9.3

IBM MQ 方案

注

在使用本资料及其支持的产品之前，请阅读第 193 页的『声明』中的信息。

本版本适用于 IBM® MQ V 9 发行版 3 以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。

当您向 IBM 发送信息时，授予 IBM 以它认为适当的任何方式使用或分发信息的非独占权利，而无需对您承担任何责任。

© Copyright International Business Machines Corporation 2007, 2025.

内容

方案.....5
开始使用 IBM MQ..... 5
规划解决方案..... 5
实现解决方案..... 6
后续操作.....16
点到点方案..... 16
规划解决方案..... 17
实现解决方案..... 18
保护点到点拓扑.....23
流式队列..... 26
流式队列配置..... 27
流式传输到远程队列和别名队列.....28
流式队列限制..... 29
流队列和事务..... 30
在集群队列之间进行流式传输.....30
使用流式队列来存储消息历史记录..... 30
发布/预订方案..... 31
方案: 创建发布/预订集群..... 31
发布/预订层次结构方案..... 37
业务支持方案.....45
介绍工作单元..... 45
方案 1: 队列管理器执行协调.....46
方案 2: 其他软件提供协调.....66
即将到期的全球工作单元.....72
恢复单元处置..... 72
安全方案..... 72
安全方案: z/OS 上的两个队列管理器.....73
安全方案: z/OS 上的队列共享组..... 80
服务器到服务器消息通道拦截示例配置.....84
使用 SSL/TLS 连接两个队列管理器..... 86
安全地将客户机连接到队列管理器..... 92
在 Windows 上迁移..... 98
规划解决方案..... 98
使用图形用户界面实现解决方案.....102
安装更高版本的 IBM MQ 以与 Windows 上的较低版本共存..... 123
多个安装的概述..... 123
将更高版本的 IBM MQ 并排安装到更低版本。..... 124
使用 **setmqenv** 命令与两个版本的 IBM MQ 一起运行..... 126
创建队列管理器..... 127
将队列管理器迁移到更高版本的 IBM MQ..... 128
在 IBM MQ 9.3 上安装修订包..... 130
Managed File Transfer 方案 (scenario).....132
MFT 常见拓扑..... 132
配置基本服务器..... 135
开始使用 IBM MQ Internet Pass-Thru.....142
验证 MQIPT 是否正常工作.....143
创建密钥环文件..... 145
创建测试证书..... 147
认证 TLS 服务器.....148
认证 TLS 客户机.....150
认证 TLS 客户机和服务器..... 153
配置 HTTP 隧道.....156

配置访问控制.....158
配置 SOCKS 代理.....160
配置 SOCKS 客户机.....161
配置 MQIPT 集群支持.....162
分配端口号.....165
使用 LDAP 服务器检索 CRL.....167
以 TLS 代理方式运行 MQIPT.....170
使用安全管理器以 TLS 代理方式运行 MQIPT.....171
使用安全出口.....174
使用安全出口将客户机连接请求路由到 IBM MQ 队列管理器服务器.....175
动态路由客户机连接请求.....178
使用证书出口来认证 TLS 服务器.....181
Kafka Connect 方案.....183
Kafka 连接公共拓扑.....184
仅支持一次.....192

声明.....193
编程接口信息.....194
商标.....194

IBM MQ 方案

每个方案都会指导您完成一组重要任务，并帮助您配置主要产品功能部件。 这些场景包括指向其他内容的有用链接，以帮助您更好地了解感兴趣的区域。

以下子主题描述了可用的 IBM MQ 方案。

Windows 开始使用 IBM MQ

此场景说明如何在 Windows 平台上开始使用 IBM MQ 。 如果从未使用过 IBM MQ 并且想要快速入门，请使用此方案。

此方案描述了在 Windows 上安装，配置和验证 IBM MQ (如果尚未将其安装在系统上) 的基本步骤。 您可以使用图形用户界面或命令行界面来完成方案的步骤。

规划解决方案

选择用于在 Windows 上安装 IBM MQ 的方法。 使用引导您完成安装和配置过程的图形用户界面和向导，或者使用命令行来执行静默安装。

概述: 交付的逻辑拓扑

完成方案后交付的逻辑拓扑。

已安装的 IBM MQ 服务器实例允许创建 IBM MQ 对象: 队列和队列管理器。 您可以使用 IBM MQ Explorer 通过队列管理器从本地队列中放入和获取消息。 此场景完成后，交付的拓扑将类似于 图 1。

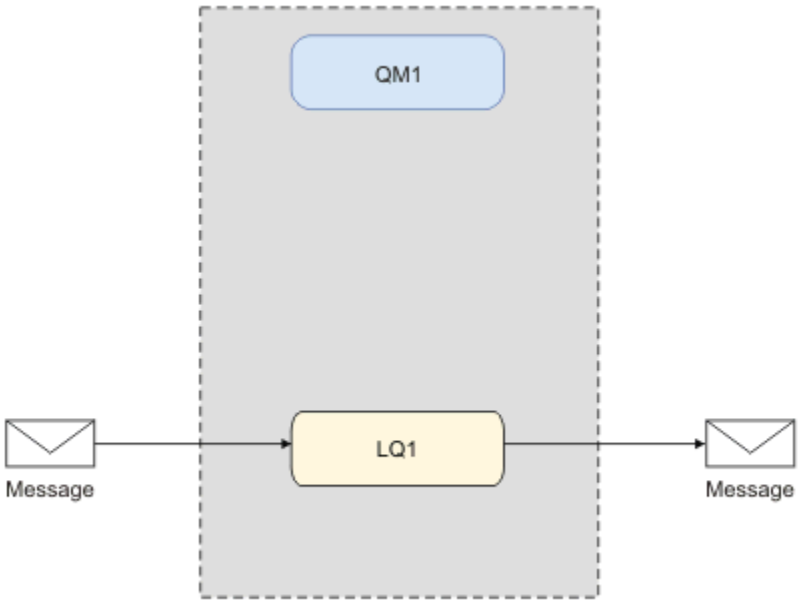


图 *1:* 将消息放在 *LQ1* 上，从 *LQ1* 获取消息。

基本概念和主要术语

描述了开始使用 IBM MQ 方案之前必须了解的基本概念和主要术语。

© Copyright IBM Corp. 2007, 2025

基本概念

通过 IBM MQ，应用程序可以针对队列读写消息。 读消息的应用程序和写消息的应用程序是相互独立的。 不要求两个应用程序同时运行。 如果没有可用的读消息应用程序，那么消息将在 IBM MQ 队列上排队，直到应用程序读取该消息。

在此方案中，您可以选择通过以下某种方式来安装和配置 IBM MQ：

第 6 页的『使用图形用户界面安装和配置』

使用图形用户界面安装期间，系统将通过若干屏幕和向导指导您应用相关选项和设置：

启动板
检查软件需求，指定网络信息并启动 IBM MQ 安装向导。

IBM MQ 安装向导
安装软件并启动“Prepare IBM MQ Wizard”。

Prepare IBM MQ Wizard
启动 IBM MQ 服务和 IBM MQ Explorer。

IBM MQ Explorer
管理队列和队列管理器。

第 10 页的『使用命令行界面安装和配置』
命令行界面安装可以是静默安装或者交互式安装。 静默安装完全可访问，本方案中涉及的即为静默安装。 使用命令行安装期间，系统将通过几个步骤指导您应用相关选项和设置：

- 安装 IBM MQ
- 创建和配置 IBM MQ 对象、队列管理器和队列。
- 使用 `amqsput` 将消息放入队列，并使用 `amqsget` 从队列获取消息，以验证安装。

使用 IBM MQ Explorer 和命令行创建 IBM MQ 对象，也可以使用可编程接口完成此操作。 当前的方案中未包括此内容。

关键词

下面是关于消息排队的关键术语列表。
有关消息排队的关键术语。

术语	描述
队列管理器	队列管理器负责维护它所拥有的队列，并负责将它接收到的所有消息存储到适当的队列中。
消息	消息是对使用它的应用程序有意义的以字节为单位的字符串。 消息用来将信息从一个应用程序传输到另一个应用程序。 应用程序可以在相同或不同的计算机上运行。
本地队列	本地队列是用于存储消息的数据结构。 该队列可以是常规队列或者传输队列。 直接从队列管理器读取消息的应用程序要读取的消息保留在常规队列中。 正在转移给其他队列管理器的消息保留在传输队列中。

实现解决方案

实现方案的解决方案。 使用安装启动板在 Windows 上安装 IBM MQ ，然后使用 IBM MQ Explorer 验证安装。

使用图形用户界面安装和配置

使用安装启动板在 Windows 上安装 IBM MQ ，并使用 IBM MQ Explorer 验证安装。 验证安装后，创建队列管理器和队列，然后尝试将消息放入队列并从队列中获取消息。

使用启动板进行安装

使用安装启动板在 Windows 上安装 IBM MQ 。

开始之前

在启动此任务之前，请完成以下检查:

- 安装时必须具有本地管理员权限。 通过 Windows 工具定义此权限。
- 确保机器名不包含任何空格。
- 确保您有足够的磁盘空间来完全安装 IBM MQ for Windows。 有关更多信息，请参阅 Multiplatforms 版上的磁盘空间需求。
- 确定是否需要为任何 IBM MQ 用户定义 Windows 域用户标识。

在安装 IBM MQ 之前，请检查系统是否满足硬件和软件需求。 有关所有受支持平台上的硬件和软件需求的更多信息，请参阅 系统要求 IBM MQ。

关于此任务

启动板和后续向导将引导您完成安装过程，并帮助您查看软件需求和 IBM MQ 设置。

此任务假定您是首次在机器上安装 IBM MQ ，并且您将使用缺省位置。 缺省情况下， IBM MQ 程序文件的位置为 `C:\Program Files\IBM\MQ`，数据和日志文件位置为 `C:\ProgramData\ IBM \ MQ`。

注: 如果要安装 IBM MQ 9.0，并且您的机器上有任何先前安装的 IBM MQ ，那么程序和数据文件的位置将与缺省位置不同。 有关更多信息，请参阅 程序和数据目录位置。 如果您先前已完成此方案，并且想要使用缺省位置通过单个全新安装重复此方案，请先除去先前的安装，然后再再次启动此方案。 要从机器卸载 IBM MQ 的现有实例，请参阅 第 15 页的『卸载 IBM MQ』。

如果您在安装过程中需要安装程序，那么安装程序包含指向进一步信息的链接。

过程

1. 启动启动板，查看并在必要时修改软件需求和网络配置。
   a) 浏览到 IBM MQ 软件目录，然后双击文件 `Setup.exe` 以启动启动板。
   b) 选择 **软件需求** 选项卡以显示 **软件需求** 设置。
   c) 检查是否已满足软件需求，以及需求条目是否显示带有 OK 字样的绿色刻度标记。 进行任何指示的更正。

      注:

      有关任何需求的详细信息，请单击该复选框以展开 "信息" 选项卡。
   d) 选择 **网络配置** 选项卡以显示 **网络配置** 设置。
   e) 选择**无**。

      注: 此场景假定您不需要为 IBM MQ 配置域用户标识。 有关配置 IBM MQ for Windows 域用户的更多信息，请单击 **更多信息**。
   f) 在启动板的 **IBM MQ 安装** 选项卡上，选择安装语言，然后单击 **启动 IBM MQ 安装程序** 以启动 IBM MQ 安装向导。

   您已通过满足或指定安装需求来完成设置 IBM MQ ，并且已启动 IBM MQ 安装向导。
2. 使用 IBM MQ 安装向导来安装软件，并启动 Prepare IBM MQ Wizard。
   a) 在 IBM MQ 安装向导中，阅读 "许可协议" ，单击 **我接受许可协议中的条款** 复选框，然后单击 **下一步**。
   b) 单击 **典型**，然后单击 **下一步**。
   c) 在 " **准备安装 IBM MQ** " 页面中，查看安装信息，然后单击 **安装**。

      注: 请注意以下详细信息:

      - 安装名称
      - 程序文件的顶级文件夹
      - 数据文件的顶级文件夹

安装了以下功能部件:

- IBM MQ 服务器
- IBM MQ: 用于管理和监视 IBM MQ 资源的图形界面
- Java 和 .NET 消息传递和 Web Service
- IBM MQ 开发工具箱

安装过程开始。 根据您的系统，安装过程可能需要几分钟时间。

在安装过程结束时， " IBM MQ 安装" 窗口将显示消息 `Installation Wizard Completed Successfully` 。

d) 单击**完成**。

您已成功安装 IBM MQ。 Prepare IBM MQ Wizard 将自动启动，并显示 **Prepare IBM MQ Wizard** 页面。

3. 使用 Prepare IBM MQ Wizard 来启动 IBM MQ 服务。
   a) 在 "欢迎使用 Prepare IBM MQ Wizard " 页面上，选择 **下一步**。

   Prepare IBM MQ Wizard 显示消息 `Status: Checking IBM MQ Configuration` 和进度条。 完成此过程后，将显示 " IBM MQ 网络配置" 页面。

   b) 在 Prepare IBM MQ Wizard 的 " IBM MQ 网络配置" 页面上，选择 **否**。

   c) 单击**下一步**。

   Prepare IBM MQ Wizard 显示消息 `Status: starting the IBM MQ Service` 和进度条。 完成此过程后，向导将显示 "完成 Prepare IBM MQ Wizard " 页面。

   d) 选择 **启动 IBM MQ 资源管理器** 并选择是否查看发行说明，然后单击 **完成** 按钮。

   IBM MQ Explorer 将启动。

已安装 IBM MQ。 您还启动了 IBM MQ Explorer。

结果

已安装并验证 IBM MQ ，并且您已准备好配置诸如队列管理器和队列之类的对象。

下一步做什么

遵循 第 8 页的『创建名为 QM1 的队列管理器』 中的指示信息。

相关概念

磁盘空间需求

Windows 系统上的硬件和软件需求

IBM MQ 简介

相关任务

在 Windows 上安装 IBM MQ 服务器

配置 IBM MQ 服务器

创建名为 *QM1* 的队列管理器

使用 IBM MQ Explorer 创建名为 QM1 的队列管理器。 队列管理器是 IBM MQ 消息传递网络中的主要组件。

开始之前

必须已安装 IBM MQ 。 如果未执行此操作，请参阅 第 6 页的『使用启动板进行安装』 以获取有关如何执行此操作的信息。

关于此任务

在此示例中，所有名称都以大写形式输入，并且由于 IBM MQ 名称区分大小写，因此您还必须以大写形式输入所有名称。

要使用 IBM MQ Explorer 创建和启动队列管理器，请完成以下步骤。

过程

1. 以管理员身份启动 IBM MQ Explorer 。
2. 在 **Navigator** 视图中，右键单击 **队列管理器** 文件夹，然后单击 **新建> 队列管理器**。 将启动 " **创建队列管理器** " 向导。
3. 在 **队列管理器名称** 字段中，输入 QM1。
4. 选中 Make this the default queue manager 复选框。
5. 在 **死信队列** 字段中，输入 SYSTEM.DEAD.LETTER.QUEUE。

   这是创建队列管理器时自动创建的死信队列的名称。
6. 将其他字段留空，然后单击 **完成**，如果该按钮处于禁用状态，请单击 **下一步**。

   如果端口号与现有队列管理器 (例如，作为缺省配置的一部分创建的队列管理器) 冲突，那么将禁用 **完成** 按钮。 您必须继续完成向导以更改缺省端口号。
7. 如果单击了 **下一步**，请继续接受缺省值，并在每个页面上单击 **下一步** ，直到到达向导的最终页面 (当 **完成** 按钮变为可用时)。 更改指定的端口号 (例如，更改为 1415) ，然后单击 **完成**。

   创建并启动队列管理器时， IBM MQ 将显示 " **创建队列管理器** " 对话框窗口。

下一步做什么

要创建队列，请参阅 第 9 页的『创建名为 LQ1 的队列』。

相关任务

在 Multiplatforms 版上创建和管理队列管理器

创建名为 *LQ1* 的队列

使用 IBM MQ Explorer 创建队列。 队列是用于存储消息的数据结构，并且是 IBM MQ 队列管理器对象。

关于此任务

在此任务中，您可以使用 IBM MQ Explorer 创建 IBM MQ 对象。

要使用 IBM MQ Explorer 创建和启动队列，请完成以下步骤。

过程

1. 在“**导航器**”视图中，展开**队列管理器**文件夹。
2. 展开队列管理器 **QM1**。
3. 右键单击 **队列** 文件夹，然后单击 **新建> 本地队列 ... 新建本地队列** 向导将启动。
4. 在 **名称** 字段中，输入 LQ1。
5. 单击**完成**。

   新队列 LQ1 将显示在 " **内容** " 视图中。 如果该队列未显示在 " **内容** " 视图中，请单击 **内容** 视图顶部的 **刷新** 按钮。

下一步做什么

您已准备好将消息放入队列。 要将消息放入队列中，请参阅 第 9 页的『将消息放入队列 LQ1』。

将消息放入队列 *LQ1*

使用 IBM MQ Explorer 将消息放入队列 LQ1 。

关于此任务

此任务假定您已创建名为 QM1 的队列管理器 (如 第 12 页的『创建名为 QM1 的队列管理器』 中所述) 和名为 LQ1 的队列 (如 第 9 页的『创建名为 LQ1 的队列』 中所述)。

要使用 IBM MQ Explorer 将消息放入队列，请完成以下步骤。

过程

1. 在**"导航器"**视图中，展开**队列管理器**文件夹。
2. 展开您创建的队列管理器 QM1。
3. 单击**队列**文件夹。 队列管理器的队列在 "内容" 视图中列出。
4. 在 "内容" 视图中，右键单击本地队列 LQ1，然后单击 **放置测试消息 ...**。
   此时会打开**"放入测试消息"**对话框。
5. 在 **消息数据** 字段中，输入一些文本，例如 Hello World，然后单击 **放入消息**。
   系统会清空**消息数据**字段，并将消息放入队列。
6. 单击**关闭**。
   在 "内容" 视图中，请注意 LQ1 **当前队列深度** 值现在为 1。 如果 **当前队列深度** 列不可见，那么您可能需要滚动到 " **内容视图**" 的右侧。

下一步做什么

要从队列获取消息，请参阅 第 10 页的『从队列 LQ1 获取消息』。

从队列 *LQ1* 获取消息

使用 IBM MQ Explorer 从队列 LQ1 获取消息。

关于此任务

此任务假定您已放入消息 QM1 ，如 第 9 页的『将消息放入队列 LQ1』中所述。

要使用 IBM MQ Explorer 从队列获取消息，请完成以下步骤。

过程

1. 在 " **Navigator** " 视图中，展开 **队列管理器** 文件夹，然后展开 QM1。
2. 单击**队列**文件夹。
3. 在 " **内容** " 视图中，右键单击本地队列 LQ1，然后单击 **浏览消息 ...**。 将打开 **消息浏览器** 以显示当前位于 QM1 上的消息的列表。
4. 双击最后一条消息以打开 "属性" 对话框。

   在属性对话框的**数据**页上，**消息数据**字段以人类可读的格式显示了消息的内容。

下一步做什么

遵循后续场景中的指示信息以探索进一步的 IBM MQ 功能。

要了解有关编写排队应用程序，连接到队列管理器以及与队列管理器断开连接，发布/预订以及打开和关闭对象的信息，请参阅 编写用于排队的过程应用程序。

使用命令行界面安装和配置

使用命令行在 Windows 上安装 IBM MQ 以执行静默安装并设置环境变量。 验证安装后，创建队列管理器和队列，然后尝试将消息放入队列并从队列中获取消息。

使用静默安装进行安装

使用命令行在 Windows 上安装 IBM MQ ，以执行静默安装，并确认已正确设置安装环境。

开始之前

在启动此任务之前，请完成以下检查:

- 安装时必须具有本地管理员权限。 通过 Windows 工具定义此权限。
- 确保机器名不包含任何空格。

- 确保您有足够的磁盘空间。 有关更多信息，请参阅 Multiplatforms 版上的磁盘空间需求。
- 确定是否需要为任何 IBM MQ 用户定义 Windows 域用户标识。

在安装 IBM MQ 之前，请检查系统是否满足硬件和软件需求。 有关所有受支持平台上的硬件和软件需求的最新详细信息，请参阅 系统要求 IBM MQ。

关于此任务

此方案假定您是首次在机器上安装 IBM MQ，并且您正在使用缺省位置。 缺省情况下， IBM MQ 9.0 程序文件的位置为 `C:\Program Files\IBM\MQ`，数据和日志文件位置为 `C:\ProgramData\ IBM \ MQ`。

注: 如果您的机器上有任何先前安装的 IBM MQ，那么程序和数据文件的缺省位置可能会更改。 有关更多信息，请参阅 程序和数据目录位置。 如果您先前已完成此方案，并且想要使用缺省位置通过单个全新安装重复此方案，请先除去先前的安装，然后再再次启动此方案。 要从机器卸载 IBM MQ 的现有实例，请参阅 第 15 页的『卸载 IBM MQ』。

Windows 上的 IBM MQ 使用 MSI 技术来安装软件。 有关使用 MSI 技术进行安装的更多信息，请参阅 使用 msiexec 进行高级安装。

要使用命令行安装 IBM MQ，必须指定以下参数:

- `/i "`*MQ_INSTALLATION_MEDIA*`\MSI\IBM MQ.msi"`，其中 *MQ_INSTALLATION_MEDIA* 是 `IBM MQ.msi` 文件的位置。 此参数指定 .msi 文件的位置。
- `/l*v` *USER_LOGFILE_LOCATION*`\install.log` 其中 *USER_LOGFILE_LOCATION* 是要将安装日志写入的位置。

  注: 要在其中创建 `install.log` 的文件夹必须存在，然后才能运行该命令。
- `/q[n|b|r|f]` `/q` 必须与 `n`， `b`， `r` 或 `f` 之一配对。 在命令提示符处运行 **msiexec** 命令将打开帮助文件，该文件显示正确的用法。
- `USEINI="`*RESPONSE_FILE*`"` 其中 *RESPONSE_FILE* 是要由静默安装使用的响应文件的名称和位置。 此方案使用样本 `Response.ini` 文件，该文件包含在 IBM MQ 安装介质中。
- `TRANSFORMS="`*TRANSFORM_FILE*`"`，其中 *TRANSFORM_FILE* 是要应用于安装的变换文件的名称。 此方案使用美国英语变换 `1033.mst`。
- `AGREETOLICENSE="YES"` 必须包含此参数，否则安装无法完成。
- `ADDLOCAL="Server"` 此参数列出要安装的组件。

过程

1. 使用命令行来执行静默安装。
   a) 要从提升的命令提示符调用静默安装，请单击 **Windows 任务栏** 上的 **启动按钮**，并在 **搜索程序和文件** 字段中输入 `cmd`。 右键单击 **cmd.exe** 程序，然后选择 **以管理员身份运行**。
   b) 在 Windows 命令提示符中，输入以下命令:

   注: 此命令显示在此处的多行上，但必须在一行上输入此命令。

   ```
   msiexec /i "MQ_INSTALLATION_MEDIA\MSI\IBM MQ.msi"
   /l*v c:\wmqinslogs\install.log
   /q USEINI="MQ_INSTALLATION_MEDIA\Response.ini"
   TRANSFORMS="1033.mst"
   AGREETOLICENSE="yes"
   ADDLOCAL="Server"
   ```

   其中 *MQ_INSTALLATION_MEDIA* 是 IBM MQ 安装介质的路径。

   注: 要在其中创建 `install.log` 的文件夹必须存在，然后才能运行该命令。

   输入命令后，命令行将返回提示符。
   c) 要查看安装进度，请打开指定的日志文件。 如果安装成功完成，那么您将看到消息 `Product: IBM MQ (Installation1) -- Installation operation completed successfully.` 从日志文件底部向上显示两段。

d) 安装完成后，服务将启动， IBM MQ 图标将显示在系统托盘中。

您已安装 IBM MQ，并且已启动 IBM MQ 服务。

2. 使用 **setmqenv** 命令为安装设置环境变量。

a) 在命令行中输入以下命令:

```
"MQ_INSTALLATION_PATH/bin/setmqenv" -s
```

其中 *MQ_INSTALLATION_PATH* 是指 IBM MQ 的安装位置。 确保将 bin 文件夹中 **setmqenv** 的路径括在引号中，以防止提示返回错误。

注: 如果使用了缺省位置，那么安装的路径将为 C:\Program Files\IBM\MQ。

b) 通过输入以下命令来检查是否正确设置了环境:

```
dspmqver
```

如果该命令成功完成，并且返回了期望的版本号和安装名称，那么将正确设置环境。 消息应包含以下行:

```
Version: n.n.n.n
```

其中， *n.n.n.n* 是版本号，如果未指定非缺省安装名称，那么该行:

```
InstName: Installation1
```

您已使用静默安装成功安装 IBM MQ 。

结果

您已执行 IBM MQ 静默安装，并确认您的环境已正确设置。

下一步做什么

- 您可以运行 Prepare IBM MQ Wizard。
- 请遵循第 12 页的『创建名为 QM1 的队列管理器』中的指示信息。

如果在安装期间迁到任何问题，请在 **msiexec** 命令中指定的位置检查安装日志，在此场景中，日志文件的位置为: c:\wmqinslogs\install.log。 请执行日志中指定的任何操作，然后重新运行安装。 您还可以检查随命令传递的参数，以屏蔽确保包含所有必需参数。

相关任务

使用 msiexec 安装服务器

将变换与 msiexec 配合使用

安装 IBM MQ -概述

创建名为 *QM1* 的队列管理器

使用命令行界面创建名为 QM1 的队列管理器。 队列管理器是 IBM MQ 消息传递网络中的主要组件。

开始之前

必须已安装 IBM MQ 。 如果未执行此操作，请参阅第 10 页的『使用静默安装进行安装』以获取有关如何执行此操作的信息。

关于此任务

在此示例中，所有名称都以大写形式输入，并且由于 IBM MQ 名称区分大小写，因此您还必须以大写形式输入所有名称。

过程

1. 以管理员身份打开命令提示符。
2. 通过输入以下命令，创建名为 QM1 的队列管理器:

```
crtmqm QM1
```

当系统创建队列管理器时，将显示以下输出:

```
C:\>crtmqm QM1
IBM MQ queue manager created.
Creating or replacing default objects for QM1.
Default objects statistics : 61 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

队列管理器已创建，并且已停止。 必须先启动队列管理器，然后才能对其进行管理，然后才能从其队列中读写消息。

3. 通过输入以下命令来启动队列管理器:

```
strmqm QM1
```

当队列管理器成功启动时，将显示以下输出:

```
C:\>strmqm QM1
IBM MQ queue manager 'QM1' starting.
5 log records accessed on queue manager 'QM1' during the log replay phase.
Log replay for queue manager 'QM1' complete.
Transaction manager state recovered for queue manager 'QM1'.
IBM MQ queue manager 'QM1' started.
```

队列管理器已启动。

下一步做什么

要创建队列，请参阅 第 13 页的『创建名为 LQ1 的队列』。

相关任务

在 Multiplatforms 版上创建和管理队列管理器

创建名为 *LQ1* 的队列

使用命令行界面创建队列。 队列是用于存储消息的数据结构，并且是 IBM MQ 队列管理器对象。

关于此任务

有三种方法可创建 IBM MQ 对象:

- 命令行。
- IBM MQ Explorer.
- 使用可编程接口。

在此任务中，您可以使用命令行来创建 IBM MQ 对象。

命令行界面具有名为 IBM MQ Script Commands (MQSC) 的脚本语言。 脚本编制工具 **runmqsc** 用于对队列管理器运行脚本。 要使用命令行界面创建和启动队列，请完成以下步骤。

过程

1. 通过输入以下命令来启动脚本编制工具:

```
runmqsc QM1
```

当脚本编制工具启动时，将显示以下输出:

```
C:\>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2025. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
```

该工具已准备好接受 MQSC 命令。

2. 通过输入以下 MQSC 命令来创建名为 LQ1 的本地队列:

```
define qlocal(LQ1)
```

创建队列时，将显示以下输出:

```
define qlocal(LQ1)
2 : define qlocal(LQ1)
AMQ8006: IBM MQ queue created.
```

3. 通过输入以下 MQSC 命令来停止脚本编制工具:

```
end
```

脚本编制工具结束时，将显示以下输出:

```
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.

C:\>
```

下一步做什么

您已准备好将消息放入队列。 要将消息放入队列中，请参阅 第 14 页的『将消息放入队列 LQ1』。

将消息放入队列 *LQ1*

使用命令行界面将消息放入队列 LQ1 。

关于此任务

IBM MQ 随附名为 **amqsput** 的样本应用程序。 此应用程序将消息放入预定义队列。

要使用命令行界面将消息放入队列，请完成以下步骤。

过程

1. 通过输入以下命令，使用 **amqsput** 样本应用程序将消息放入队列 LQ1:

```
amqsput LQ1 QM1
```

样本应用程序启动时，将显示以下输出:

```
C:\>amqsput LQ1 QM1
Sample AMQSPUT0 start
target queue is LQ1
```

2. 输入 Hello World 并按 Enter 键。 您在名为 QM1 的队列管理器管理的队列 LQ1 上放置了一条包含文本 "Hello World" 的消息。
3. 要结束 **amqsput**，请按 **Enter** 键。 将显示以下输出:

```
C:\>amqsput LQ1 QM1
Sample AMQSPUT0 start
target queue is LQ1
Hello World

Sample AMQSPUT0 end
```

下一步做什么

要从队列获取消息，请参阅 第 15 页的『从队列 LQ1 获取消息』。

从队列 *LQ1* 获取消息

使用命令行界面从队列 LQ1 获取消息。

关于此任务

IBM MQ 随附名为 **amqsget** 的样本应用程序。 此应用程序从队列中读取消息。

要使用命令行界面从队列获取消息，请完成以下步骤。

过程

通过输入以下命令，使用 **amqsget** 样本应用程序来读取队列 LQ1 上的消息:

```
amqsget LQ1 QM1
```

样本应用程序启动时，将显示以下输出:

```
C:\>amqsget LQ1 QM1
Sample AMQSGET0 start
message <Hello World>
no more messages
Sample AMQSGET0 end
```

amqsget 应用程序在读取消息后 30 秒结束。

下一步做什么

遵循后续场景中的指示信息以探索进一步的 IBM MQ 功能。

要了解有关编写排队应用程序，连接到队列管理器以及与队列管理器断开连接，发布/预订以及打开和关闭对象的信息，请参阅 编写用于排队的过程应用程序。

卸载 IBM MQ

停止，然后卸载 IBM MQ，包括除去任何队列管理器及其对象。 在此任务结束时，您已准备好重新安装 IBM MQ。

关于此任务

本任务描述了使用下载的安装映像在 Windows 上卸载 IBM MQ 的步骤。

"入门" 方案将指导您使用启动板或命令行来完成安装 IBM MQ 的选项。 虽然可以有多个 IBM MQ 安装，但此方案基于单个服务器上的新安装。 因此，如果要重复此场景，或者尝试其他安装方法，那么必须首先卸载现有 IBM MQ 组件 (包括任何现有队列管理器及其对象)，以便可以重新开始执行全新安装。

您可能还需要卸载，以便可以对本部分中的某些其他方案执行全新安装。

过程

1. 停止 IBM MQ 服务。
   a) 右键单击系统托盘中的 **IBM MQ** 图标，然后单击 **停止 IBM MQ** 以停止 IBM MQ 服务。
   将显示包含以下消息的对话框:

   ```
   关闭 IBM MQ 安装 "Installation1" 将终止所有正在运行的队列管理器和
   该安装的 IBM MQ 进程 (Microsoft Failover Cluster 控制下的进程除外)。
   确定要继续吗？
   ```

   b) 单击 **是**，然后等待 IBM MQ 停止。
   c) 当 IBM MQ 停止时，右键单击系统托盘中的 **IBM MQ** 图标，然后单击 **退出**

2. 开始卸载过程。

   下载包含安装映像的压缩文件，然后将其解压缩到临时目录中。 浏览至该目录，然后双击 `setup.exe`。

   此时将显示 " IBM MQ **安装启动板** " 窗口。

3. 除去 IBM MQ。

   a) 单击 **IBM MQ 安装**。

   b) 单击 **启动 IBM MQ 安装程序** ，然后单击 **下一步** ，直到显示带有欢迎消息的 IBM MQ **程序维护窗格** 。

   如果未显示此窗格，那么当前未安装 IBM MQ for Windows 。

   c) 单击 **维护或升级现有实例**。 选择 **Installation1** 以将其除去。 单击 **下一步** ，然后在 " **程序维护**" 窗格中单击 **除去**，然后单击 **下一步**。

   此时将显示 "除去服务器功能部件" 窗格。

   d) 选择 **除去**: 除去现有队列管理器及其对象。

   单击**下一步**。

   将显示 "除去 IBM MQ " 窗格，其中包含要除去的安装的摘要。

   e) 单击 **除去** 以继续。

   如果显示一条消息指出找到了锁定的文件，请确保没有 IBM MQ 程序正在运行; 请参阅 在 Windows 系统上卸载 IBM MQ。

   卸载 IBM MQ 时，将显示一条指示完成的消息。

   f) 单击**完成**。

   您已成功卸载 IBM MQ。

相关任务

在 Windows 系统上卸载 IBM MQ

后续操作

完成 " IBM MQ 入门" 方案后的下一步操作。

Windows AIX Linux 有关帮助您安装和升级的教程，请参阅 用于在 AIX， Linux®和 Windows 上安装和升级 IBM MQ 的教程集合。 教程涵盖:

- 为 IBM MQ 准备主机。
- 下载 IBM MQ 代码。
- 安装和卸载 IBM MQ 代码以及应用修订包。
- 从 IBM MQ 的一个版本升级到另一个版本，并将队列管理器从一个主机移动到另一个主机。

您可以在 IBM MQ 产品文档中查看其他主题。 您可能想要查看以下部分:

- 管理 IBM MQ

  IBM MQ 提供了您可以使用的控制命令。 在此场景中使用以下两个命令: **crtmqm** 和 **strmqm**。 此部分还提供了有关消息排队的良好概述。

- 使用 MQSC 命令管理 IBM MQ

  在此场景中，您使用 `define qlocal('LQ1')` 命令来定义名为 LQ1 的本地队列。 此命令是 MQSC 命令。 IBM MQ 系统管理员使用这些命令来管理其队列管理器。 本部分介绍了这些命令，并说明了如何使用这些命令。 在 MQSC 命令 参考部分中按字母顺序详细描述了这些命令。

- 配置队列管理器集群

  本节描述如何在称为集群的虚拟组中组织，使用和管理队列管理器。 集群可确保集群中的每个队列管理器都知道同一集群中的所有其他队列管理器。 集群还使复杂队列管理器网络的管理更简单。

点到点方案

在点到点拓扑中连接两个 IBM MQ 队列管理器以启用分布式排队。

关于此任务

创建两个队列管理器以及相应的队列和通道，以创建单向点到点消息传递基础结构。 在单独的主机上创建队列管理器以允许通过网络进行通信。 作为方案的扩展，将传输层安全性添加到通道以实现数据的安全通信。

规划解决方案

点到点消息传递是 IBM MQ 中最简单的消息传递形式。 在点到点消息传递中，发送应用程序必须先知道有关接收应用程序的某些信息，然后才能发送消息。 发送应用程序将需要一种方法来寻址远程队列。 使用点到点消息传递将消息发送到具有样本应用程序的远程队列管理器。

概述: 交付的逻辑拓扑

完成方案后交付的逻辑拓扑。

点到点基础结构允许在不同主机上的队列管理器之间进行一次定向消息传递。 队列管理器 1 (在主机 1 上) 将消息发送到队列管理器 2 (在主机 2 上)。 此场景完成后，交付的拓扑将类似于 图 1。

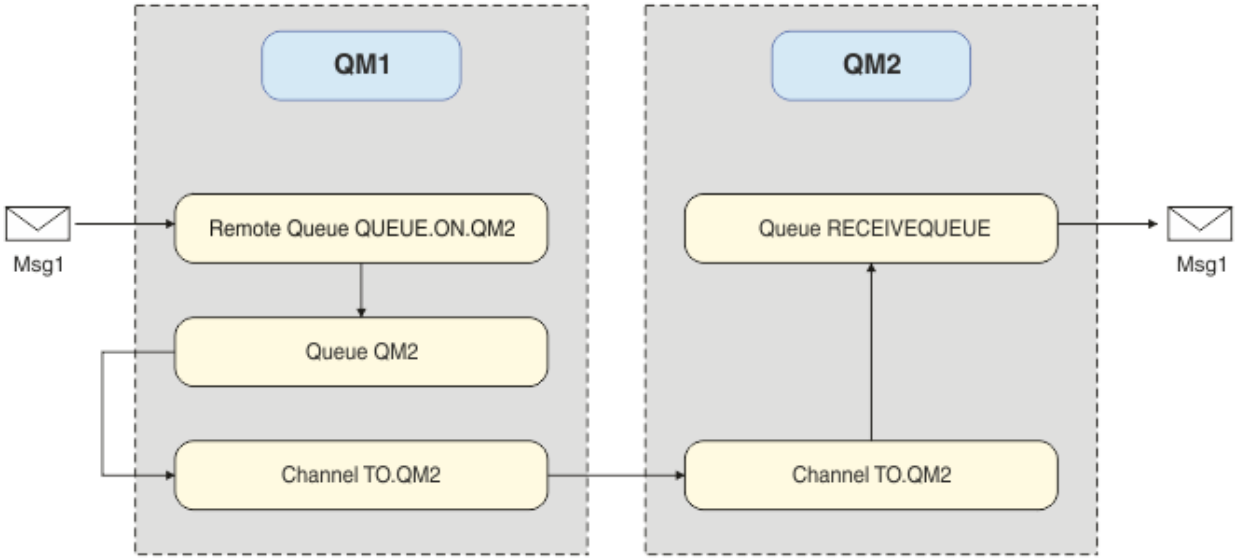


图 *2: QM1* 向 *QM2* 发送消息

基本概念和主要术语

完成点到点方案必须知道的基本概念和关键术语的描述。

基本概念

通过 IBM MQ，应用程序可以针对队列读写消息。 读消息的应用程序和写消息的应用程序是相互独立的。 不要求两个应用程序同时运行。 如果没有可用的读消息应用程序，那么消息将在 IBM MQ 队列上排队，直到应用程序读取该消息。

关键词

下面是关于消息排队的关键术语列表。

有关消息排队的关键术语。

术语	描述
队列管理器	队列管理器负责维护它所拥有的队列，并负责将它接收到的所有消息存储到适当的队列中。
消息	消息是对使用它的应用程序有意义的以字节为单位的字符串。 消息用来将信息从一个应用程序传输到另一个应用程序。 应用程序可以在相同或不同的计算机上运行。
本地队列	本地队列是用于存储消息的数据结构。 该队列可以是常规队列或者传输队列。 直接从队列管理器读取消息的应用程序要读取的消息保留在常规队列中。 正在转移给其他队列管理器的消息保留在传输队列中。
远程队列	远程队列用于将消息发送到另一个队列管理器。
通道	通道用于在队列管理器之间发送和接收消息。
侦听器	侦听器是接受来自其他队列管理器或客户机应用程序的网络请求并启动关联通道的进程。

实现解决方案

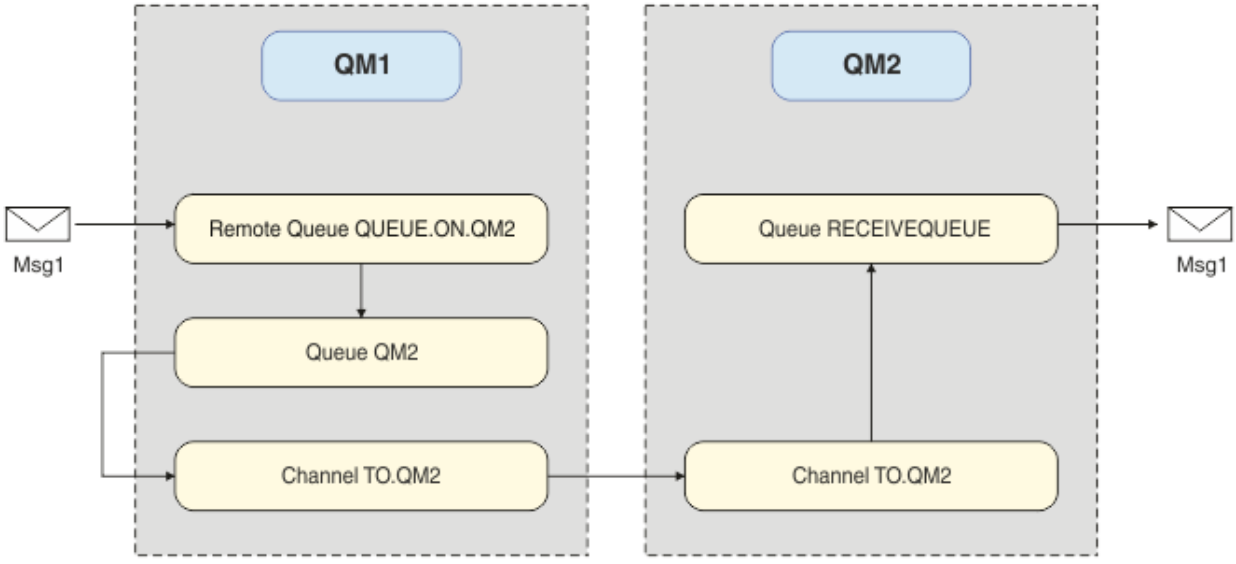
实现方案的解决方案。 在两个不同的主机上创建两个 IBM MQ 队列管理器，源队列管理器用于发送消息，目标队列管理器用于接收消息。

开始之前

此方案的起点是已验证的现有 IBM MQ 安装。 有关安装 IBM MQ 的指示信息，请遵循 在 Windows 上安装 IBM MQ 服务器中的步骤。

关于此任务

使用命令行界面创建两个队列管理器，定义所需的侦听器，队列和通道。 交付的逻辑拓扑显示通过实现解决方案添加的函数。



创建队列管理器

创建 IBM MQ 队列管理器以将消息发送到目标队列管理器。

开始之前

- 必须已安装 IBM MQ 。 有关安装 IBM MQ 的更多信息，请参阅 安装和卸载。

关于此任务
使用命令行界面创建 IBM MQ 队列管理器。

过程
1. 创建名为 QM1 的队列管理器。 在命令行上，输入:

```
crtmqm QM1
```

   将显示以下消息以确认已创建队列管理器:

```
IBM MQ queue manager created.
Creating or replacing default objects for QM1.
Default objects statistics : 61 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

2. 启动队列管理器。 在命令行上，输入:

```
strmqm QM1
```

   将显示以下消息以确认队列管理器已启动:

```
IBM MQ queue manager 'QM1' starting.
5 log records accessed on queue manager 'QM1' during the log replay phase.
Log replay for queue manager 'QM1' complete.
Transaction manager state recovered for queue manager 'QM1'.
IBM MQ queue manager 'QM1' started.
```

结果
将创建并启动 IBM MQ 队列管理器 QM1 。

下一步做什么
要创建要与 QM1 配合使用的队列，请遵循 第 19 页的『创建请求』中的指示信息。

创建请求
创建由 IBM MQ 队列管理器管理的 IBM MQ 队列。

开始之前
您必须具有如 第 18 页的『创建队列管理器』中所述设置的 IBM MQ 队列管理器。

关于此任务
启动 **MQSC** 接口以管理连接到队列管理器的对象。 创建传输队列和远程队列定义。 退出 **MQSC** 接口。

过程
1. 在命令行上，输入:

```
runmqsc QM1
```

   在确认消息后，该工具已准备好接受命令。
2. 创建名为 QM2 的传输队列。 最好为传输队列提供与远程队列管理器相同的名称。 在 MQSC 接口中，输入:

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') USAGE(XMITQ)
```

   将创建传输队列。
3. 创建名为 QUEUE.ON.QM2。 远程队列定义必须引用为远程主机上的本地队列提供的名称。 在 MQSC 接口中，输入:

```
DEFINE QREMOTE(QUEUE.ON.QM2) DESCR('Remote queue for QM2') XMITQ(QM2) RNAME(RECEIVEQUEUE)
RQMNAME(QM2)
```

将创建远程队列定义。

4. 输入 end 以退出 MQSC 接口。

下一步做什么

要创建用于连接到目标队列管理器的发送方通道，请遵循 第 20 页的『创建发送通道』中的指示信息。

创建发送通道

在源队列管理器上创建发送者通道，该通道用于连接目标队列管理器。

开始之前

要创建使用 TLS 的通道，请按照 "第 25 页的『创建通道以使用 TLS』中的说明操作。 如果想在不使用 TLS 安全性的情况下测试解决方案，可以在测试后进行此操作。

关于此任务

启动 "**MQSC** 接口，管理连接到队列管理器的对象，并创建发送者通道。 该通道用于连接名为 QM2 的目标队列管理器。

过程

1. 在命令行中键入

```
runmqsc QM1
```

出现确认信息后，工具就可以接受命令了。

2. 创建名为 TO.QM2 的发送通道。 在 MQSC 界面，键入

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(SDR) CONNAME(' remoteHost ') TRPTYPE(TCP) XMITQ(QM2)
```

注: 变量 *remoteHost* 是目标队列管理器的主机名或 IP 地址。

发送通道已创建。

下一步做什么

要创建分布式队列管理器拓扑结构，请按照 "第 20 页的『创建分布式队列管理器拓扑』中的说明操作。

创建分布式队列管理器拓扑

点到点消息传递是 IBM MQ 中最简单的消息传递形式。 在点到点消息传递中，发送应用程序必须先知道有关接收应用程序的某些信息，然后才能发送消息。 发送应用程序将需要一种方法来寻址远程队列。 使用点到点消息传递将消息发送到具有样本应用程序的第二个队列管理器。

开始之前

您必须已设置源队列管理器，如 第 18 页的『创建队列管理器』中所述。

关于此任务

在远程主机上创建目标队列管理器。 使用样本应用程序来验证源队列管理器与目标队列管理器之间的通信。

创建队列管理器

创建 IBM MQ 队列管理器以从远程队列管理器接收消息。

开始之前

必须已安装 IBM MQ 。 有关安装 IBM MQ 的更多信息，请参阅 在 Windows 上安装 IBM MQ 服务器。

关于此任务
使用命令行界面创建 IBM MQ 队列管理器。

过程
1. 创建名为 QM2 的队列管理器。 在命令行上，输入:

```
crtmqm QM2
```

   此时会显示以下消息:

```
IBM MQ queue manager created.
Creating or replacing default objects for QM2.
Default objects statistics : 61 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

2. 启动队列管理器。 在命令行上，输入:

```
strmqm QM2
```

   将显示以下消息以确认队列管理器已启动:

```
IBM MQ queue manager 'QM2' starting.
5 log records accessed on queue manager 'QM2' during the log replay phase.
Log replay for queue manager 'QM2' complete.
Transaction manager state recovered for queue manager 'QM2'.
IBM MQ queue manager 'QM2' started.
```

结果
将创建并启动 IBM MQ 队列管理器 QM2 。

下一步做什么
要创建要与 QM2 配合使用的队列，请遵循 第 21 页的『创建队列』中的指示信息。

创建队列
创建用于在目标队列管理器上接收消息的本地队列，以及用于接受入站通道连接的侦听器。

关于此任务
启动 **runmqsc** 脚本编制工具后，可以使用 MQSC 命令来创建本地队列和侦听器。

过程
1. 通过输入以下命令来启动脚本编制工具:

```
runmqsc QM2
```

   将显示一条消息以确认工具已启动。
2. 创建名为 RECEIVEQUEUE 的本地队列。 队列必须与源队列管理器上的远程队列定义中引用的名称相同。 在 MQSC 接口中，输入:

```
DEFINE QLOCAL(RECEIVEQUEUE) DESCR('Receiving queue')
```

   已创建本地队列。
3. 创建名为 LISTENER1 的侦听器。 在 MQSC 接口中，输入:

```
DEFINE LISTENER(LISTENER1) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
```

   注: 端口 1414 是 IBM MQ 的缺省端口。 如果选择其他端口号，那么必须将其添加到发送队列管理器上发送方通道的 CONNAME 。
4. 启动侦听器，以便它可以接受入站连接。 在 MQSC 接口中，输入:

```
START LISTENER(LISTENER1)
```

注: 由于侦听器是使用选项 CONTROL （QMGR)创建的，因此下次启动队列管理器时，该侦听器也将自动启动。

5. 输入 end 以退出 **MQSC** 接口。

下一步做什么

要创建接收方通道以在源队列管理器与目标队列管理器之间创建连接，请遵循 第 22 页的『创建接收通道』 中的指示信息。

创建接收通道

为目标队列管理器创建接收器通道，以实现源队列管理器和目标队列管理器之间的通信。

开始之前

要创建使用 TLS 的通道，请按照 "第 25 页的『创建通道以使用 TLS』 中的说明操作。 如果想在不使用 TLS 安全性的情况下测试解决方案，可以在测试后进行此操作。

关于此任务

使用 "**MQSC** 接口创建一个由 QM2 管理的接收通道。

过程

1. 在命令行中键入

```
runmqsc QM2
```

出现确认信息后，工具就可以接受命令了。

2. 创建名为 TO.QM2 的接收通道。 通道名称必须与源队列管理器上的发送者通道相同。 在 MQSC 界面，键入

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP)
```

创建接收通道。

下一步做什么

要在源队列管理器上启动发送者通道，进而在目标队列管理器上启动接收者通道，请按照 "第 22 页的『启动发送方通道』 中的指令操作。

启动发送方通道

在源队列管理器上启动发送方通道，在目标队列管理器上启动接收方通道。 可以将消息从源队列管理器发送到目标队列管理器。

关于此任务

启动 **MQSC** 接口以管理连接到队列管理器的对象。 启动发送方通道以连接到目标队列管理器，从而启用通信。 当启动源通道时，接收方通道会自动启动。

过程

1. 在命令行上，输入:

```
runmqsc QM1
```

在确认消息后，该工具已准备好接受命令。

2. 在源队列管理器上启动发送方通道。 在 MQSC 接口中，输入:

```
START CHANNEL(TO.QM2)
```

发送方通道启动，目标队列管理器上的接收方通道也启动。

3. 请检查通道是否正在运行。 在 MQSC 接口中，输入：

```
DISPLAY CHSTATUS(TO.QM2)
```

如果通道正在运行，那么您将看到它报告 STATUS （RUNNING）。 如果它在 STATUS 中报告任何其他值，请检查 错误日志。

下一步做什么

要验证源队列管理器是否可以向目标队列管理器发送消息，请遵循 第 23 页的『验证解决方案』 中的指示信息。

验证解决方案

验证源队列管理器是否能将信息放入远程队列。 验证目标队列管理器能否从队列中获取报文。

关于此任务

使用示例应用程序 "**amqsput** 和 "**amqsget** 验证解决方案。

过程

1. 从源队列管理器向目标队列管理器 QM2 发送信息。
   a) 在命令行界面键入

```
amqsput QUEUE.ON.QM2 QM1
```

   您必须使用远程队列定义的名称将信息发送到目标队列管理器。
   此时会显示以下消息：

```
Sample AMQSPUT0 start
target queue is QUEUE.ON.QM2
```

   b) 输入 "Hello world.，按两次 Enter。
2. 获取目标队列管理器上的信息。
   a) 在命令行界面键入

```
amqsget RECEIVEQUEUE QM2
```

   此时会显示以下消息：

```
Sample AMQSGET0 start
message <Hello world.>
no more messages
Sample AMQSGET0 end
```

结果

目标队列管理器收到了源队列管理器发来的信息，验证了点对点通信已经实现。

下一步做什么

如果要为解决方案添加安全性，请按照 "第 23 页的『保护点到点拓扑』 中的说明操作。

保护点到点拓扑

保护点到点拓扑，以便可以在生产环境中传输消息。

关于此任务
保护源和目标队列管理器对象，以便授予正确的访问级别。 定义哪些用户组有权访问队列和队列管理器。 使用数字签名证书通过传输层安全性 (TLS) 进行连接来保护网络连接。

保护源队列管理器对象
为源队列管理器上的对象设置授权值。

关于此任务
使用 "**setmqaut** 命令向运行应用程序的用户组授予权限。

过程
1. 要授予指定用户组连接队列管理器的权限，请在命令行界面上键入

```
setmqaut -m QM1 -t qmgr -g userGroup +connect
```

2. 要向指定用户组授予远程队列定义的 *put* 权限，请在命令行界面上键入

```
setmqaut -m QM1 -t q -n "QUEUE.ON.QM2" -g userGroup +put
```

保护目标队列管理器对象
为目标队列管理器上的对象设置授权值。

关于此任务
使用 "**setmqaut** 命令向运行应用程序的用户组授予权限。

过程
1. 要授予指定用户组连接队列管理器的权限，请在命令行界面上键入

```
setmqaut -m QM2 -t qmgr -g userGroup +connect
```

2. 要在远程队列定义中授予指定用户组获取授权，请在命令行界面中键入

```
setmqaut -m QM2 -t q -n "RECEIVEQUEUE" -g userGroup +get
```

保护网络
保护源队列管理器与远程队列管理器之间的网络连接。

关于此任务
使用签名证书来验证源队列管理器和远程队列管理器的真实性。 使用 TLS 网络传输消息以加密消息。

准备队列管理器以使用 *TLS*
IBM MQ 队列管理器的密钥存储库用于存储队列管理器的个人证书和公共认证中心 (CA) 证书。 来自 IBM MQ 队列管理器的个人证书请求必须由 CA 签署，公用证书由其他实体用于认证 IBM MQ 队列管理器。

开始之前
您必须在文件中具有公用认证中心证书。

关于此任务
创建 IBM MQ 队列管理器的密钥存储库，导入认证中心的签署者证书并创建队列管理器的个人证书请求。

过程

1. 为名为 key.kdb 的队列管理器创建 CMS 密钥存储库文件。 浏览至 Qmgrs\QM1\ssl 目录，并在命令行上输入:

   ```
   runmqckm -keydb -create -db key.kdb -pw passw0rd -type cms -stash
   ```

   注: 对于此简单示例，我们使用了密码 passw0rd。 您可能希望选择不同的密码，并更改以下每个命令以改为使用您自己的密码。
2. 在命令行上将文件中包含的 CA 证书添加到密钥存储库中，请输入:

   ```
   runmqckm -cert -add -file CA-certificate-file -db key.kdb -pw passw0rd -label TrustedCA
   ```
3. 请求将写入名为 QM1req.req 的请求文件的个人证书。

   在命令行上，输入:

   ```
   runmqckm -certreq -create -db key.kdb -pw passw0rd -label ibmwebspheremqqm1
   -dn CN="QM1" -size 1024 -file QM1req.req
   -sig_alg SHA1WithRSA
   ```

   此示例中显示了缺省证书标签名称。 如果您愿意，可以设置自己的名称。 有关详细信息，请参阅 数字证书标签。
4. 将证书请求文件发送到 CA ，他们将发放数字签名证书。 将接收到的已签名证书文件放在要接收到队列管理器的密钥存储库中的合适位置。
5. 将签名的个人证书接收到队列管理器的密钥存储库中。

   ```
   runmqckm -cert -receive -file Signed-certificate-file -db key.kdb -pw passw0rd -format ascii
   ```
6. 对每个队列管理器完成这些步骤，并相应地更改队列管理器名称。

下一步做什么

要通过发送方和接收方通道启用安全通信，请遵循 第 25 页的『创建通道以使用 TLS』中的指示信息。

创建通道以使用 *TLS*

创建使用 TLS 创建连接的新通道。

开始之前

要通过使用 TLS 的通道进行通信，首先必须具有连接的每一端所需的证书。 要创建所需证书，请遵循 第 24 页的『准备队列管理器以使用 TLS』中的指示信息。

关于此任务

使用 MQSC 接口定义设置了 TLS 属性的通道。 即使通过使用 REPLACE 关键字在先前步骤中定义了没有 TLS 的通道，也可以执行此任务。

过程

1. 在命令行上，输入:

   ```
   runmqsc QM1
   ```
2. 在 QM1 上的 MQSC 接口中创建名为 TO.QM2 的发送方通道，请输入:

   ```
   DEFINE CHANNEL(TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP)
   CONNAME('remoteHost') XMITQ(QM2)
   SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
   DESCR('Sender channel using TLS from QM1 to QM2')
   REPLACE
   ```

   注: 变量 *remoteHost* 是目标队列管理器的主机名或 IP 地址。

可以为通道指定 CERTLABL 属性。 如果执行此操作，那么它必须与先前在 第 24 页的『准备队列管理器以使用 TLS』的步骤 3 中运行的 **runmqckm** 命令的 **-label** 参数上的值相匹配。 有关证书标签的更多信息，请参阅 数字证书标签，了解需求。

3. 输入 end 以退出 MQSC 接口。
4. 在命令行上，输入:

```
runmqsc QM2
```

5. 在 QM2 上创建名为 TO.QM2，在 MQSC 接口中，输入:

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) SSLCAUTH(REQUIRED)
DESCR('Receiver channel using TLS from QM1 to QM2')
REPLACE
```

6. 输入 end 以退出 MQSC 接口。

下一步做什么

要验证源队列管理器是否可以使用 TLS 将消息发送到目标队列管理器，请遵循 第 23 页的『验证解决方案』中的指示信息。

V 9.3.0 流式队列

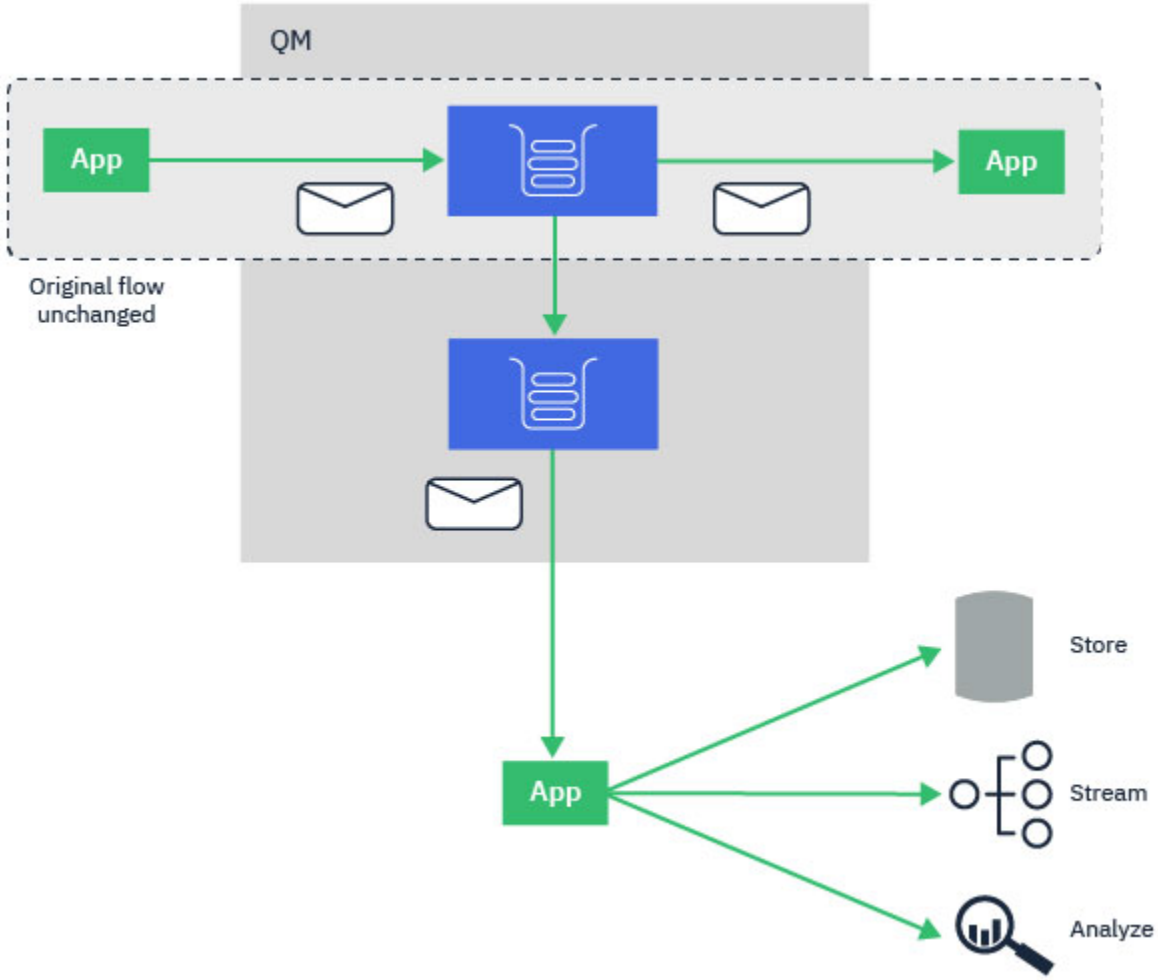
IBM MQ 的流式队列功能允许您配置队列，以将每条消息的近乎相同的副本放置到另一个队列中。

在需要创建消息副本的特定场景中，流式队列可能很有用。 例如:

- 使用 Kafka Connect source connector for IBM MQ 将消息流式传输到 Apache Kafka 。 有关更多信息，请参阅 kafka_connect_mq_source。
- 对通过系统的数据执行分析。
- 存储消息以供稍后恢复。
- 捕获一组要在开发和测试系统中使用的消息。
- 使用来自系统事件队列的 IBM MQ 事件消息，并将其他副本发送到其他队列或主题。

在所有这些场景中，您可以配置流式队列，以确保原始消息不受流式流程的影响。 这可确保核心业务应用程序不会观察到来自流式方法的任何影响。

以下插图对此进行了描述:



相关概念

流式队列安全性

流式队列和 AMS

V 9.3.0 流式队列配置

IBM MQ 的流式队列功能由管理员在各个队列上配置，消息由队列管理器而不是应用程序本身进行流式传输。

这意味着在几乎所有情况下，将消息放入原始队列的应用程序都完全不知道正在进行流式方法。 同样，使用来自原始队列的消息的应用程序不知道已发生消息流。

注: IBM MQ 客户机库的版本不需要升级以使用流式队列，并且流式进程完全不会更改原始消息。

您可以通过以下两种方式之一来配置流式队列:

尽最大努力

在此方式下，队列管理器认为更重要的是，传递原始消息不受传递流消息的影响。

如果可以传递原始消息，但流式消息无法传递，那么原始消息仍将传递到其队列。 此方式最适合那些应用程序，在这些应用程序中，原始业务应用程序保持不受流式流程影响非常重要。

必须复制

在此方式下，队列管理器确保将原始消息和流式消息都成功传递到其队列。

如果由于某种原因，无法将流式消息传递到其队列，例如，由于第二个队列已满，那么原始消息也不会传递到其队列。 放置应用程序接收到错误原因码，必须尝试再次放置消息。

请参阅 如何配置流式队列，以获取有关添加到启用消息流式方法的本地队列和模型队列的其他属性的信息。

流式消息

在大多数情况下，传递到第二个队列的消息的副本是原始消息的副本。 这包括所有消息描述符字段，包括消息标识和相关标识。 流式消息旨在作为原始消息的非常接近的副本，以便更容易找到这些消息，并在必要时将其重放回另一个 IBM MQ 系统中。

有一些消息描述符字段未保留在流式消息上。 在将流式消息放入第二个队列之前，将对其进行以下更改:

- 流消息的到期时间设置为 MQEI_UNLIMITED，而不考虑原始消息的到期时间。 如果已在辅助队列上设置 **CAPEXPRY**，那么将使用其值来设置流式消息的到期时间。
- 如果在原始消息上设置了下列任何报告选项，那么不会在流式消息上启用这些选项。 这是为了确保不会将任何意外的报告消息传递到并非旨在接收这些消息的应用程序:
  - 活动报告
  - 到期报告
  - 异常报告
  - 到达时确认 (COA)
  - 交货确认 (COD)

由于流式消息的几乎相同性质，辅助队列的大部分属性对流式消息的消息描述符字段没有影响。 例如，辅助队列的 **DEFPSIST** 和 **DEFPRTY** 属性对流式消息没有影响。

以下异常适用于流式消息:

- **CAPEXPRY** 属性

  如果已使用 **CAPEXPRY** 属性配置辅助队列，那么会将此到期上限应用于流式消息的到期时间。
- **DEFBIND** 表示集群队列

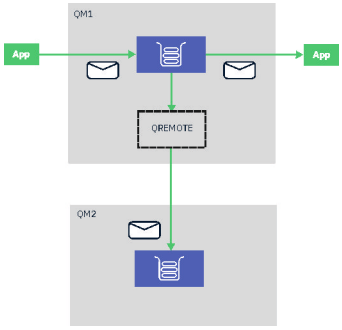
  如果辅助队列是集群队列，那么将使用辅助队列的 **DEFBIND** 属性中设置的绑定选项来放入流式消息。

V 9.3.0 流式传输到远程队列和别名队列

可以将消息流式传输到远程队列和别名队列。 例如，可以使用 STREAMQ (MY.REMOTE.Q) 配置 Q1，其中 MY.REMOTE.Q 是远程队列定义。

流式传输到远程队列

通过将消息从本地队列流式传输到远程队列，可以将重复消息发送到 IBM MQ 网络中另一个队列管理器上的队列，如下图所示:



流式传输到别名队列

通过将消息流式传输到别名队列，可以将重复消息发送到别名队列的目标。 由于别名队列的目标也可以是主题，因此可以将重复消息发送到发布/预订主题。 别名主题的任何订户都将收到重复消息的副本。 通过这种

方式，您可以创建原始消息的多个副本。 但是，发布/预订消息的现有规则将应用于复制的消息。 这意味着发送给订户的消息将与原始消息不同，包括:

- 具有新的消息标识。
- 具有生成的相关标识，具体取决于预订的配置。
- 将 UserIdentifier 字段设置为运行队列管理器的用户，而不是放置消息的用户。
- PutAppl 名称显示队列管理器的名称，而不是放置应用程序的名称。

注意:

1. 无法在远程队列或别名队列本身上配置 **STREAMQ** 属性。 您只能将消息流至这些消息，而不能从这些消息流至这些消息。
2. 如果消息流至队列别名，那么队列别名的目标不能设置其 **STREAMQ** 属性。

V 9.3.0 流式队列限制

在 IBM MQ 中使用流式队列时，不支持某些配置，此处记录了这些配置。

以下列表指定了不受支持的配置:

- 定义相互流式传输的队列链，例如， Q1->Q2， Q2->Q3， Q3->Q4
- 定义流式队列循环，例如 Q1->Q2， Q2->Q1
- 使用提供的目标定义预订，其中该目标定义了 STREAMQ
- 在使用 USAGE (XMITQ) 配置的队列上定义 STREAMQ

  注: STREAMQ 可以是远程队列，但不能在远程队列定义上配置 STREAMQ 属性。
- 修改动态队列的 STREAMQ 属性
- 将 STREAMQ 设置为任何以 SYSTEM. * 开头的值， SYSTEM.DEFAULT.LOCAL.QUEUE
- 在任何名为 SYSTEM. * 的队列上定义 STREAMQ ， 但有以下例外:
  - SYSTEM.DEFAULT.LOCAL.QUEUE
  - SYSTEM.ADMIN.ACCOUNTING.QUEUE
  - SYSTEM.ADMIN.ACTIVITY.QUEUE
  - SYSTEM.ADMIN.CHANNEL.EVENT
  - SYSTEM.ADMIN.COMMAND.EVENT
  - SYSTEM.ADMIN.CONFIG.EVENT
  - SYSTEM.ADMIN.LOGGER.EVENT
  - SYSTEM.ADMIN.PERFM.EVENT
  - SYSTEM.ADMIN.PUBSUB.EVENT
  - SYSTEM.ADMIN.QMGR.EVENT
  - SYSTEM.ADMIN.STATISTICS.QUEUE
  - SYSTEM.DEFAULT.MODEL.QUEUE
  - SYSTEM.JMS.TEMPQ.MODEL
- 将 STREAMQ 设置为模型队列的名称
- V 9.3.1 z/OS 在未应用 APAR PH49686 时，在 IBM MQ 9.3.0 处的共享队列上定义 STREAMQ。 在应用 APAR PH49686 或从 IBM MQ 9.3.1 时，已在 IBM MQ 9.3.0 取消此限制。
- V 9.3.1 z/OS 在未应用 APAR PH49686 时，将 STREAMQ 设置为 IBM MQ 9.3.0 处的共享队列的名称。 在应用 APAR PH49686 或从 IBM MQ 9.3.1 时，已在 IBM MQ 9.3.0 取消此限制。

V 9.3.0 流队列和事务

流式队列功能允许将消息放入一个队列，以复制到另一个队列。 在大多数情况下，这两条消息将放在工作单元下各自的队列中。

如果原始消息是使用 MQPMO_SYNCPOINT 放入的，那么重复消息将放入针对原始放入启动的同一工作单元下的流队列中。

如果将原始项与 MQPMO_NO_SYNCPOINT 一起放置，那么即使原始项未请求工作单元，也将启动该工作单元。 这样做有两个原因:

1. 它确保如果无法传递原始消息，那么不会传递重复消息。 如果还传递了原始消息，那么流式队列功能仅将消息传递到流式队列。
2. 通过在工作单元内执行这两个操作，可以提高性能

在工作单元中未传递消息的唯一时间是原始 MQPUT 与 MQPMO_NO_SYNCPOINT 非持久时，并且队列的 **STRMQOS** 属性设置为 BESTEF (尽最大努力)。

注意:

1. 附加放入流队列的数据不会计入 MAXUMSGS 限制。
2. 对于使用 STRMQOS (BESTEF) 配置的队列，未能传递重复消息不会导致回滚工作单元。

V 9.3.0 在集群队列之间进行流式传输

可以将消息从本地队列流至集群队列，并将消息从集群队列实例流至本地队列。

流式传输到集群队列

如果您具有在其中传递原始消息的本地队列，并且希望将每条消息的副本流式传输到集群队列的一个或多个实例，那么这很有用。 这可能是为了对重复消息的处理进行工作负载均衡，也可能只是为了将重复消息流式传输到集群中其他位置的另一个队列。

将消息流式传输到集群队列时，将使用集群工作负载均衡算法来分发消息。 根据集群队列的 DEFBIND 属性选择集群队列实例。

例如，如果使用 DEFBIND (OPEN) 配置集群队列，那么将在打开原始队列时选择集群队列的实例。 所有重复消息都将转至同一集群队列实例，直到应用程序重新打开原始队列为止。

如果使用 DEFBIND (NOTFIXED) 配置集群队列，那么将为每个 MQPUT 操作选择集群队列的实例。

注: 您应该为 DEFBIND 属性配置具有相同值的所有集群队列实例。

从集群队列流式传输

如果您已经将消息发送到集群队列的多个实例，并且希望将每条消息的副本作为集群队列实例传递到同一队列管理器上的流式队列，那么这很有用。

将原始消息传递到其中一个集群队列实例时，集群接收方通道会将重复消息传递到流队列。

V 9.3.1 Multi 使用流式队列来存储消息历史记录

您可以使用流式队列在有限的时间段内保留消息历史记录，并通过在消息流至的队列上配置 CAPEXPRY 属性来实现此目的。

介绍

当消息从一个队列流向另一个队列时，消息上设置的任何到期值都会重置为重复副本的 MQEI_UNLIMITED 值。 缺省情况下，如果没有应用程序使用消息，那么这将导致在要流式传输到的队列上稳步构建消息。

在此场景中，您希望保留消息的副本一段有限的时间，以便能够访问这些消息。 例如，如果消费应用程序意外除去了原始消息。

不可能无限期地保留每条消息的副本，为了防止流至的队列变满，有两个选项:

- 运行应用程序以经常除去消息
- 配置具有到期时间的消息，从而导致 IBM MQ 在一段时间后将其除去。

第二个选项可以方便得多，因为它不需要您运行和维护应用程序，只是为了防止队列填满。

配置 CAPEXPRY

强制缩短到期时间 主题描述了如何在队列上配置 CAPEXPRY。 对于此场景，必须在要将消息流式传输到的队列上设置 CAPEXPRY 属性。

注: 您不需要更改从中传送消息的队列上 CAPEXPRY 属性的值。

请为重复消息选择合适的到期时间，同时考虑以下注意事项:

1. 您可能需要访问消息的时间
2. 根据将消息放入原始队列的速率，队列的 MAXDEPTH 属性需要是什么
3. 存储重复消息所需的存储量。

   这可能需要考虑队列管理器文件系统大小以及队列的 MAXFSIZE 属性。

访问重复消息 (如果需要)

如果应该发生需要您访问和恢复部分或全部重复消息的问题，请使用 **dmpmqmsg** 命令来帮助访问这些消息。

dmpmqmsg 具有以下选项:

- 从队列中读取消息并将副本写入文件以稍后访问
- 从文件中读取消息并将其写回队列以供应用程序使用

一旦 **dmpmqmsg** 将消息头写入文件，就可以对其进行编辑。 例如，通过将文件中每条消息的 EXP 值更改为 *-1*，可以将消息的到期时间重置为 MQEI_UNLIMITED ，然后 **dmqmqmsg** 将这些消息放回队列以进行处理。

性能注意事项

将重复消息流式传输到另一个队列并在不再需要这些消息时将其到期的成本很小。 但是，与手动将副本放入第二个队列并使应用程序在一段时间后以破坏性方式将其除去相比，成本要小得多。 流式队列性能报告 提供有关此方案的性能的更多信息。

相关概念

流式队列安全性

流式队列和 AMS

发布/预订方案

演示发布/预订集群和发布/预订层次结构的使用的两组方案。

以下子主题描述了可用的发布/预订方案:

方案: 创建发布/预订集群

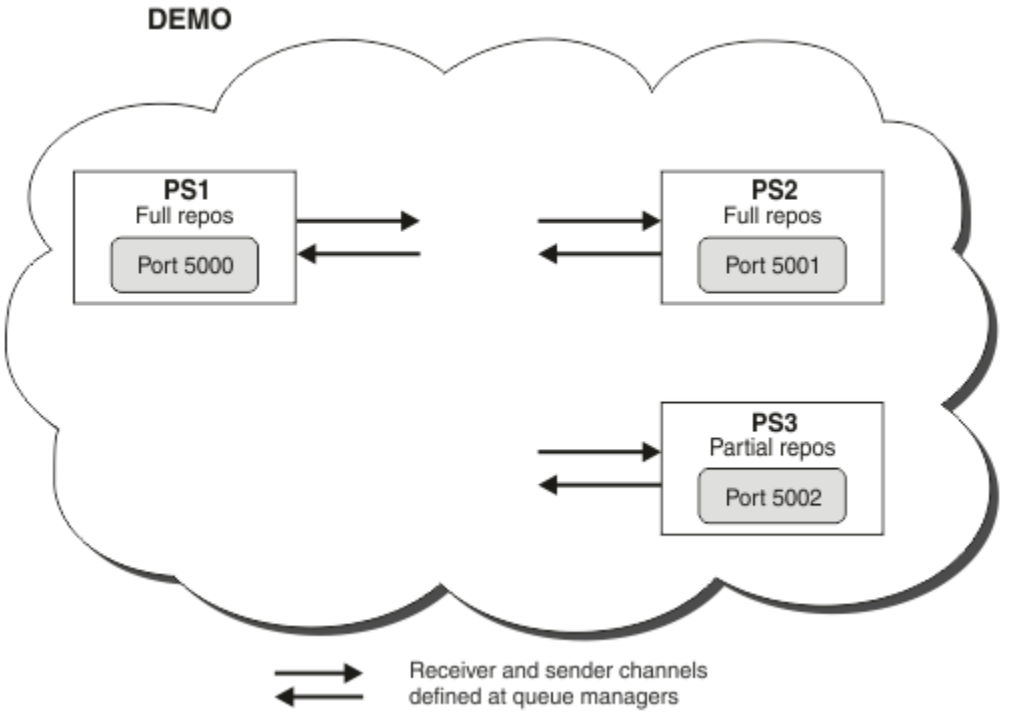
在此场景中，您将创建一个简单的三个队列管理器集群，并将其配置为允许在一个队列管理器上创建的预订接收由连接到另一个队列管理器的应用程序发布的消息。

开始之前

此方案的起点是现有 IBM MQ 安装。 有关安装 IBM MQ 的指示信息，请遵循 在 Windows 上安装 IBM MQ 服务器中的步骤。

关于此任务

通过完成此场景中的步骤，您首先创建以下集群:



。

此集群由三个队列管理器组成，其中两个队列管理器定义为完整存储库队列管理器。

然后在队列管理器 PS3 上定义集群主题。 通过创建集群主题，您已将集群设置为发布/预订集群。 要测试发布/预订集群，请在任何队列管理器上预订该主题，然后从另一个队列管理器将消息发布到该主题，并检查您的预订是否接收到该消息。

相关任务

设计发布/预订集群

配置队列管理器集群

创建并启动队列管理器

创建并启动三个队列管理器，分别称为 PS1、PS2 和 PS3。

过程

1. 创建并启动队列管理器 PS1。
   a) 创建队列管理器。

      在命令行中输入以下命令:

      ```
      crtmqm PS1
      ```

   b) 启动队列管理器。

      在命令行中输入以下命令:

      ```
      strmqm PS1
      ```

2. 重复步骤 1 创建并启动队列管理器 PS2。
3. 重复步骤 1 创建并启动队列管理器 PS3。

下一步做什么

现在您可以配置第一个队列管理器了。

配置第一个队列管理器

使用 MQSC 接口定义 PS1 的监听器和接收器通道，将队列管理器设置为群集的完整存储库，并定义 PS1 至 PS2 的发送器通道，以便两个完整存储库交换信息。

开始之前

本任务假定您已完成 "第 32 页的『创建并启动队列管理器』中的步骤。

过程

1. 定义并启动 PS1 的监听器。
   a) 启动 MQSC 界面。

      在命令行中输入以下命令：

      ```
      runmqsc PS1
      ```

   b) 定义听众。

      输入以下 MQSC 命令：

      ```
      DEFINE LISTENER(PS1_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5000)
      ```

   c) 启动侦听器。

      输入以下 MQSC 命令：

      ```
      START LISTENER(PS1_LS)
      ```

2. 将队列管理器设置为群集的完整存储库。

   输入以下 MQSC 命令：

   ```
   ALTER QMGR REPOS(DEMO)
   ```

3. 为 PS1 定义一个接收器通道，以便群集中的其他队列管理器与其通信。

   输入以下 MQSC 命令：

   ```
   DEFINE CHANNEL(DEMO.PS1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')
   CLUSTER(DEMO)
       DESCR('TCP Cluster-receiver channel for queue manager PS1')
   ```

4. 定义一个从 PS1 到 PS2 的发送通道，让两个完整的存储库交换信息。

   输入以下 MQSC 命令：

   ```
   DEFINE CHANNEL(DEMO.PS2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5001)')
   CLUSTER(DEMO)
       DESCR('TCP Cluster-sender channel from PS1 to queue manager PS2')
   ```

下一步做什么

现在您可以配置第二个队列管理器了。

配置第二个队列管理器

使用 MQSC 接口定义 PS2 的监听器和接收器通道，将队列管理器设置为群集的完整存储库，并定义 PS2 至 PS1 的发送器通道，以便两个完整存储库交换信息。

开始之前
本任务假定您已完成 "第 33 页的『配置第一个队列管理器』中的步骤。

过程
1. 定义并启动 PS2 的监听器。
   a) 启动 MQSC 界面。
      在命令行中输入以下命令：
      ```
      runmqsc PS2
      ```
   b) 定义听众。
      输入以下 MQSC 命令：
      ```
      DEFINE LISTENER(PS2_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5001)
      ```
   c) 启动侦听器。
      输入以下 MQSC 命令：
      ```
      START LISTENER(PS2_LS)
      ```
2. 将队列管理器设置为群集的完整存储库。
   输入以下 MQSC 命令：
   ```
   ALTER QMGR REPOS(DEMO)
   ```
3. 为 PS2 定义一个接收器通道，以便群集中的其他队列管理器与其通信。
   输入以下 MQSC 命令：
   ```
   DEFINE CHANNEL(DEMO.PS2) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5001)')
   CLUSTER(DEMO)
       DESCR('TCP Cluster-receiver channel for queue manager PS2')
   ```
4. 定义一个从 PS2 到 PS1 的发送通道，让两个完整的存储库交换信息。
   输入以下 MQSC 命令：
   ```
   DEFINE CHANNEL(DEMO.PS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')
   CLUSTER(DEMO)
       DESCR('TCP Cluster-sender channel from PS2 to PS1')
   ```

下一步做什么
现在您可以配置第三个队列管理器了。

配置第三个队列管理器
使用 MQSC 接口定义 PS3 的监听器和接收器通道。 Join PS3 into the cluster by defining a sender channel from PS3 to one of the full repository queue managers.

开始之前
本任务假定您已完成 "第 33 页的『配置第二个队列管理器』中的步骤。

过程
1. 定义并启动 PS3 的监听器。
   a) 启动 MQSC 界面。
      在命令行中输入以下命令：

```
runmqsc PS3
```

b) 定义听众。

输入以下 MQSC 命令：

```
DEFINE LISTENER(PS3_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5002)
```

c) 启动侦听器。

输入以下 MQSC 命令：

```
START LISTENER(PS3_LS)
```

2. 为 PS3 定义一个接收通道，以便集群中的其他队列管理器与之通信。

输入以下 MQSC 命令：

```
DEFINE CHANNEL(DEMO.PS3) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5002)')
CLUSTER(DEMO)
    DESCR('TCP Cluster-receiver channel for queue manager PS3')
```

3. 定义从 PS3 到其中一个完整存储库队列管理器（例如 PS1）的发送通道。 这将把 PS3 加入群集。

输入以下 MQSC 命令：

```
DEFINE CHANNEL(DEMO.PS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')
CLUSTER(DEMO)
    DESCR('TCP Cluster-sender channel from PS3 to PS1')
```

4. 验证 PS3 是否已成功加入群集。

输入以下 MQSC 命令：

```
 DISPLAY CLUSQMGR(*) QMTYPE
```

该命令返回三个条目，QM1、QM2 和 QM3 各一个。 QM1 和 QM2 的 "**QMTYPE** 应为 REPOS，QM3 的 "**QMTYPE** 应为 NORMAL。

下一步做什么

现在您可以定义群组主题了。

定义集群主题

发布和预订应用程序可以发布到任何主题字符串，而无需定义受管主题对象。 但是，如果发布应用程序连接到与创建预订的队列管理器不同的集群队列管理器，那么必须定义受管主题对象并将其添加到集群。 要使主题成为集群主题，请在其定义中指定集群的名称。

开始之前

此任务假定您已完成 第 34 页的『配置第三个队列管理器』中的步骤。

关于此任务

受管主题对象标识主题树中通过其主题字符串进行集群的点。 发布和预订应用程序可以在该点或以下使用任何主题字符串，并且它们的消息会在队列管理器之间自动传输。

定义集群主题时，还应选择其路由模型。 有关集群中发布路由的更多信息，请参阅 设计发布/预订集群。

对于此场景，我们使用缺省路由 *DIRECT*。 这意味着将消息从发布队列管理器直接发送到预订队列管理器。

过程

1. 在 PS3 上定义集群主题 SCORES 。

要使主题成为集群主题，请指定集群的名称，并设置集群路由 (**CLROUTE**) 要用于此主题的发布和预订。

a) 启动 MQSC 接口。

在命令行中，输入以下命令:

```
runmqsc PS3
```

b) 定义集群主题 SCORES。

输入以下 MQSC 命令:

```
DEFINE TOPIC(SCORES) TOPICSTR('/Sport/Scores') CLUSTER(DEMO) CLROUTE(DIRECT)
```

c) 输入 end 以退出 PS3 的 MQSC 接口。

2. 验证 PS1 上的主题定义。

a) 启动 PS1 的 MQSC 接口。

在命令行中，输入以下命令:

```
runmqsc PS1
```

b) 显示集群主题 SCORES 的集群状态。

输入以下 MQSC 命令:

```
DISPLAY TCLUSTER(SCORES) CLSTATE
```

集群主题 SCORES 的 **CLSTATE** 显示为 ACTIVE。

下一步做什么

有关此任务的更详细探索，请参阅 配置发布/预订集群。

现在，您已准备好验证解决方案。 请参阅 第 36 页的『测试发布/预订集群』。

测试发布/预订集群

通过发布和预订来自集群中不同队列管理器的主题字符串来测试发布/预订集群。

开始之前

此任务假定您已完成第 35 页的『定义集群主题』中的步骤。

关于此任务

通过使用命令行以及 IBM MQ 随附的 amqspub 和 amqssub 样本应用程序，您可以从一个队列管理器发布主题，并与其他队列管理器一起预订该主题。 将消息发布到主题时，预订队列管理器将接收到该消息。

过程

1. 在命令行中，输入以下命令:

```
amqspub /Sport/Scores/Football PS1
```

2. 同时，在单独的命令行中，输入以下命令:

```
amqssub /Sport/Scores/Football PS2
```

```
amqssub /Sport/Scores/Football PS3
```

3. 在第一个命令行中，输入消息。

消息显示在两个预订命令行中。

注: 如果在 10 秒内未收到发布，那么 amqssub 应用程序将超时。

结果
已完成发布/预订集群设置。

下一步做什么
尝试为主题树的不同分支定义不同的主题对象，并使用不同的路由模型。

发布/预订层次结构方案
演示发布/预订层次结构的使用的三种方案。 这三个方案都设置了相同的简单发布/预订拓扑。 在每个方案中，队列管理器都依赖于不同的方法来连接到层次结构中的相邻队列管理器。

以下子主题中描述了可用的发布/预订层次结构方案:

相关概念
发布/预订层次结构

发布/订阅层次结构方案 1：使用带有队列管理器名称别名的点对点通道
这是三个方案中的第一个方案，这三个方案以不同方式设置了发布/订阅层次结构，以建立队列管理器之间的连接。 该方案设置了一个发布/订阅层次结构，使用带有队列管理器名称别名的点对点通道。

关于此任务
这组方案都使用一个名为 "QM1 的父队列管理器和两个名为 "QM2 和 "QM3"的子队列管理器。

情景 1 分成几个较小的部分，以便于了解整个过程。

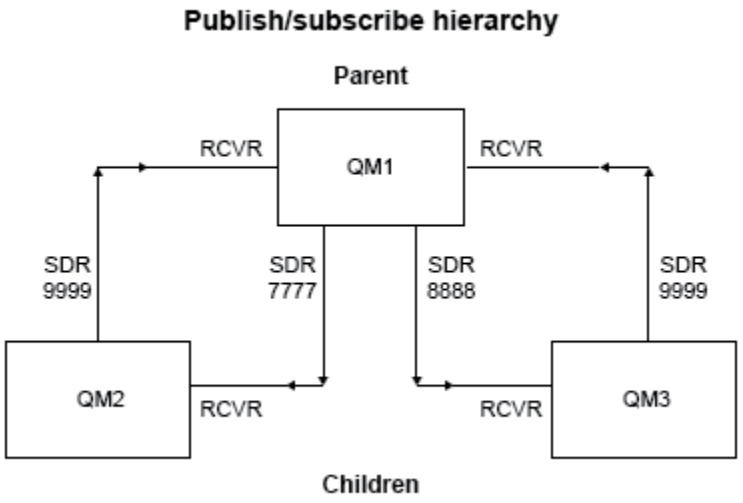


图 *3:* 拓扑图显示了典型的发布者/订阅层次结构中队列管理器之间的关系。

过程
1. 创建队列管理器。
   a) 使用以下命令创建并启动名为 "QM1、"QM2 和 "QM3 的三个队列管理器:

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM1
strmqm QM1

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM2
strmqm QM2

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM3
strmqm QM3
```

b) 在所有三个队列管理器上使用以下命令启用队列管理器发布/订阅模式：

```
ALTER QMGR PSMODE(ENABLED)
```

2. 使用与父队列管理器名称相同的队列管理器别名，在队列管理器之间建立点对点通道连接。

a) 在 "QM2 到 "QM1 之间定义传输队列和队列管理器别名。 为 "QM1 定义一个发送通道，并为 "QM1 上创建的发送通道为 "QM2 定义一个接收通道：

```
DEFINE QLOCAL(QM1.XMITQ) USAGE(XMITQ)

DEFINE QREMOTE (QM1) RNAME('') RQMNAME(QM1) XMITQ(QM1.XMITQ)

DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1.XMITQ)
TRPTYPE(TCP)

DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(RCVR) TRPTYPE(TCP)
```

b) 在 "QM3 到 "QM1 之间定义传输队列和队列管理器别名。 为 "QM1 定义发送信道，为 "QM3 定义在 "QM1 上创建的发送信道的接收信道：

```
DEFINE QLOCAL(QM1.XMITQ) USAGE(XMITQ)

DEFINE QREMOTE (QM1) RNAME('') RQMNAME(QM1) XMITQ(QM1.XMITQ)

DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1.XMITQ)
TRPTYPE(TCP)

DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(RCVR) TRPTYPE(TCP)
```

c) Define a transmission queue and queue manager alias on QM1 to QM2 and QM3. 为 "QM2 和 "QM3"定义发送通道，为 "QM2 和 "QM3 创建的发送通道定义接收通道，为 "QM1 创建接收通道：

```
DEFINE QLOCAL(QM2.XMITQ) USAGE(XMITQ)

DEFINE QREMOTE (QM2) RNAME('') RQMNAME(QM2) XMITQ(QM2.XMITQ)

DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(SDR) CONNAME('localhost(7777)') XMITQ(QM2.XMITQ)
TRPTYPE(TCP)

DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)

DEFINE QLOCAL(QM3.XMITQ) USAGE(XMITQ)

DEFINE QREMOTE (QM3) RNAME('') RQMNAME(QM3) XMITQ(QM3.XMITQ)

DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(SDR) CONNAME('localhost(8888)') XMITQ(QM3.XMITQ)
TRPTYPE(TCP)

DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)
```

d) 在队列管理器上启动相应的监听器：

```
runmqlsr -m QM1 -t TCP -p 9999 &
runmqlsr -m QM2 -t TCP -p 7777 &
runmqlsr -m QM3 -t TCP -p 8888 &
```

e) 启动以下频道

i) 在 QM1 上：

```
START CHANNEL('QM1.TO.QM2')

START CHANNEL('QM1.TO.QM3')
```

ii) 在 QM2 上：

```
START CHANNEL('QM2.TO.QM1')
```

iii) 在 QM3 上：

```
START CHANNEL('QM3.TO.QM1')
```

f) 检查所有通道是否都已启动：

```
DISPLAY CHSTATUS('QM1.TO.QM2')
DISPLAY CHSTATUS('QM1.TO.QM3')
DISPLAY CHSTATUS('QM2.TO.QM1')
DISPLAY CHSTATUS('QM3.TO.QM1')
```

g)

3. 连接队列管理器并定义主题。

   Connect the child queue managers QM2 and QM3 to the parent queue manager QM1.

   a) 在 "QM2 和 "QM3"上，将父队列管理器设置为 "QM1

```
ALTER QMGR PARENT (QM1)
```

   b) 在所有队列管理器上运行以下命令，检查子队列管理器是否连接到父队列管理器：

```
DISPLAY PUBSUB TYPE(ALL)
```

   显示命令输出。 例如，下面是 QM1 的输出结果，主要细节突出显示：

```
DISPLAY PUBSUB ALL
1 : DISPLAY PUBSUB ALL
AMQ8723: Display pub/sub status details.
QMNAME(QM1)                TYPE(LOCAL)
STATUS(ACTIVE)              SUBCOUNT(6)
TPCOUNT(9)
AMQ8723: Display pub/sub status details.
QMNAME(QM2) TYPE(CHILD)
STATUS(ACTIVE) SUBCOUNT(NONE)
TPCOUNT(NONE)
AMQ8723: Display pub/sub status details.
QMNAME(QM3) TYPE(CHILD)
STATUS(ACTIVE) SUBCOUNT(NONE)
TPCOUNT(NONE)
```

4. 使用 "amqspub.exe 和 "amqssub.exe 应用程序发布和订阅主题。

   a) 在第一个命令窗口中运行该命令：

```
amqspub Sport/Soccer QM2
```

   b) 在第二个命令窗口中运行该命令：

```
amqssub Sport/Soccer QM1
```

   c) 在第三个命令窗口中运行该命令：

```
amqssub Sport/Soccer QM3
```

结果

第二个和第三个命令窗口中的 "amqssub.exe 应用程序接收第一个命令窗口中发布的信息。

发布/预订层次结构方案 2: 将同名的点到点通道用于传输队列和远程队列管理器

这是三种场景中的第二种场景，它们以不同方式设置发布/预订层次结构，以建立队列管理器之间的连接。此方案设置一个发布/预订层次结构，该层次结构使用与远程队列管理器具有相同传输队列名称的点到点通道。

关于此任务

这组方案都使用一个名为 "QM1 的父队列管理器和两个名为 "QM2 和 "QM3"的子队列管理器。

此场景复用 第 37 页的『发布/订阅层次结构方案 1：使用带有队列管理器名称别名的点对点通道』中的步骤 1， 3 和 4。

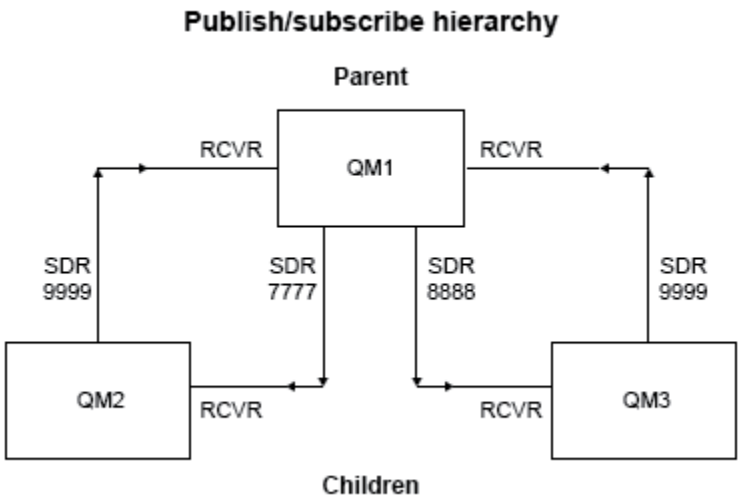


图 *4:* 此拓扑图显示典型发布者/预订层次结构中的队列管理器之间的关系。

过程

1. 创建队列管理器。
   a) 使用以下命令创建并启动名为 "QM1、"QM2 和 "QM3 的三个队列管理器：

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM1
strmqm QM1

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM2
strmqm QM2

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM3
strmqm QM3
```

   b) 在所有三个队列管理器上使用以下命令启用队列管理器发布/订阅模式：

```
ALTER QMGR PSMODE(ENABLED)
```

2. 使用与父队列管理器同名的传输队列在队列管理器之间建立点到点通道连接。
   a) 在 QM2 到 QM1 上定义传输队列。 为 QM1 上创建的 QM2 的发送方通道定义到 QM1 的发送方通道和接收方通道：

```
DEFINE QLOCAL(QM1) USAGE(XMITQ)

DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1)
TRPTYPE(TCP)

DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(RCVR) TRPTYPE(TCP)
```

   b) 在 QM3 到 QM1 上定义传输队列。 定义 QM1 的发送方通道以及在 QM1 上为 QM3 创建的发送方通道的接收方通道：

```
DEFINE QLOCAL(QM1) USAGE(XMITQ)

DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1)
TRPTYPE(TCP)

DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(RCVR) TRPTYPE(TCP)
```

c) 在 QM1 到 QM2 和 QM3 上定义传输队列。 定义到 QM2 和 QM3 的发送方通道，以及在 QM2 和 QM3 上为 QM1 创建的发送方通道的接收方通道:

```
DEFINE QLOCAL(QM2) USAGE(XMITQ)

DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(SDR) CONNAME('localhost(7777)') XMITQ(QM2)
TRPTYPE(TCP)

DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)

DEFINE QLOCAL(QM3) USAGE(XMITQ)

DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(SDR) CONNAME('localhost(8888)') XMITQ(QM3)
TRPTYPE(TCP)

DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)
```

d) 在队列管理器上启动相应的侦听器:

```
runmqlsr -m QM1 -t TCP -p 9999 &
runmqlsr -m QM2 -t TCP -p 7777 &
runmqlsr -m QM3 -t TCP -p 8888 &
```

e) 启动以下通道:

i) 在 QM1 上:

```
START CHANNEL('QM1.TO.QM2')

START CHANNEL('QM1.TO.QM3')
```

ii) 在 QM2 上:

```
START CHANNEL('QM2.TO.QM1')
```

iii) 在 QM3 上:

```
START CHANNEL('QM3.TO.QM1')
```

f) 检查所有通道是否已启动:

```
DISPLAY CHSTATUS('QM1.TO.QM2')

DISPLAY CHSTATUS('QM1.TO.QM3')

DISPLAY CHSTATUS('QM2.TO.QM1')

DISPLAY CHSTATUS('QM3.TO.QM1')
```

3. 连接队列管理器并定义主题。

Connect the child queue managers QM2 and QM3 to the parent queue manager QM1.

a) 在 "QM2 和 "QM3"上，将父队列管理器设置为 "QM1

```
ALTER QMGR PARENT (QM1)
```

b) 在所有队列管理器上运行以下命令，检查子队列管理器是否连接到父队列管理器:

```
DISPLAY PUBSUB TYPE(ALL)
```

显示命令输出。 例如，下面是 QM1 的输出结果，主要细节突出显示:

```
DISPLAY PUBSUB ALL
1 : DISPLAY PUBSUB ALL
AMQ8723: Display pub/sub status details.
QMNAME(QM1)                TYPE(LOCAL)
STATUS(ACTIVE)              SUBCOUNT(6)
TPCOUNT(9)
AMQ8723: Display pub/sub status details.
QMNAME(QM2) TYPE(CHILD)
STATUS(ACTIVE) SUBCOUNT(NONE)
TPCOUNT(NONE)
AMQ8723: Display pub/sub status details.
```

```
QMNAME(QM3) TYPE(CHILD)
STATUS(ACTIVE) SUBCOUNT(NONE)
TPCOUNT(NONE)
```

4. 使用 "amqspub.exe 和 "amqssub.exe 应用程序发布和订阅主题。
   a) 在第一个命令窗口中运行该命令：

   ```
   amqspub Sport/Soccer QM2
   ```

   b) 在第二个命令窗口中运行该命令：

   ```
   amqssub Sport/Soccer QM1
   ```

   c) 在第三个命令窗口中运行该命令：

   ```
   amqssub Sport/Soccer QM3
   ```

结果

第二个和第三个命令窗口中的 amqssub.exe 应用程序接收在第一个命令窗口中发布的消息。

相关任务

第 37 页的『发布/订阅层次结构方案 1：使用带有队列管理器名称别名的点对点通道』
这是三个方案中的第一个方案，这三个方案以不同方式设置了发布/订阅层次结构，以建立队列管理器之间的连接。 该方案设置了一个发布/订阅层次结构，使用带有队列管理器名称别名的点对点通道。

第 42 页的『发布/预订层次结构方案 3: 使用集群通道添加队列管理器』
这是三种场景中的第三种场景，它们以不同方式设置发布/预订层次结构以建立队列管理器之间的连接。 此方案使用集群通道将队列管理器添加到层次结构。

将队列管理器连接到发布/预订层次结构

发布/预订层次结构方案 3: 使用集群通道添加队列管理器

这是三种场景中的第三种场景，它们以不同方式设置发布/预订层次结构以建立队列管理器之间的连接。 此方案使用集群通道将队列管理器添加到层次结构。

关于此任务

这组方案都使用一个名为 "QM1 的父队列管理器和两个名为 "QM2 和 "QM3"的子队列管理器。

注: 此方案仅使用集群配置将队列管理器连接在一起，而不是通过集群主题传播发布/预订流量。 在同一集群中定义队列管理器之间的子/父层次结构关系时，将根据主题树中主题的发布和预订作用域设置在队列管理器之间传播发布内容。 请务必不要使用主题的集群名称设置将主题添加到集群中。 如果使用集群名称，那么拓扑将成为发布/预订集群，并且不需要定义子级/父级层次结构关系。 请参阅 第 31 页的『方案: 创建发布/预订集群』 和 规划分布式发布/预订网络。

此场景复用 第 37 页的『发布/订阅层次结构方案 1：使用带有队列管理器名称别名的点对点通道』中的步骤 1， 3 和 4。

此方案创建名为 DEMO 的集群，其中 QM1 和 QM2 是完整存储库， QM3 是部分存储库。 队列管理器 QM1 是队列管理器 QM2 和 QM3 的父代。

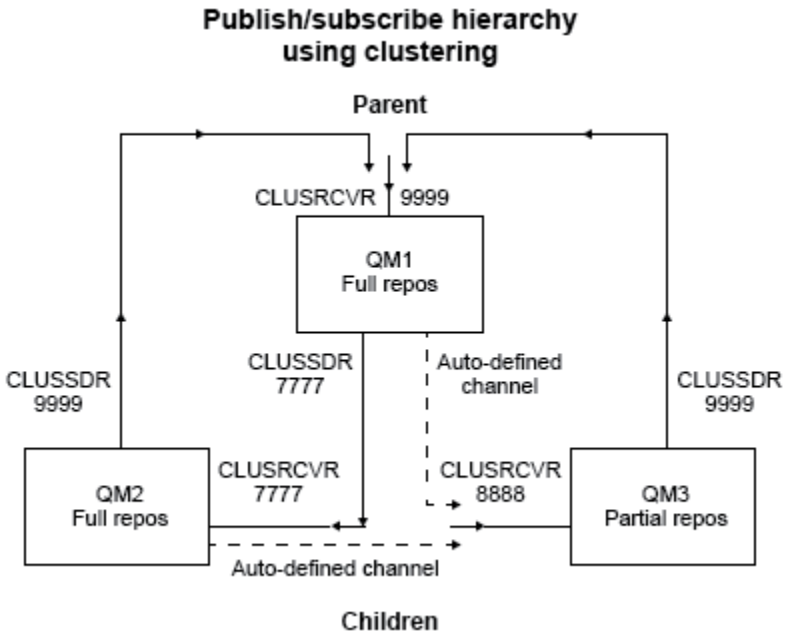


图 *5:* 显示使用集群通道的队列管理器之间的关系的拓扑图。

过程

1. 创建队列管理器。

   a) 使用以下命令创建并启动名为 "QM1、"QM2 和 "QM3 的三个队列管理器：

   ```
   crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM1
   strmqm QM1

   crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM2
   strmqm QM2

   crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM3
   strmqm QM3
   ```

   b) 在所有三个队列管理器上使用以下命令启用队列管理器发布/订阅模式：

   ```
   ALTER QMGR PSMODE(ENABLED)
   ```

2. 在集群的队列管理器之间建立点到点通道连接。

   a) 在 QM1 和 QM2 上，将 **REPOS** 参数设置为集群 DEMO 的名称：

   ```
   ALTER QMGR REPOS(DEMO)
   ```

   b) 在队列管理器上启动相应的侦听器：

   ```
   runmqlsr -m QM1 -t TCP -p 9999 &
   runmqlsr -m QM2 -t TCP -p 7777 &
   runmqlsr -m QM3 -t TCP -p 8888 &
   ```

   c) 在每个队列管理器上定义集群接收方通道：

      i) 在 QM1 上：

      ```
      DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('localhost(9999)')
      CLUSTER(DEMO)
      ```

      ii) 在 QM2 上：

      ```
      DEFINE CHANNEL(TO.QM2) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('localhost(7777)')
      CLUSTER(DEMO)
      ```

iii) 在 QM3 上:

```
DEFINE CHANNEL(TO.QM3) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('localhost(8888)')
CLUSTER(DEMO)
```

d) 定义集群中每个队列管理器上的完整存储库的集群发送方通道:

i) 在 QM1 上:

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('localhost(7777)')
CLUSTER(DEMO)
```

ii) 在 QM2 上:

```
DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('localhost(9999)')
CLUSTER(DEMO)
```

iii) QM3 可以具有到 QM1 或 QM2 上的完整存储库的集群发送方通道。 此示例定义到 QM1 的通道:

```
DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('localhost(9999)')
CLUSTER(DEMO)
```

3. 连接队列管理器并定义主题。

Connect the child queue managers QM2 and QM3 to the parent queue manager QM1.

a) 在 "QM2 和 "QM3"上，将父队列管理器设置为 "QM1

```
ALTER QMGR PARENT (QM1)
```

b) 在所有队列管理器上运行以下命令，检查子队列管理器是否连接到父队列管理器:

```
DISPLAY PUBSUB TYPE(ALL)
```

显示命令输出。 例如，下面是 QM1 的输出结果，主要细节突出显示:

```
DISPLAY PUBSUB ALL
1 : DISPLAY PUBSUB ALL
AMQ8723: Display pub/sub status details.
QMNAME(QM1)                TYPE(LOCAL)
STATUS(ACTIVE)              SUBCOUNT(6)
TPCOUNT(9)
AMQ8723: Display pub/sub status details.
QMNAME(QM2) TYPE(CHILD)
STATUS(ACTIVE) SUBCOUNT(NONE)
TPCOUNT(NONE)
AMQ8723: Display pub/sub status details.
QMNAME(QM3) TYPE(CHILD)
STATUS(ACTIVE) SUBCOUNT(NONE)
TPCOUNT(NONE)
```

4. 使用 "amqspub.exe 和 "amqssub.exe 应用程序发布和订阅主题。

a) 在第一个命令窗口中运行该命令:

```
amqspub Sport/Soccer QM2
```

b) 在第二个命令窗口中运行该命令:

```
amqssub Sport/Soccer QM1
```

c) 在第三个命令窗口中运行该命令:

```
amqssub Sport/Soccer QM3
```

结果

第二个和第三个命令窗口中的 amqssub.exe 应用程序接收在第一个命令窗口中发布的消息。

相关任务

第 37 页的『发布/订阅层次结构方案 1：使用带有队列管理器名称别名的点对点通道』
这是三个方案中的第一个方案，这三个方案以不同方式设置了发布/订阅层次结构，以建立队列管理器之间的连接。 该方案设置了一个发布/订阅层次结构，使用带有队列管理器名称别名的点对点通道。

第 39 页的『发布/预订层次结构方案 2: 将同名的点到点通道用于传输队列和远程队列管理器』
这是三种场景中的第二种场景，它们以不同方式设置发布/预订层次结构，以建立队列管理器之间的连接。 此方案设置一个发布/预订层次结构，该层次结构使用与远程队列管理器具有相同传输队列名称的点到点通道。

将队列管理器连接到发布/预订层次结构

业务支持方案

通过使用事务支持，您可以使应用程序能够可靠地处理数据库。

本部分介绍了事务支持。 使应用程序能够将 IBM MQ 与数据库产品配合使用所需的工作跨应用程序编程和系统管理领域。 将此处的信息与 落实和回退工作单元一起使用。

我们首先介绍构成事务的工作单元，然后描述您启用 IBM MQ 以将事务与数据库进行协调的方式。

相关概念

第 45 页的『介绍工作单元』
本主题介绍并定义工作单元，落实，回退和同步点的一般概念。 它还包含说明全局工作单元的两种方案。

介绍工作单元

本主题介绍并定义工作单元，落实，回退和同步点的一般概念。 它还包含说明全局工作单元的两种方案。

当程序将消息放入工作单元中的队列时，仅当程序 落实 工作单元时，这些消息才会对其他程序可视。 要落实工作单元，所有更新都必须成功，才能保持数据完整性。

如果程序检测到错误并决定不使 put 操作永久存在，那么它可以 回退 工作单元。 当程序执行回退时， IBM MQ 通过除去该工作单元放在队列上的消息来复原队列。

同样，当程序从工作单元中的一个或多个队列获取消息时，这些消息将保留在队列上，直到程序落实工作单元为止，但这些消息不可供其他程序检索。 当程序落实工作单元时，将从队列中永久删除这些消息。 如果程序回退工作单元，那么 IBM MQ 会通过使消息可供其他程序检索来复原队列。

在最简单的情况下，将在任务结束时作出落实或回退更改的决定。 但是，应用程序在任务中的其他逻辑点同步数据更改可能更有用。 这些逻辑点称为同步点 (或同步点) ，在两个同步点之间处理一组更新的时间段称为工作单元。 可以有几个 MQGET 调用和 MQPUT 调用作为一个工作单元的一部分。

通过 IBM MQ，我们需要区分 本地 和 全局 工作单元:

本地工作单元
是将唯一操作放入并从 IBM MQ 队列中获取的操作，并且使用 单阶段落实 进程在队列管理器中提供每个工作单元的协调。

当要更新的唯一资源是由单个 IBM MQ 队列管理器管理的队列时，请使用本地工作单元。 使用 MQCMIT 动词落实更新或使用 MQBACK 回退更新。

除了使用本地工作单元所涉及的日志管理以外，没有任何系统管理任务。 在将 MQPUT 和 MQGET 调用与 MQCMIT 和 MQBACK 配合使用的应用程序中，请尝试使用 MQPMO_SYNCPOINT 和 MQGMO_SYNCPOINT 选项。 (有关日志管理的信息，请参阅 管理日志文件。)

全局工作单元
是否还会更新其他资源 (例如关系数据库中的表)。 如果涉及多个 资源管理器 ，那么需要使用 两阶段落实 进程来协调全局工作单元的 事务管理器 软件。

如果还需要包含对关系数据库管理器软件 (例如 Db2， Oracle， Sybase 和 Informix) 的更新，请使用全局工作单元。

有几种使用全局工作单元的可能方案。 此处记录了两种场景:

1. 首先，队列管理器本身充当事务管理器。 在此场景中， MQI 动词控制全局工作单元; 它们在应用程序中使用 MQBEGIN 动词启动，然后使用 MQCMIT 落实或使用 MQBACK 回退。
2. 在第二个角色中，事务管理员角色由其他软件 (例如 TXSeries， Encina 或 Tuxedo) 执行。 在此场景中，事务管理器软件提供的 API 用于控制工作单元 (例如， EXEC CICS SYNCPOINT for TXSeries)。

以下部分描述了使用全局工作单元所需的所有步骤 (按两种方案组织):

- 第 46 页的『方案 1: 队列管理器执行协调』
- 第 66 页的『方案 2: 其他软件提供协调』

Multi 方案 1: 队列管理器执行协调

在方案 1 中，队列管理器充当事务管理器。 在此场景中， MQI 动词控制全局工作单元; 它们在应用程序中使用 MQBEGIN 动词启动，然后使用 MQCMIT 落实或使用 MQBACK 回退。

Multi 隔离级别

在 IBM MQ 中，根据在数据库中实现的事务隔离设计，队列上的消息可能在数据库更新之前可视。

当 IBM MQ 队列管理器作为 XA 事务管理器工作时，要协调 XA 资源管理器的更新，请遵循以下落实协议:

1. 准备所有 XA 资源管理器。
2. 落实 IBM MQ 队列管理器资源管理器。
3. 落实其他资源管理器。

在步骤 2 和 3 之间，应用程序可能会看到一条落实到队列的消息，但数据库中的相应行不会反映此消息。

如果配置了数据库以使应用程序的数据库 API 调用等待暂挂更新完成，那么这不是问题。

您可以通过以不同方式配置数据库来解决此问题。 所需配置的类型称为 "隔离级别"。 有关隔离级别的更多信息，请参阅数据库文档。 或者，您可以配置队列管理器以按以下反向顺序落实资源管理器:

1. 准备所有 XA 资源管理器。
2. 落实其他资源管理器。
3. 落实 IBM MQ 队列管理器资源管理器。

更改协议时，将最后落实 IBM MQ 队列管理器，因此从队列中读取消息的应用程序只有在完成相应的数据库更新后才会看到消息。

要配置队列管理器以使用此已更改的协议，请设置 **AMQ_REVERSE_COMMIT_ORDER** 环境变量。

在运行 **strmqm** 以启动队列管理器的环境中设置此环境变量。 例如，在启动队列管理器之前，请在 shell 中运行以下命令:

```
export AMQ_REVERSE_COMMIT_ORDER=1
```

注: 设置此环境变量可能会导致每个事务有额外的日志条目，因此这将对每个事务的性能产生较小的影响。

Multi 数据库协调

当队列管理器协调全局工作单元本身时，可以在工作单元内集成数据库更新。 即，可以编写混合 MQI 和 SQL 应用程序，并且可以使用 MQCMIT 和 MQBACK 动词来落实或回滚对队列和数据库的更改。

队列管理器使用 *X/Open* 分布式事务处理*: XA* 规范中描述的两阶段落实协议来实现此目标。 要落实工作单元时，队列管理器首先会询问每个参与的数据库管理器是否准备落实其更新。 仅当所有参与者 (包括队列管理器本身) 准备落实时，才会落实所有队列和数据库更新。 如果任何参与者无法准备其更新，那么将改为回退工作单元。

通常，全局工作单元通过以下方法在应用程序中实现 (伪代码):

```
MQBEGIN
MQGET (在消息选项中包含标志 MQGMO_SYNCPOINT)
MQPUT (在消息选项中包含标志 MQPMO_SYNCPOINT)
```

SQL INSERT
MQCMIT

MQBEGIN 的目的是表示全局工作单元的开始。 MQCMIT 的目的是表示全局工作单元的结束，并使用两阶段落实协议在所有参与的资源管理器中完成该工作单元。

当工作单元 (也称为 事务) 时 使用 MQCMIT 成功完成，在该工作单元中执行的所有操作都是永久或不可逆的。 如果工作单元因任何原因而失败，那么将改为回退所有操作。 无法使工作单元中的一个操作成为永久操作，而另一个操作已回退。 这是工作单元的原则: 要么工作单元内的所有操作都是永久性的，要么都不是永久性的。

注:

1. 应用程序员可以通过调用 MQBACK 来强制回退工作单元。 如果在调用 MQCMIT 之前应用程序或数据库失败 ，那么队列管理器也会回退工作单元。
2. 如果应用程序在不调用 MQCMIT 的情况下调用 MQDISC ，那么队列管理器的行为就像调用了 MQCMIT 一样，并落实工作单元。

在 MQBEGIN 和 MQCMIT 之间，队列管理器不会对数据库进行任何调用以更新其资源。 即，更改数据库表的唯一方式是由您的代码 (例如，伪代码中的 SQL INSERT) 进行更改。

如果队列管理器在落实协议期间失去与任何数据库管理器的联系，那么将提供完全恢复支持。 如果数据库管理器在处于不确定状态时变为不可用，即，它已成功准备落实，但尚未收到落实或回退决策，那么队列管理器将记住工作单元的结果，直到该结果成功交付到数据库为止。 同样，如果队列管理器因未完成的落实操作而终止，那么将在队列管理器重新启动时记住这些操作。 如果应用程序意外终止，那么工作单元的完整性不会受到损害，但结果取决于应用程序在流程中终止的位置，如 第 47 页的表 2 中所述。

下表概述了在数据库或应用程序失败时发生的情况:

表 *1:* 数据库服务器发生故障时发生的情况

故障发生	**结果**
在应用程序调用 MQCMIT 之前。	工作单元已回退。
在应用程序调用 MQCMIT 期间， **之前** 所有数据库都指示已成功准备。	工作单元已回退，原因码为 MQRC_BACKED_OUT。
在应用程序调用 MQCMIT 期间， **之后** 所有数据库都指示已成功准备，但之前都指示已成功落实。	工作单元由队列管理器保持在可恢复状态，原因码为 MQRC_OUTCOME_PENDING。
在应用程序调用 MQCMIT 期间， **之后** 所有数据库都指示它们已成功落实。	已落实工作单元，原因码为 MQRC_NONE。
在应用程序调用 MQCMIT 之后。	已落实工作单元，原因码为 MQRC_NONE。

表 *2:* 应用程序失败时发生的情况

故障发生	**结果**
在应用程序调用 MQCMIT 之前。	工作单元已回退。
在应用程序调用 MQCMIT 期间， **之前** 队列管理器已接收到应用程序的 MQCMIT 请求。	工作单元已回退。
在应用程序调用 MQCMIT 期间， **在** 队列管理器接收到应用程序的 MQCMIT 请求之后。	队列管理器尝试使用两阶段落实 (取决于数据库产品成功执行并落实其部分工作单元) 进行落实。

如果从 MQCMIT 返回的原因码为 MQRC_OUTCOME_PENDING ，那么队列管理器将记住工作单元，直到它能够重新建立与数据库服务器的联系，并告诉它落实其部分工作单元为止。 请参阅 第 61 页的『丢失与 XA 资源管理器的联系时的注意事项』 ，以获取有关如何以及何时执行恢复的信息。

队列管理器使用 XA 接口与数据库管理器通信，如 *X/Open* 分布式事务处理*: XA* 规范中所述。 这些函数调用的示例包括 xa_open ， xa_start ， xa_end ， xa_prepare 和 xa_commit。 我们使用术语 事务管理器 和 资源管理器 与 XA 规范中使用的术语相同。

Multi 限制

数据库协调支持存在限制。

具有以下限制：

- 在 MQI 客户机应用程序中 **不** 支持在 IBM MQ 工作单元内协调数据库更新的功能。 在客户机应用程序中使用 MQBEGIN 失败。 调用 MQBEGIN 的程序必须在与队列管理器相同的机器上作为 服务器 应用程序运行。

  注: 服务器 应用程序是已与必需的 IBM MQ 服务器库链接的程序; 客户机 应用程序是已与必需的 IBM MQ 客户机库链接的程序。 请参阅 为 IBM MQ MQI clients 和 构建过程应用程序 构建应用程序，以获取有关编译和链接以过程语言编写的程序的详细信息。
- 数据库服务器可以驻留在与队列管理器服务器不同的机器上，只要数据库客户机与队列管理器安装在同一机器上，它就支持此功能。 请参阅数据库产品的文档以确定其客户机软件是否可用于两阶段落实系统。
- 虽然队列管理器充当资源管理器 (为了参与方案 2 全局工作单元)，但无法使一个队列管理器在其方案 1 全局工作单元中协调另一个队列管理器。

Multi 切换装入文件

交换机装入文件是由 IBM MQ 应用程序和队列管理器中的代码装入的共享库 (Windows 系统上的 DLL)。 其目的是简化数据库的客户机共享库的装入，并将指针返回到 XA 函数。

必须在启动队列管理器之前指定交换机装入文件的详细信息。 详细信息放在 AIX, Linux, and Windows 系统上的 qm.ini 文件中。

- 在 Windows 和 Linux (x86 和 x86-64 平台) 系统上，使用 IBM MQ Explorer 来更新 qm.ini 文件。
- 在所有其他系统上，直接编辑文件 qm.ini。

如果交换机装入文件的 C 源支持方案 1 全局工作单元，那么会随 IBM MQ 安装一起提供该文件的 C 源。 源包含名为 MQStart 的函数。 装入交换机装入文件时，队列管理器将调用此函数，这将返回称为 *XA* 交换机的结构的地址。

XA 切换结构存在于数据库客户机共享库中，并且包含许多函数指针，如 第 48 页的表 3 中所述:

表 *3: XA* 切换函数指针

函数指针名称	**XA 函数**	**用途**
xa_open_entry	xa_open	连接到数据库
xa_close_entry	xa_close	与数据库断开连接
xa_start_entry	xa_start	启动全局工作单元的分支
xa_end_entry	xa_end	暂挂全局工作单元的分支
xa_rollback_entry	XA_rollback	回滚全局工作单元的分支
xa_prepare_entry	xa_prepare	准备落实全局工作单元的分支
xa_commit_entry	xa_commit	落实全局工作单元的分支
xa_recover_entry	xa_recover	从数据库中发现它是否具有不确定的工作单元
xa_forget_entry	xa_算了	允许数据库忘记全局工作单元的分支
xa_complete_entry	xa_complete	完成全局工作单元的分支

在应用程序中的第一个 MQBEGIN 调用期间，作为 MQBEGIN 的一部分执行的 IBM MQ 代码将装入切换装入文件，并在数据库共享库中调用 xa_open 函数。 类似地，在队列管理器启动期间以及在其他后续场合，某些队列管理器进程装入切换装入文件并调用 xa_open。

您可以使用 动态注册来减少 xa_ * 调用数。 有关此优化方法的完整描述，请参阅 第 64 页的『XA 动态注册』。

Multi 配置系统以进行数据库协调

在数据库管理器可以参与由队列管理器协调的全局工作单元之前，必须执行若干任务。 这些描述如下:

- 第 49 页的『安装和配置数据库产品』
- 第 49 页的『创建交换机装入文件』
- 第 50 页的『向队列管理器添加配置信息』
- 第 51 页的『编写和修改应用程序』
- 第 52 页的『测试系统』

Multi 安装和配置数据库产品

要安装和配置数据库产品，请参阅产品自己的文档。 本部分中的本主题描述了常规配置问题以及它们如何与 IBM MQ 与数据库之间的互操作相关。

数据库连接

与队列管理器建立标准连接的应用程序与单独的本地队列管理器代理进程中的线程相关联。 (不是 *fastpath* 连接的连接是此上下文中的 标准 连接。 请参阅 使用 MQCONNX 调用连接到队列管理器。)

当应用程序发出 **MQBEGIN** 时，它和代理进程都将调用数据库客户机库中的 **xa_open** 函数。 为此，数据库客户机库代码 将 连接到 来自应用程序和队列管理器进程的工作单元中涉及的数据库。 只要应用程序保持与队列管理器的连接，就会维护这些数据库连接。

如果数据库仅支持有限数量的用户或连接，那么这是一个重要的注意事项，因为正在与数据库建立两个连接以支持一个应用程序。

客户机/服务器配置

装入到 IBM MQ 队列管理器和应用程序进程中的数据库客户机库 **必须** 能够发送到其服务器并从其服务器接收。 请确保:

- 数据库的客户机/服务器配置文件具有正确的详细信息
- 相关环境变量是在队列管理器 **和** 应用程序进程的环境中设置的

Multi 创建交换机装入文件

IBM MQ 随附样本 makefile，用于为受支持的数据库管理器构建切换装入文件。

MQ_INSTALLATION_PATH 表示 IBM MQ 安装所在的高级目录。

样本 makefile 与构建交换机装入文件所需的所有关联 C 源文件一起安装在以下目录中:

- 对于 IBM MQ for Windows，在 *MQ_INSTALLATION_PATH*`\tools\c\samples\xatm\` 目录中
- 对于 IBM MQ for UNIX 和 Linux 系统，在 *MQ_INSTALLATION_PATH*`/samp/xatm/` 目录中

用于构建交换机装入文件的样本目标包括:

- 对于 Db2，为 db2swit。
- 对于 Oracle， oraswit
- 对于 Informix，为 infswit
- 对于 Sybase， sybswit

Windows 生成的交换机文件放置在 `C:\Program Files\IBM\MQ\exits` 中。

Linux UNIX 如果您有 32 位队列管理器，那么样本 make 文件 xaswit.mak 将在 `/var/mqm/exits` 中安装 32 位交换机装入文件。

Linux UNIX 如果您有 64 位队列管理器，那么样本 make 文件 xaswit.mak 将在 /var/mqm/exits 中安装 32 位交换机装入文件，在 /var/mqm/exits64 中安装 64 位交换机装入文件。

V 9.3.0 Linux UNIX 如果系统不支持 32 位编译，请将 64 位目标用于数据库:

- 对于 Db2， db2swit64
- 对于 Oracle， oraswit64
- 对于 Informix， infswit64
- 对于 Sybase， sybswit64

文件安全性

由于 IBM MQ 无法控制的原因，操作系统可能无法通过 IBM MQ 装入交换机装入文件。 如果发生此情况，那么会将错误消息写入 IBM MQ 错误日志，并且可能导致 MQBEGIN 调用失败。 为了帮助确保操作系统不会使交换机装入文件的装入失败，您必须满足以下要求:

1. 切换装入文件必须在 qm.ini 文件中提供的位置中可用。
2. 交换机装入文件必须可供需要装入该文件的所有进程 (包括队列管理器进程和应用程序进程) 访问。
3. 交换机装入文件所依赖的所有库 (包括数据库产品提供的库) 都必须存在且可访问。

Multi 向队列管理器添加配置信息

为数据库管理器创建切换装入文件并将其放在安全位置时，必须向队列管理器指定该位置。

要指定位置，请执行以下步骤:

- 在 Windows 和 Linux (x86 和 x86-64 平台) 系统上，使用 IBM MQ Explorer。 在队列管理器属性面板中的 XA 资源管理器下指定切换装入文件的详细信息。
- 在所有其他系统上，在队列管理器的 qm.ini 文件的 XAResourceManager 节中指定交换机装入文件的详细信息。

为队列管理器要协调的数据库添加 XAResourceManager 节。 最常见的情况是只有一个数据库，因此只有一个 XAResourceManager 节。 有关涉及多个数据库的更复杂的配置的详细信息，请参阅 第 60 页的『多个数据库配置』。 XAResourceManager 节的属性如下所示:

名称 = 名称
用户选择的用于标识资源管理器的字符串。 实际上，它为 XAResourceManager 节提供了名称。 该名称是必需的，长度最多可以为 31 个字符。

您选择的名称必须唯一; 此 qm.ini 文件中必须只有一个具有此名称的 XAResourceManager 节。 该名称也应该有意义，因为当使用 dspmqtrn 命令时，队列管理器使用它在队列管理器错误日志消息和输出中都引用此资源管理器。 (请参阅 第 62 页的『使用 dspmqtrn 命令显示未完成的工作单元』 以获取更多信息。)

选择名称并启动队列管理器后，请勿更改 "名称" 属性。 有关更改配置信息的更多详细信息，请参阅 第 63 页的『更改配置信息』。

SwitchFile= 名称
这是先前构建的 XA 切换装入文件的名称。 这是必需属性。 队列管理器和 IBM MQ 应用程序进程中的代码两次尝试装入交换机装入文件:

1. 在队列管理器启动时
2. 在 IBM MQ 应用程序进程中首次调用 MQBEGIN 时

交换机装入文件的安全性和许可权属性必须允许这些进程执行此操作。

XAOpenString= 字符串
这是 IBM MQ 代码在其对数据库管理器的 xa_open 函数的调用中传递的数据字符串。 这是可选属性; 如果省略此属性，那么将假定字符串长度为零。

队列管理器和 IBM MQ 应用程序进程中的代码两次调用 xa_open 函数:

1. 在队列管理器启动时

2. 在 IBM MQ 应用程序进程中首次调用 MQBEGIN 时

此字符串的格式特定于每个数据库产品，将在该产品的文档中进行描述。 通常， xa_open 字符串包含认证信息 (用户名和密码) ，以允许连接到队列管理器和应用程序进程中的数据库。

从 IBM MQ 8.0.0 Fix Pack 4 开始，当 XAOpenString 包含密码时，您可以获取 IBM MQ 以保护此信息，而不是使密码在 `qm.ini` 文件中以纯文本显示。 IBM MQ 将用户名和密码 (以加密形式) 存储在另一个文件中，并使用这些凭证连接到数据库。 有关详细信息，请参阅 保护数据库认证详细信息。

XACloseString= 字符串

这是 IBM MQ 代码在其对数据库管理器的 xa_close 函数的调用中传递的数据字符串。 这是可选属性; 如果省略此属性，那么将假定字符串长度为零。

队列管理器和 IBM MQ 应用程序进程中的代码两次调用 xa_close 函数:

1. 在队列管理器启动时
2. 在 IBM MQ 应用程序进程中调用 MQDISC 时，先前已调用 MQBEGIN

此字符串的格式特定于每个数据库产品，将在该产品的文档中进行描述。 通常，该字符串为空，并且通常会从 XAResourceManager 节中省略 XACloseString 属性。

ThreadOfControl=THREAD |PROCESS (缺省值)

ThreadOf 控制值可以是 THREAD 或 PROCESS。 队列管理器将其用于序列化目的。 这是可选属性; 如果省略此属性，那么将采用值 PROCESS。

如果数据库客户机代码允许线程在不序列化的情况下调用 XA 函数，那么 ThreadOfControl 的值可以是 THREAD。 队列管理器假定它可以在必要时同时从多个线程调用数据库客户机共享库中的 XA 函数。

如果数据库客户机代码不允许线程以此方式调用其 XA 函数，那么 ThreadOfControl 的值必须为 PROCESS。 在这种情况下，队列管理器会序列化对数据库客户机共享库的所有调用，以便一次仅从特定进程中发出一个调用。 如果应用程序使用多个线程运行，那么您可能还需要确保应用程序执行类似的序列化。

请注意，数据库产品以这种方式处理多线程进程的能力是该产品的供应商面临的问题。 有关是否可以将 ThreadOf 控制属性设置为 THREAD 或 PROCESS 的详细信息，请参阅数据库产品的文档。 如果可以，建议将 ThreadOf 控件设置为 THREAD。 如果不确定，那么 更安全 选项将其设置为 PROCESS ，尽管您将失去使用 THREAD 的潜在性能优势。

Multi 编写和修改应用程序

如何实现全局工作单元。

第 45 页的『介绍工作单元』中描述了随 IBM MQ 安装一起提供的方案 1 全局工作单元的样本应用程序。

通常，全局工作单元通过以下方法在应用程序中实现 (伪代码):

```
MQBEGIN
MQGET
MQPUT
SQL INSERT
MQCMIT
```

MQBEGIN 的目的是表示全局工作单元的开始。 MQCMIT 的目的是表示全局工作单元的结束，并使用两阶段落实协议在所有参与的资源管理器中完成该工作单元。

在 MQBEGIN 和 MQCMIT 之间，队列管理器不会对数据库进行任何调用以更新其资源。 即，更改数据库表的唯一方式是由您的代码 (例如，伪代码中的 SQL INSERT) 进行更改。

就数据库而言，队列管理器的作用是告诉它全局工作单元何时开始，何时结束，以及全局工作单元是应该落实还是回滚。

就应用程序而言，队列管理器执行两个角色: 资源管理器 (其中资源是队列上的消息) 和全局工作单元的事务管理器。

从提供的样本程序开始，处理在这些程序中进行的各种 IBM MQ 和数据库 API 调用。 相关的 API 调用完全记录在 样本 IBM MQ 过程程序， MQI 中使用的数据类型以及 (对于数据库自己的 API) 数据库自己的文档中。

Multi 测试系统
您知道是否仅通过在测试期间运行应用程序和系统来正确配置这些应用程序和系统。 您可以通过构建和运行提供的其中一个样本程序来测试系统的配置 (队列管理器与数据库之间的成功通信)。

Multi 配置 Db2

Db2 支持和配置信息。

Db2 的受支持级别是在 系统要求 IBM MQ 页面上定义的。

注: 在队列管理器为 64 位的平台上不支持 Db2 的 32 位实例。

请执行以下操作:

1. 请检查环境变量设置。
2. 创建 Db2 交换机装入文件。
3. 添加资源管理器配置信息。
4. 必要时更改 Db2 配置参数。

请结合 第 49 页的『配置系统以进行数据库协调』中提供的一般信息阅读本信息。

警告: 如果在 AIX and Linux 平台上运行 db2profile ，那么将设置环境变量 LIBPATH 和 LD_LIBRARY_PATH。 建议使用 unset 这些环境变量。 请参阅 crtmqenv 或 setmqenv 以获取更多信息。

检查 Db2 环境变量设置

确保为队列管理器进程 **以及应用程序进程在** 中设置了 Db2 环境变量。 尤其是，必须始终设置启动队列管理器的 Db2INSTANCE 环境变量 **前** 。 DB2INSTANCE 环境变量标识包含要更新的 Db2 数据库的 Db2 实例。 例如:

- 在 AIX and Linux 系统上，使用:

```
export DB2INSTANCE=db2inst1
```

- 在 Windows 系统上，使用:

```
set DB2INSTANCE=Db2
```

在具有 Db2 数据库的 Windows 上，必须将用户 MUSR_MQADMIN 添加到 DB2USERS 组以允许队列管理器启动。

创建 Db2 交换机装入文件

创建 Db2 切换装入文件的最简单方法是使用样本文件 xaswit.mak， IBM MQ 提供该样本文件来为各种数据库产品构建切换装入文件。

Windows 在 Windows 系统上，可以在目录 *MQ_INSTALLATION_PATH*\tools\c\samples\xatm 中找到 xaswit.mak 。 *MQ_INSTALLATION_PATH* 表示 IBM MQ 安装所在的高级目录。 要使用 Microsoft Visual C++ 创建 Db2 切换装入文件，请使用:

```
nmake /f xaswit.mak db2swit.dll
```

生成的交换机文件放置在 C:\Program Files\IBM\MQ\exits 中。

Linux UNIX 您可以在目录 *MQ_INSTALLATION_PATH*/samp/xatm 中找到 xaswit.mak 。 *MQ_INSTALLATION_PATH* 表示 IBM MQ 安装所在的高级目录。

编辑 xaswit.mak 以 取消注释 与您正在使用的 Db2 版本相应的行。 然后使用以下命令执行 makefile:

```
make -f xaswit.mak db2swit
```

生成的 32 位交换机装入文件将放在 /var/mqm/exits 中。

生成的 64 位交换机装入文件放置在 /var/mqm/exits64 中。

V 9.3.0 如果系统不支持 32 位编译，请使用仅 64 位目标:

```
make -f xaswit.mak db2swit64
```

添加 Db2 的资源管理器配置信息

您必须修改队列管理器的配置信息，以将 Db2 声明为全局工作单元中的参与者。 在 第 50 页的『向队列管理器添加配置信息』中的更多详细信息中描述了如何以此方式修改配置信息。

- 在 Windows 和 Linux (x86 和 x86-64 平台) 系统上，使用 IBM MQ Explorer。 在队列管理器属性面板中的 XA 资源管理器下指定切换装入文件的详细信息。
- 在所有其他系统上，在队列管理器的 qm.ini 文件的 XAResourceManager 节中指定交换机装入文件的详细信息。

第 53 页的图 6 是 UNIX 样本，显示一个 XAResourceManager 条目，其中要协调的数据库名为 mydbname，此名称在 XAOpenString 中指定:

```
XAResourceManager:
  Name=mydb2
  SwitchFile=db2swit
  XAOpenString=mydbname,myuser,mypasswd,toc=t
  ThreadOfControl=THREAD
```

图 *6: UNIX* 上 *Db2* 的样本 *XAResourceManager* 条目

注:

1. ThreadOfControl=THREAD 不能与 8 之前的 Db2 版本配合使用。 将 ThreadOfControl 和 XAOpenString 参数 toc 设置为下列其中一个组合:
   - ThreadOfControl=THREAD 和 toc=t
   - ThreadOfControl=PROCESS 和 toc=p

   如果要使用 jdbcdb2 XA 切换装入文件来启用 JDBC/JTA 协调，那么必须使用 ThreadOfControl=PROCESS 和 toc=p。

更改 Db2 配置参数

对于队列管理器正在协调的每个 Db2 数据库，必须设置数据库特权，更改 tp_mon_name 参数，并重置 maxappls 参数。 要进行此操作，请执行以下步骤:

设置数据库特权

队列管理器进程在 AIX and Linux 系统上以有效用户和组 mqm 运行。 在 Windows 系统上，它们以启动队列管理器的用户身份运行。 这可以是以下之一:

1. 发出 strmqm 命令的用户，或者
2. 运行 IBM MQ Service COM 服务器的用户

缺省情况下，此用户称为 MUSR_MQADMIN。

如果未在 xa_open 字符串上指定用户名和密码，那么 Db2 将使用 **运行队列管理器的用户** 来认证 xa_open 调用。 如果此用户 (例如， AIX and Linux 系统上的用户 mqm) 在数据库中没有最低特权，那么数据库将拒绝认证 xa_open 调用。

同样的注意事项也适用于您的应用程序进程。 如果未在 xa_open 字符串上指定用户名和密码，那么 Db2 将使用运行应用程序的用户来认证在第一个 MQBEGIN 期间发出的 xa_open 调用。 同样，此用户必须在数据库中具有最低特权才能工作。

例如，通过发出以下 Db2 命令，在 mydbname 数据库中授予 mqm 用户连接权限:

```
db2 connect to mydbname
db2 grant connect on database to user mqm
```

有关安全性的更多信息，请参阅 第 61 页的『安全性注意事项』 。

Windows **更改 TP_MON_NAME 参数**

仅对于 Windows 系统上的 Db2 ，请更改 TP_MON_NAME 配置参数，以命名 Db2 用于调用队列管理器以进行动态注册的 DLL。

使用命令 `db2 update dbm cfg using TP_MON_NAME mqmax` 来命名 MQMAX.DLL 作为 Db2 用于调用队列管理器的库。 这必须存在于 PATH 中的目录中。

重置 maxappls 参数

您可能需要查看 *maxappls* 参数的设置，这将限制可以连接到数据库的最大应用程序数。 请参阅 第 49 页的『安装和配置数据库产品』 。

Multi 配置 Oracle

Oracle 支持和配置信息。

完成以下步骤:

1. 检查环境变量设置。
2. 创建 Oracle 交换机装入文件。
3. 添加资源管理器配置信息。
4. 根据需要更改 Oracle 配置参数。

有关 IBM MQ 支持的 Oracle 级别的当前列表，请参阅 系统要求 IBM MQ。

检查 Oracle 环境变量设置

确保为队列管理器进程以及应用程序进程设置了 Oracle 环境变量。 尤其是，在启动队列管理器之前，请始终设置以下环境变量:

ORACLE_HOME

Oracle 主目录。 例如，在 AIX and Linux 系统上，使用:

```
export ORACLE_HOME=/opt/oracle/product/8.1.6
```

在 Windows 系统上，使用:

```
set ORACLE_HOME=c:\oracle\ora81
```

ORACLE_SID

正在使用的 Oracle SID。 如果要将 Net8 用于客户机/服务器连接，那么可能不必设置此环境变量。 请参阅 Oracle 文档。

以下示例是在 AIX and Linux 系统上设置此环境变量的示例:

```
export ORACLE_SID=sid1
```

Windows 系统上的等效项为:

```
set ORACLE_SID=sid1
```

注: 必须将 PATH 环境变量设置为包含二进制文件目录 (例如 `ORACLE_INSTALL_DIR/VERSION/32BIT_NAME/bin` 或 `ORACLE_INSTALL_DIR/VERSION/64BIT_NAME/bin`)，否则您可能会看到一条消息，指出机器中缺少 oraclient 库。

如果在 Windows 64 位系统上运行队列管理器，那么必须仅安装 64 位 Oracle 客户机。 由 64 位队列管理器装入的交换机装入文件必须访问 Oracle 64 位客户机库。

创建 Oracle 交换机装入文件

要创建 Oracle 交换机装入文件，请使用样本文件 `xaswit.mak`， IBM MQ 提供此文件来为各种数据库产品构建交换机装入文件。

Windows 在 Windows 系统上，可以在目录 `C:\Program Files\IBM\MQ\tools\c\samples\xatm` 中找到 `xaswit.mak` 。 要使用 Microsoft Visual C++ 创建 Oracle 交换机装入文件，请使用:

```
nmake /f xaswit.mak oraswit.dll
```

注: 这些交换机装入文件只能与 C 应用程序配合使用。 对于 Java 应用程序，请参阅 使用 IBM MQ classes for Java 的 JTA/JDBC 协调。

生成的交换机文件放置在 *MQ_INSTALLATION_PATH*`\exits` 中。 *MQ_INSTALLATION_PATH* 表示 IBM MQ 安装所在的高级目录。

Linux UNIX 您可以在目录 *MQ_INSTALLATION_PATH*`/samp/xatm` 中找到 `xaswit.mak` 。 *MQ_INSTALLATION_PATH* 表示 IBM MQ 安装所在的高级目录。

编辑 `xaswit.mak` 以取消注释与您正在使用的 Oracle 版本相应的行。 然后使用以下命令执行 makefile:

```
make -f xaswit.mak oraswit
```

安装 IBM MQ 时， `MQ_INSTALLATION_PATH/samp/xatm` 的内容是只读的，因此要编辑 `xaswit.mak`，将 `samp/xatm` 中的所有文件复制到另一个目录，修改 `xaswit.mak`，然后从该目录运行 `make -f xaswit.makoraswit` 。

生成的 32 位交换机装入文件放置在 `/var/mqm/exits` 中。

生成的 64 位交换机装入文件放置在 `/var/mqm/exits64` 中。

V 9.3.0 如果系统不支持 32 位编译，请使用仅 64 位目标:

```
make -f xaswit.mak oraswit64
```

为 Oracle 添加资源管理器配置信息

您必须修改队列管理器的配置信息，以将 Oracle 声明为全局工作单元中的参与者。 在 第 50 页的『向队列管理器添加配置信息』中更详细地描述了如何以此方式修改队列管理器的配置信息。

- 在 Windows 和 Linux (x86 和 x86-64 平台) 系统上，使用 IBM MQ Explorer。 在队列管理器属性面板中的 XA 资源管理器下指定切换装入文件的详细信息。
- 在所有其他系统上，在队列管理器的 `qm.ini` 文件的 XAResourceManager 节中指定交换机装入文件的详细信息。

第 55 页的图 7 是显示 XAResourceManager 条目的 AIX and Linux 系统样本。 必须将 `LogDir` 添加到 XA 打开字符串，以便将所有错误和跟踪信息记录到同一位置。

```
XAResourceManager:
  Name=myoracle
  SwitchFile=oraswit
  XAOpenString=Oracle_XA+Acc=P/myuser/mypasswd+SesTm=35+LogDir=/tmp+threads=true
  ThreadOfControl=THREAD
```

图 *7: AIX and Linux* 平台上 *Oracle* 的样本 *XAResourceManager* 条目

注:

1. 在 第 55 页的图 7 中， xa_open 字符串已与四个参数配合使用。 可以包括其他参数，如 Oracle 的文档中所述。

2. 使用 IBM MQ 参数 `ThreadOfControl=THREAD` 时，必须使用 XAResourceManager 节中的 Oracle 参数 `+threads=true` 。

有关 xa_open 字符串的更多信息，请参阅 *Oracle8 Server Application Developer 's Guide* 。

更改 Oracle 配置参数

对于队列管理器正在协调的每个 Oracle 数据库，必须查看最大会话数并设置数据库特权。 为此，请完成以下步骤:

查看最大会话数
您可能必须查看 LICENSE_MAX_SESSIONS 和 PROCESSES 设置以考虑属于队列管理器的进程所需的其他连接。 有关更多详细信息，请参阅 第 49 页的『安装和配置数据库产品』。

设置数据库特权
xa_open 字符串中指定的 Oracle 用户名必须具有访问 DBA_PENDING_TRANSACTIONS 视图的特权，如 Oracle 文档中所述。

可以使用以下示例命令来授予必需的特权:

```
grant select on DBA_PENDING_TRANSACTIONS to myuser;
```

Multi 配置 Informix

Informix 支持和配置信息。

完成以下步骤:

1. 确保已安装相应的 Informix 客户端 SDK:
   - 32 位队列管理器和应用程序需要 32 位 Informix 客户机 SDK。
   - 64 位队列管理器和应用程序需要 64 位 Informix 客户机 SDK。
2. 确保正确创建 Informix 数据库。
3. 检查环境变量设置。
4. 构建 Informix 交换机装入文件。
5. 添加资源管理器配置信息。

有关 IBM MQ 支持的 Informix 级别的当前列表，请参阅 系统要求 IBM MQ。

确保正确创建 Informix 数据库

必须创建要由 IBM MQ 队列管理器协调的每个 Informix 数据库，并指定 **log** 参数。 例如:

```
create database mydbname with log;
```

IBM MQ 队列管理器无法协调在创建时未指定 `log` 参数的 Informix 数据库。 如果队列管理器尝试协调在创建时未指定 `log` 参数的 Informix 数据库，那么对 Informix 的 xa_open 调用将失败，并生成大量 FFST 错误。

检查 Informix 环境变量设置

确保为队列管理器进程 **以及应用程序进程在** 中设置了 Informix 环境变量。 尤其是，在启动队列管理器 **之前** ，请始终设置以下环境变量:

INFORMIXDIR
Informix 产品安装的目录。

- 对于 32 位 AIX and Linux 应用程序，请使用以下命令:

```
export INFORMIXDIR=/opt/informix/32-bit
```

- 对于 64 位 AIX and Linux 应用程序，请使用以下命令:

```
export INFORMIXDIR=/opt/informix/64-bit
```

- 对于 Windows 应用程序，请使用以下命令:

```
set INFORMIXDIR=c:\informix
```

对于具有必须同时支持 32 位和 64 位应用程序的 64 位队列管理器的系统，您需要同时安装 Informix 32 位和 64 位客户机 SDK。 用于创建切换装入文件的样本 makefile `xaswit.mak` 还会设置两个产品安装目录。

INFORMIXSERVER
Informix 服务器的名称。 例如，在 AIX and Linux 系统上，使用:

```
export INFORMIXSERVER=hostname_1
```

在 Windows 系统上，使用:

```
set INFORMIXSERVER=hostname_1
```

ONCONFIG
Informix 服务器配置文件的名称。 例如，在 AIX and Linux 系统上，使用:

```
export ONCONFIG=onconfig.hostname_1
```

在 Windows 系统上，使用:

```
set ONCONFIG=onconfig.hostname_1
```

创建 Informix 交换机装入文件

要创建 Informix 切换装入文件，请使用样本文件 xaswit.mak， IBM MQ 提供该样本文件来为各种数据库产品构建切换装入文件。

Windows 在 Windows 系统上，您可以在目录 *MQ_INSTALLATION_PATH*`\tools\c\samples\xatm` 中找到 xaswit.mak 。 *MQ_INSTALLATION_PATH* 表示 IBM MQ 安装所在的高级目录。 要使用 Microsoft Visual C++ 创建 Informix 切换装入文件，请使用:

```
nmake /f xaswit.mak infswit.dll
```

生成的交换机文件放置在 `C:\Program Files\IBM\MQ\exits` 中。

Linux UNIX 您可以在目录 *MQ_INSTALLATION_PATH*`/samp/xatm` 中找到 xaswit.mak 。 *MQ_INSTALLATION_PATH* 表示 IBM MQ 安装所在的高级目录。

编辑 xaswit.mak 以 取消注释 与您正在使用的 Informix 版本相应的行。 然后使用以下命令执行 makefile:

```
make -f xaswit.mak infswit
```

生成的 32 位交换机装入文件放置在 `/var/mqm/exits` 中。

生成的 64 位交换机装入文件放置在 `/var/mqm/exits64` 中。

V 9.3.0 如果系统不支持 32 位编译，请使用仅 64 位目标:

```
make -f xaswit.mak infswit64
```

添加 Informix 的资源管理器配置信息

您必须修改队列管理器的配置信息， 以将 Informix 声明为全局工作单元中的参与者。 在 第 50 页的『向队列管理器添加配置信息』中更详细地描述了如何以此方式修改队列管理器的配置信息。

- 在 Windows 和 Linux (x86 和 x86-64 平台) 系统上，使用 IBM MQ Explorer。 在队列管理器属性面板中的 XA 资源管理器下指定切换装入文件的详细信息。
- 在所有其他系统上，在队列管理器的 qm.ini 文件的 XAResourceManager 节中指定交换机装入文件的详细信息。

第 58 页的图 8 是 UNIX 样本，显示 qm.ini XAResourceManager 条目，其中要协调的数据库名为 mydbname，此名称在 XAOpenString 中指定:

```
XAResourceManager:
  Name=myinformix
  SwitchFile=infswit
  XAOpenString=DB=mydbname@myinformixserver\;USER=myuser\;PASSWD=mypasswd
  ThreadOfControl=THREAD
```

图 *8: UNIX* 上 *Informix* 的样本 *XAResourceManager* 条目

注: 缺省情况下， UNIX 上的样本 xaswit.mak 将创建一个使用线程 Informix 库的交换机装入文件。 使用这些 "Informix 库时，必须确保 ThreadOfControl 设置为 THREAD。 在 "第 58 页的图 8 中，qm.ini 文件 XAResourceManagerstanza 属性 ThreadOfControl 设置为 THREAD。 指定 THREAD 时，必须使用 Informix 线程库和 IBM MQ 线程 API 库来构建应用程序。

XAOpenString 属性必须包含数据库名称，后跟 @ 符号，然后后跟 Informix 服务器名称。

要使用非线程 "Informix 库，必须确保 qm.ini 文件 XAResourceManagerstanza 属性 ThreadOfControl 设置为 PROCESS。 您还必须对样本 xaswit.mak:

1. 取消注释非线程交换机装入文件的生成。
2. 注释掉线程切换装入文件的生成。

Multi Sybase 配置

Sybase 支持和配置信息。

完成以下步骤:

1. 确保已安装 Sybase XA 库，例如通过安装 XA DTM 选项。
2. 检查环境变量设置。
3. 启用 Sybase XA 支持。
4. 创建 Sybase 交换机装入文件。
5. 添加资源管理器配置信息。

有关 IBM MQ 支持的 Sybase 级别的当前列表，请参阅 系统要求 IBM MQ。

检查 Sybase 环境变量设置

确保为队列管理器进程以及应用程序进程设置了 Sybase 环境变量。 尤其是，在启动队列管理器之前，请始终设置以下环境变量:

SYBASE

Linux AIX Sybase 产品安装的位置。 例如，在 AIX and Linux 系统上，使用:

```
export SYBASE=/sybase
```

Windows 在 Windows 系统上，使用:

```
set SYBASE=c:\sybase
```

SYBASE_OCS

已安装 Sybase 客户机文件的 SYBASE 下的目录。

Linux AIX 例如，在 AIX and Linux 系统上，使用:

```
export SYBASE_OCS=OCS-12_0
```

Windows 在 Windows 系统上，使用:

```
set SYBASE_OCS=OCS-12_0
```

启用 Sybase XA 支持

在 Sybase XA 配置文件 $SYBASE/$SYBASE_OCS/xa_config 中，为要更新的 Sybase 服务器的每个连接定义逻辑 Resource Manager (LRM)。 以下是 $SYBASE/$SYBASE_OCS/xa_config 的内容示例:

```
# The first line must always be a comment

[xa]

  LRM=lrmname
  server=servername
```

创建 Sybase 交换机装入文件

要创建 Sybase 切换装入文件，请使用 IBM MQ 随附的样本文件。

Windows 在 Windows 系统上，可以在目录 C:\Program Files\IBM\MQ\tools\c\samples\xatm 中找到 xaswit.mak 。 要使用 Microsoft Visual C++ 创建 Sybase 切换装入文件，请使用:

```
nmake /f xaswit.mak sybswit.dll
```

生成的交换机文件放置在 C:\Program Files\IBM\MQ\exits 中。

Linux UNIX 您可以在目录 *MQ_INSTALLATION_PATH*/samp/xatm 中找到 xaswit.mak 。 *MQ_INSTALLATION_PATH* 表示 IBM MQ 安装所在的高级目录。

编辑 xaswit.mak 以 取消注释 与您正在使用的 Sybase 版本相应的行。 然后使用以下命令执行 makefile:

```
make -f xaswit.mak sybswit
```

生成的 32 位交换机装入文件将放在 /var/mqm/exits 中。

生成的 64 位交换机装入文件放置在 /var/mqm/exits64 中。

V 9.3.0 如果系统不支持 32 位编译，请使用仅 64 位目标:

```
make -f xaswit.mak sybswit64
```

注: AIX 在 AIX 上，已修改样本 makefile ，如以下示例中所示，以便您可以选择不同的 SYBLINKFLAG64 值，具体取决于您是使用 Sybase 15 ESD#5 还是更低版本的 Sybase。

```
SYBLINKFLAGS32=-brtl
# The following line is for Sybase 15
#SYBLINKFLAGS64=-brtl
# The following line is for Sybase 16
SYBLINKFLAGS64=-bstatic -bdynamic
```

您需要对 makefile 进行的唯一更改是确保仅取消注释其中一个 SYBLINKFLAGS64 值。 缺省值为 Sybase 16 ，这是要用于 15 #ESD5 及更高版本的值。

生成的任何 XA 切换文件都将链接到 Sybase 的特定发行版，并且不得移动到其他平台。

如果更改了 Sybase 的级别，那么应该重建 XA 切换文件。

为 Sybase 添加资源管理器配置信息

您必须修改队列管理器的配置信息，以将 Sybase 声明为全局工作单元中的参与者。 第 50 页的『向队列管理器添加配置信息』中更详细地描述了如何修改配置信息。

- Windows Linux 在 Windows 和 Linux (x86 和 x86-64 平台) 系统上，使用 IBM MQ Explorer。在队列管理器属性面板中的 XA 资源管理器下指定切换装入文件的详细信息。
- 在所有其他系统上，在队列管理器的 `qm.ini` 文件的 XAResourceManager 节中指定交换机装入文件的详细信息。

Linux AIX 以下示例显示了 AIX and Linux 上 Sybase 的样本 XAResourceManager 条目，该条目使用与 Sybase XA 配置文件 `$SYBASE/$SYBASE_OCS/xa_config` 中的 lrmname LRM 定义相关联的数据库。 如果要记录 XA 函数调用，请包含日志文件名:

```
XAResourceManager:
  Name=mysybase
  SwitchFile=sybswit
  XAOpenString=-Uuser -Ppassword -Nlrmname -L/tmp/sybase.log -Txa
  ThreadOfControl=THREAD
```

将多线程程序与 Sybase 配合使用

如果使用多线程程序，且 "IBM MQ 全局工作单元包含对 Sybase 的更新，则必须使用 THREAD 作为 ThreadOfControl 参数的值。 还要确保将程序 (和交换机装入文件) 与线程安全 Sybase 库 (_r 版本) 链接。 使用 THREAD 值作为 ThreadOfControl）参数，如上例所示。

Multi 多个数据库配置

如果要配置队列管理器以便可以在全局工作单元中包含对多个数据库的更新，请为每个数据库添加 XAResourceManager 节。

如果数据库全部由同一数据库管理器管理，那么每个节定义一个单独的数据库。 每节都指定了相同的 *SwitchFile*，但 *XAOpenString* 的内容不同，因为它指定了要更新的数据库名称。 例如，"第 60 页的图 9 中显示的节将队列管理器与 "Db2 数据库 *MQBankDB* 和 *MQFeeDB* 一起配置到 "AIX and Linux 系统上。

要点: 不能有多个节指向同一数据库。 此配置在任何情况下都不起作用，如果您尝试此配置，那么它将失败。

您将收到格式为 `when the MQ code makes its second xa_open call in any process in this environment, the database software fails the second xa_open with a -5 error, XAER_INVAL` 的错误。

```
XAResourceManager:
Name=DB2 MQBankDB
SwitchFile=db2swit
XAOpenString=MQBankDB

XAResourceManager:
Name=DB2 MQFeeDB
SwitchFile=db2swit
XAOpenString=MQFeeDB
```

图 *9:* 多个 *Db2* 数据库的样本 *XAResourceManager* 条目

如果要更新的数据库由不同的数据库管理器管理，请为每个数据库管理器添加 XAResourceManager 节。 在这种情况下，每个节指定不同的 *SwitchFile*。 例如，如果 *MQFeeDB* 由 Oracle 而不是 Db2 管理，请在 AIX and Linux 系统上使用以下节:

```
XAResourceManager:
Name=DB2 MQBankDB
SwitchFile=db2swit
XAOpenString=MQBankDB

XAResourceManager:
Name=Oracle MQFeeDB
SwitchFile=oraswit
XAOpenString=Oracle_XA+Acc=P/myuser/mypassword+SesTm=35+LogDir=/tmp/ora.log+DB=MQFeeDB
```

图 *10: Db2* 和 *Oracle* 数据库的样本 *XAResourceManager* 条目

原则上，可以使用单个队列管理器配置的数据库实例数没有限制。

注: 有关支持在全局工作单元中的多个数据库更新中包含 Informix 数据库的信息，请查看产品自述文件。

Multi 安全性注意事项

在 XA 模型下运行数据库的注意事项。

提供以下信息仅供参考。 在所有情况下，请参阅数据库管理器随附的文档，以确定在 XA 模型下运行数据库的安全性影响。

应用程序进程表示使用 MQBEGIN 动词启动全局工作单元。 应用程序问题通过在 xa_open 入口点调用其客户机库代码来连接到所有参与数据库的第一个 MQBEGIN 调用。 所有数据库管理器都提供了在其 XAOpenString 中提供用户标识和密码的机制。 这是认证信息流动的唯一时间。

请注意，在 AIX and Linux 平台上，快速路径应用程序在进行 MQI 调用时必须使用有效用户标识 mqm 运行。

Multi 丢失与 XA 资源管理器的联系时的注意事项

队列管理器允许数据库管理器不可用。 这意味着您可以独立于数据库服务器启动和停止队列管理器。 恢复联系人后，队列管理器和数据库再同步。 您还可以使用 rsvmqtrn 命令来手动解析不确定的工作单元。

在正常操作中，完成配置步骤后仅需要最少量的管理。 由于队列管理器允许数据库管理器不可用，因此使管理作业更容易执行。 这尤其意味着:

- 队列管理器可以随时启动，而无需首先启动每个数据库管理器。
- 如果其中一个数据库管理器变为不可用，那么队列管理器不需要停止并重新启动。

这允许您独立于数据库服务器启动和停止队列管理器。

每当队列管理器与数据库之间失去联系时，它们都需要在两者再次变为可用时再同步。 再同步是完成涉及该数据库的任何不确定工作单元的过程。 通常，这会自动发生，而无需用户干预。 队列管理器要求数据库提供不确定的工作单元列表。 然后，它指示数据库落实或回滚其中每个不确定的工作单元。

当队列管理器启动时，它将与每个数据库再同步。 当单个数据库变为不可用时，仅当队列管理器通知该数据库再次可用时，才需要对该数据库进行再同步。

当使用 MQBEGIN 启动新的全局工作单元时，队列管理器将自动恢复与先前不可用的数据库的联系。 它通过在数据库客户机库中调用 xa_open 函数来执行此操作。 如果此 xa_open 调用失败，那么 MQBEGIN 将返回完成代码 MQCC_WARNING 和原因码 MQRC_PARTICIPANT_NOT_AVAILABLE。 您可以稍后重试 MQBEGIN 调用。

请勿继续尝试涉及在 MQBEGIN 期间指示失败的数据库更新的全局工作单元。 将不存在与可通过其进行更新的数据库的连接。 唯一的选项是结束程序，或者定期重试 MQBEGIN ，以希望数据库可能再次可用。

或者，可以使用 rsvmqtrn 命令来显式解析所有不确定的工作单元。

Multi 不确定的工作单元

如果在指示数据库管理器准备之后丢失了与队列管理器的联系，那么数据库可能具有不确定的工作单元。 在数据库服务器从队列管理器接收结果 (落实或回滚) 之前，它需要保留与更新关联的数据库锁定。

由于这些锁定会阻止其他应用程序更新或读取数据库记录，因此需要尽快进行再同步。

如果由于某种原因，您无法等待队列管理器自动与数据库再同步，那么可以使用数据库管理器提供的工具来手动落实或回滚数据库更新。 在 *X/Open* 分布式事务处理*: XA* 规范中，这称为作出 启发式 决策。 仅将其用作最后手段，因为可能会损害数据完整性; 例如，当所有其他参与者已落实其更新时，您可能会错误地回滚数据库更新。

最好重新启动队列管理器，或者在重新启动数据库后使用 `rsvmqtrn` 命令来启动自动再同步。

Multi 使用 *dspmqtrn* 命令显示未完成的工作单元

您可以使用带有 **-i** 参数的 **dspmqtrn** 命令来显示内部发起的不确定事务。

当数据库管理器不可用时，可以使用 **dspmqtrn** 命令来检查涉及该数据库的未完成全局工作单元的状态。

dspmqtrn 命令仅显示其中一个或多个参与者处于不确定状态的工作单元。 参与者正在等待队列管理器的决策，以落实或回滚已准备的更新。

对于这些全局工作单元中的每个工作单元，将在 **dspmqtrn** 的输出中显示每个参与者的状态。 如果工作单元未更新特定资源管理器的资源，那么不会显示该工作单元。

关于不确定的工作单元，据说资源管理器已执行下列其中一项操作:

准备
资源管理器已准备好落实其更新。

已落实
资源管理器已落实其更新。

已回滚
资源管理器已回滚其更新。

参与
资源管理器是参与者，但尚未准备，落实或回滚其更新。

重新启动队列管理器时，它会要求每个具有 XAResourceManager 节的数据库获取其不确定全局工作单元的列表。 如果数据库尚未重新启动，或者在其他情况下不可用，那么队列管理器无法向数据库交付这些工作单元的最终结果。 当数据库再次可用时，会将不确定工作单元的结果第一次传递到数据库。

在这种情况下，报告数据库管理器处于 `prepared` 状态，直到发生再同步为止。

每当 **dspmqtrn** 命令显示不确定的工作单元时，它都会首先列出可能参与的所有可能的资源管理器。 为这些资源分配了唯一标识 `RMId`，在报告其与不确定工作单元相关的状态时，将使用该标识来代替资源管理器的 名称 。

样本 dspmqtrn 输出 显示发出以下命令的结果:

```
dspmqtrn -m MY_QMGR
```

```
AMQ7107: Resource manager 0 is MQSeries.
AMQ7107: Resource manager 1 is DB2 MQBankDB.
AMQ7107: Resource manager 2 is DB2 MQFeeDB.

AMQ7056: Transaction number 0,1.
    XID: formatID 5067085, gtrid_length 12, bqual_length 4
         gtrid [3291A50600000201374657374]
         bqual [00000001]
AMQ7105: Resource manager 0 has committed.
AMQ7104: Resource manager 1 has prepared.
AMQ7104: Resource manager 2 has prepared.
```

其中 事务号 是可与 **rsvmqtrn** 命令配合使用的事务的标识。 请参阅 AMQ7xxx: IBM MQ 产品消息 以获取更多信息。 *XID* 变量是 *X/Open XA* 规范 的一部分; 有关此规范的最新信息，请参阅: https://publications.opengroup.org/c193。

图 *11:* 样本 *dspmqtrn* 输出

Sample dspmqtrn output 中的输出显示有三个资源管理器与队列管理器相关联。 第一个是资源管理器 0，它是队列管理器本身。 其他两个资源管理器实例是 MQBankDB 和 MQFeeDB Db2 数据库。

此示例仅显示单个不确定工作单元。 将针对所有三个资源管理器发出一条消息，这意味着已对工作单元中的队列管理器和两个 Db2 数据库进行了更新。

对队列管理器 0 进行的更新已 **落实**。 对 Db2 数据库的更新处于 prepared 状态，这意味着在调用 Db2 以落实对 MQBankDB 和 MQFeeDB 数据库的更新之前，它必须已变为不可用。

不确定工作单元具有称为 XID (事务标识) 的外部标识。 这是队列管理器提供给 Db2 的数据片段，用于标识其在全局工作单元中的部分。

有关 **dspmqtrn** 的更完整信息，以及另一个使用示例 (dspmqtrn -m *<qmgrname>* -q -a)，请参见 **dspmqtrn**.

Multi 使用 *rsvmqtrn* 命令解析未完成的工作单元

队列管理器和 Db2 再同步时，未完成的工作单元完成。

第 62 页的图 11 中显示的输出显示尚未将落实决策传递到两个 Db2 数据库的单个不确定工作单元。

要完成此工作单元，队列管理器和 Db2 需要在下次 Db2 变为可用时再同步。 队列管理器使用新工作单元的启动作为重新与 Db2 联系的机会。 或者，您可以指示队列管理器使用 **rsvmqtrn** 命令显式再同步。

在重新启动 Db2 后立即执行此操作，以便尽快释放与不确定工作单元相关联的任何数据库锁定。 使用 -a 选项，它指示队列管理器解析所有不确定的工作单元。 在以下示例中， Db2 已重新启动，因此队列管理器可以解析不确定的工作单元:

```
> rsvmqtrn -m MY_QMGR -a
Any in-doubt transactions have been resolved.
```

Multi 混合结果和错误

虽然队列管理器使用两阶段落实协议，但这不会完全消除某些工作单元完成混合结果的可能性。 在此情况下，一些参与者会落实其更新，而一些参与者会回退其更新。

以混合结果完成的工作单元会产生严重影响，因为本应更新为单个工作单元的共享资源不再处于一致状态。

主要在对工作单元做出启发式决策时导致混合结果，而不是允许队列管理器解决不确定的工作单元本身。 此类决策不在队列管理器的控制范围内。

每当队列管理器检测到混合结果时，它都会生成 FFST 信息并在其错误日志中记录故障，并显示以下两条消息之一:

- 如果数据库管理器回滚而不是落实:

```
AMQ7606 A transaction has been committed but one or more resource
        managers have rolled back.
```

- 如果数据库管理器落实而不是回滚:

```
AMQ7607 A transaction has been rolled back but one or more resource
        managers have committed.
```

进一步的消息标识试探性损坏的数据库。 然后，您负责在本地恢复受影响数据库的一致性。 这是一个复杂的过程，您需要首先隔离错误落实或回滚的更新，然后手动撤销或重做数据库更改。

Multi 更改配置信息

在成功启动队列管理器以协调全局工作单元之后，请勿更改任何资源管理器配置信息。

如果需要更改配置信息，您可以随时执行此操作，但这些更改直到重新启动队列管理器之后才会生效。

如果除去数据库的资源管理器配置信息，那么将有效除去队列管理器与该数据库管理器联系的能力。

从不 更改任何资源管理器配置信息中的 名称 属性。 此属性向队列管理器唯一地标识该数据库管理器实例。 如果更改此唯一标识，那么队列管理器假定已除去数据库并添加了全新的实例。 队列管理器仍将未完成的工作单元与旧的 名称相关联，这可能使数据库处于不确定状态。

Multi 除去数据库管理器实例

如果需要从配置中永久除去数据库，请确保在重新启动队列管理器之前该数据库未处于不确定状态。

数据库产品提供用于列出不确定事务的命令。 如果存在任何不确定事务，请首先允许队列管理器与数据库再同步。 通过启动队列管理器来执行此操作。 您可以使用 **rsvmqtrn** 命令或数据库自己的命令来查看不确定的工作单元，以验证是否已执行再同步。 一旦您确信已进行再同步，请结束队列管理器并除去数据库的配置信息。

如果未能观察到此过程，那么队列管理器仍会记住涉及该数据库的所有不确定工作单元。 每次重新启动队列管理器时，都会发出一条警告消息 AMQ7623。 如果再也不打算在队列管理器中配置该数据库，可使用 **rsvmqtrn** 命令的-r 选项指示队列管理器忘记该数据库参与其可疑事务。 仅当所有参与者都已完成不确定事务时，队列管理器才会忘记此类事务。

有时可能需要临时除去某些资源管理器配置信息。 在 AIX and Linux 系统上，最好通过注释掉节来实现此目标，以便稍后可以轻松恢复此节。 如果队列管理器每次与特定数据库或数据库管理器联系时都发生错误，那么您可能决定执行此操作。 临时除去相关的资源管理器配置信息允许队列管理器启动涉及所有其他参与者的全局工作单元。 以下是已注释掉的 XAResourceManager 节的示例:

```
# This database has been temporarily removed
#XAResourceManager:
#   Name=mydb2
#   SwitchFile=db2swit
#   XAOpenString=mydbname,myuser,mypassword,toc=t
#   ThreadOfControl=THREAD
```

图 *12:* 已注释掉 *AIX and Linux* 系统上的 *XAResourceManager* 节

在 Windows 系统上，使用 IBM MQ Explorer 来删除有关数据库管理器实例的信息。 在恢复时，请非常小心地在 名称 字段中输入正确的名称。 如果错误输入名称，那么可能会迁到不确定问题，如 第 63 页的『更改配置信息』中所述。

Multi XA 动态注册

XA 规范提供了一种减少事务管理器对资源管理器进行的 xa_* 调用数的方法。 此优化称为 动态注册。

Db2 支持动态注册。 其他数据库可能支持此操作; 请参阅数据库产品的文档以获取详细信息。

为什么动态注册优化有用? 在应用程序中，某些全局工作单元可能包含对数据库表的更新; 其他工作单元可能不包含此类更新。 如果未对数据库的表进行持久更新，那么无需将该数据库包含在 MQCMIT 期间发生的落实协议中。

无论您的数据库是否支持动态注册，您的应用程序都会在 IBM MQ 连接上的第一个 MQBEGIN 调用期间调用 xa_open 。 它还会在后续 MQDISC 调用上调用 xa_close 。 后续 XA 调用的模式取决于数据库是否支持动态注册:

如果您的数据库不支持动态注册 ...

每个全局工作单元都涉及 IBM MQ 代码对数据库客户机库进行的若干 XA 函数调用，而无论您是否在工作单元中对该数据库的表进行了持久更新。 其中包括:

- 应用程序进程中的 xa_start 和 xa_end 。 这些用于声明全局工作单元的开始和结束。
- 来自队列管理器代理进程 amqzlaa0 的 xa_prepare， xa_commit 和 xa_rollback 。 这些用于交付全局工作单元的结果: 落实或回滚决策。

此外，队列管理器代理进程还会在第一个 MQBEGIN 期间调用 xa_open 。

如果您的数据库支持动态注册 ...

IBM MQ 代码仅执行必需的那些 XA 函数调用。 对于 **未** 涉及数据库资源的持久更新的全局工作单元， **不会** 对数据库进行 XA 调用。 对于 **已** 涉及此类持久更新的全局工作单元，调用的目的是:

- 来自应用程序进程的 xa_end ，用于声明全局工作单元的结束。
- 来自队列管理器代理进程 amqzlaa0 的 xa_prepare， xa_commit 和 xa_rollback 。 这些用于交付全局工作单元的结果: 落实或回滚决策。

要使动态注册生效，数据库必须能够在执行持久更新时告知 IBM MQ 它希望包含在当前全局工作单元中。 IBM MQ 提供了用于此目的的 ax_reg 函数。

在应用程序进程中运行的数据库的客户机代码将找到 ax_reg 函数并对其进行调用，以 动态注册 它在当前全局工作单元中执行持久工作的事实。 作为对此 ax_reg 调用的响应， IBM MQ 记录了数据库已参与的记录。 如果这是此 IBM MQ 连接上的第一个 ax_reg 调用，那么队列管理器代理进程将调用 xa_open。

数据库客户机代码在进程中运行时 (例如，在 SQL UPDATE 调用期间或数据库的客户机 API 中的任何调用) 发出此 ax_reg 调用。

Multi 错误条件

在 XA 动态注册中，可能会在队列管理器中发生混淆故障。

一个常见示例是，如果在启动队列管理器之前忘记正确设置数据库环境变量，那么队列管理器对 xa_open 的调用将失败。 不能使用全局工作单元。

要避免此情况，请确保在启动队列管理器之前设置了相关环境变量。 查看数据库产品的文档以及 第 52 页的『配置 Db2』， 第 54 页的『配置 Oracle』和 第 58 页的『Sybase 配置』中提供的建议。

对于所有数据库产品，队列管理器在队列管理器启动时调用 xa_open 一次，作为恢复会话的一部分 (如 第 61 页的『丢失与 XA 资源管理器的联系时的注意事项』 中所述)。 如果未正确设置数据库环境变量，但这不会导致队列管理器无法启动，那么此 xa_open 调用将失败。 这是因为数据库客户机库使用相同的 xa_open 错误代码来指示数据库服务器不可用。 IBM MQ 不会将此视为严重错误，因为队列管理器必须能够开始在涉及该数据库的全局工作单元之外继续处理数据。

在 IBM MQ 连接上的第一个 MQBEGIN 期间 (如果未使用动态注册) 或数据库客户机代码对 IBM MQ-provided ax_reg 函数的调用期间 (如果正在使用动态注册)，将从队列管理器进行对 xa_open 的后续调用。

任何错误条件 (有时为 FFST 报告) 的 **计时** 取决于您是否正在使用动态注册:

- 如果您正在使用动态注册，那么 MQBEGIN 调用可能会成功，但 SQL UPDATE (或类似) 数据库调用将失败。
- 如果您未使用动态注册，那么 MQBEGIN 调用将失败。

确保在应用程序和队列管理器进程中正确设置环境变量。

Multi 汇总 *XA* 调用

以下是由于控制全局工作单元的各种 MQI 调用而对数据库客户机库中的 XA 函数进行的调用的列表。 这不是 XA 规范中描述的协议的完整描述; 它作为简要概述提供。

请注意， xa_start 和 xa_end 调用始终由应用程序进程中的 IBM MQ 代码调用，而 xa_prepare, xa_commit 和 xa_rollback 始终从队列管理器代理程序进程 amqzlaa0 调用。

此表中显示的 xa_open 和 xa_close 调用都是从应用程序进程进行的。 队列管理器代理进程在 第 65 页的『错误条件』中描述的情况下调用 xa_open 。

表 *4: XA* 函数调用摘要

MQI 调用	**使用动态注册进行的 XA 调用**	**在未进行动态注册的情况下进行的 XA 调用**
第一个 MQBEGIN	xa_open	xa_open xa_start
后续 MQBEGIN	无 XA 调用	xa_start
MQCMIT (在当前全局工作单元期间调用 **而不调用** ax_reg)	无 XA 调用	xa_end xa_prepare xa_commit xa_rollback

表 4: XA 函数调用摘要 (继续)		
MQI 调用	**使用动态注册进行的 XA 调用**	**在未进行动态注册的情况下进行的 XA 调用**
MQCMIT (，**在当前全局工作单元期间调用** ax_reg)	xa_end xa_prepare xa_commit xa_rollback	不适用。 不会以非动态方式调用 ax_reg。
MQBACK (在当前全局工作单元期间调用 **而不调用** ax_reg)	无 XA 调用	xa_end xa_rollback
MQBACK (，**在当前全局工作单元期间调用** ax_reg)	xa_end xa_rollback	不适用。 不会以非动态方式调用 ax_reg。
MQDISC ，其中首先调用了 MQCMIT 或 MQBACK。 如果不是，那么首先在 MQDISC 期间执行 MQCMIT 处理。	xa_close	xa_close

注意:

1. 对于 MQCMIT ，如果 xa_prepare 成功，那么将调用 xa_commit 。 否则，将调用 xa_rollback 。

方案 2: 其他软件提供协调

在方案 2 中，外部事务管理器协调全局工作单元，在事务管理器的 API 控制下启动并落实这些工作单元。 MQBEGIN、MQCMIT 和 MQBACK 动词均不可用。

本部分描述了此场景，包括:

- 第 66 页的『外部同步点协调』
- 第 68 页的『使用 CICS』
- 第 71 页的『使用 Microsoft Transaction Server (COM +)』

外部同步点协调

全局工作单元还可以由符合 X/Open XA 的外部事务管理器进行协调。 在此， IBM MQ 队列管理器参与但不协调工作单元。

由外部事务管理器协调的全局工作单元中的控制流如下所示:

1. 应用程序告知外部同步点协调程序 (例如， TXSeries) 它想要启动事务。
2. 同步点协调程序告知已知资源管理器 (例如， IBM MQ) 有关当前事务的信息。
3. 应用程序向与当前事务关联的资源管理器发出调用。 例如，应用程序可以向 IBM MQ 发出 MQGET 调用。
4. 应用程序向外部同步点协调程序发出落实或回退请求。
5. 同步点协调程序通过向每个资源管理器 (通常使用两阶段落实协议) 发出相应的调用来完成事务。

在 系统要求 IBM MQ 上定义了可以为 IBM MQ 参与的事务提供两阶段落实过程的受支持外部同步点协调程序级别。

本节的其余部分描述了如何启用外部工作单元。

IBM MQ XA 开关结构

参与外部协调工作单元的每个资源管理器都必须提供 XA 切换结构。 此结构定义资源管理器的功能以及要由同步点协调程序调用的功能。

IBM MQ 提供了此结构的两个版本:

- 用于静态 XA 资源管理的 *MQRMIXASwitch*
- *MQRMIXASwitchDynamic* 用于动态 XA 资源管理

请参阅事务管理器文档以确定是使用静态还是动态资源管理接口。 只要事务管理器支持，我们就建议您使用动态 XA 资源管理。

某些 64 位事务管理器将 XA 规范中的 *long* 类型视为 64 位，而某些事务管理器将其视为 32 位。 IBM MQ 支持两种模型:

- 如果事务管理器是 32 位，或者事务管理器是 64 位，但将 `long` 类型视为 32 位，请使用 第 67 页的表 5 中列出的交换机装入文件。
- 如果事务管理器是 64 位，并且将 `long` 类型视为 64 位，请使用 第 67 页的表 6 中列出的交换机装入文件。

某些 64 位事务管理器将 `long` 类型视为 64 位。 已知以下 64 位事务管理器需要备用 64 位交换机装入文件:

- Tuxedo

如果您不确定事务管理器使用的模型，请查阅事务管理器文档。

表 *5: XA* 切换装入文件名

平台	交换机装入文件名 (服务器)	交换机装入文件名 (扩展事务客户机)
Windows	*mqmxa.dll*	*mqcxa.dll*
AIX (非线程)	*libmqmxa.a*	*libmqcxa.a*
AIX (线程)	*libmqmxa_r.a*	*libmqcxa_r.a*
Linux (非线程)	*libmqmxa.so*	*libmqcxa.so*
Linux (线程)	*libmqmxa_r.so*	*libmqcxa_r.so*

表 *6:* 备用 *64* 位 *XA* 切换装入文件名

平台	交换机装入文件名 (服务器)	交换机装入文件名 (扩展事务客户机)
AIX (非线程)	*libmqmxa64.a*	*libmqcxa64.a*
AIX (线程)	*libmqmxa64_r.a*	*libmqcxa64_r.a*
Linux (非线程)	*libmqmxa64.so*	*libmqcxa64.so*
Linux (线程)	*libmqmxa64_r.so*	*libmqcxa64_r.so*

某些外部同步点协调程序 (非 CICS) 要求参与工作单元的每个资源管理器在 XA 开关结构的名称字段中提供其名称。 IBM MQ 资源管理器名称为 MQSeries_XA_RMI。

同步点协调程序定义 IBM MQ XA 切换结构如何链接到同步点协调程序。 第 68 页的『使用 CICS』 中提供了有关将 IBM MQ XA 交换机结构与 CICS 链接的信息。 有关将 IBM MQ XA 开关结构与其他符合 XA 的同步点协调程序链接的信息，请参阅随这些产品提供的文档。

以下注意事项适用于将 IBM MQ 与所有符合 XA 的同步点协调程序配合使用:

- 期望事务管理器库代码 (作为应用程序员已调用的 API 的一部分运行) 在调用 MQCONN 之前的某个时间点将 **xa_open** 调用到 IBM MQ 中。

必须在执行 MQCONN 调用的同一线程上执行 **xa_open** 调用。 此需求的原因是 XA 规范要求使用线程来暗示上下文。

请注意，这是样本程序 amqstxsx.c 中采用的方法。 此样本程序假定 **xa_open** 调用在其函数 tpopen 中从事务管理器的库代码进入 IBM MQ。

如果在同一线程上没有 **xa_open** 调用，那么在 MQCONN 调用之前， IBM MQ 队列管理器连接将不会与 XA 上下文相关联。

有关更多信息，请参阅 MQCTL。

- 同步点协调程序在任何 xa_open 调用上传递的 xa_info 结构包含 IBM MQ 队列管理器的名称。 该名称采用与传递给 MQCONN 调用的队列管理器名称相同的格式。 如果在 xa_open 调用上传递的名称为空，那么将使用缺省队列管理器。

  或者， xa_info 结构可以包含 *TPM* 和 *AXLIB* 参数的值。 *TPM* 参数指定正在使用的事务管理器。 有效值为 CICS， TUXEDO 和 ENCINA。 *AXLIB* 参数指定包含事务管理器的 ax_reg 和 ax_unreg 函数的库的名称。 有关这些参数的更多信息，请参阅 配置扩展事务客户机。 如果 xa_info 结构包含其中任一参数，那么除非正在使用缺省队列管理器，否则将在 *QMNAME* 参数中指定队列管理器名称。
- 一次只能有一个队列管理器参与由外部同步点协调程序实例协调的事务。 同步点协调程序有效连接到队列管理器，并且遵循一次仅支持一个连接的规则。
- 所有包含对外部同步点协调程序的调用的应用程序都只能连接到参与由外部协调程序管理的事务的队列管理器 (因为它们已有效地连接到该队列管理器)。 但是，此类应用程序在退出之前必须发出 MQCONN 调用以获取连接句柄和 MQDISC 调用。
- 具有由外部同步点协调程序协调的资源更新的队列管理器必须在外部同步点协调程序之前启动。 同样，同步点协调程序必须在队列管理器之前结束。
- 如果外部同步点协调程序异常终止，请在重新启动同步点协调程序之前停止并重新启动队列管理器，以确保正确解决发生故障时未落实的任何消息传递操作。

使用 CICS

CICS 是 TXSeries 的其中一个元素。

在 系统要求 IBM MQ 信息中定义了 XA 兼容 (并使用两阶段落实过程) 的 TXSeries 版本。

IBM MQ 还支持其他事务管理器。 请参阅 系统要求 IBM MQ Web 页面，以获取有关受支持软件的信息的链接。

两阶段落实过程的需求

将 CICS 两阶段落实过程与 IBM MQ 配合使用时，两阶段落实过程的需求。 这些要求不适用于 z/OS。

请注意以下需求：

- IBM MQ 和 CICS 必须位于同一物理机器上。
- IBM MQ 在 IBM MQ MQI client 上不支持 CICS 。
- 必须先启动队列管理器，并在 XAD 资源定义节中指定其名称，**然后再** 尝试启动 CICS。 如果将 IBM MQ 的 XAD 资源定义节添加到 CICS 区域，那么未能执行此操作将阻止您启动 CICS 。
- 一次只能从单个 CICS 区域访问一个 IBM MQ 队列管理器。
- CICS 事务必须先发出 MQCONN 请求，然后才能访问 IBM MQ 资源。 MQCONN 调用必须指定在 CICS 区域的 XAD 资源定义节的 XAOpen 条目上指定的 IBM MQ 队列管理器的名称。 如果此条目为空，那么 MQCONN 请求必须指定缺省队列管理器。
- 访问 IBM MQ 资源的 CICS 事务必须在返回到 CICS 之前从该事务发出 MQDISC 调用。 未能执行此操作可能意味着 CICS 应用程序服务器仍处于连接状态，从而使队列处于打开状态。 此外，如果未安装任务终止出口 (请参阅 第 71 页的『样本任务终止出口』)，那么 CICS 应用程序服务器稍后可能会异常结束，可能在后续事务期间异常结束。
- 必须确保 CICS 用户标识 (cics) 是 mqm 组的成员，以便 CICS 代码具有调用 IBM MQ 的权限。

  对于在 CICS 环境中运行的事务，队列管理器会调整其授权和确定上下文的方法，如下所示：

- 队列管理器将查询 CICS 用于运行事务的用户标识。 这是对象权限管理器检查的用户标识，用于上下文信息。
- 在消息上下文中，应用程序类型为 MQAT_CICS。
- 将从 CICS 事务名称复制上下文中的应用程序名称。

ALW 常规 *XA* 支持

提供了 XA 切换装入模块，使您能够将 CICS 与 IBM MQ for AIX, Linux, and Windows 系统链接。 此外，还提供了样本源代码文件，使您能够为其他事务消息开发 XA 交换机。 IBM i 常规 XA 在 IBM i 上不受支持。

提供的交换机装入模块的名称如下所示:

表 *7: CICS* 应用程序的基本代码 *:XA* 初始化例程	
C（源代码）	**C (exec)-将下列其中一项添加到您的 XAD.Stanza**
amqzscix.c	AIX amqzsc- TXSeries (对于 AIX 5.1)
amqzscin.c	Windows mqmc4swi - TXSeries for Windows 5.1

Multi 构建用于 *TXSeries for Multiplatforms* 的库

构建用于 TXSeries for Multiplatforms 的库时，请使用此信息。

预构建的交换机装入文件 是可与 CICS 程序配合使用的共享库 (在 Windows 系统上称为 *DLL*) ，这些程序需要使用 XA 协议的 2 阶段落实事务。 这些预构建库的名称在表 CICS 应用程序的基本代码 :XA 初始化例程 中。 以下目录中还提供了样本源代码:

表 *8: AIX, Linux, and Windows* 操作系统上的安装目录		
平台	**目录**	**源文件**
Linux AIX AIX and Linux	*MQ_INSTALLATION_PATH*/samp/	amqzscix.c
Windows Windows	*MQ_INSTALLATION_PATH*\Tools\c \ 样本	amqzscin.c

其中 *MQ_INSTALLATION_PATH* 是 IBM MQ 的安装目录。

注: 必须使用针对交换机装入文件所使用的 IBM MQ 版本构建的 CICS 交换机装入文件。

要从样本源构建交换机装入文件，请遵循适合于您的操作系统的指示信息:

AIX AIX

发出以下命令:

```
export MQM_HOME=/usr/mqm
echo "amqzscix" > tmp.exp
xlc_r $MQM_HOME/samp/amqzscix.c -I/usr/lpp/cics/include -I$MQM_HOME/inc -e amqzscix -bE:tmp.exp -bM:SRE
    -o amqzsc /usr/lpp/cics/lib/regxa_swxa.o -L$MQM_HOME/lib -L/usr/lpp/cics/lib -lcicsrt -lEncina
    -lEncServer -lpthreads -lsarpc -lmqmcics_r -lmqmxa_r -lmqzi_r -lmqmcs_r
rm tmp.exp
```

Linux Linux 平台

发出以下命令:

```
gcc -m32 -shared -fPIC -o amqzscix amqzscix.c
    \-IMQ_INSTALLATION_PATH/inc -I CICS_INSTALLATION_PATH/include
```

```
\-LMQ_INSTALLATION_PATH/lib -Wl,-rpath=MQ_INSTALLATION_PATH/lib
\-Wl,-rpath=/usr/lib -Wl,-rpath-link,/usr/lib -Wl,--no-undefined
-Wl,--allow-shlib-undefined \-L CICS_LIB_PATH/regxa_swxa.o \-lpthread -ldl -lc
-shared -lmqzi_r -lmqmxa_r -lmqmcics_r -ldl -lc
```

Windows

请按照以下步骤操作：

1. 使用 cl 命令通过至少编译以下变量来构建 amqzscin.obj：

   ```
   cl.exe -c -I EncinaPath\include -I MQ_INSTALLATION_PATH\include -Gz -LD amqzscin.c
   ```

2. 创建名为 mqmc1415.def 的模块定义文件，其中包含以下行：

   ```
   LIBRARY MQMC4SWI
   EXPORTS
   CICS_XA_Init
   ```

3. 使用 **lib** 命令通过至少使用以下选项来构建导出文件和导入库：

   ```
   lib -def:mqmc4swi.def -out:mqmc4swi.lib
   ```

   如果 lib 命令成功，那么还会构建 mqmc4swi.exp 文件。

4. 使用链接命令通过至少使用以下选项来构建 mqmc4swi.dll：

   ```
   link.exe -dll -nod -out:mqmc4swi.dll
       amqzscin.obj CicsPath\lib\regxa_swxa.obj
       mqmc4swi.exp mqmcics4.lib
       CicsPath\lib\libcicsrt.lib
       DcePath\lib\libdce.lib DcePath\lib\pthreads.lib
       EncinaPath\lib\libEncina.lib
       EncinaPath\lib\libEncServer.lib
       msvcrt.lib kernel32.lib
   ```

IBM MQ XA 支持和 Tuxedo

AIX, Linux, and Windows 系统上的 IBM MQ 可以在 xa_start 中无限期地阻止 Tuxedo 协调的 XA 应用程序。

仅当 Tuxedo 在单个全局事务中协调的两个或多个进程尝试使用同一事务分支标识 (XID) 访问 IBM MQ 时，才会发生此情况。 如果 Tuxedo 为全局事务中的每个进程提供不同的 XID 以用于 IBM MQ，那么不会发生此情况。

为避免此问题，请在 Tuxedo 服务器组中的单个全局事务标识 (gtrid) 下配置 Tuxedo 中访问 IBM MQ 的每个应用程序。 同一服务器组中的进程在代表单个 gtrid 访问资源管理器时使用相同的 XID，因此容易在 IBM MQ 中的 xa_start 中受到阻止。 不同服务器组中的进程在访问资源管理器时使用单独的 XID，因此不必在 IBM MQ 中序列化其事务工作。

启用 *CICS* 两阶段落实过程

要使 CICS 能够使用两阶段落实过程来协调包含 MQI 调用的事务，请将 CICS XAD 资源定义节条目添加到 CICS 区域。 请注意，本主题不适用于 z/OS。

以下是为 IBM MQ for Windows 添加 XAD 节条目的示例，其中 *Drive* 是安装了 IBM MQ 的驱动器 (例如 D:)。

```
cicsadd -cxad -rcics_region \
     ResourceDescription="MQM XA Product Description" \
     SwitchLoadFile="Drive:\Program Files\IBM\IBM MQ\bin\mqmc4swi.dll" \
     XAOpen=queue_manager_name
```

对于扩展事务客户机，请使用切换装入文件 mqcc4swi.dll。

以下是为 IBM MQ for AIX or Linux 系统添加 XAD 节条目的示例，其中 *MQ_INSTALLATION_PATH* 表示安装了 IBM MQ 的高级目录：

```
cicsadd -cxad -rcics_region \
      ResourceDescription="MQM XA Product Description" \
      SwitchLoadFile="MQ_INSTALLATION_PATH/lib/amqzsc" \
      XAOpen=queue_manager_name
```

对于扩展事务客户机，请使用交换机装入文件 amqczsc。

请参阅 CICS 文档以获取有关使用 **cicsadd** 命令的信息。

对 IBM MQ 的调用可以包含在 CICS 事务中，并且将根据 CICS 的指示落实或回滚 IBM MQ 资源。 此支持不可用于客户机应用程序。

您必须从 CICS 事务发出 MQCONN 以访问 IBM MQ 资源，然后在退出时发出相应的 MQDISC 。

启用 *CICS* 用户出口

CICS 用户出口 点 (通常称为 用户出口) 是 CICS 模块中的一个位置，在此位置 CICS 可以将控制权转移到您已编写的程序 (用户出口 程序)，在此位置 CICS 可以在出口程序完成其工作时恢复控制权。

在使用 CICS 用户出口之前，请阅读适用于您平台的 *CICS Administration Guide* 。

样本任务终止出口

IBM MQ 为 CICS 任务终止出口提供样本源代码。

样本源代码位于以下目录中:

表 *9: CICS* 任务终止出口		
平台	**目录**	**源文件**
AIX and Linux 系统	*MQ_INSTALLATION_PATH*/samp	amqzscgx.c
Windows	*MQ_INSTALLATION_PATH*\Tools\c \ 样本	amqzscgn.c

MQ_INSTALLATION_PATH 表示 IBM MQ 安装所在的高级目录。

样本任务终止出口的构建指示信息包含在每个源文件中的注释中。

此出口由 CICS 在正常和异常任务终止时 (在执行任何同步点之后) 调用。 在出口程序中不允许可恢复的工作。

这些函数仅在 CICS 版本支持 XA 接口的 IBM MQ 和 CICS 上下文中使用。 CICS 将这些库称为 programs 或 user exits。

CICS 具有多个用户出口，并且 amqzscgx (如果使用) 在 CICS 上定义并启用为 Task termination user exit (UE014015)，即出口号 15。

当 CICS 调用任务终止出口时， CICS 已通知 IBM MQ 该任务的终止状态，并且 IBM MQ 已执行相应的操作 (落实或回滚)。 出口执行的所有操作都是发出 MQDISC 以进行清除。

安装和配置 CICS 系统以使用任务终止出口的一个目的是保护系统免受发生故障的应用程序代码的某些后果的影响。 例如，如果 CICS 事务在未首先调用 MQDISC 的情况下异常结束，并且未安装任务终止出口，那么您可能会看到 (大约 10 秒内) CICS 区域的后续不可恢复故障。 这是因为将不会发布在 cicsas 进程中运行的 IBM MQ 运行状况线程，也不会为其提供清除和返回的时间。 症状可能是 cicsas 进程立即结束，已将 FFST 报告写入 /var/mqm/errors 或 Windows 上的等效位置。

使用 Microsoft Transaction Server (COM +)

COM + (Microsoft Transaction Server) 旨在帮助用户在典型的中间层服务器中运行业务逻辑应用程序。

有关重要信息，请参阅 Windows 上只能与主安装一起使用的功能。

COM + 将工作划分为 活动，这些活动通常是业务逻辑的简短独立块，例如 将资金从帐户 *A* 转移到帐户 *B*。 COM + 在很大程度上依赖于对象方向，尤其是 COM; 松散地， COM + 活动由 COM (业务) 对象表示。

COM + 是操作系统的集成部分。

COM + 为业务对象管理员提供三种服务，消除了业务对象程序员的许多烦恼:

- 事务管理

- 安全性
- 资源池

通常，将 COM + 与前端代码一起使用，而前端代码是 COM + 中保存的对象和后端服务 (例如数据库) 的 COM + ，并在 COM + 业务对象和后端之间进行 IBM MQ 桥接。

前端代码可以是独立程序，也可以是由 Microsoft Internet Information Server (IIS) 托管的活动服务器页面 (ASP)。 前端代码可以与 COM + 及其业务对象在同一台计算机上，通过 COM 进行连接。 或者，前端代码可以在不同的计算机上，通过 DCOM 进行连接。 您可以使用不同的客户机在不同的情况下访问同一个 COM + 业务对象。

后端代码可以与 COM + 及其业务对象位于同一台计算机上，也可以位于通过任何 IBM MQ 支持的协议进行连接的另一台计算机上。

即将到期的全球工作单元

可以将队列管理器配置为在预先配置的不活动时间间隔后使全局工作单元到期。

要启用此行为，请设置以下环境变量:

- **AMQ_TRANSACTION_EXPIRY_RESCAN**=重新扫描时间间隔 *(以毫秒计)*
- **AMQ_XA_TRANSACTION_EXPIRY**=超时时间间隔 *(以毫秒计)*

注意: 环境变量仅影响 OPEN 组中 XA 规范 表 6-4 中处于 **Idle** 状态的事务。

即，未在任何应用程序线程上关联但外部事务管理器软件尚未调用 **xa_prepare** 函数调用的事务。

外部事务管理器仅保留已准备，已落实或已回滚的事务的日志。 如果外部事务管理器由于任何原因而关闭，那么在返回时，它会将已准备，已落实和已回滚的事务驱动至完成，但尚未准备的任何活动事务将成为孤立事务。 要避免此情况，请将 **AMQ_XA_TRANSACTION_EXPIRY** 设置为允许应用程序执行 MQI 事务性 API 调用与完成事务之间的预期时间间隔 (在其他资源管理器上执行事务性工作)。

要确保在 **AMQ_XA_TRANSACTION_EXPIRY** 到期后及时清除，请将 **AMQ_TRANSACTION_EXPIRY_RESCAN** 值设置为小于 **AMQ_XA_TRANSACTION_EXPIRY** 时间间隔的值，理想情况下，以便在 **AMQ_XA_TRANSACTION_EXPIRY** 时间间隔内多次发生重新扫描。

恢复单元处置

IBM MQ for z/OS 提供了恢复处置单元。 此功能允许您配置在连接到同一队列共享组 (QSG) 中的另一个队列管理器时，是否可以驱动 2 阶段落实事务的第二阶段 (例如，在恢复期间)。

IBM MQ for z/OS V7.0.1 和更高版本支持恢复单元处置。

恢复单元处置

恢复处置单元与应用程序的连接以及随后启动的任何事务相关。 有两个可能的恢复单元。

- GROUP 恢复单元处置标识事务应用程序在逻辑上已连接到队列共享组，并且与任何特定队列管理器都没有亲缘关系。 在连接到 QSG 中的任何队列管理器时，可以查询并解决它启动的任何 2 阶段落实事务，这些事务已完成落实进程的 phase-1 (即，它们处于不确定状态)。 在恢复方案中，这意味着事务协调程序不必重新连接到可能不可用的同一队列管理器。
- QMGR 恢复处置单元标识应用程序与它所连接的队列管理器具有直接亲缘关系，并且它启动的任何事务也具有此处置。

  在恢复方案中，事务协调程序必须重新连接到同一队列管理器以查询和解决任何不确定事务，而不考虑该队列管理器是否属于队列共享组。

z/OS 有关如何实现此功能的详细信息，请参阅 队列共享组中的恢复单元处置。

安全方案

演示将安全性应用于不同配置的一组方案。

以下子主题中描述了可用的安全方案:

相关任务

z/OS 在 z/OS 上设置安全性

z/OS 安全方案: z/OS 上的两个队列管理器

在此场景中，应用程序使用 **MQPUT1** 调用将消息放入队列管理器 QM1 上的队列。 然后，使用 TCP 和 LU 6.2 通道将某些消息转发到 QM2 上的队列。 TCP 通道可以使用 SSL/TLS，也可以不使用 SSL/TLS。 应用程序可以是批处理应用程序或 CICS 应用程序，并且使用 MQPMO_SET_ALL_CONTEXT 选项放置消息。

第 73 页的图 13 对此作了说明。

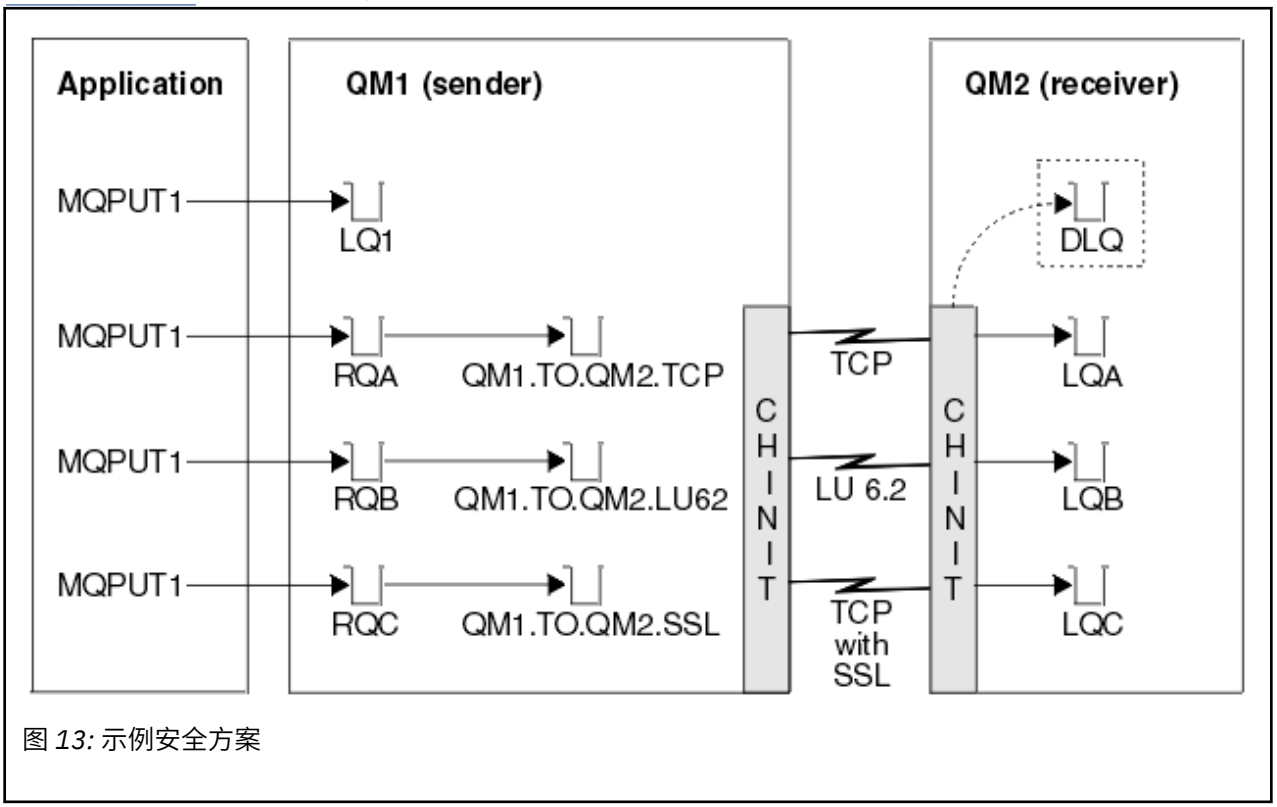


图 *13:* 示例安全方案

对队列管理器进行了以下假定:

- 所有必需的 IBM MQ 定义都已预定义或已通过队列管理器启动时处理的 CSQINP2 数据集进行。

  如果没有这些对象，那么您需要对定义这些对象所需的命令具有相应的访问权限。
- 在启动队列管理器和通道启动程序之前，已定义所需的所有 RACF 概要文件，并且已授予相应的访问权限。

  如果没有这些概要文件，那么您需要适当的权限来发出定义所有所需概要文件所需的 RACF 命令，并授予这些概要文件相应的访问权限。 您还需要适当的权限来发出 MQSC 安全命令，以开始使用新的安全概要文件。
- 已创建所需的所有数字证书并将其连接到密钥环。 作为 SSL/TLS 握手的一部分由 QM1 发送的数字证书可由 QM2 系统上的 RACF 识别，因为它也安装在该 RACF 概要文件中，或者因为存在匹配的证书名称文件 (CNF) 过滤器。

相关任务

z/OS 在 z/OS 上设置安全性

z/OS 两个队列管理器方案的安全开关设置

切换设置和 RACF 概要文件。

为这两个队列管理器设置了以下安全开关:

- 子系统安全性开启
- 队列安全性开启
- 备用用户安全性
- 上下文安全性开启
- 流程安全性关闭
- 名称列表安全性关闭
- 主题安全性关闭
- 连接安全性开启
- 命令安全性开启
- 命令资源安全性开启

在 MQADMIN 类中定义了以下概要文件以关闭进程，名称列表和主题安全性:

```
QM1.NO.PROCESS.CHECKS
QM1.NO.NLIST.CHECKS
QM1.NO.TOPIC.CHECKS
QM2.NO.PROCESS.CHECKS
QM2.NO.NLIST.CHECKS
QM2.NO.TOPIC.CHECKS
```

z/OS 两个队列管理器方案中的队列管理器 QM1

QM1 的队列和通道。

在队列管理器 QM1:

LQ1
本地队列。

RQA
具有以下属性的远程队列定义:

- RNAME (LQA)
- RQMNAME (QM2)
- XMITQ (QM1.TO.QM2.TCP)

RQB
具有以下属性的远程队列定义:

- RNAME (LQB)
- RQMNAME (QM2)
- XMITQ (QM1.TO.QM2.LU62)

RQC
具有以下属性的远程队列定义:

- RNAME (LQC)
- RQMNAME (QM2)
- XMITQ (QM1.TO.QM2.TLS)

QM1.TO.QM2.TCP
传输队列。

QM1.TO.QM2.LU62
传输队列。

QM1.TO.QM2.TLS
传输队列。

在 QM1:

QM1.TO.QM2.TCP
具有以下属性的发送方通道定义:

- CHLTYPE (SDR)
- TRPTYPE (TCP)
- XMITQ (QM1.TO.QM2.TCP)
- CONNAME (QM2TCP)

QM1.TO.QM2.LU62
具有以下属性的发送方通道定义:

- CHLTYPE (SDR)
- TRPTYPE (LU62)
- XMITQ (QM1.TO.QM2.LU62)
- CONNAME (QM2LU62)

(请参阅 z/OS 上通道启动程序的安全性注意事项，以获取有关设置 APPC 安全性的信息。)

QM1.TO.QM2.TLS
具有以下属性的发送方通道定义:

- CHLTYPE (SDR)
- TRPTYPE (TCP)
- XMITQ (QM1.TO.QM2.TLS)
- CONNAME (QM2TCP)
- SSLCIPH (TLS_RSA_WITH_AES_128_CBC_SHA256)

z/OS 两个队列管理器方案中的队列管理器 QM2

QM2 的队列和通道。

已在队列管理器 QM2:

LQA
本地队列。

LQB
本地队列。

LQC
本地队列。

DLQ
用作死信队列的本地队列。

已在 QM2:

QM1.TO.QM2.TCP
具有以下属性的接收方通道定义:

- CHLTYPE (RCVR)
- TRPTYPE (TCP)
- PUTAUT (CTX)
- MCAUSER (MCATCP)

QM1.TO.QM2.LU62
具有以下属性的接收方通道定义:

- CHLTYPE (RCVR)
- TRPTYPE (LU62)
- PUTAUT (CTX)

- MCAUSER (MCALU62)

(请参阅 z/OS 上通道启动程序的安全性注意事项，以获取有关设置 APPC 安全性的信息。)

QM1.TO.QM2.TLS
具有以下属性的接收方通道定义:

- CHLTYPE (RCVR)
- TRPTYPE (TCP)
- PUTAUT (CTX)
- MCAUSER (MCASSL)
- SSLCIPH (TLS_RSA_WITH_AES_128_CBC_SHA256)

z/OS 在两个队列管理器方案中使用的用户标识

方案中用户标识的说明。

使用以下用户标识:

BatchId
批处理应用程序 (作业或 TSO 标识)

MSGUSR
MQMD 中的 *UserIdentifier* (上下文用户标识)

MOVER1
QM1 通道启动程序地址空间用户标识

MOVER2
QM2 通道启动程序地址空间用户标识

MCATCP
在没有 SSL/TLS 接收方通道定义的 TCP/IP 上指定了 MCAUSER

MCALU62
在 LU 6.2 接收方通道定义上指定的 MCAUSER

MCASSL
在具有 SSL/TLS 接收方通道定义的 TCP/IP 上指定的 MCAUSER

CICSAD1
CICS 地址空间标识

CICSTX1
CICS 任务用户标识

CERTID
RACF 与流证书关联的用户标识。

z/OS 两个队列管理器方案所需的安全概要文件和访问权

有关两个队列管理器方案的批处理或 CICS 实现所需的安全概要文件和访问权的信息。

下表显示了使两个队列管理器方案工作所需的安全概要文件。 还需要其他安全概要文件，具体取决于您是在执行方案的批处理还是 CICS 实现。 有关更多信息，请参阅 第 77 页的『批处理应用程序所需的安全概要文件』 和 第 79 页的『CICS 应用程序所需的安全概要文件』。

表 *10:* 示例方案的安全概要文件.

此表中的四列显示了两个队列管理器方案的类，概要文件，用户标识和访问权。

类	概要文件	用户标识	访问
MQCONN	QM1.CHIN	MOVER1	读取

表 *10:* 示例方案的安全概要文件.

此表中的四列显示了两个队列管理器方案的类，概要文件，用户标识和访问权。

(继续)

类	概要文件	用户标识	访问
MQADMIN	QM1.RESLEVEL	BATCHID CICSAD1 MOVER1	无
MQADMIN	QM1.CONTEXT.**	MOVER1	CONTROL
MQQUEUE	QM1.SYSTEM.COMMAND.INPUT	MOVER1	更新
MQQUEUE	QM1.SYSTEM.CHANNEL.SYNCQ	MOVER1	更新
MQQUEUE	QM1.SYSTEM.CHANNEL.INITQ	MOVER1	更新
MQQUEUE	QM1.SYSTEM.COMMAND.REPLY.MODEL	MOVER1	更新
MQQUEUE	QM1.SYSTEM.ADMIN.CHANNEL.EVENT	MOVER1	更新
MQQUEUE	QM1.QM1.TO.QM2.TCP	MOVER1	更改
MQQUEUE	QM1.QM1.TO.QM2.LU62	MOVER1	更改
MQQUEUE	QM1.QM1.TO.QM2.TLS	MOVER1	更改
MQCONN	QM2.CHIN	MOVER2	读取
MQADMIN	QM2.RESLEVEL	MOVER2	无
MQADMIN	QM2.CONTEXT.**	MOVER2	CONTROL
MQQUEUE	QM2.SYSTEM.COMMAND.INPUT	MOVER2	更新
MQQUEUE	QM2.SYSTEM.CHANNEL.SYNCQ	MOVER2	更新
MQQUEUE	QM2.SYSTEM.CHANNEL.INITQ	MOVER2	更新
MQQUEUE	QM2.SYSTEM.COMMAND.REPLY.MODEL	MOVER2	更新
MQQUEUE	QM2.SYSTEM.ADMIN.CHANNEL.EVENT	MOVER2	更新
MQQUEUE	QM2.DLQ	MOVER2	更新

z/OS 批处理应用程序所需的安全概要文件

两个队列管理器方案的批处理实现所需的其他安全概要文件。

批处理应用程序在 QM1 上的用户标识 BATCHID 下运行。 它连接到队列管理器 QM1 并将消息放入以下队列:

- LQ1
- RQA
- RQB
- RQC

它使用 MQPMO_SET_ALL_CONTEXT 选项。 在消息描述符 (MQMD) 的 *UserIdentifier* 字段中找到的备用用户标识为 MSGUSR。

队列管理器 QM1:

表 *11:* 队列管理器 *QM1* 上批处理应用程序的样本安全概要文件

类	概要文件	用户标识	访问
MQCONN	QM1.BATCH	BatchId	读取

表 *11:* 队列管理器 *QM1* 上批处理应用程序的样本安全概要文件 *(继续)*			
类	**概要文件**	**用户标识**	**访问**
MQADMIN	QM1.CONTEXT.**	BatchId	CONTROL
MQQUEUE	QM1.LQ1	BatchId	更新
MQQUEUE	QM1.RQA	BatchId	更新
MQQUEUE	QM1.RQB	BatchId	更新
MQQUEUE	QM1.RQC	BatchId	更新

对于放入队列管理器 QM1 上队列 RQA 的消息，队列管理器 QM2 上需要以下概要文件 (对于不使用 TLS 的 TCP/IP 通道):

表 *12:* 使用 *TCP/IP* 而不是 *TLS* 的队列管理器 *QM2* 的样本安全概要文件			
类	**概要文件**	**用户标识**	**访问**
MQADMIN	QM2.ALTERNATE.USER.MSGUSR	MCATCP MOVER2	更新
MQADMIN	QM2.CONTEXT.**	MCATCP MOVER2	CONTROL
MQQUEUE	QM2.LQA	MOVER2 MSGUSR	更新
MQQUEUE	QM2.DLQ	MOVER2 MSGUSR	更新

注意:

1. 在消息的 MQMD 中传递的用户标识将用作队列管理器 QM2 上 MQPUT1 的用户标识，因为接收方通道是使用 PUTAUT (CTX) 和 MCAUSER (MCATCP) 定义的。
2. 接收方通道定义的 MCAUSER 字段设置为 MCATCP; 除了通道启动程序地址空间用户标识外，还将使用此用户标识对备用用户标识和上下文概要文件执行的检查。
3. 消息描述符 (MQMD) 中的 MOVER2 用户标识和 *UserIdentifier* 用于针对队列进行资源检查。
4. MOVER2 和 MSGUSR 用户标识都需要访问死信队列，以便无法放入目标队列的消息可以在该队列中发送。
5. 由于 RESLEVEL 设置为 NONE ，因此将在执行的所有三项检查中检查两个用户标识。

对于放入队列管理器 QM1 (对于 LU 6.2 通道) 上的队列 RQB 的消息，队列管理器 QM2 上需要以下概要文件:

表 *13:* 使用 *LU 6.2* 的队列管理器 *QM2* 的样本安全概要文件			
类	**概要文件**	**用户标识**	**访问**
MQADMIN	QM2.ALTERNATE.USER.MSGUSR	MCALU62 MOVER1	更新
MQADMIN	QM2.CONTEXT.**	MCALU62 MOVER1	CONTROL
MQQUEUE	QM2.LQB	MOVER1 MSGUSR	更新
MQQUEUE	QM2.DLQ	MOVER1 MSGUSR	更新

注意:

1. 在消息的 MQMD 中传递的用户标识将用作队列管理器 QM2 上 MQPUT1 的用户标识，因为接收方通道是使用 PUTAUT (CTX) 和 MCAUSER (MCALU62) 定义的。
2. MCA 用户标识设置为接收方通道定义 (MCALU62) 的 MCAUSER 字段的值。
3. 由于 LU 6.2 支持通道的通信系统上的安全性，因此从网络接收的用户标识将用作通道用户标识 (MOVER1)。

4. 由于 RESLEVEL 设置为 NONE，因此将在执行的所有三项检查中检查两个用户标识。
5. MCALU62 和 MOVER1 用于对备用用户标识和上下文概要文件执行的检查， MSGUSR 和 MOVER1 用于对队列概要文件执行的检查。
6. MOVER1 和 MSGUSR 用户标识都需要访问死信队列，以便无法放入目标队列的消息可以在那里发送。

对于放入队列管理器 QM1 (对于使用 TLS 的 TCP/IP 通道) 上的队列 RQC 的消息，队列管理器 QM2 上需要以下概要文件:

表 *14:* 使用 *TCP/IP* 和 *TLS* 的队列管理器 *QM2* 的样本安全概要文件

类	概要文件	用户标识	访问
MQADMIN	QM2.ALTERNATE.USER.MSGUSR	MCASSL CERTID	更新
MQADMIN	QM2.CONTEXT.**	MCASSL CERTID	CONTROL
MQQUEUE	QM2.LQC	CERTID MSGUSR	更新
MQQUEUE	QM2.DLQ	CERTID MSGUSR	更新

注意:

1. 在消息的 MQMD 中传递的用户标识将用作队列管理器 QM2 上 MQPUT1 的用户标识，因为接收方通道是使用 PUTAUT (CTX) 和 MCAUSER (MCASSL) 定义的。
2. MCA 用户标识设置为接收方通道定义 (MCASSL) 的 MCAUSER 字段的值。
3. 由于通道作为 TLS 握手的一部分从 QM1 流出的证书可能安装在 QM2 的系统上，或者可能与 QM2 的系统上的证书名称过滤器匹配，因此在该匹配期间找到的用户标识将用作通道用户标识 (CERTID)。
4. 由于 RESLEVEL 设置为 NONE，因此将在执行的所有三项检查中检查两个用户标识。
5. MCASSL 和 CERTID 用于对备用用户标识和上下文概要文件执行的检查， MSGUSR 和 MOVER1 用于对队列概要文件执行的检查。
6. CERTID 和 MSGUSR 用户标识都需要访问死信队列，以便无法放入目标队列的消息可以在那里发送。

z/OS *CICS* 应用程序所需的安全概要文件

两个队列管理器方案的 CICS 实现所需的其他安全概要文件。

CICS 应用程序使用 CICS 地址空间用户标识 CICSAD1 和 CICS 任务用户标识 CICSTX1。 队列管理器 QM1 上所需的安全概要文件与批处理应用程序所需的安全概要文件不同。 队列管理器 QM2 上所需的概要文件与批处理应用程序的概要文件相同。

队列管理器 QM1:

表 *15:* 队列管理器 *QM1* 上 *CICS* 应用程序的样本安全概要文件

类	概要文件	用户标识	访问
MQCONN	QM1.CICS	CICSAD1	读取
MQADMIN	QM1.CONTEXT.**	CICSAD1 CICSTX1	CONTROL
MQQUEUE	QM1.LQ1	CICSAD1 CICSTX1	更新
MQQUEUE	QM1.RQA	CICSAD1 CICSTX1	更新
MQQUEUE	QM1.RQB	CICSAD1 CICSTX1	更新

z/OS 安全方案: z/OS 上的队列共享组

在此场景中，应用程序使用 **MQPUT1** 调用将消息放入队列管理器 QM1 上的队列。 然后，使用 TCP 和 LU 6.2 通道将某些消息转发到 QM2 上的队列。 应用程序是批处理应用程序，使用 MQPMO_SET_ALL_CONTEXT 选项放置消息。

第 73 页的图 13 对此作了说明。

对队列管理器进行了以下假定:

- 所有必需的 IBM MQ 定义都已预定义或已通过队列管理器启动时处理的 CSQINP2 数据集进行。

  如果没有这些对象，那么您需要对定义这些对象所需的命令具有相应的访问权限。
- 在启动队列管理器和通道启动程序之前，已定义所需的所有 RACF 概要文件，并且已授予相应的访问权限。

  如果没有这些概要文件，那么您需要适当的权限来发出定义所有所需概要文件所需的 RACF 命令，并授予这些概要文件相应的访问权限。 您还需要适当的权限来发出 MQSC 安全命令，以开始使用新的安全概要文件。

相关任务

z/OS 在 z/OS 上设置安全性

z/OS 队列共享组方案的安全开关设置

切换设置和 RACF 概要文件。

为队列共享组设置了以下安全开关:

- 子系统安全性开启
- 队列共享组安全性开启
- 队列管理器安全性关闭
- 队列安全性开启
- 备用用户安全性
- 上下文安全性开启
- 流程安全性关闭
- 名称列表安全性关闭
- 主题安全性关闭
- 连接安全性开启
- 命令安全性开启
- 命令资源安全性开启

在 MQADMIN 类中定义了以下概要文件以关闭进程，名称列表，主题和队列管理器级别安全性:

```
QSGA.NO.PROCESS.CHECKS
QSGA.NO.NLIST.CHECKS
QSGA.NO.TOPIC.CHECKS
QSGA.NO.QMGR.CHECKS
```

z/OS 队列共享组方案中的队列管理器 QM1

QM1 的队列和通道。

在队列管理器 QM1:

LQ1
本地队列。

RQA
具有以下属性的远程队列定义:

- RNAME (LQA)
- RQMNAME (QM2)
- XMITQ (QM1.TO.QM2.TCP)

RQB
具有以下属性的远程队列定义:

- RNAME (LQB)
- RQMNAME (QM2)
- XMITQ (QM1.TO.QM2.LU62)

QM1.TO.QM2.TCP
传输队列。

QM1.TO.QM2.LU62
传输队列。

在 QM1:

QM1.TO.QM2.TCP
具有以下属性的发送方通道定义:

- CHLTYPE (SDR)
- TRPTYPE (TCP)
- XMITQ (QM1.TO.QM2.TCP)
- CONNAME (QM2TCP)

QM1.TO.QM2.LU62
具有以下属性的发送方通道定义:

- CHLTYPE (SDR)
- TRPTYPE (LU62)
- XMITQ (QM1.TO.QM2.LU62)
- CONNAME (QM2LU62)

(请参阅 z/OS 上通道启动程序的安全性注意事项，以获取有关设置 APPC 安全性的信息。)

z/OS 队列共享组方案中的队列管理器 QM2

QM2 的队列和通道。

已在队列管理器 QM2:

LQA
本地队列。

LQB
本地队列。

DLQ
用作死信队列的本地队列。

已在 QM2:

QM1.TO.QM2.TCP
具有以下属性的接收方通道定义:

- CHLTYPE (RCVR)
- TRPTYPE (TCP)
- PUTAUT (CTX)

- MCAUSER (MCATCP)

QM1.TO.QM2.LU62
具有以下属性的接收方通道定义:

- CHLTYPE (RCVR)
- TRPTYPE (LU62)
- PUTAUT (CTX)
- MCAUSER (MCALU62)

(请参阅 z/OS 上通道启动程序的安全性注意事项，以获取有关设置 APPC 安全性的信息。)

z/OS 队列共享组方案中使用的用户标识

方案中用户标识的说明。

使用以下用户标识:

BatchId
批处理应用程序 (作业或 TSO 标识)

MSGUSR
MQMD 中的 *UserIdentifier* (上下文用户标识)

MOVER1
QM1 通道启动程序地址空间用户标识

MOVER2
QM2 通道启动程序地址空间用户标识

MCATCP
在 TCP/IP 接收方通道定义上指定了 MCAUSER

MCALU62
在 LU 6.2 接收方通道定义上指定的 MCAUSER

z/OS 队列共享组方案所需的安全概要文件和访问权

队列共享组方案的批处理或 CICS 实现的安全概要文件和访问权。

下表显示了使队列共享组方案工作所需的安全概要文件。 此场景的批处理实现还需要 第 83 页的『批处理应用程序所需的安全概要文件』中描述的其他安全概要文件。

表 *16:* 示例方案的安全概要文件.
此表中的四列显示队列共享组方案的类，概要文件，用户标识和访问权。

类	概要文件	用户标识	访问
MQCONN	QSGA.CHIN	MOVER1 MOVER2	读取
MQADMIN	QSGA.RESLEVEL	BATCHID MOVER1 MOVER2	无
MQADMIN	QSGA.CONTEXT.**	MOVER1 MOVER2	CONTROL
MQQUEUE	QSGA.SYSTEM.COMMAND.INPUT	MOVER1 MOVER2	更新
MQQUEUE	QSGA.SYSTEM.CHANNEL.SYNCQ	MOVER1 移动者	更新

表 *16:* 示例方案的安全概要文件.

此表中的四列显示队列共享组方案的类，概要文件，用户标识和访问权。

(继续)

类	概要文件	用户标识	访问
MQQUEUE	QSGA.SYSTEM.CHANNEL.INITQ	MOVER1 MOVER2	更新
MQQUEUE	QSGA.SYSTEM.COMMAND.REPLY.MODEL	MOVER1 MOVER2	更新
MQQUEUE	QSGA.SYSTEM.ADMIN.CHANNEL.EVENT	MOVER1 MOVER2	更新
MQQUEUE	QSGA.SYSTEM.QSG.CHANNEL.SYNCQ	MOVER1 MOVER2	更新
MQQUEUE	QSGA.SYSTEM.QSG.TRANSMIT.QUEUE	MOVER1 MOVER2	更新
MQQUEUE	QSGA.QM1.TO.QM2.TCP	MOVER1	更改
MQQUEUE	QSGA.QM1.TO.QM2.LU62	MOVER1	更改
MQQUEUE	QSGA.DLQ	MOVER2	更新

z/OS 批处理应用程序所需的安全概要文件

队列共享组方案的批处理实现所需的其他安全概要文件。

批处理应用程序在 QM1 上的用户标识 BATCHID 下运行。 它连接到队列管理器 QM1 并将消息放入以下队列:

- LQ1
- RQA
- RQB

它使用 MQPMO_SET_ALL_CONTEXT 选项。 在消息描述符 (MQMD) 的 *UserIdentifier* 字段中找到的用户标识为 MSGUSR。

队列管理器 QM1:

表 *17:* 队列管理器 *QM1* 上批处理应用程序的样本安全概要文件

类	概要文件	用户标识	访问
MQCONN	QSGA.BATCH	BatchId	读取
MQADMIN	QSGA.CONTEXT.**	BatchId	CONTROL
MQQUEUE	QSGA.LQ1	BatchId	更新
MQQUEUE	QSGA.RQA	BatchId	更新
MQQUEUE	QSGA.RQB	BatchId	更新

对于放入队列管理器 QM1 (对于 TCP/IP 通道) 上队列 RQA 的消息，队列管理器 QM2 上需要以下概要文件:

表 *18:* 使用 *TCP/IP* 的队列管理器 *QM2* 的样本安全概要文件

类	概要文件	用户标识	访问
MQADMIN	QSGA.ALTERNATE.USER.MSGUSR	MCATCP MOVER2	更新
MQADMIN	QSGA.CONTEXT.**	MCATCP MOVER2	CONTROL

表 *18:* 使用 *TCP/IP* 的队列管理器 *QM2* 的样本安全概要文件 *(继续)*			
类	**概要文件**	**用户标识**	**访问**
MQQUEUE	QSGA.LQA	MOVER2 MSGUSR	更新
MQQUEUE	QSGA.DLQ	MOVER2 MSGUSR	更新

注意:

1. 在消息的 MQMD 中传递的用户标识将用作队列管理器 QM2 上 MQPUT1 的用户标识，因为接收方通道是使用 PUTAUT (CTX) 和 MCAUSER (MCATCP) 定义的。
2. 接收方通道定义的 MCAUSER 字段设置为 MCATCP; 除了通道启动程序地址空间用户标识外，还将使用此用户标识对备用用户标识和上下文概要文件执行的检查。
3. 消息描述符 (MQMD) 中的 MOVER2 用户标识和 *UserIdentifier* 用于针对队列进行资源检查。
4. MOVER2 和 MSGUSR 用户标识都需要访问死信队列，以便无法放入目标队列的消息可以在该队列中发送。
5. 由于 RESLEVEL 设置为 NONE ，因此将在执行的所有三项检查中检查两个用户标识。

对于放入队列管理器 QM1 (对于 LU 6.2 通道) 上的队列 RQB 的消息，队列管理器 QM2 上需要以下概要文件:

表 *19:* 使用 *LU 6.2* 的队列管理器 *QM2* 的样本安全概要文件			
类	**概要文件**	**用户标识**	**访问**
MQADMIN	QSGA.ALTERNATE.USER.MSGUSR	MCALU62 MOVER1	更新
MQADMIN	QSGA.CONTEXT.**	MCALU62 MOVER1	CONTROL
MQQUEUE	QSGA.LQB	MOVER1 MSGUSR	更新
MQQUEUE	QSGA.DLQ	MOVER1 MSGUSR	更新

注意:

1. 在消息的 MQMD 中传递的用户标识将用作队列管理器 QM2 上 MQPUT1 的用户标识，因为接收方通道是使用 PUTAUT (CTX) 和 MCAUSER (MCALU62) 定义的。
2. MCA 用户标识设置为接收方通道定义 (MCALU62) 的 MCAUSER 字段的值。
3. 由于 LU 6.2 支持通道的通信系统上的安全性，因此从网络接收的用户标识将用作通道用户标识 (MOVER1)。
4. 由于 RESLEVEL 设置为 NONE ，因此将在执行的所有三项检查中检查两个用户标识。
5. MCALU62 和 MOVER1 用于对备用用户标识和上下文概要文件执行的检查， MSGUSR 和 MOVER1 用于对队列概要文件执行的检查。
6. MOVER1 和 MSGUSR 用户标识都需要访问死信队列，以便无法放入目标队列的消息可以在那里发送。

z/OS 服务器到服务器消息通道拦截示例配置

服务器到服务器消息通道拦截需要配置通道定义以及 Advanced Message Security 策略，以确保入站和出站消息可以正确受保护或不受保护。 根据通道是入站还是出站，配置会有所不同。

入站通道 (inbound channel)

以下示例显示了类型为接收方的入站通道的典型配置，并提供了保护不受保护的入站消息所需的 AMS 策略的详细信息:

图 *14:* 入站配置

示例显示:

- 队列管理器 QMA
- 通道 TO.QMA
- 本地队列 DESTQ

使用以下代码:

```
DEFINE CHANNEL(TO.QMA) CHLTYPE(RCVR) SSLCAUTH(REQUIRED) SSLCIPH(ANY_TLS12) TRPTYPE(TCP)
SPLPROT(ASPOLICY)

DEFINE QLOCAL(DESTQ) DESCR('AMS PROTECTED QUEUE')

setmqspl -m QMA -p DESTQ -e AES256 -r CN=TEST,O=ORG,C=US
```

注: 上述文本中描述的策略仅对消息进行加密; 即， AMS 机密性。

请参阅 setmqspl 和 消息安全策略 (CSQOUTIL)， 以获取有关在 z/OS 上使用 **setmqspl** 的信息。

出站通道 (outbound channel)

以下示例显示了类型为发送方的出站通道的典型配置。 此示例提供了保护放入远程队列的消息以及取消保护和发送从传输队列获取的消息所需的 AMS 策略的详细信息:

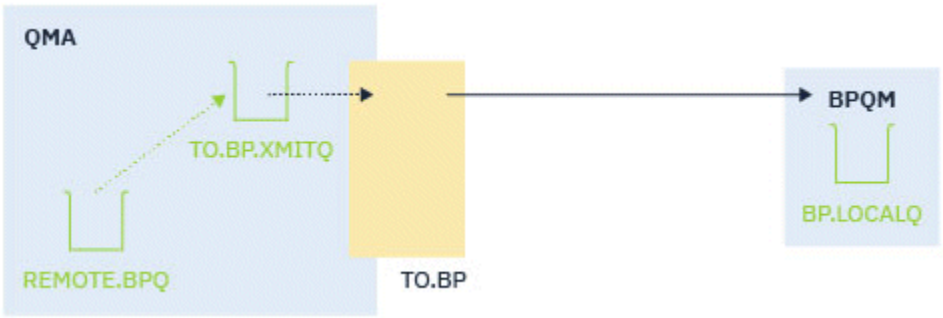


图 *15:* 出站配置

示例显示了以下内容:

- 队列管理器 QMA
- 通道 TO.BP
- 本地传输队列 TO.BP.XMITQ
- 远程队列 REMOTE.BPQ

使用以下代码:

```
DEFINE CHANNEL(TO.BP) CHLTYPE(SDR) SSLCAUTH(REQUIRED) SSLCIPH(ANY_TLS12) TRPTYPE(TCP)
SPLPROT(REMOVE) CONNAME('server(1414)') XMITQ(TO.BP.XMITQ)
```

```
DEFINE QLOCAL(TO.BP.XMITQ) DESCR('TRANSMISSION QUEUE FOR TO.BP') USAGE(XMITQ)

DEFINE QREMOTE(REMOTE.BPQ) DESCR('REMOTE QUEUE TO BP') RNAME(BP.LOCALQ) RQMANME(BPQM)
XMITQ(TO.BP.XMITQ)

setmqspl -m QMA -p TO.BP.XMITQ -e AES256 -r CN=TEST,O=ORG,C=US

setmqspl -m QMA -p REMOTE.BPQ -e AES256 -r CN=TEST,O=ORG,C=US
```

注: 上述文本中描述的策略仅对消息进行加密; 即， AMS 机密性。

使用 SSL/TLS 连接两个队列管理器

使用 TLS 加密安全协议的安全通信涉及设置通信通道和管理将用于认证的数字证书。

要设置 SSL/TLS 安装，必须定义通道以使用 TLS。 您还必须获取和管理数字证书。 在测试系统上，可以使用自签名证书或本地认证中心 (CA) 颁发的证书。 在生产系统上，请勿使用自签名证书。

有关创建和管理证书的完整信息，请参阅以下主题:

- IBM i 在 IBM i 上使用 SSL 或 TLS
- ALW 在 AIX, Linux, and Windows 系统上使用 SSL 或 TLS
- z/OS 在 z/OS 上使用 SSL 或 TLS

此主题集合介绍了设置 SSL/TLS 通信所涉及的任务，并提供了有关完成这些任务的逐步指导。

您可能还想要测试 SSL/TLS 客户机认证，这是协议的可选部分。 在 SSL/TLS 握手期间， SSL/TLS 客户机始终从服务器获取并验证数字证书。 通过 IBM MQ 实现， SSL/TLS 服务器始终从客户机请求证书。

注意:

1. 在此上下文中， SSL/TLS 客户机是指启动握手的连接。
2. z/OS 当 z/OS 队列管理器充当 SSL/TLS 客户机的角色时，队列管理器仅发送证书。

仅当 SSL/TLS 客户机可以找到具有匹配标签的证书时，才会发送证书。 请参阅 数字证书标签 以获取详细信息。

如果发送了客户机证书，那么 SSL/TLS 服务器将始终验证该客户机证书。 如果客户机未发送证书，那么仅当使用设置为 REQUIRED 的 **SSLCAUTH** 参数或设置为 **SSLPEER** 参数的值来定义充当 SSL/TLS 服务器的通道结束时，认证才会失败。 有关匿名连接队列管理器的更多信息，即，当 SSL/TLS 客户机未发送证书时，请参阅 第 91 页的『使用单向认证连接两个队列管理器』。

使用自签名证书对两个队列管理器进行相互认证

遵循以下样本指示信息，使用自签名 TLS 证书在两个队列管理器之间实现相互认证。

关于此任务

场景:

- 您有两个需要安全通信的队列管理器 QM1 和 QM2。 您需要在 QM1 和 QM2 之间执行相互认证。
- 您已决定使用自签名证书测试安全通信。

生成的配置如下所示:

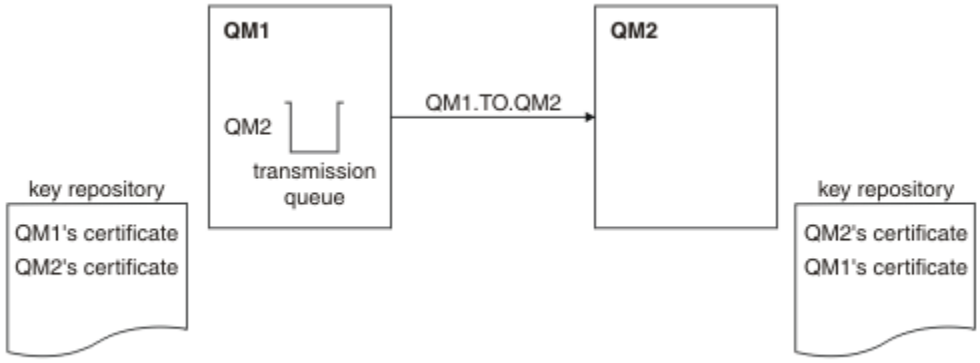


图 *16:* 由此任务生成的配置

在 第 87 页的图 16 中， QM1 的密钥存储库包含 QM1 的证书和来自 QM2 的公用证书。 QM2 的密钥存储库包含 QM2 的证书以及来自 QM1 的公用证书。

过程

1. 根据操作系统，在每个队列管理器上准备密钥存储库:
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上。
2. 为每个队列管理器创建自签名证书:
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上。
3. 抽取每个证书的副本:
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上。
4. 使用诸如 FTP z/OS 之类的实用程序将 QM1 证书的公用部分传输到 QM2 系统，反之亦然，如 交换自签名证书 中所述。
5. 将合作伙伴证书添加到每个队列管理器的密钥存储库:
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上。
6. 在 QM1 上，通过发出类似以下示例的命令来定义发送方通道和关联的传输队列:

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
   SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) DESCR('Sender channel using TLS from QM1 to QM2')

   DEFINE QLOCAL(QM2) USAGE(XMITQ)
   ```

   此示例使用 CipherSpec TLS_RSA。 通道两端的 CipherSpecs 必须相同。
7. 在 QM2 上，通过发出类似于以下示例的命令来定义接收方通道:

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS from QM1 to QM2')
   ```

   通道必须与您在步骤 第 87 页的『6』 中定义的发送方通道同名，并使用相同的 CipherSpec。
8. 启动通道。

z/OS 在 z/OS 上，请参阅 启动发送方通道。

结果

创建密钥存储库和通道，如 第 87 页的图 16 中所述

下一步做什么

使用 DISPLAY 命令检查任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出。

从队列管理器 QM1，输入以下命令:

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM1.TO.QM2)                     CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=QM2,OU=<Department>,O=<Organization>,ST=<State>,C=<Country>")

SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=<Department>,O=<Organization>,ST=<State>,C=
<Country>")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

从队列管理器 QM2 中，输入以下命令:

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM2.TO.QM1)                     CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QM1)
   SSLCERTI("CN=QM1,OU=<Department>,O=<Organization>,ST=<State>,C=<Country>")

SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM1,OU=<Department>,O=<Organization>,ST=<State>,C=
<Country>")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
   XMITQ( )
```

在每种情况下， SSLPEER 的值必须与步骤 第 87 页的『2』中创建的伙伴证书中的 DN 的值匹配。 颁发者名称与同级名称匹配，因为证书是自签名的。

SSLPEER 是可选的。 如果指定了此参数，那么必须设置其值，以便允许伙伴证书中的 DN (在步骤 第 87 页的『2』中创建)。 有关使用 SSLPEER 的更多信息，请参阅 IBM MQ SSLPEER 值的规则。

使用 CA 签名的证书对两个队列管理器进行相互认证

遵循以下样本指示信息，以使用 CA 签署的 TLS 证书在两个队列管理器之间实现相互认证。

关于此任务

场景:

- 您有两个名为 QM1 和 QM2 的队列管理器，它们需要安全地进行通信。 您需要在 QM1 和 QM2 之间执行相互认证。
- 将来您计划在生产环境中使用此网络，因此您决定从一开始就使用 CA 签署的证书。

生成的配置如下所示:

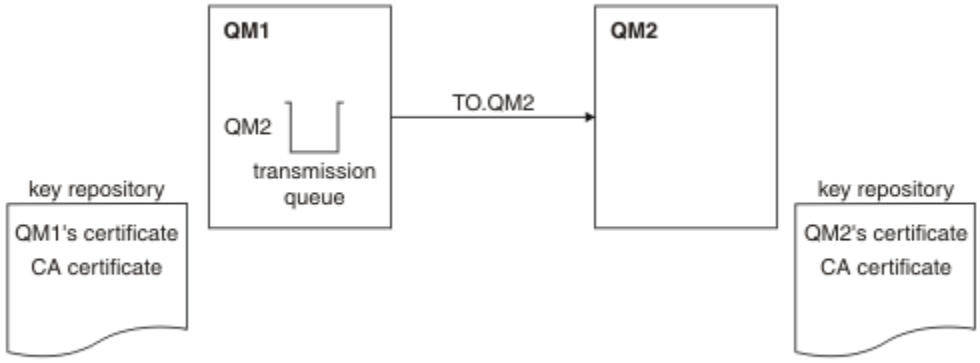


图 *17:* 由此任务生成的配置

在 第 89 页的图 17 中， QM1 的密钥存储库包含 QM1 的证书和 CA 证书。 QM2 的密钥存储库包含 QM2 的证书和 CA 证书。

在此示例中， QM1 的证书和 QM2 的证书都由同一 CA 发放。 如果 QM1 的证书和 QM2 的证书由不同的 CA 发放，那么 QM1 和 QM2 的密钥存储库必须同时包含两个 CA 证书。

过程

1. 根据企业使用的一个或多个操作系统，在每个队列管理器上准备密钥存储库:
   - IBM i 在 IBM i 系统上。
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上。
2. 请求每个队列管理器的 CA 签名证书。
   您可以对两个队列管理器使用不同的 CA。
   - IBM i 在 IBM i 系统上。
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上。
3. 将认证中心证书添加到每个队列管理器的密钥存储库:
   如果队列管理器使用不同的认证中心，那么必须将每个认证中心的 CA 证书添加到两个密钥存储库。
   - IBM i 请勿在 IBM i 系统上执行此步骤。
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上。
4. 将 CA 签名的证书接收到每个队列管理器的密钥存储库中:
   - IBM i 在 IBM i 系统上。
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上。
5. 在 QM1 上，通过发出类似于以下示例的命令来定义发送方通道和关联的传输队列:

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP)
CONNAME(QM2.MACH.COM) XMITQ(QM2) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
DESCR('Sender channel using TLS from QM1 to QM2')
```

```
DEFINE QLOCAL(QM2) USAGE(XMITQ)
```

此示例使用 CipherSpec TLS_RSA_WITH_AES_128_CBC_SHA256。 通道两端的 CipherSpecs 必须相同。

6. 在 QM2 上，通过发出类似于以下示例的命令来定义接收方通道:

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) SSLCAUTH(REQUIRED)
DESCR('Receiver channel using TLS to QM2')
```

通道必须与您在步骤 第 89 页的『5』中定义的发送方通道同名，并使用相同的 CipherSpec。

7. 启动通道:

- IBM i 在 IBM i 系统上。
- ALW 在 AIX, Linux, and Windows 系统上。
- z/OS 在 z/OS 系统上。

结果

将创建密钥存储库和通道，如 第 89 页的图 17 中所述。

下一步做什么

使用 DISPLAY 命令检查任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出。

从队列管理器 QM1，输入以下命令:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(SDR)
   CONNAME(192.0.0.2)                      CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=<Division> CA,OU=<Department>,O=<Organization>,ST=<State>,C=<Country>")

SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM2,OU=<Department>,O=<Organization>,ST=<State>,C=
<Country>")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

从队列管理器 QM2 中，输入以下命令:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(RCVR)
   CONNAME(192.0.0.1)                      CURRENT
   RQMNAME(QM1)
   SSLCERTI("CN=<Division> CA,OU=<Department>,O=<Organization>,ST=<State>,C=<Country>")

SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM1,OU=<Department>,O=<Organization>,ST=<State>,C=
<Country>")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
   XMITQ( )
```

在每种情况下， SSLPEER 的值必须与步骤 第 89 页的『2』中创建的伙伴证书中的专有名称 (DN) 的值匹配。 签发者名称与对步骤 第 89 页的『4』中添加的个人证书进行签名的 CA 证书的主体集 DN 相匹配。

使用单向认证连接两个队列管理器

遵循以下样本指示信息来修改具有相互认证的系统，以允许队列管理器使用单向认证连接到另一个系统; 即，当 SSL/TLS 客户机未发送证书时。

关于此任务

场景:

- 您的两个队列管理器 (QM1 和 QM2) 已设置为 第 88 页的『使用 CA 签名的证书对两个队列管理器进行相互认证』。
- 您希望更改 QM1 ，以便它使用单向认证连接到 QM2。

生成的配置如下所示:

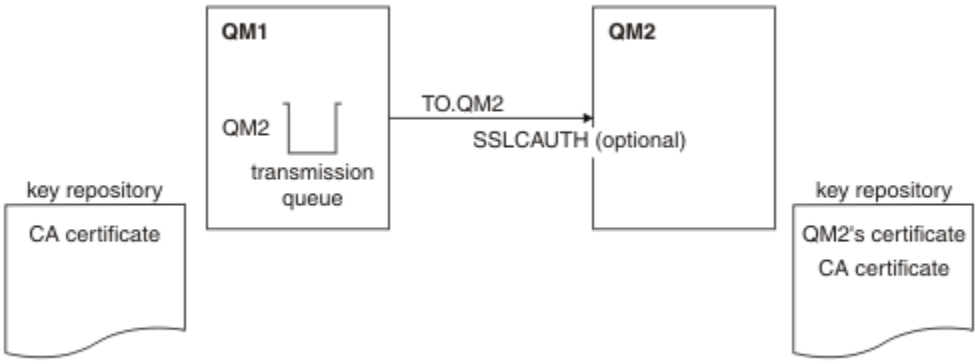


图 *18:* 允许单向认证的队列管理器

过程

1. 从其密钥存储库中除去 QM1 的个人证书:
   - IBM i 在 IBM i 系统上除去证书。
   - ALW 在 AIX, Linux, and Windows 上除去证书。
   - z/OS 在 z/OS 系统上除去证书。 执行此步骤两次，以除去 QMA 的个人证书和缺省证书。

   有关如何标注证书的详细信息，请参阅 数字证书标签。
2. 可选： 在 QM1 上，如果先前已运行任何 SSL/TLS 通道，请刷新 SSL/TLS 环境 ，如 刷新 TLS 环境 中所述。
3. 允许接收方上的匿名连接 ，如 允许接收方通道上的匿名连接 中所述。

   密钥存储库和通道已更改，如 第 91 页的图 18 中所示
4. 如果发送方通道未运行，请将其启动。

   注: 如果发送方通道正在运行，并且您在步骤 2 中发出了 REFRESH SECURITY TYPE (SSL) 命令，那么通道将自动重新启动。

   在通道的服务器端，通道状态屏幕上存在对等名称参数值指示客户机证书已流动。
5. 通过发出一些 DISPLAY 命令验证任务是否已成功完成。

   如果任务成功，那么生成的输出类似于以下示例中显示的输出:
   - 从 QM1 队列管理器中，输入以下命令:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

生成的输出将类似于以下示例:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(SDR)
   CONNAME(192.0.0.1)                      CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=IBM MQ CA,OU=IBM MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=IBM MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

- 从 QM2 队列管理器中，输入以下命令:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

生成的输出将类似于以下示例:

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(RCVR)
   CONNAME(192.0.0.2)                      CURRENT
   RQMNAME(QMA)                            SSLCERTI( )
   SSLPEER( )                              STATUS(RUNNING)
   SUBSTATE(RECEIVE)                       XMITQ( )
```

在 QM2 上， SSLPEER 字段为空，显示 QM1 未发送证书。 在 QM1 上， SSLPEER 的值与 QM2 个人证书中的 DN 的值相匹配。

安全地将客户机连接到队列管理器

使用 TLS 加密安全协议的安全通信涉及设置通信通道和管理将用于认证的数字证书。

要设置 SSL/TLS 安装，必须定义通道以使用 TLS。 您还必须获取和管理数字证书。 在测试系统上，可以使用自签名证书或本地认证中心 (CA) 颁发的证书。 在生产系统上，请勿使用自签名证书。

有关创建和管理证书的完整信息，请参阅以下主题:

- IBM i 在 IBM i 上使用 SSL 或 TLS
- ALW 在 AIX, Linux, and Windows 系统上使用 SSL 或 TLS
- z/OS 在 z/OS 上使用 SSL 或 TLS

此主题集合介绍了设置 SSL/TLS 通信所涉及的任务，并提供了有关完成这些任务的逐步指导。

您可能还想要测试 SSL/TLS 客户机认证，这是协议的可选部分。 在 SSL/TLS 握手期间， SSL/TLS 客户机始终从服务器获取并验证数字证书。 通过 IBM MQ 实现， SSL/TLS 服务器始终从客户机请求证书。

IBM i ALW 在 IBM i AIX, Linux, and Windows 系统上，仅当 SSL/TLS 客户机具有正确 IBM MQ 格式的标签 (`ibmwebspheremq` 后跟小写的登录用户标识) 或 **CERTLABL** 属性的值时， SSL/TLS 客户机才会发送证书。 请参阅 数字证书标签 (Digital certificate labels)。

如果发送了客户机证书，那么 SSL/TLS 服务器将始终验证该客户机证书。 如果客户机未发送证书，那么仅当使用 **SSLCAUTH** 参数设置为 REQUIRED 或 **SSLPEER** 参数值设置定义充当 SSL/TLS 服务器的通道结束时，认证才会失败。 有关以匿名方式连接队列管理器的更多信息，请参阅 第 96 页的『以匿名方式将客户机连接到队列管理器』。

相关概念

IBM MQ classes for Java 中的 TLS CipherSpec 和 CipherSuite

IBM MQ classes for JMS 中的 TLS CipherSpec 和 CipherSuite

相关任务

对受管 .NET 客户机使用证书

使用自签名证书对客户机和队列管理器进行相互认证

遵循以下样本指示信息，通过使用自签名 TLS 证书来实现客户机与队列管理器之间的相互认证。

关于此任务

IBM i IBM i 上的 DCM 不支持自签名证书，因此此任务不适用于 IBM i 系统。

场景：

- 您具有需要安全通信的客户机 C1 和队列管理器 QM1。 您需要在 C1 和 QM1 之间执行相互认证。
- 您已决定使用自签名证书来测试安全通信。

生成的配置如下所示：

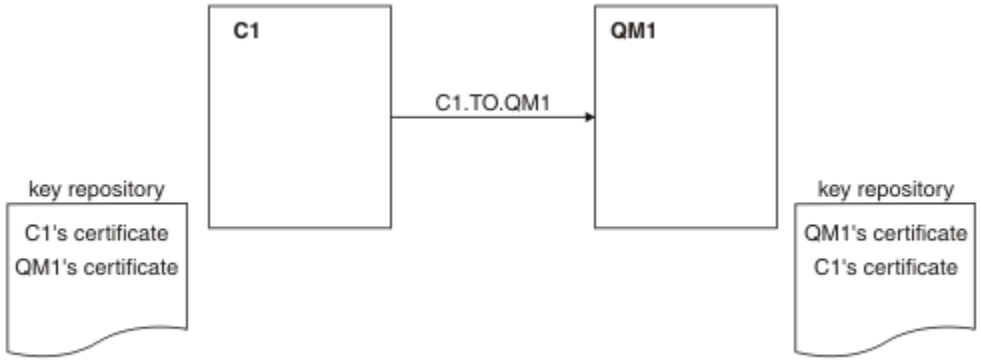


图 *19:* 由此任务生成的配置

在 第 93 页的图 19 中， QM1 的密钥存储库包含 QM1 的证书和来自 C1 的公用证书。 C1 的密钥存储库包含 C1 的证书以及来自 QM1 的公用证书。

过程

1. 根据操作系统，在客户机和队列管理器上准备密钥存储库：
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上 (仅限队列管理器)。
2. 为客户机和队列管理器创建自签名证书：
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上 (仅限队列管理器)。
3. 抽取每个证书的副本：
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上。
4. 使用 FTP 之类的实用程序将 C1 证书的公用部分传输到 QM1 系统，反之亦然。

   z/OS 对于 z/OS，请参阅 交换自签名证书。
5. 将伙伴证书添加到客户机和队列管理器的密钥存储库：

- ALW 在 AIX, Linux, and Windows 系统上。
- z/OS 在 z/OS 系统上。

6. 在队列管理器上发出命令 REFRESH SECURITY TYPE(SSL)。
7. 通过以下任一方式定义客户机连接通道:
   - 将 MQCONNX 调用与 C1 上的 MQSCO 结构配合使用，如 使用 MQCNO 在 IBM MQ MQI client 上创建客户机连接通道中所述。
   - 使用客户机通道定义表，如 在服务器上创建服务器连接和客户机连接定义中所述。
8. 在 QM1 上，通过发出类似于以下示例的命令来定义服务器连接通道:

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS from C1 to QM1')
```

   通道必须与您在步骤 6 中定义的客户机连接通道同名，并使用相同的 CipherSpec。

结果

将创建密钥存储库和通道，如 第 93 页的图 19 中所述。

下一步做什么

使用 **DISPLAY** 命令检查任务是否已成功完成。 如果任务成功，那么生成的输出与以下示例中显示的输出相似。

从队列管理器 QM1，输入以下命令:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(192.0.0.1)                      CURRENT
   SSLCERTI("CN=QM1,OU=IBM MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=IBM MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                           SUBSTATE(RECEIVE)
```

可选择设置通道定义的 **SSLPEER** 过滤器属性。 如果设置了通道定义 **SSLPEER**，那么其值必须与步骤 2 中创建的伙伴证书中的主体集 DN 相匹配。 成功连接后， **DISPLAY CHSTATUS** 输出中的 **SSLPEER** 字段将显示远程客户机证书的主题 DN。

使用 CA 签名的证书对客户机和队列管理器进行相互认证

遵循以下样本指示信息，通过使用 CA 签署的 TLS 证书在客户机和队列管理器之间实现相互认证。

关于此任务

场景:

- 您具有需要安全通信的客户机 C1 和队列管理器 QM1。 您需要在 C1 和 QM1 之间执行相互认证。
- 将来您计划在生产环境中使用此网络，因此您决定从一开始就使用 CA 签署的证书。

生成的配置如下所示:

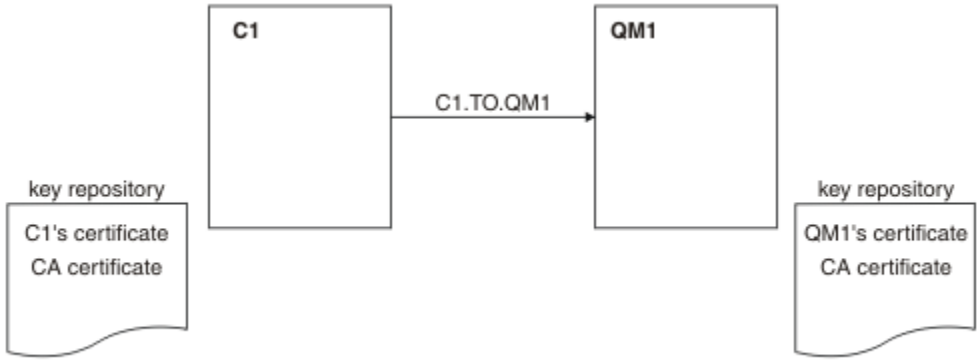


图 *20:* 由此任务生成的配置

在 第 95 页的图 20 中， C1 的密钥存储库包含 C1 的证书和 CA 证书。 QM1 的密钥存储库包含 QM1 的证书和 CA 证书。 在此示例中， C1 的证书和 QM1 的证书都由同一 CA 发放。 如果 C1 的证书和 QM1 的证书是由不同的 CA 发放的，那么 C1 和 QM1 的密钥存储库必须同时包含两个 CA 证书。

过程

1. 根据操作系统，在客户机和队列管理器上准备密钥存储库:
   - IBM i 在 IBM i 系统上。
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上 (仅限队列管理器)。
2. 请求客户机和队列管理器的 CA 签名证书。
   您可以对客户机和队列管理器使用不同的 CA。
   - IBM i 在 IBM i 系统上。
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上 (仅限队列管理器)。
3. 将认证中心证书添加到客户机和队列管理器的密钥存储库。
   如果客户机和队列管理器使用不同的认证中心，那么必须将每个认证中心的 CA 证书添加到两个密钥存储库。
   - IBM i 请勿在 IBM i 系统上执行此步骤。
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上 (仅限队列管理器)。
4. 将 CA 签名的证书接收到客户机和队列管理器的密钥存储库:
   - IBM i 在 IBM i 系统上。
   - ALW 在 AIX, Linux, and Windows 系统上。
   - z/OS 在 z/OS 系统上 (仅限队列管理器)。
5. 通过以下任一方式定义客户机连接通道:
   - 将 MQCONNX 调用与 C1 上的 MQSCO 结构配合使用，如 使用 MQCNO 在 IBM MQ MQI client 上创建客户机连接通道中所述。
   - 使用客户机通道定义表，如 在服务器上创建服务器连接和客户机连接定义中所述。

6. 在 QM1 上，通过发出类似于以下示例的命令来定义服务器连接通道:

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) SSLCAUTH(REQUIRED)
DESCR('Receiver channel using TLS from C1 to QM1')
```

通道必须与您在步骤 6 中定义的客户机连接通道同名，并使用相同的 CipherSpec。

结果

将创建密钥存储库和通道，如 第 95 页的图 20 中所述。

下一步做什么

使用 DISPLAY 命令检查任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出。

从队列管理器 QM1 中，输入以下命令:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
```

生成的输出类似于以下示例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(192.0.0.1)                      CURRENT
   SSLCERTI("CN=IBM MQ CA,OU=IBM MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=IBM MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

DISPLAY CHSTATUS 输出中的 SSLPEER 字段显示在步骤 2 中创建的远程客户机证书的主题 DN。 签发者名称与对步骤 4 中添加的个人证书进行签名的 CA 证书的主体集 DN 相匹配。

以匿名方式将客户机连接到队列管理器

遵循以下样本指示信息来修改具有相互认证的系统，以允许队列管理器以匿名方式连接到另一个队列管理器。

关于此任务

场景:

- 您的队列管理器和客户机 (QM1 和 C1) 已设置为与 第 94 页的『使用 CA 签名的证书对客户机和队列管理器进行相互认证』中一样。
- 您希望更改 C1 ，以便它以匿名方式连接到 QM1。

生成的配置如下所示:

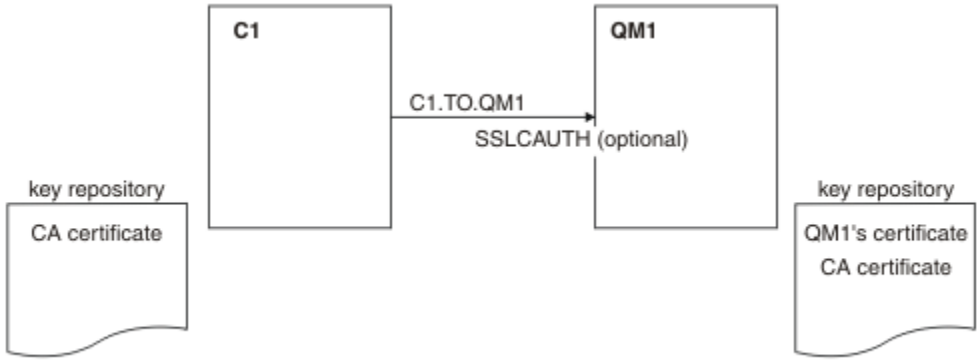


图 *21:* 允许匿名连接的客户机和队列管理器

过程

1. 根据操作系统，从 C1 的密钥存储库中除去个人证书:
   - IBM i IBM i 系统。
   - ALW AIX, Linux, and Windows 系统。

   证书标签是 `ibmwebspheremq` 后跟小写的登录用户标识或 **CERTLABL** 属性的值。 请参阅 数字证书标签 (Digital certificate labels)。
2. 重新启动客户机应用程序，或者使客户机应用程序关闭并重新打开所有 SSL/TLS 连接。
3. 通过发出以下命令，允许队列管理器上的匿名连接:

```
ALTER CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)
```

结果

密钥存储库和通道已更改，如 第 97 页的图 21 中所示

下一步做什么

在通道的服务器端，通道状态屏幕上存在对等名称参数值指示客户机证书已流动。

通过发出一些 DISPLAY 命令验证任务是否已成功完成。 如果任务成功，那么生成的输出类似于以下示例中显示的输出:

从队列管理器 QM1，输入以下命令:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

生成的输出将类似于以下示例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(192.0.0.1)                      CURRENT
   SSLCERTI( )                             SSLPEER( )
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

SSLCERTI 和 SSLPEER 字段为空，显示 C1 未发送证书。

Windows 在 Windows 上迁移

从现有 IBM MQ 9.2 安装开始，此方案将引导您完成将数据升级和迁移到 IBM MQ 9.3 所需的关键任务。 这两个版本都安装在同一 Windows 环境中。

Windows 规划解决方案

请查看本节中的主题，以了解此场景中涵盖的内容，企业可能希望遵循此场景的原因以及场景所建议的解决方案的概述。

相关任务

规划您的安装

Windows 假设前提

此场景对用于设置和使用样本 IT 配置的系统进行了若干假设。 这些假定包括您正在使用的产品的操作系统和版本，以及是否为 IBM MQ 配置了安全性。

此场景假定以下几点:

- 您正在将一台具有 Windows 操作系统的计算机用于此方案，您将在此方案上安装初始 IBM MQ 9.2 配置，然后安装 IBM MQ 9.3。

  注: 此场景中未描述集群。 提供了有关安装样本 IBM MQ 单服务器配置的指示信息，您可以将此配置用作尝试与最初开发方案相同的方案的起点。
- 您正在使用以下版本的 IBM MQ:
  - 对于初始样本配置，您正在使用 IBM MQ 9.2。
  - 对于迁移后配置，您正在使用 IBM MQ 9.3。
- 此场景不描述 IBM MQ 的安全性配置。 如果确实配置了安全性，那么您仍应该能够完成此方案。
- 您可以使用 Windows 命令提示符和图形用户界面 IBM MQ Explorer 来完成本场景中描述的任务。

相关概念

迁移路径

Windows 业务概述

公司希望将 Windows 操作系统上的现有 IBM MQ 9.2 IT 配置迁移到 IBM MQ 9.3。

该公司决定将其业务解决方案迁移到 IBM MQ 9.3 以获取业务价值，包括:

- 使用 IBM MQ 9.3 中提供的新功能和更新功能。
- 浏览 IBM MQ 9.3 中提供的新发行版类型; 持续交付发行版 (CDR)。
- 受益于 Windows 平台上的 LDAP 授权。

相关概念

IBM MQ 发行版类型和版本控制

相关信息

适用于长期支持和持续交付发行版的 IBM MQ 常见问题及回答

Windows 选择迁移路径

在 IBM MQ 9.2 和 IBM MQ 9.3 之间进行迁移时，可以采用若干迁移路径。

本主题概述了以下迁移路径:

- 单阶段路径 (也称为独立路径) ， 迁移
- 并排迁移路径
- 多阶段迁移路径

注: 此场景仅描述单阶段和并行迁移方法。

请考虑每条路径的优点和限制，以确定哪些路径最适合您的需求:

单阶段迁移

在单阶段迁移中，产品的最新版本的安装将替换同一安装位置中的较早版本。

单阶段迁移的优点是它尽可能少地更改较早版本上队列管理器的配置。 现有应用程序从装入较早版本的库切换到自动装入最新版本的库。 使用此方法，系统将不可用于整个过程。

并行迁移

在并行迁移中，您将安装最新版本的 IBM MQ 以及继续与较早版本关联的队列管理器。

准备就绪后，将队列管理器和应用程序迁移到最新版本。

使用此方法，因为您在启动任何队列管理器之前卸载较早版本，所以可以将最新版本的安装指定为主安装。

请参阅 选择主安装 以获取更多信息。

多阶段迁移

在多阶段迁移中，您将安装产品的最新版本以及继续与较早版本关联的正在运行的队列管理器。 您可以使用最新版本的安装来创建队列管理器并运行新应用程序。 当您准备好从较早版本开始迁移队列管理器和应用程序时，可以逐个执行此操作。 迁移到最新版本完成后，请卸载先前版本，并使最新版本安装成为主安装。

使用多阶段方法，在卸载较低版本之前，必须配置环境以运行将队列管理器连接到最新版本的应用程序。 您还必须提供用于运行 IBM MQ 命令的路径。 这两项任务都是使用 **setmqenv** 命令完成的。

相关概念

迁移概念和方法概述

相关任务

在 AIX and Linux 上迁移: 单阶段

在 AIX and Linux 上迁移: 并排

在 AIX and Linux 上迁移: 多阶段

相关参考

影响迁移的更改列表

Windows 技术解决方案

此场景描述了从较低版本的 IBM MQ 迁移到较高版本的两种方法，其中两个版本都在 Windows 操作系统上运行，并且位于同一服务器上。

Windows 概述: 初始 *IT* 配置

公司使用安装在具有 Windows 操作系统的服务器上的 IBM MQ 9.2 提供的现有 IT 配置。 此场景描述将初始 IT 配置迁移到同一服务器上的 IBM MQ 9.3 提供的等效 IT 配置。

初始 IT 配置包含管理员配置或使用的多个组件，如 第 100 页的图 22 中所示:

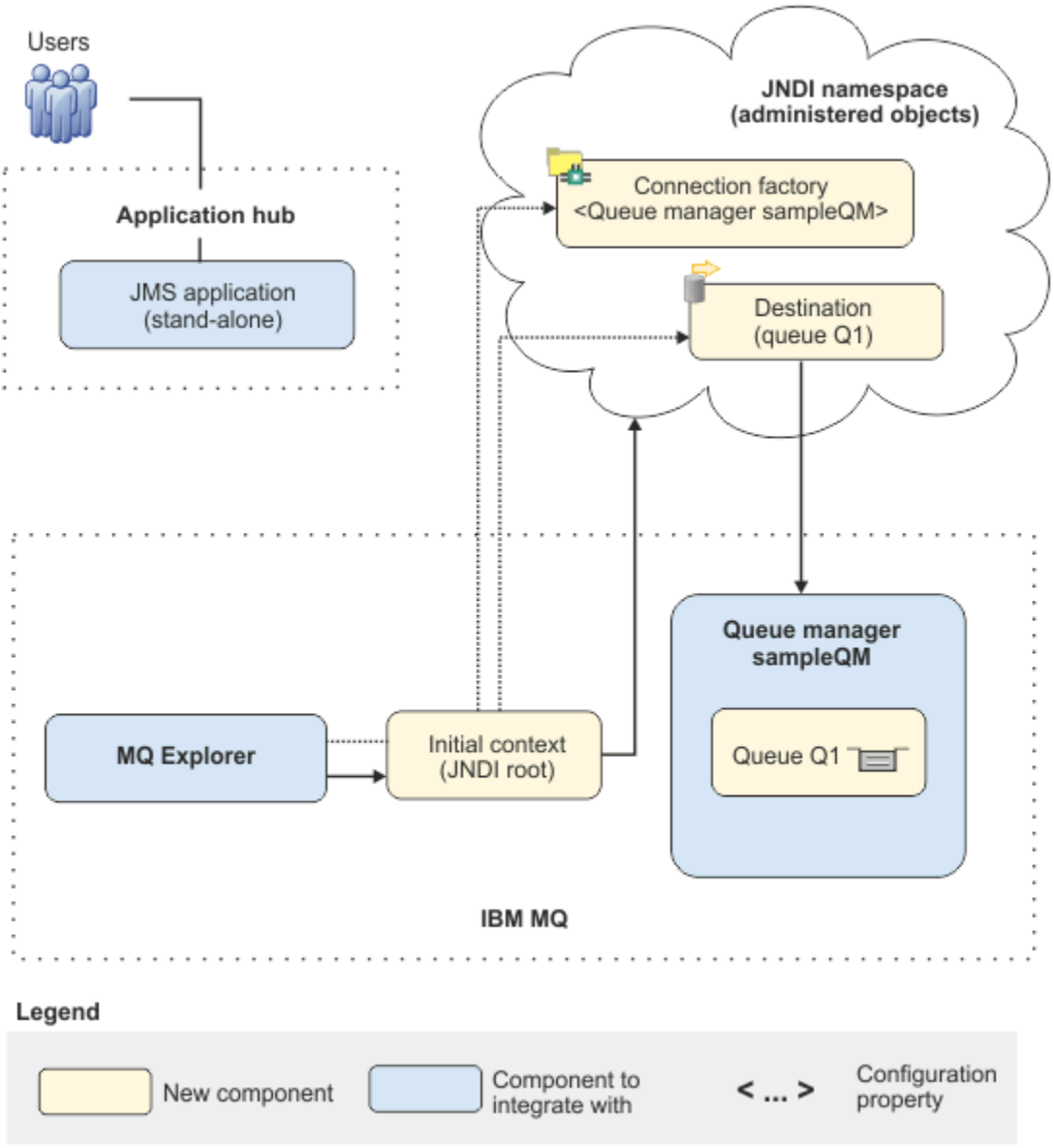


图 *22:* 初始 *IT* 配置

JMS 应用程序

业务用户与之交互的独立应用程序，例如，用于注册订单。 应用程序将 Java Message Service (JMS) 用于异步消息传递。

- JMS 是受广泛支持的 Java EE 消息传递标准。 因此，基于 JMS 的应用程序可跨许多消息传递产品进行移植。
- JMS 提供了消息传递层详细信息中的抽象级别，从而简化了应用程序开发过程。
- JMS 提供异步通信，使应用程序无需等待应答即可运行，这与紧密耦合的系统 (例如远程过程调用 (RPC)) 不同。
- 使用 JMS 的应用程序不会直接指定用于访问资源的详细信息。 而是查找并使用受管 JMS 对象，例如连接工厂和目标。

对于某些情况，其他消息传递标准可能比 JMS 更适合。 例如， IBM Message Service Client for C ， C++ 和 .NET(也称为 XMS) 是提供与 JMS for non-Java 应用程序类似的优点的 API。 因此，如果您正在使用 .NET 平台，或者要将现有 C++ 应用程序与较新的 Java EE 应用程序集成，那么 XMS 更适合。

应用程序使用点到点消息传递将消息发送到基础结构中的队列，并处理应答消息，以向业务用户提供相应的响应。

在此消息传递模型中，应用程序将消息发送到队列，而另一个应用程序从队列接收消息并确认接收消息。 此模型是最简单的消息传递形式，因为它只涉及两个端点。 此模型也最适合于方案样本应用程序: 单个客户机请求来自单个服务器的信息。

在备用消息传递模型 "发布/预订" 中，发布者将消息发布到消息主题。 订户预订主题以接收消息。 发布者和订户之间没有任何相互相关的信息，零个或零个以上的收件人接收到该消息。

队列管理器 sampleQM
提供初始消息传递基础结构的 IBM MQ 队列管理器。 它托管 JMS 应用程序使用的队列。

Q1 [消息队列]
JMS 应用程序将消息发送到的 IBM MQ 队列。

JNDI 名称空间
Java Naming Directory Interface JNDI 名称空间用于保存 JMS 受管对象，应用程序可以使用这些对象连接到 IBM MQ 并访问目标以发送或接收消息。

JNDI 是 Java EE 的一部分，为应用程序提供了一种标准方法来访问各种类型的命名和目录服务，以检索应用程序组件。 例如，可以使用 JNDI 来访问文件系统上的命名服务以检索打印机对象的位置，或者访问 LDAP 服务器上的目录服务以检索包含标识和密码信息的用户对象。 因此， JNDI 增强了基于 JMS 的应用程序的可移植性，并使这些应用程序更容易相互集成并集成到现有系统中。 对于 JMS 消息传递，您可以使用 JNDI 来存储表示消息的目标目标的对象，或者存储用于在应用程序与其消息传递目标之间创建连接的连接工厂。

任何有权访问 JNDI 名称空间的应用程序或进程都可以使用相同的受管对象。 可以在 JNDI 中更改受管对象的属性，所有应用程序或进程都能够从这些相同的更改中获益。

初始上下文 (initial context)
初始上下文定义 JNDI 名称空间的根。 要使用 IBM MQ Explorer 来创建和配置受管对象，请首先添加用于定义 JNDI 名称空间的根的初始上下文。 同样， JMS 应用程序首先获取初始上下文，然后才能从 JNDI 名称空间中检索受管对象。

连接工厂， myCF
JMS 连接工厂对象定义了一组用于连接的标准配置属性。 应用程序使用连接工厂来创建与 IBM MQ 的连接。

目标， myQueue
JMS 目标可以是主题或队列。 在此场景中，目标是一个队列，并标识应用程序向其发送消息和/或从中接收消息的 IBM MQ 队列。 应用程序在 JNDI 名称空间中查找目标以创建与 IBM MQ 队列的连接。

Windows 概述: 交付的逻辑拓扑

公司已从 IBM MQ 9.2 迁移到 IBM MQ 9.3。

迁移后的 IT 配置保持不变，如 第 102 页的图 23 中所示。 将迁移队列管理器和样本应用程序，并且业务可从最新版本的 IBM MQ 中的新功能中获益。

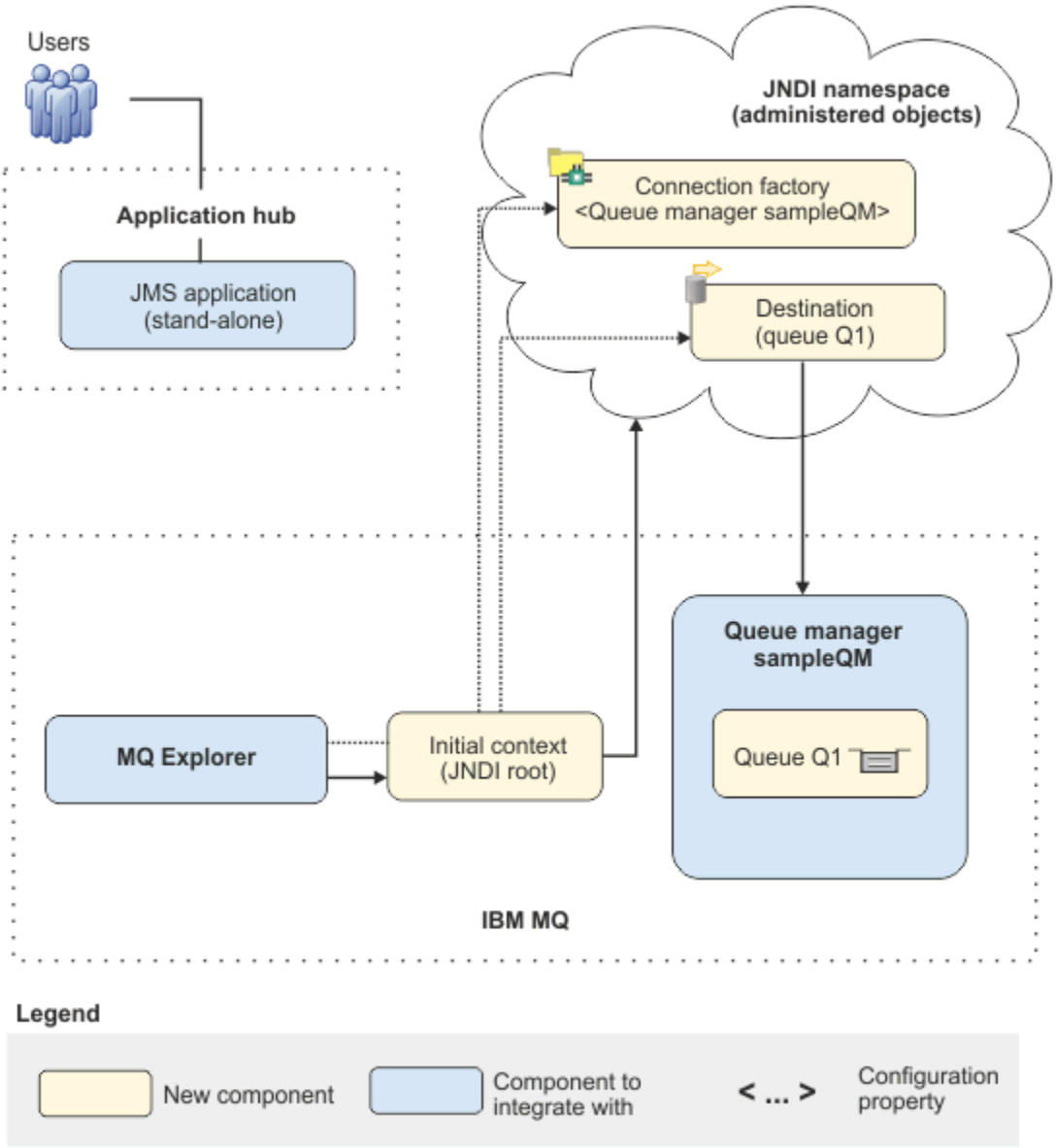


图 *23:* 交付的 *IT* 配置

Windows 使用图形用户界面实现解决方案

在此场景中实施解决方案涉及使用图形用户界面从在 Windows 操作系统上运行的 IBM MQ 的较早发行版迁移到也在 Windows 操作系统上运行的较高发行版，然后验证队列管理器和队列是否已成功迁移到较高发行版。

开始之前

如果要试用此方案，请首先遵循指示信息来设置样本消息传递基础结构的副本，如 第 103 页的『创建初始 IT 配置』中所述。 此样本配置基于 IBM MQ 9.2。

关于此任务

此场景中描述的迁移过程显示了如何将队列管理器从 IBM MQ 9.2 迁移到 IBM MQ 9.3。

此方案包含迁移方法的两个选项。 您可以选择试用这些选项中的一个或其他选项，也可以同时试用这两个选项。

选项 1: 单阶段迁移

在单阶段迁移中，产品的更高版本的安装将替换同一安装位置中的较低版本。

单阶段迁移的优点是它尽可能少地更改较早版本上队列管理器的配置。 现有应用程序从装入较早版本的库切换到自动装入最新版本的库。 使用此方法，系统将不可用于整个过程。

选项 2: 并行迁移

在并行迁移中，将 IBM MQ 的更高版本与较低版本一起安装。 队列管理器将继续与较低版本相关联，直到您准备好将它们迁移到较高版本为止。

通过并行方法，因为在启动更高版本中的任何已迁移队列管理器之前先卸载较低版本，所以可以将更高版本的安装指定为主安装。

过程

1. 创建样本初始 IT 配置以用作方案的起点，如 第 103 页的『创建初始 IT 配置』中所述。
2. 选择要用于迁移产品的方法，然后遵循所选选项的指示信息:
   - 第 111 页的『选项 1: 单阶段迁移』
   - 第 117 页的『选项 2: 并行迁移』

相关任务

在 AIX and Linux 上迁移: 单阶段

在 AIX and Linux 上迁移: 并排

将队列管理器从先前版本迁移到 Windows 上的最新版本

选择主安装

Windows 创建初始 IT 配置

此方案是使用样本初始 (IT) 配置开发的。 遵循指示信息来设置此样本配置，以便以与最初开发的方法相同的方式试用该方案。

关于此任务

此方案的初始 IT 配置 (在 第 99 页的『概述: 初始 IT 配置』中进行了描述) 包含为 IBM MQ Explorer 添加的初始上下文，用于连接到 JNDI 名称空间的根。 JNDI 名称空间包含为样本 JMS 应用程序添加的用于连接到 IBM MQ 的连接工厂，以及为样本 JMS 应用程序添加的用于连接到 IBM MQ 队列的目标。 该 IBM MQ 队列也已添加到初始 IT 配置中，并由样本 JMS 应用程序使用。

过程

1. 安装 IBM MQ 9.2 并验证安装。
2. 配置 JNDI 名称空间和受管对象。
3. 验证样本 IT 配置。

Windows 使用启动板安装 *IBM MQ 9.3*

使用安装启动板和向导来安装要设置为初始 IT 配置以用作此方案起点的 IBM MQ 版本。

开始之前

在启动此任务之前，请完成以下检查:

- 安装时必须具有本地管理员权限。
- 确保机器名不包含任何空格。
- 确保您有足够的磁盘空间。 有关更多信息，请参阅 Multiplatforms 版上的磁盘空间需求。

对于此场景，无需确定是否需要为任何 IBM MQ 用户定义 Windows 域用户标识，因为此需求不在此场景的范围内。 请参阅 为 IBM MQ 创建 Active Directory 和 DNS 域以获取更多信息。

在安装 IBM MQ 之前，请检查系统是否满足硬件和软件需求。 有关硬件和软件需求的最新详细信息，请参阅系统要求 IBM MQ。

关于此任务

此任务描述在 Windows 操作系统上安装 IBM MQ 的基本步骤。

安装程序包含指向更多信息的链接。 安装过程包含以下部分:

1. 启动安装过程。
2. 使用安装启动板来检查和安装软件需求，指定网络信息，然后启动 IBM MQ 安装向导。
3. 使用 IBM MQ 安装向导来安装软件，然后启动 Prepare IBM MQ Wizard。
4. 使用 Prepare IBM MQ Wizard 来启动 IBM MQ 服务。

过程

1. 启动安装过程。

   在 Windows Explorer 中，浏览至安装映像下载到的临时文件夹，然后双击 `setup.exe`。

   安装启动板已启动。
2. 使用启动板，复审并在必要时修改软件需求和网络配置。
   a) 单击 **软件需求** 按钮以显示 **软件需求** 选项卡。
   b) 检查是否已满足软件需求，以及需求条目是否显示带有 OK 字样的绿色刻度标记。 进行任何指示的更正。

      注: 要查看有关任何需求的更多详细信息，请单击加号 (+) 按钮。
   c) 单击 **网络配置** 按钮以显示 **网络配置** 选项卡。
   d) 单击 **否** 单选按钮。

      注: 此场景假定您不需要为 IBM MQ 配置域用户标识。 有关配置 IBM MQ for Windows 域用户的更多信息，请单击 **更多信息** 按钮。
   e) 在启动板的 **IBM MQ 安装** 选项卡上，选择安装语言，然后单击 **启动 IBM MQ 安装程序** 按钮以启动 IBM MQ 安装向导。

   您已完成安装需求的 IBM MQ 复审，进行了任何必需的修改，并已启动 IBM MQ 安装向导。
3. 使用 IBM MQ 安装向导来安装软件，并启动 Prepare IBM MQ Wizard。
   a) 在 IBM MQ 安装向导中，阅读 "许可协议" 并选中 **我接受许可协议中的条款** 复选框，然后单击 **下一步**。
   b) 单击 **典型**，然后单击 **下一步**。
   c) 在 " **准备安装 IBM MQ** " 页面上，查看安装信息，然后单击 **安装**。

      注: 以下详细信息:

      - 安装名称
      - 程序文件的顶级文件夹
      - 数据文件的顶级文件夹

      将安装以下功能部件:

      - IBM MQ 服务器
      - IBM MQ: 用于管理和监视 IBM MQ 资源的图形界面
      - Java™ 和 .NET 消息传递和 Web Service
      - IBM MQ 开发工具箱

      安装过程开始。 根据您的系统，安装过程可能需要几分钟时间。

在安装过程结束时， " IBM MQ 安装" 窗口将显示消息 `Installation Wizard Completed Successfully` 。

d) 单击**完成**。

您已成功安装 IBM MQ。 Prepare IBM MQ Wizard 会自动启动，显示 **欢迎使用 Prepare IBM MQ Wizard** 页面。

4. 使用 "准备 MQ " 向导来启动 IBM MQ 服务。

a) 在 "欢迎使用 Prepare IBM MQ Wizard " 页面上，选择 **下一步**。

Prepare IBM MQ Wizard 显示消息 `Status: Checking IBM MQ Configuration` 和进度条。 完成此过程后，将显示 " IBM MQ 网络配置" 页面。

b) 在 Prepare IBM MQ Wizard 的 " IBM MQ 网络配置" 页面上，选择 **否**。

c) 单击**下一步**。

Prepare IBM MQ Wizard 显示消息 `Status: starting the IBM MQ Service` 和进度条。 完成此过程后，向导将显示 "完成 Prepare IBM MQ Wizard " 页面。

d) 选择 **启动 IBM MQ Explorer** 并选择是否启动记事本以查看发行说明，然后单击 **完成**。

IBM MQ Explorer 。

您已安装 IBM MQ，并且已启动 IBM MQ Explorer。

结果

IBM MQ 已安装在您的计算机上。

下一步做什么

您已准备好创建此场景中使用的受管对象，如 第 105 页的『配置 JNDI 名称空间和受管对象』中所述。

相关概念

Windows 系统上的硬件和软件需求

IBM MQ 简介

相关任务

在 Windows 上安装 IBM MQ 服务器

配置 IBM MQ 服务器

Windows 配置 *JNDI* 名称空间和受管对象

在 IBM MQ Explorer 中定义 JNDI 名称空间的初始上下文，然后在名称空间中定义样本应用程序可以使用的受管对象。

关于此任务

在此任务中，您将在 IBM MQ 中创建以下对象。

- 位于本地文件系统中的 JNDI 名称空间。 使用文件系统，因为它是样本方案的最简单 JNDI 机制。

  JNDI 名称空间可以在文件系统，轻量级目录访问协议 (LDAP) 服务器或其他 JNDI 实现上。 如果要在 LDAP 服务器或其他 JNDI 实现上使用 JNDI 名称空间，那么必须配置 JNDI 名称空间并修改样本应用程序以引用该实现所需的 JNDI 名称空间。
- JNDI 名称空间中的受管对象。 JMS 应用程序可以查找受管对象以连接到 IBM MQ 并访问用于发送或接收消息的 IBM MQ 目标。

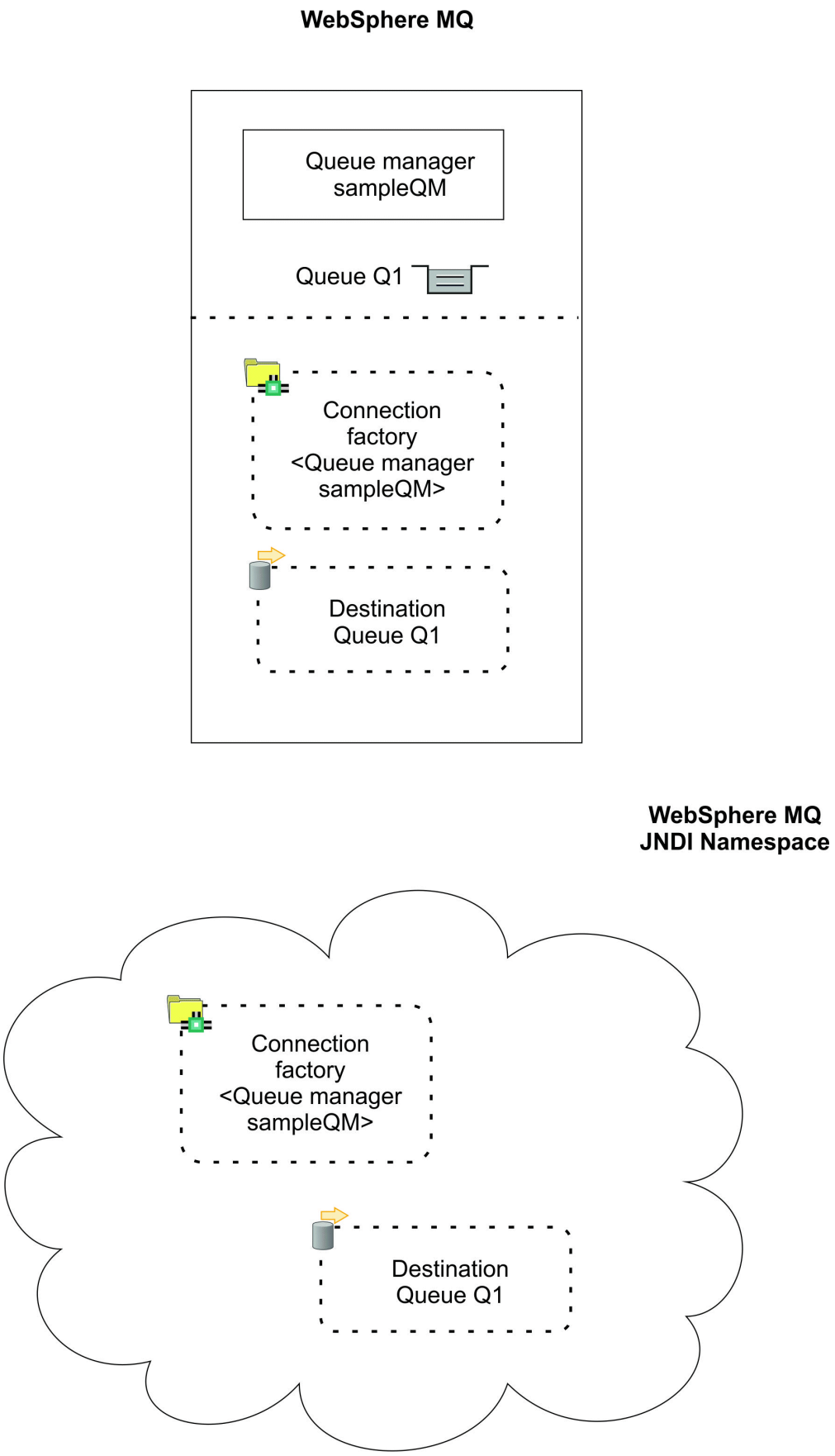


图 *24:* 在 *IBM MQ* 中创建的对象

过程

1. 通过单击 **启动** > **所有程序** > **IBM MQ** > **IBM MQ Explorer** 来启动 IBM MQ Explorer(如果尚未启动)。
   如果 IBM MQ Explorer 正在运行并显示 "欢迎" 页面，请关闭 "欢迎" 页面以开始管理 IBM MQ 对象。
2. 创建要用于样本应用程序的队列管理器。
   a) 右键单击 **队列管理器** ，然后选择 **新建** > **队列管理器 ...**。
   "创建队列管理器" 向导将启动。
   b) 在 **队列管理器名称** 字段中，输入 `sampleQM`。
   您可以为队列管理器选择其他名称，但必须记住在以后的配置步骤中使用该名称，以代替 `sampleQM`。

   注: 该名称不得超过 48 个字符，来自以下集合:

   - 大写或小写字符 (A-Z a-z)
   - 数字字符 (0-9)
   - 句点 (.)
   - 正斜杠 (/)
   - 下划线 (_)
   - 百分号 (%)

   名称区分大小写。 同一类型的对象必须具有不同的名称。 例如，两个队列不能具有相同的名称，但队列管理器和队列可以。
   c) 在 **死信队列** 字段中，输入 `SYSTEM.DEAD.LETTER.QUEUE`。
   此字段是创建队列管理器时自动创建的死信队列的名称。
   死信队列存储无法传递到其正确目标的消息，例如，因为队列已满。 所有队列管理器都应该具有关联的死信队列。
   d) 将其他字段留空，然后单击 **完成**，如果该按钮处于禁用状态，请单击 **下一步**。
   如果端口号与现有队列管理器 (例如，作为缺省配置的一部分创建的队列管理器) 冲突，那么将禁用 **完成** 按钮。 您必须继续完成向导以更改缺省端口号。
   e) 如果单击了 **下一步**，请继续接受缺省值，并在每个页面上单击 **下一步** ，直到到达向导的最终页面 (当 **完成** 按钮变为可用时)。 更改指定的端口号 (例如，更改为 1415) ，然后单击 **完成**。
   创建并启动队列管理器时， IBM MQ 将显示一个对话框窗口。
3. 为 *JNDI* 名称空间 添加初始上下文，然后将 IBM MQ Explorer 连接到该上下文
   必须先添加初始上下文以定义存储受管对象的 JNDI 名称空间的根，然后才能使用 IBM MQ Explorer 来创建和配置 JMS 受管对象。

   无论何时要使用 IBM MQ Explorer 在 JNDI 名称空间中创建或管理受管对象，都必须将 IBM MQ Explorer 连接到 JNDI 名称空间的初始上下文。

   a) 在 IBM MQ Explorer - **"Navigator " 窗格**中，右键单击 **JMS 受管对象**，然后选择 **添加初始上下文 ...**。
   此操作将显示 **“ 连接详细信息 ”** 页面。
   b) 在 **“ JNDI 名称空间位于何处? ”**下， 选中 **文件系统** 复选框。
   c) 在 **绑定目录** 字段中，输入 `C:\JNDI-Directory`。
   此值与样本 JMS 应用程序中指定的 JNDI 名称空间位置相匹配。 如果必须指定其他 JNDI 目录，那么必须修改应用程序以进行匹配。
   如果系统上不存在该目录，那么窗口将显示消息 `Specified location does not exist or is not readable`。 单击 **浏览 ...** 要打开文件系统窗口，请浏览至 `Local Disk (C:)`，然后单击 **新建文件夹** 以创建 `JNDI-Directory` 文件夹。 单击**确定**。

   单击**下一步**。
   d) 在 " **用户首选项** " 页面上，保留缺省设置。

   - **上下文昵称**: JNDI 名称空间的位置将用作昵称以在 IBM MQ Explorer 中显示初始上下文。

- **完成时立即连接**: 此选项在完成创建初始上下文时将 IBM MQ Explorer 连接到 JNDI 名称空间，以便您可以立即创建受管对象。
- **在启动时自动重新连接到上下文**: 未选择此选项，因为通常您不需要 IBM MQ Explorer 在每次关闭并重新打开时自动重新连接到初始上下文 IBM MQ Explorer。

如果通常使用 IBM MQ Explorer 在 JNDI 名称空间中创建或管理受管对象，那么可以选中 **在启动时自动重新连接到上下文** 复选框，以使 IBM MQ Explorer 在启动时自动重新连接到初始上下文 IBM MQ Explorer 。 此选项使您不必手动将 IBM MQ Explorer 连接到初始上下文。

单击 **完成** 以创建并显示初始上下文。

4. 创建连接工厂受管对象。

连接工厂受管对象定义了一组用于连接的标准配置属性。 应用程序使用连接工厂来创建与 IBM MQ 的连接。

a) 在 IBM MQ Explorer - **Navigator** 窗格中，展开 **JMS 受管对象**，然后展开标记为 **file:/C: /JNDI-Directory/**的初始上下文。

b) 右键单击 **连接工厂**，然后选择 **新建** > **连接工厂 ...**。

此操作将显示 " **新建连接工厂** " 向导

c) 在 "名称" 字段中，输入 myCF

样本 JMS 应用程序包含用于查找名称为 myCF 的连接工厂的代码。 如果必须使用其他名称，那么必须修改应用程序以进行匹配。

IBM MQ 用于消息传递提供程序，因为样本应用程序使用点到点消息传递。

单击**下一步**。

d) 将连接工厂类型保留为 **连接工厂**，因为此选项对于常规 JMS 使用最为灵活。

独立于域的连接工厂使 JMS 应用程序能够同时使用点到点消息传递和发布/预订消息传递，尤其是当您希望 JMS 应用程序在同一事务下执行两种类型的消息传递时。

如果 JMS 应用程序旨在仅使用点到点消息传递或仅使用发布/预订消息传递，那么您可以在创建连接工厂时选择特定消息传递域，并创建特定于域的 (队列或主题) 连接工厂。

e) 将对 XA 事务的支持保留为未选中状态。

样本应用程序不使用符合 XA 的事务。

IBM MQ JMS 支持绑定方式下符合 XA 的事务。 如果希望样本应用程序使用符合 XA 的事务，那么必须修改样本应用程序。

单击**下一步**。

f) 将传输保留为 **绑定**。

使用连接工厂的样本 JMS 应用程序与队列管理器在同一计算机上运行，因此可以使用绑定方式传输。此选项表示 JMS 应用程序直接连接到队列管理器，并提供比备用客户机方式的性能优势。

单击 **下一步**，然后再次单击 **下一步** 。

g) 在 " **更改属性** " 页面上，从左侧菜单中选择 **连接** ，然后在 " **连接** " 窗格中选择 sampleQM 作为 **基本队列管理器**。

基本队列管理器是应用程序将连接到的队列管理器。 如果您希望应用程序能够连接到多个队列管理器，请将此值留空。

h) 单击**完成**。

IBM MQ 将显示一个对话框窗口，以显示已成功创建对象。 单击 **确定** 以关闭对话框窗口。

5. 创建目标受管对象。

目标受管对象标识应用程序向其发送消息和/或从中接收消息的 IBM MQ 队列。 应用程序在 JNDI 名称空间中查找目标以创建与 IBM MQ 队列的连接。

在发布/预订消息传递中，目标标识主题而不是队列。

a) 在 IBM MQ Explorer - **Navigator** 窗格中，展开 **JMS 受管对象**，然后展开标记为 **file:/C: /JNDI-Directory/**的初始上下文。

b) 右键单击 **目标**，然后选择 **新建** > **目标 ...**。

此时将显示 " **新建目标** " 向导。

c) 在 **名称** 字段中，输入 myQueue。

将 **类型** 保留为 **队列**。

样本 JMS 应用程序包含用于查找名称为 myQueue 的目标的代码。 样本 JMS 应用程序使用点到点消息传递，因此需要类型为队列的目标。 类型为主题的目标用于发布/预订消息传递。

d) 选中 **启动向导以创建匹配的 MQ 队列** 复选框。

目标对象需要匹配的 IBM MQ 队列，并且可以方便地使用 IBM MQ Explorer 来同时创建这两个队列。 完成 " **新建目标** " 向导时，将打开 " **创建 MQ 队列** " 向导，其中许多目标详细信息映射到 IBM MQ 队列。

单击**下一步**。

再次单击**下一步**。

e) 在 " **“ 更改属性 ”** " 页面上，单击 **选择 ...** 位于 **队列管理器**旁边。 选择先前创建的 sampleQM 队列管理器，然后单击 **确定**。

f) 指定 Q1 作为 IBM MQ 队列的名称。

您可以为队列选择其他名称，但必须记住在以后的配置步骤中使用该名称来代替 Q1。

注: 该名称不得超过 48 个字符，来自以下集合:

- 大写或小写字符 (A-Z a-z)
- 数字字符 (0-9)
- 句点 (.)
- 正斜杠 (/)
- 下划线 (_)
- 百分号 (%)

名称区分大小写。 同一类型的对象必须具有不同的名称。 例如，两个队列不能具有相同的名称，但队列管理器和队列可以。

g) 单击**完成**。

此时将启动 " **创建 MQ 队列** " 向导。

如果向导未启动，那么您可能未在先前步骤中选中 **启动向导以创建匹配的 MQ 队列** 复选框。 在 IBM MQ Explorer - **Navigator** 窗格中，展开 **sampleQM** 队列管理器，右键单击 **队列**，然后选择 **新建** > **本地队列 ...**。

6. 创建匹配的 IBM MQ 队列。

先前创建的目标受管对象表示 IBM MQ 队列。 此队列是存储 JMS 消息的位置。

a) 单击 **下一步** 以接受先前指定的 sampleQM 队列管理器。

b) 单击**下一步**。

c) 单击 **完成** 以使用先前创建的目标受管对象中的信息来创建 IBM MQ 队列。

IBM MQ 显示一个对话框窗口，其中包含成功创建对象的消息。

现在，新队列在队列管理器下的 **队列** 部分中可视。

结果

现在，您已创建使用样本 JMS 应用程序所需的 IBM MQ 对象。

下一步做什么

现在，您已准备好验证是否已正确配置 IBM MQ 以用于样本应用程序，如 第 110 页的『验证样本 IT 配置』中所述。

Windows 验证样本 *IT* 配置

运行样本独立 JMS 应用程序以通过 IBM MQ 发送和接收消息，并验证是否已正确配置 IBM MQ 以用于样本应用程序。

开始之前

下载样本应用程序包。 单击以下链接并将文件保存到要安装 IBM MQ 的计算机: sampleJMSApp.zip，然后解压缩内容。 该软件包包含用于运行应用程序的样本 JMS 应用程序 `.jar` 文件和批处理文件。

- 样本 `sampleJMSApp.jar` 文件和 `.cmd` 文件必须位于同一目录中。
- `.cmd` 文件使用环境变量来设置用于运行 JMS 应用程序的类路径。 运行 JMS 应用程序时，如果看到 Java `java.lang.NoClassDefFoundError`，那么可能需要调整命令文件中的类路径行。

关于此任务

JMS 应用程序由发送初始消息的请求者客户机和接收消息并发送应答的响应者客户机组成。 提供的批处理文件执行以下操作:

- `runresponder.cmd` 将打开命令提示符窗口，响应程序客户机将在该窗口中启动，然后等待消息。
- `runrequester.cmd` 打开单独的命令提示符窗口，请求者客户机在该窗口中启动，然后发送请求消息并接收应答。

通过两个命令提示符窗口，您可以分别更清楚地看到请求者和响应者的操作。

过程

1. 双击 `runresponder.cmd` 文件。

   在标记为 **响应程序窗口**的命令提示符窗口中，响应程序客户机将启动，然后等待消息。

   ```
   > Connection factory located in JNDI.> Destination located in JNDI.> Creating connection to
   QueueManager.> Created connection.
   > Waiting for message.
   ```

2. 双击 `runrequester.cmd` 文件。

   在 "**请求者**" 窗口中，观察请求者消息。 在 "**响应程序**" 窗口中，观察更新的响应程序消息; 它从请求者客户机接收到的消息以及它发送的应答消息。

结果

在标记为 **请求者窗口**的命令提示符窗口中，请求者客户机显示连接状态，发送的消息，然后显示从响应者客户机接收到的应答消息:

```
> Connection factory located in JNDI.> Destination located in JNDI.> Creating connection to
QueueManager.> Connection created.
> Sending stock request for 'BakedBeans'> Sent Message
ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02
> Received Message ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902 for 'B
akedBeans - 15 tins in stock'
> Closing connection to QueueManager.> Closed Connection.
-----------------------------------------------------
In this window, observe the messages sent through IBM MQ:
- The request message sent
- The reply message received
-----
When ready, press any key to close this window
Press any key to continue . . .
```

在 "**响应程序**" 窗口中，观察已更新的响应程序消息; 它从请求者客户机接收到的消息以及它发送的应答消息:

```
> Connection factory located in JNDI.> Destination located in JNDI.> Creating connection to
QueueManager.> Created connection.
> Waiting for message.

> Received Message ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02 for 'B
akedBeans'
> Sending Reply Message 'BakedBeans - 15 tins in stock'> Sent Message
ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902
```

```
> Closing connection to QueueManager.> Closed connection.
------------------------------------------------------
In this window, observe the updated responder messages
- The request message received (from the requester)
- The reply message sent
-----
When ready, press any key to close this window
Press any key to continue . . .
```

两个命令窗口中显示的消息验证样本应用程序的请求者和响应程序客户机是否可以通过 IBM MQ 相互通信。

下一步做什么

现在，您已准备好开始使用以下两个迁移选项之一将样本 IBM MQ 9.2 安装迁移到 IBM MQ 的更高发行版:

- 要使用单阶段迁移方法进行迁移，请遵循 第 111 页的『选项 1: 单阶段迁移』中的指示信息。
- 要使用并行迁移方法进行迁移，请遵循 第 117 页的『选项 2: 并行迁移』中的指示信息。

选项 1: 单阶段迁移

此方案的选项 1 显示了在使用单阶段迁移方法时如何从较早版本的 IBM MQ 迁移到更高版本。 通过单阶段迁移，更高版本的 IBM MQ 的安装将替换同一安装位置中的较低版本。

开始之前

此方案的起点是 第 99 页的『概述: 初始 IT 配置』中描述的初始 IT 配置。

在启动此任务之前，请遵循 第 103 页的『创建初始 IT 配置』 中的指示信息来设置初始 IT 配置。

关于此任务

通过单阶段迁移，您可以选择在安装较新版本之前卸载较低版本的 IBM MQ ，或者在不首先卸载较低版本的情况下安装较新版本 (即迁移到位)。 在这两种情况下，较新发行版都安装在与较早发行版相同的目录中。 此方案的选项 1 显示了在安装较新版本之前卸载较旧版本的单阶段迁移。 在卸载过程中不会除去队列管理器数据，这意味着将保留此场景中使用的样本队列管理器，并在您安装较新版本的 IBM MQ 时进行检测。

过程

1. 停止队列管理器 在较低版本的 IBM MQ 中运行，并备份队列管理器数据。
2. 卸载要从中迁移的较低版本的 IBM MQ ，而不除去队列管理器数据。
3. 使用启动板 安装 IBM MQ 9.3 。
4. 使用 IBM MQ Explorer 来验证新的 IBM MQ 9.3 安装。

   验证队列管理器是否已从先前发行版成功迁移，以及您是否可以将消息放入已迁移的队列中并从这些队列中获取消息。

相关任务

在 AIX and Linux 上迁移: 单阶段

Windows 准备迁移

在迁移到更高版本的 IBM MQ 之前，必须首先停止队列管理器并备份队列管理器数据。

关于此任务

如果在未首先备份系统的情况下从先前版本的 IBM MQ 迁移，那么如果决定不继续迁移，那么无法还原到先前版本。 在安装新版本之前备份系统使您能够在必要时回退升级。 但是，如果执行了回退升级，那么无法恢复由更高版本的 IBM MQ 执行的任何工作，例如对消息和对象的更改。

在执行备份之前，请停止要备份的队列管理器，对于此场景，此队列管理器为 sampleQM。 如果尝试对正在运行的队列管理器进行备份，那么备份可能不一致，因为复制文件时正在进行任何更新。

过程

1. 打开 IBM MQ Explorer。

   单击 **启动** > **所有应用程序** > **IBM MQ** > **IBM MQ Explorer**。
2. 停止队列管理器 sampleQM。

   a) 在 " Navigator " 视图中，右键单击队列管理器 sampleQM。

   b) 单击**停止**。

   此时将打开 " **结束队列管理器** " 窗口。

   c) 选择 **受控**，然后单击 **确定**。

   选择 **受控** 选项将以受控的有序方式停止队列管理器。 强制队列管理器停止的 **立即** 选项通常仅在受控停止未能成功完成时使用。

   队列管理器停止。 在 IBM MQ 中，队列管理器 sampleQM 旁边的图标已更改为包含指向下方的红色箭头。
3. 关闭 IBM MQ Explorer。
4. 备份队列管理器数据。

   复制以下所有数据，确保包含所有备份目录。 某些目录可能为空，但如果需要在以后复原备份，那么您需要所有这些目录，因此也请保存这些目录。

   - 位于 `C:\ProgramData\ IBM \ MQ\Qmgrs` 中的队列管理器数据。
   - 位于 `C:\ProgramData\ IBM \ MQ\log` 中的队列管理器的日志文件目录，包括日志控制文件 `amqhlctl.lfh`。
   - 位于 `C:\ProgramData\ IBM \ MQ\Config` 中的配置文件。
   - IBM MQ 9.2 .ini 文件和注册表条目。 队列管理器信息存储在 .ini 文件中，可用于还原到产品的先前版本。
5. 停止 IBM MQ。

   a) 停止 IBM MQ 服务。

   右键单击系统托盘中的 **IBM MQ** 图标，然后单击 **停止 IBM MQ**。

   将显示包含以下消息的对话框:

   ```
   关闭 IBM MQ 将终止所有正在运行的队列管理器和
   IBM MQ 个进程。 确定要继续吗？ (AMQ4102)
   ```

   b) 单击 **是**，然后等待 IBM MQ 停止。

   c) 当 IBM MQ 已停止时，右键单击系统托盘中的 **IBM MQ** 图标，然后单击 **退出**。

结果

您已停止要迁移到 IBM MQ 的更高发行版的队列管理器，并已备份队列管理器数据。

下一步做什么

现在，您已准备好卸载 IBM MQ ，如 第 112 页的『卸载较低版本』中所述。

相关任务

备份队列管理器数据

Windows 卸载较低版本

使用控制面板卸载较低版本。 对于 Windows 上的单阶段迁移，在安装更高版本之前卸载产品的较低版本是可选的。

开始之前

在启动此任务之前，必须首先停止队列管理器，关闭 IBM MQ Explorer 并停止 IBM MQ ，如 第 111 页的『准备迁移』中所述。

关于此任务

在此任务中，使用 Windows 控制面板卸载 IBM MQ 。 在卸载过程中不会除去队列管理器数据，这意味着将保留此场景中使用的样本队列管理器，并且可以在安装产品的较新版本时进行检测。

是否必须先卸载较低版本的产品，然后再安装较高版本取决于您的操作系统。 在 Windows 系统上，卸载是可选的，您也可以在不卸载较早版本的情况下安装更新版本。 请注意，在这种情况下，您将在安装期间看到的某些选项和消息将与先卸载先前版本时显示的选项和消息不同。 有关在安装较新版本之前必须在其上卸载较低版本的平台的更多信息，请参阅 在 AIX and Linux 上迁移: 单阶段。

过程

1. 通过单击 **开始> 控制面板> 卸载程序**打开 Windows 控制面板。
2. 在 " **程序和功能部件** " 窗口中，找到要除去的安装的条目，例如 IBM WebSphere MQ (Installation1) ，然后单击 **卸载**。

   卸载过程将开始并运行至完成。 此过程完成后，将从您的计算机中除去较早版本的 IBM MQ ，并且不再显示在程序列表中。

结果

已从计算机中除去较早版本的 IBM MQ 。 但是，尚未除去队列管理器数据。

下一步做什么

现在，您已准备好安装更高版本的 IBM MQ ，如 第 113 页的『使用启动板安装 IBM MQ 9.3』中所述。

相关任务

在 Windows 系统上卸载 IBM MQ

Windows 使用启动板安装 *IBM MQ 9.3*

使用安装启动板和向导在安装了较低版本的 IBM MQ 的同一 Windows 计算机上安装较高版本的。

开始之前

在启动此任务之前，请下载包含安装映像的压缩文件，然后将其解压缩到临时目录中。

此任务假定您先前已卸载要从其进行迁移的较低版本的 IBM MQ ，如 第 112 页的『卸载较低版本』中所述。 如果在不首先卸载较低版本的情况下安装较高版本，那么您在安装过程中看到的某些选项和消息将与本任务中描述的选项和消息不同。

在启动此任务之前，请完成以下检查:

- 安装时必须具有本地管理员权限。 通过 Windows 工具定义此权限。
- 确保机器名不包含任何空格。
- 确保您有足够的磁盘空间。 有关更多信息，请参阅 Multiplatforms 版上的磁盘空间需求。

对于此场景，无需确定是否需要为任何 IBM MQ 用户定义 Windows 域用户标识，因为此需求不在此场景的范围内。 请参阅 为 IBM MQ 创建 Active Directory 和 DNS 域以获取更多信息。

在安装 IBM MQ 之前，请检查系统是否满足硬件和软件需求。 有关硬件和软件需求的最新详细信息，请参阅系统要求 IBM MQ。

关于此任务

此任务描述从较低版本迁移时在 Windows 操作系统上安装 IBM MQ 的基本步骤。

注: 对于 IBM MQ 9.0 和更高版本，缺省程序和数据目录位置相同。 因此，从 IBM MQ 9.0 迁移到更高版本时，不需要更改程序和数据目录的规范。 但是，从 IBM MQ 9.0 之前的版本迁移时，需要考虑的缺省位置存在差异。 有关更多信息，请参阅 Windows 上的程序和数据目录位置。

如果您在安装过程中需要安装程序，那么安装程序包含指向进一步信息的链接。 安装过程包含以下部分:

1. 使用启动板来检查和安装软件需求，指定网络信息，然后启动 IBM MQ 安装向导。
2. 使用 IBM MQ 安装向导来安装软件，并启动 Prepare IBM MQ Wizard。
3. 使用 Prepare IBM MQ Wizard 来启动 IBM MQ 服务。

过程

1. 启动安装过程。

   在 Windows Explorer 中，浏览至安装映像下载到的临时文件夹，然后双击 `setup.exe`。

   安装启动板已启动。
2. 启动启动板，查看并在必要时修改软件需求和网络配置。
   a) 浏览到 IBM MQ 软件目录，然后双击文件 `Setup.exe` 以启动启动板。
   b) 单击 **软件需求** 按钮以显示 **软件需求** 选项卡。
   c) 检查是否已满足软件需求，以及需求条目是否显示带有 OK 字样的绿色刻度标记。 进行任何指示的更正。

      注:

      有关任何需求的详细信息，请单击该复选框以展开 "信息" 选项卡。
   d) 单击 **网络配置** 按钮以显示 **网络配置** 选项卡。
   e) 单击 **否** 单选按钮。

      注: 此场景假定您不需要为 IBM MQ 配置域用户标识。 有关配置 IBM MQ for Windows 域用户的更多信息，请单击 **更多信息** 按钮。
   f) 在启动板的 **IBM MQ 安装** 选项卡上，选择安装语言，然后单击 **启动 IBM MQ 安装程序** 以启动 IBM MQ 安装向导。

   您已通过满足或指定安装需求来完成设置 IBM MQ ，并且已启动 IBM MQ 安装向导。
3. 使用 IBM MQ 安装向导来安装软件，并启动 Prepare IBM MQ Wizard。
   a) 在 IBM MQ 安装向导中，阅读 "许可协议" ，单击 **我接受许可协议中的条款** 复选框，然后单击 **下一步**。
   b) 单击 **典型**，然后单击 **下一步**。
   c) 在 " **准备安装 IBM MQ** " 页面上，查看安装信息，然后单击 **安装**。

      安装信息包含以下详细信息:

      - 安装名称
      - 程序文件的顶级文件夹
      - 数据文件的顶级文件夹

      安装了以下功能部件:

      - IBM MQ 服务器
      - IBM MQ: 用于管理和监视 IBM MQ 资源的图形界面
      - Java™ 和 .NET 消息传递和 Web Service
      - IBM MQ 开发工具箱

      安装过程开始。 根据您的系统，安装过程可能需要几分钟时间。

      在安装过程结束时， " IBM MQ 安装" 窗口将显示消息 `Installation Wizard Completed Successfully` 。
   d) 单击**完成**。

   您已成功安装 IBM MQ。 Prepare IBM MQ Wizard 会自动启动，显示 **欢迎使用 Prepare IBM MQ Wizard** 页面。
4. 使用 Prepare IBM MQ Wizard 来启动 IBM MQ 服务。
   a) 在 "欢迎使用 Prepare IBM MQ Wizard " 页面上，选择 **下一步**。

Prepare IBM MQ Wizard 显示消息 `Status: Checking IBM MQ Configuration` 和进度条。 完成此过程后，将显示 " IBM MQ 网络配置" 页面。

b) 在 Prepare IBM MQ Wizard 的 " IBM MQ 网络配置" 页面上，选择 **否**。

c) 单击**下一步**。

Prepare IBM MQ Wizard 显示消息 `Status: starting the IBM MQ Service` 和进度条。 完成此过程后，向导将显示 "完成 Prepare IBM MQ Wizard " 页面。

d) 选择 **启动 IBM MQ Explorer** 并选择是否启动记事本以查看发行说明，然后单击 **完成** 按钮。

IBM MQ Explorer 。

结果

您已安装 IBM MQ，并且已启动 IBM MQ Explorer。

下一步做什么

现在，您已安装了更高版本的 IBM MQ，您已准备好检查样本队列管理器是否已成功迁移，以及您是否能够将消息放入已迁移的队列并从这些队列中获取消息，如 第 115 页的『验证 IBM MQ 9.3 安装』中所述。

相关概念

从何处查找可下载的安装映像

相关任务

在 Windows 上安装 IBM MQ 服务器

验证 *IBM MQ 9.3* 安装

安装 IBM MQ 9.3 后，请使用 IBM MQ Explorer 来验证队列管理器和队列是否已成功从先前发行版迁移，然后验证您是否可以使用样本应用程序。

关于此任务

当您检查已迁移的队列管理器 sampleQM 是否在 IBM MQ Explorer 的 " Navigator " 视图中可视时，请验证您是否可以将消息放入已迁移的队列并从该队列获取消息，然后检查您是否仍可以运行样本应用程序。

过程

1. 如果 IBM MQ Explorer 未在运行，请立即将其启动。

   单击 **开始> 所有程序> IBM MQ > IBM MQ Explorer**。
2. 验证您的队列管理器是否已成功迁移到更高版本的 IBM MQ:

   a) 在“导航器”视图中，展开**队列管理器**文件夹。

   b) 检查您是否可以在 **队列管理器** 文件夹中看到队列管理器 sampleQM 。

   c) 展开队列管理器 sampleQM，单击 **队列** 文件夹，并检查您是否可以在 "内容" 视图中看到队列 Q1 。
3. 如果队列管理器 sampleQM 尚未启动，请立即将其启动。

   a) 在 " Navigator " 视图中，展开队列管理器节点。

   b) 右键单击队列管理器 sampleQM，然后单击 **启动**。
4. 验证是否可以将消息放入队列 Q1。

   a) 在“导航器”视图中，展开**队列管理器**文件夹。

   b) 展开队列管理器 sampleQM，然后单击 **队列** 文件夹。

   c) 在 "内容" 视图中，右键单击队列 Q1，然后单击 **放置测试消息**。

   此时会打开**“放入测试消息”**对话框。

   d) 在 **消息数据** 字段中，输入一些文本，例如 `Hello queue!`，然后单击 **放入消息**。

   系统会清空**消息数据**字段，并将消息放入队列。

   e) 单击**关闭**。

在 "内容" 视图中，请注意该队列的 **当前队列深度** 值现在为 1。 如果 "当前队列深度" 列不可见，那么您可能需要滚动到 "内容" 视图的右侧。

5. 验证是否可以从队列 Q1 获取消息。
   a) 在 " Navigator " 视图中，展开 **队列管理器** 文件夹。
   b) 展开队列管理器 sampleQM ，然后单击 **队列** 文件夹。
   c) 在 "内容" 视图中，右键单击队列 Q1，然后单击 **浏览消息**。
   "消息" 浏览器将打开，以显示当前位于队列中的消息的列表。
   d) 双击最后一条消息以打开 "属性" 对话框。
   在属性对话框的**数据**页上，**消息数据**字段以人类可读的格式显示了消息的内容。
6. 验证您是否可以运行样本应用程序。
   a) 双击 `runresponder.cmd` 文件。
   在标记为 **响应程序窗口**的命令提示符窗口中，响应程序客户机将启动，然后等待消息。

```
> 位于 JNDI 中的连接工厂。> 位于 JNDI 中的目标。> 正在创建与 QueueManager 的连接。> 已创建连接。
> 正在等待消息。
```

   b) 双击 `runrequester.cmd` 文件。
   在 " **请求者**" 窗口中，观察请求者消息。 在 " **响应程序**" 窗口中，观察更新的响应程序消息; 它从请求者客户机接收到的消息以及它发送的应答消息。

在标记为 **请求者窗口**的命令提示符窗口中，请求者客户机显示连接状态，发送的消息，然后显示从响应者客户机接收到的应答消息:

```
> 位于 JNDI 中的连接工厂。> 位于 JNDI 中的目标。> 正在创建与 QueueManager 的连接。> 已创建连接。
> 正在发送 "BakedBeans" 的库存请求> 已发送消息
ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02
> 接收到针对 "B" 的消息 ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902
akedBeans -库存中的 15 个容器
> 正在关闭与 QueueManager 的连接。> 已关闭连接。
------------------------------------------------------------
在此窗口中，观察通过 IBM MQ 发送的消息:
-发送的请求消息
-收到的应答消息
-----
准备就绪后，按任意键以关闭此窗口
按任意键以继续 . . .
```

在 " **响应程序**" 窗口中，观察已更新的响应程序消息; 它从请求者客户机接收到的消息以及它发送的应答消息:

```
> 位于 JNDI 中的连接工厂。> 位于 JNDI 中的目标。> 正在创建与 QueueManager 的连接。> 已创建连接。
> 正在等待消息。

> 接收到针对 "B" 的消息 ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02
akedBeans'
> 正在发送应答消息 "BakedBeans -15 Tins in stock"> 已发送消息
ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902
> 正在关闭与 QueueManager 的连接。> 已关闭连接。
------------------------------------------------------------
在此窗口中，观察已更新的响应者消息
-接收到的请求消息 （来自请求者）
-发送的应答消息
-----
准备就绪后，按任意键以关闭此窗口
按任意键以继续 . . .
```

两个命令窗口中显示的消息验证样本应用程序的请求者和响应程序客户机是否可以通过 IBM MQ 相互通信。

结果

您已成功迁移到更高版本的 IBM MQ。

Windows 选项 2: 并行迁移

此方案的选项 2 显示了在使用并行迁移方法时如何从 IBM MQ 的较早发行版迁移到更高发行版。 通过并行迁移，可以安装 IBM MQ 的最新版本以及要从中进行迁移的较早版本。 队列管理器和应用程序将继续与先前发行版相关联，直到您将它们迁移到更高发行版为止。

开始之前

此方案的起点是 第 99 页的『概述: 初始 IT 配置』中描述的初始 IT 配置。

在启动此任务之前，请遵循 第 103 页的『创建初始 IT 配置』 中的指示信息来设置初始 IT 配置。

关于此任务

遵循此场景中描述的并行迁移方法时，请将更高版本与较低版本一起安装在备用位置。 由于在启动更高版本中的任何队列管理器之前卸载较低版本，因此可以将更高版本的 IBM MQ 的安装指定为主安装。 有关主安装的更多信息，请参阅 主安装。

过程

1. 使用启动板 安装 IBM MQ 9.3 ， 然后验证安装。
2. 停止在较早版本的 IBM MQ 中运行的队列管理器。
3. 卸载较低版本的 IBM MQ。
4. 使 IBM MQ 9.3 成为主安装。
5. 可选： 将队列管理器与 IBM MQ 9.3 关联
6. 使用 IBM MQ Explorer 来 验证 IBM MQ 9.3 安装。
   检查队列管理器是否已从先前发行版成功迁移，以及您是否可以将消息放入已迁移的队列中并从这些队列中获取消息。

相关任务

在 Windows 上迁移: 并排

Windows 使用启动板安装 *IBM MQ 9.3*

使用安装启动板和向导在 Windows 上安装更高版本的 IBM MQ 以及更低版本。

开始之前

在完成此任务之前，请完成以下检查:

- 安装时必须具有本地管理员权限。 通过 Windows 工具定义此权限。
- 确保机器名不包含任何空格。
- 确保您有足够的磁盘空间 (最多 1005 MB) 来完全安装 IBM MQ for Windows。
- 确定是否需要为任何 IBM MQ 用户定义 Windows 域用户标识。

在安装 IBM MQ 之前，请检查系统是否满足硬件和软件需求。 有关所有受支持平台上的硬件和软件需求的最新详细信息，请参阅 系统要求 IBM MQ。

关于此任务

此任务描述了在 Windows 上安装 IBM MQ 的基本步骤 (如果尚未将其安装在系统上)。

此任务假定您将使用缺省 IBM MQ 程序和数据文件位置。

注: 对于 IBM MQ 9.0 和更高版本，缺省程序和数据目录位置相同。 因此，从 IBM MQ 9.0 迁移到更高版本时，不需要更改程序和数据目录的规范。 但是，从 IBM MQ 9.0 之前的版本迁移时，需要考虑的缺省位置存在差异。 有关更多信息，请参阅 Windows 上的程序和数据目录位置。

如果您在安装过程中需要安装程序，那么安装程序包含指向进一步信息的链接。 安装过程包含以下部分:

1. 使用启动板来检查和安装软件需求，指定网络信息，然后启动 IBM MQ 安装向导。
2. 使用 IBM MQ 安装向导来安装软件，并启动 Prepare IBM MQ Wizard。
3. 使用 Prepare IBM MQ Wizard 来启动 IBM MQ 服务。

过程

1. 启动安装过程。

   在 Windows Explorer 中，浏览至安装映像下载到的临时文件夹，然后双击 `setup.exe`。

   安装启动板已启动。
2. 使用启动板，复审并在必要时修改软件需求和网络配置。
   a) 单击 **软件需求** 按钮以显示 **软件需求** 选项卡。
   b) 检查是否已满足软件需求，以及需求条目是否显示带有 OK 字样的绿色刻度标记。 进行任何指示的更正。

      注: 要查看有关任何需求的更多详细信息，请单击加号 (+) 按钮。
   c) 单击 **网络配置** 按钮以显示 **网络配置** 选项卡。
   d) 单击 **否** 单选按钮。

      注: 此场景假定您不需要为 IBM MQ 配置域用户标识。 有关配置 IBM MQ for Windows 域用户的更多信息，请单击 **更多信息** 按钮。
   e) 在启动板的 **IBM MQ 安装** 选项卡上，选择安装语言，然后单击 **启动 IBM MQ 安装程序** 按钮以启动 IBM MQ 安装向导。

   您已完成安装需求的 IBM MQ 复审，进行了任何必需的修改，并已启动 IBM MQ 安装向导。
3. 使用 IBM MQ 安装向导来安装软件，并启动 Prepare IBM MQ Wizard。

   IBM MQ 安装向导将检查现有安装，并显示可供您使用的升级或安装选项。 在此场景中，有两个选项:

   - 安装会使现有的安装保持不变
   - 升级 8.0.0.5 安装 "安装 1"

   a) 选择 **安装使现有安装不受影响**，然后单击 **下一步**。
   b) 阅读 "许可协议"，单击 **我接受许可协议中的条款** 复选框，然后单击 **下一步**。
   c) 单击 **典型**，然后单击 **下一步**。
   d) 在 " **准备安装 IBM MQ** " 页面上，查看显示的安装信息，然后单击 **安装**。

      安装信息包含以下详细信息:

      - 安装名称
      - 程序文件的顶级文件夹
      - 数据文件的顶级文件夹

      安装了以下功能部件:

      - IBM MQ 服务器
      - IBM MQ: 用于管理和监视 IBM MQ 资源的图形界面
      - Java™ 和 .NET 消息传递和 Web Service
      - IBM MQ 开发工具箱

      安装过程开始。 根据您的系统，安装过程可能需要几分钟时间。

      在安装过程结束时， " IBM MQ 安装" 窗口将显示消息 `Installation Wizard Completed Successfully`。
   e) 单击**完成**。

   您已成功安装 IBM MQ。 Prepare IBM MQ Wizard 会自动启动，显示 **欢迎使用 Prepare IBM MQ Wizard** 页面。

4. 使用 Prepare IBM MQ Wizard 来启动 IBM MQ 服务。
   a) 在 "欢迎使用 Prepare IBM MQ Wizard " 页面上，选择 **下一步**。
      Prepare IBM MQ Wizard 显示消息 `Status: Checking IBM MQ Configuration` 和进度条。 完成此过程后，将显示 " IBM MQ 网络配置" 页面。
   b) 在 Prepare IBM MQ Wizard 的 " IBM MQ 网络配置" 页面上，选择 **否**。
   c) 单击**下一步**。
      Prepare IBM MQ Wizard 显示消息 `Status: starting the IBM MQ Service` 和进度条。 完成此过程后，向导将显示 "完成 Prepare IBM MQ Wizard " 页面。
   d) 选择 **启动 IBM MQ Explorer** 并选择是否启动记事本以查看发行说明，然后单击 **完成** 按钮。
      IBM MQ Explorer 。

下一步做什么

您已与较低版本一起安装了更高版本的 IBM MQ ，但已在另一个安装目录中启动了 IBM MQ Explorer。

现在，您已准备好停止在较低版本的 IBM MQ 中运行的队列管理器，如 第 119 页的『停止队列管理器』中所述。

Windows 停止队列管理器

在迁移到更高版本的 IBM MQ 之前，必须首先停止队列管理器并备份队列管理器数据。

关于此任务

在执行备份之前，请停止要备份的队列管理器。 如果尝试对正在运行的队列管理器进行备份，那么备份可能不一致，因为复制文件时正在进行任何更新。

过程

1. 打开 IBM MQ Explorer。
   单击 **启动** > **所有应用程序** > **IBM MQ** > **IBM MQ Explorer**。
2. 停止队列管理器 sampleQM。
   a) 在 " Navigator " 视图中，右键单击队列管理器 sampleQM。
   b) 单击**停止**。
      此时将打开 " **结束队列管理器** " 窗口。
   c) 选择 **受控**，然后单击 **确定**。
      选择 **受控** 选项将以受控的有序方式停止队列管理器。 强制队列管理器停止的 **立即** 选项通常仅在受控停止未能成功完成时使用。

   队列管理器停止。 在 IBM MQ 中，队列管理器 sampleQM 旁边的图标已更改为包含指向下方的红色箭头。
3. 关闭 IBM MQ Explorer。
4. 备份队列管理器数据。
   复制以下所有数据，确保包含所有备份目录。 某些目录可能为空，但如果需要在以后复原备份，那么您需要所有这些目录，因此也请保存这些目录。
   - 位于 `C:\ProgramData\ IBM \ MQ\Qmgrs` 中的队列管理器数据。
   - 位于 `C:\ProgramData\ IBM \ MQ\log` 中的队列管理器的日志文件目录，包括日志控制文件 `amqhlctl.lfh`。
   - 位于 `C:\ProgramData\ IBM \ MQ\Config` 中的配置文件。
   - IBM MQ 9.2 .ini 文件和注册表条目。 队列管理器信息存储在 .ini 文件中，可用于还原到产品的先前版本。
5. 停止 IBM MQ。
   a) 停止 IBM MQ 服务。

右键单击系统托盘中的 **IBM MQ** 图标，然后单击 **停止 IBM MQ**。

将显示包含以下消息的对话框:

```
关闭 IBM MQ 将终止所有正在运行的队列管理器和
IBM MQ 个进程。 确定要继续吗？ (AMQ4102)
```

b) 单击 **是**，然后等待 IBM MQ 停止。

c) 当 IBM MQ 已停止时，右键单击系统托盘中的 **IBM MQ** 图标，然后单击 **退出**。

下一步做什么

停止队列管理器后，您已准备好使它们与 IBM MQ 的更高版本的新安装相关联，如 第 121 页的『使队列管理器与 IBM MQ 9.3 相关联』中所述。

Windows 卸载较低版本

使用 Windows 控制面板卸载产品的较早版本。

开始之前

在启动此任务之前，必须首先停止队列管理器，关闭 IBM MQ Explorer 并停止 IBM MQ，如 第 119 页的『停止队列管理器』中所述。

关于此任务

在此任务中，使用 Windows 控制面板卸载 IBM MQ。 在卸载过程中不会除去队列管理器数据，这意味着将保留此场景中使用的样本队列管理器，并且可以在安装产品的较新版本时进行检测。

过程

1. 通过单击 **开始> 控制面板> 卸载程序**打开 Windows 控制面板。
2. 在 " **程序和功能部件** " 窗口中，找到要除去的安装的条目，例如 IBM WebSphere MQ (Installation1)，然后单击 **卸载**。

   卸载过程将开始并运行至完成。 此过程完成后，将从您的计算机中除去较早版本的 IBM MQ，并且不再显示在程序列表中。

结果

该产品的较早版本已从您的计算机中除去。 但是，尚未除去队列管理器数据。

下一步做什么

现在，您已准备好使 IBM MQ 的更高版本成为主安装，如 第 120 页的『使 IBM MQ 9.3 成为主安装』中所述。

相关任务

在 Windows 系统上卸载 IBM MQ

Windows 使 *IBM MQ 9.3* 成为主安装

在 IBM MQ 的更高版本的新安装中启动任何队列管理器之前，您可以选择将更高版本作为主安装。

关于此任务

在支持 IBM MQ 的多个安装的系统上，主安装是 IBM MQ 系统范围位置所引用的安装。 具有主安装是可选的，但很方便。

遵循此场景中描述的并行迁移方法时，由于在启动更高发行版中的任何队列管理器之前卸载了较低版本，因此可以将产品的较高版本的安装指定为主安装。

有关主安装的更多信息，请参阅 主安装。

过程

1. 通过在命令提示符中输入 **dspmqinst** 命令来检查当前主安装。
   命令提示符显示任何当前安装的详细信息。 当前主安装具有以下行 `Primary: Yes` 。
2. 使用 **setmqinst** 命令可更改当前主安装。
   在命令提示符中，输入：

   ```
   setmqinst -x -n Installation_Name
   ```

   其中 *Installation_Name>* 是当前主安装的名称。
   如果命令成功，那么命令提示符将显示消息 '*Installation_Name*' (*Filepath*) has been unset as the Primary Installation。
3. 使用 **setmqinst** 命令将新的 IBM MQ 9.3 安装设置为主安装。
   在命令提示符中，输入：

   ```
   setmqinst -i -n V9_Installation
   ```

   其中， *V9_Installation* 是 IBM MQ 9.3 安装的名称。

   如果命令成功，那么命令提示符将显示消息 '*V9_Installation*' (*Filepath*) has been set as the primary installation. You must restart the operating system to complete the update.

   注: 按照成功消息中的指示，必须重新启动操作系统才能完成更新。

下一步做什么

您已准备好使已迁移的队列管理器与更高版本的 IBM MQ 相关联，如 第 121 页的『使队列管理器与 IBM MQ 9.3 相关联』中所述。

Windows 使队列管理器与 *IBM MQ 9.3* 相关联

使用 "传输队列管理器" 向导将 sampleQM 队列管理器与更高版本的 IBM MQ 的安装相关联。

开始之前

在启动此任务之前，请确保已按 第 119 页的『停止队列管理器』中所述停止队列管理器，否则将无法完成传输。

关于此任务

IBM MQ Explorer 的 "传输队列管理器" 向导功能使您能够将一个或多个队列管理器从其他安装传输到当前安装。 此向导等同于 **setmqm** 命令，但无需输入所需的路径和参数。 只能传输已停止的队列管理器；但会显示正在运行的队列管理器，以供参考。

在产品的更高版本的安装上传输并启动队列管理器后，无法将其迁移回较低版本。

过程

1. 启动 IBM MQ Explorer。
   单击 **启动** > **所有应用程序** > **IBM MQ** > **IBM MQ Explorer**。
2. 在 " Navigator " 视图中，右键单击队列管理器节点，然后选择 **传输队列管理器**。
3. 右键单击，然后选择队列管理器 sampleQM，然后单击 **传输**。
   将对所选队列管理器调用 **setmqm** 命令。 如果传输成功，那么导航器树将更新以包含已传输的队列管理器。 如果出现任何问题，那么将显示一个对话框，其中显示来自该命令的错误消息。
4. 启动队列管理器 sampleQM。
   a) 在 " Navigator " 视图中，展开队列管理器节点。
   b) 右键单击队列管理器的名称，然后单击 **启动**。

结果

您已成功使队列管理器 sampleQM 与更高版本的 IBM MQ 相关联。

下一步做什么

通过确认您可以将消息放入队列并从队列中获取消息，验证是否已成功迁移队列管理器 sampleQM ，如 第 122 页的『验证 IBM MQ 9.3 安装』中所述。

验证 *IBM MQ 9.3* 安装

安装更高版本的 IBM MQ 后，请使用 IBM MQ Explorer 来验证队列管理器和队列是否已成功从较低发行版迁移，然后验证您是否可以使用样本应用程序。

关于此任务

当您检查已迁移的队列管理器 sampleQM 是否在 IBM MQ Explorer 的 " Navigator " 视图中可视时，请验证您是否可以将消息放入已迁移的队列并从该队列获取消息，然后检查您是否仍可以运行样本应用程序。

过程

1. 如果 IBM MQ Explorer 未在运行，请立即将其启动。
   单击 **开始> 所有程序> IBM MQ > IBM MQ Explorer**。
2. 验证您的队列管理器是否已成功迁移到更高版本的 IBM MQ:
   a) 在“导航器”视图中，展开**队列管理器**文件夹。
   b) 检查您是否可以在 **队列管理器** 文件夹中看到队列管理器 sampleQM 。
   c) 展开队列管理器 sampleQM，单击 **队列** 文件夹，并检查您是否可以在 "内容" 视图中看到队列 Q1 。
3. 如果队列管理器 sampleQM 尚未启动，请立即将其启动。
   a) 在 " Navigator " 视图中，展开队列管理器节点。
   b) 右键单击队列管理器 sampleQM，然后单击 **启动**。
4. 验证是否可以将消息放入队列 Q1。
   a) 在“导航器”视图中，展开**队列管理器**文件夹。
   b) 展开队列管理器 sampleQM，然后单击 **队列** 文件夹。
   c) 在 "内容" 视图中，右键单击队列 Q1，然后单击 **放置测试消息**。
   此时会打开“**放入测试消息**”对话框。
   d) 在 **消息数据** 字段中，输入一些文本，例如 `Hello queue!`，然后单击 **放入消息**。
   系统会清空**消息数据**字段，并将消息放入队列。
   e) 单击**关闭**。
   在 "内容" 视图中，请注意该队列的 **当前队列深度** 值现在为 1。 如果 "当前队列深度" 列不可见，那么您可能需要滚动到 "内容" 视图的右侧。
5. 验证是否可以从队列 Q1 获取消息。
   a) 在 " Navigator " 视图中，展开 **队列管理器** 文件夹。
   b) 展开队列管理器 sampleQM ，然后单击 **队列** 文件夹。
   c) 在 "内容" 视图中，右键单击队列 Q1，然后单击 **浏览消息**。
   "消息" 浏览器将打开，以显示当前位于队列中的消息的列表。
   d) 双击最后一条消息以打开 "属性" 对话框。
   在属性对话框的**数据**页上，**消息数据**字段以人类可读的格式显示了消息的内容。
6. 验证您是否可以运行样本应用程序。
   a) 双击 `runresponder.cmd` 文件。
   在标记为 **响应程序窗口**的命令提示符窗口中，响应程序客户机将启动，然后等待消息。

```
> 位于 JNDI 中的连接工厂。> 位于 JNDI 中的目标。> 正在创建与 QueueManager 的连接。> 已创建连接。
> 正在等待消息。
```

b) 双击 `runrequester.cmd` 文件。

在 "**请求者**" 窗口中，观察请求者消息。 在 "**响应程序**" 窗口中，观察更新的响应程序消息; 它从请求者客户机接收到的消息以及它发送的应答消息。

在标记为 **请求者窗口**的命令提示符窗口中，请求者客户机显示连接状态，发送的消息，然后显示从响应者客户机接收到的应答消息:

```
> 位于 JNDI 中的连接工厂。> 位于 JNDI 中的目标。> 正在创建与 QueueManager 的连接。> 已创建连接。
> 正在发送 "BakedBeans" 的库存请求> 已发送消息
ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02
> 接收到针对 "B" 的消息 ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902
akedBeans -库存中的 15 个容器
> 正在关闭与 QueueManager 的连接。> 已关闭连接。
------------------------------------------------------------
在此窗口中，观察通过 IBM MQ 发送的消息:
-发送的请求消息
-收到的应答消息
-----
准备就绪后，按任意键以关闭此窗口
按任意键以继续 . . .
```

在 "**响应程序**" 窗口中，观察已更新的响应程序消息; 它从请求者客户机接收到的消息以及它发送的应答消息:

```
> 位于 JNDI 中的连接工厂。> 位于 JNDI 中的目标。> 正在创建与 QueueManager 的连接。> 已创建连接。
> 正在等待消息。

> 接收到针对 "B" 的消息 ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02
akedBeans'
> 正在发送应答消息 "BakedBeans -15 Tins in stock"> 已发送消息
ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902
> 正在关闭与 QueueManager 的连接。> 已关闭连接。
------------------------------------------------------------
在此窗口中，观察已更新的响应者消息
-接收到的请求消息 （来自请求者）
-发送的应答消息
-----
准备就绪后，按任意键以关闭此窗口
按任意键以继续 . . .
```

两个命令窗口中显示的消息验证样本应用程序的请求者和响应程序客户机是否可以通过 IBM MQ 相互通信。

结果

您已成功迁移到更高版本的 IBM MQ。

下一步做什么

有关迁移和升级的更多信息，请参阅 维护和迁移。

Windows LTS 安装更高版本的 IBM MQ 以与 Windows 上的较低版本共存

此方案显示安装 Long Term Support (LTS) 版本的 IBM MQ 的所有步骤，同时共存 ("并排") 使用较早版本的产品。 此外，这些步骤还包括将修订包安装到更高版本。 您应该始终安装最新的修订包级别。

多个安装的概述

此方案中使用的多个安装以及硬件和软件的描述。

关于此任务

在同一主机中多次安装 IBM MQ (多安装) 的此功能的一个重要方面是，此类型的安装不需要在使用产品的更高版本执行活动时停止产品的较低版本上的队列管理器和应用程序。

即，产品较新版本的安装不会影响产品较旧版本上应用程序的运行。 当您尝试将产品的较低版本上的队列管理器多阶段迁移到较高版本时，这很有用。

对于此方案， IBM MQ 8.0.0 用作产品的较低版本， IBM MQ 9.1.0 用作产品的较高版本。

使用 Long Term Support 版本的 IBM MQ

此方案将 LTS 版本的 IBM MQ 用于更高版本的产品。

CD 如果要将 Continuous Delivery (CD) 版本的 IBM MQ 用于更高版本的产品，那么必须先卸载您正在使用的 CD 版本 (例如 IBM MQ 9.1.1) ，然后再安装更高版本 (例如 IBM MQ 9.1.5)。 有关更多信息，请参阅 从一个 Continuous Delivery 发行版迁移到另一个发行版。

用于此场景的硬件和软件

操作系统

Windows 10
主机名： `johndoe1.fyre.<yourdomainname>.com`

队列管理器

QM80
使用 IBM MQ 8.0.0 创建; 以保留在 IBM MQ 8.0.0

QM 联格观察团
使用 IBM MQ 8.0.0 创建; 要迁移到 IBM MQ 9.1.0

QM910
使用 IBM MQ 9.1.0 创建; 以保留在 IBM MQ 9.1.0

将更高版本的 IBM MQ 并排安装到更低版本。

如何将 IBM MQ 9.3 并排安装到同一机器上的现有产品版本。 IBM MQ 9.1 安装将不会指定为主安装。

开始之前

确保系统上安装了 IBM MQ 9.1 。 如果未安装 IBM MQ 9.1 ，请使用以下指示信息 在 Windows 上安装 IBM MQ 服务器来安装产品。

您需要选择 定制 选项，以便可以显式选择客户机。 有关更多信息，请参阅 Windows 的安装方法。

您还需要调用带有 **-n** 参数的 **setmqenv** 命令，以将安装名称设置为 `Installation1`。 如果您使用批处理文件来调用该命令，那么这很有用。

关于此任务

所有队列管理器数据都存储在公共目录结构中。 尽管每个版本的 IBM MQ 的可执行代码都存储在不同的目录结构中，但所有队列管理器 (无论其版本如何) 的数据都存储
在 :MQ_DATA_PATH=`C:\ProgramData\IBM\MQ` 中。

要安装 IBM MQ 9.3，请执行以下操作:

过程

1. 以管理员身份登录。
   IBM MQ 9.3 将安装在缺省目录 `C:\Program Files\IBM\MQ` 中。
2. 转至下载文件所在的目录，例如 `C:\downloads\mq9300`。
3. 解压缩已下载的文件。
   这些文件将解压缩到名为 `MQServer` 的新子目录中。
4. 切换到新目录并发出命令 **setup.exe** 以启动安装程序。
   a) 单击 软件需求 以检查您的企业是否安装了必需软件。

请参阅 检查 Windows 上的需求 以获取更多信息。 在此场景中，系统具有必需的需求。

b) 单击 网络配置。

在此场景中，机器不是域的一部分，因此无需指示域用户，问题的答案为 No。

c) 单击 *IBM MQ* 安装。

d) 单击 启动 *IBM MQ* 安装程序。

安装程序检测到系统上存在其他安装，并显示以下消息:

升级或安装
升级现有安装或将新版本与其一起安装

对于此场景，请保持其他安装不变，因此选择第一个条目 安装使现有安装保持不变。

e) 单击 **下一步** 并接受许可证。

f) 针对安装选项选择 定制 。

有关更多信息，请参阅 Windows 的安装方法。

g) 单击**下一步**。

在

安装详细信息
定义安装详细信息

您会看到:

安装名称
Installation2

程序文件的安装文件夹
`C:\ProgramFiles\IBM\MQ`

将显示要安装的功能部件的列表。 请注意，某些项:

- 缺省情况下未安装
- 如果要安装这些产品，那么需要选择这些产品。

对于此场景，请选择 *MQI* 客户机，并且无论如何都应选择此选项。

h) 单击**下一步**。

在标题为 准备好安装 *IBM MQ* 的屏幕中，您将看到 "安装名称" ， "位置" 和 "要安装的组件" 等的摘要。

i) 单击 **安装** 以继续。

开始将文件复制到安装目录结构中。 复制文件后，将显示以下对话框: `The IBM MQ Installation Wizard has successfully installed IBM MQ`。

j) 单击**完成**。

安装后，您将看到以下对话框: `Welcome to the Prepare IBM MQ Wizard`。

由于此方案未使用 Windows 域，因此请接受缺省值 **No**。

接受以启动 IBM MQ Explorer，除非您有非常具体的原因，否则可以接受 使用新工作空间启动 *MQ Explorer* 的缺省设置。

结果

您已成功安装另一个版本的 IBM MQ for Windows 以及现有版本的产品。

下一步做什么

您需要运行 **setmqenv** 命令以在任一版本上使用这些命令。 有关详细信息，请参阅第 126 页的『使用 setmqenv 命令与两个版本的 IBM MQ 一起运行』。

使用 setmqenv 命令与两个版本的 IBM MQ 一起运行

安装活动已完成，您现在可以验证包含 IBM MQ 代码的目录。

开始之前

确保已正确安装 C:\Program Files\IBM\WebSphere MQ\ 中的 IBM MQ 8.0.0 产品 Installation1 和 C:\Program Files\IBM\MQ 中的 IBM MQ 9.1 产品 Installation2。

关于此任务

使用 **dspmqinst** 命令可显示系统中安装的版本的安装信息，而无需查看目录，使用 **dspmqver** 命令可显示版本信息。

在 Windows 系统上，信息保留在注册表中，安装配置信息保留在以下键中: Computer\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation。

要点: 不得直接编辑或引用此密钥。

过程

1. 使用 **dspmqinst** 命令显示安装信息。

```
InstName:      Installation1
InstDesc:
Identifier:    1
InstPath:      C:\Program Files\IBM\WebSphere MQ
Version:       8.0.0.9
Primary:       Yes
State:         Available
MSIProdCode:   {74F6B169-7CE6-4EFB-8A03-2AA7B2DBB57C}
MSIMedia:      8.0 Server
MSIInstanceId: 1

InstName:      Installation2
InstDesc:
Identifier:    2
InstPath:      C:\Program Files\IBM\MQ
Version:       9.1.0.0
Primary:       No
State:         Available
MSIProdCode:   {5D3ECA81-BF8D-4E80-B36C-CBB1D69BC110}
MSIMedia:      9.1 Server
MSIInstanceId: 1
```

   注: 相应的安装名称 (InstName) 很重要。

2. 显示使用 **dspmqver** 进行缺省 (或主) 安装的 IBM MQ 版本:

```
C:\> dspmqver
Name:        WebSphere MQ
Version:     8.0.0.9
Level:       p800-009-180321.1
BuildType:   IKAP - (Production)
Platform:    WebSphere MQ for Windows (x64 platform)
Mode:        64-bit
O/S:         Windows 10 Professional x64 Edition, Build 18363
InstName:    Installation1
InstDesc:
Primary:     Yes
InstPath:    C:\Program Files\IBM\WebSphere MQ
DataPath:    C:\ProgramData\IBM\MQ
MaxCmdLevel: 802
LicenseType: Production
```

   运行该命令后，您将收到以下消息: Note there are a number (1) of other installations, use the '-i' parameter to display them。

3. 发出命令 C:\> where dspmqver，您将看到有关主安装的信息:

```
C:\> where dspmqver
C:\Program Files\IBM\WebSphere MQ\bin64\dspmqver.exe
C:\Program Files\IBM\WebSphere MQ\bin\dspmqver.exe
```

4. 要查看有关 IBM MQ 9.1 产品的信息，请发出以下命令: `C:\> setmqenv -n Installation2`。
5. 再次发出命令 `C:\> where dspmqver`，您将看到有关第二个安装的信息:

```
C:\> where dspmqver
C:\Program Files\IBM\MQ\bin64\dspmqver.exe
C:\Program Files\IBM\MQ\bin\dspmqver.exe
```

6. 再次发出命令 `C:\dspmqver`。
   现在，您将看到:

```
C:\> dspmqver
Name:        IBM MQ
Version:     9.1.0.0
Level:       p910-L180705
BuildType:   IKAP - (Production)
Platform:    IBM MQ for Windows (x64 platform)
Mode:        64-bit
O/S:         Windows 10 Professional x64 Edition, Build 18363
InstName:    Installation2
InstDesc:
Primary:     No
InstPath:    C:\Program Files\IBM\MQ
DataPath:    C:\ProgramData\IBM\MQ
MaxCmdLevel: 910
LicenseType: Production
```

7. 发出命令 `C:\ set MQ`，使用 **setmqenv** 后，您将看到有关第二个安装的信息。

```
C:\> set MQ
MQ_DATA_PATH=C:\ProgramData\IBM\MQ
MQ_ENV_MODE=64
MQ_FILE_PATH=C:\Program Files\IBM\MQ
MQ_INSTALLATION_NAME=Installation2
MQ_INSTALLATION_PATH=C:\Program Files\IBM\MQ
MQ_JAVA_DATA_PATH=C:\ProgramData\IBM\MQ
MQ_JAVA_INSTALL_PATH=C:\Program Files\IBM\MQ\java
MQ_JAVA_LIB_PATH=C:\Program Files\IBM\MQ\java\lib64
MQ_JRE_PATH=C:\Program Files\IBM\MQ\java\jre
```

   您可以创建将使用指定语法运行 **setmqenv** 命令的批处理文件。 确保在 PATH 中的目录中具有此批处理文件，例如 `C:\WinTools`。
   例如，您可以创建具有以下内容的批处理文件 `set-mq-910.bat`:

```
REM Setup the environment to run MQ 9.1
CALL "C:\Program Files\IBM\MQ\bin\setmqenv" -n Installation2
REM Adding Samples to the path
SET PATH=%PATH%;%MQ_FILE_PATH%\Tools\c\Samples\Bin;%MQ_FILE_PATH%\Tools\c\Samples\Bin64
;%MQ_FILE_PATH%\Tools\jms\samples;%MQ_JAVA_INSTALL_PATH%\bin\ dspmqver -f 2
```

   注意:

   a. 调用 **setmqenv** 时需要使用 "CALL" 参数。 如果没有此参数，那么 **setmqenv** 的处理将结束批处理，并且不允许运行以下语句。 即，通过 CALL 参数，允许处理批处理文件中的其他语句。
   b. 如果将 IBM MQ 目录添加到 PATH 中，例如 C-samples 的位置 :PATH= ...;`C:\Program Files\IBM\MQ\tools\c\Samples\Bin`; ... 此目录将在下次命令运行时由 **setmqenv** 除去。

      如果您希望能够从 IBM MQ 9.1 运行 C-samples，那么需要上一个批处理文件中的最后一行，以便将这些样本的目录放回 PATH 中。

      另请注意，使用 MQ_FILE_PATH 是为了将正确的目录结构用于 IBM MQ 9.1: SET `PATH=%PATH%;%MQ_FILE_PATH%\tools\c\Samples\Bin`。

创建队列管理器

如何使用 **crtmqm** 命令创建队列管理器。 您可以使用 IBM MQ Explorer 来执行此任务。

开始之前

将 **dspmq** 命令与 **-o** 安装 和 **-s** 参数配合使用，以显示当前队列管理器的安装名称和状态。

```
C:\> dspmq -o installation -s
QMNAME(QM80)                                  STATUS(Running) INSTNAME(Installation1)
  INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)
QMNAME(QMMIG)                                 STATUS(Running) INSTNAME(Installation1)
  INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)
```

关于此任务

您必须打开 Windows 命令提示符，并将自己设置为 管理员 以执行以下过程。 如果您尝试在不使用命令提示符的情况下发出 **crtmqm** 命令，那么会收到消息 AMQ7077: **您无权执行请求的操作**。

过程

1. 选择 **启动 > Windows 系统 > 命令提示符 > 更多 > 以管理员身份运行**
   创建的窗口的标题为 管理员*:* 命令提示符。

   注: 命令提示符中 IBM MQ 的版本为 IBM MQ 9.1。
2. 运行 **setmqenv** 命令或可能已创建的批处理文件 **set-mq-930**。
   请参阅 第 126 页的『使用 setmqenv 命令与两个版本的 IBM MQ 一起运行』 以了解详细信息。
   在任一情况下，您都将看到 V 9.3.0.0
3. 发出以下命令: `C:\> crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM930`
   您将看到以下输出:

   ```
   IBM MQ queue manager created.
   Directory 'C:\ProgramData\IBM\MQ\qmgrs\QM930' created.
   The queue manager is associated with installation 'Installation2'.
   Creating or replacing default objects for queue manager 'QM930'.
   Default objects statistics : 87 created. 0 replaced. 0 failed.
   Completing setup.
   Setup completed.
   ```
4. 发出以下命令以启动队列管理器 `C:\> strmqm QM930`
   您将看到以下输出。 请注意指示安装和运行队列管理器的版本的行:

   ```
   IBM MQ queue manager 'QM930' starting.
   The queue manager is associated with installation 'Installation2'.
   5 log records accessed on queue manager 'QM930' during the log replay phase.
   Log replay for queue manager 'QM930' complete.
   Transaction manager state recovered for queue manager 'QM910'.
   IBM MQ queue manager 'QM930' started using V9.3.0.0.
   ```
5. 再次发出命令 `C:\> dspmq -o installation -s` 以显示已安装的队列管理器:

   ```
   C:\> dspmq -o installation -s
   QMNAME(QM80)                                  STATUS(Running) INSTNAME(Installation1)
     INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)
   QMNAME(QMMIG)                                 STATUS(Running) INSTNAME(Installation1)
     INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)
   QMNAME(QM910)                                 STATUS(Running) INSTNAME(Installation2)
     INSTPATH(C:\Program Files\IBM\MQ) INSTVER(9.3.0.0)
   ++ Cannot use MQ 9.3.0 administrative commands to run an MQ 9.1 queue manager
   ```

   要点: 不能在与您所使用的 IBM MQ 版本不同的版本上使用管理命令。

   如果尝试执行此操作，那么将收到消息 AMQ5691E: `Queue manager <qmname> is associated with a different installation (<installation name>)`。

将队列管理器迁移到更高版本的 IBM MQ

安装 IBM MQ 9.3 后，现在要迁移和升级要与 IBM MQ 9.3 配合使用的 IBM MQ 9.1 队列管理器。

关于此任务

您需要执行两个主要步骤:

1. 使用 **setmqm** 命令将队列管理器与期望的安装 (即 IBM MQ 版本) 相关联。
2. 在所需版本下使用 **strmqm** 命令，这将更新新版本的队列管理器数据。

对于此场景，使用 IBM MQ 9.1 创建的队列管理器 QMMIG 将用作迁移过程的说明。

注意: 将队列管理器迁移到更高版本的 IBM MQ 后，无法再将其与先前版本的 IBM MQ 配合使用。 迁移过程会更改许多文件和对象定义，无法返回。

过程

1. 使用 **dmpmqcfg** 命令备份队列管理器。
   请参阅 备份和复原 IBM MQ 队列管理器数据 和 备份队列管理器配置以获取更多信息。
   a) 要指定所有属性 (包括缺省属性) (不包括在输出中的 **setmqaut** 除外)，请指定以下命令:

   ```
   dmpmqcfg -m QMgr -a > QMgr.dmpmqcfg.out.all.mqsc
   ```

   b) 要捕获 **setmqaut** 格式的属性，请发出以下命令:

   ```
    dmpmqcfg -m QMgr -o setmqaut > QMgr.dmpmqcfg.setmqaut.bat
   ```

   注: 包含 **setmqaut** 命令的输出文件在每个命令中包含队列管理器的名称。 因此，如果要将命令复原到另一个队列管理器中，那么需要编辑该文件并指定所需的队列管理器名称。
2. 要复原:
   a) **runmqsc** 的命令，发出:

   ```
   runmqsc Qmgr < QMgr.dmpmqcfg.out.mqsc
   ```

   或

   ```
   runmqsc Qmgr < QMgr.dmpmqcfg.out.all.mqsc
   ```

   b) **setmqaut** 命令问题:

   ```
   QMgr.dmpmqcfg.setmqaut.bat
   ```
3. 要迁移的队列管理器位于 IBM MQ 9.1，因此您需要运行将运行环境设置为 IBM MQ 9.3 的脚本:

```
C:\> set-mq-80
```

a) 发出命令 C:\> dspmqver 以检查队列管理器在 IBM MQ 9.1 上运行的版本。
b) 发出命令 C:\> where dspmq 以检查队列管理器是否正在运行:

```
C:\Program Files\IBM\WebSphere MQ\bin64\dspmq.exe
C:\Program Files\IBM\WebSphere MQ\bin\dspmq.exe
```

c) 发出命令 C:\> dspmq -m QMMIG -o installation -s

```
QMNAME(QMMIG)               STATUS(Running) INSTNAME(Installation1)
INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)
```

d) 发出命令 C:\> runmqsc QMMIG

```
display qmgr cmdlevel version
       1 : display qmgr cmdlevel
AMQ8408: Display Queue Manager details.
  QMNAME(QMMIG)                           CMDLEVEL(910)
  VERSION(09001009)
end
```

并注意 CMDLEVEL 位于 IBM MQ 9.1。

e) 通过发出命令 C:\ endmqm -i QMMIG 来停止队列管理器
您将收到以下消息:

```
WebSphere MQ queue manager 'QMMIG' ending.
WebSphere MQ queue manager 'QMMIG' ended.
```

4. 通过发出命令 C:\> set-mq-930 (如果已创建批处理文件) 或使用 **setmqenv** 命令来更改环境以运行 IBM MQ 9.3 命令，然后通过发出 **dspmqver** 命令来检查版本。
5. 通过发出命令 C:\> dspmq -o installation -s 来显示队列管理器的状态。
您将收到以下输出:

```
QMNAME(QM91)                   STATUS(Running)INSTNAME(Installation1)
INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)

QMNAME(QMMIG)                              STATUS(Ended immediately) INSTNAME(Installation1)
INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)

QMNAME(QM910)                              STATUS(Running) INSTNAME(Installation2)
INSTPATH(C:\Program Files\IBM\MQ) INSTVER(9.3.0.0)
```

注意: 队列管理器 QM 联格仍与 Installation1 (IBM MQ 9.1) 关联，而 Installation2 与 IBM MQ 9.3 关联。

因此，您需要解除队列管理器 QMMIG 与 Installation1 的关联，并将其与 Installation2 关联

6. 发出以下命令以将队列管理器 QMMIG 与 Installation2 相关联

```
C:\> setmqm -m QMMIG -n Installation2
```

您会收到以下消息:

```
The setmqm command completed successfully.
```

这将通知您 QMMIG 现在与 IBM MQ 9.3 关联。
7. 通过发出命令 C:\> strmqm QMMIG 来启动队列管理器 QMMIG
由于这是第一次在先前与旧版本一起使用的队列管理器上发出 IBM MQ 9.3 **strmqm** 命令，因此会进行迁移。
您将看到类似于以下内容的输出:

```
IBM MQ queue manager 'QMMIG' starting.
The queue manager is associated with installation 'Installation2'.
5 log records accessed on queue manager 'QMMIG' during the log replay phase.
Log replay for queue manager 'QMMIG' complete.
Transaction manager state recovered for queue manager 'QMMIG'.
Migrating objects for queue manager 'QMMIG'.
Default objects statistics : 5 created. 0 replaced. 0 failed.
IBM MQ queue manager 'QMMIG' started using V9.3.0.0.
```

8. 发出以下命令 C:\> runmqsc QMMIG 以显示队列管理器的属性，并记下 VERSION 和 CMDLEVEL 字段:

```
display qmgr cmdlevel version
    1 : display qmgr cmdlevel version
AMQ8408I: Display Queue Manager details.
   QMNAME(QMMIG)                           CMDLEVEL(930)
   VERSION(09300000)
end
```

结果

您已成功将队列管理器迁移到更高版本的产品。

在 IBM MQ 9.3 上安装修订包

如何在具有 IBM MQ 的多版本安装的系统上的已安装 IBM MQ 9.3 上安装修订包。

开始之前

确保已将队列管理器 QMMIG 迁移到 IBM MQ 9.3。 有关更多信息，请参阅 第 128 页的『将队列管理器迁移到更高版本的 IBM MQ』。

关于此任务

对于此场景，存在另一个已安装版本 (IBM MQ 9.1)，并且将不会停止在该其他版本下运行的队列管理器，以证明您可以继续使用这些其他版本，同时对 IBM MQ 9.3.0.n 执行维护活动。

请注意，安装更新时未选择任何高级选项。

过程

1. 以管理员身份登录。
   a) 如果已创建批处理文件，请发出命令 `C:\> set-mq-930`，或者使用 **setmqenv** 命令来确保您在 IBM MQ 9.3.0.n 上，其中 n 在此场景中为 `0`。
   b) 通过发出命令 `C:\> dspmq -o installation -s` 来显示队列管理器的状态。
   您将收到以下输出:

```
QMNAME(QM80)                    STATUS(Running)INSTNAME(Installation1)
INSTPATH(C:\Program Files\IBM\MQ) INSTVER(9.1.0.0)

QMNAME(QMMIG)                             STATUS(Ended unexpectedly)
INSTNAME(Installation2)
INSTPATH(C:\Program Files\IBM\MQ) INSTVER(9.3.0.0)

QMNAME(QM910)                             STATUS(Ended immediately) INSTNAME(Installation2)
INSTPATH(C:\Program Files\IBM\MQ) INSTVER(9.3.0.0)
```

2. 停止 IBM MQ 9.3.0 进程。
   请参阅 在 Windows 上应用和除去维护，以获取有关停止进程的更多信息。
   总结:
   - 向队列管理器 QM 联格和 QM930 发出 **endmqm immediate** 命令。
   - 通过右键单击任务栏中的 IBM MQ 图标，然后单击 **停止 IBM MQ**，停止安装的 IBM MQ 服务。
3. 找到修订包。 此方案正在使用 IBM MQ 9.3。0.5。
   转至 IBM MQ 的建议修订 以获取最新修订的列表:
   - Continuous Delivery 发行版以及 Continuous Delivery 发行版的修订列表
   - Long Term Support 发行版的累积安全性更新和 Long Term Support 发行版的修订列表

   注: 您应该通过访问并下载相应的修订包来检查您是否正在使用最新的修订包。
   a) 单击相应的选项卡。
   对于此场景，它是 *V9.3.0.5 LTS*
   b) 从 Fix Central 或 Passport Advantage 下载 Windows 软件。 如果需要下载整个产品，
   在此场景中，您将文件集放入以下目录中: `C:\downloads\mq9305`; 文件名为 `9.3.0-IBM-MQ-Win64-FP0005.zip`。
4. 从 `.zip` 文件中抽取文件，然后运行以下命令: **IBM-MQ-9.3.0-FP0005.exe**。
   您将看到 安装 *Anywhere* 对话框，其中包含以下信息 `InstallAnywhere is preparing to install ...`
   您需要等到准备工作结束，这可能需要几分钟时间。
5. 当出现窗口 *IBM MQ (*修订包 *9.3.0.5* 文件*)* 时，单击 **确定** 以继续。
   a) 当显示 " 简介 " 部分时，单击 **下一步**。
   b) 当显示 安装类型 部分时，选择更适合您企业的选项 (可能是 **装入文件并应用修订包**)，然后单击 **下一步**。
   c) 当显示 " 信息 " 部分时，单击 **下一步**。

d) 显示 目标文件夹 部分时，选择缺省位置 `C:\Program Files\IBM\source\MQ 9.3.0.5`，然后单击 **下一步**。

e) 当显示 " 高级选项 " 部分时，单击 **下一步**。

f) 显示 安装前摘要 部分时，请检查显示的信息，然后单击 **安装**。

g) 正在装入代码，请稍候。

装入 部分具有进度指示器，当过程完成时，将勾选 装入完成 ，并显示 应用修订包 部分。

h) 单击**完成**。

由于系统上存在多个安装，因此将显示一个对话框，您可以在其中选择要升级的安装。 在本例中为 `Installation2 (9.3.0.0)`。

i) 单击**确定**。

j) 接受 备份文件夹 的缺省值，然后单击 **应用**。

将显示更多具有进度指示器的对话框，并且已应用最终对话框状态 修订包 *9.3.0.5* 。 单击 "完成" 以结束。

k) 单击**完成**。

结果

您已成功升级 IBM MQ for Windows 版本以及产品的现有版本。

Managed File Transfer 方案 (scenario)

常见 Managed File Transfer 拓扑简介，以及演示如何通过显示如何设置系统和传输测试消息来使用 Managed File Transfer 功能的方案。

- 常见拓扑
- 配置基本服务器

MFT 常见拓扑

本部分描述了常见的 Managed File Transfer 拓扑。 每张图中的双向箭头表示与队列管理器的连接。

请参阅第 135 页的『连接注意事项』，以获取有关队列管理器连接选项的更多信息。

包含一个队列管理器的基本拓扑

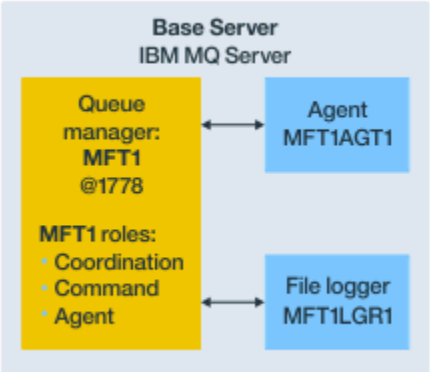


图 *25:* 包含一个队列管理器的基本拓扑

基本拓扑表示包含协调队列管理器的完整配置。 配置名称与协调队列管理器名称相同。 如果协调队列管理器名称为 MFT1，那么配置名称为 MFT1。

基本拓扑是您完成的第一个 Managed File Transfer 配置。 在完成基本配置后，来自远程服务器的伙伴代理将添加到基本配置以交换文件。

基本拓扑不在基本拓扑服务器之外交换文件。 但是，基本拓扑支持您将文件移动相同服务器中的不同位置，并且可用于开发目的。

包含一个伙伴代理的基本拓扑

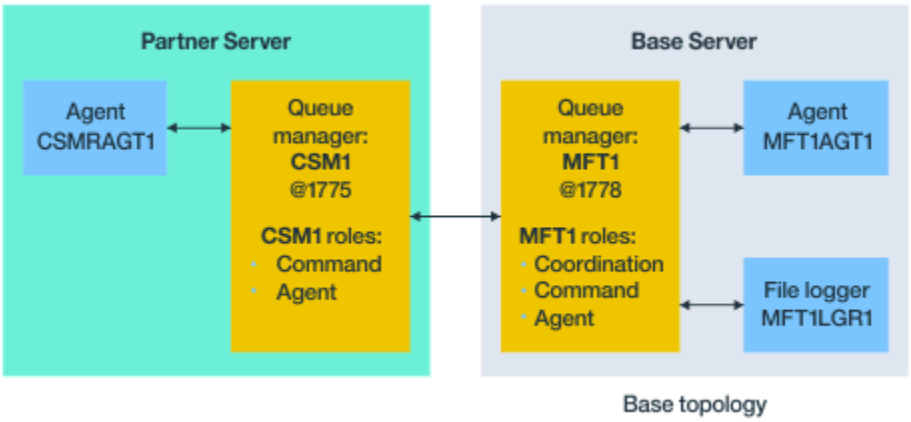


图 *26:* 包含一个伙伴代理的基本拓扑

此拓扑可在两个代理之间交换文件。 可通过与添加的第一个代理类似的方式添加额外的伙伴代理。

您可以将单个队列管理器用于所有三个 Managed File Transfer 队列管理器角色，或者可以将专用队列管理器用于特定角色。

例如，您可以有一个专门用于协调队列管理器角色的队列管理器，并且命令和代理角色可共享另一个队列管理器。

远程代理队列管理器（位于不同于基本配置的服务器中）与基本配置协调队列管理器之间的连接必须配置为 IBM MQ 客户机或 MQI 通道。

与协调队列管理器的连接通过 **fteSetupCoordination** 命令来建立。 如果未将协调队列管理器连接配置为 IBM MQ 客户机通道，那么在从伙伴代理服务器发出诸如 **fteListAgents** 之类的命令时，命令将失败。

具有单独的协调队列管理器和一个伙伴代理的基本拓扑

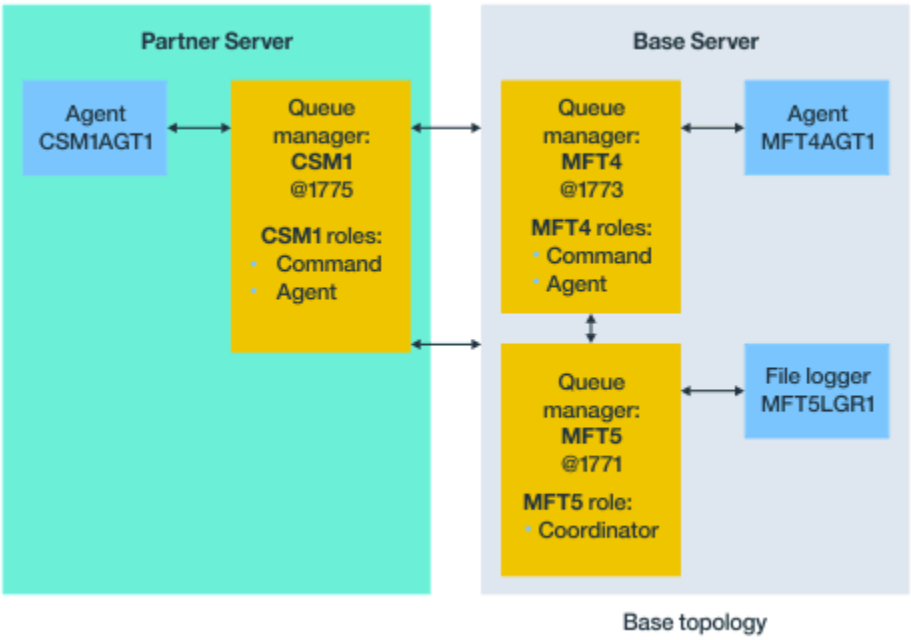


图 *27:* 具有单独的协调队列管理器和一个伙伴代理的基本拓扑

在图 3 的基本拓扑中，在基本服务器上，队列管理器 MFT4 由命令和代理角色共享，并且队列管理器 MFT5 专门用于协调队列管理器角色。

拓扑中的所有队列管理器（包括基本拓扑中的队列管理器 MFT4 和 MFT5）之间必须存在连接。

在伙伴服务器队列管理器上，队列管理器 CSM1 具有代理和命令队列管理器角色。

此拓扑可在两个代理之间交换文件。 每个伙伴代理必须连接到一个队列管理器，如图中所示。 可通过与添加的第一个伙伴代理类似的方式，添加额外的伙伴代理。

具有 Managed File Transfer Agent 伙伴的基本拓扑

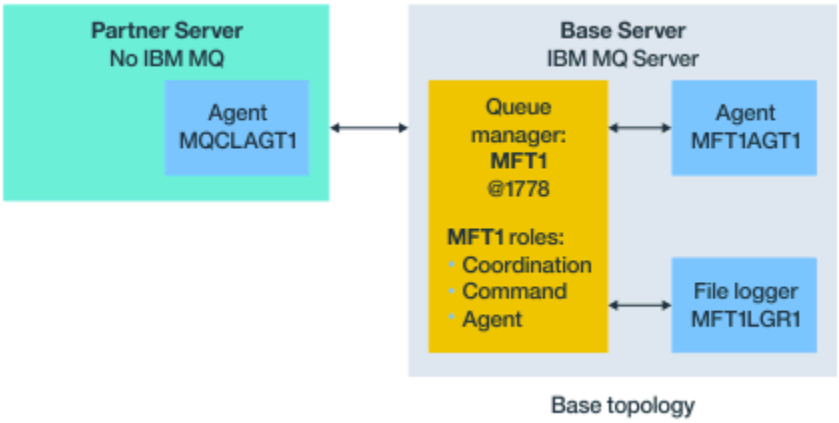


图 *28:* 具有 *Managed File Transfer Agent* 伙伴的基本拓扑

此拓扑可在两个代理之间交换文件。

伙伴代理中的服务器（在图中描述为 MQCLAGT1）未安装 IBM MQ 服务器。

伙伴代理将使用与 IBM MQ 已安装服务器相同的命令进行配置，但存在一些例外情况：

- 此伙伴代理的配置必须使用与一个或多个基本队列管理器的 IBM MQ 客户机连接。
- 无需运行在伙伴代理服务器中通过配置命令创建的协调队列管理器角色 IBM MQ 定义。 基本服务器中已存在协调队列管理器定义。

  但是，您必须：

  – 复制在伙伴服务器中创建代理时所生成的代理对象定义
  – 将定义文件传输到基本配置服务器，以及
  – 在标识为基本服务器中的代理队列管理器的队列管理器中创建定义。

    在此情况下，MFT1 充当所有三个角色，并且您将在 MFT1 队列管理器中为代理 MQCLAGT1 创建对象。

  作为将对象定义复制到基本服务器的替代方法，可以在代理队列管理器所在的基本服务器上针对代理 MQCLAGT1 运行 **fteDefine** 命令。 使用 **fteDefine** 命令生成的定义以在代理队列管理器上创建必需的代理定义。

  例如，在显示的图中，您会将文件 `MQCLAGT1_create.mqsc` 从伙伴服务器中的代理目录复制到基本配置服务器，并在 MFT1 队列管理器中创建必需的代理定义。

  在伙伴代理服务器上完成的配置将创建 Managed File Transfer 配置目录和必需的属性文件。

在伙伴服务器上，从 IBM MQ 9.1.0 开始，可以安装 Managed File Transfer 来自 Fix Central 的可再分发客户机。

注: MQMFT 可再分发组件客户机不同于 IBM MQ 可再分发组件客户机之处在于，它已打包且无需使用 **genmqpkg** 实用程序。 有关更多信息，请参阅可再分发组件客户机。

连接注意事项

在先前的图中，各个代理和队列管理器之间的每条线都表示与队列管理器的连接。

此连接可以是：

- 本地连接
- 绑定或消息通道连接，或者
- IBM MQ 客户机或 MQI 连接。

配置中选择的连接类型取决于指定的参数。

- 在指定不含其他连接参数的队列管理器名称参数时，那么您指定的是绑定连接。

  如果使用的队列管理器位于 Managed File Transfer 配置本地，那么在用于基本配置服务器中时，还表示本地连接。
- 如果指定队列管理器名称参数以及相应的主机、端口和通道名称参数，那么您指定的是 IBM MQ 客户机连接。

代理位于与代理队列管理器相同的主机上时，会产生本地连接的绑定类型规范更高效。

配置基本服务器

如何设置具有一个单独的配置队列管理器的基本服务器。

开始之前

以下示例假定：

- 您已查看第 135 页的『连接注意事项』部分并了解如何影响配置中队列管理器的连接类型。
- 您拥有有效的 IBM MQ 基础结构。 请参阅配置 IBM MQ 队列管理器以获取有关设置队列管理器的信息。
- IBM MQ 安全任务已完成。

  所有系统资源（例如，文件访问权）都配置了充足的安全性。

  对于 Managed File Transfer 安全配置，请参阅 保护 Managed File Transfer 和 限制用户对 MFT 代理操作的权限。
- 在通过使用样本程序发送和接收消息或者使用样本 **amqscnxc** 测试 IBM MQ 客户机类型连接来配置 IBM MQ 之后，将测试所有 IBM MQ 连接。

  通过在样本代码中定义通道连接，**amqscnxc** 样本连接到队列管理器，这类似于 Managed File Transfer 使用 MQI 或 IBM MQ 客户机类型连接时的连接方式。
- 指示信息假定您用于基本配置的服务器已安装一个 IBM MQ 版本。 如果基本服务器中有多个 IBM MQ 安装，那么必须小心地将正确的文件路径用于想要使用的 IBM MQ 版本。
- 在这些指示信息中使用的队列管理器不需要连接认证。

  虽然可以更简单地完成第一个配置而无需连接认证，但是如果企业需要立即使用连接认证，请参阅 MFT 和 IBM MQ 连接认证以获取有关如何配置 `MQMFTCredentials.xml` 凭证文件的指示信息。

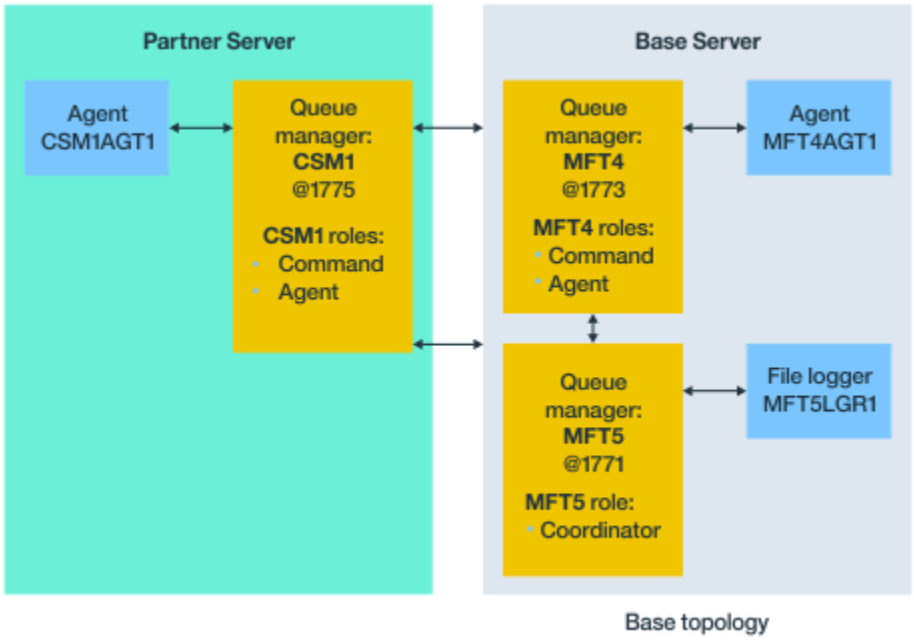


图 *29:* 具有单独的协调队列管理器和一个伙伴代理的基本拓扑

关于此任务

示例配置的队列管理器角色为:

- 基本服务器
  - 队列管理器 MFT5 是协调队列管理器
  - 队列管理器 MFT4 用作代理 MFT4AGT1 的代理队列管理器，而且还充当基本服务器上 MFT5 配置的命令队列管理器。
- 伙伴服务器
  - 队列管理器 CSM1 同时作为代理 CSM1AGT1 的代理队列管理器和伙伴服务器上 MFT5 配置的命令队列管理器。
  - 基本服务器上的队列管理器 MFT5 是协调队列管理器。

过程

1. 配置协调队列管理器
2. 配置命令队列管理器
3. 设置代理
4. 设置记录器
5. 配置伙伴服务器

下一步做什么

设置具有 MQMFT 的 MQExplorer，以便可测试示例设置。

配置协调队列管理器

如何配置协调队列管理器来协调文件传输。

开始之前

确保在为此场景设置的队列管理器之间具有完整连接。

关于此任务

此任务设置协调队列管理器 MFT5，并且此部分中的指示信息假定您使用的是一个 IBM MQ 安装。

如果您有多个安装，那么在启动任何配置任务之前，必须使用 setmqenv 命令将 IBM MQ 路径设置为所需的 IBM MQ 版本。

过程

1. 以 Managed File Transfer 管理员身份登录。
2. 发出以下命令以识别协调队列管理器并设置配置目录结构：

   ```
   fteSetupCoordination -coordinationQMgr MFT5
   ```

   协调队列管理器目录
   C:*data*\mqft\config\MFT5

   coordination.properties 文件
   C:*data*\mqft\config\MFT5\coordination.properties

   此命令还会生成一个必须针对协调队列管理器运行的 MQSC 命令文件：
   C:*data*\mqft\config\MFT5\MFT5.mqsc。
3. 切换到 C:*data*\mqft\config\MFT5 目录。
4. 通过运行以下命令，配置队列管理器以充当协调队列管理器。
   您需要提供在步骤“第 137 页的『2』”中发出的命令所生成的 MQSC 命令文件：

   ```
   runmqsc MFT5 < MFT5.mqsc > mft5.txt
   ```
5. 使用首选编辑器打开 mft5.txt 结果文件。 确保已成功创建定义。

下一步做什么

设置命令队列管理器。

配置命令队列管理器

如何配置命令队列管理器。

开始之前

确保您已配置协调队列管理器。 请参阅 第 136 页的『配置协调队列管理器』 以获取更多信息。

关于此任务

此任务标识命令队列管理器。

过程

发出以下命令：

```
fteSetupCommands -connectionQMgr MFT4
```

您将收到以下消息：BFGCL0245I: The file C:*data*\mqft\config\MFT4\command.properties has been created successfully.

命令队列管理器不需要外部 IBM MQ 定义。 运行 **fteSetupCommands** 后，将在 MFT5 配置目录中创建 command.properties 文件。

下一步做什么

设置代理。

设置代理

如何准备文件传输代理 MFT4AGT1，包括必须运行的 MQSC 脚本。

开始之前

您应该已经设置命令队列管理器。 请参阅 第 137 页的『配置命令队列管理器』 以获取更多信息。

关于此任务

此任务准备 Windows 文件传输代理 MFT4AGT1。

过程

1. 发出以下命令：

   ```
   fteCreateAgent -agentName MFT4AGT1 -agentQMgr MFT4
   ```

   在使用 **fteCreateAgent** 命令创建代理后，会将 agents 目录以及代理 MFT4AGT1 的子目录添加到 MFT5 目录。

   在 *data*\MFT5\agents\MFT4AGT1 目录中，查找：

   - agent.properties 文件
   - MFT4AGT1_create.mqsc 文件，其中包含代理所需的 IBM MQ 定义。
2. 切换到 *data*\MFT5\agents\MFT4AGT1 目录，并通过发出以下命令来创建必需的代理队列管理器定义：

   ```
   runmqsc MFT4 < MFT4AGT1_create.mqsc > mft4.txt
   ```
3. 使用首选编辑器打开 mft4.txt 结果文件并确保已成功创建定义。
4. 通过输入以下命令启动代理：**fteStartAgent** MFT4AGT1。
5. 通过输入以下命令显示代理：**fteListAgents**。

   您应该会看到类似于以下内容的输出：

   ```
   5655-MFT, 5724-H72 Copyright IBM Corp. 2008, 2025.  ALL RIGHTS RESERVED
   BFGPR0127W: No credentials file has been specified to connect to IBM MQ.
   Therefore, the assumption is that IBM MQ authentication has been disabled.
   Agent Name:     Queue Manager Name:     Status:
   MFT4AGT1        MFT4                    READY
   ```

   注: 如果尚未在 Managed File Transfer 环境中启用连接认证，那么可忽略 BFGPR0127W 消息。

   如果发出 **ftelistAgents** 命令并接收到以下消息: BFGCL0014W: **不存在与当前选择标准匹配的代理程序。** 请参阅 如果 **fteListAgents** 命令未列出 MFT 代理程序，该怎么办。

下一步做什么

设置记录器。

设置记录器

需要文件或数据库记录器以保留与配置的传输活动相关的历史记录和审计信息。 在此示例中，您将创建文件记录器。

开始之前

您必须已设置：

- 配置队列管理器
- 命令队列管理器

- 代理程序

过程

1. 发出以下命令：

   ```
   fteCreateLogger -loggerQMgr MFT5 -loggerType FILE
   -fileLoggerMode CIRCULAR -fileSize 5MB -fileCount 3 MFT5lgr1
   ```

   在运行 **fteCreateLogger** 命令后，将创建具有 MFT5LGR1 子目录的 *data*\mqft\config\MFT5\loggers 目录。

   MFT5LGR1 子目录保留 logger.properties 文件。 另外，在该目录中还有一个名为 MFT5LGR1_create.mqsc 的文件，其中包含记录器所需的 IBM MQ 定义。
2. 切换到目录 *data*\mqft\config\MFT5\loggers\MFT5LGR1。
3. 运行关联的 MQSC 命令文件。

   ```
   runmqsc MFT5 < MFT5_create.mqsc
   ```

   以创建记录器所需的定义。

   a) 查看对象定义的结果以确认已成功创建必需的对象。
4. 通过发出以下命令启动记录器：**fteStartLogger** MFT5LGR1。
5. 查看 *data*\mqft\logs\MFT5\loggers\MFT5LGR1\logs 中 output0.log 文件的内容。

   在有关记录器的一些信息之后，最后一句应包含以下消息：BFGDB0023I: The logger has completed startup activities and is now running。

   在记录器第一次启动时，偶尔可能不会将日志信息写入到 output0.log。 如果 output0.log 文件为空，请通过输入 **fteStopLogger** MFT5LGR1 并按 **Enter** 键来重新启动记录器。

   通过输入 **fteStartLogger** MFTULGR1 并按 **Enter** 键来重新启动记录器。 现在，文件 output0.log 将显示数据。

   在第一次启动代理时，相同的行为将扩展到 output0.log 文件的代理版本。

   通过使用 **fteStopAgent** 和 **fteStartAgent** 命令来停止和启动代理。 然后，您将看到日志数据写入代理 output0.log 文件。

结果

您已配置基本服务器，其中包含此配置的协调队列管理器。

下一步做什么

现在，您针对包含远程代理的伙伴服务器执行类似工作。

配置伙伴服务器

在基本服务器具有单独的协调队列管理器时如何配置伙伴服务器

开始之前

确保已完全完成设置包含配置队列管理器的基本服务器的所有任务。

关于此任务

针对 IBM MQ 和安全配置以及 IBM MQ 路径的假定也适用于伙伴服务器。

首先设置 MFT5 配置目录并使用 **fteSetupCoordination** 命令来标识协调队列管理器。

过程

1. 通过发出以下命令来创建伙伴服务器配置目录：

```
fteSetupCoordination -coordinationQMgr MFT5
-coordinationQMgrHost 177.16.20.15 -coordinationQMgrPort 1771
-coordinationQMgrChannel MQMFT.MFT5.SVRCONN
```

注意：

a. 在协调队列管理器位于与伙伴服务器不同的服务器上时，必须将与基本服务器协调队列管理器的连接定义为客户机连接。

如果未在伙伴服务器上将协调队列管理器连接定义为 IBM MQ 客户机连接，将导致连接到协调队列管理器的任何 Managed File Transfer 命令失败。

连接到协调队列管理器的命令的示例为 **fteListAgents**。

b. 您无需创建 IBM MQ 定义，因为在配置基本服务器时已完成协调队列管理器所需的定义。

2. 通过发出以下命令来标识命令队列管理器：

```
fteSetupCommands -connectionQMgr CSM1
```

命令队列管理器不需要任何外部 IBM MQ 定义。

3. 通过发出以下命令来标识伙伴代理队列管理器并创建伙伴代理队列管理器：

```
fteCreateAgent -agentName CSM1AGT1 -agentQMgr CSM1
```

4. 切换到 CSM1AGT1 目录。
5. 通过发出以下命令来创建代理所需的 IBM MQ 定义：

```
runmqsc CSM1 < CSM1AGT1_create.mqsc > csm1.txt
```

a) 使用首选编辑器打开文件 csm1.txt 以确认已成功创建代理所需的所有定义。

6. 通过发出以下命令来启动代理：

```
fteStartAgent CSM1AGT1
```

7. 通过输入 **fteListAgents** 来显示代理
   您应该会看到类似于以下内容的输出：

```
C:\>fteListAgents
5655-MFT, 5724-H72 Copyright IBM Corp. 2008, 2025.  ALL RIGHTS RESERVED
BFGPR0127W: No credentials file has been specified to connect to IBM MQ. Therefo
re, the assumption is that IBM MQ authentication has been disabled.
Agent Name:     Queue Manager Name:     Status:
CSM1AGT1        CSM1                    READY
MFT4AGT1        MFT4                    READY
```

注： 如果尚未在 Managed File Transfer 环境中启用连接认证，那么可忽略 BFGPR0127W 消息。

如果发出 **ftelistAgents** 命令并接收到以下消息: BFGCL0014W: **不存在与当前选择标准匹配的代理程序。** 请参阅 如果 **fteListAgents** 命令未列出 MFT 代理程序，该怎么办。

如果其中一个代理程序的状态为 UNREACHABLE，请参阅 在代理程序显示为处于 UNKNOWN 状态时要执行的操作。

使用 MFT 设置 IBM MQ Explorer

此任务帮助您将 IBM MQ Explorer 连接到 Managed File Transfer 配置。

过程

1. 启动 IBM MQ Explorer。
2. 在左侧的 " Navigator " 面板中，向下滚动并展开文件夹: Managed File Transfer。

您会看到协调队列管理器的条目：MFT5

3. 右键单击 MFT5 并选择**连接**。
   a) 选择显示的下拉菜单中的 Agents，并确保 MFT4AGT1 和 CSMAGT1 这两个代理均处于 Ready 状态。

下一步做什么

使用 IBM MQ Explorer 测试示例设置。

使用 IBM MQ Explorer 来测试文件传输

此任务提供了一个示例，说明如何将 IBM IBM MQ Explorer 与 Managed File Transfer 配合使用，以在设置 IBM MQ Explorer 后测试文件传输，如上一个主题中所述。

开始之前

确保您拥有一个有效的系统，代理处于 READY 状态并且 IBM MQ Explorer 正常工作。 有关更多信息，请参阅 第 140 页的『使用 MFT 设置 IBM MQ Explorer』。

关于此任务

确定用于测试传输的文件以及要将其复制到的目录。 对于此示例，假定使用 C:\temp\mft 目录中的文件 test-file.txt。

```
C:\temp\mft> dir *
Date stamp 61 test-file.txt
1 File(s) 61 bytes
```

过程

1. 在 Windows 中启动 IBM MQ Explorer
2. 在左侧的 " Navigator " 面板中，展开文件夹: Managed File Transfer。
   您会看到协调队列管理器的条目：MFT5
3. 右键单击 MFT5 并选择**连接**。
4. 连接后，右键单击 MFT5 并选择**新建传输**
   a) 使用下拉菜单以针对源代理选择 MFT4AGT1，并针对目标代理选择 CSMAGT1。
   b) 单击**下一步**。
   c) 单击下一个窗口上的**添加**。
      此时将显示一个宽对话框。 左侧用于 Source，右侧用于 Destination。
5. 在 Source 面板上：
   a) 选择**文本传输**，因为文件是文本。
   b) 选择**浏览**以查找文件。
      在本例中，文件为 C:\temp\mft\test-file.txt。

      注意: 请勿单击**确定**，因为您需要完成 Destination 面板。
6. 在 Destination 面板上：
   a) 输入要为目标上的文件指定的名称，例如，test-file.txt。
      支持使用相对路径。 完整路径的顶部是启动目标代理的用户标识的主目录。
   b) 如果需要此选项，请选择 Overwrite files if present。
   c) 单击**确定**。
      选中的文件将在**“新建传输”**面板中显示。
7. 如果 MFT5 配置菜单已关闭，并且显示 +MFT5，请通过单击 **+** 号来展开该菜单。

8. 保留在选中的 MFT 配置上。
   接下来，通过执行以下过程来检查传输的状态。
9. 单击协调队列管理器 MFT5 下面的**传输日志**。
10. 查看紧靠在 **传输日志** 顶部面板下方的 `Managed File Transfer - Current Transfer progress panel` 中的状态，并等待传输完成。
    如果传输显示成功且背景为绿色，那么您已成功完成配置测试。
    如果传输失败且背景为红色，那么发生了错误。
    在大多数情况下，您可以使用顶部**传输日志**面板下方的滚动条来查看故障原因的摘要。
    a) 如果无法确定传输失败的原因，请在顶部**传输日志**面板中双击该传输的条目。
    b) 选择所显示弹出面板的左侧窗格中的 XML。
    c) 滚动浏览信息以确定错误原因。
    d) 执行所需的纠正并重新测试传输。

开始使用 IBM MQ Internet Pass-Thru

这些方案显示如何设置一些简单的 IBM MQ Internet Pass-Thru (MQIPT) 配置。 您也可以使用这些任务来确认已成功安装该产品。

开始之前

在开始使用这些方案之前，请确保已完成以下先决条件:

- 您熟悉在 IBM MQ 上定义队列管理器，队列和通道。
- 已经安装 IBM MQ 客户机和服务器。
- MQIPT 安装在 Windows 系统上名为 `C:\mqipt` 的目录中。 （这些示例是为 Windows 系统而编写的，但可以在任何受支持的平台上运行。） 有关安装 MQIPT 的更多信息，请参阅 安装 MQIPT。
- 客户机、服务器和所有 MQIPT 实例都安装在单独的计算机上。
- 熟悉如何使用 **amqsputc** 命令来将消息放入队列中。
- 熟悉如何使用 **amqsgetc** 命令来从队列中获取消息。
- 熟悉如何在 IBM MQ 中设置客户机权限。

关于此任务

完成先决条件后，请完成以下初始步骤，以设置准备好在方案中工作的系统。

过程

1. 在 IBM MQ 服务器上，完成以下任务:
   - 定义名为 `MQIPT.QM1` 的队列管理器。
   - 定义名为 `MQIPT.CONN.CHANNEL` 的服务器连接通道。
   - 定义名为 `MQIPT.LOCAL.QUEUE` 的本地队列。
   - 在端口 1414 上启动针对 `MQIPT.QM1` 的 TCP/IP 侦听器。 如果另一个应用程序已在使用端口 1414，请选择空闲的端口地址并将以下示例中的端口值替换为该端口地址。
   - 确保将连接认证和通道认证配置为允许使用您的用户标识从客户端机器建立客户机连接。 如果将连接认证设置为需要应用程序为客户机连接提供认证凭证，那么在运行 **amqsputc** 和 **amqsgetc** 命令之前，需要设置下列其中一个环境变量:

     MQSAMP_USER_ID
     如果要使用用户标识和密码向队列管理器进行认证，请设置为用于连接认证的用户标识。

     V 9.3.4 Linux AIX **MQSAMP_TOKEN**
     如果要使用认证令牌向队列管理器进行认证，请设置为非空白值。

2. 通过使用 **amqsputc** 命令将消息放在队列管理器的本地队列上，然后使用 **amqsgetc** 命令检索消息，测试从 IBM MQ 客户机到队列管理器的路由。

   要准备此部分中的方案，请创建并编辑 `mqipt.conf` 文件，如下所示:

   a. 将可在 MQIPT 安装目录的 `samples` 子目录中找到的 `mqiptSample.conf` 文件复制到所选 MQIPT 主目录中的 `mqipt.conf` 。 以下场景将使用 `C:\mqiptHome` 作为 MQIPT 主目录。

   b. 除了 `mqipt.conf` 外，还创建两个名为 `errors` 和 `logs` 的目录。 设置这些目录的文件许可权，使运行 MQIPT 的用户标识可以写入这些目录。

   c. 删除 `mqipt.conf` 文件中的所有路径。

   d. 在其余 `[global]` 部分中，检查 **ClientAccess** 是否存在并设置为 `true`。

下一步做什么

设置系统后，您已准备好启动以下方案:

- 第 143 页的『验证 MQIPT 是否正常工作』
- 第 145 页的『创建密钥环文件』
- 第 147 页的『创建测试证书』
- 第 148 页的『认证 TLS 服务器』
- 第 150 页的『认证 TLS 客户机』
- 第 156 页的『配置 HTTP 隧道』
- 第 158 页的『配置访问控制』
- 第 160 页的『配置 SOCKS 代理』
- 第 161 页的『配置 SOCKS 客户机』
- 第 162 页的『配置 MQIPT 集群支持』
- 第 165 页的『分配端口号』
- 第 167 页的『使用 LDAP 服务器检索 CRL』
- 第 170 页的『以 TLS 代理方式运行 MQIPT』
- 第 171 页的『使用安全管理器以 TLS 代理方式运行 MQIPT』
- 第 174 页的『使用安全出口』
- 第 175 页的『使用安全出口将客户机连接请求路由到 IBM MQ 队列管理器服务器』
- 第 178 页的『动态路由客户机连接请求』
- 第 181 页的『使用证书出口来认证 TLS 服务器』

验证 MQIPT 是否正常工作

使用此简单配置设置可确保正确安装了 MQIPT 。

开始之前

- 开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。

关于此任务

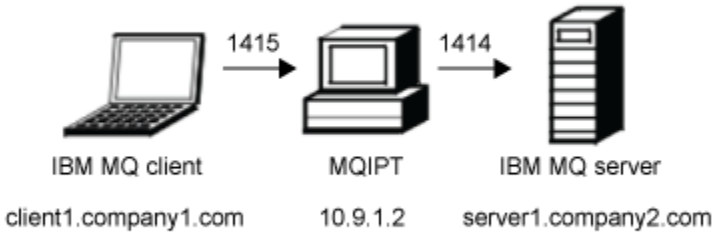


图 *30:* 安装验证测试网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）通过 MQIPT 连接到 IBM MQ 服务器（称为 server1.company2.com，位于端口 1414 上）。

过程

要验证 MQIPT 能否正常工作，请完成以下步骤：

1. 定义 MQIPT 路由。

   在 MQIPT 计算机上，编辑 `mqipt.conf` 以添加以下路径定义：

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

2. 启动 MQIPT。

   打开命令提示符并输入以下命令：

   ```
   C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
   ```

   其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt1` 是要提供给 MQIPT 实例的名称。

   以下消息指示 MQIPT 已成功启动：

   ```
   5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
   MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
   MQCPI004 Reading configuration information from mqipt.conf
   MQCPI152 MQIPT name is ipt1
   MQCPI021 Password checking has been enabled on the command port
   MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
   MQCPI006 Route 1415 has started and will forward messages to:
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....using MQ protocol
   MQCPI078 Route 1415 ready for connection requests
   ```

3. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令：

   a) 设置 **MQSERVER** 环境变量：

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

   b) 留言：

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
   Hello world
   ```

   输入消息字符串之后，按 Enter 键两次。

   c) 获取消息：

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

此时会返回消息“Hello world”。

4. 要停止 IBM MQ，请输入以下命令:

```
mqiptAdmin -stop -n ipt1
```

创建密钥环文件

在此场景中，您可以请求证书并创建密钥环文件以允许 MQIPT 来使用 TLS。

开始之前

开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。

此任务假定您使用 **mqiptKeyman** (iKeyman) 从可信认证中心 (CA) 请求新证书，并且您的个人证书将以文件 (例如 `server.cer`) 形式返回给您。 这足以执行服务器认证。 如果需要客户机认证，那么必须请求第二个证书 (例如， `client.cer`) ，并执行以下步骤两次以创建两个密钥环文件。

关于此任务

您可以使用 **mqiptKeycmd** 命令行界面 (CLI) 或 **mqiptKeyman** GUI 来请求证书。 然后，您应该指定包含 **SSLServerKeyRing** 或 **SSLClientKeyRing** MQIPT 路由属性，具体取决于该证书是供入站连接还是出站连接使用。

过程

使用下列其中一种方法来创建密钥环文件:

- 使用 **mqiptKeycmd** 命令行界面 (CLI)

  a) 输入以下命令以创建新的 PKCS #12 密钥环文件:

  ```
  mqiptKeycmd -keydb -create -db server_name.pfx -pw password -type pkcs12
  ```

  其中:

  – **-db** 指定密钥环文件 (*server_name*.pfx) 的名称。

  – **-pw** 指定以后必须使用 **mqiptPW** 命令进行加密的密钥环密码 (*password*)。

  b) 输入以下命令以创建新的证书请求:

  ```
  mqiptKeycmd -certreq -create -db server_name.pfx -pw password -type pkcs12
              -file cert_file_name.req -label label -dn DN_identity
              -sig_alg signature_algorithm -size key_size
  ```

  其中:

  – **-file** 指定所请求证书的文件名。

  – **-label** 指定您所选的唯一名称（建议不要包含空格字符）。

  – **-dn** 指定 MQIPT 路径的相应专有名称标识；例如， `"CN=Test Certificate,OU=Sales,O=Example,C=US"`。

  – **-sig_alg** 指定散列算法; 例如， `SHA256WithRSA`。

  – **-size** 指定公用密钥的大小；例如，2048。

  如果使用给定的示例值，该命令将创建使用 2048 位 RSA 公用密钥的数字证书以及将 RSA 与 SHA-256 散列算法结合使用的数字签名。

  创建证书时，请谨慎选择能满足贵组织安全性需求的公用密钥加密算法、密钥大小和数字签名算法。 有关更多信息，请参阅 MQIPT 的数字证书注意事项 。

将该命令创建的证书请求文件 (*cert_file_name*.req) 发送到认证中心进行签署。

c) 从 CA 接收已签名的个人证书时，请输入以下命令以将该证书添加到服务器密钥环:

```
mqiptKeycmd -cert -receive -db server_name.pfx -pw password
            -type pkcs12 -file cert_file_name.crt
```

- 使用 **mqiptKeyman** GUI

a) 通过运行以下命令打开 GUI:

```
mqiptKeyman
```

b) 单击 **密钥数据库文件** > **新建**。

c) 选择密钥数据库的类型 PKCS12。

d) 输入新密钥环文件的文件名和位置。

单击**确定**。

e) 输入并确认新密钥环文件的密码。

这是以后必须使用 **mqiptPW** 命令进行加密的密钥环密码。 单击 **确定** 以创建新的个人证书密钥环文件。

f) 通过单击 **创建** > **新建证书请求**来创建证书请求。

g) 在**密钥标签**字段中输入新证书的标签。

该标签可以是您所选的任何唯一名称（建议不要包含空格字符）。

h) 选择能满足贵组织安全性需求的密钥大小和数字签名算法。

有关更多信息，请参阅 MQIPT 的数字证书注意事项 。

i) 在可选的专有名称字段中，输入 MQIPT 路径的相应专有名称标识。

j) 输入要创建的证书请求的文件名，然后单击**确定**。

这样会生成该证书请求，并使用您指定的名称来保存该证书请求。 将此文件发送到认证中心进行签署。

k) 从 CA 接收已签名的个人证书时，必须在密钥环文件中接收该证书。

在**"密钥数据库内容"**面板中，从下拉列表中选择**个人证书**。 然后单击**接收**。

l) 输入用于存储已签署证书的文件的名称，然后单击**确定**。

下一步做什么

您还必须确保签署个人证书的 CA 的 CA 证书存在于 CA 密钥环文件中。 根据 MQIPT 配置， CA 密钥环文件可能是与个人证书密钥环文件不同的文件。

要使用单独的 CA 密钥环文件，可以使用 MQIPT 随附的名为 `sslCAdefault.pfx` 的样本 CA 密钥环文件，也可以创建新的 PKCS #12 密钥环文件。 您需要将签署个人证书的 CA 的公用 CA 证书添加到 CA 密钥环，除非该证书已存在于样本密钥环文件中。 公用 CA 证书可能已随您的个人证书一起返回。 如果没有，那么必须从提供个人证书的同一 CA 请求 CA 证书，然后将其添加到密钥环。

要添加 CA 证书，可以使用 **mqiptKeycmd** CLI 或 **mqiptKeyman** GUI。

要使用 **mqiptKeycmd** CLI 添加 CA 证书，请发出以下命令:

```
mqiptKeycmd -cert -add -db sslCAdefault.pfx -pw password -type pkcs12
            -file ca_file_name.crt -label label
```

其中:

- **-db** 指定 CA 密钥环文件名，在本例中为 `sslCAdefault.pfx`。
- **-pw** 指定密钥环密码。 名为 `sslCAdefault.pfx` 的样本 CA 密钥环文件的密码为 `mqiptSample`。
- **-file** 指定认证中心所返回的文件的名称。
- **-label** 指定您所选的唯一名称（建议不要使用空格字符）。

要使用 iKeyman GUI 来添加认证中心证书:

- 在“密钥数据库内容”面板中，从下拉列表中选择**签署者证书**。
- 单击**添加**。
- 输入包含认证中心证书的文件的名称，然后单击**确定**。
- 输入认证中心证书的标签。 该标签可以是您所选的任何唯一名称（建议不要使用空格字符）。 单击**确定**。

通过发出以下命令对密钥环密码进行加密:

```
mqiptPW
```

输入提示时要加密的密钥环密码。 根据证书是供入站连接还是出站连接使用，将 **mqipt.conf** 配置文件中相应属性的值设置为 **mqiptPW** 命令输出的加密密码; 例如， **SSLServerKeyRingPW** 或 **SSLClientKeyRingPW**。 有关加密密钥环密码的更多信息，请参阅 加密存储的密码 以加密密钥环密码。

要将这些新的密钥环文件用于服务器认证，请将这些密钥环文件放在名为 ssl 的目录中的 MQIPT 主目录下，并设置以下路由属性:

```
SSLClientCAKeyRing=C:\\mqiptHome\\ssl\\sslCAdefault.pfx
SSLClientCAKeyRingPW=encrypted_password
SSLServerKeyRing=C:\\mqiptHome\\ssl\\myServer.pfx
SSLServerKeyRingPW=encrypted_password
SSLServerCAKeyRing=C:\\mqiptHome\\ssl\\sslCAdefault.pfx
SSLServerCAKeyRingPW=encrypted_password
```

请参阅方案 第 148 页的『认证 TLS 服务器』，以获取有关配置 MQIPT 以使用 TLS 的更多信息。

创建测试证书

在此场景中，可以创建一个自签名证书以用于测试 MQIPT 路径。 MQIPT 路径可以使用此证书来向远程同级标识自身。

自签名证书在测试场景中很有用，在这些场景中，您必须确保 TLS 连接，而无需向认证中心 (CA) 支付证书费用。 但是，不应在生产环境中使用自签名证书。 如果需要用于生产环境的证书，请参阅第 145 页的『创建密钥环文件』。

开始之前

- 开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。

关于此任务

您可以使用 **mqiptkeycmd** (iKeyman) 命令行界面 (CLI) 或 **mqiptkeyman** GUI 来请求证书。 然后，您应该在 **SSLServerKeyRing** 或 **SSLClientKeyRing** MQIPT 路由属性中指定包含该证书的密钥环文件，具体取决于该证书是供入站连接还是出站连接使用。

过程

请使用以下某种方法来创建测试证书:

- 使用命令行界面 (CLI)

  a) 输入以下命令以创建新的 PKCS #12 密钥环文件:

  ```
  mqiptKeycmd -keydb -create -db server_name.pfx -pw password -type pkcs12
  ```

  其中:

  – **-db** 指定密钥环文件 (*server_name*.pfx) 的名称。

  – **-pw** 指定以后必须使用 **mqiptPW** 实用程序进行加密的密钥环密码 (*password*)。

  b) 输入以下命令以创建用于测试的自签名个人证书:

```
mqiptKeycmd -cert -create -db server_name.pfx -pw password -type pkcs12
            -label label -dn DN_identity
            -sig_alg signature_algorithm -size key_size
```

其中：

- **-label** 指定您所选的唯一名称（建议不要包含空格字符）。
- **-dn** 指定 MQIPT 路径的相应专有名称标识；例如，"CN=Test Certificate,OU=Sales,O=Example,C=US"。
- **-sig_alg** 指定散列算法; 例如， SHA256WithRSA。
- **-size** 指定公用密钥的大小；例如，2048。

如果使用给定的示例值，该命令将创建使用 2048 位 RSA 公用密钥的数字证书以及将 RSA 与 SHA-256 散列算法结合使用的数字签名。

创建证书时，请谨慎选择能满足贵组织安全性需求的公用密钥加密算法、密钥大小和数字签名算法。有关更多信息，请参阅 MQIPT 的数字证书注意事项 。

- 使用 GUI

a) 通过运行以下命令打开 GUI:

```
mqiptKeyman
```

b) 单击 **密钥数据库文件** > **新建**。

c) 选择密钥数据库的类型；例如，PKCS12。

d) 输入新密钥环文件的文件名和位置。

单击**确定**。

e) 输入新密钥环文件的密码。

再次输入该密码进行确认。 这是以后必须使用 **mqiptPW** 实用程序加密的密钥环密码。 单击 **确定** 以创建新的个人证书密钥环文件。

f) 通过单击 **创建** > **新建自签名证书**来创建新的自签名个人证书。

g) 在**密钥标签**字段中输入新证书的标签。

该标签可以是您所选的任何唯一名称（建议不要包含空格字符）。

h) 选择能满足贵组织安全性需求的密钥大小和数字签名算法。

有关更多信息，请参阅 MQIPT 的数字证书注意事项 。

i) 在可选的专有名称字段中，输入 MQIPT 路径的相应专有名称标识，然后单击**确定**。

下一步做什么

通过发出以下命令对密钥环密码进行加密:

```
mqiptPW
```

输入提示时要加密的密钥环密码。 根据证书是供入站连接还是出站连接使用，将 **mqipt.conf** 配置文件中相应属性的值设置为 **mqiptPW** 命令输出的加密密码; 例如， **SSLServerKeyRingPW** 或 **SSLClientKeyRingPW**。 有关加密密钥环密码的更多信息，请参阅 加密存储的密码 以加密密钥环密码。

认证 TLS 服务器

在此场景中，您可以使用样本 (sslSample.pfx) 密钥环文件中的测试证书 (在 MQIPT 安装目录的 samples/ssl 子目录中随 MQIPT 一起提供) 来测试 TLS 连接。

开始之前

在开始使用此方案之前，请确保已完成 第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务，并已阅读主题 MQIPT 中的 SSL/TLS 支持。

关于此任务

通过两个 MQIPT 实例在 IBM MQ 客户机与 IBM MQ 服务器之间建立连接。 MQIPT 1 与 MQIPT 2 之间的连接使用 TLS， MQIPT 1 充当 TLS 客户机， MQIPT 2 充当 TLS 服务器。

在 TLS 握手期间，服务器将其测试证书发送到客户机，并且客户机使用其证书副本 (设置了信任即对等标志) 来认证服务器。 将使用密码套件 SSL_RSA_WITH_AES_256_CBC_SHA256。 此方案中的 `mqipt.conf` 配置文件基于在 第 143 页的『验证 MQIPT 是否正常工作』 方案中创建的配置文件。 有关如何创建此示例中使用的测试证书的详细信息，请参阅第 147 页的『创建测试证书』。

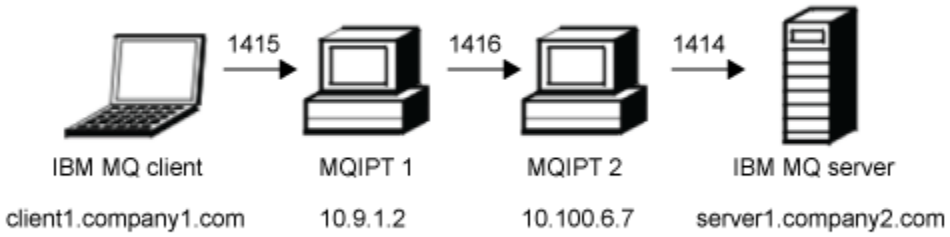


图 *31: SSL/TLS* 服务器网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）通过 MQIPT 的两个实例来连接到 IBM MQ 服务器（称为 server1.company2.com，位于端口 1414 上）。

过程

要认证 TLS 服务器，请完成以下步骤:

1. 在 MQIPT 1 系统上:

   a) 编辑 `mqipt.conf` 并添加以下路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=10.100.6.7
   DestinationPort=1416
   SSLClient=true
   SSLClientKeyRing=C:\\mqipt\\samples\\ssl\\sslSample.pfx
   SSLClientKeyRingPW=<mqiptPW>1!PCaB1HWrFMOp43ngjwgArg==!6N/vsbqru7iqMhFN+wozxQ==
   SSLClientCipherSuites=SSL_RSA_WITH_AES_256_CBC_SHA256
   ```

   b) 打开命令提示符，并启动 MQIPT:

   ```
   C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
   ```

   其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt1` 是要提供给 MQIPT 实例的名称。

   以下消息指示 MQIPT 已成功启动:

   ```
   5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
   MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
   MQCPI004 Reading configuration information from mqipt.conf
   MQCPI152 MQIPT name is ipt1
   MQCPI021 Password checking has been enabled on the command port
   MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
   MQCPI006 Route 1415 is starting and will forward messages to :
   MQCPI034 ....10.100.6.7(1416)
   MQCPI035 ....using MQ protocol
   MQCPI036 ....SSL Client side enabled with properties :
   MQCPI139 ......secure socket protocols <NULL>
   MQCPI031 ......cipher suites SSL_RSA_WITH_AES_256_CBC_SHA256
   MQCPI032 ......key ring file C:\\mqipt\\samples\\ssl\\sslSample.pfx
   MQCPI047 ......CA key ring file <NULL>
   ```

```
MQCPI071 ......site certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 ......peer certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI078 Route 1415 ready for connection requests
```

2. 在 MQIPT 2 系统上:

   a) 编辑 `mqipt.conf` 并添加以下路由定义:

```
[route]
ListenerPort=1416
Destination=Server1.company2.com
DestinationPort=1414
SSLServer=true
SSLServerKeyRing=C:\\mqipt\\samples\\ssl\\sslSample.pfx
SSLServerKeyRingPW=<mqiptPW>1!PCaB1HWrFMOp43ngjwgArg==!6N/vsbqru7iqMhFN+wozxQ==
SSLServerCipherSuites=SSL_RSA_WITH_AES_256_CBC_SHA256
```

   b) 打开命令提示符，并启动 MQIPT:

```
C:
cd \mqipt\bin
mqipt .. -n ipt2
```

   其中 `..` 指示 MQIPT 配置文件 `mqipt.conf` 位于父目录中， `ipt2` 是要提供给 MQIPT 实例的名称。

   以下消息指示 MQIPT 已成功启动:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqipt\logs will be used to store the log files
MQCPI006 Route 1416 is starting and will forward messages to :
MQCPI034 ....Server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI037 ....SSL Server side enabled with properties :
MQCPI139 ......secure socket protocols <NULL>
MQCPI031 ......cipher suites SSL_RSA_WITH_AES_256_CBC_SHA256
MQCPI032 ......key ring file C:\\mqipt\\samples\\ssl\\sslSample.pfx
MQCPI047 ......CA key ring file <NULL>
MQCPI071 ......site certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 ......peer certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI033 ......client authentication set to false
MQCPI078 Route 1416 ready for connection requests
```

3. 在 IBM MQ 客户机的命令提示符中，输入以下命令:

   a) 设置 **MQSERVER** 环境变量:

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

   b) 放置消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

   输入消息字符串之后，按 Enter 键两次。

   c) 获取消息:

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

   此时会返回消息“Hello world”。

认证 TLS 客户机

在此场景中，您可以使用样本测试证书来测试 TLS 连接，以执行服务器和客户机认证。

开始之前

在开始使用此方案之前，请确保已完成 第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务，并已阅读主题 MQIPT 中的 SSL/TLS 支持。

关于此任务

通过两个 MQIPT 实例在 IBM MQ 客户机与 IBM MQ 服务器之间建立连接。 MQIPT 1 与 MQIPT 2 之间的连接使用 TLS， MQIPT 1 充当 TLS 客户机， MQIPT 2 充当 TLS 服务器。

在 TLS 握手期间，服务器将其测试证书发送到客户机。 客户机使用具有“作为同级信任”标记的证书副本来认证服务器。 然后，客户机会将其测试证书发送到服务器。 服务器使用具有“作为同级信任”标记的证书副本来认证客户机。 将使用密码套件 SSL_RSA_WITH_AES_256_CBC_SHA256。 此方案中的 `mqipt.conf` 配置文件基于在 第 143 页的『验证 MQIPT 是否正常工作』 方案中创建的配置文件。

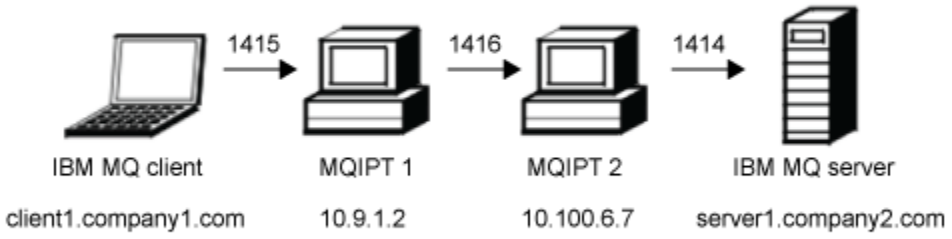


图 *32: SSL/TLS* 客户机网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）通过 MQIPT 的两个实例来连接到 IBM MQ 服务器（称为 server1.company2.com，位于端口 1414 上）。

过程

要认证 TLS 客户机，请完成以下步骤:

1. 在 MQIPT 1 系统上:

   a) 编辑 `mqipt.conf` 并添加以下路由定义:

```
[route]
ListenerPort=1415
Destination=10.100.6.7
DestinationPort=1416
SSLClient=true
SSLClientKeyRing=C:\\mqipt\\samples\\ssl\\sslSample.pfx
SSLClientKeyRingPW=<mqiptPW>1!PCaB1HWrFMOp43ngjwgArg==!6N/vsbqru7iqMhFN+wozxQ==
SSLClientCipherSuites=SSL_RSA_WITH_AES_256_CBC_SHA256
```

   b) 打开命令提示符，并启动 MQIPT:

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

   其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt1` 是要提供给 MQIPT 实例的名称。

   以下消息指示 MQIPT 已成功启动:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 is starting and will forward messages to :
```

```
MQCPI034 ....10.100.6.7(1416)
MQCPI035 ....using MQ protocol
MQCPI036 ....SSL Client side enabled with properties :
MQCPI139 ......secure socket protocols <NULL>
MQCPI031 ......cipher suites SSL_RSA_WITH_AES_256_CBC_SHA256
MQCPI032 ......key ring file C:\\mqipt\\samples\\ssl\\sslSample.pfx
MQCPI047 ......CA key ring file <NULL>
MQCPI071 ......site certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 ......peer certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI078 Route 1415 ready for connection requests
```

2. 在 MQIPT 2 系统上:

   a) 编辑 `mqipt.conf` 并添加以下路由定义:

```
[route]
ListenerPort=1416
Destination=Server1.company2.com
DestinationPort=1414
SSLServer=true
SSLServerAskClientAuth=true
SSLServerKeyRing=C:\\mqipt\\samples\\ssl\\sslSample.pfx
SSLServerKeyRingPW=<mqiptPW>1!PCaB1HWrFMOp43ngjwgArg==!6N/vsbqru7iqMhFN+wozxQ==
SSLServerCipherSuites=SSL_RSA_WITH_AES_256_CBC_SHA256
```

   b) 打开命令提示符，并启动 MQIPT:

```
C:
cd \mqipt\bin
mqipt .. -n ipt2
```

   其中 `..` 指示 MQIPT 配置文件 `mqipt.conf` 位于父目录中， `ipt2` 是要提供给 MQIPT 实例的名称。

   以下消息指示 MQIPT 已成功启动:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqipt\logs will be used to store the log files
MQCPI006 Route 1416 is starting and will forward messages to :
MQCPI034 ....Server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI037 ....SSL Server side enabled with properties :
MQCPI139 ......secure socket protocols <NULL>
MQCPI031 ......cipher suites SSL_RSA_WITH_AES_256_CBC_SHA256
MQCPI032 ......key ring file C:\\mqipt\\samples\\ssl\\sslSample.pfx
MQCPI047 ......CA key ring file <NULL>
MQCPI071 ......site certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 ......peer certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI033 ......client authentication set to true
MQCPI078 Route 1416 ready for connection requests
```

3. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令:

   a) 设置 **MQSERVER** 环境变量:

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

   b) 放置消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

   输入消息字符串之后，按 Enter 键两次。

   c) 获取消息:

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

   此时会返回消息“Hello world”。

认证 TLS 客户机和服务器

您可以将 MQIPT 作为 TLS 服务器和客户机运行，以终止入局 TLS 会话，并使用单独的 TLS 连接将数据转发到目标。

开始之前

在开始使用此方案之前，请确保完成 第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务，并已阅读主题 MQIPT 中的 SSL/TLS 支持。

注: 为了方便起见，此方案使用自签名证书。 请勿在生产环境中使用任何自签名证书。 而是获取由可信认证中心 (CA) 签署的证书。

关于此任务

通过 MQIPT 的单个实例在 IBM MQ 客户机与 IBM MQ 服务器之间建立连接。 IBM MQ 客户机与 MQIPT 之间以及 MQIPT 与 IBM MQ 服务器之间的连接都使用 TLS。 因此， MQIPT 路由是 TLS 服务器和 TLS 客户机。

在客户机与 MQIPT 之间的 TLS 握手期间，客户机与 MQIPT 相互发送其证书以认证连接。 在客户机与 MQIPT 之间建立连接时， MQIPT 将建立与 IBM MQ 服务器的单独 TLS 连接。 MQIPT 和 IBM MQ 服务器将其证书相互发送以认证连接。

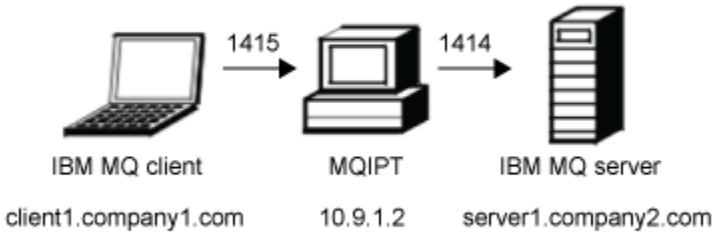


图 *33: SSL/TLS* 服务器和客户机网络图

此图显示通过单个 MQIPT 实例从 IBM MQ 客户机 (称为 client1.company1.com) 到 IBM MQ 服务器 (在端口 1414 上称为 server1.company2.com) 的连接。

过程

要将 MQIPT 路由配置为 TLS 服务器和 TLS 客户机，请完成以下步骤:

1. 在运行客户机的系统上，完成以下步骤以将 IBM MQ 客户机配置为使用 TLS 连接。
   a) 为客户机创建密钥存储库。
   输入以下命令以在 `C:\ProgramData\IBM\MQ` 目录中创建名为 `clientkey.kdb` 的新密钥存储库:

   ```
   runmqakm -keydb -create -db C:\ProgramData\IBM\MQ\clientkey.kdb -pw password -stash
   ```

   其中 *password* 是密钥存储库密码。
   有关更多信息，请参阅在 AIX, Linux, and Windows 上设置密钥存储库。
   b) 在步骤 第 153 页的『1.a』中创建的客户机密钥存储库中为客户机创建个人证书。
   输入以下命令，为客户机创建标签为 `clientcert` 的新自签名证书:

   ```
   runmqakm -cert -create -db C:\ProgramData\IBM\MQ\clientkey.kdb -stashed
            -label clientcert -dn CN=Client
   ```

   有关更多信息，请参阅在 AIX, Linux, and Windows 上创建自签名个人证书。

c) 从客户机密钥存储库中抽取证书。

输入以下命令以抽取客户机证书:

```
runmqakm -cert -extract -db C:\ProgramData\IBM\MQ\clientkey.kdb -stashed -label clientcert
         -target C:\ProgramData\IBM\MQ\client.crt -format ascii
```

有关更多信息，请参阅 从 AIX, Linux, and Windows 上的密钥存储库中抽取自签名证书的公共部分。

d) 将抽取的证书文件复制到运行 MQIPT 的系统。

2. 在运行 IBM MQ 服务器的系统上，完成以下步骤以将队列管理器配置为使用 TLS 连接。

a) 为此队列管理器创建密钥存储库。

输入以下命令为队列管理器创建名为 `key.kdb` 的新密钥存储库。

```
runmqakm -keydb -create -db C:\ProgramData\IBM\MQ\qmgrs\MQIPT!QM1\ssl\key.kdb -pw
password -stash
```

其中 *password* 是密钥存储库密码。

确保为 mqm 用户授予对 `C:\ProgramData\IBM\MQ\qmgrs\MQIPT!QM1\ssl\key.sth` 隐藏文件的读访问权，该命令将创建该文件以存储加密密钥存储库密码。

有关更多信息，请参阅在 AIX, Linux, and Windows 上设置密钥存储库。

b) 在步骤 第 154 页的『2.a』 中创建的队列管理器密钥存储库中为队列管理器创建个人证书。

输入以下命令，为队列管理器创建标签为 `ibmwebspheremqmqipt.qm1` 的新自签名证书:

```
runmqakm -cert -create -db C:\ProgramData\IBM\MQ\qmgrs\MQIPT!QM1\ssl\key.kdb -stashed
         -label ibmwebspheremqmqipt.qm1 -dn CN=MQIPT.QM1
```

有关更多信息，请参阅在 AIX, Linux, and Windows 上创建自签名个人证书。

c) 从队列管理器密钥存储库中抽取证书。

输入以下命令以抽取队列管理器证书:

```
runmqakm -cert -extract -db C:\ProgramData\IBM\MQ\qmgrs\MQIPT!QM1\ssl\key.kdb -stashed
-label ibmwebspheremqmqipt.qm1
         -target C:\ProgramData\IBM\MQ\qmgrs\MQIPT!QM1\ssl\mqipt.qm1.crt -format ascii
```

有关更多信息，请参阅 从 AIX, Linux, and Windows 上的密钥存储库中抽取自签名证书的公共部分。

d) 将抽取的证书文件复制到运行 MQIPT 的系统。

e) 发出以下 MQSC 命令以变更 MQIPT.CONN.CHANNEL 服务器连接通道以使用 TLS:

```
ALTER CHANNEL(MQIPT.CONN.CHANNEL) CHLTYPE(SVRCONN) TRPTYPE(TCP)
SSLCIPH(ANY_TLS12_OR_HIGHER)
```

3. 在运行 MQIPT 的系统上，完成以下步骤以将 MQIPT 路由配置为使用 TLS。

a) 输入以下命令为 MQIPT 创建 PKCS #12 密钥存储库:

```
mqiptKeycmd -keydb -create -db C:\mqiptHome\ssl\mqipt.p12 -pw password -type pkcs12
```

其中 *password* 是密钥存储库密码。

b) 在步骤 第 154 页的『3.a』 中创建的 MQIPT 密钥存储库中为 MQIPT 创建个人证书。

输入以下命令，以创建标签为 `mqiptcert` for MQIPT 的新自签名证书:

```
mqiptKeycmd -cert -create -db C:\mqiptHome\ssl\mqipt.p12 -pw password -type pkcs12
            -label mqiptcert -dn "CN=MQIPT Test Certificate"
```

其中 *password* 是在步骤 第 154 页的『3.a』 中创建密钥存储库时指定的密钥存储库密码。

c) 将客户机证书和队列管理器证书添加到 MQIPT 密钥存储库。

输入以下命令以将证书添加到 MQIPT 密钥存储库:

```
mqiptKeycmd -cert -add -db C:\mqiptHome\ssl\mqipt.p12 -pw password -type pkcs12 -label
clientcert
            -file client.crt -format ascii
```

```
mqiptKeycmd -cert -add -db C:\mqiptHome\ssl\mqipt.p12 -pw password -type pkcs12 -label
qm1cert
            -file mqipt.qm1.crt -format ascii
```

其中 *password* 是密钥存储库密码， *client.crt* 是您在步骤 第 154 页的『1.c』 中创建的客户机证书文件， *mqipt.qm1.crt* 是您在步骤 第 154 页的『2.c』 中创建的队列管理器证书。

d) 从密钥存储库中抽取 MQIPT 证书。

输入以下命令以抽取 MQIPT 证书:

```
mqiptKeycmd -cert -extract -db C:\mqiptHome\ssl\mqipt.p12 -pw password -type pkcs12
            -label mqiptcert -target C:\mqiptHome\ssl\mqipt.crt -format ascii
```

其中 *password* 是密钥存储库密码。

e) 将抽取的证书文件复制到运行客户机的系统和运行 IBM MQ 服务器的系统。

f) 输入以下命令以加密 MQIPT 密钥存储库密码:

```
mqiptPW
```

提示时，输入在步骤 第 154 页的『3.a』 中创建密钥存储库时指定的密钥存储库密码。

g) 编辑 `mqipt.conf` 文件并添加以下路由定义:

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SSLServer=true
SSLServerKeyRing=C:\\mqiptHome\\ssl\\mqipt.p12
SSLServerKeyRingPW=encrypted_password
SSLClient=true
SSLClientKeyRing=C:\\mqiptHome\\ssl\\mqipt.p12
SSLClientKeyRingPW=encrypted_password
```

其中 *encrypted_password* 是通过在步骤 第 155 页的『3.f』 中运行 **mqiptPW** 命令创建的加密密钥存储库密码。

4. 将 MQIPT 证书添加到客户机密钥存储库和队列管理器密钥存储库。

a) 在运行客户机的系统上，输入以下命令以将 MQIPT 证书添加到客户机密钥存储库:

```
runmqakm -cert -add -db C:\ProgramData\IBM\MQ\clientkey.kdb -stashed
         -label mqiptcert -file mqipt.crt -format ascii
```

其中 *mqipt.crt* 是您在步骤 第 155 页的『3.d』 中创建的 MQIPT 证书文件。

b) 在运行 IBM MQ 服务器的系统上，输入以下命令以将 MQIPT 证书添加到队列管理器密钥存储库:

```
runmqakm -cert -add -db C:\ProgramData\IBM\MQ\qmgrs\MQIPT!QM1\ssl\key.kdb -stashed
         -label mqiptcert -file mqipt.crt -format ascii
```

其中 *mqipt.crt* 是您在步骤 第 155 页的『3.d』 中创建的 MQIPT 证书文件。

有关更多信息，请参阅 在 AIX, Linux, and Windows 系统上将 CA 证书 (或自签名证书的公共部分) 添加到密钥存储库。

5. 在运行 MQIPT 的系统上，打开命令提示符并输入以下命令以启动 MQIPT:

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt1` 是要提供给 MQIPT 实例的名称。

以下消息指示 MQIPT 已成功启动:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 is starting and will forward messages to :
```

```
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI036 ....SSL Client side enabled with properties :
MQCPI139 ......secure socket protocols <NULL>
MQCPI031 ......cipher suites <NULL>
MQCPI032 ......key ring file C:\\mqiptHome\\ssl\\mqipt.p12
MQCPI047 ......CA key ring file <NULL>
MQCPI071 ......site certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 ......peer certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI037 ....SSL Server side enabled with properties :
MQCPI139 ......secure socket protocols <NULL>
MQCPI031 ......cipher suites <NULL>
MQCPI032 ......key ring file C:\\mqiptHome\\ssl\\mqipt.p12
MQCPI047 ......CA key ring file <NULL>
MQCPI071 ......site certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 ......peer certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI033 ......client authentication set to false
MQCPI078 Route 1415 ready for connection requests
```

6. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令以运行 TLS 样本程序:

```
AMQSSSLC -m MQIPT.QM1 -c MQIPT.CONN.CHANNEL -x 10.9.1.2(1415)
         -k "C:\ProgramData\IBM\MQ\clientkey" -l clientcert -s ANY_TLS12_OR_HIGHER
```

以下消息指示应用程序已成功连接到队列管理器:

```
Connection established to queue manager MQIPT.QM1
```

配置 HTTP 隧道

在此场景中，您可以通过 HTTP 测试两个 MQIPT 实例之间的简单连接。

开始之前

开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。

关于此任务

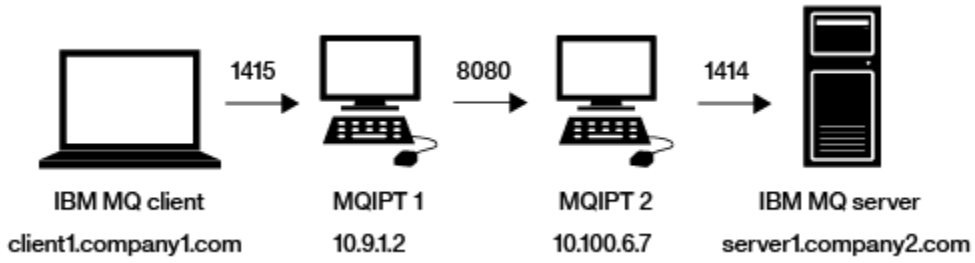


图 *34: HTTP* 隧道网络图

此图显示从 IBM MQ 客户机 (在端口 1415 上称为 client1.company1.com) 到两个 MQIPT 实例的连接，通过 HTTP 隧道连接，最后连接到 IBM MQ 服务器 (在端口 1414 上称为 server1.company2.com)。

过程

要在 MQIPT 的两个实例之间配置 HTTP 隧道，请完成以下步骤:

1. 在 MQIPT 1 系统上:
   a) 编辑 `mqipt.conf` 并添加以下路由定义:

```
[route]
ListenerPort=1415
```

```
Destination=10.100.6.7
DestinationPort=8080
HTTP=true
HTTPServer=10.100.6.7
HTTPServerPort=8080
```

b) 打开命令提示符，并启动 MQIPT：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt1` 是要提供给 MQIPT 实例的名称。

以下消息指示 MQIPT 已成功启动：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 is starting and will forward messages to :
MQCPI034 ....10.100.6.7(8080)
MQCPI035 ....using HTTP
MQCPI066 ....and HTTP server at 10.100.6.7(8080)
MQCPI078 Route 1415 ready for connection requests
```

2. 在 MQIPT 2 系统上：

a) 编辑 `mqipt.conf` 并添加以下路由定义：

```
[route]
ListenerPort=8080
Destination=Server1.company2.com
DestinationPort=1414
```

b) 打开命令提示符，并启动 MQIPT：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt2
```

其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt2` 是要提供给 MQIPT 实例的名称。

以下消息指示 MQIPT 已成功启动：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 8080 is starting and will forward messages to :
MQCPI034 ....Server1.company2.com(1414)
MQCPI035 ....using MQ protocols
MQCPI078 Route 8080 ready for connection requests
```

3. 在 IBM MQ 客户机的命令提示符中，输入以下命令：

a) 设置 **MQSERVER** 环境变量：

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

b) 放置消息：

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

输入消息字符串之后，按 Enter 键两次。

c) 获取消息：

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

此时会返回消息“Hello world”。

配置访问控制

在此场景中，您可以将 MQIPT 设置为仅接受来自特定客户机的连接，方法是使用 Java security manager 在 MQIPT 侦听器端口上添加安全性检查。

开始之前

- 开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。

关于此任务

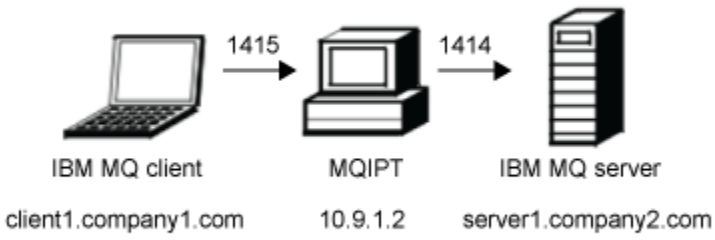


图 *35:* 访问控制网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）通过 MQIPT 连接到 IBM MQ 服务器（称为 server1.company2.com，位于端口 1414 上）。

过程

要配置访问控制，请完成以下步骤：

1. 设置 MQIPT：
   a) 通过在命令提示符处输入以下命令，将样本 Java security manager 策略复制到 MQIPT 主目录：

   ```
   copy C:\mqipt\samples\mqiptSample.policy C:\mqiptHome\mqipt.policy
   ```

   b) 使用以下命令启动策略工具实用程序：

   ```
   C:\mqipt\java\jre\bin\policytool
   ```

   c) 单击 **文件** > **打开**，然后选择 `C:\mqiptHome\mqipt.policy.`。
   d) 单击 **编辑策略条目**，然后将 `CodeBase` 从：

   ```
   file:/C:/Program Files/IBM/IBM MQ Internet Pass-Thru/lib/com.ibm.mq.ipt.jar
   ```

   to:

   ```
   file:/C:/mqipt/lib/com.ibm.mq.ipt.jar
   ```

   e) 将 `IBM MQ Internet Pass-Thru`，`errors` 和 `logs` 目录的文件许可权从：

   ```
   C:\Program Files\IBM\IBM MQ Internet Pass-Thru
   ```

   to:

   ```
   C:\mqiptHome
   ```

f) 将其他文件许可权从：

```
C:\Program Files\IBM\IBM MQ Internet Pass-Thru
```

to:

```
C:\mqipt
```

g) 单击**添加许可权**。

按以下方式填写字段：

许可权: java.net.SocketPermission
目标: client1.company1.com:1024-
操作: **接受、侦听、解析**

h) 单击 **文件> 保存** 以保存对策略文件的更改。

i) 编辑 mqipt.conf。

i) 将以下两个属性添加到 [global] 部分:

```
SecurityManager=true
SecurityManagerPolicy=C:\mqiptHome\mqipt.policy
```

ii) 添加以下路由定义:

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

2. 启动 MQIPT:

打开命令提示符并输入以下命令：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 C:\mqiptHome 指示 MQIPT 配置文件 mqipt.conf 的位置， ipt1 是要提供给 MQIPT 实例的名称。

以下消息指示 MQIPT 已成功启动:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI055 Setting the java.security.policy to C:\mqiptHome\mqipt.policy
MQCPI053 Starting the Java Security Manager
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI078 Route 1415 ready for connection requests
```

3. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令:

a) 设置 **MQSERVER** 环境变量：

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

b) 放置消息：

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

输入消息字符串之后，按 Enter 键两次。

c) 获取消息：

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

此时会返回消息“Hello world”。

配置 SOCKS 代理

在此场景中，可以将 MQIPT 用作 SOCKS 代理。

开始之前

- 开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。
- 在整个 IBM MQ 计算机上或者仅在 IBM MQ 客户机应用程序 **amqsputc** 和 **amqsgetc** 上启用 SOCKS。
- 按以下方式配置 SOCKS 客户机：
  1. 将 MQIPT 用作 SOCKS 代理。
  2. 启用 SOCKS 5 支持。
  3. 禁用用户认证。
  4. 限制与 MQIPT 网络地址的连接。

关于此任务

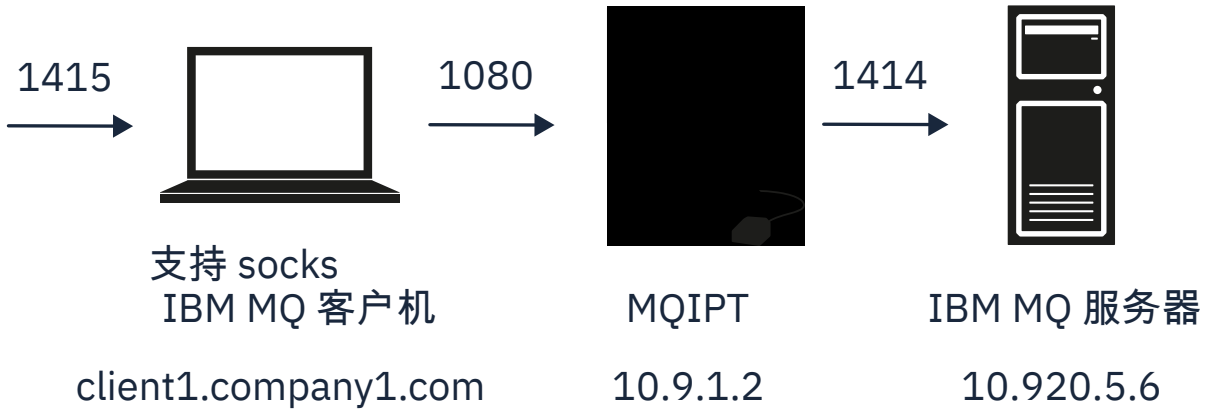


图 *36: SOCKS* 代理网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）通过 MQIPT 连接到 IBM MQ 服务器（称为 server1.company2.com，位于端口 1414 上）。

过程

要配置 SOCKS 代理，请完成以下步骤：

1. 配置并启动 MQIPT：

   a) 编辑 `mqipt.conf` 并添加以下路由定义：

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

   在 SOCKS 握手过程中，将忽略 **Destination** 和 **DestinationPort** 路由属性的值，因为将从 IBM MQ 客户机获取真正的目标。

   b) 打开命令提示符，并启动 MQIPT：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt1` 是要提供给 MQIPT 实例的名称。

以下消息指示 MQIPT 已成功启动:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1080 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI052 ....SOCKS server side enabled
MQCPI078 Route 1080 ready for connection requests
```

2. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令:

a) 设置 **MQSERVER** 环境变量:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.20.5.6(1414)
```

b) 放置消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

输入消息字符串之后，按 Enter 键两次。

c) 获取消息:

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

此时会返回消息“Hello world”。

配置 SOCKS 客户机

在此场景中，可以使用现有 SOCKS 代理来运行 MQIPT，就像它已启用 SOCKS 一样。

这与方案 第 160 页的『配置 SOCKS 代理』类似，只是 MQIPT 建立了支持 SOCKS 的连接，而不是 IBM MQ 客户机。

开始之前

在开始使用此方案之前，请确保已完成 第 142 页的『开始使用 IBM MQ Internet Pass-Thru』 中列出的先决条件任务

关于此任务

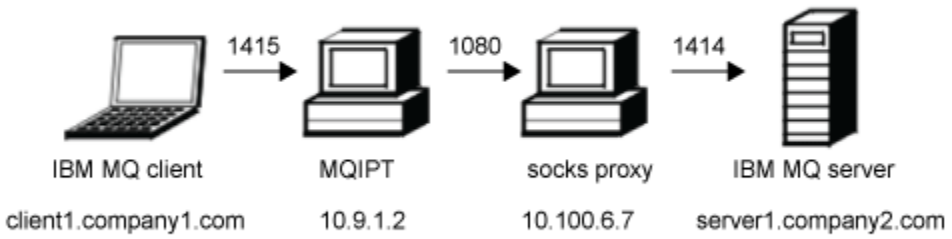


图 *37: SOCKS* 客户机网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）依次通过 MQIPT 和 SOCKS 代理（位于端口 1080 上）连接到 IBM MQ 服务器（称为 server1.company2.com，位于端口 1414 上）。

过程

要配置 SOCKS 客户机，请完成以下步骤：

1. 设置 MQIPT。

   在 MQIPT 计算机上，编辑 `mqipt.conf` 以添加以下路径定义：

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

2. 启动 MQIPT。

   打开命令提示符并输入以下命令：

   ```
   C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
   ```

   其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt1` 是要提供给 MQIPT 实例的名称。

   以下消息指示 MQIPT 已成功启动：

   ```
   5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
   MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
   MQCPI004 Reading configuration information from mqipt.conf
   MQCPI152 MQIPT name is ipt1
   MQCPI021 Password checking has been enabled on the command port
   MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
   MQCPI006 Route 1415 has started and will forward messages to :
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....using MQ protocol
   MQCPI039 ....and SOCKS proxy at 10.9.6.7(1080)
   MQCPI078 Route 1415 ready for connection requests
   ```

3. 在 IBM MQ 客户机的命令提示符中，输入以下命令：

   a) 设置 **MQSERVER** 环境变量：

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

   b) 放置消息：

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
   Hello world
   ```

   输入消息字符串之后，按 Enter 键两次。

   c) 获取消息：

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

   此时会返回消息“Hello world”。

配置 MQIPT 集群支持

在此场景中，可以设置集群环境。

开始之前

- 开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。
- 在 IBM MQ 服务器 LONDON 上：

  - 定义名为 LONDON 的队列管理器。
  - 定义名为 MQIPT.CONN.CHANNEL 的服务器连接通道。
  - 在端口 1414 上启动针对 LONDON 的 TCP/IP 侦听器。
  - 在队列管理器中启用 SOCKS。
- 在 IBM MQ 服务器 NEWYORK 上：
  - 定义名为 NEWYORK 的队列管理器。
  - 定义名为 MQIPT.CONN.CHANNEL 的服务器连接通道。
  - 在端口 1414 上启动针对 NEWYORK 的 TCP/IP 侦听器。
  - 在队列管理器中启用 SOCKS。

注: 要在队列管理器中启用 SOCKS，请在整个计算机上或仅在 IBM MQ 服务器应用程序中启用 SOCKS。 按以下方式配置 SOCKS 客户机：

- 使客户机指向作为 SOCKS 代理的 MQIPT。
- 启用 SOCKS V5 支持。
- 禁用用户认证。
- 仅建立与 MQIPT 的远程连接。

关于此任务

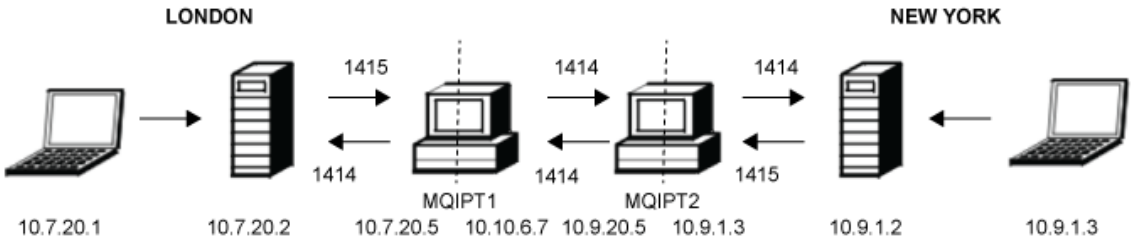


图 *38:* 集群网络图

此图显示了 IBM MQ 客户机通过 MQIPT 连接到 IBM MQ 服务器。

在同一台计算机上，只能有一个应用程序侦听单个给定端口。 如果端口 1414 已在使用中，请选择空闲的端口并将以下示例中的端口值替换为此端口。

这样，便可以通过将消息放入 LONDON 服务器上的本地队列中并从 NEWYORK 服务器检索该消息，来测试队列管理器之间的路径。

过程

要配置 MQIPT 集群支持，请完成以下步骤：

1. 设置 LONDON 服务器。

   打开命令提示符并输入以下命令：

```
runmqsc
DEFINE CHANNEL(TO.LONDON) +
       CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
       CLUSTER(INVENTORY) +
       CONNAME('10.10.6.7(1414)')
DEFINE CHANNEL(TO.NEWYORK) +
       CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
       CLUSTER(INVENTORY) +
       CONNAME('10.9.20.5(1414)')
```

2. 设置 NEWYORK 服务器。

打开命令提示符并输入以下命令:

```
runmqsc
ALTER QMGR REPOS(INVENTORY)
DEFINE QLOCAL(MQIPT.LOCAL.QUEUE) +
       CLUSTER(INVENTORY)
DEFINE CHANNEL(TO.NEWYORK) +
       CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
       CLUSTER(INVENTORY) +
       CONNAME('10.9.20.5(1414)')
DEFINE CHANNEL(TO.LONDON) +
       CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
       CLUSTER(INVENTORY) +
       CONNAME('10.10.6.7(1414)')
```

3. 设置 MQIPT 1。

   编辑 `mqipt.conf` 并添加以下路由定义:

```
[route]
Name=LONDON to NEWYORK
ListenerPort=1415
Destination=10.9.20.5
DestinationPort=1414
SocksServer=true

[route]
Name=MQIPT1 to LONDON
ListenerPort=1414
Destination=10.7.20.2
DestinationPort=1414
```

4. 启动 MQIPT 1。

   打开命令提示符并输入以下命令:

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

   其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt1` 是要提供给 MQIPT 实例的名称。

   以下消息指示 MQIPT 已成功启动:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....10.9.20.5(1414)
MQCPI035 ....using MQ protocol
MQCPI052 ....SOCKS server side enabled
MQCPI078 Route 1415 ready for connection requests
MQCPI006 Route 1414 has started and will forward messages to :
MQCPI034 ....10.7.20.2(1414)
MQCPI035 ....using MQ protocol
MQCPI078 Route 1414 ready for connection requests
```

5. 设置 MQIPT 2。

   编辑 `mqipt.conf` 并添加以下路由定义:

```
[route]
Name=NEWYORK to LONDON
ListenerPort=1415
Destination=10.10.6.7
DestinationPort=1414
SocksServer=true

[route]
Name=MQIPT2 to NEWYORK
ListenerPort=1414
Destination=10.9.1.2
DestinationPort=1414
```

6. 启动 MQIPT 2。

打开命令提示符并输入以下命令:

```
C:
cd \mqipt\bin
mqipt .. -n ipt2
```

其中 `..` 指示 MQIPT 配置文件 `mqipt.conf` 位于父目录中， `ipt2` 是要提供给 MQIPT 实例的名称。

以下消息指示 MQIPT 已成功启动:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqipt\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....10.10.6.7(1414)
MQCPI035 ....using MQ protocol
MQCPI052 ....SOCKS server side enabled
MQCPI078 Route 1415 ready for connection requests
MQCPI006 Route 1414 has started and will forward messages to :
MQCPI034 ....10.9.1.2(1414)
MQCPI035 ....using MQ protocol
MQCPI078 Route 1414 ready for connection requests
```

7. 在 LONDON IBM MQ 客户机 (10.7.20.1) 的命令提示符中，输入以下命令:
   a) 设置 **MQSERVER** 环境变量:

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.7.20.2(1414)
   ```

   b) 放置消息:

   ```
   amqsputc MQIPT.LOCAL.QUEUE LONDON
   Hello world
   ```

   输入消息字符串之后，按 Enter 键两次。

   这会使 LONDON 队列管理器将消息发送到 NEWYORK 队列管理器上的队列中。
8. 在 NEWYORK IBM MQ 客户机 (10.9.1.3) 的命令提示符中，输入以下命令:
   a) 设置 **MQSERVER** 环境变量:

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

   b) 获取消息:

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

   此时会返回消息“Hello world”。

分配端口号

可以控制在建立出站连接时使用的本地端口地址。 例如，如果防火墙仅允许特定范围内的端口号，那么可以使用 MQIPT 来确保输出源自有效的端口。

开始之前

- 开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。
- 在多宿主计算机上安装 MQIPT。

关于此任务

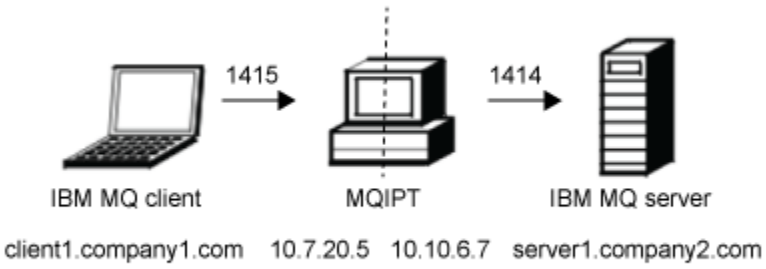


图 *39:* 端口分配网络图

此图显示从 IBM MQ 客户机 (端口 1415 上的 client1.company1.com) 到 MQIPT 到 IBM MQ 服务器 (端口 1414 上的 server1.company2.com) 的连接。

过程

要分配端口号，请完成以下步骤:

1. 设置 MQIPT。

   编辑 `mqipt.conf` 并添加以下路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   LocalAddress=10.10.6.7
   OutgoingPort=2000
   MaxConnectionThreads=20
   ```

2. 启动 MQIPT。

   在 IBM MQ 系统上打开命令提示符，然后输入以下命令:

   ```
   C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
   ```

   其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt1` 是要提供给 MQIPT 实例的名称。

   以下消息指示 MQIPT 已成功启动:

   ```
   5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
   MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
   MQCPI004 Reading configuration information from mqipt.conf
   MQCPI152 MQIPT name is ipt1
   MQCPI021 Password checking has been enabled on the command port
   MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
   MQCPI006 Route 1415 is starting and will forward messages to :
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....using MQ protocol
   MQCPI069 ....binding to local address 10.10.6.7 when making new connections
   MQCPI070 ....using local port address range 2000-2019 when making new connections
   MQCPI078 Route 1415 ready for connection requests
   ```

3. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令:

   a) 设置 **MQSERVER** 环境变量:

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.7.20.5(1415)
   ```

   b) 放置消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

输入消息字符串之后，按 Enter 键两次。

c) 获取消息：

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

此时会返回消息“Hello world”。

使用 LDAP 服务器检索 CRL

可以将 MQIPT 配置为使用 LDAP 服务器检索证书撤销列表 (CRL)。

开始之前

- 开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。
- 确保 MQIPT 2 具有由可信认证中心 (CA) 颁发的个人证书，该证书存储在名为 `myCert.pfx` 的密钥环文件中。
- 确保 MQIPT 1 具有将用于认证 MQIPT 2 发送的证书的可信 CA 证书的副本。 此证书存储在名为 `caCerts.pfx` 的密钥环文件中。
- 用于访问密钥环的密码已使用 **mqiptPW** 命令加密。

关于此任务

在此场景中，您可以将 IBM MQ 客户机连接到队列管理器 (QM)，并将 IBM MQ 消息放入目标队列中。 在 MQIPT 1 上运行 MQIPT 跟踪将显示正在使用的 LDAP 服务器。

要演示 CRL 的工作方式，请确保可信 CA 撤销了 MQIPT 2 所使用的个人证书。 然后，不允许 IBM MQ 客户机连接到 QM，因为将拒绝从 MQIPT 1 到 MQIPT 2 的连接。

本方案无意说明如何安装和设置 LDAP 服务器，也无意说明如何创建包含个人证书或可信证书的密钥环文件。 它假设 LDAP 服务器是由已知的可信认证中心提供。 将不使用 LDAP 备份服务器，但可以通过添加相应的路径属性来实现此功能。

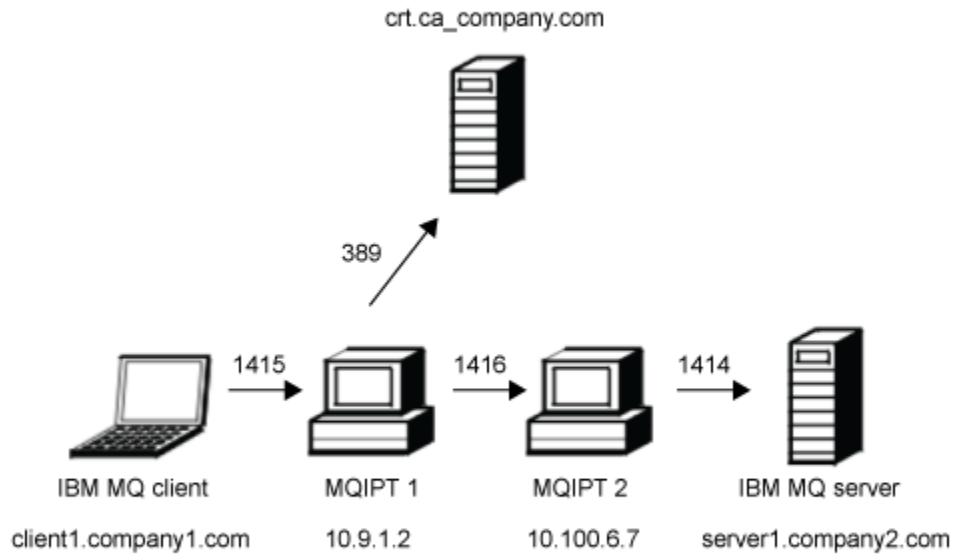


图 *40: LDAP* 服务器网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）通过 MQIPT 的两个实例来连接到 IBM MQ 服务器（称为 server1.company2.com，位于端口 1414 上）。 第一个 MQIPT 实例具有与 LDAP 服务器（称为 crl.ca_company.com，位于端口 389 上）的连接。

过程

要使用 LDAP 服务器检索 CRL，请完成以下步骤：

1. 在 MQIPT 1 系统上：

   a) 编辑 `mqipt.conf` 并添加以下路由定义：

   ```
   [route]
   ListenerPort=1415
   Destination=10.100.6.7
   DestinationPort=1416
   SSLClient=true
   SSLClientCAKeyRing=C:\mqiptHome\ssl\caCerts.pfx
   SSLClientCAKeyRingPW=encrypted_key_ring_password
   LDAP=true
   LDAPServer1=crl.ca_company.com
   LDAPServer1Timeout=4
   ```

   其中， *encrypted_key_ring_password* 是使用 **mqiptPW** 命令加密的 `caCerts.pfx` 密钥环的密码。

   b) 打开命令提示符，并启动 MQIPT：

   ```
   C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
   ```

   其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt1` 是要提供给 MQIPT 实例的名称。

   以下消息指示 MQIPT 已成功启动：

   ```
   5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
   MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
   MQCPI004 Reading configuration information from mqipt.conf
   ```

```
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....10.100.6.7(1416)
MQCPI035 ....using MQ protocol
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......CipherSuites <NULL>
MQCPI032 ......key ring file <NULL>
MQCPI047 ......CA key ring file C:\mqiptHome\ssl\caCerts.pfx
MQCPI071 ......site certificate uses UID=*,CN=*,T=*,OU=*,DC=*,O=*,
                                                STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 ......peer certificate uses UID=*,CN=*,T=*,OU=*,DC=*,O=*,
                                                STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI075 ....LDAP main server at crl.ca_company.com(389)
MQCPI086 ......timeout of 4 second(s)
MQCPI084 ....CRL cache expiry timeout is 1 hour(s)
MQCPI085 ....CRLs will be saved in the key-ring file(s)
MQCPI078 Route 1415 ready for connection requests
```

2. 在 MQIPT 2 系统上:

   a) 编辑 `mqipt.conf` 并添加以下路由定义:

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

   其中， *encrypted_key_ring_password* 是使用 **mqiptPW** 命令加密的 `myCert.pfx` 密钥环的密码。

   b) 打开命令提示符，并启动 MQIPT:

   ```
   C:
   cd \mqipt\bin
   mqipt .. -n ipt2
   ```

   其中 `..` 指示 MQIPT 配置文件 `mqipt.conf` 位于父目录中， `ipt2` 是要提供给 MQIPT 实例的名称。

   以下消息指示成功完成:

   ```
   5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
   MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
   MQCPI004 Reading configuration information from mqipt.conf
   MQCPI152 MQIPT name is ipt2
   MQCPI021 Password checking has been enabled on the command port
   MQCPI011 The path C:\mqipt\logs will be used to store the log files
   MQCPI006 Route 1416 is starting and will forward messages to :
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....using MQ protocol
   MQCPI037 ....SSL Server side enabled with properties :
   MQCPI031 ......CipherSuites <NULL>
   MQCPI032 ......key ring file C:\mqipt\ssl\myCert.pfx
   MQCPI047 ......CA key ring file <NULL>
   MQCPI071 ......site certificate uses UID=*,CN=*,T=*,OU=*,DC=*,O=*,
                                                   STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
   MQCPI038 ......peer certificate uses UID=*,CN=*,T=*,OU=*,DC=*,O=*,
                                                   STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
   MQCPI033 ......client authentication set to false
   MQCPI078 Route 1416 ready for connection requests
   ```

3. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令:

   a) 设置 **MQSERVER** 环境变量:

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

   b) 放置消息:

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
   Hello world
   ```

   输入消息字符串之后，按 Enter 键两次。

c) 获取消息：

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

此时会返回消息“Hello world”。

以 TLS 代理方式运行 MQIPT

您可以在 TLS 代理方式下运行 MQIPT，以便它接受从 IBM MQ TLS 客户机到 IBM MQ TLS 服务器的 TLS 连接请求。

开始之前

开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。

关于此任务

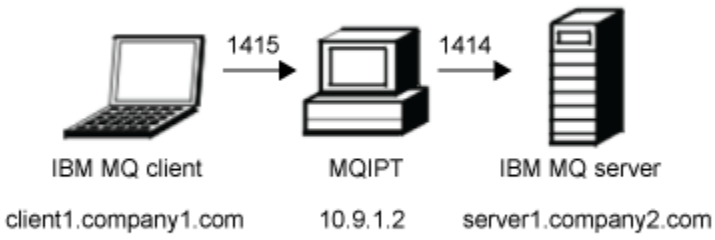


图 *41: SSL/TLS* 代理方式网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）通过 MQIPT 连接到 IBM MQ 服务器（称为 server1.company2.com，位于端口 1414 上）。

有关为 IBM MQ 配置 TLS 的更多信息，请参阅 使用 SSL/TLS。

过程

要以 TLS 代理方式运行 MQIPT，请完成以下步骤:

1. 配置 IBM MQ 客户机和服务器以使用 TLS 连接。
   a) 为此队列管理器创建密钥存储库。
   有关更多信息，请参阅在 AIX, Linux, and Windows 上设置密钥存储库。
   b) 在 `C:\ProgramData\IBM\MQ` 目录中为客户机创建密钥存储库。 将其命名为 *clientkey.kdb*。
   c) 在步骤 第 170 页的『1.a』中创建的队列管理器密钥存储库中为队列管理器创建个人证书。
   有关更多信息，请参阅在 AIX, Linux, and Windows 上创建自签名个人证书。
   d) 在步骤 第 170 页的『1.b』中创建的客户机密钥存储库中为客户机创建个人证书。
   e) 从服务器密钥存储库中抽取个人证书，并将其添加到客户机存储库。
   有关更多信息，请参阅抽取 AIX, Linux, and Windows 上密钥库中自签名证书的公共部分和将 CA 证书（或自签名证书的公共部分）添加到 AIX, Linux, and Windows 系统上的密钥库中。
   f) 从客户机密钥存储库中抽取个人证书，并将其添加到服务器密钥存储库。
   g) 变更 MQIPT.CONN.CHANNEL 服务器连接通道使用 MQSC 命令来使用 TLS:

```
ALTER CHANNEL(MQIPT.CONN.CHANNEL) CHLTYPE(SVRCONN) TRPTYPE(TCP)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

2. 要以 TLS 代理方式运行 MQIPT，请完成以下步骤:

   a) 编辑 `mqipt.conf` 并添加以下路由定义:

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

   b) 启动 MQIPT。

   打开命令提示符并输入以下命令:

   ```
   C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
   ```

   其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt1` 是要提供给 MQIPT 实例的名称。

   以下消息指示 MQIPT 已成功启动:

   ```
   5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
   MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
   MQCPI004 Reading configuration information from mqipt.conf
   MQCPI152 MQIPT name is ipt1
   MQCPI021 Password checking has been enabled on the command port
   MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
   MQCPI006 Route 1415 has started and will forward messages to :
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....using SSLProxyMode protocol
   MQCPI078 Route 1415 ready for connection requests
   ```

3. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令以运行 TLS 样本程序:

   ```
   AMQSSSLC -m MQIPT.QM1 -c MQIPT.CONN.CHANNEL -x 10.9.1.2(1415)
            -k "C:\ProgramData\IBM\MQ\clientkey" -l cert_label -s
   TLS_RSA_WITH_AES_128_CBC_SHA256
   ```

   其中， `cert_label` 是您在步骤 第 170 页的『1.d』中创建的客户机证书的标签。

使用安全管理器以 TLS 代理方式运行 MQIPT

您可以在 TLS 代理方式下运行 MQIPT，以便它接受从 IBM MQ TLS 客户机到 IBM MQ TLS 服务器的 TLS 连接请求。 通过将安全管理器与 MQIPT 配合使用，可以限制可接收消息的地址。

开始之前

注: Deprecated 不推荐将 Java security manager 与 MQIPT 配合使用，因为在 Java 的未来发行版中已不推荐使用 Java security manager 以将其除去。

开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。

关于此任务

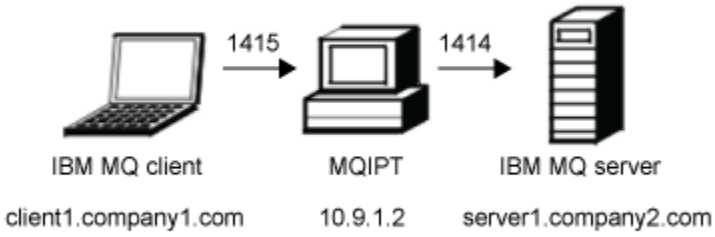


图 *42: SSL/TLS* 代理方式网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）通过 MQIPT 连接到 IBM MQ 服务器（称为 server1.company2.com，位于端口 1414 上）。

有关为 IBM MQ 配置 TLS 的更多信息，请参阅 使用 SSL/TLS。

过程

要使用安全管理器以 TLS 代理方式运行 MQIPT ，请完成以下步骤:

1. 配置 IBM MQ 客户机和服务器以使用 TLS 连接。
   a) 为此队列管理器创建密钥存储库。
   有关更多信息，请参阅在 AIX, Linux, and Windows 上设置密钥存储库。
   b) 在 `C:\ProgramData\IBM\MQ` 目录中为客户机创建密钥存储库。 将其命名为 *clientkey.kdb*。
   c) 在步骤 第 172 页的『1.a』中创建的队列管理器密钥存储库中为队列管理器创建个人证书。
   有关更多信息，请参阅在 AIX, Linux, and Windows 上创建自签名个人证书。
   d) 在步骤 第 172 页的『1.b』中创建的客户机密钥存储库中为客户机创建个人证书。
   e) 从服务器密钥存储库中抽取个人证书，并将其添加到客户机存储库。
   有关更多信息，请参阅抽取 AIX, Linux, and Windows 上密钥库中自签名证书的公共部分和将 CA 证书（或自签名证书的公共部分）添加到 AIX, Linux, and Windows 系统上的密钥库中。
   f) 从客户机密钥存储库中抽取个人证书，并将其添加到服务器密钥存储库。
   g) 变更 MQIPT.CONN.CHANNEL 服务器连接通道使用 MQSC 命令来使用 TLS:

   ```
   ALTER CHANNEL(MQIPT.CONN.CHANNEL) CHLTYPE(SVRCONN) TRPTYPE(TCP)
   SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
   ```

2. 在 MQIPT 计算机上 (请参阅图) ，通过在命令提示符处输入以下命令，将样本 Java security manager 策略复制到 MQIPT 主目录:

   ```
   copy C:\mqipt\samples\mqiptSample.policy C:\mqiptHome\mqipt.policy
   ```

3. 使用以下命令启动策略工具实用程序:

   ```
   C:\mqipt\java\jre\bin\policytool
   ```

   在策略工具中:
   a) 单击 **文件** > **打开** ，然后选择 `C:\mqiptHome\mqipt.policy.`。
   b) 选择:

   ```
   file:/C:/Program Files/IBM/IBM MQ Internet Pass-Thru/lib/com.ibm.mq.ipt.jar
   ```

   然后单击**编辑策略条目**

c) 将 CodeBase 从:

```
file:/C:/Program Files/IBM/IBM MQ Internet Pass-Thru/lib/com.ibm.mq.ipt.jar
```

to:

```
file:/C:/mqipt/lib/com.ibm.mq.ipt.jar
```

d) 将 IBM MQ Internet Pass-Thru， errors 和 logs 目录的文件许可权从:

```
C:\Program Files\IBM\IBM MQ Internet Pass-Thru
```

to:

```
C:\mqiptHome
```

e) 将其他文件许可权从:

```
C:\Program Files\IBM\IBM MQ Internet Pass-Thru
```

to:

```
C:\mqipt
```

f) 单击**添加许可权**。
按以下方式填写字段:

许可权: java.net.SocketPermission
目标: client1.company1.com:1024-
操作: **接受、侦听、解析**

g) 单击 **文件> 保存** 以保存对策略文件的更改。

4. 编辑 mqipt.conf。 将以下属性添加到 [global] 部分，并添加以下路由定义:

```
[global]
SecurityManager=true
SecurityManagerPolicy=C:\mqiptHome\mqipt.policy

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

5. 启动 MQIPT。
打开命令提示符并输入以下命令:

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 C:\mqiptHome 指示 MQIPT 配置文件 mqipt.conf 的位置， ipt1 是要提供给 MQIPT 实例的名称。

以下消息指示 MQIPT 已成功启动:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI055 Setting the java.security.policy to C:\mqiptHome\mqipt.policy
MQCPI053 Starting the Java Security Manager
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\mqipt\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using SSLProxyMode protocol
MQCPI078 Route 1415 ready for connection requests
```

6. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令以运行 TLS 样本程序:

```
AMQSSSLC -m MQIPT.QM1 -c MQIPT.CONN.CHANNEL -x 10.9.1.2(1415)
         -k "C:\ProgramData\IBM\MQ\clientkey" -l cert_label -s
TLS_RSA_WITH_AES_128_CBC_SHA256
```

其中， `cert_label` 是您在步骤 第 172 页的『1.d』中创建的客户机证书的标签。

使用安全出口

在此场景中，可以使用所提供的样本安全出口 SampleSecurityExit，从而仅允许其所用通道名称以字符串 `MQIPT.` 开头的客户机连接。

开始之前

- 开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。
- 安装 JavaJava 8.0 JDK。
- 将 Java `bin` 子目录添加到 **PATH** 环境变量中。

关于此任务

此场景中使用的样本出口为 `SampleSecurityExit.java`。 MQIPT 随附于 MQIPT 安装目录的 `samples/exits` 子目录中。

如果使用建议的服务器连接通道名称 `MQIPT.CONN.CHANNEL` (在其中大多数场景中使用)，那么将允许客户机连接完成，并且可以将 IBM MQ 消息放入队列中。

要证明安全出口正在按预期工作，请使用不以字符 `MQIPT.` (例如， `TEST.CONN.CHANNEL`) 开头的任何名称定义另一个服务器连接通道，然后重试 **amqsputc** 命令，但已更改 **MQSERVER** 环境变量以使用新通道名称。 此时将拒绝连接，并且将返回 2059 (MQRC_Q_MGR_NOT_AVAILABLE) 错误。

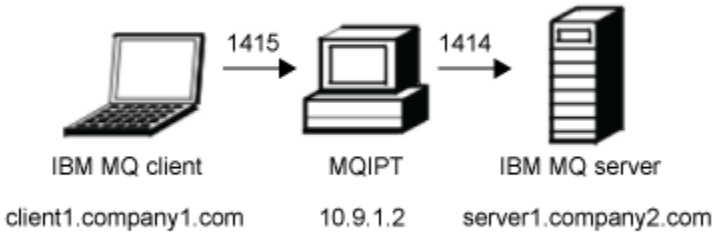


图 *43:* 安全出口网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）通过 MQIPT 连接到 IBM MQ 服务器（称为 server1.company2.com，位于端口 1414 上）。

过程

要使用安全出口，请完成以下步骤：

1. 在 MQIPT 计算机上：
   a) 通过在命令提示符中发出以下命令，在 MQIPT 主目录中创建名为 `exits` 的目录：

   ```
   md C:\mqiptHome\exits
   ```

   b) 输入以下命令以编译出口。 如果未更改出口代码，那么不必执行此操作，因为 MQIPT 随附了已编译的样本出口。

```
C:
cd \mqipt\samples\exits
javac -classpath C:\mqipt\lib\com.ibm.mq.ipt.jar;. SampleSecurityExit.java
```

c) 输入以下命令以将编译的出口类文件 SampleSecurityExit.class 复制到 C:\mqiptHome\exits 目录:

```
copy C:\mqipt\samples\exits\SampleSecurityExit.class C:\mqiptHome\exits
```

d) 编辑 mqipt.conf 以添加以下路径定义:

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

e) 打开命令提示符，并启动 MQIPT:

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 C:\mqiptHome 指示 MQIPT 配置文件 mqipt.conf 的位置， ipt1 是要提供给 MQIPT 实例的名称。

以下消息指示 MQIPT 已成功启动:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI079 ....using security exit C:\mqiptHome\exits\SampleSecurityExit
MQCPI080 ......and timeout of 30 seconds
MQCPI078 Route 1415 ready for connection requests
```

2. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令:

a) 设置 **MQSERVER** 环境变量:

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

b) 放置消息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

输入消息字符串之后，按 Enter 键两次。

c) 获取消息:

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

此时会返回消息“Hello world”。

使用安全出口将客户机连接请求路由到 IBM MQ 队列管理器服务器

在此场景中，可以使用循环法方式动态地将客户机连接请求路由到一组三个 IBM MQ 队列管理器服务器。 该组中每个服务器上的队列管理器必须完全相同。

开始之前

- 开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。
- 安装 Java 8.0 JDK。

- 将 Java bin 子目录添加到 **PATH** 环境变量中。

关于此任务

此场景中使用的样本出口为 SampleRoutingExit.java。 MQIPT 随附于 MQIPT 安装目录的 samples/exits 子目录中。

已编译出口类文件的名称和位置是使用 MQIPT **SecurityExitName** 和 **SecurityExitPath** 属性定义的。

将从名为 SampleRoutingExit.conf 的配置文件中读取要使用的队列管理器和服务器名称的列表。 The exit expects the configuration file to exist in the exits subdirectory of the MQIPT home directory.

首次运行 **amqsputc** 命令时，会在第一个服务器上的 MQIPT.LOCAL.QUEUE 队列中放置 IBM MQ 消息。 第二次运行该命令时，会在第二个服务器上的该队列中放置该消息，以此类推。 如果使用此设置，那么 **amqsgetc** 命令无法检索刚放到队列中的消息，因为 **amqsgetc** 命令所使用的客户机连接请求已传递到列表中的下一个队列。 但是，运行 **amqsputc** 命令三次，后跟三个 **amqsgetc** 命令，可确保以相同的顺序检索每条消息。

当然，通过使用另一个 IBM MQ 客户机，直接连接到队列管理器 (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。

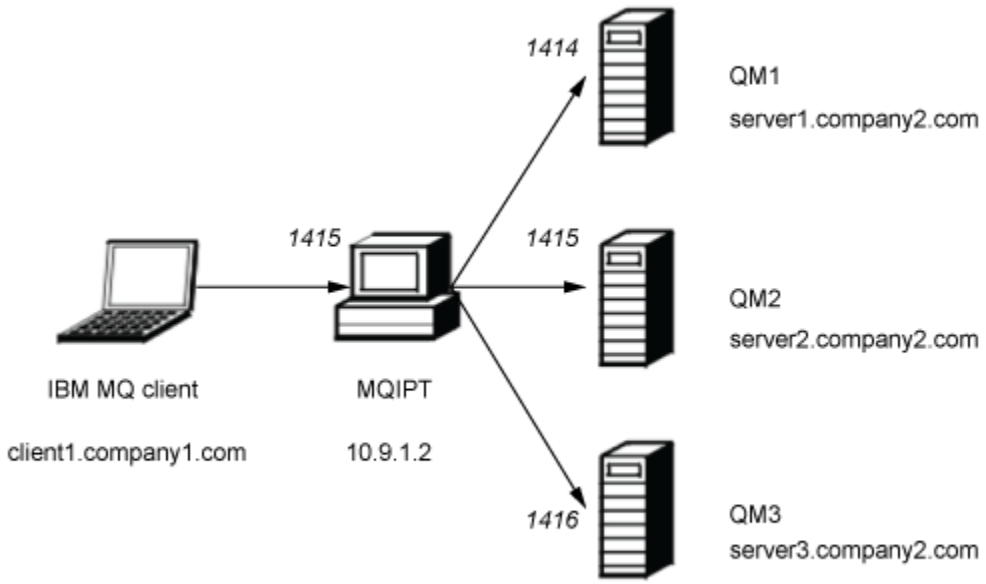


图 *44:* 路由安全出口网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）通过 MQIPT 连接到三个 IBM MQ 服务器（称为 server1.company2.com、server2.company2.com 和 server3.company2.com）。

过程

要随后使用安全出口将客户机连接请求路由到三个不同的 IBM MQ 队列管理器服务器，请完成以下步骤：

1. 在三个独立服务器上创建三个相同的队列管理器，名为 MQIPT.QM1。
   每个队列管理器都具有名为 MQIPT.CONN.CHANNEL 的 SVRCONN 通道以及名为 MQIPT.LOCAL.QUEUE 的空本地队列。
2. 在 MQIPT 服务器上：

a) 通过在命令提示符中发出以下命令，在 MQIPT 主目录中创建名为 exits 的目录:

```
md C:\mqiptHome\exits
```

b) 在 C:\mqiptHome\exits 目录 (其中 C:\mqiptHome 是 mqipt.conf 文件所在的目录) 中，创建名为 SampleRoutingExit.conf 的样本配置文件，其中包含三个队列管理器的名称。

例如，配置文件可能包含以下条目:

```
server1.company2.com:1414
server2.company2.com:1415
server3.company2.com:1416
```

确保该文件中的第一个条目之前没有空白行，并确保每个条目都是有效的服务器名称。 如果使用了不同的服务器名称，请根据您的环境来更改这些名称。

c) 打开命令提示符并输入以下命令以编译出口。 如果未更改出口代码，那么不必执行此操作，因为 MQIPT 随附了已编译的样本出口。

```
C:
cd \mqipt\samples\exits
javac -classpath C:\mqipt\lib\com.ibm.mq.ipt.jar;. SampleRoutingExit.java
```

d) 输入以下命令以将编译的出口类文件 SampleRoutingExit.class 复制到 C:\mqiptHome\exits 目录:

```
copy C:\mqipt\samples\exits\SampleRoutingExit.class C:\mqiptHome\exits
```

e) 编辑 mqipt.conf 以添加以下路径定义:

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SecurityExit=true
SecurityExitPath=C:\mqiptHome\exits
SecurityExitName=SampleRoutingExit
```

请注意，如果将 SampleRoutingExit.conf 放在缺省 C:\mqiptHome\exits 目录中，那么不必设置 **SecurityExitPath** 。

f) 启动 MQIPT。

打开命令提示符并输入以下命令:

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 C:\mqiptHome 指示 MQIPT 配置文件 mqipt.conf 的位置， ipt1 是要提供给 MQIPT 实例的名称。

以下消息指示 MQIPT 已成功启动:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI079 ....using security exit C:\mqiptHome\exits\SampleRoutingExit
MQCPI080 ......and timeout of 30 seconds
MQCPI078 Route 1415 ready for connection requests
```

3. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令:

a) 设置 **MQSERVER** 环境变量:

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

b) 放置三条消息:

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

输入每个消息字符串之后，按 Enter 键两次。

c) 获取这些消息：

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

将返回消息 `Hello world 1`， `Hello world 2` 和 `Hello world 3` 。

动态路由客户机连接请求

在此场景中，可以基于所用通道的名称，动态地将客户机连接请求路由到目标服务器。

开始之前

- 开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。
- 安装 Java 8.0 JDK。
- 将 Java `bin` 子目录添加到 **PATH** 环境变量中。

关于此任务

如果使用队列管理器的名称作为通道名称的第一部分，那么只需要使用一个 MQIPT 路由来处理所有连接请求。 例如，要连接到 QM1，SVRCONN 通道的名称可以为 QM1.MQIPT.CHANNEL。

此场景中使用的样本出口为 `SampleOneRouteExit.java`。 MQIPT 随附于 MQIPT 安装目录的 `samples/exits` 子目录中。

已编译出口类文件的名称和位置是使用 MQIPT **SecurityExitName** 和 **SecurityExitPath** 属性定义的。

将从名为 `SampleOneRouteExit.conf` 的配置文件中读取要使用的队列管理器和服务器名称的列表。 出口期望配置文件与出口类文件存在于同一目录中。

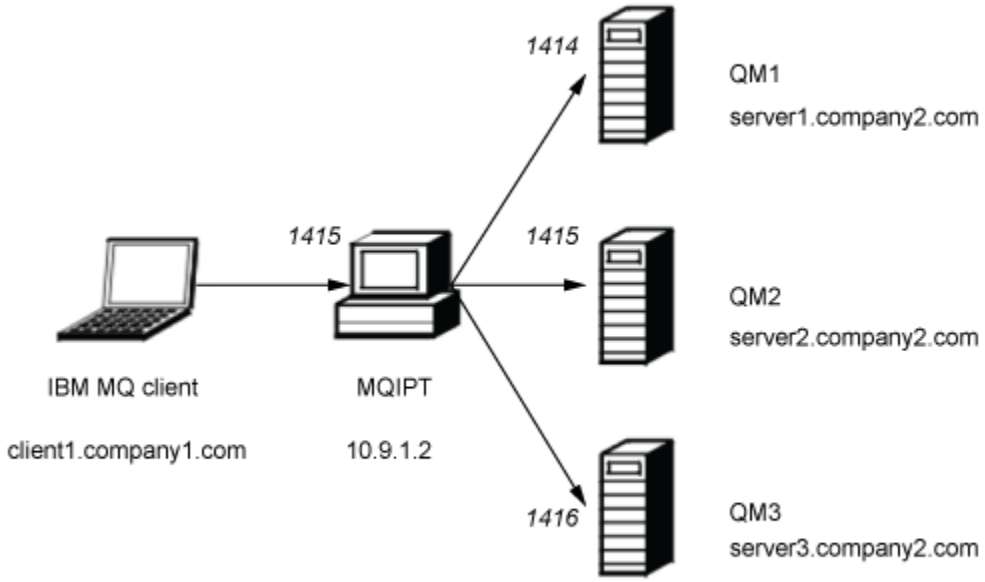


图 *45:* 动态选择一个路由出口网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）通过 MQIPT 连接到三个 IBM MQ 服务器（称为 server1.company2.com、server2.company2.com 和 server3.company2.com）。

过程

要动态地路由客户机连接请求，请完成以下步骤：

1. 在三个独立服务器上创建三个不同的队列管理器。

   每个队列管理器都具有按自身命名的 SVRCONN 通道，例如，队列管理器 QM1 上的 QM1.MQIPT.CHANNEL 以及名为 MQIPT.LOCAL.QUEUE 的空本地队列。
2. 在 MQIPT 服务器上：

   a) 通过在命令提示符中发出以下命令，在 MQIPT 主目录中创建名为 `exits` 的目录：

   ```
   md C:\mqiptHome\exits
   ```

   b) 在 `C:\mqiptHome\exits` 目录 (其中 `C:\mqiptHome` 是 `mqipt.conf` 文件所在的目录) 中，创建名为 `SampleOneRouteExit.conf` 的样本配置文件，其中包含三个队列管理器的名称。

   例如，配置文件可能包含以下条目：

   ```
   QM1 server1.company2.com:1414
   QM2 server2.company2.com:1415
   QM3 server3.company2.com:1416
   ```

   确保该文件中的第一个条目之前没有空白行，并确保每个条目都是有效的服务器名称。 如果使用了不同的服务器名称，请根据您的环境来更改这些名称。

   请注意，列表中的所有队列管理器名称都必须唯一。 如果多次列出同一名称，即使队列管理器位于不同的服务器上，也只注册该名称的最后一个条目。

   c) 打开命令提示符并输入以下命令以编译出口。 如果未更改出口代码，那么不必执行此操作，因为 MQIPT 随附了已编译的样本出口。

```
C:
cd \mqipt\samples\exits
javac -classpath C:\mqipt\lib\com.ibm.mq.ipt.jar;. SampleOneRouteExit.java
```

d) 输入以下命令以将编译的出口类文件 SampleOneRouteExit.class 复制到 C:\mqiptHome\exits 目录:

```
copy C:\mqipt\samples\exits\SampleOneRouteExit.class C:\mqiptHome\exits
```

e) 编辑 mqipt.conf 并添加以下路由定义:

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

f) 打开命令提示符，并启动 MQIPT:

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt2
```

其中 C:\mqiptHome 指示 MQIPT 配置文件 mqipt.conf 的位置， ipt2 是要提供给 MQIPT 实例的名称。

以下消息指示 MQIPT 已成功启动:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI079 ....using security exit C:\mqiptHome\exits\SampleOneRouteExit
MQCPI080 ......and timeout of 5 seconds
MQCPI078 Route 1415 ready for connection requests
```

3. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令:

a) 设置 **MQSERVER** 环境变量:

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

b) 放置消息:

```
amqsputc MQIPT.LOCAL.QUEUE QM1
Hello world 1
```

输入消息字符串之后，按 Enter 键两次。

消息由 MQIPT 路由到 QM1 ，因为 SVRCONN 通道名称以 QM1 开头。

c) 从 QM1: 获取消息

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

此时会返回消息 Hello world 1。

d) 重置 **MQSERVER** 环境变量:

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

e) 放置消息:

```
amqsputc MQIPT.LOCAL.QUEUE QM2
Hello world 2
```

输入消息字符串之后，按 Enter 键两次。

消息由 MQIPT 路由到 QM2 ，因为 SVRCONN 通道名称以 QM2 开头。

f) 从 QM2: 获取消息

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

此时会返回消息 `Hello world 2`。

g) 再次重置 **MQSERVER** 环境变量：

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

h) 放置消息：

```
amqsputc MQIPT.LOCAL.QUEUE QM3
Hello world 3
```

输入消息字符串之后，按 Enter 键两次。

消息由 MQIPT 路由到 QM3 ，因为 SVRCONN 通道名称以 QM3 开头。

i) 从 QM3: 获取消息

```
amqsgetc MQIPT.LOCAL.QUEUE QM3
```

此时会返回消息 `Hello world 3`。

使用证书出口来认证 TLS 服务器

在此场景中，您可以使用证书出口来认证 TLS 连接。

开始之前

- 开始使用此场景之前，请确保已完成第 142 页的『开始使用 IBM MQ Internet Pass-Thru』中列出的先决条件任务。
- 安装 Java 8.0 JDK。
- 将 Java `bin` 子目录添加到 **PATH** 环境变量中。

关于此任务

此方案执行与 第 148 页的『认证 TLS 服务器』 方案相同的功能，并添加了证书出口。

此场景中使用的样本出口为 `SampleCertificateExit.java`。 MQIPT 随附于 MQIPT 安装目录的 `samples/exits` 子目录中。

通过更改 **SSLExitData** 属性的值，可以允许或拒绝两个 MQIPT 服务器之间的 TLS 连接。

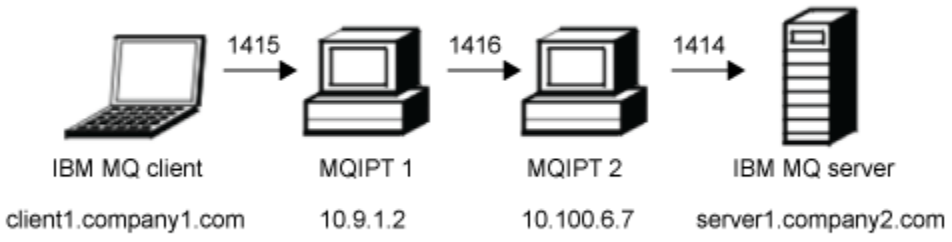


图 *46: SSL/TLS* 服务器网络图

此图显示了 IBM MQ 客户机（称为 client1.company1.com，位于端口 1415 上）通过 MQIPT 的两个实例来连接到 IBM MQ 服务器（称为 server1.company2.com，位于端口 1414 上）。

过程

要使用证书出口来认证 TLS 服务器，请完成以下步骤:

1. 在 MQIPT 1 系统上:
   a) 通过在命令提示符中发出以下命令，在 MQIPT 主目录中创建名为 `exits` 的目录:

   ```
   md C:\mqiptHome\exits
   ```

   b) 打开命令提示符并输入以下命令以编译出口。 如果未更改出口代码，那么不必执行此操作，因为 MQIPT 随附了已编译的样本出口。

   ```
   C:
   cd \mqipt\samples\exits
   javac -classpath C:\mqipt\lib\com.ibm.mq.ipt.jar;. SampleCertificateExit.java
   ```

   c) 输入以下命令以将编译的出口类文件 `SampleCertificateExit.class` 复制到 `C:\mqiptHome\exits` 目录:

   ```
   copy C:\mqipt\samples\exits\SampleCertificateExit.class C:\mqiptHome\exits
   ```

   d) 编辑 `mqipt.conf` 并添加以下路由定义:

   ```
   [route]
   ListenerPort=1415
   Destination=9.100.6.7
   DestinationPort=1416
   SSLClient=true
   SSLClientKeyRing=C:\mqipt\samples\ssl\sslSample.pfx
   SSLClientKeyRingPW=<mqiptPW>1!PCaB1HWrFMOp43ngjwgArg==!6N/vsbqru7iqMhFN+wozxQ==
   SSLClientExit=true
   SSLExitName=SampleCertificateExit
   SSLExitPath=C:\mqiptHome\exits
   SSLExitData=allow
   ```

   e) 打开命令提示符，并启动 MQIPT:

   ```
   C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
   ```

   其中 `C:\mqiptHome` 指示 MQIPT 配置文件 `mqipt.conf` 的位置， `ipt1` 是要提供给 MQIPT 实例的名称。

   以下消息指示 MQIPT 已成功启动:

   ```
   5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
   MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
   MQCPI004 Reading configuration information from mqipt.conf
   MQCPI152 MQIPT name is ipt1
   MQCPI021 Password checking has been enabled on the command port
   MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
   MQCPI006 Route 1415 has started and will forward messages to :
   MQCPI034 ....9.100.6.7(1416)
   MQCPI035 ....using MQ protocol
   MQCPI036 ....SSL Client side enabled with properties :
   MQCPI031 ......CipherSuites <null>
   MQCPI032 ......keyring file C:\mqipt\samples\ssl\sslSample.pfx
   MQCPI047 ......CA keyring file <null>
   MQCPI038 ......peer certificate uses UID=*,CN=*,T=*,OU=*,DC=*,O=*,
                                                 STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
   MQCPI129 ......using certificate exit C:\mqiptHome\exits\SampleCertificateExit
   MQCPI131 ......and certificate exit data 'allow'
   MQCPI078 Route 1415 ready for connection requests
   ```

2. 在 MQIPT 2 系统上:
   a) 编辑 `mqipt.conf` 并添加以下路由定义:

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

```
SSLServerKeyRing=C:\mqipt\samples\ssl\sslSample.pfx
SSLServerKeyRingPW=C:\mqipt\samples\ssl\sslSample.pwd
```

b) 打开命令提示符，并启动 MQIPT：

```
C:
cd \mqipt\bin
mqipt .. -n ipt2
```

其中 `..` 指示 MQIPT 配置文件 `mqipt.conf` 位于父目录中， `ipt2` 是要提供给 MQIPT 实例的名称。

以下消息指示 MQIPT 已成功启动：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2025. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqipt\logs will be used to store the log files
MQCPI006 Route 1416 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI037 ....SSL Server side enabled with properties :
MQCPI031 ......CipherSuites <null>
MQCPI032 ......key ring file C:\mqipt\samples\ssl\sslSample.pfx
MQCPI047 ......CA key ring file <null>
MQCPI038 ......peer certificate uses UID=*,CN=*,T=*,OU=*,DC=*,O=*,
                                                STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI033 ......client authentication set to false
MQCPI078 Route 1416 ready for connection requests
```

3. 在 IBM MQ 客户机系统上的命令提示符处，输入以下命令：

a) 设置 **MQSERVER** 环境变量：

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

b) 放置消息：

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

输入消息字符串之后，按 Enter 键两次。

c) 获取消息：

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

此时会返回消息“Hello world”。

MQ Adv. V 9.3.3 MQ Adv. VUE Kafka Connect 方案

由于 IBM MQ 和 Apache Kafka 专门处理消息传递范围的不同方面，一个是连接，另一个是数据，因此解决方案通常需要数据在两者之间流动。 您可以使用 Kafka Connect 来实现此目的。

Kafka Connect 提供了一个框架，用于将数据从外部系统移动到 Kafka 集群或从 Kafka 集群移动到外部系统。 这是由连接器实现的。

有许多不同类型的连接器可用， IBM 提供了用于 IBM MQ 的连接器。 连接器有两种不同的类型：

- 源连接器将数据从外部系统传输到 Kafka。

  IBM MQ 源连接器使用来自 IBM MQ 队列的消息，并将其作为事件发布到 Kafka 主题中。
- 接收器连接器将数据从 Kafka 传输到外部系统。

  IBM MQ 接收器连接器使用来自 Kafka 主题的事件，并将这些事件作为消息发送到 MQ 队列。

更多信息，请参阅 "Kafka 连接和连接器。

Kafka Connect 方案可能包括：

- 将 IBM MQ 用作连接主干的核心银行系统。 您想要获取通过 IBM MQ 移动的消息的副本，并将其推送到 Kafka 以进行分析
- 您希望扩展核心银行系统以将数据发送到 Kafka 中，但仅希望数据在银行交易成功完成时输入 Kafka ，因此请使用 IBM MQ 作为交易网桥
- 您需要从多平台将数据输入到 z/OS 中。 多平台开发团队具有使用 Kafka 的经验， z/OS 团队希望利用 IBM MQ 与 CICS / IMS 的集成

从 IBM MQ 9.3.3 开始，如果您的企业具有 IBM MQ Advanced for z/OS VUE 权利， IBM MQ Advanced for Multiplatforms 权利或 IBM MQ Appliance 权利，那么您可以访问 IBM 提供的权利以及受支持的源和接收器连接器。

V 9.3.4 LTS 如果您享有 "IBM MQ Advanced 权利，则也适用于 "IBM MQ 9.3.4，以及适用 APARPH56722 的 "Long Term Support。

先前，您可以免费获取连接器，而无需支持，或者使用 IBM Event Streams 权利获取支持。

注意:

1. 这些方法可以与 Kafka 的任何变体 (例如， Apache Kafka 和 IBM Event Streams) 配合使用。
2. 仅为两个 IBM 连接器提供支持，而不支持 Kafka Connect 框架本身。

MQ Adv. V 9.3.3 MQ Adv. VUE Kafka 连接公共拓扑

本部分描述了通过 IBM 连接器将 IBM MQ 与 Kafka 集成时可以使用的三种方法。

请参阅 第 190 页的『获取连接器』 以获取有关获取连接器的更多信息，并参阅 第 191 页的『使用连接器』 以获取有关队列管理器连接和配置选项的更多信息。

直接到队列 (源)

要使用 IBM MQ 将数据发送到 Kafka 的应用程序可以将这些消息发送到 IBM MQ 源连接器所使用的队列。 然后， IBM MQ 源连接器会获取这些消息并将其传输到相关 Kafka 主题。

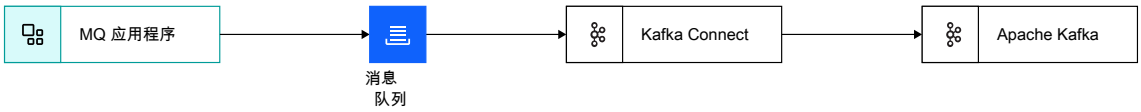


图 *47:* 直接到队列 *(*源*)*

当应用程序需要将数据发送到 Kafka 并且尚未将数据发送到 IBM MQ 时，应使用此方法。

使用 IBM MQ 发送数据意味着可以在与其他更新 (例如，到数据库) 协调的事务内完成消息发送。 此方法还可避免需要设置与 Kafka 的潜在短暂连接，而改为使用与 IBM MQ 的现有连接。

流式队列副本 (源)

在许多情况下，需要获取通过 IBM MQ 移动的现有数据的副本，并将其发送到 Kafka 中，例如，用于分析。 从 IBM MQ 9.3 开始，可以使用流式队列轻松实现此目的。 流式队列允许将放入一个队列的消息由队列管理器复制到第二个队列，而不会影响使用第一个队列的应用程序。 请参阅 第 26 页的『流式队列』，以了解更多信息。

例如：

```
DEF QL(TO.APP) STREAMQ(TO.KAFKA) STRMQOS(MUSTDUP)
DEF QL(TO.KAFKA)
```

表示将消息发送到 TO.APP 时，必须将该消息的副本发送到 TO.KAFKA。 然后， IBM MQ 源连接器从 TO.KAFKA 获取这些消息，并将其传输到相关 Kafka 主题。

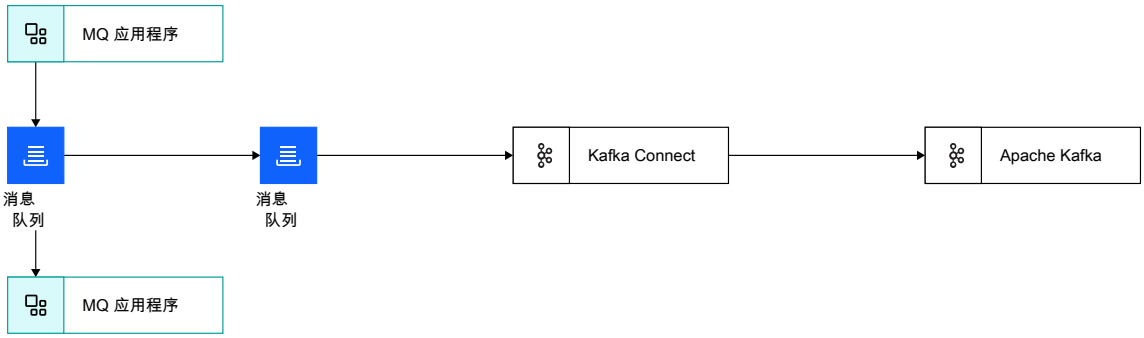


图 *48:* 流式队列副本 *(*源*)*

启用流式队列不会影响现有应用程序，因为原始消息不会更改。 发送到第二个队列的消息与具有相同有效内容，消息标识，相关标识等的原始消息相同。

直接到队列 (接收器)

与源连接器一样，可将接收器连接器配置为将数据从 Kafka 主题直接接收到队列中。

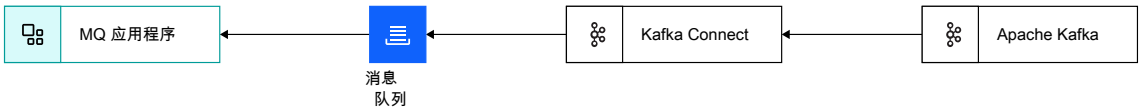


图 *49:* 直接到队列 *(*接收器*)*

通过 IBM MQ 接收数据意味着消息的接收可以在与其他更新 (例如数据库) 协调的事务中完成。

此方法还避免了需要设置与 Kafka 的潜在短期连接，而改为使用与 IBM MQ 的现有连接。

获取连接器

随时间推移，随 IBM MQ 提供的连接器版本会发生更改。

IBM MQ 9.3.3 附带的版本 1 连接器提供至少一次的消息传递。

V 9.3.4 从 IBM MQ 9.3.4 交付的版本 2 连接器至少提供一次，精确一次的消息传递。

V 9.3.4 有关至少一次交付与精确一次交付之间的差异以及如何配置精确一次交付的更多信息，请参阅第 192 页的『仅支持一次』。

MQ Adv. VUE MQ Adv. z/OS 对于 "IBM MQ 9.3.3 中的 "Continuous Delivery 和 "IBM MQ Advanced for z/OS Value Unit Edition 权限，" V 9.3.4 并从 "IBM MQ 9.3.4 获得 "IBM MQ Advanced for z/OS 的权利、中的连接器及其样本在连接器包组件 "`kafka-connect` 目录下的 "z/OS UNIX System Services（USS）中提供。 LTS 对于 Long Term Support，如果您拥有 IBM MQ Advanced 权限并已申请 APAR PH56722，则可以从本节所述来源获得连接器后使用它们。

MQ Adv. 在 IBM MQ Advanced for Multiplatforms 和 IBM MQ Appliance 中，登录 Fix Central 并搜索 `V.R.M.F-IBM-MQ-Kafka-Connectors.tar.gz`，例如 `9.3.3.0-IBM-MQ-Kafka-Connectors.tar.gz`，即可获取这些连接器和所需的配置文件。

MQ Adv. MQ Adv. VUE MQ Adv. z/OS 以下是每个 IBM MQ 版本随附的连接器:

IBM MQ 版本号	IBM MQ for Multiplatforms tar 文件名	源连接器版本	接收器连接器版本	一次性交付支持
V 9.3.3 9.3.3	9.3.3.0-IBM-MQ-Kafka_Connectors.tar.gz	1.3.2	1.5.0	否
V 9.3.4 9.3.4	9.3.4.0-IBM-MQ-Kafka_Connectors.tar.gz	2.0	2.0	Yes

您还可以通过 GitHub:的发布页面获取第 1 版连接器:

源

Kafka-connect-mq-source

接收端

Kafka-connect-mq-接收器

请注意， IBM 支持从这些链接获取的任何版本。

注意:

1. 但是，您应该始终采用最新版本的连接器，并定期检查更新。 IBM MQ 随附的连接器是产品交付时的最新连接器，并且定期更新为最新版本。
2. 如果通过对 IBM MQ Advanced for z/OS Value Unit Edition， IBM MQ Advanced for Multiplatforms 或 IBM MQ Appliance 的权利提供对 IBM MQ 连接器的支持，那么连接器必须连接到使用该权利运行的队列管理器。
3. 早于 "IBM MQ 9.3.3 的 "IBM MQ Advanced for z/OS Value Unit Edition、"IBM MQ Advanced for Multiplatforms 或 "IBM MQ Appliance 版本也可以使用 "IBM MQ 连接器，但连接器必须从 GitHub,上的发布页面下载（参见前文）。
4. V 9.3.4 早于 "IBM MQ 9.3.4 的 "IBM MQ Advanced for z/OS 的 "Continuous Delivery 版本也提供了 "IBM MQ 连接器的权限，但连接器必须从 GitHub,上的发布页面下载（参见前文）。

使用连接器

连接器使用属性或 JSON 文件进行配置。 样本文件随连接器一起提供。

以下位置提供了有关配置选项以及如何设置连接器的详细信息:

源连接器: Kafka 连接 IBM MQ 的源连接器
接收器连接器: Kafka 连接 IBM MQ 的接收器连接器

V 9.3.4 要在源连接器中仅启用一次支持，请参阅 运行 MQ 源连接器，对于接收器连接器，请参阅 运行 MQ 接收器连接器。

V 9.3.4 有关至少一次交付与精确一次交付之间的差异以及如何配置精确一次交付的更多信息，请参阅 第 192 页的『仅支持一次』。

要使 Kafka Connect 运行 IBM MQ 连接器，它需要在其类路径上具有连接器 JAR 和各种 IBM MQ JAR 文件。 需要以下 JAR 文件:

jms.jar
com.ibm.mq.allclient.jar
org.json.jar
V 9.3.5 bcpkix-jdk18on.jar (来自 IBM MQ 9.3.5)
bcpkix-jdk15to18.jar (IBM MQ 9.3.3 和 IBM MQ 9.3.4)
V 9.3.5 bcprov-jdk18on.jar (来自 IBM MQ 9.3.5)
bcprov-jdk15to18.jar (IBM MQ 9.3.3 和 IBM MQ 9.3.4)
V 9.3.5 bcutil-jdk18on.jar (来自 IBM MQ 9.3.5)
bcutil-jdk15to18.jar (IBM MQ 9.3.3 和 IBM MQ 9.3.4)

例如:

源连接器

V 9.3.5 从 IBM MQ 9.3.5 中:

```
export CLASSPATH=$CLASSPATH:/path-to-kafka-jars/kafka-connect-mq-source-1.3.2.jar:
/path-to-mq-jars/jms.jar:/path-to-mq-jars/com.ibm.mq.allclient.jar:/path-to-mq-jars/org.json.jar:
/path-to-mq-jars/bcpkix-jdk18on.jar:/path-to-mq-jars/bcprov-jdk18on.jar:/path-to-mq-jars/bcutil-
jdk18on.jar
```

IBM MQ 9.3.3 和 IBM MQ 9.3.4

```
export CLASSPATH=$CLASSPATH:/path-to-kafka-jars/kafka-connect-mq-source-1.3.2.jar:
/path-to-mq-jars/jms.jar:/path-to-mq-jars/com.ibm.mq.allclient.jar:/path-to-mq-jars/org.json.jar:
/path-to-mq-jars/bcpkix-jdk15to18.jar:/path-to-mq-jars/bcprov-jdk15to18.jar:/path-to-mq-jars/bcutil-
jdk15to18.jar
```

接收器连接器

V 9.3.5 从 IBM MQ 9.3.5 中:

```
export CLASSPATH=$CLASSPATH:/path-to-kafka-jars/kafka-connect-mq-sink-1.5.0.jar:
/path-to-mq-jars/jms.jar:/path-to-mq-jars/com.ibm.mq.allclient.jar:/path-to-mq-jars/org.json.jar:
/path-to-mq-jars/bcpkix-jdk18on.jar:/path-to-mq-jars/bcprov-jdk18on.jar:/path-to-mq-jars/bcutil-
jdk18on.jar
```

IBM MQ 9.3.3 和 IBM MQ 9.3.4

```
export CLASSPATH=$CLASSPATH:/path-to-kafka-jars/kafka-connect-mq-sink-1.5.0.jar:
/path-to-mq-jars/jms.jar:/path-to-mq-jars/com.ibm.mq.allclient.jar:/path-to-mq-jars/org.json.jar:
/path-to-mq-jars/bcpkix-jdk15to18.jar:/path-to-mq-jars/bcprov-jdk15to18.jar:/path-to-mq-jars/bcutil-
jdk15to18.jar
```

其中:

path-to-kafka-jars 是 IBM MQ 连接器安装位置的路径
path-to-mq-jars 是安装 IBM JMS 客户机的位置的路径。

z/OS 如果在 "连接器包" 组件中的 z/OS 上运行，那么 USS_ROOT/kafka-connect/source/kafka-connect-mq-source.jar 将指向最新版本的源连接器，而 USS_ROOT/kafka-connect/sink/kafka-connect-mq-sink.jar 将指向最新版本的接收器连接器。

Kafka Connect 和 IBM MQ 连接器可以在具有 Java 虚拟机的任何平台上运行。 它们不必与队列管理器或与其连接的 Kafka 集群在同一平台上运行。

但是，如果队列管理器与 Kafka 集群之间存在较长的距离，那么应将连接器定位到相对靠近队列管理器的位置; 理想情况下，这些连接器位于同一可用性区域或数据中心内。

在 z/OS 上使用连接器

z/OS

在所有平台 (包括 z/OS) 上运行的队列管理器完全支持这些连接器。 与 z/OS 队列管理器的连接可以通过服务器连接通道进行，也可以通过本地绑定进行。

在 IBM z/OS 和 IBM MQ for z/OS 上的性能测试环境中，通过在 z/OS UNIX System Services (USS) 中的 z/OS 上运行连接器并使用本地绑定连接到队列管理器来获取最佳性能。 此处提供了有关这些结果的详细信息: Kafka Connectors for IBM MQ - MQ for z/OS 透视图。

在 z/OS 上的 USS 中运行 Kafka Connect 需要执行一些额外的设置步骤; 此处提供了有关这些步骤的文档: 在 IBM z/OS 上运行连接器。

V 9.3.4 MQ Adv. MQ Adv. VUE MQ Adv. z/OS 仅支持一次

有两个版本的 IBM MQ Kafka 连接器 1 和 2。 版本 2 连接器提供对精确一次和至少一次消息传递的支持，而版本 1 连接器提供对至少一次消息传递的支持。

至少一次消息传递意味着在发生故障时，在 IBM MQ， IBM MQ Kafka 连接器或 Kafka 中:

- 对于源连接器， IBM MQ 消息不会丢失，但可能会多次传递到 Kafka，从而生成重复的 Kafka 消息。
- 对于接收器连接器， Kafka 消息不会丢失，但可能会多次传递到 IBM MQ，从而生成重复的 IBM MQ 消息。

精确一次消息传递意味着在发生故障时，在 IBM MQ， IBM MQ Kafka 连接器或 Kafka 中:

- 对于源连接器， IBM MQ 消息不会丢失，并且将传递到 Kafka，而不会出现重复的 Kafka 消息。
- 对于接收器连接器， Kafka 消息不会丢失，并且将传递到 IBM MQ，而不会出现重复的 IBM MQ 消息。

仅在 IBM MQ 或 IBM Event Streams 随附的 V 2 连接器中提供了精确一次支持。 它在 V 1 连接器中不可用。

版本 2 连接器可以至少一次或精确一次方式运行。 通过 Kafka 的相应配置并通过使用 "状态队列" 来启用一次支持。 以精确一次方式运行的连接器的每个实例都需要其自己的状态队列。

以精确一次方式运行的连接器的吞吐量和可伸缩性小于以至少一次方式运行的连接器的吞吐量和可伸缩性。仅当应用程序未设计为处理重复消息时，才启用精确一次方式。

请参阅 运行 MQ 源连接器 以获取有关在源连接器中配置精确一次方式的详细信息，以及 运行 MQ 接收器连接器 以获取有关在接收器连接器中配置精确一次方式的详细信息。

声明

本信息是为在美国国内供应的产品和服务而编写的。

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

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

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

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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

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

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

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

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation
软件互操作性协调员，部门 49XA
北纬 3605 号公路
罗切斯特，明尼苏达州 55901
U.S.A.

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

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

此处包含的任何性能数据都是在受控环境中测得的。 因此，在其他操作环境中获得的数据可能会有明显的不同。 有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。 此外，有些测量是通过推算而估计的， 实际结果可能会有差异。 本文档的用户应当验证其特定环境的适用数据。

© Copyright IBM Corp. 2007, 2025

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。 IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。 有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含日常商业运作所使用的数据和报表的示例。 为了尽可能全面地说明这些数据和报表，这些示例包括个人、公司、品牌和产品的名称。 所有这些名字都是虚构的，若现实生活中实际业务企业使用的名字和地址与此相似，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。 如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。 这些示例并未在所有条件下作全面测试。 因此，IBM 不能担保或默示这些程序的可靠性、可维护性或功能。

如果您正在查看本信息的软拷贝，图片和彩色图例可能无法显示。

编程接口信息

编程接口信息 (如果提供) 旨在帮助您创建用于此程序的应用软件。

本书包含有关允许客户编写程序以获取 WebSphere MQ 服务的预期编程接口的信息。

但是，该信息还可能包含诊断、修改和调优信息。 提供诊断、修改和调优信息是为了帮助您调试您的应用程序软件。

要点: 请勿将此诊断，修改和调整信息用作编程接口，因为它可能会发生更改。

商标

IBM IBM 徽标 ibm.com 是 IBM Corporation 在全球许多管辖区域的商标。 当前的 IBM 商标列表可从 Web 上的 “Copyright and trademark information”www.ibm.com/legal/copytrade.shtml 获取。 其他产品和服务名称可能是 IBM 或其他公司的商标。

Microsoft 和 Windows 是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

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

此产品包含由 Eclipse 项目 (https://www.eclipse.org/) 开发的软件。

Java 和所有基于 Java 的商标和徽标是 Oracle 和/或其附属公司的商标或注册商标。

部件号:

(1P) P/N: