因此也定制管理客户机的可访问性）的样本文件

设置 internet pass-thru

第一次启动 MQIPT 之前，请将样本配置文件 mqiptSample.conf 复制为 mqipt.conf。请参阅第 69 页的第 19 章，『管理和配置 internet pass-thru』以获取进一步的信息。

从命令行启动 internet pass-thru

以 root 用户身份登录，然后更改目录至 bin 目录。例如:

```
cd /opt/mqipt/bin
mqipt ..
```

不使用任何选项运行 mqipt 脚本时，将对配置文件（mqipt.conf）使用缺省位置 “.”。要指定不同的位置:

```
mqipt <directory name>
```

消息将出现在显示 MQIPT 状态的控制台中。如果发生错误，请参阅第 145 页的『问题确定』。下列消息是 MQIPT 成功启动的消息示例:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在从 /opt/mqipt/mqipt.conf 读取配置信息
MQCPI008 在端口 1881 上侦听控制命令
MQCPI011 路径 /opt/mqipt/logs 将用于存储日志文件
MQCPI006 路由 1418 已经启动并将转发消息到:
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....正在使用 MQ 协议
MQCPI078 路由 1418 用于连接请求准备就绪
MQCPI006 路由 1415 已经启动并将转发消息到:
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....正在使用 MQ 协议
MQCPI036 ....使用下列属性启用 SSL 客户机端:
```

```
MQCPI031 ......密码套件 <null>
MQCPI032 ......密钥环文件 /opt/mqipt/KeyMan.pfx
MQCPI038 ......专有名称 CN=*Doe O=IBM OU=* L=* ST=* C=*
MQCPI078 路由 1415 用于连接请求准备就绪
```

第一次调用 MQIPT 时，将在 mqipt 主目录下自动创建下列子目录:

- 保存连接日志的 “logs” 目录
- 写入任何 First Failure Support Technology（FFST）和跟踪记录的 “errors” 目录

自动启动 internet pass-thru

要在系统启动时自动启动 MQIPT，请运行 mqiptService 脚本。例如:

```
cd /opt/mqipt/bin
mqiptService -install
```

要使 MQIPT 不自动启动:

```
cd /opt/mqipt/bin
mqiptService -remove
```

从命令行启动管理客户机

打开命令提示符，更改目录至 bin 目录并运行 mqiptGui。例如:

```
cd /opt/mqipt/bin
mqiptGui
```

为了使管理客户机能穿过防火墙连接到外面的 MQIPT，请指定主机名或地址及端口号:

```
mqiptGui <socksHostName> <socksPort>
```

缺省 socksPort 为 1080。

管理客户机主窗口中出现的消息显示了管理客户机的状态。

卸载 internet pass-thru

从系统中卸载 MQIPT 之前，请不要让它自动启动（如『自动启动 internet pass-thru』中所述）。以 root 用户身份登录，然后运行 pkgrm 命令:

```
pkgrm mqipt
```

第 15 章 在 AIX 上安装 internet pass-thru

本章描述如何在 AIX 系统上安装 MQIPT:

下载和安装文件

MQIPT 可从 WebSphere MQ SupportPac Web 页面下载，地址如下:

```
http://www.ibm.com/webspheremq/supportpacs
```

请按照说明进行下载。

以 root 用户身份登录，解压并解包 ms81_aix.tar.Z 到一个临时目录。运行 installp 命令，如本示例中所示:

```
cd /tmp
uncompress -fv ms81_aix.tar.Z
tar xvf ms81_aix.tar
installp -d . -a mqipt-RT
```

本示例假设 ms81_aix.tar.Z 位于 /tmp 目录中。

MQIPT 包含下表中显示的文件，其中包括用于管理客户机 GUI 的文件。

文件	用途
Readme.txt	没有包含在出版物中的最新消息
mqiptSample.conf	样本配置文件
ssl/sslSample.pfx	测试密钥环文件
ssl/sslSample.pwd	测试密钥环文件的密码文件
ssl/sslCAdefault.pfx	样本认证中心（CA）密钥环文件
ssl/sslCAdefault.pwd	样本 CA 密钥环文件的密码文件
ssl/KeyMan.zip	KeyMan 实用程序
exits/ SampleOneRouteExit.java	样本安全性出口
exits/ SampleOneRouteExit.conf	SampleOneRouteExit 的配置文件
exits/SampleRoutingExit.java	样本安全性出口
exits/SampleRoutingExit.conf	SampleRoutingExit 的配置文件
exits/SampleSecurityExit.java	样本安全性出口
lib/MQipt.jar	包含运行时、类和属性文件
lib/ADV_mqipt_normal. class	“normal” 节点的 Network Dispatcher 顾问程序

文件	用途
lib/ADV_mqipt_replace. class	“replace” 节点的 Network Dispatcher 顾问程序
lib/mqipt1414Sample.ssl	Network Dispatcher 顾问程序的样本触发器文件
bin/mqipt	从命令行运行 MQIPT 的快捷方式
bin/mqiptAdmin	停止 MQIPT 和刷新文件信息的快捷方式
bin/mqiptPW	加密用于打开密钥环文件的密码
bin/mqiptVersion	显示 MQIPT 的版本号
bin/mqiptService	用于安装 MQIPT 以使其能在系统启动时自动启动。
bin/mqiptEnv	定义 mqipt.jar 文件的位置，它只供其它脚本使用。
web/MQIPTServlet.war	用于 servlet 版本的 Web 归档文件
doc/<lang>/html/ <filename>.zip	HTML 格式的 *internet pass-thru* 手册的主文件。请参阅第 167 页的『文献目录』以获取有关软拷贝文档的更多信息。
lib/guiadmin.jar	包含用于管理客户机 GUI 的运行时、类和属性文件
bin/mqiptGui	从命令行运行管理客户机的快捷方式
bin/customSample. 属性	用于定制外观（因此也定制管理客户机的可访问性）的样本文件

设置 internet pass-thru

第一次启动 MQIPT 之前，请将样本配置文件 mqiptSample.conf 复制为 mqipt.conf。请参阅第 69 页的第 19 章，『管理和配置 internet pass-thru』以获取进一步的信息。

从命令行启动 internet pass-thru

以 root 用户身份登录，然后更改目录至 bin 目录。例如:

```
cd /usr/opt/mqipt/bin
mqipt ..
```

不使用任何选项运行 mqipt 脚本时，将对配置文件（mqipt.conf）使用缺省位置 “.”。要指定不同的位置:

```
mqipt <directory name>
```

消息将出现在显示 MQIPT 状态的控制台中。如果发生错误，请参阅第 145 页的『问题确定』。下列消息是 MQIPT 成功启动的消息示例:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在从 /usr/opt/mqipt/mqipt.conf 读取配置信息
MQCPI008 在端口 1881 上侦听控制命令
MQCPI011 路径 /usr/opt/mqipt/logs 将用于存储日志文件
MQCPI006 路由 1418 已经启动并将转发消息到:
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....正在使用 MQ 协议
MQCPI078 路由 1418 用于连接请求准备就绪
MQCPI006 路由 1415 已经启动并将转发消息到:
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....正在使用 MQ 协议
MQCPI036 ....使用下列属性启用 SSL 客户机端:
```

```
MQCPI031 ......密码套件 <null>
MQCPI032 ......密钥环文件 /usr/opt/mqipt/KeyMan.pfx
MQCPI038 ......专有名称 CN=*Doe O=IBM OU=* L=* ST=* C=*
MQCPI078 路由 1415 用于连接请求准备就绪
```

第一次调用 MQIPT 时，将在 mqipt 主目录下自动创建下列子目录:

- 保存连接日志的 “logs” 目录
- 写入任何 First Failure Support Technology（FFST）和跟踪记录的 “errors” 目录

自动启动 internet pass-thru

要在系统启动时自动启动 MQIPT，请运行 mqiptService 脚本以在 inittab 中添加一个条目。例如:

```
cd /usr/opt/mqipt/bin
../mqiptService -install
```

要使 MQIPT 不自动启动，从 inittab 中除去其相应的条目:

```
cd /usr/opt/mqipt/bin
../mqiptService -remove
```

从命令行启动管理客户机

打开命令提示符，更改目录至 bin 目录并运行 mqiptGui。例如:

```
cd /usr/opt/mqipt/bin
../mqiptGui
```

为了使管理客户机能穿过防火墙连接到外面的 MQIPT，请指定主机名或地址及端口号:

```
mqiptGui <socksHostName> <socksPort>
```

缺省 socksPort 为 1080。

管理客户机主窗口中出现的消息显示了管理客户机的状态。

卸载 internet pass-thru

从系统中卸载 MQIPT 之前，请不要让它自动启动（如『自动启动 internet pass-thru』中所述）。以 root 用户身份登录，然后运行 installp 命令:

```
installp -u mqipt-RT
```

第 16 章 在 HP-UX 上安装 internet pass-thru

本章描述如何在 HP-UX 系统上安装 MQIPT:

下载和安装文件

MQIPT 可从 WebSphere MQ SupportPac Web 页面下载，地址如下:

```
http://www.ibm.com/webspheremq/supportpacs
```

按照说明进行下载。

以 root 用户身份登录，解压并解包 ms81_hp11.tar.Z 到一个临时目录。运行 swinstall 命令，如本示例中所示:

```
login root
cd /tmp
uncompress -fv ms81_hp11.tar.Z
tar xvf ms81_hp11.tar
swinstall -s /tmp MQIPT.MQIPT-RT
```

本示例假设 ms81_hp11.tar.Z 位于 /tmp 目录中。

MQIPT 包含下表中显示的文件，其中包括用于管理客户机 GUI 的文件。

文件	用途
Readme.txt	没有包含在出版物中的最新消息
mqiptSample.conf	样本配置文件
ssl/sslSample.pfx	测试密钥环文件
ssl/sslSample.pwd	测试密钥环文件的密码文件
ssl/sslCAdefault.pfx	样本认证中心（CA）密钥环文件
ssl/sslCAdefault.pwd	样本 CA 密钥环文件的密码文件
ssl/KeyMan.zip	KeyMan 实用程序
exits/ SampleOneRouteExit.java	样本安全性出口
exits/ SampleOneRouteExit.conf	SampleOneRouteExit 的配置文件
exits/SampleRoutingExit.java	样本安全性出口
exits/SampleRoutingExit.conf	SampleRoutingExit 的配置文件
exits/SampleSecurityExit.java	样本安全性出口
lib/MQipt.jar	包含运行时、类和属性文件
lib/ADV_mqipt_normal. class	“normal” 节点的 Network Dispatcher 顾问程序

文件	用途
lib/ADV_mqipt_replace. class	“replace” 节点的 Network Dispatcher 顾问程序
lib/mqipt1414Sample.ssl	Network Dispatcher 顾问程序的样本触发器文件
bin/mqipt	从命令行运行 MQIPT 的快捷方式
bin/mqiptAdmin	停止 MQIPT 和刷新文件信息的快捷方式
bin/mqiptPW	加密用于打开密钥环文件的密码
bin/mqiptVersion	显示 MQIPT 的版本号
bin/mqiptService	用于安装 MQIPT 以使其能在系统启动时自动启动。
bin/mqiptEnv	定义 mqipt.jar 文件的位置，它只供其它脚本使用。
bin/mqiptFork	用于在系统启动期间启动 MQIPT
web/MQIPTServlet.war	用于 servlet 版本的 Web 归档文件
doc/<lang>/html/ <filename>.zip	HTML 格式的 *internet pass-thru* 手册的主文件。请参阅第 167 页的『文献目录』以获取有关软拷贝文档的更多信息。
lib/guiadmin.jar	包含用于管理客户机 GUI 的运行时、类和属性文件
bin/mqiptGui	从命令行运行管理客户机 GUI 的快捷方式
bin/customSample. 属性	用于定制外观（因此也定制管理客户机的可访问性）的样本文件

设置 internet pass-thru

第一次启动 MQIPT 之前，请将样本配置文件 mqiptSample.conf 复制为 mqipt.conf。请参阅第 69 页的第 19 章，『管理和配置 internet pass-thru』以获取进一步的信息。

从命令行启动 internet pass-thru

以 root 用户身份登录，然后更改目录至 bin 目录。例如:

```
cd /opt/mqipt/bin
mqipt ..
```

不使用任何选项运行 mqipt 脚本时，将对配置文件（mqipt.conf）使用缺省位置 “.”。要指定不同的位置:

```
mqipt <directory name>
```

消息将出现在显示 MQIPT 状态的控制台中。如果发生错误，请参阅第 145 页的『问题确定』。下列消息是 MQIPT 成功启动的消息示例:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在从 /opt/mqipt/mqipt.conf 读取配置信息
MQCPI008 在端口 1881 上侦听控制命令
MQCPI011 路径 /opt/mqipt/logs 将用于存储日志文件
MQCPI006 路由 1418 已经启动并将转发消息到:
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....正在使用 MQ 协议
MQCPI078 路由 1418 用于连接请求准备就绪
MQCPI006 路由 1415 已经启动并将转发消息到:
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....正在使用 MQ 协议
MQCPI036 ....使用下列属性启用 SSL 客户机端:
```

```
MQCPI031 ......密码套件 <null>
MQCPI032 ......密钥环文件 /opt/mqipt/KeyMan.pfx
MQCPI038 ......专有名称 CN=*Doe O=IBM OU=* L=* ST=* C=*
MQCPI078 路由 1415 用于连接请求准备就绪
```

第一次调用 MQIPT 时，将在 mqipt 主目录下自动创建下列子目录:

- 保存连接日志的 “logs” 目录
- 写入任何 First Failure Support Technology（FFST）和跟踪记录的 “errors” 目录

自动启动 internet pass-thru

要在系统启动时自动启动 MQIPT，请运行 mqiptService 脚本。例如:

```
cd /opt/mqipt/bin
mqiptService -install
```

这假设 JDK 1.4 已经安装在名为 /opt/java1.4 的目录中。如果不是这样，编辑文件 mqipt.ske 并更改 PATH 变量以指向 JDK 的位置。运行 mqiptService -install 命令之前您必须应用此更改。

当 MQIPT 作为服务启动时，它将把 console.log 文件写到 logs 子目录。该子目录是在第一次运行 MQIPT 时创建的，因此在尝试将 MQIPT 作为服务启动之前必须至少启动它一次。

要使 MQIPT 不自动启动:

```
cd /opt/mqipt/bin
mqiptService -remove
```

从命令行启动管理客户机

打开命令提示符，更改目录至 bin 目录并运行 mqiptGui。例如:

```
cd /opt/mqipt/bin
mqiptGui
```

为了使管理客户机能穿过防火墙连接到外面的 MQIPT，请指定主机名或地址及端口号:

```
mqiptGui <socksHostName> <socksPort>
```

缺省 socksPort 为 1080。

管理客户机主窗口中出现的消息显示了管理客户机的状态。

卸载 internet pass-thru

从系统中卸载 MQIPT 之前，请不要让它自动启动（如『自动启动 internet pass-thru』中所述）。以 root 用户身份登录，然后运行 swremove 命令:

```
swremove MQIPT
```

第 17 章 在 Linux 上安装 internet pass-thru

本章描述如何在 Linux 系统上安装 MQIPT:

下载和安装文件

MQIPT 可从 WebSphere MQ SupportPac Web 页面下载，地址如下:

```
http://www.ibm.com/webspheremq/supportpacs
```

按照说明进行下载。

以 root 用户身份登录，解压并解包 ms81_linux.tar.z 到一个临时目录。运行 rpm 命令，如本示例中所示:

```
login root
cd /tmp
uncompress -fv ms81_linux.tar.z
tar xvf ms81_linux.tar
cd i386
rpm -i WebSphereMQ-IPT-1.3.0-0.i386.rpm
```

本示例假设 ms81_linux.tar.z 位于 /tmp 目录中。

MQIPT 包含下表中显示的文件，其中包括用于管理客户机 GUI 的文件。

文件	用途
Readme.txt	没有包含在出版物中的最新消息
mqiptSample.conf	样本配置文件
ssl/sslSample.pfx	测试密钥环文件
ssl/sslSample.pwd	测试密钥环文件的密码文件
ssl/sslCAdefault.pfx	样本认证中心（CA）密钥环文件
ssl/sslCAdefault.pwd	样本 CA 密钥环文件的密码文件
ssl/KeyMan.zip	KeyMan 实用程序
exits/ SampleOneRouteExit.java	样本安全性出口
exits/ SampleOneRouteExit.conf	SampleOneRouteExit 的配置文件
exits/SampleRoutingExit.java	样本安全性出口
exits/SampleRoutingExit.conf	SampleRoutingExit 的配置文件
exits/SampleSecurityExit.java	样本安全性出口
lib/libmqiptqos.so	用于 TQoS 的虚拟库

文件	用途
bin/mqiptQoS	用于使用实 TQoS 库
lib/MQipt.jar	包含运行时、类和属性文件
lib/ADV_mqipt_normal. class	“normal” 节点的 Network Dispatcher 顾问程序
lib/ADV_mqipt_replace. class	“replace” 节点的 Network Dispatcher 顾问程序
lib/mqipt1414Sample.ssl	Network Dispatcher 顾问程序的样本触发器文件
lib/libiptqos.so	用于服务质量支持的运行时库
bin/mqipt	从命令行运行 MQIPT 的快捷方式
bin/mqiptAdmin	停止 MQIPT 和刷新文件信息的快捷方式
bin/mqiptPW	加密用于打开密钥环文件的密码
bin/mqiptVersion	显示 MQIPT 的版本号
bin/mqiptService	用于安装 MQIPT 以使其能在系统启动时自动启动。
bin/mqiptEnv	定义 mqipt.jar 文件的位置，它只供其它脚本使用。
web/MQIPTServlet.war	用于 servlet 版本的 Web 归档文件
doc/<lang>/html/ <filename>.zip	HTML 格式的 *internet pass-thru* 手册的主文件。请参阅第 167 页的『文献目录』以获取有关软拷贝文档的更多信息。
lib/guiadmin.jar	包含用于管理客户机 GUI 的运行时、类和属性文件
bin/mqiptGui	从命令行运行管理客户机 GUI 的快捷方式
bin/customSample. 属性	用于定制外观（因此也定制管理客户机的可访问性）的样本文件

设置 internet pass-thru

第一次启动 MQIPT 之前，请将样本配置文件 mqiptSample.conf 复制为 mqipt.conf。请参阅第 69 页的第 19 章，『管理和配置 internet pass-thru』以获取进一步的信息。

从命令行启动 internet pass-thru

以 root 用户身份登录，然后更改目录至 bin 目录。例如:

```
cd /opt/mqipt/bin
mqipt ..
```

不使用任何选项运行 mqipt 脚本时，将对配置文件（mqipt.conf）使用缺省位置 “.”。要指定不同的位置:

```
mqipt <directory name>
```

消息将出现在显示 MQIPT 状态的控制台中。如果发生错误，请参阅第 145 页的『问题确定』。下列消息是 MQIPT 成功启动的消息示例:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在从 /opt/mqipt/mqipt.conf 读取配置信息
MQCPI008 正在端口 1881 上侦听控制命令
MQCPI011 路径 /opt/mqipt/logs 将用于存储日志文件
MQCPI006 路由 1418 已经启动并将转发消息到:
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....正在使用 MQ 协议
MQCPI078 路由 1418 用于连接请求准备就绪
MQCPI006 路由 1415 已经启动并将转发消息到:
```

```
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....正在使用 MQ 协议
MQCPI036 ....使用下列属性启用 SSL 客户机端:
MQCPI031 ......密码套件 <null>
MQCPI032 ......密钥环文件 /opt/mqipt/KeyMan.pfx
MQCPI038 ......专有名称 CN=*Doe O=IBM OU=* L=* ST=* C=*
MQCPI078 路由 1415 用于连接请求准备就绪
```

第一次调用 MQIPT 时，将在 mqipt 主目录下自动创建下列子目录:

- 保存连接日志的 “logs” 目录
- 写入任何 First Failure Support Technology（FFST）和跟踪记录的 “errors” 目录

自动启动 internet pass-thru

要在系统启动时自动启动 MQIPT，请运行 mqiptService 脚本。例如:

```
cd /opt/mqipt/bin
mqiptService -install
```

当 MQIPT 作为服务启动时，它将把 console.log 文件写到 logs 子目录。该子目录是在第一次运行 MQIPT 时创建的，因此在尝试将 MQIPT 作为服务启动之前必须至少启动它一次。

要使 MQIPT 不自动启动:

```
cd /opt/mqipt/bin
mqiptService -remove
```

从命令行启动管理客户机

打开命令提示符，更改目录至 bin 目录并运行 mqiptGui。例如:

```
cd /opt/mqipt/bin
mqiptGui
```

为了使管理客户机能穿过防火墙连接到外面的 MQIPT，请指定主机名或地址及端口号:

```
mqiptGui <socksHostName> <socksPort>
```

缺省 socksPort 为 1080。

管理客户机主窗口中出现的消息显示了管理客户机的状态。

卸载 internet pass-thru

从系统中卸载 MQIPT 之前，请不要让它自动启动（如『自动启动 internet pass-thru』中所述）。以 root 用户身份登录，然后运行 swremove 命令:

```
rpm -e WebSphereMQ-IPT-1.3.0-0
```

第 18 章 通用 UNIX 安装

以 tar 文件的形式提供所有公共 MQIPT 文件的磁盘映象以供一般使用。此文件的目的是允许 MQIPT 安装在那些使用其自己的安装映象的 MQIPT 所不支持的 UNIX 平台上。目的是允许 tar 文件以尽可能小的更改解包至指定的位置，允许在任何支持 Java 1.4 的平台上实现 MQIPT。可能需要更改在 bin 子目录中找到的 mqiptEnv 脚本以反映已安装文件的位置。

- 『下载和安装文件』
- 第 66 页的『设置 internet pass-thru』
- 第 66 页的『从命令行启动 internet pass-thru』
- 第 67 页的『自动启动 internet pass-thru』
- 第 67 页的『从命令行启动管理客户机』
- 第 67 页的『卸载 internet pass-thru』

下载和安装文件

MQIPT 可从 WebSphere MQ SupportPac Web 页面下载，地址如下:

```
http://www.ibm.com/webspheremq/supportpacs
```

按照说明进行下载。

作为 root 用户登录，将 ms81.tar 解包至目标目录，如本示例中所示:

```
login root
cd /
mkdir mqipt
cd mqipt
cp /tmp/ms81.tar /mqipt/.
tar xvf ms81.tar
```

本示例假设 ms81.tar 下载到 /tmp 目录。

MQIPT 包含下表中显示的文件，其中包括用于管理客户机 GUI 的文件。

文件	用途
Readme.txt	没有包含在出版物中的最新消息
mqiptSample.conf	样本配置文件
ssl/sslSample.pfx	测试密钥环文件
ssl/sslSample.pwd	测试密钥环文件的密码文件
ssl/sslCAdefault.pfx	样本认证中心（CA）密钥环文件
ssl/sslCAdefault.pwd	样本 CA 密钥环文件的密码文件
ssl/KeyMan.zip	KeyMan 实用程序
exits/ SampleOneRouteExit.java	样本安全性出口
exits/ SampleOneRouteExit.conf	SampleOneRouteExit 的配置文件
exits/SampleRoutingExit.java	样本安全性出口
exits/SampleRoutingExit.conf	SampleRoutingExit 的配置文件

文件	用途
exits/SampleSecurityExit.java	样本安全性出口
lib/MQipt.jar	包含运行时、类和属性文件
lib/ADV_mqipt_normal. class	“normal” 节点的 Network Dispatcher 顾问程序
lib/ADV_mqipt_replace. class	“replace” 节点的 Network Dispatcher 顾问程序
lib/mqipt1414Sample.ssl	Network Dispatcher 顾问程序的样本触发器文件
bin/mqipt	从命令行运行 MQIPT 的快捷方式
bin/mqiptAdmin	停止 MQIPT 和刷新文件信息的快捷方式
bin/mqiptPW	加密用于打开密钥环文件的密码
bin/mqiptVersion	显示 MQIPT 的版本号
bin/mqiptService	用于安装 MQIPT 以使其能在系统启动时自动启动。
bin/mqiptEnv	定义 mqipt.jar 文件的位置，它只供其它脚本使用。
web/MQIPTServlet.war	用于 servlet 版本的 Web 归档文件
doc/<lang>/html/ <filename>.zip	HTML 格式的 *internet pass-thru* 手册的主文件。请参阅第 167 页的『文献目录』以获取有关软拷贝文档的更多信息。
lib/guiadmin.jar	包含用于管理客户机 GUI 的运行时、类和属性文件
bin/mqiptGui	从命令行运行管理客户机的快捷方式
bin/customSample. 属性	用于定制外观（因此也定制管理客户机的可访问性）的样本文件

设置 internet pass-thru

第一次启动 MQIPT 之前，请将样本配置文件 mqiptSample.conf 复制为 mqipt.conf。请参阅第 69 页的第 19 章，『管理和配置 internet pass-thru』以获取进一步的信息。

本示例假设 MQIPT 将解包至称为 mqipt 的目录。您必须使用运行时库的新位置更新 mqiptEnv 脚本。MQIPT_CP 变量的缺省值为:

```
MQIPT_CP=/opt/mqipt/lib/MQipt.jar:/opt/mqipt/lib/guiadmin.jar
```

对于我们的示例，这必须更改为:

```
MQIPT_CP=/mqipt/opt/mqipt/lib/MQipt.jar:/mqipt/opt/mqipt/lib/guiadmin.jar
```

在使用任何运行时脚本之前还必须先更新它们，并更改 mqiptEnv 脚本位置的全限定路径名。因此，例如，在使用 mqipt 脚本之前，编辑它并将注释 Get classpath 后的语句从:

```
/opt/mqipt/bin/mqiptEnv
```

更改为:

```
/mqipt/opt/mqipt/bin/mqiptEnv
```

从命令行启动 internet pass-thru

以 root 用户身份登录，然后更改目录至 bin 目录。例如:

```
cd /mqipt/opt/mqipt/bin
mqipt ..
```

不使用任何选项运行 mqipt 脚本时，将对配置文件（mqipt.conf）使用缺省位置 “.”。要指定不同的位置:

```
mqipt <directory name>
```

消息将出现在显示 MQIPT 状态的控制台中。如果发生错误，请参阅第 145 页的『问题确定』。下列消息是 MQIPT 成功启动的消息示例:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在从 /mqipt/opt/mqipt/mqipt.conf 读取配置信息
MQCPI008 正在端口 1881 上侦听控制命令
MQCPI011 路径 /mqipt/opt/mqipt/logs 将用于存储日志文件
MQCPI006 路由 1418 已经启动并将转发消息到:
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....正在使用 MQ 协议
MQCPI078 路由 1418 用于连接请求准备就绪
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....正在使用 MQ 协议
MQCPI036 ....使用下列属性启用 SSL 客户机端:
MQCPI031 ......密码套件 <null>
MQCPI032 ......密钥环文件 /mqipt/opt/mqipt/KeyMan.pfx
MQCPI038 ......专有名称 CN=*Doe O=IBM OU=* L=* ST=* C=*
MQCPI078 路由 1415 用于连接请求准备就绪
```

第一次调用 MQIPT 时，将在 mqipt 主目录下自动创建下列子目录:

- 保存连接日志的 “logs” 目录
- 写入任何 First Failure Support Technology（FFST）和跟踪记录的 “errors” 目录

自动启动 internet pass-thru

自动启动服务是平台特定的。单独提供了 mqiptService 脚本，它作为如何在 Sun Solaris 系统上完成此操作的示例。取决于系统要求，使用平台特定的实用程序将 MQIPT 安装为系统服务可能更为容易。

从命令行启动管理客户机

打开命令提示符，更改目录至 bin 目录并运行 mqiptGui。例如:

```
cd /mqipt/opt/mqipt/bin
../mqiptGui
```

为了使管理客户机能穿过防火墙连接到外面的 MQIPT，请指定主机名或地址及端口号:

```
mqiptGui <socksHostName> <socksPort>
```

缺省 socksPort 为 1080。

管理客户机主窗口中出现的消息显示了管理客户机的状态。

卸载 internet pass-thru

由于 MQIPT 不是使用系统可安装映象安装的，因此，可通过删除它安装到的目录结构来卸载它。

如果 MQIPT 配置为作为系统服务运行，卸载代码之前先除去服务。

第 19 章 管理和配置 internet pass-thru

您可以通过更改配置文件 mqipt.conf 来配置 MQIPT。建议您使用管理客户机来执行配置，您也可以使用选择的编辑器来进行配置。下面对这两种技术及其相关的参考信息都进行了描述:

使用 internet pass-thru 管理客户机

您可以使用管理客户机配置和更新一个或多个 MQIPT。它显示 MQIPT 的全局属性和路由特定的属性。

注意，管理客户机不需要 Java 1.4 作为先决条件。

唯一在管理客户机上本地存储的数据是 MQIPT 列表，它们存储在称为 client.conf 的文件中。在管理客户机中显示全局和路由属性之前，始终从 MQIPT 中检索它们。

启动管理客户机

使用 mqiptGui 脚本启动管理客户机，该脚本可在 MQIPT 的 bin 子目录中找到。请参阅每个平台的安装章节以获取关于启动管理客户机的信息。

第一次启动管理客户机时，将显示一个对话框，它将提示您输入 MQIPT 的连接信息。所需的信息包含:

MQIPT 名称
用于描述此 MQIPT 的名称。尽管此信息并不是必需的，但建议您提供此信息。

网络地址
MQIPT 驻留所在系统的地址 – 名称服务器能够识别的名称（点分十进制地址）或 localhost（如果 MQIPT 与客户机位于同一台机器上）。

命令端口
MQIPT 侦听命令所在的端口号。

超时
这是管理客户机将等待到 MQIPT 连接的秒数。尽可能地降低此值，以减少窗口的刷新时间。

访问密码
与 MQIPT 通信时使用的密码。只有当密码检查有效时，才填充此字段。（如果 MQIPT 配置文件中提供了 AccessPW，并且该 AccessPW 为非空字符串时，则密码检查有效。）

保存密码
如果将此复选框留空，则在会话的持续时间内或在除去此 MQIPT 之前将记住此密码。如果选择此复选框，则将为将来的会话保存此密码。

图 8. 第一次访问 MQIPT 的窗口

管理 MQIPT

一次只能更新一个 MQIPT，因此，如果从列表中选择了另一个 MQIPT，则必须在继续之前应用所有未完成的更改。在使用“应用”菜单选项之前，对任何属性所做的更改都不会影响 MQIPT。

从列表中选择一个 MQIPT 时将从此 MQIPT 中检索全局和路由属性。如果此 MQIPT 不在运行，或者指定了不正确的“命令端口”，将发出一条错误消息。您可以从“连接”菜单选项中对“主机名”和“命令端口”进行更改。

双击列表中的一个 MQIPT 后将显示路由列表。选择一个路由以显示其属性。您可以根据您的要求定制这些属性。

当应用这些更改时，会对此配置文件做上时间戳记并将其发送回 MQIPT，而更改也将立即生效。任何现有注释行都将丢失。

您可以使用“添加路由”菜单选项来添加路由。使用时将为此新路由显示一组由全局属性定义的缺省属性。

属性的继承

有一些分层的方法，可以根据这些方法在管理客户机中设置 MQIPT 和路由的属性:

1. 每个属性具有缺省值，如果在配置文件中未提及此属性，或者管理客户机中的用户操作未明确设置此属性，则假设此缺省值。
2. 在 MQIPT 上设置的全局属性由该 MQIPT 上的每个路由假设，除非有与此相反的特定路由信息。在配置文件中，这意味着将 global 节中设置的属性传播到所有路由，除非在 route 节中设置其它属性。将 MQIPT 上由管理客户机用户设置的属性传播到所有路由，除非在路由上明确设置某个属性。
3. 无论是缺省值还是全局设置，路由支持违反该路由的任何设置。

文件菜单选项

当选择“文件”菜单时，将显示与管理此树相关的大部分选项。

添加 MQIPT

调出与第一次使用客户机时出现的相同对话框，在第 69 页的『启动管理客户机』中描述。

除去 MQIPT

仅从管理客户机的树中除去当前突出显示的 MQIPT。它不影响 MQIPT 的运行。

保存配置

将树的 MQIPT 节点保存到管理客户机的配置文件，以便可以在下一次启动时读回这些节点。只保存 MQIPT 节点。始终从 MQIPT 检索全局和路由属性。

退出

停止管理客户机运行。但是，管理客户机首先检查树或当前 MQIPT 是否已更改；如果其中之一或两者都已更改，则将为您显示一个或几个对话框，询问您是要保存客户机，还是将更改应用到 MQIPT，还是两者都执行。

MQIPT 菜单选项

连接

更改 MQIPT 的访问参数。在树视图中反映这些更改。它调出类似于第 69 页的『启动管理客户机』中描述的窗口。

密码

更改远程 MQIPT 的密码属性。此操作调出一密码对话框，您应该在其中填写下列条目:

- **当前密码**: 作为对不适当的使用的检查，您必须演示在更改之前知道当前密码。如果当前无有效密码，则此字段保留为空白。
- **新密码**: 如果您要停止在此 MQIPT 上使用这些密码，这是新密码或空白。
- **再次输入新密码**: 通过要求您重复相同的信息以防止您在前一字段中输入错误。
- **保存密码**: 用于确定是否本地保存新密码，以及此 MQIPT 的其它访问属性。

添加路由

将路由添加到所选的 MQIPT。请参阅第 72 页的图 9 以获取详细信息。每个路由对于 MQIPT 必须都有一个唯一的侦听器端口。

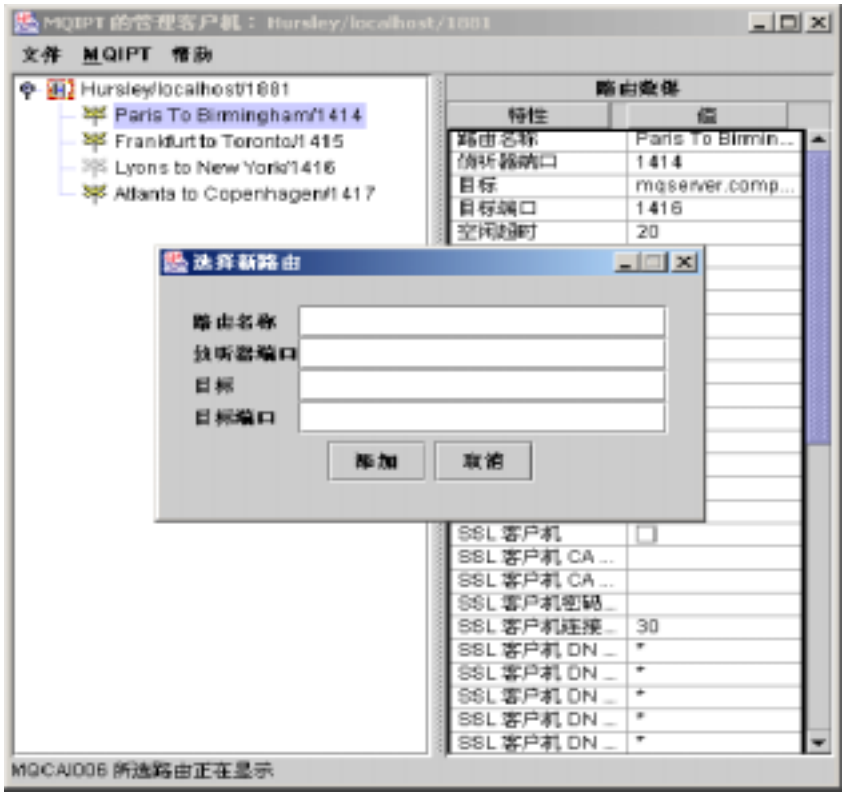

图 9. 添加路由

删除路由

从 MQIPT 删除所选的路由。此删除不影响 MQIPT，除非使用“应用”菜单选项。

应用

当您满意对 MQIPT 配置所做的更改时，此选项将一个新的配置文件发送到保存它的 MQIPT。新设置将立即有效。

刷新

从所选的 MQIPT 读配置文件，并刷新此显示。

停止

将停止命令发送到 MQIPT 以使其停止运行。此命令后，您将与 MQIPT 失去联系。忽略此命令，除非打开全局属性 RemoteShutdown。

可以按照与 MQIPT 全局信息相同的方式更新路由信息。当您更改路由的任何属性时，您必须在它们生效之前应用这些更改。您可以通过选择“MQIPT／应用”菜单选项或在提示您保存配置时回答“是”来执行此操作。

帮助菜单选项

帮助

使用 Netscape 显示关于如何使用管理客户机的信息，并在左侧窗格中选择“管理和配置 internet pass-thru”。在使用管理客户机前，您必须解压缩在 `<lang>/html` 子目录中查找到的文件。

关于

显示一弹出的窗口，其中有管理客户机版本的信息。

使用 internet pass-thru 行方式命令

如果您选择不使用管理客户机，您可以使用行方式命令来管理和配置 internet pass-thru。

使用行方式命令管理 internet pass-thru

使用您的选项编辑器，更改配置文件 mqipt.conf 以满足您的要求。请参阅『配置参考信息』以获取您可以更改的属性列表。

如果 mqipt.conf 的 global 节指定 CommandPort 的值，则 MQIPT 在此端口上侦听下列 ASCII 管理命令:

```
mqiptAdmin -refresh {hostname {port} }      sends the refresh command
mqiptAdmin -stop    {hostname {port} }      sends the stop command
```

此 mqiptAdmin 脚本在 bin 子目录中。

如果不提供，则主机名缺省为 localhost，端口缺省为 1881。

STOP

MQIPT 关闭所有连接，停止侦听进入连接，然后退出。使用管理客户机的 “MQIPT／停止” 菜单选项具有相同的效果。忽略此命令，除非 mqipt.conf 文件指定 RemoteShutDown=true。

REFRESH

MQIPT 重新读取 mqipt.conf。如果它发现:

- 任何当前活动的路由现在标记为非活动（或正在完全缺少），则它将关闭这些路由，并停止侦听这些路由上的进入连接。
- 在当前不运行的配置文件中标记为活动的任何路由，则它将启动这些路由。
- 当前运行的路由的配置参数已更改，则它将更改的值应用到那些路由。只要有可能发生的地方（例如，对跟踪设置的更改），它将在不中断运行连接的情况下执行。对于一些参数更改（例如，对目标的更改），MQIPT 已在实现更改并重新启动路由之前关闭所有连接。

倘若管理客户机未更改任何 MQIPT 设置，则使用管理客户机的 “MQIPT／应用” 菜单选项具有相同的效果。

在 Windows 上，还可以从 “开始” -> “程序” 菜单打开这些管理功能。

配置参考信息

MQIPT 使用称为 mqipt.conf 的配置文件来定义路由和控制 MQIPT 服务器的操作。这个文件包括一组节。存在一个 global 节以及已通过 MQIPT 定义的每个路由的附加节。

每个节包含名称／值属性对。一些属性只能出现在 global 节中，一些只能出现在 route 节中，还有一些即可以出现在 route 节中也可以出现在 global 节中。如果属性同时出现在 route 节和 global 节中，则 route 节中的属性值覆盖全局值，但仅限于所讨论的路由。这样，可使用 global 节来建立用于那些未在个别 route 节中设置的属性的缺省值。

global 节以包含字符 [global] 的行开始，并在第一个 route 节开始时结束。global 节必须放在文件中的所有 route 节之前。每个 route 节以包含字符 [route] 的行开始，并在下一个 route 节开始或配置文件结束时结束。

忽略任何未经识别的关键字名称（即，任何名称／值对，其中的名称不是在此文档中定义的名称之一）。如果出现在 route 节中的名称／值对具有一个可识别名，但具有无效的值（例如，`MinConnectionThreads=x` 或 `HTTP=unsure`），将禁用此路由（即，它不侦听任何进入连接）。如果出现在 global 节中的名称／值具有一个可识别名，但具有无效的值，则将禁用所有路由并且 MQIPT 不启动。其中的属性列为取出值 `true` 和 `false`，可以将大写和小写混合使用。

可通过编辑 mqipt.conf 文件或使用管理客户机 GUI 对任何属性进行更改。要应用任何更改，管理员可以从管理客户机 GUI 或使用 mqiptAdmini 脚本发出刷新命令。

如果其它属性已启用，则对某些属性的更改将只引起路由重新启动。例如，对 HTTP 属性的任何更改仅当 HTTP 属性也启用时才有效。

当路由重新启动时，已有的连接将终止。要覆盖这个行为，将 RouteRestart 属性设置为 false。这将阻止路由重新启动，允许已有的连接保持活动直至重新启用 RouteRestart 属性。

要获取有关如何设置某些简单配置的信息，请参阅第 91 页的第 20 章，『internet pass-thru 入门』。有关样本配置，请参阅 MQIPT 主目录中的 `mqiptSample.conf` 文件。

属性总结

表 3 显示:

- 所有属性
- 属性应用于 global 节还是应用于 route 节，还是应用于两者
- 如果 route 节和 global 节中都缺少某一属性，则使用缺省值

表 3. 配置属性的总结

属性名	全局	路由	缺省
AccessPW	是	否	<null>
Active	是	是	true
ClientAccess	是	是	false
CommandPort	是	否	<null>
ConnectionLog	是	否	true
Destination	否	是	<null>
DestinationPort	否	是	1414
HTTP[6,7]	是	是	false
HTTPChunking[1]	是	是	false
HTTPProxy[1]	是	是	<null>
HTTPProxyPort[1]	是	是	8080
HTTPS[1]	是	是	false
HTTPServer[1]	是	是	<null>
HTTPServerPort[1]	是	是	<null>
IdleTimeout	是	是	0
IgnoreExpiredCRLs	是	是	false
LDAP	是	是	false
LDAPIgnoreErrors[10]	是	是	false

表 3. 配置属性的总结 (续)

属性名	全局	路由	缺省
LDAPCacheTimeout[10]	是	是	24
LDAPSaveCRL[10]	是	是	false
LDAPServer1[10]	是	是	<null>
LDAPServer1Port[10]	是	是	389
LDAPServer1Userid[10]	是	是	<null>
LDAPServer1Password[10]	是	是	<null>
LDAPServer1Timeout[10]	是	是	0
LDAPServer2[10]	是	是	<null>
LDAPServer2Port[10]	是	是	389
LDAPServer2Userid[10]	是	是	<null>
LDAPServer2Password[10]	是	是	<null>
LDAPServer2Timeout[10]	是	是	0
ListenerPort	否	是	<null>
LocalAddress	是	是	<null>
LogDir（这仅对 MQIPTServlet 有效）	否	否	<null>
MaxConnectionThreads	是	是	100
MaxLogFileSize	是	否	50
MinConnectionThreads	是	是	5
Name	否	是	<null>
NDAdvisor	是	是	false
NDAdvisorReplaceMode[4]	是	是	false
OutgoingPort	否	是	0
QMgrAccess	是	是	true
QoS（只能在 Linux 上使用）	是	是	false
QosToCaller[9]	是	是	1
QosToDest[9]	是	是	1
RemoteShutdown	是	否	false
RouteRestart	是	是	true
SecurityExit	是	是	false
SecurityExitName[11]	是	是	<null>
SecurityExitPath[11]	是	是	<ipthome>\exits
SecurityExitTimeout[11]	是	是	5
SecurityManager	是	否	false
SecurityManagerPolicy	是	否	<null>
ServletClient[1]	是	是	false
SocksClient	是	是	false
SocksProxyHost[8]	是	是	<null>
SocksProxyPort[8]	是	是	1080
SocksServer[7]	是	是	false

表 3. 配置属性的总结 (续)

属性名	全局	路由	缺省
SSLClient	是	是	false
SSLClientCAKeyRing[2]	是	是	<null>
SSLClientCAKeyRingPW[2]	是	是	<null>
SSLClientCipherSuites[2]	是	是	<null>
SSLClientConnectTimeout[2]	是	是	30
SSLClientDN_C[2]	是	是	"*"[5]
SSLClientDN_CN[2]	是	是	"*"[5]
SSLClientDN_L[2]	是	是	"*"[5]
SSLClientDN_O[2]	是	是	"*"[5]
SSLClientDN_OU[2]	是	是	"*"[5]
SSLClientDN_ST[2]	是	是	"*"[5]
SSLClientKeyRing[2]	是	是	<null>
SSLClientKeyRingPW[2]	是	是	<null>
SSLClientSiteDN_C[2]	是	是	"*"[5]
SSLClientSiteDN_CN[2]	是	是	"*"[5]
SSLClientSiteDN_L[2]	是	是	"*"[5]
SSLClientSiteDN_O[2]	是	是	"*"[5]
SSLClientSiteDN_OU[2]	是	是	"*"[5]
SSLClientSiteDN_ST[2]	是	是	"*"[5]
SSLClientSiteLabel[2]	是	是	<null>
SSLProxyMode	是	是	false
SSLServer[6]	是	是	false
SSLServerAskClientAuth[3]	是	是	false
SSLServerCAKeyRing[3]	是	是	<null>
SSLServerCAKeyRingPW[3]	是	是	<null>
SSLServerCipherSuites[3]	是	是	<null>
SSLServerDN_C[3]	是	是	"*"[5]
SSLServerDN_CN[3]	是	是	"*"[5]
SSLServerDN_L[3]	是	是	"*"[5]
SSLServerDN_O[3]	是	是	"*"[5]
SSLServerDN_OU[3]	是	是	"*"[5]
SSLServerDN_ST[3]	是	是	"*"[5]
SSLServerKeyRing[3]	是	是	<null>
SSLServerKeyRingPW[3]	是	是	<null>
SSLServerSiteDN_C[3]	是	是	"*"[5]
SSLServerSiteDN_CN[3]	是	是	"*"[5]
SSLServerSiteDN_L[3]	是	是	"*"[5]
SSLServerSiteDN_O[3]	是	是	"*"[5]
SSLServerSiteDN_OU[3]	是	是	"*"[5]
SSLServerSiteDN_ST[3]	是	是	"*"[5]

表 3. 配置属性的总结 (续)

属性名	全局	路由	缺省
SSLServerSiteLabel[3]	是	是	<null>
Trace	是	是	0
UriName（请参阅第 第 88 页的『UriName』 页以获取关于缺省设置的详细信息。）[1]	是	是	

注:

1. 将 HTTP 设置为 true 以使这些属性有效。
2. 将 SSLClient 设置为 true 以使这些属性有效。
3. 将 SSLServer 设置为 true 以使这些属性有效。
4. 将 NDAdvisor 设置为 true 以使这些属性有效。
5. “*”符号代表通配符。
6. HTTP 和 SSLServer 不能一起使用。HTTP 属性仅用于定义转发连接。自动检测 ListenerPort 上的进入数据，设置 SSLServer 将导致运行时异常。
7. HTTP 和 SocksServer 不能一起使用。HTTP 属性仅用于定义转发连接。自动检测 ListenerPort 上的进入数据，设置 SocksServer 将导致运行时异常。
8. 将 SocksClient 设置为 true 以使这些属性有效。
9. 将 QoS 设置为 true 以使这些属性有效。
10. 将 LDAP 设置为 true 以使这些属性有效。
11. 将 SecurityExit 设置为 true 以使这些属性有效。

Global 节参考信息

global 节可以包含下列属性和第 78 页的『route 节参考信息』中的所有属性，除了 ListenerPort、Destination、DestinationPort、Name 和 OutgoingPort。

AccessPW

当管理控制器将命令发送给 MQIPT 时所使用的密码。如果此属性不存在或设置为空白，则不进行任何检查。

CommandPort

MQIPT 从 mqiptAdmin 实用程序或管理客户机侦听配置命令所在的 TCP/IP 端口。您可按照与任何其它属性相同的方法从管理客户机更改命令端口。请注意，不要更改连接属性。当您将新设置应用到 MQIPT 时，管理客户机自动更改连接属性。

如果 CommandPort 属性不存在，MQIPT 不侦听配置命令。如果您想在此命令端口上侦听，建议您使用 1881。管理客户机不具有 CommandPort 的缺省值，但 1881 是您使用行方式命令时的缺省值。

ConnectionLog

true 或 false。当为 true 时，MQIPT 将所有连接尝试（成功或相反）记录到 logs 子目录中，并将断开的事件记录到文件 mqiptYYYYMMDDHHmmSS.log 中。缺省值为 true。当此属性从 true 更改为 false，MQIPT 将关闭现有连接日志并创建新的连接日志。当属性复位为 true 时，将使用新的连接日志。

MaxLogFileSize

连接日志文件的最大大小（以 KB 指定）。当文件大小超过此最大值时，将制作备份

mqipt.back，并开始使用新文件。只保留一个备份文件；每次当主日志文件填满时，将擦除较早的备份。缺省值为 50，允许的最小值为 5。

RemoteShutDown

true 或 false。当为 true（并存在命令端口）时，无论何时在命令端口上接收到 STOP 命令，MQIPT 将关闭。缺省值为 false。

SecurityManager

将此属性设置为 true 以为 MQIPT 的这个实例启用 Java 安全性管理器。这依赖于授权的正确许可权。请参阅第 29 页的『Java 安全性管理器』以获取更多信息。此属性的缺省值为 false。

SecurityManagerPolicy

策略文件的全限定文件名。如果未设置此属性，则仅使用缺省系统和用户策略文件。如果 Java 安全性管理器已启用，则禁用并重新启用 Java 安全性管理器后，对此属性的更改才能生效。

route 节参考信息

此 route 节可包含下列属性:

Active

仅当 Active 值设置为 true 时，路由才接受进入连接。这意味着您可以通过设置 Active=false 来暂时地切断对目标的访问，而不必从配置文件中删除 route 节。如果您将此属性更改为 false，当发出 REFRESH 命令时，此路由将停止。此路由的所有连接将终止。

ClientAccess

仅当 ClientAccess 值设置为 true 时，路由才允许进入客户机通道连接。请注意，您或许可以配置 MQIPT 以仅接受客户机请求、仅接受队列管理器请求，或同时接受这两种请求。一起使用此属性和 QMgrAccess 属性。如果您将此属性更改为 false，当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

Destination

此路由将与其连接的队列管理器（或后继 MQIPT）的主机名（或点分十进制 IP 地址）。每个 route 节**必须**包含一个显式 Destination 值。允许您具有几个指向同一 Destination 的 route 节。如果对此属性的某一更改影响路由，当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

DestinationPort

此路由将连接的 Destination 主机上的端口。它适用于多个路由指向相同的 Destination 和 DestinationPort 组合。每个 route 节**必须**包含一个显式 DestinationPort 值。如果对此属性的某一更改影响路由，当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

HTTP

将它设置为 true，以便使路由负责发出出站 HTTP 隧道请求（即，通过 HTTP 与另一个 MQIPT 进行通信）。设置为 false，使路由定向于 WebSphere MQ 队列管理器。如果您更改此属性，当发出 REFRESH 命令时，路由将停止并重新启动。到此路由的所有连接将终止。要使用 HTTP 分块，请将此属性设置为 true。此属性不能与下列属性一起使用:

- QoS

- SocksClient
- SSLClient
- SSLProxyMode

HTTPChunking

将它设置为 true，以便使路由负责使用带分块的 HTTP 隧道来发出出站请求。HTTP 属性也必须设置为 true。当您不使用 HTTP 分块时，将其设置为 false。如果您更改此属性（且 HTTP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。到此路由的所有连接将终止。

HTTPProxy

此路由的所有连接所使用的 HTTP 代理的主机名（或点分十进制 IP 地址）。如果还定义了 HTTPServer，则向 HTTPProxy 发出 CONNECT 请求（而不是通常的 POST 请求）。如果您更改此属性（且 HTTP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

HTTPProxyPort

将在 HTTP 代理上使用的端口地址。缺省值为 8080，除非 HTTPS 设置为 true，且不存在 HTTPServer，则缺省值为 443。如果您更改此属性（且 HTTP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。到此路由的所有连接将终止。

HTTPServer

此路由的所有连接所使用的 HTTP 服务器的主机名（或点分十进制 IP 地址）。如果您更改此属性（且 HTTP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

HTTPS

启用此属性以生成 HTTPS 请求。HTTP 属性也必须是启用的。如果您更改此属性（且 HTTP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。到此路由的所有连接将终止。

HTTPServerPort

将在 HTTP 服务器上使用的端口地址。缺省值为 8080，除非 HTTPS 设置为 true，则缺省值为 443。如果您更改此属性（且 HTTP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

IdleTimeout

时间（以分钟计），过了这段时间后将关闭空闲连接。请注意，队列管理器到队列管理器的通道还具有 DISCINT 属性。如果您设置 IdleTimeout 参数，请注意 DISCINT。值 0 表明无空闲超时。只有当重新启动此路由时，对此属性的更改才生效。

IgnoreExpiredCRLs

将此属性设置为 true 以忽略到期的 CRL。缺省值为 false。

注意

如果您启用此属性，可使用已撤销的证书来生成 SSL 连接。

LDAP

将此属性设置为 true 以在使用 SSL 连接时启用使用 LDAP 服务器。MQIPT 将使用 LDAP 服务器来检索 CRL 和 ARL。SSLClient 或 SSLServer 属性也必须启用以使该属性生效。

LDAPIgnoreErrors

将此属性设置为 true 以在执行 LDAP 搜索时忽略任何连接或超时错误。如果 MQIPT 无法成功执行搜索，除非启用了此属性，否则它将不会允许客户机连接完成。成功搜索的含义是检索到 CRL，或对于指定的 CA，没有可用的 CRL。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

注意

如果您启用此属性，可使用已撤销的证书来生成 SSL 连接。

LDAPCacheTimeout

当从 LDAP 服务器检索到 CRL 时，它将内部存储在临时高速缓存的 MQIPT 中。此高速缓存中的条目将在特定的超时（由此属性指定）后到期。缺省值为 24 小时。将超时值指定为 0 意味着高速缓存中的条目将在路由重新启动后才到期。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

LDAPSaveCRL

将此属性设置为 true 以使用从 LDAP 服务器检索到的任何 CRL 更新给定的密钥环文件。密钥环文件是由 SSLClientKeyRing、SSLClientCAKeyRing、SSLServerKeyRing 和 SSLServerCAKeyRing 属性指定的。这暗示着 MQIPT 必须对密钥环文件有写访问权。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

LDAPServer1

将此属性设置为主 LDAP 服务器的主机名或 IP 地址。如果启用了 LDAP，则必须设置此属性。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

LDAPServer1Port

将此属性设置为主 LDAP 服务器的侦听端口地址。它的缺省值为 389。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

LDAPServer1Userid

将此属性设置为访问主 LDAP 服务器所需的用户标识。如果需要授权访问主 LDAP 服务器，则必须设置此属性。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

LDAPServer1Password

将此属性设置为访问主 LDAP 服务器所需的密码。如果设置了 LDAPServer1Userid，则必须设置此属性。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

LDAPServer1Timeout

将此属性设置为 MQIPT 将等待来自主 LDAP 服务器的响应的时间秒数。它的缺省值为 0，它表明连接不会超时。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

LDAPServer2

将此属性设置为备份 LDAP 服务器的主机名或 IP 地址。此属性是可选的。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

LDAPServer2Port

将此属性设置为备份 LDAP 服务器的侦听端口地址。它的缺省值为 389。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

LDAPServer2Userid

将此属性设置为访问备份 LDAP 服务器所需的用户标识。如果需要授权访问备份 LDAP 服务器，则必须设置此属性。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

LDAPServer2Password

将此属性设置为访问备份 LDAP 服务器所需的密码。如果启用了 LDAPServer2，则必须设置此属性。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

LDAPServer2Timeout

将此属性设置为 MQIPT 将等待来自备份 LDAP 服务器的响应的时间秒数。它的缺省值为 0，它表明连接不会超时。如果您更改此属性（且 LDAP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

ListenerPort

路由应该侦听进入请求所在的端口号。每个 route 节**必须**包含一个显式 ListenerPort 值；此外，在每个节中设置的 ListenerPort 值必须不同。可以使用任何有效的端口号（包括端口 80 和 443），只要所选的端口并没有由任何其它在同一主机上运行的 TCP/IP 侦听器使用。

LocalAddress

所有连接绑定至的本地 IP 地址。如果您更改此属性，当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

LogDir

使用此属性来定义日志和跟踪文件的目录名。不会影响对此属性的更改，除非已停止并重新启动 MQIPTServlet。缺省值为 <null>。此属性仅对 MQIPTServlet 有效

MaxConnectionThreads

最大连接线程数，因此也是可由此路由处理的最大并发连接数。如果达到此限制，则 `MaxConnectionThreads` 值还表明一旦所有线程在使用时，将排队的连接数。超出此数，则拒绝后继连接请求。允许的最小值大于 1 或等于 MinConnectionThreads 值。如果对此属性的某一更改影响路由，当发出 REFRESH 命令时将使用新值。所有连接立即挑选新值。不终止此路由。

MinConnectionThreads

最小连接线程（处理此路由上的进入连接的线程）数。这是当启动路由时分配的线

程数，分配的线程总数在路由活动期间不小于该值。允许的最小值为 0，并且该值必须小于对 MaxConnectionThreads 指定的值。只有当重新启动此路由时，对此属性的更改才生效。

Name

帮助标识路由的可选名称。它出现在控制台消息和跟踪信息中。只有当重新启动此路由时，对此属性的更改才生效。

NDAdvisor

将此属性设置为 true，以便 Network Dispatcher 管理路由以允许此路由响应来自定制顾问程序的请求。如果您将此属性更改为 false，当发出 REFRESH 命令时，此路由将停止。此路由的所有连接将终止。要使用 NDAdvisorReplaceMode 属性，请将此属性设置为 true。

NDAdvisorReplaceMode

将此属性设置为 true，以使用 Network Dispatcher 定制顾问程序的“替换”方式。您必须已为此路由的 ListenerPort 地址启动 mqipt_replace 定制顾问程序。将此属性设置为 false，以使用“正常”方式。您必须将 NDAdvisor 属性设置为 true，以使用此属性。

OutgoingPort

这是外出连接使用的启动端口地址。端口地址的范围匹配该路由的 MaxConnectionThread 值。缺省值 0 将使用系统定义的端口地址。如果您更改此属性，当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

QMgrAccess

只有当 QMgrAccess 的值设置为值 true 时，此路由才允许进入队列管理器通道连接（例如，发送方通道）。如果您将此属性更改为 false，当发出 REFRESH 命令时，此路由将停止。此路由的所有连接将终止。

QoS

将此属性设置为 true，以启用此路由上所有连接的“服务质量”。只能在 Linux 上启用此属性。如果您更改此属性，当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。此属性不能与下列一起使用:

- HTTP
- SSLClient
- SSLProxyMode
- SSLServer

QosToCaller

此属性将所有流量的优先级从 MQIPT 机器设置到连接的发起方。例如，将此属性设置为低优先级 1、中优先级 2 和高优先级 3（缺省为 1）。如果您更改此属性（且 QoS 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。终止与此路由的所有连接

QosToDest

此属性将所有流量的优先级从 MQIPT 机器设置到连接目标（由 Destination 属性定义）。例如，将此属性设置为低优先级 1、中优先级 2 和高优先级 3（缺省为 1）。如果您更改此属性（且 QoS 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。终止与此路由的所有连接

RouteRestart

将此属性设置为 false，防止当更改了其它路由属性并发出 REFRESH 命令时路由重新启动。此属性的缺省值为 true。

SecurityExit

将此属性设置为 true 以启用用户定义的安全性出口。此属性的缺省值为 false。

SecurityExitName

用户定义的安全性出口的类名。如果 SecurityExit 设置为 true，则必须设置此属性。如果您更改此属性（且 SecurityExit 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SecurityExitPath

包含用户定义的安全性出口的全限定路径名。如果没有设置此属性，则缺省值为 exits 子目录。此属性也可以定义包含用户定义的安全性出口的 jar 文件的名称。如果您更改此属性（且 SecurityExit 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SecurityExitTimeout

MQIPT 使用超时值来确定当验证连接请求时等待响应的时间长度（以秒为单位）。缺省值为 5 秒。如果您更改此属性（且 SecurityExit 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

ServletClient

当连接到 MQIPT servlet 时，将此属性设置为 true。HTTP 属性也必须设置为 true。如果您更改此属性（HTTP 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。

SocksClient

将此属性设置为 true，使此路由作为 Socks 客户机，并使用 SocksProxyHost 和 SocksProxyPort 属性通过 Socks 代理定义所有连接。如果您更改此属性，当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。此属性不能与下列一起使用:

- HTTP
- SocksServer
- SSLClient
- SSLProxyMode

SocksProxyHost

此路由的所有连接所使用的 Socks 代理的主机名（或点分十进制 IP 地址）。如果您更改此属性（且 SocksClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。终止与此路由的所有连接

SocksProxyPort

将在 Socks 代理上使用的端口地址。缺省值为 1080。如果您更改此属性（且 SocksClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。终止与此路由的所有连接

SocksServer

将此属性设置为 true，使此路由作为 Socks 代理，并接受 Socks 客户机连接。如果您更改此属性，当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。此属性不能与下列一起使用:

- SocksClient

- SSLProxyMode
- SSLServer

SSLClient

将此属性设置为 true，使此路由作为 SSL 客户机并进行外出 SSL 连接。设置为 true 隐含表示目的地是充当 SSL 服务器的另一个 MQIPT 或 HTTP 代理／服务器。您必须使用 SSLClientKeyRing 或 SSLClientCAKeyRing 属性指定密钥环文件的名称。如果您更改此属性，当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。此属性不能与下列一起使用:

- HTTP
- QoS
- SSLProxyMode

SSLClientCAKeyRing

包含 CA 证书的密钥环文件的全限定文件名，此文件用于认证来自 SSL 服务器的证书。在 Windows 平台上，您必须使用双反斜杠 (\\) 作为文件分隔符。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientCAKeyRingPW

包含打开客户机 CA 密钥环的密码的全限定文件名。在 Windows 平台上，您必须使用双反斜杠 (\\) 作为文件分隔符。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientCipherSuites

在 SSL 客户机端使用的 SSL 密码套件名。它可以为一个或多个支持的密码套件。如果您将其保留为空白，则 SSL 客户机使用来自 SSLClientKeyRing 的支持密码套件。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientConnectTimeout

将此属性设置为 SSL 客户机等待接受 SSL 连接的秒数。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientDN_C

使用此属性接受从此国家或地区名称的 SSL 服务器接收到的证书。此名称可以用星号（*）作为前缀或后缀来扩展其作用域。如果您不指定此属性，则表示“所有国家或地区名称”。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientDN_CN

使用此属性接受从此公共名称的 SSL 服务器接收到的证书。此名称可以用星号（*）作为前缀或后缀来扩展其作用域。如果您不指定此属性，则表示“所有国家或地区名称”。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientDN_L

使用此属性接受从此位置的 SSL 服务器接收到的证书。此名称可以用星号（*）作为前缀或后缀来扩展其作用域。如果您不指定此属性，则表示“所有位置”。如果

| 您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientDN_O

| 使用此属性接受从此组织的 SSL 服务器接收到的证书。此名称可以用星号（*）作为前缀或后缀来扩展其作用域。如果您不指定此属性，则表示“所有组织”。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientDN_OU

| 使用此属性接受从此组织部门的 SSL 服务器接收到的证书。此名称可以用星号（*）作为前缀或后缀来扩展其作用域。如果您不指定此属性，则表示“所有组织部门”。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientDN_ST

| 使用此属性接受从该省／直辖市的 SSL 服务器接收到的证书。此名称可以用星号（*）作为前缀或后缀来扩展其作用域。如果您不指定此属性，则表示“所有省／直辖市”。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientKeyRing

| 包含客户机证书的密钥环文件的全限定文件名。在 **Windows 平台**上，您必须使用双反斜杠 (\\) 作为文件分隔符。如果您将 SSLClient 设置为 true，您必须指定 SSLClientKeyRing。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientKeyRingPW

| 包含打开客户机密钥环的密码的全限定文件名。在 **Windows 平台**上，您必须使用双反斜杠 (\\) 作为文件分隔符。如果您将 SSLClient 设置为 true，您必须指定 SSLClientKeyRingPW。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientSiteDN_C

| 使用此属性指定国家或地区名称以选择发送到 SSL 服务器的证书。如果您不指定此属性，则表示“任何国家或地区名称”。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientSiteDN_CN

| 使用此属性指定公共名称以选择发送到 SSL 服务器的证书。如果您不指定此属性，则表示“任何公共名称”。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientSiteDN_L

| 使用此属性指定位置名称以选择发送到 SSL 服务器的证书。如果您不指定此属性，则表示“任何位置名称”。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientSiteDN_O

| 使用此属性指定组织名称以选择发送到 SSL 服务器的证书。如果您不指定此属性，则表示“任何组织名称”。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientSiteDN_OU

使用此属性指定组织部门名称以选择发送到 SSL 服务器的证书。如果您不指定此属性，则表示“任何组织部门名称”。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientSiteDN_ST

使用此属性指定省／直辖市名称以选择发送到 SSL 服务器的证书。如果您不指定此属性，则表示“任何省／直辖市名称”。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLClientSiteLabel

使用此属性指定标号名称以选择发送到 SSL 服务器的证书。如果您不指定此属性，则表示“任何标号名称”。如果您更改此属性（且 SSLClient 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLProxyMode

将此属性设置为 true，使此路由只接受 SSL 客户机连接请求，并使此请求直接通往目标。如果您更改此属性，当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。此属性不能与下列一起使用:

- HTTP
- QoS
- SocksClient
- SSLClient
- SSLServer

SSLServer

将此属性设置为 true，使路由作为 SSL 服务器，并接受进入 SSL 连接。设置 true 表示调用程序是作为 SSL 客户机的另一个 MQIPT。如果您更改此属性，当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。此属性不能与下列一起使用:

- QoS
- SocksServer
- SSLProxyMode

SSLServerCAKeyRing

包含 CA 证书的密钥环文件的全限定文件名，此文件用于认证来自 SSL 客户机的证书。在 Windows 平台上，您必须使用双反斜杠 (\\) 作为文件分隔符。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerCAKeyRingPW

包含用于打开服务器 CA 密钥环的密码的全限定文件名。在 Windows 平台上，您必须使用双反斜杠 (\\) 作为文件分隔符。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerAskClientAuth

使用此属性请求 SSL 服务器的 SSL 客户机认证。此 SSL 客户机必须具有自己的

证书以发送到 SSL 服务器。从密钥环文件检索此证书。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerCipherSuites

在 SSL 服务器端使用的 SSL 密码套件名。它可以为一个或多个支持的密码套件。如果您将其保留为空白，则 SSL 服务器使用来自 SSLServerKeyRing 的支持密码套件。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerDN_C

使用此属性接受从此国家或地区名称的 SSL 客户机接收到的证书。此名称可以用星号（*）作为前缀或后缀来扩展其作用域。如果您不指定此属性，则表示“所有国家或地区名称”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerDN_CN

使用此属性接受从此公共名的 SSL 客户机接收到的证书。此名称可以用星号（*）作为前缀或后缀来扩展其作用域。如果您不指定此属性，则表示“所有公共名”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerDN_L

使用此属性接受从此位置的 SSL 客户机接收到的证书。此名称可以用星号（*）作为前缀或后缀来扩展其作用域。如果您不指定此属性，则表示“所有位置”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerDN_O

使用此属性接受从此组织的 SSL 客户机接收到的证书。此名称可以用星号（*）作为前缀或后缀来扩展其作用域。如果您不指定此属性，则表示“所有组织”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerDN_OU

使用此属性接受从此组织部门的 SSL 客户机接收到的证书。此名称可以用星号（*）作为前缀或后缀来扩展其作用域。如果您不指定此属性，则表示“所有组织部门”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerDN_ST

使用此属性接受从该省／直辖市的 SSL 客户机接收到的证书。此名称可以用星号（*）作为前缀或后缀来扩展其作用域。如果您不指定此属性，则表示“所有省／直辖市”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerKeyRing

包含服务器证书的密钥环文件的全限定文件名。在 **Windows 平台**上，您必须使用双反斜杠 (\\) 作为文件分隔符。如果您将 SSLServer 设置为 true，则您必须指定 SSLServerKeyRing。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerKeyRingPW

包含打开服务器密钥环的密码的全限定文件名。在 **Windows 平台**上，您必须使用

双反斜杠 (\\) 作为文件分隔符。如果您将 SSLServer 设置为 true，则您必须指定 SSLServerKeyRingPW。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerSiteDN_C

使用此属性指定国家或地区名称以选择发送到 SSL 客户机的证书。如果您不指定此属性，则表示“任何国家或地区名称”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerSiteDN_CN

使用此属性指定公共名称以选择发送到 SSL 客户机的证书。如果您不指定此属性，则表示“任何公共名称”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerSiteDN_L

使用此属性指定位置名称以选择发送到 SSL 客户机的证书。如果您不指定此属性，则表示“任何位置名称”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerSiteDN_O

使用此属性指定组织名称以选择发送到 SSL 客户机的证书。如果您不指定此属性，则表示“任何组织名称”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerSiteDN_OU

使用此属性指定组织部门名称以选择发送到 SSL 客户机的证书。如果您不指定此属性，则表示“任何组织部门名称”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerSiteDN_ST

使用此属性指定省／直辖市名称以选择发送到 SSL 客户机的证书。如果您不指定此属性，则表示“任何省／直辖市名称”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

SSLServerSiteLabel

使用此属性指定标号名称以选择发送到 SSL 客户机的证书。如果您不指定此属性，则表示“任何标号名称”。如果您更改此属性（且 SSLServer 设置为 true），当发出 REFRESH 命令时，路由将停止并重新启动。此路由的所有连接将终止。

Trace

必需的跟踪级别可以由 0-5 范围内的整数指定。值 0 意味着无跟踪；5 请求完全跟踪。

如果对此属性的某一更改影响路由，当发出 REFRESH 命令时将使用新值。所有连接立即挑选新值。不终止此路由。

UriName

虽然这些缺省值将满足大部分配置，但是在使用 HTTP 代理或 MQIPT servlet 时，此属性可用于更改资源的统一资源标识名称。HTTP 代理的缺省为:

```
HTTP://<destination>:<destination_port>/mqipt
```

MQIPT servlet 的缺省为:

HTTP://<destination>:<destination_port>/MQIPTServlet

如果您更改此属性（HTTP 或 ServletClient 设置为 True），当发出 REFRESH 命令时，路由将停止并重新启动。

第 20 章 internet pass-thru 入门

本章帮助您初步了解 MQIPT: 它将为您介绍了一些简单配置的设置，以确认本产品安装成功。

本章包含下列各节:

假设

对于每个示例，我们作了下列假设:

- 您正在使用 Windows NT（尽管这些示例将在任何支持的平台上运行）
- 您熟悉在 WebSphere MQ 上定义队列管理器、队列和通道
- 您已安装了 WebSphere MQ 客户机和服务器
- MQIPT 安装在名为 `C:\mqipt` 的目录中（在 Windows 上）
- 客户机、服务器以及每个 MQIPT 分别安装在不同的机器上
- 您熟悉使用 `amqsputc` 命令将消息放入队列
- 您熟悉使用 `amqsgetc` 命令从队列中取出消息

在 WebSphere MQ 服务器上，您已执行下列操作:

- 定义了名为 MQIPT.QM1 的队列管理器
- 定义了名为 MQIPT.CONN.CHANNEL 的服务器连接通道
- 定义了名为 MQIPT.LOCAL.QUEUE 的本地队列
- 在端口 1414 上为 MQIPT.QM1 启动了 TCP/IP 侦听器

只有一个应用程序可以在相同机器上的给定端口地址上侦听。如果端口 1414 已被使用，则选择一个空闲的端口地址并在下面的示例中替换它。

一旦您已完成这些操作，您就可以通过使用 amqsputc 命令将消息放入队列管理器的本地队列，并使用 amqsgetc 命令检索该消息，从而测试从 WebSphere MQ 客户机到队列管理器的路由。

示例配置

下列示例以图及步进说明的形式表示，您可以使用每个图右侧的勾选框来跟踪示例的进展。在某些示例中，要求您编辑 mqipt.conf 文件，该文件可在 MQIPT 主目录中找到。

在开始之前，请确保您已执行下列操作:

- 将 mqiptSample.conf 复制为 mqipt.conf
- 编辑 mqipt.conf 并删除所有路由
- 将 ClientAccess 条目更改为 True
- 将 Destination 从 mqserver.company2.com 更改为您的队列管理器的目的地
- 将 DestinationPort 地址更改为您的队列管理器所使用的地址
- 请阅读第 91 页的『假设』

安装验证测试

这是一个简单配置，它用于确保 MQIPT 已正确安装。

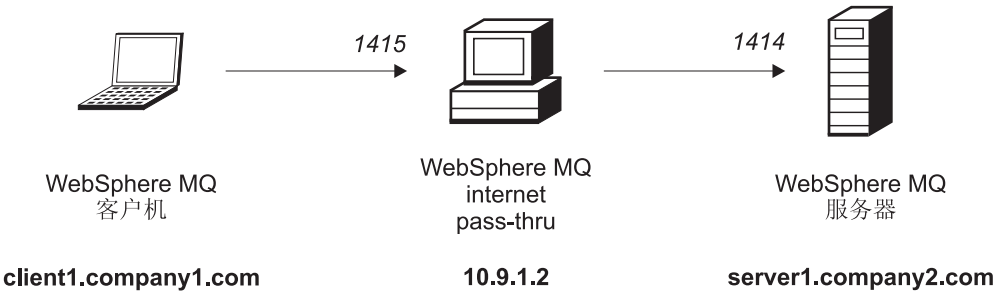

图 10. IVT 网络图

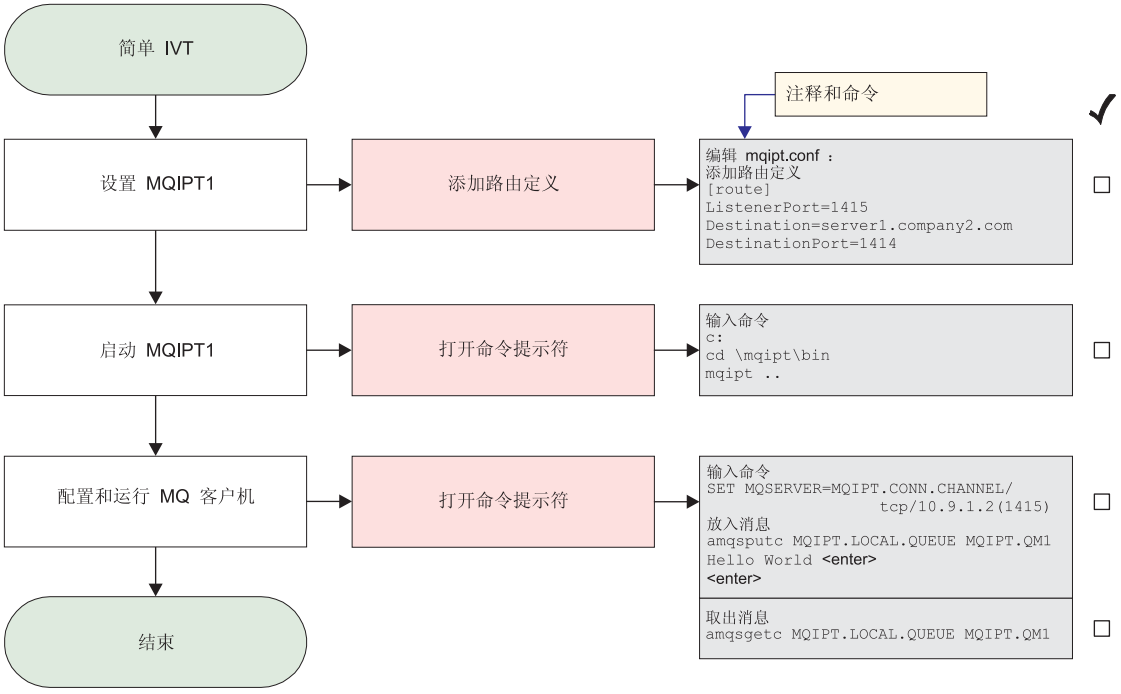

图 11. IVT 配置

1. 设置 MQIPT1

   编辑 mqipt.conf 并添加路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   ```

2. 启动 MQIPT1

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \mqipt\bin
   mqipt ..
   ```

   下列消息表明成功完成:

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
   MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
   MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
   MQCPI006 路由 1415 已启动，并将转发消息至:
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....使用 MQ 协议
   MQCPI078 路由 1415 用于连接请求准备就绪
   ```

3. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

4. 使用以下命令放入消息:

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   Hello world <enter>
    <enter>
   ```

5. 使用以下命令取出消息:

   ```
   amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   ```

您将看见 “Hello world”。

SSL 服务器认证

在此示例中，您将通过使 WebSphere MQ 客户机通过两个 MQIPT 连接到 WebSphere MQ 服务器，从而使用样本测试证书（`sslsample.pfx` 密钥环文件）测试 SSL 连接。在 SSL 握手期间，服务器将其测试证书发送到客户机。而此客户机则将使用其证书副本（带有信任为同等标志）来认证服务器。将使用缺省密码套件 SSL_RSA_WITH_RC4_128_MD5。（基于从第 92 页的『安装验证测试』创建的 `mqipt.conf`。）要获取关于如何创建测试证书以在此示例中使用的详细信息，请参阅第 111 页的『创建 SSL 测试证书』。

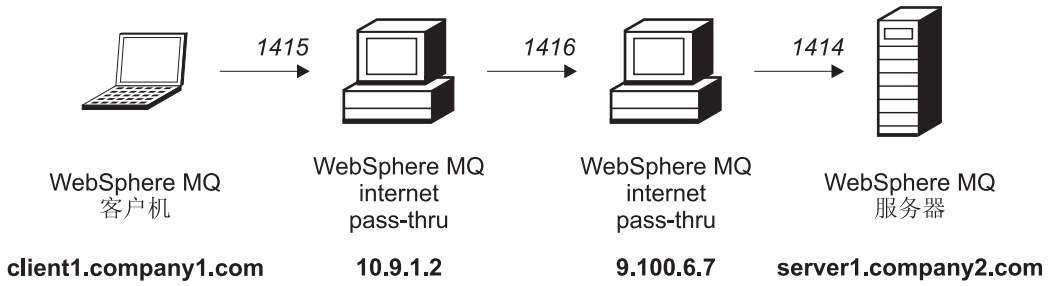

图 12. SSL 服务器网络图

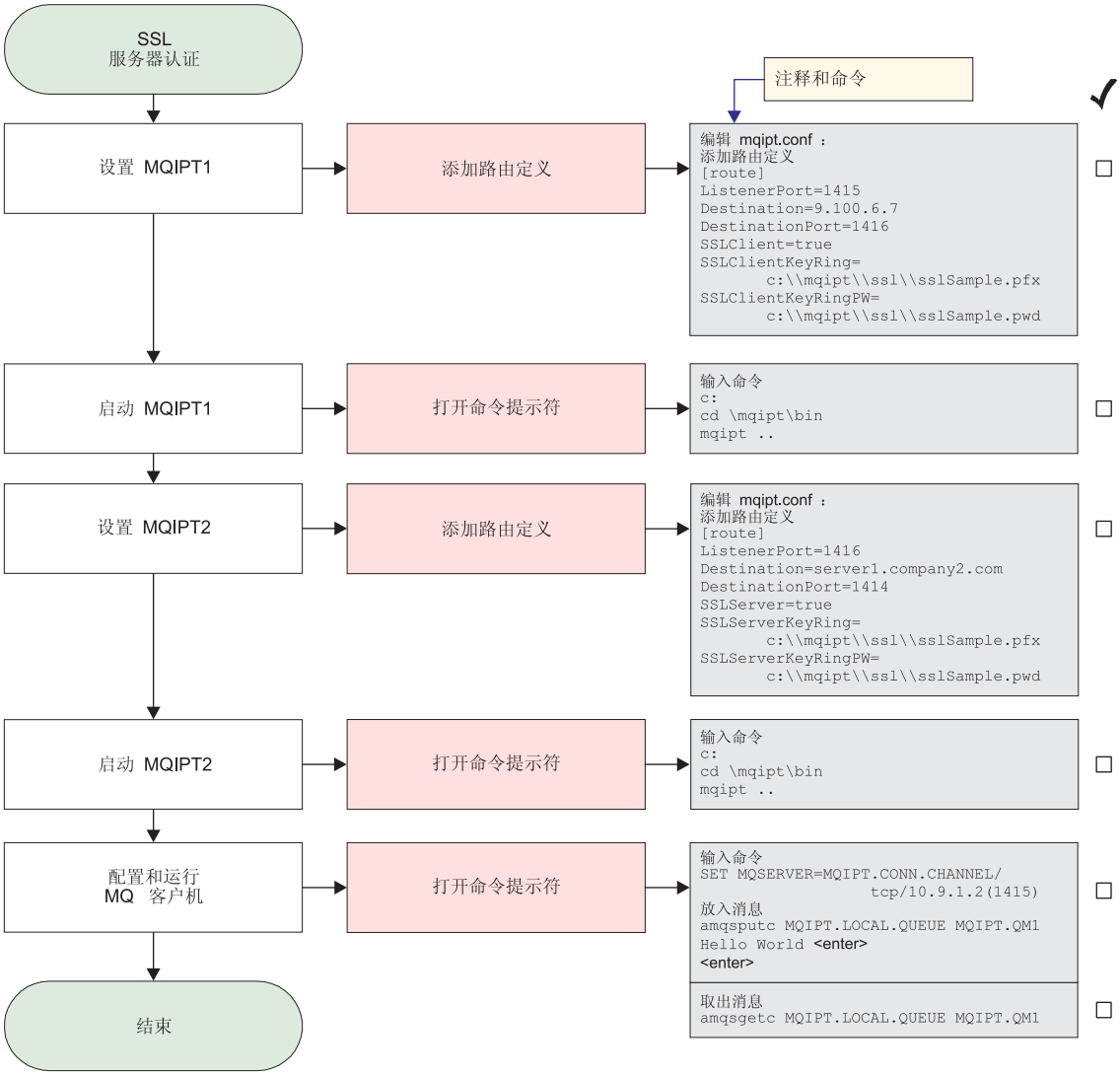

图 13. SSL 服务器认证

1. 设置 MQIPT1

   编辑 `mqipt.conf` 并添加路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=9.100.6.7
   DestinationPort=1416
   SSLClient=true
   SSLClientKeyRing=C:\\mqipt\\sslSample.pfx
   SSLClientKeyRingPW=C:\\mqipt\\sslSample.pwd
   ```

2. 启动 MQIPT1

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \mqipt\bin
   mqipt ..
   ```

   下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI011 路径 c:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....9.100.6.7(1416)
MQCPI035 ....使用 MQ 协议
MQCPI036 ....使用下列属性启用 SSL 客户机端:
MQCPI031 ......密码套件 <null>
MQCPI032 ......密钥环文件 c:\mqipt\sslSample.pfx
MQCPI047 ......CA 密钥环文件 <null>
MQCPI038 ......专有名称 CN=* O=* OU=* L=* ST=* C=*
MQCPI078 路由 1415 用于连接请求准备就绪
```

3. 设置 MQIPT2

   编辑 mqipt.conf 并添加路由定义:

   ```
   [route]
   ListenerPort=1416
   Destination=Server1.company2.com
   DestinationPort=1414
   SSLClient=true
   SSLServerKeyRing=C:\\mqipt\\sslSample.pfx
   SSLServerKeyRingPW=C:\\mqipt\\sslSample.pwd
   ```

4. 启动 MQIPT2

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \mqipt\bin
   mqipt
   ```

   下列消息表明成功完成:

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
   MQCPI011 路径 c:\mqipt\logs 将用于存储日志文件
   MQCPI006 路由 1416 已启动，并将转发消息至:
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....使用 MQ 协议
   MQCPI037 ....使用下列属性启用 SSL 服务器端:
   MQCPI031 ......密码套件 <null>
   MQCPI032 ......密钥环文件 c:\mqipt\sslSample.pfx
   MQCPI047 ......CA 密钥环文件 <null>
   MQCPI038 ......专有名称 CN=* O=* OU=* L=* ST=* C=*
   MQCPI033 ......客户机认证设置为 false
   MQCPI078 路由 1416 用于连接请求准备就绪
   ```

5. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

6. 使用以下命令放入消息:

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   Hello world <enter>
    <enter>
   ```

7. 使用以下命令取出消息:

   ```
   amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   ```

您将看见 “Hello world”。

SSL 客户机认证

在此示例中，您将使用样本测试证书测试 SSL 连接。这将执行服务器和客户机认证。在 SSL 握手期间，服务器将其测试证书发送到客户机。客户机将使用其证书副本（带有信任为对等标志）来认证服务器。然后客户机将其测试证书发送到服务器。服务器将使用其证书副本（带有信任为对等标志）来认证客户机。将使用缺省密码套件 SSL_RSA_WITH_RC4_128_MD5。（基于从第 92 页的『安装验证测试』创建的 `mqipt.conf`。）

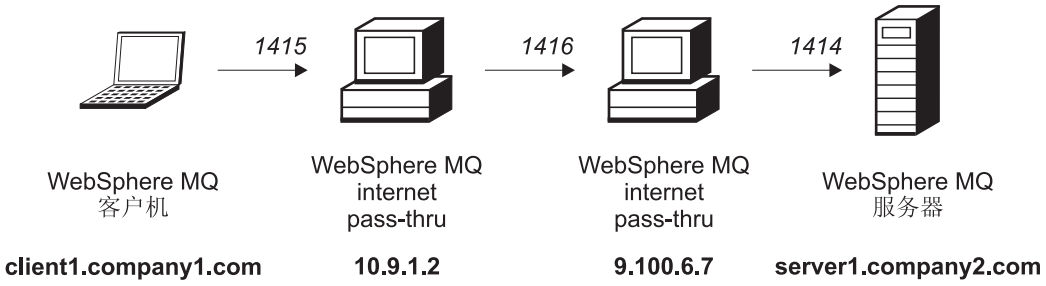

图 14. SSL 客户机网络图

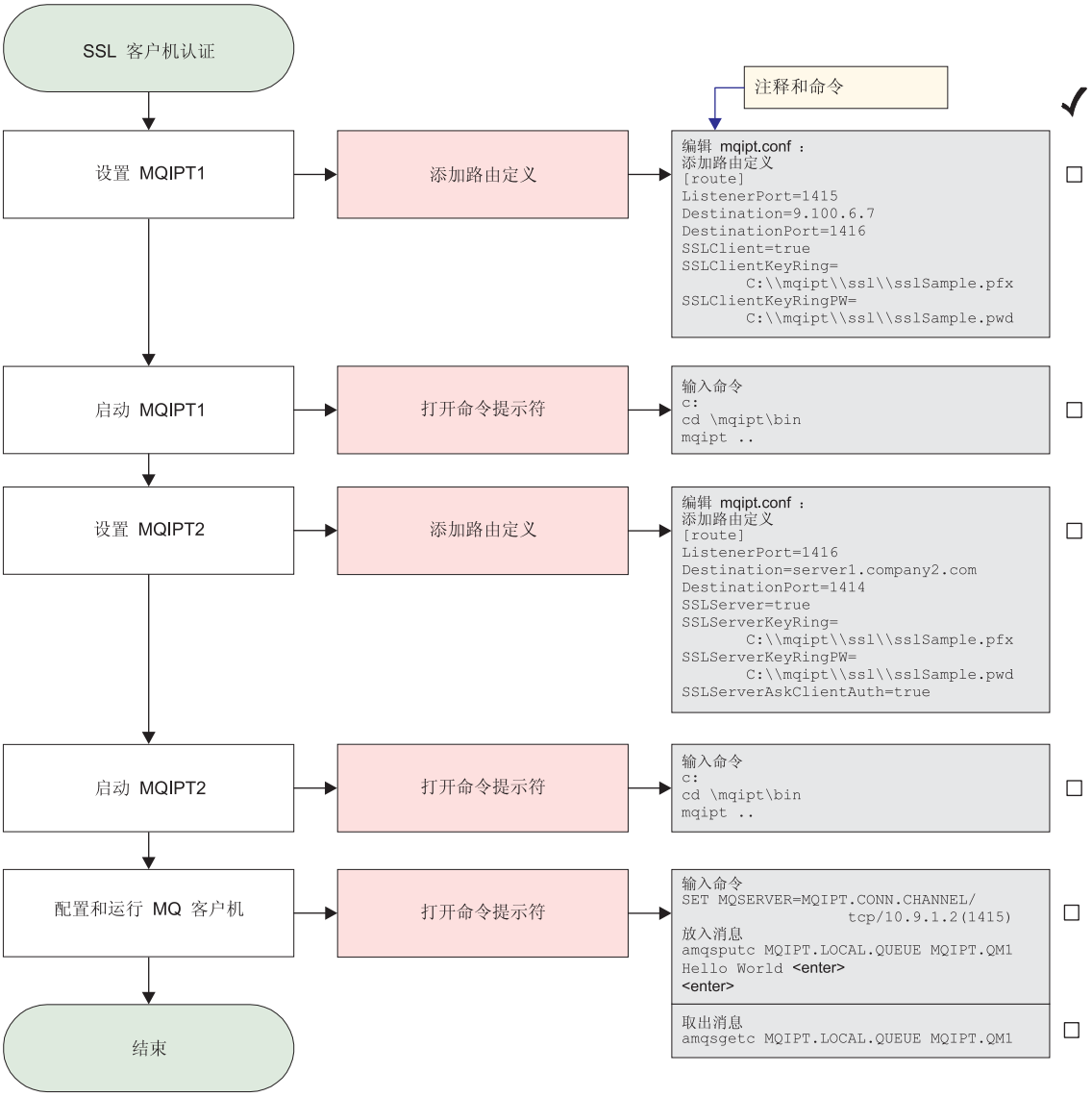

图 15. SSL 客户机认证

1. 设置 MQIPT1

   编辑 mqipt.conf 并添加路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=9.100.6.7
   DestinationPort=1416
   SSLClient=true
   SSLClientKeyRing=C:\\mqipt\\sslSample.pfx
   SSLClientKeyRingPW=C:\\mqipt\\sslSample.pwd
   ```

2. 启动 MQIPT1

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \mqipt\bin
   mqipt ..
   ```

   下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI011 路径 c:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....9.100.6.7(1416)
MQCPI035 ....使用 MQ 协议
MQCPI036 ....使用下列属性启用 SSL 客户机端:
MQCPI031 ......密码套件 <null>
MQCPI032 ......密钥环文件 c:\mqipt\sslSample.pfx
MQCPI047 ......CA 密钥环文件 <null>
MQCPI038 ......专有名称 CN=* O=* OU=* L=* ST=* C=*
MQCPI078 路由 1415 用于连接请求准备就绪
```

3. 设置 MQIPT2

   编辑 mqipt.conf 并添加路由定义:

```
[route]
ListenerPort=1416
Destination=Server1.company2.com
DestinationPort=1414
SSLClient=true
SSLServerKeyRing=C:\\mqipt\\sslSample.pfx
SSLServerKeyRingPW=C:\\mqipt\\sslSample.pwd
```

4. 启动 MQIPT2

   打开命令提示符并输入下列内容:

```
c:
cd \mqipt\bin
mqipt
```

   下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI011 路径 c:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1416 已启动，并将转发消息至:
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....使用 MQ 协议
MQCPI037 ....使用下列属性启用 SSL 服务器端:
MQCPI031 ......密码套件 <null>
MQCPI032 ......密钥环文件 c:\mqipt\sslSample.pfx
MQCPI047 ......CA 密钥环文件 <null>
MQCPI038 ......专有名称 CN=* O=* OU=* L=* ST=* C=*
MQCPI033 ......客户机认证设置为 true
MQCPI078 路由 1416 用于连接请求准备就绪
```

5. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

6. 使用以下命令放入消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
Hello world <enter>
 <enter>
```

7. 使用以下命令取出消息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
```

您将看见 “Hello world”。

HTTP 代理配置

在此示例中，您将使用 HTTP 代理（IBM Caching Proxy）来测试连接。CP 必须为级别 3.6 或更高级别，您还必须检查下列内容:

- `ProxyPersistence` 必须为 `on`，这将允许持久连接
- `MaxPersistRequest` 是否为 5000，这是断开连接之前单个连接上所允许的请求数
- `PersistTimeout` 是否为 12hrs，这是允许连接存在的时间

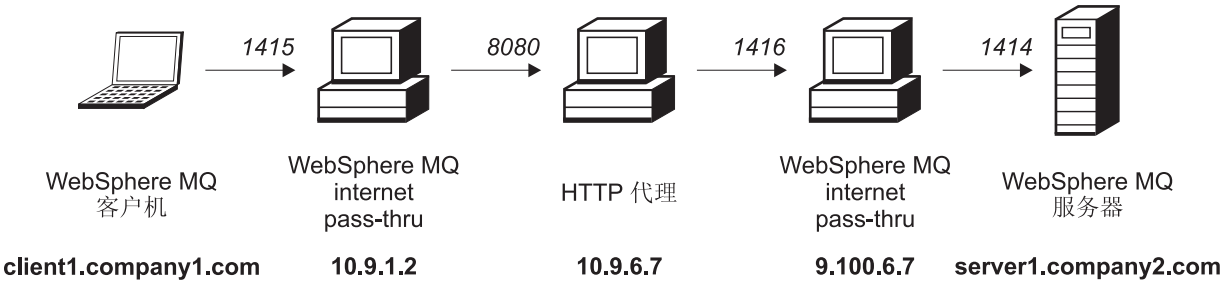

图 16. HTTP 代理网络图

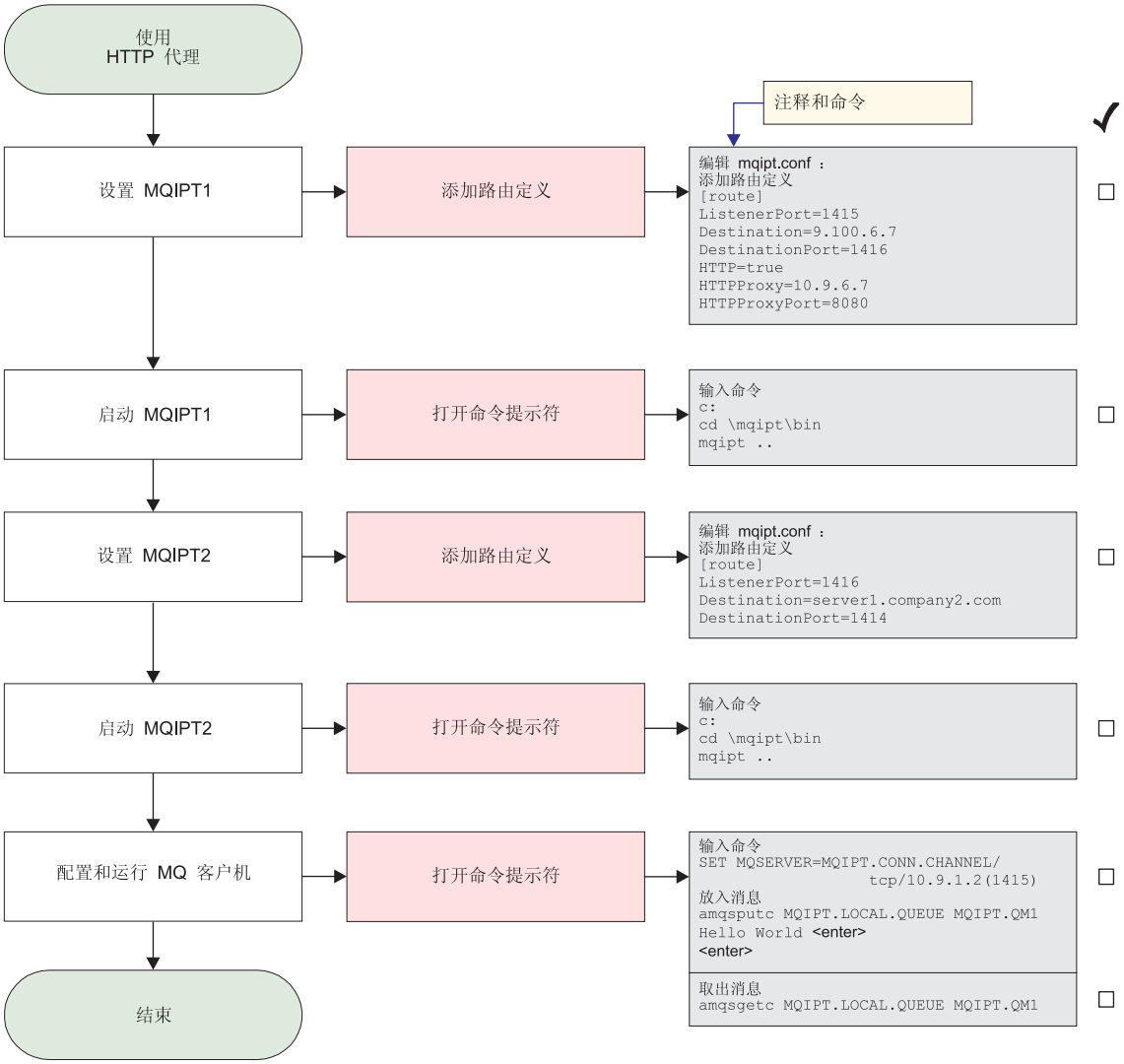

图 17. HTTP 代理配置

1. 设置 MQIPT1

   编辑 `mqipt.conf` 并添加路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=9.100.6.7
   DestinationPort=1416
   HTTP=true
   HTTPProxy=true
   HTTPProxyPort=8080
   ```

2. 启动 MQIPT1

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \mqipt\bin
   mqipt ..
   ```

   下列消息表明成功完成:

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
   MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
   ```

```
MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....9.100.6.7(1416)
MQCPI035 ....使用 HTTP
MQCPI024 ....和位于 10.9.6.7(1080) 的 HTTP 代理
MQCPI078 路由 1415 用于连接请求准备就绪
```

3. 设置 MQIPT2

   编辑 mqipt.conf 并添加路由定义:

   ```
   [route]
   ListenerPort=1416
   Destination=Server1.company2.com
   DestinationPort=1414
   ```

4. 启动 MQIPT2

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \mqipt\bin
   mqipt
   ```

   下列消息表明成功完成:

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
   MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
   MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
   MQCPI006 路由 1416 已启动，并将转发消息至:
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....使用 MQ 协议
   MQCPI078 路由 1416 用于连接请求准备就绪
   ```

5. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

6. 使用以下命令放入消息:

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   Hello world <enter>
    <enter>
   ```

7. 使用以下命令取出消息:

   ```
   amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   ```

您将看见 “Hello world”。

配置访问控制

在此示例中，您将通过使用 Java 安全性管理器在 MQIPT 侦听器端口上添加安全性检查来设置您的 MQIPT，从而只接受来自特定客户机的连接。

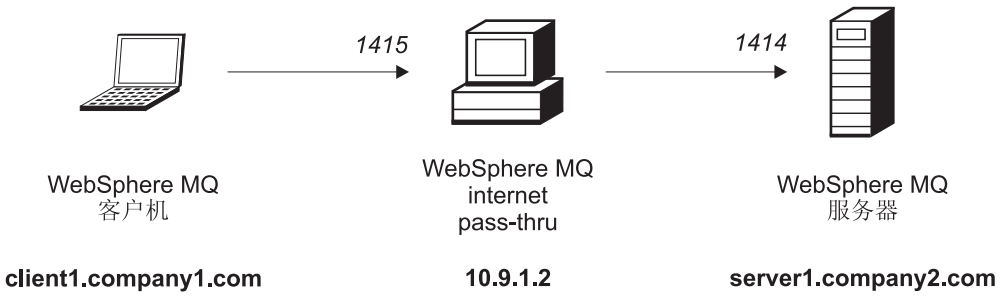

图 18. 访问控制网络图

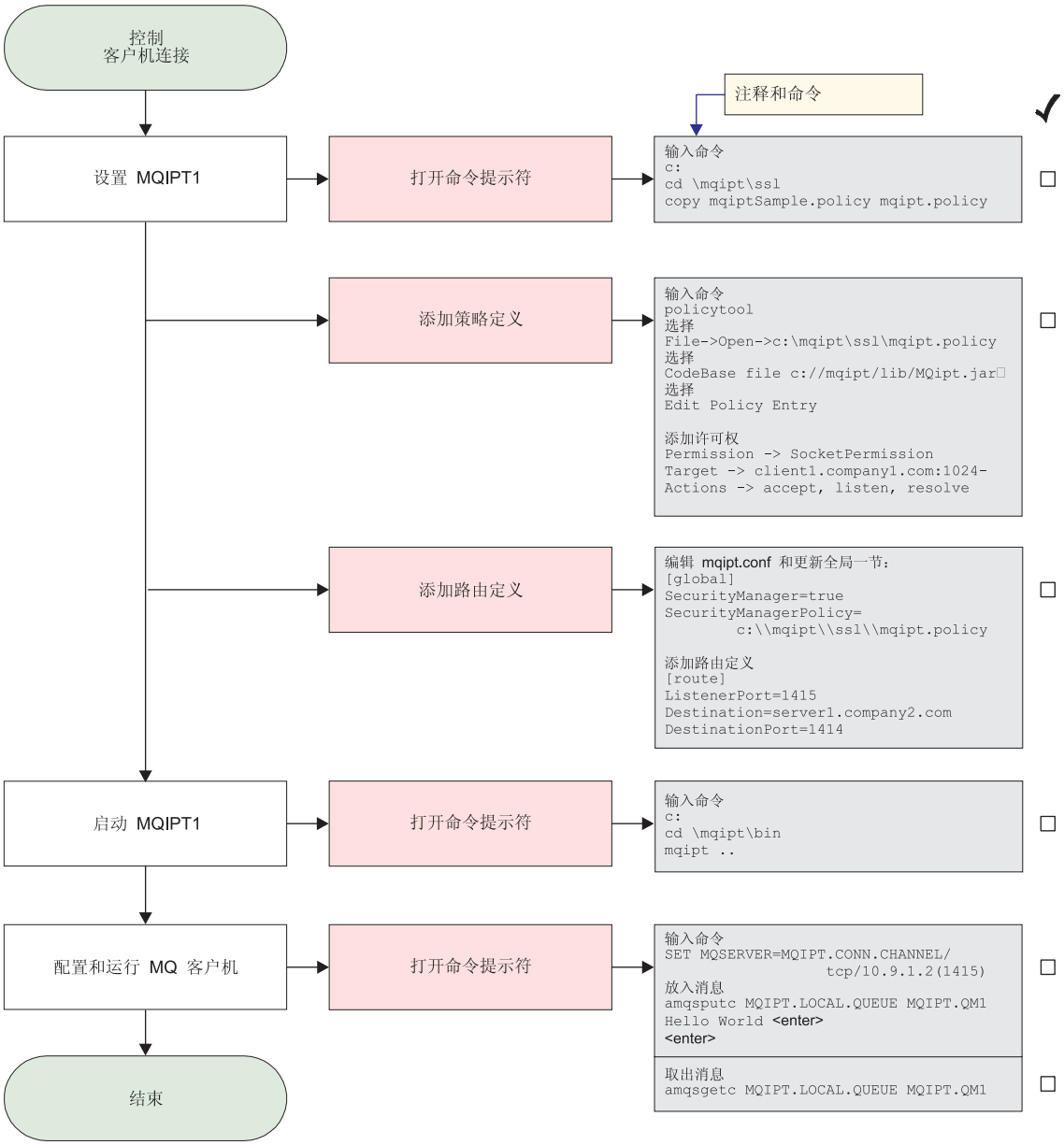

图 19. 访问控制配置

1. 设置 MQIPT1

a. 打开命令提示符并输入下列内容:

```
c:
cd \mqipt\ssl
copy c:\mqipt\ssl\mqiptSample.policy to mqipt.policy
```

b. 使用以下命令添加策略定义:

```
policytool
```

1) 选择 “文件” -> “打开” -> “c:\mqipt\ssl\mqipt.policy”

2) 选择:

```
file://C:/Program Files/IBM/WebSphere MQ internet pass-thru/lib/MQipt.jar
```

3) 将 CodeBase 从:

```
file://C:/Program Files/IBM/WebSphere MQ internet pass-thru/lib/MQipt.jar
```

更改为:

```
file://C:/mqipt/lib/MQipt.jar
```

4) 将所有许可权从:

```
C:\\Program Files\\IBM\\WebSphere MQ internet pass-thru
```

更改为:

```
C:\\mqipt
```

5) 添加 SocketPermission:

```
Permission=SocketPermission
Target=client1.company1.com:1024-
Actions=accept, listen, resolve
```

c. 编辑 mqipt.conf 并添加:

1) global 节的两个属性:

```
[global]
SecurityManager=true
SecurityManagerPolicy=c:\\mqipt\\ssl\\mqipt.policy
```

2) 路由定义:

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
```

2. 启动 MQIPT1

打开命令提示符并输入下列内容:

```
c:
cd \mqipt\bin
mqipt ..
```

下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
MQCPI055 将 java.security.policy 设置为 c:\mqipt\mqipt.policy
MQCPI053 正在启动 Java 安全性管理器
MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....使用 MQ 协议
MQCPI078 路由 1415 用于连接请求准备就绪
```

3. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

4. 使用以下命令放入消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
Hello world <enter>
 <enter>
```

5. 使用以下命令取出消息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
```

您将看见 “Hello world”。

配置服务质量（QoS）

对于此示例，我们假设 TQoS 已与 MQIPT 安装在同一台机器上。

在此示例中，您将对 MQIPT 路由上的所有通道应用服务质量（QoS）。只有在 Linux 平台上运行 MQIPT 时，才能实现此应用。此样本将为从 MQIPT 发送到 WebSphere MQ 客户机的所有数据设置 “平均” 优先级，并为发送到 WebSphere MQ 服务器的所有数据设置 “良好” 优先级。使用下面列出的样本 pagent 策略，下列优先级可应用于 QosToCaller 和 QosToDest:

- 1 – 平均
- 2 – 良好
- 3 – 很好

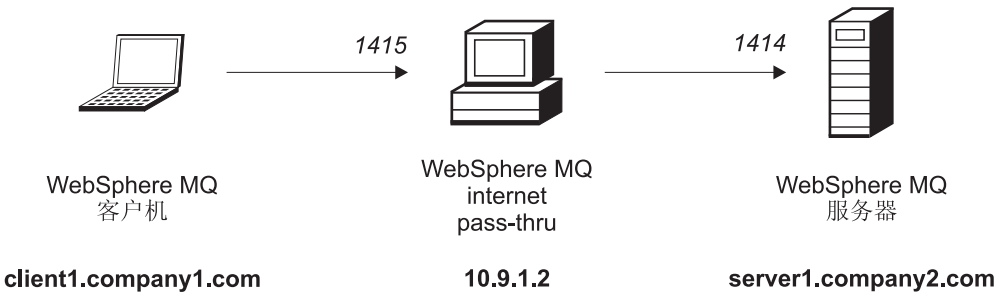

图 20. QoS 网络图

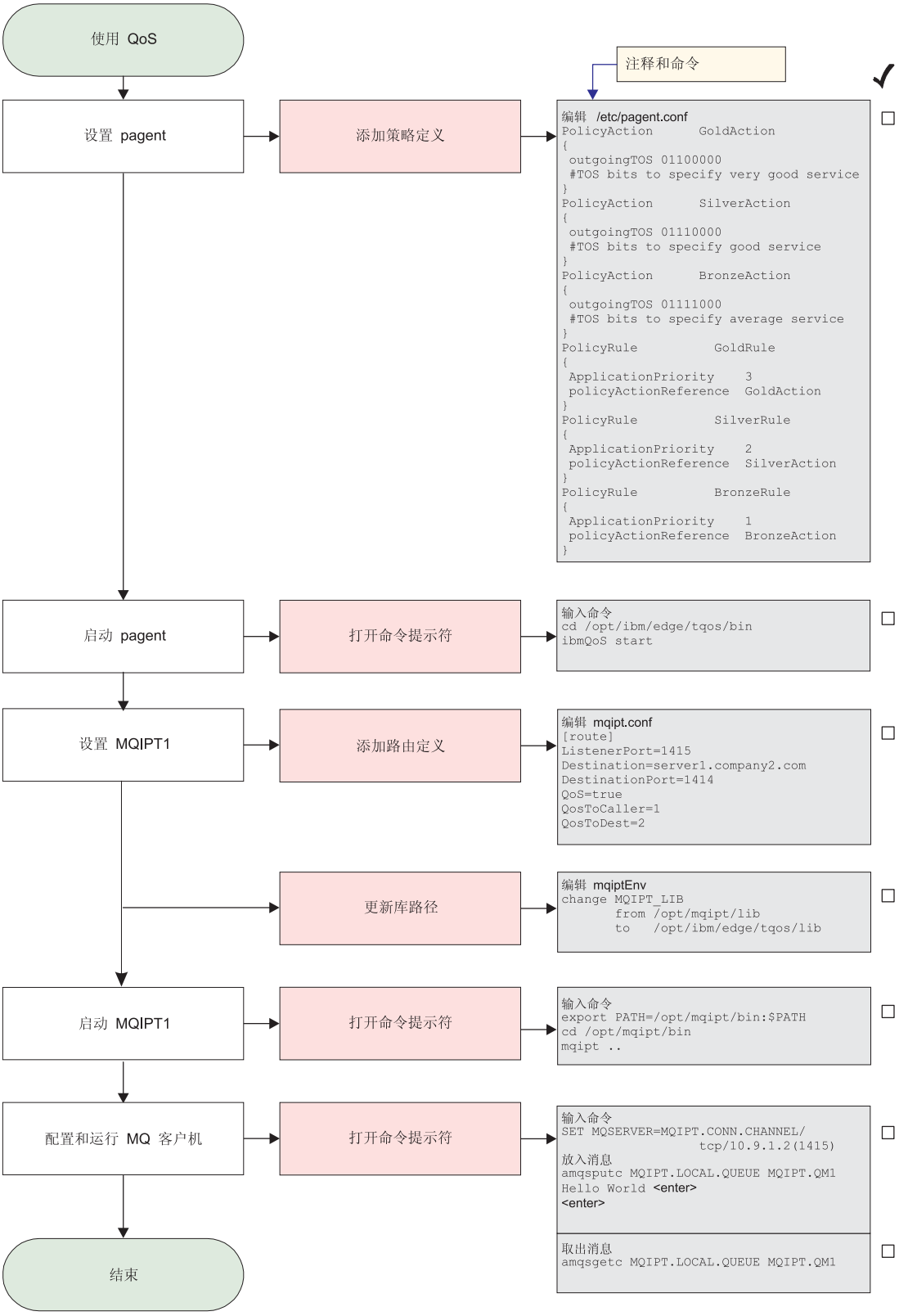

图 21. QoS 配置

1. 设置 pagent

   编辑 `/etc/pagent.conf` 并添加下列内容:

```
PolicyAction      GoldAction
{
 outgoingTOS 01100000
 #TOS bits to specify very good service
}
PolicyAction      SilverAction
{
 outgoingTOS 01110000
 #TOS bits to specify good service
}
PolicyAction      BronzeAction
{
 outgoingTOS 01111000
 #TOS bits to specify average service
}
PolicyRule          GoldRule
{
 ApplicationPriority    3
 policyActionReference  GoldAction
}
PolicyRule          SilverRule
{
 ApplicationPriority    2
 policyActionReference  SilverAction
}
PolicyRule          BronzeRule
{
 ApplicationPriority    1
 policyActionReference  BronzeAction
}
```

要打开上定义的规则的性能数据收集，请使用语句 PolicyPerformanceCollection 并启用它。请参阅 Pagent.conf 以获取该语句的描述和格式。

2. 启动 pagent

   打开命令提示符并输入下列内容:

   ```
   cd /opt/ibm/edge/tqos/bin
   ibmQoS start
   ```

3. 设置 MQIPT1

   编辑 mqipt.conf 并添加路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   QoS=true
   QosToCaller=1
   QosToDest=2
   ```

4. 更新库路径

   编辑 mqiptEnv（可在 /opt/mqipt/bin 中找到）并将 MQIPT_LIB 从:

   ```
   /opt/mqipt/lib
   ```

   更改为:

   ```
   /opt/ibm/edge/tqos/lib
   ```

5. 启动 MQIPT1

   打开命令提示符并输入下列内容:

   ```
   export PATH=/opt/mqipt/bin:$PATH
   cd /opt/mqipt/bin
   mqipt ..
   ```

下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在读取来自 /opt/mqipt/mqipt.conf 的配置信息
MQCPI011 路径 /opt/mqipt/logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....使用 MQ 协议
MQCPI049 ....QoS 优先级为 dest = 2，caller = 1
MQCPI078 路由 1415 用于连接请求准备就绪
```

6. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

7. 使用以下命令放入消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
Hello world <enter>
 <enter>
```

8. 使用以下命令取出消息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
```

您将看见 “Hello world”。

配置 SOCKS 代理

在此示例中，您可以将 MQIPT 作为 SOCKS 代理。在运行此样本之前必须 socks 化 WebSphere MQ 客户机，并且 SOCKS 配置必须指向作为 SOCKS 代理的 MQIPT。可以随意定义 MQIPT Destination 和 DestinationPort 属性，因为真正的目标在 socks 握手过程期间从 WebSphere MQ 客户机获取的。

启动之前，您必须 socks 化整台机器或仅 socks 化 WebSphere MQ 客户机应用程序（amqsputc / amqsgetc）。您还必须将 SOCKS 客户机配置为:

- 指向作为 Socks 代理的 MQIPT
- 启用 Socks V5 支持
- 禁用用户认证
- 仅连接到 MQIPT 网络地址

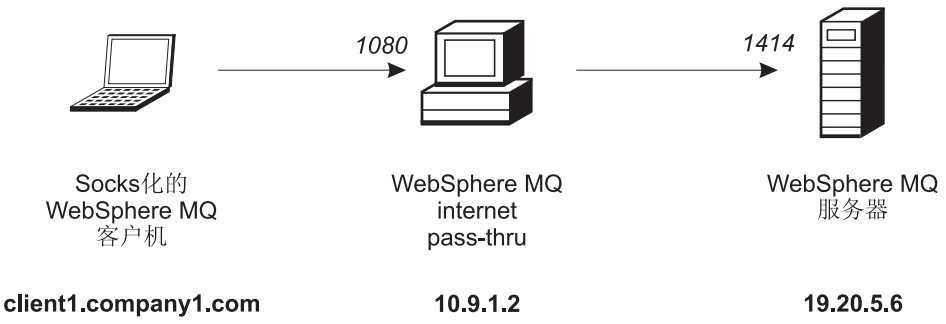

图 22. SOCKS 代理网络图

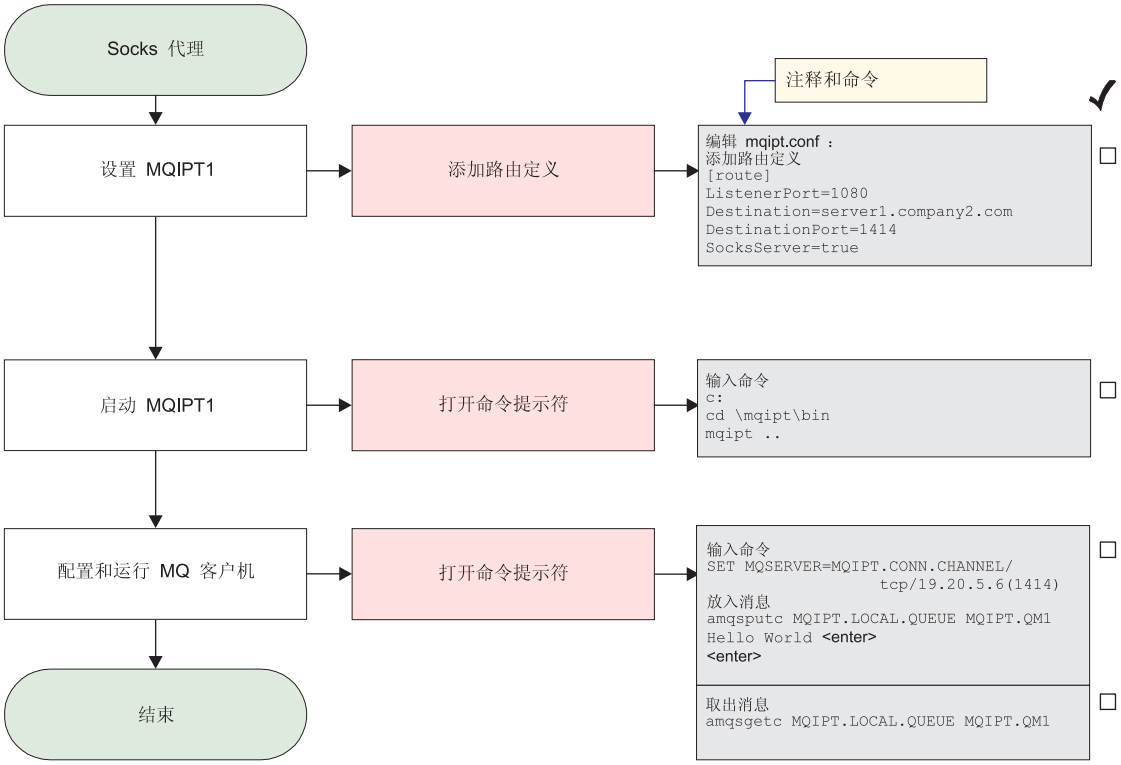

图 23. SOCKS 代理配置

1. 设置 MQIPT1

   编辑 mqipt.conf 并添加路由定义:

   ```
   [route]
   ListenerPort=1080
   Destination=server1.company2.com
   DestinationPort=1414
   SocksServer=true
   ```

2. 启动 MQIPT1

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \mqipt\bin
   mqipt ..
   ```

   下列消息表明成功完成:

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
   MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
   MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
   MQCPI006 路由 1080 已启动，并将转发消息至:
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....使用 MQ 协议
   MQCPI052 ....启用 Socks 服务器端
   MQCPI078 路由 1080 用于连接请求准备就绪
   ```

3. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/19.20.5.6(1414)
   ```

4. 使用以下命令放入消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
Hello world <enter>
 <enter>
```

5. 使用以下命令取出消息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
```

您将看见 “Hello world”。

配置 SOCKS 客户机

在此示例中，您将使用现有 SOCKS 代理运行 MQIPT（如同它已被 socks 化）。这与第 108 页的『配置 SOCKS 代理』类似，除了 MQIPT 进行的是 socks 化连接，而不是 WebSphere MQ 客户机。

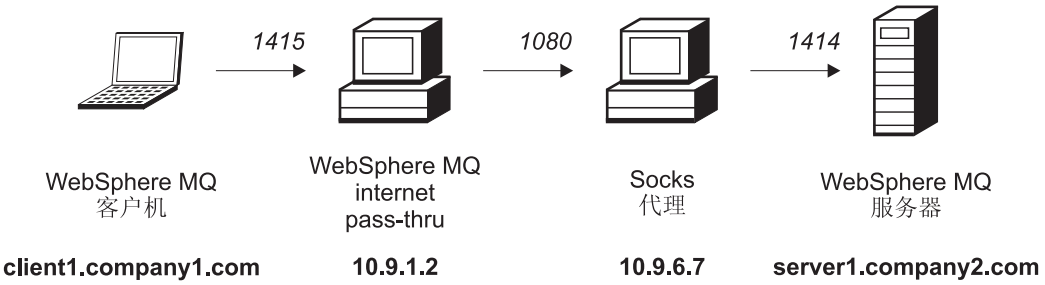

图 24. SOCKS 客户机网络图

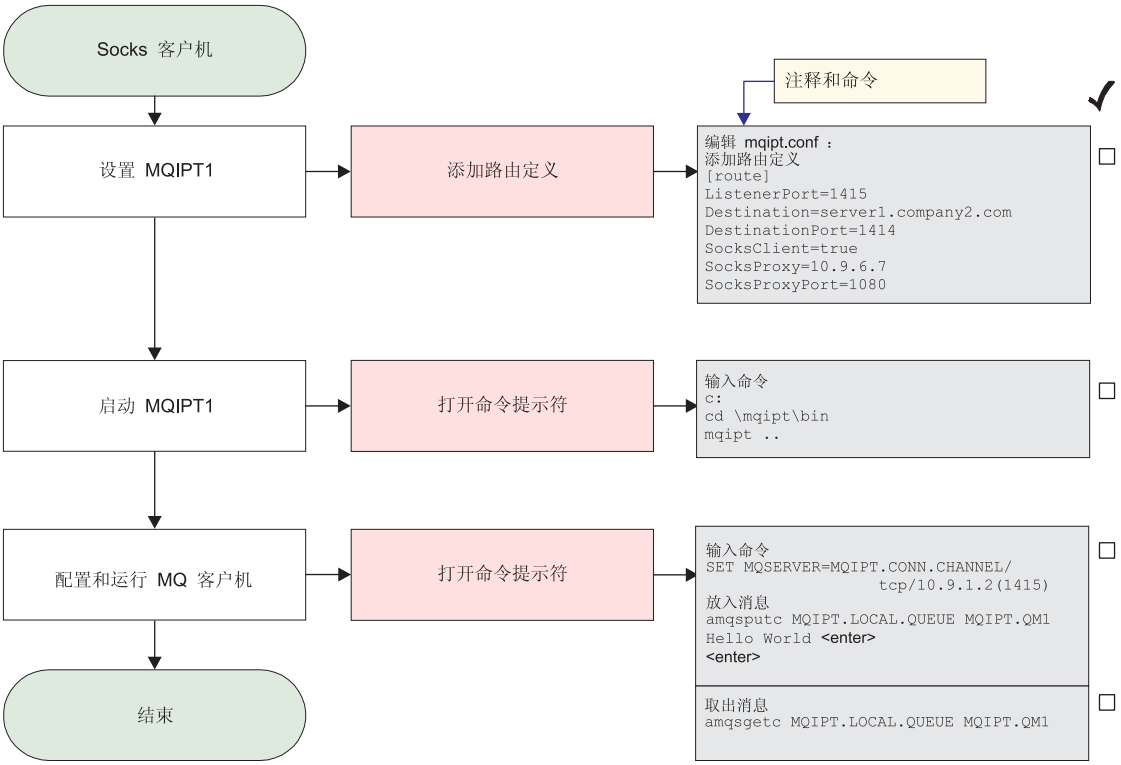

图 25. SOCKS 客户机配置

1. 设置 MQIPT1

编辑 mqipt.conf 并添加路由定义:

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SocksClient=true
SocksProxy=10.9.6.7
SocksProxyPort=1080
```

2. 启动 MQIPT1

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \mqipt\bin
   mqipt ..
   ```

   下列消息表明成功完成:

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
   MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
   MQCPI022 已对命令端口禁用密码检查
   MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
   MQCPI006 路由 1415 已启动，并将转发消息至:
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....使用 MQ 协议
   MQCPI039 ....和位于 10.9.6.7(1080) 的 Socks 代理
   MQCPI078 路由 1415 用于连接请求准备就绪
   ```

3. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

4. 使用以下命令放入消息:

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   Hello world <enter>
    <enter>
   ```

5. 使用以下命令取出消息:

   ```
   amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   ```

您将看见 “Hello world”。

创建 SSL 测试证书

在此示例中，我们将向您显示如何创建可用于测试 MQIPT 路由的自签署证书。此证书将打开信任为对等标志。

1. 启动 KeyMan
2. 选择 “新建...”
3. 选择 “PKCS#12 令牌”
4. 选择 “操作 -> 生成密钥”

   新密钥对将出现在列表 “RSA／1024 位” 中
5. 选择此新密钥对
6. 选择 “操作 -> 创建证书”
7. 选择 “自签署证书”
8. 输入证书详细信息。

   您将看到一个对话框，其中解释了专用证书将与密钥连接，您可以选择输入标号

9. 选择新证书
10. 显示证书的详细信息
11. 更改证书属性
12. 打开信任为对等标志
13. 关闭对话框，选择“文件”->“保存”
14. 输入密码（例如，myPassWord）
15. 输入新密钥环文件的文件名（例如，c:\mqipt\ssl\testRoute1414.pfx）

    您必须保留“文件格式为 PKCS#12／PFX” – **不要选取**“将密钥环包含在 Java 类中”
16. 创建一个包含您在上面所使用的密码（myPassWord）的文本文件。

    例如，c:\mqipt\ssl\testRoute1414.pwd

此密钥环文件现在就可在示例第 94 页的『SSL 服务器认证』中使用了。

配置 MQIPT Servlet

除了第 91 页的『假设』，本示例还作了以下假设:

- Tomcat Application Server 已安装在以下目录中:

  `c:\jakarta-tomcat-4.0.1`

  您可以从以下网址下载 Tomcat:

  `http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0.3/`
- IBM Web Traffic Express 已安装在:

  `c:\wte`

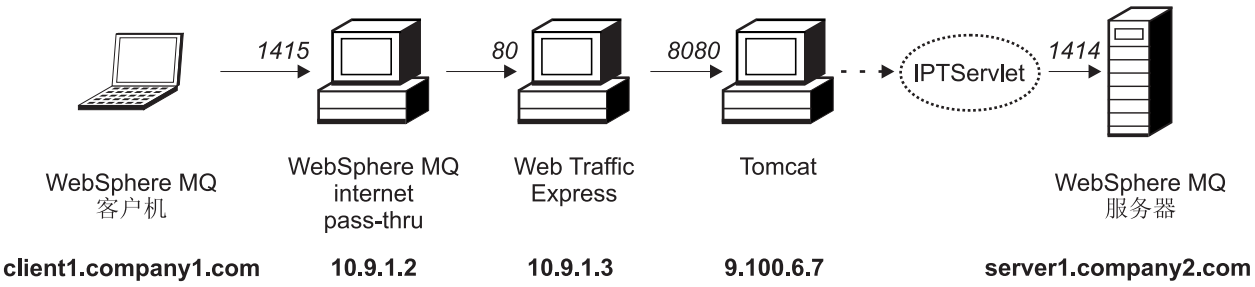

图 26. Servlet 网络图

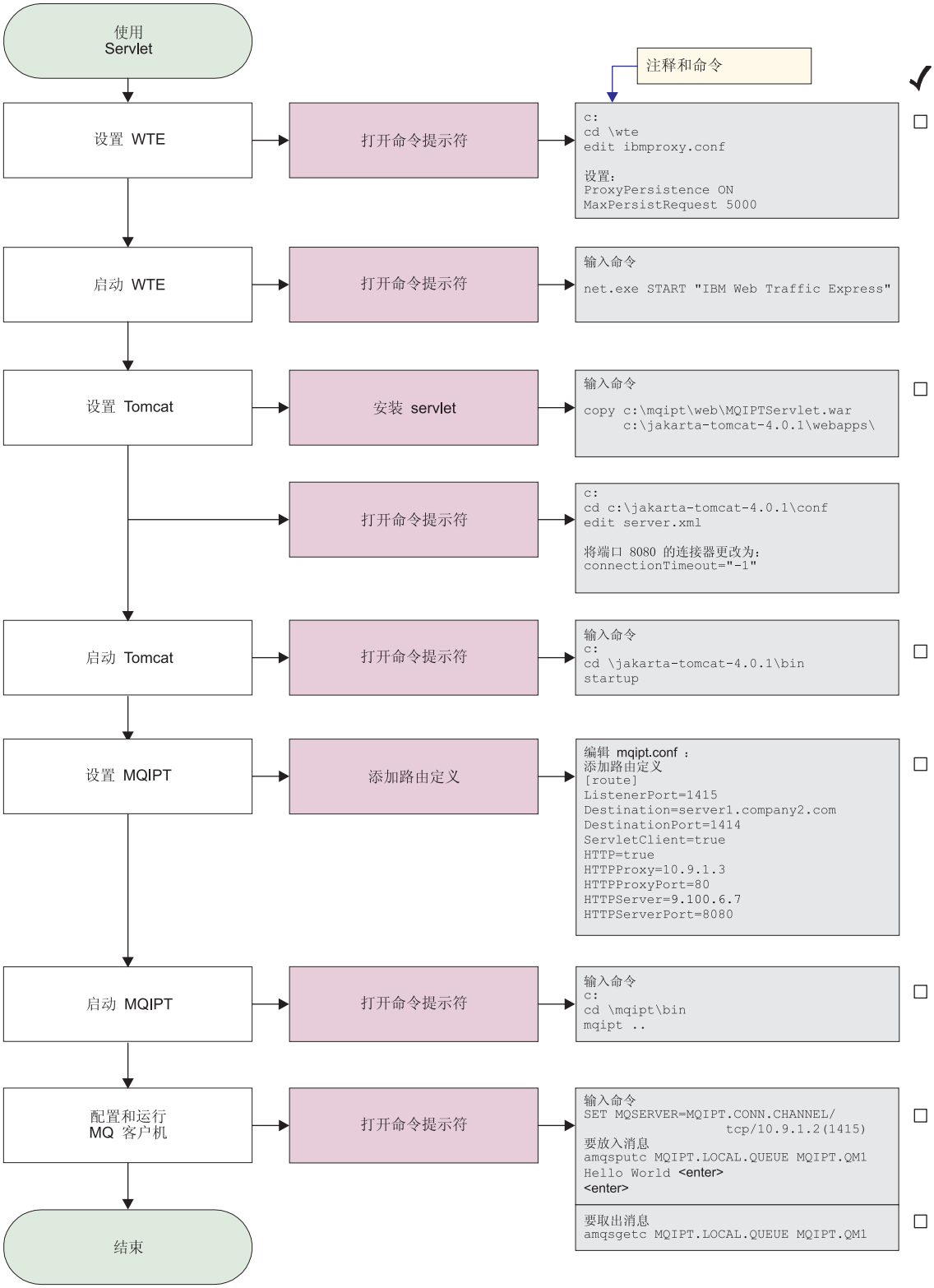

图 27. Servlet 配置

1. 安装 Web Traffic Express

   编辑 `c:\wte\ibmroxy.conf` 并设置以下属性:

```
ProxyPersistence ON
MaxPersistRequest  5000
```

2. 启动 Web Traffic Express

   打开命令提示符并输入下列内容:

   ```
   net.exe Start "IBM Web Traffic Express"
   ```

3. 设置 Tomcat

   要安装 Servlet，将

   ```
   c:\mqipt\web\MQIPTServlet.war
   ```

   复制为:

   ```
   c:\jakarta-tomcat-4.0.1\webapps
   ```

   编辑 c:\jakarta-tomcat-4.0.1\conf\server.xml，在端口 8443 上启用连接器，并将 ConnectionTimeout 属性设置为 -1。

4. 启动 Tomcat

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \jakarta-tomcat-4.0.1\bin
   startup
   ```

5. 设置 MQIPT1

   编辑 mqipt.conf 并添加路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   ServletClient=true
   HTTP=true
   HTTPProxy=10.9.1.3
   HTTPProxyPort=80
   HTTPServer=9.100.6.7
   HTTPServerPort=8080
   ```

6. 启动 MQIPT1

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \mqipt\bin
   mqipt ..
   ```

   下列消息表明成功完成:

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
   MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
   MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
   MQCPI006 路由 1415 已启动，并将转发消息至:
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....使用 HTTP
   MQCPI024 ....和位于 10.9.1.3(80) 的 HTTP 代理
   MQCPI066 ....和位于 9.100.6.7(8080) 的 HTTP 服务器
   MQCPI059 ....启用 servlet 客户机
   MQCPI078 路由 1415 用于连接请求准备就绪
   ```

7. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

8. 使用以下命令放入消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
Hello world <enter>
 <enter>
```

9. 使用以下命令取出消息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
```

您将看见 “Hello world”。

HTTPS 配置

除了第 91 页的『假设』，本示例还作了以下假设:

- Tomcat Application Server 已安装在以下目录中:

  ```
  c:\jakarta-tomcat-4.0.1
  ```

  您可以从以下网址下载 Tomcat:

  ```
  http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0.3/
  ```

- IBM Web Traffic Express 已安装在:

  ```
  c:\wte
  ```

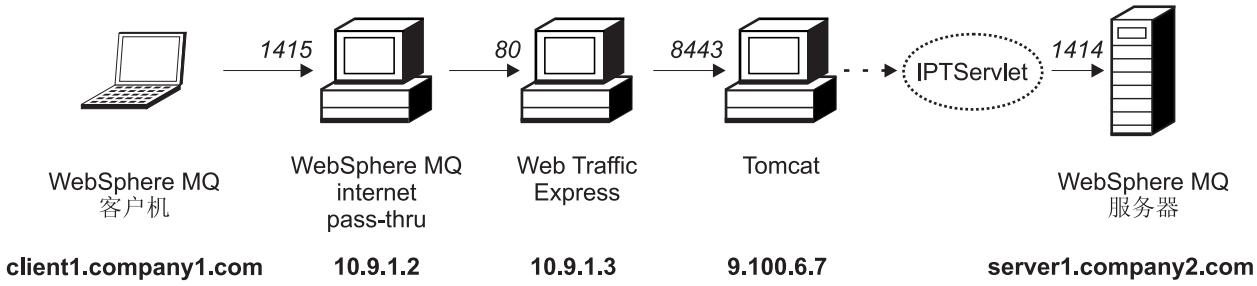

图 28. HTTPS 网络图

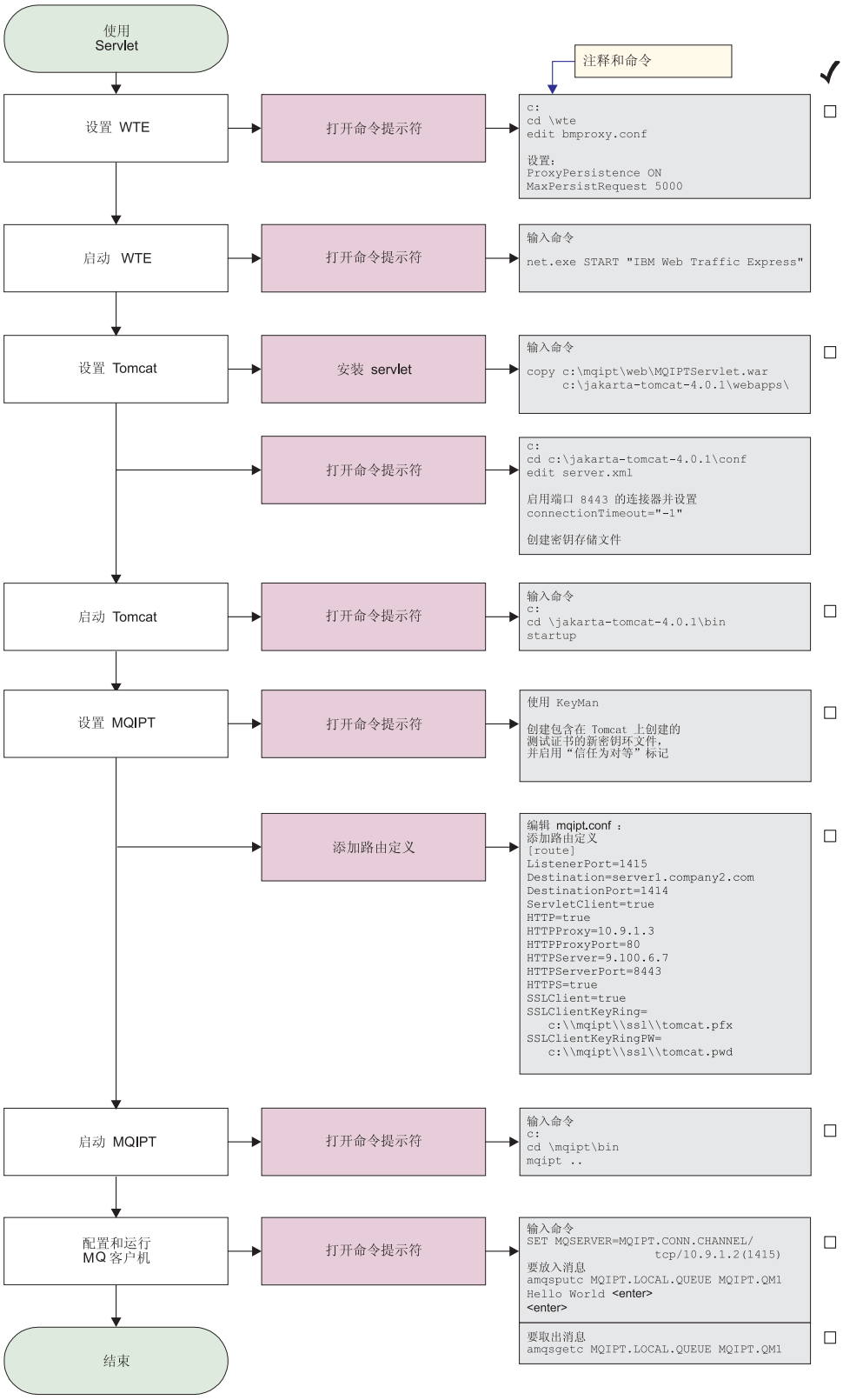

图 29. HTTPS 配置

1. 安装 Web Traffic Express

   编辑 `c:\wte\ibmroxy.conf` 并设置以下属性:

```
ProxyPersistence ON
MaxPersistRequest  5000
```

2. 启动 Web Traffic Express

   打开命令提示符并输入下列内容:

   ```
   net.exe Start "IBM Web Traffic Express"
   ```

3. 设置 Tomcat

   要安装 Servlet，将

   ```
   c:\mqipt\web\MQIPTServlet.war
   ```

   复制为:

   ```
   c:\jakarta-tomcat-4.0.1\webapps
   ```

   编辑 c:\jakarta-tomcat-4.0.1\conf\server.xml，在端口 8443 上启用连接器，并将 ConnectionTimeout 属性设置为 -1。

   使用可从以下地址获取的 Tomcat 文档:

   ```
   http://jakarta.apache.org/tomcat/tomcat-4.0-doc/index.html
   ```

   然后按 “SSL Configuration HOW-TO” 中的说明在端口 8443 上启用 SSL 连接。创建包含测试自签署证书的密钥环文件，这将创建称为 C:\winnt\profiles\<userid>\.keystore 的文件。

4. 启动 Tomcat

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \jakarta-tomcat-4.0.1\bin
   startup
   ```

5. 将新的密钥存储文件从 Tomcat 机器复制到 MQIPT 机器。使用 KeyMan 打开新的密钥存储文件（缺省密码为 changeit），并打开 “信任为对等” 标记（请参阅第 111 页的『创建 SSL 测试证书』以获取更多信息）。将此文件保存为 c:\mqipt\ssl\tomcat.pfx，并创建称为 c:\mqipt\ssl\tomcat.pwd 的文本文件，该文件包含密码 changeit。

6. 设置 MQIPT1

   编辑 mqipt.conf 并添加路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   ServletClient=true
   HTTP=true
   HTTPProxy=10.9.1.3
   HTTPProxyPort=80
   HTTPServer=9.100.6.7
   HTTPServerPort=8443
   HTTPS=true
   SSLClient=true
   SSLClientKeyRing=c:\\mqipt\\ssl\\tomcat.pfx
   SSLClientKeyRingPW=c:\\mqipt\\ssl\\tomcat.pwd
   ```

7. 启动 MQIPT1

   打开命令提示符并输入下列内容:

```
c:
cd \mqipt\bin
mqipt ..
```

下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....使用 HTTP
MQCPI024 ....和位于 10.9.1.3(80) 的 HTTP 代理
MQCPI066 ....和位于 9.100.6.7(8080) 的 HTTP 服务器
MQCPI059 ....启用 servlet 客户机
MQCPI036 ....使用下列属性启用 SSL 客户机端:
MQCPI031 ......密码套件 <null>
MQCPI032 ......密钥环文件 c:\mqipt\ssl\tomcat.pfx
MQCPI047 ......CA 密钥环文件 <null>
MQCPI038 ......专有名称 CN=* O=* OU=* L=* ST=* C=*
MQCPI078 路由 1415 用于连接请求准备就绪
```

8. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

9. 使用以下命令放入消息:

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   Hello world <enter>
    <enter>
   ```

10. 使用以下命令取出消息:

    ```
    amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
    ```

您将看见 “Hello world”。

配置 MQIPT 群集支持

对于此示例，除了第 91 页的『假设』以外，您还必须已执行下列步骤:

在 WebSphere MQ 服务器 LONDON 上:

- 定义了一个名为 LONDON 的队列管理器
- 定义了一个名为 MQIPT.CONN.CHANNEL 的服务器连接通道
- 在端口 1414 上为 LONDON 启动了 TCP/IP 侦听器
- Socks 化此队列管理器

在 WebSphere MQ 服务器 NEWYORK 上:

- 定义了一个名为 NEWYORK 的队列管理器
- 定义了一个名为 MQIPT.CONN.CHANNEL 的服务器连接通道
- 在端口 1414 上为 NEWYORK 启动了 TCP/IP 侦听器
- Socks 化此队列管理器

若要 socks 化此列管理器，请 socks 化整台机器或仅 socks 化 WebSphere MQ 服务器应用程序。将 SOCKS 客户机配置为:

- 指向作为 SOCKS 代理的 MQIPT
- 启用 SOCKS V5 支持

- 禁用用户认证
- 仅远程连接到 MQIPT

只有一个应用程序可以在同一机器的给定端口地址上侦听，如果端口 1414 已在使用，则选择空闲端口地址并在示例中替换它。一旦您已完成这一操作，您可以通过将一条消息放入 LONDON 上的本地队列，并从 NEWYORK 中检索该消息，从而测试队列管理器之间的路由。

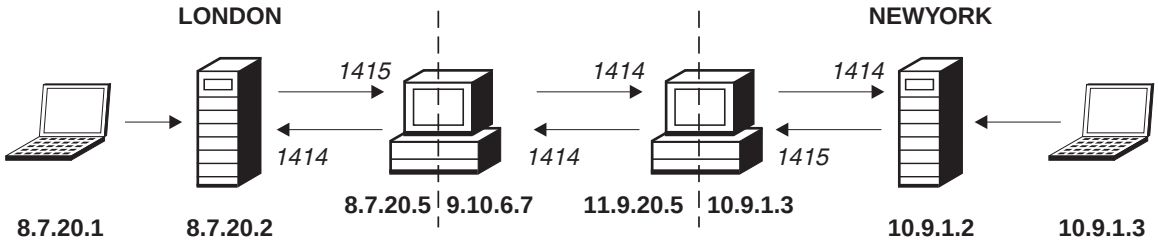

图 30. 群集网络图

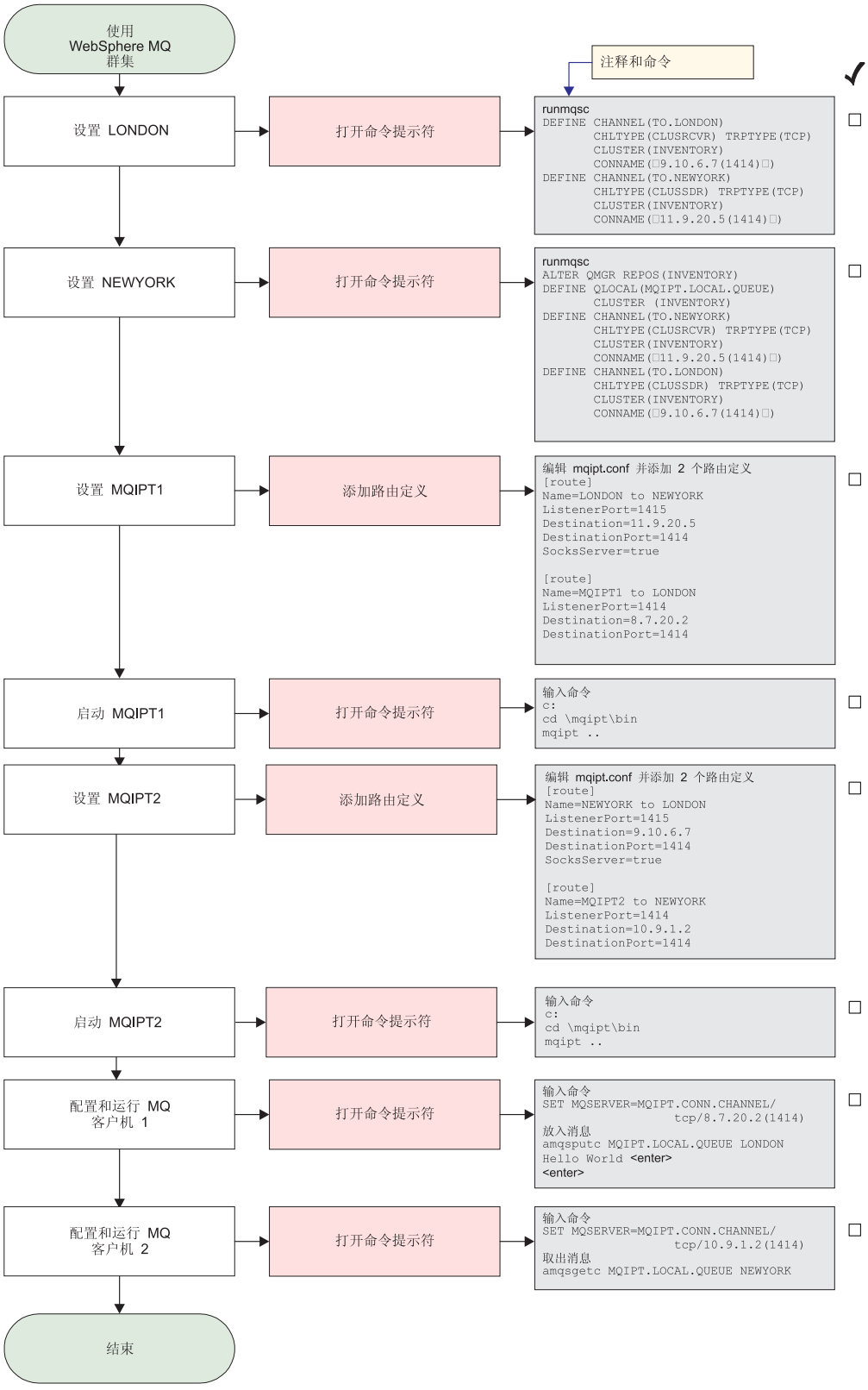

图 31. 群集配置

1. 设置 LONDON

   打开命令提示符并输入下列内容:

```
runmqsc
DEFINE CHANNEL(TO.LONDON) +
       CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
       CLUSTER(INVENTORY) +
       CONNAME('9.10.6.7(1414)')
DEFINE CHANNEL(TO.NEWYORK) +
       CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
       CLUSTER(INVENTORY) +
       CONNAME('11.9.20.5(1414)')
```

2. 设置 NEWYORK

   打开命令提示符并输入下列内容:

```
runmqsc
ALTER QMGR REPOS(INVENTORY)
DEFINE QLOCAL(MQIPT.LOCAL.QUEUE) +
       CLUSTER(INVENTORY)
DEFINE CHANNEL(TO.NEWYORK) +
       CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
       CLUSTER(INVENTORY) +
       CONNAME('11.9.20.5(1414)')
DEFINE CHANNEL(TO.LONDON) +
       CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
       CLUSTER(INVENTORY) +
       CONNAME('9.10.6.7(1414)')
```

3. 设置 MQIPT1

   编辑 mqipt.conf 并添加两个路由定义:

```
[route]
Name=LONDON to NEWYORK
ListenerPort=1415
Destination=11.9.20.5
DestinationPort=1414
SocksServer=true
```

```
[route]
Name=MQIPT1 to LONDON
ListenerPort=1414
Destination=8.7.20.2
DestinationPort=1414
```

4. 启动 MQIPT1

   打开命令提示符并输入下列内容:

```
c:
cd \mqipt\bin
mqipt ..
```

   下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....11.9.20.5(1414)
MQCPI035 ....使用 MQ 协议
MQCPI052 ....启用 Socks 服务器端
MQCPI078 路由 1415 用于连接请求准备就绪
MQCPI006 路由 1414 已启动，并将转发消息至:
MQCPI034 ....8.7.20.2(1414)
MQCPI035 ....使用 MQ 协议
MQCPI078 路由 1414 用于连接请求准备就绪
```

5. 设置 MQIPT2

   编辑 mqipt.conf 并添加两个路由定义:

```
[route]
Name=NEWYORK to LONDON
ListenerPort=1415
Destination=9.10.6.7
DestinationPort=1414
SocksServer=true
[route]
Name=MQIPT2 to NEWYORK
ListenerPort=1414
Destination=10.9.1.2
DestinationPort=1414
```

6. 启动 MQIPT2

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \mqipt\bin
   mqipt ..
   ```

   下列消息表明成功完成:

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
   MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
   MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
   MQCPI006 路由 1415 已启动，并将转发消息至:
   MQCPI034 ....9.10.6.7(1414)
   MQCPI035 ....使用 MQ 协议
   MQCPI052 ....启用 Socks 服务器端
   MQCPI078 路由 1415 用于连接请求准备就绪
   MQCPI006 路由 1414 已启动，并将转发消息至:
   MQCPI034 ....10.9.1.2(1414)
   MQCPI035 ....使用 MQ 协议
   MQCPI078 路由 1414 用于连接请求准备就绪
   ```

7. 在第一个 WebSphere MQ 客户机（8.7.20.1）的命令提示符下，输入下列内容:

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/8.7.20.2(1414)
   ```

8. 使用以下命令放入消息:

   ```
   amqsputc MQIPT.LOCAL.QUEUE LONDON
   Hello world <enter>
    <enter>
   ```

9. 在第二个 WebSphere MQ 客户机（10.9.1.3）的命令提示符下，输入下列内容:

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1414)
   ```

10. 在第二个 WebSphere MQ 客户机上，使用以下命令取出此消息:

    ```
    amqsgetc MQIPT.LOCAL.QUEUE NEWYORK
    ```

您将看见 “Hello world”。

创建密钥环文件

此样本假设您已使用 Ketman 请求来自可信 CA 的新证书，并且在文件中（例如，`server.cer`）您的个人证书已返回给您。这将足够执行服务器认证。如果您需要客户机认证，则您需请求第二个证书（例如，`client.cer`）并执行下列步骤两次，以创建两个密钥环文件。

1. 启动 KeyMan
2. 选择 “新建...”
3. 选择 “PKCS#12 令牌”
4. 选择 “操作 -> 生成密钥”

新密钥对将出现在列表“RSA／1024 位”中

5. 选择此新密钥对
6. 选择“操作 -> 请求证书”

   按照联机说明进行操作
7. 选择“文件 -> 保存”
8. 输入密码
9. 输入新密钥环文件的文件名

   例如，c:\mqipt\ssl\myServer.pfx
10. 保留“文件格式为 PKCS#12 / PFX” - **不要选取**“将密钥环包含在 Java 类中”
11. 选择“文件 -> 退出”
12. 创建一个包含您在上面所使用的密码（myPassWord）的文本文件。

    例如，c:\mqipt\ssl\myServer.pwd

当您取回您的证书时，打开原始密钥环文件（myServer.pfx 文件）。然后:

1. 启动 KeyMan
2. 选择“打开现有...”.
3. 选择“本地资源”
4. 选择“打开文件...”
5. 输入个人证书文件的文件名

   例如，c:\mqipt\ssl\myServer.pfx
6. 输入密码
7. 选择“文件 -> 导入”
8. 选择“本地资源”
9. 选择“打开文件...”
10. 输入 server.cer

    您会看到一个对话框，其中解释专用证书将与专用密钥连接。
11. 选择“文件 -> 保存”
12. 选择“文件 -> 退出”

重复这些步骤，以从 client.cer 文件创建 myClient.pfx。使用 KeyMan 检查样本 CA 密钥环文件 sslCAdefault.pfx 的内容，以查看您的个人证书是否由所列出的其中一个 CA 签署。如果是的话，则您可以使用此样本 CA 密钥环文件。如果不是，则您需要创建一个包含签署您个人证书的公用 CA 证书的密钥环文件。它可能已与您的个人证书一起返回。如果没有，您将需要向为您提供个人证书的 CA 请求 CA 证书，并将其导入该证书 sslCAdefault.pfx。此 CA 密钥环文件既可以用于客户机也可以用于服务器端。要使用这些新密钥环文件进行服务器认证，请参阅示例第 94 页的『SSL 服务器认证』示例并设置下列路由属性:

```
SSLClientCAKeyRing=c:\\mqipt\\ssl\\sslCAdefault.pfx
SSLClientCAKeyRingPW=c:\\mqipt\\ssl\\sslCAdefault.pwd
SSLServerKeyRing=c:\\mqipt\\ssl\\myServer.pfx
SSLServerKeyRingPW=c:\\mqipt\\ssl\\myServer.pwd
SSLServerCAKeyRing=c:\\mqipt\\ssl\\sslCAdefault.pfx
SSLServerCAKeyRingPW=c:\\mqipt\\ssl\\sslCAdefault.pwd
```

要使用这些新密钥环文件进行客户机及服务器认证，请参阅第 97 页的『SSL 客户机认证』示例并设置下列路由属性:

```
SSLClientKeyRing=c:\\mqipt\\ssl\\myClient.pfx
SSLClientKeyRingPW=c:\\mqipt\\ssl\\myClient.pwd
SSLClientCAKeyRing=c:\\mqipt\\ssl\\sslCAdefault.pfx
SSLClientCAKeyRingPW=c:\\mqipt\\ssl\\sslCAdefault.pwd
SSLServerKeyRing=c:\\mqipt\\ssl\\myServer.pfx
SSLServerKeyRingPW=c:\\mqipt\\ssl\\myServer.pwd
SSLServerCAKeyRing=c:\\mqipt\\ssl\\sslCAdefault.pfx
SSLServerCAKeyRingPW=c:\\mqipt\\ssl\\sslCAdefault.pwd
```

分配端口地址

本示例显示了如何控制生成外出连接时使用的本地端口地址。对于本示例，我们假设您将 MQIPT 安装在多主机机器上。

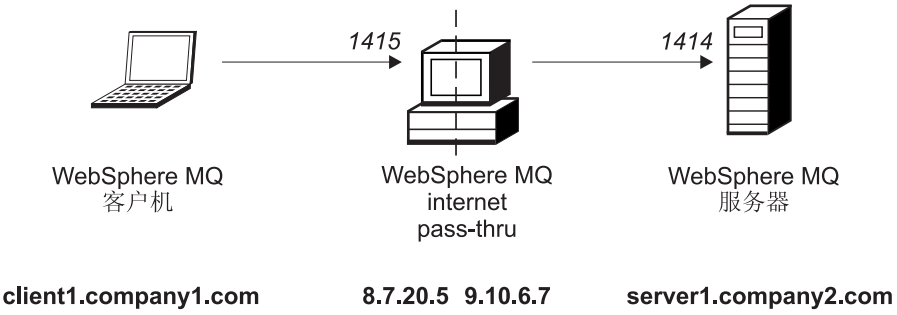

图 32. 端口分配网络图

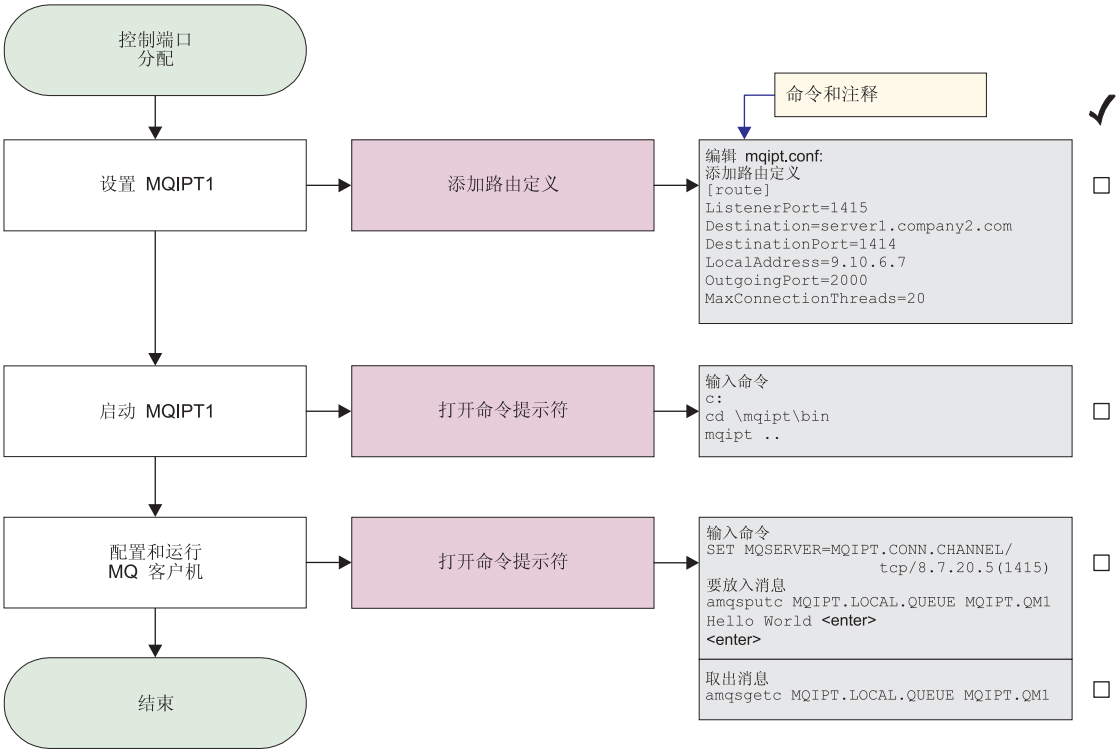

图 33. 端口分配配置

1. 设置 MQIPT1

   编辑 mqipt.conf 并添加路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   LocalAddress=9.10.6.7
   OutgoingPort=2000
   MaxConnectionThreads=20
   ```

2. 启动 MQIPT1

   打开命令提示符并输入下列内容:

   ```
   c:
   cd \mqipt\bin
   mqipt ..
   ```

   下列消息表明成功完成:

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
   MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
   MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
   MQCPI006 路由 1415 已启动，并将转发消息至:
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....使用 MQ 协议
   MQCPI069 ....正在绑定到本地地址 9.10.6.7
   MQCPI070 ....正在使用本地端口地址范围 2000-2019
   MQCPI078 路由 1415 用于连接请求准备就绪
   ```

3. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/8.7.20.5(1415)
   ```

4. 使用以下命令放入消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
Hello world <enter>
 <enter>
```

5. 使用以下命令取出消息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
```

您将看见 “Hello world”。

使用 LDAP 服务器

此样本演示了如何配置 MQIPT 以使用 LDAP 服务器来检索 CRL。此样本的意图不是用于说明如何安装和设置 LDAP 服务器或如何创建包含私人或可信证书的密钥环文件。假设 LDAP 服务器从已知和可信的认证中心（CA）可用。没有使用备份 LDAP 服务器，但可通过添加合适的路由属性，容易地实现使用备份 LDAP 服务器。

对于本示例，我们作了以下假设:

- IPT2 有一张由可信 CA 发出的个人证书，该证书存储在称为 myCert.pfx 的密钥环文件中，用于打开密钥环文件的加密过的密码存储在文件 myCert.pwd 中。
- IPT1 有一份可信 CA 证书的副本，它将用于认证 IPT2 发送的证书。该证书存储在称为 caCert.pfx 的密钥环文件中，用于打开密钥环文件的加密过的密码存储在文件 caCert.pwd 中。
- 已使用 mqiptPW 脚本创建了加密的密码文件。

运行此样本将使 WMQ 客户机能连接到队列管理器（QM）并将 WMQ 消息放入目标队列。在 IPT1 上运行 MQIPT 跟踪将显示使用的 LDAP 服务器，但要演示 CRL 是如何工作的，可信 CA 需要撤销 IPT2 使用的个人证书。则，在这种情况下，不会允许 WMQ 客户机连接到 QM，因为从 IPT1 到 IPT2 的连接将被拒绝。

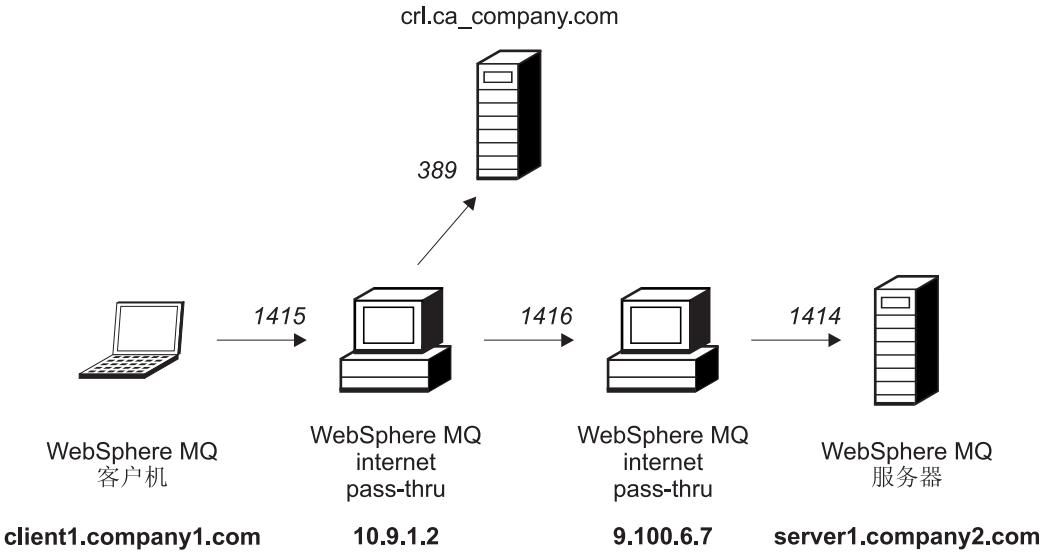

图 34. LDAP 服务器网络图

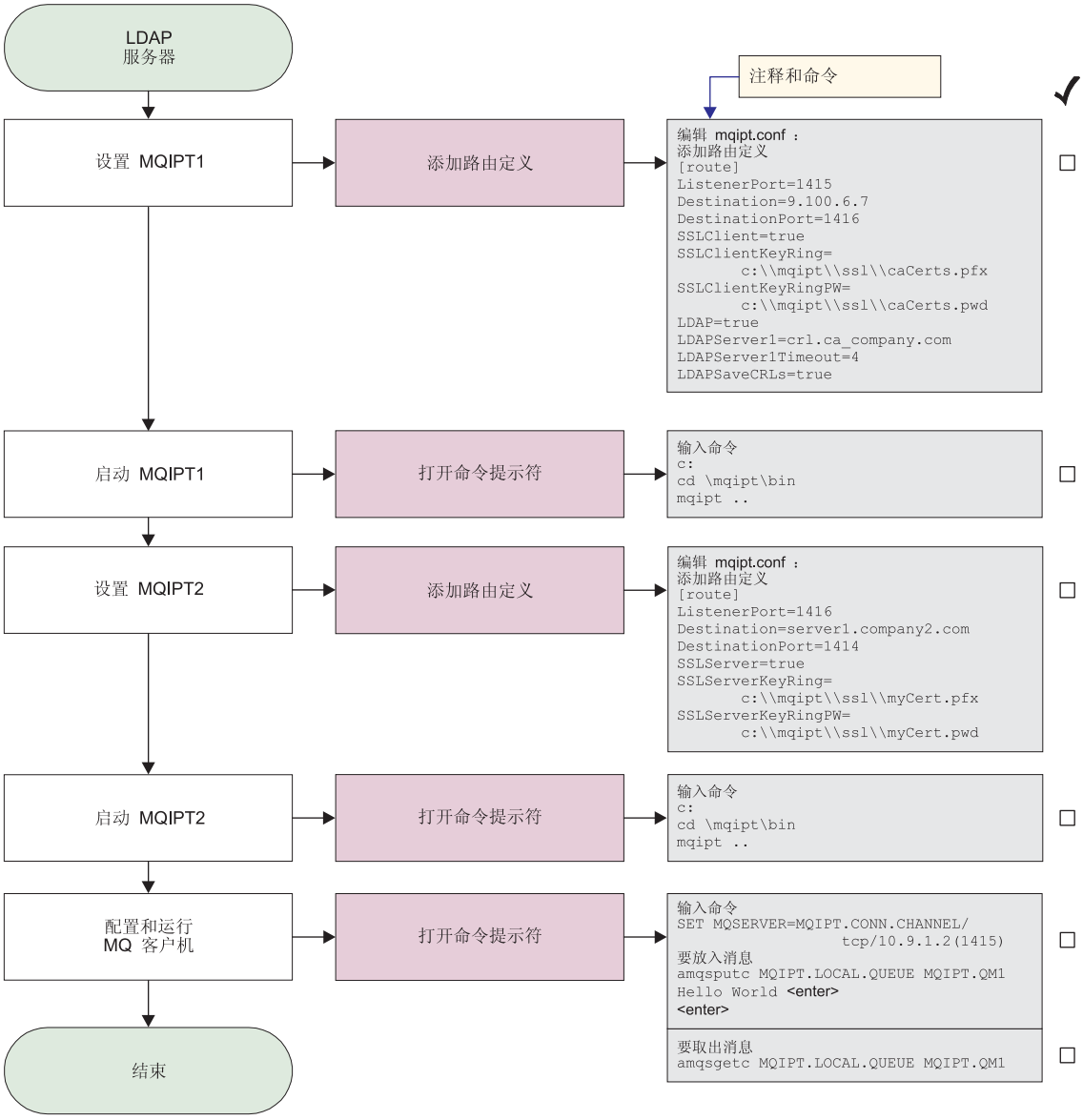

图 35. LDAP 服务器配置

1. 在 IPT1 上

   编辑 mqipt.conf 并添加路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=9.100.6.7
   DestinationPort=1416
   SSLClient=true
   SSLClientKeyRing=c:\\mqipt\\ssl\\caCerts.pfx
   SSLClientKeyRingPW=c:\\mqipt\\ssl\\caCerts.pwd
   LDAP=true
   LDAPServer1=crl.ca_company.com
   LDAPServer1Timeout=4
   LDAPSaveCRLs=true
   ```

   打开命令提示符:

```
c:
cd \mqipt\bin
mqipt ..
```

下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....9.100.6.7(1416)
MQCPI035 ....使用 MQ 协议
MQCPI036 ....使用下列属性启用 SSL 客户机端:
MQCPI031 ......密码套件 <NULL>
MQCPI032 ......密钥环文件 <NULL>
MQCPI047 ......CA 密钥环文件 c:\mqipt\ssl\caCerts.pfx
MQCPI071 ......站点证书使用 CN=* O=* OU=* L=* ST=* C=*
MQCPI038 ......对等证书使用 CN=* O=* OU=* L=* ST=* C=*
MQCPI075 ....在 crl.ca_company.com(389) 处的 LDAP 主服务器
MQCPI086 ......超时 4 秒
MQCPI084 ....CRL 高速缓存到期超时为 1 小时
MQCPI085 ....CRL 将保存在密钥环文件中
MQCPI078 路由 1415 用于连接请求准备就绪
```

2. 在 IPT2 上

   编辑 mqipt.conf 并添加路由定义:

```
[route]
ListenerPort=1416
Destination=server1.company2.com
DestinationPort=1414
SSLServer=true
SSLServerKeyRing=c:\\mqipt\\ssl\\myCert.pfx
SSLServerKeyRingPW=c:\\mqipt\\ssl\\myCert.pwd
```

   打开命令提示符:

```
c:
cd \mqipt\bin
mqipt ..
```

   下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 IBM WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1416 正在启动，并将转发消息至:
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....使用 MQ 协议
MQCPI037 ....使用下列属性启用 SSL 服务器端:
MQCPI031 ......密码套件 <NULL>
MQCPI032 ......密钥环文件 c:\mqipt\ssl\myCert.pfx
MQCPI047 ......CA 密钥环文件 <NULL>
MQCPI071 ......站点证书使用 CN=* O=* OU=* L=* ST=* C=*
MQCPI038 ......对等证书使用 CN=* O=* OU=* L=* ST=* C=*
MQCPI033 ......客户机认证设置为 false
MQCPI078 路由 1416 用于连接请求准备就绪
```

3. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
```

4. 使用以下命令放入消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world <enter>
<enter>
```

5. 使用以下命令取出消息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

您将看见 “Hello world”。

SSL 代理方式

此样本演示如何以 SSL 代理方式运行 MQIPT，因此它将接受来自 SSL 客户机的 SSL 连接请求，并将其通过隧道传递到 SSL 服务器。假设 WMQ 客户机和服务器都是 V5.3 且配置为使用 SSL 连接。

要获取为 WMQ 设置 SSL 的进一步信息，请参阅 “WebSphere MQ Security Version 5.3” SC34-6079-01。

对于本示例，我们作了以下假设:

- MQClient 和 QM 已设置为使用 SSL 通道。

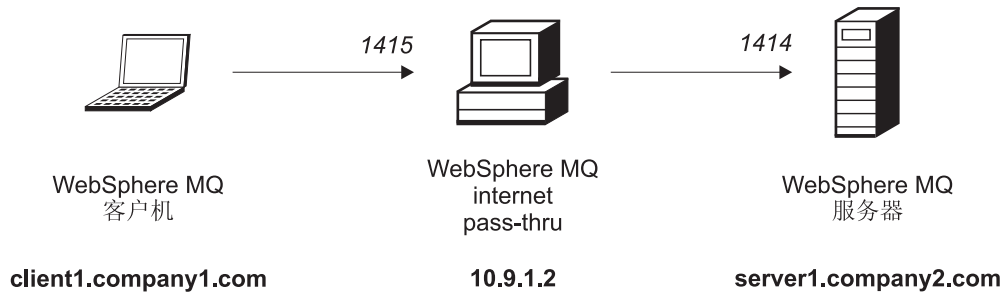

图 36. SSL 代理方式网络图

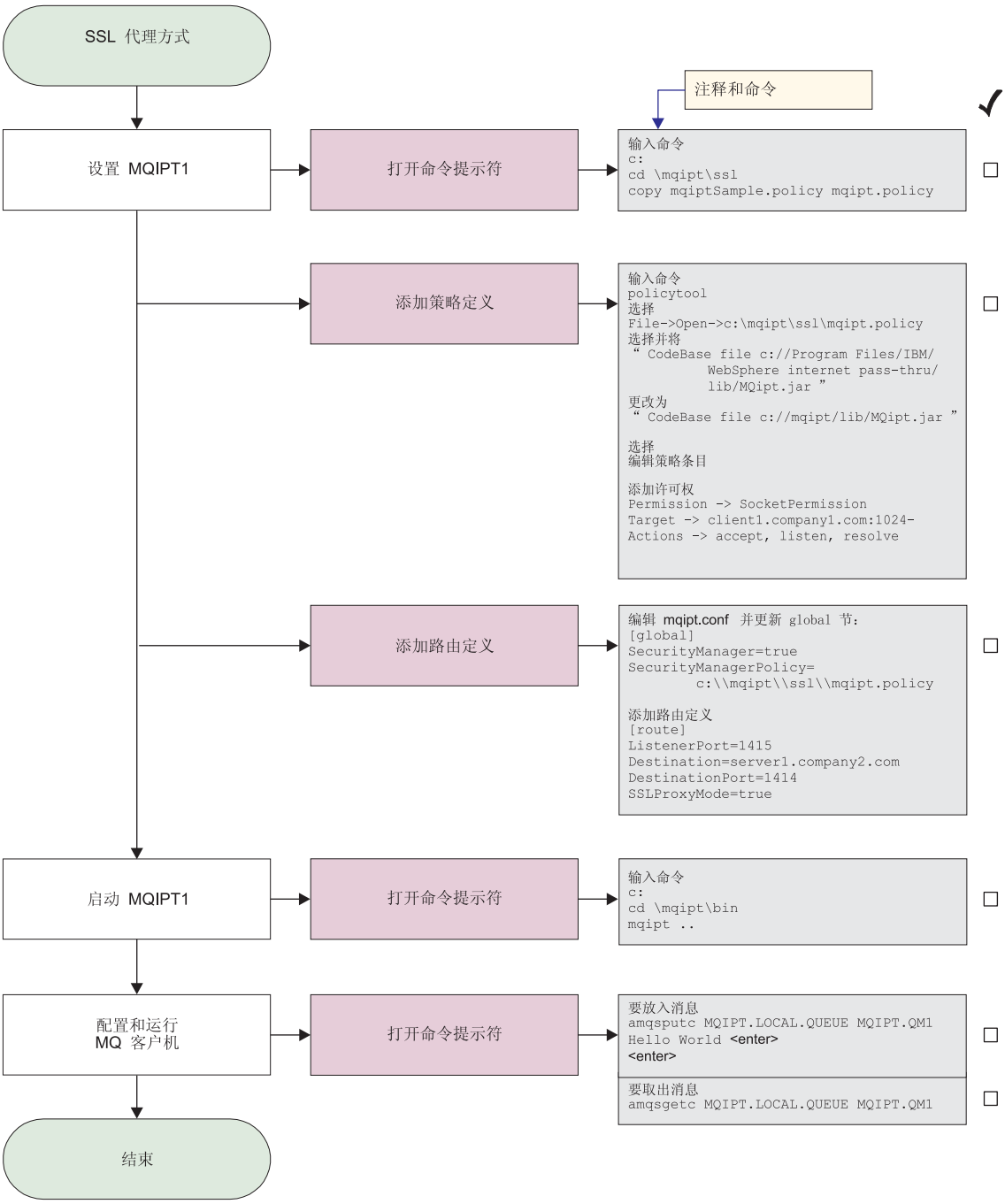

图 37. SSL 代理方式配置

1. 在 IPT1 上
   a. 打开命令提示符并输入下列内容:

      ```
      copy c:\mqipt\ssl\mqiptSample.policy to mqipt.policy
      ```

   b. 使用以下命令添加策略定义:

      ```
      policytool
      ```

      1) 选择: **“File” —> “Open” —> c:\mqipt\ssl\mqipt.policy**
      2) 选择:

         “file://C:/Program Files/IBM/WebSphere MQ internet pass-thru/lib/MQipt.jar”

3) 将 CodeBase 从:

```
"file://C:/Program Files/IBM/WebSphere MQ internet pass-thru/lib/MQipt.jar"
```

更改为:

```
"file://C:/mqipt/lib/MQipt.jar"
```

4) 将所有许可权从:

```
"C:\\Program Files\\IBM\\WebSphere MQ internet pass-thru"
```

更改为:

```
"C:\\mqipt"
```

5) 添加 SocketPermission:

```
Permission=SocketPermission
Target = "client1.company1.com:1024-"
Actions = "accept, listen, resolve"
```

2. 编辑 mqipt.conf，将下列两个属性添加到 global 节和路由定义:

```
[global]
SecurityManager=true
SecurityManagerPolicy=c:\\mqipt\\ssl\\mqipt.policy

[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SSLProxyMode=true
```

3. 打开命令提示符:

```
c:
cd \mqipt\bin
mqipt ..
```

下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....使用 SSLProxyMode
MQCPI078 路由 1415 用于连接请求准备就绪
```

4. 使用以下命令放入消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world <enter>
<enter>
```

5. 使用以下命令取出消息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

您将看见 “Hello world”。

Apache 重写

对于本示例，我们作了以下假设:

- Apache HTTP 服务器安装在 c:\apache
- IBM Web Traffic Express 已安装在 c:\wte

此样本显示了如何使用重写伪指令将 HTTP 请求转换为内部 Apache 代理重定向。必须装入代理和重写模块，但因为 Apache 不是真正以代理方式工作的，所有代理伪指令可保留为注释。

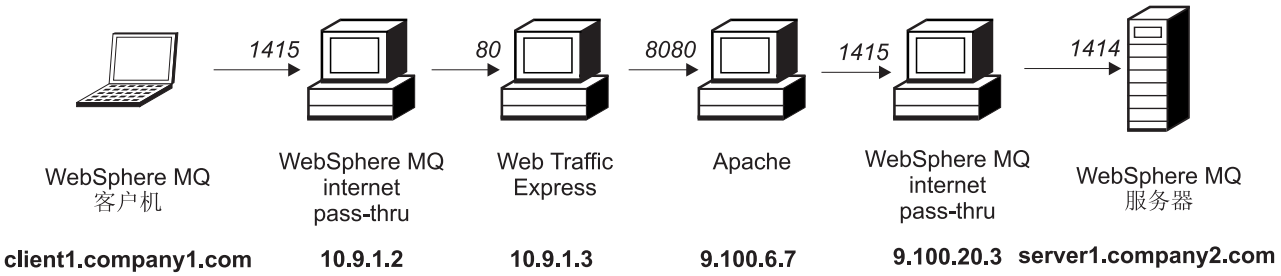

图 38. Apache 重写网络图

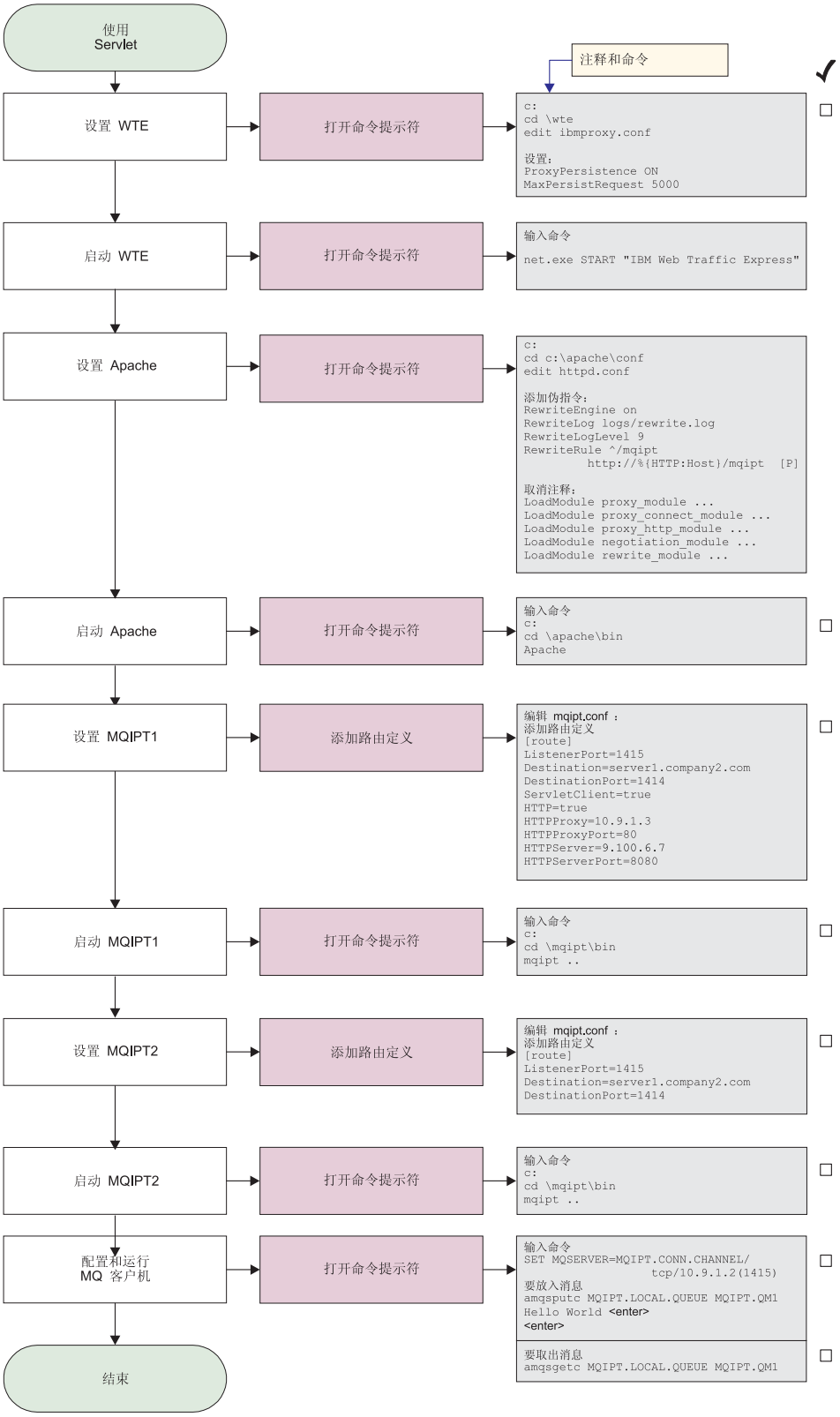

图 39. Apache 重写配置

1. 在 WTE 上

   编辑 `c:\wte\ibmroxy.conf`

更改下列属性:

```
ProxyPersistence ON
MaxPersistRequest  5000
```

2. 在 Apache 上

   编辑 c:\apache\conf\httpd.conf

```
RewriteEngine on
RewriteLog logs/rewrite.log
RewriteLogLevel 9
RewriteRule ^/mqipt  http://%{HTTP:Host}/mqipt  [P]

LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule rewrite_module modules/mod_rewrite.so

start Apache
```

3. 在 IPT1 上

   编辑 mqipt.conf 并添加路由定义:

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
HTTP=true
HTTPProxy=10.9.1.3
HTTPProxyPort=80
HTTPServer=9.100.6.7
HTTPServerPort=8080
```

   打开命令提示符:

```
c:
cd \mqipt\bin
mqipt ..
```

   下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....使用 HTTP
MQCPI024 ....和位于 10.9.1.3(80) 的 HTTP 代理
MQCPI066 ....和位于 9.100.6.7(8080) 的 HTTP 服务器
MQCPI078 路由 1415 用于连接请求准备就绪
```

4. 在 IPT2 上

   编辑 mqipt.conf 并添加路由定义:

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
```

   打开命令提示符:

```
c:
cd \mqipt\bin
mqipt ..
```

下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在读取来自 C:\mqipt\mqipt.conf 的配置信息
MQCPI011 路径 C:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....使用 MQ 协议
MQCPI078 路由 1415 用于连接请求准备就绪
```

5. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
```

6. 使用以下命令放入消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world <enter>
<enter>
```

7. 使用以下命令取出消息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

您将看见 “Hello world”。

安全性出口

对于本示例，我们作了以下假设:

- 已安装 Java 1.4 SDK
- Java bin 子目录已添加到 PATH 环境变量

这是一个简单的测试，它显示如何使用提供的称为 SampleSecurityExit 的样本安全性出口。此安全性出口编写为只允许使用以字符 “MQIPT.” 开头的通道名称的客户机连接。

使用名称为 “MQIPT.CONN.CHANNEL” 的建议 srvconn 通道名称（如这些样本中的大部分使用的），将允许客户机连接完成，且 WMQ 消息可放入队列。

要证明安全性出口如期待的那样工作，定义名称不是以字符 “MQIPT.” 开头的另一个 srvconn 通道（例如， “TEST.CONN.CHANNEL” ），然后再次尝试 amqsputc 命令，但将 MQSERVER 环境变量更改为使用新的通道名称。此时，连接将被拒绝，并给出错误 2059。

要显示 “TEST.CONN.CHANNEL” 是在不使用安全性出口的情况下工作，将 MQSERVER 环境变量设置为直接指向 WMQ 侦听器端口（例如，1414），因此 MQIPT 是不使用的。此时，amqsputc 命令将按期待的那样工作。

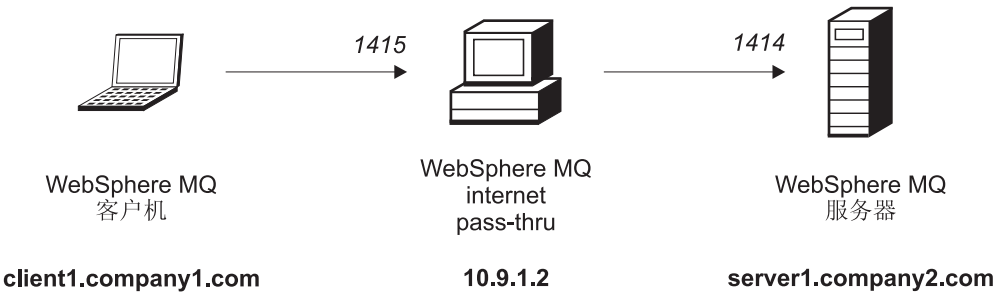

图 40. 安全性出口网络图

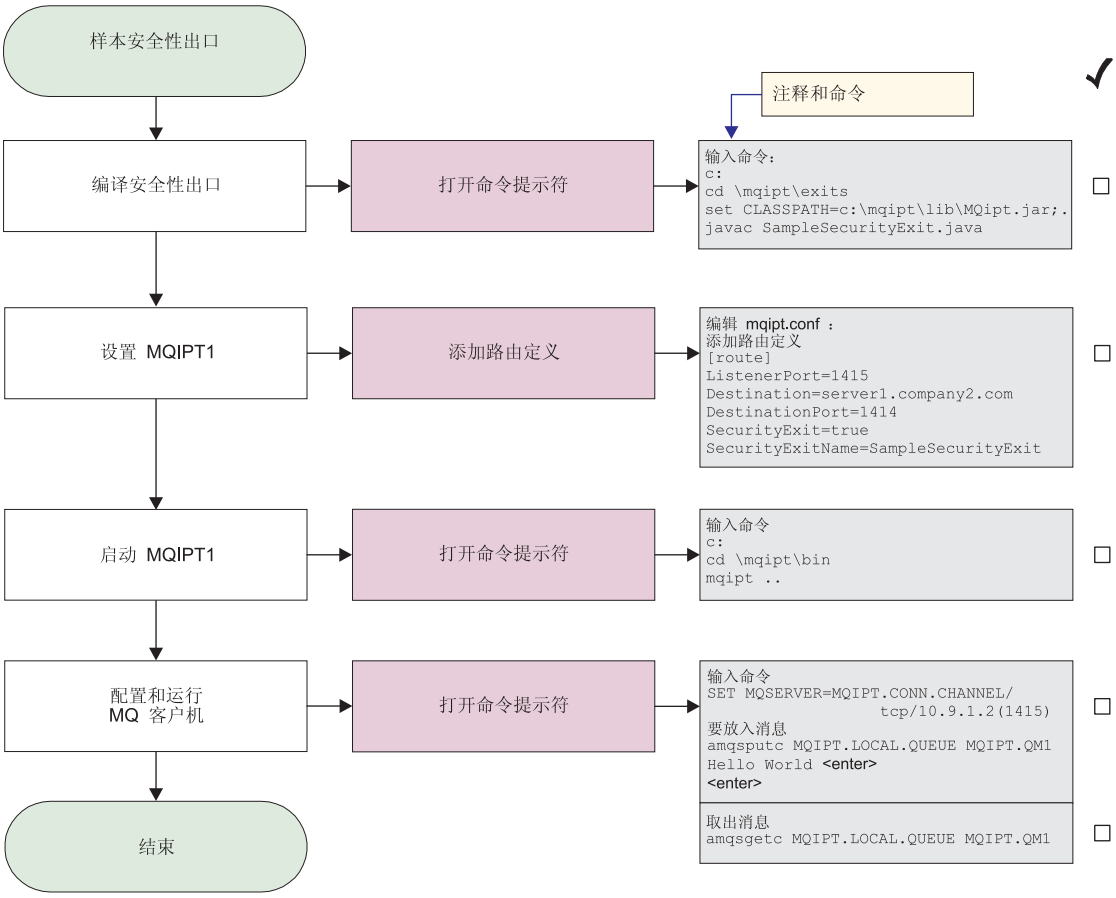

图 41. 安全性出口配置

1. 在 IPT1 上

   打开命令提示符:

   ```
   c:
   cd \mqipt\exits
   set CLASSPATH=c:\mqipt\lib\MQipt.jar;.
   javac SampleSecurityExit.java
   ```

   编辑 `mqipt.conf` 并添加路由定义:

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SecurityExit=true
SecurityExitName=SampleSecurityExit
```

打开命令提示符:

```
c:
cd \mqipt\bin
mqipt ..
```

下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在从 c:\mqipt\mqipt.conf 读取配置信息
MQCPI011 路径 c:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....使用 MQ 协议
MQCPI079 ....正在使用安全性出口 c:\mqipt\exits\SampleSecurityExit
MQCPI080 ......和超时 5 秒
MQCPI078 路由 1415 用于连接请求准备就绪
```

2. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
```

3. 使用以下命令放入消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world <enter>
<enter>
```

4. 使用以下命令取出消息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

您将看见 “Hello world”。

路由安全性出口

对于本示例，我们作了以下假设:

- 已安装 Java 1.4 SDK
- Java bin 子目录已添加到 PATH 环境变量
- 在三台单独的服务器上创建了三个等同的队列管理器

这是一个工作示例，它将客户机连接请求以循环方式动态路由到一组 WMQ 队列管理器服务器。组中的每台服务器上的队列管理器必须是其它每一台服务器的镜像映象。

将从配置文件读取服务器名列表。配置文件的名称和位置是由 SecurityExitName 和 SecurityExitPath 属性定义的。称为 SampleRoutingExit.conf 的样本配置文件包含下列条目:

```
server1.company.com:1414
server2.company.com:1415
server3.company.com:1416
```

您必须更改这些服务器名以匹配您的环境。

第一次发出 amqsputc 命令时，WMQ 消息将放入 server1 上的 QM 上的 MQIPT.LOCAL.QUEUE。第二次发出此命令时，消息将出现在 server2 上的 QM 上，依此类推。使用此设置时，amqsgetc 命令不可能检索刚放入队列的消息，因为 amqsgetc 命令使用的客户机连接请求将传递到列表中的下一个 QM。但发出后跟三条 amqsgetc 命令的三条 amqsputc 命令可确保以相同的顺序检索每一条消息。当然，通过使用另一台直接连接到 QM 的 WMQ 客户机（即，在此样本中不使用 MQIPT），您可以选择性地从任何队列管理器检索消息。

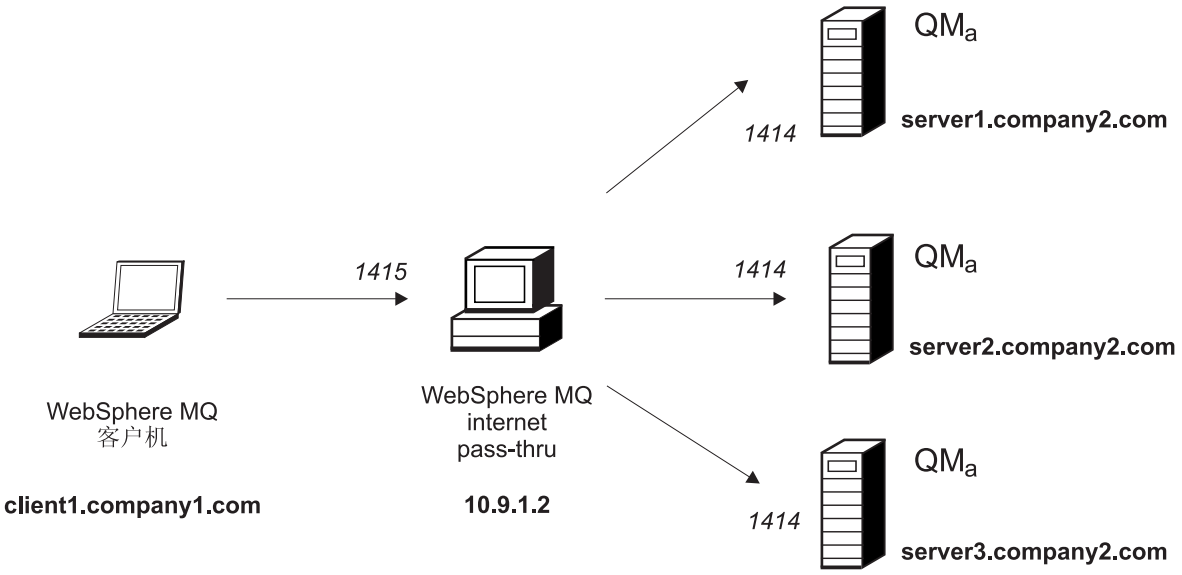

图 42. 路由安全性出口网络图

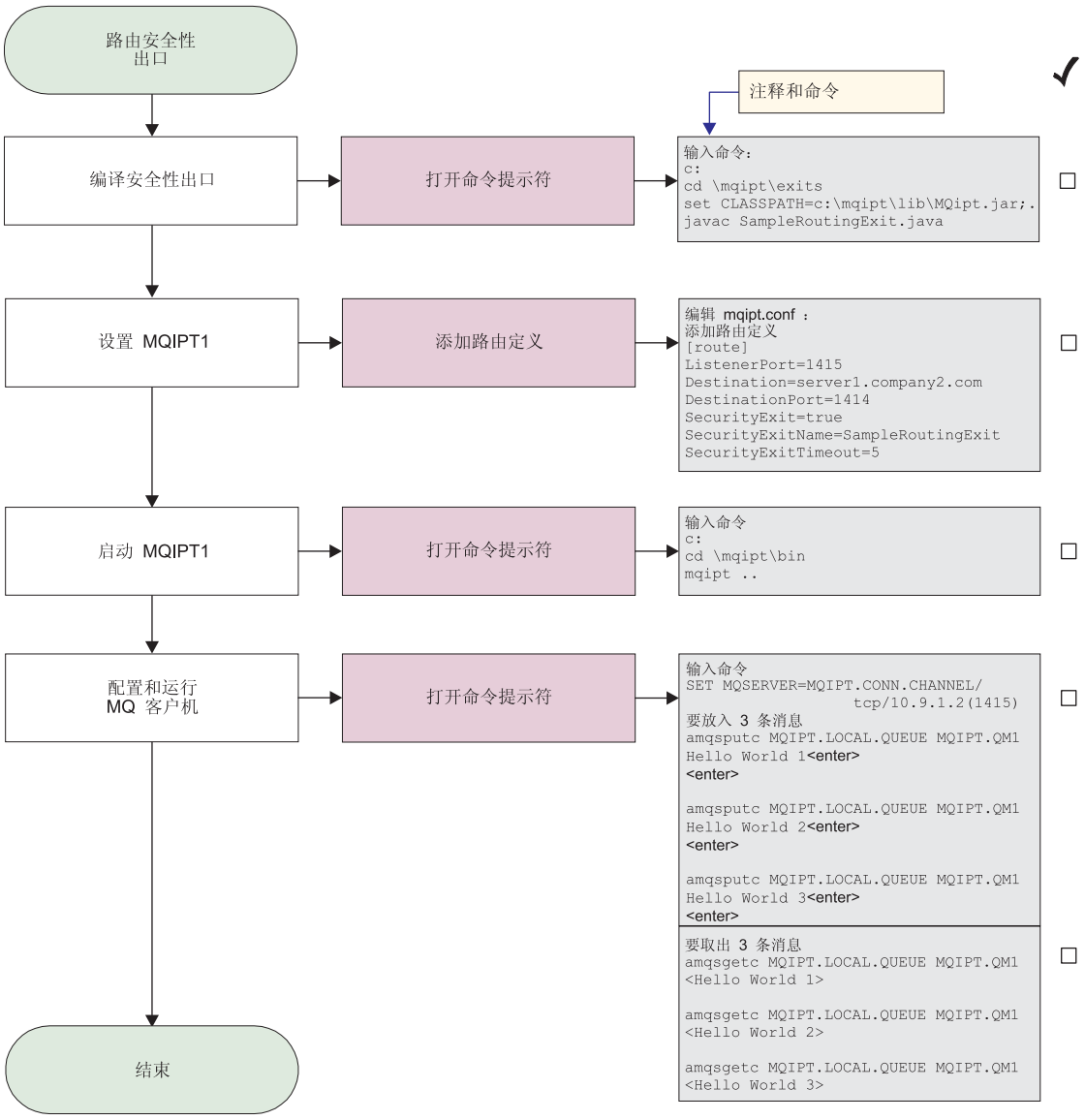

图 43. 路由安全性出口配置

1. 在 IPT1 上

   打开命令提示符:

   ```
   c:
   cd \mqipt\exits
   set CLASSPATH=c:\mqipt\lib\MQipt.jar;.
   javac SampleRoutingExit.java
   ```

   编辑 mqipt.conf 并添加路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   SecurityExit=true
   SecurityExitName=SampleRoutingExit
   ```

   打开命令提示符:

```
c:
cd \mqipt\bin
mqipt ..
```

下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在从 c:\mqipt\mqipt.conf 读取配置信息
MQCPI011 路径 c:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....使用 MQ 协议
MQCPI079 ....正在使用安全性出口 c:\mqipt\exits\SampleRoutingExit
MQCPI080 ......和超时 5 秒
MQCPI078 路由 1415 用于连接请求准备就绪
```

2. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
```

3. 使用下列命令放入三条消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world 1 <enter>
<enter>
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world 2 <enter>
<enter>
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world 3 <enter>
<enter>
```

4. 使用下列命令取出消息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

您将看到 “Hello world 1” 、 “Hello world 2” 和 “Hello world 3”。

动态一个路由出口

对于本示例，我们作了以下假设:

- 已安装 Java 1.4 SDK
- Java bin 子目录已添加到 PATH 环境变量
- 在三台单独的服务器上创建了三个不同的队列管理器

这是一个工作示例，它显示如何根据正在使用的通道的名称，将客户机连接请求动态路由到目标服务器。通道名称的第一部分是队列管理器的名称（因此，例如要连接到 QM1，则 svrconn 通道的名称将为 QM1.MQIPT.CONN.CHANNEL）。使用此通道命名约定，MQIPT 只需要一个路由以服务所有连接请求。

将从配置文件读取队列管理器和服务器名列表。配置文件的名称和位置是由 SecurityExitName 和 SecurityExitPath 属性定义的。称为 SampleOneRouteExit.conf 的样本配置文件包含下列条目:

```
QM1   server1.company.com:1414
QM2   server2.company.com:1415
QM3   server3.company.com:1416
```

您必须更改这些服务器名以匹配您的环境。

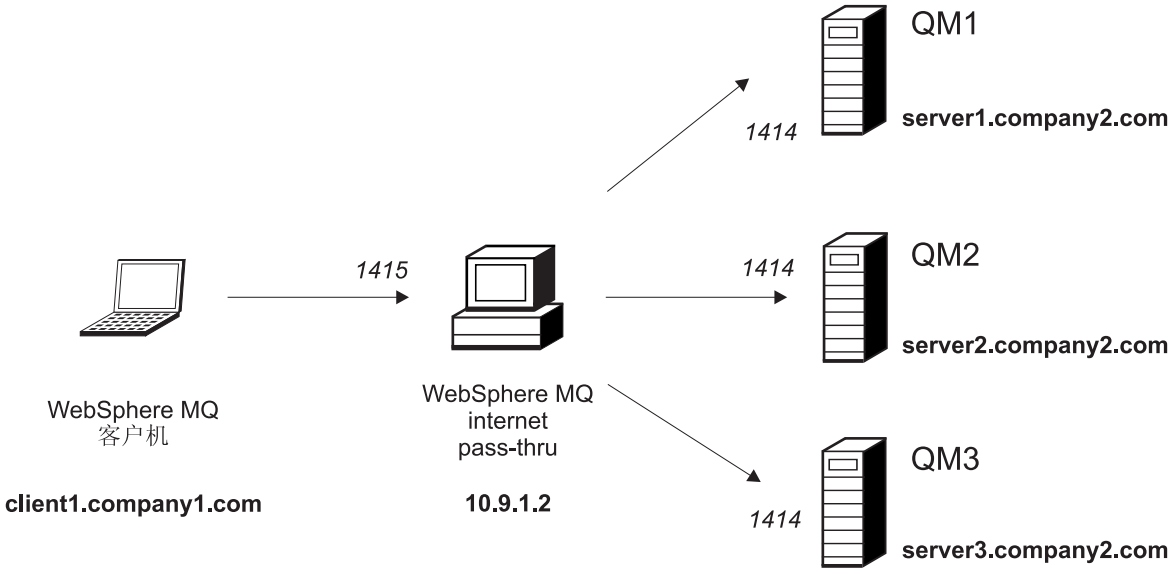

图 44. 动态一个路由出口网络图

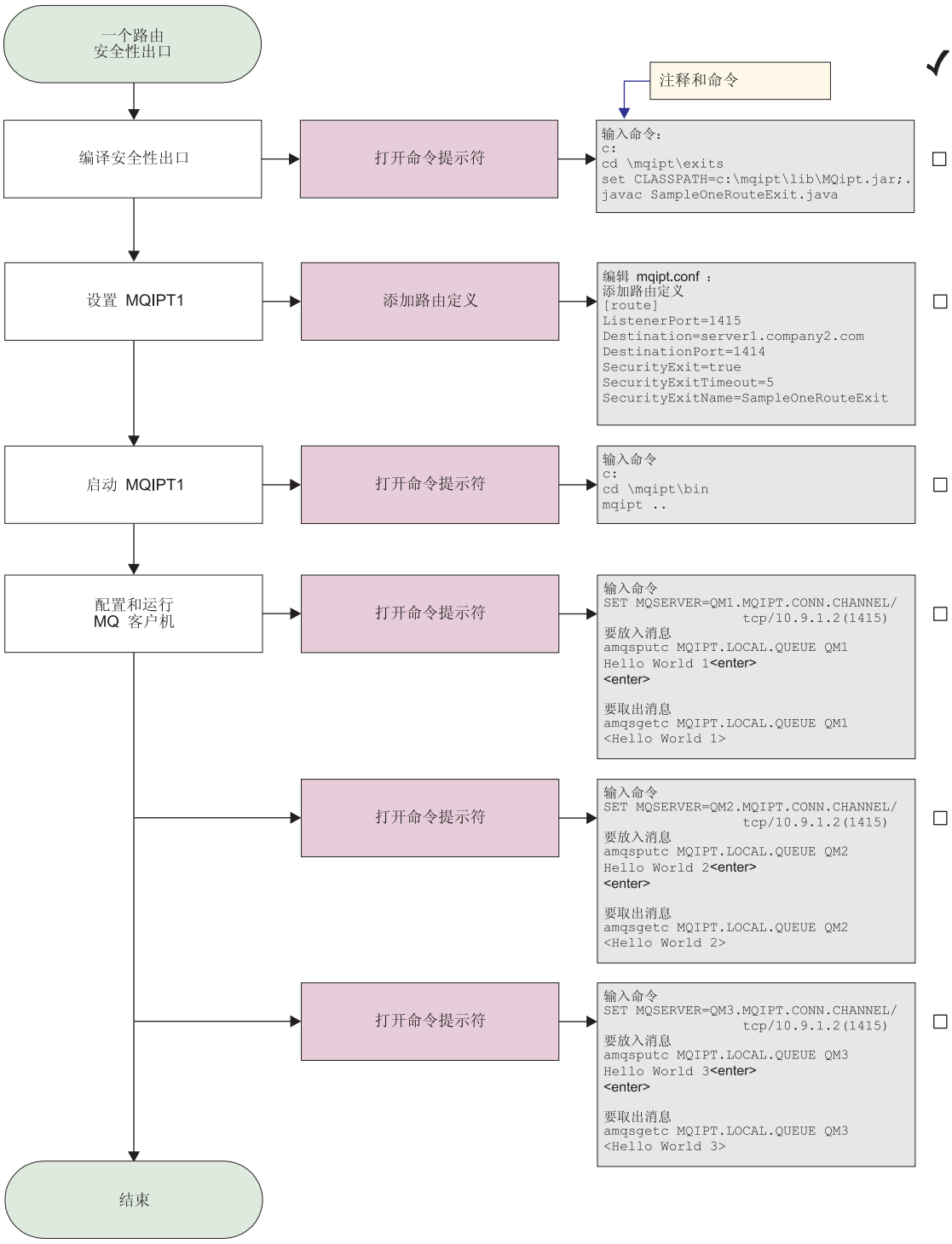

图 45. 动态一个路由出口配置

1. 在 IPT1 上

   打开命令提示符:

   ```
   c:
   cd \mqipt\exits
   set CLASSPATH=c:\mqipt\lib\MQipt.jar;.
   javac SampleOneRouteExit.java
   ```

编辑 mqipt.conf 并添加路由定义:

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SecurityExit=true
SecurityExitName=SampleOneRouteExit
```

打开命令提示符:

```
c:
cd \mqipt\bin
mqipt ..
```

下列消息表明成功完成:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru V1.3.0 正在启动
MQCPI004 正在从 c:\mqipt\mqipt.conf 读取配置信息
MQCPI011 路径 c:\mqipt\logs 将用于存储日志文件
MQCPI006 路由 1415 已启动，并将转发消息至:
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....使用 MQ 协议
MQCPI079 ....正在使用安全性出口 c:\mqipt\exits\SampleOneRouteExit
MQCPI080 ......和超时 5 秒
MQCPI078 路由 1415 用于连接请求准备就绪
```

2. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

   ```
   SET MQSERVER=QM1.MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
   ```

3. 使用以下命令放入消息:

   ```
   amqsputc MQIPT.LOCAL.QUEUE QM1
   Hello world 1 <enter>
   <enter>
   ```

4. 使用以下命令取出消息:

   ```
   amqsgetc MQIPT.LOCAL.QUEUE QM1
   ```

   您将看到 “Hello world 1”。

5. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

   ```
   SET MQSERVER=QM2.MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
   ```

6. 使用以下命令放入消息:

   ```
   amqsputc MQIPT.LOCAL.QUEUE QM2
   Hello world 2 <enter>
   <enter>
   ```

7. 使用以下命令取出消息:

   ```
   amqsgetc MQIPT.LOCAL.QUEUE QM2
   ```

   您将看见 “Hello world 2”。

8. 在 WebSphere MQ 客户机的命令提示符下，输入下列命令:

   ```
   SET MQSERVER=QM3.MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
   ```

9. 使用以下命令放入消息:

   ```
   amqsputc MQIPT.LOCAL.QUEUE QM3
   Hello world 3 <enter>
   <enter>
   ```

10. 使用以下命令取出消息:

    ```
    amqsgetc MQIPT.LOCAL.QUEUE QM3
    ```

您将看到 “Hello world 3”。

第 21 章 照管 internet pass-thru

本章通过下列标题描述如何使 internet pass-thru 保持运行:

- 『维护』
- 『问题确定』
- 第 148 页的『性能调整』

维护

您应将定期备份下列文件作为正常备份过程中的一部分:

- 配置文件，mqipt.conf
- mqipt.conf 中用下列属性定义的 SSL 密钥环文件:
  - SSLClientKeyRing
  - SSLClientCAKeyRing
  - SSLServerKeyRing
  - SSLServerCAKeyRing
- mqipt.conf 中用下列属性定义的 SSL 密钥环密码文件:
  - SSLClientKeyRingPW
  - SSLClientCAKeyRingPW
  - SSLServerKeyRingPW
  - SSLServerCAKeyRingPW
- 管理客户机配置文件，client.conf，它包含管理客户机已知的所有 MQIPT 的相关连接信息。

问题确定

如果您遇到某个问题，则首先要检查一些常见错误:

- 刚安装了 MQIPT 系统，但未重新引导该系统。
- 在直接连接到队列管理器的路由上已将 HTTP 设置为 true。
- 在直接连接到队列管理器的路由上已将 SSLClient 设置为 true。
- 未正确设置 CLASSPATH。
- 未正确设置 PATH。
- 为密钥环文件存储的密码是区分大小写的。

下一步则是按照第 146 页的图 46 中显示的流程图进行操作。图中的号码请参阅以下注释。

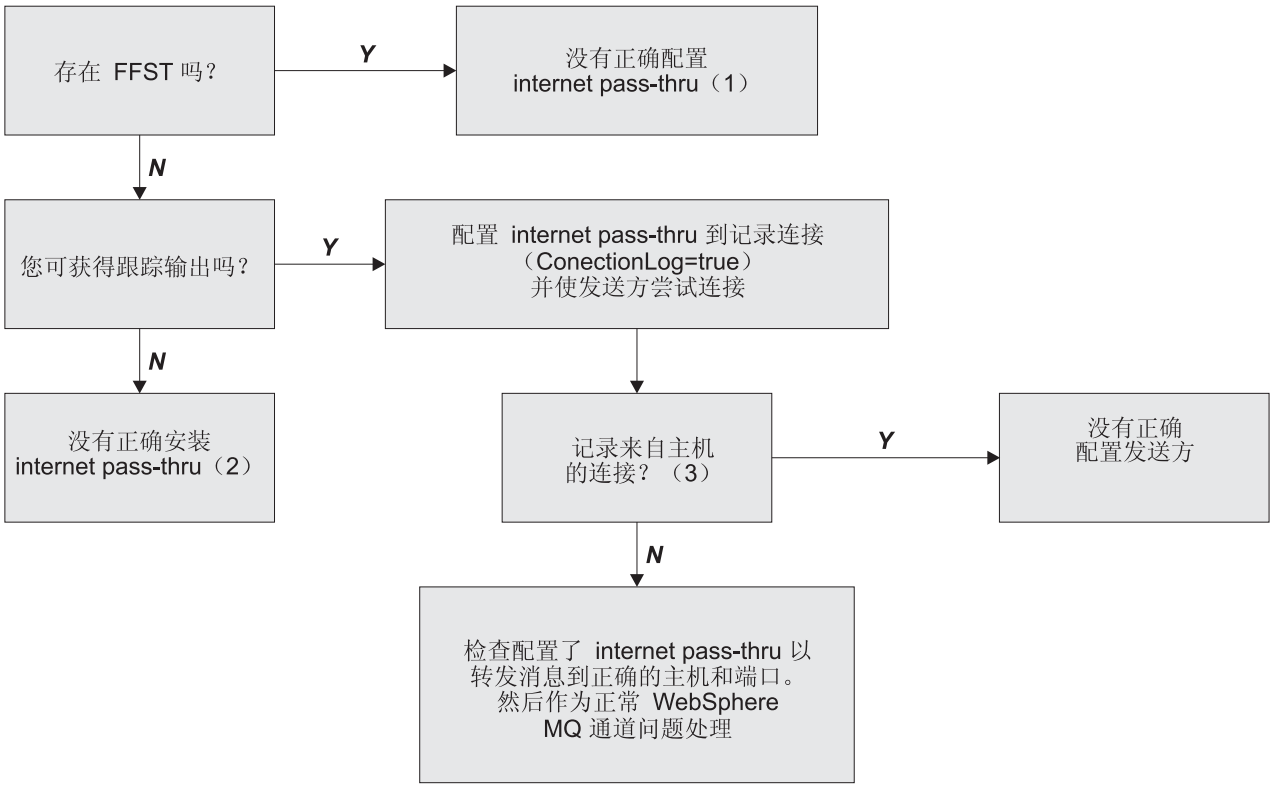

图 46. 问题确定流程图

注:

1. 如果您发现任何 FFST 报告（在 errors 子目录中），您就知道 MQIPT 已正确安装。可能存在配置问题。

   每个 FFST 报告一个引起 MQIPT 或路由终止其启动进程的问题。解决引起每个 FFST 的问题。然后删除旧的 FFST 并重新启动或刷新 MQIPT。
2. 如果没有正确安装 MQIPT，则检查所有文件是否已放置在正确的位置，且是否已经更新 CLASSPATH。要检查它是否正确，请尝试手工启动 MQIPT。
3. 手工启动 MQIPT。

   打开命令提示符。转至 `bin` 子目录并输入:

   ```
   mqipt xxx
   ```

   其中 *xxx* 是 MQIPT 主目录；在此情况下，它是 “`..`”。

   这将启动 MQIPT 并在主目录中查找配置。在 `errors` 子目录中查找任何错误消息和 FFST。

   查看来自 MQIPT 的文本输出，以找出错误消息并更正这些错误。检查 FFST 并改正所有错误。如果配置文件中 global 节中存在问题，则 MQIPT 将不会启动。如果配置文件的 route 节中存在问题，则此路由将不会启动。

自动启动 internet pass-thru

如果您将 MQIPT 安装为 Windows NT 服务，并且已将其启动状态更改为“自动”，则它将在系统启动时启动。在尝试将 MQIPT 安装为 Windows NT 服务之前，总是先手工启动一次 MQIPT，以确认安装正确。请参阅第 47 页的『使用 Windows 服务控制程序』获取更多详细信息。

如果您接收到错误消息“无法定位 DLL...”，则表明您使用了错误的 mqiptService 程序，或者您没有正确配置系统 PATH 环境变量。PATH 必须包含 JNI 运行时库的位置。您可在 JDK 的 client 子目录中找到此文件（jvm.dll）。

检查端到端连接性

如果 MQIPT 已正确安装，则下一步将是检查是否正确设置路由。

在配置文件 mqipt.conf 中，将 ConnectionLog 属性设置为 true。启动或刷新 MQIPT 并尝试连接。此连接日志将创建在主目录下的 logs 目录中。如果没有创建该日志，您就可以知道没有正确安装 MQIPT。如果没有记录连接尝试，则表明未正确设置发送方。如果记录了尝试，则检查 MQIPT 是否将消息转发到正确的地址。

跟踪错误

MQIPT 提供了一个详细的执行跟踪工具，该工具由跟踪属性控制。可以分别对每个路由进行跟踪。跟踪文件将写入 *xxx*\errors 目录（其中 *xxx* 是包含 mqipt.conf 的目录。产生的每个跟踪文件的名称都使用以下格式:

iptroute*nnnnn*.trc

其中 *nnnnn* 是路由所侦听的端口号。不直接与任何特殊路由（例如，线程句柄命令输入）关联的跟踪输出将写入一个名为 iptmain.trc 的独立文件。

意外的致命错误则作为 FFST 记录写入 *xxx*\errors 目录（其中 *xxx* 是包含 mqipt.conf 的目录）中保存的出错日志文件。FFST 文件的格式如下:

ipt*xxx*.FFST

其中 *xxx* 是 FFST 生成的序列（1 代表最旧）。在长期运行的系统中，可能会达到系统可生成的最大数。在此情况下，所生成的任何 FFST 将写入文件 mqipt0.FFST。如果创建了文件 mqipt0.FFST，则您应在第一时机停止并重新启动 MQIPT，并且删除旧的文件。

报告问题

如果您确实必须向 IBM 服务中心报告问题，那么如果您能够提供下列信息，将会有助于更快地解决此问题:

- 提供您所用机器的简单网络图，包括 IP 地址。
- 如果使用多个 MQIPT，则使每台 MQIPT 机器上的系统时钟同步 – 这将有助于匹配每个 MQIPT 中的跟踪条目
- 擦除旧的跟踪文件
- 运行客户机以制造此问题 – 这样跟踪文件就会只包含此问题的一个实例。
- 发送所有 MQIPT .trc 和 .log 文件的副本

性能调整

这里有一些关于调整系统的指示。

线程池管理

线程池和空闲超时规范组合起来使用可以调整每个路由的相对性能。

连接线程

将为每个 MQIPT 路由指定一个工作的并发运行线程（这些线程处理进入通信请求）池。初始化时，将创建一个线程池（其大小在路由的 MinConnectionThreads 属性中指定），并指定一个线程来处理第一个进入请求。当此请求到达时，该线程立即开始对此请求进行处理，且下一个线程被指定为“就绪”以准备处理下一个进入请求。当所有线程都被指定为工作时，将创建新的线程，将其添加到工作池中，并指定其工作。这样，池将不断地增长，直至达到 MaxConnectionThreads。当工作线程数达到 MaxConnectionThreads，下一个进入请求将等待，直至线程被释放回工作池。这是路由的最大工作量，超出此工作量后便无法接受其它请求。当对话结束或指定的空闲超时周期过后，就会将线程释放回池中。

空闲超时

缺省情况下，不会因为工作线程不活动而终止该线程。当线程已指定给某一对话时，在正常关闭此对话、取消激活路由或关闭 MQIPT 前，该线程将保持指定给此对话。您也可以指定空闲超时间隔，这样就能够终止在指定时间周期（以分钟计）内不活动的任何任何线程。监控线程保持对线程空闲时间的定期检查，并终止那些已经超出该阈值的线程。然后通过将这些线程放回工作池以回收使用它们。

第 22 章 消息

当从命令行运行时，MQIPT 会在控制台上显示少量信息、警告和错误消息。

注意:

- MQCAxxxx 消息是管理客户机消息。
- MQCPxxxx 消息是 MQIPT 消息。
- MQCxIxxx 消息是信息性消息。
- MQCxWxxx 消息是警告消息。
- MQCxExxx 消息是错误消息。

MQCAE001 未知主机: {0}

解释: 无法找到 MQIPT 主机。

用户回答: 检查您是否正确指定了 MOIPT 所在的主机名。

MQCAE002 下列错误由系统报告: {0}

解释: 发生一个错误。执行系统命令后报告了一个错误。

MQCAE005 没有定义有效的目标地址

解释: 添加路由时，目标字段被保留为空白。

用户回答: 输入一个有效的目标地址。

MQCAE006 没有定义有效的目标端口

解释: 添加路由时，目标端口地址字段被保留为空白。

用户回答: 输入一个有效的目标端口地址。

MQCAE007 没有定义有效的侦听器端口

解释: 添加路由时，侦听器端口地址字段被保留为空白。

用户回答: 输入一个有效的侦听器端口地址，范围在 1 到 65535 之间。

MQCAE008 没有定义有效的网络地址

解释: 添加 MQIPT 时，网络地址字段被保留为空白。

用户回答: 输入一个有效的网络地址。

MQCAE009 没有定义有效的命令端口

解释: 添加 MQIPT 时，使用了无效的命令端口。

用户回答: 输入一个有效的命令端口地址，范围在 1 到 65535 之间。

MQCAE010 无法显示联机帮助

解释: 联机帮助文件可用，但是无法显示。

用户回答: 确保您安装了 web 浏览器，且它在系统 PATH 环境变量中是可用的。

MQCAE011 无法分析参数

解释: 发生了一个内部错误，该错误导致试图更新表中不存在的参数。

用户回答: 如果此情况仍然存在，请联系 IBM 技术支持。

MQCAE012 无法找到联机帮助文件 {0}

解释: 无法找到文件 “passtfrm.htm”。

用户回答: 确保此文件在 doc 语言子目录中是可访问的。

MQCAE013 尝试显示联机帮助时中断

解释: 显示联机帮助时发生系统错误。

用户回答: 再试一次。如果此情况仍然存在，请联系 IBM 技术支持。

MQCAE015 未识别您刚刚输入的密码

解释: MQIPT 希望得到有效的密码，最后一条命令中使用的密码不正确。它必须与配置文件中定义的密码是匹配的。

用户回答: 使用 **MQIPT-> 连接**面板更改密码，然后重试最后一条命令。

MQCAE016 节点不匹配

解释: 树中选择的节点和内存中保存的数据之间存在内部不一致。

用户回答: 关闭管理客户机，然后重试命令。如果此情况仍然存在，请联系 IBM 技术支持。

MQCAE017 无法创建消息 {0} 的 NLS 文本

解释: 未找到定义的消息号码的 NLS 文本。

用户回答: “guiadmin.properties” 文件可能损坏，无法找到指定的消息号码。检查下列事项:

- 查看 Readme 文件以查找可能的新消息
- “guiadmin.jar” 文件在系统 CLASSPATH 中
- “guiadmin.properties” 文件在 “guiadmin.jar” 文件中
- 消息号码在 “guiadmin.properties” 文件中

MQCAE018 无法创建消息 MQCAE017 的 NLS 文本

解释: 无法在系统属性列表中找到消息号码 {0}。

用户回答: “guiadmin.properties” 可能损坏，检查下列事项:

- “guiadmin.jar” 文件在系统 CLASSPATH 中
- “guiadmin.properties” 文件在 “guiadmin.jar” 文件中
- 消息号码在 “guiadmin.properties” 文件中

MQCAE019 没有重复输入建议的新密码

解释: 更改密码时，没有输入两次以用于验证。

用户回答: 在相应的字段中再次输入新的密码。

MQCAE020 更改 MQIPT 访问参数失败

解释: 试图更改 MQIPT 访问参数时，检测到一个内部错误。

用户回答: 关闭管理客户机，然后重试命令。如果此情况仍然存在，请联系 IBM 技术支持。

MQCAE021 标识 MQIPT 时发生内部故障

解释: 试图在 MQIPT 上保存配置文件时，检测到一个内部错误。

用户回答: 关闭管理客户机，然后重试命令。如果此情况仍然存在，请联系 IBM 技术支持。

MQCAE022 保存 MQIPT 配置时发生内部故障

解释: 试图在 MQIPT 上保存配置文件时，检测到一个内部错误。

用户回答: 关闭管理客户机，然后重试命令。如果此情况仍然存在，请联系 IBM 技术支持。

MQCAE023 MQIPT {0} 未识别您的密码。

解释: MQIPT 希望得到有效的密码，最后一条命令中使用的密码不正确。它必须与配置文件中定义的密码是匹配的

用户回答: 使用菜单 **MQIPT-> 连接**面板更改密码，然后重试最后一条命令。

MQCAE024 MQIPT {0} 不识别该命令。

解释: 管理客户机发送一条命令到 MQIPT，但是该命令无法被识别。

用户回答: 确保管理客户机使用的代码的版本与 MQIPT 的是相同的。

MQCAE025 MQIPT {0} 发送配置文件失败。

解释: MQIPT 试图发送配置文件，但失败。

用户回答: 关闭管理客户机，然后重试命令。如果这不起作用，停止并重新启动 MQIPT。

MQCAE026 在 MQIPT {0} 上远程关机是禁用的。

解释: 试图远程关闭 MQIPT 失败，因为配置文件中没有启用远程关闭。

用户回答: 要启用 MQIPT 远程关闭，编辑配置文件并设置 RemoteShutDown 属性为 true。

MQCAE027 不支持感观 {0}。

解释: 您使用的平台的建议感观不可用。

用户回答: 处理继续使用系统的缺省感观。

MQCAE028 找不到感观类 {0}。

解释: 您使用的平台的建议感观不可用。

用户回答: 处理继续使用系统的缺省感观。

MQCAE029 最小连接线程数必须是非负数并且不比最大连接线程数大

解释: 最小连接线程数的值必须小于或等于最大连接线程数值。

用户回答: 相应地更改值。

MQCAE030 最大连接线程数必须大于零并且至少和最小连接线程数一样大

解释: 最大连接线程数的值必须大于或等于最小连接线程数值。

用户回答: 相应地更改值。

MQCAE031 端口号必须在范围 0 到 65535 之间

解释: 您试图设置不符合规范的值。

用户回答: 相应地更改值。

MQCAE032 跟踪必须在范围 0 到 5 之间

解释: 您试图设置不符合规范的值。

用户回答: 相应地更改值。

MQCAE033 最大日志文件大小必须在范围 5 到 50 之间

解释: 您试图设置不符合规范的值。

用户回答: 相应地更改值。

MQCAE049 在任何 MQIPT 上没有路由被选中

解释: 试图不先选择要删除的路由，就删除路由。

用户回答: 选择一个路由，然后重试命令。

MQCAE050 无法连接到 MQIPT {0}

解释: 管理客户机无法连接到指定的 MQIPT。

用户回答: 这可能是以下任一原因引起的:

- MQIPT 没有运行。
- MQIPT 没有在它的命令端口上侦听。
- 只有一个管理客户机使用 MQIPT CommandPort。
- 请求超时。

MQCAE051 无法读来自 MQIPT {0} 的应答

解释: 从 MQIPT 接收到一个应答，它不遵从期待的协议。

用户回答: 确保管理客户机使用的代码的版本与 MQIPT 的是相同的。

MQCAE052 配置未保存

解释: 从 MQIPT 接收到有效的应答，但接下来保存到配置文件失败。

用户回答: 检查 MQIPT 对配置文件是否可以进行写访问。

MQCAE053 MQIPT 未确认配置的保存

解释: 配置文件已发送到 MQIPT，但 MQIPT 未能确认。

用户回答: 这可能是以下任一原因引起的:

- MQIPT 没有运行。
- MQIPT 没有在它的命令端口上侦听。
- 只有一个管理客户机使用 MQIPT CommandPort。
- 请求超时。

MQCAE054 MQIPT 数据没有被刷新

解释: 已经建立与 MQIPT 的联系，但是管理客户机无法读取配置文件。

用户回答: 这可能是以下任一原因引起的:

1. MQIPT 发生故障
2. 请求超时。

MQCAE055 在 MQIPT 上没有 MQIPT 或路由被选中

解释: 因为没有选择 MQIPT 或路由，您选择的菜单选项无法执行。

用户回答: 选择相应的 MQIPT 或者路由，然后再试一次。

MQCAE056 重复侦听器端口被拒绝

解释: 指定的侦听器端口被拒绝，因为它已经被另一路由使用。

用户回答: 选择不同的侦听器端口，然后再试一次。

MQCAI002 MQIPT 已从显示中除去

解释: 您在树中选择的节点的 MQIPT 已被从客户机的内存中除去。

MQCAI003 新路由已添加到显示中

解释: 您刚才指定的新的路由已添加到当前 MQIPT。

MQCAI004 路由已从显示中除去

解释: 您在树中选择的路由已被从客户机的内存中除去。

MQCAI005 所选 MQIPT 正在显示

解释: 您在树中选择的 MQIPT 的全局参数在表中显示。

MQCAI006　所选路由正在显示

解释:　您在树中选择的路由的参数在表中显示。

MQCAI007　客户机配置已保存

解释:　已保存树中所有 MQIPT 的访问参数。

MQCAI008　显示联机帮助成功

解释:　根据要求显示了联机帮助。

MQCAI009　表已被更新

解释:　您刚才在表中输入的值已被用于更新内存中的模型。

MQCAI010　没有 MQIPT 或路由被选中

解释:　因为用于操作的信息不足，因此不采取任何操作。

MQCAI011　用户操作已被取消

解释:　您取消了操作（包括前面启动的弹出窗口）。

MQCAI014　配置已保存在 MQIPT 上

解释:　新的配置文件被保存在当前在树中选择的 MQIPT 上，它已用于重新启动 MQIPT。

MQCAI015　联机帮助已经终止

解释:　根据要求显示了联机帮助，接着又终止了。

MQCAI017　选择“文件”/“添加 MQIPT”以将一个 MQIPT 添加到树中

解释:　当树中没有 MQIPT 时，会出现此消息；它告诉你如何添加一个 MQIPT。

MQCAI018　新 MQIPT 已添加到显示中

解释:　按指示，向树添加了一个新的 MQIPT。

MQCAI019　MQIPT 访问参数已被更改

解释:　当前在树中选择的 MQIPT 的访问参数已更改。

MQCAI021　在树上选择一个 MQIPT 或路由以显示它的内容

解释:　当表中没有显示的信息时会出现此消息，它告诉您如何看到一些信息。

MQCAI022　命令端口已更改

解释:　指示要求更改的 MQIPT 命令端口现在已经更改。

MQCAI023　密码已更改

解释:　接下来所有与您刚才更改的 MQIPT 的通信将使用新的密码。

MQCAI025　MQIPT {0} 已被刷新。

解释:　您在 MQIPT 中保存的信息被从它的配置文件读取的信息更新。

MQCAI026　MQIPT {0} 已接收到关机请求。

解释:　MQIPT 确认了关闭请求的收到信号，现在将关闭。

MQCAI027　客户机配置已被刷新

解释:　管理客户机中显示的信息被本地 “client.conf” 文件刷新。

MQCAI028　MQIPT {0} 是活动的

解释:　MQIPT 成功响应 ping 请求。

MQCAI029　MQIPT {0} 不是活动的

解释:　MQIPT 没有在指定时间内响应 ping 请求。

用户回答:　这可能是以下任一原因引起的:

- MQIPT 没有运行。
- MQIPT 没有在它的命令端口上侦听。
- 请求超时。超时时间可通过更改 MQIPT 的连接信息中的超时属性来增加。

MQCAI030　路由 {0} 是活动的

解释:　MQIPT 成功响应 ping 请求。

MQCAI031　路由 {0} 不是活动的

解释:　MQIPT 路由没有在指定时间内响应 ping 请求。

用户回答:　这可能是以下任一原因引起的:

- MQIPT 没有运行。
- MQIPT 没有在它的命令端口上侦听。
- 请求超时。超时时间可通过更改 MQIPT 的连接信息中的超时属性来增加。

MQCAI100　此脚本用于启动 {0} 的管理客户机。指定 SOCKS 代理将允许此管理员客户机通过防火墙和 MQIPT 对话。

解释:　mqiptGui 脚本的联机帮助消息。

MQCAI101　命令格式是:

解释:　mqiptGui 脚本的联机帮助消息。

MQCAI102　mqiptGui {socks_host{socks_port}}

解释:　mqiptGui 脚本的联机帮助消息。

MQCAI103　SOCKS 代理的 socks_host-host 名称（可选）

解释:　mqiptGui 脚本的联机帮助消息。

MQCAI104　socks_port-SOCKS 代理端口地址（可选的 - 缺省值 1080）

解释:　mqiptGui 脚本的联机帮助消息。

MQCPE000　处理消息 {0} 时，无法定位消息数据

解释:　无法在系统属性列表中找到消息号码 {0}。

用户回答:　“mqipt.properties” 文件已经损坏，无法找到指定的消息号码。检查下列事项:

- “MQipt.jar” 文件在系统 CLASSPATH 中
- “mqipt.properties” 文件在 “MQipt.jar” 文件中
- 消息号码在 “mqipt.properties” 文件中

MQCPE001　目录不存在或不是目录

解释:　初始化期间没有找到必需的目录。本消息引用 MQIPT 配置文件 mqipt.conf 中或者缺省目录上的 MQIPT 命令行启动选项中指定的目录。

用户回答:　指定正确的目录，然后重试命令。

MQCPE004　在端口 {0} 上路由启动失败

解释:　不可能使用指定的 ListenerPort 号启动路由。

用户回答:　路由启动期间发生 I/O 错误。检查其它相邻的错误消息和日志记录以提供问题的进一步解释。

MQCPE005　无法找到配置文件 {0}

解释:　指定的目录中没有找到 MQIPT 配置文件 “mqipt.conf”

用户回答:　指定正确的目录，然后重试命令。

MQCPE006　路由数已超出 {0}。MQIPT 将启动，但此配置是不支持的。

解释:　您的配置已经超过了一个 MQIPT 实例最大支持路由数目。操作不会停机，但可能导致系统变得不稳定或者过载。不支持超过声明的路由最大数目的配置。

用户回答:　考虑启动其它的 MQIPT 实例，以减少每个实例的路由数目。

MQCPE007　在侦听器端口 {0} 上路由没有重新启动

解释:　REFRESH 操作时，指定 ListenerPort 上操作的路由没有使用新的配置重新启动。

用户回答:　检查其它相邻的错误消息以获取问题的进一步解释。

MQCPE008　为侦听器端口 {0} 定义了重复路由

解释:　多个路由定义了相同的 ListenerPort 值。

用户回答:　从配置文件除去重复的路由，然后重试命令。

MQCPE009　日志目录 {0} 无效。

解释:　文本中显示的日志路径不存在或者现在不可访问。

用户回答:　检查目录是否存在，且可被 MQIPT 访问。

MQCPE010　侦听器或命令端口号 {0} 无效

解释:　为命令端口或侦听器端口参数提供的端口号无效。

用户回答:　指定一个未使用的有效端口号。要获取关于您的网络中使用的端口号的指导，请向网络管理员咨询。

MQCPE011　跟踪级别 {0} 不在有效范围 0 - 5 之间

解释:　请求了指定的跟踪选项，但它不在有效范围 0-5 中。

用户回答:　指定一个 0 - 5 之间的跟踪值。

MQCPE012　属性 {1} 的值 {0} 无效

解释:　指定了无效的属性值。

用户回答:　请参阅本用户指南以获取每个控制参数的有效值的完整详细信息。

MQCPE013　在路由 {0} 中未找到 ListenerPort 属性

解释:　MQIPT 检测到在配置文件中路由没有包含 ListenerPort 属性。ListenerPort 属性是每个路由的主要和唯一标识符，因此是必需的。

用户回答:　为给定的路由指定有效的 ListenerPort。

MQCPE014 ListenerPort 属性值 {0} 无效

解释: 为路由的 ListenerPort 属性指定了无效的端口地址。

用户回答: 端口地址的范围必须在 0 - 65535 之间。检查配置文件中的每个 ListenerPort。

MQCPE015 没有查找到消息号 {0} 的文本

解释: 遇到一个内部错误，但它没有可用的描述。

用户回答: “mqipt.properties” 文件已经损坏，无法找到指定的消息号码。检查下列事项:

- 查看 Readme 文件以查找可能的新消息
- “MQipt.jar” 文件在系统 CLASSPATH 中
- “mqipt.properties” 文件在 “MQipt.jar” 文件中
- 消息号码在 “mqipt.properties” 文件中

MQCPE016 最大连接线程数是 {0}，但小于最小连接线程数，即 {1}

解释: 您的配置指定的最小连接线程数目的值超过了最大连接线程数目。

用户回答: 这可能是单个路由中的错误，全局属性和路由属性之间的冲突，或者是覆盖系统缺省值的路由属性。请参阅本用户指南前面的一些章节以获取有效值和适用缺省值的完整详细信息。

MQCPE017 抛出异常 {0}，导致 MQIPT 关机

解释: MQIPT 异常终止且已经关闭。发生这种情况可能是因为环境条件或约束（例如内存溢出）。

用户回答: 如果此情况仍然存在，请联系 IBM 技术支持。

MQCPE018 ListenerPort 属性为空 - 路由将不启动

解释: 路由中省略了 ListenerPort 号。

用户回答: 编辑配置文件并添加一个有效的 ListenerPort。

MQCPE019 在下列内容前没有查找到节 {0}: {1}

解释: 配置文件中发生顺序错误。

用户回答: 编辑配置文件并确保所有 [route] 条目在 [global] 条目之后。

MQCPE020 MaxConnectionThreads 的新值是 {0}。它必须比当前值 {1} 大

解释: 路由启动之后，MaxConnectionThread 属性只可以增加。

用户回答: 编辑配置文件并更改 MaxConnectionThread 属性。

MQCPE021 没有为路由 {0} 提供 “目标” 属性

解释: “目标” 属性是路由中的 “必需字段”，但在指定的路由中省略了该属性。

用户回答: 编辑配置文件并为给定的路由添加 “目标” 属性。

MQCPE022 CommandPort 值 {0} 不在有效范围 1 - 65535 之间。

解释: CommandPort 属性超出 1-65535 的范围。

用户回答: 编辑配置文件并将 CommandPort 属性更改为有效的端口地址。

MQCPE023 来自管理客户机 {0} 的关机请求被忽略，因为这是禁用的。

解释: 试图远程关闭 MQIPT 失败，因为配置文件中没有启用远程关闭。

用户回答: 要启用 MQIPT 远程关闭，编辑配置文件并设置 RemoteShutDown 属性为 true。

MQCPE024 MQIPT 控制器接收到了未识别的命令。

解释: MQIPT 从它的命令端口接收到未识别的命令。

用户回答: 检查 “mqipt.log” 文件以获取命令的身份。

MQCPE025 连接主机 {0}，端口 {1} 上的服务器失败。

解释: 行方式（非 GUI）管理客户机与 MQIPT 通信失败。

用户回答: 确保配置文件中 CommandPort 属性指定为 {1}，且 MQIPT 在 {0} 上运行。

MQCPE026 没有接收到来自主机 {0}，端口 {1} 上服务器的应答。

解释: 行方式（非 GUI）管理客户机已与 MQIPT 连接，但没有接收到应答。

用户回答: 这表明请求超时或者 MQIPT 有问题。

MQCPE027 来自 MQIPT 的应答未识别。

解释: 行方式（非 GUI）管理客户机从 MQIPT 接收未识别的应答。

用户回答: 检查 mqiptAdmin 脚本使用的 “MQipt.jar” 文件版本与 MQIPT 使用的 “MQipt.jar” 文件版本相同。

MQCPE028 检测到的无效节: {0}

解释: 配置文件中找到声明的未识别的节。

用户回答: 只有 [global] 和 [route] 节在配置文件中是有效的。

MQCPE029 不能刷新日志输出。

解释: 有些消息可能没有写到日志，因为无法刷新通信缓冲区。

用户回答: 检查 MQIPT 主目录磁盘是否已变满，且 MQIPT 是否仍能访问 logs 子目录。

MQCPE030 没有在 CLASSPATH 中找到 {0}。

解释: 系统环境 CLASSPATH 变量中没有找到指定的 jar 文件。

用户回答: 添加指定的文件到系统 CLASSPATH。

MQCPE031 未找到 {0} 类。

解释: 显示 MQIPT 版本号时生成此消息。无法在 MQIPT jar 文件中找到指定的类或者系统环境 CLASSPATH 变量已经损坏。

用户回答: 检查指定的类文件在 “MQipt.jar” 文件中，且 “MQipt.jar” 文件在系统 CLASSPATH 中。

MQCPE033 向 {0} 处的管理客户机发送配置文件失败

解释: 发送配置文件到管理客户机时发生错误。

用户回答: 检查配置文件在 MQIPT 主目录中，且不被另一个进程共享。

MQCPE034 {0} 处的管理客户机没有提供正确密码。

解释: 配置文件中的 AccessPW 属性与管理客户机提供的 AccessPW 属性不匹配。

用户回答: 更改配置文件中的 AccessPW 属性，或者在管理客户机中保存密码。

MQCPE035 在端口 {0} 上启动命令侦听器失败

解释: 在指定端口地址上启动命令侦听器时发生 I/O 错误。

用户回答: 检查配置文件中 CommandPort 属性使用的端口地址。

MQCPE038 MQIPT 没有象期待的那样启动

解释: 此消息是 mqipt 创建子进程进程生成的，它把 MQIPT 作为系统服务启动。

用户回答: 检查出错日志以获取更多信息。您可以尝试增加 IPTFork 检查 MQIPT 是否正在运行的之前使用的休眠时间。编辑 mqiptFork 脚本并增加传递给 IPTFork 的参数。

MQCPE039 运行 mqipt 脚本时发生 I/O 错误

解释: 从创建子进程进程启动 MQIPT 期间发生了一个错误。

用户回答: 检查系统 PATH 环境变量是否包含 JDK 的位置，且 mqipt 脚本具有执行权限。

MQCPE040 运行 mqipt 脚本时发生中断

解释: 从创建子进程进程启动 MQIPT 之后发生了一个错误。

用户回答: 检查出错日志以获取更多信息。如果此情况仍然存在，请联系 IBM 技术支持。

MQCPE041 不支持的 Java 级别 - {0}

解释: 使用指定的 Java 级别启动了 MQIPT。

用户回答: 检查用户指南中的先决条件以获取更多信息。

MQCPE042 和路由 {0} 上的下列属性存在冲突:

解释: 有些属性无法与其它属性一起使用。此消息在有冲突的属性列表的前面。

用户回答: 检查下面的错误消息并采取相应的措施。

MQCPE043 ....{0} 和 {1}

解释: 无法在同一路由上同时设置下列两个属性。

用户回答: 编辑配置文件并禁用给定路由中指定的属性之一。

MQCPE044 {0} 仅在 {1} 操作系统上有效

解释: 有些 MQIPT 的功能只在特定的平台上有效。

用户回答: 编辑配置文件并禁用指定的属性。

MQCPE045 ....缺少 HTTP 代理名称

解释: 如果 HTTP 属性设置为 ture，则必须设置 HTTPProxy 属性。

用户回答: 编辑配置文件并为给定的路由定义 HTTPProxy。

MQCPE046 因为 Pagent 初始化失败，所以不允许 {0}

解释: Pagent 是提供 MQIPT 的服务质量的应用程序。启动期间 MQIPT 初始化该程序失败，而给定路由的 QoS 属性设置为 ture。

用户回答: 编辑配置文件并禁用给定路由的 Qos。

MQCPE047 Pagent 初始化失败

解释: Pagent 是提供 MQIPT 的服务质量的应用程序。启动期间 MQIPT 初始化该程序失败。

用户回答: 如果没有使用 Pagent，可忽略此错误消息。但您必须把 QoS 属性设置为 false。

MQCPE048 在端口 {0} 上路由启动失败，异常是: {1}

解释: 不可能使用指定的 ListenerPort 号启动路由。

用户回答: 检查其它相邻的错误消息和日志记录以提供问题的进一步解释。

MQCPE049 启动或停止 Java 安全性管理器 {0} 时出错

解释: 试图启动或停止 Java 安全性管理器时，抛出了一个异常。

用户回答: 之前已启用了 Java 安全性管理器，但是没有启用运行时许可权。添加用于 setSecurityManager 的 RuntimePermission 到您的本地策略文件。必须重新启动 MQIPT 以使更改生效。

MQCPE050 管理客户机端口 {0} 上发生安全性异常

解释: 接受来自管理客户机的连接时，抛出了一个安全性异常。

用户回答: 之前启用了 Java 安全性管理器，但是没有把许可权授予错误消息中标识的主机。要使主机能连接到 MQIPT，添加 SocketPermission 以接受／解析 CommandPort 的端口地址上的连接。必须重新启动 Java 安全性管理器以使更改生效。

MQCPE051 接受路由 {0} 上的连接时发生安全性异常

解释: 接受指定路由上的连接时，抛出了一个安全性异常。

用户回答: 之前启用了 Java 安全性管理器，但是没有把许可权授予错误消息中标识的主机。要使主机能连接到此路由，添加 SocketPermission 以接受／解析 ListenerPort 的连接。必须重新启动 Java 安全性管理器以使更改生效。

MQCPE052 路由 {0} 上的连接请求失败: {1}

解释: 此消息在连接日志中出现以记录连接请求的安全性异常。

用户回答: 之前启用了 Java 安全性管理器，但是没有把许可权授予错误消息中标识的主机。要使主机能连接到此路由，添加 SocketPermission 以接受／解析 ListenerPort 的连接。必须重新启动 Java 安全性管理器以使更改生效。

MQCPE053 连接到 {0}（{1}）时发生安全性异常

解释: 在指定路由上进行连接时，抛出了一个安全性异常。

用户回答: 之前启用了 Java 安全性管理器，但是没有把许可权授予错误消息中标识的主机。要使主机能连接到此路由，添加 SocketPermission 以接受／解析 ListenerPort 的连接。必须重新启动 Java 安全性管理器以使更改生效。

MQCPE054 到 {0}（{1}）的连接请求失败: {2}

解释: 此消息在连接日志中出现以记录到目标主机的连接请求的安全性异常。

用户回答: 之前启用了 Java 安全性管理器，但是没有把许可权授予错误消息中标识的主机。要使主机能连接到此路由，添加 SocketPermission 以接受／解析 ListenerPort 的连接。必须重新启动 Java 安全性管理器以使更改生效。

MQCPE055 ....缺少 Socks 代理名称

解释: 如果 SocksClient 属性设置为 ture，则必须设置 SocksProxy 属性。

用户回答: 编辑配置文件并为给定的路由定义 SocksProxy。

MQCPE056 和路由属性冲突

解释: 有些属性无法与其它属性一起使用。

用户回答: 检查控制台消息以获取错误的详细信息并采取相应的措施。

MQCPE057 未识别 SSL 协议（{0}）

解释: 路由已处于 SSL 代理方式，但是初始数据流是未识别的。

用户回答: 确保此路由只有 SSL 连接。

MQCPE058 经过 {0}（{1}）到 {2}（{3}）的 CONNECT 请求失败

解释: 向 HTTP 代理发送了 HTTP CONNECT 请求，以创建到 HTTP 服务器的 SSL 隧道。HTTP 代理没有对此请求发送回 “200 OK” 响应。

用户回答: 这可能是由多种问题引起的。在路由上启用跟踪，然后重试连接。跟踪文件中将显示真正的错误。

MQCPE059 没有定义过的密钥环文件

解释: 定义了 SSL 客户机或服务器，但没有为它至少指定一个密钥环文件。

用户回答: 使用客户机端的 SSLClientKeyRing 和 SSLClientCAKeyRing 属性，或服务器端的 SSLServerKeyRing 和 SSLServerCAKeyRing 属性来定义密钥环文件，然后重新启动路由。

MQCPE060 将 SSL 客户机连接超时设置为 {0} 秒时发生运行时错误

解释: 设置超时值时在客户机端发生 SSL 运行时错误。

用户回答: 检查 SSLClientConnectTimeout 属性中指定的值是否有效。在给定的路由上运行跟踪将显示进一步的错误信息。

MQCPE061 没有启用的密码套件

解释: 已启动了 SSL 客户机或服务器连接，但 MQIPT 无法确定有效的密码套件。

用户回答: 检查定义的密钥环文件中是否存在有效的证书。用于生成证书的专用和公用密钥，以及使用的加密算法必须遵照支持的密码套件的列表，该列表可在 MQIPT 书籍中找到。

MQCPE062 设置 SSL 密码套件 {0} 时发生运行时错误

解释: 在客户机或服务器端定义了不支持的 SSL 密码套件。

用户回答: 检查 SSLClientCipherSuites 或 SSLServerCipherSuites 中指定的值是否有效，以及该连接上是否支持该值。在给定的路由上运行跟踪将显示启用的密码套件的列表。MQIPT 书籍包含支持的密码套件的列表。

MQCPE063 文件 {0} 已存在 - 使用 replace 选项

解释: 为 mqiptPW 脚本指定的文件名参数已存在。

用户回答: 选择另一个文件名或使用替换选项。

MQCPE064 生成解密密钥时发生运行时错误: {0}

解释: 生成对用于打开密钥环文件的密码进行解密的密码密钥时发生错误。

用户回答: 应改正消息中列出的运行时错误，然后再次运行命令。

MQCPE065 缺少 LDAP 服务器名

解释: 如果 LDAP 属性设置为 true，则必须设置 LDAPServer1 或 LDAPServer2 属性。

用户回答: 编辑配置文件并为给定的路由定义 LDAPServer*。

MQCPE066 LDAPServer{0}Password 属性缺少 LDAP 密码

解释: 指定了 LDAP 用户标识，但没有为它指定密码。

用户回答: 编辑配置文件并为给定的路由定义 LDAPServer*Password。

MQCPE067 LDAP 服务器缺少 SSLClient 或 SSLServer

解释: 如果 LDAP 属性设置为 true，则必须设置 SSLClient 或 SSLServer 属性。

用户回答: 编辑配置文件并为给定的路由定义 SSLClient 或 SSLServer。

MQCPE068 缺少安全性出口名称

解释: 如果 SecurityExit 属性设置为 ture，则必须设置 SecurityExitName 属性。

用户回答: 编辑配置文件并为给定的路由定义 SecurityExitName。

MQCPE069 安全性出口响应中无效的端口地址 {0}

解释: SecurityExitResponse 中指定的端口地址无效。

用户回答: 端口地址的范围必须在 1024 - 65535 之间。

MQCPE070 安全性出口响应中未知的原因码 {0)

解释: 不支持 SecurityExitResponse 中指定的原因码。

用户回答: 请参阅 MQIPT 书籍以获取支持的原因码的列表。

MQCPE071 写到 {0} 时出错

解释: 创建或更新指定的文件时发生错误。错误消息还包含抛出的异常。

用户回答: 应改正异常中列出的错误，然后再次运行命令。

MQCPE072 安全性出口 {0} 中发生未知错误

解释: 验证连接请求时，在用户定义的安全性出口中发生错误。

用户回答: 启用安全性出口中的跟踪，然后再次尝试连接请求。错误将记录在安全性出口跟踪文件中。

MQCPI001 {0} 正在启动

解释: 此 MQIPT 实例开始执行。接下来会有进一步的初始化消息。

MQCPI002 {0} 正在关闭

解释: MQIPT 即将关闭。这可能是 STOP 命令引发的。或是由于存在影响启动成功的配置错误而自动发生的。还可能是 REFRESH 操作引起的。

MQCPI003 {0} 完全关闭

解释: 关闭进程已完成。所有 MQIPT 进程现在已结束。

MQCPI004 从 {0} 读配置信息

解释: 正在从此消息中描述的目录读取 MQIPT 配置文件 mqipt.conf。

MQCPI005 指定的侦听器端口为不活动 – {0} -> {1}（{2}）

解释: 消息中引用的路由已被标记为非活动的。此路由将不接受任何通信请求。

MQCPI006 路由 {0} 正在启动并将转发消息到:

解释: 在此消息显示的侦听器端口启动了一个路由。此消息后跟列出所有与此路由关联的属性的其它消息。当路由接受连接准备就绪时，将发出消息 MQCPI078。

MQCPI007 路由 {0} 已停止

解释: 指定 ListenerPort 上操作的路由正在关闭。通常是在向 MQIPT 发出 REFRESH 命令和更改了路由配置的时候发生此操作。

MQCPI008 在端口 {0} 上侦听控制命令

解释: 此 MQIPT 实例正在指定的端口上侦听控制命令。

MQCPI009 接收的控制命令: {0}

解释: 此消息表明在命令端口上接收到控制命令。在适用的地方，消息中包含了详细信息。

MQCPI010 正在停止 {0} 上的命令端口

解释: REFRESH 操作期间，命令端口不再在新的配置中使用。指定端口上不再接受命令。

MQCPI011 路径 {0} 将用于存储日志文件

解释: 根据当前配置，日志将输出到此消息中所描述的位置。

用户回答: 如果配置改变了且请求了 REFRESH 操作，这可能会改变。

MQCPI012 MinConnectionThreads 值的更改在启动路由后不生效

解释: 路由启动时指定了最小的连接线程数目，直到 MQIPT 重新启动之后才能更改。

MQCPI013 从 {0} 到主机 {1} 的连接关闭

解释: 此消息在连接日志中出现以记录连接活动。

MQCPI014 未识别 Eyecatcher 协议（{0}）

解释: 此消息在连接日志中出现以记录连接活动。

MQCPI015 已在此路由上禁用客户机访问

解释: 此消息在连接日志中出现以记录连接活动。

MQCPI016 已在此路由上禁用队列管理器访问

解释: 此消息在连接日志中出现以记录连接活动。

MQCPI017 {0} 上的队列管理器连接到主机 {1}

解释: 此消息在连接日志中出现以记录连接活动。

MQCPI018　{0} 上的客户机连接到主机 {1}

解释:　此消息在连接日志中出现以记录连接活动。

MQCPI019　已创建 {0} 路由 – 这超出最大支持路由数，即是 {1}

解释:　已超过最大支持的路由数目。

用户回答:　MQIPT 将继续运行，但建议创建第二个 MQIPT 实例并在这两个实例中分配这些路由。

MQCPI020　配置文件已被发送到管理客户机。

解释:　由于来自管理客户机的请求，已发送了配置文件。

MQCPI021　密码检查在命令端口上已启用。

解释:　此消息表明访问命令端口需要密码。

MQCPI022　密码检查在命令端口上是禁用的。

解释:　此消息表明访问命令端口不需要密码。

MQCPI024　....使用 HTTP 代理 {0}（{1}）

解释:　此消息表明此路由的外出连接将使用 HTTP 代理。

MQCPI025　管理客户机 {0} 请求的刷新已完成。

解释:　由于接收到了 REFRESH 命令，MQIPT 重新读取了其配置文件并重新启动。

MQCPI026　管理客户机 {0} 已请求关机。

解释:　由于接收到了 STOP 命令，MQIPT 正在关闭。

MQCPI027　{0} 发送到端口 {2} 上的 {1}

解释:　这在系统控制台上显示由行方式（非 GUI）管理客户机发出到指定的 MQIPT 的命令。

MQCPI031　.....密码套件 {0}

解释:　此消息列出此路由使用的密码套件。

MQCPI032　.....密钥环文件 {0}

解释:　此消息给出此路由的密钥环的文件名。

MQCPI033　.....客户机认证设置为 {0}

解释:　此消息定义 SSL 服务器是否要对此路由进行客户机认证。

MQCPI034　....{0}（{1}）

解释:　此消息显示此路由的目标和目标端口地址。

MQCPI035　....使用 {0}

解释:　此消息显示了目标使用的协议。它可以是 MQSeries 协议、HTTP 隧道或 HTTP 分块。

MQCPI036　....使用下列属性启用 SSL 客户机端:

解释:　此消息表明路由将使用 SSL 发送数据到目标主机。

MQCPI037　....使用下列属性启用 SSL 服务器端:

解释:　此消息表明路由将使用 SSL 接收来自发送主机的数据。

MQCPI038　.....对等证书使用 {0}

解释:　此消息列出用于控制对等证书的认证的专有名称。

MQCPI039　....通过 Socks 代理 {0}({1})

解释:　此消息表明此路由的外出连接将使用 Socks 代理（这是在从命令行启动 MQIPT 时定义的）。

MQCPI040　命令端口已被管理客户机 {0} 访问

解释:　此消息写到系统控制台和 MQIPT 日志文件（如果启用了记录日志）。MQIPT 接收到来自管理客户机的连接。

MQCPI041　....将以 {0} 方式应答 Network Dispatcher 顾问程序请求

解释:　当路由启动时，此消息写到系统控制台。用于显示 MQIPT 将使用哪种方式来应答 Network Dispatcher 顾问程序。有效的选项是“正常”和“替换”方式。

MQCPI042　路由 {0} 上的最大连接数已达到 – 将阻塞更多的请求

解释:　当给定的路由达到最大连接数时，此消息写到系统控制台。接下来更多请求将被阻塞，直到有空闲的连接或者增大 MaxConnectionThreads 值为止。

MQCPI043　现在不阻塞路由 {0} 上的连接

解释:　当给定的路由不被连接请求阻塞时，此消息写到系统控制台。

MQCPI044　MQIPT 已从系统启动启动

解释:　MQIPT 已作为系统服务启动。

MQCPI045　从系统启动启动 MQIPT

解释:　MQIPT 将作为系统服务启动。

MQCPI046　当 MQIPT 从系统启动启动时，休眠 {0} 秒

解释:　创建子进程进程在检查 MQIPT 是否已作为系统服务成功启动之前休眠这些时间。

MQCPI047　.....CA 密钥环文件 {0}

解释:　此消息给出此路由的 CA 密钥环的文件名。

MQCPI048　由管理客户机 {0} 发出的 ping 完成

解释:　IPTController 给管理客户机的响应消息。

MQCPI049　....目标的 QoS 优先级 = {0}，调用程序的 QoS 优先级 = {1}

解释:　这表明此路由的两个方向的流量的优先级。

MQCPI050　添加条目到 inittab 以在系统启动时自动启动 MQIPT

解释:　用户运行了 mqiptService 脚本以把 MQIPT 作为系统服务启动。

MQCPI051　从 inittab 除去在系统启动时自动启动 MQIPT 的条目

解释:　用户运行了 mqiptService 脚本以不再把 MQIPT 作为系统服务启动。

MQCPI052　....启用 Socks 服务器端

解释:　此路由将充当 SOCKS 服务器（代理）并将接受来自 socks 化的应用程序的连接。

MQCPI053　正在启动 Java 安全性管理器

解释:　由于 SecurityManager 属性设置为 ture，将启动缺省 Java 安全性管理器

MQCPI054　正在停止 Java 安全性管理器

解释:　由于 SecurityManager 属性设置为 false，将停止缺省 Java 安全性管理器。

MQCPI055　正在设置 java.security.policy 为 {0}

解释:　即将启动缺省 Java 安全性管理器，并将使用提供的策略文件。

MQCPI056　必须重新启动 Java 安全性管理器以使用新的策略文件

解释:　已更改了 SecurityManagerPolicy 属性，但是要等到重新启动 Java 安全性管理器之后才能生效。

用户回答:　将 SecurityManager 属性更改为 false，发出刷新命令以停止 Java 安全性管理器。然后把 SecurityManager 改回 true，并发出另一条刷新命令以使用新的策略文件启动 Java 安全性管理器。

MQCPI057　....启用跟踪级别 {0}

解释:　当路由启动时，此消息写到系统控制台。用于显示此路由启用的跟踪级别。

MQCPI058　....和 {0} 的 URI 名称

解释:　当路由启动时，此消息写到系统控制台。用于显示此路由的统一资源标识符名称。

MQCPI059　....启用 servlet 客户机

解释:　当路由启动时，此消息写到系统控制台。此路由将连接到 MQIPT servlet。

MQCPI060　正在安装文件以在系统启动时自动启动 MQIPT

解释:　用户运行了 mqiptService 脚本以把 MQIPT 作为系统服务启动。

MQCPI061　正在除去在系统启动时自动启动 MQIPT 的文件

解释:　用户运行了 mqiptService 脚本以不再把 MQIPT 作为系统服务启动。

MQCPI064　....此路由上无 SSL 认证

解释:　当路由启动时，此消息写到系统控制台。此消息表明此路由没有使用 SSL 认证（因为指定了匿名密码套件）。

MQCPI065 ....以 SSL 代理方式

解释: 当路由启动时，此消息写到系统控制台。此消息表明路由工作在 SSL 代理方式下。

MQCPI066 ....和在 {0}{1} 处的 HTTP 服务器

解释: 此消息表明将使用此 HTTP 服务器生成用于此路由的外出连接。

MQCPI067 正在设置到 TQoS 运行时库的链接

解释: 用户运行了 mqiptQoS 脚本以链接到实 TQoS 运行时库。

MQCPI068 正在除去到 TQoS 运行时库的链接

解释: 用户运行了 mqiptQoS 脚本以除去到实 TQoS 运行时库的链接。

MQCPI069 ....正在绑定到本地地址 {0}

解释: 此消息显示每个连接绑定到的本地 IP 地址。这应该只在多主机系统上使用。

MQCPI070 ....正在使用本地端口地址范围 {0}-{10}

解释: 此消息显示将用于连接的本地端口地址。这将允许防火墙管理员限制来自 MQIPT 的连接。

MQCPI071 站点证书使用 {0}

解释: 此消息列出用于控制站点证书选择的专有名称。

MQCPI072 .....和证书标号 {0}

解释: 此消息列出用于控制站点证书选择的标号名称。

MQCPI073 已更新的文件 {0}

解释: 已更新为 mqiptPW 脚本指定的文件名。

MQCPI074 已创建的文件 {0}

解释: 已创建为 mqiptPW 脚本指定的文件名。

MQCPI075 ....在 {0}（{1}）处的 LDAP 主服务器

解释: 此消息列出用于 CRL 支持的主 LDAP 服务器的名称。

MQCPI076 ....在 {0}（{1}）处的 LDAP 备份服务器

解释: 此消息列出用于 CRL 支持的备份 LDAP 服务器的名称。

MQCPI077 ....将忽略 LDAP 错误

解释: 此消息表明将忽略从 LDAP 接收到的任何错误。

MQCPI078 路由 {0} 用于连接请求准备就绪

解释: 当路由用于接受连接请求准备就绪时，将显示此消息。

MQCPI079 ....正在使用安全性出口 {0}

解释: 当路由启动时，此消息写到系统控制台。用于显示安全性出口的全限定名。

MQCPI080 .....和超时 {0} 秒

解释: 当路由启动时，此消息写到系统控制台。用于显示安全性出口的超时值。

MQCPI081 启动用于 WebSphere MQ internet pass-thru 的消息

解释: 将 WebSphereMQ internet pass-thru 的消息作为一个服务启动

MQCPI082 停止 WebSphere MQ internet pass-thru 的消息

解释: 将 WebSphere MQ internet pass-thru 的消息作为一个服务停止

MQCPI083 ....刷新命令不会重新启动路由

解释: 此消息表明当发出刷新命令时，路由不会重新启动。

MQCPI084 ....CRL 高速缓存到期超时为 {0} 小时

解释: 此控制台消息显示 CRL（或 ARL）将在 MQIPT 高速缓存中保留多久。

MQCPI085 ....CRL 将保存在密钥环文件中

解释: 此控制台消息表明从 LDAP 服务器检索的任何 CRL（或 ARL）将保存在密钥环文件中，该文件连接到关联的 CA 证书。

MQCPI086 .....超时 {0} 秒

解释: 当路由启动时，此消息写到系统控制台。用于显示连接到 LDAP 服务器的超时值。

MQCPI087 .....用户标识为 {0}

解释: 当路由启动时，此消息写到系统控制台。用于显示连接到 LDAP 服务器的用户标识名称。

MQCPI100 此脚本用于启动 {0}

解释: 来自 mqipt 脚本的联机帮助消息。

MQCPI101 命令格式是:

解释: 来自 mqipt 脚本的联机帮助消息。

MQCPI102 mqipt {dir_name}

解释: 来自 mqipt 脚本的联机帮助消息。

MQCPI103 dir_name – 包含 mqipt.conf 的目录

解释: 来自 mqipt 脚本的联机帮助消息。

MQCPI106 此脚本用于显示当前版本号

解释: 来自 mqiptVersion 脚本的联机帮助消息。

MQCPI107 mqiptVersion {-v}

解释: 来自 mqiptVersion 脚本的联机帮助消息。

MQCPI108 其中 -v 还将显示构建时间戳记

解释: 来自 mqiptVersion 脚本的联机帮助消息。

MQCPI109 此脚本用于从另一个 JVM 的系统启动来启动 {0}，且只在 mqipt.ske 中使用。使用 mqipt 脚本以从命令行启动 MQIPT。

解释: 来自 mqiptFork 脚本的联机帮助消息。

MQCPI110 此类用于在控制台上显示一个简单的 NLS 消息

解释: 来自 IPTMessages 类的联机帮助消息。

MQCPI111 java com.ibm.mq.ipt.IPTMessages (message_id1) {message_id2} {message_id...}

解释: 来自 IPTMessages 类的联机帮助消息。

MQCPI112 其中 message_id 与文件 mqipt.properties 中的一个键匹配

解释: 来自 IPTMessages 类的联机帮助消息。

MQCPI113 此脚本用于将 MQIPT 作为系统服务进行管理

解释: 来自 mqiptService 脚本的联机帮助消息。

MQCPI114 mqiptService (-install | -remove)

解释: 来自 mqiptService 脚本的联机帮助消息。

MQCPI115 -install 将安装文件以在系统启动时自动启动 MQIPT

解释: 来自 mqiptService 脚本的联机帮助消息。

MQCPI116 -remove 将除去系统启动时自动启动 MQIPT 的文件

解释: 来自 mqiptService 脚本的联机帮助消息。

MQCPI117 此脚本用于管理到 TQoS 运行时库的链接

解释: 来自 mqiptService 脚本的联机帮助消息。

MQCPI118 mqiptQoS (-install | -remove)

解释: 来自 mqiptService 脚本的联机帮助消息。

MQCPI119 -install 将设置到实 TQoS 运行时库的链接

解释: 来自 mqiptService 脚本的联机帮助消息。

MQCPI120 -remove 将除去到实 TQoS 运行时库的链接

解释: 来自 mqiptService 脚本的联机帮助消息。

MQCPI121 使用此脚本加密密码并将它存储在一个文件中

解释: 来自 mqiptPW 脚本的联机帮助消息。

MQCPI122 mqiptPW password file_name { -replace }

解释: 来自 mqiptPW 脚本的联机帮助消息。

MQCPI123 password – 用于打开密钥环文件的密码

解释: 来自 mqiptPW 脚本的联机帮助消息。

MQCPI124 file_name – 加密的密码将存储在此文件中

解释: 来自 mqiptPW 脚本的联机帮助消息。

MQCPI125 replace 选项必须用于更新现有文件

解释: 来自 mqiptPW 脚本的联机帮助消息。

MQCPI126 mqipt (-start | -stop)

解释: 来自 mqiptQoS 脚本的联机帮助消息。

MQCPW001 对于 {0}, CRL 到期

解释: 当从 LDAP 服务器或密钥环文件检索到 CRL（或 ARL）时，将显示此消息。

用户回答: 更新 LDAP 服务器或密钥环文件中的指定 CRL。

MQCPW002 使用 CRL 更新密钥环文件 {0} 时出错

解释: 当启用了 LDAPSaveCRLs 属性且无法更新指定的密钥环文件时，将显示此消息。

用户回答: 指定的文件可能已损坏。检查下列事项:

1. 必须已为 MQIPT 启用了写访问
2. 文件没有被另一个应用程序打开

MQCPW003 ....到期的 CRL 将被忽略

解释: 此控制台消息表明任何到期的 CRL（或 ARL）将被忽略且可以允许连接请求。

附录. 声明

本条款不适用任何这样的条款与当地法律不一致的国家或地区。

国际商业机器公司以“按现状”的基础提供本出版物，不附有任何形式的（无论是明示的，还是默示的）保证，包括（但不限于）对非侵权性、适销性和适用于某特定用途的默示保证。某些国家或地区在某些交易中不允许免除明示或默示的保证。因此本条款可能不适用于您。

本出版物中对 IBM 产品、程序或服务的引用并不暗示 IBM 将在所有有 IBM 业务的国家或地区中提供这些产品、程序或服务。

本出版物中任何对 IBM 许可程序或其它 IBM 产品的引用并非意在明示或暗示只能使用 IBM 的程序或其它产品。只要不侵犯任何知识产权，任何同等功能的程序都可以代替 IBM 产品。在与其它产品结合使用时，除了那些由 IBM 明确指定的产品之外，其评估和验证均由用户自行负责。

IBM 可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可证。有关许可证查询的事宜，用户可以与 IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, USA 书面联系。

本文档中所包含的信息未提交进行任何正式的 IBM 测试，并且以“按现状”分发。此信息的使用或任何这些技术的实现由客户自行负责，并且取决于客户将它们评估和集成到客户操作环境的能力。当 IBM 在某一特定环境中复查每一项的准确性时，不能保证在其它地方得到相同或相似的结果。尝试在自己的环境中应用这些技术的客户应自行承担风险。

商标

下列术语是国际商业机器公司在美国和／或其它国家或地区的商标:

AIX	FFST	First Failure Support Technology
IBM	IBMLink	MQSeries
SupportPac	WebSphere	

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

Java 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和／或其它国家或地区的商标。

UNIX 是 The Open Group 在美国和其它国家或地区的注册商标。

其它公司、产品和服务名称可能是其它公司的商标或服务标记。

文献目录

本书以 HTML 格式作为安装产品的一部分提供。该 HTML 文件包含在 doc\<locale>\html\<filename>.zip 目录中的自解压 zip 文件中。在使用管理客户机前，您必须解压缩（unzip）在 <locale>\html 子目录中找到的文件。本书出版过下列语言版本，请参阅下表以了解各语言及相应的文件名:

表 4. 语言和文件名摘要

语言	语言环境	HTML 文件名
简体中文	zh_CN	amqyzb01.zip
德语	de_DE	amqygb01.zip
日语	ja_JP	amqyjb01.zip
韩国语	ko_KR	amqykb01.zip
巴西葡萄牙语	pt_BR	amqybb01.zip
西班牙语	es_ES	amqysb01.zip
美国英语	en_US	amqyab01.zip

可到以下 URL 下载翻译过的 PDF:

http://www.ibm.com/webspheremq/downloads

它提供了以下语言版本:

表 5. PDF 语言和文件名

语言	语言环境	PDF 文件名
简体中文	zh_CN	amqyzb01.pdf
德语	de_DE	amqygb01.pdf
日语	ja_JP	amqyjb01.pdf
韩国语	ko_KR	amqykb01.pdf
巴西葡萄牙语	pt_BR	amqybb01.pdf
西班牙语	es_ES	amqysb01.pdf
美国英语	en_US	amqyab01.pdf

您还会找到下列有用的出版物:

- *WebSphere MQ Intercommunication*, SC34-6059
- *《WebSphere MQ 系统管理指南》*, S152-0262
- *《WebSphere MQ 客户机》*, G152-0367
- *WebSphere MQ Queue Manager Clusters*, SC34-6061

这些书提供关于 WebSphere MQ 通道及其属性的定义信息 – 尤其是 CONNAME 的定义。

WebSphere MQ 出版物可从下列地址获取:

http://www.ibm.com/webspheremq/library

索引

[T]

[W]

[X]

[Z]

A

C

D

F

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

W

X

将您的意见发送给 IBM

如果您对本书中的任何内容有特别欣赏或者非常不赞成的地方，请使用下面列出的一种方法将您的意见发送给 IBM。

欢迎对本书中您认为是明确的错误或疏忽的部分，以及本书的准确性、组织性、主旨或完整性提出您的意见。

请将您的意见仅限于本书中的信息以及信息表达的方式。

有关 IBM 产品或系统功能的意见，请与 IBM 代表或 IBM 授权的分销商联系。

当您将意见发送到 IBM 后，即授予 IBM 非专有权，IBM 可以它认为合适的任何方式使用或分发您的意见，而无须对您承担任何责任。

可以按下列任何一种方式将您的意见发送给 IBM:

- 邮寄到以下地址:

  IBM 中国公司上海分公司，汉化部
  中国上海市淮海中路 333 号瑞安广场 10F
  邮政编码: 200021
- 通过传真，请使用以下号码:
  - 中国: 021–63857881
  - 其它国家或地区: (86–21)63857881
- 以电子方式发送，请使用适当的网络标识:
  - IBMLink™: ibmcn(ctscrcf)
  - 因特网: ctscrcf@cn.ibm.com

无论您使用哪种方法，请确保包含了下列信息:

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