9.2

配置 *IBM MQ*

注

在使用本资料及其支持的产品之前，请阅读第 895 页的『声明』中的信息。

本版本适用于 IBM® MQ V 9 发行版 2 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，而无须对您承担任何责任。

© Copyright International Business Machines Corporation 2007, 2025.

内容

配置.....7
在 Multiplatforms 版上创建队列管理器..... 7
可配置临时目录.....10
Userdata 目录..... 10
创建缺省队列管理器..... 11
使现有队列管理器成为缺省队列管理器.....12
创建队列管理器后备份配置文件.....13
配置客户机与服务器之间的连接..... 13
要使用的通信类型.....14
配置扩展事务客户机..... 16
定义 MQI 通道.....25
创建和使用 AMQP 通道.....25
在不同平台上创建服务器连接和客户机连接定义..... 30
在服务器上创建服务器连接和客户机连接定义.....34
MQI 通道的通道出口程序..... 49
将客户机连接到队列共享组..... 52
使用 IBM MQ 环境变量.....53
环境变量描述..... 55
在 Multiplatforms 版上的 .ini 文件中更改 IBM MQ 配置信息..... 71
IBM MQ 配置文件， mqs.ini.....72
队列管理器配置文件， qm.ini..... 82
安装配置文件， mqinst.ini..... 127
IBM MQ MQI client 配置文件， mqclient.ini..... 128
活动跟踪配置文件 mqat.ini..... 150
配置分布式队列.....152
IBM MQ 分布式排队技术..... 153
分布式队列管理简介..... 170
监视和控制 AIX, Linux, and Windows 上的通道.....197
监视和控制 IBM i 上的通道.....216
配置队列管理器集群..... 235
配置统一集群..... 335
配置发布/预订消息传递..... 351
设置排队的发布/预订消息属性.....352
正在启动排队的发布/预订..... 353
正在停止已排队的发布/预订.....353
添加流..... 354
删除流..... 355
添加预订点.....355
配置分布式发布/预订网络..... 356
配置多个安装.....372
在多安装环境中连接应用程序..... 372
更改主安装.....379
使队列管理器与安装相关联..... 380
在系统上查找 IBM MQ 的安装..... 381
配置高可用性，恢复和重新启动..... 382
自动客户机重新连接..... 383
控制台消息监视..... 388
高可用性配置..... 392
日志记录: 确保消息未丢失.....526
备份和复原 IBM MQ 队列管理器数据..... 550
对集群错误恢复的更改 (在除 z/OS 以外的服务器上)..... 557
配置 JMS 资源..... 558

在 JNDI 名称空间中配置连接工厂和目标 559
使用 IBM MQ Explorer 配置 JMS 对象 562
使用管理工具配置 JMS 对象 563
在 WebSphere Application Server 中配置 JMS 资源 570
配置应用程序服务器以使用最新的资源适配器维护级别 577
配置 JMS **PROVIDERVERSION** 属性 579
除去 WebSphere Application Server 持久预订 587
配置 Managed File Transfer 589
多平台上的 MFT 配置选项 590
z/OS 上的 MFT 配置选项 591
下载和配置 Redistributable Managed File Transfer components 591
创建 MFT 代理或记录器命令数据集 597
配置 Managed File Transfer for z/OS 598
配置 MFT on IBM i 626
为首次使用配置 MFT 627
配置 MFT 代理队列管理器 635
配置 MFT 记录器 643
配置 Connect:Direct 网桥 670
配置 IBM MQ Console 和 REST API 674
mqweb 服务器的基本配置 675
配置安全性 678
配置 HTTP 主机名 678
配置 HTTP 和 HTTPS 端口 679
配置响应超时 680
配置自动启动 681
配置日志记录 681
配置 LTPA 令牌 684
配置远程队列管理器连接行为 685
配置 administrative REST API 网关 686
配置 messaging REST API 687
为 MFT 配置 REST API 689
调整 mqweb 服务器 JVM 691
IBM MQ Console 和 REST API 安装组件的文件结构 692
在 Linux 或 Windows 上定义 Aspera gateway 连接 694
配置 IBM MQ 以用于 IBM Cloud Private 测量服务 699
配置队列管理器以与 IBM Cloud Private 上的测量服务实例配合使用 701
通过 HTTP 代理连接到 IBM Cloud Private 测量服务 703
对与测量服务的连接进行故障诊断 703
配置 IBM MQ 以用于 Salesforce 推送主题和平台事件 704
配置 IBM MQ Bridge to Salesforce 705
IBM MQ Bridge to Salesforce 的其他配置选项 709
为 Salesforce 平台事件创建事件消息 711
运行 IBM MQ Bridge to Salesforce 716
配置 IBM MQ 以用于区块链 717
为 IBM MQ Bridge to blockchain 创建配置文件 720
示例 Hyperledger Fabric 网络凭证文件 721
运行 IBM MQ Bridge to blockchain 723
IBM MQ Bridge to blockchain 的其他配置选项 729
在 z/OS 上配置队列管理器 730
准备在 z/OS 上定制队列管理器 731
设置 IBM MQ for z/OS 734
在 z/OS 上测试队列管理器 790
在 z/OS 上设置与其他队列管理器的通信 797
将 IBM MQ 用于 IMS 823
将 IBM MQ 与 CICS 结合使用 829
将服务升级并应用于 Language Environment 或 z/OS Callable Services 830
在 IMS 中使用 OTMA 出口 832
使用 IBM z/OSMF 自动执行 IBM MQ 835

配置 IBM MQ Advanced for z/OS VUE..... 845
启用与远程 z/OS 队列管理器的 MFT 代理程序连接..... 845
配置 IBM MQ Advanced for z/OS VUE 以用于区块链.....846
配置 IBM MQ Internet Pass-Thru..... 855
MQIPT 中的 HTTP 支持..... 856
MQIPT 中的 SOCKS 支持.....857
MQIPT 中的 SSL/TLS 支持.....858
Java security manager 在 MQIPT 中.....883
MQIPT 中的安全出口.....885
MQIPT 中的端口号控制..... 888
对 MQIPT 中存储的密码进行加密.....889
MQIPT 的其他安全注意事项.....890
MQIPT 中的连接日志.....891
使用容器配置 IBM MQ Internet Pass-Thru.....893
配置流式队列.....893

声明.....895
编程接口信息.....896
商标..... 896

配置 IBM MQ

在一台或多台计算机上创建一个或多个队列管理器，并在开发，测试和生产系统上配置这些队列管理器以处理包含业务数据的消息。

关于此任务

在配置 IBM MQ 之前，请阅读 IBM MQ 技术概述中的 IBM MQ 概念。 了解如何在 规划中规划 IBM MQ 环境。

您可以使用多种不同的方法在 IBM MQ 中创建，配置和管理队列管理器及其相关资源。 这些方法包括命令行界面，图形用户界面和管理 API。 有关这些接口的更多信息，请参阅 管理 IBM MQ。

有关如何创建，启动，停止和删除队列管理器的指示信息，请参阅 第 7 页的『在 Multiplatforms 版上创建队列管理器』。

有关如何创建将 IBM MQ 安装和应用程序连接在一起所需的组件的信息，请参阅 第 152 页的『配置分布式队列』。

有关如何使用不同方法将客户机连接到 IBM MQ 服务器的指示信息，请参阅 第 13 页的『配置客户机与服务器之间的连接』。

有关如何配置队列管理器集群的指示信息，请参阅 第 235 页的『配置队列管理器集群』。

您可以通过更改配置信息来更改 IBM MQ 或队列管理器的行为。 有关更多信息，请参阅 第 71 页的『在 Multiplatforms 版上的 .ini 文件中更改 IBM MQ 配置信息』。 通常，您不需要重新启动队列管理器以使任何配置更改生效，但本产品文档中声明的情况除外。

z/OS 有关如何配置 IBM MQ for z/OS 的指示信息，请参阅 第 730 页的『在 z/OS 上配置队列管理器』。

相关概念

IBM MQ 技术概述

相关任务

管理本地 IBM MQ 对象

管理远程 IBM MQ 对象

IBM i 管理 IBMi

z/OS 管理 IBM MQ for z/OS

规划

z/OS 在 z/OS 上规划 IBM MQ 环境

第 730 页的『在 z/OS 上配置队列管理器』
使用以下指示信息在 IBM MQ for z/OS 上配置队列管理器。

Multi 在 Multiplatforms 版上创建队列管理器

必须先创建并启动至少一个队列管理器及其关联对象，然后才能使用消息和队列。 队列管理器管理与其关联的资源，特别是其拥有的队列。 它为消息排队接口 (MQI) 调用和命令的应用程序提供排队服务，以创建，修改，显示和删除 IBM MQ 对象。

开始之前

要点: IBM MQ 不支持包含空格的机器名。 如果在具有包含空间的机器名的计算机上安装 IBM MQ，那么无法创建任何队列管理器。

在创建队列管理器之前，必须考虑多个要点，尤其是在生产环境中。 完成以下核对表:

© Copyright IBM Corp. 2007, 2025

与队列管理器关联的安装

要创建队列管理器，请使用 IBM MQ control 命令 **crtmqm**。 **crtmqm** 命令会自动将队列管理器与从中发出 **crtmqm** 命令的安装相关联。 对于在队列管理器上运行的命令，必须从与该队列管理器关联的安装发出该命令。 您可以使用 setmqm 命令来更改队列管理器的关联安装。 请注意， Windows 安装程序不会将执行安装的用户添加到 mqm 组，有关更多详细信息，请参阅 在 AIX, Linux®, and Windows 上管理 IBM MQ 的权限。

命名约定

使用大写名称，以便可以在所有平台上与队列管理器通信。 请记住，名称是在您输入名称时指定的。 为了避免大量输入的不便，请不要使用不必要的长名称。

指定唯一的队列管理器名称

创建队列管理器时，请确保没有其他队列管理器在网络中的任何位置具有相同的名称。 在创建队列管理器时，不会检查队列管理器名称，并且不唯一的名称会阻止您创建用于分布式排队的通道。 此外，如果使用网络进行发布/预订消息传递，那么预订将与创建它们的队列管理器名称相关联。 因此，如果集群或层次结构中的队列管理器具有相同的名称，那么可能会导致发布无法访问这些队列管理器。

确保唯一性的一种方法是使用每个队列管理器名称的唯一节点名作为前缀。 例如，如果节点名为 ACCOUNTS，那么可以将队列管理器命名为 ACCOUNTS.SATURN.QUEUE.MANAGER，其中 SATURN 标识特定队列管理器，而 QUEUE.MANAGER 是可以提供给所有队列管理器的扩展。 或者，可以省略此项，但请注意 ACCOUNTS.SATURN 和 ACCOUNTS.SATURN.QUEUE.MANAGER 是不同的队列管理器名称。

如果要使用 IBM MQ 与其他企业进行通信，那么还可以包含您自己的企业名称作为前缀。 这在示例中没有显示出来，因为这使它们更难以为继。

注: 控制命令中的队列管理器名称区分大小写。 这意味着允许您创建两个名称为 jupiter.queue.manager 和 JUPITER.queue.manager 的队列管理器。 但是，最好避免这种并发症。

限制队列管理器数

您可以在资源允许的情况下创建任意数量的队列管理器。 但是，由于每个队列管理器都需要自己的资源，因此在一个节点上具有一个具有 100 个队列的队列管理器通常比具有十个队列的队列管理器更好。

在生产系统中，可以使用单个队列管理器来利用许多处理器，但较大的服务器可以更有效地使用多个队列管理器来运行。

指定缺省队列管理器

每个节点都应具有缺省队列管理器，但可以在没有缺省队列管理器的节点上配置 IBM MQ 。 缺省队列管理器是应用程序在 MQCONN 调用中未指定队列管理器名称时连接到的队列管理器。 在不指定队列管理器名称的情况下调用 runmqsc 命令时，也是处理 MQSC 命令的队列管理器。

将队列管理器指定为缺省队列管理器会替换为该节点指定的任何现有缺省队列管理器。

更改缺省队列管理可能会影响其他用户或应用程序。 此更改对当前连接的应用程序没有影响，因为它们可以在任何其他 MQI 调用中使用来自其原始连接调用的句柄。 此句柄确保将调用定向到同一队列管理器。 连接 之后 的任何应用程序都已更改缺省队列管理器连接到新的缺省队列管理器。 这可能是您想要的，但在更改缺省值之前，您应该将其考虑在内。

第 11 页的『创建缺省队列管理器』中描述了如何创建缺省队列管理器。

指定死信队列

死信队列是本地队列，如果无法将消息路由到其预期目标，那么会将这些消息放入其中。

网络中的每个队列管理器上均应具有死信队列，这一点很重要。 如果没有定义死信队列，那么应用程序中的错误可能会导致关闭通道，并且您可能无法接收到对管理命令的应答。

例如，如果应用程序尝试将消息放在另一个队列管理器上的队列上，但给出了错误的队列名称，那么通道将停止，并且消息将保留在传输队列上。 然后，其他应用程序无法将此通道用于其消息。

如果队列管理器具有死信队列，那么通道不受影响。 将未传递的消息放在接收端的死信队列上，使通道及其传输队列可用。

创建队列管理器时，请使用 **-u** 标志来指定死信队列的名称。 您还可以使用 MQSC 命令来变更已定义的队列管理器的属性，以指定要使用的死信队列。 请参阅 显示和改变队列管理器属性 ，以获取 MQSC 命令 ALTER 的示例。

指定缺省传输队列

传输队列是一个本地队列，传输到远程队列管理器的消息在传输之前将在该本地队列上排队。 缺省传输队列是没有显式定义传输队列时所使用的队列。 可为每个队列管理器分配一个缺省传输队列。

创建队列管理器时，请使用 **-d** 标志来指定缺省传输队列的名称。 这实际上不会创建队列; 您必须在以后显式地执行此操作。 请参阅 使用本地队列 以获取更多信息。

指定所需的日志记录参数

可以在 crtmqm 命令上指定日志记录参数，包括日志记录类型以及日志文件的路径和大小。

在开发环境中，缺省日志记录参数应该足够。 但是，您可以在以下情况下更改缺省值:

- 您具有无法支持大型日志的低端系统配置。
- 您预计会有大量长消息同时出现在您的队列中。
- 您预计会有大量持久消息通过队列管理器传递。

设置日志记录参数后，只能通过删除队列管理器并使用相同名称但使用不同日志记录参数重新创建队列管理器来更改其中一些参数。

有关日志记录参数的更多信息，请参阅 第 382 页的『配置高可用性，恢复和重新启动』。

AIX **仅适用于 IBM MQ for UNIX 系统**

在使用 **crtmqm** 命令之前，可以创建队列管理器目录 /var/mqm/qmgrs/*qmgr*，即使在单独的本地文件系统上也是如此。 使用 **crtmqm** 时，如果 /var/mqm/qmgrs/*qmgr* 目录存在，为空，并且由 mqm 拥有，那么它将用于队列管理器数据。 如果目录不是由 mqm 拥有，那么创建将失败并返回 First Failure Support Technology (FFST) 消息。 如果该目录不为空，那么将创建新目录。

关于此任务

要创建队列管理器，请使用 IBM MQ control 命令 **crtmqm**。 有关更多信息，请参阅 **crtmqm**。 **crtmqm** 命令会自动创建所需的缺省对象和系统对象 (请参阅 系统缺省对象)。 缺省对象构成您创建的任何对象定义的基础; 队列管理器操作需要系统对象。

Windows 在 Windows 系统上，您可以选择使用 **crtmqm** 命令的 *sax* 选项来启动队列管理器的多个实例。

创建队列管理器及其对象后，可以使用 **strmqm** 命令来启动队列管理器。

过程

- 有关帮助您创建和管理队列管理器的信息，请参阅以下子主题:
  - 第 11 页的『创建缺省队列管理器』
  - 第 12 页的『使现有队列管理器成为缺省队列管理器』
  - 第 13 页的『创建队列管理器后备份配置文件』

相关概念

使用队列管理器

相关任务

创建名为 QM1 的队列管理器

第 71 页的『在 Multiplatforms 版上的 .ini 文件中更改 IBM MQ 配置信息』
您可以通过编辑配置 (.ini) 文件中的信息来更改 IBM MQ 或单个队列管理器的行为，以满足安装需要。 您还可以更改 IBM MQ MQI clients 的配置选项。

第 730 页的『在 z/OS 上配置队列管理器』
使用以下指示信息在 IBM MQ for z/OS 上配置队列管理器。

相关参考

系统和缺省对象

克特姆

Linux AIX V 9.2.0 可配置临时目录

可配置的临时目录定义队列管理器的数据临时位置。 这可用于允许将 AIX and Linux 域套接字放在 Red Hat® OpenShift® 环境中的非安装文件系统上。

在 IBM MQ 9.2.0 之前的 AIX and Linux 平台上，当队列管理器正在运行时，将在 `/var/mqm/sockets` 目录下创建 AIX and Linux 域套接字。 在容器内运行队列管理器 (将 `/var/mqm` 作为已安装的文件系统) 时，某些 Linux 平台可能会阻止创建这些域套接字，因为它们允许容器外的某些进程干扰容器内的操作。 此问题阻止 IBM MQ 在缺省安全上下文下的 Red Hat OpenShift 容器平台中运行。

从 IBM MQ 9.2.0 开始，可以使用 **EphemeralPrefix** 属性来配置临时目录的位置。 如果不使用此属性，那么不会看到行为发生任何更改。

在 `mqs.ini` 中创建队列管理器条目 (使用 **crtmqm** 或 **addmqinf** 命令) 时，如果您执行以下操作，那么将添加 **EphemeralPrefix** 属性:

- 在 AllQueueManagers 节中设置 **DefaultEphemeralPrefix** 属性。
- 设置 MQ_EPHEMERAL_PREFIX 环境变量。
- 仅对 **addmqinf** 命令指定 **-v** *EphemeralPrefix* 。

您还可以在现有队列管理器停止时显式地将 **EphemeralPrefix** 属性添加到该队列管理器，并在重新启动队列管理器时添加此属性。

如果指定 **EphemeralPrefix** 属性，那么当队列管理器启动时，会导致在该前缀下而不是其通常位置下创建队列管理器的数据临时数据。 即:

- 通常在 `/var/mqm/sockets/<QM>` 下存在的套接字文件现在将在 `/<EphemeralPrefix>/sockets/<QM>` 下
- 通常在 `/<Prefix>/qmgrs/<QM>/@<Subpool>` 下存在的子池文件现在将在 `/<EphemeralPrefix>/qmgrs/<QM>/@<Subpool>` 下

注意:

- `/var/mqm/sockets/@SYSTEM` 保留在其固定位置中，并且不是 **EphemeralPrefix** 属性的一部分。
- AMQCLCHL.TAB 保留在 `/<Prefix>/qmgrs/<QM>/@ipcc` 下，并且不是 **EphemeralPrefix** 属性的一部分。

EphemeralPrefix 属性可包含的字符数取决于您的平台:

- Linux AIX 在 AIX and Linux 平台上，限制为 12 个字符。
- IBM i 在 IBM i 上，限制为 24 个字符。

如果指定的 **EphemeralPrefix** 属性太长或不存在，那么会收到 AMQ7001E 消息:

```
AMQ7001E: 为队列管理器指定的位置无效
```

.

Multi V 9.2.0 Userdata 目录

您可以使用 `userdata` 目录来存储持久应用程序状态。

每个 IBM MQ 队列管理器都有一个用于其持久状态的专用文件系统，其中包括其队列数据和恢复日志。 文件系统包含可用于存储应用程序的持久状态信息的 `userdata` 目录。 请参阅 Unix 和 Linux Systems 上的目录内容 和 Windows 系统上的目录内容。

`userdata` 目录在许多情况下很有用，例如:

- 在 RDQM 配置中，当队列管理器故障转移到另一个节点时，应用程序信息也会移动 (请参阅 第 477 页的『存储持久应用程序状态』)。
- 对于多实例队列管理器，使其应用程序状态与其队列管理器数据位于共享网络文件系统上。
- 更一般地说，其中应用程序是配置的队列管理器服务。

如果选择将应用程序状态存储在 userdata 目录中，那么您必须知道写入此位置的数据可能会消耗分配给队列管理器的可用磁盘空间。 必须确保有足够的磁盘空间可供队列管理器写入队列数据，日志和其他持久状态信息。

userdata 目录具有 mqm 用户和组所有权，并且具有全局可读性，因此用户无需在 IBM MQ 管理员组 (即 mqm) 中即可访问该目录。 您无法修改 userdata 目录的许可权，但可以使用所需的任何所有权和许可权在其中创建内容。

Multi 创建缺省队列管理器

缺省队列管理器是应用程序在 MQCONN 调用中未指定队列管理器名称时连接到的队列管理器。 在不指定队列管理器名称的情况下调用 **runmqsc** 命令时，也是处理 MQSC 命令的队列管理器。 要创建队列管理器，请使用 IBM MQ control 命令 **crtmqm**。

开始之前

在创建缺省队列管理器之前，请先阅读 第 7 页的『在 Multiplatforms 版上创建队列管理器』中描述的注意事项。

AIX 当您使用 **crtmqm** 在 AIX and Linux 上创建队列管理器时，如果 /var/mqm/qmgrs/*qmgr* 目录已存在，由 mqm 拥有并且为空，那么它将用于队列管理器数据。 如果目录不是由 mqm 拥有，那么队列管理器的创建将失败并显示 First Failure Support Technology (FFST) 消息。 如果该目录不为空，那么将为队列管理器数据创建新目录。

即使 /var/mqm/qmgrs/*qmgr* 目录已存在于单独的本地文件系统上，此注意事项也适用。

关于此任务

使用 **crtmqm** 命令创建队列管理器时，该命令会自动创建所需的缺省对象和系统对象。 缺省对象构成您创建的任何对象定义的基础，并且队列管理器操作需要系统对象。

通过在命令中包含相关参数，还可以定义 (例如) 队列管理器要使用的缺省传输队列的名称以及死信队列的名称。

Windows 在 Windows 上，可以使用 **crtmqm** 命令的 **sax** 选项来启动队列管理器的多个实例。

有关 **crtmqm** 命令及其语法的更多信息，请参阅 **crtmqm**。

过程

- 要创建缺省队列管理器，请使用带有 **-q** 标志的 **crtmqm** 命令。
  crtmqm 命令的以下示例创建名为 SATURN.QUEUE.MANAGER 的缺省队列管理器:

  ```
  crtmqm -q -d MY.DEFAULT.XMIT.QUEUE -u SYSTEM.DEAD.LETTER.QUEUE SATURN.QUEUE.MANAGER
  ```

  其中:

  -q
  指示此队列管理器是缺省队列管理器。

  -d MY.DEFAULT.XMIT.QUEUE
  此队列管理器要使用的缺省传输队列的名称。

  注: IBM MQ 不会为您创建缺省传输队列; 您必须自行定义。

  -u SYSTEM.DEAD.LETTER.QUEUE
  是 IBM MQ 在安装时创建的缺省死信队列的名称。

  SATURN.QUEUE.MANAGER
  此队列管理器的名称。 这必须是在 crtmqm 命令上指定的最后一个参数。

下一步做什么

创建队列管理器及其对象后，请使用 **strmqm** 命令来 启动队列管理器。

相关概念

使用本地队列

相关任务

第 13 页的『创建队列管理器后备份配置文件』

IBM MQ 配置信息存储在 AIX, Linux, and Windows 上的配置文件中。 创建队列管理器后，备份配置文件。 然后，如果您创建另一个导致问题的队列管理器，那么可以在除去问题源后恢复备份。

显示和改变队列管理器属性

相关参考

系统和缺省对象

Multi 使现有队列管理器成为缺省队列管理器

您可以使用文本编辑器手动将现有队列管理器设置为缺省队列管理器，也可以在 Windows 和 Linux 上使用 IBM MQ Explorer 将其设置为缺省队列管理器。

关于此任务

要使用文本编辑器使现有队列管理器成为缺省队列管理器，请完成以下步骤。

Windows Linux 在 Windows 和 Linux (x86 和 x86-64 平台) 系统上，如果您希望使用 IBM MQ Explorer 来进行此更改，请参阅 第 12 页的『使用 IBM MQ Explorer 使队列管理器成为缺省队列管理器』。

创建缺省队列管理器时，其名称将插入到 IBM MQ 配置文件 (`mqs.ini`) 中 `DefaultQueueManager` 节的 `Name` 属性中。 如果节及其内容不存在，那么将自动创建这些节及其内容。

过程

- 要使现有队列管理器成为缺省队列管理器，请将 `Name` 属性上的队列管理器名称更改为新的缺省队列管理器的名称。 您可以使用文本编辑器手动执行此操作。
- 如果节点上没有缺省队列管理器，并且要使现有队列管理器成为缺省队列管理器，请自行创建具有所需名称的 *`DefaultQueueManager`* 节。
- 如果您意外使另一个队列管理器成为缺省队列管理器，并且想要还原为原始缺省队列管理器，请编辑 `mqs.ini` 中的 `DefaultQueueManager` 节，将不需要的缺省队列管理器替换为您想要的缺省队列管理器。

相关任务

第 71 页的『在 Multiplatforms 版上的 .ini 文件中更改 IBM MQ 配置信息』

您可以通过编辑配置 (`.ini`) 文件中的信息来更改 IBM MQ 或单个队列管理器的行为，以满足安装需要。 您还可以更改 IBM MQ MQI clients 的配置选项。

Windows Linux 使用 IBM MQ Explorer 使队列管理器成为缺省队列管理器

在 Windows 和 Linux (x86 和 x86-64 平台) 系统上，可以使用 IBM MQ Explorer 使现有队列管理器成为缺省队列管理器。

关于此任务

要使用 IBM MQ Explorer 使现有队列管理器成为 Windows 和 Linux (x86 和 x86-64 平台) 系统上的缺省队列管理器，请完成以下步骤。

如果您希望使用文本编辑器手动进行此更改，请参阅 第 12 页的『使现有队列管理器成为缺省队列管理器』。

过程

1. 打开 IBM MQ Explorer。
2. 右键单击 **IBM MQ**，然后选择 **属性 ...**。 此时将显示 " **IBM MQ** " 面板。
3. 在 **缺省队列管理器名称** 字段中输入缺省队列管理器的名称。
4. 单击**确定**。

ALW 创建队列管理器后备份配置文件

IBM MQ 配置信息存储在 AIX, Linux, and Windows 上的配置文件中。 创建队列管理器后，备份配置文件。 然后，如果您创建另一个导致问题的队列管理器，那么可以在除去问题源后恢复备份。

关于此任务

作为一般规则，每次创建新的队列管理器时都备份配置文件。

有两种类型的配置文件:

- 安装产品时，将创建 IBM MQ 配置文件 (`mqs.ini`)。 它包含每次创建或删除队列管理器时都会更新的队列管理器的列表。 每个节点有一个 `mqs.ini` 文件。
- 创建新的队列管理器时，将自动创建新的队列管理器配置文件 (`qm.ini`)。 这包含队列管理器的配置参数。

如果已安装 AMQP 服务，那么必须备份另一个配置文件:

- Windows 在 Windows 系统上: `amqp_win.properties`
- Linux AIX 在 AIX and Linux 系统上: `amqp_unix.properties`

相关任务

第 71 页的『在 Multiplatforms 版上的 .ini 文件中更改 IBM MQ 配置信息』
您可以通过编辑配置 (`.ini`) 文件中的信息来更改 IBM MQ 或单个队列管理器的行为，以满足安装需要。 您还可以更改 IBM MQ MQI clients 的配置选项。

第 550 页的『备份和复原 IBM MQ 队列管理器数据』
通过备份队列管理器和队列管理器数据，仅备份队列管理器配置以及使用备份队列管理器，可以保护队列管理器免受硬件故障可能导致的损坏。

配置客户机与服务器之间的连接

要配置 IBM MQ MQI clients 与服务器之间的通信链路，请决定通信协议，定义链路两端的连接，启动侦听器以及定义通道。

关于此任务

在 IBM MQ 中，对象之间的逻辑通信链路称为 通道。 用于将 IBM MQ MQI clients 连接到服务器的通道称为 MQI 通道。 您可以在链路的每一端设置通道定义，以便 IBM MQ MQI client 上的 IBM MQ 应用程序可以与服务器上的队列管理器进行通信。

在定义 MQI 通道之前，必须确定要使用哪种通信形式，并在通道的每个端定义连接。

如果要在 IBM MQ MQI client 与位于不同物理网络上的队列管理器之间定义 MQI 通道，或者定义通过防火墙进行通信的 MQI 通道，那么使用 IBM MQ Internet Pass-Thru 可以简化配置。 有关更多信息，请参阅 IBM MQ Internet Pass-Thru。

过程

1. 决定您要使用的通信形式。
   请参阅 第 14 页的『要使用的通信类型』。
2. 定义通道每一端的连接。

要定义连接，必须执行以下操作:

a) 配置连接。

b) 记录通道定义所需的参数值。

c) 通过启动 侦听器，使服务器能够检测来自 IBM MQ MQI client 的入局网络请求。

相关概念

第 128 页的『IBM MQ MQI client 配置文件， mqclient.ini』

您可以使用文本文件中的属性来配置客户机。 这些属性可由环境变量或其他特定于平台的方式覆盖。

相关任务

第 53 页的『使用 IBM MQ 环境变量』

您可以使用命令来显示当前设置或重置 IBM MQ 环境变量的值。

将 IBM MQ MQI 客户机应用程序连接到队列管理器

相关参考

显示 CHLAUTH

SET CHLAUTH

要使用的通信类型

不同的平台支持不同的通信协议。 您选择的传输协议取决于 IBM MQ MQI client 和服务器平台的组合。

MQI 通道的传输协议类型

根据您的客户机和服务器平台， MQI 通道最多有四种类型的传输协议:

- TCP/IP
- LU 6.2
- NetBIOS
- SPX

定义 MQI 通道时，每个通道定义都必须指定传输协议 (传输类型) 属性。 服务器不限于一个协议，因此不同的通道定义可以指定不同的协议。 对于 IBM MQ MQI clients，具有使用不同传输协议的备用 MQI 通道可能很有用。

您选择的传输协议还取决于 IBM MQ 客户机和服务器平台的特定组合。 下表中显示了可能的组合。

表 *1:* 传输协议- *IBM MQ MQI client* 与服务器平台的组合

传输协议	**IBM MQ MQI client**	**IBM MQ 服务器**
TCP/IP 第 15 页的『1』	IBM i AIX Linux Windows	IBM i AIX Linux Windows z/OS

表 1: 传输协议- *IBM MQ MQI client* 与服务器平台的组合 (继续)		
传输协议	**IBM MQ MQI client**	**IBM MQ 服务器**
LU 6.2	AIX AIX Linux Linux 第 15 页的『2』 Windows Windows	IBM i IBM i AIX AIX Linux Linux 第 15 页的『2』 Windows Windows z/OS z/OS
NetBIOS	Windows Windows	Windows Windows
SPX	Windows Windows	Windows Windows

注意:

1. CD MQ Adv. V 9.2.0 使用 TCP/IP 的消息通道可以指向 IBM Aspera fasp.io Gateway，它提供了可显着提高网络吞吐量的快速 TCP/IP 隧道。 请参阅 在 Linux 或 Windows 上定义 Aspera gateway 连接。
2. Linux 除外 (POWER 平台)

相关概念

第 206 页的『在 Windows 上定义 TCP 连接』
通过在发送端配置通道以指定目标地址，并通过在接收端运行侦听器程序来定义 TCP 连接。

第 212 页的『在 AIX and Linux 上定义 TCP 连接』
发送端的通道定义指定目标的地址。 在接收端为连接配置了侦听器或 inet 守护程序。

第 229 页的『在 IBM i 上定义 TCP 连接』
您可以使用 "连接名称" 字段在通道定义中定义 TCP 连接。

第 816 页的『在 z/OS 上定义 TCP 连接』
要定义 TCP 连接，需要配置许多设置。

第 208 页的『在 Windows 上定义 LU 6.2 连接』
必须配置 SNA，以便可以在两台机器之间建立 LU 6.2 对话。

第 215 页的『在 AIX and Linux 上定义 LU 6.2 连接』
必须配置 SNA，以便可以在两台机器之间建立 LU 6.2 对话。

第 230 页的『在 IBM i 上定义 LU 6.2 连接』
使用标准 LU 6.2 连接的方式名， TP 名和连接名来定义 LU 6.2 通信详细信息。

第 209 页的『在 Windows 上定义 NetBIOS 连接』
NetBIOS 连接仅适用于运行 Windows 的客户机和服务器。 IBM MQ 在与其他 IBM MQ 产品建立 NetBIOS 连接时使用三种类型的 NetBIOS 资源: 会话，命令和名称。 其中每个资源都有一个限制，缺省情况下或在安装 NetBIOS 期间通过选择来建立该限制。

相关任务

第 694 页的『在 Linux 或 Windows 上定义 Aspera gateway 连接』
IBM Aspera fasp.io Gateway 提供了快速 TCP/IP 隧道，可以显着提高 IBM MQ 的网络吞吐量。 在任何授权平台上运行的队列管理器都可以通过 Aspera gateway 进行连接。 网关本身部署在 Red Hat 或 Ubuntu Linux 或 Windows 上。

相关参考

第 16 页的『TCP/IP 连接限制』

可以在单个 TCP/IP 端口上排队的未完成连接请求数取决于平台。 如果达到限制，那么会发生错误。

第 818 页的『使用 APPC/MVS 为 z/OS 定义 LU6.2 连接』
要定义 LU6.2 连接，需要配置许多设置。

TCP/IP 连接限制

可以在单个 TCP/IP 端口上排队的未完成连接请求数取决于平台。 如果达到限制，那么会发生错误。

此连接限制与可以连接到 IBM MQ 服务器的最大客户机数不同。 您可以将更多客户机连接到服务器，直至达到服务器系统资源所确定的级别。 下表显示了连接请求的积压值:

表 *2:* 在 *TCP/IP* 端口上排队的最大未完成连接请求数

服务器平台	最大连接请求数
AIX	100
	20
Linux	100
IBM i	255
	100
Windows Windows 服务器	100
Windows Windows 工作站	100
z/OS	255

如果达到连接限制，那么客户机将收到来自 MQCONN 调用的返回码 MQRC_HOST_NOT_AVAILABLE，以及客户机错误日志中的 AMQ9202 错误 (AIX and Linux 系统上的 `/var/mqm/errors/AMQERR0n.LOG` 或 Windows 上的 IBM MQ 客户机安装的 errors 子目录中的 `amqerr0n.log`)。 如果客户机重试 MQCONN 请求，那么可能成功。

要增加您可以发出的连接请求数，并避免由此限制生成错误消息，您可以让多个侦听器每个侦听器在不同的端口上侦听，或者具有多个队列管理器。

配置扩展事务客户机

此主题集合描述了如何配置每个事务管理器类别的扩展事务功能。

对于每个平台，扩展事务客户机为以下外部事务管理器提供支持:

符合 XA 的事务管理器
扩展事务客户机提供 XA 资源管理器接口以支持 XA 兼容的事务管理器，例如 CICS 和 Tuxedo。

Windows **Microsoft Transaction Server (仅限 Windows 系统)**
仅在 Windows 系统上， XA 资源管理器接口还支持 Microsoft Transaction Server (MTS)。 随扩展事务客户机提供的 IBM MQ MTS 支持提供了 MTS 与 XA 资源管理器接口之间的网桥。

WebSphere Application Server
较低版本的 IBM WebSphere MQ 受支持 WebSphere Application Server 4 或 5，并且要求您执行某些配置任务以使用扩展事务客户机。 WebSphere Application Server 6 和更高版本包含 IBM WebSphere MQ 或 IBM MQ 消息传递提供程序，因此您不需要使用扩展事务客户机。

配置符合 XA 的事务管理器

首先配置 IBM MQ 基本客户机，然后使用这些主题中的信息配置扩展事务功能。

注: 本部分假定您已基本了解 "开放式组" 在 分布式事务处理*: XA* 规范中发布的 XA 接口。

要配置扩展事务客户机，必须首先配置 IBM MQ 基本客户机，如以下所述:

- AIX 在 AIX 上安装 IBM MQ 客户机
- Linux 在 Linux 上安装 IBM MQ 客户机
- Windows 在 Windows 上安装 IBM MQ 客户机
- IBM i 在 IBM i 上安装 IBM MQ 客户机

通过使用平台的信息，您可以为 XA 兼容的事务管理器 (例如， CICS 和 Tuxedo) 配置扩展事务功能。

事务管理器使用与连接到队列管理器的客户机应用程序所使用的 MQI 通道，作为资源管理器与队列管理器进行通信。 当事务管理器发出资源管理器 (xa_) 函数调用时， MQI 通道用于将调用转发到队列管理器，并接收来自队列管理器的输出。

事务管理器可以通过发出 xa_open 调用作为资源管理器打开队列管理器来启动 MQI 通道，或者客户机应用程序可以通过发出 MQCONN 或 MQCONNX 调用来启动 MQI 通道。

- 如果事务管理器启动 MQI 通道，并且客户机应用程序稍后在同一线程上调用 MQCONN 或 MQCONNX，那么 MQCONN 或 MQCONNX 调用将成功完成，并且会向应用程序返回连接句柄。 应用程序未收到带有 MQRC_ALREADY_CONNECTED 原因码的 MQCC_WARNING 完成代码。
- 如果客户机应用程序启动 MQI 通道，并且事务管理器稍后在同一线程上调用 xa_open，那么会使用 MQI 通道将 xa_open 调用转发到队列管理器。

在故障后的恢复情境中，当没有客户机应用程序在运行时，事务管理器可以使用专用 MQI 通道来恢复发生故障时队列管理器参与的任何不完整工作单元。

将扩展事务客户机与 XA 兼容的事务管理器配合使用时，请注意以下条件:

- 在单个线程中，一次只能将客户机应用程序连接到一个队列管理器。 仅当使用扩展事务客户机时，此限制才适用; 使用 IBM MQ 基本客户机的客户机应用程序可以在单个线程中同时连接到多个队列管理器。
- 客户机应用程序的每个线程都可以连接到不同的队列管理器。
- 客户机应用程序无法使用共享连接句柄。

要配置扩展事务功能，必须向充当资源管理器的每个队列管理器的事务管理器提供以下信息:

- xa_open 字符串
- 指向 XA 开关结构的指针

当事务管理器调用 xa_open 以将队列管理器作为资源管理器打开时，它会在调用时将 xa_open 字符串作为自变量 *xa_info* 传递给扩展事务客户机。 扩展事务客户机通过以下方式使用 xa_open 字符串中的信息:

- 启动到服务器队列管理器的 MQI 通道 (如果客户机应用程序尚未启动 MQI 通道)
- 检查事务管理器作为资源管理器打开的队列管理器是否与客户机应用程序连接到的队列管理器相同
- 要查找事务管理器的 ax_reg 和 ax_unreg 函数，如果队列管理器使用动态注册

有关 xa_open 字符串的格式以及有关扩展事务客户机如何使用 xa_open 字符串中的信息的更多详细信息，请参阅 第 18 页的『xa_open 字符串的格式』。

XA 切换结构使事务管理器能够找到扩展事务客户机提供的 xa _ 函数，并指定队列管理器是否使用动态注册。 有关随扩展事务客户机提供的 XA 开关结构的信息，请参阅 第 21 页的『XA 开关结构』。

有关如何为特定事务管理器配置扩展事务功能的信息，以及有关将事务管理器与扩展事务客户机配合使用的任何其他信息，请参阅以下部分:

- 第 23 页的『为 CICS 配置扩展事务客户机』
- 第 24 页的『为 Tuxedo 配置扩展事务客户机』

相关概念

第 20 页的『xa_open 字符串的 CHANNEL， TRPTYPE， CONNAME 和 QMNAME 参数』
使用此信息来了解扩展事务客户机如何使用这些参数来确定要连接到的队列管理器。

第 21 页的『xa_open 的其他错误处理』

xa_open 调用在某些情况下失败。

相关任务

第 22 页的『将扩展事务客户机与 TLS 通道配合使用』
无法使用 xa_open 字符串设置 TLS 通道。 遵循以下指示信息以使用客户机通道定义表 (ccdt)。

相关参考

第 20 页的『TPM 和 AXLIB 参数』
扩展事务客户机使用 TPM 和 AXLIB 参数来查找事务管理器的 ax_reg 和 ax_unreg 函数。 仅当队列管理器使用动态注册时，才会使用这些函数。

第 21 页的『在扩展事务处理失败后恢复』
发生故障后，事务管理器必须能够恢复任何不完整的工作单元。 为此，事务管理器必须能够以资源管理器身份打开在发生故障时参与不完整工作单元的任何队列管理器。

z/OS 扩展事务客户机连接的 *IBM MQ for z/OS* 注意事项

某些 XA 事务管理器使用与连接到 IBM MQ for z/OS 的客户机通常可用的功能不兼容的事务协调调用序列。

在检测到不兼容序列的情况下， IBM MQ for z/OS 可能会针对连接发出异常终止，并向客户机返回错误响应。

例如， `xa_prepare` 接收异常终止 5C6-00D4007D，并将返回码 -3 (XAER_RMERR) 返回到客户机。

另一个示例是 `xa_end` 接收异常终止 5C6-00D40079。

确保已在 IBM MQ for z/OS 上启用对 XA 客户机连接的更改，这允许事务管理器在另一个连接上准备事务。

注意:

- 缺省情况下未启用更改。 要使用此更改，必须在 XA 客户机使用的 SVRCONN 通道的描述字段中的任何位置指定关键字 CSQSERVICE1 (大写)。
- 具有 CSQSERVICE1 关键字的通道具有以下限制:
  - 不允许使用 GROUP 恢复单元处置。 仅允许使用 QMGR 恢复处置单元。 处置由 xa_open 调用上给定的名称确定。 如果使用队列共享组名，那么 XA 连接将请求组恢复单元。

    在 **xa_info** 参数中指定队列共享组名的 xa_open 调用失败，并返回 *xaer_inval*。
  - 不允许使用 *MQGMO_LOCK* 和 *MQGMO_UNLOCK* 选项。 使用 *MQGMO_LOCK* 或 *MQGMO_UNLOCK* 的 MQGET 调用失败，发生 MQRC_ENVIRONMENT_ERROR。

相关概念

第 16 页的『配置符合 XA 的事务管理器』
首先配置 IBM MQ 基本客户机，然后使用这些主题中的信息配置扩展事务功能。

xa_open 字符串的格式

xa_open 字符串包含定义的参数名称和值对。

xa_open 字符串具有以下格式:

```
parm_name1 = parm_value1, parm_name2 = parm_value2, ...
```

其中 *parm_name* 是参数的名称， *parm_value* 是参数的值。 参数的名称不区分大小写，但除非另有说明，否则参数的值区分大小写。 您可以按任何顺序指定参数。

参数的名称，含义和有效值如下所示:

名称
含义和有效值

通道
MQI 通道的名称。

这是可选参数。 如果提供了此参数，那么还必须提供 CONNAME 参数。

TRPTYPE
MQI 通道的通信协议。 以下协议是有效值:

LU62
SNA LU 6.2

NETBIOS
NetBIOS

SPX
IPX/SPX

TCP
TCP/IP

这是可选参数。 如果省略该值，那么将采用 TCP 的缺省值。 参数的值不区分大小写。

CONNAME
MQI 通道服务器端的队列管理器的网络地址。 此参数的有效值取决于 TRPTYPE 参数的值:

LU62
符号目标名称，用于标识 CPI-C 辅助信息条目。

伙伴 LU 的网络限定名不是有效值，也不是伙伴 LU 别名。 这是因为没有其他参数来指定事务程序 (TP) 名称和方式名。

NETBIOS
NetBIOS 名称。

SPX
4 字节网络地址， 6 字节节点地址和可选的 2 字节套接字号。 这些值必须以十六进制表示法指定。 句点必须分隔网络和节点地址，如果提供了套接字号，那么必须将其括在括号中。 例如:

```
0a0b0c0d.804abcde23a1(5e86)
```

如果省略套接字号，那么将采用缺省值 5e86 。

TCP
主机名或 IP 地址 (可选) ，后跟括号中的端口号。 如果省略了端口号，那么将采用缺省值 1414。 可以使用分号分隔符来指定队列管理器的多个主机和端口，例如:

```
host1(1415);host2(1416);host3(1417)
```

这是可选参数。 如果提供了此参数，那么还必须提供 CHANNEL 参数。

QMNAME
MQI 通道服务器端的队列管理器的名称。 名称不能为空白或单个星号 (*) ，也不能以星号开头。 这意味着参数必须按名称标识特定队列管理器。

这是必需参数。

当客户机应用程序连接到特定队列管理器时，必须由同一队列管理器处理任何事务恢复。

如果应用程序正在连接到 z/OS 队列管理器，那么应用程序可以指定特定队列管理器的名称或队列共享组 (QSG) 的名称。 通过使用队列管理器名称或队列共享组名，应用程序控制它是参与具有 QMGR 恢复处置单元或 GROUP 恢复处置单元的事务。 恢复处置的 GROUP 单元允许在 QSG 的任何成员上处理事务的恢复。 要使用 GROUP 恢复单元，必须启用 **GROUPUR** 队列管理器属性。

z/OS 有关使用 GROUP 恢复单元的更多信息，请参阅 队列共享组中的恢复处置单元。

TPM
正在使用的事务管理器。 有效值为 CICS 和 TUXEDO。

扩展事务客户机将此参数和 AXLIB 参数用于相同目的。 有关这些参数的更多信息，请参阅 TPM 和 AXLIB 参数。

这是可选参数。 参数的值不区分大小写。

AXLIB
包含事务管理器的 ax_reg 和 ax_unreg 函数的库的名称。

这是可选参数。

UID
提供给队列管理器以进行认证的用户标识。 如果提供了此参数，那么还必须提供 **PWD** 参数。 如果已认证所提供的用户标识和密码，那么将使用该用户标识来标识事务管理器的连接。 用户标识和密码填充 MQCONNX 调用上的 MQCSP 对象。

UID 和 **PWD** 参数对客户机和服务器绑定都有效。

PWD
提供给队列管理器以进行认证的密码。 如果提供了此参数，那么还必须提供 **UID** 参数。

警告: 在某些情况下，客户机应用程序的 MQCSP 结构中的密码将通过纯文本网络发送。 要确保客户机应用程序密码受到适当保护，请参阅 IBM MQCSP 密码保护。

以下是 xa_open 字符串的示例:

```
channel=MARS.SVR,trptype=tcp,conname=MARS(1415),qmname=MARS,tpm=cics
```

xa_open 字符串的 *CHANNEL*， *TRPTYPE*， *CONNAME* 和 *QMNAME* 参数

使用此信息来了解扩展事务客户机如何使用这些参数来确定要连接到的队列管理器。

如果在 xa_open 字符串中提供了 **CHANNEL** 和 **CONNAME** 参数，那么扩展事务客户机将使用这些参数和 **TRPTYPE** 参数来启动到服务器队列管理器的 MQI 通道。

如果未在 xa_open 字符串中提供 **CHANNEL** 和 **CONNAME** 参数，那么扩展事务客户机将使用 MQSERVER 环境变量的值来启动 MQI 通道。 如果未定义 MQSERVER 环境变量，那么扩展事务客户机将使用 **QMNAME** 参数所标识的客户机通道定义中的条目。

在每种情况下，扩展事务客户机都会检查 **QMNAME** 参数的值是否是 MQI 通道的服务器端的队列管理器的名称。 如果不是，那么 xa_open 调用将失败，事务管理器将向应用程序报告失败。

如果应用程序连接到版本低于 IBM WebSphere MQ 7.0.1 的队列管理器，那么 xa_open 调用成功，但事务具有 QMGR 恢复处置单元。 z/OS 确保需要 GROUP 恢复单元处置的应用程序仅连接到 IBM WebSphere MQ 7.0.1 或更高版本的队列管理器。

z/OS 如果应用程序在 **QMNAME** 参数字段中使用队列共享组名，并且在它所连接的队列管理器上禁用了 GROUPUR 属性，那么 xa_open 调用将失败。

z/OS 如果应用程序客户机正在连接到位于 IBM WebSphere MQ 7.0.1 或更高版本的 z/OS 队列管理器，那么可以为 **QMNAME** 参数指定队列共享组 (QSG) 名称。 这允许应用程序客户机参与具有 GROUP 恢复处置单元的事务。 有关 GROUP 恢复处置单元的更多信息，请参阅 恢复处置单元。

当客户机应用程序稍后在事务管理器用来发出 xa_open 调用的同一线程上调用 MQCONN 或 MQCONNX 时，应用程序将接收由 xa_open 调用启动的 MQI 通道的连接句柄。 未启动第二个 MQI 通道。 扩展事务客户机检查 MQCONN 或 MQCONNX 调用上 **QMgrName** 参数的值是否为 MQI 通道服务器端的队列管理器的名称。 如果不是，那么 MQCONN 或 MQCONNX 调用将失败，原因码为 MQRC_ANOTHER_Q_MGR_CONNECTED。 如果 **QMgrName** 参数的值为空白或单个星号 (*)，或者以星号开头，那么 MQCONN 或 MQCONNX 调用将失败，原因码为 MQRC_Q_MGR_NAME_ERROR。

如果客户机应用程序已通过在事务管理器在同一线程上调用 xa_open 之前调用 MQCONN 或 MQCONNX 来启动 MQI 通道，那么事务管理器将改为使用此 MQI 通道。 未启动第二个 MQI 通道。 扩展事务客户机将检查 xa_open 字符串中 **QMNAME** 参数的值是否为服务器队列管理器的名称。 如果不是，那么 xa_open 调用将失败。

如果客户机应用程序首先启动 MQI 通道，那么 MQCONN 或 MQCONNX 调用上 **QMgrName** 参数的值可以是空白或单个星号 (*)，也可以以星号开头。 但是，在这些情况下，您必须确保应用程序所连接的队列管理器与稍后在同一线程上调用 xa_open 时事务管理器打算作为资源管理器打开的队列管理器相同。 因此，如果 *QMgrName* 参数的值按名称显式标识队列管理器，那么您可能会迁到较少的问题。

TPM 和 *AXLIB* 参数

扩展事务客户机使用 TPM 和 AXLIB 参数来查找事务管理器的 ax_reg 和 ax_unreg 函数。 仅当队列管理器使用动态注册时，才会使用这些函数。

如果在 xa_open 字符串中提供了 TPM 参数，但未提供 AXLIB 参数，那么扩展事务客户机将根据 TPM 参数的值来假定 AXLIB 参数的值。 请参阅 第 21 页的表 3 以获取 AXLIB 参数的假定值。

表 *3: AXLIB* 参数的假定值

TPM 的值	**平台**	**AXLIB 的假定值**
CICS	AIX	/usr/lpp/encina/lib/libEncServer.a(EncServer_shr.o)
CICS	Windows Windows 系统	libEnc 服务器
Tuxedo	AIX	/usr/lpp/tuxedo/lib/libtux.a(libtux.so.60)
Tuxedo	Windows Windows 系统	利布图

如果在 xa_open 字符串中提供了 AXLIB 参数，那么扩展事务客户机将使用其值来覆盖基于 TPM 参数值的任何假定值。 AXLIB 参数还可用于 TPM 参数没有指定值的事务管理器。

xa_open 的其他错误处理

xa_open 调用在某些情况下失败。

本节中的主题描述了 xa_open 调用失败的情况。 如果发生以下任何情况，那么此操作也会失败:

- xa_open 字符串中存在错误。
- 没有足够的信息来启动 MQI 通道。
- 尝试启动 MQI 通道时发生问题 (例如，服务器队列管理器未在运行)。

在扩展事务处理失败后恢复

发生故障后，事务管理器必须能够恢复任何不完整的工作单元。 为此，事务管理器必须能够以资源管理器身份打开在发生故障时参与不完整工作单元的任何队列管理器。

因此，在对任何配置信息进行更改之前，必须确保已解决所有不完整的工作单元。

或者，您必须确保配置更改不会影响事务管理器打开其需要打开的队列管理器的能力。 以下是此类配置更改的示例:

- 更改 xa_open 字符串的内容
- 更改 MQSERVER 环境变量的值
- 更改客户机通道定义表 (CCDT) 中的条目
- 删除服务器连接通道定义

XA 开关结构

每个平台上的扩展事务客户机都提供了两个 XA 开关结构。

这些开关结构为:

MQRMIXASwitch
当充当资源管理器的队列管理器未使用动态注册时，事务管理器将使用此切换结构。

MQRMIXASwitchDynamic
当充当资源管理器的队列管理器使用动态注册时，事务管理器将使用此切换结构。

这些交换机结构位于 第 22 页的表 4 中显示的库中。

表 4: 包含 *XA* 开关结构的 *IBM MQ* 库	
平台	**包含 XA 开关结构的库**
AIX AIX Linux Linux	*MQ_INSTALLATION_PATH*/lib/libmqcxa
Windows Windows 系统	*MQ_INSTALLATION_PATH*\bin\mqcxa.dll [1]

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

每个交换机结构中 IBM MQ 资源管理器的名称为 MQSeries_XA_RMI，但许多队列管理器可以共享同一交换机结构。

相关概念

第 22 页的『动态注册和扩展事务处理』
使用动态注册是一种优化形式，因为它可以减少事务管理器发出的 xa _ 函数调用数。

动态注册和扩展事务处理

使用动态注册是一种优化形式，因为它可以减少事务管理器发出的 xa _ 函数调用数。

如果队列管理器不使用动态注册，那么事务管理器将涉及每个工作单元中的队列管理器。 事务管理器通过调用 xa_start， xa_end 和 xa_prepare 来执行此操作，即使队列管理器没有在工作单元中更新的资源也是如此。

如果队列管理器使用动态注册，那么事务管理器将通过假定队列管理器未参与工作单元来启动，并且不会调用 xa_start。 然后，仅当在同步点控制中更新了队列管理器的资源时，该队列管理器才会参与工作单元。 如果发生此情况，那么扩展事务客户机将调用 ax_reg 以注册队列管理器的参与。

将扩展事务客户机与 *TLS* 通道配合使用

无法使用 xa_open 字符串设置 TLS 通道。 遵循以下指示信息以使用客户机通道定义表 (ccdt)。

关于此任务

由于 xa_open xa_info 字符串的大小有限，因此无法使用连接到队列管理器的 xa_open 字符串方法传递设置 TLS 通道所需的所有信息。 因此，必须使用客户机通道定义表，或者如果事务管理器允许，在发出 xa_open 调用之前使用 MQCONNX 创建通道。

要使用客户机通道定义表，请执行以下步骤:

过程

1. 指定仅包含必需的 qmname (队列管理器名称) 参数的 xa_open 字符串，例如:
   `XA_Open_String=qmname=MYQM`
2. 使用队列管理器来定义具有必需 TLS 参数的 CLNTCONN (客户机/连接) 通道。 在 CLNTCONN 定义的 QMNAME 属性中包含队列管理器名称。 这将与 xa_open 字符串中的 qmname 匹配。
3. 使 CLNTCONN 定义可用于客户机通道定义表 (CCDT) 中的客户机系统，或者在 Windows 上的 Active Directory 中的客户机系统。
4. 如果您正在使用 CCDT，请使用环境变量 MQCHLLIB 和 MQCHLTAB 来标识包含 CLNTCONN 通道定义的 CCDT。 在客户机应用程序和事务管理器使用的环境中设置这些变量。

结果

这为事务管理器提供了针对相应队列管理器的通道定义，其中包含正确认证所需的 TLS 属性，包括 SSLCIPH 和 CipherSpec。

为 *CICS* 配置扩展事务客户机

通过将 XAD 资源定义添加到 CICS 区域，配置扩展事务客户机以供 CICS 使用。

使用 CICS resource definition online (RDO) 命令 **cicsadd** 添加 XAD 资源定义。 XAD 资源定义指定以下信息:

- xa_open 字符串
- 交换机装入文件的标准路径名

提供了一个交换机装入文件，供 CICS 在以下每个平台上使用:

- AIX
- Windows

每个切换装入文件都包含一个函数，该函数返回一个指向用于动态注册的 XA 切换结构 MQRMIXASwitchDynamic 的指针。 请参阅 第 23 页的表 5 以获取每个交换机装入文件的标准路径名。

表 *5:* 交换机装入文件

平台	切换装入文件
AIX Linux	*MQ_INSTALLATION_PATH*/lib/amqczsc
Windows	*MQ_INSTALLATION_PATH*\bin\mqcc4swi.dll [1]

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

以下是 Windows 系统的 XAD 资源定义示例:

```
cicsadd -c xad -r REGION1 WMQXA \
    ResourceDescription="IBM MQ queue manager MARS" \
    XAOpen="channel=MARS.SVR,trptype=tcp,conname=MARS(1415),qmname=MARS,tpm=cics" \
    SwitchLoadFile="C:\Program Files\IBM\MQ\bin\mqcc4swi.dll"
```

有关将 XAD 资源定义添加到 CICS 区域的更多信息，请参阅适用于您的平台的 *CICS Administration Reference* 和 *CICS Administration Guide* 。

请注意有关将 CICS 与扩展事务客户机配合使用的以下信息:

- 只能将 IBM MQ 的一个 XAD 资源定义添加到 CICS 区域。 这意味着只能有一个队列管理器与一个区域相关联，并且在该区域中运行的所有 CICS 应用程序都只能连接到该队列管理器。 如果要运行连接到其他队列管理器的 CICS 应用程序，那么必须在其他区域中运行这些应用程序。
- 区域中的每个应用程序服务器在初始化时调用 xa_open ，并启动到与该区域关联的队列管理器的 MQI 通道。 这意味着必须在应用程序服务器启动之前启动队列管理器，否则 xa_open 调用将失败。 稍后由应用程序服务器处理的所有 IBM MQ MQI client 应用程序都使用相同的 MQI 通道。
- 当 MQI 通道启动并且通道的客户机端没有安全出口时，从客户机系统流向服务器连接 MCA 的用户标识为 `cics`。 在某些情况下，当服务器连接 MCA 随后尝试代表客户机应用程序访问队列管理器资源时，队列管理器使用此用户标识进行权限检查。 如果此用户标识用于权限检查，那么必须确保它具有访问其需要访问的所有资源的权限。

  有关队列管理器何时使用此用户标识进行权限检查的信息，请参阅 保护。
- 第 24 页的表 6 中列出了为在 IBM MQ 客户机系统上使用而提供的 CICS 任务终止出口。 配置这些出口的方式与配置 IBM MQ 服务器系统的相应出口的方式相同。 因此，有关此信息，请参阅 启用 CICS 用户出口。

表 6: CICS 任务终止出口		
平台	**源**	**库**
AIX AIX Linux Linux	amqzscgx.c	阿姆格茨格
Windows Windows 系统	amqzscgn.c	mqcc1415.dll

为 *Tuxedo* 配置扩展事务客户机

要配置供 Tuxedo 使用的 XAD 资源定义，请更新 UBBCONFIG 文件和资源管理器表。

要配置供 Tuxedo 使用的 XAD 资源定义，请执行以下操作:

- 在应用程序的 Tuxedo UBBCONFIG 文件的 GROUPS 部分中，使用 **OPENINFO** 参数指定 xa_open 字符串。 有关如何执行此操作的示例，请参阅提供用于 Tuxedo 样本程序的样本 UBBCONFIG 文件。

  AIX 在以下平台上，该文件的名称为 `ubbstxcx.cfg`:

  – AIX

  Windows Windows，文件名为 `ubbstxcn.cfg`。
- 在 Tuxedo 资源管理器表中队列管理器的条目中，指定 XA 切换结构的名称以及包含该结构的库的标准路径名:

  – AIX 在 AIX 上，指定 `udataobj/RM`。

  – Windows 在 Windows 上，指定 `udataobj\rm`。

  有关如何对每个平台执行此操作的示例，请参阅 TUXEDO 样本。 Tuxedo 支持资源管理器的动态注册，因此您可以使用 MQRMIXASwitch 或 MQRMIXASwitchDynamic。

Windows Microsoft 事务服务器

在可以将 Microsoft Transaction Server (MTS) 用作事务管理器之前，不需要其他配置。 但是，有一些要点需要注意。

请注意有关将 MTS 与扩展事务客户机配合使用的以下信息:

- MTS 应用程序在连接到服务器队列管理器时始终启动 MQI 通道。 MTS 以事务管理器的角色，然后使用相同的 MQI 通道与队列管理器进行通信。
- 发生故障后， MTS 必须能够恢复任何不完整的工作单元。 要执行此操作， MTS 必须能够与在发生故障时参与不完整工作单元的任何队列管理器进行通信。

  当 MTS 应用程序连接到服务器队列管理器并启动 MQI 通道时，扩展事务客户机会从 MQCONN 或 MQCONNX 调用的参数中抽取足够的信息，以便在发生故障后重新启动该通道 (如果需要)。 扩展事务客户机将信息传递给 MTS ，而 MTS 将信息记录在其日志中。

  如果 MTS 应用程序发出 MQCONN 调用，那么此信息只是队列管理器的名称。 如果 MTS 应用程序发出 MQCONNX 调用并提供通道定义结构 MQCD ，那么该信息还包括 MQI 通道的名称，服务器队列管理器的网络地址以及通道的通信协议。

  在恢复情况下， MTS 会将此信息传递回扩展事务客户机，而扩展事务客户机使用此信息来重新启动 MQI 通道。

  因此，如果您需要更改任何配置信息，请确保在进行更改之前已解决所有不完整的工作单元。 或者，确保配置更改不会影响扩展事务客户机使用 MTS 记录的信息重新启动 MQI 通道的能力。 以下是此类配置更改的示例:

  - 更改 MQSERVER 环境变量的值
  - 更改客户机通道定义表 (CCDT) 中的条目
  - 删除服务器连接通道定义
- 将扩展事务客户机与 MTS 配合使用时，请注意以下条件:
  - 在单个线程中，一次只能将客户机应用程序连接到一个队列管理器。
  - 客户机应用程序的每个线程都可以连接到不同的队列管理器。
  - 客户机应用程序无法使用共享连接句柄。

定义 MQI 通道

要创建新通道，您必须使用相同的通道名称和兼容的通道类型来创建 **两个** 通道定义，一个用于连接的每一端。 在这种情况下，通道类型为 *server-connection* 和 *client-connection*。

用户定义的通道

当服务器未自动定义通道时，可以通过两种方法来创建通道定义并授予 IBM MQ MQI client 机器上的 IBM MQ 应用程序对通道的访问权。

详细描述了这两种方法:

1. 在 IBM MQ 客户机上创建一个通道定义，在服务器上创建另一个通道定义。

   这适用于 IBM MQ MQI client 和服务器平台的任何组合。 在系统上入门或测试设置时，请使用此选项。

   有关如何使用此方法的详细信息，请参阅 第 30 页的『在不同平台上创建服务器连接和客户机连接定义』。
2. 在服务器上创建两个通道定义。

   在同时设置多个通道和 IBM MQ MQI client 机器时使用此方法。

   有关如何使用此方法的详细信息，请参阅 第 34 页的『在服务器上创建服务器连接和客户机连接定义』。

自动定义的通道

除 z/OS 以外的平台上的 IBM MQ 产品包含可在服务器上自动创建通道定义 (如果不存在) 的功能部件。

如果从客户机接收到入站连接请求，并且在该队列管理器上找不到相应的服务器连接定义，那么 IBM MQ 将自动创建定义并将其添加到队列管理器。 自动定义基于缺省服务器连接通道 SYSTEM.AUTO.SVRCONN。 通过使用带有 CHAD 参数的 ALTER QMGR 命令 (或带有 ChannelAutoDef 参数的 PCF 命令 "更改队列管理器") 来更新队列管理器对象，可以启用服务器连接定义的自动定义。

相关概念

第 178 页的『通道控制功能』
通道控制功能为您提供用于定义，监视和控制通道的工具。

ALW 创建和使用 AMQP 通道

将 AMQP 服务组件的 IBM MQ 支持安装到 IBM MQ 安装时，可以运行 IBM MQ MQSC 命令 (**runmqsc**) 来定义，变更，删除，启动和停止通道。 您还可以查看通道的状态。

开始之前

此任务假定您已安装 AMQP 通道。 要完成此任务，可在安装 IBM MQ 时选择 AMQP 服务组件。 有关更多信息，请遵循您平台的链接，然后查找 "AMQP 服务" 的表行:

- AIX 用于 AIX 系统的 IBM MQ 组件
- Linux 用于 Linux 系统的 IBM MQ rpm 组件
- Linux 针对 Linux Ubuntu 系统的 IBM MQ Debian 组件

- Windows 用于 Windows 系统的 IBM MQ 功能

要与队列管理器建立测试连接，您可以使用实现 OASIS AMQP 1.0 协议的任何 AMQP 客户机，例如 MQ Light 和 Apache Qpid 客户机，例如 Apache Qpid Proton 和 Apache Qpid JMS。

此任务基于 MQ Light Node.js 客户机。 但是，对于任何客户机，与 IBM MQ 队列管理器有关的步骤均相同。

关于此任务

以下过程假设您具有现有的队列管理器。

LTS 如果需要新的队列管理器，那么应包含一个样本脚本（位于 *mqinstall*/amqp/samples 目录中）。 该脚本用于创建新队列管理器，启动 AMQP 服务，并创建名为 SAMPLE.AMQP.CHANNEL 的新通道，然后启动该通道。

注: AMQP 通道不支持用户定义的 AMQP 服务。 AMQP 通道只支持系统缺省 SYSTEM.AMQP.SERVICE 服务。 只能为每个队列管理器定义此服务的一个实例。

Windows Linux LTS 如果运行样本脚本 (SampleMQM.sh on Linux 或 SampleMQM.bat on Windows) ，那么可以在 第 27 页的『6』上启动以下过程。

LTS 您可以使用缺省通道 SYSTEM.DEF.AMQP 来测试与队列管理器的 MQ Light 连接，也可以创建新通道。

V 9.2.2 从 IBM MQ 9.2.2 开始，只能使用缺省通道 SYSTEM.DEF.AMQP，用于测试与队列管理器的 MQ Light 连接。

以下过程使用缺省通道。

过程

1. 从 *mqinstall*/bin/ 目录启动 **runmqsc** :

   ```
   runmqsc QMNAME
   ```

2. (仅当队列管理器为 IBM MQ 9.0.4 或更早版本时才需要。) 检查 AMQP 功能是否已安装并正常工作。
   使用 **START SERVICE** 命令来启动 IBM MQ 服务，该服务控制 JVM:

   ```
   START SERVICE(SYSTEM.AMQP.SERVICE)
   ```

   注: 从 IBM MQ 9.1 开始，SYSTEM.AMQP.SERVICE 将 **CONTROL** 属性设置为 *QMGR*。 这将导致队列管理器启动时会自动启动服务。 通过将 **CONTROL** 属性设置为 *MANUAL*，您可以在队列管理器启动时阻止服务启动。

   启动队列管理器时，将自动启动 AMQP 服务和 AMQP 通道（如果已定义）。

3. 设置 MCAUSER 用户标识。
   当 AMQP 客户机连接到通道时，该通道将指定 MCAUSER 用户标识（与队列管理器的连接上将使用此用户标识）。 MCAUSER 的缺省值为空。 您必须先指定 MCAUSER 值（必须是有权发布和预订 IBM MQ 主题的有效 IBM MQ 用户），然后任何 AMQP 客户机才能够连接到队列管理器。

   注: Windows 在 Windows 上，在 IBM MQ 9.1.1 之前，仅对于长度最多为 12 个字符的用户标识支持 MCAUSER 用户标识设置。 V 9.2.0 从 IBM MQ 9.1.1 Continuous Delivery 和从 IBM MQ 9.2.0 Long Term Support 开始，将除去 12 个字符的限制。

   a) 使用 **ALTER CHANNEL** 命令来设置 MCAUSER 用户标识:

   ```
   ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) MCAUSER(User ID)
   ```

   b) 使用以下两个 **setmqaut** 命令来授权 MCAUSER 用户标识发布和预订主题:

```
setmqaut -m QMNAME -t topic -n SYSTEM.BASE.TOPIC -p MCAUSER
 -all +pub +sub
```

和

```
setmqaut -m QMNAME -t qmgr -p MCAUSER -all +connect
```

如果在添加或变更 MCAUSER 用户标识期间运行通道，那么必须停止并重新启动该通道。

注: 如果未设置 MCAUSER 用户标识，或者 MCAUSER 用户标识无权发布或预订 IBM MQ 主题，那么您将在 AMQP 客户机中收到一条错误信息。

4. 使用 **START CHANNEL** 命令来启动缺省 SYSTEM.DEF.AMQP 通道:

```
START CHANNEL(SYSTEM.DEF.AMQP)
```

5. 如果要检查通道状态，请使用 **DISPLAY CHSTATUS** 命令:

```
DISPLAY CHSTATUS(SYSTEM.DEF.AMQP) CHLTYPE(AMQP)
```

当通道正常运行时，命令输出中会显示 STATUS(RUNNING)。

6. 更改缺省端口。

AMQP 1.0 连接的缺省端口是 5672。 如果您已使用端口 5672（如果先前装了 MQ Light，就可能存在这种情况），那么需要更改 AMQP 通道使用的端口。 使用 **ALTER CHANNEL** 命令来更改端口:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) PORT(NEW PORT NUMBER)
```

7. 如果您不想使用通道认证 (CHLAUTH) 规则来阻止或过滤与 AMQP 通道的连接，请按如下所示禁用队列管理器上的通道认证:

```
alter qmgr chlauth(disabled)
```

建议不要禁用生产队列管理器上的连接认证。 您应在开发环境中禁用连接认证。

或者，将队列管理器通道认证规则配置为允许与 AMQP 通道建立特定连接。

8. 可选： 如果您想要使用队列管理器的已配置密钥存储库在通道上启用 SSL/TLS 加密，那么必须将通道的 SSLCIPH 属性设置为相应的密码规范。 缺省情况下，密码规范为空，这意味着通道上不使用 SSL/TLS 加密。 使用 **ALTER CHANNEL** 命令来设置密码规范。 例如:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLCIPH(CIPHER SPECIFICATION)
```

此外，SSL/TLS 加密还有许多其他相关的通道配置选项，您可以按如下所示进行设置:

- 缺省情况下，具有对应于队列管理器 **CERTLABL** 属性的标签的队列管理器密钥存储库中的证书是通道的 SSL/TLS 加密所使用的名称。 您可以通过设置 **CERTLABL** 来选择其他证书。 使用 **ALTER CHANNEL** 命令为所需证书指定标签:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) CERTLABL(CERTIFICATE LABEL)
```

- 您可以将通道设置为需要来自 SSL/TLS 客户机连接的证书。 您可以通过设置 **SSLCAUTH** 属性来选择是否需要来自 SSL/TLS 客户机连接的证书。 使用 **ALTER CHANNEL** 命令设置是否需要来自 SSL/TLS 客户机连接的证书。 例如:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLCAUTH(REQUIRED or OPTIONAL)
```

- 如果将 **SSLCAUTH** 属性设置为 REQUIRED，那么可以检查来自客户机的证书的专有名称 (DN)。 要从客户机检查证书的专有名称，请设置 **SSLPEER** 属性。 使用 **ALTER CHANNEL** 命令可检查来自客户机的证书的专有名称。 例如:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLPEER (DN SPECIFICATION)
```

或者，您还可以使用通道认证记录来允许或阻止连接，因为与使用 **SSLPEER** 属性相比，此方法提供了更大的粒度。 有关设置 **SSLPEER** 和使用通道认证记录作为替代方法的更多信息，请参阅 SSL 对等。

9. 通过运行以下命令来安装 MQ Light Node.js 客户机：

```
npm install mqlight
```

10. 浏览至 node_modules/mqlight/samples 目录，并运行样本接收方应用程序：
    - 如果您使用的是缺省端口号，那么可以运行样本接收方应用程序：

      ```
      node recv.js
      ```

    - 如果已将 AMQP 通道配置为使用其他端口号，那么可以使用参数来指定新端口号以运行样本接收方应用程序：

      ```
      node recv.js -s amqp://localhost:6789
      ```

    如果成功连接到缺省通道，那么将显示以下消息：

    ```
    Connected to amqp://localhost:5672 using client-id recv_e79c55d
    Subscribed to pattern: public
    ```

    该应用程序现在已连接到队列管理器，并等待接收消息。 其预订了主题 public。

    注: client-id 是自动生成的，除非您使用 -i 参数指定了一个客户机标识。
11. 在新命令窗口中，浏览至 node_modules/mqlight/samples 目录，并通过运行以下命令来运行样本发送方应用程序：

    ```
    node send.js
    ```

    在接收方应用程序的命令窗口中，将显示“Hello World”消息。
12. 使用 **AMQSSUB** IBM MQ 样本接收 MQ Light 样本消息。
    在 Linux 和 Windows 上，可以在以下位置找到样本：
    - Linux Linux 上的 *mqinstall*/samp/bin 目录。
    - Windows Windows 上的 *mqinstall*/Tools\c\Samples\Bin 目录。

    a) 通过运行以下命令来运行该样本：

    ```
    amqssub public QM-name.
    ```

    b) 通过重新运行以下命令来将消息发送到 IBM MQ 应用程序：

    ```
    node send.js
    ```

13. 使用 **DEFINE CHANNEL** 命令可创建更多 AMQP 通道：

    ```
    DEFINE CHANNEL(MY.AMQP.CHANNEL) CHLTYPE(AMQP) PORT(2345)
    ```

    定义通道后，必须使用 **START CHANNEL** 命令手动启动该通道：

    ```
    START CHANNEL(MY.AMQP.CHANNEL)
    ```

    要检查通道是否在正常运行，您可以指定新通道的端口来运行样本接收方应用程序：

    ```
    node recv.js -s amqp://localhost:2345
    ```

下一步做什么

您可以使用以下命令来显示 IBM MQ 连接，停止通道和删除通道：

DISPLAY CONN(*) TYPE(*CONN*) WHERE (*CHANNEL EQ SYSTEM.DEF.AMQP*)
显示 AMQP 通道在队列管理器上建立的 IBM MQ 连接。

DISPLAY CHSTATUS(*) CHLTYPE(*AMQP*) CLIENTID(*) ALL
显示已连接到指定通道的 AMQP 客户机的列表。

STOP CHANNEL (*MY.AMQP.CHANNEL*)
阻止 AMQP 通道，并且关闭它正在侦听的端口。

DELETE CHANNEL (*MY.AMQP.CHANNEL*)
删除您创建的任何通道。

注: 请勿删除缺省通道 SYSTEM.DEF.AMQP。

您可以使用 **runmqsc** 或 PCF 来确定 AMQP 功能是否已安装到 IBM MQ 安装中，以及是否存在与其关联的队列管理器:

- 使用 **runmqsc**，显示队列管理器的属性并检查 AMQPCAP(YES)。
- 使用 PCF，使用 **MQCMD_INQUIRE_Q_MGR** 命令，并确认 MQIA_AMQP_CAPABILITY 的值。

相关任务
开发 AMQP 客户机应用程序
保护 AMQP 客户机安全
相关参考
strmqm

ALW 从队列管理器中除去 AMQP 通道

您可以从安装目录中除去相应文件夹，以从队列管理器中除去 AMQP 通道。

过程

1. 停止队列管理器。
2. 除去对 AMQP 服务组件 API 的 IBM MQ 支持:
   - AIX 在 AIX 上，运行以下命令:
     ```
     installp -u mqm.amqp.rte
     ```
   - Linux 在 Linux 上，除去 AMQP RPM。 如果您在安装前已对 RPM 重新打包，请指定重新打包后的 RPM 的名称。
     ```
     rpm -e MQSeriesAMQP
     ```
   - Windows 在 Windows 上，从 IBM MQ 安装中除去 amqp 文件夹。 确保未除去 IBM MQ 安装路径中的其他文件或文件夹。
3. 重新启动队列管理器。

相关任务
开发 AMQP 客户机应用程序
保护 AMQP 客户机安全

ALW AMQP 通道日志文件

AMQP 通道的日志文件与 IBM MQ 日志文件存储在相同的 IBM MQ 数据目录中。

Windows 上的缺省数据目录是 C:\ProgramData\IBM\MQ。

Linux 上的缺省数据目录为 /var/mqm。

AMQP 通道将日志信息写入 IBM MQ 数据目录中的以下日志文件中:

- amqp.stdout，写入 qmgrs/*QM-name* 文件夹。
- amqp.stderr，写入 qmgrs/*QM-name* 文件夹。
- amqp_*.log ，写入 qmgrs/*QM-name*/errors 文件夹。

如果 MQ Light 客户机接收到认证或授权错误，那么管理员可以在 amqp_0.log 文件和 MQ AMQERR*.log 文件中找到有关安全性失败原因的详细信息。

任何 FDC 文件都创建为 AMQP*.FDC 文件，并写入 *data-directory*/errors 文件夹。

某些配置文件将写入 qmgrs/*QM-name*/amqp 目录。 您无需编辑此目录中的任何文件。

相关概念

AIX, Linux, and Windows 上的错误日志

相关任务

开发 AMQP 客户机应用程序

保护 AMQP 客户机安全

在不同平台上创建服务器连接和客户机连接定义

您可以在应用通道定义的计算机上创建每个通道定义。 但是，对于如何在客户端计算机上创建通道定义存在一些限制。

关于此任务

在所有平台上，可以使用 IBM MQ 脚本 (MQSC) 命令，可编程命令格式 (PCF) 命令或 IBM MQ Explorer 在服务器上定义服务器连接通道。

z/OS 在 z/OS 上，您还可以使用 "操作" 和 "控制" 面板。

IBM i 在 IBM i 上，您还可以使用面板界面。

由于 MQSC 命令仅在安装了 IBM MQ 作为 IBM MQ MQI client 的机器上不可用，因此您必须使用不同的方法在客户机上定义客户机连接通道。

当 **runmqsc** 时，以下注意事项适用:

- 您可以指定 **-c** 参数和 **-u** 参数 (可选)，以将 **runmqsc** 作为客户机连接到要管理的队列管理器。
- 如果使用 **-u** 参数来提供用户标识，那么系统会提示您输入匹配的密码。
- 如果已使用 CHCKLOCL(REQUIRED) 或 CHCKLOCL(REQDADM)配置 CONNAUTH AUTHINFO 记录，那么必须使用 **-u** 参数，否则将无法使用 **runmqsc** 管理队列管理器。

过程

- 要在服务器上定义服务器连接通道，请参阅 第 30 页的『在服务器上定义服务器连接通道』。
- 要在 IBM MQ MQI client 上创建客户机连接通道，请参阅 第 31 页的『使用 MQSERVER 在 IBM MQ MQI client 上创建客户机连接通道』。

在服务器上定义服务器连接通道

必要时启动 MQSC，然后定义服务器连接通道。

过程

1. 可选： 如果服务器平台不是 z/OS，请先创建并启动队列管理器，然后启动 MQSC 命令。
   a) 创建名为 QM1 的队列管理器，例如:

   ```
   crtmqm QM1
   ```

   b) 启动队列管理器:

   ```
   strmqm QM1
   ```

   c) 启动 MQSC 命令:

```
runmqsc QM1
```

2. 使用所选名称和通道类型 *server-connection* 定义通道。

```
DEFINE CHANNEL(CHAN1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
DESCR('Server-connection to Client_1')
```

此通道定义与在服务器上运行的队列管理器相关联。

3. 使用以下命令以允许对队列管理器进行入站连接访问:

```
SET CHLAUTH(CHAN1) TYPE(ADDRESSMAP) ADDRESS('IP address') MCAUSER('userid')
```

- 其中 SET CHLAUTH 使用上一步中定义的通道的名称。
- 其中 *"IP address"* 是客户机的 IP 地址。
- 其中， *"userid"* 是要提供给通道以用于对目标队列进行访问控制的标识。 此字段区分大小写。

您可以选择使用多个不同的属性来标识入站连接。 此示例使用 IP 地址。 备用属性包括客户机用户标识和 TLS 主题专有名称。 有关更多信息，请参阅 通道认证记录

使用 MQSERVER 在 IBM MQ MQI client 上创建客户机连接通道

您可以使用 **MQSERVER** 环境变量在客户机工作站上定义客户机连接通道。

关于此任务

您可以使用 **MQSERVER** 环境变量来指定客户机连接通道的简单定义。 很简单，您只能使用此方法指定通道的一些属性。

如果使用 **MQSERVER** 环境变量来定义 IBM MQ MQI client 机器与服务器之间的通道，那么这是可供应用程序使用的唯一通道，并且不会引用客户机通道定义表 (CCDT)。

如果 MQCONN 或 MQCONNX 请求指定的队列管理器不是侦听器所连接的队列管理器，或者如果无法识别 **MQSERVER** 参数 *TransportType* ，那么 MQCONN 或 MQCONNX 请求将失败，返回码为 MQRC_Q_MGR_NAME_ERROR。

Linux AIX 在 AIX and Linux 上，您可以定义 **MQSERVER** ，如以下其中一个示例中所示:

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56(2002)'
export MQSERVER=CHANNEL1/LU62/BOX99
```

然后，所有 MQCONN 或 MQCONNX 请求都尝试使用您定义的通道，除非已从提供给 MQCONNX 的 MQCNO 结构中引用 MQCD 结构，在这种情况下， MQCD 结构指定的通道优先于 **MQSERVER** 环境变量指定的任何通道。

MQSERVER 环境变量优先于 **MQCHLLIB** 和 **MQCHLTAB** 环境变量所指向的任何客户机通道定义。

过程

- 根据您的平台，使用下列其中一个命令来指定 **MQSERVER** 的通道定义。
  - Windows 在 Windows 上，指定简单通道定义，如下所示:

    ```
    SET MQSERVER=ChannelName/TransportType/ConnectionName
    ```

    例如:

    ```
    export  MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
    ```

  - Linux AIX 在 AIX and Linux 上，指定简单通道定义，如下所示:

    ```
    export MQSERVER=ChannelName/TransportType/ConnectionName
    ```

例如：

```
SET  MQSERVER=SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)
```

- IBM i 在 IBM i 上，指定简单通道定义，如下所示：

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('ChannelName/TransportType/ConnectionName')
```

例如：

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)')
```

注意：

- *ChannelName* 必须与服务器上定义的名称相同。 它不能包含正斜杠 (/) 字符，因为此字符用于分隔通道名称，传输类型和连接名称。 当 **MQSERVER** 环境变量用于定义客户机通道时，将使用最大消息长度 (**MAXMSGL**) 100 MB。 因此，对通道有效的最大消息大小是在服务器上的 SVRCONN 通道中指定的值。
- *TransportType* 可以是 LU62， TCP， NETBIOS 和 SPX 之一，具体取决于您的 IBM MQ 客户机平台。
- Linux AIX 在 AIX and Linux 上， *TransportType* 区分大小写，并且必须为大写。 如果无法识别传输类型，那么 MQCONN 或 MQCONNX 调用将返回 2058
- *ConnectionName* 是对通信协议 (*TransportType*) 定义的服务器的名称。 它必须是标准网络名，例如 AMACHINE.ACOMPANY.COM(1414)。
- *ConnectionName* 可以是连接名称的逗号分隔列表。 列表中的连接名称以类似于客户机连接表中的多个连接的方式使用。 连接名称列表可用作队列管理器组的替代方法，以指定客户机要尝试的多个连接。 如果要配置多实例队列管理器，那么可以使用连接名称列表来指定不同的队列管理器实例。

- 要取消 **MQSERVER** 并返回到 **MQCHLLIB** 和 **MQCHLTAB** 指向的客户机通道定义表，请输入以下命令：
  - Linux AIX 在 AIX and Linux 上：

```
unset MQSERVER
```

  - Windows 在 Windows 上：

```
SET MQSERVER=
```

示例

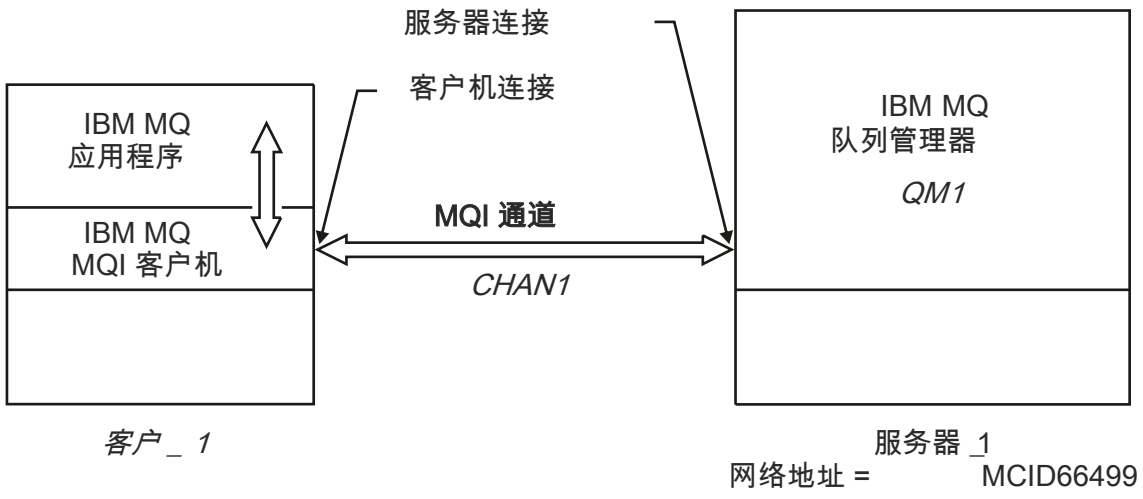


图 *1:* 简单通道定义的示例

要创建 第 32 页的图 1 中显示的简单通道定义，请使用以下命令:

- Linux AIX 在 AIX and Linux 上:

```
export MQSERVER=CHANNEL1/TCP/'MCID66499'
```

- Windows 在 Windows 上:

```
SET MQSERVER=CHANNEL1/TCP/MCID66499
```

注: 有关如何更改 TCP/IP 端口号的信息，请参阅 第 33 页的『TCP/IP 缺省端口』。

以下是简单通道定义的更多示例:

- Windows 在 Windows 上:

```
SET MQSERVER=CHANNEL1/TCP/9.20.4.56
SET MQSERVER=CHANNEL1/NETBIOS/BOX643
```

- Linux AIX 在 AIX and Linux 上:

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56'
export MQSERVER=CHANNEL1/LU62/BOX99
```

其中 BOX99 是 LU 6.2 ConnectionName。

- IBM i 在 IBM i 上:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('CHANNEL1/TCP/9.20.4.56(1416)')
```

在 IBM MQ MQI client 上，所有 **MQCONN** 或 **MQCONNX** 请求都尝试使用已定义的通道，除非在提供给 **MQCONNX** 的 MQCNO 结构中引用的 MQCD 结构中覆盖该通道。

相关任务

第 53 页的『使用 IBM MQ 环境变量』
您可以使用命令来显示当前设置或重置 IBM MQ 环境变量的值。

第 34 页的『使用 MQCNO 在 IBM MQ MQI client 上创建客户机连接通道』
您可以使用 MQCONNX 调用上的 MQCNO 结构在客户机工作站上定义客户机连接通道。

TCP/IP 缺省端口

缺省情况下，对于 TCP/IP， IBM MQ 假定通道将连接到端口 1414。

您可以通过以下方法更改此操作:

- 将括号中的端口号添加为 ConnectionName:的最后一部分:
  - Linux AIX 在 AIX and Linux 上:

```
export  MQSERVER='ChannelName/TransportType/ConnectionName(PortNumber)'
```

  - Windows 在 Windows 上:

```
SET  MQSERVER=ChannelName/TransportType/ConnectionName(PortNumber)
```

- 通过将端口号添加到协议名称来更改 `mqclient.ini` 文件，例如:

```
TCP:
port=2001
```

- 将 IBM MQ 添加到服务文件，如 第 213 页的『在 AIX and Linux 上使用 TCP/IP 侦听器』中所述。

SPX 缺省套接字

缺省情况下，对于 SPX， IBM MQ 假定通道将连接到套接字 5E86。

您可以通过以下方法更改此操作:

- 将括号中的套接字编号添加为 ConnectionName:的最后一部分:

```
SET  MQSERVER=ChannelName/TransportType/ConnectionName(SocketNumber)
```

对于 SPX 连接，请以 `network.node(socket)` 格式指定 ConnectionName 和套接字。 如果 IBM MQ 客户机和服务器位于同一网络上，那么无需指定网络。 如果您正在使用缺省套接字，那么无需指定套接字。

- 通过将端口号添加到协议名称来更改 `qm.ini` 文件，例如:

```
SPX:
socket=5E87
```

使用 MQCNO 在 IBM MQ MQI client 上创建客户机连接通道

您可以使用 MQCONNX 调用上的 MQCNO 结构在客户机工作站上定义客户机连接通道。

关于此任务

IBM MQ MQI client 应用程序可以在 **MQCONNX** 调用上使用连接选项结构 MQCNO 来引用包含客户机连接通道定义的通道定义结构 MQCD。

通过这种方式，客户机应用程序可以在运行时指定通道的 **ChannelName**， **TransportType** 和 **ConnectionName** 属性，从而使客户机应用程序能够同时连接到多个服务器队列管理器。

请注意，如果使用 **MQSERVER** 环境变量定义通道，那么无法在运行时指定 **ChannelName**，**TransportType** 和 **ConnectionName** 属性。

客户机应用程序还可以指定通道的属性，例如 **MaxMsgLength** 和 **SecurityExit**。 指定此类属性使客户机应用程序能够为非缺省值的属性指定值，并允许在 MQI 通道的客户机端调用通道出口程序。

如果通道使用传输层安全性 (TLS)，那么客户机应用程序还可以在 MQCD 结构中提供与 TLS 相关的信息。可以在 TLS 配置选项结构 MQSCO 中提供与 TLS 相关的其他信息，此结构也由 **MQCONNX** 调用上的 MQCNO 结构引用。

有关 MQCNO， MQCD 和 MQSCO 结构的更多信息，请参阅 MQCNO， MQCD 和 MQSCO。

注: MQCONNX 的样本程序称为 **amqscnxc**。 另一个名为 **amqssslc** 的样本程序演示 MQSCO 结构的使用。

相关任务

第 31 页的『使用 MQSERVER 在 IBM MQ MQI client 上创建客户机连接通道』
您可以使用 **MQSERVER** 环境变量在客户机工作站上定义客户机连接通道。

在服务器上创建服务器连接和客户机连接定义

您可以在服务器上创建这两个定义，然后使客户机连接定义可供客户机使用。

关于此任务

首先定义服务器连接通道，然后定义客户机连接通道:

- 在所有平台上，都可以使用 IBM MQ Script (MQSC) 命令，可编程命令格式 (PCF) 命令在服务器上定义服务器连接通道。
- Windows Linux 在 Linux 和 Windows 上，还可以使用 IBM MQ Explorer。
- z/OS 在 z/OS 上，还可以使用 "操作和控制" 面板。
- IBM i 在 IBM i 上，您还可以使用面板界面。

使用客户机通道定义表 (CCDT) 使在服务器上创建的客户机连接通道定义可供客户机使用。

过程

1. 要定义服务器连接通道，请参阅 第 46 页的『在服务器上定义服务器连接通道』。
2. 要定义客户机连接通道，请参阅 第 47 页的『在服务器上定义客户机连接通道』。

相关任务

第 36 页的『配置二进制格式 CCDT』
客户机通道定义表 (CCDT) 确定客户机应用程序用于连接到队列管理器的通道定义和认证信息。 在多平台上，创建队列管理器时会自动创建包含缺省设置的二进制 CCDT。 您可以使用 **runmqsc** 命令来更新二进制 CCDT。

第 46 页的『在服务器上定义服务器连接通道』
为队列管理器创建服务器连接通道定义。

第 47 页的『在服务器上定义客户机连接通道』
定义了服务器连接通道后，现在定义相应的客户机连接通道。

第 48 页的『访问客户机连接通道定义』
通过复制或共享客户机应用程序，然后在客户机计算机上指定其位置和名称，可以使客户机通道定义表 (CCDT) 可供客户机应用程序使用。 从 IBM MQ 9.0 开始，产品还提供了通过 URL 查找客户机通道定义表 (CCDT) 的能力。

配置客户机通道定义表

客户机通道定义表 (CCDT) 定义客户机连接通道及其属性。 客户机读取此文件以确定要连接到的队列管理器。 CCDT 文件可以是 JSON 或二进制格式。

关于此任务

队列管理器未读取 CCDT 文件。 它仅用于向客户机提供通道定义和认证信息。

V 9.2.0 在 IBM MQ 9.2.0 之前， CCDT 仅以二进制格式提供。 从 IBM MQ 9.2.0 开始，您还可以创建 JavaScript 对象表示法 (JSON) 格式的 CCDT。

创建队列管理器时，将自动创建二进制格式 CCDT。 仅使用 **runmqsc** 命令更新存储在此表中的客户机通道定义。

V 9.2.0 JSON 格式 CCDT 是具有 .json 扩展名的平面文本文件。 您可以手动创建和更新此表，这比使用 **runmqsc** 命令限制更少。

z/OS 在应用程序服务器中运行的 z/OS JMS 客户机使用 CCDT 来引用远程队列管理器连接详细信息。 从 IBM MQ for z/OS 9.1 开始， IBM MQ Advanced for z/OS Value Unit Edition 允许 JMS 客户机远程连接到其他 z/OS LPAR 上的队列管理器。 因此，这些客户机也可以使用 CCDT。

要帮助您配置 CCDT 以使用客户机，请从以下任务中进行选择:

过程

- 第 36 页的『配置二进制格式 CCDT』
- V 9.2.0
  第 37 页的『配置 JSON 格式 CCDT』
- 第 44 页的『CDT 的位置』
- 第 45 页的『对 CCDT 的 URL 访问权』

相关任务

第 335 页的『配置统一集群』
统一集群允许应用程序针对规模和可用性进行设计，并且可以连接到该统一集群中的任何队列管理器。

相关参考

MQI 客户机: 客户机通道定义表 (CCDT)

配置二进制格式 *CCDT*

客户机通道定义表 (CCDT) 确定客户机应用程序用于连接到队列管理器的通道定义和认证信息。 在多平台上，创建队列管理器时会自动创建包含缺省设置的二进制 CCDT。 您可以使用 **runmqsc** 命令来更新二进制 CCDT。

开始之前

V 9.2.0 从 IBM MQ 9.1.2 开始，您还可以使用 JavaScript 对象表示法 (JSON) 格式创建 CCDT，与使用二进制 CCDT 相比，使用此备用格式具有一些优点。 请参阅 第 37 页的『配置 JSON 格式 CCDT』。

所有平台上的客户机都可以查看和使用 CCDT。 但是，只能在 IBM MQ for Multiplatforms 下创建和修改二进制 CCDT。

关于此任务

Multi 在 多平台 上：

- 将在队列管理器的数据目录下的 `@ipcc` 目录中自动创建二进制 CCDT。
- 除了自动创建外，与队列管理器关联的二进制 CCDT 还与对象定义保持同步。 定义，变更或删除客户机通道对象时，将在同一操作中同时更新队列管理器对象定义和 CCDT 中的条目。

  注意:

  – IBM MQ CCDT 文件的设计是，仅在实际定义了用户定义的所有客户机连接通道之后，才会缩减 CCDT 文件。 当删除客户机连接通道时，它只是在 CCDT 文件中标记为已删除，但在物理上未将其除去。

  – 要强制 CCDT 文件缩小，请在删除一个或多个客户机连接通道后发出以下命令:

  ```
  rcrmqobj -m QM80 -t clchltab
  ```

- 您可以使用 **runmqsc** 命令来更改二进制 CCDT 的位置和内容。

所有平台上的客户机都可以查看和使用二进制 CCDT。

过程

- Multi

  创建缺省二进制 CCDT。

  在 多平台上，创建队列管理器时将创建名为 `AMQCLCHL.TAB` 的缺省二进制 CCDT。

  缺省情况下， AMQCLCHL.TAB 位于服务器上的以下目录中:

  – IBM i 在 IBM i 上，在集成文件系统中:

  ```
  /QIBM/UserData/mqm/qmgrs/QUEUEMANAGERNAME/&ipcc
  ```

  – Linux AIX 在 AIX and Linux 系统上:

  /prefix/qmgrs/*QUEUEMANAGERNAME*/@ipcc

  QUEUEMANAGERNAME 引用的目录的名称在 AIX and Linux 系统上区分大小写。 如果队列管理器名称中包含特殊字符，那么目录名称可能与队列管理器名称不同。

  – Windows 在 Windows 上:

  MQ_INSTALLATION_PATH\data\qmgrs*QUEUEMANAGERNAME*\@ipcc

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

  但是，您可能已选择对队列管理器数据使用其他目录。 您可以在使用 **crtmqm** 命令时指定参数 **-md** *DataPath* 。 如果执行此操作，那么 `AMQCLCHL.TAB` 位于您指定的 *DataPath* 的 `@ipcc` 目录中。

- 找到 CCDT:

- 在客户端计算机上
- 在多个客户机共享的位置
- 在服务器上作为共享文件

请参阅 第 44 页的『CDT 的位置』。

a) 直接在客户机上创建二进制 CCDT。

- 将 runmqsc 命令与 **-n** 参数配合使用。
- 将在 **MQCHLLIB** 指示的位置中创建 CCDT，并使用 **MQCHLTAB** 指示的文件名 (缺省情况下为 AMQCLCHL.TAB) 创建 CCDT。
- **重要信息:** 如果指定 **-n** 参数，那么不得指定任何其他参数。

b) 更改位置。

您可以通过设置 **MQCHLLIB** 来更改 CCDT 的路径。 请记住，如果您在同一服务器上具有多个队列管理器，那么它们将共享同一个 CCDT 位置。

- 访问 CCDT

  您可以访问 CCDT:

  - 通过定义 **MQCCDTURL** 环境变量从文件， FTP 或 HTTP URL 远程访问。
  - 通过在本地设置 **MQCHLLIB** 和 **MQCHLTAB** 环境变量。
  - 在本地，通过在客户机配置文件中定义 CHANNELS 节的 **ChannelDefinitionDirectory** 和 **ChannelDefinitionFile** 属性。

  请参阅 第 44 页的『CDT 的位置』 以获取各种示例。
- 查看或编辑 CCDT 内容。

  您可以使用 **runmqsc** 命令来查看 CCDT 内容:

  1. 将环境变量设置为 访问 CCDT
  2. 运行命令 `runmqsc -n`
  3. 运行命令 DISPLAY CHANNEL (*)，例如

  Multi 在 多平台上，您还可以使用 **runmqsc** 命令编辑二进制 CCDT 内容。 CCDT 的每个条目表示与特定队列管理器的客户机连接。 当您使用 **DEFINE CHANNEL** 命令定义客户机连接通道时，将添加一个新条目，当您使用 **ALTER CHANNEL** 命令变更客户机连接通道时，将更新该条目。 有关使用该命令的更多示例，请参阅 **runmqsc** 。
- 为客户机提供用于检查 TLS 证书撤销的认证信息。

  a) 定义包含认证信息对象的名称列表。

  b) 将队列管理器属性 **SSLCRLNL** 设置为名称列表的名称。

相关概念

使用已撤销证书

相关任务

第 37 页的『配置 JSON 格式 CCDT』

客户机通道定义表 (CCDT) 确定客户机应用程序用于连接到队列管理器的通道定义和认证信息。 使用文本编辑器来创建和更新 JavaScript 对象表示法 (JSON) CCDT。

V 9.2.0 配置 *JSON* 格式 *CCDT*

客户机通道定义表 (CCDT) 确定客户机应用程序用于连接到队列管理器的通道定义和认证信息。 使用文本编辑器来创建和更新 JavaScript 对象表示法 (JSON) CCDT。

开始之前

Multi 如果您正在使用 IBM MQ for Multiplatforms，那么可以改为使用在创建队列管理器时自动创建的二进制 CCDT。 请参阅 第 36 页的『配置二进制格式 CCDT』。

关于此任务

JSON 格式的 CCDT 模式的文件名为:

Linux

```
/opt/mqm/lib/ccdt_schema.json
```

Windows

```
C:\Program Files\IBM\MQ\bin\ccdt_schema.json
```

没有缺省 JSON CCDT， IBM MQ 未提供任何工具来创建或编辑 JSON 格式的 CCDT。 尽管如此，在手动开发 JSON CCDT 时，与使用 **runmqsc** 命令来处理二进制 CCDT 时相比，您有更多配置选项:

- 您无需使用 IBM MQ for Multiplatforms 来创建和编辑 JSON CCDT 文件。
- 通过使用 JSON 格式，可以定义同名的重复通道定义。 在云上部署 IBM MQ 时，可以使用此方法使部署具有可伸缩性和高可用性。
- JSON 文件是人类可读的，可简化队列管理器配置。
- 平面文件格式可与以下内容集成:
  - 用于跟踪 CCDT 历史记录的版本控制工具
  - 持续交付中的自动化工具
- 您不需要专门的工具来维护 CCDT 文件。
- 文件较小。
- 此格式提供向后和向前兼容性。

注意:

1. JSON 标准将重复键视为有效，但是， JSON 解析器仅在分配属性时采用重复键的最后一个读取值。 因此，定义重复通道时，每个通道都必须是分配给 "channel" 键的数组值的元素。
2. JSON CCDT 中的键值和属性值**必须**完全按照模式中的规定提供。 特别要注意的是，JSON 解析是区分大小写的。
3. JSON CCDT 不支持为证书撤销列表 (CRL) 和联机证书状态协议 (OCSP) 响应程序位置信息存储轻量级目录访问协议 (LDAP) 服务器位置。

表 7: 按平台划分的编码需求

平台	JMS 客户机编码	C 客户机编码
IBM i IBM i	ASCII	EBCDIC
ALW AIX, Linux, and Windows	ASCII	ASCII
z/OS z/OS	ASCII 或 EBCDIC	不适用

注意: 通过 JSON CCDT (包括不包含所有属性的 稀疏) 为通道提供任何定义时，将使用 JSON 中未指定的任何内容的缺省值来构造完整的通道定义。

因此，您必须为不需要缺省值的每个属性提供特定值。

过程

- 创建 JSON CCDT

  a) 使用通用文本编辑器创建具有 `.json` 扩展名的平面文件。

  b) 定义 CCDT。

  请参阅 第 41 页的『JSON CCDT 示例』 和 第 39 页的『JSON CCDT 支持的通道属性』。
- 找到 CCDT:
  - 在客户端计算机上

- 在多个客户机共享的位置
- 在服务器上作为共享文件

请参阅 第 44 页的『CDT 的位置』。

- 验证 JSON CCDT

  使用 JSON linter 针对模式验证 CCDT。

  请参阅 如何根据模式验证 IBM MQ CCDT JSON 文件，以获取有关如何创建具有两个通道的 CCDT 文件并验证其是否有效的信息。

  CCDT 模式包含在产品和客户机包中:

  - Linux AIX 在 AIX and Linux 系统上:

    产品软件包和客户机软件包中分别包含 `$MQ_INSTALLATION_PATH/lib` 和 `/lib`。

  - Windows 在 Windows 上:

    产品软件包和客户机软件包中分别包含 `%MQ_INSTALLATION_PATH%\bin` 和 `\bin`。

  注意:

  - JSON 指针联机可用。
  - 模式使用 "`required`" 键定义必需属性。
  - 模式使用 "`type`" 键定义属性数据类型。

- 访问 CCDT

  您可以访问 CCDT:

  - 通过定义 **MQCCDTURL** 环境变量从文件， FTP 或 HTTP URL 远程访问。
  - 通过在本地设置 **MQCHLLIB** 和 **MQCHLTAB** 环境变量。
  - 在本地，通过在客户机配置文件中定义 CHANNELS 节的 **ChannelDefinitionDirectory** 和 **ChannelDefinitionFile** 属性。

  请参阅 第 44 页的『CDT 的位置』 以获取各种示例。

- 查看或编辑 CCDT 内容

  CCDT 的每个条目表示与特定队列管理器的客户机连接。 您可以使用文本编辑器来查看或编辑 CCDT 内容。

  如果您只想查看 CCDT，那么还可以使用 **runmqsc** 命令执行此操作，如下所示:

  1. 设置环境变量以允许您访问 CCDT，如上一步中所述。
  2. 运行命令 `runmqsc -n`。 有关更多信息，请参阅 runmqsc。
  3. 运行 **DISPLAY CHANNEL** 命令。 例如，运行 `DISPLAY CHANNEL(*)`。

相关概念

使用已撤销证书

相关任务

第 36 页的『配置二进制格式 CCDT』

客户机通道定义表 (CCDT) 确定客户机应用程序用于连接到队列管理器的通道定义和认证信息。 在多平台上，创建队列管理器时会自动创建包含缺省设置的二进制 CCDT。 您可以使用 **runmqsc** 命令来更新二进制 CCDT。

第 335 页的『配置统一集群』

统一集群允许应用程序针对规模和可用性进行设计，并且可以连接到该统一集群中的任何队列管理器。

V 9.2.0 *JSON CCDT* 支持的通道属性

JSON CCDT 支持的客户机连接通道属性的列表。 此列表是二进制 CCDT 支持的属性的子集。

属性映射

这些属性将插入到以下通道对象中:

```
{ "channel": [ {$CHANNEL_1_KEY_VALUE_LIST },..., {$CHANNEL_N_KEY_VALUE_LIST } ] }
```

其中 $CHANNEL_X_KEY_VALUE_LIST 是下表中列出的属性的逗号分隔列表。

请参阅 第 41 页的『JSON CCDT 示例』 以获取基本用例。

JSON 模式在 /opt/mqm/lib/ccdt_schema.json 中提供。 要发现哪些值对每个属性有效，请查看 JSON 模式。

下表列出了 JSON 对象，键和数据类型以及相应的二进制通道属性定义。

注意: 必需属性为通道 **name** 和通道 **type**。 如果还定义了 **portRange**，那么还需要 *low* 和 *high* 属性。

JSON 对象	JSON 密钥	JSON 数据类型	二进制属性定义
通道 (数组)	名称	字符串	通道
通道 (数组)	类型	字符串	CHLTYPE
channel.clientConnection	queueManager	字符串	QMNAME
channel.clientConnection.connection （数组）	主机	字符串	CONNAME
channel.clientConnection.connection	port	Int	CONNAME
channel.compression.header (数组)	头	字符串	COMPHDR
channel.compression.message (数组)	消息	字符串	COMPMSG
channel.connectionManagement	亲缘关系 (affinity)	字符串	AFFINITY
channel.connectionManagement	clientWeight	Int	CLNTWGHT
channel.connectionManagement	defaultReconnect	字符串	DEFRECON
channel.connectionManagement	disconnectInterval	Int	DISCINT
channel.connectionManagement	heartbeatInterval	Int	HBINT
channel.connectionManagement	keepAliveInterval	Int	KAINT
channel.connectionManagement	sharingConversations	Int	SHARECNV
channel.connectionManagement.localAddress (数组)	主机	字符串	LOCLADDR
channel.connectionManagement.localAddress (数组)	port	Int	LOCLADDR
channel.connectionManagement.localAddress.portRange	high	Int	LOCLADDR
channel.connectionManagement.localAddress.portRange	low	Int	LOCLADDR
channel.exits.receive (数组)	名称	字符串	RCVEXIT
channel.exits.receive (数组)	userData	字符串	RCVDATA
channel.exits.security	名称	字符串	SCYEXIT
channel.exits.security	userData	字符串	SCYDATA
channel.exits.send (数组)	名称	字符串	SENDEXIT

JSON 对象	JSON 密钥	JSON 数据类型	二进制属性定义
channel.exits.send (数组)	userData	字符串	SENDDATA
channel.general	description	字符串	DESCR
channel.general	maximumMessageLength	Int	MAXMSGL
channel.timestamps	变更时间	字符串	ALTDATE 和 ALTTIME
channel.transmissionSecurity	certificateLabel	字符串	CERTLABL
channel.transmissionSecurity	certificatePeerName	字符串	SSLPEER
channel.transmissionSecurity	cipherSpecification	字符串	SSLCIPH

注意:

- 可以将 `channel.connectionManagement.localAddress` 定义为下列其中一个键组合:
  - 主机和端口
  - 主机和 portRange
  - port
  - portRange
- `channel.timestamps altered` JSON 键是可选的，如果未定义，那么该值缺省为 JSON CCDT 文件上次修改时间。 但是，如果环境配置为从 URL 访存 CCDT ，那么缺省值为上次下载文件的时间。
- `channel.clientConnection.connection` 必须同时包含主机和端口密钥。
- 更改后的键是单个字符串，用于封装 ALTDATE 和 ALTTIME 属性。
- 传输类型只能是 TCP ，因此未在模式中定义以下属性:
  - **TRPTYPE**
  - **USERID**
  - **PASSWORD**
  - **MODENAME**
  - **TPNAME**

相关参考

通道类型的通道属性

V 9.2.0 *JSON CCDT 示例*

使用本主题中列出的示例作为需求的基础。

打开通用文本编辑器并复制下列其中一个示例:

- 第 41 页的『定义简单客户机连接』
- 第 42 页的『使用 TLS 定义一个通道和一个队列管理器』
- 第 42 页的『定义一个通道和一个不使用 TLS 的队列管理器』
- 第 42 页的『定义两个同名通道』
- 第 43 页的『客户机连接通道的 CCDT 通道属性定义的完整列表』

定义简单客户机连接

```
{
  "channel":
  [
```

```
    {
      "general":
      {
        "description": "a channel"
      },
      "name": "channel",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "localhost",
            "port": 1414
          }
        ],
        "queueManager": "QM1"
      },
      "type": "clientConnection"
    }
  ]
}
```

使用 TLS 定义一个通道和一个队列管理器

```
{
  "channel": [
    {
      "name": "SSL.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "aztlan1.fyre.ibm.com",
            "port": 1419
          }
        ],
        "queueManager": "QM92TLS"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "TLS_AES_128_GCM_SHA256",
        "certificateLabel": "ibmwebspheremqadministrator",
      },
      "type": "clientConnection"
    }
  ]
}
```

定义一个通道和一个不使用 TLS 的队列管理器

```
{
  "channel": [
    {
      "name": "SYSTEM.DEF.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "aztlan1.fyre.ibm.com",
            "port": 1414
          }
        ],
        "queueManager": "QM92"
      },
      "type": "clientConnection"
    }
  ]
}
```

定义两个同名通道

每个通道都连接到两个不同的队列管理器:

```
{
  "channel":
  [
```

```
    {
      "general":
      {
        "description": "First channel"
      },
      "name": "channel",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "localhost",
            "port": 1414
          }
        ],
        "queueManager": "QM1"
      },
      "type": "clientConnection"
    },
    {
      "general":
      {
        "description": "Second channel"
      },
      "name": "channel",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "localhost",
            "port": 1415
          }
        ],
        "queueManager": "QM2"
      },
      "type": "clientConnection"
    }
  ]
}
```

客户机连接通道的 CCDT 通道属性定义的完整列表

```
{
  "channel":
  [
    {
      "compression":
      {
        "header": [ "system" ],
        "message": [ "zlibfast" ]
      },
      "connectionManagement":
      {
        "sharingConversations": 10,
        "clientWeight": 1,
        "affinity": "none",
        "defaultReconnect": "yes",
        "heartbeatInterval": 600,
        "keepAliveInterval": -1,
        "localAddress":
        [
          {
            "portRange":
            {
              "low": 2020,
              "high": 3030
            }
          }
        ]
      },
      "exits":
      {
        "receive":
        [
          {
          "name": "",
          "userData": ""
          }
```

```
        ],
        "security":
        {
          "name": "",
          "userData": ""
        },
        "send":
        [
           {
             "name": "",
             "userData": ""
           }
        ]
      },
      "general":
      {
        "description": "First channel",
        "maximumMessageLength": 4194304
      },
      "name": "the_channel",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "localhost",
            "port": 1414
          }
        ],
        "queueManager": "QM1"
      },
      "timestamps":
      {
        "altered": "2018-12-04T15:37:22.000Z"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "",
        "certificateLabel": "",
        "certificatePeerName": ""
      },
      "type": "clientConnection"
    }
  ]
}
```

相关参考

通道类型的通道属性

按字母顺序排列的通道属性

CDT 的位置

IBM MQ 支持从文件， FTP 或 HTTP URL 检索 CCDT。 您可以将 CCDT 作为共享文件供客户机访问，而它仍然位于服务器上。 或者，您可以通过将 CCDT 复制到个别客户端计算机或通过将 CCDT 复制到多个客户端共享的位置来分发 CCDT。

如果使用 FTP 复制文件，请使用 `bin` 选项来设置二进制方式; 请勿使用缺省 `ASCII` 方式。 无论您选择哪种方法使 CCDT 可用，位置都必须是安全的，以防止对通道进行未经授权的更改。

从 IBM MQ 9.0 开始，可以在可通过 URL 访问的中央位置托管 CCDT ，从而无需针对每个已部署的客户机单独更新 CCDT。 IBM MQ 9.0 添加了本机 (C/C + + ， COBOL 和 RPG) 和非受管 .NET 应用程序从 URL (无论是本地文件， FTP 还是 HTTP 资源) 中提取 CCDT 的功能。

IBM MQ 客户机的缺省高速缓存行为是，仅当文件修改时间与上次检索时不同时，才会拉低 CCDT 文件。 与大多数客户机配置选项一样，可以通过多种方式提供 URL 位置:

- **CCDTUrlPtr** 和 **CCDTUrlOffset** 通过 MQCNO 结构传递到 MQCONNX MQI 调用
- **MQCCDTURL** 环境变量
- `mqclient.ini` 的 Channels 节中的 **ChannelDefinitionDirectory** 属性

支持已认证和未认证的 URL。 以下是一些示例:

```
export MQCCDTURL=ftp://myuser:password@myhost.sample.com//var/mqm/qmgrs/QMGR/@ipcc/AMQCLCHL.TAB
```

```
export MQCCDTURL=http://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc/AMQCLCHL.TAB
```

如果要将此支持与 FTP 或 HTTP 配合使用，那么仍需要在服务器上托管 CCDT 文件，但通过在 IBM MQ 9.0 添加的支持，所有客户机应用程序都可以自动获取对通道定义的更改，而无需手动推送更新或需要在每个客户机上安装联网文件系统。 有关更多信息，请参阅 第 45 页的『对 CCDT 的 URL 访问权』。

如何在客户机上指定 CCDT 的位置

在客户机系统上，可以通过以下方式指定 CCDT 的位置:

- 使用环境变量 MQCHLLIB 来指定表所在的目录，使用 MQCHLTAB 来指定表的文件名。
- 使用客户机配置文件。 在 CHANNELS 节中，使用属性 ChannelDefinitionDirectory 指定表所在的目录，使用 ChannelDefinitionFile 指定文件名。
- 通过为在中央位置托管的 CCDT 提供 URL (文件， FTP 或 HTTP) ，如前所述。

如果都通过客户机配置文件和使用环境变量指定位置，那么环境变量优先。 您可以使用此功能在客户机配置文件中指定标准位置，并在必要时使用环境变量覆盖该位置。

如果使用 URL 来提供 CCDT 的位置，那么本机客户机应用程序查找客户机通道定义的优先顺序如 第 45 页的『对 CCDT 的 URL 访问权』中所述。

对 *CCDT* 的 *URL* 访问权

您可以在可通过 URL 访问的中心位置托管客户机通道定义表 (CCDT) ，从而无需为每个已部署的客户机单独更新 CCDT。

从 IBM MQ 9.0 开始，可以通过以下任一方式通过 URL 来查找客户机通道定义表:

- 通过使用 MQCNO 进行编程
- 使用环境变量 用于连接队列管理器的客户端通道定义使用以下环境变量定位，其优先顺序为: "**MQSERVER**、"**MQCHLLIB** 和 "**MQCHLTAB**"。

  注意: 环境变量选项只能用于作为客户机连接的本机程序，即 C、COBOL 或 C++ 应用程序。 环境变量对 Java、JMS 或受管 .NET 应用程序无任何影响。

  IBM MQ 支持从文件， FTP 或 HTTP URL 检索 CCDT。
- 通过使用 mqclient.ini 文件 CHANNELS 节。

环境变量 **MQCCDTURL** 允许您提供文件， ftp 或 http URL 作为可从中获取客户机通道定义表的单个值。

您还可以使用 **MQCHLLIB** 环境变量指定的目录路径 (或 第 137 页的『客户机配置文件的 CHANNELS 节』中的 **ChannelDefinitionDirectory** 属性指定的路径) 通过文件， ftp 或 http URL 以及现有的本地文件系统目录 (即 /var/mqm) 来查找 CCDT 文件。 请注意， **MQCHLLIB** 值是目录词干，并与 **MQCHLTAB** 结合使用以派生标准 URL。

系统通过 URL 中编码的凭证来支持连接时的基本认证:

已认证的连接

```
export MQCHLLIB=ftp://myuser:password@myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=http://myuser:password@myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
```

未认证的连接

```
export MQCHLLIB=ftp://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=http://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=file:///var/mqm/qmgrs/QMGR/@ipcc
```

注: 与 JMS 一样，如果要使用已认证的连接，就必须提供 URL 中编码的用户名和密码。

现在，本机客户机应用程序查找客户机通道定义的优先顺序为:

1. 由 **ClientConnOffset** 和 **ClientConnPtr** 在 MQCNO 中提供的 MQCD。
2. 由 MQCNO 中的 **CCDTUrlOffset** 和 **CCDTUrlPtr** 提供的 URL。

3. MQSERVER 环境变量。
4. 如果定义了 `mqclient.ini` 文件，并且其中包含 ServerConnectionParms，那么会使用它定义的通道。有关更多信息，请参阅 第 128 页的『IBM MQ MQI client 配置文件， mqclient.ini』 和 **MQCCDTURL**。
5. 第 61 页的『MQCCDTURL』 环境变量。
6. **MQCHLLIB** 和 **MQCHLTAB** 环境变量。
7. 第 137 页的『客户机配置文件的 CHANNELS 节』中的 **ChannelDefinitionDirectory** 和 **ChannelDefinitionFile** 。

要点: 使用 URL 访问 CCDT 文件将始终打开文件的只读副本，即使在使用 `file://` 协议时也是如此。

尝试打开 CCDT 文件进行写访问（例如，在通过客户机使用 **runmqsc** DEFINE CHANNEL 命令时），将返回一条错误消息，指示无法打开该文件进行写访问。

但是，可使用 **runmqsc** 读取通道和认证信息定义。

相关任务

第 48 页的『访问客户机连接通道定义』
通过复制或共享客户机应用程序，然后在客户机计算机上指定其位置和名称，可以使客户机通道定义表 (CCDT) 可供客户机应用程序使用。 从 IBM MQ 9.0 开始，产品还提供了通过 URL 查找客户机通道定义表 (CCDT) 的能力。

第 36 页的『配置二进制格式 CCDT』
客户机通道定义表 (CCDT) 确定客户机应用程序用于连接到队列管理器的通道定义和认证信息。 在多平台上，创建队列管理器时会自动创建包含缺省设置的二进制 CCDT。 您可以使用 **runmqsc** 命令来更新二进制 CCDT。

将 CCDT 与 IBM MQ classes for JMS 配合使用

相关参考

CCDTURL
MQCNO - 连接选项
XMSC_WMQ_CCDTURL

Windows Active Directory 中的客户机连接通道

在支持 Active Directory 的 Windows 系统上， IBM MQ 在 Active Directory 中发布客户机连接通道以提供动态客户机/服务器绑定。

定义客户机连接通道对象时，会将这些对象写入名为 AMQCLCHL.TAB 。 如果客户机连接通道使用 TCP/IP 协议，那么 IBM MQ 服务器还会在 Active Directory 中发布这些协议。 当 IBM MQ 客户机确定如何连接到服务器时，它将使用以下搜索顺序来查找相关的客户机连接通道对象定义:

1. MQCONNX MQCD 数据结构
2. MQSERVER 环境变量
3. 客户机通道定义文件
4. Active Directory

此顺序表示任何当前应用程序不受任何更改影响。 您可以将 Active Directory 中的这些条目视为客户机通道定义文件中的记录，并且 IBM MQ 客户机以相同的方式处理这些条目。 要配置和管理对在 Active Directory 中发布客户机连接通道定义的支持，请使用 `setmqscp` 命令，如 setmqscp 中所述。

在服务器上定义服务器连接通道

为队列管理器创建服务器连接通道定义。

过程

1. 在服务器上，使用所选名称和通道类型 *server-connection* 定义通道。
   例如:

```
DEFINE CHANNEL(CHAN2) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
DESCR('Server-connection to Client_2')
```

2. 使用以下命令以允许对队列管理器进行入站连接访问:

```
SET CHLAUTH(CHAN2) TYPE(ADDRESSMAP) ADDRESS('IP address') MCAUSER('userid')
```

- 其中 SET CHLAUTH 使用上一步中定义的通道的名称。
- 其中 *"IP address"* IP address 是客户机的 IP 地址。
- 其中， *"userid"* 是要提供给通道以用于对目标队列进行访问控制的标识。 此字段区分大小写。

您可以选择使用多个不同的属性来标识入站连接。 此示例使用 IP 地址。 备用属性包括客户机用户标识和 TLS 主题专有名称。 有关更多信息，请参阅 通道认证记录

此通道定义与在服务器上运行的队列管理器相关联。

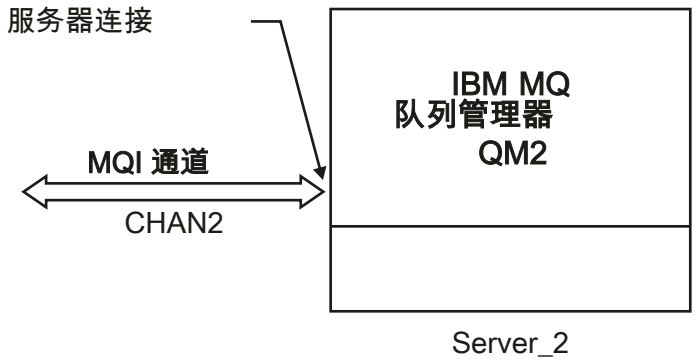


图 *2:* 定义服务器连接通道

在服务器上定义客户机连接通道

定义了服务器连接通道后，现在定义相应的客户机连接通道。

开始之前

定义服务器连接通道。

过程

1. 定义与服务器连接通道同名但通道类型为 *client-connection* 的通道。 必须声明连接名称 (CONNAME)。 对于 TCP/IP ，连接名称是服务器的网络地址或主机名。 还建议指定您希望在客户机环境中运行的 IBM MQ 应用程序连接到的队列管理器名称 (QMNAME)。 通过更改队列管理器名称，您可以定义一组通道以连接到不同的队列管理器。

```
DEFINE CHANNEL(CHAN2) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(9.20.4.26) QMNAME(QM2) DESCR('Client-connection to Server_2')
```

2. 使用以下命令以允许对队列管理器进行入站连接访问:

```
SET CHLAUTH(CHAN2) TYPE(ADDRESSMAP) ADDRESS('IP-address') MCAUSER('userid')
```

- 其中 SET CHLAUTH 使用上一步中定义的通道的名称。
- 其中 *"IP address"* 是客户机的 IP 地址。
- 其中， *"userid"* 是要提供给通道以用于对目标队列进行访问控制的标识。 此字段区分大小写。

您可以选择使用多个不同的属性来标识入站连接。 此示例使用 IP 地址。 备用属性包括客户机用户标识和 TLS 主题专有名称。 有关更多信息，请参阅 通道认证记录

结果

Multi在 多平台上，此通道定义存储在称为客户机通道定义表 (CCDT) 的文件中，该文件与队列管理器相关联。 客户机通道定义表可以包含多个客户机连接通道定义。 有关客户机通道定义表的更多信息，以及有关如何在 z/OS 上存储客户机连接通道定义的相应信息，请参阅 第 36 页的『配置二进制格式 CCDT』。

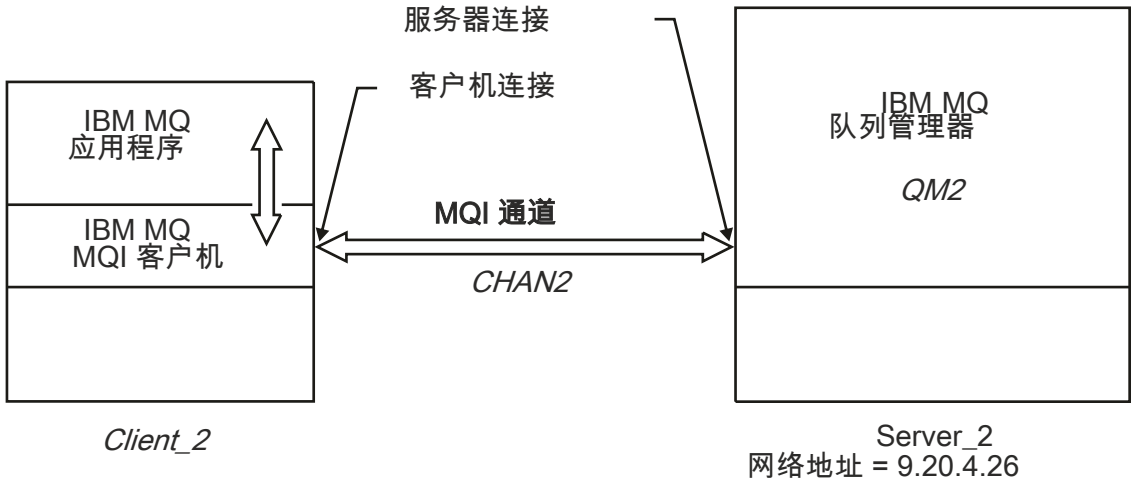


图 *3:* 定义客户机连接通道

访问客户机连接通道定义

通过复制或共享客户机应用程序，然后在客户机计算机上指定其位置和名称，可以使客户机通道定义表 (CCDT) 可供客户机应用程序使用。 从 IBM MQ 9.0 开始，产品还提供了通过 URL 查找客户机通道定义表 (CCDT) 的能力。

开始之前

此任务假定您已在 CCDT 中定义所需的客户机连接通道。 请参阅 第 35 页的『配置客户机通道定义表』。

关于此任务

要使客户机应用程序使用客户机通道定义表 (CCDT)，必须使 CCDT 可供其使用并指定其位置和名称。 有几种方法可以做到这一点:

- 您可以将 CCDT 复制到客户端计算机。
- 您可以将 CCDT 复制到多个客户机共享的位置。
- 您可以将 CCDT 作为共享文件供客户机访问，而它仍然位于服务器上。

从 IBM MQ 9.0， IBM MQ，本机 (C/C + +， COBOL 和 RPG) 和非受管 .NET 应用程序可以从 URL (无论是本地文件， ftp 还是 http 资源) 中提取集中位置中托管的 CCDT。

过程

1. 通过下列其中一种方法使 CCDT 可供客户机应用程序使用:
   a) 可选： 将 CCDT 复制到客户端计算机。
   b) 可选： 将 CCDT 复制到由多个客户机共享的位置。
   c) 可选： 将 CCDT 保留在服务器上，但使其可由客户机共享。
   d) 可选： 为中央位置托管的 CCDT 定义本地文件， ftp 或 http URL，以便本机 (C/C + +， COBOL 和 RPG) 和非受管 .NET 应用程序可以从此 URL 中提取 CCDT。

   无论您为 CCDT 选择哪个位置，该位置都必须是安全的，以防止对通道进行未经授权的更改。
2. 在客户机上，通过以下三种方法之一指定包含 CCDT 的文件的位置和名称:

a) 可选： 使用客户机配置文件的 CHANNELS 节。 有关更多信息，请参阅第 137 页的『客户机配置文件的 CHANNELS 节』。

b) 可选： 使用环境变量 MQCHLLIB 和 MQCHLTAB。

例如，可以通过输入以下命令来设置环境变量:

- 在 AIX and Linux 系统:

```
export MQCHLLIB= MQ_INSTALLATION_PATH/qmgrs/ QUEUEMANAGERNAME /@ipcc
export MQCHLTAB=AMQCLCHL.TAB
```

- IBM i 在 IBM i 系统上:

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE('/QIBM/UserData/mqm/qmgrs/QUEUEMANAGERNAME/@ipcc')
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(AMQCLCHL.TAB)
```

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

c) 可选： 仅在 Windows 上，使用 **setmqscp** 控制命令在 Active Directory 中发布客户机连接通道定义。

d) 通过使用 MQCNO 编程，使用环境变量或使用 `mqclient.ini` 文件节，通过 URL 提供集中托管的 CCDT 的位置。 有关更多信息，请参阅第 44 页的『CDT 的位置』和第 45 页的『对 CCDT 的 URL 访问权』。

如果设置了 MQSERVER 环境变量，那么 IBM MQ 客户机将优先使用 MQSERVER 指定的客户机连接通道定义，而不是客户机通道定义表中的任何定义。

相关概念

第 45 页的『对 CCDT 的 URL 访问权』
您可以在可通过 URL 访问的中心位置托管客户机通道定义表 (CCDT)，从而无需为每个已部署的客户机单独更新 CCDT。

相关任务

第 36 页的『配置二进制格式 CCDT』
客户机通道定义表 (CCDT) 确定客户机应用程序用于连接到队列管理器的通道定义和认证信息。 在多平台上，创建队列管理器时会自动创建包含缺省设置的二进制 CCDT。 您可以使用 **runmqsc** 命令来更新二进制 CCDT。

相关参考

MQI 客户机: 客户机通道定义表 (CCDT)

ALW MQI 通道的通道出口程序

AIX, Linux, and Windows 上的 IBM MQ MQI client 环境提供了三种类型的通道出口。

这些字段为:

- 发送出口
- 接收出口
- 安全出口

这些出口在客户机和通道的服务器端都可用。 如果您正在使用 MQSERVER 环境变量，那么出口不可用于应用程序。 在 消息传递通道的通道出口程序中说明了通道出口。

发送和接收出口协同工作。 您可以使用以下几种可能的方法:

- 拆分和重新组装消息
- 压缩和解压缩消息中的数据 (此功能作为 IBM MQ 的一部分提供，但您可能希望使用其他压缩技术)
- 对用户数据进行加密和解密 (此功能作为 IBM MQ 的一部分提供，但您可能希望使用其他加密技术)
- 对发送和接收的每条消息进行日志记录

您可以使用安全出口来确保正确识别 IBM MQ 客户机和服务器，并控制访问。

如果通道实例的服务器端的发送或接收出口需要对与其关联的连接执行 MQI 调用，那么它们将使用 MQCXP Hconn 字段中提供的连接句柄。 您必须知道客户机连接发送和接收出口无法进行 MQI 调用。

相关概念

第 50 页的『客户机连接上的安全出口』
您可以使用安全出口程序来验证通道另一端的合作伙伴是否真实。 将安全出口应用于客户机连接时适用特殊注意事项。

用户出口、API 出口和 IBM MQ 可安装服务

相关任务

扩展队列管理器设施

相关参考

第 50 页的『出口路径』
在客户机配置文件中定义了通道出口位置的缺省路径。 通道出口在初始化通道时装入。

第 51 页的『在发送或接收出口程序中标识 API 调用』
将 MQI 通道用于客户机时，代理缓冲区的字节 10 标识调用发送或接收出口时正在使用的 API 调用。 这对于确定哪些通道流包含用户数据很有用，可能需要加密或数字签名之类的处理。

ALW 出口路径

在客户机配置文件中定义了通道出口位置的缺省路径。 通道出口在初始化通道时装入。

在 AIX, Linux, and Windows 系统上，将在安装 IBM MQ MQI client 期间向系统添加客户机配置文件。 在此文件中使用以下节定义了客户机上通道出口位置的缺省路径:

```
ClientExitPath:
ExitsDefaultPath= string
ExitsDefaultPath64= string
```

其中 *string* 是适合于平台的格式的文件位置

初始化通道时，在 MQCONN 或 MQCONNX 调用之后，将搜索客户机配置文件。 读取 ClientExitPathstanza，并加载通道定义中指定的任何通道出口。

ALW 客户机连接上的安全出口

您可以使用安全出口程序来验证通道另一端的合作伙伴是否真实。 将安全出口应用于客户机连接时适用特殊注意事项。

第 51 页的图 4 说明了在客户机连接中使用安全出口，使用 IBM MQ 对象权限管理器来认证用户。 SecurityParmsPtr 或 SecurityParmsOffset 是在客户机上的 MQCNO 结构中设置的，并且通道两端都有安全出口。 在正常安全消息交换结束，并且通道已准备好运行之后，从 MQCXP SecurityParms 字段访问的 MQCSP 结构将传递到客户机上的安全出口。 出口类型设置为 MQXR_SEC_PARMS。 安全出口可以选择对用户标识和密码不执行任何操作，也可以更改其中之一或两者。 然后将从出口返回的数据发送到通道的服务器连接端。 MQCSP 结构在通道的服务器连接端上重建，并传递到从 MQCXP SecurityParms 字段访问的服务器连接安全出口。 安全出口接收并处理此数据。 此处理通常是为了撤销对客户机出口中的用户标识和密码字段所作的任何更改，然后使用这些更改来授权队列管理器连接。 生成的 MQCSP 结构是使用队列管理器系统上 MQCNO 结构中的 SecurityParmsPtr 引用的。

MQCXP SecurityParms 字段传回的内存地址必须保持可寻址且保持不变，直到 MQXR_TERM。 在针对 MQXR_TERM 调用出口之前，出口不得使内存失效或释放回系统。

如果在 MQCNO 结构中设置了 SecurityParmsPtr 或 SecurityParmsOffset ，并且仅在通道的一端存在安全出口，那么该安全出口将接收并处理 MQCSP 结构。 诸如加密之类的操作不适用于单个用户出口，因为没有用于执行补充操作的出口。

如果未在 MQCNO 结构中设置 SecurityParmsPtr 和 SecurityParmsOffset ，并且在通道的任一端或两端都有安全出口，那么将调用一个或多个安全出口。 任一安全出口都可以返回其自己的 MQCSP 结构 (通过 SecurityParmsPtr 寻址); 直到安全性出口终止 (MQXR_TERM 的 ExitReason) 之后，才会再次调用该安全出口。 出口写程序可以在该阶段释放用于 MQCSP 的内存。

当服务器连接通道实例共享多个对话时，对安全出口的调用模式在第二个和后续对话上受到限制。

对于第一个对话，模式与通道实例未共享对话时的模式相同。 对于第二个和后续对话，从不使用 MQXR_INIT， MQXR_INIT_SEC 或 MQXR_SEC_MSG 调用安全出口。 它是使用 MQXR_SEC_PARMS 调用的。

在具有共享对话的通道实例中，仅针对上次运行的对话调用 MQXR_TERM。

每个对话都有机会在出口的 MQXR_SEC_PARMS 调用中更改 MQCD; 在通道的服务器连接端上，此功能可用来更改值，例如，在连接到队列管理器之前的 MCAUserIdentifier 或 LongMCAUserIdPtr 值。

Server-connection exit	Client-connection exit
	Invoked with MQXR_INIT Responds with MQXCC_OK
Invoked with MQXR_INIT Responds with MQXCC_OK	
	Invoked with MQXR_INIT_SEC Responds with MQXCC_OK
Invoked with MQXR_INIT_SEC Responds with MQXCC_OK	
	Invoked with MQXR_SEC_PARMS Responds with MQXCC_OK
Invoked with MQXR_SEC_PARMS Responds with MQXCC_OK	
Data transfer begins	
Invoked with MQXR_TERM Responds with MQXCC_OK	Invoked with MQXR_TERM Responds with MQXCC_OK

图 *4:* 使用安全性参数的客户机连接的客户机连接发起的协议交换

注: 在 IBM WebSphere MQ 7.1 发行版之前构造的安全出口应用程序可能需要更新。 有关更多信息，请参阅通道安全出口程序。

ALW 在发送或接收出口程序中标识 API 调用

将 MQI 通道用于客户机时，代理缓冲区的字节 10 标识调用发送或接收出口时正在使用的 API 调用。 这对于确定哪些通道流包含用户数据很有用，可能需要加密或数字签名之类的处理。

下表显示了处理 API 调用时出现在通道流的字节 10 中的数据。

注: 这些不是此字节的唯一值。 还有其他 **reserved** 值。

表 *8:* 识别 *API* 调用

API 调用	**请求的字节 10 的值**	**应答的字节 10 的值**
MQCONN 第 52 页的『1』和 第 52 页的『2』	X'81'	X'91'

表 8: 识别 API 调用 (继续)		
API 调用	**请求的字节 10 的值**	**应答的字节 10 的值**
MQ 描述第 52 页的『1』	X'82'	X'92'
MQ 打开第 52 页的『3』	X'83'	X'93'
MQCLOSE	X'84'	X'94'
MQGET 第 52 页的『4』	X'85'	X'95'
MQPUT 第 52 页的『4』	X'86'	X'96'
MQPUT1 请求 第 52 页的『4』	X'87'	X'97'
MQSET 请求	X'88'	X'98'
MQINQ 请求	X'89'	X'99'
MQCMIT 请求	X'8A'	X'9A'
MQBACK 请求	X'8B'	X'9B'
MQSTAT 请求	X'8D'	X'9D'
MQSUB 请求	X'8E'	X'9E'
MQSUBRQ 请求	X'8F'	X'9F'
xa_start 请求	X'A1'	X'B1'
xa_end 请求	X'A2'	X'B2'
xa_open 请求	X'A3'	X'B3'
xa_close 请求	X'A4'	X'B4'
xa_prepare 请求	X'A5'	X'B5'
xa_commit 请求	X'A6'	X'B6'
xa_rollback 请求	X'A7'	X'B7'
xa_算了请求	X'A8'	X'B8'
xa_recover 请求	X'A9'	X'B9'
xa_complete 请求	X'AA'	X'BA'

注意:

1. 客户机与服务器之间的连接由客户机应用程序使用 MQCONN 启动。 因此，特别是对于此命令，还有其他几个网络流。 这同样适用于终止网络连接的 MQDISC。
2. 对于客户机/服务器连接， MQCONNX 的处理方式与 MQCONN 相同。
3. 如果打开大型分发列表，那么每个 MQOPEN 调用可能有多个网络流，以便将所有必需数据传递到 SVRCONN MCA。
4. 大型消息可以超过传输段大小。 如果发生这种情况，那么可能有许多网络流是由单个 API 调用产生的。

z/OS 将客户机连接到队列共享组

通过在作为队列共享组成员的服务器上的客户机与队列管理器之间创建 MQI 通道，可以将客户机连接到队列共享组。

关于此任务

队列共享组由一组可访问同一组共享队列的队列管理器组成。 有关共享队列的更多信息，请参阅 共享队列和队列共享组。

放入共享队列的客户机可以连接到队列共享组的任何成员。 连接到队列共享组的好处是可以增加前端和后端可用性以及增加容量。 您可以连接到特定队列管理器或通用接口。

直接连接到队列共享组中的队列管理器将使您能够将消息放入共享目标队列，这将提高后端可用性。

连接到队列共享组的通用接口将打开与该组中某个队列管理器的会话。 这将提高前端可用性，因为客户机队列管理器可以与组中的任何队列管理器连接。 当您不想连接到队列共享组中的特定队列管理器时，使用通用接口连接到组。

通用接口可以是综合系统分发器 VIPA 地址或 VTAM 通用资源名称，也可以是队列共享组的另一个公共接口。 有关设置通用接口的更多详细信息，请参阅 使用队列共享组为 IBM MQ for z/OS 设置通信。

过程

要连接到队列共享组的通用接口，您需要创建可由该组中的任何队列管理器访问的通道定义。 要执行此操作，必须在组中的每个队列管理器上具有相同的定义。

1. 定义 SVRCONN 通道，如以下示例中所示:

   ```
   DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
   QSGDISP(GROUP)
   ```

   服务器上的通道定义存储在共享 Db2 存储库中。 队列共享组中的每个队列管理器都会生成定义的本地副本，确保在发出 MQCONN 或 MQCONNX 调用时始终连接到正确的服务器连接通道。
2. 定义 CLNTCONN 通道，如以下示例中所示:

   ```
   DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
   CONNAME( VIPA address ) QMNAME(QSG1) +
   DESCR('Client-connection to Queue Sharing Group QSG1') QSGDISP(GROUP)
   ```

结果

由于队列共享组的通用接口存储在客户机连接通道的 CONNAME 字段中，因此您现在可以连接到该组中的任何队列管理器，并将其放入该组拥有的共享队列中。

使用 IBM MQ 环境变量

您可以使用命令来显示当前设置或重置 IBM MQ 环境变量的值。

关于此任务

您可以通过以下方式使用环境变量:

- 设置系统概要文件中的变量以进行永久更改
- 从命令行发出命令以仅对此会话进行更改
- 要为一个或多个变量提供依赖于正在运行的应用程序的特定值，请向应用程序使用的命令脚本文件添加命令

对于每个环境变量，可以使用命令来显示当前设置或重置变量的值。 除非另有声明，否则这些命令在所有受支持的平台上都可用。 命令的格式取决于您的平台。 例如:

- Linux AIX 在 AIX and Linux 上:

  ```
  export [environment variable]=value
  ```

- Windows 在 Windows 上：

```
Set [environment variable]=value
```

- IBM i 在 IBM i 上：

```
ADDENVVAR ENVVAR(environment variable) VALUE(xx)
```

- MQ Appliance 对于 IBM MQ Appliance，请参阅 IBM MQ Appliance 文档中的 在 IBM MQ Appliance 。

如果适用， IBM MQ 会将缺省值用于尚未设置的变量。

注: z/OS IBM MQ for z/OS 不支持任何 IBM MQ 环境变量。 如果要将此平台用作服务器，请参阅 客户机通道定义表 ，以获取有关如何在 z/OS 上生成客户机通道定义表的信息。 您仍可以在客户机平台上使用 IBM MQ 环境变量。

过程

- Windows
  在 Windows 上，对于每个环境变量，请使用以下命令来显示当前设置或重置变量的值:
  - 要除去环境变量的值，请使用以下命令:

    ```
    SET MQSERVER=
    ```

  - 要显示环境变量的当前设置，请使用以下命令:

    ```
    SET MQSERVER
    ```

  - 要显示会话的所有环境变量，请使用以下命令:

    ```
    set
    ```

- Linux AIX
  在 AIX and Linux 上，对于每个环境变量，请使用以下命令来显示当前设置或重置变量的值:
  - 要除去环境变量的值，请使用以下命令:

    ```
    unset MQSERVER
    ```

  - 要显示环境变量的当前设置，请使用以下命令:

    ```
    echo $MQSERVER
    ```

  - 要显示会话的所有环境变量，请使用以下命令:

    ```
    set
    ```

相关任务

为 IBM MQ classes for JMS 设置环境变量

与 IBM MQ classes for Java 相关的环境变量

在 service.env 文件中定义其他环境变量

第 71 页的『在 Multiplatforms 版上的 .ini 文件中更改 IBM MQ 配置信息』

您可以通过编辑配置 (`.ini`) 文件中的信息来更改 IBM MQ 或单个队列管理器的行为，以满足安装需要。 您还可以更改 IBM MQ MQI clients 的配置选项。

相关参考

在 MFT 属性中使用环境变量

环境变量描述

旨在供客户使用的服务器和客户机环境变量的描述。

使用范例

- Linux AIX 在 AIX and Linux 系统上，使用以下格式: `export [environment variable]=value`。
- Windows 在 Windows 系统上，使用以下格式: `Set [environment variable]=value`。
- IBM i 在 IBM i 系统上，使用以下格式: `ADDENVVAR ENVVAR(environment variable) VALUE(xx)`。
- MQ Appliance 对于 IBM MQ Appliance，请参阅 IBM MQ Appliance 文档中的 在 IBM MQ Appliance 。

AMQ_ALLOWED_CIPHERS

Multi V 9.2.0

从 IBM MQ 9.2.0 开始，您可以使用 **AMQ_ALLOWED_CIPHERS** 环境变量来指定允许与多平台上的 IBM MQ 通道配合使用的 CipherSpecs 的定制列表。 环境变量采用与 `.ini` 文件的 **AllowedCipherSpecs** SSL 节属性相同的值:

- 单个 CipherSpec 名称，或
- 要重新启用的 IBM MQ CipherSpec 名称的逗号分隔列表，或者
- `ALL` 的特殊值，表示所有 CipherSpecs (不推荐)。

注: 建议不要启用 **ALL** CipherSpecs ，因为这将启用 SSL 3.0 和 TLS 1.0 协议以及大量弱加密算法。

有关更多信息，请参阅 TLS 握手中的 CipherSpec 顺序中的 在 Multiplatforms 版上提供已启用 CipherSpecs 的定制列表 。

AMQ_BAD_COMMS_DATA_FDCS

AMQ_BAD_COMMS_DATA_FDCS 环境变量在设置为任何值时生效。

如果 IBM MQ 通过 TCP/IP 从主机接收的数据格式不正确，例如，由于网络客户机已连接到 IBM MQ 侦听器端口并尝试与不受支持的应用程序协议进行通信，那么队列管理器会将 AMQ9207E 错误消息写入队列管理器错误日志。 IBM MQ 侦听器支持来自队列管理器消息通道代理程序 (MCA) 以及来自 MQI， JMS 和 XMS 客户机应用程序的 TCP/IP 连接。

注: IBM MQ 侦听器不支持 AMQP 和 MQTT 客户机使用的应用程序协议，这些客户机应改为连接到在适用的 AMQP 通道或 MQXR 遥测服务中配置的网络端口。

可能还会写入包含 IBM MQ 已接收的无效数据的故障数据捕获 (FDC) 记录。 然而，如果这是与远程端对话的开始，且格式是简单的已知格式，例如来自 HTTP 网页浏览器的 GET 请求，则不会生成 FFST 文件。 如果要覆盖此值以导致针对任何错误数据 (包括简单已知格式) 写入 FFST 文件，那么可以将 **AMQ_BAD_COMMS_DATA_FDCS** 环境变量设置为任何值 (例如， TRUE) 并重新启动队列管理器。

AMQ_CONVAMQICNEWLINE

Multi

从 IBM MQ 9.1.0 Fix Pack 2 和 IBM MQ 9.1.2 开始，您可以使用 **AMQ_CONVEBCDICNEWLINE** 环境变量来指定 IBM MQ 如何将 EBCDIC NL 字符转换为 ASCII 格式。 环境变量采用与 `mqs.ini` 的 **ConvEBCDICNewline** 属性相同的值，即 `NL_TO_LF`， `TABLE` 或 `ISO` (请参阅 mqs.ini 文件的所有队列管理器节)。 例如，在无法使用 `mqs.ini` 文件的情况下，可以使用 **AMQ_CONVEBCDICNEWLINE** 环境变量而不是 **ConvEBCDICNewline** 节属性在客户端上提供 **ConvEBCDICNewline** 功能。 如果同时设置了节属性和环境变量，那么节属性优先。

有关更多信息，请参阅 编码字符集之间的数据转换 。

AMQ_DIAGNOSTIC_MSG_SEVERITY

从 IBM MQ 9.1 开始，如果针对 IBM MQ 进程将 **AMQ_DIAGNOSTIC_MSG_SEVERITY** 环境变量设置为 1，那么当 IBM MQ 进程将消息写入错误日志或控制台时，会将消息严重性作为单个大写字母字符附加到消息号。

缺省情况下，已设置 **AMQ_DIAGNOSTIC_MSG_SEVERITY** 启用的行为。 您可以通过将该环境变量设置为 0 来关闭此行为。

有关更多信息，请参阅 使用错误日志。

AMQ_DISABLE_CLIENT_AMS

如果在尝试从较低版本的产品连接到队列管理器并且正在使用下列其中一个客户机时报告了 2085 (MQRC_UNKNOWN_OBJECT_NAME) 错误，那么可以使用 **AMQ_DISABLE_CLIENT_AMS** 环境变量在客户机上禁用 IBM MQ Advanced Message Security (AMS):

- 除 IBM Java runtime environment (JRE) 以外的 Java runtime environment (JRE)
- IBM MQ IBM MQ classes for JMS 或 IBM MQ classes for Java 客户机。

注: 不能将 **AMQ_DISABLE_CLIENT_AMS** 环境变量用于 C 客户机。 您需要改为使用 **MQS_DISABLE_ALL_INTERCEPT** 环境变量。

有关更多信息，请参阅 在客户机上禁用高级消息安全性。

AMQ_DMPMQCFG_QSGDISP_DEFAULT

从 IBM MQ 9.1.0 Fix Pack 5 和 IBM MQ 9.1.5 开始，缺省情况下， **dmpmqcfg** 命令使用的队列管理器处置查询仅查询 QSGDISP (QMGR) 定义。 您可以使用 **AMQ_DMPMQCFG_QSGDISP_DEFAULT** 环境变量来查询其他定义，该变量可以设置为下列其中一个值:

实时
仅包括使用 QSGDISP (QMGR) 或 QSGDISP (COPY) 定义的对象。

全部
包含使用 QSGDISP (QMGR) 和 QSGDISP (COPY) 定义的对象。 如果队列管理器是队列共享组的成员，那么还包括 QSGDISP (GROUP) 和 QSGDISP (SHARED)。

COPY
仅包括使用 QSGDISP (COPY) 定义的对象

GROUP
仅包括使用 QSGDISP (GROUP) 定义的对象; 目标队列管理器必须是队列共享组的成员。

QMGR
仅包括使用 QSGDISP (QMGR) 定义的对象。 如果使用此环境变量来匹配 **dmpmqcfg** 的现有行为，那么这是缺省行为。

PRIVATE
仅包括使用 QSGDISP (QMGR) 或 QSGDISP (COPY) 定义的对象。

SHARED
仅包括使用 QSGDISP (SHARED) 定义的对象。

AMQ_许可方指标

从 IBM MQ 9.2.0 开始，设置 **AMQ_LICENSING_METRIC=VPCMonthlyPeak** 环境变量将导致队列管理器上载与每月 VPC 许可证类型相关的数据，而不是上载与基于每小时容器的许可证相关的数据的缺省行为。

有关配置 IBM MQ 以用于 IBM Cloud Private 测量服务的更多信息，请参阅 IBM Cloud Private 文档中的 IBM Cloud Private 测量服务 。

AMQ_LDAP_TRACE

从 IBM MQ 9.1.0 Fix Pack 4 和 IBM MQ 9.1.4 开始，如果 **AMQ_LDAP_TRACE** 环境变量设置为非空值，那么可以在不停止或启动队列管理器的情况下打开和关闭 LDAP 客户机跟踪。

有关更多信息，请参阅 启用 LDAP 客户机库代码的动态跟踪。

AMQ_MQS_INI_LOCATION

在 AIX and Linux 系统上，可以通过在 **AMQ_MQS_INI_LOCATION** 环境变量中设置 `mqs.ini` 文件的位置来变更用于 `mqs.ini` 文件的位置。 必须在系统级别设置此环境变量。

有关 `mqs.ini` 文件 (包括目录位置) 的更多信息，请参阅 IBM MQ 配置文件 mqs.ini。

AMQ_NO_BAD_COMMS_DATA_FDCS

AMQ_NO_BAD_COMMS_DATA_FDCS 环境变量在设置为任何值时生效。

如果 IBM MQ 在尝试将非 IBM MQ 客户机连接到 IBM MQ TCP/IP 侦听器时无法识别初始数据传输，那么会导致队列管理器将 AMQ9207E 错误消息写入队列管理器错误日志。 还会写入故障数据捕获 (FDC) 记录。 您可以使用 **AMQ_NO_BAD_COMMS_DATA_FDCS** 环境变量来禁止生成这些诊断文件。 当 **AMQ_NO_BAD_COMMS_DATA_FDCS** 设置为任何值 (例如， TRUE) 时，这将指示 IBM MQ 在初始通信流上报告 AMQ9207E 错误消息时不生成 FFST。 要生效，应在启动队列管理器和侦听器进程之前设置环境变量。

如果客户机将有效的 IBM MQ 协议流发送到队列管理器，然后发送无效数据，那么将继续生成 FDC ，因为这指示存在需要进一步调查的客户机问题。

注: 从 IBM MQ 9.2.0 开始，缺省情况下禁止在初始通信流上报告 AMQ9207E 错误消息时捕获 FFST。

AMQ_NO_IPV6

AMQ_NO_IPV6 环境变量在设置为任何值时生效。 设置此环境变量时，将在尝试连接时禁用 IPv6 。

AMQ_REVERSE_COMMIT_ORDER

AMQ_REVERSE_COMMIT_ORDER 环境变量配置队列管理器，以便在完成相应的数据库更新后，在 XA 事务中落实 IBM MQ 队列管理器更改。 仅在完成相应的数据库更新后，从队列中读取消息的应用程序才会看到消息。

注: 如果不阅读和了解 隔离级别中描述的方案，请不要设置 **AMQ_REVERSE_COMMIT_ORDER** 。

AMQ_SSL_ALLOW_DEFAULT_CERT

从 IBM MQ 9.0.0 Fix Pack 1 和 IBM MQ 9.0.2 开始，如果未设置 **AMQ_SSL_ALLOW_DEFAULT_CERT** 环境变量，那么仅当证书包含 `ibmwebspheremq`*`userid`* 的标签名称时，应用程序才能使用客户机密钥库中的个人证书连接到队列管理器。 设置 **AMQ_SSL_ALLOW_DEFAULT_CERT** 环境变量时，证书不需要标签名称 `ibmwebspheremq`*`userid`*。 即，用于连接到队列管理器的证书可以是缺省证书，前提是密钥存储库中存在缺省证书，并且密钥存储库不包含前缀为 `ibmwebspheremq`*`userid`* 的个人证书。

值 1 允许使用缺省证书。

应用程序可以使用 `mqclient.ini` 文件中 SSL 节的 **CertificateLabel** 设置，而不是使用 **AMQ_SSL_ALLOW_DEFAULT_CERT** 环境变量。 有关更多信息，请参阅 数字证书标签，了解需求 和 客户机配置文件的 SSL 节。

AMQ_SSL_LDAP_SERVER_VERSION

在 CRL 服务器需要使用特定版本的 LDAP 协议的情况下， **AMQ_SSL_LDAP_SERVER_VERSION** 环境变量可用于确保 IBM MQ 加密组件使用 LDAP v2 或 LDAP v3 。

将环境变量设置为用于启动队列管理器或通道的环境中的相应值:

- 要请求使用 LDAP v2，请设置 AMQ_SSL_LDAP_SERVER_VERSION=2。
- 要请求使用 LDAP v3，请设置 AMQ_SSL_LDAP_SERVER_VERSION=3。

此环境变量不会影响 IBM MQ 队列管理器为用户认证或用户授权建立的 LDAP 连接。

AMQ_USE_ZLIBNX

V 9.2.1 AIX

在 AIX 上，**AMQ_USE_ZLIBNX** 环境变量可用于使消息通道代理程序 (MCA) 能够在使用 ZLIBFAST 或 ZLIBHIGH 技术时使用硬件加速的 zlibNX 库来压缩和解压消息数据。

提示: 大小超过 2 KB 的高度可压缩消息最有可能通过减少 CPU 使用率而从选择使用 zlibNX 库中获益。

zlibNX 库在具有技术级别 4 扩展包的 IBM AIX 7.2 和更高版本中提供。 如果设置了环境变量，并且未安装 zlibNX 库 (`/usr/opt/zlibNX/lib/libz.a`)，那么消息通道代理程序将使用 IBM MQ for AIX 安装中提供的标准 zlib 库。

主

IBM i Linux AIX

在 AIX， Linux 和 IBM i 上， **HOME** 环境变量指定要搜索 `mqclient.ini` 文件的目录的名称。 此文件包含 IBM MQ MQI clients 所使用的配置信息。

有关更多信息，请参阅 IBM MQ MQI 客户机配置文件 mqclient.ini 和 客户机配置文件的位置。

HOMEDRIVE 和 HOMEPATH

Windows

要使用，必须同时设置 **HOMEDRIVE** 和 **HOMEPATH** 环境变量。 它们在 Windows 系统上用于指定要搜索 `mqclient.ini` 文件的目录的名称。 此文件包含 IBM MQ MQI clients 所使用的配置信息。

有关更多信息，请参阅 IBM MQ MQI 客户机配置文件 mqclient.ini 和 客户机配置文件的位置。

LDAP 基本 DN

LDAP_BASEDN 是运行 LDAP 样本程序所需的环境变量。 它指定目录搜索的基本专有名称。

LDAP_HOST

LDAP_HOST 是用于运行 LDAP 样本程序的可选环境变量。 它指定运行 LDAP 服务器的主机的名称; 如果未指定，那么缺省为本地主机。

LDAP 版本

LDAP_VERSION 是用于运行 LDAP 样本程序的可选环境变量。 它指定要使用的 LDAP 协议的版本，可以是 2 或 3。 现在，大多数 LDAP 服务器都支持版本 3 的协议; 它们都支持旧版本 2。 此样本与协议的任一版本同样有效，如果未指定此样本，那么缺省为 V 2。

MQ_CHANNEL_SUPPRESS_INTERVAL

MQ_CHANNEL_SUPPRESS_INTERVAL 环境变量指定要禁止使用 **MQ_CHANNEL_SUPPRESS_MSGS** 定义的消息写入错误日志的时间间隔 (以秒为单位)，以及在禁止之前允许消息在指定的时间间隔内出现的次数。 缺省值为 60,5，这意味着在该消息的前五次出现之后，将在 60 秒时间间隔内禁止给定消息的任何其他出现。
有关更多信息，请参阅 在 Multiplatforms 版上禁止来自错误日志的通道错误消息。

MQ_CHANNEL_SUPPRESS_INTERVAL 环境变量与 `qm.ini` 文件中的 SuppressInterval 相当。

MQ_CHANNEL_SUPPRESS_MSGS

MQ_CHANNEL_SUPPRESS_MSGS 环境变量禁止错误日志中的通道错误消息。 您可以指定禁止的消息列表。 **MQ_CHANNEL_SUPPRESS_MSGS** 与 **MQ_CHANNEL_SUPPRESS_INTERVAL** 结合使用，后者指定每条消息在被禁止之前出现的次数以及禁止消息的时间长度。 有关更多信息，请参阅 在 Multiplatforms 版上禁止来自错误日志的通道错误消息。

MQ_CHANNEL_SUPPRESS_MSGS 环境变量与 qm.ini 文件中的 SuppressMessage 相当，但您可以使用该环境变量来禁止任何通道消息，而对于 qm.ini 方法有一个限制列表。

MQ_CONNECT_TYPE

Multi

在 Multiplatforms 版上，可以将 **MQ_CONNECT_TYPE** 环境变量与 MQCONNX 调用上使用的 MQCNO 结构的 "选项" 字段中指定的绑定类型结合使用。 **MQ_CONNECT_TYPE** 仅对 STANDARD 绑定有任何影响。 对于其他绑定，将忽略 **MQ_CONNECT_TYPE** 。

有关更多信息，请参阅 将 MQCONNX 调用选项与 MQ_CONNECT_TYPE 配合使用。

MQ_CROSS_QUEUE_ORDER_ALL

将 **MQ_CROSS_QUEUE_ORDER_ALL** 环境变量设置为非零值时，将在工作单元中维护消息放置顺序。 这意味着，如果将工作单元 (UoW) 中的消息放入多个队列 (例如， Q1，然后是 Q2) ，那么在发出 MQCMIT 时，将以它们 PUT 的相同队列顺序传递消息并使这些消息可用。

在多队列管理器环境中，在启动每个队列管理器之前， **MQ_CROSS_QUEUE_ORDER_ALL** 必须存在并且在发送端和接收端都具有非空值。

MQ_EPHEMERAL_PREFIX

ALW V 9.2.0

MQ_EPHEMERAL_PREFIX 环境变量指定队列管理器的临时目录的路径，当队列管理器正在运行时，将在此目录中保留临时队列管理器数据。

除了通过更改 "mqs.ini 文件中 AllQueueManagersstanza 的 "**DefaultEphemeralPrefix** 属性中的 "**EphemeralPrefix** 属性来更改短暂前缀外，还可以使用环境变量 "**MQ_EPHEMERAL_PREFIX** 来覆盖 "**crtmqm** 命令的 "**EphemeralPrefix**。 有关更多信息，请参阅 可配置临时目录。

MQ_FILE_PATH

Windows

MQ_FILE_PATH 环境变量是在 Windows 平台上安装运行时包期间配置的。 此环境变量包含与 Windows 注册表中的以下键相同的数据:

```
HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName\FilePath
```

有关更多信息，请参阅 setmqenv (set IBM MQ environment) 和 crtmqenv (create IBM MQ environment)。

MQ_JAVA_DATA_PATH

MQ_JAVA_DATA_PATH 环境变量指定 IBM MQ classes for JMS 和 IBM MQ classes for Java 的日志和跟踪输出目录。 它由 IBM MQ classes for JMS 和 IBM MQ classes for Java 随附的脚本使用。

有关更多信息，请参阅 设置 IBM MQ JMS 类的环境变量 和 与 IBM MQ Java 类相关的环境变量。

MQ_JAVA_INSTALL_PATH

MQ_JAVA_INSTALL_PATH 环境变量指定安装 IBM MQ classes for JMS 的目录，如 针对 IBM MQ classes for JMS 安装的内容中所示，以及 IBM MQ classes for Java (如 IBM MQ classes for Java 安装目录 中所示)。

有关更多信息，请参阅 设置 IBM MQ JMS 类的环境变量 和 与 IBM MQ Java 类相关的环境变量。

MQ_JAVA_LIB_PATH

MQ_JAVA_LIB_PATH 环境变量指定存储 IBM MQ classes for JMS 和 IBM MQ classes for Java 库的目录。 IBM MQ classes for JMS 或 IBM MQ classes for Java 随附的某些脚本 (例如 IVTRun) 使用此环境变量。

有关更多信息，请参阅 设置 IBM MQ JMS 类的环境变量 和 与 IBM MQ Java 类相关的环境变量。

MQ_SET_NODELAYACK

AIX

MQ_SET_NODELAYACK 环境变量在 AIX 上关闭 TCP 延迟应答。

设置此环境变量时，该设置通过使用 TCP_NODELAYACK 选项调用操作系统的 setsockopt 调用来关闭 TCP 延迟应答。 只有 AIX 支持此函数，因此 **MQ_SET_NODELAYACK** 环境变量仅对 AIX 有影响。

MQ_USER_NAME

Linux V 9.2.0

如果用户没有密码文件条目，或者用户名称与其用户标识符匹配，则可以使用 **MQ_USER_NAME** 环境变量来允许 IBM MQ 非安装映像在 Linux 上选择用户名。 例如，在 OpenShift 中使用发布/订阅层次结构时，就需要这样做。

MQ_USER_NAME 的值不能与系统上已有的用户匹配，且必须小于或等于 12 个字节。

MQAPI_TRACE_LOGFILE

样本 API 出口程序生成对具有 **MQAPI_TRACE_LOGFILE** 环境变量中定义的前缀的用户指定文件的 MQI 跟踪。

有关更多信息，请参阅 API 出口样本程序。

MQApplName

ALW V 9.2.0

如果尚未选择应用程序名称，那么可以使用 **MQAPPLNAME** 环境变量作为用于标识与队列管理器的连接的名称。 仅使用前 28 个字符，它们不能全部为空白或空。

有关更多信息，请参阅 使用受支持编程语言中的应用程序名称。

MQCCSID

MQCCSID 环境变量指定要使用的编码字符集编号，并覆盖配置了服务器的 CCSID 值。 **MQCCSID** 可用于覆盖应用程序的本机 CCSID 并指定要使用的编码字符集号，例如，如果本机 CCSID 是不受支持的 CCSID 或不是必需的 CCSID。

要设置 **MQCCSID**，请使用下列其中一个命令:

- Linux AIX 在 AIX and Linux 上:

```
export MQCCSID=number
```

- Windows 在 Windows 上:

```
SET MQCCSID=number
```

- IBM i 在 IBM i 上:

```
ADDENVVAR ENVVAR(MQCCSID) VALUE(number)
```

有关更多信息，请参阅 选择客户机或服务器 CCSID。

MQCCDTURL

MQCCDTURL 环境变量提供了设置 **MQCHLLIB** 和 **MQCHLTAB** 环境变量组合的等效功能。 它允许您提供一个文件、ftp 或 http URL 作为单个值，从中可以获得本地程序（即 C、COBOL 或 C++应用程序）作为客户端连接的客户端通道定义表。

注: 使用环境变量来提供 URL 对 Java 、 JMS 或托管的 .NET 应用程序没有影响。

IBM MQ 支持从文件、ftp 或 http 中检索 CCDT。 URL 但是，"**MQCCDTURL** 不接受 FTP 值。 它只接受 URL 值或使用“file://”协议指定的文件。

For example, when using **MQCCDTURL** in place of **MQCHLLIB** and **MQCHLTAB** with a local file on AIX and Linux, this command:

```
export MQCCDTURL=file:///var/mqm/qmgrs/QMGR/@ipcc/MYCHL.TAB
```

相当于:

```
export MQCHLLIB=/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLTAB=MYCHL.TAB
```

您还可以为 Windows 指定 JSON 文件，如以下示例中所示:

```
set MQCCDTURL=file:/c:/mq-channels/CCDT-QMGR1.json
```

相当于:

```
set MQCHLLIB=C:\mq-channels
set MQCHLTAB=CCDT-QMGR1.json
```

如需了解更多信息，请访问 URL ， 访问 CCDT。

MQCERTLABL

MQCERTLABL 环境变量定义通道定义的证书标签，以供 IBM MQ 用于查找在 TLS 握手期间发送的个人证书。

有关更多信息，请参阅 数字证书标签， 了解需求。

MQCERTVPOL

MQCERTVPOL 环境变量指定要使用的证书验证策略的类型。 此环境变量将覆盖客户机配置文件的 SSL 节中的 **CertificateValPolicy** 属性。

可以将 **MQCERTVPOL** 设置为以下两个值之一:

ANY
使用底层安全套接字库支持的任何证书验证策略。 此设置为缺省设置。

RFC5280
仅使用符合 RFC 5280 标准的证书验证。

要设置 **MQCERTVPOL**，请使用下列其中一个命令:

- Linux AIX 对于 AIX and Linux 系统:

  ```
  export MQCERTVPOL= value
  ```

- Windows 对于 Windows 系统:

  ```
  SET MQCERTVPOL= value
  ```

- IBM i 对于 IBM i 系统:

```
ADDENVVAR ENVVAR(MQCERTVPOL) VALUE(value)
```

有关更多信息，请参阅 IBM MQ 中的证书验证策略 和 在 IBM MQ 中配置证书验证策略。

MQCHLLIB

MQCHLLIB 环境变量指定包含客户机通道定义表 (CCDT) 的文件的目录路径。 此文件在服务器上创建，但可复制到 IBM MQ MQI client 工作站。

要设置 **MQCHLLIB**，请使用下列其中一个命令:

- Windows 在 Windows 上:

```
SET MQCHLLIB=pathname
```

例如:

```
SET MQCHLLIB=C:\wmqtest
```

- Linux AIX 对于 AIX and Linux 系统:

```
export MQCHLLIB=pathname
```

- IBM i 对于 IBM i:

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE(pathname)
```

如果未设置 **MQCHLLIB**，那么客户机的路径缺省为:

- Linux AIX 在 AIX and Linux 上: `/var/mqm/`
- Windows 在 Windows 上: *MQ_INSTALLATION_PATH*
- IBM i 在 IBM i 上: `/QIBM/UserData/mqm/`

对于 **crtmqm** 和 **strmqm** 命令，路径缺省为两组路径之一。 如果设置了 *datapath*，那么路径缺省为第一个集合之一。 如果未设置 *datapath*，那么路径缺省为第二个集合之一。

- Linux AIX 在 AIX and Linux 上: *datapath*`/@ipcc`
- Windows 在 Windows 上: *datapath*`\@ipcc`
- IBM i 在 IBM i 上: *datapath*`/&ipcc`

或者:

- Linux AIX 在 AIX and Linux 上: `/prefix/qmgrs/`*qmgrname*`/@ipcc`
- Windows 在 Windows 上: *MQ_INSTALLATION_PATH*`\data\qmgrs\`*qmgrname*`\@ipcc`
- IBM i 在 IBM i 上: `/`*prefix*`/qmgrs/`*qmgrname*`/&ipcc`

其中:

- *MQ_INSTALLATION_PATH* 表示 IBM MQ 安装所在的高级目录。
- 如果存在，那么 *datapath* 是队列管理器节中定义的 DataPath 的值。
- *prefix* 是队列管理器节中定义的前缀的值。 前缀通常是下列其中一个值:
  - Linux AIX AIX and Linux 系统上的 `/var/mqm`。

  - IBM i IBM i 上的 /QIBM/UserData/mqm/ 。
- *qmgrname* 是队列管理器节中定义的 Directory 属性的值。 该值可能与实际队列管理器名称不同。 该值可能已更改以替换特殊字符。
- 定义队列管理器节的位置取决于平台:
  - IBM i Linux AIX 在 IBM i, AIX and Linux 上的 mqs.ini 文件中。
  - Windows 在 Windows 上的注册表中。

注意:

1. z/OS 如果使用 IBM MQ for z/OS 作为服务器，那么必须将该文件保留在 IBM MQ 客户机工作站上。
2. 如果设置了此属性，那么 MQCHLLIB 将覆盖用于查找 CCDT 的路径。
3. MQCHLLIB 可以包含一个与 MQCHLTAB 环境变量结合使用的 （参见 ）。 URL 第 45 页的『对 CCDT 的 URL 访问权』
4. 环境变量 (例如 **MQCHLLIB**) 可以特定于平台的方式作用域限定为进程，作业或系统范围。
5. 如果在服务器上设置 **MQCHLLIB** 系统范围，那么它会为服务器上的所有队列管理器设置到 CCDT 文件的相同路径。 如果未设置 **MQCHLLIB** 环境变量，那么每个队列管理器的路径都不同。 队列管理器在 **crtmqm** 或 **strmqm** 命令上读取 **MQCHLLIB** 的值 (如果已设置)。
6. 如果在一个服务器上创建多个队列管理器，那么区分很重要，原因如下。 如果在系统范围内设置 **MQCHLLIB** ，那么每个队列管理器都会更新相同的 CCDT 文件。 该文件包含来自服务器上所有队列管理器的客户机连接定义。 如果多个队列管理器 (例如， SYSTEM.DEF.CLNTCONN) 上存在相同的定义，那么该文件将包含最新定义。 创建队列管理器时，如果设置了 **MQCHLLIB** ，那么将在 CCDT 中更新 SYSTEM.DEF.CLNTCONN 。 更新将覆盖由其他队列管理器创建的 SYSTEM.DEF.CLNTCONN 。 如果您修改了先前的定义，那么您的修改将丢失。 因此，您必须考虑查找将 **MQCHLLIB** 设置为服务器上的系统范围环境变量的替代方法。
7. 客户机连接定义上的 MQSC 和 PCF NOREPLACE 选项不会检查 CCDT 文件的内容。 将替换先前创建但未由此队列管理器创建的同名客户机连接通道定义，而不考虑 NOREPLACE 选项。 如果定义先前由同一队列管理器创建，那么不会替换该定义。
8. **rcrmqobj** -t clchltab 命令将删除并重新创建 CCDT 文件。 将仅使用在运行该命令的队列管理器上创建的客户机连接定义来重新创建该文件。
9. 其他用于更新 CCDT 的命令仅修改具有相同通道名称的客户机连接通道。 不会更改文件中的其他客户机连接通道。
10. **MQCHLLIB** 的路径不需要引号。

更多信息，请参阅 CCDT 的地点 、 URL 访问 CCDT 以及使用环境变量将客户端应用程序连接到队列管理器。

MQCHLTAB

MQCHLTAB 环境变量指定包含客户机通道定义表 (CCDT) 的文件的名称。 缺省文件名为 AMQCLCHL.TAB。

要设置 **MQCHLTAB**，请使用下列其中一个命令:

- Linux AIX 在 AIX and Linux 上:

```
export MQCHLTAB=filename
```

- Windows 在 Windows 上:

```
SET MQCHLTAB=filename
```

- IBM i 在 IBM i 上:

```
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(filename)
```

例如：

```
SET MQCHLTAB=ccdf1.tab
```

与客户机相同，服务器上的 **MQCHLTAB** 环境变量指定客户机通道定义表的名称。

更多信息，请参阅 CCDT 的地点 、 URL 访问 CCDT 以及使用环境变量将客户端应用程序连接到队列管理器。

MQCLNTCF

MQCLNTCF 环境变量指定 IBM MQ MQI client 配置文件的位置。 此文件包含 IBM MQ MQI clients 所使用的配置信息。

您可以使用 **MQCLNTCF** 环境变量来修改 `mqclient.ini` 文件的文件路径。

这个环境变量的格式为完整的 URL。 这意味着文件名可能不一定是 `mqclient.ini`，这有助于将文件放在网络连接的文件系统上。 更多信息，请参阅 IBM MQ MQI 客户端配置文件、 mqclient.ini 和客户端配置文件的位置。

MQDOTNET_TRACE_ON

MQDOTNET_TRACE_ON 环境变量用于对 IBM MQ .NET 可再分发客户机启用跟踪。 值等于或小于 0 将不启用跟踪， 1 将启用缺省跟踪，而大于 1 的值将启用详细信息跟踪。

有关更多信息，请参阅安装 IBM MQ classes for .NET Standard。

MQIPADDRV

MQIPADDRV 环境变量指定要用于通道连接的 IP 协议。 它具有可能的字符串值 "MQIPADDR_IPV4" 或 "MQIPADDR_IPV6"。 这些值与 **ALTER QMGR IPADDRV** 中的 IPv4 和 IPv6 以及客户机配置文件的 TCP 节的 **IPAddressVersion** 属性具有相同的含义。 如果未设置环境变量，那么将采用 "MQIPADDR_IPV4" 。

要设置 **MQIPADDRV**，请使用下列其中一个命令：

- Linux AIX 在 AIX and Linux 上：

  ```
  export MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6"/>
  ```

- Windows 在 Windows 上：

  ```
  SET MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6
  ```

- IBM i 在 IBM i 上：

  ```
  ADDENVVAR ENVVAR(MQIPADDRV) VALUE(MQIPADDR_IPV4|MQIPADDR_IPV6)
  ```

MQ 许可证

Linux V 9.2.0

在 Linux 系统上，可以在安装产品后使用 **MQLICENSE** 环境变量来接受或查看 IBM MQ 许可证。

有关您可能希望或需要执行此操作的原因的更多信息，请参阅 IBM MQ 上针对 Linux 的许可证接受

可以将 **MQLICENSE** 环境变量设置为以下两个值之一：

接受
接受安装后的许可证。

视图
显示许可证 (如果已接受许可证)。

要接受安装后的许可证，请使用以下命令：

```
export MQLICENSE=accept
```

要查看许可证，请使用以下命令:

```
export MQLICENSE=view
```

注: 您还可以使用以下命令来接受和显示许可证:

- mqlicense (安装后接受许可证)
- dspmqlic (显示 IBM MQ 许可证)

MQMAXERRORLOGSIZE

Multi

MQMAXERRORLOGSIZE 环境变量指定复制到备份的队列管理器错误日志的大小。

有关更多信息，请参阅 使用错误日志。

MQNAME

Windows

MQNAME 环境变量指定 IBM MQ 进程可以使用的本地 NetBIOS 名称。 NetBIOS 连接仅适用于运行 Windows 的客户机和服务器。

要设置 **MQNAME**，请使用以下命令:

```
SET MQNAME=Your_env_Name
```

例如:

```
SET MQNAME=CLIENT1
```

如果要在 IBM MQ MQI client 上同时运行多个 IBM MQ 应用程序，那么某些 NetBIOS 实现需要每个应用程序的唯一名称 (由 **MQNAME** 设置)。

有关更多信息，请参阅 定义 IBM MQ 本地 NetBIOS 名称。

MQNOREMPOOL

设置 **MQNOREMPOOL** 环境变量时，它会关闭通道池并使通道作为侦听器的线程运行。

有关更多信息，请参阅 MCATYPE (消息通道代理程序类型)。

MQPSE_TRACE_LOGFILE

运行 "发布出口样本程序" AMQSPSE0 时使用 **MQPSE_TRACE_LOGFILE** 环境变量，该程序是出口的样本 C 程序，用于在将发布交付到订户之前拦截该发布。 在要跟踪的应用程序进程中，此环境变量描述必须将跟踪文件写入的位置。

有关更多信息，请参阅 "发布出口" 样本程序。

MQS_AMSCRED_KEYFILE

您可以使用 **MQS_AMSCRED_KEYFILE** 环境变量来覆盖或提供要在 IBM MQ Advanced Message Security (AMS) 应用程序运行时使用的初始密钥文件，或者使用 **runamscred** 命令保护密钥库配置文件时使用的初始密钥文件。

有关更多信息，请参阅 将密钥库和证书与 AMS 和 保护 IBM MQ 组件配置文件中的密码配合使用。

MQS_DISABLE_ALL_INTERCEPT

如果在尝试从较低版本的产品连接到队列管理器并且将 IBM MQ 与本机 C 客户机配合使用时报告了 2085 (MQRC_UNKNOWN_OBJECT_NAME) 错误，那么可以使用 **MQS_DISABLE_ALL_INTERCEPT** 环境变量来禁用 IBM MQ Advanced Message Security (AMS)。

注: 只能将 **MQS_DISABLE_ALL_INTERCEPT** 环境变量用于 C 客户机。 对于 Java 客户机，需要改为使用 **AMQ_DISABLE_CLIENT_AMS** 环境变量。

有关更多信息，请参阅 在客户机上禁用高级消息安全性。

MQS_IPC_HOST

由于 IPC 文件系统对象必须由系统区分，因此将队列管理器运行所在的每个系统的一个子目录添加到目录路径中。 如果生成的主机名值产生问题，那么可以使用 **MQS_IPC_HOST** 环境变量来设置主机名。

有关更多信息，请参阅 在多平台上共享 IBM MQ 文件。

MQS_KEYSTORE_CONF

MQS_KEYSTORE_CONF 环境变量指定 IBM MQ Advanced Message Security (AMS) 的密钥库配置文件的位置 (如果该文件不在 *home_directory*/.mqs/keystore.conf 的缺省位置)。

有关更多信息，请参阅 将密钥库和证书用于 AMS。

如果您在 Managed File Transfer 上迁到问题，请参阅 MFT 未从 AMS 中的密钥库配置文件读取密钥库属性时要执行的操作。

MQS_TRACE_OPTIONS

AIX

对于 AIX 上的选择性组件跟踪，请使用 **MQS_TRACE_OPTIONS** 环境变量来分别激活高详细信息和参数跟踪功能。

注: 仅当 IBM 支持人员指示您这样做时，才设置 **MQS_TRACE_OPTIONS** 环境变量。

有关更多信息，请参阅 AIX and Linux 上的跟踪。

MQSERVER

MQSERVER 环境变量用于定义最小通道。 **MQSERVER** 指定 IBM MQ 服务器的位置以及要使用的通信方法。

注: 不能使用 **MQSERVER** 来定义 TLS 通道或具有通道出口的通道。 有关如何定义 TLS 通道的更多信息，请参阅 使用 TLS 保护通道。

以下示例显示如何设置 **MQSERVER**:

- Linux AIX 在 AIX and Linux 上:

```
export  MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
```

- Windows 在 Windows 上:

```
SET  MQSERVER=SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)
```

- IBM i 在 IBM i 上:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)')
```

注:

- 通道名称不能包含正斜杠 (/) 字符，因为此字符用于分隔通道名称，传输类型和连接名称。 当使用 **MQSERVER** 环境变量来定义客户机通道时，将使用最大消息长度 (MAXMSGL) 100 MB。 因此，对通道有效的最大消息大小是在服务器上的 SVRCONN 通道中指定的值。
- 传输类型可以是 LU62 ， TCP ， NETBIOS 和 SPX 之一，具体取决于您的 IBM MQ 客户机平台。
- 连接名称必须是标准网络名。 例如，AMACHINE.ACOMPANY.COM(1414)。
- 连接名称可以是以逗号分隔的连接名称列表。 列表中的连接名称以类似于客户机连接表中的多个连接的方式使用。 连接名称列表可用作队列管理器组的替代方法，以指定客户机要尝试的多个连接。 如果要配置多实例队列管理器，那么可以使用连接名称列表来指定不同的队列管理器实例。

如果使用 **MQSERVER** 环境变量来定义 IBM MQ MQI client 机器与服务器之间的通道，那么这是可供应用程序使用的唯一通道，并且不会引用客户机通道定义表 (CCDT)。

有关更多信息，请参阅 在 IBM MQ MQI 客户机上使用 MQSERVER 创建客户机连接通道。

MQSNOAUT

警告: 建议不要使用此功能。

当您将 **MQSNOAUT** 环境变量设置为任何值时，它将禁用对象权限管理器 (OAM) 并阻止任何安全性检查。 这可能适用于测试环境。 这包括授权和连接认证功能。 TLS ， 通道认证记录和安全出口不受影响。

MQSNOAUT 环境变量仅在创建队列管理器时生效。

警告: 要启用 OAM ， 必须删除队列管理器，删除环境变量，然后在不指定 **MQSNOAUT** 的情况下重新创建队列管理器。

有关更多信息，请参阅 在 AIX， Linux 和 Windows 系统上防止安全访问检查。

MQSPREFIX

作为更改缺省前缀的替代方法，您可以使用 **MQSPREFIX** 环境变量来覆盖 **crtmqm** 命令的 **DefaultPrefix** 。

更多信息，请参阅 "IBM MQ 文件名和 "AllQueueManagers stanza of the mqs.ini file"。

MQSSLCRYP

ALW

MQSSLCRYP 环境变量包含可用于配置系统上存在的加密硬件的参数字符串。

LTS 允许的值与 **ALTER QMGR** 命令的 **SSLCRYP** 参数的值相同。

V 9.2.3 允许的值与客户机配置文件的 SSL 节中的 SSLCryptoHardware 字段相同。

要设置 **MQSSLCRYP**，请使用下列其中一个命令:

- Linux AIX 在 AIX and Linux 系统上:

```
export MQSSLCRYP=string
```

- Windows 在 Windows 系统上:

```
SET MQSSLCRYP=string
```

有关更多信息，请参阅 保护 IBM MQ 组件配置文件中的密码中的 在 AIX, Linux, and Windows 和 IBM MQ clients 上配置使用加密硬件的加密硬件 。

MQSSLFIPS

MQSSLFIPS 环境变量指定在 IBM MQ 中执行密码术时是否仅使用 FIPS 认证的算法。 您可以将此环境变量设置为 YES 或 NO 。缺省值为 NO。 这些值与 **ALTER QMGR** 命令的 **SSLFIPS** 参数相同。

要设置 **MQSSLFIPS**，请使用下列其中一个命令:

- Linux AIX 在 AIX and Linux 系统上:

```
export MQSSLFIPS=YES|NO
```

- Windows 在 Windows 系统上:

```
SET MQSSLFIPS=YES|NO
```

- IBM i 在 IBM i 上:

```
ADDENVVAR ENVVAR(MQSSLFIPS) VALUE(YES|NO)
```

使用加密硬件会影响 FIPS 认证的算法的使用。 有关更多信息，请参阅 指定在 MQI 客户机上运行时仅使用 FIPS 认证的 CipherSpecs。

MQSSLKEYR

MQSSLKEYR 环境变量指定以系统格式保存属于用户的数字证书的密钥存储库的位置。 主干格式表示它包含完整路径和不带扩展名的文件名。

要设置 **MQSSLKEYR**，请使用下列其中一个命令:

- Linux AIX 在 AIX and Linux 系统上:

```
export MQSSLKEYR=pathname
```

- Windows 在 Windows 系统上:

```
SET MQSSLKEYR=pathname
```

- IBM i 在 IBM i 上:

```
ADDENVVAR ENVVAR(MQSSLKEYR) VALUE(pathname)
```

此环境变量没有缺省值。

有关更多信息，请参阅 **ALTER QMGR** 命令的 **SSLKEYR** 参数。

MQSSLPROXY

MQSSLPROXY 环境变量指定了 HTTP 代理服务器的主机名和端口号，供 GSKit 进行 OCSP 检查。

要设置 **MQSSLPROXY**，请使用下列其中一个命令:

- Linux AIX 在 AIX and Linux 系统上:

```
export MQSSLPROXY="string"
```

- Windows 在 Windows 系统上:

```
SET MQSSLPROXY= string
```

您指定的字符串 **MQSSLPROXY** 可以是主机名，也可以是 HTTP 代理服务器的网络地址， GSKit 将使用该地址进行 OCSP 检查。 此地址可以后跟可选端口号 (括在括号内)。 如果不指定端口号，那么将使用缺省 HTTP 端口 80。

Linux AIX 例如，在 AIX and Linux 系统上，可以使用下列其中一个命令:

- ```
  export MQSSLPROXY="proxy.example.com(80)"
  ```
- ```
  export MQSSLPROXY="127.0.0.1"
  ```

有关更多信息，请参阅 使用联机证书状态协议 (OCSP)。

MQSSLRESET

MQSSLRESET 环境变量指定在重新协商 TLS 密钥之前在 TLS 通道上发送和接收的未加密字节数。 它可以设置为 0 到 999 999 范围内的整数。 缺省值为 0 ，指示永远不会重新协商密钥。 如果指定 1 字节到 32 KB 范围内的 TLS 密钥重置计数，那么 TLS 通道将使用 32 KB 的密钥重置计数。 此密钥重置计数是为了避免对小型 TLS 密钥重置值进行过多的密钥重置。

要设置 **MQSSLRESET**，请使用下列其中一个命令:

- Linux AIX 在 AIX and Linux 系统上:

  ```
  export MQSSLRESET=integer
  ```
- Windows 在 Windows 系统上:

  ```
  SET MQSSLRESET=integer
  ```
- IBM i 在 IBM i 上:

  ```
  ADDENVVAR ENVVAR(MQSSLRESET) VALUE(integer)
  ```

有关更多信息，请参阅 重置 SSL 和 TLS 密钥。

MQSUITEB

ALW

您可以在 AIX, Linux, and Windows 平台上配置 IBM MQ 以符合 NSA Suite B 标准。

MQSUITEB 环境变量指定是否使用符合 Suite B 的密码术。 如果要使用 Suite B 密码术，那么可以通过将 **MQSUITEB** 设置为下列其中一项来指定密码术的强度:

- NONE
- 128_BIT, 192_BIT
- 128_BIT
- 192_BIT

您可以使用逗号分隔列表来指定多个值。 将值 NONE 与任何其他值配合使用无效。

有关更多信息，请参阅 配置 IBM MQ for Suite B。

MQTCPTIMEOUT

MQTCPTIMEOUT 环境变量指定 IBM MQ 等待 TCP 连接调用的时间长度。

ODQ_MSG

如果使用与 **runmqdlq** 不同的死信队列处理程序，那么样本的源 amqsdlq 可供您用作基础。 该样本类似于产品中提供的死信处理程序，但跟踪和错误报告不同。 使用 **ODQ_MSG** 环境变量来设置包含错误消息和参考消息的文件的名称。 提供的文件称为 amqsdlq.msg。

有关更多信息，请参阅 死信队列处理程序样本。

ODQ_TRACE

如果使用与 **runmqdlq** 不同的死信队列处理程序，那么样本的源 amqsdlq 可供您用作基础。 该样本类似于产品中提供的死信处理程序，但跟踪和错误报告不同。 要启用跟踪，请将 **ODQ_TRACE** 环境变量设置为 YES 或 yes。

有关更多信息，请参阅 死信队列处理程序样本。

ONCONFIG

ONCONFIG 环境变量指定 Informix 服务器配置文件的名称。 例如，在 AIX and Linux 系统上，使用以下命令:

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

在 Windows 系统上，使用以下命令:

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

有关更多信息，请参阅配置 Informix。

WCF_TRACE_ON

WCF 定制通道可以使用两种不同的跟踪方法。 这两种跟踪方法可以独立激活，也可以一起激活。 每种方法都会生成各自的跟踪文件，所以当两种跟踪方法都激活时，会生成两个输出文件。 可以使用四种组合来启用和禁用 2 种不同的跟踪方法。 除了这些用于启用 WCF 跟踪的组合外，还可以使用 **WCF_TRACE_ON** 环境变量来启用 XMS .NET 跟踪。

有关更多信息，请参阅 跟踪 IBM MQ 的 WCF 定制通道。

WMQSOAP_HOME

在 IBM MQ 中正确安装和配置 .NET SOAP over JMS 服务托管环境之后，完成其他配置步骤时，将使用 **WMQSOAP_HOME** 环境变量。 可从本地队列管理器进行访问。

有关更多信息，请参阅 WCF 客户机到 IBM MQ 样本托管的 .NET 服务 和 WCF 客户机到由 IBM MQ 样本托管的 Axis Java 服务。

XMS_TRACE_ON、XMS_TRACE_FILE_NUMBER、XMS_TRACE_FILE_PATH、XMS_TRACE_FILE_SIZE、XMS_TRACE_FORMAT 和 XMS_TRACE_SPECIFICATION

如果您正在使用 IBM MQ classes for XMS .NET Framework，那么可以从应用程序配置文件以及 XMS 环境变量配置跟踪。

V 9.2.0 如果使用 IBM MQ classes for XMS .NET Standard，那么必须通过 XMS 环境变量来配置跟踪。 通常在 IBM 支持人员的指导下使用跟踪。

要为 XMS .NET 应用程序启用和配置跟踪，请在运行应用程序之前设置以下环境变量:

XMS_TRACE_ON
如果设置了 **XMS_TRACE_ON** 环境变量，那么缺省情况下将启用所有跟踪。

xms_trace_file_number
XMS_TRACE_FILE_NUMBER 环境变量指定要保留的跟踪文件数量。 缺省值为 4（1 个活动文件和 3 个归档文件）。 允许的最小数目是 2。

XMS_TRACE_FILE_PATH
XMS_TRACE_FILE_PATH 环境变量指定跟踪和 FFDC 记录写入的目录的标准路径名 (如果您希望将这些记录从当前工作目录写入备用位置)。

xms_trace_file_size
XMS_TRACE_FILE_SIZE 环境变量指定跟踪文件的最大允许大小。 当文件达到此大小时，会将其归档并重命名。 默认最大值为 20 MB。 文件大小以字节为单位。 例如，XMS_TRACE_FILE_SIZE=“20000000”。

XMS_TRACE_FORMAT
XMS_TRACE_FORMAT 环境变量指定必需的跟踪格式，可以是 BASIC 或 ADVANCED。

XMS_TRACE_SPECIFICATION
XMS_TRACE_SPECIFICATION 环境变量将覆盖在应用程序配置文件的 跟踪部分中定义的跟踪设置。 **XMS_TRACE_SPECIFICATION** 仅适用于 IBM MQ classes for XMS .NET Framework 。

有关更多信息，请参阅 跟踪 XMS .NET 应用程序 和 使用 XMS 环境变量跟踪 XMS .NET 应用程序。

相关任务

第 53 页的『使用 IBM MQ 环境变量』
您可以使用命令来显示当前设置或重置 IBM MQ 环境变量的值。

Multi 在 Multiplatforms 版上的 .ini 文件中更改 IBM MQ 配置信息

您可以通过编辑配置 (.ini) 文件中的信息来更改 IBM MQ 或单个队列管理器的行为，以满足安装需要。 您还可以更改 IBM MQ MQI clients 的配置选项。

关于此任务

您可以通过更改一组用于管理 IBM MQ 的配置属性 (或参数) 上指定的值，在节点或队列管理器级别更改 IBM MQ 配置信息。

配置文件 (或节文件) 包含一个或多个节，这些节是 .ini 文件中具有公共函数或定义系统部分 (例如日志函数，通道函数和可安装服务) 的一组行。 您可以在以下配置文件中修改 IBM MQ 配置属性:

IBM MQ 配置文件， mqs.ini
mqs.ini 文件将对整个节点产生影响。 每个 IBM MQ 安装都有一个 mqs.ini 文件。

由于 IBM MQ 配置文件用于查找与队列管理器关联的数据，因此不存在或不正确的配置文件可能会导致某些或所有 MQSC 命令失败。 此外，应用程序无法连接到未在 IBM MQ 配置文件中定义的队列管理器。

队列管理器配置文件， qm.ini
qm.ini 文件影响特定队列管理器的更改。 节点上的每个队列管理器都有一个 qm.ini 文件。

IBM MQ MQI client 配置文件， mqclient.ini
IBM MQ MQI clients 的配置选项单独保存在客户机配置文件 (通常名为 mqclient.ini) 中。

活动跟踪配置文件 mqat.ini
mqat.ini 文件用于配置活动跟踪行为。

例如，在以下情况下，您可能需要编辑配置文件:

- 您丢失了配置文件。 (如果可以，请从备份中恢复。)
- 您需要将一个或多个队列管理器移至新目录。
- 您需要更改缺省队列管理器。 如果意外删除现有队列管理器，那么可能会发生此情况。
- IBM 支持人员建议您执行此操作。

要点: 您对配置文件所作的任何更改通常不会生效，直到下次启动队列管理器为止。

关于编辑配置文件的要注意的要点:

- 根据以下优先级设置配置文件的属性值:
  - 在命令行上输入的参数优先于配置文件中定义的值。
  - qm.ini 文件中定义的值优先于 mqs.ini 文件中定义的值。
- 安装后，可以编辑 IBM MQ 配置文件中的缺省值。
- 备份队列管理器时，请记住同时包含其配置文件 (qm.ini) 和中央 IBM MQ 配置文件 (mqs.ini)。

- 如果对配置文件属性设置了不正确的值，那么效果与完全缺少该属性相同。 将忽略该值，并发出操作员消息以指示问题。
- IBM i 在 IBM i 上， `.ini` 文件是驻留在 IFS 中的流文件。
- `mqat.ini` 文件的格式有许多语法规则。 有关更多信息，请参阅应用程序活动跟踪 使用 `mqat.ini` 配置活动跟踪行为。

ALW 在 AIX 或 Linux 上，安装配置文件 `mqinst.ini` 包含有关所有 IBM MQ 安装的信息。 不得直接编辑或引用 `mqinst.ini` 文件，因为其格式不固定，并且可能发生更改。 而是必须使用命令对其进行编辑。 有关更多信息，请参阅第 127 页的『安装配置文件， mqinst.ini』。

过程

1. 在编辑配置文件之前，请将其备份，以便在需要时可以将其还原为副本。
2. 通过下列其中一种方式编辑 `.ini` 配置文件:
   - 使用标准文本编辑器手动操作。 通过在注释文本之前添加 “;” 或 “#” 字符，可以将注释包括在配置文件中。 如果要使用 “;” 或 “#” 字符而不使用该字符表示注释，那么可以使用 “\” 字符作为该字符的前缀。 然后，该字符将用作配置数据的一部分。
   - 通过使用用于更改节点上的队列管理器配置的命令自动执行。 有关更多信息，请参阅 命令参考。

     Windows 例如， Windows specific command **amqmdain** 将自动更新 `qm.ini` 属性的子集。 有关更多信息，请参阅 amqmdain。
   - Windows Linux 在 Linux (x86 和 x86-64) 和 Windows 上，您可以使用 IBM MQ Explorer 来更新 `qm.ini` 属性的子集。 有关更多信息，请参阅 使用 MQ Explorer 配置 IBM MQ。

     注: 由于更改可安装服务及其组件会产生重大影响，因此可安装服务在 IBM MQ Explorer 中是只读的。 因此，必须通过编辑 `qm.ini` 文件对可安装服务进行任何更改。 有关更多信息，请参阅第 114 页的『qm.ini 文件的服务节』。

相关任务
管理 IBM MQ

Multi IBM MQ 配置文件， mqs.ini

IBM MQ 配置文件 `mqs.ini` 包含与节点上所有队列管理器相关的信息。 它是在安装期间自动创建的。

注: 有关如何以及何时编辑 `mqs.ini` 文件以及对该文件所作的任何更改何时生效的更多信息，请参阅 第 71 页的『在 Multiplatforms 版上的 .ini 文件中更改 IBM MQ 配置信息』。

目录位置

Linux AIX 在 AIX and Linux 上，数据目录和日志目录始终分别为 `/var/mqm` 和 `/var/mqm/log` 。

Windows 在 Windows 系统上，数据目录 `mqs.ini` 的位置以及日志目录的位置存储在注册表中，因为它们的位置可能有所不同。 安装配置信息 (包含在 AIX and Linux 系统上的 `mqinst.ini` 中) 也位于注册表中，因为 Windows 上没有 `mqinst.ini` 文件 (请参阅 第 127 页的『安装配置文件， mqinst.ini』)。

Windows Windows 系统的 `mqs.ini` 文件由 `HKLM\SOFTWARE\IBM\IBM MQ` 键中指定的 WorkPath 提供。 它包含:

- 队列管理器名称
- 缺省队列管理器的名称
- 与每个文件关联的文件的位置

IBM i 在 IBM i 上， `mqs.ini` 文件存储在 `/QIBM/UserData/mqm` 中。 该文件包含:

- 队列管理器的名称。
- 缺省队列管理器的名称。
- 与每个队列管理器关联的文件的位置。
- 标识任何 API 出口的信息 (请参阅 配置 API 出口 以获取更多信息)。

特别是， `mqs.ini` 文件用于查找与每个队列管理器关联的数据。

AIX and Linux 的示例 `mqs.ini` 文件

Linux AIX

```
#********************************************************************#
#* Module Name: mqs.ini                                             *#
#* Type       : IBM MQ Machine-wide Configuration File              *#
#* Function   : Define IBM MQ resources for an entire machine       *#
#********************************************************************#
#* Notes      :                                                     *#
#* 1) This is the installation time default configuration           *#
#*                                                                  *#
#********************************************************************#
AllQueueManagers:
#********************************************************************#
#* The path to the qmgrs directory, below which queue manager data  *#
#* is stored                                                        *#
#********************************************************************#
DefaultPrefix=/var/mqm

LogDefaults:
   LogPrimaryFiles=3
   LogSecondaryFiles=2
   LogFilePages=4096
   LogType=CIRCULAR
   LogBufferPages=0
   LogDefaultPath=/var/mqm/log

QueueManager:
   Name=saturn.queue.manager
   Prefix=/var/mqm
   Directory=saturn!queue!manager
   InstallationName=Installation1

QueueManager:
   Name=pluto.queue.manager
   Prefix=/var/mqm
   Directory=pluto!queue!manager
   InstallationName=Installation2

DefaultQueueManager:
   Name=saturn.queue.manager

ApiExitTemplate:
   Name=OurPayrollQueueAuditor
   Sequence=2
   Function=EntryPoint
   Module=/usr/ABC/auditor
   Data=123

ApiExitCommon:
   Name=MQPoliceman
   Sequence=1
   Function=EntryPoint
   Module=/usr/MQPolice/tmqp
   Data=CheckEverything
```

Windows 的示例 `mqs.ini` 文件

Windows

```
#********************************************************************#
#* Module Name: mqs.ini                                             *#
#* Type       : IBM MQ Machine-wide Configuration File              *#
#* Function   : Define IBM MQ resources for an entire machine       *#
#********************************************************************#
#* Notes      :                                                     *#
```

```
#* 1) This is the installation time default configuration              *#
#*                                                                     *#
#*********************************************************************#
AllQueueManagers:
#*********************************************************************#
#* The path to the qmgrs directory, below which queue manager data     *#
#* is stored                                                           *#
#*********************************************************************#
DefaultPrefix=C:\ProgramData\IBM\MQ

LogDefaults:
   LogPrimaryFiles=3
   LogSecondaryFiles=2
   LogFilePages=4096
   LogType=CIRCULAR
   LogBufferPages=0
   LogDefaultPath=C:\ProgramData\IBM\MQ\log

QueueManager:
   Name=saturn.queue.manager
   Prefix=C:\ProgramData\IBM\MQ
   Directory=saturn!queue!manager
   InstallationName=Installation1

QueueManager:
   Name=pluto.queue.manager
   Prefix=C:\ProgramData\IBM\MQ
   Directory=pluto!queue!manager
   InstallationName=Installation2

DefaultQueueManager:
   Name=saturn.queue.manager

ApiExitTemplate:
   Name=OurPayrollQueueAuditor
   Sequence=2
   Function=EntryPoint
   Module=C:\usr\ABC\auditor
   Data=123

ApiExitCommon:
   Name=MQPoliceman
   Sequence=1
   Function=EntryPoint
   Module=C:\usr\MQPolice\tmpq
   Data=CheckEverything
```

IBM i 的示例 mqs.ini 文件

IBM i

```
#*********************************************************************#
#* Module Name: mqs.ini                                                *#
#* Type       : IBM MQ Configuration File                              *#
#* Function   : Define IBM MQ resources for the node                   *#
#*                                                                     *#
#*********************************************************************#
#* Notes      :                                                        *#
#* 1) This is an example IBM MQ configuration file                     *#
#*                                                                     *#
#*********************************************************************#
AllQueueManagers:
#*********************************************************************#
#* The path to the qmgrs directory, within which queue manager data    *#
#* is stored                                                           *#
#*********************************************************************#
DefaultPrefix=/QIBM/UserData/mqm

QueueManager:
Name=saturn.queue.manager
Prefix=/QIBM/UserData/mqm
Library=QMSATURN.Q
Directory=saturn!queue!manager

QueueManager:
Name=pluto.queue.manager
Prefix=/QIBM/UserData/mqm
Library=QMPLUTO.QU
Directory=pluto!queue!manager
```

```
DefaultQueueManager:
Name=saturn.queue.manager
```

注意:

1. 节点上的 IBM MQ 使用队列管理器和日志的缺省位置。
2. 队列管理器 saturn.queue.manager 是节点的缺省队列管理器。 与此队列管理器关联的文件的目录已自动转换为文件系统的有效文件名。
3. 由于 IBM MQ 配置文件用于查找与队列管理器关联的数据，因此不存在或不正确的配置文件可能会导致部分或全部 IBM MQ 命令失败。 此外，应用程序无法连接到未在 IBM MQ 配置文件中定义的队列管理器。

mqs.ini 节

注意: 本主题链接到 mqs.ini 文件中有关节的更多信息。 每个节都包含有关该节中的参数的信息。

Multi mqs.ini 文件节和属性的摘要

IBM MQ 配置文件 mqs.ini 的节的属性摘要，其中包含指向更多信息的链接。

表 9: *mqs.ini* 文件的节	
节和属性	**属性的描述**
AllQueueManagers 节	
DefaultPrefix	qmgrs 目录的路径，在此目录中保留队列管理器数据。
ALW V 9.2.0 DefaultEphemeral 前缀	保留队列管理器临时数据的目录的路径。
Multi ConvEBCDICNewline	IBM MQ 如何将 EBCDIC NL 字符转换为 ASCII 格式
ApiExit 公共节和 ApiExit 模板节	
Name	传递到 MQAXP 结构中 ExitInfoName 字段的 API 出口的描述性名称。
函数	指向包含 API 出口代码的模块的函数入口点名称。
模块	包含 API 出口代码的模块。
数据	要传递到 MQAXP 结构的 ExitData 字段中 API 出口的数据。
Sequence	调用此 API 出口的顺序与其他 API 出口有关。
DefaultQueueManager 节	
Name	处理未显式指定队列管理器名称的任何命令的队列管理器的名称。
ExitProperties 节	
CLWLMode	集群工作空间 (CLWL) 出口是否以 FAST 方式或 SAFE 方式运行。
LogDefaults 节	
LogPrimaryFiles	创建队列管理器时分配的日志文件。
LogSecondaryFiles	耗尽主文件时分配的日志文件。

表 9: *mqs.ini* 文件的节 (继续)	
节和属性	**属性的描述**
LogFilePages	日志文件页数。 (日志文件大小以 4 KB 页面为单位指定。)
LogType	队列管理器要使用的日志记录类型 (循环或线性)。
LogBufferPages	分配给缓冲区记录以进行写入的内存量，指定缓冲区大小 (以 4 KB 页面为单位)。
LogDefaultPath	队列管理器的日志文件所在的目录。
LogWriteIntegrity	记录器用于可靠地写入日志记录的方法。
QueueManager 节	
Name	队列管理器的名称。
前缀	存储队列管理器文件的位置。
目录	存储队列管理器文件的 `prefix\QMGRS` 目录下的子目录的名称。
DataPath	创建队列管理器时提供的显式数据路径，将覆盖前缀和目录作为队列管理器数据的路径。
InstallationName	与此队列管理器关联的 IBM MQ 安装的名称。
V 9.2.0 EphemeralPrefix	存储队列管理器临时数据的位置。

Multi mqs.ini 文件的 AllQueueManagers 节

AllQueueManagers 节可以指定存储与队列管理器关联的文件的 `qmgrs` 目录的路径，可执行库的路径以及将 EBCDIC 格式数据转换为 ASCII 格式的方法。

使用 `mqs.ini` 文件中的 AllQueueManagers 节来指定有关所有队列管理器的信息。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer `General` 和 `Extended` IBM MQ 属性页面。

DefaultPrefix= *directory_name*

此属性指定 `qmgrs` 目录的路径，在此目录中保留队列管理器数据。

如果更改队列管理器的缺省前缀，请复制在安装时创建的目录结构。 特别是，必须创建 qmgrs 结构。 在更改缺省前缀之前停止 IBM MQ ，并且仅在将结构移至新位置并更改缺省前缀后重新启动 IBM MQ 。

注: ALW 请勿删除 AIX and Linux 系统上的 `/var/mqm/errors` 目录或 Windows 系统上的 `\errors` 目录。

作为更改缺省前缀的替代方法，您可以使用环境变量 **MQSPREFIX** 来覆盖 **crtmqm** 命令的 **DefaultPrefix** 。

由于操作系统限制，请保持提供的路径足够短，以使路径长度与任何队列管理器名称的总和最大长度为 70 个字符。

Multi **DefaultEphemeral 前缀 = *directory_name***

此属性指定目录的路径，在此目录中保留队列管理器临时数据 (例如 IPC 套接字) ，并且仅用于在创建队列管理器时设置队列管理器的 **EphemeralPrefix** 。 此外，如果更改缺省值，那么必须自行创建目录。 您必须创建具有许可权的临时数据目录，以允许 IBM MQ 组访问权写入该目录。

作为更改 `mqs.ini` 文件的替代方法，您可以使用环境变量 **MQ_EPHEMERAL_PREFIX** 来覆盖 **crtmqm** 命令的 **DefaultEphemeralPrefix** 。

由于操作系统限制，缺省临时前缀限制为:

- Linux AIX AIX and Linux 平台上的 12 个字符。
- IBM i IBM i 上的 24 个字符。

MQ Appliance **DefaultEphemeralPrefix** 在 IBM MQ Appliance 上不受支持。

Multi **ConvEBCDICNewline= NL_TO_LF | TABLE | ISO**
EBCDIC 代码页包含 ASCII 代码页不支持的换行符 (NL) (尽管 ASCII 的某些 ISO 变体包含等效字符)。 使用 **ConvEBCDICNewline** 属性可指定 IBM MQ 如何将 EBCDIC NL 字符转换为 ASCII 格式。

IBM i 在 IBM MQ for IBM i 上， CCSID 1253 被视为 ISO CCSID， NL_TO_LF 同时影响 ISO 和 ASCII 转换。

z/OS **ConvEBCDICNewline** 属性在 z/OS 上不可用。 z/OS 上的行为等同于 `ConvEBCDICNewline=TABLE`。 请注意，其他平台上的缺省值可能不同。

NL_TO_LF
对于所有 EBCDIC 到 ASCII 的转换，将 EBCDIC NL 字符 (X'15 ') 转换为 ASCII 换行符 LF (X'0A')。

NL_TO_LF 是缺省值。

TABLE
根据在您的平台上用于所有 EBCDIC 到 ASCII 转换的转换表，转换 EBCDIC NL 字符。

此类型转换的影响可能因平台而异，也可能因语言而异; 即使在同一平台上，如果使用不同的 CCSID，行为也可能有所不同。

ISO
转换:

- 使用 TABLE 方法的 ISO CCSID
- 使用 NL_TO_CF 方法的所有其他 CCSID

第 77 页的表 10 中显示了可能的 ISO CCSID。

表 *10:* 可能的 *ISO CCSID* 列表

CCSID	**代码集**
819	ISO8859-1
912	ISO8859-2
915	ISO8859-5
1089	ISO8859-6
813	ISO8859-7
916	ISO8859-8
920	ISO8859-9
1051	roman8

如果 ASCII CCSID 不是 ISO 子集，那么 **ConvEBCDICNewline** 缺省为 NL_TO_LF。

从 IBM MQ 9.1.0 Fix Pack 2 和 IBM MQ 9.1.2 开始，您可以使用 **AMQ_CONVEBCDICNEWLINE** 环境变量而不是 **ConvEBCDICNewline** 节属性，例如，在无法使用 `mqs.ini` 文件的情况下，在客户机端提供 **ConvEBCDICNewline** 功能。 环境变量采用与 **ConvEBCDICNewline** 属性相同的值 (NL_TO_LF, TABLE 或 ISO)。 如果同时设置了属性和环境变量，那么节属性优先。

Multi ApiExit mqs.ini 文件的公共和 ApiExit 模板节

ApiExit 模板和 ApiExit 公共节标识所有队列管理器的 API 出口例程。

使用 mqs.ini 文件中的 ApiExit 模板和 ApiExit 公共节来标识所有队列管理器的 API 出口例程。 (要识别各个队列管理器的 API 出口例程，请使用 ApiExitLocal 节，如 第 92 页的『qm.ini 文件中的 ApiExitLocal 章节』中所述。)

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer Exits IBM MQ 属性页面。

Windows 在 Windows 上，还可以使用 **amqmdain** 命令来更改 API 出口的条目。

有关使用这些属性的更多信息，请参阅 配置 API 出口。

Name=ApiExit_name
传递到 MQAXP 结构中 ExitInfoName 字段的 API 出口的描述性名称。

此名称必须唯一，长度不能超过 48 个字符，且只包含对于 IBM MQ 对象名称（例如，队列名称）有效的字符。

Function=function_name
指向包含 API 出口代码的模块的函数入口点名称。 此入口点是 MQ_INIT_EXIT 函数。

此字段的长度受限于 MQ_EXIT_NAME_LENGTH。

Module=module_name
包含 API 出口代码的模块。

如果此字段包含模块的完整路径名，那么将照原样使用它。 如果此字段仅包含模块名称，那么将使用 qm.ini 文件的 ExitPath 节中的 **ExitsDefaultPath** 属性来查找模块。

在支持单独线程化库的平台上，您必须提供 API 出口模块的非线程化版本和线程化版本。 线程化的版本必须有 _r 后缀。 IBM MQ 应用程序存根的线程化版本会在加载给定模块之前将 _r 隐式附加到给定模块名称。

此字段的长度限制为平台支持的最大路径长度。

Data=data_name
要传递到 MQAXP 结构的 ExitData 字段中 API 出口的数据。

如果您包含此属性，前导和结尾的空白将被移除，而剩余的字符串则被截断为 32 个字符，并且将结果传递给出口。 如果您省略此属性，则向出口传递 32 位空白的缺省值。

此字段的最大长度是 32 个字符。

Sequence=sequence_number
调用此 API 出口的顺序与其他 API 出口有关。 先调用序号较低的出口，然后调用序号较高的出口。 出口的序列号不必连续。 1、2、3 顺序与 7、42、1096 顺序的结果相同。 如果两个出口具有相同的序列号，由队列管理器确定首先调用哪一个。 通过在 MQAXP 中 ExitChainAreaPtr 指示的 ExitChainArea 中放入时间或标记或写自己的日志文件，可以在事件后区分调用了哪个出口。

此属性是无符号的数字值。

Multi mqs.ini 文件的 DefaultQueueManager 节

DefaultQueueManager 节指定节点的缺省队列管理器。

使用 mqs.ini 文件中的 DefaultQueueManager 节来指定缺省队列管理器。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer General IBM MQ 属性页面。

名称 = *default_queue_manager*
缺省队列管理器处理未显式指定队列管理器名称的任何命令。 如果创建新的缺省队列管理器，那么将自动更新 **DefaultQueueManager** 属性。 如果您无意中创建了新的缺省队列管理器，然后想要还原为原始队列管理器，请手动变更 **DefaultQueueManager** 属性。

Multi mqs.ini 文件的 ExitProperties 节

ExitProperties 节指定队列管理器出口程序使用的配置选项。

使用 mqs.ini 文件中的 ExitProperties 节来指定队列管理器出口程序使用的配置选项。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer Extended IBM MQ 属性页面。

CLWLMode= SAFE (缺省值) | FAST
集群工作负载 (CLWL) 出口允许您指定集群中要打开的集群队列以响应 MQI 调用 (例如， MQOPEN 和 MQPUT)。 CLWL 出口以 FAST 方式或 SAFE 方式运行，具体取决于您在 **CLWLMode** 属性上指定的值。 如果省略 **CLWLMode** 属性，那么集群工作负载出口将以 SAFE 方式运行。

安全
在与队列管理器不同的进程中运行 CLWL 出口。 这是缺省值。

如果在 SAFE 方式下运行时用户编写的 CLWL 出口出现问题，那么会发生以下情况:

- CLWL 服务器进程 (amqzlwa0) 失败。
- 队列管理器重新启动 CLWL 服务器进程。
- 将在错误日志中向您报告该错误。 如果正在进行 MQI 调用，那么您将以返回码的形式接收通知。

保留队列管理器的完整性。

注: 在单独的进程中运行 CLWL 出口可能会影响性能。

FAST
在队列管理器进程中内联运行集群出口。

指定此选项可通过避免与在 SAFE 方式下运行相关联的进程切换成本来提高性能，但这样做会牺牲队列管理器完整性。 仅当您确信 CLWL 出口没有问题，并且您特别关心性能时，才应该以 FAST 方式运行 CLWL 出口。

如果 CLWL 出口以 FAST 方式运行时出现问题，那么队列管理器将失败，并且您将面临队列管理器完整性受损的风险。

Multi mqs.ini 文件的 LogDefaults 节

LogDefaults 节指定有关所有队列管理器的日志缺省值的信息。

使用 mqs.ini 文件中的 LogDefaults 节来指定有关所有队列管理器的日志缺省值的信息。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer Default log settings IBM MQ 属性页面。

如果节不存在，那么将使用 IBM MQ 缺省值。 创建队列管理器时，日志属性将用作缺省值，但如果在 **crtmqm** 命令中指定日志属性，那么可以覆盖这些日志属性。 有关此命令的更多信息，请参阅 **crtmqm**。

创建队列管理器后，将从 第 108 页的『qm.ini 文件的日志节』中描述的设置中获取该队列管理器的日志属性。

注: 为新 IBM MQ 安装提供的 LogDefaults 节不包含属性的任何显式值。 缺少属性意味着在创建新的队列管理器时将使用此值的缺省值。 LogDefaults 节的缺省值显示在 第 73 页的『AIX and Linux 的示例 mqs.ini 文件』 和 第 73 页的『Windows 的示例 mqs.ini 文件』中。 LogBufferPages 属性的值为零表示 512。

缺省前缀 (在 第 76 页的『mqs.ini 文件的 AllQueueManagers 节』中指定) 和为特定队列管理器指定的日志路径 (在 第 108 页的『qm.ini 文件的日志节』中指定) 允许队列管理器及其日志位于不同的物理驱动器上。 这是建议的方法，尽管缺省情况下它们位于同一驱动器上。

有关计算日志大小的信息，请参阅 第 532 页的『计算日志大小』。

注: 以下参数列表中给出的限制是 IBM MQ 设置的限制。 操作系统限制可能会减小可能的最大日志大小。

LogPrimaryFiles = 3 (缺省值) |*2-254* (Windows) |*2-510* (AIX and Linux)
创建队列管理器时分配的日志文件。

您可以拥有的主日志文件的最小数目为 2 ， 最大数目为 254 (在 Windows 上) 或 510 (在 AIX and Linux 上)。 缺省值为 3。

主日志文件和辅助日志文件的总数不得超过 255 (在 Windows 上) 或 511 (在 AIX and Linux 上)，并且不得小于 3。

创建或启动队列管理器时检查该值。 您可以在创建队列管理器后对其进行更改。 但是，在重新启动队列管理器之前，该值的更改不会生效，并且可能不会立即生效。

LogSecondary 文件 = 2 (缺省值) |*1-253* (Windows) |*1-509* (AIX and Linux)
耗尽主文件时分配的日志文件。

辅助日志文件的最小数目为 1，最大数目为 253 (在 Windows 上) 或 509 (在 AIX and Linux 上)。 缺省数字为 2。

主日志文件和辅助日志文件的总数不得超过 255 (在 Windows 上) 或 511 (在 AIX and Linux 上)，并且不得小于 3。

启动队列管理器时将检查该值。 您可以更改此值，但更改直到队列管理器重新启动后才会生效，即使这样也可能不会立即生效。

LogFile 页面数 = *number*
日志数据保存在一系列称为日志文件的文件中。 日志文件大小以 4 KB 页面为单位指定。

缺省日志文件页面数为 4096，给出的日志文件大小为 16 MB。

在 AIX and Linux 上，日志文件页面的最小数目为 64，而在 Windows 上，日志文件页面的最小数目为 32; 在这两种情况下，最大数目都为 65 535。

注: 无法更改队列管理器创建期间指定的日志文件的大小。

LogType= 循环 (缺省值) | 链接
要使用的日志的类型。 缺省值为 **循环**。

循环
使用日志启动重新启动恢复，以回滚在系统停止时正在进行的事务。

请参阅 第 527 页的『日志记录类型』，以获取更全面的循环日志记录说明。

线性
用于重新启动恢复和介质或正向恢复 (通过重放日志内容来创建丢失或损坏的数据)。

请参阅 第 527 页的『日志记录类型』，以获取更全面的线性日志记录说明。

如果要更改缺省值，可以编辑 LogType 属性，也可以使用 **crtmqm** 命令指定线性日志记录。

从 IBM MQ 9.1.0 开始，您可以在创建队列管理器后更改日志记录方法。 有关更多信息，请参阅 migmqlog。

LogBuffer 页面数 = 0 (缺省值) |*0-4096*
分配给缓冲区记录以进行写入的内存量，指定缓冲区大小 (以 4 KB 页面为单位)。

缓冲区页面的最小数目为 18，最大数目为 4096。 更大的缓冲区会导致更高的吞吐量，尤其是对于更大的消息。

如果指定 0 (缺省值)，那么队列管理器将选择大小为 512 (2048 KB)。

如果指定 1 到 17 范围内的数字，那么队列管理器缺省为 18 (72 KB)。 如果指定 18 到 4096 范围内的数字，那么队列管理器将使用指定的数字来设置分配的内存。

LogDefault 路径 = *directory_name*
队列管理器的日志文件所在的目录。 该目录驻留在队列管理器可以写入的本地设备上，并且最好位于与消息队列不同的驱动器上。 指定另一个驱动器将在发生系统故障时提供额外的保护。

默认值为:

- Windows *DefaultPrefix*\log for IBM MQ for Windows，其中 *DefaultPrefix* 是在 All Queue Managers IBM MQ 属性页面上的 DefaultPrefix 属性上指定的值。 此值在安装时设置。
- Linux AIX 针对 AIX 和 Linux 系统的 /var/mqm/log。

或者，可以使用 **-ld** 标志在 **crtmqm** 命令上指定目录的名称。 创建队列管理器时，还会在队列管理器目录下创建一个目录，此目录用于保存日志文件。 此目录的名称基于队列管理器名称。 这将确保日志文件路径是唯一的，并且符合目录名称长度的任何限制。

如果未在 **crtmqm** 命令上指定 **-ld**，那么将使用 mqs.ini 文件中 **LogDefaultPath** 属性的值。

队列管理器名称将附加到目录名，以确保多个队列管理器使用不同的日志目录。

创建队列管理器时，将在配置信息中的日志属性中创建 **LogPath** 值，提供队列管理器日志的完整目录名称。 此值用于在启动或删除队列管理器时查找日志。

LogWriteIntegrity=SingleWrite|DoubleWrite|TripleWrite（默认值）
记录器用于可靠地写入日志记录的方法。

TripleWrite (缺省值)
这是缺省方法。

请注意，您可以选择 **DoubleWrite**，但如果选择此项，那么系统会将此解释为 **TripleWrite**。

SingleWrite
只有在托管 "IBM MQ 恢复日志的文件系统和设备明确保证 4KB 写入的原子性时，才应使用 **SingleWrite**。

即，当写入某个 4KB 页面因任何原因而失败时，只存在两种可能的状态：映像前或映像后。 不可能存在中间状态。

注: 如果您的持久工作负载中有足够的并行性，那么设置除缺省值 **TripleWrite** 以外的任何其他值都有最小的潜在好处。

请参阅 第 110 页的『LogWriteIntegrity -使用 SingleWrite 或 TripleWrite』 以获取更多信息。

Multi mqs.ini 文件的 QueueManager 节

QueueManager 节指定队列管理器目录的位置。

每个队列管理器都有一个 QueueManager 节。 此节的属性指定队列管理器名称以及包含与该队列管理器关联的文件的目录的名称。 目录的名称基于队列管理器名称，但如果队列管理器名称不是有效的文件名，那么将进行变换。 有关名称变换的更多信息，请参阅 了解 IBM MQ 文件名。

名称 = *queue_manager_name*
队列管理器的名称。

前缀 = *prefix*
存储队列管理器文件的位置。 缺省情况下，此值与 mqs.ini 文件中 所有队列管理器 节的 **DefaultPrefix** 属性上指定的值相同。

目录 = *name*
存储队列管理器文件的 *prefix*\QMGRS 目录下的子目录的名称。 此名称基于队列管理器名称，但如果存在重复的名称或队列管理器名称不是有效的文件名，那么可以变换此名称。

DataPath= 路径
创建队列管理器时提供的显式数据路径，这将覆盖 **Prefix** 和 **Directory** 作为队列管理器数据的路径。

InstallationName= 名称
与此队列管理器关联的 IBM MQ 安装的名称。 与此队列管理器进行交互时，必须使用来自此安装的命令。

IBM i **库 = *name***
存储与此队列管理器相关的 IBM i 对象 (例如日志和日志接收器) 的库的名称。 此名称基于队列管理器名称，但如果存在重复的名称，或者如果队列管理器名称不是有效的库名，那么可以变换此名称。

V 9.2.0 **EphemeralPrefix= 名称**
存储队列管理器临时数据的位置。

缺省情况下，此值不存在，这意味着数据存储在 Prefix 位置下。

该值是在创建队列管理器时根据 **MQ_EPHEMERAL_PREFIX** 环境变量的值或 mqs.ini 文件中 AllQueueManagers 节的 **DefaultEphemeralPrefix** 属性设置的。

IBM i 由于操作系统限制，缺省临时前缀在 IBM i 上限制为 24 个字符。

相关任务

第 380 页的『使队列管理器与安装相关联』

创建队列管理器时，它将自动与发出 **crtmqm** 命令的安装相关联。 在 AIX, Linux, and Windows 上，可以使用 **setmqm** 命令来更改与队列管理器关联的安装。

Windows 高级配置和电源接口 (Advanced Configuration and Power Interface， ACPI)

Windows 支持高级配置和电源接口 (ACPI) 标准。 这使具有启用 ACPI 的硬件的 Windows 用户能够在系统进入暂挂方式并从暂挂方式恢复时停止和重新启动通道。

使用 IBM MQ Explorer 中的 ACPI IBM MQ 属性页面来指定 IBM MQ 在系统接收暂挂请求时的行为方式。

请注意，仅当 "警报监视器" 正在运行时，才会应用 ACPI IBM MQ 属性页面中指定的设置。 如果 "警报监视器" 正在运行，那么任务栏上会显示 "警报监视器" 图标。

DoDialog= Y | N
显示暂挂请求时的对话框。

DenySuspend= Y | N
拒绝暂挂请求。 如果 DoDialog= N，或者如果无法显示 DoDialog= Y 和对话框 (例如，因为笔记本外盖已关闭)，那么将使用此选项。

CheckChannelsRunning=Y | N
检查是否有任何通道正在运行。 结果可以确定其他设置的结果。

下表概述了这些参数的每个组合的影响:

DoDialog	DenySuspend	CheckChannels 正在运行	操作
N	N	N	接受暂挂请求。
N	N	Y	接受暂挂请求。
N	Y	N	拒绝暂挂请求。
N	Y	Y	如果有任何通道正在运行，那么将拒绝暂挂请求; 如果不接受该请求。
Y	N	N	显示对话框 (请参阅 注释 ; 接受暂挂请求)。 这是缺省值。
Y	N	Y	如果没有正在运行的通道接受暂挂请求; 如果它们显示在对话框中 (请参阅 注释); 接受请求)。
Y	Y	N	显示对话框 (注释 ; 拒绝暂挂请求)。
Y	Y	Y	如果没有正在运行的通道接受暂挂请求; 如果它们显示的是对话框 (注意 ; 拒绝请求)。

注: 在操作要显示对话框的情况下，如果无法显示对话框 (例如，由于笔记本盖已关闭)，那么将使用 DenySuspend 选项来确定是接受还是拒绝暂挂请求。

ALW 队列管理器配置文件， qm.ini

队列管理器配置文件 qm.ini 包含与特定队列管理器相关的信息。

每个队列管理器都有一个队列管理器配置文件。 创建与其关联的队列管理器时，将自动创建 qm.ini 文件。

注: 有关如何以及何时编辑 qm.ini 文件以及您对该文件所作的任何更改何时生效的更多信息，请参阅 第 71 页的『在 Multiplatforms 版上的 .ini 文件中更改 IBM MQ 配置信息』。

从 IBM MQ 9.0.4 和 IBM MQ 9.0.0 Fix Pack 2 开始， **strmqm** 命令在完全启动队列管理器之前检查 qm.ini 文件中的 CHANNELS 和 SSL 节的语法，这使您更容易看到错误，并在 **strmqm** 发现 qm.ini 文件包含任何错误时快速进行更正。 有关更多信息，请参阅 strmqm。

qm.ini 文件的位置

Linux AIX 在 AIX and Linux 系统上， qm.ini 文件保存在队列管理器所占用的目录树的根目录中。 例如，队列管理器 QMNAME 的配置文件的路径和名称是:

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

Windows 在 Windows 系统上， qm.ini 文件的位置由 HKLM\SOFTWARE\IBM\WebSphere MQ 键中指定的 WorkPath 提供。 例如，名为 QMNAME 的队列管理器的配置文件的路径和名称如下所示:

```
CMQ\qmgrs\QMNAME\qm.ini
```

IBM i qm.ini 文件保存在 *mqmdata directory*/*QMNAME*/qm.ini 中，其中缺省情况下 *mqmdata directory* 是 /QIBM/UserData/mqm， *QMNAME* 是应用初始化文件的队列管理器的名称。

注: 您可以在 mqs.ini 文件中更改 *mqmdata directory* 。

队列管理器名称的长度最多可以为 48 个字符。 但是，这并不保证名称有效或唯一。 因此，将根据队列管理器名称生成目录名称。 此过程称为 名称变换。 有关描述，请参阅 IBM MQ 文件名 和 IBM i 上的对象名。

qm.ini 节

注意:

- 本主题链接到 qm.ini 文件中有关节的更多信息。 每个节都包含有关该节中的参数的信息，包括适当的示例。
- 每个节都显示该节所应用于的一个或多个 IBM MQ for Multiplatforms 平台。

Multi V 9.2.0 启动时自动配置 qm.ini

从 IBM MQ 9.2.0 开始，您可以将队列管理器配置为在每次队列管理器启动时自动应用包含 qm.ini 覆盖的文件或文件集的内容。

您可以使用此选项来具有可以在下次队列管理器重新启动时修改并自动重放的配置。 例如，如果 qm.ini 覆盖位于已安装的驱动器上，那么可能具有集中配置，在此配置中，最新版本将在每个队列管理器启动时应用于这些队列管理器。

您可以使用此功能通过使用自动集群功能来简化创建统一集群的过程。 有关此示例，请参阅 第 346 页的『创建新的统一集群』。

注: 这些覆盖仅在队列管理器启动时应用，并且不会影响队列管理器的创建。 例如，不能使用此函数设置主日志文件数。

开始之前

您可以使用:

1. 单个文件，并创建包含对 qm.ini 文件的更改的文本文件。
2. 一组 qm.ini 格式文件:
   - 标识将存在配置的目录，以及
   - 在该目录中，创建文件，每个文件都具有扩展名 .ini，例如 qminisettings.ini。

一个或多个文件只需要包含节，并且 **attribute**=*value* 设置用于更改的项。 例如，要更新 Channels 节中的 **MaxChannels** 属性，该文件可能包含:

```
Channels:
 MaxChannels=1234
```

请注意，在 qm.ini 覆盖文件中，以 # 作为前缀的任何行都将被视为注释。

启用 qm.ini 文件属性的自动配置

您可以通过将 **-ii** 标志用于 **crtmqm** 命令并指向特定文件或目录来配置新的队列管理器。 提供的值作为属性 **IniConfig** 存储在 qm.ini 文件中的 AutoConfig 节下。

您可以通过添加指向有效文件或目录的 AutoConfig 节属性 **IniConfig** 来配置现有队列管理器以启用自动 MQSC 配置。 例如:

```
AutoConfig:
   IniConfig=C:\MQ_Configuration\uniclus.ini
```

自动配置如何工作?

在队列管理器启动期间，将验证 AutoConfig 节属性 **IniConfig** 所标识的配置，以确保语法有效，然后将其作为单个 cached.ini 文件存储在队列管理器数据树中的 autocfg 目录中。

在处理一个目录中的多个文件时，将按字母顺序处理这些文件。

在首次启动队列管理器期间，无法读取文件或目录会阻止队列管理器启动，并向控制台和队列管理器错误日志显示相应的错误消息。

在后续重新启动时，如果指向的文件或目录不可读，那么将使用先前高速缓存的文件，并且写入队列管理器错误日志的消息会突出显示此内容。

使用 **strmqm** 命令时，在调用队列管理器之前，会将 cached.ini 文件的内容作为覆盖项应用于 qm.ini 文件。

这确实意味着对于备用队列管理器，将在处理 **strmqm** 命令时读取设置，而不是在队列管理器变为活动时读取设置。

如何构建替换 qm.ini 文件?

首次配置自动初始化配置并启动队列管理器时，会将当前 qm.ini 文件的副本作为 base_qm.ini 复制到队列管理器的数据目录内的 autoconfig 子目录中。 从这里开始，这被视为基线。

在每次队列管理器启动 (即 **strmqm** 时间) 时，将废弃当前活动的 qm.ini 文件，并将其替换为 base_qm.ini 的副本。 然后， cached.ini 文件中的配置将应用于此文件。

一旦队列管理器处于自动配置控制之下，应该通过一个或多个文件 (指向使用 AutoConfig 节中的 **IniConfig** 属性) 来执行对 qm.ini 文件的所有更改。

由于将在队列管理器启动时除去现有 qm.ini 文件，因此仅将所提供的 qm.ini 文件中使用 **IniConfig** 属性的配置应用于队列管理器基本行。

如果在队列管理器的先前启动时通过自动初始化配置更改了节或属性，那么除非仍在 **IniConfig** 属性所标识的文件中标识这些更改，否则将除去这些更改。

由于在队列管理器启动时重新创建了 qm.ini 文件，这意味着将丢失对 qm.ini 文件的任何手动更改。 如果确实需要使更改具有持久性，并且无法使用 **IniConfig** 属性进行该更改，那么可以执行以下任一操作:

- 对 base_qm.ini 文件本身进行更改。
- 删除 base_qm.ini 文件。

  如果删除此文件，那么将在下一个队列管理器启动时根据 qm.ini 文件的当前内容重新创建 base_qm.ini。 这将 硬化 所有当前更改，作为未来启动的新基线。

Multi qm.ini 文件节和属性的摘要

队列管理器配置文件 qmi.ini 的节的属性摘要，其中包含指向更多信息的链接。

表 11: *qm.ini* 文件的节	
节和属性	**属性的描述**
Windows **AccessMode 节**	
Windows 访问组 [1]	Windows 安全组，其成员将被授予对所有队列管理器数据文件的完全访问权。
ApiExitLocal 节	
Name	传递到 MQAXP 结构中 ExitInfoName 字段的 API 出口的描述性名称。
函数	指向包含 API 出口代码的模块的函数入口点名称。
模块	包含 API 出口代码的模块。
数据	要传递到 MQAXP 结构的 ExitData 字段中 API 出口的数据。
Sequence	调用此 API 出口的顺序与其他 API 出口有关。
V 9.2.0 **AutoCluster 节**	
V 9.2.0 类型	自动集群的类型。 唯一有效的选项是 Uniform ， 它表示统一集群。
V 9.2.0 ClusterName	自动集群的名称。
V 9.2.0 RepositoryName1	自动集群中第一个完整存储库的队列管理器名称。
V 9.2.0 Repository1Conname	自动集群成员应如何连接到队列管理器的连接名称 (CONNAME) 值。
V 9.2.0 RepositoryName2	自动集群中第二个完整存储库的队列管理器名称。
V 9.2.0 Repository2Conname	自动集群成员应如何连接到队列管理器的连接名称 (CONNAME) 值。
V 9.2.0 **AutoConfig 节**	
V 9.2.0 MQSCConfig	完整文件路径或目录路径，其中所有 *.mqsc 文件都将在每个队列管理器启动时应用于队列管理器。
V 9.2.0 IniConfig	完整文件路径或目录路径，其中所有 *.ini 文件都将应用于每个队列管理器启动时的 qm.ini 文件。
通道节	
MaxChannels	允许的当前通道的最大数目。
MaxActiveChannels	允许在任何时间处于活动状态的最大通道数。
MaxInitiators	发起方的最大数目。
MQIBindType	应用程序的绑定。
IBM i ThreadedListener	是否启动 RUNMQLSR 或 AMQCLMAA 作为监听器。
PipeLineLength	通道将使用的最大并发线程数。

表 11: qm.ini 文件的节 (继续)	
节和属性	**属性的描述**
AdoptNewMCA	哪些类型的通道可以停止现有通道实例，以便当 IBM MQ 接收到启动通道的请求时，可以启动新的通道实例，但发现该通道的实例已在运行。
AdoptNewMCATimeout	新通道实例等待旧通道实例结束的时间量 (以秒计)。
AdoptNewMCACheck	启用 **AdoptNewMCA** 属性时所需的检查类型。
ChlauthEarlyAdopt	处理连接认证和通道认证规则的顺序。
PasswordProtection	在 MQCSP 结构中设置受保护的密码，而不是使用 TLS。
IgnoreSeqNumberMismatch	控制队列管理器在通道启动期间如何处理序号不匹配。
连接节	
DefaultBindType	在不同进程中运行的应用程序和队列管理器是共享某些资源，还是在它们之间不共享任何资源。
DiagnosticMessages 节	
name	节的名称。
服务	此节正在启用的服务。
ExcludeMessage	不写入队列管理器错误日志的消息。
SuppressMessage	将仅在指定时间间隔内写入队列管理器错误日志一次的消息。
IBM i SuppressInterval	仅将 **SuppressMessage** 中指定的消息写入队列管理器错误日志一次的时间间隔 (以秒计)。
严重性	严重性级别的逗号分隔列表。
FilePath	写入日志文件的路径。 (仅当服务属性设置为 "文件" 时才受支持。)
FilePrefix	日志文件的前缀。 (仅当服务属性设置为 "文件" 时才受支持。)
FileSize	日志回滚的大小。 (仅当服务属性设置为 "文件" 时才受支持。)
FORMAT	文件的格式。 (仅当服务属性设置为 "文件" 时才受支持。)
Linux AIX 系统日志	使用 JSON 格式 诊断消息规范将任何未过滤的消息发送到系统日志的 Syslog 服务。
Linux AIX 标识	与系统日志条目关联的 ident 值。 (仅当服务属性设置为 Syslog 时才支持。)
ExitPath 节	
ExitsDefaultPath	队列管理器系统 (32 位) 上用户出口程序的路径。
ExitsDefaultPath64	队列管理器系统 (64 位) 上的用户出口程序的路径。
ExitPropertiesLocal 节段	
CLWLMode	集群工作空间 (CLWL) 出口是否以 FAST 方式或 SAFE 方式运行。
IBM i Linux AIX **文件系统节**	

表 11: qm.ini 文件的节 (继续)	
节和属性	**属性的描述**
IBM i Linux AIX ValidateAuth	允许不是 mqm 组成员的用户访问错误目录和文件。
日志节	
LogPrimaryFiles	创建队列管理器时分配的日志文件。
LogSecondaryFiles	耗尽主文件时分配的日志文件。
LogFilePages	日志文件页数。 (日志文件大小以 4 KB 页面为单位指定。)
LogType	队列管理器要使用的日志记录类型 (循环或线性)。
LogBufferPages	分配给缓冲区记录以进行写入的内存量，指定缓冲区大小 (以 4 KB 页面为单位)。
LogPath	队列管理器的日志文件所在的目录。
LogWriteIntegrity	记录器用于可靠地写入日志记录的方法。
LogManagement	用于手动或由队列管理器管理日志扩展数据块的方法。
Windows **LU62 节**	
Windows TPName	要在远程站点上启动的 TP 名称。
Windows Library1	APPC DLL 的名称。
Windows Library2	与 Library1 相同，如果代码存储在两个单独的库中。
CP4I **NativeHAInstance 节**	
第 111 页的『名称』	创建队列管理器实例时使用的实例名称。
第 111 页的『ReplicationAddress』	实例的主机名， IPv4 点分十进制或 IPv6 十六进制格式地址。
CP4I **NativeHALocalInstance 节段**	
第 111 页的『LocalName』	NativeHALocalInstancestanza 的名称，取自创建 Native HA 队列管理器时指定的日志副本实例名称。
第 112 页的『KeyRepository』	保存要用于保护日志复制流量的数字证书的密钥存储库的位置。
第 112 页的『CertificateLabel』	用于标识用于保护日志复制流量的数字证书的证书标签。
第 112 页的『CipherSpec』	用于保护日志复制流量的 MQ CipherSpec 。
第 112 页的『LocalAddress』	接受日志复制流量的本地网络接口地址。
第 112 页的『HeartbeatInterval』	脉动信号间隔定义本机 HA 队列管理器的活动实例发送网络脉动信号的频率 (以毫秒计)。
第 112 页的『HeartbeatTimeout』	脉动信号超时定义本机 HA 队列管理器的副本实例在确定活动实例无响应之前等待的时间长度。
第 112 页的『RetryInterval』	重试时间间隔定义本机 HA 队列管理器应重试失败复制链接的频率 (以毫秒计)。
Windows **NETBIOS 节**	

表 11: qm.ini 文件的节 (继续)	
节和属性	**属性的描述**
Windows LocalName	LAN 上用于识别此机器的名称。
Windows AdapterNum	LAN 适配器的编号。
Windows NumSess	要分配的会话数。
Windows NumCmds	要分配的命令数。
Windows NumNames	要分配的名称数。
Windows Library1	NetBIOS DLL 的名称。
QMErrorLog 节	
ErrorLogSize	指定将其复制到备份的队列管理器错误日志的大小。
ExcludeMessage	指定不写入队列管理器错误日志的消息。
SuppressMessage	指定仅在指定的时间间隔内写入队列管理器错误日志一次的消息。
SuppressInterval	指定 SuppressMessage 中指定的消息仅写入队列管理器错误日志一次的时间间隔 (以秒计)。
Linux AIX **受限方式节** [2]	
Linux AIX ApplicationGroup	本地传输队列的名称，如果未显式为其目标定义传输队列，那么将放置远程消息。
安全性节	
ClusterQueueAccessControl	检查集群队列或集群队列管理器上托管的标准队列的访问控制。
Windows GroupModel	在 Windows 上确定用户的组成员资格时，对象权限管理器 (OAM) 是否检查全局组。
服务节	
Name	必需服务的名称。
EntryPoints	为服务定义的入口点数。
Windows SecurityPolicy	在 Windows 上，每个队列管理器的安全策略
Linux AIX SecurityPolicy	在 AIX and Linux 上，队列管理器是使用基于用户的授权还是基于组的授权。 V 9.2.1 从 IBM MQ 9.2.1 开始，您还可以创建非操作系统用户名。
SharedBindingsUserId	仅对于共享绑定，无论是来自 MQZ_AUTHENTICATE_USER 函数的 IdentityContext 结构中的 UserIdentifier 字段是有效用户标识还是实际用户标识。

表 11: qm.ini 文件的节 (继续)	
节和属性	**属性的描述**
FastpathBindingsUserId	仅对于快速路径绑定，无论是来自 MQZ_AUTHENTICATE_USER 函数的 IdentityContext 结构中的 UserIdentifier 字段是有效用户标识还是实际用户标识。
IsolatedBindingsUserId	仅对于隔离绑定，无论是 IdentityContext 结构中的 UserIdentifier 字段 (来自 MQZ_AUTHENTICATE_USER 函数) 是有效用户标识还是实际用户标识。
ServiceComponent 节	
服务	必需服务的名称。
Name	服务组件的描述性名称。
模块	将包含这个组件代码的模块的名称。
ComponentDataSize	每次调用时传递到组件的组件数据区的大小 (以字节计)。
Windows **SPX 节**	
Windows 套接字	以十六进制表示法表示的 SPX 套接字号。
Windows BoardNum	LAN 适配器号。
Windows KeepAlive	打开或关闭 KeepAlive 功能。
Windows Library1	SPX DLL 的名称。
Windows Library2	与 LibraryName1 相同，如果代码存储在两个单独的库中。
Windows ListenerBacklog	覆盖 SPX 侦听器的缺省未完成请求数。
SSL 节	
V 9.2.1 OutboundSNI	指定支持 SNI 的客户机在启动 TLS 连接时是将 SNI 设置为远程系统的目标 IBM MQ 通道名称还是设置为主机名。
AllowOutboundSNI	指定在启动 TLS 连接时，支持 SNI 的客户机是否将 SNI 设置为远程系统的目标 IBM MQ 通道名称。 **注意:** V 9.2.1 从 IBM MQ 9.2.1 开始，不推荐使用此属性，请改为使用 **OutboundSNI** 。
V 9.2.0 AllowedCipherSpecs	指定订购并支持与 Multiplatforms 版上的 IBM MQ 通道配合使用的 CipherSpecs 的定制列表。
V 9.2.0 AllowTLSV13	队列管理器是否能够使用 TLS 1.3 CipherSpecs。
CDPCheckExtensions	队列管理器上的 TLS 通道是否尝试检查在 CrlDistributionPoint 证书扩展名中命名的 CDP 服务器。
V 9.2.0 MinimumRSAKeySize	指定 RSA 证书必须具有才能接受的最小密钥大小。
OCSPAuthentication	无法从 OCSP 服务器确定撤销状态时要执行的操作。
OCSPCheckExtensions	队列管理器上的 TLS 通道是否尝试检查 AuthorityInfoAccess 证书扩展名中命名的 OCSP 服务器。

表 11: qm.ini 文件的节 (继续)	
节和属性	**属性的描述**
V 9.2.0 OCSPTimeout	执行撤销检查时等待 OCSP 响应程序的秒数。
ALW V 9.2.0 PeerCertChainValidation	IBM Global Security Kit (GSKit) 证书验证设置。
SSLHTTPProxyName	要由 GSKit 用于 OCSP 检查的 HTTP 代理服务器的主机名或网络地址。
V 9.2.0 SSLHTTPConnectTimeout	执行撤销检查时等待成功建立到 HTTP 服务器的网络连接的秒数。
子池节 第 91 页的『3』	此节由 IBM MQ 创建。 请勿更改。
ShortSubpoolName 第 91 页的『3』	对应于在 /var/mqm/sockets 目录中创建的目录和符号链接的名称， IBM MQ 使用该目录和链接在其正在运行的进程之间进行内部通信。
TCP 节	
端口	TCP/IP 会话的缺省端口号 (十进制表示法)。
Windows Library1	TCP/IP 套接字 DLL 的名称。
KeepAlive	打开或关闭 KeepAlive 功能。
ListenerBacklog	覆盖 TCP/IP 侦听器的缺省未完成请求数。
Connect_Timeout	尝试连接套接字超时之前的秒数。
SndBuffSize	通道发送端使用的 TCP/IP 发送缓冲区的大小 (以字节计)。
RcvBuffSize	通道接收端使用的 TCP/IP 接收缓冲区的大小 (以字节计)。
RcvSndBuffSize	接收方通道的发送方端使用的 TCP/IP 发送缓冲区的大小 (以字节为单位)。
RcvRcvBuffSize	接收方通道的接收端使用的 TCP/IP 接收缓冲区的大小 (以字节计)。
SvrSndBuffSize	客户机连接服务器连接通道的服务器端使用的 TCP/IP 发送缓冲区的大小 (以字节计)。
SvrRcvBuffSize	客户机连接服务器连接通道的服务器端使用的 TCP/IP 接收缓冲区的大小 (以字节计)。
Multi V 9.2.4 SecureCommsOnly	指定是允许纯文本通信，缺省值还是不允许纯文本通信。
调整参数节	
SuppressDspAuthFail	如果连接缺少对对象的 + dsp 权限，那么队列管理器是否禁止生成授权事件以及在授权检查失败时将 AMQ8077 错误消息写入错误日志。
ImplSyncOpenOutput	在可以对同步点之外的持久放置启用隐式同步点之前，为放置打开队列的应用程序的最小数目。
V 9.2.0 UniformClusterName	要用作统一集群的 IBM MQ 集群的名称。
V 9.2.0.4 V 9.2.4 OAMLdapConnectTimeout	LDAP 客户机将等待与服务器建立 TCP 连接的最长时间 (以秒为单位)。

表 11: qm.ini 文件的节 (继续)	
节和属性	**属性的描述**
V 9.2.0.4 V 9.2.4 OAMLdapQueryTimeLimit	LDAP 客户机将等待从服务器接收对 LDAP 请求的响应的最长时间 (以秒计)。
V 9.2.0.10 OAMLdapResponseWarningTime	如果与 LDAP 服务器的连接所用时间超过 **OAMLdapResponseWarningTime** 参数指定的阈值秒数，那么会将 AMQ5544W 消息写入错误日志。
ExpiryInterval	指示队列管理器扫描队列以查找其他队列活动尚未清除的到期消息的频率。 这是时间间隔 (以秒为单位)。
V 9.2.0 **Variables 节**	
V 9.2.0 *attribute=value*	在 MQSC 定义期间用作插入的名称和关联值。
XAResourceManager 节	
Name	资源管理器实例。
SwitchFile	包含资源管理器的 XA 切换结构的装入文件的标准名称。
XAOpenString	要传递到资源管理器的 xa_open 入口点的数据字符串。
XACloseString	要传递到资源管理器的 xa_close 入口点的数据字符串。
ThreadOfControl	当队列管理器需要从其自己的多线程进程之一调用资源管理器时，队列管理器用于序列化的值。 对于 Windows 是必需的。

注意:

1. AccessMode 节由 **crtmqm** 命令上的 **-a [r]** 选项设置。 创建队列管理器后，请勿更改 AccessMode 节。
2. RestrictedMode 节由 **crtmqm** 命令上的 **-g** 选项设置。 创建队列管理器后，请勿更改此节。 如果不使用 **-g** 选项，那么不会在 qm.ini 文件中创建该节。
3. 创建队列管理器时，"IBM MQ 会自动写入 Subpool stanza 以及该 stanza 中的属性 ShortSubpoolName。 IBM MQ 为 ShortSubpoolName 选择了一个值。 请勿变更此值。

Windows qm.ini 文件的 AccessMode 节

访问方式仅适用于 Windows 服务器。 qm.ini 文件的 AccessMode 节由 **crtmqm** 命令上的 -a [r] 选项设置。 创建队列管理器后，请勿更改 AccessMode 节。

使用访问组 (-a [r]) **crtmqm** 命令的选项，用于指定 Windows 安全组，该安全组的成员将被授予对所有队列管理器数据文件的完全访问权。 根据使用的语法，该组可以是本地组或全局组。 组名的有效语法如下:

LocalGroup
域名/*GlobalGroup* 名称
GlobalGroup 名称 @ 域名

在运行带有 -a [r] 选项的 **crtmqm** 命令之前，必须定义其他访问组。

如果使用 -ar 而不是 -a 指定组，那么不会授予本地 mqm 组对队列管理器数据文件的访问权。 如果托管队列管理器数据文件的文件系统不支持本地定义的组的访问控制条目，请使用此选项。

该组通常是全局安全性组，用于使多实例队列管理器能够访问共享队列管理器数据和日志文件夹。 使用额外的安全访问组可在文件夹上设置读写许可权，或共享包含的队列管理器数据和日志文件。

附加安全访问组是使用名为 mqm 的本地组设置对包含队列管理器数据和日志的文件夹的许可权的替代方法。 与本地组 mqm 不同，您可以将其他安全访问组设置为本地组或全局组。 它必须是全局组，才能在包含多实例队列管理器使用的数据和日志文件的共享文件夹上设置许可权。

Windows 操作系统会检查读写队列管理器数据和日志文件的访问许可权。 它将检查运行队列管理器进程的用户标识的许可权。 检查的用户标识取决于您是将队列管理器作为服务启动还是以交互方式将其启动。 如果是将队列管理器作为服务启动的，那么 Windows 系统检查的用户标识是使用“**准备** IBM MQ”向导配置的用户标识。 如果是以交互方式启动队列管理器的，那么 Windows 系统检查的用户标识是运行 **strmqm** 命令的用户标识。

该用户标识必须是本地 mqm 组的成员才能启动队列管理器。 如果该用户标识是额外的安全访问组的成员，那么队列管理器可以通过使用该组读写提供了许可权的文件。

限制: 只能在 Windows 操作系统上指定额外的安全访问组。 如果在其他操作系统上指定额外的安全访问组，那么 **crtmqm** 命令将返回错误。

示例节

```
AccessMode:
SecurityGroup=wmq\wmq
```

相关概念

第 444 页的『保护 Windows 上的非共享队列管理器数据和日志目录和文件』
第 441 页的『保护 Windows 上的共享队列管理器数据和日志目录及文件』

相关任务

第 419 页的『在 Windows 上的域工作站或服务器上创建多实例队列管理器』

相关参考

crtmqm (创建队列管理器)

Multi qm.ini 文件中的 ApiExitLocal 章节

ApiExitLocal 节指定队列管理器的 API 退出例程。

对于服务器，修改 "qm.ini 文件的 ApiExitLocal 节，以确定队列管理器的 API 退出例程。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer Exits 队列管理器属性页面。

对于客户端，修改 "mqclient.ini 文件中的 ApiExitLocal 节段，以确定队列管理器的 API 退出例程。

概述

ApiExitLocal 节仅允许指定单个 Module，但需要提供四个模块，如下所示:

- 32 位无线程
- 32 位线程
- 64 位非线程
- 64 位线程

请注意， IBM MQ 会将 _r 附加到提供的模块名称以标识出口的线程版本，但 IBM MQ 不会为 32 位和 64 位变体提供直接等效机制。

prefix/mqm/samp/bin 中提供的 amqsaxe0 和 amqsaxe0_r 版本是针对构建它们的平台上的队列管理器本机大小 (现在全部为 64 位) 而构建的，并且只能由以相同本机大小运行的应用程序使用。

如果提供了未限定的模块名称，那么 IBM MQ 将在 /var/mqm/exits 中查找 32 位变体，在 /var/mqm/exits64 中查找 64 位变体

例如， module=amqsaxe 表示:

```
/var/mqm/exits/amqsaxe  - 32 bit unthreaded variant
/var/mqm/exits/amqsaxe_r - 32 bit threaded variant
/var/mqm/exits64/amqsaxe - 64 bit unthreaded variant
/var/mqm/exits64/amqsaxe_r - 64 bit threaded variant
```

Windows 在 Windows 系统上，还可以使用 **amqmdain** 命令来更改 API 出口的条目。 (要识别所有队列管理器的 API 退出例程，可使用 ApiExitCommon 和 ApiExitTemplate 节段，如 "第 77 页的『ApiExit mqs.ini 文件的公共和 ApiExit 模板节』所述）

请注意，要使 API 出口正常工作，必须将来自服务器的消息发送到未转换的客户机。 在 API 出口处理消息后，必须在客户机上转换消息。 因此，这要求您已在客户机上安装所有转换出口。

有关使用这些属性的更多信息，请参阅 配置 API 出口。

参数

Name=ApiExit_name
传递到 MQAXP 结构中 ExitInfoName 字段的 API 出口的描述性名称。

此名称必须唯一，长度不能超过 48 个字符，且只包含对于 IBM MQ 对象名称（例如，队列名称）有效的字符。

Function=function_name
指向包含 API 出口代码的模块的函数入口点名称。 此入口点是 MQ_INIT_EXIT 函数。

此字段的长度受限于 MQ_EXIT_NAME_LENGTH。

Module=module_name
包含 API 出口代码的模块。

如果此字段包含模块的完整路径名，那么将照原样使用它。 如果此字段仅包含模块名称，那么将使用 qm.ini 文件的 ExitPath 节中的 **ExitsDefaultPath** 属性来查找模块。

在支持单独线程化库的平台上，您必须提供 API 出口模块的非线程化版本和线程化版本。 线程化的版本必须有 _r 后缀。 IBM MQ 应用程序存根的线程化版本会在加载给定模块之前将 _r 隐式附加到给定模块名称。

此字段的长度限制为平台支持的最大路径长度。

Data=data_name
要传递到 MQAXP 结构的 ExitData 字段中 API 出口的数据。

如果您包含此属性，前导和结尾的空白将被移除，而剩余的字符串则被截断为 32 个字符，并且将结果传递给出口。 如果您省略此属性，则向出口传递 32 位空白的缺省值。

此字段的最大长度是 32 个字符。

Sequence=sequence_number
调用此 API 出口的顺序与其他 API 出口有关。 先调用序号较低的出口，然后调用序号较高的出口。 出口的序列号不必连续。 1、2、3 顺序与 7、42、1096 顺序的结果相同。 如果两个出口具有相同的序列号，由队列管理器确定首先调用哪一个。 通过在 MQAXP 中 ExitChainAreaPtr 指示的 ExitChainArea 中放入时间或标记或写自己的日志文件，可以在事件后区分调用了哪个出口。

此属性是无符号的数字值。

示例节

```
ApiExitLocal:
   Name=ClientApplicationAPIchecker
   Sequence=3
   Function=EntryPoint
   Module=/usr/Dev/ClientAppChecker
   Data=9.20.176.20
```

Multi V 9.2.0 qm.ini 文件的 AutoCluster 节

当队列管理器开始识别集群是否是自动集群的成员，并且可以识别集群的完整存储库时，将使用 AutoCluster 节。

以下属性对于 AutoCluster 节是必需的:

类型 =统一
指定自动集群的类型，唯一有效的选项是 统一，它表示统一集群。

群集名称=<字符串
集群的名称，即自动集群名称。

以下属性是 AutoCluster 节的可选属性，但您必须成对提供这些属性:

存储库名称 1 =*<String>*
这是自动集群中第一个完整存储库的队列管理器名称。 这可以是此队列管理器的名称，也可以是另一个队列管理器的名称。

存储库 1Conname=*<Connection Name String>*
这是自动集群成员应如何连接到此队列管理器的连接名称 (CONNAME) 值。

存储库名称 =*<String>*
这是自动集群中第二个完整存储库的队列管理器名称。 这可以是此队列管理器的名称，也可以是另一个队列管理器的名称。

存储库 2Conname=*<Connection name string>*
这是自动集群成员应如何连接到此队列管理器的连接名称 (CONNAME) 值。

示例节

```
AutoCluster:
        Repository1Name=QM1
        Repository2Name=QM2
        Repository1Conname=127.0.0.1(1414)
        Repository2Conname=127.0.0.1(1415)
        ClusterName=UNIFORMCLUSTER1
        Type=Uniform
```

相关概念
第 337 页的『自动应用程序均衡』
自动应用程序均衡通过启用 IBM MQ 统一集群来密切管理整个集群中的应用程序分发，而不是依赖于随机化或手动将应用程序锁定到特定队列管理器，从而大大增强了应用程序分发和可用性。

相关任务
第 346 页的『创建新的统一集群』
如何创建新的统一集群。

相关参考
第 349 页的『使用自动集群配置』
您可以通过更改 qm.ini 配置信息来配置 IBM MQ 以启用自动配置。

Multi V 9.2.0 qm.ini 文件的 AutoConfig 节

AutoConfig 节的属性经常用作设置统一集群的一部分。

注: 只能将 AutoCluster 节用于统一集群。

MQSCConfig=*<Path>*
该路径可以是完整文件路径，也可以是目录的路径，在此路径中，所有 *.mqsc 文件都将在每次队列管理器启动时应用于队列管理器。

有关更多信息，请参阅 在启动时从 MQSC 脚本自动配置。

IniConfig=<Path>
该路径可以是完整文件路径，也可以是目录的路径，在此路径中，所有 `*.ini` 文件都将应用于每个队列管理器启动时的 `qm.ini` 文件。

有关更多信息，请参阅第 83 页的『启动时自动配置 qm.ini』。

V 9.2.2 **ConfigTimeout**
队列管理器等待自动配置完成的值 (以秒计)。 在该时间之后，队列管理器将继续启动，并且可供应用程序连接。

缺省行为是无超时。 这意味着在完成所有自动配置命令之前，队列管理器不可供应用程序连接。

您不应该仅仅因为配置需要很长时间而配置此属性，因为应用程序可能能够在完成应用于它们的配置 (例如，创建应用程序所需的队列) 之前进行连接。

示例节

```
AutoConfig:
        MQSCConfig=/tmp/auto.mqsc
        IniConfig=/tmp/auto.ini
        ConfigTimeout=120
```

相关概念
第 337 页的『自动应用程序均衡』
自动应用程序均衡通过启用 IBM MQ 统一集群来密切管理整个集群中的应用程序分发，而不是依赖于随机化或手动将应用程序锁定到特定队列管理器，从而大大增强了应用程序分发和可用性。

相关任务
第 346 页的『创建新的统一集群』
如何创建新的统一集群。

相关参考
第 349 页的『使用自动集群配置』
您可以通过更改 `qm.ini` 配置信息来配置 IBM MQ 以启用自动配置。

Multi qm.ini 文件的通道节

Channels 节的属性确定通道的配置。

z/OS 此信息不适用于 IBM MQ for z/OS。

使用 `qm.ini` 文件中的 `CHANNELS` 节来指定有关通道的信息。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer `Channels` 队列管理器属性页面。

MaxChannels= 100（默认）/*number*
允许的 当前 通道的最大数量。

缺省值是 100。

您可以将 **MaxChannels** 设置为其他值，以限制当前通道的最大数量 (如果需要)。 对于 IBM MQ Appliance，缺省值为 999 999 999 ，不应更改。

MaxActiveChannels= *MaxChannels_value*
允许在任何时间处于 活动 状态的最大通道数。 缺省值是为 **MaxChannels** 属性指定的值。

MaxInitiators= 3（默认）/编号
发起方的最大数目。 缺省值和最大值都是 3。

MQIBindType=FASTPATH|STANDARD
应用程序的绑定:

FASTPATH
通道使用 MQCONNX FASTPATH 进行连接; 没有代理程序进程。

标准
使用 STANDARD 连接通道。

IBM i **ThreadedListener= NO|YES**
是启动 RUNMQLSR（**是**）还是 AMQCLMAA（**否**）作为监听器。

如果指定 ThreadedListener=YES,所有通道都将作为单个作业的线程运行。 这就将连接数限制在单个作业可用资源的范围内。

如果指定 "ThreadedListener=NO,，非线程监听器 (AMQCLMAA) 会为每个入站 TCP/IP 通道启动一个新的应答器任务 (AMQCRSTA)。 这种技术的缺点是，启动一个新的 AMQCRSTA 作业不如在 RUNMQLSR 作业中启动一个线程快，因此非线程监听器的连接时间比线程监听器慢。

PipeLineLength = 1 |*number*
通道将使用的最大并发线程数。 缺省值是 1。 任何大于 1 的值都将被视为 2。

使用管道时，请将通道两端的队列管理器配置为具有大于 1 的 **PipeLineLength** 。

注: 管道传输仅对 TCP/IP 通道有效。

请参阅 多线程支持-流水线 以获取更多信息。

AdoptNewMCA= NO（默认） |SVR|SDR|RCVR|CLUSRCVR|ALL|FASTPATH
如果 IBM MQ 接收到启动通道的请求，但发现通道的实例已在运行，那么在某些情况下，必须先停止现有通道实例，然后才能启动新的通道实例。 **AdoptNewMCA** 属性允许您控制可以通过此方式结束哪些类型的通道。

如果为特定通道类型指定 **AdoptNewMCA** 属性，但由于匹配的通道实例已在运行，因此新通道无法启动:

1. 新通道尝试通过请求其结束来停止前一个通道。
2. 如果先前通道服务器在 **AdoptNewMCATimeout** 等待时间间隔到期时未响应此请求，那么先前通道服务器的线程或进程将结束。
3. 如果先前的通道服务器在步骤 2 之后尚未结束，并且在 **AdoptNewMCATimeout** 等待时间间隔第二次到期之后， IBM MQ 将以 CHANNEL IN USE 错误结束通道。

AdoptNewMCA 功能适用于服务器，发送方，接收方和集群接收方通道。 对于发送方或服务器通道，只有一个具有特定名称的通道实例可以在接收队列管理器中运行。 对于接收方或集群接收方通道，具有特定名称的通道的多个实例可能正在接收队列管理器中运行，但在任何时候都只能从特定远程队列管理器运行一个实例。

注: AdoptNewMCA 在请求者或服务器连接通道上不受支持。

从以下列表中指定一个或多个以逗号或空格分隔的值:

否
AdoptNewMCA 功能部件不是必需的。 这是缺省值。

SVR
采用服务器通道。

SDR
采用发送方通道。

RCVR
采用接收方通道。

CLUSRCVR
采用集群接收方通道。

全部
采用除 FASTPATH 通道以外的所有通道类型。

FASTPATH
如果该通道是 FASTPATH 通道，请采用该通道。 仅当还指定了相应的通道类型时，才会发生此情况，例如: AdoptNewMCA=RCVR,SVR,FASTPATH。

注意！: AdoptNewMCA 属性在使用 FASTPATH 通道时可能会有不可预知的行为。 为 FASTPATH 信道启用 AdoptNewMCA 属性时要格外小心。

AdoptNewMCATimeout= 60（默认） |1 - 3600
新通道实例等待旧通道实例结束的时间量 (以秒计)。 请输入 1 - 3600 范围内的值。 缺省值是 60。

AdoptNewMCACheck=QM|ADDRESS|NAME|ALL
启用 AdoptNewMCA 属性时所需的检查类型。 如果可能，请执行完全检查以保护您的通道不被关闭，不慎或恶意关闭。 至少检查通道名称是否匹配。

指定以下一个或多个值 (对于 *QM*， *NAME* 或 *ALL*，用逗号或空格分隔):

QM
检查队列管理器名称是否匹配。

请注意，队列管理器名称本身是匹配的，而不是 QMID。

ADDRESS
请检查通信源 IP 地址。 例如， TCP/IP 地址。

注: 以逗号分隔的 CONNAME 值适用于目标地址，因此与此选项无关。

在多实例队列管理器从 hosta 故障转移到 hostb 的情况下，来自该队列管理器的任何出站通道都将使用 hostb 的源 IP 地址。 如果与 "hosta 不同，则 "通过新媒体中心检查=地址无法匹配。

您可以将 SSL 或 TLS 用于相互认证，以防止攻击者中断现有正在运行的通道。 或者，将 HACMP 类型的解决方案与 IP 接管 (而不是多实例队列管理器) 配合使用，或者使用网络负载均衡器来屏蔽源 IP 地址。

名称
请检查通道名称是否匹配。

全部
检查匹配的队列管理器名称，通信地址以及匹配的通道名称。

缺省值为 AdoptNewMCACheck=NAME,ADDRESS,QM。

ChlauthEarlyAdopt=Y（默认） |N
处理连接认证和通道认证规则的顺序是确定 IBM MQ 客户机应用程序连接的安全上下文的重要因素。

注意: 如果 qm.ini 文件中不存在 **ChlauthEarlyAdopt** ，那么缺省值为 N，但是，从 IBM MQ 9.0.4 开始，将使用 **ChlauthEarlyAdopt**=Y 自动添加到 qm.ini 文件来创建所有队列管理器。

仅当在队列管理器上的连接认证 AUTHINFO 对象上设置了 ADOPTCTX (YES) 时，**ChlauthEarlyAdopt** 才会采用已提供给队列管理器以进行连接认证的用户标识。

ChlauthEarlyAdopt 的有效值为以下值:

Y
在应用通道认证规则之前，通道使用队列管理器连接认证来验证并采用应用程序提供的用户标识和密码凭证。 在此操作方式下，通道认证规则与连接认证检查所生成的用户标识相匹配。

N
通道会延迟应用程序提供的用户标识和密码凭证的连接认证验证，直到应用通道认证规则为止。 请注意，在此操作方式下，通道认证阻塞和映射规则不能考虑用户标识和密码验证的结果。

例如，缺省认证信息对象设置为 **ADOPTCTX(YES)**，用户 fred 已登录。 配置了以下两个 CHLAUTH 规则:

```
SET CHLAUTH('MY.CHLAUTH') TYPE(ADDRESSMAP) DESCR('Block all access by
default') ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
SET CHLAUTH('MY.CHLAUTH') TYPE(USERMAP) DESCR('Allow user bob and force
CONNAUTH') CLNTUSER('bob') CHCKCLNT(REQUIRED) USERSRC(CHANNEL)
```

发出以下命令，目的是将该命令作为用户 bob 的已采用安全上下文进行认证:

```
runmqsc -c -u bob QMGR
```

实际上，队列管理器使用 fred 而不是 bob 的安全上下文，并且连接失败。

要使用 bob 的安全上下文，必须将 **ChlauthEarlyAdopt** 设置为 Y。

PasswordProtection = 兼容|始终|可选|warn
从 IBM MQ 8.0 开始，在 MQCSP 结构中设置受保护密码，而不是使用 TLS。

MQCSP 密码保护对于测试和开发目的很有用，因为使用 MQCSP 密码保护比设置 TLS 加密更简单，但并不安全。

有关更多信息，请参阅 MQCSP 密码保护。

IgnoreSeqNumberMismatch=NO（默认） |YES
通道两端的消息通道代理程序 (MCA) 各自保留通过通道发送的消息数以保持同步。 可能会丢失同步，例如，如果删除了一端的通道定义，然后重新创建了该定义。 在这些情况下，可能需要 RESET CHANNEL 来确认同步数据已丢失，并允许该通道继续启动。

必须在接收方队列管理器上设置 **IgnoreSeqNumberMismatch** 属性。

实际上，此属性在接收方通道上执行重置通道命令。

此属性控制队列管理器如何使用以下值在通道启动期间处理序号不匹配的情况:

否
在通道再同步期间将检查通道序号，如果两个 MCA 不同意相同的序号，那么将报告错误消息 AMQ9526 ，并且通道将无法启动。

YES
在通道再同步期间将检查通道序号，但如果两个 MCA 不同意相同的序号，那么将报告警告消息 AMQ9703 ，并且通道启动将继续。 在正常情况下不应该需要此属性值。 当已知同步数据已丢失 (例如在灾难恢复期间) 时，此选项将避免需要手动确认每个序号不匹配。 指定此值的效果与管理员自动发出 **RESET CHANNEL** 以响应每个序号不匹配的情况类似。

V 9.2.0 **ChlauthIgnoreUserCase=N（默认） |Y**
允许队列管理器使 CHLAUTH 规则中的用户名匹配不区分大小写。 此选项允许:

- 要不区分大小写地匹配 CHLAUTH TYPE (USERMAP) 规则中的 CLNTUSER
- CHLAUTH TYPE (BLOCKUSER) 规则中的 USERLIST 将以不区分大小写的方式进行匹配

ChlauthIgnoreUserCase 的有效值为以下值:

N
通道认证规则尝试将客户机用户标识与区分大小写匹配，例如，指定 CLNTUSER ("Fred") 的规则将与 "fred" 或 "Fred" 不匹配，它将仅与用户标识 "Fred" 匹配。 这是缺省值。

Y
通道认证规则尝试将客户机用户标识与不区分大小写匹配，例如，指定 CLNTUSER ("Fred") 的 TYPE (USERMAP) 或 TYPE (USERBLOCK) 的通道认证规则将与大小写的任何变体匹配，例如用户标识 "Fred" ， "Fred" 和 "fred" 全部匹配。

请注意，在匹配通道认证规则时忽略用户标识的情况时，可能会有多个规则匹配。 如果发生此情况，那么将取消定义匹配的规则。 例如，通过以下规则，如果用户 "fred" 通过 CLIENT 通道连接到队列管理器，那么可能会将其映射到 "mquser1" 或 "mquser2":

```
SET CHLAUTH('CLIENT') TYPE(USERMAP) CLNTUSER('fred') USERSRC(MAP) MCAUSER('mquser1')
SET CHLAUTH('CLIENT') TYPE(USERMAP) CLNTUSER('FRED') USERSRC(MAP) MCAUSER('mquser2')
```

要避免在使用 ChlauthIgnoreUserCase=Y 时出现任何不确定性，请避免定义 CHLAUTH 规则，这些规则将在使用不区分大小写的匹配时重叠并导致不同的行为。

ChlauthIssueWarn = y
如果希望在 **SET CHLAUTH** 命令上设置 WARN = YES 属性时生成消息 AMQ9787 ，请设置此属性。

示例节

```
Channels:
   MaxChannels=200
   MaxActiveChannels=100
   MQIBindType=STANDARD
   PipelineLength=2
```

相关概念

第 179 页的『通道状态』

一个通道可以随时处于许多状态中的一种状态。 某些州也有子州。 从给定状态，通道可以进入其他状态。

Multi qm.ini 文件的连接节

Connection 节定义缺省绑定类型。

使用 qm.ini 文件中的 Connection 节来指定缺省绑定类型。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer Extended 队列管理器属性页面。

注: 如果需要连接节，那么必须创建连接节。

DefaultBindType= 共享（默认）|隔离

如果 **DefaultBindType** 设置为 **隔离**，那么应用程序和队列管理器将在单独的进程中运行，并且不会在它们之间共享任何资源。

如果 **DefaultBindType** 设置为 SHARED，那么应用程序和队列管理器将在不同的进程中运行，但某些资源在它们之间共享。

缺省值为 SHARED。

注意: DefaultBindType 适用于所有 MQCONN 调用以及使用带有 MQCNO_STANDARD_BINDING 的 MQCONNX 的任何调用。

更改 **DefaultBindType** 可能会导致某些应用程序性能下降。

示例节

```
Connection:
  DefaultBindType=SHARED
```

诊断消息记录

IBM MQ 的诊断消息日志是一种机制，用于允许 IBM MQ 系统的各个组件报告与 IBM MQ 配置以及运行时状态更改和问题相关的诊断消息。

这些日志有时称为 IBM MQ 错误日志，但始终包含 IBM MQ 信息和警告消息以及错误消息。 向这些日志报告的 IBM MQ 的三个主要组件是:

- 队列管理器
- IBM MQ Client
- IBM MQ 系统的其余部分

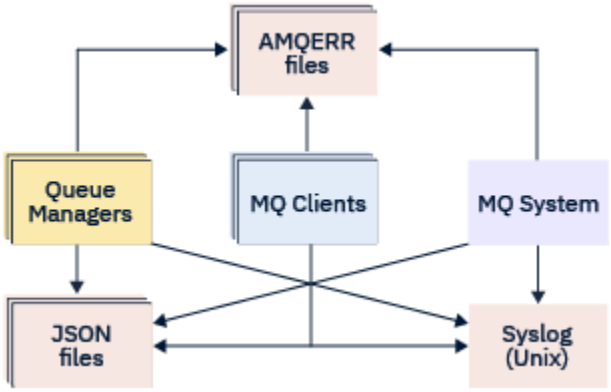
IBM MQ 支持通过多种称为 诊断消息服务的不同方法来报告诊断消息，从而允许采用定制方法来记录和使用此信息:

- AMQERRnn 日志文件
- JSON 格式的日志文件
- AIX JSON 格式的系统日志

IBM MQ 输出的 JSON 格式为单行 JSON 对象，因此 JSON 日志或 Syslog 记录的每一行都表示有效的 JSON 对象。 日志整体不封装为单个 JSON 对象。

下图显示了队列管理器， IBM MQ 客户机和 IBM MQ 系统可以使用描述的方法 所有 报告诊断消息。

图 *5: IBM MQ* 的不同部分如何报告诊断消息



如何配置 IBM MQ 诊断日志:

诊断日志是使用 qm.ini 文件中的节定义和定制的，特定于需要它们的 IBM MQ 组件。 每个唯一的日志记录端点都在其自己的节标题下在 ini 文件中定义，以及在其中定义的任何定制。 定制可以包括:

- 在发生回滚之前要回绕的日志文件的大小; 不适用于 Syslog
- 基于日志消息严重性的任何过滤，以及
- 要禁止的任何特定消息代码。

可以将 IBM MQ 配置为写入三种类型的日志记录端点中的任何一个或全部，从而允许特定日志节实现特定角色。 同样，可以定义多个文件服务。 例如:

- JSON 格式可通过本地和云环境中的自动化工具更轻松地进行解析。
- Syslog 输出允许 IBM MQ 组件根据系统上的其他产品将诊断信息集成到公共操作系统日志记录位置。
- 根据允许特定日志文件 (例如，仅记录系统中的严重错误) 的严重性过滤的日志端点。

无论配置的诊断日志记录的样式如何，除了使用的任何其他日志记录配置外，将始终写入保存在 IBM MQ 系统日志目录 (/var/mqm/errors/AMQERRnn.log) 和特定队列管理器日志目录 (/var/mqm/qmgrs/<qmgr_name>/errors/AMQERRnn.log) 下的传统诊断文件。

仅对于队列管理器，可通过指定 第 102 页的『诊断消息服务节』的属性来执行这些必需日志的可选配置。

不同的节区域

其他节可以应用于 IBM MQ 的不同区域。

队列管理器 (qm.ini)
应用于队列管理器生成的日志消息

系统 (mqs.ini)
应用于系统生成的日志消息。 此选项并非特定于队列管理器，除非队列管理器无法访问或写入其自己的日志。

模板 (mqs.ini)
一个或多个节作为模板，在创建队列管理器时复制到 qm.ini 。

客户机 (mqclient.ini)
应用于客户机操作，例如，以客户机方式将 **runmqsc** 应用于远程队列管理器。

在 JSON 格式化日志和传统格式化日志之间进行转换

已增强 mqrc 命令，允许在 JSON 和传统格式的日志之间以及不同语言之间进行大量转换。

相关参考
第 102 页的『诊断消息服务节』
可用的诊断消息服务选项支持定制 IBM MQ 诊断日志记录，以便可以将日志输出定向到 IBM MQ 的不同组件中的不同日志端点。

第 101 页的『QMErrorLog 节』
使用 qm.ini 文件中的队列管理器错误日志节 QMErrorLog 来定制 IBM MQ 错误日志的操作和内容。

第 104 页的『诊断消息服务』
可以在配置文件的 DiagnosticSystemMessages, DiagnosticMessages 和 DiagnosticMessagesTemplate 章节中定义以下诊断信息服务及其特定服务属性:

Multi *QMErrorLog 节*

使用 qm.ini 文件中的队列管理器错误日志节 QMErrorLog 来定制 IBM MQ 错误日志的操作和内容。

QMErrorLog 服务是传统 IBM MQ 诊断日志记录服务，用于输出与队列管理器相关的诊断消息。 QMErrorLog 服务连续运行，无法关闭，但可以在一定程度上进行定制。

您可以使用 qm.ini 文件中的 QMErrorLog 节来从写入队列管理器错误日志中排除某些消息。 您还可以禁止在给定时间间隔内将消息写入错误日志。

Windows Linux 或者，您可以使用 IBM MQ Explorer 中的 扩展队列管理器属性页面 来排除和禁止具有 **排除的消息**， **禁止的消息** 和 **禁止的消息时间间隔** 属性的消息，而不是直接编辑 qm.ini 文件。

注意:

- Windows 仅当您正在 Windows 平台上使用本地队列管理器时，才能使用 IBM MQ Explorer 进行更改。
- QMErrorLog 节不适用于 IBM MQ 系统配置文件 mqs.ini 或客户机配置文件 (通常名为 mqclient.ini)。

以下属性可以包含在 QMErrorLog 节中:

ErrorLog 大小 = *maxsize*
指定将其复制到备份的队列管理器错误日志的大小。 **maxsize** 必须在 32768 到 2147483648 字节的范围内。 如果未指定 **ErrorLogSize** ，那么将使用缺省值 33554432 字节 (32 MB)。

如果需要，可以使用此属性将最大大小减少回先前的最大大小 2 MB。

您可以使用 **MQMAXERRORLOGSIZE** 环境变量来设置日志大小。

ExcludeMessage= *msgIds*
指定不写入队列管理器错误日志的消息。

请参阅 第 102 页的『诊断消息服务节』 中的 ExcludeMessage 以获取更多信息。

SuppressMessage= *msgIds*
指定仅在指定的时间间隔内写入队列管理器错误日志一次的消息。 如果在 SuppressMessage 和 ExcludeMessage 中都指定了相同的消息标识，那么将排除该消息。

此选项不适用于 mqclient.ini 中定义的诊断消息服务。 有关更多信息，请参阅 第 102 页的『诊断消息服务节』 中的 SuppressMessage 。

SuppressInterval= 长度
指定仅将 SuppressMessage 中指定的消息写入队列管理器错误日志一次的时间间隔 (以秒计)。 *length* 必须在 1 到 86400 秒的范围内。 如果未指定 SuppressInterval ，那么将使用缺省值 30 秒。

示例节

```
QMErrorLog:
   ErrorLogSize=262144
   ExcludeMessage=7234
   SuppressMessage=9001,9002,9202
   SuppressInterval=30
```

相关概念
第 82 页的『队列管理器配置文件， qm.ini』
队列管理器配置文件 qm.ini 包含与特定队列管理器相关的信息。

相关参考

第 102 页的『诊断消息服务节』

可用的诊断消息服务选项支持定制 IBM MQ 诊断日志记录，以便可以将日志输出定向到 IBM MQ 的不同组件中的不同日志端点。

Multi 诊断消息服务节

可用的诊断消息服务选项支持定制 IBM MQ 诊断日志记录，以便可以将日志输出定向到 IBM MQ 的不同组件中的不同日志端点。

您可以使用具有下列其中一个名称的节来启用其他诊断消息服务:

- **DiagnosticSystemMessages**

  定义生成转至系统错误日志的诊断消息时使用的服务。 在 `mqs.ini` 或 `mqclient.ini` 文件中有效。

  客户机应用程序在 mqclient.ini 文件中使用 **DiagnosticSystemMessages** 节，在 mqs.ini 中，**DiagnosticSystemMessages** 节控制没有队列管理器上下文的服务器应用程序的消息。

  您可以配置另外将所有消息写入系统日志服务的队列管理器和应用程序。

- **DiagnosticMessages**

  定义生成转至队列管理器错误日志的诊断消息时使用的服务。 仅在 `qm.ini` 文件中有效。

- **DiagnosticMessagesTemplate**

  创建队列管理器时从 `mqs.ini` 文件复制到 `qm.ini` 文件中的 **DiagnosticMessages** 的节。

要显示诊断消息，请使用 mqrc 命令。

节的属性

注意: 服务和节的名称是必需的。

name=<stanzaname>
节的名称。 该值在 ini 文件中必须唯一。

服务 = 服务类型

此属性定义此节所启用的服务 (其中服务的名称不区分大小写)。

例如，要将 `syslog` 作为附加服务启用，请输入以下命令:

```
Service=syslog
```

请参阅 第 104 页的『诊断消息服务』 及其可用于诊断消息服务节的特定属性。

您可以将以下可选属性添加到节中:

- ExcludeMessage
- SuppressMessage
- SuppressInterval
- 第 104 页的『严重性』

ExcludeMessage= ***msgIds***
指定不写入队列管理器错误日志的消息。 如果大量使用 IBM MQ 系统，并且许多通道正在停止和启动，那么会将大量参考消息发送到 z/OS 控制台和硬拷贝日志。 IBM MQ - IMS 网桥和缓冲区管理器还可能生成大量参考消息，因此排除消息会阻止您接收大量消息 (如果需要)。 *msgIds* 包含以下消息标识的逗号分隔列表:

5211-超过最大属性名长度。
5973-已禁止分布式发布/预订
5974-已禁止分布式发布/预订发布
6254-系统无法动态装入共享库

IBM i 7163 - 作业启动的消息（仅限 IBM i）
7234 - 装入的消息数
8245-实体没有足够的权限来显示对象
9001 - 通道程序正常结束
9002 - 通道程序已启动
9202 - 远程主机不可用
9208-从主机接收时出错
9209-连接已关闭
9228-无法启动通道响应程序
9489-超过 SVRCONN 最大实例数限制
9490-超过每个客户机限制的 SVRCONN 最大实例数
9508-无法连接到队列管理器
9524 - 远程队列管理器不可用
9528 - 用户请求了关闭通道
9545-断开连接时间间隔已到期
9558-远程通道不可用
9637-通道缺少证书
9776-通道已被用户标识阻塞
9777-通道被 NOACCESS 映射阻塞
9782-连接被地址阻塞
9999 - 通道程序异常结束

SuppressMessage= *msgIds*
指定仅在指定的时间间隔内写入队列管理器错误日志一次的消息。 如果大量使用 IBM MQ 系统，并且许多通道正在停止和启动，那么会将大量参考消息发送到 z/OS 控制台和硬拷贝日志。 IBM MQ - IMS 网桥和缓冲区管理器还可能生成大量参考消息，因此禁止消息会阻止您接收大量重复消息 (如果需要)。 时间间隔由 `SuppressInterval` 指定。 *msgIds* 包含以下消息标识的逗号分隔列表:

5211-超过最大属性名长度。
5973-已禁止分布式发布/预订
5974-已禁止分布式发布/预订发布
6254-系统无法动态装入共享库
IBM i 7163 - 作业启动的消息（仅限 IBM i）
7234 - 装入的消息数
8245-实体没有足够的权限来显示对象
9001 - 通道程序正常结束
9002 - 通道程序已启动
9202 - 远程主机不可用
9208-从主机接收时出错
9209-连接已关闭
9228-无法启动通道响应程序
9489-超过 SVRCONN 最大实例数限制
9490-超过每个客户机限制的 SVRCONN 最大实例数
9508-无法连接到队列管理器
9524 - 远程队列管理器不可用
9528 - 用户请求了关闭通道
9545-断开连接时间间隔已到期
9558-远程通道不可用
9637-通道缺少证书
9776-通道已被用户标识阻塞
9777-通道被 NOACCESS 映射阻塞
9782-连接被地址阻塞

9999 - 通道程序异常结束

如果在 SuppressMessage 和 ExcludeMessage 中都指定了相同的消息标识，那么将排除该消息。

此选项不适用于 MQ client.ini 中定义的诊断消息服务。

SuppressInterval= 长度
指定仅将 **SuppressMessage** 中指定的消息写入队列管理器错误日志一次的时间间隔 (以秒计)。 *length* 必须在 1-86400 秒范围内。 如果未指定 **SuppressInterval** ，那么将使用缺省值 30 秒。

严重性
以逗号分隔的严重性级别列表，其中严重性级别的名称不区分大小写。 允许的值为:

- I (或 Information 或 0)
- W (或警告或 10)
- E (或错误或 20 和 30)
- S (或 Stop 或 40)
- T (或 System 或 50)

注意:

1. 缺省值为 all
2. 只有所选严重性级别中的消息才会提供给服务。

   或者，可以使用显示指定错误级别和所有更高级别的加号字符 (+)。 例如，要显示所有错误:

```
Severities=E+
```

相关参考
第 101 页的『QMErrorLog 节』
使用 qm.ini 文件中的队列管理器错误日志节 QMErrorLog 来定制 IBM MQ 错误日志的操作和内容。

第 104 页的『诊断消息服务』
可以在配置文件的 DiagnosticSystemMessages, DiagnosticMessages 和 DiagnosticMessagesTemplate 章节中定义以下诊断信息服务及其特定服务属性:

Multi 诊断消息服务
可以在配置文件的 DiagnosticSystemMessages, DiagnosticMessages 和 DiagnosticMessagesTemplate 章节中定义以下诊断信息服务及其特定服务属性:

定义了以下诊断消息服务:

文件
此服务以类似于 QMErrorLog 服务的方式将任何未过滤的消息发送到文件。 根据指定的 **Format**，将使用现有文本格式或指定的 JSON 格式。 缺省情况下，有三个名为 AMQERR01.LOG， AMQERR02.LOG 和 AMQERR03.LOG 或 AMQERR01.json， AMQERR02.json 和 AMQERR03.json 的文件，这取决于 **Format** 属性以及这些基于配置的大小的回滚。

以下属性仅在 "文件" 节中受支持:

FilePath
写入日志文件的路径。 缺省位置与 AMQERR01.log 文件 (即系统或队列管理器) 相同。 路径必须是绝对路径，但可以包含可替换插入。 例如:

+ MQ_Q_MGR_DATA_PATH +
队列管理器诊断消息目录的父代的完整路径。 缺省值如下所示:

- AIX 在 AIX and Linux 平台上: /var/mqm/qmgrs/<QM_name>
- Windows 在 Windows 上， C:\Program Data\IBM\MQ\qmgrs\<QM_name>

+ MQ_DATA_PATH +
系统诊断消息目录的父代的完整路径。 缺省值如下所示:

- AIX 在 AIX and Linux 平台上:`/var/mqm`
- Windows 在 Windows 上: `C:\Program Data\IBM\MQ`

如果此路径未使用现有错误目录，那么必须使用相应的许可权来创建此路径。

FilePrefix
日志文件的前缀。 缺省值为 AMQERR。

FileSize
日志回滚的大小。 缺省值为 32MB，与 第 101 页的『QMErrorLog 节』的 **ErrorLogSize** 属性一样，在语义上完全相同。

注: ErrorLogSize 属性仅适用于缺省错误日志服务，而不适用于定制诊断服务。

您可以使用 **MQMAXERRORLOGSIZE** 环境变量来设置日志大小。

Format
文件的格式。 该值可以是 *text* (对于其他 QMErrorLog 样式服务) 或 *json*(缺省值)。

根据此属性的设置，该文件的后缀为 `.LOG` 或 `.json` 。

例如，编辑队列管理器的 `qm.ini` 文件，并添加以下节:

```
DiagnosticMessages:
   Service = File
   Name = JSONLogs
   Format = json
   FilePrefix = AMQERR
```

重新启动后，队列管理器将在其 ERRORS 目录中具有 `AMQERR0x.json` 文件。

您可以定义多个文件服务。 这允许配置，如以下示例中所示，在这些示例中，不同标记的消息将拆分为不同的日志集:

```
DiagnosticMessages:
  Name=ErrorsToFile
  Service=File
  Severities=E+
  FilePrefix=OnlyErrors

DiagnosticMessages:
  Name=NonErrorstoFile
  Service=File
  Severities=I,W
  FilePrefix=InformationandWarning
```

Linux AIX **系统日志**
Syslog 服务在 Windows 或 IBM i 上不可用

只能定义一个 Syslog 服务，并且 Syslog 服务使用 JSON 格式 诊断消息规范将任何未过滤的消息发送到 syslog。 该信息将按表中显示的顺序添加到系统日志中，从 msgID 开始并插入。

通过以下方式将消息的严重性映射到系统日志级别:

表 *12:* 将消息严重性映射到系统日志级别

严重	**级别**
0	日志信息
10	LOG_WARNING
20	LOG_ERR
30	LOG_ERR
40	日志警报
50	日志警报

仅在 syslog 节中支持以下属性:

Ident
定义与系统日志条目关联的 **ident** 值。 缺省值为 *ibm-mq*。

以下示例显示了发送到 Syslog 的错误消息:

```
DiagnosticMessages:
  Name=ErrorsToSyslog
  Ident=mq
  Service=Syslog
  Severities=E+
```

请参阅 第 102 页的『诊断消息服务节』，以获取有关通用节属性的更多信息。

注意:

1. 仅对于 "文件" 服务，可以有多个节，每个节具有不同的名称。 只有使用序列中的最终名称的定义才会生效。
2. 仅当重新启动队列管理器时，对节值的更改才会生效。

Multi qm.ini 文件的 ExitPath 节

ExitPath 节指定队列管理器系统上用户出口程序的路径。

使用 qm.ini 文件中的 ExitPath 节来指定队列管理器系统上用户出口程序的路径。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer Exits 队列管理器属性页面。

ExitsDefault 路径 = *string*
ExitsDefault 路径属性指定以下位置:

- 客户机的 32 位通道出口
- 服务器的 32 位通道出口和数据转换出口
- 未限定的 XA 切换装入文件

ExitsDefaultPath64= 字符串
ExitsDefaultPath64 属性指定以下位置:

- 客户机的 64 位通道出口
- 服务器的 64 位通道出口和数据转换出口
- 未限定的 XA 切换装入文件

示例节

```
ExitPath:
   ExitsDefaultPath=/var/mqm/exits
   ExitsDefaultPath64=/var/mqm/exits64
```

Multi qm.ini 文件的 ExitPropertiesLocal 章节

ExitPropertiesLocal 节指定队列管理器上的退出属性信息。

使用 "qm.ini 文件中的 ExitPropertiesLocal 节指定队列管理器的退出属性信息。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer 集群队列管理器属性页面。

Windows 或者，在 Windows 上，可以使用 **amqmdain** 命令指定此信息。

缺省情况下，此设置继承自机器范围配置的 ExitProperties 节中的 **CLWLMode** 属性 (如 第 78 页的『mqs.ini 文件的 ExitProperties 节』 中所述)。 仅当要以其他方式配置此队列管理器时，才应更改此设置。 可以使用 "集群队列管理器属性" 页面上的集群工作负载方式属性来覆盖各个队列管理器的此值。

使用 `mqs.ini` 文件中的 ExitProperties 节来指定队列管理器出口程序使用的配置选项。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer Extended IBM MQ 属性页面。

CLWLMode= SAFE (缺省值) | FAST
集群工作负载 (CLWL) 出口允许您指定集群中要打开的集群队列以响应 MQI 调用 (例如， `MQOPEN` 和 `MQPUT`)。 CLWL 出口以 FAST 方式或 SAFE 方式运行，具体取决于您在 **CLWLMode** 属性上指定的值。 如果省略 **CLWLMode** 属性，那么集群工作负载出口将以 SAFE 方式运行。

安全
在与队列管理器不同的进程中运行 CLWL 出口。 这是缺省值。

如果在 SAFE 方式下运行时用户编写的 CLWL 出口出现问题，那么会发生以下情况:

- CLWL 服务器进程 (amqzlwa0) 失败。
- 队列管理器重新启动 CLWL 服务器进程。
- 将在错误日志中向您报告该错误。 如果正在进行 MQI 调用，那么您将以返回码的形式接收通知。

保留队列管理器的完整性。

注: 在单独的进程中运行 CLWL 出口可能会影响性能。

FAST
在队列管理器进程中内联运行集群出口。

指定此选项可通过避免与在 SAFE 方式下运行相关联的进程切换成本来提高性能，但这样做会牺牲队列管理器完整性。 仅当您确信 CLWL 出口没有问题，并且您特别关心性能时，才应该以 FAST 方式运行 CLWL 出口。

如果 CLWL 出口以 FAST 方式运行时出现问题，那么队列管理器将失败，并且您将面临队列管理器完整性受损的风险。

示例节

```
ExitPropertiesLocal:
      CLWLMode=SAFE
```

IBM i Linux AIX qm.ini 文件的文件系统节

文件系统节指定队列管理器错误日志上设置的许可权是保持不变还是更改回其缺省值。

在大多数情况下，对错误日志文件设置的缺省许可权预期很有用，因此大多数 IBM MQ 管理员都不需要更改这些许可权。

但是，您的 IBM MQ 管理员可能希望更改其错误日志文件的许可权，在这种情况下，他们应该设置 `Filesystem` 节选项 **ValidateAuth**=*No*，这将导致队列管理器在此后保持许可权不变。

缺省行为 (不带 **ValidateAuth**=*No*) 是队列管理器检查队列管理器错误日志的文件许可权，并将其更改回其缺省值。 此检查可能随时发生，包括在队列管理器结束或启动操作期间。

示例节

```
Filesystem:
      ValidateAuth=No
```

Multi qm.ini 文件的日志节

"日志" 节指定有关在队列管理器上进行日志记录的信息。

使用 qm.ini 文件中的 Log 节来指定有关在队列管理器上进行日志记录的信息。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer 日志队列管理器属性页面。

缺省情况下，这些设置继承自为队列管理器的缺省日志设置指定的设置 (如 第 79 页的『mqs.ini 文件的 LogDefaults 节』中所述)。 仅当要以其他方式配置此队列管理器时，才会更改这些设置。

有关计算日志大小的信息，请参阅 第 532 页的『计算日志大小』。

注: 以下参数列表中给出的限制由 IBM MQ 设置。 操作系统限制可能会减小可能的最大日志大小。

LogPrimaryFiles= 3（默认）|2-*254*（"Windows"）|2-*510*（"AIX and Linux"系统）
创建队列管理器时分配的日志文件。

您可以拥有的主日志文件的最小数目为 2 ，最大数目为 254 (在 Windows 上) 或 510 (在 AIX and Linux 系统上)。 缺省值为 3。

主日志文件和辅助日志文件的总数在 Windows 上不得超过 255 ，在 AIX and Linux 系统上不得超过 511 ，并且不得小于 3。

创建或启动队列管理器时检查该值。 您可以在创建队列管理器后对其进行更改。 但是，在重新启动队列管理器之前，该值的更改不会生效，并且可能不会立即生效。

LogSecondaryFiles= 2（默认）|1-*253*（"Windows"）|1-*509*（"AIX and Linux"系统）
耗尽主文件时分配的日志文件。

辅助日志文件的最小数目为 1 ，最大数目为 253 (在 Windows 上) 或 509 (在 AIX and Linux 系统上)。 缺省数字为 2。

主日志文件和辅助日志文件的总数在 Windows 上不得超过 255 ，在 AIX and Linux 系统上不得超过 511 ，并且不得小于 3。

启动队列管理器时将检查该值。 您可以更改此值，但更改直到队列管理器重新启动后才会生效，即使这样也可能不会立即生效。

LogFilePages= 编号
日志数据保存在一系列称为日志文件的文件中。 日志文件大小以 4 KB 页面为单位指定。

缺省日志文件页面数为 4096 ，给出的日志文件大小为 16 MB。

在 AIX and Linux 系统上，日志文件页面的最小数目为 64 ，而在 Windows 上，日志文件页面的最小数目为 32; 在这两种情况下，最大数目都为 65 535。

注: 无法更改队列管理器创建期间指定的日志文件的大小。

LogType= CIRCULAR（默认）|LINEAR|REPLICATED
队列管理器要使用的日志记录类型。 缺省值为 **循环**。 请参阅 第 79 页的『mqs.ini 文件的 LogDefaults 节』中 **LogType** 属性的描述，以获取有关创建具有所需日志记录类型的队列管理器的信息。

循环
使用日志启动重新启动恢复，以回滚在系统停止时正在进行的事务。

请参阅 第 527 页的『日志记录类型』，以获取更全面的循环日志记录说明。

线性
用于重新启动恢复和介质或正向恢复 (通过重放日志内容来创建丢失或损坏的数据)。

请参阅 第 527 页的『日志记录类型』，以获取更全面的线性日志记录说明。

CP4I **复制**
由本机 HA 组用于将日志数据从活动实例复制到副本实例。

请参阅 第 527 页的『日志记录类型』，以获取复制日志记录的更完整说明。

注: 无法通过在 qm.ini 文件中修改此属性来更改队列管理器的 **LogType** 。 要更改队列管理器的 **LogType** ， 必须使用 **migmqlog** 命令。

LogBufferPages= 0（默认）*|0-4096*
分配给缓冲区记录以进行写入的内存量，指定缓冲区大小 (以 4 KB 页面为单位)。

缓冲区页面的最小数目为 18 ，最大数目为 4096。 更大的缓冲区会导致更高的吞吐量，尤其是对于更大的消息。

如果指定 0 (缺省值) ，那么队列管理器将选择大小。

如果指定 1 到 17 范围内的数字，那么队列管理器缺省为 18 (72 KB)。 如果指定 18 到 4096 范围内的数字，那么队列管理器将使用指定的数字来设置分配的内存。

启动队列管理器时将检查该值。 该值可以在规定的限度内增加或减少。 但是，在下次启动队列管理器之前，该值的更改不会生效。

LogPath= 目录名
队列管理器的日志文件所在的目录。 这必须存在于队列管理器可写入的本地设备上，并且最好存在于与消息队列不同的驱动器上。 指定另一个驱动器将在发生系统故障时提供额外的保护。

默认值为:

- Windows Windows 中的 CMQ\log 。
- Linux AIX AIX and Linux 系统中的 /var/mqm/log 。

可以使用 **-ld** 标志在 **crtmqm** 命令上指定目录的名称。 创建队列管理器时，还会在队列管理器目录下创建一个目录，此目录用于保存日志文件。 此目录的名称基于队列管理器名称。 这将确保日志文件路径是唯一的，并且符合目录名称长度的任何限制。

如果未在 **crtmqm** 命令上指定 -ld ，那么将使用 **LogDefaultPath** 属性的值。

在 AIX 和 Linux 系统中，用户标识 mqm 和组 mqm 必须具有日志文件的完整权限。 如果更改这些文件的位置，那么必须自行授予这些权限。 如果日志文件位于随产品提供的缺省位置中，那么不需要执行此操作。

LogWriteIntegrity=SingleWrite|DoubleWrite|TripleWrite（默认值）
记录器用于可靠地写入日志记录的方法。

TripleWrite (缺省值)
这是缺省方法。

请注意，您可以选择 **DoubleWrite**，但如果选择此项，那么系统会将此解释为 **TripleWrite**。

SingleWrite
只有在托管 "IBM MQ 恢复日志的文件系统和设备明确保证 4KB 写入的原子性时，才应使用 **SingleWrite**。

即，当写入某个 4KB 页面因任何原因而失败时，只存在两种可能的状态：映像前或映像后。 不可能存在中间状态。

注: 如果您的持久工作负载中有足够的并行性，那么设置除缺省值 **TripleWrite** 以外的任何其他值都有最小的潜在好处。

请参阅 第 110 页的『LogWriteIntegrity -使用 SingleWrite 或 TripleWrite』 以获取更多信息。

LogManagement= 手动（默认） | 自动 | 存档
用于手动或由队列管理器管理日志扩展数据块的方法。 缺省值为 Manual。

仅当 **LogType** 为 LINEAR 时，此属性才适用。

如果您更改 **LogManagement** 值，那么在重新启动队列管理器之后，更改才会生效。

如果找到无法识别的属性值，那么直到更正该值之后，队列管理器才会启动。

LogManagement 属性在 IBM i 上无效。

手动 (缺省值)
手动管理日志扩展数据块。 指定该选项意味着队列管理器不会复用或删除日志扩展数据块，即使不再需要将这些日志扩展数据块用于恢复也是如此。

自动
日志扩展数据块由队列管理器自动管理。 指定该选项意味着当不再需要将这些日志扩展数据块用于恢复后，队列管理器即可复用或删除日志扩展数据块。 不允许归档。

归档
日志扩展数据块由队列管理器管理，但必须在每个日志扩展数据块的归档完成时通知队列管理器。

指定该选项意味着当通知队列管理器不再需要将某日志扩展数据块用于恢复并且该日志扩展数据块已归档后，队列管理器即可复用或删除日志扩展数据块。

您可以使用 **RESET QMGR** MQSC 命令或 Reset Queue Manager PCF 命令来执行此通知。

示例节

```
Log:
   LogPrimaryFiles=3
   LogSecondaryFiles=2
   LogFilePages=4096
   LogType=CIRCULAR
   LogBufferPages=0
   LogPath=/var/mqm/log/saturn!queue!manager/
```

注: LogBufferPages 的值为零表示值为 512。

LogWriteIntegrity -使用 *SingleWrite* 或 *TripleWrite*

qm.ini 文件的 Log 节中 **LogWriteIntegrity** 选项的设置确定了 IBM MQ 中的记录器用于将日志记录写出到恢复日志的算法。 缺省设置为 *TripleWrite* ，此设置在几乎所有可能的情况下都是安全的

仅当要写入部分日志页面时， **LogWriteIntegrity** 的设置才会产生任何影响。 对于具有合理并发活动量的队列管理器，这种情况很少发生。

SingleWrite

SingleWrite 选择一种算法，在非常不寻常的情况下，该算法的性能优于缺省 *TripleWrite* 设置。 *SingleWrite* 设置是安全的，仅当底层存储平台在所有情况下都可以绝对保证以原子方式写入同步写入 MQ 恢复日志的 4KB 页面。

仅当托管 IBM MQ 恢复日志的文件系统或设备显式保证 4KB 写入的原子性时，才应使用 *SingleWrite* 设置。即，当写入 4KB 页面由于任何原因而失败时，只有两种可能的状态应该是 "前映像" 或 "后映像" ，并且不应该存在中间状态。 在所有其他情况下，您应该 *TripleWrite*。

在具有足够并行性的系统上，队列管理器仅写入完整的日志数据页面，如果达到高百分比的完整页面，那么 *SingleWrite* 与 *TripleWrite* 之间没有显着的性能差异。

在具有少量并行性的系统上， *SingleWrite* 具有显着的性能优势，但是首选解决方案通常是增加并行性，而不是使用 *SingleWrite*。

请注意，很难可靠地确定 4KB 写入的原子性，而对底层软件或硬件的更改可能会使任何此类保证失效。

如果您怀疑存储基础结构现在提供了必需的保证，并且在所有情况下将来的任何时候，都应该使用 *TripleWrite*。

Windows qm.ini 文件的 LU62 节 (仅限 Windows)

LU62 节指定 SNA LU 6.2 协议配置参数。 这些参数将覆盖通道的缺省属性。

使用 qm.ini 文件中的 LU62 节来指定 SNA LU 6.2 协议配置参数。 它们将覆盖通道的缺省属性。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer LU6.2 队列管理器属性页面。

TPName
要在远程站点上启动的 TP 名称。

Library1= *DLLName 1*
APPC DLL 的名称。

缺省值为 WCPIC32。

Library2= *DLLName2*
与 Library1 相同，如果代码存储在两个单独的库中。

缺省值为 WCPIC32。

CP4I qm.ini 文件的 NativeHAInstance 节

对于容器中的 IBM MQ， NativeHAInstance 节指定本机 HA 配置中的三个节点如何相互通信。

注: 此信息仅适用于容器环境。 请参阅 使用 IBM MQ 操作程序配置本机 HA 或 创建本机 HA 组 (如果创建自己的容器)。

添加三个 NativeHAInstance 节，一个用于本机 HA 组中的每个队列管理器实例 (包括本地实例)。 添加以下属性:

名称
指定创建队列管理器实例时使用的实例名称。

ReplicationAddress
指定实例的主机名， IPv4 点分十进制或 IPv6 十六进制格式地址。 可以将地址指定为主机名， IPv4 点分十进制或 IPv6 十六进制格式地址。 复制地址必须可解析且可从组中的每个实例路由。 必须在方括号中指定用于日志复制的端口号，例如:

```
ReplicationAddress=host1.example.com(4444)
```

示例节

以下示例显示了 qm.ini 文件中用于指定本机 HA 配置的三个节点的 NativeHAInstance 节。

```
NativeHAInstance:
  Name=node-1
  ReplicationAddress=host1.example.com(4444)
NativeHAInstance:
  Name=node-2
  ReplicationAddress=host2.example.com(4444)
NativeHAInstance:
  Name=node-3
  ReplicationAddress=host3.example.com(4444)
```

相关概念
第 111 页的『qm.ini 文件的 NativeHALocalInstance 节』
对于容器中的 IBM MQ， NativeHALocalInstance 节控制本机 HA 配置的操作。

CP4I qm.ini 文件的 NativeHALocalInstance 节

对于容器中的 IBM MQ， NativeHALocalInstance 节控制本机 HA 配置的操作。

注: 此信息仅适用于容器环境。 请参阅 使用 IBM MQ 操作程序配置本机 HA 或 创建本机 HA 组 (如果创建自己的容器)。

创建本机 HA 配置时，会自动将 NativeHALocal 实例节添加到每个节点上的 qm.ini 文件中。 然后，您可以编辑 qm.ini 文件并定制 NativeHALocal 实例。

LocalName
NativeHALocal 实例节的名称，取自创建本机 HA 队列管理器时指定的日志副本实例名称。

您可以选择将以下属性添加到 NativeHALocalInstance 节:

KeyRepository
保存要用于保护日志复制流量的数字证书的密钥存储库的位置。 该位置以词干格式提供，即，它包含不带扩展名的完整路径和文件名。 如果省略了 KeyRepository 节属性，那么将以纯文本在实例之间交换日志复制数据。

CertificateLabel
用于标识用于保护日志复制流量的数字证书的证书标签。 如果提供了 KeyRepository 但省略了 CertificateLabel ，那么将使用缺省值 ibmwebspheremq*queue_manager* 。

CipherSpec
用于保护日志复制流量的 MQ CipherSpec 。 如果提供了此节属性，那么还必须提供 KeyRepository 。 如果提供了 KeyRepository 但省略了 CipherSpec ，那么将使用缺省值 ANY 。

LocalAddress
接受日志复制流量的本地网络接口地址。 如果提供了此节属性，那么它将使用格式 "[addr] [(port)]" 来标识本地网络接口和/或端口。 可以将网络地址指定为主机名， IPv4 点分十进制或 IPv6 十六进制格式。 如果省略此属性，那么队列管理器将尝试绑定到所有网络接口，它将使用 NativeHAInstances 节中与本地实例名称匹配的 ReplicationAddress 中指定的端口。

HeartbeatInterval
脉动信号间隔定义本机 HA 队列管理器的活动实例发送网络脉动信号的频率 (以毫秒计)。 脉动信号间隔值的有效范围是 500 (0.5 秒) 到 60000 (1 分钟) ，超出此范围的值将导致队列管理器无法启动。 如果省略此属性，那么将使用缺省值 5000 (5 秒)。 每个实例必须使用相同的脉动信号间隔。

HeartbeatTimeout
脉动信号超时定义本机 HA 队列管理器的副本实例在确定活动实例无响应之前等待的时间长度。 脉动信号间隔超时值的有效范围为 500 (0.5 秒) 到 120000 (2 分钟)。 脉动信号超时的值必须大于或等于脉动信号间隔。

无效值导致队列管理器无法启动。 如果省略此属性，那么副本将在启动进程以选择新的活动实例之前等待 2 x HeartbeatInterval 。 每个实例必须使用相同的脉动信号超时。

RetryInterval

重试时间间隔定义本机 HA 队列管理器应重试失败复制链接的频率 (以毫秒计)。 重试时间间隔的有效范围为 500 (0.5 秒) 到 120000 (2 分钟)。 如果省略此属性，那么副本将在重试失败的复制链接之前等待 2 x HeartbeatInterval 。

SSLFipsRequired

指定在发送日志复制流量中使用密码术时是否仅使用 FIPS 认证的算法。 设置为 Yes 或 No。

EncryptionPolicySuiteB

指定日志复制流量是否使用符合 Suite-B 的密码术以及使用的强度级别。 设置为下列其中一个值:

NONE
不使用符合 Suite-B 的密码术。 此设置为缺省设置。

128_BIT,192_BIT
将安全强度设置为 128 位和 192 位级别。

128_BIT
将安全强度设置为 128 位级别。

192_BIT
将安全强度设置为 192 位级别。

示例节

以下示例显示了 qm.ini 文件中用于指定节点局部名的 NativeHALocalInstance 节。

```
NativeHALocalInstance:
  LocalName=node-1
```

相关概念
第 111 页的『qm.ini 文件的 NativeHAInstance 节』

对于容器中的 IBM MQ， NativeHAInstance 节指定本机 HA 配置中的三个节点如何相互通信。

相关参考

runmqicred (保护 IBM MQ 客户机密码)

Windows qm.ini 文件的 NETBIOS 节 (仅限 Windows)

qm.ini 文件中的 NETBIOS 节指定 NetBIOS 协议配置参数。 这些参数将覆盖通道的缺省属性。

使用 qm.ini 文件中的 NETBIOS 节来指定 NetBIOS 协议配置参数。 它们将覆盖通道的缺省属性。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer Netbios 队列管理器属性页面。

LocalName= 名称
LAN 上用于识别此机器的名称。

AdapterNum= 0 (缺省值) | *adapter_number*
LAN 适配器的编号。 缺省值为适配器 0。

NumSess= 1 (缺省值) | *number_of_sessions*
要分配的会话数。 缺省值是 1。

NumCmds= 1 (缺省值) | *number_of_commands*
要分配的命令数。 缺省值是 1。

NumNames= 1 (缺省值) | *number_of_names*
要分配的名称数。 缺省值是 1。

Library1= *DLLName1*
NetBIOS DLL 的名称。

缺省值为 NETAPI32。

Linux AIX qm.ini 文件的 RestrictedMode 节

RestrictedMode 节指定包含允许运行 MQI 应用程序，更新所有 IPCC 资源和更改某些队列管理器目录内容的成员的组的名称。 此节仅适用于 AIX and Linux 系统。

RestrictedMode 节由 **crtmqm** 命令上的 **-g** 选项设置。 如果不使用 **-g** 选项，那么不会在 qm.ini 文件中创建该节。

在某些目录下， IBM MQ 应用程序在连接到队列管理器数据目录中的队列管理器时创建文件。 为了使应用程序能够在这些目录中创建文件，它们被授予全局写访问权:

- /var/mqm/sockets/*QMgrName*/@ipcc/ssem/*hostname*/
- /var/mqm/sockets/*QMgrName*/@app/ssem/*hostname*/
- /var/mqm/sockets/*QMgrName*/zsocketapp/*hostname*/

其中 *QMGRNAME* 是队列管理器的名称， *hostname* 是主机名。

在某些系统上，授予所有用户对这些目录的写访问权是不可接受的。 例如，不需要访问队列管理器的用户。 受限方式修改存储队列管理器数据的目录的许可权。 然后，只能由指定应用程序组的成员访问这些目录。 对用于与队列管理器通信的 System V IPC 共享内存的许可权也会以相同方式进行修改。

应用程序组是其成员有权执行以下操作的组的名称:

- 运行 MQI 应用程序
- 更新气专委的所有资源
- 更改某些队列管理器目录的内容

要对队列管理器使用受限方式，请执行以下操作:

- 队列管理器的创建者必须位于 mqm 组和应用程序组中。
- mqm 用户标识必须在应用程序组中。

- 要管理队列管理器的所有用户都必须位于 mqm 组和应用程序组中。
- 要运行 IBM MQ 应用程序的所有用户都必须在应用程序组中。

由不在应用程序组中的用户发出的任何 MQCONN 或 MQCONNX 调用都将失败，原因码为 MQRC_Q_MGR_NOT_AVAILABLE。

要点: 在许多操作系统上，为了将用户添加到要识别的组中，有问题的用户必须注销并重新登录。

受限方式使用 IBM MQ 授权服务进行操作。 因此，您还必须授予用户连接到 IBM MQ 的权限，并使用 IBM MQ 授权服务访问用户所需的资源。

ALW 可以在 在 AIX, Linux, and Windows 系统上设置安全性中找到有关配置 IBM MQ 授权服务的更多信息。

仅当授权服务提供的控制未提供队列管理器资源的足够隔离时，才使用 IBM MQ 受限方式。

相关参考

crtmqm (创建队列管理器)

Multi qm.ini 文件的安全节

安全性节指定对象权限管理器 (OAM) 的选项。

ClusterQueueAccessControl=RQMName|Xmitq
设置此属性以检查集群队列管理器上托管的集群队列或标准队列的访问控制。

RQMNAME
为远程托管队列的访问控制而检查的概要文件是命名队列或命名队列管理器概要文件。

XMITQ
针对远程托管队列的访问控制检查的概要文件将解析为 SYSTEM.CLUSTER.TRANSMIT.QUEUE。

Xmitq 是缺省值。

Windows **GroupModel=GlobalGroups**
此属性确定在确定 Windows 上用户的组成员资格时， OAM 是否检查全局组。

缺省值是不检查全局组。

GlobalGroups
OAM 检查全局组。

在设置了 GlobalGroups 的情况下，授权命令 **setmqaut**， **dspmqaut** 和 **dmpmqaut** 接受全局组名称; 请参阅 **setmqaut** -g 参数。

注: 设置 ClusterQueueAcessControl=RQMName 并将授权服务的定制实现设置为小于 MQZAS_VERSION_6 将导致队列管理器未启动。 在此实例中，设置 ClusterQueueAcessControl=Xmitq 或将定制授权服务升级到 MQZAS_VERSION_6 或更高版本。

示例节

```
Security:
      ClusterQueueAccessControl=Xmitq
      GroupModel=GlobalGroups
```

Multi qm.ini 文件的服务节

服务节用于对可安装服务进行更改。 此节包含服务的名称以及为服务定义的入口点数。

注: Windows Linux 更改可安装服务及其组件有重大影响。 因此，可安装服务在 IBM MQ Explorer 中是只读的。

对于服务中的每个组件，还必须指定包含该组件的代码的模块的名称和路径。 请使用 ServiceComponent 节来执行此操作。

Service 和 **ServiceComponent** 节可以按任意顺序出现，并且这些节下的节键也可以按任何顺序出现。 对于这两个节中的任何一个节，所有节键都必须存在。 如果某个节键重复，将使用后一个节键。

启动时，队列管理器依次处理配置文件中的每个服务组件条目。 然后装入指定的组件模块，调用组件的入口点（它必须是组件初始化的入口点），传递给它一个配置句柄。

名称=AuthorizationService（默认）|NameService 名称服务
必需服务的名称。

AuthorizationService
对于 IBM MQ，授权服务组件称为对象权限管理器或 OAM。 创建队列管理器时，将自动添加 AuthorizationService 节及其关联的 ServiceComponent 节。 手动添加其他 ServiceComponent 节。

Linux AIX 队列管理器配置文件中的以下节定义了 IBM MQ for AIX 上的两个授权服务组件。 *MQ_INSTALLATION_PATH* 表示 IBM MQ 安装所在的高级目录。

```
 Service:
    Name=AuthorizationService
    EntryPoints=13

 ServiceComponent:
    Service=AuthorizationService
    Name=MQSeries.UNIX.auth.service
Module= MQ_INSTALLATION_PATH/lib/amqzfu
    ComponentDataSize=0

 ServiceComponent:
    Service=AuthorizationService
    Name=user.defined.authorization.service
    Module=/usr/bin/udas01
    ComponentDataSize=96
```

图 *6: qm.ini* 中的 *AIX and Linux* 授权服务节

Linux AIX 服务组件节 (`MQSeries.UNIX.auth.service`) 定义缺省权限服务组件 OAM。 如果您移除此节并重新启动队列管理器，OAM 将被禁用并且不会进行权限检查。

Windows 您还可以使用 IBM MQ 服务来添加 `SecurityPolicy` 属性。 仅当 *`Service`* 节上指定的服务是授权服务 (即缺省 OAM) 时， `SecurityPolicy` 属性才适用。 `SecurityPolicy` 属性允许您为每个队列管理器指定安全策略。 可能的值为:

Default
如果您希望缺省安全策略生效，请指定 `Default`。 如果 Windows 安全标识 (NT SID) 没有传递到特定用户标识的 OAM，将尝试通过搜索相关安全数据库来获取相应的 SID。

NTSIDsRequired
要求在执行安全检查时将 NT SID 传递给 OAM。

Windows 服务组件节 `MQSeries.WindowsNT.auth.service` 定义了缺省授权服务组件 OAM。 如果您移除此节并重新启动队列管理器，OAM 将被禁用并且不会进行权限检查。

NameService
缺省情况下，未提供名称服务。 如果需要名称服务，那么必须手动添加 `NameService` 节。

Linux AIX 以下名称服务的 AIX and Linux 配置文件节示例指定了由 ABC 公司（虚拟）提供的名称服务组件。

```
# Stanza for name service
 Service:
    Name=NameService
    EntryPoints=5

# Stanza for name service component, provided by ABC
 ServiceComponent:
    Service=NameService
    Name=ABC.Name.Service
    Module=/usr/lib/abcname
    ComponentDataSize=1024
```

图 7: *qm.ini* 中的名称服务节（对于 *AIX and Linux* 系统）

EntryPoints= 条目数
为服务定义的入口点数。

其中包含初始化和终止入口点。

Windows **SecurityPolicy= Default|NTSIDsRequired**
在 Windows 系统上，仅当指定的服务是缺省授权服务 (即 OAM) 时， **SecurityPolicy** 属性才适用。 **SecurityPolicy** 属性允许您为每个队列管理器指定安全策略。

可能的值为：

缺省
使用缺省安全策略以使其生效。 如果 Windows 安全标识 (NT SID) 没有传递到特定用户标识的 OAM，将尝试通过搜索相关安全数据库来获取相应的 SID。

NTSIDsRequired
执行安全性检查时，将 NT SID 传递到 OAM。

有关更多信息，请参阅 Windows 安全标识 (SID)。

另请参阅 配置授权服务节: Windows 系统。

Linux AIX **SecurityPolicy=user|group|外部用户**
在 AIX and Linux 系统上，该值指定队列管理器是使用基于用户的授权还是基于组的授权。 值不区分大小写。

V 9.2.1 从 IBM MQ 9.2.1 开始，可以使用 `UserExternal` 的其他选项。 如果选择此选项，那么可以为系统未知的用户名定义权限记录，最多 12 个字符，即：

- 必须符合 用于命名 IBM MQ 对象的规则
- 可用于检查和设置权限

您可以修改现有队列管理器以使用此附加选项，而不会丢失任何当前配置。

但是，如果创建非操作系统用户名，那么该用户将被视为不属于除 nobody 组以外的任何组。 有关组的更多信息，请参阅 AIX, Linux, and Windows 上的主体和组。

如果不包含 **SecurityPolicy** 属性，那么将使用 `default` ，这将使用基于组的授权。

重新启动队列管理器以使更改生效。 另请参阅 配置授权服务节: Windows 系统。

SharedBindingsUserId= 用户类型
仅当指定的服务是缺省授权服务 (即 OAM) 时， **SharedBindingsUserId** 属性才适用。 **SharedBindingsUserId** 属性仅用于与共享绑定的关系。 此值允许您指定 *IdentityContext* 结构中来自 MQZ_AUTHENTICATE_USER 函数的 *UserIdentifier* 字段是有效用户标识还是实际用户标识。

有关 MQZ_AUTHENTICATE_USER 函数的信息，请参阅 MQZ_AUTHENTICATE_USER-Authenticate user。

可能的值为：

缺省
UserIdentifier 字段的值设置为实际用户标识。

实数
UserIdentifier 字段的值设置为实际用户标识。

有效
UserIdentifier 字段的值设置为有效用户标识。

FastpathBindingsUserId= 用户类型
仅当指定的服务是缺省授权服务 (即 OAM) 时， **FastpathBindingsUserId** 属性才适用。 **FastpathBindingsUserId** 属性仅与快速路径绑定的关系配合使用。 此值允许您指定 *IdentityContext* 结构中来自 MQZ_AUTHENTICATE_USER 函数的 *UserIdentifier* 字段是有效用户标识还是实际用户标识。

有关 MQZ_AUTHENTICATE_USER 函数的信息，请参阅 MQZ_AUTHENTICATE_USER-Authenticate user。

可能的值为：

缺省
UserIdentifier 字段的值设置为实际用户标识。

实数
UserIdentifier 字段的值设置为实际用户标识。

有效
UserIdentifier 字段的值设置为有效用户标识。

IsolatedBindingsUserId= 用户类型
仅当指定的服务是缺省授权服务 (即 OAM) 时， **IsolatedBindingsUserId** 属性才适用。 **IsolatedBindingsUserId** 属性仅用于与隔离绑定的关系。 此值允许您指定 *IdentityContext* 结构中来自 MQZ_AUTHENTICATE_USER 函数的 *UserIdentifier* 字段是有效用户标识还是实际用户标识。

有关 MQZ_AUTHENTICATE_USER 函数的信息，请参阅 MQZ_AUTHENTICATE_USER-Authenticate user。

可能的值为：

缺省
UserIdentifier 字段的值设置为有效用户标识。

实数
UserIdentifier 字段的值设置为实际用户标识。

有效
UserIdentifier 字段的值设置为有效用户标识。

有关可安装服务和组件的更多信息，请参阅 AIX, Linux, and Windows 的可安装服务和组件。

有关一般安全服务的更多信息，请参阅 在 AIX and Linux 系统上设置安全性。

示例节

```
Service:
   Name=AuthorizationService
   EntryPoints=14
```

相关概念
适用于 AIX， Linux 和 Windows 的可安装服务和组件
相关参考
IBM i 上的可安装服务和组件
可安装服务参考信息

Multi qm.ini 文件的 ServiceComponent 节

ServiceComponent 节指定服务组件的信息。 添加新的可安装服务时，必须指定服务组件信息。 缺省情况下，授权服务节存在，并且关联的组件 OAM 处于活动状态。

Service 和 **ServiceComponent** 节可以按任意顺序出现，并且这些节下的节键也可以按任何顺序出现。 对于这两个节中的任何一个节，所有节键都必须存在。 如果某个节键重复，将使用后一个节键。

启动时，队列管理器依次处理配置文件中的每个服务组件条目。 然后装入指定的组件模块，调用组件的入口点（它必须是组件初始化的入口点），传递给它一个配置句柄。

服务 = *service_name*
必需服务的名称。 这必须与服务配置信息的 Name 属性上指定的值匹配。

名称 = *component_name*
服务组件的描述性名称。 这必须是唯一的，并且仅包含对 IBM MQ 对象的名称 (例如，队列名称) 有效的字符。 此名称出现在由服务生成的操作员消息中。 我们建议此名称以公司商标或类似的区分字符串开头。

模块 = *module_name*
将包含这个组件代码的模块的名称。 这必须是完整路径名。

ComponentDataSize= 大小
每次调用时传递到组件的组件数据区的大小 (以字节计)。 如果不需要组件数据，请指定零。

示例节

```
ServiceComponent:
   Service=AuthorizationService
   Name=MQSeries.UNIX.auth.service
   Module=amqzfu
   ComponentDataSize=0
```

有关显示 AuthorizationService 节及其关联的 ServiceComponent 节和 NameService 节及其关联的 ServiceComponent 节的进一步示例，请参阅 第 114 页的『qm.ini 文件的服务节』。

相关概念
适用于 AIX， Linux 和 Windows 的可安装服务和组件
相关参考
第 114 页的『qm.ini 文件的服务节』
服务节用于对可安装服务进行更改。 此节包含服务的名称以及为服务定义的入口点数。
IBM i 上的可安装服务和组件
可安装服务参考信息

Windows qm.ini 文件的 SPX 节 (仅限 Windows)

SPX 节指定 SPX 协议配置参数。 这些参数将覆盖通道的缺省属性。

使用 qm.ini 文件中的 SPX 节来指定 SPX 协议配置参数。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer SPX 队列管理器属性页面。

套接字 = 5E86 (缺省值) | *socket_number*
以十六进制表示法表示的 SPX 套接字号。 缺省值为 X'5E86'。

BoardNum= 0 (缺省值) | *adapter_number*
LAN 适配器号。 缺省值为适配器 0。

KeepAlive= NO | 是
打开或关闭 KeepAlive 功能。

KeepAlive=YES 使 SPX 定期检查连接的另一端是否仍然可用。 如果不是，那么将关闭通道。

Library1= *DLLName1*
SPX DLL 的名称。

缺省值为 WSOCK32.DLL。

Library2= *DLLName2*
与 LibraryName1 相同，如果代码存储在两个单独的库中。

缺省值为 WSOCK32.DLL。

ListenerBacklog= 数字
覆盖 SPX 侦听器的缺省未完成请求数。

在 SPX 上接收时，将设置最大未完成的连接请求数。 这可视为在 SPX 套接字上等待侦听器接受请求的积压请求。 缺省侦听器储备值显示在 第 119 页的表 13 中。

表 *13:* 缺省未完成的连接请求 *(SPX)*

平台	缺省 ListenerBacklog 值
Windows 服务器	100
Windows 工作站	5

注: 某些操作系统支持比所显示的缺省值更大的值。 使用此选项可避免达到连接限制。

相反，某些操作系统可能会限制 SPX 任务列表的大小，因此有效的 SPX 任务列表可能小于此处的请求。

如果积压达到 第 119 页的表 13 中显示的值，那么将拒绝 SPX 连接，并且通道无法启动。 对于消息通道，这将导致通道进入 RETRY 状态，并在稍后重试连接。 对于客户机连接，客户机从 MQCONN 接收到 MQRC_Q_MGR_NOT_AVAILABLE 原因码，并且应该稍后重试连接。

Multi qm.ini 文件的 SSL 节

SSL 节用于在队列管理器上配置 TLS 通道。

联机证书状态协议 (OCSP)

证书可包含 AuthorityInfoAccess 扩展名。 此扩展指定要通过联机证书状态协议 (OCSP) 联系的服务器。 要允许队列管理器上的 SSL 或 TLS 通道使用 AuthorityInfoAccess 扩展，必须确保其中指定的 OCSP 服务器可用、配置正确并可通过网络访问。 有关更多信息，请参阅 使用已撤销的证书。

CrlDistributionPoint(CDP)

证书可包含 CrlDistributionPoint 扩展名。 此扩展包含一个 URL，该 URL 标识用于下载证书撤销列表 (CRL) 的协议以及要联系的服务器。

如果要允许队列管理器上的 SSL 或 TLS 通道使用 CrlDistributionPoint 扩展，请确保其中指定的 CDP 服务器可用、配置正确并可通过网络访问。

SSL 节

使用 `qm.ini` 文件中的 SSL 节来配置队列管理器上的 TLS 通道如何尝试使用以下工具，以及在使用这些工具时发生问题时如何作出反应。

在以下每种情况下，如果提供的值不是列出的有效值之一，那么将采用缺省值。 未写入任何提及指定了无效值的错误消息。

V 9.2.1 **OutboundSNI = CHANNEL | HOSTNAME**
如果 **OutboundSNI** 设置为 CHANNEL，那么在启动 TLS 连接时，支持 SNI 的客户机会将 SNI 设置为远程系统的目标 IBM MQ 通道名称。

如果此属性设置为 HOSTNAME，那么支持 SNI 的客户机会将 SNI 头设置为主机名，从而导致出站连接请求在 TLS 握手期间接收远程队列管理器的缺省证书，因此无法使用每个通道的证书。

注: 如果 **OutboundSNI**=HOSTNAME 用于连接到配置了证书标签的远程通道，那么将拒绝该连接，并在远程队列管理器错误日志中显示 MQRC_SSL_INITIALIZATION_ERROR 和 AMQ9673 消息。

AllowOutboundSNI=YES（默认） |NO

如果启用，支持 SNI 的客户机将在启动 TLS 连接时将 SNI 设置为远程系统的目标 IBM MQ 通道名称。如果此属性设置为 NO，那么支持 SNI 的客户机将不设置 SNI 头，这将导致出站连接请求在 TLS 握手期间接收远程队列管理器的缺省证书，因此无法使用每个通道的证书。

注意: V 9.2.1 从 IBM MQ 9.2.1 开始，不推荐对 Continuous Delivery 使用 **AllowOutboundSNI** 属性，此属性仅可用于向后兼容性目的。

设置为 YES 的 **AllowOutboundSNI** 提供与设置为 CHANNEL 的 **OutboundSNI** 相同的功能，而设置为 NO 的 **AllowOutboundSNI** 提供与设置为 HOSTNAME 的 **OutboundSNI** 相同的功能。

如果 SSL 节中同时存在 **AllowOutboundSNI** 和 **OutboundSNI** 属性，那么 **OutboundSNI** 的设置优先。

V 9.2.0 ***AllowedCipherSpecs=name|namelist|ALL***

指定订购并支持与 Multiplatforms 版上的 IBM MQ 通道配合使用的 CipherSpecs 的定制列表。

- 单个 CipherSpec 名称。
- 要重新启用的 IBM MQ CipherSpec 名称的逗号分隔列表。
- ALL 的特殊值，表示所有 CipherSpecs (不推荐)。

注: 不应选择 "**ALL**"CipherSpecs,，因为这样会启用 SSL "3.0 和 TLS "1.0 协议以及大量弱加密算法。

有关更多信息，请参阅 在 IBM MQ for Multiplatforms 上的 CipherSpec 顺序的 TLS 握手中提供已订购和已启用的 CipherSpecs 的定制列表。

IBM i ALW V 9.2.0 **AllowTLSV13 =Y | YES | T | TRUE| N | NO | F | FALSE**

指定队列管理器是否能够使用 TLS 1.3 CipherSpecs。

- Y (缺省值) ， YES (缺省值) ， T (缺省值) 或 TRUE (缺省值): 启用允许队列管理器使用 TLS 1.3 CipherSpecs 的 TLS 1.3 。
- N， NO， F 或 FALSE: 禁用 TLS 1.3，这意味着队列管理器无法使用 TLS 1.3 CipherSpecs。

有关更多信息，请参阅 启用 CipherSpecs。

CDPCheckExtensions=YES|NO（默认值）

指定此队列管理器上的 TLS 通道是否尝试检查在 CrlDistributionPoint 证书扩展名中命名的 CDP 服务器。

- YES: TLS 通道尝试检查 CDP 服务器以确定是否撤销数字证书。
- NO (缺省值): TLS 通道不会尝试检查 CDP 服务器。 此值是缺省值。

ALW V 9.2.0 **MinimumRSAKeySize=int**

指定在 TLS 握手期间接受 RSA 证书必须具有的最小密钥大小。 允许任何等于或大于 0 的值。 如果未指定，那么缺省为 1。

OCSPAuthentication=REQUIRED (缺省值) | WARN | OPTIONAL

指定无法从 OCSP 服务器确定撤销状态时要执行的操作。

如果启用了 OCSP 检查，那么 TLS 通道程序会尝试与 OCSP 服务器联系。

如果通道程序无法联系任何 OCSP 服务器，或者如果没有服务器可以提供证书的撤销状态，那么将使用 **OCSPAuthentication** 参数的值。

- REQUIRED (缺省值): 未能确定撤销状态会导致连接关闭，但发生错误。 此值是缺省值。
- WARN: 未能确定撤销状态会导致在队列管理器错误日志中写入警告消息，但允许连接继续。
- OPTIONAL: 未能确定撤销状态允许连接以静默方式继续。 未提供任何警告或错误。

OCSPCheckExtensions= 是（默认） | 否

指定此队列管理器上的 TLS 通道是否尝试检查在 AuthorityInfoAccess 证书扩展中命名的 OCSP 服务器。

- YES (缺省值): TLS 通道尝试检查 OCSP 服务器以确定是否撤销数字证书。 此值是缺省值。
- NO: TLS 通道不会尝试检查 OCSP 服务器。

ALW V 9.2.0 **OCSPTimeout= *number***

执行撤销检查时等待 OCSP 响应程序的秒数。

V 9.2.0.2 V 9.2.3 对于 来自 IBM MQ 9.2.0 Fix Pack 2 的 Long Term Support 和来自 IBM MQ 9.2.3 的 Continuous Delivery，如果设置了值 0 ，那么将使用缺省超时 30 秒。

如果未设置任何值，那么将使用 IBM MQ 缺省值 30 秒。

SSLHTTPProxyName= *string*

该字符串是要由 IBM Global Security Kit (GSKit) 用于 OCSP 检查的 HTTP 代理服务器的主机名或网络地址。 此地址可以后跟可选的端口号，用括号括起。 如果不指定端口号，那么将使用缺省 HTTP 端口 80。

AIX 对于 AIX 上的 32 位客户机，网络地址只能是 IPv4 地址。

在其他平台上，网络地址可以是 IPv4 或 IPv6 地址。

例如，如果防火墙阻止访问 OCSP 响应程序的 URL ，那么可能需要此属性。

V 9.2.0 ALW ***PeerCertChainValidation=string***

该字符串可以是以下两个值之一:

- `Usepeerchain` **[缺省]**: 同级提供的证书链可用于在验证证书时弥合任何信任链间隔。 根证书除外。
- `Truststoreonly` **[不推荐]**: 将仅使用信任库中的证书来验证同级的证书。

ALW V 9.2.0 **SSLHTTPConnectTimeout= 数字 */0***

执行撤销检查时等待成功建立到 HTTP 服务器的网络连接的秒数。

如果未设置任何值，那么将使用 IBM MQ 缺省值 0 (关闭)。

示例节

```
SSL:
       OutboundSNI=CHANNEL
       AllowedCipherSpecs=TLS13 CipherSpec list
       AllowTLSV13=Y
       CDPCheckExtensions=NO
       MinimumRSAKeySize=1
       OCSPAuthentication=REQUIRED
       OCSPCheckExtensions=YES
       OCSPTimeout=30
       PeerCertChainValidation=Usepeerchain
       SSLHTTPConnectTimeout=0
```

注意:

- V 9.2.1 **OutboundSNI** 的缺省值为 `Channel`。
- V 9.2.0 **TLS13 CipherSpec** 列表是特定 CipherSpecs 的列表，而不是别名密码。 如果仅需要 TLS1.3 密码，那么必须列出这些密码。 例如:

  TLS_CHACHA20_POLY1305_SHA256
  TLS_AES_256_GCM_SHA384
  TLS_AES_128_GCM_SHA256
  TLS_AES_128_CCM_SHA256

TLS_AES_128_CCM_8_SHA256

- V 9.2.0 除非您已启用弱密码，否则 **AllowTLSV13** 的缺省值为 Y，在这种情况下，会将其关闭 (除非您显式地将其打开)。
- **CDPCheckExtensions** 的值只能为 Yes 或 No。
- V 9.2.0 **PeerCertChainValidation** 的值只能是 Usepeerchain 或 Truststoreonly。

Multi qm.ini 文件的子池节

此节由 IBM MQ 创建。 请勿对其进行更改。

创建队列管理器时， IBM MQ 会自动写入该节中的子池节和属性 **ShortSubpoolName** 。 IBM MQ 选择 **ShortSubpoolName** 的值。 请勿变更此值。

该名称对应于在 /var/mqm/sockets 目录中创建的目录和符号链接， IBM MQ 用于在其正在运行的进程之间进行内部通信。

Multi qm.ini 文件的 TCP 节

TCP 节指定传输控制协议/Internet Protocol (TCP/IP) 配置参数。 这些参数将覆盖通道的缺省属性。

使用 qm.ini 文件中的 TCP 节来指定 TCP/IP 配置参数。

Windows Linux 或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer SPX TCP 队列管理器属性页面。

端口 = 1414 (缺省值) | *port_number*
TCP/IP 会话的缺省端口号 (十进制表示法)。 IBM MQ 的 熟知 端口号为 1414。

Windows **Library1= *DLLName1* (仅限 Windows)**
TCP/IP 套接字 DLL 的名称。

缺省值是 WSOCK32。

Multi V 9.2.4 **SecureCommsOnly= NO|N|FALSE|F|TRUE|T|YES|Y**
指定是否允许纯文本通信。

SecureCommsOnly=NO|N|FALSE|F
允许纯文本通信，并在队列管理器启动时输出警告消息。

SecureCommsOnly=YES|Y|TRUE|T
不允许纯文本通信，当队列管理器启动时将输出参考消息。

KeepAlive= NO（默认） |YES
打开或关闭 KeepAlive 功能。 KeepAlive=YES 会使 TCP/IP 定期检查连接的另一端是否仍然可用。 如果不是，那么将关闭通道。

ListenerBacklog=number
覆盖 TCP/IP 侦听器的缺省未完成请求数。

在 TCP/IP 上接收时，将设置最大未完成的连接请求数。 这可视为在 TCP/IP 端口上等待侦听器接受请求的积压请求。 缺省侦听器储备值显示在 第 122 页的表 14 中。

表 *14:* 缺省未完成的连接请求数 *(TCP)*

平台	缺省 ListenerBacklog 值
Windows Windows 服务器	100
Linux Linux	100
AIX AIX V5.3 或更高版本	100

注: 某些操作系统支持比所显示的缺省值更大的值。 使用此选项可避免达到连接限制。

相反，某些操作系统可能会限制 TCP 积压的大小，因此有效 TCP 积压可能小于此处请求的大小。

如果积压达到 第 122 页的表 14 中显示的值，那么将拒绝 TCP/IP 连接，并且通道无法启动。 对于消息通道，这将导致通道进入 RETRY 状态，并在稍后重试连接。 对于客户机连接，客户机从 MQCONN 接收 MQRC_Q_MGR_NOT_AVAILABLE 原因码，并在稍后重试连接。

以下属性组可用于控制 TCP/IP 所使用的缓冲区大小。 这些值将直接传递到操作系统的 TCP/IP 层。 使用这些属性时应非常小心。 如果未正确设置值，那么可能会对 TCP/IP 性能产生负面影响。 有关这将如何影响性能的更多信息，请参阅适用于您环境的 TCP/IP 文档。 值为零表示操作系统将管理缓冲区大小，而不是由 IBM MQ 固定缓冲区大小。

Connect_Timeout= 0 (缺省值) |number
尝试连接套接字超时之前的秒数。 缺省值零指定不存在连接超时。

IBM MQ 通道进程通过非阻塞套接字连接。 因此，如果套接字的另一端未就绪，那么 connect () 将立即返回 *EINPROGRESS* 或 *EWOULDBLOCK*。 在此之后，当报告通信错误时，将再次尝试连接，最多 20 次此类尝试。

如果 Connect_Timeout 设置为非零值，那么 IBM MQ 将等待指定的时间段 (通过 select () 调用) ，以便套接字准备就绪。 这将增加后续 connect () 调用成功的机会。 在由于网络上的高负载而导致连接需要一些等待周期的情况下，此选项可能是有益的。

SndBuffSize = number |0 (缺省值)
通道发送端使用的 TCP/IP 发送缓冲区的大小 (以字节计)。 可以用更具体的通道类型（如 RcvSndBuffSize）来覆盖该说明值。 如果该值设置为零，那么将使用操作系统缺省值。 如果未设置任何值，那么将使用 IBM MQ 缺省值 32768。

Multi 从 IBM MQ 8.0 开始，将使用缺省设置 0 自动创建新的队列管理器 (请参阅 第 124 页的『示例节』)。

RcvBuffSize = number |0 (缺省值)
通道接收端使用的 TCP/IP 接收缓冲区的大小 (以字节计)。 可以用更具体的通道类型（如 RcvRcvBuffSize）来覆盖该说明值。 如果该值设置为零，那么将使用操作系统缺省值。 如果未设置任何值，那么将使用 IBM MQ 缺省值 32768。

Multi 从 IBM MQ 8.0 开始，将使用缺省设置 0 自动创建新的队列管理器 (请参阅 第 124 页的『示例节』)。

RcvSndBuffSize = number |0 (缺省值)
接收方通道的发送方端使用的 TCP/IP 发送缓冲区的大小 (以字节为单位)。 如果该值设置为零，那么将使用操作系统缺省值。 如果未设置任何值，那么将使用 IBM MQ 缺省值 32768。

Multi 从 IBM MQ 8.0 开始，将使用缺省设置 0 自动创建新的队列管理器 (请参阅 第 124 页的『示例节』)。

RcvRcvBuffSize = number |0 (缺省值)
接收方通道的接收端使用的 TCP/IP 接收缓冲区的大小 (以字节计)。 如果该值设置为零，那么将使用操作系统缺省值。 如果未设置任何值，那么将使用 IBM MQ 缺省值 32768。

Multi 从 IBM MQ 8.0 开始，将使用缺省设置 0 自动创建新的队列管理器 (请参阅 第 124 页的『示例节』)。

SvrSndBuffSize = number |0 (缺省值)
客户机连接服务器连接通道的服务器端使用的 TCP/IP 发送缓冲区的大小 (以字节计)。 如果该值设置为零，那么将使用操作系统缺省值。 如果未设置任何值，那么将使用 IBM MQ 缺省值 32768。

Multi 从 IBM MQ 8.0 开始，将使用缺省设置 0 自动创建新的队列管理器 (请参阅 第 124 页的『示例节』)。

SvrRcvBuffSize = number |0 (缺省值)
客户机连接服务器连接通道的服务器端使用的 TCP/IP 接收缓冲区的大小 (以字节计)。 如果该值设置为零，那么将使用操作系统缺省值。 如果未设置任何值，那么将使用 IBM MQ 缺省值 32768。

Multi 从 IBM MQ 8.0 开始，将使用缺省设置 0 自动创建新的队列管理器 (请参阅 第 124 页的『示例节』)。

示例节

```
TCP:
   SndBuffSize=0
   RcvBuffSize=0
   RcvSndBuffSize=0
   RcvRcvBuffSize=0
   ClntSndBuffSize=0
   ClntRcvBuffSize=0
   SvrSndBuffSize=0
   SvrRcvBuffSize=0
```

注: Multi 对于 Multiplatforms 版上的新队列管理器， `qm.ini file` 的 TCP 节中的缺省 TCP 发送和接收缓冲区大小设置为由操作系统管理。 如上例中所示，将使用缺省设置 0 为发送和接收缓冲区自动创建新的队列管理器。 这仅适用于新的队列管理器。 将保留从较早版本的 IBM MQ 迁移的队列管理器的 TCP 发送和接收缓冲区设置。

如果从 `qm.ini` 文件中除去 TCP 缓冲区大小属性，那么缺省缓冲区将设置为 32K。 使用此缺省值时应谨慎，因为 32K 可能不是所有消息传递方案的相应缓冲区。

如果 TCP 发送和接收缓冲区属性设置为零，那么将使用操作系统缺省值。 选择这些缺省值的方法将因操作系统而异，但通常可以在 "tcp" 或 get/setsockopt () 操作系统手册页中找到。

Multi qm.ini 文件的 TuningParameters 节

TuningParameters 节指定用于调整队列管理器的选项。

SuppressDspAuthFail= YES |NO (缺省值)
设置为 YES 时，如果连接缺少对对象的 + dsp 权限，那么队列管理器将禁止生成授权事件以及在授权检查失败时将 AMQ8077 错误消息写入错误日志。

ImplSyncOpenOutput=值
ImplSyncOpenOutput 是在可能对同步点之外的持久放置启用隐式同步点之前，为 put 打开队列的应用程序的最小数目。 **ImplSyncOpenOutput** 的缺省值为 2。

这将产生以下效果: 如果只有一个应用程序打开了该队列以执行放置操作，那么将关闭 **ImplSyncOpenOutput** 。

指定 `ImplSyncOpenOutput=1` 表示始终考虑隐式同步点。 您可以设置任何正整数值。 如果您从不希望添加隐式同步点，请设置 `ImplSyncOpenOutput=OFF`。

V 9.2.0 **UniformCluster 名称 =集群的名称**
要用作统一集群的 IBM MQ 集群的名称。

V 9.2.0.4 V 9.2.4 **OAMLdapConnectTimeout=time|0 (缺省值)**
LDAP 客户机将等待与服务器建立 TCP 连接的最长时间 (以秒为单位)。 如果要通过连接名称列表提供多个 LDAP 服务器，那么超时将应用于每个单独的连接尝试，因此如果达到此超时，那么将尝试连接到名称列表中的下一个条目。

`time` 的最大值为 3600 秒，而值 `0`(最小值和缺省值) 表示等待不受限制。

V 9.2.0.4 V 9.2.4 **OAMLdapQueryTimeLimit=time|0 (缺省值)**
在建立连接并发送 LDAP 请求后， LDAP 客户机将等待从服务器接收对 LDAP 请求的响应的最长时间 (以秒计)。

`time` 的最大值为 3600 秒，而值 `0`(最小值和缺省值) 表示等待不受限制。

V 9.2.0.10 **OAMLdapResponseWarningTime=阈值**
如果与 LDAP 服务器的连接所用时间超过 **OAMLdapResponseWarningTime** 参数指定的阈值秒数，那么会将 AMQ5544W 消息写入错误日志。 缺省阈值为 10 秒。

ExpiryInterval
指示队列管理器扫描队列以查找其他队列活动尚未清除的到期消息的频率。 这是时间间隔 (以秒为单位)。

缺省情况下，到期扫描程序大约每 5 分钟在生产 IBM MQ 构建上运行一次。

警告: 通常不需要变更 **ExpiryInterval** 值，您应该仅在 IBM 支持人员的指导下修改此值。

示例节

V 9.2.0.10

```
TuningParameters:
       SuppressDspAuthFail=NO
       ImplSyncOpenOutput=2
       OAMLdapConnectTimeout=60
       OAMLdapQueryTimeLimit=60
       OAMLdapResponseWarningTime=10
       ExpiryInterval=300
```

相关概念
隐式同步点

Multi V 9.2.0 qm.ini 文件的 variables 节

Variables 节指定用于自动统一集群的配置变量。

在集群接收方通道的 CONNAME 和通道名称 MQSC 字段的自动集群配置期间，可以使用 "变量" 节中列出的属性。 配置变量不能在 MQSC 脚本的任何其他元素中使用。

属性=值
指定在 MQSC 定义期间用作插入的名称和关联值。

创建队列管理器时，可以使用 **crtmqm** 命令上的 **-iv** 命令行选项来提供 *attribute=value* 对。

示例节

```
Variables:
       CONNAME=127.0.0.1(1414)
```

相关概念
第 337 页的『自动应用程序均衡』
自动应用程序均衡通过启用 IBM MQ 统一集群来密切管理整个集群中的应用程序分发，而不是依赖于随机化或手动将应用程序锁定到特定队列管理器，从而大大增强了应用程序分发和可用性。

相关任务
第 346 页的『创建新的统一集群』
如何创建新的统一集群。

相关参考
第 349 页的『使用自动集群配置』
您可以通过更改 qm.ini 配置信息来配置 IBM MQ 以启用自动配置。

Multi qm.ini 文件的 XAResourceManager 节

XAResourceManager 节指定有关队列管理器协调的全局工作单元中涉及的资源管理器的信息。

使用 qm.ini 文件中的 XAResourceManager 节来指定有关队列管理器所协调的全局工作单元中涉及的资源管理器的信息。

Windows Linux或者，在 Linux (x86 和 x86-64) 和 Windows 上，使用 IBM MQ Explorer XA 资源管理器队列管理器属性页面。

为参与全局工作单元的资源管理器的每个实例手动添加 XA 资源管理器配置信息; 未提供缺省值。

有关资源管理器属性的更多信息，请参阅 数据库协调 。

名称 = *name* (必需)
此属性标识资源管理器实例。

Name 值的长度最多可以为 31 个字符。 您可以使用其 XA-switch 结构中定义的资源管理器的名称。 但是，如果要使用同一资源管理器的多个实例，那么必须为每个实例构造唯一名称。 例如，可以通过在 Name 字符串中包含数据库的名称来确保唯一性。

IBM MQ 在消息和 dspmqtrn 命令的输出中使用 Name 值。

在关联队列管理器已启动且资源管理器名称生效后，请勿更改资源管理器实例的名称，或从配置信息中删除其条目。

SwitchFile= 名称（必填）
包含资源管理器的 XA 切换结构的装入文件的标准名称。

如果要将 64 位队列管理器与 32 位应用程序配合使用，那么 name 值应仅包含包含资源管理器的 XA 切换结构的装入文件的基本名称。

32 位文件将从 ExitsDefaultPath 指定的路径加载到应用程序中。

64 位文件将从 ExitsDefaultPath64 指定的路径装入到队列管理器中。

XAOpenString= 字符串（可选）
要传递到资源管理器的 xa_open 入口点的数据字符串。 字符串的内容取决于资源管理器本身。 例如，该字符串可以标识此资源管理器实例要访问的数据库。 有关定义此属性的更多信息，请参阅:

- 为 Db2 添加资源管理器配置信息
- 为 Oracle 添加资源管理器配置信息
- 为 Sybase 添加资源管理器配置信息
- 为 Informix 添加资源管理器配置信息

并查阅资源管理器文档以获取相应的字符串。

XACloseString= 字符串（可选）
要传递到资源管理器的 xa_close 入口点的数据字符串。 字符串的内容取决于资源管理器本身。 有关定义此属性的更多信息，请参阅:

- 为 Db2 添加资源管理器配置信息
- 为 Oracle 添加资源管理器配置信息
- 为 Sybase 添加资源管理器配置信息
- 为 Informix 添加资源管理器配置信息

并查阅数据库文档以获取相应的字符串。

ThreadOfControl=THREAD|PROCESS
Windows此属性对于 Windows 是必需的。 当队列管理器需要从其自己的一个多线程进程调用资源管理器时，它将使用此值进行序列化。

线程
资源管理器完全 线程感知。 在多线程 IBM MQ 进程中，可以同时从多个线程对外部资源管理器进行 XA 函数调用。

PROCESS
资源管理器不是 线程安全。 在多线程 IBM MQ 进程中，一次只能对资源管理器进行一次 XA 函数调用。

ThreadOfControl 条目不适用于多线程应用程序进程中队列管理器发出的 XA 函数调用。 通常，在不同线程上具有并发工作单元的应用程序要求每个资源管理器都支持此操作方式。

示例节

```
XAResourceManager:
   Name=DB2 Resource Manager Bank
   SwitchFile=/usr/bin/db2swit
   XAOpenString=MQBankDB
   XACloseString=
   ThreadOfControl=THREAD
```

注: XAResourceManager 节的最大数目限制为 255。 但是，您应该只使用少量节，以避免事务性能下降。

IBM i IBM i 的示例 qm.ini 文件

显示如何在 IBM i 的队列管理器配置文件中排列属性组的示例。

```
#********************************************************************#
#* Module Name: qm.ini                                              *#
#* Type       : IBM MQ queue manager configuration file             *#
# Function    : Define the configuration of a single queue manager *#
#*                                                                  *#
#********************************************************************#
#* Notes      :                                                     *#
#* 1) This file defines the configuration of the queue manager     *#
#*                                                                  *#
#********************************************************************#
Log:
LogPath=QMSATURN.Q
LogReceiverSize=65536

CHANNELS:
MaxChannels = 20       ; Maximum number of channels allowed.
                       ; Default is 100.
MaxActiveChannels = 10 ; Maximum number of channels allowed to be
                       ; active at any time. The default is the
                       ; value of MaxChannels.

TCP:                   ; TCP/IP entries.
KeepAlive = Yes        ; Switch KeepAlive on.
SvrSndBuffSize=20000   ; Size in bytes of the TCP/IP send buffer for each
                       ; channel instance. Default is 32768.
SvrRcvBuffSize=20000   ; Size in bytes of the TCP/IP receive buffer for each
                       ; channel instance. Default is 32768.
Connect_Timeout=10000  ; Number of seconds before an attempt to connect the
                       ; channel instance times out. Default is zero (no timeout).

QMErrorLog:
ErrorLogSize = 262144
ExcludeMessage = 7234
SuppressMessage = 9001,9002,9202
SuppressInterval = 30

TuningParameters:
   ImplSyncOpenOutput=2
```

ALW 安装配置文件， mqinst.ini

在 AIX 或 Linux 上，安装配置文件 `mqinst.ini` 包含有关所有 IBM MQ 安装的信息。 在 Windows 上，安装配置信息位于注册表中。

mqinst.ini 文件的位置

Linux AIX

`mqinst.ini` 文件位于 AIX and Linux 系统上的 `/etc/opt/mqm` 目录中。 它包含有关哪个安装 (如果有) 是主安装的信息以及每个安装的以下信息:

- 安装名称
- 安装描述
- 安装标识
- 安装路径

要点: 不得直接编辑或引用 `mqinst.ini` 文件，因为其格式不固定，并且可能发生更改。

安装标识 (仅供内部使用) 是自动设置的，不得更改。

您必须使用以下命令来创建，删除，查询和修改文件中的值，而不是直接编辑 `mqinst.ini` 文件:

crtmqinst 用于创建条目。
dltmqinst 用于删除条目。
dspmqinst 以显示条目。
setmqinst 用于设置条目。

Windows 上的安装配置信息

Windows

Windows 上没有 `mqinst.ini` 文件。 安装配置信息位于注册表中，并保存在以下键中:

```
HKLM\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName
```

要点: 不能直接编辑或引用此键，因为其格式不固定，并且可能发生更改。

而是必须使用以下命令来查询和修改注册表中的值:

dspmqinst 以显示条目。
setmqinst 用于设置条目。

在 Windows 上， **crtmqinst** 和 **dltmqinst** 命令不可用。 安装和卸载过程将处理所需注册表条目的创建和删除。

Multi IBM MQ MQI client 配置文件， mqclient.ini

您可以使用文本文件中的属性来配置客户机。 这些属性可由环境变量或其他特定于平台的方式覆盖。

您可以使用类似于队列管理器配置文件 `qm.ini` 的文本文件来配置 IBM MQ MQI clients 。 该文件包含多个节，每个节都包含格式为 **attribute-name** = *value* 的若干行。

IBM MQ MQI client 配置文件通常命名为 `mqclient.ini` ，但您可以选择为其指定其他名称。 此文件中的配置信息适用于以下平台:

- ALW AIX, Linux, and Windows
- IBM i IBM i

  注: 在 IBM i 上，没有缺省 `mqclient.ini` 文件。 但是，您可以在 IBM i Integrated File System (IFS) 中创建文件。

有关更多信息，请参阅第 130 页的『客户机配置文件的位置』。

注: z/OS z/OS 平台不能用于运行 IBM MQ 客户机。 因此， IBM MQ for z/OS 上不存在 `mqclient.ini` 文件。

IBM MQ MQI client 配置文件中的属性适用于使用以下内容的客户机:

- MQI
- IBM MQ classes for Java
- IBM MQ classes for JMS
- IBM MQ classes for .NET
- XMS

虽然 IBM MQ MQI client 配置文件中的属性适用于大多数 IBM MQ 客户机，但有些属性不是由受管 .NET 和 XMS .NET 客户机读取，而是由使用 IBM MQ classes for Java 或 IBM MQ classes for JMS 的客户机读取。 有关更多信息，请参阅 第 131 页的『哪些 IBM MQ 客户机可以读取每个属性』。

配置功能适用于客户机应用程序与任何队列管理器建立的所有连接，而不是特定于与队列管理器的个别连接。 可以通过编程方式 (例如，使用 MQCD 结构或使用客户机通道定义表 (CCDT)) 来配置与单个队列管理器的连接相关的属性。

V 9.2.0 以下是客户机配置文件的示例:

```
#* Module Name: mqclient.ini                                          *#
#* Type       : IBM MQ MQI client configuration file                  *#
#  Function   : Define the configuration of a client                  *#
#*                                                                    *#
#**********************************************************************#
#* Notes      :                                                       *#
#* 1) This file defines the configuration of a client                 *#
#*                                                                    *#
#**********************************************************************#

ClientExitPath:
   ExitsDefaultPath=/var/mqm/exits
   ExitsDefaultPath64=/var/mqm/exits64

TCP:
   Library1=DLLName1
   KeepAlive = Yes
   ClntSndBuffSize=0
   ClntRcvBuffSize=0
   Connect_Timeout=0

MessageBuffer:
   MaximumSize=-1
   Updatepercentage=-1
   PurgeTime=0

LU62:
   TPName
   Library1=DLLName1
   Library2=DLLName2

PreConnect:
    Module=myMod
    Function=myFunc
    Data=ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
    Sequence=1

CHANNELS:
DefRecon=YES
      ServerConnectionParms=SALES.SVRCONN/TCP/hostname.x.com(1414)

Connection:
     ApplName=ExampleApplName
```

无法使用客户机配置文件来设置多个通道连接。

在 IBM WebSphere MQ 7.0 之前的发行版中支持的环境变量在后续发行版中继续受支持，如果此类环境变量与客户机配置文件中的等效值匹配，那么该环境变量将覆盖客户机配置文件值。

对于使用 IBM MQ classes for JMS 的客户机应用程序，您还可以通过以下方式覆盖客户机配置文件:

- 通过在 JMS 配置文件中设置属性。
- 通过设置 Java 系统属性，这也会覆盖 JMS 配置文件。

对于 .NET 客户机，您还可以使用 .NET 应用程序配置文件来覆盖客户机配置文件和等效环境变量。

配置文件中的注释

Linux AIX

可以使用分号 ";" 和散列 "#" 字符来标记配置文件中注释的开始。 这可以将整行标记为注释，或者在不会包含在设置值中的行的末尾表示注释。

如果值需要这些字符中的任何一个，那么必须使用反斜杠字符 "\" 对该字符进行转义。

以下示例显示了配置文件中注释的用法:

```
# Example of an SSL stanza with comments
SSL:
    ClientRevocationChecks=REQUIRED ; Example of an end of line comment
    SSLCryptoHardware=GSK_PKCS11=/driver\;label\;password\;SYMMETRIC_CIPHER_ON # Example of
escaped comment characters.
```

相关概念

第 131 页的『哪些 IBM MQ 客户机可以读取每个属性』

IBM MQ MQI client 配置文件中的大部分属性可由 C 客户机和非受管 .NET 客户机使用。 但是，有些属性不是由受管 .NET 和 XMS .NET 客户机读取，也不是由使用 IBM MQ classes for Java 或 IBM MQ classes for JMS 的客户机读取。

相关参考

第 130 页的『客户机配置文件的位置』

IBM MQ MQI client 配置文件可以保存在多个位置中。

Multi 客户机配置文件的位置

IBM MQ MQI client 配置文件可以保存在多个位置中。

客户机应用程序使用以下搜索路径来查找 IBM MQ MQI client 配置文件:

1. 环境变量 MQCLNTCF 指定的位置。

   此环境变量的格式为完整 URL。 这意味着文件名可能不一定是 `mqclient.ini`，并有助于将文件放在网络连接的文件系统上。

   注意:

   - C， .NET 和 XMS 客户机仅支持 `file:` 协议; 如果 URL 字符串未以 `protocol:` 开头，那么将采用 `file:` 协议
   - 要允许 Java 1.4.2 JRE (不支持读取环境变量)，可以使用 MQCLNTCF Java 系统属性覆盖 MQCLNTCF 环境变量。
2. 应用程序的当前工作目录中名为 `mqclient.ini` 的文件。
3. AIX, Linux, and Windows 系统的 IBM MQ 数据目录中名为 `mqclient.ini` 的文件。

   注意:

   - IBM MQ 数据目录在某些平台 (例如 IBM i 和 z/OS) 上不存在，或者在客户机已与其他产品一起提供的情况下不存在。

     IBM i 在 IBM i 上，没有缺省 `mqclient.ini` 文件。 但是，可以在目录 `/QIBM/UserData/mqm/`中的 IBM i Integrated File System (IFS) 中创建该文件，并定义环境变量 **MQCLNTCF** 以指向该文件。 例如:

     ```
     ADDENVVAR ENVVAR(MQCLNTCF) VALUE('QIBM/UserData/mqm/mqclient.ini') REPLACE(*YES)
     ```

     有关环境变量的更多示例，请参阅 环境变量。

     z/OS z/OS 平台不能用于运行 IBM MQ 客户机。 因此， IBM MQ for z/OS 上不存在 `mqclient.ini` 文件。
   - Linux AIX 在 AIX and Linux 系统上，目录为 `/var/mqm`
   - Windows 在 Windows 平台上，您在安装期间配置环境变量 MQ_DATA_PATH 以指向数据目录。 通常为 CMQ

     注: 如果仅安装客户机，那么环境变量可能是 MQ_FILE_PATH。
   - 要允许不支持读取环境变量的 Java 1.4.2 JRE，可以使用 MQ_DATA_PATH Java 系统属性手动覆盖 MQ_DATA_PATH 环境变量。
4. 在适合于平台且可供用户访问的标准目录中名为 `mqclient.ini` 的文件:
   - 对于所有 Java 客户机，这是 `user.home` Java 系统属性的值。

- Linux AIX 对于 AIX and Linux 平台上的 C 客户机，这是 HOME 环境变量的值。
- Windows 对于 Windows 上的 C 客户机，这是 HOMEDRIVE 和 HOMEPATH 环境变量的并置值。

Multi 哪些 IBM MQ 客户机可以读取每个属性

IBM MQ MQI client 配置文件中的大部分属性可由 C 客户机和非受管 .NET 客户机使用。 但是，有些属性不是由受管 .NET 和 XMS .NET 客户机读取，也不是由使用 IBM MQ classes for Java 或 IBM MQ classes for JMS 的客户机读取。

表 *15:* 哪些属性适用于每种类型的客户机

mqclient.ini 节名称和属性	**描述**	**C 和非受管 .NET**	**Java**	**JMS**	**受管 .NET**	**受管 XMS .NET**
CHANNELS 节						
CCSID	要使用的编码字符集编号。	Yes	否	否	Yes	Yes
ChannelDefinitionDirectory	包含客户机通道定义表的文件的目录路径。	Yes	否	否	Yes	Yes
ChannelDefinitionFile	包含客户机通道定义表的文件的名称。	Yes	否	否	Yes	Yes
ReconDelay	用于为可自动重新连接的客户机程序配置重新连接延迟的管理选项。	Yes	否	Yes	Yes	Yes
DefRecon	这是一个管理选项，用于使客户机程序能够自动重新连接，或者禁用已写入以自动重新连接的客户机程序的自动重新连接。	Yes	否	Yes	Yes	Yes
MQReconnectTimeout	重新连接到客户机的超时(以秒计)。	Yes	否	否	Yes	否
ServerConnectionParms	IBM MQ 服务器的位置以及要使用的通信方法。	Yes	否	否	Yes	Yes
Put1DefaultAlwaysSync	通过选项 MQPMO_RESPONSE_AS_Q_DEF 控制 MQPUT1 函数调用的行为。	Yes	Yes	Yes	Yes	Yes

表 15: 哪些属性适用于每种类型的客户机 (继续)

mqclient.ini 节名称和属性	描述	C 和非受管 .NET	Java	JMS	受管 .NET	受管 XMS .NET
PasswordProtection	允许您在 MQCSP 结构中设置受保护密码，而不是使用 SSL 或 TLS。	Yes	Yes	Yes	Yes	Yes
ClientExit 路径节						
ExitsDefaultPath	指定客户机的 32 位通道出口的位置。	Yes	Yes	Yes	Yes	Yes
ExitsDefaultPath64	指定客户机的 64 位通道出口的位置。	Yes	Yes	Yes	Yes	Yes
JavaExitsClassPath	运行 Java 出口时要添加到类路径的值。	否	Yes	Yes	否	否
V 9.2.0 V 9.2.0 **连接节**						
ApplName	客户机配置文件中指定的应用程序名称。	Yes	否	否	否	否
JMQI 节						
useMQCSPauthentication	控制 IBM MQ classes for Java 和 IBM MQ classes for JMS 应用程序在向队列管理器认证时是否应使用兼容性方式或 MQCSP 认证方式。	否	Yes	Yes	否	否
MessageBuffer 节						
MaximumSize	预读缓冲区的大小 (以千字节为单位)，范围为 1 到 999 999。	Yes	Yes	Yes	Yes	Yes
PurgeTime	清除保留在预读缓冲区中的消息的时间间隔 (以秒计)。	Yes	Yes	Yes	Yes	Yes

表 15: 哪些属性适用于每种类型的客户机 (继续)

mqclient.ini 节名称和属性	描述	C 和非受管 .NET	Java	JMS	受管 .NET	受管 XMS .NET
UpdatePercentage	1-100 范围内的更新百分比值，用于计算阈值以确定客户机应用程序何时向服务器发出新请求。	Yes	Yes	Yes	Yes	Yes
preconnect 节						
数据	存储连接定义的存储库的 URL。	Yes	否	否	否	否
函数	包含 PreConnect 出口代码的库中的函数入口点名称。	Yes	否	否	否	否
模块	包含 API 出口代码的模块名称。	Yes	否	否	否	否
Sequence	相对于其他出口调用此出口的顺序。	Yes	否	否	否	否
安全性节						
DisableClientAMS	对与队列管理器的客户机连接禁用或启用 AMS 。	Yes	Yes	Yes	否	否
SSL 节						
V 9.2.1 OutboundSNI	指定支持 SNI 的客户机在启动 TLS 连接时是将 SNI 设置为远程系统的目标 IBM MQ 通道名称，还是设置为主机名。	Yes	Yes	Yes	V 9.2.0.4 V 9.2.3 是	否

表 15: 哪些属性适用于每种类型的客户机 (继续)

mqclient.ini 节名称和属性	描述	C 和非受管 .NET	Java	JMS	受管 .NET	受管 XMS .NET
AllowOutboundSNI	指定在启动 TLS 连接时，支持 SNI 的客户机是否将 SNI 设置为远程系统的目标 IBM MQ 通道名称。 **注意:** V 9.2.1 从 IBM MQ 9.2.1 开始，不推荐使用此属性，请改为使用 **OutboundSNI**。	Yes	Yes	Yes	否	否
V 9.2.0 AllowTLSV13	队列管理器是否能够使用 TLS 1.3 CipherSpecs。	是 (C/C++ 客户机)	否	否	否	否
CDPCheckExtensions	指定此队列管理器上的 SSL 或 TLS 通道是否尝试检查 CrlDistributionPoint 证书扩展中指定的 CDP 服务器。	Yes	否	否	否	否
CertificateLabel	通道定义的证书标签。	Yes	否	否	否	否
CertificateVal 策略	确定所使用的证书验证类型。	Yes	否	否	否	否
ClientRevocation 检查	确定在客户机连接调用使用 SSL/TLS 通道时如何配置证书撤销检查。	Yes	否	否	否	否
EncryptionPolicySuiteB	确定通道是否使用符合 Suite-B 的密码术以及要使用的强度级别。	Yes	否	否	否	否

表 15: 哪些属性适用于每种类型的客户机 (继续)

mqclient.ini 节名称和属性	描述	C 和非受管 .NET	Java	JMS	受管 .NET	受管 XMS .NET
V 9.2.5 Environment Scope	控制 IBM MQ 是针对整个过程使用单个 IBM Global Security Kit (GSKit) 环境，还是针对每个连接使用 GSKit 环境。	是 (C 客户机)	否	否	否	否
V 9.2.0 MinimumRSAKey 大小	指定 RSA 证书必须具有才能接受的最小密钥大小。	是 (C/C++ 客户机)	否	否	否	否
OCSPAuthentication	当启用 OCSP 且 OCSP 撤销检查无法确定证书撤销状态时，定义 IBM MQ 的行为。	Yes	否	否	否	否
OCSPCheckExtensions	控制 IBM MQ 是否对 AuthorityInfo 访问证书扩展执行操作。	Yes	否	否	否	否
V 9.2.0 OCSPTimeout	执行撤销检查时等待 OCSP 响应程序的秒数。	Yes	否	否	否	否
V 9.2.0 V 9.2.0 ALW V 9.2.0 PeerCertChainValidation	GSKit 证书验证设置。	Yes	否	否	否	否
SSLCryptoHardware	设置配置系统上存在的 PKCS #11 加密硬件所需的参数字符串。	Yes	否	否	否	否
SSLFipsRequired	指定在 IBM MQ 中执行密码术时是否仅使用 FIPS 认证的算法。	Yes	否	否	否	否

表 15: 哪些属性适用于每种类型的客户机 (继续)

mqclient.ini 节名称和属性	描述	C 和非受管 .NET	Java	JMS	受管 .NET	受管 XMS .NET
SSLHTTPProxyName	该字符串是要由 GSKit 用于 OCSP 检查的 HTTP 代理服务器的主机名或网络地址。	Yes	否	否	否	否
SSLKeyRepository	保存用户数字证书的密钥存储库的位置 (以系统格式表示)。	Yes	否	否	否	否
SSLKeyReset 计数	在重新协商密钥之前，在 SSL 或 TLS 通道上发送和接收的未加密字节数。	Yes	否	否	否	否
TCP 节						
ClntRcvBuffSize	客户机连接服务器连接通道的客户机端使用的 TCP/IP 接收缓冲区的大小 (以字节计)。	Yes	Yes	Yes	Yes	Yes
ClntSndBuffSize	客户机连接服务器连接通道的客户机端所使用的 TCP/IP 发送缓冲区的大小 (以字节计)。	Yes	Yes	Yes	Yes	Yes
Connect_Timeout	尝试连接套接字超时之前的秒数。	Yes	Yes	Yes	否	否
IPAddressVersion	指定要用于通道连接的 IP 协议。	Yes	否	否	Yes	Yes
KeepAlive	打开或关闭 KeepAlive 功能。	Yes	Yes	Yes	Yes	Yes
Windows Library1	(仅在 Windows 上) TCP/IP 套接字 DLL 的名称。	Yes	否	否	否	否

Multi V 9.2.4 客户机配置文件的 Application 节

使用 Application 节来指定影响使用此配置进行连接的特定应用程序的统一集群均衡行为的属性。 此节中的值优先于 ApplicationDefaults 节，但可由通过程序提供的 MQBNO 结构覆盖。

注: 此节的每个属性的描述都指示哪些 IBM MQ 客户机可读取此属性。 要获取所有 IBM MQ MQI client 配置文件节的摘要表，请参阅每个客户机可读取的 IBM MQ 属性。

以下属性可以包含在 Application 节中:

名称 = ApplicationName
标识选项应用于的应用程序名称。

类型 = 简单，*ReqRep*，受管
向 IBM MQ 指示此应用程序参与的 IBM MQ 活动的常规模式。

BalanceTimeout = *Never,Immediate,0-999999999*，缺省值
向 IBM MQ 指示可以中断应用程序活动以允许重新平衡之前的超时; 从不，或者值最多为 999,999,999 秒，缺省值为 10 秒。

BalanceOptions = 无，*IgnTrans*
未设置任何平衡选项，或者允许当前在事务中涉及的应用程序立即中断。

Multi V 9.2.4 客户机配置文件的 ApplicationDefaults 节

使用 ApplicationDefaults 节来指定影响使用此配置进行连接的客户机应用程序的缺省统一集群均衡行为的属性。 这些缺省值可由特定于应用程序的 Application 节或通过程序提供的 MQBNO 结构覆盖。

注: 此节的每个属性的描述都指示哪些 IBM MQ 客户机可读取此属性。 要获取所有 IBM MQ MQI client 配置文件节的摘要表，请参阅每个客户机可读取的 IBM MQ 属性。

以下属性可以包含在 ApplicationDefaults 节中:

类型 = 简单，*ReqRep*，受管
向 IBM MQ 指示此应用程序参与的 IBM MQ 活动的常规模式。

BalanceTimeout = *Never,Immediate,0-999999999*，缺省值
向 IBM MQ 指示可以中断应用程序活动以允许重新平衡之前的超时; 从不，或者值最多为 999,999,999 秒，缺省值为 10 秒。

BalanceOptions = 无，*IgnTrans*
未设置任何平衡选项，或者允许当前在事务中涉及的应用程序立即中断。

Multi 客户机配置文件的 CHANNELS 节

使用 CHANNELS 节来指定有关客户机通道的信息。

注: 此节的每个属性的描述都指示哪些 IBM MQ 客户机可读取此属性。 要获取所有 IBM MQ MQI client 配置文件节的摘要表，请参阅每个客户机可读取的 IBM MQ 属性。

CHANNELS 节中可以包含以下属性:

CCSID = *number*
要使用的编码字符集编号。

此属性可以由 C，非受管 .NET，受管 .NET 和受管 XMS .NET 客户机读取。

CCSID 号等同于 MQCCSID 环境变量。

ChannelDefinitionDirectory=*path*
包含客户机通道定义表的文件的目录路径。

此属性可以由 C，非受管 .NET，受管 .NET 和受管 XMS .NET 客户机读取。

Windows 在 Windows 系统上，缺省值为 IBM MQ 数据和日志文件目录，通常为 CMQ。

Linux AIX 在 AIX and Linux 系统上，缺省值为 /var/mqm。

ChannelDefinitionDirectory 可以包含一个 URL，与 ChannelDefinitionFile 属性结合使用（见 "第 45 页的『对 CCDT 的 URL 访问权』）。

ChannelDefinitionDirectory 路径相当于环境变量 "**MQCHLLIB**。

ChannelDefinitionFile=*filename|AMQCLCHL*AMQCLCHL.TAB
包含客户机通道定义表的文件的名称。

此属性可以由 C，非受管 .NET，受管 .NET 和受管 XMS .NET 客户机读取。

客户机通道定义表等同于 **MQCHLTAB** 环境变量。

ReconDelay = (delay [, rand]) (delay [, rand]) ...
ReconDelay 属性提供了一个管理选项，用于为可自动重新连接的客户机程序配置重新连接延迟。

此属性可以由 C，非受管 .NET， IBM MQ classes for JMS，受管 .NET 和受管 XMS .NET 客户机读取。

以下是示例配置:

```
ReconDelay=(1000,200)(2000,200)(4000,1000)
```

所显示的示例定义了 1 秒的初始延迟，以及最多 200 毫秒的随机时间间隔。 下一个延迟是 2 秒加上最多 200 毫秒的随机时间间隔。 所有后续延迟为 4 秒，外加最多 1000 毫秒的随机时间间隔。

DefRecon = NO|YES|QMGR |DISABLED
DefRecon 属性提供了一个管理选项，用于使客户机程序能够自动重新连接，或禁用已写入以自动重新连接的客户机程序的自动重新连接。 如果程序使用与重新连接不兼容的选项 (例如 MQPMO_LOGICAL_ORDER)，那么可以选择设置后者。

此属性可由 C，非受管 .NET， IBM MQ classes for JMS，受管 .NET 和受管 XMS .NET 客户机读取。

IBM MQ classes for Java 不支持自动客户机重新连接。

DefRecon 选项的解释取决于是否还在客户机程序中设置了 MQCNO_RECONNECT_* 值以及设置了哪些值。

如果客户机程序使用 MQCONN 进行连接，或使用 MQCONNX 设置 MQCNO_RECONNECT_AS_DEF 选项，那么 DefRecon 设置的重新连接值将生效。 如果未在程序中设置重新连接值，或者通过 DefRecon 选项，那么不会自动重新连接客户机程序。

NO
除非被 **MQCONNX** 覆盖，否则不会自动重新连接客户机。

YES
除非被 **MQCONNX** 覆盖，否则客户机将自动重新连接。

QMGR
除非被 **MQCONNX** 覆盖，否则客户机将自动重新连接，但仅重新连接到同一队列管理器。 QMGR 选项具有与 MQCNO_RECONNECT_Q_MGR 相同的效果。

DISABLED
即使客户机程序使用 **MQCONNX** MQI 调用请求，也会禁用重新连接。

自动客户机重新连接取决于两个值:

- 应用程序中设置的重新连接选项
- mqclient.ini 文件中的 DefRecon 值

表 *16:* 自动重新连接取决于应用程序和 *mqclient.ini* 文件中设置的值

mqclient.ini 中的 DefRecon 值	**应用程序中设置的重新连接选项**			
	MQCNO_RECONNECT	MQCNO_RECONNECT_Q_MGR	MQCNO_RECONNECT_AS_DEF	MQCNO_RECONNECT_DISABLED
NO	YES	QMGR	否	否

表 16: 自动重新连接取决于应用程序和 *mqclient.ini* 文件中设置的值 (继续)				
mqclient.ini 中的 DefRecon 值	**应用程序中设置的重新连接选项**			
YES	YES	QMGR	YES	否
QMGR	YES	QMGR	QMGR	否
DISABLED	否	否	否	否

MQReconnectTimeout

客户机中的自动客户机重新连接功能尝试重新建立连接的最大时间 (以秒计)。 缺省值是 1800 秒（30 分钟）。

此属性可以由 C 客户机和非受管 .NET 客户机以及受管 .NET 客户机读取。

IBM MQ classes for JMS 客户机可以使用连接工厂属性 CLIENTRECONNECTTIMEOUT 指定要重新连接的超时。 该属性的缺省值是 1800 秒（30 分钟）。

IBM MQ classes for XMS .NET 客户机可以使用以下属性指定用于重新连接的超时:

- 连接工厂属性 CLIENTRECONNECTTIMEOUT。 该属性的缺省值是 1800 秒（30 分钟）。 此属性仅对受管方式有效。
- 属性 XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT。 该属性的缺省值是 1800 秒（30 分钟）。 此属性仅对受管方式有效。

ServerConnectionParms

ServerConnectionParms 相当于 MQSERVER 环境变量，用于指定 "IBM MQ 服务器的位置和使用的通信方法。

此属性可以由 C，非受管 .NET，受管 .NET 和受管 XMS .NET 客户机读取。

ServerConnectionParms 属性只能定义简单通道，不能用来定义 TLS 通道或带有通道出口的通道。 它是一个格式为 "频道名称/"运输类型"/连接名称的字符串，"*ConnectionName* 必须是一个完全合格的网络名称。 *ChannelName* 不能包含正斜线 (/) 字符，因为该字符用于分隔通道名、传输类型和连接名。

当使用 ServerConnectionParms 定义客户端通道时，最大报文长度为 100 MB。 因此，对通道有效的最大消息大小是在服务器上的 SVRCONN 通道中指定的值。

请注意，只能建立单个客户机通道连接。 例如，如果有两个条目:

```
ServerConnectionParms=R1.SVRCONN/TCP/localhost(1963)
ServerConnectionParms=R2.SVRCONN/TCP/localhost(1863)
```

仅使用第二个。

将 *ConnectionName* 指定为所声明传输类型的名称的逗号分隔列表。 通常，只需要一个名称。 您可以提供多个 主机名 以配置具有相同属性的多个连接。 将按照在连接列表中指定这些连接的顺序来尝试这些连接，直到成功建立连接为止。 如果没有成功的连接，那么客户机将再次开始处理。 连接列表是队列管理器组的替代方法，用于为可重新连接的客户机配置连接。

Put1DefaultAlwaysSync = NO (缺省值) | YES

通过选项 MQPMO_RESPONSE_AS_Q_DEF 控制 MQPUT1 函数调用的行为。

此属性可以由 C，非受管 .NET， IBM MQ classes for Java， IBM MQ classes for JMS，受管 .NET 和受管 XMS .NET 客户机读取。

否

如果 MQPUT1 与 MQPMO_SYNCPOINT 一起设置，那么其行为方式为 MQPMO_ASYNC_RESPONSE。 同样，如果 MQPUT1 与 MQPMO_NO_SYNCPOINT 一起设置，那么其行为方式为 MQPMO_SYNC_RESPONSE。 这是缺省值。

YES
MQPUT1 的行为就像设置了 MQPMO_SYNC_RESPONSE 一样，而不考虑是设置了 MQPMO_SYNCPOINT 还是 MQPMO_NO_SYNCPOINT 。

PasswordProtection = 兼容|始终|可选
从 IBM MQ 8.0 开始，允许您在 MQCSP 结构中设置受保护密码，而不是使用 TLS。

此属性可以由 C，非受管 .NET， IBM MQ classes for Java， IBM MQ classes for JMS，受管 .NET 和受管 XMS .NET 客户机读取。

MQCSP 密码保护对于测试和开发目的很有用，因为使用 MQCSP 密码保护比设置 TLS 加密更简单，但并不安全。

请参阅 MQCSP 密码保护 以获取更多信息。

相关任务
将 IBM MQ MQI 应用程序连接到队列管理器

Multi 客户机配置文件的 ClientExit 路径节

使用 ClientExitPath 节来指定客户机上通道出口的缺省位置。

注: 此节的每个属性的描述都指示哪些 IBM MQ 客户机可读取此属性。 要获取所有 IBM MQ MQI client 配置文件节的摘要表，请参阅每个客户机可读取的 IBM MQ 属性。

以下属性可以包含在 ClientExitPath 节中:

ExitsDefault 路径 = *string*
指定客户机的 32 位通道出口的位置。

此属性可由 C，非受管 .NET，受管 .NET，受管 XMS .NET， IBM MQ classes for Java 和 IBM MQ classes for JMS 客户机读取。 IBM MQ classes for Java 和 IBM MQ classes for JMS 客户机使用此属性来查找未在 Java 中写入的 32 位通道出口。

ExitsDefaultPath64 = 字符串
指定客户机的 64 位通道出口的位置。

此属性可由 C，非受管 .NET，受管 .NET，受管 XMS .NET， IBM MQ classes for Java 和 IBM MQ classes for JMS 客户机读取。 IBM MQ classes for Java 和 IBM MQ classes for JMS 客户机使用此属性来查找未在 Java 中写入的 64 位通道出口。

JavaExitsClassPath = 字符串
运行 Java 出口时要添加到类路径的值。 这将被任何其他语言的出口忽略。

此属性可由 IBM MQ classes for Java 和 IBM MQ classes for JMS 客户机读取。

在 JMS 配置文件中，将为 JavaExitsClass 路径名提供标准 com.ibm.mq.cfg。 前缀和此全名也用于 IBM WebSphere MQ 7.0 或更高版本的系统属性。 在 IBM WebSphere MQ 6.0 处，此属性是使用系统属性 com.ibm.mq.exitClasspath 指定的，该属性记录在 IBM WebSphere MQ 6.0 自述文件中。 不推荐使用 com.ibm.mq.exitClasspath 。 如果同时存在 JavaExitsClassPath 和 exitClasspath ，那么将采用 JavaExitsClassPath 。 如果仅存在 exitClasspath 用法，那么在 IBM WebSphere MQ 7.0 或更高版本中仍采用此用法。

Multi V 9.2.0 客户机配置文件的连接节

使用连接节来指定应用程序名称。

注: 此节的每个属性的描述都指示哪些 IBM MQ 客户机可读取此属性。 要获取所有 IBM MQ MQI client 配置文件节的摘要表，请参阅每个客户机可读取的 IBM MQ 属性。

以下属性可以包含在 Connection 节中:

ApplName = ExampleApplname
可以在客户机配置文件中指定应用程序名称。

此属性可供 C 和非受管 .NET 客户机使用。

Multi 客户机配置文件的 JMQI 节

使用 JMQI 节为 IBM MQ classes for Java 和 IBM MQ classes for JMS 使用的 Java 消息队列接口 (JMQI) 指定配置参数。

注: 此节的每个属性的描述都指示哪些 IBM MQ 客户机可读取此属性。 要获取所有 IBM MQ MQI client 配置文件节的摘要表，请参阅每个客户机可读取的 IBM MQ 属性。

以下属性可以包含在 JMQI 节中:

useMQCSPauthentication = NO|是
控制 IBM MQ classes for Java 和 IBM MQ classes for JMS 应用程序在向队列管理器认证时是否应使用兼容性方式或 MQCSP 认证方式。

此属性可由 IBM MQ classes for Java 和 IBM MQ classes for JMS 客户机读取。

此属性可以具有以下值:

否
向队列管理器认证时使用兼容性方式。 这是 IBM MQ 9.2.1 之前的版本中的缺省值。

YES
向队列管理器认证时使用 MQCSP 认证方式。 V 9.2.1 这是 IBM MQ 9.2.1 中的缺省值。

还有其他几种方法可用于设置优先于 **useMQCSPauthentication** 属性值的认证方式。 有关兼容性方式和 MQCSP 认证方式的更多信息，请参阅 使用 Java 客户机进行连接认证。

Windows 客户机配置文件的 LU62， NETBIOS 和 SPX 节

仅在 Windows 系统上，使用这些节来指定指定网络协议的配置参数。

LU62 节

使用 LU62 节来指定 SNA LU 6.2 协议配置参数。 以下属性可以包含在此节中:

Library1 = *DLLName*|WCPIC32
APPC DLL 的名称。

Library2 = *DLLName*|WCPIC32
与 Library1 相同，如果代码存储在两个单独的库中。

TPName
要在远程站点上启动的 TP 名称。

NETBIOS 节

使用 NETBIOS 节来指定 NetBIOS 协议配置参数。 以下属性可以包含在此节中:

AdapterNum = *number*|0
LAN 适配器的编号。

Library1 = *DLLName*|NETAPI32
NetBIOS DLL 的名称。

LocalName = 名称
LAN 上用于识别此计算机的名称。

这等同于 MQNAME 环境变量。

NumCmds = *number*|1
要分配的命令数。

NumSess = 数字|1
要分配的会话数。

SPX 节

使用 SPX 节来指定 SPX 协议配置参数。 以下属性可以包含在此节中:

BoardNum = *number*|0
LAN 适配器号。

KeepAlive = YES|NO
打开或关闭 KeepAlive 功能。

KeepAlive = YES 使 SPX 定期检查连接的另一端是否仍然可用。 如果不是，那么将关闭通道。

Library1 = *DLLName*|WSOCK32。DLL
SPX DLL 的名称。

Library2 = *DLLName*|WSOCK32。DLL
与 Library1 相同，如果代码存储在两个单独的库中。

套接字 = *number*|5E86
以十六进制表示法表示的 SPX 套接字号。

Multi 客户机配置文件的 MessageBuffer 节

使用 MessageBuffer 节来指定有关消息缓冲区的信息。

注: 此节的每个属性的描述都指示哪些 IBM MQ 客户机可读取此属性。 要获取所有 IBM MQ MQI client 配置文件节的摘要表，请参阅每个客户机可读取的 IBM MQ 属性。

以下属性可以包含在 MessageBuffer 节中:

MaximumSize = *integer*|1
预读缓冲区的大小 (以千字节为单位) ，范围为 1 到 999 999。

此属性可以由 C ，非受管 .NET， IBM MQ classes for Java， IBM MQ classes for JMS，受管 .NET 和受管 XMS .NET 客户机读取。

存在以下特殊值:

-1
客户机确定相应的值。

0
已针对客户机禁用预读。

PurgeTime = *integer*|600
清除保留在预读缓冲区中的消息的时间间隔 (以秒计)。

此属性可以由 C ，非受管 .NET， IBM MQ classes for Java， IBM MQ classes for JMS，受管 .NET 和受管 XMS .NET 客户机读取。

如果客户机应用程序根据 MsgId 或 CorrelId 选择消息，那么预读缓冲区可能包含发送到具有先前请求的 MsgId 或 CorrelId 的客户机的消息。 然后，这些消息将滞留在预读缓冲区中，直到使用相应的 MsgId 或 CorrelId 发出 MQGET 为止。 您可以通过设置 PurgeTime 从预读缓冲区中清除消息。 将自动清除在预读缓冲区中保留时间超过清除时间间隔的任何消息。 这些消息已从队列管理器上的队列中除去，因此除非正在浏览这些消息，否则它们将丢失。

有效范围在范围 1 到 999 999 秒之间，或者特殊值 0 ，表示不执行清除。

UpdatePercentage = 整数|-1
1-100 范围内的更新百分比值，用于计算阈值以确定客户机应用程序何时向服务器发出新请求。 特殊值-1 表示由客户端决定适当的值。

此属性可以由 C ，非受管 .NET， IBM MQ classes for Java， IBM MQ classes for JMS，受管 .NET 和受管 XMS .NET 客户机读取。

客户机定期向服务器发送请求，指示客户机应用程序已使用的数据量。 当客户机通过 MQGET 调用检索的字节数 *n* 超过阈值 *T* 时，将发送请求。 每次向服务器发送新请求时， *n* 都会重置为零。

阈值 T 计算如下:

```
T = Upper - Lower
```

上限与 *MaximumSize* 属性指定的预读缓冲区大小 (以千字节为单位) 相同。 其缺省值为 100 Kb。

下限低于上限，由 *UpdatePercentage* 属性指定。 此属性是 1 到 100 范围内的数字，缺省值为 20。 下限计算如下:

```
Lower = Upper x UpdatePercentage / 100
```

示例 1:

MaximumSize 和 UpdatePercentage 属性采用其缺省值 100 Kb 和 20 Kb。

客户机调用 MQGET 以检索消息，并重复执行此操作。 这将一直持续到 MQGET 已使用 n 个字节。

使用计算

```
T = Upper - Lower
```

T 为 (100-20) = 80 Kb。

因此，当 MQGET 调用从队列中除去 80 Kb 时，客户机会自动发出新请求。

示例 2:

MaximumSize 属性采用其缺省值 100 Kb ，并且为 UpdatePercentage 选择了值 40。

客户机调用 MQGET 以检索消息，并重复执行此操作。 这将一直持续到 MQGET 已使用 n 个字节。

使用计算

```
T = Upper - Lower
```

T 为 (100-40) = 60 Kb

因此，当 MQGET 调用从队列中除去 60 Kb 时，客户机会自动发出新请求。 这比使用缺省值的 EXAMPLE 1 更快。

因此，选择较大的阈值 *T* 会降低将请求从客户机发送到服务器的频率。 相反，选择较小的阈值 *T* 会增加从客户机发送到服务器的请求的频率。

但是，选择大阈值 *T* 可能意味着预读的性能增益会随着预读缓冲区变为空的可能性增加而降低。 发生这种情况时， MQGET 调用可能必须暂停，等待数据从服务器到达。

Multi 客户机配置文件的 PreConnect 节

使用 PreConnect 节在 `mqclient.ini` 文件中配置 PreConnect 出口。

注: 此节的每个属性的描述都指示哪些 IBM MQ 客户机可读取此属性。 要获取所有 IBM MQ MQI client 配置文件节的摘要表，请参阅每个客户机可读取的 IBM MQ 属性。

可以将以下属性包含在 PreConnect 节中:

数据 = *user_data*

此属性指定传递到预连接出口的用户数据。 传递到预连接出口的数据特定于您正在使用的预连接出口的实现以及期望传递的数据。

此属性可由 C 和非受管 .NET 客户机读取。

例如，此属性可用于指定存储连接定义的存储库的 URL ，例如，使用 LDAP 服务器时:

```
Data = ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
```

函数 = *myFunc*

包含 PreConnect 出口代码的库中的函数入口点名称。

此属性可由 C 和非受管 .NET 客户机读取。

函数定义遵循 PreConnect 出口原型 MQ_PRECONNECT_EXIT。

此字段的最大长度是 MQ_EXIT_NAME_LENGTH。

模块 = *myMod*

包含 API 出口代码的模块名称。

此属性可由 C 和非受管 .NET 客户机读取。

如果此字段包含模块的完整路径名，那么将照原样使用该名称。

序列 = *sequence_number*
相对于其他出口调用此出口的顺序。 先调用序号较低的出口，然后调用序号较高的出口。 出口的序列编号不需要连续；1、2、3 序列与 7、42、1096 序列的结果相同。 此属性是无符号的数字值。

此属性可由 C 和非受管 .NET 客户机读取。

可以在 `mqclient.ini` 文件中定义多个 PreConnect 节。 每个出口的处理顺序由节的 Sequence 属性确定。

相关任务
使用存储库的预连接出口引用连接定义

Multi 客户机配置文件的安全性节

使用 "安全性" 节来禁用或启用 AMS 以用于与队列管理器的客户机连接。

注: 此节的每个属性的描述都指示哪些 IBM MQ 客户机可读取此属性。 要获取所有 IBM MQ MQI client 配置文件节的摘要表，请参阅每个客户机可读取的 IBM MQ 属性。

以下属性可以包含在 Security 节中:

DisableClientAMS = NO|是
DisableClientAMS 属性允许您在使用 IBM WebSphere MQ 7.5 或更高版本的客户机从较低版本的产品连接到队列管理器时禁用 IBM MQ Advanced Message Security (AMS)，并报告 2085 (`MQRC_UNKNOWN_OBJECT_NAME`) 错误。

从 IBM WebSphere MQ 7.5 开始，将在 IBM MQ 客户机中自动启用 IBM MQ Advanced Message Security (AMS)，因此缺省情况下，客户机尝试检查队列管理器中对象的安全策略。 但是，较低版本的产品 (例如 IBM WebSphere MQ 7.1) 上的服务器未启用 AMS，这将导致报告 2085 (`MQRC_UNKNOWN_OBJECT_NAME`) 错误。

以下示例显示如何使用 DisableClientAMS 属性:

- 要禁用 AMS:

```
Security:
DisableClientAMS=Yes
```

- 要启用 AMS，请执行以下操作:

```
Security:
DisableClientAMS=No
```

此属性可由 C， IBM MQ classes for Java 和 IBM MQ classes for JMS 客户机读取。

相关参考
在客户机上禁用 Advanced Message Security

Multi 客户机配置文件的 SSL 节

使用 SSL 节来指定有关使用 TLS 的信息。

注: 此节的每个属性的描述都指示哪些 IBM MQ 客户机可读取此属性。 要获取所有 IBM MQ MQI client 配置文件节的摘要表，请参阅每个客户机可读取的 IBM MQ 属性。

以下属性可以包含在 SSL 节中:

AllowOutboundSNI = YES （默认） |NO
如果启用，支持 SNI 的客户机将在启动 TLS 连接时将 SNI 设置为远程系统的目标 IBM MQ 通道名称。 如果此属性设置为 NO，那么支持 SNI 的客户机将不设置 SNI 头，这将导致出站连接请求在 TLS 握手期间接收远程队列管理器的缺省证书，因此无法使用每个通道的证书。

此属性可以由 C， 非受管 .NET， IBM MQ classes for Java 和 IBM MQ classes for JMS 客户机读取。

属性值由 Java/JMS 客户机以区分大小写的方式进行解释，因此应该将 YES/NO 值设置为大写。

注意: V 9.2.1 从 IBM MQ 9.2.1 开始，不推荐使用 **AllowOutboundSNI** 属性，此属性仅可用于向后兼容性目的。

设置为 YES 的 **AllowOutboundSNI** 提供与设置为 CHANNEL 的 **OutboundSNI** 相同的功能，而设置为 NO 的 **AllowOutboundSNI** 提供与设置为 HOSTNAME 的 **OutboundSNI** 相同的功能。

如果 SSL 节中同时存在 **AllowOutboundSNI** 和 **OutboundSNI** 属性，那么 **OutboundSNI** 的设置优先。

IBM i ALW V 9.2.0 **AllowTLSV13 =Y | YES | T | TRUE| N | NO | F | FALSE**

指定队列管理器是否能够使用 TLS 1.3 CipherSpecs (请参阅 启用 CipherSpecs)。

此属性可由 C/C++ 客户机读取。

此属性具有以下可能的值:

- Y (缺省值) ， YES (缺省值) ， T (缺省值) 或 TRUE (缺省值): 启用允许队列管理器使用 TLS 1.3 CipherSpecs 的 TLS 1.3 。
- N， NO， F 或 FALSE: 禁用 TLS 1.3，这意味着队列管理器无法使用 TLS 1.3 CipherSpecs。

注: 使用 MQI 客户机时，将推断 **AllowTLSV13** 的值，除非在应用程序正在使用的 `mqclient.ini` 文件的 SSL 节中显式指定该值。 有关更多信息，请参阅 IBM MQ MQI 客户机和 TLS 1.3。

CDPCheckExtensions = YES|NO (缺省值)

CDPCheckExtensions 指定队列管理器上的 TLS 通道是否尝试检查在 CrlDistributionPoint 证书扩展名中命名的 CDP 服务器。

此属性可由 C 和非受管 .NET 客户机读取。

此属性具有以下可能的值:

- **是**： TLS 信道尝试检查 CDP 服务器，以确定数字证书是否已被撤销。
- NO: TLS 通道不会尝试检查 CDP 服务器。 此值是缺省值。

CertificateLabel = 字符串

通道定义的证书标签。

此属性可由 C 和非受管 .NET 客户机读取。

请参阅 证书标签 (CERTLABL) 以获取更多信息。

CertificateValPolicy = 字符串

确定所使用的证书验证类型。

此属性可由 C 和非受管 .NET 客户机读取。

此属性具有以下可能的值:

ANY
使用底层安全套接字库支持的任何证书验证策略。 此设置为缺省设置。

RFC5280
仅使用符合 RFC 5280 标准的证书验证。

ClientRevocationChecks = 要求|可选|禁用

确定在客户机连接调用使用 TLS 通道时如何配置证书撤销检查。 另请参阅 **OCSPAuthentication**。

此属性可由 C 和非受管 .NET 客户机读取。

此属性具有以下可能的值:

REQUIRED (缺省值)
尝试从 CCDT 装入证书撤销配置并按配置执行撤销检查。 如果无法打开 CCDT 文件或无法验证证书 (例如，由于 OCSP 或 CRL 服务器不可用) ，那么 MQCONN 调用将失败。 如果 CCDT 不包含撤销配置，但这不会导致通道失败，那么不会执行撤销检查。

Windows 在 Windows 系统上，还可以使用 Active Directory 进行 CRL 撤销检查。 不能将 Active Directory 用于 OCSP 撤销检查。

如果您正在使用 MQSCO 或 CCDT，那么连接成功。 如果没有 CCDT 文件，并且也未提供 MQSCO，那么连接将失败，原因码为 2059，错误日志报告为 AMQ9518E: 找不到文件 "/var/mqm/AMQCLCHL.TAB"。

可选
对于 REQUIRED，但是如果无法装入证书撤销配置，那么通道不会失败。

DISABLED
不会尝试从 CCDT 装入证书撤销配置，也不会执行证书撤销检查。

注: 如果使用的是 MQCONNX 而不是 MQCONN 调用，那么可以选择通过 MQSCO 提供认证信息记录 (MQAIR)。 因此，如果无法打开 CCDT 文件，那么 MQCONNX 的缺省行为不会失败，而是假定您正在提供 MQAIR (即使您选择不这样做)。

EncryptionPolicySuiteB = 字符串
确定通道是否使用符合 Suite-B 的密码术以及要使用的强度级别。

此属性可由 C 和非受管 .NET 客户机读取。

此属性具有以下可能的值:

NONE
不使用符合 Suite-B 的密码术。 此设置为缺省设置。

128_BIT,192_BIT
将安全强度设置为 128 位和 192 位级别。

128_BIT
将安全强度设置为 128 位级别。

192_BIT
将安全强度设置为 192 位级别。

V 9.2.5 ALW **EnvironmentScope= 过程 | 连接**
控制 IBM MQ 是针对整个过程使用单个 IBM Global Security Kit (GSKit) 环境，还是针对每个连接使用 GSKit 环境。

此属性可由 C 客户机读取。

此属性具有以下可能的值:

PROCESS
单个 GSKit 环境用于进程创建的多个连接。 使用此设置意味着在进程中的所有活动 TLS 连接都停止之前， TLS 密钥库更改将不可用。

该值为缺省值。

CONNECTION
将为同一进程中的每个连接创建 GSKit 环境。 启用这意味着该进程启动的任何新 TLS 连接都将立即获取 TLS 密钥库更改。

警告: 启用此操作方式将导致应用程序使用额外的 CPU 和内存资源来创建每个 GSKit 环境。 此资源消耗随着每个额外的并发 TLS 连接而增加。

ALW V 9.2.0 **MinimumRSAKeySize=int**
指定 RSA 证书必须具有才能接受的最小密钥大小。 允许任何等于或大于 0 的值。 如果未指定，那么缺省为 1。

此属性可由 C/C++ 客户机读取。

OCSPAuthentication = OPTIONAL|REQUIRED|警告
当启用 OCSP 且 OCSP 撤销检查无法确定证书撤销状态时，定义 IBM MQ 的行为。 另请参阅 **ClientRevocationChecks**。

此属性可由 C 和非受管 .NET 客户机读取。

如果没有定义 OCSP 应答器，则不执行 OCSP 撤销检查，允许继续连接。

此属性具有以下可能的值：

可选
接受具有 OCSP 检查无法确定的撤销状态的任何证书，并且不会生成警告或错误消息。 SSL 或 TLS 连接将继续，就像未进行撤销检查一样。

必需
OCSP 检查必须针对检查的每个 SSL 或 TLS 证书生成最终撤销结果。 将拒绝具有无法验证的撤销状态的任何 SSL 或 TLS 证书，并显示错误消息。 如果启用了队列管理器 SSL 事件消息，那么将生成 ReasonQualifier 为 MQRQ_SSL_HANDSHAKE_ERROR 的 MQRC_CHANNEL_SSL_ERROR 消息。 已关闭连接。

该值为缺省值。

WARN
如果 OCSP 撤销检查无法确定任何 SSL 或 TLS 证书的撤销状态，那么将在队列管理器错误日志中报告警告。 如果启用了队列管理器 SSL 事件消息，那么将生成 ReasonQualifier 为 MQRQ_SSL_UNKNOWN_撤销的 MQRC_CHANNEL_SSL_WARNING 消息。 允许连接继续。

OCSPCheckExtensions = 是 | 否
控制 IBM MQ 是否对 AuthorityInfoAccess 证书扩展采取行动。

此属性可由 C 和非受管 .NET 客户机读取。

如果设置为 " 否 "， IBM MQ 将忽略 AuthorityInfoAccess 证书扩展，并不尝试 OCSP 安全检查。 缺省值为 YES。

ALW V 9.2.0 **OCSPTimeout = 数字**
执行撤销检查时等待 OCSP 响应程序的秒数。

此属性可由 C 和非受管 .NET 客户机读取。

V 9.2.0.2 V 9.2.3 对于 来自 IBM MQ 9.2.0 Fix Pack 2 的 Long Term Support 和来自 IBM MQ 9.2.3 的 Continuous Delivery，如果设置了值 0，那么将使用缺省超时 30 秒。

如果未设置任何值，那么将使用 IBM MQ 缺省值 30 秒。

V 9.2.1 **OutboundSNI = channel | hostname**
如果 **OutboundSNI** 设置为 CHANNEL，那么在启动 TLS 连接时，支持 SNI 的客户机会将 SNI 设置为远程系统的目标 IBM MQ 通道名称。

如果此属性设置为 HOSTNAME，那么支持 SNI 的客户机会将 SNI 头设置为主机名，从而导致出站连接请求在 TLS 握手期间接收远程队列管理器的缺省证书，因此无法使用每个通道的证书。

此属性可以由 C， 非受管 .NET， IBM MQ classes for Java 和 IBM MQ classes for JMS 客户机读取。

属性值由 Java/JMS 客户机以区分大小写的方式进行解释，因此应该将 YES/NO 值设置为大写。

V 9.2.0.4 V 9.2.3 从 IBM MQ 9.2.0 Fix Pack 4 for Long Term Support 和 IBM MQ 9.2.3 for Continuous Delivery 开始， IBM MQ 受管 .NET 客户机已更新为将 SERVERNAME 设置为各自的主机名 (如果 **OutboundSNI** 属性设置为 HOSTNAME)。 这允许 IBM MQ 受管 .NET 客户机使用 Red Hat OpenShift 路由连接到队列管理器。 请注意，在 IBM MQ 9.2.0 Fix Pack 4 中，仅从 `mqclient.ini` 文件添加并支持 **OutboundSNI** 属性; 不能从 .NET 应用程序设置该属性。

注: 如果 **OutboundSNI** 设置为 HOSTNAME 的应用程序连接到配置了证书标签的通道，那么会使用 MQRC_SSL_INITIALIZATION_ERROR 拒绝该应用程序，并且会在队列管理器错误日志中显示 AMQ9673 消息。

ALW V 9.2.0 ***PeerCertChainValidation=string***
此属性可由 C 客户机和非受管 .NET 客户机读取。

该字符串可以是以下两个值之一：

- `Usepeerchain` **[缺省]**: 同级提供的证书链可用于在验证证书时弥合任何信任链间隔。 根证书除外。

- `Truststoreonly` **[不推荐]**: 将仅使用信任库中的证书来验证同级的证书。

SSLCryptoHardware = 字符串
设置配置系统上存在的 PKCS #11 加密硬件所需的参数字符串。

此属性可由 C 和非受管 .NET 客户机读取。

指定以下格式的字符串: `GSK_PKCS11 = driver path and filename;token label;token password;symmetric cipher setting;`

`GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;passw0rd;SYMMETRIC_CIPHER_ON` 例如:

驱动程序路径是共享库的绝对路径，提供对 PKCS #11 卡的支持。 驱动程序文件名是共享库的名称。 PKCS #11 驱动程序路径和文件名所需的值的示例为 `/usr/lib/pkcs11/PKCS11_API.so`。 要通过 GSKit 访问对称密码操作，请指定对称密码设置参数。 此参数的值为:

对称密码关闭
请勿访问对称密码操作。 此设置为缺省设置。

对称密码开启
访问对称密码操作。

Linux AIX 提供字符串的不同组成部分时，必须使用反斜杠字符对分号字符进行转义，因为分号字符被视为注释。 例如: `"\;"`

V 9.2.3 您应该保护 **SSLCryptoHardware** 属性字符串中包含的令牌密码。 请参阅 使用加密硬件的 IBM MQ 客户机，以获取更多信息。

LTS 字符串的最大长度为 256 个字符。

V 9.2.3 为了处理加密密码，现在对字符串的长度没有限制。

缺省值为空。 如果指定的字符串格式不正确，那么将生成错误。

SSLFipsRequired = 是|否
指定在 IBM MQ 中执行密码术时是否仅使用 FIPS 认证的算法。

此属性可由 C 客户机和非受管 .NET 客户机读取。

如果配置了加密硬件，那么使用的加密模块是由硬件产品提供的那些模块。 根据正在使用的硬件产品，这些产品可能是 FIPS 认证的，也可能不是 FIPS 认证的特定级别。

SSLHTTPProxyName = 字符串
该字符串是 GSKit 用于 OCSP 检查的 HTTP 代理服务器的主机名或网络地址。 此地址可以后跟可选的端口号，用括号括起。 如果不指定端口号，那么将使用缺省 HTTP 端口 80。

此属性可由 C 客户机和非受管 .NET 客户机读取。

AIX 对于 AIX 上的 32 位客户机，网络地址只能是 IPv4 地址。

在其他平台上，网络地址可以是 IPv4 或 IPv6 地址。

例如，如果防火墙阻止访问 OCSP 应答器的 URL，则可能需要使用该属性。

ALW V 9.2.0 **SSLHTTPConnectTimeout = 数字 | 0**
执行撤销检查时，为成功建立与 HTTP 服务器的网络连接而等待的秒数。

此属性可由 C 和非受管 .NET 客户机读取。

如果未设置任何值，那么将使用 IBM MQ 缺省值 0 (关闭)。

SSLKeyRepository = 路径名
保存用户数字证书的密钥存储库的位置 (以系统格式表示)。 即，它包含不带扩展名的完整路径和文件名。

此属性可由 C 客户机和非受管 .NET 客户机读取。

SSLKeyResetCount = 整数|0
在重新协商密钥之前，在 TLS 通道上发送和接收的未加密字节数。

此属性可由 C 客户机和非受管 .NET 客户机读取。

该值必须在 0-999999999 范围内。

缺省值为 0，这意味着永远不会重新协商密钥。

如果指定值 1-32768，那么 TLS 通道将使用密钥重置计数 32768 (32Kb)。 这是为了避免过多的密钥重置，这将发生在小型密钥重置值上。

Multi 客户机配置文件的 TCP 节

使用 TCP 节来指定 TCP 网络协议配置参数。

注: 此节的每个属性的描述都指示哪些 IBM MQ 客户机可读取此属性。 要获取所有 IBM MQ MQI client 配置文件节的摘要表，请参阅每个客户机可读取的 IBM MQ 属性。

以下属性可以包含在 TCP 节中:

ClntRcvBuffSize = 数字 | 0
客户机连接服务器连接通道的客户机端使用的 TCP/IP 接收缓冲区的大小 (以字节计)。

此属性可以由 C，非受管 .NET， IBM MQ classes for Java， IBM MQ classes for JMS，受管 .NET 和受管 XMS .NET 客户机读取。

如果没有设置值，则使用 IBM MQ 默认值 32768。

如果该值设置为零，那么将使用操作系统缺省值。 值为零表示操作系统将管理缓冲区大小，而不是由 IBM MQ 固定缓冲区大小。

V 9.2.0.35 从 IBM MQ 9.2.0 Fix Pack 35，如果 IBM MQ Java 客户端将该值设为零， Java Message Queuing Interface (JMQI) 就会使用网络层提供的缓冲区大小。 这解决了之前的一个问题，即在与队列管理器通信时，将该属性设置为零会错误地导致 JMQI 使用大小为 32K 的 TCP/IP 接收缓冲区。 接收缓冲区大小将写入 JMSCS0062IBM MQ classes for Java 或 IBM MQ classes for JMS 日志文件中的信息。

ClntSndBuffSize = 数字 | 0
客户机连接服务器连接通道的客户机端所使用的 TCP/IP 发送缓冲区的大小 (以字节计)。

此属性可以由 C，非受管 .NET， IBM MQ classes for Java， IBM MQ classes for JMS，受管 .NET 和受管 XMS .NET 客户机读取。

如果没有设置值，则使用 IBM MQ 默认值 32768。

如果该值设置为零，那么将使用操作系统缺省值。 值为零表示操作系统将管理缓冲区大小，而不是由 IBM MQ 固定缓冲区大小。

V 9.2.0.35 从 IBM MQ 9.2.0 Fix Pack 35，如果 IBM MQ Java 客户端将该值设为零， Java Message Queuing Interface (JMQI) 就会使用网络层提供的缓冲区大小。 这解决了之前的一个问题，即在与队列管理器通信时，将该属性设置为零会错误地导致 JMQI 使用大小为 32K 的 TCP/IP 发送缓冲区。 发送缓冲区大小将写入 JMSCS0061IBM MQ classes for Java 或 IBM MQ classes for JMS 日志文件中的消息。

Connect_Timeout = 数字
尝试连接套接字超时之前的秒数。

如果 **ConnectTimeout** = 0，那么调用 connect () 是分块调用，而 connect () 调用超时取决于内部重试的值，这将导致大约两分钟的超时。

此属性可以由 C，非受管 .NET， IBM MQ classes for Java 和 IBM MQ classes for JMS 客户机读取。

IBM MQ 通道进程通过非阻塞套接字连接。 因此，如果套接字的另一端未就绪，那么 connect () 将立即返回 *EINPROGRESS* 或 *EWOULDBLOCK*。 在此之后，将不会尝试重新连接。

如果 Connect_Timeout 设置为非零值，那么 IBM MQ 将等待指定的时间段 over select () 调用，以便套接字准备就绪。 这将增加后续 connect () 调用成功的机会。 在由于网络上的高负载而导致连接需要一些等待周期的情况下，此选项可能是有益的。

Connect_Timeout、 ClntSndBuffSize, 和 ClntRcvBuffSize 参数之间没有关系。

IPAddressVersion = MQIPADDR_IPV4 | MQIPADDR_IPV6
指定要用于通道连接的 IP 协议。

此属性可以由 C， 非受管 .NET，受管 .NET 和受管 XMS .NET 客户机读取。

它具有可能的字符串值 MQIPADDR_IPV4 或 MQIPADDR_IPV6。 这些值与 **ALTER QMGR IPADDRV** 中的 IPV4 和 IPV6 具有相同的含义。

KeepAlive = 是 | 否
打开或关闭 KeepAlive 功能。 KeepAlive=YES 使 TCP/IP 定期检查连接的另一端是否仍然可用。 如果不是，那么将关闭通道。

此属性可以由 C， 非受管 .NET， IBM MQ classes for Java， IBM MQ classes for JMS，受管 .NET 和受管 XMS .NET 客户机读取。

Windows **Library1 = *DLL* 名称 | WSOCK32**
(仅限 Windows) TCP/IP 套接字 DLL 的名称。

此属性可由 C 和非受管 .NET 客户机读取。

Multi 活动跟踪配置文件 mqat.ini

活动跟踪配置文件 mqat.ini 用于配置活动跟踪行为。 此文件用于定义报告活动跟踪数据的级别和频率。 该文件还提供了一种方法来定义规则，以根据应用程序的名称来启用和禁用活动跟踪。

mqat.ini 文件遵循与 mqs.ini 和 qm.ini 文件相同的节键/参数/值对格式。 该文件由单个节 " AllActivity 跟踪" 组成，该节用于配置缺省情况下针对所有活动跟踪报告活动跟踪数据的级别和频率。 该文件还可以包含多个 ApplicationTrace 节。 这些节中的每个节都根据与规则的连接的应用程序名称匹配，为一个或多个连接的跟踪行为定义一条规则。 有关更多信息，请参阅 应用程序活动跟踪 和 使用 mqat.ini 配置活动跟踪行为。

队列管理器应用许多规则来确定要用于连接的节设置。 (可选) 您可以覆盖与 ApplicationTrace 节匹配的那些连接的 AllActivityTrace 节下的全局跟踪级别和频率设置。 有关更多信息，请参阅 使用 mqat.ini 配置活动跟踪行为。

目录位置

IBM i Linux AIX 在 AIX and Linux 和 IBM i 系统上， mqat.ini 位于与 qm.ini 文件相同的队列管理器数据目录中。

Windows 在 Windows 系统上， mqat.ini 位于队列管理器数据目录 C:\Program Files\IBM\WebSphere MQ\qmgrs\queue_manager_name 中。 运行要跟踪的应用程序的用户需要具有读取此文件的许可权。

Multi AllActivity mqat.ini 文件的 Trace 节

mqat.ini 配置文件的 AllActivityTrace 节指定用于配置队列管理器的跟踪级别的参数。

单个 AllActivity 跟踪节定义应用于所有 IBM MQ 连接的活动跟踪设置，除非被覆盖。

AllActivityTrace 节中的各个值可以被 ApplicationTrace 节中的更具体的信息覆盖。

如果指定了多个 AllActivity 跟踪节，那么将使用最后一个节中的值。 所选 AllActivity 跟踪中缺少的参数采用缺省值。 将忽略先前 AllActivity 跟踪节中的参数和值。

ActivityInterval
跟踪消息之间的时间间隔 (以秒为单位)。 活动跟踪不使用计时器线程，因此跟踪消息不会在经过时间的确切时刻写入，而是在经过时间间隔之后执行第一个 MQI 操作时写入。 如果此值为 0， 那么将在连接断开时 (或达到活动计数时) 写入跟踪消息。 缺省值为 1。

ActivityCount
跟踪消息之间的 MQI 操作数。 如果此值为 0 ，那么将在连接断开 (或活动时间间隔已过) 时写入跟踪消息。 缺省值为 100。

TraceLevel
针对每个操作跟踪的参数详细信息量。 各个操作的描述详细说明了针对每个跟踪级别包含哪些参数。 设置为 LOW， MEDIUM 或 HIGH。 缺省为 MEDIUM。

TraceMessage 数据
针对 MQGET， MQPUT， MQPUT1 和回调操作跟踪的消息数据量 (以字节计)。 缺省值为 0。

StopOnGetTrace 消息
可以设置为 ON 或 OFF。 缺省为 ON。

SubscriptionDelivery
可以设置为 **已批处理** 或 **立即**。 确定当存在一个或多个活动跟踪预订时是否使用 **ActivityInterval** 和 **ActivityCount** 参数。 如果将此参数设置为 IMMEDIATE，那么当跟踪数据具有匹配的预订时，将使用有效值 1 覆盖 **ActivityInterval** 和 **ActivityCount** 值。 不会将每个活动跟踪记录与来自同一连接的其他记录一起批处理，而是立即交付到预订而不会延迟。 IMMEDIATE 设置将增加收集活动跟踪数据的性能开销。 缺省设置为 BATCHED。

相关任务
使用 mqat.ini 配置活动跟踪行为

Multi mqat.ini 文件的 ApplicationTrace 节

mqat.ini 配置文件可以包含多个 ApplicationTrace 节。 这些节中的每个节都根据与规则的连接的应用程序名称相匹配来定义一个或多个连接的跟踪行为规则。

可以为 ApplicationTrace 节设置以下值:

跟踪
可设置为 ON 或 OFF 的活动跟踪开关。 **Trace** 参数是必需参数，没有缺省值。 它可以在特定于应用程序的节中用于确定活动跟踪对于当前应用程序节的作用域是否处于活动状态。 请注意，此值将覆盖队列管理器的 **ACTVTRC** 和 **ACTVCONO** 设置。

ApplName
ApplName 参数指定为字符串，并且是必需参数，没有缺省值。 此值用于确定 ApplicationTrace 节应用于哪些应用程序。 它与 API 出口上下文结构 (相当于 MQMD.PutApplName) 中的 **ApplName** 值匹配。 **ApplName** 值的内容因应用程序环境而异。

在多平台上，仅 MQAXC.ApplName 与节中的值匹配。 进行比较时，将忽略最右边路径分隔符左边的字符。

可以在 **ApplName** 值的末尾使用单个通配符 (*) 来匹配该点之后的任意数量的字符。 如果 **ApplName** 值设置为单个通配符 (*)，那么 **ApplName** 值将与所有应用程序匹配。

IBM i **ApplFunction**
ApplFunction 参数指定为字符串。 缺省值为 *。 此参数的值用于限定 ApplicationTrace 节和 **ApplName** 值应用于的应用程序。

该节是可选的，仅对 IBM i 队列管理器有效。 可以在 **ApplName** 值的末尾使用单个通配符 (*) 来匹配任意数量的字符。 例如，指定 **ApplName** = * 和 **ApplFunction** = *AMQSPUT0* 的 ApplicationTrace 节适用于来自任何作业的 AMQSPUT0 程序的所有调用。

ApplClass
ApplClass 参数定义应用程序的类，并且可以设置为以下值:

- 用户
- MCA
- ALL (这是缺省值)

有关 **AppType** 值如何对应于 IBM MQ 连接的说明，请参阅 使用 mqat.ini 配置活动跟踪行为中的 表 3 。

(可选) 对于与 ApplicationTrace 节匹配的那些连接，可以覆盖 AllActivityTrace 节下的全局跟踪级别和频率设置。

可以在 ApplicationTrace 节下设置以下参数。 如果未设置这些值，那么将从 AllActivity 跟踪节 设置继承该值:

ActivityInterval
跟踪消息之间的时间间隔 (以秒为单位)。 活动跟踪不使用计时器线程，因此跟踪消息不会在经过时间的确切时刻写入，而是在经过时间间隔之后执行第一个 MQI 操作时写入。 如果此值为 0，那么将在连接断开时 (或达到活动计数时) 写入跟踪消息。 缺省值为 1。

ActivityCount
跟踪消息之间的 MQI 操作数。 如果此值为 0，那么将在连接断开 (或活动时间间隔已过) 时写入跟踪消息。 缺省值为 100。

TraceLevel
针对每个操作跟踪的参数详细信息量。 各个操作的描述详细说明了针对每个跟踪级别包含哪些参数。 设置为 LOW， MEDIUM 或 HIGH。 缺省为 MEDIUM。

TraceMessage 数据
针对 MQGET， MQPUT， MQPUT1 和回调操作跟踪的消息数据量 (以字节计)。 缺省值为 0。

StopOnGetTrace 消息
可以设置为 ON 或 OFF。 缺省为 ON。

相关任务
使用 mqat.ini 配置活动跟踪行为

配置分布式队列

此部分提供有关 IBM MQ 安装之间的相互通信的更详细信息，包括队列定义，通道定义，触发和同步点过程

开始之前

在阅读本部分之前，了解 分布式排队和集群中引入的通道，队列和其他概念很有用。

如果需要连接位于不同物理网络上的两个队列管理器，或者需要通过防火墙进行通信，那么使用 IBM MQ Internet Pass-Thru 可能会简化配置。 有关更多信息，请参阅 IBM MQ Internet Pass-Thru。

过程

- 使用以下子主题中的信息通过分布式排队连接应用程序:
  - 第 153 页的『IBM MQ 分布式排队技术』
  - 第 170 页的『分布式队列管理简介』
  - 第 173 页的『如何将消息发送到另一个队列管理器』
  - 第 191 页的『触发通道』
  - 第 189 页的『消息的安全性』
  - ALW 第 197 页的『监视和控制 AIX, Linux, and Windows 上的通道』
  - IBM i 第 216 页的『监视和控制 IBM i 上的通道』

相关概念
第 734 页的『设置 IBM MQ for z/OS』
使用本主题作为定制 IBM MQ for z/OS 系统的逐步指南。

相关任务
第 13 页的『配置客户机与服务器之间的连接』
要配置 IBM MQ MQI clients 与服务器之间的通信链路，请决定通信协议，定义链路两端的连接，启动侦听器以及定义通道。

第 235 页的『配置队列管理器集群』

集群提供了一种用于以简化初始配置和持续管理的方式互连队列管理器的机制。 您可以定义集群组件，以及创建和管理集群。

第 71 页的『在 Multiplatforms 版上的 .ini 文件中更改 IBM MQ 配置信息』
您可以通过编辑配置 (`.ini`) 文件中的信息来更改 IBM MQ 或单个队列管理器的行为，以满足安装需要。 您还可以更改 IBM MQ MQI clients 的配置选项。

第 730 页的『在 z/OS 上配置队列管理器』
使用以下指示信息在 IBM MQ for z/OS 上配置队列管理器。

第 797 页的『在 z/OS 上设置与其他队列管理器的通信』
本部分描述了在开始使用分布式排队之前需要进行的 IBM MQ for z/OS 准备工作。

IBM MQ 分布式排队技术

本部分中的子主题描述了在规划通道时使用的方法。 这些子主题描述了帮助您规划如何将队列管理器连接在一起以及管理应用程序之间的消息流的方法。

有关消息通道规划示例，请参阅:

- ALW AIX, Linux, and Windows 的消息通道规划示例
- IBM i IBM i 的消息通道规划示例
- z/OS z/OS 的消息通道规划示例
- z/OS 使用队列共享组的 z/OS 的消息通道规划示例

相关概念
通道
消息排队简介
分布式排队和集群
相关任务
第 152 页的『配置分布式队列』
此部分提供有关 IBM MQ 安装之间的相互通信的更详细信息，包括队列定义，通道定义，触发和同步点过程
相关参考
示例配置信息

消息流控制 (message flow control)

消息流控制是一项任务，涉及在队列管理器之间设置和维护消息路由。 对于通过多个队列管理器的多跳的路由而言，这很重要。 本部分描述如何在系统上使用队列，别名队列定义和消息通道来实现消息流控制。

您可以使用 第 152 页的『配置分布式队列』中引入的多种方法来控制消息流。 如果您的队列管理器位于集群中，那么将使用不同的方法来控制消息流，如 第 153 页的『消息流控制 (message flow control)』中所述。 z/OS 如果队列管理器位于队列共享组中，并且已启用组内排队 (IGQ)，那么消息流可由 IGQ 代理程序控制。 组内排队中描述了这些代理程序。

您可以使用以下对象来实现消息流控制:

- 传输队列
- 消息通道
- 远程队列定义
- 队列管理器别名定义
- 应答队列别名定义

在 对象类型中描述了队列管理器和队列对象。 分布式排队组件中描述了消息通道。 以下方法使用这些对象在系统中创建消息流:

- 将消息放入远程队列

- 通过特定传输队列进行路由
- 接收消息
- 通过系统传递消息
- 分隔消息流
- 将消息流切换到另一个目标
- 将应答队列名称解析为别名

注

本节中描述的所有概念都与网络中的所有节点相关，并且包括消息通道的发送和接收端。 因此，在大多数示例中仅说明了一个节点。 例外情况是，此示例需要管理员在消息通道的另一端进行显式合作。

在继续使用各个方法之前，重新映射名称解析的概念以及使用远程队列定义的三种方法很有用。 请参阅分布式排队和集群。

相关概念

第 154 页的『传输头中的队列名称』
目标队列名称与传输头中的消息一起传递，直到到达目标队列为止。

第 154 页的『如何创建队列管理器和应答别名』
本主题说明了创建远程队列定义的三种方法。

传输头中的队列名称

目标队列名称与传输头中的消息一起传递，直到到达目标队列为止。

应用程序使用的队列名称 (即逻辑队列名称) 由队列管理器解析为目标队列名称。 换言之，物理队列名称。 此目标队列名称与消息一起在单独的数据区 (传输头) 中传输，直到到达目标队列为止。 然后，将除去传输头。

创建并行服务类时，可更改此队列名称的队列管理器部分。 请记住，当到达服务等级转移结束时，将队列管理器名称返回到原始名称。

如何创建队列管理器和应答别名

本主题说明了创建远程队列定义的三种方法。

以三种不同的方式使用远程队列定义对象。 第 155 页的表 17 说明了如何定义以下三种方法中的每种方法:

- 使用远程队列定义来重新定义本地队列名称。

  打开队列时，应用程序仅提供队列名称，此队列名称是远程队列定义的名称。

  远程队列定义包含目标队列和队列管理器的名称。 (可选) 定义可以包含要使用的传输队列的名称。 如果未提供传输队列名称，那么队列管理器将使用从远程队列定义获取的队列管理器名称作为传输队列名称。 如果未定义此名称的传输队列，但定义了缺省传输队列，那么将使用缺省传输队列。
- 使用远程队列定义来重新定义队列管理器名称。

  应用程序或通道程序在打开队列时提供队列名称和远程队列管理器名称。

  如果您提供了与队列管理器名称同名的远程队列定义，并且将定义中的队列名称留空，那么队列管理器会将打开的调用中的队列管理器名称替换为定义中的队列管理器名称。

  此外，该定义可以包含要使用的传输队列的名称。 如果未提供传输队列名称，那么队列管理器将获取来自远程队列定义的队列管理器名称作为传输队列名称。 如果未定义此名称的传输队列，但定义了缺省传输队列，那么将使用缺省传输队列。
- 使用远程队列定义来重新定义应答队列名称。

  每次应用程序将消息放入队列时，它都可以为应答消息提供应答队列的名称，但队列管理器名称为空。

  如果提供与应答队列同名的远程队列定义，那么本地队列管理器会将应答队列名称替换为定义中的队列名称。

可以在定义中提供队列管理器名称，但不能提供传输队列名称。

表 *17:* 使用远程队列定义对象的三种方法			
用法	**队列管理器名称**	**队列名称**	**传输队列的名称**
1. 远程队列定义 (在 OPEN 调用时)			
在调用中提供	空白或本地 QM	(*) 必需	不适用
在定义中提供	必需	必需	可选
2. 队列管理器别名 (在 OPEN 调用时)			
在调用中提供	(*) 必需，而不是本地 QM	必需	不适用
在定义中提供	必需	空白	可选
3. 应答队列别名 (在 PUT 调用时)			
在调用中提供	空白	(*) 必需	不适用
在定义中提供	可选	可选	空白

注: (*) 表示此名称是定义对象的名称

有关正式描述，请参阅 队列名称解析。

将消息放在远程队列上

您可以使用远程队列定义对象将队列名称解析为传输队列到相邻队列管理器。

在分布式排队环境中，无论消息来自本地系统中的应用程序，还是通过来自相邻系统的通道到达，传输队列和通道都是指向某个位置的所有消息的焦点。 第 155 页的图 8 显示了将消息放置在名为 "QA_norm" 的逻辑队列上的应用程序。 名称解析使用远程队列定义 "QA_norm" 来选择传输队列 QMB。 然后，它将传输头添加到消息中，声明 "QA_norm at QMB"。

例如，从 "Channel_back" 上的相邻系统到达的消息具有物理队列名称为 "QA_norm at QMB" 的传输头。 这些消息在传输队列 QMB 上保持不变。

通道将消息移动到相邻队列管理器。

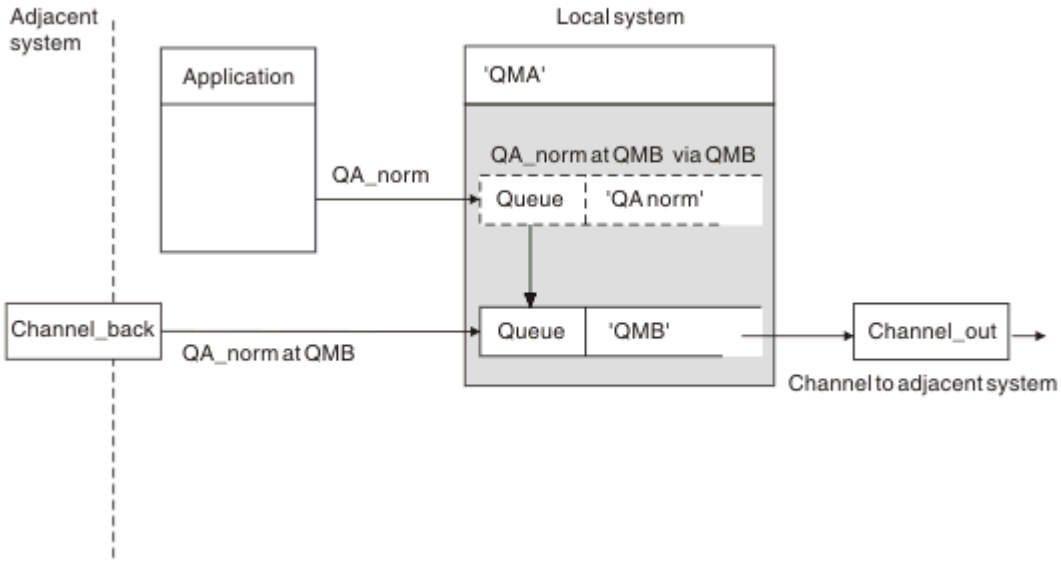


图 *8:* 远程队列定义用于将队列名解析为传输队列到相邻队列管理器

如果您是 IBM MQ 系统管理员，那么必须:

- 定义来自相邻系统的消息通道
- 定义到相邻系统的消息通道
- 创建传输队列 QMB
- 定义远程队列对象 "QA_norm" 以将应用程序使用的队列名称解析为目标队列名称，目标队列管理器名称和传输队列名称

在集群环境中，您只需要在本地队列管理器中定义集群接收方通道。 您不需要定义传输队列或远程队列对象。 请参阅 集群 (Clusters)。

有关名称解析的更多信息

远程队列定义的作用是定义物理目标队列名称和队列管理器名称。 这些名称将放入消息的传输头中。

来自相邻系统的入局消息已具有由原始队列管理器执行的此类名称解析。 因此，它们具有显示物理目标队列名称和队列管理器名称的传输头。 这些消息不受远程队列定义影响。

相关参考
队列名称解析

选择传输队列

您可以使用远程队列定义来允许不同的传输队列将消息发送到同一相邻队列管理器。

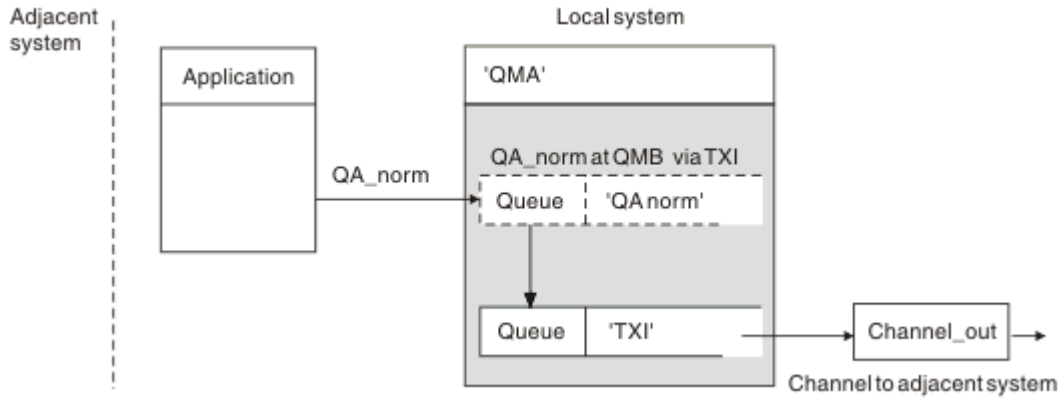


图 9: 远程队列定义允许使用其他传输队列

在分布式排队环境中，当您需要将消息流从一个通道更改为另一个通道时，请使用相同的系统配置，如 第 155 页的『将消息放在远程队列上』 中的 第 155 页的图 8 中所示。 本主题中的 第 156 页的图 9 显示了如何使用远程队列定义将消息通过不同的传输队列 (因此通过不同的通道) 发送到同一相邻队列管理器。

对于 第 156 页的图 9 中显示的配置，必须提供远程队列对象 "QA_norm" 和传输队列 "TX1"。 必须提供 "QA_norm" 以在远程队列管理器，传输队列 "TX1" 和队列管理器 "QMB_priority" 中选择 "QA_norm" 队列。 在与系统相邻的通道的定义中指定 "TX1"。

消息将放置在传输队列 "TX1" 上，其中的传输头包含 "QA_norm at QMB_priority" ，并通过通道发送到相邻系统。

此图中已排除 channel_back ，因为它需要队列管理器别名。

在集群环境中，您不需要定义传输队列或远程队列定义。 有关更多信息，请参阅第 236 页的『定义集群队列』 。

接收消息

您可以配置队列管理器以从其他队列管理器接收消息。 必须确保不会发生无意的名称解析。

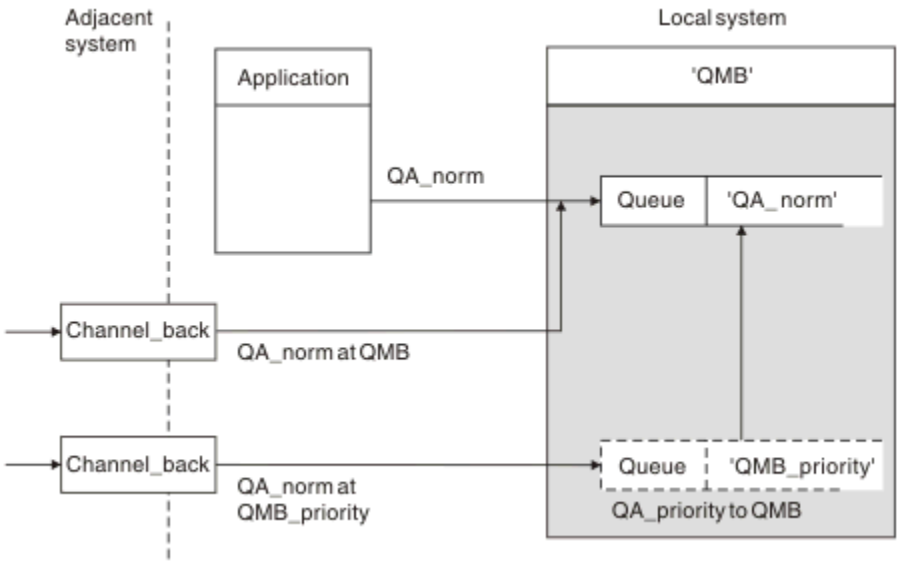


图 *10:* 直接接收消息并解析别名队列管理器名称

除了安排发送消息外，系统管理员还必须安排从相邻队列管理器接收消息。 接收到的消息在传输头中包含目标队列管理器和队列的物理名称。 它们与来自同时指定队列管理器名称和队列名称的本地应用程序的消息相同。 由于此处理，您需要确保进入系统的消息不会执行无意的名称解析。 请参阅 第 157 页的图 10 以了解此方案。

对于此配置，必须准备:

- 用于从相邻队列管理器接收消息的消息通道
- 用于将入局消息流 "QMB_priority" 解析为本地队列管理器名称 "QMB" 的队列管理器别名定义
- 本地队列 "QA_norm" (如果它不存在)

接收别名队列管理器名称

在此图中使用队列管理器别名定义时，未选择其他目标队列管理器。 通过此本地队列管理器并寻址到 "QMB_priority" 的消息适用于队列管理器 "QMB"。 别名队列管理器名称用于创建单独的消息流。

通过系统传递消息

您可以通过三种方式通过系统传递消息: 使用位置名，使用队列管理器的别名或选择传输队列。

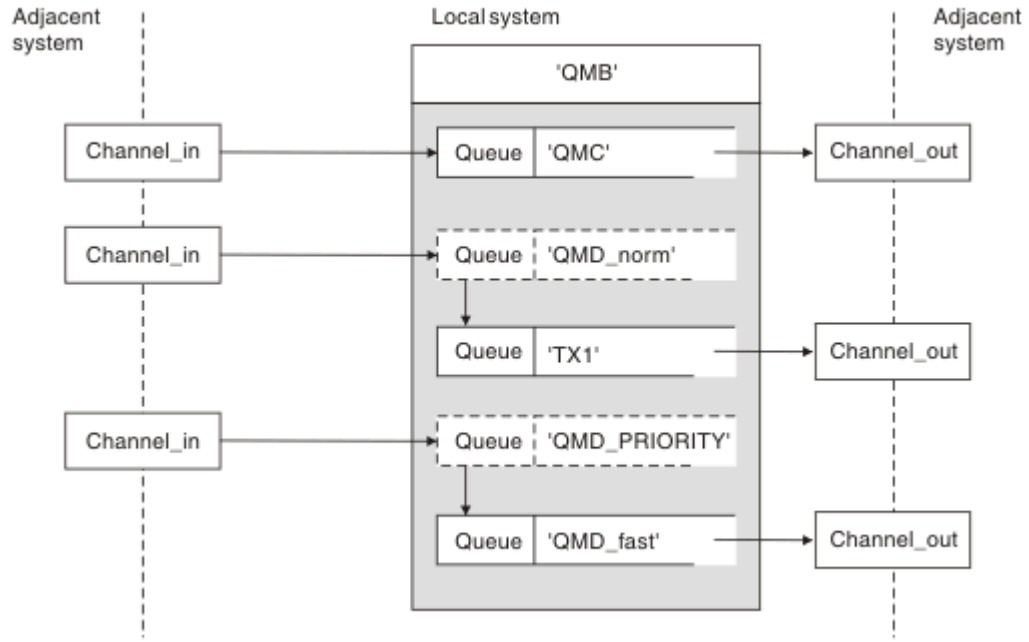


图 *11:* 通过系统传递消息的三种方法

第 156 页的『接收消息』中的 第 157 页的图 10 中显示的方法显示了如何捕获别名流。 第 158 页的图 11 说明了通过整合先前描述的方法来构建网络的方式。

配置显示了一个通道，用于传递三个具有不同目标的消息:

1. QB 在 QMC
2. QB 在 QMD_norm
3. QB 在 QMD_PRIORITY

您必须在未更改的情况下通过系统传递第一个消息流。 必须通过不同的传输队列和通道传递第二个消息流。 对于第二个消息流，您还必须将别名队列管理器名称 QMD_norm 的消息解析为队列管理器 QMD。 第三消息流选择不同的传输队列而不进行任何其他更改。

在集群环境中，消息通过集群传输队列传递。 通常，单个传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 将所有消息传输到队列管理器所属的所有集群中的所有队列管理器; 请参阅 队列管理器集群。 您可以为队列管理器所属的集群中的所有或部分队列管理器定义单独的传输队列。

以下方法描述适用于分布式排队环境的技术。

使用这些方法

对于这些配置，必须准备:

- 输入通道定义
- 输出通道定义
- 传输队列:
  - QMC
  - TX1
  - QMD_fast
- 队列管理器别名定义:
  - QMD_norm 与 QMD_norm 到 QMD 到 TX1
  - QMD_PRIORITY 与 QMD_PRIORITY 到 QMD_PRIORITY 到 QMD_fast

注: 示例中显示的任何消息流都不会更改目标队列。 队列管理器名称别名提供消息流的分隔。

方法 1: 使用传入位置名

您将接收包含另一个位置名 (例如 QMC) 的传输头的消息。 最简单的配置是创建具有该名称 QMC 的传输队列。 服务传输队列的通道将消息按原样传递到下一个目标。

方法 2: 对队列管理器使用别名

第二种方法是使用队列管理器别名对象定义，但指定新的位置名 QMD 和特定的传输队列 TX1。 此操作:

- 终止由队列管理器名称别名 QMD_norm(即，指定的服务类 QMD_norm) 设置的别名消息流。
- 将这些消息上的传输头从 QMD_norm 更改为 QMD。

方法 3: 选择传输队列

第三种方法是使用与目标位置 QMD_PRIORITY 相同的名称来定义队列管理器别名对象。 使用队列管理器别名定义来选择特定传输队列 QMD_fast 以及其他通道。 这些消息上的传输头保持不变。

分隔消息流

您可以使用队列管理器别名来创建单独的消息流，以将消息发送到同一队列管理器。

在分布式排队环境中，由于多种原因，可能需要将同一队列管理器中的消息分隔到不同的消息流中。 例如:

- 您可能需要为大型，中型和小型消息提供单独的流。 此需求也适用于集群环境，在此情况下，您可以创建重叠的集群。 您可以这样做的原因有很多，例如:
  - 让不同的组织有自己的管理。
  - 允许单独管理独立应用程序。
  - 创建服务类。 例如，您可以有一个名为 STAFF 的集群，该集群是名为决议的集群的子集。 将消息放入 STAFF 集群中公布的队列时，将使用受限通道。 将消息放入同学们集群中公布的队列时，可以使用常规通道或受限通道。
  - 创建测试和生产环境。
- 可能需要通过与本地生成的消息的路径不同的路径来路由入局消息。
- 您的安装可能需要在特定时间 (例如，隔夜) 调度消息移动，然后需要将消息存储在保留队列中，直到调度为止。

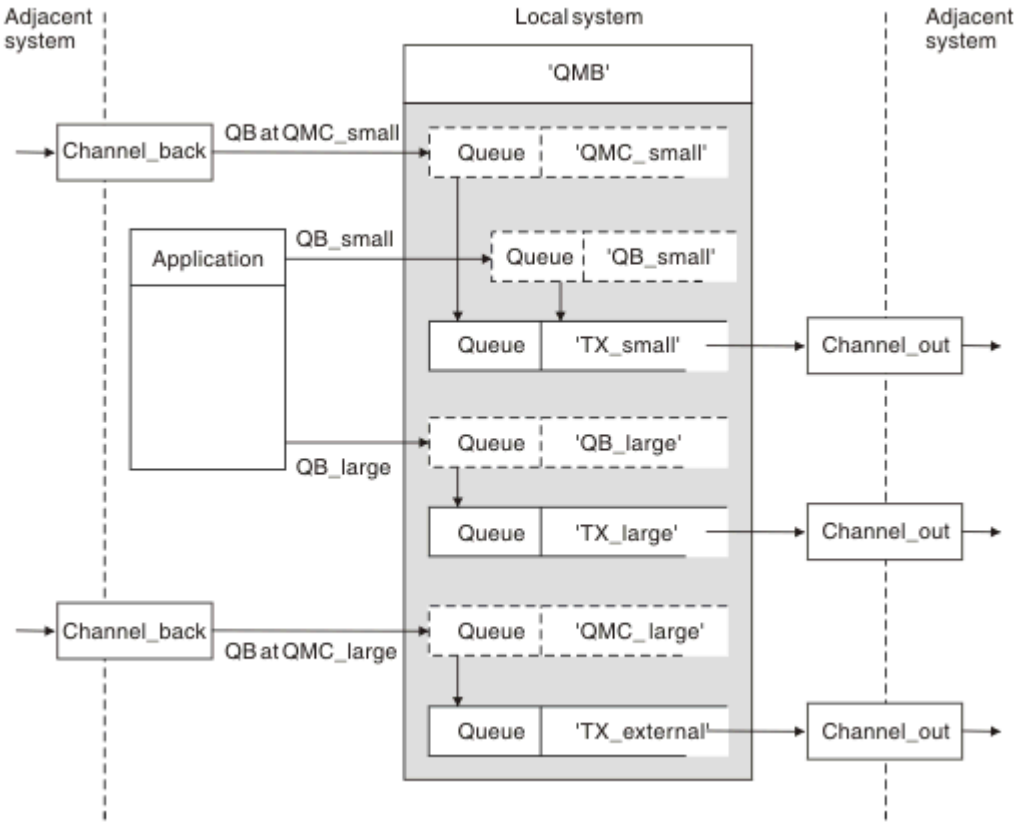


图 *12:* 分隔消息流

在 第 160 页的图 12 中显示的示例中，这两个入局流将别名队列管理器名称 "QMC_small" 和 "QMC_large"。 为这些流提供队列管理器别名定义以捕获本地队列管理器的这些流。 您有一个应用程序对两个远程队列进行寻址，并且需要使这些消息流保持独立。 您提供两个远程队列定义，用于指定相同的位置 "QMC"，但指定不同的传输队列。 此定义使流保持独立，并且在远端不需要任何额外的内容，因为它们在传输头中具有相同的目标队列管理器名称。 您提供:

- 入局通道定义
- 两个远程队列定义 QB_small 和 QB_large
- 两个队列管理器别名定义 QMC_small 和 QMC_large
- 三个发送通道定义
- 三个传输队列: TX_small， TX_large 和 TX_external

与邻近系统的协调

当您使用队列管理器别名来创建单独的消息流时，需要与消息通道远程端的系统管理员协调此活动，以确保相应的队列管理器别名在那里可用。

将消息集中到不同的位置

您可以将发往各个位置的消息集中到单个通道。

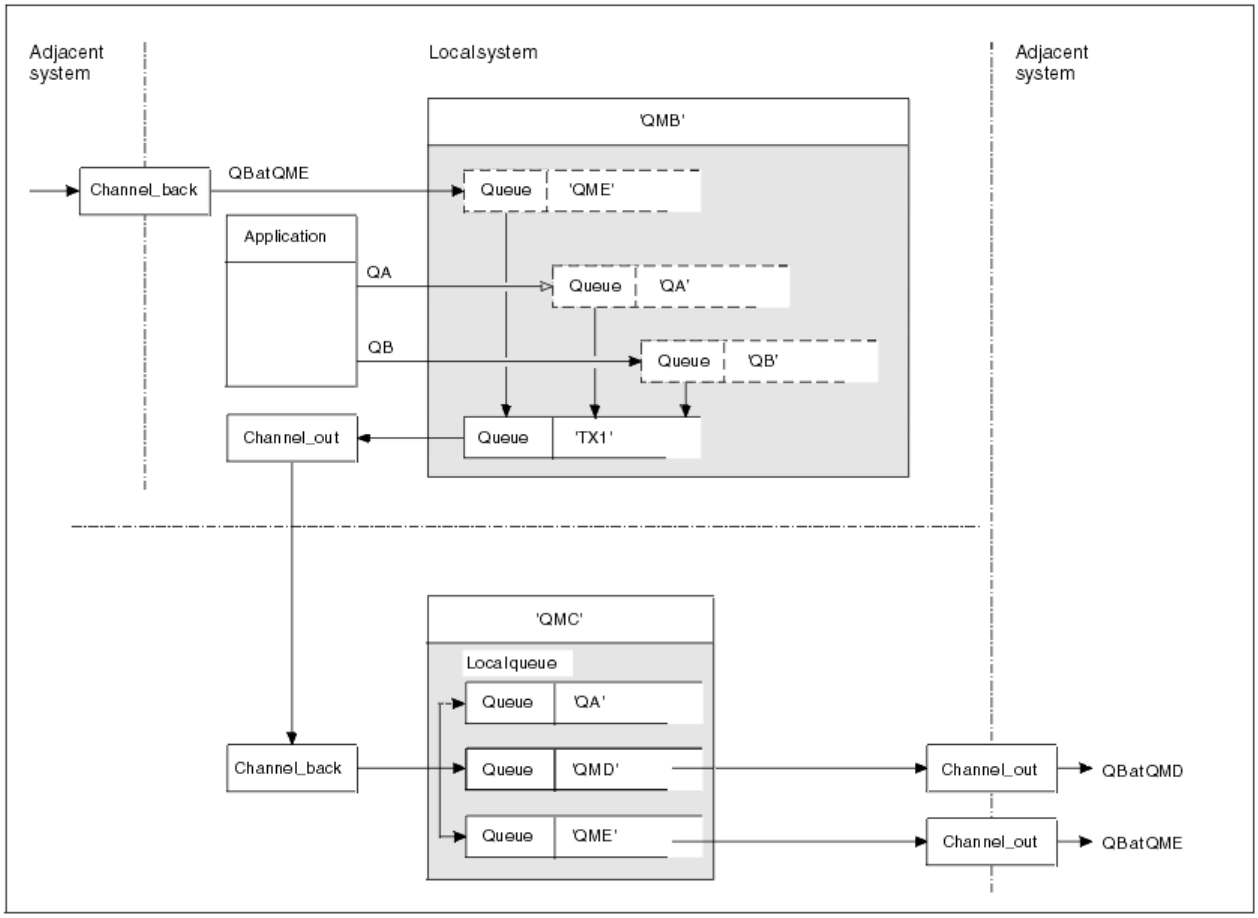


图 *13:* 将消息流组合到通道

第 161 页的图 13 说明了一种分布式排队技术，用于将发往不同位置的消息集中到一个通道上。 两种可能的用途是:

- 通过网关集中消息流量
- 在节点之间使用宽带宽高速公路

在此示例中，来自不同源 (本地和相邻) 以及具有不同目标队列和队列管理器的消息将通过传输队列 "TX1" 流向队列管理器 QMC。 队列管理器 QMC 根据目标传递消息。 一个设置为传输队列 "QMD"，用于继续传输到队列管理器 QMD。 另一个设置为传输队列 "QME"，用于继续传输到队列管理器 QME。 其他消息将放在本地队列 "QA" 上。

您必须提供:

- 通道定义
- 传输队列 TX1
- 远程队列定义:
  - QA 与 "QA 在 QMC 通过 TX1"
  - QB 与 "QB at QMD through TX1"
- 队列管理器别名定义:
  - QME 与 "QME 到 TX1"

配置 QMC 的补充管理员必须提供:

- 接收具有相同通道名称的通道定义
- 具有关联发送通道定义的传输队列 QMD
- 具有关联发送通道定义的传输队列 QME

- 本地队列对象 QA。

将消息流转移到另一个目标

您可以使用队列管理器别名和传输队列来重新定义某些消息的目标。

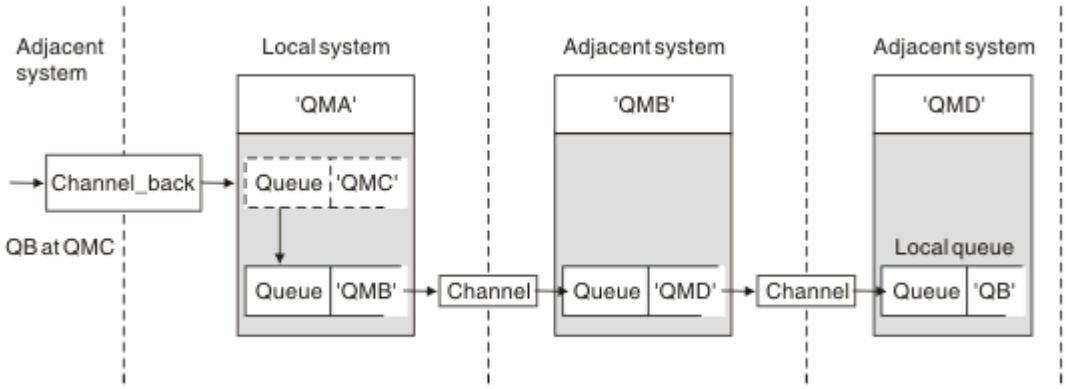


图 *14:* 将消息流转移到另一个目标

第 162 页的图 14 说明了如何重新定义特定消息的目标。 发送到 QMA 的入局消息将以 "QB at QMC" 为目标。 它们通常到达 QMA，并被放置在称为 QMC 的传输队列上，该传输队列已成为 QMC 通道的一部分。 QMA 必须将消息转移到 QMD，但只能通过 QMB 访问 QMD。 当您需要将服务从一个位置移动到另一个位置时，此方法很有用，并且允许订户继续临时发送消息，直到他们调整到新地址为止。

将发往特定队列管理器的入局消息重新路由到其他队列管理器的方法使用:

- 队列管理器别名，用于将目标队列管理器更改为另一个队列管理器，并选择到相邻系统的传输队列
- 用于为相邻队列管理器提供服务的传输队列
- 相邻队列管理器上的传输队列，用于继续路由到目标队列管理器

您必须提供:

- Channel_back 定义
- 队列管理器别名对象定义 QMC， QB 位于 QMD 到 QMB
- Channel_out 定义
- 关联的传输队列 QMB

正在配置 QMB 的补充管理员必须提供:

- 相应的 channel_back 定义
- 传输队列 QMD
- QMD 的关联通道定义

可以在集群环境中使用别名。 有关更多信息，请参阅第 314 页的『队列管理器别名和集群』。

将消息发送到分发列表

您可以使用单个 MQPUT 调用使应用程序向多个目标发送消息。

在除 z/OS 以外的所有平台上的 IBM MQ 中，应用程序可以使用单个 MQPUT 调用将消息发送到多个目标。 您可以在分布式排队环境和集群环境中执行此操作。 您必须在分发列表中定义目标，如 分发列表中所述。

并非所有队列管理器都支持分发列表。 当 MCA 与合作伙伴建立连接时，它会确定合作伙伴是否支持分发列表，并相应地在传输队列上设置标志。 如果应用程序尝试发送以分发列表为目标的消息，但合作伙伴不支持分发列表，那么发送 MCA 会拦截该消息并将其放入每个预期目标的传输队列中一次。

接收 MCA 确保在所有预期目标安全地接收发送到分发列表的消息。 如果任何目标发生故障，那么 MCA 将确定哪些目标发生故障。 然后，它可以为它们生成异常报告，并且可以尝试将消息再次发送到它们。

应答队列

您可以使用应答队列创建完整的远程队列处理循环。

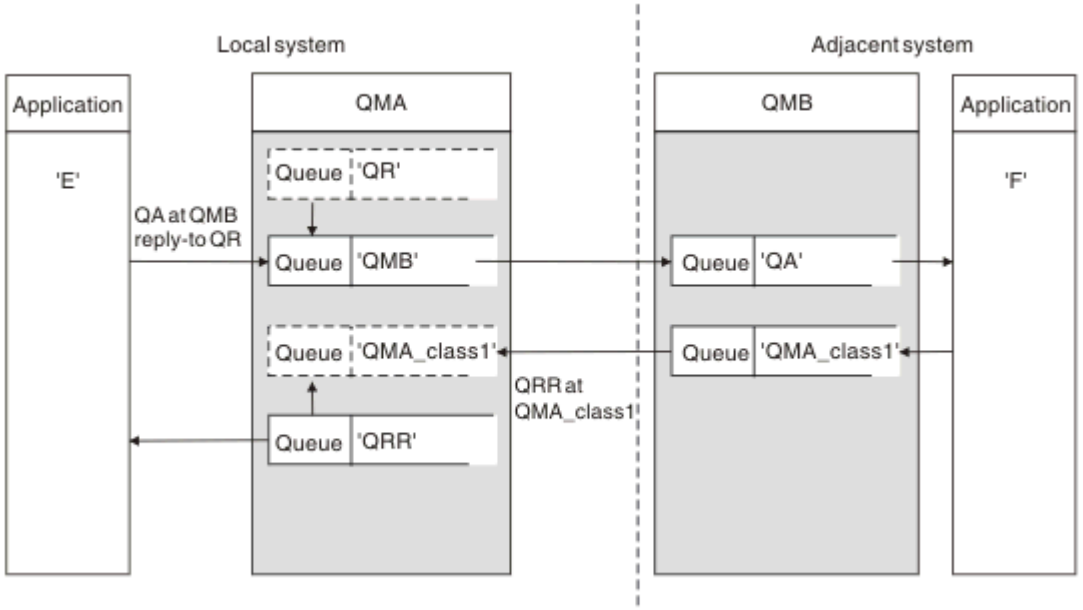


图 *15: PUT* 调用期间的应答队列名称替换

第 163 页的图 15 中显示了使用应答队列的完整远程队列处理循环。 此循环适用于分布式排队环境和集群环境。 详细信息如 第 169 页的表 21 中所示。

应用程序在 QMB 处打开 QA，并将消息放入该队列中。 将为消息提供应答队列名称 QR，而不指定队列管理器名称。 队列管理器 QMA 查找应答队列对象 QR，并从中抽取 QRR 的别名和队列管理器名称 QMA_class1。 这些名称将放入消息的应答字段中。

来自 QMB 的应用程序的应答消息将发送到位于 QMA_class1 的 QRR。 队列管理器使用队列管理器别名定义 QMA_class1 将消息流至自身，并将消息流至队列 QRR。

此场景描述了向应用程序提供工具以选择应答消息的服务类的方式。 该类由位于 QMB 的传输队列 QMA_class1 以及位于 QMA 的队列管理器别名定义 QMA_class1 实现。 通过这种方式，您可以更改应用程序的应答队列，以便在不涉及应用程序的情况下分隔流。 应用程序始终为此特定服务类选择 QR。 您可以使用应答队列定义 QR 来更改服务类。

您必须创建:

- 应答队列定义 QR
- 传输队列对象 QMB
- Channel_out 定义
- Channel_back 定义
- 队列管理器别名定义 QMA_class1
- 本地队列对象 QRR (如果不存在)

相邻系统上的补充管理员必须创建:

- 接收通道定义
- 传输队列对象 QMA_class1
- 关联的发送通道
- 本地队列对象 QA。

应用程序使用:

- 放入调用中的应答队列名称 QR

- 获取调用中的队列名称 QRR

通过这种方式，您可以根据需要更改服务等级，而不涉及应用程序。 您可以更改应答别名 "QR" 以及传输队列 "QMA_class1" 和队列管理器别名 "QMA_class1"。

如果在将消息放入队列时找不到应答别名对象，那么将在空白应答队列管理器名称字段中插入本地队列管理器名称。 应答队列名称保持不变。

名称解析限制

由于在放入原始消息时已对 "QMA" 处的应答队列执行名称解析，因此不允许在 "QMB" 处进行进一步的名称解析。 应答应用程序将消息与应答队列的物理名称放在一起。

应用程序必须知道它们用于应答队列的名称与要在其中找到返回消息的实际队列的名称不同。

例如，当为使用具有应答队列别名 "C1_alias" 和 "C2_alias" 的应用程序提供了两类服务时，应用程序会将这些名称用作消息放置调用中的应答队列名称。 但是，对于 "C1_alias" 和 "C2" (对于 "C2_alias") ，应用程序实际上期望消息显示在队列 "C1" 中。

但是，应用程序能够对应答别名队列进行查询调用，以自行检查它必须用于获取应答消息的实际队列的名称。

相关概念

第 154 页的『如何创建队列管理器和应答别名』
本主题说明了创建远程队列定义的三种方法。

第 164 页的『应答队列别名示例』
此示例说明如何使用应答别名为返回的消息选择其他路由 (传输队列)。 使用此设施需要与应用程序合作更改应答队列名称。

第 166 页的『示例的工作方式』
此示例的说明以及队列管理器如何使用应答队列别名。

第 166 页的『应答队列别名预评估』
从将消息放在远程队列上的应用程序到从别名应答队列中除去应答消息的同一应用程序的过程的预评估。

应答队列别名示例

此示例说明如何使用应答别名为返回的消息选择其他路由 (传输队列)。 使用此设施需要与应用程序合作更改应答队列名称。

如 第 164 页的图 16 中所示，返回路径必须可用于应答消息，包括传输队列，通道和队列管理器别名。

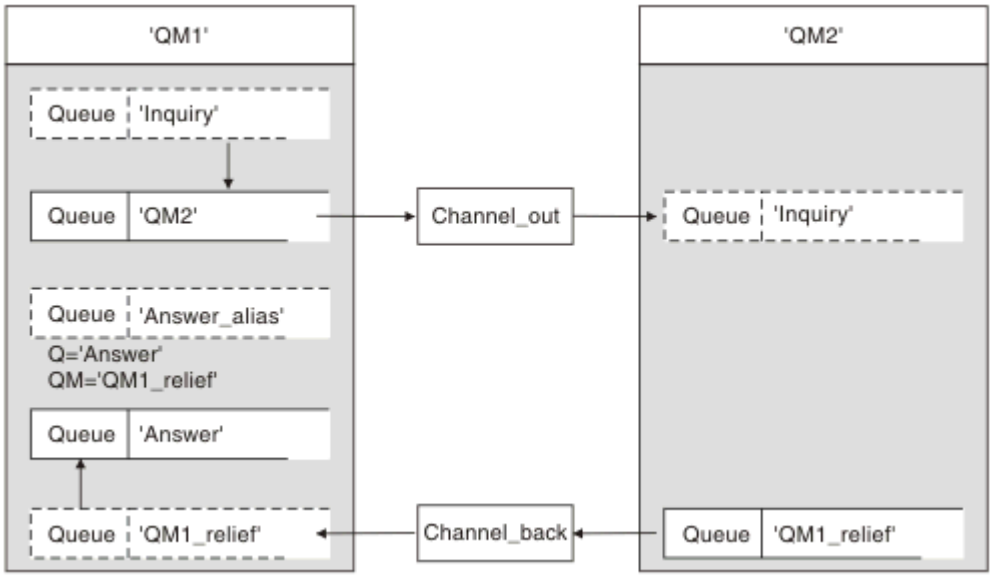


图 *16:* 应答队列别名示例

此示例适用于 "QM1" 上的请求者应用程序，这些应用程序将消息发送到 "QM2" 上的服务器应用程序。 服务器上的消息将通过使用传输队列 "QM1_relief" 的备用通道返回 (缺省返回通道将与传输队列 "QM1" 一起提供)。

应答队列别名是对名为 "Answer_alias" 的远程队列定义的特定使用。 QM1 上的应用程序在其放入队列 "查询" 的所有消息的应答字段中包含此名称 "Answer_alias"。

应答队列定义 "Answer_alias" 定义为 "在 QM1_relief 处回答"。 QM1 上的应用程序期望其应答显示在名为 "Answer" 的本地队列中。

QM2 上的服务器应用程序使用接收到的消息的应答字段来获取针对 QM1 上的请求者的应答消息的队列和队列管理器名称。

此示例中在 QM1 处使用的定义

QM1 上的 IBM MQ 系统管理员必须确保应答队列 "Answer" 与其他对象一起创建。 使用 "*" 标记的队列管理器别名的名称必须与应答队列别名定义中的队列管理器名称一致，也使用 "*" 标记。

Object	**定义**	
本地传输队列	QM2	
远程队列定义	对象名	查询
	远程队列管理器名称	QM2
	远程队列名称	查询
	传输队列的名称	QM2 (缺省值)
队列管理器别名	对象名	QM1_relief *
	队列管理器名称	QM1
	队列名称	（空白）
应答队列别名	对象名	答案别名
	远程队列管理器名称	QM1_relief *
	远程队列名称	答案

将定义放在 QM1 上

应用程序使用应答队列别名填充应答字段，并将队列管理器名称字段留空。

字段	**内容**
队列名称	查询
队列管理器名称	（空白）
应答队列名称	答案别名
应答队列管理器	（空白）

此示例中在 QM2 处使用的定义

QM2 上的 IBM MQ 系统管理员必须确保本地队列对于入局消息存在，并且正确命名的传输队列对于应答消息可用。

Object	**定义**
本地队列	查询
传输队列	QM1_relief

将定义放在 QM2 上

QM2 上的应用程序从原始消息中检索应答队列名称和队列管理器名称，并在将应答消息放入应答队列时使用这些名称。

字段	内容
队列名称	答案
队列管理器名称	QM1_relief

示例的工作方式

此示例的说明以及队列管理器如何使用应答队列别名。

在此示例中， QM1 上的请求者应用程序始终使用 "Answer_alias" 作为 put 调用的相关字段中的应答队列。他们始终从名为 "Answer" 的队列中检索其消息。

回复队列别名定义可供 QM1 系统管理员用于更改回复队列 "Answer" 的名称以及返回路径 "QM1_relief" 的名称。

更改队列名称 "Answer" 通常无用，因为 QM1 应用程序在此队列中期望其答案。 但是， QM1 系统管理员能够根据需要更改返回路径 (服务等级)。

队列管理器如何使用应答队列别名

当应用程序的 put 调用中包含的应答队列名称与应答队列别名相同，并且队列管理器部分为空白时，队列管理器 QM1 将从应答队列别名中检索定义。

队列管理器将 put 调用中的应答队列名称替换为定义中的队列名称。 它将 put 调用中的空白队列管理器名称替换为定义中的队列管理器名称。

这些名称与消息描述符中的消息一起携带。

表 *18:* 应答队列别名

字段名称	Put 调用	传输头
应答队列名称	答案别名	答案
应答队列管理器名称	（空白）	QM1_relief

应答队列别名预评估

从将消息放在远程队列上的应用程序到从别名应答队列中除去应答消息的同一应用程序的过程的预评估。

要完成此示例，请让我们查看过程。

1. 应用程序打开名为 "查询" 的队列，并将消息放入其中。 应用程序将消息描述符的应答字段设置为:

应答队列名称	答案别名
应答队列管理器名称	（空白）

2. 队列管理器 "QM1" 通过检查名为 "Answer_alias" 的远程队列定义来响应空白队列管理器名称。 如果找不到任何消息，那么队列管理器会将其自己的名称 "QM1" 放在消息描述符的 "应答队列管理器" 字段中。
3. 如果队列管理器找到名为 "Answer_alias" 的远程队列定义，那么它将从该定义中抽取队列名称和队列管理器名称 (队列名称 = "Answer" 和队列管理器名称 = "QM1_relief")。 然后将它们放入消息描述符的应答字段中。
4. 队列管理器 "QM1" 使用远程队列定义 "查询" 来确定预期目标队列位于队列管理器 "QM2" 上，并且消息放置在传输队列 "QM2" 上。 "QM2" 是发送给队列管理器 "QM2" 上的队列的消息的缺省传输队列名称。
5. 当队列管理器 "QM1" 将消息放入传输队列时，它会向消息添加传输头。 此头包含目标队列 "查询" 和目标队列管理器 "QM2" 的名称。
6. 消息到达队列管理器 "QM2" ，并放置在 "查询" 本地队列上。

7. 应用程序从此队列获取消息并处理消息。 应用程序准备应答消息，并将此应答消息放在来自原始消息的消息描述符的应答队列名称上:

应答队列名称	答案
应答队列管理器名称	QM1_relief

8. 队列管理器 "QM2" 执行 put 命令。 如果发现队列管理器名称 "QM1_relief" 是远程队列管理器，那么它会将消息放置在具有相同名称 "QM1_relief" 的传输队列上。 将为消息提供一个传输头，其中包含目标队列 "Answer" 和目标队列管理器 "QM1_relief" 的名称。
9. 消息将传输到队列管理器 "QM1"。 队列管理器可识别队列管理器名称 "QM1_relief" 是别名，从别名定义 "QM1_relief" 中抽取物理队列管理器名称 "QM1"。
10. 然后，队列管理器 "QM1" 将消息放在传输头 "Answer" 中包含的队列名称上。
11. 应用程序从队列 "Answer" 中抽取其应答消息。

联网注意事项

在分布式排队环境中，由于仅使用队列名称和队列管理器名称来寻址消息目标，因此某些规则适用。

1. 其中提供了队列管理器名称，并且该名称与本地队列管理器名称不同:
   - 必须提供具有相同名称的传输队列。 此传输队列必须是将消息移动到另一个队列管理器的消息通道的一部分，或者
   - 必须存在队列管理器别名定义才能将队列管理器名称解析为相同或其他队列管理器名称以及可选的传输队列，或者
   - 如果无法解析传输队列名称，并且已定义缺省传输队列，那么将使用缺省传输队列。
2. 在仅提供队列名称的情况下，必须在本地队列管理器上提供具有相同名称的任何类型的队列。 此队列可以是远程队列定义，解析为: 到相邻队列管理器的传输队列，队列管理器名称和可选传输队列。

要了解此方法在集群环境中的工作方式，请参阅 集群。

z/OS 如果队列管理器正在队列共享组 (QSG) 中运行，并且已启用组内排队 (IGQ)，那么可以使用 SYSTEM.QSG.TRANSMIT.QUEUE。 有关更多信息，请参阅 组内排队。

请考虑在分布式排队环境中消息通道将消息从一个队列管理器移动到另一个队列管理器的场景。

要移动的消息源自网络中的任何其他队列管理器，并且某些消息可能会到达具有未知队列管理器名称作为目标的消息。 例如，当队列管理器名称已更改或已从系统中除去时，可能会发生此问题。

通道程序在找不到这些消息的传输队列时识别此情况，并将这些消息放在未传递的消息 (死信) 队列上。 您负责查找这些消息并安排将其转发到正确的目标。 或者，将它们返回给发起方，在此可以确定发起方。

如果在原始消息中请求了报告消息，那么将在这些情况下生成异常报告。

名称解析约定

更改目标队列标识的名称解析 (即，逻辑名称到物理名称的更改)，仅出现一次，并且仅在发端队列管理器中出现。

仅当分隔和组合消息流时，才必须使用各种别名可能性的后续使用。

返回路由

消息可以包含队列和队列管理器的名称形式的返回地址。 此返回地址表单可以在分布式排队环境和集群环境中使用。

此地址通常由创建消息的应用程序指定。 可以由随后处理消息的任何应用程序 (包括用户出口应用程序) 对其进行修改。

无论此地址的来源如何，任何处理消息的应用程序都可以选择使用此地址将应答，状态或报告消息返回到发端应用程序。

这些响应消息的路由方式与原始消息的路由方式不同。 您需要注意，您创建到其他队列管理器的消息流需要相应的返回流。

物理名称冲突

目标应答队列名称已解析为原始队列管理器上的物理队列名称。 不能在响应队列管理器上再次解析此问题。

可能存在名称冲突问题，只能通过物理和逻辑队列名称的网络范围协议来防止这些问题。

管理队列名称转换

创建队列管理器别名定义或远程队列定义时，将对包含该名称的每条消息执行名称解析。 必须管理此情境。

此描述提供给与单个系统相关的应用程序设计者和通道规划者，该系统具有到相邻系统的消息通道。 它采用本地通道规划和控制视图。

当您创建队列管理器别名定义或远程队列定义时，将对携带该名称的每条消息执行名称解析，而不考虑消息的来源。 要监督此情况 (可能涉及队列管理器网络中的大量队列) ，请保留以下表:

- 与解析的队列名称，解析的队列管理器名称和解析的传输队列名称相关的源队列和源队列管理器的名称，具有解析方法
- 与以下相关的源队列的名称:
  - 已解析的目标队列名称
  - 已解析的目标队列管理器名称
  - 传输队列
  - 消息通道名称
  - 相邻系统名称
  - 应答队列名称

注: 在此上下文中使用术语 *source* 是指打开队列以放置消息时，应用程序或通道程序提供的队列名称或队列管理器名称。

第 168 页的表 19， 第 168 页的表 20 和 第 169 页的表 21 中显示了其中每个表的示例。

这些表中的名称派生自本节中的示例，而此表并非作为一个节点中队列名称解析的实际示例。

表 *19:* 队列管理器 *QMA* 上的队列名称解析

打开队列时指定的源队列	打开队列时指定的源队列管理器	已解析队列名称	已解析队列管理器名称	已解析的传输队列名称	解决方案类型
QA_norm	-	QA_norm	QMB	QMB	远程队列
（任何）	QMB	-	-	QMB	（无）
QA_norm	-	QA_norm	QMB	TX1	远程队列
QB	QMC	QB	QMD	QMB	队列管理器别名

表 *20:* 队列管理器 *QMB* 上的队列名称解析

打开队列时指定的源队列	打开队列时指定的源队列管理器	已解析队列名称	已解析队列管理器名称	已解析的传输队列名称	解决方案类型
QA_norm	-	QA_norm	QMB	-	（无）
QA_norm	QMB	QA_norm	QMB	-	（无）
QA_norm	QMB_PRIORITY	QA_norm	QMB	-	队列管理器别名
（任何）	QMC	（任何）	QMC	QMC	（无）

表 20: 队列管理器 *QMB* 上的队列名称解析 (继续)

打开队列时指定的源队列	打开队列时指定的源队列管理器	已解析队列名称	已解析队列管理器名称	已解析的传输队列名称	解决方案类型
（任何）	QMD_norm	（任何）	QMD_norm	TX1	队列管理器别名
（任何）	QMD_PRIORITY	（任何）	QMD_PRIORITY	QMD_fast	队列管理器别名
（任何）	QMC_small	（任何）	QMC_small	TX_small	队列管理器别名
（任何）	QMC_large	（任何）	QMC_large	TX_external	队列管理器别名
QB_small	QMC	QB_small	QMC	TX_small	远程队列
QB_large	QMC	QB_large	QMC	TX_large	远程队列
（任何）	QME	（任何）	QME	TX1	队列管理器别名
QA	QMC	QA	QMC	TX1	远程队列
QB	QMD	QB	QMD	TX1	远程队列

表 21: 队列管理器 *QMA* 上的应答队列名称转换

应用程序设计		应答别名定义	
本地 QMGR	**消息的队列名称**	**应答队列别名**	**重新定义为**
QMA	快速反应	QR	QRR 位于 QMA_class1

通道消息序列编号

通道使用序号来检查是否以从传输队列中获取消息的相同顺序传递消息。

只有当信息开始在信道上流动时，才会首次检测到信道序列号不匹配。 如果出现不匹配，则意味着通道任一侧的持续同步数据丢失；例如，灾难恢复（DR）配置，或当通道有疑问时，批处理结束被中断。

重置或忽略序号不匹配，请参阅 *qm.ini* 文件的 *Channels* 节中的 **IgnoreSeqNumberMismatch**，不存在丢失或复制一批消息的风险，并且不会重置通道的不确定状态。

可以使用 DISPLAY CHSTATUS 显示此信息。 对于批处理中传输的最后一条消息，序号和称为 LUWID 的标识存储在持久存储器中。 这些值在通道启动期间使用，以确保链路的两端都同意已成功传输的消息。

顺序检索消息

如果应用程序将一连串报文放入同一个目标队列，在满足以下条件的情况下，单个应用程序可通过一连串 MQGET 操作依次检索这些报文：

- 所有放入请求都是从同一应用程序完成的。
- 所有放置请求都来自同一工作单元，或者所有放置请求都是在工作单元外部发出的。
- 所有消息都具有相同的优先级。
- 所有消息都具有相同的持久性。
- 对于远程排队，配置使得从发出 put 请求的应用程序到目标队列管理器和目标队列只能有一条路径，通过其队列管理器，通过相互通信。
- 消息不会放入死信队列 (例如，如果队列暂时已满)。
- 获取消息的应用程序不会故意更改检索顺序，例如，通过指定特定 *MsgId* 或 *CorrelId* 或使用消息优先级。
- 只有一个应用程序正在执行获取操作以从目标队列检索消息。 如果有多个应用程序，那么这些应用程序必须设计为获取发送应用程序放入的每个序列中的所有消息。

注: 来自其他任务和工作单元的消息可能与序列交织在一起，即使序列是从单个工作单元中放入的。

如果无法满足这些条件，并且目标队列上的消息顺序很重要，那么可以将应用程序编码为使用其自己的消息序号作为消息的一部分，以确保消息的顺序。

快速非持久消息的检索顺序

快速通道上的非持久消息可能会覆盖同一通道上的持久消息，因此无法按顺序到达。 接收 MCA 会立即将非持久消息放在目标队列上，并使其可视。 直到下一个同步点才会显示持久消息。

回送测试

回送测试 是非 z/OS 平台上的一种技术，允许您测试通信链路，而无需实际链接到另一台机器。

您可以在两个队列管理器之间设置一个连接，就好像它们位于不同的机器上一样，但您可以通过循环返回到同一机器上的另一个进程来测试该连接。 此技术意味着您可以在不需要活动网络的情况下测试通信代码。

您执行此操作的方式取决于您正在使用的产品和协议。

在 Windows 系统上，可以使用 "loopback" 适配器。

有关更多信息，请参阅您正在使用的产品的文档。

路由跟踪和活动记录

您可以通过两种方式确认消息通过一系列队列管理器所采用的路由。

您可以使用 IBM MQ 通过控制命令提供的显示路由应用程序 **dspmqrte**，也可以使用活动记录。 监控参考中描述了这两个主题。

分布式队列管理简介

分布式队列管理 (DQM) 用于定义和控制队列管理器之间的通信。

分布式队列管理:

- 使您能够定义和控制队列管理器之间的通信通道
- 为您提供消息通道服务，以将消息从一种称为传输队列的 本地队列类型移至本地系统上的通信链路，并从通信链路移至目标队列管理器上的本地队列
- 使用面板，命令和程序为您提供用于监视通道操作和诊断问题的工具

通道定义使通道名称与传输队列，通信链路标识和通道属性相关联。 通道定义在不同平台上以不同方式实现。 消息发送和接收由称为 消息通道代理程序 (MCA) 的程序控制，这些程序使用通道定义来启动和控制通信。

MCA 则由 DQM 本身控制。 该结构依赖于平台，但通常包括侦听器和触发器监视器以及操作员命令和面板。

消息通道 是用于将消息从一个队列管理器移动到另一个队列管理器的单向管道。 因此，消息通道具有两个端点，由一对 MCA 表示。 每个端点都有其消息通道结束的定义。 例如，一端定义发送方，另一端定义接收方。

有关如何定义通道的详细信息，请参阅:

- ALW 第 197 页的『监视和控制 AIX, Linux, and Windows 上的通道』
- z/OS 第 800 页的『监视和控制 z/OS 上的通道』
- IBM i 第 216 页的『监视和控制 IBM i 上的通道』

有关消息通道规划示例，请参阅:

- ALW AIX, Linux, and Windows 的消息通道规划示例
- IBM i IBM i 的消息通道规划示例
- z/OS z/OS 的消息通道规划示例

- z/OS 使用队列共享组的 z/OS 的消息通道规划示例

有关通道出口的信息，请参阅 消息传递通道的通道出口程序。

相关概念

第 171 页的『消息发送和接收』
下图显示了分布式队列管理模型，详细描述了传输消息时实体之间的关系。 它还显示用于控制的流。

第 178 页的『通道控制功能』
通道控制功能为您提供用于定义，监视和控制通道的工具。

第 189 页的『无法传递消息时会发生什么情况?』
当无法传递消息时， MCA 可以通过多种方式对其进行处理。 它可以重试，可以返回到发送方，也可以将其放在死信队列上。

第 193 页的『初始化和配置文件』
通道初始化数据的处理取决于 IBM MQ 平台。

第 195 页的『消息的数据转换』
在不同队列管理器上的队列之间发送 IBM MQ 消息时，可能需要数据转换。

第 195 页的『编写您自己的消息通道代理程序』
IBM MQ 允许您编写自己的消息通道代理程序 (MCA) 程序或从独立软件供应商安装程序。

第 195 页的『分布式队列管理要考虑的其他事项』
准备 IBM MQ 以进行分布式队列管理时要考虑的其他主题。 本主题涵盖未交付的消息队列，正在使用的队列，系统扩展和用户出口程序以及作为可信应用程序运行的通道和侦听器。

相关参考

示例配置信息

消息发送和接收

下图显示了分布式队列管理模型，详细描述了传输消息时实体之间的关系。 它还显示用于控制的流。

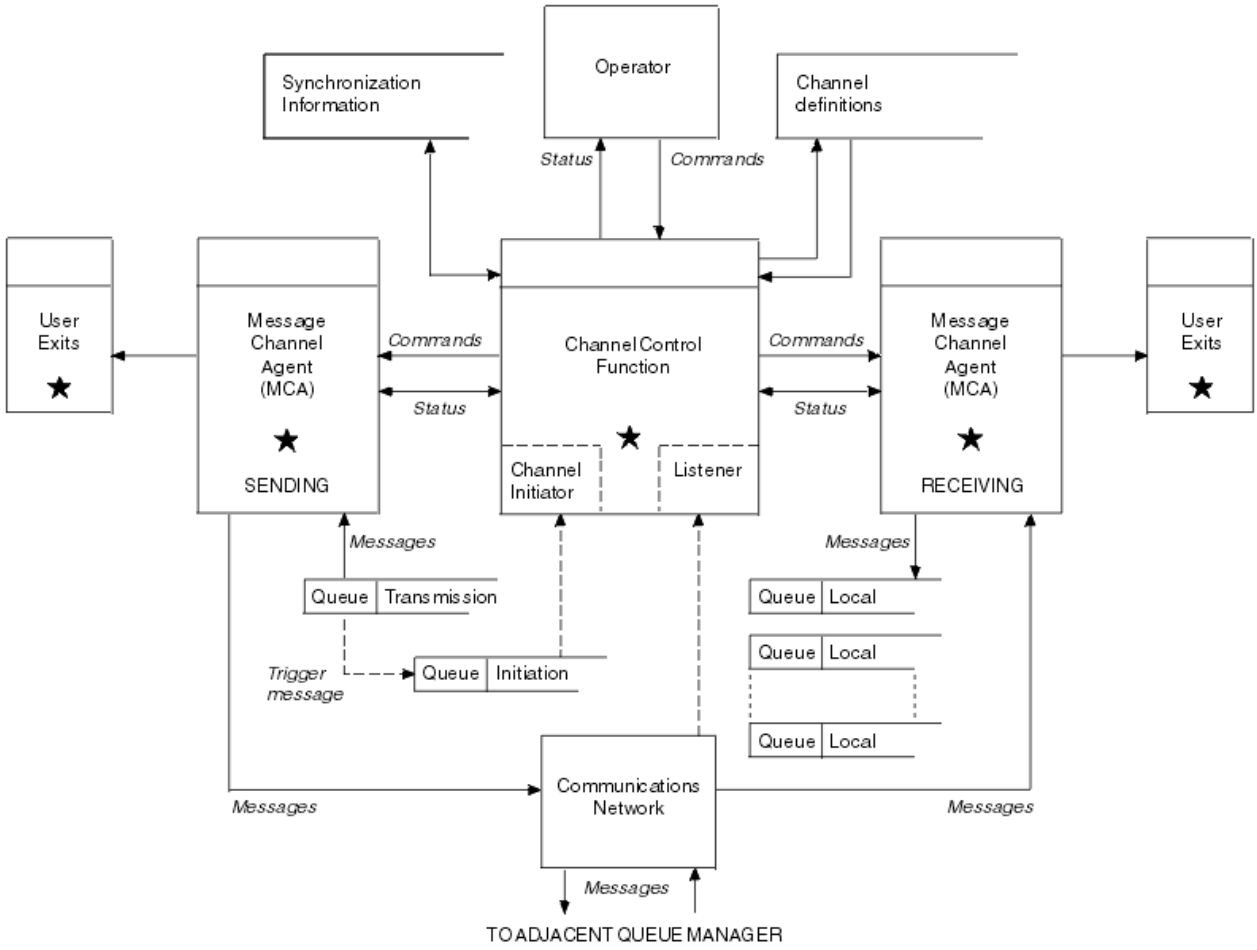


图 *17:* 分布式队列管理模型

注:

1. 每个通道有一个 MCA，具体取决于平台。 对于特定队列管理器，可能有一个或多个通道控制功能。
2. MCA 和通道控制功能的实现高度依赖于平台。 它们可以是程序，进程或线程，它们可以是单个实体，也可以是由多个独立或链接部分组成的多个实体。
3. 所有标记为星型的组件都可以使用 MQI。

通道参数

MCA 通过以下几种方式之一接收其参数:

- 如果由命令启动，那么将在数据区中传递通道名称。 然后， MCA 直接读取通道定义以获取其属性。
- 对于发送方，在某些情况下，可以通过队列管理器触发器自动启动 MCA。 将从触发器进程定义中检索通道名称 (如果适用)，并将其传递到 MCA。 其余处理与先前描述的相同。 仅当服务器通道是标准通道时，才必须将其设置为触发，即，它们指定要连接到的 CONNAME。
- 如果由发送方，服务器，请求者或客户机连接远程启动，那么将在来自伙伴消息通道代理程序的初始数据中传递通道名称。 MCA 直接读取通道定义以获取其属性。

通道定义中未定义的某些属性也可协商:

拆分消息
如果一端不支持拆分消息，那么不会发送拆分消息。

转换功能
如果一端无法在需要时执行必要的代码页转换或数字编码转换，那么另一端必须对其进行处理。 如果这两个端都不支持，那么在需要时，通道无法启动。

分发列表支持
如果一端不支持分发列表，那么合作伙伴 MCA 会在其传输队列中设置一个标志，以便它知道拦截要用于多个目标的消息。

通道状态和序号

消息通道代理程序保留每个通道的当前序号和逻辑工作单元号以及通道的常规状态的记录。 某些平台允许您显示此状态信息以帮助您控制通道。

如何将消息发送到另一个队列管理器

本节描述了在队列管理器之间发送消息的最简单方法，包括所需的先决条件和授权。 还可以使用其他方法将消息发送到远程队列管理器。

在将消息从一个队列管理器发送到另一个队列管理器之前，需要执行以下步骤:

1. 检查所选通信协议是否可用。
2. 启动队列管理器。
3. 启动通道启动程序。
4. 启动侦听器。

您还需要具有正确的 IBM MQ 安全权限才能创建所需的对象。

要将消息从一个队列管理器发送到另一个队列管理器:

- 在源队列管理器上定义以下对象:
  - 发送发通道
  - 远程队列定义
  - 启动队列 (z/OS 在 z/OS 上必需，否则 可选)
  - 传输队列
  - 死信队列
- 在目标队列管理器上定义以下对象:
  - 接收方通道
  - 目标队列
  - 死信队列

您可以使用多种不同的方法来定义这些对象，具体取决于您的 IBM MQ 平台:

- 在所有平台上，可以使用 MQSC 命令 中描述的 IBM MQ 脚本命令 (MQSC)， 自动化管理任务或 IBM MQ Explorer 中描述的可编程命令格式 (PCF) 命令。
- z/OS 在 z/OS 上，还可以使用 管理 IBM MQ for z/OS 中描述的 "操作" 和 "控制" 面板。
- IBM i 在 IBM i 上，您还可以使用面板界面。

请参阅以下子主题以获取有关创建组件以将消息发送到另一个队列管理器的更多信息:

相关概念

第 153 页的『IBM MQ 分布式排队技术』
本部分中的子主题描述了在规划通道时使用的方法。 这些子主题描述了帮助您规划如何将队列管理器连接在一起以及管理应用程序之间的消息流的方法。

第 170 页的『分布式队列管理简介』
分布式队列管理 (DQM) 用于定义和控制队列管理器之间的通信。

第 191 页的『触发通道』
IBM MQ 提供了用于在满足队列上的特定条件时自动启动应用程序的工具。 此工具称为触发。

第 189 页的『消息的安全性』

除了 IBM MQ 的典型恢复功能外，分布式队列管理还确保通过使用消息通道两端之间协调的同步点过程来正确传递消息。 如果此过程检测到错误，那么它会关闭通道，以便您可以调查问题，并将消息安全地保留在传输队列中，直到通道重新启动为止。

相关任务

第 7 页的『在 Multiplatforms 版上创建队列管理器』
必须先创建并启动至少一个队列管理器及其关联对象，然后才能使用消息和队列。 队列管理器管理与其关联的资源，特别是其拥有的队列。 它为消息排队接口 (MQI) 调用和命令的应用程序提供排队服务，以创建，修改，显示和删除 IBM MQ 对象。

第 197 页的『监视和控制 AIX, Linux, and Windows 上的通道』
对于 DQM ，您需要创建，监视和控制到远程队列管理器的通道。 您可以使用命令，程序， IBM MQ Explorer，通道定义的文件以及同步信息的存储区来控制通道。

第 216 页的『监视和控制 IBM i 上的通道』
使用 DQM 命令和面板来创建，监视和控制到远程队列管理器的通道。 每个队列管理器都有一个 DQM 程序，用于控制与兼容远程队列管理器的互连。

第 13 页的『配置客户机与服务器之间的连接』
要配置 IBM MQ MQI clients 与服务器之间的通信链路，请决定通信协议，定义链路两端的连接，启动侦听器以及定义通道。

第 235 页的『配置队列管理器集群』
集群提供了一种用于以简化初始配置和持续管理的方式互连队列管理器的机制。 您可以定义集群组件，以及创建和管理集群。

第 797 页的『在 z/OS 上设置与其他队列管理器的通信』
本部分描述了在开始使用分布式排队之前需要进行的 IBM MQ for z/OS 准备工作。

定义通道

要将消息从一个队列管理器发送到另一个队列管理器，必须定义两个通道。 必须在源队列管理器上定义一个通道，在目标队列管理器上定义一个通道。

在源队列管理器上

定义通道类型为 SENDER 的通道。 您需要指定以下内容:

- 要使用的传输队列的名称 (XMITQ 属性)。
- 伙伴系统的连接名称 (CONNAME 属性)。
- 您正在使用的通信协议的名称 (TRPTYPE 属性)。 在 IBM MQ for z/OS 上，协议必须是 TCP 或 LU6.2。在其他平台上，您不必指定此项。 您可以保留该值以从缺省通道定义中选取值。

通道属性中提供了所有通道属性的详细信息。

在目标队列管理器上

定义通道类型为 RECEIVER 且与发送方通道同名的通道。

指定正在使用的通信协议的名称 (TRPTYPE 属性)。 在 IBM MQ for z/OS 上，协议必须是 TCP 或 LU6.2。在其他平台上，您不必指定此项。 您可以保留该值以从缺省通道定义中选取值。

接收方通道定义可以是通用的。 这意味着如果您有多个队列管理器与同一接收方进行通信，那么发送通道都可以为接收方指定相同的名称，并且一个接收方定义将全部应用于这些队列管理器。

定义通道后，可以使用 PING CHANNEL 命令对其进行测试。 此命令将特殊消息从发送方通道发送到接收方通道，并检查是否返回了该消息。

注: 响应消息通道代理程序将忽略 TRPTYPE 参数的值。 例如，发送方通道定义上 TCP 的 TRPTYPE 成功以接收方通道定义上的 TRPTYPE LU62 作为伙伴启动。

定义队列

要将消息从一个队列管理器发送到另一个队列管理器，必须最多定义 6 个队列。 必须在源队列管理器上最多定义四个队列，在目标队列管理器上最多定义两个队列。

在源队列管理器上

- 远程队列定义

在此定义中，指定以下内容:

远程队列管理器名称
目标队列管理器的名称。

远程队列名称
目标队列管理器上的目标队列的名称。

传输队列的名称
传输队列的名称。 您不必指定此传输队列名称。 如果不使用，那么将使用与目标队列管理器同名的传输队列。 如果这不存在，那么将使用缺省传输队列。 建议您为传输队列提供与目标队列管理器相同的名称，以便缺省情况下找到该队列。

- 启动队列定义

  z/OS 此项为必填。 必须使用名为 SYSTEM.CHANNEL.INITQ。

  Multi 这是可选操作。 请考虑命名启动队列 SYSTEM.CHANNEL.INITQ。

- 传输队列定义

  将 USAGE 属性设置为 XMITQ 的本地队列。 IBM i 如果您正在使用 IBM MQ for IBM i 本机接口，那么 USAGE 属性为 *TMQ。

- 死信队列定义

  定义可将未传递的消息写入的死信队列。

在目标队列管理器上

- 本地队列定义

  目标队列。 此队列的名称必须与源队列管理器上远程队列定义的远程队列名称字段中指定的名称相同。

- 死信队列定义

  定义可将未传递的消息写入的死信队列。

相关概念

第 175 页的『创建传输队列』
在可以启动通道 (请求者通道除外) 之前，必须按本节中所述定义传输队列。 必须在通道定义中指定传输队列。

第 175 页的『在 IBM i 上创建传输队列』
您可以使用 "创建 MQM 队列" 面板在 IBM i 平台上创建传输队列。

创建传输队列
在可以启动通道 (请求者通道除外) 之前，必须按本节中所述定义传输队列。 必须在通道定义中指定传输队列。

为每个发送消息通道定义一个将 USAGE 属性设置为 XMITQ 的本地队列。 如果要在远程队列定义中使用特定传输队列，请按所示创建远程队列。

要创建传输队列，请使用 IBM MQ 命令 (MQSC) ，如以下示例中所示:

创建传输队列示例

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

创建远程队列示例

```
DEFINE QREMOTE(PAYROLL) DESCR('Remote queue for QM2') +
XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

请考虑将传输队列命名为远程系统上的队列管理器名称，如示例中所示。

在 *IBM i* 上创建传输队列
您可以使用 "创建 MQM 队列" 面板在 IBM i 平台上创建传输队列。

必须为每个发送消息通道定义一个 "使用情况" 字段属性设置为 *TMQ 的本地队列。

如果要使用远程队列定义，请使用同一命令来创建类型为 *RMT 的队列，并使用 *NORMAL。

要创建传输队列，请从命令行使用 CRTMQMQ 命令向您显示第一个队列创建面板; 请参阅 第 176 页的图 18。

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Queue name . . . . . .

Queue type . . . . . .  ____     *ALS, *LCL, *MDL, *RMT

Message Queue Manager name . . .  *DFT_____________________________
_____









Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
+
```

图 *18:* 创建队列 *(1)*

输入队列的名称，并指定要创建的队列类型: 本地，远程或别名。 对于传输队列，请在此面板上指定 "本地" (*LCL) ，然后按 Enter 键。

将显示 "创建 MQM 队列" 面板的第二页; 请参阅 第 176 页的图 19。

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Queue name . . . . . > HURS.2.HURS.PRIORIT

Queue type . . . . . > *LCL     *ALS, *LCL, *MDL, *RMT
Message Queue Manager name . . .  *DFT
Replace . . . . . .  *NO      *NO, *YES
Text 'description' . . . . . .  '                    '
Put enabled . . . . . .  *YES     *SYSDFTQ, *NO, *YES
Default message priority . . . .  0       0-9, *SYSDFTQ
Default message persistence . .  *NO      *SYSDFTQ, *NO, *YES
Process name . . . . . .  '                    '
Triggering enabled . . . . . .  *NO      *SYSDFTQ, *NO, *YES
Get enabled . . . . . .  *YES     *SYSDFTQ, *NO, *YES
Sharing enabled . . . . . .  *YES     *SYSDFTQ, *NO, *YES



More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

图 *19:* 创建队列 *(2)*

更改显示的任何缺省值。 向下按页面以滚动到下一个屏幕; 请参阅 第 177 页的图 20。

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Default share option . . . . . *YES     *SYSDFTQ, *NO, *YES
Message delivery sequence . . .  *PTY     *SYSDFTQ, *PTY, *FIFO
Harden backout count . . . . . *NO      *SYSDFTQ, *NO, *YES
Trigger type . . . . . *FIRST    *SYSDFTQ, *FIRST, *ALL...
Trigger depth . . . . . 1       1-999999999, *SYSDFTQ
Trigger message priority . . . . 0       0-9, *SYSDFTQ
Trigger data . . . . . '                   '
Retention interval . . . . . 999999999   0-999999999, *SYSDFTQ
Maximum queue depth . . . . . 5000     1-24000, *SYSDFTQ
Maximum message length . . . . . 4194304    0-4194304, *SYSDFTQ
Backout threshold . . . . . 0       0-999999999, *SYSDFTQ
Backout requeue queue . . . . . '                  '
Initiation queue . . . . . '                  '



More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

图 *20:* 创建队列 *(3)*

在此面板的 "使用情况" 字段中输入 *TMQ 作为传输队列，并更改其他字段中显示的任何缺省值。

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Usage . . . . . *TMQ     *SYSDFTQ, *NORMAL, *TMQ
Queue depth high threshold . . .  80      0-100, *SYSDFTQ
Queue depth low threshold . . .  20      0-100, *SYSDFTQ
Queue full events enabled . . .  *YES     *SYSDFTQ, *NO, *YES
Queue high events enabled . . .  *YES     *SYSDFTQ, *NO, *YES
Queue low events enabled . . . .  *YES     *SYSDFTQ, *NO, *YES
Service interval . . . . . 999999999   0-999999999, *SYSDFTQ
Service interval events . . . .  *NONE     *SYSDFTQ, *HIGH, *OK, *NONE
Distribution list support . . .  *NO      *SYSDFTQ, *NO, *YES
Cluster Name . . . . . *SYSDFTQ
Cluster Name List . . . . . *SYSDFTQ
Default Binding . . . . . *SYSDFTQ   *SYSDFTQ, *OPEN, *NOTFIXED




Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

图 *21:* 创建队列 *(4)*

当您确信字段包含正确的数据时，请按 Enter 键以创建队列。

启动通道

将消息放在源队列管理器上定义的远程队列上时，这些消息将存储在传输队列上，直到通道启动为止。 当通道已启动时，消息将传递到远程队列管理器上的目标队列。

使用 START CHANNEL 命令在发送队列管理器上启动通道。 启动发送通道时，接收通道将自动启动 (由侦听器启动) ，并将消息发送到目标队列。 必须运行消息通道的两端才能传输消息。

由于通道的两端位于不同的队列管理器上，因此可以使用不同的属性对它们进行定义。 要解决任何差异，在通道启动时，两端之间存在初始数据协商。 通常，通道的两端使用需要较少资源的属性进行操作。 这使更大的系统能够在消息通道的另一端容纳较小的系统的较小资源。

发送 MCA 先拆分大型消息，然后再通过通道发送这些消息。 它们在远程队列管理器上重新组合。 这一点对用户来说并不明显。

MCA 可以使用多个线程来传输消息。 此过程 (称为 管道传送) 使 MCA 能够以更少的等待状态更高效地传输消息。 管道传送提高了通道性能。

通道控制功能

通道控制功能为您提供用于定义，监视和控制通道的工具。

通过面板，程序或从命令行向通道控制功能发出命令。 面板接口还显示通道状态和通道定义数据。 您可以使用可编程命令格式或 第 197 页的『监视和控制 AIX, Linux, and Windows 上的通道』中详细描述的那些 IBM MQ 命令 (MQSC) 和控制命令。

这些命令属于以下组:

- 通道管理
- 通道控制
- 通道状态监视

通道管理命令处理通道的定义。 它们使您能够:

- 创建通道定义
- 复制通道定义
- 变更通道定义
- 删除通道定义

通道控制命令管理通道的操作。 它们使您能够:

- 启动通道
- 停止通道
- 与合作伙伴重新同步 (在某些实施中)
- 重置消息序号
- 解析不确定的消息批处理
- Ping; 通过通道发送测试通信

通道监视显示通道的状态，例如:

- 当前通道设置
- 通道是处于活动状态还是处于不活动状态
- 通道是否以同步状态终止

相关概念

在何处查找信息以帮助确定问题

准备通道

在尝试启动消息通道或 MQI 通道之前，必须准备该通道。 必须确保本地和远程通道定义的所有属性都正确且兼容。

通道属性 描述通道定义和属性。

虽然您设置了显式通道定义，但在通道启动时执行的通道协商可能会覆盖所定义的一个或其他值。 这种行为是正常的，对用户来说并不明显，并且是以这种方式安排的，这样否则不兼容的定义可以一起工作。

接收器和服务器连接通道的自动定义

在除 z/OS 以外的所有平台上的 IBM MQ 中，如果没有相应的通道定义，那么对于启用了自动定义的接收方或服务器连接通道，将自动创建定义。 定义是使用以下命令创建的:

1. 相应的模型通道定义 SYSTEM.AUTO.RECEIVER 或 SYSTEM.AUTO.SVRCONN。 自动定义的模型通道定义与系统缺省值 SYSTEM.DEF.RECEIVER 和 SYSTEM.DEF.SVRCONN，描述字段除外， "自动定义" 后跟 49 个空格。 系统管理员可以选择更改提供的模型通道定义的任何部分。
2. 来自合作伙伴系统的信息。 来自合作伙伴的值用于通道名称和序号合并值。
3. 通道出口程序，可用于更改由自动定义创建的值。 请参阅 通道自动定义出口程序 (Channel auto-definition exit program)。

然后，将检查描述以确定它是否已被自动定义出口改变，或者因为模型定义已更改。 如果前 44 个字符仍然是 "自动定义"，后跟 29 个空格，那么将添加队列管理器名称。 如果最后 20 个字符仍为空白，那么将添加本地时间和日期。

当定义已创建并存储时，通道启动将继续进行，就像该定义一直存在一样。 批处理大小，传输大小和消息大小将与合作伙伴协商。

定义其他对象

在可以启动消息通道之前，必须在其队列管理器中定义 (或启用自动定义) 两端。 它要服务的传输队列必须在发送端定义到队列管理器。 通信链路必须已定义且可用。 您可能需要准备其他 IBM MQ 对象，例如远程队列定义，队列管理器别名定义和应答队列别名定义，以实现 第 152 页的『配置分布式队列』中描述的方案。

有关定义 MQI 通道的信息，请参阅 第 25 页的『定义 MQI 通道』。

每个传输队列有多个消息通道

可以为每个传输队列定义多个通道，但其中只有一个通道可以在任何一个时间处于活动状态。 请考虑此选项，以在队列管理器之间提供备用路由，从而进行流量均衡和链路故障纠正操作。 如果使用传输队列的前一个通道已终止，并且在发送端留下一批不确定的消息，那么另一个通道无法使用该传输队列。 有关更多信息，请参阅 第 188 页的『处理不确定通道』。

启动通道

可能导致通道以四种方式之一开始传输消息。 可以是:

- 由操作员 (非接收方，集群接收方或服务器连接通道) 启动。
- 从传输队列触发。 此方法仅适用于发送方通道和标准服务器通道 (那些指定 CONNAME 的通道)。 您必须准备必要的对象以触发通道。
- 从应用程序 (不是接收方，集群接收方或服务器连接通道) 启动。
- 由发送方，集群发送方，请求者，服务器或客户机连接通道从网络远程启动。 以此方式启动接收方，集群接收方以及可能的服务器和请求者通道传输; 服务器连接通道也是如此。 通道本身必须已启动 (即，已启用)。

注: 由于通道已 "启动"，因此不一定要传输消息。 相反，可以 "启用" 以在发生先前描述的四个事件中的一个事件时开始传输。 启用和禁用通道是使用 START 和 STOP 操作员命令实现的。

通道状态

一个通道可以随时处于许多状态中的一种状态。 某些州也有子州。 从给定状态，通道可以进入其他状态。

第 180 页的图 22 显示了所有可能的通道状态的层次结构以及应用于每个通道状态的子状态。

第 181 页的图 23 显示了通道状态之间的链接。 这些链接适用于所有类型的消息通道和服务器连接通道。

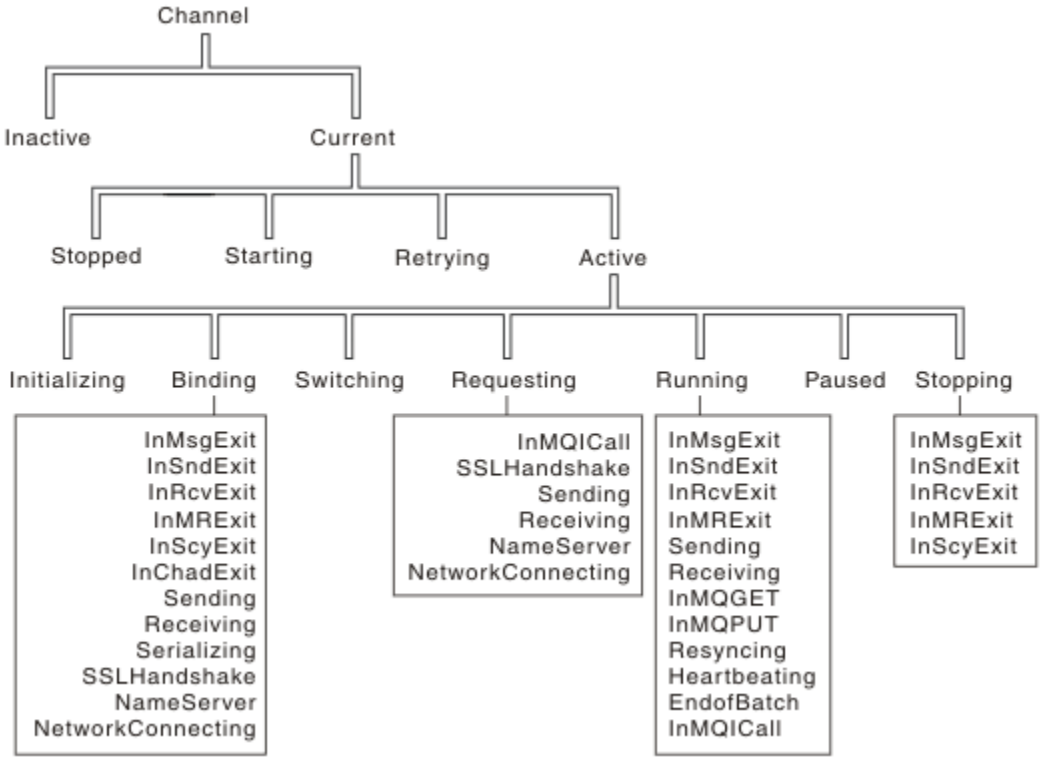


图 *22:* 通道状态和子状态

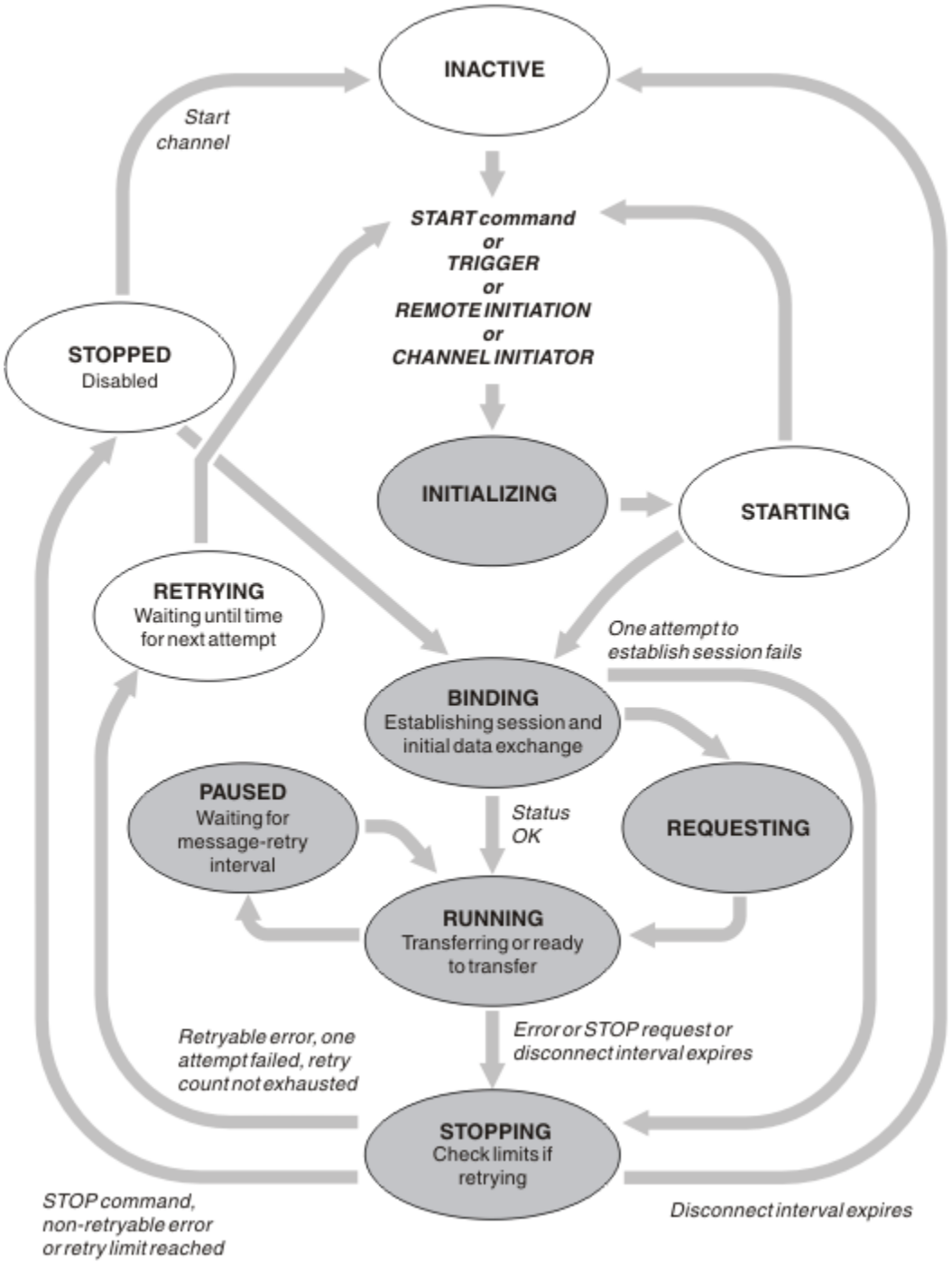


图 *23:* 通道状态之间的流

当前和活动

如果通道处于非不活动状态，那么该通道为 *current* 。 当前通道处于 活动 状态，除非它处于 "正在重试"，"已停止" 或 "正在启动" 状态。 通道处于活动状态时，它正在使用资源，并且进程或线程正在运行。 在 第 181 页的图 23 中突出显示了活动通道的七种可能状态 ("正在初始化"， "正在绑定"， "正在切换"， "正在请求"， "正在运行"， "已暂停" 或 "正在停止")。

活动通道还可以显示子状态，提供有关该通道所执行的操作的更多详细信息。 每个状态的子状态都显示在 第 180 页的图 22 中。

当前和活动

如果通道处于非不活动状态，那么该通道为 “current” 。 当前通道处于 “活动” 状态，除非它处于 "正在重试" ， "已停止" 或 "正在启动" 状态。

如果通道处于 "活动" 状态，那么可能还会显示子状态，以提供有关该通道所执行的操作的更多详细信息。

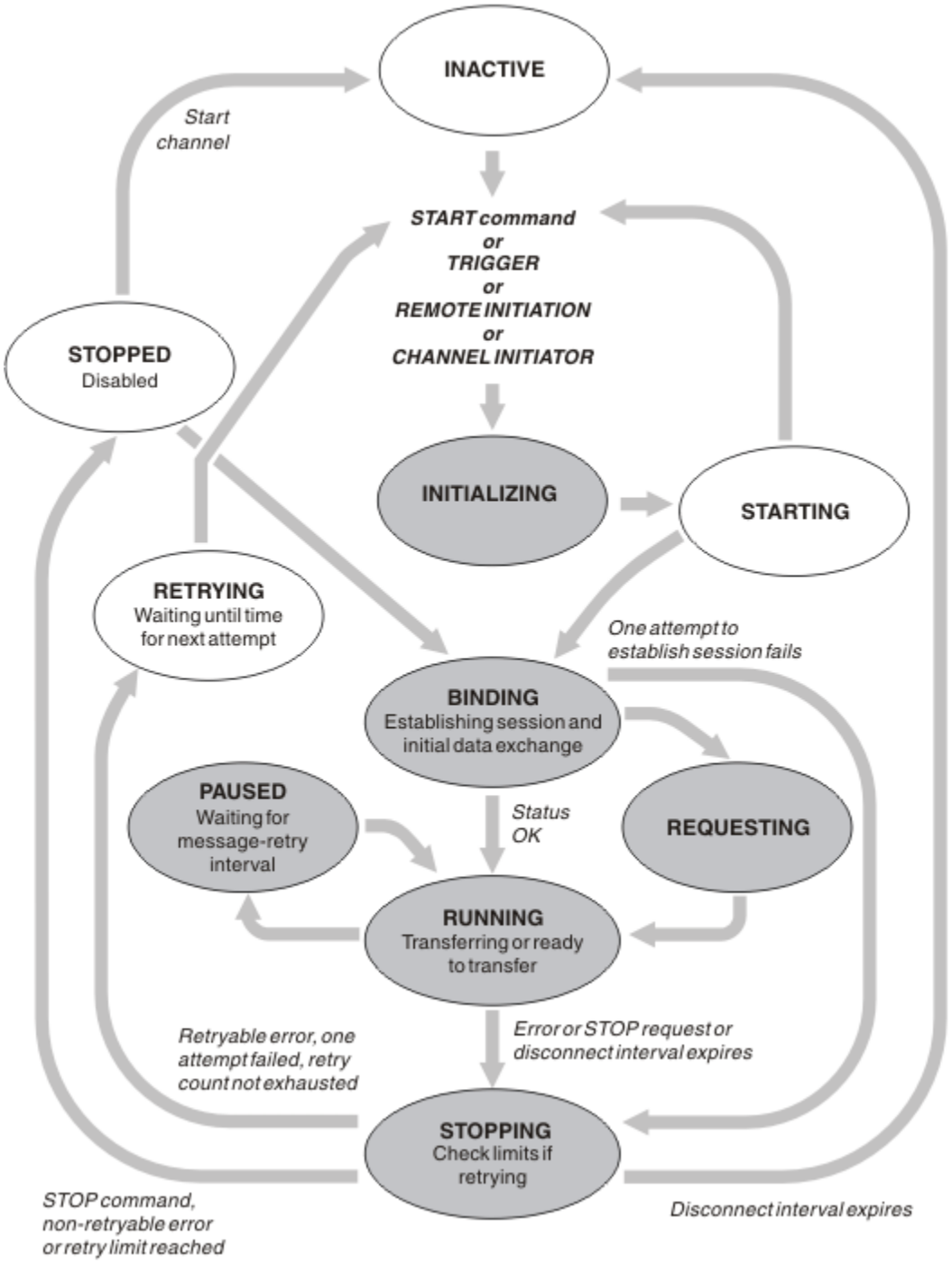


图 *24:* 通道状态之间的流

注:

1. When a channel is in one of the six states highlighted in 第 182 页的图 24 (INITIALIZING, BINDING, REQUESTING, RUNNING, PAUSED, or STOPPING), it is consuming resource and a process or thread is running; the channel is 活动.
2. 当通道处于 STOPPED 状态时，会话可能处于活动状态，因为下一个状态未知。

指定当前通道的最大数量

您可以指定一次可以为当前的最大通道数。 此数字是在通道状态表中具有条目的通道数，包括正在重试的通道数和已停止的通道数。 为您的平台指定以下内容:

- z/OS 使用 ALTER QMGR MAXCHL 命令。
- IBM i 编辑队列管理器初始化文件。
- Linux AIX 编辑队列管理器配置文件。
- 使用 IBM MQ Explorer。

有关使用初始化或配置文件设置的值的更多信息，请参阅 分布式排队的配置文件节。 有关指定最大通道数的更多信息，请参阅以下主题:

- ALW 管理 IBM MQ。
- IBM i 管理 IBM MQ for IBM i。
- z/OS 管理 IBM MQ for z/OS。

注:

1. 此数字中包含服务器连接通道。
2. 通道必须是当前通道，然后才能变为活动通道。 如果通道已启动，但无法成为当前通道，那么启动将失败。

指定最大活动通道数

您还可以指定最大活动通道数，以防止系统被许多启动通道重载。 如果使用此方法，请将断开连接时间间隔属性设置为较低的值，以允许等待通道在其他通道终止时立即启动。

每当正在重试的通道尝试与其伙伴建立连接时，它都必须成为活动通道。 如果尝试失败，那么它将保持当前通道处于不活动状态，直到下次尝试为止。 通道重试的次数和频率由重试计数和重试时间间隔通道属性确定。 这些属性都有短值和长值。 请参阅 通道属性 以获取更多信息。

当通道必须成为活动通道 (因为已发出 START 命令，或由于已触发，或由于是另一次重试尝试的时间)，但由于活动通道数已达到最大值而无法执行此操作时，通道将等待另一个通道实例释放其中一个活动插槽以停止活动。 但是，如果通道由于正在远程启动而正在启动，并且当时没有可用的活动插槽，那么将拒绝远程启动。

当除请求者通道以外的通道尝试变为活动状态时，它将进入 "正在启动" 状态。 即使有活动插槽立即可用，也会发生此状态，尽管此状态仅在短时间内处于 "正在启动" 状态。 但是，如果通道必须等待活动插槽，那么它在等待时处于 "正在启动" 状态。

请求者通道未进入 "正在启动" 状态。 如果请求者通道由于活动通道数已达到限制而无法启动，那么通道将异常结束。

当通道 (请求者通道除外) 无法获取活动插槽，因此等待一个通道时，会将消息写入日志 z/OS 或 z/OS 控制台 并生成事件。 当稍后释放一个槽并且通道能够获取该槽时，将生成另一个消息和事件。 如果通道能够直接获取插槽，那么不会生成任何这些事件和消息。

如果在通道等待变为活动状态时发出 STOP CHANNEL 命令，那么通道将进入 STOPPED 状态。 发生 "通道已停止" 事件。

服务器连接通道包含在最大活动通道数中。

有关指定最大活动通道数的更多信息，请参阅以下主题:

- ALW 管理 IBM MQ。
- IBM i 管理 IBM MQ for IBM i。
- z/OS 管理 IBM MQ for z/OS。

通道错误
通道上的错误导致通道停止进一步的传输。 如果通道是发送方或服务器，那么它将进入 RETRY 状态，因为问题可能自行清除。 如果无法进入 RETRY 状态，那么通道将进入 STOPPED 状态。

对于发送通道，关联的传输队列设置为 GET (DISABLED) 并关闭触发。 (带有 STATUS (STOPPED) 的 STOP 命令将其发送到 STOPPED 状态; 只有断开连接时间间隔到期或带有 STATUS (INACTIVE) 的 STOP 命令才会使其正常结束并变为不活动状态。) 处于 STOPPED 状态的通道需要操作员干预才能重新启动 (请参阅 第 188 页的『重新启动已停止的通道』)。

注: 对于 IBM i IBM i AIX, Linux, and Windows 系统，必须运行通道启动程序才能尝试重试。 如果通道启动程序不可用，那么通道将变为不活动状态，并且必须手动重新启动。 如果您正在使用脚本来启动通道，请先确保通道启动程序正在运行，然后再尝试运行该脚本。

长重试计数 (LONGRTY) 描述了重试的工作方式。 如果该错误清除，那么通道将自动重新启动，并且将重新启用传输队列。 如果在未清除错误的情况下达到重试限制，那么通道将进入 STOPPED 状态。 操作员必须手动重新启动已停止的通道。 如果该错误仍然存在，那么不会再次重试。 当它成功启动时，将重新启用传输队列。

z/OS 如果通道启动程序在通道处于 "正在重试" 或 "已停止" 状态时停止，那么重新启动通道启动程序时将记住通道状态。 但是，如果通道启动程序在通道处于 STOPPED 状态时停止，那么将重置 SVRCONN 通道类型的通道状态。

Multi 如果队列管理器在通道处于 "正在重试" 或 "已停止" 状态时停止，那么重新启动队列管理器时将记住通道状态。 从 IBM MQ 8.0 开始，这也适用于 SVRCONN 通道。 先前，如果通道启动程序在通道处于 STOPPED 状态时停止，那么会重置 SVRCONN 通道类型的通道状态。

如果通道由于该队列已满或禁止放入而无法将消息放入目标队列，那么通道可以在某个时间间隔 (在消息重试时间间隔属性中指定) 重试该操作多次 (在消息重试时间间隔属性中指定)。 或者，您可以编写自己的消息重试出口，以确定导致重试的情况以及尝试次数。 通道在等待消息重试时间间隔完成时进入 PAUSED 状态。

请参阅 通道属性 以获取有关通道属性的信息，并参阅 消息传递通道的通道出口程序 以获取有关消息重试出口的信息。

服务器连接通道限制

您可以设置服务器连接通道限制，以防止客户机应用程序使用 **MAXINST** 参数耗尽队列管理器通道资源，并防止单个客户机应用程序使用 **MAXINSTC** 参数耗尽服务器连接通道容量。

使用 **DEFINE CHANNEL** 命令设置 **MAXINST** 和 **MAXINSTC** 。

单个队列管理器上可随时处于活动状态的通道的最大总数。 服务器连接通道实例总数包含在最大活动通道数中。

如果未指定可启动的服务器连接通道的最大并发实例数，那么单个客户机应用程序 (连接到单个服务器连接通道) 可能会耗尽可用的最大活动通道数。 当达到最大活动通道数时，将阻止在队列管理器上启动任何其他通道。 要避免此情况，必须限制可启动的单个服务器连接通道的并发实例数，而不考虑是哪个客户机启动了这些实例。

如果限制的值降低到低于当前正在运行的服务器连接通道实例数 (甚至为零) ，那么不会影响正在运行的通道。 只有在足够多的现有实例停止运行，以使当前正在运行的实例数小于限制值之后，才能启动新实例。

此外，许多不同的客户机连接通道可以连接到单个服务器连接通道。 对可启动的单个服务器连接通道的同时实例数的限制 (无论哪个客户机启动了这些实例) 可防止任何客户机耗尽队列管理器的最大活动通道容量。 如果您也不限制可从单个客户机启动的单个服务器连接通道的同时实例数，那么单个发生故障的客户机应用程序可能会打开如此多的连接，从而耗尽分配给单个服务器连接通道的通道容量，从而阻止需要使用该通道的其他客户机连接到该通道。 要避免此情况，必须限制可从单个客户机启动的单个服务器连接通道的并发实例数。

如果将个别客户机限制的值减小到低于当前从个别客户机运行的服务器连接通道的实例数 (即使是零) ，那么正在运行的通道不受影响。 但是，无法从超出新限制的单个客户机启动服务器连接通道的新实例，直到该客户机的足够现有实例停止运行，以使当前正在运行的实例数小于此参数的值。

相关参考
通道属性和通道类型
DEFINE CHANNEL

正在检查通道的另一端是否仍然可用

您可以使用脉动信号间隔，保持活动时间间隔和接收超时来检查通道的另一端是否可用。

脉动信号数

您可以使用脉动信号间隔通道属性来指定在传输队列上没有消息时将从发送 MCA 传递流，如 脉动信号间隔 (HBINT)中所述。

保持活动状态

在 IBM MQ for z/OS 中，如果使用 TCP/IP 作为传输协议，那么还可以为 **Keepalive** 时间间隔通道属性 (KAINT) 指定值。 建议将 **Keepalive** 时间间隔的值设置为高于脉动信号间隔的值，并将其设置为小于断开连接值的值。 您可以使用此属性为每个通道指定超时值，如 保持活动时间间隔 (KAINT)中所述。

在 IBM MQ for IBM i AIX, Linux, and Windows 系统中，如果使用 TCP 作为传输协议，那么可以设置 `keepalive=yes`。 如果指定此选项，那么 TCP 会定期检查连接的另一端是否仍然可用。 不是，通道终止。 此选项在 保持活动时间间隔 (KAINT)中进行了描述。

如果您有不可靠的通道报告 TCP 错误，那么使用 **Keepalive** 选项意味着您的通道更有可能恢复。

您可以指定时间间隔来控制 **Keepalive** 选项的行为。 更改时间间隔时，仅影响更改后启动的 TCP/IP 通道。 确保为时间间隔选择的值小于通道的断开连接时间间隔的值。

有关使用 **Keepalive** 选项的更多信息，请参阅 **DEFINE CHANNEL** 命令中的 **KAINT** 参数。

接收超时

如果使用 TCP 作为传输协议，那么如果在一段时间内未收到任何数据，那么也会关闭空闲非 MQI 通道连接的接收端。 此时间段 接收超时 值是根据 HBINT (脉动信号间隔) 值确定的。

在 IBM MQ for IBM i， AIX, Linux, and Windows 系统中，按如下所示设置 接收超时 值:

1. 对于初始流数，在进行任何协商之前， 接收超时 值是通道定义中的 HBINT 值的两倍。
2. 在通道协商 HBINT 值后，如果 HBINT 设置为小于 60 秒，那么 接收超时 值设置为该值的两倍。 如果 HBINT 设置为 60 秒或更长时间，那么 接收超时 值设置为比 HBINT 的值大 60 秒。

在 IBM MQ for z/OS 中， 接收超时 值设置如下:

1. 对于初始流数，在进行任何协商之前， 接收超时 值是通道定义中的 HBINT 值的两倍。
2. 如果设置了 RCVTIME ，那么超时设置为下列其中一项:
   - 协商的 HBINT 乘以一个常量
   - 协商的 HBINT 加上常量秒数
   - 恒定秒数

   取决于 RCVTTYPE 参数，并且受 RCVTMIN 施加的任何限制 (如果适用)。 当配置了 RCVTTYPE (EQUAL) 时， RCVTMIN 不适用。 如果使用常量值 RCVTIME 并且使用脉动信号间隔，请不要指定小于脉动信号间隔的 RCVTIME。 有关 RCVTIME ， RCVTMIN 和 RCVTTYPE 属性的详细信息，请参阅 ALTER QMGR 命令。

注:

1. 如果任一值为零，那么没有超时。
2. 对于不支持脉动信号的连接， HBINT 值在步骤 2 中协商为零，因此没有超时，因此必须使用 TCP/IP KEEPALIVE。
3. 对于使用共享对话的客户机连接，脉动信号可以一直流经通道 (从两端) ，而不仅仅是在 MQGET 处于未完成状态时。

4. 对于未使用共享对话的客户机连接，仅当客户机发出带有等待的 MQGET 调用时，才会从服务器流出脉动信号。 因此，建议不要将脉动信号间隔设置为对于客户机通道太小。 例如，如果脉动信号设置为 10 秒，那么 MQCMIT 调用失败 (使用 MQRC_CONNECTION_BROKEN) ，如果由于在此时间内没有数据流动而需要超过 20 秒才能落实。 这可能发生在大型工作单元上。 但是，如果为脉动信号间隔选择了相应的值，那么不会发生此情况，因为只有等待的 MQGET 需要很长时间。

   如果 SHARECNV 不为零，那么客户机将使用全双工连接，这意味着客户机可以 (并且可以) 在所有 MQI 调用期间执行脉动信号
5. 在 IBM WebSphere MQ 7 客户机通道中，脉动信号可以来自服务器和客户机端。 任一端的超时都基于 2*HBINT (对于小于 60 秒的 HBINT) 和 HBINT+60 (对于超过 60 秒的 HBINT)。
6. 在两次脉动信号间隔之后取消连接是有效的，因为期望至少在每个脉动信号间隔都有一个数据或脉动信号流。 但是，将脉动信号间隔设置得太小可能会导致问题，尤其是在使用通道出口时。 例如，如果 HBINT 值为 1 秒，并且使用了发送或接收出口，那么接收端在取消通道之前仅等待 2 秒。 如果 MCA 正在执行诸如加密消息之类的任务，那么此值可能太短。

建议设置

IBM MQ for z/OS

作为初始起点，您可以使用:

```
/cpf ALTER QMGR TCPKEEP(YES) RCVTTYPE(ADD) RCVTIME(60) ADOPTMCA(ALL) ADOPTCHK(ALL)
```

其中 `cpf` 是队列管理器子系统的命令前缀。

有关各种参数的更多信息，请参阅 ALTER QMGR 和 IBM MQ 网络可用性 。

如果发送方的 IP 地址可以转换为多个地址，那么您可能需要将 ADOPTCHK 设置为 QMNAME 而不是 ALL。

IBM MQ for Multiplatforms

在 `qm.ini` 中，添加以下信息:

```
TCP:
KeepAlive=Yes
CHANNELS:
AdoptNewMCA=ALL
AdoptNewMCACheck=ALL
```

请参阅 ALTER QMGR， 用于分布式排队的配置文件节和 第 95 页的『qm.ini 文件的通道节』 以获取更多信息。

如果发件人的 IP 地址可以转换为多个地址，那么您可能需要将 **AdoptNewMCACheck** 设置为 `QMNAME` 而不是 `ALL`。

采用 *MCA*

"采用 MCA" 功能使 IBM MQ 能够取消接收方通道并在其位置启动新的接收方通道。

如果通道失去联系，那么接收方通道可以处于 "通信接收" 状态。 重新建立通信时，发送方通道会尝试重新连接。 如果远程队列管理器发现接收方通道已在运行，那么它不允许启动同一接收方通道的另一个版本。 此问题需要用户干预以纠正问题或使用系统保持活动。

"采用 MCA" 功能可自动解决问题。 它使 IBM MQ 能够取消接收方通道并在其位置启动新的接收方通道。

相关任务

管理 IBM MQ
管理 IBM MQ for z/OS
管理 IBM MQ for IBM i

停止和停顿通道

您可以在断开连接时间间隔到期之前停止并停顿通道。

消息通道设计为仅由断开连接时间间隔通道属性控制有序终止的队列管理器之间长时间运行的连接。 除非操作员需要在断开连接时间间隔到期之前终止通道，否则此机制正常工作。 在以下情况下可能会发生此需求:

- 系统停顿
- 资源保护
- 通道一端的单方操作

在这种情况下，您可以停止通道。 您可以使用以下命令来执行此操作:

- STOP CHANNEL MQSC 命令
- 停止通道 PCF 命令
- IBM MQ 资源管理器
- IBM i z/OS 其他特定于平台的机制，具体如下:

  z/OS **对于 z/OS：**
  "停止通道" 面板

  IBM i **对于 IBM i：**
  ENDMQMCHL CL 命令或 WRKMQMCHL 面板上的 END 选项

有三个选项可用于使用以下命令停止通道:

QUIESCE
QUIESCE 选项尝试在停止通道之前结束当前消息批处理。

FORCE
FORCE 选项尝试立即停止通道，并可能要求通道在重新启动时再同步，因为通道可能处于不确定状态。

z/OS 在 IBM MQ for z/OS 上， FORCE 会中断正在进行的任何消息重新分配，这可能会使 BIND_NOT_FIXED 消息部分重新分配或不正常。

TERMINATE
TERMINATE 选项尝试立即停止通道，并终止该通道的线程或进程。

z/OS 在 IBM MQ for z/OS 上， TERMINATE 中断正在进行的任何消息重新分配，这可能导致 BIND_NOT_FIXED 消息部分重新分配或无序。

所有这些选项都使通道处于 STOPPED 状态，需要操作员干预才能将其重新启动。

在发送端停止通道是有效的，但需要操作员干预才能重新启动。 在通道的接收端，由于 MCA 正在等待来自发送端的数据，并且无法从接收端启动通道的 有序 终止; 停止命令处于暂挂状态，直到 MCA 从其等待数据返回为止。

因此，根据所需的操作特性，有三种建议使用通道的方法:

- 如果您希望通道长时间运行，请注意只能从发送端开始有序终止。 当通道中断 (即停止) 时，需要操作员干预 (START CHANNEL 命令) 才能重新启动这些通道。
- 如果您希望通道仅在存在要传输的消息时才处于活动状态，请将断开连接时间间隔设置为相当低的值。 缺省设置为高，因此建议不要用于需要此控制级别的通道。 由于难以中断接收通道，因此最经济的选择是让通道自动断开连接，并根据工作负载需求重新连接。 对于大多数通道，可以启发式地建立断开连接时间间隔的相应设置。
- 可以使用脉动信号间隔属性使发送 MCA 在没有要发送的消息的时间段内将脉动信号流发送到接收 MCA。 此操作将使接收 MCA 脱离其等待状态，并使其有机会在不等待断开连接时间间隔到期的情况下停顿通道。 请为脉动信号间隔指定小于断开连接时间间隔值的值。

  注:

  1. 建议您将断开连接时间间隔设置为较低的值，或者对服务器通道使用脉动信号。 此值较低是为了允许请求者通道异常结束 (例如，由于通道已取消) 的情况，当服务器通道没有要发送的消息时。 如果将断开连接时间间隔设置为高且未使用脉动信号，那么服务器不会检测到请求者已结束 (仅在下次尝试向请求者发送消息时才会执行此操作)。 当服务器仍在运行时，它会保持传输队列处于打开状态以进行独占输入，以便获取到达队列的任何更多消息。 如果尝试从请求者重新启动通道，那么启动请求会接收到

错误，因为服务器仍打开传输队列以进行互斥输入。 需要停止服务器通道，然后从请求者重新启动该通道。

重新启动已停止的通道

当通道进入 STOPPED 状态时，您必须手动重新启动通道。

关于此任务

对于发送方或服务器通道，当通道进入 STOPPED 状态时，关联的传输队列已设置为 GET (DISABLED)，并且触发已关闭。 接收到启动请求时，将自动重置这些属性。

z/OS 如果通道启动程序在通道处于 "正在重试" 或 "已停止" 状态时停止，那么重新启动通道启动程序时将记住通道状态。 但是，如果通道启动程序在通道处于 STOPPED 状态时停止，那么将重置 SVRCONN 通道类型的通道状态。

Multi 如果队列管理器在通道处于 "正在重试" 或 "已停止" 状态时停止，那么重新启动队列管理器时将记住通道状态。 从 IBM MQ 8.0 开始，这也适用于 SVRCONN 通道。 先前，如果通道启动程序在通道处于 STOPPED 状态时停止，那么会重置 SVRCONN 通道类型的通道状态。

过程

- 通过下列其中一种方式重新启动通道:
  - 通过使用 START CHANNEL MQSC 命令。
  - 通过使用 启动通道 PCF 命令。
  - 通过使用 IBM MQ Explorer
  - z/OS 在 z/OS 上，使用 启动通道面板。
  - IBM i 在 IBM i 上，使用 STRMQMCHL CL 命令 或 WRKMQMCHL 面板上的 START 选项。

处理不确定通道

不确定通道是对已发送和接收消息的远程通道存在疑问的通道。

请注意此消息与队列管理器之间的区别，该队列管理器对应该将哪些消息落实到队列有疑问。

您可以使用 "批处理脉动信号通道" 参数 (BATCHHB) 来减少通道处于不确定状态的机会。 指定此参数的值时，发送方通道会先检查远程通道是否仍处于活动状态，然后再执行任何进一步的操作。 如果未接收到响应，那么会将接收方通道视为不再处于活动状态。 消息可以回滚和重新路由，并且发送方通道不会处于不确定状态。 这会将通道可能处于不确定状态的时间缩短到发送方通道验证接收方通道是否仍处于活动状态，以及验证接收方通道是否已接收到已发送的消息之间的时间段。 有关批处理脉动信号参数的更多信息，请参阅 通道属性 。

通常会自动解决不确定通道问题。 即使通信丢失，并且通道在接收状态未知的发送方处与消息批处理存在疑问时，也会在重新建立通信时解决此情况。 为此目的保留了序号和 LUWID 记录。 通道处于不确定状态，直到已交换 LUWID 信息，并且通道只能有一批消息处于不确定状态。

必要时，您可以手动再同步通道。 术语 *manual* 包括使用包含 IBM MQ 系统管理命令的操作程序或程序。 手动再同步过程如下所示。 此描述使用 MQSC 命令，但您也可以使用 PCF 等效命令。

1. 使用 DISPLAY CHSTATUS 命令可查找通道的 **每个** 端的上次落实的逻辑工作单元标识 (LUWID)。 使用以下命令执行此操作:
   - 对于通道的不确定端:

     ```
     DISPLAY CHSTATUS( name ) SAVED CURLUWID
     ```

     您可以使用 CONNAME 和 XMITQ 参数来进一步标识通道。
   - 对于通道的接收方:

     ```
     DISPLAY CHSTATUS( name ) SAVED LSTLUWID
     ```

可以使用 CONNAME 参数来进一步标识通道。

这些命令不同，因为只有通道的发送方可能存在疑问。 接收方从不存疑。

在 IBM MQ for IBM i 上，可以使用 STRMQMMQSC 命令或 "使用 MQM 通道状态" CL 命令 WRKMQMCHST 从文件执行 DISPLAY CHSTATUS 命令

2. 如果两个 LUWID 相同，那么接收方已落实发送方认为存在疑问的工作单元。 发送方现在可以从传输队列中除去不确定消息并重新启用该消息。 这是通过以下通道 RESOLVE 命令完成的:

```
RESOLVE CHANNEL( name ) ACTION(COMMIT)
```

3. 如果两个 LUWID 不同，那么接收方未落实发送方认为存在疑问的工作单元。 发送方需要将不确定消息保留在传输队列上，然后重新发送这些消息。 这是通过以下通道 RESOLVE 命令完成的:

```
RESOLVE CHANNEL( name ) ACTION(BACKOUT)
```

IBM i 在 IBM MQ for IBM i 上，可以使用 "解析 MQM 通道" 命令 RSVMQMCHL。

此过程完成后，通道将不再处于不确定状态。 如果需要，传输队列现在可以由另一个通道使用。

消息的安全性

除了 IBM MQ 的典型恢复功能外，分布式队列管理还确保通过使用消息通道两端之间协调的同步点过程来正确传递消息。 如果此过程检测到错误，那么它会关闭通道，以便您可以调查问题，并将消息安全地保留在传输队列中，直到通道重新启动为止。

同步点过程具有额外的优势，因为它尝试在通道启动时恢复 不确定 情境。 (不确定 是已请求同步点但请求结果未知的恢复单元的状态。) 与此设施关联的还有两个功能:

1. 使用落实或回退进行解析
2. 重置序号

只有在特殊情况下才会使用这些功能，因为在大多数情况下通道会自动恢复。

快速非持久消息

非持久消息速度 (NPMSPEED) 通道属性可用于指定要更快地传递通道上的任何非持久消息。 有关此属性的更多信息，请参阅 非持久消息速度 (NPMSPEED)。

如果通道在快速传输非持久消息时终止，那么消息可能会丢失，并且应由应用程序在需要时安排其恢复。

如果接收通道无法将消息放入其目标队列，那么将其放入死信队列 (如果已定义)。 否则，将废弃该消息。

注: 如果通道的另一端不支持该选项，那么通道将以正常速度运行。

未发送的消息

有关无法传递消息时发生的情况的信息，请参阅 第 189 页的『无法传递消息时会发生什么情况?』。

无法传递消息时会发生什么情况?

当无法传递消息时， MCA 可以通过多种方式对其进行处理。 它可以重试，可以返回到发送方，也可以将其放在死信队列上。

第 190 页的图 25 显示了当 MCA 无法将消息放入目标队列时发生的处理。 (显示的选项并不适用于所有平台。)

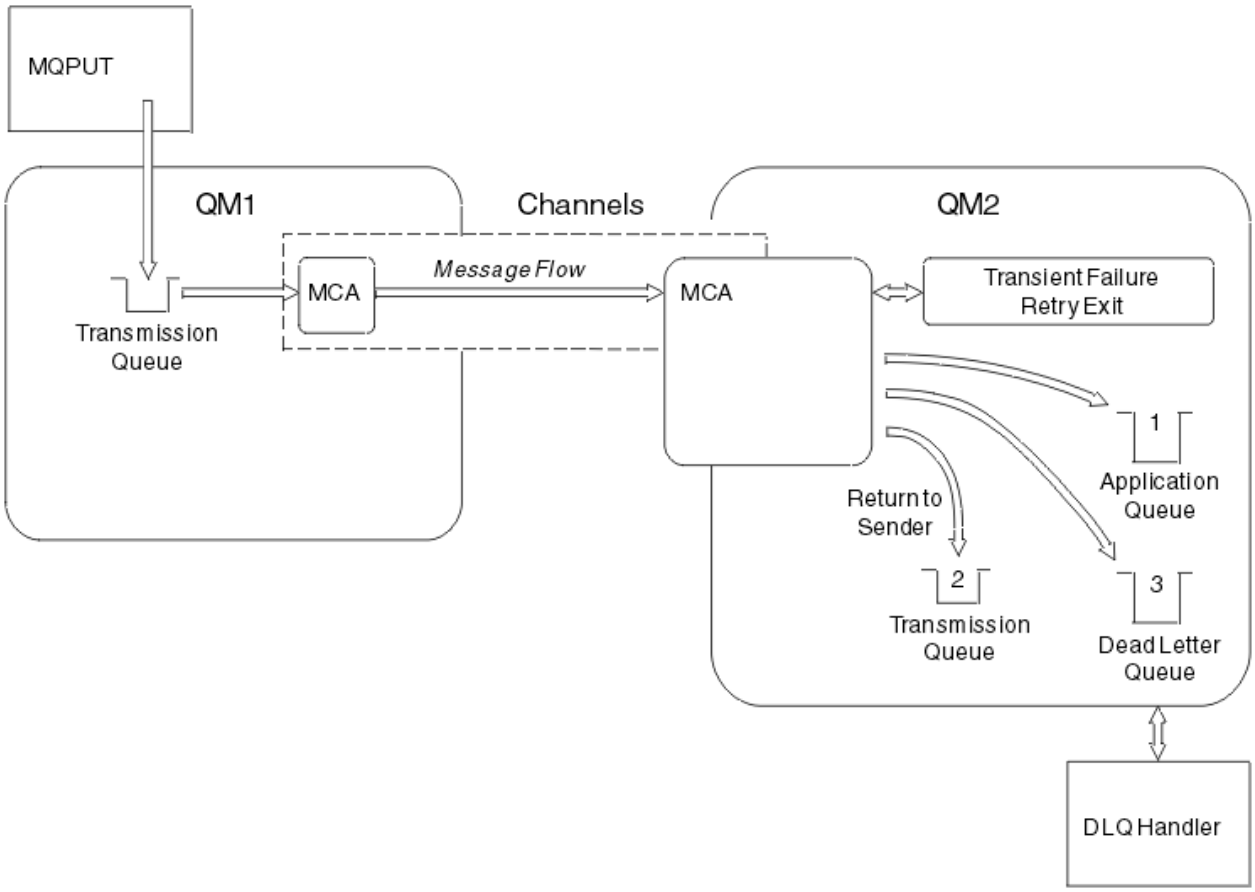


图 *25:* 无法传递消息时发生的情况

如图所示， MCA 可以用它无法传递的消息做几件事。 所执行的操作由定义通道时指定的选项以及消息的 MQPUT 报告选项确定。

1. 消息重试 (message-retry)

   如果 MCA 无法将消息放入目标队列，原因可能是暂时性的 (例如，由于队列已满) ，那么 MCA 可以等待并稍后重试该操作。 您可以确定 MCA 是否等待，等待时间以及尝试次数。

   - 您可以在定义通道时指定 MQPUT 错误的消息重试时间和时间间隔。 如果由于队列已满或禁止放入而无法将消息放入目标队列，那么 MCA 将在指定的时间间隔内按指定的次数尝试操作。
   - 您可以编写自己的消息重试出口。 该出口使您能够指定希望 MCA 在哪些条件下重试 MQPUT 或 MQOPEN 操作。 定义通道时指定出口的名称。

2. 返回发送方 (return-to-sender)

   如果消息重试失败，或者迁到其他类型的错误，那么 MCA 可以将消息发送回发起方。 要启用 "返回到发送方" ，将消息放入原始队列时，需要在消息描述符中指定以下选项:

   - MQRO_EXCEPTION_WITH_FULL_DATA 报告选项
   - MQRO_DISCARD_MSG 报告选项
   - 应答队列和应答队列管理器的名称

   如果 MCA 无法将消息放入到目标队列，那么它将生成一个包含原始消息的异常报告，并将该报告放入传输队列中，此传输队列随后发送至原始消息中指定的应答队列。 （如果应答队列与 MCA 在相同的队列管理器上，那么该消息将直接放入该队列，而不是传输队列。）

3. 死信队列

   如果无法传递或返回消息，那么会将其放入死信队列 (DLQ)。 您可以使用 DLQ 处理程序来处理消息。 此处理在针对 IBM MQ for UNIX， Linux 和 Windows 系统的 处理死信队列上的消息 以及针对 z/OS 系统的

死信队列处理程序实用程序 (CSQUDLQH) 中进行了描述。 如果死信队列不可用，那么发送 MCA 会将消息保留在传输队列上，并且通道会停止。 在快速通道上，无法写入死信队列的非持久消息将丢失。

在 IBM WebSphere MQ 7.0 上，如果未定义本地死信队列，那么远程队列不可用或未定义，并且没有远程死信队列，那么发送方通道将进入 RETRY ，并且消息将自动回滚到传输队列。

相关参考

使用死信队列 (USEDLQ)

触发通道

IBM MQ 提供了用于在满足队列上的特定条件时自动启动应用程序的工具。 此工具称为触发。

此说明旨在作为触发概念的概述。 有关完整描述，请参阅 使用触发器启动 IBM MQ 应用程序。

有关特定于平台的信息，请参阅以下内容:

- 对于 AIX, Linux, and Windows，请参阅第 192 页的『正在 AIX, Linux, and Windows 上触发通道。』
- IBM i 对于 IBM i，请参阅第 192 页的『IBM MQ for IBM i 中的触发通道』
- z/OS 对于 z/OS，请参阅第 799 页的『传输队列和触发通道』

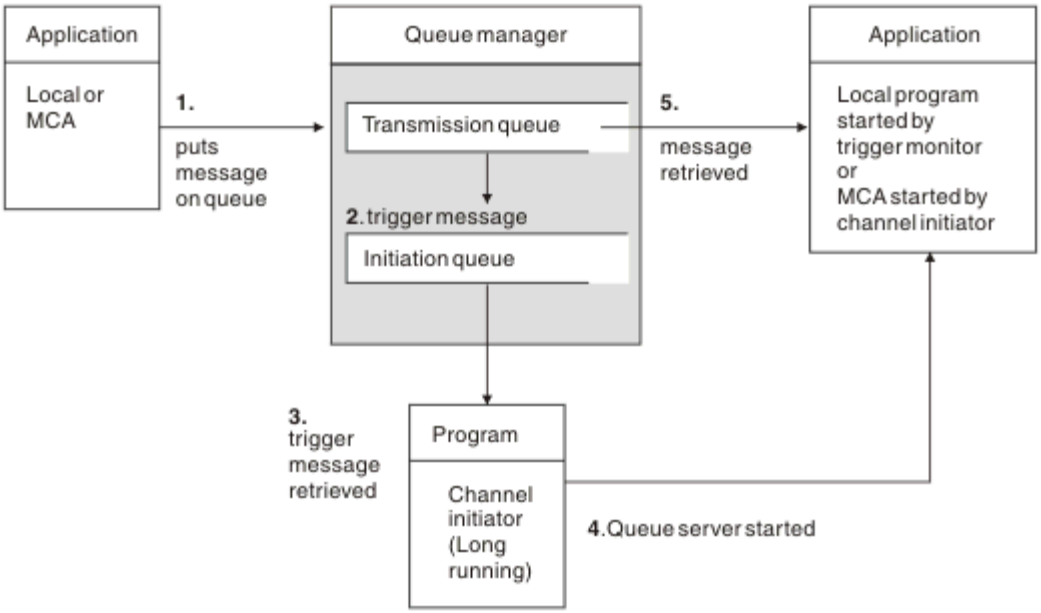


图 *26:* 触发的概念

第 191 页的图 26 中显示了触发所需的对象。 它显示以下事件序列:

1. 本地队列管理器将来自应用程序或来自消息通道代理程序 (MCA) 的消息放置在传输队列上。
2. 当满足触发条件时，本地队列管理器将触发器消息放置在启动队列上。
3. 长时间运行的通道启动程序监视启动队列，并在消息到达时检索消息。
4. 通道启动程序根据其中包含的信息来处理触发器消息。 此信息可能包含通道名称，在此情况下将启动相应的 MCA。
5. 本地应用程序或 MCA (已触发) 从传输队列中检索消息。

要设置此方案，您需要:

- 使用启动队列 (即， SYSTEM.CHANNEL.INITQ)。
- 确保启动队列 (SYSTEM.CHANNEL.INITQ) 存在。

- 确保通道启动程序可用且正在运行。 必须在其启动命令中为通道启动程序提供启动队列的名称。
  z/OS 在 z/OS 上，启动队列的名称是固定的，因此不用于启动命令。
- (可选) 为触发创建进程定义 (如果它不存在)，并确保 *UserData* 字段包含它所服务的通道的名称。 您可以在传输队列的 **TriggerData** 属性中指定通道名称，而不是创建进程定义。 对于 IBM i IBM i AIX, Linux, and Windows 系统， IBM MQ 允许将通道名称指定为空白，在这种情况下，将使用具有此传输队列的第一个可用通道定义。
- 确保传输队列定义包含为其提供服务的进程定义的名称 (如果适用)，启动队列名称以及您认为最适合的触发特征。 触发器控制属性允许根据需要启用或不启用触发。

注:

1. 通道启动程序充当监视用于启动通道的启动队列的 "触发器监视器"。
2. 可以使用启动队列和触发进程来触发任意数量的通道。
3. 可以定义任意数量的启动队列和触发器进程。
4. 建议使用触发器类型 FIRST，以避免在通道启动时淹没系统。

正在 AIX, Linux, and Windows 上触发通道。

ALW

您可以在 IBM MQ 中创建流程定义，以定义要触发的流程。 使用 MQSC 命令 DEFINE PROCESS 来创建进程定义，以命名要在消息到达传输队列时触发的进程。 进程定义的 USERDATA 属性包含由传输队列提供服务的通道的名称。

定义本地队列 (QM4)，指定将触发器消息写入启动队列 (IQ) 以触发启动通道的应用程序 (QM3.TO.QM4):

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(P1) USAGE(XMITQ)
```

定义要启动的应用程序 (进程 P1):

```
DEFINE PROCESS(P1) USERDATA(QM3.TO.QM4)
```

或者，对于 IBM MQ for UNIX， Linux 和 Windows 系统，您可以通过在传输队列的 TRIGDATA 属性中指定通道名称来消除对进程定义的需求。

定义本地队列 (QM4)。 指定将触发器消息写入缺省启动队列 SYSTEM.CHANNEL.INITQ，以触发启动通道 (QM3.TO.QM4) 的应用程序 (进程 P1):

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ)
USAGE(XMITQ) TRIGDATA(QM3.TO.QM4)
```

如果未指定通道名称，那么通道启动程序将搜索通道定义文件，直到找到与指定传输队列相关联的通道为止。

IBM MQ for IBM i 中的触发通道

IBM i

IBM MQ for IBM i 中通道的触发是通过通道启动程序进程实现的。 启动队列 SYSTEM.CHANNEL.INITQ，除非通过改变队列管理器 SCHINIT 属性将其禁用。

为通道设置传输队列，指定 SYSTEM.CHANNEL.INITQ 作为启动队列，并对队列启用触发。 通道启动程序启动指定此传输队列的第一个可用通道。

```
CRTMQMQ QNAME(MYXMITQ1) QTYPE(*LCL) MQMNAME(MYQMGR)
TRGENBL(*YES) INITQNAME(SYSTEM.CHANNEL.INITQ)
USAGE(*TMQ)
```

您最多可以使用 STRMQMCHLI 命令手动启动三个通道启动程序进程，并指定不同的启动队列。 您还可以指定多个能够处理传输队列的通道，并选择要启动的通道。 仍提供此功能以与先前发行版兼容。 不推荐使用其用法。

注: 一次只有一个通道可以处理传输队列。

```
STRMQMCHLI QNAME(MYINITQ)
```

为通道设置传输队列，指定 TRGENBL (*YES) ，并在 TRIGDATA 字段中指定通道名称，以选择要尝试启动的通道。 例如:

```
CRTMQMQ QNAME(MYXMITQ2) QTYPE(*LCL) MQMNAME(MYQMGR)
TRGENBL(*YES) INITQNAME(MYINITQ)
USAGE(*TMQ) TRIGDATA(MYCHANNEL)
```

相关概念

第 193 页的『启动和停止通道启动程序』
使用通道启动程序进程实现触发。

相关任务

第 152 页的『配置分布式队列』
此部分提供有关 IBM MQ 安装之间的相互通信的更详细信息，包括队列定义，通道定义，触发和同步点过程

相关参考

AIX, Linux, and Windows 上的通道程序

IBM i IBM i 上的相互通信作业

IBM i IBM i 上的通道状态

启动和停止通道启动程序

使用通道启动程序进程实现触发。

此通道启动程序进程是使用 MQSC 命令 START CHINIT 启动的。 除非您正在使用缺省启动队列，否则请在命令上指定启动队列的名称。 例如，要使用 START CHINIT 命令为缺省队列管理器启动队列 IQ ，请输入:

```
START CHINIT INITQ(IQ)
```

缺省情况下，将使用缺省启动队列 SYSTEM.CHANNEL.INITQ。 如果要手动启动所有通道启动程序，请执行以下步骤:

1. 创建并启动队列管理器。
2. 将队列管理器的 SCHINIT 属性更改为 MANUAL
3. 结束并重新启动队列管理器

在 IBM MQ for Multiplatforms 系统中，将自动启动通道启动程序。 您可以启动的通道启动程序数量有限。 缺省值和最大值都是 3。 您可以在 AIX and Linux 系统的 qm.ini 文件和 Windows 系统的注册表中使用 MAXINITIATORS 对此进行更改。

有关运行通道启动程序命令 **runmqchi** 和其他控制命令的详细信息，请参阅 IBM MQ 控制命令 。

停止通道启动程序

启动队列管理器时，将自动启动缺省通道启动程序。 当队列管理器停止时，将自动停止所有通道启动程序。

初始化和配置文件

通道初始化数据的处理取决于 IBM MQ 平台。

IBM MQ for z/OS

z/OS

在 IBM MQ for z/OS 中，使用 **ALTER QMGR** MQSC 命令指定初始化和配置信息。 如果将 **ALTER QMGR** 命令放在 CSQINP2 初始化输入数据集中，那么每次启动队列管理器时都会处理这些命令。

要在每次启动通道启动程序时运行 MQSC 命令 (例如 **START LISTENER**)，请将这些命令放在 CSQINPX 初始化输入数据集中，并在通道启动程序启动式任务过程中指定可选 DD 语句 CSQINPX。

有关 CSQINP2 和 CSQINPX 的更多信息，请参阅 定制初始化输入数据集和 ALTER QMGR。

IBM MQ for Multiplatforms

Multi

在 IBM MQ for Multiplatforms 中，有一些配置文件用于保存有关 IBM MQ 安装的基本配置信息。

有两个配置文件: 一个适用于机器，另一个适用于单个队列管理器。

IBM MQ 配置文件

此文件保存与 IBM MQ 系统上的所有队列管理器相关的信息。 该文件名为 `mqs.ini`。 此操作在第 72 页的『IBM MQ 配置文件， mqs.ini』中进行了描述。

队列管理器配置文件

此文件保存与一个特定队列管理器相关的配置信息。 该文件称为 `qm.ini`。

它是在队列管理器创建期间创建的，可以保存与队列管理器的任何方面相关的配置信息。 该文件中保存的信息包括日志配置与 IBM MQ 配置文件中的缺省值有何不同的详细信息。

队列管理器配置文件保存在队列管理器占用的目录树的根目录中。 例如，对于 **DefaultPath** 属性，名为 QMNAME 的队列管理器的队列管理器配置文件将为:

对于 AIX and Linux 系统:

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

对于 Windows 系统:

```
CMQ\qmgrs\QMNAME\qm.ini
```

IBM i 对于 IBM i:

```
/QIBM/UserData/mqm/qmgrs/QMNAME/qm.ini
```

以下是 `qm.ini` 的摘录。 它指定 TCP/IP 侦听器将在端口 2500 上进行侦听，当前通道的最大数目为 200，活动通道的最大数目为 100。

```
TCP:
Port=2500
CHANNELS:
MaxChannels=200
MaxActiveChannels=100
```

您可以指定要由出站通道使用的 TCP/IP 端口范围。 一种方法是使用 `qm.ini` 文件来指定端口值范围的开始和结束。 以下示例显示指定了一系列通道的 `qm.ini` 文件:

```
TCP:
StrPort=2500
EndPort=3000
CHANNELS:
MaxChannels=200
MaxActiveChannels=100
```

如果为 **StrPort** 或 **EndPort** 指定值，那么必须为两者指定值。 **EndPort** 的值必须始终大于 **StrPort** 的值。

通道尝试使用指定范围内的每个端口值。 当连接成功时，端口值是通道随后使用的端口。

有关 `qm.ini` 文件的更多信息，请参阅 第 82 页的『队列管理器配置文件， qm.ini』。

消息的数据转换

在不同队列管理器上的队列之间发送 IBM MQ 消息时，可能需要数据转换。

IBM MQ 消息由两部分组成:

- 消息描述符中的控制信息
- 应用程序数据

当在不同队列管理器上的队列之间发送时，这两个部分中的任一部分都可能需要数据转换。 有关应用程序数据转换的信息，请参阅 应用程序数据转换。

编写您自己的消息通道代理程序

IBM MQ 允许您编写自己的消息通道代理程序 (MCA) 程序或从独立软件供应商安装程序。

您可能希望编写自己的 MCA 程序，以使 IBM MQ 通过自己的专有通信协议进行互操作，或者通过 IBM MQ 不支持的协议发送消息。 (您无法编写自己的 MCA，以便在另一端与 IBM MQ 提供的 MCA 进行互操作。)

如果您决定使用 IBM MQ 未提供的 MCA，那么必须考虑以下几点。

消息发送和接收
您必须编写一个发送应用程序，该应用程序从应用程序放置消息的任何位置 (例如从传输队列) 获取消息，并通过要与之进行通信的协议将这些消息发送出去。 您还必须编写接收应用程序，以从该协议获取消息并将其放入目标队列。 发送和接收应用程序使用消息队列接口 (MQI) 调用，而不是任何特殊接口。

必须确保仅传递一次消息。 可以使用同步点协调来帮助进行此交付。

通道控制功能
您必须提供自己的管理功能以控制通道。 不能将 IBM MQ 通道管理功能用于配置 (例如， DEFINE CHANNEL 命令) 或监视 (例如， DISPLAY CHSTATUS) 通道。

初始化文件
如果您需要一个初始化文件，那么必须提供自己的初始化文件。

应用程序数据转换
您可能希望允许对发送到其他系统的消息进行数据转换。 如果是这样，请在 MQGET 调用上使用 MQGMO_CONVERT 选项，从应用程序放置消息的位置 (例如传输队列) 检索消息。

用户出口
请考虑是否需要用户出口。 如果是这样，您可以使用 IBM MQ 使用的相同接口定义。

触发
如果应用程序将消息放入传输队列，那么可以设置传输队列属性，以便在消息到达队列时触发发送 MCA。

通道启动程序
您可能必须提供自己的通道启动程序。

分布式队列管理要考虑的其他事项

准备 IBM MQ 以进行分布式队列管理时要考虑的其他主题。 本主题涵盖未交付的消息队列，正在使用的队列，系统扩展和用户出口程序以及作为可信应用程序运行的通道和侦听器。

未送达消息的队列

要确保处理到达未传递消息队列 (也称为死信队列或 DLQ) 的消息，请创建可触发或定期运行的程序以处理这些消息。

Linux AIX 在 AIX and Linux 系统上随 IBM MQ 提供了 DLQ 处理程序; 有关更多信息，请参阅 样本 DLQ 处理程序 amqsdlq。

IBM i 有关 IBM MQ for IBM i 的更多信息，请参阅 IBM MQ for IBM i 死信队列处理程序。

正在使用的队列

即使未传输消息，接收方通道的 MCA 也可以保持目标队列处于打开状态。 这将导致队列显示为 “正在使用”。

最大通道数

IBM i 在 IBM MQ for IBM i 上，可以指定系统中允许的最大通道数以及一次可以处于活动状态的最大通道数。 在目录 QIBM/UserData/mqm/qmgrs/*queue_manager_name* 中的 `qm.ini` 文件中指定这些数字。 请参阅 分布式排队的配置文件节。

系统扩展和用户出口程序

在通道定义中提供了一个工具，以允许在消息处理期间在定义的时间运行额外的程序。 这些程序并非随 IBM MQ 一起提供，而是可由每个安装根据本地需求提供。

要运行，这些用户出口程序必须具有预定义的名称，并且在调用通道程序时可用。 用户出口程序的名称包含在消息通道定义中。

有一个定义的控制块接口，用于将控制移交给这些程序，以及处理从这些程序返回的控制。

将在 消息传递通道的通道出口程序中找到调用这些程序的精确位置以及控制块和名称的详细信息。

将通道和侦听器作为可信应用程序运行

如果性能是环境中的重要考虑因素，并且环境稳定，那么可以使用 FASTPATH 绑定以可信方式运行通道和侦听器。 有两个因素会影响通道和侦听器是否以可信方式运行:

- 环境变量 MQ_CONNECT_TYPE=FASTPATH 或 MQ_CONNECT_TYPE = STANDARD。 这是区分大小写的。 如果指定的值无效，那么将忽略该值。
- `qm.ini` 或注册表文件的 Channels 节中的 MQIBindType 。 您可以将此设置为 FASTPATH 或 STANDARD ，并且它不区分大小写。 缺省值为“标准”。

您可以将 MQIBindType 与环境变量结合使用，以实现必需的效果，如下所示:

MQIBindType	**环境变量**	**结果**
标准	未定义	标准
FASTPATH	未定义	FASTPATH
标准	标准	标准
FASTPATH	标准	标准
标准	FASTPATH	标准
FASTPATH	FASTPATH	FASTPATH
标准	CLIENT	CLIENT
FASTPATH	CLIENT	标准
标准	LOCAL	标准
FASTPATH	LOCAL	标准

总之，实际上只有两种方式可以让频道和听众以可信赖的方式运行:

1. 通过在 `qm.ini` 或注册表中指定 MQIBindType= FASTPATH ， 而不指定环境变量。
2. 通过在 `qm.ini` 或注册表中指定 MQIBindType= FASTPATH 并将环境变量设置为 FASTPATH。

请考虑以可信方式运行侦听器，因为侦听器是稳定的进程。 除非您正在使用不稳定的通道出口或命令 STOP CHANNEL MODE (TERMINATE) ， 否则请考虑将通道作为可信通道运行。

ALW 监视和控制 AIX, Linux, and Windows 上的通道

对于 DQM，您需要创建，监视和控制到远程队列管理器的通道。 您可以使用命令，程序， IBM MQ Explorer，通道定义的文件以及同步信息的存储区来控制通道。

关于此任务

您可以使用以下类型的命令来控制通道:

IBM MQ 命令 (MQSC)
您可以在 AIX, Linux, and Windows 系统中的 MQSC 会话中使用 MQSC 作为单个命令。 要发出更复杂的命令或多个命令，可以将 MQSC 构建到一个文件中，然后从命令行运行该文件。 有关详细信息，请参阅 MQSC 命令。 本节提供了一些使用 MQSC 进行分布式排队的简单示例。

通道命令是 IBM MQ 命令 (MQSC) 的子集。 您可以使用 MQSC 和控制命令来执行以下操作:

- 创建，复制，显示，更改和删除通道定义
- 无法重新建立链接时启动和停止通道， ping，重置通道序号以及解析不确定消息
- 显示有关通道的状态信息

控制命令
您还可以在命令行上针对其中某些功能发出 控制命令 。 有关详细信息，请参阅 使用控制命令进行管理。

可编程命令格式命令
有关详细信息，请参阅 PCF 命令。

Windows Linux **IBM MQ Explorer**
在 Linux 和 Windows 系统上，可以使用 IBM MQ Explorer。 这提供了一个图形管理界面来执行管理任务，作为使用控制命令或 MQSC 命令的替代方法。 通道定义作为队列管理器对象保留。

每个队列管理器都有一个 DQM 组件，用于控制与兼容远程队列管理器的互连。 存储区包含序号和 逻辑工作单元 *(LUW)* 标识。 这些用于通道同步目的。

有关使用不同类型的命令设置和控制消息通道时可用的功能的列表，请参阅 第 198 页的表 22。

过程

- 第 197 页的『设置和控制通道所需的功能』
- 第 200 页的『对象入门』
- 第 205 页的『在 Windows 上设置通信』
- 第 211 页的『在 AIX and Linux 上设置通信』

相关任务

第 216 页的『监视和控制 IBM i 上的通道』

使用 DQM 命令和面板来创建，监视和控制到远程队列管理器的通道。 每个队列管理器都有一个 DQM 程序，用于控制与兼容远程队列管理器的互连。

相关参考
ALW AIX, Linux, and Windows 上的通道程序
ALW AIX, Linux, and Windows 的消息通道规划示例
示例配置信息
通道属性

ALW 设置和控制通道所需的功能

可能需要许多 IBM MQ 功能来设置和控制通道。 本主题中对通道函数进行了说明。

您可以使用 IBM MQ 提供的缺省值创建通道定义，指定通道名称，要创建的通道类型，要使用的通信方法，传输队列名称和连接名称。

通道两端的通道名称必须相同，并且在网络中是唯一的。 但是，必须将使用的字符限制为对 IBM MQ 对象名有效的字符。

有关其他与通道相关的功能，请参阅以下主题:

- 第 200 页的『对象入门』
- 第 200 页的『创建关联对象』
- 第 200 页的『创建缺省对象』
- 第 201 页的『创建通道』
- 第 201 页的『显示频道』
- 第 201 页的『显示通道状态』
- 第 202 页的『使用 Ping 检查链接』
- 第 202 页的『启动通道』
- 第 203 页的『停止通道』
- 第 204 页的『重命名通道』
- 第 204 页的『重置通道』
- 第 205 页的『在通道上解析不确定消息』

第 198 页的表 22 显示了您可能需要的 IBM MQ 函数的完整列表。

表 *22: AIX, Linux, and Windows* 系统中所需的功能

函数	**控制命令**	**MQSC**	**IBM MQ 资源管理器等效项?**
队列管理器函数			
更改队列管理器		ALTER QMGR	Yes
创建队列管理器	crtmqm		Yes
删除队列管理器	dltmqm		Yes
显示队列管理器		显示 Qmgr	Yes
结束队列管理器	endmqm		Yes
Ping 队列管理器		PING QMGR	否
启动队列管理器	strmqm		Yes
命令服务器功能			
显示命令服务器	dspmqcsv		否
结束命令服务器	endmqcsv		否
启动命令服务器	strmqcsv		否
队列函数			
更改队列		ALTER QALIAS ALTER QLOCAL ALTER QMODEL ALTER QREMOTE 请参阅 ALTER 队列。	Yes
清除队列		清除 QLocal	Yes

表 22: AIX, Linux, and Windows 系统中所需的功能 (继续)			
函数	**控制命令**	**MQSC**	**IBM MQ 资源管理器等效项?**
创建队列		DEFINE QALIAS DEFINE QLOCAL DEFINE QMODEL DEFINE QREMOTE 请参阅 DEFINE 队列。	Yes
删除队列		DELETE QALIAS DELETE QLOCAL DELETE QMODEL DELETE QREMOTE 请参阅 DELETE 队列。	Yes
DISPLAY QUEUE		DISPLAY QUEUE	Yes
进程函数			
更改进程		变更过程	Yes
创建进程		DEFINE PROCESS	Yes
删除进程		删除进程	Yes
显示过程		显示过程	Yes
通道函数			
更改通道		ALTER CHANNEL	Yes
创建通道		DEFINE CHANNEL	Yes
删除通道		删除通道	Yes
显示通道		显示通道	Yes
显示通道状态		DISPLAY CHSTATUS	Yes
结束通道		停止通道	Yes
Ping 通道		Ping 通道	Yes
重置通道		重置通道	Yes
解析通道		解析通道	Yes
运行通道	运行 mqchl	启动通道	Yes
运行通道启动程序	运行 mqchi	START CHINIT	否
运行侦听器 [1]	运行 mqlsr	启动侦听器	否
结束侦听器	endmqlsr，仅在以下平台上: • AIX • Windows Windows 系统		否

注:

1. 当队列管理器启动时，可能会自动启动侦听器。

ALW 对象入门

必须先定义通道，并且它们的关联对象必须存在并且可供使用，然后才能启动通道。 本节将向您展示如何执行此操作。

使用 IBM MQ 命令 (MQSC) 或 IBM MQ Explorer 来执行以下操作:

1. 定义消息通道和关联对象
2. 监视和控制消息通道

您可能需要定义的关联对象包括:

- 传输队列
- 远程队列定义
- 队列管理器别名定义
- 应答队列别名定义
- 应答本地队列
- 触发过程 (MCA)
- 消息通道定义

必须先定义每个通道的特定通信链路并使其可用，然后才能运行通道。 有关如何定义 LU 6.2， TCP/IP，NetBIOS， SPX 和 DECnet 链接的描述，请参阅安装的特定通信指南。 另请参阅 示例配置信息。

有关创建和使用对象的更多信息，请参阅以下子主题:

ALW 创建关联对象

MQSC 用于创建关联对象。

使用 MQSC 来创建队列和别名对象: 传输队列，远程队列定义，队列管理器别名定义，应答队列别名定义和应答本地队列。

还要以类似方式创建用于触发 (MCA) 的进程的定义。

有关显示如何创建所有必需对象的示例，请参阅 AIX, Linux, and Windows 的消息通道规划示例。

ALW 创建缺省对象

创建队列管理器时，将自动创建缺省对象。 这些对象是队列，通道，进程定义和管理队列。 创建缺省对象后，您可以随时通过使用 -c 选项运行 strmqm 命令来替换这些对象。

使用 crtmqm 命令创建队列管理器时，该命令还会启动程序以创建一组缺省对象。

1. 将依次创建每个缺省对象。 程序会保留已成功定义的对象数，已存在和已替换的对象数以及尝试失败的次数的计数。
2. 程序会向您显示结果，如果发生任何错误，请将您引导至相应的错误日志以获取详细信息。

当程序完成运行时，可以使用 strmqm 命令来启动队列管理器。

有关 crtmqm 和 strmqm 命令的更多信息，请参阅 使用控制命令进行管理 。

更改缺省对象

当您指定 -c 选项时，将在创建对象时临时启动队列管理器，然后再次关闭该队列管理器。 使用 -c 选项发出 strmqm 将使用缺省值刷新现有系统对象 (例如，通道定义的 MCAUSER 属性设置为空白)。 如果要启动队列管理器，那么必须在不使用 -c 选项的情况下再次使用 strmqm 命令。

如果要更改缺省对象，您可以创建自己的旧 amqscoma.tst 文件版本并对其进行编辑。

ALW 创建通道

创建两个通道定义，在连接的每个端创建一个通道定义。 在第一个队列管理器上创建第一个通道定义。 然后，在链路的另一端的第二个队列管理器上创建第二个通道定义。

必须使用相同的通道名称来定义两端。 两端必须具有兼容的通道类型，例如: 发送方和接收方。

要为链路的一端创建通道定义，请使用 MQSC 命令 DEFINE CHANNEL。 包括通道的名称，连接的此端的通道类型，连接名称，描述 (如果需要) ，传输队列的名称 (如果需要) 和传输协议。 还包括您希望与所需通道类型的系统缺省值不同的任何其他属性 (使用先前收集的信息)。

您可以帮助确定 通道属性中通道属性的值。

注: 建议您唯一地命名网络中的所有通道。 在通道名称中包含源队列管理器名称和目标队列管理器名称是执行此操作的良好方法。

创建通道示例

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) +
DESCR('Sender channel to QM2') +
CONNAME(QM2) TRPTYPE(TCP) XMITQ(QM2) CONVERT(YES)
```

在 MQSC 的所有示例中，命令显示为显示在命令文件中，并在 AIX, Linux, and Windows 中输入。 这两种方法看起来完全相同，但要以交互方式发出命令，必须首先启动 MQSC 会话。 输入 `runmqsc`(对于缺省队列管理器) 或 `runmqsc` *qmname* (其中 *qmname* 是所需队列管理器的名称)。 然后输入任意数量的命令，如示例中所示。

对于可移植性，请将命令的行长度限制为 72 个字符。 使用并置字符 +，如所示在多行上继续:

- Windows 在 Windows 上，使用 Ctrl-z 在命令行结束条目。
- Linux AIX 在 AIX and Linux 上，使用 Ctrl-d。
- 或者，在 AIX, Linux, and Windows 上，使用 **end** 命令。

ALW 显示频道

使用 MQSC 命令 DISPLAY CHANNEL 显示通道的属性。

如果没有请求特定属性，且指定的通道名不是通用名，则默认使用 DISPLAY CHANNEL 命令的 ALL 参数。

通道属性中描述了这些属性。

显示通道示例

```
DISPLAY CHANNEL(QM1.TO.QM2) TRPTYPE,CONVERT

DISPLAY CHANNEL(QM1.TO.*) TRPTYPE,CONVERT

DISPLAY CHANNEL(*) TRPTYPE,CONVERT

DISPLAY CHANNEL(QM1.TO.QMR34) ALL
```

ALW 显示通道状态

使用 MQSC 命令 DISPLAY CHSTATUS ，指定通道名称以及您是希望通道的当前状态还是已保存信息的状态。

DISPLAY CHSTATUS 适用于所有消息通道。 它不适用于除服务器连接通道以外的 MQI 通道。

显示的信息包括:

- 通道名称
- 通信连接名称
- 通道的不确定状态 (如果适用)
- 上一个序号

- 传输队列名称 (如果适用)
- 不确定标识 (如果适用)
- 上次落实的序号
- 逻辑工作单元标识
- 进程标识
- Windows 线程标识 (仅限 Windows)

显示通道状态示例

```
DISPLAY CHSTATUS(*) CURRENT
DISPLAY CHSTATUS(QM1.TO.*) SAVED
```

在通道上至少传输了一批消息之后，保存的状态才适用。 当通道停止 (使用 STOP CHL 命令) 和队列管理器结束时，也会保存状态。

ALW 使用 *Ping* 检查链接

使用 MQSC 命令 PING CHANNEL 与远程端交换固定数据消息。

Ping 使系统主管确信链接可用且正常运行。

Ping 不涉及使用传输队列和目标队列。 它使用通道定义，相关通信链路和网络设置。 仅当通道当前未处于活动状态时，才能使用此参数。

它仅可用于发送方，服务器和集群发送方通道。 在链路的远端启动相应的通道，并执行启动参数协商。 将正常通知错误。

消息交换的结果显示为 `Ping complete` 或错误消息。

使用 LU 6.2 执行 ping 操作

调用 Ping 时，缺省情况下没有用户标识或密码流向接收端。 如果需要用户标识和密码，那么可以在通道定义中的起始端创建用户标识和密码。 如果在通道定义中输入了密码，那么在保存之前将由 IBM MQ 对其进行加密。 然后在流经对话之前对其进行解密。

ALW 启动通道

对发送方，服务器和请求者通道使用 MQSC 命令 START CHANNEL。 要使应用程序能够交换消息，必须为入站连接启动侦听器程序。

如果在队列管理器触发的情况下设置了通道，那么不需要 START CHANNEL。

启动时，发送 MCA 将读取通道定义并打开传输队列。 发出通道启动序列，该序列远程启动接收方或服务器通道的相应 MCA。 当它们已启动时，发送方和服务器将处理到达传输队列的等待消息，并在它们到达时进行传输。

当您使用触发或运行通道作为线程时，请确保通道启动程序可用于监视启动队列。 缺省情况下，通道启动程序作为队列管理器的一部分启动。

但是， TCP 和 LU 6.2 提供了其他功能:

- Linux AIX 对于 AIX and Linux 上的 TCP ，可以将 inetd 配置为启动通道。 inetd 作为单独的进程启动。
- Linux AIX 对于 AIX and Linux 中的 LU 6.2 ，配置 SNA 产品以启动 LU 6.2 响应程序进程。
- Windows 对于 Windows 中的 LU 6.2 ，使用 SNA 服务器可以使用 TpStart (随 SNA 服务器提供的实用程序) 来启动通道。 TpStart 作为单独的进程启动。

使用 "启动" 选项始终会导致通道在必要时再同步。

要获得成功，请执行以下操作:

- 必须存在本地和远程通道定义。 如果接收方或服务器连接通道没有相应的通道定义，那么如果通道是自动定义的，那么将自动创建缺省通道定义。 请参阅 通道自动定义出口程序 (Channel auto-definition exit program)。
- 传输队列必须存在，并且没有其他使用它的通道。
- 必须存在本地和远程 MCA。
- 通信链路必须可用。
- 队列管理器必须正在运行，本地和远程。
- 消息通道不能已在运行。

将向屏幕返回一条消息，确认已接受启动通道的请求。 要确认启动命令已成功，请检查错误日志或使用 DISPLAY CHSTATUS。 错误日志包括:

Windows **Windows**

MQ_DATA_PATH\qmgrs\qmname\errors\AMQERR01.LOG (对于名为 qmname 的每个队列管理器)

MQ_DATA_PATH\qmgrs\@SYSTEM\errors\AMQERR01.LOG (对于一般错误)

MQ_DATA_PATH 表示安装 IBM MQ 的高级目录。

注: 在 Windows 上，您还会在 Windows 系统应用程序事件日志中获取消息。

Linux AIX **AIX and Linux**

/var/mqm/qmgrs/qmname/errors/AMQERR01.LOG (对于名为 qmname 的每个队列管理器)

/var/mqm/qmgrs/@SYSTEM/errors/AMQERR01.LOG (对于一般错误)

在 AIX, Linux, and Windows 上，使用 **runmqlsr** 命令来启动 IBM MQ 侦听器进程。 缺省情况下，针对通道连接的任何入站请求都会导致侦听器进程启动 MCA 作为 amqrmppa 进程的线程。

```
runmqlsr -t tcp -m QM2
```

对于出站连接，必须通过以下三种方式之一启动通道:

1. 根据 MCATYPE 参数，使用 MQSC 命令 START CHANNEL (指定通道名称) 将通道作为进程或线程启动。 (如果通道作为线程启动，那么它们是通道启动程序的线程。)

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

2. 使用控制命令 runmqchl 作为进程启动通道。

   ```
   runmqchl -c QM1.TO.QM2 -m QM1
   ```

3. 使用通道启动程序来触发通道。

ALW 停止通道

使用 MQSC 命令 STOP CHANNEL 来请求通道停止活动。 在操作员再次启动通道之前，通道不会启动新消息批次。

有关重新启动已停止的通道的信息，请参阅 第 188 页的『重新启动已停止的通道』。

可以向除 MQCHT_CLNTCONN 以外的任何类型的通道发出此命令。

您可以选择所需的停止类型:

停止停顿示例

```
STOP CHANNEL(QM1.TO.QM2) MODE(QUIESCE)
```

此命令请求通道以有序方式关闭。 完成当前的消息批处理，并与通道的另一端执行同步点过程。 如果通道空闲，那么此命令不会终止接收通道。

Stop force 示例

```
STOP CHANNEL(QM1.TO.QM2) MODE(FORCE)
```

此选项会立即停止通道，但不会终止通道的线程或进程。 通道未完成当前消息批次的处理，因此可能会使通道处于不确定状态。 通常，请考虑使用停顿停止选项。

Stop terminate 示例

```
STOP CHANNEL(QM1.TO.QM2) MODE(TERMINATE)
```

此选项会立即停止通道，并终止通道的线程或进程。

Stop (quiesce) stopped 示例

```
STOP CHANNEL(QM1.TO.QM2) STATUS(STOPPED)
```

此命令未指定 MODE，因此缺省为 MODE (QUIESCE)。 它请求停止通道，以使其无法自动重新启动，但必须手动启动。

Stop (quiesce) inactive 示例

```
STOP CHANNEL(QM1.TO.QM2) STATUS(INACTIVE)
```

此命令未指定 MODE，因此缺省为 MODE (QUIESCE)。 它请求使通道处于不活动状态，以便在需要时自动重新启动。

ALW 重命名通道

使用 MQSC 重命名消息通道。

使用 MQSC 执行以下步骤:

1. 使用 STOP CHANNEL 来停止通道。
2. 使用 DEFINE CHANNEL 创建具有新名称的重复通道定义。
3. 使用 DISPLAY CHANNEL 来检查是否已正确创建该通道。
4. 使用 DELETE CHANNEL 删除原始通道定义。

如果您决定重命名消息通道，请记住，一个通道有两个通道定义，每一端都有一个通道定义。 确保同时在两端重命名通道。

ALW 重置通道

使用 MQSC 命令 RESET CHANNEL 来更改消息序号。

RESET CHANNEL 命令可用于任何消息通道，但不适用于 MQI 通道 (client-connection 或 server-connection)。 第一条消息在下次启动通道时启动新序列。

如果在发送方或服务器通道上发出该命令，那么它会在通道重新启动时通知另一端更改。

相关概念

第 200 页的『对象入门』
必须先定义通道，并且它们的关联对象必须存在并且可供使用，然后才能启动通道。 本节将向您展示如何执行此操作。

第 178 页的『通道控制功能』
通道控制功能为您提供用于定义，监视和控制通道的工具。

相关任务

第 152 页的『配置分布式队列』
此部分提供有关 IBM MQ 安装之间的相互通信的更详细信息，包括队列定义，通道定义，触发和同步点过程

相关参考

重置通道

ALW 在通道上解析不确定消息

当消息被发送方或服务器挂起时，使用 MQSC 命令 RESOLVE CHANNEL。 例如，因为链接的一端已终止，并且没有其恢复的前景。

RESOLVE CHANNEL 命令接受两个参数之一 :BACKOUT 或 COMMIT。 回退会将消息复原到传输队列，而 "落实" 会废弃这些消息。

通道程序不会尝试与合作伙伴建立会话。 而是确定表示不确定消息的逻辑工作单元标识 (LUWID)。 然后，根据请求发出以下问题:

- BACKOUT 以将消息复原到传输队列; 或
- 用于从传输队列中删除消息的 COMMIT。

要使决议成功:

- 通道必须处于不活动状态
- 通道必须处于不确定状态
- 通道类型必须是发送方，服务器或集群发送方
- 本地通道定义必须存在
- 本地队列管理器必须正在运行

相关概念

第 200 页的『对象入门』
必须先定义通道，并且它们的关联对象必须存在并且可供使用，然后才能启动通道。 本节将向您展示如何执行此操作。

第 178 页的『通道控制功能』
通道控制功能为您提供用于定义，监视和控制通道的工具。

相关任务

第 152 页的『配置分布式队列』
此部分提供有关 IBM MQ 安装之间的相互通信的更详细信息，包括队列定义，通道定义，触发和同步点过程

相关参考

解析通道

Windows 在 Windows 上设置通信

当启动分布式排队管理通道时，它会尝试使用通道定义中指定的连接。 要成功执行此操作，必须定义连接并且该连接可用。 本节说明如何使用可用于 IBM MQ for Windows 系统的通信形式来执行此操作。

开始之前

您可能会发现，请参阅 示例配置- IBM MQ for Windows。

CD MQ Adv. V 9.2.0 使用 TCP/IP 的消息通道可以指向 IBM Aspera fasp.io Gateway，它提供了可显着提高网络吞吐量的快速 TCP/IP 隧道。 在任何授权平台上运行的队列管理器都可以通过 Aspera gateway 进行连接。 网关本身部署在 Red Hat 或 Ubuntu Linux 或 Windows 上。 请参阅 在 Linux 或 Windows 上定义 Aspera gateway 连接。

关于此任务

在 Windows 上为 IBM MQ 设置通信时，可以从以下类型的通信中进行选择:

- TCP/IP
- LU 6.2
- NetBIOS

过程

- 有关为 Windows 系统设置通信的信息，请参阅所选通信类型的子主题:
  - 第 206 页的『在 Windows 上定义 TCP 连接』
  - 第 208 页的『在 Windows 上定义 LU 6.2 连接』
  - 第 209 页的『在 Windows 上定义 NetBIOS 连接』

  在使用除 TCP/IP 以外的通信协议的环境中，并非 IBM MQ for Windows 的所有功能和设施都可用。 不可用的项为 IBM MQ Explorer。

相关任务

第 197 页的『监视和控制 AIX, Linux, and Windows 上的通道』
对于 DQM，您需要创建，监视和控制到远程队列管理器的通道。 您可以使用命令，程序， IBM MQ Explorer，通道定义的文件以及同步信息的存储区来控制通道。

第 13 页的『配置客户机与服务器之间的连接』
要配置 IBM MQ MQI clients 与服务器之间的通信链路，请决定通信协议，定义链路两端的连接，启动侦听器以及定义通道。

第 211 页的『在 AIX and Linux 上设置通信』
当启动分布式排队管理通道时，它会尝试使用通道定义中指定的连接。 要成功执行此操作，必须定义连接并且该连接可用。 本节说明如何使用可用于 IBM MQ for UNIX or Linux 系统的通信形式来执行此操作。

相关参考

第 14 页的『要使用的通信类型』
不同的平台支持不同的通信协议。 您选择的传输协议取决于 IBM MQ MQI client 和服务器平台的组合。

Windows 在 *Windows* 上定义 *TCP* 连接

通过在发送端配置通道以指定目标地址，并通过在接收端运行侦听器程序来定义 TCP 连接。

开始之前

CD MQ Adv. V 9.2.0 使用 TCP/IP 的消息通道可以指向 IBM Aspera fasp.io Gateway，它提供了可显着提高网络吞吐量的快速 TCP/IP 隧道。 在任何授权平台上运行的队列管理器都可以通过 Aspera gateway 进行连接。 网关本身部署在 Red Hat 或 Ubuntu Linux 或 Windows 上。 请参阅 在 Linux 或 Windows 上定义 Aspera gateway 连接。

发送结束

在通道定义的 "连接名称" 字段中指定目标机器的主机名或 TCP 地址。

要连接到的端口缺省为 1414。 端口号 1414 由因特网分配号码管理局分配给 IBM MQ。

要使用非缺省端口号，请在通道对象定义的连接名称字段中指定该端口号，因此:

```
DEFINE CHANNEL('channel name') CHLTYPE(SDR) +
          TRPTYPE(TCP) +
          CONNAME('OS2ROG3(1822)') +
          XMITQ('XMitQ name') +
          REPLACE
```

其中， OS2ROG3 是远程队列管理器的 DNS 名称， 1822 是所需的端口。 (这必须是接收端的侦听器正在侦听的端口。)

必须停止并重新启动正在运行的通道，才能获取对通道对象定义的任何更改。

您可以通过在 IBM MQ for Windows 的 `.ini` 文件中指定缺省端口号来更改该端口号:

```
TCP:
Port=1822
```

注: 要选择要使用的 TCP/IP 端口号， IBM MQ 将使用它按以下顺序找到的第一个端口号:

1. 在通道定义或命令行中显式指定的端口号。 此数字允许覆盖通道的缺省端口号。
2. `.ini` 文件的 TCP 节中指定的端口属性。 此数字允许覆盖队列管理器的缺省端口号。
3. 缺省值 1414。 这是因特网号码分配机构为入站和出站连接分配给 IBM MQ 的号码。

有关使用 qm.ini 设置的值的更多信息，请参阅 分布式排队的配置文件节。

在 TCP 上接收

要启动接收通道程序，必须启动侦听器程序以检测入局网络请求并启动关联通道。 您可以使用 IBM MQ 侦听器。

接收通道程序是响应来自发送通道的启动请求而启动的。

要启动接收通道程序，必须启动侦听器程序以检测入局网络请求并启动关联通道。 您可以使用 IBM MQ 侦听器。

要运行 IBM MQ 随附的侦听器 (将新通道作为线程启动) ， 请使用 runmqlsr 命令。

使用 **runmqlsr** 命令的基本示例:

```
runmqlsr -t tcp [-m QMNAME] [-p 1822]
```

方括号指示可选参数; 缺省队列管理器不需要 `QMNAME` ， 如果使用缺省值 (1414) ， 那么不需要端口号。 此端口号不能超过 65535。

注: 要选择要使用的 TCP/IP 端口号， IBM MQ 将使用它按以下顺序找到的第一个端口号:

1. 在通道定义或命令行中显式指定的端口号。 此数字允许覆盖通道的缺省端口号。
2. `.ini` 文件的 TCP 节中指定的端口属性。 此数字允许覆盖队列管理器的缺省端口号。
3. 缺省值 1414。 这是因特网号码分配机构为入站和出站连接分配给 IBM MQ 的号码。

为了获得最佳性能，请将 IBM MQ 侦听器作为可信应用程序运行，如 第 196 页的『将通道和侦听器作为可信应用程序运行』中所述。 请参阅 可信应用程序的限制 ，以获取有关可信应用程序的信息

使用 TCP/IP SO_KEEPALIVE 选项

如果要使用 Windows SO_KEEPALIVE 选项，那么必须向注册表添加以下条目:

```
TCP:
KeepAlive=yes
```

有关 SO_KEEPALIVE 选项的更多信息，请参阅 第 185 页的『正在检查通道的另一端是否仍然可用』。

在 "Windows 中， "`HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters` 注册表中的 "Windows KeepAliveTime 选项的值控制着检查连接前的时间间隔。 缺省值为两小时。

使用 TCP 侦听器储备选项

在 TCP 中，除非在服务器和客户机之间进行三方握手，否则将处理不完整的连接。 这些连接称为未完成的连接请求。 为这些未完成的连接请求设置了最大值，可以将其视为在 TCP 端口上等待侦听器接受请求的积压请求。

请参阅 第 214 页的『在 IBM MQ for Multiplatforms 上使用 TCP 侦听器储备选项』 以获取更多信息以及 Windows 的特定值。

Windows 在 *Windows* 上定义 *LU 6.2* 连接

必须配置 SNA ， 以便可以在两台机器之间建立 LU 6.2 对话。

配置 SNA 后，继续如下操作。

请参阅下表以获取信息。

表 *23:* 远程队列管理器平台的本地 *Windows* 系统上的设置

远程平台	TPNAME	主题方案路径
z/OS 或 MVS/ESA 不带 CICS	与有关远程队列管理器的相应辅助信息中的相同。	-
z/OS 或 MVS/ESA 使用 CICS	CKRC (发送方) CKSV (请求者) CKRC (服务器)	-
IBM i	与 IBM i 系统上的路由条目中的比较值相同。	-
AIX and Linux 系统	与有关远程队列管理器的相应辅助信息中的相同。	*MQ_INSTALLATION_PATH*/bin/amqcrs6a
Windows	如 Windows Run Listener 命令或使用 Windows 上的 TpSetup 定义的可调用事务程序中所指定。	*MQ_INSTALLATION_PATH*\bin\amqcrs6a

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

如果同一机器上有多个队列管理器，请确保通道定义中的 TPname 是唯一的。

有关配置 AnyNet SNA over TCP/IP 的最新信息，请参阅以下联机 IBM 文档: AnyNet SNA over TCP/IP 和 SNA Node Operations。

相关概念

第 208 页的『在 Windows 上的 LU 6.2 上发送端』
从您正在使用的 LU 6.2 产品的管理应用程序创建 CPI-C 端对象 (符号目标)。 在通道定义的 "连接名称" 字段中输入此名称。 同时创建指向合作伙伴的 LU 6.2 链接。

第 208 页的『在 Windows 上的 LU 6.2 上接收』
接收通道程序是响应来自发送通道的启动请求而启动的。

Windows 在 *Windows* 上的 *LU 6.2* 上发送端

从您正在使用的 LU 6.2 产品的管理应用程序创建 CPI-C 端对象 (符号目标)。 在通道定义的 "连接名称" 字段中输入此名称。 同时创建指向合作伙伴的 LU 6.2 链接。

在 CPI-C 端对象中，输入接收机器上的伙伴 LU 名， TP 名和方式名。 例如:

```
Partner LU Name              OS2ROG2
Partner TP Name              recv
Mode Name                    #INTER
```

Windows 在 *Windows* 上的 *LU 6.2* 上接收

接收通道程序是响应来自发送通道的启动请求而启动的。

要启动接收通道程序，必须启动侦听器程序以检测入局网络请求并启动关联的通道。 使用 RUNMQLSR 命令启动此侦听器程序，并提供要侦听的 TpName 。 或者，可以在 SNA Server for Windows 下使用 TpStart 。

使用 RUNMQLSR 命令

用于启动侦听器的命令示例:

```
RUNMQLSR -t LU62 -n RECV -m QMNAME
```

其中 "RECV 是另一端（发送端）指定的 TpName，即 "远程端启动的"TpName"。 此命令的最后一部分中使用的 **-m** 参数是可选的，并且对于缺省队列管理器不是必需的。

可以在一台机器上运行多个队列管理器。 必须为每个队列管理器分配不同的 TpName ，然后为每个队列管理器启动一个侦听器程序。 例如：

```
RUNMQLSR -t LU62 -m QM1 -n TpName1
RUNMQLSR -t LU62 -m QM2 -n TpName2
```

为了获得最佳性能，请将 IBM MQ 侦听器作为可信应用程序运行，如 将通道和侦听器作为可信应用程序运行中所述。 请参阅 可信应用程序的限制 ，以获取有关可信应用程序的信息。

您可以使用以下命令停止在处于不活动状态的队列管理器上运行的所有 IBM MQ 侦听器:

```
ENDMQLSR -m QMNAME
```

在 Windows 上使用 Microsoft SNA Server

您可以使用 TpSetup (来自 SNA Server SDK) 来定义可调用 TP ，然后驱动 amqcrs6a.exe，也可以手动设置各种注册表值。 应传递到 amqcrs6a.exe 的参数为:

```
-m QM -n TpName
```

其中 *QM* 是队列管理器名称， *TpName* 是 TP 名称。 有关更多信息，请参阅 *Microsoft SNA Server APPC Programmers Guide* 或 *Microsoft SNA Server CPI-C Programmers Guide* 。

如果未指定队列管理器名称，那么将采用缺省队列管理器。

Windows 在 *Windows* 上定义 *NetBIOS* 连接

NetBIOS 连接仅适用于运行 Windows 的客户机和服务器。 IBM MQ 在与其他 IBM MQ 产品建立 NetBIOS 连接时使用三种类型的 NetBIOS 资源: 会话，命令和名称。 其中每个资源都有一个限制，缺省情况下或在安装 NetBIOS 期间通过选择来建立该限制。

每个正在运行的通道 (无论其类型如何) 都使用一个 NetBIOS 会话和一个 NetBIOS 命令。 IBM NetBIOS 实现允许多个进程使用相同的本地 NetBIOS 名称。 因此，只有一个 NetBIOS 名称需要可供 IBM MQ 使用。 其他供应商的实现 (例如 Novell 的 NetBIOS 仿真) 要求每个进程具有不同的本地名称。 从您正在使用的 NetBIOS 产品的文档中验证您的需求。

在所有情况下，请确保每种类型的足够资源都已可用，或者增加配置中指定的最大值。 对值所作的任何更改都需要系统重新启动。

在系统启动期间， NetBIOS 设备驱动程序显示可供应用程序使用的会话数，命令数和名称。 这些资源可用于在同一系统上运行的任何基于 NetBIOS 的应用程序。 因此，在 IBM MQ 需要获取这些资源之前，其他应用程序可能会使用这些资源。 您的 LAN 网络管理员应该能够为您阐明这一点。

相关概念

第 210 页的『定义 IBM MQ 本地 NetBIOS 名称』
可以通过三种方式指定 IBM MQ 通道进程所使用的本地 NetBIOS 名称。

第 210 页的『建立队列管理器 NetBIOS 会话，命令和名称限制』
可以通过两种方式指定 NetBIOS 会话，命令和名称的队列管理器限制。

第 210 页的『建立 LAN 适配器号』
要使通道成功跨 "NetBIOS,工作，两端的适配器支持必须兼容。 IBM MQ 允许您使用 qm.ini 文件的 NETBIOS 节中的 AdapterNum 值并通过在 runmqlsr 命令上指定 **-a** 参数来控制 LAN 适配器 (LANA) 编号的选择。

第 211 页的『启动 NetBIOS 连接』
定义启动连接所需的步骤。

第 211 页的『定义 NetBIOS 连接的目标侦听器』
定义要在 NetBIOS 连接的接收端执行的步骤。

Windows 定义 *IBM MQ* 本地 *NetBIOS* 名称

可以通过三种方式指定 IBM MQ 通道进程所使用的本地 NetBIOS 名称。

按优先顺序，这三种方法是:

1. RUNMQLSR 命令的 **-l** 参数中指定的值，例如:

```
RUNMQLSR -t NETBIOS -l my_station
```

2. 具有由命令建立的值的 MQNAME 环境变量:

```
SET MQNAME= my_station
```

   您可以为每个进程设置 MQNAME 值。 或者，可以在 Windows 注册表中的系统级别对其进行设置。

   如果您正在使用需要唯一名称的 NetBIOS 实现，那么必须在启动 IBM MQ 进程的每个窗口中发出 SET MQNAME 命令。 MQNAME 值是任意的，但对于每个进程必须是唯一的。

3. 队列管理器配置文件 qm.ini 中的 NETBIOS 节。 例如:

```
NETBIOS:

LocalName= my_station
```

注:

1. 由于支持的 NetBIOS 产品实现的变体，建议您使每个 NetBIOS 名称在网络中唯一。 否则，可能会出现不可预测的结果。 如果您在建立 NetBIOS 通道时迂到问题，并且队列管理器错误日志中存在显示 NetBIOS 返回码 X'15 ' 的错误消息，请查看您是否使用了 NetBIOS 名称。
2. 在 Windows 上，不能将机器名称用作 NetBIOS 名称，因为 Windows 已使用该名称。
3. 发送方通道启动要求使用 MQNAME 环境变量或 qm.ini 文件中的 LocalName 指定 NetBIOS 名称。

Windows 建立队列管理器 *NetBIOS* 会话，命令和名称限制

可以通过两种方式指定 NetBIOS 会话，命令和名称的队列管理器限制。

按照优先顺序，这些方法是:

1. RUNMQLSR 命令中指定的值:

```
-s Sessions
-e Names
-o Commands
```

   如果命令中没有指定-m 操作数，则这些值仅适用于默认队列管理器。

2. 队列管理器配置文件 qm.ini 中的 NETBIOS 节。 例如:

```
NETBIOS:

NumSess= Qmgr_max_sess
NumCmds= Qmgr_max_cmds
NumNames= Qmgr_max_names
```

Windows 建立 *LAN* 适配器号

要使通道成功跨 "NetBIOS,工作，两端的适配器支持必须兼容。 IBM MQ 允许您使用 qm.ini 文件的 NETBIOS 节中的 AdapterNum 值并通过在 runmqlsr 命令上指定 **-a** 参数来控制 LAN 适配器 (LANA) 编号的选择。

IBM MQ 用于 NetBIOS 连接的缺省 LAN 适配器号为 0。 验证系统上正在使用的数字，如下所示:

在 Windows 上，无法通过操作系统直接查询 LAN 适配器号。 而是使用 LANACFG.EXE 命令行实用程序，可从 Microsoft 获取。 该工具的输出显示虚拟 LAN 适配器号及其有效绑定。 有关 LAN 适配器编号的更多信息，请参阅 Microsoft 知识库文章 138037 *HOWTO:* 在 *32* 位环境中使用 *LANA* 编号。

在队列管理器配置文件 qm.ini: 的 NETBIOS 节中指定正确的值

```
NETBIOS:
AdapterNum= n
```

其中 n 是此系统的正确 LAN 适配器号。

Windows 启动 *NetBIOS* 连接

定义启动连接所需的步骤。

要启动连接，请在发送端执行以下步骤:

1. 使用 MQNAME 或 LocalName 值定义 NetBIOS 站名。
2. 验证系统上正在使用的 LAN 适配器号，并使用 AdapterNum 指定正确的文件。
3. 在通道定义的 ConnectionName 字段中，指定目标侦听器程序正在使用的 NetBIOS 名称。 在 Windows 上， NetBIOS 通道必须作为线程运行。 通过在通道定义中指定 MCATYPE (THREAD) 来执行此操作。

```
DEFINE CHANNEL (chname) CHLTYPE(SDR) +
TRPTYPE(NETBIOS) +
CONNAME(your_station) +
XMITQ(xmitq) +
MCATYPE(THREAD) +
REPLACE
```

Windows 定义 *NetBIOS* 连接的目标侦听器

定义要在 NetBIOS 连接的接收端执行的步骤。

在接收端，请执行以下步骤:

1. 使用 MQNAME 或 LocalName 值定义 NetBIOS 站名。
2. 验证系统上正在使用的 LAN 适配器号，并使用 AdapterNum 指定正确的文件。
3. 定义接收方通道:

```
DEFINE CHANNEL (chname) CHLTYPE(RCVR) +
TRPTYPE(NETBIOS) +
REPLACE
```

4. 启动 IBM MQ 侦听器程序以建立站并使其能够与之联系。 例如:

```
RUNMQLSR -t NETBIOS -l your_station [-m qmgr]
```

此命令将 `your_station` 建立为等待联系的 NetBIOS 站。 NetBIOS 站名在整个 NetBIOS 网络中必须唯一。

为了获得最佳性能，请将 IBM MQ 侦听器作为可信应用程序运行，如 第 196 页的『将通道和侦听器作为可信应用程序运行』中所述。 请参阅 可信应用程序的限制，以获取有关可信应用程序的信息。

您可以使用以下命令停止在处于不活动状态的队列管理器上运行的所有 IBM MQ 侦听器:

```
ENDMQLSR [-m QMNAME]
```

如果未指定队列管理器名称，那么将采用缺省队列管理器。

Linux AIX 在 AIX and Linux 上设置通信

当启动分布式排队管理通道时，它会尝试使用通道定义中指定的连接。 要成功执行此操作，必须定义连接并且该连接可用。 本节说明如何使用可用于 IBM MQ for UNIX or Linux 系统的通信形式来执行此操作。

开始之前

您可能会发现参考以下部分很有用:

- AIX 示例配置- IBM MQ for AIX
- Linux 示例配置- IBM MQ for Linux

CD MQ Adv. V 9.2.0使用 TCP/IP 的消息通道可以指向 IBM Aspera fasp.io Gateway，它提供了可显着提高网络吞吐量的快速 TCP/IP 隧道。 在任何授权平台上运行的队列管理器都可以通过 Aspera gateway 进行连接。 网关本身部署在 Red Hat 或 Ubuntu Linux 或 Windows 上。 请参阅 在 Linux 或 Windows 上定义 Aspera gateway 连接。

关于此任务

当启动分布式排队管理通道时，它会尝试使用通道定义中指定的连接。 要成功，需要定义连接并使其可用。本节说明如何执行此操作。

在 AIX and Linux 上为 IBM MQ 设置通信时，可以从以下类型的通信中进行选择:

- TCP/IP
- LU 6.2

每个通道定义必须仅指定一个作为传输协议 (传输类型) 属性。 队列管理器可以使用一个或多个协议。

对于 IBM MQ MQI clients，具有使用不同传输协议的备用通道可能很有用。 有关 IBM MQ MQI clients 的更多信息，请参阅 IBM MQ MQI clients 概述。

过程

有关为 AIX 或 Linux 系统设置通信的信息，请参阅所选通信类型的子主题:

- 第 212 页的『在 AIX and Linux 上定义 TCP 连接』
- 第 215 页的『在 AIX and Linux 上定义 LU 6.2 连接』
- MQ Adv. MQ Adv. VUE V 9.2.0第 694 页的『在 Linux 或 Windows 上定义 Aspera gateway 连接』

相关任务

第 197 页的『监视和控制 AIX, Linux, and Windows 上的通道』
对于 DQM，您需要创建，监视和控制到远程队列管理器的通道。 您可以使用命令，程序， IBM MQ Explorer，通道定义的文件以及同步信息的存储区来控制通道。

第 13 页的『配置客户机与服务器之间的连接』
要配置 IBM MQ MQI clients 与服务器之间的通信链路，请决定通信协议，定义链路两端的连接，启动侦听器以及定义通道。

第 205 页的『在 Windows 上设置通信』
当启动分布式排队管理通道时，它会尝试使用通道定义中指定的连接。 要成功执行此操作，必须定义连接并且该连接可用。 本节说明如何使用可用于 IBM MQ for Windows 系统的通信形式来执行此操作。

相关参考

第 14 页的『要使用的通信类型』
不同的平台支持不同的通信协议。 您选择的传输协议取决于 IBM MQ MQI client 和服务器平台的组合。

Linux AIX 在 *AIX and Linux* 上定义 *TCP* 连接

发送端的通道定义指定目标的地址。 在接收端为连接配置了侦听器或 inet 守护程序。

开始之前

CD MQ Adv. V 9.2.0使用 TCP/IP 的消息通道可以指向 IBM Aspera fasp.io Gateway，它提供了可显着提高网络吞吐量的快速 TCP/IP 隧道。 在任何授权平台上运行的队列管理器都可以通过 Aspera gateway 进行连接。 网关本身部署在 Red Hat 或 Ubuntu Linux 或 Windows 上。 请参阅 在 Linux 或 Windows 上定义 Aspera gateway 连接。

发送结束

在通道定义的 "连接名称" 字段中指定主机名或目标机器的 TCP 地址。 要连接到的端口缺省为 1414。 端口号 1414 由因特网分配号码管理局分配给 IBM MQ。

要使用非缺省端口号，请更改连接名称字段，以便:

```
Connection Name REMHOST(1822)
```

其中 REMHOST 是远程机器的主机名， 1822 是所需的端口号。 (这必须是接收端的侦听器正在侦听的端口。)

或者，可以通过在队列管理器配置文件 (qm.ini) 中指定端口号来更改端口号:

```
TCP:
Port=1822
```

有关使用 qm.ini 设置的值的更多信息，请参阅 分布式排队的配置文件节。

在 TCP 上接收

您可以使用 TCP/IP 侦听器 (即 inet 守护程序 (inetd)) 或 IBM MQ 侦听器。

现在，某些 Linux 分发版使用扩展的 inet 守护程序 (xinetd) 而不是 inet 守护程序。 有关如何在 Linux 系统上使用扩展 inet 守护程序的信息，请参阅 在 Linux 上建立 TCP 连接 。

相关概念

第 213 页的『在 AIX and Linux 上使用 TCP/IP 侦听器』
要在 AIX and Linux 上启动通道，必须编辑 `/etc/services` 文件和 `inetd.conf` 文件

第 214 页的『在 IBM MQ for Multiplatforms 上使用 TCP 侦听器储备选项』
在 TCP 中，除非在服务器和客户机之间进行三方握手，否则将处理不完整的连接。 这些连接称为未完成的连接请求。 为这些未完成的连接请求设置了最大值，可以将其视为在 TCP 端口上等待侦听器接受请求的积压请求。

第 215 页的『使用 IBM MQ 侦听器』
要运行随 IBM MQ 提供的侦听器 (将新通道作为线程启动) ，请使用 `runmqlsr` 命令。

第 215 页的『使用 TCP/IP SO_KEEPALIVE 选项』
在某些 AIX and Linux 系统上，您可以定义 TCP 在检查连接是否仍然可用之前等待的时间长度，以及在第一次检查失败时再次尝试连接的频率。 这是内核可调参数，或者可以在命令行中输入。

Linux AIX 在 *AIX and Linux* 上使用 *TCP/IP* 侦听器

要在 AIX and Linux 上启动通道，必须编辑 `/etc/services` 文件和 `inetd.conf` 文件

请遵循以下指示信息:

1. 编辑 `/etc/services` 文件:

   注: 要编辑 `/etc/services` 文件，您必须以超级用户或 root 用户身份登录。 您可以更改此值，但它必须与发送端指定的端口号匹配。

   将以下行添加到该文件:

   ```
   MQSeries 1414/tcp
   ```

   其中 1414 是 IBM MQ 所需的端口号。 此端口号不能超过 65535。
2. 在 `inetd.conf` 文件中添加一行以调用程序 amqcrsta ，其中 *MQ_INSTALLATION_PATH* 表示安装了 IBM MQ 的高级目录:

   ```
   MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta
   [-m Queue_Man_Name]
   ```

在 inetd 重新读取配置文件后，更新处于活动状态。 要执行此操作，请从 root 用户标识发出以下命令:

- AIX 在 AIX 上:

  ```
  refresh -s inetd
  ```

- Linux AIX 在其他 AIX and Linux 系统上:

```
kill -1 process_number
```

当 inetd 启动的侦听器程序从 inetd 继承语言环境时，可能不采用 (合并) MQMDE 并将其作为消息数据放在队列上。 要确保遵守 MQMDE，必须正确设置语言环境。 inetd 设置的语言环境可能与为 IBM MQ 进程使用的其他语言环境选择的语言环境不匹配。 要设置语言环境:

1. 创建一个 shell 脚本，该脚本将语言环境变量 LANG， LC_COLLATE， LC_CTYPE，LC_货币，LC_NUMERIC， LC_TIME 和 LC_MESSAGES 设置为用于其他 IBM MQ 进程的语言环境。
2. 在同一 shell 脚本中，调用侦听器程序。
3. 修改 `inetd.conf` 文件以调用 shell 脚本来代替侦听器程序。

可以在服务器上具有多个队列管理器。 对于每个队列管理器，必须向这两个文件中的每个文件添加一行。 例如:

```
MQSeries1   1414/tcp
MQSeries2   1822/tcp
```

```
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
```

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

如果在单个 TCP 端口上排队的未完成连接请求数存在限制，那么这将避免生成错误消息。 有关未完成的连接请求数的信息，请参阅 第 214 页的『在 IBM MQ for Multiplatforms 上使用 TCP 侦听器储备选项』。

Multi 在 *IBM MQ for Multiplatforms* 上使用 *TCP* 侦听器储备选项

在 TCP 中，除非在服务器和客户机之间进行三方握手，否则将处理不完整的连接。 这些连接称为未完成的连接请求。 为这些未完成的连接请求设置了最大值，可以将其视为在 TCP 端口上等待侦听器接受请求的积压请求。

缺省侦听器储备值显示在 第 214 页的表 24 中。

表 *24:* 在 *TCP/IP* 端口上排队的最大未完成连接请求数

服务器平台	**最大连接请求数**
AIX AIX	100
Linux Linux	100
IBM i IBM i	255
Windows Windows 服务器	100

如果积压达到 第 214 页的表 24 中显示的值，那么将拒绝 TCP/IP 连接，并且通道无法启动。

对于 MCA 通道，这将导致通道进入 RETRY 状态，并在稍后再次尝试连接。

但是，为了避免此错误，您可以在 `qm.ini` 文件中添加条目:

```
TCP:
ListenerBacklog = n
```

这将覆盖缺省的最大未完成请求数 (请参阅 第 214 页的表 24) 用于 TCP/IP 侦听器。

注: 某些操作系统支持大于缺省值的值。 如果需要，可以使用此值来避免达到连接限制。

要在启用了 **储备** 选项的情况下运行侦听器，请执行以下操作:

- 使用 `runmqlsr -b` 命令，或者
- 使用 MQSC 命令 **DEFINE LISTENER**，将 BACKLOG 属性设置为必需值。

有关 **runmqlsr** 命令的信息，请参阅 runmqlsr。 有关 DEFINE LISTENER 命令的信息，请参阅 定义侦听器。

相关概念

第 817 页的『在 z/OS 上使用 TCP 侦听器储备选项』

在 TCP/IP 上接收时，将设置最大未完成的连接请求数。 这些未完成的请求可被视为在 TCP/IP 端口上等待侦听器接受请求的 积压 请求。

Linux AIX 使用 *IBM MQ* 侦听器

要运行随 IBM MQ 提供的侦听器 (将新通道作为线程启动)，请使用 `runmqlsr` 命令。

例如：

```
runmqlsr -t tcp [-m QMNAME] [-p 1822]
```

方括号指示可选参数; 缺省队列管理器不需要 `QMNAME`，如果使用缺省值 (1414)，那么不需要端口号。 此端口号不能超过 65535。

为了获得最佳性能，请将 IBM MQ 侦听器作为可信应用程序运行，如 第 196 页的『将通道和侦听器作为可信应用程序运行』中所述。 请参阅 可信应用程序的限制，以获取有关可信应用程序的信息。

您可以使用以下命令停止在处于不活动状态的队列管理器上运行的所有 IBM MQ 侦听器:

```
endmqlsr [-m QMNAME]
```

如果未指定队列管理器名称，那么将采用缺省队列管理器。

Linux AIX 使用 *TCP/IP SO_KEEPALIVE* 选项

在某些 AIX and Linux 系统上，您可以定义 TCP 在检查连接是否仍然可用之前等待的时间长度，以及在第一次检查失败时再次尝试连接的频率。 这是内核可调参数，或者可以在命令行中输入。

如果要使用 SO_KEEPALIVE 选项 (有关更多信息，请参阅 第 185 页的『正在检查通道的另一端是否仍然可用』) 必须将以下条目添加到队列管理器配置文件 (qm.ini):

```
TCP:
KeepAlive=yes
```

有关更多信息，请参阅 AIX 或 Linux 系统的文档。

Linux AIX 在 *AIX and Linux* 上定义 *LU 6.2* 连接

必须配置 SNA，以便可以在两台机器之间建立 LU 6.2 对话。

有关配置基于 TCP/IP 的 SNA 的最新信息，请参阅以下联机 IBM 文档: Communications Server。

必须配置 SNA，以便可以在两个系统之间建立 LU 6.2 对话。

请参阅 *Multiplatform APPC Configuration Guide* 和下表以获取信息。

表 *25:* 远程队列管理器平台的本地 *AIX* 或 *Linux* 系统上的设置

远程平台	**TPNAME**	**主题方案路径**
不带 CICS 的 z/OS	与远程队列管理器的辅助信息中的相应 TPName 相同。	-
z/OS 使用 CICS	CKRC (发送方) CKSV (请求者) CKRC (服务器)	-
IBM i	与 IBM i 系统上的路由条目中的比较值相同。	-
AIX and Linux 系统	与远程队列管理器的辅助信息中的相应 TPName 相同。	*MQ_INSTALLATION_PATH*/bin/amqcrs6a

表 25: 远程队列管理器平台的本地 *AIX* 或 *Linux* 系统上的设置 *(继续)*		
远程平台	**TPNAME**	**主题方案路径**
Windows	如 Windows Run Listener 命令或使用 Windows 上的 TpSetup 定义的可调用事务程序中所指定。	*MQ_INSTALLATION_PATH*\bin\amqcrs6a

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

如果同一机器上有多个队列管理器，请确保通道定义中的 TPname 是唯一的。

相关概念

第 216 页的『在 AIX and Linux 上的 LU 6.2 上发送端』
在 AIX and Linux 系统上，创建 CPI-C 端对象 (符号目标)，并在通道定义的 "连接名称" 字段中输入此名称。同时创建指向合作伙伴的 LU 6.2 链接。

第 216 页的『在 AIX and Linux 上的 LU 6.2 上接收』
在 AIX and Linux 系统上，在接收端创建侦听连接， LU 6.2 逻辑连接概要文件和 TPN 概要文件。

Linux AIX 在 *AIX and Linux* 上的 *LU 6.2* 上发送端

在 AIX and Linux 系统上，创建 CPI-C 端对象 (符号目标)，并在通道定义的 "连接名称" 字段中输入此名称。同时创建指向合作伙伴的 LU 6.2 链接。

在 CPI-C 端对象中，输入接收机器上的伙伴 LU 名，事务程序名和方式名。 例如：

```
Partner LU Name              REMHOST
Remote TP Name               recv
Service Transaction Program  no
Mode Name                    #INTER
```

当 IBM MQ 尝试建立 SNA 会话时，将在 CPI-C 支持的情况下使用 SECURITY PROGRAM。

Linux AIX 在 *AIX and Linux* 上的 *LU 6.2* 上接收

在 AIX and Linux 系统上，在接收端创建侦听连接， LU 6.2 逻辑连接概要文件和 TPN 概要文件。

在 TPN 概要文件中，输入可执行文件的完整路径和事务程序名称:

```
Full path to TPN executable    MQ_INSTALLATION_PATH/bin/amqcrs6a
Transaction Program name       recv
User ID                        0
```

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

在可以设置用户标识的系统上，指定属于 mqm 组的用户。

AIX 在 AIX 上，设置 APPCTPN (事务名称) 和 APPCLLU (本地 LU 名称) 环境变量 (可以将配置面板用于所调用的事务程序)。

您可能需要使用缺省队列管理器以外的队列管理器。 如果是这样，请定义一个调用以下内容的命令文件:

```
amqcrs6a -m Queue_Man_Name
```

然后调用命令文件。

IBM i 监视和控制 IBM i 上的通道

使用 DQM 命令和面板来创建，监视和控制到远程队列管理器的通道。 每个队列管理器都有一个 DQM 程序，用于控制与兼容远程队列管理器的互连。

关于此任务

以下列表是通道控制功能组件的简要描述:

- 通道定义作为队列管理器对象保留。
- 通道命令是 IBM MQ for IBM i 命令集的子集。

  使用命令 GO CMDMQM 显示完整的 IBM MQ for IBM i 命令集。
- 您可以使用通道定义面板或命令来执行以下操作:
  - 创建，复制，显示，更改和删除通道定义
  - 无法重新建立链接时启动和停止通道， ping ，重置通道序号以及解析不确定消息
  - 显示有关通道的状态信息
- 还可以使用 MQSC 来管理通道
- 还可以使用 IBM MQ Explorer 来管理通道
- 序号和 逻辑工作单元 *(LUW)* 标识存储在同步文件中，并用于通道同步目的。

您可以使用命令和面板来定义消息通道和关联对象，以及监视和控制消息通道。 通过使用 F4=Prompt 键，可以指定相关队列管理器。 如果不使用提示，那么将采用缺省队列管理器。 使用 F4=Prompt 时，将显示一个附加面板，您可以在其中输入相关队列管理器名称，有时还可以输入其他数据。

您需要使用面板定义的对象包括:

- 传输队列
- 远程队列定义
- 队列管理器别名定义
- 应答队列别名定义
- 应答本地队列
- 消息通道定义

有关使用这些对象所涉及的概念的更多信息，请参阅 第 152 页的『配置分布式队列』。

必须完全定义通道，并且它们的关联对象必须存在并且可供使用，然后才能启动通道。

此外，必须先定义每个通道的特定通信链路并使其可用，然后才能运行通道。 有关如何定义 LU 6.2 和 TCP/IP 链路的描述，请参阅安装的特定通信指南。

过程

- 有关创建和使用对象的更多信息，请参阅:
  - 第 218 页的『在 IBM i 上创建对象』
  - 第 218 页的『在 IBM i 上创建通道』
  - 第 220 页的『在 IBM i 上启动通道』
  - 第 220 页的『在 IBM i 上选择通道』
  - 第 221 页的『在 IBM i 上浏览通道』
  - 第 222 页的『在 IBM i 上重命名通道』
  - 第 223 页的『在 IBM i 上使用通道状态』
  - 第 223 页的『IBM i 上的 "使用通道" 选项』

相关概念

第 228 页的『为 IBM i 设置通信』
当启动分布式排队管理通道时，它会尝试使用通道定义中指定的连接。 要使其成功，需要定义连接并使其可用。

相关任务

第 13 页的『配置客户机与服务器之间的连接』
要配置 IBM MQ MQI clients 与服务器之间的通信链路，请决定通信协议，定义链路两端的连接，启动侦听器以及定义通道。

相关参考

示例配置- IBM MQ for IBM i

IBM MQ for IBM i 的消息通道规划示例

IBM MQ for IBM i CL 命令

IBM i 在 IBM i 上创建对象

可以使用 CRTMQMQ 命令来创建队列和别名对象。

您可以创建队列和别名对象，例如: 传输队列，远程队列定义，队列管理器别名定义，应答队列别名定义和应答本地队列。

有关缺省对象的列表，请参阅 系统和缺省对象。

IBM i 在 IBM i 上创建通道

可以从 "创建通道" 面板或在命令行上使用 CRTMQMCHL 命令来创建通道。

要创建通道:

1. 使用 "使用 MQM 通道" 面板 (WRKMQMCHL) 中的 F6 。

   或者，从命令行使用 CRTMQMCHL 命令。

   无论采用哪种方式，都将显示 "创建通道" 面板。 类型:

   - 提供的字段中通道的名称
   - 链路此端的通道类型
2. 按 Enter 键。

注: 您必须唯一地命名网络中的所有通道。 如 显示所有通道的网络图中所示，包括通道名称中的源和目标队列管理器名称是一个很好的方法。

将验证您的条目并立即报告错误。 更正任何错误并继续。

将向您显示所选通道类型的相应通道设置面板。 使用先前收集的信息填写字段。 按 Enter 键以创建通道。

您可以在帮助面板中的通道定义面板的描述以及 通道属性中获得有关确定各个字段的内容的帮助。

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Channel name . . . . . > CHANNAME_____________
Channel type . . . . . > *SDR__    *RCVR, *SDR, *SVR, *RQSTR...
Message Queue Manager name     *DFT_____________________________
_____
Replace . . . . . .  *NO_     *NO, *YES
Transport type . . . . . .  *TCP_____    *LU62, *TCP, *SYSDFTCHL
Text 'description' . . . . . > 'Example Channel Definition'_____________
_____________________________
Connection name . . . . . .  *SYSDFTCHL_____________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
_________________________________________________
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

图 *27:* 创建通道 *(1)*

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Transmission queue . . . . . .  'TRANSMISSION_QUEUE_NAME'_____________________
_____________________
Message channel agent . . . . .  *NONE______  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________  Name
Message channel agent user ID .  *SYSDFTCHL__ Character value...
Coded Character Set Identifier   *SYSDFTCHL__ 0-9999, *SYSDFTCHL
Batch size . . . . . .  50_________  1-9999, *SYSDFTCHL
Disconnect interval . . . . . .  6000_______  1-999999, *SYSDFTCHL
Short retry interval . . . . . .  60_________  0-999999999, *SYSDFTCHL
Short retry count . . . . . .  10_________  0-999999999, *SYSDFTCHL
Long retry interval . . . . . .  1200_______  0-999999999, *SYSDFTCHL
Long retry count . . . . . .  999999999__  0-999999999, *SYSDFTCHL
Security exit . . . . . .  *NONE_____  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________  Name
Security exit user data . . . .  *SYSDFTCHL_____________________

More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

图 *28:* 创建通道 *(2)*

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Send exit . . . . . .  *NONE______  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________ Name
+ for more values   __________
Send exit user data . . . . . .  _____________________________
+ for more values  _____________________________
Receive exit . . . . . .  *NONE_____  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________  Name
+ for more values  __________
__________
Receive exit user data . . . . .  _____________________________
+ for more values  _____________________________
Message exit . . . . . .  *NONE_____  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________  Name
+ for more values  __________

__________
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

图 *29:* 创建通道 *(3)*

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Message exit user data . . . . .  _________________________________
+ for more values  _________________________________
Convert message . . . . . .  *SYSDFTCHL_  *YES, *NO, *SYSDFTCHL
Sequence number wrap . . . . . .  999999999__  100-999999999, *SYSDFTCHL
Maximum message length . . . . .  4194304____  0-4194304, *SYSDFTCHL
Heartbeat interval . . . . . .  300________  0-999999999, *SYSDFTCHL
Non Persistent Message Speed . .  *FAST_____  *FAST, *NORMAL, *SYSDFTCHL
Password . . . . . .  *SYSDFTCHL_  Character value, *BLANK...
Task User Profile . . . . . .  *SYSDFTCHL_  Character value, *BLANK...
Transaction Program Name . . . .  *SYSDFTCHL




Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

图 *30:* 创建通道 *(4)*

IBM i 在 IBM i 上启动通道

您可以从 "使用通道" 面板或通过在命令行上使用 STRMQMCHL 命令来启动通道。

侦听器仅对 TCP 有效。 对于 SNA 侦听器，必须配置通信子系统。

要使应用程序能够交换消息，必须使用 STRMQMLSR 命令为入站连接启动侦听器程序。

对于出站连接，必须通过下列其中一种方式启动通道:

1. 根据 MCATYPE 参数，使用 CL 命令 STRMQMCHL (指定通道名称) 将通道作为进程或线程启动。 (如果通道作为线程启动，那么它们是通道启动程序的线程。)

   ```
   STRMQMCHL CHLNAME(QM1.TO.QM2) MQNAME(MYQMGR)
   ```

2. 使用通道启动程序来触发通道。 当队列管理器启动时，将自动启动一个通道启动程序。 可以通过更改该队列管理器的 qm.ini 文件中的 chinit 节来消除此自动启动。
3. 使用 WRKMQMCHL 命令开始 "使用通道" 面板并选择选项 14 以启动通道。

IBM i 在 IBM i 上选择通道

您可以从 "使用通道" 面板中选择通道。

要选择通道，请使用 WRKMQMCHL 命令从 "使用通道" 面板开始:

1. 将光标移至与所需通道名称关联的选项字段。
2. 输入选项号。
3. 按 Enter 键以激活您的选择。

如果选择多个通道，那么将按顺序激活这些选项。

```
Work with MQM Channels

Queue Manager Name . . :  CNX

Type options, press Enter.
2=Change  3=Copy  4=Delete  5=Display  8=Work with Status  13=Ping
14=Start  15=End  16=Reset  17=Resolve

Opt   Name           Type    Transport   Status
CHLNIC          *RCVR    *TCP     INACTIVE
CORSAIR.TO.MUSTANG    *SDR    *LU62    INACTIVE
FV.CHANNEL.MC.DJE1    *RCVR    *TCP     INACTIVE
FV.CHANNEL.MC.DJE2    *SDR    *TCP     INACTIVE
FV.CHANNEL.MC.DJE3    *RQSTR   *TCP     INACTIVE
FV.CHANNEL.MC.DJE4    *SVR    *TCP     INACTIVE
FV.CHANNEL.PETER     *RCVR    *TCP     INACTIVE
FV.CHANNEL.PETER.LU   *RCVR    *LU62    INACTIVE
FV.CHANNEL.PETER.LU1   *RCVR    *LU62    INACTIVE
More...
Parameters or command
===>
F3=Exit  F4=Prompt  F5=Refresh  F6=Create  F9=Retrieve  F12=Cancel
F21=Print
```

图 *31:* 使用通道

IBM i 在 IBM i 上浏览通道

可以从 "显示通道" 面板浏览通道，也可以在命令行上使用 DSPMQMCHL 命令来浏览通道。

要浏览通道的设置，请使用 WRKMQMCHL 命令从 "显示通道" 面板开始:

1. 针对所需通道名称输入选项 5 (显示)。
2. 按 Enter 键以激活您的选择。

如果选择多个通道，那么将按顺序显示这些通道。

或者，可以从命令行使用 DSPMQMCHL 命令。

这将导致显示相应的 "显示通道" 面板以及该通道的当前设置的详细信息。 这些字段在 通道属性中进行了描述。

```
Display MQM Channel

Channel name . . . . . :  ST.JST.2TO1
Queue Manager Name . . . . . :  QMREL
Channel type . . . . . :  *SDR
Transport type . . . . . :  *TCP
Text 'description' . . . . . :  John's sender to WINSDOA1

Connection name . . . . . :  MUSTANG


Transmission queue . . . . . :  WINSDOA1

Message channel agent . . . . :
Library . . . . . :
Message channel agent user ID :  *NONE
Batch interval . . . . . :  0
Batch size . . . . . :  50
Disconnect interval . . . . . :  6000



F3=Exit  F12=Cancel  F21=Print
```

图 *32:* 显示 *TCP/IP* 通道 *(1)*

```
Display MQM Channel

Short retry interval . . . . . :  60
Short retry count . . . . . :  10
Long retry interval . . . . . :  6000
Long retry count . . . . . :  10
Security exit . . . . . :
Library . . . . . :
Security exit user data . . . :
Send exit . . . . . :
Library . . . . . :
Send exit user data . . . . . :
Receive exit . . . . . :
Library . . . . . :
Receive exit user data . . . . :
Message exit . . . . . :
Library . . . . . :
Message exit user data . . . . :
More...


F3=Exit  F12=Cancel  F21=Print
```

图 *33:* 显示 *TCP/IP* 通道 *(2)*

```
Display MQM Channel

Sequence number wrap . . . . . :  999999999
Maximum message length . . . . :  10000
Convert message . . . . . :  *NO
Heartbeat interval . . . . . :  300
Nonpersistent message speed . .  *FAST









Bottom



F3=Exit  F12=Cancel  F21=Print
```

图 *34:* 显示 *TCP/IP* 通道 *(3)*

IBM i 在 IBM i 上重命名通道

您可以从 "使用通道" 面板重命名通道。

要重命名消息通道，请从 "使用通道" 面板开始:

1. 结束通道。
2. 使用选项 3 (复制) 创建具有新名称的重复项。
3. 使用选项 5 (显示) 来检查是否已正确创建该选项。
4. 使用选项 4 (删除) 来删除原始通道。

如果您决定重命名消息通道，请确保同时重命名两个通道结束。

IBM i 在 IBM i 上使用通道状态

您可以从 "使用通道状态" 面板处理通道状态。

使用 WRKMQMCHST 命令可显示显示通道状态的一组面板中的第一个面板。 选择 "变更视图" (F11) 时，可以按顺序查看状态面板。

或者，从 "使用 MQM 通道" 面板中选择选项 8 (使用状态) 也会显示第一个状态面板。

```
MQSeries Work with Channel Status

Type options, press Enter.
5=Display  13=Ping  14=Start  15=End  16=Reset  17=Resolve


Opt Name          Connection      Indoubt   Last Seq
CARTS_CORSAIR_CHAN  GBIBMIYA.WINSDOA1    NO         1
CHLNIC         9.20.2.213       NO         3
FV.CHANNEL.PETER2   9.20.2.213       NO       6225
JST.1.2         9.20.2.201       NO        28
MP_MUST_TO_CORS    9.20.2.213       NO       100
MUSTANG.TO.CORSAIR  GBIBMIYA.WINSDOA1    NO        10
MP_CORS_TO_MUST    9.20.2.213       NO       101
JST.2.3         9.5.7.126        NO        32
PF_WINSDOA1_LU62   GBIBMIYA.IYA80020    NO        54
PF_WINSDOA1_LU62   GBIBMIYA.WINSDOA1    NO       500
ST.JCW.EXIT.2TO1.CHL 9.20.2.213       NO       216

Bottom
Parameters or command
===>
F3=Exit  F4=Prompt  F5=Refresh  F6=Create  F9=Retrieve  F11=Change view
F12=Cancel  F21=Print
```

图 *35:* 第一组通道状态面板

"使用通道状态" 面板中可用的选项包括:

菜单选项	描述
5=Display	显示通道设置。
13=Ping	在适当情况下启动 Ping 操作。
14=Start	启动通道。
15=End	停止通道。
16=Reset	重置通道序号。
17=Resolve	手动解决不确定通道情况。

IBM i IBM i 上的 "使用通道" 选项

使用命令 WRKMQMCHL 可访问 "使用通道" 面板，它允许您监视列出的所有通道的状态，并针对所选通道发出命令。

"使用通道" 面板中可用的选项包括:

菜单选项	描述
第 224 页的『2=Change』	更改通道的属性。
第 224 页的『3=Copy』	将通道的属性复制到新通道。
第 224 页的『4=Delete』	删除通道。

菜单选项	描述
第 224 页的『5=Display』	显示通道的当前设置。
第 225 页的『6=Create』	显示 "创建通道" 面板
第 225 页的『8=Work 状态』	显示通道状态面板。
第 226 页的『13=Ping』	运行 Ping 工具以通过与远程端交换固定数据消息来测试与相邻系统的连接。
第 226 页的『14=Start』	启动所选通道，或重置已禁用的接收方通道。
第 227 页的『15=End』	请求通道关闭。
第 227 页的『16=Reset』	请求通道在链路的此端重置序号。 要启动通道，两端的数字必须相等。
第 227 页的『17=Resolve』	请求通道在不与另一端建立连接的情况下解析不确定消息。
第 228 页的『18=显示权限』	显示 IBM MQ 对象权限
第 228 页的『19=授予权限』	授予 IBM MQ 对象权限
第 228 页的『20=撤销权限』	撤销 IBM MQ 对象权限
第 228 页的『21=恢复对象』	恢复 IBM MQ 对象
第 228 页的『22=记录映像』	记录 IBM MQ 对象图像

IBM i 2=Change

使用 "更改" 选项来更改现有通道定义。

"更改" 选项或 CHGMQMCHL 命令更改现有通道定义 (通道名称除外)。 在通道定义面板中输入要更改的字段，然后按 Enter 键以保存更新后的定义。

IBM i 3=Copy

使用 "复制" 选项来复制现有通道。

"复制" 选项使用 CPYMQMCHL 命令来复制现有通道。 "复制" 面板使您能够定义新的通道名称。 但是，必须将使用的字符限制为对 IBM i 对象名有效的字符; 请参阅 管理 IBM MQ for IBM i。

在 "复制" 面板上按 Enter 键以显示当前设置的详细信息。 您可以更改任何新的通道设置。 按 Enter 键以保存新的通道定义。

IBM i 4=Delete

使用 "删除" 选项可删除所选通道。

将显示一个面板以确认或取消您的请求。

IBM i 5=Display

使用 "显示" 选项来显示通道的当前定义。

此选项显示面板，其中的字段显示参数的当前值，并针对用户输入进行保护。

IBM i 6=Create

使用 "创建" 选项可显示 "创建通道" 面板。

使用 "创建" 选项，或从命令行输入 CRTMQMCHL 命令以获取 "创建通道" 面板。 有一些 "创建通道" 面板的示例，从 第 218 页的图 27 开始。

通过此面板，您可以从使用 IBM MQ for IBM i 提供的缺省值填充的字段屏幕中创建通道定义。 输入通道的名称，选择要创建的通道类型以及要使用的通信方法。

按 Enter 键时，将显示面板。 在此面板和其余面板中的所有必填字段中输入信息，然后按 Enter 键以保存定义。

通道两端的通道名称必须相同，并且在网络中是唯一的。 但是，必须将使用的字符限制为对 IBM MQ for IBM i 对象名有效的字符。

所有面板都具有 IBM MQ for IBM i 为某些字段提供的缺省值。 您可以定制这些值，也可以在创建或复制通道时更改这些值。 要定制值，请参阅 *IBM MQ for IBM i* 系统管理。

您可以创建自己的一组通道缺省值，方法是设置具有每种通道类型所需缺省值的虚拟通道，并在每次想要创建新的通道定义时复制这些虚拟通道。

相关参考

通道属性

IBM i 8=Work 状态

使用 "使用状态" 可查看详细的通道状态信息。

状态列告诉您通道是处于活动状态还是不活动状态，并且会持续显示在 "使用 MQM 通道" 面板中。 使用选项 8 (使用状态) 可查看显示的更多状态信息。 或者，可以从命令行使用 WRKMQMCHST 命令显示此信息。 请参阅第 223 页的『在 IBM i 上使用通道状态』。

- 通道名称
- 通道类型
- 通道状态
- 通道实例
- 远程队列管理器
- 传输队列的名称
- 通信连接名称
- 通道的不确定状态
- 上一个序号
- 不确定消息数
- 不确定序号
- 传输队列上的消息数
- 逻辑工作单元标识
- 不确定的逻辑工作单元标识
- 通道子状态
- 通道监视
- 头压缩
- 消息压缩
- 压缩时间指示符
- 压缩率指示符
- 传输队列时间指示符
- 网络时间指示符
- 退出时间指示符

- 批处理大小指示符
- 当前共享对话
- 最大共享对话数

IBM i *13=Ping*

使用 Ping 选项与远程端交换固定数据消息。

成功的 IBM MQ Ping 使系统主管确信通道可用且正常运行。

Ping 不涉及使用传输队列和目标队列。 它使用通道定义，相关通信链路和网络设置。

它仅可从发送方和服务器通道获取。 在链路的远侧启动相应的通道，并执行启动参数协商。 将正常通知错误。

消息交换的结果会在 Ping 面板中为您呈现，并且是返回的消息文本，加上发送消息的时间，以及收到回复的时间。

使用 LU 6.2 执行 ping 操作

在 IBM MQ for IBM i 中调用 Ping 时，将使用请求此功能的用户的用户标识来运行 Ping，而通道程序的正常运行方式是对通道程序采用 QMQM 用户标识。 用户标识流向接收端，并且必须在接收端对要分配的 LU 6.2 对话有效。

IBM i *14=Start*

使用 "启动" 选项可手动启动通道。

"启动" 选项可用于发送方，服务器和请求者通道。 如果已设置通道并触发队列管理器，那么不需要执行此操作。

"启动" 选项还用于接收方，服务器连接，集群发送方和集群接收方通道。 启动处于 STOPPED 状态的接收方通道意味着可以从远程通道启动该通道。

启动时，发送 MCA 将读取通道定义文件并打开传输队列。 发出通道启动序列，该序列远程启动接收方或服务器通道的相应 MCA。 当它们已启动时，发送方和服务器将处理到达传输队列的等待消息，并在它们到达时进行传输。

使用触发时，必须启动持续运行的触发器进程以监视启动队列。 可以使用 STRMQMCHLI 命令来启动进程。

在通道的远端，可能会启动接收进程以响应来自发送端的通道启动。 对于 LU 6.2 和 TCP/IP 连接的通道，执行此操作的方法不同:

- LU 6.2 连接的通道不需要在通道的接收端执行任何显式操作。
- TCP 连接的通道需要侦听器进程连续运行。 此进程等待来自链路的远程端的通道启动请求，并启动在该连接的通道定义中定义的进程。

  当远程系统为 IBM i 时，可以使用 STRMQMLSR 命令。

使用 "启动" 选项始终会导致通道在必要时再同步。

要获得成功，请执行以下操作:

- 必须存在本地和远程通道定义。 如果接收方或服务器连接通道没有相应的通道定义，那么如果通道是自动定义的，那么将自动创建缺省通道定义。 请参阅 通道自动定义出口程序 (Channel auto-definition exit program)。
- 传输队列必须存在，必须为 GET 启用，并且没有其他通道使用该传输队列。
- 必须存在本地和远程 MCA。
- 通信链路必须可用。
- 队列管理器必须正在运行 (本地和远程)。
- 消息通道必须处于不活动状态。

要传输消息，必须存在远程队列和远程队列定义。

将向面板返回一条消息，确认已接受启动通道的请求。 要确认 "启动" 进程已成功，请检查系统日志，或按 F5 (刷新屏幕)。

IBM i *15=End*

使用 "结束" 来停止通道活动

使用 "结束" 选项来请求通道停止活动。 通道不再发送任何消息。

在按 Enter 键之前选择 F4 ，以选择通道是变为 STOPPED 还是 INACTIVE ，以及是使用 CONTROLLED 或 IMMEDIATE 停止来停止通道。 操作员必须重新启动已停止的通道才能再次激活。 可以触发不活动通道。

stop immediate

使用 "立即停止" 可停止通道而不完成任何工作单元。

此选项将终止通道进程。 因此，通道未完成当前消息批次的处理，因此无法使通道处于不确定状态。 一般来说，运营者最好使用受控的停止选项。

停止受控

使用 "受控停止" 在当前工作单元结束时停止通道。

此选项请求通道有序关闭; 当前消息批次已完成，并与通道的另一端执行同步点过程。

重新启动已停止的通道

当通道进入 STOPPED 状态时，必须手动重新启动该通道。 您可以通过以下方式重新启动通道:

- 通过使用 **START CHANNEL** MQSC 命令。
- 通过使用 **Start Channel** PCF 命令。
- 通过使用 IBM MQ Explorer。
- z/OS 在 z/OS 上，使用 "启动通道" 面板。
- IBM i 在 IBM i 上，使用 **STRMQMCHL CL** 命令或 WRKMQMCHL 面板上的 **START** 选项。

对于发送方或服务器通道，当通道进入 STOPPED 状态时，关联的传输队列已设置为 GET (DISABLED) ，并且触发已关闭。 接收到启动请求时，将自动重置这些属性。

z/OS 如果通道启动程序在通道处于 "正在重试" 或 "已停止" 状态时停止，那么重新启动通道启动程序时将记住通道状态。 但是，如果通道启动程序在通道处于 STOPPED 状态时停止，那么将重置 SVRCONN 通道类型的通道状态。

Multi 如果队列管理器在通道处于 "正在重试" 或 "已停止" 状态时停止，那么重新启动队列管理器时将记住通道状态。 从 IBM MQ 8.0 开始，这也适用于 SVRCONN 通道。 先前，如果通道启动程序在通道处于 STOPPED 状态时停止，那么会重置 SVRCONN 通道类型的通道状态。

IBM i *16=Reset*

使用 "重置" 选项来强制新的消息序列。

"重置" 选项将更改消息序号。 请谨慎使用，并且仅在使用 "解决" 选项来解决任何不确定情况之后才使用。 此选项仅在发送方或服务器通道上可用。 第一条消息在下次启动通道时启动新序列。

IBM i *17=Resolve*

使用 "解析" 选项来强制本地落实或回退传输队列中保留的不确定消息。

当消息被发送方或服务器保留为不确定时，使用 "解析" 选项，例如，因为链路的一端已终止，并且它没有恢复的希望。 "解析" 选项接受以下两个参数之一 :BACKOUT 或 COMMIT。 回退会将消息复原到传输队列，而 "落实" 会废弃这些消息。

通道程序不会尝试与合作伙伴建立会话。 而是确定表示不确定消息的逻辑工作单元标识 (LUWID)。 然后，根据请求发出以下问题:

- BACKOUT 以将消息复原到传输队列; 或
- 用于从传输队列中删除消息的 COMMIT。

要使决议成功:

- 通道必须处于不活动状态
- 通道必须处于不确定状态
- 通道类型必须是发送方或服务器
- 通道定义 (本地) 必须存在
- 队列管理器必须正在运行，本地

IBM i *18*=显示权限

使用 "显示权限" 选项可显示授权用户对特定 IBM MQ 对象执行的操作。

对于所选对象和用户， DSPMQAUT 命令显示用户必须对 IBM MQ 对象执行操作的权限。 如果用户是多个组的成员，那么该命令将显示所有组对该对象的组合权限。

IBM i *19*=授予权限

使用 "授予权限" 选项向另一个用户或用户组授予对 IBM MQ 对象执行操作的权限。

GRTMQMAUT 命令仅对 QMQMADM 组中的用户可用。 QMQMADM 中的用户通过按名称标识用户或通过向 *PUBLIC 中的所有用户授予权限，向其他用户授予对命令中指定的 IBM MQ 对象执行操作的权限。

IBM i *20*=撤销权限

使用 "撤销" 权限来除去对用户的对象执行操作的权限。

RVKMQMAUT 命令仅对 QMQMADM 组中的用户可用。 QMQMADM 组中的用户通过按名称标识用户或通过撤销 *PUBLIC 中所有用户的权限，从其他用户处除去对命令中指定的 IBM MQ 对象执行操作的权限。

IBM i *21*=恢复对象

使用 "恢复" 对象可从 IBM MQ 日志中存储的信息复原受损对象。

恢复对象使用 "重新创建 MQ 对象" 命令 (RCRMQMOBJ) 来恢复该命令中指定的所有受损对象。 如果对象未损坏，那么不会对该对象执行任何操作。

IBM i *22*=记录映像

使用 "记录" 图像来减少恢复一组对象所需的日志接收器数，并最大限度地缩短恢复时间。

RCDMQMIMG 命令对命令中选择的所有对象执行检查点。 它使集成文件系统 (IFS) 中对象的当前值与有关对象的后续信息 (例如，日志接收器中记录的 MQPUTs 和 MQGET) 同步。

当该命令完成时， IFS 中的对象是最新的，并且不再需要这些日志接收器来恢复这些对象。 可以拆离任何断开连接的日志接收器 (只要它们不需要存在以恢复其他对象)。

IBM i 为 IBM i 设置通信

当启动分布式排队管理通道时，它会尝试使用通道定义中指定的连接。 要使其成功，需要定义连接并使其可用。

DQM 是 IBM MQ for IBM i 的远程排队工具。 它为 IBM MQ for IBM i 队列管理器提供通道控制程序，这些程序构成通信链路的接口，由系统操作员控制。

当启动分布式排队管理通道时，它会尝试使用通道定义中指定的连接。 要使其成功，需要定义连接并使其可用。 本节说明如何确保连接已定义且可用。

在可以启动通道之前，必须按本节中所述定义传输队列，并且必须包含在消息通道定义中。

您可以在 IBM MQ for IBM i 系统之间选择以下两种通信形式的以太网:

- 第 229 页的『在 IBM i 上定义 TCP 连接』

  对于 TCP，可以使用主机地址，并按 *IBM i* 通信配置参考中所述设置这些连接。

  在 TCP 环境中，为每个分布式服务分配一个唯一的 TCP 地址，远程机器可以使用该地址来访问该服务。 TCP 地址由主机名/编号和端口号组成。 所有队列管理器都使用这样的数字通过 TCP 相互通信。
- 第 230 页的『在 TCP 上接收』

  此通信形式需要定义 IBM i SNA 逻辑单元类型 6.2 (LU 6.2)，该逻辑单元类型提供服务本地队列管理器的 IBM i 系统与服务远程队列管理器的系统之间的物理链路。 请参阅 *IBM i Communication Configuration Reference*，以获取有关在 IBM i 中配置通信的详细信息。

在需要时，还必须准备触发安排，定义必要的进程和队列。

CD MQ Adv. V 9.2.0 使用 TCP/IP 的消息通道可以指向 IBM Aspera fasp.io Gateway，它提供了可显着提高网络吞吐量的快速 TCP/IP 隧道。 在任何授权平台上运行的队列管理器都可以通过 Aspera gateway 进行连接。 网关本身部署在 Red Hat 或 Ubuntu Linux 或 Windows 上。 请参阅 在 Linux 或 Windows 上定义 Aspera gateway 连接。

相关任务

第 216 页的『监视和控制 IBM i 上的通道』
使用 DQM 命令和面板来创建，监视和控制到远程队列管理器的通道。 每个队列管理器都有一个 DQM 程序，用于控制与兼容远程队列管理器的互连。

相关参考

示例配置- IBM MQ for IBM i
IBM MQ for IBM i 的消息通道规划示例
IBM i 上的相互通信作业
IBM i 上的通道状态

IBM i 在 *IBM i* 上定义 *TCP* 连接

您可以使用 "连接名称" 字段在通道定义中定义 TCP 连接。

通道定义包含一个字段 CONNECTION NAME，其中包含目标的 TCP 网络地址或主机名 (例如 ABCHOST)。 TCP 网络地址可以采用 IPv4 点分十进制格式 (例如 127.0.0.1) 或 IPv6 十六进制格式 (例如 2001:DB8:0:0:0:0:0:0)。 如果 CONNECTION NAME 是主机名或名称服务器，那么将使用 IBM i 主机表将主机名转换为 TCP 主机地址。

完整 TCP 地址需要端口号; 如果未提供此号码，那么将使用缺省端口号 1414。 在连接的起始端 (发送方，请求者和服务器通道类型) 上，可以提供连接的可选端口号，例如:

```
Connection name 127.0.0.1 (1555)
```

在这种情况下，启动端尝试连接到端口 1555 上的接收程序。

CD MQ Adv. V 9.2.0 使用 TCP/IP 的消息通道可以指向 IBM Aspera fasp.io Gateway，它提供了可显着提高网络吞吐量的快速 TCP/IP 隧道。 在任何授权平台上运行的队列管理器都可以通过 Aspera gateway 进行连接。 网关本身部署在 Red Hat 或 Ubuntu Linux 或 Windows 上。 请参阅 在 Linux 或 Windows 上定义 Aspera gateway 连接。

使用 TCP 侦听器储备选项

在 TCP 中，除非在服务器和客户机之间进行三方握手，否则将处理不完整的连接。 这些连接称为未完成的连接请求。 为这些未完成的连接请求设置了最大值，可以将其视为在 TCP 端口上等待侦听器接受请求的积压请求。

请参阅 第 214 页的『在 IBM MQ for Multiplatforms 上使用 TCP 侦听器储备选项』 以获取更多信息以及 IBM i 的特定值。

相关概念

第 230 页的『在 TCP 上接收』

接收通道程序是响应来自发送通道的启动请求而启动的。 要响应启动请求，必须启动侦听器程序以检测入局网络请求并启动关联通道。 使用 STRMQMLSR 命令启动此侦听器程序。

IBM i 在 *TCP* 上接收

接收通道程序是响应来自发送通道的启动请求而启动的。 要响应启动请求，必须启动侦听器程序以检测入局网络请求并启动关联通道。 使用 STRMQMLSR 命令启动此侦听器程序。

您可以为每个队列管理器启动多个侦听器。 缺省情况下， STRMQMLSR 命令使用端口 1414 ，但您可以覆盖此值。 要覆盖缺省设置，请将以下语句添加到所选队列管理器的 qm.ini 文件。 在此示例中，需要侦听器使用端口 2500:

```
TCP:
Port=2500
```

qm.ini 文件位于此 IFS 目录中：/队列管理器名称 /QIBM/UserData/mqm/qmgrs/

当启动 TCP 侦听器时，此新值是只读的。 如果已运行侦听器，那么该程序不会看到此更改。 要使用新值，请停止侦听器并再次发出 STRMQMLSR 命令。 现在，每当使用 STRMQMLSR 命令时，侦听器都缺省为新端口。

或者，可以在 STRMQMLSR 命令上指定其他端口号。 例如：

```
STRMQMLSR MQMNAME( queue manager name ) PORT(2500)
```

此更改使侦听器在侦听器作业的持续时间内缺省为新端口。

使用 TCP SO_KEEPALIVE 选项

如果要使用 SO_KEEPALIVE 选项（更多信息，请参阅 "第 185 页的『正在检查通道的另一端是否仍然可用』），则必须在队列管理器配置文件中添加以下条目（IFS 目录中为 "qm.ini，"/QIBM/UserData/mqm/qmgrs/为 "队列管理器名称）：

```
TCP:
KeepAlive=yes
```

然后，必须发出以下命令:

```
CFGTCP
```

选择选项 3 (更改 TCP 属性)。 现在可以指定时间间隔 (以分钟为单位)。 您可以指定 1 到 40320 分钟范围内的值; 缺省值为 120。

使用 TCP 侦听器储备选项

在 TCP 上接收时，将设置最大未完成的连接请求数。 此数目可被视为在 TCP 端口上等待侦听器接受请求的积压 请求。

IBM i 上的缺省侦听器任务列表值为 255。 如果积压达到此值，那么将拒绝 TCP 连接，并且通道无法启动。

对于 MCA 通道，这会导致通道进入 RETRY 状态，并在稍后重试连接。

对于客户机连接，客户机从 MQCONN 接收到 MQRC_Q_MGR_NOT_AVAILABLE 原因码，并且可以稍后重试连接。

但是，为了避免此错误，您可以在 qm.ini 文件中添加一个条目:

```
ListenerBacklog = n
```

这将覆盖 TCP 侦听器的缺省最大未完成请求数 (255)。

注: 某些操作系统支持大于缺省值的值。 如果需要，可以使用此值来避免达到连接限制。

IBM i 在 *IBM i* 上定义 *LU 6.2* 连接

使用标准 LU 6.2 连接的方式名， TP 名和连接名来定义 LU 6.2 通信详细信息。

链路的起始端必须具有路由项定义以补充此 CSI 对象。 *IBM i Programming: Work Management Guide* 中提供了有关管理来自远程 LU 6.2 系统的工作请求的更多信息。

请参阅 *Multiplatform APPC Configuration Guide* 和下表以获取信息。

表 *26:* 远程队列管理器平台的本地 *IBM i* 系统上的设置	
远程平台	**TPNAME**
z/OS 或 MVS	与有关远程队列管理器的相应辅助信息中的相同。
IBM i	与 IBM i 系统上的路由条目中的比较值相同。
AIX and Linux 系统	在远程 LU 6.2 配置中定义的可调用事务程序。
Windows	如 Windows Run Listener 命令或使用 Windows 上的 TpSetup 定义的可调用事务程序中所指定。

如果在同一计算机上具有多个队列管理器，请确保通道定义中的 TPname 是唯一的。

相关概念

第 231 页的『正在启动结束 (发送方)』
使用 CRTMQMCHL 命令来定义传输类型为 *LU62 的通道。

第 233 页的『已启动结束 (接收方)』
使用 CRTMQMCHL 命令来定义传输类型为 *LU62 的消息通道链路的接收端。

IBM i 正在启动结束 (发送方)

使用 CRTMQMCHL 命令来定义传输类型为 *LU62 的通道。

在 IBM MQ for IBM i V5.3 或更高版本中，可选择使用 CSI 对象。

图 LU 6.2 通信设置面板-启动结束中显示了启动结束面板。 要获取显示的完整面板，请从第一个面板中按 F10 。

```
Create Comm Side Information (CRTCSI)

Type choices, press Enter.

Side information . . . . . > WINSDOA1   Name
Library . . . . . >  QSYS    Name, *CURLIB
Remote location . . . . . > WINSDOA1   Name
Transaction program . . . . . > MQSERIES

Text 'description' . . . . . .  *BLANK


Additional Parameters

Device . . . . . .  *LOC     Name, *LOC
Local location . . . . . .  *LOC     Name, *LOC, *NETATR
Mode . . . . . .  JSTMOD92   Name, *NETATR
Remote network identifier . . .  *LOC     Name, *LOC, *NETATR, *NONE
Authority . . . . . .  *LIBCRTAUT  Name, *LIBCRTAUT, *CHANGE...

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

图 *36: LU 6.2* 通信设置面板启动结束

完成启动结束字段，如下所示:

辅助信息

为此定义提供一个用于存储要创建的辅助信息对象的名称，例如 WINSDOA1。

注: 对于 LU 6.2，消息通道定义与通信连接之间的链路是发送端消息通道定义的 **连接名称** 字段。 此字段包含 CSI 对象的名称。

库
存储此定义的库的名称。

CSI 对象必须在为消息通道提供服务的程序 (例如 QSYS ， QMQM 和 QGPL) 可访问的库中可用。

如果名称不正确，缺少或找不到，那么在通道启动时会发生错误。

远程位置
指定程序与之通信的远程位置名。

简而言之，此必需参数包含远程系统上的伙伴的逻辑单元名称，如用于两个系统之间通信链路的设备描述中所定义。

通过在远程系统上发出命令 DSPNETA 并查看缺省本地位置名，可以找到 **远程位置** 名称。

事务程序
指定要启动的远程系统上事务程序的名称 (最多 64 个字符)。 它可以是事务进程名称，程序名，通道名称或与路由条目中的 **比较值** 匹配的字符串。

此参数是必需的。

注: 要指定 SNA 服务事务程序名，请输入服务事务程序名的十六进制表示法。 例如，要指定十六进制表示为 21F0F0F1 的服务事务程序名，请输入 X'21F0F0F1'。

有关 SNA 服务事务程序名的更多信息，请参阅 LU 类型 6.2 的 *SNA Transaction Programmer 's Reference* 手册。

如果接收端是另一个 IBM i 系统，那么 **事务程序** 名称用于将发送端的 CSI 对象与接收端的路由条目相匹配。 对于目标 IBM i 系统上的每个队列管理器，此名称必须唯一。 请参阅 Initiated end (接收方)下的 **要调用的程序** 参数。 另请参阅 "添加路由条目" 面板中的 **比较数据: 比较值** 参数。

文本描述
用于提醒您打算使用此连接的描述 (最多 50 个字符)。

设备
指定用于远程系统的设备描述的名称。 可能的值为:

***LOC**
设备由系统确定。

设备名
指定与远程位置关联的设备的名称。

本地位置
指定本地位置名。 可能的值为:

***LOC**
本地位置名由系统确定。

***NETATR**
使用系统网络属性中指定的 LCLLOCNAME 值。

本地位置名
指定位置的名称。 如果要指示远程位置的特定位置名，请指定本地位置。 可使用 DSPNETA 命令找到位置名。

模式
指定用于控制会话的方式。 此名称与公共编程接口 (CPI)-通信 Mode_Name 相同。 可能的值为:

***NETATR**
将使用网络属性中的方式。

空白
使用 8 个空白字符。

方式名
指定远程位置的方式名。

注: 由于该方式确定通信会话的传输优先级，因此根据要发送的消息的优先级 (例如 MQMODE_HI , MQMODE_MED 和 MQMODE_LOW) 来定义不同的方式可能很有用。 (可以有多个 CSI 指向同一位置。)

远程网络标识
指定与远程位置一起使用的远程网络标识。 可能的值为：

***LOC**
使用远程位置的远程网络标识。

***NETATR**
使用网络属性中指定的远程网络标识。

***NONE**
远程网络没有名称。

远程网络标识
指定远程网络标识。 在远程位置使用 DSPNETA 命令来查找此网络标识的名称。 它是远程位置的 "本地网络标识"。

权限
指定要授予对该对象没有特定权限的用户，不在权限列表上的用户以及对该对象没有特定权限的组概要文件的用户的权限。 可能的值为：

***LIBCRTAUT**
对象的公共权限取自指定库的 CRTAUT 参数。 此值在创建时确定。 如果创建对象后库的 CRTAUT 值发生更改，那么新值不会影响现有对象。

*** 更改**
更改权限允许用户对对象执行基本功能，但是用户无法更改对象。 更改权限提供对象操作权限和所有数据权限。

***ALL**
用户可以执行所有操作，但限于所有者或由权限列表管理权限控制的操作除外。 用户可以控制对象的存在，并指定对象的安全性，更改对象以及对对象执行基本功能。 用户可以更改对象的所有权。

*** 使用**
使用权限提供对象操作权限和读权限。

*** 排除**
排除权限会阻止用户访问对象。

权限列表
指定具有用于辅助信息的权限的权限列表的名称。

IBM i 已启动结束 (接收方)
使用 CRTMQMCHL 命令来定义传输类型为 *LU62 的消息通道链路的接收端。

将 CONNECTION NAME 字段留空，并确保相应的详细信息与通道的发送端匹配。 有关详细信息，请参阅 创建通道。

要使发起端能够启动接收通道，请在发起端向子系统添加路由项。 子系统必须是分配 LU 6.2 会话中使用的 APPC 设备的子系统。 因此，它必须具有该设备的有效通信项。 路由项调用启动消息通道接收端的程序。

使用 IBM i 命令 (例如 ADDRTGE) 来定义由通信会话启动的链路的结束。

启动的结束面板显示在 LU 6.2 通信设置面板-添加路由条目中。

```
Add Routing Entry (ADDRTGE)

Type choices, press Enter.

Subsystem description . . . . .  QCMN     Name
Library . . . . . .   *LIBL    Name, *LIBL, *CURLIB
Routing entry sequence number .  1       1-9999
Comparison data:
Compare value . . . . . .  MQSERIES

Starting position . . . . . .  37      1-80
Program to call . . . . . .  AMQCRC6B   Name, *RTGDTA
Library . . . . . .  QMAS400    Name, *LIBL, *CURLIB
Class . . . . . .  *SBSD     Name, *SBSD
Library . . . . . .  *LIBL     Name, *LIBL, *CURLIB
Maximum active routing steps . .  *NOMAX    0-1000, *NOMAX
Storage pool identifier . . . .  1       1-10



Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

图 *37: LU 6.2* 通信设置面板启动的结束

子系统描述
此定义所在的子系统的名称。 使用 IBM i WRKSBSD 命令来查看和更新路由项的相应子系统描述。

路由项序号
子系统中用于标识此通信定义的唯一编号。 可以使用 1-9999 范围内的值。

比较数据: 比较值
要与 **事务程序** 参数启动会话时接收到的字符串进行比较的文本字符串，如 图 1 中所示。 字符串派生自发送方 CSI 的 "事务程序" 字段。

比较数据: 起始位置
要开始比较的字符串中的字符位置。

注: 起始位置字段是字符串中用于比较的字符位置，此位置始终为 37。

要调用的程序
运行要调用以启动会话的入站消息程序的程序的名称。

将为缺省队列管理器调用程序 AMQCRC6A。 此程序随 IBM MQ for IBM i 提供，用于设置环境，然后调用 AMQCRS6A。

对于其他队列管理器:

- 每个队列管理器都有一个特定的 LU 6.2 可调用程序位于其库中。 此程序称为 AMQCRC6B ，在创建队列管理器时自动生成。
- 每个队列管理器都需要一个具有要添加的唯一路由数据的特定路由条目。 此路由数据必须与请求系统提供的 **事务程序** 名称匹配 (请参阅 启动结束 (发送方))。

LU 6.2 通信设置面板-显示路由条目中显示了一个示例:

```
Display Routing Entries
System:  MY400
Subsystem description:  QCMN      Status:  ACTIVE

Type options, press Enter.
5=Display details

Start
Opt  Seq Nbr  Program    Library    Compare Value   Pos
10   *RTGDTA           'QZSCSRVR'     37
20   *RTGDTA           'QZRCSRVR'     37
30   *RTGDTA           'QZHQTRG'     37
50   *RTGDTA           'QVPPRINT'     37
60   *RTGDTA           'QNPSERVR'     37
70   *RTGDTA           'QNMAPINGD'    37
80   QNMAREXECD  QSYS     'AREXECD'     37
90   AMQCRC6A   QMQMBW    'MQSERIES'     37
100   *RTGDTA           'QTFDWNLD'     37
150   *RTGDTA           'QMFRCVR'     37


F3=Exit  F9=Display all detailed descriptions  F12=Cancel
```

图 *38: LU 6.2* 通信设置面板启动的结束

在 LU 6.2 通信设置面板-显示路由条目中，序号 90 表示缺省队列管理器，并提供与 IBM MQ for IBM i 的先前发行版 (即 V3R2， V3R6， V3R7 和 V4R2) 的配置的兼容性。 这些发行版仅允许一个队列管理器。 序号 92 和 94 表示使用库 QMALPHA 和 QMBETA 创建的另外两个名为 ALPHA 和 BETA 的队列管理器。

注: 通过使用不同的路由数据，每个队列管理器可以有多个路由条目。 根据所使用的类，这些条目提供了不同作业优先级的选项。

类

用于通过此路由项启动的步骤的类的名称和库。 该类定义路由步骤运行环境的属性，并指定作业优先级。 必须指定相应的类条目。 例如，使用 WRKCLS 命令来显示现有类或创建类。 *IBM i Programming: Work Management Guide* 中提供了有关管理来自远程 LU 6.2 系统的工作请求的更多信息。

关于工作管理的说明

AMQCRS6A 作业无法利用 工作管理 中记录的正常 IBM i 工作管理功能，因为它未以与其他 IBM MQ 作业相同的方式启动。 要更改 LU62 接收方作业的运行时属性，可以进行下列其中一项更改:

- 更改在 AMQCRS6A 作业的路由条目上指定的类描述
- 更改通信项上的作业描述

请参阅 *IBM i* 编程*:* 工作管理指南 ， 以获取有关配置通信作业的更多信息。

配置队列管理器集群

集群提供了一种用于以简化初始配置和持续管理的方式互连队列管理器的机制。 您可以定义集群组件，以及创建和管理集群。

开始之前

有关集群概念的简介，请参阅 集群。

在设计队列管理器集群时，必须做出一些决策。 请参阅 示例集群 和 设计集群。

相关任务

第 358 页的『将群集主题定义移至不同的队列管理器』
对于主题主机路由或直接路由群集，在停用队列管理器时，或由于群集队列管理器发生故障或在相当长一段时间内不可用，可能需要移动群集主题定义。

相关参考
删除主题

定义集群的组件

集群由队列管理器，集群通道和集群队列组成。 您可以定义集群队列，并修改缺省集群对象的某些方面。 您可以获取有关自动定义的通道的配置和状态信息，以及有关各个集群发送方通道与传输队列之间的关系的信息。

请参阅以下子主题以获取有关定义每个集群组件的信息:

相关概念
集群的组件
集群通道
相关任务
定义集群主题
第 244 页的『设置新集群』
遵循以下指示信息来设置示例集群。 单独的指示信息描述在 TCP/IP 上设置集群， LU 6.2 以及使用单个传输队列或多个传输队列。 通过将消息从一个队列管理器发送到另一个队列管理器来测试集群工作。

第 254 页的『将队列管理器添加至集群』
遵循以下指示信息将队列管理器添加到您创建的集群。 使用单集群传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 将消息传输到集群队列和主题。

定义集群队列

集群队列是由集群队列管理器托管并可供集群中其他队列管理器使用的队列。将集群队列定义为托管该队列的集群队列管理器上的本地队列。 指定队列所属的集群的名称。

以下示例显示了使用 CLUSTER 选项定义集群队列的 **runmqsc** 命令:

```
DEFINE QLOCAL(Q1) CLUSTER(SALES)
```

集群队列定义将播发给集群中的其他队列管理器。 集群中的其他队列管理器无需相应的远程队列定义即可将消息放入集群队列。 可以使用集群名称列表在多个集群中播发集群队列。

在播发队列时，集群中的任何队列管理器都可以将消息放入该队列中。 要放入消息，队列管理器必须从完整存储库中查明托管该队列的位置。 然后，它会将一些路由信息添加到消息，并将消息放到集群传输队列上。

z/OS 集群队列可以是 IBM MQ for z/OS 中队列共享组的成员所共享的队列。

正在绑定

您可以创建一个集群，在该集群中，多个队列管理器托管同一集群队列的实例。 确保序列中的所有消息都发送到队列的同一实例。 您可以使用 MQOPEN 调用上的 MQOO_BIND_ON_OPEN 选项将一系列消息绑定到特定队列。

集群传输队列

队列管理器可以将集群中其他队列管理器的消息存储在多个传输队列上。 您可以采用两种不同的方式配置队列管理器以将消息存储在多个集群传输队列上。 如果将队列管理器属性 **DEFCLXQ** 设置为 CHANNEL，那么将自动从 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 为每个集群发送方通道创建不同的集群传输队列。 如果将 CLCHNAME 传输队列选项设置为与一个或多个集群发送方通道匹配，队列管理器可以将匹配通道的消息存储在该传输队列上。

注意: 如果要将专用 SYSTEM.CLUSTER.TRANSMIT.QUEUES 与从低于 IBM WebSphere MQ 7.5 的产品版本升级的队列管理器配合使用，请确保 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 将 SHARE/NOSHARE 选项设置为 **SHARE**。

在发送之前，会将另一个队列管理器上的集群队列的消息放在集群传输队列上。 集群发送方通道将消息从集群传输队列传输到其他队列管理器上的集群接收方通道。 缺省情况下，一个系统定义的集群传输队列保存要

传输到其他集群队列管理器的所有消息。 队列称为 SYSTEM.CLUSTER.TRANSMIT.QUEUE。 属于集群的队列管理器可以将此集群传输队列上的消息发送到同一集群中的任何其他队列管理器。

缺省情况下，将在除 z/OS 以外的每个队列管理器上创建单个 SYSTEM.CLUSTER.TRANSMIT.QUEUE 队列的定义。 z/OS 在 z/OS 上，可以使用提供的样本 **CSQ4INSX** 来定义定义定义。

您可以配置队列管理器，以使用多个传输队列将消息传输到其他集群队列管理器。 您可以手动定义其他集群传输队列，或者让队列管理器自动创建队列。

要使队列由队列管理器自动创建，请将队列管理器属性 DEFCLXQ 从 SCTQ 更改为 CHANNEL。 结果是队列管理器为创建的每个集群发送方通道创建一个单独的集群传输队列。 传输队列作为模型队列 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 中的永久动态队列创建。 每个永久动态队列的名称为 SYSTEM.CLUSTER.TRANSMIT. *ChannelName*。 在本地传输队列属性 CLCHNAME 中设置与每个永久动态集群传输队列相关联的集群发送方通道的名称。 远程集群队列管理器的消息放在关联集群发送方通道的永久动态集群传输队列上，而不是放在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上。

要手动创建群集传输队列，请创建一个本地队列，将 USAGE 属性设置为 "XMITQ，并将 CLCHNAME 属性设置为可解析到一个或多个群集发送器通道的通用通道名；请参阅 ClusterChannelName。 如果手动创建集群传输队列，那么您可以选择将传输队列与单个集群发送方通道或多个集群发送方通道相关联。 CLCHNAME 属性是通用名称，这意味着可以在名称中放置多个通配符 “*”。

除了您手动创建的用于将队列管理器连接到完整存储库的初始集群发送方通道外，将自动创建集群发送方通道。 当存在要传输到集群队列管理器的消息时，将自动创建这些消息。 创建它们的名称与接收目标队列管理器上该特定集群的集群消息的集群接收方通道的名称相同。

如果遵循集群接收方通道的命名约定，那么可以为 CLCHNAME 定义通用值，以将不同类型的集群消息过滤到不同的传输队列。 例如，如果遵循 *ClusterName. QmgrName* 的集群接收方通道的命名约定，那么通用名称 *ClusterName*.* 会将不同集群的消息过滤到不同的传输队列上。 您必须手动定义传输队列，并将每个传输队列中的 CLCHNAME 设置为 *ClusterName*.*。

对集群传输队列与集群发送方通道的关联所作的更改不会立即生效。 集群发送方通道正在服务的当前关联传输队列可能包含正在由集群发送方通道传输的消息。 仅当集群发送方通道未处理当前关联的传输队列上的消息时，队列管理器才能将集群发送方通道的关联更改为不同的传输队列。 当没有消息保留在要由集群发送方通道处理的传输队列上时，或者当消息处理暂挂且集群发送方通道没有 “进行中” 消息时，可能会发生此情况。 发生此情况时，集群发送方通道的任何未处理消息都会传输到新关联的传输队列，并且集群发送方通道的关联会发生更改。

您可以创建解析为集群传输队列的远程队列定义。 在定义中，队列管理器 QMX 与本地队列管理器位于同一集群中，并且没有传输队列 QMX。

```
DEFINE QREMOTE(A) RNAME(B) RQMNAME(QMX)
```

在队列名称解析期间，集群传输队列优先于缺省传输队列。 放入 A 的消息存储在集群传输队列上，然后发送到 QMX 上的远程队列 B 。

队列管理器还可以与不属于集群的其他队列管理器进行通信。 您必须以与分布式排队环境中相同的方式定义通道和到另一个队列管理器的传输队列。

注: 应用程序必须写入解析为集群传输队列的队列，并且不得直接写入集群传输队列。

远程队列的自动定义

集群中的队列管理器不需要集群中的远程队列的远程队列定义。 集群队列管理器从完整存储库中找到远程队列的位置。 它将路由信息添加到消息并将其放入集群传输队列中。 IBM MQ 会自动创建等同于远程队列定义的定义，以便可以发送消息。

不能更改或删除自动创建的远程队列定义。 但是，通过将 DISPLAY QUEUE **runmqsc** 命令与 CLUSINFO 属性配合使用，您可以查看队列管理器上的所有本地队列以及所有集群队列，包括远程队列管理器上的集群队列。 例如：

```
DISPLAY QUEUE(*) CLUSINFO
```

相关概念
集群队列
如何选择要使用的集群传输队列类型
相关参考
ClusterChannelName (MQCHAR20)

使用自动定义的集群发送方通道

在通过使其初始 CLUSSDR 和 CLUSRCVR 定义向集群引入队列管理器之后，当需要将消息移动到集群中的另一个队列管理器时， IBM MQ 会自动生成其他集群发送方通道定义。 您可以查看有关自动定义的集群发送方通道的信息，但无法对其进行修改。 要修改其行为，可以使用通道自动定义出口。

开始之前

有关自动定义的通道的简介，请参阅 自动定义的集群发送方通道。

关于此任务

根据需要由集群创建自动定义的集群发送方通道，并且这些通道保持活动状态，直到使用正常断开连接时间间隔规则将其关闭为止。

可以自动定义集群发送方通道 (CLUSSDR) 以移动应用程序消息和内部集群管理消息。 例如，在 发布/预订集群 (已定义集群主题的集群) 中，可以在部分存储库之间定义通道以允许交换 "代理预订" 状态。 当长时间内不需要 (不活动) 自动定义的 CLUSSDR 时，会将其从部分存储库的集群信息高速缓存中除去，并且不再在该队列管理器上显示。

Multi 在 多平台上， OAM (对象权限管理器) 不知道是否存在自动定义的集群发送方通道。 如果在自动定义的集群发送方通道上发出 **start**， **stop**， **ping**， **reset** 或 **resolve** 命令，那么 OAM 会检查您是否有权对匹配的集群接收方通道执行相同的操作。

z/OS 在 z/OS 上，可以采用与任何其他通道相同的方式保护自动定义的集群发送方通道。

过程

- 显示有关给定集群队列管理器的自动定义通道的信息。

  无法使用 DISPLAY CHANNEL **runmqsc** 命令来查看自动定义的通道。 要查看自动定义的通道，请使用以下命令:

  ```
  DISPLAY CLUSQMGR(qMgrName)
  ```

- 显示给定 CLUSRCVR 的自动定义通道的状态。

  要显示与您创建的 CLUSRCVR 通道定义相对应的自动定义的 CLUSSDR 通道的状态，请使用以下命令:

  ```
  DISPLAY CHSTATUS(channelname)
  ```

- 使用通道自动定义出口来修改自动定义的通道的行为。

  如果要编写用户出口程序以定制集群发送方通道或集群接收方通道，那么可以使用 IBM MQ 通道自动定义出口。 例如，可以在集群环境中使用通道自动定义出口来进行以下任何修改:

  - 定制通信定义，即 SNA LU6.2 名称。
  - 添加或除去其他出口，例如，安全出口。
  - 更改通道出口的名称。

    CLUSSDR 通道出口的名称是从 CLUSRCVR 通道定义自动生成的，因此可能不适合您的需要-尤其是当通道的两端位于不同的平台上时。

    在不同的平台上，出口名称的格式不同。 例如:

    - z/OS 在 z/OS 平台上， SCYEXIT (安全出口名称) 参数的格式为 SCYEXIT('SECEXIT')

- Windows 在 Windows 平台上， SCYEXIT (*security exit name*) 参数的格式为 SCYEXIT('*drive*:*path**library* (secexit)')

注: z/OS 如果没有通道自动定义出口，那么 z/OS 队列管理器将从通道另一端的 CLUSRCVR 通道定义派生 CLUSSDR 通道出口名称。 要从非 z/OS 名称派生 z/OS 出口名称，请使用以下算法:

- 多平台 上的出口名称的一般格式为 *path*/*library* (*function*)。
- 如果存在 *function* ，那么最多使用 8 个字符。
- 否则，最多使用 8 个字符的 库 。

例如:

- /var/mqm/exits/myExit.so(MsgExit) 转换为 MSGEXIT
- /var/mqm/exits/myExit 转换为 MYEXIT
- /var/mqm/exits/myExit.so(ExitLongName) 转换为 EXITLONG

- 对于早于 IBM WebSphere MQ 7 的队列管理器，请将 **PROPCTL** 属性设置为值 NONE。

  每个自动定义的集群发送方通道都基于相应的集群接收方通道。 在 IBM WebSphere MQ 7 之前，集群接收方通道没有 **PROPCTL** 属性，因此此属性在自动定义的集群发送方通道中设置为 COMPAT 。

  如果集群需要使用 **PROPCTL** 从从 IBM WebSphere MQ 7 或更高版本的队列管理器到较低版本的 IBM MQ 上的队列管理器的消息中除去应用程序头 (例如 RFH2)，那么必须编写将 **PROPCTL** 设置为值 NONE 的通道自动定义出口。

- 使用通道属性 LOCLADDR 来控制寻址的各个方面。
  - 要使出站 (TCP) 通道能够使用特定 IP 地址，端口或端口范围，请使用通道属性 LOCLADDR。 如果您有多个网卡，并且希望通道使用特定的网卡进行出站通信，那么这很有用。
  - 要在 CLUSSDR 通道上指定虚拟 IP 地址，请在手动定义的 CLUSSDR 上使用 LOCLADDR 中的 IP 地址。 要指定端口范围，请使用 CLUSRCVR 中的端口范围。
  - 如果集群需要使用 LOCLADDR 来获取出站通信通道以绑定到特定 IP 地址，那么您可以编写通道自动定义出口，以将 LOCLADDR 值强制转换为其自动定义的任何 CLUSSDR 通道。 您还必须在手动定义的 CLUSSDR 通道中指定该值。
  - 如果您希望集群中的所有队列管理器将特定端口或端口范围用于其所有出站通信，请将端口号或端口范围放在 CLUSRCVR 通道的 LOCLADDR 中。

  注: 请勿将 IP 地址放在 CLUSRCVR 通道的 LOCLADDR 字段中，除非所有队列管理器都在同一服务器上。 LOCLADDR IP 地址将传播到使用 CLUSRCVR 通道连接的所有队列管理器的自动定义的 CLUSSDR 通道。

  Multi 在 多平台上，可以设置用于所有未定义本地地址的发送方通道的缺省本地地址值。 缺省值是通过在启动队列管理器之前设置 MQ_LCLADDR 环境变量来定义的。 值的格式与 MQSC 属性 LOCLADDR 的格式相匹配。

相关参考

本地地址 (LOCLADDR)

使用缺省集群对象

您可以通过运行 MQSC 或 PCF 命令，以与任何其他通道定义相同的方式更改缺省通道定义。 请勿更改缺省队列定义， SYSTEM.CLUSTER.HISTORY.QUEUE 除外。

有关这些对象的完整列表，请参阅 缺省集群对象。 以下列表仅包含可更改的对象。

SYSTEM.CLUSTER.HISTORY.QUEUE

集群中的每个队列管理器都有一个名为 SYSTEM.CLUSTER.HISTORY.QUEUE 的本地队列。 SYSTEM.CLUSTER.HISTORY.QUEUE 用于存储集群状态信息的历史记录以用于服务目的。

在缺省对象设置中， SYSTEM.CLUSTER.HISTORY.QUEUE 设置为 PUT (ENABLED)。 要禁止历史记录收集，请将设置更改为 PUT (DISABLED)。

SYSTEM.CLUSTER.TRANSMIT.QUEUE
每个队列管理器都有一个名为 SYSTEM.CLUSTER.TRANSMIT.QUEUE 的本地队列的定义。 SYSTEM.CLUSTER.TRANSMIT.QUEUE 是所有消息到集群中所有队列和队列管理器的缺省传输队列。

您可以通过更改队列管理器属性 DEFXMITQ z/OS (在 z/OS 上除外)，将每个集群发送方通道的缺省传输队列更改为 SYSTEM.CLUSTER.TRANSMIT.*ChannelName*。 无法删除 SYSTEM.CLUSTER.TRANSMIT.QUEUE。 它还用于定义授权检查所使用的缺省传输队列是 SYSTEM.CLUSTER.TRANSMIT.QUEUE 还是 SYSTEM.CLUSTER.TRANSMIT.*ChannelName*。

相关概念
缺省集群对象

使用集群传输队列和集群发送方通道

集群队列管理器之间的消息存储在集群传输队列上，并由集群发送方通道转发。 在任何时间点，集群发送方通道都与一个传输队列相关联。 如果更改通道的配置，那么它可能会在下次启动时切换到其他传输队列。 此交换机的处理是自动化的，并且是事务性的。

运行以下 MQSC 命令以显示与集群发送方通道关联的传输队列:

```
DISPLAY CHSTATUS(*) WHERE(CHLTYPE EQ CLUSSDR)
```

```
AMQ8417: Display Channel Status details.
CHANNEL(TO.QM2)          CHLTYPE(CLUSSDR)
CONNAME(9.146.163.190(1416))    CURRENT
RQMNAME(QM2)            STATUS(STOPPED)
SUBSTATE( )            XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

在已停止的集群发送方通道的已保存通道状态中显示的传输队列可能会在该通道再次启动时发生更改。 第 240 页的『按集群发送方通道选择缺省传输队列』 描述了选择缺省传输队列的过程; 第 241 页的『按集群发送方通道选择手动定义的传输队列』 描述了选择手动定义的传输队列的过程。

当任何集群发送方通道启动时，它将重新检查其与传输队列的关联。 如果传输队列的配置或队列管理器缺省值发生更改，那么可能会将通道与另一个传输队列重新关联。 如果由于配置更改而使用其他传输队列重新启动通道，那么将执行将消息传输到新关联的传输队列的过程。 第 242 页的『将集群发送方通道切换到其他传输队列的过程如何工作』 描述了将集群发送方通道从一个传输队列传输到另一个传输队列的过程。

集群发送方通道的行为与发送方通道和服务器通道不同。 它们与同一传输队列保持关联，直到更改通道属性 **XMITQ** 为止。 如果更改发送方或服务器通道上的传输队列属性并将其重新启动，那么不会将消息从旧传输队列传输到新传输队列。

集群发送方通道与发送方通道或服务器通道之间的另一个区别是，多个集群发送方通道可以打开一个集群传输队列，但只有一个发送方通道或服务器通道可以打开一个正常传输队列。 在 IBM WebSphere MQ 7.5 之前，集群连接共享单集群传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE。 从 IBM WebSphere MQ 7.5 开始，您可以选择集群发送方通道不共享传输队列。 不强制实施排他性; 它是配置的结果。 您可以配置消息在集群中所采用的路径，以使其不与在其他应用程序之间流动的消息共享任何传输队列或通道。 请参阅 集群: 规划如何配置集群传输队列 和 第 285 页的『添加集群和集群传输队列以隔离从网关队列管理器发送的集群消息流量』。

按集群发送方通道选择缺省传输队列

集群传输队列是系统缺省队列 (名称以 SYSTEM.CLUSTER.TRANSMIT 开头) 或手动定义的队列。 集群发送方通道通过以下两种方式之一与集群传输队列相关联: 通过缺省集群传输队列机制或通过手动配置。

缺省集群传输队列设置为队列管理器属性 **DEFCLXQ**。 其值为 SCTQ 或 CHANNEL。 新的和迁移的队列管理器设置为 SCTQ。 可以将值更改为 CHANNEL。

如果设置了 SCTQ，那么缺省集群传输队列为 SYSTEM.CLUSTER.TRANSMIT.QUEUE。 每个集群发送方通道都可以打开此队列。 打开队列的集群发送方通道是未与手动定义的集群传输队列相关联的集群发送方通道。

如果设置了 CHANNEL，那么队列管理器可以为每个集群发送方通道创建单独的永久动态传输队列。 每个队列都命名为 SYSTEM.CLUSTER.TRANSMIT. *ChannelName*，并根据模型队列 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 创建。 未与手动定义的集群传输队列相关联的每个集群发

送方通道都与永久动态集群传输队列相关联。 当队列管理器需要此集群发送方通道所服务的集群目标的单独集群传输队列，并且不存在任何队列时，将由该队列管理器创建该队列。

某些集群目标可由与手动定义的传输队列相关联的集群发送方通道提供，而其他集群目标可由一个或多个缺省队列提供。 在集群发送方通道与传输队列的关联中，手动定义的传输队列始终优先于缺省传输队列。

第 241 页的图 39 中说明了集群传输队列的优先顺序。 唯一未与手动定义的集群传输队列关联的集群发送方通道是 CS.QM1。 它未与手动定义的传输队列相关联，因为传输队列的 **CLCHNAME** 属性中没有任何通道名称与 CS.QM1 匹配。

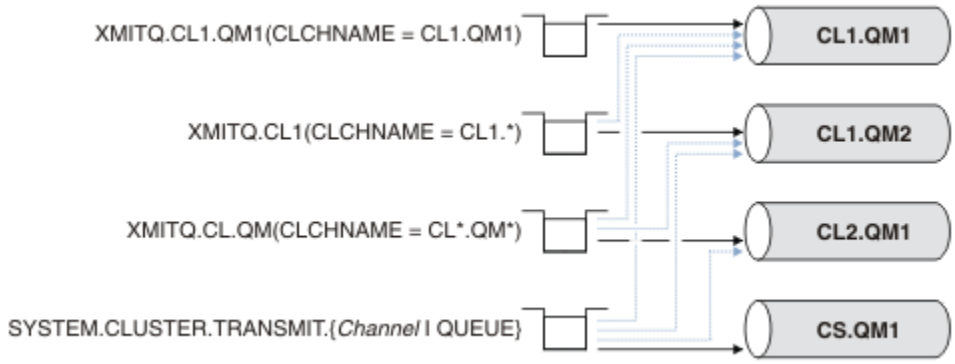


图 *39:* 传输队列/集群发送方通道优先顺序

按集群发送方通道选择手动定义的传输队列

手动定义的队列将传输队列属性 **USAGE** 属性设置为 XMITQ，集群通道名称属性 **CLCHNAME** 设置为特定或通用通道名称。

如果 **CLCHNAME** 队列属性中的名称与集群发送方通道名称匹配，那么该通道将与队列相关联。 如果名称不包含通配符，那么该名称是完全匹配项，如果名称包含通配符，那么该名称是最佳匹配项。

如果多个传输队列上的 **CLCHNAME** 定义与同一集群发送方通道匹配，那么这些定义将表示重叠。 要解决模糊性，在匹配之间存在优先顺序。 精确匹配始终优先。 第 241 页的图 39 显示了传输队列与集群发送方通道之间的关联。 黑色箭头显示实际关联，灰色箭头和潜在关联。 第 241 页的图 39 中传输队列的优先顺序为:

XMITQ.CL1.QM1
传输队列 XMITQ.CL1.QM1 的 **CLCHNAME** 属性设置为 CL1.QM1。 **CLCHNAME** 属性 CL1.QM1 的定义没有通配符，并且优先于其他传输队列上定义的与通配符匹配的任何其他 CLCHNAME 属性。 队列管理器存储要由 XMITQ.CL1.QM1 传输队列上的 CL1.QM1 集群发送方通道传输的任何集群消息。 唯一的例外情况是多个传输队列的 **CLCHNAME** 属性设置为 CL1.QM1。 在这种情况下，队列管理器会将 CL1.QM1 集群发送方通道的消息存储在其中任何队列上。 它在通道启动时任意选择队列。 当通道再次启动时，它可能会选择另一个队列。

XMITQ.CL1
传输队列 XMITQ.CL1 的 **CLCHNAME** 属性设置为 CL1.*。 **CLCHNAME** 属性 CL1.*的定义具有一个尾部通配符，该通配符与以 CL1.开头的任何集群发送方通道的名称相匹配。 队列管理器存储要由其名称以 CL1. 开头的任何集群发送方通道在传输队列 XMITQ.CL1 上传输的任何集群消息，除非存在具有更具体匹配的传输队列 (例如，队列 XMITQ.CL1.QM1)。 一个尾部通配符使定义比没有通配符的定义更不具体，比有多个通配符或后跟更多尾部字符的通配符的定义更具体。

XMITQ.CL.QM
XMITQ.CL.QM 是其 **CLCHNAME** 属性设置为 CL*.QM*的传输队列的名称。 CL*.QM* 的定义有两个通配符，它们与以 CL.开头的任何集群发送方通道的名称相匹配，并且包含或以 QM 结尾。 与使用一个通配符的匹配相比，该匹配的具体程度较低。

SYSTEM.CLUSTER.TRANSMIT. *channelName*|QUEUE
如果没有传输队列具有与队列管理器要使用的集群发送方通道的名称匹配的 **CLCHNAME** 属性，那么队列管理器将使用缺省集群传输队列。 缺省集群传输队列是单个系统集群传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 或队列管理器为特定集群发送方通道

SYSTEM.CLUSTER.TRANSMIT. *channelName* 创建的系统集群传输队列。 哪个队列是缺省队列取决于队列管理器 **DEFXMITQ** 属性的设置。

提示: 除非您明确需要重叠定义，否则请避免使用这些定义，因为它们可能导致难以理解的复杂配置。

将集群发送方通道切换到其他传输队列的过程如何工作

要更改集群发送方通道与集群传输队列的关联，请随时更改任何传输队列的 **CLCHNAME** 参数或队列管理器参数 **DEFCLXQ** 。 不会立即发生任何情况。 仅当通道启动时才会发生更改。 当它启动时，它将检查是否继续从同一传输队列转发消息。 三种类型的更改会改变集群发送方通道与传输队列的关联。

1. 将集群发送方通道当前关联的传输队列的 **CLCHNAME** 参数重新定义为不太具体或为空，或者在通道停止时删除集群传输队列。

   现在，某些其他集群传输队列可能与通道名称更匹配。 或者，如果没有其他传输队列与集群发送方通道的名称匹配，那么关联必须还原为缺省传输队列。
2. 重新定义任何其他集群传输队列的 **CLCHNAME** 参数，或者添加集群传输队列。

   与集群发送方通道当前关联的传输队列相比，另一个传输队列的 **CLCHNAME** 参数现在可能更适合集群发送方通道。 如果集群发送方通道当前与缺省集群传输队列相关联，那么它可能与手动定义的集群传输队列相关联。
3. 如果集群发送方通道当前与缺省集群传输队列相关联，请更改 **DEFCLXQ** 队列管理器参数。

如果集群发送方通道的关联发生更改，那么当该通道启动时，它会将其关联切换到新的传输队列。 在切换期间，将确保不会丢失任何消息。 按通道将消息传输到远程队列管理器的顺序将消息传输到新的传输队列。

切记: 与集群中的任何消息转发一样，您必须将消息放入组中，以确保必须按顺序传递的消息按顺序传递。 在极少数情况下，消息可能会在集群中变得混乱。

切换过程将执行以下事务性步骤。 如果交换机进程中断，那么当通道再次重新启动时，将恢复当前事务步骤。

步骤 1-处理来自原始传输队列的消息

集群发送方通道与新的传输队列相关联，该传输队列可能与其他集群发送方通道共享。 集群发送方通道的消息继续放置在原始传输队列上。 过渡切换进程将消息从原始传输队列传输到新的传输队列。 集群发送方通道将消息从新的传输队列转发到集群接收方通道。 通道状态显示仍与旧传输队列关联的集群发送方通道。

切换过程也会继续传输新到达的消息。 此步骤将继续执行，直到交换机进程要转发的剩余消息数达到零为止。 当消息数达到零时，该过程将移至步骤 2。

在步骤 1 期间，通道的磁盘活动会增加。 持久消息从第一传输队列落实到第二传输队列。 在正常传输消息的过程中，将这些消息放在传输队列上并从传输队列中除去这些消息时，除了此磁盘活动外，还将落实这些消息。 理想情况下，在切换过程中没有任何消息到达，因此可以尽快进行转换。 如果消息确实到达，那么将由切换过程处理这些消息。

步骤 2-处理来自新传输队列的消息

只要集群发送方通道的原始传输队列上没有保留任何消息，就会将新消息直接放置在新的传输队列上。 通道状态显示集群发送方通道与新的传输队列相关联。 以下消息将写入队列管理器错误日志:
“ AMQ7341 **通道** *ChannelName* **的传输队列为** *QueueName* 。”

多个集群传输队列和集群传输队列属性

您可以选择将集群消息转发到不同的队列管理器，这些队列管理器将消息存储在单个集群传输队列或多个队列上。 对于一个队列，您有一组要设置和查询的集群传输队列属性; 对于多个队列，您有多组。 对于某些属性，具有多个集合是一个优势: 例如，查询队列深度会告诉您有多少消息等待由一个或一组通道 (而不是所有通道) 转发。 对于其他属性，具有多个集合是一个缺点: 例如，您可能不想为每个集群传输队列配置相同的访问许可权。 因此，始终会针对 SYSTEM.CLUSTER.TRANSMIT.QUEUE 的概要文件检查访问许可权，而不是针对特定集群传输队列的概要文件检查访问许可权。 如果要应用更精细的安全性检查，请参阅 访问控制和多个集群传输队列。

多个集群发送方通道和多个传输队列

在集群发送方通道上转发消息之前，队列管理器将该消息存储在集群传输队列上。 它选择连接到消息目标的集群发送方通道。 它可以选择所有连接到同一目标的集群发送方通道。 目标可能是同一个物理队列，由多个集群发送方通道连接到单个队列管理器。 目标还可能是多个具有相同队列名称的物理队列，这些队列托管在同一集群中的不同队列管理器上。 如果选择连接到目标的集群发送方通道，那么工作负载均衡算法将选择一个集群发送方通道。 此选项取决于多个因素; 请参阅 集群工作负载管理算法。

在 第 243 页的图 40 中， CL1.QM1， CL1.QM2 和 CS.QM1 都是可能指向同一目标的通道。 例如，如果在 QM1 和 QM2 上的 CL1 中定义 Q1 ，那么 CL1.QM1 和 CL1.QM2 都在两个不同的队列管理器上提供到同一目标 Q1 的路由。 如果通道 CS.QM1 也在 CL1 中，那么它也是 Q1 的消息可以采用的通道。 CS.QM1 的集群成员资格可能由集群名称列表定义，这就是通道名称在其构造中不包含集群名称的原因。 根据工作负载均衡参数和发送应用程序， Q1 的某些消息可能放置在每个传输队列 XMITQ.CL1.QM1， XMITQ.CL1 和 SYSTEM.CLUSTER.TRANSMIT.CS.QM1 上。

如果您打算分离出消息流量，以便同一目标的消息不会与不同目标的消息共享队列或通道，那么必须考虑如何首先将流量划分到不同的集群发送方通道上，然后如何将特定通道的消息划分到不同的传输队列上。 同一集群上的集群队列，在同一队列管理器上，通常共享相同的集群通道。 仅定义多个集群传输队列不足以将集群消息流量分隔到不同的队列上。 除非将不同目标队列的消息分隔到不同的通道上，否则这些消息共享同一个集群传输队列。

用于分隔消息所采用的通道的简单方法是创建多个集群。 在每个集群中的任何队列管理器上，仅定义一个集群队列。 然后，如果为每个集群/队列管理器组合定义不同的集群接收方通道，那么每个集群队列的消息不会与其他集群队列的消息共享集群通道。 如果为集群通道定义单独的传输队列，那么发送队列管理器仅为每个传输队列上的一个集群队列存储消息。 例如，如果您希望两个集群队列不共享资源，那么可以将它们放在同一队列管理器上的不同集群中，也可以放在同一集群中的不同队列管理器上。

选择集群传输队列不会影响工作负载均衡算法。 工作负载均衡算法选择要转发消息的集群发送方通道。 它将消息放在由该通道服务的传输队列上。 如果调用工作负载均衡算法以再次选择 (例如，如果通道停止) ，那么它可能能够选择另一个通道来转发消息。 如果它确实选择了另一个通道，并且新通道转发来自另一个集群传输队列的消息，那么工作负载均衡算法会将该消息传输到另一个传输队列。

在 第 243 页的图 40 中，两个集群发送方通道 CS.QM1 和 CS.QM2 与缺省系统传输队列相关联。 当工作负载均衡算法在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 或任何其他集群传输队列上存储消息时，用于转发消息的集群发送方通道的名称将存储在消息的相关标识中。 每个通道仅转发那些与通道名称的相关标识匹配的消息。

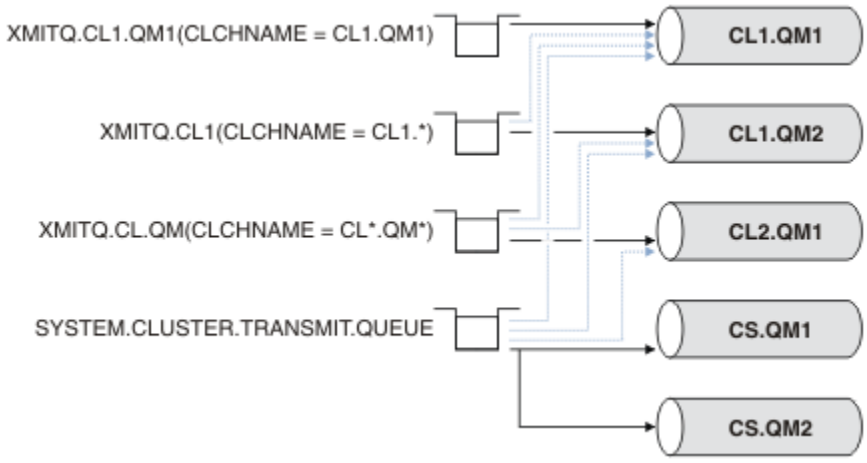


图 *40:* 多个集群发送方通道

如果 CS.QM1 停止，那么将检查该集群发送方通道的传输队列上的消息。 可以由另一个通道转发的那些消息将由工作负载均衡算法重新处理。 它们的相关标识将重置为备用集群发送方通道名称。 如果备用集群发送方通道为 CS.QM2，那么消息将保留在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上。 如果备用通道为 CL1.QM1，那么工作负载均衡算法会将消息传输到 XMITQ.CL1.QM1。 当集群发送方通道重新启动时，该通道将再次传输新消息以及未针对其他集群发送方通道标记的消息。

您可以更改正在运行的系统上的传输队列与集群发送方通道之间的关联。 您可以更改传输队列上的 **CLCHNAME** 参数，或者更改 **DEFCLXQ** 队列管理器参数。 当受更改影响的通道重新启动时，它将启动传输队列切换过程; 请参阅 第 242 页的『将集群发送方通道切换到其他传输队列的过程如何工作』。

切换传输队列的过程在通道重新启动时启动。 工作负载重新平衡过程在通道停止时启动。 这两个进程可以并行运行。

简单的情况是，停止集群发送方通道不会导致重新平衡进程更改集群发送方通道，即转发队列上的任何消息。 在这种情况下，任何其他集群发送方通道都无法将消息转发到正确的目标。 如果没有用于将消息转发到其目标的备用集群发送方通道，那么在集群发送方通道停止后，将对同一集群发送方通道保持标记消息。 当通道启动时，如果交换机处于暂挂状态，那么切换进程会将消息移至另一传输队列，在该队列中，这些消息由同一集群发送方通道处理。

更复杂的情况是，多个集群发送方通道可以处理到同一目标的某些消息。 停止并重新启动集群发送方通道以触发传输队列开关。 在许多情况下，在重新启动通道时，工作负载均衡算法已经将消息从原始传输队列移至由不同集群发送方通道提供服务的不同传输队列。 只有那些无法由其他集群发送方通道转发的消息仍要传输到新的传输队列。 在某些情况下，如果通道快速重新启动，那么可以由工作负载均衡算法传输的某些消息将保留。 在这种情况下，某些剩余消息由工作负载均衡进程切换，而某些消息由切换传输队列的进程切换。

相关概念

集群通道

第 532 页的『计算日志大小』
估算队列管理器需要的日志大小。

相关任务

集群: 使用多个集群传输队列进行应用程序隔离

集群: 规划如何配置集群传输队列

第 275 页的『使用网关队列管理器创建两个重叠的集群』
按照任务中的指示信息来构造具有网关队列管理器的重叠集群。 将集群用作以下示例的起始点，这些示例用于将消息从一个应用程序中的消息隔离到集群中的其他应用程序。

第 256 页的『将队列管理器添加到集群: 单独的传输队列』
遵循以下指示信息将队列管理器添加到您创建的集群。 使用多个集群传输队列将消息传输到集群队列和主题。

第 282 页的『添加集群传输队列以隔离从网关队列管理器发送的集群消息流量』
修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 传输到应用程序之后，解决方案会使用额外的集群传输队列将消息流量分离到集群中的单个队列管理器。

第 285 页的『添加集群和集群传输队列以隔离从网关队列管理器发送的集群消息流量』
修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 传输到应用程序之后，解决方案会使用其他集群将消息隔离到特定集群队列。

设置新集群

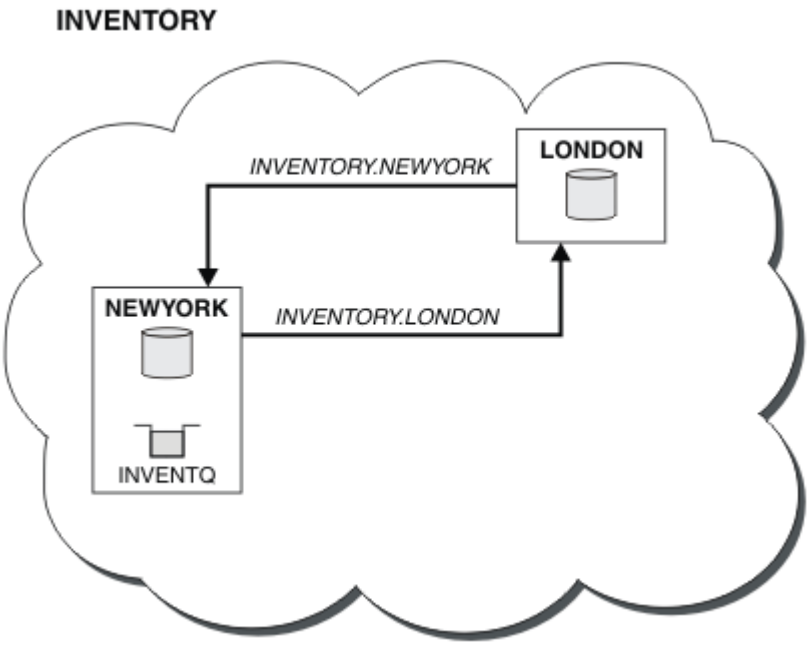
遵循以下指示信息来设置示例集群。 单独的指示信息描述在 TCP/IP 上设置集群， LU 6.2 以及使用单个传输队列或多个传输队列。 通过将消息从一个队列管理器发送到另一个队列管理器来测试集群工作。

开始之前

- 您可以使用随 IBM MQ Explorer 提供的其中一个向导来创建集群，例如此任务创建的集群，而不是遵循这些指示信息。 右键单击 "队列管理器集群" 文件夹，然后单击 **新建 > 队列管理器集群**，并遵循向导中给出的指示信息。
- 有关帮助您了解设置集群所执行的步骤的背景信息，请参阅 第 236 页的『定义集群队列』， 集群通道 和侦听器。

关于此任务

您正在为链式商店设置新的 IBM MQ 网络。 该店有两家分店，一家在伦敦，一家在纽约。 每个商店的数据和应用程序由运行单独队列管理器的系统托管。 这两个队列管理器称为 LONDON 和 NEWYORK。 库存应用程序在纽约的系统上运行，连接到队列管理器 NEWYORK。 应用程序由 NEWYORK 托管的 INVENTQ 队列上的消息到达驱动。 将在名为 INVENTORY 的集群中链接两个队列管理器 LONDON 和 NEWYORK，以便它们都可以将消息放入 INVENTQ。



此集群如下所示:

您可以配置集群中的每个队列管理器，以使用不同的集群传输队列将消息发送到集群中的其他队列管理器。

设置集群的指示信息因传输协议，传输队列数或平台而异。 您可以选择三种组合。 所有组合的验证过程保持不变。

INVENTORY 是一个小型集群。 但是，它作为概念的证明是有用的。 了解此集群的重要内容是它为未来增强功能提供的作用域。

过程

- 第 245 页的『使用 TCP/IP 设置集群，每个队列管理器具有单个传输队列』
- 第 248 页的『使用每个队列管理器的多个传输队列在 TCP/IP 上设置集群』
- 第 250 页的『在 z/OS 上使用 LU 6.2 设置集群』
- 第 253 页的『验证集群』

相关概念

集群

集群与分布式排队的比较

集群的组件

相关任务

第 235 页的『配置队列管理器集群』

集群提供了一种用于以简化初始配置和持续管理的方式互连队列管理器的机制。 您可以定义集群组件，以及创建和管理集群。

使用 *TCP/IP* 设置集群，每个队列管理器具有单个传输队列

这是描述简单集群的不同配置的三个主题之一。

开始之前

有关正在创建的集群的概述，请参阅 第 244 页的『设置新集群』。

队列管理器属性 **DEFCLXQ** 必须保留为其缺省值 SCTQ。

关于此任务

执行以下步骤以使用传输协议 TCP/IP 在 多平台 上设置集群。 z/OS 在 z/OS 上，必须遵循 第 816 页的『在 z/OS 上定义 TCP 连接』 中的指示信息来设置 TCP/IP 连接，而不是在步骤 第 246 页的『4』 中定义侦听器。 否则， z/OS 的步骤相同，但会将错误消息写入控制台，而不是写入队列管理器错误日志。

过程

1. 决定集群的组织及其名称。

   您决定将两个队列管理器 LONDON 和 NEWYORK 链接到集群中。 只有两个队列管理器的集群在使用分布式排队的网络上仅提供边际优势。 这是一个很好的启动方法，它为未来的扩展提供了空间。 打开商店的新分支时，可以轻松将新的队列管理器添加到集群中。 添加新的队列管理器不会破坏现有网络; 请参阅 第 254 页的『将队列管理器添加至集群』。

   目前，您正在运行的唯一应用程序是库存应用程序。 集群名称为 INVENTORY。

2. 确定哪些队列管理器将保存完整存储库。

   在任何集群中，必须至少指定一个队列管理器，或者最好指定两个队列管理器，以保存完整存储库。 在此示例中，只有两个队列管理器 LONDON 和 NEWYORK，这两个队列管理器都保存完整存储库。

   a. 您可以按任何顺序执行其余步骤。

   b. 继续执行这些步骤时，可能会将警告消息写入队列管理器日志。 这些消息是您尚未添加的定义缺失的结果。

   ```
   Examples of the responses to the commands are shown in a box
   like this after each step in this task.
   These examples show the responses returned by IBM MQ for AIX.
   The responses vary on other platforms.
   ```

   c. 在继续执行这些步骤之前，请确保队列管理器已启动。

3. 更改队列管理器定义以添加存储库定义。

   在要保存完整存储库的每个队列管理器上，使用 ALTER QMGR 命令并指定 REPOS 属性来变更本地队列管理器定义:

   ```
   ALTER QMGR REPOS(INVENTORY)
   ```

   ```
   1 : ALTER QMGR REPOS(INVENTORY)
   AMQ8005: IBM MQ queue manager changed.
   ```

   例如，如果输入:

   a. `runmqsc LONDON`

   b. `ALTER QMGR REPOS(INVENTORY)`

   LONDON 已更改为完整存储库。

4. 定义侦听器。

   定义一个侦听器，用于为集群中的每个队列管理器接受来自其他队列管理器的网络请求。 在 LONDON 队列管理器上，发出以下命令:

   ```
   DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
   ```

   CONTROL 属性确保侦听器在队列管理器启动时启动和停止。

未在定义侦听器时将其启动，因此必须使用以下 MQSC 命令首次手动启动该侦听器:

```
START LISTENER(LONDON_LS)
```

对集群中的所有其他队列管理器发出类似的命令，从而更改每个队列管理器的侦听器名称。

有多种方法来定义这些侦听器，如 侦听器中所示。

5. 为 LONDON 队列管理器定义 CLUSRCVR 通道。

   在集群中的每个队列管理器上，您定义一个集群接收方通道，队列管理器可以在该通道上接收消息。 请参阅 集群接收方通道: CLUSRCVR 。 CLUSRCVR 通道定义队列管理器的连接名称。 连接名称存储在存储库中，其他队列管理器可以在这些存储库中引用该名称。 CLUSTER 关键字显示队列管理器从集群中的其他队列管理器接收消息的可用性。

   在此示例中，通道名称为 INVENTORY.LONDON，连接名称 (CONNAME) 是队列管理器所在机器的网络地址，即 LONDON.CHSTORE.COM。 网络地址可以输入为字母数字 DNS 主机名，也可以输入为 IPv4 点分十进制格式的 IP 地址。 例如， 192.0.2.0 或 IPv6 十六进制格式; 例如 2001:DB8:0204:acff:fe97:2c34:fde0:3485。 未指定端口号，因此将使用缺省端口 (1414)。

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-receiver channel for queue manager LONDON')
   ```

   ```
   1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-receiver channel for queue manager LONDON')
   AMQ8014: WebSphere MQ channel created.
   07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
   ```

6. 为 NEWYORK 队列管理器定义 CLUSRCVR 通道。

   如果通道侦听器正在使用缺省端口 (通常为 1414) ，并且集群在 z/OS 上不包含队列管理器，那么可以省略 CONNAME

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
   ```

7. 在 LONDON 队列管理器上定义 CLUSSDR 通道。

   您可以手动定义从每个完整存储库队列管理器到集群中每个其他完整存储库队列管理器的 CLUSSDR 通道。 请参阅 集群发送方通道: CLUSSDR 。 在这种情况下，只有两个队列管理器，这两个队列管理器都保存完整存储库。 它们都需要一个手动定义的 CLUSSDR 通道，该通道指向另一个队列管理器上定义的 CLUSRCVR 通道。 CLUSSDR 定义上给出的通道名称必须与相应 CLUSRCVR 定义上的通道名称匹配。 当队列管理器具有同一集群中的集群接收方通道和集群发送方通道的定义时，将启动集群发送方通道。

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
   ```

   ```
   1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
   AMQ8014: WebSphere MQ channel created.
   07/09/98 13:00:18  Channel program started.
   ```

8. 在 NEWYORK 队列管理器上定义 CLUSSDR 通道。

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
   ```

9. 定义集群队列 INVENTQ

在 NEWYORK 队列管理器上定义 INVENTQ 队列，并指定 CLUSTER 关键字。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

CLUSTER 关键字导致将队列通告到集群。 一旦定义了队列，它就可供集群中的其他队列管理器使用。 他们可以向其发送消息，而不必为其进行远程队列定义。

所有定义都已完成。 在所有平台上，在每个队列管理器上启动侦听器程序。 侦听器程序等待入局网络请求，并在需要时启动集群接收方通道。

下一步做什么

现在，您已准备好 验证集群。

相关任务

第 248 页的『使用每个队列管理器的多个传输队列在 TCP/IP 上设置集群』
这是描述简单集群的不同配置的三个主题之一。

第 250 页的『在 z/OS 上使用 LU 6.2 设置集群』
这是描述简单集群的不同配置的树主题之一。

使用每个队列管理器的多个传输队列在 *TCP/IP* 上设置集群

这是描述简单集群的不同配置的三个主题之一。

开始之前

有关正在创建的集群的概述，请参阅 第 244 页的『设置新集群』。

关于此任务

执行以下步骤以使用传输协议 TCP/IP 在 多平台 上设置集群。 存储库队列管理器配置为使用不同的集群传输队列将消息相互发送，并发送到集群中的其他队列管理器。 如果将队列管理器添加到同样要使用不同传输队列的集群，请执行任务 第 256 页的『将队列管理器添加到集群: 单独的传输队列』。

过程

1. 决定集群的组织及其名称。

   您决定将两个队列管理器 LONDON 和 NEWYORK 链接到集群中。 只有两个队列管理器的集群在使用分布式排队的网络上仅提供边际优势。 这是一个很好的启动方法，它为未来的扩展提供了空间。 打开商店的新分支时，可以轻松将新的队列管理器添加到集群中。 添加新的队列管理器不会破坏现有网络; 请参阅 第 254 页的『将队列管理器添加至集群』。

   目前，您正在运行的唯一应用程序是库存应用程序。 集群名称为 INVENTORY。

2. 确定哪些队列管理器将保存完整存储库。

   在任何集群中，必须至少指定一个队列管理器，或者最好指定两个队列管理器，以保存完整存储库。 在此示例中，只有两个队列管理器 LONDON 和 NEWYORK，这两个队列管理器都保存完整存储库。

   a. 您可以按任何顺序执行其余步骤。

   b. 继续执行这些步骤时，可能会将警告消息写入队列管理器日志。 这些消息是您尚未添加的定义缺失的结果。

   ```
   Examples of the responses to the commands are shown in a box
   like this after each step in this task.
   These examples show the responses returned by IBM MQ for AIX.
   The responses vary on other platforms.
   ```

   c. 在继续执行这些步骤之前，请确保队列管理器已启动。

3. 更改队列管理器定义以添加存储库定义。

   在要保存完整存储库的每个队列管理器上，使用 ALTER QMGR 命令并指定 REPOS 属性来变更本地队列管理器定义:

   ```
   ALTER QMGR REPOS(INVENTORY)
   ```

   ```
   1 : ALTER QMGR REPOS(INVENTORY)
   AMQ8005: IBM MQ queue manager changed.
   ```

   例如，如果输入:

   a. `runmqsc LONDON`

   b. `ALTER QMGR REPOS(INVENTORY)`

   LONDON 已更改为完整存储库。

4. 更改队列管理器定义，以便为每个目标创建单独的集群传输队列。

   ```
   ALTER QMGR DEFCLXQ(CHANNEL)
   ```

   在添加到集群的每个队列管理器上，决定是否使用单独的传输队列。 请参阅主题 第 254 页的『将队列管理器添加至集群』 和 第 256 页的『将队列管理器添加到集群: 单独的传输队列』。

5. 定义侦听器。

   定义一个侦听器，用于为集群中的每个队列管理器接受来自其他队列管理器的网络请求。 在 LONDON 队列管理器上，发出以下命令:

   ```
   DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
   ```

   CONTROL 属性确保侦听器在队列管理器启动时启动和停止。

   未在定义侦听器时将其启动，因此必须使用以下 MQSC 命令首次手动启动该侦听器:

   ```
   START LISTENER(LONDON_LS)
   ```

   对集群中的所有其他队列管理器发出类似的命令，从而更改每个队列管理器的侦听器名称。

   有多种方法来定义这些侦听器，如 侦听器中所示。

6. 为 LONDON 队列管理器定义 CLUSRCVR 通道。

   在集群中的每个队列管理器上，您定义一个集群接收方通道，队列管理器可以在该通道上接收消息。 请参阅 集群接收方通道: CLUSRCVR 。 CLUSRCVR 通道定义队列管理器的连接名称。 连接名称存储在存储库中，其他队列管理器可以在这些存储库中引用该名称。 CLUSTER 关键字显示队列管理器从集群中的其他队列管理器接收消息的可用性。

   在此示例中，通道名称为 INVENTORY.LONDON，连接名称 (CONNAME) 是队列管理器所在机器的网络地址，即 LONDON.CHSTORE.COM。 网络地址可以输入为字母数字 DNS 主机名，也可以输入为 IPv4 点分十进制格式的 IP 地址。 例如， 192.0.2.0 或 IPv6 十六进制格式; 例如 2001:DB8:0204:acff:fe97:2c34:fde0:3485。 未指定端口号，因此将使用缺省端口 (1414)。

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-receiver channel for queue manager LONDON')
   ```

   ```
   1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-receiver channel for queue manager LONDON')
   AMQ8014: WebSphere MQ channel created.
   07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
   ```

7. 为 NEWYORK 队列管理器定义 CLUSRCVR 通道。

如果通道侦听器正在使用缺省端口 (通常为 1414)，并且集群在 z/OS 上不包含队列管理器，那么可以省略 CONNAME

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

8. 在 LONDON 队列管理器上定义 CLUSSDR 通道。

   您可以手动定义从每个完整存储库队列管理器到集群中每个其他完整存储库队列管理器的 CLUSSDR 通道。 请参阅 集群发送方通道: CLUSSDR 。 在这种情况下，只有两个队列管理器，这两个队列管理器都保存完整存储库。 它们都需要一个手动定义的 CLUSSDR 通道，该通道指向另一个队列管理器上定义的 CLUSRCVR 通道。 CLUSSDR 定义上给出的通道名称必须与相应 CLUSRCVR 定义上的通道名称匹配。 当队列管理器具有同一集群中的集群接收方通道和集群发送方通道的定义时，将启动集群发送方通道。

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
   ```

   ```
   1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
   AMQ8014: WebSphere MQ channel created.
   07/09/98 13:00:18  Channel program started.
   ```

9. 在 NEWYORK 队列管理器上定义 CLUSSDR 通道。

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
   ```

10. 定义集群队列 INVENTQ

    在 NEWYORK 队列管理器上定义 INVENTQ 队列，并指定 CLUSTER 关键字。

    ```
    DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
    ```

    ```
    1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
    AMQ8006: WebSphere MQ queue created.
    ```

    CLUSTER 关键字导致将队列通告到集群。 一旦定义了队列，它就可供集群中的其他队列管理器使用。 他们可以向其发送消息，而不必为其进行远程队列定义。

    所有定义都已完成。 在所有平台上，在每个队列管理器上启动侦听器程序。 侦听器程序等待入局网络请求，并在需要时启动集群接收方通道。

下一步做什么

现在，您已准备好 验证集群。

相关任务

第 245 页的『使用 TCP/IP 设置集群，每个队列管理器具有单个传输队列』
这是描述简单集群的不同配置的三个主题之一。

第 250 页的『在 z/OS 上使用 LU 6.2 设置集群』
这是描述简单集群的不同配置的树主题之一。

在 *z/OS* 上使用 *LU 6.2* 设置集群

这是描述简单集群的不同配置的树主题之一。

开始之前

有关正在创建的集群的概述，请参阅 第 244 页的『设置新集群』。

过程

1. 决定集群的组织及其名称。

   您决定将两个队列管理器 LONDON 和 NEWYORK 链接到集群中。 只有两个队列管理器的集群在使用分布式排队的网络上仅提供边际优势。 这是一个很好的启动方法，它为未来的扩展提供了空间。 打开商店的新分支时，可以轻松将新的队列管理器添加到集群中。 添加新的队列管理器不会破坏现有网络; 请参阅第 254 页的『将队列管理器添加至集群』。

   目前，您正在运行的唯一应用程序是库存应用程序。 集群名称为 INVENTORY。

2. 确定哪些队列管理器将保存完整存储库。

   在任何集群中，必须至少指定一个队列管理器，或者最好指定两个队列管理器，以保存完整存储库。 在此示例中，只有两个队列管理器 LONDON 和 NEWYORK，这两个队列管理器都保存完整存储库。

   a. 您可以按任何顺序执行其余步骤。

   b. 继续执行这些步骤时，可能会将警告消息写入 z/OS 系统控制台。 这些消息是您尚未添加的定义缺失的结果。

   c. 在继续执行这些步骤之前，请确保队列管理器已启动。

3. 更改队列管理器定义以添加存储库定义。

   在要保存完整存储库的每个队列管理器上，使用 ALTER QMGR 命令并指定 REPOS 属性来变更本地队列管理器定义:

   ```
   ALTER QMGR REPOS(INVENTORY)
   ```

   ```
   1 : ALTER QMGR REPOS(INVENTORY)
   AMQ8005: IBM MQ queue manager changed.
   ```

   例如，如果输入:

   a. `runmqsc LONDON`

   b. `ALTER QMGR REPOS(INVENTORY)`

   LONDON 已更改为完整存储库。

4. 定义侦听器。

   z/OS 请参阅 z/OS 和第 819 页的『在 LU 6.2 上接收』上的通道启动程序。

   未在定义侦听器时将其启动，因此必须使用以下 MQSC 命令首次手动启动该侦听器:

   ```
   START LISTENER(LONDON_LS)
   ```

   对集群中的所有其他队列管理器发出类似的命令，从而更改每个队列管理器的侦听器名称。

5. 为 LONDON 队列管理器定义 CLUSRCVR 通道。

   在集群中的每个队列管理器上，您定义一个集群接收方通道，队列管理器可以在该通道上接收消息。 请参阅 集群接收方通道: CLUSRCVR 。 CLUSRCVR 通道定义队列管理器的连接名称。 连接名称存储在存储库中，其他队列管理器可以在这些存储库中引用该名称。 CLUSTER 关键字显示队列管理器从集群中的其他队列管理器接收消息的可用性。

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
   CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
   MODENAME('#INTER') TPNAME('MQSERIES')
   DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
   ```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

6. 为 NEWYORK 队列管理器定义 CLUSRCVR 通道。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager NEWYORK')
```

7. 在 LONDON 队列管理器上定义 CLUSSDR 通道。

   您可以手动定义从每个完整存储库队列管理器到集群中每个其他完整存储库队列管理器的 CLUSSDR 通道。 请参阅 集群发送方通道: CLUSSDR。 在这种情况下，只有两个队列管理器，这两个队列管理器都保存完整存储库。 它们都需要一个手动定义的 CLUSSDR 通道，该通道指向另一个队列管理器上定义的 CLUSRCVR 通道。 CLUSSDR 定义上给出的通道名称必须与相应 CLUSRCVR 定义上的通道名称匹配。 当队列管理器具有同一集群中的集群接收方通道和集群发送方通道的定义时，将启动集群发送方通道。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(CPIC) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18  Channel program started.
```

8. 在 NEWYORK 队列管理器上定义 CLUSSDR 通道。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from NEWYORK to repository at LONDON')
```

9. 定义集群队列 INVENTQ

   在 NEWYORK 队列管理器上定义 INVENTQ 队列，并指定 CLUSTER 关键字。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

   CLUSTER 关键字导致将队列通告到集群。 一旦定义了队列，它就可供集群中的其他队列管理器使用。 他们可以向其发送消息，而不必为其进行远程队列定义。

   所有定义都已完成。 在所有平台上，在每个队列管理器上启动侦听器程序。 侦听器程序等待入局网络请求，并在需要时启动集群接收方通道。

下一步做什么

现在，您已准备好 验证集群。

相关任务

第 245 页的『使用 TCP/IP 设置集群，每个队列管理器具有单个传输队列』
这是描述简单集群的不同配置的三个主题之一。

第 248 页的『使用每个队列管理器的多个传输队列在 TCP/IP 上设置集群』
这是描述简单集群的不同配置的三个主题之一。

验证集群

对等主题描述简单集群的三种不同配置。 本主题说明如何验证集群。

开始之前

本主题假定您正在验证通过下列其中一项任务创建的集群:

- 第 245 页的『使用 TCP/IP 设置集群，每个队列管理器具有单个传输队列』.
- 第 248 页的『使用每个队列管理器的多个传输队列在 TCP/IP 上设置集群』.
- 第 250 页的『在 z/OS 上使用 LU 6.2 设置集群』.

有关已创建的集群的概述，请参阅 第 244 页的『设置新集群』。

关于此任务

您可以通过以下一种或多种方式验证集群:

1. 运行管理命令以显示集群和通道属性。
2. 运行样本程序以在集群队列上发送和接收消息。
3. 编写您自己的程序以将请求消息发送到集群队列，并将响应消息回复到非集群应答队列。

过程

发出 DISPLAY **runmqsc** 命令以验证集群。

您看到的响应应该与后续步骤中的响应类似。

1. 从 NEWYORK 队列管理器中，运行 **DISPLAY CLUSQMGR** 命令:

```
dis clusqmgr(*)
```

```
1 : dis clusqmgr(*)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(NEWYORK)       CLUSTER(INVENTORY)
CHANNEL(INVENTORY.NEWYORK)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(LONDON)       CLUSTER(INVENTORY)
CHANNEL(INVENTORY.LONDON)
```

2. 从 NEWYORK 队列管理器中，运行 **DISPLAY CHANNEL STATUS** 命令:

```
dis chstatus(*)
```

```
1 : dis chstatus(*)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.NEWYORK)  XMITQ( )
CONNAME(192.0.2.0)      CURRENT
CHLTYPE(CLUSRCVR)       STATUS(RUNNING)
RQMNAME(LONDON)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.LONDON) XMITQ(SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON)
CONNAME(192.0.2.1)      CURRENT
CHLTYPE(CLUSSDR)       STATUS(RUNNING)
RQMNAME(LONDON)
```

使用 **amqsput** 在两个队列管理器之间发送消息。

3. 在 LONDON 上，运行命令 **amqsput INVENTQ LONDON**。

   输入一些消息，后跟空白行。

4. 在 NEWYORK 上，运行命令 **amqsget INVENTQ NEWYORK**。

   现在，您将看到在 LONDON 上输入的消息。 15 秒后，程序结束。

使用您自己的程序在两个队列管理器之间发送消息。

在以下步骤中， LONDON 将消息放入 NEWYORK 处的 INVENTQ ，并在其队列 LONDON_reply 上接收应答。

5. 在 LONDON 上，将消息放入集群队列。
   a) 定义名为 LONDON_reply 的本地队列。
   b) 将 MQOPEN 选项设置为 MQOO_OUTPUT。
   c) 发出 MQOPEN 调用以打开队列 INVENTQ。
   d) 将报文描述符中的 *ReplyToQ* 名称设置为 "LONDON_reply。
   e) 发出 MQPUT 调用以放入消息。
   f) 落实消息。
6. 在 NEWYORK 上，接收集群队列上的消息，并将应答放入应答队列。
   a) 将 MQOPEN 选项设置为 MQOO_BROWSE。
   b) 发出 MQOPEN 调用以打开队列 INVENTQ。
   c) 发出 MQGET 调用以从 INVENTQ 获取消息。
   d) 从报文描述符中读取 *ReplyToQ* 名称。
   e) 将 *ReplyToQ* 名称写入对象描述符的 ObjectName 字段。
   f) 将 MQOPEN 选项设置为 MQOO_OUTPUT。
   g) 发出 MQOPEN 调用以在队列管理器 LONDON 上打开 LONDON_reply 。
   h) 发出 MQPUT 调用以将消息放入 LONDON_reply。
7. 在 LONDON 上，接收应答。
   a) 将 MQOPEN 选项设置为 MQOO_BROWSE。
   b) 发出 MQOPEN 调用以打开队列 LONDON_reply。
   c) 发出 MQGET 调用以从 LONDON_reply 获取消息。

将队列管理器添加至集群

遵循以下指示信息将队列管理器添加到您创建的集群。 使用单集群传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 将消息传输到集群队列和主题。

开始之前

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

场景:

- 如 第 244 页的『设置新集群』中所述设置 INVENTORY 集群。 它包含两个队列管理器 LONDON 和 NEWYORK，这两个队列管理器都保存完整存储库。
- 队列管理器 PARIS 由主安装拥有。 如果不是，那么必须运行 **setmqenv** 命令来为 PARIS 所属的安装设置命令环境。
- 所有三个系统之间都存在 TCP 连接，并且使用在队列管理器控制下启动的 TCP 侦听器配置队列管理器。

关于此任务

1. 正在巴黎设置链式商店的新分支，您希望将名为 PARIS 的队列管理器添加到集群中。
2. 队列管理器 PARIS 通过将消息放入 INVENTQ 队列，将库存更新发送到在纽约系统上运行的应用程序。

执行以下步骤以将队列管理器添加到集群。

过程

1. 决定首先引用哪个完整存储库 PARIS 。

   集群中的每个队列管理器都必须引用一个或多个完整存储库。 它从完整存储库收集有关集群的信息，因此构建自己的部分存储库。 选择任一存储库作为完整存储库。 一旦将新的队列管理器添加到集群，它就

会立即了解其他存储库。 有关对队列管理器的更改的信息将直接发送到两个存储库。 在此示例中，仅出于地理原因，您将 PARIS 链接到队列管理器 LONDON。

注: 在启动队列管理器 PARIS 后，按任何顺序执行其余步骤。

2. 在队列管理器 PARIS 上定义 CLUSRCVR 通道。

   集群中的每个队列管理器都必须定义集群接收方通道，在该通道上可以接收消息。 在 PARIS 上，定义:

   ```
   DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for queue manager PARIS')
   ```

   集群接收方通道会公布队列管理器的可用性，以便从集群 INVENTORY 中的其他队列管理器接收消息。 请勿在其他队列管理器上为集群接收方通道 INVENTORY.PARIS 的发送端创建定义。 需要时，将自动生成其他定义。 请参阅 集群通道 (Cluster channel)。

3. z/OS

   在 IBM MQ for z/OS 上启动通道启动程序。

4. 在队列管理器 PARIS 上定义 CLUSSDR 通道。

   向集群添加不是完整存储库的队列管理器时，仅定义一个集群发送方通道以建立与完整存储库的初始连接。 请参阅 集群发送方通道: CLUSSDR 。

   在 PARIS 上，将名为 INVENTORY.LONDON 的 CLUSSDR 通道定义为具有网络地址 LONDON.CHSTORE.COM 的队列管理器。

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from PARIS to repository at LONDON')
   ```

5. 可选： 如果要将先前已从同一集群中除去的队列管理器添加到集群，请检查该队列管理器现在是否显示为集群成员。 否则，请完成以下额外步骤:

   a) 在要添加的队列管理器上发出 **REFRESH CLUSTER** 命令。

   此步骤将停止集群通道，并为本地集群高速缓存提供一组新的序列号，确保这些序列号在集群的其余部分中是最新的。

   ```
   REFRESH CLUSTER(INVENTORY) REPOS(YES)
   ```

   注: 对于大型集群，使用 **REFRESH CLUSTER** 命令可能会对正在运行的集群造成干扰，此外每隔 27 天在集群对象向所有相关队列管理器自动发送状态更新时也可能有干扰。 请参阅在大型集群中刷新可能会影响集群的性能和可用性。

   b) 重新启动 CLUSSDR 通道

   (例如，使用 START CHANNEL 命令)。

   c) 重新启动 CLUSRCVR 通道。

结果

下图显示了此任务设置的集群。

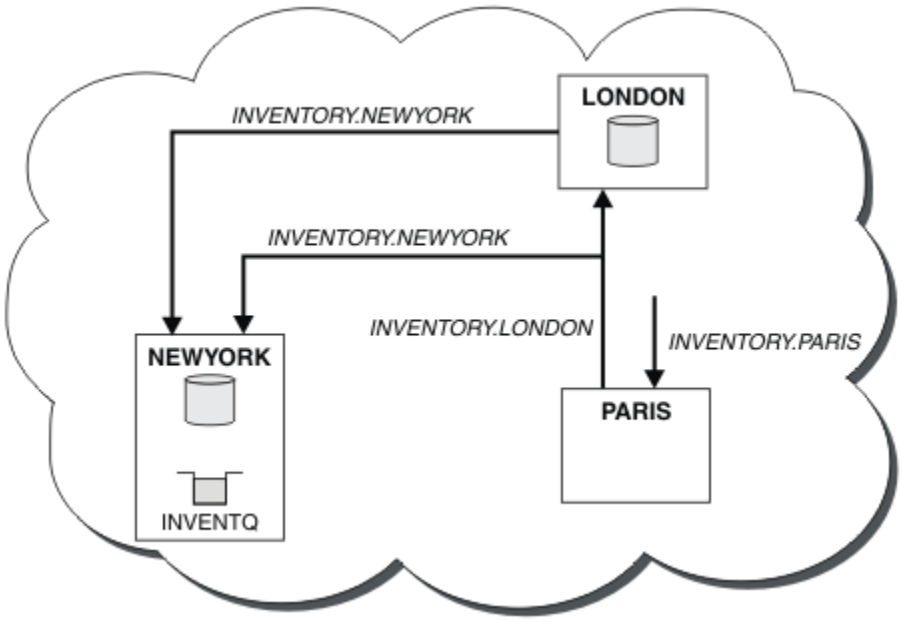


图 *41:* 具有三个队列管理器的 *INVENTORY* 集群

通过仅创建两个定义 (CLUSRCVR 定义和 CLUSSDR 定义) ，我们将队列管理器 PARIS 添加到集群中。

现在， PARIS 队列管理器从位于 LONDON 的完整存储库中了解到 INVENTQ 队列由队列管理器 NEWYORK 托管。 当巴黎系统托管的应用程序尝试将消息放入 INVENTQ 时， PARIS 会自动定义集群发送方通道以连接到集群接收方通道 INVENTORY.NEWYORK。 当将其队列管理器名称指定为目标队列管理器并提供应答队列时，应用程序可以接收响应。

将队列管理器添加到集群: 单独的传输队列

遵循以下指示信息将队列管理器添加到您创建的集群。 使用多个集群传输队列将消息传输到集群队列和主题。

开始之前

- 队列管理器不是任何集群的成员。
- 集群存在; 有一个完整的存储库，此队列管理器可以直接连接到该存储库，并且该存储库可用。 有关创建集群的步骤，请参阅 第 244 页的『设置新集群』。

关于此任务

此任务是 第 254 页的『将队列管理器添加至集群』的替代任务，在此任务中，您将队列管理器添加到将集群消息放置在单个传输队列上的集群。

在此任务中，您将队列管理器添加到集群，该集群会自动为每个集群发送方通道创建单独的集群传输队列。

要使队列的定义数保持较小，缺省值是使用单个传输队列。 如果要监视发送到不同队列管理器和不同集群的流量，那么使用单独的传输队列是有利的。 您可能还希望分离到不同目标的流量，以实现隔离或性能目标。

过程

1. 更改缺省集群通道传输队列类型。

   更改队列管理器 PARIS:

   ```
   ALTER QMGR DEFCLXQ(CHANNEL)
   ```

   队列管理器每次创建集群发送方通道以将消息发送到队列管理器时，都会创建集群传输队列。 传输队列仅供此集群发送方通道使用。 传输队列是永久动态的。 它是从名为

SYSTEM.CLUSTER.TRANSMIT.*ChannelName* 的模型队列
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 创建的。

注意: 如果要将专用 SYSTEM.CLUSTER.TRANSMIT.QUEUES 与从低于 IBM WebSphere MQ 7.5 的产品版本升级的队列管理器配合使用，请确保 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 将 SHARE/NOSHARE 选项设置为 **SHARE**。

2. 决定首先引用哪个完整存储库 PARIS 。

   集群中的每个队列管理器都必须引用一个或多个完整存储库。 它从完整存储库收集有关集群的信息，因此构建自己的部分存储库。 选择任一存储库作为完整存储库。 一旦将新的队列管理器添加到集群，它就会立即了解其他存储库。 有关对队列管理器的更改的信息将直接发送到两个存储库。 在此示例中，仅出于地理原因，您将 PARIS 链接到队列管理器 LONDON。

   注: 在启动队列管理器 PARIS 后，按任何顺序执行其余步骤。

3. 在队列管理器 PARIS 上定义 CLUSRCVR 通道。

   集群中的每个队列管理器都必须定义集群接收方通道，在该通道上可以接收消息。 在 PARIS 上，定义:

   ```
   DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for queue manager PARIS')
   ```

   集群接收方通道会公布队列管理器的可用性，以便从集群 INVENTORY 中的其他队列管理器接收消息。 请勿在其他队列管理器上为集群接收方通道 INVENTORY.PARIS 的发送端创建定义。 需要时，将自动生成其他定义。 请参阅 集群通道 (Cluster channel)。

4. 在队列管理器 PARIS 上定义 CLUSSDR 通道。

   向集群添加不是完整存储库的队列管理器时，仅定义一个集群发送方通道以建立与完整存储库的初始连接。 请参阅 集群发送方通道: CLUSSDR 。

   在 PARIS 上，将名为 INVENTORY.LONDON 的 CLUSSDR 通道定义为具有网络地址 LONDON.CHSTORE.COM 的队列管理器。

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from PARIS to repository at LONDON')
   ```

   队列管理器自动从模型队列 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 创建永久动态集群传输队列 SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON 。 它将传输队列的 CLCHNAME 属性设置为 INVENTORY.LONDON。

结果

下图显示了此任务设置的集群。

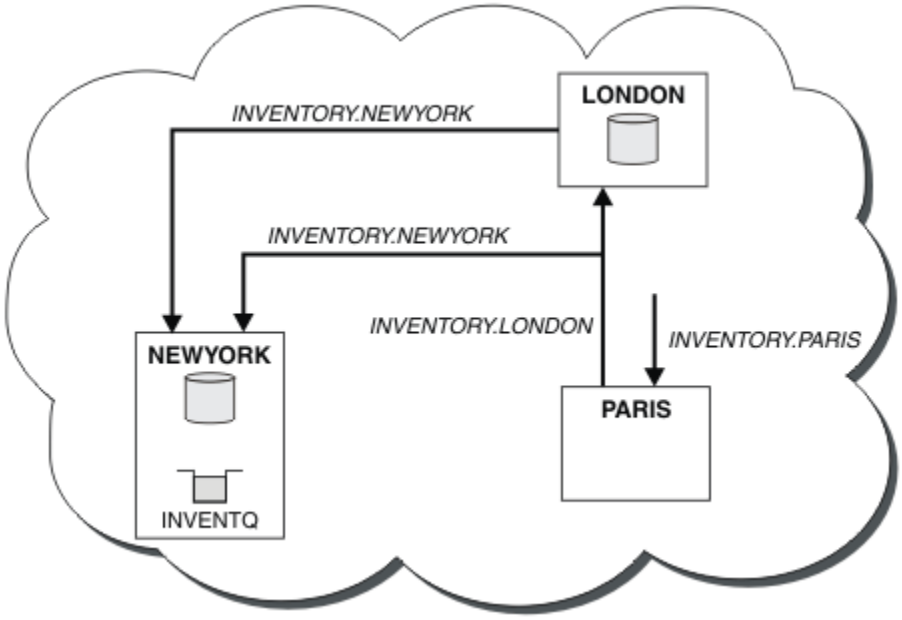


图 *42:* 具有三个队列管理器的 *INVENTORY* 集群

通过仅创建两个定义 (CLUSRCVR 定义和 CLUSSDR 定义) ，我们将队列管理器 PARIS 添加到集群中。

现在， PARIS 队列管理器从位于 LONDON 的完整存储库中了解到 INVENTQ 队列由队列管理器 NEWYORK 托管。 当巴黎系统托管的应用程序尝试将消息放入 INVENTQ 时， PARIS 会自动定义集群发送方通道以连接到集群接收方通道 INVENTORY.NEWYORK。 当将其队列管理器名称指定为目标队列管理器并提供应答队列时，应用程序可以接收响应。

相关概念

如何选择要使用的集群传输队列类型

相关任务

使用 DHCP 将队列管理器添加到集群

使用 DHCP 将队列管理器添加到集群。 该任务演示了在 CLUSRCVR 定义上省略 CONNAME 值。

使用 *DHCP* 将队列管理器添加到集群

使用 DHCP 将队列管理器添加到集群。 该任务演示了在 CLUSRCVR 定义上省略 CONNAME 值。

开始之前

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

该任务演示了两个特殊功能:

- 在 CLUSRCVR 定义中省略 CONNAME 值的能力。
- 在 CLUSSDR 定义上使用 +QMNAME+ 的能力。

z/OS 上未提供任何功能部件。

场景:

- 已按 第 244 页的『设置新集群』中所述设置 INVENTORY 集群。 它包含两个队列管理器 LONDON 和 NEWYORK，这两个队列管理器都保存完整存储库。
- 正在巴黎设置链式商店的新分支，您希望将名为 PARIS 的队列管理器添加到集群中。
- 队列管理器 PARIS 通过将消息放在 INVENTQ 队列上，将库存更新发送到在纽约系统上运行的应用程序。
- 所有三个系统之间都存在网络连接。
- 网络协议是 TCP。

- PARIS 队列管理器系统使用 DHCP，这意味着在系统重新启动时 IP 地址可能会更改。
- PARIS 和 LONDON 系统之间的通道根据定义的命名约定进行命名。 此约定使用 LONDON 上完整存储库队列管理器的队列管理器名称。
- PARIS 队列管理器的管理员没有有关 LONDON 存储库上队列管理器的名称的信息。 LONDON 存储库上的队列管理器的名称可能会更改。

关于此任务

执行以下步骤以使用 DHCP 将队列管理器添加到集群。

过程

1. 决定首先引用哪个完整存储库 PARIS 。

   集群中的每个队列管理器都必须引用一个或多个完整存储库。 它从完整存储库收集有关集群的信息，因此构建自己的部分存储库。 选择任一存储库作为完整存储库。 一旦将新的队列管理器添加到集群，它就会立即了解其他存储库。 有关对队列管理器的更改的信息将直接发送到两个存储库。 在此示例中，我们选择纯粹出于地理原因将 PARIS 链接到队列管理器 LONDON。

   注: 在启动队列管理器 PARIS 后，按任何顺序执行其余步骤。

2. 在队列管理器 PARIS 上定义 CLUSRCVR 通道。

   集群中的每个队列管理器都需要定义可接收消息的集群接收方通道。 在 PARIS 上，定义:

   ```
   DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR)
   TRPTYPE(TCP) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for queue manager PARIS')
   ```

   集群接收方通道会公布队列管理器的可用性，以便从集群 INVENTORY 中的其他队列管理器接收消息。 您无需在集群接收方通道上指定 CONNAME 。 您可以通过省略 CONNAME 或指定 CONNAME(' ')来请求 IBM MQ 从系统中查找连接名称。 IBM MQ 使用系统的当前 IP 地址生成 CONNAME 值; 请参阅 CONNAME 。 对于集群接收方通道 INVENTORY.PARIS 的发送端，不需要在其他队列管理器上创建定义。 需要时，将自动生成其他定义。

3. 在队列管理器 PARIS 上定义 CLUSSDR 通道。

   集群中的每个队列管理器都需要定义一个集群发送方通道，在该通道上可以将消息发送到其初始完整存储库。 在 PARIS 上，对网络地址为 LONDON.CHSTORE.COM 的队列管理器的名为 INVENTORY.+QMNAME+ 的通道进行以下定义。

   ```
   DEFINE CHANNEL(INVENTORY.+QMNAME+) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from PARIS to repository at LONDON')
   ```

4. 可选： 如果要将先前已从同一集群中除去的队列管理器添加到集群，请检查该队列管理器现在是否显示为集群成员。 否则，请完成以下额外步骤:

   a) 在要添加的队列管理器上发出 **REFRESH CLUSTER** 命令。

   此步骤将停止集群通道，并为本地集群高速缓存提供一组新的序列号，确保这些序列号在集群的其余部分中是最新的。

   ```
   REFRESH CLUSTER(INVENTORY) REPOS(YES)
   ```

   注: 对于大型集群，使用 **REFRESH CLUSTER** 命令可能会对正在运行的集群造成干扰，此外每隔 27 天在集群对象向所有相关队列管理器自动发送状态更新时也可能有干扰。 请参阅在大型集群中刷新可能会影响集群的性能和可用性。

   b) 重新启动 CLUSSDR 通道

   (例如，使用 START CHANNEL 命令)。

   c) 重新启动 CLUSRCVR 通道。

结果

此任务设置的集群与 第 254 页的『将队列管理器添加至集群』相同:

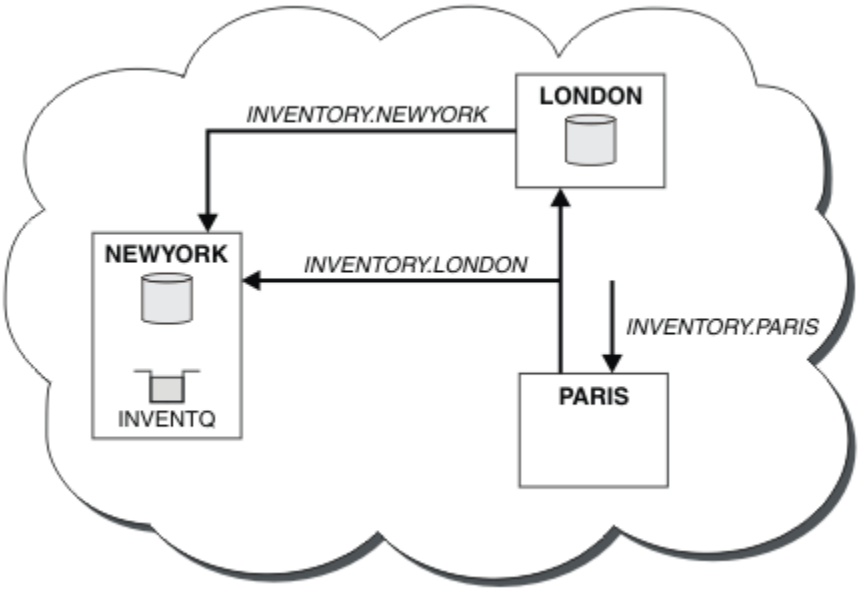


图 *43:* 具有三个队列管理器的 *INVENTORY* 集群

通过仅创建两个定义 (CLUSRCVR 定义和 CLUSSDR 定义) ，我们将队列管理器 PARIS 添加到集群中。

在 PARIS 队列管理器上，将启动包含字符串 +QMNAME+ 的 CLUSSDR 。 在 LONDON 系统 IBM MQ 上，将 +QMNAME+ 解析为队列管理器名称 (LONDON)。 然后， IBM MQ 将名为 INVENTORY.LONDON 的通道的定义与相应的 CLUSRCVR 定义相匹配。

IBM MQ 将已解析的通道名称发送回 PARIS 队列管理器。 在 PARIS 上，名为 INVENTORY.+QMNAME+ 的通道的 CLUSSDR 通道定义将替换为 INVENTORY.LONDON 的内部生成的 CLUSSDR 定义。 此定义包含已解析的通道名称，但其他定义与您创建的 +QMNAME+ 定义相同。 还会使用新解析的通道名称来更新集群存储库的通道定义。

注:

1. 使用 +QMNAME+ 名称创建的通道将立即变为不活动状态。 它从不用于传输数据。
2. 通道出口可能会看到一次调用与下一次调用之间的通道名称更改。

现在， PARIS 队列管理器从位于 LONDON 的存储库中了解到 INVENTQ 队列由队列管理器 NEWYORK 托管。 当巴黎系统托管的应用程序尝试将消息放入 INVENTQ, PARIS 时，会自动定义集群发送方通道以连接到集群接收方通道 INVENTORY.NEWYORK。 当将其队列管理器名称指定为目标队列管理器并提供应答队列时，应用程序可以接收响应。

相关任务

将队列管理器添加到集群: 单独的传输队列
遵循以下指示信息将队列管理器添加到您创建的集群。 使用多个集群传输队列将消息传输到集群队列和主题。

相关参考

DEFINE CHANNEL

添加托管队列的队列管理器

将另一个队列管理器添加到集群，以托管另一个 INVENTQ 队列。 请求交替发送到每个队列管理器上的队列。 无需对现有 INVENTQ 主机进行任何更改。

开始之前

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

场景：

- 已按 第 254 页的『将队列管理器添加至集群』中所述设置 INVENTORY 集群。 它包含三个队列管理器; LONDON 和 NEWYORK 都保存完整存储库，而巴黎保存部分存储库。 库存应用程序在纽约的系统上运行，并连接到 NEWYORK 队列管理器。 应用程序由 INVENTQ 队列上的消息到达驱动。
- 多伦多正在设立一家新商店。 要提供额外的容量，您希望在多伦多和纽约的系统上运行库存应用程序。
- 所有四个系统之间都存在网络连接。
- 网络协议是 TCP。

注: 队列管理器 TORONTO 仅包含部分存储库。 如果要将完整存储库队列管理器添加到集群，请参阅 第 264 页的『将完整存储库移至另一个队列管理器』。

关于此任务

执行以下步骤以添加用于托管队列的队列管理器。

过程

1. 决定首先引用哪个完整存储库 TORONTO 。

   集群中的每个队列管理器都必须引用一个或多个完整存储库。 它从完整存储库收集有关集群的信息，因此构建自己的部分存储库。 您选择的存储库没有特别重要的意义。 在此示例中，我们选择 NEWYORK。 一旦新的队列管理器加入集群，它就会与这两个存储库进行通信。

2. 定义 CLUSRCVR 通道。

   集群中的每个队列管理器都需要定义可接收消息的集群接收方通道。 在 TORONTO 上，定义 CLUSRCVR 通道:

   ```
   DEFINE CHANNEL(INVENTORY.TORONTO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(TORONTO.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for TORONTO')
   ```

   TORONTO 队列管理器通过其集群接收方通道来发布其可用性，以从 INVENTORY 集群中的其他队列管理器接收消息。

3. 在队列管理器 TORONTO 上定义 CLUSSDR 通道。

   集群中的每个队列管理器都需要定义一个集群发送方通道，在该通道上可以将消息发送到其第一个完整存储库。 在这种情况下，请选择 NEWYORK。 TORONTO 需要以下定义:

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from TORONTO to repository at NEWYORK')
   ```

4. 可选： 如果要将先前已从同一集群中除去的队列管理器添加到集群，请检查该队列管理器现在是否显示为集群成员。 否则，请完成以下额外步骤:

   a) 在要添加的队列管理器上发出 **REFRESH CLUSTER** 命令。

   此步骤将停止集群通道，并为本地集群高速缓存提供一组新的序列号，确保这些序列号在集群的其余部分中是最新的。

   ```
   REFRESH CLUSTER(INVENTORY) REPOS(YES)
   ```

   注: 对于大型集群，使用 **REFRESH CLUSTER** 命令可能会对正在运行的集群造成干扰，此外每隔 27 天在集群对象向所有相关队列管理器自动发送状态更新时也可能有干扰。 请参阅在大型集群中刷新可能会影响集群的性能和可用性。

   b) 重新启动 CLUSSDR 通道

   (例如，使用 START CHANNEL 命令)。

c) 重新启动 CLUSRCVR 通道。

5. 查看库存应用程序以获取消息亲缘关系。

   在继续之前，请确保清单应用程序与消息处理顺序没有任何依赖关系，并在多伦多的系统上安装该应用程序。

6. 定义集群队列 INVENTQ。

   INVENTQ 队列 (已由 NEWYORK 队列管理器主管) 也将由 TORONTO 主管。 在 TORONTO 队列管理器上定义它，如下所示:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

结果

第 262 页的图 44 显示了此任务设置的 INVENTORY 集群。

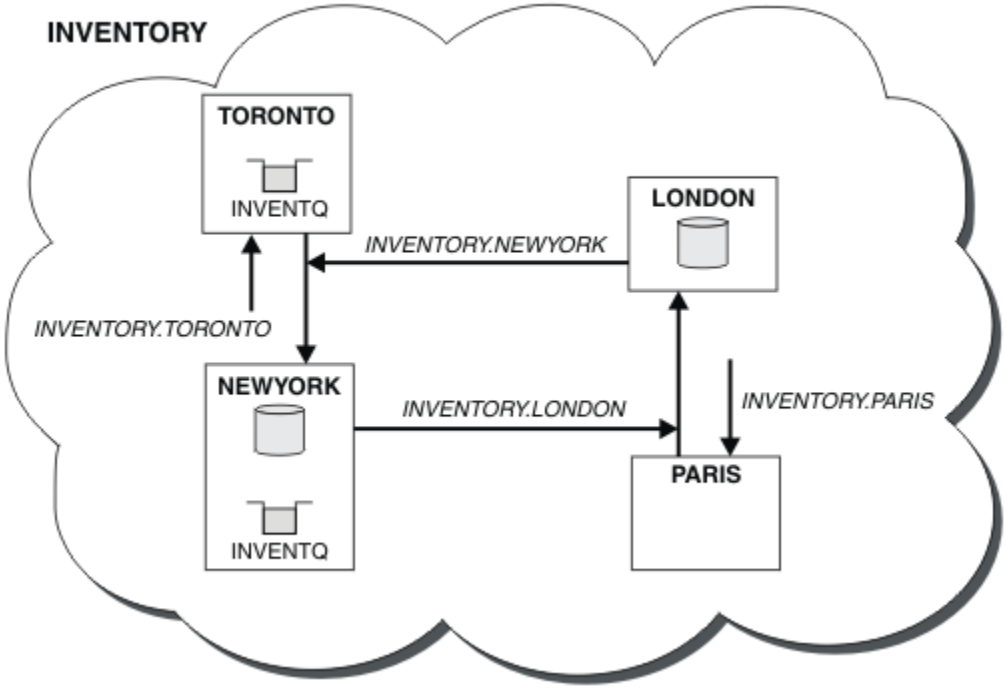


图 *44:* 具有四个队列管理器的 *INVENTORY* 集群

INVENTQ 队列和库存应用程序现在托管在集群中的两个队列管理器上。 这将提高其可用性，加快消息吞吐量，并允许在两个队列管理器之间分配工作负载。 TORONTO 或 NEWYORK 放入 INVENTQ 的消息将尽可能由本地队列管理器上的实例处理。 LONDON 或 PARIS 放入的消息交替路由到 TORONTO 或 NEWYORK，以便均衡工作负载。

对集群的此修改已完成，而不必更改队列管理器 NEWYORK， LONDON 和 PARIS 上的定义。 这些队列管理器中的完整存储库将自动更新为能够将消息发送到 INVENTQ (TORONTO) 所需的信息。 如果其中一个 NEWYORK 或 TORONTO 队列管理器变为不可用，并且具有足够的容量，那么库存应用程序将继续工作。 如果库存应用程序同时在两个位置进行托管，那么该应用程序必须能够正常工作。

从此任务的结果中可以看到，您可以在多个队列管理器上运行相同的应用程序。 您可以集群到均匀分布的工作负载。

应用程序可能无法处理两个位置中的记录。 例如，假设您决定添加客户帐户查询并更新在 LONDON 和 NEWYORK 中运行的应用程序。 一个帐户记录只能保存在一个位置。 您可以决定使用数据分区技术来控制请求的分布。 您可以拆分记录的分布。 您可以安排将一半记录 (例如帐号 00000-49999) 保存在 LONDON 中。 另一半 (范围为 50000-99999) 保存在 NEWYORK 中。 然后，您可以编写集群工作负载出口程序以检查所有消息中的帐户字段，并将消息路由到相应的队列管理器。

下一步做什么

现在，您已完成所有定义，如果尚未这样做，请在 IBM MQ for z/OS 上启动通道启动程序。 在所有平台上，在队列管理器 TORONTO 上启动侦听器程序。 侦听器程序等待入局网络请求，并在需要时启动集群接收方通道。

z/OS 向现有集群添加队列共享组

将 z/OS 上的队列共享组添加到现有集群。

开始之前

注:

1. 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。
2. 仅在 IBM MQ for z/OS 上支持队列共享组。 此任务不适用于其他平台。

场景：

- 已按 第 244 页的『设置新集群』中所述设置 INVENTORY 集群。 它包含两个队列管理器 LONDON 和 NEWYORK。
- 您要将队列共享组添加到此集群。 组 QSGP 由三个队列管理器 P1， P2 和 P3 组成。 它们共享将由 P1 定义的 INVENTQ 队列的实例。

关于此任务

执行以下步骤以添加托管共享队列的新队列管理器。

过程

1. 确定队列管理器首先引用的完整存储库。

   集群中的每个队列管理器都必须引用一个或多个完整存储库。 它从完整存储库收集有关集群的信息，因此构建自己的部分存储库。 您选择的完整存储库没有特别重要的意义。 在此示例中，选择 NEWYORK。 一旦队列共享组加入集群，它就会与两个完整存储库进行通信。

2. 定义 CLUSRCVR 通道。

   集群中的每个队列管理器都需要定义可接收消息的集群接收方通道。 在 P1， P2 和 P3 上，定义:

   ```
   DEFINE CHANNEL(INVENTORY.Pn) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(Pn.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for sharing queue manager')
   ```

   集群接收方通道会公布每个队列管理器的可用性，以从集群 INVENTORY 中的其他队列管理器接收消息。

3. 为队列共享组定义 CLUSSDR 通道。

   集群的每个成员都需要定义一个集群发送方通道，在该通道上可以将消息发送到其第一个完整存储库。 在这种情况下，我们选择了 NEWYORK。 队列共享组中的其中一个队列管理器需要以下组定义。 该定义确保每个队列管理器都具有集群发送方通道定义。

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) QSGDISP(GROUP)
   DESCR('Cluster-sender channel to repository at NEWYORK')
   ```

4. 定义共享队列。

   在 P1 上定义队列 INVENTQ，如下所示:

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) QSGDISP(SHARED) CFSTRUCT(STRUCTURE)
   ```

   在新的队列管理器上启动通道启动程序和侦听器程序。 侦听器程序侦听入局网络请求，并在需要时启动集群接收方通道。

结果

第 264 页的图 45 显示此任务设置的集群。

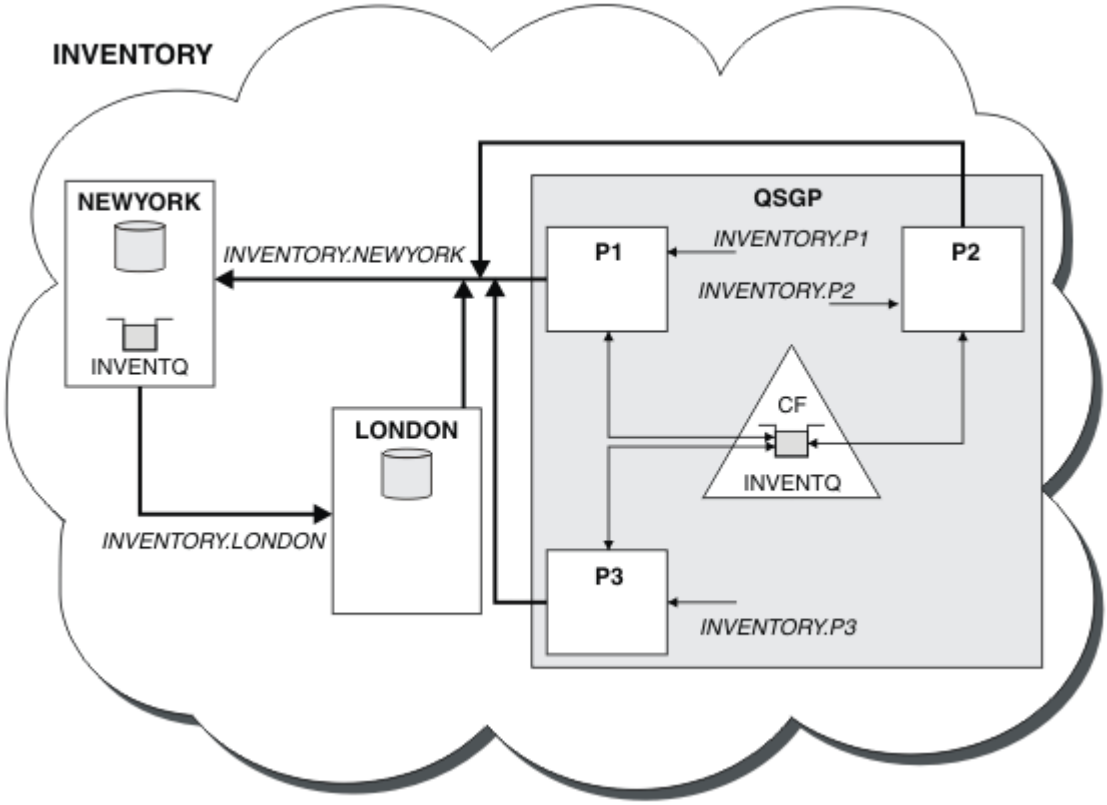


图 *45:* 集群和队列共享组

现在， LONDON 放置在 INVENTQ 队列上的消息将在作为主管队列的四个队列管理器周围交替路由。

下一步做什么

让队列共享组的成员托管集群队列的好处是该组的任何成员都可以应答请求。 在这种情况下，在共享队列上接收消息后， P1 可能变为不可用。 队列共享组的另一成员可以改为应答。

将完整存储库移至另一个队列管理器

将完整存储库从一个队列管理器移至另一个队列管理器，从第二个存储库中保存的信息构建新存储库。

开始之前

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

场景：

- 已按 第 254 页的『将队列管理器添加至集群』中所述设置 INVENTORY 集群。
- 出于业务原因，您现在希望从队列管理器 LONDON 中除去完整存储库，并将其替换为队列管理器 PARIS 中的完整存储库。 NEWYORK 队列管理器将继续保存完整存储库。

关于此任务

执行以下步骤以将完整存储库移至另一个队列管理器。

过程

1. 变更 PARIS 以使其成为完整的存储库队列管理器。

   在 PARIS 上，发出以下命令:

   ```
   ALTER QMGR REPOS(INVENTORY)
   ```

2. 在 PARIS 上添加 CLUSSDR 通道

   PARIS 当前具有指向 LONDON 的集群发送方通道。 LONDON 不再包含集群的完整存储库。 PARIS 必须具有指向 NEWYORK 的新集群发送方通道，其他完整存储库现在保存在该通道中。

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from PARIS to repository at NEWYORK')
   ```

3. 在 NEWYORK 上定义指向 PARIS 的 CLUSSDR 通道

   当前 NEWYORK 具有指向 LONDON 的集群发送方通道。 现在，另一个完整存储库已移至 PARIS，您需要在 NEWYORK 处添加指向 PARIS 的新集群发送方通道。

   ```
   DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from NEWYORK to repository at PARIS')
   ```

   将集群发送方通道添加到 PARIS 时， PARIS 将从 NEWYORK 学习有关集群的信息。 它使用 NEWYORK 中的信息构建自己的完整存储库。

4. 检查队列管理器 PARIS 现在是否具有完整存储库

   检查队列管理器 PARIS 是否已从队列管理器 NEWYORK 上的完整存储库构建了自己的完整存储库。 发出下列命令:

   ```
   DIS QCLUSTER(*) CLUSTER (INVENTORY)
   DIS CLUSQMGR(*) CLUSTER (INVENTORY)
   ```

   检查这些命令是否显示此集群中与 NEWYORK 相同的资源的详细信息。

   注: 如果队列管理器 NEWYORK 不可用，那么此信息构建无法完成。 在任务完成之前，请勿继续执行下一步。

5. 更改 LONDON 上的队列管理器定义

   最后，更改位于 LONDON 的队列管理器，以使其不再包含集群的完整存储库。 在 LONDON 上，发出以下命令:

   ```
   ALTER QMGR REPOS(' ')
   ```

   队列管理器不再接收任何集群信息。 在 30 天后，存储在其完整存储库中的信息将到期。 现在，队列管理器 LONDON 将构建其自己的部分存储库。

6. 除去或更改任何未完成的定义。

   当您确定集群的新安排按预期工作时，请除去或更改不再正确的手动定义的 CLUSSDR 定义。

   - 在 PARIS 队列管理器上，必须停止并删除到 LONDON 的集群发送方通道，然后发出启动通道命令，以便集群可以再次使用自动通道:

     ```
     STOP CHANNEL(INVENTORY.LONDON)
     DELETE CHANNEL(INVENTORY.LONDON)
     START CHANNEL(INVENTORY.LONDON)
     ```

   - 在 NEWYORK 队列管理器上，必须停止并删除到 LONDON 的集群发送方通道，然后发出启动通道命令，以便集群可以再次使用自动通道:

     ```
     STOP CHANNEL(INVENTORY.LONDON)
     DELETE CHANNEL(INVENTORY.LONDON)
     START CHANNEL(INVENTORY.LONDON)
     ```

- 将集群中所有队列管理器上指向 LONDON 的所有其他手动定义的集群发送方通道替换为指向 NEWYORK 或 PARIS 的通道。 删除通道后，请始终发出 **start channel** 命令，以便集群可以再次使用自动通道。 在这个小小的例子中，没有其他。 要检查是否存在任何其他您遗忘的队列管理器，请从每个队列管理器发出 DISPLAY CHANNEL 命令，并指定 TYPE(CLUSSDR)。 例如：

```
DISPLAY CHANNEL(*) TYPE(CLUSSDR)
```

请务必在将完整存储库从 LONDON 移动到 PARIS 之后尽快执行此任务。 在执行此任务之前的时间内，已手动定义名为 INVENTORY.LONDON 的 CLUSSDR 通道的队列管理器可能会使用此通道发送信息请求。

在 LONDON 不再是完整存储库之后，如果它接收到此类请求，那么它会将错误消息写入其队列管理器错误日志。 以下示例显示在 LONDON 上可能会看到哪些错误消息：

- AMQ9428: Unexpected publication of a cluster queue object received
- AMQ9432: Query received by a non-repository queue manager

队列管理器 LONDON 不会响应信息请求，因为它不再是完整存储库。 请求来自 LONDON 的信息的队列管理器必须依赖于 NEWYORK 以获取集群信息，直到将其手动定义的 CLUSSDR 定义更正为指向 PARIS 为止。 长期而言，不能允许将此情境作为有效配置。

结果

第 266 页的图 46 显示此任务设置的集群。

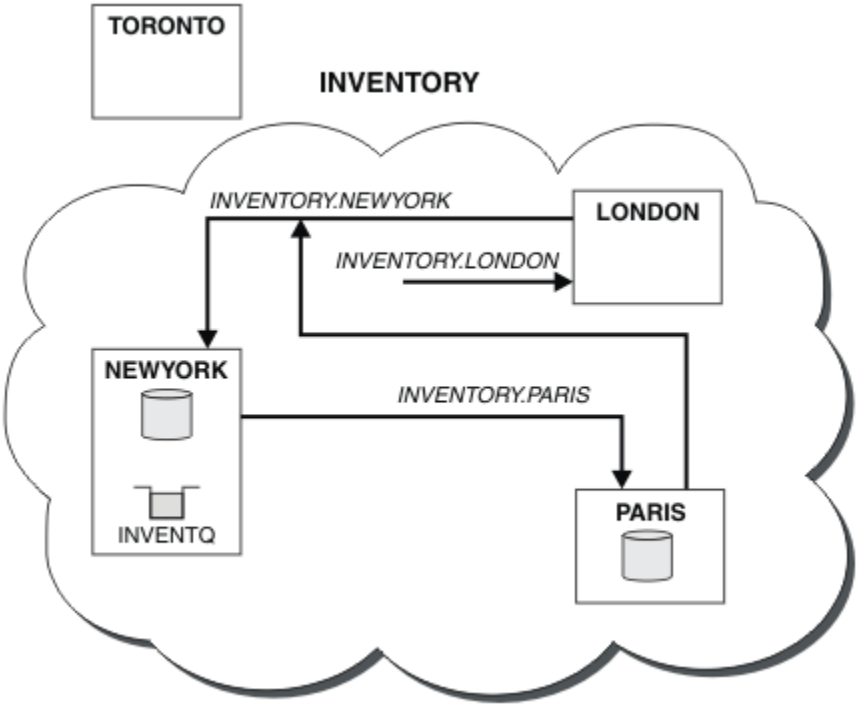


图 *46:* 将完整存储库移至 *PARIS* 的 *INVENTORY* 集群

在集群中建立通信

当有要传递的消息时，需要通道启动程序来启动通信通道。 通道侦听器等待启动通道的另一端以接收消息。

开始之前

要在集群中的队列管理器之间建立通信，请使用其中一个受支持的通信协议来配置链路。 受支持的协议包括：

- 任何平台上的 TCP 或 LU 6.2

- Windows Windows 系统上的 NetBIOS 或 SPX

作为此配置的一部分，您还需要通道启动程序和通道侦听器，就像使用分布式排队一样。

关于此任务

所有集群队列管理器都需要一个通道启动程序来监视系统定义的启动队列 SYSTEM.CHANNEL.INITQ。SYSTEM.CHANNEL.INITQ 是所有传输队列 (包括集群传输队列) 的启动队列。

每个队列管理器都必须具有通道侦听器。 通道侦听器程序等待入局网络请求，并在需要时启动相应的接收方通道。 频道听众的实现是特定于平台的，但是有一些共同的特点。

在所有 IBM MQ 平台上，都可以使用 **START LISTENER** 命令启动侦听器。

Multi 在 Multiplatforms 版上，您可以与队列管理器同时自动启动侦听器。 要自动启动侦听器，请将 LISTENER 对象的 CONTROL 属性设置为 QMGR 或 STARTONLY。

z/OS 非共享侦听器端口 (INDISP (QMGR)) 必须用于 z/OS 上的 CLUSRCVR 通道以及到 z/OS 的 CLUSSDR 通道。

过程

1. 启动通道启动程序。
   - z/OS 在 z/OS 上，每个队列管理器都有一个通道启动程序，它作为单独的地址空间运行。 您可以使用在队列管理器启动过程中发出的 **MQSC** START CHINIT 命令将其启动。
   - ALW 在 AIX, Linux, and Windows 上，启动队列管理器时，如果队列管理器属性 SCHINIT 设置为 QMGR，那么将自动启动通道启动程序。 否则，可以使用 **runmqsc** START CHINIT 命令或 **runmqchi** 控制命令来启动该命令。
   - IBM i 在 IBM i 上，启动队列管理器时，如果队列管理器属性 SCHINIT 设置为 QMGR，那么将自动启动通道启动程序。 否则，可以使用 **runmqsc** START CHINIT 命令或 **runmqchi** 控制命令来启动该命令。
2. 启动通道侦听器。
   - z/OS 在 z/OS 上，使用 IBM MQ 提供的通道侦听器程序。 要启动 IBM MQ 通道侦听器，请使用 **MQSC** 命令 START LISTENER，您在通道启动程序启动过程中发出该命令。 例如：

     ```
     START LISTENER PORT(1414) TRPTYPE(TCP)
     ```

     或：

     ```
     START LISTENER LUNAME(LONDON.LUNAME) TRPTYPE(LU62)
     ```

     队列共享组的成员可以对每个队列管理器使用共享侦听器而不是侦听器。 请勿将共享侦听器与集群配合使用。 具体而言，请勿使 CLUSRCVR 通道的 CONNAME 成为队列共享组的共享侦听器的地址。 如果您这样做，那么队列管理器可能会接收到它们没有定义的队列的消息。
   - IBM i 在 IBM i 上，使用 IBM MQ 提供的通道侦听器程序。 要启动 IBM MQ 通道侦听器，请使用 **CL** 命令 STRMQMLSR。 例如：

     ```
     STRMQMLSR MQMNAME(QM1) PORT(1414)
     ```
   - Windows 在 Windows 上，使用 IBM MQ 提供的通道侦听器程序或操作系统提供的工具。

     要启动 IBM MQ 通道侦听器，请使用 RUNMQLSR 命令。 例如：

     ```
     RUNMQLSR -t tcp -p 1414 -m QM1
     ```

- Linux AIX 在 AIX and Linux 上，使用 IBM MQ 提供的通道侦听器程序或操作系统提供的工具; 例如， **inetd** 用于 TCP 通信。

  要启动 IBM MQ 通道侦听器，请使用 **runmqlsr** 命令。 例如：

  ```
  runmqlsr -t tcp -p 1414 -m QM1
  ```

  要使用 **inetd** 来启动通道，请配置两个文件：

  a. 编辑文件 /etc/services。 您必须以超级用户或 root 用户身份登录。 如果以下行不在文件中，请按如下所示进行添加：

  ```
  MQSeries    1414/tcp   # WebSphere MQ channel listener
  ```

  其中 1414 是 IBM MQ 所需的端口号。 您可以更改端口号，但它必须与发送端指定的端口号匹配。

  b. 编辑文件 /etc/inetd.conf。 如果该文件中没有以下行，请按如下所示添加该文件：

  ```
  MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta
  -m queue.manager.name
  ```

  其中， *MQ_INSTALLATION_PATH* 将替换为安装了 IBM MQ 的高级目录。

  在 **inetd** 重新读取配置文件后，更新将变为活动状态。 从 root 用户标识发出以下命令：

  AIX 在 AIX 上：

  ```
  refresh -s inetd
  ```

  Linux 在 Linux 上：

  a. 使用以下命令查找 **inetd** 的进程标识：

  ```
  ps -ef | grep inetd
  ```

  b. 运行相应的命令。

  对于 Linux:

  ```
  kill -1 inetd processid
  ```

将现有网络转换为集群

将已存在的分布式排队网络转换为集群，并添加额外的队列管理器以增加容量。

开始之前

在 第 244 页的『设置新集群』 到 第 264 页的『将完整存储库移至另一个队列管理器』 中，您创建并扩展了新集群。 接下来的两项任务将探索另一种方法: 将现有队列管理器网络转换为集群。

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

场景：

- 已经建立了 IBM MQ 网络，连接了一家连锁店的全国分支机构。 它具有中心和辐射结构: 所有队列管理器都连接到一个中央队列管理器。 中央队列管理器位于运行库存应用程序的系统上。 应用程序由 INVENTQ 队列上的消息到达来驱动，对于这些消息，每个队列管理器都有一个远程队列定义。

  此网络在 第 269 页的图 47 中进行了说明。

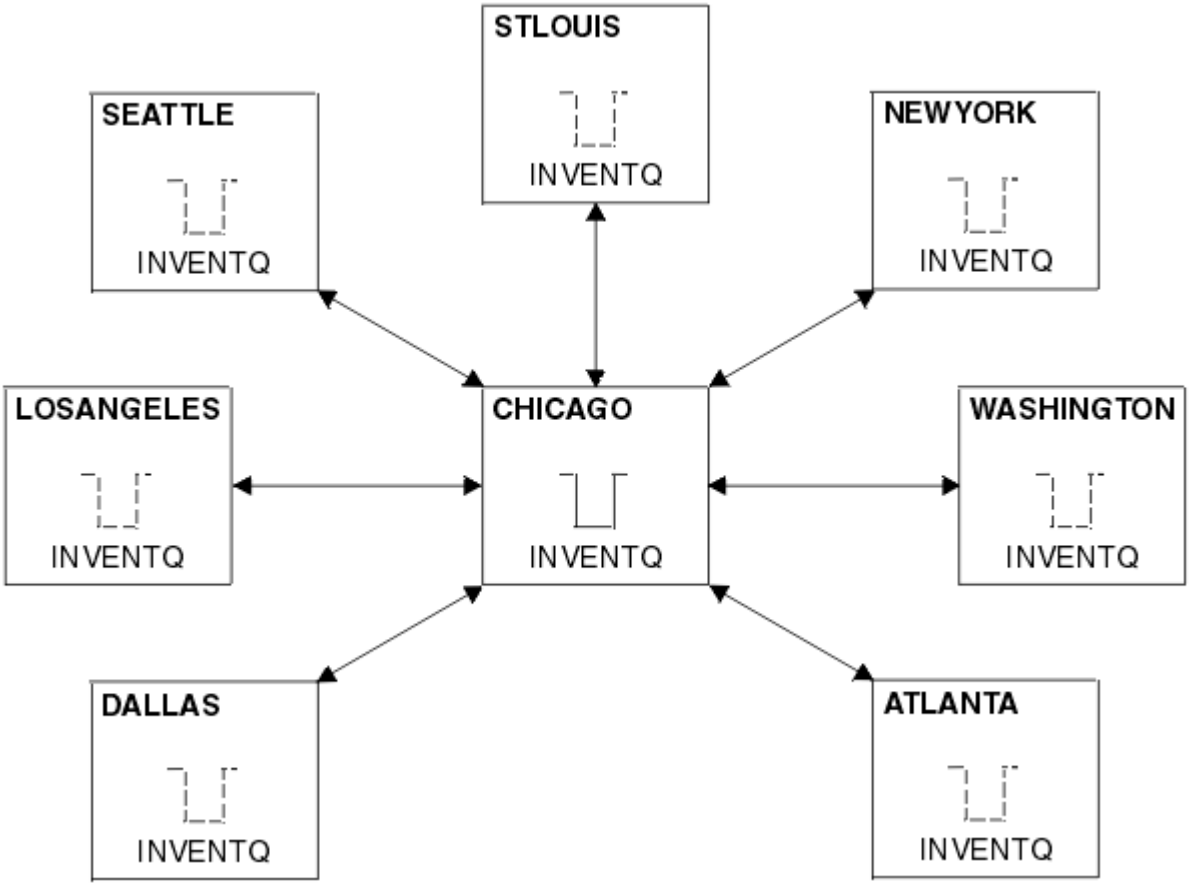


图 *47:* 集线器和辐射网络

- 为了简化管理，您要将此网络转换为集群，并在中央站点上创建另一个队列管理器以共享工作负载。

  集群名称为 CHNSTORE。

  注: 选择了集群名称 CHNSTORE 以允许使用不超过最大长度 20 个字符的格式 *cluster_name.queue_manager_name* 的名称创建集群接收方通道名称，例如 CHNSTORE.WASHINGTON。
- 这两个中央队列管理器都将托管完整存储库，并且可供库存应用程序访问。
- 库存应用程序将由由任一中央队列管理器托管的 INVENTQ 队列上的消息到达来驱动。
- 库存应用程序将是唯一并行运行且可由多个队列管理器访问的应用程序。 所有其他应用程序继续像以前一样运行。
- 所有分支都具有与两个中央队列管理器的网络连接。
- 网络协议是 TCP。

关于此任务

执行以下步骤以将现有网络转换为集群。

过程

1. 查看库存应用程序以获取消息亲缘关系。

   在继续之前，请确保应用程序可以处理消息亲缘关系。 消息亲缘关系是在两个应用程序之间交换的会话式消息之间的关系，其中消息必须由特定队列管理器处理或按特定顺序处理。 有关消息亲缘关系的更多信息，请参阅: 第 333 页的『处理消息亲缘关系』
2. 更改两个中央队列管理器，使它们成为完整的存储库队列管理器。

两个队列管理器 CHICAGO 和 CHICAGO2 位于此网络的中心。 您已决定将与链商店集群关联的所有活动集中到这两个队列管理器上。 除了 INVENTQ 队列的库存应用程序和定义外，您还希望这些队列管理器托管集群的两个完整存储库。 在两个队列管理器中的每个队列管理器上，发出以下命令:

```
ALTER QMGR REPOS(CHNSTORE)
```

3. 在每个队列管理器上定义 CLUSRCVR 通道。

   在集群中的每个队列管理器上，定义集群接收方通道和集群发送方通道。 首先定义哪个通道无关紧要。

   创建 CLUSRCVR 定义以将每个队列管理器，其网络地址和其他信息宣传到集群。 例如，在队列管理器 ATLANTA 上:

   ```
   DEFINE CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(ATLANTA.CHSTORE.COM) CLUSTER(CHNSTORE)
   DESCR('Cluster-receiver channel')
   ```

4. 在每个队列管理器上定义 CLUSSDR 通道

   在每个队列管理器上创建 CLUSSDR 定义，以将该队列管理器链接到一个或多个完整存储库队列管理器。 例如，您可以将 ATLANTA 链接到 CHICAGO2:

   ```
   DEFINE CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(CHICAGO2.CHSTORE.COM) CLUSTER(CHNSTORE)
   DESCR('Cluster-sender channel to repository queue manager')
   ```

5. 在 CHICAGO2 上安装库存应用程序。

   您已在队列管理器 CHICAGO 上具有库存应用程序。 现在，您需要在队列管理器 CHICAGO2 上生成此应用程序的副本。

6. 在中央队列管理器上定义 INVENTQ 队列。

   在 CHICAGO 上，修改队列 INVENTQ 的本地队列定义，以使该队列可供集群使用。 发出以下命令:

   ```
   ALTER QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
   ```

   在 CHICAGO2 上，对同一队列进行定义:

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
   ```

   在 z/OS 上，可以使用 **CSQUTIL** 的 COMMAND 函数的 MAKEDEF 选项在 CHICAGO 上的 INVENTQ 的 CHICAGO2 上生成精确副本。

   当您创建这些定义时，将向位于 CHICAGO 和 CHICAGO2 的完整存储库发送一条消息，并更新其中的信息。 队列管理器在将消息放入 INVENTQ 时从完整存储库中发现消息的目标可供选择。

7. 检查是否已传播集群更改。

   检查您在上一步中创建的定义是否已通过集群传播。 在完整存储库队列管理器上发出以下命令:

   ```
   DIS QCLUSTER(INVENTQ)
   ```

添加新的互连集群

添加与现有集群共享某些队列管理器的新集群。

开始之前

注:

1. 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。
2. 在启动此任务之前，请检查队列名称冲突并了解后果。 您可能需要重命名队列或设置队列别名，然后才能继续。

场景:

- 已按 第 268 页的『将现有网络转换为集群』中所述设置 IBM MQ 集群。
- 将实现名为 MAILORDER 的新集群。 此集群由 CHNSTORE 集群中的四个队列管理器， CHICAGO，CHICAGO2， SEATTLE 和 ATLANTA 以及另外两个队列管理器 HARTFORD 和 OMAHA 组成。 MAILORDER 应用程序在连接到队列管理器 OMAHA 的 Omaha 系统上运行。 它由集群中的其他队列管理器驱动，这些队列管理器将消息放在 MORDERQ 队列上。
- MAILORDER 集群的完整存储库在两个队列管理器 CHICAGO 和 CHICAGO2 上进行维护。
- 网络协议是 TCP。

关于此任务

执行以下步骤以添加新的互连集群。

过程

1. 创建集群名称的名称列表。

   CHICAGO 和 CHICAGO2 上的完整存储库队列管理器现在将保存集群 CHNSTORE 和 MAILORDER 的完整存储库。 首先，创建包含集群名称的名称列表。 在 CHICAGO 和 CHICAGO2 上定义名称列表，如下所示:

   ```
   DEFINE NAMELIST(CHAINMAIL)
   DESCR('List of cluster names')
   NAMES(CHNSTORE, MAILORDER)
   ```

2. 更改两个队列管理器定义。

   现在更改位于 CHICAGO 和 CHICAGO2 的两个队列管理器定义。 目前，这些定义显示队列管理器保存集群 CHNSTORE 的完整存储库。 更改该定义以显示队列管理器保存 CHAINMAIL 名称列表中列出的所有集群的完整存储库。 更改 CHICAGO 和 CHICAGO2 队列管理器定义:

   ```
   ALTER QMGR REPOS(' ') REPOSNL(CHAINMAIL)
   ```

3. 更改 CHICAGO 和 CHICAGO2 上的 CLUSRCVR 通道。

   位于 CHICAGO 和 CHICAGO2 的 CLUSRCVR 通道定义显示这些通道在集群 CHNSTORE 中可用。 您需要更改集群接收方定义以显示通道可用于 CHAINMAIL 名称列表中列出的所有集群。 在 CHICAGO 处更改集群接收方定义:

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

   在 CHICAGO2 处，输入以下命令:

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSRCVR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

4. 在 CHICAGO 和 CHICAGO2 上变更 CLUSSDR 通道。

   更改两个 CLUSSDR 通道定义以添加名称列表。 在 CHICAGO 处，输入以下命令:

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

   在 CHICAGO2 处，输入以下命令:

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

5. 在 SEATTLE 和 ATLANTA 上创建名称列表。

由于 SEATTLE 和 ATLANTA 将是多个集群的成员，因此必须创建包含集群名称的名称列表。 在 SEATTLE 和 ATLANTA 上定义名称列表，如下所示:

```
DEFINE NAMELIST(CHAINMAIL)
DESCR('List of cluster names')
NAMES(CHNSTORE, MAILORDER)
```

6. 更改 SEATTLE 和 ATLANTA 上的 CLUSRCVR 通道。

   位于 SEATTLE 和 ATLANTA 的 CLUSRCVR 通道定义显示这些通道在集群 CHNSTORE 中可用。 更改 cluster-receive 通道定义以显示这些通道可用于 CHAINMAIL 名称列表中列出的所有集群。 在 SEATTLE 处，输入以下命令:

   ```
   ALTER CHANNEL(CHNSTORE.SEATTLE) CHLTYPE(CLUSRCVR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

   在 ATLANTA 处，输入以下命令:

   ```
   ALTER CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

7. 在 SEATTLE 和 ATLANTA 上变更 CLUSSDR 通道。

   更改两个 CLUSSDR 通道定义以添加名称列表。 在 SEATTLE 处，输入以下命令:

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

   在 ATLANTA 处，输入以下命令:

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

8. 在 HARTFORD 和 OMAHA 上定义 CLUSRCVR 和 CLUSSDR 通道。

   在两个新的队列管理器 HARTFORD 和 OMAHA 上，定义集群接收方和集群发送方通道。 生成定义的顺序无关紧要。 在 HARTFORD 处，输入:

   ```
   DEFINE CHANNEL(MAILORDER.HARTFORD) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(HARTFORD.CHSTORE.COM) CLUSTER(MAILORDER)
   DESCR('Cluster-receiver channel for HARTFORD')

   DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
   DESCR('Cluster-sender channel from HARTFORD to repository at CHICAGO')
   ```

   在 OMAHA 处，输入:

   ```
   DEFINE CHANNEL(MAILORDER.OMAHA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(OMAHA.CHSTORE.COM) CLUSTER(MAILORDER)
   DESCR('Cluster-receiver channel for OMAHA')

   DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
   DESCR('Cluster-sender channel from OMAHA to repository at CHICAGO')
   ```

9. 在 OMAHA 上定义 MORDERQ 队列。

   完成此任务的最后一个步骤是在队列管理器 OMAHA 上定义队列 MORDERQ 。 在 OMAHA 处，输入:

   ```
   DEFINE QLOCAL(MORDERQ) CLUSTER(MAILORDER)
   ```

10. 检查是否已传播集群更改。

    检查您使用先前步骤创建的定义是否已通过集群传播。 在完整存储库队列管理器上发出以下命令:

    ```
    DIS QCLUSTER (MORDERQ)
    DIS CLUSQMGR
    ```

11.

结果

此任务设置的集群显示在 第 273 页的图 48 中。

现在，我们有两个重叠的集群。 这两个集群的完整存储库都保存在 CHICAGO 和 CHICAGO2 中。 在 OMAHA 上运行的邮购应用程序独立于在 CHICAGO 上运行的库存应用程序。 但是， CHNSTORE 集群中的某些队列管理器也位于 MAILORDER 集群中，因此它们可以将消息发送到任一应用程序。 在执行此任务以重叠两个集群之前，请注意发生队列名称冲突的可能性。

假设在集群 CHNSTORE 中的 NEWYORK 和集群 MAILORDER 中的 OMAHA 上，有一个名为 ACCOUNTQ 的队列。 如果重叠集群，然后 SEATTLE 上的应用程序将消息放入队列 ACCOUNTQ，那么该消息可以转至 ACCOUNTQ 的任一实例。

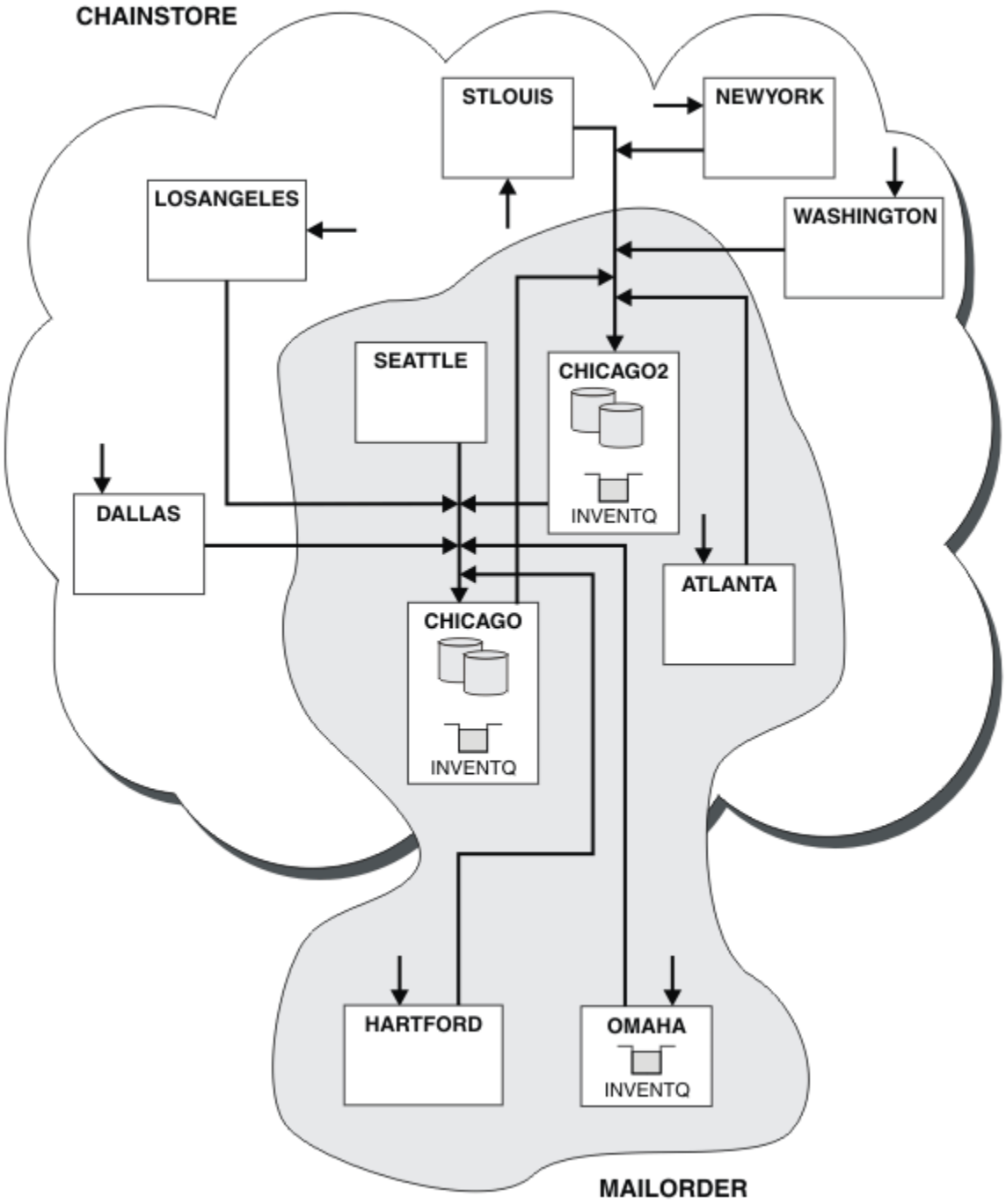


图 *48:* 互连集群

下一步做什么

假设您决定将 MAILORDER 集群与 CHNSTORE 集群合并，以形成一个名为 CHNSTORE 的大型集群。

要将 MAILORDER 集群与 CHNSTORE 集群合并，以便 CHICAGO 和 CHICAGO2 保存完整存储库:

- 更改 CHICAGO 和 CHICAGO2 的队列管理器定义，除去指定名称列表 (CHAINMAIL) 的 REPOSNL 属性，并将其替换为指定集群名称 (CHNSTORE) 的 REPOS 属性。 例如:

```
ALTER QMGR(CHICAGO) REPOSNL(' ') REPOS(CHNSTORE)
```

- 在 MAILORDER 集群中的每个队列管理器上，更改所有通道定义和队列定义，以将 CLUSTER 属性的值从 MAILORDER 更改为 CHNSTORE。 例如，在 HARTFORD 处，输入:

```
ALTER CHANNEL(MAILORDER.HARTFORD) CLUSTER(CHNSTORE)
```

  在 OMAHA 处输入:

```
ALTER QLOCAL(MORDERQ) CLUSTER(CHNSTORE)
```

- 更改所有指定集群名称列表 CHAINMAIL(即 CHICAGO， CHICAGO2， SEATTLE 和 ATLANTA 上的 CLUSRCVR 和 CLUSSDR 通道定义) 的定义，以改为指定集群 CHNSTORE。

从此示例中，您可以看到使用名称列表的优势。 您可以更改名称列表 CHAINMAIL 的值，而不是更改 CHICAGO 和 CHICAGO2 的队列管理器定义。 同样，您可以通过更改名称列表来实现所需结果，而不是在 CHICAGO， CHICAGO2， SEATTLE 和 ATLANTA 处更改 CLUSRCVR 和 CLUSSDR 通道定义。

相关任务

除去集群网络
从网络中除去集群并复原分布式排队配置。

除去集群网络

从网络中除去集群并复原分布式排队配置。

开始之前

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

场景:

- 已按 第 268 页的『将现有网络转换为集群』中所述设置 IBM MQ 集群。
- 现在将从系统中除去此集群。 队列管理器网络将像在实现集群之前那样继续运行。

关于此任务

执行以下步骤以除去集群网络。

过程

1. 从 CHNSTORE 集群中除去集群队列。

   在 CHICAGO 和 CHICAGO2 上，修改队列 INVENTQ 的本地队列定义以从集群中除去队列。 发出以下命令:

   ```
   ALTER QLOCAL(INVENTQ) CLUSTER(' ')
   ```

   变更队列时，将在集群中更新和传播完整存储库中的信息。 使用 MQOO_BIND_NOT_FIXED 的活动应用程序以及使用 MQOO_BIND_AS_Q_DEF (其中已使用 DEFBIND(NOTFIXED)定义队列) 的应用程序在下一次尝试的 MQPUT 或 MQPUT1 调用时失败。 返回原因码 MQRC_UNKNOWN_OBJECT_NAME 。

   您不必首先执行步骤 1 ，但如果不执行，请改为在步骤 4 之后执行。
2. 停止所有有权访问集群队列的应用程序。

停止有权访问集群队列的所有应用程序。 如果未执行此操作，那么在步骤 5 中刷新集群时，某些集群信息可能会保留在本地队列管理器上。 当所有应用程序都已停止并且集群通道已断开连接时，将除去此信息。

3. 从完整存储库队列管理器中除去存储库属性。

   在 CHICAGO 和 CHICAGO2 上，修改队列管理器定义以除去存储库属性。 要执行此操作，请发出以下命令:

   ```
   ALTER QMGR REPOS(' ')
   ```

   队列管理器通知集群中的其他队列管理器不再保存完整存储库。 当其他队列管理器接收到此信息时，您会看到一条消息，指示完整存储库已结束。 您还会看到一条或多条消息，指示不再有任何存储库可用于集群 CHNSTORE。

4. 除去集群通道。

   在 CHICAGO 上，除去集群通道:

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) CLUSTER(' ')
   ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR) CLUSTER(' ')
   ```

   注: 首先发出 CLUSSDR 命令，然后发出 CLUSRCVR 命令很重要。 不要先发出 CLUSRCVR 命令，然后发出 CLUSSDR 命令。 这样做将创建具有 STOPPED 状态的不确定通道。 然后需要发出 START CHANNEL 命令以恢复已停止的通道; 例如， START CHANNEL(CHNSTORE.CHICAGO)。

   您会看到指示集群 CHNSTORE 没有存储库的消息。

   如果未按步骤 1 中所述除去集群队列，请立即执行此操作。

5. 停止集群通道。

   在 CHICAGO 上，使用以下命令停止集群通道:

   ```
   STOP CHANNEL(CHNSTORE.CHICAGO2)
   STOP CHANNEL(CHNSTORE.CHICAGO)
   ```

6. 对集群中的每个队列管理器重复步骤 4 和 5。
7. 停止集群通道，然后从每个队列管理器中除去集群通道和集群队列的所有定义。
8. 可选： 清除队列管理器所保存的高速缓存集群信息。

   虽然队列管理器不再是集群的成员，但它们各自保留有关集群的信息的高速缓存副本。 如果要除去此数据，请参阅任务 第 298 页的『将队列管理器复原到其集群前状态』。

9. 替换 INVENTQ 的远程队列定义

   为了使网络能够继续工作，请在每个队列管理器上替换 INVENTQ 的远程队列定义。

10. 整理集群。

    不再需要删除任何队列或通道定义。

相关任务

添加新的互连集群
添加与现有集群共享某些队列管理器的新集群。

使用网关队列管理器创建两个重叠的集群

按照任务中的指示信息来构造具有网关队列管理器的重叠集群。 将集群用作以下示例的起始点，这些示例用于将消息从一个应用程序中的消息隔离到集群中的其他应用程序。

关于此任务

用于说明隔离集群消息流量的示例集群配置显示在 第 276 页的图 49 中。 该示例在 《集群：使用多个集群传输队列实现应用隔离》中进行了描述。

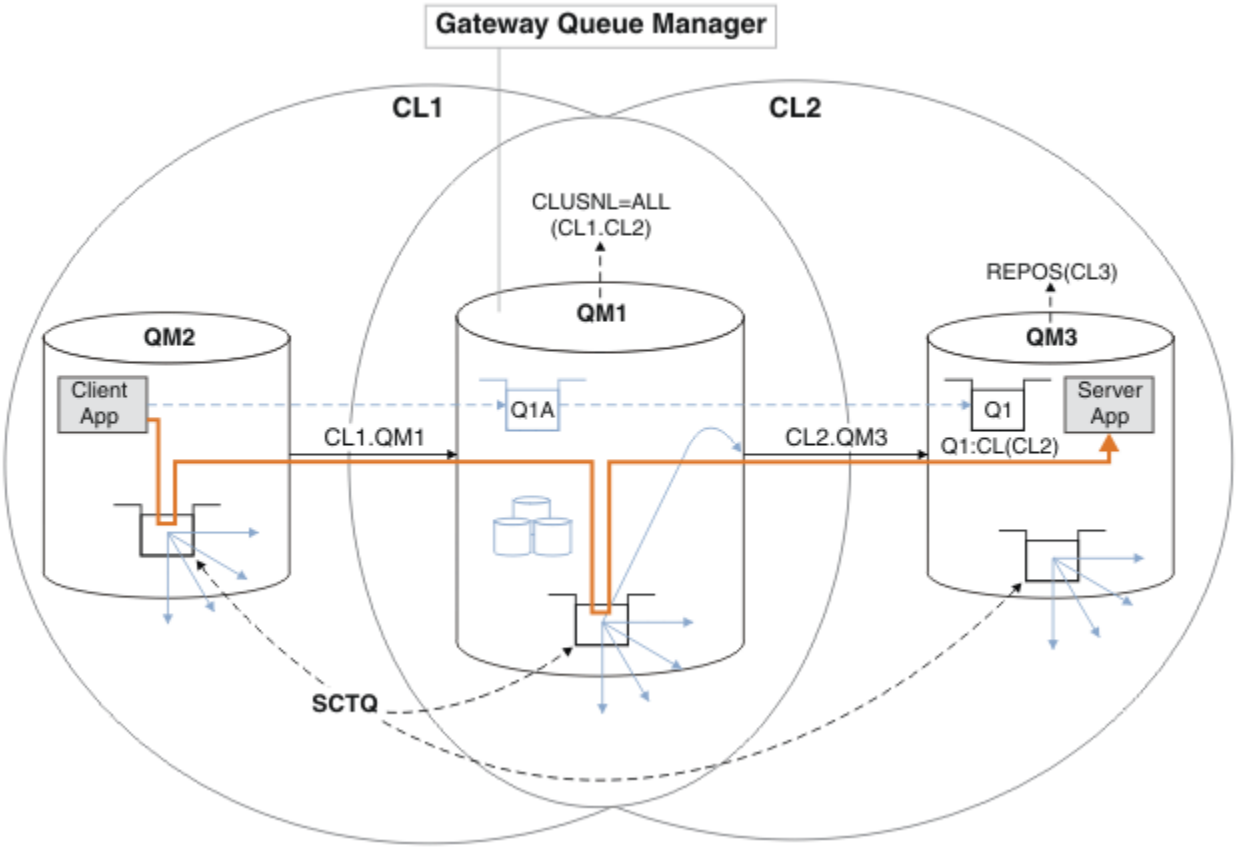


图 *49:* 使用 *IBM MQ* 集群将客户机/服务器应用程序部署到中心和辐射体系结构

为了减少构建示例的步骤，配置保持简单而非真实。 此示例可能表示由两个不同的组织创建的两个集群的集成。 有关更现实的方案，请参阅 集群: 规划如何配置集群传输队列。

执行以下步骤来构造集群。 在以下将消息流量从客户机应用程序隔离到服务器应用程序的示例中使用了这些集群。

指令中添加了几个额外的队列管理器，以便每个集群都有两个存储库。 出于性能方面的考虑，网关队列管理器 QM1 不作为存储库使用。

过程

1. 创建并启动网关队列管理器 QM1 ， 以及队列管理器 QM2 、 QM3 、 QM4 、 QM5 。

   ```
   crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QM n
   strmqm QmgrName
   ```

   注: QM4 和 QM5 是集群的备份完整存储库。
2. 为每个队列管理器定义并启动监听器。

   ```
   *... On QM n
   DEFINE LISTENER(TCP141 n) TRPTYPE(TCP) IPADDR(hostname) PORT(141 n) CONTROL(QMGR) REPLACE
   START LISTENER(TCP141 n)
   ```
3. 为所有集群创建集群名称列表。

   ```
   *... On gateway queue manager QM1
   DEFINE NAMELIST(ALL) NAMES(CL1, CL2) REPLACE
   ```
4. 将 QM2 和 QM4 完整存储库用于 CL1，将 QM3 和 QM5 完整存储库用于 CL2。

   a) 对于 CL1:

```
*... On QM2 and QM4
ALTER QMGR REPOS(CL1) DEFCLXQ(SCTQ)
```

b) 对于 CL2：

```
*... On QM3 and QM5
ALTER QMGR REPOS(CL2) DEFCLXQ(SCTQ)
```

5. 为每个队列管理器和集群添加集群发送方通道和集群接收方通道。

   在 QM2， QM3， QM4 和 QM5 上运行以下命令，其中 *c*， *n* 和 *m* 采用每个队列管理器的 第 277 页的表 27 中显示的值：

表 *27:* 用于创建集群 *1* 和 *2* 的参数值

队列管理器	**集群** ***c***	**其他存储库** ***n***	**此存储库** ***m***
QM2	1	4	2
QM4	1	2	4
QM3	2	5	3
QM5	2	3	5

```
*... On QMm
DEFINE CHANNEL(CLc.QMn) CHLTYPE(CLUSSDR) CONNAME('localhost(141n)') CLUSTER(CLc) REPLACE
DEFINE CHANNEL(CLc.QMm) CHLTYPE(CLUSRCVR) CONNAME('localhost(141m)') CLUSTER(CLc) REPLACE
```

6. 将网关队列管理器 QM1 添加到每个群集。

```
*... On gateway queue manager QM1
DEFINE CHANNEL(CL1.QM2) CHLTYPE(CLUSSDR) CONNAME('localhost(1412)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL1.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL2.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL2) REPLACE
DEFINE CHANNEL(CL2.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL2) REPLACE
```

7. 将本地队列 Q1 添加到集群 CL2 中的队列管理器 QM3 。

```
*... On QM3
DEFINE QLOCAL(Q1) CLUSTER(CL2) REPLACE
```

8. 在网关队列管理器 QM1 上为每个集群队列添加队列别名。

```
*... On gateway queue manager QM1
DEFINE QALIAS(Q1A) CLUSNL(ALL) TARGET(Q1) TARGTYPE(QUEUE) DEFBIND(NOTFIXED) REPLACE
```

9. 将所有群集队列管理器的群集队列管理器别名定义添加到网关队列管理器 QM1 。

   每个需要响应（Reply）的局部存储库都需要一个队列管理器别名。 队列管理器别名的目的是：如果应用程序正在处理包含 **ReplyTo** 属性的请求消息（该属性来自不同集群中的队列管理器），则应用程序可以通过队列管理器别名回复请求。

   注: 在下面的示例中，队列管理器的别名是 QM2 和 QM3 ，与它们所指向的 QM2 和 QM3 的真实队列管理器名称（ **RQMNAME** ）相同。 如果您不希望应用程序知道真正的队列管理器名称，可以使用不同的名称。 例如，如果队列管理器的别名 PAYROLLQM 比 QM2 对应用程序更有意义，则可以将其命名为 。

```
*... On gateway queue manager QM1
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSNL(ALL) REPLACE
```

   提示: 网关队列管理器 QM1 上的队列管理器别名定义传输了指向另一个群集中队列管理器的消息。 参见 集群队列管理器别名。

下一步做什么

1. 通过使用队列别名定义 Q1A 将消息从 QM2 发送到 QM3 上的 Q1 来测试队列别名定义。

   a. 在 QM2 上运行样本程序 **amqsput** 以放置消息。

   ```
   C:\IBM\MQ>amqsput Q1A QM2
   Sample AMQSPUT0 start
   target queue is Q1A
   Sample request message from QM2 to Q1 using Q1A

   Sample AMQSPUT0 end
   ```

   b. 运行样本程序 **amqsget** 以从 QM3 上的 Q1 获取消息

   ```
   C:\IBM\MQ>amqsget Q1 QM3
   Sample AMQSGET0 start
   message <Sample request message from QM2 to Q1 using Q1A>
   no more messages
   Sample AMQSGET0 end
   ```

2. 通过发送请求消息并在临时动态应答队列上接收应答消息来测试队列管理器别名定义。

   该图显示了应答消息返回到临时动态队列 (称为 RQ) 的路径。 连接到 QM3 的服务器应用程序使用队列管理器名称 QM2 打开应答队列。 队列管理器名称 QM2 被定义为网关队列管理器 QM1 上的集群队列管理器别名。 QM3 将回复信息发送到网关队列管理器 。 QM1 QM1 将消息路由到 QM2。

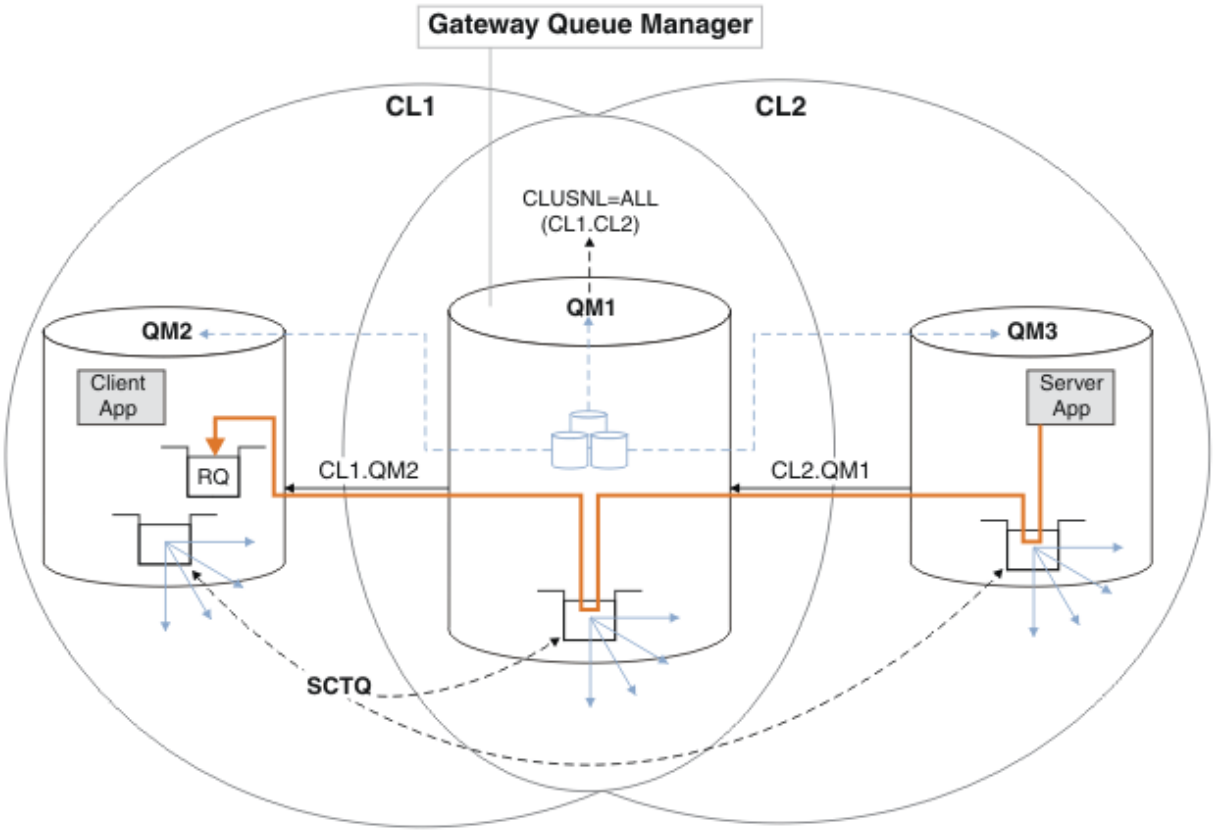


图 *50:* 使用队列管理器别名将应答消息返回到其他集群

路由工作方式如下。 每个集群中的每个队列管理器在网关队列管理器 QM1 上都有一个队列管理器别名定义。 别名在所有集群中进行集群。 从每个别名到队列管理器的灰色虚线箭头显示，每个队列管理器别名都解析为至少一个集群中的实际队列管理器。 在这种情况下， QM2 别名将同时在集群 CL1 和 CL2 中进

行集群，并解析为 CL1 中的实际队列管理器 QM2 。 服务器应用程序使用对队列名称 RQ 的应答以及对队列管理器名称 QM2 的应答来创建应答消息。 CL2 消息被路由到网关队列管理器 QM1 ，因为队列管理器别名定义 QM2 是在群集 QM1 上定义的，而队列管理器 QM2 不在群集 CL2 中。 由于无法将消息发送到目标队列管理器，因此会将其发送到具有别名定义的队列管理器。

网关队列管理器 QM1 将消息放入集群传输队列，以便传输到 QM2 。 网关队列管理器 QM1 将消息路由到 QM2 ，因为 QM1 上 QM2 的队列管理器别名定义将 QM2 定义为真正的目标队列管理器。 定义不是循环的，因为别名定义只能引用实际定义; 别名不能指向自身。 真正的定义由网关队列管理器 QM1 解决，因为 QM1 和 QM2 都在同一个集群 CL1 中。 CL1 网关队列管理器 QM1 从存储库中为 QM2 找到连接信息，并将消息路由到 QM2 。 若要由 QM1 重新路由消息，服务器应用程序必须打开回复队列，并将 DEFBIND 选项设置为 MQBND_BIND_NOT_FIXED 。 如果服务器应用程序以 MQBND_BIND_ON_OPEN 选项打开了回复队列，则邮件不会被重新路由，而是进入死信队列。

a. 使用 QM3 上的触发器创建集群请求队列。

```
*... On QM3
DEFINE QLOCAL(QR) CLUSTER(CL2) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
```

b. 在网关队列管理器 QM1 上创建 QR 的集群队列别名定义。

```
*... On gateway queue manager QM1
DEFINE QALIAS(QRA) CLUSNL(ALL) TARGET(QR) TARGTYPE(QUEUE) DEFBIND(NOTFIXED) REPLACE
```

c. 创建进程定义以在 QM3 上启动样本回送程序 **amqsech** 。

```
*... On QM3
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

d. 在 QM2 上为样本程序 **amqsreq** 创建模型队列以创建临时动态应答队列。

```
*... On QM2
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

e. 通过使用队列别名定义 QRA 将请求从 QM2 发送到 QM3 上的 QR 来测试队列管理器别名定义。

i) 在 QM3 上运行触发器监视器程序。

```
runmqtrm -m QM3
```

输出为

```
C:\IBM\MQ>runmqtrm -m QM3
5724-H72 (C) Copyright IBM Corp. 1994, 2025. ALL RIGHTS RESERVED.
01/02/2012  16:17:15: IBM MQ trigger monitor started.

_____________________________________________________
01/02/2012  16:17:15: Waiting for a trigger message
```

ii) 在 QM2 上运行样本程序 **amqsreq** 以放入请求并等待应答。

```
C:\IBM\MQ>amqsreq QRA QM2
Sample AMQSREQ0 start
server queue is QRA
replies to 4F2961C802290020
A request message from QM2 to QR on QM3

response <A request message from QM2 to QR on QM3>
no more replies
Sample AMQSREQ0 end
```

相关概念

访问控制和多个集群传输队列

集群: 使用多个集群传输队列进行应用程序隔离

相关任务

集群: 规划如何配置集群传输队列

第 256 页的『将队列管理器添加到集群: 单独的传输队列』

遵循以下指示信息将队列管理器添加到您创建的集群。 使用多个集群传输队列将消息传输到集群队列和主题。

添加远程队列定义以隔离从网关队列管理器发送的消息

修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 该解决方案使用集群队列远程定义以及单独的发送方通道和传输队列。

开始之前

通过执行该任务中的步骤，构造 使用 IBM MQ 集群将客户机/服务器应用程序部署到中心和辐射体系结构 in 第 275 页的『使用网关队列管理器创建两个重叠的集群』 中显示的重叠集群。

关于此任务

该解决方案使用分布式排队将 `Server App` 应用程序的消息与网关队列管理器上的其他消息流量分开。 必须在 `QM1` 上定义集群远程队列定义，以将消息转移到不同的传输队列和不同的通道。 远程队列定义必须包含对仅存储 `QM3` 上 `Q1` 的消息的特定传输队列的引用。 在 第 280 页的图 51 中，集群队列别名 `Q1A` 由远程队列定义 `Q1R` 补充，并添加了传输队列和发送方通道。

在此解决方案中，将使用公共 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 返回任何应答消息。

此解决方案的优势在于，可以轻松地在同一集群中的同一队列管理器上分隔多个目标队列的流量。 该解决方案的缺点是无法在不同队列管理器上的 `Q1` 的多个副本之间使用集群工作负载均衡。 要克服此缺点，请参阅 第 282 页的『添加集群传输队列以隔离从网关队列管理器发送的集群消息流量』。 您还必须管理从一个传输队列到另一个传输队列的交换机。

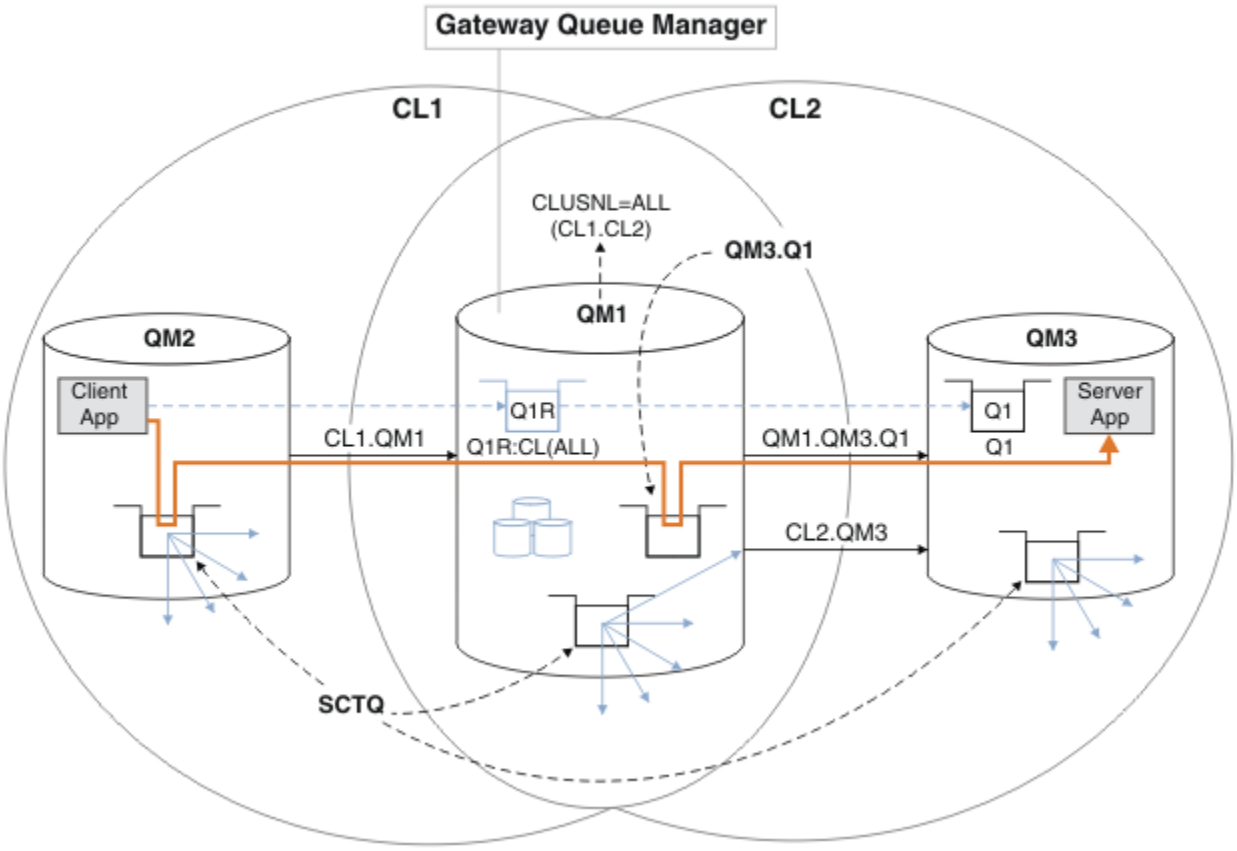


图 *51:* 使用远程队列定义部署到中心和辐射集群体系结构的客户机/服务器应用程序

过程

1. 创建通道以将 Q1 的消息流量与网关队列管理器分开

   a) 在网关队列管理器 QM1 上创建到目标队列管理器 QM3 的发送方通道。

   ```
   DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(SDR) CONNAME(QM3HostName(1413)) XMITQ(QM3.Q1) REPLACE
   ```

   b) 在目标队列管理器 QM3 上创建接收方通道。

   ```
   DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(RCVR) REPLACE
   ```

2. 在网关队列管理器上为到 Q1 的消息流量创建传输队列

   ```
   DEFINE QLOCAL(QM3.Q1) USAGE(XMITQ) REPLACE
   START CHANNEL(QM1.QM3.Q1)
   ```

   启动与传输队列关联的通道，将传输队列与通道关联。 一旦传输队列与通道相关联，该通道将自动启动。

3. 使用集群远程队列定义来补充网关队列管理器上 Q1 的集群队列别名定义。

   ```
   DEFINE QREMOTE CLUSNL(ALL) RNAME(Q1) RQMNAME(QM3) XMITQ(QM3.Q1) REPLACE
   ```

下一步做什么

通过使用网关队列管理器 QM1 上的集群队列远程定义 Q1R 从 QM2 向 QM3 上的 Q1 发送消息来测试配置。

1. 在 QM2 上运行样本程序 **amqsput** 以放置消息。

   ```
   C:\IBM\MQ>amqsput Q1R QM2
   Sample AMQSPUT0 start
   target queue is Q1R
   Sample request message from QM2 to Q1 using Q1R

   Sample AMQSPUT0 end
   ```

2. 运行样本程序 **amqsget** 以从 QM3 上的 Q1 获取消息

   ```
   C:\IBM\MQ>amqsget Q1 QM3
   Sample AMQSGET0 start
   message <Sample request message from QM2 to Q1 using Q1R>
   no more messages
   Sample AMQSGET0 end
   ```

相关概念

访问控制和多个集群传输队列

相关任务

添加集群传输队列以隔离从网关队列管理器发送的集群消息流量

修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 传输到应用程序之后，解决方案会使用额外的集群传输队列将消息流量分离到集群中的单个队列管理器。

添加集群和集群传输队列以隔离从网关队列管理器发送的集群消息流量

修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 传输到应用程序之后，解决方案会使用其他集群将消息隔离到特定集群队列。

将缺省值更改为单独的集群传输队列以隔离消息流量

您可以更改队列管理器在传输队列上存储集群队列或主题的消息的缺省方式。 更改缺省值为您提供了一种方法来隔离网关队列管理器上的集群消息。

集群: 使用多个集群传输队列进行应用程序隔离

集群: 规划如何配置集群传输队列

第 256 页的『将队列管理器添加到集群: 单独的传输队列』

遵循以下指示信息将队列管理器添加到您创建的集群。 使用多个集群传输队列将消息传输到集群队列和主题。

添加集群传输队列以隔离从网关队列管理器发送的集群消息流量

修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 传输到应用程序之后，解决方案会使用额外的集群传输队列将消息流量分离到集群中的单个队列管理器。

开始之前

1. 网关队列管理器必须在 IBM WebSphere MQ 7.5 或更高版本上。
2. 通过执行该任务中的步骤，构造 使用 IBM MQ 集群将客户机/服务器应用程序部署到中心和辐射体系结构 in 第 275 页的『使用网关队列管理器创建两个重叠的集群』 中显示的重叠集群。

关于此任务

在网关队列管理器 QM1 上，添加传输队列并设置其队列属性 CLCHNAME。 将 CLCHNAME 设置为 QM3 上集群接收方通道的名称; 请参阅 第 283 页的图 52。

与 第 280 页的『添加远程队列定义以隔离从网关队列管理器发送的消息』中描述的解决方案相比，此解决方案具有许多优点:

- 它需要更少的额外定义。
- 它支持在同一集群 CL2 中的不同队列管理器上的目标队列 Q1 的多个副本之间进行工作负载均衡。
- 当通道重新启动时，网关队列管理器会自动切换到新配置，而不会释放任何消息。
- 网关队列管理器继续按接收消息的顺序转发消息。 它执行此操作，即使在发生交换机时，队列 Q1 的消息仍在 QM3 上 SYSTEM.CLUSTER.TRANSMIT.QUEUE。

用于在 第 283 页的图 52 中隔离集群消息流量的配置不会导致与在 第 280 页的『添加远程队列定义以隔离从网关队列管理器发送的消息』中使用远程队列的配置一样大的流量隔离。 如果 CL2 中的队列管理器 QM3 正在托管多个不同的集群队列和服务器应用程序，那么所有这些队列将共享集群通道 CL2.QM3，从而将 QM1 连接到 QM3。 第 283 页的图 52 中通过灰色箭头说明了其他流，灰色箭头表示从 SYSTEM.CLUSTER.TRANSMIT.QUEUE 到集群发送方通道 CL2.QM3 的潜在集群消息流量。

补救措施是将队列管理器限制为在特定集群中托管一个集群队列。 如果队列管理器已在托管多个集群队列，那么要满足此限制，必须创建另一个队列管理器或创建另一个集群; 请参阅 第 285 页的『添加集群和集群传输队列以隔离从网关队列管理器发送的集群消息流量』。

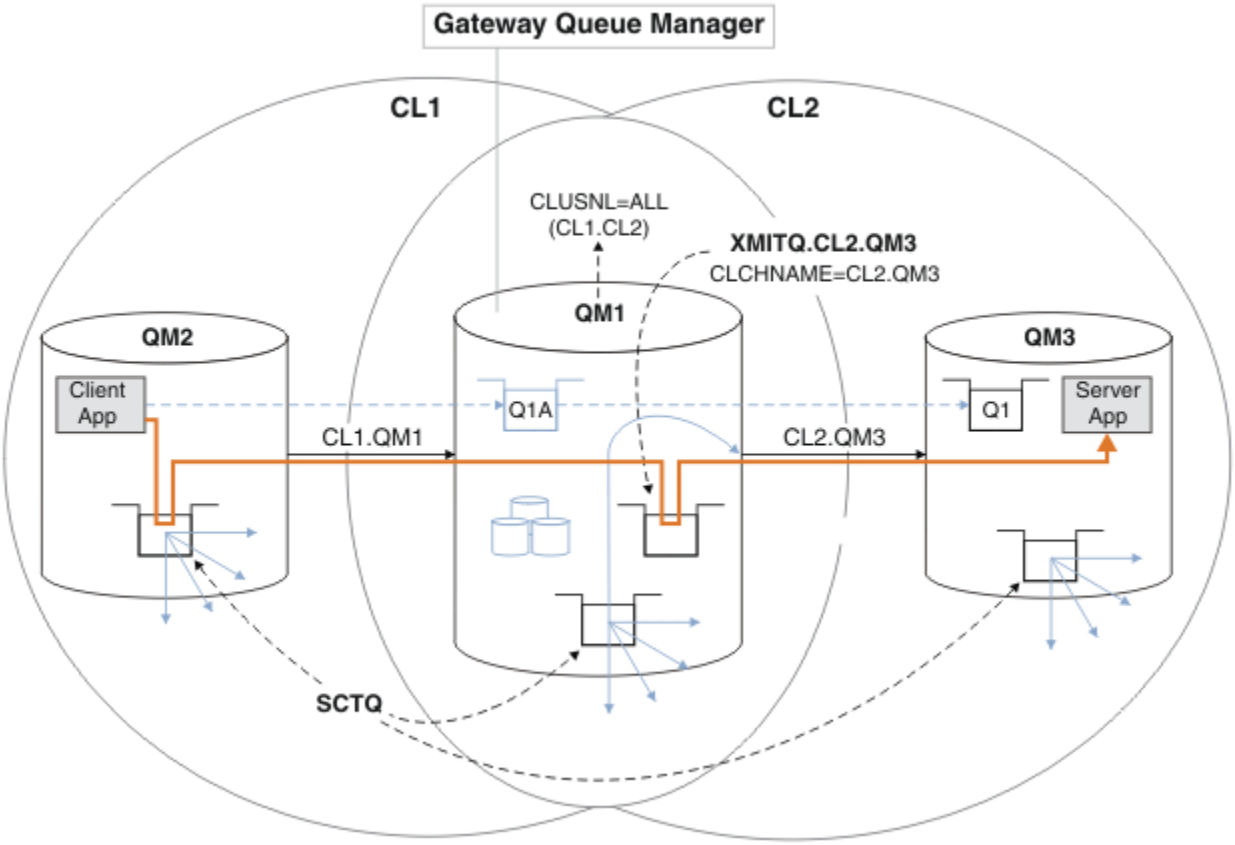


图 *52:* 使用其他集群传输队列部署到中心和辐射体系结构的客户机/服务器应用程序。

过程

1. 在网关队列管理器 QM1 上为集群发送方通道 CL2.QM3 创建额外的集群传输队列。

```
 *... on QM1
DEFINE QLOCAL(XMITQ.CL2.QM3) USAGE(XMITQ) CLCHNAME(CL2.QM3)
```

2. 切换到使用传输队列 XMITQ.CL2.QM3。
   a) 停止集群发送方通道 CL2.QM3。

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

   响应是接受该命令:

```
AMQ8019: Stop IBM MQ channel accepted.
```

   b) 检查通道 CL2.QM3 是否已停止

   如果通道未停止，那么可以使用 FORCE 选项再次运行 **STOP CHANNEL** 命令。 设置 FORCE 选项的示例是如果通道未停止，并且您无法重新启动其他队列管理器来同步通道。

```
*... On QM1
start
```

   响应是通道状态的摘要

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)              CHLTYPE(CLUSSDR)
```

```
CONNAME(127.0.0.1(1413))      CURRENT
RQMNAME(QM3)                  STATUS(STOPPED)
SUBSTATE(MQGET)               XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

c) 启动通道 CL2.QM3。

```
*... On QM1
START CHANNEL(CL2.QM3)
```

响应是接受该命令:

```
AMQ8018: Start IBM MQ channel accepted.
```

d) 检查通道是否已启动。

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

响应是通道状态的摘要:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                  CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))          CURRENT
RQMNAME(QM3)                      STATUS(RUNNING)
SUBSTATE(MQGET)                   XMITQ(XMITQ.CL2.QM3)
```

e) 检查传输队列是否已切换。

监视网关队列管理器错误日志以获取消息 “ AMQ7341 **通道** CL2.QM3 **是** XMITQ.CL2.QM3 ”。

下一步做什么

通过使用队列别名定义 Q1A 将消息从 QM2 发送到 QM3 上的 Q1 ，测试单独的传输队列

1. 在 QM2 上运行样本程序 **amqsput** 以放置消息。

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```

2. 运行样本程序 **amqsget** 以从 QM3 上的 Q1 获取消息

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

相关概念

访问控制和多个集群传输队列

第 240 页的『使用集群传输队列和集群发送方通道』

集群队列管理器之间的消息存储在集群传输队列上，并由集群发送方通道转发。 在任何时间点，集群发送方通道都与一个传输队列相关联。 如果更改通道的配置，那么它可能会在下次启动时切换到其他传输队列。 此交换机的处理是自动化的，并且是事务性的。

相关任务

添加远程队列定义以隔离从网关队列管理器发送的消息
修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 该解决方案使用集群队列远程定义以及单独的发送方通道和传输队列。

添加集群和集群传输队列以隔离从网关队列管理器发送的集群消息流量
修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 传输到应用程序之后，解决方案会使用其他集群将消息隔离到特定集群队列。

将缺省值更改为单独的集群传输队列以隔离消息流量
您可以更改队列管理器在传输队列上存储集群队列或主题的消息的缺省方式。 更改缺省值为您提供了一种方法来隔离网关队列管理器上的集群消息。

集群: 使用多个集群传输队列进行应用程序隔离

集群: 规划如何配置集群传输队列

第 256 页的『将队列管理器添加到集群: 单独的传输队列』
遵循以下指示信息将队列管理器添加到您创建的集群。 使用多个集群传输队列将消息传输到集群队列和主题。

添加集群和集群传输队列以隔离从网关队列管理器发送的集群消息流量

修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 传输到应用程序之后，解决方案会使用其他集群将消息隔离到特定集群队列。

开始之前

编写任务中的步骤以修改 第 283 页的图 52 中说明的配置。

1. 网关队列管理器必须在 IBM WebSphere MQ 7.5 或更高版本上。
2. 通过执行该任务中的步骤，构造 使用 IBM MQ 集群将客户机/服务器应用程序部署到中心和辐射体系结构 in 第 275 页的『使用网关队列管理器创建两个重叠的集群』 中显示的重叠集群。
3. 执行 第 282 页的『添加集群传输队列以隔离从网关队列管理器发送的集群消息流量』 中的 第 283 页的图 52 中的步骤以在不使用其他集群的情况下创建解决方案。 将其用作此任务中步骤的基础。

关于此任务

如果目标集群队列是队列管理器上的唯一集群队列，那么用于将消息流量隔离到 第 282 页的『添加集群传输队列以隔离从网关队列管理器发送的集群消息流量』 中的单个应用程序的解决方案有效。 如果不是，你有两个选择。 将队列移至另一个队列管理器，或者创建集群以将队列与队列管理器上的其他集群队列隔离。

此任务将指导您完成添加集群以隔离目标队列的步骤。 仅为此目的添加集群。 在实践中，当您正在设计集群和集群命名方案时，请执行系统地隔离某些应用程序的任务。 每次队列需要隔离时，都添加一个集群，最终可能会有许多要管理的集群。 在此任务中，通过添加集群 CL3 以隔离 QM3 上的 Q1 来更改 第 282 页的『添加集群传输队列以隔离从网关队列管理器发送的集群消息流量』 中的配置。 应用程序将继续在整个更改过程中运行。

新定义和已更改的定义将在 第 286 页的图 53 中突出显示。 更改的摘要如下所示: 创建集群，这意味着您还必须创建新的完整集群存储库。 在此示例中， QM3 是 CL3 的其中一个完整存储库。 为 QM1 创建集群发送方和集群接收方通道，以将网关队列管理器添加到新集群。 更改 Q1 的定义以将其切换到 CL3。 修改网关队列管理器上的集群名称列表，并添加集群传输队列以使用新的集群通道。 最后，将队列别名 Q1A 切换到新的集群名称列表。

IBM MQ 无法自动将消息从您在 第 282 页的『添加集群传输队列以隔离从网关队列管理器发送的集群消息流量』 中添加的传输队列 XMITQ.CL2.QM3 传输到新的传输队列 XMITQ.CL3.QM3。 仅当两个传输队列都由同一集群发送方通道提供服务时，它才能自动传输消息。 相反，该任务描述了手动执行交换机的一种方法，这可能适合您。 传输完成后，您可以选择将缺省集群传输队列用于 QM3 上的其他 CL2 集群队列。 或者，您可以继续使用 XMITQ.CL2.QM3。 如果您决定还原为缺省集群传输队列，那么网关队列管理器会自动为您管理交换机。

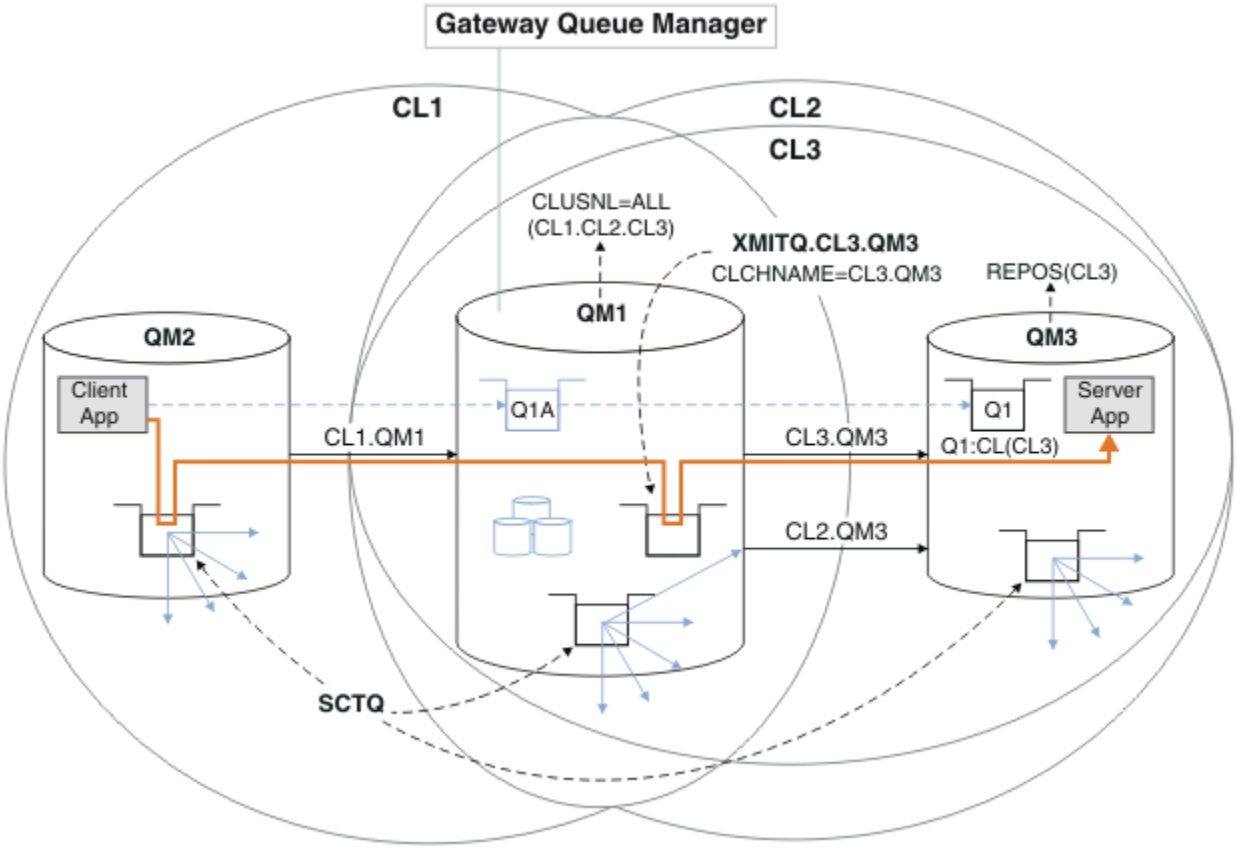


图 *53:* 使用其他集群来分隔网关队列管理器中的消息流量，该消息流量转至同一队列管理器上的多个集群队列中的一个

过程

1. 更改队列管理器 QM3 和 QM5，使它们成为 CL2 和 CL3 的存储库。

   要使一个队列管理器成为多个集群的成员，它必须使用集群名称列表来标识它是其成员的集群。

   ```
   *... On QM3 and QM5
   DEFINE NAMELIST(CL23) NAMES(CL2, CL3) REPLACE
   ALTER QMGR REPOS(' ') REPOSNL(CL23)
   ```

2. 定义 CL3 的队列管理器 QM3 和 QM5 之间的通道。

   ```
   *... On QM3
   DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
   DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSRCVR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE

   *... On QM5
   DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
   DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
   ```

3. 将网关队列管理器添加到 CL3。

   通过将 QM1 添加到 CL3 作为部分存储库来添加网关队列管理器。 通过将集群发送方通道和集群接收方通道添加到 QM1 来创建部分存储库。

   此外，将 CL3 添加到连接到网关队列管理器的所有集群的名称列表。

   ```
   *... On QM1
   DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
   DEFINE CHANNEL(CL3.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL3) REPLACE
   ALTER NAMELIST(ALL) NAMES(CL1, CL2, CL3)
   ```

4. 将集群传输队列添加到网关队列管理器 QM1，以用于发送到 QM3 上的 CL3 的消息。

最初，停止集群发送方通道从传输队列传输消息，直到您准备好切换传输队列为止。

```
*... On QM1
DEFINE QLOCAL(XMITQ.CL3.QM3) USAGE(XMITQ) CLCHNAME(CL3.QM3) GET(DISABLED) REPLACE
```

5. 从现有集群传输队列 XMITQ.CL2.QM3 中漏出消息。

   此子过程旨在保留 Q1 中的消息顺序，以与它们到达网关队列管理器的顺序相匹配。 使用集群时，不会完全保证消息排序，但很可能。 如果需要保证消息排序，那么应用程序必须定义消息顺序; 请参阅 从队列中检索消息的顺序。

   a) 将 QM3 上的目标队列 Q1 从 CL2 更改为 CL3。

   ```
   *... On QM3
   ALTER QLOCAL(Q1) CLUSTER(CL3)
   ```

   b) 监视 XMITQ.CL3.QM3 ，直到开始向其传递消息为止。

   当 Q1 到 CL3 的交换机传播到网关队列管理器时，将开始将消息传递到 XMITQ.CL3.QM3 。

   ```
   *... On QM1
   DISPLAY QUEUE(XMITQ.CL3.QM3) CURDEPTH
   ```

   c) 监视 XMITQ.CL2.QM3 ，直到它没有消息等待传递到 QM3 上的 Q1 。

   注: XMITQ.CL2.QM3 可能正在存储属于 CL2 的 QM3 上其他队列的消息，在这种情况下，深度可能不会变为零。

   ```
   *... On QM1
   DISPLAY QUEUE(XMITQ.CL2.QM3) CURDEPTH
   ```

   d) 启用从新集群传输队列 XMITQ.CL3.QM3 获取

   ```
   *... On QM1
   ALTER QLOCAL(XMITQ.CL3.QM3) GET(ENABLED)
   ```

6. 如果不再需要旧集群传输队列 XMITQ.CL2.QM3，请将其除去。

   CL2 on QM3 中集群队列的消息还原为使用网关队列管理器 QM1 上的缺省集群传输队列。 缺省集群传输队列为 SYSTEM.CLUSTER.TRANSMIT.QUEUE 或 SYSTEM.CLUSTER.TRANSMIT.CL2.QM3。 取决于 QM1 上的队列管理器属性 **DEFCLXQ** 的值是 SCTQ 还是 CHANNEL。 当集群发送方通道 CL2.QM3 下次启动时，队列管理器会自动从 XMITQ.CL2.QM3 传输消息。

   a) 将传输队列 XMITQ.CL2.QM3 从集群传输队列更改为正常传输队列。

   这将中断传输队列与任何集群发送方通道的关联。 作为响应， IBM MQ 会在集群发送方通道下次启动时自动将消息从 XMITQ.CL2.QM3 传输到缺省集群传输队列。 在此之前， CL2 on QM3 的消息将继续放在 XMITQ.CL2.QM3 上。

   ```
   *... On QM1
   ALTER QLOCAL(XMITQ.CL2.QM3) CLCHNAME(' ')
   ```

   b) 停止集群发送方通道 CL2.QM3。

   停止并重新启动集群发送方通道将启动从 XMITQ.CL2.QM3 到缺省集群传输队列的消息传输。 通常，您将手动停止并启动通道以启动传输。 如果通道在其断开连接时间间隔到期时关闭后重新启动，那么传输将自动启动。

   ```
   *... On QM1
   STOP CHANNEL(CL2.QM3)
   ```

   响应是接受该命令:

   ```
   AMQ8019: Stop IBM MQ channel accepted.
   ```

c) 检查通道 CL2.QM3 是否已停止

如果通道未停止，那么可以使用 FORCE 选项再次运行 **STOP CHANNEL** 命令。 设置 FORCE 选项的示例是如果通道未停止，并且您无法重新启动其他队列管理器来同步通道。

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

响应是通道状态的摘要

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))                CURRENT
RQMNAME(QM3)                            STATUS(STOPPED)
SUBSTATE(MQGET)                         XMITQ(XMITQ.CL2.QM3)
```

d) 启动通道 CL2.QM3。

```
*... On QM1
START CHANNEL(CL2.QM3)
```

响应是接受该命令:

```
AMQ8018: Start IBM MQ channel accepted.
```

e) 检查通道是否已启动。

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

响应是通道状态的摘要:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)      CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))  CURRENT
RQMNAME(QM3)        STATUS(RUNNING)
SUBSTATE(MQGET)      XMITQ(SYSTEM.CLUSTER.TRANSMIT. QUEUE|CL2.QM3)
```

f) 监视网关队列管理器错误日志以获取消息 “ AMQ7341 **通道** CL2.QM3 **是** SYSTEM.CLUSTER.TRANSMIT。 *QUEUE|CL2.QM3* ”.

g) 删除集群传输队列 XMITQ.CL2.QM3。

```
*... On QM1
DELETE QLOCAL(XMITQ.CL2.QM3)
```

下一步做什么

通过使用队列别名定义 Q1A 将消息从 QM2 发送到 QM3 上的 Q1 来测试单独集群的队列

1. 在 QM2 上运行样本程序 **amqsput** 以放置消息。

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```

2. 运行样本程序 **amqsget** 以从 QM3 上的 Q1 获取消息

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

相关概念

访问控制和多个集群传输队列

第 240 页的『使用集群传输队列和集群发送方通道』

集群队列管理器之间的消息存储在集群传输队列上，并由集群发送方通道转发。 在任何时间点，集群发送方通道都与一个传输队列相关联。 如果更改通道的配置，那么它可能会在下次启动时切换到其他传输队列。 此交换机的处理是自动化的，并且是事务性的。

相关任务

添加远程队列定义以隔离从网关队列管理器发送的消息

修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 该解决方案使用集群队列远程定义以及单独的发送方通道和传输队列。

添加集群传输队列以隔离从网关队列管理器发送的集群消息流量

修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 传输到应用程序之后，解决方案会使用额外的集群传输队列将消息流量分离到集群中的单个队列管理器。

将缺省值更改为单独的集群传输队列以隔离消息流量

您可以更改队列管理器在传输队列上存储集群队列或主题的消息的缺省方式。 更改缺省值为您提供了一种方法来隔离网关队列管理器上的集群消息。

集群: 使用多个集群传输队列进行应用程序隔离

集群: 规划如何配置集群传输队列

第 256 页的『将队列管理器添加到集群: 单独的传输队列』

遵循以下指示信息将队列管理器添加到您创建的集群。 使用多个集群传输队列将消息传输到集群队列和主题。

将缺省值更改为单独的集群传输队列以隔离消息流量

您可以更改队列管理器在传输队列上存储集群队列或主题的消息的缺省方式。 更改缺省值为您提供了一种方法来隔离网关队列管理器上的集群消息。

开始之前

1. 网关队列管理器必须在 IBM WebSphere MQ 7.5 或更高版本上。
2. 通过执行该任务中的步骤，构造 使用 IBM MQ 集群将客户机/服务器应用程序部署到中心和辐射体系结构 in 第 275 页的『使用网关队列管理器创建两个重叠的集群』 中显示的重叠集群。

关于此任务

要实现具有多个集群队列的体系结构，网关队列管理器必须在 IBM WebSphere MQ 7.5 或更高版本上。 使用多个集群传输队列的所有操作都是更改网关队列管理器上的缺省集群传输队列类型。 将 QM1 上队列管理器属性 **DEFCLXQ** 的值从 SCTQ 更改为 CHANNEL ; 请参阅 第 290 页的图 54。 该图显示了一个消息流。 对于流向其他队列管理器或流向其他集群的流，队列管理器会创建其他永久动态集群传输队列。 每个集群发送方通道从不同的集群传输队列传输消息。

除非您是首次将网关队列管理器连接到集群，否则更改不会立即生效。 该任务包含管理对现有配置的更改的典型案例的步骤。 要设置队列管理器以在首次加入集群时使用单独的集群传输队列，请参阅 第 256 页的『将队列管理器添加到集群: 单独的传输队列』。

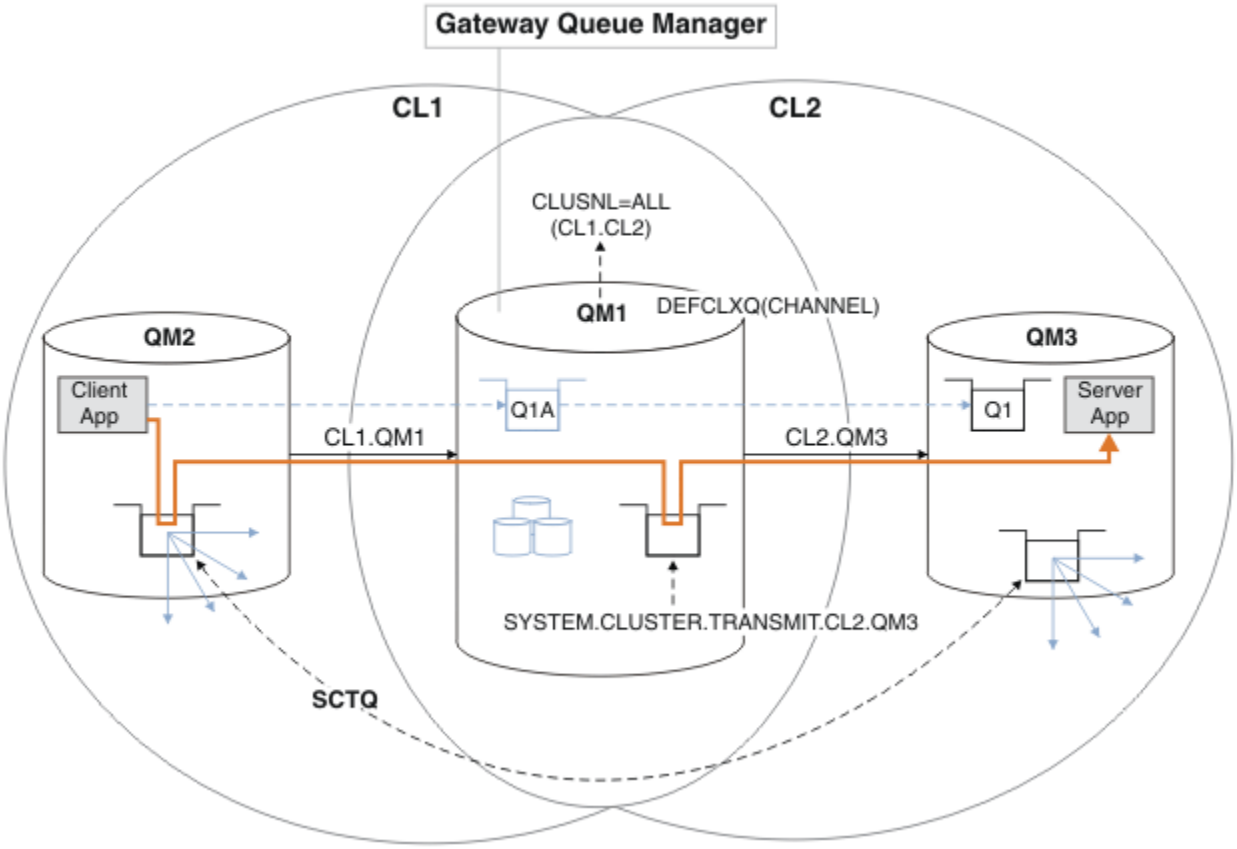


图 *54:* 客户机/服务器应用程序已部署到具有网关队列管理器上的独立集群传输队列的中心和辐射体系结构。

过程

1. 更改网关队列管理器以使用单独的集群传输队列。

   ```
   *... On QM1
   ALTER QMGR DEFCLXQ(CHANNEL)
   ```

2. 切换到单独的集群传输队列。

   未运行的任何集群发送方通道在下次启动时将切换到使用单独的集群传输队列。

   要切换正在运行的通道，请重新启动队列管理器，或者执行以下步骤:

   a) 列出正在使用 SYSTEM.CLUSTER.TRANSMIT.QUEUE 运行的集群发送方通道。

   ```
   *... On QM1
   DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
   ```

   响应是通道状态报告的列表:

   ```
   AMQ8417: Display Channel Status details.
   CHANNEL(CL1.QM2)            CHLTYPE(CLUSSDR)
   CONNAME(127.0.0.1(1412))    CURRENT
   RQMNAME(QM2)                STATUS(RUNNING)
   SUBSTATE(MQGET)             XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
   AMQ8417: Display Channel Status details.
   CHANNEL(CL2.QM3)            CHLTYPE(CLUSSDR)
   CONNAME(127.0.0.1(1413))    CURRENT
   RQMNAME(QM3)                STATUS(RUNNING)
   SUBSTATE(MQGET)             XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
   ```

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)             CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1415))     CURRENT
RQMNAME(QM5)                 STATUS(RUNNING)
SUBSTATE(MQGET)              XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)             CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1414))     CURRENT
RQMNAME(QM4)                 STATUS(RUNNING)
SUBSTATE(MQGET)              XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

b) 停止正在运行的通道

对于列表中的每个通道，运行以下命令:

```
*... On QM1
STOP CHANNEL(ChannelName)
```

其中， *ChannelName* 是每个 CL1.QM2, CL1.QM4, CL1.QM3, CL1.QM5。

响应是接受该命令:

```
AMQ8019: Stop IBM MQ channel accepted.
```

c) 监视哪些通道已停止

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
```

响应是仍在运行的通道和已停止的通道的列表:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)             CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1412))     CURRENT
RQMNAME(QM2)                 STATUS(STOPPED)
SUBSTATE( )                  XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)             CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))     CURRENT
RQMNAME(QM3)                 STATUS(STOPPED)
SUBSTATE( )                  XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)             CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1415))     CURRENT
RQMNAME(QM5)                 STATUS(STOPPED)
SUBSTATE( )                  XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)             CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1414))     CURRENT
RQMNAME(QM4)                 STATUS(STOPPED)
SUBSTATE( )                  XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

d) 启动每个已停止的通道。

对所有正在运行的通道执行此步骤。 如果通道未停止，那么可以使用 FORCE 选项再次运行 **STOP CHANNEL** 命令。 设置 FORCE 选项的示例是如果通道未停止，并且您无法重新启动其他队列管理器来同步通道。

```
*... On QM1
START CHANNEL(CL2.QM5)
```

响应是接受该命令:

```
AMQ8018: Start IBM MQ channel accepted.
```

e) 监视正在切换的传输队列。

监视网关队列管理器错误日志以获取消息 “ AMQ7341 **通道** CL2.QM3 **是** SYSTEM.CLUSTER.TRANSMIT。 *QUEUE*|*CL2.QM3* ”.

f) 检查是否不再使用 SYSTEM.CLUSTER.TRANSMIT.QUEUE

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
DISPLAY QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE) CURDEPTH
```

响应是通道状态报告的列表以及 SYSTEM.CLUSTER.TRANSMIT.QUEUE 的深度:

```
AMQ8420: Channel Status not found.
AMQ8409: Display Queue details.
QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE)    TYPE(QLOCAL)
CURDEPTH(0)
```

g) 监视已启动的通道

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

响应是通道列表，在这种情况下，已使用新的缺省集群传输队列运行:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1412))                CURRENT
RQMNAME(QM2)                            STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL1.QM2)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))                CURRENT
RQMNAME(QM3)                            STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM3)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1415))                CURRENT
RQMNAME(QM5)                            STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM5)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1414))                CURRENT
RQMNAME(QM4)                            STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL1.QM4)
```

下一步做什么

1. 通过将消息从 QM2 发送到 QM3 上的 Q1 ，使用队列别名定义 Q1A 解析队列名称，测试自动定义的集群传输队列

   a. 在 QM2 上运行样本程序 **amqsput** 以放置消息。

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```

b. 运行样本程序 **amqsget** 以从 QM3 上的 Q1 获取消息

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

2. 请考虑是否通过在生成集群队列消息的队列管理器上配置集群队列的安全性来重新配置安全性。

相关概念

访问控制和多个集群传输队列

相关任务

添加远程队列定义以隔离从网关队列管理器发送的消息
修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 该解决方案使用集群队列远程定义以及单独的发送方通道和传输队列。

添加集群传输队列以隔离从网关队列管理器发送的集群消息流量
修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 传输到应用程序之后，解决方案会使用额外的集群传输队列将消息流量分离到集群中的单个队列管理器。

添加集群和集群传输队列以隔离从网关队列管理器发送的集群消息流量
修改使用网关队列管理器的重叠集群的配置。 修改消息从网关队列管理器传输到应用程序后，不使用与其他集群消息相同的传输队列或通道。 传输到应用程序之后，解决方案会使用其他集群将消息隔离到特定集群队列。

集群: 使用多个集群传输队列进行应用程序隔离

集群: 规划如何配置集群传输队列

第 256 页的『将队列管理器添加到集群: 单独的传输队列』
遵循以下指示信息将队列管理器添加到您创建的集群。 使用多个集群传输队列将消息传输到集群队列和主题。

从队列管理器中除去集群队列

在多伦多禁用 INVENTQ 队列。 将所有库存消息发送到纽约，并在多伦多的 INVENTQ 队列为空时将其删除。

开始之前

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

场景：

- 已按 第 260 页的『添加托管队列的队列管理器』中所述设置 INVENTORY 集群。 它包含四个队列管理器。 LONDON 和 NEWYORK 都保存完整存储库。 PARIS 和 TORONTO 保存部分存储库。 库存应用程序在纽约和多伦多的系统上运行，并由 INVENTQ 队列上的消息到达驱动。
- 由于工作负载减少，您不再希望在多伦多运行库存应用程序。 您想要禁用由队列管理器 TORONTO 托管的 INVENTQ 队列，并将 TORONTO 订阅源消息发送到 NEWYORK 中的 INVENTQ 队列。
- 所有四个系统之间都存在网络连接。
- 网络协议是 TCP。

关于此任务

执行以下步骤以除去集群队列。

过程

1. 指示队列不再可用。

   要从集群中除去队列，请从本地队列定义中除去集群名称。 更改 TORONTO 上的 INVENTQ ， 以使其无法从集群的其余部分进行访问:

   ```
   ALTER QLOCAL(INVENTQ) CLUSTER(' ')
   ```

2. 请检查队列是否不再可用。

   在完整存储库队列管理器 LONDON 或 NEWYORK 上，通过发出以下命令来检查队列管理器 TORONTO 是否不再主管该队列:

   ```
   DIS QCLUSTER (INVENTQ)
   ```

   如果 ALTER 命令已成功完成，那么结果中不会列出 TORONTO 。

3. 禁用队列。

   在 TORONTO 处禁用 INVENTQ 队列，以便无法向其写入更多消息:

   ```
   ALTER QLOCAL(INVENTQ) PUT(DISABLED)
   ```

   现在，使用 MQOO_BIND_ON_OPEN 传输到此队列的消息将转至死信队列。 您需要停止所有应用程序将消息显式地放入此队列管理器上的队列。

4. 监视队列，直到它为空。

   使用 DISPLAY QUEUE 命令并指定属性 IPPROCS， OPPROCS 和 CURDEPTH 来监视队列，或者在 IBM i 上使用 **WRKMQMSTS** 命令。 当输入和输出进程数以及队列的当前深度均为零时，队列为空。

5. 监视通道以确保没有不确定消息。

   要确保通道 INVENTORY.TORONTO 上没有不确定消息，请在其他每个队列管理器上监视名为 INVENTORY.TORONTO 的集群发送方通道。 从每个队列管理器发出指定 INDOUBT 参数的 DISPLAY CHSTATUS 命令:

   ```
   DISPLAY CHSTATUS(INVENTORY.TORONTO) INDOUBT
   ```

   如果存在任何不确定消息，那么必须先解决这些消息，然后再继续。 例如，您可以尝试发出 RESOLVE 通道命令或停止并重新启动通道。

6. 删除本地队列。

   当您确信没有更多消息要传递到 TORONTO 上的库存应用程序时，可以删除队列:

   ```
   DELETE QLOCAL(INVENTQ)
   ```

7. 现在，您可以从多伦多的系统中除去库存应用程序

   除去应用程序可避免重复，并在系统上节省空间。

结果

此任务设置的集群类似于先前任务设置的集群。 不同之处在于 INVENTQ 队列在队列管理器 TORONTO 上不再可用。

当您在步骤 1 中使队列退出服务时， TORONTO 队列管理器会向两个完整存储库队列管理器发送一条消息。 它将状态更改通知给他们。 完整存储库队列管理器将此信息传递给集群中请求更新与 INVENTQ 相关的信息的其他队列管理器。

当队列管理器将消息放入 INVENTQ 队列时，更新后的部分存储库指示 INVENTQ 队列仅在 NEWYORK 队列管理器上可用。 消息将发送到 NEWYORK 队列管理器。

下一步做什么

在此任务中，只有一个要除去的队列，只有一个要从中除去的集群。

假设有许多队列引用包含许多集群名称的名称列表。 例如， TORONTO 队列管理器可能不仅托管 INVENTQ，还托管 PAYROLLQ， SALESQ 和 PURCHASESQ。 TORONTO 使这些队列在所有相应的集群 INVENTORY，PAYROLL， SALES 和 PURCHASES 中可用。 在 TORONTO 队列管理器上定义集群名称的名称列表:

```
DEFINE NAMELIST(TOROLIST)
DESCR('List of clusters TORONTO is in')
NAMES(INVENTORY, PAYROLL, SALES, PURCHASES)
```

将名称列表添加到每个队列定义:

```
DEFINE QLOCAL(INVENTQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PAYROLLQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(SALESQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PURCHASESQ) CLUSNL(TOROLIST)
```

现在假设要从 SALES 集群中除去所有这些队列，因为 SALES 操作将由 PURCHASES 操作接管。 您只需更改 TOROLIST 名称列表，即可从其中除去 SALES 集群的名称。

如果要从名称列表中的某个集群中除去单个队列，请创建包含其余集群名称列表的名称列表。 然后更改队列定义以使用新的名称列表。 要从 INVENTORY 集群中除去 PAYROLLQ ，请执行以下操作:

1. 创建名称列表:

   ```
   DEFINE NAMELIST(TOROSHORTLIST)
   DESCR('List of clusters TORONTO is in other than INVENTORY')
   NAMES(PAYROLL, SALES, PURCHASES)
   ```

2. 更改 PAYROLLQ 队列定义:

   ```
   ALTER QLOCAL(PAYROLLQ) CLUSNL(TOROSHORTLIST)
   ```

从集群中除去队列管理器: 最佳实践

在队列管理器可以与集群中的至少一个完整存储库正常通信的情况下，从集群中除去队列管理器。

开始之前

对于至少有一个完整存储库可用且可由要除去的队列管理器联系的场景，此方法是最佳实践。 此方法涉及最少的手动干预，并允许队列管理器协商从集群的受控撤销。 如果要除去的队列管理器无法与完整存储库联系，请参阅 第 297 页的『从群集中删除队列管理器：替代方法』。

关于此任务

此示例任务从 INVENTORY 集群中除去队列管理器 LONDON 。 INVENTORY 集群按 第 254 页的『将队列管理器添加至集群』中所述进行设置，并按 第 293 页的『从队列管理器中除去集群队列』中所述进行修改。

从集群中除去队列管理器的过程比添加队列管理器的过程更复杂。

当队列管理器加入集群时，集群的现有成员不知道新的队列管理器，因此不与之进行交互。 必须在连接队列管理器上创建新的发送方和接收方通道，以便它可以连接到完整存储库。

从集群中除去队列管理器时，连接到队列管理器的应用程序可能正在使用诸如集群中其他位置托管的队列之类的对象。 此外，连接到集群中其他队列管理器的应用程序可能正在使用目标队列管理器上托管的对象。 由于这些应用程序，当前队列管理器可能会创建其他发送方通道，以建立与集群成员的通信，而不是与其用来加入集群的完整存储库的通信。 集群中的每个队列管理器都有一个用于描述其他集群成员的数据的高速缓存副本。 这可能包括要除去的内容。

过程

1. 在从集群中除去队列管理器之前，请确保队列管理器不再主管集群所需的资源:

- 如果队列管理器托管完整存储库，请从 第 264 页的『将完整存储库移至另一个队列管理器』完成步骤 1-6。 如果不将要除去的队列管理器的完整存储库功能移至其他队列管理器，那么仅需要完成步骤 5 和 6。
- 如果队列管理器托管集群队列，请从 第 293 页的『从队列管理器中除去集群队列』完成步骤 1-7。
- 如果队列管理器托管集群主题，请删除这些主题 (例如，使用 DELETE TOPIC 命令) ，或者将它们移至其他主机，如 第 358 页的『将群集主题定义移至不同的队列管理器』中所述。

  注: 如果从集群中除去队列管理器，并且该队列管理器仍托管集群主题，那么该队列管理器可能会继续尝试将发布交付到留在集群中的队列管理器，直到删除该主题为止。

2. 在队列管理器 LONDON 上变更手动定义的集群接收方通道以将其从集群中移除:

   ```
   ALTER CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) CLUSTER(' ')
   ```

3. 在队列管理器 LONDON 上变更手动定义的集群发送方通道以将其从集群中移除:

   ```
   ALTER CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) CLUSTER(' ')
   ```

   集群中的其他队列管理器了解到此队列管理器及其集群资源不再是集群的一部分。

4. 在队列管理器 LONDON 上监视集群传输队列，直到没有消息等待流向集群中的任何完整存储库。

   ```
   DISPLAY CHSTATUS(INVENTORY.PARIS) XQMSGSA
   ```

   如果消息保留在传输队列上，请先确定未将其发送到 PARIS 和 NEWYORK 完整存储库的原因，然后再继续。

结果

队列管理器 LONDON 不再是集群的一部分。 但是，它仍可以充当独立队列管理器。

下一步做什么

可以通过对集群的其余成员发出以下命令来确认这些更改的结果:

```
DISPLAY CLUSQMGR(LONDON)
```

队列管理器将继续显示，直到自动定义的集群发送方通道停止为止。 您可以等待此情况发生，或者通过发出以下命令继续监视活动实例:

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

当您确信没有更多消息传递到此队列管理器时，可以通过在集群的其余成员上发出以下命令来停止到 LONDON 的集群发送方通道:

```
STOP CHANNEL(INVENTORY.LONDON) STATUS(INACTIVE)
```

在整个集群中传播更改后，如果没有更多消息传递到此队列管理器，请停止并删除 LONDON 上的 CLUSRCVR 通道:

```
STOP CHANNEL(INVENTORY.LONDON)
DELETE CHANNEL(INVENTORY.LONDON)
```

如果手动定义的传输队列正在用于此通道，并且 CLCHNAME 模式与任何其他现有或计划的通道都不匹配，那么您可能想要删除该传输队列。 例如:

```
DELETE QLOCAL(PARIS.CUSTOM.XMITQ)
```

注: 如果是自动定义的传输队列或共享的 SYSTEM.CLUSTER.TRANSMIT.QUEUE 正在使用中，不需要执行此步骤。

稍后可以将除去的队列管理器重新添加到集群中，如 第 254 页的『将队列管理器添加至集群』中所述。 除去的队列管理器将继续对集群的其余成员进行最长 90 天的高速缓存知识。 如果您不想等到此高速缓存到期，那么可以按 第 298 页的『将队列管理器复原到其集群前状态』中所述强制除去此高速缓存。

相关任务

从集群中除去队列管理器 (使用 IBM MQ Explorer)

相关参考

ALTER CHANNEL (改变通道设置)

DISPLAY CHANNEL (显示通道定义)

DISPLAY CHSTATUS (显示通道状态)

DISPLAY CLUSQMGR (显示集群队列管理器的通道信息)

STOP CHANNEL (停止通道)

从群集中删除队列管理器：替代方法

在由于重大系统或配置问题导致队列管理器无法与群集中的任何完整存储库通信的情况下，从群集中移除队列管理器。

开始之前

这种从群集中手动删除队列管理器的替代方法会停止并删除将被删除的队列管理器链接到群集的所有群集通道，并强行将队列管理器从群集中删除。 这种方法适用于正在移除的队列管理器无法与任何完整存储库通信的情况。 这可能（例如）是因为队列管理器停止工作，或队列管理器与群集之间出现长期通信故障。 否则，请使用最常用的方法："第 295 页的『从集群中除去队列管理器: 最佳实践』 。

关于此任务

此示例任务从 INVENTORY 集群中除去队列管理器 LONDON 。 INVENTORY 集群按 第 254 页的『将队列管理器添加至集群』中所述进行设置，并按 第 293 页的『从队列管理器中除去集群队列』中所述进行修改。

从集群中除去队列管理器的过程比添加队列管理器的过程更复杂。

当队列管理器加入集群时，集群的现有成员不知道新的队列管理器，因此不与之进行交互。 必须在连接队列管理器上创建新的发送方和接收方通道，以便它可以连接到完整存储库。

从集群中除去队列管理器时，连接到队列管理器的应用程序可能正在使用诸如集群中其他位置托管的队列之类的对象。 此外，连接到集群中其他队列管理器的应用程序可能正在使用目标队列管理器上托管的对象。 由于这些应用程序，当前队列管理器可能会创建其他发送方通道，以建立与集群成员的通信，而不是与其用来加入集群的完整存储库的通信。 集群中的每个队列管理器都有一个用于描述其他集群成员的数据的高速缓存副本。 这可能包括要除去的内容。

在无法等待队列管理器从容离开群集的紧急情况下，该程序可能比较合适。

过程

1. 从群集中删除队列管理器之前，请确保队列管理器不再托管群集所需的资源：
   - 如果队列管理器托管完整存储库，请从 第 264 页的『将完整存储库移至另一个队列管理器』完成步骤 1-6。 如果不将要除去的队列管理器的完整存储库功能移至其他队列管理器，那么仅需要完成步骤 5 和 6。
   - 如果队列管理器托管集群队列，请从 第 293 页的『从队列管理器中除去集群队列』完成步骤 1-7。
   - 如果队列管理器托管集群主题，请删除这些主题 (例如，使用 DELETE TOPIC 命令) ，或者将它们移至其他主机，如 第 358 页的『将群集主题定义移至不同的队列管理器』中所述。

     注: 如果从集群中除去队列管理器，并且该队列管理器仍托管集群主题，那么该队列管理器可能会继续尝试将发布交付到留在集群中的队列管理器，直到删除该主题为止。
2. 停止用于与群集中其他队列管理器通信的所有通道。 使用 "MODE(FORCE) 停止队列管理器 "LONDON 上的 "CLUSRCVR 通道。 否则，您可能需要等待发送方队列管理器停止通道：

```
STOP CHANNEL(INVENTORY.LONDON) MODE(FORCE)
STOP CHANNEL(INVENTORY.TORONTO)
```

```
STOP CHANNEL(INVENTORY.PARIS)
STOP CHANNEL(INVENTORY.NEWYORK)
```

3. 监控队列管理器 "LONDON 上的通道状态，直到通道停止：

```
DISPLAY CHSTATUS(INVENTORY.LONDON)
DISPLAY CHSTATUS(INVENTORY.TORONTO)
DISPLAY CHSTATUS(INVENTORY.PARIS)
DISPLAY CHSTATUS(INVENTORY.NEWYORK)
```

   通道停止后，将不再向群集中的其他队列管理器发送或从其他队列管理器发送应用程序信息。
4. 删除队列管理器 "LONDON 上手动定义的集群通道：

```
DELETE CHANNEL(INVENTORY.NEWYORK)
DELETE CHANNEL(INVENTORY.TORONTO)
```

5. 集群中的其余队列管理器仍然知道被移除的队列管理器，并可能继续向其发送消息。 要清除剩余队列管理器中的知识，请在其中一个完整存储库上重置群集中已移除的队列管理器：

```
RESET CLUSTER(INVENTORY) ACTION(FORCEREMOVE) QMNAME(LONDON) QUEUES(YES)
```

   如果群集中可能有另一个队列管理器与已移除的队列管理器名称相同，请指定已移除队列管理器的 "**QMID**。

结果

队列管理器 LONDON 不再是集群的一部分。 但是，它仍可以充当独立队列管理器。

下一步做什么

可以通过对集群的其余成员发出以下命令来确认这些更改的结果：

```
DISPLAY CLUSQMGR(LONDON)
```

队列管理器将继续显示，直到自动定义的集群发送方通道停止为止。 您可以等待此情况发生，或者通过发出以下命令继续监视活动实例：

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

After the changes are propagated throughout the cluster, and no more messages are being delivered to this queue manager, delete the CLUSRCVR channel on LONDON:

```
DELETE CHANNEL(INVENTORY.LONDON)
```

稍后可以将除去的队列管理器重新添加到集群中，如 第 254 页的『将队列管理器添加至集群』中所述。 除去的队列管理器将继续对集群的其余成员进行最长 90 天的高速缓存知识。 如果您不想等到此高速缓存到期，那么可以按 第 298 页的『将队列管理器复原到其集群前状态』中所述强制除去此高速缓存。

相关参考

删除通道

显示通道

DISPLAY CHSTATUS

显示 clusqmgr

停止通道

Reset Cluster

将队列管理器复原到其集群前状态

从集群中除去队列管理器时，它将保留其余集群成员的知识。 此知识最终将到期并自动删除。 但是，如果您希望立即将其删除，那么可以使用本主题中的步骤。

开始之前

假定队列管理器已从集群中除去，并且不再在集群中执行任何工作。 例如，其队列不再接收来自集群的消息，并且没有应用程序在等待消息到达这些队列中。

关于此任务

从集群中除去队列管理器时，它最多会保留其余集群成员的知识 90 天。 这可能具有系统优势，尤其是在队列管理器快速重新加入集群时。 当此知识最终到期时，将自动将其删除。 但是，您可能希望手动删除此信息的原因有很多。 例如：

- 您可能希望确认已停止此队列管理器上先前使用了集群资源的每个应用程序。 在剩余集群成员的知识到期之前，任何此类应用程序都将继续写入传输队列。 删除集群知识后，当此类应用程序尝试使用集群资源时，系统会生成一条错误消息。
- 显示队列管理器的状态信息时，您可能不希望看到有关剩余集群成员的到期信息。

此任务使用 INVENTORY 集群作为示例。 已从 INVENTORY 集群中除去 LONDON 队列管理器，如 第 295 页的『从集群中除去队列管理器: 最佳实践』中所述。 要删除集群其余成员的知识，请在 LONDON 队列管理器上发出以下命令。

过程

1. 从此队列管理器中除去集群中其他队列管理器的所有内存：

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

2. 监视队列管理器，直到所有集群资源都已消失：

```
DISPLAY CLUSQMGR(*) CLUSTER(INVENTORY)
DISPLAY QCLUSTER(*) CLUSTER(INVENTORY)
DISPLAY TOPIC(*) CLUSTER(INVENTORY)
```

相关概念

集群

集群与分布式排队的比较

集群组件

维护队列管理器

从集群暂挂并恢复队列管理器以执行维护。

关于此任务

有时，您可能需要对属于集群的队列管理器执行维护。 例如，您可能需要对其队列中的数据进行备份，或者对软件应用修订。 如果队列管理器托管任何队列，那么必须暂挂其活动。 维护完成后，可以恢复其活动。

过程

1. 通过发出 SUSPEND QMGR **runmqsc** 命令暂挂队列管理器：

```
SUSPEND QMGR CLUSTER(SALES)
```

SUSPEND **runmqsc** 命令通知 SALES 集群中的队列管理器已暂挂此队列管理器。

SUSPEND QMGR 命令的目的只是建议其他队列管理器避免向此队列管理器发送消息 (如果可能)。 这并不意味着队列管理器已禁用。 此队列管理器必须处理的一些消息仍将发送到该队列管理器，例如，当此队列管理器是集群队列的唯一主机时。

当队列管理器处于暂挂状态时，工作负载管理例程将避免向其发送消息。 必须由该队列管理器处理的消息包括本地队列管理器发送的消息。

IBM MQ 使用工作负载均衡算法来确定适合的目标，而不是尽可能选择本地队列管理器。

a) 通过在 SUSPEND QMGR 命令上使用 FORCE 选项来强制暂挂队列管理器:

```
SUSPEND QMGR CLUSTER(SALES) MODE(FORCE)
```

MODE(FORCE) 强制停止来自集群中其他队列管理器的所有入站通道。 如果未指定 MODE(FORCE)，那么将应用缺省 MODE(QUIESCE) 。

2. 执行任何必需的维护任务。
3. 通过发出 RESUME QMGR **runmqsc** 命令来恢复队列管理器:

```
RESUME QMGR CLUSTER(SALES)
```

结果

RESUME **runmqsc** 命令通知完整存储库队列管理器再次可用。 完整存储库队列管理器将此信息传播给请求更新此队列管理器相关信息的其他队列管理器。

维护集群传输队列

尽一切努力保持集群传输队列可用。 它们对于集群的性能至关重要。 z/OS 在 z/OS 上，将集群传输队列的 INDXTYPE 设置为 CORRELID。

开始之前

- 确保集群传输队列未变满。
- 请注意不要发出 ALTER **runmqsc** 命令来将其设置为禁用或意外禁用。
- 确保集群传输队列存储在 z/OS 上的介质 (例如 z/OS 页集) 不会变满。

关于此任务

z/OS

以下过程仅适用于 z/OS。

过程

将集群传输队列的 INDXTYPE 设置为 CORRELID

刷新集群队列管理器

您可以使用 REFRESH CLUSTER 命令从本地存储库中除去自动定义的通道和自动定义的集群对象。 不会丢失任何消息。

开始之前

IBM 支持中心可能会要求您使用该命令。 请不要在未仔细考虑的情况下使用该命令。 例如，对于大型集群，使用 **REFRESH CLUSTER** 命令可能会在集群进行时对其产生破坏性影响，并且在集群对象自动向所有相关队列管理器发送状态更新之后的 27 天时间间隔再次发生此情况。 请参阅 集群: 使用 REFRESH CLUSTER 最佳实践。

关于此任务

队列管理器可以在集群中进行全新启动。 在正常情况下，您不需要使用 REFRESH CLUSTER 命令。

过程

从队列管理器发出 REFRESH CLUSTER **MQSC** 命令以从本地存储库中除去自动定义的集群队列管理器和队列对象。

此命令仅除去引用其他队列管理器的对象，而不除去与本地队列管理器相关的对象。 该命令还会除去自动定义的通道。 它将除去集群传输队列上没有消息且未连接到完整存储库队列管理器的通道。

结果

实际上， REFRESH CLUSTER 命令允许对队列管理器的完整存储库内容进行冷启动。 IBM MQ 确保不会从队列中丢失任何数据。

相关信息
集群：使用 REFRESH CLUSTER 最佳实践

恢复集群队列管理器

使用 REFRESH CLUSTER **runmqsc** 命令更新有关队列管理器的集群信息。 从时间点备份恢复队列管理器后，请遵循此过程。

开始之前
您已从时间点备份复原集群队列管理器。

关于此任务
要恢复集群中的队列管理器，请复原该队列管理器，然后使用 REFRESH CLUSTER **runmqsc** 命令更新集群信息。

注: 对于大型集群，使用 **REFRESH CLUSTER** 命令可能会对正在运行的集群造成干扰，此外每隔 27 天在集群对象向所有相关队列管理器自动发送状态更新时也可能有干扰。 请参阅在大型集群中刷新可能会影响集群的性能和可用性。

过程

针对队列管理器参与的所有集群，在复原的队列管理器上发出 REFRESH CLUSTER 命令。

下一步做什么

无需在任何其他队列管理器上发出 REFRESH CLUSTER 命令。

相关信息
集群：使用 REFRESH CLUSTER 最佳实践

配置集群通道以实现可用性

如果存在间歇性网络中断，请遵循良好的配置实践以保持集群通道平稳运行。

开始之前

集群使您无需定义通道，但仍需要对其进行维护。 用于集群中队列管理器之间通信的通道技术与用于分布式排队的通道技术相同。 要了解集群通道，您需要熟悉以下事项:

- 通道的运行方式
- 如何查找其状态
- 如何使用通道出口

关于此任务
您可能要特别考虑以下几点:

过程

配置集群通道时请考虑以下几点

- 选择集群发送方通道和集群接收方通道上的 HBINT 或 KAINT 的值，这些值不会给网络带来大量脉动信号或保持活动流的负担。 如果您的网络有时会变慢并引入此长度的延迟，那么小于约 10 秒的时间间隔会产生错误故障。
- 设置 BATCHHB 值以减少导致已发生错误的消息的窗口，因为该消息在发生故障的通道上处于不确定状态。 如果给予批处理更长时间进行填充，那么失败通道上的不确定批处理更有可能发生。 如果通道上的消息流量是零星的，并且消息之间的时间间隔很长，那么失败的批处理更有可能发生。
- 如果通道的集群发送方端发生故障，然后在脉动信号或保持活动检测到故障之前尝试重新启动，那么会出现问题。 如果通道的集群接收方端保持活动状态，那么将拒绝通道发送方重新启动。 为避免此故障，请安排在集群发送方通道尝试重新启动时终止并重新启动集群接收方通道。

  z/OS **打开 IBM MQ for z/OS**

  使用 **ALTER QMGR** 上的 **ADOPTMCA** 和 **ADOPTCHK** 参数来控制通道的集群接收方端处于活动状态的问题。

  Multi **打开多平台**

  使用 qm.ini 文件或 Windows 注册表中的 **AdoptNewMCA**， **AdoptNewMCATimeout** 和 **AdoptNewMCACheck** 属性控制通道剩余活动的集群接收方端的问题。

示例

有关如何在 IBM MQ for z/OS 和 IBM MQ for Multiplatforms 上实现这些设置的示例，请参阅 第 186 页的『建议设置』 。

检查是否已完成用于分布式网络的异步命令

在分布式网络中使用时，许多命令是异步的。 根据命令，以及发出时的网络状态，可能需要大量时间才能完成。 队列管理器在完成时不会发出消息，因此您需要其他方法来检查命令是否已完成。

关于此任务

您对集群进行的几乎任何配置更改都可能以异步方式完成。 这是因为在集群中运行的内部管理和更新周期。 对于发布/预订层次结构，任何影响预订的配置更改都可能以异步方式完成。 这在命令的名称中并不总是很明显。

以下 MQSC 命令可能都以异步方式完成。 这些命令中的每个命令都具有等效的 PCF，并且大多数命令也可从 IBM MQ Explorer 中获取。 在没有工作负载的小型网络上运行时，这些命令通常在几秒钟内完成。 但是，对于更大和更繁忙的网络，情况并非如此。 此外， **REFRESH CLUSTER** 命令可能需要更长时间，尤其是在同时在多个队列管理器上发出时。

要确信这些命令已完成，请检查远程队列管理器上是否存在期望的对象。

过程

- ALTER QMGR

  对于 ALTER QMGR PARENT 命令，请使用 DISPLAY PUBSUB TYPE(PARENT) ALL 来跟踪请求的父关系的状态。

  对于 ALTER QMGR REPOS 和 ALTER QMGR REPOSNL 命令，请使用 DISPLAY CLUSQMGR QMTYPE 来确认完成。
- DEFINE CHANNEL， ALTER CHANNEL 和 DELETE CHANNEL

  对于表 ALTER CHANNEL parameters 中列出的所有参数，请使用 DISPLAY CLUSQMGR 命令来监视何时已将更改传播到集群。
- DEFINE NAMELIST， ALTER NAMELIST 和 DELETE NAMELIST。

  如果在 **QMgr** 对象的 **CLUSNL** 属性上使用 **NAMELIST**，那么队列或集群通道可能会影响该对象。 根据需要监视受影响的对象。

  对 SYSTEM.QPUBSUB.QUEUE.NAMELIST 的更改可能会影响在发布/预订层次结构中创建或取消代理预订。 使用 DISPLAY SUB SUBTYPE(PROXY) 命令来监视此情况。

- DEFINE 队列， ALTER 队列和 DELETE 队列。

  对于表 可由 DISPLAY QUEUE 命令返回的参数中列出的所有参数，请使用 DISPLAY QCLUSTER 命令来监视何时已将更改传播到集群。

- DEFINE SUB 和 DELETE SUB

  在主题字符串上定义第一个预订时，可以在发布/预订层次结构或发布/预订集群中创建代理预订。 同样，当您删除主题字符串上的最后一个预订时，可能会取消发布/预订层次结构或发布/预订集群中的代理预订。

  要检查定义或删除预订的命令是否已完成，请检查分布式网络中的其他队列管理器上是否存在期望的代理预订。 如果要在集群中使用 直接路由 ，请检查集群中的其他部分存储库上是否存在期望的代理预订。 如果要在集群中使用 主题主机路由 ，请检查匹配的主题主机上是否存在期望的代理预订。 使用以下 MQSC 命令:

  ```
  DISPLAY SUB(*) SUBTYPE(PROXY)
  ```

  在集群或层次结构中发出以下等效预订和取消预订 MQI 调用时，请使用相同的检查:

  – 使用 MQSUB 进行预订。
  – 通过将 MQCLOSE 与 MQCO_REMOVE_SUB 配合使用来取消预订。

- DEFINE TOPIC， ALTER TOPIC 和 DELETE TOPIC

  要检查定义，更改或删除集群主题的命令是否已完成，请在集群中的其他部分存储库中显示该主题 (如果您正在使用 直接路由) 或在其他主题主机上 (如果您正在使用 主题主机路由)。

  对于表 可由 DISPLAY TOPIC 命令返回的参数中列出的所有参数，请使用 DISPLAY TCLUSTER 命令来监视何时已将更改传播到集群。

  注:

  – **CLUSTER** 参数可能会影响在发布/预订集群中创建或取消代理预订。
  – **PROXYSUB** 和 **SUBSCOPE** 参数可能会影响在发布/预订层次结构或发布/预订集群中创建或取消代理预订。
  – 使用 DISPLAY SUB SUBTYPE(PROXYSUB) 命令来监视此情况。

- 刷新集群

  如果您正在运行 **REFRESH CLUSTER** 命令，请轮询集群命令队列深度。 等待它到达零，并保持为零，然后再查找对象。

  1. 使用以下 MQSC 命令来检查集群命令队列深度是否为零。

     ```
     DISPLAY QL(SYSTEM.CLUSTER.COMMAND.QUEUE) CURDEPTH
     ```

  2. 重复检查，直到队列深度达到零，并在后续检查中保持为零。

  REFRESH CLUSTER 命令会除去并重新创建对象，并且在大型配置中可能需要大量时间才能完成。 请参阅针对发布/预订集群的 REFRESH CLUSTER 注意事项。

- REFRESH QMGR 类型 (PROXYSUB)

  要检查 **REFRESH QMGR TYPE(PROXYSUB)** 命令是否已完成，请检查是否已在分布式网络中的其他队列管理器上更正代理预订。 如果要在集群中使用 直接路由 ，请检查是否已在集群中的其他部分存储库上更正代理预订。 如果要在集群中使用 主题主机路由 ，请检查是否已在匹配的主题主机上更正期望的代理预订。 使用以下 MQSC 命令:

  ```
  DISPLAY SUB(*) SUBTYPE(PROXYSUB)
  ```

- Reset Cluster

  要检查 **RESET CLUSTER** 命令是否已完成，请使用 DISPLAY CLUSQMGR。

- RESET QMGR 类型 (PUBSUB)

  要检查 **RESET QMGR** 命令是否已完成，请使用 DISPLAY PUBSUB TYPE(PARENT|CHILD)。

注: RESET QMGR 命令可能会导致取消发布/预订层次结构或发布/预订集群中的代理预订。 使用 DISPLAY SUB SUBTYPE(PROXYSUB) 命令来监视此情况。

- 您可能还希望监视其他系统队列，这些队列在命令完成时趋向于队列深度为零。

  例如，您可能想要监视 SYSTEM.INTER.QMGR.CONTROL 队列和 SYSTEM.INTER.QMGR.FANREQ 队列。 请参阅 监视集群中的代理预订流量 和 均衡发布/预订网络中的生产者和使用者。

下一步做什么

如果这些检查未确认异步命令已完成，那么可能发生了错误。 要进行调查，请先检查发出该命令的队列管理器的日志，然后 (对于集群) 检查集群完整存储库日志。

相关概念

z/OS z/OS 上 CLUSTER 命令的异步行为

将消息路由到集群或从集群路由消息

使用队列别名，队列管理器别名和远程队列定义将集群连接到外部队列管理器和其他集群。

有关将消息路由到集群和从集群路由消息的详细信息，请参阅以下子主题:

相关概念

集群

集群与分布式排队的比较

集群的组件

第 314 页的『队列管理器别名和集群』
在将消息发送到集群或从集群发送消息时，使用队列管理器别名来隐藏队列管理器的名称，以及对发送到集群的消息进行工作负载均衡。

第 318 页的『队列别名和集群』
使用队列别名来隐藏集群队列的名称，对队列进行集群，采用不同的属性或采用不同的访问控制。

第 317 页的『应答队列别名和集群』
应答队列别名定义用于指定应答信息的备用名称。 应答队列别名定义可以与集群一起使用，就像在分布式排队环境中一样。

相关任务

第 235 页的『配置队列管理器集群』
集群提供了一种用于以简化初始配置和持续管理的方式互连队列管理器的机制。 您可以定义集群组件，以及创建和管理集群。

第 244 页的『设置新集群』
遵循以下指示信息来设置示例集群。 单独的指示信息描述在 TCP/IP 上设置集群， LU 6.2 以及使用单个传输队列或多个传输队列。 通过将消息从一个队列管理器发送到另一个队列管理器来测试集群工作。

配置集群的请求/应答

从集群外部的队列管理器配置请求/应答消息路径。 通过使用网关队列管理器作为集群之间的通信路径，隐藏集群的内部详细信息。

开始之前

第 305 页的图 55 显示了名为 QM3 的队列管理器，该队列管理器位于名为 DEMO 的集群外部。 QM3 可以是 IBM MQ 产品上不支持集群的队列管理器。 QM3 托管名为 Q3 的队列，定义如下:

```
DEFINE QLOCAL(Q3)
```

集群中有两个名为 QM1 和 QM2 的队列管理器。 QM2 托管名为 Q2 的集群队列，其定义如下:

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO)
```

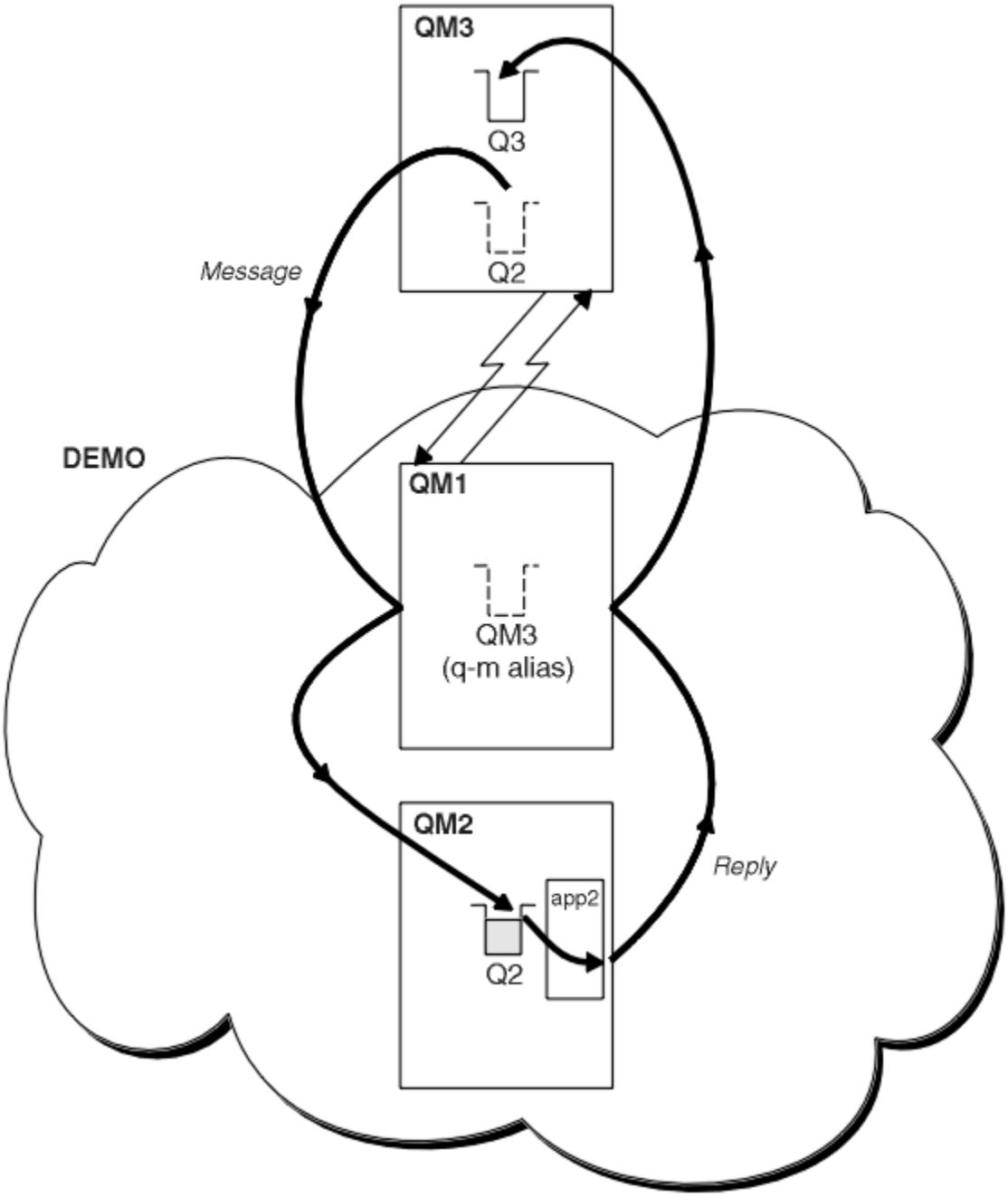


图 *55:* 从集群外部的队列管理器放入

关于此任务

遵循过程中的建议来设置请求和应答消息的路径。

过程

1. 将请求消息发送到集群。

   请考虑集群外部的队列管理器如何将消息放入集群内部的队列 Q2 QM2 中。 集群外部的队列管理器必须对其将消息放入的集群中的每个队列具有 QREMOTE 定义。

   a) 为 QM3 上的 Q2 定义远程队列。

   ```
   DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(QM2) XMITQ(QM1)
   ```

由于 QM3 不是集群的一部分，因此它必须使用分布式排队方法进行通信。 因此，它还必须具有发送方通道和到 QM1 的传输队列。 QM1 需要相应的接收方通道。 第 305 页的图 55 中未显式显示通道和传输队列。

在该示例中，位于 QM3 的应用程序发出 MQPUT 调用以将消息放入 Q2。 QREMOTE 定义导致使用从 QM1 传输队列获取消息的发送方通道将消息路由到 QM2 处的 Q2 。

2. 从集群接收应答消息。

使用队列管理器别名为集群外部的队列管理器的应答创建返回路径。 网关 QM1 为集群 QM3 外部的队列管理器发布队列管理器别名。 它通过将集群属性添加到 QM3 的队列管理器别名定义，将 QM3 广告给集群中的队列管理器。 队列管理器别名定义类似于远程队列定义，但带有空白 RNAME。

a) 为 QM1 上的 QM3 定义队列管理器别名。

```
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

我们必须考虑用于将回复从 QM1 转发到 QM3 的传输队列的名称选项。 QREMOTE 定义中的隐式内容 (通过省略 XMITQ 属性) 是传输队列的名称 QM3。 但是， QM3 与我们期望使用队列管理器别名向集群的其余部分发布的名称相同。 IBM MQ 不允许您为传输队列和队列管理器别名指定相同的名称。 一种解决方案是创建传输队列，以将具有不同名称的消息转发到 QM3 到队列管理器别名。

b) 在 QREMOTE 定义中提供传输队列名称。

```
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO) XMITQ(QM3.XMIT)
```

新的队列管理器别名将名为 QM3.XMIT 的新传输队列与 QM3 队列管理器别名进行耦合。 这是一个简单而正确的解决方案，但并不完全令人满意。 它已破坏为传输队列提供与目标队列管理器相同的名称的传输队列的命名约定。 是否有任何备用解决方案保留传输队列命名约定?

产生此问题的原因是请求者缺省将 QM3 作为从 QM3 发送的请求消息中的应答队列管理器名称传递。 QM2 上的服务器使用 QM3 应答队列管理器名称在其应答中寻址 QM3 。 解决方案需要 QM1 将 QM3 通告为队列管理器别名，以向其返回应答消息，并阻止 QM1 将 QM3 用作传输队列的名称。

QM3 上的应用程序需要将应答队列管理器别名传递给 QM1 以获取应答消息，而不是缺省提供 QM3 作为应答队列管理器名称。 网关队列管理器 QM1 会将队列管理器别名通告给 QM3 而不是 QM3 本身的应答，从而避免与传输队列的名称发生冲突。

c) 为 QM1 上的 QM3 定义队列管理器别名。

```
DEFINE QREMOTE(QM3.ALIAS) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

需要对配置命令进行两项更改。

i) 位于 QM1 的 QREMOTE 现在将我们的队列管理器别名 QM3.ALIAS 发布到集群的其余部分，并将其耦合到实际队列管理器的名称 QM3。 QM3 又是用于将应答队列发送回 QM3 的传输队列的名称

ii) 客户机应用程序在构造请求消息时必须提供 QM3.ALIAS 作为应答队列管理器的名称。 您可以通过两种方法之一向客户机应用程序提供 QM3.ALIAS 。

- 由 MQMD 中的 MQPUT 构造的应答队列管理器名称字段中的代码 QM3.ALIAS 。 如果您正在使用动态队列进行回复，那么必须以这种方式执行此操作。
- 提供应答队列名称时，请使用应答队列别名 Q3.ALIAS，而不是应答队列。

```
DEFINE QREMOTE(Q3.ALIAS) RNAME(Q3) RQMNAME(QM3.ALIAS)
```

下一步做什么

注: 无法演示如何将应答队列别名与 **AMQSREQ0** 配合使用。 它使用参数 3 中提供的队列名称或缺省 SYSTEM.SAMPLE.REPLY 模型队列打开应答队列。 您需要修改提供另一个包含应答队列别名的参数的样本，以命名 MQPUT 的应答队列管理器别名。

相关概念

队列管理器别名和集群

在将消息发送到集群或从集群发送消息时，使用队列管理器别名来隐藏队列管理器的名称，以及对发送到集群的消息进行工作负载均衡。

应答队列别名和集群
应答队列别名定义用于指定应答信息的备用名称。 应答队列别名定义可以与集群一起使用，就像在分布式排队环境中一样。

队列别名和集群
使用队列别名来隐藏集群队列的名称，对队列进行集群，采用不同的属性或采用不同的访问控制。

相关任务

从集群配置请求/应答
配置从集群到集群外部的队列管理器的请求/应答消息路径。 隐藏集群中的队列管理器如何使用网关队列管理器在集群外部进行通信的详细信息。

从集群外部配置工作负载均衡
配置从集群外部的队列管理器到集群队列的任何副本的消息路径。 结果是将来自集群外部的请求与集群队列的每个实例进行工作负载均衡。

配置集群之间的消息路径
使用网关队列管理器将集群连接在一起。 通过在网关队列管理器上定义集群队列或集群队列管理器别名，使队列或队列管理器对所有集群可见。

第 307 页的『隐藏集群目标队列管理器的名称』
将消息路由到集群中任何队列管理器上定义的集群队列，而不指定队列管理器。

隐藏集群目标队列管理器的名称
将消息路由到集群中任何队列管理器上定义的集群队列，而不指定队列管理器。

开始之前

- 避免将集群内的队列管理器的名称透露给集群外的队列管理器。
  - 解析对在集群中托管队列的队列管理器的引用会除去执行工作负载均衡的灵活性。
  - 这也使您难以更改在集群中托管队列的队列管理器。
  - 替代方法是将 RQMNAME 替换为集群管理员提供的队列管理器别名。
  - 第 307 页的『隐藏集群目标队列管理器的名称』 描述了使用队列管理器别名将集群外部的队列管理器与集群内部的队列管理器管理分离。
- 但是，命名传输队列的建议方法是向它们提供目标队列管理器的名称。 传输队列的名称显示集群中队列管理器的名称。 您必须选择要遵循的规则。 您可以选择使用队列管理器名称或集群名称来命名传输队列:

  使用网关队列管理器名称命名传输队列
  向集群外部的队列管理器披露网关队列管理器名称是隐藏集群队列管理器名称的规则的合理例外。

  使用集群的名称命名传输队列
  如果未遵循使用目标队列管理器的名称命名传输队列的约定，请使用集群名称。

关于此任务

修改任务 第 304 页的『配置集群的请求/应答』，以在集群中隐藏目标队列管理器的名称。

过程

在此示例中，请参阅 第 308 页的图 56，在名为 DEMO 的网关队列管理器 QM1 上定义队列管理器别名:

```
DEFINE QREMOTE(DEMO) RNAME(' ') RQMNAME(' ')
```

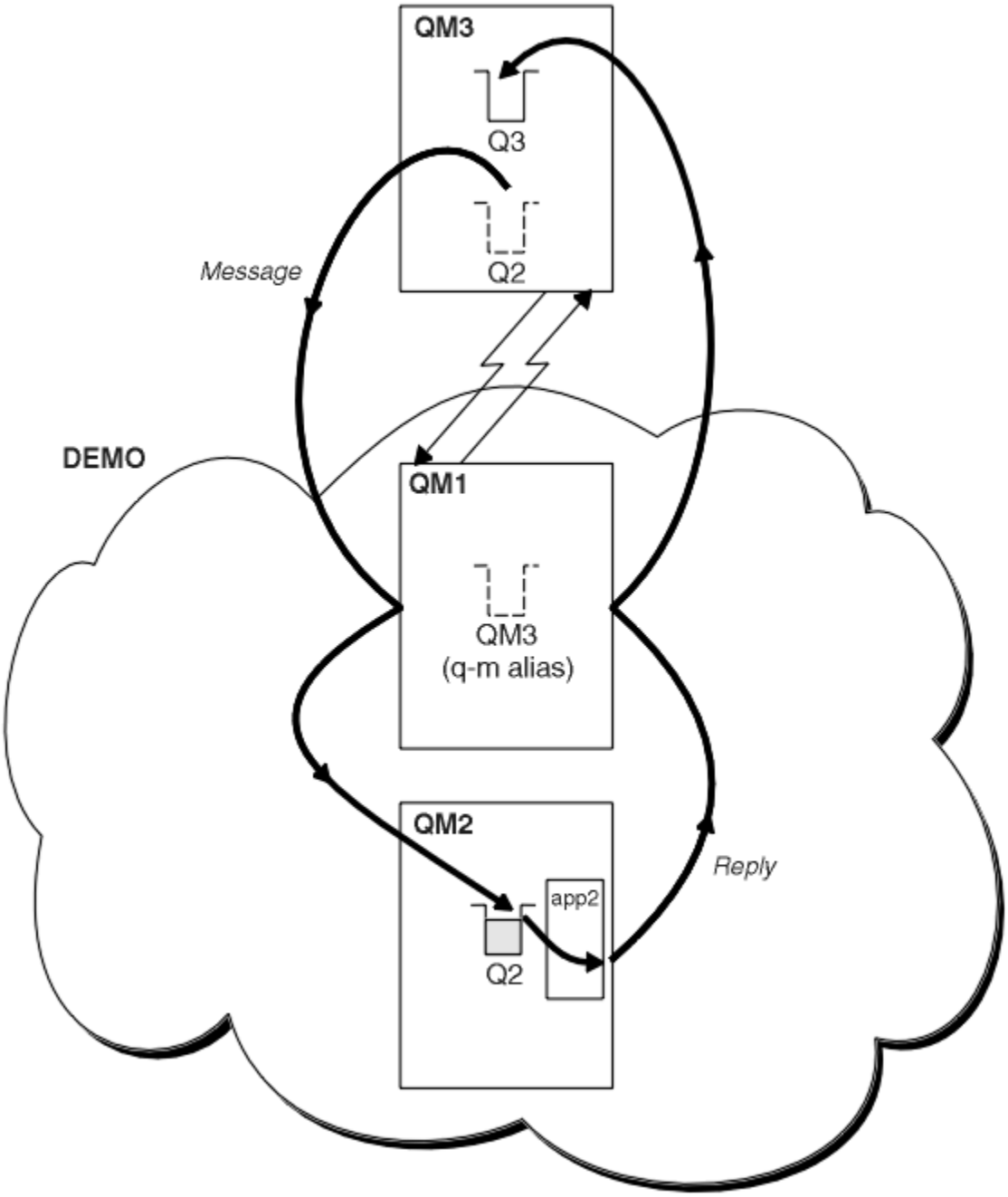


图 *56:* 从集群外部的队列管理器放入

QM1 上的 QREMOTE 定义使网关队列管理器知道队列管理器别名 DEMO 。 QM3 集群外部的队列管理器可以使用队列管理器别名 DEMO 将消息发送到 DEMO 上的集群队列，而不必使用实际队列管理器名称。

如果采用使用集群名称来命名连接到集群的传输队列的约定，那么 Q2 的远程队列定义将变为:

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(DEMO) XMIT(DEMO)
```

结果

在 DEMO 上发往 Q2 的消息将放置在 DEMO 传输队列上。 从传输队列中，它们由发送方通道传输到网关队列管理器 QM1。 网关队列管理器将消息路由到托管集群队列 Q2 的集群中的任何队列管理器。

从集群配置请求/应答

配置从集群到集群外部的队列管理器的请求/应答消息路径。 隐藏集群中的队列管理器如何使用网关队列管理器在集群外部进行通信的详细信息。

开始之前

第 309 页的图 57 显示了集群 DEMO 内的队列管理器 QM2。 它将请求发送到集群外部的队列管理器上托管的队列 Q3。 回复将返回到集群内的 QM2 处的 Q2 。

要与集群外部的队列管理器进行通信，集群内的一个或多个队列管理器充当网关。 网关队列管理器具有到集群外部的队列管理器的通信路径。 在此示例中， QM1 是网关。

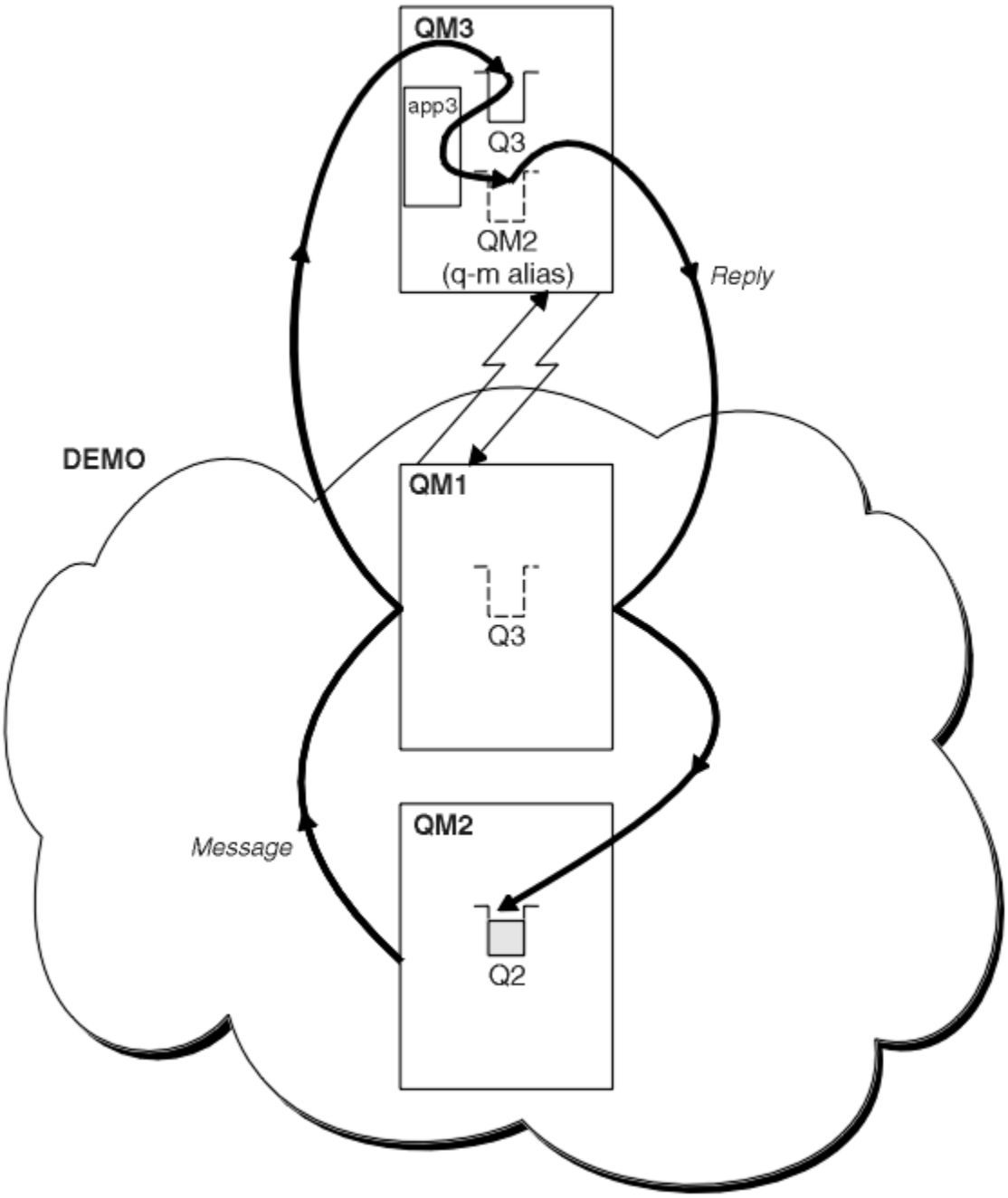


图 *57:* 放入集群外部的队列管理器

关于此任务

遵循指示信息来设置请求和应答消息的路径

过程

1. 从集群发送请求消息。

   请考虑集群内的队列管理器 QM2 如何将消息放入集群外的队列 Q3 的 QM3 中。

   a) 在 QM1 上创建将远程队列 Q3 通告到集群的 QREMOTE 定义

   ```
   DEFINE QREMOTE(Q3) RNAME(Q3) RQMNAME(QM3) CLUSTER(DEMO)
   ```

   它还具有到集群外部的队列管理器的发送方通道和传输队列。 QM3 具有相应的接收方通道。 这些通道未显示在 第 309 页的图 57 中。

   QM2 上的应用程序发出 MQPUT 调用，指定要将应答发送到的目标队列和队列。 目标队列为 Q3 ，应答队列为 Q2。

   此消息将发送到 QM1，后者使用其远程队列定义将队列名称解析为 QM3 处的 Q3 。
2. 从集群外部的队列管理器接收应答消息。

   集群外部的队列管理器必须具有集群中要向其发送消息的每个队列管理器的队列管理器别名。 队列管理器别名还必须指定到网关队列管理器的传输队列的名称。 在此示例中， QM3 需要 QM2 的队列管理器别名定义:

   a) 在 QM3 上创建队列管理器别名 QM2

   ```
   DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) XMITQ(QM1)
   ```

   QM3 还需要发送方通道和到 QM1 的传输队列，而 QM1 需要相应的接收方通道。

   然后， QM3 上的应用程序 **app3** 可以通过发出 MQPUT 调用并指定队列名称 Q2 和队列管理器名称 QM2 将应答发送到 QM2。

下一步做什么

您可以定义集群中的多个路由。

相关概念

队列管理器别名和集群
在将消息发送到集群或从集群发送消息时，使用队列管理器别名来隐藏队列管理器的名称，以及对发送到集群的消息进行工作负载均衡。

应答队列别名和集群
应答队列别名定义用于指定应答信息的备用名称。 应答队列别名定义可以与集群一起使用，就像在分布式排队环境中一样。

队列别名和集群
使用队列别名来隐藏集群队列的名称，对队列进行集群，采用不同的属性或采用不同的访问控制。

相关任务

配置集群的请求/应答
从集群外部的队列管理器配置请求/应答消息路径。 通过使用网关队列管理器作为集群之间的通信路径，隐藏集群的内部详细信息。

从集群外部配置工作负载均衡
配置从集群外部的队列管理器到集群队列的任何副本的消息路径。 结果是将来自集群外部的请求与集群队列的每个实例进行工作负载均衡。

配置集群之间的消息路径
使用网关队列管理器将集群连接在一起。 通过在网关队列管理器上定义集群队列或集群队列管理器别名，使队列或队列管理器对所有集群可见。

从集群外部配置工作负载均衡

配置从集群外部的队列管理器到集群队列的任何副本的消息路径。 结果是将来自集群外部的请求与集群队列的每个实例进行工作负载均衡。

开始之前

配置示例，如 第 304 页的『配置集群的请求/应答』中的 第 305 页的图 55 所示。

关于此任务

在此场景中，集群外部的队列管理器 (第 312 页的图 58 中的 QM3) 将请求发送到队列 Q2。 Q2 托管在集群 DEMO 中的两个队列管理器 QM2 和 QM4 上。 这两个队列管理器都使用缺省绑定选项 NOTFIXED 进行配置，以便使用工作负载均衡。 来自 QM3(集群外部的队列管理器) 的请求将通过 QM1 发送到 Q2 的任一实例。

QM3 不是集群的一部分，而是使用分布式排队方法进行通信。 它必须具有到 QM1 的发送方通道和传输队列。 QM1 需要相应的接收方通道。 第 312 页的图 58 中未显式显示通道和传输队列。

该过程扩展了 第 304 页的『配置集群的请求/应答』中的 第 305 页的图 55 中的示例。

过程

1. 为 QM3 上的 Q2 创建 QREMOTE 定义。

   ```
   DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(Q3) XMITQ(QM1)
   ```

   为 QM3 将消息放入的集群中的每个队列创建 QREMOTE 定义。
2. 在 QM1 上创建队列管理器别名 Q3 。

   ```
   DEFINE QREMOTE(Q3) RNAME(' ') RQMNAME(' ')
   ```

   Q3 不是实际队列管理器名称。 它是集群中队列管理器别名定义的名称，将队列管理器别名 Q3 等同于空白， ' '
3. 在每个 QM2 和 QM4 上定义名为 Q2 的本地队列。

   ```
   DEFINE QLOCAL(Q2) CLUSTER(DEMO) DEFBIND(NOTFIXED)
   ```
4. QM1 是网关队列管理器，没有特殊定义。

结果

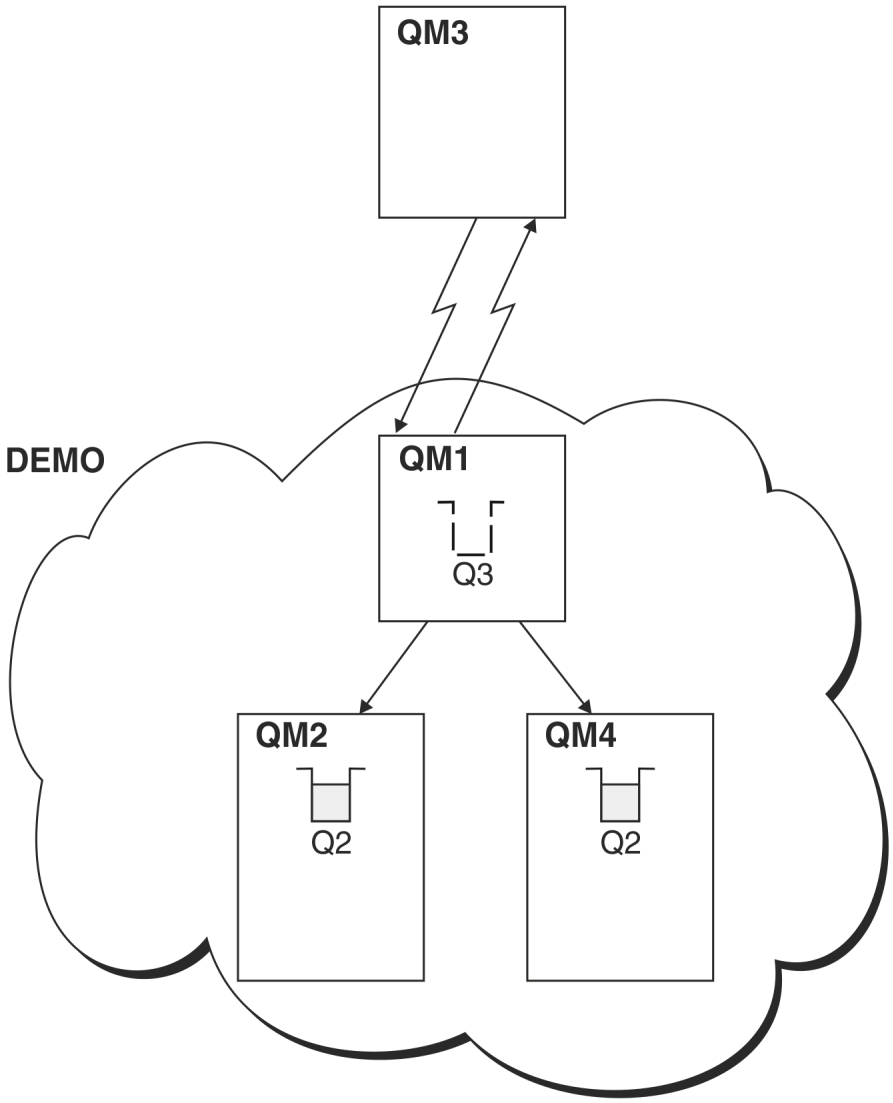


图 *58:* 从集群外部的队列管理器放入

当 QM3 上的应用程序发出 MQPUT 调用以将消息放入 Q2 时， QM3 上的 QREMOTE 定义会导致消息通过网关队列管理器 QM1 进行路由。 当 QM1 接收到该消息时，它知道该消息仍用于名为 Q2 的队列并执行名称解析。 QM1 检查其本地定义，但找不到 Q2 的任何定义。 然后， QM1 会检查其集群配置，并发现它知道集群 DEMO 中有两个 Q2 实例。 QM1 现在可以使用工作负载均衡在驻留在 QM2 和 QM4 上的 Q2 实例之间分发消息。

相关概念

队列管理器别名和集群
在将消息发送到集群或从集群发送消息时，使用队列管理器别名来隐藏队列管理器的名称，以及对发送到集群的消息进行工作负载均衡。

应答队列别名和集群
应答队列别名定义用于指定应答信息的备用名称。 应答队列别名定义可以与集群一起使用，就像在分布式排队环境中一样。

队列别名和集群
使用队列别名来隐藏集群队列的名称，对队列进行集群，采用不同的属性或采用不同的访问控制。

相关任务

配置集群的请求/应答

从集群外部的队列管理器配置请求/应答消息路径。 通过使用网关队列管理器作为集群之间的通信路径，隐藏集群的内部详细信息。

从集群配置请求/应答
配置从集群到集群外部的队列管理器的请求/应答消息路径。 隐藏集群中的队列管理器如何使用网关队列管理器在集群外部进行通信的详细信息。

配置集群之间的消息路径
使用网关队列管理器将集群连接在一起。 通过在网关队列管理器上定义集群队列或集群队列管理器别名，使队列或队列管理器对所有集群可见。

相关信息
队列名称解析
名称解析

配置集群之间的消息路径

使用网关队列管理器将集群连接在一起。 通过在网关队列管理器上定义集群队列或集群队列管理器别名，使队列或队列管理器对所有集群可见。

关于此任务

可以有许多较小的集群，而不是将所有队列管理器分组到一个大型集群中。 每个集群都有一个或多个队列管理器充当网桥。 这样做的优点是可以限制集群中队列和队列管理器名称的可视性。 请参阅 重叠集群。 使用别名来更改队列和队列管理器的名称，以避免名称冲突或遵守本地命名约定。

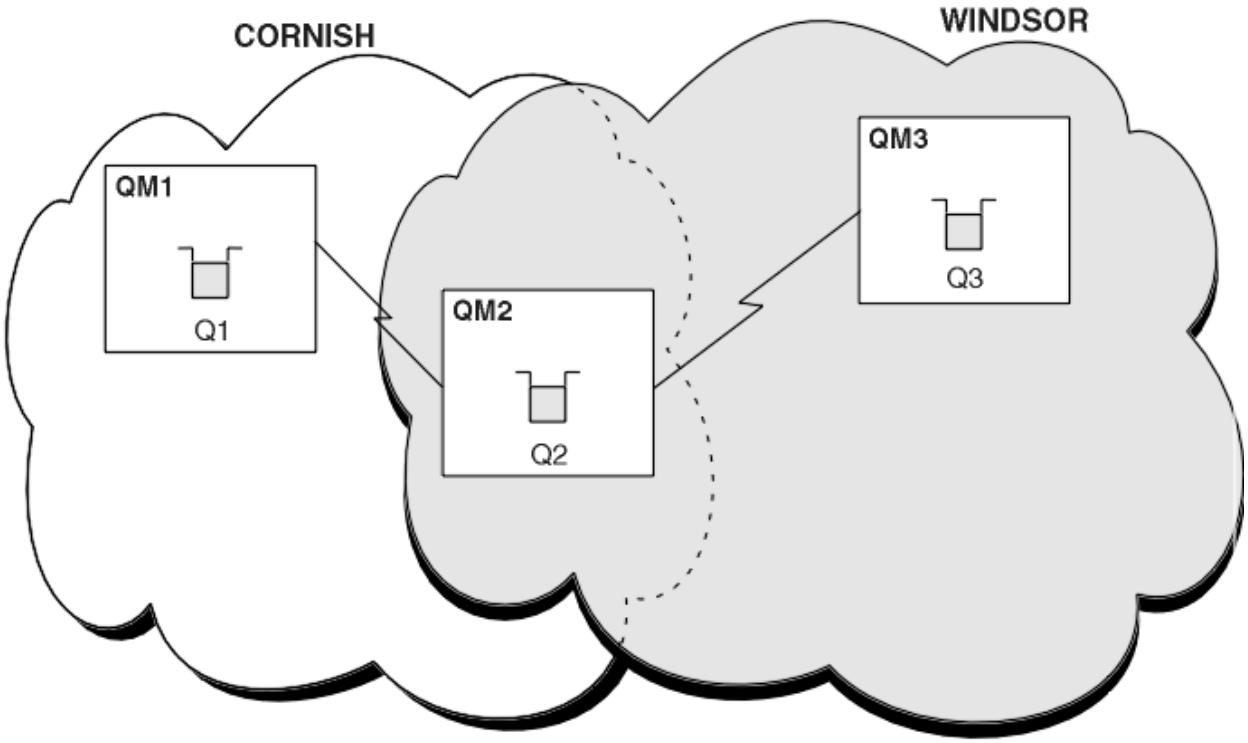


图 *59:* 跨集群的桥接

第 313 页的图 59 显示了两个集群，它们之间有一个网桥。 可能有不止一座桥。

使用以下过程配置集群:

过程

1. 在 QM1 上定义集群队列 Q1 。

```
DEFINE QLOCAL(Q1) CLUSTER(CORNISH)
```

2. 在 QM3 上定义集群队列 Q3 。

```
DEFINE QLOCAL(Q3) CLUSTER(WINDSOR)
```

3. 在 QM2 上创建名为 CORNISHWINDSOR 的名称列表，其中包含两个集群的名称。

```
DEFINE NAMELIST(CORNISHWINDSOR) DESCR('CornishWindsor namelist')
NAMES(CORNISH, WINDSOR)
```

4. 在 QM2 上定义集群队列 Q2

```
DEFINE QLOCAL(Q2) CLUSNL(CORNISHWINDSOR)
```

下一步做什么

QM2 是两个集群的成员，并且是它们之间的网桥。 对于要在网桥中显示的每个队列，需要在网桥上使用 QALIAS 定义。 例如，在 第 313 页的图 59 中，在 QM2 上，您需要:

```
DEFINE QALIAS(MYQ3) TARGET(Q3) CLUSTER(CORNISH) DEFBIND(NOTFIXED)
```

通过使用队列别名，连接到 CORNISH 中队列管理器的应用程序 (例如 QM1) 可以将消息放入 Q3。 它将 Q3 称为 MYQ3。 消息将路由到 QM3 处的 Q3 。

打开队列时，需要将 DEFBIND 设置为 NOTFIXED 或 QDEF。 如果将 DEFBIND 保留为缺省值 OPEN，那么队列管理器会将别名定义解析为托管该别名的网桥队列管理器。 网桥不会转发消息。

对于要使其可视的每个队列管理器，您需要一个队列管理器别名定义。 例如，在 QM2 上，您需要:

```
DEFINE QREMOTE(QM1) RNAME(' ') RQMNAME(QM1) CLUSTER(WINDSOR)
```

通过在 MQOPEN 调用上显式命名 QM1 ，连接到 WINDSOR 中任何队列管理器的应用程序 (例如 QM3) 可以将消息放入 QM1 上的任何队列。

相关概念

队列管理器别名和集群
在将消息发送到集群或从集群发送消息时，使用队列管理器别名来隐藏队列管理器的名称，以及对发送到集群的消息进行工作负载均衡。

应答队列别名和集群
应答队列别名定义用于指定应答信息的备用名称。 应答队列别名定义可以与集群一起使用，就像在分布式排队环境中一样。

队列别名和集群
使用队列别名来隐藏集群队列的名称，对队列进行集群，采用不同的属性或采用不同的访问控制。

相关任务

配置集群的请求/应答
从集群外部的队列管理器配置请求/应答消息路径。 通过使用网关队列管理器作为集群之间的通信路径，隐藏集群的内部详细信息。

从集群配置请求/应答
配置从集群到集群外部的队列管理器的请求/应答消息路径。 隐藏集群中的队列管理器如何使用网关队列管理器在集群外部进行通信的详细信息。

从集群外部配置工作负载均衡
配置从集群外部的队列管理器到集群队列的任何副本的消息路径。 结果是将来自集群外部的请求与集群队列的每个实例进行工作负载均衡。

队列管理器别名和集群

在将消息发送到集群或从集群发送消息时，使用队列管理器别名来隐藏队列管理器的名称，以及对发送到集群的消息进行工作负载均衡。

使用带有空白 RNAME 的远程队列定义创建的队列管理器别名有五种用途:

发送消息时重新映射队列管理器名称

可以使用队列管理器别名将 MQOPEN 调用中指定的队列管理器名称重新映射到另一个队列管理器。 它可以是集群队列管理器。 例如，队列管理器可能具有队列管理器别名定义:

```
DEFINE QREMOTE(YORK) RNAME(' ') RQMNAME(CLUSQM)
```

YORK 可用作名为 CLUSQM 的队列管理器的别名。 当执行此定义的队列管理器上的应用程序将消息放入队列管理器 YORK 时，本地队列管理器会将名称解析为 CLUSQM。 如果本地队列管理器未被称为 CLUSQM，那么它会将消息放在要移动到 CLUSQM 的集群传输队列上。 它还会将传输头更改为 CLUSQM 而不是 YORK。

注: 该定义仅适用于生成该定义的队列管理器。 要将别名发布到整个集群，需要将 CLUSTER 属性添加到远程队列定义。 然后，将来自目标为 YORK 的其他队列管理器的消息发送到 CLUSQM。

发送消息时改变或指定传输队列

别名判别可用于将集群连接到非集群系统。 例如，集群 ITALY 中的队列管理器可以与名为 PALERMO 的队列管理器通信，该队列管理器位于集群外部。 要进行通信，集群中的其中一个队列管理器必须充当网关。 从网关队列管理器发出以下命令:

```
DEFINE QREMOTE(ROME) RNAME(' ') RQMNAME(PALERMO) XMITQ(X) CLUSTER(ITALY)
```

该命令是队列管理器别名定义。 它将 ROME 定义为一个队列管理器，通过该队列管理器，来自集群 ITALY 中的任何队列管理器的消息都可以多跳到达其位于 PALERMO 的目标。 将放入队列管理器名称设置为 ROME 的队列的消息发送到具有队列管理器别名定义的网关队列管理器。 一旦出现，消息就会放在传输队列 X 上，并由非集群通道移动到队列管理器 PALERMO。

在此示例中，名称 ROME 的选择并不重要。 QREMOTE 和 RQMNAME 的值可能相同。

在接收消息时确定目标

当队列管理器接收到消息时，它会从传输头中抽取目标队列和队列管理器的名称。 它在传输头中查找与队列管理器同名的队列管理器别名定义。 如果找到一个，那么它会将队列管理器别名定义中的 RQMNAME 替换为传输头中的队列管理器名称。

以这种方式使用队列管理器别名有两个原因:

- 将消息定向到另一个队列管理器
- 将队列管理器名称更改为与本地队列管理器相同

在网关队列管理器中使用队列管理器别名在不同集群中的队列管理器之间路由消息。

应用程序可以使用队列管理器别名将消息发送到另一个集群中的队列。 该队列不必是集群队列。 队列在一个集群中定义。 应用程序已连接到另一个集群中的队列管理器。 网关队列管理器连接两个集群。 如果未将队列定义为集群队列，那么要进行正确的路由，应用程序必须使用队列名称和集群队列管理器别名来打开队列。 有关配置的示例，请参阅 第 275 页的『使用网关队列管理器创建两个重叠的集群』，从中获取图 1 中说明的应答消息流。

该图显示了应答消息返回到临时动态队列 (称为 RQ) 的路径。 连接到 QM3 的服务器应用程序使用队列管理器名称 QM2 打开应答队列。 队列管理器名称 QM2 被定义为网关队列管理器 QM1 上的集群队列管理器别名。 QM3 将回复信息发送到网关队列管理器 。 QM1 QM1 将消息路由到 QM2。

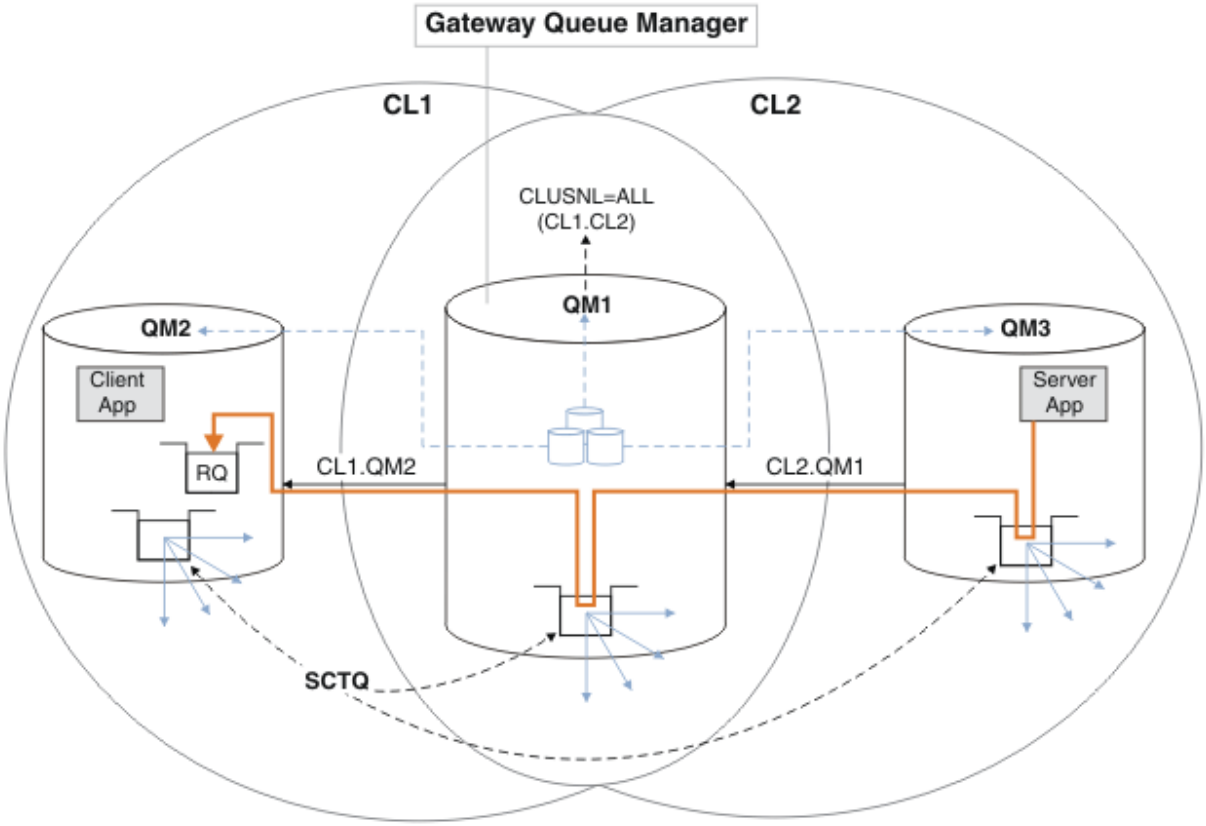


图 *60:* 使用队列管理器别名将应答消息返回到其他集群

路由工作方式如下。 每个集群中的每个队列管理器在网关队列管理器 QM1 上都有一个队列管理器别名定义。 别名在所有集群中进行集群。 从每个别名到队列管理器的灰色虚线箭头显示，每个队列管理器别名都解析为至少一个集群中的实际队列管理器。 在这种情况下， QM2 别名将同时在集群 CL1 和 CL2 中进行集群，并解析为 CL1 中的实际队列管理器 QM2 。 服务器应用程序使用对队列名称 RQ 的应答以及对队列管理器名称 QM2 的应答来创建应答消息。 CL2 消息被路由到网关队列管理器 QM1 ，因为队列管理器别名定义 QM2 是在群集 QM1 上定义的，而队列管理器 QM2 不在群集 CL2 中。 由于无法将消息发送到目标队列管理器，因此会将其发送到具有别名定义的队列管理器。

网关队列管理器 QM1 将消息放入集群传输队列，以便传输到 QM2 。 网关队列管理器 QM1 将消息路由到 QM2 ，因为 QM1 上 QM2 的队列管理器别名定义将 QM2 定义为真正的目标队列管理器。 定义不是循环的，因为别名定义只能引用实际定义; 别名不能指向自身。 真正的定义由网关队列管理器 QM1 解决，因为 QM1 和 QM2 都在同一个集群 CL1 中。 CL1 网关队列管理器 QM1 从存储库中为 QM2 找到连接信息，并将消息路由到 QM2 。 若要由 QM1 重新路由消息，服务器应用程序必须打开回复队列，并将 DEFBIND 选项设置为 MQBND_BIND_NOT_FIXED 。 如果服务器应用程序以 MQBND_BIND_ON_OPEN 选项打开了回复队列，则邮件不会被重新路由，而是进入死信队列。

将队列管理器用作集群中的网关，以实现来自集群外部的消息的工作负载均衡。

在集群中的多个队列管理器上定义名为 EDINBURGH 的队列。 您希望集群机制平衡从集群外部进入该队列的消息的工作负载。

来自集群外部的队列管理器需要传输队列和发送方通道到集群中的一个队列管理器。 此队列称为网关队列管理器。 要利用缺省工作负载均衡机制，必须应用下列其中一个规则:

- 网关队列管理器不得包含 EDINBURGH 队列的实例。
- 网关队列管理器在 ALTER QMGR 上指定 CLWLUSEQ(ANY) 。

有关来自集群外部的工作负载均衡示例，请参阅 第 310 页的『从集群外部配置工作负载均衡』

相关概念

应答队列别名和集群

应答队列别名定义用于指定应答信息的备用名称。 应答队列别名定义可以与集群一起使用，就像在分布式排队环境中一样。

队列别名和集群
使用队列别名来隐藏集群队列的名称，对队列进行集群，采用不同的属性或采用不同的访问控制。

相关任务

配置集群的请求/应答
从集群外部的队列管理器配置请求/应答消息路径。 通过使用网关队列管理器作为集群之间的通信路径，隐藏集群的内部详细信息。

从集群配置请求/应答
配置从集群到集群外部的队列管理器的请求/应答消息路径。 隐藏集群中的队列管理器如何使用网关队列管理器在集群外部进行通信的详细信息。

从集群外部配置工作负载均衡
配置从集群外部的队列管理器到集群队列的任何副本的消息路径。 结果是将来自集群外部的请求与集群队列的每个实例进行工作负载均衡。

配置集群之间的消息路径
使用网关队列管理器将集群连接在一起。 通过在网关队列管理器上定义集群队列或集群队列管理器别名，使队列或队列管理器对所有集群可见。

应答队列别名和集群

应答队列别名定义用于指定应答信息的备用名称。 应答队列别名定义可以与集群一起使用，就像在分布式排队环境中一样。

例如：

- 队列管理器 VENICE 上的应用程序使用 MQPUT 调用向队列管理器 PISA 发送消息。 应用程序在消息描述符中提供以下应答队列信息：

  ```
  ReplyToQ='QUEUE'
  ReplyToQMgr=''
  ```

- 为了可以在 PISA 上的 OTHERQ 上接收发送到 QUEUE 的应答，请在 VENICE 上创建用作应答队列别名的远程队列定义。 该别名仅在创建该别名的系统上有效。

  ```
  DEFINE QREMOTE(QUEUE) RNAME(OTHERQ) RQMNAME(PISA)
  ```

RQMNAME 和 QREMOTE 可以指定相同的名称，即使 RQMNAME 本身是集群队列管理器也是如此。

相关概念

队列管理器别名和集群
在将消息发送到集群或从集群发送消息时，使用队列管理器别名来隐藏队列管理器的名称，以及对发送到集群的消息进行工作负载均衡。

队列别名和集群
使用队列别名来隐藏集群队列的名称，对队列进行集群，采用不同的属性或采用不同的访问控制。

相关任务

配置集群的请求/应答
从集群外部的队列管理器配置请求/应答消息路径。 通过使用网关队列管理器作为集群之间的通信路径，隐藏集群的内部详细信息。

从集群配置请求/应答
配置从集群到集群外部的队列管理器的请求/应答消息路径。 隐藏集群中的队列管理器如何使用网关队列管理器在集群外部进行通信的详细信息。

从集群外部配置工作负载均衡
配置从集群外部的队列管理器到集群队列的任何副本的消息路径。 结果是将来自集群外部的请求与集群队列的每个实例进行工作负载均衡。

配置集群之间的消息路径
使用网关队列管理器将集群连接在一起。 通过在网关队列管理器上定义集群队列或集群队列管理器别名，使队列或队列管理器对所有集群可见。

队列别名和集群

使用队列别名来隐藏集群队列的名称，对队列进行集群，采用不同的属性或采用不同的访问控制。

QALIAS 定义用于创建用于识别队列的别名。 您可能出于多种原因创建别名:

- 您希望开始使用其他队列，但不希望更改应用程序。
- 您不希望应用程序知道它们要将消息放入的队列的实际名称。
- 您可能具有与定义队列的命名约定不同的命名约定。
- 您的应用程序可能无权使用其实名来访问队列，而只能使用其别名来访问队列。

使用 DEFINE QALIAS 命令在队列管理器上创建 QALIAS 定义。 例如，运行以下命令:

```
DEFINE QALIAS(PUBLIC) TARGET(LOCAL) CLUSTER(C)
```

此命令将名为 PUBLIC 的队列通告给集群 C 中的队列管理器。 PUBLIC 是解析为名为 LOCAL 的队列的别名。 发送到 PUBLIC 的消息将路由到名为 LOCAL 的队列。

您还可以使用队列别名定义将队列名称解析为集群队列。 例如，运行以下命令:

```
DEFINE QALIAS(PRIVATE) TARGET(PUBLIC)
```

此命令使队列管理器能够使用名称 PRIVATE 来访问名称为 PUBLIC 的集群中其他位置公布的队列。 由于此定义不包含 CLUSTER 属性，因此它仅适用于生成此属性的队列管理器。

相关概念

队列管理器别名和集群
在将消息发送到集群或从集群发送消息时，使用队列管理器别名来隐藏队列管理器的名称，以及对发送到集群的消息进行工作负载均衡。

应答队列别名和集群
应答队列别名定义用于指定应答信息的备用名称。 应答队列别名定义可以与集群一起使用，就像在分布式排队环境中一样。

相关任务

配置集群的请求/应答
从集群外部的队列管理器配置请求/应答消息路径。 通过使用网关队列管理器作为集群之间的通信路径，隐藏集群的内部详细信息。

从集群配置请求/应答
配置从集群到集群外部的队列管理器的请求/应答消息路径。 隐藏集群中的队列管理器如何使用网关队列管理器在集群外部进行通信的详细信息。

从集群外部配置工作负载均衡
配置从集群外部的队列管理器到集群队列的任何副本的消息路径。 结果是将来自集群外部的请求与集群队列的每个实例进行工作负载均衡。

配置集群之间的消息路径
使用网关队列管理器将集群连接在一起。 通过在网关队列管理器上定义集群队列或集群队列管理器别名，使队列或队列管理器对所有集群可见。

使用集群进行工作负载管理

通过在集群中的不同队列管理器上定义队列的多个实例，可以将为队列提供服务的工作分布在多个服务器上。 有几个因素可以防止在发生故障时将消息重新排队到另一个队列管理器。

除了设置集群以减少系统管理，您还可以创建集群，在这些集群中，多个队列管理器托管同一队列的实例。

您可以组织集群，使其中的队列管理器相互克隆。 每个队列管理器都能够运行相同的应用程序，并具有相同队列的本地定义。 z/OS 例如，在 z/OS 并行综合系统中，克隆的应用程序可以访问共享 Db2 或 Virtual Storage Access Method (VSAM) 数据库中的数据。 您可以通过具有多个应用程序实例在队列管理器之间分布工作负载。 应用程序的每个实例接收消息并独立于其他实例运行。

以这种方式使用集群的优点如下:

- 提高了队列和应用程序的可用性。
- 更快的消息吞吐量。
- 更均衡地分配网络中的工作负载。

托管特定队列实例的任何一个队列管理器都可以处理以该队列为目标的消息，并且应用程序在发送消息时不会命名队列管理器。 如果集群包含同一队列的多个实例，那么 IBM MQ 会选择要将消息路由到的队列管理器。 根据队列管理器和队列的可用性以及与队列管理器，队列和通道相关联的许多特定于集群工作负载的属性来选择合适的目标。 请参阅 集群中的工作负载均衡。

z/OS 在 IBM MQ for z/OS 中，队列共享组中的队列管理器可以将集群队列作为共享队列进行托管。 共享集群队列可供同一队列共享组中的所有队列管理器使用。 例如，在 具有同一队列的多个实例的集群中，队列管理器 QM2 和/或 QM4 可以是共享队列管理器。 每个都具有队列 Q3 的定义。 与 QM4 相同的队列共享组中的任何队列管理器都可以读取放入共享队列 Q3 的消息。 每个队列共享组最多可包含 32 个队列管理器，每个队列管理器具有对相同数据的访问权。 队列共享可显着提高消息吞吐量。

请参阅以下子主题以获取有关用于工作负载管理的集群配置的更多信息:

相关概念

集群与分布式排队的比较

分布式排队和集群

集群的组件

集群通道

为 MQPUT 禁用集群队列时将发生的情况

第 304 页的『将消息路由到集群或从集群路由消息』

使用队列别名，队列管理器别名和远程队列定义将集群连接到外部队列管理器和其他集群。

相关任务

编写和编译集群工作负载出口

第 235 页的『配置队列管理器集群』

集群提供了一种用于以简化初始配置和持续管理的方式互连队列管理器的机制。 您可以定义集群组件，以及创建和管理集群。

第 244 页的『设置新集群』

遵循以下指示信息来设置示例集群。 单独的指示信息描述在 TCP/IP 上设置集群， LU 6.2 以及使用单个传输队列或多个传输队列。 通过将消息从一个队列管理器发送到另一个队列管理器来测试集群工作。

第 310 页的『从集群外部配置工作负载均衡』

配置从集群外部的队列管理器到集群队列的任何副本的消息路径。 结果是将来自集群外部的请求与集群队列的每个实例进行工作负载均衡。

相关参考

集群发送方通道上设置的工作负载均衡不工作

集群队列监视样本程序 (AMQSCLM)

具有多个队列实例的集群的示例

在具有多个队列实例的集群的此示例中，消息将路由到队列的不同实例。 您可以强制将消息发送到队列的特定实例，并且可以选择将消息序列发送到其中一个队列管理器。

第 320 页的图 61 显示了一个集群，在该集群中，队列 Q3 有多个定义。 如果 QM1 上的应用程序将消息放入 Q3，那么它不一定知道 Q3 的哪个实例将处理其消息。 如果应用程序正在 QM2 或 QM4，上运行，其中存在 Q3 的本地实例，那么缺省情况下会打开 Q3 的本地实例。 通过设置 CLWLUSEQ 队列属性，可以将队列的本地实例视为队列的远程实例。

MQOPEN 选项 DefBind 控制在发出 MQOPEN 调用时还是从传输队列传输消息时选择目标队列管理器。

如果将 DefBind 设置为 MQBND_BIND_NOT_FIXED ，那么可以将消息发送到传输消息时可用的队列实例。 这可避免以下问题:

- 当消息到达目标队列管理器时，目标队列不可用。
- 队列的状态已更改。

- 已使用集群队列别名放入消息，并且在定义了集群队列别名的实例的队列管理器上不存在目标队列的实例。

如果在传输时发现了这些问题，那么将查找目标队列的另一个可用实例并重新路由消息。 如果没有可用的队列实例，那么会将消息放在死信队列上。

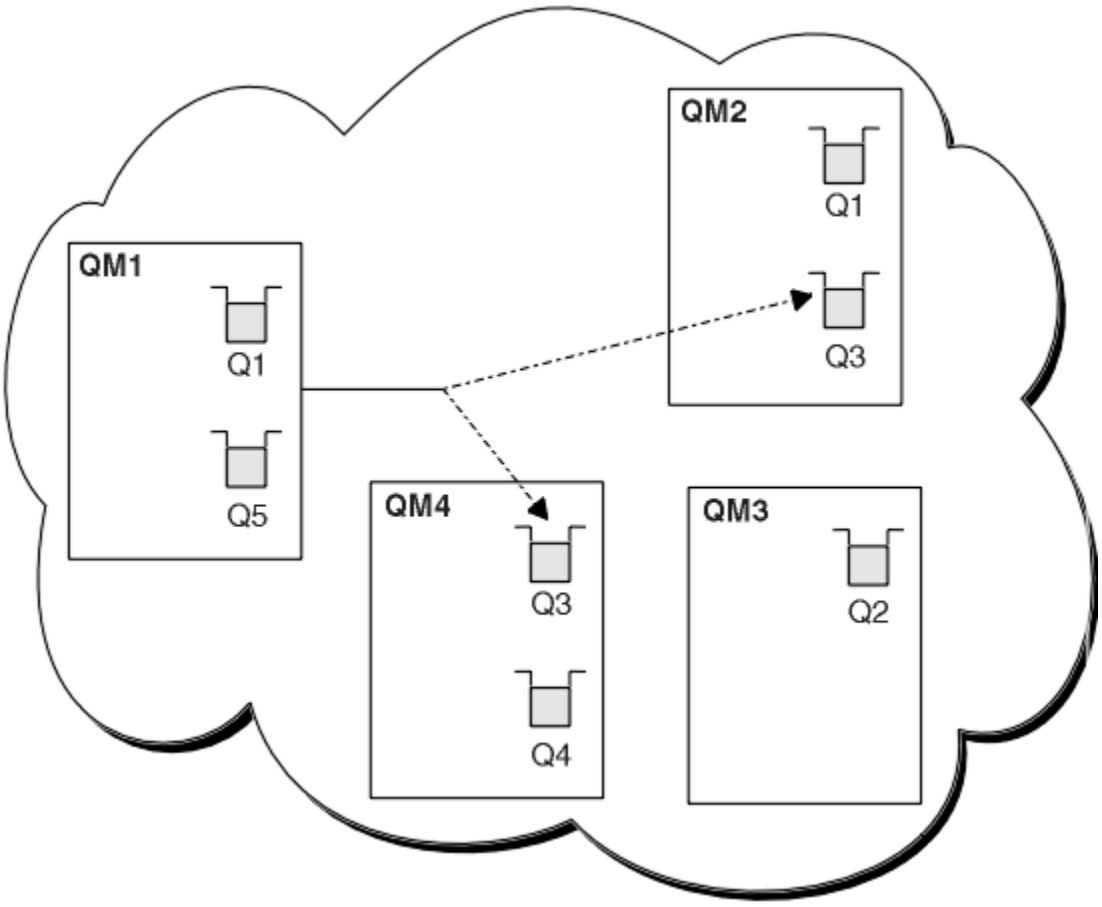


图 *61:* 具有同一队列的多个实例的集群

如果已将消息分配给具有 MQBND_BIND_ON_OPEN 的固定队列管理器或通道，那么会导致无法重新路由消息。 MQOPEN 上绑定的消息从不重新分配到其他通道。 另请注意，仅当集群通道实际发生故障时，才会进行消息重新分配。 如果通道已发生故障，那么不会发生重新分配。

如果目标队列管理器不服务，那么系统会尝试重新路由消息。 这样做时，它不会通过运行丢失消息的风险或创建重复项来影响消息的完整性。 如果队列管理器失败并使消息处于不确定状态，那么不会重新路由该消息。

z/OS 在 IBM MQ for z/OS 上，在消息重新分配过程完成之前，通道不会完全停止。 在方式设置为 FORCE 或 TERMINATE 的情况下停止通道会中断进程，因此如果执行此操作，那么某些 BIND_NOT_FIXED 消息可能已重新分配到另一个通道，或者这些消息可能不符合顺序。

注: z/OS

1. 在设置具有同一队列的多个实例的集群之前，请确保消息之间没有相互依赖关系。 例如，需要按特定顺序或由同一队列管理器进行处理。
2. 使同一队列的不同实例的定义相同。 否则，您将从不同的 MQINQ 调用中获取不同的结果。

相关概念

应用程序编程和集群

您无需进行任何编程更改即可利用同一队列的多个实例。 但是，除非将消息序列发送到队列的同一实例，否则某些程序无法正常工作。

相关任务

添加本地托管队列的队列管理器
遵循以下指示信息来添加 INVENTQ 实例，以提供在巴黎和纽约运行库存应用程序系统的额外容量。

在集群中使用两个网络
遵循以下指示信息在有两个不同网络的 TOKYO 中添加新商店。 这两者都需要可用于与东京的队列管理器进行通信。

在集群中使用主网络和辅助网络
遵循以下指示信息使一个网络成为主网络，另一个网络成为备份网络。 如果主网络存在问题，请使用备份网络。

添加队列以充当备份
遵循以下指示信息在芝加哥为现在在纽约运行的库存系统提供备份。 只有在纽约系统出现问题时，才会使用芝加哥系统。

限制使用的通道数
遵循以下指示信息以限制在各种队列管理器上安装价格检查应用程序时每个服务器运行的活动通道数。

添加托管队列的功能更强大的队列管理器
遵循以下指示信息，通过在洛杉矶以及纽约运行库存系统来提供额外的容量，其中洛杉矶可以处理两倍于纽约的消息数。

添加本地托管队列的队列管理器

遵循以下指示信息来添加 INVENTQ 实例，以提供在巴黎和纽约运行库存应用程序系统的额外容量。

开始之前

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

方案:

- 已按 向集群添加新队列管理器中所述设置 INVENTORY 集群。 它包含三个队列管理器; LONDON 和 NEWYORK 都保存完整存储库， PARIS 保存部分存储库。 库存应用程序在纽约的系统上运行，并连接到 NEWYORK 队列管理器。 应用程序由 INVENTQ 队列上的消息到达驱动。
- 我们要添加 INVENTQ 的实例，以提供额外的容量来运行巴黎和纽约的库存应用程序系统。

关于此任务

执行以下步骤以添加在本地托管队列的队列管理器。

过程

1. 变更 PARIS 队列管理器。

   要让巴黎的应用程序使用巴黎的 INVENTQ 和纽约的应用程序，我们必须通知队列管理器。 在 PARIS 上，发出以下命令:

   ```
   ALTER QMGR CLWLUSEQ(ANY)
   ```
2. 查看库存应用程序以获取消息亲缘关系。

   在继续之前，请确保库存应用程序不依赖于消息的处理顺序。 有关更多信息，请参阅处理消息亲缘关系。
3. 在巴黎的系统上安装库存应用程序。
4. 定义集群队列 INVENTQ。

   已由 NEWYORK 队列管理器托管的 INVENTQ 队列也将由 PARIS 托管。 在 PARIS 队列管理器上定义它，如下所示:

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
   ```

现在，您已完成所有定义，如果尚未完成，请在 IBM MQ for z/OS 上启动通道启动程序。 在所有平台上，在队列管理器 PARIS 上启动侦听器程序。 侦听器侦听入局网络请求，并在需要时启动集群接收方通道。

结果

第 322 页的图 62 显示此任务设置的集群。

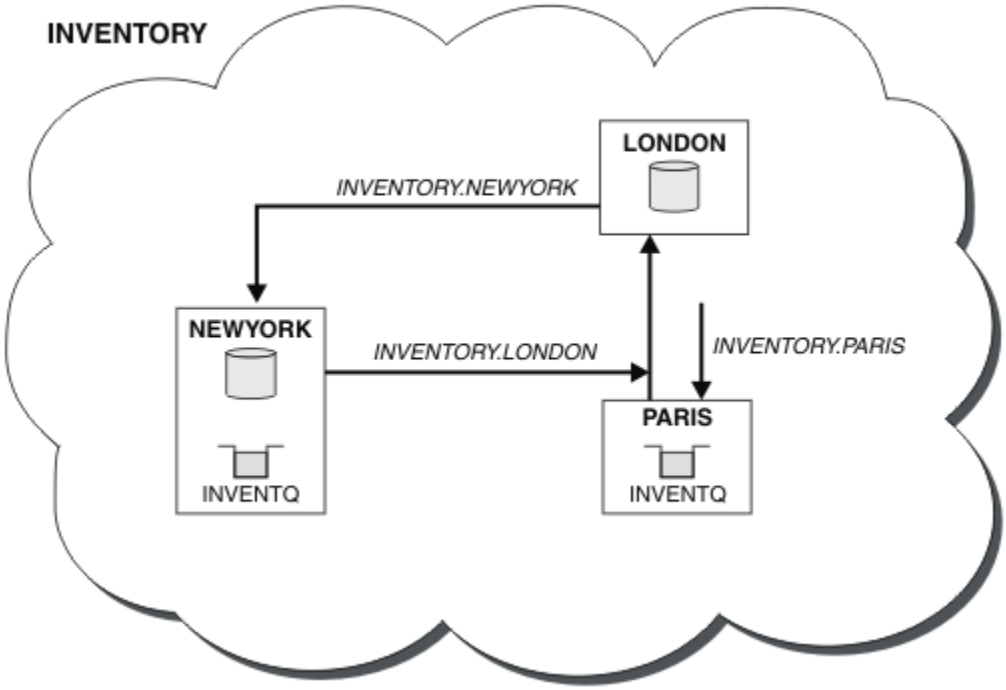


图 *62:* 具有三个队列管理器的 *INVENTORY* 集群

在不改变队列管理器 NEWYORK 或 LONDON 的情况下完成了对此集群的修改。 这些队列管理器中的完整存储库将自动更新为能够将消息发送到 INVENTQ (PARIS) 所需的信息。

下一步做什么

INVENTQ 队列和库存应用程序现在托管在集群中的两个队列管理器上。 这将提高其可用性，加快消息吞吐量，并允许在两个队列管理器之间分配工作负载。 由任何队列管理器 LONDON， NEWYORK 和 PARIS 放入 INVENTQ 的消息将交替路由到 PARIS 或 NEWYORK，以便均衡工作负载。

相关概念

具有多个队列实例的集群的示例
在具有多个队列实例的集群的此示例中，消息将路由到队列的不同实例。 您可以强制将消息发送到队列的特定实例，并且可以选择将消息序列发送到其中一个队列管理器。

应用程序编程和集群
您无需进行任何编程更改即可利用同一队列的多个实例。 但是，除非将消息序列发送到队列的同一实例，否则某些程序无法正常工作。

相关任务

在集群中使用两个网络
遵循以下指示信息在有两个不同网络的 TOKYO 中添加新商店。 这两者都需要可用于与东京的队列管理器进行通信。

在集群中使用主网络和辅助网络
遵循以下指示信息使一个网络成为主网络，另一个网络成为备份网络。 如果主网络存在问题，请使用备份网络。

添加队列以充当备份
遵循以下指示信息在芝加哥为现在在纽约运行的库存系统提供备份。 只有在纽约系统出现问题时，才会使用芝加哥系统。

限制使用的通道数
遵循以下指示信息以限制在各种队列管理器上安装价格检查应用程序时每个服务器运行的活动通道数。

添加托管队列的功能更强大的队列管理器
遵循以下指示信息，通过在洛杉矶以及纽约运行库存系统来提供额外的容量，其中洛杉矶可以处理两倍于纽约的消息数。

在集群中使用两个网络

遵循以下指示信息在有两个不同网络的 TOKYO 中添加新商店。 这两者都需要可用于与东京的队列管理器进行通信。

开始之前

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

方案:

- 已按 "向集群添加队列管理器" 中所述设置 INVENTORY 集群。 它包含三个队列管理器; LONDON 和 NEWYORK 都保存完整存储库， PARIS 保存部分存储库。 库存应用程序在纽约的系统上运行，并连接到 NEWYORK 队列管理器。 应用程序由 INVENTQ 队列上的消息到达驱动。
- 正在 TOKYO 中添加新商店，其中有两个不同的网络。 这两者都需要可用于与东京的队列管理器进行通信。

关于此任务

执行以下步骤以在集群中使用两个网络。

过程

1. 决定首先引用哪个完整存储库 TOKYO 。

   集群中的每个队列管理器都必须引用一个或多个完整存储库，以收集有关集群的信息。 它会构建自己的部分存储库。 您选择的存储库没有特别重要的意义。 在此示例中，选择了 NEWYORK 。 一旦新的队列管理器加入集群，它就会与这两个存储库进行通信。

2. 定义 CLUSRCVR 通道。

   集群中的每个队列管理器都需要定义可接收消息的集群接收方。 此队列管理器需要能够在每个网络上进行通信。

   ```
   DEFINE CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME('TOKYO.NETB.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
   channel using network B for TOKYO')
   ```

   ```
   DEFINE CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME('TOKYO.NETA.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
   channel using network A for TOKYO')
   ```

3. 在队列管理器 TOKYO. 上定义 CLUSSDR 通道

   集群中的每个队列管理器都需要定义一个集群发送方通道，在该通道上可以将消息发送到其第一个完整存储库。 在这种情况下，我们选择了 NEWYORK，因此 TOKYO 需要以下定义:

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
   channel from TOKYO to repository at NEWYORK')
   ```

现在，您已完成所有定义，如果尚未这样做，请在 IBM MQ for z/OS 上启动通道启动程序。 在所有平台上，在队列管理器 PARIS 上启动侦听器程序。 侦听器程序侦听入局网络请求，并在需要时启动集群接收方通道。

结果

第 324 页的图 63 显示此任务设置的集群。

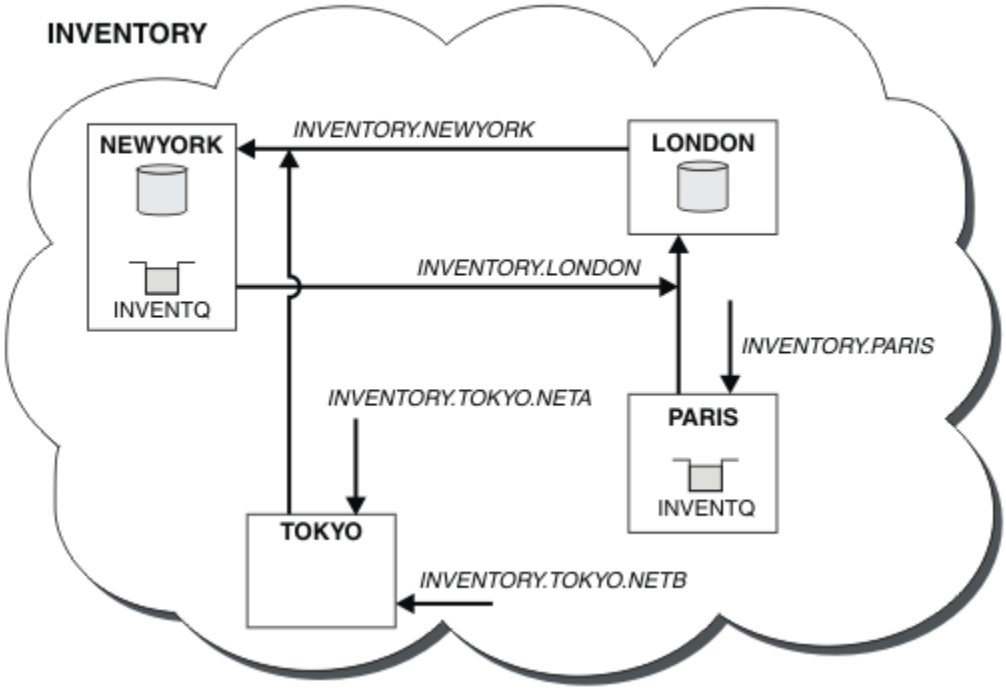


图 *63: INVENTORY* 集群，具有四个队列管理器

通过仅创建三个定义，我们将队列管理器 TOKYO 添加到集群中，并且有两个不同的网络路由可供其使用。

相关概念

具有多个队列实例的集群的示例
在具有多个队列实例的集群的此示例中，消息将路由到队列的不同实例。 您可以强制将消息发送到队列的特定实例，并且可以选择将消息序列发送到其中一个队列管理器。

应用程序编程和集群
您无需进行任何编程更改即可利用同一队列的多个实例。 但是，除非将消息序列发送到队列的同一实例，否则某些程序无法正常工作。

相关任务

添加本地托管队列的队列管理器
遵循以下指示信息来添加 INVENTQ 实例，以提供在巴黎和纽约运行库存应用程序系统的额外容量。

在集群中使用主网络和辅助网络
遵循以下指示信息使一个网络成为主网络，另一个网络成为备份网络。 如果主网络存在问题，请使用备份网络。

添加队列以充当备份
遵循以下指示信息在芝加哥为现在在纽约运行的库存系统提供备份。 只有在纽约系统出现问题时，才会使用芝加哥系统。

限制使用的通道数
遵循以下指示信息以限制在各种队列管理器上安装价格检查应用程序时每个服务器运行的活动通道数。

添加托管队列的功能更强大的队列管理器

遵循以下指示信息，通过在洛杉矶以及纽约运行库存系统来提供额外的容量，其中洛杉矶可以处理两倍于纽约的消息数。

第 254 页的『将队列管理器添加至集群』
遵循以下指示信息将队列管理器添加到您创建的集群。 使用单集群传输队列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 将消息传输到集群队列和主题。

在集群中使用主网络和辅助网络

遵循以下指示信息使一个网络成为主网络，另一个网络成为备份网络。 如果主网络存在问题，请使用备份网络。

开始之前

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

方案:

- 已按 第 323 页的『在集群中使用两个网络』中所述设置 INVENTORY 集群。 它包含四个队列管理器; LONDON 和 NEWYORK 都保存完整存储库; PARIS 和 TOKYO 保存部分存储库。 库存应用程序在纽约的系统上运行，并连接到队列管理器 NEWYORK。 TOKYO 队列管理器具有两个不同的网络，可以在这些网络上进行通信。
- 您希望使其中一个网络成为主网络，而另一个网络成为备份网络。 如果主网络存在问题，您计划使用备份网络。

关于此任务

使用 NETPRTY 属性来配置集群中的主网络和辅助网络。

过程

更改 TOKYO 上的现有 CLUSRCVR 通道。

要指示网络 A 通道是主通道，而网络 B 通道是辅助通道，请使用以下命令:

a)
```
ALTER CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) NETPRTY(2) DESCR('Main
cluster-receiver channel for TOKYO')
```
b)
```
ALTER CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) NETPRTY(1)
DESCR('Backup cluster-receiver channel for TOKYO')
```

下一步做什么

通过配置具有不同网络优先级的通道，您现在已定义到具有主网络和辅助网络的集群。 只要主网络可用，使用这些通道的集群中的队列管理器就会自动使用主网络。 队列管理器故障转移以在主网络不可用时使用辅助网络。

相关概念

具有多个队列实例的集群的示例
在具有多个队列实例的集群的此示例中，消息将路由到队列的不同实例。 您可以强制将消息发送到队列的特定实例，并且可以选择将消息序列发送到其中一个队列管理器。

应用程序编程和集群
您无需进行任何编程更改即可利用同一队列的多个实例。 但是，除非将消息序列发送到队列的同一实例，否则某些程序无法正常工作。

相关任务

添加本地托管队列的队列管理器
遵循以下指示信息来添加 INVENTQ 实例，以提供在巴黎和纽约运行库存应用程序系统的额外容量。

在集群中使用两个网络
遵循以下指示信息在有两个不同网络的 TOKYO 中添加新商店。 这两者都需要可用于与东京的队列管理器进行通信。

添加队列以充当备份
遵循以下指示信息在芝加哥为现在在纽约运行的库存系统提供备份。 只有在纽约系统出现问题时，才会使用芝加哥系统。

限制使用的通道数
遵循以下指示信息以限制在各种队列管理器上安装价格检查应用程序时每个服务器运行的活动通道数。

添加托管队列的功能更强大的队列管理器
遵循以下指示信息，通过在洛杉矶以及纽约运行库存系统来提供额外的容量，其中洛杉矶可以处理两倍于纽约的消息数。

添加队列以充当备份

遵循以下指示信息在芝加哥为现在在纽约运行的库存系统提供备份。 只有在纽约系统出现问题时，才会使用芝加哥系统。

开始之前

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

方案:

- 已按 第 254 页的『将队列管理器添加至集群』中所述设置 INVENTORY 集群。 它包含三个队列管理器; LONDON 和 NEWYORK 都保存完整存储库， PARIS 保存部分存储库。 库存应用程序在纽约的系统上运行，并连接到 NEWYORK 队列管理器。 应用程序由 INVENTQ 队列上的消息到达驱动。
- 正在芝加哥设立一个新商店，为现在在纽约运行的库存系统提供备份。 芝加哥系统只有在纽约系统出现问题时才使用。

关于此任务

请遵循以下步骤来添加队列以充当备份。

过程

1. 决定首先引用哪个完整存储库 CHICAGO 。

   集群中的每个队列管理器都必须引用一个或多个完整存储库，以收集有关集群的信息。 它会构建自己的部分存储库。 您为任何特定队列管理器选择的存储库没有特别重要的意义。 在此示例中，选择了 NEWYORK 。 一旦新的队列管理器加入集群，它就会与这两个存储库进行通信。
2. 定义 CLUSRCVR 通道。

   集群中的每个队列管理器都需要定义可接收消息的集群接收方。 在 CHICAGO 上，定义:

   ```
   DEFINE CHANNEL(INVENTORY.CHICAGO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(CHICAGO.CMSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-receiver
   channel for CHICAGO')
   ```
3. 在队列管理器 CHICAGO 上定义 CLUSSDR 通道。

   集群中的每个队列管理器都需要定义一个集群发送方通道，在该通道上可以将消息发送到其第一个完整存储库。 在这种情况下，我们选择了 NEWYORK，因此 CHICAGO 需要以下定义:

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
   channel from CHICAGO to repository at NEWYORK')
   ```
4. 更改现有集群队列 INVENTQ。

   已由 NEWYORK 队列管理器托管的 INVENTQ 是队列的主要实例。

   ```
   ALTER QLOCAL(INVENTQ) CLWLPRTY(2)
   ```
5. 查看库存应用程序以获取消息亲缘关系。

   在继续之前，请确保库存应用程序不依赖于消息的处理顺序。

6. 在 CHICAGO 中的系统上安装库存应用程序。
7. 定义备份集群队列 INVENTQ

   已由 NEWYORK 队列管理器托管的 INVENTQ 也将由 CHICAGO 作为备份进行托管。 在 CHICAGO 队列管理器上定义它，如下所示:

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) CLWLPRTY(1)
   ```

   现在，您已完成所有定义，如果尚未这样做，请在 IBM MQ for z/OS 上启动通道启动程序。 在所有平台上，在队列管理器 CHICAGO 上启动侦听器程序。 侦听器程序侦听入局网络请求，并在需要时启动集群接收方通道。

结果

第 327 页的图 64 显示此任务设置的集群。

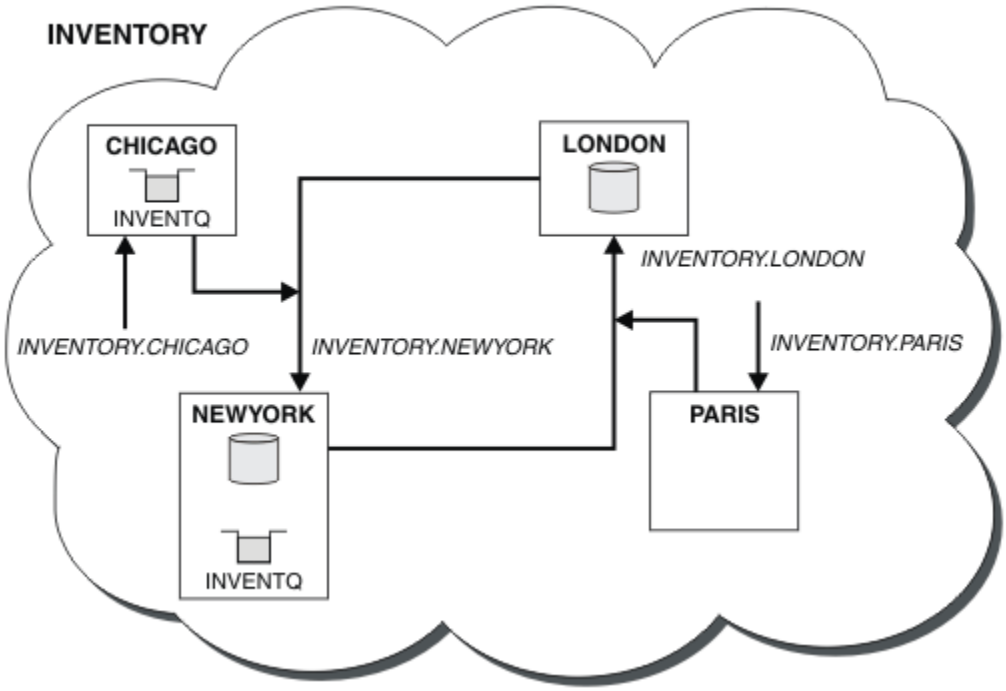


图 *64:* 具有四个队列管理器的 *INVENTORY* 集群

INVENTQ 队列和库存应用程序现在托管在集群中的两个队列管理器上。 CHICAGO 队列管理器是备份。 放入 INVENTQ 的消息将路由到 NEWYORK，除非在将其发送到 CHICAGO 时不可用。

注:

远程队列管理器的可用性取决于该队列管理器的通道状态。 当通道启动时，它们的状态会发生多次更改，其中一些状态比集群工作负载管理算法更不理想。 在实践中，这意味着可以在启动高优先级 (主) 目标的通道时选择低优先级 (备份) 目标。

如果需要确保没有消息转至备份目标，请不要使用 CLWLPRTY。 请考虑使用单独的队列，或者使用 CLWLRANK 从主队列到备份的手动切换。

相关概念

具有多个队列实例的集群的示例
在具有多个队列实例的集群的此示例中，消息将路由到队列的不同实例。 您可以强制将消息发送到队列的特定实例，并且可以选择将消息序列发送到其中一个队列管理器。

应用程序编程和集群
您无需进行任何编程更改即可利用同一队列的多个实例。 但是，除非将消息序列发送到队列的同一实例，否则某些程序无法正常工作。

相关任务

添加本地托管队列的队列管理器
遵循以下指示信息来添加 INVENTQ 实例，以提供在巴黎和纽约运行库存应用程序系统的额外容量。

在集群中使用两个网络
遵循以下指示信息在有两个不同网络的 TOKYO 中添加新商店。 这两者都需要可用于与东京的队列管理器进行通信。

在集群中使用主网络和辅助网络
遵循以下指示信息使一个网络成为主网络，另一个网络成为备份网络。 如果主网络存在问题，请使用备份网络。

限制使用的通道数
遵循以下指示信息以限制在各种队列管理器上安装价格检查应用程序时每个服务器运行的活动通道数。

添加托管队列的功能更强大的队列管理器
遵循以下指示信息，通过在洛杉矶以及纽约运行库存系统来提供额外的容量，其中洛杉矶可以处理两倍于纽约的消息数。

限制使用的通道数

遵循以下指示信息以限制在各种队列管理器上安装价格检查应用程序时每个服务器运行的活动通道数。

开始之前

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

方案:

- 将在各种队列管理器上安装价格检查应用程序。 为了使使用的通道数保持在较低数目，每个服务器运行的活动通道数都受到限制。 应用程序由 PRICEQ 队列上的消息到达驱动。
- 四个服务器队列管理器托管价格检查应用程序。 两个查询队列管理器将消息发送到 PRICEQ 以查询价格。 还有两个队列管理器配置为完整存储库。

关于此任务

执行以下步骤以限制使用的通道数。

过程

1. 选择两个完整存储库。

   选择两个队列管理器作为价格检查集群的完整存储库。 它们称为 REPOS1 和 REPOS2。

   发出以下命令:

   ```
   ALTER QMGR REPOS(PRICECHECK)
   ```
2. 在每个队列管理器上定义 CLUSRCVR 通道。

   在集群中的每个队列管理器上，定义集群接收方通道和集群发送方通道。 首先定义的是无关紧要的。

   ```
   DEFINE CHANNEL(PRICECHECK.SERVE1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(SERVER1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-receiver channel')
   ```
3. 在每个队列管理器上定义 CLUSSDR 通道。

   在每个队列管理器上创建 CLUSSDR 定义，以将该队列管理器链接到一个或多个完整存储库队列管理器。

   ```
   DEFINE CHANNEL(PRICECHECK.REPOS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(REPOS1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-sender channel to
   repository queue manager')
   ```
4. 安装价格检查应用程序。
5. 在所有服务器队列管理器上定义 PRICEQ 队列。

对每个命令发出以下命令:

```
DEFINE QLOCAL(PRICEQ) CLUSTER(PRICECHECK)
```

6. 限制查询所使用的通道数

   在查询队列管理器上，我们通过在每个队列管理器上发出以下命令来限制所使用的活动通道数:

   ```
   ALTER QMGR CLWLMRUC(2)
   ```

7. 如果尚未执行此操作，请在 IBM MQ for z/OS 上启动通道启动程序。 在所有平台上，启动侦听器程序。

   侦听器程序侦听入局网络请求，并在需要时启动集群接收方通道。

结果

第 329 页的图 65 显示此任务设置的集群。

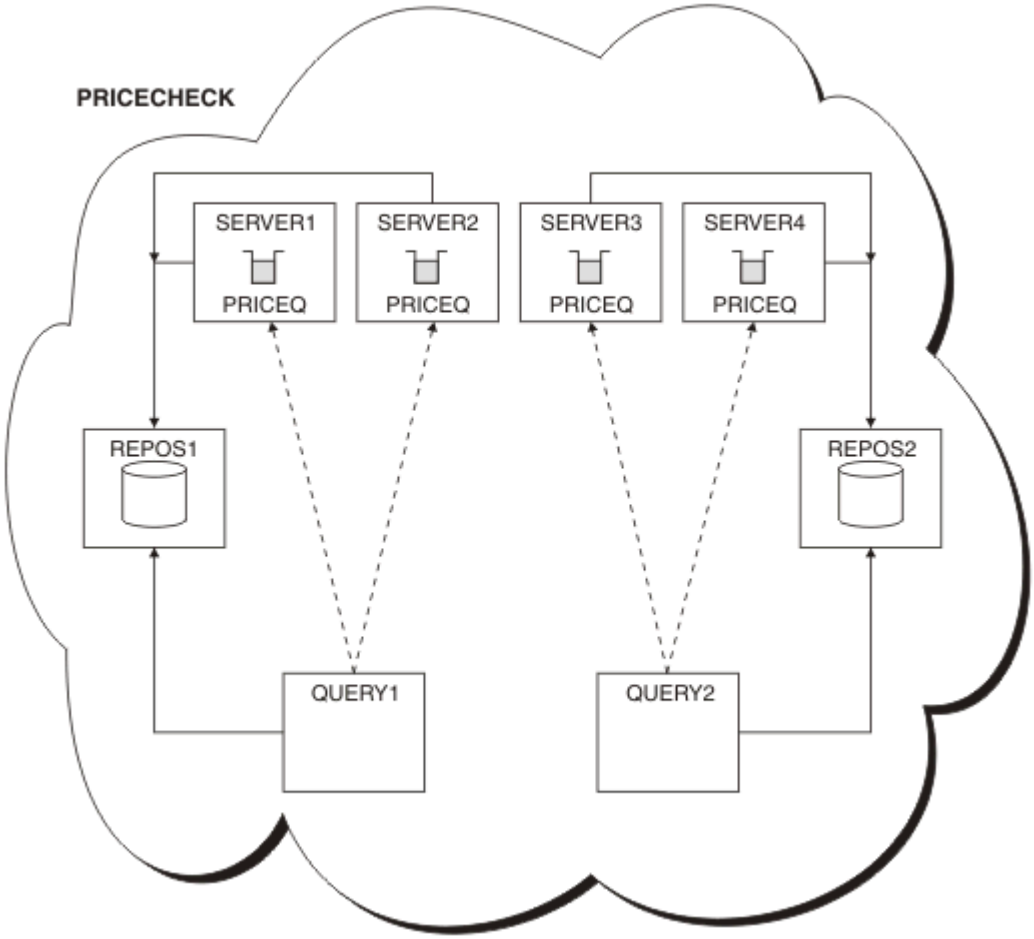


图 *65: PRICECHECK* 集群，具有四个服务器队列管理器，两个存储库和两个查询队列管理器

虽然 PRICECHECK 集群中有四个可用的 PRICEQ 队列实例，但每个查询队列管理器仅使用其中两个实例中的两个实例。 例如， QUERY1 队列管理器只有到 SERVER1 和 SERVER2 队列管理器的活动通道。 如果 SERVER1 变得不可用，那么 QUERY1 队列管理器将开始使用另一个队列管理器，例如 SERVER3。

相关概念

具有多个队列实例的集群的示例
在具有多个队列实例的集群的此示例中，消息将路由到队列的不同实例。 您可以强制将消息发送到队列的特定实例，并且可以选择将消息序列发送到其中一个队列管理器。

应用程序编程和集群
您无需进行任何编程更改即可利用同一队列的多个实例。 但是，除非将消息序列发送到队列的同一实例，否则某些程序无法正常工作。

相关任务

添加本地托管队列的队列管理器
遵循以下指示信息来添加 INVENTQ 实例，以提供在巴黎和纽约运行库存应用程序系统的额外容量。

在集群中使用两个网络
遵循以下指示信息在有两个不同网络的 TOKYO 中添加新商店。 这两者都需要可用于与东京的队列管理器进行通信。

在集群中使用主网络和辅助网络
遵循以下指示信息使一个网络成为主网络，另一个网络成为备份网络。 如果主网络存在问题，请使用备份网络。

添加队列以充当备份
遵循以下指示信息在芝加哥为现在在纽约运行的库存系统提供备份。 只有在纽约系统出现问题时，才会使用芝加哥系统。

添加托管队列的功能更强大的队列管理器
遵循以下指示信息，通过在洛杉矶以及纽约运行库存系统来提供额外的容量，其中洛杉矶可以处理两倍于纽约的消息数。

添加托管队列的功能更强大的队列管理器

遵循以下指示信息，通过在洛杉矶以及纽约运行库存系统来提供额外的容量，其中洛杉矶可以处理两倍于纽约的消息数。

开始之前

注: 要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

场景:

- 已按 第 254 页的『将队列管理器添加至集群』中所述设置 INVENTORY 集群。 它包含三个队列管理器: LONDON 和 NEWYORK 都保存完整存储库， PARIS 保存部分存储库并放置来自 INVENTQ 的消息。 库存应用程序在纽约连接到 NEWYORK 队列管理器的系统上运行。 应用程序由 INVENTQ 队列上的消息到达驱动。
- 洛杉矶正在设立一家新商店。 为了提供额外的容量，您希望在洛杉矶以及纽约运行库存系统。 新队列管理器可处理的消息数是纽约的两倍。

关于此任务

执行以下步骤以添加用于托管队列的功能更强大的队列管理器。

过程

1. 决定首先引用哪个完整存储库 LOSANGELES 。
2. 集群中的每个队列管理器都必须引用一个或多个完整存储库，以收集有关集群的信息。 它会构建自己的部分存储库。 您选择的存储库没有特别重要的意义。 在此示例中，选择了 NEWYORK 。 一旦新的队列管理器加入集群，它就会与这两个存储库进行通信。

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from LOSANGELES to repository at NEWYORK')
   ```

3. 在队列管理器 LOSANGELES 上定义 CLUSRCVR 通道。

   集群中的每个队列管理器都必须定义集群接收方通道，在该通道上可以接收消息。 在 LOSANGELES 上，定义:

   ```
   DEFINE CHANNEL(INVENTORY.LOSANGELES) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(LOSANGELES.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for queue manager LOSANGELES')
   CLWLWGHT(2)
   ```

集群接收方通道会公布队列管理器的可用性，以便从集群 INVENTORY 中的其他队列管理器接收消息。 将 CLWLWGHT 设置为 2 可确保洛杉矶队列管理器获得的库存消息数是纽约的两倍 (当 NEWYORK 的通道设置为 1 时)。

4. 更改队列管理器 NEWYORK 上的 CLUSRCVR 通道。

   确保洛杉矶队列管理器获取的库存消息数是纽约的两倍。 更改集群接收方通道的定义。

   ```
   ALTER CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) CLWLWGHT(1)
   ```

5. 查看库存应用程序以获取消息亲缘关系。

   在继续之前，请确保库存应用程序不依赖于消息的处理顺序。

6. 在洛杉矶的系统上安装库存应用程序
7. 定义集群队列 INVENTQ。

   INVENTQ 队列 (已由 NEWYORK 队列管理器主管) 也将由 LOSANGELES 主管。 在 LOSANGELES 队列管理器上定义它，如下所示:

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
   ```

   现在，您已完成所有定义，如果尚未这样做，请在 IBM MQ for z/OS 上启动通道启动程序。 在所有平台上，在队列管理器 LOSANGELES 上启动侦听器程序。 侦听器程序侦听入局网络请求，并在需要时启动集群接收方通道。

结果

第 330 页的『添加托管队列的功能更强大的队列管理器』 显示此任务设置的集群。

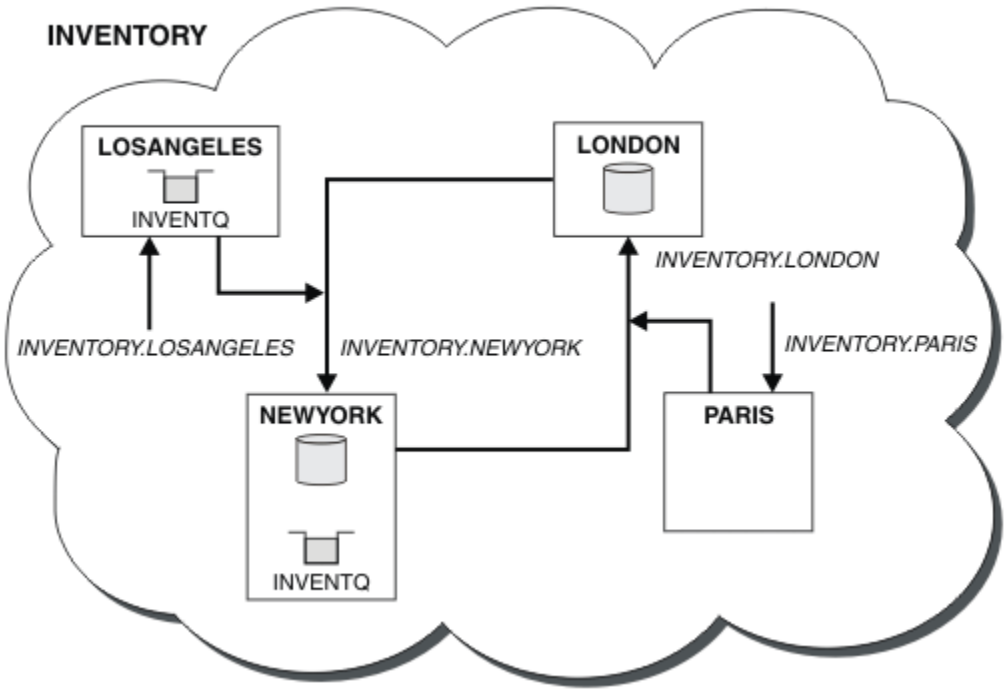


图 *66:* 具有四个队列管理器的 *INVENTORY* 集群

对集群的此修改已完成，而无需更改队列管理器 LONDON 和 PARIS。 这些队列管理器中的存储库将自动更新为能够将消息发送到 INVENTQ (LOSANGELES) 所需的信息。

下一步做什么

INVENTQ 队列和库存应用程序托管在集群中的两个队列管理器上。 该配置可提高其可用性，加快消息吞吐量，并允许在两个队列管理器之间分配工作负载。 LOSANGELES 或 NEWYORK 放入 INVENTQ 的消息将尽可

能由本地队列管理器上的实例处理。 LONDON 或 PARIS 放入的消息将路由到 LOSANGELES 或 NEWYORK，并且发送到 LOSANGELES 的消息数是此数的两倍。

相关概念

具有多个队列实例的集群的示例
在具有多个队列实例的集群的此示例中，消息将路由到队列的不同实例。 您可以强制将消息发送到队列的特定实例，并且可以选择将消息序列发送到其中一个队列管理器。

应用程序编程和集群
您无需进行任何编程更改即可利用同一队列的多个实例。 但是，除非将消息序列发送到队列的同一实例，否则某些程序无法正常工作。

相关任务

添加本地托管队列的队列管理器
遵循以下指示信息来添加 INVENTQ 实例，以提供在巴黎和纽约运行库存应用程序系统的额外容量。

在集群中使用两个网络
遵循以下指示信息在有两个不同网络的 TOKYO 中添加新商店。 这两者都需要可用于与东京的队列管理器进行通信。

在集群中使用主网络和辅助网络
遵循以下指示信息使一个网络成为主网络，另一个网络成为备份网络。 如果主网络存在问题，请使用备份网络。

添加队列以充当备份
遵循以下指示信息在芝加哥为现在在纽约运行的库存系统提供备份。 只有在纽约系统出现问题时，才会使用芝加哥系统。

限制使用的通道数
遵循以下指示信息以限制在各种队列管理器上安装价格检查应用程序时每个服务器运行的活动通道数。

应用程序编程和集群

您无需进行任何编程更改即可利用同一队列的多个实例。 但是，除非将消息序列发送到队列的同一实例，否则某些程序无法正常工作。

应用程序可以使用 MQOPEN 调用打开队列。 应用程序使用 MQPUT 调用将消息放入打开的队列中。 应用程序可以使用 MQPUT1 调用将单个消息放入尚未打开的队列中。

如果设置具有同一队列的多个实例的集群，那么没有特定的应用程序编程注意事项。 但是，要从集群的工作负载管理方面获益，您可能需要修改应用程序。 如果设置了具有同一队列的多个定义的网络，请查看应用程序以获取消息亲缘关系。

例如，假设您有两个应用程序依赖于以问题和答案形式在它们之间流动的一系列消息。 您可能希望答案返回到发送问题的同一队列管理器。 重要的是，工作负载管理例程不会将消息发送到托管应答队列副本的任何队列管理器。

您可能有需要按顺序处理消息的应用程序 (例如，发送必须按顺序检索的消息批次的数据库复制应用程序)。 使用分段消息还可能导致亲缘关系问题。

打开目标队列的本地或远程版本

请注意队列管理器如何选择是使用目标队列的本地版本还是远程版本。

1. 队列管理器打开目标队列的本地版本以读取消息或设置队列的属性。
2. 如果至少满足下列其中一个条件，那么队列管理器将打开要将消息写入的目标队列的任何实例:
   - 目标队列的本地版本不存在。
   - 队列管理器在 ALTER QMGR 上指定 CLWLUSEQ(ANY) 。
   - 队列管理器上的队列指定 CLWLUSEQ(ANY)。

相关概念

具有多个队列实例的集群的示例
在具有多个队列实例的集群的此示例中，消息将路由到队列的不同实例。 您可以强制将消息发送到队列的特定实例，并且可以选择将消息序列发送到其中一个队列管理器。

相关任务

添加本地托管队列的队列管理器
遵循以下指示信息来添加 INVENTQ 实例，以提供在巴黎和纽约运行库存应用程序系统的额外容量。

在集群中使用两个网络
遵循以下指示信息在有两个不同网络的 TOKYO 中添加新商店。 这两者都需要可用于与东京的队列管理器进行通信。

在集群中使用主网络和辅助网络
遵循以下指示信息使一个网络成为主网络，另一个网络成为备份网络。 如果主网络存在问题，请使用备份网络。

添加队列以充当备份
遵循以下指示信息在芝加哥为现在在纽约运行的库存系统提供备份。 只有在纽约系统出现问题时，才会使用芝加哥系统。

限制使用的通道数
遵循以下指示信息以限制在各种队列管理器上安装价格检查应用程序时每个服务器运行的活动通道数。

添加托管队列的功能更强大的队列管理器
遵循以下指示信息，通过在洛杉矶以及纽约运行库存系统来提供额外的容量，其中洛杉矶可以处理两倍于纽约的消息数。

处理消息亲缘关系
消息亲缘关系很少是良好的编程设计的一部分。 您需要除去消息亲缘关系以完全使用集群。 如果无法除去消息亲缘关系，那么可以强制使用同一通道将相关消息传递到同一队列管理器。

如果您有具有消息亲缘关系的应用程序，请先除去这些亲缘关系，然后再开始使用集群。

除去消息亲缘关系可提高应用程序的可用性。 应用程序将具有消息亲缘关系的一批消息发送到队列管理器。 队列管理器在仅接收部分批处理后失败。 发送队列管理器必须等待它恢复并处理不完整的消息批处理，然后才能发送更多消息。

除去消息亲缘关系还可提高应用程序的可伸缩性。 具有亲缘关系的一批消息可以在等待后续消息时锁定目标队列管理器上的资源。 这些资源可能长时间处于锁定状态，从而阻止其他应用程序执行其工作。

此外，消息亲缘关系会阻止集群工作负载管理例程做出队列管理器的最佳选择。

要除去亲缘关系，请考虑以下可能性:

- 在消息中携带状态信息
- 维护任何队列管理器可访问的非易失性存储器中的状态信息，例如在 Db2 数据库中
- 复制只读数据，以便多个队列管理器可以访问这些数据

如果不适合修改应用程序以除去消息亲缘关系，那么有许多可能的问题解决方案。

在 MQOPEN 调用上命名特定目标

在每个 MQOPEN 调用上指定远程队列名称和队列管理器名称，所有使用该对象句柄放入队列的消息都转至同一队列管理器 (可能是本地队列管理器)。

在每个 MQOPEN 调用上指定远程队列名称和队列管理器名称有缺点:

- 未执行工作负载均衡。 您不会利用集群工作负载均衡的优势。
- 如果目标队列管理器是远程的，并且有多个通道指向目标队列管理器，那么消息可能采用不同的路由，并且仍未保留消息序列。
- 如果队列管理器具有与目标队列管理器同名的传输队列的定义，那么消息将进入该传输队列而不是集群传输队列。

在 "应答队列管理器" 字段中返回队列管理器名称

允许接收批处理中第一条消息的队列管理器在其响应中返回其名称。 它使用消息描述符的 ReplyToQMgr 字段来执行此操作。 然后，发送端的队列管理器可以抽取应答队列管理器名称，并在所有后续消息上指定该名称。

使用来自响应的 ReplyToQMgr 信息有缺点:

- 请求队列管理器必须等待对其第一条消息的响应
- 在发送后续消息之前，必须编写其他代码以查找并使用 ReplyToQMgr 信息
- 如果有多个到队列管理器的路由，那么可能不会保留消息序列

在 MQOPEN 调用上设置 MQOO_BIND_ON_OPEN 选项

使用 MQOPEN 调用上的 MQOO_BIND_ON_OPEN 选项强制将所有消息放入同一目标。 将 消息组 与集群配合使用时，必须指定 MQOO_BIND_ON_OPEN 或 MQOO_BIND_ON_GROUP ，以确保在同一目标处处理组中的所有消息。

通过打开队列并指定 MQOO_BIND_ON_OPEN，可强制将发送到此队列的所有消息发送到队列的同一实例。 MQOO_BIND_ON_OPEN 将所有消息绑定到同一队列管理器以及同一路由。 例如，如果存在到同一目标的 IP 路由和 NetBIOS 路由，那么将在打开队列时选择其中之一，并且将使用获取的对象句柄对放入同一队列的所有消息执行此选择。

通过指定 MQOO_BIND_ON_OPEN ，可强制将所有消息路由到同一目标。 因此，具有消息亲缘关系的应用程序不会中断。 如果目标不可用，那么消息将保留在传输队列上，直到它再次可用为止。

如果在打开队列时在对象描述符中指定了队列管理器名称，那么 MQOO_BIND_ON_OPEN 也适用。 可能有多个到指定队列管理器的路径。 例如，可能存在多条网络路径，或者另一个队列管理器可能已定义别名。 如果指定 MQOO_BIND_ON_OPEN，那么将在打开队列时选择路由。

注: 这是推荐的技术。 但是，它在多中继段配置中不起作用，在该配置中，队列管理器会发布集群队列的别名。 在应用程序将同一队列管理器上的不同队列用于不同消息组的情况下，这也无济于事。

在 MQOPEN 调用上指定 MQOO_BIND_ON_OPEN 的替代方法是修改队列定义。 在队列定义上，指定 DEFBIND(OPEN)，并允许 MQOPEN 调用上的 DefBind 选项缺省为 MQOO_BIND_AS_Q_DEF。

在 MQOPEN 调用上设置 MQOO_BIND_ON_GROUP 选项

使用 MQOPEN 调用上的 MQOO_BIND_ON_GROUP 选项，强制将组中的所有消息放入同一目标。 将 消息组 与集群配合使用时，必须指定 MQOO_BIND_ON_OPEN 或 MQOO_BIND_ON_GROUP ，以确保在同一目标处处理组中的所有消息。

通过打开队列并指定 MQOO_BIND_ON_GROUP，可强制将组中发送到此队列的所有消息发送到该队列的同一实例。 MQOO_BIND_ON_GROUP 将组中的所有消息绑定到同一队列管理器，也绑定到同一路由。 例如，如果存在到同一目标的 IP 路由和 NetBIOS 路由，那么在打开队列时将选择其中一个路由，并且将使用获取的对象句柄对放入同一队列的组中的所有消息执行此选择。

通过指定 MQOO_BIND_ON_GROUP ，可强制将组中的所有消息路由到同一目标。 因此，具有消息亲缘关系的应用程序不会中断。 如果目标不可用，那么消息将保留在传输队列上，直到它再次可用为止。

如果在打开队列时在对象描述符中指定了队列管理器名称，那么 MQOO_BIND_ON_GROUP 也适用。 可能有多个到指定队列管理器的路径。 例如，可能存在多条网络路径，或者另一个队列管理器可能已定义别名。 如果指定 MQOO_BIND_ON_GROUP，那么将在打开队列时选择路由。

要使 MQOO_BIND_ON_GROUP 生效，必须在 MQPUT 上包含 MQPMO_LOGICAL_ORDER put 选项。 您可以将消息的 MQMD 中的 **GroupId** 设置为 MQGI_NONE，并且必须在消息的 MQMD **MsgFlags** 字段中包含以下消息标志:

- 组中的最后一条消息: MQMF_LAST_MSG_IN_GROUP
- 组中的所有其他消息: MQMF_MSG_IN_GROUP

如果指定了 MQOO_BIND_ON_GROUP ，但未对消息进行分组，那么行为等同于 MQOO_BIND_NOT_FIXED。

注: 这是确保将组中的消息发送到同一目标的建议方法。 但是，它在多中继段配置中不起作用，在该配置中，队列管理器会发布集群队列的别名。

在 MQOPEN 调用上指定 MQOO_BIND_ON_GROUP 的替代方法是修改队列定义。 在队列定义上，指定 DEFBIND(GROUP)，并允许 MQOPEN 调用上的 DefBind 选项缺省为 MQOO_BIND_AS_Q_DEF。

编写定制集群工作负载出口程序

您可以通过编写集群工作负载出口程序来规避消息亲缘关系问题，而不是修改应用程序。 编写集群工作负载出口程序并不容易，并且不是建议的解决方案。 该程序必须设计为通过检查消息内容来识别亲缘关系。 在识别亲缘关系后，程序必须强制工作负载管理实用程序将所有相关消息路由到同一队列管理器。

Multi V 9.2.0 配置统一集群

统一集群允许应用程序针对规模和可用性进行设计，并且可以连接到该统一集群中的任何队列管理器。

开始之前

有关集群的简介，请参阅 集群。 有关统一集群的简介，请参阅 第 335 页的『关于统一集群』。

关于此任务

统一集群使用 IBM MQ 集群在队列管理器之间进行通信，并在队列之间平衡工作负载。 但是，它们在以下方面与典型 IBM MQ 集群不同:

- 统一集群通常在集群中具有较少数量的队列管理器。 不应创建具有超过 10 个队列管理器的统一集群。
- 集群的每个成员都具有几乎相同的配置。
- 集群通常由单个应用程序或一组相关应用程序使用。
- 连接到集群的应用程序实例数应该大于或等于队列管理器数。

通过使用自动配置和自动集群支持，您可以简化创建统一集群的过程，从而使统一集群成员之间的配置保持一致。

过程

- 了解有关统一集群的信息
- 创建统一集群
- 创建统一集群
- 从统一集群暂挂队列管理器

Multi V 9.2.0 关于统一集群

统一集群部署的目标是，应用程序可以针对规模和可用性进行设计，并且可以连接到统一集群中的任何队列管理器。 这将除去对特定队列管理器的任何依赖关系，从而更好地实现消息传递流量的可用性和工作负载均衡。 z/OS 统一集群在 IBM MQ for z/OS 上不可用; 队列共享组提供统一集群的许多功能。

统一集群是 IBM MQ 集群的特定模式，它提供高度可用且水平缩放的小型队列管理器集合。 这些队列管理器的配置几乎相同，因此应用程序可以将它们作为单个组与之交互。 通过自动确保应用程序实例在队列管理器之间均匀分布，可以更轻松地确保使用集群中的每个队列管理器。

统一集群将除去管理员必须执行的一些手动步骤，以创建和管理一组独立，相互连接的队列管理器。 它们将一些客户机连接逻辑从客户机移动到队列管理器，其中有关应用程序活动级别的信息可以通知客户机的决策，以及它们应该连接到哪些队列管理器。

通过使用自动配置和自动集群支持，可以简化统一集群的初始创建，并随后使统一集群成员之间的配置保持一致。 使用此功能时，一个配置文件描述集群，另一个配置文件表示要应用于统一集群中所有队列管理器的 MQSC 配置。 在每次队列管理器重新启动时，都会重新应用配置，并且集群会自动形成。 有关使用此功能的更多详细信息，请参阅 第 345 页的『创建统一集群』。

要充分利用统一集群，还应将每个应用程序扩展为多个匹配实例，最好至少具有与队列管理器相同的实例(如果没有更多实例)。

无论大小如何， IBM MQ 集群都提供了多种功能:

- 所有集群资源的目录，可由集群中的任何成员发现

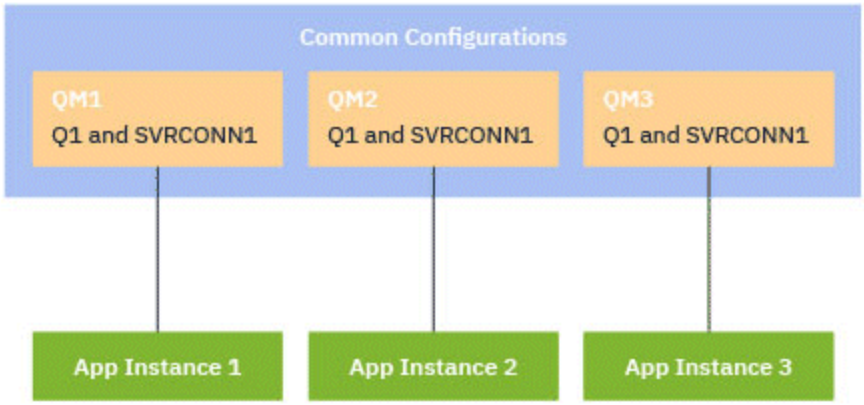
- 自动通道创建和连接
- 使用消息工作负载均衡在多个匹配队列之间进行水平缩放
- 基于可用性的动态消息路由

统一集群使用 IBM MQ 集群在队列管理器之间进行通信，并在队列之间平衡工作负载。 但是，它们在以下方面与典型 IBM MQ 集群不同:

- 统一集群通常在集群中具有较少数量的队列管理器。 不应创建具有超过 10 个队列管理器的统一集群。
- 集群的每个成员都具有几乎相同的配置。
- 集群通常由单个应用程序或一组相关应用程序使用。
- 连接到集群的应用程序实例数应该大于或等于队列管理器数。

在统一集群模式中，集群中的所有队列管理器都提供相同的消息传递服务。 例如，您可以将所有集群成员配置为定义相同的本地队列，并允许客户机应用程序连接到集群的任何成员。 您可能还定义了相同的服务器连接通道，也可能定义了相同的权限记录，通道认证规则等。 但是，集群成员在对象和配置方面仍可能存在一些差异。 例如，某些应用程序在连接到队列管理器时可能会创建临时动态队列。 另外，某些配置更新可能会在一段时间内跨成员进行推广; 例如，新的或更新的证书。 与常规 IBM MQ 集群一样，两个队列管理器将需要其他配置以使它们成为完整的存储库队列管理器。

下图显示了队列管理器具有类似的配置。 它们定义名为 Q1 的相同队列和相同的服务器连接通道 SVRCONN1。



请注意，对于具有相同服务器连接通道名称的多个队列管理器，要使用单个客户机通道定义表 (CCDT)，必须使用 IBM MQ 9.1.2 中引入的更新的 CCDT 格式。 请参阅 JSON 格式的客户机通道定义表 (client channel definition table)。

应用程序名称 和应用程序实例

应用程序名称显示为 **DISPLAY CONN(*) TYPE CONN** 命令的 APPLTAG 属性。 从 IBM MQ 9.1.2 开始，将更改应用程序名称的设置方式。

应用程序实例是一组紧密相关的连接，为该应用程序提供一个 执行单元 。 通常，这是一个单一的操作系统进程，可以具有多个线程和关联的 IBM MQ 连接。

有关应用程序名称 和应用程序实例的更多信息，请参阅 应用程序开发概念。

可重新连接的客户机

可以移动可重新连接的客户机以实现均匀的工作负载分布，而根据定义，不可重新连接的客户机无法重新连接到另一个队列管理器。 但是，可能仍有充分的理由将不可重新连接的客户机连接到统一集群中: 例如，因为客户机会创建某种形式的持久状态，而某些其他机制用于确保每个队列管理器中都有正在运行的应用程序实例。

本地绑定的应用程序

期望统一集群将 IBM MQ 应用程序作为客户机应用程序 (而不是本地绑定的应用程序) 进行连接。 不会阻止本地绑定的应用程序连接到统一集群成员，但是统一集群无法通过本地绑定的应用程序实现均匀的工作负载分布，因为它们无法连接到集群的任何其他成员。

相关任务

以受支持的编程语言指定应用程序名称

Multi V 9.2.0 自动应用程序均衡

自动应用程序均衡通过启用 IBM MQ 统一集群来密切管理整个集群中的应用程序分发，而不是依赖于随机化或手动将应用程序锁定到特定队列管理器，从而大大增强了应用程序分发和可用性。

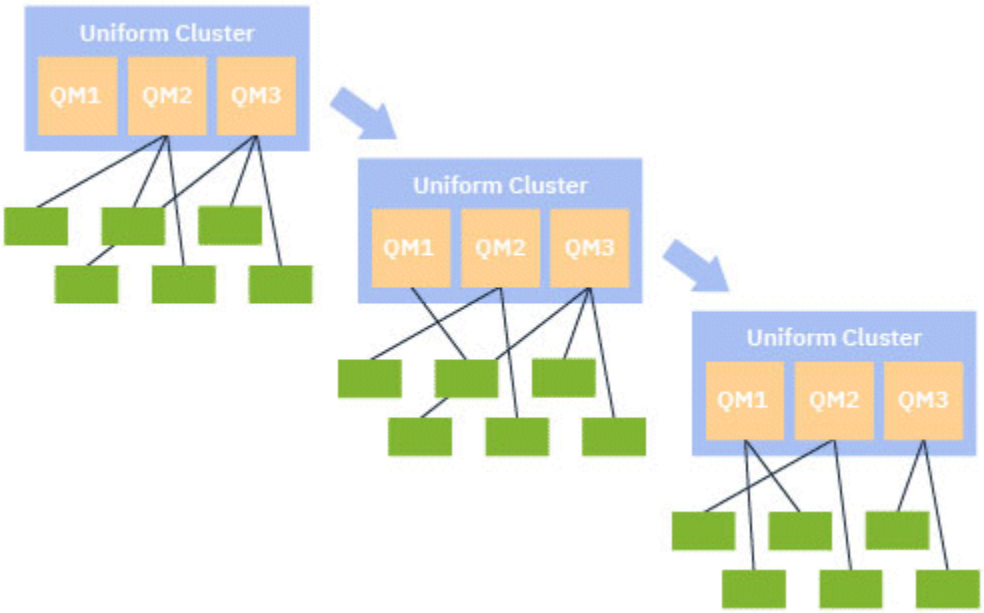
从 IBM MQ 9.2.0 开始，对于以 C， JMS， IBM MQ .NET 和 XMS .NET 编写的应用程序，支持在一组集群队列管理器之间进行自动均衡。

当存在至少与队列管理器相同的应用程序实例时，统一集群会不断确保每个队列管理器至少连接一个应用程序实例。

应用程序可以除去与队列管理器的特定亲缘关系，而是使用客户机通道定义表 (CCDT) 来安全地随机化与统一集群中的队列管理器组的连接。 由于以下原因，应用程序可以执行此操作:

- 当有足够的使用应用程序实例时，始终存在应用程序处理消息的实例。
- 停止队列管理器时，任何已连接的应用程序实例在集群中的其余队列管理器之间均匀分布。
- 启动队列管理器时，将自动重新均衡连接到集群中其他队列管理器的任何应用程序实例，以包含新启动的队列管理器。

这意味着统一集群将持续确保应用程序处于最佳分布状态，从而最大限度地实现消息处理，即使在发生计划内和计划外中断的情况下也是如此。



为了实现自动平衡，统一集群中的队列管理器定期在它们之间共享信息。 他们通过在主题树的保留 $SYS/MQ 分支下发布系统主题上的元数据来执行此操作。 统一集群中的每个队列管理器都预订其他队列管理器发布的消息，并构建统一集群中应用程序的状态图。

队列管理器监视整个集群中客户机应用程序的分布。 当连接到特定队列管理器的应用程序数足够少而确定集群不平衡时，该队列管理器会将系统主题上的请求发布到集群中的另一个队列管理器。

接收到消息时，目标队列管理器会请求其某个客户机应用程序重定向至请求队列管理器。 客户机应用程序接收重定向请求，关闭其连接并重新连接到请求队列管理器。 这种自动平衡机制对应用程序是透明的。 有关更多信息，请参阅第 338 页的『自动平衡的工作方式』。

通过定期分发有关已连接应用程序的元数据，统一集群可以在一段时间内实现客户机应用程序与队列管理器的大致均衡比率。 为了防止发生快速连续重定向事件，自动平衡算法会限制发出重定向请求的速率。

您可以在集群中的队列管理器之间监视应用程序的当前状态，并监视应用程序实例。 有关更多信息，请参阅监视应用程序均衡。 您还可以通过应用程序均衡来解决各种问题，如 应用程序均衡故障诊断中所述。

重新平衡仅对连接时间较长的应用程序有用。 如果您有连接时间较短的客户机应用程序 (例如，为了定期连接和断开与不同队列管理器的连接而编写的客户机应用程序)，那么应将它们配置为不可重新连接。 这将从队列管理器尝试均衡的应用程序集中除去这些应用程序。

相关概念

第 339 页的『自动平衡如何使用自动重新连接』

从 IBM MQ 9.2.0 开始，统一集群自动均衡使用 IBM MQ 的现有自动重新连接功能的增强功能。

Multi V 9.2.0 自动平衡的工作方式

在统一集群中，根据应用程序名称将客户机连接分组在一起。 使用同一应用程序名称连接到统一集群中任何成员的应用程序，都将被视为等同于使用同一应用程序名称的任何其他应用程序。

自动均衡可确保应用程序实例在集群成员之间均匀分布; 请参阅 第 336 页的『应用程序名称 和应用程序实例』 以获取更多信息。 使用 DISPLAY APSTATUS 命令可显示连接到队列管理器或统一集群的一个或多个应用程序和应用程序实例的状态。

例如，您可以将保险请求应用程序的所有实例设置为具有应用程序名称 "INSURANCE.REQUESTS"。 来自此应用程序的相关连接将在适当情况下自动分组到实例中，并在每个实例的基础上执行所有均衡。

当应用程序的新实例连接到统一集群的成员时，自动均衡算法将评估哪些队列管理器具有最少的 INSURANCE.REQUESTS 并将某些连接重定向到这些队列管理器。

仅在以下情况下启用自动均衡:

- 通道的 SHARECNV 值大于零。
- 出现以下情况之一:
  - 客户机应用程序指定 MQCNO_RECONNECT
  - `mqclient.ini` 文件指定 **Defrecon**=YES

注: 具有队列管理器亲缘关系的应用程序 (例如，由于持久预订或对队列的动态应答) 无法安全地进行重新平衡，并且应该使用 MQCNO_RECONNECT_QMGR 或完全不使用重新连接选项。

当客户机重定向到备用队列管理器时，它将像往常一样使用本地客户机通道定义表 (CCDT) 来查找新目标的连接信息。 因此，客户机使用包含统一集群的每个成员的条目的 CCDT 以及用于平衡初始连接的任何队列管理器组，这对于自动平衡的平稳高效操作非常重要。

使用 JSON 格式 CCDT 可简化此过程，因为它允许使用同一服务器连接名称的多个连接。 有关更多信息，请参阅 JSON 格式的客户机通道定义表。

相关概念

第 339 页的『自动平衡如何使用自动重新连接』

从 IBM MQ 9.2.0 开始，统一集群自动均衡使用 IBM MQ 的现有自动重新连接功能的增强功能。

ALW V 9.2.0 *JMS* 应用程序的自动均衡

当 Java 消息服务 (JMS) 应用程序自动均衡时， JMS 应用程序创建的 IBM MQ 连接的底层组将一起移动。

注:

注:

V 9.2.3 从 IBM MQ 9.2.3 开始，配置 ActivationSpecs 时， **dynamicallyBalanced** 属性可用。 此属性指定是否可以请求 MDB 作为统一集群中应用程序均衡的一部分从其他队列管理器接收消息。 有关更多信息，请参阅为入站通信配置资源适配器。

对于 Long Term Support 和 Continuous Delivery before IBM MQ 9.2.3，仅在 Java Standard Edition 环境中提供了对 JMS 应用程序自动均衡的支持。

对于处理 JMS 连接，统一集群具有 应用程序实例的概念。 对于 JMS， 应用程序实例 定义为 JMS 连接及其所有关联的 JMS 会话。

将在对应于 JMS 连接的客户机连接上分配唯一连接标记，然后将同一标记应用于对应于该 JMS 连接创建的 JMS 会话的客户机连接。

例如，如果一对客户机应用程序正在针对具有单个活动队列管理器 (队列管理器 1) 的统一集群运行 JMS 应用程序:

- 客户机 1 创建一个连接工厂，在该工厂上设置应用程序名称 "App1"，并创建 JMS 连接和三个 JMS 会话。 客户机 1 在队列管理器 1 中创建四个客户机连接，每个客户机连接共享同一个连接标记，并将其视为 "App1" 的单个实例。
- 客户机 2 还会创建一个连接工厂，在该工厂上设置应用程序名称 "App1"，并创建 JMS 连接和两个 JMS 会话。 客户机 2 创建三个客户机连接，每个客户机连接共享同一个连接标记 (不同于分配给客户机 1 的连接标记)，并将其视为 "App1" 的单个单独实例。
- 因此，队列管理器会看到两个 "App1" 实例

执行自动均衡时，将移动应用程序实例。 队列管理器选择应用程序实例 (一组共享同一连接标记的客户机连接)，并请求该实例移至另一队列管理器。 客户机代码接收请求并确保所有相关连接 (对应于 JMS 连接及其关联的 JMS 会话) 移至新的队列管理器。

例如，采用先前概述的应用程序实例集，并假定新的队列管理器 (队列管理器 2) 在统一集群中启动。

队列管理器 2 没有工作，但队列管理器 1 有 2 个 "App1" 实例，因此队列管理器 2 请求队列管理器 1 将 "App1" 实例传输到队列管理器 2。

队列管理器 1 选择要移动的 "App1" 实例。 对于本示例，假定它选择由客户机 1 创建的实例。

- 队列管理器 1 向客户机 1 发送请求，以将其实例 "App1" 移至 QM2。
- 客户机关闭其与队列管理器 1 的四个现有客户机连接，并创建与队列管理器 2 的四个新连接。
- 通常不应干扰 JMS 连接及其 JMS 会话 (处理中的短暂暂停除外)。

注:

如果在移动应用程序实例时正在进行某些操作，那么应用程序可能会接收到 JMS 异常。

JMS 异常将具有链接的 IBM MQ 异常，可以从中检索原因码以确定故障原因。

期望的原因码如下所示:

MQRC_CALL_中断
例如，当持久消息 (JMS 中的缺省值) 放在同步点之外，但该操作被重新连接中断时，会发生此情况。

MQRC_BACKED_OUT
例如，当尝试将消息放入同步点内时，会发生此情况，而重新连接会中断此情况。

相关概念

第 338 页的『自动平衡的工作方式』
在统一集群中，根据应用程序名称将客户机连接分组在一起。 使用同一应用程序名称连接到统一集群中任何成员的应用程序，都将被视为等同于使用同一应用程序名称的任何其他应用程序。

第 339 页的『自动平衡如何使用自动重新连接』
从 IBM MQ 9.2.0 开始，统一集群自动均衡使用 IBM MQ 的现有自动重新连接功能的增强功能。

Multi V 9.2.0 自动平衡如何使用自动重新连接

从 IBM MQ 9.2.0 开始，统一集群自动均衡使用 IBM MQ 的现有自动重新连接功能的增强功能。

在 IBM MQ IBM MQ 9.2.0 之前的版本中，自动重新连接功能会根据提供的连接详细信息 (通常是连接名称列表或客户机通道定义表 (CCDT)) 自动重新连接到队列管理器的备用实例或其他队列管理器。

在某些情况下， IBM MQ 客户机将以静默方式执行重新连接，而不会发现应用程序已发生此情况。 要重新连接到哪个队列管理器的决策完全取决于连接名称列表中的连接名称序列或 CCDT 中的工作负载均衡配置。

从 IBM MQ 9.2.0 开始，可以将重新连接请求发送到包含客户机应该重新连接到的队列管理器的提示的客户机。 在许多重新连接方案 (例如，队列管理器故障或发出命令 **endmqm -r** 的管理员) 中，队列管理器名称不包含在提示信息中，自动重新连接行为像当前一样工作。

但是，如果已配置统一集群，那么自动应用程序均衡会定期向客户机发送重新连接请求，以实现均衡集群。 在这些情况下，统一集群会在重新连接提示中指定队列管理器名称，以确保将客户机连接移至具有最少连接的队列管理器。

要实现自动平衡，必须执行以下操作:

- IBM MQ 应用程序使用 CCDT 从其检索连接信息。
- CCDT 包含统一集群中每个队列管理器的一个条目

如果情况并非如此，那么集群无法在集群的所有成员之间自动均衡应用程序。

如果应用程序正在使用低于 IBM MQ 9.2.0 的 IBM MQ 客户机版本，并且配置为支持自动客户机重新连接，那么统一集群可能会向其发送请求以完成其重新连接步骤。

将不要求客户机重新连接到特定队列管理器，而是通过与其他重新连接事件相同的重新连接逻辑序列。 通过确保将客户机配置为使用包含集群中每个成员的均匀加权条目的 CCDT，可以在 IBM MQ 9.2.0 之前在统一集群中实现客户机应用程序的均匀分布。

在连接到需要其他实例的队列管理器之前，应用程序可以进行多次重新连接尝试，因此，这是在集群中实现应用程序均匀分布的效率较低的方法。 在这些环境中实现自动平衡可能需要更长时间。

IBM MQ 个客户机不支持自动客户机重新连接

如果应用程序正在使用不支持自动客户机重新连接的 IBM MQ 客户机版本，那么该应用程序可能会从 MQI 调用接收到故障返回码。

如果应用程序尚未设计为手动处理故障和执行重新连接，那么可能需要对这些应用程序禁用自动平衡。

注: 将对标识为可重新连接的任何应用程序启用自动平衡，即，应用程序在其有效连接选项中具有 MQCNO_RECONNECT。

相关任务

第 346 页的『创建新的统一集群』
如何创建新的统一集群。

Multi V 9.2.4 影响统一集群中的应用程序重新平衡

通过自动应用程序均衡 (统一集群的功能)，可以要求应用程序连接在其生命周期中的任何时间点移动到备用队列管理器。

介绍

从 IBM MQ 9.2.4 开始，平衡算法会自动尝试考虑应用程序的状态，以最大程度地减少对应用程序流的中断。 通过向 IBM MQ 提供有关此应用程序所执行的应用程序类型或 IBM MQ 活动模式的其他信息，可以对其进行调整以适合特定应用程序或应用程序实例。

通常，开发或部署客户机应用程序的人员可能最适合了解此模式并将此信息提供给队列管理器 (请参阅 部署灵活且可扩展的客户机应用程序)，但也可能由管理员进行调整。

请注意，如果队列管理器无法在合理时间段内实现应用程序的均匀分布，那么应用程序连接可能仍会重新均衡到其他队列管理器，而不会在其 IBM MQ 流中等待一段方便的时间。

这也可以进行调整以满足需求。 如果更重要的是快速实现应用程序的均匀分布，那么您可以将产品配置为等待更少的时间，以找到合适的时间来重新平衡应用程序。 或者，如果更重要的是防止对应用程序的干扰，那么可以将产品配置为始终等待方便的时间来移动应用程序。

有关更多概述信息，请参阅 部署灵活且可扩展的客户机应用程序 。

对于 .NET 应用程序，请参阅 第 343 页的『影响 .NET 中的应用程序重新平衡』 以获取更多信息。

对于 XMS.NET 应用程序，请参阅 ConnectionFactory 以获取更多信息。

缺省应用程序平衡行为

缺省情况下，所有应用程序都会考虑应用程序与队列管理器交互的事务/工作单元状态。

对于本地事务，自动应用程序均衡可避免向事务中当前涉及的应用程序发出重新均衡请求。 虽然这并不能消除应用程序接收回退返回码的可能性，因为达到配置的重新平衡超时或真正的中断仍可能导致此类返回码，但这确实意味着应用程序通常不会在事务处于中间时被要求重新连接。

对于几乎在完成前一个事务之后立即开始新事务的应用程序，在重新平衡完成时，新事务中的初始调用可能会有延迟。 这确保自动应用程序均衡仍能够在统一集群中的队列管理器之间实现应用程序的均匀分布。

如果应用程序使用运行时间较长的事务，那么您可能希望考虑增大重新平衡超时的值，或者完全禁用此约束。 请参阅 第 341 页的『配置均衡行为』，以获取有关如何在 MQI 和 .NET 中控制此操作的链接，或者 "设计客户机应用程序以实现容错和可伸缩性" 以实现代码级别等效项。

配置均衡行为

要在 IBM MQ 重新均衡应用程序时精确影响，某些客户机应用程序环境可以在连接时提供有关所使用的消息传递模式的信息。

此信息在称为 均衡选项的新结构中提供。

对于 MQI，请参阅 第 341 页的『使用 MQI 配置均衡行为』。

对于此结构的 .NET 客户机等效项，请参阅 影响 .NET 中的应用程序重新平衡。

其他客户机环境当前不支持在连接时提供此结构。

使用 *MQI* 配置均衡行为

要在 IBM MQ 重新均衡应用程序时精确影响，某些客户机应用程序环境可以在连接时提供有关所使用的消息传递模式的信息。

在 MQI 中，平衡选项结构称为 MQBNO。

如果程序中未提供 均衡选项，那么支持客户机会在与客户机应用程序一起部署的 `client.ini` 文件中的 应用程序节 或 ApplicationDefaults 节 中派生此信息。

注: 这些节完全相同，但 `Application` 版本包含 **Name** 字段，用于标识这些选项应用于的应用程序。

如果提供了任一形式的节，那么所有字段都必须存在，但如果未显式设置，那么假定为 none 的 **BalanceOptions** 除外。

提供选项的首选顺序为:

1. MQBNO 结构由 CONNX 上的应用程序提供并完整使用
2. 或者，匹配的指定 `Application` 节 (如果存在) 仅用于生成一个节
3. 或者， `ApplicationDefaults` 节 (如果存在) 仅用于生成一个
4. 或者，没有此连接的 MQBNO 流。

您可以从 MQBNO 结构或 `client.ini` 文件提供三个关键信息段:

1. **ApplicationType** 或应用程序模式。

   此字段向 IBM MQ 指示此应用程序参与的 IBM MQ 活动的常规模式。

   支持三种类型的应用程序:

   简单
   不应应用 第 341 页的『缺省应用程序平衡行为』 中描述的缺省值以外的任何特定规则。

   请求/应答
   在每个 MQPUT 调用之后，响应消息需要匹配的 MQGET 调用。 有关更多详细信息，请参阅 第 342 页的『请求/应答平衡』。

受管客户机
重新平衡请求总是立即分派给客户机，客户机在它认为适当的时候重新平衡，例如， JEE 资源适配器将以这种方式注册。

2. **Timeout** ，在此之后，重新平衡可能会中断应用程序活动
3. 特定 **BalanceOptions**

请求/应答平衡

当应用程序类型指定为 **Request-Reply** (或 MQBNO_BALTYPE_REQREP) 时，期望应用程序实例执行的每个 PUT 都有一个响应 GET。 如果应用程序实例涉及多个线程，或者以批处理方式处理请求和响应，那么在任何给定时间都可能有多个请求和响应正在运行。

在发送的请求数等于接收的响应数或超过 MQBNO_BAL_TIMEOUT 的回退值之前，应用程序被视为没有资格移动。

此异常是针对请求消息配置消息到期时发生的情况。 假定应在请求消息的到期时间间隔内接收响应，并且当所有请求消息都已到期时，均衡算法不再等待其他响应，然后再考虑实例是否适合移动。

如果多个请求未完成，那么只会考虑发送的请求消息中的最新到期时间。 使用有意义的到期值时，应该将应用程序的 **Timeout** 均衡参数配置为至少与任何已发送的消息到期值一样高，以避免缩短任何期望的请求/响应到期窗口。

上述模式仅适用于期望具有没有未完成请求的时间段的应用程序。 例如，不断发送和接收消息的复杂多线程应用程序可能永远没有资格在此模式下进行重新平衡。

注意:

- 未尝试关联特定请求和响应，因此，如果一批飞行消息中的较早响应到期，那么应用程序可能仍等到最新请求到期后才有资格进行平衡。
- 特别需要注意的是，如果由于类似原因将 MQEI_UNLIMITED 和即将到期的消息组合在一起。

  如果到期时间有限的请求消息未完成，并且发送了到期时间为 MQEI_UNLIMITED 的新消息，那么平衡算法会将 MQEI_UNLIMITED 超时 不 考虑在内，这将继续遵守当前的最新到期时间。

  否则，先前已到期的响应可能会阻止应用程序有资格移动。 相应地，如果 MQEI_UNLIMITED 应答未完成，但随后发送到期请求，那么等待时间将缩短到最长 (有限) 到期时间。

  通常，您应该避免单个应用程序实例在均衡应用程序中同时发送到期和非到期请求消息，因为开发者或管理员更难以准确跟踪或定义重新均衡的资格。
- 在确定等待应答的时间长度时，仅考虑发送应用程序 (MQMD.**Expiry**) 指定的到期时间。 对此值的后续修改 (例如，使用 CAPEXPRY) 不会影响等待时间。

可能重新平衡应用程序的示例

示例 1

您已编写了一个将消息置于同步点下的应用程序，并通过发出 MQCMIT 调用来落实消息批处理。 MQCMIT 调用完成后，应用程序开始将消息放在新的同步点下。

建议的 IBM MQ 配置
缺省选项足够

结果
一旦满足配置的事务数，将在 MQCMIT 调用成功 (或失败) 后移动应用程序实例。

缺省情况下，如果一批消息超过 10 秒，那么如果已请求重新平衡，那么可能会回滚该消息。 如果您期望事务定期超过此限制并要求允许这样做，那么可以相应地扩展 **Timeout** 。

示例 2

在处理请求之后，您已编写了一个将一条消息放入集群队列实例的应用程序，另一个应用程序使用一条消息应答本地临时动态队列。 当从本地队列以破坏性方式读取请求时，应用程序将放入其下一条请求消息。

建议的 IBM MQ 配置
将 类型 设置为 MQBNO_BALTYPE_REQREP

结果
当应用程序完成 MQGET 调用时，将移动应用程序实例，此时应用程序实例将移至另一个队列管理器。将在新的队列管理器上执行任何后续 MQPUT 调用。

MQBNO

Multi V 9.2.4 影响 *.NET* 中的应用程序重新平衡

从 IBM MQ 9.2.4 开始，当您使用 MQQueueManager 类连接到队列管理器时，可以使用应用程序中的散列表来设置均衡选项属性。

以下常量是用于影响 .NET 中的应用程序均衡的常量:

重新平衡应用程序类型
平衡操作的类型; 由常量 **MQC.BALANCING_APPLICATION_TYPE_PROPERTY** 表示

- 必须使用此属性来设置 MQBNO 结构的 **ApplicationType** 字段。

  必须设置整数类型的值，可能的值为:

  MQC.BALANCING_APPLICATION_TYPE_SIMPLE
  简单均衡; 除了 第 340 页的『影响统一集群中的应用程序重新平衡』中描述的规则外，不会应用任何特定规则。 这是缺省值。

  MQC.BALANCING_APPLICATION_TYPE_REQUEST_REPLY
  请求-应答平衡; 在每个 **MQPUT** 调用之后，响应消息需要一个匹配的 **MQGET** 调用。 将延迟均衡，直到接收到此类消息，或者已超出请求消息 **EXPIRY** 。

  如果应用程序启用了重新连接并且未设置此属性，那么将使用 **MQC.BALANCING_APPLICATION_TYPE_SIMPLE**

重新平衡选项
由发出应用程序设置的平衡选项; 由常量 **MQC.BALANCING_OPTIONS_PROPERTY** 表示

- 必须使用此属性来设置 MQBNO 结构的 **BalanceOptions** 字段。

  必须设置整数类型的值，可能的值为:

  MQC.BALANCING_OPTIONS_NONE
  未设置任何选项。 这是缺省值

  MQC.BALANCING_OPTIONS_IGNORE_TRANSACTIONS
  设置此选项将允许重新平衡应用程序，即使在事务过程中也是如此。

  如果应用程序启用了重新连接并且未设置此属性，那么将使用 **MQC.BALANCING_OPTIONS_NONE** 。

重新平衡超时
超时，在此超时之后，重新平衡可能会中断应用程序活动; 由常量 **MQC.BALANCING_TIMEOUT_PROPERTY** 表示

- 必须使用此属性来设置 MQBNO 结构的 超时 字段。

  必须设置整数类型的值，可能的值为:

  MQC.BALANCING_TIMEOUT_AS_DEFAULT
  设置的缺省超时值。 这是缺省值

  MQC.BALANCING_TIMEOUT_IMMEDIATE
  发生立即超时

  MQC.BALANCING_TIMEOUT_NEVER
  未发生超时

  注: 必须仅从定义的值提供一个值，或者提供 0-999999999 秒的值。

部署灵活且可扩展的客户机应用程序
MQBNO

Multi V 9.2.0 统一集群的限制和注意事项

在配置统一集群时要考虑的限制和其他要点。

注: 有关配置统一集群时的一般需求，另请参阅 第 346 页的『创建新的统一集群』。

队列管理器之间一致性的重要性

缺省情况下，任何声明自己为 `reconnectable` 的应用程序都可以随时重新均衡到统一集群中的备用队列管理器。 这意味着必须在统一集群中的所有队列管理器上声明此类应用程序所需的任何资源 (例如，队列，主题或权限记录)。

未对队列管理器配置的一致性进行管理。 由系统管理员来配置集群的成员，以便他们具有类似的配置。

但是，您可以通过使用 在启动时从 MQSC 脚本自动配置 功能来帮助一致性，以共享用于为集群定义对象的 MQSC 脚本，从而确保所有对象都具有相同的定义。 有关更多信息，请参阅 第 346 页的『创建新的统一集群』。

此一致性扩展至集群的完整存储库队列管理器。 虽然对于传统 IBM MQ 集群，通常认为最佳实践是将完整存储库分隔到独立系统上，但在统一集群中，模型是完整存储库完全参与集群，并与其他节点一起处理应用程序工作负载。

注:

如果您正在使用较早的 Continuous Delivery 发行版运行统一集群，那么在从 IBM MQ 9.1.4 迁移到 IBM MQ 9.1.5 或更高版本期间，当某些队列管理器运行较旧的发行版和较新发行版时，应用程序不一定在集群之间均衡。 迁移所有队列管理器后，将恢复正常均衡。

运行包含 IBM MQ 9.1.5 或更高版本的队列管理器的统一集群后，不得将较早发行版中的队列管理器引入到统一集群中。 即，所有队列管理器都必须为 IBM MQ 9.1.5 或更高版本。 如果从较早发行版引入队列管理器，那么将发出包含错误代码 MQRCCF_CLUSTER_TOPIC_CONFLICT 的 FDC。

重叠的统一集群和传统 IBM MQ 集群

统一集群队列管理器最多可以参与一个统一集群，并且它也可以是任意数目的标准 IBM MQ 集群的成员。 将统一集群视为充当更广泛的集群中的单个队列管理器可能很有用。

请注意下列注意事项:

- 作为完整存储库的统一群集队列管理器只能是统一群集本身的完整存储库。
- 同样，部分存储库队列管理器虽然是统一群集的成员，但也可能属于更广泛的传统 IBM MQ 群集，因此不能用作统一群集之外的存储库。

  有关详细信息，请参阅 如何选择群集队列管理器来保存完整的存储库。

原因在于，作为传统 IBM MQ 群集和统一群集组合的完整存储库的队列管理器，会导致群集缓存中保存的数据在统一群集成员之间出现分歧，因此不利于按预期使用统一群集功能。

要将单个完整存储库队列管理器替换为统一集群，请将完整存储库与正在其上进行的应用程序工作分开，并仅将应用程序工作移至统一集群中。

对统一集群使用自动定义时，无法共享集群通道以在其他集群中使用，即，将 **CLUSTER** 属性设置为自动集群，并且 **CLUSNL** 属性必须为空。

应用程序平衡注意事项

应用程序实例并非始终均衡，尤其是在以下情况下:

- 当集群中的应用程序实例数少于队列管理器数时。
- 在客户机应用程序连接到集群或离开集群后的短时间内。

为了防止客户机应用程序过于频繁地重新平衡 (尤其是在应用程序连接即将到来和正在进行时)，对统一集群要求客户机应用程序重新平衡的频率设置了限制。 在一段时间的高连接或断开连接活动后，可能需要几分钟时间才能在统一集群中均衡其余应用程序实例。

有关更多信息，请参阅 应用程序平衡故障诊断。

应用程序亲缘关系

并非所有应用程序都适合在统一集群之间进行自动重新平衡。 仅对指定了 **MQCNO_RECONNECT** 的应用程序进行重新均衡。 与特定队列管理器具有亲缘关系的应用程序必须指定 **MQCNO_NO_RECONNECT** 选项或 **MQCNO_RECONNECT_Q_MGR**。 后者允许 HA 故障转移但不允许重新平衡。

创建与队列管理器的隐式亲缘关系的应用程序示例:

- 用于创建持久预订的应用程序。
- 例如，创建永久动态队列以接收应答消息的应用程序。
- 期望严格消息排序或需要序列中所有消息的应用程序将由同一应用程序实例或两者进行处理。

这些应用程序必须指定 **MQCNO_NO_RECONNECT** 或 **MQCNO_RECONNECT_Q_MGR** 选项，而不是 **MQCNO_RECONNECT**。

有关更多信息，请参阅 重新连接选项。

消息可用性

虽然应用程序均衡可以在失败或暂时不可用的队列管理器周围重新平衡连接，但统一集群不会在其成员之间复制消息数据。 对于数据可用性，如果节点发生故障，那么还必须将统一集群的每个成员配置为高可用性。 许多数据复制和高可用性解决方案都可用，并且可以与统一集群相结合，以实现最大的服务和数据可用性，例如:

- 支持容器编排自动重新启动的容器实例的复制存储器。 有关更多信息，请参阅 单个弹性队列管理器。
- RDQM 队列管理器。 有关更多信息，请参阅 RDQM 高可用性。
- 多实例队列管理器。 有关更多信息，请参阅多实例队列管理器。
- 本机 HA。 有关更多信息，请参阅 本机 HA。
- IBM MQ Appliance HA。 有关更多信息，请参阅高可用性。

统一集群的可扩展性和性能

要在统一集群中的队列管理器之间更紧密地集成和共享应用程序状态，需要比传统 IBM MQ 集群中更高级别的相互通信。 因此，建议不要缩放到单个统一集群中的大量队列管理器，因为额外的通信会对性能产生不利影响。

出于性能和管理原因，最好将统一集群视为充当向多个相关应用程序提供消息传递的单个传统队列管理器，但并非整个企业中的唯一消息传递服务。 在此模式中，少量队列管理器 (最多 10 个) 通常足以支持大量客户机应用程序连接。 通过应用程序均衡，可以轻松地从少量队列管理器 (例如 3 队列管理器) 开始，并通过添加更多队列管理器进行扩展。

注意: 在不具有建议特征的集群中启用统一集群行为，尤其是使用具有大量队列管理器的集群，可能会对性能产生严重影响。

相关概念

第 337 页的『自动应用程序均衡』
自动应用程序均衡通过启用 IBM MQ 统一集群来密切管理整个集群中的应用程序分发，而不是依赖于随机化或手动将应用程序锁定到特定队列管理器，从而大大增强了应用程序分发和可用性。

ALW V 9.2.0 创建统一集群

通过使用自动配置和自动集群支持，可以简化统一集群的初始创建，并随后保持统一集群成员之间的配置完全相同。

开始之前

在创建统一集群之前，应先阅读 第 344 页的『统一集群的限制和注意事项』。

关于此任务

您指示要将特定 IBM MQ 集群视为统一集群，方法是在 qm.ini 文件中提供至少包含 **Type**=统一 和 **ClusterName**=*<uniform cluster name>*的 AutoCluster 部分。

您可以选择使用 自动集群创建通过相同的 .ini 节来配置底层 IBM MQ 集群。 使用此自动集群支持来设置集群时，请提供用于描述集群及其完整存储库的配置文件。

如果要启动的队列管理器列示为其中一个完整存储库，那么将自动使其成为完整存储库。 同样，定义集群接收方通道时，将自动定义到完整存储库的集群发送方通道。

过程

要利用需要统一集群的其他功能，必须完成以下任一步骤:

- 将现有集群转换为统一集群，这符合 第 335 页的『关于统一集群』中描述的模式。
- 为此，请 创建新的统一集群 。

ALW V 9.2.0 创建新的统一集群

如何创建新的统一集群。

过程

1. 创建一个文件，描述您希望集群本身在完整存储库方面的外观。

   对于任何集群，两个完整存储库充当有关集群的信息的中心存储库。

   具体而言，您需要描述此集群中两个完整存储库的名称和连接名称。

   注: 这是在创建任何内容 (包括队列管理器) 之前完成的，下面的过程包括创建这些队列管理器。

   例如，假设您正在设置名为 UNICLUS 的统一集群，其中包含队列管理器成员 QMA， QMB， QMC 和 QMD。 在此示例中， QMA 和 QMB 将是完整存储库， QMC 和 QMD 将作为部分存储库。 样本配置文件 uniclus.ini:

   ```
   AutoCluster:
      Repository2Conname=QMA.dnsname(1414)
      Repository2Name=QMA
      Repository1Conname=QMB.dnsname(1414)
      Repository1Name=QMB
      ClusterName=UNICLUS
      Type=Uniform
   ```

   RepositoryNConname 字段用作其他集群成员的 *conname* 属性，以向其定义集群发送方 (CLUSSDR)，并且可以是多实例队列管理器的连接列表，并且可以选择包含端口。

2. 创建样本配置文件 uniclus.mqsc，其中包含要应用于所有集群成员的 MQSC 定义。

   此文件中需要一个必需行，它是集群接收方通道 (CLUSRCVR) 的定义，具有自动集群名称的 CLUSTER 属性 (通常通过 + AUTOCL + insert) 和包含 + QMNAME + insert 的通道名称。

   这描述了统一集群的其他成员如何连接到每个队列管理器，并用作如何连接到其他队列管理器的模板。 示例定义可能类似于:

   ```
   define channel('+AUTOCL+_+QMNAME+') chltype(clusrcvr) trptype(tcp)
   conname(+CONNAME+) cluster('+AUTOCL+') replace
   ```

   配置自动集群时，集群接收方通道的定义可以使用 CLUSTER， CONNAME 和 CHANNEL 字段中的一些附加插入，以使该定义在统一集群中的所有队列管理器上都相同。 其中包括:

   + AUTOCL +
   自动集群名称

   + QMNAME +
   要创建的队列管理器的名称

   + CONNAME +
   在队列管理器创建期间使用 **-iv** 参数或 Variables qm.ini 节定义的变量，用于连接名称参数字符串。 变量的名称可以是任何值。

请记住，通道名称限制为 20 个字符，因此插入的值以及替换插入的时间都需要符合该限制。 示例文件可能类似于以下内容:

```
*##########################################################################
* Compulsory section for all uniform cluster queue managers
*##########################################################################
define channel('+AUTOCL+_+QMNAME+') chltype(clusrcvr) trptype(tcp) conname(+CONNAME+)
cluster('+AUTOCL+') replace
*
*##########################################################################
* Configuration for all queue managers
*##########################################################################
define QL(APPQ) maxdepth(99999999) replace
define QL(APPQ2) maxdepth(99999999) replace
define channel(CLIENTCHL) chltype(svrconn) trptype(tcp) replace
```

3. 使这两个文件在将托管统一集群成员的每台机器上都可用。

   例如， `/shared/uniclus.ini` 和 `/shared/uniclus.mqsc`。

4. 在其中每台机器上，创建队列管理器。

   在命令行上，提供:

   a. 在期望的端口上启动侦听器的请求

   b. 自动 INI 配置请求 (**-ii**) 指向自动集群设置文件 (uniclus.ini)

   c. 针对自动 MQSC 配置 (**-ic**) 的请求，该请求指向包含统一集群的 CLUSRCVR 定义的 MQSC 配置文件。

   d. 此队列管理器的 CONNAME。

   在 QMA 的主机上:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMA.dnsname(1414) QMA
strmqm QMA
```

   统一集群中的每个队列管理器都是使用几乎相同的命令行创建的-对于统一集群，将自动处理完整存储库与部分存储库之间的所有差异。

   在 QMB 的主机上:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMB.dnsname(1414) QMB
strmqm QMB
```

   在 QMC 的主机上:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMC.dnsname(1414) QMC
strmqm QMC
```

   在 QMD 的主机上:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMD.dnsname(1414) QMD
strmqm QMD
```

   自动发生的情况:

   启动队列管理器时，会将 `uniclus.ini` 文件中的定义应用于 `qm.ini` 文件。 有关更多信息，请参阅 在启动时自动配置 qm.ini。 这会将 **AutoCluster** 定义添加到 `qm.ini` 文件中。

   如果队列管理器在 **AutoCluster** 节中被指定为完整存储库之一，那么它将自动转换为完整存储库，类似于发出 MQSC 命令 ALTER QMGR REPOS (**ClusterName**)，否则它将转换为部分存储库，类似于发出 MQSC 命令 ALTER QMGR REPOS ('')。

   当处理自动集群的集群接收方通道的定义时，会在 **AutoCluster** 节中定义从此队列管理器到所有完整存储库的集群发送方通道 (如果这是其中一个完整存储库，那么不包括本地队列管理器)。 这些发送方通道从已定义的本地集群接收方继承所有公共通道属性。

注意: 虽然这些通道是在没有进一步手动干预的情况下创建的，但它们是可以显示和管理的管理通道对象，如同任何其他通道定义一样。 您不应将这些对象与 "自动定义的" 集群发送方通道混淆，这些通道是由集群以路由消息流量的方式按需过渡创建的。

下一步做什么

验证统一集群设置

如果正确设置了 **ClusterName** 参数，并且队列管理器是指定集群的成员，那么将发出消息 AMQ9883 以确认集群现在标识为统一集群。

然后，可以使用统一集群功能，例如自动应用程序均衡。 在队列管理器启动期间，如果已设置此参数，但该名称不是有效的 IBM MQ 集群名称，那么将忽略该名称并发出错误消息 AMQ9882 。

如果该名称是有效的集群名称，但不存在所标识集群的集群通道，那么会向队列管理器错误日志发出警告消息 AMQ9881 ，以允许管理员识别并更正此情况。

验证自动化集群设置

如果已使用自动集群支持来设置统一集群，那么可以使用 runmqsc 命令来验证现在是否正确配置了指定为完整存储库的队列管理器:

```
QMA:
     1 : dis qmgr repos
AMQ8408I: Display Queue Manager details.
   QMNAME(QMA)                             REPOS(UNICLUS)
```

而部分存储库未配置为存储库:

```
QMC:
     1 : dis qmgr repos
AMQ8408I: Display Queue Manager details.
   QMNAME(QMC)                             REPOS( )
```

此外，您应该能够看到已使用配置 MQSC 文件中的通道名称将集群发送方通道 (CLUSSDR) 从每个队列管理器配置到其他完整存储库:

```
QMA:
     1 : dis chl(UNICLUS*) conname
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMA)                    CHLTYPE(CLUSRCVR)
   CONNAME(QMA.dnsname(1414))
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMB)                    CHLTYPE(CLUSSDR)
   CONNAME(QMB.dnsname(1414))

QMC:
     1 : dis chl(UNICLUS*) conname
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMA)                    CHLTYPE(CLUSSDR)
   CONNAME(QMA.dnsname(1414))
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMB)                    CHLTYPE(CLUSSDR)
   CONNAME(QMB.dnsname(1414))
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMC)                    CHLTYPE(CLUSRCVR)
   CONNAME(QMC.dnsname(1414))
```

相关概念

第 335 页的『关于统一集群』
统一集群部署的目标是，应用程序可以针对规模和可用性进行设计，并且可以连接到统一集群中的任何队列管理器。 这将除去对特定队列管理器的任何依赖关系，从而更好地实现消息传递流量的可用性和工作负载均衡。 z/OS 统一集群在 IBM MQ for z/OS 上不可用; 队列共享组提供统一集群的许多功能。

第 344 页的『统一集群的限制和注意事项』
在配置统一集群时要考虑的限制和其他要点。

ALW V 9.2.0 将现有集群转换为统一集群

您可以使用此过程将现有集群转换为统一集群。

关于此任务

如果将现有集群转换为统一集群，那么必须确保所有集群成员上都存在支持队列管理器之间的应用程序平衡所需的任何定义。

过程

1. 启用 IBM MQ 发布预订，包括所有队列管理器上的远程 (集群) 发布预订。

   这是统一集群功能的先决条件，因此必须确保队列管理器的 PSMODE 和 PSCLUS 属性都设置为缺省值 ENABLED。
2. 将 **AutoCluster** 部分添加到 `qm.ini` 文件中 IBM MQ 集群的名称，如 MQSC 对象定义 (例如集群通道) 中所使用。

   例如，如果集群的名称为 UNICLUS ，请在 `qm.ini` 文件中添加或修改 AutoCluster 节，如下所示:

   ```
   AutoCluster:
         ClusterName=UNICLUS
       Type=Uniform
   ```
3. 重新启动队列管理器以应用新设置。
4. 请考虑使用自动配置作为一种机制，以确保所有统一集群成员在启动时应用相同的配置。

   有关更多详细信息，请参阅 在启动时从 MQSC 脚本自动配置 。

相关概念

第 335 页的『关于统一集群』
统一集群部署的目标是，应用程序可以针对规模和可用性进行设计，并且可以连接到统一集群中的任何队列管理器。 这将除去对特定队列管理器的任何依赖关系，从而更好地实现消息传递流量的可用性和工作负载均衡。 z/OS 统一集群在 IBM MQ for z/OS 上不可用; 队列共享组提供统一集群的许多功能。

第 344 页的『统一集群的限制和注意事项』
在配置统一集群时要考虑的限制和其他要点。

Multi V 9.2.0 使用自动集群配置

您可以通过更改 `qm.ini` 配置信息来配置 IBM MQ 以启用自动配置。

注: 只能将 AutoCluster 节用于统一集群。

要配置的节

您可以更改以下节:

AutoConfig
在 `qm.ini` 文件中定义。 当队列管理器启动时，它将标识要应用的自动配置文件。

使用统一集群时，应使用此机制来分发相同的集群配置。

AutoCluster
在 `qm.ini` 文件中定义。 在队列管理器开始识别集群是否是自动集群的成员时使用，并且可以识别集群的完整存储库。

变量
在 `qm.ini` 文件中定义。 包含一些队列管理器变量。

AutoConfig 节的属性

在 AutoConfig 节中允许以下两个属性:

MQSCConfig=*<Path>*
该路径是完整文件路径，或者是在每次队列管理器启动时所有文件 `*.mqsc` 应用于队列管理器的目录的路径。

有关更多信息，请参阅 在启动时从 MQSC 脚本自动配置。

IniConfig=<Path>
该路径是完整的文件路径，或者是在每次队列管理器启动时所有文件 `*.ini` 都应用于 `qm.ini` 文件的目录的路径。

有关更多信息，请参阅第 83 页的『启动时自动配置 qm.ini』。

这些属性通常用作设置统一集群的一部分。 有关更多信息，请参阅第 346 页的『创建新的统一集群』。

示例节:

```
AutoConfig:
MQSCConfig=C:\MQ_Configuration\uniclus.mqsc
IniConfig=C:\MQ_Configuration\uniclus.ini
```

AutoCluster 节的属性

以下属性对于 AutoCluster 节是必需的:

类型 =统一
指定自动集群的类型，唯一有效的选项是 统一，它表示统一集群。

群集名称=<字符串
集群的名称，即自动集群名称。

以上属性的存在可实现统一集群的应用程序均衡。 有关更多详细信息，请参阅 第 337 页的『自动应用程序均衡』。

此外，如果在此节中描述了集群，那么可以执行集群的简化设置。 有关更多信息，请参阅第 346 页的『创建新的统一集群』。 使用此选项时，可以命名两个队列管理器，并为此自动集群的完整存储库提供其连接名称。

以下属性是 AutoCluster 节的可选属性，但您必须成对提供这些属性:

存储库名称 1 =<String>
这是自动集群中第一个完整存储库的队列管理器名称。 这可以是此队列管理器的名称，也可以是另一个队列管理器的名称。

存储库 1Conname=<Connection Name String>
这是自动集群成员应如何连接到此队列管理器的连接名称 (CONNAME) 值。

此外，您可以确定集群的第二个完整存储库:

Repository2Name=*<String>*
Repository2Conname=*<Connection name string>*

示例节:

```
AutoCluster:
   Repository2Conname=myFR1.hostname(1414)
   Repository2Name=QMFR1
   Repository1Conname= myFR2.hostname(1414)
   Repository1Name=QMFR2
   ClusterName=UNICLUS
   Type=Uniform
```

Variables 节的属性

`attribute=value` 对在属性字段中有效。 可以在创建队列管理器时使用 **crtmqm** 命令上的 **-iv** 命令行选项来提供这些内容。

在集群接收方通道的 CONNAME 和通道名称 MQSC 字段的自动集群配置期间，可以使用 "变量" 节中列出的属性。

V 9.2.0 从统一集群暂挂队列管理器

在统一集群正常运行期间，可重新连接的客户机应用程序实例可能随时自动重新均衡到集群中的任何队列管理器。 如果要在一段时间内 (例如在维护操作或问题确定期间) 阻止应用程序连接到特定队列管理器，请使用 SUSPEND QMGR 命令。

发出命令 SUSPEND QMGR CLUSTER (统一集群名称)

除了从 IBM MQ 集群中暂挂的通常影响外，在统一集群中， SUSPEND 命令还会阻止将可重新连接的应用程序重新均衡到此队列管理器。

发出该命令时，会立即将与队列管理器的任何此类现有连接重新均衡到集群中的其他可用队列管理器。

注: 如果统一集群中的所有队列管理器都已暂挂，那么应用程序仍将连接到一个或多个暂挂的队列管理器。

为避免将新连接添加到要维护的队列管理器，您应该停止一个或多个由客户机应用程序使用的服务器连接通道，例如发出以下 runmqsc 命令:

```
STOP CHANNEL(svrconn channel name)
```

例如，如果这些通道也用于连接维护窗口期间所需的管理应用程序，那么可能无法执行此操作。 因此，暂挂的队列管理器会定期检查附加的可重新连接的应用程序

如果存在任何可重新连接的应用程序，那么会将它们重新均衡到集群中的其他可用队列管理器。 现在可以在暂挂的队列管理器上执行维护。

注: 未被视为可移动的应用程序不受初始命令或后续重新扫描的影响，并且仍连接到暂挂的队列管理器; 请参阅 MOVCOUNT 以获取更多详细信息。

要恢复暂挂的队列管理器:

1. 如果需要，通过发出以下命令，启动服务器连接通道以恢复接受新的应用程序连接:

   ```
   START CHANNEL(svrconn channel name)
   ```

2. 发出以下 runmqsc 命令:

   ```
   RESUME QMGR CLUSTER(uniform cluster name)
   ```

队列管理器将恢复与统一集群的其余部分的通信，如果需要恢复平衡，那么可重新连接的客户机应用程序实例将重定向到此队列管理器。

配置发布/预订消息传递

您可以启动，停止和显示已排队的发布/预订的状态。 您还可以添加和除去流，以及从代理层次结构中添加和删除队列管理器。

过程

- 请参阅以下子主题以获取有关控制已排队的发布/预订的更多信息:
  - 第 352 页的『设置排队的发布/预订消息属性』
  - 第 353 页的『正在启动排队的发布/预订』
  - 第 353 页的『正在停止已排队的发布/预订』
  - 第 354 页的『添加流』
  - 第 355 页的『删除流』
  - 第 355 页的『添加预订点』
  - 第 361 页的『在发布/预订网络中组合主题空间』

设置排队的发布/预订消息属性

您可以使用队列管理器属性来控制某些发布/预订消息属性的行为。 您在 qm.ini 文件的 *Broker* 节中控制的其他属性。

关于此任务

您可以设置以下发布/预订属性: 有关详细信息，请参阅 队列管理器参数

表 *28:* 发布/预订配置参数	
描述	**MQSC 参数名称**
命令消息重试计数	**PSRTYCNT**
废弃无法传递的命令输入消息	**PSNPMSG**
无法传递的命令响应消息之后的行为	**PSNPRES**
在同步点下处理命令消息	**PSSYNCPT**

Broker 节用于管理以下配置设置:

- PersistentPublishRetry=yes | **强制**

  如果指定 Yes，那么如果通过排队的发布/预订接口发布持久消息失败，并且未请求任何否定应答，那么将重试发布操作。

  如果您请求了否定响应消息，那么将发送否定响应，并且不再重试。

  如果指定 **强制**，那么如果通过排队的发布/预订接口发布持久消息失败，那么将重试发布操作，直到成功处理该消息为止。 未发送否定响应。

- NonPersistentPublishRetry= yes | force

  如果指定 Yes，那么如果通过排队的发布/预订接口发布非持久消息失败，并且未请求否定应答，那么将重试发布操作。

  如果您请求了否定响应消息，那么将发送否定响应，并且不再重试。

  如果指定了 **强制**，那么如果通过排队的发布/预订接口发布非持久消息失败，那么将重试发布操作，直到成功处理为止。 未发送否定响应。

  注: 如果要对非持久消息启用此功能，那么除了设置 NonPersistentPublishRetry 值外，还必须确保队列管理器属性 **PSSYNCPT** 设置为 Yes。

  执行此操作还可能影响处理非持久发布的性能，因为 STREAM 队列中的 **MQGET** 现在发生在同步点下。

- PublishBatch **大小** =编号

  代理通常在同步点内处理发布消息。 单独落实每个发布可能效率低下，在某些情况下，代理可以在单个工作单元中处理多条发布消息。 此参数指定可在单个工作单元中处理的最大发布消息数

  PublishBatchSize 的缺省值为 5。

- PublishBatch **时间间隔** =编号

  代理通常在同步点内处理发布消息。 单独落实每个发布可能效率低下，在某些情况下，代理可以在单个工作单元中处理多条发布消息。 此参数指定批处理中的第一条消息与同一批处理中包含的任何后续发布之间的最长时间 (以毫秒为单位)。

  批处理时间间隔 0 指示最多可以处理 PublishBatchSize 消息，前提是这些消息立即可用。

  PublishBatchInterval 的缺省值为零。

过程

使用 IBM MQ Explorer ， 可编程命令或 **runmqsc** 命令来变更用于控制发布/预订行为的队列管理器属性。

示例

ALTER QMGR PSNPRES (SAFE)

正在启动排队的发布/预订

您可以通过设置队列管理器的 PSMODE 属性来启动已排队的发布/预订。

开始之前

阅读 PSMODE 的描述以了解发布/预订的三种方式:

- COMPAT
- DISABLED
- ENABLED

关于此任务

设置 QMGR PSMODE 属性以启动排队的发布/预订接口 (也称为代理) 和/或发布/预订引擎 (也称为 V 7 发布/预订)。 要启动已排队的发布/预订，您需要将 PSMODE 设置为 ENABLED。 缺省值为 ENABLED。

过程

如果尚未启用已排队的发布/预订接口，请使用 IBM MQ Explorer 或 **runmqsc** 命令来启用该接口。

示例

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

下一步做什么

IBM MQ 处理已排队的发布/预订命令和发布/预订消息队列接口 (MQI) 调用。

正在停止已排队的发布/预订

通过设置队列管理器的 PSMODE 属性来停止已排队的发布/预订。

开始之前

阅读 PSMODE 的描述以了解发布/预订的三种方式:

- COMPAT
- DISABLED
- ENABLED

关于此任务

设置 QMGR PSMODE 属性以停止排队的发布/预订接口 (也称为代理) 和/或发布/预订引擎 (也称为版本 7 发布/预订)。 要停止排队的发布/预订，需要将 PSMODE 设置为 COMPAT。 要完全停止发布/预订引擎，请将 PSMODE 设置为 DISABLED。

过程

使用 IBM MQ Explorer 或 **runmqsc** 命令来禁用排队的发布/预订接口。

示例

```
ALTER QMGR PSMODE (COMPAT)
```

添加流

您可以手动添加流以允许应用程序之间进行数据隔离，或者允许与 IBM WebSphere MQ 6 发布/预订层次结构进行互操作。

开始之前

熟悉发布/预订流的操作方式。 请参阅 Streams 和主题。

关于此任务

使用 PCF 命令， **runmqsc** 或 IBM MQ Explorer 来执行这些步骤。

注: 您可以按任意顺序执行步骤 1 和 2。 仅在步骤 1 和 2 都已完成后执行步骤 3。

过程

1. 定义与 IBM WebSphere MQ 6 流同名的本地队列。
2. 定义与 IBM WebSphere MQ 6 流同名的本地主题。
3. 将队列的名称添加到名称列表 SYSTEM.QPUBSUB.QUEUE.NAMELIST
4. 对发布/预订层次结构中位于 IBM WebSphere MQ 7.1 或更高版本的所有队列管理器重复此操作。

添加 'Sport'

在共享流 'Sport'的示例中， IBM WebSphere MQ 6.0 和 IBM WebSphere MQ 7.1 队列管理器在同一发布/预订层次结构中工作。 IBM WebSphere MQ 6 队列管理器共享名为 'Sport'的流。 此示例显示如何使用与 IBM WebSphere MQ 6 流 'Sport'共享的主题字符串 'Sport' 在名为 'Sport'的 IBM WebSphere MQ 7.1 队列管理器上创建队列和主题。

使用主题字符串 'Soccer/Results'发布到主题 'Sport'的 IBM WebSphere MQ 7.1 发布应用程序将创建生成的主题字符串 'Sport/Soccer/Results'。 在 IBM WebSphere MQ 7.1 队列管理器上，主题 'Sport'的订户 (主题字符串为 'Soccer/Results') 接收发布内容。

在 IBM WebSphere MQ 6 队列管理器上，流 'Sport'的订户 (主题字符串为 'Soccer/Results') 接收发布内容。

```
runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
define qlocal('Sport')
     1 : define qlocal('Sport')
AMQ8006: IBM MQ queue created.
define topic('Sport') topicstr('Sport')
     2 : define topic('Sport') topicstr('Sport')
AMQ8690: IBM MQ topic created.
alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM',
'SYSTEM.BROKER.ADMIN.STREAM')
     3 : alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM',
'SYSTEM.BROKER.ADMIN.STREAM')
AMQ8551: IBM MQ namelist changed.
```

注: 您需要向 **alter namelist** 命令提供名称列表对象中的现有名称以及要添加的新名称。

下一步做什么

有关流的信息将传递到层次结构中的其他代理。

如果代理为 IBM WebSphere MQ 6，请将其作为 IBM WebSphere MQ 6 代理进行管理。 即，您可以选择手动创建流队列，或者让代理在需要时动态创建流队列。 队列基于模型队列定义 SYSTEM.BROKER.MODEL.STREAM。

如果代理为 IBM WebSphere MQ 7.1，那么必须手动配置层次结构中的每个 IBM WebSphere MQ 7.1 队列管理器。

删除流

您可以从 IBM WebSphere MQ 7.1 或更高版本的队列管理器中删除流。

开始之前

在删除流之前，必须确保该流没有剩余的预订，并停顿所有使用该流的应用程序。 如果发布继续流向已删除的流，那么将系统恢复到干净的工作状态需要大量的管理工作。

过程

1. 查找托管此流的所有已连接代理。
2. 取消所有代理上流的所有预订。
3. 从名称列表 SYSTEM.QPUBSUB.QUEUE.NAMELIST 中除去队列 (与流同名)。
4. 从队列中删除或清除与流同名的所有消息。
5. 删除与流同名的队列。
6. 删除关联的主题对象。

下一步做什么

在托管流的所有其他已连接 IBM WebSphere MQ 7.1 或更高版本的队列管理器上重复步骤 3 到 5。

添加预订点

如何使用新的预订点来扩展已从较早版本的 IBM Integration Bus 迁移的现有已排队的发布/预订应用程序。

开始之前

1. 请检查是否尚未在 SYSTEM.QPUBSUB.SUBPOINT.NAMELIST 中定义预订点。
2. 检查是否存在与预订点同名的主题对象或主题字符串。

关于此任务

IBM WebSphere MQ 7.1 或更高版本的应用程序不使用预订点，但它们可以使用预订点迁移机制与执行此操作的现有应用程序进行互操作。

要点: 已从 IBM MQ 8.0 中除去预订点迁移机制。 如果需要迁移现有应用程序，那么在迁移到最新版本之前，必须执行产品版本的文档中描述的过程。

预订点不适用于使用 MQRFH1 头 (已从 IBM WebSphere MQ 6 或更早版本迁移) 的已排队发布/预订程序。

无需添加预订点即可使用为 IBM WebSphere MQ 7.1 或更高版本编写的集成发布/预订应用程序。

过程

1. 将预订点的名称添加到 SYSTEM.QPUBSUB.SUBPOINT.NAMELIST。
   - 在 z/OS 上， **NLTYPE** 是缺省值 NONE。
   - 在同一发布/预订拓扑中连接的每个队列管理器上重复该步骤。
2. 添加主题对象，最好为其提供预订点的名称，主题字符串与预订点的名称相匹配。
   - 如果预订点位于集群中，请在集群主题主机上添加主题对象作为集群主题。
   - 如果存在与预订点的名称具有相同主题字符串的主题对象，请使用现有主题对象。 您必须了解预订点复用现有主题的后果。 如果现有主题是现有应用程序的一部分，那么必须解决两个名称相同的主题之间的冲突。
   - 如果存在与预订点同名的主题对象，但存在不同的主题字符串，请创建具有不同名称的主题。
3. 将 **Topic** 属性 **通配符** 设置为值 BLOCK。

   阻止对 # 或 * 的预订将通配符预订隔离到预订点，请参阅 通配符和预订点。

4. 设置主题对象中所需的任何属性。

示例

此示例显示用于添加两个预订点 (USD 和 GBP) 的 **runmqsc** 命令文件。

```
DEFINE TOPIC(USD) TOPICSTR(USD)
DEFINE TOPIC(GBP) TOPICSTR(GBP) WILDCARD(BLOCK)
ALTER NL(SYSTEM.QPUBSUB.SUBPOINT.NAMELIST) NAMES(SYSTEM.BROKER.DEFAULT.SUBPOINT, USD, GBP)
```

注:

1. 在使用 **ALTER** 命令添加的预订点列表中包含缺省预订点。 **ALTER** 将删除名称列表中的现有名称。
2. 在更改名称列表之前定义主题。 队列管理器仅在队列管理器启动时以及更改名称列表时检查名称列表。

配置分布式发布/预订网络

连接到分布式发布/预订拓扑的队列管理器共享公共联合主题空间。 在一个队列管理器上创建的预订可以接收由连接到拓扑中另一个队列管理器的应用程序发布的消息。

您可以控制通过将集群或层次结构中的队列管理器连接在一起而创建的主题空间的范围。 在发布/预订集群中，对于要跨集群的主题空间的每个分支，必须对主题对象进行 "集群"。 在层次结构中，必须配置每个队列管理器以在层次结构中标识其 "父代"。

您可以通过选择每个发布和预订是本地发布和预订还是全局发布和预订，进一步控制拓扑中的发布和预订流。 本地发布和预订不会传播到发布者或订户所连接的队列管理器之外。

相关概念

分布式发布/预订网络

发布作用域

预订作用域

主题空间

相关任务

定义集群主题

配置发布/预订集群

在队列管理器上定义主题。 要使主题成为集群主题，请设置 **CLUSTER** 属性。 要选择要用于此主题的发布和预订的路由，请设置 **CLROUTE** 属性。

开始之前

某些集群配置无法容纳直接路由的发布/预订的开销。 在使用此配置之前，请浏览 设计发布/预订集群中详细描述的注意事项和选项。

要在集群中传播对集群的更改，必须始终至少有一个完整存储库可用。 在启动此任务之前，请确保存储库可用。

另请参阅 Routing for publish/subscribe clusters: Notes on behavior。

场景:

- 已按 第 254 页的『将队列管理器添加至集群』中所述设置 INVENTORY 集群。 它包含三个队列管理器; LONDON 和 NEWYORK 都保存完整存储库， PARIS 保存部分存储库。

关于此任务

在集群中的队列管理器上定义主题时，需要指定该主题是否为集群主题，以及 (如果是) 集群中针对此主题的发布和预订的路由。 要使该主题成为集群主题，请使用集群的名称配置 TOPIC 对象上的 **CLUSTER** 属性。 通过在集群中的队列管理器上定义集群主题，使该主题可供整个集群使用。 要选择要在集群中使用的消息路由，请将 TOPIC 对象上的 **CLROUTE** 属性设置为下列其中一个值:

- **DIRECT**

- **TOPICHOST**

缺省情况下，主题路由为 **DIRECT**。 这是 IBM MQ 8.0 之前的唯一选项。 当在队列管理器上配置直接路由集群主题时，集群中的所有队列管理器都可识别集群中的所有其他队列管理器。 在执行发布和预订操作时，每个队列管理器都可以直接连接到集群中的任何其他队列管理器。 请参阅 直接路由发布/预订集群。

从 IBM MQ 8.0 开始，您可以改为将主题路由配置为 **TOPICHOST**。 在您使用主题主机路由时，集群中的所有队列管理器都会知晓托管了路由主题定义的集群队列管理器（即，已定义主题对象的队列管理器）。 在执行发布和预订操作时，集群中的队列管理器只会连接到这些主题主机队列管理器，而不会彼此直接连接。 主题主机队列管理器负责将发布从执行发布的队列管理器路由至具有匹配预订的队列管理器。 请参阅 主题主机路由发布/预订集群。

注: 在对主题对象进行集群 (通过设置 **CLUSTER** 属性) 之后，无法更改 **CLROUTE** 属性的值。 必须先将该对象取消集群（将 **CLUSTER** 设置为 ' '），然后您才能更改此值。 对主题取消集群会将主题定义转换成本地主题，这将导致在某个时间段内，不会将发布传递到远程队列管理器上的预订；在执行此更改时应将此考虑在内。 请参阅定义与另一个队列管理器中集群主题具有相同名称的非集群主题的影响。 如果尝试在集群时更改 **CLROUTE** 属性的值，那么系统会生成 MQRCCF_CLROUTE_NOT_ALTERABLE 异常。

过程

1. 选择用于托管主题的队列管理器。

   任何集群队列管理器都可以托管主题。 选择三个队列管理器 (LONDON， NEWYORK 或 PARIS) 之一，并配置 TOPIC 对象的属性。 如果计划使用直接路由，那么您选择的队列管理器不会产生任何操作差异。 如果计划使用主题主机路由，那么所选队列管理器对路由发布具有其他职责。 因此，对于主题主机路由，请选择在其中一个功能更强大的系统上托管且具有良好网络连接的队列管理器。
2. 定义队列管理器上的主题。

   要使主题成为集群主题，请在定义主题时包含集群名称，并设置要用于此主题的发布和预订的路由。 例如，要在 LONDON 队列管理器上创建直接路由集群主题，请按如下所示创建主题:

   ```
   DEFINE TOPIC(INVENTORY) TOPICSTR('/INVENTORY') CLUSTER(INVENTORY) CLROUTE(DIRECT)
   ```

   通过在集群中的队列管理器上定义集群主题，使该主题可供整个集群使用。

   有关使用 **CLROUTE** 的更多信息，请参阅 DEFINE TOPIC (CLROUTE) 和 Routing for publish/subscribe clusters: Notes on behavior。

结果

集群已准备好接收主题的发布和预订。

下一步做什么

如果已配置主题主机路由的发布/预订集群，那么可能要为此主题添加第二个主题主机。 请参阅第 359 页的『向主题主机路由集群添加额外的主题主机』。

如果您有多个单独的发布/预订集群，例如，由于您的组织在地理上分散，您可能希望将一些集群主题传播到所有集群中。 您可以通过连接层次结构中的集群来执行此操作。 请参阅第 364 页的『组合多个集群的主题空间』。 您还可以控制哪些发布从一个集群流向另一个集群。 请参阅 第 366 页的『组合和隔离多个集群中的主题空间』。

相关概念

组合发布和预订作用域

从 IBM WebSphere MQ 7.0 开始，发布和预订作用域独立工作以确定队列管理器之间的发布流。

在发布/预订网络中组合主题空间

将队列管理器的主题空间与发布/预订集群或层次结构中的其他队列管理器组合在一起。 将发布/预订集群以及发布/预订集群与层次结构组合在一起。

相关任务

将群集主题定义移至不同的队列管理器

对于主题主机路由或直接路由群集，在停用队列管理器时，或由于群集队列管理器发生故障或在相当长一段时间内不可用，可能需要移动群集主题定义。

向主题主机路由集群添加额外的主题主机
在主题主机路由的发布/预订集群中，可以使用多个队列管理器，通过在这些队列管理器上定义相同的集群主题对象，将发布路由到预订。 这可用于提高可用性和工作负载均衡。 为同一集群主题对象添加额外的主题主机时，可以使用 **PUB** 参数来控制何时开始通过新主题主机路由发布。

将队列管理器连接到发布/预订层次结构
将子队列管理器连接到层次结构中的父队列管理器。 如果子队列管理器已经是另一个层次结构或集群的成员，那么此连接会将层次结构连接在一起，或者将集群连接到层次结构。

将队列管理器与发布/预订层次结构断开连接
将子队列管理器与发布/预订层次结构中的父队列管理器断开连接。

设计发布/预订集群
分布式发布/预订故障诊断
禁止集群发布/预订

将群集主题定义移至不同的队列管理器

对于主题主机路由或直接路由群集，在停用队列管理器时，或由于群集队列管理器发生故障或在相当长一段时间内不可用，可能需要移动群集主题定义。

关于此任务

在一个群集中，同一群组主题对象可以有多个定义。 对于主题主机路由群集来说，这是正常状态，而对于直接路由群集来说，这是不正常的状态。 更多信息，请参阅同名的多个群集主题定义。

要将群集主题定义移动到群集中的不同队列管理器而不中断出版流，请完成以下步骤。 该过程将一个定义从队列管理器 QM1 移至队列管理器 QM2。

过程

1. 在 QM2 上复制群组主题定义。

   对于直接路由，设置所有属性以符合 QM1 的定义。

   对于主题主机路由，首先将新主题主机定义为 "PUB(DISABLED)。 这样，QM2 就能了解群集中的订阅情况，但不会开始路由发布。
2. 等待信息通过群集传播。

   等待新的群集主题定义由完整存储库队列管理器传播到群集中的所有队列管理器。 使用 "**DISPLAY CLUSTER** 命令显示每个群组成员的群组主题，并检查是否有来自 QM2 的定义。

   对于主题主机路由，等待 QM2 上的新主题主机得知所有订阅。 比较 QM2 和 QM1 已知的代理订阅。 查看队列管理器上代理订阅的一种方法是发出以下 "**runmqsc** 命令：

   ```
   DISPLAY SUB(*) SUBTYPE(PROXY)
   ```
3. 对于主题主机路由，将 QM2 上的主题主机重新定义为 "PUB(ENABLED)，然后将 QM1 上的主题主机重新定义为 "PUB(DISABLED)。

   现在，QM2 上的新主题主机已经知道了其他队列管理器上的所有订阅，主题主机可以开始路由出版物。

   通过使用 PUB(DISABLED)设置来停止通过 QM1 的报文流量，可以确保在删除群集主题定义时，QM1 中没有正在运行的出版物。
4. 删除 QM1 中的群组主题定义。

   只有当队列管理器可用时，才能从 QM1 中删除定义。 否则，在重新启动或强行删除 QM1 之前，必须同时运行这两个定义。

   If QM1 remains unavailable for a long time, and during that time you need to modify the clustered topic definition on QM2, the QM2 definition is newer than the QM1 definition, and therefore usually prevails.

在此期间，如果 QM1 和 QM2 上的定义存在差异，则会将错误写入两个队列管理器的错误日志，提醒您注意冲突的群集主题定义。

如果 QM1 永远不会返回群集，例如因为硬件故障而意外退役，作为最后手段，您可以使用 RESET CLUSTER 命令强制弹出队列管理器。 **RESET CLUSTER** 会自动删除目标队列管理器上托管的所有主题对象。

相关概念

组合发布和预订作用域
从 IBM WebSphere MQ 7.0 开始，发布和预订作用域独立工作以确定队列管理器之间的发布流。

在发布/预订网络中组合主题空间
将队列管理器的主题空间与发布/预订集群或层次结构中的其他队列管理器组合在一起。 将发布/预订集群以及发布/预订集群与层次结构组合在一起。

相关任务

配置发布/预订集群
在队列管理器上定义主题。 要使主题成为集群主题，请设置 **CLUSTER** 属性。 要选择要用于此主题的发布和预订的路由，请设置 **CLROUTE** 属性。

向主题主机路由集群添加额外的主题主机
在主题主机路由的发布/预订集群中，可以使用多个队列管理器，通过在这些队列管理器上定义相同的集群主题对象，将发布路由到预订。 这可用于提高可用性和工作负载均衡。 为同一集群主题对象添加额外的主题主机时，可以使用 **PUB** 参数来控制何时开始通过新主题主机路由发布。

将队列管理器连接到发布/预订层次结构
将子队列管理器连接到层次结构中的父队列管理器。 如果子队列管理器已经是另一个层次结构或集群的成员，那么此连接会将层次结构连接在一起，或者将集群连接到层次结构。

将队列管理器与发布/预订层次结构断开连接
将子队列管理器与发布/预订层次结构中的父队列管理器断开连接。

向主题主机路由集群添加额外的主题主机

在主题主机路由的发布/预订集群中，可以使用多个队列管理器，通过在这些队列管理器上定义相同的集群主题对象，将发布路由到预订。 这可用于提高可用性和工作负载均衡。 为同一集群主题对象添加额外的主题主机时，可以使用 **PUB** 参数来控制何时开始通过新主题主机路由发布。

开始之前

在多个队列管理器上定义相同的集群主题对象仅在功能上对主题主机路由集群有用。 在直接路由集群中定义多个匹配主题不会更改其行为。 此任务仅适用于主题主机路由的集群。

此任务假定您已阅读文章 多个同名集群主题定义，特别是以下部分:

- 主题主机路由集群中的多个集群主题定义
- PUB 参数的特殊处理

关于此任务

当队列管理器成为路由主题主机时，它必须首先了解集群中已预订的所有相关主题是否存在。 如果在添加其他主题主机时将发布内容发布到这些主题，并且在该主机得知集群中的其他队列管理器上存在预订之前将发布内容路由到新主机，那么新主机不会将该发布内容转发到这些预订。 这会导致预订不命中发布。

发布不会通过已显式将集群主题对象 **PUB** 参数设置为 DISABLED 的主题主机队列管理器进行路由，因此您可以使用此设置来确保在添加额外主题主机的过程中没有预订未命中发布。

注: 当队列管理器托管已定义为 PUB(DISABLED)的集群主题时，连接到该队列管理器的发布程序无法发布消息，并且该队列管理器上的匹配预订不会接收在集群中的其他队列管理器上发布的发布。 因此，必须仔细考虑在存在预订和发布应用程序连接的队列管理器上定义主题主机路由主题。

过程

1. 配置新主题主机，并将新主题主机初始定义为 PUB(DISABLED)。

   这允许新主题主机了解集群中的预订，但不允许开始路由发布。

   有关配置主题主机的信息，请参阅 第 356 页的『配置发布/预订集群』。
2. 确定新主题主机何时已了解所有预订。

   为此，请将新主题主机已知的代理预订与现有主题主机已知的代理预订进行比较。 查看代理预订的一种方法是发出以下 **runmqsc** 命令: DISPLAY SUB(*) SUBTYPE(PROXY)
3. 将新主题主机重新定义为 PUB(ENABLED)。

   在新主题主机了解其他队列管理器上的所有预订之后，该主题可以开始路由发布。

相关概念

组合发布和预订作用域
从 IBM WebSphere MQ 7.0 开始，发布和预订作用域独立工作以确定队列管理器之间的发布流。

在发布/预订网络中组合主题空间
将队列管理器的主题空间与发布/预订集群或层次结构中的其他队列管理器组合在一起。 将发布/预订集群以及发布/预订集群与层次结构组合在一起。

相关任务

配置发布/预订集群
在队列管理器上定义主题。 要使主题成为集群主题，请设置 **CLUSTER** 属性。 要选择要用于此主题的发布和预订的路由，请设置 **CLROUTE** 属性。

将群集主题定义移至不同的队列管理器
对于主题主机路由或直接路由群集，在停用队列管理器时，或由于群集队列管理器发生故障或在相当长一段时间内不可用，可能需要移动群集主题定义。

将队列管理器连接到发布/预订层次结构
将子队列管理器连接到层次结构中的父队列管理器。 如果子队列管理器已经是另一个层次结构或集群的成员，那么此连接会将层次结构连接在一起，或者将集群连接到层次结构。

将队列管理器与发布/预订层次结构断开连接
将子队列管理器与发布/预订层次结构中的父队列管理器断开连接。

组合发布和预订作用域

从 IBM WebSphere MQ 7.0 开始，发布和预订作用域独立工作以确定队列管理器之间的发布流。

发布可以流向在发布/预订拓扑中连接的所有队列管理器，也可以仅流向本地队列管理器。 同样适用于代理预订。 与预订匹配的发布由这两个流的组合管理。

出版物和预订都可以限定为 QMGR 或 ALL。 如果发布者和订户都连接到同一队列管理器，那么作用域设置不会影响订户从该发布者接收到的发布。

如果发布者和订户连接到不同的队列管理器，那么这两个设置都必须为 ALL 才能接收远程发布。

假设发布程序连接到不同的队列管理器。 如果希望订户接收来自任何发布者的发布，请将预订作用域设置为 ALL。 然后，对于每个发布者，您可以决定是否将其发布的范围限制为发布者本地的订户。

假设订户连接到不同的队列管理器。 如果您希望将来自发布者的发布发送到所有订户，请将发布范围设置为 ALL。 如果您希望订户仅从连接到同一队列管理器的发布程序接收发布，请将预订作用域设置为 QMGR。

示例: 足球结果服务

假设你是足球联赛的成员球队。 每个团队都有一个队列管理器连接到发布/预订集群中的所有其他团队。

各代表队使用主题 Football/result/*Home team name/Away team name* 公布在主场进行的所有比赛的结果。 斜体字中的字符串是变量主题名称，而发布是匹配的结果。

每个俱乐部还使用主题字符串 Football/myteam/*Home team name/Away team name* 重新发布仅针对俱乐部的结果。

这两个主题都将发布到整个集群。

联盟设置了以下订阅，以便任何球队的球迷都可以通过三种有趣的方式订阅结果。

请注意，您可以使用 SUBSCOPE(QMGR)设置集群主题。 主题定义将传播到集群的每个成员，但预订的作用域只是本地队列管理器。 因此，每个队列管理器上的订户从同一预订接收不同的发布。

接收所有结果

```
DEFINE TOPIC(A) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(ALL)
```

接收所有主目录结果

```
DEFINE TOPIC(B) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(QMGR)
```

由于预订具有 QMGR 作用域，因此仅会匹配在主目录中发布的结果。

接收我的所有团队结果

```
DEFINE TOPIC(C) TOPICSTR('Football/myteam/') CLUSTER SUBSCOPE(QMGR)
```

由于预订具有 QMGR 作用域，因此仅匹配本地重新发布的本地团队结果。

相关概念

在发布/预订网络中组合主题空间
将队列管理器的主题空间与发布/预订集群或层次结构中的其他队列管理器组合在一起。 将发布/预订集群以及发布/预订集群与层次结构组合在一起。

分布式发布/预订网络

发布作用域

预订作用域

相关任务

配置发布/预订集群
在队列管理器上定义主题。 要使主题成为集群主题，请设置 **CLUSTER** 属性。 要选择要用于此主题的发布和预订的路由，请设置 **CLROUTE** 属性。

将群集主题定义移至不同的队列管理器
对于主题主机路由或直接路由群集，在停用队列管理器时，或由于群集队列管理器发生故障或在相当长一段时间内不可用，可能需要移动群集主题定义。

向主题主机路由集群添加额外的主题主机
在主题主机路由的发布/预订集群中，可以使用多个队列管理器，通过在这些队列管理器上定义相同的集群主题对象，将发布路由到预订。 这可用于提高可用性和工作负载均衡。 为同一集群主题对象添加额外的主题主机时，可以使用 **PUB** 参数来控制何时开始通过新主题主机路由发布。

将队列管理器连接到发布/预订层次结构
将子队列管理器连接到层次结构中的父队列管理器。 如果子队列管理器已经是另一个层次结构或集群的成员，那么此连接会将层次结构连接在一起，或者将集群连接到层次结构。

将队列管理器与发布/预订层次结构断开连接
将子队列管理器与发布/预订层次结构中的父队列管理器断开连接。

在发布/预订网络中组合主题空间

将队列管理器的主题空间与发布/预订集群或层次结构中的其他队列管理器组合在一起。 将发布/预订集群以及发布/预订集群与层次结构组合在一起。

您可以使用 **CLUSTER**， **PUBSCOPE** 和 **SUBSCOPE** 属性，发布/预订集群以及发布/预订层次结构的构建块来创建不同的发布/预订主题空间。

从从单个队列管理器扩展到发布/预订集群的示例开始，以下场景说明了不同的发布/预订拓扑。

相关概念

组合发布和预订作用域
从 IBM WebSphere MQ 7.0 开始，发布和预订作用域独立工作以确定队列管理器之间的发布流。

分布式发布/预订网络
主题空间

相关任务

配置发布/预订集群
在队列管理器上定义主题。 要使主题成为集群主题，请设置 **CLUSTER** 属性。 要选择要用于此主题的发布和预订的路由，请设置 **CLROUTE** 属性。

将群集主题定义移至不同的队列管理器
对于主题主机路由或直接路由群集，在停用队列管理器时，或由于群集队列管理器发生故障或在相当长一段时间内不可用，可能需要移动群集主题定义。

向主题主机路由集群添加额外的主题主机
在主题主机路由的发布/预订集群中，可以使用多个队列管理器，通过在这些队列管理器上定义相同的集群主题对象，将发布路由到预订。 这可用于提高可用性和工作负载均衡。 为同一集群主题对象添加额外的主题主机时，可以使用 **PUB** 参数来控制何时开始通过新主题主机路由发布。

将队列管理器连接到发布/预订层次结构
将子队列管理器连接到层次结构中的父队列管理器。 如果子队列管理器已经是另一个层次结构或集群的成员，那么此连接会将层次结构连接在一起，或者将集群连接到层次结构。

将队列管理器与发布/预订层次结构断开连接
将子队列管理器与发布/预订层次结构中的父队列管理器断开连接。

定义集群主题

在发布/预订集群中创建单个主题空间

扩展发布/预订系统以在多个队列管理器上运行。 使用发布/预订集群为每个发布者和订户提供单个相同的主题空间。

开始之前

您已在单个版本 7 队列管理器上实现发布/预订系统。

始终使用自己的根主题创建主题空间，而不是依赖继承 SYSTEM.BASE.TOPIC 的属性。 如果将发布/预订系统扩展至集群，那么可以在集群主题主机上将根主题定义为集群主题，然后在整个集群中共享所有主题。

关于此任务

现在，您希望扩展系统以支持更多发布者和订户，并在整个集群中显示每个主题。

过程

1. 创建要与发布/预订系统配合使用的集群。
   如果您具有现有传统集群，那么出于性能原因，最好为新的发布预订系统设置新集群。 可以将相同的服务器用于两个集群的集群存储库
2. 选择一个队列管理器 (可能是其中一个存储库) 作为集群主题主机。
3. 确保要在整个发布/预订集群中显示的每个主题都解析为管理主题对象。
   设置用于命名发布/预订集群的 **CLUSTER** 属性。

下一步做什么

将发布程序和订户应用程序连接到集群中的任何队列管理器。

创建具有 **CLUSTER** 属性的管理主题对象。 还会在整个集群中传播这些主题。 发布程序和订户程序使用管理主题，以便不会通过连接到集群中的不同队列管理器来改变其行为

如果需要 SYSTEM.BASE.TOPIC 在每个队列管理器上充当集群主题，那么需要在每个队列管理器上对其进行修改。

相关概念

分布式发布/预订网络
主题空间

相关任务

将 IBM WebSphere MQ 7 或更高版本的队列管理器添加到现有 IBM WebSphere MQ 6 主题空间
扩展现有 IBM WebSphere MQ 6 发布/预订系统以与 IBM WebSphere MQ 7 或更高版本的队列管理器进行互操作，共享相同的主题空间。

组合多个集群的主题空间
创建跨多个集群的主题空间。 发布到一个集群中的主题，然后在另一个集群中预订该主题。

组合和隔离多个集群中的主题空间
将某些主题空间与特定集群隔离，并组合其他主题空间以使它们在所有已连接的集群中都可访问。

发布和预订多个集群中的主题空间
使用重叠的集群发布和预订多个集群中的主题。 只要集群中的主题空间不重叠，就可以使用此方法。

定义集群主题

将 *IBM WebSphere MQ 7* 或更高版本的队列管理器添加到现有 *IBM WebSphere MQ 6* 主题空间

扩展现有 IBM WebSphere MQ 6 发布/预订系统以与 IBM WebSphere MQ 7 或更高版本的队列管理器进行互操作，共享相同的主题空间。

开始之前

您具有现有的 IBM WebSphere MQ 6 发布/预订系统。

您已在新服务器上安装 IBM WebSphere MQ 7 或更高版本，并配置了队列管理器。

关于此任务

您希望扩展现有 IBM WebSphere MQ 6 发布/预订系统以使用 IBM WebSphere MQ 7 或更高版本的队列管理器。

您已决定稳定使用排队的发布/预订接口的 IBM WebSphere MQ 6 发布/预订系统的开发。 您打算使用 IBM WebSphere MQ 7 或更高版本的 MQI 向系统添加扩展。 您现在没有重写已排队的发布/预订应用程序的计划。

您打算将来将 IBM WebSphere MQ 6 队列管理器升级到 IBM WebSphere MQ 7 或更高版本。 现在，您将继续在 IBM WebSphere MQ 7 或更高版本的队列管理器上运行现有已排队的发布/预订应用程序。

过程

1. 创建一组发送方/接收方通道，以将 IBM WebSphere MQ 7 或更高版本的队列管理器与双向的其中一个 IBM WebSphere MQ 6 队列管理器连接。
2. 使用目标队列管理器的名称创建两个传输队列。 如果由于某种原因无法使用目标队列管理器的名称作为传输队列名称，请使用队列管理器别名。
3. 配置传输队列以触发发送方通道。
4. 如果 IBM WebSphere MQ 6 发布/预订系统使用流，请将这些流添加到 IBM WebSphere MQ 7 或更高版本的队列管理器，如 第 354 页的『添加流』中所述。
5. 检查 IBM WebSphere MQ 7 或更高版本的队列管理器 **PSMODE** 是否设置为 ENABLE。
6. 更改其 **PARENT** 属性以引用其中一个 IBM WebSphere MQ 6 队列管理器。
7. 检查队列管理器之间的父子关系在两个方向上都处于活动状态。

下一步做什么

完成该任务后， IBM WebSphere MQ 6 和 IBM WebSphere MQ 7 或更高版本的队列管理器将共享相同的主题空间。 例如，您可以执行以下所有任务。

- 在 IBM WebSphere MQ 6 与 IBM WebSphere MQ 7 或更高版本的队列管理器之间交换发布和预订。
- 在 IBM WebSphere MQ 7 或更高版本的队列管理器上运行现有 IBM WebSphere MQ 6 发布/预订程序。
- 在 IBM WebSphere MQ 6 或 IBM WebSphere MQ 7 或更高版本的队列管理器上查看和修改主题空间。

- 编写 IBM WebSphere MQ 7 或更高版本的发布/预订应用程序，并在 IBM WebSphere MQ 7 或更高版本的队列管理器上运行这些应用程序。
- 使用 IBM WebSphere MQ 7 或更高版本的应用程序创建新的发布和预订，并与 IBM WebSphere MQ 6 应用程序交换这些发布和预订。

相关概念

分布式发布/预订网络

主题空间

相关任务

在发布/预订集群中创建单个主题空间
扩展发布/预订系统以在多个队列管理器上运行。 使用发布/预订集群为每个发布者和订户提供单个相同的主题空间。

组合多个集群的主题空间
创建跨多个集群的主题空间。 发布到一个集群中的主题，然后在另一个集群中预订该主题。

组合和隔离多个集群中的主题空间
将某些主题空间与特定集群隔离，并组合其他主题空间以使它们在所有已连接的集群中都可访问。

发布和预订多个集群中的主题空间
使用重叠的集群发布和预订多个集群中的主题。 只要集群中的主题空间不重叠，就可以使用此方法。

定义集群主题

组合多个集群的主题空间

创建跨多个集群的主题空间。 发布到一个集群中的主题，然后在另一个集群中预订该主题。

开始之前

此任务假定您具有现有的直接路由发布/预订集群，并且要将某些集群主题传播到所有集群中。

注: 不能对主题主机路由的发布/预订集群执行此操作。

关于此任务

要将发布从一个集群传播到另一个集群，您需要在层次结构中将这些集群连接在一起; 请参阅 第 365 页的图 67。 分层连接在已连接的队列管理器之间传播预订和发布，集群在每个集群中传播集群主题，而不是在集群之间传播集群主题。

这两种机制的组合将在所有集群之间传播集群主题。 您需要在每个集群中重复集群主题定义。

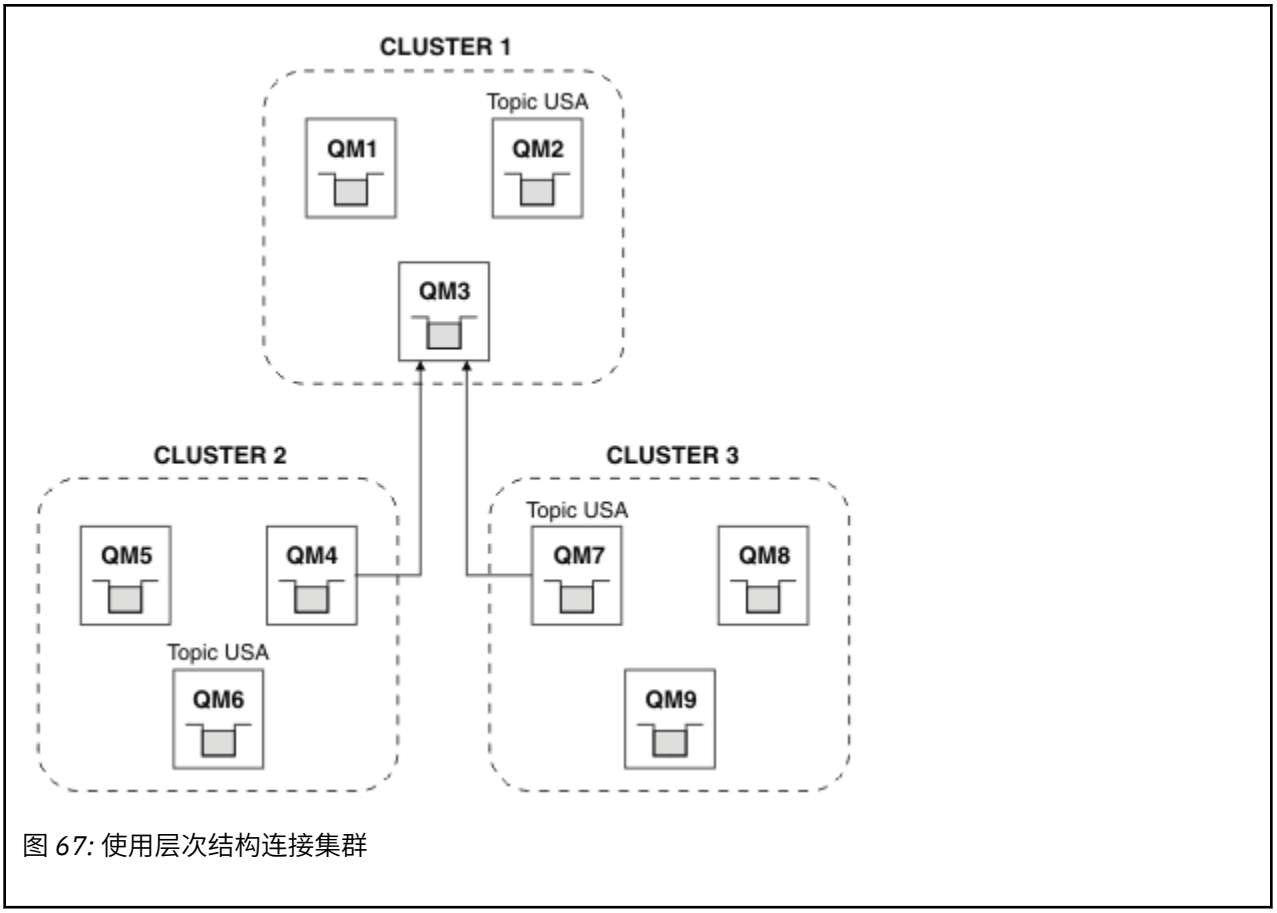


图 *67:* 使用层次结构连接集群

以下步骤将集群连接到层次结构中。

过程

1. 创建两组发送方/接收方通道以双向连接 QM3 和 QM4 以及 QM3 和 QM7。 必须使用传统发送方/接收方通道和传输队列 (而不是集群) 来连接层次结构。
2. 使用目标队列管理器的名称创建三个传输队列。 如果由于某种原因无法使用目标队列管理器的名称作为传输队列名称，请使用队列管理器别名。
3. 配置传输队列以触发发送方通道。
4. 检查 QM3 的 **PSMODE** ， QM4 和 QM7 是否设置为 ENABLE。
5. 将 QM4 和 QM7 的 **PARENT** 属性变更为 QM3。
6. 检查队列管理器之间的父子关系在两个方向上都处于活动状态。
7. 使用属性 **CLUSTER** ('CLUSTER 1') ， **CLUSTER** ('CLUSTER 2') 和 **CLUSTER** ('CLUSTER 3') 创建管理主题 USA 在集群 1 ， 2 和 3 中的三个集群主题主机队列管理器中的每一个上。 集群主题主机不需要是分层连接的队列管理器。

下一步做什么

现在，您可以在 第 365 页的图 67 中发布或预订集群主题 USA 。 发布预订流向所有三个集群中的发布者和订户。

假定您未在其他集群中创建 USA 作为集群主题。 如果仅在 QM7 上定义 USA ，那么将在 QM7， QM8， QM9 和 QM3 之间交换 USA 的发布和预订。 在 QM7 QM8 上运行的发布者和订户 QM9 将继承管理主题 USA 的属性。 QM3 上的发布者和订户将继承 QM3 上的 SYSTEM.BASE.TOPIC 的属性。

另请参阅第 366 页的『组合和隔离多个集群中的主题空间』。

相关概念

分布式发布/预订网络

主题空间

相关任务

在发布/预订集群中创建单个主题空间
扩展发布/预订系统以在多个队列管理器上运行。 使用发布/预订集群为每个发布者和订户提供单个相同的主题空间。

将 IBM WebSphere MQ 7 或更高版本的队列管理器添加到现有 IBM WebSphere MQ 6 主题空间
扩展现有 IBM WebSphere MQ 6 发布/预订系统以与 IBM WebSphere MQ 7 或更高版本的队列管理器进行互操作，共享相同的主题空间。

组合和隔离多个集群中的主题空间
将某些主题空间与特定集群隔离，并组合其他主题空间以使它们在所有已连接的集群中都可访问。

发布和预订多个集群中的主题空间
使用重叠的集群发布和预订多个集群中的主题。 只要集群中的主题空间不重叠，就可以使用此方法。

定义集群主题

组合和隔离多个集群中的主题空间

将某些主题空间与特定集群隔离，并组合其他主题空间以使它们在所有已连接的集群中都可访问。

开始之前

检查主题 第 364 页的『组合多个集群的主题空间』。 这可能足以满足您的需求，而无需添加额外的队列管理器作为网桥。

注: 只能使用直接路由的发布/预订集群来完成此任务。 不能使用主题主机路由集群来执行此操作。

关于此任务

第 364 页的『组合多个集群的主题空间』 中的 第 365 页的图 67 中显示的拓扑的潜在改进是隔离未在所有集群之间共享的集群主题。 通过创建不在任何集群中的桥接队列管理器来隔离集群; 请参阅 第 367 页的图 68。 使用桥接队列管理器来过滤哪些发布和预订可以从一个集群流向另一个集群。

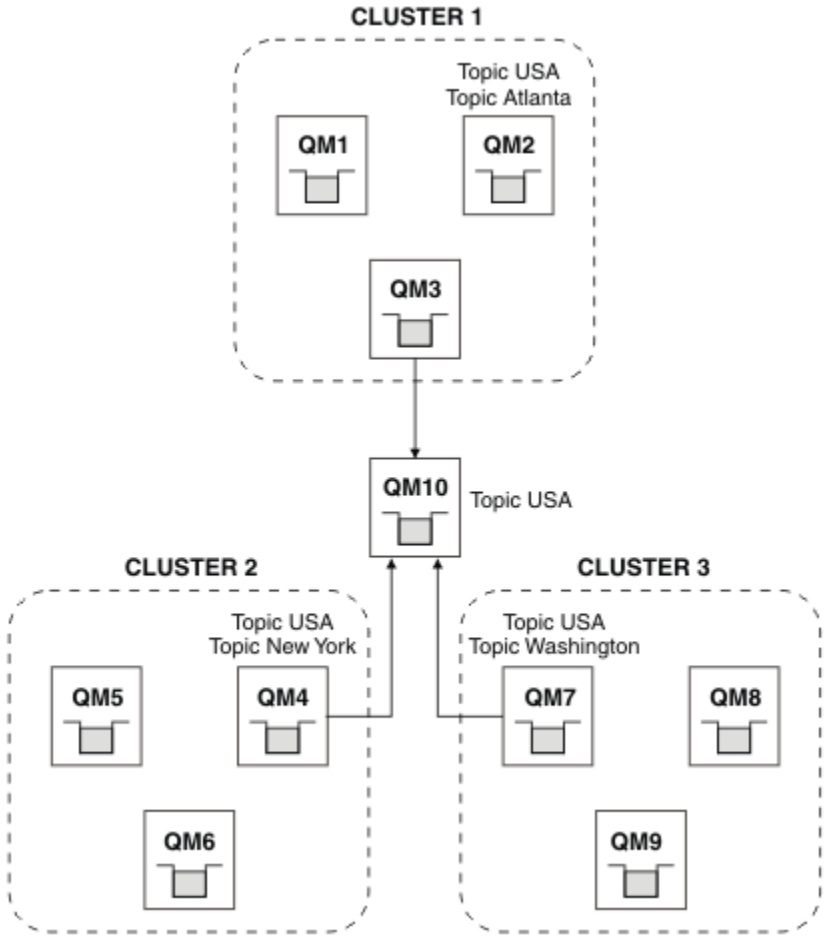


图 *68:* 桥接集群

使用网桥来隔离您不希望在其他集群上的网桥中公开的集群主题。 在 第 367 页的图 68 中， USA 是所有集群中共享的集群主题， Atlanta， New York 和 Washington 是仅在一个集群中共享的集群主题。

使用以下过程对配置进行建模:

过程

1. 将所有 SYSTEM.BASE.TOPIC 主题对象修改为具有 **SUBSCOPE** (QMGR) 和 **PUBSCOPE** (QMGR) 在所有队列管理器上。

   除非显式设置 **SUBSCOPE** (ALL) ， 否则不会将任何主题 (甚至集群主题) 传播到其他队列管理器 和 **PUBSCOPE** (ALL) 集群主题的根主题。
2. 使用属性 **CLUSTER** (*clustername*) ， **SUBSCOPE** (ALL) 定义要在每个集群中共享的三个集群主题主机队列管理器上的主题 和 **PUBSCOPE** (ALL)。

   如果要在所有集群之间共享某些集群主题，请在每个集群中定义相同的主题。 使用每个集群的集群名称作为集群属性。
3. 对于要在所有集群之间共享的集群主题，请在网桥队列管理器 (QM10) 上使用属性 **SUBSCOPE** (ALL) 和 **PUBSCOPE** (ALL) 再次定义这些主题。

示例

在 第 367 页的图 68 中的示例中，仅从 USA 继承的主题在所有三个集群之间传播。

下一步做什么

使用 **SUBSCOPE** (ALL) 在网桥队列管理器上定义的主题的预订 和 **PUBSCOPE** (ALL) 在集群之间传播。

每个集群中使用属性 **CLUSTER** (*clustername*) 和 **SUBSCOPE** (ALL) 定义的主题的预订 和 **PUBSCOPE** (ALL) 在每个集群中传播。

任何其他预订都是队列管理器的本地预订。

相关概念

分布式发布/预订网络

主题空间

发布作用域

预订作用域

相关任务

在发布/预订集群中创建单个主题空间

扩展发布/预订系统以在多个队列管理器上运行。 使用发布/预订集群为每个发布者和订户提供单个相同的主题空间。

将 IBM WebSphere MQ 7 或更高版本的队列管理器添加到现有 IBM WebSphere MQ 6 主题空间

扩展现有 IBM WebSphere MQ 6 发布/预订系统以与 IBM WebSphere MQ 7 或更高版本的队列管理器进行互操作，共享相同的主题空间。

组合多个集群的主题空间

创建跨多个集群的主题空间。 发布到一个集群中的主题，然后在另一个集群中预订该主题。

发布和预订多个集群中的主题空间

使用重叠的集群发布和预订多个集群中的主题。 只要集群中的主题空间不重叠，就可以使用此方法。

定义集群主题

发布和预订多个集群中的主题空间

使用重叠的集群发布和预订多个集群中的主题。 只要集群中的主题空间不重叠，就可以使用此方法。

开始之前

在集群之间的交集中使用一些队列管理器创建多个传统集群。

关于此任务

由于各种不同的原因，您可能选择了重叠集群。

1. 您有数量有限的高可用性服务器或队列管理器。 您决定将所有集群存储库以及集群主题主机部署到这些存储库。
2. 您具有使用网关队列管理器进行连接的现有传统队列管理器集群。 您希望将发布/预订应用程序部署到同一集群拓扑。
3. 您有几个自包含的发布/预订应用程序。 出于性能原因，最好保持发布/预订集群较小且与传统集群分开。 您已决定将应用程序部署到不同的集群。 但是，您还希望在一个队列管理器上监视所有发布/预订应用程序，因为您只许可了该监视应用程序的一个副本。 此队列管理器必须有权访问所有集群中的集群主题的发布。

通过确保在非重叠主题空间中定义主题，可以将主题部署到重叠的发布/预订集群中，请参阅 第 369 页的图 69。 如果主题空间重叠，那么部署到重叠的集群会导致问题。

由于发布/预订集群重叠，因此您可以使用重叠中的队列管理器来发布和预订任何主题空间。

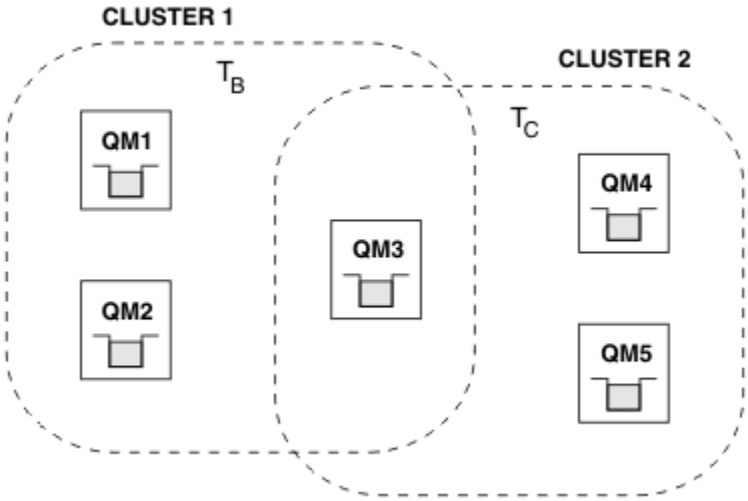


图 69: 重叠集群，非重叠主题空间

过程

创建确保主题空间不重叠的方法。

例如，为每个主题空间定义唯一的根主题。 使根主题成为集群主题。

a) `DEFINE TOPIC(B) TOPICSTR('B') CLUSTER('CLUSTER 1') ...`

b) `DEFINE TOPIC(C) TOPICSTR('C') CLUSTER('CLUSTER 2') ...`

示例

在 第 369 页的图 69 中，连接到 QM3 的发布者和订户可以发布或预订 T_B 或 T_C

下一步做什么

将两个集群中使用主题的发布者和订户连接到重叠中的队列管理器。

将只能使用特定集群中的主题的发布者和订户连接到不在重叠中的队列管理器。

相关概念

分布式发布/预订网络

主题空间

相关任务

在发布/预订集群中创建单个主题空间

扩展发布/预订系统以在多个队列管理器上运行。 使用发布/预订集群为每个发布者和订户提供单个相同的主题空间。

将 IBM WebSphere MQ 7 或更高版本的队列管理器添加到现有 IBM WebSphere MQ 6 主题空间

扩展现有 IBM WebSphere MQ 6 发布/预订系统以与 IBM WebSphere MQ 7 或更高版本的队列管理器进行互操作，共享相同的主题空间。

组合多个集群的主题空间

创建跨多个集群的主题空间。 发布到一个集群中的主题，然后在另一个集群中预订该主题。

组合和隔离多个集群中的主题空间

将某些主题空间与特定集群隔离，并组合其他主题空间以使它们在所有已连接的集群中都可访问。

定义集群主题

将队列管理器连接到发布/预订层次结构

将子队列管理器连接到层次结构中的父队列管理器。 如果子队列管理器已经是另一个层次结构或集群的成员，那么此连接会将层次结构连接在一起，或者将集群连接到层次结构。

开始之前

1. 发布/预订层次结构中的队列管理器必须具有唯一的队列管理器名称。
2. 发布/预订层次结构依赖于 “已排队的发布/预订” 队列管理器功能。 必须在父队列管理器和子队列管理器上都启用此功能。 请参阅第 353 页的『正在启动排队的发布/预订』。
3. 发布/预订关系依赖于队列管理器发送方和接收方通道。 有两种方法可以建立通道:
   - 将父队列管理器和子队列管理器都添加到 IBM MQ 集群。 请参阅 第 254 页的『将队列管理器添加至集群』。
   - 建立从子队列管理器到父队列管理器以及从父队列管理器到子队列管理器的发送方/接收方通道对。 每个通道都需要使用与目标队列管理器同名的传输队列，或者使用与目标队列管理器同名的队列管理器别名。 有关如何建立点到点通道连接的更多信息，请参阅 第 153 页的『IBM MQ 分布式排队技术』。

   有关在每种类型的通道配置上配置层次结构的示例，请参阅以下一组发布/预订层次结构方案:
   - 场景 1: 使用具有队列管理器名称别名的点到点通道
   - 场景 2: 将同名的点到点通道用于传输队列和远程队列管理器
   - 场景 3: 使用集群通道添加队列管理器

关于此任务

使用 ALTER QMGR PARENT （*PARENT_NAME*）**runmqsc** 命令将子代连接到父代。 此配置在子队列管理器上执行，其中 *PARENT_NAME* 是父队列管理器的名称。

过程

ALTER QMGR PARENT (*PARENT_NAME*)

示例

第一个示例显示如何将队列管理器 QM2 作为 QM1 的子代进行连接，然后查询 QM2 以确认其已成功成为 **STATUS** 为 ACTIVE 的子代:

```
C:>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2
alter qmgr parent(QM1)
     1 : alter qmgr parent(QM1)
AMQ8005: IBM MQ queue manager changed.
display pubsub all
     2 : display pubsub all
AMQ8723: Display pub/sub status details.
   QMNAME(QM2)                             TYPE(LOCAL)
   STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
   QMNAME(QM1)                             TYPE(PARENT)
STATUS(ACTIVE)
```

下一个示例显示了查询 QM1 以获取其连接的结果:

```
C:\Documents and Settings\Admin>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
display pubsub all
     2 : display pubsub all
AMQ8723: Display pub/sub status details.
   QMNAME(QM1)                             TYPE(LOCAL)
   STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
```

```
    QMNAME(QM2)                             TYPE(CHILD)
    STATUS(ACTIVE)
```

如果 **STATUS** 未显示为 ACTIVE，请检查子代与父代之间的通道是否已正确配置并正在运行。 请检查这两个队列管理器错误日志以获取可能的错误。

下一步做什么

缺省情况下，一个队列管理器上的发布者和订户所使用的主题与层次结构中其他队列管理器上的发布者和订户共享。 可以将受管主题配置为通过使用 **SUBSCOPE** 和 **PUBSCOPE** 主题属性来控制共享级别。 请参阅第 356 页的『配置分布式发布/预订网络』。

相关概念

组合发布和预订作用域

从 IBM WebSphere MQ 7.0 开始，发布和预订作用域独立工作以确定队列管理器之间的发布流。

在发布/预订网络中组合主题空间

将队列管理器的主题空间与发布/预订集群或层次结构中的其他队列管理器组合在一起。 将发布/预订集群以及发布/预订集群与层次结构组合在一起。

流和主题

发布/预订消息传递

相关任务

配置发布/预订集群

在队列管理器上定义主题。 要使主题成为集群主题，请设置 **CLUSTER** 属性。 要选择要用于此主题的发布和预订的路由，请设置 **CLROUTE** 属性。

将群集主题定义移至不同的队列管理器

对于主题主机路由或直接路由群集，在停用队列管理器时，或由于群集队列管理器发生故障或在相当长一段时间内不可用，可能需要移动群集主题定义。

向主题主机路由集群添加额外的主题主机

在主题主机路由的发布/预订集群中，可以使用多个队列管理器，通过在这些队列管理器上定义相同的集群主题对象，将发布路由到预订。 这可用于提高可用性和工作负载均衡。 为同一集群主题对象添加额外的主题主机时，可以使用 **PUB** 参数来控制何时开始通过新主题主机路由发布。

将队列管理器与发布/预订层次结构断开连接

将子队列管理器与发布/预订层次结构中的父队列管理器断开连接。

相关参考

显示发布预订

将队列管理器与发布/预订层次结构断开连接

将子队列管理器与发布/预订层次结构中的父队列管理器断开连接。

关于此任务

使用 **ALTER QMGR** 命令将队列管理器与代理层次结构断开连接。 您可以随时按任何顺序断开队列管理器的连接。

当队列管理器之间的连接正在运行时，将发送更新父代的相应请求。

过程

```
ALTER QMGR PARENT( '')
```

示例

```
C:\Documents and Settings\Admin>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2.
    1 : alter qmgr parent('')
AMQ8005: IBM MQ queue manager changed.
    2 : display pubsub type(child)
AMQ8147: IBM MQ object  not found.
```

```
display pubsub type(parent)
    3 : display pubsub type(parent)
AMQ8147: IBM MQ object not found.
```

下一步做什么

您可以删除不再需要的任何流，队列和手动定义的通道。

相关概念

组合发布和预订作用域

从 IBM WebSphere MQ 7.0 开始，发布和预订作用域独立工作以确定队列管理器之间的发布流。

在发布/预订网络中组合主题空间

将队列管理器的主题空间与发布/预订集群或层次结构中的其他队列管理器组合在一起。 将发布/预订集群以及发布/预订集群与层次结构组合在一起。

相关任务

配置发布/预订集群

在队列管理器上定义主题。 要使主题成为集群主题，请设置 **CLUSTER** 属性。 要选择要用于此主题的发布和预订的路由，请设置 **CLROUTE** 属性。

将群集主题定义移至不同的队列管理器

对于主题主机路由或直接路由群集，在停用队列管理器时，或由于群集队列管理器发生故障或在相当长一段时间内不可用，可能需要移动群集主题定义。

向主题主机路由集群添加额外的主题主机

在主题主机路由的发布/预订集群中，可以使用多个队列管理器，通过在这些队列管理器上定义相同的集群主题对象，将发布路由到预订。 这可用于提高可用性和工作负载均衡。 为同一集群主题对象添加额外的主题主机时，可以使用 **PUB** 参数来控制何时开始通过新主题主机路由发布。

将队列管理器连接到发布/预订层次结构

将子队列管理器连接到层次结构中的父队列管理器。 如果子队列管理器已经是另一个层次结构或集群的成员，那么此连接会将层次结构连接在一起，或者将集群连接到层次结构。

ALW 配置多个安装

在同一系统上使用多个安装时，必须配置安装和队列管理器。

关于此任务

此信息适用于 AIX, Linux, and Windows。

过程

- 使用以下链接中的信息来配置安装:
  - 第 379 页的『更改主安装』
  - 第 380 页的『使队列管理器与安装相关联』
  - 第 372 页的『在多安装环境中连接应用程序』

ALW 在多安装环境中连接应用程序

在 AIX, Linux, and Windows 系统上，如果装入了 IBM WebSphere MQ 7.1 或更高版本的库，那么 IBM MQ 将自动使用相应的库，而无需执行任何进一步的操作。 IBM MQ 使用与应用程序所连接的队列管理器相关联的安装中的库。

以下概念用于说明应用程序连接到 IBM MQ 的方式:

链接
编译应用程序时，应用程序将链接到 IBM MQ 库，以获取随后在应用程序运行时装入的函数导出。

正在装入

运行应用程序时，将找到并装入 IBM MQ 库。 用于查找库的特定机制因操作系统以及应用程序的构建方式而异。 有关如何在多安装环境中查找和装入库的更多信息，请参阅 第 374 页的『正在装入 IBM MQ 库』。

连接

当应用程序连接到正在运行的队列管理器 (例如，使用 MQCONN 或 MQCONNX 调用) 时，它将使用装入的 IBM MQ 库进行连接。

当服务器应用程序连接到队列管理器时，装入的库必须来自与队列管理器关联的安装。 对于系统上的多个安装，在选择操作系统用于查找要装入的 IBM MQ 库的机制时，此限制会引入新的挑战:

- 当 **setmqm** 命令用于更改与队列管理器关联的安装时，需要装入的库将更改。
- 当应用程序连接到由不同安装拥有的多个队列管理器时，需要装入多组库。

但是，如果找到并装入了 IBM WebSphere MQ 7.1 或更高版本的库，那么 IBM MQ 将装入并使用相应的库，而无需执行任何进一步操作。 当应用程序连接到队列管理器时， IBM MQ 将从与该队列管理器关联的安装装入库。

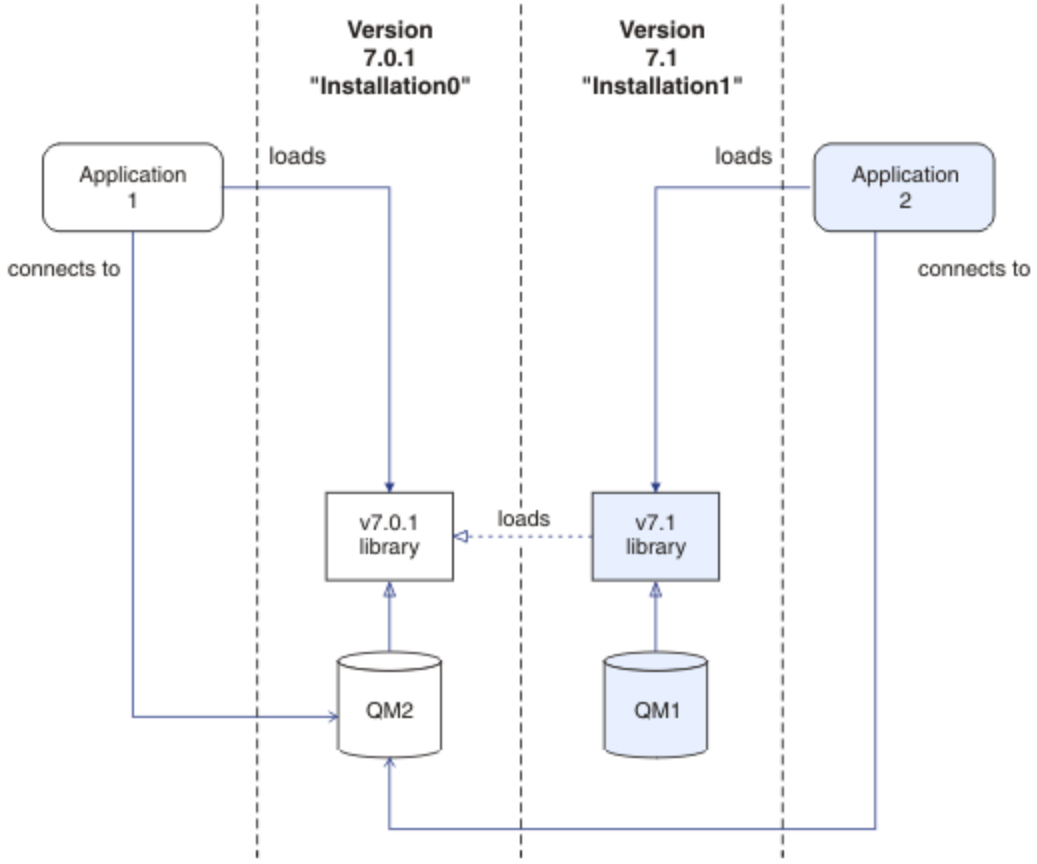


图 *70:* 在多安装环境中连接应用程序

例如， 第 373 页的图 70 显示了具有 IBM WebSphere MQ 7.0.1 安装 (Installation0) 和 IBM WebSphere MQ 7.1 安装 (Installation1) 的多安装环境。 两个应用程序连接到这些安装，但它们装入不同的库版本。

Application 1 直接装入 IBM WebSphere MQ 7.0.1 库。 当 application 1 连接到 QM2 时，将使用 IBM WebSphere MQ 7.0.1 库。 如果 application 1 尝试连接到 QM1，或者如果 QM2 与 Installation1 相关联，那么 application 1 将失败并返回 2059 (080B) (RC2059) :MQRC_Q_MGR_NOT_AVAILABLE 错误。 应用程序失败，因为 IBM WebSphere MQ 7.0.1 库无法装入其他库版本。 即，如果直接装入 IBM WebSphere MQ 7.0.1 库，那么不能使用与更高版本的 IBM MQ 的安装相关联的队列管理器。

Application 2 直接装入 IBM WebSphere MQ 7.1 库。 当 application 2 连接到 QM2 时， IBM WebSphere MQ 7.1 库将装入并使用 IBM WebSphere MQ 7.0.1 库。 如果 application 2 连接到 QM1，或者如果 QM2 与 Installation1 相关联，那么将装入 IBM WebSphere MQ 7.1 库，并且应用程序将按预期工作。

在 AIX, Linux, and Windows 上的多安装队列管理器共存中更详细地考虑迁移方案和连接具有多个安装的应用程序。

有关如何装入 IBM WebSphere MQ 7.1 库的更多信息，请参阅 第 374 页的『正在装入 IBM MQ 库』。

支持和限制

如果找到并装入了以下任何 IBM WebSphere MQ 7.1 或更高版本的库，那么产品可以自动装入并使用相应的库:

- C 服务器库
- C++ 服务器库
- XA 服务器库
- COBOL 服务器库
- COM + 服务器库
- .NET 处于非受管方式

IBM MQ 还会在绑定方式下自动装入并使用 Java 和 JMS 应用程序的相应库。

对于使用多个安装的应用程序，存在许多限制。 有关更多信息，请参阅 第 376 页的『使用多个安装的应用程序的限制』。

相关概念

第 376 页的『使用多个安装的应用程序的限制』
在多安装环境中使用 CICS 服务器库，快速路径连接，消息句柄和出口时存在一些限制。

第 374 页的『正在装入 IBM MQ 库』
在决定如何装入 IBM MQ 库时，需要考虑许多因素，包括: 您的环境，是否可以更改现有应用程序，是否需要主安装，安装 IBM MQ 的位置以及 IBM MQ 的位置是否可能更改。

相关任务

选择主安装

第 379 页的『更改主安装』
您可以使用 **setmqinst** 命令将安装设置或取消设置为主安装。

第 380 页的『使队列管理器与安装相关联』
创建队列管理器时，它将自动与发出 **crtmqm** 命令的安装相关联。 在 AIX, Linux, and Windows 上，可以使用 **setmqm** 命令来更改与队列管理器关联的安装。

ALW 正在装入 IBM MQ 库

在决定如何装入 IBM MQ 库时，需要考虑许多因素，包括: 您的环境，是否可以更改现有应用程序，是否需要主安装，安装 IBM MQ 的位置以及 IBM MQ 的位置是否可能更改。

此信息适用于 IBM WebSphere MQ 7.1 或更高版本的库。

IBM MQ 库的定位和装入方式取决于您的安装环境:

- 在 AIX and Linux 系统上，如果在缺省位置安装了 IBM WebSphere MQ 7.1 或更高版本的副本，那么现有应用程序将继续以与先前版本相同的方式工作。 但是，如果应用程序需要 /usr/lib 中的符号链接，那么必须选择 IBM WebSphere MQ 7.1 或更高版本的安装作为主安装，或者手动创建符号链接。
- 如果 IBM WebSphere MQ 7.1 或更高版本安装在非缺省位置 (如果还安装了 IBM WebSphere MQ 7.0.1)，那么您可能需要更改现有应用程序，以便装入正确的库。

如何查找和装入 IBM MQ 库还取决于如何设置任何现有应用程序以装入库。 有关如何装入库的更多信息，请参阅 第 376 页的『操作系统库装入机制』。

最好确保由操作系统装入的 IBM MQ 库是与队列管理器相关联的库。

用于装入 IBM MQ 库的方法因平台而异，并且每种方法都具有优点和缺点。

表 *29:* 装入库的选项的优点和缺点

平台	选项	优势	缺陷
Linux AIX AIX and Linux 系统	设置或更改应用程序的嵌入式运行时搜索路径 (RPath)。 此选项要求您重新编译并链接应用程序。 有关编译和链接应用程序的更多信息，请参阅 构建过程应用程序。	• 变化的范围是明确的。	• 您必须能够重新编译并链接应用程序。 • 如果 IBM MQ 的位置发生更改，那么必须更改 RPath。
AIX and Linux 系统	使用带有 **-k** 或 **-l** 选项的 setmqenv 或 crtmqenv 来设置 *LD_LIBRARY_PATH* 环境变量。 AIX 在 AIX 上，此环境变量为 *LIBPATH*	• 不需要更改现有应用程序。 • 覆盖应用程序中的嵌入式 RPaths。 • 易于更改变量 (如果 IBM MQ 的位置发生更改)。	• 出于安全原因， `setuid` 和 `setgid` 应用程序或以其他方式构建的应用程序可能会忽略 *LD_LIBRARY_PATH* 。 • 特定于环境，因此必须在运行应用程序的每个环境中进行设置。 • 对依赖于 *LD_LIBRARY_PATH* 的其他应用程序的可能影响。 • Linux Linux: 用于构建应用程序的编译器可能会禁用 *LD_LIBRARY_PATH*。 有关更多信息，请参阅 Linux 的运行时链接注意事项 。
Windows Windows 系统	使用 setmqenv 或 crtmqenv 设置 PATH 变量。	• 不需要对现有应用程序进行任何更改。 • 易于更改变量 (如果 IBM MQ 的位置发生更改)。	• 特定于环境，因此必须在运行应用程序的每个环境中进行设置。 • 可能对其他应用程序产生影响。
ALW AIX, Linux, and Windows 系统	将主安装设置为 IBM WebSphere MQ 7.1 或更高版本的安装。 请参阅 第 379 页的『更改主安装』。 有关主安装的更多信息，请参阅 选择主安装。	• 不需要对现有应用程序进行任何更改。 • 如果 IBM MQ 的位置发生更改，那么可轻松更改主安装。 • 给出与先前版本的 IBM MQ 相似的行为。	• 安装 IBM WebSphere MQ 7.0.1 时，无法将主安装设置为 IBM WebSphere MQ 7.1 或更高版本。 • Linux AIX AIX and Linux: 如果 `/usr/lib` 不在缺省搜索路径中，那么不起作用。

Linux 的库装入注意事项

Linux

使用某些版本的 gcc (例如， V 3.2.x) 编译的应用程序可以具有无法使用 *LD_LIBRARY_PATH* 环境变量覆盖的嵌入式 RPath。 您可以使用 `readelf -d applicationName` 命令来确定应用程序是否受影响。 如果 RPATH 符号存在且 RUNPATH 符号不存在，那么无法覆盖 RPath。

操作系统库装入机制

在 Windows 系统上，将搜索多个目录以查找库:

- 从中装入应用程序的目录。
- 当前目录。
- *PATH* 环境变量 (全局 *PATH* 变量和当前用户的 *PATH* 变量) 中的目录。

Linux AIX 在 AIX and Linux 系统上，可能使用了许多方法来查找要装入的库:

- 使用 *LD_LIBRARY_PATH* 环境变量 (也是 AIX 上的 库路径)。 如果设置了此变量，那么它将定义一组要搜索所需 IBM MQ 库的目录。 如果在这些目录中找到了任何库，那么首选使用可能使用其他方法找到的任何库。
- 使用嵌入式搜索路径 (RPath)。 应用程序可能包含一组用于搜索 IBM MQ 库的目录。 如果未设置 *LD_LIBRARY_PATH* ，或者如果未使用该变量找到必需的库，那么将在 RPath 中搜索这些库。 如果现有应用程序使用 RPath ，但无法重新编译和链接应用程序，那么必须在缺省位置安装 IBM WebSphere MQ 7.1 ，或者使用其他方法来查找库。
- 使用缺省库路径。 如果在搜索 *LD_LIBRARY_PATH* 变量和 RPath 位置后找不到 IBM MQ 库，那么将搜索缺省库路径。 通常，此路径包含 `/usr/lib` 或 `/usr/lib64`。 如果在搜索缺省库路径后找不到库，那么由于缺少依赖关系，应用程序无法启动。

您可以使用操作系统机制来了解应用程序是否具有嵌入式搜索路径。 例如:

- AIX AIX: **dump**
- Linux Linux: **readelf**

相关概念

第 376 页的『使用多个安装的应用程序的限制』
在多安装环境中使用 CICS 服务器库，快速路径连接，消息句柄和出口时存在一些限制。

第 372 页的『在多安装环境中连接应用程序』
在 AIX, Linux, and Windows 系统上，如果装入了 IBM WebSphere MQ 7.1 或更高版本的库，那么 IBM MQ 将自动使用相应的库，而无需执行任何进一步的操作。 IBM MQ 使用与应用程序所连接的队列管理器相关联的安装中的库。

相关任务

选择主安装
第 379 页的『更改主安装』
您可以使用 **setmqinst** 命令将安装设置或取消设置为主安装。

第 380 页的『使队列管理器与安装相关联』
创建队列管理器时，它将自动与发出 **crtmqm** 命令的安装相关联。 在 AIX, Linux, and Windows 上，可以使用 **setmqm** 命令来更改与队列管理器关联的安装。

ALW 使用多个安装的应用程序的限制

在多安装环境中使用 CICS 服务器库，快速路径连接，消息句柄和出口时存在一些限制。

CICS 服务器库

如果您正在使用 CICS 服务器库，那么 IBM MQ 不会自动为您选择正确的库级别。 您必须使用应用程序所连接的队列管理器的相应库级别来编译和链接应用程序。 有关更多信息，请参阅 构建库以用于 TXSeries for Multiplatforms V 5。

消息句柄

使用特殊值 MQHC_UNASSOCIATED_HCONN 的消息句柄限制为与进程中装入的第一个安装配合使用。 如果特定安装无法使用消息句柄，那么将返回原因码 MQRC_HMSG_NOT_AVAILABLE 。

此限制会影响消息属性。 不能使用消息句柄从一个安装上的队列管理器获取消息属性，并将它们放到另一个安装上的队列管理器。 有关消息句柄的更多信息，请参阅 MQCRTMH-创建消息句柄。

出口

在多安装环境中，必须更新现有出口以用于 IBM WebSphere MQ 7.1 或更高版本的安装。 必须使用更新后的命令重新生成使用 **crtmqcvx** 命令生成的数据转换出口。

所有出口都必须使用 MQIEP 结构编写，不能使用嵌入式 RPATH 来查找 IBM MQ 库，并且不能链接到 IBM MQ 库。 有关更多信息，请参阅 在 AIX, Linux, and Windows 上编写出口和可安装服务 。

快速路径

在具有多个安装的服务器上，使用与 IBM WebSphere MQ 7.1 或更高版本的快速路径连接的应用程序必须遵循以下规则：

1. 队列管理器必须与应用程序从中装入 IBM MQ 运行时库的同一个安装相关联。 应用程序不能使用与另一个安装关联的队列管理器的快速路径连接。 尝试建立连接会导致错误，原因码为 MQRC_INSTALLATION_MISMATCH。
2. 以非快速路径连接到与应用程序从中装入 IBM MQ 运行时库的同一个安装关联的队列管理器将阻止应用程序以快速路径进行连接，除非以下任一情况成立：
   - 应用程序首次向与快速路径连接相同的安装相关联的队列管理器建立连接。
   - 这将设置环境变量 AMQ_SINGLE_INSTALLATION。
3. 用非快速路径连接到与 IBM WebSphere MQ 7.1 或更高版本安装相关联的队列管理器，对应用程序是否可以通过快速路径进行连接无影响。
4. 您无法既连接到与 IBM WebSphere MQ 7.0.1 安装相关联的队列管理器，又用快速路径连接到与 IBM WebSphere MQ 7.1 或更高版本安装相关联的队列管理器。

通过设置 AMQ_SINGLE_INSTALLATION，您可以将与队列管理器的任何连接设置为快速路径连接。 否则，将适用几乎相同的限制：

- 安装必须是从中装入 IBM MQ 运行时库的同一个安装。
- 同一进程上的每条连接必须指向同一安装。 如果尝试连接到与其他安装关联的队列管理器，那么连接将失败，原因码为 MQRC_INSTALLATION_MISMATCH。 请注意，如果设置了 AMQ_SINGLE_INSTALLATION，那么此限制将应用于所有连接，而不仅仅是快速路径连接。
- 通过快速路径连接仅连接一个队列管理器。

相关参考

MQCONNX-连接队列管理器 (扩展)
MQIEP 结构
2583 (0A17) (RC2583): MQRC_INSTALLATION_MISMATCH
2587 (0A1B) (RC2587): MQRC_HMSG_NOT_AVAILABLE
2590 (0A1E) (RC2590): MQRC_FASTPATH_NOT_AVAILABLE

ALW 在多安装环境中连接 .NET 应用程序

缺省情况下，应用程序使用主安装中的 .NET 组合件。 如果没有主安装，或者您不想使用主安装组合件，那么必须更新应用程序配置文件或 *DEVPATH* 环境变量。

如果系统上存在主安装，那么会将该安装的 .NET 组合件和策略文件注册到全局组合件高速缓存 (GAC)。 可以在每个安装的安装路径中找到所有其他安装的 .NET 组合件，但这些组合件不会向 GAC 注册。 因此，缺省情况下，应用程序使用主安装中的 .NET 组合件运行。 如果存在下列任何情况，那么必须更新应用程序配置文件：

- 您没有主安装。
- 您不希望应用程序使用主安装组合件。
- 主安装的 IBM MQ 版本低于编译应用程序所使用的版本。

有关如何更新应用程序配置文件的信息，请参阅 第 378 页的『使用应用程序配置文件连接 .NET 应用程序』。

如果以下情况成立，那么必须更新 *DEVPATH* 环境变量:

- 您希望应用程序使用来自非主安装的组合件，但主安装的版本与非主安装的版本相同。

有关如何更新 *DEVPATH* 变量的更多信息，请参阅 第 378 页的『使用 DEVPATH 连接 .NET 应用程序』。

使用应用程序配置文件连接 .NET 应用程序

在应用程序配置文件中，必须设置各种标记以将应用程序重定向到使用并非来自主安装的组合件。

下表显示了需要对应用程序配置文件进行的特定更改，以允许 .NET 应用程序使用特定组合件进行连接:

表 *30:* 配置应用程序以使用特定组合件

	使用较低版本的 IBM MQ 编译的应用程序	**使用更高版本的 IBM MQ 编译的应用程序**
使用更高版本的 IBM MQ 主安装来运行应用程序。 (GAC 中的更高版本组合件):	不需要进行任何更改	不需要进行任何更改
运行具有较低版本的 IBM MQ 主安装的应用程序。 (GAC 中的较早版本组合件):	不需要进行任何更改	在应用程序配置文件中: • 使用 *bindingRedirect* 标记来指示使用 GAC 中的较早版本的组合件
使用更高版本的 IBM MQ 非主安装来运行应用程序。 (安装文件夹中的更高版本组合件):	在应用程序配置文件中: • 使用 *codebase* 标记指向更高版本组合件的位置 • 使用 *bindingRedirect* 标记来指示使用更高版本的组合件	在应用程序配置文件中: • 使用 *codebase* 标记指向更高版本组合件的位置
运行具有较低版本的 IBM MQ 非主安装的应用程序。 (安装文件夹中的较早版本组合件):	在应用程序配置文件中: • 使用 *codebase* 标记指向较早版本组合件的位置 • 包含标记 *publisherpolicy Apply=no*	在应用程序配置文件中: • 使用 *codebase* 标记指向较早版本组合件的位置 • 使用 *bindingRedirect* 标记来指示使用较早版本的组合件 • 包含标记 *publisherpolicy Apply=no*

样本应用程序配置文件 `NonPrimaryRedirect.config` 在文件夹 *MQ_INSTALLATION_PATH*`\tools\dotnet\samples\base` 中提供。 可以使用任何非主安装的 IBM MQ 安装路径来修改此文件。 该文件还可以使用 *linkedConfiguration* 标记直接包含在其他配置文件中。 为 `nmqsget.exe.config` 和 `nmqsput.exe.config` 提供了样本。 这两个样本都使用 *linkedConfiguration* 标记并包含 `NonPrimaryRedirect.config` 文件。

使用 DEVPATH 连接 .NET 应用程序

您可以使用 *DEVPATH* 环境变量来查找组合件。 由 *DEVPATH* 变量指定的组合件优先于 GAC 中的任何组合件。 有关何时使用此变量的更多信息，请参阅 *DEVPATH* 上相应的 Microsoft 文档。

要使用 *DEVPATH* 环境变量查找组合件，必须将 *DEVPATH* 变量设置为包含要使用的组合件的文件夹。 然后，必须更新应用程序配置文件并添加以下运行时配置信息:

```
<configuration>
<runtime>
<developmentMode developerInstallation="true" />
</runtime>
</configuration>
```

相关概念

第 372 页的『在多安装环境中连接应用程序』

在 AIX, Linux, and Windows 系统上，如果装入了 IBM WebSphere MQ 7.1 或更高版本的库，那么 IBM MQ 将自动使用相应的库，而无需执行任何进一步的操作。 IBM MQ 使用与应用程序所连接的队列管理器相关联的安装中的库。

多个安装

相关任务

选择主安装

使用 .NET

ALW 更改主安装

您可以使用 **setmqinst** 命令将安装设置或取消设置为主安装。

关于此任务

此任务适用于 AIX, Linux, and Windows。

主要安装是必需的系统范围位置所引用的安装。 有关主安装的更多信息以及选择主安装的注意事项，请参阅选择主安装。

如果 IBM WebSphere MQ 7.1 或更高版本的安装与 IBM WebSphere MQ 7.0.1 的安装共存，那么 IBM WebSphere MQ 7.0.1 安装必须是主安装。 当安装了 IBM WebSphere MQ 7.1 或更高版本时，会将其标记为主安装，并且无法使 IBM WebSphere MQ 7.1 或更高版本的安装成为主安装。

Windows 在 Windows 上的安装过程中，可以指定安装将作为主安装。

Linux AIX 在 AIX and Linux 系统上，必须在安装后发出 **setmqinst** 命令以将安装设置为主安装。

过程

- 要将安装设置为主安装，请完成以下步骤:
  a) 通过输入以下命令来检查安装是否已经是主安装:

  ```
  MQ_INSTALLATION_PATH/bin/dspmqinst
  ```

  其中， *MQ_INSTALLATION_PATH* 是 IBM WebSphere MQ 7.1 或更高版本安装的安装路径。
  b) 如果将现有 IBM WebSphere MQ 7.1 或更高版本安装设置为主安装，请在继续下一步之前 取消设置 。

  如果在系统上安装了 IBM WebSphere MQ 7.0.1 ，那么无法更改主安装。
  c) 确保您已使用相应的权限登录:

  – AIX 以 root 用户身份在 AIX and Linux 上。

  – Windows 作为 Windows 系统上的 Administrators 组的成员。
  d) 输入下列其中一个命令:

  – 要使用要作为主安装的安装路径来设置主安装:

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -p MQ_INSTALLATION_PATH
```

– 要使用要作为主安装的安装的名称来设置主安装:

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -n installationName
```

e) Windows
在 Windows 系统上，重新启动系统。

- 要将安装取消设置为主安装，请完成以下步骤:
  a) 通过输入以下命令来检查哪个安装是主安装:

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

其中， *MQ_INSTALLATION_PATH* 是 IBM WebSphere MQ 7.1 或更高版本安装的安装路径。

如果 IBM WebSphere MQ 7.0.1 是主安装，那么无法取消设置主安装。

b) 确保您已使用相应的权限登录:

– AIX 以 root 用户身份在 AIX and Linux 上。

– Linux 作为 Windows 系统上的 Administrators 组的成员。

- 输入下列其中一个命令:
  – 要使用安装路径取消设置主安装，您不再希望成为主安装:

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -p MQ_INSTALLATION_PATH
```

  – 要使用安装的名称取消设置主安装，您不再希望成为主安装:

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -n installationName
```

相关任务
卸载，升级和维护主安装
选择安装名称
相关参考
只能与 Windows 上的主安装配合使用的功能部件
外部库和控制命令链接到 AIX and Linux 上的主安装
塞特姆因斯特

ALW 使队列管理器与安装相关联

创建队列管理器时，它将自动与发出 **crtmqm** 命令的安装相关联。 在 AIX, Linux, and Windows 上，可以使用 **setmqm** 命令来更改与队列管理器关联的安装。

关于此任务

队列管理器与该队列管理器的限制相关联的安装，以便该队列管理器只能由该安装中的命令进行管理。 有三个关键例外:

- **setmqm** 将更改与队列管理器关联的安装。 必须从要与队列管理器关联的安装 (而不是队列管理器当前关联的安装) 发出此命令。 **setmqm** 命令指定的安装名称必须与从中发出该命令的安装相匹配。
- 通常必须从与队列管理器关联的安装发出 **strmqm** 。 但是，首次在 IBM WebSphere MQ 7.1 或更高版本的安装上启动 IBM WebSphere MQ 7.0.1 或更低版本的队列管理器时，可以使用 **strmqm** 。 在这种情况下，**strmqm** 会启动队列管理器，并将其与从中发出命令的安装相关联。
- **dspmq** 显示有关系统上所有队列管理器的信息，而不仅仅是那些与 **dspmq** 命令的相同安装相关联的队列管理器。 `dspmq -o installation` 命令显示有关哪些队列管理器与哪些安装相关联的信息。

对于 HA 环境， **addmqinf** 命令会自动将队列管理器与从中发出 **addmqinf** 命令的安装相关联。 只要从与 **addmqinf** 命令相同的安装中发出 **strmqm** 命令，就无需进一步设置。 要使用其他安装启动队列管理器，必须首先使用 **setmqm** 命令更改关联的安装。

当您要将队列管理器与安装相关联时，可以通过以下方式使用 **setmqm** 命令:

- 在等效版本的 IBM MQ 之间移动个别队列管理器。 例如，将队列管理器从测试移至生产系统。
- 将个别队列管理器从较低版本的 IBM MQ 迁移到较高版本的 IBM MQ。 在版本之间迁移队列管理器具有您必须了解的各种含义。 有关迁移的更多信息，请参阅 维护和迁移。

过程

1. 从当前与队列管理器关联的安装中使用 **endmqm** 命令停止队列管理器。
2. 使用该安装中的 **setmqm** 命令将队列管理器与另一个安装相关联。
   例如，要将队列管理器 QMB 设置为与名为 Installation2 的安装相关联，请从 Installation2:

   ```
   MQ_INSTALLATION_PATH/bin/setmqm -m QMB -n Installation2
   ```

   其中， *MQ_INSTALLATION_PATH* 是 Installation2 的安装路径。
3. 使用 **strmqm** 命令从现在与队列管理器关联的安装启动队列管理器。
   此命令执行任何必需的队列管理器迁移，并导致队列管理器可供使用。

下一步做什么

如果已删除与队列管理器相关联的安装，或者如果队列管理器状态信息不可用，那么 **setmqm** 命令无法将队列管理器与另一个安装相关联。 在这种情况下，请执行以下操作:

1. 使用 **dspmqinst** 命令可查看系统上的其他安装。
2. 手动修改 mqs.ini 中 QueueManager 节的 InstallationName 字段以指定另一个安装。
3. 使用该安装中的 **dltmqm** 命令来删除队列管理器。

相关概念

第 381 页的『在系统上查找 IBM MQ 的安装』
如果系统上存在多个 IBM MQ 安装，那么可以检查安装的版本及其位置。

第 72 页的『IBM MQ 配置文件， mqs.ini』
IBM MQ 配置文件 mqs.ini 包含与节点上所有队列管理器相关的信息。 它是在安装期间自动创建的。

相关任务

选择主安装

相关参考

阿达姆金夫

dspmq

长石

恩德姆

塞特姆

strmqm

ALW 在系统上查找 IBM MQ 的安装

如果系统上存在多个 IBM MQ 安装，那么可以检查安装的版本及其位置。

您可以使用以下方法来查找系统上的 IBM MQ 安装:

- 使用 **dspmqver** 命令。 如果是从 IBM WebSphere MQ 7.0.1 安装发出的，那么此命令不会提供系统上所有安装的详细信息。
- 使用平台安装工具来查询 IBM MQ 的安装位置。 然后从 IBM WebSphere MQ 7.1 或更高版本的安装中使用 **dspmqver** 命令。 以下命令是可用于查询已安装 IBM MQ 的位置的命令示例:

  - AIX 在 AIX 系统上，可以使用 **lslpp** 命令:

    ```
    lslpp -R ALL -l mqm.base.runtime
    ```

  - Linux 在 Linux 系统上，可以使用 **rpm** 命令:

    ```
    rpm -qa --qf "%{NAME}-%{VERSION}-%{RELEASE}\t%{INSTPREFIXES}\n" | grep MQSeriesRuntime
    ```

  - Windows 在 Windows 系统上，可以使用 **wmic** 命令。 此命令可能会安装 wmic 客户机:

    ```
    wmic product where "(Name like '%MQ%') AND (not Name like '%bitSupport')" get Name,
    Version, InstallLocation
    ```

- Linux AIX 在 AIX and Linux 系统上，发出以下命令以了解 IBM MQ 的安装位置:

  ```
  cat /etc/opt/mqm/mqinst.ini
  ```

  然后从 IBM WebSphere MQ 7.1 或更高版本的安装中使用 **dspmqver** 命令。

- Windows 要显示系统上安装的详细信息，请在 32 位 Windows 上发出以下命令:

  ```
  reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation" /s
  ```

- Windows 在 64 位 Windows 上，发出以下命令:

  ```
  reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IBM\WebSphere MQ\Installation" /s
  ```

reg.exe 命令仅显示 IBM WebSphere MQ 7.1 或更高版本安装的信息。

相关概念
多个安装
相关参考
dspmqver
长石

配置高可用性，恢复和重新启动

通过在队列管理器发生故障时保持队列可用性，以及在服务器或存储器发生故障后恢复消息，可以使应用程序具有高可用性。

关于此任务

z/OS 在 z/OS 上，将高可用性内置到平台中。 您还可以使用队列共享组来提高服务器应用程序可用性。 请参阅共享队列和队列共享组。

Multi 在 多平台上，您可以通过使用客户机重新连接在一组队列管理器之间自动切换客户机，或者在队列管理器发生故障后自动切换到多实例队列管理器的新活动实例，从而提高客户机应用程序可用性。 IBM MQ classes for Java 不支持客户机自动重新连接。 多实例队列管理器配置为在多个服务器上作为单个队列管理器运行。 将服务器应用程序部署到此队列管理器。 如果运行活动实例的服务器失败，那么执行将自动切换到另一个服务器上同一队列管理器的备用实例。 如果将服务器应用程序配置为作为队列管理器服务运行，那么当备用实例成为主动运行的队列管理器实例时，将重新启动这些应用程序。

在多平台上提高服务器应用程序可用性的另一种方法是将服务器应用程序部署到队列管理器集群中的多台计算机。 从 IBM WebSphere MQ 7.1 开始，集群错误恢复将重新运行导致问题的操作，直到问题得到解决为止。 请参阅 对除 z/OS 以外的服务器上的集群错误恢复的更改。 您还可以将 IBM MQ for Multiplatforms 配置为特定于平台的集群解决方案的一部分，例如:

- Microsoft 集群服务器
- IBM i IBM i 上的 HA 集群

- Linux AIX PowerHA for AIX (以前为 HACMP on AIX) 和其他 UNIX and Linux 集群解决方案

Linux 在 Linux 系统上，可以配置复制的数据队列管理器 (RDQM) 以实现高可用性或灾难恢复解决方案。 为了实现高可用性，将在由三个 Linux 服务器组成的组中的每个节点上配置同一队列管理器的实例。 三个实例之一是活动实例。 来自活动队列管理器的数据将同步复制到其他两个实例，因此在发生某种故障时，其中一个实例可以接管。 对于灾难恢复，队列管理器在一个站点的主节点上运行，而该队列管理器的辅助实例位于另一个站点的恢复节点上。 在主实例和辅助实例之间复制数据，如果主节点由于某种原因而丢失，那么可以将辅助实例制作成主实例并启动。

CP4I 本机 HA 是针对容器的高可用性解决方案。 本机 HA 使用日志复制使在不同节点上运行的队列管理器的三个实例保持最新。 一个实例在任何时候都处于活动状态，并处理消息。 活动队列管理器将其日志更新发送到其他两个实例以保持其更新。 如果活动实例失败，那么其中一个副本实例将自动接管活动角色。

MQ Appliance 高可用性或灾难恢复解决方案的另一个选项是部署一对 IBM MQ 设备。 请参阅 IBM MQ Appliance 文档中的 高可用性 和 灾难恢复 。

消息传递系统确保将输入到系统中的消息传递到其目标。 IBM MQ 可以使用 **dspmqrte** 命令在消息从一个队列管理器移动到另一个队列管理器时跟踪该消息的路由。 如果系统发生故障，那么可以根据故障类型以及系统配置方式以各种方式恢复消息。 IBM MQ 维护处理接收，传输和传递消息的队列管理器活动的恢复日志。 它将这些日志用于三种类型的恢复:

1. 以计划方式停止 IBM MQ 时的 重新启动恢复。
2. 故障恢复，当故障停止时 IBM MQ。
3. 介质恢复，用于恢复受损对象。

在所有情况下，恢复会将队列管理器恢复到队列管理器停止时所处的状态，但会回滚任何正在进行的事务，从而从队列中除去队列管理器停止时正在进行的任何更新。 恢复会恢复所有持久消息; 在此过程中，可能会丢失非持久消息。

警告: 不能将恢复日志移动到其他操作系统。

自动客户机重新连接

通过配置多个组件，可以使客户机应用程序自动进行重新连接，而不必编写任何其他代码。

自动客户机重新连接是内联的。 将在客户机应用程序中的任意点自动复原连接，并且将全部复原用于打开对象的句柄。

相比之下，手动重新连接需要客户机应用程序使用 MQCONN 或 MQCONNX 重新创建连接，并重新打开对象。 自动客户机重新连接适合许多但并非所有客户机应用程序。

第 384 页的表 31 列出了必须安装在客户机工作站上的 IBM MQ 客户机支持的最早发行版。 必须将客户机工作站升级到这些级别之一，应用程序才能使用自动客户机重新连接。 第 384 页的表 32 列出了启用自动客户机重新连接的其他需求。

通过程序访问重新连接选项，客户机应用程序可以设置重新连接选项。 除 JMS 和 XMS 客户机外，如果客户机应用程序有权访问重新连接选项，那么它还可以创建事件处理程序来处理重新连接事件。

现有客户机应用程序可能能够从重新连接支持中获益，而无需重新编译和链接:

- 对于非 JMS 客户机，设置 mqclient.ini 环境变量 DefRecon 以设置重新连接选项。 使用 CCDT 连接到队列管理器。 如果客户机要连接到多实例队列管理器，请提供 CCDT 中活动队列管理器实例和备用队列管理器实例的网络地址。 对于复制的数据队列管理器或 IBM MQ Appliance 上的 HA 队列管理器，可以指定活动队列管理器和备用队列管理器使用的浮动 IP 地址来简化配置。
- 对于 JMS 客户机，请在连接工厂配置中设置重新连接选项。 在 Java EE 服务器的 EJB 容器内运行时，MDB 可以使用 IBM MQ 资源适配器 (或侦听器端口 (如果在 WebSphere Application Server 中运行)) 的激活规范提供的重新连接机制来重新连接到 IBM MQ 。 但是，如果应用程序不是 MDB (或正在 Web 容器中运行) ，那么应用程序必须实现其自己的重新连接逻辑，因为在此场景中不支持自动客户机重新连接。

IBM MQ 资源适配器提供此重新连接功能，用于将消息传递到消息驱动的 bean ，但其他 Java EE 元素 (例如 servlet) 必须实现它们自己的重新连接。

注: IBM MQ classes for Java 不支持自动客户机重新连接。

表 *31:* 支持的客户端

客户机接口	客户机	程序对重新连接选项的访问权	重新连接支持
消息传递 API	C， C + + ， COBOL ，非受管 Visual Basic ， XMS (Windows 上的非受管 XMS)	7.0.1	7.0.1
	JMS (JSE 和 Java EE 客户机容器和受管容器)	7.0.1.3	7.0.1.3
	IBM MQ classes for Java	不支持	不支持
	受管 XMS 和受管 .NET 客户机 :C# ， Visual Basic ,	7.1	7.1
其他 API	Windows Communication Foundation (非受管 [1])	不支持	7.0.1
	Windows Communication Foundation (受管 [1])	不支持	不支持
	轴 1	不支持	不支持
	轴 2	不支持	7.0.1.3
	HTTP (Web 2.0)	不支持	7.0.1.3

1. 在 WCF 绑定配置中设置受管或非受管方式。

自动重新连接具有以下配置需求:

表 *32:* 自动重新连接配置需求

组件	要求	不符合要求的影响
IBM MQ MQI client 安装	请参阅第 384 页的表 31	MQRC_OPTIONS_ERROR
IBM MQ 服务器安装	级别 7.0.1	MQRC_OPTIONS_ERROR
通道	SHARECNV > 0	MQRC_ENVIRONMENT_ERROR
应用程序环境	必须为线程	MQRC_ENVIRONMENT_ERROR
MQI	值为下列其中一项: • MQCONNX， MQCNO **选项** 设置为 MQCNO_RECONNECT 或 MQCNO_RECONNECT_Q_MGR。 • Defrecon=YES|QMGR 在 mqclient.ini 中 • 在 JMS 中，设置连接工厂的 **CLIENTRECONNECTOPTIONS** 属性。	连接中断或队列管理器结束或失败时的 MQCC_FAILED 。

第 385 页的图 71 显示了客户机重新连接中涉及的组件之间的主要交互。

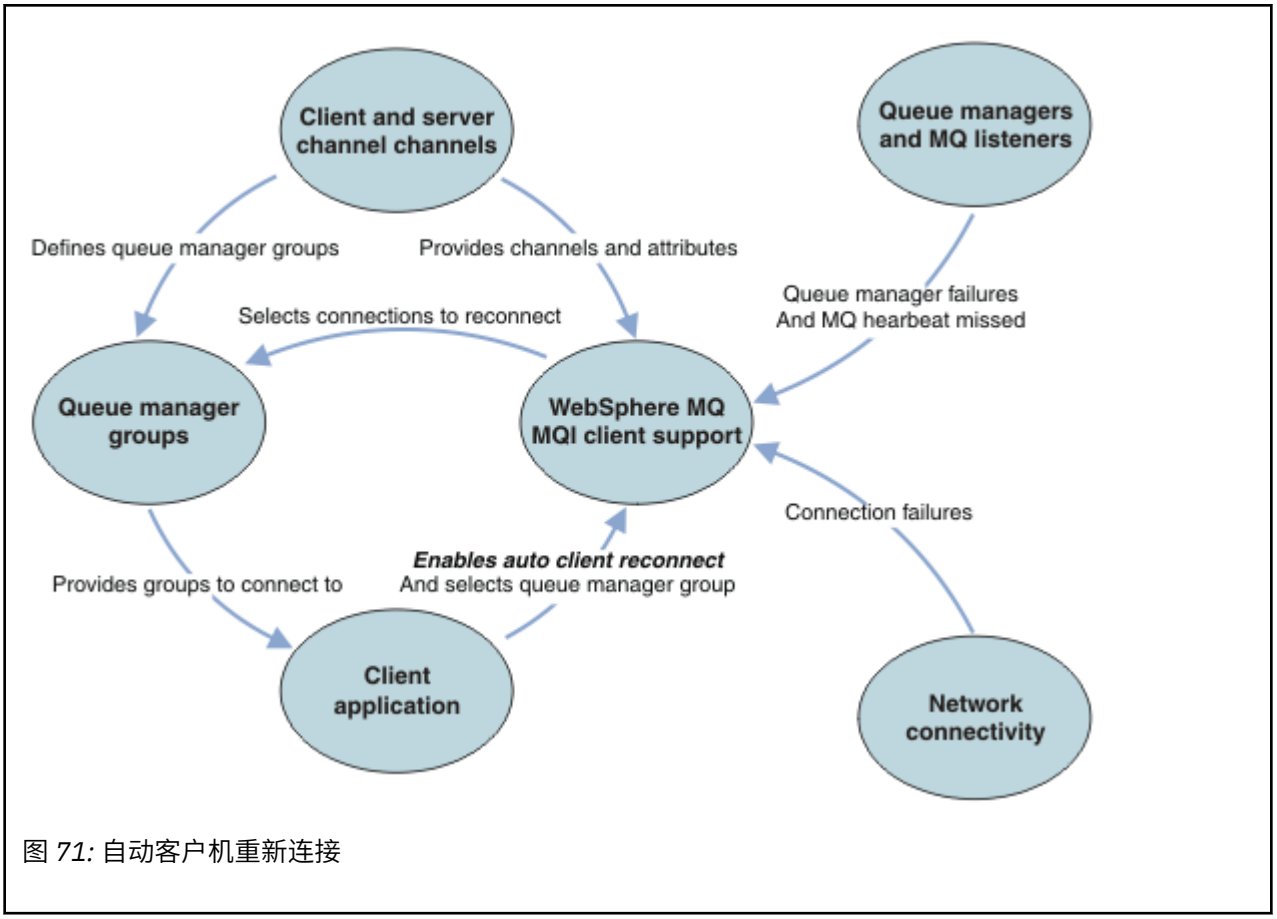


图 *71:* 自动客户机重新连接

客户机应用程序

客户机应用程序是 IBM MQ MQI client。 有关 JMS 客户机的自动客户机重新连接的详细信息，请参阅 使用自动 JMS 客户机重新连接。

- 缺省情况下，不会自动重新连接客户机。 通过设置 MQCONNX MQCNO **选项** MQCNO_RECONNECT 或 MQCNO_RECONNECT_Q_MGR 来启用自动客户机重新连接。
- 许多应用程序的编写方式使得它们能够利用自动重新连接，而无需其他编码。 通过在 `mqclient.ini` 配置文件的通道节中设置 DefRecon 属性，为现有程序启用自动重新连接，而不进行任何编码更改。
- 使用以下三个选项之一:
  1. 修改程序，以使逻辑不受重新连接的影响。 例如，您可能必须在同步点内发出 MQI 调用，然后重新提交回退事务。 如果已回退事务，那么异步使用者应检查它们是否已 "暂挂"。
  2. 添加事件处理程序以检测重新连接，并在重新建立连接时恢复客户机应用程序的状态。
  3. 请勿启用自动重新连接: 而是断开客户机连接并发出新的 MQCONN 或 MQCONNX MQI 调用以查找在同一队列管理器组中运行的另一个队列管理器实例。

  有关这三个选项的更多详细信息，请参阅 第 458 页的『应用程序恢复』。
- 重新连接到同名的队列管理器并不保证您已重新连接到队列管理器的同一实例。

  使用 MQCNO 选项 MQCNO_RECONNECT_Q_MGR 来重新连接到同一队列管理器的实例。
- 客户机可以注册事件处理程序，以便可以通知它重新连接的状态。 无法使用事件处理程序中传递的 MQHCONN 。 提供了以下原因码:

  MQRC_RECONNEC 丁
  连接失败，系统正在尝试重新连接。 如果进行了多次重新连接尝试，那么您将接收到多个 MQRC_RECONNECTING 事件。

  MQRC_RECONNECTED
  已建立重新连接并且已成功重新建立所有句柄。

MQRC_RECONNECT_FAILED
重新连接不成功。

MQRC_RECONNECT_QMID_MATCH
可重新连接的连接指定了 MQCNO_RECONNECT_Q_MGR，且连接尝试重新连接到其他队列管理器。

MQRC_RECONNECT_Q_MGR_REQD
在客户机程序中指定了需要重新连接到同一队列管理器的选项，例如 MQGET 调用中的 MQMO_MATCH_MSG_TOKEN 。

- 可重新连接的客户机只能在连接后自动重新连接。 即，如果失败，那么不会再次尝试 MQCONNX 调用本身。 例如，如果从 MQCONNX 接收到返回码 2543 - MQRC_STANDBY_Q_MGR ，请在短暂延迟后重新发出调用。

MQRC_RECONNECT_不兼容
当应用程序尝试使用 MQPMO_LOGICAL_ORDER (带有 MQPUT 和 MQPUT1) 或 MQGMO_LOGICAL_ORDER (带有 MQGET) 时，将返回此原因码 当设置了重新连接选项时。 返回原因码的原因是确保应用程序在此类情况下从不使用重新连接。

MQRC_CALL_中断
如果在执行 "落实" 调用期间连接中断，并且客户机重新连接，那么将返回此原因码。 同步点外部持久消息的 MQPUT 也会导致将相同的原因码返回到应用程序。

高可用性队列管理器

高可用性队列管理器具有一个活动实例以及一个或多个队列管理器备用实例。 活动队列管理器与备用队列管理器同步，这样当活动实例发生故障时，备用队列管理器可以自动接管。 有许多不同的解决方案可提供高可用性队列管理器，请参阅 第 392 页的『高可用性配置』。

通过使用自动客户机重新连接，在高可用性队列管理器激活其备用实例后，可以简化重新启动 IBM MQ MQI client 应用程序的过程。

高可用性队列管理器的备用实例通常位于与活动实例不同的网络地址。 在客户机连接定义表 (CCDT) 中包含两个实例的网络地址。 为 **CONNAME** 参数提供网络地址列表，或者为 CCDT 中的队列管理器定义多行。 复制的数据队列管理器和 IBM MQ Appliance 高可用性队列管理器支持浮动 IP 地址，其中指定单个地址以用于活动或备用队列管理器。

队列管理器组

通常， IBM MQ MQI clients 会重新连接到队列管理器组中的任何队列管理器。 有时，您希望 IBM MQ MQI client 仅重新连接到同一队列管理器。 它可能与队列管理器具有亲缘关系。

您可以选择客户机应用程序是始终连接并重新连接到同名的队列管理器，同一队列管理器，还是连接到客户机连接表中使用相同 QMNAME 值定义的任何一组队列管理器。

- 客户机通道定义中的队列管理器名称属性 QMNAME 是队列管理器组的名称。
- 在客户机应用程序中，如果将 MQCONN 或 MQCONNX QmgrName 参数的值设置为队列管理器名称，那么客户机仅连接到具有该名称的队列管理器。 如果以星号 (*) 作为队列管理器名称的前缀，那么客户机将连接到队列管理器组中具有相同 QMNAME 值的任何队列管理器。 有关完整说明，请参阅 CCDT 中的队列管理器组。

您可以阻止客户机重新连接到其他队列管理器。 设置 MQCNO 选项 MQCNO_RECONNECT_Q_MGR。 如果 IBM MQ MQI client 重新连接到其他队列管理器，那么它将失败。 如果设置 MQCNO 选项 MQCNO_RECONNECT_Q_MGR，请勿在同一队列管理器组中包含其他队列管理器。 如果客户机重新连接到的队列管理器与它连接到的队列管理器不同，那么客户机将返回错误。

队列共享组

z/OS 客户机自动重新连接到 z/OS 队列共享组，使用与任何其他环境相同的机制进行重新连接。 客户机将重新连接到为原始连接配置的相同队列管理器选择。 例如，使用客户机通道定义表时，管理员应确保该表中的所有条目都解析为同一个 z/OS 队列共享组。

客户机和服务器通道定义

客户机和服务器通道定义定义客户机应用程序可以重新连接到的队列管理器组。 这些定义用于管理重新连接的选择和计时以及其他因素 (例如安全性); 请参阅相关主题。 在两个组中列出了要考虑用于重新连接的最相关的通道属性:

客户机连接属性

连接亲缘关系 (AFFINITY) 亲缘关系
连接亲缘关系

客户机通道权重 (CLNTWGHT) CLNTWGHT
客户机通道权重。

连接名称 (CONNAME) CONNAME
连接信息。

脉动信号间隔 (HBINT) HBINT
脉动信号间隔。 在服务器连接通道上设置脉动信号间隔。

保持活动时间间隔 (KAINT) KAINT
保持活动时间间隔。 在服务器连接通道上设置保持活动时间间隔。

z/OS 请注意， KAINT 仅适用于 z/OS 。

队列管理器名称 (QMNAME) QMNAME
队列管理器名称。

服务器连接属性

脉动信号间隔 (HBINT) HBINT
脉动信号间隔。 在客户机连接通道上设置脉动信号间隔。

保持活动时间间隔 (KAINT) KAINT
保持活动时间间隔。 在客户机连接通道上设置保持活动时间间隔。

z/OS 请注意， KAINT 仅适用于 z/OS 。

KAINT 是网络层脉动信号， HBINT 是客户机与队列管理器之间的 IBM MQ 脉动信号。 将这些脉动信号设置为更短的时间有两个目的:

1. 通过模拟连接上的活动，负责关闭不活动连接的网络层软件不太可能关闭您的连接。
2. 如果连接关闭，那么会缩短检测到中断连接之前的延迟。

缺省 TCP/IP 保持活动时间间隔为 2 小时。 请考虑将 KAINT 和 HBINT 属性设置为更短的时间。 请勿假定网络的正常行为符合自动重新连接的需要。 例如，一些防火墙可以在不到 10 分钟后关闭不活动的 TCP/IP 连接。

网络连接

只有由网络传递到 IBM MQ MQI client 的网络故障才由客户机的自动重新连接功能处理。

- 传输自动执行的重新连接对于 IBM MQ 不可见。
- 设置 HBINT 有助于处理对 IBM MQ 不可见的网络故障。

队列管理器和 IBM MQ 侦听器

客户机重新连接由服务器故障，队列管理器故障，网络连接故障以及管理员切换到另一个队列管理器实例触发。

- 如果您正在使用多实例队列管理器，那么当您将控制从活动队列管理器实例切换到备用实例时，会发生客户机重新连接的其他原因。
- 使用缺省 **endmqm** 命令结束队列管理器不会触发自动客户机重新连接。 在 **endmqm** 命令上添加 `-r` 选项以请求自动客户机重新连接，或者添加 `-s` 选项以在关闭后传输到备用队列管理器实例。

IBM MQ MQI client 自动重新连接支持

如果在 IBM MQ MQI client 中使用自动客户机重新连接支持，那么客户机应用程序将自动重新连接并继续处理，而无需发出 MQCONN 或 MQCONNX MQI 调用以重新连接到队列管理器。

- 自动客户机重新连接由下列其中一个实例触发:
  - 队列管理器故障
  - 结束队列管理器并在 **endmqm** 命令上指定 -r， reconnect 和选项
- MQCONNX MQCNO 选项控制您是否已启用自动客户机重新连接。 重新连接选项中描述了这些选项。
- 自动客户机重新连接会代表应用程序发出 MQI 调用，以将连接句柄和句柄复原到其他打开的对象，以便程序可以在处理因连接中断而导致的任何 MQI 错误后恢复正常处理。 请参阅第 460 页的『自动重新连接的客户机的恢复』。
- 如果为连接编写了通道出口程序，那么该出口将接收到这些额外的 MQI 调用。
- 您可以注册重新连接事件处理程序，该事件处理程序在重新连接开始和完成时触发。

虽然预期的重新连接时间不超过一分钟，但重新连接可能需要更长时间，因为队列管理器可能有许多资源需要管理。 在此期间，客户机应用程序可能持有不属于 IBM MQ 资源的锁定。 您可以配置一个超时值，以限制客户机等待重新连接的时间。 值 (以秒计) 在 mqclient.ini 文件中设置。

```
Channels:
MQReconnectTimeout = 1800
```

超时到期后不会进行重新连接尝试。 当系统检测到超时已到期时，将返回 MQRC_RECONNECT_FAILED 错误。

相关概念

可重新连接的客户机

相关任务

停止队列管理器

z/OS 控制台消息监视

在 IBM MQ for z/OS 上，队列管理器或通道启动程序发出了一些应视为特别重要的参考消息。 这些消息本身并不表示存在问题，但对于跟踪很有用，因为它们确实指示了可能需要解决的潜在问题。

这些控制台消息的存在还可能指示用户应用程序正在将大量消息放入页集，这可能是较大问题的症状:

- 用户应用程序中 PUTs 消息的问题，例如不受控制的循环。
- 来自队列的 GET 消息不再起作用的用户应用程序。

要监视的控制台消息

以下列表概述了可能指示较大问题的消息。 确定是否需要使用系统自动化来跟踪这些消息，并提供相应的文档，以便有效跟进任何潜在问题。

CSQI004I: *csect-name* 考虑建立索引 *queue-name* BY *index-type* for *connection-type* CONNECTION *connection-name*， *num-msgs* MESSAGES SKIP

- 队列管理器检测到应用程序通过消息标识或相关标识从未定义索引的队列接收消息。
- 请考虑通过将本地队列对象 *queue-name* 和 INDXTYPE 属性更改为具有值 *index-type* 来为标识的队列建立索引。

CSQI031I: *csect-name* PAGE SET *psid* 的新 EXTENT 已成功格式化

- 检查分配给此页集的队列的 curdepth。
- 调查未能处理消息的原因。

.

CSQI041I: *csect-name* JOB *jobname* USER *userid* 访问页集时发生错误 *psid*

- 确定是否将页集分配给队列管理器。
- 发出 **DISPLAY USAGE** 命令以确定页集的状态。
- 请检查队列管理器作业记录以获取其他错误消息。

.

CSQI045I: *csect-name* 日志 RBA 已达到 *rba*。 计划日志复位

- 计划在方便的时候停止队列管理器并重置日志。
- 如果队列管理器正在使用 6 字节的日志 RBA，请考虑将队列管理器转换为使用 8 字节的日志 RBA。

CSQI046E: *csect-name* 日志 RBA 已达到 *rba*。 执行日志复位

- 计划在方便的时候停止队列管理器并重置日志。
- 如果队列管理器正在使用 6 字节的日志 RBA，请考虑将队列管理器转换为使用 8 字节的日志 RBA。

CSQI047E: *csect-name* 日志 RBA 已达到 *rba*。 停止队列管理器并复位日志

- 立即停止队列管理器并重置日志。
- 如果队列管理器正在使用 6 字节的日志 RBA，请考虑将队列管理器转换为使用 8 字节的日志 RBA。

CSQJ004I: ACTIVE LOG COPY *n* INACTIVE， LOG IN SINGLE MODE， ENDRBA= *ttt*

- 队列管理器已激活 "单一" 日志记录方式。 这通常指示日志卸载问题。
- 发出 **DISPLAY LOG** 命令以确定用于对活动日志和归档日志进行双工的设置。 此屏幕还显示需要卸载处理的活动日志数。
- 检查队列管理器作业记录以获取其他错误消息

CSQJ031D: *csect-name*，必须重置日志 RBA 范围。 应答 "Y" 以继续启动或应答 "N" 以关闭

- 停止队列管理器并尽快重置日志，然后重置日志。
- 如果队列管理器正在使用 6 字节的日志 RBA，请考虑将队列管理器转换为使用 8 字节的日志 RBA。

CSQJ032E: *csect-name alert-lvl* -接近结束日志 RBA 范围 *max-rba*。 当前日志 RBA 为 *current-rba*。

- 计划停止队列管理器并尽快重置日志。
- 如果队列管理器正在使用 6 字节的日志 RBA，请考虑将队列管理器转换为使用 8 字节的日志 RBA。

CSQJ110E: LAST COPY*n* ACTIVE LOG DATA SET IS *nnn* PERCENT FULL

- 通过执行显示请求来确定与日志卸载过程相关的未完成请求，执行步骤以完成其他等待卸载任务。 执行必需的操作以满足任何请求，并允许卸载继续。
- 请考虑是否有足够的活动日志数据集。 如果需要，可以使用 DEFINE LOG 命令动态添加其他日志数据集。

CSQJ111A: 在活动日志数据集中空间输出

- 执行显示请求以确保没有与日志卸载过程相关的未完成请求。 执行必需的操作以满足任何请求，并允许卸载继续。
- 请考虑是否有足够的活动日志数据集。 如果需要，可以使用 DEFINE LOG 命令动态添加其他日志数据集。
- 如果延迟是由于缺少卸载所需的资源而导致的，那么必须提供必要的资源以允许卸载完成，从而允许日志记录继续进行。 有关从此情况恢复的信息，请参阅 归档日志问题。

CSQJ114I: 归档数据集出错，卸载继续，仅生成一个归档数据集

- 请检查队列管理器作业记录以获取其他错误消息。
- 生成归档日志的第二个副本，并手动更新 BSDS。

CSQJ115E: OFFLOAD 失败，无法分配归档数据集

查看消息 CSQJ103E 或 CSQJ073E 的错误状态信息。 请更正导致数据集分配错误的条件，以便在重试时可以执行卸载。

CSQJ136I: 无法为连接分配磁带机-标识 = *xxxx* CORRELATION-标识 = *yyyyyy*， *m* ALLOCATED *n* ALLOWED

- 请检查队列管理器作业记录以获取其他错误消息。

.

CSQJ151I: *csect-name* ERROR READING RBA *rrr*， CONNECTION-ID= *xxxx* CORRELATION-ID= *yyyyyy* REASON CODE= *ccc*

- 请检查队列管理器作业记录以获取其他消息。
- 发出 **DISPLAY CONN** 命令以确定哪个连接未落实其活动。
- 确保应用程序可以落实其更新。

CSQJ160I: 已找到长时间运行的 Uow， URID= *urid* CONNECTION NAME= *name*

- 请检查队列管理器作业记录以获取其他消息。
- 发出 **DISPLAY CONN** 命令以确定哪个连接未落实其活动。
- 确保应用程序可以落实其更新。

CSQJ161I: UOW NOT AFTER *n* OFFLOADS， URID= *urid* CONNECTION NAME= *name*

- 确定是否将页集分配给队列管理器。
- 发出 **DISPLAY USAGE** 命令以确定页集的状态。
- 请检查队列管理器作业记录以获取其他消息。

CSQP011E: CONNECT ERROR STATUS *ret-code* PAGE SET *psid*

- 检查分配给此页集的队列的 curdepth。
- 调查处理消息失败的原因。

CSQP013I: *csect-name* 为页集 psid 创建的新扩展数据块。 现在将格式化新扩展数据块

- 检查分配给此页集的队列的 curdepth。
- 调查处理消息失败的原因。
- 确定是否需要将队列重定位到另一个页集。
- 如果卷已满，请确定是否需要使页集成为多卷数据集。 如果页集已经是多卷，请考虑向正在使用的存储器组添加更多卷。 如果有更多空间可用，请通过将页集 **EXPAND** 方法设置为 **SYSTEM** 来重试扩展。 如果需要重试，请将 **EXPAND** 切换到 **SYSTEM**，然后返回到正常设置。

CSQP014E: *csect-name* EXPANSION FOR PAGE SET psid 失败。 将其延期的未来请求将被拒绝

- 检查分配给此页集的队列的 curdepth。
- 调查处理消息失败的原因。
- 确定是否需要将队列重定位到另一个页集。

CSQP016E: *Csect-name* PAGE SET *psid* 已达到最大扩展数据块数。 不能再次扩展

- 检查分配给此页集的队列的 curdepth。
- 调查处理消息失败的原因。

CSQP017I: *csect-name* EXPANSION STARTED FOR PAGE SET *psid*

发出 DISPLAY THREAD 命令以确定 IBM MQ 中工作单元的状态。

CSQP047E: 不可用的页集可能导致问题-执行操作以纠正此情况

- 遵循系统程序员响应。

CSQQ008I: 恩 队列管理器中的恢复单元仍处于不确定状态 *Qqqq*

- 调查死信队列的状态。 确保未禁用死信队列 PUT。
- 请确保死信队列未达到 MAXMSG 限制。

CSQQ113I: *psb-name region-id* 无法处理此消息

- 检查 CSQOUTX 数据集以确定 CSQINPX 故障的原因。
- 可能无法处理某些命令。

CSQX035I: *csect-name* 与队列管理器 *qmgr-name* 的连接正在停止或中断， MQCC= *mqcc* MQRC= *mqrc* (*mqrc-text*)

- 请检查 MQRC 以确定故障原因。
- 这些代码记录在 IBM MQ for z/OS 消息、完成代码和原因码中。

CSQX032I: *csect-name* 初始化命令处理程序已终止

- 请检查 MQRC 以确定故障原因。
- 这些代码记录在 IBM MQ for z/OS 消息、完成代码和原因码中。

CSQX048I: *csect-name* 无法转换 *name* 的消息， MQCC= *mqcc* MQRC= *mqrc* (*mqrc-text*)

- 检查作业记录以确定 TCP/IP 故障的原因。
- 检查 TCP/IP 地址空间中是否存在错误。

CSQX234I: *csect-name* 侦听器已停止， TRPTYPE= *trptype* INDISP= 处置

- 如果侦听器未停止，请遵循 **STOP** 命令，检查 TCP/IP 地址空间中是否存在错误。
- 遵循系统程序员响应。

CSQX407I: *csect-name* 集群队列 *q-name* 定义不一致

- 集群中的多个集群队列具有不一致的值。 调查并解决差异。

CSQX411I: *csect-name* 存储库管理器已停止

- 如果存储库管理器由于错误而停止，请检查作业记录以获取消息。

CSQX417I: *csect-name* 集群发送方保留用于已除去的队列管理器 *qmgr-name*

- 遵循系统程序员响应。

CSQX418I: *csect-name* 只有一个存储库用于集群 *cluster_name*

- 为了提高高可用性，应使用两个完整存储库配置集群。

CSQX419I: *csect-name* 没有集群 *cluster_name* 的集群接收方

- 遵循系统程序员响应。

CSQX420I: *csect-name* 没有集群 *cluster_name* 的存储库

- 遵循系统程序员响应。

CSQX448E: *csect-name* 存储库管理器由于错误而正在停止。 在 *n* 秒后重新启动

- 遵循系统程序员响应。

  此消息每 600 秒 (10 分钟) 发出一次，直到 SYSTEM.CLUSTER.COMMAND.QUEUE :

  ```
  ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
  ```

  在启用队列之前，可能需要手动干预来解决导致存储库管理器在发出第一条 CSQX448E 消息之前结束的问题。

CSQX548E: *csect-name* 发送到本地死信队列的消息，通道 *channel-name* 原因 =*mqrc* (*mqrc-text*)

- 遵循系统程序员响应。

CSQX788I: *csect-name* 使用函数 "*func*" 查找地址 *address* 的 DNS 耗时 *n* 秒

- 遵循系统程序员响应。

CSQY222E: *csect-name* 队列管理器本地存储空间严重不足——请采取行动

- 队列管理器在扩展专用区域中的虚拟存储空间严重不足。 应采取行动缓解这种情况，并避免队列管理器可能出现的异常终止。

CSQY225E: *csect-name* 队列管理器严重缺少高于 bar 的本地存储器-执行操作

- 队列管理器正严重缺乏高于界线的虚拟存储器。 应采取行动缓解这种情况，并避免队列管理器可能出现的异常终止。

CSQ5038I: *csect-name* 服务任务 service-task 自 hh.mm.ss.nnnnnn 以来一直无响应。 检查 Db2 的问题

- 遵循系统程序员响应。

高可用性配置

如果要在高可用性 (HA) 配置中操作 IBM MQ 队列管理器，那么可以设置队列管理器以使用高可用性管理器 (例如 PowerHA for AIX (以前为 HACMP)) 或 Microsoft 集群服务 (MSCS) ，或使用 IBM MQ 多实例队列管理器。 在 Linux 系统上，您还可以部署复制的数据队列管理器 (RDQM) ，这些数据队列管理器使用基于定额的组来提供高可用性。 另一个选项 "本机 HA" 针对容器部署。

MQ Appliance 高可用性或灾难恢复解决方案的另一个选项是部署一对 IBM MQ 设备。 请参阅 IBM MQ Appliance 文档中的 高可用性 和 灾难恢复 。

您需要了解以下配置定义:

队列管理器集群
由一台或多台计算机上的两个或多个队列管理器组成的组，提供自动互连，并允许在它们之间共享队列以实现负载均衡和冗余。 从 IBM WebSphere MQ 7.1 开始，集群错误恢复将重新运行导致问题的操作，直到问题得到解决为止。

HA 集群
HA 集群是由两台或更多台计算机和资源 (例如磁盘和网络) 组成的组，它们连接在一起并以如下方式配置: 如果发生故障，那么高可用性管理器 (例如 HACMP (AIX and Linux) 或 MSCS (Windows)) 将执行 故障转移。 故障转移将应用程序的状态数据从发生故障的计算机传输到集群中的另一台计算机，并在那里重新启动它们的操作。 这将提供在 HA 集群中运行的服务的高可用性。 第 393 页的『HA 集群与队列管理器集群的关系』中描述了 IBM MQ 集群与 HA 集群之间的关系。

多实例队列管理器
两台计算机上配置了相同的队列管理器。 通过启动多个实例，其中一个实例成为活动实例，另一个实例成为备用实例。 如果活动实例发生故障，则由另一台计算机上运行的备用实例自动接管。 您可以使用多实例队列管理器基于 IBM MQ 配置自己的高可用性消息传递系统，而不需要诸如 HACMP 或 MSCS 之类的集群技术。 HA 集群和多实例队列管理器是使队列管理器高度可用的替代方法。 请勿通过将多实例队列管理器放在 HA 集群中来组合这些队列管理器。

高可用性复制数据队列管理器 (HA RDQM)
在由三个 Linux 服务器组成的组中的每个节点上配置的相同队列管理器的实例。 三个实例之一是活动实例。 来自活动队列管理器的数据将同步复制到其他两个实例，因此在发生某种故障时，其中一个实例可以接管。 服务器的分组由 Pacemaker 控制，复制由 DRBD 控制。

灾难恢复复制数据队列管理器 (DR RDQM)
队列管理器在一个站点的主节点上运行，而该队列管理器的辅助实例位于另一个站点的恢复节点上。 在主实例和辅助实例之间复制数据，如果主节点由于某种原因而丢失，那么可以将辅助实例制作成主实例并启动。 这两个节点都必须是 Linux 服务器。 复制由 DRBD 控制。

灾难恢复/高可用性复制数据队列管理器 (DR/HA RDQM)
您可以配置在一个站点上的高可用性组上运行的复制数据队列管理器 (RDQM) ，但如果发生一些导致第一个组不可用的灾难，那么可以故障转移到另一个站点上的另一个高可用性组。 这称为 DR/HA RDQM。

CP4I **本机 HA**
本机 HA 是针对 IBM MQ 的容器部署的高可用性解决方案。 本机 HA 使用日志复制使在不同节点上运行的队列管理器的三个实例保持最新。 一个实例在任何时候都处于活动状态，并处理消息。 活动队列管理器将其日志更新发送到其他两个实例以保持其更新。 如果活动实例失败，那么其中一个副本实例将自动接管活动角色。

多实例队列管理器与 HA 集群之间的差异

多实例队列管理器和 HA 集群是实现队列管理器高可用性的替代方法。 以下是强调两种方法之间差异的一些要点。

多实例队列管理器包含以下功能:

- 集成到 IBM MQ 中的基本故障转移支持
- 比 HA 集群更快的故障转移
- 简单配置和操作
- 与 IBM MQ Explorer 集成

多实例队列管理器的限制包括:

- 需要高可用性，高性能的网络存储
- 更复杂的网络配置，因为队列管理器在故障转移时更改 IP 地址

HA 集群包含以下功能:

- 协调多个资源 (例如应用程序服务器或数据库) 的能力
- 更灵活的配置选项，包括包含两个以上节点的集群
- 可以多次故障转移而无需操作员干预
- 在故障转移过程中接管队列管理器的 IP 地址

HA 集群的限制包括:

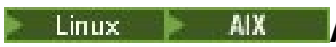
- 需要额外的产品购买和技能
- 需要可在集群节点之间切换的磁盘
- HA 集群的配置相对复杂
- 故障转移在历史上是相当缓慢的，但最近的 HA 集群产品正在对此进行改进
- 如果用于监视资源 (例如队列管理器) 的脚本中存在缺陷，那么可能会发生不必要的故障转移

HA 集群与队列管理器集群的关系

队列管理器集群提供跨队列管理器集群队列的可用实例的消息负载均衡。 这提供比单个队列管理器更高的可用性，因为在队列管理器发生故障后，消息传递应用程序仍可以向队列管理器集群队列的存活实例发送消息并访问这些实例。 但是，虽然队列管理器集群会自动将新消息路由到集群中的可用队列管理器，但当前在不可用队列管理器上排队的消息直到该队列管理器重新启动后才可用。 因此，仅队列管理器集群无法提供所有消息数据的高可用性，也无法自动检测队列管理器故障并自动触发队列管理器重新启动或故障转移。 高可用性 (HA) 集群提供这些功能。 这两种类型的集群可以一起使用，效果很好。 有关队列管理器集群的简介，请参阅 设计集群。

相关概念

IBM MQ Advanced container 的高可用性



AIX and Linux 上的 HA 集群

可以将 IBM MQ 与 AIX and Linux 平台上的高可用性 (HA) 集群配合使用: 例如， PowerHA for AIX (以前称为 HACMP)， Veritas Cluster Server， HP Serviceguard 或具有 Red Hat Cluster Suite 的 Red Hat Enterprise Linux 集群。

在 IBM WebSphere MQ 7.0.1 之前，提供了 SupportPac MC91 以帮助配置 HA 集群。 在队列管理器存储其数据的位置上， IBM WebSphere MQ 7.0.1 提供了比先前版本更大的控制程度。 这使在 HA 集群中配置队列管理器更容易。 不再需要 SupportPac MC91 随附的大多数脚本，并且将撤销 SupportPac 。

本部分介绍了 第 394 页的『HA 集群配置』， HA 集群与队列管理器集群的关系， 第 394 页的『IBM MQ Client』和 第 394 页的『IBM MQ 在 HA 集群中运行』，并指导您完成步骤和提供示例脚本，您可以通过这些脚本来配置具有 HA 集群的队列管理器。

请参阅特定于您的环境的 HA 集群文档，以获取有关本节中描述的配置步骤的帮助。

HA 集群配置

在本部分中，术语 *node* 用于引用运行操作系统和 HA 软件的实体; 在此用法中， "computer" ， "system" 或 "machine" 或 "partition" 或 "刀片" 可能被视为同义词。 您可以使用 IBM MQ 来帮助设置备用或接管配置，包括所有集群节点正在运行 IBM MQ 工作负载的相互接管。

备用 配置是最基本的 HA 集群配置，其中一个节点执行工作，而另一个节点仅充当备用节点。 备用节点不执行工作，称为空闲; 此配置有时称为 冷备用。 这样的配置需要高度的硬件冗余。 为了节省硬件，可以将此配置扩展为具有多个具有单个备用节点的工作程序节点。 这一点是备用节点可以接管任何其他工作程序节点的工作。 此配置仍称为备用配置，有时称为 "N+1" 配置。

接管 配置是一种更高级的配置，在此配置中，所有节点都执行一些工作，如果发生节点故障，那么可以接管关键工作。

单侧接管 配置是备用节点执行一些额外的非关键工作和不可移动工作的配置。 此配置类似于备用配置，但备用节点正在执行 (非关键) 工作。

相互接管 配置是所有节点都在执行高可用性 (可移动) 工作的配置。 此类型的 HA 集群配置有时也称为 "主动/主动" ，以指示所有节点都在主动处理关键工作负载。

使用扩展备用配置或任一接管配置时，请务必考虑可能放置在可接管其他节点工作的节点上的峰值负载。 此类节点必须具有足够的容量才能保持可接受的性能级别。

HA 集群与队列管理器集群的关系

队列管理器集群可减少管理，并提供跨队列管理器集群队列实例的消息负载均衡。 它们还提供比单个队列管理器更高的可用性，因为在队列管理器发生故障后，消息传递应用程序仍可访问队列管理器集群队列的存活实例。 但是，仅队列管理器集群无法自动检测队列管理器故障，也无法自动触发队列管理器重新启动或故障转移。 HA 集群提供了这些功能。 这两种类型的集群可以一起使用，效果很好。

IBM MQ Client

必须写入与可能需要重新启动或接管的队列管理器通信的 IBM MQ 客户机以容许中断的连接，并且必须反复尝试重新连接。 IBM WebSphere MQ 7 在处理客户机通道定义表 (CCDT) 时引入了有助于连接可用性和工作负载均衡的功能; 但是，在使用故障转移系统时，这些功能并不直接相关。

事务性功能允许 IBM MQ MQI client 参与两阶段事务，只要客户机连接到同一队列管理器即可。 事务功能无法使用技术 (例如， IP 负载均衡器) 从队列管理器列表中进行选择。 当您使用 HA 产品时，队列管理器将保留其身份 (名称和地址) ，无论它在哪个节点上运行，因此事务性功能可与受 HA 控制的队列管理器配合使用。

IBM MQ 在 HA 集群中运行

所有 HA 集群都具有故障转移单元的概念。 这是一组定义，其中包含构成高可用性服务的所有资源。 故障转移单元包括服务本身及其依赖的所有其他资源。

HA 解决方案对故障转移单元使用不同的术语:

- 在 PowerHA for AIX 上，故障转移单元称为 资源组。
- 在 Veritas Cluster Server 上，它称为 服务组。
- 在 Serviceguard 上，它称为 包。

本主题使用术语 资源组 表示故障转移单元。

IBM MQ 的最小故障转移单元是队列管理器。 通常，包含队列管理器的资源组还包含专供资源组使用的卷组或磁盘组中的共享磁盘，以及用于连接到队列管理器的 IP 地址。 还可以将其他 IBM MQ 资源 (例如，侦听器或同一资源组中的触发器监视器) 作为单独的资源包含在内，或者在队列管理器本身的控制下包含在内。

要在 HA 集群中使用的队列管理器必须在集群中的节点之间共享的磁盘上具有其数据和日志。 HA 集群确保一次只有集群中的一个节点可以写入磁盘。 HA 集群可以使用监视器脚本来监视队列管理器的状态。

可以将单个共享磁盘用于与队列管理器相关的数据和日志。 但是，正常做法是使用单独的共享文件系统，以便可以独立调整大小。

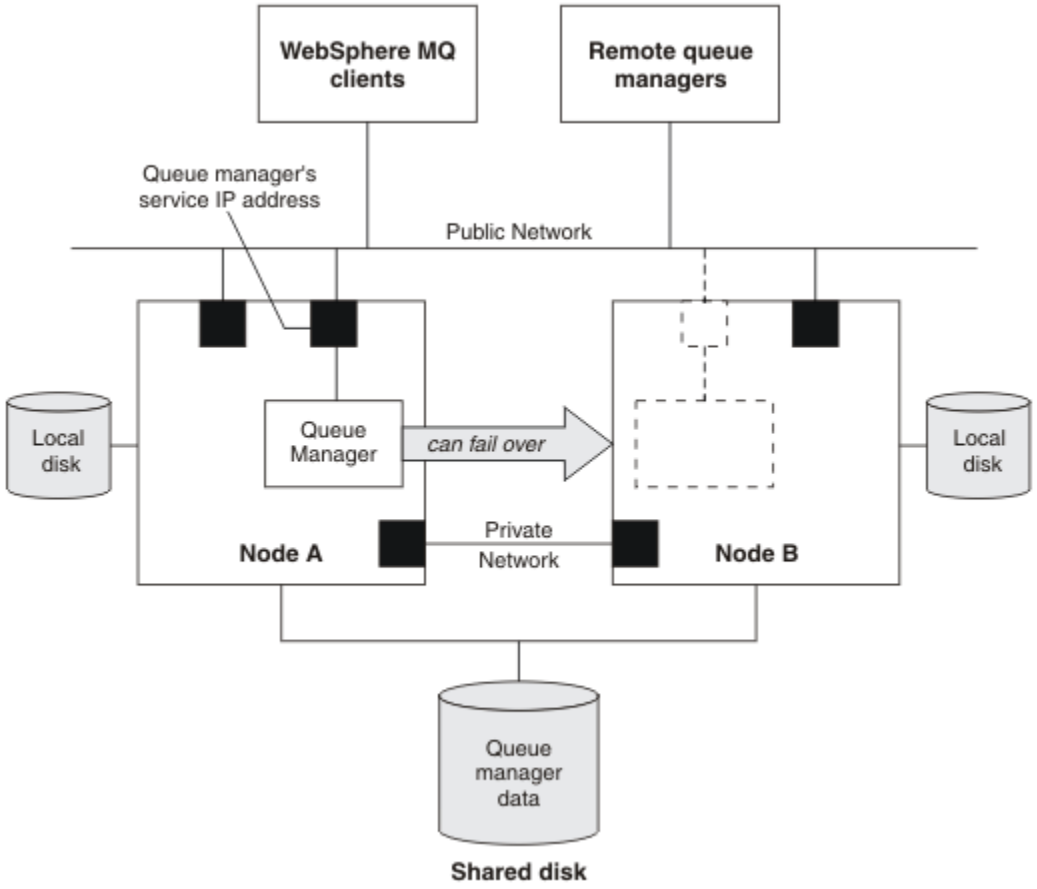


图 72: *HA* 集群

图 1 显示了具有两个节点的 HA 集群。 HA 集群正在管理已在资源组中定义的队列管理器的可用性。 这是主动/被动或冷备用配置，因为只有一个节点 (节点 A) 当前正在运行队列管理器。 队列管理器是使用其数据和日志文件在共享磁盘上创建的。 队列管理器具有也由 HA 集群管理的服务 IP 地址。 队列管理器取决于共享磁盘及其服务 IP 地址。 当 HA 集群将队列管理器从节点 A 故障转移到节点 B 时，它首先将队列管理器的从属资源移动到节点 B 上，然后启动队列管理器。

如果 HA 集群包含多个队列管理器，那么在故障转移后， HA 集群配置可能会导致两个或多个队列管理器在同一节点上运行。 HA 集群中的每个队列管理器都必须分配有自己的端口号，该端口号在任何特定时间处于活动状态的集群节点上使用。

通常， HA 集群以 root 用户身份运行。 IBM MQ 以 mqm 用户身份运行。 IBM MQ 的管理已授予 mqm 组的成员。 确保 mqm 用户和组都存在于所有 HA 集群节点上。 用户标识和组标识必须在集群中一致。 不允许 root 用户管理 IBM MQ ; 启动，停止或监视脚本的脚本必须切换到 mqm 用户。

注: 必须在所有节点上正确安装 IBM MQ ; 不能共享产品可执行文件。

Linux AIX 在 *AIX and Linux* 上配置共享磁盘

HA 集群中的 IBM MQ 队列管理器要求数据文件和日志文件位于共享磁盘上的公共指定远程文件系统中。

关于此任务

图 1 显示了 HA 集群中队列管理器的可能布局。 队列管理器的数据和日志目录都在安装在 /MQHA/QM1 上的共享磁盘上。 发生故障转移时，会在 HA 集群的节点之间切换此磁盘，以便在重新启动队列管理器的任何位置提供数据。 `mqs.ini` 文件具有 QM1 队列管理器的节。 `qm.ini` 文件中的 Log 节具有 LogPath 的值。

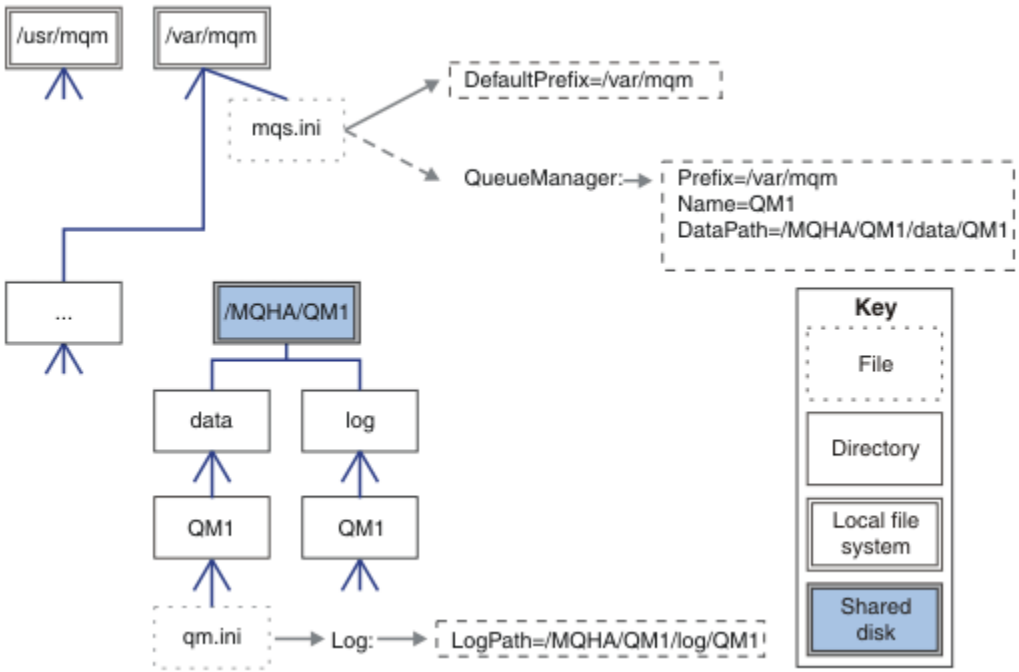


图 *73:* 共享名为 *data* 和 *log* 的目录

过程

1. 确定队列管理器的文件系统的安装点的名称。
   例如， `/MQHA/qmgrname/data` 表示队列管理器的数据文件， `/MQHA/qmgrname/log` 表示其日志文件。
2. 创建卷组 (或磁盘组) 以包含队列管理器的数据和日志文件。
   此卷组由与队列管理器位于同一资源组中的高可用性 (HA) 集群管理。
3. 在卷组中为队列管理器的数据和日志文件创建文件系统。
4. 依次为每个节点创建文件系统的安装点，并确保可以安装文件系统。
   mqm 用户必须拥有安装点。

Linux AIX 在 *AIX and Linux* 上创建 *HA* 集群队列管理器

在高可用性集群中使用队列管理器的第一步是在其中一个节点上创建队列管理器。

关于此任务

要创建要在 HA 集群中使用的队列管理器，必须首先选择集群中要在其中创建队列管理器的某个节点，然后在此节点上完成以下步骤。

过程

1. 在节点上安装队列管理器的文件系统。
2. 使用 **crtmqm** 命令创建队列管理器。
   例如:

   ```
   crtmqm -md /MQHA/qmgrname/data -ld /MQHA/qmgrname/log qmgrname
   ```
3. 使用 **strmqm** 命令手动启动队列管理器。
4. 完成队列管理器的任何初始配置，例如创建队列和通道，并将队列管理器设置为在队列管理器启动时自动启动侦听器。
5. 使用 **endmqm** 命令停止队列管理器。

6. 使用 **dspmqinf** 命令来显示 **addmqinf** 命令:

   ```
   dspmqinf -o command qmgrname
   ```

   其中， qmgrname 是队列管理器的名称。

   有关使用 **addmqinf** 命令的更多信息，请参阅 第 397 页的『将队列管理器配置添加到 AIX and Linux 上的其他 HA 集群节点』。

   以类似于以下示例的方式显示 **addmqinf** 命令:

   ```
   addmqinf -sQueueManager -vName=qmgrname -vDirectory=qmgrname \
   -vPrefix=/var/mqm -vDataPath=/MQHA/qmgrname/data/qmgrname
   ```

7. 请仔细记录显示的命令。
8. 卸装队列管理器的文件系统。

下一步做什么

现在，您已准备好完成 第 397 页的『将队列管理器配置添加到 AIX and Linux 上的其他 HA 集群节点』中描述的步骤。

Linux AIX 将队列管理器配置添加到 *AIX and Linux* 上的其他 *HA* 集群节点

必须将队列管理器配置信息添加到 HA 集群中的其他节点。

开始之前

在完成此任务之前，必须已完成 第 396 页的『在 AIX and Linux 上创建 HA 集群队列管理器』中的步骤。 创建队列管理器后，必须通过在每个其他节点上完成以下步骤，将队列管理器的配置信息添加到 HA 集群中的每个其他节点。

关于此任务

创建要在 HA 集群中使用的队列管理器时，必须首先选择集群中要在其中创建队列管理器的节点之一，如 第 396 页的『在 AIX and Linux 上创建 HA 集群队列管理器』中所述。

过程

1. 安装队列管理器文件系统。
2. 将队列管理器配置信息添加到节点。

   可以通过两种方法来添加配置信息:

   - 通过直接编辑 `/var/mqm/mqs.ini` 。
   - 通过发出 **dspmqinf** 命令在 第 396 页的『在 AIX and Linux 上创建 HA 集群队列管理器』中的步骤 6 中显示的 **addmqinf** 命令。

3. 启动和停止队列管理器以验证配置。

   必须从与 **addmqinf** 命令相同的 IBM MQ 安装发出用于启动和停止队列管理器的命令。 要从与队列管理器当前关联的安装不同的安装启动和停止队列管理器，必须首先使用 **setmqm** 命令设置与队列管理器关联的安装。 有关更多信息，请参阅 setmqm。

4. 卸装队列管理器文件系统。

Linux AIX 用于在 *AIX and Linux* 上启动 *HA* 集群队列管理器的示例 *shell* 脚本

队列管理器在 HA 集群中表示为资源。 HA 集群必须能够启动和停止队列管理器。 在大多数情况下，您可以使用 shell 脚本来启动队列管理器。 必须使用网络文件系统或通过将这些脚本复制到每个本地磁盘，使这些脚本在集群中的所有节点上的相同位置可用。

注: 在重新启动失败的队列管理器之前，必须断开应用程序与该队列管理器实例的连接。 如果不执行此操作，那么队列管理器可能无法正确重新启动。

此处提供了合适的 shell 脚本的示例。 您可以根据需要进行定制，并在 HA 集群的控制下使用这些定制来启动队列管理器。

以下 shell 脚本是如何从 HA 集群用户切换到 mqm 用户以便能够成功启动队列管理器的示例:

```
#!/bin/ksh

# A simple wrapper script to switch to the mqm user.

su mqm -c name_of_your_script $*
```

以下 shell 脚本是如何在不对队列管理器的当前状态进行任何假定的情况下启动队列管理器的示例。 请注意，它使用极其突然的方法来结束属于队列管理器的任何进程:

```
#!/bin/ksh
#
# This script robustly starts the queue manager.
#
# The script must be run by the mqm user.

# The only argument is the queue manager name. Save it as QM variable
QM=$1

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

# End any queue manager processes which might be running.

srchstr="( |-m)$QM *.*$"
for process in amqzmuc0 amqzxma0 amqfcxba amqfqpub amqpcsea amqzlaa0 \
               amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqrmppa \
               amqzfuma amqzmuf0 amqzmur0 amqzmgr0
 do
  ps -ef | tr "\t" " " | grep $process | grep -v grep | \
    egrep "$srchstr" | awk '{print $2}'| \
       xargs kill -9 > /dev/null 2>&1
done

# It is now safe to start the queue manager.
# The strmqm command does not use the -x flag.
strmqm ${QM}
```

您可以修改脚本以启动其他相关程序。

Linux AIX 用于在 *AIX and Linux* 上停止 *HA* 集群队列管理器的示例 *shell* 脚本

在大多数情况下，您可以使用 shell 脚本来停止队列管理器。 此处提供了合适的 shell 脚本的示例。 您可以根据需要定制这些属性，并使用这些属性来停止受 HA 集群控制的队列管理器。

以下脚本是如何立即停止队列管理器而不假定该队列管理器的当前状态的示例。 该脚本必须由 mqm 用户运行。 因此，可能需要将此脚本合并到 shell 脚本中，以将用户从 HA 集群用户切换到 mqm。 (第 397 页的『用于在 AIX and Linux 上启动 HA 集群队列管理器的示例 shell 脚本』中提供了示例 shell 脚本。)

```
#!/bin/ksh
#
# The script ends the QM by using two phases, initially trying an immediate
# end with a time-out and escalating to a forced stop of remaining
# processes.
#
# The script must be run by the mqm user.
#
# There are two arguments: the queue manager name and a timeout value.
QM=$1
TIMEOUT=$2

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi
```

```
if [ -z "$TIMEOUT" ]
then
  echo "ERROR! No timeout specified"
  exit 1
fi

for severity in immediate brutal
do
  # End the queue manager in the background to avoid
  # it blocking indefinitely. Run the TIMEOUT timer
  # at the same time to interrupt the attempt, and try a
  # more forceful version. If the brutal version fails,
  # nothing more can be done here.

  echo "Attempting ${severity} end of queue manager '${QM}'"
  case $severity in

  immediate)
    # Minimum severity of endmqm is immediate which severs connections.
    # HA cluster should not be delayed by clients
    endmqm -i ${QM} &
    ;;

  brutal)
    # This is a forced means of stopping queue manager processes.

    srchstr="( |-m)$QM *.*$"
    for process in amqzmuc0 amqzxma0 amqfcxba amqfqpub amqpcsea amqzlaa0 \
                amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqrmppa \
                amqzfuma amqzmuf0 amqzmur0 amqzmgr0
    do
      ps -ef | tr "\t" " " | grep $process | grep -v grep | \
         egrep "$srchstr" | awk '{print $2}'| \
            xargs kill -9 > /dev/null 2>&1
    done

  esac

  TIMED_OUT=yes
  SECONDS=0
  while (( $SECONDS < ${TIMEOUT} ))
  do
   TIMED_OUT=yes
   i=0
   while [ $i -lt 5 ]
   do
     # Check for execution controller termination
     srchstr="( |-m)$QM *.*$"
     cnt=`ps -ef | tr "\t" " " | grep amqzxma0 | grep -v grep | \
       egrep "$srchstr" | awk '{print $2}' | wc -l `
     i=`expr $i + 1`
     sleep 1
     if [ $cnt -eq 0 ]
     then
       TIMED_OUT=no
       break
     fi
   done

   if [ ${TIMED_OUT} = "no" ]
   then
     break
   fi

   echo "Waiting for ${severity} end of queue manager '${QM}'"
   sleep 1
  done # timeout loop

  if [ ${TIMED_OUT} = "yes" ]
  then
    continue        # to next level of urgency
  else
    break           # queue manager is ended, job is done
  fi

done # next phase
```

注: 根据为特定队列管理器运行的进程，此脚本中包含的队列管理器进程的列表可能不是完整列表，或者可能包含的进程数超过为该队列管理器运行的进程数:

```
for process in amqzmuc0 amqzxma0 amqfcxba amqfqpub amqpcsea amqzlaa0 \
               amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqrmppa \
               amqzfuma amqzmuf0 amqzmur0 amqzmgr0
```

可以根据配置的功能部件以及为特定队列管理器运行的进程，在此列表中包含或排除进程。 有关进程的完整列表以及有关按特定顺序停止进程的信息，请参阅 在 UNIX 和 Linux 上手动停止队列管理器。

Linux AIX 在 *AIX and Linux* 上监视 *HA* 集群队列管理器

通常为高可用性 (HA) 集群提供一种方法来定期监视队列管理器的状态。 在大多数情况下，您可以使用 shell 脚本来执行此操作。 此处提供了合适的 shell 脚本的示例。 您可以根据需要定制这些脚本，并使用它们来进行特定于环境的其他监视检查。

从 IBM WebSphere MQ 7.1 开始，可以在系统上共存多个 IBM MQ 安装。 有关多个安装的更多信息，请参阅 多个安装。 如果您打算在多个安装 (包括 IBM WebSphere MQ 7.1 或更高版本的安装) 中使用监视脚本，那么可能需要执行一些其他步骤。 如果您具有主安装，或者正在使用版本低于 IBM WebSphere MQ 7.1 的脚本，那么不需要指定 *MQ_INSTALLATION_PATH* 即可使用该脚本。 否则，以下步骤将确保正确识别 *MQ_INSTALLATION_PATH* :

1. 从 IBM WebSphere MQ 7.1 安装使用 **crtmqenv** 命令来标识队列管理器的正确 *MQ_INSTALLATION_PATH* :

   ```
   crtmqenv -m qmname
   ```

   对于由 *qmname* 指定的队列管理器，此命令返回正确的 *MQ_INSTALLATION_PATH* 值。
2. 使用相应的 *qmname* 和 *MQ_INSTALLATION_PATH* 参数运行监视脚本。

注: PowerHA for AIX 不提供向队列管理器的监视程序提供参数的方法。 必须为每个队列管理器创建单独的监视程序，以封装队列管理器名称。 以下是在 AIX 上用于封装队列管理器名称的脚本示例:

```
#!/bin/ksh
su mqm -c name_of_monitoring_script qmname  MQ_INSTALLATION_PATH
```

其中 *MQ_INSTALLATION_PATH* 是可选参数，用于指定与队列管理器 *qmname* 相关联的 IBM MQ 的安装路径。

以下脚本对 **runmqsc** 挂起的可能性不稳健。 通常， HA 集群会将挂起的监视脚本视为故障，并且其本身对这种可能性非常稳健。

但是，该脚本允许队列管理器处于启动状态。 这是因为 HA 集群通常在启动队列管理器后立即开始对其进行监视。 某些 HA 集群区分资源的启动阶段和运行阶段，但需要配置启动阶段的持续时间。 由于启动队列管理器所花费的时间取决于它必须执行的工作量，因此很难选择启动队列管理器所花费的最大时间。 如果选择的值过低，那么 HA 集群会错误地假定队列管理器在未完成启动时失败。 这可能导致无尽的故障转移序列。

此脚本必须由 mqm 用户运行; 因此，可能需要将此脚本合并到 shell 脚本中，以将用户从 HA 集群用户切换到 mqm (在 第 397 页的『用于在 AIX and Linux 上启动 HA 集群队列管理器的示例 shell 脚本』 中提供了示例 shell 脚本):

```
#!/bin/ksh
#
# This script tests the operation of the queue manager.
#
# An exit code is generated by the runmqsc command:
# 0  => Either the queue manager is starting or the queue manager is running and responds.
#       Either is OK.
# >0 => The queue manager is not responding and not starting.
#
# This script must be run by the mqm user.
QM=$1
MQ_INSTALLATION_PATH=$2

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

if [ -z "$MQ_INSTALLATION_PATH" ]
then
```

```
  # No path specified, assume system primary install or MQ level < 7.1.0.0
  echo "INFO: Using shell default value for MQ_INSTALLATION_PATH"
else
  echo "INFO: Prefixing shell PATH variable with $MQ_INSTALLATION_PATH/bin"
  PATH=$MQ_INSTALLATION_PATH/bin:$PATH
fi

# Test the operation of the queue manager. Result is 0 on success, non-zero on error.
echo "ping qmgr" | runmqsc ${QM} > /dev/null 2>&1
pingresult=$?

if [ $pingresult -eq 0 ]
then # ping succeeded

  echo "Queue manager '${QM}' is responsive"
  result=0

else # ping failed

  # Don't condemn the queue manager immediately, it might be starting.
  srchstr="( |-m)$QM *.*$"
  cnt=`ps -ef | tr "\t" " " | grep strmqm | grep "$srchstr" | grep -v grep \
              | awk '{print $2}' | wc -l`
  if [ $cnt -gt 0 ]
  then
    # It appears that the queue manager is still starting up, tolerate
    echo "Queue manager '${QM}' is starting"
    result=0
  else
    # There is no sign of the queue manager starting
    echo "Queue manager '${QM}' is not responsive"
    result=$pingresult
  fi

fi

exit $result
```

Linux AIX 将队列管理器置于 *AIX and Linux* 上的 *HA* 集群控制下

必须使用队列管理器的 IP 地址和共享磁盘来配置受 HA 集群控制的队列管理器。

关于此任务

要使队列管理器受 HA 集群控制，必须定义资源组以包含队列管理器及其所有关联资源。

过程

1. 创建包含队列管理器，队列管理器的卷或磁盘组以及队列管理器的 IP 地址的资源组。
   IP 地址是虚拟 IP 地址，而不是计算机的 IP 地址。
2. 验证 HA 集群是否正确在集群节点之间切换资源并准备好控制队列管理器。

Linux AIX 在 *AIX and Linux* 上删除 *HA* 集群队列管理器

您可能希望从不再需要运行队列管理器的节点中除去队列管理器。

关于此任务

要从 HA 集群中的节点除去队列管理器，必须除去其配置信息。

过程

1. 从 HA 集群中除去节点，以便 HA 集群不再尝试在此节点上激活队列管理器。
2. 使用以下 **rmvmqinf** 命令来除去队列管理器的配置信息:

   ```
   rmvmqinf qmgrname
   ```

3. 可选： 要完全删除队列管理器，请使用 **dltmqm** 命令。

   要点: 请注意，使用 **dltmqm** 命令删除队列管理器将完全删除队列管理器的数据和日志文件。

删除队列管理器后，可以使用 **rmvmqinf** 命令从其他节点中除去剩余的配置信息。

Windows 支持 Microsoft Cluster Service (MSCS)

引入并设置 MSCS 以支持虚拟服务器的故障转移。 MSCS 也称为 Windows Server Failover Clustering (WSFC)。

此信息仅适用于 IBM MQ for Windows 。

注: 从 Windows Server 2016 开始， Microsoft Cluster Service (MSCS) 的新名称为 Windows Server Failover Clustering (WSFC)。

MSCS/WSFC 使您能够将服务器连接到集群中，从而提高数据和应用程序的可用性，并使系统更易于管理。 MSCS/WSFC 可以自动检测并从服务器或应用程序故障中恢复。

MSCS/WSFC 支持虚拟服务器的故障转移，这些服务器对应于应用程序， Web 站点，打印队列或文件共享 (例如，包括其磁盘主轴，文件和 IP 地址)。

故障转移 是一个过程， MSCS/WSFC 在集群中的一台计算机上检测应用程序中的故障，并有序地关闭中断的应用程序，将其状态数据传输到另一台计算机，然后在那里重新启动应用程序。

有关如何配置和使用故障转移集群的信息，请参阅子主题。

Windows 介绍 *MSCS* 集群

MSCS 集群 (也称为 WSFC 集群) 是由两台或更多台计算机组成的组，它们连接在一起并以如下方式进行配置: 如果发生故障， MSCS 将执行 故障转移，将应用程序的状态数据从发生故障的计算机转移到集群中的另一台计算机，然后在该集群中重新启动其操作。

注: 从 Windows Server 2016 开始， Microsoft Cluster Service (MSCS) 的新名称为 Windows Server Failover Clustering (WSFC)。

第 392 页的『高可用性配置』 包含 MSCS 集群，多实例队列管理器和 IBM MQ 集群之间的比较。

在此部分及其下级主题中，术语 *cluster* 由其自身使用时， **always** 表示 MSCS 集群。 这与本指南中其他地方描述的 IBM MQ 集群不同。

双机集群由两台计算机 (例如 A 和 B) 组成，这两台计算机使用 虚拟 *IP* 地址共同连接到网络以进行客户机访问。 它们也可能通过一个或多个专用网络互相连接。 A 和 B 至少共享一个磁盘，以供每一台计算机上要使用的服务器应用程序使用。 还有另一个共享磁盘，它必须是独立磁盘的冗余阵列 (*RAID*) 级别 1 ，专门用于 MSCS; 这称为 定额 磁盘。 MSCS 监视这两台计算机以检查硬件和软件是否在正常运行。

在这样的简单设置中，两台计算机上都安装了所有应用程序，但只有计算机 A 与实时应用程序一起运行; 计算机 B 只是在运行和等待。 如果计算机 A 迁到一系列问题中的任何一个问题， MSCS 会有序地关闭中断的应用程序，将其状态数据转移到另一台计算机，并在那里重新启动应用程序。 这称为 故障转移。 可以使应用程序 集群感知 ，以便它们与 MSCS 完全交互并正常进行故障转移。

包含两台计算机的集群的典型设置如第 403 页的图 74 所示。

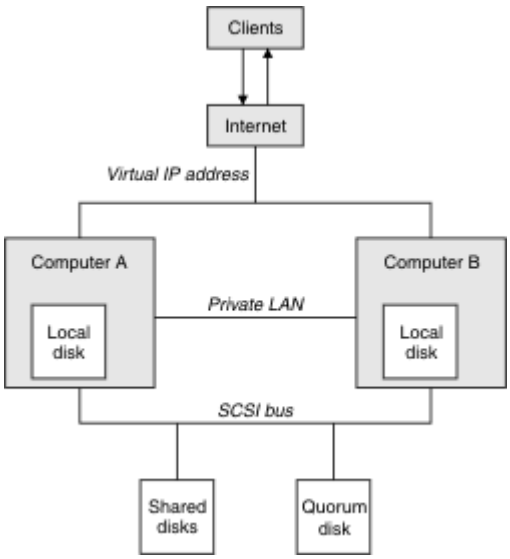


图 *74:* 双计算机 *MSCS* 集群

每台计算机都可以访问共享磁盘，但在 MSCS 控制下一次只能有一台访问。 在故障转移事件中，MSCS 会将访问切换到其他计算机。 共享磁盘自身通常是 RAID，但不是必需的。

每台计算机都连接到外部网络以便客户机访问，并且都有 IP 地址。 但是，与此集群通信的外部客户机仅知道一个 虚拟 *IP* 地址， MSCS 会相应地路由集群中的 IP 流量。

MSCS 还通过一个或多个专用连接或公用网络在两台计算机之间执行自己的通信，例如使用脉动信号监视它们的状态并同步它们的数据库。

Windows 为 *MSCS* 集群设置 *IBM MQ*

通过使队列管理器成为到 MSCS 的故障转移单元，可以配置 IBM MQ 以进行集群。 您将队列管理器定义为 MSCS 的资源，然后 MSCS 可以对其进行监视，并在发生问题时将其传输到集群中的另一台计算机。

注: 从 Windows Server 2016 开始， Microsoft Cluster Service (MSCS) 的新名称为 Windows Server Failover Clustering (WSFC)。

要为此设置系统，请首先在集群中的每台计算机上安装 IBM MQ 。

由于队列管理器与 IBM MQ 安装名称相关联，因此集群中所有计算机上的 IBM MQ 安装名称应该相同。 请参阅 安装和卸载。

队列管理器本身仅需要存在于您创建它们的计算机上。 发生故障转移时， MSCS 会在另一台计算机上启动队列管理器。 但是，队列管理器必须将其日志和数据文件放在集群共享磁盘上，而不是本地驱动器上。 如果已在本地驱动器上安装了队列管理器，那么可以使用 IBM MQ 随附的工具对其进行迁移; 请参阅 第 405 页的『将队列管理器移至 MSCS 存储器』。 如果要创建新的队列管理器以用于 MSCS ，请参阅 第 405 页的『创建队列管理器以用于 MSCS』。

安装和迁移后，使用 MSCS 集群管理员使 MSCS 了解您的队列管理器; 请参阅 第 406 页的『将队列管理器置于 MSCS 控制之下』。

如果您决定从 MSCS 控制中除去队列管理器，请使用 第 411 页的『从 MSCS 控件中除去队列管理器』中描述的过程。

Windows 设置对称性和 *MSCS*

当应用程序从一个节点切换到另一个节点时，它必须以相同的方式运行，而不考虑节点。 确保这一点的最佳方法是使环境相同。

如果可以，请在每台计算机上设置具有相同硬件，操作系统软件，产品软件和配置的集群。 特别是，请确保两台计算机上安装的所有必需软件在版本，维护级别， SupportPacs，路径和出口方面都相同，并确保存在公共名称空间 (安全环境) ，如 第 404 页的『MSCS 安全性』中所述。

Windows *MSCS* 安全性

对于成功的 MSCS 安全性，请遵循以下准则。

准则如下:

- 确保在集群中的每台计算机上都有相同的软件安装。
- 在集群中创建公共名称空间 (安全环境)。
- 创建域的 MSCS 集群成员的节点，其中作为 集群所有者 的用户帐户是域帐户。
- 使集群上的其他用户帐户也具有域帐户，以便它们在两个节点上都可用。 如果您已有域，并且与 IBM MQ 相关的帐户是域帐户，那么将自动执行此操作。 如果当前没有域，请考虑设置 迷你域 以满足集群节点和相关帐户的需求。 您的目标是使两个计算机的集群看起来像单个计算资源。

  请记住，一台计算机的本地帐户在另一台计算机上不存在。 即使您在另一台计算机上创建具有相同名称的帐户，其安全标识 (SID) 也不同，因此，当您的应用程序移动到另一个节点时，该节点上不存在许可权。

在故障转移或移动期间， IBM MQ MSCS 支持可确保包含队列管理器对象的所有文件都具有对目标节点的同等许可权。 显式地，代码会检查管理员和 mqm 组以及 SYSTEM 帐户是否具有完全控制权，如果 Everyone 对旧节点具有读访问权，那么会在目标节点上添加该许可权。

您可以使用域帐户来运行 IBM MQ 服务。 确保它存在于集群中每台计算机上的本地 mqm 组中。

Windows 将多个队列管理器与 *MSCS* 配合使用

如果在一台计算机上运行多个队列管理器，那么可以选择其中一个设置。

设置如下所示:

- 单个组中的所有队列管理器。 在此配置中，如果任何队列管理器发生问题，那么将组中的所有队列管理器作为组故障转移到另一台计算机。
- 每个组中的单个队列管理器。 在此配置中，如果队列管理器发生问题，那么仅会将其故障转移到其他计算机，而不会影响其他队列管理器。
- 前两种设置的混合。

Windows 集群方式和 *MSCS*

您可以使用 Windows 上的 IBM MQ 运行集群系统的两种方式: 主动/被动或主动/主动。

注: 如果要将 MSCS 与 Microsoft Transaction Server (COM +) 一起使用，那么不能使用主动/主动方式。

主动/被动方式

在主动/被动方式下，计算机 A 具有正在运行的应用程序，而计算机 B 是备份，仅当 MSCS 检测到问题时才使用。

只能将此方式与一个共享磁盘配合使用，但是，如果任何应用程序导致故障转移，那么必须将 **所有** 应用程序作为一个组进行传输 (因为一次只有一台计算机可以访问共享磁盘)。

您可以将 MSCS 配置为使用 A 作为 首选 计算机。 然后，当计算机 A 已修复或更换并再次正常工作时，MSCS 会检测到此情况并自动将应用程序切换回计算机 A。

如果运行多个队列管理器，请考虑为每个队列管理器都有一个单独的共享磁盘。 然后将每个队列管理器放在 MSCS 中的单独组中。 这样，任何队列管理器都可以故障转移到其他计算机，而不会影响其他队列管理器。

主动/主动方式

在主动/主动方式下，计算机 A 和 B 都有正在运行的应用程序，并且每台计算机上的组都设置为使用另一台计算机作为备份。 如果在计算机 A 上检测到故障， MSCS 会将状态数据传输到计算机 B ，并在那里重新启动应用程序。 然后计算机 B 运行自己的应用程序和 A 的应用程序。

对于此设置，您至少需要两个共享磁盘。 您可以将 MSCS 配置为 A 作为 A 应用程序的首选计算机，将 B 配置为 B 应用程序的首选计算机。 故障转移和修复后，每个应用程序都会自动回到自己的计算机上。

对于 IBM MQ，这意味着您可以运行两个队列管理器，例如，在 A 和 B 中的每个队列管理器上运行一个队列管理器，每个队列管理器利用其自己计算机的全部电源。 在计算机 A 上发生故障后，两个队列管理器都将在计算机 B 上运行。 这将意味着共享一台计算机的电源，降低了以速度处理大量数据的能力。 但是，在查找并修复 A 上的故障时，关键应用程序仍可用。

Windows 创建队列管理器以用于 *MSCS*

此过程确保以适合在 Microsoft Cluster Service (MSCS) 控制下进行准备和放置的方式创建新的队列管理器。

注: 从 Windows Server 2016 开始， Microsoft Cluster Service (MSCS) 的新名称为 Windows Server Failover Clustering (WSFC)。

首先在本地驱动器上创建队列管理器及其所有资源，然后将日志文件和数据文件迁移到共享磁盘。 (您可以撤销此操作。) **请勿** 尝试在共享驱动器上使用其资源创建队列管理器。

您可以通过两种方式 (从命令提示符或在 IBM MQ Explorer 中) 创建用于 MSCS 的队列管理器。 使用命令提示符的优点是创建了 *stopped* 队列管理器，并将其设置为 手动启动，这已准备好用于 MSCS。 (IBM MQ Explorer 会自动启动新的队列管理器，并在创建后将其设置为自动启动。 您必须对此进行更改。)

从命令提示符创建队列管理器

遵循以下步骤从命令提示符创建队列管理器，以便与 MSCS 配合使用:

1. 确保将环境变量 MQSPREFIX 设置为引用本地驱动器，例如 C:\IBM MQ。 如果您对此进行了更改，请重新引导机器，以便系统帐户能够进行更改。 如果未设置此变量，那么将在队列管理器的 IBM MQ 缺省目录中创建队列管理器。
2. 使用 **crtmqm** 命令创建队列管理器。 例如，要在缺省目录中创建名为 mscs_test 的队列管理器，请使用:

   ```
   crtmqm mscs_test
   ```

3. 转至第 405 页的『将队列管理器移至 MSCS 存储器』。

使用 IBM MQ Explorer 创建队列管理器

执行以下步骤以使用 IBM MQ Explorer 创建队列管理器，以便与 MSCS 配合使用:

1. 从 "开始" 菜单启动 IBM MQ Explorer 。
2. 在 " Navigator " 视图中，展开树节点以查找 **队列管理器** 树节点。
3. 右键单击 **队列管理器** 树节点，然后选择**新建** > **队列管理器**。 将显示 "创建队列管理器" 面板。
4. 完成对话框 (步骤 1)，然后单击 **下一步>**。
5. 完成对话框 (步骤 2)，然后单击 **下一步>**。
6. 完成对话框 (步骤 3)，确保未选择 **启动队列管理器** 和 **创建服务器连接通道**，然后单击 **下一步>**。
7. 完成对话框 (步骤 4)，然后单击 **完成**。
8. 转至第 405 页的『将队列管理器移至 MSCS 存储器』。

Windows 将队列管理器移至 *MSCS* 存储器

此过程将配置现有队列管理器，以使其适合置于 Microsoft Cluster Service (MSCS) 控制之下。

要实现此目的，请将日志文件和数据文件移至共享磁盘，以便在发生故障时可供其他计算机使用。 例如，现有队列管理器可能具有 C:\WebSphere MQ\log*QMname* 和 C:\WebSphere MQ\qmgrs*QMname* 之类的路径。

注意: 请勿尝试手动移动文件; 请使用作为 IBM MQ MSCS 支持的一部分提供的实用程序，如本主题中所述。

如果要移动的队列管理器使用 TLS 连接，并且 TLS 密钥存储库位于本地机器上的队列管理器数据目录中，那么该密钥存储库将与队列管理器的其余部分一起移动到共享磁盘。 缺省情况下，用于指定 TLS 密钥存储库位置 SSLKEYR 的队列管理器属性设置为队列管理器数据目录下的 *MQ_INSTALLATION_PATH*\qmgrs\QMGRNAME\ssl\key。 *MQ_INSTALLATION_PATH* 表示 IBM MQ 安装

所在的高级目录。 hamvmqm 命令不会修改此队列管理器属性。 在此情况下，必须使用 IBM MQ Explorer 或 MQSC 命令 ALTER QMGR 修改队列管理器属性 SSLKEYR 以指向新的 TLS 密钥存储库文件。

注: 从 Windows Server 2016 开始， Microsoft Cluster Service (MSCS) 的新名称为 Windows Server Failover Clustering (WSFC)。

程序如下:

1. 关闭队列管理器，并检查是否没有错误。
2. 如果队列管理器的日志文件或队列文件已存储在共享磁盘上，请跳过此过程的其余部分并直接进入 第 406 页的『将队列管理器置于 MSCS 控制之下』。
3. 对队列文件和日志文件进行完全介质备份，并将备份存储在安全的位置 (请参阅 第 413 页的『队列管理器日志文件』 以了解这很重要的原因)。
4. 如果您已有合适的共享磁盘资源，请继续执行步骤 6。 否则，请使用 MSCS 集群管理员来创建类型为 共享磁盘 的资源，该资源具有足够的容量来存储队列管理器日志文件和数据 (队列) 文件。
5. 通过使用 MSCS 集群管理员将共享磁盘从一个集群节点移动到另一个集群节点，然后再次进行测试。
6. 确保共享磁盘在本地存储队列管理器日志和数据文件的集群节点上处于联机状态。
7. 运行实用程序以移动队列管理器，如下所示:

   ```
   hamvmqm /m qmname /dd " e: \
   IBM MQ " /ld " e: \
   IBM MQ \log"
   ```

   将队列管理器名称替换为 *qmname*，将共享磁盘盘符替换为 *e*，并将所选目录替换为 *IBM MQ*。 如果这些目录尚不存在，那么将创建这些目录。
8. 使用 IBM MQ Explorer 测试队列管理器以确保其正常工作。 例如:
   a. 右键单击队列管理器树节点，然后选择 **启动**。 队列管理器将启动。
   b. 右键单击 **队列** 树节点，然后选择 **新建** > **本地队列 ...**。 并为队列提供名称。
   c. 单击**完成**。
   d. 右键单击队列，然后选择 **放置测试消息 ...**。 此时将显示 "放置测试消息" 面板。
   e. 输入一些消息文本，然后单击 **放置测试消息**，然后关闭面板。
   f. 右键单击队列，然后选择 **浏览消息 ...**。 将显示 "消息浏览器" 面板。
   g. 确保消息位于队列上，然后单击 **关闭**。 "消息浏览器" 面板将关闭。
   h. 右键单击队列，然后选择 **清除消息 ...**。 将清除队列上的消息。
   i. 右键单击队列，然后选择 **删除 ...**。 将显示确认面板，单击 **确定**。 此时会删除该队列。
   j. 右键单击队列管理器树节点，然后选择 **停止 ...**。 将显示 "结束队列管理器" 面板。
   k. 单击**确定**。 队列管理器停止。
9. 以 IBM MQ 管理员身份确保将队列管理器的启动属性设置为手动。 在 IBM MQ Explorer 中，将 "启动" 字段设置为队列管理器属性面板中的 manual 。
10. 转至第 406 页的『将队列管理器置于 MSCS 控制之下』。

Windows 将队列管理器置于 *MSCS* 控制之下

如何将队列管理器置于 Microsoft Cluster Service (MSCS) 控制下，包括先决条件任务。

注: 从 Windows Server 2016 开始， Microsoft Cluster Service (MSCS) 的新名称为 Windows Server Failover Clustering (WSFC)。

在将队列管理器置于 MSCS/WSFC 控制之下之前

在将队列管理器置于 MSCS/WSFC 控制之下之前，请完成以下步骤:

1. 确保 IBM MQ 及其 MSCS/WSFC 支持安装在集群中的两台机器上，并且每台计算机上的软件都相同，如第 403 页的『为 MSCS 集群设置 IBM MQ』中所述。

2. 使用 **haregtyp** 实用程序将 IBM MQ 注册为所有集群节点上的 MSCS 资源类型。 请参阅第 415 页的『支持 MSCS 实用程序』。
3. 如果尚未执行此操作，请 创建队列管理器以用于 MSCS/WSFC。
4. 如果您已创建队列管理器，或者该队列管理器已存在，请确保已执行 第 405 页的『将队列管理器移至 MSCS 存储器』中的过程。
5. 如果队列管理器正在运行，请使用命令提示符或 IBM MQ Explorer 将其停止。
6. 在继续执行本主题中的以下任何 Windows 过程之前，请测试共享驱动器的 MSCS/WSFC 操作。

Windows Server 2012， 2016 ， 2019 或 2022

要在 Windows Server 2012 或更高版本上将队列管理器置于 MSCS/WSFC 控制下，请使用以下过程:

1. 登录到托管队列管理器的集群节点计算机，或者以具有集群管理许可权的用户身份登录到远程工作站，并连接到托管队列管理器的集群节点。
2. 启动 "故障转移集群管理工具"。
3. 右键单击 **故障转移集群管理> 连接集群 ...** 以打开与集群的连接。
4. 与 MSCS 集群管理员在先前版本的 Windows 上使用的组方案不同，故障转移集群管理工具使用服务和应用程序的概念。 已配置的服务或应用程序包含要进行集群的一个应用程序所需的所有资源。 您可以在 WSFC 下配置队列管理器，如下所示:
   a. 右键单击集群，然后选择 **配置角色** 以启动配置向导。
   b. 在 "选择服务或应用程序" 面板上选择 **其他服务器** 。
   c. 选择相应的 IP 地址作为客户机访问点。

      此地址应该是客户机和其他队列管理器用于连接到 虚拟 队列管理器的未使用 IP 地址。 此 IP 地址不是任一节点的正常 (静态) 地址; 它是它们之间 浮动 的额外地址。 虽然 WSFC 处理此地址的路由，但它 **不会** 验证是否可以访问该地址。
   d. 分配存储设备以供队列管理器独占使用。 需要先创建此设备作为资源实例，然后才能对其进行分配。

      您可以使用一个驱动器来存储日志和队列文件，也可以将它们拆分为多个驱动器。 在任何一种情况下，如果每个队列管理器都有自己的共享磁盘，请确保此队列管理器所使用的所有驱动器都是此队列管理器所独有的，即，没有任何其他驱动器依赖于这些驱动器。 还要确保为队列管理器使用的每个驱动器创建资源实例。

      磁带机的资源类型取决于您正在使用的 SCSI 支持; 请参阅 SCSI 适配器指示信息。 可能已存在每个共享驱动器的组和资源。 如果是这样，那么不需要为每个驱动器创建资源实例。 将其从当前组移至为队列管理器创建的组。

      对于每个驱动器资源，请将可能的所有者设置为两个节点。 将从属资源设置为 "无"。
   e. 在 "选择资源类型" 面板上选择 **MQSeries MSCS** 资源。
   f. 完成向导中的其余步骤。
5. 在使资源联机之前， MQSeries MSCS 资源需要其他配置:
   a. 选择新定义的服务，该服务包含名为 "New MQSeries MSCS" 的资源。
   b. 右键单击 MQ 资源上的 **属性** 。
   c. 配置资源:
      - `Name` ; 选择一个便于识别它所针对的队列管理器的名称。
      - `Run in a separate Resource Monitor` ; 实现更好的隔离
      - `Possible owners` ; 设置两个节点
      - `Dependencies` ; 添加此队列管理器的驱动器和 IP 地址。

        警告: 未能添加这些依赖关系意味着 IBM MQ 尝试在故障转移期间将队列管理器状态写入错误的集群磁盘。 由于许多进程可能正在尝试同时写入此磁盘，因此可能会阻止某些 IBM MQ 进程运行。
      - `Parameters` ; 如下所示:

- QueueManagerName (必需); 此资源要控制的队列管理器的名称。 此队列管理器必须存在于本地计算机上。
- PostOnlineCommand (可选); 您可以指定每当队列管理器资源将其状态从脱机更改为联机时要运行的程序。 有关更多详细信息，请参阅 第 414 页的『MSCS 中的 PostOnline 命令和 PreOffline 命令』。
- PreOfflineCommand (可选); 您可以指定每当队列管理器资源将其状态从联机更改为脱机时要运行的程序。 有关更多详细信息，请参阅 第 414 页的『MSCS 中的 PostOnline 命令和 PreOffline 命令』。

  注: *looksAlive* 轮询时间间隔设置为缺省值 5000 毫秒。 *isAlive* 轮询时间间隔设置为缺省值 60000 毫秒。 这些缺省值只能在资源定义完成后修改。 有关更多详细信息，请参阅 第 411 页的『MSCS 上的 looksAlive 和 isAlive 轮询』。

d. (可选) 设置首选节点 (但请注意 第 414 页的『在 MSCS 中使用首选节点』 中的注释)

e. 缺省情况下， 故障转移策略 设置为合理值，但您可以调整用于控制 资源故障转移 和 组故障转移 的阈值和时间段，以匹配队列管理器上的负载。

6. 通过使队列管理器在 MSCS 集群管理员中联机并使其承受测试工作负载来测试该队列管理器。 如果您正在试验测试队列管理器，请使用 IBM MQ Explorer。 例如:

   a. 右键单击 **队列** 树节点，然后选择 **新建 > 本地队列 ...**。 并为队列提供名称。

   b. 单击**完成**。 将创建队列，并显示在内容视图中。

   c. 右键单击队列，然后选择 **放置测试消息 ...**。 此时将显示 "放置测试消息" 面板。

   d. 输入一些消息文本，然后单击 **放置测试消息**，然后关闭面板。

   e. 右键单击队列，然后选择 **浏览消息 ...**。 将显示 "消息浏览器" 面板。

   f. 确保消息位于队列上，然后单击 **关闭**。 "消息浏览器" 面板将关闭。

   g. 右键单击队列，然后选择 **清除消息 ...**。 将清除队列上的消息。

   h. 右键单击队列，然后选择 **删除 ...**。 将显示确认面板，单击 **确定**。 此时会删除该队列。

7. 测试是否可以使用 MSCS 集群管理员使队列管理器脱机并恢复联机状态。

8. 模拟故障转移。

   在 MSCS 集群管理员中，右键单击包含队列管理器的组，然后选择 `Move Group`。 这可能需要一些时间。 (如果要在其他时间将队列管理器快速移动到其他节点，请遵循 第 405 页的『将队列管理器移至 MSCS 存储器』中的过程。) 您还可以右键单击并选择 `Initiate Failure`; 操作 (本地重新启动或故障转移) 取决于当前状态和配置设置。

Windows Server 2008

要将队列管理器置于 Windows Server 2008 上的 MSCS 控制下，请使用以下过程:

1. 登录到托管队列管理器的集群节点计算机，或者以具有集群管理许可权的用户身份登录到远程工作站，并连接到托管队列管理器的集群节点。
2. 启动 "故障转移集群管理工具"。
3. 右键单击 **故障转移集群管理> 管理集群 ...** 以打开与集群的连接。
4. 与 MSCS 集群管理员在先前版本的 Windows 上使用的组方案不同，故障转移集群管理工具使用服务和应用程序的概念。 已配置的服务或应用程序包含要进行集群的一个应用程序所需的所有资源。 您可以在 MSCS 下配置队列管理器，如下所示:

   a. 右键单击 **服务和应用程序> 配置服务或应用程序 ...** 以启动配置向导。

   b. 在 " **选择服务或应用程序** " 面板上选择 **其他服务器** 。

   c. 选择相应的 IP 地址作为客户机访问点。

      此地址应该是客户机和其他队列管理器用于连接到 虚拟 队列管理器的未使用 IP 地址。 此 IP 地址不是任一节点的正常 (静态) 地址; 它是它们之间 浮动 的额外地址。 虽然 MSCS 处理此地址的路由，但它 **不会** 验证是否可以访问该地址。

   d. 分配存储设备以供队列管理器独占使用。 需要先创建此设备作为资源实例，然后才能对其进行分配。

您可以使用一个驱动器来存储日志和队列文件，也可以将它们拆分为多个驱动器。 在任何一种情况下，如果每个队列管理器都有自己的共享磁盘，请确保此队列管理器所使用的所有驱动器都是此队列管理器所独有的，即，没有任何其他驱动器依赖于这些驱动器。 还要确保为队列管理器使用的每个驱动器创建资源实例。

磁带机的资源类型取决于您正在使用的 SCSI 支持; 请参阅 SCSI 适配器指示信息。 可能已存在每个共享驱动器的组和资源。 如果是这样，那么不需要为每个驱动器创建资源实例。 将其从当前组移至为队列管理器创建的组。

对于每个驱动器资源，请将可能的所有者设置为两个节点。 将从属资源设置为 "无"。

e. 在 " **选择资源类型** " 面板上选择 **MQSeries MSCS** 资源。

f. 完成向导中的其余步骤。

5. 在使资源联机之前， MQSeries MSCS 资源需要其他配置:

a. 选择新定义的服务，该服务包含名为 "New MQSeries MSCS" 的资源。

b. 右键单击 MQ 资源上的 **属性** 。

c. 配置资源:

- `Name` ; 选择一个便于识别它所针对的队列管理器的名称。
- `Run in a separate Resource Monitor` ; 实现更好的隔离
- `Possible owners` ; 设置两个节点
- `Dependencies` ; 添加此队列管理器的驱动器和 IP 地址。

  警告: 未能添加这些依赖关系意味着 IBM MQ 尝试在故障转移期间将队列管理器状态写入错误的集群磁盘。 由于许多进程可能正在尝试同时写入此磁盘，因此可能会阻止某些 IBM MQ 进程运行。
- `Parameters` ; 如下所示:
  - `QueueManagerName` (必需); 此资源要控制的队列管理器的名称。 此队列管理器必须存在于本地计算机上。
  - `PostOnlineCommand` (可选); 您可以指定每当队列管理器资源将其状态从脱机更改为联机时要运行的程序。 有关更多详细信息，请参阅 第 414 页的『MSCS 中的 PostOnline 命令和 PreOffline 命令』。
  - `PreOfflineCommand` (可选); 您可以指定每当队列管理器资源将其状态从联机更改为脱机时要运行的程序。 有关更多详细信息，请参阅 第 414 页的『MSCS 中的 PostOnline 命令和 PreOffline 命令』。

    注: *looksAlive* 轮询时间间隔设置为缺省值 5000 毫秒。 *isAlive* 轮询时间间隔设置为缺省值 60000 毫秒。 这些缺省值只能在资源定义完成后修改。 有关更多详细信息，请参阅 第 411 页的『MSCS 上的 looksAlive 和 isAlive 轮询』。

d. (可选) 设置首选节点 (但请注意 第 414 页的『在 MSCS 中使用首选节点』 中的注释)

e. 缺省情况下， 故障转移策略 设置为合理值，但您可以调整用于控制 资源故障转移 和 组故障转移 的阈值和时间段，以匹配队列管理器上的负载。

6. 通过使队列管理器在 MSCS 集群管理员中联机并使其承受测试工作负载来测试该队列管理器。 如果您正在试验测试队列管理器，请使用 IBM MQ Explorer。 例如:

a. 右键单击 **队列** 树节点，然后选择 **新建 > 本地队列 ...**。 并为队列提供名称。

b. 单击**完成**。 将创建队列，并显示在内容视图中。

c. 右键单击队列，然后选择 **放置测试消息 ...**。 这样会显示 " **放置测试消息** " 面板。

d. 输入一些消息文本，然后单击 **放置测试消息**，然后关闭面板。

e. 右键单击队列，然后选择 **浏览消息 ...**。 此时将显示 " **消息浏览器** " 面板。

f. 确保消息位于队列上，然后单击 **关闭**。 " **消息浏览器** " 面板将关闭。

g. 右键单击队列，然后选择 **清除消息 ...**。 将清除队列上的消息。

h. 右键单击队列，然后选择 **删除 ...**。 将显示确认面板，单击 **确定**。 此时会删除该队列。

7. 测试是否可以使用 MSCS 集群管理员使队列管理器脱机并恢复联机状态。

8. 模拟故障转移。

   在 MSCS 集群管理员中，右键单击包含队列管理器的组，然后选择 `Move Group`。 这可能需要一些时间。 (如果要在其他时间将队列管理器快速移动到其他节点，请遵循 第 405 页的『将队列管理器移至 MSCS 存储器』中的过程。) 您还可以右键单击并选择 `Initiate Failure`；操作 (本地重新启动或故障转移) 取决于当前状态和配置设置。

Windows 2003

要将队列管理器置于 Windows 2003 上的 MSCS 控制下，请使用以下过程：

1. 登录到托管队列管理器的集群节点计算机，或者以具有集群管理许可权的用户身份登录到远程工作站，并连接到托管队列管理器的集群节点。
2. 启动 MSCS 集群管理员。
3. 打开与集群的连接。
4. 创建要用于包含队列管理器资源的 MSCS 组。 以明显与该组相关的队列管理器的方式命名该组。 每个组可以包含多个队列管理器，如 第 404 页的『将多个队列管理器与 MSCS 配合使用』中所述。

   将该组用于所有其余步骤。
5. 为队列管理器使用的每个 SCSI 逻辑驱动器创建一个资源实例。

   您可以使用一个驱动器来存储日志和队列文件，也可以将它们拆分为多个驱动器。 在任何一种情况下，如果每个队列管理器都有自己的共享磁盘，请确保此队列管理器所使用的所有驱动器都是此队列管理器所独有的，即，没有任何其他驱动器依赖于这些驱动器。 还要确保为队列管理器使用的每个驱动器创建资源实例。

   磁带机的资源类型取决于您正在使用的 SCSI 支持; 请参阅 SCSI 适配器指示信息。 可能已存在每个共享驱动器的组和资源。 如果是这样，那么不需要为每个驱动器创建资源实例。 将其从当前组移至为队列管理器创建的组。

   对于每个驱动器资源，请将可能的所有者设置为两个节点。 将从属资源设置为 "无"。
6. 为 IP 地址创建资源实例。

   创建 IP 地址资源 (资源类型 *IP address*)。 此地址应该是客户机和其他队列管理器用于连接到 虚拟 队列管理器的未使用 IP 地址。 此 IP 地址不是任一节点的正常 (静态) 地址; 它是它们之间 浮动 的额外地址。 虽然 MSCS 处理此地址的路由，但它 **不会** 验证是否可以访问该地址。
7. 为队列管理器创建资源实例。

   创建类型为 *IBM MQ MSCS* 的资源。 向导会提示您输入各种项，包括以下项：

   - `Name`；选择一个便于识别它所针对的队列管理器的名称。
   - `Add to group`；使用您创建的组
   - `Run in a separate Resource Monitor`；实现更好的隔离
   - `Possible owners`；设置两个节点
   - `Dependencies`；添加此队列管理器的驱动器和 IP 地址。

     警告: 未能添加这些依赖关系意味着 IBM MQ 尝试在故障转移期间将队列管理器状态写入错误的集群磁盘。 由于许多进程可能正在尝试同时写入此磁盘，因此可能会阻止某些 IBM MQ 进程运行。
   - `Parameters`；如下所示：
     - `QueueManagerName` (必需); 此资源要控制的队列管理器的名称。 此队列管理器必须存在于本地计算机上。
     - `PostOnlineCommand` (可选); 您可以指定每当队列管理器资源将其状态从脱机更改为联机时要运行的程序。 有关更多详细信息，请参阅 第 414 页的『MSCS 中的 PostOnline 命令和 PreOffline 命令』。
     - `PreOfflineCommand` (可选); 您可以指定每当队列管理器资源将其状态从联机更改为脱机时要运行的程序。 有关更多详细信息，请参阅 第 414 页的『MSCS 中的 PostOnline 命令和 PreOffline 命令』。

注: *looksAlive* 轮询时间间隔设置为缺省值 5000 毫秒。 *isAlive* 轮询时间间隔设置为缺省值 30000 毫秒。 这些缺省值只能在资源定义完成后修改。 有关更多详细信息，请参阅 第 411 页的『MSCS 上的 looksAlive 和 isAlive 轮询』。

8. (可选) 设置首选节点 (但请注意 第 414 页的『在 MSCS 中使用首选节点』 中的注释)
9. 缺省情况下， 故障转移策略 (在组的属性中定义) 设置为合理值，但您可以调整用于控制 资源故障转移 和 组故障转移 的阈值和周期，以与队列管理器上的负载相匹配。
10. 通过使队列管理器在 MSCS 集群管理员中联机并使其承受测试工作负载来测试该队列管理器。 如果您正在试验测试队列管理器，请使用 IBM MQ Explorer。 例如:
    a. 右键单击 **队列** 树节点，然后选择 **新建 > 本地队列 ...**。 并为队列提供名称。
    b. 单击**完成**。 将创建队列，并显示在内容视图中。
    c. 右键单击队列，然后选择 **放置测试消息 ...**。 这样会显示 " **放置测试消息** " 面板。
    d. 输入一些消息文本，然后单击 **放置测试消息**，然后关闭面板。
    e. 右键单击队列，然后选择 **浏览消息 ...**。 此时将显示 " **消息浏览器** " 面板。
    f. 确保消息位于队列上，然后单击 **关闭**。 " **消息浏览器** " 面板将关闭。
    g. 右键单击队列，然后选择 **清除消息 ...**。 将清除队列上的消息。
    h. 右键单击队列，然后选择 **删除 ...**。 将显示确认面板，单击 **确定**。 此时会删除该队列。
11. 测试是否可以使用 MSCS 集群管理员使队列管理器脱机并恢复联机状态。
12. 模拟故障转移。

    在 MSCS 集群管理员中，右键单击包含队列管理器的组，然后选择 `Move Group`。 这可能需要一些时间。 (如果要在其他时间将队列管理器快速移动到其他节点，请遵循 第 405 页的『将队列管理器移至 MSCS 存储器』中的过程。) 您还可以右键单击并选择 `Initiate Failure`; 操作 (本地重新启动或故障转移) 取决于当前状态和配置设置。

Windows *MSCS* 上的 *looksAlive* 和 *isAlive* 轮询

looksAlive 和 *isAlive* 是 Microsoft Cluster Service (MSCS) 重新调用资源类型提供的库代码并请求资源执行检查以确定其工作状态的时间间隔。 这最终确定 MSCS 是否尝试对资源进行故障转移。

注: 从 Windows Server 2016 开始， Microsoft Cluster Service (MSCS) 的新名称为 Windows Server Failover Clustering (WSFC)。

在每次 *looksAlive* 时间间隔过去 (缺省值为 5000 毫秒) 时，都会调用队列管理器资源来执行其自己的检查，以确定其状态是否令人满意。

在每次 *isAlive* 时间间隔过去 (缺省值为 30000 毫秒) 时，都会对队列管理器资源进行另一次调用，以便它执行另一次检查以确定该资源是否正常运行。 这将启用两个级别的资源类型检查。

1. *looksAlive* 状态检查，用于确定资源是否显示正常运行。
2. 更重要的 *isAlive* 检查可确定队列管理器资源是否处于活动状态。

如果确定队列管理器资源不处于活动状态，那么 MSCS 将基于其他高级 MSCS 选项，针对该资源以及关联的从属资源触发故障转移到集群中的另一个节点。 有关更多信息，请参阅 MSCS 文档。

Windows 从 *MSCS* 控件中除去队列管理器

您可以从 Microsoft Cluster Service (MSCS) 控制中除去队列管理器，并将其返回到手动管理。

注: 从 Windows Server 2016 开始， Microsoft Cluster Service (MSCS) 的新名称为 Windows Server Failover Clustering (WSFC)。

您不需要从 MSCS 控制中除去队列管理器以进行维护操作。 您可以通过使用 MSCS 集群管理员使队列管理器暂时脱机来执行此操作。 从 MSCS 控制中除去队列管理器是更持久的更改; 仅当您决定不再希望 MSCS 进一步控制队列管理器时才执行此操作。

如果正在除去的队列管理器使用 TSL 连接，那么必须使用 IBM MQ Explorer 或 MQSC 命令 `ALTER QMGR` 修改队列管理器属性 SSLKEYR ，以指向本地目录上的 TLS 密钥存储库文件。

该过程为:

1. 使用 MSCS 集群管理员使队列管理器资源脱机，如 第 412 页的『使队列管理器从 MSCS 脱机』 中所述
2. 销毁资源实例。 这不会破坏队列管理器。
3. (可选) 将队列管理器文件从共享驱动器迁移回本地驱动器。 要执行此操作，请参阅第 412 页的『从 MSCS 存储器返回队列管理器』。
4. 测试队列管理器。

使队列管理器从 MSCS 脱机

要使队列管理器从 MSCS 脱机，请执行以下步骤:

1. 启动 MSCS 集群管理员。
2. 打开与集群的连接。
3. 如果您正在使用 Windows 2012 ，请选择 `Groups` 或 `Role` ，然后打开包含要移动的队列管理器的组。
4. 选择队列管理器资源。
5. 右键单击并选择 `Offline`。
6. 等待完成。

从 MSCS 存储器返回队列管理器

此过程将队列管理器配置为回到其计算机的本地驱动器上，即，它成为 正常 IBM MQ 队列管理器。 要实现此目的，请从共享磁盘中移动日志文件和数据文件。 例如，现有队列管理器可能具有 `E:\WebSphere MQ\log\`*QMname* 和 `E:\WebSphere MQ\qmgrs\`*QMname* 之类的路径。 请*勿*尝试手动移动文件; 请使用作为 IBM MQ MSCS 支持的一部分提供的 **hamvmqm** 实用程序:

1. 对队列文件和日志文件进行完全介质备份，并将备份存储在安全的位置 (请参阅 第 413 页的『队列管理器日志文件』 以了解这很重要的原因)。
2. 确定要使用的本地驱动器，并确保其具有足够的容量来存储队列管理器日志文件和数据 (队列) 文件。
3. 确保当前文件所在的共享磁盘在要将队列管理器日志和数据文件移动到的集群节点上处于联机状态。
4. 运行实用程序以移动队列管理器，如下所示:

   ```
   hamvmqm /m qmname /dd " c:\
   IBM MQ " /ld "c:\
   IBM MQ \log"
   ```

   将队列管理器名称替换为 *qmname*，将本地磁盘盘符替换为 *c*，并将所选目录替换为 *IBM MQ* (如果这些目录尚不存在，那么将创建这些目录)。
5. 测试队列管理器以确保其工作 (如 第 405 页的『将队列管理器移至 MSCS 存储器』 中所述)。

Windows 有关使用 *MSCS* 的提示和技巧

本部分包含一些常规信息，可帮助您有效使用 IBM MQ 对 Microsoft Cluster Service (MSCS) 的支持。

注: 从 Windows Server 2016 开始， Microsoft Cluster Service (MSCS) 的新名称为 Windows Server Failover Clustering (WSFC)。

将队列管理器从一台机器故障转移到另一台机器需要多长时间? 这在很大程度上取决于队列管理器上的工作负载量以及流量组合，例如，在同步点内持久的流量，以及在失败之前落实的流量。 IBM 测试提供了大约 1 分钟的故障转移和故障恢复时间。 这在负载非常轻的队列管理器上，实际时间会因负载而有很大差异。

Windows 验证 *MSCS* 是否正常工作

请执行以下步骤以确保您具有正在运行的 MSCS 集群。

以 第 405 页的『创建队列管理器以用于 MSCS』 开头的任务描述假定您有一个正在运行的 MSCS 集群，您可以在该集群中创建，迁移和销毁资源。 如果要确保您具有这样的集群:

1. 使用 MSCS 集群管理员创建组。
2. 在该组中，创建通用应用程序资源的实例，并指定系统时钟 (路径名 `C:\winnt\system32\clock.exe` 和工作目录 `C:\`)。

3. 确保可以使资源联机，可以将包含该资源的组移动到其他节点，并且可以使该资源脱机。

Windows 手动启动和 *MSCS*
对于由 MSCS 管理的队列管理器，必须将启动属性设置为手动。 这将确保 IBM MQ MSCS 支持可以在不立即启动队列管理器的情况下重新启动 MQSeries 服务。

IBM MQ MSCS 支持需要能够重新启动服务，以便它可以执行监视和控制，但自身必须保持控制哪些队列管理器正在运行以及哪些机器上运行。 有关更多信息，请参阅 第 405 页的『将队列管理器移至 MSCS 存储器』。

Windows *MSCS* 和队列管理器
使用 MSCS 时有关队列管理器的注意事项。

在另一个节点上创建匹配的队列管理器

要使集群与 IBM MQ 配合工作，节点 A 上的每个集群都需要节点 B 上的相同队列管理器。 但是，您不需要显式地创建第二个。 您可以在一个节点上创建或准备队列管理器，将其移至另一个节点 (如 第 405 页的『将队列管理器移至 MSCS 存储器』中所述)，并在该节点上完全复制该队列管理器。

缺省队列管理器

请勿使用 MSCS 控制下的缺省队列管理器。 队列管理器没有使其成为缺省值的属性; IBM MQ 会保留其自己的单独记录。 如果将队列管理器设置为故障转移时的缺省值，那么它不会成为其中的缺省值。 使所有应用程序按名称引用特定队列管理器。

删除队列管理器

一旦队列管理器移动了节点，它的详细信息就存在于这两台计算机上的注册表中。 要将其删除时，请在一台计算机上正常执行此操作，然后运行 第 415 页的『支持 MSCS 实用程序』中描述的实用程序以清除另一台计算机上的注册表。

对现有队列管理器的支持

您可以将现有队列管理器置于 MSCS 控制之下，前提是可以将队列管理器日志文件和队列文件放在两台机器之间的共享 SCSI 总线上的磁盘上 (请参阅 第 403 页的图 74)。 创建 MSCS 资源时，需要使队列管理器短暂脱机。

如果要创建新的队列管理器，请独立于 MSCS 创建该队列管理器，对其进行测试，然后将其置于 MSCS 控制之下。 请参阅:

- 第 405 页的『创建队列管理器以用于 MSCS』
- 第 405 页的『将队列管理器移至 MSCS 存储器』
- 第 406 页的『将队列管理器置于 MSCS 控制之下』

告诉 MSCS 要管理哪些队列管理器

通过使用 MSCS 集群管理员为每个此类队列管理器创建资源实例，可以选择将哪些队列管理器置于 MSCS 控制之下。 此过程向您提供资源列表，您可以从这些资源中选择希望该实例管理的队列管理器。

队列管理器日志文件

将队列管理器移至 MSCS 存储器时，会将其日志和数据文件移至共享磁盘 (有关示例，请参阅 第 405 页的『将队列管理器移至 MSCS 存储器』)。

在移动之前，建议先完全关闭队列管理器，并对数据文件和日志文件进行完全备份。

多个队列管理器

IBM MQ MSCS 支持允许您在每台机器上运行多个队列管理器，并将各个队列管理器置于 MSCS 控制之下。

Windows 始终使用 *MSCS* 来管理集群

请勿尝试使用控制命令或 IBM MQ Explorer 直接在 MSCS 控制下的任何队列管理器上执行启动和停止操作。 请改为使用 MSCS 集群管理员使队列管理器联机或脱机。

使用 MSCS 集群管理员部分是为了防止 MSCS 报告队列管理器处于脱机状态 (事实上您已在 MSCS 的控制范围之外启动该队列管理器) 可能引起的混淆。 更严重的是， MSCS 检测到在不使用 MSCS 的情况下停止队列管理器为故障，从而启动到其他节点的故障转移。

Windows 在 *MSCS* 中以主动/主动方式工作

MSCS 集群中的两台计算机都可以在主动/主动方式下运行队列管理器。 您不需要让完全空闲的机器充当备用机器 (但如果需要，可以采用主动/被动方式)。

如果计划使用两台机器来运行工作负载，请为每台机器提供足够的容量 (处理器，内存和辅助存储器) ，以便以令人满意的性能级别运行整个集群工作负载。

注: 如果要将 MSCS 与 Microsoft Transaction Server (COM +) 一起使用，那么 **不能** 使用主动/主动方式。 这是因为，要将 IBM MQ 与 MSCS 和 COM + 配合使用:

- 使用 IBM MQ COM + 支持的应用程序组件必须与分布式事务协调程序 (DTC) (COM + 的一部分) 在同一台计算机上运行。
- 队列管理器也必须在同一计算机上运行。
- DTC 必须配置为 MSCS 资源，因此随时只能在集群中的一台计算机上运行。

Windows *MSCS* 中的 *PostOnline* 命令和 *PreOffline* 命令

使用这些命令可将 IBM MQ MSCS 支持与其他系统集成。 您可以使用它们来发出 IBM MQ 命令，但有一些限制。

在 "参数" 中对类型为 `IBM MQ MSCS` 的资源指定这些命令。 您可以使用它们将 IBM MQ MSCS 支持与其他系统或过程集成。 例如，可以指定用于发送邮件消息，激活寻呼机或生成其他形式的警报以供其他监视系统捕获的程序的名称。

当资源从脱机更改为联机时，将调用 PostOnline 命令; 对于从联机更改为脱机的更改，将调用 PreOffline 命令。 在调用时，缺省情况下将从 Windows 系统目录运行这些命令。 因为 IBM MQ 使用 32 位资源监视器进程，所以在 Windows 64 位系统上，这是 `\Windows\SysWOW64` 目录而不是 `\Windows\system32` 目录。 有关更多信息，请参阅有关 Windows x64 环境中文件重定向的 Microsoft 文档。 这两个命令都在用于运行 MSCS 集群服务的用户帐户下运行; 并且以异步方式调用; IBM MQ MSCS 支持不会在继续之前等待这些命令完成。 这将消除它们可能阻止或延迟进一步集群操作的任何风险。

您还可以使用这些命令来发出 IBM MQ 命令，例如，重新启动请求者通道。 但是，这些命令是在队列管理器的状态更改时运行的，因此它们不打算执行长时间运行的功能，并且不得对队列管理器的当前状态进行假定; 很有可能，在队列管理器联机后，管理员发出了脱机命令。

如果要运行依赖于队列管理器状态的程序，请考虑创建 `MSCS Generic Application` 资源类型的实例，将它们与队列管理器资源放在同一 MSCS 组中，并使它们依赖于队列管理器资源。

Windows 在 *MSCS* 中使用首选节点

在 MSCS 中使用 "主动/主动" 方式为每个队列管理器配置 首选节点 时，此方法很有用。 但是，通常最好不要设置首选节点，而是依靠手动故障恢复。

与其他一些相对无状态的资源不同，队列管理器可能需要一段时间才能从一个节点故障转移 (或回退) 到另一个节点。 为避免不必要的中断，请先测试恢复的节点，然后再将队列管理器故障恢复到该节点。 这将阻止使用 `immediate` 故障恢复设置。 您可以将故障恢复配置为在一天中的特定时间之间发生。

可能最安全的路由是将队列管理器手动移回所需节点，当您确定该节点已完全恢复时。 这将阻止使用 `preferred node` 选项。

Windows 在 *MSCS* 上安装时发生 *COM +* 错误

在新安装的 MSCS 集群上安装 IBM MQ 时，可能会发现 "应用程序事件" 日志中报告了源 COM + 和事件标识 4691 的错误。

这意味着当未将 Microsoft 分布式事务协调程序 (MSDTC) 配置为在 Microsoft 集群服务器 (MSCS) 环境中运行时，您将尝试在该环境中运行 IBM MQ 。 有关在集群环境中配置 MSDTC 的信息，请参阅 Microsoft 文档。

Windows 支持 *MSCS* 实用程序

可以在命令提示符处运行的 IBM MQ 支持 Microsoft Cluster Service (MSCS) 实用程序的列表。

注: 从 Windows Server 2016 开始， Microsoft Cluster Service (MSCS) 的新名称为 Windows Server Failover Clustering (WSFC)。

对 MSCS 的 IBM MQ 支持包含以下实用程序:

注册/注销资源类型

`haregtyp.exe`

在 注销 IBM MQ MSCS 资源类型之后，您无法再创建该类型的任何资源。 如果在集群中仍有该类型的实例，那么 MSCS 不会允许您注销资源类型:

1. 通过使用 MSCS 集群管理员，停止在 MSCS 控制下运行的任何队列管理器，方法是使它们脱机，如第 412 页的『使队列管理器从 MSCS 脱机』中所述。
2. 使用 MSCS 集群管理员，删除资源实例。
3. 在命令提示符处，通过输入以下命令来注销资源类型:

```
haregtyp /u
```

如果要 注册 类型 (或稍后重新注册) ，请在命令提示符处输入以下命令:

```
haregtyp /r
```

成功注册 MSCS 库后，如果自安装 IBM MQ 以来未重新引导系统，那么必须重新引导系统。

将队列管理器移至 MSCS 存储器

`hamvmqm.exe`

请参阅 第 405 页的『将队列管理器移至 MSCS 存储器』。

从节点中删除队列管理器

`hadltmqm.exe`

请考虑以下情况: 您在集群中有一个队列管理器，该队列管理器已从一个节点移动到另一个节点，现在要将其销毁。 使用 IBM MQ Explorer 在其当前所在的节点上将其删除。 它的注册表项仍存在于另一台计算机上。 要删除这些内容，请在该计算机上的提示符处输入以下命令:

```
hadltmqm /m qmname
```

其中 qmname 是要除去的队列管理器的名称。

检查并保存设置详细信息

`amqmsysn.exe`

此实用程序将显示一个对话框，其中显示 IBM MQ MSCS 支持设置的完整详细信息，例如，如果您调用 IBM 支持，那么可能会请求此设置。 可以选择将详细信息保存到文件中。

Multi 多实例队列管理器

多实例队列管理器是在不同服务器上配置的同一队列管理器的多个实例。 其中一个队列管理器实例定义为活动实例，另一个实例定义为备用实例。 如果活动实例发生故障，那么备用服务器上的多实例队列管理器将自动重新启动。

多实例队列管理器配置示例

第 416 页的图 75 显示了队列管理器 QM1 的多实例配置示例。 IBM MQ 安装在两个服务器上，其中一个服务器是备用服务器。 已创建一个队列管理器，即 QM1。 QM1 的一个实例处于活动状态，并且正在一个服务器上运行。 QM1 的另一个实例以备用方式在另一个服务器上运行，它不执行任何活动处理，但已准备好在

QM1 的活动实例发生故障时接管活动实例。 (在多实例配置中，队列管理器只能有一个活动实例和一个备用实例。)

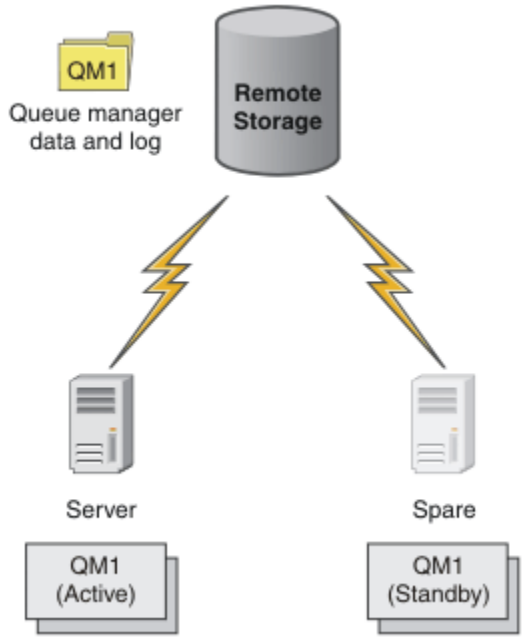


图 *75:* 多实例队列管理器

当您打算将队列管理器用作多实例队列管理器时，请使用 **crtmqm** 命令在其中一个服务器上创建单个队列管理器，将其队列管理器数据和日志放在共享网络存储器中。 在另一服务器上，而不是再次创建队列管理器，请使用 **addmqinf** 命令来创建对队列管理器数据的引用，并在网络存储器上进行日志记录。

现在，可以在任何一个服务器上运行队列管理器。 每个服务器都引用相同的队列管理器数据和日志；只有一个队列管理器，并且此队列管理器一次仅在一个服务器上处于活动状态。

队列管理器可以作为单一实例队列管理器或多实例队列管理器运行。 在这两种情况下，都只有一个队列管理器实例处于运行状态并处理请求。 区别在于，作为多实例队列管理器运行时，未运行活动队列管理器实例的服务器将作为备用实例运行并准备好在活动服务器发生故障时自动接管活动实例。

您在两个服务器上启动队列管理器的顺序是您对哪个实例先变成活动状态可以进行的唯一控制。 第一个对队列管理器数据获取读/写锁定的实例将成为活动实例。

在另一服务器启动后，您可以使用切换选项将控制权转移到备用实例以停止活动实例，从而将活动实例切换到该服务器。

QM1 的活动实例在处于运行状态时对共享队列管理器数据和日志文件夹具有独占访问权。 在活动实例出现故障时，QM1 的备用实例将检测到这一情况，并变成活动实例。 它将以先前活动实例所遗留的状态来接管 QM1 数据和日志，并接受来自客户机和通道的重新连接请求。

活动实例可能会由于各种原因而发生故障，这将导致备用实例执行接管：

- 托管活动队列管理器实例的服务器发生故障。
- 主管活动队列管理器实例的服务器与主管文件系统的服务器之间的连接发生故障。
- IBM MQ 检测到队列管理器进程无响应，并随后关闭队列管理器。

您可以将队列管理器配置信息添加到多个服务器，并选择任意两个服务器作为活动/备用服务器对运行。 共有两个实例的限制。 不能有两个备用实例和一个活动实例。

构建高可用性解决方案所需的其他组件

多实例队列管理器是高可用性解决方案的一个组成部分。 您需要其他组件来构建有用的高可用性解决方案。

- 客户机和通道重新连接，以使 IBM MQ 连接到将接管运行活动队列管理器实例的计算机。

- 高性能共享网络文件系统 (NFS) ，可正确管理锁定并提供针对介质和文件服务器故障的保护。

  要点: 必须先停止环境中运行的所有多实例队列管理器实例，然后才能在 NFS 驱动器上执行维护。 确保在发生 NFS 故障时有要恢复的队列管理器配置备份。
- 用于杜绝基本基础结构中发生单点故障的弹性网络和电源。
- 支持故障转移的应用程序。 特别是，您需要注意事务性应用程序和浏览 IBM MQ 队列的应用程序的行为。
- 监视并管理活动和备用实例，以确保这些实例处于运行状态，并在活动实例发生故障时将其重新启动。 虽然多实例队列管理器自动重新启动，但您需要确保备用实例处于运行状态、随时可以接管并且发生故障的实例已作为新的备用实例恢复联机状态。

当备用队列管理器处于活动状态时， IBM MQ MQI clients 和通道会自动重新连接到备用队列管理器。 可以在相关主题中找到有关重新连接以及高可用性解决方案中的其他组件的更多信息。 IBM MQ classes for Java 不支持客户机自动重新连接。

受支持的平台

您可以在 IBM WebSphere MQ 7.0.1 和更高版本支持的任何非 z/OS 平台上创建多实例队列管理器。

IBM WebSphere MQ 7.0.1 和更高版本支持 MQI 客户机自动重新连接。

创建多实例队列管理器

创建多实例队列管理器，在一个服务器上创建队列管理器，并在另一个服务器上配置 IBM MQ 。 多实例队列管理器共享队列管理器数据和日志。

创建多实例队列管理器所涉及的大部分工作是设置共享队列管理器数据和日志文件。 您必须在网络存储器上创建共享目录，并使用网络共享使这些目录可供其他服务器使用。 这些任务需要由具有管理权限的人员 (例如 AIX and Linux 系统上的 *root*) 执行。 步骤如下所示:

1. 为数据和日志文件创建共享。
2. 在一台服务器上创建队列管理器。
3. 在第一个服务器上运行命令 **dspmqinf** 以收集队列管理器配置数据并将其复制到剪贴板。
4. 使用复制的数据运行命令 **addmqinf** 以在第二个服务器上创建队列管理器配置。

您不会运行 **crtmqm** 以在第二个服务器上再次创建队列管理器。

文件访问控制

所有其他服务器上的用户和组 `mqm` 都有权访问共享，您需要注意到这一点。

在 AIX and Linux 上，您需要在所有系统上使 `mqm` 的 `uid` 和 `gid` 相同。 您可能需要在每个系统上编辑 `/etc/passwd` 以针对 `mqm` 设置公共 `uid` 和 `gid` ，然后重新引导系统。

在 Microsoft Windows 上，运行队列管理器进程的用户标识必须对包含队列管理器数据和日志文件的目录具有完全控制许可权。 您可以通过两种方式配置许可权:

1. 创建具有全局组作为备用安全主体的队列管理器。 授权全局组对包含队列管理器数据和日志文件的目录具有完全控制访问权; 请参阅 第 441 页的『保护 Windows 上的共享队列管理器数据和日志目录及文件』。 使正在运行队列管理器的用户标识成为全局组的成员。 不能使本地用户成为全局组的成员，因此队列管理器进程必须以域用户标识运行。 域用户标识必须是本地组 `mqm` 的成员。 任务 第 419 页的『在 Windows 上的域工作站或服务器上创建多实例队列管理器』演示了如何使用以此方式保护的文件来设置多实例队列管理器。
2. 在域控制器上创建队列管理器，以便本地 `mqm` 组具有域作用域 “domain local”。 保护与域本地 `mqm` 的文件共享，并在同一域本地 `mqm` 组下的队列管理器的所有实例上运行队列管理器进程。 任务 第 432 页的『在 Windows 域控制器上创建多实例队列管理器』演示了如何使用以此方式保护的文件来设置多实例队列管理器。

配置信息

通过修改有关每个服务器的 IBM MQ 队列管理器配置信息，根据需要配置任意数量的队列管理器实例。 每个服务器都必须在兼容修订级别安装相同版本的 IBM MQ 。 **dspmqinf** 和 **addmqinf** 命令可帮助您配置其他

队列管理器实例。 或者，您可以直接编辑 mqs.ini 和 qm.ini 文件。 主题 第 452 页的『在 Linux 上创建多实例队列管理器』， 第 419 页的『在 Windows 上的域工作站或服务器上创建多实例队列管理器』和 第 432 页的『在 Windows 域控制器上创建多实例队列管理器』 是显示如何配置多实例队列管理器的示例。

在 AIX, Linux, and Windows 系统上，您可以通过将单个 mqs.ini 文件放在网络共享上并将 **AMQ_MQS_INI_LOCATION** 环境变量设置为指向该文件来共享该文件。

限制

1. 仅在具有相同操作系统，体系结构和环境的服务器上配置同一队列管理器的多个实例。 例如，两台机器都必须是 32 位或 64 位。
2. 所有 IBM MQ 安装必须处于发行版级别 7.0.1 或更高版本。
3. 通常，活动安装和备用安装保持在相同的维护级别。 请参阅每次升级的维护指示信息，以检查是否必须同时升级所有安装。

   请注意，主动和被动队列管理器的维护级别必须相同。
4. 仅在使用同一 IBM MQ 用户，组和访问控制机制配置的队列管理器之间共享队列管理器数据和日志。

   IBM i 例如，在 Linux 服务器上设置的网络共享可以包含单独的队列管理器数据和 AIX and Linux 队列管理器的日志，但不能包含 IBM i 使用的队列管理器数据。

   IBM i 只要共享不同，您就可以在 IBM i 和 AIX and Linux 系统的同一联网存储器上创建多个共享。 您可以为不同的共享提供不同的所有者。 此限制是用于 AIX and Linux 和 IBM i 之间的 IBM MQ 用户和组的不同名称的结果。 用户和组可以具有相同的 uid 和 gid 这一事实不会放宽限制。
5. 在 AIX and Linux 系统上，使用可中断的**硬**安装而不是**软**安装在联网存储器上配置共享文件系统。 在系统调用中断队列管理器之前，可中断的硬安装将强制该队列管理器挂起。 软安装不保证服务器发生故障后的数据一致性。
6. 共享日志和数据目录不能存储在 FAT 或 NFSv3 文件系统上。 对于 Windows 上的多实例队列管理器，网络存储器必须由 Windows 网络使用的公共因特网文件系统 (CIFS) 协议访问。
7. z/OS z/OS 不支持多实例队列管理器。 使用队列共享组。

   可重新连接的客户机使用 z/OS 队列管理器。

Windows *Windows* 域和多实例队列管理器

Windows 上的多实例队列管理器要求共享其数据和日志。 共享必须可供在不同服务器或工作站上运行的队列管理器的所有实例访问。 配置队列管理器并作为 Windows 域的一部分进行共享。 队列管理器可以在域工作站或服务器上运行，也可以在域控制器上运行。

要点: 缺省情况下，以 Windows 10 V1607 和 Windows Server 2016 开头的计算机比较早版本的 Windows 限制更多。

此更改限制了允许对安全帐户管理器 (SAM) 进行远程调用的客户机，并且可能会影响 IBM MQ ， 导致队列管理器无法启动。 当 IBM MQ 配置为域帐户时，对 SAM 的访问对于 IBM MQ 的运行至关重要。

在配置多实例队列管理器之前，请阅读 第 444 页的『保护 Windows 上的非共享队列管理器数据和日志目录和文件』 和 第 441 页的『保护 Windows 上的共享队列管理器数据和日志目录及文件』 以查看如何控制对队列管理器数据和日志文件的访问。 这些主题是教育性的; 如果要直接设置 Windows 域中多实例队列管理器的共享目录，请参阅 第 419 页的『在 Windows 上的域工作站或服务器上创建多实例队列管理器』。

在域工作站或服务器上运行多实例队列管理器

从 IBM WebSphere MQ 7.1 开始，多实例队列管理器在作为域成员的工作站或服务器上运行。 要在 Windows 上运行多实例队列管理器，您需要一个域控制器，一个文件服务器以及两个工作站或服务器，这些工作站或服务器运行连接到同一域的同一队列管理器。

使您能够在域中的任何服务器或工作站上运行多实例队列管理器的更改是，您现在可以创建具有其他安全组的队列管理器。 附加安全组在 **crtmqm** 命令中的 -a 参数中传递。 通过组保护包含队列管理器数据和日志的目录。 运行队列管理器进程的用户标识必须是该组的成员。 当队列管理器访问这些目录时， Windows 会检查用户标识是否有权访问这些目录。 通过同时指定组和用户标识域作用域，运行队列管理器进程的用户标识

具有来自全局组的凭证。 当队列管理器在另一服务器上运行时，运行队列管理器进程的用户标识可以具有相同的凭证。 用户标识不必相同。 它必须是备用安全组的成员，以及本地 mqm 组的成员。

有关创建多实例队列管理器的详细信息，请参阅 第 419 页的『在 Windows 上的域工作站或服务器上创建多实例队列管理器』 。

需要执行多个步骤来配置域以及域服务器和工作站。 您必须了解 Windows 如何授权队列管理器访问其数据和日志目录。 如果您不确定如何授权队列管理器进程访问其日志和数据文件，请阅读主题 第 444 页的『保护 Windows 上的非共享队列管理器数据和日志目录和文件』。 本主题包含两个任务，用于帮助您了解所需步骤。 任务为 第 445 页的『读取和写入本地 mqm 组授权的数据和日志文件』 和 第 448 页的『读取和写入备用本地安全组授权的数据和日志文件』。 另一主题 第 441 页的『保护 Windows 上的共享队列管理器数据和日志目录及文件』说明了如何使用备用安全组保护包含队列管理器数据和日志文件的共享目录。 本主题包含四个任务，用于设置 Windows 域，创建文件共享，安装 IBM MQ for Windows 以及配置队列管理器以使用该共享。 这些任务如下所示:

1. 第 422 页的『在 Windows 上创建 Active Directory 和 DNS 域』.
2. 第 425 页的『在 Windows 域中的服务器或工作站上安装 IBM MQ』.
3. 第 427 页的『在 Windows 上为队列管理器数据和日志文件创建共享目录』.
4. 第 429 页的『读取和写入备用全局安全组授权的共享数据和日志文件』.

然后，可以使用域来执行任务 第 419 页的『在 Windows 上的域工作站或服务器上创建多实例队列管理器』。 执行以下任务以探索在将知识传输到生产域之前设置多实例队列管理器。

在域控制器上运行多实例队列管理器

可以使用域 mqm 组来保护队列管理器数据。 如 第 441 页的『保护 Windows 上的共享队列管理器数据和日志目录及文件』 主题所述，您无法在工作站或服务器上与本地 mqm 组共享受保护的目录。 但是，在域控制器上，所有组和主体都具有域作用域。 如果在域控制器上安装 IBM MQ for Windows，那么将使用可共享的域 mqm 组来保护队列管理器数据和日志文件。 执行任务 第 432 页的『在 Windows 域控制器上创建多实例队列管理器』 中的步骤以在域控制器上配置多实例队列管理器。

相关信息

管理授权和访问控制

如何使用 Windows Server 集群节点作为域控制器

Windows 在 *Windows* 上的域工作站或服务器上创建多实例队列管理器

示例显示如何在属于 Windows 域的工作站或服务器上的 Windows 上设置多实例队列管理器。 服务器不必是域控制器。 该设置演示了所涉及的概念，而不是生产规模。 此示例基于 Windows Server 2008。 这些步骤在其他版本的 Windows Server 上可能有所不同。

在生产规模配置中，您可能必须将配置定制为现有域。 例如，您可以定义不同的域组以授权不同的共享，并对运行队列管理器的用户标识进行分组。

示例配置由三个服务器组成:

sun
: Windows Server 2008 域控制器。 它拥有包含 *Sun*， *mars* 和 *venus* 的 *wmq.example.com* 域。 为了便于说明，它还用作文件服务器。

mars
: 用作第一个 IBM MQ 服务器的 Windows Server 2008。 它包含名为 *QMGR* 的多实例队列管理器的一个实例。

venus
: 用作第二个 IBM MQ 服务器的 Windows Server 2008。 它包含名为 *QMGR* 的多实例队列管理器的第二个实例。

将示例中的斜体名称替换为您选择的名称。

开始之前

在 Windows 上，您不需要验证计划在其上存储队列管理器数据和日志文件的文件系统。 检查过程 验证共享文件系统行为适用于 AIX and Linux。 在 Windows 上，检查始终成功。

执行以下任务中的步骤。 这些任务将创建域控制器和域，在一台服务器上安装 IBM MQ for Windows，并为数据和日志文件创建文件共享。 如果要配置现有域控制器，您可能会发现在新的 Windows Server 2008 上尝试这些步骤很有用。 您可以调整步骤以适应您的域。

1. 第 422 页的『在 Windows 上创建 Active Directory 和 DNS 域』.
2. 第 425 页的『在 Windows 域中的服务器或工作站上安装 IBM MQ』.
3. 第 427 页的『在 Windows 上为队列管理器数据和日志文件创建共享目录』.
4. 第 429 页的『读取和写入备用全局安全组授权的共享数据和日志文件』.

关于此任务

此任务是一系列任务之一，用于配置域控制器和域中的两个服务器以运行队列管理器的实例。 在此任务中，您将配置第二个服务器 *venus*，以运行队列管理器 *QMGR* 的另一个实例。 遵循此任务中的步骤来创建队列管理器的第二个实例 *QMGR*，并测试它是否正常工作。

此任务与前一部分中的四个任务不同。 它包含将单个实例队列管理器转换为多实例队列管理器的步骤。 所有其他步骤都是单实例或多实例队列管理器的公共步骤。

过程

1. 配置第二个服务器以运行 IBM MQ for Windows。
   a) 执行任务 第 425 页的『在 Windows 域中的服务器或工作站上安装 IBM MQ』 中的步骤以创建第二个域服务器。 在此任务序列中，第二个服务器称为 *venus*。

   提示: 使用两个服务器中每个服务器上的 IBM MQ 的相同安装缺省值来创建第二个安装。 如果缺省值不同，那么您可能必须在 IBM MQ 配置文件 `mqs.ini` 的 *QMGR* **QueueManager** 节中定制 `Prefix` 和 `InstallationName` 变量。 这些变量是指每个服务器上的每个安装和队列管理器可能不同的路径。 如果路径在每个服务器上保持不变，那么配置多实例队列管理器会更简单。
2. 在 *venus* 上创建 *QMGR* 的第二个实例。
   a) 如果 *mars* 上的 *QMGR* 不存在，请执行任务 第 429 页的『读取和写入备用全局安全组授权的共享数据和日志文件』以创建该任务
   b) 检查 `Prefix` 和 `InstallationName` 参数的值对于 *venus* 是否正确。

   在 *mars* 上，运行 **dspmqinf** 命令:

   ```
   dspmqinf QMGR
   ```

   系统响应:

   ```
   QueueManager:
   Name=QMGR
   Directory=QMGR
   Prefix=CMQ
   DataPath=\\sun\wmq\data\QMGR
   InstallationName=Installation1
   ```

   c) 将 **QueueManager** 节的机器可读格式复制到剪贴板。

   在 *mars* 上，使用 `-o command` 参数再次运行 **dspmqinf** 命令。

   ```
   dspmqinf -o command QMGR
   ```

   系统响应:

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="CMQ"
-v DataPath=\\sun\wmq\data\QMGR
```

d) 在 *venus* 上，从剪贴板运行 **addmqinf** 命令以在 *venus* 上创建队列管理器实例。

必要时调整命令以适应 Prefix 或 InstallationName 参数中的差异。

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="CMQ"
-v DataPath=\\sun\wmq\data\QMGR
```

```
IBM MQ configuration information added.
```

3. 在 *venus* 上启动队列管理器 *QMGR*，允许备用实例。

a) 检查 *QMGR* on *mars* 是否已停止。

在 *mars* 上，运行 **dspmq** 命令:

```
dspmq -m QMGR
```

系统响应取决于队列管理器的停止方式; 例如:

```
C:\Users\Administrator>dspmq -m QMGR
QMNAME(QMGR) STATUS(Ended immediately)
```

b) 在 *venus* 上，运行 **strmqm** 命令以启动 *QMGR*，从而允许备用数据库:

```
strmqm -x QMGR
```

系统响应:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

结果

要测试多实例队列管理器切换，请执行以下步骤:

1. 在 *mars* 上，运行 **strmqm** 命令以启动 *QMGR* 允许 standbys:

```
strmqm -x QMGR
```

系统响应:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
A standby instance of queue manager 'QMGR' has been started.
The active instance is running elsewhere.
```

2. 在 *venus* 上，运行 **endmqm** 命令:

```
endmqm -r -s -i QMGR
```

venus 上的系统响应:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended, permitting switchover to
a standby instance.
```

在 *mars* 上:

```
dspmq
QMNAME(QMGR) STATUS(Running as standby)
C:\Users\wmquser2>dspmq
QMNAME(QMGR) STATUS(Running as standby)
C:\Users\wmquser2>dspmq
QMNAME(QMGR)  STATUS(Running)
```

下一步做什么

要使用样本程序验证多实例队列管理器，请参阅 第 439 页的『在 Windows 上验证多实例队列管理器』。

Windows 在 *Windows* 上创建 *Active Directory* 和 *DNS* 域

此任务在名为 *sun* 的 Windows 2008 域控制器上创建域 *wmq.example.com* 。 它使用正确的权限和一个用户在域中配置 Domain mqm 全局组。

在生产规模配置中，您可能必须将配置定制为现有域。 例如，您可以定义不同的域组以授权不同的共享，并对运行队列管理器的用户标识进行分组。

示例配置由三个服务器组成:

sun
: Windows Server 2008 域控制器。 它拥有包含 *Sun*， *mars* 和 *venus* 的 *wmq.example.com* 域。 为了便于说明，它还用作文件服务器。

mars
: 用作第一个 IBM MQ 服务器的 Windows Server 2008。 它包含名为 *QMGR* 的多实例队列管理器的一个实例。

venus
: 用作第二个 IBM MQ 服务器的 Windows Server 2008。 它包含名为 *QMGR* 的多实例队列管理器的第二个实例。

将示例中的斜体名称替换为您选择的名称。

开始之前

1. 任务步骤与已安装但未配置任何角色的 Windows Server 2008 一致。 如果要配置现有域控制器，您可能会发现在新的 Windows Server 2008 上尝试这些步骤很有用。 您可以调整步骤以适应您的域。

关于此任务

在此任务中，您将在新的域控制器上创建 Active Directory 和 DNS 域。 然后，将其配置为准备好在加入该域的其他服务器和工作站上安装 IBM MQ 。 如果您不熟悉安装和配置 Active Directory 以创建 Windows 域，请执行此任务。 必须创建 Windows 域才能创建多实例队列管理器配置。 该任务并非旨在以最佳方式指导您配置 Windows 域。 要在生产环境中部署多实例队列管理器，必须查阅 Windows 文档。

在该任务期间，执行以下步骤:

1. 安装 Active Directory。
2. 添加域

3. 将域添加到 DNS。
4. 创建全局组 Domain mqm 并为其提供正确的权限。
5. 添加用户并使其成为全局组 Domain mqm 的成员。

此任务是一组相关任务之一，用于说明如何访问队列管理器数据和日志文件。 这些任务显示如何创建有权读写存储在所选目录中的数据和日志文件的队列管理器。 它们与任务 第 418 页的『Windows 域和多实例队列管理器』一起执行。

对于该任务，域控制器主机名为 *sun*，两个 IBM MQ 服务器称为 *mars* 和 *venus*。 域称为 *wmq.example.com*。 您可以将任务中的所有斜体名称替换为您自己选择的名称。

过程

1. 以本地管理员或 Workgroup 管理员身份登录到域控制器 *sun*。

   如果服务器已配置为域控制器，那么您必须以域管理员身份登录。
2. 运行 Active Directory 域服务向导。

   a) 单击 **启动** > **运行 ...** 输入 dcpromo ，然后单击 **确定**。

   如果尚未安装 Active Directory 二进制文件，那么 Windows 将自动安装这些文件。
3. 在向导的第一个窗口中，保留 **使用高级方式安装** 复选框。 单击 **下一步** > **下一步** ，然后单击 **在新森林中创建新域** > **下一步**。
4. 在 **森林根域的 FQDN** 字段中输入 *wmq.example.com* 。 单击 **下一步**。
5. 在 "设置森林功能级别" 窗口中，从 **森林功能级别** > **下一步**列表中选择 **Windows Server 2003** 或更高版本。

   IBM MQ 支持的 Windows Server 的最旧级别为 Windows Server 2003。
6. 可选： 在 "设置域功能级别" 窗口中，从 **域功能级别** > **下一步**列表中选择 **Windows Server 2003** 或更高版本。

   仅当将森林功能级别设置为 **Windows Server 2003** 时，才需要执行此步骤。
7. 此时将打开 "其他域控制器选项" 窗口，其中选择了 **DNS 服务器** 作为附加选项。 单击 **下一步** 和 **是** 以清除警告窗口。

   提示: 如果已安装 DNS 服务器，那么不会向您显示此选项。 如果要精确地执行此任务，请从此域控制器中除去所有角色，然后再次启动。
8. 使 Database， Log Files 和 SYSVOL 目录保持不变; 单击 **下一步**。
9. 在 "目录服务复原方式管理员密码" 窗口的 **密码** 和 **确认密码** 字段中输入密码。 单击 **下一步** > **下一步**。在最终向导窗口中选择 **完成时重新引导** 。
10. 域控制器重新引导时，请以 *wmq*\Adminstrator 身份登录。

    服务器管理器将自动启动。
11. 打开 *wmq.example.com*\Users 文件夹

    a) 打开 **服务器管理器** > **角色** > **Active Directory 域服务** > ***wmq.example.com*** > **用户**。
12. 右键单击 **用户** > **新建** > **组**。

    a) 在**组名**字段中输入组名。

    注: 首选的组名是 Domain mqm。 完全按照所示进行输入。

    - 调用组 Domain mqm 将修改域工作站或服务器上 Prepare IBM MQ Wizard 的行为。 它会使 Prepare IBM MQ Wizard 自动将组 Domain mqm 添加到域中 IBM MQ 的每个新安装上的本地 mqm 组。
    - 您可以在没有 Domain mqm 全局组的域中安装工作站或服务器。 如果执行此操作，那么必须定义具有与 Domain mqm 组相同的属性的组。 必须使该组或作为其成员的用户成为本地 mqm 组的成员(无论 IBM MQ 安装在域中的位置)。 您可以将域用户分为多个组。 创建多个域组，每个组对应一组要单独管理的安装。 根据域用户管理的安装，将域用户分为不同域组。 将每个域组添加到不同 IBM MQ 安装的本地 mqm 组。 只有域组中作为特定本地 mqm 组成员的域用户才可以创建、管理和运行此安装的队列管理器。

- 您在域中的工作站或服务器上安装 IBM MQ 时指定的域用户必须是 Domain mqm 组的成员，或者所定义的与 Domain mqm 组具有相同属性的替代组的成员。

b) 单击**全局**，将其保留为**组作用域**，或将其更改为**通用**。 单击**安全性**，将其保留为**组类型**。 单击**确定**。

13. 将权限 **允许 读取组成员资格** 和 **允许 读取 groupMembershipSAM** 添加到 Domain mqm 全局组的权限。

a) 在 Server Manager 操作栏中，单击**视图 > 高级功能**

b) 在 "服务器管理器" 导航树中，单击 **用户**

c) 在 "用户" 窗口中，右键单击 **Domain mqm > 属性**

d) 单击 **安全性 > 高级 > 添加 ...**。 输入 Domain mqm，然后单击 **检查名称 > 确定**。

名称 字段预填充了字符串 Domain mqm (*domain name*\Domain mqm)。

e) 单击**属性**。 在**应用于**列表中，选择**后代用户对象**。

f) 从 **许可权** 列表中，选择 **读取组成员资格** 和 **读取 groupMembershipSAM 允许** 复选框; 单击 **确定 > 应用 > 确定 > 确定**。

14. 将两个或更多用户添加到 Domain mqm 全局组。

示例中的一个用户 *wmquser1* 运行 IBM MQ 服务，另一个用户 *wmquser2* 以交互方式使用。
域用户需要创建在域配置中使用备用安全组的队列管理器。 虽然管理员具有运行 **crtmqm** 命令的权限，但用户标识不足以成为管理员。 可以是管理员的域用户必须是本地 mqm 组以及备用安全组的成员。
在此示例中，您将使 *wmquser1* 和 *wmquser2* 成为 Domain mqm 全局组的成员。 Prepare IBM MQ Wizard 会自动将 Domain mqm 配置为运行向导的本地 mqm 组的成员。
您必须提供不同的用户才能在单台计算机上针对 IBM MQ 的每个安装运行 IBM MQ 服务。 您可以在不同计算机上复用相同的用户。

a) 在 "服务器管理器" 导航树中，单击 **用户 > 新建 > 用户**

b) 在 "新建对象-用户" 窗口中，在 **用户登录名** 字段中输入 *wmquser1* 。 在 **名字** 字段中输入 *WebSphere* ，在 **姓氏** 字段中输入 *MQ1* 。 单击**下一步**。

c) 在 **密码** 和 **确认密码** 字段中输入密码，然后取消选中 **用户必须在下次登录时更改密码** 复选框。 单击 **下一步 > 完成**。

d) 在 "用户" 窗口中，右键单击 ***WebSphere MQ*** **> 添加到组 ...**。 输入 Domain mqm，然后单击 **检查名称 > 确定 > 确定**。

e) 重复步骤 a 到 d 以将 *WebSphere MQ2* 添加为 *wmquser2* 。

15. 将 IBM MQ 作为服务运行。

如果需要将 IBM MQ 作为服务运行，然后授予域用户 (从域管理员处获取) 作为服务运行的访问权，请执行以下过程:

a) 单击 **启动 > 运行 ...**。

输入命令 secpol.msc 并单击**确定**。

b) 打开 **安全设置 > 本地策略 > 用户权限分配**。

在策略列表中，右键单击 **作为服务登录 > 属性**。

c) 单击 **添加用户或组 ...**

输入从域管理员处获取的用户的名称，然后单击 **检查名称**

d) 如果 Windows 安全性窗口提示，请输入具有足够权限的帐户用户或管理员的用户名和密码，然后单击 **确定> 应用> 确定**。

关闭“本地安全策略”窗口。

注: 在 Windows Server 2008 和 Windows Server 2012 上，缺省情况下会启用用户帐户控制 (UAC)。

UAC 功能会限制用户在某些操作系统工具上执行的操作，即使他们是管理员组的成员。 您必须采取相应的步骤才能克服此限制。

下一步做什么

继续执行下一个任务 第 425 页的『在 Windows 域中的服务器或工作站上安装 IBM MQ』。

相关任务

Windows 在 Windows 域中的服务器或工作站上安装 IBM MQ

Windows 在 Windows 上为队列管理器数据和日志文件创建共享目录

Windows 读取和写入备用全局安全组授权的共享数据和日志文件

Windows 在 *Windows* 域中的服务器或工作站上安装 *IBM MQ*

在此任务中，您将在 *wmq.example.com* Windows 域中的服务器或工作站上安装和配置 IBM MQ 。

在生产规模配置中，您可能必须将配置定制为现有域。 例如，您可以定义不同的域组以授权不同的共享，并对运行队列管理器的用户标识进行分组。

示例配置由三个服务器组成:

sun
Windows Server 2008 域控制器。 它拥有包含 *Sun*， *mars* 和 *venus* 的 *wmq.example.com* 域。 为了便于说明，它还用作文件服务器。

mars
用作第一个 IBM MQ 服务器的 Windows Server 2008。 它包含名为 *QMGR* 的多实例队列管理器的一个实例。

venus
用作第二个 IBM MQ 服务器的 Windows Server 2008。 它包含名为 *QMGR* 的多实例队列管理器的第二个实例。

将示例中的斜体名称替换为您选择的名称。

开始之前

要点: 缺省情况下，以 Windows 10 V1607 和 Windows Server 2016 开头的计算机比较早版本的 Windows 限制更多。

此更改限制了允许对安全帐户管理器 (SAM) 进行远程调用的客户机，并且可能会影响 IBM MQ，导致队列管理器无法启动。 当 IBM MQ 配置为域帐户时，对 SAM 的访问对于 IBM MQ 的运行至关重要。

1. 执行 第 422 页的『在 Windows 上创建 Active Directory 和 DNS 域』 中的步骤为域 *wmq.example.com* 创建域控制器 *sun*。 更改斜体名称以适合您的配置。
2. 请参阅 Windows 系统上的硬件和软件需求，以了解可在其上运行 IBM MQ 的其他 Windows 版本。

关于此任务

在此任务中，您将名为 *mars* 的 Windows Server 2008 配置为 *wmq.example.com* 域的成员。 安装 IBM MQ，并将安装配置为作为 *wmq.example.com* 域的成员运行。

此任务是一组相关任务之一，用于说明如何访问队列管理器数据和日志文件。 这些任务显示如何创建有权读写存储在所选目录中的数据和日志文件的队列管理器。 它们与任务 第 418 页的『Windows 域和多实例队列管理器』一起执行。

对于该任务，域控制器主机名为 *sun*，两个 IBM MQ 服务器称为 *mars* 和 *venus*。 域称为 *wmq.example.com*。 您可以将任务中的所有斜体名称替换为您自己选择的名称。

过程

1. 将域控制器 *sun.wmq.example.com* 添加到 *mars* 作为 DNS 服务器。
   a) 在 *mars* 上，以 *mars*\Administrator 身份登录，然后单击 **启动**。
   b) 右键单击 **网络** > **属性** > **管理网络连接**。

c) 右键单击网络适配器，然后单击 **属性**。

系统通过 "本地区域连接属性" 窗口进行响应，该窗口列出了连接所使用的项。

d) 从 "局域网连接属性" 窗口的项列表中选择 **Internet Protocol 版本 4** 或 **Internet Protocol IBM WebSphere MQ 6** 。 单击 **属性** > **高级 ...** 然后单击 **DNS** 选项卡。

e) 在 DNS 服务器地址下，单击 **添加 ...**。

f) 输入域控制器 (也是 DNS 服务器) 的 IP 地址，然后单击 **添加**。

g) 单击 **附加这些 DNS 后缀** > **添加 ...**。

h) 输入 *wmq.example.com* ，然后单击 **添加**。

i) 在 **此连接的 DNS 后缀** 字段中输入 *wmq.example.com* 。

j) 选择 **在 DNS 中注册此连接的地址** ，并 **在 DNS 注册中使用此连接的后缀**。 单击 **确定** > **确定** > **关闭**

k) 打开命令窗口，然后输入命令 **ipconfig /all** 以查看 TCP/IP 设置。

2. 在 *mars* 上，将计算机添加到 *wmq.example.com* 域。

a) 单击**开始**

b) 右键单击 **计算机** > **属性**。 在 "计算机名称" ， "域" 和 "工作组设置" 部分中，单击 **更改设置**。

c) 在 "系统属性" 窗口中，单击 **更改 ...**。

d) 单击 "域" ，输入 *wmq.example.com*，然后单击 **确定**。

e) 输入有权允许计算机加入域的域控制器管理员的 **用户名** 和 **密码** ，然后单击 **确定**。

f) 单击 **确定** > **确定** > **关闭** > **立即重新启动** 以响应 “欢迎使用 *wmq.example.com* 域” 消息。

3. 检查计算机是否是 *wmq.example.com* 域的成员

a) 在 *sun* 上，以 *wmq*\Administrator 身份登录到域控制器。

b) 打开 **服务器管理器** > **Active Directory 域服务** > ***wmq.example.com*** > **计算机** 并检查 *mars* 是否在 "计算机" 窗口中正确列出。

4. 在 *mars* 上安装 IBM MQ for Windows。

有关运行 IBM MQ for Windows 安装向导的更多信息; 请参阅 在 Windows 上安装 IBM MQ 服务器 。

a) 在 *mars* 上，以本地管理员 *mars*\Administrator 身份登录。

b) 在 IBM MQ for Windows 安装介质上运行 **Setup** 命令。

IBM MQ 启动板应用程序将启动。

c) 单击 **软件需求** 以检查是否已安装必备软件。

d) 单击 **网络配置** > **是** 以配置域用户标识。

任务 第 422 页的『在 Windows 上创建 Active Directory 和 DNS 域』为这组任务配置域用户标识。

e) 单击 **IBM MQ 安装**，选择安装语言，然后单击 "启动 IBM MQ 安装程序"。

f) 确认许可协议，然后单击 **下一步** > **下一步** > **安装** 以接受缺省配置。 等待安装完成，然后单击 **完成**。

您可能想要更改安装的名称，安装不同的组件，为队列管理器数据和日志配置不同的目录，或者安装到不同的目录中。 如果是这样，请单击 **定制** 而不是 **典型**。

IBM MQ 已安装，安装程序将启动 Prepare IBM MQ Wizard。

要点: 请不要运行向导。

5. 配置要使用 **作为服务运行** 权限来运行 IBM MQ 服务的用户。

选择是配置本地 mqm 组， Domain mqm 组，还是配置要使用权限运行 IBM MQ 服务的用户。 在该示例中，您将向用户授予权限。

a) 单击 **启动** > **运行 ...**， 输入命令 **secpol.msc** ，然后单击 **确定**。

b) 打开 **安全设置** > **本地策略** > **用户权限分配**。 在策略列表中，右键单击 **作为服务登录** > **属性** 。

c) 单击 **添加用户或组 ...** 并输入 *wmquser1* ，然后单击 **检查名称**

d) 输入域管理员 *wmq*\Administrator 的用户名和密码，然后单击 **确定** > **应用** > **确定**。 关闭“本地安全策略”窗口。

6. 运行 Prepare IBM MQ Wizard。

   有关更多信息，请参阅 使用 Prepare IBM MQ Wizard 配置 IBM MQ。

   a) IBM MQ 安装程序会自动运行 Prepare IBM MQ Wizard 。

      要手动启动向导，请在 **启动** > **所有程序** > **IBM MQ** 文件夹中找到 Prepare IBM MQ Wizard 的快捷方式。 选择与多安装配置中的 IBM MQ 安装相对应的快捷方式。

   b) 单击 **下一步** ，并保持单击 **是** 以响应问题 “确定网络中是否存在 Windows 2000 或更高版本的域控制器”。

   c) 在第一个 "为 Windows 域用户配置 IBM MQ for Windows " 窗口中，单击 **是** > **下一步** 。

   d) 在第二个 "为 Windows 域用户配置 IBM MQ for Windows " 窗口中，在 **域** 字段中输入 *wmq* 。 在 **用户名** 字段中输入 *wmquser1* ，并在 **密码** 字段中输入密码 (如果设置了密码)。 单击**下一步**。

      向导使用 *wmquser1* 配置并启动 IBM MQ 。

   e) 在向导的最后一页中，根据需要选中或清除复选框，然后单击 **完成**。

下一步做什么

1. 执行任务 第 445 页的『读取和写入本地 mqm 组授权的数据和日志文件』以验证安装和配置是否正常工作。
2. 执行任务 第 427 页的『在 Windows 上为队列管理器数据和日志文件创建共享目录』以配置文件共享来存储多实例队列管理器的数据和日志文件。

相关任务

Windows 在 Windows 上创建 Active Directory 和 DNS 域

Windows 在 Windows 上为队列管理器数据和日志文件创建共享目录

Windows 读取和写入备用全局安全组授权的共享数据和日志文件

相关参考

IBM MQ Windows 服务所需的用户权限

Windows 在 *Windows* 上为队列管理器数据和日志文件创建共享目录

此任务是一组相关任务之一，用于说明如何访问队列管理器数据和日志文件。 这些任务显示如何创建有权读写存储在所选目录中的数据和日志文件的队列管理器。

在生产规模配置中，您可能必须将配置定制为现有域。 例如，您可以定义不同的域组以授权不同的共享，并对运行队列管理器的用户标识进行分组。

示例配置由三个服务器组成:

sun
Windows Server 2008 域控制器。 它拥有包含 *Sun*， *mars* 和 *venus* 的 *wmq.example.com* 域。 为了便于说明，它还用作文件服务器。

mars
用作第一个 IBM MQ 服务器的 Windows Server 2008。 它包含名为 *QMGR* 的多实例队列管理器的一个实例。

venus
用作第二个 IBM MQ 服务器的 Windows Server 2008。 它包含名为 *QMGR* 的多实例队列管理器的第二个实例。

将示例中的斜体名称替换为您选择的名称。

开始之前

1. 要完全按照记录执行此任务，请执行任务 第 422 页的『在 Windows 上创建 Active Directory 和 DNS 域』中的步骤，以在域控制器 *sun* 上创建域 *sun.wmq.example.com* 。 更改斜体名称以适合您的配置。

关于此任务

此任务是一组相关任务之一，用于说明如何访问队列管理器数据和日志文件。 这些任务显示如何创建有权读写存储在所选目录中的数据和日志文件的队列管理器。 它们与任务 第 418 页的『Windows 域和多实例队列管理器』一起执行。

在该任务中，创建包含数据和日志目录的共享以及授权访问该共享的全局组。 将授权共享的全局组的名称传递到其 -a 参数中的 **crtmqm** 命令。 全局组使您能够灵活地将此共享的用户与其他共享的用户分开。 如果不需要此灵活性，请授权与 Domain mqm 组共享，而不是创建新的全局组。

此任务中用于共享的全局组称为 *wmqha*，共享称为 *wmq*。 它们是在 Windows 域 *wmq.example.com* 中的域控制器 *sun* 上定义的。 共享对全局组 *wmqha* 具有完全控制许可权。 将任务中的斜体名称替换为您选择的名称。

对于此任务，域控制器与文件服务器是同一服务器。 在实际应用程序中，在不同服务器之间拆分目录和文件服务以获取性能和可用性。

您必须将队列管理器运行时所使用的用户标识配置为两个组的成员。 它必须是 IBM MQ 服务器上的本地 mqm 组以及 *wmqha* 全局组的成员。

在此任务集中，当队列管理器作为服务运行时，它以用户标识 *wmquser1* 运行，因此 *wmquser1* 必须是 *wmqha* 的成员。 当队列管理器以交互方式运行时，它以用户标识 *wmquser2* 运行，因此 *wmquser2* 必须是 *wmqha* 的成员。 *wmquser1* 和 *wmquser2* 都是全局组 Domain mqm 的成员。 Domain mqm 是 *mars* 和 *venus* IBM MQ 服务器上本地 mqm 组的成员。 因此， *wmquser1* 和 *wmquser2* 是两个 IBM MQ 服务器上的本地 mqm 组的成员。

过程

1. 以域管理员身份登录到域控制器 *sun.wmq.example.com* 。
2. 创建全局组 *wmqha*。
   a) 打开 **服务器管理器** > **角色** > **Active Directory 域服务** > ***wmq.example.com*** > **用户**。
   b) 打开 *wmq.example.com*\Users 文件夹
   c) 右键单击 **用户** > **新建** > **组**。
   d) 在 **组名** 字段中输入 *wmqha* 。
   e) 将 **全局** 作为 **组作用域** 单击，将 **安全性** 作为 **组类型**单击。 单击**确定**。
3. 将域用户 *wmquser1* 和 *wmquser2* 添加到全局组 *wmqha*。
   a) 在 "服务器管理器" 导航树中，单击 **用户** ，然后右键单击用户列表中的 ***wmqha*** > **属性** 。
   b) 单击 *wmqha* "属性" 窗口中的 "成员" 选项卡。
   c) 单击 **添加 ...** ; 输入 *wmquser1* ； *wmquser2* ，然后单击 **检查名称** > **确定** > **应用** > **确定**。
4. 创建目录树以包含队列管理器数据和日志文件。
   a) 打开命令提示符。
   b) 输入以下命令:

```
md c:\wmq\data, c:\wmq\logs
```

5. 授权全局组 *wmqha* 对 *c:\wmq* 目录和共享具有完全控制许可权。
   a) 在 Windows Explorer 中，右键单击 ***c:\wmq*** > **属性**。
   b) 单击 **安全性** 选项卡，然后单击 **高级** > **编辑 ...**。
   c) 取消选中 **包括来自此对象的所有者的可继承许可权**复选框。 在 " Windows 安全性" 窗口中单击 **复制** 。

d) 在 **许可权条目** 列表中选择用户的行，然后单击 **除去**。 在 **许可权条目**列表中保留 SYSTEM , Administrators 和 CREATOR OWNER 的行。

e) 单击 **添加 ...**， 并输入全局组 *wmqha* 的名称。 单击 **检查名称** > **确定**。

f) 在 " wmq 的许可权条目" 窗口中，在 **许可权**列表中选择 **完全控制** 。

g) 单击 **确定** > **应用** > **确定** > **确定** > **确定**

h) 在 Windows Explorer 中，右键单击 ***c: \wmq*** > **共享 ...**。

i) 单击 **高级共享 ...** 并选中 **共享此文件夹** 复选框。 将共享名称保留为 *wmq*。

j) 单击 **许可权** > **添加 ...**。 并输入全局组 *wmqha* 的名称。 单击 **检查名称** > **确定**。

k) 在 **组或用户名**列表中选择 *wmqha* 。 选中 ***wmqha* 的许可权** 列表中的 **完全控制** 复选框; 单击 **应用**。

l) 在 **组或用户名**列表中选择 *Administrators* 。 选中 **管理员的许可权** 列表中的 **完全控制** 复选框; 单击 **应用** > **确定** > **确定** > **关闭**。

下一步做什么

检查是否可以从每个 IBM MQ 服务器读取文件并将其写入共享目录。 检查 IBM MQ 服务用户标识 *wmquser1* 和交互式用户标识 *wmquser2*。

1. 如果您正在使用远程桌面，那么必须将 *wmq**wmquser1* 和 *wmquser2* 添加到 *mars* 上的本地组 Remote Desktop Users 。
   a. 以 *wmq*\Administrator 身份登录到 *mars*
   b. 运行 **lusrmgr.msc** 命令以打开 "本地用户和组" 窗口。
   c. 单击**组**。 右键单击 **远程桌面用户** > **属性** > **添加 ...**。 输入 *wmquser1* ； *wmquser2* ，然后单击 **检查名称**。
   d. 输入域管理员 *wmq*\Administrator 的用户名和密码，然后单击 **确定** > **应用** > **确定**。
   e. 关闭 "本地用户和组" 窗口。
2. 以 *wmq**wmquser1* 身份登录到 *mars* 。
   a. 打开 " Windows 资源管理器" 窗口，然后输入 *sun**wmq*。

      系统通过打开 *sun.wmq.example.com* 上的 *wmq* 共享进行响应，并列出数据和日志目录。
   b. 通过在 data 子目录中创建文件，添加一些内容，读取内容，然后将其删除，来检查 *wmquser1* 的许可权。
3. 以 *wmq**wmquser2* 身份登录到 *mars* ，然后重复检查。
4. 执行下一个任务，以创建队列管理器来使用共享数据和日志目录; 请参阅 第 429 页的『读取和写入备用全局安全组授权的共享数据和日志文件』。

相关任务

Windows 在 Windows 上创建 Active Directory 和 DNS 域

Windows 在 Windows 域中的服务器或工作站上安装 IBM MQ

Windows 读取和写入备用全局安全组授权的共享数据和日志文件

Windows 读取和写入备用全局安全组授权的共享数据和日志文件

此任务显示如何在 **crtmqm** 命令上使用 -a 标志。 -a 标志允许队列管理器使用备用安全组访问其在远程文件共享上的日志和数据文件。

在生产规模配置中，您可能必须将配置定制为现有域。 例如，您可以定义不同的域组以授权不同的共享，并对运行队列管理器的用户标识进行分组。

示例配置由三个服务器组成:

sun
Windows Server 2008 域控制器。 它拥有包含 *Sun*， *mars* 和 *venus* 的 *wmq.example.com* 域。 为了便于说明，它还用作文件服务器。

mars
用作第一个 IBM MQ 服务器的 Windows Server 2008。 它包含名为 *QMGR* 的多实例队列管理器的一个实例。

venus
用作第二个 IBM MQ 服务器的 Windows Server 2008。 它包含名为 *QMGR* 的多实例队列管理器的第二个实例。

将示例中的斜体名称替换为您选择的名称。

开始之前

执行以下任务中的步骤。 这些任务将创建域控制器和域，在一台服务器上安装 IBM MQ for Windows，并为数据和日志文件创建文件共享。 如果要配置现有域控制器，您可能会发现在新的 Windows Server 2008 上尝试这些步骤很有用。 您可以调整步骤以适应您的域。

1. 第 422 页的『在 Windows 上创建 Active Directory 和 DNS 域』.
2. 第 425 页的『在 Windows 域中的服务器或工作站上安装 IBM MQ』.
3. 第 427 页的『在 Windows 上为队列管理器数据和日志文件创建共享目录』.

关于此任务

此任务是一组相关任务之一，用于说明如何访问队列管理器数据和日志文件。 这些任务显示如何创建有权读写存储在所选目录中的数据和日志文件的队列管理器。 它们与任务 第 418 页的『Windows 域和多实例队列管理器』一起执行。

在此任务中，您将创建一个队列管理器，该队列管理器将其数据和日志存储在文件服务器上的远程目录中。 对于本示例，文件服务器与域控制器是同一服务器。 包含数据和日志文件夹的目录与授予全局组 wmqha 的完全控制许可权共享。

过程

1. 以本地管理员 *mars*\Administrator 身份登录到域服务器 *mars*。
2. 打开命令窗口。
3. 重新启动 IBM MQ 服务。

   您必须重新启动服务，以使其运行时使用的用户标识获取为其配置的其他安全凭证。

   输入命令:

   ```
   endmqsvc
   strmqsvc
   ```

   系统响应:

   ```
   5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
   The MQ service for installation 'Installation1' ended successfully.
   ```

   并:

   ```
   5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
   The MQ service for installation 'Installation1' started successfully.
   ```
4. 创建队列管理器。

   ```
   crtmqm -a wmq\wmqha -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\sun\wmq\data -ld \\sun\wmq\logs
   QMGR
   ```

   必须通过指定全局组 *"wmq\wmqha"*的完整域名来指定备用安全组 *wmqha* 的域 *wmq*。

   您必须阐明共享 *sun**wmq* 的通用命名约定 (UNC) 名称，而不能使用映射的驱动器引用。

系统响应：

```
IBM MQ queue manager created.
Directory '\\sun\wmq\data\QMGR' created.
The queue manager is associated with installation '1'
Creating or replacing default objects for queue manager 'QMGR'
Default objects statistics : 74 created. 0 replaced.
Completing setup.
Setup completed.
```

下一步做什么

通过将消息放入队列并获取消息来测试队列管理器。

1. 启动队列管理器。

   ```
   strmqm QMGR
   ```

   系统响应：

   ```
   IBM MQ queue manager 'QMGR' starting.
   The queue manager is associated with installation '1'.
   5 log records accessed on queue manager 'QMGR' during the log
   replay phase.
   Log replay for queue manager 'QMGR' complete.
   Transaction manager state recovered for queue manager 'QMGR'.
   IBM MQ queue manager 'QMGR' started using V7.1.0.0.
   ```

2. 创建测试队列。

   ```
   echo define qlocal(QTEST) | runmqsc QMGR
   ```

   系统响应：

   ```
   5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
   Starting MQSC for queue manager QMGR.


   1 : define qlocal(QTEST)
   AMQ8006: IBM MQ queue created.
   One MQSC command read.
   No commands have a syntax error.
   All valid MQSC commands were processed.
   ```

3. 使用样本程序 **amqsput** 放置测试消息。

   ```
   echo 'A test message' | amqsput QTEST QMGR
   ```

   系统响应：

   ```
   Sample AMQSPUT0 start
   target queue is QTEST
   Sample AMQSPUT0 end
   ```

4. 使用样本程序 **amqsget** 获取测试消息。

   ```
   amqsget QTEST QMGR
   ```

   系统响应：

```
Sample AMQSGET0 start
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGET0 end
```

5. 停止队列管理器。

   ```
   endmqm -i QMGR
   ```

   系统响应：

   ```
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ended.
   ```

6. 删除队列管理器。

   ```
   dltmqm QMGR
   ```

   系统响应：

   ```
   IBM MQ queue manager 'QMGR' deleted.
   ```

7. 删除您创建的目录。

   提示： 将 /Q 选项添加到命令中，以防止命令提示删除每个文件或目录。

   ```
   del /F /S C:\wmq\*.*
   rmdir /S C:\wmq
   ```

相关任务

Windows 在 Windows 上创建 Active Directory 和 DNS 域

Windows 在 Windows 域中的服务器或工作站上安装 IBM MQ

Windows 在 Windows 上为队列管理器数据和日志文件创建共享目录

Windows 在 *Windows* 域控制器上创建多实例队列管理器

示例显示如何在域控制器上的 Windows 上设置多实例队列管理器。 该设置演示了所涉及的概念，而不是生产规模。 此示例基于 Windows Server 2008。 这些步骤在其他版本的 Windows Server 上可能有所不同。

该配置使用迷你域或 “domainlet” 的概念; 请参阅 Windows 2000， Windows Server 2003 和 Windows Server 2008 集群节点作为域控制器。 要将多实例队列管理器添加到现有域，请参阅 第 419 页的『在 Windows 上的域工作站或服务器上创建多实例队列管理器』。

示例配置由三个服务器组成:

sun
: 用作第一个域控制器的 Windows Server 2008 服务器。 它定义包含 *sun*， *earth* 和 *mars* 的 *wmq.example.com* 域。 它包含名为 *QMGR* 的多实例队列管理器的一个实例。

earth
: Windows Server 2008 用作第二个域控制器 IBM MQ 服务器。 它包含名为 *QMGR* 的多实例队列管理器的第二个实例。

mars
: 用作文件服务器的 Windows Server 2008。

将示例中的斜体名称替换为您选择的名称。

开始之前

1. 在 Windows 上，您不需要验证计划在其上存储队列管理器数据和日志文件的文件系统。 检查过程 验证共享文件系统行为适用于 AIX and Linux。 在 Windows 上，检查始终成功。
2. 执行 第 422 页的『在 Windows 上创建 Active Directory 和 DNS 域』 中的步骤以创建第一个域控制器。
3. 执行 第 435 页的『将第二个 Windows 域控制器添加到示例域』 中的步骤以添加第二个域控制器，在两个域控制器上安装 IBM MQ for Windows ，并验证安装。
4. 执行 第 437 页的『在示例域中的 Windows 域控制器上安装 IBM MQ』 中的步骤以在两个域控制器上安装 IBM MQ 。

关于此任务

在同一域中的文件服务器上，为队列管理器日志和数据目录创建共享。 接下来，创建在其中一个域控制器上使用文件共享的多实例队列管理器的第一个实例。 在另一个域控制器上创建另一个实例，最后验证配置。 您可以在域控制器上创建文件共享。

在样本中， *sun* 是第一个域控制器， *earth* 是第二个域控制器， *mars* 是文件服务器。

过程

1. 创建要包含队列管理器数据和日志文件的目录。
   a) 在 *mars* 上，输入命令:

   ```
   md c:\wmq\data , c:\wmq\logs
   ```

2. 共享要包含队列管理器数据和日志文件的目录。

   您必须允许对域本地组 mqm 以及用于创建队列管理器的用户标识进行完全控制访问。 在此示例中，作为 Domain Administrators 成员的用户标识具有创建队列管理器的权限。

   文件共享必须位于与域控制器位于同一域中的服务器上。 在此示例中，服务器 *mars* 与域控制器位于同一域中。

   a) 在 Windows Explorer 中，右键单击 ***c: \wmq*** > **属性**。
   b) 单击 **安全性** 选项卡，然后单击 **高级** > **编辑 ...**。
   c) 取消选中 **包括来自此对象的所有者的可继承许可权**复选框。 在 " Windows 安全性" 窗口中单击 **复制** 。
   d) 在 **许可权条目** 列表中选择用户的行，然后单击 **除去**。 在 **许可权条目**列表中保留 SYSTEM , Administrators 和 CREATOR OWNER 的行。
   e) 单击 **添加 ...**， 并输入域本地组 *mqm* 的名称。 单击 **检查名称**
   f) 作为对 " Windows 安全性" 窗口的响应，输入 Domain Administrator 的名称和密码，然后单击 **确定** > **确定**。
   g) 在 " wmq 的许可权条目" 窗口中，在 **许可权**列表中选择 **完全控制** 。
   h) 单击 **确定** > **应用** > **确定** > **确定** > **确定**
   i) 重复步骤 e 到 h 以添加 Domain Administrators。
   j) 在 Windows Explorer 中，右键单击 ***c: \wmq*** > **共享 ...**。
   k) 单击 **高级共享 ...** 并选中 **共享此文件夹** 复选框。 将共享名称保留为 *wmq*。
   l) 单击 **许可权** > **添加 ...**。 并输入域本地组 *mqm* ； Domain Administrators 的名称。 单击**检查名称**。
   m) 作为对 " Windows 安全性" 窗口的响应，输入 Domain Administrator 的名称和密码，然后单击 **确定** > **确定**。
3. 在第一个域控制器 *sun* 上创建队列管理器 *QMGR* 。

   ```
   crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\mars\wmq\data -ld \\mars\wmq\logs QMGR
   ```

系统响应：

```
IBM MQ queue manager created.
Directory '\\mars\wmq\data\QMGR' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QMGR'.
Default objects statistics : 74 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

4. 在 *sun* 上启动队列管理器，允许备用实例。

```
strmqm -x QMGR
```

系统响应：

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

5. 在 *earth* 上创建 *QMGR* 的第二个实例。

a) 检查 Prefix 和 InstallationName 参数的值对于 *earth* 是否正确。

在 *sun* 上，运行 **dspmqinf** 命令：

```
dspmqinf QMGR
```

系统响应：

```
QueueManager:
Name=QMGR
Directory=QMGR
Prefix=CMQ
DataPath=\\mars\wmq\data\QMGR
InstallationName=Installation1
```

b) 将 **QueueManager** 节的机器可读格式复制到剪贴板。

在 *sun* 上，使用 -o command 参数再次运行 **dspmqinf** 命令。

```
dspmqinf -o command QMGR
```

系统响应：

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="CMQ"
-v DataPath=\\mars\wmq\data\QMGR
```

c) 在 *earth* 上，从剪贴板运行 **addmqinf** 命令以在 *earth* 上创建队列管理器实例。

必要时调整命令以适应 Prefix 或 InstallationName 参数中的差异。

```
addmqinf -s QueueManager -v Name= QMGR
-v Directory= QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"
-v DataPath=\\mars\wmq\data\QMGR
```

```
  IBM MQ configuration information added.
```

6. 在 *earth* 上启动队列管理器的备用实例。

```
strmqm -x QMGR
```

系统响应:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
A standby instance of queue manager 'QMGR' has been started. The active
instance is running elsewhere.
```

结果

验证队列管理器是否从 *sun* 切换到 *earth*:

1. 在 *sun* 上，运行以下命令:

```
endmqm -i -r -s QMGR
```

sun 上的系统响应:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended, permitting switchover to
a standby instance.
```

2. 在 *earth* 上，重复输入命令:

```
dspmq
```

系统响应:

```
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running)
```

下一步做什么

要使用样本程序验证多实例队列管理器，请参阅 第 439 页的『在 Windows 上验证多实例队列管理器』。

相关任务

第 435 页的『将第二个 Windows 域控制器添加到示例域』
第 437 页的『在示例域中的 Windows 域控制器上安装 IBM MQ』

相关信息

Windows 2000， Windows Server 2003 和 Windows Server 2008 集群节点作为域控制器

Windows 将第二个 *Windows* 域控制器添加到示例域

将第二个域控制器添加到 *wmq.example.com* 域以构造 Windows 域，在该域中在域控制器和文件服务器上运行多实例队列管理器。

示例配置由三个服务器组成:

sun
用作第一个域控制器的 Windows Server 2008 服务器。 它定义包含 *sun*， *earth* 和 *mars* 的 *wmq.example.com* 域。 它包含名为 *QMGR* 的多实例队列管理器的一个实例。

earth
Windows Server 2008 用作第二个域控制器 IBM MQ 服务器。 它包含名为 *QMGR* 的多实例队列管理器的第二个实例。

mars
用作文件服务器的 Windows Server 2008。

将示例中的斜体名称替换为您选择的名称。

开始之前

1. 执行 第 422 页的『在 Windows 上创建 Active Directory 和 DNS 域』 中的步骤为域 *wmq.example.com* 创建域控制器 *sun*。 更改斜体名称以适合您的配置。
2. 在缺省工作组 WORKGROUP 中的服务器上安装 Windows Server 2008。 对于此示例，服务器名为 *earth*。

关于此任务

在此任务中，您将名为 *earth* 的 Windows Server 2008 配置为 *wmq.example.com* 域中的第二个域控制器。

此任务是一组相关任务之一，用于说明如何访问队列管理器数据和日志文件。 这些任务显示如何创建有权读写存储在所选目录中的数据和日志文件的队列管理器。 它们与任务 第 418 页的『Windows 域和多实例队列管理器』一起执行。

过程

1. 将域控制器 *sun.wmq.example.com* 添加到 *earth* 作为 DNS 服务器。
   a) 在 *earth* 上，以 *earth*\Administrator 身份登录，然后单击 **启动**。
   b) 右键单击 **网络** > **属性** > **管理网络连接**。
   c) 右键单击网络适配器，然后单击 **属性**。

      系统通过 "本地区域连接属性" 窗口进行响应，该窗口列出了连接所使用的项。
   d) 从 "局域网连接属性" 窗口的项列表中选择 **Internet Protocol 版本 4** 或 **Internet Protocol IBM WebSphere MQ 6** 。 单击 **属性** > **高级 ...** 然后单击 **DNS** 选项卡。
   e) 在 DNS 服务器地址下，单击 **添加 ...**。
   f) 输入域控制器 (也是 DNS 服务器) 的 IP 地址，然后单击 **添加**。
   g) 单击 **附加这些 DNS 后缀** > **添加 ...**。
   h) 输入 *wmq.example.com* ，然后单击 **添加**。
   i) 在 **此连接的 DNS 后缀** 字段中输入 *wmq.example.com* 。
   j) 选择 **在 DNS 中注册此连接的地址** ，并 **在 DNS 注册中使用此连接的后缀**。 单击 **确定** > **确定** > **关闭**
   k) 打开命令窗口，然后输入命令 **ipconfig /all** 以查看 TCP/IP 设置。
2. 以本地管理员或 Workgroup 管理员身份登录到域控制器 *sun*。

   如果服务器已配置为域控制器，那么您必须以域管理员身份登录。
3. 运行 Active Directory 域服务向导。
   a) 单击 **启动** > **运行 ...** 输入 dcpromo ，然后单击 **确定**。

   如果尚未安装 Active Directory 二进制文件，那么 Windows 将自动安装这些文件。
4. 将 *earth* 配置为 *wmq.example.com* 域中的第二个域控制器。
   a) 在向导的第一个窗口中，保留 **使用高级方式安装** 复选框。 单击 **下一步** > **下一步** ，然后单击 **创建将域控制器添加到现有域** > **下一步**。

b) 在 **输入此林中任何域的名称 ...** 中输入 *wmq* 字段。 单击 **备用凭证** 单选按钮，然后单击 **设置 ...**。 输入域管理员的名称和密码，然后单击 **确定** > **下一步** > **下一步** > **下一步**。

c) 在 "其他域控制器选项" 窗口中，接受已选择的 **DNS 服务器** 和 **全局目录** 选项; 单击 **下一步** > **下一步**。

d) 在 "目录服务复原方式管理员密码" 上，输入 **密码** 和 **确认密码**，然后单击 **下一步** > **下一步**。

e) 提示输入 **网络凭证**时，请输入域管理员的密码。 在最终向导窗口中选择 **完成时重新引导** 。

f) 一段时间后，可能会打开一个窗口，其中包含有关 DNS 授权的 **DCPromo** 错误; 请单击 **确定**。 服务器将重新引导。

结果

当 *earth* 已重新引导时，请以域管理员身份登录。 检查 wmq.example.com 域是否已复制到 *earth*。

下一步做什么

继续安装 IBM MQ ; 请参阅 第 437 页的『在示例域中的 Windows 域控制器上安装 IBM MQ』。

相关任务

Windows 在示例域中的 Windows 域控制器上安装 IBM MQ

第 422 页的『在 Windows 上创建 Active Directory 和 DNS 域』

Windows 在示例域中的 *Windows* 域控制器上安装 *IBM MQ*

在 *wmq.example.com* 域中的两个域控制器上安装和配置 IBM MQ 的安装。

示例配置由三个服务器组成:

sun
用作第一个域控制器的 Windows Server 2008 服务器。 它定义包含 *sun*， *earth* 和 *mars* 的 *wmq.example.com* 域。 它包含名为 *QMGR* 的多实例队列管理器的一个实例。

earth
Windows Server 2008 用作第二个域控制器 IBM MQ 服务器。 它包含名为 *QMGR* 的多实例队列管理器的第二个实例。

mars
用作文件服务器的 Windows Server 2008。

将示例中的斜体名称替换为您选择的名称。

开始之前

1. 执行 第 422 页的『在 Windows 上创建 Active Directory 和 DNS 域』 中的步骤为域 *wmq.example.com* 创建域控制器 *sun*。 更改斜体名称以适合您的配置。
2. 执行 第 435 页的『将第二个 Windows 域控制器添加到示例域』 中的步骤为域 *wmq.example.com* 创建第二个域控制器 *earth*。 更改斜体名称以适合您的配置。
3. 请参阅 Windows 系统上的硬件和软件需求 ， 以了解可在其上运行 IBM MQ 的其他 Windows 版本。

关于此任务

在 *wmq.example.com* 域中的两个域控制器上安装和配置 IBM MQ 的安装。

过程

1. 在 *sun* 和 *earth* 上安装 IBM MQ 。

   有关更多信息，请参阅 在 Windows 上安装 IBM MQ 服务器 。

   a) 在 *sun* 和 *earth* 上，以域管理员 *wmq*\Administrator 身份登录。

   b) 在 IBM MQ for Windows 安装介质上运行 **Setup** 命令。

IBM MQ 启动板应用程序将启动。

c) 单击 **软件需求** 以检查是否已安装必备软件。

d) 单击 **网络配置** > **否**。

您可以为此安装配置域用户标识，也可以不配置域用户标识。 创建的用户标识是域本地用户标识。

e) 单击 **IBM MQ 安装**，选择安装语言，然后单击 "启动 IBM MQ 安装程序"。

f) 确认许可协议，然后单击 **下一步** > **下一步** > **安装** 以接受缺省配置。 等待安装完成，然后单击 **完成**。

如果要更改安装的名称，安装不同的组件，为队列管理器数据和日志配置不同的目录，或者安装到不同的目录中，请单击 **定制** 而不是 **典型**。

IBM MQ 已安装，安装程序将启动 Prepare IBM MQ Wizard。

IBM MQ for Windows 安装将配置域本地组 mqm 和域组 Domain mqm。 它使 Domain mqm 成为 mqm 的成员。 同一域中的后续域控制器共享 mqm 和 Domain mqm 组。

2. 在 *earth* 和 *sun* 上，运行 Prepare IBM MQ Wizard。

有关更多信息，请参阅 使用 Prepare IBM MQ Wizard 配置 IBM MQ。

a) IBM MQ 安装程序会自动运行 Prepare IBM MQ Wizard 。

要手动启动向导，请在 **启动** > **所有程序** > **IBM MQ** 文件夹中找到 Prepare IBM MQ Wizard 的快捷方式。 选择与多安装配置中的 IBM MQ 安装相对应的快捷方式。

b) 单击 **下一步** ，并保留 **未单击** 以响应问题 “确定网络中是否存在 Windows 2000 或更高版本的域控制器”[1].

c) 在向导的最后一页中，根据需要选中或清除复选框，然后单击 **完成**。

Prepare IBM MQ Wizard 在第一个域控制器上创建域本地用户 MUSR_MQADMIN ，在第二个域控制器上创建另一个域本地用户 MUSR_MQADMIN1 。 向导将在每个控制器上创建 IBM MQ 服务，并以 MUSR_MQADMIN 或 MUSR_MQADMIN1 作为登录该服务的用户。

3. 定义有权创建队列管理器的用户。

用户必须有权在本地登录，并且是域本地 mqm 组的成员。 在域控制器上，域用户无权在本地登录，但管理员有权登录。 缺省情况下，没有用户同时具有这两个属性。 在此任务中，将域管理员添加到域本地 mqm 组。

a) 打开 **服务器管理器** > **角色** > **Active Directory 域服务** > ***wmq.example.com*** > **用户**。

b) 右键单击 **域管理员** > **添加到组 ...** 和类型 mqm ; 单击 **检查名称** > **确定** > **确定**

结果

1. 检查 Prepare IBM MQ Wizard 是否已创建域用户 MUSR_MQADMIN:
   a. 打开 **服务器管理器** > **角色** > **Active Directory 域服务** > ***wmq.example.com*** > **用户**。
   b. 右键单击 **MUSR_MQADMIN** > **属性 ...** > **成员**，并查看它是否是 Domain users 和 mqm 的成员。
2. 检查 MUSR_MQADMIN 是否有权作为服务运行:
   a. 单击单击 **启动** > **运行 ...**， 输入命令 **secpol.msc** ，然后单击 **确定**。
   b. 打开 **安全设置** > **本地策略** > **用户权限分配**。 在策略列表中，右键单击 **作为服务登录** > **属性** ，请参阅 MUSR_MQADMIN 列示为有权作为服务登录。 单击**确定**。

下一步做什么

1. 执行任务 第 445 页的『读取和写入本地 mqm 组授权的数据和日志文件』以验证安装和配置是否正常工作。
2. 返回到任务 第 432 页的『在 Windows 域控制器上创建多实例队列管理器』，以完成在域控制器上配置多实例队列管理器的任务。

[1] 您可以配置域的安装。 由于域控制器上的所有用户和组都具有域作用域，因此没有任何差别。 安装 IBM MQ 更简单，就像它不在域中一样。

相关任务

Windows 将第二个 Windows 域控制器添加到示例域

相关参考

IBM MQ Windows 服务所需的用户权限

Windows 在 *Windows* 上验证多实例队列管理器

使用样本程序 **amqsghac**， **amqsphac** 和 **amqsmhac** 来验证多实例队列管理器配置。 本主题提供了用于验证 Windows Server 2003 上的多实例队列管理器配置的示例配置。

高可用性样本程序使用自动客户机重新连接。 当连接的队列管理器发生故障时，客户机尝试重新连接到同一队列管理器组中的队列管理器。 样本的描述 高可用性样本程序演示了使用单个实例队列管理器进行客户机重新连接以实现简单性。 您可以将相同的样本与多实例队列管理器配合使用，以验证多实例队列管理器配置。

此示例使用 第 432 页的『在 Windows 域控制器上创建多实例队列管理器』中描述的多实例配置。 使用配置来验证多实例队列管理器是否切换到备用实例。 使用 **endmqm** 命令停止队列管理器，并使用 `-s`，转换和选项。 客户机程序重新连接到新的队列管理器实例，并在稍作延迟后继续处理新实例。

客户机安装在运行 Windows 7 Service Pack 1 的 400 MB VMware 映像中。 出于安全原因，它与运行多实例队列管理器的域服务器连接在同一 VMware 仅主机网络上。 它正在共享包含客户机连接表的 `/MQHA` 文件夹，以简化配置。

使用 IBM MQ Explorer 验证故障转移

在使用样本应用程序验证故障转移之前，请在每个服务器上运行 IBM MQ Explorer 。 使用 " **添加远程队列管理器> 直接连接到多实例队列管理器** " 向导将两个队列管理器实例添加到每个资源管理器。 确保两个实例都在运行，允许备用。 关闭运行带有活动实例的 VMware 映像的窗口，以虚拟方式关闭服务器电源，或者停止活动实例，从而允许切换到备用实例并重新连接客户机以重新连接。

注意: 如果关闭服务器电源，请确保它不是托管 `MQHA` 文件夹的服务器!

注: 允许切换到备用实例 选项可能在 " **停止队列管理器** " 对话框上不可用。 缺少该选项，因为队列管理器正在作为单个实例队列管理器运行。 您必须已在不使用 **允许备用实例** 选项的情况下启动该实例。 如果您停止队列管理器的请求被拒绝，请查看 " **详细信息** " 窗口，可能没有备用实例正在运行。

使用样本程序验证故障转移

选择要运行活动实例的服务器

您可能已选择其中一个服务器来托管 `MQHA` 目录或文件系统。 如果计划通过关闭运行活动服务器的 " VMware " 窗口来测试故障转移，请确保它不是托管 `MQHA` 的服务器!

在运行活动队列管理器实例的服务器上

1. 修改 *ipaddr1* 和 *ipaddr2* ，并将以下命令保存在 `N:\hasample.tst.` 中

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN)  TRPTYPE(TCP) +
MCAUSER(' ') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(' ipaddr1 (1414), ipaddr2 (1414)') QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

   注: 通过将 **MCAUSER** 参数留空，客户机用户标识将发送到服务器。 客户机用户标识必须具有对服务器的正确许可权。 替代方法是将 SVRCONN 通道中的 **MCAUSER** 参数设置为您在服务器上配置的用户标识。

2. 打开带有路径 `N:\` 的命令提示符并运行以下命令:

```
runmqsc -m QM1 < hasample.tst
```

3. 通过检查 **runmqsc** 命令的输出，验证侦听器是否正在运行并具有队列管理器控制。

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

或者，使用 TCPIP 侦听器正在运行且具有 `Control = Queue Manager` 的 IBM MQ Explorer 。

在客户机上

1. 将服务器上的共享目录 `C:\MQHA` 映射到客户机上的 `N:\` 。
2. 打开带有路径 `N:\`的命令提示符。 设置环境变量 `MQCHLLIB` 以指向服务器上的客户机通道定义表 (CCDT):

```
SET MQCHLLIB=N:\data\QM1\@ipcc
```

3. 在命令提示符处输入命令:

```
start amqsghac TARGET QM1
start amqsmhac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

注: 如果有问题，请在命令提示符处启动应用程序，以便在控制台上打印原因码，或者查看 `N:\data\QM1\errors` 文件夹中的 AMQERR01.LOG 文件。

在运行活动队列管理器实例的服务器上

1. 请完成下面任意一项任务:
   - 关闭运行带有活动服务器实例的 VMware 映像的窗口。
   - 使用 IBM MQ Explorer，停止活动队列管理器实例，允许切换到备用实例并指示可重新连接的客户机重新连接。
2. 这三个客户机最终检测到连接已断开，然后重新连接。 在此配置中，如果关闭服务器窗口，那么重新建立所有三个连接大约需要 7 分钟时间。 某些连接是在其他连接之前重新建立的。

结果

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start

17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsghac TARGET QM1
Sample AMQSGHAC start
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

Windows 保护 *Windows* 上的共享队列管理器数据和日志目录及文件

本主题描述如何使用全局备用安全组来保护队列管理器数据和日志文件的共享位置。 您可以在不同服务器上运行的队列管理器的不同实例之间共享位置。

通常，您不会为队列管理器数据和日志文件设置共享位置。 安装 IBM MQ for Windows 时，安装程序将为在该服务器上创建的任何队列管理器创建您选择的主目录。 它使用本地 mqm 组保护目录，并配置 IBM MQ 服务的用户标识以访问目录。

使用安全组保护共享文件夹时，允许访问该文件夹的用户必须具有该组的凭证。 假定使用名为 *mars* 的服务器上的本地 mqm 组来保护远程文件服务器上的文件夹。 使运行队列管理器进程的用户成为 *mars* 上本地 mqm 组的成员。 用户具有与远程文件服务器上文件夹的凭证匹配的凭证。 通过使用这些凭证，队列管理器能够访问其数据和文件夹中的日志文件。 在其他服务器上运行队列管理器进程的用户是不具有匹配凭证的其他本地 mqm 组的成员。 当队列管理器在与 *mars* 不同的服务器上运行时，它无法访问它在 *mars* 上运行时创建的数据和日志文件。 即使您使该用户成为域用户，它也具有不同的凭证，因为它必须从 *mars* 上的本地 mqm 组获取凭证，并且无法从其他服务器执行此操作。

为队列管理器提供全局备用安全组可解决此问题; 请参阅 第 442 页的图 76。 使用全局组保护远程文件夹。 在 *mars* 上创建全局组时，将该全局组的名称传递给队列管理器。 在 **crtmqm** 命令上使用 -a [r] 参数将全局组名作为备用安全组传递。 如果将队列管理器传输到其他服务器上运行，那么将随该队列管理器一起传输安全组的名称。 该名称在 qm.ini 文件的 **AccessMode** 节中作为 SecurityGroup 传输; 例如:

```
AccessMode:
SecurityGroup=wmq\wmq
```

qm.ini 中的 **AccessMode** 节还包含 RemoveMQMAccess ; 例如:

```
AccessMode:
RemoveMQMAccess=true|false
```

如果使用值 true 指定此属性，并且还提供了访问组，那么不会授予本地 mqm 组对队列管理器数据文件的访问权。

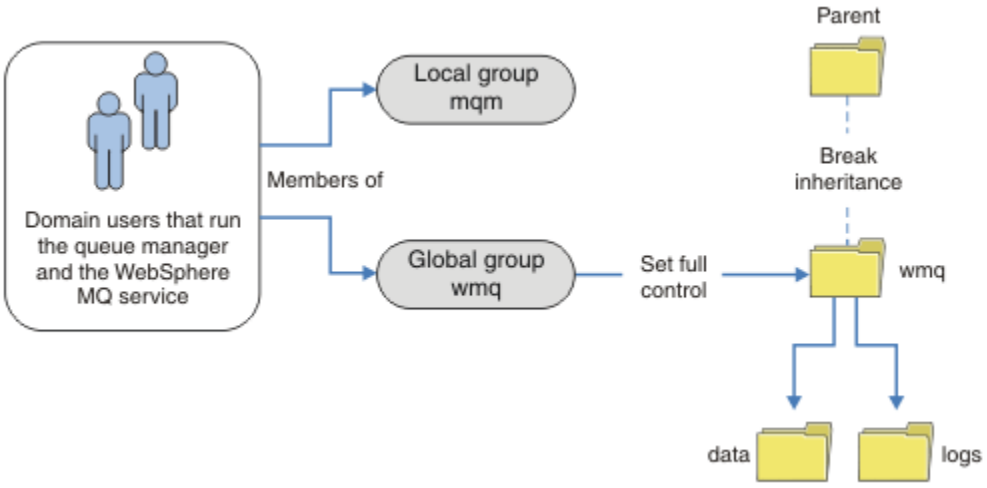


图 *76:* 使用备用全局安全组 *(1)* 保护队列管理器数据和日志

对于要与队列管理器进程一起运行以具有全局安全组的匹配凭证的用户标识，该用户标识还必须具有全局作用域。 不能使本地组或主体成为全局组的成员。 在 第 442 页的图 76 中，运行队列管理器进程的用户显示为域用户。

如果要部署许多 IBM MQ 服务器，那么 第 442 页的图 76 中的用户分组不方便。 您需要重复将用户添加到每个 IBM MQ 服务器的本地组的过程。 而是在域控制器上创建 `Domain mqm` 全局组，并使运行 IBM MQ 成员的用户成为 `Domain mqm` 组的成员; 请参阅 第 442 页的图 77。 将 IBM MQ 作为域安装安装时， Prepare IBM MQ Wizard 会自动使 `Domain mqm` 组成为本地 `mqm` 组的成员。 相同的用户同时位于全局组 `Domain mqm` 和 `wmq` 中。

提示: 相同的用户可以在不同的服务器上运行 IBM MQ，但在单个服务器上，您必须具有不同的用户才能将 IBM MQ 作为服务运行，并以交互方式运行。 对于服务器上的每个安装，您还必须具有不同的用户。 通常，`Domain mqm` 包含多个用户。

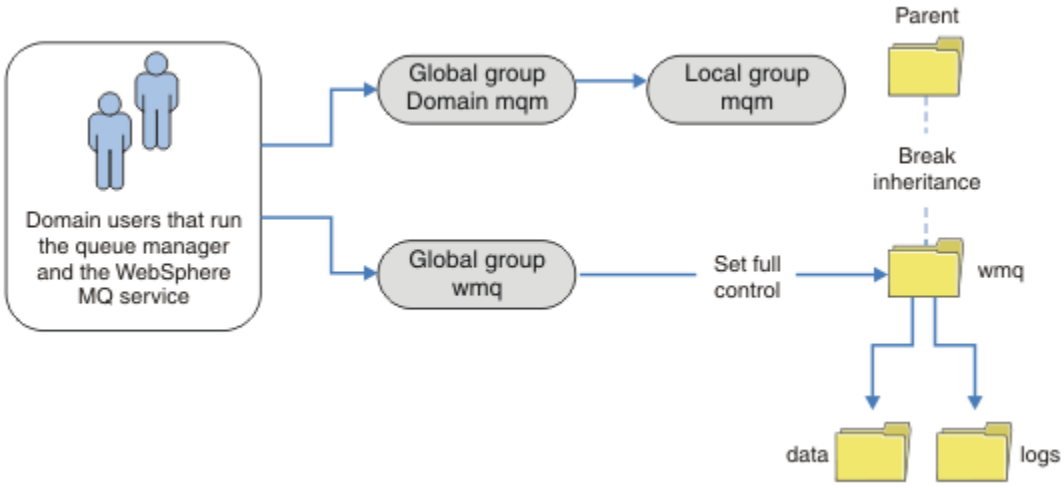


图 *77:* 使用备用全局安全组 *(2)* 保护队列管理器数据和日志

第 442 页的图 77 中的组织目前不必要地复杂。 该安排有两个具有相同成员的全局组。 您可以简化组织，并仅定义一个全局组; 请参阅 第 443 页的图 78。

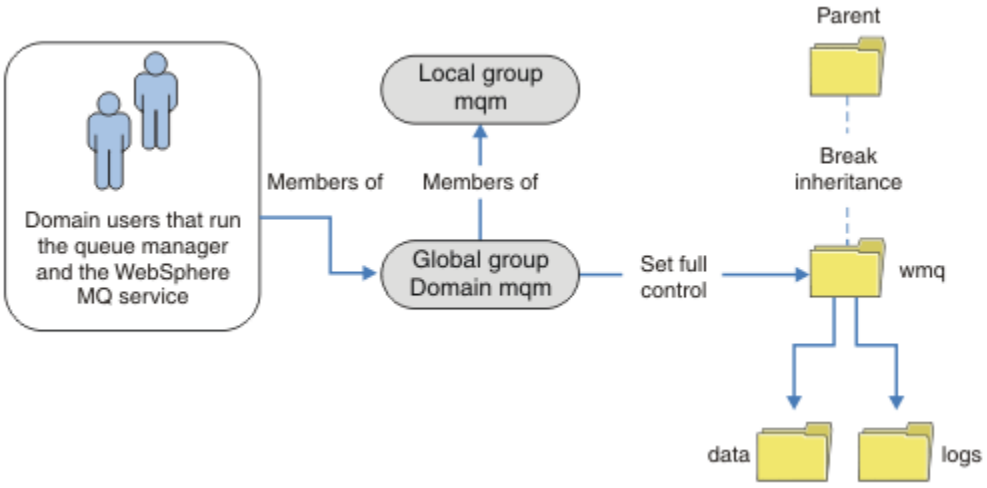


图 *78:* 使用备用全局安全组 *(3)* 保护队列管理器数据和日志

或者，您可能需要更精细的访问控制，将不同的队列管理器限制为能够访问不同的文件夹; 请参阅 第 444 页的图 79。 在 第 444 页的图 79 中，在不同的全局组中定义了两组域用户，以保护不同的队列管理器日志和数据文件。 将显示两个不同的本地 mqm 组，这些组必须位于不同的 IBM MQ 服务器上。 在此示例中，队列管理器分为两个集合，并将不同的用户分配给这两个集合。 这两个集合可能是测试队列管理器和生产队列管理器。 备用安全组称为 wmq1 和 wmq2。 您必须根据全局组 wmq1 和 wmq2 是在测试还是生产部门中，手动将它们添加到正确的队列管理器。 配置无法利用 IBM MQ 的安装将 Domain mqm 传播到本地 mqm 组 (如 第 443 页的图 78 中) ，因为有两组用户。

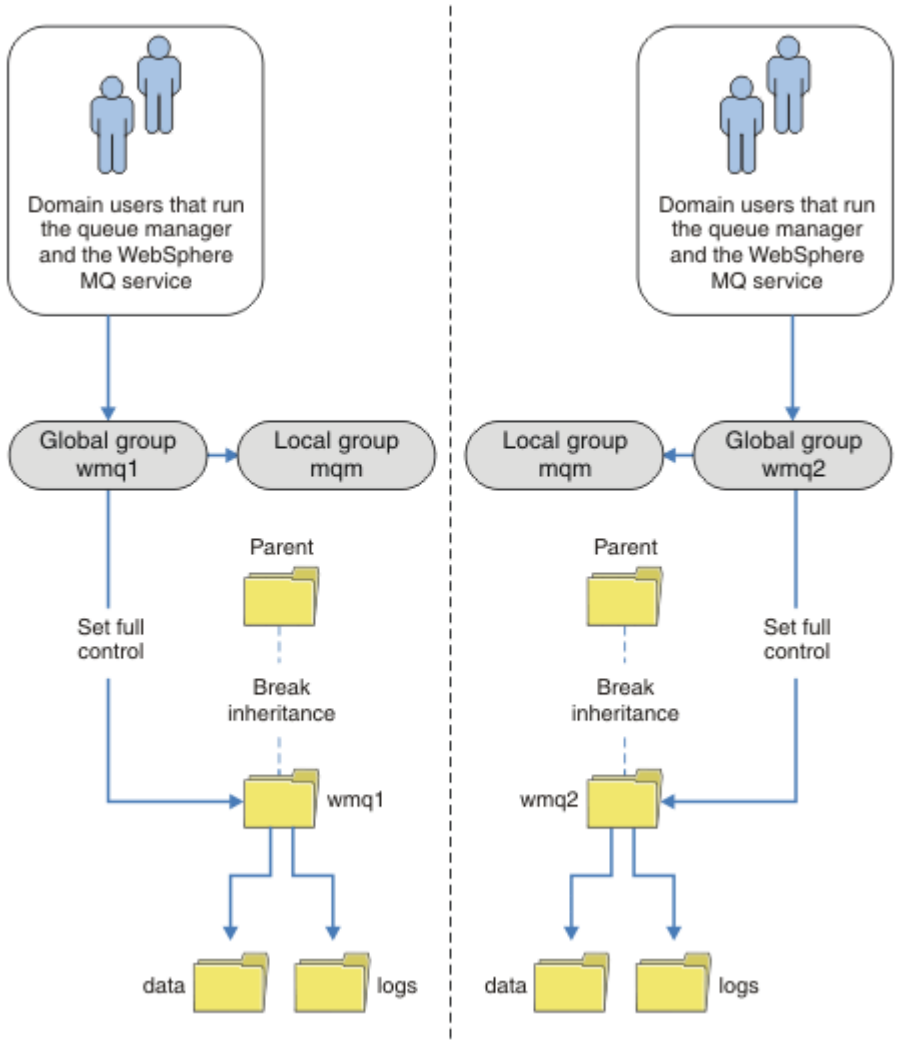


图 *79:* 使用备用全局安全性主体 *(4)* 保护队列管理器数据和日志

对两个部门进行分区的替代方法是将它们放置在两个窗口域中。 在这种情况下，您可以返回到使用 第 443 页的图 78 中显示的更简单的模型。

Windows 保护 *Windows* 上的非共享队列管理器数据和日志目录和文件

本主题描述如何使用本地 mqm 组和备用安全组来保护队列管理器数据和日志文件的备用位置。

通常，您不会为队列管理器数据和日志文件设置备用位置。 安装 IBM MQ for Windows 时，安装程序会为创建的任何队列管理器创建您选择的主目录。 它使用本地 mqm 组保护目录，并配置 IBM MQ 服务的用户标识以访问目录。

两个示例演示如何为 IBM MQ 配置访问控制。 这些示例显示了如何在安装所创建的数据和日志路径上未包含的目录中使用其数据和日志创建队列管理器。 在第一个示例 第 445 页的『读取和写入本地 mqm 组授权的数据和日志文件』中，通过本地 mqm 组授权来允许访问队列和日志目录。 第二个示例 第 448 页的『读取和写入备用本地安全组授权的数据和日志文件』的不同之处在于，对目录的访问权由备用安全组授权。 当仅在一台服务器上运行的队列管理器访问目录时，使用备用安全组保护数据和日志文件时，您可以选择保护具有不同本地组或主体的不同队列管理器。 当在不同服务器 (例如，使用多实例队列管理器) 上运行的队列管理器访问目录时，唯一的选择是使用备用安全组保护数据和日志文件; 请参阅 第 441 页的『保护 Windows 上的共享队列管理器数据和日志目录及文件』。

配置队列管理器数据和日志文件的安全许可权不是 Windows 上的常见任务。 安装 IBM MQ for Windows 时，请指定队列管理器数据和日志的目录，或者接受缺省目录。 安装程序使用本地 mqm 组自动保护这些目

录，并授予其完全控制许可权。 安装过程确保运行队列管理器的用户标识是本地 mqm 组的成员。 您可以修改目录上的其他访问许可权以满足您的访问需求。

如果将数据和日志文件目录移至新位置，那么必须配置新位置的安全性。 如果备份队列管理器并将其复原到其他计算机，或者如果将队列管理器更改为多实例队列管理器，那么可以更改目录的位置。 您可以选择两种方法来保护新位置中的队列管理器数据和日志目录。 您可以通过限制对本地 mqm 组的访问来保护目录，也可以限制对您选择的任何安全组的访问。

使用本地 mqm 组保护目录的步骤数最少。 设置对数据和日志目录的许可权，以允许本地 mqm 组完全控制。典型方法是复制现有许可权集，从而从父代中除去继承。 然后，您可以除去或限制其他主体的许可权。

如果使用与 "准备 IBM MQ " 向导设置的服务不同的用户标识运行队列管理器，那么该用户标识必须是本地 mqm 组的成员。 任务 第 445 页的『读取和写入本地 mqm 组授权的数据和日志文件』将指导您完成这些步骤。

您还可以使用备用安全组来保护队列管理器数据和日志文件。 使用备用安全组保护队列管理器数据和日志文件的过程包含许多引用 第 445 页的图 80 的步骤。 本地组 wmq 是备用安全组的示例。

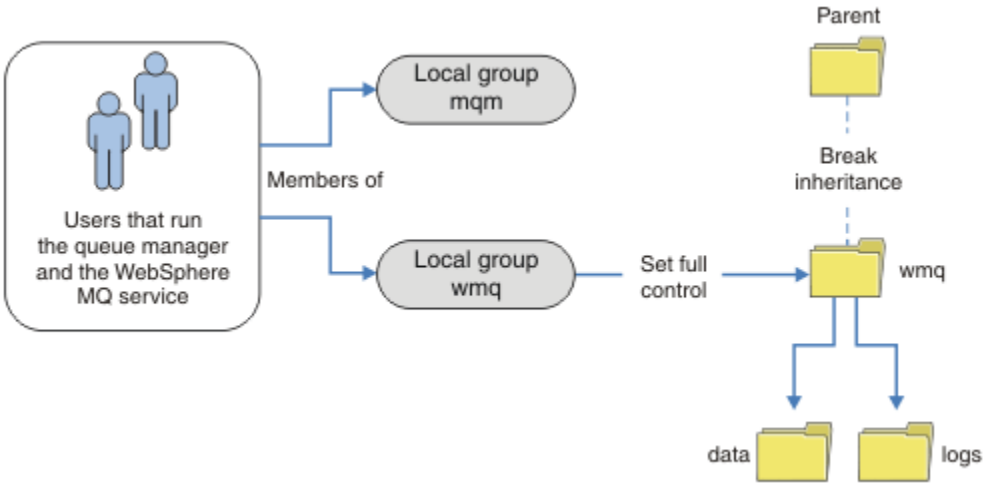


图 *80:* 使用备用本地安全组 *wmq* 保护队列管理器数据和日志

1. 为队列管理器数据和日志，公共目录或公共父目录创建单独的目录。
2. 复制目录或父目录的现有继承许可权集，并根据您的需要对其进行修改。
3. 通过授予备用组 wmq 对这些目录的完全控制许可权，保护要包含队列管理器和日志的目录。
4. 为所有运行队列管理器进程的用户标识提供备用安全组或主体的凭证:
   a. 如果将用户定义为备用安全主体，那么该用户必须与队列管理器运行时所使用的用户相同。 用户必须是本地 mqm 组的成员。
   b. 如果将本地组定义为备用安全组，请将队列管理器将在其下运行的用户添加到备用组。 用户还必须是本地 mqm 组的成员。
   c. 如果将全局组定义为备用安全组，请参阅 第 441 页的『保护 Windows 上的共享队列管理器数据和日志目录及文件』。
5. 使用 -a 参数在 **crtmqm** 命令上指定备用安全组或主体来创建队列管理器。

Windows 读取和写入本地 *mqm* 组授权的数据和日志文件

此任务说明如何创建队列管理器及其存储在您选择的任何目录中的数据和日志文件。 对文件的访问权由本地 mqm 组保护。 未共享目录。

开始之前

1. 安装 IBM MQ for Windows 作为主安装。

2. 运行 Prepare IBM MQ Wizard。

   有关更多信息，请参阅 使用 Prepare IBM MQ Wizard 配置 IBM MQ。

   对于此任务，请将安装配置为使用本地用户标识或域用户标识运行。 最终，要完成 第 418 页的『Windows 域和多实例队列管理器』中的所有任务，必须为域配置安装。
3. 使用管理员权限登录以执行任务的第一部分。

关于此任务

此任务是一组相关任务之一，用于说明如何访问队列管理器数据和日志文件。 这些任务显示如何创建有权读写存储在所选目录中的数据和日志文件的队列管理器。 它们与任务 第 418 页的『Windows 域和多实例队列管理器』一起执行。

在 Windows 上，您可以在选择的任何目录中为 IBM MQ for Windows 创建缺省数据和日志路径。 安装和配置向导会自动授予本地 mqm 组以及运行队列管理器进程的用户标识对这些目录的访问权。 如果创建队列管理器，并为队列管理器数据和日志文件指定不同的目录，那么必须配置对这些目录的完全控制许可权。

在此示例中，您通过授予本地 mqm 组对目录 *c:\wmq* 的许可权，为队列管理器提供对其数据和日志文件的完全控制。

crtmqm 命令创建在工作站使用 IBM MQ 服务启动时自动启动的队列管理器。

该任务具有说明性; 它使用您可以更改的特定值。 可以更改的值以斜体显示。 在任务结束时，遵循指示信息以除去您所做的所有更改。

过程

1. 打开命令提示符。
2. 输入以下命令:

   ```
   md c:\wmq\data, c:\wmq\logs
   ```
3. 设置对目录的许可权以允许本地 mqm 组读写访问权。

   ```
   cacls c:\wmq/T /E /G mqm:F
   ```

   系统响应:

   ```
   processed dir: c:\wmq
   processed dir: c:\wmq\data
   processed dir: c:\wmq\logs
   ```
4. 可选： 切换到作为本地 mqm 组成员的用户标识。

   您可以继续作为管理员，但对于现实的生产配置，请继续使用具有更多受限权限的用户标识。 用户标识必须至少是本地 mqm 组的成员。

   如果将 IBM MQ 安装配置为域的一部分，请使用户标识成为 Domain mqm 组的成员。 “准备 IBM MQ ”向导使 Domain mqm 全局组成为本地 mqm 组的成员，因此您不必使用户标识直接成为本地 mqm 组的成员。
5. 创建队列管理器。

   ```
   crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
   ```

   系统响应:

   ```
   IBM MQ queue manager created.
   Directory 'c:\wmq\data\QMGR' created.
   The queue manager is associated with installation '1'
   Creating or replacing default objects for queue manager 'QMGR'
   Default objects statistics : 74 created. 0 replaced.
   ```

```
Completing setup.
Setup completed.
```

6. 检查队列管理器创建的目录是否在 *c:\wmq* 目录中。

```
dir c:\wmq/D /B /S
```

7. 检查文件是否具有本地 mqm 组的读写或完全控制许可权。

```
cacls c:\wmq\*.*
```

下一步做什么

通过将消息放入队列并获取消息来测试队列管理器。

1. 启动队列管理器。

```
strmqm QMGR
```

系统响应：

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation '1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. 创建测试队列。

```
echo define qlocal(QTEST) | runmqsc QMGR
```

系统响应：

```
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QMGR.


1 : define qlocal(QTEST)
AMQ8006: IBM MQ queue created.
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

3. 使用样本程序 **amqsput** 放置测试消息。

```
echo 'A test message' | amqsput QTEST QMGR
```

系统响应：

```
Sample AMQSPUT0 start
target queue is QTEST
Sample AMQSPUT0 end
```

4. 使用样本程序 **amqsget** 获取测试消息。

```
amqsget QTEST QMGR
```

系统响应：

```
Sample AMQSGET0 start
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGET0 end
```

5. 停止队列管理器。

   ```
   endmqm -i QMGR
   ```

   系统响应：

   ```
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ended.
   ```

6. 删除队列管理器。

   ```
   dltmqm QMGR
   ```

   系统响应：

   ```
   IBM MQ queue manager 'QMGR' deleted.
   ```

7. 删除您创建的目录。

   提示： 将 /Q 选项添加到命令中，以防止命令提示删除每个文件或目录。

   ```
   del /F /S C:\wmq\*.*
   rmdir /S C:\wmq
   ```

相关概念

第 418 页的『Windows 域和多实例队列管理器』
Windows 上的多实例队列管理器要求共享其数据和日志。 共享必须可供在不同服务器或工作站上运行的队列管理器的所有实例访问。 配置队列管理器并作为 Windows 域的一部分进行共享。 队列管理器可以在域工作站或服务器上运行，也可以在域控制器上运行。

相关任务

Windows 读取和写入备用本地安全组授权的数据和日志文件

此任务显示如何在 **crtmqm** 命令上使用 -a 标志。 该标志为队列管理器提供了备用本地安全组，使其能够访问其日志和数据文件。

第 429 页的『读取和写入备用全局安全组授权的共享数据和日志文件』
第 419 页的『在 Windows 上的域工作站或服务器上创建多实例队列管理器』

Windows 读取和写入备用本地安全组授权的数据和日志文件
此任务显示如何在 **crtmqm** 命令上使用 -a 标志。 该标志为队列管理器提供了备用本地安全组，使其能够访问其日志和数据文件。

开始之前

1. 安装 IBM MQ for Windows 作为主安装。
2. 运行 Prepare IBM MQ Wizard。

   有关更多信息，请参阅 使用 Prepare IBM MQ Wizard 配置 IBM MQ。

   对于此任务，请将安装配置为使用本地用户标识或域用户标识运行。 最终，要完成 第 418 页的『Windows 域和多实例队列管理器』中的所有任务，必须为域配置安装。
3. 使用管理员权限登录以执行任务的第一部分。

关于此任务

此任务是一组相关任务之一，用于说明如何访问队列管理器数据和日志文件。 这些任务显示如何创建有权读写存储在所选目录中的数据和日志文件的队列管理器。 它们与任务 第 418 页的『Windows 域和多实例队列管理器』一起执行。

在 Windows 上，您可以在选择的任何目录中为 IBM MQ for Windows 创建缺省数据和日志路径。 安装和配置向导会自动授予本地 mqm 组以及运行队列管理器进程的用户标识对这些目录的访问权。 如果创建队列管理器，并为队列管理器数据和日志文件指定不同的目录，那么必须配置对这些目录的完全控制许可权。

在此示例中，您为队列管理器提供了具有对目录的完全控制权限的备用安全本地组。 备用安全组授予队列管理器许可权以管理目录中的文件。 备用安全组的主要用途是授权备用安全全局组。 使用备用安全全局组来设置多实例队列管理器。 在此示例中，您将配置本地组以熟悉备用安全组的使用，而无需在域中安装 IBM MQ 。 将本地组配置为备用安全组是不寻常的。

crtmqm 命令创建在工作站使用 IBM MQ 服务启动时自动启动的队列管理器。

该任务具有说明性; 它使用您可以更改的特定值。 可以更改的值以斜体显示。 在任务结束时，遵循指示信息以除去您所做的所有更改。

过程

1. 设置备用安全组。

   备用安全组通常是域组。 在此示例中，您将创建使用本地备用安全组的队列管理器。 通过本地备用安全组，您可以使用不属于域的 IBM MQ 安装来执行该任务。

   a) 运行 **lusrmgr.msc** 命令以打开 "本地用户和组" 窗口。

   b) 右键单击 **组** > **新建组 ...**

   c) 在 **组名** 字段中，输入 *altmqm* ，然后单击 **创建** > **关闭**。

   d) 标识运行 IBM MQ 服务的用户标识。

      i) 单击 **启动** > **运行 ...**， 输入 services.msc ，然后单击 **确定**。

      ii) 单击服务列表中的 IBM MQ 服务，然后单击 "登录" 选项卡。

      iii) 请记住用户标识并关闭 "服务资源管理器"。

   e) 将运行 IBM MQ 服务的用户标识添加到 *altmqm* 组。 还请添加用于登录以创建队列管理器的用户标识，并以交互方式运行该用户标识。

      Windows 通过检查正在运行队列管理器进程的用户标识的权限，检查队列管理器访问数据和日志目录的权限。 用户标识必须是授权目录的 *altmqm* 组的成员 (直接或通过全局组间接)。
      如果将 IBM MQ 作为域的一部分安装，并且要执行 第 419 页的『在 Windows 上的域工作站或服务器上创建多实例队列管理器』中的任务，那么在 第 422 页的『在 Windows 上创建 Active Directory 和 DNS 域』 中创建的域用户标识为 *wmquser1* 和 *wmquser2*。
      如果未将队列管理器作为域的一部分进行安装，那么运行 IBM MQ 服务的缺省本地用户标识为 MUSR_MQADMIN。 如果您打算在没有管理员权限的情况下执行任务，请创建作为本地 mqm 组成员的用户。
      执行以下步骤以将 *wmquser1* 和 *wmquser2* 添加到 *altmqm*。 如果您的配置不同，请将您的名称替换为用户标识和组。

      i) 在组列表中，右键单击 **altmqm** > **属性** > **添加 ...**。

      ii) 在 "选择用户，计算机或组" 窗口中，输入 *wmquser1* ； *wmquser2* ，然后单击 **检查名称**。

      iii) 在 " Windows 安全性" 窗口中输入域管理员的名称和密码，然后单击 **确定** > **确定** > **应用** > **确定**。

2. 打开命令提示符。
3. 重新启动 IBM MQ 服务。

   您必须重新启动服务，以使其运行时使用的用户标识获取为其配置的其他安全凭证。

输入命令:

```
endmqsvc
strmqsvc
```

系统响应:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
The MQ service for installation 'Installation1' ended successfully.
```

并:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
The MQ service for installation 'Installation1' started successfully.
```

4. 输入以下命令:

```
md c:\wmq\data, c:\wmq\logs
```

5. 设置对目录的许可权以允许本地用户 *user* 读写访问权。

```
cacls c:\wmq/T /E /G altmqm:F
```

系统响应:

```
processed dir: c:\wmq
processed dir: c:\wmq\data
processed dir: c:\wmq\logs
```

6. 可选: 切换到作为本地 mqm 组成员的用户标识。

   您可以继续作为管理员，但对于现实的生产配置，请继续使用具有更多受限权限的用户标识。 用户标识必须至少是本地 mqm 组的成员。
   如果将 IBM MQ 安装配置为域的一部分，请使用户标识成为 Domain mqm 组的成员。“准备 IBM MQ ”向导使 Domain mqm 全局组成为本地 mqm 组的成员，因此您不必使用户标识直接成为本地 mqm 组的成员。

7. 创建队列管理器。

```
crtmqm -a altmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

系统响应:

```
IBM MQ queue manager created.
Directory 'c:\wmq1\data\QMGR' created.
The queue manager is associated with installation '1'
Creating or replacing default objects for queue manager 'QMGR'
Default objects statistics : 74 created. 0 replaced.
Completing setup.
Setup completed.
```

8. 检查队列管理器创建的目录是否在 *c:\wmq* 目录中。

```
dir c:\wmq/D /B /S
```

9. 检查文件是否具有本地 mqm 组的读写或完全控制许可权。

```
cacls c:\wmq\*.*
```

下一步做什么

通过将消息放入队列并获取消息来测试队列管理器。

1. 启动队列管理器。

   ```
   strmqm QMGR
   ```

   系统响应：

   ```
   IBM MQ queue manager 'QMGR' starting.
   The queue manager is associated with installation '1'.
   5 log records accessed on queue manager 'QMGR' during the log
   replay phase.
   Log replay for queue manager 'QMGR' complete.
   Transaction manager state recovered for queue manager 'QMGR'.
   IBM MQ queue manager 'QMGR' started using V7.1.0.0.
   ```

2. 创建测试队列。

   ```
   echo define qlocal(QTEST) | runmqsc QMGR
   ```

   系统响应：

   ```
   5724-H72 (C) Copyright IBM Corp. 1994, 2025.  ALL RIGHTS RESERVED.
   Starting MQSC for queue manager QMGR.


   1 : define qlocal(QTEST)
   AMQ8006: IBM MQ queue created.
   One MQSC command read.
   No commands have a syntax error.
   All valid MQSC commands were processed.
   ```

3. 使用样本程序 **amqsput** 放置测试消息。

   ```
   echo 'A test message' | amqsput QTEST QMGR
   ```

   系统响应：

   ```
   Sample AMQSPUT0 start
   target queue is QTEST
   Sample AMQSPUT0 end
   ```

4. 使用样本程序 **amqsget** 获取测试消息。

   ```
   amqsget QTEST QMGR
   ```

   系统响应：

   ```
   Sample AMQSGET0 start
   message A test message
   Wait 15 seconds ...
   no more messages
   Sample AMQSGET0 end
   ```

5. 停止队列管理器。

```
endmqm -i QMGR
```

系统响应：

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended.
```

6. 删除队列管理器。

```
dltmqm QMGR
```

系统响应：

```
IBM MQ queue manager 'QMGR' deleted.
```

7. 删除您创建的目录。

提示： 将 /Q 选项添加到命令中，以防止命令提示删除每个文件或目录。

```
del /F /S C:\wmq\*.*
rmdir /S C:\wmq
```

相关任务

Windows 读取和写入本地 mqm 组授权的数据和日志文件
此任务说明如何创建队列管理器及其存储在您选择的任何目录中的数据和日志文件。 对文件的访问权由本地 mqm 组保护。 未共享目录。

Linux 在 *Linux* 上创建多实例队列管理器
在 Linux 上设置多实例队列管理器的示例。 此设置的规模较小，旨在演示所涉及的概念。 在其他 UNIX 平台上，步骤有所不同。

开始之前

本例基于 Red Hat Enterprise Linux 5 和 Windows 7。 IBM MQ 的当前版本需要这两个产品的更高版本。 然而，设置步骤与本文所述的大致相同。

关于此任务

此示例是在 3 GB RAM 正在运行 Windows 7 Service Pack 1 的 2 GHz 笔记本电脑上设置的。 两个 VMware 虚拟机（ Server1 和 Server2 ）以 640 MB 的镜像运行 Red Hat Enterprise Linux 5。 Server1 托管网络文件系统（ ）、队列管理器日志和 HA 实例。 NFS 通常情况下， NFS 服务器不会同时托管队列管理器实例；这样做是为了简化示例。 Server2 mounts 的队列管理器使用备用实例记录。 Server1 一个 IBM MQ MQI 客户端安装在额外的 400 MB VMware 镜像上，该镜像运行 Windows 7 Service Pack 1 并运行示例高可用性应用程序。 为了确保安全，所有虚拟机都被配置成只包含 VMware 主机的网络的组成部分。

注： 应仅将队列管理器数据放在 NFS 服务器上。 在 NFS 上，将以下三个选项与 mount 命令配合使用以使系统安全：

- **诺埃克**
  通过使用此选项，可以阻止二进制文件在 NFS 上运行，这将阻止远程用户在系统上运行不需要的代码。
- **诺苏伊德**
  通过使用此选项，可防止使用 set-user-identifier 和 set-group-identifier 位，这将阻止远程用户获取更高的特权。
- **节点 v**
  通过使用此选项，可以停止字符并阻止使用或定义特殊设备，这将阻止远程用户走出 chroot 监狱。

过程

1. 以 root 用户身份登录。
2. 请阅读 安装 IBM MQ -概述 ，并遵循相应的链接来安装 IBM MQ，创建 mqm 用户和组，然后定义 /var/mqm。
3. 完成任务 验证共享文件系统行为 以检查文件系统是否支持多实例队列管理器。
4. 对于 Server1，请完成以下步骤:
   a. 在将要共享的公共文件夹 /MQHA 中创建日志和数据目录。 例如:
      i) **mkdir** */MQHA*
      ii) **mkdir** */MQHA/logs*
      iii) **mkdir** */MQHA/qmgrs*
5. 对于 Server2，请完成以下步骤:
   a. 创建用于安装共享文件系统的文件夹 /MQHA。 保持路径与 Server1 上的路径相同。 例如:
      i) **mkdir** */MQHA*
6. 确保 MQHA 目录由用户和组 mqm 拥有，并且用户和组的访问许可权设置为 rwx 。 例如， **ls -al** 显示 drwxrwxr-x mqm mqm 4096 Nov 27 14:38 MQDATA 。
   a. **chown -R** mqm:mqm */MQHA*
   b. **chmod -R** ug+rwx */MQHA*
7. 通过输入以下命令来创建队列管理器: **crtmqm -ld** */MQHA/logs* **-md** */MQHA/qmgrs QM1*
8. 添加 [2]/MQHA *(rw,sync,no_wdelay,fsid=0) 到 /etc/exports
9. 对于 Server1，请完成以下步骤:
   a. 启动 NFS 守护程序: */etc/init.d/* **nfs** start
   b. 从 Server1:

```
dspmqinf -o command QM1
```

   并将结果复制到剪贴板:

```
addmqinf -s QueueManager
-v Name=QM1
-v Directory=QM1
-v Prefix=/var/mqm
-v DataPath=/MQHA/qmgrs/QM1
```

10. 对于 Server2，请完成以下步骤:
    a. 通过输入以下命令来安装导出的文件系统 /MQHA : **mount -t** nfs4 **-o** hard,intr *Server1:/ /MQHA*
    b. 将队列管理器配置命令粘贴到 Server2:

```
addmqinf -s QueueManager
-v Name=QM1
-v Directory=QM1
-v Prefix=/var/mqm
-v DataPath=/MQHA/qmgrs/QM1
```

11. 使用**-x** 参数 **strmqm -x** QM1 按任一顺序启动队列管理器实例。

    必须从与 **addmqinf** 命令相同的 IBM MQ 安装发出用于启动队列管理器实例的命令。 要从其他安装启动和停止队列管理器，必须首先使用 **setmqm** 命令设置与队列管理器相关联的安装。 有关更多信息，请参阅 setmqm。

[2] '*' 允许所有可以访问此安装 /MQHA 的机器进行读/写。 限制对生产机器的访问。

Linux 在 *Linux* 上验证多实例队列管理器

使用样本程序 **amqsghac**， **amqsphac** 和 **amqsmhac** 来验证多实例队列管理器配置。 本主题提供了用于验证 Linux Red Hat Enterprise 5 上的多实例队列管理器配置的示例配置。

高可用性样本程序使用自动客户机重新连接。 当连接的队列管理器发生故障时，客户机尝试重新连接到同一队列管理器组中的队列管理器。 样本的描述 高可用性样本程序演示了使用单个实例队列管理器进行客户机重新连接以实现简单性。 您可以将相同的样本与多实例队列管理器配合使用，以验证多实例队列管理器配置。

此示例使用 第 452 页的『在 Linux 上创建多实例队列管理器』中描述的多实例配置。 使用配置来验证多实例队列管理器是否切换到备用实例。 使用 **endmqm** 命令停止队列管理器，并使用 `-s`，转换和选项。 客户机程序重新连接到新的队列管理器实例，并在稍作延迟后继续处理新实例。

在此示例中，客户机正在 Windows 7 Service Pack 1 系统上运行。 系统正在托管两个正在运行多实例队列管理器的 VMware Linux 服务器。

使用 IBM MQ Explorer 验证故障转移

在使用样本应用程序验证故障转移之前，请在每个服务器上运行 IBM MQ Explorer 。 使用 " **添加远程队列管理器> 直接连接到多实例队列管理器** " 向导将两个队列管理器实例添加到每个资源管理器。 确保两个实例都在运行，允许备用。 关闭运行带有活动实例的 VMware 映像的窗口，以虚拟方式关闭服务器电源，或者停止活动实例，从而允许切换到备用实例。

注: 如果关闭服务器电源，请确保它不是托管 `/MQHA` 的服务器!

注: 允许切换到备用实例 选项可能在 " **停止队列管理器** " 对话框上不可用。 缺少该选项，因为队列管理器正在作为单个实例队列管理器运行。 您必须已在不使用 **允许备用实例** 选项的情况下启动该实例。 如果您停止队列管理器的请求被拒绝，请查看 **详细信息** 窗口，这可能是因为没有备用实例在运行。

使用样本程序验证故障转移

选择要运行活动实例的服务器

您可能已选择其中一个服务器来托管 `MQHA` 目录或文件系统。 如果计划通过关闭运行活动服务器的 " VMware " 窗口来测试故障转移，请确保它不是托管 `MQHA` 的服务器!

在运行活动队列管理器实例的服务器上

注: 在将 `MCAUSER` 设置为 `mqm` 的情况下运行 `SVRCONN` 通道可方便地减少示例中的配置步骤数。 如果选择了另一个用户标识，并且您的系统设置与示例中使用的系统不同，那么您可能会迁到访问许可权问题。 请勿在公开的系统上使用 `mqm` 作为 `MCAUSER` ; 这可能会严重影响安全性。

1. 修改 *ipaddr1* 和 *ipaddr2* ，并将以下命令保存在 `/MQHA/hasamples.tst.` 中

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN)  TRPTYPE(TCP) +
MCAUSER('mqm') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(' ipaddr1 (1414), ipaddr2
(1414)') QMNAME(QM1) REPLACE
START  CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
START  LISTENER(LISTENER.TCP)
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

2. 打开带有路径 `/MQHA` 的终端窗口，然后运行以下命令:

```
runmqsc -m QM1 < hasamples.tst
```

3. 通过检查 **runmqsc** 命令的输出，验证侦听器是否正在运行并具有队列管理器控制。

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

或者，使用 TCPIP 侦听器正在运行且具有 `Control = Queue Manager` 的 IBM MQ Explorer 。

在客户机上

1. 将客户机连接表 AMQCLCHL.TAB 从服务器上的 /MQHA/qmgrs/QM1.000/@ipcc 复制到客户机上的 C:\ 。
2. 打开带有路径 C:\ 的命令提示符，并设置环境变量 MQCHLLIB 以指向客户机通道定义表 (CCDT)

```
SET MQCHLLIB=C:\
```

3. 在命令提示符处输入命令:

```
start amqsghac TARGET QM1
start amqsmhac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

在运行活动队列管理器实例的服务器上

1. 请完成下面任意一项任务:
   - 关闭运行带有活动服务器实例的 VMware 映像的窗口。
   - 使用 IBM MQ Explorer，停止活动队列管理器实例，允许切换到备用实例并指示可重新连接的客户机重新连接。
2. 这三个客户机最终检测到连接已断开，然后重新连接。 在此配置中，如果关闭服务器窗口，那么重新建立所有三个连接大约需要 7 分钟时间。 某些连接是在其他连接之前重新建立的。

结果

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start

17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsghac TARGET QM1
Sample AMQSGHAC start
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

Multi 删除多实例队列管理器

在 Multiplatforms 版上，要完全删除多实例队列管理器，请使用 **dltmqm** 命令来删除该队列管理器，然后使用 **rmvmqinf** 或 **dltmqm** 命令从其他服务器中除去实例。

运行 **dltmqm** 命令以删除在定义了该队列管理器的任何服务器上的其他服务器上定义了实例的队列管理器。 您无需在创建 **dltmqm** 命令的服务器上运行该命令。 然后，在具有队列管理器定义的所有其他服务器上运行 **rmvmqinf** 或 **dltmqm** 命令。

您只能删除已停止的队列管理器。 删除队列管理器时，没有任何实例处于运行状态，严格而言，该队列管理器既不是单一实例队列管理器也不是多实例队列管理器；它只是在远程共享目录中有队列管理器数据和日志的队列管理器。 删除队列管理器时，将删除其队列管理器数据和日志，并且将从发出 **dltmqm** 命令的服务器上的 `mqs.ini` 文件中除去队列管理器节。 删除队列管理器时，您必须有权访问包含队列管理器数据和日志的网络共享目录。

在先前已创建队列管理器实例的其他服务器上，这些服务器上的 `mqs.ini` 文件中也存在条目。 您需要依次访问每个服务器，并通过运行命令 **rmvmqinf** 队列管理器节名称来除去队列管理器节。

Linux AIX 在 AIX and Linux 系统上，如果您在网络存储器中放置了公共 `mqs.ini` 文件，并通过在每个服务器上设置 `AMQ_MQS_INI_LOCATION` 环境变量从所有服务器中引用该文件，那么需要仅从其中一个服务器中删除队列管理器，因为只有一个 `mqs.ini` 文件需要更新。

示例

第一个服务器

dltmqm QM1

其他定义了实例的服务器

rmvmqinf QM1 ，或

dltmqm QM1

Multi 启动和停止多实例队列管理器

在多平台上作为单个实例或多实例队列管理器启动和停止配置的队列管理器。

在一对服务器上定义多实例队列管理器后，可以将该队列管理器作为单实例队列管理器或多实例队列管理器在任一服务器上运行。

要运行多实例队列管理器，请使用 **strmqm** -x *QM1* 命令在其中一个服务器上启动队列管理器; -x 选项允许实例进行故障转移。 它将成为 活动实例。 使用相同的 **strmqm** -x *QM1* 命令在另一服务器上启动备用实例; -x 选项允许该实例作为备用实例启动。

队列管理器现在正在运行，其中一个活动实例正在处理所有请求，另一个备用实例已准备好在活动实例失败时接管。 将授予活动实例对队列管理器数据和日志的独占访问权。 备用数据库等待被授予对队列管理器数据和日志的独占访问权。 当备用数据库被授予独占访问权时，它将成为活动实例。

您还可以通过在活动实例上发出 **endmqm** -s 命令来手动将控制切换到备用实例。 **endmqm** -s 命令在不关闭备用数据库的情况下关闭活动实例。 将释放队列管理器数据和日志上的互斥访问锁定，备用数据库将接管。

您还可以将在不同服务器上配置了多个实例的队列管理器作为单个实例队列管理器来启动和停止。 如果在不使用 **strmqm** 命令上的 -x 选项的情况下启动队列管理器，那么将阻止其他机器上配置的队列管理器实例作为备用实例启动。 如果尝试启动另一个实例，那么您将收到不允许队列管理器实例作为备用实例运行的响应。

如果使用不带 -s 选项的 **endmqm** 命令停止多实例队列管理器的活动实例，那么活动实例和备用实例都将停止。 如果使用带有 -x 选项的 **endmqm** 命令停止备用实例，那么它将停止为备用实例，并且活动实例将继续运行。 如果没有备用数据库上的 -x 选项，那么无法发出 **endmqm** 。

只有两个队列管理器实例可以同时运行; 一个是活动实例，另一个是备用实例。 如果同时启动两个实例，那么 IBM MQ 无法控制哪个实例成为活动实例; 它由网络文件系统确定。 获取对队列管理器数据的独占访问权的第一个实例将成为活动实例。

注: 在重新启动失败的队列管理器之前，必须断开应用程序与该队列管理器实例的连接。 如果不执行此操作，那么队列管理器可能无法正确重新启动。

Multi 共享文件 系统

在 Mulitplatform 上，多实例队列管理器使用联网文件系统来管理队列管理器实例。

多实例队列管理器使用文件系统锁定与共享队列管理器数据和日志的组合来自动执行故障转移。 只有一个队列管理器实例可以独占访问共享队列管理器数据和日志。 当它获得访问权时，它将成为活动实例。 未成功获取独占访问权的其他实例将作为备用实例等待，直到队列管理器数据和日志变为可用为止。

联网文件系统负责释放它对活动队列管理器实例持有的锁定。 如果活动实例以某种方式发生故障，那么联网文件系统将释放它对活动实例持有的锁定。 一旦释放互斥锁定，等待锁定的备用队列管理器就会尝试获取该锁定。 如果成功，那么它将成为活动实例，并且具有对共享文件系统上的队列管理器数据和日志的独占访问权。 然后继续启动。

相关主题 规划文件系统支持 描述了如何设置和检查文件系统是否支持多实例队列管理器。

多实例队列管理器无法防止您在文件系统中发生故障。 有多种方法可以保护您的数据。

- 投资可靠的存储，例如冗余磁盘阵列 (RAID) ，并将其包含在具有网络弹性的网络文件系统中。
- 将 IBM MQ 线性日志备份到备用介质，如果主日志介质发生故障，请使用备用介质上的日志进行恢复。 您可以使用备份队列管理器来管理此进程。

Multi 多个队列管理器实例

多实例队列管理器具有弹性，因为它在发生故障后使用备用队列管理器实例来恢复队列管理器可用性。

复制队列管理器实例是提高队列管理器进程可用性的非常有效的方法。 使用简单的可用性模型，仅用于说明: 如果队列管理器的一个实例的可靠性为 99% (一年内，累积停机时间为 3.65 天) ，那么添加队列管理器的另一个实例会将可用性提高到 99.99% (一年内，累积停机时间约为 1 小时)。

这是一个太简单的模型，无法为您提供实际的可用性数字估计值。 要以现实方式对可用性进行建模，您需要收集故障之间的平均时间 (MTBF) 和平均修复时间 (MTTR) 的统计信息，以及故障与修复时间之间时间的概率分布。

术语 "多实例队列管理器" 是指共享队列管理器数据和日志的队列管理器的活动实例和备用实例的组合。 多实例队列管理器通过使一个队列管理器实例在一个服务器上处于活动状态，另一个队列管理器实例在另一个服务器上处于备用状态，准备在活动实例失败时自动接管来防止队列管理器进程失败。

Multi 故障转移或转换

备用队列管理器实例在请求 (转换) 时或在活动实例发生故障 (故障转移) 时从活动实例接管。

- 当备用实例启动以响应向活动队列管理器实例发出的 **endmqm** -s 命令时，将发生 转换 。 可以指定 **endmqm** parameters -c， -i 或 -p 来控制队列管理器的突然停止方式。

  注: 仅当备用队列管理器实例已启动时，才会进行转换。 **endmqm** -s 命令释放活动队列管理器锁定并允许转换: 它不会启动备用队列管理器实例。
- 由于实例似乎意外停止 (即，未发出 **endmqm** 命令) ，因此释放了对活动实例持有的队列管理器数据的锁定时，将发生 故障转移 。

当备用实例作为活动实例进行接管时，它会将一条消息写入队列管理器错误日志。

当队列管理器发生故障或切换时，可重新连接的客户机将自动重新连接。 您不需要在 **endmqm** 命令上包含 `-r` 标志来请求客户机重新连接。 IBM MQ classes for Java 不支持客户机自动重新连接。

如果发现即使已发生故障转移并且备用实例已变为活动状态，也无法重新启动发生故障的实例，请检查本地连接到发生故障的实例的应用程序是否已与发生故障的实例断开连接。

本地连接的应用程序必须结束失败的队列管理器实例或与该实例断开连接，才能重新启动失败的实例。 使用共享绑定 (这是缺省设置) 的任何本地连接的应用程序，这些应用程序将保持与失败实例的连接，以防止实例重新启动。

如果无法结束本地连接的应用程序，或确保它们在本地队列管理器实例失败时断开连接，请考虑使用隔离的绑定。 使用隔离绑定的本地连接的应用程序不会阻止重新启动本地队列管理器实例，即使它们未断开连接也是如此。

Multi 通道和客户机重新连接

在备用队列管理器实例处于活动状态后，通道和客户机重新连接是复原消息处理的重要部分。

多实例队列管理器实例安装在具有不同网络地址的服务器上。 您需要为 IBM MQ 通道和客户机配置所有队列管理器实例的连接信息。 当备用数据库接管时，客户机和通道将自动重新连接到位于新网络地址的新活动队列管理器实例。 IBM MQ classes for Java 不支持客户机自动重新连接。

该设计与 HA-CMP 等高可用性环境的工作方式不同。 HA-CMP 为集群提供虚拟 IP 地址，并将该地址传输到活动服务器。 IBM MQ 重新连接不会更改或重新路由 IP 地址。 它通过使用您在通道定义和客户机连接中定义的网络地址重新连接来工作。 作为管理员，您需要定义通道定义中的网络地址以及与任何多实例队列管理器的所有实例的客户机连接。 配置多实例队列管理器的网络地址的最佳方法取决于连接:

队列管理器通道
通道的 CONNAME 属性是连接名称的逗号分隔列表; 例如， `CONNAME('127.0.0.1(1234),192.0.2.0(4321)')`。 将按连接列表中指定的顺序尝试连接，直到成功建立连接为止。 如果没有成功的连接，那么通道会尝试重新连接。

集群通道

通常，无需其他配置即可使多实例队列管理器在集群中工作。

如果队列管理器连接到存储库队列管理器，那么存储库将发现队列管理器的网络地址。 它引用队列管理器中 CLUSRCVR 通道的 CONNAME 。 在 TCPIP 上，队列管理器会自动设置 CONNAME (如果省略) 或将其配置为空白。 备用实例接管时，其 IP 地址会将先前活动实例的 IP 地址替换为 CONNAME。

如果需要，可以使用队列管理器实例的网络地址列表手动配置 CONNAME 。

客户机连接
客户机连接可以使用连接列表或队列管理器组来选择备用连接。 需要对客户机进行编译以使用 IBM WebSphere MQ 7.0.1 客户机库或更好的客户机库运行。 它们必须至少连接到 IBM WebSphere MQ 7.0.1 队列管理器。

发生故障转移时，重新连接需要一些时间。 备用队列管理器必须完成其启动。 连接到失败队列管理器的客户机必须检测连接故障，并启动新的客户机连接。 如果新客户机连接选择新活动的备用队列管理器，那么客户机将重新连接到同一队列管理器。

如果客户机在重新连接期间处于 MQI 调用的中间，那么它必须允许在调用完成之前进行长时间等待。

如果在消息通道上的批处理传输期间发生故障，那么将回滚并重新启动批处理。

切换比故障转移更快，仅需要停止队列管理器的一个实例并启动另一个实例。 对于只有少量日志记录要重放的队列管理器，最多可能需要几秒钟的时间进行转换。 要估算故障转移所花费的时间，您需要添加检测到故障所花费的时间。 根据网络和文件系统的不同，检测最多需要 10 秒的时间，可能需要几分钟。

Multi 应用程序恢复

应用程序恢复是故障转移后应用程序处理的自动延续。 故障转移后的应用程序恢复需要仔细设计。 某些应用程序需要意识到已发生故障转移。

应用程序恢复的目标是使应用程序仅在短暂延迟的情况下继续处理。 在继续新处理之前，应用程序必须回退并重新提交在失败期间正在处理的工作单元。

应用程序恢复的问题是丢失 IBM MQ MQI client 与队列管理器之间共享并存储在队列管理器中的上下文。 IBM MQ MQI client 复原了大部分上下文，但上下文的某些部分无法可靠地复原。 以下部分描述了应用程序恢复的一些属性以及它们如何影响连接到多实例队列管理器的应用程序的恢复。

事务消息传递

从传递消息的角度来看，故障转移不会更改 IBM MQ 消息传递的持久属性。 如果消息是持久的，并且在工作单元中正确管理，那么在故障转移期间不会丢失消息。

从事务处理的角度来看，事务会在故障转移后回退或落实。

未落实的事务将回滚。 故障转移后，可重新连接的应用程序会接收到 MQRC_BACKED_OUT 原因码，以指示事务已失败。 然后需要再次重新启动事务。

已落实的事务是已达到两阶段落实的第二阶段的事务，或者是已开始 MQCMIT 的单阶段 (仅消息) 事务。

如果队列管理器是事务协调程序，并且 MQCMIT 在失败之前已开始其两阶段落实的第二阶段，那么事务将成功完成。 完成由队列管理器控制，并在队列管理器再次运行时继续。 在可重新连接的应用程序中， MQCMIT 调用正常完成。

在单阶段落实 (仅涉及消息) 中，已启动落实处理的事务一旦再次运行，就会在队列管理器的控制下正常完成。 在可重新连接的应用程序中， MQCMIT 正常完成。

可重新连接的客户机可以在队列管理器的控制下使用单相事务作为事务协调程序。 扩展事务客户机不支持重新连接。 如果在事务客户机连接时请求重新连接，那么连接将成功，但无法重新连接。 该连接的行为就像它不可重新连接一样。

应用程序重新启动或恢复

故障转移会中断应用程序。 发生故障后，应用程序可以从头开始重新启动，也可以在中断后恢复处理。 后者称为 自动客户机重新连接。 IBM MQ classes for Java 不支持客户机自动重新连接。

通过 IBM MQ MQI client 应用程序，您可以设置连接选项以自动重新连接客户机。 选项为 MQCNO_RECONNECT 或 MQCNO_RECONNECT_Q_MGR。 如果未设置任何选项，那么客户机不会尝试自动重新连接，并且队列管理器故障会将 MQRC_CONNECTION_BROKEN 返回到客户机。 您可以设计客户机以尝试通过发出新的 MQCONN 或 MQCONNX 调用来启动新连接。

必须重新启动服务器程序; 当队列管理器或服务器发生故障时，队列管理器无法在它们正在处理的位置自动重新连接这些程序。 当多实例队列管理器实例发生故障时，通常不会在备用队列管理器实例上重新启动 IBM MQ 服务器程序。

您可以通过两种方式自动执行 IBM MQ 服务器程序以在备用服务器上重新启动:

1. 将服务器应用程序打包为队列管理器服务。 当备用队列管理器重新启动时，会将其重新启动。
2. 编写您自己的故障转移逻辑，例如，由备用队列管理器实例在启动时写入的故障转移日志消息触发。 然后，应用程序实例需要在启动后调用 MQCONN 或 MQCONNX ， 以创建与队列管理器的连接。

检测故障转移

某些应用程序需要了解故障转移，而其他应用程序则不需要了解故障转移。 请考虑这两个示例。

1. 通过消息传递通道获取或接收消息的消息传递应用程序通常不需要通道另一端的队列管理器正在运行: 如果通道另一端的队列管理器在备用实例上重新启动，那么不太可能受影响。
2. IBM MQ MQI client 应用程序处理来自一个队列的持久消息输入，并将持久消息响应作为单个工作单元的一部分放到另一个队列上: 如果它通过重新启动工作单元在同步点内处理来自 MQPUT， MQGET 或 MQCMIT 的 MQRC_BACKED_OUT 原因码，那么不会丢失任何消息。 此外，应用程序不需要执行任何特殊处理来处理连接故障。

但是，在第二个示例中，假设应用程序正在浏览队列以使用 MQGET 选项 MQGMO_MSG_UNDER_CURSOR 来选择要处理的消息。 重新连接会重置浏览光标，并且 MQGET 调用不会返回正确的消息。 在此示例中，应用程序必须知道已发生故障转移。 此外，在对光标下的消息发出另一个 MQGET 之前，应用程序必须恢复浏览光标。

丢失浏览光标是重新连接后应用程序上下文如何更改的一个示例。 第 460 页的『自动重新连接的客户机的恢复』中记录了其他案例。

在故障转移之后，您有三种针对 IBM MQ MQI client 应用程序的备用设计模式。 其中只有一个不需要检测故障转移。

无重新连接

在此模式中，当连接中断时，应用程序将停止当前连接上的所有处理。 要使应用程序继续处理，它必须建立与队列管理器的新连接。 应用程序完全负责传输在新连接上继续处理所需的任何状态信息。 以这种方式编写在失去连接后与队列管理器重新连接的现有客户机应用程序。

在连接丢失后，客户机将从下一个 MQI 调用接收原因码，例如 MQRC_CONNECTION_BROKEN 或 MQRC_Q_MGR_NOT_AVAILABLE 。 应用程序必须废弃其所有 IBM MQ 状态信息 (例如队列句柄) ，并发出新的 MQCONN 或 MQCONNX 调用以建立新的连接，然后重新打开它需要处理的 IBM MQ 对象。

缺省 MQI 行为是在与队列管理器的连接丢失后，队列管理器连接句柄变为不可用。 缺省值相当于在 MQCONNX 上设置 MQCNO_RECONNECT_DISABLED 选项，以防止在故障转移后应用程序重新连接。

故障转移容错

编写应用程序，以使其不受故障转移影响。 有时，小心错误处理足以处理故障转移。

重新连接感知

向队列管理器注册 MQCBT_EVENT_HANDLER 事件处理程序。 当客户机开始尝试重新连接到服务器时，会将事件处理程序与 MQRC_RECONNECTING 一起发布，并在成功重新连接后将事件处理程序与 MQRC_RECONNECTED 一起发布。 然后，您可以运行例程以重新建立可预测的状态，以便客户机应用程序能够继续处理。

自动重新连接的客户机的恢复

故障转移是意外事件，要使自动重新连接的客户机按设计工作，重新连接的后果必须是可预测的。

将意外故障转变为可预测且可靠的恢复的主要因素是使用事务。

在上一部分中，给出了一个示例 第 459 页的『2』，它是使用本地事务来协调 MQGET 和 MQPUT 的 IBM MQ MQI client 。 客户机发出 MQCMIT 或 MQBACK 调用以响应 MQRC_BACKED_OUT 错误，然后重新提交回退的事务。 队列管理器故障导致事务回退，并且客户机应用程序的行为确保不会丢失任何事务和消息。

请注意，对于回调，如果回调使用者参数状态为 :MQCS_SUSPENDED_USER_ACTION ，那么可能需要恢复使用应用程序。

并非所有程序状态都作为事务的一部分进行管理，因此重新连接的后果变得更加难以理解。 您需要了解重新连接如何更改 IBM MQ MQI client 的状态，以便设计客户机应用程序以避免队列管理器故障转移。

您可能决定在没有任何特殊故障转移代码的情况下设计应用程序，使用与其他错误相同的逻辑处理重新连接错误。 或者，您可以选择识别重新连接需要特殊错误处理，并向 IBM MQ 注册事件处理程序以运行例程来处理故障转移。 例程可以处理重新连接处理本身，或者设置一个标志以向主程序线程指示当它恢复处理时需要执行恢复处理。

IBM MQ MQI client 环境了解故障转移本身，并在重新连接后恢复尽可能多的上下文，方法是在客户机中存储一些状态信息，并代表客户机应用程序发出其他 MQI 调用以恢复其 IBM MQ 状态。 例如，将复原在故障点打开的对象的句柄，并以相同名称打开临时动态队列。 但有一些变化是不可避免的，你需要你的设计来处理这些变化。 这些更改可分为五种类型:

1. 从 MQI 调用返回新的或先前未诊断的错误，直到应用程序恢复一致的新上下文状态为止。

   接收新错误的一个示例是在重新连接之前尝试在保存上下文后传递上下文时返回码 MQRC_CONTEXT_NOT_AVAILABLE 。 重新连接后无法复原上下文，因为未将安全上下文传递到未经授权的客户机程序。 这样做会让恶意应用程序获取安全上下文。

   通常，应用程序以精心设计的方式处理常见和可预测的错误，并将不常见的错误降级为通用错误处理程序。 错误处理程序可能会从 IBM MQ 断开连接并重新连接，甚至完全停止该程序。 要提高连续性，您可能需要以不同的方式处理一些错误。
2. 非持久消息可能会丢失。
3. 事务将回滚 (也可能暂挂异步使用者，请参阅前面的文本)。

4. 在同步点外部使用的 MQGET 或 MQPUT 调用可能会中断，可能丢失消息。
5. 由于 MQI 调用中的等待时间过长，导致计时引起错误。

以下部分中列出了有关丢失上下文的一些详细信息。

- 除非使用 NPMCLASS （HIGH）选项将非持久消息放入队列，并且队列管理器故障未中断在关闭时存储非持久消息的选项，否则将废弃非持久消息。
- 当连接中断时，非持久预订将丢失。 在重新连接时，将重新建立该连接。 请考虑使用持久预订。
- 将重新计算获取等待时间间隔; 如果超出其限制，那么将返回 MQRC_NO_MSG_AVAILABLE。 同样，将重新计算预订到期时间，以提供相同的总体到期时间。
- 浏览光标在队列中的位置丢失; 通常在第一条消息之前重新建立。
  - MQGET 用于指定 MQGMO_BROWSE_MSG_UNDER_CURSOR 或 MQGMO_MSG_UNDER_CURSOR 的调用失败，原因码为 MQRC_NO_MSG_AVAILABLE。
  - 为浏览而锁定的消息将解锁。
  - 浏览具有句柄作用域的已标记消息是未标记的，可以再次浏览。
  - 在大多数情况下，未对合作浏览标记的消息进行标记。
- 安全上下文已丢失。 尝试使用已保存的消息上下文 (例如，将消息与 MQPMO_PASS_ALL_CONTEXT 一起放置) 失败，并返回 MQRC_CONTEXT_NOT_AVAILABLE。
- 消息令牌丢失。 使用消息令牌的 MQGET 将返回原因码 MQRC_NO_MSG_AVAILABLE。

  注: *MsgId* 和 *CorrelId*(因为它们是消息的一部分) 在故障转移期间随消息一起保留，因此 MQGET 使用 MsgId 或 CorrelId 按预期工作。
- 未落实的事务中放在同步点下的队列上的消息不再可用。
- 按逻辑顺序或在消息组中处理消息会在重新连接后产生返回码 MQRC_RECONNECT_INCOMPATIBLE 。
- MQI 调用可能会返回 MQRC_RECONNECT_FAILED 而不是客户机今天通常接收到的更一般的 MQRC_CONNECTION_BROKEN 。
- 如果 IBM MQ MQI client 不知道消息是否已成功传递到队列管理器，那么在 MQPUT 调用外部同步点期间重新连接将返回 MQRC_CALL_INTERRUPTED 。 MQCMIT 期间的重新连接行为类似。
- 将返回 MQRC_CALL_INTERRUPTED -在成功重新连接后-如果 IBM MQ MQI client 未收到来自队列管理器的响应以指示成功或失败
  - 使用 MQPUT 调用在同步点外部传递持久消息。
  - 使用 MQPUT1 调用在同步点外部传递持久消息或具有缺省持久性的消息。
  - 使用 MQCMIT 调用落实事务。 只有在成功重新连接后，才会返回响应。
- 通道将作为新实例重新启动 (它们也可能是不同的通道) ，因此不会保留通道退出状态。
- 在恢复已打开临时动态队列的可重新连接客户机的过程中，将复原临时动态队列。 不会复原临时动态队列上的任何消息，但已打开队列或已记住队列名称的应用程序能够继续处理。

  如果队列正由创建该队列的应用程序以外的应用程序使用，那么可能无法快速复原该队列，无法在下次引用该队列时显示该队列。 例如，如果客户机创建临时动态队列作为应答队列，并且要通过通道将应答消息放在队列上，那么该队列可能无法及时恢复。 在这种情况下，通道通常会将应答消息放在死信队列上。

  如果可重新连接的客户机应用程序按名称打开临时动态队列 (因为另一个应用程序已创建临时动态队列) ，那么发生重新连接时， IBM MQ MQI client 无法重新创建临时动态队列，因为它没有要从中创建临时动态队列的模型。 在 MQI 中，只有一个应用程序可以按模型打开临时动态队列。 其他希望使用临时动态队列的应用程序必须使用 MQPUT1 或服务器绑定，或者能够在失败时重试重新连接。

  只能将非持久消息放入临时动态队列，并且在故障转移期间这些消息将丢失; 对于在重新连接期间使用 MQPUT1 将消息放入临时动态队列的情况，此丢失情况成立。 如果在 MQPUT1 期间发生故障转移，那么可能不会放入消息，尽管 MQPUT1 成功。 此问题的一个变通方法是使用永久动态队列。 任何服务器绑定应用程序都可以按名称打开临时动态队列，因为它不可重新连接。

Multi 数据恢复和高可用性

使用多实例队列管理器的高可用性解决方案必须包含在存储故障后恢复数据的机制。

多实例队列管理器会增加队列管理器进程的可用性，但不会增加队列管理器用于存储消息的其他组件 (例如文件系统) 以及其他信息的可用性。

使数据高度可用的一种方法是使用联网的弹性数据存储。 您可以使用联网文件系统和弹性数据存储来构建自己的解决方案，也可以购买集成解决方案。 如果要将弹性与灾难恢复相结合，那么可以进行异步磁盘复制 (允许数十或数百公里的磁盘复制)。

您可以配置将不同的 IBM MQ 目录映射到存储介质的方式，以最佳地使用该介质。 对于 多实例 队列管理器，两种类型的 IBM MQ 目录和文件之间存在重要区别。

必须在队列管理器实例之间共享的目录。
必须在队列管理器的不同实例之间共享的信息位于两个目录中: `qmgrs` 和 `logs` 目录。 这些目录必须位于共享网络文件系统上。 建议您使用提供持续高可用性和出色性能的存储介质，因为随着消息的创建和删除，数据会不断变化。

不 具有 要在队列管理器实例之间共享的目录和文件。
某些其他目录不必在队列管理器的不同实例之间共享，而是通过使用镜像文件系统以外的方法快速复原。

- IBM MQ 可执行文件和 tools 目录。 通过重新安装或从备份文件归档进行备份和复原来进行替换。
- 为整个安装修改的配置信息。 配置信息由 IBM MQ(例如 AIX, Linux, and Windows 系统上的 `mqs.ini` 文件) 管理，或者由您自己的配置管理 (例如 **MQSC** 配置脚本) 进行部分管理。 使用文件归档进行备份和复原。
- 安装范围的输出，例如跟踪，错误日志和 FFDC 文件。 这些文件存储在缺省数据目录中的 `errors` 和 `trace` 子目录中。 AIX and Linux 系统上的缺省数据目录为 `/var/mqm`。 在 Windows 上，缺省数据目录是 IBM MQ 安装目录。

您还可以使用备份队列管理器，使用线性日志记录对多实例队列管理器进行常规介质备份。 备份队列管理器不会提供与镜像文件系统一样快的恢复，并且不会恢复自上次备份以来的更改。 与在发生本地化存储器故障后恢复队列管理器相比，备份队列管理器机制更适合在非现场灾难恢复场景中使用。

组合 IBM MQ 可用性解决方案

应用程序正在使用其他 IBM MQ 功能来提高可用性。 多实例队列管理器补充了其他高可用性功能。

IBM MQ 集群提高队列可用性

您可以通过创建集群队列的多个定义来提高队列可用性; 最多可以在集群中的每个管理器上创建一个队列。

假设集群成员失败，然后将新消息发送到集群队列。 除非消息 有 要转至失败的队列管理器，否则会将消息发送至集群中具有队列定义的另一个正在运行的队列管理器。

虽然集群极大地提高了可用性，但有两种相关的故障场景会导致消息延迟。 使用多实例队列管理器构建集群可降低消息延迟的可能性。

已 Marooned 消息
如果集群中的队列管理器发生故障，那么不会将更多可路由到集群中其他队列管理器的消息路由到发生故障的队列管理器。 在重新启动失败的队列管理器之前，将对已发送的消息进行序列化。

亲缘关系

亲缘关系是用于描述在两个其他独立的计算之间共享的信息的术语。 例如，向服务器发送请求消息的应用程序与期望处理应答的同一应用程序之间存在亲缘关系。 另一个示例将是消息序列，即根据先前的消息处理每条消息。

如果将消息发送到集群队列，那么需要考虑亲缘关系。 是否需要将连续消息发送到同一队列管理器，或者每条消息是否可以转至集群的任何成员?

如果确实需要将消息发送到集群中的同一队列管理器并且失败，那么消息将在发送方的传输队列中等待，直到失败的集群队列管理器再次运行为止。

如果为集群配置了多实例队列管理器，那么等待失败的队列管理器重新启动的延迟仅限于备用数据库接管时的一分钟左右的顺序。 当备用数据库运行时，将继续处理已暂停的消息，将启动到新激活的队列管理器实例的通道，并且在传输队列中等待的消息将开始流动。

配置集群以克服失败队列管理器延迟的消息的一种可能方法是将两个不同的队列管理器部署到集群中的每个服务器，并安排一个队列管理器为活动队列管理器，一个队列管理器为不同队列管理器的备用实例。 这是主动/备用配置，它会增加集群的可用性。

除了具有减少管理和提高可伸缩性的优点外，集群还继续提供额外的可用性元素，以补充多实例队列管理器。 集群可防止其他类型的故障影响队列管理器的活动实例和备用实例。

不间断服务
集群提供不间断服务。 集群接收到的新消息将发送到活动队列管理器以进行处理。 请勿依赖多实例队列管理器来提供不间断服务，因为备用队列管理器需要时间来检测故障并完成其启动，使其通道重新连接，以及使失败的消息批处理重新提交。

本地化中断
活动服务器，备用服务器和文件系统服务器之间的距离存在实际限制，因为它们需要以毫秒速度进行交互以提供可接受的性能。

集群队列管理器需要数秒的交互速度，并且可以在地理上分散在世界任何地方。

操作错误
通过使用两种不同的机制来提高可用性，可降低操作错误 (例如人为错误) 影响可用性工作的可能性。

队列共享组增加消息处理可用性

z/OS 仅在 z/OS 上提供的队列共享组允许一组队列管理器共享服务队列。 如果一个队列管理器发生故障，那么其他队列管理器将继续处理队列中的所有消息。 多实例队列管理器在 z/OS 上不受支持，并且仅作为更广泛的消息传递体系结构的一部分来补充队列共享组。

IBM MQ 客户机可提高应用程序可用性

IBM MQ MQI client 程序可以根据队列管理器可用性，连接权重和亲缘关系来连接到队列管理器组中的不同队列管理器。 通过在与运行队列管理器的机器不同的机器上运行应用程序，您可以提高解决方案的整体可用性，前提是存在重新连接应用程序的方法 (如果该应用程序所连接的队列管理器实例发生故障)。

队列管理器组用于提高客户机可用性，方法是将客户机与已停止的队列管理器解除耦合，并在一组队列管理器 (如 IP sprayer) 之间对客户机连接进行负载均衡。 客户机应用程序必须与失败的队列管理器没有亲缘关系 (例如，对特定队列的依赖关系) ，否则无法恢复处理。

自动客户机重新连接和多实例队列管理器通过解决一些亲缘关系问题来提高客户机可用性。 IBM MQ classes for Java 不支持客户机自动重新连接。

您可以设置 MQCNO 选项 MQCNO_RECONNECT_Q_MGR，以强制客户机重新连接到同一队列管理器:

1. 如果先前连接的单个实例队列管理器未在运行，那么将重试连接，直到队列管理器再次运行为止。
2. 如果将队列管理器配置为多实例队列管理器，那么客户机将重新连接到处于活动状态的任何实例。

通过自动重新连接到同一队列管理器，将恢复队列管理器代表客户机保存的许多状态信息，例如它打开的队列和它预订的主题。 如果客户机已打开动态应答队列以接收对请求的应答，那么也会复原与应答队列的连接。

MQ Adv. Linux RDQM 高可用性

RDQM (复制数据队列管理器) 是 Linux 平台上可用的高可用性解决方案。

RDQM 配置由在高可用性 (HA) 组中配置的三台服务器组成，每台服务器都具有一个队列管理器实例。 其中一个实例是运行中的队列管理器，可将其数据同步复制到另外两个实例。 如果运行此队列管理器的服务器发生故障，那么另一个队列管理器实例就会启动，并且具有可操作的最新数据。 队列管理器的三个实例可以选择共享浮动 IP 地址，因此只需要使用单个 IP 地址来配置客户机。 任何时候只能有一个队列管理器实例在运行，即使 HA 组因网络问题而分区也不例外。 运行队列管理器的服务器称为 "主" ，其他两个服务器各称为 "辅助"。

三个节点用于大大降低出现裂脑情况的可能性。 在双节点高可用性系统中，当两个节点之间的连接中断时，可能会发生裂脑。 在没有连接的情况下，两个节点可以同时运行队列管理器，从而累积不同的数据。 当连接恢复时，有两个不同版本的数据 ("分割-大脑") ，需要手动干预来决定保留哪个数据集，以及丢弃哪个数据集。

RDQM 使用具有定额的三节点系统来避免裂脑情况。 可与至少一个其他节点通信的节点构成定额。 队列管理器只能在具有定额的节点上运行。 队列管理器不能在未连接到至少一个其他节点的节点上运行，因此不能同时在两个节点上运行:

- 如果单个节点发生故障，那么队列管理器可以在其他两个节点中的一个节点上运行。 如果两个节点发生故障，那么队列管理器无法在剩余节点上运行，因为该节点没有定额 (剩余节点无法判断其他两个节点是否已发生故障，或者它们仍在运行，并且已失去连接)。
- 如果单个节点失去连接，那么队列管理器无法在此节点上运行，因为该节点没有定额。 队列管理器可以在其余两个节点中的一个节点上运行，这两个节点具有定额。 如果所有节点都失去连接，那么队列管理器无法在任何节点上运行，因为没有任何节点具有定额。

注: IBM MQ Console 不支持复制的数据队列管理器。 您可以将 IBM MQ Explorer 与复制的数据队列管理器配合使用，但这不会显示特定于 RDQM 功能部件的信息。

这三个节点的组配置由 Pacemaker 处理。 三个节点之间的复制由 DRBD 处理。 (请参阅 https://clusterlabs.org/pacemaker/ 以获取有关 Pacemaker 和 https://docs.linbit.com/docs/users-guide-9.0/ 的信息，以获取有关 DRBD 的信息。)

您可以使用 第 551 页的『备份队列管理器数据』中描述的过程来备份复制的数据队列管理器。 停止队列管理器并对其进行备份不会影响 RDQM 配置执行的节点监视。

下图显示了一个典型部署，其中 RDQM 在 HA 组中的三个节点中的每个节点上运行。

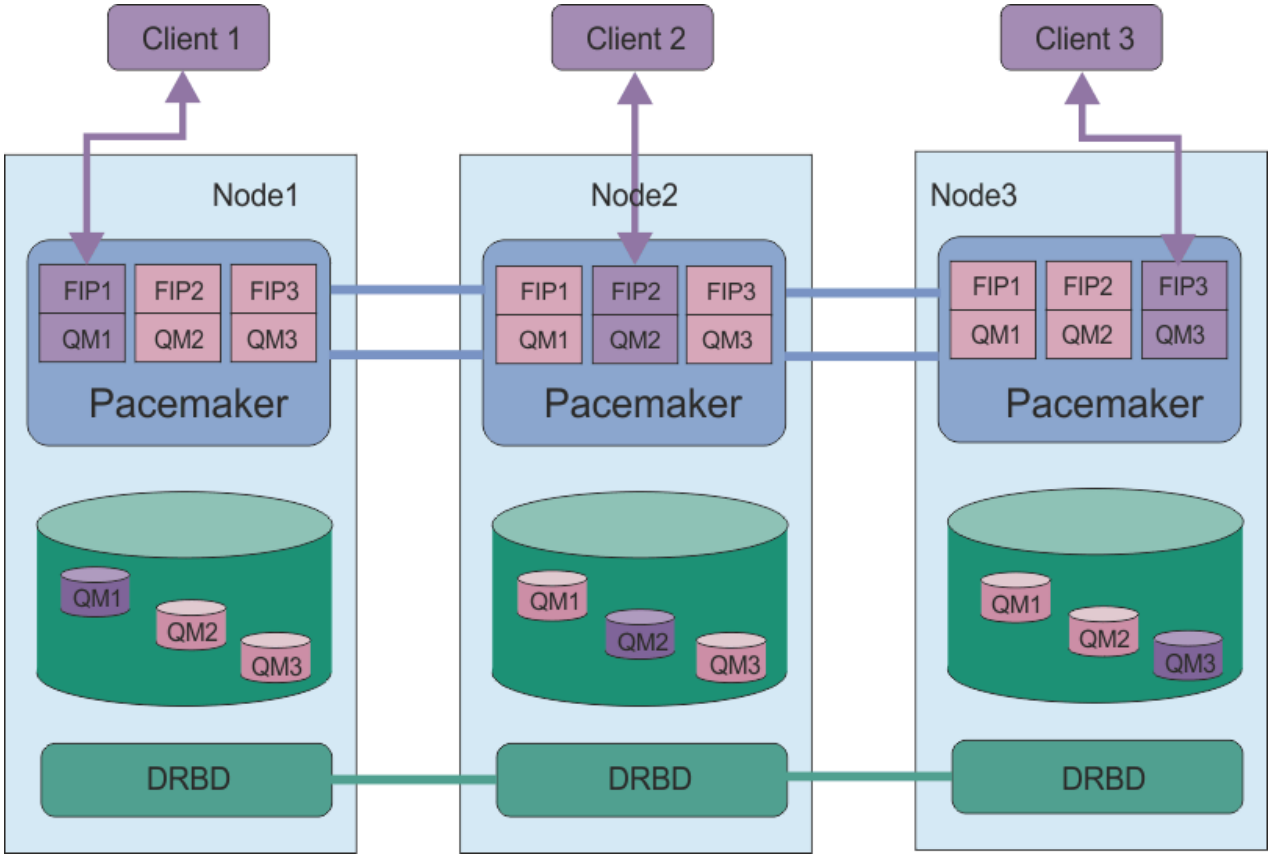


图 *81:* 具有三个 *RDQM* 的 *HA* 组的示例

在下一个图中， Node3 已失败， Pacemaker 链接已丢失，队列管理器 QM3 将改为在 Node2 上运行。

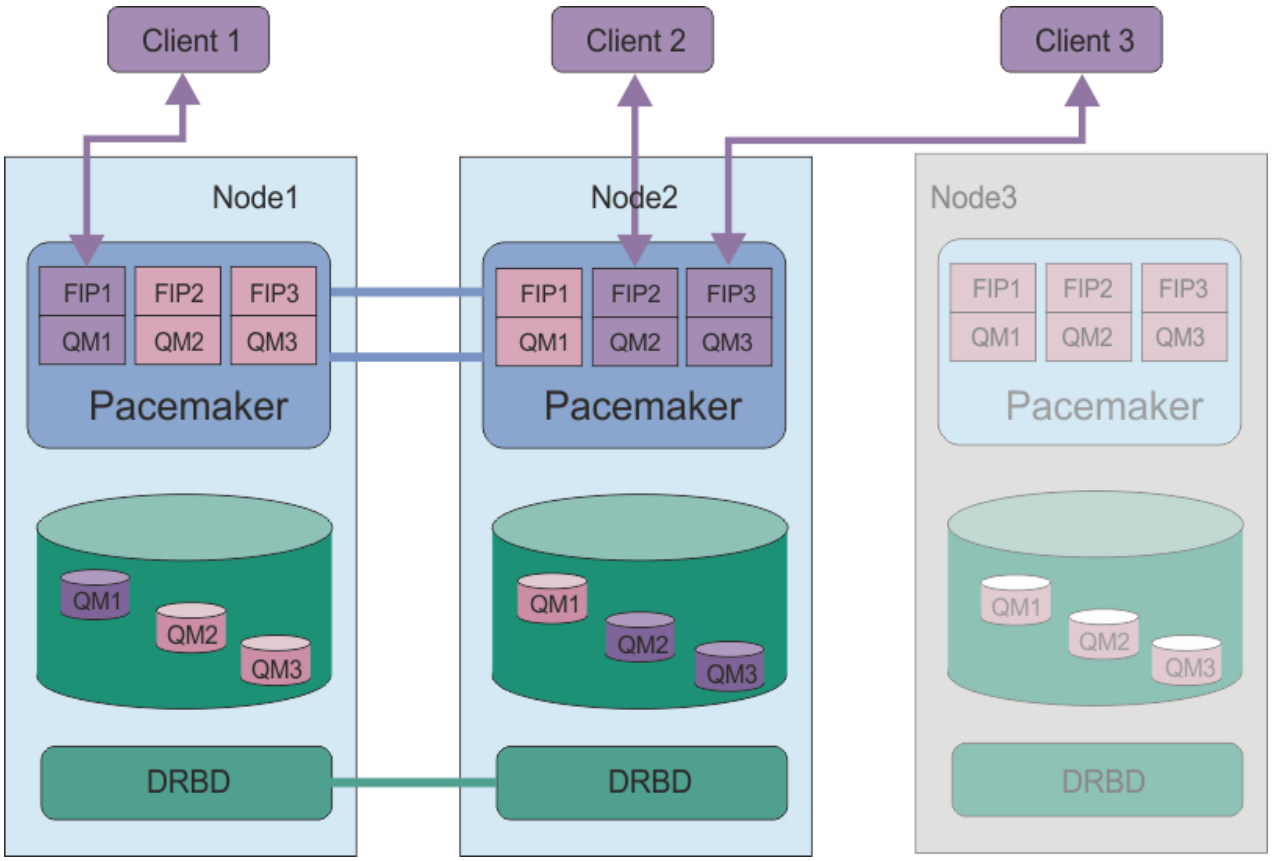


图 *82: node3* 失败后的示例

注: 当队列管理器故障转移到另一个节点时，它们将保留其在故障转移时所处的状态。 正在运行的队列管理器将启动，已停止的队列管理器将保持停止状态。

相关任务

安装 RDQM (复制的数据队列管理器)

应用 RDQM 的维护级别更新

迁移复制的数据队列管理器

对 RDQM 配置进行故障诊断

Linux *RDQM HA 解决方案的需求*

在配置 RDQM 高可用性 (HA) 组之前，必须满足许多需求。

系统需求

在配置 RDQM HA 组之前，必须在要包含在 HA 组中的三个服务器中的每个服务器上完成一些配置。

- 每个节点都需要一个名为 `drbdpool` 的卷组。 每个复制的数据队列管理器的存储器将作为此卷组中每个队列管理器的单独逻辑卷进行分配。 为了获得最佳性能，此卷组应由一个或多个对应于内部磁盘驱动器 (最好是 SSD) 的物理卷组成。 可以在安装 RDQM HA 解决方案之前或之后创建 `drbdpool` ，但必须先创建 `drbdpool` ，然后才能实际创建任何 RDQM。 使用 **vgs** 命令检查卷组配置。 输出应该类似于以下内容:

  ```
  VG       #PV #LV #SN Attr   VSize   VFree
  drbdpool   1   9   0 wz--n- <16.00g <7.00g
  rhel       1   2   0 wz--n- <15.00g     0
  ```

  特别是，请检查属性的第六列 (即 `wz--nc`) 中是否没有 `c` 字符。 `c` 指示已启用集群，如果已启用集群，那么必须删除卷组并在不进行集群的情况下重新创建该卷组。
- 创建 `drbdpool` 卷组后，不执行其他任何操作。 IBM MQ 管理在 `drbdpool` 中创建的逻辑卷以及安装这些逻辑卷的方式和位置。

- 每个节点最多需要三个用于配置 RDQM 支持的接口:
  - Pacemaker 用于监视 HA 组的主接口。
  - Pacemaker 用于监视 HA 组的备用接口。
  - 同步数据复制的接口，称为复制接口。 对于在 HA 组中运行的所有复制数据队列管理器的预期工作负载，这应该具有足够的带宽来支持复制需求。

  您可以配置 HA 组，以便将同一 IP 地址用于所有三个接口，将单独的 IP 地址用于每个接口，或者将同一 IP 地址用于主接口和备用接口，并将单独的 IP 地址用于复制接口。

  对于最大容错，这些接口应该是独立的网络接口卡 (NIC)。
- DRBD 要求 HA 组中的每个节点都具有有效的因特网主机名 (`uname -n` 返回的值) ，如 RFC 1123 所修订的 RFC 952 所定义。
- 如果 HA 组中的节点之间存在防火墙，那么防火墙必须允许一系列端口上的节点之间的流量。 提供了样本脚本 `/opt/mqm/samp/rdqm/firewalld/configure.sh`，用于在 RHEL 中运行标准防火墙时打开必需的端口。 必须以 `root` 用户身份运行脚本。 如果您正在使用其他一些防火墙，请检查服务定义 `/usr/lib/firewalld/services/rdqm*` 以查看需要打开哪些端口。 该脚本为 DRBD , Pacemaker 和 IBM MQ 添加以下永久 firewallD 服务规则:
  - MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-drbd.xml 允许 TCP 端口 7000-7100。
  - MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-pacemaker.xml 允许 UDP 端口 5404-5407
  - MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-mq.xml 允许 TCP 端口 1414 (如果需要其他端口，那么必须编辑脚本)
- 如果系统在强制方式下使用 SELinux ，那么必须运行 `semanage` 命令:

```
semanage permissive -a drbd_t
```

网络要求

建议您在同一数据中心内的 RDQM HA 组中找到三个节点。

如果选择在不同数据中心内查找节点，请注意以下限制:

- 随着数据中心之间等待时间的增加，性能会快速下降。 虽然 IBM 将支持最长 5 毫秒的等待时间，但您可能会发现应用程序性能无法容忍超过 1 到 2 毫秒的等待时间。
- 通过复制链路发送的数据不受使用 IBM MQ AMS 时可能存在的任何其他加密的限制。

(可选) 您可以配置浮动 IP 地址，以使客户机能够对复制的数据队列管理器 (RDQM) 使用相同的 IP 地址，而不考虑运行该数据队列管理器的 HA 组中的哪个节点。 浮动地址绑定到 RDQM 的主节点上的指定物理接口。 如果 RDQM 故障转移，并且另一个节点成为主节点，那么浮动 IP 将绑定到新主节点上同名的接口。 三个节点上的物理接口都必须具有相同的名称，并且属于与浮动 IP 地址相同的子网。

配置集群的用户需求

您可以用户 `root` 身份配置 RDQM HA 组。 如果不想配置为 `root`，请改为配置为 `mqm` 组中的用户。 对于 `mqm` 组中用于配置 RDQM 集群的用户，必须满足以下需求:

- `mqm` 用户必须能够使用 sudo 在组成 RDQM HA 组的三个服务器中的每一个服务器上运行命令。
- 如果 `mqm` 用户可以使用不带密码的 SSH 在组成 RDQM HA 组的三个服务器中的每一个服务器上运行命令，那么该用户只需在其中一个服务器上运行命令。
- `mqm` 用户在所有三个服务器上都必须具有相同的 UID。
- `mqm` 组在所有三个服务器上都必须具有相同的 GID。

您必须配置 sudo ， 以便 `mqm` 用户可以使用 root 用户权限运行以下命令:

```
/opt/mqm/bin/crtmqm
/opt/mqm/bin/dltmqm
```

```
/opt/mqm/bin/rdqmadm
/opt/mqm/bin/rdqmstatus
```

使用队列管理器的用户需求

要创建，删除或配置复制的数据队列管理器 (RDQM)，必须使用同时属于 mqm 和 haclient 组的用户标识 (haclient 组在 Pacemaker 安装期间创建)。

Linux 设置无密码 *SSH*

您可以设置无密码 SSH，这样只需在 HA 组中的一个节点上发出配置命令。

关于此任务

要设置无密码 SSH，必须在每个节点上配置 "mqmID，然后在每个节点上为该用户生成一个密钥。 然后将密钥分发给其他节点，并测试连接，将每个节点添加到已知主机列表中。 最后，您锁定了 "mqmID。

注: 该说明假定您定义的 HA 组具有独立的主接口、备用接口和复制接口，因此您在主接口和备用接口上定义了无密码 SSH 访问。 如果计划使用单个 IP 地址配置系统，则可通过该单个接口定义无密码 SSH 访问。

过程

1. 在三个节点上分别完成以下步骤，设置 "mqm 用户并生成 SSH 密钥:
   a) 将 "mqm 主目录更改为 "/home/mqm

   ```
   usermod -d /home/mqm mqm
   ```

   b) 创建 "/home/mqm 目录:

   ```
   mkhomedir_helper mqm
   ```

   c) 添加 "mqm 密码:

   ```
   passwd mqm
   ```

   d) 以 "mqm 的身份运行交互式 shell:

   ```
   su mqm
   ```

   e) 生成 "mqm 验证密钥:

   ```
   ssh-keygen -t rsa -f /home/mqm/.ssh/id_rsa -N ''
   ```

2. 在三个节点中的每一个节点上，完成以下步骤，将该节点的密钥添加到其他两个节点上，并测试每个节点的主地址和（如果使用）备用地址的连接:
   a) 将密钥添加到远程节点

   ```
   ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_primary_address
   ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_alternate_address
   ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_primary_address
   ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_alternate_address
   ```

   b) 检查无密码 ssh 并更新远程节点的已知主机:

   ```
   ssh remote_node1_primary_address uname -n
   ssh remote_node1_alternate_address uname -n
   ssh remote_node2_primary_address uname -n
   ssh remote_node2_alternate_address uname -n
   ```

   对于每个连接，系统都会提示您确认是否要继续。 逐一确认，更新已知主机。 在尝试使用无密码 SSH 配置 HA 组之前，必须完成此操作。

   c) 以 "mqm 退出交互式 shell:

   ```
   exit
   ```

3. 在每个节点上，以根用户身份完成以下步骤，删除 "mqm 密码并锁定 ID：

   a) 删除 "mqm 密码：

   ```
   passwd -d mqm
   ```

   b) 锁定 "mqm：

   ```
   passwd -l mqm
   ```

4. 在每个节点上，以根用户身份完成以下步骤，为 "mqm 用户设置 sudo 访问权限：

   a) Edit the sudoers file by using the **visudo** command:

   ```
   visudo
   ```

   b) 搜索"## Allows people in group wheel to run all commands"行，并在该行下方添加以下文本：

   ```
   #%mqm    ALL=(ALL)       ALL
   ```

   c) 搜索"## Same thing without a password"行，并在该行下方添加以下文本：

   ```
   %mqm            ALL=(ALL)       NOPASSWD: ALL
   ```

Linux 定义 *Pacemaker* 集群 *(HA* 组*)*

HA 组是 Pacemaker 集群。 通过编辑 /var/mqm/rdqm.ini 文件并运行 **rdqmadm** 命令来定义 Pacemaker 集群。

关于此任务

请参阅 https://clusterlabs.org/pacemaker/ 以获取有关 Pacemaker 的信息。 如果 mqm 用户可以使用 sudo，那么可以在 mqm 组中以用户身份创建 Pacemaker 集群。 如果用户还可以在没有密码的情况下通过 SSH 连接到每个服务器，那么您只需要编辑 rdqm.ini 文件并在其中一个服务器上运行 **rdqmadm** 以创建 Pacemaker 集群。 否则，必须在要成为节点的每个服务器上创建文件并以 root 身份运行命令。

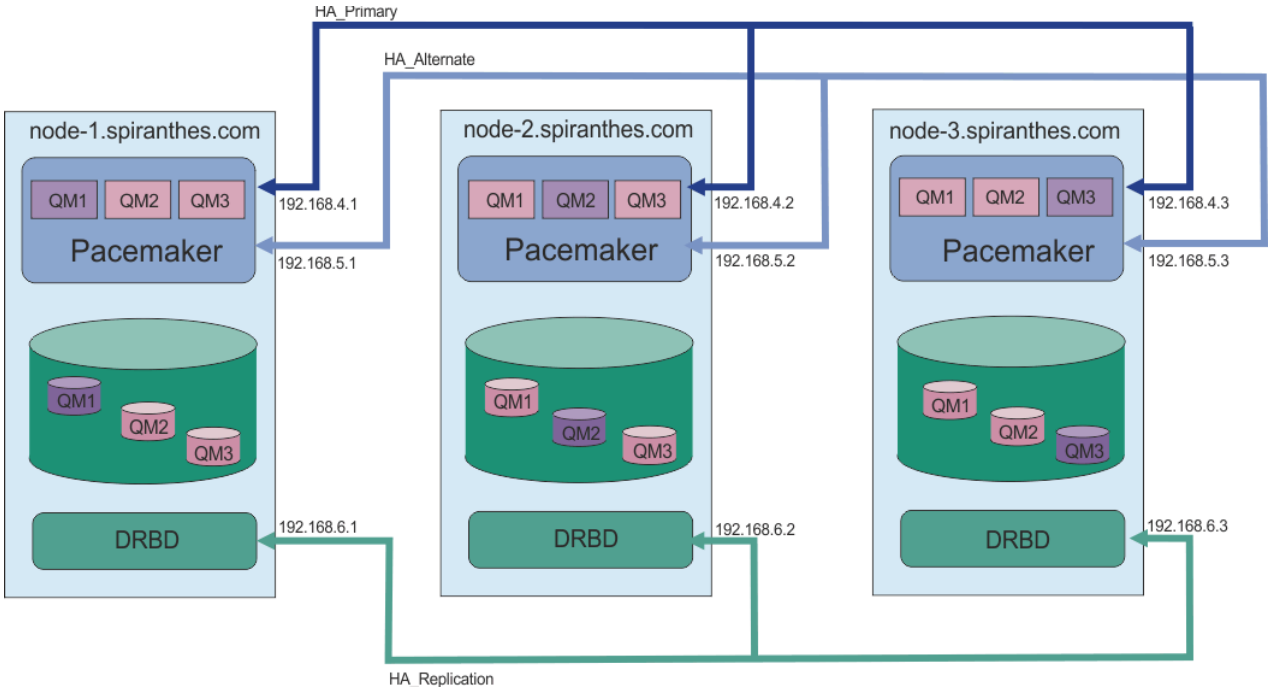
rdqm.ini 文件提供 Pacemaker 集群中所有节点的 IP 地址。 您可以定义用于 Pacemaker 监视系统的 HA_Primary 和 HA_Secondary 接口，但如果需要， Pacemaker 可以将名为 HA_Replication 的复制接口用于此目的。 对于 HA 组中运行的所有 RDQM 的预期工作负载， HA_Replication 接口必须具有足够的带宽来支持复制需求。

V 9.2.0 对于 RHEL 8 安装，必须提供每个节点的名称，该名称必须是 **uname -n** 命令返回的主机名。 对于 RHEL 7 安装，节点名的规范是可选的。

以下示例文件显示了对每个接口使用单独 IP 地址的示例 Pacemaker 集群的配置：

```
Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Primary=192.168.4.1
  HA_Alternate=192.168.5.1
  HA_Replication=192.168.6.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Primary=192.168.4.2
  HA_Alternate=192.168.5.2
  HA_Replication=192.168.6.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Primary=192.168.4.3
  HA_Alternate=192.168.5.3
  HA_Replication=192.168.6.3
```

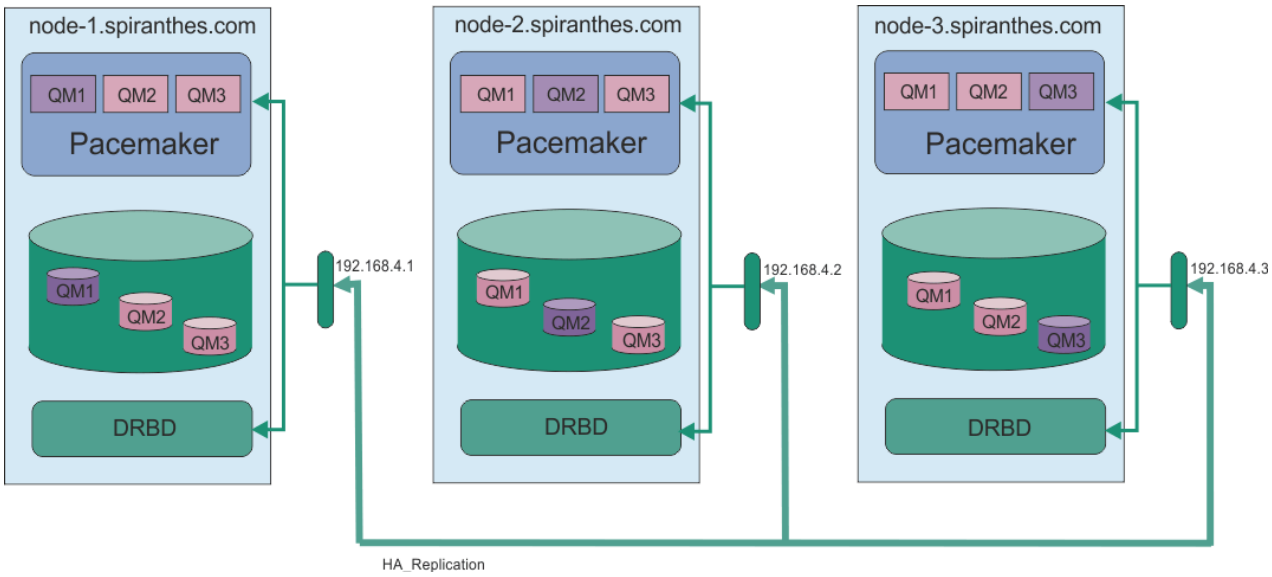
下图说明了此配置：



以下示例文件显示了使用 HA_Replication 接口进行监视的示例 Pacemaker 集群的配置。 在此情况下，仅指定 HA_Replication 接口:

```
Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Replication=192.168.4.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Replication=192.168.4.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Replication=192.168.4.3
```

下图说明了此配置:

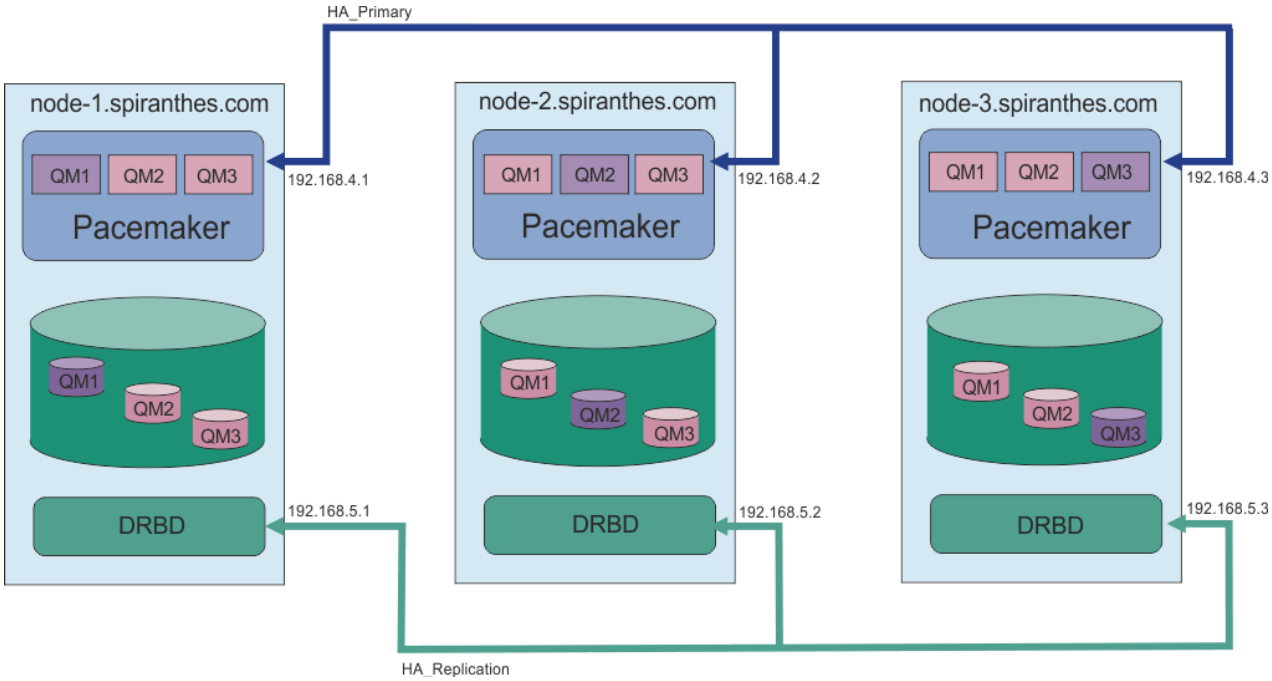


如果要使用两个 IP 地址，那么 rdqm.ini 文件针对每个节点具有 HA_Primary 和 HA_Replication 字段，但没有 HA_Alternate 字段:

```
Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Primary=192.168.4.1
```

```
  HA_Replication=192.168.5.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Primary=192.168.4.2
  HA_Replication=192.168.5.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Primary=192.168.4.3
  HA_Replication=192.168.5.3
```

下图说明了此配置:



在配置中的所有 `rdqm.ini` 文件中，指定节点的顺序必须相同。 三个节点必须具有一个公共视图，其中一个是 Node1，另一个是 Node2 ，以此类推。

过程

- 要以用户 `root` 身份定义 Pacemaker 集群，请执行以下操作:
  a) 在三个服务器之一上编辑 `/var/mqm/rdqm.ini` 文件，以便该文件定义集群。
  b) 将该文件复制到将成为 Pacemaker 集群中的节点的其他两个服务器。
  c) 在三个服务器中的每个服务器上，以 `root` 身份运行以下命令:

  ```
  rdqmadm -c
  ```

- 要将 Pacemaker 集群定义为每个节点上 `mqm` 组中的用户，请执行以下操作:
  a) 确保用户 `mqm` 可以使用 **sudo** 来运行命令。
  b) 在三个服务器之一上编辑 `/var/mqm/rdqm.ini` 文件，以便该文件定义 Pacemaker 集群。
  c) 将 `/var/mqm/rdqm.ini` 复制到将成为 Pacemaker 集群中节点的其他两个服务器。
  d) 在每个服务器上运行以下命令:

  ```
  rdqmadm -c
  ```

- 要将 Pacemaker 集群定义为来自一个节点的 `mqm` 组中的用户:
  a) 确保用户 `mqm` 可以使用 **sudo** 来运行命令，并且可以选择使用不带密码的 SSH 连接到每个服务器。
  b) 在三个服务器之一上编辑 `/var/mqm/rdqm.ini` 文件，以便该文件定义 Pacemaker 集群。
  c) 运行以下命令:

```
rdqmadm -c
```

相关参考
rdqmadm (管理复制的数据队列管理器集群)

Linux 删除 *Pacemaker* 集群 *(HA* 组*)*
HA 组是 Pacemaker 集群。 您可以通过运行带有 -u 选项的 **rdqmadm** 命令来删除 Pacemaker 集群配置。

关于此任务

如果任何节点上仍存在任何复制的数据队列管理器，那么无法删除 Pacemaker 集群配置。

过程

- 要删除 Pacemaker 集群配置，请从任何节点输入以下命令:

```
rdqmadm -u
```

相关参考
rdqmadm (管理复制的数据队列管理器集群)

Linux **创建 *HA RDQM***
您可以使用 **crtmqm** 命令来创建高可用性复制数据队列管理器 (RDQM)。

关于此任务

如果 mqm 用户可以使用 sudo，那么可以在 mqm 组中以用户身份创建高可用性复制数据队列管理器 (RDQM)。 如果用户还可以在没有密码的情况下通过 SSH 连接到每个节点，那么只需要在一个节点上运行 create RDQM 命令即可在所有三个节点上创建 RDQM。 否则，您必须是 root 才能创建 RDQM，并且必须在所有三个节点上运行命令。

注: HA 组中存在 129 个队列管理器的限制。 如果尝试创建的数量超过此数量，那么尝试将失败。

过程

- 要以 mqm 组中的用户身份创建 RDQM，请执行以下操作:
  a) 确保 mqm 用户可以使用 **sudo** 来运行命令，并且可以使用不带密码的 SSH 连接到每个服务器。
  b) 输入以下命令:

  ```
  crtmqm -sx [-fs FilesystemSize] qmname
  ```

  其中 *qmname* 是复制的数据队列管理器的名称。 您可以选择指定队列管理器的文件系统大小 (即，在 drbdpool 卷组中创建的逻辑卷的大小)。

  此命令尝试以 mqm 用户身份使用 SSH 连接到集群中的其他节点。 如果连接成功，那么将在节点上创建队列管理器的辅助实例。 否则，必须创建辅助实例，然后运行 **crtmqm -sx** 命令 (如用户 root 所述)。
- 要以用户 root 身份创建 RDQM:
  a) 在要托管 RDQM 辅助实例的每个节点上输入以下命令:

  ```
  crtmqm -sxs [-fs FilesystemSize] qmname
  ```

  其中 *qmname* 是复制的数据队列管理器的名称。 您可以选择指定队列管理器的文件系统大小 (即，在 drbdpool 卷组中创建的逻辑卷的大小)。 必须在 HA 组中的所有三个节点上为 RDQM 指定相同的文件系统大小。

  此命令将创建 RDQM 的辅助实例。
  b) 在其余节点上，输入以下命令:

```
crtmqm -sx [-fs FilesystemSize] qmname
```

其中 *qmname* 是复制的数据队列管理器的名称。 您可以选择指定队列管理器的文件系统大小。

此命令确定队列管理器的辅助实例是否存在于其他两个节点上。 如果存在辅助队列，那么该命令将创建并启动主队列管理器。 如果辅助服务器不存在，那么将指示您在每个节点上运行 **crtmqm -sxs** 命令。

除 DataPath (**-md**) 和 LogPath (**-ld**) 自变量外，对于创建标准 Linux 队列管理器有效的所有自变量也对主复制数据队列管理器有效。

注: 创建 RDQM 时，将为复制链路分配高于 7000 的下一个可用端口号。 如果发现所选端口由另一个应用程序使用，那么 **crtmqm** 命令将失败并返回 AMQ6543 错误，并且该端口将添加到排除列表中。 必须删除队列管理器的辅助实例，然后再次运行 **crtmqm** 命令。

相关参考
克特姆

Linux 删除 *HA RDQM*
您可以使用 **dltmqm** 命令来删除高可用性复制数据队列管理器 (RDQM)。

关于此任务

必须运行该命令以在 RDQM 的主节点上删除 RDQM。 必须先结束 RDQM。 如果该用户具有必需的 sudo 特权，那么可以 mqm 用户身份运行该命令。 否则，必须以 root 用户身份运行该命令。 删除与主队列管理器关联的资源后，该命令会尝试使用 ssh 删除辅助队列管理器以连接到其他节点。 如果此删除操作失败，那么必须在其他节点上手动运行 dltmqm 以完成此过程。 在辅助节点上，如果尚未删除主队列管理器，那么该命令将失败。

过程

- 要删除 RDQM ，请输入以下命令:

```
dltmqm RDQM_name
```

相关参考
德尔特姆

MQ Adv. Linux 迁移队列管理器以成为 *HA RDQM* 队列管理器
您可以迁移现有队列管理器以成为高可用性 (HA) 复制数据队列管理器 (RDQM) ， 方法是备份其持久数据，然后将数据复原到具有相同名称的新创建的 RDQM 队列管理器。

关于此任务

HA 复制数据队列管理器需要专用逻辑卷 (文件系统) 以及磁盘复制和 HA 控制的配置。 仅当创建新的队列管理器时，才会配置这些组件。 可以迁移现有队列管理器以使用 RDQM ， 方法是备份其持久数据，然后将数据复原到具有相同名称的新创建的 RDQM 队列管理器。 此过程将在创建备份时保留队列管理器配置，状态和持久消息。

注: 只能从与 RDQM 安装版本相同或更低的 IBM MQ 版本迁移队列管理器。 操作系统和体系结构也必须相同。 否则，必须在目标平台上创建新的队列管理器，请参阅 将队列管理器移至其他操作系统。

在迁移队列管理器之前，应满足以下条件:

- 评估高可用性需求，请参阅 第 463 页的『RDQM 高可用性』 。
- 查看连接到队列管理器的应用程序和队列管理器。 请考虑将连接路由到正在运行队列管理器的 RDQM 节点所需的更改。 例如，如果配置 RDQM 高可用性，那么可以考虑使用浮动 IP 地址，请参阅 第 478 页的『创建和删除浮动 IP 地址』 。
- 为所选配置供应或标识现有 RDQM 节点。 有关 RDQM 的系统需求的信息，请参阅 第 465 页的『RDQM HA 解决方案的需求』 。

- 在每个节点上安装 IBM MQ Advanced (包括 RDQM 功能部件)。
- 配置 RDQM HA 组配置，请参阅 第 468 页的『定义 Pacemaker 集群 (HA 组)』。
- (可选) 使用测试队列管理器验证 RDQM 配置，然后可以将其删除。 建议先测试配置以确定并解决任何问题，然后再迁移队列管理器。
- 请查看队列管理器的安全性配置，然后在每个 RDQM 节点上复制所需的本地用户和组。
- 查看队列管理器和通道配置以确定是否使用了 API 出口，通道出口或数据转换出口。 在每个 RDQM 节点上安装必需的出口。
- 查看已定义的任何队列管理器服务，然后在每个 RDQM 节点上安装和配置所需的进程。

过程

1. 备份现有队列管理器:
   a) 通过发出等待关闭命令 `endmqm -w` 或立即关闭命令 `endmqm -i` 来停止现有队列管理器。 此步骤对于确保备份中的数据一致非常重要。
   b) 通过查看 IBM MQ 配置文件 `mqs.ini` 来确定队列管理器数据目录的位置。 在 Linux 上，此文件位于 `/var/mqm` 目录中。 有关 `mqs.ini` 的更多信息，请参阅 第 72 页的『IBM MQ 配置文件，mqs.ini』。

      在文件中找到队列管理器的 `QueueManager` 节。 如果节包含名为 `DataPath` 的密钥，那么其值为队列管理器数据目录。 如果该键不存在，那么可以使用 `Prefix` 和 `Directory` 键的值来确定队列管理器数据目录。 队列管理器数据目录是这些值的并置，格式为 *prefix*/qmgrs/*directory*。 有关 QueueManager 节的更多信息，请参阅 第 81 页的『mqs.ini 文件的 QueueManager 节』。
   c) 创建队列管理器数据目录的备份。 在 Linux 上，可以使用 **tar** 命令来执行此操作。 例如，要备份队列管理器的数据目录，可以使用以下命令。 注意命令的最后一个参数，它是一个句点 (dot):

      ```
      tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
      ```

   d) 通过查看 IBM MQ 队列管理器配置文件 `qm.ini` 来确定队列管理器日志目录的位置。 此文件位于队列管理器数据目录中。 有关该文件的更多信息，请参阅 第 82 页的『队列管理器配置文件， qm.ini』。

      队列管理器日志目录定义为 `Log` 节中 `LogPath` 键的值。 有关该节的信息，请参阅 第 108 页的『qm.ini 文件的日志节』。
   e) 创建队列管理器日志目录的备份。 在 Linux 上，可以使用 tar 命令来执行此操作。 例如，要备份队列管理器的日志目录，可以使用以下命令。 注意命令的最后一个参数，它是一个句点 (dot):

      ```
      tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
      ```

   f) 如果队列管理器使用的任何证书存储库不在队列管理器数据目录中，请创建这些存储库的备份。 确保同时备份密钥数据库文件和密码隐藏文件。 有关队列管理器密钥存储库的信息，请参阅 SSL/TLS 密钥存储库 和 查找队列管理器的密钥存储库。 有关在队列管理器配置为使用 AMS 消息通道代理程序 (MCA) 拦截时查找 AMS 密钥库的信息，请参阅 消息通道代理程序 (MCA) 拦截。
   g) 不再需要现有队列管理器，因此可以将其删除。 但是，在可能的情况下，仅应在目标系统上成功复原现有队列管理器后将其删除。 延迟删除可确保在迁移过程未成功完成时可以重新启动队列管理器。

      注: 如果延迟删除现有队列管理器，请勿将其重新启动。 由于在迁移期间丢失了对队列管理器配置或状态的进一步更改，因此队列管理器保持结束很重要。
2. 准备主 RDQM 节点:
   a) 创建与您备份的队列管理器同名的新 RDQM 队列管理器。 请确保 **crtmqm** 为 RDQM 队列管理器分配的文件系统足够大，足以包含现有队列管理器的数据，主日志和辅助日志，以及用于将来扩展的一些额外空间。 有关如何创建 RDQM 队列管理器的信息，请参阅 第 471 页的『创建 HA RDQM』。
   b) 确定队列管理器的主 RDQM 节点。 有关如何确定主节点的信息，请参阅 rdqmstatus (display RDQM status)。
   c) 在主 RDQM 节点上，如果 RDQM 队列管理器已启动，请使用 `endmqm -w` 或 `endmqm -i` 命令将其停止。

d) 在主 RDQM 节点上，确定 RDQM 队列管理器的数据和日志目录的位置 (使用步骤 1b 和 1d 中描述的方法)。

e) 在主 RDQM 节点上，删除 RDQM 队列管理器数据和日志目录的内容，但不删除目录本身。

3. 在主 RDQM 节点上复原队列管理器:

a) 将队列管理器数据和日志目录的备份复制到主 RDQM 节点，以及队列管理器所使用的证书存储库的任何单独备份。

b) 将队列管理器数据目录的备份复原到新的 RDQM 队列管理器的空数据目录，确保保留文件所有权和许可权。 如果备份是使用步骤 1c 中的示例 tar 命令创建的，那么 root 用户可以使用以下命令来复原备份:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

c) 将队列管理器日志目录的备份复原到新的 RDQM 队列管理器的空日志目录，确保保留文件所有权和许可权。 如果备份是使用步骤 1e 中的示例 tar 命令创建的，那么 root 用户可以使用以下命令来将其复原:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

d) 在 RDQM 队列管理器的数据目录中编辑复原的队列管理器配置文件 `qm.ini`。 更新 `Log` 节中 `LogPath` 键的值，以指定 RDQM 队列管理器的日志目录。

查看配置文件中定义的其他文件路径，并在必要时进行更新。 例如，您可能需要更新以下路径:

- 诊断消息服务生成的错误日志文件的路径。
- 队列管理器所需的出口的路径。
- 如果队列管理器是 XA 事务协调程序，那么这是交换机装入文件的路径。

e) 如果队列管理器配置为使用 AMS 消息通道代理程序 (MCA) 拦截，请将 AMS 密钥库复制到新的 RDQM 安装，然后复审并更新配置。 密钥库必须在每个 RDQM 节点上都可用，因此如果它不在队列管理器的复制文件系统中，那么必须改为将其复制到每个节点。 有关更多信息，请参阅 消息通道代理程序 (MCA) 拦截。

f) 验证 **dspmq** 命令是否显示了队列管理器，并且其状态是否报告为 "已结束"。 以下示例显示 RDQM HA 队列管理器的样本输出:

```
$ dspmq -o status -o ha

QMNAME(QM1) STATUS(Ended normally) HA(Replicated)
```

g) 通过使用 **rdqmstatus** 命令显示队列管理器的状态，验证复原的队列管理器数据是否已复制到辅助 RDQM 节点。 在每个节点上， HA 状态应报告为 `Normal` 。 以下示例显示 RDQM HA 队列管理器的样本输出:

```
$ rdqmstatus -m QM1
Node:                           mqhavm10-adm
Queue manager status:           Ended normally
Queue manager file system:      50MB used, 0.2GB allocated [42%]
HA role:                        Primary
HA status:                      Normal
HA control:                     Disabled
HA current location:            This node
HA preferred location:          This node
HA floating IP interface:       None
HA floating IP address:         None

Node:                           mqhavm11-adm
HA status:                      Normal

Node:                           mqhavm12-adm
HA status:                      Normal
```

h) 在主 RDQM 节点上启动队列管理器。

i) 连接到队列管理器并更新 SSLKEYR 队列管理器属性的值，以指定队列管理器证书存储库的新位置。 缺省情况下，此属性的值设置为 *queue_manager_data_directory*/ssl/key。 证书存储库必须

位于每个 RDQM 节点上的相同位置。 如果存储库未位于队列管理器的复制文件系统中，那么必须改为将其复制到每个节点。

j) 查看队列管理器的 IBM MQ 对象定义，并更新引用已更改的网络设置， IBM MQ 安装目录或队列管理器数据目录 (包括以下对象) 的对象属性值:

- 侦听器使用的本地 IP 地址 (IPADDR 属性)。
- 通道使用的本地 IP 地址 (LOCLADDR 属性)。
- 为集群接收方通道定义的本地 IP 地址 (CONNAME 属性)。
- 为通信信息对象定义的本地 IP 地址 (GRPADDR 属性)。
- 为流程和服务对象定义定义的系统路径。

k) 停止并重新启动队列管理器以确保更改生效。

l) 对连接到已迁移队列管理器的远程队列管理器以及应用程序的等效设置重复步骤 3j ，包括:

- 通道连接名称 (CONNAME 属性)。
- 通道认证规则，根据其 IP 地址或主机名来限制来自队列管理器的入站连接。
- 客户机通道定义表 (CCDT) ， 域名设置 (DNS) ， 网络路由或等效连接信息。

m) 执行队列管理器到每个 RDQM 节点的受管故障转移，以确保已成功建立所需的配置，请参阅 第 477 页的『设置 RDQM 的首选位置』。

调整 *HA RDQM* 队列管理器的文件系统大小
要调整现有高可用性 (HA) 复制数据队列管理器 (RDQM) 的文件系统大小，请备份其持久数据，然后将数据复原到具有相同名称但具有不同大小的文件系统的新创建的 RDQM 队列管理器。

关于此任务

HA 复制数据队列管理器需要专用逻辑卷 (文件系统) 以及磁盘复制和 HA 控制的配置。 仅当创建新的队列管理器时，才会配置这些组件。 文件系统在创建后无法调整大小，因为它在每个节点上必须具有相同的大小。 要调整现有复制数据队列管理器 (RDQM) 的文件系统大小，您可以备份其持久数据，然后将数据复原到具有相同名称但具有不同大小的文件系统的新创建的 RDQM 队列管理器。 此过程将在创建备份时保留队列管理器配置，状态和持久消息。

过程

1. 备份主 RDQM 节点上的现有 RDQM 队列管理器:
   a) 确定队列管理器的主 RDQM 节点。 有关如何确定主节点的信息，请参阅 rdqmstatus (display RDQM status) 。
   b) 在主 RDQM 节点上，如果 RDQM 队列管理器已启动，请使用 **endmqm -w** 或 **endmqm -i** 命令将其停止。
   c) 通过查看 IBM MQ 配置文件 `mqs.ini` 来确定队列管理器数据目录的位置。 在 Linux 上，此文件位于 `/var/mqm` 目录中。 有关 `mqs.ini` 的更多信息，请参阅 第 72 页的『IBM MQ 配置文件，mqs.ini』。

   在文件中找到队列管理器的 `QueueManager` 节。 队列管理器数据目录是名为 `DataPath` 的键的值。 有关 `QueueManager` 节的更多信息，请参阅 第 81 页的『mqs.ini 文件的 QueueManager 节』。
   d) 创建队列管理器数据目录的备份。 在 Linux 上，可以使用 **tar** 命令来执行此操作。 例如，要备份队列管理器的数据目录，可以使用以下命令。 请注意命令的最后一个参数，它是单个句点 (.) 字符:

   ```
   tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
   ```

   e) 通过查看 IBM MQ 队列管理器配置文件 `qm.ini` 来确定队列管理器日志目录的位置。 此文件位于队列管理器数据目录中。 有关该文件的更多信息，请参阅 第 82 页的『队列管理器配置文件， qm.ini』。

   队列管理器日志目录定义为 `Log` 节中 `LogPath` 键的值。 有关该节的信息，请参阅 第 108 页的『qm.ini 文件的日志节』。

f) 创建队列管理器日志目录的备份。 在 Linux 上，可以使用 **tar** 命令来执行此操作。 例如，要备份队列管理器的日志目录，可以使用以下命令。 请注意命令的最后一个参数，它是单个句点 (.) 字符:

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

g) 删除现有 RDQM 队列管理器。

2. 使用所需大小的文件系统复原队列管理器:

a) 创建与您备份的队列管理器同名的新 RDQM 队列管理器。 确保按 **crtmqm** 为 RDQM 队列管理器分配的文件系统是您需要的大小，并且它足够大，可以包含现有队列管理器的数据，主日志和辅助日志，以及用于将来扩展的一些额外空间。 有关如何创建 RDQM 队列管理器的信息，请参阅 第 471 页的『创建 HA RDQM』。

b) 确定队列管理器的主 RDQM 节点。 有关如何确定主节点的信息，请参阅 rdqmstatus (display RDQM status)。

c) 在主 RDQM 节点上，如果 RDQM 队列管理器已启动，请使用 **endmqm -w** 或 **endmqm -i** 命令将其停止。

d) 在主 RDQM 节点上，确定 RDQM 队列管理器的数据和日志目录的新位置 (使用步骤 1c 和 1e 中描述的方法)。

e) 在主 RDQM 节点上，删除 RDQM 队列管理器数据和日志目录的内容，但不删除目录本身。

f) 在主 RDQM 节点上，将队列管理器数据目录的备份复原到新的 RDQM 队列管理器的空数据目录，确保保留文件所有权和许可权。 如果备份是使用步骤 1d 中的示例 **tar** 命令创建的，那么 root 用户可以使用以下命令来复原备份:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

g) 在主 RDQM 节点上，将队列管理器日志目录的备份复原到新的 RDQM 队列管理器的空日志目录，确保保留文件所有权和许可权。 如果备份是使用步骤 1f 中的示例 **tar** 命令创建的，那么 root 用户可以使用以下命令来复原备份:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

h) 在主 RDQM 节点上，编辑新 RDQM 队列管理器的数据目录中复原的队列管理器配置文件 `qm.ini`。 更新 `Log` 节中 `LogPath` 键的值，以指定在步骤 2d 中确定的新 RDQM 队列管理器的日志目录。 查看配置文件中定义的其他文件路径，并在必要时进行更新。 例如，您可能需要更新以下路径:

- 诊断消息服务生成的错误日志文件的路径。
- 队列管理器所需的出口的路径。
- 如果队列管理器是 XA 事务协调程序，那么这是交换机装入文件的路径。

i) 验证 **dspmq** 命令是否显示了队列管理器，并且其状态是否报告为 "已结束"。 以下示例显示 RDQM HA 队列管理器的样本输出:

```
$ dspmq -o status -o ha
QMNAME(QM1) STATUS(Ended normally) HA(Replicated)
```

j) 通过使用 **rdqmstatus** 命令显示队列管理器的状态，验证复原的队列管理器数据是否已复制到辅助 RDQM 节点。 在每个节点上， HA 状态应报告为 `Normal` 。 以下示例显示 RDQM HA 队列管理器的样本输出:

```
$ rdqmstatus -m QM1
Node: mqhavm10-adm
Queue manager status:         Ended normally
Queue manager file system:    50MB used, 0.2GB
allocated [42%]
HA role:                      Primary
HA status:                    Normal
HA control:                   Disabled
HA current location:          This node
HA preferred location:        This node
HA floating IP interface:     None
HA floating IP address:       None
Node:                         mqhavm11-adm
HA status:                    Normal
```

```
Node:                           mqhavm12-adm
HA status:                      Normal
```

k) 在主 RDQM 节点上启动队列管理器。

l) 执行队列管理器到每个 RDQM 节点的受管故障转移，以确保已成功建立所需的配置，请参阅 第 477 页的『设置 RDQM 的首选位置』。

V 9.2.0 存储持久应用程序状态

您可以存储与应用程序相关的持久状态信息以及其他队列管理器数据。

每个 IBM MQ 队列管理器都有一个用于其持久状态的专用文件系统，其中包括其队列数据和恢复日志。 在 RDQM 配置中，文件系统由在 Linux 系统 (节点) 之间复制的逻辑卷支持。 文件系统包含可用于存储应用程序的持久状态信息的 `userdata` 目录。 因此，当复制的数据队列管理器移动到 RDQM 配置中的另一个节点上运行时，您可以使用应用程序上下文以及队列管理器上下文。 请参阅 Unix 和 Linux Systems 上的目录内容。

如果选择将应用程序状态存储在 `userdata` 目录中，那么您必须知道写入此位置的数据可能会消耗分配给队列管理器的可用磁盘空间。 必须确保有足够的磁盘空间可供队列管理器写入队列数据，日志和其他持久状态信息。

`userdata` 目录具有 mqm 用户和组所有权，并且具有全局可读性，因此用户无需在 IBM MQ 管理员组 (即 mqm) 中即可访问该目录。 您无法修改 `userdata` 目录的许可权，但可以使用所需的任何所有权和许可权在其中创建内容。

在 RDQM 队列管理器故障转移期间，将结束队列管理器，并在其当前 RDQM 节点上卸载其文件系统。 然后安装文件系统，并在 RDQM 配置中的另一个节点上重新启动队列管理器。 如果进程具有其某个文件的打开句柄，那么无法卸装文件系统。 为了确保队列管理器故障转移可以完成，如果无法卸装队列管理器文件系统，那么将向具有打开文件句柄的进程发送 SIGTERM 信号，如果未释放打开的句柄，那么将后跟 SIGKILL。 您的应用程序必须设计为正确响应 SIGTERM。 如果将应用程序或进程配置为队列管理器服务，那么在受管故障转移期间，可以在卸载文件系统之前在队列管理器关闭期间结束这些应用程序或进程。 如果应用程序或进程未配置为队列管理器服务，或者发生非受管故障转移 (例如丢失定额)，那么很可能将发送信号以释放文件系统。

Linux 设置 *RDQM* 的首选位置

复制数据队列管理器 (RDQM) 的首选位置标识当 RDQM 可用时应运行该 RDQM 的节点。

关于此任务

首选位置是当 HA 组处于正常状态 (所有可用节点和连接) 时， Pacemaker 应该在其上运行队列管理器的节点的名称。 创建队列管理器时，会将 "首选位置" 初始化为主节点的名称。 您可以运行命令以在三个节点中的任何一个节点上设置 "首选位置"。 您必须是同时属于 `mqm` 和 `haclient` 组的用户。

过程

- 要将本地节点或指定节点指定为指定队列管理器的首选位置，请输入以下命令:

  ```
  rdqmadm -p -m qmname [ -n nodename[,nodename  ]
  ```

  其中 *qmname* 是要为其指定首选位置的 RDQM 的名称，而 *nodename* 是 (可选) 首选节点的名称。

  如果 HA 组处于正常状态并且 "首选位置" 不是当前主节点，那么将在新的 "首选位置" 上停止并重新启动队列管理器。 您可以指定两个节点名的逗号分隔列表，以指定首选位置的第二个首选项。

- 要清除 "首选位置"，以便队列管理器在复原时不会自动返回到节点，请输入以下命令:

  ```
  rdqmadm -p -m qmname -d
  ```

相关参考

rdqmadm (管理复制的数据队列管理器集群)

Linux 创建和删除浮动 *IP* 地址

浮动 IP 地址使客户机能够对复制的数据队列管理器 (RDQM) 使用相同的 IP 地址，而不管它在 HA 组中的哪个节点上运行。 (使用浮动 IP 地址是可选的。)

关于此任务

您可以使用 **rdqmint** 命令创建或删除浮动 IP 地址。 浮动地址绑定到 RDQM 的主节点上的指定物理接口。 如果 RDQM 故障转移，并且另一个节点成为主节点，那么浮动 IP 将绑定到新主节点上同名的接口。 三个节点上的物理接口必须属于与浮动 IP 地址相同的子网。 下图说明了浮动 IP 地址的使用。

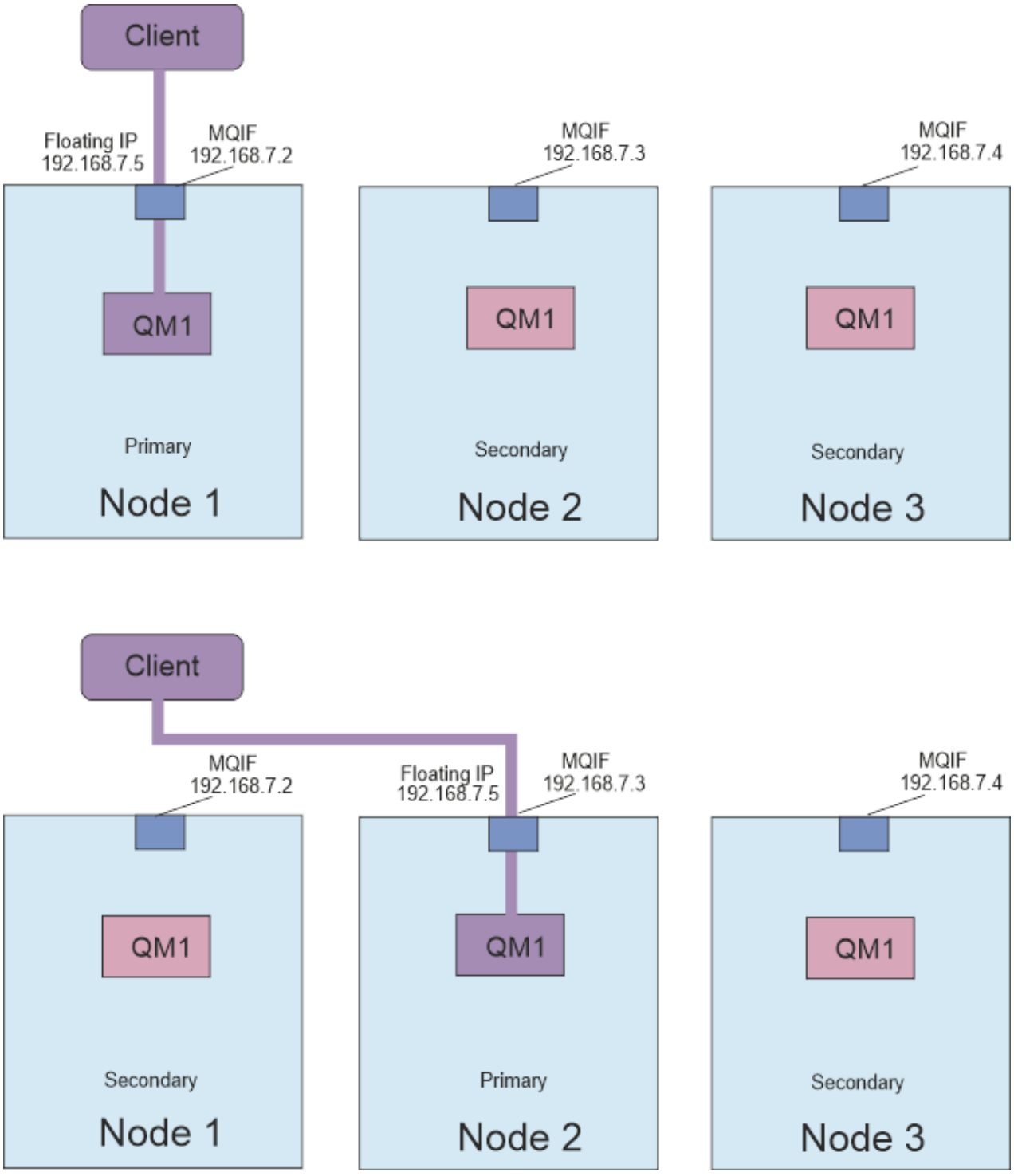


图 *83:* 浮动 *IP* 地址

您必须是 mqm 和 haclient 组中的用户才能运行 **rdqmint** 命令。 您可以在 RDQM 的主节点上或其中一个辅助节点上创建或删除浮动 IP 地址。

注: 不能对多个 RDQM 使用相同的浮动 IP 地址，每个 RDQM 的浮动 IP 地址必须唯一。

过程

- 要为 RDQM 创建浮动 IP 地址，请输入以下命令:

  ```
  rdqmint -m qmname -a  -f ipv4address -l interfacename
  ```

  其中:

  qmname
  要为其创建浮动 IP 地址的 RDQM 的名称。

  ipv4address
  ipv4 格式的浮动 IP 地址。

  浮动 IP 地址必须是尚未在任何 HA 节点上定义的有效 IPv4 地址，并且它必须与为本地接口定义的静态 IP 地址属于同一子网。

  interfaceName
  要绑定到的主节点上的物理接口的名称。

  例如:

  ```
  rdqmint -m QM1 -a -f 192.168.7.5 -l MQIF
  ```

- 要删除现有浮动 IP 地址，请输入以下命令:

  ```
  rdqmint -m qmname -d
  ```

相关参考
rdqmint (添加或删除 RDQM 的浮动 IP 地址)

Linux 启动，停止和显示 *HA RDQM* 的状态

使用标准 IBM MQ 控制命令的变体来启动，停止和查看复制的数据队列管理器 (RDQM) 的当前状态。

关于此任务

必须以属于 mqm 和 haclient 组的用户身份运行用于启动，停止和查看复制数据队列管理器 (RDQM) 的当前状态的命令。

您必须运行命令以在该队列管理器的主节点上启动和停止队列管理器。

过程

- 要启动 RDQM ，请在 RDQM 的主节点上输入以下命令:

  ```
  strmqm qmname
  ```

  其中 *qmname* 是要启动的 RDQM 的名称。

  RDQM 已启动， Pacemaker 开始管理 RDQM。 如果要指定任何其他 strmqm 选项，那么必须将 -ns 选项与 strmqm 一起指定。

- 要停止 RDQM ，请在 RDQM 的主节点上输入以下命令:

  ```
  endmqm qmname
  ```

  其中 *qmname* 是要停止的 RDQM 的名称。

  Pacemaker 停止管理 RDQM ，然后结束 RDQM。 停止 RDQM 时，可以使用所有其他 **endmqm** 参数。

- 要查看 RDQM 的状态，请输入以下命令:

```
dspmq
```

输出的状态信息取决于您是在 RDQM 的主节点还是辅助节点上运行该命令。 如果在主节点上运行，那么将显示 **dspmq** 返回的其中一条正常状态消息。 如果在辅助节点上运行该命令，那么将显示状态 `running elsewhere`。 例如，如果在节点 RDQM7 上运行 **dspmq**，那么可能会返回以下信息:

```
QMNAME(RDQM8)                             STATUS(Running elsewhere)
QMNAME(RDQM9)                             STATUS(Running elsewhere)
QMNAME(RDQM7)                             STATUS(Running)
```

如果主节点不可用，或者 **dspmq** 由不是 `root` 的用户或 `haclient` 组的成员运行，那么将报告 `Unavailable` 状态。 例如:

```
QMNAME(RDQM8)               STATUS(Unavailable)
QMNAME(RDQM9)               STATUS(Unavailable)
QMNAME(RDQM7)               STATUS(Unavailable)
```

可以输入命令 **dspmq -o ha** (或 **dspmq -o HA**) 以查看节点已知的队列管理器列表，以及它们是否为 RDQM，例如:

```
dspmq -o ha

QMNAME(RDQM8)                                           HA(Replicated)
QMNAME(RDQM9)                                           HA(Replicated)
QMNAME(RDQM7)                                           HA(Replicated)
QMNAME(QM7)                                             HA()
```

相关参考

dspmq (显示队列管理器)

endmqm (结束队列管理器)

strmqm (启动队列管理器)

V 9.2.2 失败的资源操作

当 RDQM 高可用性配置的 Pacemaker 组件在 HA 组中的某个节点上迁到资源问题时，将发生失败的资源操作。

RDQM HA 解决方案使用 Pacemaker 来监视和管理资源 (请参阅 第 463 页的『RDQM 高可用性』)。 如果 Pacemaker 在对节点上的资源执行操作时迁到错误，那么它会使用失败的资源操作来记录此信息。 某些失败的资源操作会阻止资源运行，必须先清除这些操作，然后 Pacemaker 才能重新启动资源。

您可以使用 **rdqmstatus -m** 命令来查看是否有任何失败的资源操作正在阻止队列管理器在一个或多个节点上启动。

然后，可以使用 **rdqmstatus -m** *qmname* **-a** 命令来查看与队列管理器关联的失败资源操作的详细信息。 通过使用 **rdqmclean** 命令来执行此操作，以清除这些失败的资源操作，从而释放所有受限资源。 (您还必须采取行动，首先解决导致资源操作失败的问题。)

以下资源由 RDQM HA 配置中的 Pacemaker 控制，并且可以是失败资源操作的主体集:

- 队列管理器
- 浮动 IP
- RDQM 控制
- 文件系统
- DR 复制 (DRBD)
- HA 复制 (DRBD)

每种类型的资源都可能发生以下类型的故障:

软
: 软故障是瞬态的， Pacemaker 会继续尝试恢复资源，直到它超时或以其他方式停止为止。

困难
: 硬错误需要行政干预。 硬错误会阻止资源在特定节点上运行。

致命
致命错误需要进行管理干预。 致命错误会阻止资源在任何节点上运行。

请参阅 第 481 页的『查看 RDQM 和 HA 组状态』 以获取状态示例，包括失败的资源队列操作。

您可以使用 **rdqmclean** 命令来清除与指定队列管理器关联的所有失败资源操作，或者清除 RDQM HA 配置中的所有失败资源操作。

注: 某些失败的资源操作不会导致队列管理器在节点上受阻。 例如，在意外队列管理器结束后， Pacemaker 尝试在发现未在其上运行的节点上重新启动队列管理器。 如果启动成功，那么不会阻止队列管理器在节点上运行。 在这种情况下，了解失败的资源操作的唯一方法是运行 **rdqmstatus -m** *qmname* **-a**。

相关任务

第 481 页的『查看 RDQM 和 HA 组状态』
您可以查看 HA 组和个别复制数据队列管理器 (RDQM) 的状态。

相关参考

Rdqmclean

rdqmstatus

Linux 查看 *RDQM* 和 *HA* 组状态

您可以查看 HA 组和个别复制数据队列管理器 (RDQM) 的状态。

关于此任务

您可以使用 **rdqmstatus** 命令来查看单个 RDQM 以及整个 HA 组的状态。

V 9.2.3 节点的摘要状态还显示有关 RDQM 所依赖的 DRBD 内核模块的信息。 升级 RDQM 时，请务必确保为系统上运行的 RHEL 内核版本安装了正确版本的 DRBD 内核模块。 状态显示操作系统内核的版本，为其构建 DRBD 模块的内核版本， DRBD 版本和 DRBD 内核模块装入状态。

您必须是 mqm 和 haclient 组中的用户才能运行 **rdqmstatus** 命令。 您可以在三个节点中的任何一个节点上运行该命令。

过程

- 要查看属于 HA 配置的节点和 RDQM 的摘要状态:

  ```
  rdqmstatus
  ```

  将显示在上运行命令的节点的身份，该节点的内核和 DRBD 详细信息， 以及 HA 配置中 RDQM 的状态，例如:

  ```
  Node:                                mqhavm07.exampleco.com
  OS kernel version:                   3.10.0-1160.15.2
  DRBD OS kernel version:              3.10.0-1160
  DRBD version:                        9.1.1
  DRBD kernel module status:           Loaded

  Queue manager name:                  RDQM8
  Queue manager status:                Running elsewhere
  HA current location:                 mqhavm08.exampleco.com
  HA preferred location:               mqhavm08.exampleco.com
  HA blocked location:                 None

  Queue manager name:                  RDQM9
  Queue manager status:                Running elsewhere
  HA current location:                 mqhavm09.exampleco.com
  HA preferred location:               mqhavm09.exampleco.com
  HA blocked location:                 None

  Queue manager name:                  RDQM7
  Queue manager status:                Running
  HA current location:                 This node
  HA preferred location:               This node
  HA blocked location:                 None
  ```

V 9.2.3 DRBD 内核模块状态为以下值之一:

已加载
指示已装入 DRBD 模块。

已部分加载
装入 DRBD 模块时可能发生，但由于不匹配而无法正常工作。

未加载
未装入 DRBD 模块。 当尚未创建任何 RDQM 队列管理器时，可以在新安装的配置上显示此消息。

未安装
指示未安装 DRBD 模块。 或者 IBM MQ 无法确定 DRBD 模块的操作系统内核版本。

仍加载先前安装的版本
如果在现有 DRBD 模块正在运行 (即 RDQM 队列管理器正在运行) 时安装了新的 DRBD 模块，那么会出现此状态。 新安装的模块在状态中报告，但并非实际正在运行的模块。

- 要查看 HA 组中三个节点的状态，请输入以下命令:

```
rdqmstatus -n
```

报告每个节点的联机或脱机状态。 例如:

```
Node mqha04(mqhavm04.example.com) is online
Node mqha05(mqhavm05.example.com) is offline
Node mqha06(mqhavm06.example.com) is online
```

- 要查看 HA 组中所有节点上特定队列管理器的状态，请输入以下命令:

```
rdqmstatus -m qmname
```

其中 *qmname* 是要查看其状态的 RDQM 的名称。 将显示当前节点上 RDQM 的状态，后跟当前节点透视图中其他两个节点的状态摘要。

- V 9.2.2
要查看 HA 组中所有节点上特定队列管理器的状态，包括任何失败资源操作的详细信息，请输入以下命令:

```
rdqmstatus -m qmname -a
```

其中 *qmname* 是要查看其状态的 RDQM 的名称。 将显示当前节点上 RDQM 的状态，后跟当前节点透视图中其他两个节点的状态摘要。 接下来是有关与 RDQM 关联的任何失败资源操作的详细信息。

- 下表汇总了 `rdqmstatus -m qmname` 命令可针对 RDQM 返回的有关当前节点的信息。

表 *33:* 当前节点状态

状态属性	**可能的值**	**显示时**
节点名	*NODENAME*	始终显示
队列管理器状态	正在运行 正在其他地方运行 已结束 不可用	始终显示
CPU	*n.nn*%	仅当当前节点具有主角色 (即, RDQM 正在此节点上运行) 时才显示
内存	已使用 *nnn*MB， 已分配 *y.y*GB	仅当当前节点具有主角色 (即, RDQM 正在此节点上运行) 时才显示
队列管理器文件系统	已使用 *nnn*MB， *y.y*GB 已分配 [*z*%]	仅当当前节点具有主角色 (即, RDQM 正在此节点上运行) 时才显示

表 33: 当前节点状态 (继续)		
状态属性	**可能的值**	**显示时**
HA 角色	主辅助未知	始终显示
HA 状态	所有节点处于备用状态 此节点处于备用状态 远程节点处于备用状态 混合 远程节点的状态	所有节点处于备用状态 当前节点处于备用状态 两个远程节点都处于备用状态 每个远程节点的不同状态 (请参阅下表以了解各个状态) 两个远程节点的状态相同 (请参阅下表以了解所有值)
HA 控制	已启用 已禁用 未知	始终显示。 显示 RDQM 是否受 Pacemaker 控制
HA 首选位置	None 此节点 未知 *NODENAME*	始终显示
V 9.2.2 V 9.2.2 HA 阻止的位置	无-未阻止队列管理器在任何节点上运行 此节点-由于一个或多个失败的资源操作，阻止队列管理器在当前节点上运行 *nodename* -由于一个或多个失败的资源操作，阻止队列管理器在 *nodename* 上运行 *nodename1*， *nodename2* -由于一个或多个失败的资源操作，队列管理器被阻止在 *nodename1* 和 *nodename2* 上运行 所有节点-由于一个或多个失败的资源操作，阻止队列管理器在所有节点上运行	始终显示
HA 浮动 IP 接口	*interface_name*	始终显示
HA 浮动 IP 地址	*IPV4_address*	始终显示

下表汇总了 `rdqmstatus -m qmname` 命令针对 HA 组中的其他节点返回的信息。

表 34: 其他节点状态		
状态属性	**可能的值**	**显示时**
节点名	*nodename*	始终显示

表 *34:* 其他节点状态 *(*继续*)*		
状态属性	**可能的值**	**显示时**
HA 状态	常规 同步操作正在执行中 远程不可用 不一致 已暂停 远程节点处于备用状态 未知	节点相互同步 与远程节点同步 无法与远程节点通信 与远程节点不同步，并且未同步复制已暂停 远程节点处于备用状态
HA 同步正在进行中	*n*。*n*%	同步进行时显示，命令作为 `root` 运行
HA 估计同步时间	*yyyy-mm-dd hh:mm:ss.nnn*	同步正在进行时显示
HA 不同步数据	*n*KB	当远程节点不可用或不一致时显示
V 9.2.2 V 9.2.2 HA 上次同步	*yyyy-mm-dd hh:mm:ss.nnn*	HA 数据不同步 (在初始同步后) 时显示。 提供上次同步数据的时间和日期。

示例

主节点上的正常状态示例:

```
Node:                                 mqhavm07.exampleco.com
Queue manager status:                  Running
CPU:                                   0.00
Memory:                                123MB
Queue manager file system:             606MB used, 1.0GB allocated [60%]
HA role:                               Primary
HA status:                             Normal
HA control:                            Enabled
HA current location:                   This node
HA preferred location:                 This node
HA preferred location:                 This node
HA blocked location:                   None
HA floating IP interface:              eth4
HA floating IP address:                192.0.2.4


Node:                                  mqhavm08.exampleco.com
HA status:                             Normal


Node:                                  mqhavm09.exampleco.com
HA status:                             Normal
```

辅助节点上的正常状态示例:

```
Node:                                  mqhavm08.exampleco.com
Queue manager status:                  Running elsewhere
HA role:                               Secondary
HA status:                             Normal
HA control:                            Enabled
HA current location:                   mqhavm07.exampleco.com
HA preferred location:                 mqhavm07.exampleco.com
HA blocked location:                   None
HA floating IP interface:              eth4
HA floating IP address:                192.0.2.4

Node:                                  mqhavm07.exampleco.com
HA status:                             Normal

Node:                                  mqhavm09.exampleco.com
HA status:                             Normal
```

正在进行同步时主节点上的状态示例:

```
Node:                                   mqhavm07.exampleco.com
Queue manager status:                   Running
CPU:                                    0.53
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Synchronization in progress
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm08.exampleco.com
HA status:                              Synchronization in progress
HA synchronization progress:            11.0%
HA estimated time to completion:        2017-09-06 14:55:05

Node:                                   mqhavm09.exampleco.com
HA status:                              Synchronization in progress
HA synchronization progress:            11.0%
HA estimated time to completion:        2017-09-06 14:55:06
```

V 9.2.2 失去同步时主节点上的状态示例:

```
Node:                                   mqhavm07.exampleco.com
Queue manager status:                   Running
CPU:                                    0.53
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Mixed
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm08.exampleco.com
HA status:                              Normal

Node:                                   mqhavm09.exampleco.com
HA status:                              Inconsistent
HA out of sync data:                    15932KB
HA last in sync:                        2017-09-06 14:55:06
```

显示多个状态的主节点的示例:

```
Node:                                   mqhavm07.exampleco.com
Queue manager status:                   Running
CPU:                                    0.02
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Mixed
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm08.exampleco.com
HA status:                              Normal

Node:                                   mqhavm09.exampleco.com
HA status:                              Inconsistent
```

V 9.2.2 显示失败资源操作的主节点的示例:

```
Node:                                 mqhavm07.exampleco.com
Queue manager status:                 Running
```

```
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              606MB used, 1.0GB allocated [60%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  mqhavm08.exampleco.com
HA blocked location:                    mqhavm08.exampleco.com
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm08.exampleco.com
HA status:                              Normal

Node:                                   mqhavm09.exampleco.com
HA status:                              Normal

Failed resource action:                 Start
Resource type:                          Filesystem
Failure node:                           mqhavm08.exampleco.com
Failure time:                           2017-09-06 12:00:00
Failure reason:                         Couldn't find directory  [/var/mqm/vols/qmname] to use
as a mount point
Blocked location:                       mqhavm08.exampleco.com
```

此状态显示 Pacemaker 无法在 12:00:00 启动节点 mqhavm08.exampleco.com 上的文件系统。 此失败的资源操作意味着阻止队列管理器在 mqhavm08.exampleco.com 上运行。 解决导致失败的资源操作的底层问题后，运行 **rdqmclean** 命令以清除失败的操作，以便 Pacemaker 可以重试该操作 (如果需要)。

V 9.2.3 显示操作系统内核版本 (RHEL 7.9) 与 DRBD 内核模块 (针对 RHEL 7.8) 不匹配的摘要状态示例。 即使状态报告已装入 DRBD 内核模块并且队列管理器正在运行，您也应该在此情况下使用针对正在运行的操作系统内核的版本来更新 DRBD 内核模块。

```
Node:                                   mqhavm07.exampleco.com
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-1127
DRBD version:                           9.1.1
DRBD kernel module status:              Loaded

Queue manager name:                     RDQM7
Queue manager status:                   Running
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
```

V 9.2.3 显示操作系统内核版本 (RHEL 7.9) 与 DRBD 内核模块 (针对 RHEL 7.6) 不匹配的摘要状态示例。 在此示例中，版本不匹配更严重，并且 DRBD 内核模块无法成功装入。 因此，队列管理器无法在其首选节点上启动，并且无法在 Unknown 中启动其 HA 状态。 要解决此故障，必须使用正在运行的操作系统内核的版本目标来更新 DRBD 内核模块。

```
Node:                                   mqhavm57.exampleco.com
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-957
DRBD version:                           9.1.2+ptf.3
DRBD kernel module status:              Partially loaded

Queue manager name:                     QM2
Queue manager status:                   Running elsewhere
HA status:                              Unknown
HA current location:                    mqhavm58.exampleco.com
HA preferred location:                  This node
HA blocked location:                    All nodes
```

相关参考

Linux rdqmstatus

更改高可用性配置中的 *IP* 地址

如果更改高可用性配置中任何接口的 IP 地址，那么高可用性操作将不再可用，并且队列管理器将不会在更改了这些地址的节点上运行。

在 rdqm.ini 文件中最多为 HA 操作指定三个 IP 地址。 如果已更改 Pacemaker 监视器地址，那么必须在执行此过程之前将其临时复原为其原始值。 否则，无法删除 HA RDQM 队列管理器。

1. 除去每个节点上的 HA 配置。 要除去 HA，请备份队列管理器，然后将其删除，请参阅 第 550 页的『备份和复原 IBM MQ 队列管理器数据』 和 第 472 页的『删除 HA RDQM』，然后除去 HA 组本身，请参阅 第 471 页的『删除 Pacemaker 集群 (HA 组)』。
2. 使用新的 IP 地址重新创建 HA 配置，请参阅 第 468 页的『定义 Pacemaker 集群 (HA 组)』。
3. 重新创建 HA 队列管理器并复原备份，请参阅 第 471 页的『创建 HA RDQM』 和 第 550 页的『备份和复原 IBM MQ 队列管理器数据』。

Linux 更换高可用性配置中发生故障的节点

如果 HA 组中的某个节点发生故障，那么可以将其替换。

关于此任务

更换节点要执行的步骤取决于方案:

- 如果要将发生故障的节点更换为具有相同配置的节点，那么可以在不中断 HA 组的情况下更换该节点。
- 如果新节点具有其他配置，那么必须删除然后重新构建 HA 组。 您可以首先从正在运行的节点备份队列管理器，然后在重建 HA 组之后复原这些队列管理器。

过程

- 如果将替换节点配置为类似于发生故障的节点 (相同的主机名，相同的 IP 地址等)，请在新节点上完成以下步骤:
  a) 创建与其他节点上的文件匹配的 rdqm.ini 文件，然后运行 rdqmadm -c 命令 (请参阅 第 468 页的『定义 Pacemaker 集群 (HA 组)』)。
  b) 运行 crtmqm -sxs *qmanager* 命令以重新创建每个复制的数据队列管理器 (请参阅 第 471 页的『创建 HA RDQM』)。
- 如果更换节点与发生故障的节点具有不同的配置:
  a) 如果需要，请备份队列管理器 (请参阅 第 550 页的『备份和复原 IBM MQ 队列管理器数据』)。
  b) 使用 **dltmqm** 命令从 HA 组中的其他节点删除复制的数据队列管理器 (请参阅 第 472 页的『删除 HA RDQM』)。
  c) 使用 **rdqmadm -u** 命令取消配置 Pacemaker 集群 (请参阅 第 471 页的『删除 Pacemaker 集群 (HA 组)』)。
  d) 使用 **rdqmadm -c** 命令重新配置 Pacemaker 集群，包括新节点的信息 (请参阅 第 468 页的『定义 Pacemaker 集群 (HA 组)』)。
  e) 如果需要 (即，如果您没有对其他节点的 SSH 访问权)，请运行 crtmqm -sxs *qmanager* 命令以在其他节点上重新创建每个复制的数据队列管理器 (请参阅 第 471 页的『创建 HA RDQM』)。
  f) 运行 crtmqm -sx *qmanager* 命令以在替换节点上创建队列管理器。
  g) 如果需要，将数据和配置复原到队列管理器 (请参阅 第 550 页的『备份和复原 IBM MQ 队列管理器数据』)。

MQ Adv. Linux RDQM 灾难恢复

RDQM (复制的数据队列管理器) 在 Linux 平台的子集上可用，并且可以提供灾难恢复解决方案。

请参阅 软件产品兼容性报告 以获取完整详细信息。

您可以创建在一台服务器上运行的灾难恢复队列管理器的主实例，以及充当恢复节点的另一台服务器上的队列管理器的辅助实例。 在队列管理器实例之间复制数据。 如果丢失主队列管理器，那么可以手动使辅助实例进入主实例并启动队列管理器，然后从同一位置恢复工作。 当队列管理器处于辅助角色时，无法启动该队列管理器。 两个节点之间数据的复制由 DRBD 处理。

您可以在主队列管理器和辅助队列管理器之间选择同步和异步数据复制。 如果选择异步选项，那么 IBM MQ PUT 或 GET 之类的操作将完成，并在将事件复制到辅助队列管理器之前返回到应用程序。 异步复制意味着在恢复情况之后，某些消息传递数据可能会丢失。 但是辅助队列管理器将处于一致状态，并且能够立即开始运行，即使它是在消息流稍早的部分启动的。

虽然您可以迁移现有队列管理器以成为 RDQM 队列管理器，但无法将灾难恢复添加到现有队列管理器 (请参阅 第 493 页的『迁移队列管理器以成为 DR RDQM 队列管理器』)。

您可以在多个不同服务器上运行多对 RDQM 队列管理器。 例如，您可以使主灾难恢复队列管理器在不同节点上运行，而其所有辅助灾难恢复队列管理器都在同一节点上运行。 下图中说明了一些示例配置。

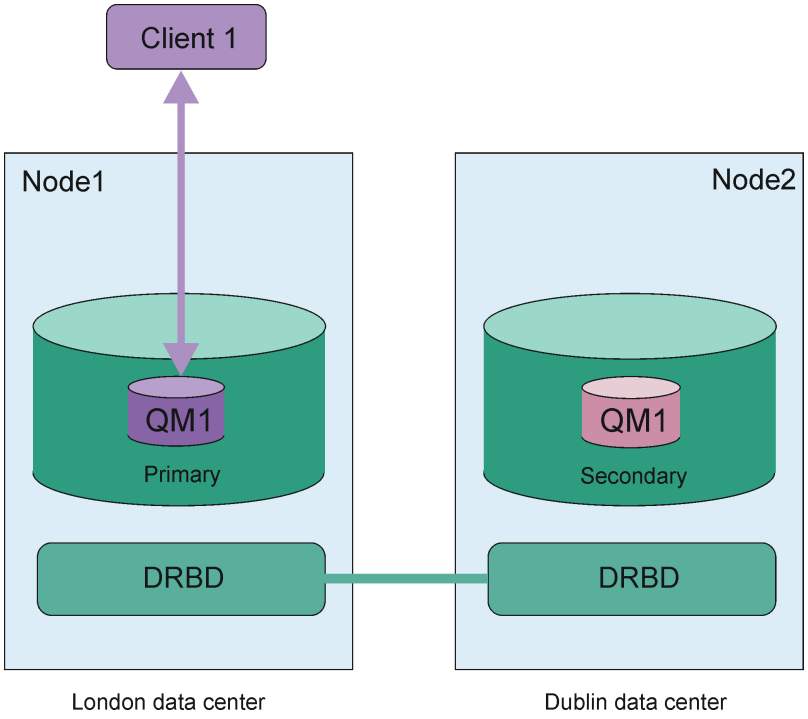


图 *84:* 单个 *RDQM* 对

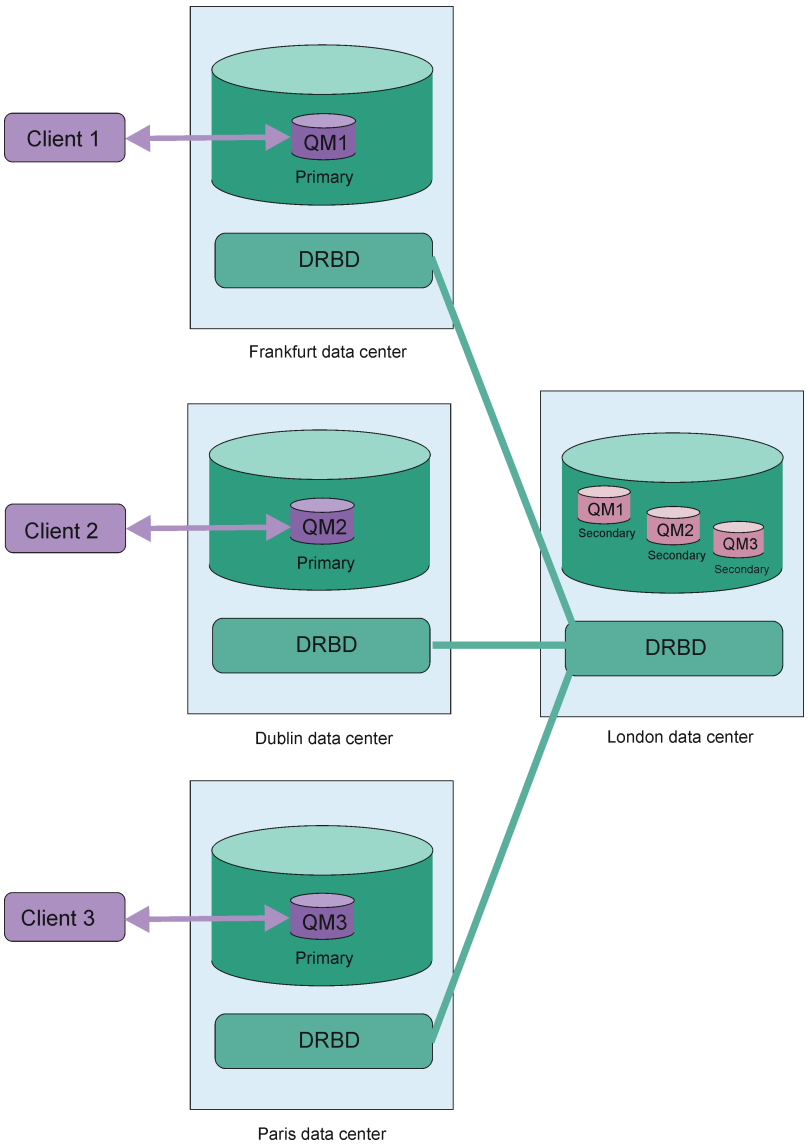


图 *85:* 同一节点中的辅助队列管理器

复制，同步和快照

在连接灾难恢复配置中的两个节点时，会将对灾难恢复队列管理器的持久数据的任何更新从队列管理器的主实例传输到辅助实例。 这称为 **复制**。

如果两个节点之间的网络连接丢失，那么将跟踪对队列管理器主实例的持久数据所作的更改。 复原网络连接时，将使用不同的进程来使辅助实例尽快达到速度。 这称为 **同步**。

同步正在进行时，辅助实例上的数据处于不一致状态。 生成辅助队列管理器数据状态的 **快照** 。 如果主节点或网络连接在同步期间发生故障，那么辅助实例将还原到此快照，并且可以启动队列管理器。 但是，将丢失自原始网络故障以来发生的任何更新。

分区数据 (脑裂)

DR RDQM 配置要求在丢失队列管理器的主实例后执行用户操作，以在恢复节点上提升和运行辅助实例。 提升辅助实例的人员 (或其他人员) 负责确保停止先前的主队列管理器。 如果原始主项保持运行，那么它可能会处理消息，并且在恢复正常操作时，队列管理器的两个实例具有不同的数据视图。 这称为分区或裂脑状态。

请考虑以下情境：

- 运行主队列管理器的节点完全失败。 您将辅助实例提升为主实例; 无法执行操作来停止原始主实例，因为它未在运行。 当修复或更换原始节点时，该节点上的队列管理器最初将成为辅助节点，并与恢复节点上的主队列管理器同步。 然后，将撤销这两个队列管理器的角色，并重新开始正常操作。 在此情况下，唯一潜在的数据丢失是在节点发生故障之前主节点尚未完成复制到辅助节点的任何数据。
- 存在影响运行队列管理器的主实例和辅助实例的节点之间的复制链路的网络故障。 在此情况下，必须确保先停止原始主项，然后再提升辅助项。 如果原始主实例仍具有其他网络连接，那么实际上有两个主实例同时运行，并且可能会产生分区数据。 (如果复制链接正在工作，那么如果主实例仍在运行，那么无法提升辅助队列管理器，该命令将失败。)
- 在运行队列管理器的主实例的节点上发生了完全网络故障。 同样，必须确保先停止主实例，然后再提升辅助实例。 如果复原网络时前一个主实例仍在运行，那么将有两个主实例，并且将再次生成分区数据。

执行受管故障转移时，不应看到队列管理器实例的 DR 状态为 `partitioned` 。 受管故障转移将结束主节点上的队列管理器，然后在完全复制数据后启动恢复节点上的队列管理器。 不期望出现分区状态，因为队列管理器已结束，并且在恢复节点上启动数据之前，数据已在节点之间同步。 如果在恢复节点上启动队列管理器，而节点之间的连接丢失，那么当连接丢失时，如果队列管理器在主节点上处于活动状态，那么可能存在数据偏差。 在此场景中，由于队列管理器数据未同步，一旦恢复连接，将会报告分区状态。 如果发生分区状态，那么您可能必须检查这两个数据集，并就要保留的集作出明智的决策。 请参阅 第 506 页的『解决 DR RDQM 中的分区 (脑裂) 问题』。

Linux *RDQM DR 解决方案的需求*

在配置 RDQM 灾难恢复 (DR) 队列管理器对之前，必须满足许多需求。

系统需求

在配置 RDQM DR 之前，必须在要托管 RDQM DR 队列管理器的每个服务器上完成一些配置。

- 每个节点都需要一个名为 `drbdpool` 的卷组。 每个灾难恢复复制数据队列管理器 (DR RDQM) 的存储器将作为此卷组中每个队列管理器的两个单独逻辑卷进行分配。 (每个队列管理器都需要两个逻辑卷来支持还原为快照操作，因此每个 DR RDQM 分配的存储空间仅为创建时指定的存储空间的两倍。) 为了获得最佳性能，此卷组应由一个或多个对应于内部磁盘驱动器 (最好是 SSD) 的物理卷组成。
- 创建 `drbdpool` 卷组后，不执行其他任何操作。 IBM MQ 管理在 `drbdpool` 中创建的逻辑卷以及安装这些逻辑卷的方式和位置。
- 每个节点都需要一个用于数据复制的接口。 考虑到所有复制的数据队列管理器的预期工作负载，这应该有足够的带宽来支持复制需求。

  对于最大容错，此接口应该是独立的网络接口卡 (NIC)。
- DRBD 要求用于 RDQM 的每个节点都具有有效的因特网主机名 (`uname -n` 返回的值) ，如 RFC 1123 修正的 RFC 952 所定义。
- 如果用于 DR RDQM 的节点之间存在防火墙，那么防火墙必须允许用于复制的端口上的节点之间的流量。 提供了样本脚本 `/opt/mqm/samp/rdqm/firewalld/configure.sh`，用于在 RHEL 中运行标准防火墙时打开必需的端口。 必须以 `root` 用户身份运行脚本。 如果您正在使用其他一些防火墙，请检查服务定义 `/usr/lib/firewalld/services/rdqm*` 以查看需要打开哪些端口。 此脚本为 DRBD 和 IBM MQ 添加以下永久 firewallD 服务规则 (如果您未使用 HA ，那么可以编辑此脚本以省略 Pacemaker 端口):
  - MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-drbd.xml 允许 TCP 端口 7000-7100。
  - MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-mq.xml 允许 TCP 端口 1414 (如果需要其他端口，那么必须编辑脚本)
- 如果系统以非许可方式使用 SELinux ，那么必须运行以下命令:

```
semanage permissive -a drbd_t
```

网络要求

建议您在不同的数据中心内找到用于灾难恢复的节点。

您应该了解以下限制:

- 随着数据中心之间等待时间的增加，性能会快速下降。 对于同步复制， IBM 将支持最长为 5 毫秒的等待时间，对于异步复制，将支持最长为 100 毫秒的等待时间。
- 通过复制链路发送的数据不受使用 IBM MQ AMS 时可能存在的任何其他加密的限制。
- 配置 RDQM 队列管理器以进行灾难恢复会产生开销，因为需要在两个 RDQM 节点之间复制数据。 与异步复制相比，同步复制会产生更大的开销。 使用同步复制时，将阻塞磁盘 I/O 操作，直到将数据写入两个节点。 使用异步复制时，必须先将数据写入主节点，然后才能继续处理。

使用队列管理器的用户需求

要创建，删除或配置复制的数据队列管理器 (RDQM) ，您必须是 root 用户，或者具有属于 mqm 组的用户标识，该用户标识被授予对以下命令的 sudo 权限:

- **crtmqm**
- **dltmqm**
- **rdqmdr**

属于 mqm 组的用户可以使用以下命令来查看 DR RDQM 的状态和状态:

- **dspmq**
- **rdqmstatus**

mqm 用户在两台服务器上都必须具有相同的 UID ，而 mqm 组在两台服务器上都必须具有相同的 GID。

Linux 创建灾难恢复 *RDQM*

您可以使用 **crtmqm** 命令来创建复制的数据队列管理器 (RDQM) ，以充当灾难恢复配置中的主数据队列管理器或辅助数据队列管理器。

关于此任务

如果用户可以使用 sudo ，那么可以在 mqm 组中以用户身份创建复制的数据队列管理器 (RDQM)。 否则，必须以 root 用户身份创建 RDQM。

必须在一个节点上创建主 RDQM DR 队列管理器。 然后，必须在另一个节点上创建同一队列管理器的辅助实例。 主实例和辅助实例必须具有相同的名称，并且分配有相同的存储量。

过程

- 要创建主 DR RDQM:
  a) 输入以下命令:

```
crtmqm -rr p [-rt (a | s)] -rl Local_IP -ri Recovery_IP -rn Recovery_Name -rp Port
[other_crtmqm_options] [-fs size] QMname
```

其中:

-rr p
指定您要创建队列管理器的主实例。

-rt a | s
-rt s 指定 DR 配置使用同步复制， **-rt a** 指定 DR 配置使用异步复制。 异步复制是缺省值。

-rl 本地 *IP*
指定要用于此队列管理器的 DR 复制的本地 IP 地址。

-ri *Recovery_IP*
指定用于在主管队列管理器的辅助实例的服务器上进行复制的接口的 IP 地址。

-rn 恢复名称
指定主管队列管理器的辅助实例的系统的名称。 该名称是在该服务器上运行 uname -n 时返回的值。 必须在该服务器上显式创建辅助队列管理器。

-rp 端口
指定要用于 DR 复制的端口。

other_crtmqm_options
您可以选择指定以下一个或多个常规 **crtmqm** 选项:

- -z
- -q
- -c 文本
- -d *DefaultTransmission* 队列
- -h *MaxHandles*
- -g *ApplicationGroup*
- -oa 用户|组
- -t *TrigInt*
- -u *DeadQ*
- -x *MaxUMsgs*
- -lp *LogPri*
- -ls *LogSec*
- -l-l
- -拉 | -lln
- -lf *LogFile* 大小
- -p 端口

-fs 大小
(可选) 指定要为队列管理器创建的文件系统的大小，即在 drbdpool 卷组中创建的逻辑卷的大小。还会创建该大小的另一个逻辑卷，以支持还原为快照操作，因此 DR RDQM 的总存储空间仅为此处指定的两倍多。

QMNAME
指定复制的数据队列管理器的名称。 此名称区分大小写。

该命令完成后，它将输出您需要在辅助节点上输入的命令，以创建队列管理器的辅助实例。 您还可以在主节点上使用 **rdqmdr** 命令来检索需要在辅助节点上运行以创建辅助队列管理器的 **crtmqm** 命令，请参阅 第 498 页的『管理 DR RDQM 的主要和次要特征』。

- 要创建辅助 DR RDQM:

  a) 在要托管 RDQM 辅助实例的节点上输入以下命令:

```
crtmqm  -rr s [-rt (a | s)] -rl Local_IP -ri Primary_IP -rn Primary_Name  -rp Port
[other_crtmqm_options] [-fs size] QMname
```

其中:

-rr s
指定要创建队列管理器的辅助实例。

-rt a | s
-rt s 指定 DR 配置使用同步复制， **-rt a** 指定 DR 配置使用异步复制。

-rl 本地 *IP*
指定要用于此队列管理器的 DR 复制的本地 IP 地址。

-ri *Primary_IP*
指定用于在主管队列管理器主实例的服务器上进行复制的接口的 IP 地址。

-rn 主名称
指定主管队列管理器主实例的系统的名称。 该名称是在该服务器上运行 uname -n 时返回的值。

-rp 端口
指定要用于 DR 复制的端口。

other_crtmqm_options
您可以选择指定以下一个或多个常规 **crtmqm** 选项:

– -z

-fs 大小
指定要为队列管理器创建的文件系统的大小，即在 drbdpool 卷组中创建的逻辑卷的大小。 如果在创建主队列管理器时指定了非缺省大小，那么必须在此处指定相同的值。

QMNAME
指定复制的数据队列管理器的名称。 这必须与您为队列管理器的主实例指定的名称相同。 请注意，该名称区分大小写。

下一步做什么

创建队列管理器的主实例和辅助实例后，必须检查两个节点上的状态以检查这两个节点是否都正确。 在两个节点上都使用 **rdqmstatus** 命令。 节点应显示正常状态，如 第 499 页的『查看 DR RDQM 状态』中所述。 如果未显示此状态，请删除辅助实例并重新创建该实例，同时注意使用正确的参数。

相关参考
克特姆

Linux 删除 *DR RDQM*
您可以使用 **dltmqm** 命令来删除灾难恢复复制数据队列管理器 (RDQM)。

关于此任务

必须运行此命令以在 RDQM 的主节点和辅助节点上删除 RDQM。 必须先结束 RDQM。 如果该用户具有必需的 sudo 特权，那么可以 mqm 用户身份运行该命令。 否则，必须以 root 用户身份运行该命令。

过程

- 要删除 DR RDQM ， 请输入以下命令:

```
dltmqm RDQM_name
```

相关参考
德尔特姆

MQ Adv. Linux 迁移队列管理器以成为 *DR RDQM* 队列管理器
您可以迁移现有队列管理器以成为灾难恢复 (DR) 复制数据队列管理器 (RDQM) ， 方法是备份其持久数据，然后将数据复原到具有相同名称的新创建的 RDQM 队列管理器。

关于此任务

DR 复制的数据队列管理器需要专用逻辑卷 (文件系统) 和磁盘复制配置。 仅当创建新的队列管理器时，才会配置这些组件。 可以迁移现有队列管理器以使用 RDQM ， 方法是备份其持久数据，然后将数据复原到具有相同名称的新创建的 RDQM 队列管理器。 此过程将在创建备份时保留队列管理器配置，状态和持久消息。

注: 只能从与 RDQM 安装版本相同或更低的 IBM MQ 版本迁移队列管理器。 操作系统和体系结构也必须相同。 否则，必须在目标平台上创建新的队列管理器，请参阅 将队列管理器移至其他操作系统。

在迁移队列管理器之前，应满足以下条件:

- 评估灾难恢复需求并参阅 第 487 页的『RDQM 灾难恢复』。
- 查看连接到队列管理器的应用程序和队列管理器。 请考虑将连接路由到正在运行队列管理器的 RDQM 节点所需的更改。
- 为所选配置供应或标识现有 RDQM 节点。 有关 RDQM 的系统需求的信息，请参阅 第 490 页的『RDQM DR 解决方案的需求』。
- 在每个节点上安装 IBM MQ Advanced (包括 RDQM 功能部件)。
- (可选) 使用测试队列管理器验证 RDQM 配置，然后可以将其删除。 建议先测试配置以确定并解决任何问题，然后再迁移队列管理器。
- 请查看队列管理器的安全性配置，然后在每个 RDQM 节点上复制所需的本地用户和组。

- 查看队列管理器和通道配置以确定是否使用了 API 出口，通道出口或数据转换出口。 在每个 RDQM 节点上安装必需的出口。
- 查看已定义的任何队列管理器服务，然后在每个 RDQM 节点上安装和配置所需的进程。

过程

1. 备份现有队列管理器:
   a) 通过发出等待关闭命令 `endmqm -w` 或立即关闭命令 `endmqm -i` 来停止现有队列管理器。 此步骤对于确保备份中的数据一致非常重要。
   b) 通过查看 IBM MQ 配置文件 `mqs.ini` 来确定队列管理器数据目录的位置。 在 Linux 上，此文件位于 `/var/mqm` 目录中。 有关 `mqs.ini` 的更多信息，请参阅 第 72 页的『IBM MQ 配置文件，mqs.ini』。

      在文件中找到队列管理器的 `QueueManager` 节。 如果节包含名为 `DataPath` 的密钥，那么其值为队列管理器数据目录。 如果该键不存在，那么可以使用 `Prefix` 和 `Directory` 键的值来确定队列管理器数据目录。 队列管理器数据目录是这些值的并置，格式为 *prefix*/qmgrs/*directory*。 有关 QueueManager 节的更多信息，请参阅 第 81 页的『mqs.ini 文件的 QueueManager 节』。
   c) 创建队列管理器数据目录的备份。 在 Linux 上，可以使用 **tar** 命令来执行此操作。 例如，要备份队列管理器的数据目录，可以使用以下命令。 注意命令的最后一个参数，它是一个句点 (dot):

      ```
      tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
      ```
   d) 通过查看 IBM MQ 队列管理器配置文件 `qm.ini` 来确定队列管理器日志目录的位置。 此文件位于队列管理器数据目录中。 有关该文件的更多信息，请参阅 第 82 页的『队列管理器配置文件，qm.ini』。

      队列管理器日志目录定义为 `Log` 节中 `LogPath` 键的值。 有关该节的信息，请参阅 第 108 页的『qm.ini 文件的日志节』。
   e) 创建队列管理器日志目录的备份。 在 Linux 上，可以使用 tar 命令来执行此操作。 例如，要备份队列管理器的日志目录，可以使用以下命令。 注意命令的最后一个参数，它是一个句点 (dot):

      ```
      tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
      ```
   f) 如果队列管理器使用的任何证书存储库不在队列管理器数据目录中，请创建这些存储库的备份。 确保同时备份密钥数据库文件和密码隐藏文件。 有关队列管理器密钥存储库的信息，请参阅 SSL/TLS 密钥存储库 和 查找队列管理器的密钥存储库。 有关在队列管理器配置为使用 AMS 消息通道代理程序 (MCA) 拦截时查找 AMS 密钥库的信息，请参阅 消息通道代理程序 (MCA) 拦截。
   g) 不再需要现有队列管理器，因此可以将其删除。 但是，在可能的情况下，仅应在目标系统上成功复原现有队列管理器后将其删除。 延迟删除可确保在迁移过程未成功完成时可以重新启动队列管理器。

      注: 如果延迟删除现有队列管理器，请勿将其重新启动。 由于在迁移期间丢失了对队列管理器配置或状态的进一步更改，因此队列管理器保持结束很重要。
2. 准备主 RDQM 节点:
   a) 创建与您备份的队列管理器同名的新 RDQM 队列管理器。 请确保 **crtmqm** 为 RDQM 队列管理器分配的文件系统足够大，足以包含现有队列管理器的数据，主日志和辅助日志，以及用于将来扩展的一些额外空间。 有关如何创建 RDQM 队列管理器的信息，请参阅 第 491 页的『创建灾难恢复 RDQM』。
   b) 确定队列管理器的主 RDQM 节点。 有关如何确定主节点的信息，请参阅 rdqmstatus (display RDQM status)。
   c) 在主 RDQM 节点上，如果 RDQM 队列管理器已启动，请使用 `endmqm -w` 或 `endmqm -i` 命令将其停止。
   d) 确定 RDQM 队列管理器的数据和日志目录的位置 (使用步骤 1b 和 1d 中描述的方法)。
   e) 删除 RDQM 队列管理器数据和日志目录的内容，但不删除目录本身。
3. 在主 RDQM 节点上复原队列管理器:
   a) 将队列管理器数据和日志目录的备份复制到主 RDQM 节点，以及队列管理器所使用的证书存储库的任何单独备份。

b) 将队列管理器数据目录的备份复原到新的 RDQM 队列管理器的空数据目录，确保保留文件所有权和许可权。 如果备份是使用步骤 1c 中的示例 tar 命令创建的，那么 root 用户可以使用以下命令来复原备份:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

c) 将队列管理器日志目录的备份复原到新的 RDQM 队列管理器的空日志目录，确保保留文件所有权和许可权。 如果备份是使用步骤 1e 中的示例 tar 命令创建的，那么 root 用户可以使用以下命令来将其复原:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

d) 在 RDQM 队列管理器的数据目录中编辑复原的队列管理器配置文件 `qm.ini`。 更新 `Log` 节中 `LogPath` 键的值，以指定 RDQM 队列管理器的日志目录。

查看配置文件中定义的其他文件路径，并在必要时进行更新。 例如，您可能需要更新以下路径:

- 诊断消息服务生成的错误日志文件的路径。
- 队列管理器所需的出口的路径。
- 如果队列管理器是 XA 事务协调程序，那么这是交换机装入文件的路径。

e) 如果队列管理器配置为使用 AMS 消息通道代理程序 (MCA) 拦截，请将 AMS 密钥库复制到新的 RDQM 安装，然后复审并更新配置。 密钥库必须在每个 RDQM 节点上都可用，因此如果它不在队列管理器的复制文件系统中，那么必须改为将其复制到每个节点。 有关更多信息，请参阅 消息通道代理程序 (MCA) 拦截。

f) 验证 **dspmq** 命令是否显示了队列管理器，并且其状态是否报告为 "已结束"。 以下示例显示了 RDQM DR 队列管理器的样本输出:

```
$ dspmq -o status -o dr

QMNAME(QM1) STATUS(Ended normally) DRROLE(Primary)
```

g) 通过使用 **rdqmstatus** 命令显示队列管理器的状态，验证复原的队列管理器数据是否已复制到辅助 RDQM 节点。 DR 状态应在每个节点上报告为 `Normal` 。 以下示例显示了 RDQM DR 队列管理器的样本输出:

```
$ rdqmstatus -m QM1
Queue manager status:                   Ended normally
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Normal
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
```

h) 在主 RDQM 节点上启动队列管理器。

i) 连接到队列管理器并更新 SSLKEYR 队列管理器属性的值，以指定队列管理器证书存储库的新位置。 缺省情况下，此属性的值设置为 *queue_manager_data_directory*`/ssl/key`。 证书存储库必须位于每个 RDQM 节点上的相同位置。 如果存储库未位于队列管理器的复制文件系统中，那么必须改为将其复制到每个节点。

j) 查看队列管理器的 IBM MQ 对象定义，并更新引用已更改的网络设置， IBM MQ 安装目录或队列管理器数据目录 (包括以下对象) 的对象属性值:

- 侦听器使用的本地 IP 地址 (IPADDR 属性)。
- 通道使用的本地 IP 地址 (LOCLADDR 属性)。
- 为集群接收方通道定义的本地 IP 地址 (CONNAME 属性)。
- 为通信信息对象定义的本地 IP 地址 (GRPADDR 属性)。
- 为流程和服务对象定义定义的系统路径。

k) 停止并重新启动队列管理器以确保更改生效。

l) 对连接到已迁移队列管理器的远程队列管理器以及应用程序的等效设置重复步骤 3j ，包括:

- 通道连接名称 (CONNAME 属性)。
- 通道认证规则，根据其 IP 地址或主机名来限制来自队列管理器的入站连接。
- 客户机通道定义表 (CCDT)，域名设置 (DNS)，网络路由或等效连接信息。

m) 执行队列管理器到每个 RDQM 节点的受管故障转移，以确保已成功建立所需的配置，请参阅 第 503 页的『切换到恢复节点』。

调整 *DR RDQM* 队列管理器的文件系统大小
要调整现有灾难恢复 (DR) 复制数据队列管理器 (RDQM) 的文件系统大小，请备份其持久数据，然后将数据复原到具有相同名称但具有不同大小的文件系统的新创建的 RDQM 队列管理器。

关于此任务

DR 复制的数据队列管理器需要专用逻辑卷 (文件系统) 和磁盘复制配置。 仅当创建新的队列管理器时，才会配置这些组件。 文件系统在创建后无法调整大小，因为它在每个节点上必须具有相同的大小。 要调整现有复制数据队列管理器 (RDQM) 的文件系统大小，您可以备份其持久数据，然后将数据复原到具有相同名称但具有不同大小的文件系统的新创建的 RDQM 队列管理器。 此过程将在创建备份时保留队列管理器配置，状态和持久消息。

过程

1. 备份主 RDQM 节点上的现有 RDQM 队列管理器:
   a) 确定队列管理器的主 RDQM 节点。 有关如何确定主节点的信息，请参阅 rdqmstatus (display RDQM status)。
   b) 在主 RDQM 节点上，如果 RDQM 队列管理器已启动，请使用 **endmqm -w** 或 **endmqm -i** 命令将其停止。
   c) 通过查看 IBM MQ 配置文件 `mqs.ini` 来确定队列管理器数据目录的位置。 在 Linux 上，此文件位于 `/var/mqm` 目录中。 有关 `mqs.ini` 的更多信息，请参阅 第 72 页的『IBM MQ 配置文件，mqs.ini』。

      在文件中找到队列管理器的 `QueueManager` 节。 队列管理器数据目录是名为 `DataPath` 的键的值。 有关 `QueueManager` 节的更多信息，请参阅 第 81 页的『mqs.ini 文件的 QueueManager 节』。
   d) 创建队列管理器数据目录的备份。 在 Linux 上，可以使用 **tar** 命令来执行此操作。 例如，要备份队列管理器的数据目录，可以使用以下命令。 请注意命令的最后一个参数，它是单个句点 (.) 字符:

      ```
      tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
      ```
   e) 通过查看 IBM MQ 队列管理器配置文件 `qm.ini` 来确定队列管理器日志目录的位置。 此文件位于队列管理器数据目录中。 有关该文件的更多信息，请参阅 第 82 页的『队列管理器配置文件，qm.ini』。

      队列管理器日志目录定义为 Log 节中 `LogPath` 键的值。 有关该节的信息，请参阅 第 108 页的『qm.ini 文件的日志节』。
   f) 创建队列管理器日志目录的备份。 在 Linux 上，可以使用 **tar** 命令来执行此操作。 例如，要备份队列管理器的日志目录，可以使用以下命令。 请注意命令的最后一个参数，它是单个句点 (.) 字符:

      ```
      tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
      ```
   g) 删除现有 RDQM 队列管理器。
2. 使用所需大小的文件系统复原队列管理器:
   a) 创建与您备份的队列管理器同名的新 RDQM 队列管理器。 确保按 **crtmqm** 为 RDQM 队列管理器分配的文件系统是您需要的大小，并且它足够大，可以包含现有队列管理器的数据，主日志和辅助日志，以及用于将来扩展的一些额外空间。 有关如何创建 RDQM 队列管理器的信息，请参阅 第 491 页的『创建灾难恢复 RDQM』。
   b) 确定队列管理器的主 RDQM 节点。 有关如何确定主节点的信息，请参阅 rdqmstatus (display RDQM status)。

c) 在主 RDQM 节点上，如果 RDQM 队列管理器已启动，请使用 **endmqm -w** 或 **endmqm -i** 命令将其停止。

d) 在主 RDQM 节点上，确定 RDQM 队列管理器的数据和日志目录的新位置 (使用步骤 1c 和 1e 中描述的方法)。

e) 在主 RDQM 节点上，删除 RDQM 队列管理器数据和日志目录的内容，但不删除目录本身。

f) 在主 RDQM 节点上，将队列管理器数据目录的备份复原到新的 RDQM 队列管理器的空数据目录，确保保留文件所有权和许可权。 如果备份是使用步骤 1d 中的示例 **tar** 命令创建的，那么 root 用户可以使用以下命令来复原备份:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

g) 在主 RDQM 节点上，将队列管理器日志目录的备份复原到新的 RDQM 队列管理器的空日志目录，确保保留文件所有权和许可权。 如果备份是使用步骤 1f 中的示例 **tar** 命令创建的，那么 root 用户可以使用以下命令来复原备份:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

h) 在主 RDQM 节点上，编辑新 RDQM 队列管理器的数据目录中复原的队列管理器配置文件 `qm.ini`。更新 `Log` 节中 `LogPath` 键的值，以指定在步骤 2d 中确定的新 RDQM 队列管理器的日志目录。 查看配置文件中定义的其他文件路径，并在必要时进行更新。 例如，您可能需要更新以下路径:

- 诊断消息服务生成的错误日志文件的路径。
- 队列管理器所需的出口的路径。
- 如果队列管理器是 XA 事务协调程序，那么这是交换机装入文件的路径。

i) 验证 **dspmq** 命令是否显示了队列管理器，并且其状态是否报告为 `ended`。 以下示例显示了 RDQM DR 队列管理器的样本输出:

```
$ dspmq -o status -o dr
QMNAME(QM1) STATUS(Ended normally) DR(Primary)
```

j) 通过使用 **rdqmstatus** 命令显示队列管理器的状态，验证复原的队列管理器数据是否已复制到辅助 RDQM 节点。 DR 状态应在每个节点上报告为 `Normal` 。 以下示例显示主节点上 RDQM DR 队列管理器的样本输出:

```
$ rdqmstatus -m QM1
Queue manager status:                   Running
CPU:                                    0.00
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Normal
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
```

以下示例显示了恢复节点上 RDQM DR 队列管理器的样本输出:

```
Queue manager status:                   Ended immediately
DR role:                                Secondary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.20.2
DR remote IP address:                   192.168.20.1
```

k) 在主 RDQM 节点上启动队列管理器。

l) 执行队列管理器到恢复节点的转换，以确保已成功建立所需的配置，请参阅 第 503 页的『切换到恢复节点』。

V 9.2.0 存储持久应用程序状态

您可以存储与应用程序相关的持久状态信息以及其他队列管理器数据。

每个 IBM MQ 队列管理器都有一个用于其持久状态的专用文件系统，其中包括其队列数据和恢复日志。 在 RDQM 配置中，文件系统由在 Linux 系统 (节点) 之间复制的逻辑卷支持。 文件系统包含可用于存储应用程序的持久状态信息的 `userdata` 目录。 因此，当复制的数据队列管理器移动到 RDQM 配置中的另一个节点上运行时，您可以使用应用程序上下文以及队列管理器上下文。 请参阅 Unix 和 Linux Systems 上的目录内容。

如果选择将应用程序状态存储在 `userdata` 目录中，那么您必须知道写入此位置的数据可能会消耗分配给队列管理器的可用磁盘空间。 必须确保有足够的磁盘空间可供队列管理器写入队列数据，日志和其他持久状态信息。

`userdata` 目录具有 mqm 用户和组所有权，并且具有全局可读性，因此用户无需在 IBM MQ 管理员组 (即 mqm) 中即可访问该目录。 您无法修改 `userdata` 目录的许可权，但可以使用所需的任何所有权和许可权在其中创建内容。

在 RDQM 队列管理器故障转移期间，将结束队列管理器，并在其当前 RDQM 节点上卸载其文件系统。 然后安装文件系统，并在 RDQM 配置中的另一个节点上重新启动队列管理器。 如果进程具有其某个文件的打开句柄，那么无法卸装文件系统。 为了确保队列管理器故障转移可以完成，如果无法卸装队列管理器文件系统，那么将向具有打开文件句柄的进程发送 SIGTERM 信号，如果未释放打开的句柄，那么将后跟 SIGKILL。 您的应用程序必须设计为正确响应 SIGTERM。 如果将应用程序或进程配置为队列管理器服务，那么在受管故障转移期间，可以在卸载文件系统之前在队列管理器关闭期间结束这些应用程序或进程。 如果应用程序或进程未配置为队列管理器服务，或者发生非受管故障转移 (例如丢失定额)，那么很可能将发送信号以释放文件系统。

Linux 管理 *DR RDQM* 的主要和次要特征

可以将辅助灾难恢复复制数据队列管理器 (DR RDQM) 更改为主 DR RDQM。 您还可以将主实例更改为辅助实例。

关于此任务

使用 **rdqmdr** 命令将 RDQM 的辅助实例更改为主实例。 如果由于某种原因丢失了主实例，那么可能需要完成此操作。 然后，可以启动队列管理器并在恢复节点上继续运行该队列管理器。

您还可以使用 **rdqmdr** 命令将 RDQM 的主实例更改为辅助实例。 例如，如果您正在重新配置系统，那么可能需要完成此操作。

您还可以在主队列管理器上使用 **rdqmdr** 来检索在恢复节点上创建该队列管理器的辅助实例所需的确切命令。

如果用户可以使用 sudo，那么可以将 **rdqmdr** 命令用作 mqm 组中的用户。 否则，必须以 root 用户身份登录。

过程

- 要将 DR RDQM 的辅助实例更改为主实例，请输入以下命令:

  ```
  rdqmdr -m QMname -p
  ```

  如果队列管理器的主实例仍在运行并且 DR 复制链接仍在运行，那么此命令将失败。
- 要将队列管理器的主实例更改为辅助实例，请输入以下命令:

  ```
  rdqmdr -m QMname -s
  ```
- 要显示配置队列管理器的辅助实例所需的 **crtmqm** 命令，请在主节点上输入以下命令:

  ```
  rdqmdr -d -m QMname
  ```

  可以在辅助节点上输入返回的 **crtmqm** 命令，以创建 RD RDQM 的辅助实例。

Linux 启动，停止和显示 *DR RDQM* 的状态

您可以使用标准 IBM MQ 控制命令的变体来启动，停止和查看灾难恢复复制数据队列管理器 (DR RDQM) 的当前状态。

关于此任务

必须以属于 mqm 组的用户身份运行用于启动，停止和查看复制数据队列管理器 (RDQM) 的当前状态的命令。

您必须运行命令以在该队列管理器的主节点 (即，队列管理器当前正在其上运行的节点) 上启动和停止队列管理器。

过程

- 要启动 DR RDQM，请在 RDQM 的主节点上输入以下命令:

  ```
  strmqm qmname
  ```

  其中 *qmname* 是要启动的 RDQM 的名称。
- 要停止 RDQM，请在 RDQM 的主节点上输入以下命令:

  ```
  endmqm qmname
  ```

  其中 *qmname* 是要停止的 RDQM 的名称。
- 要查看 RDQM 的状态，请输入以下命令:

  ```
  dspmq -m QMname
  ```

  输出的状态信息取决于您是在 RDQM 的主节点还是辅助节点上运行该命令。 如果在主节点上运行，那么将显示 **dspmq** 返回的其中一条正常状态消息。 如果在辅助节点上运行该命令，那么将显示状态 Ended immediately。 例如，如果在节点 RDQM7 上运行 **dspmq**，那么可能会返回以下信息:

  ```
  QMNAME(DRQM8)                           STATUS(Ended immediately)
  QMNAME(DRQM7)                           STATUS(Running)
  ```

  您可以将自变量与 dspmq 配合使用，以确定是否配置了 RDQM 以进行灾难恢复，以及该 RDQM 当前是主实例还是辅助实例:

  ```
  dspmq -m QMname -o (dr | DR)
  ```

  将显示下列其中一个响应:

  DRROLE()
  指示未配置队列管理器以进行灾难恢复。

  DRROLE(Primary)
  指示队列管理器已配置为主 DR。

  DRROLE(Secondary)
  指示队列管理器已配置为 DR 辅助项。

相关参考

dspmq
恩德姆
strmqm

Linux 查看 *DR RDQM* 状态

您可以查看节点上所有灾难恢复复制数据队列管理器 (DR RDQM) 的状态，或者查看指定 DR RDQM 的详细信息。

关于此任务

您可以使用 **rdqmstatus** 命令来查看所有 DR RDQM 或各个 RDQM 的状态。

V 9.2.3 节点的摘要状态还显示有关 RDQM 所依赖的 DRBD 内核模块的信息。 升级 RDQM 时，请务必确保为系统上运行的 RHEL 内核版本安装了正确版本的 DRBD 内核模块。 状态显示操作系统内核的版本，为其构建 DRBD 模块的内核版本， DRBD 版本和 DRBD 内核模块装入状态。

您必须是 mqm 组中的用户才能运行 **rdqmstatus** 命令。 可以在 DR RDQM 对的任一节点上运行该命令。

过程

- 要查看节点上所有 DR RDQM 的摘要状态，请在该节点上运行以下命令:

```
rdqmstatus
```

显示节点上 DR RDQM 的状态，例如:

```
Node:                                 mqhavm07.exampleco.com
OS kernel version:                    3.10.0-1160.15.2
DRBD OS kernel version:               3.10.0-1160
DRBD version:                         9.1.1
DRBD kernel module status:            Loaded

Queue manager name:                   DRQM8
Queue manager status:                 Ended immediately
DR role:                              Secondary

Queue manager name:                   DRQM7
Queue manager status:                 Running
DR role:                              Primary
```

V 9.2.3 DRBD 内核模块状态为以下值之一:

已加载
指示已装入 DRBD 模块。

已部分加载
装入 DRBD 模块时可能发生，但由于不匹配而无法正常工作。

未加载
未装入 DRBD 模块。 当尚未创建任何 RDQM 队列管理器时，可以在新安装的配置上显示此消息。

未安装
指示未安装 DRBD 模块。 或者 IBM MQ 无法确定 DRBD 模块的操作系统内核版本。

仍加载先前安装的版本
如果在现有 DRBD 模块正在运行 (即 RDQM 队列管理器正在运行) 时安装了新的 DRBD 模块，那么会出现此状态。 新安装的模块在状态中报告，但并非实际正在运行的模块。

- 要查看特定 RDQM 的状态，请输入以下命令:

```
rdqmstatus -m  qmname
```

下表汇总了返回的信息。

表 *35:* 状态属性

状态属性	**可能的值**	**显示时**
队列管理器状态	状态 (由 dspmq 显示)	始终显示
CPU	*n.nn*%	仅当当前节点上的 RDQM 具有主角色时显示
内存	*nnn*MB	仅当当前节点上的 RDQM 具有主角色时显示
队列管理器文件系统	*nnn*MB 已使用， *n.n*GB 已分配 [*n*%]	仅当当前节点上的 RDQM 具有主角色时显示
DR 角色	主 辅助 未知	始终显示

表 35: 状态属性 (继续)		
状态属性	**可能的值**	**显示时**
DR 状态	常规	正常操作
	同步操作正在执行中	正在进行同步。
	分区	在 DR 复制网络不可用时，已在两个节点上启动队列管理器
	远程系统不可用	与另一个节点的连接已丢失
	不一致	同步正在进行中，但已中断
	正在还原到快照	用户已选择还原到队列管理器进入 "不一致" 状态时生成的快照。
	未配置远程系统	已配置 RDQM 的主实例，但未配置任何辅助实例
	协商失败	其中一个节点已设置为同步复制，另一个节点已设置为异步复制
DR 类型	同步或异步	始终显示
DR 端口	*port_number* (用于复制此队列管理器的数据的 TCP/IP 端口)	始终显示
DR 本地 IP 地址	此队列管理器为 DR 复制的本地 IP 地址	始终显示
DR 远程 IP 地址	此队列管理器要为 DR 复制到的远程 IP 地址	始终显示
DR 不同步数据	*n*KB	当远程节点不可用或不一致时显示
DR 同步进度	*n*%	同步正在进行时显示
DR 预计完成时间	*YYYY-MM-DD HH:MM:SS*	同步正在进行时显示
快照还原进度	*n*%	DR 状态为 `Reverting to snapshot` 时显示。 状态计数下降，因此 0% 显示完成
V 9.2.2 V 9.2.2 DR 上次同步时间	*YYYY-MM-DD HH:MM:SS*	在 DR 数据不同步 (在初始同步后) 时显示。 提供上次同步数据的时间和日期。

示例

主节点上的正常状态示例:

```
Queue manager status:                   Running
CPU:                                    0.00
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Normal
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
```

辅助节点上的正常状态示例:

```
Queue manager status:                   Ended immediately
DR role:                                Secondary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.20.2
DR remote IP address:                   192.168.20.1
```

正在进行同步时主节点上的状态示例:

```
Queue manager status:                   Running
CPU:                                    0.53
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Synchronization in progress
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
DR synchronization progress:            11.0%
DR estimated time to completion:        2017-09-06 14:55:05
```

主节点的示例，显示其已分区:

```
Queue manager status:                   Running
CPU:                                    0.02
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Partitioned
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
```

V 9.2.2 主节点的示例，显示它与辅助节点不同步:

```
Queue manager status:                   Running
CPU:                                    0.00
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Remote unavailable
DR type:                                Asynchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
DR out of sync data:                    15932KB
DR last in sync:                        2020-07-27 16:01:47
```

V 9.2.3 显示操作系统内核版本 (RHEL 7.9) 与 DRBD 内核模块 (针对 RHEL 7.8) 不匹配的摘要状态示例。 尽管状态报告已装入 DRBD 内核模块并且期望的队列管理器正在运行，但在此情况下，您应该使用针对正在运行的操作系统内核的版本来更新 DRBD 内核模块。

```
Node:                                   mqhavm07.exampleco.com
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-1127
DRBD version:                           9.1.1
DRBD kernel module status:              Loaded

Queue manager name:                     DRQM8
Queue manager status:                   Ended immediately
DR role:                                Secondary

Queue manager name:                     DRQM7
Queue manager status:                   Running
DR role:                                Primary
```

V 9.2.3 显示操作系统内核版本 (RHEL 7.9) 与 DRBD 内核模块 (针对 RHEL 7.6) 不匹配的摘要状态示例。 在此示例中，版本不匹配更严重，并且 DRBD 内核模块无法成功装入。 QM3 是 DR 队列管理器，本来

应该是主要实例，但由于 DRBD 内核模块未完全装入，因此它报告为次要实例，其 DR 状态为 Unknown。要解决此故障，必须使用正在运行的操作系统内核的版本目标来更新 DRBD 内核模块。

```
Node:                                   mqhavm57.exampleco.com
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-957
DRBD version:                           9.1.2+ptf.3
DRBD kernel module status:              Partially loaded

Queue manager name:                     QM3
Queue manager status:                   Status not available
DR role:                                Secondary
DR status:                              Unknown
```

相关参考

Linux rdqmstatus

Linux 在灾难恢复环境中运行

在许多情况下，您可能希望在灾难恢复配置中切换到辅助队列管理器。

灾难恢复

在主站点上的主队列管理器完全丢失之后，在恢复站点上启动辅助队列管理器。 应用程序重新连接到恢复站点上的队列管理器，辅助队列管理器处理应用程序消息。 还原到先前配置所采取的步骤取决于故障原因。 例如，主节点完全丢失与临时丢失。

要了解在主站点暂时丢失后要执行的步骤，请参阅 第 503 页的『切换到恢复节点』。 有关在发生永久故障后执行的步骤，请参阅 第 504 页的『在灾难恢复配置中替换发生故障的节点』。

灾难恢复测试支持

您可以通过临时切换到辅助实例并检查应用程序是否可以成功连接来测试灾难恢复配置。 您遵循与在主节点发生临时故障后切换时相同的过程，请参阅 第 503 页的『切换到恢复节点』。

正在还原到快照

如果在进行同步时主节点发生故障，那么可以在同步开始之前还原到辅助队列管理器数据的快照。 然后，辅助服务器将恢复到一致状态，并且可以作为主服务器运行。 要还原到快照，请使辅助卷成为主卷，如 第 503 页的『切换到恢复节点』中所述。 在启动队列管理器之前，必须检查是否已完成对快照的还原 (通过使用 **rdqmstatus** 命令)。

Linux 切换到恢复节点

如果主站点发生灾难，请执行步骤以切换到恢复站点。

关于此任务

在主站点上丢失主队列管理器之后，您将恢复站点上的辅助队列管理器设置为主队列管理器并将其启动。 应用程序重新连接到恢复站点上的队列管理器，并且队列管理器处理应用程序消息。 您还可以使用此过程来测试恢复节点。

要点: 在提升原始辅助实例之前，必须确保队列管理器的主实例无法运行，或者已停止并进入辅助实例。 否则，可能会累积分区数据。

您必须以 root 用户身份登录，或者以属于 mqm 组且具有必需 sudo 配置的用户身份登录。

过程

1. 如果使用此过程来测试辅助队列管理器 (即，主实例仍在运行) ，那么必须停止主实例并将其重新指定为辅助实例:

   ```
   endmqm qmname
   rdqmdr -m qmname -s
   ```

2. 通过在恢复节点上输入以下命令，使辅助队列管理器成为主队列管理器:

   ```
   rdqmdr -m qmname -p
   ```

3. 通过输入以下命令来启动队列管理器:

```
strmqm qmname
```

4. 确保应用程序重新连接到恢复队列管理器上的队列管理器。 如果您已使用备用连接名称列表定义通道，并指定主队列管理器和辅助队列管理器，那么应用程序将自动连接到新的主队列管理器。

下一步做什么

复原发生故障的节点时，如果两个节点之间的链路正常工作，那么队列管理器无法在此节点上启动，因为它正在您提升辅助队列管理器实例的恢复节点上运行。 要恢复正常操作，必须停止恢复节点上的队列管理器，然后将原始节点上的队列管理器提升回主角色。

相关参考

strmqm

Rdqmdr

测试恢复 *RDQM* 队列管理器

您可以测试 RDQM 灾难恢复配置中队列管理器的恢复实例是否正常运行，而无需中断主站点。

关于此任务

您可以通过禁用主节点和恢复节点之间的接口来测试恢复队列管理器。 您可以将辅助队列管理器变成主队列管理器，然后测试独立队列管理器。 测试完成后，还原接口并删除测试队列管理器。 然后，在灾难恢复配置中将队列管理器重新创建为辅助队列管理器。

过程

1. 禁用主节点和恢复节点之间的网络连接。
2. 在恢复节点上，将队列管理器设为主节点:

```
rdqmdr -m QMname -p
```

   其中 *qmname* 是队列管理器的名称。
3. 启动队列管理器:

```
strmqm QMname
```

4. 将应用程序连接到队列管理器，并测试它们是否按预期运行。
5. 结束队列管理器:

```
endmqm QMname
```

6. 删除队列管理器:

```
dltmqm QMname
```

7. 恢复主设备和恢复设备之间的网络连接。
8. 在主节点上运行以下命令，检索第一次配置灾难恢复时使用的 "**crtmqm** 命令。

```
rdqmdr -d -m QMname
```

9. 在恢复节点上运行生成的 "**crtmqm** 命令，重新创建辅助队列管理器。 主节点上的主队列管理器会与辅助队列管理器同步数据，以更新数据。

Linux 在灾难恢复配置中替换发生故障的节点

如果在灾难恢复配置中丢失了其中一个节点，那么可以通过遵循此过程来替换该节点并恢复灾难恢复配置。

关于此任务

如果发生灾难，导致主站点中的节点无法修复，那么您可以在队列管理器在恢复节点上运行时替换发生故障的节点，然后复原原始灾难恢复配置。 替换节点必须采用故障节点的身份: 名称和 IP 地址必须相同。

您必须以 root 用户身份登录，或者以属于 mqm 组且具有必需 sudo 配置的用户身份登录。

过程

在主站点上丢失队列管理器之后，请执行以下步骤:

1. 在恢复节点上，运行以下命令以使辅助队列管理器承担主角色:

   ```
   rdqmdr -m QMname -p
   ```

   其中 *qmname* 是队列管理器的名称。
2. 检索需要在替换主节点上运行以重新配置灾难恢复的命令:

   ```
   rdqmdr -m QMname -d
   ```

   复制此命令的输出。
3. 运行以下命令以启动队列管理器:

   ```
   strmqm QMname
   ```
4. 确保应用程序重新连接到恢复节点上的队列管理器。 如果您已使用备用连接名称列表定义通道，并指定主队列管理器和辅助队列管理器，那么应用程序将自动连接到新的主队列管理器。
5. 更换主站点上发生故障的节点，并将其配置为具有与原始节点上用于灾难恢复的名称和 IP 地址相同的名称和 IP 地址。 然后，通过运行在步骤 2 中复制的 **crtmqm** 命令来配置灾难恢复。 现在，您具有队列管理器的辅助实例，并且主实例将其数据与辅助实例同步。
6. 结束当前主实例。
7. 完成同步后，使正在恢复节点上运行的主实例再次进入辅助实例:

   ```
   rdqmdr -m QMname -s
   ```
8. 在替换主节点上，使队列管理器的辅助实例成为主实例:

   ```
   rdqmdr -m QMname -p
   ```
9. 在替换主节点上，启动队列管理器:

   ```
   strmqm QMname
   ```

   现在，您已恢复了在主站点发生故障之前的配置。

相关参考

strmqm
Rdqmdr
恩德姆

解决 *DR RDQM* 中的不一致问题

如果队列管理器的主实例与辅助实例之间的同步失败，那么可以报告 DR 状态 `inconsistent` 。

关于此任务

由于在同步操作期间与主实例的复制连接丢失，因此在队列管理器的辅助实例上报告了不一致的状态。 您可能必须采取行动来解决此情况。 请考虑以下事件序列:

1. DR 主队列管理器与 DR 辅助队列管理器同步
2. 主和辅助之间的复制链路丢失
3. 主和辅助之间的复制链路已复原

4. 当 DR 辅助队列管理器与 DR 主队列管理器同步时，将发生再同步。 在此期间，将针对两个队列管理器报告 `synchronization in progress` 的 DR 状态。
5. 如果在再同步期间复制再次丢失，那么 DR 辅助服务器上的状态将报告为 `Inconsistent`。

如果主管主队列管理器的节点仍在运行，并且可以复原复制链路，那么将自动进行再同步。 在不执行任何操作的情况下，将解决不一致状态。

如果主管主队列管理器的节点不再可操作，那么可以通过在辅助队列管理器上实现还原为快照来解决不一致状态。 此操作会将数据还原为最后一个已知的良好状态。

过程

要解决不一致状态:

1. 在恢复节点上，使辅助实例成为主实例:

   ```
   rdqmdr -m qmname -p
   ```

   "还原为快照" 操作将启动。
2. 在恢复节点上，检查队列管理器的状态，以查看何时完成还原为快照操作:

   ```
   rdqmstatus -m qmname
   ```
3. 当队列管理器状态为 `Normal` 时，启动队列管理器:

   ```
   strmqm qmname
   ```

解决 *DR RDQM* 中的分区 *(*脑裂*)* 问题
如果灾难恢复对中的两个队列管理器同时以主角色运行，那么可能会发生分区问题。

关于此任务

如果在恢复节点上提升了队列管理器的辅助实例，而原始主实例继续在主节点上运行，那么实际上将运行同一队列管理器的两个版本，每个版本都有自己的队列管理器数据视图。 每个节点上队列管理器的 DR 状态报告为 `Partitioned`。

您必须确定两个队列管理器中的哪个具有最正确的数据视图，并在废弃另一个队列管理器时保留该集。 您可以使用 **rdqmdr** 命令来完成此操作。

过程

- 要在恢复节点上保留队列管理器中的数据，请执行以下操作:
  a) 确保两个队列管理器实例都已停止。
  b) 指定主节点上的队列管理器是辅助节点:

  ```
  rdqmdr -m qmname -s
  ```

  c) 指定恢复节点上的队列管理器是主节点:

  ```
  rdqmdr -m qmname -p
  ```

  同步开始，将恢复节点上的队列管理器中的数据复制到主节点。
  d) 检查同步的状态:

  ```
  rdqmstatus -m qmname
  ```

  e) 同步完成后，将恢复节点上的队列管理器降级:

  ```
  rdqmdr -m qmname -s
  ```

  f) 在主节点上提升队列管理器，然后将其启动:

```
rdqmdr -m qmname -p
strmqm qmname
```

- 要在主节点上保留来自队列管理器的数据，请执行以下操作：
  a) 确保两个队列管理器实例都已停止。
  b) 指定恢复节点上的队列管理器是辅助队列管理器：

```
rdqmdr -m qmname -s
```

  c) 指定主节点上的队列管理器是主节点：

```
rdqmdr -m qmname -p
```

  同步开始，将主节点上的队列管理器中的数据复制到恢复节点。
  d) 检查同步的状态：

```
rdqmstatus -m qmname
```

  e) 同步完成后，在主节点上启动队列管理器：

```
strmqm qmname
```

更改灾难恢复配置中的 *IP* 地址
如果更改灾难恢复配置中任一接口的 IP 地址，那么无法在两个节点之间进行复制。

如果需要更改任一 DR 节点的复制接口的 IP 地址，那么必须使用以下过程：

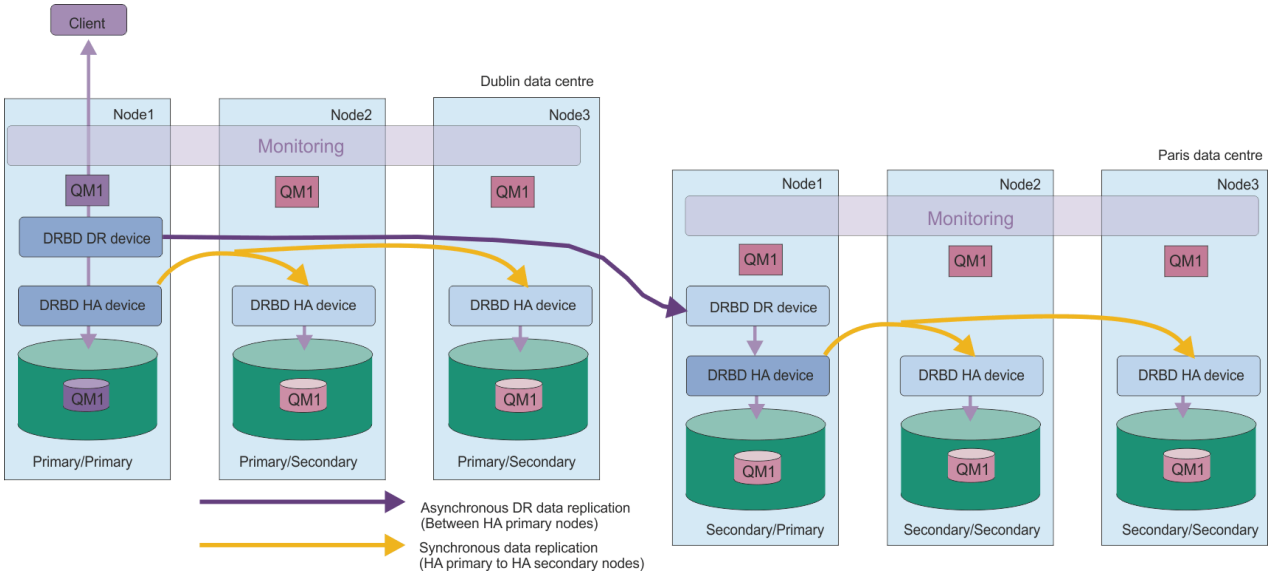
1. 删除每个节点上的 DR 队列管理器。 首先备份队列管理器，然后将其删除，请参阅 第 550 页的『备份和复原 IBM MQ 队列管理器数据』 和 第 493 页的『删除 DR RDQM』。
2. 重新创建 DR 队列管理器，指定新的 IP 地址，并复原备份，请参阅 第 491 页的『创建灾难恢复 RDQM』 和 第 550 页的『备份和复原 IBM MQ 队列管理器数据』。

Linux V 9.2.0 RDQM 灾难恢复和高可用性

您可以配置在一个站点上的高可用性组上运行的复制数据队列管理器 (RDQM)，但如果发生一些导致第一个组不可用的灾难，那么可以故障转移到另一个站点上的另一个高可用性组。 这称为 DR/HA RDQM。

DR/HA RDQM 将高可用性 RDQM (请参阅 第 463 页的『RDQM 高可用性』) 和灾难恢复 RDQM (请参阅 第 487 页的『RDQM 灾难恢复』) 的功能部件组合在一起。

下图显示了示例 DR/HA RDQM。



主站点上的 DR/HA RDQM 与灾难恢复站点之间的复制始终是异步的。 通过异步复制，操作 (例如， IBM MQ PUT 或 GET) 完成，并在将事件复制到辅助队列管理器之前返回到应用程序。

如果需要，您可以有两个活动站点，而不是 "main" 和 "recovery" 站点，因此在正常操作期间，某些 DR/HA RDQMs 在一个站点上运行，而某些站点在另一个站点上运行。 如果发生灾难并且一个站点变得不可用，那么所有 DR/HA RDQM 都在同一站点的同一 HA 组上运行。

每个 HA 组的配置方式与普通 HA 组相同。 可以为每个 HA 组中的 DR/HA RDQM 定义浮动 IP 地址。 对于每个 HA 组，浮动 IP 地址可以相同或不同。

不能将现有 RDQM 升级为 DR/HA RDQM，必须创建 DR/HA RDQM。 (如果需要，您可以备份现有 RDQM 的数据，将其删除，将其重新创建为 DR/HA RDQM，然后复原数据，请参阅 第 550 页的『备份和复原 IBM MQ 队列管理器数据』。)

要配置 DR/HA RDQM，必须完成以下主要步骤:

1. 在 "main" 站点上配置 HA 组。
2. 在 "recovery" 站点上配置 HA 组。
3. 在 "主" 站点中 HA 组的一个节点上创建主/主 DR/HA RDQM。
4. 在 "主" 站点中的其他两个节点上创建主/辅助 DR/HA RDQM。
5. 定义应用程序的浮动 IP 地址以在 "主" 站点上 HA 组的任何节点上运行时访问 DR/HA RDQM。
6. 在 "recovery" 站点上 HA 组的一个节点上创建辅助/主 DR/HA RDQM。
7. 在 "恢复" 站点中的其他两个节点上创建辅助/辅助 DR/HA RDQM。
8. 当应用程序在 "恢复" 站点上的 HA 组的任何节点上运行时，为该应用程序定义浮动 IP 地址以访问 DR/HA RDQM。

以下主题中提供了有关其中每个步骤的详细信息。

Linux V 9.2.0 *DR/HA RDQM* 解决方案的需求

DR/HA RDQM 解决方案的需求与 HA RDQM 解决方案和 DR RDQM 解决方案的需求相同。

有关配置的 HA 部分的需求的详细信息，请参阅 第 465 页的『RDQM HA 解决方案的需求』。

有关配置的 DR 部分的详细信息，请参阅 第 490 页的『RDQM DR 解决方案的需求』。

Linux V 9.2.0 为 *DR/HA RDQM* 配置 *HA* 组

必须在主站点和恢复站点上都创建 HA 组。 如果任一站点上都有现有 HA 组，那么可以在该 HA 组中创建 DR/HA RDQM。 (现有的 RDQM 将一如既往地继续运作。)

此过程与针对 RDQM 高可用性描述的过程相同，请参阅 第 468 页的『定义 Pacemaker 集群 (HA 组)』。

定义高可用性组时，请指定用于由 `rdqm.ini` 文件中的每个节点进行监视和复制的 IP 地址。 创建 HA 组以支持 DR/HA RDQMs 时，还可以指定由您定义的 HA 组用于 DR 复制的 IP 地址，以及由 DR 对的另一个 HA 组中的节点用于 DR 复制的 IP 地址。 (如果未在 `rdqm.ini` 文件中指定 DR 复制 IP 地址，那么可以在创建 DR/HA RDQM 时在命令行中指定这些地址。)

如果要配置现有 HA 组，那么可以将 DR 复制 IP 地址添加到现有 `rdqm.ini` 文件。 更新 `rdqm.ini` 后无需再次运行 **rdqmadm**，但必须先更新 `rdqm.ini`，然后才能创建任何 DR/HA RDQM。

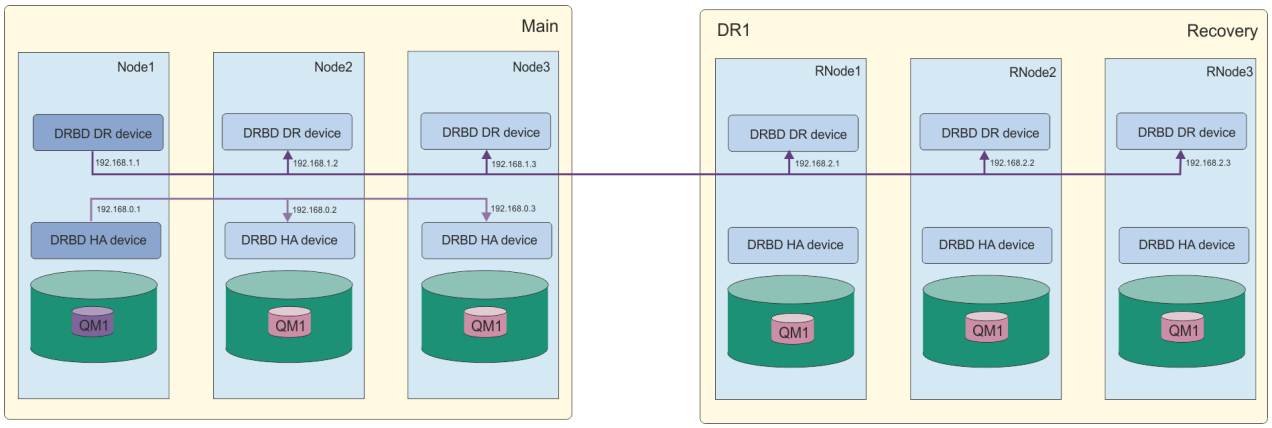
使用 `Node` 节中的 `DR_Replication` 属性来指定要定义的 HA 组上的 DR 复制接口，例如:

```
Node:
  Name=Node1
  HA_Replication=192.168.0.1
  DR_Replication=192.168.1.1
Node:
  Name=Node2
  HA_Replication=192.168.0.2
  DR_Replication=192.168.1.2
Node:
  Name=Node3
  HA_Replication=192.168.0.3
  DR_Replication=192.168.1.3
```

使用 DRGroup 节来指定远程 HA 组的 DR 复制地址，例如：

```
DRGroup:
  Name=DR1
  DR_Replication=192.168.2.1
  DR_Replication=192.168.2.2
  DR_Replication=192.168.2.3
```

下图说明了此配置：



如果在创建 DR/HA RDQM 时未在 rdqm.ini 文件或命令行中为本地 HA 组中的节点指定 DR 复制 IP 地址，那么为每个节点定义的 HA_Replication 接口将用于 DR 复制。 必须在 rdqm.ini 文件或 **crtmqm** 命令行中指定远程 HA 组 DR 复制地址。

Linux V 9.2.0 创建 *DR/HA RDQM*

您可以使用 **crtmqm** 命令在 DR/HA 配置中创建复制的数据队列管理器 (RDQM)。

关于此任务

如果用户可以使用 sudo，那么可以创建 DR/HA RDQM 作为 mqm 组中的用户。 否则，必须以 root 用户身份创建 RDQM。

必须创建多个 DR/HA RDQM：

- 在 "main" 站点上的 HA 组上：
  - 在希望队列管理器在正常情况下运行的节点上，创建主/主 DR/HA RDQM。
  - 在 HA 组中的其他两个节点中的每个节点上，创建主/辅助 DR/HA RDQM。
- 在 "recovery" 站点上的 HA 组上：
  - 在队列管理器将在故障转移到恢复站点时运行的节点上，创建辅助/主 DR/HA RDQM。 在 "main" 站点上创建主/主队列管理器时，可以使用命令输出。
  - 在 HA 组中的其他两个节点中的每个节点上，创建辅助/辅助 DR/HA RDQM。

所有队列管理器实例必须具有相同的名称并分配相同的存储量。

过程

- 要创建主/主 DR/HA RDQM：

  a) 输入以下命令：

```
crtmqm -sx -rr p
              [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
              (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
              -rp DRPort
              [-z] [-q] [-c Text] [-d DefXmitQ] [-h MaxHandles]
              [-g ApplicationGroup] [-oa user|group]
              [-t TrigInt] [-u DeadQ] [-x MaxUMsgs]
              [-lp LogPri] [-ls LogSec]
```

```
              [-lc | -ll | -lla | -lln] [-lf LogFileSize]
              [-p Port] [-fs FilesystemSize] QMgrName
```

其中:

-sx
指示初始 HA 角色是主角色。

-rr p
指示初始 DR 角色是主角色。

-rl *DRLocalIP1*, *DRLocalIP2*, *DRLocalIP3*
(可选) 指定本地站点 (即 "主" 站点) 上三个节点上的 DR 接口的 IP 地址。 如果未指定，那么将使用 `rdqm.ini` 文件中指定的 IP 地址。

-ri *DRRemoteIP1*, *DRRemoteIP2*, *DRRemoteIP3*
指定远程站点 (即 "恢复" 站点) 上三个节点上的 DR 接口的 IP 地址。 必须指定此参数或 `-rn` 参数。

-rn *GroupName*
指定 `rdqm.ini` 文件中指定的远程 HA 组名。 必须指定 `-ri` 或 `-rn`。

-rp 端口
指定要用于 DR 复制的端口。

other_crtmqm_options
您可以选择指定以下一个或多个常规 **crtmqm** 选项:

- -z
- -q
- -c 文本
- -d *DefaultTransmission* 队列
- -h *MaxHandles*
- -g *ApplicationGroup*
- -oa 用户 | 组
- -t *TrigInt*
- -u *DeadQ*
- -x *MaxUMsgs*
- -lp *LogPri*
- -ls *LogSec*
- -l-l
- -拉 | -lln
- -lf *LogFile* 大小
- -p 端口

-fs 大小
(可选) 指定要为队列管理器创建的文件系统的大小，即在 drbdpool 卷组中创建的逻辑卷的大小。 还会创建该大小的另一个逻辑卷，以支持还原为快照操作，因此 DR RDQM 的总存储空间仅为此处指定的两倍多。

QMNAME
指定复制的数据队列管理器的名称。 此名称区分大小写。

该命令完成后，它将输出您可以在恢复站点上输入的命令，以创建队列管理器的辅助/主实例。

- 要在 HA 组中的其他两个节点上创建主/辅助 DR/HA RDQM:

  a) 在每个节点上输入以下命令:

```
crtmqm -sxs -rr p
              [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
              (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
```

```
              -rp DRPort
              [-fs FilesystemSize] QMgrName
```

其中:

-sxs
指示初始 HA 角色是辅助角色。

-rr p
指示初始 DR 角色是主角色。

-rl *DRLocalIP1*, *DRLocalIP2*, *DRLocalIP3*
(可选) 指定本地站点 (即 "主" 站点) 上三个节点上的 DR 接口的 IP 地址。 如果未指定，那么将使用 `rdqm.ini` 文件中指定的 IP 地址。

-ri *DRRemoteIP1*, *DRRemoteIP2*, *DRRemoteIP3*
指定远程站点 (即 "恢复" 站点) 上三个节点上的 DR 接口的 IP 地址。 必须指定此参数或 `-rn` 参数。

-rn *GroupName*
指定 `rdqm.ini` 文件中指定的远程 HA 组名。 必须指定 `-ri` 或 `-rn`。

-rp 端口
指定要用于 DR 复制的端口。

-fs 大小
指定要为队列管理器创建的文件系统的大小，即在 drbdpool 卷组中创建的逻辑卷的大小。 如果在创建主 RDQM/主 RDQM 时指定了非缺省大小，那么必须在此处指定相同的值。

QMNAME
指定主/辅助 RDQM 的名称。 这必须与您为 RDQM 的主/主实例指定的名称相同。 请注意，该名称区分大小写。

- 要在队列管理器故障转移到恢复站点时将运行该队列管理器的节点上创建辅助/主 DR/HA RDQM ，请执行以下操作:

  a) 在主站点上创建主/主 DR/HA 时使用命令输出，或者输入以下命令:

```
crtmqm -sx -rr s
              [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
              (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
              -rp DRPort
              [-fs FilesystemSize] QMgrName
```

-sx
指示初始 HA 角色是主角色。

-rr s
指示初始 DR 角色是辅助角色。

-rl *DRLocalIP1*, *DRLocalIP2*, *DRLocalIP3*
(可选) 指定本地站点 (即 "恢复" 站点) 上三个节点上的 DR 接口的 IP 地址。 如果未指定，那么将使用 `rdqm.ini` 文件中指定的 IP 地址。

-ri *DRRemoteIP1*, *DRRemoteIP2*, *DRRemoteIP3*
指定远程站点 (即 "主" 站点) 上三个节点上的 DR 接口的 IP 地址。 必须指定此参数或 `-rn` 参数。

-rn *GroupName*
指定 `rdqm.ini` 文件中指定的远程 HA 组名。 必须指定 `-ri` 或 `-rn`。

-rp 端口
指定要用于 DR 复制的端口。

-fs 大小
(可选) 指定要为队列管理器创建的文件系统的大小，即在 drbdpool 卷组中创建的逻辑卷的大小。 还会创建该大小的另一个逻辑卷，以支持还原为快照操作，因此 DR RDQM 的总存储空间仅为此处指定的两倍多。

QMNAME
指定复制的数据队列管理器的名称。 此名称区分大小写。

- 要在恢复站点上的其他两个节点上创建辅助/辅助 HA/DR RDQM:

a) 在每个节点上输入以下命令:

```
crtmqm -sxs -rr s
             [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
             (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
             -rp DRPort
             [-fs FilesystemSize] QMgrName
```

-sxs
指示初始 HA 角色是主角色。

-rr s
指示初始 DR 角色是辅助角色。

-rl *DRLocalIP1*, *DRLocalIP2*, *DRLocalIP3*
(可选) 指定本地站点上三个节点上 DR 接口的 IP 地址。 如果未指定，那么将使用 `rdqm.ini` 文件中指定的 IP 地址。

-ri *DRRemoteIP1*, *DRRemoteIP2*, *DRRemoteIP3*
指定远程站点上三个节点上 DR 接口的 IP 地址。 必须指定此参数或 `-rn` 参数。

-rn *GroupName*
指定 `rdqm.ini` 文件中指定的远程 HA 组名。 必须指定 `-ri` 或 `-rn`。

-rp 端口
指定要用于 DR 复制的端口。

-fs 大小
(可选) 指定要为队列管理器创建的文件系统的大小，即在 drbdpool 卷组中创建的逻辑卷的大小。 还会创建该大小的另一个逻辑卷，以支持还原为快照操作，因此 DR RDQM 的总存储空间仅为此处指定的两倍多。

QMNAME
指定复制的数据队列管理器的名称。 此名称区分大小写。

注: 创建 RDQM 时，将为 HA 复制链路分配高于 7000 的下一个可用端口号。 如果发现所选端口由另一个应用程序使用，那么 **crtmqm** 命令将失败并返回 AMQ6543 错误，并且该端口将添加到排除列表中。 必须删除队列管理器的辅助实例，然后再次运行 **crtmqm** 命令。

下一步做什么

创建所有 DR/HA RDQM 后，必须检查主实例/主实例和辅助实例/主实例上的状态以检查所有这些实例是否正确。 在节点上使用 **rdqmstatus** 命令。 节点应显示正常状态，如 第 514 页的『查看 DR/HA RDQM 和 HA 组状态』中所述。 如果未显示此状态，请删除辅助/主实例并重新创建该实例，同时注意使用正确的参数。

相关任务

第 509 页的『创建 DR/HA RDQM』
您可以使用 **crtmqm** 命令在 DR/HA 配置中创建复制的数据队列管理器 (RDQM)。

相关参考

克特姆

Linux V 9.2.0 删除 *DR/HA RDQM*

使用 **dltmqm** 命令可删除 DR/HA 复制的数据队列管理器 (RDQM)。

关于此任务

必须运行该命令以删除主节点/主节点和辅助节点/主节点上的 RDQM。 必须先结束 RDQM。 如果该用户具有必需的 sudo 特权，那么可以 mqm 用户身份运行该命令。 否则，必须以 root 用户身份运行该命令。

过程

- 要删除 DR/HA RDQM ，请输入以下命令:

```
dltmqm RDQM_name
```

相关参考
德尔特姆

Linux V 9.2.0 创建浮动 *IP* 地址

您可以在 DR/HA RDQM 配置中为每个 HA 组创建浮动 IP 地址。

浮动 IP 地址使客户机能够对 DR/HA RDQM 使用相同的 IP 地址，而不考虑运行该地址的 HA 组中的哪个节点。 如果两个 HA 组具有用于应用程序连接的专用/隔离网络，那么可以为这两个组定义相同的浮动 IP 地址。 但是，您仍必须在每个 HA 组上定义该浮动 IP 地址两次。

使用与 HA RDQM 相同的方法来创建和删除浮动 IP 地址。 请参阅第 478 页的『创建和删除浮动 IP 地址』。

Linux V 9.2.0 启动，停止和显示 *DR/HA RDQM* 的状态

使用标准 IBM MQ 控制命令的变体来启动，停止和查看 DR/HA RDQM 的当前状态。

关于此任务

必须以属于 `mqm` 和 `haclient` 组的用户身份运行用于启动，停止和查看 DR/HA RDQM 的当前状态的命令。

您必须运行命令以在该队列管理器的主节点上启动和停止队列管理器。

过程

- 要启动 RDQM，请在 RDQM 的主节点上输入以下命令:

  ```
  strmqm qmname
  ```

  其中 *qmname* 是要启动的 DR/HA RDQM 的名称。

  RDQM 已启动， Pacemaker 开始管理 RDQM。 如果要指定任何其他 `strmqm` 选项，那么必须将 `-ns` 选项与 `strmqm` 一起指定。
- 要停止 RDQM，请在 DR/HA RDQM 的主节点上输入以下命令:

  ```
  endmqm qmname
  ```

  其中 *qmname* 是要停止的 RDQM 的名称。

  Pacemaker 停止管理 RDQM，然后结束 RDQM。 停止 RDQM 时，可以使用所有其他 **endmqm** 参数。
- 要查看 RDQM 的状态，请输入以下命令:

  ```
  dspmq -m QMname
  ```

  输出的状态信息取决于您是在 RDQM 的主节点还是辅助节点上运行该命令。 如果在主节点上运行，那么将显示 **dspmq** 返回的其中一条正常状态消息。 如果在辅助节点上运行该命令，那么将显示状态 `Ended immediately`。 例如，如果在节点 RDQM7 上运行 **dspmq**，那么可能会返回以下信息:

  ```
  QMNAME(DRQM8)                           STATUS(Ended immediately)
  QMNAME(DRQM7)                           STATUS(Running)
  ```

  您可以将自变量与 `dspmq` 配合使用，以确定是否配置了 RDQM 以进行灾难恢复，以及该 RDQM 当前是主实例还是辅助实例:

  ```
  dspmq -m QMname -o (dr | DR)
  ```

  将显示下列其中一个响应:

  DRROLE()
  指示未配置队列管理器以进行灾难恢复。

  DRROLE(Primary)
  指示队列管理器已配置为主 DR。

DRROLE(Secondary)
指示队列管理器已配置为 DR 辅助项。

使用 **dspmq -o all** 命令可查看 DR/HA RDQM 的灾难恢复和高可用性信息。 例如，如果在运行 DR/HA RDQM 的节点上运行 **dspmq -o all**，那么您将看到以下状态信息:

```
QMNAME(TESTQM1)                                           STATUS(Running) HA(Replicated)
DRROLE(Primary)
```

相关参考

dspmq (显示队列管理器)
endmqm (结束队列管理器)
strmqm (启动队列管理器)

V 9.2.2 *DR/HA* 配置中的资源操作失败

当 RDQM 高可用性配置的 Pacemaker 组件在 HA 组中的某个节点上迁到资源问题时，将发生失败的资源操作。

对于 RDQM DR/HA 配置中的任一 HA 配置，可能会发生失败的资源操作。 您可以使用 **rdqmstatus** 命令来查看失败的资源操作，并使用 **rdqmclean** 命令来清除这些操作 (在解决失败原因之后)。 此过程与没有 DR 组件的 RDQM HA 配置相同。 请参阅第 480 页的『失败的资源操作』以获取更多详细信息。

相关任务

第 514 页的『查看 DR/HA RDQM 和 HA 组状态』
您可以查看 DR/HA 复制数据队列管理器 (RDQM) 的 HA 状态和 DR 角色。

第 481 页的『查看 RDQM 和 HA 组状态』
您可以查看 HA 组和个别复制数据队列管理器 (RDQM) 的状态。

相关参考

Rdqmclean
rdqmstatus

Linux V 9.2.0 查看 *DR/HA RDQM* 和 *HA* 组状态

您可以查看 DR/HA 复制数据队列管理器 (RDQM) 的 HA 状态和 DR 角色。

关于此任务

您可以使用 **rdqmstatus** 命令来查看各个 RDQM 的状态，或者获取 HA 组已知的所有 RDQM 的状态概述。

V 9.2.3 节点的摘要状态还显示有关 RDQM 所依赖的 DRBD 内核模块的信息。 升级 RDQM 时，请务必确保为系统上运行的 RHEL 内核版本安装了正确版本的 DRBD 内核模块。 状态显示操作系统内核的版本，为其构建 DRBD 模块的内核版本， DRBD 版本和 DRBD 内核模块装入状态。

您必须是 mqm 和 haclient 组中的用户才能运行 **rdqmstatus** 命令。 您可以在任一 HA 组中的任何节点上运行该命令。

过程

- 要查看属于 HA 配置的节点和 RDQM 的摘要状态:

```
rdqmstatus
```

将显示运行该命令的节点的身份以及 HA 配置中 RDQM 的状态，以及它们的当前 DR 角色，例如:

```
Node:                                 main-alice
OS kernel version:                    3.10.0-1160.15.2
DRBD OS kernel version:               3.10.0-1160
DRBD version:                         9.1.1
DRBD kernel module status:            Loaded

Queue manager name:                   RDQM1
Queue manager status:                 Running elsewhere
```

```
HA current location:                    main-charlie
HA preferred location:                  main-charlie
HA blocked location:                    None

Queue manager name:                     RDQM9
Queue manager status:                   Running elsewhere
HA current location:                    main-bob
HA preferred location:                  main-bob
HA blocked location:                    None
DR role:                                Primary

Queue manager name:                     RDQM7
Queue manager status:                   Running
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
DR role:                                Primary
```

在此示例中， RDQM7 和 RDQM8 都是 DR/HA RDQM ， 而 RDQM1 是 HA RDQM ， 未配置为能够切换到灾难恢复站点。

V 9.2.3 DRBD 内核模块状态为以下值之一:

已加载
指示已装入 DRBD 模块。

已部分加载
装入 DRBD 模块时可能发生，但由于不匹配而无法正常工作。

未加载
未装入 DRBD 模块。 当尚未创建任何 RDQM 队列管理器时，可以在新安装的配置上显示此消息。

未安装
指示未安装 DRBD 模块。 或者 IBM MQ 无法确定 DRBD 模块的操作系统内核版本。

仍加载先前安装的版本
如果在现有 DRBD 模块正在运行 (即 RDQM 队列管理器正在运行) 时安装了新的 DRBD 模块，那么会出现此状态。 新安装的模块在状态中报告，但并非实际正在运行的模块。

- 要查看 HA 组中所有节点上特定队列管理器的状态，请输入以下命令:

```
rdqmstatus -m qmname
```

其中 *qmname* 是要查看其状态的 RDQM 的名称。 将显示当前节点上 RDQM 的状态，后跟当前节点透视图中其他两个节点的状态摘要。

- V 9.2.2
要查看 HA 组中所有节点上特定队列管理器的状态，包括任何失败资源操作的详细信息，请输入以下命令:

```
rdqmstatus -m qmname -a
```

其中 *qmname* 是要查看其状态的 RDQM 的名称。 将显示当前节点上 RDQM 的状态，后跟当前节点透视图中其他两个节点的状态摘要。 接下来是有关与 RDQM 关联的任何失败资源操作的详细信息。

- 下表汇总了 `rdqmstatus -m qmname` 命令可针对 RDQM 返回的有关当前节点的信息。

表 *36:* 当前节点状态

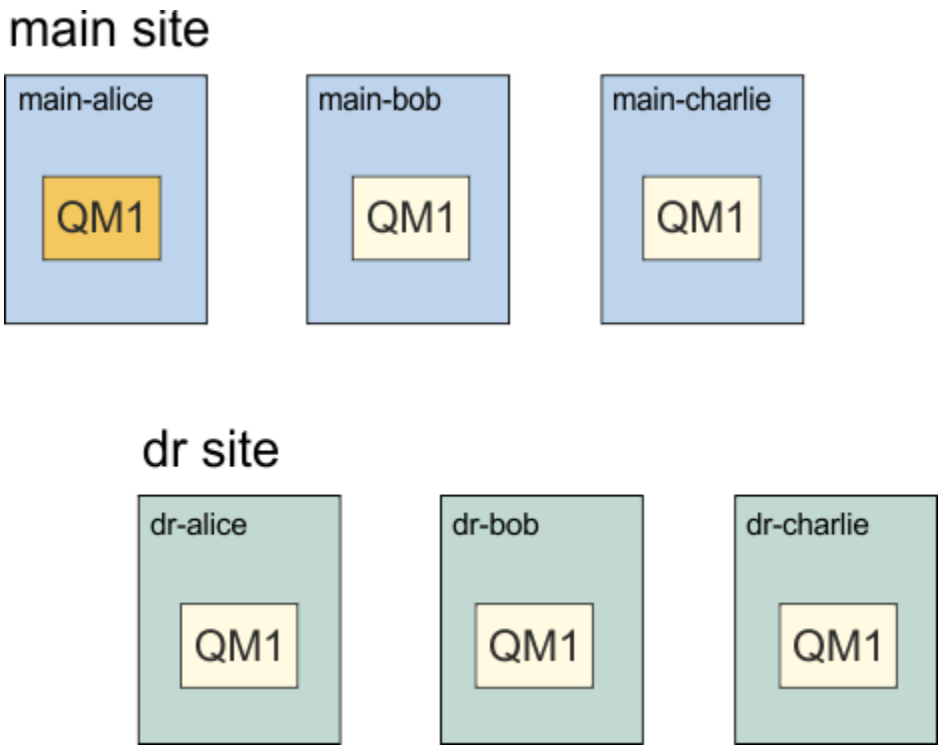
状态属性	**可能的值**	**显示时**
节点名	*NODENAME*	始终显示
队列管理器状态	队列管理器的状态 (对 **dspmq** 命令有效的状态之一)	始终显示
CPU	*n.nn*%	仅当 RDQM 在此节点上运行时才显示

表 *36:* 当前节点状态 *(*继续*)*		
状态属性	**可能的值**	**显示时**
内存	已使用 *nnn*MB	仅当 RDQM 在此节点上运行时才显示
队列管理器文件系统	已使用 *nnn*MB， *y.y*GB 已分配 [*z*%]	仅当 RDQM 在此节点上运行时才显示
HA 角色	主 辅助 未知	始终显示
HA 状态	所有节点处于备用状态 此节点处于备用状态 远程节点处于备用状态 混合	所有节点处于备用状态 当前节点处于备用状态 两个远程节点都处于备用状态 每个远程节点的不同状态
HA 控制	已启用 已禁用 未知	始终显示。 显示 RDQM 是否受 Pacemaker 控制
HA 首选位置	无 此节点 未知 *NODENAME*	始终显示
V 9.2.2 V 9.2.2 HA 阻止的位置	无-未阻止队列管理器在任何节点上运行 此节点-由于一个或多个失败的资源操作，阻止队列管理器在当前节点上运行 *nodename* -由于一个或多个失败的资源操作，阻止队列管理器在 *nodename* 上运行 *nodename1*， *nodename2* -由于一个或多个失败的资源操作，队列管理器被阻止在 *nodename1* 和 *nodename2* 上运行 所有节点-由于一个或多个失败的资源操作，阻止队列管理器在所有节点上运行	始终显示
HA 浮动 IP 接口	*interface_name*	始终显示
HA 浮动 IP 地址	*IPV4_address*	始终显示
DR 角色	主 辅助 辅助暂挂 未知	始终显示

表 *36:* 当前节点状态 *(*继续*)*		
状态属性	**可能的值**	**显示时**
DR 状态	正常 正在进行同步 已分区 远程系统不可用 不一致 正在还原到快照 未配置远程系统 协商失败	一切都好 正在进行同步。 用户已启动队列 每个节点上的管理器， DR 复制网络 unavailable. 与其他节点的连接 已丢失。 同步正在进行中，但已中断。 用户已选择还原为 在以下情况下生成的快照: 队列管理器输入了 状态不一致。 已配置主数据库 但二级却没有 主节点与辅助节点之间的初始协商失败。 这可能是由不兼容的复制类型导致的，或者如果使用较小的文件系统大小配置了辅助节点。
DR 状态 (在 HA 辅助节点上)	请参阅 *HA_Primary_Node*	在 HA 辅助节点上显示，因为 DR 状态仅在 HA 主节点上已知。
DR 端口	用于复制此队列管理器的数据的 TCP/IP 端口。	始终显示。
DR 本地 IP 地址	此队列管理器将用于 DR 复制的本地 IP 地址	始终显示。
DR 远程 IP 地址列表	此队列管理器将用于 DR 复制的远程 IP 地址。 三个 IP 地址的逗号分隔列表。	始终显示。
DR 当前远程 IP 地址	此队列管理器为 DR 复制而连接到的当前远程 IP。	对于具有活动 DR 连接的 HA 主项。
DR 当前远程 IP 地址 (在 HA 辅助节点上)	请参阅 *HA_Primary_Node*	在 HA 辅助节点上显示，因为 DR 连接仅在 HA 主节点上
DR 不同步数据	*x*KB	当远程节点不可用或不一致时显示。
DR 同步进度	*y*%	正在进行同步时显示。
DR 预计完成时间	*yyyy-MM-dd HH:mm:ss*	正在进行同步时显示。
快照还原进度	*y*%	DR 状态为 "正在恢复到快照" 时显示
V 9.2.2 V 9.2.2 DR 上次同步时间	*yyyy-MM-dd HH:mm:ss*	在 DR 数据不同步 (在初始同步后) 时显示。 提供上次同步数据的时间和日期。

示例

这些示例说明了在以下 DR/HA 配置的各个节点上运行的命令 **rdqmstatus -m qm1**：



作为 DR 主节点和 HA 主节点的正常状态示例:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           192.168.2.1

Node:                                   main-bob
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

作为 DR 主节点和 HA 辅助节点的正常状态示例:

```
Node:                                   main-bob
Queue manager status:                   Running elsewhere
HA role:                                Secondary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    main-alice
HA preferred location:                  main-alice
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              See main-alice
DR port:                                3000
DR local IP address:                    192.168.1.2
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           See main-alice
```

```
Node:                                   main-alice
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

作为 DR 辅助和 HA 主节点的节点上的正常状态示例:

```
Node:                                   dr-alice
Queue manager status:                   Ended immediately
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Secondary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.2.1
DR remote IP address list:              192.168.1.1,192.168.1.2,192.168.1.3
DR current remote IP address:           192.168.1.1

Node:                                   dr-bob
HA status:                              Normal

Node:                                   dr-charlie
HA status:                              Normal
```

作为 DR 辅助和 HA 辅助的节点上的正常状态示例:

```
Node:                                   dr-bob
Queue manager status:                   Ended immediately
HA role:                                Secondary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    dr-alice
HA preferred location:                  dr-alice
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Secondary
DR status:                              See dr-alice
DR port:                                3000
DR local IP address:                    192.168.2.2
DR remote IP address list:              192.168.1.1,192.168.1.2,192.168.1.3
DR current remote IP address:           See dr-alice

Node:                                   dr-alice
HA status:                              Normal

Node:                                   dr-charlie
HA status:                              Normal
```

在作为 DR 主节点和 HA 主节点的节点上正在进行 DR 同步的示例:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           192.168.2.1
```

```
DR synchronization progress:            11.0%
DR estimated time to completion:        2018-09-06 14:55:05

Node:                                   main-bob
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

在作为 DR 主节点和 HA 主节点的节点上分区的 DR 示例:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Partitioned
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           192.168.2.1
DR out of sync data:                    372KB

Node:                                   main-bob
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

V 9.2.2 作为 DR 主节点和 HA 主节点的节点上的 DR 不同步示例:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Remote unavailable
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           Unknown
DR out of sync data:                    372KB
DR last in sync:                        2020-02-02 20:22:02

Node:                                   main-bob
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

V 9.2.2 作为 DR 主节点和 HA 主节点的 HA 不同步示例:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
```

```
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           192.168.2.1

Node:                                   main-bob
HA status:                              Inconsistent
HA out of sync data:                    15932KB
HA last in sync:                        2020-02-02 20:22:02

Node:                                   main-charlie
HA status:                              Normal
```

V 9.2.3 显示操作系统内核版本 (RHEL 7.9) 与 DRBD 内核模块 (针对 RHEL 7.8) 不匹配的摘要状态示例。 即使状态报告已装入 DRBD 内核模块并且队列管理器正在运行，您也应该在此情况下使用针对正在运行的操作系统内核的版本来更新 DRBD 内核模块。

```
Node:                                   main-alice
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-1127
DRBD version:                           9.1.1
DRBD kernel module status:              Loaded

Queue manager name:                     QM1
Queue manager status:                   Running
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
DR role:                                Primary
```

V 9.2.3 显示操作系统内核版本 (RHEL 7.9) 与 DRBD 内核模块 (针对 RHEL 7.6) 不匹配的摘要状态示例。 在此示例中，版本不匹配更严重，并且 DRBD 内核模块无法成功装入。 QM1 是一个 HA/DR 队列管理器，移动到另一个节点，其 HA 状态未知，其 DR 状态未知。 要解决此故障，必须使用正在运行的操作系统内核的版本目标来更新 DRBD 内核模块。

```
Node:                                   main-alice
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-957
DRBD version:                           9.1.2+ptf.3
DRBD kernel module status:              Partially loaded

Queue manager name:                     QM1
Queue manager status:                   Running elsewhere
HA status:                              Unknown
HA current location:                    main-bob
HA preferred location:                  This node
HA blocked location:                    None
DR role:                                Primary
DR status:                              Unknown
```

相关参考

Linux rdqmstatus

Linux 在 *DR/HA* 环境中运行

在 DR/HA 环境中运行时，对于高可用性和灾难恢复有不同的注意事项。

如果运行 DR/HA RDQM 的节点发生故障，那么 RDQM 会自动故障转移到该 HA 组中的另一个节点。 如果整个站点发生故障，那么必须在恢复站点的 HA 组中的首选节点上手动启动 RDQM。 此处的注意事项与普通 DR RDQM 的注意事项相同，请参阅 第 503 页的『在灾难恢复环境中运行』 以获取更多信息。

如果其中一个节点完全发生故障，并且需要更换，请参阅 第 504 页的『在灾难恢复配置中替换发生故障的节点』 和 第 487 页的『更换高可用性配置中发生故障的节点』 以获取指导信息。

Linux V 9.2.0 在 *DR/HA* 配置中更换发生故障的节点

如果任一 HA 组中的某个节点发生故障，那么可以将其替换。

关于此任务

该过程根据您要更换的节点是 DR 配置中的主节点还是辅助节点而有所不同。 在任一情况下，新节点都必须具有与您要替换的节点相同的配置，即，它必须具有相同的主机名，相同的 IP 地址等。

您还可能会迁到在主站点或恢复站点完全失去 HA 组并且必须替换整个 HA 组的情况。

过程

- 对于作为 DR 配置中的主节点的替换节点，请在新节点上完成以下步骤:
  a) 创建与其他节点上的文件匹配的 `rdqm.ini` 文件，然后运行 `rdqmadm -c` 命令 (请参阅 第 468 页的『定义 Pacemaker 集群 (HA 组)』)。
  b) 运行 `crtmqm -sxs -rr p` *qmanager* 命令以重新创建每个 DR/HA RDQM (请参阅 第 509 页的『创建 DR/HA RDQM』)。
- 对于作为 DR 配置中辅助节点的更换节点，请在新节点上完成以下步骤:
  a) 创建与其他节点上的文件匹配的 `rdqm.ini` 文件，然后运行 `rdqmadm -c` 命令 (请参阅 第 468 页的『定义 Pacemaker 集群 (HA 组)』)。
  b) 运行 `crtmqm -sx -rr s` *qmanager* 命令以重新创建每个 DR/HA RDQM (请参阅 第 509 页的『创建 DR/HA RDQM』)。
- 要替换整个 HA 组，请完成以下步骤:
  a) 如果丢失 DR 主站点 (即主站点) 上的整个 HA 组，那么必须执行以下步骤以执行到 DR 辅助站点的受管故障转移，从而继续运行 DR/HA RDQM (请参阅 第 503 页的『在灾难恢复环境中运行』)。 (如果在恢复站点上丢失了整个 HA 组，那么 DR/HA RDQM 将继续在主站点上运行，而无需您的干预。)
  b) 在三个替换节点上重新创建 HA 组，如 第 508 页的『为 DR/HA RDQM 配置 HA 组』中所述。
  c) 在新的 HA 组上重新创建 DR/HA RDQM ，如 第 509 页的『创建 DR/HA RDQM』中所述。
  d) 如果需要，请执行从恢复站点到主站点的受管故障转移。

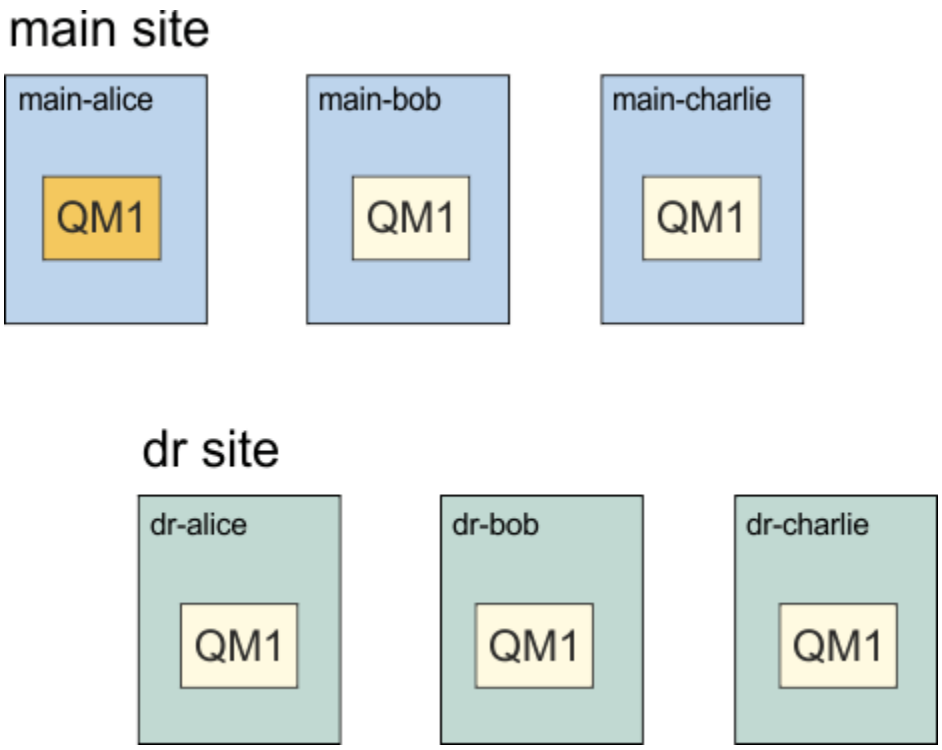
Linux V 9.2.0 *DR/HA RDQM* 工作示例

此示例显示如何创建和删除 DR/HA RDQM。

创建 DR/HA RDQM

示例配置有两个站点，分别名为 "main" 和 "dr"。 每个站点都有三个节点，分别名为 "alice" ， "bob" 和 "charlie"。 节点的全名由其站点名称和名称组成，因此为 "main-alice" ， "dr-alice" 等。

以下步骤创建在 main-alice 上运行的名为 QM1 的 DR/HA RDQM。 主 alice 节点是 HA 和 DR 主节点。



如果在 rdqm.ini 文件中指定了本地和远程 DR IP 地址，那么不需要在命令行上指定任何 IP 地址，并且可以通过在 main-alice 上运行以下命令来创建名为 QM1 的 DR/HA RDQM:

```
crtmqm -sx -rr p -rn DR1 -rp 7001 QM1
```

如果在 rdqm.ini 文件中指定了本地 DR IP 地址，那么可以在命令行上指定远程 DR IP 地址:

```
crtmqm -sx -rr p -ri 192.168.2.1,192.168.2.2,192.168.2.3 -rp 7001 QM1
```

如果在 rdqm.ini 文件中未指定 DR IP 地址，那么可以在命令行上同时指定远程和本地 DR IP 地址:

```
crtmqm -sx -rr p -rl 192.168.1.1,192.168.1.2,192.168.1.3 -ri
192.168.2.1,192.168.2.2,192.168.2.3 -rp 7001 QM1
```

以下示例中显示了响应创建 QM1 的输出:

```
Creating replicated data queue manager configuration.
Secondary queue manager created on 'main-bob'.
Secondary queue manager created on 'main-charlie'.
IBM MQ queue manager created.
Directory '/var/mqm/vols/qm1/qmgr/qm1' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QM1'.
Default objects statistics : 83 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
Enabling replicated data queue manager.
Replicated data queue manager enabled.
Issue the following command on the remote HA group to create the DR/HA secondary queue manager:
crtmqm -sx -rr s -rl 192.168.2.1,192.168.2.2,192.168.2.3 -ri
192.168.1.1,192.168.1.2,192.168.1.3 -rp 7001 -fs 3072M QM1
```

从消息复制命令以在 dr-alice 上创建 QM1 的 DR 辅助实例:

```
crtmqm -sx -rr s -rl 192.168.2.1,192.168.2.2,192.168.2.3 -ri
192.168.1.1,192.168.1.2,192.168.1.3 -rp 7001 -fs 3072M QM1
```

在 dr-alice 上输出以下消息:

```
Creating replicated data queue manager configuration.
Secondary queue manager created on 'dr-bob'.
```

```
Secondary queue manager created on 'dr-charlie'.
IBM MQ secondary queue manager created.
Enabling replicated data queue manager.
```

测试 DR 辅助服务器

要测试 QM1 的灾难恢复功能，请在 main-alice 上运行以下命令以使 QM1 成为 DR 辅助实例:

```
rdqmdr -m QM1 -s
Queue manager 'QM1' has been made the DR secondary on this node.
```

在 dr-alice 上运行以下命令，以使 QM1 成为该节点上的 DR 主实例:

```
rdqmdr -m QM1 -p
Queue manager 'QM1' has been made the DR primary on this node.
```

删除 DR/HA RDQM

要删除名为 QM1 的 DR/HA RDQM ，请首先在 main-alice 上结束队列管理器:

```
endmqm -w QM1
Replicated data queue manager disabled.
Waiting for queue manager 'QM1' to end.
IBM MQ queue manager 'QM1' ended.
```

然后，在 main-alice 上运行以下命令以删除 QM1:

```
dltmqm QM1
Removing replicated data queue manager configuration.
Secondary queue manager deleted on 'main-bob'.
Secondary queue manager deleted on 'main-charlie'.
IBM MQ queue manager 'QM1' deleted.
```

最后，必须删除 dr-alice 上的 QM1 :

```
dltmqm QM1
Removing replicated data queue manager configuration.
Secondary queue manager deleted on 'dr-bob'.
Secondary queue manager deleted on 'dr-charlie'.
IBM MQ queue manager 'QM1' deleted.
```

相关概念

第 521 页的『在 DR/HA 环境中运行』
在 DR/HA 环境中运行时，对于高可用性和灾难恢复有不同的注意事项。

相关任务

第 509 页的『创建 DR/HA RDQM』
您可以使用 **crtmqm** 命令在 DR/HA 配置中创建复制的数据队列管理器 (RDQM)。

第 512 页的『删除 DR/HA RDQM』
使用 **dltmqm** 命令可删除 DR/HA 复制的数据队列管理器 (RDQM)。

CP4I 本机 HA

本机 HA 是在 IBM MQ 的容器部署上可用的高可用性解决方案。

本机 HA 配置由三个节点 (例如，可以是三个 Kubernetes pod) 组成，每个节点都具有队列管理器的实例。一个实例是活动队列管理器，处理消息并写入其日志。 每当写入日志时，活动队列管理器都会将数据发送到其他两个实例 (称为 "副本")。 每个副本写入自己的日志，确认数据，然后从复制的日志更新自己的队列数据。 如果运行活动队列管理器的节点发生故障，那么队列管理器的其中一个副本实例将接管活动角色，并具有要使用的当前数据。

有关详细概述，请参阅本文档的 "容器" 部分中的 本机 HA 。

下图显示了在三个容器中部署了队列管理器的三个实例的典型部署。

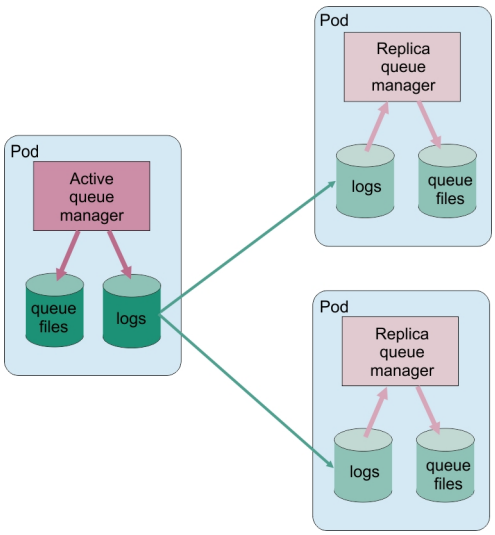


图 *86:* 本机 *HA* 配置示例

CP4I 创建本机 *HA* 解决方案

创建本机 HA 解决方案的建议方法是使用 IBM MQ Operator。 或者，您可以创建自己的容器并手动配置本机 HA。

注: 此信息仅适用于容器环境。

要使用 IBM MQ Operator 创建本机 HA 解决方案，请参阅 本机 HA 以获取概述和 示例: 配置本机 HA 队列管理器 以获取详细指示信息。

要创建自己的容器并手动配置本机 HA，请参阅 创建自己的容器时创建本机 HA 组。

CP4I 结束本机 *HA* 队列管理器

对于容器中的 IBM MQ，可以使用 **endmqm** 命令来结束属于本机 HA 组的活动队列管理器或副本队列管理器。

关于此任务

注: 此信息仅适用于容器环境。

停止属于本机 HA 组的队列管理器的过程取决于它是活动实例还是副本实例。 当您结束任一类型的实例时，将进行检查以确保结束实例不会破坏本机 HA 组的定额。 如果定额将被破坏，那么 **endmqm** 命令将失败。

当您发出 **endmqm** 命令时，将警告组中的其他实例发生此情况，以便它们不会在连接中断时报告错误。

如果活动实例由于结束或断开过多副本实例的连接而失去定额，那么活动实例将在完全结束之前等待可配置的时间量。 这将允许一段时间来正常地关闭处理，而不是仅使应用程序的连接中断。 此超时值可以由 `qm.ini` 文件的 `NativeHALocalInstance` 节中的 `QuorumConnectivityTimeout` 属性指定。 缺省值为 0 秒。

过程

- 要结束队列管理器的活动实例，请在运行该活动实例的节点上发出以下命令:

  ```
  endmqm -s QMgrName
  ```

  - 指定 `-r` 选项以帮助客户机应用程序重新连接到另一个实例。
  - 如果此实例不是本机 HA 组中的活动实例，那么该命令将失败。
  - 如果结束此活动实例将导致组定额失败，那么该命令将失败。 (如果其他实例在您运行此命令的同时结束或变为不可用，那么定额检查可能不会检测到此情况，本机 HA 组将结束，并且仅当有足够的实例可用时才能重新启动。)

当活动队列管理器结束时，其中一个副本实例将接管活动角色。 不能指定接管哪个副本，这由组内的协商确定，并取决于哪个副本具有最新的事务日志。

- 要结束队列管理器的副本实例，请发出以下命令:

```
endmqm -x QMgrName
```

  - 如果此实例是活动实例，那么该命令将失败。
  - 如果结束此副本实例将导致组定额失败，那么该命令将失败。 (如果其他实例在您运行此命令的同时结束或变为不可用，那么定额检查可能不会检测到此情况，本机 HA 组将结束，并且仅当有足够的实例可用时才能重新启动。)

  注: 您还可以在本机 HA 实例上使用带有 **endmqm** 命令的 -c ， -i ， -p 或 -w 开关，而不考虑它们的角色。 队列管理器实例结束，忽略对组定额的影响。 但是，仍与该组中的其他实例共享信息。 可以将这些开关与 -s 一起用于活动实例。 不能将这些交换机与 -x 交换机一起用于副本实例。

相关参考

endmqm (结束队列管理器)

日志记录: 确保消息未丢失

IBM MQ 在恢复日志中记录对队列管理器控制的持久数据的所有重要更改。

这包括创建和删除对象，持久消息更新，事务状态，对对象属性的更改以及通道活动。 该日志包含通过以下方法恢复对消息队列的所有更新所需的信息:

- 保留队列管理器更改的记录
- 保留队列更新记录以供重新启动进程使用
- 使您能够在硬件或软件发生故障后复原数据

但是， IBM MQ 还依赖于托管其文件 (包括日志文件) 的磁盘系统。 如果磁盘系统本身不可靠，那么包括日志信息在内的信息仍可能丢失。

警告: 不能将恢复日志移动到其他操作系统。

日志的外观

日志由主文件和辅助文件以及控制文件组成。 您可以定义日志文件的数量和大小以及它们在文件系统中的存储位置。

IBM MQ 日志由两个组件组成:

1. 一个或多个日志数据文件。
2. 日志控制文件

日志数据的文件也称为日志扩展数据块。

有许多日志扩展数据块包含要记录的数据。 您可以定义数字和大小 (如 第 79 页的『mqs.ini 文件的 LogDefaults 节』中所述)，或者采用系统缺省值 3 个主扩展数据块和 2 个辅助扩展数据块。

三个主扩展数据块和两个辅助扩展数据块中的每个扩展数据块缺省为 16 MB。

创建队列管理器时，预先分配的日志扩展数据块数是已分配的 主 日志扩展数据块数。 如果未指定数字，那么将使用缺省值。

IBM MQ 使用两种类型的日志记录:

- 循环
- 线性

  用于线性日志记录的日志扩展数据块数可能非常大，具体取决于介质图像记录的频率。

请参阅 第 527 页的『日志记录类型』以获取更多信息。

ALW 在 IBM MQ for AIX or Linux 系统中，如果尚未更改日志路径，那么将在以下目录下创建日志扩展数据块:

```
/var/mqm/log/QMgrName
```

Windows 在 IBM MQ for Windows 中，如果未更改日志路径，那么将在以下目录下创建日志扩展数据块:

```
CMQ\log\QMgrName
```

IBM MQ 从这些主日志扩展数据块开始，但如果主日志空间不足，那么它将分配 辅助 日志扩展数据块。 它会动态执行此操作，并在日志空间需求减少时将其除去。 缺省情况下，最多可以分配两个辅助日志扩展数据块。 您可以更改此缺省分配，如 第 71 页的『在 Multiplatforms 版上的 .ini 文件中更改 IBM MQ 配置信息』中所述。

日志扩展数据块以字母 S 或字母 R 作为前缀。 活动扩展数据块，不活动扩展数据块和多余扩展数据块以 S 作为前缀，而复用扩展数据块以 R 作为前缀。

备份或复原队列管理器时，请备份和复原所有活动，不活动和多余的扩展数据块以及日志控制文件。

注: 您不需要备份和复原复用扩展数据块。

日志控制文件

日志控制文件包含描述日志扩展数据块状态所需的信息，例如它们的大小和位置以及下一个可用扩展数据块的名称。

要点: 日志控制文件仅供内部队列管理器使用。

队列管理器将与恢复日志状态关联的控制数据保留在日志控制文件中，并且您不得修改日志控制文件的内容。

日志控制文件位于日志路径中，称为 `amqhlctl.lfh`。 备份或复原队列管理器时，请确保已备份和复原日志控制文件以及日志扩展数据块。

日志记录类型

在 IBM MQ 中，有两种维护队列管理器活动记录的方法: 循环日志记录和线性日志记录。 第三种类型的日志记录 (已复制) 仅由本机 HA 配置使用。

循环日志记录

如果您只想重新启动恢复，请使用循环日志记录，并使用日志来回滚系统停止时正在进行的事务。

循环日志记录将所有重新启动数据保存在日志文件环中。 日志记录将填充环中的第一个文件，接着移至下一个文件，依此类推，直至所有文件已满。 然后，它返回至环中的第一个文件，并再次开始。 只要该产品在使用中，那么此操作将会继续进行，并且它具有永远不会用完日志文件的优点。

IBM MQ 保留重新启动队列管理器所需的日志条目，而不会丢失数据，直到不再需要这些日志条目来确保队列管理器数据恢复为止。 第 529 页的『使用检查点来确保完全恢复』中描述了释放日志文件以供复用的机制。

线性日志记录 (linear logging)

如果希望重新启动恢复和介质恢复 (通过重放日志内容来重新创建丢失或损坏的数据)，请使用线性日志记录。 线性日志记录将日志数据保留在连续的日志文件序列中。

(可选) 日志文件可以是:

- 已复用，但仅当重新启动恢复或介质恢复不再需要它们时才会复用。
- 手动归档以进行长期存储和分析。

介质映像的频率确定何时可以复用线性日志文件，并且是必须为线性日志文件提供多少磁盘空间的主要因素。

您可以将队列管理器配置为根据时间或日志使用情况自动获取定期介质映像，也可以手动调度介质映像。

管理员决定要实施的策略以及对磁盘空间使用情况的影响。 重新启动恢复所需的日志文件必须始终可用，而仅用于介质恢复的日志文件可以归档到更长期的存储器，例如磁带。

如果管理员启用自动日志管理和自动介质映像，那么线性日志记录的行为方式与非常大的循环日志类似，但通过介质恢复来提高对介质故障的冗余性。

从 IBM MQ 9.1.0 开始，可以使用 migmqlog 命令将队列管理器的现有日志类型从线性更改为循环，或者从循环更改为线性。

复制的日志记录

CP4I

使用复制的日志记录来配置本机 HA 配置。 创建本机 HA 组时，将在不同节点上创建三个队列管理器。 指定已复制的日志记录类型以及每个队列管理器的唯一实例名称。 本机 HA 配置通过使活动实例将日志数据复制到两个副本实例来提供高可用性解决方案。 如果活动实例失败，那么其中一个副本实例将接管活动角色。 日志复制可确保很少 (如果有) 数据丢失。 有关更多详细信息，请参阅 第 524 页的『本机 HA』。 复制的日志相当于启用了自动日志管理和自动介质映像的线性日志。

未处于活动状态的线性日志扩展数据块

Multi

从 IBM MQ 9.1.0 开始，如果您正在使用自动日志管理 (包括归档)，那么记录器会跟踪未处于活动状态的线性日志扩展数据块。

注意: 如果您正在使用自动日志管理，而不进行归档，那么此过程不支持使用备份队列管理器。

ALW 如果不再需要日志扩展数据块进行恢复，并且在必要时归档了日志扩展数据块，那么记录器将在方便的时候删除日志扩展数据块或复用日志扩展数据块。

复用的日志扩展数据块将重命名为日志序列中的下一个。 将定期写入消息 AMQ7490，指示已创建，删除或复用的扩展数据块数。

记录器选择要保持可供复用的扩展数据块数以及何时删除这些扩展数据块。

活动日志

在线性日志记录和循环日志记录中都有许多文件被认为是 活动的 文件。 活动日志是重新启动恢复可能引用的最大日志空间量 (无论您是使用循环日志记录还是线性日志记录)。

活动日志文件数通常小于配置文件中定义的主日志文件数。 (请参阅 第 532 页的『计算日志大小』 以获取有关定义数字的信息。)

请注意，活动日志空间不包括介质恢复所需的空间，并且用于线性日志记录的日志文件数可能非常大，这取决于您的消息流和介质映像的频率。

不活动日志

当不再需要日志文件进行重新启动恢复时，它将变为 不活动。 可以将重新启动恢复或介质恢复不需要的日志文件视为多余的日志文件。

使用自动日志管理时，队列管理器会控制这些多余日志文件的处理。 如果您已选择手动日志管理，那么管理员将负责管理 (例如，删除和归档) 多余的日志文件 (如果它们不再对您的操作感兴趣)。

请参阅 第 537 页的『管理日志』，以获取有关日志文件处置的更多信息。

辅助日志文件

虽然为线性日志记录定义了辅助日志文件，但它们不会用于正常操作。 如果可能由于长时间存在的事务而导致无法从活动池中释放文件，因为重新启动可能仍需要该文件，那么会对辅助文件进行格式化并将其添加到活动日志文件池中。

如果可用的辅助文件数已用完，那么将拒绝大多数需要日志活动的进一步操作的请求，并将 MQRC_RESOURCE_PROBLEM 返回码返回到应用程序，并且将考虑任何长时间运行的事务进行异步回滚。

注意: 所有类型的测井都可以应对意外的电力损失，假设没有硬件故障。

使用检查点来确保完全恢复

循环日志记录和线性日志记录队列管理器都支持重新启动恢复。 无论队列管理器的先前实例在重新启动时如何突然终止 (例如，断电) ，队列管理器都会在终止时将其持久状态恢复到正确的事务状态。

重新启动恢复取决于要维护的磁盘完整性。 同样，操作系统应确保磁盘完整性，而无论操作系统终止可能发生的情况如何突然。

在未维护磁盘完整性的极不寻常事件中，线性日志记录 (和介质恢复) 提供了一些进一步的冗余和可恢复性选项。 随着 RAID 等越来越常见的技术，越来越少出现磁盘完整性问题，很多企业配置循环日志记录，只使用重启恢复。

IBM MQ 设计为经典的 "未写日志记录" 资源管理器。 对消息队列的持久更新分两个阶段进行:

1. 表示更新的日志记录将可靠地写入恢复日志
2. 队列文件或缓冲区以对系统最有效但不一定一致的方式进行更新。

因此，日志文件可能比底层队列缓冲区和文件状态更新更新。

如果允许这种情况有增无减，那么将需要大量日志重放，以便在崩溃恢复之后使队列状态保持一致。

IBM MQ 使用 `checkpoints` 来限制崩溃恢复后所需的日志重放量。 控制日志文件是否被称为 "活动" 的关键事件是 `checkpoint`。

IBM MQ 检查点是一个点:

- 恢复日志与对象文件之间的一致性。
- 这标识了日志中的一个位置，从该位置可以保证后续日志记录的正向重放在队列管理器可能已结束时将队列恢复到正确的逻辑状态。

在检查点期间， IBM MQ 会根据需要将较旧的更新清空到队列文件，以限制需要重放的日志记录量，从而使队列在崩溃恢复后恢复到一致状态。

最近的完整检查点标记日志中必须在崩溃恢复期间执行重放的点。 因此，检查点的频率是记录检查点的开销与这些检查点所隐含的潜在恢复时间的改善之间的一种权衡。

从 IBM MQ 9.1.0 开始，记录器更频繁地调度检查点 (因此下一个检查点是在上一个检查点完成之前调度的) ，因为记录器正在尝试将活动日志保留在主日志扩展数据块中。 如果无法执行此操作，那么将记录 AMQ7466 错误。

日志中最近完成的检查点的起始位置是确定日志文件是活动还是不活动的关键因素之一。 另一个关键因素是与当前活动事务进行的第一个持久更新相关的第一个日志记录在日志中的位置。

如果在第二个或更高版本的日志文件中记录了新检查点，并且没有当前事务引用第一个日志文件中的日志记录，那么第一个日志文件将变为不活动状态。 对于循环日志记录，第一个日志文件现在可供复用。 对于线性日志记录，通常仍需要第一个日志文件进行介质恢复。

如果配置循环日志记录或自动日志管理，那么队列管理器将管理不活动的日志文件。 如果使用手动日志管理来配置线性日志记录，那么它将成为根据操作需求来管理不活动文件的管理任务。

IBM MQ 会自动生成检查点。 它们是在以下时间拍摄的:

- 队列管理器何时启动
- 关闭时

- 日志记录空间正在低度运行时
- Multi 自上一个检查点执行以来记录了 50,000 次操作
- z/OS 在记录自上一个检查点以来的 *number_of_operations* 之后，其中 *number_of_operations* 是 **LOGLOAD** 属性中设置的操作数。

当 IBM MQ 重新启动时，它会在日志中找到最新的检查点记录。 此信息保存在每个检查点结束时更新的检查点文件中。 将转发自检查点以来执行的所有操作。 这称为重放阶段。

重放阶段会将队列恢复到它们在系统故障或关闭之前所处的逻辑状态。 在重放阶段，将创建当发生系统故障或关闭时正在进行的事务的列表。

Multi 将发出消息 AMQ7229 和 AMQ7230 以指示重放阶段的进度。

为了了解要回退或落实的操作， IBM MQ 会访问与正在进行的事务关联的每个活动日志记录。 这称为恢复阶段。

Multi 将发出消息 AMQ7231， AMQ7232 和 AMQ7234 以指示恢复阶段的进度。

在恢复阶段访问所有必需的日志记录后，将依次解析每个活动事务，并且将回退或落实与该事务相关联的每个操作。 这称为解决阶段。

Multi 将发出消息 AMQ7233 以指示解决阶段的进度。

z/OS 在 z/OS 上，重新启动处理由各种阶段组成。

1. 根据页集所需的介质恢复以及回退工作单元和获取不确定工作单元的锁定所需的最早日志记录，建立恢复日志范围。
2. 一旦确定了日志范围，就会执行正向日志读取，使页面设置达到最新状态，还会锁定与不确定或正在运行的工作单元相关的任何消息。
3. 当已完成正向日志读取时，将向后读取日志以回退发生故障时正在进行或正在回退的任何工作单元。

z/OS 您可能会看到的消息示例:

```
CSQR001I +MQOX RESTART INITIATED
CSQR003I +MQOX RESTART - PRIOR CHECKPOINT RBA=00000001E48C0A5E
CSQR004I +MQOX RESTART - UR COUNTS -  806
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN BACKOUT=0
CSQR030I +MQOX Forward recovery log range  815
from RBA=00000001E45FF7AD to RBA=00000001E48C1882
CSQR005I +MQOX RESTART - FORWARD RECOVERY COMPLETE -  816
IN COMMIT=0, INDOUBT=0
CSQR032I +MQOX Backward recovery log range  817
from RBA=00000001E48C1882 to RBA=00000001E48C1882
CSQR006I +MQOX RESTART - BACKWARD RECOVERY COMPLETE -  818
INFLIGHT=0, IN BACKOUT=0
CSQR002I +MQOX RESTART COMPLETED
```

注: 如果要读取大量日志，那么会定期发出消息 CSQR031I (正向恢复) 和 CSQR033I (逆向恢复) 以显示进度。

在 第 531 页的图 87 中， IBM MQ 不再需要最新检查点 Checkpoint 2 之前的所有记录。 可以从检查点信息和任何后续日志条目中恢复队列。 对于循环日志记录，可以复用检查点之前释放的任何文件。 对于线性日志，不再需要访问已释放的日志文件以进行正常操作并变为不活动状态。 在此示例中，将队列头指针移动到最新检查点 Checkpoint 2 处，然后该检查点将变为新的队列头 Head 2。 现在可以复用日志文件 1。

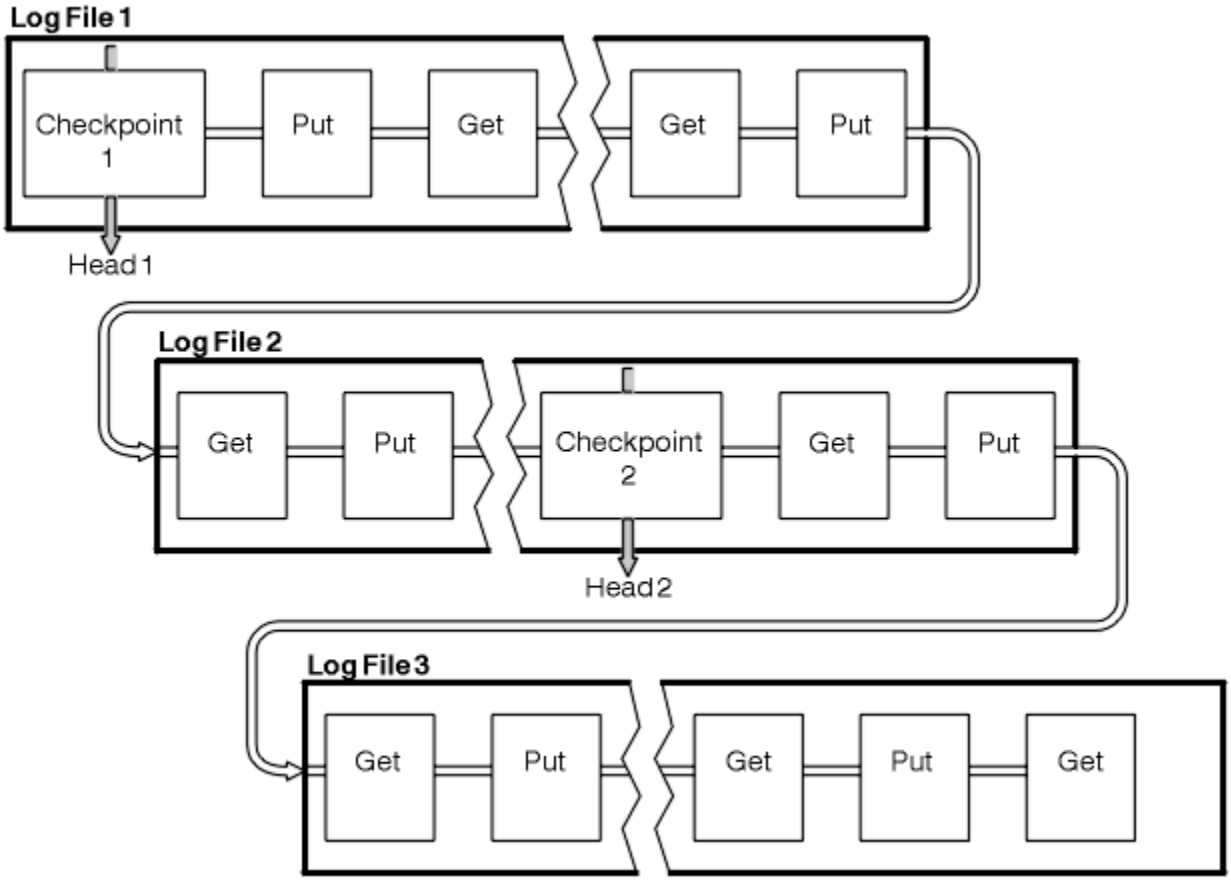


图 *87:* 检查点

使用长时间运行的事务的检查点

长时间运行的事务如何影响日志文件的复用。

第 532 页的图 88 显示了长时间运行的事务如何影响日志文件的复用。 在此示例中，长时间运行的事务已在显示的第一个检查点之后输入日志，显示为 LR 1。 直到第三个检查点之后，事务才会完成 (在 LR 2 点)。 将保留从 LR 1 开始的所有日志信息，以允许在该事务完成之前 (如果需要) 恢复该事务。

在长时间运行的事务完成后，在 LR 2 上，日志头将逻辑移至 Checkpoint 3 (最新记录的检查点)。 不再需要包含 Checkpoint 3 头 2 之前的日志记录的文件。 如果您正在使用循环日志记录，那么可以复用该空间。

如果在长时间运行的事务完成之前主日志文件完全已满，那么可以使用辅助日志文件来避免日志已满。

将完全由队列管理器控制的活动 (例如，检查点) 调度为尝试将活动保留在主日志中。

但是，当需要辅助日志空间来支持队列管理器控制之外的行为 (例如，某个事务的持续时间) 时，队列管理器会尝试使用任何已定义的辅助日志空间，以允许该活动完成。

如果在使用总日志空间的 80% 时该活动未完成，那么队列管理器将启动回收日志空间的操作，而不管这对应用程序有影响。

当移动日志头并且您正在使用循环日志记录时，主日志文件可能符合复用条件，并且记录器在填充当前文件后，将复用可供其使用的第一个主文件。 如果您正在使用线性日志记录，那么日志头仍将移至活动池下，并且第一个文件将变为不活动状态。 将格式化新的主文件并将其添加到池底部，以便为将来的日志记录活动做好准备。

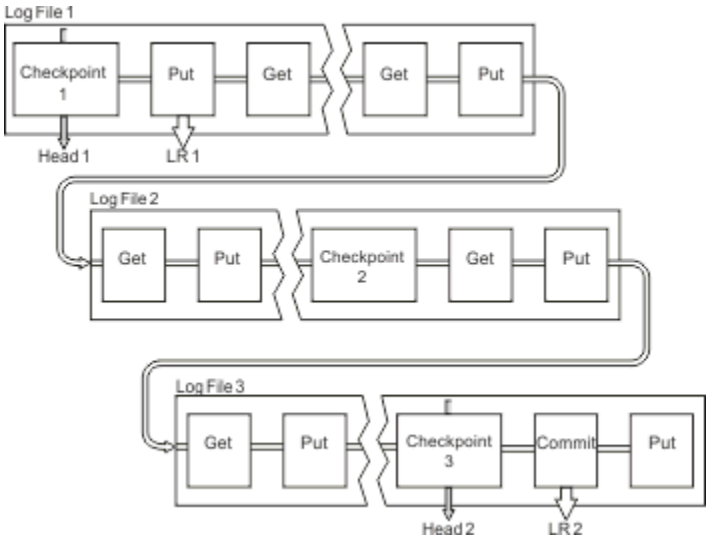


图 *88:* 使用长时间运行的事务进行检查点

计算日志大小

估算队列管理器需要的日志大小。

在确定队列管理器是使用循环日志记录还是线性日志记录之后，您需要估算队列管理器所需的 活动日志 的大小。 活动日志的大小由以下日志配置参数确定:

LogFilePages
每个主日志文件和辅助日志文件的大小 (以 4K 页为单位)

LogPrimaryFiles
预分配的主日志文件数

LogSecondaryFiles
在主日志文件变满时可以创建以使用的辅助日志文件数

注意:

1. 您可以在队列管理器每次启动时更改主日志文件和辅助日志文件的数目，尽管您可能不会立即注意到对辅助日志进行的更改的影响。
2. 您无法更改日志文件大小; 必须在创建队列管理器 **之前** 确定日志文件大小。
3. 主日志文件数和日志文件大小决定了创建队列管理器时预分配的日志空间量。
4. 主日志文件和辅助日志文件的总数在 AIX and Linux 系统上不能超过 511 ，在 Windows 上不能超过 255 ，如果存在长时间运行的事务，那么将限制队列管理器可用于重新启动恢复的最大日志空间量。 队列管理器可能需要用于介质恢复的日志空间量不共享此限制。
5. 使用 循环 日志记录时，队列管理器将复用主日志空间和辅助日志空间。 队列管理器将在日志文件变满时分配辅助日志文件，并且序列中的下一个主日志文件不可用。

   请参阅 第 532 页的『我的活动日志应该有多大?』 ，以获取有关需要分配的日志数的信息。 主日志扩展数据块按顺序使用，并且该顺序不会更改。

   例如，如果您有三个主日志 `0`， `1` 和 `2`，那么使用顺序为 `0`，`1`，`2` 后跟 `1`，`2`，`0`， `2`，`0`，`1`，返回到 `0`，`1`，`2` 等。 您已分配的任何辅助日志都将根据需要进行穿插。
6. 主日志文件可供在检查点期间复用。 在采用检查点之前，队列管理器会同时考虑主日志空间和辅助日志空间，因为日志空间量正在低度运行。

   队列管理器尝试以在主扩展数据块中保留日志使用情况的方式调度检查点。

请参阅 第 79 页的『mqs.ini 文件的 LogDefaults 节』 以获取更多信息。

我的活动日志应该有多大?

估算队列管理器需要的活动日志大小。

活动日志的大小受以下限制:

```
logsize = (primaryfiles + secondaryfiles) * logfilepages * 4096
```

当队列管理器将每秒的最大数据量写入磁盘时，日志应该足够大，以应对运行时间最长的事务。

如果运行时间最长的事务运行 N 秒，并且队列管理器每秒写入磁盘的最大数据量在日志中为 B 字节/秒，那么日志应该至少为:

```
logsize >= 2 * (N+1) * B
```

当您在峰值工作负载上运行时，队列管理器可能正在将每秒的最大数据量写入磁盘，也可能是在记录介质映像时。

如果事务运行时间过长，以至于包含其第一个日志记录的日志扩展数据块未包含在活动日志中，那么队列管理器将从具有最早日志记录的事务开始，一次回滚一个活动事务。

在使用最大数量的主文件和辅助文件之前，队列管理器需要使旧的日志扩展数据块处于不活动状态，并且队列管理器需要分配另一个日志扩展数据块。

决定在允许队列管理器回滚之前，您希望运行时间最长的事务的时间长度。 运行时间最长的事务可能正在等待速度较慢的网络流量，或者在事务设计不佳的情况下，正在等待用户输入。

您可以通过发出以下 **runmqsc** 命令来调查运行时间最长的事务的运行时间:

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

发出 dspmqtrn -a 命令将显示处于所有状态的所有 XA 和非 XA 命令。

发出此命令将列出为所有当前事务写入第一个日志记录的日期和时间。

注意: 为了计算日志大小，重要的是自第一个日志记录写入以来的时间，而不是自应用程序或事务启动以来的时间。 将运行时间最长的事务的长度向上舍入到最接近的秒。 这是因为在队列管理器中进行了优化。

第一个日志记录可以在应用程序启动后很长时间内写入，如果应用程序启动，例如发出 MQGET 调用，在实际获取消息之前等待一段时间。

通过查看观察到的最大日期和时间输出

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

您最初发出的命令，从当前日期和时间开始，您可以估算运行时间最长的事务的运行时间。

确保在运行时间最长的事务处于峰值工作负载时重复运行此 **runmqsc** 命令，以便您不会低估运行时间最长的事务的长度。

在 IBM MQ 8.0 中，使用操作系统工具，例如 UNIX 平台上的 **iostat** 。

从 IBM MQ 9.0 开始，您可以通过发出以下命令来发现队列管理器每秒写入日志的字节数:

```
amqsrua -m qmgr -c DISK -t Log
```

写入的逻辑字节数，显示队列管理器每秒写入日志的字节数。 例如:

```
$ amqsrua -m mark -c DISK -t Log
Publication received PutDate:20160920 PutTime:15383157 Interval:4 minutes,39.579 seconds
Log - bytes in use 37748736
Log - bytes max 50331648
Log file system - bytes in use 316243968
Log file system - bytes max 5368709120
Log - physical bytes written 4334030848 15501948/sec
Log - logical bytes written 3567624710 12760669/sec
Log - write latency 411 uSec
```

在此示例中，每秒写入日志的逻辑字节数为 12760669/sec 或每秒大约 12 个 MiB 。

使用

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

显示运行时间最长的事务为:

```
CONN(57E14F6820700069)
EXTCONN(414D51436D61726B2020202020202020)
TYPE(CONN)
APPLTAG(msginteg_r)                     UOWLOGDA(2016-09-20)
UOWLOGTI(16.44.14)
```

由于当前日期和时间为 2016-09-20 16.44.19，因此此事务已运行 5 秒。 但是，在队列管理器将事务回滚之前，需要允许这些事务运行 10 秒。 因此，您的日志大小应该是:

```
2 * (10 + 1) * 12 = 264 MiB
```

.

日志文件数必须能够包含最大的预期日志大小 (在前面的文本中计算)。 这将是:

最小日志文件数 = (所需日志大小)/(**LogFilePages** * 日志文件页面大小 (4096))

使用缺省 **LogFilePages**(4096) 和 264MiB 的日志大小估算 (在前面的文本中计算) ，日志文件的最小数目应该为:

```
264MiB / (4096 x 4096) = 16.5
```

即， 17 个日志文件。

如果您调整日志大小，以便预期的工作负载在主文件中运行:

- 在需要额外的日志空间时，辅助文件提供了一些应急操作。
- 循环日志记录始终使用预先分配的主文件，这比分配和取消分配辅助文件的速度略快。
- 队列管理器仅使用主文件中剩余的空间来计算何时采用下一个检查点。

因此，在前面的示例中，设置以下值，以便工作负载在主日志文件中运行:

- **LogFilePages** = 4096
- **LogPrimaryFiles** = 17
- **LogSecondaryFiles** = 5

请注意下列事项:

- 在此示例中， 5 辅助日志超过活动日志空间的 20%。

  从 IBM MQ 9.1.0 开始，记录器尝试将工作负载单独保留在主文件中。 因此，当一小部分主文件已满时，记录器会调度检查点。

  具有辅助文件是一种意外情况，以防有任何意外长时间运行的事务。

  您应该知道，当使用的日志空间占总日志空间的 80% 以上时，队列管理器将执行操作以减少日志空间使用量。
- 无论您使用的是线性日志记录还是循环日志记录，都请执行相同的计算。

  无论您是在计算线性活动日志还是循环活动日志的大小，都没有区别，因为活动日志的概念在线性日志记录和循环日志记录中都是相同的。
- 仅介质恢复所需的日志扩展数据块不在活动日志中，因此不会计入主文件数和辅助文件数。
- 从 IBM MQ 9.1.0 开始， DISPLAY QMSTATUS LOG 的 *LOGUTIL* 字段可用于帮助您计算大约所需的活动日志大小。

  此字段旨在允许您对所需日志大小进行合理估计，而无需不断采样，以便确定运行时间最长的事务的持续时间或队列管理器的峰值吞吐量。

我的 LogFile 页面应该有多大?

通常，使 LogFile 页面足够大，以便您能够在不达到最大主文件数的情况下轻松增加活动日志的大小。 几个大的日志文件比许多小的日志文件更可取，因为几个大的日志文件使您能够更灵活地增加您需要的日志大小。

对于线性日志记录，非常大的日志文件可能会生成性能变量。 对于非常大的日志文件，创建和格式化新的日志文件或归档旧的日志文件有一个更大的步骤。 这更多是手动和归档日志管理的问题，因为使用自动日志管理时，很少创建新的日志文件。

如果我使我的日志太小会发生什么?

估算日志的最小大小时需要考虑的点。

如果使日志太小:

- 长时间运行的事务将回退。
- 下一个检查点要在上一个检查点结束之前启动。

要点: 无论您对日志大小的估算多么不准确，都将保持数据完整性。

请参阅 第 529 页的『使用检查点来确保完全恢复』 以获取检查点的说明。 如果活动日志扩展数据块中剩余的日志空间量变短，那么队列管理器会更频繁地调度检查点。

检查点需要一些时间; 它不是瞬时的。 需要在检查点中记录的数据越多，检查点所花费的时间越长。 如果日志是小型检查点，那么可能重叠，这意味着在先前检查点结束之前请求下一个检查点。 如果发生此情况，那么将写入错误消息。

如果长时间运行的事务回退，或者检查点重叠，那么队列管理器将继续处理工作负载。 短期事务继续正常运行。

但是，队列管理器未以最佳方式运行，并且性能可能下降。 您应该使用足够的日志空间重新启动队列管理器。

如果我使日志过大会发生什么情况?

估算日志的最大大小时需要考虑的点。

如果使日志过大:

- 您可能会增加紧急重新启动所需的时间，尽管这不太可能。
- 您正在使用不必要的磁盘空间。
- 允许长时间运行的事务。

要点: 无论您对日志大小的估算多么不准确，都将保持数据完整性。

为了帮助您估算日志的最大大小，可以使用日志利用率统计信息。 有关更多信息，请参阅 第 541 页的『决定如何设置 IMGLOGLN 和 IMGINTVL』 和 ALTER QMGR。

有关队列管理器在重新启动时如何读取日志的描述，请参阅 第 529 页的『使用检查点来确保完全恢复』 。 队列管理器将从最后一个检查点重放日志，然后解析在队列管理器结束时处于活动状态的所有事务。

要解析事务，队列管理器将读取与该事务关联的所有日志记录。 这些日志记录可能早于最后一个检查点。

通过为队列管理器分配非常大的日志，您将授予队列管理器在重新启动时读取日志记录的许可权，尽管通常队列管理器不必执行此操作。 在不太可能发生这种情况的情况下，该过程可能需要很长时间。

如果检查点在队列管理器结束之前意外停止，那么这将显着增加具有大型日志的队列管理器的重新启动时间。 限制日志大小会限制紧急重新启动时间。

要避免这些问题，应确保:

- 您的工作负载可以轻松地装入不过大的日志中。
- 避免长时间运行事务。

我的日志文件系统应该有多大?

估算队列管理器所需的日志文件系统大小。

请务必使日志文件系统足够大，以便队列管理器有足够的空间来写入其日志。 如果队列管理器完全填充日志文件系统，那么它将写入 FFDC，回滚事务，并且可能会突然终止队列管理器。

您为日志预留的磁盘空间量必须至少与活动日志一样大。 具体有多大取决于:

- 您选择的日志类型 (线性或循环)
- 活动日志的大小 (主文件，辅助文件和日志文件页面)
- 您选择的日志管理 (手动，自动或归档)
- 在对象损坏的情况下，您的应急计划。

如果选择循环日志，那么日志文件系统应该

```
LogFilesystemSize >= (PrimaryFiles + SecondaryFiles + 1) * LogFileSize
```

这允许队列管理器写入所有主文件和辅助文件。 在特殊情况下，队列管理器可能会写入超出辅助数据块数的额外扩展数据块。 上述算法会将此考虑在内。

如果选择线性日志，那么日志文件系统应显着大于活动日志。

如果选择手动日志管理，那么队列管理器将根据需要继续写入新的日志扩展数据块，并且您负责在不再需要这些扩展数据块时将其删除 (并将其归档)。

日志文件系统需要的大小在很大程度上取决于您删除多余或不活动扩展数据块的策略。

您可以决定在扩展数据块变为不活动状态 (不需要重新启动恢复) 时立即对其进行归档和删除，也可以决定仅归档和删除多余的扩展数据块 (不需要介质或重新启动恢复)。

如果仅归档和删除多余的扩展数据块，并且具有损坏的对象，那么 **MEDIALOG** 将不会前进，因此不会有更多扩展数据块变为多余的扩展数据块。 您将停止归档和删除扩展数据块，直到您解决问题 (可能通过恢复对象)。

除非停止工作负载，否则需要多少时间来解决问题取决于日志文件系统的大小。 因此，最好在使用线性日志记录时使用慷慨的日志文件系统。

如果选择线性日志以及自动或归档日志管理，那么队列管理器将复用日志扩展数据块。

可供复用的日志扩展数据块以字母 R 作为前缀。 记录介质映像时，由于归档了多余的扩展数据块，因此队列管理器可以复用这些扩展数据块。

因此，复用扩展数据块小于在介质映像之间写入日志的数据长度:

```
ReuseExtents <= LogDataLengthBetweenMediaImages
```

当自动记录介质映像并设置 **IMGLOGLN** 时， LogDataLengthBetweenMediaImages 可以是两倍 **IMGLOGLN**，因为 **IMGLOGLN** 是目标而不是固定最大值。

手动记录介质映像或按时间间隔自动记录介质映像时， LogDataLengthBetweenMediaImages 取决于您的工作负载以及获取映像之间的时间间隔。

除了活动扩展数据块和复用扩展数据块外，还有不活动扩展数据块 (仅介质恢复需要) 和多余扩展数据块 (重新启动或介质恢复不需要)。

使用自动或归档日志管理时，队列管理器不会复用介质恢复所需的扩展数据块。 因此，不活动扩展数据块的数量取决于您获取介质映像的频率，以及您是手动还是自动获取这些映像。

IMGINTVL 和 **IMGLOGLN** 是目标，不是介质映像之间的固定最小值或最大值。 但是，在估算您可能需要的日志文件系统的最大大小时，不太可能将自动介质映像记录到两次以上 **IMGINTVL** 或 **IMGLOGLN**。

使用自动或归档日志管理来调整日志文件系统的大小时，还应考虑在队列或其他对象损坏时可能发生的情况。 在这种情况下，队列管理器无法获取受损对象的介质映像，并且 **MEDIALOG** 将不会向前移动。

如果工作负载继续，那么不活动日志将无节制地增长，因为仍需要介质恢复所需的最旧扩展数据块，并且无法复用。 如果工作负载继续，那么在队列管理器开始回滚事务，甚至可能突然结束之前，您将需要直到日志文件系统完全填满以解决问题。

因此，对于自动日志管理和归档日志管理:

```
LogFilesystemSize > (PrimaryFiles + SecondaryFiles +
(((TimeBetweenMediaImages *2) + TimeNeededToResolveDamagedObject) * ExtentsUsedPerHour))
* LogFilePages
```

注: 上述算法假定只要不再需要每个扩展数据块进行介质恢复，就会针对每个扩展数据块调用 **SET LOG ARCHIVED** 以进行归档日志管理。

管理日志

从 IBM MQ 9.1.0 开始，产品支持线性日志的自动日志管理和自动介质恢复。 循环日志几乎是自我管理，但有时需要干预来解决空间问题。

注: IBM i 自动日志管理和归档日志管理在 IBM i 上无效。

在循环日志记录时，队列管理器将回收日志文件中的已释放空间。 此活动对于用户来说并不明显，您通常不会看到已使用的磁盘空间量减少，因为分配的空间会快速复用。

从 IBM MQ 9.1.0 中，可以在使用循环日志记录时删除辅助文件。 请参阅 RESET QMGR **TYPE(REDUCELOG)** 以获取更多信息。

在线性日志记录中，如果长时间未使用检查点，或者如果长时间运行的事务很久以前写入了日志记录，那么日志可能会填充。 队列管理器尝试经常执行检查点以避免第一个问题。

Multi 如果日志已满，那么将发出消息 AMQ7463 。 此外，如果日志由于长时间运行的事务已阻止释放空间而填满，那么将发出消息 AMQ7465 。

在日志记录中，只有自上次完成检查点启动以来写入的日志记录以及由任何活动事务写入的日志记录才需要重新启动队列管理器。

随着时间的推移，写入的最旧日志记录将不需要重新启动队列管理器。

检测到长时间运行的事务时，将调度活动以异步回滚该事务。 如果由于某些意外原因导致异步回滚失败，那么在此情况下，某些 MQI 调用将返回 MQRC_RESOURCE_PROBLEM。

请注意，将保留空间以落实或回滚所有进行中的事务，因此 **MQCMIT** 或 **MQBACK** 不应失败。

以这种方式回滚事务的应用程序无法执行在同一事务下指定同步点的后续 **MQPUT** 或 **MQGET** 操作。

尝试在此状态的同步点下放置或获取消息将返回 MQRC_BACKED_OUT。 然后，应用程序可以发出 **MQCMIT**(返回 MQRC_BACKED_OUT) 或 **MQBACK** 并启动新事务。 当耗用过多日志空间的事务已回滚时，将释放日志空间，并且队列管理器将继续正常运行。

自动介质成像

如果使用线性日志记录，可以定期拍摄媒体图像。 这些图像可以让您在对象损坏时进行恢复。

默认情况下，介质成像调度是一个手动过程。 您还可以将介质成像配置为自动。 如果配置了自动介质成像，队列管理器会在工作量允许时安排成像。 简化了管理，并将对性能的影响降至最低。

您可以使用 **ALTER QMGR**MQSC 命令设置以下队列管理器属性，从而启用自动介质成像功能，并定义队列管理器如何实现该功能：

IMGSCHED
将此属性设置为 "AUTO 可启用队列管理器的自动介质成像功能。 如果启用了自动介质成像，则可使用这两个属性来调整拍摄图像的时间：

IMGINTVL
指定记录介质图像的目标频率。 以分钟为单位指定数值，默认为 60 分钟。 设置为 "0FF 表示不使用时间间隔来影响介质成像。

IMGLOGLN
指定写入日志的目标数量，然后拍摄介质图像。 指定以 MB 为单位的值，设置为 "0FF 则不使用写入的日志量来影响介质成像。 缺省值为 0FF。

IMGRCOVO
使用此属性可指定可恢复的对象类型。 将此属性设置为 "YES，可获取身份验证信息、通道、客户端连接、监听器、命名列表、进程、别名队列、远程队列和服务对象的媒体图像。

IMGRCOVQ
使用此属性可设置本地队列和 PERMADYN 队列类型的默认值。 设置为 "YES 可拍摄媒体图像，否则设置为 "NO。

相关信息
ALTER QMGR（更改队列管理器设置）

自动日志管理
您可以为线性日志配置自动日志管理，以便重复使用日志扩展。

默认情况下，必须手动管理线性日志，这意味着必须删除或归档恢复中不再需要的日志文件，以避免磁盘空间耗尽。

如果启用自动日志管理，日志扩展会自动循环使用，因此不会出现空间不足的情况。

您可以在创建队列管理器时指定日志管理设置，也可以编辑 "qm.ini 文件来更改现有队列管理器的日志管理。 您可以从三种日志管理模式中选择一种：

手动
日志扩展由人工管理。 这是缺省设置。

自动
当重启或介质恢复不再需要日志扩展时，队列管理器会自动重复使用日志扩展

归档
这与 Automatic 相同，只是队列管理器在重用之前会等待通知，说明某个范围已被归档。

创建队列管理器时，可通过使用 "**crtmqm** 命令的以下参数来指定日志管理设置：

-lla
指定使用自动日志管理。

-lln
指定使用归档日志管理。

-ll
指定使用手动日志管理（默认值）。

对于现有队列管理器，可以通过编辑队列管理器 "qm.ini 文件的 "Log 节来更改日志管理设置。 添加以下设置之一：

LogManagement=Automatic
指定使用自动日志管理。

LogManagement=Archive
指定使用归档日志管理。

如果选择归档日志管理方法，则必须在归档日志范围时通知队列管理器。 使用以下命令进行此操作：

```
SET LOG ARCHIVED(logfilename)
```

您也可以使用 PCF 命令 "**MQCMD_SET_LOG**。

相关信息
crtmqm（创建队列管理器）
多平台上的 SET LOG（通知日志归档完成
MQCMD_SET_LOG（通知日志存档完成），关于 "AIX, Linux, and Windows

磁盘变满时发生的情况
当队列管理器配置为使用线性日志记录时，队列管理器日志记录组件将通过以下方式对磁盘已满情况作出反应。

如果包含日志文件的磁盘变满，那么：

- 仅当创建所需大小的新日志文件时，队列管理器才会发现此情况，在需要此文件时，它会提前执行此操作。
- 当操作系统从将文件扩展至所需大小的请求中返回错误时，它将发现磁盘已满情况。

- 队列管理器向队列管理器错误日志发出消息 AMQ6708 。
- 首次故障支持技术 (FFST) 记录将写入系统范围的错误目录。 此记录提供磁盘已满情况的详细信息，如果您需要联系 IBM 支持人员，那么应保留此记录。

将以固定大小创建日志文件，而不是随着日志记录写入日志文件而扩展日志文件。 这意味着 IBM MQ 只能在创建新文件时耗尽磁盘空间; 不能在将记录写入日志时耗尽空间。 IBM MQ 始终知道现有日志文件中有多少可用空间，并相应地管理文件中的空间。

从 IBM MQ 9.1.0 开始，使用线性日志记录时，您可以选择使用:

- 日志扩展数据块的自动管理。

  有关新日志属性的更多信息，请参阅 DISPLAY QMSTATUS 。

  另请参阅以下命令或其 PCF 等效命令:

  - RESET QMGR
  - 针对分布式平台的 SET LOG
- 控制媒体图像使用的选项。

  请参阅 ALTER QMGR 命令和 ALTER QUEUE 以获取有关以下内容的更多信息:

  - IMGINTVL
  - IMGLOGLN
  - IMGRCOVO
  - IMGRCOVQ
  - IMGSCHED

循环日志记录返回资源问题。

如果仍然耗尽空间，请检查队列管理器配置文件中的日志配置是否正确。 您可能可以减少主日志文件或辅助日志文件的数量，以便日志不会超出可用空间。

不能更改现有队列管理器的日志文件大小。 队列管理器要求所有日志扩展数据块大小相同。

管理日志文件

为日志文件分配足够的空间。 对于线性日志记录，您可以在不再需要旧日志文件时将其删除。

特定于循环日志记录的信息

如果您正在使用循环日志记录，请确保在配置系统时有足够的空间来保存日志文件 (请参阅 第 79 页的『mqs.ini 文件的 LogDefaults 节』 和 第 108 页的『qm.ini 文件的日志节』)。 日志所使用的磁盘空间量不会超出配置的大小，包括需要时要创建的辅助文件的空间。

特定于线性日志记录的信息

如果您正在使用线性日志，那么日志文件会随着数据的记录而不断添加，并且使用的磁盘空间量会随着时间而增加。 如果记录数据的速率很高，那么新的日志文件将快速使用磁盘空间。

随着时间的推移，不再需要线性日志的较旧日志文件来重新启动队列管理器或对任何损坏的对象执行介质恢复。 以下方法确定仍需要哪些日志文件:

记录器事件消息
发生重要事件 (例如，记录媒体图像) 时，将生成记录器事件消息。 记录器事件消息的内容指定队列管理器重新启动和介质恢复仍需要的日志文件。 有关记录器事件消息的更多信息，请参阅 记录器事件

队列管理器状态
运行 MQSC 命令 DISPLAY QMSTATUS 或 PCF 命令 INQUIRE QUEUE MANAGER STATUS 将返回队列管理器信息，包括所需日志文件的详细信息。 有关 MQSC 命令的更多信息，请参阅 使用 MQSC 命令进行管理，有关 PCF 命令的信息，请参阅 自动化管理任务。

队列管理器消息
队列管理器定期发出一对消息以指示需要哪些日志文件:

- 消息 AMQ7467I 提供重新启动队列管理器所需的最旧日志文件的名称。 此日志文件和所有更新的日志文件必须在队列管理器重新启动期间可用。
- 消息 AMQ7468I 提供了介质恢复所需的最旧日志文件的名称。

要确定 "较旧" 和 "较新" 的日志文件，请使用日志文件编号而不是文件系统应用的修改时间。

适用于两种类型的日志记录的信息

只有队列管理器重新启动所需的日志文件，活动日志文件才需要联机。 可以将不活动的日志文件复制到归档介质 (例如用于灾难恢复的磁带) ，并从日志目录中除去这些文件。 介质恢复不需要的不活动日志文件可被视为多余的日志文件。 如果您的操作不再关注多余的日志文件，那么可以将其删除。

如果找不到需要的任何日志文件，那么将发出操作员消息 AMQ6767E 。 使日志文件和所有后续日志文件可供队列管理器使用，然后重试该操作。

自动清除日志扩展数据块-仅线性日志记录

Multi

从 IBM MQ 9.1.0 开始，您可以选择使用恢复不再需要的线性日志扩展数据块的自动管理。

使用 qm.ini 文件的 Log 节中的 **LogManagement** 属性，或者使用 IBM MQ Explorer 来设置自动管理。 有关更多信息，请参阅 第 108 页的『qm.ini 文件的日志节』。

请参阅 **DISPLAY QMSTATUS** 的 LOG 参数，以获取有关日志操作的更多详细信息，以及有关使用日志的以下命令:

- RESET QMGR
- 设置日志

自动获取介质映像-仅线性日志记录

从 IBM MQ 9.1.0 开始，提供了一个整体开关来控制队列管理器是否自动写入介质映像，缺省情况是尚未设置该开关。

您可以使用以下队列管理器属性来控制是否发生自动介质成像以及进程的频率:

IMGSCHED
队列管理器是否自动写入介质映像

IMGINTVL
写入媒体图像的频率 (以分钟为单位)

IMGLOGLN
自对象的先前介质映像以来写入的日志兆字节数。

如果在工作负载非常繁重的一天中有一个关键时间，并且您希望确保系统吞吐量不会因获取自动介质映像而受到影响，那么您可能希望通过设置 **IMGSCHED***(MANUAL)*来临时关闭自动介质映像。

您可以在工作负载期间随时切换 **IMGSCHED** 。

注意: 如果未使用介质映像，那么将不会向前移动 **MEDIALOG** ，因此您需要归档扩展数据块或确保有足够的磁盘空间。

您还可以控制其他用户定义对象的自动和手动介质映像:

- 认证信息
- 通道
- 客户机连接
- 侦听器
- 名称列表
- 进程
- 别名队列

- 本地队列
- 服务
- Topic

对于内部系统对象 (例如对象目录和队列管理器对象) ，队列管理器会根据需要自动写入介质映像。

有关属性的更多信息，请参阅 ALTER QMGR 。

您还可以仅对本地和永久动态队列启用或禁用自动和手动介质映像。 您可以使用 **IMGRCOVQ** 队列属性来执行此操作。

有关 **IMGRCOVQ** 属性的更多信息，请参阅 ALTER QUEUE 。

注意:

1. 仅当您使用线性日志记录时，才支持介质映像。 如果您已启用自动介质映像，但正在使用循环日志记录，那么将发出错误消息并禁用队列管理器的自动介质映像属性。
2. 如果已启用自动介质映像，但未指定日志的频率 (分钟或兆字节) ，那么将发出错误消息，并且不会写入自动介质映像。
3. 如果需要，可以使用 rcdmqimg 在设置 **IMGSCHED***(AUTO)*时手动记录介质映像。

   这使您能够在适合您企业的时候 (例如，当您的系统安静时) 拍摄媒体图像。 自动媒体成像会考虑这些手动媒体图像，因为采用手动媒体图像会重置时间间隔和日志长度，在此之前会拍摄下一个自动媒体图像。
4. 从 IBM MQ 9.1.0 开始，队列管理器仅在介质映像中写入持久消息，而不写入非持久消息。 这可以在迁移到 IBM MQ 9.1.0 或更高版本时减小介质映像的大小

决定如何设置 IMGLOGLN 和 IMGINTVL

使 **IMGLOGLN** 和 **IMGINTVL** 足够大，因此队列管理器仅花费一小部分时间来记录介质映像，但足够小，以便:

- 损坏的物体可以在合理的时间内恢复，并且
- 足够小，以便您的日志适合您的磁盘，而不会耗尽空间。

如果设置 **IMGLOGLN**，那么最佳实践是使 **IMGLOGLN** 是队列上的数据量的许多倍，是工作负载的数据率的许多倍。 使 **IMGLOGLN** 越大，队列管理器记录介质映像所花费的时间越少。

同样，如果设置 **IMGINTVL**，那么最佳实践是使 **IMGINTVL** 是队列管理器记录介质映像所花费的时间量的很多倍。 您可以通过手动记录介质图像来了解记录介质图像所需的时间。

如果使 **IMGLOGLN** 和 **IMGINTVL** 过大，那么恢复受损对象可能需要很长时间，因为必须重放自上次介质映像以来的所有扩展数据块。

使 **IMGLOGLN** 和 **IMGINTVL** 足够小，以便您可以接受恢复受损对象所花费的最大时间。

使 **IMGLOGLN** 和 **IMGINTVL** 非常大，意味着日志会非常大，因为很少记录介质图像。

注意: 确保此大小的日志适合您的日志文件系统，因为如果日志文件系统完全填满，那么您的工作负载将回退。

您可以同时设置 **IMGINTVL** 和 **IMGLOGLN**。 这可能有助于确保在繁重工作 (由 **IMGLOGLN** 控制) 期间定期获取自动介质映像，但在工作负载非常轻 (由 **IMGINTVL** 控制) 时仍会偶尔获取自动介质映像。

IMGINTVL 和 **IMGLOGLN** 是获取自动介质映像的时间间隔和日志数据长度的目标。

不应将这些属性视为固定最大值或最小值。 事实上，如果队列管理器认为这是一个非常好的时机，那么队列管理器可能会决定提前调度自动介质映像:

- 因为队列是空的，所以取介质映像在性能方面是最高效的，并且
- 一段时间未记录媒体图像

有时，自动介质映像之间的间隔可能比 **IMGINTVL** 和/或 **IMGLOGLN** 之间的间隔长一些。

如果队列上的数据量接近 **IMGLOGLN**，那么介质映像之间的间隔可能大于 **IMGLOGLN** 。 如果记录介质映像所需的时间几乎与 **IMGINTVL** 相同，那么介质映像之间的间隔可能大于 **IMGINTVL** 。

这是糟糕的做法，因为队列管理器将花费大量时间记录介质图像。

当使用自动媒体图像记录时，队列管理器会为每个对象分别记录一个媒体图像和队列，因此队列管理器会为每个对象分别跟踪图像之间的时间间隔和日志长度。

随着时间的推移，媒体图像的记录变得交错，而不是同时记录所有对象的媒体图像。 这一惊人的传播了记录媒体图像的性能影响，是使用自动记录媒体图像比人工记录的另一个优势。

手动获取介质映像-仅线性日志记录

记录队列的介质映像涉及将该队列中的所有持久消息写入日志。 对于包含大量消息数据的队列，这涉及将大量数据写入日志，而此过程会在系统发生时影响其性能。

记录其他对象的媒体图像可能比较快，因为其他对象的媒体图像不包含用户数据。

您需要仔细考虑何时记录队列的介质映像，以便该过程不会干扰您的峰值工作负载。

必须定期记录所有对象的介质映像，以便更新介质恢复所需的最早日志扩展数据块。

记录队列的介质映像的良好时间是在其为空时，因为此时不会将任何消息数据写入日志。 反之，一个不好的时间是当队列很深或上面有非常大的消息时。

记录队列的介质映像的最佳时间是在系统处于静默状态时; 而错误时间是在峰值工作负载期间。 例如，如果您的工作负载在午夜始终处于安静状态，那么您可能决定在每晚的午夜记录媒体图像。

交错记录每个队列可将性能影响传播出去，从而降低其效果。 自上次记录介质映像以来的时间越长，记录这些映像就越重要，因为介质恢复所需的日志扩展数据块数量正在增加。

注: 执行介质恢复时，必须同时在日志文件目录中提供所有必需的日志文件。 确保获取您可能想要恢复的任何对象的常规介质映像，以避免耗尽磁盘空间来存放所有必需的日志文件。

例如，要获取队列管理器中所有对象的介质映像，请运行 **rcdmqimg** 命令，如以下示例中所示:

Windows **在 Windows 上**

```
rcdmqimg -m QMNAME -t all *
```

Linux AIX **在 AIX and Linux 上**

```
rcdmqimg -m QMNAME -t all "*"
```

运行 **rcdmqimg** 会将介质日志序号 (LSN) 向前移动。 有关日志序号的更多详细信息，请参阅 第 548 页的『使用 dmpmqlog 命令转储日志内容』。 **rcdmqimg** 不会自动运行，因此必须手动运行或从您创建的自动任务运行。 有关此命令的更多信息，请参阅 rcdmqimg 和 dmpmqlog。

如果您已选择将线性日志记录与队列管理器控制的自动介质映像配合使用，那么不需要使用 **rcdmqimg** 手动记录介质映像来管理日志空间。

注: 在运行 rcdmqimg 命令时，还可以发出消息 AMQ7467 和 AMQ7468 。

部分媒体图像

最好仅将 IBM MQ 消息用于预期在不久的将来使用的数据，以便每条消息在队列中的时间相对较短。

相反，使用 IBM MQ 消息长期存储数据 (如数据库) 的做法很差。

确保队列相对较浅也是好的做法，不好的做法是要有深度队列，其消息已在队列中排了很长时间。

通过遵循这些准则，您可以使队列管理器优化自动记录媒体图像的性能。

记录空队列的媒体图像非常高效 (从性能角度来看) ，而获取队列上包含大量数据的媒体图像非常低效，因为所有这些数据都必须写入媒体图像中的日志。

对于最近在其上放置消息的浅层队列，队列管理器可以进行进一步优化。

如果队列上当前的所有消息都是在最近放入的，那么队列管理器可能能够在放入所有消息之前代表某个时间(恢复点) 记录介质映像，因此能够记录空队列的映像。 这个过程在性能方面是非常低的成本。

如果随后获取了位于恢复点的队列上的所有消息，那么无需将这些消息记录在介质映像中，因为它们不再位于队列上。

这称为 部分介质映像。 然后，在不太可能需要恢复队列的情况下，将重放自上次介质映像以来与此队列相关的所有日志记录，以便复原所有最近放入的消息。

即使在恢复点的队列上存在一些当前在队列上的消息 (因此必须记录在部分介质映像中) ，记录此较小的部分介质映像仍然比记录所有消息的完整介质映像更高效。

确保消息在队列中保持短暂时间可能会提高自动记录媒体图像的性能。

确定多余的日志文件-仅线性日志记录
对于循环日志记录，从不从日志目录中删除数据。 管理线性日志文件时，请务必确保可以删除或归档哪些文件。 此信息将帮助您做出此决策。

请勿使用文件系统的修改时间来确定 "较旧" 的日志文件。 仅使用日志文件号。 队列管理器对日志文件的使用遵循复杂的规则，包括在需要日志文件之前对其进行预分配和格式化。 如果尝试使用这些时间来确定相对寿命，那么您可能会看到具有修改时间的日志文件会产生误导。

要确定所需的最旧的日志文件，有三个位置可供您使用:

- DISPLAY QMSTATUS 命令
- 记录器事件消息，最后
- 错误日志消息

对于 DISPLAY QMSTATUS 命令，要确定需要的最旧的日志扩展数据块:

- 重新启动队列管理器，发出命令 `DISPLAY QMSTATUS RECLOG`。
- 执行介质恢复，发出命令 `DISPLAY QMSTATUS MEDIALOG`。
- 确定归档通知的名称，发出命令 `DISPLAY QMSTATUS ARCHLOG`。

通过发出命令 **RESET QMGR TYPE(REDUCELOG)**，可以减少使用循环日志记录时的辅助日志扩展数据块数。

通常，较低的日志文件编号意味着较旧的日志。 除非您的日志文件周转率非常高，否则在 10 年内每天 3000 个日志文件的顺序中，您不需要满足 9 999 999 的数字回绕。 在这种情况下，您可以归档数字小于 RECLOG 值的任何日志文件，并且可以删除数字小于 RECLOG 和 MELOG 值的任何日志文件。

注意: 日志文件回绕，因此 9 999 999 之后的下一个数字为零。

日志文件位置
为日志文件选择位置时，请记住，如果 IBM MQ 由于缺少磁盘空间而无法格式化新日志，那么该操作将受到严重影响。

如果您正在使用循环日志，请确保驱动器上有足够的空间至少用于配置的主日志文件。 还为至少一个辅助日志文件留出空间，如果日志必须增大，那么需要该空间。

如果您正在使用线性日志，那么将允许更多空间; 日志所消耗的空间会随着数据的记录而不断增加。

您应该将日志文件与队列管理器数据放在单独的磁盘驱动器上。

此设备上的数据完整性至关重要-您应该允许内置冗余。

还可以将日志文件放在镜像排列的多个磁盘驱动器上。 这可防止包含日志的驱动器发生故障。 如果没有镜像，可能会强制您返回到 IBM MQ 系统的最后一个备份。

V 9.2.0 *Coldstart:* 缺少日志扩展数据块或日志扩展数据块损坏时要执行的操作

如果您的企业丢失了重新启动恢复所需的部分或全部日志扩展数据块，那么队列管理器将无法重放恢复日志，因此无法重新启动。 如果您要求队列管理器在恢复日志以任何方式损坏时重新启动，但以维护数据完整性为代价，那么可以执行此操作，尽管强烈建议不要执行此操作。 此过程称为队列管理器 *coldstarting* 。

要点: 只有在特殊情况下才应考虑启动队列管理器，并且该队列管理器具有此页面上描述的数据完整性风险。 IBM 建议您重建队列管理器，以响应损坏的数据文件，而不使用 cold 面启动。

如果出于操作原因需要 coldstart ，请与 IBM 支持代表联系以查看问题的根本原因。 您应该尽早将冷启动队列管理器替换为重建的队列管理器。

Coldstart 的影响

在 coldstart 上，队列管理器创建空恢复日志，并依赖于队列文件和其他对象文件中处于现有状态的数据。 由于队列文件中的数据可能不一致，因此消息可能丢失，重复，损坏或不一致。

队列管理器将所有其他持久对象的配置存储在恢复日志以及对象文件中。 其他内部状态数据也记录在恢复日志中，因此在 coldstart 上，内部状态数据被重置，所有这些其他配置数据都可能不准确。

冷启动的影响是不可预测和广泛的，因此除非绝对必要，否则您应该避免冷启动。 在 cold 起始之后，队列和对象文件中的信息可能如此不一致，以至于队列管理器根本不会重新启动。

如果队列管理器重新启动，那么无法通过简单的方法来发现可以依赖哪些消息数据或配置以及哪些消息数据或配置无法依赖哪些消息数据或配置。 此外，在冷启动之后，队列可能会损坏，因此变得完全不可用。

此外，如果可以从特定队列获取消息或将其放入特定队列，那么该队列上的消息可能已损坏，缺失或重复。 事务和通道可能处于不确定状态。 即使队列管理器 coldstart 成功并且队列看起来完好无损，但 coldstart 的不可预测影响可能要到以后才能实现。

需要 coldstart 时要执行的操作

执行 coldstart 不应视为标准操作实践， IBM 强烈建议您不要执行此操作。 但是，如果您处于一定需要对队列管理器进行冷启动的位置，请联系 IBM MQ 支持 。

对于线性队列管理器，冷启动队列管理器的过程比循环队列管理器复杂得多。 在 IBM MQ 9.1.3 中，冷启动过程已大大简化，不再涉及复制或重命名日志扩展数据块。

从 IBM MQ 9.1.3 开始，请联系 IBM 支持人员，他们将为您提供一个密钥，您可以将该密钥传递到 **strmqm** 命令以冷启动队列管理器。

注意: IBM MQ 9.1.3 coldstart 命令仍然具有与手动 coldstart 相同的丢失数据完整性的风险， IBM 强烈建议您执行此操作。

消除未来的冷启动: 请求

strmqm 命令需要一个用于 coldstart 的密钥，因为 IBM MQ 希望您在需要 coldstart 时联系 IBM MQ 支持人员，因为 IBM MQ 非常希望了解您是如何进入此情境的。

显然冷启动是最好避免的事情。 IBM MQ 已尽了很大努力来确保您将不需要对队列管理器进行冷启动，并且 IBM 很希望发现产品是否还有其他功能可用来缓解必须进行冷启动的情况。

避免冷启动的预防措施

创建队列管理器时的缺省日志记录方法是循环日志记录。 通过循环日志记录，您允许队列管理器具有给定大小的特定数量的主日志扩展数据块和辅助日志扩展数据块。 创建足够大的日志文件系统以包含所有主日志扩展数据块和辅助日志扩展数据块，您不应该需要对其进行管理。

或者，您可以使用线性日志记录而不是循环。 线性日志记录使您能够在队列和其他对象不太可能损坏的情况下恢复这些队列和对象。 但缺省情况下，线性日志记录要求您删除重新启动或介质恢复不再需要的日志扩展数据块。 这称为手动日志管理。

以这种方式管理日志扩展数据块时，可能会无意中删除过多的日志扩展数据块，因此最终不得不进行 coldstart。 要降低此风险，请使用自动日志管理，以便队列管理器代表您管理日志扩展数据块。

最佳实践是将恢复日志放入仅包含恢复日志的单独日志文件系统中。 如果将恢复日志与队列管理器的其余部分放在同一文件系统中，那么有时可能会发现该文件系统意外填满，原因可能是队列文件较大。 使队列管理器的日志目录成为单独的文件系统，或者使用 **crtmqm** 命令上的 **-ld** 命令行选项指定不同的日志文件系统。

如果保存队列文件的文件系统已满，那么您可能无法放入这些队列，但队列管理器将继续运行。 如果包含恢复日志的文件系统已满，那么队列管理器将突然结束，直到您释放一些空间后才会重新启动。

请注意，不要删除重新启动恢复所需的日志扩展数据块，否则您可能会发现自己需要执行 coldstart。 有时，您可能会发现需要执行 coldstart ，因为包含其恢复日志的磁盘发生故障。 最佳实践是将恢复日志放在复制的磁盘上，从而降低磁盘崩溃的风险。

将消息和配置移至新的替换队列管理器可避免先前已冷启动的队列管理器发生持续问题的可能性。

请记下先前已冷启动的队列管理器，即使它们在很久以前已冷启动并且在此期间已停止，重新启动和迁移。 当您联系 IBM 支持人员时，假设队列管理器先前是否已被冷启动，如果已被冷启动，请提供尽可能多的信息，以了解导致冷启动需求的原因。

使用日志进行恢复

您可以使用日志中的信息来帮助您从故障中恢复。

有几种方法会损坏您的数据。 IBM MQ 可帮助您从以下位置恢复:

- 损坏的数据对象
- 系统中的电源损耗
- 通信故障

本节将查看如何使用日志从这些问题中恢复。

从电源丢失或通信故障中恢复

IBM MQ 可以从通信故障和断电中恢复。 它有时也可以从其他类型的问题中恢复，例如无意中删除文件。

在通信故障的情况下，持久消息将保留在队列中，直到接收应用程序将其除去为止。 如果正在传输该消息，那么它将保留在传输队列上，直到可以成功传输该消息为止。 要从通信故障中恢复，通常可以使用失败的链路重新启动通道。

如果失去电源，那么当重新启动队列管理器时， IBM MQ 会将队列复原到发生故障时的已落实状态。 这将确保不会丢失任何持久消息。 将废弃非持久消息; 当 IBM MQ 突然停止时，这些消息无法存活。

恢复受损对象

可以通过一些方法使 IBM MQ 对象变为不可用，例如，由于无意损坏。 然后，您必须恢复完整的系统或部分系统。 需要执行的操作取决于何时检测到损坏，所选日志方法是否支持介质恢复以及哪些对象已损坏。

介质恢复

从 IBM MQ 9.1.0 开始，在线性日志记录队列管理器上，只能为可恢复的对象记录介质映像。 例如，您需要考虑 **IMGRCOVO** 和 **IMGRCOVQ** 选项。

同样，只能从线性日志记录队列管理器上的对象的介质映像中恢复定义为介质可恢复的对象子集。 如果未定义为介质可恢复的对象已损坏，那么该对象的选项与循环日志记录队列管理器的选项相同。

介质恢复根据线性日志中记录的信息重新创建对象。 例如，如果无意中删除了对象文件，或者由于某种其他原因而变得不可用，那么介质恢复可以重新创建该文件。 对象的介质恢复所需的日志中的信息称为 介质映像。

介质映像是一系列日志记录，其中包含可从其重新创建对象本身的对象的映像。

重新创建对象所需的第一个日志记录称为其 介质恢复记录 ; 它是对象的最新媒体图像的开头。 每个对象的介质恢复记录是检查点期间记录的信息片段之一。

根据对象的介质映像重新创建该对象时，还需要重放描述自上一个映像以来对该对象执行的更新的任何日志记录。

例如，考虑具有在将持久消息放入队列之前获取的队列对象图像的本地队列。 为了重新创建对象的最新图像，除了重放图像本身外，还需要重放记录将消息放入队列的日志条目。

创建对象时，写入的日志记录包含足够的信息以完全重新创建该对象。 这些记录构成对象的第一个介质映像。 然后，在每次关闭时，队列管理器会自动记录介质映像，如下所示:

- 非本地的所有进程对象和队列的图像
- 空本地队列的图像

还可以使用 **rcdmqimg** 命令手动记录介质映像，如 rcdmqimg 中所述。 此命令将写入 IBM MQ 对象的介质映像。

如果设置了 **IMGSCHED***(AUTO)* ，那么队列管理器会自动记录介质映像。 有关更多信息，请参阅 ALTER QMGR 以获取有关 **IMGINTVL** 和 **INGLOGLN** 的信息。

当已写入介质映像时，仅需要保存介质映像的日志以及在此时间之后创建的所有日志才能重新创建损坏的对象。 创建介质映像的优点取决于可用存储量以及创建日志文件的速度等因素。

从介质映像恢复

在队列管理器启动期间，队列管理器会自动从其介质映像中恢复某些对象。 如果队列涉及在队列管理器上次关闭时未完成的任何事务，并且在重新启动处理期间发现该事务已损坏或已损坏，那么它将自动恢复该队列。

您必须使用 **rcrmqobj** 命令手动恢复其他对象，该命令将重放日志中的记录以重新创建 IBM MQ 对象。 将根据在日志中找到的对象的最新映像以及在保存映像到发出重新创建命令之间的所有适用日志事件来重新创建该对象。 如果 IBM MQ 对象损坏，那么可以执行的唯一有效操作是删除该对象或通过此方法重新创建该对象。 无法以这种方式恢复非持久消息。

请参阅 rcrmqobj 以获取 **rcrmqobj** 命令的更多详细信息。

当尝试对对象进行介质恢复时，包含介质恢复记录的日志文件以及所有后续日志文件必须在日志文件目录中可用。 如果找不到必需的文件，那么将发出操作员消息 AMQ6767 ，并且介质恢复操作将失败。 如果不获取要重新创建的对象的常规介质映像，那么可能没有足够的磁盘空间来存放重新创建对象所需的所有日志文件。

存在哪些对象文件

队列管理器将 **runmqsc** 中定义的对象的属性存储在磁盘上的文件中。 这些对象文件位于队列管理器的数据目录下的子目录中。

Linux AIX 例如，在 AIX and Linux 平台上，通道存储在 `/var/mqm/qmgrs/qmgr/channel` 中。

这些对象文件中的数据是对象的介质图像。 如果这些对象文件被删除或损坏，那么存储在该文件中的对象将损坏。 通过使用线性日志记录队列管理器，可以使用 rcrmqobj 命令从日志中恢复受损对象。

大多数对象文件仅包含对象的属性，因此通道文件包含通道的属性。 例外情况如下所示：

- 目录

  对象目录对所有类型的所有对象进行编目，并存储在 `qmanager/QMQMOBJCAT` 中。
- 同步文件

  同步文件包含与所有通道关联的内部状态数据。
- 队列

  队列文件包含该队列上的消息以及该队列的属性。

请注意， **runmqsc** 或 IBM MQ Explorer 中未公开目录或同步文件对象。

可以记录目录和队列管理器，但不能恢复。 如果这些对象损坏，那么队列管理器将提前结束，并且这些对象将在重新启动时自动恢复。

由于持久预订存储在系统队列中，因此未在要记录或恢复的对象中列出预订。 要记录或恢复持久预订，请记录或恢复 SYSTEM.DURABLE.SUBSCRIBER.QUEUE 。

在启动期间恢复受损对象

如果队列管理器在启动期间发现损坏的对象，那么它所执行的操作取决于对象的类型以及队列管理器是否配置为支持介质恢复。

如果队列管理器对象已损坏，那么除非队列管理器可以恢复该对象，否则无法启动该队列管理器。 如果队列管理器配置了线性日志，因此支持介质恢复，那么 IBM MQ 会自动尝试从其介质映像重新创建队列管理器对象。 如果选择的日志方法不支持介质恢复，那么可以复原队列管理器的备份或删除队列管理器。

如果在队列管理器停止时有任何事务处于活动状态，那么包含放入或进入这些事务的持久未落实消息的本地队列也需要成功启动队列管理器。 如果发现其中任何一个本地队列已损坏，并且队列管理器支持介质恢复，那么它会自动尝试从其介质映像重新创建这些队列。 如果无法恢复任何队列，那么 IBM MQ 无法启动。

如果在不支持介质恢复的队列管理器上启动处理期间发现包含未落实消息的任何受损本地队列，那么会将这些队列标记为受损对象，并忽略其上的未落实消息。 这种情况是因为无法对此类队列管理器上的受损对象执行介质恢复，剩下的唯一操作是删除这些对象。 将发出消息 AMQ7472 以报告任何损坏。

在其他时间恢复受损对象

仅在启动期间自动恢复对象的介质。 在其他情况下，当检测到对象损坏时，将发出操作员消息 AMQ7472 ，并且使用该对象的大多数操作将失败。 如果队列管理器对象在队列管理器启动后的任何时候损坏，那么队列管理器将执行先发制人的关闭。 当对象已损坏时，您可以将其删除，如果队列管理器正在使用线性日志，请尝试使用 **rcrmqobj** 命令从其介质映像中恢复该对象 (请参阅 rcrmqobj 以获取更多详细信息)。

如果队列 (或其他对象) 损坏，那么 **MEDIALOG** 将不会前进。 这是因为 **MEDIALOG** 是介质恢复所需的最早扩展数据块。 如果您的工作负载在继续，那么 **CURRLOG** 仍将继续前行，因此将写入新的扩展数据块。 根据您的配置 (包括 **LogManagement** 设置) ，这可能会开始填充日志文件系统。 如果日志文件系统完全填充，那么事务将回滚，并且队列管理器可能会突然结束。 因此，当队列损坏时，您可能只有有限的时间在队列管理器结束之前执行操作。 您有多少时间，取决于工作负载导致队列管理器写入新扩展数据块的速率以及日志文件系统中的可用空间量。

如果您正在使用手动日志管理，那么可能是归档重新启动恢复不需要的扩展数据块，然后从日志文件系统中删除这些扩展数据块，即使介质恢复仍需要这些扩展数据块。 只要您可以在需要时从归档中复原这些文件，就可以接受此操作。 当队列损坏且 **MEDIALOG** 停止前进时，此策略不会导致日志文件系统填充。 但是，如果仅归档和删除重新启动或介质恢复不需要的扩展数据块，那么当队列损坏时，日志文件系统将开始填充。

如果您正在使用自动或归档日志管理，那么队列管理器将不会复用介质恢复仍需要的扩展数据块，即使您可能已归档这些扩展数据块并使用 SET LOG ARCHIVE 通知队列管理器也是如此。 因此，如果队列损坏，那么日志文件系统将开始填充。

如果队列损坏，那么将写入 OBJECT DAMAGED FFDC ，并且 **MEDIALOG** 将停止向前移动。 当您在 **runmqsc** 中显示受损对象的状态时，可以从 FFDC 标识该对象，或者因为它是具有最旧 **MEDIALOG** 的对象。

如果日志文件系统正在填充，并且您担心由于日志文件系统已满而使工作负载回退，那么恢复对象或停顿工作负载可能会停止此情况发生。

保护 IBM MQ 日志文件

当队列管理器正在运行时，请勿接触日志文件，可能无法恢复。 使用超级用户或 mqm 权限来保护日志文件免遭无意修改。

当 IBM MQ 队列管理器正在运行时，请勿手动除去活动日志文件。 如果用户无意中删除队列管理器需要重新启动的日志文件，那么 IBM MQ **不会** 发出任何错误并继续处理数据 包括持久消息。 队列管理器正常关闭，但可能无法重新启动。 因此，无法恢复消息。

有权除去活动队列管理器正在使用的日志的用户还有权删除其他重要队列管理器资源 (例如队列文件，对象目录和 IBM MQ 可执行文件)。 因此，它们可能会以 IBM MQ 无法保护自身的方式损坏正在运行或休眠的队列管理器 (可能是由于经验不足)。

授予超级用户或 mqm 权限时请谨慎操作。

使用 dmpmqlog 命令转储日志内容

如何使用 dmpmqlog 命令转储队列管理器日志的内容。

使用 dmpmqlog 命令可转储队列管理器日志的内容。 缺省情况下，将转储所有活动日志记录，即，该命令从日志的头开始转储 (通常是最后完成的检查点的开头)。

通常只能在队列管理器未运行时转储日志。 由于队列管理器在关闭期间采用检查点，因此日志的活动部分通常包含少量日志记录。 但是，您可以使用 dmpmqlog 命令通过下列其中一个选项来转储更多日志记录，以更改转储的开始位置:

- 从日志的 *base* 开始转储。 日志的基础是日志文件中包含日志头的第一个日志记录。 在这种情况下，转储的额外数据量取决于日志文件中的日志头位置。 如果它接近日志文件的开头，那么仅转储少量附加数据。 如果主机靠近日志文件的末尾，那么将转储更多数据。
- 将转储的起始位置指定为单个日志记录。 每个日志记录都由唯一的 日志序号 *(LSN)*标识。 对于循环日志记录，此起始日志记录不能在日志的基础之前; 此限制不适用于线性日志。 在运行该命令之前，您可能需要恢复不活动的日志文件。 必须指定有效的 LSN (取自先前的 dmpmqlog 输出) 作为起始位置。

  例如，通过线性日志记录，可以从最后一个 dmpmqlog 输出指定 nextlsn 。 nextlsn 显示在 Log File Header 中，并指示要写入的下一个日志记录的 LSN。 使用此位置作为开始位置来格式化自上次转储日志以来写入的所有日志记录。
- **仅适用于线性日志**，您可以指示 dmpmqlog 从任何给定日志文件扩展数据块开始格式化日志记录。 在这种情况下， dmpmqlog 期望在活动日志文件所在的目录中找到此日志文件以及每个后续日志文件。 此选项不适用于循环日志，其中 dmpmqlog 无法在日志基础之前访问日志记录。

dmpmqlog 命令的输出是 Log File Header 和一系列格式化的日志记录。 队列管理器使用多条日志记录来记录对其数据的更改。

某些已格式化的信息仅供内部使用。 以下列表包含最有用的日志记录:

日志文件标题
每个日志都有一个日志文件头，这始终是 dmpmqlog 命令格式化的第一个内容。 它包含以下字段:

logactive	主日志扩展数据块数。
loginactive	辅助日志扩展数据块的数量。
logsize	每个扩展数据块的 4 KB 页面数。
基线	日志扩展数据块中包含日志头的第一个 LSN。
nextlsn	要写入的下一个日志记录的 LSN。
标题	日志头的日志记录的 LSN。
尾	标识日志尾部位置的 LSN。
hflag1	日志是循环还是 LOG RETAIN (线性)。
HeadExtent 标识	包含日志头的日志扩展数据块。

日志记录头
日志中的每个日志记录都有一个包含以下信息的固定头:

LSN	日志序号。
LogRecd 类型	日志记录的类型。
XTranid	与此日志记录关联的事务标识 (如果有)。 MQI 的 *TranType* 指示仅 IBM MQ 事务。 其他资源管理器涉及 XA 的 *TranType* 。 同一工作单元中涉及的更新具有相同的 *XTranid*。
QueueName	与此日志记录关联的队列 (如果有)。
QID	队列的唯一内部标识。
PrevLSN	同一事务 (如果有) 中先前日志记录的 LSN。

启动队列管理器
这将记录队列管理器已启动的日志。

StartDate 队列管理器的启动日期。

StartTime 队列管理器的启动时间。

停止队列管理器
这是队列管理器已停止的日志。

StopDate 队列管理器停止的日期。

StopTime 队列管理器停止的时间。

ForceFlag 使用的关闭类型。

开始检查点
这表示队列管理器检查点的启动。

结束检查点
这表示队列管理器检查点的结束。

ChkPtLSN 启动此检查点的日志记录的 LSN。

放入消息
这将记录放入队列的持久消息。 如果将消息置于同步点下，那么日志记录头包含非空 *XTranid*。 记录的其余部分包含:

MapIndex 队列上消息的标识。 它可用于匹配用于从队列中获取此消息的相应 MQGET 。 在这种情况下，可以找到包含相同 QueueName 和 MapIndex 的后续 *Get Message* 日志记录。 此时，可以复用 MapIndex 标识以将后续消息放入该队列。

数据 包含在此日志记录的十六进制转储中的是各种内部数据，后跟消息描述符 (eyecatcher MD) 的表示，然后是消息数据本身。

放置部件
对于单个日志记录而言过大的持久消息将记录为多个 *Put Part* 日志记录，后跟单个 *Put Message* 记录。 如果有 *Put Part* 条记录，那么 *PrevLSN* 字段将 *Put Part* 条记录和最终 *Put Message* 条记录链接在一起。

数据 继续前一个日志记录处于关闭状态的消息数据。

获取消息
仅记录持久消息的获取。 如果消息在同步点下，那么日志记录头包含非空 *XTranid*。 记录的其余部分包含:

MapIndex 标识从队列中检索的消息。 包含相同 *QueueName* 和 *MapIndex* 的最新 *Put Message* 日志记录标识检索的消息。

QPriority 从队列检索的消息的优先级。

启动事务
指示新事务的开始。 TranType MQI 指示仅 IBM MQ 事务。 TranType XA 指示涉及其他资源管理器的事务类型。 此事务进行的所有更新都将具有相同的 *XTranid*。

准备事务
指示队列管理器已准备好落实与指定 *XTranid* 关联的更新。 此日志记录作为涉及其他资源管理器的两阶段落实的一部分写入。

落实事务
指示队列管理器已落实事务所做的所有更新。

回滚事务
这表示队列管理器打算回滚事务。

结束事务
这表示回滚事务结束。

事务表
此记录是在同步点期间写入的。 它记录已进行持久更新的每个事务的状态。 对于每个事务，将记录以下信息:

XTranid	事务标识。
FirstLSN	与事务关联的第一个日志记录的 LSN。
LastLSN	与事务关联的最后一个日志记录的 LSN。

事务参与者
此日志记录由队列管理器的 XA 事务管理器组件写入。 它记录参与事务的外部资源管理器。 对于每个参与者，将记录以下内容:

RMName	资源管理器的名称。
RMID	资源管理器标识。 这还会记录在后续 *Transaction Prepared* 日志记录中，这些日志记录记录了资源管理器参与的全局事务。
SwitchFile	此资源管理器的交换机装入文件。
XAOpenString	此资源管理器的 XA 打开字符串。
XACloseString	此资源管理器的 XA 关闭字符串。

事务已准备
此日志记录由队列管理器的 XA 事务管理器组件写入。 它指示已成功准备指定的全局事务。 将指示每个参与的资源管理器落实。 每个已准备好的资源管理器的 *RMID* 将记录在日志记录中。 如果队列管理器本身正在参与事务，那么将存在 *RMID* 为零的 *Participant Entry* 。

事务忘记
此日志记录由队列管理器的 XA 事务管理器组件写入。 将落实决策传递给每个参与者后，它将遵循 *Transaction Prepared* 日志记录。

清除队列
这将记录以下事实: 已清除队列上的所有消息，例如，使用 MQSC 命令 CLEAR QUEUE。

“队列”属性
这将记录队列属性的初始化或更改。

创建对象
这将记录 IBM MQ 对象的创建。

ObjName	创建的对象的名称。
UserId	执行创建的用户标识。

删除对象
这将记录 IBM MQ 对象的删除。

ObjName	已删除的对象的名称。

备份和复原 IBM MQ 队列管理器数据

通过备份队列管理器和队列管理器数据，仅备份队列管理器配置以及使用备份队列管理器，可以保护队列管理器免受硬件故障可能导致的损坏。

关于此任务

警告: 如果将队列管理器移至其他操作系统，那么必须非常小心。 有关更多信息，请参阅 将队列管理器移至其他操作系统 。

您可以定期采取措施，保护队列管理器免受硬件故障可能导致的损坏。 有三种保护队列管理器的方法:

备份队列管理器数据
如果硬件发生故障，那么可能会强制队列管理器停止。 如果由于硬件故障而丢失了任何队列管理器日志数据，那么队列管理器可能无法重新启动。 如果备份队列管理器数据，那么您可能能够恢复部分或全部丢失的队列管理器数据。

通常，备份队列管理器数据的频率越高，在发生导致丢失恢复日志完整性的硬件故障时丢失的数据就越少。

要备份队列管理器数据，队列管理器不得正在运行。

仅备份队列管理器配置
如果硬件发生故障，那么可能会强制队列管理器停止。 如果队列管理器配置和日志数据都由于硬件故障而丢失，那么队列管理器无法重新启动或无法从日志中恢复。 如果备份队列管理器配置，那么可以根据保存的定义重新创建队列管理器及其所有对象。

要备份队列管理器配置，该队列管理器必须正在运行。

使用备份队列管理器
如果硬件故障严重，那么队列管理器可能不可恢复。 在此情况下，如果不可恢复队列管理器具有专用备份队列管理器，那么可以激活备份队列管理器来代替不可恢复队列管理器。 如果定期更新，那么备份队列管理器日志可以包含日志数据，其中包括不可恢复队列管理器中的最后一个完整日志。

当现有队列管理器仍在运行时，可以更新备份队列管理器。

过程

- 要备份和复原队列管理器数据，请参阅:
  - 第 551 页的『备份队列管理器数据』.
  - 第 552 页的『复原队列管理器数据』.
- 要备份和复原队列管理器配置，请参阅:
  - 第 553 页的『备份队列管理器配置』
  - 第 553 页的『复原队列管理器配置』
- 要创建，更新和启动备份队列管理器，请参阅 第 554 页的『使用备份队列管理器』 。

备份队列管理器数据

备份队列管理器数据可帮助您防止由硬件错误导致的数据可能丢失。

开始之前

在开始备份队列管理器之前，请确保队列管理器未在运行。 如果尝试备份正在运行的队列管理器，那么备份可能不一致，因为复制文件时正在进行更新。 如果可能，请通过运行 **endmqm -w** 命令 (等待关闭) 来停止队列管理器，仅当此操作失败时，才使用 **endmqm -i** 命令 (立即关闭)。

关于此任务

要获取队列管理器数据的备份副本，请完成以下任务:

过程

1. 通过使用配置文件中的信息，搜索队列管理器将其数据和日志文件放置在其中的目录。
   有关更多信息，请参阅第 71 页的『在 Multiplatforms 版上的 .ini 文件中更改 IBM MQ 配置信息』 。

注: 将变换目录中显示的名称，以确保它们与您使用 IBM MQ 的平台兼容。 有关名称转换的更多信息，请参阅 了解 IBM MQ 文件名。

2. 复制所有队列管理器的数据和日志文件目录，包括所有子目录。

   确保不遗漏任何文件，特别是日志控制文件 (如 第 526 页的『日志的外观』中所述) 和配置文件 (如 第 193 页的『初始化和配置文件』中所述)。 某些目录可能为空，但您需要所有这些目录在以后复原备份。

   对于循环日志记录，请同时备份队列管理器数据和日志文件目录，以便可以复原一组一致的队列管理器数据和日志。

   对于线性日志记录，请同时备份队列管理器数据和日志文件目录。 如果有相应的完整日志文件序列可用，那么只能复原队列管理器数据文件。

3. 保留文件的所有权。

   Linux AIX 对于 IBM MQ for UNIX 和 Linux 系统，您可以使用 **tar** 命令执行此操作。 (如果队列大于 2 GB ，那么不能使用 **tar** 命令。 有关更多信息，请参阅 启用大型队列。

   注: 升级到 IBM WebSphere MQ 7.5 和更高版本时，请确保备份 `qm.ini` 文件和注册表条目。 队列管理器信息存储在 `qm.ini` 文件中，可用于还原到先前版本的 IBM MQ。

相关任务

停止队列管理器

第 13 页的『创建队列管理器后备份配置文件』

IBM MQ 配置信息存储在 AIX, Linux, and Windows 上的配置文件中。 创建队列管理器后，备份配置文件。 然后，如果您创建另一个导致问题的队列管理器，那么可以在除去问题源后恢复备份。

复原队列管理器数据

执行以下步骤以复原队列管理器数据的备份。

开始之前

在启动备份之前，请确保队列管理器未在运行。

在集群中复原队列管理器的备份时，请参阅 第 301 页的『恢复集群队列管理器』 和 集群: 可用性，多实例和灾难恢复 以获取更多信息。

注: 升级到更高版本的 IBM MQ 时，请确保备份 **.ini** 文件和注册表条目。 队列管理器信息存储在 **.ini** 文件中，可用于还原到先前版本的 IBM MQ。

过程

1. 通过使用配置文件中的信息，查找队列管理器将其数据和日志文件放置在其中的目录。
2. 清空要将备份数据放入的目录。
3. 将备份的队列管理器数据和日志文件复制到正确的位置。

   确保您有日志控制文件以及日志文件。

   对于循环日志记录，请同时备份队列管理器数据和日志文件目录，以便可以复原一组一致的队列管理器数据和日志。

   对于线性日志记录，请同时备份队列管理器数据和日志文件目录。 如果有相应的完整日志文件序列可用，那么只能复原队列管理器数据文件。

4. 更新配置信息文件。

   检查 IBM MQ 和队列管理器配置文件是否一致，以便 IBM MQ 可以在正确的位置查找复原的数据。

5. 请检查生成的目录结构，以确保您具有所有必需的目录。

   有关 IBM MQ 目录和子目录的更多信息，请参阅 Windows 系统上的目录结构 和 AIX and Linux 系统上的目录内容。

结果

如果正确备份和复原了数据，那么队列管理器现在将启动。

Multi 备份队列管理器配置

如果队列管理器配置和日志数据由于硬件故障而丢失，并且队列管理器无法重新启动或无法从日志中恢复，那么备份队列管理器配置可帮助您根据其定义重建队列管理器。

关于此任务

ALW 在 AIX, Linux, and Windows 上，可以使用 **dmpmqcfg** 命令来转储 IBM MQ 队列管理器的配置。

IBM i 在 IBM i 上，可以使用 "转储 MQ 配置 (**DMPMQMCFG**)" 命令来转储队列管理器的配置对象和权限。

过程

1. 确保队列管理器正在运行。
2. 根据您的平台，使用下列其中一个命令来备份队列管理器配置:

- ALW On AIX, Linux, and Windows: Execute the Dump MQ Configuration command, **dmpmqcfg**, using the default formatting option of (-f mqsc) MQSC and all attributes (-a), use standard output redirection to store the definitions into a file. 例如:

```
dmpmqcfg -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
```

- IBM i 在 IBM i 上: 使用缺省格式化选项 OUTPUT (*MQSC) 和 EXPATTR (*ALL) 执行转储 MQ 配置命令 (**DMPMQMCFG**)，使用 TOFILE 和 TOMBR 将定义存储到物理文件成员中。 例如:

```
DMPMQMCFG MQMNAME(MYQMGR) OUTPUT(*MQSC) EXPATTR(*ALL) TOFILE(QMQMSAMP/QMQSC)
TOMBR(MYQMGRDEF)
```

相关任务

第 553 页的『复原队列管理器配置』
通过首先确保队列管理器正在运行，然后针对您的平台运行相应的命令，可以从备份复原队列管理器的配置。

相关参考

dmpmqcfg (转储队列管理器配置)
转储 MQ 配置 (DMPMQMCFG)

Multi 复原队列管理器配置

通过首先确保队列管理器正在运行，然后针对您的平台运行相应的命令，可以从备份复原队列管理器的配置。

关于此任务

ALW 在 AIX, Linux, and Windows 上，可以使用 **runmqsc** 命令来复原 IBM MQ 队列管理器的配置。

IBM i 在 IBM i 上，可以使用 **STRMQMMQSC** 命令来复原队列管理器的配置对象和权限。

过程

1. 确保队列管理器正在运行。

请注意，如果数据和日志损坏无法通过其他方法恢复，那么可能已重新创建队列管理器。

2. 根据您的平台，使用下列其中一个命令来复原队列管理器配置:

   - ALW 在 AIX, Linux, and Windows 上，针对队列管理器运行 **runmqsc** ，使用标准输入重定向从转储 MQ 配置 (**dmpmqcfg**) 命令生成的脚本文件复原定义 (请参阅 第 553 页的『备份队列管理器配置』)。 例如:

     ```
     runmqsc MYQMGR < /mq/backups/MYQMGR.mqsc
     ```

   - IBM i 在 IBM i 上: 对队列管理器运行 **STRMQMMQSC** ，并使用 **SRCMBR** 和 **SRCFILE** 参数从 "转储 MQ 配置" (**DMPMQMCFG**) 命令生成的物理文件成员复原定义 (请参阅 第 553 页的『备份队列管理器配置』)。 例如:

     ```
     STRMQMMQSC MQMNAME(MYQMGR) SRCFILE(QMQMSAMP/QMQSC) SRCMBR(MYQMGR)
     ```

相关任务

第 553 页的『备份队列管理器配置』
如果队列管理器配置和日志数据由于硬件故障而丢失，并且队列管理器无法重新启动或无法从日志中恢复，那么备份队列管理器配置可帮助您根据其定义重建队列管理器。

相关参考

dmpmqcfg (转储队列管理器配置)
runmqsc (运行 MQSC 命令)
转储 MQ 配置 (DMPMQMCFG)
启动 IBM MQ 命令 (STRMQMMQSC)

使用备份队列管理器

现有队列管理器可以具有专用备份队列管理器以用于灾难恢复。

关于此任务

备份队列管理器是现有队列管理器的不活动副本。 如果现有队列管理器由于严重硬件故障而变为不可恢复，那么可以使备份队列管理器联机以替换不可恢复的队列管理器。

必须定期将现有队列管理器日志文件复制到备份队列管理器，以确保备份队列管理器仍然是有效的灾难恢复方法。 对于要复制的日志文件，不需要停止现有队列管理器，但是仅当队列管理器已完成对其的写入时，才应该复制日志文件; 请参阅 第 555 页的『更新备份队列管理器』 以获取有关如何确保不再写入特定日志文件的信息，以便可以安全地复制该日志文件。

注: 由于将持续更新现有队列管理器日志，因此现有队列管理器日志与复制到备份队列管理器日志的日志数据之间始终存在细微差异。 对备份队列管理器的定期更新可最大限度减少两个日志之间的差异。

如果需要使备份队列管理器联机，那么必须将其激活，然后启动。 在启动备份队列管理器之前激活备份队列管理器是一项预防措施，可防止意外启动备份队列管理器。 激活备份队列管理器后，无法再对其进行更新。

要点: 一旦旧备份队列管理器成为新的活动队列管理器，无论出于何种原因，都不再有备份队列管理器。 这实际上是异步复制的一种形式，因此新的活动队列管理器应该在逻辑上落后于旧的活动队列管理器。 因此，旧的活动队列管理器不再充当新的活动队列管理器的备份。

过程

- 有关使用备份队列管理器的信息，请参阅以下主题:
  - 第 555 页的『创建备份队列管理器』
  - 第 555 页的『更新备份队列管理器』
  - 第 556 页的『启动备份队列管理器』

相关概念

第 526 页的『日志记录: 确保消息未丢失』

IBM MQ 在恢复日志中记录对队列管理器控制的持久数据的所有重要更改。

创建备份队列管理器

创建备份队列管理器作为现有队列管理器的不活动副本。

关于此任务

要点: 使用线性日志记录时，只能使用备份队列管理器。

备份队列管理器需要以下内容:

- 具有与现有队列管理器相同的属性，例如队列管理器名称，日志记录类型和日志文件大小。
- 与现有队列管理器位于同一平台上。
- 要处于等于或高于现有队列管理器的代码级别。

过程

1. 使用控制命令 **crtmqm** 为现有队列管理器创建备份队列管理器。
2. 复制所有现有队列管理器的数据和日志文件目录 (包括所有子目录)，如 第 551 页的『备份队列管理器数据』中所述。
3. 使用从现有队列管理器获取的副本覆盖备份队列管理器的数据和日志文件目录 (包括所有子目录)。
4. 在备份队列管理器上运行 **strmqm** 控制命令，如以下示例中所示:

   ```
   strmqm -r BackupQMName
   ```

   此命令将队列管理器标记为 IBM MQ 中的备份队列管理器，并重放所有复制的日志扩展数据块，以使备份队列管理器与现有队列管理器保持一致。

相关参考

crtmqm (创建队列管理器)

strmqm (启动队列管理器)

更新备份队列管理器

要确保备份队列管理器仍然是有效的灾难恢复方法，必须定期更新该备份队列管理器。

关于此任务

定期更新会减少备份队列管理器日志与当前队列管理器日志之间的差异。 在备份队列管理器之前，无需停止该队列管理器。

警告: 如果将一组不连续的日志复制到备份队列管理器日志目录，那么将仅重放直到找到第一个缺失日志的位置的日志。

过程

1. 在要备份的队列管理器上发出以下 Script (MQSC) 命令:

   ```
   RESET QMGR TYPE(ADVANCELOG)
   ```

   这将停止对当前日志的任何写入，然后将队列管理器日志记录推进到下一个日志扩展数据块。 这将确保您备份记录到当前时间的所有信息。
2. 通过在要备份的队列管理器上发出以下 Script (MQSC) 命令，获取 (新的) 当前活动日志扩展数据块号:

   ```
   DIS QMSTATUS CURRLOG
   ```

3. 将更新后的日志扩展数据块文件从当前队列管理器日志目录复制到备份队列管理器日志目录。

   复制自上次更新以来的所有日志扩展数据块，最多 (但不包括) 第 555 页的『2』中记录的当前扩展数据块。 仅复制以 "S ..." 开头的日志扩展数据块文件。
4. 在备份队列管理器上运行 **strmqm** 控制命令，如以下示例中所示:

```
strmqm -r BackupQMName
```

这将重放所有复制的日志扩展数据块，并使备份队列管理器与队列管理器一起执行步骤。 重放完成时，您将收到一条消息，该消息标识重新启动恢复所需的所有日志扩展数据块以及介质恢复所需的所有日志扩展数据块。

相关参考

重置队列管理器

显示 QM 状态

strmqm (启动队列管理器)

启动备份队列管理器

您可以将备份队列管理器替换为不可恢复的队列管理器。

关于此任务

在集群中复原队列管理器的备份时，请参阅 第 301 页的『恢复集群队列管理器』 和 集群: 可用性，多实例和灾难恢复 以获取更多信息。

如果不可恢复队列管理器具有专用备份队列管理器，那么可以激活备份队列管理器以代替不可恢复队列管理器。

将不可恢复队列管理器替换为备份队列管理器时，可能会丢失不可恢复队列管理器中的某些队列管理器数据。 丢失的数据量取决于最近一次更新备份队列管理器的时间。 最近一次更新时，队列管理器数据丢失越少。

注: 即使队列管理器数据和日志文件保存在不同的目录中，也请确保同时备份和复原这些目录。 如果队列管理器数据和日志文件的年龄不同，那么队列管理器未处于有效状态，并且可能不会启动。 即使它确实启动，您的数据也可能损坏。

过程

1. 运行 **strmqm** 控制命令以激活备份队列管理器，如以下示例中所示:

   ```
   strmqm -a BackupQMName
   ```

   已激活备份队列管理器。 既然它处于活动状态，那么无法再更新备份队列管理器。
2. 运行 **strmqm** 控制命令以启动备份队列管理器，如以下示例中所示:

   ```
   strmqm BackupQMName
   ```

   IBM MQ 将此视为重新启动恢复，并使用备份队列管理器中的日志。 在备份队列管理器的上次更新期间，将发生重放，因此仅回滚上次记录的检查点中的活动事务。
3. 重新启动所有通道。
4. 请检查生成的目录结构，以确保您具有所有必需的目录。

   有关 IBM MQ 目录和子目录的更多信息，请参阅 规划文件系统支持。
5. 确保您有日志控制文件以及日志文件。 还要检查 IBM MQ 和队列管理器配置文件是否一致，以便 IBM MQ 可以在正确的位置查找复原的数据。

结果

如果数据已正确备份和复原，那么队列管理器现在将启动。

相关任务

第 188 页的『重新启动已停止的通道』
当通道进入 STOPPED 状态时，您必须手动重新启动通道。

相关参考

strmqm (启动队列管理器)

对集群错误恢复的更改 (在除 z/OS 以外的服务器上)

从 IBM WebSphere MQ 7.1 开始，队列管理器将重新运行导致问题的操作，直到问题得到解决为止。 如果在五天后问题仍未解决，那么队列管理器将关闭以防止高速缓存变得过时。

在 IBM WebSphere MQ 7.1 之前，如果队列管理器检测到管理集群的本地存储库管理器存在问题，那么它会更新错误日志。 在某些情况下，它随后停止管理集群。 队列管理器继续与集群交换应用程序消息，这依赖于其日益过时的集群定义高速缓存。 从 IBM WebSphere MQ 7.1 开始，队列管理器将重新运行导致问题的操作，直到问题得到解决为止。 如果在五天后问题仍未解决，那么队列管理器将关闭以防止高速缓存变得过时。 随着高速缓存变得越来越过时，它会导致更多问题。 7.1 或更高版本中有关集群错误的已更改行为不适用于 z/OS。

集群管理的每个方面都由本地存储库管理器进程 `amqrrmfa` 针对队列管理器进行处理。 该进程在所有队列管理器上运行，即使没有集群定义也是如此。

在 IBM WebSphere MQ 7.1 之前，如果队列管理器在本地存储库管理器中检测到问题，那么会在短时间间隔后停止存储库管理器。 队列管理器保持运行，处理应用程序消息和打开队列的请求，以及发布或预订主题。

在存储库管理器停止后，可供队列管理器使用的集群定义的高速缓存变得更加过时。 随着时间的推移，消息被路由到错误的目标，应用程序失败。 应用程序尝试打开尚未传播到本地队列管理器的集群队列或发布主题失败。

除非管理员在错误日志中检查了存储库消息，否则管理员可能无法实现集群配置存在问题。 如果在更长时间内无法识别故障，并且队列管理器未更新其集群成员资格，那么将发生更多问题。 不稳定会影响集群中的所有队列管理器，并且集群会出现不稳定。

从 IBM WebSphere MQ 7.1 开始， IBM MQ 采用不同的方法来处理集群错误。 存储库管理器不会停止存储库管理器并继续运行，而是重新运行失败的操作。 如果队列管理器检测到存储库管理器存在问题，那么它遵循两个操作过程之一。

1. 如果错误不会损害队列管理器的操作，那么队列管理器会将消息写入错误日志。 它每 10 分钟重新运行一次失败的操作，直到操作成功为止。 缺省情况下，您有 5 天时间来处理错误; 如果失败，那么队列管理器会将消息写入错误日志并关闭。 您可以推迟 5 天关闭。
2. 如果错误影响了队列管理器的操作，那么队列管理器会将消息写入错误日志，并立即关闭。

影响队列管理器操作的错误是队列管理器无法诊断的错误，或者是可能产生不可预见后果的错误。 此类型的错误通常会导致队列管理器写入 FFST 文件。 影响队列管理器操作的错误可能是由 IBM MQ 中的错误导致的，也可能是由管理员或程序导致的，执行了一些意外操作，例如结束 IBM MQ 进程。

错误恢复行为中的更改点是限制队列管理器继续运行的时间，并且有越来越多的不一致的集群定义。 随着集群定义中不一致的数量增加，异常应用程序行为的可能性也随之增加。

缺省选择是在五天后关闭队列管理器，这是在限制不一致的次数与在检测到问题并解决问题之前保持队列管理器可用之间的一种妥协。

您可以延长队列管理器无限期关闭之前的时间，同时修复问题或等待计划的队列管理器关闭。 五天的停留使队列管理器持续运行一个漫长的周末，让您有时间对任何问题作出反应或延长重新启动队列管理器之前的时间。

纠正行动

您可以选择操作来处理集群错误恢复问题。 第一个选择是监视和修正问题，第二个选择是监视和推迟修正问题，最后一个选择是继续像 IBM WebSphere MQ 7.1 之前的发行版一样管理集群错误恢复。

1. 监视队列管理器错误日志以获取错误消息 AMQ9448 和 AMQ5008，并解决问题。

   AMQ9448 指示存储库管理器在运行命令后返回了错误。 此错误会标记是否开始每 10 分钟重试一次命令，并最终在 5 天后停止队列管理器，除非您推迟关闭。

   AMQ5008 指示队列管理器已停止，因为缺少 IBM MQ 进程。 AMQ5008 来自存储库管理器的结果将在五天后停止。 如果存储库管理器停止，那么队列管理器将停止。
2. 监视队列管理器错误日志以获取错误消息 AMQ9448，并推迟解决问题。

如果禁用从 SYSTEM.CLUSTER.COMMAND.QUEUE 获取消息，那么存储库管理器将停止尝试运行命令，并且无限期地继续操作而不处理任何工作。 但是，将释放存储库管理器保留给队列的任何句柄。 由于存储库管理器未停止，因此队列管理器不会在五天后停止。

运行 MQSC 命令以禁止从 SYSTEM.CLUSTER.COMMAND.QUEUE 获取消息:

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(DISABLED)
```

要恢复从 SYSTEM.CLUSTER.COMMAND.QUEUE 接收消息，请运行 MQSC 命令:

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
```

3. 将队列管理器还原为与 IBM WebSphere MQ 7.1 之前相同的集群错误恢复行为。

   您可以设置队列管理器调整参数，以在存储库管理器停止时保持队列管理器运行。

   在 qm.ini 文件的 TuningParameters 节中，调整参数为 TolerateRepositoryFailure。 要防止队列管理器停止，如果存储库管理器停止，请将 TolerateRepositoryFailure 设置为 TRUE; 请参阅 第 558 页的图 89。

   重新启动队列管理器以启用 TolerateRepositoryFailure 选项。

   如果发生集群错误，导致存储库管理器无法成功启动，因此队列管理器无法启动，请将 TolerateRepositoryFailure 设置为 TRUE 以在没有存储库管理器的情况下启动队列管理器。

特别考虑

在 IBM WebSphere MQ 7.1 之前，一些管理不属于集群的队列管理器的管理员停止了 amqrrmfa 进程。 停止 amqrrmfa 不会影响队列管理器。

在 IBM WebSphere MQ 7.1 或更高版本中停止 amqrrmfa 会导致队列管理器停止，因为它被视为队列管理器故障。 除非设置队列管理器调整参数 TolerateRepositoryFailure，否则不得在 7.1 或更高版本中停止 amqrrmfa 进程。

示例

```
TuningParameters:
        TolerateRepositoryFailure=TRUE
```

图 *89:* 在 *qm.ini* 中将 *TolerateRepositoryFailure* 设置为 *TRUE*

相关参考

队列管理器配置文件， qm.ini

配置 JMS 资源

JMS 应用程序创建和配置连接到 IBM MQ 以及访问发送或接收消息的目标所需的资源的其中一种方法是使用 Java 命名和目录接口 (JNDI) 从命名和目录服务 (称为 JNDI 名称空间) 中的某个位置检索受管对象。 必须先创建并配置受管对象，然后 JMS 应用程序才能从 JNDI 名称空间中检索受管对象。

关于此任务

您可以使用以下任一工具在 IBM MQ 中创建和配置受管对象:

IBM MQ Explorer

您可以使用 IBM MQ Explorer 来创建和管理存储在 LDAP ， 本地文件系统或其他位置中的 JMS 对象定义。

IBM MQ JMS 管理工具

IBM MQ JMS 管理工具是可用于创建和配置存储在 LDAP ， 本地文件系统或其他位置中的 IBM MQ JMS 对象的命令行工具。 JMS 管理工具使用类似于 **runmqsc** 的语法，并且还支持脚本编制。

管理工具使用配置文件来设置某些属性的值。 提供了样本配置文件，您可以在开始使用该工具配置 JMS 资源之前编辑该文件以适合您的系统。 有关配置文件的更多信息，请参阅第 564 页的『配置 JMS 管理工具』。

部署到 WebSphere Application Server 的 IBM MQ JMS 应用程序需要从应用程序服务器 JNDI 存储库访问 JMS 对象。 因此，如果在 WebSphere Application Server 和 IBM MQ 之间使用 JMS 消息传递，那么必须在 WebSphere Application Server 中创建与您在 IBM MQ 中创建的对象相对应的对象。

IBM MQ Explorer 和 IBM MQ JMS 管理工具不能用于管理存储在 WebSphere Application Server 中的 IBM MQ JMS 对象。 相反，您可以使用以下任一工具在 WebSphere Application Server 中创建和配置受管对象:

WebSphere Application Server 管理控制台
WebSphere Application Server 管理控制台是一个基于 Web 的工具，可用于管理 WebSphere Application Server 中的 IBM MQ JMS 对象。

WebSphere Application Server wsadmin 脚本编制客户机
WebSphere Application Server wsadmin 脚本编制客户机提供专门的命令来管理 WebSphere Application Server 中的 IBM MQ JMS 对象。

如果要使用 JMS 应用程序从 WebSphere Application Server 中访问 IBM MQ 队列管理器的资源，那么必须使用 WebSphere Application Server 中包含 IBM MQ classes for JMS 版本的 IBM MQ 消息传递提供程序。 WebSphere Application Server 随附的 IBM MQ 资源适配器由使用 IBM MQ 消息传递提供程序执行 JMS 消息传递的所有应用程序使用。 在应用 WebSphere Application Server 修订包时，通常会自动更新 IBM MQ 资源适配器，但是如果先前已手动更新资源适配器，那么必须手动更新配置以确保正确应用维护。

相关任务
编写 IBM MQ classes for JMS 应用程序
相关参考
runmqsc

在 JNDI 名称空间中配置连接工厂和目标

JMS 应用程序通过 Java 命名和目录接口 (JNDI) 访问命名和目录服务中的受管对象。 JMS 受管对象存储在命名和目录服务中称为 JNDI 名称空间的位置。 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目标。

关于此任务

JMS 应用程序使用上下文在命名和目录服务中查找 JMS 对象的名称:

初始上下文 (initial context)
初始上下文定义 JNDI 名称空间的根。 对于命名和目录服务中的每个位置，您需要指定初始上下文，以提供一个起始点， JMS 应用程序可从该起始点解析命名和目录服务的该位置中受管对象的名称。

子上下文
一个上下文可以具有一个或多个子上下文。 子上下文是 JNDI 名称空间的子部分，可以包含受管对象，例如连接工厂和目标以及其他子上下文。 子上下文本身并不是对象；它仅仅是子上下文中对象的命名约定的扩展。

您可以使用 IBM MQ Explorer 或 IBM MQ JMS 管理工具来创建上下文。

在 IBM MQ classes for JMS 应用程序可以从 JNDI 名称空间检索受管对象之前，必须首先使用 IBM MQ Explorer 或 IBM MQ JMS 管理工具创建受管对象。 您可以创建和配置以下类型的 JMS 对象:

连接工厂
JMS 连接工厂对象定义了一组用于连接的标准配置属性。 JMS 应用程序使用连接工厂来创建与 IBM MQ 的连接。 您可以创建特定于两个消息传递域之一 (点到点消息传递域和发布/预订消息传递域) 的连接工厂。 或者，从 JMS 1.1 开始，您可以创建独立于域的连接工厂，这些连接工厂可用于点到点消息传递和发布/预订消息传递。

Destination
JMS 目标是表示客户机生成的消息的目标以及 JMS 应用程序使用的消息源的对象。 JMS 应用程序可以使用单个目标对象来放置消息并从中获取消息，或者应用程序可以使用单独的目标对象。 有两种类型的目标对象:

- 点到点消息传递中使用的 JMS 队列目标
- 发布/预订消息传递中使用的 JMS 主题目标

下图显示了在 IBM MQ JNDI 名称空间中创建的 JMS 对象的示例。

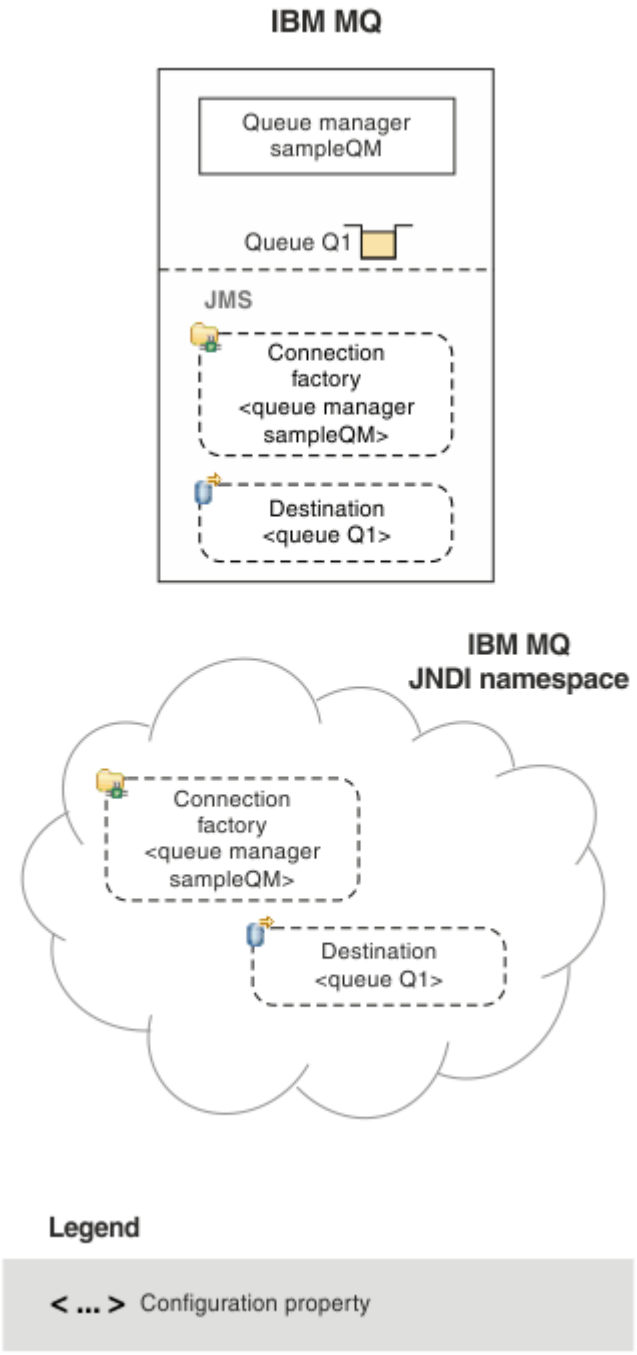


图 *90:* 在 *IBM MQ* 中创建了 *JMS* 个对象

如果在 WebSphere Application Server 与 IBM MQ 之间使用 JMS 消息传递，那么必须在 WebSphere Application Server 中创建相应的对象以用于与 IBM MQ 进行通信。 在 WebSphere Application Server 中创建其中一个对象时，该对象将存储在 WebSphere Application Server JNDI 名称空间中，如下图所示。

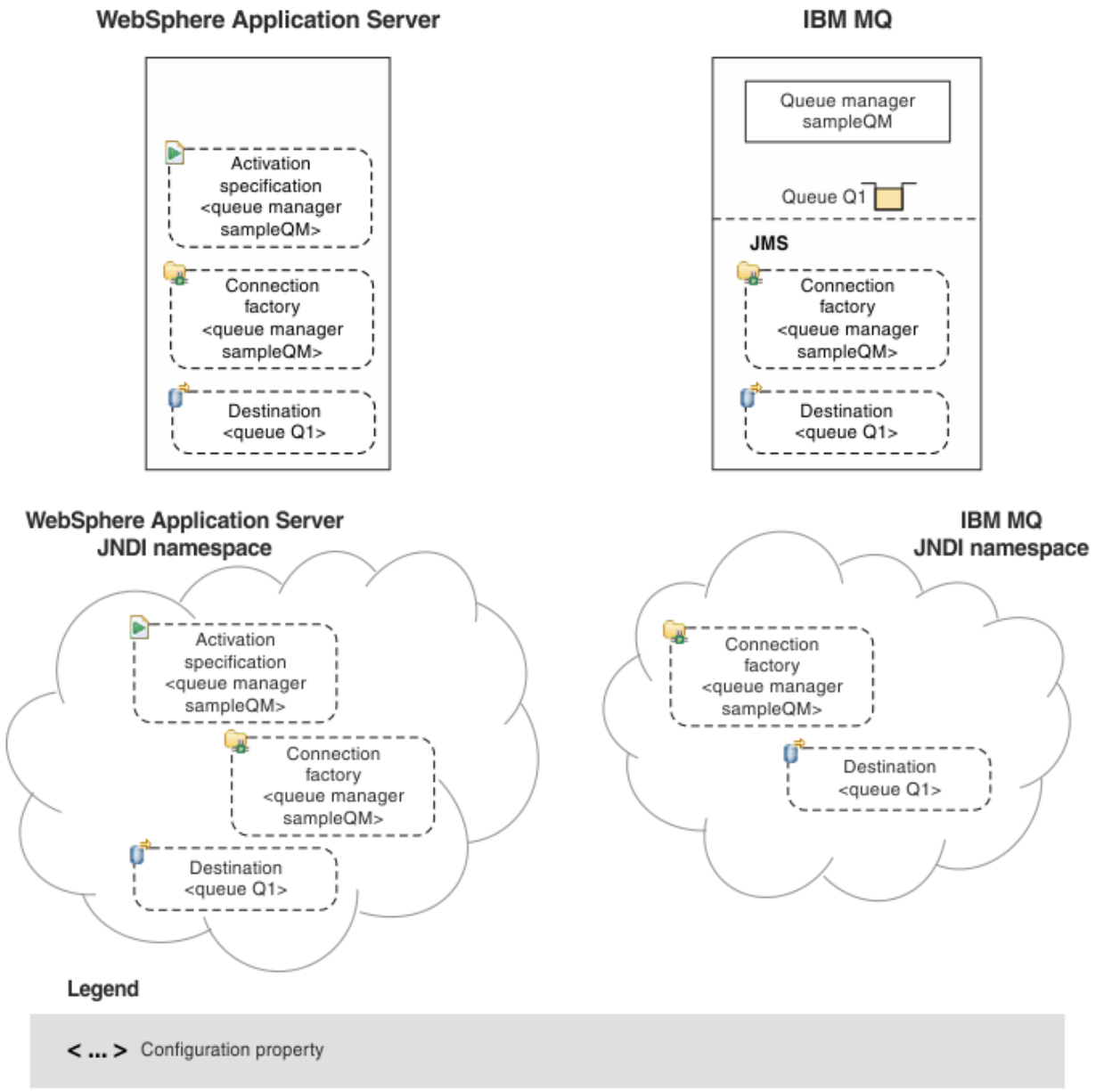


图 *91:* 在 *WebSphere Application Server* 中创建的对象以及 *IBM MQ* 中的相应对象

如果应用程序使用消息驱动的 bean (MDB)，那么连接工厂仅用于出站消息，而入站消息由激活规范接收。激活规范是 Java EE Connector Architecture 1.5 (JCA 1.5) 标准的一部分。 JCA 1.5 提供了将 JMS 提供程序 (例如 IBM MQ) 与 Java EE 应用程序服务器 (例如 WebSphere Application Server) 集成的标准方法。 JMS 激活规范可以与一个或多个消息驱动的 bean (MDB) 相关联，并提供这些 MDB 侦听到达目标的消息所必需的配置。

您可以使用 WebSphere Application Server 管理控制台或 wsadmin 脚本编制命令来创建和配置所需的 JMS 资源。

过程

- 要使用 IBM MQ Explorer 为 IBM MQ 配置 JMS 对象，请参阅 第 562 页的『使用 IBM MQ Explorer 配置 JMS 对象』。
- 要使用 IBM MQ JMS 管理工具为 IBM MQ 配置 JMS 对象，请参阅 第 563 页的『使用管理工具配置 JMS 对象』。

- 要为 WebSphere Application Server 配置 JMS 对象，请参阅 第 570 页的『在 WebSphere Application Server 中配置 JMS 资源』。

结果

IBM MQ classes for JMS 应用程序可以从 JNDI 名称空间检索受管对象，如果需要，可以使用 IBM JMS 扩展或 IBM MQ JMS 扩展来设置或更改其一个或多个属性。

相关任务

使用 JNDI 检索 JMS 应用程序中的受管对象

在 IBM MQ classes for JMS 应用程序中创建并配置连接工厂和目标

使用 IBM MQ Explorer 配置 JMS 对象

使用 IBM MQ Explorer 图形用户界面可从 IBM MQ 对象创建 JMS 对象，从 JMS 对象创建 IBM MQ 对象，以及用于管理和监视其他 IBM MQ 对象。

关于此任务

IBM MQ Explorer 是图形用户界面，您可在其中管理和监视 IBM MQ 对象，而不论这些对象是由本地计算机托管还是在远程系统上托管。 IBM MQ Explorer 可在 Windows 和 Linux x86-64 上运行。 它可以远程连接到在包括 z/OS 在内的任何受支持平台上运行的队列管理器，从而使您能够从控制台查看，探索和更改整个消息传递主干。

在 IBM MQ Explorer 中，所有连接工厂都存储在相应上下文和子上下文中的 "连接工厂" 文件夹中。

您可以使用 IBM MQ Explorer 执行以下类型的任务，可以从 IBM MQ Explorer 中的现有对象上下文执行，也可以从 "创建新对象" 向导中执行:

- 从以下任何 IBM MQ 对象创建 JMS 连接工厂:
  - IBM MQ 队列管理器 (无论是在本地计算机上还是在远程系统上)。
  - IBM MQ 通道。
  - IBM MQ 侦听器。
- 使用 JMS 连接工厂将 IBM MQ 队列管理器添加到 IBM MQ Explorer 。
- 从 IBM MQ 队列创建 JMS 队列。
- 从 JMS 队列创建 IBM MQ 队列。
- 从 IBM MQ 主题创建 JMS 主题，该主题可以是 IBM MQ 对象或动态主题。
- 从 JMS 主题创建 IBM MQ 主题。

过程

- 如果 IBM MQ Explorer 尚未运行，请将其启动。

  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。
- 如果尚未执行此操作，请创建初始上下文，以定义将 JMS 对象存储在命名和目录服务中的 JNDI 名称空间的根目录。

  当您已将初始上下文添加至 IBM MQ Explorer 后，就可以在 JNDI 名称空间中创建连接工厂对象、目标对象和子上下文。

  初始上下文显示在 " Navigator " 视图的 JMS 受管对象文件夹中。 请注意，虽然会显示 JNDI 名称空间的完整内容，但在 IBM MQ Explorer 中，您只能编辑存储在其中的 IBM MQ classes for JMS 对象。 有关更多信息，请参阅添加初始上下文。
- 创建并配置所需的子上下文和 JMS 受管对象。

  有关更多信息，请参阅 创建和配置 JMS 受管对象。
- 配置 IBM MQ。

  有关更多信息，请参阅 使用 IBM MQ Explorer 配置 IBM MQ 。

相关概念

IBM MQ Explorer 简介

相关任务

在 IBM MQ classes for JMS 应用程序中创建并配置连接工厂和目标

使用管理工具配置 JMS 对象

您可以使用 IBM MQ JMS 管理工具来定义八种类型的 IBM MQ classes for JMS 对象的属性，并将其存储在 JNDI 名称空间中。 然后，应用程序可以使用 JNDI 从名称空间检索这些受管对象。

关于此任务

下表显示了您可以使用动词创建，配置和处理的八种类型的受管对象。 "关键字" 列显示可在 第 563 页的表 37 中显示的命令中替代 *TYPE* 的字符串。

表 *37:* 管理工具处理的 *JMS* 对象类型

对象类型	关键字	描述
MQConnectionFactory	CF	JMS ConnectionFactory 接口的 IBM MQ 实现。 这表示用于在点到点域和发布/预订域中创建连接的工厂对象。
MQQueueConnectionFactory	QCF	JMS QueueConnectionFactory 接口的 IBM MQ 实现。 这表示用于在点到点域中创建连接的工厂对象。
MQTopicConnectionFactory	TCF	JMS TopicConnectionFactory 接口的 IBM MQ 实现。 这表示用于在发布/预订域中创建连接的工厂对象。
MQQueue	Q	JMS 队列接口的 IBM MQ 实现。 这表示点到点域中消息的目标。
MQTopic	T	JMS 主题接口的 IBM MQ 实现。 这表示发布/预订域中消息的目标。
MQXAConnectionFactory 第 563 页的『1』	XACF	JMS XAConnectionFactory 接口的 IBM MQ 实现。 这表示用于在点到点域和发布/预订域中创建连接的工厂对象，以及这些连接使用 XA 版本的 JMS 类的工厂对象。
MQXAQueueConnection 工厂 第 563 页的『1』	XAQCF	JMS XAQueueConnectionFactory 接口的 IBM MQ 实现。 这表示用于在使用 XA 版本的 JMS 类的点到点域中创建连接的工厂对象。
MQXATopicConnection 工厂 第 563 页的『1』	XATCF	JMS XATopicConnectionFactory 接口的 IBM MQ 实现。 这表示用于在使用 XA 版本的 JMS 类的发布/预订域中创建连接的工厂对象。

注:

1. 提供这些类供应用程序服务器的供应商使用。 它们不太可能对应用程序员直接有用。

有关如何配置这些对象的更多信息，请参阅 第 569 页的『配置 JMS 对象』。

您需要使用此工具的属性类型和值在 IBM MQ classes for JMS 对象的属性中列出。

您还可以使用此工具来处理 JNDI 中的目录名称空间子上下文，如 第 567 页的『配置子上下文』中所述。

您还可以使用 IBM MQ Explorer 创建和配置 JMS 受管对象。

相关任务

在 IBM MQ classes for JMS 应用程序中创建并配置连接工厂和目标

使用 JNDI 检索 JMS 应用程序中的受管对象

配置 JMS 管理工具

IBM MQ JMS 管理工具使用配置文件来设置某些属性的值。 提供了样本配置文件，您可以对其进行编辑以适合您的系统。

关于此任务

配置文件是由一组由等号 (=) 分隔的键/值对组成的纯文本文件。 通过设置配置文件中定义的三个属性的值来配置管理工具。 以下示例显示了这三个属性:

```
#Set the service provider
INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
#Set the initial context
PROVIDER_URL=ldap://polaris/o=ibm_us,c=us
#Set the authentication type
SECURITY_AUTHENTICATION=none
```

(在此示例中，行的第一列中的井号 (#) 指示注释或未使用的行。)

IBM MQ 随附了用作缺省配置文件的样本配置文件。 样本文件称为 `JMSAdmin.config`，可在 *MQ_JAVA_INSTALL_PATH*/bin 目录中找到。 您可以编辑此样本文件以定义系统所需的设置，或者创建您自己的配置文件。

启动管理工具时，可以使用 `-cfg` 命令行参数指定要使用的配置文件，如 第 565 页的『启动管理工具』中所述。 如果在调用工具时未指定配置文件名，那么该工具会尝试装入缺省配置文件 (`JMSAdmin.config`)。 它首先在当前目录中搜索此文件，然后在 *MQ_JAVA_INSTALL_PATH*/bin 目录中搜索此文件，其中 *MQ_JAVA_INSTALL_PATH* 是 IBM MQ classes for JMS 安装的路径。

存储在 LDAP 环境中的 JMS 对象的名称必须符合 LDAP 命名约定。 这些约定之一是对象和上下文名称必须包含前缀，例如 cn= (公共名称) 或 ou= (组织单元)。 管理工具允许您引用没有前缀的对象和上下文名称，从而简化了 LDAP 服务提供者的使用。 如果未提供前缀，那么工具会自动向您提供的名称添加缺省前缀。 对于 LDAP ，这是 cn=。 如果需要，可以通过在配置文件中设置 **NAME_PREFIX** 属性来更改缺省前缀。

注: 您可能需要配置 LDAP 服务器以存储 Java 对象。 有关更多信息，请参阅 LDAP 服务器的文档。

过程

1. 通过配置 **INITIAL_CONTEXT_FACTORY** 属性来定义工具使用的服务提供者。
   此属性的受支持值如下所示:
   - com.sun.jndi.ldap.LdapCtxFactory (对于 LDAP)
   - com.sun.jndi.fscontext.RefFSContextFactory (针对文件系统上下文)
   - z/OS com.ibm.jndi.LDAPCtxFactory 仅在 z/OS 上受支持，并提供对 LDAP 服务器的访问权。 However, this class is incompatible with com.sun.jndi.ldap.LdapCtxFactory, in that objects created using one InitialContextFactory cannot be read or modified using the other.

   您还可以使用管理工具通过 JMSAdmin 配置文件中定义的三个参数来连接到其他 JNDI 上下文。 要使用不同的 InitialContextFactory:

   a) 将 **INITIAL_CONTEXT_FACTORY** 属性设置为必需的类名。

   b) 使用 "**USE_INITIAL_DIR_CONTEXT**、"**NAME_PREFIX** 和 "**NAME_READABILITY_MARKER** 属性定义 "InitialContextFactory 的行为。
      这些属性的设置在样本配置文件注释中进行了描述。

   如果使用其中一个受支持的 **INITIAL_CONTEXT_FACTORY** 值，那么无需定义 **USE_INITIAL_DIR_CONTEXT**， **NAME_PREFIX** 和 **NAME_READABILITY_MARKER** 属性。 但是，如果要覆盖系统缺省值，那么可以为这些属性提供值。 例如，如果对象存储在 LDAP 环境中，那么可以通过将 **NAME_PREFIX** 属性设置为必需前缀来更改工具添加到对象和上下文名称的缺省前缀。

如果省略了三个 InitialContextFactory 属性中的一个或多个，管理工具会根据其他属性的值提供合适的默认值。

2. 通过配置 **PROVIDER_URL** 属性来定义会话初始上下文的 URL。

   此 URL 是该工具执行的所有 JNDI 操作的根。 支持此属性的两种形式:

   - ldap://hostname/contextname
   - 文件: [drive:] /pathname

   LDAP URL 的格式可能有所不同，具体取决于您的 LDAP 提供程序。 请参阅 LDAP 文档以获取更多信息。

3. 定义 JNDI 是否通过配置 **SECURITY_AUTHENTICATION** 属性将安全凭证传递给服务提供者。

   仅当使用 LDAP 服务提供者时，才会使用此属性，并且可以采用以下三个值之一:

   none (匿名认证)
   如果将此参数设置为 none，那么 JNDI 不会将任何安全凭证传递到服务提供者，并且会执行 匿名认证 。

   simple (简单认证)
   如果将该参数设置为 simple，那么会将安全凭证通过 JNDI 传递到底层服务提供者。 这些安全凭证的格式为用户专有名称 (用户 DN) 和密码。

   CRAM-MD5 (CRAM-MD5 认证机制)
   如果将该参数设置为 CRAM-MD5，那么会将安全凭证通过 JNDI 传递到底层服务提供者。 这些安全凭证的格式为用户专有名称 (用户 DN) 和密码。

   如果未提供 **SECURITY_AUTHENTICATION** 属性的有效值，那么该属性缺省为 none。

   如果需要安全凭证，那么在工具初始化时会提示您输入这些凭证。 您可以通过在 JMSAdmin 配置文件中设置 **PROVIDER_USERDN** 和 **PROVIDER_PASSWORD** 属性来避免此情况。

   注: 如果不使用这些属性，那么输入的文本 (包括密码) 将回传到屏幕。 这可能涉及安全问题。

   该工具本身不进行认证; 认证任务将委派给 LDAP 服务器。 LDAP 服务器管理员必须设置并维护对目录的不同部分的访问特权。 请参阅 LDAP 文档以获取更多信息。 如果认证失败，那么工具将显示相应的错误消息并终止。

   有关安全性和 JNDI 的更多详细信息，请参阅 Oracle Java Web 站点 (Oracle Technology Network for Java Developers) 上的文档。

启动管理工具

管理工具具有一个命令行界面，您可以以交互方式使用该界面，也可以使用该界面来启动批处理进程。

关于此任务

交互方式提供命令提示符，您可以在其中输入管理命令。 在批处理方式下，用于启动工具的命令包含包含管理命令脚本的文件的名称。

过程

交互方式

- 要以交互方式启动工具，请输入以下命令:

```
JMSAdmin [-t] [-v] [-cfg config_filename]
```

其中:

-t
启用跟踪 (缺省值为 trace off)

跟踪文件在 "%MQ_JAVA_DATA_PATH%"\errors (Windows) 中生成 或 /var/mqm/trace (AIX and Linux)。 跟踪文件的名称格式如下:

mqjms_*PID*.trc

其中 *PID* 是 JVM 的进程标识。

-v
生成详细输出 (缺省值为 terse 输出)

-cfg 配置文件名
指定备用配置文件。 如果省略此参数，那么将使用缺省配置文件 JMSAdmin.config。 有关配置文件的更多信息，请参阅第 564 页的『配置 JMS 管理工具』。

将显示命令提示符，指示工具已准备好接受管理命令。 此提示最初显示为:

```
InitCtx>
```

指示当前上下文 (即，所有命名和目录操作当前引用的 JNDI 上下文) 是 **PROVIDER_URL** 配置参数中定义的初始上下文。 有关此参数的更多信息，请参阅第 564 页的『配置 JMS 管理工具』。

当您遍历目录名称空间时，提示会更改以反映这一点，以便提示始终显示当前上下文。

批处理模式

- 要以批处理方式启动工具，请输入以下命令:

  ```
  JMSAdmin < test.scp
  ```

  其中 *test.scp* 是包含管理命令的脚本文件。 有关更多信息，请参阅 第 566 页的『使用管理命令』。 文件中的最后一条命令必须是 "END 命令。

使用管理命令

管理工具接受由管理动词及其相应参数组成的命令。

关于此任务

下表列出了在使用管理工具输入命令时可以使用的管理动词。

表 *38:* 管理动词

动词	**缩写**	**描述**
更改	ALT	更改受管对象的至少一个属性
定义	DEF	创建和存储受管对象，或者创建子上下文
DISPLAY	DIS	显示一个或多个存储的受管对象的属性或当前上下文的内容
DELETE	DEL	从名称空间中除去一个或多个受管对象，或者除去空子上下文
更改	chg	更改当前上下文，允许用户在初始上下文下的任何位置遍历目录名称空间 (暂挂安全隔离)
COPY	CP	创建已存储的受管对象的副本，将其存储在备用名称下
移动	MV	更改存储受管对象的名称
END		关闭管理工具

过程

- 如果管理工具尚未启动，请按 第 565 页的『启动管理工具』中所述将其启动。
  将显示命令提示符，指示工具已准备好接受管理命令。 此提示最初显示为:

  ```
  InitCtx>
  ```

  要更改当前上下文，请使用 CHANGE 动词，如 第 567 页的『配置子上下文』中所述。
- 按以下格式输入命令:

```
verb [param]*
```

其中 **verb** 是 第 566 页的表 38 中列出的其中一个管理动词。 所有有效命令都包含一个动词，该动词以其标准或短格式出现在命令的开头。 动词名称不区分大小写。

- 要终止命令，请按 Enter 键，除非要同时输入多个命令，在这种情况下，请在按 Enter 键之前直接输入加号 (+)。

  通常，要终止命令，请按 Enter 键。 但是，可以通过在按 Enter 键之前直接输入加号 (+) 来覆盖此值。 这使您能够输入多行命令，如以下示例中所示:

```
DEFINE Q(BookingsInputQueue) +
QMGR(QM.POLARIS.TEST) +
QUEUE(BOOKINGS.INPUT.QUEUE) +
CCSID(437)
```

- 要关闭管理工具，请使用 **END** 动词。

  此动词不能采用任何参数。

配置子上下文

您可以使用动词 **CHANGE**， **DEFINE**， **DISPLAY** 和 **DELETE** 来配置目录名称空间子上下文。

关于此任务

下表描述了这些动词的用法。

表 *39:* 用于处理子上下文的命令的语法和描述

命令语法	**描述**
DEFINE CTX (ctxName)	尝试创建名称为 ctxName 的当前上下文的子上下文。 如果存在安全违例，子上下文已存在或提供的名称无效，那么将失败。
显示 CTX	显示当前上下文的内容。 使用 a 对受管对象进行注释，使用 [D]对子上下文进行注释。 还会显示每个对象的 Java 类型。
DELETE CTX (ctxName)	尝试删除名称为 ctxName 的当前上下文的子上下文。 如果上下文未找到，非空或存在安全违例，那么将失败。
CHANGE CTX (ctxName)	更改当前上下文，以便它现在引用名为 ctxName 的子上下文。 可以提供 ctxName 的两个特殊值之一: **= 向上** 移至当前上下文的父代 **= INIT** 直接移至初始上下文 如果指定的上下文不存在或存在安全违例，那么将失败。

存储在 LDAP 环境中的 JMS 对象的名称必须符合 LDAP 命名约定。 这些约定之一是对象和上下文名称必须包含前缀，例如 cn= (公共名称) 或 ou= (组织单元)。 管理工具允许您引用没有前缀的对象和上下文名称，从而简化了 LDAP 服务提供者的使用。 如果未提供前缀，那么工具会自动向您提供的名称添加缺省前缀。 对于 LDAP ，这是 cn=。 如果需要，可以通过在配置文件中设置 **NAME_PREFIX** 属性来更改缺省前缀。 有关更多信息，请参阅第 564 页的『配置 JMS 管理工具』。

注: 您可能需要配置 LDAP 服务器以存储 Java 对象。 有关更多信息，请参阅 LDAP 服务器的文档。

创建 JMS 对象

要创建 JMS 连接工厂和目标对象并将其存储在 JNDI 名称空间中，请使用 DEFINE 动词。 要在 LDAP 环境中存储对象，必须为其提供符合特定约定的名称。 管理工具可以通过向对象名添加缺省前缀来帮助您遵守 LDAP 命名约定。

关于此任务

DEFINE 动词使用您指定的类型，名称和属性创建受管对象。 新对象存储在当前上下文中。

存储在 LDAP 环境中的 JMS 对象的名称必须符合 LDAP 命名约定。 这些约定之一是对象和上下文名称必须包含前缀，例如 cn= (公共名称) 或 ou= (组织单元)。 管理工具允许您引用没有前缀的对象和上下文名称，从而简化了 LDAP 服务提供者的使用。 如果未提供前缀，那么工具会自动向您提供的名称添加缺省前缀。 对于 LDAP，这是 cn=。 如果需要，可以通过在配置文件中设置 **NAME_PREFIX** 属性来更改缺省前缀。 有关更多信息，请参阅第 564 页的『配置 JMS 管理工具』。

注: 您可能需要配置 LDAP 服务器以存储 Java 对象。 有关更多信息，请参阅 LDAP 服务器的文档。

过程

1. 如果管理工具尚未启动，请按第 565 页的『启动管理工具』中所述将其启动。
   将显示命令提示符，指示工具已准备好接受管理命令。
2. 确保命令提示符显示要在其中创建新对象的上下文。
   当您启动管理工具时，提示最初显示为:

   ```
   InitCtx>
   ```

   要更改当前上下文，请使用 CHANGE 动词，如第 567 页的『配置子上下文』中所述。
3. 要创建连接工厂，队列目标或主题目标，请使用以下命令语法:

   ```
   DEFINE TYPE (name) [property]*
   ```

   即，输入 DEFINE 动词，后跟 *TYPE* （name）受管对象引用，后跟零个或多个 属性 (请参阅 IBM MQ classes for JMS 对象的属性)。
4. 要创建连接工厂，队列目标或主题目标，请使用以下命令语法:

   ```
   DEFINE TYPE (name) [property]*
   ```

5. 要显示新创建的对象，请使用具有以下命令语法的 DISPLAY 动词:

   ```
   DISPLAY TYPE (name)
   ```

示例

以下示例显示了使用 DEFINE 动词在初始上下文中创建的名为 testQueue 的队列。 由于此对象存储在 LDAP 环境中，虽然未输入带有前缀的对象名称 testQueue，但该工具会自动添加一个对象以确保符合 LDAP 命名约定。 提交命令 DISPLAY Q(testQueue) 还会导致添加此前缀。

```
InitCtx> DEFINE Q(testQueue)

InitCtx> DISPLAY CTX

Contents of InitCtx

a cn=testQueue       com.ibm.mq.jms.MQQueue

1 Object(s)
0 Context(s)
1 Binding(s), 1 Administered
```

创建 *JMS* 对象的样本错误条件

创建对象时，可能会出现许多常见错误情况。

以下是这些错误条件的示例:

CipherSpec 已映射到 CipherSuite

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SSLCIPHERSUITE(RC4_MD5_US)
WARNING: Converting CipherSpec RC4_MD5_US to
CipherSuite SSL_RSA_WITH_RC4_128_MD5
```

对象的属性无效

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PRIORITY(4)
Unable to create a valid object, please check the parameters supplied
Invalid property for a QCF: PRI
```

属性值的类型无效

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) CCSID(english)
Unable to create a valid object, please check the parameters supplied
Invalid value for CCS property: English
```

属性冲突-客户机/bin 编码

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) HOSTNAME(polaris.hursley.ibm.com)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: Client-bindings attribute clash
```

属性冲突-退出初始化

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SECEXITINIT(initStr)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: ExitInit string supplied
without Exit string
```

属性值超出有效范围

```
InitCtx/cn=Trash> DEFINE Q(testQ) PRIORITY(12)
Unable to create a valid object, please check the parameters supplied
Invalid value for PRI property: 12
```

未知的属性

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PIZZA(ham and mushroom)
Unable to create a valid object, please check the parameters supplied
Unknown property: PIZZA
```

以下是从 JMS 应用程序中查找 JNDI 受管对象时在 Windows 上可能出现的错误条件的示例。

1. 如果使用 "WebSphereJNDI 提供程序 com.ibm.websphere.naming.WsnInitialContextFactory,必须使用斜线 (/) 访问子上下文中定义的管理对象；例如，jms/MyQueueName 如果使用反斜线 (\)，则会产生 InvalidNameException。
2. 如果使用 OracleJNDI 提供程序 com.sun.jndi.fscontext.RefFSContextFactory,必须使用反斜杠 (\)来访问子上下文中定义的管理对象；例如，ctx1\\fred。 如果使用正斜线 (/)，则会产生 NameNotFoundException。

配置 JMS 对象

可以使用动词 ALTER， DEFINE， DISPLAY， DELETE， COPY 和 MOVE 来处理目录名称空间中的受管对象。

关于此任务

第 569 页的表 40 总结了这些动词的用法。 将 *TYPE* 替换为表示所需受管对象的关键字，如 第 563 页的『使用管理工具配置 JMS 对象』中所述。

表 *40:* 用于处理受管对象的命令的语法和描述

命令语法	描述
ALTER *TYPE* (名称) [属性] *	尝试使用提供的属性来更新受管对象的属性。 如果存在安全违例，如果找不到指定的对象，或者如果提供的新属性无效，那么将失败。

表 *40:* 用于处理受管对象的命令的语法和描述 *(*继续*)*	
命令语法	**描述**
DEFINE *TYPE* (名称) [属性] *	尝试使用提供的属性创建类型为 *TYPE* 的受管对象，并将其存储在当前上下文中的名称 name 下。 如果存在安全违例，如果提供的名称无效或存在该名称的对象，或者如果提供的属性无效，那么将失败。
DISPLAY *TYPE* (名称)	显示在当前上下文中名称 name 下绑定的类型为 *TYPE* 的受管对象的属性。 如果对象不存在，或者存在安全违例，那么将失败。
DELETE *TYPE* (名称)	尝试从当前上下文中除去名称为 name 的 *TYPE* 类型的受管对象。 如果对象不存在，或者存在安全违例，那么将失败。
COPY *TYPE* (nameA) 类型 (nameB)	生成类型为 *TYPE* 的受管对象的副本，其名称为 nameA，命名为副本 nameB。 这一切都发生在当前上下文的作用域内。 如果要复制的对象不存在，名称为 nameB 的对象存在或者存在安全违例，那么将失败。
MOVE *TYPE* (nameA) *TYPE* (nameB)	将名为 nameA 的类型为 *TYPE* 的受管对象移动 (重命名) 到 nameB。 这一切都发生在当前上下文的作用域内。 如果要移动的对象不存在，名称为 nameB 的对象存在或者存在安全违例，那么将失败。

在 WebSphere Application Server 中配置 JMS 资源

要在 WebSphere Application Server 中配置 JMS 资源，可以使用管理控制台或 wsadmin 命令。

关于此任务

Java Message Service (JMS) 应用程序通常依赖于外部配置的对象，这些对象描述应用程序如何连接到其 JMS 提供程序及其访问的目标。 JMS 应用程序使用 Java Naming Directory Interface (JNDI) 在运行时访问以下类型的对象:

- 激活规范 (由 Java EE 应用程序服务器使用)
- 统一连接工厂 (具有 JMS 1.1，独立于域的 (统一) 连接工厂首选于特定于域的队列连接工厂和主题连接工厂)
- 主题连接工厂 (由 JMS 1.0 应用程序使用)
- 队列连接工厂 (由 JMS 1.0 应用程序使用)
- 队列
- 主题

通过 WebSphere Application Server 中的 IBM MQ 消息传递提供程序， Java Message Service (JMS) 消息传递应用程序可以将 IBM MQ 系统用作 JMS 消息传递资源的外部提供程序。 要启用此方法，请在 WebSphere Application Server 中配置 IBM MQ 消息传递提供程序，以定义用于连接到 IBM MQ 网络上的任何队列管理器的 JMS 资源。

您可以使用 WebSphere Application Server 为应用程序 (例如，队列连接工厂) 配置 IBM MQ 资源，并管理与 JMS 目标关联的消息和预订。 您可以通过 IBM MQ 来管理安全性。

相关任务

将 IBM MQ 与 WebSphere Application Server 结合使用

WebSphere Application Server 主题

使用 IBM MQ 消息传递提供程序进行互操作

使用 IBM MQ 消息传递提供程序管理消息传递

管理控制台面板名称到命令名和 IBM MQ 名称的映射

使用管理控制台配置 JMS 资源

您可以使用 WebSphere Application Server 管理控制台来配置 IBM MQ JMS 提供程序的激活规范，连接工厂和目标。

关于此任务

您可以使用 WebSphere Application Server 管理控制台来创建，查看或修改以下任何资源:

- 激活规范
- 独立于域的连接工厂 (JMS 1.1 或更高版本)
- 队列连接工厂
- 主题连接工厂
- 队列
- 主题

以下步骤概述了您可以使用管理控制台来配置 JMS 资源以用于 IBM MQ 消息传递提供程序的方法。 每个步骤都包含 WebSphere Application Server 产品文档中主题的名称，您可以参阅该主题以获取更多信息。 请参阅 相关链接 ，以获取指向 IBM Documentation 中的这些主题的链接。

在混合版本的 WebSphere Application Server 单元中，可以管理所有版本的节点上的 IBM MQ 资源。 但是，某些属性并非在所有版本上都可用。 在此情况下，只有该特定节点的属性才会显示在管理控制台中。

过程

要创建或配置激活规范以用于 IBM MQ 消息传递提供程序:

- 要创建激活规范，请使用 "创建 IBM MQ JMS 资源" 向导。

  您可以使用向导来指定激活规范的所有详细信息，也可以选择使用客户机通道定义表 (CCDT) 来指定 IBM MQ 的连接详细信息。 当您使用向导指定连接详细信息时，可以选择单独输入主机和端口信息，或者如果您正在使用多实例队列管理器，那么可以选择以连接名称列表的形式输入主机和端口信息。 有关更多信息，请参阅 为 *IBM MQ* 消息传递提供程序创建激活规范。
- 要查看或更改激活规范的配置属性，请使用管理控制台 IBM MQ 消息传递提供程序连接工厂设置面板。

  这些配置属性控制如何创建与关联队列和主题的连接。 有关更多信息，请参阅 为 *IBM MQ* 消息传递提供程序配置激活规范。

要创建或配置用于 IBM MQ 消息传递提供程序的统一连接工厂，队列连接工厂或主题连接工厂:

- 要创建连接工厂，请首先选择要创建的连接工厂类型，然后使用 "创建 IBM MQ JMS 资源" 向导来指定详细信息。
  - 如果 JMS 应用程序仅打算使用点到点消息传递，请为点到点消息传递域创建特定于域的连接工厂，该连接工厂可用于创建专门用于点到点消息传递的连接。
  - 如果 JMS 应用程序仅用于使用发布/预订消息传递，请为发布/预订消息传递域创建特定于域的连接工厂，可用于创建专门用于发布/预订消息传递的连接。
  - 对于 JMS 1.1 或更高版本，创建可同时用于点到点消息传递和发布/预订消息传递的独立于域的连接工厂，允许应用程序在同一事务下执行点到点和发布/预订工作。

  您可以选择是使用向导来指定连接工厂的所有详细信息，还是选择使用客户机通道定义表 (CCDT) 来指定 IBM MQ 的连接详细信息。 当您使用向导指定连接详细信息时，可以选择单独输入主机和端口信息，或者如果您正在使用多实例队列管理器，那么可以选择以连接名称列表的形式输入主机和端口信息。 有关更多信息，请参阅 为 *IBM MQ* 消息传递提供程序创建连接工厂。

要查看或更改连接工厂的配置属性:

- 将管理控制台连接工厂设置面板用于要配置的连接工厂类型。

  配置属性控制如何创建与关联队列和主题的连接。 有关更多信息，请参阅 为 *IBM MQ* 消息传递提供程序配置集合工厂， 为 *IBM MQ* 消息传递提供程序配置队列集合工厂或 为 *IBM MQ* 消息传递提供程序配置主题集合工厂。

要使用 IBM MQ 消息传递提供程序为点到点消息传递配置 JMS 队列目标:

- 使用管理控制台 IBM MQ 消息传递提供程序队列设置面板来定义以下类型的属性:
  - 常规属性，包括管理和 IBM MQ 队列属性。
  - 用于指定如何连接到托管队列的队列管理器的连接属性。
  - 用于控制与 IBM MQ 消息传递提供程序目标建立的连接的行为的高级属性。
  - 队列目标的任何定制属性。

  有关更多信息，请参阅 为 *IBM MQ* 消息传递提供程序配置队列。

要使用 IBM MQ 消息传递提供程序为发布/预订消息传递创建或配置 JMS 主题目标:

- 使用 IBM MQ 消息传递提供程序主题设置面板来定义以下类型的属性:
  - 常规属性，包括管理和 IBM MQ 主题属性。
  - 用于控制与 IBM MQ 消息传递提供程序目标建立的连接的行为的高级属性。
  - 队列目标的任何定制属性。

  有关更多信息，请参阅 为 *IBM MQ* 消息传递提供程序配置主题。

相关概念

第 415 页的『多实例队列管理器』
多实例队列管理器是在不同服务器上配置的同一队列管理器的多个实例。 其中一个队列管理器实例定义为活动实例，另一个实例定义为备用实例。 如果活动实例发生故障，那么备用服务器上的多实例队列管理器将自动重新启动。

相关任务

第 36 页的『配置二进制格式 CCDT』
客户机通道定义表 (CCDT) 确定客户机应用程序用于连接到队列管理器的通道定义和认证信息。 在多平台上，创建队列管理器时会自动创建包含缺省设置的二进制 CCDT。 您可以使用 **runmqsc** 命令来更新二进制 CCDT。

第 351 页的『配置发布/预订消息传递』
您可以启动，停止和显示已排队的发布/预订的状态。 您还可以添加和除去流，以及从代理层次结构中添加和删除队列管理器。

WebSphere Application Server 主题

IBM MQ 消息传递提供程序激活规范
为 IBM MQ 消息传递提供程序创建激活规范
为 IBM MQ 消息传递提供程序配置激活规范
为 IBM MQ 消息传递提供程序创建连接工厂
为 IBM MQ 消息传递提供程序配置统一连接工厂
为 IBM MQ 消息传递提供程序配置队列连接工厂
为 IBM MQ 消息传递提供程序配置主题连接工厂
为 IBM MQ 消息传递提供程序配置队列
为 IBM MQ 消息传递提供程序配置主题

使用 wsadmin 脚本编制命令配置 JMS 资源

您可以使用 WebSphere Application Server wsadmin 脚本编制命令来创建，修改，删除或显示有关 JMS 激活规范，连接工厂，队列和主题的信息。 您还可以显示和管理 IBM MQ 资源适配器的设置。

关于此任务

以下步骤概述了您可以使用 WebSphere Application Server wsadmin 命令来配置 JMS 资源以用于 IBM MQ 消息传递提供程序的方法。 有关如何使用这些命令的更多信息，请参阅 相关链接 以获取指向 WebSphere Application Server 产品文档的链接。

要运行命令，请使用 wsadmin 脚本编制客户机的 AdminTask 对象。

使用命令创建新对象或进行更改后，将更改保存到主配置。 例如，请使用以下命令:

```
AdminConfig.save()
```

要查看可用 IBM MQ 消息传递提供程序管理命令的列表以及每个命令的简要描述，请在 wsadmin 提示符处输入以下命令:

```
print AdminTask.help('WMQAdminCommands')
```

要查看给定命令的概述帮助，请在 wsadmin 提示符处输入以下命令:

```
print AdminTask.help('command_name')
```

过程

要列出在发出命令的作用域中定义的所有 IBM MQ 消息传递提供程序资源，请使用以下命令。

- 要列出激活规范，请使用 **listWMQActivationSpecs** 命令。
- 要列出连接工厂，请使用 **listWMQConnectionFactories** 命令。
- 要列出队列类型目标，请使用 **listWMQQueues** 命令。
- 要列出主题类型目标，请使用 **listWMQTopics** 命令。

要在特定作用域为 IBM MQ 消息传递提供程序创建 JMS 资源，请使用以下命令。

- 要创建激活规范，请使用 **createWMQActivationSpec** 命令。

  您可以通过指定用于建立连接的所有参数来创建激活规范，也可以创建激活规范，以便它使用客户机通道定义表 (CCDT) 来查找要连接的队列管理器。
- 要创建连接工厂，请使用 **createWMQConnectionFactory** 命令，使用 **-type** 参数指定要创建的连接工厂的类型:
  - 如果 JMS 应用程序仅打算使用点到点消息传递，请为点到点消息传递域创建特定于域的连接工厂，该连接工厂可用于创建专门用于点到点消息传递的连接。
  - 如果 JMS 应用程序仅用于使用发布/预订消息传递，请为发布/预订消息传递域创建特定于域的连接工厂，可用于创建专门用于发布/预订消息传递的连接。
  - 对于 JMS 1.1 或更高版本，创建可同时用于点到点消息传递和发布/预订消息传递的独立于域的连接工厂，允许应用程序在同一事务下执行点到点和发布/预订工作。

  缺省类型为独立于域的连接工厂。
- 要创建队列类型目标，请使用 **createWMQQueue** 命令。
- 要创建主题类型目标，请使用 **createWMQTopic** 命令。

要在特定作用域修改 IBM MQ 消息传递提供程序的 JMS 资源，请使用以下命令。

- 要修改激活规范，请使用 **modifyWMQActivationSpec** 命令。

  不能更改激活规范的类型。 例如，您无法创建激活规范，在其中手动输入所有配置信息，然后将其修改为使用 CCDT。
- 要修改连接工厂，请使用 **modifyWMQConnectionFactory** 命令。
- 要修改队列类型目标，请使用 **modifyWMQQueue** 命令。
- 要修改主题类型目标，请使用 **modifyWMQTopic** 命令。

要在特定作用域中删除 IBM MQ 消息传递提供程序的 JMS 资源，请使用以下命令。

- 要删除激活规范，请使用 **deleteWMQActivationSpec** 命令。
- 要删除连接工厂，请使用 **deleteWMQConnectionFactory** 命令。
- 要删除队列类型目标，请使用 **deleteWMQQueue** 命令。
- 要删除主题类型目标，请使用 **deleteWMQTopic** 命令。

要显示有关特定 IBM MQ 消息传递提供程序资源的信息，请使用以下命令。

- 要显示与特定激活规范关联的所有参数及其值，请使用 **showWMQActivationSpec** 命令。
- 要显示与特定连接工厂关联的所有参数及其值，请使用 **showWMQConnectionFactory** 命令。

- 要显示与特定队列类型目标关联的所有参数及其值，请使用 **showWMQQueue** 命令。
- 要显示与主题类型目标相关联的所有参数及其值，请使用 **deleteWMQTopic** 命令。

要管理 IBM MQ 资源适配器或 IBM MQ 消息传递提供程序的设置，请使用以下命令。

- 要管理安装在特定作用域中的 IBM MQ 资源适配器的设置，请使用 **manageWMQ** 命令。
- 要显示所有参数及其可由 **manageWMQ** 命令设置的值，请使用 **showWMQ** 命令。 这些设置与 IBM MQ 资源适配器或 IBM MQ 消息传递提供程序相关。 **showWMQ** 命令还显示在 IBM MQ 资源适配器上设置的任何定制属性。

相关概念

第 415 页的『多实例队列管理器』
多实例队列管理器是在不同服务器上配置的同一队列管理器的多个实例。 其中一个队列管理器实例定义为活动实例，另一个实例定义为备用实例。 如果活动实例发生故障，那么备用服务器上的多实例队列管理器将自动重新启动。

相关任务

第 36 页的『配置二进制格式 CCDT』
客户机通道定义表 (CCDT) 确定客户机应用程序用于连接到队列管理器的通道定义和认证信息。 在多平台上，创建队列管理器时会自动创建包含缺省设置的二进制 CCDT。 您可以使用 **runmqsc** 命令来更新二进制 CCDT。

第 351 页的『配置发布/预订消息传递』
您可以启动，停止和显示已排队的发布/预订的状态。 您还可以添加和除去流，以及从代理层次结构中添加和删除队列管理器。

WebSphere Application Server 个主题

createWMQActivationSpec 命令
createWMQConnectionFactory 命令
createWMQQueue 命令
createWMQTopic 命令
deleteWMQActivationSpec 命令
deleteWMQConnectionFactory 命令
deleteWMQQueue 命令
deleteWMQTopic 命令
listWMQActivationSpecs 命令
listWMQConnectionFactories 命令
listWMQQueues 命令
listWMQTopics 命令
modifyWMQActivationSpec 命令
modifyWMQConnectionFactory 命令
modifyWMQQueue 命令
modifyWMQTopic 命令
showWMQActivationSpec 命令
showWMQConnectionFactory 命令
showWMQQueue 命令
showWMQTopic 命令
showWMQ 命令
manageWMQ 命令

使用 JMS 2.0 共享预订

在 WebSphere Application Server traditional 9.0 中，可以配置 JMS 2.0 共享预订并将其与 IBM MQ 9.0 配合使用。

关于此任务

JMS 2.0 规范引入了共享预订的概念，这使单个预订能够由一个或多个使用者打开。 这些消息在所有这些使用者之间共享。 只要这些使用者连接到同一队列管理器，就没有任何限制。

共享预订可以是持久预订，也可以是非持久预订，其语义与现在称为非共享预订的语义相同。

为了使使用者能够识别要使用的预订，它需要提供预订名称。 这类似于非共享持久预订，但在所有需要共享预订的情况下都需要预订名称。 但是，对于持久共享预订，不需要 clientID; 可以提供一个客户机标识，但它不是必需的。

虽然可以将共享预订视为一种负载均衡机制，但在 IBM MQ 和 JMS 2.0 规范中都没有任何关于如何在使用者之间分发消息的承诺。

在 WebSphere Application Server traditional 9.0 中，预先安装了 IBM MQ 9.0 资源适配器。

以下步骤说明如何使用 WebSphere Application Server traditional 管理控制台将激活规范配置为使用共享持久预订或共享非持久预订。

过程

首先在 JNDI 中创建对象。

1. 以正常方式在 JNDI 中创建主题目标 (请参阅 第 571 页的『使用管理控制台配置 JMS 资源』)。
2. 创建激活规范 (请参阅 第 571 页的『使用管理控制台配置 JMS 资源』)。
   您可以创建具有所需属性的激活规范。 如果要使用持久预订，可以在创建时选择该预订并指定名称。 如果要使用非持久预订，那么此时不能指定名称。 而是需要为预订名称创建定制属性。

更新使用所需定制属性创建的激活规范。 您可能需要指定两个定制属性:

- 在所有情况下，都必须创建定制属性以指定此激活规范应使用共享预订。
- 如果预订创建为非持久预订，那么需要将预订名称属性设置为定制属性。

下表显示了可以为每个定制属性指定的有效值:

属性名	类型	有效值
sharedSubscription	字符串	true 和 false
subscriptionName	字符串	非零长度的 Java 字符串

3. 从 **激活规范集合** 表单中显示的列表中选择激活规范。
   激活规范的详细信息显示在 **IBM MQ 消息传递提供程序激活规范设置** 表单中。
4. 在 **IBM MQ 消息传递提供程序激活规范设置** 表单上，单击 **定制属性**。
   这样会显示 " **定制属性** " 表单。
5. 如果您正在使用非持久预订，请创建 subscriptionName 定制属性。
   在激活规范的 " **定制属性** " 面板上，单击 **新建**，然后输入以下详细信息:

   名称

   定制属性的名称，在本例中为 `subscriptionName`。

   值

   定制属性的值。 可以在 **值** 字段中使用 JNDI 名称，例如 `WASSharedSub0ne`。

   类型

   定制属性的类型。 从列表中选择定制属性类型，在此情况下必须为 `java.lang.String`。
6. 对于共享持久预订和共享非持久预订，请创建 sharedSubscription 定制属性。
   在激活规范的 " **定制属性** " 面板上，单击 **新建**，然后输入以下详细信息:

   名称

   定制属性的名称，在本例中为 `sharedSubscription`。

值

定制属性的值。 要指定激活规范使用共享预订，请将值设置为 `true`。 如果您以后想要停止对此激活规范使用共享预订，那么可以通过将此定制属性的值设置为 `false` 来执行此操作。

类型

定制属性的类型。 从列表中选择定制属性类型，在此情况下必须为 `java.lang.String`。

7. 设置属性后，请重新启动应用程序服务器。

然后，当消息到达时，将驱动激活规范的消息驱动的 bean (MDB)，但只有 MDB 共享发送的消息。

相关概念

克隆和共享的预订

预订耐久性

相关任务

配置入站通信的资源适配器

WebSphere Application Server traditional 9.0 的相关信息

为 IBM MQ 消息传递提供程序配置主题

IBM MQ 消息传递提供程序激活规范

为 IBM MQ 消息传递提供程序创建激活规范

为 IBM MQ 消息传递提供程序配置激活规范

配置 IBM MQ 消息传递提供程序 JMS 资源的定制属性

使用 JMS 2.0 ConnectionFactory 和 Destination Lookup 属性

在 WebSphere Application Server traditional 9.0 中，可以为激活规范的 ConnectionFactoryLookup 和 DestinationLookup 属性提供要用于首选其他激活规范属性的受管对象的 JNDI 名称。

关于此任务

JMS 2.0 规范指定激活规范上用于驱动消息驱动的 bean (MDB) 的两个附加属性。 以前，每个供应商都必须在激活规范上指定定制属性，以提供连接到消息传递系统所需的详细信息，并定义要从中获取消息的目标。

现在，可以使用标准 connectionFactoryLookup 和 destinationLookup 属性来提供要查找和使用的相关对象的 JNDI 名称。 在 WebSphere Application Server traditional 9.0 中，预先安装了 IBM MQ 9.0 资源适配器。

以下步骤显示如何使用 WebSphere Application Server traditional 管理控制台来定制和使用这两个属性。

过程

首先在 JNDI 中创建对象。

1. 在 JNDI 中正常创建 ConnectionFactory (请参阅 第 571 页的『使用管理控制台配置 JMS 资源』)。
2. 正常在 JNDI 中创建目标 (请参阅 第 571 页的『使用管理控制台配置 JMS 资源』)。

   Destination 对象必须具有正确的值。
3. 使用所需的任何值创建激活规范 (请参阅 第 571 页的『使用管理控制台配置 JMS 资源』)。

   您可以创建具有所需属性的激活规范。 但是，您应牢记以下注意事项:

   - 如果您希望 IBM MQ 资源适配器使用 Java EE 连接工厂和目标查找属性，那么创建激活规范时使用的属性不太相关 (请参阅 ActivationSpec ConnectionFactoryLookup 和 DestinationLookup 属性。
   - 但是，仍必须在激活规范上指定尚未在连接工厂或目标上定义的任何属性。 因此，您必须定义连接使用者属性和其他属性，以及实际创建连接时使用的认证信息。
   - 在连接工厂上定义的属性中， ClientID 属性具有特殊处理。 这是因为常见方案是将单个连接工厂用于多个激活规范。 这简化了管理，但是 JMS 规范需要唯一的客户机标识，因此激活规范需要能够覆盖 ConnectionFactory 上设置的任何值。 如果未在激活规范上设置 ClientID，那么将使用连接工厂上的任何值。

请按步骤 第 577 页的『4』中所述使用 WebSphere Application Server 管理控制台更新使用两个新定制属性创建的激活规范，或者改为按步骤 第 577 页的『5』中所述使用注释。

4. 在 WebSphere Application Server 管理控制台中更新激活规范。
   需要在激活规范的定制属性面板上设置这两个属性。 这些属性不存在于主激活规范面板或 "激活规范创建" 向导中。
   a) 从 **激活规范集合** 表单中显示的列表中选择激活规范。
      激活规范的详细信息显示在 **IBM MQ 消息传递提供程序激活规范设置** 表单中。
   b) 在 **IBM MQ 消息传递提供程序激活规范设置** 表单上，单击 **定制属性**。
      这样会显示 " **定制属性** " 表单。
   c) 在 **定制属性** 表单上，创建类型为 java.lang.String 的两个新定制属性。
      在每种情况下，单击 **新建** ，然后输入定制属性的以下详细信息:

      名称

      定制属性的名称， connectionFactoryLookup 或 destinationLookup。

      值

      定制属性的值。 可以在 **值** 字段中使用 JNDI 名称，例如 QuoteCF 和 QuoteQ。

      类型

      定制属性的类型。 从列表中选择定制属性类型，在此情况下必须为 java.lang.String。

   已部署的 MDB 现在将使用这些值来创建连接工厂和目标。 部署 MDB 时，不需要设置 JNDI 值配置。
5. 使用注释代替激活规范。
   也可以使用 MDB 代码中的注释来指定值。 例如，使用 JNDI names QuoteCF 和 QuoteQ，代码如下所示:

```
@MessageDriven(activationConfig = {
        @ActivationConfigProperty(propertyName =  "destinationType" , propertyValue =
"javax.jms.Topic" ),
        @ActivationConfigProperty(propertyName =  "destinationLookup" , propertyValue =
"QuoteQ" ),
        @ActivationConfigProperty(propertyName =  "connectionFactoryLookup" , propertyValue
=  "QuoteCF" )}, mappedName =  "LookupMDB" )
        @TransactionAttribute(TransactionAttributeType.REQUIRED)
        @TransactionManagement(TransactionManagementType.CONTAINER)
        publicclass LookupMDB implements MessageListener {
```

相关任务

配置入站通信的资源适配器

WebSphere Application Server traditional 9.0 的相关信息

为 IBM MQ 消息传递提供程序配置统一连接工厂

为 IBM MQ 消息传递提供程序配置主题

IBM MQ 消息传递提供程序激活规范

为 IBM MQ 消息传递提供程序创建激活规范

为 IBM MQ 消息传递提供程序配置激活规范

配置 IBM MQ 消息传递提供程序 JMS 资源的定制属性

配置应用程序服务器以使用最新的资源适配器维护级别

要确保 IBM MQ 资源适配器在应用 WebSphere Application Server 修订包时自动更新到最新的可用维护级别，可以将环境中的所有服务器配置为使用已应用于每个节点安装的 WebSphere Application Server 修订包中包含的最新版本的资源适配器。

开始之前

要点: 如果要在任何平台上使用 WebSphere Application Server 8.5 或更低版本，请不要将 IBM MQ 8.0 或更高版本的资源适配器安装到应用程序服务器中。 只能将 IBM MQ 8.0 或更高版本的资源适配器部署到支持 JMS 2.0 的应用程序服务器中。 但是， WebSphere Application Server 8.5 或更低版本仅支持 JMS 1.1。 这

些版本的 WebSphere Application Server 随附 IBM WebSphere MQ 7.0 资源适配器，可用于使用 BINDINGS 或 CLIENT 传输来连接到 IBM MQ 8.0 队列管理器。

关于此任务

如果下列任何情况适用于您的配置，并且您希望将环境中的所有服务器配置为使用最新版本的 IBM MQ 资源适配器，请使用此任务:

- 在应用 WebSphere Application Server 7.0.0 Fix Pack 1 或更高版本之后，环境中任何应用程序服务器的 JVM 日志将显示以下 IBM MQ 资源适配器版本信息:

  ```
  WMSG1703I:RAR 实现版本 7.0.0.0-k700-L080820
  ```

- 环境中任何应用程序服务器的 JVM 日志都包含以下条目:

  ```
  WMSG1625E: 无法检测
  the IBM MQ messaging provider code at the specified path <null>
  ```

- 先前已手动更新一个或多个节点，以使用 IBM MQ 资源适配器的特定维护级别，该级别现在已被当前 WebSphere Application Server 维护级别中包含的最新版本的资源适配器取代。

示例所引用的 *profile_root* 目录是 WebSphere Application Server 概要文件的主目录，例如 `C:\Program Files\IBM\WebSphere\AppServer1`。

当您对环境中的所有单元和单服务器安装执行以下步骤时，当应用新的 WebSphere Application Server 修订包时，您的服务器将自动接收对 IBM MQ 资源适配器的维护。

过程

1. 启动应用程序服务器。 如果概要文件是网络部署配置的一部分，请启动 Deployment Manager 和所有 Node Agent。 如果概要文件包含管理代理程序，请启动管理代理程序。
2. 检查 IBM MQ 资源适配器的维护级别。
   a) 打开命令提示符窗口并切换到 *profile_root*`\bin` 目录。
   例如，输入 `cd C:\Program Files\IBM\WebSphere\AppServer1\bin`。
   b) 通过输入 `wsadmin.bat -lang jython` 来启动 wsadmin 工具，然后在提示时输入用户名和密码。
   c) 输入以下命令，然后按两次 "返回":

```
wmqInfoMBeansUnsplit = AdminControl.queryNames("WebSphere:type=WMQInfo,*")
wmqInfoMBeansSplit = AdminUtilities.convertToList(wmqInfoMBeansUnsplit)
for wmqInfoMBean in wmqInfoMBeansSplit: print wmqInfoMBean; print AdminControl.invoke(wmqInfoMBean,
'getInfo', '')
```

   您还可以在 Jacl 中运行此命令。 有关如何执行此操作的更多信息，请参阅 WebSphere Application Server 产品文档中的 确保服务器使用最新的可用 *IBM MQ* 资源适配器维护级别 。
   d) 在命令的显示输出中找到 `WMSG1703I` 消息，并检查资源适配器级别。
   例如，对于 WebSphere Application Server 7.0.1 Fix Pack 5，消息应该为:

   ```
   WMSG1703I: RAR 实现版本 7.0.1.3-k701-103-100812
   ```

   此消息显示版本为 7.0.1.3-k701-103-100812，这是此修订包的正确资源适配器级别。 但是，如果改为显示以下消息，那么这意味着您需要将资源适配器调整为 WebSphere Application Server 7.0.1 Fix Pack 5 的正确维护级别。

   ```
   WMSG1703I: RAR 实现版本 7.0.0.0-k700-L080820
   ```

3. 将以下 Jython 脚本复制到名为 `convertWMQRA.py` 的文件中，然后将其保存到概要文件根目录中，例如 `C:\Program Files\IBM\WebSphere\AppServer1\bin`。

```
ras = AdminUtilities.convertToList(AdminConfig.list('J2CResourceAdapter'))

for ra in ras :
  desc = AdminConfig.showAttribute(ra, "description")
  if (desc == "WAS 7.0 Built In MQ Resource Adapter") or (desc == "WAS 7.0.0.1 Built In MQ Resource
Adapter"):
    print "Updating archivePath and classpath of " + ra
    AdminConfig.modify(ra, [['archivePath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar"]])
    AdminConfig.unsetAttributes(ra, ['classpath'])
    AdminConfig.modify(ra, [['classpath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar"]])
```

```
    AdminConfig.save()
  #end if
#end for
```

提示: 保存文件时，请确保将其保存为 python 文件而不是文本文件。

4. 使用 WebSphere Application Server wsadmin 工具来运行您刚刚创建的 Jython 脚本。

   打开命令提示符并浏览到 WebSphere Application Server 的主目录 (例如 `C:\Program Files\IBM\WebSphere\AppServer1\bin` 目录) 中的 `\bin` 目录，然后输入以下命令并按 "返回":

   ```
   wsadmin -lang jython -f convertWMQRA.py
   ```

   如果提示执行此操作，请输入您的用户名和密码。

   注: 如果对属于网络部署配置的概要文件运行该脚本，那么该脚本将更新该配置中需要更新的所有概要文件。 如果存在预先存在的配置文件不一致，那么可能需要完全再同步。
5. 如果您正在网络部署配置中运行，请确保节点代理程序完全重新同步。 有关更多信息，请参阅 "使用 wsadmin 脚本编制工具同步节点" 或 "添加，管理和除去节点"。
6. 停止概要文件中的所有服务器。 如果概要文件是网络部署配置的一部分，请同时停止配置中的任何集群成员，停止配置中的所有 Node Agent，然后停止 Deployment Manager。 如果概要文件包含管理代理程序，请停止管理代理程序。
7. 从 *profile_root*/bin 目录运行 **osgiCfgInit** 命令。

   `osgiCfgInit` 命令将重置 OSGi 运行时环境所使用的类高速缓存。 如果概要文件是网络部署配置的一部分，请从属于该配置的每个概要文件的 `profile_root/bin` 目录运行 **osgiCfgInit** 命令。
8. 重新启动概要文件中的所有服务器。 如果概要文件是网络部署配置的一部分，那么还请重新启动配置中的任何集群成员，重新启动配置中的所有 Node Agent，然后重新启动 Deployment Manager。 如果概要文件包含管理代理程序，请重新启动管理代理程序。
9. 重复步骤 2 以检查资源适配器现在是否处于正确的级别。

下一步做什么

如果在执行本主题中描述的步骤后继续迁到问题，并且先前已使用 WebSphere Application Server 管理控制台中的 JMS 提供程序设置面板上的 **更新资源适配器** 按钮在环境中的任何节点上更新 IBM MQ 资源适配器，那么您可能迁到 APAR PM10308 中描述的问题。

相关任务

使用 IBM MQ 资源适配器

WebSphere Application Server 8.5.5 的相关信息

确保服务器使用最新的可用 IBM MQ 资源适配器维护级别

使用 wsadmin 脚本编制工具同步节点

添加，管理和除去节点

JMS 提供程序设置

配置 JMS PROVIDERVERSION 属性

IBM MQ 消息传递提供者具有三种运行方式：正常方式、带有限制的正常方式和迁移方式。 您可以设置 JMS **PROVIDERVERSION** 属性，以选择 JMS 应用程序用于发布和预订的这些方式。

关于此任务

可以通过设置 PROVIDERVERSION 连接工厂属性来主要控制 IBM MQ 消息传递提供程序操作方式的选择。 如果未指定方式，那么还可以自动选择操作方式。

PROVIDERVERSION 属性区分三种 IBM MQ 消息传递提供程序操作方式:

IBM MQ 消息传递提供者正常方式
正常方式使用 IBM MQ 队列管理器的所有功能来实现 JMS。 此方式优化为使用 JMS 2.0 API 和功能。

IBM MQ 消息传递提供者带有限制的正常方式
具有限制的正常方式使用 JMS 2.0 API，但不使用新功能，即共享预订，延迟交付和异步发送。

IBM MQ 消息传递提供者迁移方式
通过迁移方式，您可以连接到 IBM MQ 8.0 或更高版本的队列管理器，但不会使用 IBM WebSphere MQ 7.0 或更高版本的队列管理器的任何功能，例如预读和流式方法。

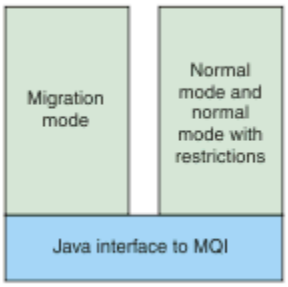


图 *92:* 消息传递提供程序方式

过程

要为特定连接工厂配置 **PROVIDERVERSION** 属性，请执行以下操作:

- 要使用 IBM MQ Explorer 配置 **PROVIDERVERSION** 属性，请参阅 配置队列管理器和对象。
- 要使用 JMS 管理工具配置 **PROVIDERVERSION** 属性，请参阅 配置队列管理器和对象。
- 要使用 IBM JMS 扩展或 IBM MQ JMS 扩展在 JMS 应用程序中配置 **PROVIDERVERSION** 属性，请参阅在 IBM MQ classes for JMS 应用程序中创建和配置连接工厂和目标。

要覆盖 JVM 中所有连接工厂的连接工厂提供程序方式设置:

- 要覆盖连接工厂提供程序方式设置，请使用 `com.ibm.msg.client.wmq.overrideProviderVersion` 属性
  如果您无法更改正在使用的连接工厂，那么可以使用 `com.ibm.msg.client.wmq.overrideProviderVersion` 属性来覆盖连接工厂上的任何设置。 此覆盖操作适用于 JVM 中的所有连接工厂，但不会修改实际的连接工厂对象。

相关任务
JMS 提供程序版本故障诊断
相关参考
PROVIDERVERSION
连接工厂属性
IBM MQ classes for JMS 对象属性之间的依赖关系

IBM MQ 消息传递提供程序操作方式

通过将连接工厂的 PROVIDERVERSION 属性设置为相应的值，可以选择 JMS 应用程序用于发布和预订的 IBM MQ 消息传递提供程序操作方式。 在某些情况下， PROVIDERVERSION 属性设置为未指定，在这种情况下， JMS 客户机使用算法来确定要使用的操作方式。

PROVIDERVERSION 属性值

可以将连接工厂 **PROVIDERVERSION** 属性设置为以下任何值:

8 - 正常方式
JMS 应用程序使用正常方式。 此方式使用 IBM MQ 队列管理器的所有功能来实现 JMS。

7 - 存在限制的正常方式
JMS 应用程序使用具有限制的正常方式。 此方式使用 JMS 2.0 API，但不使用诸如共享预订、延迟传递或异步发送之类的新功能。

6 - 迁移方式
JMS 应用程序使用迁移方式。 在迁移方式下， IBM MQ classes for JMS 使用与 IBM WebSphere MQ 6.0 随附的功能部件和算法类似的功能部件和算法。

unspecified (缺省值)
JMS 客户机使用算法来确定使用哪种操作方式。

为 **PROVIDERVERSION** 属性指定的值必须是字符串。 如果指定选项 8、7 或 6，那么可以采用以下任何格式执行此操作:

- V.R.M.F
- V.R.M
- V.R
- V

其中 V、R、M 和 F 是大于或等于零的整数值。 额外的 R、M 和 F 值是可选的，可供您在需要细颗粒控制时使用。 例如，如果要使用 **PROVIDERVERSION** 级别的 7，那么可以设置 **PROVIDERVERSION** = 7， 7.0，7.0.0 或 7.0.0.0。

连接工厂对象的类型

可以为以下类型的连接工厂对象设置 **PROVIDERVERSION** 属性:

- MQConnectionFactory
- MQQueueConnectionFactory
- MQTopicConnectionFactory
- MQXAConnectionFactory
- MQXAQueueConnectionFactory
- MQXAQueueConnectionFactory
- MQXAQueueConnectionFactory
- MQXATopicConnectionFactory

有关这些不同类型的连接工厂的更多信息，请参阅 第 563 页的『使用管理工具配置 JMS 对象』。

相关概念
IBM MQ 消息传递提供程序

PROVIDERVERSION 正常方式

正常方式使用 IBM MQ 队列管理器的所有功能来实现 JMS。 此方式优化为使用 JMS 2.0 API 和功能。

以下流程图显示了 JMS 客户机为确定是否能够创建正常方式连接而执行的检查。

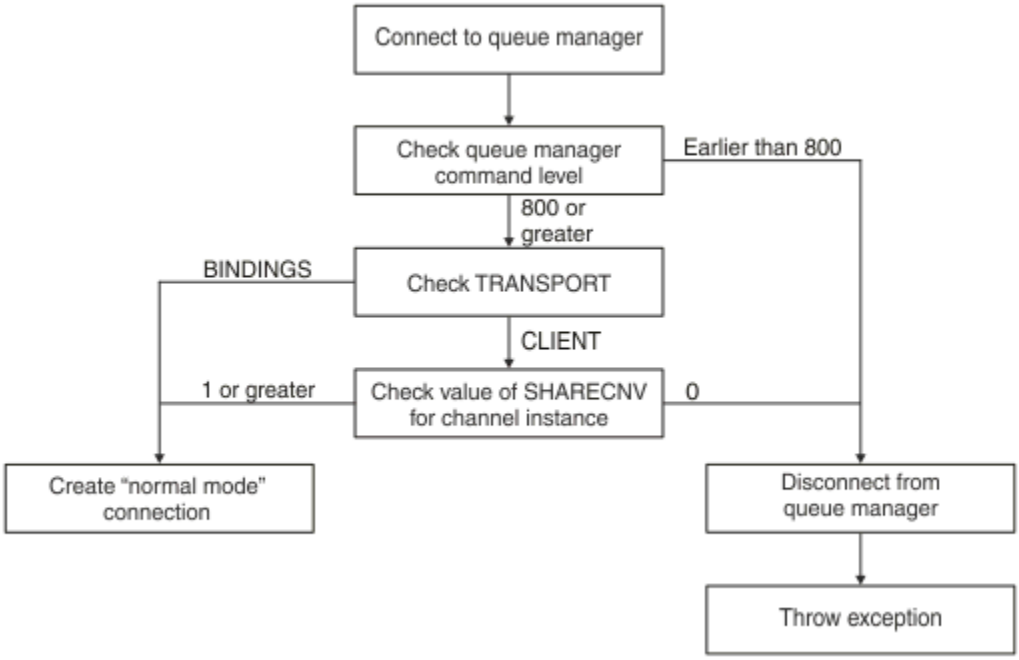


图 *93: PROVIDERVERSION* 正常方式

如果连接工厂设置中指定的队列管理器的命令级别为 800 或更高级别，并且连接工厂的 **TRANSPORT** 属性设置为 BINDINGS，那么将创建正常方式连接，而不检查任何其他属性。

如果连接工厂设置中指定的队列管理器的命令级别为 800 或更高级别，并且 **TRANSPORT** 属性设置为 CLIENT，那么还会检查服务器连接通道上的 **SHARECNV** 属性。 需要执行此检查的原因是 IBM MQ 消息传递提供程序正常方式会使用共享对话功能。 因此，为使正常方式连接尝试成功，用于控制可共享对话数的 **SHARECNV** 属性的值必须大于等于 1。

如果流程图中显示的所有检查都成功，那么将创建与队列管理器的正常方式连接，然后可以使用所有 JMS 2.0 API 和功能 (即，异步发送，延迟交付和共享预订)。

尝试创建正常方式连接可能会因以下任一原因而失败：

- 连接工厂设置中指定的队列管理器的命令级别低于 800。 在此情况下，`createConnection` 方法将失败，并返回异常 `JMSFMQ0003`。
- 服务器连接通道上的 **SHARECNV** 属性设置为 `0`。 如果此属性的值不为 1 或更大，那么 `createConnection` 方法将失败，发生异常 `JMSCC5007`。

相关信息

IBM MQ classes for JMS 对象属性之间的依赖关系
DEFINE CHANNEL（SHARECNV 属性）
TRANSPORT

具有限制的 *PROVIDERVERSION* 正常方式

具有限制的正常方式使用 JMS 2.0 API，但不使用新的 IBM MQ 8.0 或更高版本功能，例如共享预订，延迟交付或异步发送。

以下流程图显示了 JMS 客户机为确定是否可以创建具有限制连接的正常方式而进行的检查。

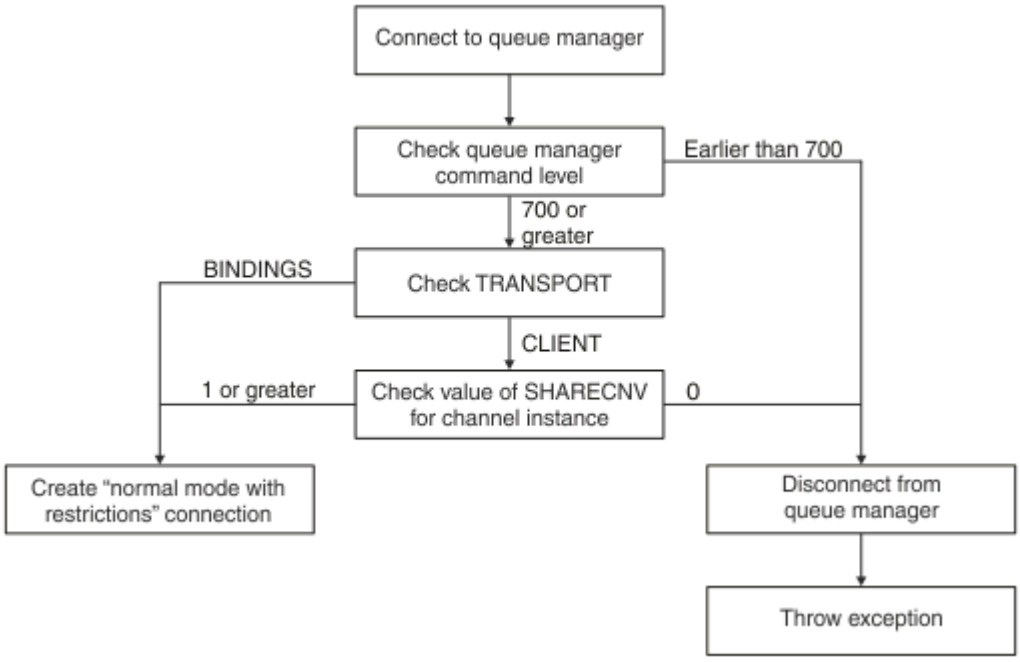


图 *94:* 具有限制的 *PROVIDERVERSION* 正常方式

如果在连接工厂设置中指定的队列管理器的命令级别为 700 或更高，并且连接工厂的 **TRANSPORT** 属性设置为 BINDINGS，那么将创建正常方式连接而不检查任何其他属性。

如果在连接工厂设置中指定的队列管理器的命令级别为 700 或更高，并且 **TRANSPORT** 属性设置为 CLIENT，那么还将检查服务器连接通道上的 **SHARECNV** 属性。 需要进行此检查，因为具有限制的 IBM MQ 消息传递提供程序正常方式使用共享对话功能。 因此，对于具有限制连接尝试成功的正常方式，用于控制可共享的对话数的 **SHARECNV** 属性必须具有值 1 或更大的值。

如果流程图中显示的所有检查都成功，那么将创建连接到队列管理器的具有限制的正常方式，然后您可以使用 JMS 2.0 API，但不能使用异步发送，延迟交付或共享预订功能。

由于以下任一原因，尝试创建具有限制的正常方式连接失败：

- 在连接工厂设置中指定的队列管理器的命令级别低于 700。 在此情况下， `createConnection` 方法失败，发生异常 `JMSFCC5008`。
- 服务器连接通道上的 **SHARECNV** 属性设置为 `0`。 如果此属性的值不为 1 或更大，那么 `createConnection` 方法将失败，发生异常 `JMSCC5007`。

相关信息

IBM MQ classes for JMS 对象属性之间的依赖关系
DEFINE CHANNEL（SHARECNV 属性）
TRANSPORT

PROVIDERVERSION 迁移方式

对于迁移方式， IBM MQ classes for JMS 使用与 IBM WebSphere MQ 6.0 随附的功能和算法类似的功能和算法，例如排队的发布/预订，在客户端上实现的选择，非多路复用通道以及用于实现侦听器的轮询。

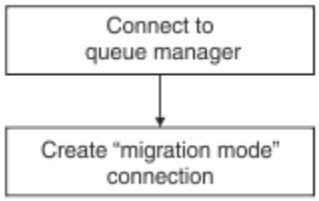


图 *95: PROVIDERVERSION* 迁移方式

如果要使用 IBM MQ Enterprise Transport V 6.0 连接到 WebSphere Message Broker 6.0 或 WebSphere Message Broker 6.1，那么必须使用迁移方式。

您可以使用迁移方式连接到 IBM MQ 8.0 队列管理器，但不会使用 IBM MQ classes for JMS 队列管理器的任何新功能，例如，预读或流式方法。 如果 IBM MQ 8.0 或更高版本的客户机连接到分布式平台上的 IBM MQ 8.0 或更高版本的队列管理器， z/OS 或 IBM MQ for z/OS 8.0 或更高版本的队列管理器， 那么消息选择由队列管理器完成，而不是在客户机系统上完成。

如果指定了 IBM MQ 消息传递提供程序迁移方式，并且 IBM MQ classes for JMS 尝试使用任何 JMS 2.0 API，那么 API 方法调用将失败，发生异常 `JMSCC5007`。

相关信息

IBM MQ classes for JMS 对象属性之间的依赖关系

TRANSPORT

PROVIDERVERSION 未指定

如果未指定连接工厂的 **PROVIDERVERSION** 属性，那么 JMS 客户机将使用算法来确定用于连接到队列管理器的操作方式。 当连接工厂与新版本的 IBM MQ classes for JMS 配合使用时，使用先前版本的 IBM MQ classes for JMS 在 JNDI 名称空间中创建的连接工厂将采用未指定的值。

如果未指定 **PROVIDERVERSION** 属性，那么将在调用 `createConnection` 方法时使用算法。 该算法检查许多连接工厂属性，以确定是否需要 IBM MQ 消息传递提供程序正常方式，具有限制的正常方式或 IBM MQ 消息传递提供程序迁移方式。 总是先尝试正常方式，然后再尝试有限制的正常方式。 如果这两种类型的连接都无法建立，那么 JMS 客户机将断开与队列管理器的连接，然后再次与队列管理器连接以尝试迁移方式连接。

检查 BROKERVER， BROKERQMGR， PSMODE 和 BROKERCONQ 属性

属性值的检查以 **BROKERVER** 属性开始，如 图 1 中所示。

如果 **BROKERVER** 属性设置为 V1，那么接下来将选中 **TRANSPORT** 属性，如 图 2 中所示。 但是，如果 **BROKERVER** 属性设置为 V2，那么将先执行 图 1 中显示的其他检查，然后再检查 **TRANSPORT** 属性。

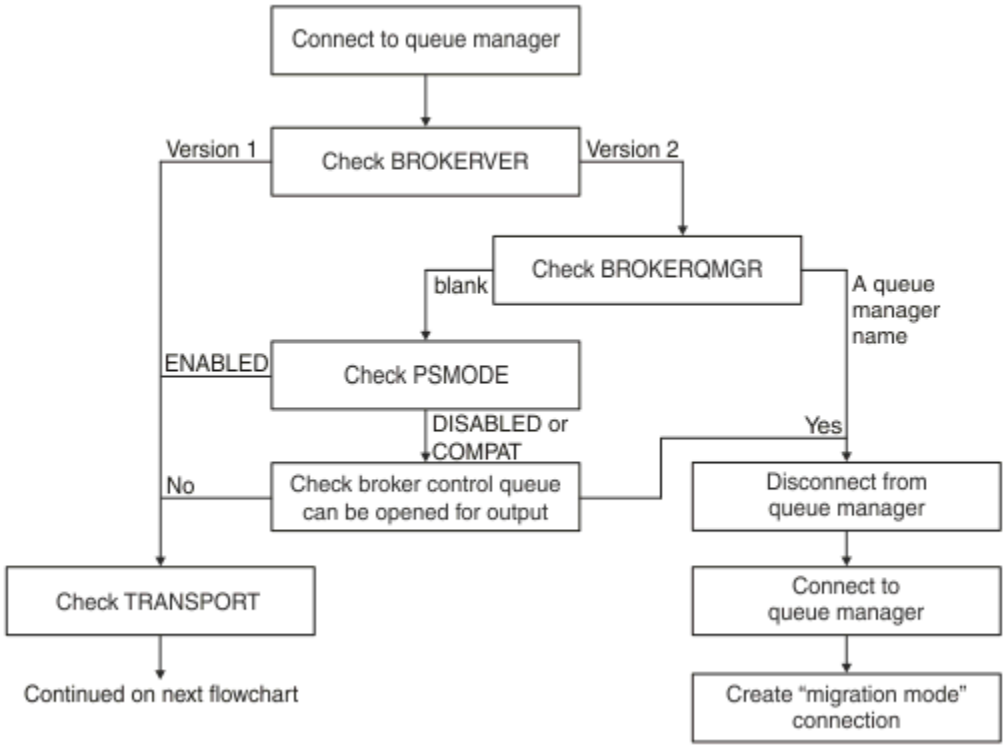


图 *96:* 未指定 *PROVIDERVERSION*

如果 **BROKERVER** 属性设置为 V2，那么要实现正常方式连接， **BROKERQMGR** 属性必须为 空白。 此外，必须将队列管理器上的 **PSMODE** 属性设置为 ENABLED ，或者不能打开由 **BROKERCONQ** 属性指定的代理控制队列以进行输出。

如果将属性值设置为正常方式连接所需的值，那么检查接下来的操作将移至 **TRANSPORT** 属性，如 图 2 中所示。

如果未根据正常方式连接的需要设置属性值，那么 JMS 客户机将断开与队列管理器的连接，然后重新连接并创建迁移方式连接。 在以下情况下会发生此情况:

- 如果 **BROKERQMGR** 属性为 blank ，并且队列管理器上的 **PSMODE** 属性设置为 COMPAT 或 DISABLED ，那么可以打开由 **BROKERCONQ** 属性指定的代理控制队列以进行输出 (即， MQOPEN 表示输出成功)。
- 如果 **BROKERQMGR** 属性指定队列名称。

检查 TRANSPORT 属性和命令级别

图 2 显示了针对队列管理器的 **TRANSPORT** 属性和命令级别进行的检查。

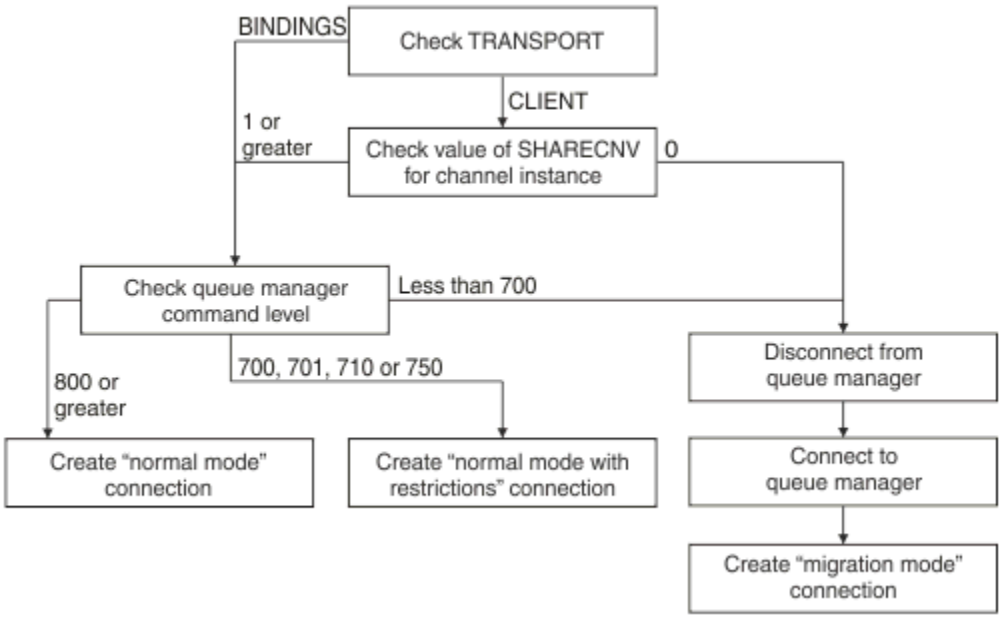


图 *97:* 未指定 *PROVIDERVERSION (续)*

将在以下任一情况下创建正常方式连接:

- 连接工厂的 **TRANSPORT** 属性设置为 BINDINGS，并且队列管理器的命令级别为 800 或更高。
- **TRANSPORT** 属性设置为 CLIENT，服务器连接通道上的 **SHARECNV** 属性的值为 1 或更高，队列管理器的命令级别为 800 或更高。

如果队列管理器的命令级别为 750，那么将创建与队列管理器的限制连接的正常方式。

如果 **TRANSPORT** 属性设置为 CLIENT 并且服务器连接通道上的 **SHARECNV** 属性的值为 0，那么还会创建迁移方式连接。

相关信息

IBM MQ classes for JMS 对象属性之间的依赖关系
ALTER QMGR (PSMODE 属性)
BROKERCONQ
BROKERQMGR
BROKERVER
DEFINE CHANNEL（SHARECNV 属性）
TRANSPORT

何时覆盖 PROVIDERVERSION 缺省设置

如果使用先前版本的 IBM MQ classes for JMS 在 JNDI 名称空间中创建的连接工厂与新版本的 IBM MQ classes for JMS 配合使用，那么连接工厂的 **PROVIDERVERSION** 属性将设置为缺省值 `unspecified`，并使用算法来确定使用的 IBM MQ 消息传递提供程序操作方式。 但是，有两种情况必须覆盖 **PROVIDERVERSION** 属性的缺省选择，以便 IBM MQ classes for JMS 可以正常工作。

注: 本主题中描述的迁移方式用于从 IBM WebSphere MQ 6.0 迁移到 IBM WebSphere MQ 7.0。 它不适用于从更高发行版进行的迁移。

IBM WebSphere MQ 6.0， WebSphere Application Server 6.0 和 WebSphere Message Broker 6 不支持，因此本主题仅供参考。

当 **PROVIDERVERSION** 属性设置为缺省值 `unspecified` 时，将使用算法来确定要使用的操作方式，如 第 584 页的『PROVIDERVERSION 未指定』中所述。 但是，不能在以下两种情况下使用此算法。

1. 如果 WebSphere Message Broker 和 WebSphere Event Broker 处于兼容性方式，那么必须为 WebSphere Message Broker 和 WebSphere Event Broker 的 **PROVIDERVERSION** 属性指定值才能正常工作。
2. 如果您正在使用 WebSphere Application Server 6.0.1， WebSphere Application Server 6.0.2 或 WebSphere Application Server 6.1，那么将使用 WebSphere Application Server 管理控制台来定义连接工厂。

   在 WebSphere Application Server 中，连接工厂上 **BROKERVER** 属性的缺省值为 V2。 使用 JMS 管理工具 **JMSAdmin** 或 IBM MQ Explorer 创建的连接工厂的 **BROKERVER** 属性的缺省值为 V1。 此属性现在在 IBM MQ 中为 `unspecified` 。

如果 **BROKERVER** 属性设置为 V2，因为它是由 WebSphere Application Server 创建的，或者连接工厂之前已用于发布/预订，并且现有队列管理器定义了 **BROKERCONQ** 属性 (因为之前已用于发布/预订消息传递) ，那么将使用 IBM MQ 消息传递提供程序迁移方式。

但是，如果您希望应用程序使用对等通信，并且应用程序正在使用已用于发布/预订的现有队列管理器，并且将 **BROKERVER** 设置为 2 的连接工厂 (这是在 WebSphere Application Server 中创建连接工厂时的缺省设置) ，那么将使用 IBM MQ 消息传递提供程序迁移方式。 在这种情况下，不需要使用 IBM MQ 消息传递提供程序迁移方式; 请改为使用 IBM MQ 消息传递提供程序正常方式。 您可以使用下列其中一种方法来处理此问题:

- 将 **BROKERVER** 设置为 1 或未指定。 您选择的选项取决于您的应用程序。
- 将 **PROVIDERVERSION** 设置为 8 或 7，这是 WebSphere Application Server 6.1 中的定制属性。

  或者，使用客户机配置属性，或者修改已连接的队列管理器以使其未设置 **BROKERCONQ** 属性，或者使该队列不可用。

在 WebSphere Application Server 中配置提供者版本信息

要在 WebSphere Application Server 中配置提供者版本信息，可以使用管理控制台或 wsadmin 命令。

过程

要在 WebSphere Application Server 中配置 IBM MQ 连接工厂或激活规范对象的提供程序版本信息，请参阅相关信息 以获取指向 WebSphere Application Server 产品文档中的进一步信息的链接。

WebSphere Application Server 8.5.5 的相关信息
IBM MQ 消息传递提供程序连接工厂设置
createWMQConnectionFactory 命令
IBM MQ 消息传递提供程序激活规范设置
createWMQActivationSpec 命令
WebSphere Application Server 8.0.0 的相关信息
IBM MQ 消息传递提供程序连接工厂设置
createWMQConnectionFactory 命令
IBM MQ 激活规范设置
createWMQActivationSpec 命令
WebSphere Application Server 7.0.0 的相关信息
IBM MQ 消息传递提供程序连接工厂设置
createWMQConnectionFactory 命令
IBM MQ 激活规范设置
createWMQActivationSpec 命令

除去 WebSphere Application Server 持久预订

将 IBM MQ 消息传递提供者与 WebSphere Application Server 7.0 和 WebSphere Application Server 8.0 结合使用时，将不会移除已绑定到激活规范的消息驱动的 bean 应用程序所创建的持久预订。 可使用 IBM MQ Explorer 或 IBM MQ 命令行实用程序来除去持久预订。

关于本任务

可以将除去持久预订的消息驱动的 bean 应用程序配置为使用侦听器端口或激活规范，前提是该应用程序正在使用 IBM MQ 消息传递提供程序正常方式 连接到 IBM MQ 的 WebSphere Application Server 7.0 或 WebSphere Application Server 8.0 实例中运行。

如果消息驱动 bean 应用程序已绑定到侦听器端口，那么 IBM MQ 消息传递提供者将在首次启动该应用程序时为该应用程序创建持久预订。 在从应用程序服务器中卸载消息驱动 bean 应用程序并重新启动应用程序服务器后，将移除该持久预订。

绑定到激活规范的消息驱动 bean 应用程序的工作方式稍有不同。 首次启动该应用程序时将为该应用程序创建持久预订。 但是，卸载该应用程序并重新启动应用程序服务器时，不会移除该持久预订。

这可能会导致 IBM MQ 发布/预订引擎上针对不再安装在 WebSphere Application Server 系统中的应用程序保留大量持久预订。 这些预订称为“孤立预订”，可能在发布/预订引擎运行时导致队列管理器上发生问题。

将消息发布到主题时，IBM MQ 发布/预订引擎将为已注册到该主题的每个持久预订创建该消息的副本，并将该副本放入内部队列中。 然后，使用该持久预订的应用程序将从该内部队列中选取并使用此消息。

如果先前使用该持久预订的消息驱动的 bean 应用程序已被卸载，那么将继续为该应用程序创建已发布消息的副本。 但是，永远不会处理这些消息，这意味着内部队列上可能保留大量永远不会被移除的消息。

开始之前

已注册到 IBM MQ 发布/预订引擎的预订都有关联的“预订名称”。

WebSphere Application Server IBM MQ 消息传递提供者为已绑定到激活规范的消息驱动 bean 创建的持久预订将具有以下格式的“预订名称”：

```
JMS:queue manager name:client identifier:subscription name
```

其中：

queue manager name
这是运行发布/预订引擎的 IBM MQ 队列管理器的名称。

client identifier
这是消息驱动 bean 所绑定到的激活规范的“客户机标识”属性值。

subscription name
这是消息驱动的 bean 应用程序已配置为使用的激活规范的“预订名称”属性值。

例如，假设我们具有已设置为连接到队列管理器 testQM 的激活规范。 该激活规范设置了以下属性：

- 客户机标识 = testClientID
- 预订名称 = durableSubscription1

如果取得持久预订的消息驱动 bean 已绑定到此激活规范，那么将在队列管理器 testQM 的 IBM MQ 发布/预订引擎上创建具有以下“预订名称”的预订：

- JMS:testQM:testClientID:durableSubscription1

已注册到给定队列管理器的 IBM MQ 发布/预订引擎的预订可以通过以下两种方式之一来进行查看：

- 第一个选项是使用 MQ Explorer。 当 MQ Explorer 已连接到用于发布/预订工作的队列管理器时，可以通过单击导航窗格中的 `IBM WebSphere MQ ->queue manager name-> Subscriptions` 条目来查看当前已向发布/预订引擎注册的订户列表。
- 查看已向发布/预订引擎注册的预订的另一种方法是使用 IBM MQ 命令行实用程序 **runmqsc** 并运行命令 **display sub**。 要执行此操作，请打开命令提示符，切换到 *WebSphere MQ*\bin 目录，然后输入以下命令以启动 **runmqsc**：

  ```
  – runmqsc queue manager name
  ```

  当启动 **runmqsc** 实用程序时，请输入以下命令以列出当前注册到发布/预订引擎（在 **runmqsc** 所连接的队列管理器上运行）的所有持久预订：

  ```
  – display sub(*) durable
  ```

要检查注册到发布/预订引擎的持久预订是否仍处于活动状态：

1. 生成已注册到发布/预订引擎的持久预订列表。
2. 对于每个持久预订：
   - 查看持久订户的预订名称，并记录 *client identifier* 和 *subscription name* 值。
   - 查看连接到此发布/预订引擎的 WebSphere Application Server 系统。 查看是否定义了任何“客户机标识”属性与 *client identifier* 值匹配且预订名称属性与 *subscription name* 匹配的激活规范。
   - 如果未找到具有与 IBM MQ 预订名称中的 客户机标识 和 预订名称 字段相匹配的客户机标识和预订名称属性的激活规范，那么不存在使用此持久预订的激活规范。 可以删除该持久预订。
   - 如果定义了与持久预订名称匹配的激活规范，那么需要执行的最后一项检查是查看是否有消息驱动的 bean 应用程序在使用此激活规范。 要执行此操作：
     - 记录取得当前所查看持久预订的激活规范的 JNDI 名称。
     - 在 WebSphere Application Server 管理控制台中，针对已安装的每个消息驱动的 bean 应用程序调出“配置”窗格。
     - 单击“配置”窗格中的“消息驱动的 Bean 侦听器绑定”链接。
     - 此时将显示包含该消息驱动的 bean 应用程序相关信息的表。 如果已选中“绑定”列中的“激活规范”单选按钮，并且“目标资源 JNDI 名称”字段包含已取得该持久预订的激活规范的 JNDI 名称，那么表示该预订仍在使用中，所以不能将其删除。
     - 如果未找到使用该激活规范的消息驱动的 bean 应用程序，那么可以删除该持久预订。

过程

识别出“孤立”持久预订之后，便可以使用 IBM MQ Explorer 或 IBM MQ 命令行实用程序 **runmqsc** 来将其删除。

要使用 IBM MQ Explorer 删除“孤立”持久预订：

1. 突出显示该预订对应的条目
2. 右键单击该条目，然后选择 **删除 ...** 从菜单中获取。 这将显示确认窗口。
3. 检查确认窗口中显示的预订名称是否正确，然后单击**是**。

IBM MQ Explorer 现在从发布/预订引擎中删除该预订，并清除与其关联的任何内部资源（例如，针对该持久预订所注册到的主题发布的未处理消息）。

要使用 IBM MQ 命令行实用程序 **runmqsc** 删除 "孤立" 持久预订，必须运行命令 **delete sub**：

1. 打开命令提示符会话
2. 浏览至 *IBM MQ*\bin 目录
3. 输入以下命令以启动 **runmqsc**：

   ```
   runmqsc queue manager name
   ```

4. 启动 **runmqsc** 实用程序时，请输入：

   ```
   delete sub(Subscription name)
   ```

   其中，*Subscriptionname* 是持久预订的预订名称，其格式如下：

   - JMS:*queue manager name*:*client identifier*:*subscription name*

配置 Managed File Transfer

安装完成以后，您可以配置 Managed File Transfer 的功能部件。

您可以利用 IBM MQ 高可用性解决方案来提高 Managed File Transfer 配置的弹性。 如果代理程序使用复制的数据队列管理器 (RDQM)，那么必须将其配置为使用浮动 IP 地址功能。 这意味着代理程序使用相同的 IP 地址与当前正在运行的三个 RDQM 实例中的任何一个进行通信，并在故障转移时自动重新连接 (请参阅 RDQM 高可用性 和 创建和删除浮动 IP 地址)。 如果使用多实例队列管理器解决方案，那么应用程序将使用

不同的 IP 地址与每个实例进行通信，由故障转移时的客户机重新连接处理 (请参阅 多实例队列管理器 和 通道和客户机重新连接)。

相关概念

使用 Managed File Transfer 的提示与技巧

相关任务

监视 MFT 资源

使用用户出口定制 MFT

配置 MQMFTCredentials.xml

保护 Managed File Transfer

指定要使用 MFT 运行的程序

故障诊断 Managed File Transfer

管理 Managed File Transfer

相关参考

MFT 命令

MFT `agent.properties` 文件

MFT 恢复和重新启动

多平台上的 MFT 配置选项

Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。 这些属性文件位于安装产品时所定义的配置目录中。

您可以具有多组配置选项，每组配置选项包含一组目录和属性文件。 这些属性文件中定义的值由所有 Managed File Transfer 命令用作缺省参数，除非在命令行中显式指定其他值。

要更改一组希望使用的缺省配置选项，可以使用 **fteChangeDefaultConfigurationOptions** 命令。 要更改一组希望用于个别命令的配置选项，您可以将参数 **-p** 与任何 Managed File Transfer 命令一起使用。

配置选项集的名称是协调队列管理器的名称，建议不要进行更改。 然而，可以更改配置选项集的名称，但必须更改 `config` 和 `logs` 目录的名称。 在以下示例中，配置选项集的名称表示为 *coordination_qmgr_name*。

配置选项目录结构

在配置产品时，会在配置目录下生成如下结构的目录和属性文件。 您也可以通过以下命令更改这些目录和属性文件：**fteSetupCoordination**、**fteSetupCommands**、**fteChangeDefaultConfiguration** 和 **fteCreateAgent**。

```
MQ_DATA_PATH/mqft/
    config/
       coordination_qmgr_name/
            coordination.properties
            command.properties
            agents/
               agent_name/
                    agent.properties
                    exits
            loggers/
               logger_name
                    logger.properties
    installations/
       installation_name/
            installation.properties
```

coordination_qmgr_name 目录是配置选项目录。 配置目录中可以有多个配置选项目录。 *agent_name* 目录是代理目录。 除了包含 `agent.properties` 文件外，该目录还包含 `exits` 目录，这是用户出口例程以及 **fteCreateBridgeAgent** 和 **fteCreateCDAgent** 命令生成的各种 XML 文件的缺省位置。 一组配置选项的 `agents` 目录中可以有多个代理目录。

属性文件

installation.properties
`installation.properties` 文件指定一组缺省配置选项的名称。 该条目将 Managed File Transfer 指向包含要使用的配置的一组结构化目录和属性文件。 通常，一组配置选项的名称是关联的协调队列管理器的名称。 有关 `installation.properties` 文件的更多信息，请参阅 MFT installation.properties 文件。

coordination.properties
`coordination.properties` 文件指定到协调队列管理器的连接详细信息。 由于一些 Managed File Transfer 安装可能共享同一协调队列管理器，因此您可以将符号链接用于共享驱动器上的通用 `coordination.properties` 文件。 有关 `coordination.properties` 文件的更多信息，请参阅 MFT coordination.properties 文件。

command.properties
MFT command.properties 文件指定发出命令时要连接的命令队列管理器以及 Managed File Transfer 联系该队列管理器所需的信息。 有关 `command.properties` 文件的更多信息，请参阅 MFT command.properties 文件。

agent.properties
每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。 `agent.properties` 文件还可以包含更改代理行为的属性。 有关 `agent.properties` 文件的更多信息，请参阅 MFT `agent.properties` 文件。

logger.properties
`logger.properties` 文件指定记录器的配置属性。 有关 `logger.properties` 文件的更多信息，请参阅 MFT 记录器配置属性。

属性文件和代码页

由于 Java 的限制，所有 Managed File Transfer 属性文件的内容都必须保留为美式英语。 如果您在一个非美式英语系统中编辑属性文件，那么必须使用 Unicode 换码序列。

z/OS z/OS 上的 MFT 配置选项

z/OS 上的 Managed File Transfer 配置选项与分布式平台的选项相同。

有关多平台上的配置选项的更多信息，请参阅第 590 页的『多平台上的 MFT 配置选项』。

在 z/OS 上，配置位置由环境变量 BFG_DATA 定义。 如果 BFG_DATA 引用的 z/OS UNIX System Services 目录下尚不存在配置，那么 MFT 命令 PDSE 库数据集的 BFGCUSTM JCL 脚本将生成创建配置所需的作业。 然后会在您运行这些生成的作业时创建配置。 配置创建依赖于引用现有可访问目录的 BFG_DATA。

您还可以使用多平台和 z/OS 上提供的相同 **fte** 命令来创建和维护配置。 有关 **fte** 命令的列表，请参阅 MFT 命令 。

Windows Linux 下载和配置 Redistributable Managed File Transfer components

Redistributable Managed File Transfer package 提供了 Redistributable Managed File Transfer Agent，您可以将其配置为连接到现有 IBM MQ 基础结构，并使用户能够传输文件而无需安装 IBM MQ。 对于来自 IBM MQ 9.2.1 的 Continuous Delivery，可再分发软件包还包含 Redistributable Managed File Transfer Logger。

开始之前

有关 Redistributable Managed File Transfer Agent 和 Redistributable Managed File Transfer Logger 的可再分发许可条款的信息，请参阅 IBM MQ 可再分发组件。

Redistributable Managed File Transfer package 组件提供了 Managed File Transfer 的功能，但存在以下异常:

- 对于 Redistributable Managed File Transfer Agent，不支持与协调，命令和代理队列管理器的绑定方式连接，您必须使用客户机方式连接。 发出命令时，必须提供在使用作为 IBM MQ 的一部分安装的 Managed File Transfer 时可选的参数: 队列管理器主机，端口，名称和通道名称。
- V 9.2.1 Redistributable Managed File Transfer Logger 仅支持 FILE 类型记录器，仅以客户机方式连接到协调队列管理器。 不支持与数据库记录器的协调队列管理器的客户机方式连接。 如果需要绑定方式连接，那么必须使用 IBM MQ 的标准安装。
- LTS 对于 Long Term Support，不包含以下命令:
  - **fteCreateCDAgent.cmd**
  - **fteCreateLogger.cmd**
  - **fteDeleteLogger.cmd**
  - **fteMigrateLogger.cmd**
  - **fteSetLoggerTraceLevel.cmd**
  - **fteShowLoggerDetails.cmd**
  - **fteStartLogger.cmd**
  - **fteStopLogger.cmd**

  有关可用命令的完整列表，请参阅已安装的 MFT 命令集。
- V 9.2.1 对于 Continuous Delivery from IBM MQ 9.2.1，不包含以下命令:
  - **fteCreateCDAgent.cmd**

  有关可用命令的完整列表，请参阅已安装的 MFT 命令集。
- 不支持 Managed File Transfer Connect:Direct。
- 不包括 IBM MQ Explorer。

Windows 必须在系统上安装可从 Microsoft 获取的 Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 库才能使用 Redistributable Managed File Transfer Agent。 请参阅 The latest supported Visual C++ downloads。

V 9.2.1 对于来自 IBM MQ 9.2.1 的 Continuous Delivery， Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 库对于 Redistributable Managed File Transfer Logger 也是必需的。

注: Redistributable Managed File Transfer package 不支持 Advanced Message Security 。

关于此任务

您可以选择下载 Redistributable Managed File Transfer package 并将 Redistributable Managed File Transfer Agent 配置为连接到现有 IBM MQ 基础架构，以允许用户在其本地环境与现有 IBM MQ 基础架构之间传输文件，而无需用户安装 IBM MQ 以获取 Managed File Transfer 功能。

V 9.2.1 从 IBM MQ 9.2.1 开始， Redistributable Managed File Transfer package 还包含 Redistributable Managed File Transfer Logger，这使您能够设置文件记录器以客户机方式连接到协调队列管理器。

过程

1. 从 Fix Central 下载 IBM MQ 可再分发的 Managed File Transfer 代理程序包 。
   a) 选择适用于您操作系统的软件包。
   归档或 .zip 文件名描述文件内容和等效的维护级别。 文件名采用以下格式:
   - Windows *V.R.M.F*-IBM-MQFA-Redist-Win64
   - Linux *V.R.M.F*-IBM-MQFA-Redist-LinuxX64

- Linux *V.R.M.F*-IBM-MQFA-Redist-LinuxS390X
- Linux *V.R.M.F*-IBM-MQFA-Redist-LinuxPPC64LE

其中 *V.R.M.F* 是版本号，例如 9.2.0.0 或 9.2.1.0。

b) 确定您要在其中将软件包解压缩的目录，例如：

- Windows C:\MFTZ
- Linux /home/MFTZ

2. 将所下载软件包的内容解压缩：

- Windows 在 Windows 上，使用 Windows Explorer 工具进行抽取。
- Linux 在 Linux 上，抽取并解压缩，如下所示：

```
gunzip V.R.M.F-IBM-MQFA-Redist-LinuxX64.tar.gz
```

然后，

```
tar xvf V.R.M.F-IBM-MQFA-Redist-LinuxX64.tar
```

其中 *V.R.M.F* 是版本号，例如 9.2.0.0 或 9.2.1.0。
创建以下目录：

- Windows Linux bin：包含所有必需的 MFT 命令
- Windows bin64：包含 Windows 64 位操作系统支持所需的库
- Windows Linux java：包含 IBM JRE 和 IBM MQ 库
- Windows Linux licenses：包含许可证文件
- Windows V 9.2.4 META-INF：包含具有代码签名信息的文件
- Windows Linux mqft：包含 Ant 支持和核心 MFT 功能支持所需的 ant 和 lib 目录
- Windows Linux swtag：包含 swidtag 文件，许可证管理器需要此文件来识别计算机上的安装

下一步做什么

您已准备好配置 Managed File Transfer Agent。 有关后续步骤，请参阅 第 593 页的『为 Redistributable Managed File Transfer Agent 创建初始配置』。

V 9.2.1 从 IBM MQ 9.2.1 开始，您还可以配置 Managed File Transfer Logger。 有关配置记录器的后续步骤，请参阅 第 595 页的『为 Redistributable Managed File Transfer Logger 创建初始配置』。

相关参考

配置 Redistributable Managed File Transfer components 时可能出现的错误

Windows Linux 为 Redistributable Managed File Transfer Agent 创建初始配置

您可以配置 Managed File Transfer Agent 以连接到现有 IBM MQ 配置。

开始之前

确保您下载并解压缩 Redistributable Managed File Transfer Agent 软件包的内容。 有关更多信息，请参阅 第 591 页的『下载和配置 Redistributable Managed File Transfer components』。

关于此任务

首先创建 Redistributable Managed File Transfer Agent 需要的环境。 然后，可以设置与 IBM MQ 服务器上运行的队列管理器的连接，然后在启动和验证代理之前配置代理和代理队列管理器。

V 9.2.1 从 IBM MQ 9.2.1 开始，您创建的环境将与 Redistributable Managed File Transfer Logger 共享。 有关更多信息，请参阅第 595 页的『为 Redistributable Managed File Transfer Logger 创建初始配置』。

过程

1. 为 Redistributable Managed File Transfer Agent 创建环境。

   运行 **fteCreateEnvironment** 命令时，将创建包含 MFT 代理程序的配置信息的 MFT 数据目录。 确保您位于解压缩下载的 Redistributable Managed File Transfer Agent 组件时创建的 bin 目录中。 运行以下命令：

   - Windows

     ```
     fteCreateEnvironment.cmd -d datapath location
     ```

   - Linux

     ```
     . ./fteCreateEnvironment -d datapath location
     ```

   此命令采用以下可选参数：

   -d
   此参数指定创建，存储和维护 MFT 配置的数据路径的位置。 如果在未指定数据位置的情况下运行 **fteCreateEnvironment**，那么将在解压缩 Redistributable Managed File Transfer Agent 的位置中创建目录 mftdata 。

   注: 如果可再分发的代理程序将作为 Windows 服务运行，那么需要在系统环境中设置 **BFG_DATA** 环境变量才能使该服务正常工作。

   V 9.2.0 **-n 安装名称**
   此参数用于指定 IBM MQ 安装的名称或唯一名称。

   您可能希望使用此参数的情况示例如下：

   - 如果要快速测试新功能或功能部件，请将可再分发包与现有配置配合使用，其中已将代理程序配置为仅以客户机方式连接到队列管理器。 (请注意，此参数不适用于配置为以绑定方式连接到队列管理器的任何代理程序。)
   - 如果要从标准 Managed File Transfer 安装迁移到 Redistributable Managed File Transfer Agent 软件包，并且要使用与标准安装所创建的配置相同的配置。 这是已安装标准 Managed File Transfer 但正在连接到另一台机器上运行的代理队列管理器的情况。

   缺省安装名称变量为 **BFG_INSTALLATION_NAME**。

   有关 **fteCreateEnvironment** 命令的更多信息，请参阅 fteCreateEnvironment (为 Redistributable Managed File Transfer Agent 设置环境)。

   您还可以设置环境变量 *BFG_DATA* 以及数据路径位置：

   ```
   BFG_DATA=Datapath location
   ```

   在创建、启动和停止代理或者其他任何命令之前，您必须确保 *BFG_DATA* 变量设置为正确的数据路径位置。

2. 设置 IBM MQ 连接。

   a) 使用 **fteSetupCoordination** 命令来设置协调队列管理器。

fteSetupCoordination 命令创建协调队列管理器所需的设置以及进一步配置所需的目录。 Redistributable Managed File Transfer Agent 在客户机方式下工作，因此您必须为此命令提供额外的参数以避免错误，因为不支持绑定方式。

```
fteSetupCoordination -coordinationQMgr PRMFTDEMO2
              -coordinationQMgrHost 9.121.59.233 -coordinationQMgrPort 3002
              -coordinationQMgrChannel SYSTEM.DEF.SVRCONN
```

有关使用 **fteSetupCoordination** 命令的更多详细信息和步骤，请参阅 fteSetup 协调。 有关如何配置协调队列管理器的信息，请参阅第 629 页的『为 MFT 配置协调队列管理器』。

b) 创建并设置命令队列管理器：

```
fteSetupCommands -p PRMFTDEMO2 -connectionQMgrHost 9.121.59.233
              -connectionQMgrPort 3002 -connectionQMgrChannel SYSTEM.DEF.SVRCONN
              -connectionQMgr PRMFTDEMO2 -f
```

有关使用 **fteSetupCommands** 命令的更多详细信息和步骤，请参阅 fteSetupCommands: create MFT command.properties 文件。

3. 为端点创建 MFT 代理程序定义。

```
fteCreateAgent -p PRMFTDEMO2 -agentQMgrHost 9.121.59.233
            -agentQMgrPort 3002 -agentQMgrChannel SYSTEM.DEF.SVRCONN
            -agentName AGENT.TRI.BANK -agentQMgr PRMFTDEMO2 -f
```

有关使用 **fteCreateAgent** 命令来配置代理和代理队列管理器的更多信息，请参阅 fteCreateAgent。

注: 您需要使用在命令输出中显示的 MQSC 命令来定义代理队列管理器上的代理对象，否则步骤 第 595 页的『4』 中的指示信息将不起作用。

在每个代理的步骤 第 594 页的『2』 和 第 595 页的『3』 中，创建有关代理队列管理器的队列和主题定义。

4. 启动代理，然后随时可以开始传输文件。

```
fteStartAgent -p PRMFTDEMO2 AGENT.TRI.BANK
```

您可以通过运行以下命令来验证代理的状态：

```
fteListAgents
```

有关使用 **fteListAgents** 命令的更多详细信息，请参阅 fteList 代理。

下一步做什么

V 9.2.1 如果要配置 Redistributable Managed File Transfer Logger，请完成 第 595 页的『为 Redistributable Managed File Transfer Logger 创建初始配置』中的步骤。

相关概念

第 589 页的『配置 Managed File Transfer』
安装完成以后，您可以配置 Managed File Transfer 的功能部件。

第 590 页的『多平台上的 MFT 配置选项』
Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。 这些属性文件位于安装产品时所定义的配置目录中。

相关参考

fteCreateTransfer：启动新的文件传输

V 9.2.1 Windows Linux 为 Redistributable Managed File Transfer Logger 创建初始配置

您可以配置 FILE 类型 Managed File Transfer Logger 以在客户机方式下连接到协调队列管理器。

开始之前

确保您下载并解压缩 Redistributable Managed File Transfer Agent 软件包的内容。 从 IBM MQ 9.2.1 开始，此软件包还包含 Redistributable Managed File Transfer Logger。 有关更多信息，请参阅第 591 页的『下载和配置 Redistributable Managed File Transfer components』。

关于此任务

Redistributable Managed File Transfer Agent 和 Redistributable Managed File Transfer Logger 共享同一环境。 创建此环境并设置 IBM MQ 连接后，可以创建并启动记录器。

过程

1. 确保已创建 Redistributable Managed File Transfer Agent 和 Redistributable Managed File Transfer Logger 的共享环境，如步骤 第 594 页的『1』 中所述，并且已按 第 593 页的『为 Redistributable Managed File Transfer Agent 创建初始配置』的步骤 第 594 页的『2』 中所述设置了 IBM MQ 连接。
   .
2. 使用 **fteCreateLogger** 命令创建文件记录器。
   例如：

   ```
   fteCreateLogger FILELOGGER -loggerType FILE -loggerQMgr PRMFTDEMO2
   -loggerQMgrHost 9.121.59.233 -loggerQMgrPort 3003 -loggerQMgrChannel SYSTEM.DEF.SVRCONN
   -fileSize 20MB -fileCount 10 -fileLoggerMode CIRCULAR
   ```

   有关使用 **fteCreateLogger** 命令的更多信息，请参阅 fteCreateLogger。
3. 使用 **fteStartLogger** 命令启动记录器。
   有关 **fteStartLogger** 命令的更多信息，请参阅 fteStartLogger。

相关概念

第 589 页的『配置 Managed File Transfer』
安装完成以后，您可以配置 Managed File Transfer 的功能部件。

第 590 页的『多平台上的 MFT 配置选项』
Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。 这些属性文件位于安装产品时所定义的配置目录中。

Windows Linux 升级 Redistributable Managed File Transfer components

您可以通过下载新的 Redistributable Managed File Transfer package 来升级 Redistributable Managed File Transfer components 。

开始之前

有关 Redistributable Managed File Transfer Agent 和 Redistributable Managed File Transfer Logger 的可再分发许可条款的信息，请参阅 IBM MQ 可再分发组件。

注: Redistributable Managed File Transfer package 不支持 Advanced Message Security 。

关于此任务

如果已安装 Redistributable Managed File Transfer components，那么可以通过下载新的可再分发软件包并将内容解压缩到同一位置来对其进行升级。

过程

1. 从 Fix Central 下载适用于您操作系统的 IBM MQ 可再分发 Managed File Transfer 代理程序包 。
2. 停止所有 Managed File Transfer 代理程序 ， 记录器 等待任何正在运行的 Managed File Transfer 命令完成。

3. 通过将下载的新可再分发软件包的内容解压缩到已安装 Redistributable Managed File Transfer components 的目录中，更新 Redistributable Managed File Transfer components 的现有安装的文件。

z/OS 创建 MFT 代理或记录器命令数据集

您可以通过 Managed File Transfer 命令模板数据集，为特定协调的特定 Managed File Transfer Agent 或 Managed File Transfer Logger 创建 PDSE 命令数据集。

关于此任务

完成以下步骤：

过程

1. 生成 MFT 命令模板 PDSE 库数据集 V 9.2.0 SCSQFCMD 的副本。

   SCSQFCMD 必须复制到新库中，例如 *prefix.agent*.JCL。 您可以使用更新版本的 SCSQFCMD(BFGCOPY) 成员，并进行以下替换：

   - 将 *++supplied-library++* 替换为 SCSQFCMD PDSE 的标准名称。
   - z/OS 将 *++service-library++* 替换为新 MFT 命令 PDSE 库数据集的标准名称。 *++service-library++* 是创建的代理或记录器服务的输出数据集。

2. 对于新的 MFT 命令 PDSE 库数据集，请编辑成员 BFGCUSTM，该成员是用于为代理或记录器定制命令的 JCL 脚本。 以 *++variable name++* 格式指定每个变量，必须将变量替换为其必需值。 要获取各种 JCL 变量的描述，请参阅 第 607 页的『z/OS JCL 变量』。 BFGSTDIN DD 语句可以定义三种类型的变量：Variables、Properties 和 Environment。 该语句格式如下：

   ```
   [Variables]
   variable1=value1
   variable2=value2
   ....
   variableN=valueN
   [Properties]
   property1=property value1
   property2=property value2
   ...
   propertyN=property valueN
   [Environment]
   custom_variable1=value1
   custom_variable2=value2
   ....
   custom_variableN=valueN
   ```

   Variables 定义每个命令需要的设置和环境变量集。

   Properties 为 MFT 配置属性定义覆盖项。 您可以根据需要添加代理和记录器属性，为您的环境定制代理或记录器。 要获取所有属性的列表，请参阅第 617 页的『配置属性文件』。 提供此工具是为了保存必须访问 MFT 配置属性文件 (保留为 z/OS UNIX System Services 文件)。

   Environment 定义了所需的其他所有定制环境变量。

3. 为新的 MFT 命令 PDSE 库数据集提交作业 BFGCUSTM。 此作业将生成 JCL 命令集，作为 PDSE 的新成员，适合于代理或记录器。 要获取完整的命令列表，请参阅 第 611 页的『z/OS 代理和记录器命令 JCL 脚本』。

   作业 BFGCUSTM 将更新包含 JCL（包含带 DISP=OLD 的 DD 语句）的库。 您必须在提交后退出编辑器，才能允许执行作业。

   检查输出作业日志，以查看 JCL 脚本是否已成功运行。 如果有任何故障，请纠正并重新提交 BFGCUSTM 作业。

   BFGCUSTM JCL 脚本还会根据需要更新 z/OS UNIX System Services MFT 配置属性文件，以将这些文件保留在步骤中。 如果 CoordinationQMgr 属性定义的配置不存在，那么将输出警告消息，并且您必须运行生成的 BFGCFCR 和 BFGCMCR 作业才能创建配置属性文件。 必须针对代理运行 BFGAGCR，针对记录器

编辑运行 BFGLGCRS。 如果指定的配置已存在，那么将使用 BFTCUSTM JCL 脚本中定义的任何属性更新此配置。

相关概念

第 591 页的『z/OS 上的 MFT 配置选项』
z/OS 上的 Managed File Transfer 配置选项与分布式平台的选项相同。

相关任务

第 607 页的『在 z/OS 上更新现有 MFT 代理程序或记录器命令数据集』
您可以更新通过 Managed File Transfer 命令模板数据集创建的 Managed File Transfer 命令 PDSE 库数据集。

z/OS 配置 Managed File Transfer for z/OS

Managed File Transfer for z/OS 需要定制以使组件能够正确运行。

关于此任务

您需要：

1. 编辑 PDSE 成员以指定配置数据
2. 定义协调队列管理器
3. 定义命令队列管理器
4. 配置一个或多个代理
5. 可选：配置记录器任务以将数据存储在 Db2 中

在以下主题中详细描述了所需执行任务的顺序。

相关概念

第 598 页的『查看 MFT 配置』
在开始之前，您需要查看系统的配置。

相关任务

安装 IBM MQ Advanced for z/OS

z/OS 查看 MFT 配置

在开始之前，您需要查看系统的配置。

对于已定义的每种 MFT 配置，Managed File Transfer (MFT) 需要一个或多个队列管理器充当以下角色：

- 协调队列管理器，用于维护配置中各个代理状态的相关信息，这些信息发布在协调程序的相关主题中。
- 一个或多个命令或连接队列管理器，用作 MFT 命令的 IBM MQ 网络入口点。
- 一个或多个代理队列管理器，用于在 MFT 代理与 IBM MQ 网络之间提供通信。

以上每个角色都可以由单独的队列管理器执行；也可以组合角色，这样，在最简单的配置中，所有角色由同一个队列管理器执行。

如果要在现有 MFT 环境中添加 z/OS 队列管理器，您需要定义 z/OS 队列管理器与此配置中的其他队列管理器之间的连接。 可以使用手动定义的传输队列或使用集群来实现此目的。

每个 MFT 代理都会与一个队列管理器通信。 如果有多个代理与同一个队列管理器通信，那么代理队列管理器将为每个代理定义多个队列：

- SYSTEM.FTE.COMMAND.*agent_name*
- SYSTEM.FTE.DATA.*agent_name*
- SYSTEM.FTE.REPLY.*agent_name*
- SYSTEM.FTE.STATE.*agent_name*
- SYSTEM.FTE.EVENT.*agent_name*

- SYSTEM.FTE.AUTHAGT1.*agent_name*
- SYSTEM.FTE.AUTHTRN1.*agent_name*
- SYSTEM.FTE.AUTHOPS1.*agent_name*
- SYSTEM.FTE.AUTHSCH1.*agent_name*
- SYSTEM.FTE.AUTHMON1.*agent_name*
- SYSTEM.FTE.AUTHADM1.*agent_name*

请注意，您可以定义要在其中使用诸如 SYSTEM.FTE.COMMAND.* 之类的概要文件的常规安全概要文件，也可以为每个代理定义特定的概要文件。

相关概念

第 599 页的『在启动前』

Managed File Transfer (MFT) 配置使用 z/OS UNIX System Services (z/OS UNIX) 和 PDSE 数据集中的文件。

z/OS 在启动前

Managed File Transfer (MFT) 配置使用 z/OS UNIX System Services (z/OS UNIX) 和 PDSE 数据集中的文件。

大多数配置和操作是使用 PDSE 中的 JCL 完成的，您需要熟悉如何在 z/OS UNIX 环境中工作。

您可以通过 ISPF 访问 OMVS，也可以在工作站上通过命令来使用 Telnet 类型会话（例如 Telnet Putty 或 SSH）。

如果通过 ISPF 使用 OMVS，那么可以使用标准 ISPF 编辑器以及浏览命令 **oedit** 和 **obrowse**。

您需要熟悉以下 z/OS UNIX 命令

表 *41:* 公共 *z/OS UNIX* 命令

命令	函数
chmod xxx path	更改文件访问许可权。
df -k path	报告文件系统中剩余的可用空间。 -k 报告可用空间 (KB)。
du -kt path	报告路径下目录的大小。 报告的大小 (以 KB 为单位)。
find path -name xxx	在路径目录中搜索名为 xxxx 的文件。 xxx 区分大小写，可以类似于 *zzz。
ls -ltrd directory	列示指定目录（而不是目录中的文件）的信息。
ls -ltr path	列出有关路径中文件的信息。
obrowse filename	浏览文件名。
oedit filename	在 OMVS 中编辑文件。

查看下表中的各项，并用您企业的相应条目来填写该表。 在编辑成员 BFGCUSTM 时需要这些值。

表 *42:* 成员 *BFGCUSTM* 所需的参数

名称	示例数据	注释
ADMIN_JOB1		作业卡。 使用相同的 JCL 卡生成所有作业。
armELEMENT	如果正在使用 ARM，那么对该代理或记录器使用 ARM 策略中指定的 ARM ELEMENT 值。 如果未在使用 ARM，那么将此参数设置为空白；例如，armELEMENT=	

表 42: 成员 *BFGCUSTM* 所需的参数 (继续)		
名称	**示例数据**	**注释**
armELEMTYPE	如果正在使用 ARM，那么使用 ARM 策略中指定的 ARM ELEMTYPE。 例如，armELEMTYPE=SYSBFGAG（针对代理）或 armELEMTYPE=SYSBFGLG（针对记录器）。 如果未在使用 ARM，那么将此参数设置为空白；例如 armELEMTYPE=	
BFG_DATA		根据需要填写
BFG_GROUP_NAME	MQM	
BFG_JAVA_HOME	`/java/java71_bit64_GA/J7.1_64/`	
BFG_JVM_PROPERTIES		根据需要填写
BFG_PROD	V 9.2.0 `/mqm/V9R2M0/mqft`	V 9.2.0 IBM MQ for z/OS UNIX System Services Components 目录下 `mqft` 目录的完整路径。
BFG_WTO	YES	获取系统日志中的 MFT 消息。
CLEAN_AGENT_PROPS	-trs	此参数指定在运行 BFGAGCL 成员时将用于清除代理的选项。 有关此参数的有效值的更多信息，请参阅 fteCleanAgent: 清除 MFT Agent。
coordinationQMgr	MQPV	必需配置
CREDENTIAL_PATH		在迁移中使用
Db2_ HLQ	SYS2.Db2。V10	
DB_PROPS_PATH		在迁移中使用
FTE_CONFIG		在迁移中使用
JOBCARD1		这是长时间运行的任务、代理和记录器的作业卡。
LIBRARY	SCEN.FTE.JCL	MFT PDSE 的名称。 您需要每个代理或记录器任务的副本。
MQ_HLQ	IBM MQ 数据集的高级限定符。 例如： `MQM.V920`	
MQ_LANG	E	
MQ_PATH	`/mqm/V9R2M0`	V 9.2.0 IBM MQ for z/OS UNIX System Services Components 安装的完整目录路径。
名称	AGENT1	
OUTPUT_CLASS	*	
PATH	`bin:/usr/bin:/usr/sbin`	

表 42: 成员 *BFGCUSTM* 所需的参数 (继续)		
名称	**示例数据**	**注释**
productId	V 9.2.0 ADVANCEDVUE	此参数用于设置要为其记录 Managed File Transfer 使用情况的产品类型。 有关此参数的有效值的信息，请参阅 fteSetProductId: set z/OS SCRT 记录产品标识。
QMGR	MQPV	
SERVICE_TYPE	AGENT 或 LOGGER	
TMPDIR	`/tmp`	临时文件的可读和可写 z/OS UNIX 路径。

此外，还必须查看以下变量并根据需要提供相应的值：

- coordinationQMgrHost=
- coordinationQMgrPort=
- coordinationQMgrChannel=
- connectionQMgr=
- connectionQMgrHost=
- connectionQMgrPort=
- connectionQMgrChannel=

这些属性是 AGENT 或 LOGGER 的常用属性。

注: 对于客户机连接，Host、Port 和 Channel 是必需的，但对于本地机器上的绑定连接，这三项应留空。

相关概念

第 601 页的『要检查的项』
确保您具有足够的磁盘空间，以及用于存储数据的目录，并且存在必需文件。

第 603 页的『编辑成员 BFGCUSTM』
在运行作业之前，必须编辑成员 BFGCUSTM，然后输入您的企业所使用参数的值。

z/OS 要检查的项

确保您具有足够的磁盘空间，以及用于存储数据的目录，并且存在必需文件。

检查您是否具有足够的磁盘空间

检查在将用于存储配置特定文件的文件系统上是否具有足够的可用磁盘空间。

如果已启用代理跟踪，那么缺省情况下会使用 100 MB 的磁盘空间。

配置文件本身很小，只有数 KB。

如果您计划使用两个代理和一个记录器，那么至少需要 300 MB。 您可以使用命令 **df -k** `path`，其中 `path` 是安装特定文件的位置。 这会提供可用空间和总空间 (KB)。

300 MB 为 307200 KB，因此，至少应预留 310000 KB。

创建并检查用于存储 Managed File Transfer 数据的目录

您需要一个用于存储 Managed File Transfer (MFT) 数据的目录。

检查您在文件系统 `df -k /var` 中是否具有足够的空间。 该文件系统至少应具有 310000 KB 可用空间。

如果尚未创建此文件系统，请使用 **mkdir** 命令；例如 **mkdir /var/mft**。

使用命令 **ls -ltrd /var/mft** 显示用户对此目录具有的许可权。

如果所有者或组不正确，请使用命令 **chown owner:group /var/mft**。

如果组的许可权不正确，请使用以下命令为所有者和组提供读、写和执行许可权。 请注意，以下命令还将为所有用户提供读和执行许可权：**chmod 775 /var/mft**。

检查文件是否存在以及您是否具有这些文件的访问权

对您将在定制期间使用的的文件使用 **ls -ltr** 命令。 例如：

```
ls -ltrd /java/java71_bit64_GA/J7.1_64/bin
```

提供

```
drwxr-xr-x  4 SYSTASK TSOUSER   8192 Nov 15 2013 /java/java71_bit64_GA/J7.1_64/bin
```

其中，drwxr-xr-x 含义如下：

d
这是目录。

rwx
所有者 *SYSTASK* 具有目录的读、写和执行访问权。

r-x
组 *TSOUSER* 中的人员可以读和执行目录中的文件。

r-x
通用访问，即，任何人都可以读或执行目录中的文件。

检查下表中指定的文件：

表 *43:* 用户需要的特定文件访问权

路径	**用户执行配置时需要的访问权**
BFG_JAVA_HOME	读和执行
/tmp	读和写
BFG_PROD	读
BFG_DATA	写
MQ_PATH	读

相关概念

第 599 页的『在启动前』
Managed File Transfer (MFT) 配置使用 z/OS UNIX System Services (z/OS UNIX) 和 PDSE 数据集中的文件。

第 602 页的『z/OS 配置的公共 MFT』
不同的 Managed File Transfer 配置概述

z/OS z/OS 配置的公共 MFT

不同的 Managed File Transfer 配置概述

Managed File Transfer 使用附加到队列管理器的代理来传输数据。

MFT 可以使用多个队列管理器：

- 一个或多个用于传输数据的队列管理器。
- 一个用于发出请求的命令队列管理器。 例如，启动传输的请求将发送到此队列管理器，并且关联的命令将路由到 MFT 代理。
- 一个用于管理工作的协调队列管理器。

有三个公共 Managed File Transfer (MFT) 配置：

1. 具有一个或多个使用本地连接的代理的单个队列管理器。 这可能用于将数据集的内容放入 IBM MQ 队列中。
2. 在使用客户机绑定的分布式机器上具有 MFT 客户机的单个队列管理器。
3. 通过通道连接的两个队列管理器，每台机器上一个或多个代理。 这些代理可以是客户机或本地绑定。

请注意以下几点：

1. MFT 以 Java 编写，并使用一些 shell 脚本和 JCL 来配置和操作 MFT。
2. Db2 状态和活动可以记录并存储到 Db2 表中。
3. MFT 配置人员必须熟悉 z/OS UNIX System Services (z/OS UNIX)。例如：
   - 具有名称类似于 `/u/userID/myfile.txt2` 的文件的目录结构
   - z/OS UNIX 命令，如：

     cd（更改目录）
     ls（列示）
     chmod（更改文件许可权）
     chown（更改可以访问文件或目录的文件所有权或组）
4. 在 z/OS UNIX 中需要以下产品才能够配置和运行 MFT：
   - Java；例如，`/java/java71_bit64_GA/J7.1_64/`
   - IBM MQ V920，例如 `/mqm/V9R2M0`。
   - Db2 JDBC 库（如果您要将 Db2 用于状态和历史记录）；例如 `/db2/db2v12/jdbc/lib`

您需要协调队列管理器。 但是，可以使用相同的队列管理器来运行代理、处理命令以及进行协调。 如果正在使用多个队列管理器，那么必须挑选一个充当协调程序。

检查您的 IBM MQ 连接

如果您具有现有 MFT 协调程序队列管理器，那么在执行配置的队列管理器与协调和命令队列管理器之间需要连接。

z/OS 复制 SCSQFCMD 以创建 JCL 库

您需要为每个代理程序和记录器创建 JCL 库。 JCL 包含用于创建和运行代理程序或记录器的配置和作业。

对于每个代理程序和记录器，通过编辑并运行 BFGCOPY 成员来创建 IBM 提供的 V 9.2.0 SCSQFCMD 库的副本。

该库用于定义代理或记录器的配置，并且在定制后，包含可用于创建所需 Managed File Transfer 配置以及代理或记录器的作业。

在此过程中创建成员 BFGCUSTM。

注: 如果您熟悉 z/OS UNIX 命令，那么可以使用在其他平台上使用的相同命令来配置 z/OS 。

相关概念

第 602 页的『z/OS 配置的公共 MFT』
不同的 Managed File Transfer 配置概述

第 603 页的『编辑成员 BFGCUSTM』
在运行作业之前，必须编辑成员 BFGCUSTM，然后输入您的企业所使用参数的值。

z/OS 编辑成员 BFGCUSTM

在运行作业之前，必须编辑成员 BFGCUSTM，然后输入您的企业所使用参数的值。

请参阅成员 BFGCUSTM 所需的参数，以获取需要特定值的参数的列表。

此外，还必须查看以下变量并根据需要提供相应的值：

- coordinationQMgrHost=
- coordinationQMgrPort=
- coordinationQMgrChannel=
- connectionQMgr=
- connectionQMgrHost=
- connectionQMgrPort=
- connectionQMgrChannel=

这些属性是 AGENT 或 LOGGER 的常用属性。

注: 对于客户机连接，Host、Port 和 Channel 是必需的，但对于本地机器上的绑定连接，这三项应留空。

如果这是 Managed File Transfer 环境中的第一个队列管理器，并且您希望对协调、命令和正在运行的代理使用相同的队列管理器，请将值设置为本地队列管理器名称。

```
coordinationQMgr=MQPV
connectionQMgr=MQPV
```

其中，MQPV 是您的本地队列管理器名称。

提交将更新 PDSE 的作业，并在指定路径下创建目录结构。

请注意，该作业要求独占使用，因此在运行该作业时，您必须停止使用 PSDE。

提示: 无论何时提交作业 BFGCUSTM，该作业都将替换所有 JCL 文件。 您应该重命名更改的每个成员。

相关概念

第 599 页的『在启动前』
Managed File Transfer (MFT) 配置使用 z/OS UNIX System Services (z/OS UNIX) 和 PDSE 数据集中的文件。

第 606 页的『创建代理』
您需要复制 PDSE 以生成特定于代理的 PDSE，例如 *user.MFT.AGENT1*。 从上一个代理或记录器配置（如果存在）中复制 PDSE。 如果这是您的第一个配置，请复制 MFT 随附的 PDSE。

z/OS 定义协调队列管理器

Managed File Transfer 要求创建的队列管理器可充当协调队列管理器。

根据您选择的配置，该队列管理器将位于本地 MVS 系统或另一台机器上。 在前一种情况下，指向它的连接是绑定连接，在后一种情况下，则为客户机连接。

成功运行配置步骤之后，PDSE 中将存在已配置的成员。

成员 BFGCFCR 定义协调队列管理器，并且此作业：

1. 在 Managed File Transfer (MFT) 目录中创建目录结构，并创建配置文件。
2. 运行 CSQUTIL 以定义 IBM MQ 资源。

如果协调队列管理器位于远程机器上，那么该作业步骤将失败。

成员 BCFCFCR 在 z/OS UNIX System Services 中创建文件并创建 MQ 定义。 此作业将：

1. 创建 MFT 主题，
2. 创建 MFT 队列
3. 将 *NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)* 更改为 *NAMES(SYSTEM.BROKER.DEFAULT.STREAM, SYSTEM.BROKER.ADMIN.STREAM, SYSTEM.FTE)*
4. 执行 *ALTER QMGR PSMODE(ENABLED)*

   在执行更改前发出 *DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)* 命令。 如果您的 NAMLIST 不是缺省值，那么应更改名称列表以将 SYSTEM.FTE 添加到名称列表。

使用您自己的前缀重命名成员 BCFCFCR（例如 CCPCFCR），因为重新定制此文件会对其进行替换。

通过插入凭证文件的名称，编辑此已重命名的成员。 例如：

```
%BFGCMD CMD=fteSetupCoordination +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
```

保存并提交此作业。 请注意，如果需要重新提交此作业，必须添加 *-f* 选项。

运行此作业时，将列出创建的 IBM MQ 资源。 您需要保护这些资源。

```
DEFINE TOPIC('SYSTEM.FTE') TOPICSTR('SYSTEM.FTE') REPLACE
ALTER TOPIC('SYSTEM.FTE') NPMSGDLV(ALLAVAIL) PMSGDLV(ALLAVAIL)
DEFINE QLOCAL(SYSTEM.FTE) LIKE(SYSTEM.BROKER.DEFAULT.STREAM) REPLACE
ALTER QLOCAL(SYSTEM.FTE) DESCR('Stream for MFT Pub/Sub interface')
* Altering namelist: SYSTEM.QPUBSUB.QUEUE.NAMELIST
* Value prior to alteration:
DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)
ALTER NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST) +
NAMES(SYSTEM.BROKER.DEFAULT.STREAM+
,SYSTEM.BROKER.ADMIN.STREAM,SYSTEM.FTE)
* Altering PSMODE. Value prior to alteration:
DISPLAY QMGR PSMODE
ALTER QMGR PSMODE(ENABLED)
```

相关任务

第 605 页的『定义命令队列管理器』
可以使用同一个队列管理器作为协调队列管理器和命令队列管理器，或者创建新的命令队列管理器。

z/OS 定义命令队列管理器

可以使用同一个队列管理器作为协调队列管理器和命令队列管理器，或者创建新的命令队列管理器。

关于此任务

您必须具有命令队列管理器，但可以使用同一个队列管理器作为协调队列管理器和命令队列管理器。 否则，您需要创建新的命令队列管理器。 该命令队列管理器可以与协调队列管理器位于相同的机器上，但不是必须这样做。

过程

1. 使用您自己的前缀重命名成员 BFGCMCR，例如 CCPCMCR。
   您必须重命名 BFGCMCR，因为重新定制此文件时会替换此文件。
2. 通过插入凭证文件的名称，编辑重命名的成员。
   例如：

   ```
   %BFGCMD CMD=fteSetupCommands +
   -credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
   ```

3. 保存并提交此作业。
   请注意，如果需要重新提交此作业，必须添加 *-f* 选项。

   此队列管理器将用于诸如 **ftePingAgent** 之类的命令。
4. 查看此成员，进行提交，然后查看输出。

下一步做什么

请参阅第 606 页的『创建代理』，以获取有关如何创建代理的信息。

相关概念

第 604 页的『定义协调队列管理器』
Managed File Transfer 要求创建的队列管理器可充当协调队列管理器。

相关任务

配置 MQMFTCredentials.xml

相关参考

MFT 凭证文件格式

z/OS 创建代理

您需要复制 PDSE 以生成特定于代理的 PDSE，例如 *user.MFT.AGENT1*。 从上一个代理或记录器配置（如果存在）中复制 PDSE。 如果这是您的第一个配置，请复制 MFT 随附的 PDSE。

查看成员 BFGCUSTM，如果需要使用其他凭证文件，请创建一个。

很多内容仍与第 603 页的『编辑成员 BFGCUSTM』中详细描述的定制中的内容相同。

您需要：

- 更改 //SYSEXEC DD DSN=SCEN.FTE.JCL.AGENT1
- 更改 LIBRARY 以便与代理 PDSE 相匹配
- 更改 SERVICE_TYPE=AGENT
- 将 NAME 更改为代理（匹配 PDSE）JOBCARD 的名称
- 更改 BFG_JVM_PROPERTIES="-Xmx1024M"

提交此作业，请记住，作业需要数据集的独占访问权。

代理的作业全都具有 *BFGAG** 格式的名称。

重命名成员 *BFGAGCR*。 此作业将更新 Managed File Transfer 目录中的文件，并使用 CSQUTIL 在本地队列管理器中创建特定于代理的队列。 指定您的凭证文件的名称，例如 `-credentialsFile //'SCEN.FTE.JCL.VB(CREDOLD)`。 如果未指定该名称，那么用于启动代理的作业不会使用凭证文件。

检查输出以确保进程已成功运行。

提示: 将 *agent.properties* 文件的路径名从作业的输出复制到代理的 PDSE 中的成员。

例如，将 `/u/userid/fte/wmqmft/mqft/config/MQPA/agents/AGENT1/agent.properties` 复制到成员 AGENT 中。

如果需要显示属性文件并且添加行 `/u/userid/fte/wmqmft/mqft/logs/MQPA/agents/AGENT1/logs`，这将很有用。

这是跟踪文件的存储位置。

相关概念

第 604 页的『定义协调队列管理器』
Managed File Transfer 要求创建的队列管理器可充当协调队列管理器。

第 606 页的『使用代理』
如何使用各种命令来确保代理正确工作。

相关任务

第 605 页的『定义命令队列管理器』
可以使用同一个队列管理器作为协调队列管理器和命令队列管理器，或者创建新的命令队列管理器。

z/OS 使用代理

如何使用各种命令来确保代理正确工作。

启动代理

重命名成员 BFGAGST，查看该成员并提交作业。

如果有效，您将收到消息 BFGAG0059I：代理已成功启动。

显示活动的代理

重命名成员 BFGAGLI，查看该成员并提交使用协调队列管理器的作业。

您必须解决任何连接问题。

Ping 通代理以检查其是否在工作

重命名成员 BFGAGPI，查看该成员并提交使用命令队列管理器的作业。

您必须解决任何连接问题。

执行测试传输

请参阅第 612 页的『执行验证传输』，以获取进一步的信息。

停止代理

重命名成员 BFGAGSP，查看该成员并提交作业。

使用成员 BFGAGST 重新启动代理。

相关概念

第 606 页的『创建代理』

您需要复制 PDSE 以生成特定于代理的 PDSE，例如 *user.MFT.AGENT1*。 从上一个代理或记录器配置（如果存在）中复制 PDSE。 如果这是您的第一个配置，请复制 MFT 随附的 PDSE。

z/OS 在 z/OS 上更新现有 MFT 代理程序或记录器命令数据集

您可以更新通过 Managed File Transfer 命令模板数据集创建的 Managed File Transfer 命令 PDSE 库数据集。

过程

1. 编辑 BFGCUSTM JCL 脚本成员，并更新 BFGSTDIN DD 语句中的变量和属性。

   如果要除去先前定义的属性，请将其值设置为空，而不是除去条目。 运行 BFGCUSTM JCL 脚本时，将应用指定的属性作为对实际代理和记录器 z/OS UNIX System Services 属性文件的更新; 将属性设置为空白值表示将除去该属性
2. 提交作业 BFGCUSTM。 此作业将重新生成 JCL 命令集，适合于代理或记录器。 要获取完整的命令列表，请参阅 第 611 页的『z/OS 代理和记录器命令 JCL 脚本』。 检查输出作业日志，以查看 JCL 脚本是否已成功运行。 如果有任何故障，请纠正并重新提交 BFGCUSTM 作业。

结果

您可以修改生成的 JCL 脚本，并添加自己的逻辑。 但是，在重新运行 BFGCUSTM 时应慎重，因为您可能会覆盖定制逻辑。

相关概念

第 591 页的『z/OS 上的 MFT 配置选项』

z/OS 上的 Managed File Transfer 配置选项与分布式平台的选项相同。

相关任务

第 597 页的『创建 MFT 代理或记录器命令数据集』

您可以通过 Managed File Transfer 命令模板数据集，为特定协调的特定 Managed File Transfer Agent 或 Managed File Transfer Logger 创建 PDSE 命令数据集。

z/OS *z/OS JCL* 变量

您可以在 BFGCUSTM 脚本中使用替换值、JCL 变量和配置属性。

下表列出了针对 MFT 命令 PDSE 库数据集中 BFGCUSTM JCL 脚本的替换值。 在提交 BFGCUSTM 作业之前，必须将这些替换值替换为适当的值。

表 *44:* 针对 *BFGCUSTM JCL* 脚本的替换值

替换变量	值
++library++	包含 MFT 命令 PDSE 库的数据集名称。

表 *44:* 针对 *BFGCUSTM JCL* 脚本的替换值 *(*继续*)*	
替换变量	**值**
++bfg_java_home++	Java 安装的位置。
V 9.2.0 + + mq_path + +	IBM MQ for z/OS UNIX System Services Components 目录的路径。 例如，`/mqm/V9R2M0`。 这用于提供 MFT 安装的完整路径，例如 `/mqm/V9R2M0/mqft`。

下表描述了针对 MFT 命令 PDSE 库数据集中（在 [Variables] 部分中）BFGCUSTM JCL 脚本的 BFGSTDIN DD 语句的环境变量。 在提交 BFGCUSTM 作业之前，必须将使用替换值（即用两个加号 ++ 括起来的值）指定的所有变量替换为适当的值。

表 *45:* 环境变量	
环境变量	**值**
LIBRARY	包含 MFT 命令 PDSE 库的数据集名称。
TMPDIR	临时文件的 z/OS UNIX System Services 目录。
BFG_PROD	IBM MQ for z/OS UNIX System Services Components 目录下 `mqft` 目录的完整路径; 例如: `/mqm/V9R2M0/mqft`。
BFG_DATA	Managed File Transfer for z/OS 的数据目录的位置，即指向 *DATA_DIR* 的路径。
BFG_JAVA_HOME	Java 安装的位置。
BFG_JVM_PROPERTIES	可选。 设置 BFG_JVM_PROPERTIES 环境变量的值。 这些属性将传递至 Java 虚拟机。

表 *45:* 环境变量 *(*继续*)*

环境变量	**值**
BFG_GROUP_NAME	mqm 文件组通常与 MFT 配置数据文件和命令关联。 因此，属于 mqm 组成员的所有用户都可以访问和更改 MFT 配置。 有关更多信息，请参阅 IBM MQ 中 MFT 的文件系统许可权。 对于 z/OS 系统，文件组是 z/OS UNIX System Services (z/OS UNIX) 文件系统实体，而不必定义 mqm 文件组。 您可以使用 BFG_GROUP_NAME 环境变量来关联 MFT 配置数据文件的 z/OS UNIX 文件系统组。 例如，在 z/OS UNIX shell 提示符处使用: `export BFG_GROUP_NAME=FTEGB` 定义要与当前 z/OS UNIX 会话的任何后续创建的配置文件相关联的组 *FTEGB* 。 您可以将 BFG_GROUP_NAME 设置为空值或将其移除。 **注:** 当首次运行 BFGCUSTM 时，如果 MFT 配置要由多个用户标识使用，那么需要将 BFG_GROUP_NAME 设置为所有必需用户标识可访问的组。 如果再次运行 BFGCUSTM ，那么不得更改 BFG_GROUP_NAME (否则，还必须更改 BFG_DATA 引用的目录中所有文件和目录的 z/OS UNIX 组文件许可权以反映新的 BFG_GROUP_NAME 设置)。 如果在将 BFG_GROUP_NAME 环境变量设置为非空白值的 z/OS 系统上运行 **fteMigrateAgent** 命令，那么该命令将检查用户是否是由 BFG_GROUP_NAME 变量指定的组的成员。 如果用户不在指定的组中，那么该命令可能会报告错误消息 BFGCL0502E: **您无权执行所请求的操作。** 无法运行。 有关用户成功运行该命令必须满足的条件的详细信息，请参阅 fteMigrateAgent。
BFG_WTO	将 BFG_WTO 设置为 YES、ON 或 TRUE 时，将会启用 z/OS 日志记录。 这可控制是否将写入到代理事件日志的消息也写入到 z/OS 操作员日志工具，从而在通过 JCL 运行代理时可以更轻松地访问自动化产品。 路由代码是 Programmer Information (11)，描述符代码为 Informational (12)。
SERVICE_TYPE	指定 MFT 命令库是用于代理还是记录器。 有效值为 AGENT 或 LOGGER。
名称	针对 SERVICE_TYPE 值的代理或记录器的名称。
QMGR	与针对 SERVICE_TYPE 值的代理或记录器关联的本地队列管理器的名称。
OUTPUT_CLASS	SYSOUT 数据集的输出类。 缺省值为 *，这将请求与作业语句中 MSGCLASS 参数相同的输出类。
MQ_PATH	IBM MQ for z/OS UNIX Components 目录的路径。
MQ_HLQ	IBM MQ 数据集的高级限定符。
MQ_LANG	需要的语言。

表 *45:* 环境变量 *(继续)*	
环境变量	**值**
DB2_HLQ	可选。 Db2 数据集的高级限定符。
JOBCARD1	JCL 命令作业的标题行 1。
JOBCARD2	JCL 命令作业的标题行 2。
JOBCARD3	JCL 命令作业的标题行 3。
ADMIN_JOB1	admin 作业的标题行 1。
ADMIN_JOB2	admin 作业的标题行 2。
ADMIN_JOB3	admin 作业的标题行 3。
FTE_CONFIG	针对迁移的现有 WMQFTE 配置。 如果无需迁移，请将其设置为空值。
CREDENTIAL_PATH	用于迁移的凭证文件的路径，例如 `/u/user1/agent3`。 用于 Managed File Transfer 迁移的凭证文件必须位于到 IBM WebSphere MQ File Transfer Edition 7.0.4.4 上的配置信息和配置文件的单独文件中。 该凭证文件仅对迁移命令 **BFGAGMG** 和 **BFGLGMG** JCL 脚本是必需的。 如果无需迁移，请将其设置为空值。 另请注意 IBM 文件传输版（FTE）不再受支持。 WebSphereMQ 文件传输版（FTE）已不再受支持。 要从 FTE 迁移至 IBM MQ 中的 Managed File Transfer 组件，请参阅迁移 Managed File Transfer。
DB_PROPS_PATH	为迁移指定数据库记录器属性文件。 仅当属性文件不使用以下缺省名称和路径时，才需要此选项：`config_directory/coordination_qmgr/databaselogger.properties`。 如果无需迁移，请将其设置为空值。

下表描述了针对 MFT 命令 PDSE 库数据集中 BFGCUSTM JCL 脚本的 BFGSTDIN DD 语句的必需 MFT 配置属性。 在提交 BFGCUSTM 作业之前，必须将使用替换值（即用两个加号 ++ 括起来的值）指定的属性替换为适当的非空值。 这些属性将为 MFT 配置属性定义覆盖项。 您可以添加代理和记录器属性，为您的环境定制代理或记录器。 要获取所有属性的列表，请参阅第 617 页的『配置属性文件』。

表 *46:* 针对 *BFGSTDIN DD* 语句的必需配置属性	
属性	**值**
coordinationQMgr	代理或记录器所关联配置的协调队列管理器的名称。
coordinationQMgrHost	可选。 正在运行协调队列管理器的系统的主机名。 如果将该属性的值留空，那么将采用绑定方式连接。
coordinationQMgrPort	可选。 协调队列管理器正在侦听的端口号。 只有当同时为 coordinationQMgrHost 属性指定了非空值时，才能使用该参数。
coordinationQMgrChannel	可选。 用于连接到协调队列管理器的通道。 只有当同时为 coordinationQMgrHost 属性指定了非空值时，才能使用该参数。
connectionQMgr	代理或记录器所关联配置的命令队列管理器的名称。
connectionQMgrHost	可选。 正在运行命令队列管理器的系统的主机名。 如果将该属性的值留空，那么将采用绑定方式连接。

表 46: 针对 *BFGSTDIN DD* 语句的必需配置属性 *(继续)*	
属性	**值**
connectionQMgrPort	可选。 命令队列管理器正在侦听的端口号。 只有当同时为 connectionQMgrHost 属性指定了非空值时，才能使用该参数。
connectionQMgrChannel	可选。 用于连接到命令队列管理器的通道。 只有当同时为 connectionQMgrHost 属性指定了非空值时，才能使用该参数。

z/OS *z/OS* 代理和记录器命令 *JCL* 脚本

MFT 命令 PDSE 库数据集中可用的 JCL 命令集。

表 47: 可以在 *MFT* 命令 *PDSE* 库数据集中找到 *JCL* 命令	
成员	**描述或 fte 命令行命令**
BFGCOPY	用于创建此库副本的作业
BFGCUSTM	用于为代理或记录器定制此库的作业
BFGZCFCR	fteSetupCoordination
BFGZCMCR	fteSetup 命令: 创建 MFT command.properties 文件
BFGZAGCR	fteCreateAgent。 仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGLGCRS	fteCreateLogger。 仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGZAGST	fteStartAgent。 仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGAGSTP	**fteStartAgent** 过程。 仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGZAGPI	ftePingAgent。 仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGZAGSP	fteStopAgent。 仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGZLGST	fteStartLogger。 仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGLGSTP	**fteStartLogger** 过程。 仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGZLGSP	fteStopLogger。 仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGZAGSH	fteShowAgentDetails. 仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGZLGSH	fteShowLoggerDetails. 仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGZCFDF	fteChangeDefaultConfigurationOptions
BFGZAGCL	fteCleanAgent。 仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。

表 47: 可以在 *MFT* 命令 *PDSE* 库数据集中找到 *JCL* 命令 (继续)	
成员	**描述或 fte 命令行命令**
BFGZAGDE	fteDeleteAgent。 仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGZLGDE	fteDeleteLogger。 仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGZPRSH	fteDisplayVersion
BFGZAGLI	fteListAgents。 仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGZMNLI	fteListMonitors
BFGZSTLI	fteListScheduledTransfers
BFGZTMLI	fteListTemplates
LTS BFGZAGMG 第 612 页的『1』	**fteMigrateAgent**。 仅当将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
LTS BFGZLGMG 第 612 页的『1』	**fteMigrateLogger**。 仅当将 SERVICE_TYPE 变量设置为 LOGGER 时创建。
BFGXCROB	**fteObfuscate** 样本
BFGZRAS	**fteRAS**
BFGZAGTC	fteSetAgentTraceLevel。 仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGZLGTC	fteSetLoggerTrace 级别。 仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGXPRAN	**fteAnt** 样本
BFGXTRCA	**fteCancelTransfer** 样本
BFGXMNCR	**fteCreateMonitor** 样本
BFGXTMCR	**fteCreateTemplate** 样本
BFGXTRCR	**fteCreateTransfer** 样本
BFGXMNDE	**fteDeleteMonitor** 样本
BFGXSTDE	**fteDeleteScheduledTransfer** 样本
BFGXTMDE	**fteDeleteTemplate** 样本

注意:

1. V 9.2.1 对于 Continuous Delivery，将在 IBM MQ 9.2.1 除去 **fteMigrateAgent**, **fteMigrateConfigurationOptions** 和 **fteMigrateLogger** 命令。

   LTS 这些命令仍可用于 IBM MQ 9.2 Long Term Support。

- 对于创建 MQSC 或引用删除脚本的命令，JCL 将要求您运行一个脚本，但该脚本已由作业运行。
- BFGZRAS 在 BGCUSTM 作业运行时创建 BFGRAS 成员。

z/OS 执行验证传输

如何执行传输以检查产品是否在正确工作。

重命名并编辑成员 BFGTRCRS。

1. 在 %BFGCMD CMD=fteCreateTransfer -h 前添加 /*。
2. 除去成员中的其他注释。
3. 为 -sa 和 -da 指定当前代理名称
4. 保存 JCL
5. 提交 JCL

此 JCL 会连接到命令队列管理器。

z/OS 配置记录任务

必须在与协调队列管理器相同的映像上运行记录任务。 您可以记录到 Db2 中。

创建记录任务

复制 PDSE 以生成特定于记录器的 PDSE。 例如，user.MFT.LOGGER。

如果需要使用其他凭证文件，请创建一个。 有关更多信息，请参阅 在 z/OS 上配置 MQMFTCredentials.xml。

查看成员 BFGCUSTM。 请注意，大多数内容仍与先前定制中的内容相同。

但是，您需要：

- 更改 //SYSEXEC DD DSN=SCEN.FTE.JCL....
- 更改 LIBRARY 以便与代理 PDSE 相匹配
- 将 QMGR 更改为协调队列管理器的名称
- 设置 SERVICE_TYPE=LOGGER
- 将 NAME 更改为记录器的名称（匹配 PDSE）
- 查看 JOBCARD 并更改作业名，使该名称与代理的作业名不同。
- 审查 BFG_JVM_PROPERTIES="-Xmx1024M"

如果正在使用 Db2 记录器，那么创建一个文件很有用，这样您就可以捕获 Db2 跟踪以帮助识别 Db2 问题。

在 JVM 属性中指定文件的名称，其中 JDBC 跟踪属性文件具有类似如下的内容：

```
db2.jcc.traceDirectory=/u/johndoe/fte
db2.jcc.traceFile=jccTrace1
db2.jcc.traceFileAppend=false
# turn on all traces
# db2.jcc.traceLevel=-1
# turn off all traces
db2.jcc.traceLevel=0
```

设置两个 JVM 属性

```
BFG_JVM_PROPERTIES=-Ddb2.jcc.propertiesFile=/u/.../sql.properties
-Ddb2.jcc.ssid=DBCA
```

其中，/u/.../sql.properties 是 Db2 跟踪属性文件的名称，*DBCA* 是 Db2 子系统的名称。

提交此作业，请注意，作业需要数据集的独占访问权。 代理的作业全部具有诸如 *BFGLG** 之类的名称。

记录到文件

有关记录到 Db2 的更多信息，请参阅第 614 页的『记录到 Db2 时创建记录任务』

重命名成员 BFGLGCRS。 此作业将更新 Managed File Transfer (MFT) 目录中的文件，并使用 CSQUTIL 在本地队列管理器中创建特定于代理的队列。

原始文件具有命令 %BFGCMD CMD=fteCreateLogger -h，用于列示命令语法。

要通过在语句前放置 /* 来创建注释掉 %BFGCMD CMD=fteCreateLogger -h 的记录器任务，请确保第一列为空。

从第二列中移除注释，然后配置语句。 例如：

```
%BFGCMD CMD=fteCreateLogger   +
-p MQPH     +
-loggerQMgr MQPH      +
-loggerType FILE        +
-fileLoggerMode circular    +
-fileSize 5MB +
-fileCount 5 +
-p MQPH +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
LOGGER
```

检查输出以查看是否已成功处理。

提示: 将 logger.properties 文件的路径名从作业的输出复制到代理的 PDSE 中的成员。

例如，复制到成员 APATH 中

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties
```

如果需要显示属性文件，那么此操作将很有用。

添加此文件的目录：

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/
```

如果要记录到文件，那么日志文件将存储在此目录中，例如 LOGGER0-20140522123654897.log。

跟踪文件位于日志子目录中，例如

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

您现在可以启动记录任务。

记录到 Db2 时创建记录任务

重命名成员 BFGLGCRS。

此作业将更新 MFT 目录中的文件，并使用 CSQUTIL 在本地队列管理器中创建特定于代理的队列。

您需要知道：

表 *48: Db2* 变量

Db2 名称	**示例**
-dbName databaseName	您可以从 Db2 子系统的消息 DSNL004I 中的位置值获取此值
- dbDriver filePath	例如/db2/db2v10/jdbc/classes/db2jcc.jar
-dbLib filePath	例如/db2/db2v10/jdbc/lib/libdb2jcct2zos_64.so

编辑文件。 原始文件具有命令 %BFGCMD CMD=fteCreateLogger -h，用于列示命令语法。

从第二列中移除注释，然后配置语句。 例如

```
%BFGCMD CMD=fteCreateLogger   +
-p MQPH     +
-loggerQMgr MQPH      +
-loggerType DATABASE      +
-dbType DB2           +
-dbName DSNDBCP         +
-dbDriver /db2/db2v10/jdbc/classes/db2jcc.jar  +
-dbLib /db2/db2v10/jdbc/lib/             +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
LOGGER
```

要通过在语句前放置 /* 来创建注释掉 %BFGCMD CMD=fteCreateLogger -h 的记录器任务，请确保第一列为空。

提交作业，并检查输出以查看是否已成功处理。

提示: 将 logger.properties 文件的路径名从作业的输出复制到代理的 PDSE 中的成员。

例如，复制到成员 APATH 中：

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties into member USS
```

如果需要显示属性文件，那么此操作将很有用。

跟踪文件位于日志子目录中，例如：

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

创建 Db2 表

您需要创建 Db2 表。 这些定义位于 z/OS UNIX System Services 文件 mqft/sql/ftelog_tables_zos.sql 中。

在您的 PDSE 中创建成员 Db2。 编辑此成员，并在命令行上使用 COPY 命令。 从 z/OS UNIX System Services 定义文件复制。

由于特定于站点的需求可能有很大差异，因此此文件仅指定表的基本结构以及它们将位于的表空间。

SQL 脚本指定了表空间，以确保它是使用具有足以容纳最大表行的页大小的缓冲池创建的。 请注意，未指定诸如 LOB 位置之类的属性。

您的数据库管理员可能需要修改该文件的副本，以定义这些与性能相关的属性。

该文件还将采用缺省模式名称 FTELOG、缺省表空间名称 FTELOGTS 和数据库名称 FTELOGDB。 您可以遵循文件开头的注释中所描述的过程，根据需要更改这些名称，以便与现有的数据库和任何本地命名约定相匹配。

要点: 使用诸如 **SPUFI** 之类的在线工具来运行命令，因为文件中存在注释，而诸如 **DSNTINAD** 之类的批处理程序不接受注释。

启动记录器任务

重命名、查看和提交成员 BFGLGST。您应该会收到消息 BFGDB0023I：记录器已完成启动活动，现在正在运行。

记录器操作

要显示记录器状态，请重命名、查看并提交成员 BFGLGSH。

要停止记录器，请重命名、查看并提交成员 BFGLGSP。

z/OS 上 MFT 的环境变量

如果直接从 z/OS UNIX System Services (z/OS UNIX) 环境或您自己的 JCL 脚本运行命令，那么在定制和配置之后，必须先设置多个环境变量，然后才能运行 Managed File Transfer 提供的配置和管理脚本。 必须在将调用脚本的每个环境中为每个用户设置这些变量。

为避免与其他产品冲突，您可以选择在主目录中创建 .wmqfterc 脚本。 然后，每个 Managed File Transfer 脚本都将调用 .wmqfterc 脚本，并且您可以使用此脚本为 Managed File Transfer 提供定制环境设置。

同时还有一个可选的环境变量 BFG_WTO，通过 JCL 运行代理时，您可以设置该环境变量以将消息发送到操作程序日志。

表 49: 必需 z/OS 环境变量	
环境变量	**值**
BFG_JAVA_HOME	Java 安装的位置。 有关受支持的 Java 级别的更多信息，请参阅 IBM MQ 的系统需求。
BFG_DATA	Managed File Transfer for z/OS 的数据目录的位置。这是指向 *DATA_DIR* 的路径。
STEPLIB	必须包含以下 IBM MQ 数据集： • SCSQAUTH • SCSQANLE • SCSQLOAD 如果要在 z/OS 系统上运行数据库记录器组件，那么 STEPLIB 还必须按所示顺序包含以下 Db2 数据集： • SDSNEXIT • SDSNLOD2 • SDSNLOAD

以下是正确配置 Managed File Transfer 的环境变量的示例 `.profile`：

V 9.2.0

```
STEPLIB=MQM.V920.SCSQAUTH:MQM.V920.SCSQANLE:MQM.V920.SCSQLOAD
PATH=/u/fteuser/bin:/u/fteuser/J7.0/bin:/bin:/usr/bin:/u/fteuser/extras/bin:/bin:$PATH
BFG_JAVA_HOME=/u/fteuser/J7.0
BFG_DATA=/u/fteuser/DATA_DIR
export PATH STEPLIB BFG_JAVA_HOME BFG_DATA
```

注意: V 9.2.0 从 z/OS UNIX 环境中调用 **fte*** 命令时不再需要 LIBPATH 环境变量，应该从任何现有 `.wmqfterc` 脚本中除去该变量

此外，您还可以设置以下环境变量：

表 *50:* 可选 *z/OS* 环境变量

环境变量	值
BFG_WTO	以下某个值将启用 BFG_WTO： • YES • ON • TRUE 以下某个值将禁用 BFG_WTO。 这些值不区分大小写。 • NULL • 否 • OFF • FALSE 启用 z/OS 日志记录。 缺省情况下，禁用该环境变量。 写入到代理事件日志的消息也会写入到 z/OS 操作程序日志设施，这在通过 JCL 运行代理时允许更轻松地访问自动化产品。 路由代码是 Programmer Information (11)，描述符代码为 Informational (12)。
BFG_GROUP_NAME	mqm 文件组通常与 Managed File Transfer 配置数据文件和命令关联。 因此，mqm 组中的所有成员用户都可以访问和更改 Managed File Transfer 配置。 有关更多信息，请参阅 IBM MQ 中 MFT 的文件系统许可权。 对于 z/OS 系统，文件组是 z/OS UNIX 文件系统实体，并且不一定定义 mqm 文件组。 您可以使用 BFG_GROUP_NAME 环境变量为 Managed File Transfer 配置数据文件定义备用的现有 z/OS UNIX 文件系统组。 例如，在 z/OS UNIX shell 提示符处： `export BFG_GROUP_NAME=FTEGB` 定义要与当前 z/OS UNIX 会话的任何后续创建的配置文件相关联的组 FTEGB。 您可以将 BFG_GROUP_NAME 设置为空值或将其移除。 如果在将 BFG_GROUP_NAME 环境变量设置为非空白值的 z/OS 系统上运行 **fteMigrateAgent** 命令，那么该命令将检查用户是否是由 BFG_GROUP_NAME 变量指定的组的成员。 如果用户不在指定的组中，那么该命令可能会报告错误消息 BFGCL0502E： **您无权执行所请求的操作。** 无法运行。 有关用户成功运行该命令必须满足的条件的详细信息，请参阅 fteMigrateAgent。

z/OS 配置属性文件

Managed File Transfer 中所使用属性的摘要。

- MFT coordination.properties 文件
- MFT command.properties 文件
- MFT agent.properties 文件
- 记录器配置属性文件

z/OS 为 z/OS 自动重新启动管理器 (ARM) 配置 MFT

Managed File Transfer 是启用了 ARM 的应用程序。

开始之前

有关启用 ARM 以及定义适用于您系统的 ARM 策略的更多信息，请参阅使用 z/OS 自动重新启动管理器 (ARM)。

如果要使用 MFT DB 记录器功能自动重新启动并重新连接到 Db2 数据库，那么 ARM 是唯一受支持的重新启动管理器。

关于此任务

使用 ARM 时，可以通过设置代理/记录器属性 armELEMTYPE 和 armELEMENT，对代理和记录器进行配置以进行重新启动。 属性 armELEMTYPE 定义了 ARM 元素的类型，而属性 armELEMENT 是 ARM 将要注册的元素的名称：

- 您可以将代理 ELEMTYPE 设置为 SYSBFGAG，将 armELEMENT 设置为与代理名称一致。
- 您可以将记录器 ELEMTYPE 设置为 SYSBFGLG，将 armELEMENT 设置为与记录器名称一致。

注: 由 ARM 配置为进行重新启动的代理和记录器只能从批处理作业或已启动的任务中成功运行。 尝试直接从 z/OS UNIX System Services 命令行启动代理程序或记录器将失败，并带有 ARM 错误原因码。

示例

以下重新启动策略的示例定义了依赖于队列管理器 FT7C 的代理 BFGFT7CAG1：

```
RESTART_ORDER
   LEVEL(3)
   ELEMENT_TYPE(SYSBFGAG,SYSBFGLG)

RESTART_GROUP(GROUP7C)
   ELEMENT(SYSMQMGRFT7C)
   ELEMENT(BFGFT7CAG1)
       RESTART_ATTEMPTS(3,300)
```

z/OS 示例: 在 z/OS 上为 Managed File Transfer 代理程序创建 JCL

使用此信息可生成可用于在 IBM MQ for z/OS 上创建和启动代理程序的一些 JCL。

复制样本库

执行以下过程：

1. 通过打开库，生成库 V 9.2.0 SCSQFCMD 的副本 (请参阅 第 603 页的『复制 SCSQFCMD 以创建 JCL 库』)。

   大多数成员 (以 BFGX， BFGY 或 BFGZ 开头的成员) 是稍后用于为代理程序生成定制 JCL 的模板。

   重要成员为 BFGCOPY。
2. 打开 BFGCOPY 并替换：

   + + 供应库 + +
   具有作为产品一部分安装的 SCSQFCMD 库的名称。

+ + 服务-库 + +
与要用于代理程序的库 (目标库) 的名称一起使用。

3. 提交作业，您有一个可以使用的新库。

编辑 BFGCUSTM

执行以下过程：

1. 打开新库，以便您可以编辑 BFGCUSTM 成员 (请参阅 第 603 页的『编辑成员 BFGCUSTM』)
2. 修改成员中包含在 ++ 字符中的所有参数，并将其替换为相应的值。 例如，将

   V 9.2.0 **+ + mq_path + +**
   z/OS UNIX System Services (z/OS UNIX) 组件目录的路径。 例如，`/mqm/V9R2M0`。

   注: 此变量有三个要替换的实例。

   + + bfg_data + +
   成为要存储 IBM MQ Managed File Transfer for z/OS 配置的 z/OS UNIX 目录。

   + + 服务类型 + +
   到单词 AGENT

   + + 代理程序名称 + +
   作为代理程序的名称

注意:

1. 某些条目 (例如， CLEAN_AGENT_PROPS 所需的 ++options++) 是不需要的，因此您应该除去这些条目。
2. 请参阅 第 599 页的『在启动前』，以获取 BFGCUSTM 成员中所有参数的完整列表以及它们应该具有的值的描述。

提交 BFGCUSTM JCL

执行以下过程：

1. 提交该作业。
2. 退出 ISPF 库。

   这是必需的，因为 BFGCUSTM 作业正在更新库，并且在库处于打开状态时无法执行此操作。
3. 当作业完成时，查看作业记录。

   您将看到许多消息，指示已在库中创建新成员。

   其中每个成员都包含可用于为代理程序执行特定任务的 JCL。 请参阅 第 611 页的『z/OS 代理和记录器命令 JCL 脚本』 以获取这些成员的列表以及它们对应的 IBM MQ Managed File Transfer 命令。

提交 BFGAGCR 以创建代理程序

新成员 BFGAGCR 包含一些通过调用 **fteCreateAgent** 命令 创建代理程序 的 JCL。

执行以下过程：

1. 打开成员 BFGAGCR。

   您应该会看到 BFGAGCR 已填充了您的名称:

   - 代理程序
   - 代理队列管理器
   - MFT 拓扑的协调队列管理器
2. 提交成员 BFGAGCR。

   当成员运行时，它将执行下列操作:

- 为代理程序创建必需的配置文件。
- 连接到代理队列管理器，并使用 CSQUTIL 创建代理需要的系统队列。
- 向协调队列管理器注册代理。

通过提交 BFGAGST 来启动代理程序

执行以下过程：

1. 提交 BFGAGST 成员。 请参阅 使用代理程序 以获取各种命令，这些命令显示代理程序正常工作。
2. 作业完成时，请检查作业记录是否包含以下消息：

```
BFGAG0058I: The agent has successfully initialized.
BFGAG0059I: The agent has been successfully started.
```

这意味着您的代理已启动，正在运行并准备好执行受管传输。

z/OS 将 MFT 代理程序移动到新的 z/OS LPAR

有时需要将 IBM MQ Managed File Transfer for z/OS 代理程序从一个 LPAR 移至另一个 LPAR ， 同时使该代理程序保持在具有相同协调和命令队列管理器的同一 IBM MQ Managed File Transfer 拓扑中。 执行此操作所需的步骤取决于最初如何创建要迁移的代理程序。

关于此任务

通过下列其中一种方式移动 IBM MQ Managed File Transfer for z/OS 代理程序：

- 如果代理程序最初是使用 V 9.2.0 SCSQFCMD 库的定制版本创建的，请使用该库在新的 LPAR 上重新创建该代理程序。
- 如果代理程序最初是通过运行 z/OS UNIX System Services (z/OS UNIX) 命令创建的，请使用这些命令在新的 LPAR 上重新创建该代理程序。

注:

调度的传输和传输模板存储在 IBM MQ Managed File Transfer 拓扑的协调队列管理器上。 此任务假定协调队列管理器不是移动工作的一部分。 在这种情况下，与要移动的代理相关联的任何已调度传输和传输模板在移动完成后仍保留在现有协调队列管理器上。

过程

- 移动使用定制版本的 SCSQFCMD 库创建的代理程序。

  如果使用定制版本的 SCSQFCMD 库创建了代理程序，那么可以使用该库在新 LPAR 上重新创建 IBM MQ Managed File Transfer for z/OS 环境和代理程序配置。 为此，请完成以下步骤：

  1. 将库的定制版本从原始 LPAR 复制到新 LPAR。
  2. 在新 LPAR 上的库的定制版本中编辑 BFGCUSTM 成员，并确保参数值仍然有效。
  3. 在新 LPAR 上运行 BFGCUSTM 成员，以创建配置环境和创建代理程序所需的所有 JCL。
  4. 运行 BFGCFCR 成员以定义要由新 LPAR 上的代理使用的协调队列管理器，并创建存储 IBM MQ Managed File Transfer 配置所需的目录结构。
  5. 接下来，运行 BFGCMCR 成员，以定义要由新 LPAR 上的代理使用的命令队列管理器。
  6. 运行 BFGAGCR 成员以重新创建代理及其配置。
  7. 请确保该代理所使用的系统队列存在于该代理的队列管理器上。

  如果要移动的代理程序具有与其关联的资源监视器，那么需要在新代理程序上重新创建这些监视器。 为此，请完成以下步骤：

  1. 在原始 LPAR 上，运行 BFGMNLI 成员以将与原始代理程序相关联的资源监视器的定义导出到 XML 文件。
  2. 将包含资源监视器定义的 XML 文件复制到新的 LPAR。

3. 使用新 LPAR 上 SCSQFCMD 库中的 BFGMNCRS 成员来导入存储在 XML 文件中的资源监视器定义。这将导致在新代理程序上创建监视器。

- 移动通过在 z/OS UNIX 中运行命令创建的代理程序。

  如果代理程序最初是通过运行 z/OS UNIX 命令创建的，那么可以使用命令在新的 LPAR 上重新创建代理程序。 为此，请完成以下步骤：

  1. 在新 LPAR 上运行 fteSetup 协调 命令，以定义要由代理使用的协调队列管理器，并创建存储 IBM MQ Managed File Transfer 配置所需的目录结构。
  2. 运行 fteSetupCommands 命令以定义要由新 LPAR 上的代理使用的命令队列管理器。
  3. 运行 fteCreateAgent 命令以重新创建代理及其配置。
  4. 请确保该代理所使用的系统队列存在于该代理的队列管理器上。

  如果要移动的代理程序具有与其关联的资源监视器，那么需要在新代理程序上重新创建这些监视器。 为此，请完成以下步骤：

  1. 在原始 LPAR 上，运行 fteListMonitors 命令并指定 **-ox** 参数，以将与原始代理关联的资源监视器的定义导出到 XML 文件。
  2. 将包含资源监视器定义的 XML 文件复制到新的 LPAR。
  3. 在指定 **-ix** 参数的新 LPAR 上运行 fteCreateMonitor 命令，以导入存储在 XML 文件中的资源监视器定义。 这将导致在新代理程序上创建监视器。

V 9.2.1 z/OS 使用 IBM MQ for z/OS 队列共享组规划 MFT 基础结构

如果您正在使用 IBM MQ Managed File Transfer (MFT)，那么当一个或多个代理程序，命令或协调队列管理器属于 IBM MQ for z/OS 队列共享组时，需要考虑以下事项。

请参阅 MFT 拓扑概述，以获取代理，命令队列管理器和协调队列管理器的描述。

代理队列管理器

通常， MFT 代理程序连接到单个代理程序队列管理器，并使用仅可由该队列管理器访问的本地队列。 通过在首次创建代理程序时向代理程序提供队列管理器名称，将通知代理程序要连接到哪个队列管理器。

通过 IBM MQ for z/OS，可以创建代理并将队列管理器名称替换为队列共享组 (QSG) 的名称。 这意味着代理可以连接到 QSG 中的任何可用队列管理器以执行文件传输。 如果代理当前连接到的队列管理器发生故障，那么代理会检测到故障并重新连接到 QSG 中的备用队列管理器。

结合 IBM MQ 9.2.0 中提供的高可用性代理程序支持，将代理程序连接到 QSG，请参阅 第 638 页的『Managed File Transfer 中的高可用性代理程序』，以允许创建非常强大的 MFT 拓扑。

例如，在下图中创建了 *Agent1*，以便其代理队列管理器是由两个队列管理器 *QM1* 和 *QM2* 组成的 QSG。 代理队列已定义为共享队列，存储在耦合设施中。

这意味着代理程序可以在 *LPAR 1* 或 *LPAR 2* 上运行，并连接到 *QM1* 或 *QM2*。 代理程序从中读取或写入的文件和数据集是共享的，这意味着可以从任一 LPAR 访问这些文件和数据集。

此外，该代理程序已配置为高可用性代理程序。 在图中，代理程序在 *LPAR 1* 中处于活动状态，并且代理程序的备用实例正在 *LPAR 2* 中运行。

此拓扑提供高弹性。 如果在 *LPAR 1* 上运行的代理程序发生故障，或者队列管理器 *QM1* 发生故障，或者 *LPAR 1* 发生故障，那么 *LPAR 2* 上的代理程序的备用实例可以从故障点接管并继续处理文件传输。

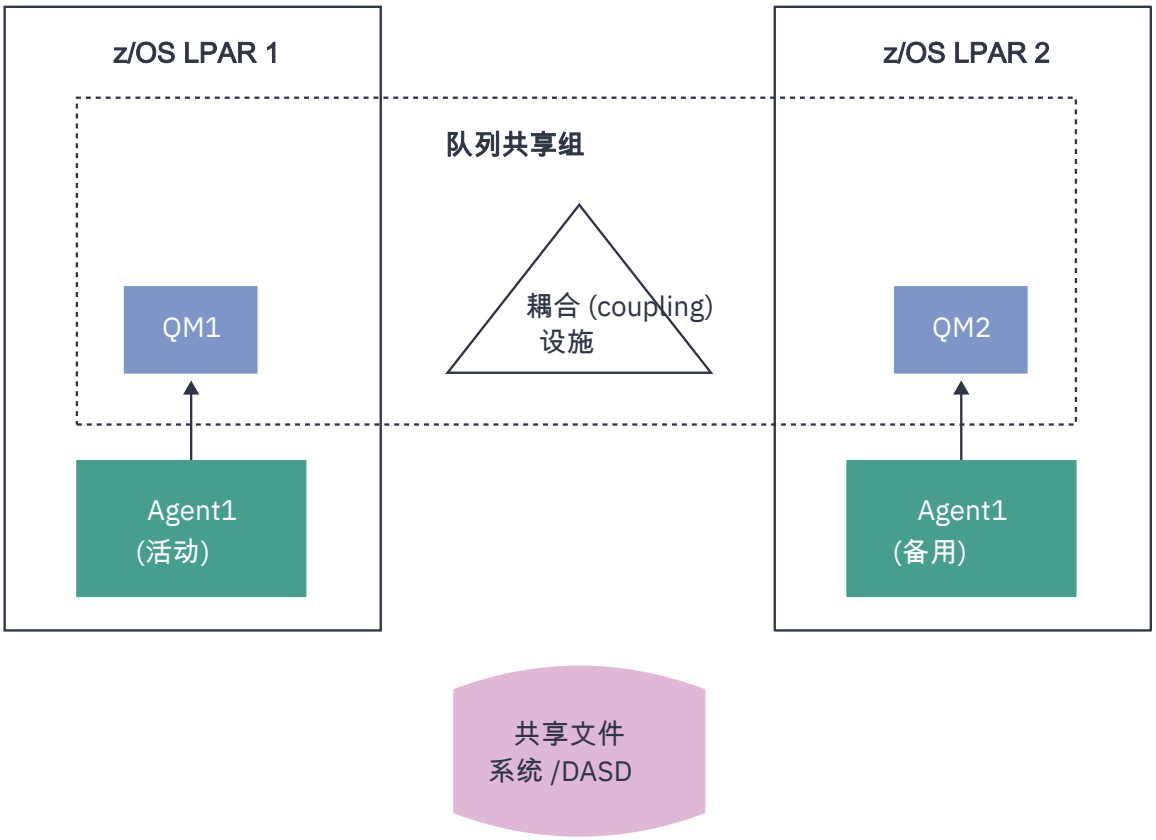


图 *98:* 使用队列共享组的高可用性 *MFT* 代理

创建使用 QSG 作为代理队列管理器的代理

使用 fteCreateAgent 命令创建代理。 执行此操作时，将为代理队列管理器提供队列共享组的名称。 例如：

```
fteCreateAgent -agentName Agent1 -agentQMgr QSG1
```

这将创建名为 *Agent1* 的代理程序，该代理程序使用作为其代理队列管理器的 QSG *QSG1* 成员的任何队列管理器。 在此配置中，代理程序使用跨内存 (绑定方式) 连接来连接到代理程序队列管理器，这意味着代理程序和队列管理器必须位于同一 LPAR 上。 这与上图 1 所示的示例完全相同。

运行 **fteCreateAgent** 命令时，它会生成一组 MQSC 命令，以在代理队列管理器上创建必需的队列。

当代理队列管理器是 QSG 时，需要修改这组命令，以便将每个队列创建为共享队列。 即，需要使用 QSGDISP (SHARED) 和 CFSTRUCT 属性提供的相应耦合设施结构来创建每个队列。

以下示例显示如何更改 MQSC 命令以创建 SYSTEM.FTE.COMMAND.AGENT1 队列作为共享队列。 缺省值的更改以粗体文本显示。

要点: 您需要对代理所使用的所有其他队列进行类似的更改。

```
DEFINE QLOCAL(SYSTEM.FTE.COMMAND.AGENT1) +
 QSGDISP(SHARED) +
 CFSTRUCT(MFTSTRUCT) +
 DEFPRTY(0) +
 DEFSOPT(SHARED) +
 GET(ENABLED) +
 INDXTYPE(CORRELID) +
 MAXDEPTH(5000) +
 MAXMSGL(4194304) +
 MSGDLVSQ(PRIORITY) +
 PUT(ENABLED) +
 RETINTVL(999999999) +
 SHARE +
```

```
NOTRIGGER +
USAGE(NORMAL) +
REPLACE
```

创建使用 QSG 作为代理队列管理器并作为客户机进行连接的代理

代理可以使用客户机通道连接到其代理队列管理器。 您可以使用此方法来允许代理程序在连接到 QSG 时在分布式平台上运行。 如果 QSG 中的所有队列管理器都获得了 IBM MQ Advanced for z/OS Value Unit Edition 的许可，那么代理程序还可以从没有本地队列管理器的 z/OS LPAR 连接到这些队列管理器。

下图中显示了此拓扑，此拓扑允许代理程序利用 QSGs 的弹性。 如果代理程序当前连接到的 QSG 中的队列管理器发生故障，那么代理程序将自动重新连接到 QSG 的其他成员并继续处理。

综合系统分发器用于在 QSG 中的可用队列管理器之间传播来自代理程序的连接。

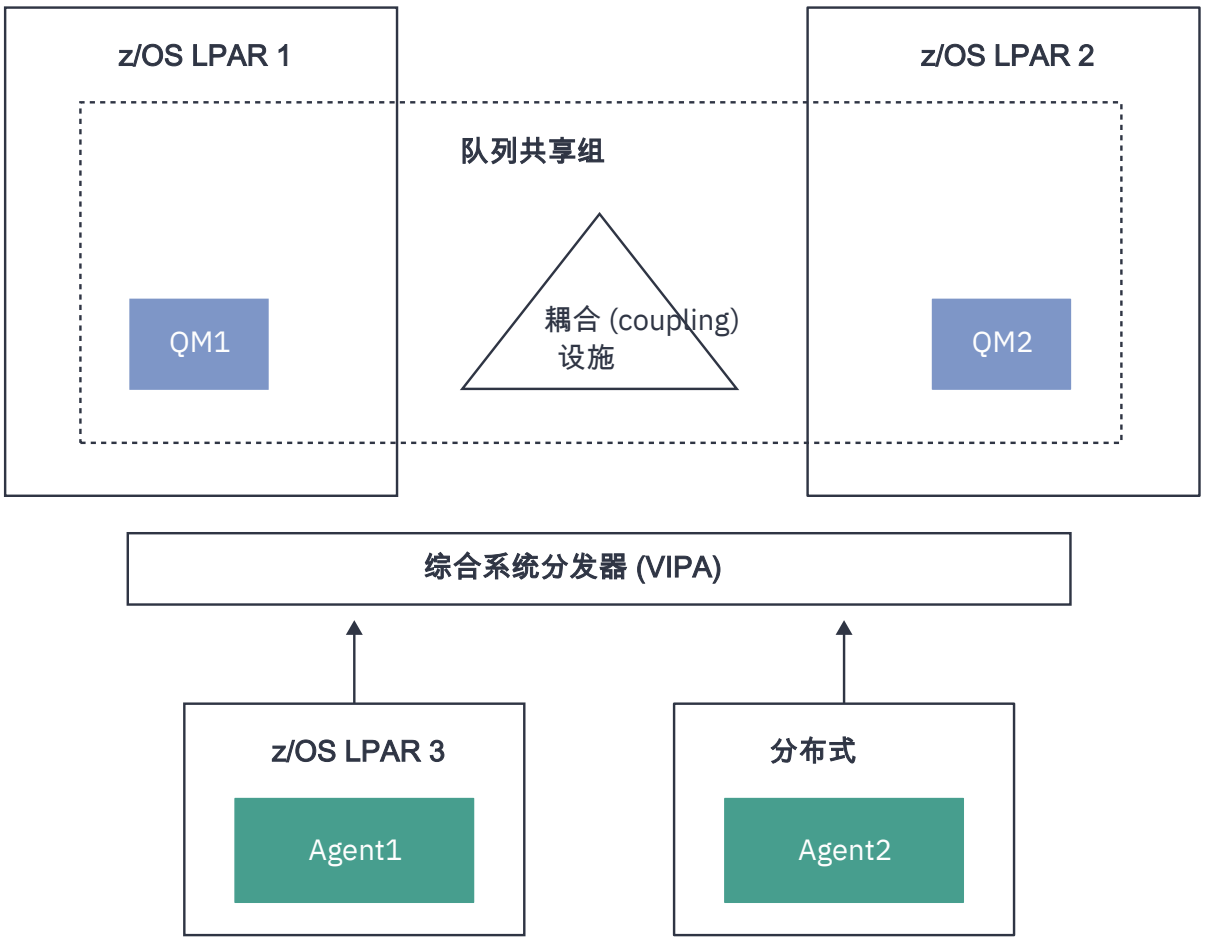


图 *99:* 作为客户机连接到队列共享组的 *MFT* 代理程序

为了使用此拓扑， QSG 中的队列管理器必须各自具有定义供代理使用的服务器连接通道。 请参阅第 52 页的『将客户机连接到队列共享组』以获取如何这样做的信息。

创建代理时，需要配置队列管理器，以便它们可以使用对 QSG 定义的通道，并通过综合系统分发器访问该通道。 例如：

```
fteCreateAgent -agentName Agent1 -agentQMgr QSG1 -agentQMgrHost vipaAddress
-agentQMgrPort sharedPort -agentQMgrChannel CHANNEL1
```

如前所述，必须调整通过运行 **fteCreateAgent** 命令生成的 MQSC 命令，以在 CFSTRUCT 属性中指定 QSGDISP (SHARED) 和相应的耦合设施结构。

命令队列管理器

MFT 命令队列管理器可以是 QSG 的一部分。 但是，指定命令队列管理器时不能使用 QSG 的名称; 必须使用特定的队列管理器名称。

协调队列管理器

MFT 协调队列管理器可以是 QSG 的一部分。 但是，与命令队列管理器一样，指定协调队列管理器时不能使用 QSG 的名称; 必须使用特定的队列管理器名称。

连接到 QSG 的命令

MFT 向受管代理，传输和代理，命令或协调队列管理器提供了许多命令。 如果队列管理器位于 QSG 中，那么只能使用那些连接到代理队列管理器的命令。

以下是连接到代理队列管理器的命令的列表:

- **fteCleanAgent**
- **fteCreateAgent**
- **fteCreateBridgeAgent**
- **fteCreateCDAgent**
- **fteDeleteAgent**

请注意，在运行其他 MFT 命令时，必须提供队列管理器的名称。

将 Managed File Transfer for z/OS 与 JZOS Java 启动程序配合使用

您可以在 IBM MQ for z/OS 系统上应用本主题中的指示信息作为在企业中使用 Managed File Transfer 的替代方法。

概述

Managed File Transfer for z/OS (MFT) 使用标准 z/OS 安装过程。 运行 MFT 命令的替代方法是使用 JCL 和 JZOS Java Launcher。

请参阅 JZOS Batch Launcher and Toolkit，以获取更多详细信息。

如果 JCL 未能正确处理，请参阅 JZOS 的常见 MFT 问题。

示例 JCL

```
//JOHNDOEA JOB 1,MSGCLASS=H
// JCLLIB ORDER=(SCEN.MFT.JCL)   (1)
// INCLUDE MEMBER=BFGJCL8       (2)
//     DD * (2A)
. ${BFG_PROD}/bin/fteBatch createAgent  (3)
export IBM_JAVA_OPTIONS="${BFG_JAVA_OPTIONS} ${BFG_LANG}"  (4)
export JZOS_MAIN_ARGS="${BFG_MAIN_ARGS}" (4)
//MAINARGS DD *
-agentName MYAGENT  (5)
-f
-agentQMgr MQPD
-p MQPD
/*
```

其中:

- (1) 是所包含的 JCL 语句的位置
- (2) 包括从 1) 中的位置指定的 JCL 成员
- (2A) 对 //STDENV 进行扩展 - 请见下文
- (3) 这是要执行的命令，不带前导 fte 前缀

- (4) 这些行是必需内容，它们用于设置 JZOS 的信息
- (5) 用于该命令的参数
- BFGJCL8 成员（您可以选择您自己的名称）调用 JZOS。 该成员包含运行 MFT 所需的 STEPLIB 和其他 JCL。

您需要包括的其他 JCL

您应该包括 IBM MQ for z/OS 库的 JCL，如果您使用的是 Db2 记录器，则要包括 Db2 库的 JCL。

例如：

```
//WMQFTE EXEC PGM=JVMLDM86,REGION=0M PARM='+T'  (1)
//STEPLIB  DD DSN=SYS1.SIEALNKE,DISP=SHR        (2)
//* MQ libraries
//   DD DSN=MQM.V920.SCSQAUTH,DISP=SHR MQ Bindings
//   DD DSN=MQM.V920.SCSQANLE,DISP=SHR MQ Bindings
//   DD DSN=MQM.V920.SCSQLOAD,DISP=SHR MQ Bindings

//* DB2 libraries
//   DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNEXIT.DBCP
//   DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNLOAD
//   DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNLOD2
//SYSOUT  DD SYSOUT=H
//SYSPRINT DD SYSOUT=H
//STDOUT  DD SYSOUT=H
//STDERR  DD SYSOUT=H

//STDENV  DD DSN=SCEN.MFT.JCL(BFGZENV8),DISP=SHR  (3)
```

其中：

- (1) 是 JZOS 程序的名称。 查看 SYS1.SIEALNKE 以了解您系统上的版本。 添加 ,PARM='+T' 以提供其他诊断。
- (2) 这是带有 JZOS 程序的数据集。
- (3) 这是 shell 脚本的成员名。 用于定义 MFT 所需的参数。 请参阅 第 625 页的『用于定义 MFT 的 shell 脚本』。

  它可以是任何数据集和成员。 该名称需要位于文件末尾，因为 JCL 作业对此进行了扩展。 请参阅 第 624 页的『示例 JCL』 中的 2A。

用于定义 MFT 的 shell 脚本

在 第 625 页的『您需要包括的其他 JCL』 示例中，使用了成员 BFGZENV8。 这是基于 JZOS 概要文件。

您需要知道：

- 安装了 Java 的位置
- IBM MQ for z/OS Java 库和 MFT 库的位置。
- 用户标识需要位于特定组内才能被视为 IBM MQ for z/OS 管理员。 您需要此组的名称
- 如果消息不以英语表示，您需要知道要指定哪种语言。

示例文件

```
# This is a shell script that configures
# any environment variables for the Java JVM.
# Variables must be exported to be seen by the launcher.
# Use PARM='+T' and set -x to debug environment script problems
set -x
# . /etc/profile
#
# Java configuration (including MQ Java interface)
#
export _BPXK_AUTOCVT="ON"
export JAVA_HOME="/java/java71_bit64_sr3_fp30/J7.1_64/"
export PATH="/bin:${JAVA_HOME}/bin/classic/"
```

```
LIBPATH="/lib:/usr/lib:${JAVA_HOME}/bin"
LIBPATH="$LIBPATH:${JAVA_HOME}/bin/classic"
LIBPATH=$LIBPATH:"/mqm/V9R2M0/java/lib/"
export LIBPATH

export BFG_JAVA_HOME="${JAVA_HOME}"
export BFG_WTO="YES"
export BFG_GROUP_NAME=MQADM
export BFG_PROD="/mqm/V9R2M0/mqft"
export BFG_CONFIG="/u/johndoe/fteconfig"
# export BFG_LANG=" -Duser.language=de "
export BFG_LANG=" "

--------------------
```

其中：

export _BPXK_AUTOCVT="ON"
对于 Unicode 转换是必须项

export JAVA_HOME="/java/java71_bit64/J7.1_64/"
是 Java 目录的位置。 指定 Java 的路径名称。 该目录包含 bin 和其他目录。

export PATH="/bin:${JAVA_HOME}/bin/classic/"
设置 Java 可执行语句的路径语句

LIBPATH="/lib:/usr/lib:${JAVA_HOME}/bin"
设置 Java 可执行语句的库路径

LIBPATH="$LIBPATH:${JAVA_HOME}/bin/classic"
在 LIBPATH 语句中添加更多 Java 库。

LIBPATH=$LIBPATH: "/mqm/V9R2M0/java/lib/"
在库路径中添加 IBM MQ for z/OS 库。 在 z/OS UNIX System Services 中指定 IBM MQ for z/OS 库的名称。

export LIBPATH
使 LIBPATH 可用于 JZOS

export BFG_JAVA_HOME="${JAVA_HOME}"
将 BFG_JAVA_HOME 设置为以上指定的 JAVA_HOME 的值

export BFG_WTO="YES"
将 BFG_WTO 设置为 YES 会使用 WTO 将消息显示在作业日志中

export BFG_GROUP_NAME=MQADM
属于指定组成员的用户标识，将视为 IBM MQ for z/OS 管理员

导出 BFG_PROD = "/mqm/V9R2M0/mqft"
是 MFT 代码所在位置的路径

export BFG_DATA="/u/johndoe/fteconfig"
是存储 MFT 配置信息的位置

 # export BFG_LANG=" -Duser.language=de "
将语言定义为德语的已注释掉的语句

export BFG_LANG=" "
指定语言为缺省值英语。

`/lib/messages/BFGNVMessages_*.properties` 中 MFT 产品的内容列出了可用的语言。 缺省值为将值保留为空白，表示将使用英语。

相关任务

第 598 页的『配置 Managed File Transfer for z/OS』
Managed File Transfer for z/OS 需要定制以使组件能够正确运行。

规划 Managed File Transfer

IBM i 配置 MFT on IBM i

要在安装后开始使用 Managed File Transfer，必须为协调队列管理器和代理完成某些配置。

关于此任务

安装后，必须运行 Managed File Transfer 为新的协调队列管理器和新的代理提供的配置脚本，然后才可使用协调队列管理器和代理来传输文件。 随后，必须启动已创建的代理。

过程

1. 对于所有新的协调队列管理器，针对协调队列管理器运行 *coordination_qmgr_name*.mqsc 文件中的 MQSC 命令。 如果协调队列管理器与安装不在同一台计算机上，那么请将 MQSC 脚本文件复制到队列管理器所在计算机，然后运行该脚本。
   a) 从 IBM i 命令行使用以下命令启动 qshell：CALL QSHELL
   b) 切换到以下目录： /QIBM/UserData/mqm/mqft/config/*coordination_qmgr_name*
   c) 发出以下命令将 *coordination_qmgr_name* 替换为您的队列管理器的名称：

   ```
   /QSYS.LIB/QMQM.LIB/RUNMQSC.PGM coordination_qmgr_name < coordination_qmgr_name.mqsc
   ```

   您可以改为手动配置协调队列管理器。 有关更多信息，请参阅第 629 页的『为 MFT 配置协调队列管理器』。
2. 对于所有新代理：针对代理队列管理器运行 *agent_name*_create.mqsc 文件中的 MQSC 命令。
   如果代理队列管理器与代理不在同一台计算机上，请将 MQSC 脚本文件复制到队列管理器所在计算机，然后运行该脚本。
   a) 从 IBM i 命令行使用以下命令启动 qshell：CALL QSHELL
   b) 切换到以下目录： /QIBM/UserData/mqm/mqft/config/*agent_qmgr_name*/agents
   c) 发出以下命令以将 *agent_qmgr_name* 替换为您的代理队列管理器的名称，将 *agent_name* 替换为您的代理的名称：

   ```
   /QSYS.LIB/QMQM.LIB/RUNMQSC.PGM agent_qmgr_name < agent_name_create.mqsc
   ```

   您可以改为手动配置代理队列管理器。 有关更多信息，请参阅第 635 页的『配置 MFT 代理队列管理器』。
3. 如果尚未在安装过程中启动 QMFT 子系统，请使用以下命令从 IBM i 命令行启动 QMFT 子系统: STRSBS SBSD(QMQMMFT/QMFT)或 STRSBS QMQMMFT/QMFT
4. 使用 **fteStartAgent** 命令启动您的新代理。
   a) 从 IBM i 命令行使用以下命令启动 qshell：CALL QSHELL
   b) 切换到以下目录： /QIBM/ProdData/mqm/bin
   c) 发出以下命令以将 AGENT 替换为您的代理名称：

   ```
   ./fteStartAgent AGENT
   ```

下一步做什么

建议您设置沙箱以限制代理可访问的文件系统区域。 使用 MFT 代理程序沙箱中描述了此功能。

相关概念

第 627 页的『为首次使用配置 MFT』
在首次使用 Managed File Transfer 代理和队列管理器之前，必须先对其执行一些配置任务。

为首次使用配置 MFT

在首次使用 Managed File Transfer 代理和队列管理器之前，必须先对其执行一些配置任务。

连接到 IBM MQ

与 IBM MQ 队列管理器的所有网络通信（包括与 Managed File Transfer 相关的通信），都涉及 IBM MQ 通道。 IBM MQ 通道代表网络链路的一端。 通道可以归类为消息通道或 MQI 通道。

Managed File Transfer 和通道

Managed File Transfer 使用 MQI 通道，以客户机方式将代理连接到其代理队列管理器，并将命令应用程序（例如，**fteCreateTransfer**）连接到其命令队列管理器和协调队列管理器。 在缺省配置中，这些连接由一个名为 SYSTEM.DEF.SVRCONN 的 SVRCONN 通道建立，SYSTEM.DEF.SVRCONN 缺省情况下存在于所有队列管理器上。 由于这些缺省设置，因此您无需对 Managed File Transfer 基本安装更改任何 MQI 通道。

消息通道端点的类型有六种，但是本主题中只讨论发送方/接收方对。 请参阅分布式排队组件以了解有关其他通道组合的信息。

所需的消息路径

IBM MQ 消息只能在消息通道上传输，因此您必须确保这些通道可用于 Managed File Transfer 所需的所有消息路径。 这些路径不必是直接的；如果需要的话，可以通过中间队列管理器传递消息。 本主题只讨论点到点的直接通信。 请参阅如何与远程队列管理器通信，以了解有关这些选项的更多信息。

Managed File Transfer 使用的通信路径有以下几种：

代理到代理
在其关联的队列管理器之间传输文件需要双向通信的任何两个代理。 由于该路径中传输批量数据，因此您可以根据自己的需要，考虑使路径尽可能短、快速且经济。

代理到协调
参与传输的代理发出的日志消息必须能够传递到协调队列管理器。

命令到代理
命令应用程序或 IBM MQ Explorer（使用命令队列管理器）连接到的所有队列管理器都必须能够向使用这些命令应用程序控制的代理的队列管理器发送消息。 要使命令显示反馈消息，应使用双向连接。

有关更多信息，请参阅为企业使用的平台验证 *IBM MQ* 安装。

相关概念

第 632 页的『配置多实例队列管理器以使用 MFT』
从 IBM WebSphere MQ 7.0.1 开始支持创建多实例队列管理器。 多实例队列管理器能够在备用服务器上自动地重新启动。 Managed File Transfer 支持连接到多个多实例代理队列管理器、一个多实例协调队列管理器和一个多实例命令队列管理器。

相关任务

第 628 页的『配置 MFT 网络队列管理器』
如果您的 Managed File Transfer 网络包含多个 IBM MQ 队列管理器，那么这些 IBM MQ 队列管理器相互之间必须能够进行远程通信。

第 629 页的『为 MFT 配置协调队列管理器』
运行 **fteSetupCoordination** 命令之后，运行 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* 目录中的 *coordination_qmgr_name*.mqsc 脚本以对协调队列管理器执行必要配置。 但是，如果您希望手动执行此配置，请在协调队列管理器上完成以下步骤。

配置 MFT 网络队列管理器

如果您的 Managed File Transfer 网络包含多个 IBM MQ 队列管理器，那么这些 IBM MQ 队列管理器相互之间必须能够进行远程通信。

关于此任务

要实现队列管理器之间的互相通信，您可以使用以下两种方法进行配置：

- 通过设置 IBM MQ 队列管理器集群。

  有关 IBM MQ 队列管理器集群以及如何配置这些集群的信息，请参阅 第 235 页的『配置队列管理器集群』。
- 通过设置队列管理器之间的通道，如下所述：

设置队列管理器之间的通道

在您的队列管理器之间，设置以下消息通道：

- 从代理队列管理器到协调队列管理器
- 从命令队列管理器到代理队列管理器。
- 从代理队列管理器到命令队列管理器（从而能够通过命令显示反馈消息）。
- 从命令队列管理器到协调队列管理器
- 从代理队列管理器到 Managed File Transfer 网络中的其他任何代理队列管理器

如果需要有关如何设置这种通信的更多信息，请首先参阅以下信息：使用 MQSC 管理远程 IBM MQ 对象。

以下是建议的几个示例步骤：

过程

1. 在 IBM MQ 队列管理器上，创建一个与协调队列管理器同名的传输队列。
   您可以使用以下 MQSC 命令：

   ```
   DEFINE QLOCAL(coordination-qmgr-name) USAGE(XMITQ)
   ```

2. 在 IBM MQ 队列管理器上，创建到 Managed File Transfer 协调队列管理器的发送方通道。

   先前步骤中创建的传输队列的名称是此通道的必需参数。

   如果需要与 Managed File Transfer for IBM WebSphere MQ 7.5 或 Managed File Transfer 代理程序进行通信，请确保发送方通道的 CONVERT 参数设置为 no。（IBM WebSphere MQ File Transfer Edition 的早期版本始终以 UTF-8 格式发布消息，这意味着任何数据转换都会损坏消息。 对于 Managed File Transfer for IBM MQ 8.0 或更高版本上的代理程序，这不是必需的，因为将以空白格式发布消息。 另请注意 IBM 文件传输版（FTE）不再受支持。 WebSphereMQ 文件传输版（FTE）已不再受支持。 要从 FTE 迁移至 IBM MQ 中的 Managed File Transfer 组件，请参阅迁移 Managed File Transfer。）

   您可以使用以下 MQSC 命令：

   ```
   DEFINE CHANNEL(channel-name) CHLTYPE(SDR) CONNAME('coordination-qmgr-host(coordination-qmgr-port)')
    XMITQ(coordination-qmgr-name) CONVERT(NO)
   ```

   注： 仅在需要时设置 CONVERT(NO)。

3. 在 Managed File Transfer 协调队列管理器上，创建到 IBM MQ 队列管理器的接收方通道。 为该接收方通道指定与 IBM MQ 队列管理器上的发送方通道相同的名称。
   您可以使用以下 MQSC 命令：

   ```
   DEFINE CHANNEL(channel-name) CHLTYPE(RCVR)
   ```

下一步做什么

接下来，遵循协调队列管理器的配置步骤: 第 629 页的『为 MFT 配置协调队列管理器』。

为 MFT 配置协调队列管理器

运行 **fteSetupCoordination** 命令之后，运行 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* 目录中的 *coordination_qmgr_name*.mqsc 脚本以对协调队列管理器执行必要配置。 但是，如果您希望手动执行此配置，请在协调队列管理器上完成以下步骤。

关于此任务

过程

1. 创建一个名为 SYSTEM.FTE 的本地队列。
2. 将 SYSTEM.FTE 队列添加到 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名称列表中。
3. 创建一个名为 SYSTEM.FTE 的主题，主题的字符串为 SYSTEM.FTE。

4. 确保 SYSTEM.FTE 主题的非持久性消息传递 (NPMSGDLV) 属性和持久性消息传递 (PMSGDLV) 属性设置为 ALLAVAIL。
5. 确保协调队列管理器的发布/预订方式 (PSMODE) 属性设置为 ENABLED。

下一步做什么

如果在已配置为协调队列管理器的队列管理器上运行 `strmqm -c` 命令，那么该命令会删除在 步骤 2 (添加 SYSTEM.FTE 队列到 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名称列表)。 这是因为 `strmqm -c` 将重新创建缺省的 IBM MQ 对象并撤销 Managed File Transfer 更改。 因此，如果已使用 `strmqm -c` 启动队列管理器，请完成以下步骤之一：

- 重新在队列管理器上运行 *coordination_qmgr_name*.mqsc 脚本。
- 重复步骤 2。

相关概念

第 627 页的『连接到 IBM MQ』
与 IBM MQ 队列管理器的所有网络通信（包括与 Managed File Transfer 相关的通信），都涉及 IBM MQ 通道。 IBM MQ 通道代表网络链路的一端。 通道可以归类为消息通道或 MQI 通道。

第 632 页的『配置多实例队列管理器以使用 MFT』
从 IBM WebSphere MQ 7.0.1 开始支持创建多实例队列管理器。 多实例队列管理器能够在备用服务器上自动地重新启动。 Managed File Transfer 支持连接到多个多实例代理队列管理器、一个多实例协调队列管理器和一个多实例命令队列管理器。

相关任务

第 628 页的『配置 MFT 网络队列管理器』
如果您的 Managed File Transfer 网络包含多个 IBM MQ 队列管理器，那么这些 IBM MQ 队列管理器相互之间必须能够进行远程通信。

相关参考

fteSetupCoordination

创建 IBM MQ 文件传输结构

您可以根据连接到同一台机器上的队列管理器的单一代理，配置 Managed File Transfer 结构。

关于此任务

MFT 配置存储在代理所在的机器上，IBM MQ DataPath 下的文件结构中。

以下样本配置适用于 IBM MQ 8.0 队列管理器中名为 SAMPLECOORD (已禁用安全性) 的 MFT 以及名为 SAMPLEAGENT 的单个 MFT 代理程序:

```
+--- config
             +--- SAMPLECOORD
                            +--- command.properties
                            +--- coordination.properties
                            +--- SAMPLECOORD.mqsc
                            +--- agents
                                           +--- SAMPLEAGENT
                                                          +--- agent.properties
                                                          +--- SAMPLEAGENT_create.mqsc
                                                          +--- SAMPLEAGENT_delete.mqsc

+---logs
             +--- SAMPLECOORD
                            +--- agents
                                           +--- SAMPLEAGENT
                                                          +--- logs
```

此示例假定队列管理器安全性已禁用。 以下命令（在 **runmqsc** 中运行）将在重新启动队列管理器后禁用安全性:

```
runmqsc queue manager
 alter qmgr CONNAUTH(NONE);
```

```
alter qmgr CHLAUTH(DISABLED);
end;
```

对于在 IBM MQ 8.0 或更高版本的 MFT 中启用了安全性的配置， **CONNAUTH** 需要所有与队列管理器连接的 MFT 命令来提供用户标识和密码凭证。 您可以为每个命令额外应用参数 **-mquserid** 和 **-mqpassword**，或者定义 MQMFTCredentials.xml 文件。 以下样本凭证文件定义了 fteuser 的用户标识，当连接到队列管理器 SAMPLECOORD 时将对其应用 MyPassword 密码：

```
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://wmqfte.ibm.com/MQMFTCredentials MQMFTCredentials.xsd">
    <tns:qmgr mqPassword="MyPassword" MyUserId="fteuser" name="SAMPELCOORD"/>
</tns:mqmftCredentials>
```

有关更多信息，请参阅 MFT 和 IBM MQ 连接认证。

注意:

- 要查找 MFT 配置目录，请使用 **fteDisplayVersion -v** 命令。
- 对于 z/OS 用户，MQMFTCredential.xml 文件可以作为分区数据集的成员来进行查找，其变量记录格式为 (RECFM=V)，或未定义记录格式 (RECFM=U)。
- 对于启用了安全性的配置，请将以下参数添加到以下步骤以将凭证与相关队列管理器相关联: -credentialsFile *full credential file path*。
- MQMFTCredential.xml 中的明文密码可以使用以下命令进行加密：

```
fteObfuscate -credentialsFile full file path to MQMFTCredentials.xml
```

过程

1. 创建协调队列管理器。

   协调队列管理器是一个队列管理器，用于接收来自其代理的所有传输日志和状态信息。 运行以下命令：

   ```
   fteSetupCoordination -coordinationQMgr coordination_qmgr_name
   ```

   这创建了基本顶级配置，并创建了 IBM MQ 脚本文件来调用 *coordination_qmgr_name*.mqsc。

   然后，需要运行以下 IBM MQ 命令，将该配置装入到队列管理器中：

   ```
   runmqsc queue manager name < coordination_qmgr_name.mqsc
   ```

   注: 要将 TCP 客户机连接到队列管理器，您可以使用：

   ```
   fteSetupCoordination -coordinationQMgr coordination_qmgr_name
   -coordinationQMgrHost coordination_qmgr_host -coordinationQMgrPort coordination_qmgr_port
   -coordinationQMgrChannel coordination_qmgr_channel
   ```

   对于已创建的 *coordination_qmgr_name*.mqsc，您将需要在协调队列管理器运行的同一台机器上运行 **runmqsc** 命令。

2. 创建命令队列管理器。

   命令队列管理器是已预先配置的一个队列管理器，以便 IBM MQ 基础结构可以将 MFT 请求路由至相关的代理。 运行以下命令：

   ```
   fteSetupCommands -connectionQMgr Command QM Name -p Coordination QM Name
   ```

   这将在协调目录中创建 command.properties 文件。 请注意， -p 是可选的，如果针对缺省协调设置了命令，那么将不需要该选项。

   注: 要将 TCP 客户机连接到队列管理器，您可以使用：

   ```
   fteSetupCommands -p coordination_qmgr_name -commandQMgr connection_qmgr_name
   -commandQMgrHost connection_qmgr_host -commandQMgrPort connection_qmgr_port
   -commandQMgrChannel connection_qmgr_channel
   ```

3. 创建代理。

代理是用于发送和接收文件的应用程序。 运行以下命令：

```
fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMgr agent_qmgr_name
```

这将在协调下创建代理配置，并创建 IBM MQ 脚本文件以调用代理配置目录中的 *agent_name*.mqsc。 运行以下 IBM MQ 命令，以将 IBM MQ 脚本文件装入到队列管理器中：

```
runmqsc agent_qmgr_name < agent_name_create.mqsc file
```

注: 要将 TCP 客户机连接到队列管理器，您可以使用：

```
fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMgr agent_qmgr_name
-agentQMgrHost agent_qmgr_host -agentQMgrPort agent_qmgr_port -agentQMgrChannel
agent_qmgr_channel
```

4. 启动代理。
   运行以下命令：

```
fteStartAgent -p coordination_qmgr_name agentName
```

代理将在后台启动，并返回命令提示符。 要检查代理是否正在运行，请运行以下命令：

```
fteListAgents -p coordination_qmgr_name
```

这将显示代理的状态。 如果代理成功运行，那么该代理将报告为处于“就绪”状态。

结果

基本 MFT 基础结构准备就绪，您现在可以使用 **fteCreateTransfer** 命令请求传输。 或者，如果 IBM MQ Explorer 可用，使用 MFT 插件创建和监视传输。

通过重复步骤 3：创建代理，可向配置添加更多代理。 如果使用了 TCP 客户机连接，那么这些代理可以位于不同机器上。 对于不同机器，必须对每台机器重复 **fteSetupCoordination** 和 **fteSetupCommands** 命令，但是，不需要再运行 mqsc 脚本。

较复杂的配置的坐标和每个代理可能有独立的队列管理器。 在这些情况中，需要将各种队列管理器连接起来。

相关任务
fteListAgents 命令未列出 MFT 代理程序时要执行的操作
相关参考
fteSetupCoordination
fteSetupCommands：创建 MFT command.properties 文件
fteCreateAgent
fteObfuscate：对敏感数据进行加密
MFT 凭证文件格式
MFT agent.properties 文件

配置多实例队列管理器以使用 MFT

从 IBM WebSphere MQ 7.0.1 开始支持创建多实例队列管理器。 多实例队列管理器能够在备用服务器上自动地重新启动。 Managed File Transfer 支持连接到多个多实例代理队列管理器、一个多实例协调队列管理器和一个多实例命令队列管理器。

配置多实例队列管理器

要点: 有关配置 IBM MQ 多实例队列管理器的信息，请参阅 第 415 页的『多实例队列管理器』。 在尝试配置多实例队列管理器以使用 Managed File Transfer 之前，请确保您已读取此信息。

将多实例队列管理器用作代理队列管理器

要使代理同时连接到多实例队列管理器的活动和备用实例，请将 agentQMgrStandby 属性添加到代理的 agent.properties 文件中。 agentQMgrStandby 属性为备用队列管理器实例定义用于客户机连接的主机名和端口号。 必须以 MQ CONNAME 格式提供属性值，即 *host_name*(*port_number*)。

agentQMgr 属性指定多实例队列管理器的名称。 agentQMgrHost 属性为活动队列管理器实例指定主机名，agentQMgrPort 属性为活动队列管理器实例指定端口号。 代理必须以客户机方式连接到多实例队列管理器的活动和备用实例。

请参阅 MFT agent.properties 文件 以获取更多信息。

该示例显示了连接到名为 QM_JUPITER 的多实例队列管理器的 AGENT1 的 agent.properties 文件的内容。 QM_JUPITER 的活动实例位于系统 host1 上，并使用端口号 1414 进行客户机连接。 QM_JUPITER 的备用实例位于系统 host2 上，并使用端口号 1414 进行客户机连接。

```
agentName=AGENT1
agentDesc=
agentQMgr=QM_JUPITER
agentQMgrPort=1414
agentQMgrHost=host1
agentQMgrChannel=SYSTEM.DEF.SVRCONN
agentQMgrStandby=host2(1414)
```

将多实例队列管理器用作协调队列管理器

要启用与多实例协调队列管理器的活动实例和备用实例的连接，请将 coordinationQMgrStandby 属性添加到 Managed File Transfer 拓扑中的所有 coordination.properties 文件。

请参阅 MFT coordination.properties 文件 以获取更多信息。

该示例显示了指定名为 QM_SATURN 的多实例协调队列管理器的连接详细信息的 coordination.properties 文件的内容。 QM_SATURN 的活动实例位于系统 coordination_host1 上，并使用端口号 1420 进行客户机连接。 QM_SATURN 的备用实例位于系统 coordination_host2 上，并使用端口号 1420 进行客户机连接。

```
coordinationQMgr=QM_SATURN
coordinationQMgrHost=coordination_host1
coordinationQMgrPort=1420
coordinationQMgrChannel=SYSTEM.DEF.SVRCONN
coordinationQMgrStandby=coordination_host2(1420)
```

Managed File Transfer 独立的记录器必须始终以绑定方式连接到其队列管理器。 在将独立的记录器与多实例协调队列管理器配合使用时，请以绑定方式将独立的记录器连接到另一队列管理器。 第 661 页的『MFT 独立记录器的备用配置』中说明了您需要执行的步骤。 必须使用多实例协调队列管理器的实例的主机名和端口号，定义独立的记录器队列管理器和协调队列管理器间的通道。 有关如何执行此操作的信息，请参阅第 415 页的『多实例队列管理器』。

IBM MQ Explorer 的 Managed File Transfer 插件以客户机方式连接到协调队列管理器。 如果多实例协调队列管理器的活动实例失败，那么协调队列管理器的备用实例将激活，并且该插件将重新进行连接。

Managed File Transfer 命令 **fteList*** 和 **fteShowAgentDetails** 可直连到协调队列管理器。 如果多实例协调的活动实例不可用，这些命令将尝试连接到协调队列管理器的备用实例。

将多实例队列管理器用作命令队列管理器

要启用与多实例命令队列管理器的活动实例和备用实例的连接，请将 connectionQMgrStandby 属性添加到 Managed File Transfer 拓扑中的所有 command.properties 文件。

请参阅 MFT command.properties 文件 以获取更多信息。

该示例显示了指定名为 QM_MARS 的多实例命令队列管理器的连接详细信息的 command.properties 文件的内容。 QM_MARS 的活动实例位于系统 command_host1 上，并使用端口号 1424 进行客户机连接。 QM_MARS 的备用实例位于系统 command_host2 上，并使用端口号 1424 进行客户机连接。

```
connectionQMgr=QM_SATURN
connectionQMgrHost=command_host1
connectionQMgrPort=1424
connectionQMgrChannel=SYSTEM.DEF.SVRCONN
connectionQMgrStandby=command_host2(1424)
```

相关概念

第 627 页的『连接到 IBM MQ』
与 IBM MQ 队列管理器的所有网络通信（包括与 Managed File Transfer 相关的通信），都涉及 IBM MQ 通道。 IBM MQ 通道代表网络链路的一端。 通道可以归类为消息通道或 MQI 通道。

相关任务

第 628 页的『配置 MFT 网络队列管理器』
如果您的 Managed File Transfer 网络包含多个 IBM MQ 队列管理器，那么这些 IBM MQ 队列管理器相互之间必须能够进行远程通信。

第 629 页的『为 MFT 配置协调队列管理器』
运行 **fteSetupCoordination** 命令之后，运行 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* 目录中的 *coordination_qmgr_name*.mqsc 脚本以对协调队列管理器执行必要配置。 但是，如果您希望手动执行此配置，请在协调队列管理器上完成以下步骤。

保留 MFT 日志消息

Managed File Transfer 将文件传输进度和日志信息发送到协调队列管理器。 协调队列管理器将此信息发布到 SYSTEM.FTE 主题的任何匹配预订。 如果无任何预订，将不保留此信息。

确保保留信息的方法

如果传输进度或日志信息对您的业务很重要，那么必须执行以下步骤之一以确保保留此信息：

- 使用 Managed File Transfer 数据库记录器将发布到 SYSTEM.FTE/Log 主题的消息复制到 Oracle 或 Db2 数据库。
- 定义 SYSTEM.FTE 主题预订（将发布存储在 IBM MQ 队列上）。 在传输任何文件传输之前定义该预订，以确保所有进度和日志消息保留在队列上。
- 编写一个应用程序，使其使用消息队列接口 (MQI) 或 IBM MQ JMS 创建持久预订并处理交付到该预订的发布。 该应用程序必须在传输任何文件之前处于运行状态，以确保应用程序接收所有进度和日志消息。

在接下来的部分中，更详细地描述了每种方法。

请勿依赖于 IBM MQ Explorer 插件来保留日志信息。

使用 Managed File Transfer 数据库记录器保留日志消息

数据库记录器是 Managed File Transfer 的可选组件，可使用它将日志信息复制到数据库中以用于分析和审计目的。 数据库记录器是您在托管协调队列管理器和数据库的系统上安装的独立 Java 应用程序。 有关数据库记录器的更多信息，请参阅第 643 页的『配置 MFT 记录器』。

使用 IBM MQ Explorer 插件保留进度和日志消息

首次启动 IBM MQ Explorer 插件的实例时，该实例将在协调队列管理器上创建持久预订。 该持久预订用于收集**“传输日志”**和**“当前传输进度”**视图中显示的信息。

持久预订的名称将作为前缀，以显示该预订是由 IBM MQ Explorer MFT 插件、主机名和用户名称（例如，MQExplorer_MFT_Plugin_HOST_TJWatson）创建的。

在管理员想要删除 IBM MQ Explorer 插件实例不再实际使用的持久预订时添加该前缀。

在协调队列管理器上使用持久预订可能会导致在 SYSTEM.MANAGED.DURABLE 队列上构建消息。 如果您具有大容量 Managed File Transfer 网络，请不要频繁使用 IBM MQ Explorer 插件，否则此消息数据可能会填满本地文件系统。

要避免发生此情况，可以指定 IBM MQ Explorer 插件使用对协调队列管理器的非持久预订。 在 IBM MQ Explorer 中执行以下步骤：

1. 选择**窗口 > 首选项 > MQ Explorer > 受管文件传输**
2. 在**传输日志订阅类型**列表中，选择 NON_DURABLE。

将发布存储在 IBM MQ 队列上

要将日志或进度消息存储在 IBM MQ 队列上，请在将消息转发到该队列的协调队列管理器上配置预订。 例如，要将所有日志消息转发到名为 LOG.QUEUE 的队列，请提交以下 MQSC 命令：

```
define sub(MY.SUB) TOPICSTR('Log/#') TOPICOBJ(SYSTEM.FTE) DEST(LOG.QUEUE)WSCHEMA(TOPIC)
```

将日志消息转发到 IBM MQ 队列后，这些消息将永久存储在该队列上，直至使用该队列的 IBM MQ 应用程序处理这些消息为止。

编写用于管理 SYSTEM.FTE 主题的持久预订的应用程序

您可以使用 IBM MQ 支持的某个应用程序编程接口，编写用于管理其自己的持久预订（针对 SYSTEM.FTE 主题）的应用程序。 这些应用程序可以接收 IBM MQ 队列或日志消息，并根据您的业务需要相应地处理这些消息。

有关可用应用程序编程接口的更多信息，请参阅开发应用程序。

配置 MFT 代理队列管理器

安装后，运行 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/agents/*agent_name* 目录中的 *agent_name*_create.mqsc 脚本以对代理队列管理器执行必要配置。 但是，如果要手动执行此配置，请在代理队列管理器上完成以下步骤。

过程

1. 创建代理操作队列。
   这些队列的名称如下：
   - SYSTEM.FTE.COMMAND.*agent_name*
   - SYSTEM.FTE.DATA.*agent_name*
   - SYSTEM.FTE.EVENT.*agent_name*
   - SYSTEM.FTE.REPLY.*agent_name*
   - SYSTEM.FTE.STATE.*agent_name*

   有关队列参数的信息，请参阅 MFT 代理队列设置。
2. 创建代理权限队列。
   这些队列的名称如下：
   - SYSTEM.FTE.AUTHADM1.*agent_name*
   - SYSTEM.FTE.AUTHAGT1.*agent_name*
   - SYSTEM.FTE.AUTHMON1.*agent_name*
   - SYSTEM.FTE.AUTHOPS1.*agent_name*
   - SYSTEM.FTE.AUTHSCH1.*agent_name*
   - SYSTEM.FTE.AUTHTRN1.*agent_name*

   有关队列参数的信息，请参阅 MFT 代理队列设置。

下一步做什么

有关创建和配置协议网桥代理的信息，请参阅 fteCreateBridgeAgent (创建和配置 MFT 协议网桥代理) 和 为 FTPS 服务器配置协议网桥。

在集群中为多个通道配置 MFT 代理

如果您希望在集群配置中使用 IBM MQ 多通道支持，请首先将 agentMultipleChannelsEnabled 属性设置为 `true`，然后完成本主题中的步骤。

关于此任务

在集群中，多通道支持只能由目标代理的队列管理器上的 IBM MQ 定义启用。

除了 Managed File Transfer 代理程序所需的标准 IBM MQ 配置步骤之外，您还必须完成本主题中的步骤，这些步骤在 配置 MFT 以首次使用中列出。

以下配置示例使用 **runmqsc** 命令。

过程

1. 为每个要使用的通道定义一个集群接收方通道。 例如，如果您要使用两个通道：

   ```
   DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
   DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
   ```

   其中：

   - *DESTQMGRNAME* 是目标代理的队列管理器的名称。
   - *MFTCLUSTER* 是 IBM MQ 集群的名称。

   建议您对通道使用 *MFTCLUSTER.DESTMGRNAME*_n 命名约定，但不作强制要求。
2. 为每个对应的通道定义一个队列管理器别名。 例如：

   ```
   DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
   DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
   ```

   必须使用 SYSTEM.FTE.*DESTQMGRNAME_n* 队列管理器别名 的命名约定，因为发送代理程序搜索此格式的队列管理器别名。 用于 *n* 的编号必须从 1 开始，而且必须是连续的。 您必须进行集群范围的定义，以使这些定义可在源代理的队列管理器上使用。

   为了使源代理和目标代理能够正确确定队列管理器别名的数量，**请勿**为队列管理器定义缺省的 XMITQ。

相关任务

第 636 页的『为多个通道配置 MFT 代理：非集群』
如果您希望在非集群配置中使用 IBM MQ 多通道支持，请首先将 agentMultipleChannelsEnabled 属性设置为 `true`，然后完成本主题中的步骤。

相关参考

MFT `agent.properties` 文件

为多个通道配置 MFT 代理：非集群

如果您希望在非集群配置中使用 IBM MQ 多通道支持，请首先将 agentMultipleChannelsEnabled 属性设置为 `true`，然后完成本主题中的步骤。

关于此任务

在非集群配置中，多通道支持由源代理和目标代理的队列管理器上的 IBM MQ 定义启用。

除了 Managed File Transfer 代理程序所需的标准 IBM MQ 配置步骤之外，您还必须完成本主题中的步骤，这些步骤在 配置 MFT 以首次使用中列出。

以下步骤假定源队列管理器和目标队列管理器之间使用发送方/接收方通道进行通信。

以下配置示例使用 **runmqsc** 命令。

过程

1. 在目标代理的队列管理器中，为每个要使用的通道定义一个接收方通道。 例如，如果您要使用两个通道：

   ```
   DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(RCVR) TRPTYPE(TCP)
   DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(RCVR) TRPTYPE(TCP)
   ```

   其中：DESTQMGRNAME 是目标代理的队列管理器的名称。

   建议您对通道使用 TO.DESTMGRNAME_n 命名约定，但不作强制要求。 接收方通道的名称必须和源代理的队列管理器上对应的发送方通道的名称匹配。
2. 在源代理的队列管理器中，为每个要使用的通道定义一个传输队列。 例如，如果您要使用两个通道：

   ```
   DEFINE QLOCAL(DESTQMGRNAME_1) USAGE(XMITQ)
   DEFINE QLOCAL(DESTQMGRNAME_2) USAGE(XMITQ)
   ```

   建议您对传输队列使用 DESTMGRNAME_n 命名约定，但不作强制要求。 您定义的传输队列会在后续步骤中的发送方通道定义和队列管理器别名定义中引用。
3. 在源代理的队列管理器中，为每个要使用的通道定义一个发送方通道。 例如，如果您要使用两个通道：

   ```
   DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
   XMITQ(DESTQMGRNAME_1)
   DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
   XMITQ(DESTQMGRNAME_2)
   ```

   建议您对通道使用 TO.DESTMGRNAME_n 命名约定，但不作强制要求。 发送方通道的名称必须和目标代理的队列管理器上对应的接收方通道的名称匹配。
4. 在源代理的队列管理器中，为每个对应的通道定义一个队列管理器别名。 例如：

   ```
   DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_1)
   DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_2)
   ```

   您必须对队列管理器的别名使用 SYSTEM.FTE.DESTQMGRNAME_n 命名约定，因为发送方代理会按此格式搜索队列管理器。 用于 *n* 的编号必须从 1 开始，而且必须是连续的。

   为了使代理能够正确确定队列管理器别名的数量，**请勿**为队列管理器定义缺省的 XMITQ。

相关任务

第 636 页的『在集群中为多个通道配置 MFT 代理』
如果您希望在集群配置中使用 IBM MQ 多通道支持，请首先将 agentMultipleChannelsEnabled 属性设置为 `true`，然后完成本主题中的步骤。

相关参考

MFT `agent.properties` 文件

Windows 通过 MSCS 配置 MFT 代理

Managed File Transfer (MFT) 代理 Microsoft Cluster Service (MSCS) 设置受支持，前提是平台受 MFT 支持，并运行 Windows 的某个版本。

关于此任务

此任务描述了为实现 MFT 代理程序的故障转移而可以遵循的两种方案：

- 方案 1: 将代理程序配置为 MSCS 资源。
- 方案 2: 将代理队列管理器和代理配置为 MSCS 资源。

过程

方案 1: 将代理程序配置为 MSCS 资源

- 要将代理程序配置为 MSCS 资源，请完成以下步骤:
  a) 在集群中的每台机器上本地安装 Managed File Transfer。
  请参阅安装 Managed File Transfer。
  b) 在集群中的主机器上创建代理程序。
  应该将代理配置为使用 CLIENT 传输连接到代理队列管理器。 确保在队列管理器上为此代理创建所有对象。 有关如何执行此操作的信息，请参阅 设置代理程序。
  c) 通过将 Windows 服务工具中代理程序服务的 **启动类型** 字段设置为 **手动**，修改代理程序以作为 Windows 服务运行，并对其进行配置，以使其不会在 Windows 重新启动时自动启动。
  有关更多信息，请参阅 将 MFT 代理作为 Windows 服务启动。
  d) 在辅助机器上重复方案 1 的步骤 第 638 页的『2』 和步骤 第 638 页的『3』 。
  这将确保集群中的另一台机器上存在日志，属性等的文件结构。 请注意，无需像步骤 第 638 页的『2』 中那样创建队列管理器对象。
  e) 在主机器上，将代理程序添加为 MSCS 控制下的 "通用服务"。
  要执行此操作:
    a. 右键单击集群，然后选择 **角色-> 添加资源-> "通用服务"**。
    b. 从 Windows 服务列表中，选择代理程序服务，然后单击 **下一步** 以完成配置向导。
  代理程序服务现在添加为 MSCS 资源。 如果发生故障转移，那么将在另一台机器上启动代理程序服务。

方案 2: 将代理队列管理器和代理配置为 MSCS 资源

- 要将代理队列管理器和代理配置为 MSCS 资源，请完成以下步骤:
  a) 配置代理队列管理器以作为 MSCS 资源运行。
  有关如何执行此操作的信息，请参阅第 406 页的『将队列管理器置于 MSCS 控制之下』。
  b) 在集群中的主机器上创建代理程序。
  应该将代理配置为使用 BINDINGS 传输连接到代理队列管理器。 确保在队列管理器上为此代理创建所有对象。 有关如何执行此操作的信息，请参阅 设置代理程序。
  c) 通过将 Windows 服务工具中代理程序服务的 **启动类型** 字段设置为 **手动**，修改代理程序以作为 Windows 服务运行，并对其进行配置，以使其不会在 Windows 重新启动时自动启动。
  有关更多信息，请参阅 将 MFT 代理作为 Windows 服务启动。
  d) 确保代理队列管理器 (受 MSCS 控制) 正在辅助机器上运行。
  在此机器上创建的代理将使用 BINDINGS 传输连接到队列管理器，因此在创建代理时需要可用。
  e) 在辅助机器上重复方案 2 的步骤 第 638 页的『2』 和步骤 第 638 页的『3』 。
  这将确保集群中的另一台机器上存在日志，属性等的文件结构。 请注意，无需像步骤 第 638 页的『2』 中那样创建队列管理器对象。
  f) 将代理程序添加为 MSCS 控制下的 "通用服务"。
  要执行此操作:
    a. 右键单击集群，然后选择 **角色-> 添加资源-> "通用服务"**。
    b. 从 Windows 服务列表中，选择代理程序服务，然后单击 **下一步** 以完成配置向导。
  g) 修改代理程序服务的资源属性以在依赖关系列表中添加队列管理器资源。
  这将确保在启动代理程序之前启动队列管理器资源。
  h) 使队列管理器资源脱机，然后使代理程序资源联机。 验证队列管理器资源和代理程序是否都已启动。
  如果发生故障转移，那么将在辅助机器上启动代理服务和代理队列管理器。

V 9.2.0 Managed File Transfer 中的高可用性代理程序

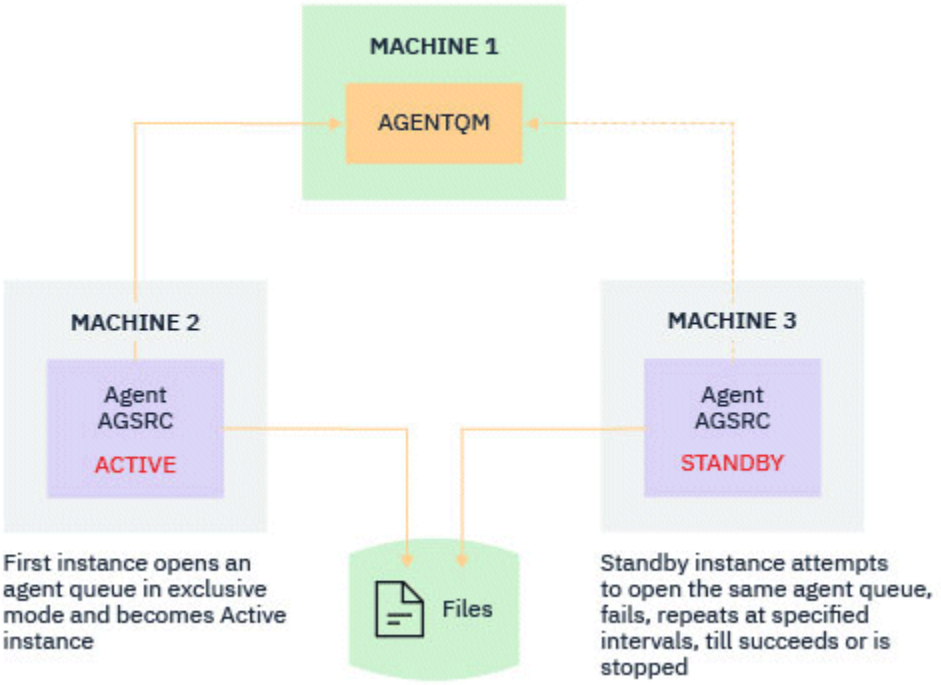
您可以在 MFT 中配置标准或网桥代理以在高可用性 (HA) 配置中运行。 HA 设置中涉及具有相同配置的一对代理程序实例，其中一个实例在一台机器上运行，而另一个实例在另一台机器上运行。 这两个实例都配置为连接到同一代理队列管理器。

概述

只有两个实例 (称为 活动实例) 中的一个实例正在处理文件传输，而另一个实例 (称为 备用实例) 处于部分初始化状态，并且无法处理任何文件传输。

当活动实例失败或失去与队列管理器的连接时，备用实例将完成其初始化，变为活动状态并开始处理文件传输。 从最后一个已知检查点恢复活动实例失败时正在进行的任何进行中传输。

下图显示了活动代理程序和备用代理程序的公共配置:



注意:

1. 代理程序的一个实例在两台不同的机器上运行，其中一个实例作为 活动实例，另一个实例作为 备用实例。
2. 代理程序的每个实例都在不同的机器上运行，其中一个实例作为活动实例，另一个实例作为备用实例。
3. 在代理的两个实例之间共享同一组代理队列。
4. 代理的两个实例都需要访问同一共享文件系统才能执行受管传输。

主动/备用代理程序实例机制通过锁定共享资源来工作。 对共享资源执行锁定的代理程序实例将成为活动实例，而另一个实例 (未能执行锁定) 将成为备用实例。

此处的共享资源是新队列 SYSTEM.FTE.HA.<agent name>。 当配置了 IBM MQ 9.1.4 或更高版本的代理程序时，将自动创建此队列。

过程工作方式

要创建 HA 代理程序，请在两台机器上创建具有相同配置参数的代理程序，方法是使用附加 **-x** 参数以及 agent.properties 文件中设置为 true 的 **highlyAvailable** 代理程序属性来运行 **fteCreateAgent** 或 **fteCreateBridgeAgent** 命令。

注意:

- 这两种配置都必须指向同一代理队列管理器。
- 必须仅在代理队列管理器上创建一次必需的代理队列。

请参阅 **fteCreateAgent** 命令，以获取有关 **-x** 参数和 agent.properties 文件的更多信息，以获取有关 **highlyAvailable** 代理程序属性的更多信息。

注: 运行 **fteCreateAgent** 或 **fteCreateBridgeAgent** 命令将创建一个 MQSC 文件，其中包含在代理队列管理器和 SYSTEM.FTE.HA.*agent name* 队列中创建 IBM MQ 对象所需的脚本。 无论您是否指定了 **-x** 参数，都将创建此 MQSC 文件。

在创建高可用性代理程序配置时， **fteCreateAgent** 或 **fteCreateBridgeAgent** 命令通过预订 SYSTEM.FTE/Agents/*agent name* 主题来检查其他位置是否存在同一代理程序的实例。 如果找到同一代理程序的实例，那么任一命令都会在文件系统上创建所需的配置，但不会再次发布该代理程序的创建。

当代理程序以 HA 方式启动时:

1. 代理程序尝试以互斥 GET 方式打开 SYSTEM.FTE.HA.*agent name* 队列。
2. 如果代理程序成功打开 SYSTEM.FTE.HA.*agent name* 队列，那么这将成为代理程序的 活动实例 ，并且将继续执行进一步的启动过程。
3. 如果尝试以互斥 GET 方式打开 SYSTEM.FTE.HA.*agent name* 队列失败，并返回 MQRC_OBJECT_IN_USE 原因码，那么表示已存在在其他位置运行的代理程序的活动实例。 因此，此实例将成为代理程序的 备用实例 。

   备用实例尝试按指定的时间间隔打开 SYSTEM.FTE.HA.*agent name* 队列。 在 agent.properties 文件中提供了用于此目的的其他代理属性 **standbyPollInterval** 。

   使用缺省值时，备用实例会尝试每 5 秒打开一次 SYSTEM.FTE.HA.*agent name* 队列。 这会重复，直到实例成功打开 SYSTEM.FTE.HA.*agent name* 队列或使用 **fteStopAgent** 命令停止为止。

   V 9.2.0.5 V 9.2.4 从 IBM MQ 9.2.4 和 IBM MQ 9.2.0 Fix Pack 5 开始，所有实例还会使用 **standbyPollInterval** 属性来确定实例从其代理程序队列管理器断开连接后，在两次重新连接尝试之间等待的时间长度。

多个备用实例

所有备用实例都尝试以互斥 GET 方式获取 SYSTEM.FTE.HA.*agent name* 队列，并且在活动实例失败后成功的实例将成为活动实例。

活动实例维护所有已知备用实例的信息，并在代理程序状态发布过程中发布该信息。 **fteShowAgentDetails** 命令，代理程序 GET REST API 响应和 IBM MQ Explorer MFT 插件的输出显示有关所有备用实例的信息。

有关更多信息，请参阅 **fteShowAgentDetails** 命令和代理程序 GET REST API 响应的示例输出。

请参阅 MFT 代理状态消息 ，以获取 XML 格式的代理状态信息示例。

版本需求

活动和备用代理程序必须为 IBM MQ 9.1.4 或更高版本。

注意:

- 在高可用性方式下，无法在 IBM MQ 9.1.4 之前配置或启动 IBM MQ 版本。
- 活动实例和备用实例都必须运行同一版本的代码。

将验证活动实例和备用实例的版本，以确保这两个实例的版本相同。 临时动态队列用于实例之间的通信。 agent.properties 文件中定义的两个代理属性 **dynamicQueuePrefix** 和 **modelQueueName** 生成临时动态队列的名称。

V 9.2.0 *Managed File Transfer* 中高可用性代理程序的必需信息

您需要了解有关在高可用性配置中运行的标准或网桥 MFT 代理的各种类型的信息。 此信息包括代理程序启动所使用的不同方法，如何在日志文件中标识代理程序的实例以及代理程序的状态信息。

启动代理程序

代理程序实例正在其他位置以非 HA 方式运行

如果尝试启动未配置为 HA 代理程序的代理程序的另一个实例，那么将首先进行检查以查看是否可以在 SYSTEM.FTE.HA.*agent name* 队列上获取锁定。

由于其他实例是以非 HA 方式启动的，因此此实例将获取 SYSTEM.FTE.HA.*agent name* 队列上的锁定。代理程序继续初始化，但稍后失败，因为命令队列由另一个实例独占打开。

在这种情况下，以下示例中显示的消息将记录到代理程序的 output0.log 文件中，并且代理程序每 30 秒继续尝试打开命令队列:

```
BFGMQ1045I: 代理的系统队列 SYSTEM.FTE.COMMAND.SRC' 配置为 NOSHARE 或
DEFSOPT (SHARED)。

BFGAG0035W: 代理在尝试打开队列时接收到 MQI 原因码 2042
'SYSTEM.FTE.COMMAND.SRC"
和通道 "MFT_HA_CHN"。  代理程序将每 30 秒重试一次操作。
```

代理程序的实例正在其他位置以 HA 方式运行

如果尝试启动未配置为 HA 代理程序的代理程序的另一个实例，那么将首先进行检查以查看是否可以在 SYSTEM.FTE.HA.*agent name* 队列上获取锁定。

由于另一个实例已作为活动实例运行，因此尝试获取锁定失败。 实例无法启动，以下错误消息将记录到代理程序的 output0.log 文件中:

```
BFGAG0194E: 此代理程序的实例已在其他位置运行。
因此，此实例无法继续，并将结束。
```

Windows 将代理程序作为 Windows 服务启动

在 Windows 上，可以将代理程序作为 Windows 服务启动。

在启动期间， Windows 将以正常方式或 HA 方式启动 MFT 代理程序。 如果代理程序配置为以 HA 方式运行，那么服务将作为活动实例或备用实例运行，具体取决于哪个实例首先获取锁定。

在日志文件中标识代理程序的实例类型

参考消息将写入代理程序的 output0.log 文件以指示实例类型。 当代理程序实例作为活动实例启动时，将写入以下消息:

```
BFGAG0193I: 代理程序已成功初始化为活动实例。
```

当代理程序实例作为备用实例启动时，将写入以下消息:

```
BFGAG0193I: 代理程序已成功初始化为备用实例。
```

代理程序状态更新

由于正在运行同一代理程序的两个实例，因此需要在代理程序状态发布中提供有关这两个实例的信息。

请注意，活动实例是发布这两个实例的状态的实例。

备用实例

发布代理程序状态时，活动实例会检查备用实例发布的时间。

agent.properties 文件中有两个用于此目的的其他属性:

- **standbyStatusExpiry** 是要将备用状态消息放入代理的命令队列的到期时间。 如果代理程序的活动实例在该时间段内未处理此消息，那么此消息将到期。

  缺省情况下， **standbyStatusExipry** 的值为 30 秒。 该消息也是低优先级的 9 消息，允许通过备用状态消息对传输请求进行优先级处理。
- **standbyStatusPublishInterval** 设置备用实例发布其状态的频率。

活动实例

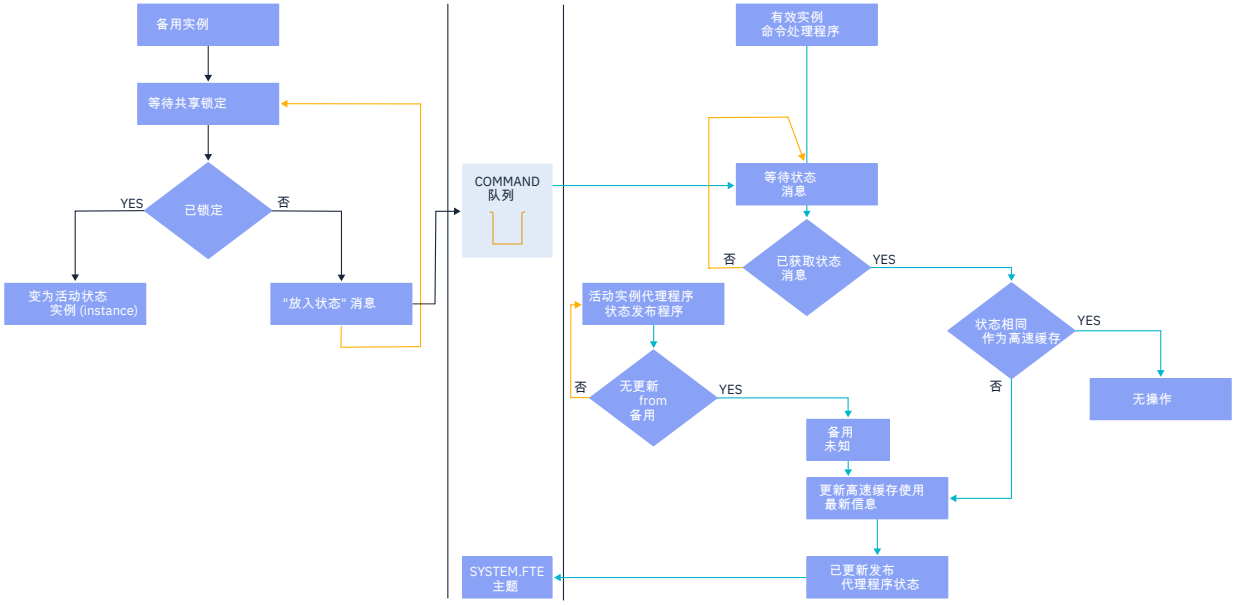
活动实例执行以下操作以处理来自备用实例的状态更新:

1. 从 SYSTEM.FTE.COMMAND.<agent name> 队列获取消息，并将消息处理委派给工作程序线程。

2. 工作程序线程从消息体中检索内容，使用备用实例信息更新代理程序状态对象，并通知代理程序状态发布程序以发布状态。
3. 代理程序状态发布程序将发布状态。

   请注意，此处执行了优化以高速缓存备用状态信息。 发出请求时，代理状态发布程序将检查具有高速缓存状态的新状态，并且仅在存在差异时发布。

下图描述了活动实例或备用实例为发布代理程序状态而遵循的流程:



V 9.2.0 废弃高可用性代理程序中的实例，故障转移和维护

高可用性 Managed File Transfer 实例可能会被废弃，可能会以各种不同的方式发生故障，并且可能需要维护。

正在废弃备用实例状态

可能存在以下情况: 活动实例忙于传输，无法处理备用实例状态消息，或者备用实例已失败，或者由于任何原因未发布状态消息。

在此类情况下，知道存在备用实例的活动代理程序将等待 `agent.properties` 文件中的 **standbyStatusDiscardTime** 属性指定的值，然后再从其列表中除去备用实例。 此属性的缺省值为 600 秒，是 **standbyStatusPublishInterval** 属性的两倍。

正常故障转移实例

必须使用带有 **-i** 选项的 **fteStopAgent** 命令来执行正常故障转移。

这将确保活动实例立即停止。 如果在没有 **-i** 选项的情况下停止代理，那么代理将继续运行，直到活动实例完成所有正在进行的传输为止，因此故障转移可能需要很长时间。

任何进行中的传输都将从最后一个已知检查点恢复。

在其他情况下对实例进行故障转移

如果活动实例以不正常的方式结束，或者整个机器发生故障，那么与代理队列的连接将中断，并且队列管理器将关闭所有打开的队列，包括 SYSTEM.FTE.HA.<agent name> 队列和连接。

因此，备用实例将获取互斥 GET 并完成代理程序初始化的其余部分。

同样，任何进行中的传输都将从最后一个已知检查点恢复。

如果与队列管理器的连接中断

客户机方式

代理进程由多个线程组成。 除缺省线程 (例如，定期发布代理状态的线程) 以外，将使用一组在传输完成后结束的线程来处理每个传输请求。

其中许多线程连接到代理队列管理器并放入和获取消息。 由于网络问题或队列管理器发生故障，这些连接中的任何连接都可能中断。 当任何线程检测到连接中断问题时，该线程会通知主线程启动恢复并结束。

然后，主线程启动另一个线程以等待与正在建立的队列管理器的连接。 重新连接后，将尝试获取代理程序的互斥 GET。 如果成功，那么代理程序将继续完成恢复并成为活动实例。 如果尝试获取互斥 GET 失败，那么实例将成为备用实例。

绑定方式

以绑定方式进行连接时，如果代理程序失去连接，那么代理程序进程将结束。 进程控制器处理代理程序的重新启动。 当代理程序重新启动时，它将完成尝试为其自身获取互斥 GET 的过程。

如果代理程序成功，那么它将成为活动实例; 否则，代理程序将成为备用实例。

应用维护级别升级

将维护应用于高可用性代理程序的步骤类似于针对多实例队列管理器记录的步骤。 有关更多信息，请参阅对 Windows 上的多实例队列管理器应用维护级别更新 或 对 AIX 上的多实例队列管理器应用维护级别更新或对 Linux 上的多实例队列管理器应用维护级别更新。

在应用维护之前，必须停止要应用维护级别的机器上运行的代理程序。 如果要更新活动实例，为了实现传输的连续性，必须将活动实例故障转移到备用实例。

升级完成后，必须启动代理程序实例，将当前活动实例故障转移到已升级的实例，然后升级备用实例。

从较低版本的产品迁移代理程序

从 IBM MQ 9.1.4 之前的 IBM MQ 版本迁移的代理程序作为非高可用性运行。 通过遵循 从较低版本迁移 Managed File Transfer 代理程序中的过程，可以使它们以高可用性方式运行。

配置 MFT 记录器

Managed File Transfer 传输文件时，它将有关其操作的信息发布到协调队列管理器上的主题。 数据库记录器是 Managed File Transfer 的可选组件，您可以使用该组件将此信息复制到数据库以供分析和审计。

有三个版本的记录器:

- ALW 独立文件记录器
- 独立数据库记录器
- Java Platform, Enterprise Edition (Java EE) 记录器

IBM i 上的记录器

IBM i

IBM i 平台上不支持 Managed File Transfer 记录器。

独立文件记录器

ALW

独立文件记录器是一种 Java 进程，它在托管协调队列管理器的系统上运行，或在托管连接到协调队列管理器的队列管理器的系统上运行。 独立文件记录器使用 IBM MQ 绑定来连接到与其关联的队列管理器。 此独立记录器使用 **fteCreateLogger** 命令来创建。

Windows 您可以将独立文件记录器作为 Windows 服务运行，以确保在从 Windows 会话注销时文件记录器继续运行，并且可以将其配置为在系统重新启动时自动启动。 有关更多信息，请参阅第 644 页的『安装 MFT 独立文件记录器』。

在以下平台上不支持独立文件记录器:

- z/OS z/OS
- IBM i IBM i

独立数据库记录器

独立数据库记录器是您在托管队列管理器和数据库的系统上安装的 Java 应用程序。 独立数据库记录器通常安装在与协调队列管理器相同的系统上，但也可以安装在与连接到协调队列管理器的任何队列管理器相同的系统上。 独立数据库记录器使用 IBM MQ 绑定来连接到与其关联的队列管理器，并使用 2 类或 4 类 JDBC 驱动程序连接到 Db2 或 Oracle 数据库。 这些类型的连接是必需的，因为独立数据库记录器使用队列管理器的 XA 支持通过队列管理器和数据库协调全局事务，从而保护数据。

Windows 如果使用的是 Windows 系统，那么可以将独立记录器作为 Windows 服务运行，以确保记录器在您从 Windows 会话注销时继续运行。 有关更多信息，请参阅 第 658 页的『安装 MFT 独立数据库记录器』 以了解独立数据库记录器。

Java EE 数据库记录器

Java EE 数据库记录器作为 EAR 文件提供，您可以将其安装到应用程序服务器中。 如果您有现有 Java EE 应用程序服务器环境可用，那么这可能比使用独立数据库记录器更方便，因为可以将 Java EE 数据库记录器与其他企业应用程序一起管理。 您还可以将 Java EE 数据库记录器安装在与托管 IBM MQ 服务器和数据库的系统不同的系统上。 支持将 Java EE 数据库记录器用于 Db2 和 Oracle 数据库。 安装在 WebSphere Application Server 7.0 上时， Java EE 数据库记录器还支持 Oracle Real Application Clusters。

有关如何配置记录器的指示信息，请参阅以下主题:

- 第 644 页的『安装 MFT 独立文件记录器』
- 第 658 页的『安装 MFT 独立数据库记录器』
- 第 662 页的『为 MFT 安装 Java EE 数据库记录器』

ALW 安装 MFT 独立文件记录器

独立文件记录器是一个 Java 进程，必须使用 IBM MQ 绑定方式或客户机方式连接到协调队列管理器。 要定义独立文件记录器，请使用 **fteCreateLogger** 命令并遵循本主题中的步骤。

关于此任务

有关独立文件记录器的更多信息，请参阅第 643 页的『配置 MFT 记录器』。 本主题中的步骤将记录器配置为连接到协调队列管理器。 有关备用记录器配置，请参阅第 661 页的『MFT 独立记录器的备用配置』

在以下平台上不支持独立文件记录器:

- z/OS z/OS
- IBM i IBM i

过程

1. 请确保您已安装 Managed File Transfer Logger 组件。 有关更多信息，请参阅 Managed File Transfer 产品选项
2. 指定协调队列管理器并将参数 -loggerType 设置为 FILE，然后运行 **fteCreateLogger** 命令以创建独立文件记录器。 有关更多信息，请参阅 fteCreateLogger。

3. 可选： 如果要使用定制格式，那么可以修改由 **fteCreateLogger** 命令创建的 XML 文件。 日志格式定义位于 FileLoggerFormat.xml 文件中。 有关更多信息，请参阅第 645 页的『MFT 独立文件记录器格式』。
4. 针对协调队列管理器运行 **fteCreateLogger** 命令提供的 MQSC 命令，以创建记录器队列。
5. 确定要运行该记录器进程的用户，并为该用户配置许可权。 有关更多信息，请参阅第 658 页的『为 MFT 独立文件记录器配置用户访问权』。
6. 可选： 您可以通过编辑在运行 **fteCreateLogger** 命令时创建的 logger.properties 文件来进一步配置独立文件记录器。 此文件是由键值对组成的 Java 属性文件。 logger.properties 文件位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* 目录中。 有关可用属性及其影响的更多信息，请参阅 MFT 记录器配置属性。
7. Windows
   可选： 如果使用的是 Windows 系统，那么可以将独立文件记录器作为 Windows 服务运行。 运行带 **-s** 参数的 **fteModifyLogger** 命令。 有关更多信息，请参阅 fteModifyLogger。
8. 使用 **fteStartLogger** 命令启动独立文件记录器。 有关更多信息，请参阅 fteStart 记录器。

   如果执行了上一步并在 Windows 上使用了带有 **-s** 参数的 **fteModifyLogger** 命令，那么独立文件记录器将作为 Windows 服务启动。
9. 检查记录器输出。 独立文件记录器生成两种类型的输出，即文件传输审计数据和记录器诊断数据。 文件传输审计数据可在 *MQ_DATA_PATH*/mqft/logs/*coordination_qmgr_name*/loggers/*logger_name*/logs 中找到。 记录器诊断数据可在 *MQ_DATA_PATH*/mqft/logs/*coordination_qmgr_name*/loggers/*logger_name* 中找到
10. 您可以通过使用 **fteStopLogger** 命令来停止该记录器。 有关更多信息，请参阅 fteStopLogger。

结果

ALW *MFT* 独立文件记录器格式

文件记录器写入的消息信息的格式可在 FileLoggerFormat.xml 文件中定义。

记录器的配置目录位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* 中。 创建新的文件记录器时，会创建该文件的一个版本，其中包含文件记录器使用的一组缺省定义。 有关缺省日志格式定义的更多信息，请参阅 MFT 独立文件记录器缺省日志格式。

如果要指定您自己的定制日志格式，请编辑 FileLoggerFormat.xml 文件。

定制的日志格式定义

日志格式定义由一组消息类型组成，每种消息类型都具有格式定义。 消息类型的格式定义由一组以 XPATH 格式提供的插入和用于分隔每个插入的分隔符组成。 对插入的排序决定了以何种顺序来将内容放置在为输出到日志文件而生成的行中。 例如，这是 callStarted 消息类型的定义：

```
<callStarted>
  <format>
    <inserts>
      <insert type="user"   width="19" ignoreNull="false">/transaction/action/
       @time</insert>
      <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6"  ignoreNull="false">type</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/agent/
       @agent</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/agent/@QMgr</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/job/name</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/transferSet/
       call/command/@type</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/transferSet/
      call/command/@name</insert>
      <insert type="system" width="0"  ignoreNull="true">callArguments</insert>
    </inserts>
    <separator></separator>
  </format>
</callStarted>
```

该格式在日志文件中生成类似以下的行：

```
2011-11-25T10:53:04;414d5120514d5f67627468696e6b20206466cf4e20004f02;[CSTR];
  AGENT1;AGENT_QM;Managed Call;executable;echo;call test;
```

在格式定义中提供的 insert 顺序是信息在日志文件行中出现的顺序。 有关定义 `FileLoggerFormat.xml` 文件格式的 XML 模式的更多信息，请参阅 独立文件记录器格式 XSD。

消息类型

FTE 代理将一系列不同的消息类型写入到 SYSTEM.FTE/Log 子主题。 有关更多信息，请参阅 SYSTEM.FTE 主题。 日志文件定义可包含以下类型的消息的格式定义：

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
scheduleSubmitTransferSet
transferStarted
transferCancelled
transferComplete
transferDelete
transferProgress
```

有关 "MFT 文件日志记录器可为每种信息类型记录的属性的更多信息，请参阅 "第 648 页的『MFT 文件日志记录器可为每种信息类型记录的属性』。 这些属性可用作 XPATH 插入，以自定义 "MFT 文件日志记录器的输出。

消息格式会有所不同。 大多数的消息类型都会在日志文件中为 SYSTEM.FTE/Log 子主题中耗用的每条日志消息写入单个行。 这导致了简单的情况，即日志格式定义中提供的 XPATH 地址与消息根相关。 以下是使用该方法来写入输出的消息类型：

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
transferStarted
transferCancelled
transferComplete
transferDelete
```

用于写入日志消息的另一种方法则使用多行来表示日志消息内传输集中的项。 在此情况下，所提供的格式会应用于日志消息内传输集中的每个项。 如果希望包含特定于传输集中每个项的信息，那么需要所提供的 XPATH 来将该项用作其 XPATH 根。 以下是使用该方法来写入输出的消息类型：

```
scheduleSubmitTransferSet
transferProgress
```

将为传输集合中的每个项都写入一行输出。 要对于传输集中所有项固定的信息仍可以使用与日志消息的根相关的 XPATH 地址。 在以下简化的 `transferProgress` 格式定义示例中，固定的是时间戳记和传输标识。 与作为其根的项相关的任何信息对于写入的每一行均会有变化。 在此示例中，写入了每个项的源和目标文件信息。

```
<transferProgress>
      <format>
        <inserts>
```

```
          <insert type="user"   width="19" ignoreNull="false">/transaction/action/
           @time</insert>
          <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="true">status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">source/file |
           source/queue</insert>
          <insert type="user"   width="0"  ignoreNull="false">source/file/@size |
           source/queue/@size</insert>
          <insert type="user"   width="5"  ignoreNull="true">source/@type</insert>
          <insert type="user"   width="6"  ignoreNull="true">source/@disposition</insert>
          <insert type="user"   width="0"  ignoreNull="false">destination/file |
            destination/queue</insert>
          <insert type="user"   width="0"  ignoreNull="false">destination/file/@size |
            destination/queue/@size</insert>
          <insert type="user"   width="5"  ignoreNull="true">destination/@type</insert>
          <insert type="user"   width="9"  ignoreNull="true">destination/@exist</insert>
          <insert type="user"   width="0"  ignoreNull="true">status/supplement</insert>
        </inserts>
        <separator></separator>
      </format>
    </transferProgress>
```

这样会以下列格式生成具有一行或多行的日志文件条目：

```
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPRO];0
  ;/src/test1.file;3575;file;leave ;/dest/test1.file;3575;file;overwrite;;
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPRO];0
  ;/src/test2.file;3575;file;leave ;/dest/test2.file;3575;file;overwrite;;
```

插入格式

定义消息类型的格式时，有两种类型的插入可用：`user` 和 `system`。 插入类型在插入元素的 `type` 属性中定义。 这两种插入类型还可以使用插入元素的 **width** 和 **ignoreNull** 属性来定制其布局。 例如：

```
<insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
```

在该示例中，插入操作会提取从位于 `/transaction/@ID` 的日志消息中找到的信息，并将该信息削减或填充为 48 个字符，然后将其写入日志。 如果 `/transaction/@ID` 的内容为 null，那么插入操作会在将字符串 null 填充为 48 个字符后予以写入，原因是 `ignoreNull` 属性已设置为 `false`。 如果 `ignoreNull` 设置为 `true`，那么会改为写入填充为 48 个字符的空字符串。 设置 width="0 " 表示不裁剪列宽，不表示裁剪宽度为 0。 可以通过此方式使用 `ignoreNull` 属性，以在日志中发现非预期的空值时进行检测。 当调试新的日志文件定义时，这很有用。

用户定义的插入

用户插入包含要在该插入中写入的信息的 XPATH 地址。 该地址引用在 FTE 日志消息中找到的一段信息。 有关日志消息格式的更多信息，请参阅：

- 文件传输日志消息格式
- 已调度的文件传输日志消息格式
- MFT 监视器日志消息格式

系统定义的插入

系统定义的插入包含某个关键字，该关键字引用无法在日志消息中找到或者不便于使用 XPATH 语言进行定义的一段信息。

受支持的系统插入有：

- `type` - 以短格式写日志消息的类型。
- `callArguments` - 以空格分隔的格式写一组提供给受管调用的自变量。
- `transferMetaData` - 以逗号分隔的 *key=value* 格式编写为传输定义的元数据条目集。

下表列出了每种消息类型的系统定义插入的“type”值。

表 *51:* 受支持消息类型及其*"type"*系统插入的摘要。	
消息类型	**"type"系统插入的值**
callCompleted	[CCOM]
callStarted	[CSTR]
monitorAction	[MACT]
monitorCreate	[MCRT]
monitorFired	[MFIR]
notAuthorized	[AUTH]
scheduleDelete	[SDEL]
scheduleExpire	[SEXP]
scheduleSkipped	[SSKP]
scheduleSubmitInfo	[SSIN]
scheduleSubmitTransfer	[SSTR]
scheduleSubmitTransferSet	[SSTS]
transferStarted	[TSTR]
transferCancelled	[TCAN]
transferComplete	[TCOM]
transferDelete	[TDEL]
transferProgress	[TPRO]

MFT 文件日志记录器可为每种信息类型记录的属性
您可以使用 "Managed File Transfer（MFT）文件记录仪可记录的每种信息类型的属性作为 XPATH 插入，自定义 "MFT 文件记录仪的输出。

信息类型：callCompleted

表 *52: callCompleted* 的属性	
属性名称	**描述**
/transaction/action/@time	生成此事件的日期和时间。
/transaction/@ID	托管呼叫标识符。
/transaction/status/@resultCode	托管调用的结果。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 CCOM。
/transaction/agent/@agent	提交托管呼叫的代理。

表 *52: callCompleted* 的属性 *(继续)*	
属性名称	**描述**
/transaction/agent/@QMgr	代理队列管理器。
/transaction/job/name	作业名
/transaction/transferSet/call/command/@type	命令类型（可执行、antscript 或 jcl）。
/transaction/transferSet/call/command/@name	托管调用执行的命令名称。
callArguments	传给命令的参数。
/transaction/transferSet/call/callResult/@outcome	命令执行的结果 - 成功、失败或错误。
/transaction/transferSet/call/callResult/result/error	有关错误的详细信息。

信息类型：callStarted

表 *53: callStarted* 的属性	
属性名称	**描述**
/transaction/action/@time	生成此事件的日期和时间。
/transaction/@ID	托管呼叫标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 CSTR。
/transaction/agent/@agent	提交托管呼叫的代理。
/transaction/agent/@QMgr	代理队列管理器。
/transaction/job/name	作业名。
/transaction/transferSet/call/command/@type	命令类型（可执行、antscript 或 jcl）。
/transaction/transferSet/call/command/@name	要执行的命令名称。
callArguments	传给命令的参数列表。

信息类型：monitorAction

表 *54: monitorAction* 的属性	
属性名称	**描述**
/monitorLog/action/@time	生成此事件的日期和时间。
/monitorLog/@referenceId	资源监控器标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 MACT。
/monitorLog/status/@resultCode	资源监控操作的结果。
/monitorLog/@monitorName	资源监控器的名称。
/monitorLog/monitorAgent/@agent	运行资源监控器的代理的名称。
/monitorLog/monitorAgent/@QMgr	代理队列管理器。
/monitorLog/action	资源监控操作类型--创建、删除、启动、停止、triggerSatisfied,、triggerNotSatisfied, triggerFail。

信息类型：monitorCreate

表 *55: monitorCreate* 的属性	
属性名称	**描述**
/monitorLog/action/@time	生成此事件的日期和时间。
/monitorLog/@referenceId	资源监控器标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 MCRT。
/monitorLog/@monitorName	资源监控器的名称。
/monitorLog/monitorAgent/@agent	创建资源监控器的代理名称。
/monitorLog/monitorAgent/@QMgr	代理队列管理器的名称。
/monitorLog/action	资源监控器创建操作。

信息类型：monitorFired

表 *56: monitorFired* 的属性	
属性名称	**描述**
/monitorLog/action/@time	生成此事件的日期和时间。
/monitorLog/@referenceId	资源监控器标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 MFIR。
/monitorLog/status/@resultCode	资源监控器启动操作的结果。
/monitorLog/@monitorName	资源监控器的名称。
/monitorLog/monitorAgent/@agent	运行资源监控器的代理的名称。
/monitorLog/monitorAgent/@QMgr	代理队列管理器的名称。
/monitorLog/action	资源监控器启动。
/monitorLog/references/taskRequest	该资源监控器提交的传输任务请求的详细信息。

信息类型：notAuthorized

表 *57: notAuthorized* 的属性	
属性名称	**描述**
/notAuthorized/action/@time	生成此事件的日期和时间。
/notAuthorized/@ID	事件的标识符
类型	以简短格式表示的日志信息类型。 该信息类型的值为 AUTH。
/notAuthorized/status/@resultCode	事件中提供的结果代码。
/notAuthorized/action	未授权。
/notAuthorized/authority	执行一项行动所需的授权。
/notAuthorized/originator/userID	提交请求的用户 ID。

表 *57: notAuthorized* 的属性 *(继续)*	
属性名称	**描述**
/notAuthorized/originator/mqmdUserID	MQ 用户 ID。
/notAuthorized/status/supplement	关于未授权活动的更多信息。

信息类型：scheduleDelete

表 *58: scheduleDelete* 的属性	
属性名称	**描述**
/schedulelog/action/@time	生成此事件的日期和时间。
/schedulelog/@ID	计划转移的标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 SDEL。
/schedulelog/status/@resultCode	预定转账删除操作的结果。
/schedulelog/sourceAgent/@agent	提交行动的代理机构名称。
/schedulelog/action	删除。
/schedulelog/originator/userID	提交计划传输删除请求的用户。
/schedulelog/status/supplement	关于预定转账删除请求结果的补充信息

信息类型：scheduleExpire

表 *59: scheduleExpire* 的属性	
属性名称	**描述**
/schedulelog/action/@time	生成此事件的日期和时间。
/schedulelog/@ID	计划转移的标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 SEXP。
/schedulelog/status/@resultCode	预定转账过期操作的结果代码。
/schedulelog/sourceAgent/@agent	提交行动的代理机构名称。
/schedulelog/action	到期
/schedulelog/originator/userID	创建计划传输的用户 ID。
/schedulelog/status/supplement	有关预定转学申请过期结果的补充信息。

信息类型：scheduleSkipped

表 *60: scheduleSkipped* 的属性	
属性名称	**描述**
/schedulelog/action/@time	生成此事件的日期和时间。
/schedulelog/@ID	计划转移的标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 SSKP。

表 *60: scheduleSkipped* 的属性 *(继续)*	
属性名称	**描述**
/schedulelog/status/@resultCode	计划转移跳过操作的结果。
/schedulelog/sourceAgent/@agent	提交行动的代理机构名称。
/schedulelog/action	跳过的时间表
/schedulelog/originator/userID	创建计划传输的用户 ID。
/schedulelog/status/supplement	有关计划跳转请求结果的补充信息。

信息类型：scheduleSubmitInfo

表 *61: scheduleSubmitInfo* 的属性	
属性名称	**描述**
/schedulelog/action/@time	生成此事件的日期和时间。
/schedulelog/@ID	计划转移的标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 SSIN。
/schedulelog/status/@resultCode	计划转移的提交结果。
/schedulelog/sourceAgent/@agent	提交行动的代理机构名称。
/schedulelog/action	提交转账申请。
/schedulelog/originator/userID	创建计划传输的用户 ID。
/schedulelog/schedule/submit/@timezone	提交传输请求时使用的时区。
/schedulelog/schedule/repeat/frequency/@interval	计划传输的时间间隔。
/schedulelog/schedule/repeat/expireCount	计划传输的次数。
/schedulelog/status/supplement	有关计划跳转请求结果的补充信息。

信息类型：scheduleSubmitTransfer

表 *62: scheduleSubmitTransfer* 的属性	
属性名称	**描述**
/schedulelog/action/@time	生成此事件的日期和时间。
/schedulelog/@ID	计划转移的标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 SSTR。
/schedulelog/sourceAgent/@agent	转移请求的源代理名称。
/schedulelog/sourceAgent/@QMgr	源代理队列管理器的名称。
/schedulelog/destinationAgent/@agent	转移请求的目的地代理名称。
/schedulelog/destinationAgent/@QMgr	目标代理队列管理器的名称。

信息类型：scheduleSubmitTransferSet

表 *63: scheduleSubmitTransferSet* 的属性	
属性名称	**描述**
/schedulelog/action/@time	生成此事件的日期和时间。
/schedulelog/@ID	计划转移的标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 SSTS。
源文件	传输请求中源文件的名称。
源/队列	传输请求中队列的名称。 源类型为队列。
source/@type	源类型
source/@disposition	传输完成后删除或保留源文件。
目的地/文件	目标文件的名称。
目的地/队列	目标队列的名称。 目的地类型为队列。
destination/@type	目标类型
destination/@exist	如果目标文件已经存在，应采取的措施。 重写或错误。

信息类型：transferStarted

表 *64: transferStarted* 属性	
属性名称	**描述**
/transaction/action/@time	生成此事件的日期和时间。
/transaction/@ID	转账的标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 TSTR。
/transaction/status/@resultCode	结果代码
/transaction/sourceAgent/@agent	源代理名称。
/transaction/sourceAgent/@QMgr	源代理队列管理器的名称。
/transaction/sourceAgent/@agentType	代理类型 - 标准或桥梁。
/transaction/destinationAgent/@agent	目的地代理的名称。
/transaction/destinationAgent/@QMgr	目标代理队列管理器的名称。
/transaction/originator/userID	提交转账请求的用户 ID。
/transaction/job/name	作业名
/transaction/scheduleLog/@ID	触发此转移请求的计划转移的标识符。
transferMetaData	与传输请求相关的元数据。

信息类型：transferCancelled

表 *65: transferCancelled* 属性	
属性名称	**描述**
/transaction/action/@time	生成此事件的日期和时间。
/transaction/@ID	转账的标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 TCAN。
/transaction/status/@resultCode	结果代码
/transaction/sourceAgent/@agent	源代理名称。
/transaction/sourceAgent/@QMgr	源代理队列管理器的名称。
/transaction/sourceAgent/@agentType	代理类型 - 标准或桥梁。
/transaction/destinationAgent/@agent	目的地代理的名称。
/transaction/destinationAgent/@QMgr	目标代理队列管理器的名称。
/transaction/destinationAgent/@agentType	目的地代理类型 - 标准或网桥。
/transaction/originator/userID	提交转账请求的用户 ID。
/transaction/job/name	作业名。
/transaction/status/supplement	有关活动的更多信息。

信息类型：transferComplete

表 *66: transferComplete* 属性	
属性名称	**描述**
/transaction/action/@time	生成此事件的日期和时间。
/transaction/@ID	转账的标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 TCOM。
/transaction/status/@resultCode	结果代码
/transaction/sourceAgent/@agent	源代理名称。
/transaction/sourceAgent/@QMgr	源代理队列管理器的名称。
/transaction/sourceAgent/@agentType	源代理类型。
/transaction/destinationAgent/@agent	目的地代理的名称。
/transaction/destinationAgent/@QMgr	目标代理队列管理器的名称。
/transaction/destinationAgent/@agentType	目的地代理类型。
/transaction/originator/userID	提交转账请求的用户 ID。
/transaction/job/name	作业名。
/transaction/status/supplement	有关转学完成情况的更多信息。
transferMetaData	与传输请求相关的元数据。

信息类型：transferDelete

表 *67: transferDelete* 属性	
属性名称	**描述**
/transaction/action/@time	生成此事件的日期和时间。
/transaction/@ID	转账的标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 TDEL。
/transaction/status/@resultCode	结果代码
/transaction/sourceAgent/@agent	源代理名称。
/transaction/sourceAgent/@QMgr	源代理队列管理器的名称。
/transaction/sourceAgent/@agentType	源代理类型 - 标准或网桥。
/transaction/destinationAgent/@agent	目的地代理的名称。
/transaction/destinationAgent/@QMgr	目标代理队列管理器的名称。
/transaction/destinationAgent/@agentType	目的地代理类型 - 标准或网桥。
/transaction/originator/userID	提交转账请求的用户 ID。
/transaction/job/name	作业名。
/transaction/status/supplement	活动的其他信息。

信息类型：transferProgress

表 *68: transferProgress* 属性	
属性名称	**描述**
/transaction/action/@time	生成此事件的日期和时间。
/transaction/@ID	转账的标识符。
类型	以简短格式表示的日志信息类型。 该信息类型的值为 TPRO。
status/@resultCode	该项目的转账结果代码。
源文件	源文件的名称。
源/队列	源队列的名称。
source/file/@size	源文件的大小。
source/queue/@size	信息的大小。 始终设为 "空"。
source/@type	源类型
source/@disposition	传输完成后要执行的操作 - 删除或保留源文件。
source/file/@alias	当使用 "Connect:Direct 网桥从 "Connect:Direct 节点向 "Managed File Transfer 代理传输数据时，"Connect:Direct 传输名称。
source/queue/@alias	如果源类型为队列，则别名队列名称。

表 68: *transferProgress* 属性 (继续)	
属性名称	**描述**
source/@correlationBoolean1	表示使用 Connect:Direct 桥接器从 "Connect:Direct 节点到 "Managed File Transfer 代理的传输是否调用用户定义的 Connect:Direct 进程的标志。
source/@correlationNum1	使用 "Connect:Direct 网桥从 "Connect:Direct 节点向 "Managed File Transfer 代理传输时的 "Connect:Direct 进程标识符。
source/@correlationString1	使用 "Connect:Direct 桥从 "Connect:Direct 节点向 "Managed File Transfer 代理传输时，"Connect:Direct 进程的名称。
目的地/文件	目标文件的名称。
目的地/队列	目标队列的名称。
destination/file/@size	目的地文件大小。
destination/queue/@size	目的地报文大小 - 始终设为 "空"。
destination/@type	目标类型。
destination/@exist	目的地存在时要执行的操作 - 覆写或出错。
destination/file/@alias	当使用 "Connect:Direct 桥接器从 "Managed File Transfer 代理向 "Connect:Direct 节点转移时，"Connect:Direct 转移名称。
destination/queue/@alias	目的地的别名队列名称。 仅当目的地类型为队列时有效。
destination/file/@truncateRecords	截断长于 LRECL 数据集属性的目标记录。 仅在目标类型为数据集时有效。
destination/@correlationBoolean1	一个标志，表示使用 "Connect:Direct 桥接器从 "Managed File Transfer 代理到 "Connect:Direct 节点的传输是否调用用户定义的 "Connect:Direct 进程。
destination/@correlationNum1	使用 "Connect:Direct 网桥从 "Managed File Transfer 代理向 "Connect:Direct 节点传输时的 "Connect:Direct 进程标识符。
destination/@correlationString1	使用 "Connect:Direct 桥从 "Managed File Transfer 代理向 "Connect:Direct 节点传输时，"Connect:Direct 进程的名称。
状态/补充	有关活动的更多信息。

ALW 从 *MFT* 独立文件记录器中排除消息类型

如果希望从文件记录器输出中排除特定消息类型，那么可以使用空消息类型元素。

示例

例如，以下格式定义阻止了文件记录器输出 `transferProgress` 消息。

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
 xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferProgress></transferProgress>
```

```
  </messageTypes>
</logFormatDefinition>
```

ALW 为 *MFT* 独立文件记录器定义定制格式

可以在日志格式定义内对定制消息类型子集进行定义以减少定制日志文件格式所需的配置量。

关于此任务

如果 messageTypes 元素未包含在 FileLoggerFormat.xml 文件中，那么该消息类型的格式使用缺省格式。 您只需指定希望使之与缺省格式不同的格式。

示例

在该示例中，格式定义使用此仅输出已启动传输的用户的简化版本来替换 transferStarted 消息类型的缺省格式。 所有其他消息类型都使用缺省格式，因为它们未包含在该日志格式定义中：

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
 xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferStarted>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/transaction/action/
           @time</insert>
          <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/originator/
           userID</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </transferStarted>
  </messageTypes>
</logFormatDefinition>
```

相关参考

MFT 独立文件记录器缺省日志格式

独立文件记录器格式 XSD

ALW 减少 *MFT* 独立文件记录器中的重复消息

重复的日志消息可能会出现在独立文件记录器的日志中。 通过使用 logger.properties 文件，您可以对独立文件记录器进行调整并减少重复消息的数量。

文件记录器日志中的重复消息

在发生故障的情况下，可能会将日志消息写入到独立文件记录器的日志中，而不会使用向 IBM MQ 落实的 SYSTEM.FTE/Log# 主题中的日志消息。 如果发生此情况，那么当独立文件记录器重新启动时，它将再次检索同一消息，并将该消息再次写入日志文件。 请计划在手动查看日志文件或者自动处理日志文件时应对出现这些重复消息的可能性。 为了帮助检测重复消息，独立文件记录器在启动时会将以下消息输出到日志文件：

```
BFGDB0054I: The file logger has successfully started
```

重复消息始终在独立文件记录器启动的时间左右出现，因为这是处理先前实例失败之前的最后一条消息的时间。 通过了解新实例已在何时启动，您可以检测出是否应预期会出现重复消息，以及是否需要处理重复消息。

减少重复消息的数量

独立文件记录器会将其处理的日志消息分组到事务中以提高性能。 此批次大小是您在发生故障的情况下可能看到的重复消息的最大数目。 要减少重复消息的数量，您可以在 `logger.properties` 文件中对以下属性进行调整：

```
wmqfte.max.transaction.messages
```

例如，通过将此项设置为 1，复制的最大消息数将减少到 1。 请注意，修改此值会影响独立文件记录器的性能，因此需要进行彻底测试以确保不会对系统造成负面影响。

`logger.properties` 文件位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* 目录中。 有关可用属性及其影响的更多信息，请参阅 MFT 记录器配置属性

ALW 为 *MFT* 独立文件记录器配置用户访问权

在测试环境中，您可以向常规用户帐户添加所需的任何新特权。 在生产环境中，建议您创建具有执行作业所需的最低许可权的新用户。

关于此任务

您必须在同一个系统上安装独立文件记录器和 IBM MQ。 请按照以下方式配置该用户的许可权：

过程

1. 确保用户具有对作为 Managed File Transfer 安装一部分所安装的文件进行读和（必要时）执行的许可权。
2. 确保该用户有权对 `logs` 目录（位于配置目录中）中的任何文件进行创建和写入。 该目录用于事件日志，如有必要，还用于诊断跟踪和“首次故障数据捕获”(FFDC) 文件。
3. 确保该用户具有其自己的组，并且还不在对协调队列管理器具有广泛许可权的任何组中。 该用户不应在 mqm 组中。 在某些平台上，还会自动为 staff 组授予队列管理器访问权；独立文件记录器用户不应位于 staff 组中。 您可以使用 IBM MQ Explorer 来查看队列管理器本身及其包含的对象的权限记录。 右键单击该对象，然后选择**对象权限 > 管理权限记录**。 在命令行上，可以使用命令 dspmqaut（显示权限）或 dmpmqaut（转储权限）。
4. 使用 IBM MQ Explorer 或 setmqaut (grant 或 revoke authority) 命令中的 " **管理权限记录** " 窗口来添加用户自己的组的权限 (在 AIX 上， IBM MQ 权限仅与组关联，而不与个人用户关联)。 所需权限如下所示：
   - 对队列管理器的 Connect 和 Inquire 许可权（IBM MQ Java 库需要 Inquire 许可权才能运行）。
   - 对 SYSTEM.FTE 主题的 SUBSCRIBE 许可权。
   - 对 SYSTEM.FTE.LOG.RJCT.*logger_name* 队列的 PUT 许可权。
   - 对 SYSTEM.FTE.LOG.CMD.*logger_name* 队列的 Get 许可权。

   给定的 REJECT 和 COMMAND 队列名称是缺省名称。 如果您在配置独立文件记录器队列时选择了不同的队列名称，请改为向这些队列名称添加许可权。

安装 MFT 独立数据库记录器

完成以下步骤以安装和配置独立数据库记录器。

关于此任务

要点: IBM i 平台上不支持 Managed File Transfer 记录器。

有关独立数据库记录器的更多信息，请参阅第 643 页的『配置 MFT 记录器』。

注: 在任何一个时间都不能针对数据库中的同一模式运行多个数据库记录器（无论独立还是 Java EE 类型）。 如果尝试执行此操作，那么在尝试向数据库写入传输日志数据时将导致冲突。

过程

1. 使用数据库的文档指导您安装数据库软件。

如果 JDBC 支持是您数据库的可选组件，那么必须安装此组件。

2. 将参数 **-loggerType** 设置为 DATABASE，然后运行 **fteCreateLogger** 命令以创建独立数据库记录器。 更多信息，请参阅 fteCreateLogger。

   缺省模式名称为 FTELOG。 如果使用 FTELOG 之外的模式名称，必须先编辑提供的 SQL 文件（根据您的数据库，为 `ftelog_tables_db2.sql` 或 `ftelog_tables_oracle.sql`）以反映此模式名称，然后继续下一步骤。 有关更多信息，请参阅 MFT 记录器配置属性中的 wmqfte.database.schema 。

3. 使用您数据库的工具创建所需的数据库表。

   Multi 在多平台上，`ftelog_tables_db2.sql` 和 `ftelog_tables_oracle.sql` 文件包含可运行来创建表的 SQL 命令。

   z/OS 在 z/OS 上，需要运行的文件取决于使用的 Db2 for z/OS 版本：

   - 对于 Db2 for z/OS 9.0 以及更低版本，运行文件 `ftelog_tables_zos.sql` 以创建表。 此文件在创建表时，将 INTEGER 数据类型用于表示所传输文件大小的字段以及表示与每个传输关联的表标识的字段。
   - 对于 Db2 for z/OS 9.1 以及更高版本，运行文件 `ftelog_tables_zos_bigint.sql` 以创建表。 此文件在创建表时，将 BIGINT 数据类型用于表示所传输文件大小的字段以及表示与每个传输关联的表标识的字段。

4. 如果使用的是 "Oracle 数据库，则在 "`listener.ora` 文件中添加 "USE_SID_AS_SERVICE_*listener_name*=on，其中 "*listener_name* 是数据库服务器上监听器的名称。 然后重新启动监听器。

   该参数指定数据库连接将 "Oracle 系统标识符 (SID) 作为数据库的服务名称。 需要使用该参数的原因是，"MFT 数据库日志记录器仅使用 SID 连接 "Oracle 数据库，但后来的 "Oracle 数据库（从 12c 版本开始）使用服务名称。 更多信息，请参阅 "Oracle 版本的文档，例如 "Oracle Net Listener Parameters in the listener.ora File。 如果不修改 "`listener.ora` 文件，运行记录仪时可能会看到以下信息：

   ```
   ORA-12505, TNS:listener does not currently know of SID given in connect descriptor
   ```

5. 针对您的记录器命令队列管理器运行 **fteCreateLogger** 命令所提供的 MQSC 命令以创建记录器队列。 独立数据库记录器使用协调队列管理器上的两个队列。 第一个队列为命令队列，其中将放置用于控制独立数据库记录器的操作的消息。 此命令队列的缺省名称是 SYSTEM.FTE.LOG.CMD.*logger_name*。 第二个队列为拒绝队列。 由于独立数据库记录器从不废弃日志消息，因此如果记录器遇到无法处理的消息，它会将该消息放置到拒绝队列上以进行检查，并可能进行重新处理。 建议您不要将队列管理器的死信队列用于此目的，因为拒绝的消息没有 DLH 头，并且拒绝的消息不应与由于其他原因放置到死信队列中的消息组合在一起。 拒绝队列的缺省名称是 SYSTEM.FTE.LOG.RJCT.*logger_name*。 这两个队列在 **fteCreateLogger** 命令生成的 MQSC 脚本文件中定义。

6. 选择用户并配置许可权

7. 可选： 您可以通过编辑步骤 第 659 页的『2』中 **fteCreateLogger** 命令创建的 `logger.properties` 文件来进一步配置独立数据库记录器。 此文件是由键值对组成的 Java 属性文件。 `logger.properties` 文件位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* 目录中。 有关可用属性及其影响的更多信息，请参阅 MFT 记录器配置属性。

8. Windows

   可选： 如果使用的是 Windows 系统，那么可以将独立数据库记录器作为 Windows 服务运行。 运行带 **-s** 参数的 **fteModifyLogger** 命令。 更多信息，请参阅 fteModifyLogger。

9. 可选： 如果使用的数据库是 Oracle，或者是远程连接到 Db2 数据库，那么将需要指定记录器将用于向数据库服务器认证的用户名和密码。 此用户名和密码在符合 `MQMFTCredentials.xsd` 模式所定义的格式的凭证文件中指定。 有关更多信息，请参阅 MFT 凭证文件格式。 创建凭证文件后，必须使用 `wmqfte.database.credentials.file` 属性在 `logger.properties` 文件中指定该凭证文件的位置。

10. 使用 **fteStartLogger** 命令启动独立数据库记录器。 缺省情况下，独立数据库记录器在后台运行，并且独立数据库记录器会将输出放置到 `logs` 目录中的文件。 如果要在前台运行独立数据库记录器，并将输出生成到控制台以及日志文件，请将 **-F** 参数添加到 **fteStartLogger** 命令。

如果执行了上一步并在 Windows 上使用了带有 **-s** 参数的 **fteModifyLogger** 命令，那么独立数据库记录器将作为 Windows 服务启动。

将 *MFT* 与远程数据库配合使用

您可以使用 Managed File Transfer 记录器与远程系统上的数据库通信。

关于此任务

如果数据库与 Managed File Transfer 安装在不同机器上，请完成以下步骤。 这些步骤对于 Db2 和 Oracle 均适用，除非另有声明。

过程

1. 将数据库客户机安装在安装有 Managed File Transfer 的系统上。
2. 将远程数据库服务器添加到本地数据库客户机配置中。 Managed File Transfer 和 IBM MQ 需要此配置更新才能正确访问数据库。
3. 在 `logger.properties` 文件中指定新属性，以连接到使用凭证文件 **wmfte.database.credentials.file** 的数据库。

   注: Managed File Transfer 的较低版本使用属性 **wmqfte.oracle.user** 或 **wmqfte.database.user**，以及 **wmqfte.oracle.password** 或 **wmqfte.database.password**。现在不推荐使用这些属性。 推荐使用 **wmfte.database.credentials.file**。
4. **仅限 Oracle：** 为了能够远程连接数据库，可将协调队列管理器的 `qm.ini` 文件中的 XAResourceManager 节更改为以下内容（确保更改数据库名称、用户名和用户密码以匹配您自己的信息）：

   `Oracle_XA+Acc=P/ftelog/qgw783jhT+SesTm=35+DB=FTEAUDIT1`**`+SqlNet=FTEAUDIT1`**`+threads=false`，更改将以粗体突出显示。
5. **仅限 Oracle：** 在 `logger.properties` 文件中，使用 **wmqfte.oracle.host** 和 **wmqfte.oracle.port** 属性指定主机和端口。 主机和端口的缺省值允许您使用本地数据库客户机，因此如果先前使用了本地数据库，那么您可能没有设置这些值。

相关参考

MFT 记录器配置属性

为 *MFT* 独立数据库记录器配置用户访问权

在测试环境中，您可以向常规用户帐户添加所需的任何新特权。 在生产环境中，建议您创建具有执行作业所需的最低许可权的新用户。

关于此任务

运行独立数据库记录器所需的用户帐户的数量和类型取决于所用的系统数。 您可以在单个系统上或跨两个系统安装独立数据库记录器、IBM MQ 和数据库。 独立数据库记录器必须和 IBM MQ 位于同一个系统上。 可以在以下拓扑中安装这些组件：

独立数据库记录器、IBM MQ 和数据库全都位于同一个系统上
: 您可以定义单个操作系统用户，以用于所有这三个组件。 这是适合独立数据库记录器的配置。 独立数据库记录器使用绑定方式连接到 IBM MQ，并使用本机连接来连接到数据库。

独立数据库记录器和 IBM MQ 位于一个系统上，而数据库位于另一个系统上
: 您为该配置创建两个用户：运行独立数据库记录器的系统上的一个操作系统用户，以及对数据库服务器上的数据库具有远程访问权的一个操作系统用户。 这是适合使用远程数据库的独立数据库记录器的配置。 独立数据库记录器使用绑定方式连接到 IBM MQ，并使用客户机连接来访问数据库。

例如，这些指示信息的其余部分假定用户名为 `ftelog`，但您也可以使用任何用户名。 请按照以下方式配置该用户的许可权：

过程

1. 确保该用户有权读取并在必要时可以执行“Managed File Transfer 远程工具和文档”安装过程中所安装的文件。
2. 确保该用户有权创建和写入 `logs` 目录（位于配置目录中）中的任何文件。 该目录用于事件日志，如有必要，还可用于诊断跟踪和 FFDC 文件。
3. 确保该用户具有自己的组，并且未同时处于对协调队列管理器具有广泛许可权的任何组中。 该用户不应在 mqm 组中。 在某些平台上，还会自动为 staff 组授予队列管理器访问权；独立数据库记录器用户不应位于 staff 组中。 您可以使用 IBM MQ Explorer 来查看队列管理器本身及其中对象的权限记录。 右键单击该对象，然后选择**对象权限 > 管理权限记录**。 在命令行上，可以使用命令 dspmqaut（显示权限）或 dmpmqaut（转储权限）。
4. 使用 IBM MQ Explorer 或 setmqaut (grant 或 revoke authority) 命令中的 " **管理权限记录** " 窗口来添加用户自己的组的权限 (在 AIX 上， IBM MQ 权限仅与组关联，而不与个人用户关联)。 所需权限如下所示：
   - 对队列管理器的 Connect 和 Inquire 许可权（IBM MQ Java 库需要 Inquire 许可权才能运行）。
   - 对 SYSTEM.FTE 主题的 SUBSCRIBE 许可权。
   - 对 SYSTEM.FTE.LOG.RJCT.*logger_name* 队列的 PUT 许可权。
   - 对 SYSTEM.FTE.LOG.CMD.*logger_name* 队列的 Get 许可权。

   给定的 REJECT 和 COMMAND 队列名称是缺省名称。 如果您在配置独立数据库记录器队列时选择了不同的队列名称，请改为向这些队列名称添加许可权。
5. 执行特定于所用数据库的用户配置。
   - 如果您的数据库是 Db2，请执行以下步骤：

     有几种机制可用于管理 Db2 的数据库用户。 这些指示信息适用于基于操作系统用户的缺省方案。
     - 确保 `ftelog` 用户未处于任何 Db2 管理组（例如，db2iadm1、db2fadm1 或 dasadm1）中
     - 为用户提供连接到数据库的许可权，以及对步骤 2：创建所需的数据库表过程中所创建表的选择、插入和更新许可权
   - 如果您的数据库是 Oracle，请执行以下步骤：
     - 确保 `ftelog` 用户未处于任何 Oracle 管理组（例如，Windows 上的 ora_dba 或 AIX and Linux 上的 dba）中
     - 为用户提供连接到数据库的许可权，以及对步骤 2：创建所需的数据库表过程中所创建表的选择、插入和更新许可权

MFT 独立记录器的备用配置

通常，不管 Managed File Transfer 独立记录器是文件还是数据库类型，它都与协调队列管理器位于同一个系统上，并以 IBM MQ 绑定方式连接到协调队列管理器。 但是，它也可以安装在与连接到协调队列管理器的任何队列管理器相同的系统上。 独立记录器使用其自动创建的预订接收消息。 这是在安装指示信息中描述的配置。

但是，如果有特定于位置方面的考虑，那么可以将独立记录器配置为以其他两种由 wmqfte.message.source.type 属性控制的方法来接收消息。 此属性在 MFT 记录器配置属性中进行了描述。

管理预订

缺省情况下，独立记录器使用缺省持久预订选项和受管预订，针对 SYSTEM.FTE/Log/# 主题创建自己的预订（即，队列管理器控制用于在消息传递到应用程序前保存消息的支持队列）。 如果预订或队列需要其他选项，那么您可以自行创建预订，设置所需的选项并将独立记录器配置为使用该预订。 请记住，为独立记录器添加许可权以使用您创建的预订。

使用此配置的一个示例是通过使用两个通配符预订对日志空间进行分区，将名称以 FINANCE 开头的代理中的日志发送到一个数据库，并将名称以 ACCOUNTING 开头的代理中的日志发送到另一个数据库。 这种配置需要两个独立记录器实例，每个实例都有各自针对所需预订的 `logger.properties` 文件， 以及各自的命令队列和拒绝队列。

要仅从名称以 ACCOUNTING 开头的代理收集日志消息，可在协调队列管理器上创建主题字符串为 SYSTEM.FTE/Log/ACCOUNTING* 的预订对象。 将**通配符用法**值设置为**字符级通配符**。 还必须将条目添加到记录器的 `logger.properties` 文件。 例如，如果使用这些设置创建名为 ACCOUNTING.LOGS 的预订对象，请将以下条目添加到 `logger.properties` 文件：

```
wmqfte.message.source.type=administrative subscription
wmqfte.message.source.name=ACCOUNTING.LOGS
```

独立记录器仅处理以主题字符串 SYSTEM.FTE/Log/ 开头的日志消息。 您可以指定限制更多的主题字符串，但不能指定限制更少的字符串。 如果您错误地指定了限制更少的字符串，那么与 SYSTEM.FTE/Log/ 之外的主题字符串相关的所有发布都将转至拒绝队列，并且独立记录器会产生错误消息 BFGDB0002E。 该错误消息表明独立记录器配置有问题。

队列

典型的拓扑是独立记录器与协调队列管理器在同一个系统上运行。 如果这不可能，您可以通过将另一队列管理器上的队列作为预订目标而在协调队列管理器上创建一个预订（使用远程队列定义或使用预订的 DESTQMGR 属性）。 然后，记录器可以在托管第二个队列管理器的系统上运行，并从队列中读取消息。 为确保事务的完整性，独立记录器必须始终以绑定方式连接到其队列管理器。 您必须在独立记录器所连接的同一队列管理器上定义拒绝队列和命令队列。 队列管理器必须为 IBM WebSphere MQ 7.5 或更高版本。

例如，要通过预订收集正放入队列 USER.QUEUE 的日志消息，可将以下条目添加到 `logger.properties` 文件：

```
wmqfte.message.source.type=queue
wmqfte.message.source.name=USER.QUEUE
```

为 MFT 安装 Java EE 数据库记录器

按照以下指示信息安装并配置 JEE 数据库记录器以配合 Managed File Transfer 一起使用。

关于此任务

有关 Java EE 数据库记录器的更多信息，请参阅主题第 643 页的『配置 MFT 记录器』。

注: 不能同时运行 Java EE 数据库记录器和独立记录器，除非这些记录器使用单独的数据库实例。

过程

1. 安装 Java EE 数据库记录器之前，必须准备好环境。 使用主题第 663 页的『准备安装 MFT 的 Java EE 数据库记录器』中的指示信息。
2. 将 Java EE 数据库记录器安装在符合 Java Platform, Enterprise Edition (Java EE) 的应用程序服务器中。 有关指示信息，请参阅第 665 页的『使用 WebSphere Application Server traditional 9.0 安装 MFT 的 Java EE 数据库记录器』

相关任务

第 663 页的『准备安装 MFT 的 Java EE 数据库记录器』
在安装 Java EE 数据库记录器之前，请遵循以下指示信息来准备 Managed File Transfer 环境。

第 665 页的『使用 WebSphere Application Server traditional 9.0 安装 MFT 的 Java EE 数据库记录器』
遵循以下指示信息，为 Managed File Transfer with WebSphere Application Server traditional 9.0 安装和配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器。

第 669 页的『为 MFT 配置 Java EE 数据库记录器的用户访问权』
为 Managed File Transfer 配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器时，需要用户帐户才能访问 IBM MQ，数据库和操作系统。 所需的操作系统用户数取决于用于托管这些组件的系统数。

第 670 页的『从独立数据库记录器迁移到 MFT 的 Java EE 数据库记录器』
您可以从独立数据库记录器迁移到 Java EE 数据库记录器。 必须停止独立数据库记录器，并安装 JEE 数据库记录器。 要避免丢失或复制日志条目，必须停止将消息发布到 SYSTEM.FTE 主题，并在安装 Java EE 数据库记录器后将其重新启动。 备份您的数据库，然后再执行迁移。

相关参考
MFT 记录器的权限

准备安装 *MFT* 的 *Java EE* 数据库记录器
在安装 Java EE 数据库记录器之前，请遵循以下指示信息来准备 Managed File Transfer 环境。

关于此任务
有关 Java EE 数据库记录器的更多信息，请参阅主题第 643 页的『配置 MFT 记录器』。

过程
1. 使用数据库的文档指导您安装数据库软件。
   如果 JDBC 支持是您数据库的可选组件，那么必须安装此组件。
2. 使用数据库提供的工具创建数据库。 数据库必须具有至少 8K 的表空间和缓冲池页面大小。
   缺省模式名称为 FTELOG。 如果使用不同于 FTELOG 的模式名称，那么必须编辑针对数据库提供的 SQL 文件（`ftelog_tables_db2.sql` 或 `ftelog_tables_oracle.sql`）以反映这一情况，然后继续下一步。

   注: `ftelog_tables_db2.sql` 和 `ftelog_tables_oracle.sql` 文件位于文件路径 `<MQ-installation-path>/mqft/sql` 中
3. 使用您数据库的工具创建所需的数据库表。

   Multi 在多平台上，`ftelog_tables_db2.sql` 和 `ftelog_tables_oracle.sql` 文件包含可运行来创建表的 SQL 命令。

   z/OS 在 z/OS 上，需要运行的文件取决于使用的 Db2 for z/OS 版本：

   - 对于 Db2 for z/OS 9.0 以及更低版本，运行文件 `ftelog_tables_zos.sql` 以创建表。 此文件在创建表时，将 INTEGER 数据类型用于表示所传输文件大小的字段以及表示与每个传输关联的表标识的字段。
   - 对于 Db2 for z/OS 9.1 以及更高版本，运行文件 `ftelog_tables_zos_bigint.sql` 以创建表。 此文件在创建表时，将 BIGINT 数据类型用于表示所传输文件大小的字段以及表示与每个传输关联的表标识的字段。
4. 如果使用的是 "Oracle 数据库，则在 "`listener.ora` 文件中添加 "`USE_SID_AS_SERVICE_`*`listener_name`*`=on`，其中 "*`listener_name`* 是数据库服务器上监听器的名称。 然后重新启动监听器。
   该参数指定数据库连接将 "Oracle 系统标识符 (SID) 作为数据库的服务名称。 需要使用该参数的原因是，"MFT 数据库日志记录器仅使用 SID 连接 "Oracle 数据库，但后来的 "Oracle 数据库（从 12c 版本开始）使用服务名称。 更多信息，请参阅 "Oracle 版本的文档，例如 "Oracle Net Listener Parameters in the listener.ora File。 如果不修改 "`listener.ora` 文件，运行记录仪时可能会看到以下信息：

   ```
   ORA-12505, TNS:listener does not currently know of SID given in connect descriptor
   ```
5. 如果通过 FTELOG 更改了模式名称，那么必须更改 EAR 文件中的模式名称。 有关更多信息，请参阅第 664 页的『在 MFT 的 Java EE 数据库记录器中更改模式名称』。
6. 在 IBM MQ 中创建拒绝队列。
   由于记录器从不废弃日志消息，因此，如果记录器遇到其无法处理的消息， 会将消息放入拒绝队列以进行检查，并有可能重新处理。 请勿将队列管理器的死信队列用于此目的， 这是因为被拒绝的消息没有 DLH 头，并且被拒绝的消息不得与由于其他原因而放入死信队列的消息混合在一起。
   `fteCreateLogger` 命令用于创建拒绝队列。 该拒绝队列的缺省名称是 SYSTEM.FTE.LOG.RJCT.*logger_name*。
7. 遵循主题第 669 页的『为 MFT 配置 Java EE 数据库记录器的用户访问权』中的指示信息。

下一步做什么
在符合 Java EE 的应用程序服务器中安装 Java EE 数据库记录器。 使用第 665 页的『使用 WebSphere Application Server traditional 9.0 安装 MFT 的 Java EE 数据库记录器』中的指示信息

在 *MFT* 的 *Java EE* 数据库记录器中更改模式名称

Java Platform, Enterprise Edition (Java EE) 数据库记录器可以使用具有非缺省模式名称的数据库。 您必须更改 Java EE 数据库记录器 EAR 文件中的模式名称。

关于此任务

要更改 Java EE 数据库记录器使用的模式的名称，请完成以下步骤:

过程

1. 通过使用以下命令，从 EAR 文件中抽取 JPA JAR 文件:

   ```
   jar -xvf ear_file lib/jpa_file
   ```

   其中:

   - *ear_file* 是 `com.ibm.wmqfte.databaselogger.jee.oracle.ear` 或 `com.ibm.wmqfte.databaselogger.jee.ear`，具体取决于您使用的是 Db2 还是 Oracle。
   - *jpa_file* 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，具体取决于您使用的是 Db2 还是 Oracle。
2. 通过使用以下命令，从 JPA JAR 文件中抽取 `persistence.xml` 文件:

   ```
   jar -xvf lib/jpa_file META_INF/persistence.xml
   ```

   其中:

   - *jpa_file* 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，具体取决于您使用的是 Db2 还是 Oracle。
3. 编辑 `persistence.xml` 文件，更改以下行:

   ```
   <property name="openjpa.jdbc.Schema" value="schema_name" />
   ```

   其中:

   - *schema_name* 是您希望使用的模式名称。
4. 通过使用以下命令，以修改后的 `persistence.xml` 文件更新 JPA JAR:

   ```
   jar -uvf lib/jpa_file META_INF/persistence.xml
   ```

   其中:

   - *jpa_file* 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，具体取决于您使用的是 Db2 还是 Oracle。
5. 通过使用以下命令，以修改后的 JPA JAR 文件更新 EAR 文件:

   ```
   jar -uvf ear_file lib/jpa_file
   ```

   其中:

   - *ear_file* 是 `com.ibm.wmqfte.databaselogger.jee.oracle.ear` 或 `com.ibm.wmqfte.databaselogger.jee.ear`，具体取决于您使用的是 Db2 还是 Oracle。
   - *jpa_file* 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，具体取决于您使用的是 Db2 还是 Oracle。

下一步做什么

使用修改后的 EAR 文件安装 Java EE 数据库记录器。

相关任务

第 665 页的『使用 WebSphere Application Server traditional 9.0 安装 MFT 的 Java EE 数据库记录器』
遵循以下指示信息，为 Managed File Transfer with WebSphere Application Server traditional 9.0 安装和配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器。

在 *WebSphere Application Server traditional 9.0* 中设置本机库路径

如果在 WebSphere Application Server traditional 9.0 上部署 Java Platform, Enterprise Edition (Java EE) 数据库记录器应用程序，并且要在应用程序与 IBM MQ 之间使用绑定方式连接，那么必须使用系统上 IBM MQ 本机库的位置配置 IBM MQ 消息传递提供程序。

关于此任务

如果未在应用程序服务器中设置本机库路径，那么您可能会在 WebSphere Application Server traditional 9.0 系统输出日志中收到以下错误消息：

```
A connection could not be made to WebSphere MQ for the following reason:
  CC=2;RC=2495;AMQ8568: The native JNI library 'mqjbnd' was not found. [3=mqjbnd]
```

使用 WebSphere Application Server traditional 9.0 管理控制台以完成以下步骤：

过程

1. 在导航窗格中，展开**资源** > **JMS** > **JMS 提供程序**。
2. 为用于创建绑定方式连接的连接工厂或激活规范选择适当范围内的 IBM MQ 消息传递提供程序。

   注: 在使用时，`Server` 范围的本机路径信息优先于更高范围的本机路径信息，而 `Node` 范围的本机路径信息优先于 `Cell` 范围的本机路径信息。
3. 在“常规属性”下的**本机库路径**字段中，输入包含 IBM MQ 本机库的目录的全名。

   例如，在 Linux 上，输入 `/opt/mqm/java/lib`。 只输入一个目录名称。
4. 单击**确定**。

   设置路径后，应将更改保存到主配置以使更改生效。
5. 重新启动应用程序服务器以刷新配置。
6. 必需： 再次重新启动应用程序服务器以装入库。

使用 *WebSphere Application Server traditional 9.0* 安装 *MFT* 的 *Java EE* 数据库记录器

遵循以下指示信息，为 Managed File Transfer with WebSphere Application Server traditional 9.0 安装和配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器。

开始之前

安装 JEE 数据库记录器应用程序之前，请按照主题第 663 页的『准备安装 MFT 的 Java EE 数据库记录器』和第 665 页的『在 WebSphere Application Server traditional 9.0 中设置本机库路径』中的指示信息进行操作。

关于此任务

有关 Java EE 数据库记录器的更多信息，请参阅第 643 页的『配置 MFT 记录器』。

过程

1. 设置 XA JDBC 提供程序：
   a) 从 WebSphere Application Server traditional 9.0 管理控制台导航中选择 **资源** > **JDBC** > **JDBC 提供程序** 。
   b) 单击**新建**，以使用控制台向导创建 JDBC 提供程序。
   c) 在向导的步骤 1 中，从**数据库类型**列表中选择正在使用的数据库，并从**提供程序类型**列表中选择关联的提供程序类型。 从**实施类型**列表中选择 **XA 数据源**。 单击**下一步**。

      z/OS 您可以除去对 `db2jcc_license_cisuz.jar` 的引用，并且应该将 `db2jcc.jar` 更改为 `db2jcc4.jar`，即，随 Db2 的最新版本一起提供的 jar 文件版本或本地版本。
   d) 在向导的步骤 2 中，确保正确设置了必需数据库 jar 文件的目录位置。 单击**下一步**。
   e) 单击摘要页面上的**完成**以创建 JDBC 提供程序。
2. 创建认证别名。 为数据源创建一个别名，并为 IBM MQ 创建一个别名：

a) 从 WebSphere Application Server traditional 9.0 管理控制台导航中选择 **安全性** > **全局安全性** 。

b) 在**认证**标题下，展开 **Java 认证和授权服务**。

c) 单击 **J2C 认证数据**。 这样会打开认证别名页面。

d) 为数据源创建认证别名：

i) 单击**新建**。

ii) 输入**别名**、**用户标识**、**密码**和**描述**的详细信息。 在**用户标识**和**密码**字段中输入的详细信息必须与在创建数据库用户时输入的详细信息相匹配。 有关更多信息，请参阅 第 669 页的『为 MFT 配置 Java EE 数据库记录器的用户访问权』。

iii) 单击**确定**。

e) 为 IBM MQ 创建一个认证别名：

i) 单击**新建**。

ii) 输入**别名**、**用户标识**、**密码**和**描述**的详细信息。 在**用户标识**和**密码**字段中输入的详细信息必须与 IBM MQ 安装的用户和密码设置相匹配。

iii) 单击**确定**。

3. 创建数据源：

a) 从 WebSphere Application Server traditional 9.0 管理控制台导航中选择 **资源** > **JDBC** > **数据源** 。

b) 选择**作用域**下拉列表，并将作用域更改为适当值。 例如，`Node=yourNode, Server=yourServer`。

c) 单击**新建**，以使用控制台向导创建数据源。

d) 在向导的步骤 1 中，在**数据源名称**字段中输入 `wmqfte-database`，并在 **JNDI 名称**字段中输入 `jdbc/wmqfte-database`。 单击**下一步**。

e) 在向导的步骤 2 中，使用**选择现有的 JDBC 提供程序**下拉列表来选择先前步骤中创建的 JDBC 提供程序。 单击**下一步**。

f) **Db2：**在向导的步骤 3 中，在**驱动程序类型**字段中输入 4。

g) **Db2：**在**数据库名称**、**服务器名称**和**端口号**字段中输入详细信息，然后单击**下一步**。

Oracle：在 **URL** 字段中输入连接 URL，并在**数据存储 帮助程序类名**字段中选择正确的数据存储帮助程序。

Oracle RAC：当连接到 Oracle Real Application Cluster 时，连接 URL 必须包含连接所有可用数据库实例所需的主机信息。

h) 在向导的步骤 4 中，从**用于 XA 恢复的认证别名**列表中，选择您在步骤 2d 中定义的数据源认证别名的名称。 从**组件管理的认证别名**和**容器管理的认证别名**列表中选择相同的名称。

i) 单击摘要页面上的**完成**以创建数据源。

4. 可选： 验证数据源的配置：

a) 从 WebSphere Application Server traditional 9.0 管理控制台导航中选择 **资源** > **JDBC** > **数据源** 。

b) 单击**测试连接**按钮。

5. 创建主题。

a) 在 WebSphere Application Server traditional 9.0 管理控制台导航中，单击**资源** > **JMS** > **主题**。

b) 选择**作用域**下拉列表，并将作用域更改为适当值。 例如，`Node=yourNode, Server=yourServer`。

c) 单击**新建**。

d) 单击 **IBM MQ 消息传递提供程序**。

e) 在主题属性页面的**“管理”**面板上，为**名称**和 **JNDI 名称**字段选择唯一值，稍后您将在配置中引用这些值。

f) 在**“IBM MQ 主题”**面板中的**主题名称**字段中，输入 `SYSTEM.FTE/Log/#`。

6. 创建激活规范：

a) 在 WebSphere Application Server traditional 9.0 管理控制台导航中，单击**资源** > **JMS** > **激活规范**。

b) 选择**作用域**下拉列表，并将作用域更改为适当值。 例如，`Node=yourNode, Server=yourServer`。

c) 单击**新建**。

d) 单击 **IBM MQ 消息传递提供程序**。

e) 在向导的步骤 1 中，为**名称**和 **JNDI 名称**字段选择唯一值，稍后您将在配置中再次引用这些值。

f) 在步骤 1.1 中，在**目标 JNDI 名称**字段中输入您在步骤 5 中所设置主题的 JNDI 名称。

g) 从**目标类型**列表中选择**主题**。

h) 在向导的步骤 1.2 中，选择**持久预订**。 在**预订名称**字段中输入 `SYSTEM.FTE.DATABASELOGGER.AUTO`。

i) 在向导的步骤 2 中，选择**在该向导中输入所有必需信息**。

j) 在步骤 2.1 中，在**队列管理器或队列共享组名**字段中输入队列管理器名称。

k) 在步骤 2.2 中，从**传输**列表中选择您所选的传输方法。 如果选择**绑定**，那么无需任何其他信息。 如果选择**客户机**或**先绑定，然后客户机**，那么输入**主机名、端口**和**服务器连接通道**的详细信息。

l) 可选： 单击**测试连接**以确认队列管理器存在。 但是，您可能预期会收到 `NOT_AUTHORIZED`，直到您在步骤 6n 中引用了认证别名为止。

m) 单击**保存**。

n) 单击您创建的激活规范的名称。 在 " **配置** " 选项卡的 **常规属性** 部分中，向下滚动到 " **高级** " 面板，并在 **客户机标识** 字段中输入用于标识 IBM MQ 连接的唯一名称。 您必须完成该步骤，否则 IBM MQ 将拒绝该连接并返回 `JMSCC0101` 错误代码。

o) 如果选择**客户机**作为传输方法，请向下滚动至**"安全性设置"**面板， 并从**认证别名**列表中选择您在步骤 8 中定义的认证别名。

p) 单击**应用**。

q) 在**配置**选项卡的**"其他属性"**部分中，单击**高级属性**。 在**"高级属性"**面板的**"连接使用程序"**部分中，将 1 输入到**最大服务器会话数**字段中。

注: 确保完成该步骤，然后再继续。 如果不这样做，会导致记录器无法正确运行。

r) 在**配置**选项卡的**"其他属性"**部分中，单击**高级属性**。 将**如果消息传递失败，停止端点**的值设置为不低于 1。

如果 **_numberOfFailedAttemptsBeforeReject** 属性的值设置为大于 1 (请参阅 9j 以获取更多信息)，请将 **消息传递失败时停止端点** 设置为至少 **_numberOfFailedAttemptsBeforeReject** 属性的值。 这样可防止在收到无法处理的消息（例如，格式不正确的传输日志消息）时端点停止。 有关更多信息，请参阅 MFT 记录器错误处理和拒绝。

7. 创建队列连接工厂。

a) 在 WebSphere Application Server traditional 9.0 管理控制台导航中，单击**资源** > **JMS** > **队列连接工厂**。

b) 选择**作用域**下拉列表，并将作用域更改为适当值。 例如，`Node=yourNode, Server=yourServer`。

c) 单击**新建**。

d) 单击 **IBM MQ 消息传递提供程序**。

e) 在向导的步骤 1 中，为**名称**和 **JNDI 名称**字段选择唯一值，稍后您将在配置中再次引用这些值。

f) 在步骤 2 中，选择**在该向导中输入所有必需信息**。

g) 在步骤 2.1 中，在**队列管理器或队列共享组名**字段中输入队列管理器名称。

h) 在步骤 2.2 中，从**传输**列表中选择您所选的传输方法。 如果选择**绑定**，那么无需任何其他信息。 如果选择**客户机**或**先绑定，然后客户机**，那么输入**主机名、端口**和**服务器连接通道**的详细信息。

i) 可选： 单击**测试连接**以确认队列管理器存在。 但是，您可能预期会收到 `NOT_AUTHORIZED`，直到您在步骤 7h 中引用了认证别名为止。

j) 如果选择了**客户机**或**先绑定，然后客户机**作为传输方法， 请单击您刚创建的队列连接工厂的名称。 向下滚动至**配置**选项卡的**"安全性设置"**面板， 从**用于 XA 恢复的认证别名**和**容器管理的认证别名**列表中选择您在步骤 2e 中定义的认证别名。

8. 在 WebSphere Application Server 中创建拒绝队列：
   a) 在 WebSphere Application Server traditional 9.0 管理控制台导航中，单击**资源** > **JMS** > **队列**。
   b) 选择**作用域**下拉列表，并将作用域更改为适当值。 例如，`Node=yourNode, Server=yourServer`。
   c) 单击**新建**。
   d) 单击 **IBM MQ 消息传递提供程序**。
   e) 为**名称**和 **JNDI 名称**字段选择唯一值，稍后您将在配置中再次引用这些值。
   f) 在**队列名称**字段中输入 `SYSTEM.FTE.LOG.RJCT.`*`logger_name`*。 确保在协调队列管理器上创建了该队列。
   g) 在**队列管理器名称**字段中输入队列管理器名称。
   h) 单击**确定**。
9. 安装 JEE 数据库记录器应用程序：
   a) 从 WebSphere Application Server traditional 9.0 管理控制台中，选择**应用程序** > **新建应用程序**。
   b) 选择**作用域**下拉列表，并将作用域更改为适当值。 例如，`Node=yourNode, Server=yourServer`。
   c) 从选项列表中选择**新建企业应用程序**。
   d) 在 " **准备应用程序安装** " 页面上，从 Managed File Transfer Service 安装的 *MQ_INSTALLATION_PATH*`/mqft/web` 目录中选择 `com.ibm.wmqfte.databaselogger.jee.ear` 文件或 `com.ibm.wmqfte.databaselogger.jee.oracle.ear` 文件，然后单击 **下一步**。
   e) 在以下屏幕上，选择**详细**以显示所有安装选项和参数，然后单击**下一步**。
   f) 单击向导步骤 1-4 中的**下一步**，接受缺省值。
   g) 在向导的步骤 5 **针对消息驱动的 Bean 绑定侦听器**中，滚动至**侦听器绑定**部分。 单击**激活规范**。
   输入以下字段的必需值：

   目标资源 JNDI 名称
   您在步骤 6d 中创建激活规范时指定的 JNDI 名称。

   目标 JNDI 名称
   您在步骤 5d 中创建主题时指定的 JNDI 名称。

   单击**下一步**。
   h) 在向导的步骤 6**"将资源引用映射到资源"**中，在**目标资源 JNDI 名称**字段中输入详细信息。 该名称是您在步骤 7c 中为拒绝队列连接工厂指定的 JNDI 名称。 单击 **下一步**。
   i) 在向导的步骤 7**"将资源环境条目引用映射到资源"**中，在**目标资源 JNDI 名称**字段中输入详细信息。 该名称是您在步骤 8d 中创建的拒绝队列的 JNDI 名称。 单击 **下一步**。
   j) 在向导的步骤 8 中， **映射 EJB 模块的环境条目**，接受缺省值 1。 单击**下一步**。

   Oracle RAC: 连接到 Oracle Real Application Cluster 时，必须将 _numberOfFailedAttemptsBeforeReject 属性的值设置为 **至少** 2。 此属性确定发生故障后记录器尝试处理审计消息的次数。 在数据库故障转移情况下，至少有可能发生一次失败。 为了避免不必要地将消息移动到拒绝队列，增加该值可进行第二次尝试，这通常会成功，因为会连接到新的数据库实例。 如果在测试时发现消息在数据库实例故障转移期间仍移动到拒绝队列，可进一步增加该值：实例之间的切换计时可能导致同一消息发生多次失败。 但是，要知道增加该值会影响所有失败情况（例如，格式不正确的消息），而不仅仅是数据库故障转移，因此请慎重增加该值，以避免不必要的重试。
   k) 在向导的步骤 9 **模块的元数据**中，单击**下一步**。
   l) 在向导的步骤 10 **摘要**中，单击**完成**。
10. 您现在可以从 WebSphere Application Server traditional 9.0 管理控制台中启动该应用程序：
   a) 从控制台导航中，选择**应用程序** > **应用程序类型** > **WebSphere 企业应用程序**。
   b) 选中集合表中**记录器**企业应用程序的复选框，然后单击**启动**。

为 *MFT* 配置 *Java EE* 数据库记录器的用户访问权

为 Managed File Transfer 配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器时，需要用户帐户才能访问 IBM MQ，数据库和操作系统。 所需的操作系统用户数取决于用于托管这些组件的系统数。

关于此任务

运行 Java EE 数据库记录器所需的用户帐户数和类型取决于所用的系统数。 需要用户帐户才能访问以下三种环境：

- 本地操作系统
- IBM MQ
- 数据库

您可以在同一个系统上或在不同系统上安装 JEE 数据库记录器、IBM MQ 和数据库。 可以在以下示例拓扑中安装这些组件：

Java EE 数据库记录器、IBM MQ 和数据库全都位于同一个系统上
您可以定义单个操作系统用户，以用于所有这三个组件。 该记录器使用绑定方式连接到 IBM MQ，并使用本机连接来连接到数据库。

Java EE 数据库记录器和 IBM MQ 位于一个系统上，而数据库位于另一个系统上
您为该配置创建两个用户：运行记录器的系统上的一个操作系统用户，以及对数据库服务器上的数据库具有远程访问权的一个操作系统用户。 该记录器使用绑定方式连接到 IBM MQ，并使用客户机连接来访问数据库。

Java EE 数据库记录器、IBM MQ 和数据库分别位于不同的系统上
您为该配置创建三个用户：用于启动应用程序服务器的操作系统用户、用于访问要使用的队列和主题的 IBM MQ 用户，以及用于访问数据库表以及将内容插入数据库表的数据库服务器用户。 该记录器使用客户机方式访问 IBM MQ，并使用客户机连接来访问数据库。

例如，这些指示信息的其余部分假定用户名为 `ftelog`，但您也可以使用任何新的或现有的用户名。 请按照以下方式配置该用户的许可权：

过程

1. 确保操作系统用户具有自己的组，并且不在对协调队列管理器具有广泛许可权的任何组中。 该用户不应在 mqm 组中。 在某些平台上，还会自动为 staff 组授予队列管理器访问权；记录器用户不应位于 staff 组中。 您可以使用 IBM MQ Explorer 来查看队列管理器本身及其中对象的权限记录。 右键单击该对象，然后选择**对象权限 > 管理权限记录**。 在命令行上，可以使用命令 dspmqaut（显示权限）或 dmpmqaut（转储权限）。
2. 使用 IBM MQ Explorer 或 setmqaut (grant 或 revoke authority) 命令中的 " **管理权限记录** " 窗口为 IBM MQ 用户自己的组添加权限 (在 AIX 上， IBM MQ 权限仅与组关联，而与个人用户无关)。 所需权限如下所示：
   - 对队列管理器的 CONNECT 和 INQUIRE 许可权（IBM MQ Java 库需要 INQUIRE 许可权才能运行）。
   - 对 SYSTEM.FTE 主题的 SUBSCRIBE 许可权。
   - 对 SYSTEM.FTE.LOG.RJCT.*logger_name* 队列的 PUT 许可权。

   给定的 REJECT 和 COMMAND 队列名称是缺省名称。 如果您在配置记录器队列时选择了不同的队列名称，请改为向这些队列名称添加许可权。
3. 执行特定于您所用数据库的数据库用户配置。
   - 如果您的数据库是 Db2，请执行以下步骤：

     注: 有几种机制可用于管理 Db2 的数据库用户。 这些指示信息适用于基于操作系统用户的缺省方案。

     – 确保 `ftelog` 用户不在任何 Db2 管理组中 (例如， db2iadm1， db2fadm1 或 dasadm1)。
     – 授予用户连接到数据库的许可权以及对您在 步骤 2 中创建的表进行选择，插入和更新的许可权: 创建所需的数据库表。
   - 如果您的数据库是 Oracle，请执行以下步骤：

– 确保 `ftelog` 用户不在任何 Oracle 管理组中 (例如， Windows 上的 ora_dba 或 AIX and Linux 上的 dba)。
– 为用户提供连接到数据库的许可权，以及对步骤 2：创建所需的数据库表过程中所创建表的选择、插入和更新许可权。

从独立数据库记录器迁移到 MFT 的 Java EE 数据库记录器

您可以从独立数据库记录器迁移到 Java EE 数据库记录器。 必须停止独立数据库记录器，并安装 JEE 数据库记录器。 要避免丢失或复制日志条目，必须停止将消息发布到 SYSTEM.FTE 主题，并在安装 Java EE 数据库记录器后将其重新启动。 备份您的数据库，然后再执行迁移。

关于此任务

过程

1. 在停止该数据库之前，请针对协调队列管理器运行以下 MQSC 命令：`ALTER QM PSMODE(COMPAT)`
   这会停止向 SYSTEM.FTE/Log 主题发布消息。 等待该记录器处理完预订中的所有消息。 缺省情况下，该预订名为 SYSTEM.FTE.LOGGER.AUTO。
2. 使用 **fteStopLogger** 命令停止数据库记录器。
3. 使用数据库软件随附的工具来备份数据库。
4. 删除属于该独立数据库记录器的预订。
   缺省情况下，该预订名为 SYSTEM.FTE.LOGGER.AUTO。
5. 如果您的数据库模式为较早版本，那么必须按顺序将该模式迁移到每个后续级别。 例如，如果您的数据库模式为 V7.0.1 并且您要迁移到 V7.0.4，那么必须将该模式从 V7.0.1 迁移到 V7.0.2，然后从 V7.0.2 迁移到 V7.0.3，再从 V7.0.3 迁移到 V7.0.4。 通过针对您必须迁移经过的每个模式版本执行以下某个操作，将数据库模式从 V*old* 迁移到 V*new*（其中 *old* 和 *new* 是描述模式版本的变量）：
   - z/OS 如果数据库是 Db2 on z/OS 并且您要在 V7.0.2 和 V7.0.3 模式之间或者 V7.0.3 和 V7.0.4 模式之间迁移，那么您必须创建新的数据库模式，然后将现有数据复制到其中。 有关更多信息，请参阅 将 Db2 on z/OS 上的数据库表迁移到 MQ V8.0 或更高版本。
   - 如果您的数据库不是 Db2，或者您使用 8K 以上的页面大小创建了数据库，那么可以通过完成以下步骤，采用与其他版本相同的方法迁移模式。
   - 如果您要在其他任何情况下在数据库表之间迁移，请完成以下步骤：
     a. 选择适合数据库平台且名称包含字符串 *old-new* 的文件。 此文件位于“远程工具和文档”安装的 *MQ_INSTALLATION_PATH*/mqft/sql 目录中。
     b. 如果您对初始模式进行了修改，请复审迁移文件以确保该文件与修改的数据库兼容。
     c. 针对您的数据库运行 SQL 文件。
6. 安装 Java EE 数据库记录器 EAR 文件。
7. 部署 Java EE 数据库记录器。 有关更多信息，请参阅 第 662 页的『为 MFT 安装 Java EE 数据库记录器』。
8. 针对协调队列管理器运行以下 MQSC 命令：`ALTER QMGR PSMODE(ENABLED)`
   这会启用向 SYSTEM.FTE/Log 主题发布消息的操作。

结果

配置 Connect:Direct 网桥

配置 Connect:Direct 网桥，以在 Managed File Transfer 网络和 Connect:Direct 网络之间传输文件。 Connect:Direct 网桥的组件包括 Connect:Direct 节点和专用于与该节点通信的 Managed File Transfer 代理。 该代理称为 Connect:Direct 网桥代理。

开始之前

组成 Connect:Direct 网桥的代理和节点必须在同一个系统上，或具有同一文件系统的访问权，例如，通过一个共享的 NFS 安装。 此文件系统用于在涉及 Connect:Direct 网桥的文件传输期间将文件临时存储在由 **cdTmpDir** 参数定义的目录中。 Connect:Direct 网桥代理和 Connect:Direct 网桥节点必须能够使用同一路径名称来对该目录进行寻址。 例如，如果代理和节点在不同的 Windows 系统上，那么这些系统必须使用相同的盘符来安装共享文件系统。 以下配置允许代理和节点使用同一路径名：

- 代理和节点在同一系统上，该系统运行 Windows 或 Linux for x86-64
- 代理在 Linux for x86-64 上，而该节点在 AIX 上
- 代理在一个 Windows 系统上，而节点在另一个 Windows 系统上

以下配置不允许代理和节点使用同一路径名：

- 代理在 Linux for x86-64 上，而该节点在 Windows 上
- 代理在 Windows 上，而该节点在 UNIX 上

请在计划安装 Connect:Direct 网桥时考虑该限制。

有关 Connect:Direct 网桥支持的操作系统版本的更多详细信息，请参阅 Web 页面 System Requirements for IBM MQ。

关于此任务

Connect:Direct 网桥代理是专用于与 Connect:Direct 节点通信的 Managed File Transfer 代理。

缺省情况下，Connect:Direct 网桥代理使用 TCP/IP 协议来连接到 Connect:Direct 节点。 如果要在 Connect:Direct 网桥代理和 Connect:Direct 节点之间建立安全连接，那么您可以使用 SSL 协议或 TLS 协议。

过程

1. 选择 Connect:Direct 网桥代理和节点的操作系统：
   a) 选择运行 Windows 或 Linux on x86-64 的系统，以用于安装 Connect:Direct 网桥代理。
   b) 选择 Connect:Direct for Windows 或 Connect:Direct for UNIX 支持的操作系统，以用于安装 Connect:Direct 网桥节点。
2. 选择并配置 Connect:Direct 节点。
   在遵照以下指示信息之前，必须已安装 Connect:Direct 节点。
   a) 选择 Connect:Direct 节点，以供 Managed File Transfer 代理与其进行通信。
   b) 检查所选 Connect:Direct 节点的网络映射。 如果网络映射中包含在 Windows 操作系统上运行的远程节点的任何条目，那么必须确保这些条目指定这些节点在 Windows 上运行。

   Windows

   如果针对 Connect:Direct 网桥选择的 Connect:Direct 节点正在 Windows 上运行，那么使用 Connect:Direct 请求程序来编辑网络映射。 确保 Windows 上运行的任何远程节点的 **操作系统** 字段设置为 **Windows**。
3. 创建并配置 Connect:Direct 网桥代理。
   a) 使用 **fteCreateCDAgent** 命令创建 Connect:Direct 网桥代理。
      - 您必须为 **cdNode** 参数提供值。 该参数指定代理用于作为 Connect:Direct 网桥的组成部分的 Connect:Direct 节点的名称。 使用您在前一部分中选择的 Connect:Direct 节点的名称。
      - 为 **cdNodeHost** 和 **cdNodePort** 参数提供值，它们用于定义代理要与其通信的 Connect:Direct 节点。

        如果不为 **cdNodeHost** 参数提供值，那么将使用本地系统的主机名或 IP 地址。 如果不为 **cdNodePort** 参数提供值，那么将使用值 1363。
      - (可选) 使用 fteCreateAgent 中的信息来确定是否需要为 **cdTmpDir** 参数指定值。

b) 将 Managed File Transfer 使用的用户凭证映射到 Connect:Direct 节点上的用户凭证。 您可以通过使用以下某方法来映射凭证：

- 创建 `ConnectDirectCredentials.xml` 文件以定义凭证映射信息。 有关更多信息，请参阅 第 672 页的『通过使用 ConnectDirectCredentials.xml 文件映射 Connect:Direct 的凭证』。
- 编写用户出口，以针对 Connect:Direct 网桥执行凭证映射。 有关更多信息，请参阅 第 674 页的『通过出口类映射 Connect:Direct 的凭证』。

4. 配置 `ConnectDirectNodeProperties.xml` 文件以包含有关远程 Connect:Direct 节点的信息。

   在遵照以下指示信息之前，必须已创建 Connect:Direct 网桥代理。

   编辑 Connect:Direct 网桥代理配置目录中的模板 `ConnectDirectNodeProperties.xml` 。 对于要定义相关信息的每个 Connect:Direct 节点或节点组，请执行以下步骤：

   a) 在 nodeProperties 元素中，创建 node 元素。

   b) 将 name 属性添加到 node 元素。 将该属性的值指定为模式，以与一个或多个远程 Connect:Direct 节点的名称匹配。

   c) 可选： 将 pattern 属性添加到 node 元素，该元素指定 name 属性中的值的模式类型。 有效值为 `regex` 和 `wildcard`。 缺省选项为 `wildcard`。

   d) 将 type 属性添加到 node 元素，该元素指定运行 name 属性所指定的远程 Connect:Direct 节点的操作系统。

   有效值包括：

   - `Windows` - 节点在 Windows 上运行
   - `UNIX` -节点在 AIX and Linux 上运行
   - z/OS `z/OS`、`zos`、`os/390` 或 `os390` - 节点在 z/OS 上运行

   该属性的值不区分大小写。 Connect:Direct 网桥不支持到其他操作系统上的远程节点的传输。

   有关更多信息，请参阅 Connect:Direct 节点属性文件格式。

5. 配置 Connect:Direct 网桥代理与 Connect:Direct 节点之间的安全连接。 有关如何执行此操作的示例，请参阅 在 Connect:Direct 网桥代理与 Connect:Direct 节点之间配置 SSL 或 TLS。

映射 Connect:Direct 的凭证

通过使用 Connect:Direct 网桥代理的缺省凭证映射功能或通过编写您自己的用户出口，将 Managed File Transfer 中的用户凭证映射到 Connect:Direct 节点上的用户凭证。 Managed File Transfer 提供了执行用户凭证映射的样本用户出口。

通过使用 *ConnectDirectCredentials.xml* 文件映射 *Connect:Direct* 的凭证

通过使用 Connect:Direct 网桥代理的缺省凭证映射功能，将 Managed File Transfer 中的用户凭证映射到 Connect:Direct 节点上的用户凭证。 Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以包含自己的凭证信息。

关于此任务

使用 **fteCreateCDAgent** 命令创建 Connect:Direct 网桥代理后，需要手动创建 `ConnectDirectCredentials.xml` 文件。 必须编辑此文件以包含主机、用户和凭证信息，然后才能使用 Connect:Direct 网桥代理。 有关更多信息，请参阅 Connect:Direct 凭证文件格式。 缺省情况下，此文件从当前用户主目录（例如 `/home/fteuser/ConnectDirectCredentials.xml`）中装入。 要使用其他位置，请使用 `ConnectDirectNodeProperties.xml` 文件中的 `<credentialsFile>` 元素指定该位置。

过程

1. 确保元素 `<tns:pnode name="Connect:Direct node host" pattern="wildcard">` 中的 name 属性包含 Connect:Direct 网桥代理连接到的 Connect:Direct 节点的名称值。 该值必须与您为 **fteCreateCDAgent -cdNode** 参数指定的值相同。

   `pattern` 属性的值可以是 `wildcard` 或 `regex`。 如果未指定该属性，那么缺省值将为 `wildcard`。

2. 将用户标识和凭证信息作为 <tns:pnode> 的子元素插入该文件。
   您可以将以下 <tns:user> 元素的一个或多个实例插入该文件：

```
<tns:user name="name"
          pattern="pattern"
          ignorecase="ignorecase"
          cdUserId="cdUserId"
          cdPassword="cdPassword"
          pnodeUserId="pnodeUserId"
          pnodePassword="pnodePassword">
</tns:user>
```

   其中：

   - *name* 是一个模式，用于匹配与 MFT 传输请求关联的 MQMD 用户标识。
   - *pattern* 指定为 name 属性指定的模式是通配符表达式，还是 Java 正则表达式。 pattern 属性的值可以是 wildcard 或 regex。 如果未指定该属性，那么缺省值将为 wildcard。
   - *ignorecase* 指定 name 属性指定的模式是否区分大小写。 如果未指定该属性，那么缺省值将为 true。
   - *cdUserId* 是 Connect:Direct 网桥代理用于连接到 <tns:pnode> 元素的 name 属性所指定的 Connect:Direct 节点的用户标识。 如果可能，请确保 *cdUserId* 是 Connect:Direct 管理员用户标识。 如果 *cdUserId* 不能为 Connect:Direct 管理员，那么请确保该用户标识在 Connect:Direct 网桥节点具有以下功能权限：

     – 对于 Windows 节点，设置以下权限。 为了便于阅读，该示例使用回车符进行了格式化：

```
View Processes in the  value: yes
 TCQ
Issue the copy         value: yes
 receive, copy send,
 run job, and run task
 Process statements
Issue the submit       value: yes
 Process statement
Monitor, submit,       value: all
 change,and delete all
 Processes
Access Process         value: all
 statistics
Use the trace tool or  value: yes
 issue traceon and
 traceoff commands
Override Process       value: yes
 options such as file
 attributes and remote
 node ID
```

     – 对于 AIX 或 Linux 节点，请在 userfile.cfg 文件中设置以下参数：

```
pstmt.copy             value: y
pstmt.upload           value: y
pstmt.download         value: y
pstmt.runjob           value: y
pstmt.runtask          value: y
cmd.submit             value: y
pstmt.submit           value: y
cmd.chgproc            value: y
cmd.delproc            value: y
cmd.flsproc            value: y
cmd.selproc            value: a
cmd.selstats           value: a
cmd.trace              value: y
snode.ovrd             value: y
```

   - *cdPassword* 是与 cdUserId 属性指定的用户标识相关联的密码。
   - 您可以选择性地指定 pnodeUserId 属性。 此属性的值是由 <tns:pnode> 元素的 name 属性指定的 Connect:Direct 节点用于提交 Connect:Direct 进程的用户标识。 如果未指定 pnodeUserId 属性，那么 Connect:Direct 节点将使用 cdUserId 属性指定的用户标识来提交 Connect:Direct 进程。

- 您可以选择性地指定 pnodePassword 属性。 该属性的值是与 pnodeUserId 属性指定的用户标识相关联的密码。

如果不存在与 MQMD 用户标识匹配的用户元素，那么传输将失败。

3. 可选： 您可以包含一个或多个 <tns:snode> 元素，作为 <tns:user> 元素的子元素。 <tns:snode> 元素指定由作为 Connect:Direct 网桥组成部分的 Connect:Direct 节点使用的凭证。 这些凭证包含 Connect:Direct 网桥节点用于连接到作为文件传输的源或目标的 Connect:Direct 节点的用户标识和密码。

   将以下一个或多个元素插入该文件：

```
<tns:snode name="name"
           pattern="pattern"
           userId="userId"
           password="password"/>
```

   其中：

   - *name* 是用于匹配作为文件传输的源或目标的 Connect:Direct 节点的名称的模式。
   - *pattern* 指定为 name 属性指定的模式是通配符表达式，还是 Java 正则表达式。 pattern 属性的值可以是 wildcard 或 regex。 如果未指定该属性，那么缺省值将为 wildcard。
   - *userId* 是由 <tns:pnode> 元素的 name 属性指定的 Connect:Direct 节点用于连接到与 <tns:snode>的 name 属性指定的模式相匹配的 Connect:Direct 节点的用户标识。
   - *password* 是与 userId 属性指定的用户标识相关联的密码。

   如果没有与文件传输的辅助节点匹配的 <tns:snode> 元素，那么这种情况不会导致传输失败。 传输将启动并且未指定与 snode 一起使用的用户标识和密码。

结果

在搜索用户名或 Connect:Direct 节点名的模式匹配时，Connect:Direct 网桥代理将从文件开头搜索到文件末尾。 找到的第一个匹配既是使用的匹配。

相关任务

第 670 页的『配置 Connect:Direct 网桥』

配置 Connect:Direct 网桥，以在 Managed File Transfer 网络和 Connect:Direct 网络之间传输文件。 Connect:Direct 网桥的组件包括 Connect:Direct 节点和专用于与该节点通信的 Managed File Transfer 代理。 该代理称为 Connect:Direct 网桥代理。

相关参考

Connect:Direct 凭证文件格式

fteCreateCDAgent：创建 Connect:Direct 网桥代理

通过出口类映射 *Connect:Direct* 的凭证

如果不希望使用 Connect:Direct 网桥代理的缺省凭证映射功能，那么可以通过编写自己的用户出口，将 Managed File Transfer 中的用户凭证映射到 Connect:Direct 节点上的用户凭证。 配置自己的凭证映射用户出口将会禁用缺省凭证映射功能。

关于此任务

针对映射 Connect:Direct 凭证而创建的用户出口必须实现接口 `com.ibm.wmqfte.exitroutine.api.ConnectDirectCredentialExit`。 有关更多信息，请参阅 CDCredentialExit.java 接口。

配置 IBM MQ Console 和 REST API

为托管 IBM MQ Console 和 REST API 的 mqweb 服务器提供了缺省配置。 必须完成一些配置任务（例如，配置安全性以允许用户登录），才能使用这些组件。 本主题描述了所有可用的配置选项。

过程

- 第 675 页的『mqweb 服务器的基本配置』
- 第 678 页的『配置安全性』
- 第 678 页的『配置 HTTP 主机名』
- 第 679 页的『配置 HTTP 和 HTTPS 端口』
- 第 680 页的『配置响应超时』
- 第 681 页的『配置自动启动』
- 第 681 页的『配置日志记录』
- 第 684 页的『配置 LTPA 令牌』
- 第 685 页的『配置远程队列管理器连接行为』
- 第 686 页的『配置 administrative REST API 网关』
- 第 687 页的『配置 messaging REST API』
- 第 689 页的『为 MFT 配置 REST API』
- 第 691 页的『调整 mqweb 服务器 JVM』
- 第 692 页的『IBM MQ Console 和 REST API 安装组件的文件结构』

mqweb 服务器的基本配置

必须先安装正确的组件并配置托管 REST API 或 IBM MQ Console 的 mqweb 服务器，然后才能开始使用 REST API 或 IBM MQ Console。

关于此任务

此任务的过程侧重于 mqweb 服务器的基本配置，以便您可以快速开始使用 REST API 和 IBM MQ Console。用于配置安全性的步骤概括了如何设置基本用户注册表，但也提供了用于配置用户和角色的其他选项。 有关为 mqweb 服务器配置安全性的更多信息，请参阅 IBM MQ 控制台和 REST API 安全性。

注: 您必须有权访问 `mqwebuser.xml` 文件才能完成此过程:

- z/OS 在 z/OS 上，您必须是对 `mqwebuser.xml` 文件具有写访问权的用户。
- Multi 在所有其他操作系统上，您必须是 特权用户 才能访问 `mqwebuser.xml` 文件。

过程

1. 安装 IBM MQ Console 和 REST API 组件:
   - AIX 在 AIX 上，安装 `mqm.web.rte` 文件集。 有关在 AIX 上安装文件集的更多信息，请参阅 AIX 安装任务。
   - IBM i 在 IBM i 上，安装 WEB 组件。 要使用此功能，还必须安装 5724L26 IBM MQ Java 消息传递和 Web Service 以及 5770JV1 Java SE 8 先决条件。 有关在 IBM i 上安装功能部件的更多信息，请参阅 IBM i 安装任务。
   - Linux 在 Linux 上，安装 `MQSeriesWeb` 组件。 有关在 Linux 上安装组件的更多信息，请参阅 Linux 安装任务。
   - Windows 在 Windows 上，安装 `Web Administration` 功能部件。 有关在 Windows 上安装功能部件的更多信息，请参阅 Windows 安装任务。
   - z/OS 安装 `IBM MQ for z/OS UNIX System Services Web Components` 功能部件。 有关在 z/OS 上安装组件和功能部件的更多信息，请参阅 z/OS 安装任务。
2. z/OS
   在 z/OS 上，通过运行 **crtmqweb** 脚本来创建托管 IBM MQ Console 和 REST API 的 mqweb 服务器。

此脚本将创建包含 mqweb 服务器配置和日志文件的 WebSphere Liberty 用户目录。 有关运行 **crtmqweb** 脚本的更多信息，请参阅 第 787 页的『创建 mqweb 服务器』。

3. z/OS
   在 z/OS 上，创建用于启动 mqweb 服务器的编目过程。
   有关更多信息，请参阅第 789 页的『为 mqweb 服务器创建过程』。
4. 将现有配置文件 mqwebuser.xml 替换为配置为提供基本安全性的基本注册表样本文件。 将 basic_registry.xml 文件从 *MQ_INSTALLATION_PATH*/web/mq/samp/configuration 目录复制到系统的相应目录，并将该文件重命名为 mqwebuser.xml:
   - Linux AIX 在 AIX and Linux 上：var/mqm/web/installations/*installationName*/servers/mqweb
   - Windows 在 Windows 上：*MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb

     其中 *MQ_DATA_PATH* 是 IBM MQ 数据路径，此路径是在 IBM MQ 安装期间选择的数据路径。 缺省情况下，此路径为 C:\ProgramData\IBM\MQ。
   - z/OS 在 z/OS 上：*WLP_user_directory*/servers/mqweb

     其中 *WLP_user_directory* 是运行 **crtmqweb** 脚本以创建 mqweb 服务器定义时指定的目录。

   basic_registry.xml 样本文件配置四个用户:

   MQADMIN
   作为 MQWebAdmin 角色成员的管理用户。

   mqreader
   作为 MQWebAdminRO 角色成员的只读管理用户。

   姆夫塔德明
   作为 MFTWebAdmin 角色成员的管理用户。

   姆夫特雷德尔
   作为 MFTWebAdminRO 角色成员的只读管理用户。

   所有用户也是 MQWebUser 角色的成员。

   有关可用角色的更多信息，请参阅 IBM MQ Console 和 REST API 上的角色
5. 可选： 编辑 mqwebuser.xml 文件以添加更多用户和组。 为这些用户和组分配相应的角色以获得使用 REST API 或 IBM MQ Console 的授权。 您还可以更改缺省情况下所定义用户的密码，并对新密码进行编码。 有关更多信息，请参阅配置用户和角色。

   注:

   - z/OS 在 z/OS 上，如果将用户添加到 MQWebUser 角色，那么还必须授予 mqweb 启动式任务用户标识对具有 MQWebUser 角色的用户标识的备用用户访问权。 例如:

     ```
     RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
     PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
     ```
   - z/OS Multi 要完成 messaging REST API 入门步骤，必须将用户添加到 mqwebuser.xml 文件。 此用户必须与系统上的现有 IBM MQ 用户同名。 遵循与 xml 文件中其他用户相同的格式，在 xml 文件中的以下行之后添加用户标识和密码: <user name="mftreader" password="mftreader"/>。
6. z/OS
   在 z/OS 上，通过输入以下命令来设置 WLP_USER_DIR 环境变量，以便该变量指向 mqweb 服务器配置:

   ```
   export WLP_USER_DIR=WLP_user_directory
   ```

其中，*WLP_user_directory* 是传递到 crtmqweb 的目录的名称。 例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

有关更多信息，请参阅第 787 页的『创建 mqweb 服务器』。

7. 缺省情况下， REST API 和 IBM MQ Console 仅在 mqweb 服务器所在的主机上可用。 输入以下命令以启用到 mqweb 服务器的远程连接：

```
setmqweb properties -k httpHost -v hostname
```

其中，*hostname* 指定 IP 地址、带域名后缀的域名服务器 (DNS) 主机名或安装了 IBM MQ 的服务器的 DNS 主机名。 使用星号 *（用双引号括起）可指定所有可用的网络接口，如以下示例中所示：

```
setmqweb properties -k httpHost -v "*"
```

8. 可选： 缺省情况下，未启用 administrative REST API for MFT 。 如果要使用此功能，必须将其启用并配置协调队列管理器：

   a) 通过输入以下命令来启用 administrative REST API for MFT :

   ```
   setmqweb properties -k mqRestMftEnabled -v true
   ```

   b) 通过输入以下命令来配置哪个队列管理器是协调队列管理器：

   ```
   setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
   ```

   其中 *qmgrName* 是协调队列管理器的名称。

   c) 要启用 POST 调用，请通过输入以下命令来配置哪个队列管理器是命令队列管理器：

   ```
   setmqweb properties -k mqRestMftCommandQmgr -v qmgrName
   ```

   其中 *qmgrName* 是命令队列管理器的名称。

9. 启动支持 REST API 和 IBM MQ Console 的 mqweb 服务器：

   - ALW 在 AIX, Linux, and Windows 上， 以 特权用户身份输入以下命令：

     ```
     strmqweb
     ```

   - IBM i 在 IBM i 上， 以 特权用户身份在 Qshell 中输入以下命令：

     ```
     /QIBM/ProdData/mqm/bin/strmqweb
     ```

   - z/OS 在 z/OS 上， 启动在 第 789 页的『为 mqweb 服务器创建过程』中创建的过程。

     将向 STDOUT DD 发出以下消息以指示 mqweb 服务器已成功启动。

     ```
     [AUDIT   ] MQWB2019I: MQ Console level: 9.2.4 - V924-CD924-L211028
     [AUDIT   ] MQWB0023I: MQ REST API level: 9.2.4 - V924-CD924-L211028
     [AUDIT   ] CWWKZ0001I: Application com.ibm.mq.rest started in 1.763 seconds.
     [AUDIT   ] CWWKZ0001I: Application com.ibm.mq.console started in 2.615 seconds.
     [AUDIT   ] CWWKF0011I: The mqweb server is ready to run a smarter planet. The mqweb
     server started in 10.016 seconds.
     ```

   您可以随时通过在 z/OS 上停止 mqweb 服务器启动式任务或使用 **endmqweb** 命令来停止 mqweb 服务器。 但是，如果 mqweb 服务器未在运行，那么无法使用 IBM MQ Console。

10. z/OS

可选： 在 z/OS 上，如果要允许系统自动化产品捕获 MQ Console 和 REST API 启动时发出的 MQWB2019I 和 MQWB0023I 消息，请配置 mqweb 服务器以将这些消息写入 MVS 控制台。 要配置 mqweb 服务器以将 MQWB2019I 和 MQWB0023I 消息写入 MVS 控制台，请编辑您在步骤 第 676 页的『4』中创建的 mqwebuser.xml 文件，并将以下行添加到该文件中：

```
<zosLogging enableLogToMVS="true" wtoMessage="MQWB2019I,MQWB0023I"/>
```

有关配置 z/OS mqweb 服务器中的日志记录的更多信息，请参阅 z/OS 日志记录 (zosLogging)。

下一步做什么

1. 配置 mqweb 服务器设置，包括启用 HTTP 连接和更改端口号。 有关更多信息，请参阅第 674 页的『配置 IBM MQ Console 和 REST API』。
2. (可选) 配置 REST API:
   a. 为 REST API 配置跨源资源共享。 缺省情况下，您无法通过与 REST API 不在相同域上托管的 Web 资源来访问 REST API。 即，未启用跨源请求。 您可以配置跨源资源共享 (CORS) 以允许来自指定 URL 的跨源请求。 有关更多信息，请参阅为 REST API 配置 CORS。
   b. 为 MFT 配置 REST API 。 有关更多信息，请参阅第 689 页的『为 MFT 配置 REST API』。
3. 使用 REST API 或 IBM MQ Console:
   - administrative REST API 入门
   - messaging REST API 入门
   - IBM MQ Console 入门

配置安全性

您可以通过编辑 `mqwebuser.xml` 文件来配置 IBM MQ Console 和 REST API 的安全性。 您可以通过配置基本用户注册表， LDAP 注册表或 WebSphere Liberty 随附的任何其他注册表类型来配置和认证用户。 然后，可以通过为用户和组分配角色来对这些用户进行授权。

关于此任务

要配置 IBM MQ Console 和 REST API 的安全性，必须配置用户和组。 然后，可以授权这些用户和组使用 IBM MQ Console 和/或 REST API。 有关配置用户和组以及认证和授权用户的更多信息，请参阅 IBM MQ 控制台和 REST API 安全性。

用户向 IBM MQ Console 认证时，将生成 LTPA 令牌。 此令牌使用户能够使用 IBM MQ Console ，而无需重新认证，直到令牌到期为止。

如果对 REST API 使用基于令牌的认证，那么当用户使用 HTTP POST 方法使用 `/login` REST API 资源登录时，将生成不同的 LTPA 令牌。 您可以配置此令牌何时到期，以及此令牌是否可用于 HTTP 和 HTTPS 连接。 有关更多信息，请参阅第 684 页的『配置 LTPA 令牌』。

过程

- IBM MQ 控制台和 REST API 安全性
- 第 684 页的『配置 LTPA 令牌』

配置 HTTP 主机名

缺省情况下，托管 IBM MQ Console 和 REST API 的 mqweb 服务器配置为仅允许本地连接。 即，只能在安装了 IBM MQ Console 和 REST API 的系统上访问 IBM MQ Console 和 REST API 。 您可以使用 **setmqweb** 命令配置主机名以允许远程连接。

开始之前

要完成此任务，您必须具有使用 **dspmqweb** 和 **setmqweb** 命令的特定权限：

- z/OS 在 z/OS 上，您必须有权运行 **dspmqweb** 和 **setmqweb** 命令，并且必须具有 `mqwebuser.xml` 文件的写访问权。
- Multi 在所有其他操作系统上，您必须是特权用户。

注意: z/OS

在 z/OS 上发出 **setmqweb** 或 **dspmqweb** 命令之前，必须设置 WLP_USER_DIR 环境变量，以便该变量指向 mqweb 服务器配置。

为此，请发出以下命令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是传递到 crtmqweb 的目录的名称。 例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

有关更多信息，请参阅创建 mqweb 服务器。

过程

- 使用以下命令查看 HTTP 主机名的当前配置：

  ```
  dspmqweb properties -a
  ```

  httpHost 字段显示 HTTP 主机名。 有关更多信息，请参阅 dspmqweb。
- 使用以下命令设置 HTTP 主机名：

  ```
  setmqweb properties -k httpHost -v hostName
  ```

  其中，*hostname* 指定 IP 地址、带域名后缀的域名服务器 (DNS) 主机名或安装了 IBM MQ 的服务器的 DNS 主机名。 使用双引号中的星号来指定所有可用的网络接口。 使用值 localhost 以仅允许本地连接。
- 使用以下命令取消设置 HTTP 主机名：

  ```
  setmqweb properties -k httpHost -d
  ```

配置 HTTP 和 HTTPS 端口

缺省情况下，托管 IBM MQ Console 和 REST API 的 mqweb 服务器使用 HTTPS 端口 9443。 已禁用与 HTTP 连接关联的端口。 您可以启用 HTTP 端口，配置其他 HTTPS 端口或禁用 HTTP 或 HTTPS 端口。 您可以使用 **setmqweb** 命令来配置端口。

开始之前

如果您启用 HTTP 端口，并且正在使用基于令牌的认证，那么必须启用同一 LTPA 令牌以用于 HTTP 和 HTTPS 连接。 有关更多信息，请参阅第 684 页的『配置 LTPA 令牌』。

要完成此任务，您必须具有使用 **dspmqweb** 和 **setmqweb** 命令的特定权限：

- z/OS 在 z/OS 上，您必须有权运行 **dspmqweb** 和 **setmqweb** 命令，并且必须具有 mqwebuser.xml 文件的写访问权。
- Multi 在所有其他操作系统上，您必须是特权用户。

注意： z/OS

在 z/OS 上发出 **setmqweb** 或 **dspmqweb** 命令之前，必须设置 WLP_USER_DIR 环境变量，以便该变量指向 mqweb 服务器配置。

为此，请发出以下命令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是传递到 crtmqweb 的目录的名称。 例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

有关更多信息，请参阅创建 mqweb 服务器。

过程

- 使用以下命令查看 HTTP 和 HTTPS 端口的当前配置:

  ```
  dspmqweb properties -a
  ```

  `httpPort` 字段显示 HTTP 端口， `httpsPort` 字段显示 HTTPS 端口。 有关更多信息，请参阅 dspmqweb。
- 通过使用以下命令启用或配置 HTTP 端口:
  - 使用以下命令启用或设置 HTTP 端口:

    ```
    setmqweb properties -k httpPort -v portNumber
    ```

    其中 *portNumber* 指定要用于 HTTP 连接的端口。 您可以使用值 `-1` 来禁用端口。
  - 使用以下命令将 HTTP 端口值重置为缺省值 `-1` :

    ```
    setmqweb properties -k httpPort -d
    ```
- 配置 HTTPS 端口:
  - 使用以下命令设置 HTTPS 端口号:

    ```
    setmqweb properties -k httpsPort -v portNumber
    ```

    其中 *portNumber* 指定要用于 HTTPS 连接的端口。 您可以使用值 `-1` 来禁用端口。
  - 使用以下命令将 HTTPS 端口号重置为缺省值 9443 :

    ```
    setmqweb properties -k httpsPort -d
    ```

配置响应超时

缺省情况下，如果将响应发送回客户机所花费的时间超过 30 秒，那么 IBM MQ Console 和 REST API 将超时。 您可以使用 **setmqweb** 命令将 IBM MQ Console 和 REST API 配置为使用不同的超时值。

开始之前

要完成此任务，您必须具有使用 **dspmqweb** 和 **setmqweb** 命令的特定权限:

- z/OS 在 z/OS 上，您必须有权运行 **dspmqweb** 和 **setmqweb** 命令，并且必须具有 `mqwebuser.xml` 文件的写访问权。
- Multi 在所有其他操作系统上，您必须是特权用户。

注意: z/OS

在 z/OS 上发出 **setmqweb** 或 **dspmqweb** 命令之前，必须设置 WLP_USER_DIR 环境变量，以便该变量指向 mqweb 服务器配置。

为此，请发出以下命令:

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是传递到 `crtmqweb` 的目录的名称。 例如:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

有关更多信息，请参阅创建 mqweb 服务器。

过程

- 使用以下命令查看请求超时的当前配置:

  ```
  dspmqweb properties -a
  ```

  `mqRestRequestTimeout` 字段显示响应超时的当前值。 有关更多信息，请参阅 dspmqweb。

- 使用以下命令设置请求超时:

  ```
  setmqweb properties -k mqRestRequestTimeout -v timeout
  ```

  其中 *timeout* 指定超时前的时间 (以秒为单位)。
- 使用以下命令将请求超时重置为缺省值 30 秒:

  ```
  setmqweb properties -k mqRestRequestTimeout -d
  ```

配置自动启动

缺省情况下， IBM MQ Console 会在 mqweb 服务器启动时自动启动。 您可以使用 **setmqweb** 命令来配置 IBM MQ Console 和 REST API 是否自动启动。

开始之前

要完成此任务，您必须具有使用 **dspmqweb** 和 **setmqweb** 命令的特定权限:

- z/OS 在 z/OS 上，您必须有权运行 **dspmqweb** 和 **setmqweb** 命令，并且必须具有 `mqwebuser.xml` 文件的写访问权。
- Multi 在所有其他操作系统上，您必须是特权用户。

注意: z/OS

在 z/OS 上发出 **setmqweb** 或 **dspmqweb** 命令之前，必须设置 WLP_USER_DIR 环境变量，以便该变量指向 mqweb 服务器配置。

为此，请发出以下命令:

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是传递到 `crtmqweb` 的目录的名称。 例如:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

有关更多信息，请参阅创建 mqweb 服务器。

过程

- 使用以下命令查看自动启动的当前配置:

  ```
  dspmqweb properties -a
  ```

  `mqRestAutostart` 字段显示 REST API 是否自动启动， `mqConsoleAutostart` 字段显示 IBM MQ Console 是否自动启动。 有关更多信息，请参阅 dspmqweb。
- 使用以下命令配置 IBM MQ Console 是否自动启动:

  ```
  setmqweb properties -k mqConsoleAutostart -v start
  ```

  其中 *start* 是值 `true` (如果您希望 IBM MQ Console 自动启动) 或 `false` 。
- 使用以下命令配置 REST API 是否自动启动:

  ```
  setmqweb properties -k mqRestAutostart -v start
  ```

  其中 *start* 是值 `true` (如果您希望 REST API 自动启动) 或 `false` 。

配置日志记录

您可以配置日志记录级别，最大日志文件大小以及托管 IBM MQ Console 和 REST API 的 mqweb 服务器所使用的最大日志文件数。 您可以使用 **setmqweb** 命令来配置日志记录。

开始之前

要完成此任务，您必须具有使用 **dspmqweb** 和 **setmqweb** 命令的特定权限：

- z/OS 在 z/OS 上，您必须有权运行 **dspmqweb** 和 **setmqweb** 命令，并且必须具有 `mqwebuser.xml` 文件的写访问权。
- Multi 在所有其他操作系统上，您必须是特权用户。

关于此任务

可以在下列其中一个目录中找到 mqweb 服务器的日志文件：

- ALW 在 AIX, Linux, and Windows 上：*MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb/logs
- z/OS 在 z/OS 上：*WLP_user_directory*/servers/mqweb/logs

  其中 *WLP_user_directory* 是运行 **crtmqweb** 脚本以创建 mqweb 服务器定义时指定的目录。

可在以下某个目录中找到 mqweb 服务器的消息传递跟踪文件：

- ALW 在 AIX, Linux, and Windows 上：*MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb
- z/OS 在 z/OS 上：*WLP_user_directory*/servers/mqweb

  其中 *WLP_user_directory* 是运行 **crtmqweb** 脚本以创建 mqweb 服务器定义时指定的目录。

有关对以下对象启用跟踪的更多信息：

- REST API，请参阅 跟踪 REST API
- V 9.2.0 New Web Console，请参阅 跟踪 New Web Console
- Dashboard Web Console，请参阅 跟踪 Dashboard Web Console

过程

- 使用以下命令查看 REST API 日志记录的当前配置：

  ```
  dspmqweb properties -a
  ```

  `maxTraceFileSize` 字段显示最大跟踪文件大小， `maxTraceFiles` 字段显示最大跟踪文件数，`traceSpec` 字段显示使用的跟踪级别。 此外， `maxMsgTraceFileSize` 字段显示最大消息传递跟踪文件大小， `maxMsgTraceFiles` 字段显示最大消息传递跟踪文件数。 有关更多信息，请参阅 dspmqweb。
- 配置最大日志文件大小：
  - 使用以下命令设置最大日志文件大小：

    ```
    setmqweb properties -k maxTraceFileSize -v size
    ```

    其中 *size* 指定每个日志文件可针对 IBM MQ Console 达到的大小 (以 MB 为单位)。
  - 使用以下命令将最大日志文件大小重置为缺省值 20 MB：

    ```
    setmqweb properties -k maxTraceFileSize -d
    ```
- 配置用于日志记录的最大文件数：
  - 使用以下命令设置用于日志记录的最大文件数：

    ```
    setmqweb properties -k maxTraceFiles -v max
    ```

    其中 *max* 指定 IBM MQ Console 的最大文件数。
  - 使用以下命令将用于记录的最大文件数重置为缺省值 2：

```
setmqweb properties -k maxTraceFiles -d
```

- 配置最大消息传递跟踪大小:
  - 使用以下命令设置最大消息跟踪大小:

    ```
    setmqweb properties -k maxMsgTraceFileSize -v size
    ```

    其中 *size* 指定每个日志文件可以达到的大小 (MB)。
  - 使用以下命令将最大日志文件大小重置为缺省值 200 MB:

    ```
    setmqweb properties -k maxMsgTraceFileSize -d
    ```
- 配置要使用的最大消息传递跟踪文件数:
  - 使用以下命令设置用于消息传递跟踪的最大文件数:

    ```
    setmqweb properties -k maxMsgTraceFiles -v max
    ```

    其中 *max* 指定最大文件数。
  - 使用以下命令将用于消息传递跟踪的最大文件数重置为缺省值 5:

    ```
    setmqweb properties -k maxMsgTraceFiles -d
    ```
- 配置所使用的日志记录级别:
  - 使用以下命令设置使用的日志记录级别:

    ```
    setmqweb properties -k traceSpec -v level
    ```

    其中 *level* 是 第 683 页的表 69 中列出的其中一个值。 该表概述了增加详细级别的日志记录级别。 当您启用日志记录级别时，还会在该级别之前启用每个级别。 例如，如果启用 ***=warning** 日志记录级别，那么还会启用 ***=severe** 和 ***=fatal** 日志记录级别。

    当 IBM 服务请求时更改此值。
  - 使用以下命令将使用的日志记录级别重置为缺省值 ***=info** :

    ```
    setmqweb properties -k traceSpec -d
    ```

表 *69:* 有效日志记录级别

值	**已应用日志记录级别**
* = 关闭	日志记录已关闭。
* = 致命	任务无法继续，组件，应用程序和服务器无法工作。
* = 严重	任务无法继续执行，但组件，应用程序和服务器仍可正常工作。 此级别也可能表明即将发生的不可恢复的错误。
* = 警告	可能发生错误或即将发生错误。 此级别还可以指示渐进故障 (例如，资源的潜在泄漏)。
*=audit	影响服务器状态或资源的重要事件
*=info	概述总体任务进度的一般信息
* = 配置	配置变更或状态
* = 详细信息	详细描述子任务进度的常规信息
* = 好	跟踪信息-常规跟踪 + 方法入口，出口和返回值
* = 更精细	跟踪信息-详细跟踪
* = 最优秀	跟踪信息-更详细的跟踪，包括调试问题所需的所有详细信息

表 69: 有效日志记录级别 (继续)	
值	**已应用日志记录级别**
* = 全部	记录所有事件

配置 LTPA 令牌

可以使用 LTPA 令牌来避免需要用户在对 mqweb 服务器的每个请求上提供用户名和密码凭证。 您可以使用 **setmqweb** 命令来配置 LTPA 令牌 cookie 的名称， LTPA 认证令牌的到期时间间隔以及是否可以由 HTTP 连接使用 LTPA 令牌。

开始之前

要完成此任务，您必须具有使用 **dspmqweb** 和 **setmqweb** 命令的特定权限：

- z/OS 在 z/OS 上，您必须有权运行 **dspmqweb** 和 **setmqweb** 命令，并且必须具有 `mqwebuser.xml` 文件的写访问权。
- Multi 在所有其他操作系统上，您必须是特权用户。

注: 如果同时将 IBM MQ Console 和令牌认证与 REST API 配合使用，那么将共享到期时间间隔。

注意: z/OS

在 z/OS 上发出 **setmqweb** 或 **dspmqweb** 命令之前，必须设置 WLP_USER_DIR 环境变量，以便该变量指向 mqweb 服务器配置。

为此，请发出以下命令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是传递到 `crtmqweb` 的目录的名称。 例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

有关更多信息，请参阅创建 mqweb 服务器。

关于此任务

当用户登录到 IBM MQ Console 时，将生成 LTPA 令牌。 如果将基于令牌的认证与 REST API 配合使用，那么当用户使用带有 HTTP POST 方法的 `/login` REST API 资源进行登录时，将生成 LTPA 令牌。 此令牌将在 cookie 中返回。 令牌用于认证用户，而无需用户使用其用户标识和密码再次登录，直到令牌到期为止。 缺省到期时间间隔为 120 分钟。

包含 LTPA 令牌的 cookie 的名称因平台而异：

- MQ Appliance 在 IBM MQ Appliance 上， LTPA 令牌为 `LtpaToken2`。 此值不能更改。
- z/OS ALW 缺省情况下，在所有其他平台上，包含 LTPA 令牌的 cookie 的名称以 `LtpaToken2` 开头，并包含可在 mqweb 服务器重新启动时更改的后缀。 此随机 cookie 名称允许多个 mqweb 服务器在同一系统上运行。 但是，如果您希望 cookie 名称保持一致值，那么可以使用 **setmqweb** 命令指定 cookie 的名称。

IBM i z/OS ALW 如果同时启用 HTTP 和 HTTPS 端口，那么可以将针对 HTTPS 请求发出的 LTPA 令牌用于 HTTP 请求。 缺省情况下会禁用此行为，但您可以使用 **setmqweb** 命令来启用此行为。

过程

- 使用以下命令查看 LTPA 令牌的当前到期时间， LTPA 令牌 cookie 的名称以及 LTPA 令牌是否可用于 HTTP 请求：

dspmqweb properties -a

- ltpaCookieName 字段显示 LTPA 令牌 cookie 的名称。 如果尚未设置 cookie 名称，那么此属性的值为 LtpaToken2_${env.MQWEB_LTPA_SUFFIX} on AIX, Linux, and Windows 或 LtpaToken2_${httpsPort} on z/OS， 。 mqweb 服务器使用 LtpaToken2_ 前缀后的变量为 cookie 生成唯一名称。 您无法设置此变量，但可以将 ltpaCookieName 更改为您选择的值。
- ltpaExpiration 字段显示 LTPA 令牌到期时间。
- 如果 LTPA 令牌可由 HTTP 请求使用，那么 secureLtpa 字段设置为 false 。

有关更多信息，请参阅 dspmqweb。

- 配置 LTPA 令牌到期:
  - 通过输入以下命令来设置 LTPA 令牌到期:

    setmqweb properties -k ltpaExpiration -v *time*

    其中 *time* 指定 LTPA 令牌到期和用户注销之前的时间 (以分钟计)。
  - 通过输入以下命令，将 LTPA 令牌到期时间重置为缺省值 120 分钟:

    setmqweb properties -k ltpaExpiration -d
- z/OS ALW

  配置 LTPA 令牌 cookie 名称:
  - 通过输入以下命令来设置 LTPA 令牌 cookie 名称:

    setmqweb properties -k ltpaCookieName -v *name*

    其中 *name* 指定 LTPA 令牌 cookie 的唯一名称。
  - 通过输入以下命令，将 LTPA 令牌 cookie 名称重置为缺省值，其中前缀 LtpaToken2_ 后跟随机字符:

    setmqweb properties -k ltpaCookieName -d
- z/OS ALW

  通过输入以下命令来配置 LTPA 令牌是否可由 HTTP 连接使用:

  setmqweb properties -k secureLtpa -v *secure*

  其中 *secure* 指定 LTPA 令牌是否可由不安全的 HTTP 连接和安全的 HTTPS 连接使用。 值 false 允许 HTTP 和 HTTPS 连接使用相同的 LTPA 令牌。

V 9.2.3 配置远程队列管理器连接行为

使用 IBM MQ Console 时，可以创建与远程队列管理器的连接。 即，您可以连接到与运行 IBM MQ Console 的 mqweb 服务器不属于同一安装的队列管理器。 您可以设置许多配置选项来控制远程队列管理器连接的行为。

开始之前

要完成此任务，您必须具有使用 **dspmqweb** 和 **setmqweb** 命令的特定权限:

- z/OS 在 z/OS 上，您必须有权运行 **dspmqweb** 和 **setmqweb** 命令，并且必须具有 mqwebuser.xml 文件的写访问权。
- Multi 在所有其他操作系统上，您必须是特权用户。

关于此任务

您可以设置以下配置选项:

- 是否允许远程队列管理器连接。
- 是可以使用 IBM MQ Console 添加连接，还是只能使用命令行添加连接。
- 当允许远程队列管理器连接时，是否在 IBM MQ Console 中显示本地队列管理器。

- 是在启动 IBM MQ Console 时自动建立远程队列管理器连接，还是在发生连接故障时自动建立远程队列管理器连接。
- 每次刷新 IBM MQ Console 中显示的远程队列管理器列表之间的时间长度。

过程

- 要查看当前远程队列管理器连接配置设置，请输入以下命令:

  ```
  dspmqweb properties -a
  ```

  - `mqConsoleRemoteSupportEnabled` 字段指示是否允许远程队列管理器连接。
  - `mqConsoleRemoteUIAdmin` 字段指示是否可以使用 IBM MQ Console 添加远程队列管理器连接。
  - `mqConsoleRemoteAllowLocal` 字段指示是否显示本地队列管理器。
  - `mqConsoleRemotePollTime` 字段指示每次刷新远程队列管理器列表之间的秒数。

  有关更多信息，请参阅 dspmqweb。
- 要阻止或允许远程队列管理器与 IBM MQ Console 连接，请输入以下命令:

  ```
  setmqweb properties -k mqConsoleRemoteSupportEnabled -v enabled
  ```

  其中， *enabled* 设置为 `true` 以允许远程队列管理器连接，或者设置为 `false` 以阻止远程队列管理器连接。
- 要防止或允许使用 IBM MQ Console 或仅通过命令行添加远程队列管理器连接，请输入以下命令:

  ```
  setmqweb properties -k mqConsoleRemoteUIAdmin -v enabled
  ```

  其中， *enabled* 设置为 `true` 以允许使用 IBM MQ Console 和命令行添加远程队列管理器连接，或者设置为 `false` 以允许仅使用命令行上的 **setmqweb remote** 命令添加远程队列管理器连接。
- 要在允许远程队列管理器连接时阻止或允许显示 IBM MQ Console 中的本地队列管理器，请输入以下命令:

  ```
  setmqweb properties -k mqConsoleRemoteAllowLocal -v enabled
  ```

  其中 *enabled* 设置为 `true` 以允许显示本地队列管理器，或者设置为 `false` 以隐藏本地队列管理器。
- 要设置每次刷新 IBM MQ Console 中显示的远程队列管理器列表之间的时间长度，请输入以下命令:

  ```
  setmqweb properties -k mqConsoleRemotePollTime -v seconds
  ```

  其中， *seconds* 设置为每次刷新远程队列管理器列表之间的秒数的整数值。

相关参考

setmqweb

长石

配置 administrative REST API 网关

缺省情况下， administrative REST API 网关处于启用状态。 启用 administrative REST API 网关后，可以使用网关队列管理器对 REST API 执行远程管理。 您可以配置用作缺省网关队列管理器的队列管理器，也可以通过使用 **setmqweb** 命令禁用 administrative REST API 网关来阻止远程管理。

关于此任务

要完成此任务，您必须具有使用 **dspmqweb** 和 **setmqweb** 命令的特定权限:

- z/OS 在 z/OS 上，您必须有权运行 **dspmqweb** 和 **setmqweb** 命令，并且必须具有 `mqwebuser.xml` 文件的写访问权。
- Multi 在所有其他操作系统上，您必须是特权用户。

当以下两个语句都为 true 时，将使用缺省网关队列管理器:

- 未在 REST 请求的 `ibm-mq-rest-gateway-qmgr` 头中指定队列管理器。

- REST API 资源 URL 中指定的队列管理器不是本地队列管理器。

有关使用 REST API 进行远程管理的更多信息，请参阅 使用 REST API 进行远程管理。

注意: z/OS

在 z/OS 上发出 **setmqweb** 或 **dspmqweb** 命令之前，必须设置 WLP_USER_DIR 环境变量，以便该变量指向 mqweb 服务器配置。

为此，请发出以下命令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是传递到 crtmqweb 的目录的名称。 例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

有关更多信息，请参阅创建 mqweb 服务器。

过程

- 使用以下命令查看 administrative REST API 网关的当前配置：

  ```
  dspmqweb properties -a
  ```

  mqRestGatewayEnabled 字段显示是否启用网关， mqRestGatewayQmgr 字段显示缺省网关队列管理器的名称。 有关更多信息，请参阅 dspmqweb。
- 使用以下命令配置是否启用 administrative REST API 网关：

  ```
  setmqweb properties -k mqRestGatewayEnabled -v enabled
  ```

  其中 *enabled* 是用于启用 administrative REST API 网关的值 **true** ，否则为 **false** 。
- 配置将哪个队列管理器用作缺省网关队列管理器：
  - 使用以下命令设置缺省网关队列管理器：

    ```
    setmqweb properties -k mqRestGatewayQmgr -v qmgrName
    ```

    其中 *qmgrName* 是与 mqweb 服务器在同一安装中的队列管理器的名称。
  - 使用以下命令取消设置缺省网关队列管理器：

    ```
    setmqweb properties -k mqRestGatewayQmgr -d
    ```

配置 messaging REST API

缺省情况下，托管 IBM MQ Console 和 REST API 的 mqweb 服务器已启用 messaging REST API 。

V 9.2.0 从 IBM MQ 9.2.0 开始，从 messaging REST API 到 IBM MQ 的连接将合用，每个队列管理器有 20 个可用连接。 当所有连接都在使用时， messaging REST API 会创建新的非合用连接以用于请求。 您可以使用 **setmqweb properties** 命令来更改合用连接的最大数目以及使用所有连接时 messaging REST API 的行为。 您还可以 使用 **setmqweb properties** 命令配置是否启用消息传递。

开始之前

要完成此任务，您必须具有使用 **dspmqweb** 和 **setmqweb** 命令的特定权限：

- z/OS 在 z/OS 上，您必须有权运行 **dspmqweb** 和 **setmqweb** 命令，并且必须具有 mqwebuser.xml 文件的写访问权。
- Multi 在所有其他操作系统上，您必须是特权用户。

要使用 messaging REST API ，必须向 mqweb 服务器认证调用者，并且该调用者必须是 MQWebUser 角色的成员。 MQWebAdmin 和 MQWebAdminRO 角色不适用于 messaging REST API。 还必须授权调用者通过 OAM 或 RACF 访问用于消息传递的队列。 有关 REST API 的安全性的更多信息，请参阅 IBM MQ Console 和 REST API 安全性。

注意: z/OS

在 z/OS 上发出 **setmqweb** 或 **dspmqweb** 命令之前，必须设置 WLP_USER_DIR 环境变量，以便该变量指向 mqweb 服务器配置。

为此，请发出以下命令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是传递到 crtmqweb 的目录的名称。 例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

有关更多信息，请参阅创建 mqweb 服务器。

关于此任务

您可以使用 **setmqweb properties** 命令来配置是否启用消息传递。

V 9.2.0 为了优化 messaging REST API 的性能，将汇聚到 IBM MQ 队列管理器的连接。 即，每个 REST 请求使用来自连接池的连接，而不是创建，使用和销毁其自己的连接。 缺省情况下，每个队列管理器池有 20 个连接可用，当所有连接都在使用中时，有三个选项可用于处理请求：

- messaging REST API 可以创建新的非合用连接以用于请求。 此行为是缺省行为。
- messaging REST API 可能会返回错误。
- messaging REST API 可以等待合用的连接变为可用。 此等待是无限期等待。

通过使用 **setmqweb properties** 命令，可以更改所有连接正在使用时合用的最大连接数以及 messaging REST API 的缺省行为。

过程

- 使用以下命令查看 HTTP messaging REST API 的当前配置：

  ```
  dspmqweb properties -a
  ```

  mqRestMessagingEnabled 字段显示是否启用了 messaging REST API 。 值为 True 表示已启用 messaging REST API ，否则为 False 。 有关更多信息，请参阅 dspmqweb。
- 使用以下命令配置 messaging REST API ：

  ```
  setmqweb properties -k mqRestMessagingEnabled -v enabled
  ```

  其中， *enabled* 是值 true (如果要启用 messaging REST API) ，否则为 false 。
- V 9.2.0

  配置 messaging REST API 的连接池：
  - 使用以下命令为每个队列管理器池配置最大连接池大小：

    ```
    setmqweb properties -k mqRestMessagingMaxPoolSize -v size
    ```

    其中 *size* 指定池的大小。

    注: 如果已设置 *mqRestMessagingMaxPoolSize* 的较大值，并且正在连接大量队列管理器，那么应考虑增大 mqweb 服务器堆的最大大小。

    请参阅 调整 mqweb 服务器 JVM 以获取更多信息。
  - 使用以下命令在使用池中的所有连接时配置 messaging REST API 的行为：

    ```
    setmqweb properties -k mqRestMessagingFullPoolBehavior -v action
    ```

    其中 *action* 指定要执行的操作。 *action* 可以是下列其中一个值：

    块
    当池中的所有连接都在使用中时，请等待连接变为可用。

Error
当池中的所有连接都在使用中时，返回错误。

溢出
当池中的所有连接都在使用时，请创建要使用的非合用连接，并在使用该连接后将其废弃。

为 MFT 配置 REST API

CD V 9.2.0 缺省情况下，托管 IBM MQ Console 和 REST 服务的 mqweb 服务器未启用 MFT 。 您可以配置是否启用 REST API for MFT ，设置协调队列管理器，并使用 **setmqweb properties** 命令指定 MFT 重新连接超时。 V 9.2.0 对于类似于 **Create transfer** 的 create REST API 命令，必须添加 mqRestMftCommandQmgr。

开始之前

V 9.2.0 您需要：

- 要行使 MFT REST 服务，您需要通过将 **mqRestMftEnabled** 设置为 *true* 来启用 MFT REST 服务，并将命令队列管理器名称设置为 mqRestMftCoordinationQmgr:

```
setmqweb properties -k mqRestMftEnabled -v true
setmqweb properties -k mqRestMftCoordinationQmgr -v <coordinationQmgrName>
```

- 要提交任何创建请求 (例如，创建传输或资源监视器) ，您需要设置 mqRestMftCommandQmgr

```
setmqweb properties -k mqRestMftCommandQmgr -v <commandQmgrName>
```

- 通过发出后跟 **strmqweb** 命令的 **endmqweb** 命令，重新启动 mqweb 服务器以强制实施刚刚设置的值。
- 通过发出 **dspmqweb** 命令来检查 Web 服务器的状态。

要完成此任务，您必须具有使用 **dspmqweb** 和 **setmqweb** 命令的特定权限：

- z/OS 在 z/OS 上，您必须有权运行 **dspmqweb** 和 **setmqweb** 命令，并且必须具有 mqwebuser.xml 文件的写访问权。
- Multi 在所有其他操作系统上，您必须是特权用户。

要使用 REST API for MFT，调用者必须向 mqweb 服务器进行认证，并且必须是一个或多个 MFTWebAdmin 或 MFTWebAdminRO 角色的成员。

注意: z/OS

在 z/OS 上发出 **setmqweb** 或 **dspmqweb** 命令之前，必须设置 WLP_USER_DIR 环境变量，以便该变量指向 mqweb 服务器配置。

为此，请发出以下命令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是传递到 crtmqweb 的目录的名称。 例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

有关更多信息，请参阅创建 mqweb 服务器。

关于此任务

当您配置 REST API for MFT 超时时，请注意，第一次尝试重新建立连接是在与协调队列管理器的连接中断后立即进行的。 如果此操作失败，那么每次重新连接尝试之间的时间间隔为 5 分钟。 因此，设置 0-5 之间的值只会导致一次尝试重新连接。

重新连接超时后，将在调用任何 REST API for MFT 资源时进行下一次重新连接尝试。 如果此重新连接尝试失败，那么 MFT 将再次尝试每 5 分钟重新连接一次，直到重新连接超时为止。

过程

- 使用以下命令查看 REST API for MFT 的当前配置:

  ```
  dspmqweb properties -a
  ```

  `mqRestMftEnabled` 字段显示是否已启用 REST API for MFT 。 如果启用了 messaging REST API ，那么值为 `True` ，否则值为 `False` 。

  `mqRestMftCoordinationQmgr` 字段显示协调队列管理器的名称。

  V 9.2.0 `mqRestMftCommandQmgr` 字段显示命令队列管理器的名称。

  `mqRestMftReconnectTimeoutInMinutes` 字段显示重新连接超时值，直到 MFT 传输 REST 服务停止尝试连接到协调队列管理器为止。 有关更多信息，请参阅 dspmqweb。
- 配置是否启用 REST API for MFT :

  a) 通过输入以下命令来配置是否启用 REST API for MFT :

  ```
  setmqweb properties -k mqRestMftEnabled -v value
  ```

  其中，如果要启用 REST API for MFT ，那么 *value* 为 `true` ，否则为 `false` 。

  b) 通过输入以下命令来重新启动 mqweb 服务器:

  ```
  endmqweb
  strmqweb
  ```
- 配置从中检索传输详细信息的协调队列管理器:

  a) 使用以下命令配置协调队列管理器:

  ```
  setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
  ```

  其中 *qmgrName* 是协调队列管理器的名称。 协调队列管理器必须位于运行 mqweb 服务器的机器上。缺省情况下，此队列管理器名称为空白。 如果未设置值，那么 MFT 的 REST API 不起作用。

  b) 通过输入以下命令来重新启动 mqweb 服务器:

  ```
  endmqweb
  strmqweb
  ```
- 配置超时 (以分钟计) ，在此时间之后， REST API for MFT 将停止尝试连接到协调队列管理器:

  a) 使用下列其中一个命令来配置超时:

  – 设置超时:

  ```
  setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -v time
  ```

  其中 *time* 指定发生超时之前的时间 (以分钟为单位)。

  如果在 `0-5` 之间设置此值，那么 REST API for MFT 仅尝试重新连接到协调队列管理器一次。 如果连接失败，那么在调用 REST API 之前不会尝试重新建立连接。

  如果此值设置为 `-1`，那么 REST API for MFT 将尝试重新连接，直到连接成功为止。

  – 将超时重置为缺省值 30 分钟:

  ```
  setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -d
  ```

  b) 通过输入以下命令来重新启动 mqweb 服务器:

  ```
  endmqweb
  strmqweb
  ```
- V 9.2.0 为创建请求配置命令队列管理器:

  a) 使用以下命令配置命令队列管理器:

```
setmqweb properties -k mqRestMftCommandQmgr -v qmgrName
```

其中 *qmgrName* 是命令队列管理器的名称。 命令队列管理器必须位于运行 mqweb 服务器的机器上。 缺省情况下，此队列管理器名称为空白。 如果未设置值，那么 create 命令的 MFT REST API 不起作用。

b) 通过输入以下命令来重新启动 mqweb 服务器:

```
endmqweb
strmqweb
```

调整 mqweb 服务器 JVM

缺省情况下， mqweb 服务器 Java 虚拟机 (JVM) 将特定于平台的缺省值用于配置参数，例如堆的最小和最大大小以及类高速缓存的大小。

关于此任务

您可能需要更改缺省值以提高性能或解决问题。 例如，如果 mqweb 服务器抛出 `java.lang.OutOfMemoryError`，那么必须增大堆的最大大小。 如果尝试装入大量队列对象，那么还应增大堆的大小。

如果在 IBM MQ Console 中显示仪表板配置信息时迁到问题，那么必须设置用于确定配置的文件编码的变量。

您可以更改 `jvm.options` 文件中的缺省值。

过程

1. 打开 `jvm.options` 文件。

   可以在下列其中一个目录中找到 `jvm.options` 文件:

   - ALW 在 AIX, Linux, and Windows 上：*MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb
   - z/OS 在 z/OS 上：*WLP_user_directory*/servers/mqweb

     其中 *WLP_user_directory* 是运行 **crtmqweb** 脚本以创建 mqweb 服务器定义时指定的目录。
2. 可选： 通过将以下行添加到文件来设置最大堆大小:

   ```
   -Xmx MaxSizem
   ```

   其中 *MaxSize* 指定堆的最大大小 (以 MB 为单位)。

   例如，以下行将最大堆大小设置为 1GB:

   ```
   -Xmx1024m
   ```
3. 可选： 通过将以下行添加到文件来设置最小堆大小:

   ```
   -Xms MinSizem
   ```

   其中 *MinSize* 指定堆的最小大小 (以 MB 为单位)。 从缺省值增加最小堆大小可减少启动 mqweb 服务器所需的时间。

   例如，以下行将最小堆大小设置为 512MB:

   ```
   -Xms512m
   ```
4. 可选： 通过将以下行添加到文件来设置类高速缓存的大小:

   ```
   -Xscmx Sizem
   ```

   其中 *Size* 指定类高速缓存大小 (以 MB 为单位)。

例如，以下行将类高速缓存大小设置为 100MB:

```
-Xscmx100m
```

Java 共享类高速缓存用于存储数据，例如装入的类和 Ahead-Of-Time (AOT) 编译的代码。

类高速缓存可显着减少启动 mqweb 服务器所花费的时间。 首次启动 mqweb 服务器时，将创建类高速缓存，并且服务器可能需要很长时间才能启动。 服务器的后续重新启动将更快，因为可以从共享类高速缓存装入类。

从缺省值增加类高速缓存的大小可以减少启动 mqweb 服务器所需的时间。

z/OS 在其他 z/OS 系统上启动 mqweb 服务器时，将重新创建类高速缓存。 因此，在综合系统中的另一个 z/OS 系统上启动 mqweb 服务器可能要比在同一系统上重新启动服务器花费更长时间。

请注意，对此值所作的更改仅在创建类高速缓存时生效。 类高速缓存是在 mqweb 服务器首次启动时或在使用 Java 类高速缓存实用程序销毁类高速缓存之后创建的。

5. 必需： 检查该文件是否包含以下行，以指定 REST API 处理数据时使用的文件编码以及 New Web Console 中的用户仪表板配置信息:

```
-Dfile.encoding=UTF-8
-Ddefault.client.encoding=UTF-8
```

6. 重新启动 mqweb 服务器。

   z/OS 在 z/OS 上，停止并重新启动 mqweb 服务器启动式任务。

   Multi 在所有其他平台上，在命令行上输入以下命令:

```
endmqweb
strmqweb
```

IBM MQ Console 和 REST API 安装组件的文件结构

有两组目录结构与 IBM MQ Console 和 REST API 安装组件相关联。 一个目录结构包含可编辑的文件。 其他目录结构包含无法编辑的文件。

可编辑文件

用户可编辑文件作为 IBM MQ Console 和 REST API 安装组件初始安装的一部分进行布局。 由于可以编辑这些文件，因此在应用维护时不会更改这些文件。

用户可编辑文件的位置取决于操作系统:

- ALW 在 AIX, Linux, and Windows 上：*MQ_DATA_PATH*/web/installations/*installationName*/
- IBM i 在 IBM i 上：*MQ_DATA_PATH*/web/installations/Installation1/
- z/OS 在 z/OS 上：*WLP_user_directory*

  其中 *WLP_user_directory* 是运行 **crtmqweb** 脚本以创建 mqweb 服务器定义时指定的目录。

在此顶级目录下，存在以下目录和文件:

目录和文件	描述
angular.persistence/	存储 IBM MQ Console 仪表板配置的目录。
servers/	WebSphere Liberty Profile 服务器目录。
servers/mqweb	包含 mqweb 服务器目录结构的目录。
servers/mqweb/logs	包含 mqweb 服务器日志的目录。

目录和文件	描述
servers/mqweb/logs/console.log	基本服务器状态和操作消息的日志。
servers/mqweb/logs/ffdc	首次故障数据捕获 (FFDC) 输出目录。
servers/mqweb/logs/messages.log	来自 mqweb 服务器的运行时消息的日志，包括 IBM MQ Console 和 REST API。 较旧的消息存储在名为 messages_*timestamp*.log 的文件中。
servers/mqweb/logs/trace.log	来自 mqweb 服务器的跟踪日志，包括 IBM MQ Console 和 REST API。 较旧的跟踪存储在名为 trace_*timestamp*.log 的文件中。 仅当启用了跟踪时，这些文件才存在。
servers/mqweb/logs/state	特定于服务器的状态。
servers/mqweb/server.xml	主服务器配置文件。 此文件是只读的。 编辑 mqwebuser.xml 文件以覆盖缺省配置。
servers/mqweb/mqwebuser.xml	IBM MQ Console 和 REST API 的配置文件。 此文件中配置的设置将覆盖缺省配置。 您必须是 特权用户 才能编辑此文件。
servers/mqweb/resources	包含各种服务器资源 (例如密钥库) 的目录。
servers/mqweb/workarea	服务器在运行时创建的目录。 此目录是在服务器首次运行后创建的。

不可编辑的文件

不可编辑的文件将作为 IBM MQ Console 和 REST API 安装组件的初始安装的一部分进行布局。 应用维护时，将更新这些文件。

用户可编辑文件的位置取决于操作系统:

- ALW 在 AIX, Linux, and Windows 上: *MQ_INSTALLATION_PATH*/web
- IBM i 在 IBM i 上: *MQ_INSTALLATION_PATH*/web
- z/OS 在 z/OS 上： *installation_directory*/web/

  其中 *installation_directory* 是 IBM MQ for z/OS UNIX System Services Components 安装路径。

此位置中存在以下目录结构和文件:

目录和文件	描述
bin/	包含 Liberty 命令的目录。 您必须是 特权用户 才能在此目录中执行脚本。
mq/	包含各种 IBM MQ 资源的目录结构。
mq/apps/	包含 IBM MQ Console 和 REST API 应用程序的目录。
mq/etc/	
mq/etc/mqweb.xml	mqweb 服务器的只读配置文件。 编辑 mqwebuser.xml 文件以进行配置更改。

目录和文件	描述
mq/libs	包含供 IBM MQ Console 和 REST API 使用的共享库的目录。
mq/samp	包含样本的目录。
mq/samp/configuration	包含可复制到 mqwebuser.xml 文件中的样本配置文件的目录。

Windows MQ Adv. Linux MQ Adv. VUE V 9.2.0 在 Linux 或 Windows 上定义 Aspera gateway 连接

IBM Aspera fasp.io Gateway 提供了快速 TCP/IP 隧道，可以显着提高 IBM MQ 的网络吞吐量。 在任何授权平台上运行的队列管理器都可以通过 Aspera gateway 进行连接。 网关本身部署在 Red Hat 或 Ubuntu Linux 或 Windows 上。

关于此任务

Aspera gateway 可用于提高队列管理器通道的性能。 如果网络具有高延迟或倾向于丢失包，那么它尤其有效，并且通常用于加快不同数据中心的队列管理器之间的连接。

注: 对于不丢失包的快速网络，在使用 Aspera gateway 时性能下降，因此在定义 Aspera gateway 连接之前和之后检查网络性能很重要。

在 IP 网络连接的每一端定义一个 Aspera gateway ，然后使用 TCP/IP 将队列管理器通道连接到每个网关。 队列管理器不需要与其使用的 Aspera gateway 在同一机器上运行，并且多个队列管理器可以使用同一网关。 唯一的限制如下:

要使用 Aspera gateway，您必须具有以下一项或多项权利:

- IBM MQ Advanced for Multiplatforms
- IBM MQ Appliance
- IBM MQ Advanced for z/OS VUE
- LTS IBM MQ Advanced for z/OS (仅限 Long Term Support)

您可以在以下任何 Linux (Red Hat 或 Ubuntu) 平台或 Windows 上部署 Aspera gateway :

- Linux for x86-64
- Linux on POWER Systems - Little Endian
- Linux for IBM Z
- V 9.2.0 V 9.2.0 Windows -有关 Windows 上的平台支持的更多信息，请参阅 IBM Aspera fasp.io Gateway 文档。

除非网关单独授权，否则 Aspera gateway 的使用仅限于 IBM MQ 消息。

使用 Aspera gateway 的队列管理器可以在任何受支持的平台上运行。 有关受支持平台的完整列表，请参阅产品文档中的发行版和平台图标。

对于与其使用的 Aspera gateway 不在同一机器上的每个队列管理器，请检查队列管理器与 Aspera gateway 之间是否具有快速网络连接。

您可以使用 toml 文件来创建用于定义网关所使用的入站和出站端口的网关定义。 样本 toml 文件随 Aspera gateway 一起提供。 出站网关定义定义从本地队列管理器到网关的连接，以及从本地网关到远程网关的连接。 入站网关定义定义从远程网关到本地网关的连接，以及从本地网关到本地队列管理器的连接。

以下步骤提供了入门和熟悉运用的基本指南。 有关更多详细信息，请参阅 IBM Aspera fasp.io Gateway 文档。

过程

1. 获取 Aspera gateway 安装映像。

Multi 对于多平台，从 Passport Advantage 下载 Aspera gateway 。 下载标记为 “IBM Aspera fasp.io Continuous Delivery Release for IBM MQ V9.2.x Multiplatform Multilingual eAssembly”。 它仅作为 Continuous Delivery (CD) 映像交付，因为此区域的更改速度加快，这意味着需要按 CD 发行版的频率进行更新，并且您可以将其安装在具有 IBM MQ Advanced for Multiplatforms 或 IBM MQ Appliance 权利的任何 IBM MQ 系统上。 要下载此文件，请访问 eAssembly, ， 进入下载页面 IBM MQ 9.2 ，然后点击最新版本的选项卡。 eAssembly 包含网关可用的所有平台的安装映像。

MQ Adv. VUE MQ Adv. z/OS 如果 IBM MQ 系统具有 IBM MQ Advanced for z/OS VUE 权利或 Long Term Support IBM MQ Advanced for z/OS 权利，那么您将从属于 SMP/E 安装的连接器包组件获取 Aspera gateway 。 有关更多信息，请参阅 IBM MQ Advanced for z/OS VUE 或 IBM MQ Advanced for z/OS 程序目录 (IBM MQ for z/OS Program Directory PDF 文件)。 安装连接器包时，它会在 z/OS UNIX System Services 中创建包含一组压缩文件的 `fasp` 目录。 这些文件包含网关可用的所有平台的安装映像。

z/OS MQ Adv. VUE MQ Adv. z/OS IBM MQ Advanced for z/OS VUE 和 IBM MQ Advanced for z/OS 的文件如下所示:

表 *70: Long Term Support* 和 *Continuous Delivery* 的文件名

平台	文件名	fasp.io 版本号
Linux for x86-64	V 9.2.0 V 9.2.0 CC6UIEN.zip	1.1.0
	V 9.2.2 CC8R1EN.zip	1.1.1 第 696 页的『1.b』
	V 9.2.3 G01WVEN.zip	1.2.0 第 696 页的『1.c』
	V 9.2.0 M0LDNEN.zip	1.3.5 第 696 页的『1.d』
Linux on POWER Systems - Little Endian	V 9.2.0 V 9.2.0 CC6UKEN.zip	1.1.0
	V 9.2.2 CC8R2EN.zip	1.1.1 第 696 页的『1.b』
	V 9.2.3 G01WWEN.zip	1.2.0 第 696 页的『1.c』
	V 9.2.0 M0LDPEN.zip	1.3.5 第 696 页的『1.d』
Linux for IBM Z	V 9.2.0 V 9.2.0 CC6ULEN.zip	1.1.0
	V 9.2.2 CC8R3EN.zip	1.1.1 第 696 页的『1.b』
	V 9.2.3 G01WXEN.zip	1.2.0 第 696 页的『1.c』
	V 9.2.0 M0LDQEN.zip	1.3.5 第 696 页的『1.d』

表 70: *Long Term Support* 和 *Continuous Delivery* 的文件名 (继续)		
平台	**文件名**	**fasp.io 版本号**
Windows	V 9.2.0 V 9.2.0 CC6UMEN.zip	1.1.0
	V 9.2.2 CC8R4EN.zip	1.1.1 第 696 页的『1.b』
	V 9.2.3 G01WYEN.zip	1.2.0 第 696 页的『1.c』
	V 9.2.0 MOLDREN.zip	1.3.5 第 696 页的『1.d』

注意:

a. CD V 9.2.0 Aspera gateway 无法在 z/OS 上本地运行。

b. V 9.2.2 fasp.io V 1.1.1 由 IBM MQ 9.2.0 上的 APAR PH35485 提供。

c. V 9.2.3 fasp.io V 1.2.0 由 IBM MQ 9.2.0 上的 APAR PH38832 提供。

d. V 9.2.0 fasp.io 版本 由 APAR 在 上提供。 1.3.5 PH63366 IBM MQ 9.2.0

2. 将 Aspera gateway 安装映像复制到将运行网关的两台机器上，然后解压缩并安装网关。

Linux

使用 RPM Package Manager (RPM) 进行安装:

```
rpm -ivh ibm-fasp.io-gateway-<version>.x86_64.rpm
```

要在 Ubuntu 上使用 RPM 进行安装，您有两个选项:

- 将 `--force-debian` 属性添加到 Aspera gateway install 命令，如 使用 rpm 在 Linux 上安装 IBM MQ 客户机中所述。
- 使用 **apt-get** 命令:

```
sudo apt-get install ./ibm-fasp.io-gateway_<version>_amd64.deb
```

Windows V 9.2.0 V 9.2.0

要在 Windows 上安装 Aspera gateway :

a. 确保您在受支持的 Windows 机器上，并使用具有管理员特权的帐户登录。

b. 下载安装程序。 缺省位置为 Downloads 文件夹: `ibm-fasp.io-gateway_<version>_win64.msi`。

c. 打开 `.msi` 文件。 安装向导启动时，请遵循提示完成安装。

3. 配置每个网关。

修改安装所创建的目录中的 `gateway.toml` 和 `logging.toml` 文件:

Linux

V 9.2.3 从 IBM Aspera fasp.io Gateway 1.2 开始，位置为 `/usr/local/etc/faspio/`。

V 9.2.2 从 IBM Aspera fasp.io Gateway 1.1 开始，位置为 `/usr/local/etc/fasp.io/`。

对于较早版本的 Aspera gateway，位置为 `/etc/fasp.io/`。

Windows

V 9.2.0 V 9.2.0 `C:\Program Files\IBM\fasp.io Gateway\config\`

使用 gateway.toml 文件来定义网关使用的入站和出站端口，使用 logging.toml 文件来定义所需的日志记录级别。 稍后在本主题中提供了编辑 gateway.toml 文件的示例。

4. 在网络连接的每一端，更改通道定义以连接到本地网关正在侦听的端口。
5. 启动每个网关服务。
   从命令提示符，运行以下命令:
   - Linux
     ```
     systemctl start fasp.io-gateway
     ```
   - Windows V 9.2.0 V 9.2.0
     ```
     net start fasp.io-gateway
     ```
6. 重新启动通道。

   现在，您的队列管理器正在通过 Aspera gateway 连接进行通信。

示例

此示例定义运行 Linux 的两台机器上的 Aspera gateway 连接。 该配置如下所示:

- 本地网关机器的 IP 地址为 9.20.193.107。 远程网关机器的 IP 地址为 9.20.192.115。
- 本地队列管理器正在 IP 地址为 9.20.121.5 的机器上运行。 远程队列管理器正在 IP 地址为 9.20.121.25 的机器上运行。 两个队列管理器都在侦听端口 1414。
- 本地队列管理器上的队列管理器通道已更改为使用 **conname** 9.20.193.107(1500) 连接到本地 Aspera gateway 。 远程队列管理器上的队列管理器通道已更改为使用 **conname** 9.20.192.115(1500) 连接到远程 Aspera gateway 。
- V 9.2.3 缺省情况下， IBM Aspera fasp.io Gateway 1.2 启用 TLS。 在此示例中，已通过在网关定义中设置 tls_enabled=false 来禁用 TLS。 如果与较低版本的网关一起运行，请删除 tls_enabled=false 行。 如果要使用网关配置 TLS ，请参阅 IBM Aspera fasp.io Gateway 文档中的 保护网关 。

1. 在本地网关机器上定义 Aspera gateway 连接:
   - 安装 Aspera gateway:
     - Linux 在 Linux 上，使用以下命令:
       ```
       rpm -ivh ibm-fasp.io-gateway-<version>.x86_64.rpm
       ```
     - Windows V 9.2.0 V 9.2.0 在 Windows 上，打开 .msi 文件以运行安装向导。
   - 修改安装所创建的目录中的 gateway.toml 文件:

     Linux

     V 9.2.3 从 IBM Aspera fasp.io Gateway 1.2 开始，位置为 /usr/local/etc/faspio/。

     V 9.2.2 从 IBM Aspera fasp.io Gateway 1.1 开始，目录为 /usr/local/etc/fasp.io/。

     对于较早版本的 Aspera gateway，目录为 /etc/fasp.io/。

     Windows

     V 9.2.0 V 9.2.0 C:\Program Files\IBM\fasp.io Gateway\config\

     编辑文件以设置本地网关定义。

     ```
     [[bridge]]
         name = "Outbound"
         [bridge.local]
                protocol = "tcp"
                    host = "9.20.193.107"
                    port = 1500
     ```

```
    tls_enabled = false

    [bridge.forward]
           protocol = "fasp"
               host = "9.20.192.115"
               port = 1600
    tls_enabled = false

[[bridge]]
    name = "Inbound"
    [bridge.local]
           protocol = "fasp"
               host = "9.20.193.107"
               port = 1600
    tls_enabled = false

    [bridge.forward]
           protocol = "tcp"
               host = "9.20.121.5"
               port = 1414
    tls_enabled = false
```

2. 重复上一步以在远程网关机器上定义 Aspera gateway 连接。 修改安装所创建的目录中的 `gateway.toml` 文件:

   Linux

   V 9.2.2 从 IBM Aspera fasp.io Gateway 1.1 开始，目录为 `/usr/local/etc/fasp.io/`。 对于较早版本的 Aspera gateway，目录为 `/etc/fasp.io/`。

   Windows

   V 9.2.0 V 9.2.0 `C:\Program Files\IBM\fasp.io Gateway\config\`

   编辑文件以设置远程网关定义:

```
[[bridge]]
    name = "Outbound"
    [bridge.local]
           protocol = "tcp"
               host = "9.20.193.107"
               port = 1500
    tls_enabled = false

    [bridge.forward]
           protocol = "fasp"
               host = "9.20.192.115"
               port = 1600
    tls_enabled = false

[[bridge]]
    name = "Inbound"
    [bridge.local]
           protocol = "fasp"
               host = "9.20.193.107"
               port = 1600
    tls_enabled = false

    [bridge.forward]
           protocol = "tcp"
               host = "9.20.121.5"
               port = 1414
    tls_enabled = false
```

3. 在连接的每一端，更改通道定义以连接到本地网关正在侦听的端口。
   - 更改本地队列管理器上的队列管理器通道，以使用 **conname** 9.20.193.107(1500) 连接到本地 Aspera gateway 。
   - 更改远程队列管理器上的队列管理器通道，以使用 **conname** 9.20.192.115(1500) 连接到远程 Aspera gateway 。
4. 通过在本地网关机器上运行以下命令来启动本地网关:
   - Linux

```
systemctl start fasp.io-gateway
```

- Windows V 9.2.0 V 9.2.0

```
net start fasp.io-gateway
```

5. 通过在远程网关机器上运行以下命令来启动远程网关:

- Linux

```
systemctl start fasp.io-gateway
```

- Windows V 9.2.0 V 9.2.0

```
net start fasp.io-gateway
```

6. 重新启动通道。

下一步做什么

Aspera gateway 会传递其接收的数据，而不会以任何方式对其进行解释。 这意味着您可以在使用 Aspera gateway 的队列管理器通道之间配置 TLS，因为网关连接无法识别 TLS 握手。 这也意味着任何受支持的 IBM MQ 平台上的队列管理器都可以使用 Aspera gateway。

要将多实例队列管理器与网关配合使用，请为队列管理器的每个实例配置网关定义。

注: Aspera gateway 仅通过队列管理器通道进行了测试。 尚未使用客户机通道进行测试。 这是因为 Aspera gateway 的预期用途是通过慢速网络来连接到远程队列管理器，而客户机应用程序通常是通过快速网络来连接到本地数据中心内的队列管理器。

相关参考

Aspera gateway 路线图

第 14 页的『要使用的通信类型』
不同的平台支持不同的通信协议。 您选择的传输协议取决于 IBM MQ MQI client 和服务器平台的组合。

IBM Aspera fasp.io Gateway 文档

Multi 配置 IBM MQ 以用于 IBM Cloud Private 测量服务

配置 IBM MQ 以与 IBM Cloud Private 测量服务配合使用，以报告和查看队列管理器启动和使用情况信息。

开始之前

在配置 IBM MQ 队列管理器以使用 IBM Cloud Private 服务之前，必须具有 IBM Cloud 帐户。 要创建帐户，请参阅 注册 IBM Cloud。

关于此任务

通过使用 IBM Cloud Private 测量服务，您可以将本地 IBM 产品连接到 IBM Cloud Private 中的服务实例，并在单个仪表板中查看组织中所有已注册的产品。

您可以配置 AIX， Linux 和 Windows 队列管理器并将其连接到测量服务实例，并查看其启动和使用信息。 但是，在 Linux 容器环境以外的平台上，无法使用数据来支持基于每小时容器的定价许可证。

V 9.2.0 要记录每月 VPC 许可证类型的使用数据，而不是缺省的每小时许可度量，请设置环境变量 AMQ_LICENSING_METRIC=VPCMonthlyPeak。 这将导致队列管理器上载与每月 VPC 许可证类型相关的数据，而不是上载与基于每小时容器的许可证相关的数据的缺省行为。

将以下属性与 qm.ini 文件中的 ReportingService 节配合使用:

APIKeyFile
具有测量服务实例 **APIKey** 值的文本文件的位置。

CapacityReporting
按以下格式定期将错误日志消息写入 AMQERR 日志:

```
4/22/2020 01:44:29 PM - Process(1274.1) User(bld-adm) Program(amqmgr0)
Host(8b3b83f2bc7d) Installation(Docker)
VRMF(9.2.0.0)
Time(2020-04-22T13:44:29.295Z)
ArithInsert1(300)
CommentInsert1(8.5)
CommentInsert2(IBM MQ Advanced)
```

CapacityReporting 属性生成的信息将插入到消息 AMQ5064 中，这使您能够更好地了解企业正在使用的 IBM MQ 量:

AMQ5064
此队列管理器已运行 300 秒。 它当前正在与 8.5 个核心一起运行。 许可证类型为 IBM MQ Advanced。

严重
0：参考

说明
这是用于跟踪使用情况的参考消息。

响应
无。

V 9.2.0 **LicensingGroup**
队列管理器所属的计费组。 这会影响在测量服务生成的报告中对数据进行分组的方式。

ServiceURL
IBM Cloud Private 服务地址。

ServiceProxy
在队列管理器无法直接访问正在运行测量服务的网络时可以使用的 HTTP 代理的 URL 和端口。

您可以查看产品安装所在的主机，您正在使用的产品版本以及运行这些产品的平台。 从针对每个产品显示的高级别使用情况度量值中，您可以概述工作负载的繁重程度。 对于 IBM MQ，您可以查看哪些队列管理器的使用率较高，哪些队列管理器的工作负载较轻。

当队列管理器配置为连接到测量服务的实例时，将向 IBM Cloud Private 报告以下信息:

- IBM MQ 队列管理器名称
- IBM MQ 队列管理器标识
- IBM MQ 安装根目录
- IBM MQ 已安装的组件 (名称和版本)
- 主机名
- 主机操作系统名称
- 主机操作系统版本
- IBM MQ 队列管理器的虚拟处理器核心 (VPC) 使用信息

您可以在测量服务实例仪表板中监视队列管理器 VPC 使用情况度量。

过程

- 配置队列管理器以用于 IBM Cloud Private 上的测量服务实例。
- 通过 HTTP 代理连接到 IBM Cloud Private 测量服务。
- 对与 IBM Cloud Private 测量服务的连接进行故障诊断。

相关参考
虚拟处理器核心 (VPC) 的定价指标

Multi V 9.2.0 配置队列管理器以与 IBM Cloud Private 上的测量服务实例配合使用

为队列管理器设置安全性和 IBM Cloud 注册信息，然后连接到已创建的测量服务实例。

关于此任务

IBM Cloud Private 测量服务 实例仪表板仅显示配置为包含安全性和 IBM Cloud Private 注册信息的队列管理器的数据。

过程

1. 遵循 ICP 记录的步骤，在以下位置创建服务标识:
   使用 IBM Cloud Private CLI 创建服务标识。
2. 遵循 ICP 记录的步骤，在以下位置创建 API 密钥:
   API 密钥管理 API。
3. 从 ICP 集群下载 TLS 证书。
   记下将证书下载到的位置。 在步骤 第 702 页的『9』 中，可以将下载的证书添加到队列管理器的密钥库。
4. 创建文本文件 `apikeyfile.txt`，并添加在先前任务中复制的 **API key** 值。
   请注意 `apikeyfile.txt` 的位置，以便您可以在 步骤 8 中包含其路径。 此文件必须可供队列管理器用户 (AIX and Linux 系统上的"*mqm*") 读取。 该文件必须仅包含 **API key** 本身，而不包含 JSON 有效内容，例如 `d9c11b45-4dda-4de4-c0b2-2e4e1004dc64`。
5. 创建队列管理器，例如 *QM1*。
   有关更多信息，请参阅 在 Multiplatforms 版上创建和管理队列管理器。
6. 启动队列管理器 *QM1*。
   有关更多信息，请参阅启动队列管理器。
7. 在运行 IBM MQ 命令之前，请记住设置 IBM MQ 命令行环境。
   运行 **setmqenv** 命令。

   AIX 在 AIX 上:

   ```
   . /usr/mqm/bin/setmqenv -s
   ```

   Linux 在 Linux 上:

   ```
   . /opt/mqm/bin/setmqenv -s
   ```

   Windows 在 Windows 上:

   ```
   "C:\Program Files\IBM\MQ\bin\setmqenv.cmd" -n installation name
   ```

8. 为队列管理器 *QM1* 创建 SSL 信任库。

   AIX 开始在 AIX 上创建信任库:

   ```
   runmqckm -keydb -create -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
   -expire 30 -stash
   ```

   Linux 在 Linux 上:

   ```
   runmqckm -keydb -create -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
   -expire 30 -stash
   ```

Windows 在 Windows 上：

```
runmqckm -keydb -create -db "MQ data directory\qmgrs\QM1\ssl\key.kdb" -pw password -type
cms -expire 30 -stash
```

9. 将您在步骤 第 701 页的『3』中下载的数字证书添加到队列管理器的信任库。

   AIX 在 AIX 上：

```
runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
-label RootCA
-file Download_location/RootCA.crt -format ascii -trust enable

runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
-label ServerCert
-file Download_location/CERT.crt -format ascii -trust enable
```

   Linux 在 Linux 上：

```
runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
-label RootCA
-file Download_location/RootCA.crt -format ascii -trust enable

runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
-label ServerCert
-file Download_location/CERT.crt -format ascii -trust enable
```

   Windows 在 Windows 上：

```
runmqckm -cert -add -db "MQ data directory\qmgrs\QM1\ssl\key.kdb" -pw password -type cms
-label RootCA
-file "Download_location\RootCA.crt" -format ascii -trust enable

runmqckm -cert -add -db "C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl\key.kdb" -pw password -type
cms -label ServerCert
-file "Download_location\CERT.crt" -format ascii -trust enable
```

10. 将具有 `apikeyfile` 路径的新 `ReportingService` 节添加到队列管理器的 `qm.ini` 文件：

```
ReportingService:
   APIKeyFile=APIKey file location/apikeyfile.txt
```

11. 将 **API host** 值添加到 `qm.ini` 文件。

    `ReportingService` 节现在包含 `apikeyfile` 和 **API host** (**ServiceURL**) 值的路径：

```
ReportingService:
   APIKeyFile=APIKey file location/apikeyfile.txt
   ServiceURL=https://productinsights-api.ng.bluemix.net
```

    保存并退出 `qm.ini` 文件。

12. 重新启动队列管理器以使更改生效。

    可能会要求您授予队列管理器进程 **amqzmur0** 访问网络的许可权。 需要此访问权才能使队列管理器与测量服务联系。

13. 在测量服务实例中查看有关队列管理器 *QM1* 的信息。

    当报告状态为 active 时，将向测量服务报告指定集成节点上所有集成服务器的启动和使用情况信息。 每 15 分钟更新一次使用信息。

14. 可选： 通过从队列管理器的 `qm.ini` 文件中除去 `ReportingService` 节并重新启动队列管理器，阻止队列管理器向测量服务报告。

15. 可选： 如果队列管理器未能向测量服务报告启动或使用情况信息，请检查队列管理器日志文件中的诊断信息。

    针对 AIX 进行修订

AIX 在 AIX 上：

```
/var/mqm/qmgrs/QM1/errors/AMQERR0*.log
```

Linux 在 Linux 上：

```
/var/mqm/qmgrs/QM1/errors/AMQERR0*.log
```

Windows 在 Windows 上：

```
C:\ProgramData\IBM\MQ\errors\AMQERR0*.log
```

结果

您创建了测量服务实例，并配置了队列管理器以连接到该实例。 您可以在测量服务实例仪表板中查看有关队列管理器的信息。

Multi V 9.2.0 通过 HTTP 代理连接到 IBM Cloud Private 测量服务

如果队列管理器正在无法直接访问 ICP 集群的系统上运行，那么您可以使用组织提供的 HTTP 代理来连接到 IBM Cloud Private 中的测量服务实例。

开始之前

您已配置安全性，已将 **API key** 和服务 URL 添加到队列管理器的 `qm.ini` 文件。

关于此任务

使用此任务来配置队列管理器，以通过组织提供的 HTTP 代理连接到 IBM Cloud Private 中的 测量服务 实例。

过程

- 将服务代理属性添加到 `qm.ini` 文件的 IBM Cloud Private 注册节。
  您可以按如下所示设置 **ServiceProxy** 属性：
  - 包含 http:// 前缀和 (可选) 端口的 URL。 如果未指定端口，那么将使用 *1080* 。

    ```
    ReportingService:
       ServiceProxy=http://myorgproxy.net:1080
    ```

  注: ServiceProxy 参数必须设置为有效的 http:// URL。 不支持其他代理协议，例如 HTTPS 和 SOCKS。
- 请在更改生效之前重新启动队列管理器。

Multi V 9.2.0 对与测量服务的连接进行故障诊断

有关将队列管理器连接到测量服务实例时可能迁到的错误的故障诊断建议。

队列管理器无法向已配置的测量服务注册或上载使用情况度量值

检查队列管理器是否有权访问网络。 API 密钥文件 的 API 密钥属性中的 **APIKey** 值不正确。 确保已安装 IBM Global Security Kit (GSKit) 组件。

qm.ini 节无效

找到无效的 `qm.ini` 节。 请检查错误日志以获取更多信息。

HTTP 服务代理参数无效

未正确配置队列管理器 ReportingService 节的 **ServiceProxy** 属性值。 队列管理器未向服务注册。 **ServiceProxy** 参数必须设置为有效的 http:// URL。 不支持其他代理协议，例如 HTTPS 和 SOCKS。

Linux 配置 IBM MQ 以用于 Salesforce 推送主题和平台事件

使用此信息通过配置然后运行 IBM MQ Bridge to Salesforce 来设置与 Salesforce 和 IBM MQ 网络的安全性和连接。

开始之前

注: Deprecated 从 2022 年 11 月 22 日起，在所有发行版中不推荐使用 IBM MQ Bridge to Salesforce (请参阅 美国声明函 222-341)。

- IBM MQ Bridge to Salesforce 在 Linux for x86-64 (64 位) 上可用。 不支持将网桥连接到在 IBM WebSphere MQ 6.0 和更低版本上运行的队列管理器。
- V 9.2.0 从 IBM MQ 9.2.0 开始，提供了更多配置选项。 主要更改是，一个队列管理器可以支持多个网桥实例，在这些实例中已正确配置了这些实例。 请参阅第 709 页的『IBM MQ Bridge to Salesforce 的其他配置选项』，以了解更多信息。
- 安装 **MQSeriesSFBridge** 软件包。 有关更多信息，请参阅 在 Linux 上安装 IBM MQ 服务器 和 Linux 系统的 IBM MQ rpm 组件。

关于此任务

Salesforce 是基于云的客户关系管理平台。 如果要使用 Salesforce 来管理客户数据和交互，那么可以使用 IBM MQ Bridge to Salesforce 来预订 Salesforce 推送主题和平台事件，然后可以将这些主题和事件发布到 IBM MQ 队列管理器。 连接到该队列管理器的应用程序可以有用的方式使用推送主题和平台事件数据。 您还可以使用网桥为 Salesforce 中的平台事件创建事件消息。

有关 IBM MQ Bridge to Salesforce 的概述，请参阅 图 1 中的图。

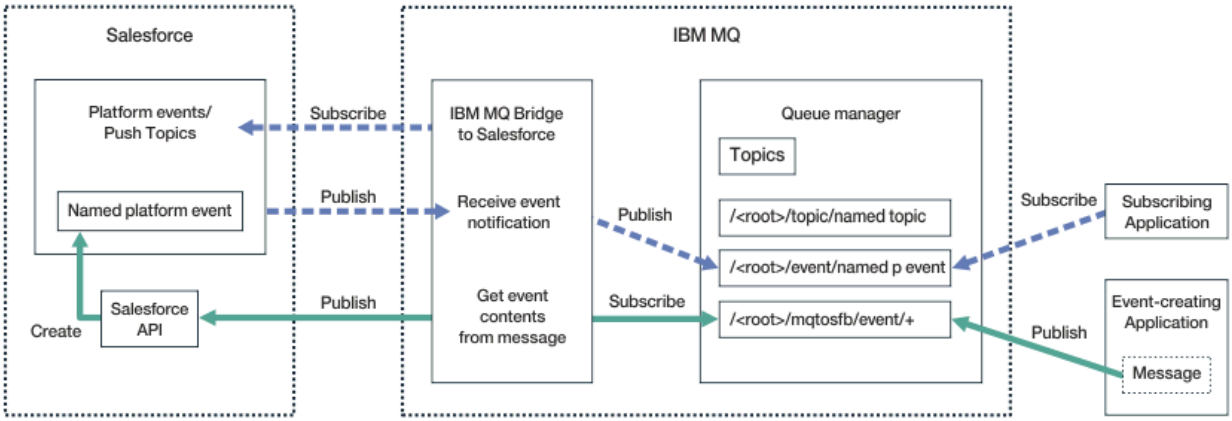


图 *100: IBM MQ Bridge to Salesforce*

推送主题是您定义的查询，可使用 Force.com Streaming API 接收通知以获取对 Salesforce 中记录的更改。 有关配置推送主题以及如何使用 Streaming API 的更多信息，请参阅 介绍流式 API 和 使用 PushTopics。

平台事件是可定制的事件消息，可定义这些事件消息以确定 Force.com 平台生成或使用的事件数据。 有关平台事件以及 Salesforce 事件之间的差异的更多信息，请参阅 企业消息传递平台事件 和 Salesforce 事件之间的差异。

- 要创建用于预订推送主题和平台事件的配置，请参阅第 705 页的『配置 IBM MQ Bridge to Salesforce』。

- 要创建用于为 Salesforce 平台事件创建事件消息的配置，请参阅 第 711 页的『为 Salesforce 平台事件创建事件消息』。

您可以通过 IBM MQ Console 以及将 **-p** 参数与 **amqsrua** 命令配合使用，通过两种方式来监视来自网桥的数据。 针对总体网桥状态发布一组数据:

- 在时间间隔内 (在 STATUS/PUSHTOPIC 树下) 处理的推送主题消息总数。
- 在此时间间隔内看到的推送主题数。
- 在时间间隔内 (在 STATUS/PLATFORM 树下) 处理的平台事件总数。
- 在此时间间隔内看到的平台事件数。
- 在时间间隔内 (在 STATUS/MQPE 树下) 处理的 IBM MQ 已创建平台事件总数。
- 在此时间间隔内看到的 IBM MQ 已创建平台事件的唯一数目。
- 在此时间间隔内看到的 IBM MQ 个已创建平台事件的发布失败次数。

对于每个已配置的 Salesforce 主题，将发布另一条消息。 IBM MQ 主题使用完整的 Salesforce 主题名称以及对象名称中的 /event 或 /topic :

- 在时间间隔内处理的消息数。

要配置 IBM MQ Console 以监视网桥数据，请参阅 配置 IBM MQ Bridge to Salesforce 中的步骤 9 和 10。 有关使用 **amqsrua** 命令的信息，请参阅 监视 IBM MQ Bridge to Salesforce。

遵循以下任务中的步骤来配置和运行 IBM MQ Bridge to Salesforce:

过程

1. 配置 IBM MQ Bridge to Salesforce。
2. 为 Salesforce 平台事件创建事件消息。
3. 运行 IBM MQ Bridge to Salesforce。

相关任务

跟踪 IBM MQ Bridge to Salesforce

相关参考

runmqsfb（运行 IBM MQ 网桥到 Salesforce）

Linux 配置 IBM MQ Bridge to Salesforce

您可以配置 IBM MQ 并输入 IBM MQ Bridge to Salesforce 参数以创建配置文件，并将 Salesforce 推送主题和平台事件连接到 IBM MQ 队列管理器。

开始之前

在开始此任务之前，请确保已在 x86-64 Linux 平台上的 IBM MQ 安装中安装了 MQSeriesSFBridge 软件包。

有关更多信息，请参阅 在 Linux 上安装 IBM MQ 服务器 和 Linux 系统的 IBM MQ rpm 组件。

关于此任务

此任务将指导您完成创建 IBM MQ Bridge to Salesforce 配置文件所需的最低设置，并成功连接到 Salesforce 和 IBM MQ ， 以便您可以预订 Salesforce 推送主题和平台事件。 有关所有参数的含义和选项的更多信息，请参阅 runmqsfb 命令。 您必须考虑自己的安全需求，并定制适合于您的部署的参数。

要创建用于为 Salesforce 平台事件创建事件消息的配置，请参阅 第 711 页的『为 Salesforce 平台事件创建事件消息』。

预订 Salesforce 推送主题和平台事件

当 IBM MQ Bridge to Salesforce 建立与 Salesforce 和 IBM MQ 的连接时，它将创建对 Salesforce 推送主题和平台事件的预订。 在建立连接之前，网桥要预订的推送主题或平台事件名称必须包含在配置文件中或添加到命令行中。

其中一个配置属性是 IBM MQ 主题树的根，将在此根下发布事件。 网桥访问此根并添加完整的 Salesforce 主题名称，例如 /MQ/SF/ROOT/topic/EscalatedCases。 连接到 IBM MQ 的监视主题和应用程序可能会在 /topic/EscalatedCases 下查找推送主题，并在 /event/NewCustomer__e 下查找平台事件。

已发布的消息包含控制信息以及包含所请求数据字段的数据结构。 对于推送主题，数据结构是 **sobject**，对于平台事件，结构是 **payload**。 如果未在 Salesforce 中定义主题或事件，那么网桥无法预订这些主题或事件。 如果网桥在尝试预订主题时迁到错误，那么网桥将停止。

无需在 IBM MQ 中定义主题对象，但必须根据树中最接近的父元素存在适当的权限。 缺省情况下，重新发布的消息仅包含原始消息中的相关数据结构。 将除去控制信息。 对于平台事件，出版物具有有效内容结构。 **网桥程序的行为** 配置参数集中的 **Publish control data with the payload** 配置选项支持重新发布整个消息，包括控制数据。 有关更多信息，请参阅配置参数。

每个推送主题和平台事件在从 Salesforce 发布时都具有关联的 *ReplayID* 。 当与服务器建立连接时，可以使用 *ReplayID* 来请求发布的起始点。 Salesforce 最多维护 24 小时的历史记录，并允许网桥不会错过最近的推送主题和平台事件，即使在生成这些主题和平台事件时未将其启动也是如此。 网桥支持两种服务质量方式:

最多一次
网桥不使用 *ReplayId* 进行重新启动。 重新启动网桥后，仅处理新生成的推送主题和平台事件。 应用程序必须准备好处理缺少的出版物。 *ReplayId* 仍由网桥跟踪并固化为队列，因此可以使用其他服务质量重新启动网桥并了解当前状态。

至少一次
ReplayId 由网桥跟踪并固化到队列。 重新启动网桥时，持久存储的 *ReplayId* 用于从服务器请求发布的起始点。 如果间隔时间不超过 24 小时，将发送较旧的出版物。 主题的 *ReplayId* 不会在每条消息上固化。 它定期写入持久消息中，并在网桥关闭时写入。 必须准备应用程序以查看重复的出版物。

ReplayId 作为消息写入新定义的队列。 在启动网桥之前，必须定义此队列 **SYSTEM.SALESFORCE.SYNCQ**。 如果 **SYSTEM.SALESFORCE.SYNCQ** 不存在，那么无论服务质量方式如何，网桥都不会继续。 提供了 MQSC 脚本，用于创建具有相关属性的队列。 必须使用 DEFSOPT（EXCL）NOSHARE 选项配置队列，以确保只有网桥程序的一个实例可以更新 **SYSTEM.SALESFORCE.SYNCQ** 队列。

要创建用于为平台事件创建事件消息的配置，请参阅 第 711 页的『为 Salesforce 平台事件创建事件消息』。

过程

1. 创建并启动队列管理器。
   a) 创建队列管理器，例如 SQM1。
   ```
   crtmqm SQM1
   ```
   b) 启动队列管理器。
   ```
   strmqm SQM1
   ```
2. **注:** 要使用现有登录和安全性 Salesforce 凭证以及自签名证书，请跳至步骤 第 706 页的『3』。
   可选： 为 Salesforce 帐户创建安全性令牌。
   a) 登录 Salesforce 帐户。
   b) 通过遵循帮助文章 Salesforce help: Reset your security token 中的步骤来创建或重置安全性令牌。
3. 在 Salesforce 中创建 CA 签名的安全证书。
   a) 从 **Force.com 主页** 的 **管理** 菜单中选择 **安全控件**，然后选择 **证书和密钥管理**。
   此时将打开 " **证书和密钥管理** " 页面。
   b) 单击 **创建 CA 签名证书**。
   此时将打开 " **证书** " 页面。
   c) 在 **标签** 字段中输入证书的名称，按 Tab，然后单击 **保存**。
   将显示 "证书和密钥详细信息" 信息。
   d) 单击 **返回到列表: 证书和密钥**。

e) 单击 **导出到密钥库**。

f) 输入密钥库的密码，然后单击 **导出**。

g) 将导出的密钥库保存到本地文件系统。

4. 使用 IBM 密钥管理 GUI 打开从 Salesforce 导出的密钥库并填充签署者证书。

a) 运行 **strmqikm** 命令以打开 IBM 密钥管理 GUI。

有关更多信息，请参阅 使用 runmqckm， runmqakm 和 strmqikm 来管理数字证书。

b) 单击 **打开密钥数据库文件**，然后浏览到 Salesforce 密钥库的位置。

c) 单击 **打开**，确保从 **密钥数据库类型** 选项中选择 **JKS**，然后单击 **确定**。

d) 输入在步骤 3f 中为密钥库创建的密码，然后单击 **确定**。

e) 从 **密钥数据库内容** 选项中选择 **签署者证书**。

f) 单击 **填充**。

g) 从 **添加 CA 证书** 列表中选择 **Verisign Inc.** 复选框，然后单击 **确定**。

5. 可选： 通过在 Salesforce 帐户中为 IBM MQ Bridge to Salesforce 创建应用程序连接来生成 OAuth 使用者密钥和私钥。

在生产环境中使用 IBM MQ Bridge to Salesforce 时，需要 **使用者密钥** 和 **使用者密钥** 代码。

a) 选择 **创建**，然后从 **Force.com 主页** " 的 **构建** 菜单中选择 **应用程序**。

将打开 "应用程序" 页面。

b) 从 " **已连接的应用程序** " 部分单击 **新建**。

这样会打开 " **新建已连接的应用程序** " 页面。

c) 在 **已连接的应用程序名称**中输入 IBM MQ Bridge to Salesforce 的名称，例如 **MQBridgeToSalesforce**。

d) 输入 **API 名称**。

如果跳进到下一个字段，那么会将 **已连接的应用程序名称** 复制到 **API 名称** 名称字段中。

e) 输入 **联系人电子邮件**。

f) 在 **API (启用 OAuth 设置)** 部分中选择 **启用 OAuth 设置** 选项。

然后介绍该节中的其他选项。

g) 添加 **回调 URL**，例如 `https://www.ibm.com`。

h) 从 " **所选 OAuth 作用域** " 子部分的 **可用 OAuth 作用域** 列表中选择 **完全访问权 (完全)** 选项，然后单击 **添加**以添加对 **所选 OAuth 作用域** 列表的完全访问权。

i) 单击**保存**。

j) 单击**继续**。

k) 记下 **使用者密钥** 和 **使用者密钥** 代码。

6. 在队列管理器上创建必需的同步队列。

```
cat /opt/mqm/mqsf/samp/mqsfbSyncQ.mqsc | runmqsc SQM1
```

同步队列在应用程序或队列管理器重新启动之间维护事件状态。 队列深度可以较小，因为队列上只需要一条消息。 一次只能对此队列运行网桥的一个实例，因此设置了用于独占访问的缺省选项。

7. 使用 IBM MQ， Salesforce 和 IBM MQ Bridge to Salesforce 行为的连接和安全参数创建配置文件。

```
runmqsfb -o new_config.cfg
```

现有值显示在方括号内。 按 `Enter` 以接受现有值，按 `Space` then `Enter` 以清除值，然后输入 `Enter` 以添加新值。

a) 输入与队列管理器 SQM1:

连接所需的最小值包括队列管理器名称， IBM MQ 基本主题根和通道名称。

```
Connection to Queue Manager
---------------------------
```

```
Queue Manager or JNDI CF   : []SQM1
MQ Base Topic              : []/sf
MQ Channel                 : []A channel you have defined or for example
SYSTEM.DEF.SVRCONN
MQ Conname                 : []
MQ Publication Error Queue : [SYSTEM.SALESFORCE.ERRORQ]
MQ CCDT URL                : []
JNDI implementation class  : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL          : []
MQ Userid                  : []
MQ Password                : []
```

注: 如果要在本地进行连接，那么不需要通道名称。 您不必在配置文件中提供队列管理器名称和基本主题，因为稍后在运行网桥时可以将它们包括在命令行上。

b) 输入用于连接到 Salesforce 的值:

连接所需的最小值为 Salesforce 用户标识，密码，安全性令牌和登录端点。 在生产环境中，可以添加用于 OAuth 安全性的使用者密钥和私钥。

```
Connection to Salesforce
------------------------
Salesforce Userid (reqd)   : []salesforce_login_email
Salesforce Password (reqd) : []salesforce_login_password
Security Token (reqd)      : []Security_Token
Login Endpoint             : [https://login.salesforce.com]
Consumer ID                : []
Consumer Secret Key        : []
```

c) 输入 TLS 连接的证书库的值:

TLS 连接所需的最小值是 TLS 证书和密钥库密码的密钥库路径。 如果未提供可信存储路径或密码，那么密钥库和密码参数将用于可信存储和密码。 如果要将 TLS 用于 IBM MQ 队列管理器连接，那么可以使用同一密钥库。

```
Certificate stores for TLS connections
--------------------------------------
Personal keystore for TLS certificates : []path_to_keystore, for example: /var/mqm/qmgrs/
SQM1/ssl/key.jks
Keystore password          : []keystore_password
Trusted store for signer certificates : []
Trusted store password     : []
Use TLS for MQ connection  : [N]
```

d) 输入值以配置 IBM MQ Bridge to Salesforce 的行为:

您不必更改或提供任何这些值，但如果您知道推送主题或平台事件名称，请在此处添加这些值。 稍后，当您准备好运行网桥时，也可以在命令行中添加这些参数。 必须在配置文件或命令行中指定日志文件。

```
Behaviour of bridge program
---------------------------
PushTopic Names          : []
Platform Event Names     : []
MQ Monitoring Frequency  : [30]
At-least-once delivery? (Y/N) : [Y]
Subscribe to MQ publications for platform events? (Y/N) : [N]
Publish control data with the payload? (Y/N) : [N]
Delay before starting to process events : [0]
Runtime logfile for copy of stdout/stderr : []
```

8. 可选： 创建 IBM MQ 服务以控制程序的执行。 编辑样本 `mqsfbService.mqsc` 文件以指向新创建的配置文件，并对命令参数进行任何其他更改。

```
cat modified mqsfbService.mqsc | runmqsc SQM1
```

9. 可选： 遵循 IBM MQ 控制台入门 中的指示信息来设置 IBM MQ Console。
10. 可选： 配置 IBM MQ Bridge to Salesforce 以作为无根用户运行。

为了能够以 无根用户身份运行 IBM MQ Bridge to Salesforce，例如在 无根容器中，必须正确设置 Java `userRoot` 和 `systemRoot` 目录，以确保运行网桥进程的用户具有读/写访问权。 要执行此操作，请设置以下 JVM 属性:

```
export MQSFB_EXTRA_JAVA_OPTIONS="-
Djava.util.prefs.userRoot=directory_with_read_write_access"
```

```
export MQSFB_EXTRA_JAVA_OPTIONS="-
Djava.util.prefs.systemRoot=directory_with_read_write_access"
```

结果

您已创建配置文件， IBM MQ Bridge to Salesforce 使用该配置文件来预订 Salesforce 推送主题和平台事件，并将其发布到 IBM MQ 网络。

下一步做什么

完成 第 716 页的『运行 IBM MQ Bridge to Salesforce』的步骤。

相关任务

跟踪 IBM MQ Bridge to Salesforce

监视 IBM MQ Bridge to Salesforce

相关参考

runmqsfb（运行 IBM MQ 网桥到 Salesforce）

Linux V 9.2.0 IBM MQ Bridge to Salesforce 的其他配置选项

从 IBM MQ 9.2.0 开始，提供了其他配置选项，这些选项允许两个主要类别的其他拓扑，即处理 "入站" (从 Salesforce 生成的事件，将发布到 IBM MQ 应用程序) 和 "出站" (IBM MQ 应用程序发布发送到 Salesforce 的事件) 工作。 此外，跟踪和日志记录工作的方式也有变化。

来自 IBM MQ 9.1.0 IBM MQ Bridge to Salesforce 的更改

从 IBM MQ 9.2.0 开始，缺省情况下，除了开始旋转的日志文件之外，不会从 IBM MQ 9.1.0 网桥更改行为。有关更多信息，请参阅 第 710 页的『循环日志』。

主要更改是队列管理器支持多个网桥实例。 要启用此功能以及其他拓扑的其余部分，必须进行一些手动配置更改。

请参阅 runmqfsb 以获取有关其他配置选项的更多信息，并参阅 第 710 页的『示例配置输出』 以获取已修订的配置信息的示例。

分离的入站工作

网桥的多个实例可以处理从 Salesforce 到 IBM MQ 的入站工作，但这些实例必须处理独立的 Salesforce 推送主题和事件集。 否则，可能会出现 IBM MQ 应用程序看到的重复事件，因为没有跨网桥协议来停止复制事件。 每个实例都使用自己的可配置同步队列来保存 **ReplayId**。

在下列情况下，这可能很有用:

- 不同的 Salesforce 主题具有不同的安全权限。 每个网桥实例都有一组不同的凭证用于访问 Salesforce。
- 您担心来自 Salesforce 的工作负载过多，以致单个网桥无法处理。 因此，您可以安排使用 "A-M" 通过一个网桥进行分区，并使用 "N-Z" 通过另一个网桥进行分区。

共享出站工作

网桥支持多个实例，以支持从 IBM MQ 发送到 Salesforce 的出站工作。 如果网桥的一个实例发生故障，那么预订同一队列管理器上相同主题的其他实例可以继续处理发布。

注: 此操作不需要对 IBM MQ 主题配置进行任何更改。

必须设置这些协作实例，以便其中一个实例最多处理来自 Salesforce 的入站工作，因为该实例必须具有对同步队列的独占访问权。

当您担心以下问题时，这可能很有用:

- 来自 IBM MQ 的工作负载。 由于对 Salesforce 的请求是同步的，因此网桥无法处理新工作，而仍在处理一条消息。 拥有多个消费者可以缓解这种情况。
- 可用性体系结构。 例如，现在可以在单独的数据中心内运行多个实例，具有更好的故障转移和灾难恢复选项。 作为 IBM MQ 客户机运行还会将网桥与队列管理器位置分隔开。

跟踪和调试交互

从 IBM MQ 9.2.0 开始，调试标志继续充当 IBM MQ 9.1.0。 即， *-d1* 提供网桥调试信息， *-d2* 为必备组件开启调试日志记录。 但是，如果在启动网桥时启用了 IBM MQ 跟踪，那么将自动开启 *-d2* 级别报告。

循环日志

从 IBM MQ 9.2.0 开始，日志文件的缺省行为是具有三个日志文件，每个文件的大小为 2 MB。 您可以使用其他配置属性来覆盖这些值。 将日志文件的现有配置属性或命令行参数作为日志的基本名称，并添加索引。

如果配置的日志文件具有:

- 无文件类型，索引将添加到文件名的末尾。

  将日志文件设置为 `abc` 会生成名为 `abc.0` 和 `abc.1` 的日志，依此类推。
- 文件类型，在文件类型之前插入索引。

  将日志文件设置为 `abc.log` 会生成名为 `abc.0.log` 和 `abc.1.log` 的日志，依此类推。

注意:

1. 由于网桥可以在具有任意用户许可权的情况下运行，因此无法对日志强制使用特定目录 (例如 `/var/mqm/qmgrs/<qm>/errors`)。
2. 相同的信息将继续写入 *stdout* 和 *stderr* 流。
3. 每当重新打开单个日志文件时，都会重新打印基本配置信息。 这些信息将始终可用，而不是仅在程序开始时打印一次。

保留日志

IBM MQ 9.2.0 拓扑使针对特定队列管理器运行网桥的多个实例的可能性更大。

为了避免实例相互干扰，并避免覆盖先前的网桥运行，如果 `.0` 日志已存在，那么网桥将不会启动。

您需要一个启动过程，在启动网桥之前删除日志的先前副本，或者将类似时间戳记的内容添加到名称中。

相关任务

第 704 页的『配置 IBM MQ 以用于 Salesforce 推送主题和平台事件』
使用此信息通过配置然后运行 IBM MQ Bridge to Salesforce 来设置与 Salesforce 和 IBM MQ 网络的安全性和连接。

跟踪 IBM MQ Bridge to Salesforce

相关参考

鲁姆夫斯布

Linux V 9.2.0 示例配置输出

示例配置输出，显示来自 IBM MQ 9.1.0 IBM MQ Bridge to Salesforce 的更改。

```
IBM MQ Bridge to Salesforce
5724-H72 (C) Copyright IBM Corp. 2017, 2025.
Level : <<unknown>>

Enter new values for the configuration attributes. The
current settings are shown.
```

```
Press ENTER to accept current values; use SPACE+ENTER
to clear values.

Connection to Queue Manager
---------------------------
Queue Manager or JNDI CF   : [V9000_A]
MQ Base Topic              : [/sf]
MQ Channel                 : []
MQ Conname                 : []
MQ Publication Error Queue : [SYSTEM.SALESFORCE.DEADQ]
MQ Replay Status Queue     : [SYSTEM.SALESFORCE.SYNCQ]
MQ CCDT URL                : []
JNDI implementation class  : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL          : []
MQ Userid                  : []
MQ Password                : []

Connection to Salesforce
------------------------
Salesforce Userid (reqd)   : [johndoe@<yourenterprise>.com]
Salesforce Password (reqd) : [********]
Security Token             : [********]
Login Endpoint             : [https://login.salesforce.com]
Consumer Key               : [3MVG9HxRZv05HarQhSy89qSKYNr1gDcv1wE3zN5kyFAa4Wxt]
Consumer Secret            : [********]

Certificate stores for TLS connections
--------------------------------------
Personal keystore for TLS certificates : [/var/mqm/ssl/key.jks]
Keystore password          : [********]
Trusted store for signer certificates : []
Trusted store password     : []
Use TLS for MQ connection  : [N]

Event processing
----------------
PushTopic Names            : []
Platform Event Names       : []
At-least-once delivery for Salesforce events? (Y/N) : [N]
At-least-once delivery for MQ publications? (Y/N) : [N]
Subscribe to MQ publications for platform events? (Y/N) : [Y]
Publish control data with the payload? (Y/N) : [Y]
Treat unknown Salesforce topic as warning (Y/N) : [N]

Behaviour of bridge program
---------------------------
Bridge unique identifier   : []
MQ Monitoring Frequency    : [30]
Delay before starting to process events : [0]
Continue to retry after maximum reconnection attempts (Y/N) : [N]
Runtime logfile for copy of stdout/stderr : [/tmp/runmqsfb.log]
Number of logfiles         : [3]
Maximum size of each logfile : [2097152]
Done.
```

相关参考

鲁姆夫斯布

Linux 为 Salesforce 平台事件创建事件消息

您可以配置 IBM MQ 并输入 `IBM MQ Bridge to Salesforce parameters` 以创建配置文件，并使用网桥为 Salesforce 平台事件创建事件消息。

开始之前

- 您在 x86-64 Linux 平台上的 IBM MQ 安装中安装了 **MQSeriesSFBridge** 软件包。

关于此任务

此任务将指导您完成创建 IBM MQ Bridge to Salesforce 配置文件所需的最低设置，并成功连接到 Salesforce 和 IBM MQ ， 以便您可以为 Salesforce 平台事件创建事件消息。 有关所有参数的含义和选项的更多信息，请参阅 runmqsfb 命令。 您必须考虑自己的安全需求，并定制适合于您的部署的参数。

要创建用于预订推送主题和平台事件的配置，请参阅 第 705 页的『配置 IBM MQ Bridge to Salesforce』。

为 Salesforce 平台事件创建事件消息

您可以使用 IBM MQ 应用程序来创建放在队列管理器主题 /*root*/mqtosfb/event/+上的消息。 网桥预订主题，从消息获取内容，并使用它来发布 Salesforce 平台事件的事件消息。 有关平台事件的更多信息，请参阅 Salesforce 开发者文档中的 使用平台事件交付定制通知 。

要使网桥能够创建事件消息，您必须提供除用于预订推送主题和平台事件的属性以外的两个属性:

- 在 **连接到队列管理器**的网桥配置属性中创建并添加 **MQ Publication Error Queue** 的名称。
- 在用于定义 **网桥程序行为**的网桥配置属性中，将 **Subscribe to MQ publications for platform events** 选项设置为 *Y*。

您需要在 Salesforce 中创建平台事件并定义内容字段，然后才能使用网桥为该平台事件创建事件消息。 平台事件名称及其内容确定您需要如何格式化网桥处理的 IBM MQ 消息。 例如，如果您的 "Salesforce 平台事件 "**Object name** 是 "*MQPlatformEvent1*，您的两个自定义字段是文本字段 "**API name**、"*MyText__c* 和 "姓名__c，那么您在 "/*root*/mqtosfb/event/MQPlatformEvent1__e 主题上发布的 "IBM MQ 消息必须是格式正确的 JSON，如下所示:

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

必须对消息进行格式化，以便 IBM MQ Bridge to Salesforce 可以将其识别为 MQFMT_STRING 格式的消息体。

请参阅步骤 第 714 页的『7』 以在 Salesforce 中创建平台事件，如果您已有要为其创建事件消息的平台事件，请跳过此步骤。 您必须格式化 IBM MQ 消息以匹配在 Salesforce 平台事件中设置的字段。 可以将 Salesforce 平台事件中的字段指定为可选或必填字段。 有关更多信息，请参阅 Salesforce 开发者文档中的 平台事件字段 。

当网桥正在运行时，它预订指定的 IBM MQ 主题。

- 如果在网桥配置中指定 **At-most-once** 服务质量，那么网桥进行的预订是非持久预订。 不会处理 IBM MQ 应用程序在网桥未运行时进行的任何发布。
- 如果在网桥配置中指定 **At-least-once** 服务质量，那么网桥进行的预订是持久预订。 这意味着网桥可以在未运行时处理由 IBM MQ 应用程序生成的发布。 持久预订需要已知预订和客户机标识。 网桥使用 *D_SUB_RUNMQSFB* 作为预订名称，并使用 *runmqsfb_1* 作为客户机标识。

如果网桥用于预订 Salesforce 推送主题和平台事件，而不用于创建事件消息，那么它会尝试删除持久预订，以防配置发生更改，并且该预订现在已孤立。

您可以除去网桥创建的持久预订，如下所示:

使用 IBM MQ Explorer。
打开网桥正在使用的队列管理器的 **预订文件夹** ，并查找以 *:D_SUB_RUNMQSFB* 结尾的预订名称，其中主题字符串为 /sf/mqtosfb/event+。 右键单击预订名称，然后单击 "删除"。 如果收到指示预订正在使用中的错误，那么网桥可能仍在运行。 请停止网桥，然后再次尝试删除预订。

使用 runmqsc 来查找和删除预订。
启动 **runmqsc** 接口并运行 DISPLAY SUB (*)。 查找以 *:D_SUB_RUNMQSFB* 结尾的预订名称 **SUB** 。 发出删除子命令并包含要删除的预订的 **SUBID** 。 例如， DELETE SUB SUBID(414D5120514D312020202020202020205C589459987E8620)

停止，然后启动具有 At-most-once 服务质量的网桥。
如果使用 **At-least-once** 服务质量 At-least-once delivery? (Y/N) :[Y]启动网桥，那么创建的预订是持久预订。 要删除预订，请在配置文件中将服务质量更改为 At-least-once delivery? (Y/N) :[N]，然后重新启动网桥。 将删除持久预订并创建非持久预订。

过程

1. 创建并启动队列管理器。
   a) 创建队列管理器，例如 PEQM1。

   ```
   crtmqm PEQM1
   ```

   b) 启动队列管理器。

```
strmqm PEQM1
```

2. **注:** 要使用现有登录和安全性 Salesforce 凭证以及自签名证书，请跳至步骤 4。

   可选： 为 Salesforce 帐户创建安全性令牌。

   a) 登录 Salesforce 帐户。

   b) 通过遵循帮助文章 Salesforce help: Reset your security token 中的步骤来创建或重置安全性令牌。

3. 在 Salesforce 中创建自签名安全证书。

   a) 从 **Force.com 主页** 的 **管理** 菜单中选择 **安全控件** ，然后选择 **证书和密钥管理**。
   此时将打开 " **证书和密钥管理** " 页面。

   b) 单击 **创建自签名证书**。
   此时将打开 " **证书** " 页面。

   c) 在 **标签** 字段中输入证书的名称，按 Tab，然后单击 **保存**。
   将显示 "证书和密钥详细信息" 信息。

   d) 单击 **返回到列表: 证书和密钥**。

   e) 单击 **导出到密钥库**。

   f) 输入密钥库的密码，然后单击 **导出**。

   g) 将导出的密钥库保存到本地文件系统。

4. 使用 IBM 密钥管理 GUI 打开从 Salesforce 导出的密钥库并填充签署者证书。

   a) 运行 **strmqikm** 命令以打开 IBM 密钥管理 GUI。 有关更多信息，请参阅 使用 runmqckm，runmqakm 和 strmqikm 来管理数字证书。

   b) 单击 **打开密钥数据库文件** ，然后浏览到 Salesforce 密钥库的位置。

   c) 单击 **打开**，确保从 **密钥数据库类型** 选项中选择 **JKS** ，然后单击 **确定**。

   d) 输入在步骤 3f 中为密钥库创建的密码，然后单击 **确定**。

   e) 从 **密钥数据库内容** 选项中选择 **签署者证书** 。

   f) 单击 **填充**。

   g) 从 **添加 CA 证书** 列表中选择 **Verisign Inc.** 复选框，然后单击 **确定**。

5. 可选： 通过在 Salesforce 帐户中为 IBM MQ Bridge to Salesforce 创建应用程序连接来生成 OAuth 使用者密钥和私钥。

   在生产环境中使用 IBM MQ Bridge to Salesforce 时，需要 **使用者密钥** 和 **使用者密钥** 代码。

   a) 选择 **创建**，然后从 **Force.com 主页** " 的 **构建** 菜单中选择 **应用程序** 。
   这样会打开**"应用程序"**页面。

   b) 从 " **已连接的应用程序** " 部分单击 **新建** 。
   这样会打开 " **新建已连接的应用程序** " 页面。

   c) 在**连接的应用程序名称**中输入 "IBM MQ Bridge to Salesforce 的名称，例如 **MQBridgeToSalesforce**。

   d) 输入 **API 名称**。
   如果跳进到下一个字段，那么会将 **已连接的应用程序名称** 复制到 **API 名称** 名称字段中。

   e) 输入 **联系人电子邮件**。

   f) 在 **API (启用 OAuth 设置)** 部分中选择 **启用 OAuth 设置** 选项。
   然后介绍该节中的其他选项。

   g) 添加 **回调 URL**，例如 `https://www.ibm.com`。

   h) 从 " **所选 OAuth 作用域** " 子部分的 **可用 OAuth 作用域** 列表中选择 **完全访问权 (完全)** 选项，然后单击 **添加**以添加对 **所选 OAuth 作用域** 列表的完全访问权。

   i) 单击**保存**。

   j) 单击**继续**。

k) 记下 **使用者密钥** 和 **使用者密钥** 代码。

6. 在队列管理器上创建必需的同步和错误队列。

```
cat /opt/mqm/mqsf/samp/mqsfbSyncQ.mqsc | runmqsc PEQM1
```

同步队列在应用程序或队列管理器重新启动之间维护事件状态。 队列深度可以较小，因为队列上只需要一条消息。 一次只能对此队列运行网桥的一个实例，因此设置了用于独占访问的缺省选项。 必须先创建错误队列，然后才能使用网桥为平台事件创建事件消息。 错误队列用于 Salesforce 无法成功处理的消息。 必须在网桥配置参数部分 **Connection to Queue Manager** 中添加错误队列名称，如步骤 第 714 页的『8.a』 中所示。

7. 可选： 在 Salesforce 帐户中创建平台事件对象。

a) 从 **Force.com 主页** 的 **开发** 菜单中选择 **平台事件** ，然后单击 **新建平台事件**。

此时将打开 " **新建平台事件** " 页面。

b) 填写 **标签** 和 **复数标签** 字段。

c) 单击**保存**。

此时将打开 " **平台事件定义详细信息** " 页面。

d) 定义 **定制字段和关系**。

例如，您可以添加两个带有标签 *MyText* 和 *Name* 的文本字段，并将 **数据类型** 字段长度分别设置为 *Text (64)* 和 *Text (32)* 。

您已创建平台事件并为其定义了 **Custom Fields and Relationships** 。 使用平台事件 平台对象名称 或 *API* 名称 作为 IBM MQ 主题，您可以将希望网桥处理的消息放入该主题。 例如，您可以使用 **AMQSPUBA** 样本将以下 JSON 格式的消息添加到 /sf/mqtosfb/event/*Salesforce Platform Object Name*/*API name* 主题:

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

您可以运行 **AMQSPUBA** 样本以在网桥启动后创建消息。 从 *MQ installation location*/samp/bin 目录中，发出以下命令:

```
./amqspub /sf/mqtosfb/event/Salesforce Platform Object Name/API name PEQM1
```

在提示符处，输入 JSON 格式的消息。

8. 使用 IBM MQ， Salesforce 和 IBM MQ Bridge to Salesforce 行为的连接和安全参数创建配置文件。

```
runmqsfb -o new_config.cfg
```

现有值显示在方括号内。 按 Enter 以接受现有值，按 Space then Enter 以清除值，然后输入 Enter 以添加新值。

a) 输入与队列管理器 PEQM1:

连接所需的最小值为队列管理器名称， IBM MQ 基本主题根，错误队列名称和通道名称。

```
Connection to Queue Manager
---------------------------
Queue Manager or JNDI CF   : []PEQM1
MQ Base Topic              : []/sf
MQ Channel                 : []A channel you have defined or for example
SYSTEM.DEF.SVRCONN
MQ Conname                 : []
MQ Publication Error Queue : [SYSTEM.SALESFORCE.ERRORQ]
MQ CCDT URL                : []
JNDI implementation class  : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL          : []
MQ Userid                  : []
MQ Password                : []
```

注: 如果要在本地进行连接，那么不需要通道名称。 您不必在配置文件中提供队列管理器名称和基本主题，因为稍后在运行网桥时可以将它们包括在命令行上。

b) 输入用于连接到 Salesforce 的值:

连接所需的最小值为 Salesforce 用户标识，密码，安全性令牌和登录端点。 在生产环境中，可以添加用于 OAuth 安全性的使用者密钥和私钥。

```
Connection to Salesforce
------------------------
Salesforce Userid (reqd)   : []salesforce_login_email
Salesforce Password (reqd) : []salesforce_login_password
Security Token (reqd)      : []Security_Token
Login Endpoint             : [https://login.salesforce.com]
Consumer ID                : []
Consumer Secret Key        : []
```

c) 输入 TLS 连接的证书库的值:

TLS 连接所需的最小值是 TLS 证书和密钥库密码的密钥库路径。 如果未提供可信存储路径或密码，那么密钥库和密码参数将用于可信存储和密码。 如果要将 TLS 用于 IBM MQ 队列管理器连接，那么可以使用同一密钥库。

```
Certificate stores for TLS connections
--------------------------------------
Personal keystore for TLS certificates : []path_to_keystore, for example: /var/mqm/qmgrs/
PEQM1/ssl/key.jks
Keystore password          : []keystore_password
Trusted store for signer certificates : []
Trusted store password     : []
Use TLS for MQ connection  : [N]
```

d) 输入值以配置 IBM MQ Bridge to Salesforce 的行为:

必须将 **Subscribe to MQ publications for platform events** 选项从缺省 *N* 更改为 *Y*，以使用网桥来创建事件消息。 您还必须在配置文件或命令行中指定日志文件。

```
Behaviour of bridge program
---------------------------
PushTopic Names            : []
Platform Event Names       : []
MQ Monitoring Frequency    : [30]
At-least-once delivery? (Y/N) : [Y]
Subscribe to MQ publications for platform events? (Y/N) : [Y]
Publish control data with the payload? (Y/N) : [N]
Delay before starting to process events : [0]
Runtime logfile for copy of stdout/stderr : []
```

9. 可选： 创建 IBM MQ 服务以控制程序的执行。 编辑样本 `mqsfbService.mqsc` 文件以指向新创建的配置文件，并对命令参数进行任何其他更改。

```
cat modified mqsfbService.mqsc | runmqsc PEQM1
```

10. 可选： 遵循 IBM MQ 控制台入门 中的指示信息来设置 IBM MQ Console。
11. 可选： 在 IBM MQ Console 实例中添加和配置窗口小部件以查看 Salesforce 数据。
    a) 单击 **添加窗口小部件**。
    这样会打开新窗口小部件。
    b) 选择 **图表**
    c) 单击新窗口小部件的标题栏中的 **配置窗口小部件** 图标。
    d) 可选： 输入**窗口小部件标题**。
    e) 从 **要监视的资源 源** 下拉菜单中选择 **Salesforce 网桥** 。
    f) 从 **资源类**下拉菜单中选择 **网桥状态**。
    g) 从 **资源类型**下拉菜单中选择 **MQ 创建的平台事件**。
    h) 从 **资源元素**下拉菜单中选择 **总计 MQ 创建的平台事件**。
    i) 单击**保存**。

您已配置 IBM MQ Console 以显示已创建的 IBM MQ 平台事件总数。 当网桥正在运行并且您开始将消息放在 `/sf/mqtosfb/event/`*`Salesforce Platform Object Name`*`/`*`API name`* 主题上时，该窗口小部件将显示网桥创建的消息事件总数。

IBM MQ Bridge to Salesforce 的消息格式和错误消息

有关 IBM MQ Bridge to Salesforce 所处理消息的格式的信息。

应用程序将消息放入特定队列管理器主题，例如 /*root*/mqtosfb/event/MQPlatformEvent1__e。 网桥预订主题，从消息获取内容，并使用它来发布 Salesforce 平台事件的事件消息。

您需要在 Salesforce 中创建平台事件并定义内容字段，然后才能使用网桥为该平台事件创建事件消息。 平台事件名称及其内容确定您需要如何格式化网桥处理的 IBM MQ 消息。 例如，如果您的 "Salesforce 平台事件 "**Object name** 是 "*MQPlatformEvent1*，您的两个自定义字段是文本字段 "**API name**、"*MyText__c* 和 "姓名__*c*，那么您在 "/*root*/mqtosfb/event/MQPlatformEvent1__e 主题上发布的 "IBM MQ 消息必须是格式正确的 JSON，如下所示：

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

网桥使用和生成的消息是 JSON 格式的文本 (MQSTR) 消息。 输入消息是一个简单的 JSON ，程序可以使用字符串并置来生成该消息。

错误消息

网桥可以检测到错误，例如，如果消息不是文本格式，或者 Salesforce 可以检测到错误，例如，如果平台事件名称不存在。 如果在处理输入消息时发生错误，那么会将消息与描述该错误的属性一起移至网桥错误队列。 该错误还会写入网桥的 *stderr* 流。

Salesforce 生成的错误是 JSON。 以下是由格式不正确的消息导致的一些错误：

平台事件内容错误，状态为 400 文本

```
[{"message":"No such column 'Name__c' on sobject of type
MQPlatformEvent2__e","errorCode":"INVALID_FIELD"}
```

平台事件名称无效，状态 404 文本

```
{"errorCode":"NOT_FOUND","message":"The requested resource does not exist"}
```

JSON 错误，状态为 400 文本

```
{"errorCode":"NOT_FOUND","message":"The requested resource does not exist"}
```

消息不是 JSON ，状态为 400 文本

```
[{"message":
  "Unexpected character ('h' (code 104)): expected a valid value (number, String, array,
object, 'true', 'false' or 'null') at [line:1, column:2]",
  "errorCode":"JSON_PARSER_ERROR"}
```

不是文本消息 (未发送到 Salesforce)

```
Error: Publication on topic ' /sf/mqtosfb/event/MQPlatformEvent1' does not contain a text
formatted message
```

Linux 运行 IBM MQ Bridge to Salesforce

运行 IBM MQ Bridge to Salesforce 以连接到 Salesforce 和 IBM MQ。 连接后，网桥可以创建对 Salesforce 主题的预订，并将消息重新发布到 IBM MQ 主题。 网桥还可以为 Salesforce 平台事件创建事件消息。

开始之前

您已完成 任务中的配置步骤：

- 第 705 页的『配置 IBM MQ Bridge to Salesforce』
- 第 711 页的『为 Salesforce 平台事件创建事件消息』

.

关于此任务

使用在前一任务中创建的配置文件来运行 IBM MQ Bridge to Salesforce。 如果配置文件中未包含所有必需参数，请确保在命令行中包含这些参数。

过程

1. 在 Salesforce 中定义要预订 的推送主题或平台事件，或者定义要为其创建事件消息的平台事件。
2. 启动 IBM MQ Bridge to Salesforce 以连接到 Salesforce 和队列管理器。 如果要运行网桥以预订 Salesforce 事件，请包含在步骤 1 中定义的推送主题或平台事件的名称。

   ```
   runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName
   ```

   连接网桥时，将返回以下消息：

   - 如果要使用网桥来预订 Salesforce 推送主题和平台事件：

     ```
     Successful connection to queue manager QM1
     Warning: Subscribing to MQ-created platform events is not enabled.
     Successful login to Salesforce at https://eu11.salesforce.com
     Ready to process events.
     ```

   - 如果要使用网桥为 Salesforce 平台事件创建事件消息：

     ```
     Successful connection to queue manager QM1
     Successful login to Salesforce at https://eu11.salesforce.com
     Successful subscription to '/sf/mqtosfb/event/+' for MQ-created platform events
     Ready to process events.
     ```

3. 可选： 如果在运行网桥后返回的消息指示未成功连接，请对与队列管理器和 Salesforce 的连接进行故障诊断。

   a) 在调试方式下使用调试选项 1 发出该命令。

   ```
   runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName -d 1
   ```

   网桥将逐步完成连接设置并以简洁方式显示处理消息。

   b) 在调试方式下使用调试选项 2 发出该命令。

   ```
   runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName -d 2
   ```

   网桥将逐步完成连接设置并以详细方式显示处理消息。 完整输出将写入日志文件。

4. 通过使用 Salesforce 界面修改数据库中的记录来生成事件。
5. 转至 IBM MQ Console，以查看对前一任务中配置的窗口小部件中显示的推送主题所作更改。

下一步做什么

使用 *MQSFB_EXTRA_JAVA_OPTIONS* 变量传入 JVM 属性，例如，用于启用 IBM MQ 跟踪。 有关更多信息，请参阅 跟踪 IBM MQ Bridge to Salesforce。

相关任务

监视 IBM MQ Bridge to Salesforce

相关参考

runmqsfb（运行 IBM MQ 网桥到 Salesforce）

MQ Adv. z/OS Linux Deprecated 配置 IBM MQ 以用于区块链

设置并运行 IBM MQ Bridge to blockchain 以安全地连接 Linux IBM MQ Advanced 或 z/OS IBM MQ Advanced for z/OS Value Unit Edition 队列管理器和 IBM Blockchain。 通过使用连接到 IBM MQ Advanced 或 IBM MQ Advanced for z/OS VUE 队列管理器的消息传递应用程序，使用网桥以异步方式连接到，查找和更新区块链中资源的状态。

开始之前

注意:

- Deprecated 从 2022 年 11 月 22 日起，在所有发行版中不推荐使用 IBM MQ Bridge to blockchain (请参阅 美国声明函 222-341)。
- V 9.2.0.21 Removed 对于 Long Term Support，将在 IBM MQ 9.2.0 CSU 21 中移除 IBM MQ Bridge to blockchain 。 如果您有将受此更改影响的应用程序，请联系 IBM 支持人员。

  注意: V 9.2.0 不再支持基于 Hyperledger Composer 构建的 IBM MQ Bridge to blockchain 。 您必须正在运行 IBM MQ 9.1.4 或更高版本才能使用基于 Hyperledger Fabric 构建的 IBM MQ Bridge to blockchain 。
- IBM MQ Bridge to blockchain 可用于连接到:
  - Linux IBM MQ Advanced，或
  - z/OS IBM MQ Advanced for z/OS VUE

  仅队列管理器。
- V 9.2.0 队列管理器必须与网桥处于同一命令级别或更高级别; 例如 IBM MQ 9.2.0。
- V 9.2.0 支持将 IBM MQ Bridge to blockchain 用于基于 Hyperledger Fabric 1.4 体系结构的区块链网络。

关于此任务

Blockchain 是一个共享的分布式数字分类帐，由表示网络中同级之间的商定交易的区块链组成。 链中的每个块都链接到前一个块，以此类推，返回到第一个事务。

IBM Blockchain 是基于 Hyperledger Fabric 构建的，您可以在本地使用 Docker 进行开发，也可以在 IBM Cloud 中的容器集群中进行开发。 您还可以在生产环境中激活和使用 IBM Blockchain 网络，以构建和管理具有高级别安全性，隐私性和性能的业务网络。 有关更多信息，请参阅 IBM Blockchain Platform。

Hyperledger Fabric 是由 Hyperledger Project 成员协作开发的开放式源代码企业区块链框架，包括作为初始代码添加者的 IBM 。 Hyperledger Project 或 Hyperledger，是一个 Linux Foundation 开源的全球协作计划，旨在推进跨行业区块链技术。 有关更多信息，请参阅 IBM Blockchain， Hyperledger 项目和 Hyperledger Fabric。

如果您已在使用 IBM MQ Advanced 或 IBM MQ Advanced for z/OS VUE 以及 IBM Blockchain，那么可以使用 IBM MQ Bridge to blockchain 从区块链网络发送简单查询，更新和接收回复。 通过这种方式，您可以将本地 IBM 软件与云区块链服务集成。

您可以在 图 1 中查看网桥操作过程的简要概述。 用户应用程序将 JSON 格式的消息放在 IBM MQ Advanced 或 IBM MQ Advanced for z/OS VUE 队列管理器上的输入/请求队列上。 网桥连接到队列管理器，从输入/请求队列获取消息，检查 JSON 是否正确格式化，然后发出查询或更新到区块链。 区块链返回的数据由网桥解析并放置在应答队列上，如原始 IBM MQ 请求消息中所定义。 用户应用程序可以连接到队列管理器，从应答队列获取响应消息，并使用该信息。

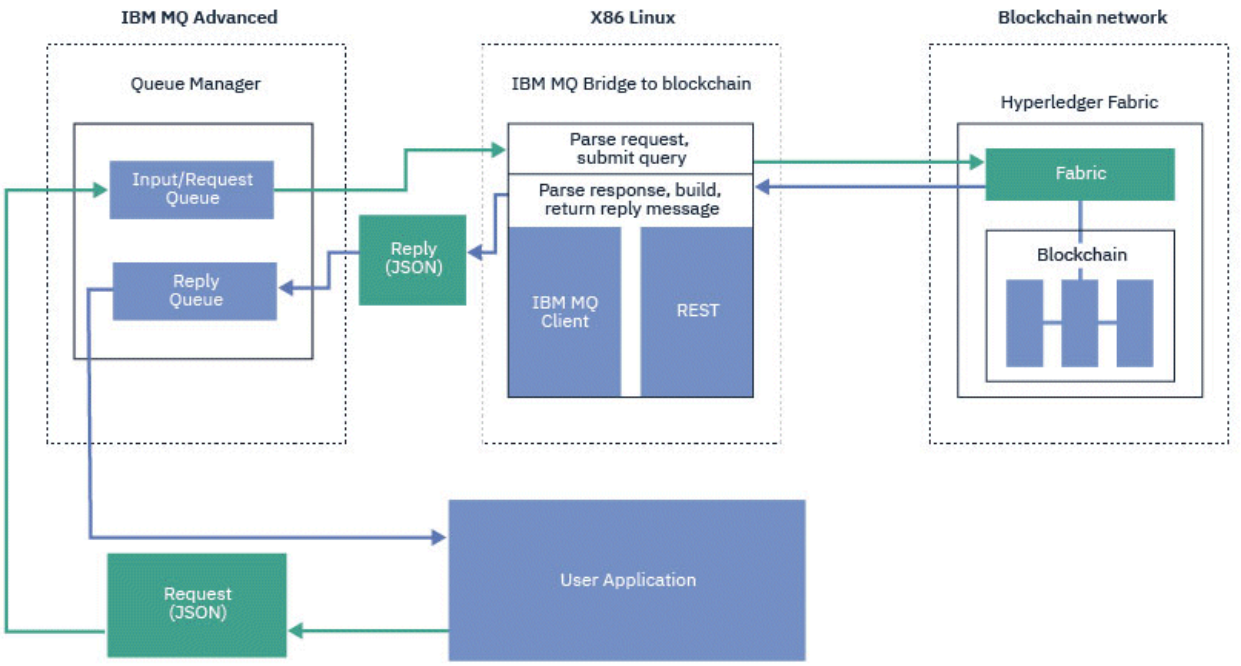


图 *101: IBM MQ Bridge to blockchain*

您可以配置 IBM MQ Bridge to blockchain 以作为参与者或同级连接到区块链网络。 当网桥运行时，消息传递应用程序请求网桥驱动链代码例程，这些例程查询或更新资源的状态，并将结果作为响应返回给消息传递应用程序。

过程

1. 创建并启动队列管理器，或者启动要与 IBM MQ Bridge to blockchain 配合使用的现有队列管理器。
   创建队列管理器:

   ```
   crtmqm adv_qmgr_name
   ```

   启动队列管理器:

   ```
   strmqm adv_qmgr_name
   ```

2. 为 **DefineQ.mqsc** 脚本中定义的网桥创建队列。
   为用于下列各项的缺省指定队列提供了样本网桥队列定义:
   - 用户凭证，例如 SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE
   - 网桥的消息输入，例如 APPL1.BLOCKCHAIN.INPUT.QUEUE
   - 来自区块链的回复，例如 APPL1.BLOCKCHAIN.REPLY.QUEUE

   从 /opt/mqm/mqbc/samp 目录中，发出以下命令:

   ```
   runmqsc adv_qmgr_name < ./DefineQ.mqsc
   ```

   不同的应用程序可以使用相同的输入队列，但您可以指定多个应答队列，每个应用程序一个应答队列。 您不必使用定义的应答队列。 如果要将动态队列用于应答，那么必须考虑其安全性配置。

结果

您已创建网桥处理来自 IBM MQ 和区块链网络的消息所需的队列。

下一步做什么

使用 IBM MQ Advanced 或 IBM MQ Advanced for z/OS VUE 队列管理器信息以及来自区块链网络的凭证为 IBM MQ Bridge to blockchain 创建配置文件。

Deprecated 为 IBM MQ Bridge to blockchain 创建配置文件

输入队列管理器和区块链网络参数，为 IBM MQ Bridge to blockchain 创建配置文件以连接到 IBM MQ 和 IBM Blockchain 网络。

开始之前

- 您已创建并配置了区块链网络。
- 您具有来自区块链网络的凭证文件。
- 您已在 x86 Linux 环境上安装 IBM MQ Bridge to blockchain。

  有关更多信息，请参阅 在 Linux 上安装 IBM MQ 服务器 和 Linux 系统的 IBM MQ rpm 组件。
- 您已启动 IBM MQ Advanced 队列管理器。

关于此任务

此任务将引导您完成创建 IBM MQ Bridge to blockchain 配置文件并成功连接到 IBM Blockchain 和 IBM MQ 网络所需的最低设置。

您可以使用网桥连接到基于 Hyperledger Fabric 1.4 architecture 的区块链网络。 要使用网桥，您需要来自区块链网络的配置信息。 在此任务中的每个步骤中，您可以找到基于两个不同配置的区块链网络的示例配置详细信息:

- 在 Docker 中运行的 Hyperledger Fabric 网络。 有关更多信息，请参阅 Hyperledger Fabric 入门， 编写第一个应用程序和 第 721 页的『示例 Hyperledger Fabric 网络凭证文件』。
- 在 IBM Cloud 中的 Kubernetes 集群中运行的 Hyperledger Fabric 网络。 有关详细信息，请参阅在 IBM Blockchain Platform 的云沙箱中进行开发。

有关所有 IBM MQ Bridge to blockchain 参数的含义和选项的更多信息，请参阅 runmqbcb 命令。 您必须考虑自己的安全需求，并定制适合于您的部署的参数。

过程

1. 运行网桥以创建配置文件。

   您需要来自区块链网络凭证文件和 IBM MQ Advanced 队列管理器的参数。

   ```
   runmqbcb -o config_file_name.cfg
   ```

   如以下示例所示，现有值显示在方括号内。 按 Enter 以接受现有值，按 Space 然后 Enter 以清除值，然后在方括号内输入，然后按 Enter 以添加新值。 您可以使用逗号或在新行上输入每个值来分隔值列表(例如同级)。 空白行结束列表。

   注: 无法编辑现有值。 您可以保留，替换或清除这些信息。
2. 输入与 IBM MQ Advanced 队列管理器的连接的值。

   连接所需的最小值包括队列管理器名称，网桥输入的名称以及您定义的身份队列。 对于与远程队列管理器的连接，您还需要 **MQ Channel** 和 **MQ Conname** (运行队列管理器的主机地址和端口)。 要使用 TLS 在步骤 第 721 页的『4』中连接到 IBM MQ，必须使用 JNDI 或 CCDT 并相应地指定 **MQ CCDT URL** 或 **JNDI implementation class** 和 **JNDI provider URL**。

   ```
   Connection to Queue Manager
   ---------------------------
   Queue Manager                      : [adv_qmgr_name]
   Bridge Input Queue                 : [APPL1.BLOCKCHAIN.INPUT.QUEUE]
   Bridge User Identity Queue         : [SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE]
   MQ Channel                         : []
   ```

```
MQ Conname                            : []
MQ CCDT URL                           : []
JNDI implementation class             :
  [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL                     : []
MQ Userid                             : []
MQ Password                           : []
```

3. 13. 输入网络的 Hyperledger Fabric 服务器凭证。
   以下代码中显示了您应该期望的内容的示例:

```
Fabric Server
--------------
Network configuration file            : []connection-tls.json
Wallet                                : []
User Name                             : []User1
Certificate                           : []<path_to_user_certificate>
Private Key                           : []<path_to_private_key>/private_key.pem
Organisation                          : []Org1MSP
```

4. 输入 TLS 连接的证书库值。
   如果没有此区域，请将其留空。

```
Certificate stores for MQ TLS connections
-----------------------------------------
Personal keystore                     : []
Keystore password                     : []
Trusted store for signer certs        : []
Trusted store password                : []
```

5. 输入要在其中写入网桥日志的日志文件的路径。

```
Behaviour of bridge program
---------------------------
Runtime logfile for copy of stdout/stderr : []bridgelog.log
Number of logfiles                    : [3]
Maximum size of each logfile (bytes) : [2097152]
```

注意: 先前，有关同级，排序节点和认证中心的详细信息存储在此网桥设置中。 但是，此信息现在存储在 网络配置文件中，该文件链接到设置的 Hyperledger Fabric 服务器部分中。

结果

您已创建 IBM MQ Bridge to blockchain 用于连接到 IBM Blockchain 网络和 IBM MQ Advanced 队列管理器的配置文件。

下一步做什么

完成 第 723 页的『运行 IBM MQ Bridge to blockchain』的步骤。

Deprecated V 9.2.0 示例 Hyperledger Fabric 网络凭证文件

在 Docker 中运行的本地实例化 Hyperledger Fabric 区块链网络中 `.yml` 文件的内容，可用于配置 IBM MQ Bridge to blockchain。

IBM MQ Bridge to blockchain 可用于连接到:

- Linux IBM MQ Advanced，或
- z/OS IBM MQ Advanced for z/OS VUE

仅队列管理器。

在完成 Hyperledger Fabric 教程入门，了解 幕后发生的事情以及使用其中一个 Hyperledger Fabric 样本启动网络之后，您应该在 `/blockchain/fabric-samples/basic-network` 文件夹中具有以下配置文件。

如果要连接到区块链网络，那么当您是 第 720 页的『为 IBM MQ Bridge to blockchain 创建配置文件』时，必须使用此文件中的配置详细信息。

```
{
  "name": "basic-network",
  "version": "1.0.0",
  "client": {
    "organization": "Org1",
    "connection": {
      "timeout": {
        "peer": {
          "endorser": "300"
        },
        "orderer": "300"
      }
    }
  },
  "channels": {
    "mychannel": {
      "orderers": [
        "orderer.example.com"
      ],
      "peers": {
        "peer0.org1.example.com": {
          "endorsingPeer": true,
          "chaincodeQuery": true,
          "ledgerQuery": true,
          "eventSource": true
        },
        "peer0.org2.example.com": {
          "endorsingPeer": true,
          "chaincodeQuery": false,
          "ledgerQuery": true,
          "eventSource": false
        }
      }
    }
  },
  "organizations": {
    "Org1": {
      "mspid": "Org1MSP",
      "peers": [
        "peer0.org1.example.com"
      ],
      "certificateAuthorities": [
        "ca-org1"
      ],
      "adminPrivateKeyPEM": {
        "path": "$<path_to_private_key>/admin_private_key"
      },
      "signedCertPEM": {
        "path": "<path_to_org_signed_cert>/Admin@org1.example.com-cert.pem"
      }
    },
    "Org2": {
      "mspid": "Org2MSP",
      "peers": [
        "peer0.org2.example.com"
      ],
      "certificateAuthorities": [
        "ca-org2"
      ]
    }
  },
  "orderers": {
    "orderer.example.com": {
      "url": "grpcs://localhost:7050",
      "mspid": "OrdererMSP",
      "grpcOptions": {
        "ssl-target-name-override": "orderer.example.com",
        "hostnameOverride": "orderer.example.com"
      },
      "tlsCACerts": {
        "path": “<path_to_orderer_cert>/ca.crt"
      },
      "adminPrivateKeyPEM": {
        "path": <path_to_orderers_private_key>/<private_key>"
      },
      "signedCertPEM": {
        "path": "<path_to_orderer_signed_cert>/Admin@example.com-cert.pem"
```

```
      }
    }
  },
  "peers": {
    "peer0.org1.example.com": {
      "url": "grpcs://localhost:7051",
      "grpcOptions": {
        "ssl-target-name-override": "peer0.org1.example.com",
        "hostnameOverride": "peer0.org1.example.com",
        "request-timeout": 120001
      },
      "tlsCACerts": {
        "path": <path_to_peer_cert>/ca.crt"
      }
    },
    "peer0.org2.example.com": {
      "url": "grpcs://localhost:9051",
      "grpcOptions": {
        "ssl-target-name-override": "peer0.org2.example.com",
        "hostnameOverride": "peer0.org2.example.com",
        "request-timeout": 120001
      },
      "tlsCACerts": {
        "path": "<path_to_peer_cert>/ca.crt"
      }
    }
  },
  "certificateAuthorities": {
    "ca-org1": {
      "url": "https://localhost:7054",
      "grpcOptions": {
        "verify": true
      },
      "tlsCACerts": {
        "path": "<path_to_ca_cert>/ca.org1.example.com-cert.pem"
      },
      "registrar": [
        {
          "enrollId": "admin",
          "enrollSecret": "adminpw"
        }
      ]
    },
    "ca-org2": {
      "url": "https://localhost:8054",
      "grpcOptions": {
        "verify": true
      },
      "tlsCACerts": {
        "path": "<path_to_ca_cert>/ca.org2.example.com-cert.pem"
      },
      "registrar": [
        {
          "enrollId": "admin",
          "enrollSecret": "adminpw"
        }
      ]
    }
  }
}
```

Deprecated 运行 IBM MQ Bridge to blockchain

运行 IBM MQ Bridge to blockchain 以连接到 IBM Blockchain 和 IBM MQ。 连接后，网桥已准备好处理请求消息，将其发送到 Hyperledger Composer 区块链网络，并接收和处理回复。

关于此任务

使用在前一任务中创建的配置文件来运行 IBM MQ Bridge to blockchain。

过程

1. 启动要用于网桥的 IBM MQ Advanced 队列管理器。

2. 启动 IBM MQ Bridge to blockchain 以连接到 Hyperledger Composer REST 服务器和 IBM MQ Advanced 队列管理器。
   运行网桥命令。

   ```
   runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
   ```

   连接网桥时，将返回类似于以下内容的输出:

   ```
   2018-05-17 14:28:16.866 BST IBM MQ Bridge to Blockchain
   5724-H72 (C) Copyright IBM Corp. 2017, 2025.

   2018-05-17 14:28:19.331 BST Ready to process input messages.
   ```

3. 可选： 如果在运行网桥后返回的消息指示连接不成功，那么对 IBM MQ Advanced 队列管理器和区块链网络的连接进行故障诊断。
   a) 在调试方式下使用调试选项 1 发出该命令。

   ```
   runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
   -d 1
   ```

   网桥将逐步完成连接设置并以简洁方式显示处理消息。
   b) 在调试方式下使用调试选项 2 发出该命令。

   ```
   runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
   -d 2
   ```

   网桥将逐步完成连接设置并以详细方式显示处理消息。 完整输出将写入日志文件。

结果

您已启动 IBM MQ Bridge to blockchain 并使用 Hyperledger Composer REST 服务器连接到队列管理器和区块链网络。

下一步做什么

- 请遵循 第 727 页的『运行 IBM MQ Bridge to blockchain 客户机样本』 中的步骤进行格式化，并向区块链网络发送查询或更新消息。
- 使用 *MQBCB_EXTRA_JAVA_OPTIONS* 变量传入 JVM 属性，例如启用 IBM MQ 跟踪。 有关更多信息，请参阅 跟踪 IBM MQ Bridge to blockchain。

z/OS Deprecated IBM MQ 9.2.0 之前的 IBM MQ Bridge to blockchain 的消息格式

有关 IBM MQ Bridge to blockchain 发送和接收的消息格式的信息。

V 9.2.0

注意: 消息格式的现有格式已过时。 从 IBM MQ 9.2.0 开始，如果您有 Hyperledger Fabric 网络，请使用 第 726 页的『来自 IBM MQ 9.2.0 的 IBM MQ Bridge to blockchain 的消息格式』中描述的消息格式。

应用程序请求 IBM MQ Bridge to blockchain 驱动 Hyperledger Composer 定义的 REST API 对区块链上保存的信息执行操作。 应用程序通过在网桥请求队列上放置请求消息来执行此操作。 REST 请求的结果由网桥格式化为应答消息。 网桥使用请求消息的 MQMD 中的 **ReplyToQ** 和 **ReplyToQMgr** 字段中包含的信息作为应答消息的目标。

请求和应答消息是 JSON 格式的文本 (MQSTR) 消息。

请求消息格式

请求消息包含三个属性:

方法 (method)
用于调用 Hyperledger Composer REST API (例如 POST， DELETE 或 GET) 的 REST 动词

path
Hyperledger Composer REST API 的路径。 这将添加到基本服务器 URL。 路径应以 "api/" 开头。

主体
特定于方法的内容。 这通常是 JSON 结构。

以下示例使用 POST 方法来路径 api/Trader，以创建新的 Trader 对象。 主体指定由用户的 Hyperledger Composer 模型定义的贸易商类，并且还指定在区块链网络中创建新的 Trader 对象所需的其他值。

```
{ "method": "POST",
  "path": "api/Trader",
  "body": {
    "$class" : "org.example.trading",
    "tradeId" : "Trader2",
    "firstName": "Jane",
    "lastName" : "Doe"
```

应答消息格式

应答消息将其相关标识设置为入站消息的消息标识。 任何用户定义的属性都将从请求消息复制到应答消息。 应答中的用户标识设置为发起方的用户标识。

statusCode 是 HTTP 状态码。 如果错误来自 IBM MQ 或网桥，那么将使用相应的 **statusCode** 。

statusType 是字符串 成功 或 失败。

对于成功的请求，应答消息中的 "**data**" 元素包含来自调用的 Hyperledger Composer REST API 的响应。

成功处理的示例:

```
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "data": [
      {
         "$class": "org.example.trading",
         "firstName": "John",
         "lastName": "Doe",
         "tradeId": "Trader1"
      },
      {
         "$class": "org.example.trading",
         "firstName": "Jane",
         "lastName": "Doe",
         "tradeId": "Trader2"
      }
   ]
}
```

所有错误响应都具有相同的字段，无论这些字段是由网桥本身生成，从对 Hyperledger Composer REST 服务器的调用，区块链还是从链代码调用生成。 例如:

- JSON 输入消息错误

```
{
   "statusCode": 400,
   "statusType": "FAILURE",
   "message": "[AMQBC021E] Error: Cannot parse input message or there are
    missing fields in the message. Missing fields appear to be: "method".”
}
```

- 未能由 Hyperledger Composer REST 服务器处理的请求

```
{
   "statusCode": 500,
   "statusType": "FAILURE",
   "message": "Error trying to invoke business network. Error: No valid responses
    from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
    error (status: 500, message: Error: Failed to add object with ID 'Trader1'
```

```
    as the object already exists)"
}
```

应用程序可以通过查看 **statusType** 字符串或数据字段的存在来判断请求是成功还是失败。 当处理输入消息时发生错误，并且网桥未将其发送到区块链时，从网桥返回的值为 MQRC 值，通常为 **MQRC_FORMAT_ERROR**。

Linux Deprecated V 9.2.0 来自 IBM MQ 9.2.0 的 IBM MQ Bridge to blockchain 的消息格式

有关 IBM MQ Bridge to blockchain 发送和接收的消息格式的信息。

应用程序请求 IBM MQ Bridge to blockchain 驱动 Hyperledger Fabric 服务器对区块链上保存的信息执行操作。 应用程序通过在网桥请求队列上放置请求消息来执行此操作。 请求的结果由网桥格式化为应答消息。 网桥使用请求消息的 MQMD 中的 **ReplyToQ** 和 **ReplyToQMgr** 字段中包含的信息作为应答消息的目标。

请求和应答消息是 JSON 格式的文本 (MQSTR) 消息，包含四个元素。

请求消息格式

请求消息包含以下属性:

操作
字符串-不区分大小写

`submit` 表示更新， `evaluate` 表示查询

网络
字符串-有时在 Hyperledger Fabric 中称为 `channel`

合同
String-要调用的智能合同或链代码包

args
数组-通常是字符串，但某些元素可以是嵌套的 JSON 对象。

`contract` 的实际自变量，包括方法名称。

例如:

```
{
  "operation" : "Evaluate",
  "network"   : "mychannel",
  "contract"  : "marbles0",
  "args"      : [ "readMarble" , "marble1" ]
}
```

注: 除了确保这些元素存在，并且消息是有效的 JSON 之外，网桥不会对内容进行验证。 网桥依赖于 Hyperledger Fabric 来处理请求或返回错误。

应答消息格式

应答消息将其相关标识设置为入站消息的消息标识。 任何用户定义的属性都将从请求消息复制到应答消息。 应答中的用户标识设置为发起方的用户标识。

statusCode 是 HTTP 状态码。 如果错误来自 IBM MQ 或网桥，那么将使用相应的 **statusCode** 。

statusType 是字符串 成功 或 失败。

对于成功的请求，应答消息中的 "**data**" 元素包含来自调用的 Hyperledger Composer REST API 的响应。

成功处理的示例:

```
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "data": [
      {
         "$class": "org.example.trading",
```

```
         "firstName": "John",
         "lastName": "Doe",
         "tradeId": "Trader1"
      },
      {
         "$class": "org.example.trading",
         "firstName": "Jane",
         "lastName": "Doe",
         "tradeId": "Trader2"
      }
   ]
}
```

所有错误响应都具有相同的字段，无论这些字段是由网桥本身生成，从对 Hyperledger Composer REST 服务器的调用，区块链还是从链代码调用生成。 例如：

- JSON 输入消息错误

```
{
   "statusCode": 400,
   "statusType": "FAILURE",
   "message": "[AMQBC021E] Error: Cannot parse input message or there are
    missing fields in the message. Missing fields appear to be: "method".”
}
```

- 未能由 Hyperledger Composer REST 服务器处理的请求

```
{
   "statusCode": 500,
   "statusType": "FAILURE",
   "message": "Error trying to invoke business network. Error: No valid responses
    from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
    error (status: 500, message: Error: Failed to add object with ID 'Trader1'
    as the object already exists)"
}
```

应用程序可以通过查看 **statusType** 字符串或数据字段的存在来判断请求是成功还是失败。 当处理输入消息时发生错误，并且网桥未将其发送到区块链时，从网桥返回的值为 MQRC 值，通常为 **MQRC_FORMAT_ERROR**。

z/OS Deprecated 运行 IBM MQ Bridge to blockchain 客户机样本

您可以使用随 IBM MQ Bridge to blockchain 提供的 JMS 客户机样本，将消息放入输入队列中，以供区块链网桥检查并查看接收到的应答。 此样本基于使用 IBM MQ Bridge to blockchain 与 Hyperledger Composer Trader 网络示例集成。

开始之前

请参阅 /trade_network 以获取更多信息

IBM MQ Bridge to blockchain 正在运行，并且已连接到 IBM MQ Advanced 或 IBM MQ Advanced for z/OS VUE 队列管理器以及区块链网络。

关于此任务

在 IBM MQ Bridge to blockchain 的 samp 目录中查找 JMS 样本应用程序 (ComposerBCBSamp.java)。

例如: <MQ_INSTALL_ROOT>/mqbc/samp/ComposerBCBSamp.java，其中 <MQ_INSTALL_ROOT> 是:

- Linux IBM MQ 的安装目录
- z/OS 安装 IBM MQ 的 z/OS UNIX 组件的 z/OS UNIX System Services 目录

.

过程

1. 编辑客户机样本 Java 源文件。

   遵循样本中的指示信息以将其配置为与 IBM MQ 环境和区块链网络相匹配。

   样本中的以下代码定义了要发送到网桥的三条 JSON 请求消息:

   a. 首先，要除去现有 "commodity"

   b. 其次，要创建新的 "commodity"， "owner" 和关联值，

   c. 最后，在前两条请求消息之后显示有关 "commodity" 的新信息

```
private static JSONObject[] createMessageBodies() {
    JSONObject[] msgs = new JSONObject[3]; // This method creates 3 messages
    JSONObject m, m2;
    String commodityName = "BC";

    // Clean out the commodity in case it's already there. If
    // it's not there, there will be an error returned from Composer.
    m = new JSONObject();
    m.put("method", "DELETE");
    m.put("path", "api/Commodity/" + commodityName);
    msgs[0] = m;

    // To add the item to the table, the
    // operation looks like this:
    //
    // { "method": "POST",
    //   "path": "api/Commodity",
    //   "body" : {
    //     "$class": "org.example.trading.Commodity",
    //     "tradingSymbol" : "BC",
    //     "description" : "BC",
    //     "mainExchange" : "HERE",
    //     "owner" : "Me",
    //     "quantity" : 100
    //   }
    // }
    // You can see this structure in the API Explorer
    m = new JSONObject();
    m.put("method", "POST");
    m.put("path", "api/Commodity");
    m2 = new JSONObject();
    m2.put("$class", " org.example.trading.Commodity");
    m2.put("tradingSymbol", commodityName);
    m2.put("description", "Blockchain Sample Description");
    m2.put("mainExchange", "My Exchange");
    m2.put("owner", "Me");
    m2.put("quantity", 100);
    m.put("body", m2);
    msgs[1] = m;

    // And list all items that have been created
    m = new JSONObject();
    m.put("method", "GET");
    m.put("path", "api/Commodity");
    msgs[2] = m;

    return msgs;
  }
```

2. 编译样本。

   指向网桥目录中提供的 IBM MQ 客户机类和 JSON4J.jar 文件。

```
javac -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar
ComposerBCBClient.java
```

3. 运行已编译的类。

```
java -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar:.
ComposerBCBClient
```

```
Starting Simple MQ Blockchain Bridge Client
Starting the connection.
Sent message:
  {"method":"DELETE"," path ":"api\/Commodity\/BC"}
Response text:
{
   "statusCode": 204,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": ""
}
SUCCESS
Sent message:
{"body":
{"$class":"org.example.trading.Commodity","owner":"Me","quantity":100,"description":"Blockcha
in Sample Description","mainExchange":"My
Exchange","tradingSymbol":"BC"},"operation":"POST","url":"Commodity"}
Response text:
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": {
      "$class": "org.example.trading.Commodity",
      "description": "Blockchain Sample Description",
      "mainExchange": "My Exchange",
      "owner": "Me",
      "quantity": 100,
      "tradingSymbol": "BC"
   }
}
SUCCESS
Sent message:
  {"method":"GET","path":"api\/Commodity"}
Response text:
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": [
      {
         "$class": "org.example.trading.Commodity",
         "description": "Blockchain Sample Description",
         "mainExchange": "My Exchange",
         "owner": "resource:org.example.trading.Trader#Me",
         "quantity": 100,
         "tradingSymbol": "BC"
      }
   ]
}
SUCCESS
```

message 字段包含成功处理的消息的 "OK"，或者在请求失败的情况下，包含有关失败原因的信息。

如果客户机接收到等待响应的超时错误，请检查网桥是否正在运行。

Linux Deprecated V 9.2.0 IBM MQ Bridge to blockchain 的其他配置选项

从 IBM MQ 9.2.0 开始，将更改跟踪和日志记录在 IBM MQ Bridge to blockchain 上的工作方式。

来自 IBM MQ 9.1.0 IBM MQ Bridge to blockchain 的更改

缺省情况下，除了现在开始轮换的日志文件之外，不会更改 IBM MQ 9.1.0 网桥的行为。 有关更多信息，请参阅 第 730 页的『循环日志』。

跟踪和调试交互

从 IBM MQ 9.2.0 开始，调试标志继续充当 IBM MQ 9.1.0。 即， *-d1* 提供网桥调试信息， *-d2* 为必备组件开启调试日志记录。 但是，如果在启动网桥时启用了 IBM MQ 跟踪，那么将自动开启 *-d2* 级别报告。

循环日志

从 IBM MQ 9.2.0 开始，日志文件的缺省行为是具有三个日志文件，每个文件的大小为 2 MB。 您可以使用其他配置属性来覆盖这些值。 将日志文件的现有配置属性或命令行参数作为日志的基本名称，并添加索引。

如果配置的日志文件具有:

- 无文件类型，索引将添加到文件名的末尾。

  将日志文件设置为 `abc` 会生成名为 `abc.0` 和 `abc.1` 的日志，依此类推。
- 文件类型，在文件类型之前插入索引。

  将日志文件设置为 `abc.log` 会生成名为 `abc.0.log` 和 `abc.1.log` 的日志，依此类推。

注意:

1. 由于网桥可以在具有任意用户许可权的情况下运行，因此无法对日志强制使用特定目录 (例如 `/var/mqm/qmgrs/<qm>/errors`)。
2. 相同的信息将继续写入 *stdout* 和 *stderr* 流。
3. 每当重新打开单个日志文件时，都会重新打印基本配置信息。 这些信息将始终可用，而不是仅在程序开始时打印一次。

z/OS 在 z/OS 上配置队列管理器

使用以下指示信息在 IBM MQ for z/OS 上配置队列管理器。

开始之前

在配置 IBM MQ 之前，请阅读有关 IBM MQ for z/OS 概念中的 IBM MQ for z/OS 概念的信息。

z/OS 请参阅 Planning your IBM MQ environment on z/OS 中有关如何规划 IBM MQ for z/OS 环境的信息。

关于此任务

安装 IBM MQ 后，必须先执行多项任务，然后才能使其可供用户使用。

过程

- 请参阅以下子主题以获取有关如何在 IBM MQ for z/OS 上配置队列管理器的信息。

相关概念

IBM MQ for z/OS 概念

相关任务

第 7 页的『在 Multiplatforms 版上创建队列管理器』

必须先创建并启动至少一个队列管理器及其关联对象，然后才能使用消息和队列。 队列管理器管理与其关联的资源，特别是其拥有的队列。 它为消息排队接口 (MQI) 调用和命令的应用程序提供排队服务，以创建，修改，显示和删除 IBM MQ 对象。

保护

第 152 页的『配置分布式队列』

此部分提供有关 IBM MQ 安装之间的相互通信的更详细信息，包括队列定义，通道定义，触发和同步点过程

第 13 页的『配置客户机与服务器之间的连接』

要配置 IBM MQ MQI clients 与服务器之间的通信链路，请决定通信协议，定义链路两端的连接，启动侦听器以及定义通道。

z/OS 管理 IBM MQ for z/OS

规划

z/OS 发出命令

相关参考

z/OS IBM MQ for z/OS 实用程序

z/OS 准备在 z/OS 上定制队列管理器

使用可安装功能部件的详细信息，本地语言功能部件以及有关测试和设置安全性的信息来定制队列管理器时，请使用本主题。

准备定制

Program Directory 列出了 IBM MQ 安装磁带的内容， IBM MQ 的程序和服务级别信息，并描述了如何使用 System Modification Program Extended (SMP/E) 来安装 IBM MQ for z/OS 。 有关程序目录的下载链接，请参阅 IBM MQ for z/OS 程序目录 PDF 文件。

安装 IBM MQ 后，必须执行多项任务，然后才能使其可供用户使用。 请参阅以下部分以获取这些任务的描述:

- 第 734 页的『设置 IBM MQ for z/OS』
- 第 790 页的『在 z/OS 上测试队列管理器』
- 在 z/OS 上设置安全性

如果要从先前版本的 IBM MQ for z/OS 进行迁移，那么不需要执行大部分定制任务。 有关必须执行的任务的更多信息，请参阅 维护和迁移 。

IBM MQ for z/OS 的可安装功能部件

IBM MQ for z/OS 包含以下功能部件:

基本版
这是必需的; 它包含所有主要功能，包括

- 管理和公用事业
- 支持使用 IBM MQ Application Programming Interface 或 C++ 的 CICS， IMS 和批处理类型应用程序
- 分布式排队设施 (同时支持 TCP/IP 和 APPC 通信)

本地语言功能
其中包含所有受支持本地语言的错误消息和面板。 每种语言都有一个与其关联的语言字母。 语言和字母为:

C
简体中文

E
U.S. 英语 (混合大小写)

F
法语

K
日语

U
U.S. 英语 (大写)

必须安装美国英语 (混合大小写) 选项。 您还可以安装一种或多种其他语言。 (其他语言的安装过程需要安装美国英语 (混合大小写) ，即使你不会使用美国英语 (混合大小写)。)

IBM MQ for z/OS UNIX System Services Components
V 9.2.0 此功能可选。 如果要构建并运行使用 Java Message Service (JMS) 连接到 IBM MQ for z/OS 的 Java 应用程序，请选择此功能。

IBM MQ for z/OS UNIX System Services Web Components
此功能可选。

如果要使用 IBM MQ Console 或 REST API，请选择此功能。

必须安装 IBM MQ for z/OS UNIX System Services Components 功能部件才能安装此功能部件。

V 9.2.0 **IBM MQ for z/OS Managed File Transfer**

此功能部件是可选的，仅当您具有 IBM MQ Advanced for z/OS， IBM MQ for z/OS Value Unit Edition (VUE)或 IBM MQ for z/OS Managed File Transfer 的权利时，才应安装此功能部件。

如果要使用 IBM MQ for z/OS 的 Managed File Transfer 功能，请选择此功能。

必须安装 IBM MQ for z/OS UNIX System Services Components 功能部件才能安装此功能部件。

安装后存在的库

IBM MQ 随附了多个单独的装入库。 第 732 页的表 71 显示了安装 IBM MQ 后可能存在的库。

表 *71:* 安装后存在的 *IBM MQ* 库

名称	描述
thlqual.SCSQANLC	包含 IBM MQ 的简体中文版本的装入模块。
thlqual.SCSQANLE	包含 U.S 的装入模块。 IBM MQ 的英语 (混合大小写) 版本。
thlqual.SCSQANLF	包含 IBM MQ 的法语版本的装入模块。
thlqual.SCSQANLK	包含日语版本的 IBM MQ 的装入模块。
thlqual.SCSQANLU	包含 U.S 的装入模块。 IBM MQ 的英语 (大写) 版本。
thlqual.SCSQASMS	包含汇编程序样本程序的源。
thlqual.SCSQAUTH	所有 IBM MQ 产品装入模块的主存储库; 它还包含缺省参数模块 CSQZPARM。 此库必须是 APF 授权的且采用 PDS-E 格式。
thlqual.SCSQCICS	包含必须包含在 CICS DFHRPL 并置中的额外装入模块。 此库必须是 APF 授权的且采用 PDS-E 格式。
thlqual.SCSQCLST	包含样本程序使用的 CLISTs。
thlqual.SCSQCOBC	包含 COBOL 副本，包括样本程序所需的副本。
thlqual.SCSQCOBS	包含 COBOL 样本程序的源。
thlqual.SCSQCPPS	包含 C++ 样本程序的源。
thlqual.SCSQC37S	包含 C 样本程序的源。
thlqual.SCSQC370	包含 C 头，包括样本程序所需的头。
thlqual.SCSQDEFS	包含 C++ 的侧定义以及用于共享排队的 Db2 DBRM。
thlqual.SCSQEXEC	包含要包含在 SYSEXEC 或 SYSPROC 并置中的 REXX 可执行文件 (如果您正在使用 IBM MQ 操作和控制面板)。
V 9.2.0 thlqual.SCSQFCMD	包含用于创建和运行 Managed File Transfer 任务的作业的模板。
thlqual.SCSQHPPS	包含 C++ 的头文件。
thlqual.SCSQINST	包含用于安装作业的 JCL。
thlqual.SCSQLINK	早期代码库。 包含在系统初始程序装入 (IPL) 时装入的装入模块。 库必须经过 APF 授权。
thlqual.SCSQLOAD	装入库。 包含非 APF 代码，用户出口，实用程序，样本，安装验证程序和适配器存根的装入模块。 库不需要 APF 授权，也不需要在链接列表中。 此库必须采用 PDS-E 格式。
thlqual.SCSQMACS	包含汇编程序宏，包括: 样本宏，产品宏和系统参数宏。

表 71: 安装后存在的 *IBM MQ* 库 (继续)

名称	描述
thlqual.SCSQMAPS	包含样本程序使用的 CICS 映射集。
thlqual.SCSQMSGC	包含要包含在 ISPMLIB 并置中的 ISPF 消息 (如果要将简体中文语言功能用于 IBM MQ 操作和控制面板)。
thlqual.SCSQMSGE	包含要包含在 ISPMLIB 并置中的 ISPF 消息 (如果您使用的是 U.S)。 IBM MQ 操作和控制面板的英语 (混合大小写) 语言功能。
thlqual.SCSQMSGF	包含要包含在 ISPMLIB 并置中的 ISPF 消息 (如果要将法语语言功能用于 IBM MQ 操作和控制面板)。
thlqual.SCSQMSGK	包含要包含在 ISPMLIB 并置中的 ISPF 消息 (如果要将日语功能用于 IBM MQ 操作和控制面板)。
thlqual.SCSQMSGU	包含要包含在 ISPMLIB 并置中的 ISPF 消息 (如果您使用的是 U.S)。 IBM MQ 操作和控制面板的英语 (大写) 语言功能。
thlqual.SCSQMVR1	包含用于分布式排队的装入模块。 此库必须是 APF 授权的且采用 PDS-E 格式。
thlqual.SCSQPLIC	包含 PL/I 包含文件。
thlqual.SCSQPLIS	包含 PL/I 样本程序的源。
thlqual.SCSQPNLA	包含要包含在 ISPPLIB 并置中的转储格式化程序的 IPCS 面板。 还包含 IBM MQ 样本程序的面板。
thlqual.SCSQPNLC	包含要包含在 ISPPLIB 并置中的 ISPF 面板 (如果要将简体中文功能用于 IBM MQ 操作和控制面板)。
thlqual.SCSQPNLE	包含要包含在 ISPPLIB 并置中的 ISPF 面板 (如果您正在使用 U.S)。 IBM MQ 操作和控制面板的英语 (混合大小写) 语言功能。
thlqual.SCSQPNLF	包含要包含在 ISPPLIB 并置中的 ISPF 面板 (如果要将法语语言功能用于 IBM MQ 操作和控制面板)。
thlqual.SCSQPNLK	包含要包含在 ISPPLIB 并置中的 ISPF 面板 (如果要将日语功能用于 IBM MQ 操作和控制面板)。
thlqual.SCSQPNLU	包含要包含在 ISPPLIB 并置中的 ISPF 面板 (如果您正在使用 U.S)。 IBM MQ 操作和控制面板的英语 (大写) 语言功能。
thlqual.SCSQPROC	包含样本 JCL 和缺省系统初始化数据集。
thlqual.SCSQSNLC	包含特殊用途功能 (例如，早期代码) 所需的 IBM MQ 模块简体中文版本的装入模块。
thlqual.SCSQSNLE	包含 U.S 的装入模块。 特殊用途函数 (例如，早期代码) 所需的 IBM MQ 模块的英语 (混合大小写) 版本。
thlqual.SCSQSNLF	包含特殊用途功能 (例如，早期代码) 所需的 IBM MQ 模块的法语版本的装入模块。
thlqual.SCSQSNLK	包含特殊用途功能 (例如，早期代码) 所需的日文版 IBM MQ 模块的装入模块。
thlqual.SCSQSNLU	包含 U.S 的装入模块。 特殊用途函数 (例如，早期代码) 所需的 IBM MQ 模块的英语 (大写) 版本。
thlqual.SCSQTBLC	包含要包含在 ISPTLIB 并置中的 ISPF 表 (如果要对 IBM MQ 操作和控制面板使用简体中文功能)。

表 71: 安装后存在的 *IBM MQ* 库 (继续)	
名称	**描述**
thlqual.SCSQTBLE	包含要包含在 ISPTLIB 并置中的 ISPF 表 (如果您使用的是 U.S)。IBM MQ 操作和控制面板的英语 (混合大小写) 语言功能。
thlqual.SCSQTBLF	包含要包含在 ISPTLIB 并置中的 ISPF 表 (如果要将法语语言功能用于 IBM MQ 操作和控制面板)。
thlqual.SCSQTBLK	包含要包含在 ISPTLIB 并置中的 ISPF 表 (如果要将日语功能用于 IBM MQ 操作和控制面板)。
thlqual.SCSQTBLU	包含要包含在 ISPTLIB 并置中的 ISPF 表 (如果您使用的是 U.S)。IBM MQ 操作和控制面板的英语 (大写) 语言功能。

注: 请勿修改或定制任何这些库。 如果要进行更改，请复制库并对副本进行更改。

相关概念

IBM MQ for z/OS 概念

第 823 页的『将 IBM MQ 用于 IMS』

IBM MQ -IMS 适配器和 IBM MQ - IMS 网桥是允许 IBM MQ 与 IMS 进行交互的两个组件。

第 829 页的『将 IBM MQ 与 CICS 结合使用』

要将 IBM MQ 与 CICS 配合使用，必须配置 IBM MQ CICS 适配器以及 (可选) IBM MQ CICS bridge 组件。

第 832 页的『在 IMS 中使用 OTMA 出口』

如果要将 IMS Open Transaction Manager Access 出口与 IBM MQ for z/OS 配合使用，请使用本主题。

相关任务

第 797 页的『在 z/OS 上设置与其他队列管理器的通信』

本部分描述了在开始使用分布式排队之前需要进行的 IBM MQ for z/OS 准备工作。

管理 IBM MQ for z/OS

相关参考

第 830 页的『将服务升级并应用于 Language Environment 或 z/OS Callable Services』

您必须执行的操作根据您使用 CALLLIBS 还是 LINK 以及 SMP/E 的版本而有所不同。

z/OS 设置 IBM MQ for z/OS

使用本主题作为定制 IBM MQ for z/OS 系统的逐步指南。

配置队列管理器的最佳方法是按所示顺序执行以下步骤:

1. 配置基本队列管理器。
2. 配置通道启动程序，该通道启动程序执行队列管理器到队列管理器的通信以及远程客户机应用程序通信。
3. 如果要加密或保护消息，请配置 Advanced Message Security for z/OS。
4. 如果要使用 IBM MQ 来传输文件，请配置 Managed File Transfer for z/OS。
5. 如果要使用管理或消息传递 REST API 或 MQ Console 从 Web 浏览器管理 IBM MQ ，请配置 mqweb 服务器。

本主题将引导您在成功安装 IBM MQ 之后完成设置的各个阶段。 "程序目录" 中描述了安装过程。 有关程序目录的下载链接，请参阅 IBM MQ for z/OS 程序目录 PDF 文件。

随 IBM MQ 一起提供的样本可帮助您进行定制。 样本数据集成员的名称以四个字符 CSQ4 开头，位于库 thlqual.SCSQPROC 中。

在执行本主题中描述的定制任务之前，必须考虑许多配置选项，因为它们会影响 IBM MQ for z/OS 的性能和资源需求。 例如，您必须决定要使用哪些全球化库。

如果要自动执行某些定制步骤，请参阅 第 835 页的『使用 IBM z/OSMF 自动执行 IBM MQ』。

配置选项

有关这些选项的更多信息，请参阅 在 z/OS 上规划。

本节中每个任务的描述指示是否:

- 该任务是设置 IBM MQ 的过程的一部分。 即，在 z/OS 系统上定制 IBM MQ 时执行一次任务。 (在并行综合系统中，必须对综合系统中的每个 z/OS 系统执行该任务，并确保以相同方式设置每个 z/OS 系统。)
- 该任务是添加队列管理器的一部分。 即，在添加队列管理器时，对每个队列管理器执行一次任务。

如果使用命令来更改各种 z/OS 系统参数并按建议执行 第 746 页的『更新 SYS1.PARMLIB 成员』，那么任何任务都不需要您对 z/OS 系统执行 IPL。

为了简化操作并帮助确定问题，请确保以相同的方式设置综合系统中的所有 z/OS 系统，以便可以在紧急情况下在任何系统上快速创建队列管理器。

为了便于维护，请考虑定义别名以引用 IBM MQ 库; 有关更多信息，请参阅 使用别名来引用 IBM MQ 库。

相关概念

IBM MQ for z/OS 概念

第 823 页的『将 IBM MQ 用于 IMS』
IBM MQ -IMS 适配器和 IBM MQ - IMS 网桥是允许 IBM MQ 与 IMS 进行交互的两个组件。

第 829 页的『将 IBM MQ 与 CICS 结合使用』
要将 IBM MQ 与 CICS 配合使用，必须配置 IBM MQ CICS 适配器以及 (可选) IBM MQ CICS bridge 组件。

第 832 页的『在 IMS 中使用 OTMA 出口』
如果要将 IMS Open Transaction Manager Access 出口与 IBM MQ for z/OS 配合使用，请使用本主题。

相关任务

第 797 页的『在 z/OS 上设置与其他队列管理器的通信』
本部分描述了在开始使用分布式排队之前需要进行的 IBM MQ for z/OS 准备工作。

管理 IBM MQ for z/OS

相关参考

第 830 页的『将服务升级并应用于 Language Environment 或 z/OS Callable Services』
您必须执行的操作根据您使用 CALLLIBS 还是 LINK 以及 SMP/E 的版本而有所不同。

z/OS 为 IBM MQ 配置 z/OS 系统

使用这些主题作为定制 IBM MQ for z/OS 系统的逐步指南。

z/OS 标识 *z/OS* 系统参数

某些任务涉及更新 z/OS 系统参数。 您需要知道在执行系统 IPL 时指定了哪些值。

- 您需要对要运行 *IBM MQ* 的每个 *z/OS* 系统执行一次此任务。
- 从先前版本进行迁移时，可能需要执行此任务。

SYS1.PARMLIB(IEASYSpp) 包含指向 SYS1.PARMLIB (其中 pp 表示用于对系统执行 IPL 的 z/OS 系统参数列表)。

您需要查找的条目包括:

对于 第 736 页的『APF 授权 IBM MQ 装入库』:
PROG=xx 或 APF=aa 指向授权程序设施 (APF) 授权库列表 (成员 PROGxx 或 IEFAPFaa)

对于 第 736 页的『更新 z/OS 链接列表和 LPA』:
LNK=kk 指向链接列表 (成员 LNKLSTkk) LPA=mm 指向 LPA 列表 (成员 LPALSTmm)

对于 第 740 页的『更新 z/OS 程序属性表』:
SCH=xx 指向程序属性表 (PPT) (成员 SCHEDxx)

对于 第 740 页的『将 IBM MQ 子系统定义到 z/OS』:
SSN=ss 指向定义的子系统列表 (成员 IEFSSNss)

z/OS APF 授权 IBM MQ 装入库

APF-授权各种库。 某些装入模块可能已获得授权。

- 您需要对要运行 *IBM MQ* 的每个 *z/OS* 系统执行一次此任务。
- 如果您正在使用队列共享组，那么必须确保 *IBM MQ* 的设置在综合系统中的每个 *z/OS* 系统上都相同。
- 从先前版本进行迁移时，可能需要执行此任务。
- 使用库外观 *(LLA):*
  - 某些 *IBM MQ* 用法可能会导致高输入/输出 *(IO)* 从库装入模块。 可通过使用操作系统的 *LLA* 工具来减少此 *IO*。
  - 此高 *IO* 可能在以下期间发生*:*
    - 具有高 *MQCONN/MQDISC* 速率的应用程序，例如在 *WLM* 存储过程中。
    - 正在装入通道出口。 如果您有频繁启动和停止的通道，请使用通道出口。
  - *SYS1.PARMLIB* 指定 *LLA* 设置。 在 *LIBRARIES* 语句中包含库名意味着程序副本将始终从 *VLF (*虚拟后备设施*)* 获取，因此在大量使用时通常不需要 *I/O*。

    包含在冻结语句中意味着没有 *I/O* 来获取相关 *DD* 语句并置目录 *(*这通常可能比程序装入本身的 *I/O* 更多*)*。

    使用操作系统命令 *" F LLA , REFRESH "* 在对这些库中的任何库进行任何更改之后。

IBM MQ 装入库 thlqual.SCSQAUTH 和 thlqual.SCSQLINK 必须是 APF 授权的。 您还必须对本地语言功能部件 (thlqual.SCSQANLx 和 thlqual.SCSQSNLx) 和分布式排队功能部件 (thlqual.SCSQMVR1) 的库进行 APF 授权。

但是， LPA 中的所有装入模块都将自动获得 APF 授权。 如果 SYS1.PARMLIB 成员 IEASYSpp 包含以下语句:

```
LNKAUTH=LNKLST
```

如果未指定 LNKAUTH , 那么缺省值为 LNKAUTH=LNKLST。

根据您选择放入 LPA 或链接列表中的内容 (请参阅 第 736 页的『更新 z/OS 链接列表和 LPA』) , 您可能不需要将库放入 APF 链接列表中

注: 必须对 IBM MQ STEPLIB 中包含的所有库进行 APF 授权。 如果在 STEPLIB 中放置未经 APF 授权的库，那么整个库并置将失去其 APF 授权。

APF 列表位于 SYS1.PARMLIB 成员 PROGxx 或 IEAAPFaa。 这些列表包含 APF 授权 z/OS 库的名称。 列表中的条目顺序不重要。 有关 APF 列表的信息，请参阅 APF 授权的库列表 。

有关调整系统的更多信息，请参阅 SupportPac MP16

如果使用具有动态格式的 PROGxx 成员，那么只需发出 z/OS 命令 SETPROG APF,ADD,DSNAME=hlq.SCSQ *XXXX*,VOLUME= *YYYYYY* 以使更改生效: 其中 *XXXX* 因库名而异，而其中 *YYYYY* 是卷。 否则，如果使用静态格式或 IEAAPFaa 成员，那么必须在系统上执行 IPL。

请注意，必须在 APF 列表中使用库的实际名称。 如果尝试使用库的数据集别名，那么授权将失败。

相关概念

第 736 页的『更新 z/OS 链接列表和 LPA』
使用新版本的早期代码库更新 LPA 库。 其他代码可以进入链接列表或 LPA。

第 731 页的『准备在 z/OS 上定制队列管理器』
使用可安装功能部件的详细信息，本地语言功能部件以及有关测试和设置安全性的信息来定制队列管理器时，请使用本主题。

z/OS 更新 z/OS 链接列表和 LPA

使用新版本的早期代码库更新 LPA 库。 其他代码可以进入链接列表或 LPA。

- 您需要对要运行 IBM MQ 的每个 z/OS 系统执行一次此任务。
- 如果您正在使用队列共享组，那么应该先将 QSG 中每个队列管理器中的早期代码刷新到 IBM MQ 9.2.0 级别，然后再将任何队列管理器迁移到 IBM MQ 9.2.0。

  在每个 LPAR 上安装最新的早期代码，然后在迁移之前的某个时刻一次刷新一个队列管理器。 您不必同时迁移所有队列管理器。
- 从先前版本迁移时，可能需要执行此任务。 有关更多详细信息，请参阅 "程序目录"。 有关程序目录的下载链接，请参阅 IBM MQ for z/OS 程序目录 PDF 文件。

注: LPA 的数据集特定于版本。 如果您正在系统中使用现有 LPA，请与系统管理员联系以决定要使用的 LPA。

早期代码

需要将某些 IBM MQ 装入模块添加到 MVS，以便 IBM MQ 充当子系统。 这些模块称为 "早期代码"，即使队列管理器未处于活动状态，也可以执行这些模块。 例如，在控制台上发出带有 IBM MQ 命令前缀的操作员命令时，此早期代码将获取控制权并检查是否需要启动队列管理器，或者是否需要将请求传递到正在运行的队列管理器。 此代码将装入到链接装配区域 (LPA) 中。 有一组 "早期" 模块，用于所有队列管理器，这些模块需要处于 IBM MQ 的最高级别。 来自较高版本的 IBM MQ 的早期代码将使用具有较低版本的 IBM MQ 的队列管理器，但不会使用相反的队列管理器。

早期代码包含以下装入模块:

- CSQ3INI 和 CSQ3EPX 库中的 thlqual.SCSQLINK
- CSQ3ECMX 库中的 thlqual.SCSQSNL*x*，其中 *x* 是您国家语言的字母:
  - C，中文
  - E，表示美式英语混合大小写
  - F，法语
  - K，代表日语
  - U，表示美国英语大写字母

IBM MQ 包括用户修改，将 thlqual.SCSQSNL*x* 库的内容移入 thlqual.SCSQLINK 并通知 SMP/E。 此用户修改称为 CSQ8UERL，在 *Program Directory for IBM MQ for z/OS* 中针对 Long Term Support 或 Continuous Delivery 进行了描述。 有关程序目录的下载链接，请参阅 IBM MQ for z/OS 程序目录 PDF 文件。

在更新 LPA 库中的早期代码后，可从下一个 z/OS IPL (使用 CLPA 选项) 到 IPL 期间从 SYS1.PARMLIB。

对于以后添加的任何新队列管理器子系统，可以立即使其可用而不执行 IPL (如 第 740 页的『将 IBM MQ 子系统定义到 z/OS』 中所述) 将其添加到 LPA，如下所示:

- 如果未使用 CSQ8UERL，请发出以下 z/OS 命令:

```
SETPROG LPA,ADD,MODNAME=(CSQ3INI,CSQ3EPX),DSNAME=thlqual.SCSQLINK
SETPROG LPA,ADD,MODNAME=(CSQ3ECMX),DSNAME=thlqual.SCSQSNLx
```

- 如果使用了 CSQ8UERL，那么可以使用以下 z/OS 命令将早期代码装入到 LPA 中:

```
SETPROG LPA,ADD,MASK=*,DSNAME=thlqual.SCSQLINK
```

- 如果您正在使用 Advanced Message Security，那么还必须发出以下 z/OS 命令以在 LPA 中包含其他模块:

```
SETPROG LPA,ADD,MODNAME=(CSQ0DRTM),DSNAME=thlqual.SCSQLINK
```

如果已应用维护，或者打算重新启动具有更高版本或发行版的 IBM MQ 的队列管理器，那么可以使用以下步骤使早期代码可供现有队列管理器使用。 未对其执行这些步骤的队列管理器将继续使用它们已在使用的早期代码版本。 不需要对 LPAR 上的所有队列管理器执行这些步骤，除非您专门尝试对所有队列管理器应用维护，或者将所有队列管理器更新为 IBM MQ 的最新版本或发行版。

1. 使用 z/OS SETPROG 命令将其添加到 LPA，如本主题中先前所述。
2. 使用 IBM MQ 命令 STOP QMGR 停止队列管理器。

3. 确保 qmgr.REFRESH.QMGR 安全概要文件。 请参阅 MQSC 命令，概要文件及其访问级别。
4. 使用 IBM MQ 命令 REFRESH QMGR TYPE (EARLY) 刷新队列管理器的早期代码。
5. 使用 IBM MQ 命令 START QMGR 重新启动队列管理器。

MQSC 命令中描述了 IBM MQ 命令 STOP QMGR， REFRESH QMGR 和 START QMGR。

其他代码

以下库中的所有 IBM MQ 提供的装入模块都是可重入的，并且可以放在 LPA 中:

- SCSQAUTH
- SCSQANL*x*，其中 *x* 是您国家语言的语言字母
- SCSQMVR1

要点: 但是，如果将库放在 LPA 中，那么无论何时应用维护，都必须将任何已更改的模块手动复制到 LPA 中。 因此，最好将 IBM MQ 装入库放在链接列表中，通过发出 z/OS 命令 MODIFY LLA REFRESH 可以在维护后更新这些库。

请参阅 修改 LNKLST 数据集的内容 以获取更多信息，以及 安全正确地使用动态 LNKLST 设施。

这特别建议用于 SCSQAUTH， 以便您不必将其包含在多个 STEPLIBs 中。 只有一个语言库 SCSQANL *x* 应该放在 LPA 或链接列表中。 链接列表库在 SYS1.PARMLIB。

分布式排队设施和 CICS bridge (而不是队列管理器本身) 需要访问语言环境 (LE) 运行时库 SCEERUN。 如果使用其中任一设施，那么需要在链接列表中包含 SCEERUN。

某些模块在队列管理器启动时装入到 ECSA 中。 在受 ECSA 约束的环境中，可以改为将这些模块放入 LPA 中。 请参阅 第 738 页的『将 IBM MQ 全局模块放入 LPA 中』 以获取更多信息。

要点: 要在 IBM MQ 9.2 上使用此工具，需要应用 APAR PH52358。

相关概念
第 740 页的『更新 z/OS 程序属性表』
IBM MQ 队列管理器需要一些其他 PPT 条目。

z/OS 将 *IBM MQ* 全局模块放入 *LPA* 中
当 IBM MQ for z/OS 队列管理器启动时，它会将其部分装入模块 (全局模块) 装入到扩展公共服务区 (ECSA) 中。 在队列管理器关闭时，将释放 ECSA。

要点: 要在 IBM MQ 9.2 上使用此工具，需要应用 APAR PH52358。

有 19 个全局模块 (位于 IBM MQ 9.2)， 每个正在运行的队列管理器大约使用 1.2 MB 的 ECSA。 在每个 LPAR 运行多个队列管理器的环境中，如果由于 ECSA 或高专用约束而需要减少 ECSA 使用量，那么可以将全局模块放入 LPA 中。

注: 虽然 CSQ7GPLM 是全局模块，但不应将其添加到 LPA。

如果队列管理器在其 STEPLIB 中找不到全局模块，并且检测到该模块在 LPA 中，那么它将直接使用 LPA 副本，而不是将该模块的副本装入到 ECSA 中。 或者，如果通常从链接列表装入队列管理器代码，那么将优先装入 LPA 中的任何全局模块，而不装入链接列表中的任何全局模块。

z/OS 公共存储器跟踪功能 (请参阅 使用公共存储器跟踪功能) 跟踪每个队列管理器的每个队列管理器的 MSTR 地址空间下的存储器，并可用于检测全局模块正在使用的空间量。

缺省情况下，全局模块位于 SCSQAUTH 装入库中。 如果队列管理器的 MSTR 地址空间通过 STEPLIB 并置找到 SCSQAUTH， 那么将优先使用来自该队列管理器的全局模块，并将其装入到 ECSA 中。

全局模块包括:

CSQ0GPLM, CSQ3AMGP, CSQ3SSGP, CSQ9PREP,
CSQ9SCNB， CSQGGPLM， CSQMCGLM， CSQMGPLM， CSQRGLM1,
CSQSLD1， CSQVGEPL， CSQVSRX， CSQWDLD2， CSQWDLD3
CSQWVZSA， CSQWZDG0， CSQWVZPS， CSQWVGTM 和 CSQZTDDM

注意:

- IBM MQ 的全局模块的名称在不同 IBM MQ 版本中保持不变。 因此，如果将全局模块装入到 LPA 中，那么这些模块应该来自单个 IBM MQ 版本，并且仅应由在同一 IBM MQ 版本中运行的队列管理器使用。
- 如果 IBM MQ 的多个版本在同一 LPAR 上运行，那么在任何给定时间，只有其中一个版本可以在 LPA 中具有其全局模块。

如果将维护应用于将全局模块装入到 LPA 中的 IBM MQ 安装，并且该维护会更新任何全局模块，那么您应该再次执行以下文本中描述的过程。

过程

要将 IBM MQ 版本中的全局模块放入 LPA 中，请执行以下步骤:

1. 创建 `thlqual.SCSQAUTH` 装入库及其内容的副本，例如: `thlqual.LOCAL.SCSQAUTH`。 确保使用外部安全管理器 (ESM) 保护此装入库免受未经授权的访问。
2. APF 授权 `thlqual.LOCAL.SCSQAUTH` 装入库; 请参阅 第 736 页的『APF 授权 IBM MQ 装入库』。
3. 使用与 `thlqual.LOCAL.SCSQAUTH` 相同的属性创建新的 `thlqual.GLOBAL.SCSQAUTH` 装入库。

   注: 此装入库不需要经过 APF 授权。 确保使用 ESM 保护此装入库免受未经授权的访问。
4. 将 19 个全局模块从 `thlqual.LOCAL.SCSQAUTH` 复制到 `thlqual.GLOBAL.SCSQAUTH`。
5. 从 `thlqual.LOCAL.SCSQAUTH` 中删除 19 个全局模块。
6. 通过以下任一方法将 19 个全局模块从 `thlqual.GLOBAL.SCSQAUTH` 放入 LPA 中:

   a. a. 正在将 `thlqual.GLOBAL.SCSQAUTH` 添加到 SYS1.PARMLIB 的 LPALSTxx 成员中。 然后，必须使用 CLPA 选项对系统进行 IPL ， 以确保将库内容装入 PLPA 中。

   b. b. 使用以下命令将模块动态添加到 LPA:

```
SETPROG
LPA,ADD,MODNAME=(CSQ0GPLM,CSQ3AMGP,CSQ3SSGP,CSQ9PREP,CSQ9SCNB,CSQGGPLM,
CSQMCGLM,CSQMGPLM,CSQRGLM1,CSQSLD1,CSQVGEPL,CSQVSRX,CSQWDLD2,CSQWDLD3,
CSQWVZSA,CSQWZDG0,CSQWVZPS,CSQWVGTM,CSQZTDDM),DSNAME= thlqual.GLOBAL.SCSQAUTH
```

   注: LPALSTxx 是在 LPA 中放置模块的首选长期方法。
7. 通过发出以下命令来验证模块是否在 LPA 中:

```
D PROG,LPA,MODNAME=CSQMCGLM
```

如果成功将模块装入到 LPA 中，那么该命令的输出应指示该模块的入口点和装入点。

对于需要使用 LPA 中的全局模块的每个队列管理器，如果通常放置:

1. 链接列表中的 `thlqual.SCSQAUTH` ，只需停止并启动队列管理器。 将从 LPA 装入全局模块，并从链接列表装入本地模块。
2. MSTR JCL STEPLIB 中的 `thlqual.SCSQAUTH` ， 请更改 JCL ， 以使 STEPLIB 使用 `thlqual.LOCAL.SCSQAUTH` 而不是 `thlqual.SCSQAUTH`。 停止并启动队列管理器; 将从 LPA 装入全局模块，并从 STEPLIB 装入本地模块。

CHIN 和 AMSM JCL 可以像任何 IBM MQ 应用程序一样继续使用 `thlqual.SCSQAUTH` 。

要还原队列管理器以将全局模块装入到 ECSA 中，请执行以下步骤:

1. 停止队列管理器
2. 在下一次 IPL 时，通过除去 LPALSTxx 定义或使用以下命令从 LPA 中除去全局模块:

```
SETPROG LPA,DELETE,MODNAME=(xxx) FORCE=YES
```

3. 如果 `thlqual.LOCAL.SCSQAUTH` 位于队列管理器的 STEPLIB 中，请将其替换为 `thlqual.SCSQAUTH`。
4. 重新启动队列管理器。

相关概念

第 736 页的『更新 z/OS 链接列表和 LPA』
使用新版本的早期代码库更新 LPA 库。 其他代码可以进入链接列表或 LPA。

z/OS 更新 *z/OS* 程序属性表

IBM MQ 队列管理器需要一些其他 PPT 条目。

- 必须对要运行 *IBM MQ* 的每个 *z/OS* 系统执行一次此任务。
- 如果您正在使用队列共享组，那么必须确保 *IBM MQ* 的设置在综合系统中的每个 *z/OS* 系统上都相同。
- 从先前版本迁移时，不需要执行此任务。
- 当您需要 *Advanced Message Security* 时，需要执行此任务的 *CSQ0DSRV* 部分。

thlqual.SCSQPROC(CSQ4SCHD) 中提供了包含所有必需 PPT 条目的样本。 确保将所需条目添加到 PPT ，您可以在 SYS1.PARMLIB(SCHEDxx)。

在 z/OS 1.12 和更高版本中， CSQYASCP 已定义到具有详细属性的操作系统，并且不再需要包含在 PARMLIB 的 SCHEDxx 成员中。

IBM MQ 队列管理器控制交换自身。 但是，如果您具有大量负载的 IBM MQ 网络，并且响应时间很关键，那么通过添加 CSQXJST PPT 条目来使 IBM MQ 通道启动程序不可交换可能会有影响 z/OS 系统其余部分的性能的风险。

如果需要 Advanced Message Security，请添加 CSQ0DSRV PPT 条目。

发出 z/OS 命令 **SET SCH=xx**，其中 *xx* 是 PARMLIB 的 SCHEDxx 成员的后缀，以使这些更改生效。

相关概念

第 740 页的『将 IBM MQ 子系统定义到 z/OS』
更新子系统名称表并决定命令前缀字符串的约定。

z/OS 配置队列管理器和通道启动程序

使用这些主题作为配置队列管理器和通道启动程序的逐步指南。

z/OS 将 *IBM MQ* 子系统定义到 *z/OS*

更新子系统名称表并决定命令前缀字符串的约定。

对每个 IBM MQ 队列管理器重复此任务。 从先前版本迁移时，不需要执行此任务。

相关概念

第 743 页的『为 IBM MQ 队列管理器创建过程』
每个 IBM MQ 子系统都需要一个编目过程来启动队列管理器。 您可以创建自己的或使用 IBM 提供的过程库。

z/OS 更新子系统名称表

定义 IBM MQ 子系统时，必须向子系统名称表添加一个条目。

z/OS 的子系统名称表，最初从 SYS1.PARMLIB 成员 IEFSSNss 包含正式定义的 z/OS 子系统的定义。 要定义每个 IBM MQ 子系统，必须通过更改 SYS1.PARMLIB，或者最好使用 z/OS 命令 SETSSI。

IBM MQ 子系统初始化支持并行处理，因此可以在 z/OS V1.12 和更高版本的 IEFSSNss 表中的 BEGINPARALLEL 关键字的上方和下方添加 IBM MQ 子系统定义语句。

如果使用 SETSSI 命令，那么更改将立即生效，并且不需要对系统执行 IPL。 确保更新 SYS1.PARMLIB 也是如此，如 第 746 页的『更新 SYS1.PARMLIB 成员』 中所述，以便这些更改在后续 IPL 之后仍然有效。

用于动态定义 IBM MQ 子系统的 SETSSI 命令为:

```
SETSSI ADD,S=ssid,I=CSQ3INI,P='CSQ3EPX,cpf,scope'
```

可以通过两种方式之一指定 IEFSSNss 中的相应信息:

- IEFSSNss 中 IBM MQ 子系统定义的关键字参数格式。 这是建议的方法。

```
SUBSYS SUBNAME(ssid) INITRTN(CSQ3INI) INITPARM('CSQ3EPX,cpf,scope')
```

- IBM MQ 子系统定义的位置参数格式。

```
ssid,CSQ3INI,'CSQ3EPX,cpf,scope'
```

请勿在一个 IEFSSNss 成员中混用这两种格式。 如果需要不同的格式，请对每种类型使用单独的 IEFSSNss 成员，将新成员的 SSN 操作数添加到 IEASYSpp SYS1.PARMLIB 成员。 要指定多个 SSN ，请使用 SSN = (aa ， bb ， ...) 在 IEASYSpp 中。

在示例中,

ssid
子系统标识。 它最长为 4 个字符。 所有字符都必须是字母数字 (大写 A 到 Z ， 0 到 9) ，必须以字母字符开头。 队列管理器将具有与子系统相同的名称，因此只能使用 z/OS 子系统名称和 IBM MQ 对象名称所允许的字符。

cpf
命令前缀字符串 (请参阅 第 741 页的『定义命令前缀字符串 (CPF)』 以获取有关 CPF 的信息)。

scope
在 z/OS 综合系统中运行时使用的系统作用域 (请参阅 第 742 页的『综合系统环境中的 CPF』 以获取有关系统作用域的信息)。

第 741 页的图 102 显示了 IEFSSNss 语句的多个示例。

```
CSQ1,CSQ3INI,'CSQ3EPX,+mqs1cpf,S'
CSQ2,CSQ3INI,'CSQ3EPX,+mqs2cpf,S'
CSQ3,CSQ3INI,'CSQ3EPX,++,S'
```

图 *102:* 用于定义子系统的样本 *IEFSSNss* 语句

注: 在子系统中创建对象时，不能更改子系统名称或使用另一子系统中的一个子系统中的页集。 要执行其中任一操作，必须从一个子系统中卸载所有对象和消息，然后将它们重新装入到另一个子系统中。

第 741 页的表 72 提供了一些示例，其中显示了由 第 741 页的图 102 中的语句定义的子系统名称和命令前缀字符串 (CPF) 的关联。

表 *72:* 与 *CPF* 关联的子系统名称

IBM MQ 子系统名称	**CPF**
CSQ1	+mqs1cpf
CSQ2	+mqs2cpf
CSQ3	++

注: z/OS 命令 SETSSI 的 ACTIVATE 和 DEACTIVATE 功能不受 IBM MQ 支持。

要检查更改的状态，请在 SDSF 中发出以下命令: `/D SSI,L`。 您将看到以 ACTIVE 状态创建的新子系统。

z/OS 定义命令前缀字符串 *(CPF)*

IBM MQ 的每个子系统实例都可以具有命令前缀字符串来标识该子系统。

对所有子系统采用系统范围的 CPF 约定，以避免冲突。 遵循以下准则:

- 将 CPF 定义为最多 8 个字符的字符串。

- 请勿使用任何其他子系统已在使用的 CPF，并避免使用系统上定义的 JES 退格字符作为字符串的第一个字符。
- 使用 第 742 页的表 74 中列出的有效字符集中的字符来定义 CPF。
- 请勿使用已定义进程的缩写 CPF 或可能与命令语法混淆的 CPF。 例如， CPF (例如 "D") 与 z/OS 命令 (例如 DISPLAY) 冲突。 要避免发生这种情况，请使用其中一个特殊字符 (如 第 742 页的表 74 中所示) 作为 CPF 字符串中的第一个或唯一字符。
- 请勿定义作为现有 CPF 的子集或超集的 CPF。 要获取示例，请参阅第 742 页的表 73。

表 *73: CPF* 子集和超集规则的示例

子系统名称	**已定义的 CPF**	**路由到的命令**
MQA	!A	MQA
MQB	!B	MQB
MQC1	!C1	MQC1
MQC2	!C2	MQC2
MQB1	!B1	MQB

用于子系统 MQB1 的命令 (使用 CPF!B1) 将路由到子系统 MQB，因为此子系统的 CPF 为!B，子集!B1. 例如，如果输入了以下命令:

```
!B1  START  QMGR
```

子系统 MQB 接收命令:

```
1 START QMGR
```

(在这种情况下，它无法处理)。

您可以通过发出 z/OS 命令 DISPLAY OPDATA 来查看存在哪些前缀。

如果您正在综合系统中运行，那么 z/OS 会在 CPF 注册时诊断此类型的任何冲突 (请参阅 第 742 页的『综合系统环境中的 CPF』 以获取有关 CPF 注册的信息)。

第 742 页的表 74 显示了定义 CPF 字符串时可以使用的字符:

表 *74: CPF* 字符串的有效字符集

字符集	**内容**
字母顺序	大写 A 到 Z，小写 a 到 z
数字	0 到 9
本国 (见附注)	@ $# (可以表示为十六进制值的字符)
特殊	. □ ( ) * & + - = ¢ < | ! ; % _ ? : >

注:

系统可识别以下十六进制表示的国家字符：@ 表示 X'7C',$ 表示 X'5B',# 表示 X'7B'。 在 U.S 以外的国家或地区， U.S。在终端键盘上表示的国家字符可能会生成不同的十六进制表示，并导致错误。 例如，在某些国家，$ 字符可能会生成 X'4A'。

分号 (;) 作为 CPF 有效，但在大多数系统上，此字符是命令定界符。

z/OS 综合系统环境中的 *CPF*

使用本主题来了解如何在综合系统的作用域内使用 CPF。

如果在综合系统环境中使用，那么 IBM MQ 会注册 CPF 以使您能够从综合系统中的任何控制台输入命令，并将该命令路由到相应的系统以执行。 命令响应将返回到发端控制台。

定义综合系统操作的作用域

作用域用于确定在综合系统环境中运行 IBM MQ 时由 IBM MQ 子系统执行的 CPF 注册类型。

作用域的可能值如下所示:

M

系统作用域。

CPF 在系统 IPL 时由 IBM MQ 向 z/OS 注册，并且在 z/OS 系统处于活动状态的整个时间内保持注册状态。

必须在连接到运行目标子系统的 z/OS 映像的控制台上输入 IBM MQ 命令，或者必须使用 ROUTE 命令将该命令定向到该映像。

如果未在综合系统中运行，请使用此选项。

S

综合系统已启动作用域。

启动 IBM MQ 子系统时， CPF 将向 z/OS 注册，并保持活动状态，直到 IBM MQ 子系统终止为止。

必须使用 ROUTE 命令将原始 START QMGR 命令定向到目标系统，但可以在连接到综合系统的任何控制台上输入所有进一步的 IBM MQ 命令，并自动将这些命令路由到目标系统。

在 IBM MQ 终止后，必须使用 ROUTE 命令将后续 START 命令定向到目标 IBM MQ 子系统。

X

综合系统 IPL 作用域。

CPF 在系统 IPL 时由 IBM MQ 向 z/OS 注册，并且在 z/OS 系统处于活动状态的整个时间内保持注册状态。

可以在连接到综合系统的任何控制台上输入 IBM MQ 命令，并将这些命令路由到正在自动执行目标系统的映像。

可以在综合系统中的一个或多个 z/OS 映像上定义具有作用域为 S 的 CPF 的 IBM MQ 子系统，因此这些映像可以共享单个子系统名称表。 但是，必须确保在要运行 IBM MQ 子系统的 z/OS 映像上发出 (或路由至) 初始 START 命令。 如果使用此选项，那么可以停止 IBM MQ 子系统并在综合系统中的其他 z/OS 映像上重新启动该子系统，而不必更改子系统名称表或对 z/OS 系统执行 IPL。

具有作用域为 X 的 CPF 的 IBM MQ 子系统只能在综合系统中的一个 z/OS 映像上定义。 如果使用此选项，那么必须为每个需要具有作用域 X 的 CPF 的 z/OS 映像 IBM MQ 子系统定义唯一的子系统名称表。

如果要使用 z/OS 自动重新启动管理器 (ARM) 来自动重新启动不同 z/OS 映像中的队列管理器，那么必须在可能重新启动该队列管理器的每个 z/OS 映像中定义每个队列管理器。 必须使用具有 CPF 作用域 S 的系统范围的唯一 4 字符子系统名称来定义每个队列管理器。

z/OS 为 *IBM MQ* 队列管理器创建过程

每个 IBM MQ 子系统都需要一个编目过程来启动队列管理器。 您可以创建自己的或使用 IBM 提供的过程库。

- 对每个 IBM MQ 队列管理器重复此任务。
- 从先前版本进行迁移时，可能需要修改编目过程。

对于子系统名称表中定义的每个 IBM MQ 子系统，请在过程库中创建用于启动队列管理器的编目过程。 IBM 提供的过程库称为 SYS1.PROCLIB，但您的安装可能使用其自己的命名约定。

队列管理器启动任务过程的名称是通过将子系统名称与字符 MSTR 并置而形成的。 例如，子系统 CSQ1 具有过程名称 CSQ1MSTR。 您定义的每个子系统都需要一个过程。

您需要包含包含所选语言的消息的库:

- thlqual.SCSQSNLE，用于美国英语混合大小写
- thlqual.SCSQSNLU，用于美国英语大写
- thlqual.SCSQSNLK，针对日语
- thlqual.SCSQSNLF，用于法语
- thlqual.SCSQSNLC，用于中文

本产品文档中的许多示例和指示信息假定您有一个名为 CSQ1 的子系统。 如果最初为安装验证和测试目的创建了名为 CSQ1 的子系统，那么您可能会发现这些示例更易于使用。

thlqual.SCSQPROC 中提供了两个样本启动式任务过程。 成员 CSQ4MSTR 对每个消息类使用一个页集，成员 CSQ4MSRR 对主要消息类使用多个页集。 将这些过程之一复制到 SYS1.PROCLIB 的成员 xxxxMSTR (其中 xxxx 是 IBM MQ 子系统的名称) ，或者如果您未使用 SYS1.PROCLIB，那么复制到过程库。 将样本过程复制到您定义的每个 IBM MQ 子系统的过程库中的成员。

复制成员后，可以使用成员中的指示信息根据每个子系统的需求对其进行定制。 有关指定队列管理器使用的存储器限制的信息，请参阅 建议的区域大小。 您还可以在 JCL 中使用符号参数，以允许在过程启动时对其进行修改。 如果您有多个 IBM MQ 子系统，那么可能会发现将 JCL 包含组用于过程的公共部分以简化将来的维护是有利的。

如果您正在使用队列共享组，那么 STEPLIB 并置必须包含 Db2 运行时目标库 SDSNLOAD ，并且必须经过 APF 授权。 仅当无法通过链接列表或 LPA 访问此库时，才需要在 STEPLIB 并置中使用此库。

注意:

1. 您可以记录引导数据集 (BSDS) ，日志和页面集的名称，以便在 JCL 中使用，然后在该过程的后续步骤中定义这些集。
2. 样本启动式任务过程 CSQ4MSTR 和 CSQ4MSRR 已更新为包含可用于定义包含传输安全性 (即 SSL 或 TLS 属性) 的 QMINI 数据集的 CSQMINI DD 卡，但保留注释掉。

   您可以使用 第 749 页的『QMINI 数据集』 来启用或禁用 TLS 1.3 支持和/或用于定义要由通道使用的 CipherSpecs 的定制列表。

相关概念

第 744 页的『为通道启动程序创建过程』
对于每个 IBM MQ 子系统，定制 CSQ4CHIN 的副本。 根据您正在使用的其他产品，您可能需要允许访问其他数据集。

z/OS 为通道启动程序创建过程

对于每个 IBM MQ 子系统，定制 CSQ4CHIN 的副本。 根据您正在使用的其他产品，您可能需要允许访问其他数据集。

- 对每个 IBM MQ 队列管理器重复此任务。
- 从先前版本进行迁移时，可能需要修改编目过程。

您需要为将要使用分布式排队的每个 IBM MQ 子系统创建一个通道启动程序启动式任务过程。

要执行此操作:

1. 将样本启动式任务过程 thlqual.SCSQPROC(CSQ4CHIN) 复制到过程库。 将过程命名为 *xxxx* CHIN ，其中 *xxxx* 是 IBM MQ 子系统的名称 (例如， CSQ1CHIN 将是队列管理器 CSQ1 的通道启动程序启动式任务过程)。
2. 为要使用的每个 IBM MQ 子系统创建一个副本。
3. 使用样本过程 CSQ4CHIN 中的指示信息，根据您的需求定制过程。 您还可以在 JCL 中使用符号参数，以允许在过程启动时对其进行修改。 管理 IBM MQ for z/OS 中的启动选项对此进行了描述。

   并置分布式排队库 thlqual.SCSQMVR1。

   需要访问 LE 运行时库 SCEERUN; 如果它不在链接列表 (SYS1.PARMLIB(LNKLSTkk) ，将其并置在 STEPLIB DD 语句中。

4. 授权过程在外部安全管理器下运行。
5. 您需要包含包含所选语言的消息的库:
   - thlqual.SCSQSNLE，用于美国英语混合大小写
   - thlqual.SCSQSNLU，用于美国英语大写
   - thlqual.SCSQSNLK，针对日语
   - thlqual.SCSQSNLF，用于法语
   - thlqual.SCSQSNLC，用于中文

通道启动程序是一个长时间运行的地址空间。 要防止在使用受限 CPU 量后终止该 CPU ， 请确认以下任一操作:

- z/OS 系统中已启动任务的缺省值为无限制 CPU; 具有 TIME = (1440,00) 的 JOBCLASS (STC) 的 JES2 配置语句可实现此目的，或者
- 将 TIME=1440 或 TIME=NOLIMIT 参数显式添加到 CSQXJST 的 EXEC 语句。

如果要使用通道出口，可以稍后将出口库 (CSQXLIB) 添加到此过程。 您需要停止并重新启动通道启动程序才能执行此操作。

如果您正在使用 TLS ，那么需要访问系统 TLS 运行时库。 此库称为 SIEALNKE。 库必须获得 APF 授权。

如果您正在使用 TCP/IP ，那么通道启动程序地址空间必须能够访问 TCPIP.DATA 数据集。 必须设置数据集的方式取决于您正在使用的 TCP/IP 产品和接口。 其中包括:

- 环境变量， RESOLVER_CONFIG
- 文件系统上的 /etc/resolv.conf
- // SYSTCPD DD 语句
- // SYSTCPDD DD 语句
- *jobname/userid*.TCPIP.DATA
- SYS1.TCPPARMS(TCPDATA)
- *zapname*.TCPIP.DATA

其中一些操作会影响启动式任务过程 JCL。 有关更多信息，请参阅 z/OS Communications Server: IP Configuration Guide。

相关概念

第 745 页的『将 IBM MQ 子系统定义到 z/OS WLM 服务类』
要在 z/OS 系统中提供 IBM MQ 相应的性能优先级，必须将队列管理器和通道启动程序地址空间分配给相应的 z/OS 工作负载管理 (WLM) 服务类。 如果未显式执行此操作，那么可能会应用不适当的缺省值。

z/OS 将 *IBM MQ* 子系统定义到 *z/OS WLM* 服务类

要在 z/OS 系统中提供 IBM MQ 相应的性能优先级，必须将队列管理器和通道启动程序地址空间分配给相应的 z/OS 工作负载管理 (WLM) 服务类。 如果未显式执行此操作，那么可能会应用不适当的缺省值。

- 对每个 *IBM MQ* 队列管理器重复此任务。
- 从先前版本迁移时，不需要执行此任务。

使用随 WLM 提供的 ISPF 对话框执行以下任务:

- 从 WLM 耦合数据集中抽取 z/OS WLM 策略定义。
- 通过向所选服务类添加队列管理器和通道启动程序启动式任务过程名称来更新此策略定义
- 在 WLM 耦合数据集上安装已更改的策略

然后使用 z/OS 命令激活此策略

```
V WLM,POLICY=policyname,REFRESH
```

有关设置性能选项的更多信息，请参阅 在 z/OS 上规划 IBM MQ 环境 。

相关概念

第 777 页的『设置 Db2 环境』
如果您正在使用队列共享组，那么必须通过定制和运行多个样本作业来创建所需的 Db2 对象。

z/OS 实施 *ESM* 安全控制

对队列管理器和通道启动程序实施安全控制。

- 对每个 *IBM MQ* 队列管理器重复此任务。
- 从先前版本进行迁移时，可能需要执行此任务。

如果使用 RACF 作为外部安全性管理器，请参阅 在 z/OS 上设置安全性，其中描述了如何实施这些安全性控制。

如果您正在使用通道启动程序，那么还必须执行以下操作:

- 如果子系统具有活动的连接安全性，请将连接安全性概要文件 ssid.CHIN 定义到外部安全性管理器 (请参阅 通道启动程序的连接安全性概要文件 以获取有关此信息)。
- 如果您正在使用传输层安全性 (TLS) 或套接字接口，请确保将运行通道启动程序的权限所使用的用户标识配置为使用 z/OS UNIX System Services，如 *z/OS UNIX System Services Planning* 文档中所述。
- 如果您正在使用 TLS，请确保配置运行通道启动程序的权限所使用的用户标识以访问 ALTER QMGR 命令的 SSLKEYR 参数中指定的密钥环。

在启动队列管理器之前，通过以下方式设置 IBM MQ 数据集和系统安全性:

- 授权队列管理器启动式任务过程在外部安全管理器下运行。
- 授权访问队列管理器数据集。
- V 9.2.0 配置 z/OS 数据集加密 (如果需要)。

  请参阅 使用数据集加密在 IBM MQ for z/OS 上静态数据的机密性 部分。 for more information.

有关如何执行此操作的详细信息，请参阅 z/OS 的安全性安装任务。

如果您正在使用 RACF，那么只要使用 RACF STARTED 类，就不需要对系统执行 IPL (请参阅 RACF 授权启动式任务过程)。

相关概念

第 746 页的『更新 SYS1.PARMLIB 成员』
要确保您的更改在 IPL 后仍然有效，必须更新 SYS1.PARMLIB

第 780 页的『为队列共享组实施 ESM 安全控制』
对队列共享组中的所有队列管理器实施安全性控制，以访问 Db2 和耦合设施列表结构。

z/OS 更新 *SYS1.PARMLIB* 成员

要确保您的更改在 IPL 后仍然有效，必须更新 SYS1.PARMLIB

- 您需要对要运行 *IBM MQ* 的每个 *z/OS* 系统执行一次此任务。
- 如果您正在使用队列共享组，那么必须确保 *IBM MQ* 的设置在综合系统中的每个 *z/OS* 系统上都相同。
- 从先前版本进行迁移时，可能需要执行此任务。

更新 SYS1.PARMLIB 成员如下所示:

1. 更新成员 IEFSSNss，如 第 740 页的『将 IBM MQ 子系统定义到 z/OS』中所述。
2. 更改 IEASYSpp，以便在执行 IPL 时使用以下成员:
   - 第 736 页的『APF 授权 IBM MQ 装入库』 中使用的 PROGxx 或 IEAAPFaa 成员
   - 第 736 页的『更新 z/OS 链接列表和 LPA』 中使用的 LNKLSTkk 和 LPALSTmm 成员
   - 第 740 页的『更新 z/OS 程序属性表』 中使用的 SCHEDxx 成员
   - 第 740 页的『将 IBM MQ 子系统定义到 z/OS』 中使用的 IEFSSNss 成员

相关概念

第 747 页的『定制初始化输入数据集』

制作样本初始化输入数据集的工作副本，并根据您的系统需求进行定制。

z/OS 定制初始化输入数据集

制作样本初始化输入数据集的工作副本，并根据您的系统需求进行定制。

- 对每个 *IBM MQ* 队列管理器重复此任务。
- 从先前版本迁移时，需要执行此任务。

每个 IBM MQ 队列管理器从 IBM MQ 初始化输入数据集中包含的一系列命令获取其初始定义。 这些数据集由队列管理器启动式任务过程中定义的 DD 名称 CSQINP1， CSQINP2 和 CSQINPT 引用。

对这些命令的响应将写入 DD 名称 CSQOUT1， CSQOUT2 和 CSQOUTT 引用的初始化输出数据集。

要保留原件，请制作每个样本的工作副本。 然后，您可以定制这些工作副本中的命令以满足您的系统需求。

如果使用多个 IBM MQ 子系统，那么如果在初始化输入数据集名称的高级限定符中包含子系统名称，那么可以更轻松地标识与每个数据集关联的 IBM MQ 子系统。

请参阅以下主题以获取有关样本的更多信息:

- 初始化数据集格式
- 使用 CSQINP1 样本
- 使用 CSQINP2 样本
- 使用 CSQINPX 样本
- 使用 CSQINPT 样本

初始化数据集格式

初始化输入数据集可以是分区数据集 (PDS) 成员或连续数据集。 它们可以是一系列并置的数据集。 定义它们的记录长度为 80 字节，其中:

- 只有列 1 到 72 才有意义。 将忽略第 73 至 80 列。
- 列 1 中带有星号 (*) 的记录将被解释为注释并被忽略。
- 将忽略空白记录。
- 每个命令都必须在新记录上启动。
- 尾部平均值从下一个记录的列 1 继续。
- 尾部 + 表示从下一个记录的第一个非空白列继续。
- 命令中允许的最大字符数为 32 762。

初始化输出数据集是连续数据集，记录长度为 125 ，记录格式为 VBA ，块大小为 629。

使用 CSQINP1 样本

数据集 thlqual.SCSQPROC 包含两个成员，这些成员包含缓冲池的定义，页集与缓冲池关联以及 ALTER SECURITY 命令。

成员 CSQ4INP1 对每个消息类使用一个页集。 消息分为以下类:

- 与系统相关的消息。
- 重要的长期消息。
- 短命消息。
- 其他消息。

成员 CSQ4INPR 对每个主要消息类使用多个页集，对每个其他类使用一个页集。 以下是主要的消息类:

- 重要的长期消息。

- 短命消息。

在队列管理器启动式任务过程的 CSQINP1 并置中包含相应的样本。

注意:

1. IBM MQ 最多支持 100 个范围在 0 到 99 之间的缓冲池。 只能从 CSQINP1 初始化数据集发出 DEFINE BUFFPOOL 命令。 样本中的定义指定四个缓冲池。
2. 必须使用 DEFINE PSID 命令在 CSQINP1 初始化数据集中定义队列管理器使用的每个页集。 页集定义将缓冲池标识与页集相关联。 如果未指定缓冲池，那么缺省情况下将使用缓冲池零。

   必须定义页集零 (00)。 它包含所有对象定义。 您最多可以为每个队列管理器定义 100 个页面集。
3. ALTER SECURITY 命令可用于变更安全性属性 TIMEOUT 和 INTERVAL。 在 CSQ4INP1 中，缺省值定义为 54 (对于 TIMEOUT) 和 12 (对于 INTERVAL)。

有关组织缓冲池和页集的信息，请参阅 在 z/OS 上规划 。

如果在队列管理器运行时动态更改缓冲池和页集定义，那么还应更新 CSQINP1 定义。 除非缓冲池定义包含 REPLACE 属性，否则仅对 IBM MQ 的冷启动保留这些更改。

使用 CSQINP2 样本

此表列出了可以包含在队列管理器启动式任务过程的 CSQINP2 并置中的 thlqual.SCSQPROC 成员及其功能的描述。 命名约定为 CSQ4IN*。 应针对您的配置修改 CSQ4INY* 成员。 您应该避免更改 CSQINS* 成员，因为在迁移到下一个发行版时将需要重新应用任何更改。 而是可以将 DEFINE 或 ALTER 命令放在 CSQ4INY* 成员中。

表 *75: thlqual.SCSQPROC* 的成员

成员名	描述
CSQ4INSG	系统对象定义。
CSQ4INSA	用于通道认证的系统对象和缺省规则。
CSQ4INSX	系统对象定义。
CSQ4INSS	如果您正在使用队列共享组，请定制并包含此成员。
CSQ4INSJ	如果您使用的是使用 JMS 的发布/预订，请定制并包含此成员。
CSQ4INSM	Advanced Message Security 的系统对象定义。
CSQ4INSR	如果您正在使用 WebSphere Application Server 或 IBM MQ V7 或更高版本中的已排队发布/预订守护程序支持的已排队发布/预订接口，请定制并包含此成员。
CSQ4DISP	CSQINP2 样本，用于显示对象定义。
CSQ4INYC	集群定义。
CSQ4INYD	分布式排队定义。
CSQ4INYG	一般定义。
CSQ4INYR	存储类定义，将多个页集用于主要的消息类。
CSQ4INYS	存储类定义，对每个消息类使用一个页集。

您只需要定义一次对象，而不是每次启动队列管理器时，因此不必每次都在 CSQINP2 中包含这些定义。 如果每次都包含这些对象，那么您将尝试定义已存在的对象，并且您将收到类似于以下内容的消息:

```
CSQM095I +CSQ1 CSQMAQLC QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) ALREADY EXISTS
CSQM090E +CSQ1 CSQMAQLC FAILURE REASON CODE X'00D44003'
CSQ9023E +CSQ1 CSQMAQLC ' DEFINE QLOCAL' ABNORMAL COMPLETION
```

此故障不会损坏对象。 如果要保留 CSQINP2 并置中的 SYSTEM 定义数据集，那么可以通过对每个对象指定 REPLACE 属性来避免失败消息。

使用 CSQINPX 样本

样本 thlqual.SCSQPROC(CSQ4INPX) 包含您每次启动通道启动程序时可能要执行的一组命令。 这些命令通常是与通道相关的命令 (例如 START LISTENER)，每次启动通道启动程序时 (而不是每次启动队列管理器时) 都需要这些命令，并且在输入数据集 CSQINP1 或 CSQINP2 中不允许这些命令。 您必须先定制此样本，然后才能将其包含在通道启动程序的 CSQINPX 数据集中。

在通道启动程序初始化结束时，将执行数据集中包含的 IBM MQ 命令，并将输出写入 CSQOUTX DD 语句指定的数据集。 输出类似于 IBM MQ 实用程序 (CSQUTIL) 的 COMMAND 函数生成的输出。 请参阅 CSQUTIL 实用程序 以获取更多详细信息。

您可以指定可从 CSQUTIL 发出的任何 IBM MQ 命令，而不仅仅是通道命令。 您可以在处理 CSQINPX 时从其他源输入命令。 无论先前命令是否成功，都将按顺序发出所有命令。

要指定命令响应时间，可以使用伪命令 COMMAND 作为数据集中的第一个命令。 这将采用单个可选关键字 RESPTIME (*nnn*)，其中 *nnn* 是等待对每个命令的响应的时间 (以秒计)。 此范围在 5 到 999 之间; 缺省值为 30。

如果 IBM MQ 检测到对四个命令的响应所用时间过长，那么将停止处理 CSQINPX，并且不会发出更多命令。 通道启动程序未停止，但会将消息 CSQU052E 写入 CSQOUTX 数据集，并将消息 CSQU013E 发送到控制台。

当 IBM MQ 成功完成 CSQINPX 的处理时，会将消息 CSQU012I 发送到控制台。

使用 CSQINPT 样本

此表列出了可以包含在队列管理器启动式任务过程的 CSQINPT 并置中的 thlqual.SCSQPROC 成员及其功能的描述。

表 *76: thlqual.SCSQPROC* 的成员

成员名	描述
CSQ4INST	系统缺省预订定义。
CSQ4INYT	发布/预订定义。

当发布/预订初始化完成时，将执行数据集中包含的 IBM MQ 命令，并将输出写入 CSQOUTT DD 语句指定的数据集。 输出类似于 IBM MQ 实用程序 (CSQUTIL) 的 COMMAND 函数生成的输出。 请参阅 CSQUTIL 实用程序 以获取更多详细信息。

相关概念

第 751 页的『创建引导程序和日志数据集』
使用提供的程序 CSQJU003 来准备引导数据集 (BSD) 和日志数据集。

z/OS V 9.2.0 V 9.2.0 *QMINI* 数据集

您可以使用 QMINI 数据集来指定在队列管理器初始化期间要读取和处理的属性。

QMINI 数据集的特征

QMINI 数据集是连续数据集或分区数据集的成员，最大记录长度为 80 字节 (72 字节表示数据， 8 字节表示行号)。

以下示例显示了连续 QMINI 数据集的属性。 当然，某些属性是基于您的环境。

```
Data Set Name . . . . : QM01.QMINI
General Data                          Current Allocation
Management class . .  : STANDARD      Allocated tracks  . : 1
Storage class  . . .  : STANDARD      Allocated extents . : 1
Volume serial . . .   : P5P21E
Device type . . . .   : 3390
Data class . . . . .  : **None**
Organization  . . .   : PS            Current Utilization
Record format . . .   : FB            Used tracks . . . . : 0
Record length . . .   : 80            Used extents  . . . : 0
```

```
Block size  . . . .   : 3120
1st extent tracks .   : 1
Secondary tracks  .   : 1          Dates
Data set name type    :            Creation date . . . : 2020/08/11
Data set encryption   : NO         Referenced date . . : ***None***
                                   Expiration date . . : ***None***
SMS Compressible  .   : NO
```

thlqual.SCSQPROC，包括：

- CSQ4QMIN 中 QMINI 数据集的样本内容。
- 在队列管理器启动 JCL 中的启动任务过程 CSQ4MSTR 和 CSQ4MSRR 中使用 // CSQMINI DD 卡指定 QMINI 数据集的示例。

注意：

- 解析数据集的代码仅解析每条记录的前 72 个字节。
- 将忽略行号，因此不需要指定行号。
- 如果行以星号字符 (*) 开头，那么该行将被视为注释。
- 在队列管理器启动期间解析 QMINI 数据集的内容。 如果成功解析内容，那么将在队列管理器作业日志中发出消息 CSQM578I 。 如果在解析期间迁到任何错误，那么会在队列管理器作业日志中发出错误消息，例如 CSQM573E，但队列管理器仍会启动。

  检查错误消息，并解决 QMINI 数据集内容中的任何问题。

  如果队列管理器无法解析 QMINI 数据集，那么可以启动通道启动程序，但无法启动配置为使用 SSL 或 TLS 的任何通道，因为安全配置设置未知。
- 如果在启动队列管理器后对数据集进行任何更新，那么必须重新启动队列管理器以获取更改。

TransportSecurity 节

从 IBM MQ for z/OS 9.2.0 开始， QMINI 数据集支持 TransportSecurity 节。 此节提供的函数与 IBM MQ for Multiplatforms 上 qm.ini 文件中的 SSL 节提供的函数相似。

TransportSecurity 节支持以下属性：

AllowTLSV13
队列管理器是否能够使用 TLS 1.3 CipherSpecs; 有效值为: *TRUE/T/YES/Y* 或 *FALSE/F/NO/N*。

对于迁移的队列管理器，缺省情况下未启用 TLS 1.3 。 您可以通过使用 TransportSecurity 节和 **AllowTLSV13**=TRUE 定义 QMINI 数据集来启用 TLS 1.3 。

对于在 z/OS 2.4 或更高版本上运行的新创建的队列管理器，缺省情况下将启用 TLS 1.3 。

AllowedCipher 规范
已启用的 CipherSpecs 的定制列表。

请参阅 在 IBM MQ for z/OS 上提供已订购和已启用的 CipherSpecs 的定制列表，以获取有关此属性的更多信息。

将忽略列表中重复的 CipherSpec 名称。

V 9.2.1 **OutboundSNI**
在启动 TLS 连接时，是将服务器名称指示 (SNI) 设置为远程系统的目标 IBM MQ 通道名称，还是设置为主机名; 有效值为: CHANNEL 或 HOSTNAME。

如果在通道对象 CERTLABL 字段上使用证书标签配置了目标通道，那么必须将 CERTLABL 设置为通道值。 如果使用 HOSTNAME 设置连接到具有 CERTLABL 设置的通道，那么连接将失败，并在远程队列管理器错误日志中显示 AMQ9673 消息。

以下示例显示如何指定 TransportSecurity 节：

```
TransportSecurity:
AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
                   ECDHE_RSA_AES_256_GCM_SHA384
AllowTLSV13=TRUE
```

z/OS 创建引导程序和日志数据集

使用提供的程序 CSQJU003 来准备引导数据集 (BSD) 和日志数据集。

- 对每个 *IBM MQ* 队列管理器重复此任务。
- LTS 如果要使用 *z/OS* 数据集加密来保护 *BSDS* 或活动日志数据集，那么必须在此步骤中分配数据集之前配置此选项。
- 从先前版本迁移时，不需要执行此任务。

  LTS 如果要迁移队列管理器并为活动日志数据集或 *BSDS* 添加 *z/OS* 数据集加密，那么需要转换数据集。

  请参阅 使用数据集加密的 IBM MQ for z/OS 上静态数据的机密性。 有关配置 *z/OS* 数据集加密和转换要加密的现有 *IBM MQ* 数据集的更多信息。

用于运行 CSQJU003 以创建单日志记录环境或双日志记录环境的样本 JCL 和访问方法服务 (AMS) 控制语句保存在 thlqual.SCSQPROC(CSQ4BSDS) 中。 定制并运行此作业以创建 BSD 和日志，并预先格式化日志。

要点: 您应该使用最新版本的 CSQ4BSDS，或者手动更新 JCL 以使用 RECORDS (850 60)。

第 743 页的『为 IBM MQ 队列管理器创建过程』中描述的启动式任务过程 CSQ4MSTR 在以下格式的语句中引用了 BSD:

```
//BSDS1   DD DSN=++HLQ++.BSDS01,DISP=SHR
//BSDS2   DD DSN=++HLQ++.BSDS02,DISP=SHR
```

日志数据集由 BSD 引用。

注:

1. 必须在 LOGDEF 步骤中的 SYSPRINT DD 语句上指定 BLKSIZE。 BLKSIZE 必须为 629。
2. 为了帮助识别来自不同队列管理器的引导数据集和日志数据集，请在这些数据集的高级限定符中包含子系统名称。
3. 如果您正在使用队列共享组，那么必须使用 SHAREOPTIONS (2 3) 定义引导程序和日志数据集。

请参阅 在 z/OS 上规划，以获取有关规划引导程序和日志数据集及其大小的信息。

从 IBM MQ 8.0 开始， 8 字节日志 RBA 增强功能可提高队列管理器的可用性，如 更大的日志相对字节地址中所述。 要在队列管理器首次启动之前在队列管理器上启用 8 字节日志 RBA，请在创建日志记录环境后执行以下步骤。

注: V 9.2.5 对于在 IBM MQ 9.2.5 或更高版本 8 字节日志 RBA 上创建的队列管理器，已启用，因此不需要执行以下步骤。

1. 使用 **IDCAMS ALTER**，将 V 1 format BSDS (使用 CSQJU003 程序创建) 重命名为类似于 ++HLQ++.V1.BSDS01 的内容。

   注: 确保重命名数据和索引组件以及 VSAM 集群。
2. 使用与已定义的属性相同的属性分配新的 BSD。 这些将成为队列管理器在启动时将使用的版本 2 格式的 BSD。
3. 运行 BSDS 转换实用程序 (CSQJUCNV) 以将版本 1 格式 BSD 转换为新版本 2 格式 BSD。
4. 成功完成转换后，请删除版本 1 格式的 BSD。

注: 如果队列管理器位于队列共享组中，那么必须按如下所示启动队列共享组中的所有队列管理器，然后才能启用 8 字节日志 RBA:

- 如果队列管理器位于 IBM MQ 9.0.0 LTS，那么它必须已通过 **OPMODE**(NEWFUNC,900) 或 **OPMODE**(*NEWFUNC,800*) 启动
- 如果队列管理器位于 IBM MQ 9.0.n CD， 或 IBM MQ 9.1.0 LTS 或更高版本，需要已在该级别启动

相关概念

第 752 页的『定义页面集』
使用提供的其中一个样本为每个队列管理器定义页集。

z/OS 定义页面集

使用提供的其中一个样本为每个队列管理器定义页集。

- 对每个 *IBM MQ* 队列管理器重复此任务。

  V 9.2.0 如果使用 *z/OS* 数据集加密来保护页集，那么必须先配置此选项，然后才能在此步骤中分配数据集。
- 从先前版本迁移时，不需要执行此任务。

  V 9.2.0 如果要迁移队列管理器并为页集添加 *z/OS* 数据集加密，那么需要转换页集。

  V 9.2.0 请参阅 使用数据集加密的 IBM MQ for z/OS 上静态数据的机密性。 有关配置 *z/OS* 数据集加密和转换要加密的现有 *IBM MQ* 数据集的更多信息。

为每个 IBM MQ 队列管理器定义单独的页集。 thlqual.SCSQPROC(CSQ4PAGE) 和 thlqual.SCSQPROC(CSQ4PAGR) 包含用于定义和格式化页集的 JCL 和 z/OS 访问方法服务 (AMS) 控制语句。 成员 CSQ4PAGE 对每个消息类使用一个页集，成员 CSQ4PAGR 对主要消息类使用多个页集。 JCL 运行提供的实用程序 CSQUTIL。 查看样本并根据您想要的页集数量和要使用的大小对其进行定制。 请参阅 规划页集和缓冲池，以获取有关页集以及如何计算合适大小的信息。

第 743 页的『为 IBM MQ 队列管理器创建过程』 中描述的启动式任务过程 CSQ4MSTR 在以下格式的语句中引用了页集:

```
//CSQP00nn DD DISP=OLD,DSN=xxxxxxxxxx
```

其中， *nn* 是 00 到 99 之间的页集编号， *xxxxxxxxxx* 是您定义的数据集。

注:

1. 如果打算使用动态页集扩展功能，请确保为每个页集定义了辅助扩展数据块。 thlqual.SCSQPROC(CSQ4PAGE) 显示如何执行此操作。
2. 为了帮助标识来自不同队列管理器的页集，请在与每个页集相关联的数据集的高级限定符中包含子系统名称。
3. 如果您打算允许 FORCE 选项与实用程序 CSQUTIL 的 FORMAT 函数配合使用，那么必须在 AMS DEFINE CLUSTER 语句上添加 REUSE 属性。

   有关 REUSE 的更多信息，请参阅 z/OS DEFINE CLUSTER 命令的 可选参数 部分。
4. 如果您的页集要大于 4 GB，那么必须使用存储管理系统 (SMS) EXTENDED ADDRESSABILITY 功能。

相关概念

第 780 页的『将 IBM MQ 条目添加到 Db2 表』
如果您正在使用队列共享组，请运行 CSQ5PQSG 实用程序以将队列共享组和队列管理器条目添加到 Db2 数据共享组中的 IBM MQ 表。

z/OS 定制系统参数模块

IBM MQ 系统参数模块控制 IBM MQ 在其操作中使用的日志记录，归档，跟踪和连接环境。 提供了缺省模块。 您应该创建自己的系统参数模块，因为某些参数 (例如数据集名称) 通常是特定于站点的。

- 根据需要对每个 *IBM MQ* 队列管理器重复此任务。
- 从先前版本进行迁移时，可能需要执行此任务。 有关详细信息，请参阅 在 z/OS 上迁移 IBM MQ。
- 要在现有队列管理器上启用 *Advanced Message Security for z/OS*，只需将 *SPLCAP* 设置为 *YES*，如 第 754 页的『使用 CSQ6SYSP』中所述。 如果是首次配置此队列管理器，请完成整个任务。

系统参数模块有四个宏，如下所示:

宏名称	用途
CSQ6SYSP	指定连接和跟踪参数，请参阅 第 754 页的『使用 CSQ6SYSP』
CSQ6LOGP	控制日志初始化，请参阅 第 762 页的『使用 CSQ6LOGP』
CSQ6ARVP	控制归档初始化，请参阅 第 766 页的『使用 CSQ6ARVP』
CSQ6USGP	控制使用情况记录，请参阅 第 771 页的『使用 CSQ6USGP』

IBM MQ 提供了缺省系统参数模块 CSQZPARM ，如果您发出 START QMGR 命令 (不带 PARM 参数) 来启动 IBM MQ 实例，那么将自动调用该模块。 CSQZPARM 位于 IBM MQ 随附的 APF 授权库 thlqual.SCSQAUTH 中。 当您启动 IBM MQ 时，这些参数的值将显示为一系列消息。

有关如何使用此命令的更多信息，请参阅 START QMGR 。

创建您自己的系统参数模块

如果 CSQZPARM 不包含所需的系统参数，那么您可以使用 thlqual.SCSQPROC(CSQ4ZPRM) 中提供的样本 JCL 来创建自己的系统参数模块。

要创建您自己的系统参数模块:

1. 生成 JCL 样本的工作副本。
2. 根据需要编辑副本中每个宏的参数。 如果从宏调用中除去任何参数，那么将在运行时自动选取缺省值。
3. 将占位符 ++NAME++ 替换为装入模块要采用的名称 (这可以是 CSQZPARM)。
4. 如果您的汇编程序不是高级汇编程序，请根据您的汇编程序需要更改 JCL。
5. 运行 JCL 以组合和链接编辑系统参数宏的定制版本以生成装入模块。 这是具有指定名称的新系统参数模块。
6. 将生成的装入模块放入 APF 授权的用户库中。
7. 向 APF 授权的用户库添加用户 READ 访问权。
8. 将此库包含在 IBM MQ 队列管理器启动式任务过程 STEPLIB 中。 此库名必须位于 STEPLIB 中的库 thlqual.SCSQAUTH 之前。
9. 启动队列管理器时调用新的系统参数模块。 例如，如果新模块名为 NEWMODS ，请发出以下命令:

   ```
   START QMGR PARM(NEWMODS)
   ```

10. 通过检查作业记录来确保命令成功完成。 日志中应该有类似如下的条目:

    ```
    CSQ9022I CDL1 CSQYASCP 'START QMGR' NORMAL COMPLETION
    ```

您还可以在队列管理器启动 JCL 中指定参数模块名称。 有关更多信息，请参阅 启动和停止队列管理器。

注: 如果选择对模块 CSQZPARM 进行命名，那么不需要在 START QMGR 命令上指定 PARM 参数。

微调系统参数模块

IBM MQ 还提供了一组由三个汇编程序源模块组成的模块，可用于对现有系统参数模块进行微调。 这些模块位于库 thlqual.SCSQASMS 中。 通常，您在测试环境中使用这些模块来更改系统参数宏中的缺省参数。 每个源模块调用不同的系统参数宏:

此汇编程序源模块 ...	调用此宏 ...
CSQFSYSP	CSQ6SYSP (连接和跟踪参数)
CSQJLOGP	CSQ6LOGP (日志初始化)

此汇编程序源模块 ...	调用此宏 ...
CSQJARVP	CSQ6ARVP (归档初始化)

以下是您使用这些模块的方式:

1. 在用户汇编程序库中制作每个汇编程序源模块的工作副本。
2. 根据需要添加或更改任何参数的值来编辑副本。
3. 组装任何已编辑模块的副本以在用户对象库中创建对象模块。
4. 链接编辑这些对象代码模块与现有系统参数模块，以生成作为新系统参数模块的装入模块。
5. 确保新的系统参数模块是用户授权库的成员。
6. 将此库包含在队列管理器启动式任务过程 STEPLIB 中。 此库必须位于 STEPLIB 中的库 thlqual.SCSQAUTH 之前。
7. 通过发出 START QMGR 命令，在 PARM 参数中指定新的模块名称来调用新的系统参数模块，就像以前一样。

在 SCSQPROC 的成员 CSQ4UZPR 中提供了样本 usermod ，用于演示如何在 SMP/E 控制下管理定制系统参数。

更改系统参数

您可以在队列管理器运行时更改某些系统参数; 请参阅 SET SYSTEM， SET LOG 和 SET ARCHIVE 命令。

将 SET 命令放入初始化输入数据集中，以便这些命令在每次启动队列管理器时生效。

相关概念

第 772 页的『定制通道启动程序参数』
使用 ALTER QMGR 来定制通道启动程序以满足您的需求。

z/OS 使用 *CSQ6SYSP*

使用本主题作为如何使用 CSQ6SYSP 设置系统参数的参考。

第 754 页的表 77 中显示了 CSQ6SYSP 的缺省参数以及是否可以使用 SET SYSTEM 命令更改每个参数。 如果要更改其中任何值，请参阅参数的详细描述。

表 *77: CSQ6SYSP* 参数的缺省值

参数	描述	缺省值	SET 命令
V 9.2.4 第 756 页的『[MQ 9.2.4 2021 年 11 月]ACCTIME』	每次收集记帐数据之间的时间 (以分钟和秒为单位)。	-1	✓
ACELIM	ACE 存储池的大小 (以 1 KB 块为单位)。	0 (无限制)	✓
CLCACHE	指定要使用的集群高速缓存的类型。	静态	-
CMDUSER	用于命令安全性检查的缺省用户标识。	CSQOPR	-
EXCLMSG	指定要从任何日志中排除的消息的列表。 此列表中的消息不会发送到 z/OS 控制台和硬拷贝日志。 因此，使用 EXCLMSG 参数从 CPU 角度排除消息比使用 第 776 页的『禁止参考消息』 中描述的方法更高效	( )	✓
EXITLIM	每次调用期间队列管理器出口可运行的时间 (以秒计)。	30	-

表 77: *CSQ6SYSP* 参数的缺省值 (继续)			
参数	**描述**	**缺省值**	**SET 命令**
EXITTCB	要用于运行队列管理器出口的已启动服务器任务数。	8	-
LOGLOAD	IBM MQ 在一个检查点开始与下一个检查点之间写入的日志记录数。	500 000	✓
MULCCAPT	确定 "度量使用情况定价" 属性，该属性控制用于收集 "度量使用情况许可证收费" (MULC) 所使用的数据的算法。	请参阅 参数描述	-
OTMACON	OTMA 连接参数。	请参阅 参数描述	-
QINDXBLD	确定队列管理器重新启动是等待所有索引重建，还是在所有索引重建之前完成。	WAIT	-
QMCCSID	队列管理器的编码字符集标识。	零	-
QSGDATA	队列共享组参数。	请参阅 参数描述	-
RESAUDIT	RESLEVEL 审计参数。	YES	-
ROUTCDE	分配给未从特定控制台请求的消息的消息路由代码。	1	-
服务	保留供 IBM 使用。	0	✓
SMFACCT	指定在启动队列管理器时是否收集 SMF 记帐数据。 请注意，仅当启动通道启动程序时，才会收集类 4 通道记帐数据。	否	-
SMFSTAT	指定启动队列管理器时是否收集 SMF 统计信息。 请注意，仅当启动通道启动程序时，才会收集 4 类通道启动程序统计信息数据。	否	-
SPLCAP	指定是否在此队列管理器上启用队列安全策略功能。 对于 Advanced Message Security for z/OS，将此参数设置为 YES。	否	-
STATIME	V 9.2.4 从 IBM MQ for z/OS 9.2.4 开始，每次收集统计信息之间的时间 (以分钟和秒为单位)。 从 IBM MQ for z/OS 9.2.0 到 9.2.3 ，每次收集统计信息和记帐数据之间的时间 (以分钟为单位)。	30	✓
TRACSTR	指定是否自动启动跟踪。	否	-
TRACTBL	要由全局跟踪工具使用的跟踪表的大小 (以 4 KB 块为单位)。	99 (396 KB)	✓
WLMTIME	两次扫描 WLM 管理的队列的队列索引之间的时间。	30	-
WLMTIMU	WLMTIME 的单位 (分钟或秒)。	分钟	-

V 9.2.4 **ACCTIME**

指定连续收集记帐数据之间的时间间隔 (以分钟和秒为单位)。

指定一个数字，可以是 -1，也可以是 0 到 1440 分钟范围内的数字 (格式为 'mmmm')，也可以是 0 到 1440 分钟范围内的数字，以及 0 到 59 秒范围内的数字 (格式为 'mmmm.ss')。

注意:

- 仅指定时间间隔 (以秒为单位) 时，必须以值 0 作为时间间隔的前缀。 最小的可能时间间隔为 1 秒: "0.01"。
- 如果指定值 0，那么将在 SMF 全局记录时间间隔收集记帐数据。 有关更多信息，请参阅 使用系统管理设施 。
- 如果指定值 -1(缺省值)，那么将按 STATIME 值指定的时间间隔收集记帐数据。

例如:

'0.30' 设置 30 秒的时间间隔。
'5.30' 设置 5 分钟和 30 秒的时间间隔。
'30' 将时间间隔设置为 30 分钟。

ACELIM

指定 ACE 存储池的最大大小（以 1 KB 块为单位）。 该值必须是 0-999999 范围内的数字。 缺省值“0”表示不施加系统限制以外的约束。

您只应在已识别为使用过量 ECSA 存储空间的队列管理器上为 ACELIM 设置值。 限制 ACE 存储池对于限制系统中的连接数以及队列管理器使用的 ECSA 存储量有影响。

当队列管理器达到该限制后，应用程序将无法获取新连接。 缺少新连接将导致 MQCONN 处理失败，通过 RRS 协调的应用程序很可能在任何 IBM MQ API 中失败。

ACE 约占连接的线程相关控制块所需的 ECSA 总量的 12.5%。 因此，例如，如果指定 ACELIM=5120，那么期望将队列管理器 (对于与线程相关的控制块) 分配的 ECSA 总量上限为大约 40960K;，即 5120 乘以 8。

为了限制队列管理器分配的 ECSA 总量，对于 5120K 的线程相关控制块，需要 ACELIM 值 640。

您可以使用统计 CLASS(3) 跟踪生成的 SMF 115 子类型 5 记录来监视“ACE/PEB”存储池的大小，并为 ACELIM 设置相应的值。

通过统计 CLASS(2) 跟踪所写入的 SMF 115 子类型 7 记录，可以获取队列管理器针对控制块使用的 ECSA 存储总量。 使用的 ECSA 存储总量是 QSRSPHBGF 和 QSRSPHBGV 字段值之和。

有关 SMF 115 统计信息记录的更多信息，请参阅解释 IBM MQ 性能统计信息。

请注意，您应考虑将 ACELIM 设置为一种机制来防止 z/OS 映像成为行为异常的队列管理器，而不是将其用作一种控制应用程序与队列管理器的连接的方式。

高速缓存

指定要使用的集群高速缓存的类型。

集群高速缓存是用于存储与集群相关的信息的存储区域。

如果集群高速缓存是静态的，那么它具有在队列管理器启动时分配的固定大小。 如果高速缓存已满，那么将发出消息 CSQM060E，并且需要更多空间的应用程序请求将收到 MQRC_CLUSTER_RESOURCE_ERROR。

如果将 CLCACHE 设置为动态，那么可以根据需要扩展集群高速缓存。 但是，您首先需要确保任何已安装的集群工作负载出口都可以通过动态高速缓存运行。

如果已安装的集群工作负载出口无法通过动态高速缓存消息 CSQM061E 运行。

为集群工作负载出口提供了 MQXCLWLN，用于以工作方式浏览集群高速缓存，而无论使用的是动态高速缓存还是静态高速缓存。

对于新的队列管理器，请设置 CLCACHE=DYNAMIC，除非您将使用不支持动态高速缓存的集群工作负载出口。

对于已使用静态高速缓存且集群中未添加许多新队列和队列管理器的现有队列管理器，继续使用 CLCACHE=STATIC 是合理的。

对于已使用静态高速缓存且位于将向其添加许多新队列或队列管理器的集群中的现有队列管理器，请开始使用 CLCACHE=DYNAMIC。

静态
当集群高速缓存是静态的时，它的大小在队列管理器启动时是固定的，足够当前的集群信息量加上一些用于扩展的空间。 当队列管理器处于活动状态时，大小无法增加。 这是缺省值。

动态
如果集群高速缓存是动态的，那么在队列管理器处于活动状态时，如果需要，可以自动增大在队列管理器启动时分配的初始大小。

CMDUSER
指定用于命令安全性检查的缺省用户标识。 必须向 ESM 定义此用户标识 (例如， RACF)。 指定名称 1 到 8 个字母数字字符。 第一个字符必须是字母。

缺省值为 CSQOPR。

EXCLMSG
指定要排除的错误消息的列表。

此列表是动态的，并使用 SET SYSTEM 命令进行更新。

缺省值为空的 list ()。

提供的消息没有 CSQ 前缀，也没有操作码后缀 (I-D-E-A)。 例如，要排除消息 CSQX500I，请将 X500 添加到此列表中。 此列表最多可包含 16 个消息标识。

要有资格包含在列表中，必须在正常启动 MSTR 或 CHIN 地址空间后发出消息，并以以下字符之一开头: E， H， I， J， L， M， N， P， R， T， V， W， X， Y， 2， 3， 5， 9。

可以将由于处理命令而发出的消息标识添加到列表中，但是不会将其排除。 例如，由于 DISPLAY USAGE PSID (*) 命令而发出消息标识，但是不能禁止此消息。

EXITLIM
指定队列管理器出口的每次调用所允许的时间 (以秒计)。 (此参数对通道出口没有影响。)

指定 5 到 9999 范围内的值。

缺省值为 30。 队列管理器轮询每 30 秒运行一次的出口。 在每个轮询上，已运行超过 EXITLIM 指定的时间的任何轮询都将强制终止。

EXITTCB
指定要用于在队列管理器中运行出口的已启动服务器任务数。 (此参数对通道出口没有影响。) 必须至少指定一个与队列管理器可能必须运行的最大出口数 (通道出口除外) 一样高的数字，否则它将因 6c6 异常终止而失败。

指定范围在 0 到 99 之间的值。 值为零表示不能运行任何出口。

缺省值为 8。

LOGLOAD
指定 IBM MQ 在一个检查点的开始与下一个检查点之间写入的日志记录数。 在写入了您指定的记录数之后，IBM MQ 会开始一个新的检查点。

指定范围在 200 到 16000 000 之间的值。

缺省值为 50 万。

值越大， IBM MQ 的性能越好; 但是，如果参数设置为较大的值，那么重新启动将花费更长时间。

建议设置:

测试系统	10 000
生产系统	500 000 在生产系统中，提供的缺省值可能导致检查点频率过高。

LOGLOAD 的值确定队列管理器检查点的频率。 值过大意味着在检查点之间会将大量数据写入日志，从而导致队列管理器在失败后的正向恢复重新启动时间增加。 值过小会导致检查点在峰值负载期间发生频率过高，从而对响应时间和处理器使用情况产生负面影响。

对于 LOGLOAD ，建议初始值为 500 000。 对于每秒 100 条消息 (即， 100 条带有落实的 MQPUT 和 100 条带有落实的 MQGET) 的 1 KB 持久消息速率，检查点之间的时间间隔大约为 5 分钟。

注: 此参数仅用作准则，并且此参数的最佳值取决于各个系统的特征。

MULCCAPT
指定要用于收集测量的使用许可计费 (MULC) 所使用的数据的算法。

标准
MULC 基于从 IBM MQ API MQCONN 调用到 IBM MQ API MQDISC 调用的时间。

已优化
MULC 基于从 IBM MQ API 调用开始到 IBM MQ API 调用结束的时间。

缺省值为 STANDARD

OTMACON
OTMA 参数。 此关键字采用五个位置参数:

OTMACON = (Group,Member,Druexit,Age,Tpipepfx)

组
这是此特定 IBM MQ 实例所属的 XCF 组的名称。

它的长度可以是 1 到 8 个字符，并且必须以大写字符输入。

缺省值为空白，这指示 IBM MQ 不得尝试连接 XCF 组。

成员
这是 XCF 组中此特定 IBM MQ 实例的成员名。

它的长度可以是 1 到 16 个字符，并且必须以大写字符输入。

缺省值为 4 字符队列管理器名称。

Druexit
这指定要由 IMS 运行的 OTMA 目标解析用户出口的名称。

它的长度可以是 1 到 8 个字符。

缺省值为 DFSYDRU0。

此参数是可选的; 如果 IBM MQ 要从未由 IBM MQ 启动的 IMS 应用程序接收消息，那么此参数是必需的。 该名称必须与 IMS 系统中编码的目标解析用户出口相对应。 有关更多信息，请参阅第 832 页的『在 IMS 中使用 OTMA 出口』。

使用寿命
这表示 IMS 先前已验证来自 IBM MQ 的用户标识的时间长度 (以秒计)。

它可以在 0 到 2 147 483 647 的范围内。

缺省值为 2 147 483 647。

建议您将此参数与 ALTER SECURITY 命令的 `interval` 参数一起设置，以保持大型机中安全高速缓存设置的一致性。

特皮佩福克斯
这表示要用于 Tpipe 名称的前缀。

它包含三个字符; 第一个字符在 A 到 Z 范围内，后续字符为 A 到 Z 或 0 到 9。 缺省值为 CSQ。

每次 IBM MQ 创建 Tpipe 时都会使用此参数; 其余名称由 IBM MQ 指定。 不能为 IBM MQ 创建的任何 Tpipe 设置完整 Tpipe 名称。

QINDXBLD
确定队列管理器重新启动是等待所有队列索引重建，还是在所有索引重建之前完成。

WAIT
队列管理器重新启动将等待所有队列索引构建完成。 这意味着在创建索引时，不会在正常 IBM MQ API 处理期间延迟任何应用程序，因为会先创建所有索引，然后任何应用程序才能连接到队列管理器。

这是缺省值。

NOWAIT
队列管理器可以在完成所有队列索引构建之前重新启动。

QMCCSID
指定队列管理器 (因此是分布式排队) 要使用的缺省编码字符集标识。

指定范围在 0 到 65535 之间的值。 该值必须表示以 本地语言作为所选语言的本机 z/OS 代码页列出的 EBCDIC 代码页。

零 (缺省值) 表示使用当前设置的 CCSID，如果未设置，那么使用 CCSID 500。 这意味着如果您已显式将 CCSID 设置为任何非零值，那么不能通过将 QMCCSID 设置为零来将其复位; 现在必须使用正确的非零 CCSID。 如果 QMCCSID 为零，那么可以通过发出命令 DISPLAY QMGR CCSID 来检查实际使用的 CCSID。

注: 队列共享组中的所有队列管理器都应该使用相同的 QMCCSID。

队列数据
队列共享组数据。 此关键字采用五个位置参数:

QSGDATA = (Qsgname,Dsgname,Db2name,Db2serv ,Db2blob)

QSGName
这是队列管理器所属的队列共享组的名称。

请参阅 用于命名 IBM MQ 对象的规则 以获取有效字符。 名称:

- 长度可以为 1 到 4 个字符
- 不得以数字开头
- 不得以 @ 结尾。

这是因为，由于实施原因，在内部使用 @ 符号填充少于四个字符的名称，

缺省值为空白，这指示队列管理器不是任何队列共享组的成员。

兹尼亚梅
这是队列管理器要连接到的 Db2 数据共享组的名称。

它的长度可以是 1 到 8 个字符，并且必须以大写字符输入。

缺省值为空白，这指示您未使用队列共享组。

Db2name
这是队列管理器要连接的 Db2 子系统或组连接的名称。

它的长度可以是 1 到 4 个字符，并且必须以大写字符输入。

缺省值为空白，这指示您未使用队列共享组。

注: Db2 子系统 (或组连接) 必须位于 Dsgname 中指定的 Db2 数据共享组中，并且所有队列管理器都必须指定相同的 Db2 数据共享组。

Db2serv
这是用于访问 Db2 的服务器任务数。

它可以在 4 到 10 的范围内。

缺省值为 4。

Db2blob
这是用于访问二进制大对象 (BLOB) 的 Db2 任务数。

它可以在 4 到 10 的范围内。

缺省值为 4。

如果指定其中一个名称参数 (即， **Qsgname**， **Dsgname** 或 **Db2name**)，那么必须为其他名称输入值，否则 IBM MQ 将失败。

RESAUDIT
指定是否为在连接处理期间执行的 RESLEVEL 安全性检查写入 RACF 审计记录。

指定以下一项:

否
未执行 RESLEVEL 审计。

YES
执行 RESLEVEL 审计。

缺省值为 YES。

ROUTCDE
指定分配给未在 MQSC 命令的直接响应中发送的消息的缺省 z/OS 消息路由代码。

指定以下一项:

1. 1 到 16 (含) 范围内的值。
2. 值的列表，用逗号分隔并括在括号中。 每个值必须在范围 1 到 16 (含) 之间。

缺省值是 1。

有关 z/OS 路由代码的更多信息，请参阅 *z/OS MVS Routing and Descriptor Codes* 手册的其中一个卷中的消息描述 。

服务
此字段保留供 IBM 使用。

SMFACCT
指定当队列管理器启动时，IBM MQ 是否自动将记帐数据发送至 SMF。

指定以下一项:

否
请勿自动开始收集记帐数据。

YES
开始自动收集缺省类 1 的记帐数据。

整数
在 1 到 4 范围内自动收集其记帐的类的列表。

缺省值为 NO。

SMFSTAT
指定在队列管理器启动时是否自动收集 SMF 统计信息。

指定以下一项:

否
请勿自动开始收集统计信息。

YES
开始自动收集缺省类 1 的统计信息。

整数
在 1 到 4 范围内自动收集统计信息的类的列表。 要收集类 2 或 3 统计信息，还必须指定类 1。

缺省值为 NO。

SPLCAP
安全策略功能通过控制在从队列中写入和读取消息时是对消息进行签名还是加密的策略，支持更高级别的消息安全性。

通过将 SPLCAP 设置为下列其中一个值，为此队列管理器配置安全策略处理:

否
在队列管理器初始化期间，未启用为队列实现消息安全策略的功能。

YES
在队列管理器初始化期间启用消息安全性功能。

队列管理器会检查 AMSPROD 属性是否设置为 AMS， ADVANCED 或 ADVANCEDVUE 之一，在这种情况下，它已获得 AMS 的许可。 否则，它将不会启动。

队列管理器还会检查必需的 AMS 配置是否到位。 如果不是，那么队列管理器将不会启动。

如果队列管理器都获得了 AMS 的许可，并且已进行必要的配置，那么队列管理器将在队列管理器初始化期间启用消息安全性功能并启动 AMSM 地址空间时启动。

缺省值为 NO。

STATIME

V 9.2.4 从 IBM MQ for z/OS 9.2.4 开始，指定连续收集统计数据之间的时间 (以分钟和秒为单位)。 如果未设置 ACCTIME 或 ACCTIME 为 -1，那么还指定连续收集记帐数据之间的时间。 以 'mmmm'格式指定 0 到 1440 分钟范围内的数字，或者以 'mmmm.ss'格式指定 0 到 1440 分钟范围内的数字和 0 -59 秒范围内的数字。 缺省值是 30 分钟。

注意:

- 仅指定时间间隔 (以秒为单位) 时，必须以值 0 作为时间间隔的前缀。 可能的最小时间间隔为 1 秒: '0.01'。
- 如果指定值 0，那么将在 SMF 全局记录时间间隔收集记帐数据。 有关更多信息，请参阅 使用系统管理设施 。
- 如果指定值 -1(缺省值) ，那么将按 STATIME 值指定的时间间隔收集记帐数据。

从 IBM MQ for z/OS 9.2.4 开始，如果指定值 0，那么将在 SMF 数据收集广播中收集统计数据。 如果未指定 ACCTIME 或为 -1，那么还会在 SMF 数据收集广播中收集记帐数据。

LTS 连续收集统计信息之间的缺省时间 (以分钟为单位)。

从 IBM MQ for z/OS 9.2.0 到 9.2.3， 时间 (以分钟为单位)。 指定 0 到 1440 分钟范围内的数字，格式为 'mmmm'。 缺省值是 30 分钟。

如果您指定零值，那么将在 SMF 数据收集广播时同时收集统计数据和记帐数据。 请参阅 使用系统管理设施 以获取有关设置此设置的信息。

TRACSTR
指定是否自动启动全局跟踪。

指定以下一项:

否
请勿自动启动全局跟踪。

YES
自动对缺省类 1 启动全局跟踪。

整数
要在 1 到 4 范围内自动启动全局跟踪的类的列表。

*****
对所有类自动启动全局跟踪。

如果未在宏中指定关键字，那么缺省值为 NO。

注: 提供的缺省系统参数装入模块 (CSQZPARM) 具有 TRACSTR=YES (在汇编程序模块 CSQFSYSP 中设置)。 如果您不希望自动启动跟踪，请创建您自己的系统参数模块，或者在队列管理器启动后发出 STOP TRACE 命令。

有关 STOP TRACE 命令的详细信息，请参阅 停止跟踪。

TRACTBL
指定全局跟踪工具存储 IBM MQ 跟踪记录的跟踪表的缺省大小 (以 4 KB 块为单位)。

指定 1 到 999 范围内的值。

缺省值为 99。 这相当于 396 KB。

注: 跟踪表的存储器在 ECSA 中分配。 因此，您必须小心地选择此值。

WLMTIME
指定每次扫描 WLM 管理的队列的索引之间的时间 (以分钟或秒为单位，具体取决于 WLMTIMU 的值)。

指定 1 到 9999 范围内的值。

缺省值为 30。

WLMTIMU
与 WLMTIME 参数配合使用的时间单位。

指定下列其中一项:

分钟
WLMTIME 表示分钟数。

SECS
WLMTIME 表示秒数。

缺省值为 MINS。

相关参考
第 762 页的『使用 CSQ6LOGP』
使用本主题作为如何使用 CSQ6LOGP 指定日志记录选项的参考。

第 766 页的『使用 CSQ6ARVP』
使用本主题作为如何使用 CSQ6ARVP 指定归档环境的参考

z/OS 使用 *CSQ6LOGP*
使用本主题作为如何使用 CSQ6LOGP 指定日志记录选项的参考。

使用 CSQ6LOGP 来建立日志记录选项。

CSQ6LOGP 参数的缺省值中显示了 CSQ6LOGP 的缺省参数以及是否可以使用 SET LOG 命令更改每个参数。 如果需要更改其中任何值，请参阅参数的详细描述。

表 *78: CSQ6LOGP* 参数的缺省值

参数	描述	缺省值	SET 命令
COMPLOG	控制是否启用日志压缩。	无	X
DEALLCT	归档磁带机在被释放之前未使用的时间长度。	零	X
INBUFF	活动日志数据集和归档日志数据集的输入缓冲区存储器大小。	60 KB	-
MAXARCH	可记录的最大归档日志卷数。	500	X
MAXCNOFF	可以并行运行的最大 CSQJOFF7 卸载任务数。	31	-
MAXRTU	分配给同时读取归档日志磁带卷的最大专用磁带单元数。	2	X
OFFLOAD	正在打开或关闭归档。	是 (ON)	-
OUTBUFF	活动日志数据集和归档日志数据集的输出缓冲区存储器大小。	4 000 KB	-
TWOACTV	单项或双项活动日志记录。	是 (对偶)	-
TWOARCH	单一或双重归档日志记录。	是 (对偶)	-
TWOBSDS	单 BSDS 或双 BSDS。	是 (双 BSDS)	-

表 78: CSQ6LOGP 参数的缺省值 (继续)			
参数	**描述**	**缺省值**	**SET 命令**
WRTHRSH	要在将输出缓冲区写入活动日志数据集之前填充的输出缓冲区数。	20	X
V 9.2.0 V 9.2.0 ZHYWRITE	指定是否启用 zHyper 写功能。	否	X

COMPLOG
指定是否启用日志压缩。

指定下列其中一项:

无
未启用日志压缩。

RLE
使用运行长度编码启用日志压缩。

ANY
队列管理器选择用于提供最大程度的日志记录压缩的压缩算法。 此选项将导致 RLE 压缩。

缺省值为“无”。

有关日志压缩的更多详细信息，请参阅 日志压缩。

DEALLCT
指定允许归档读磁带机在释放之前保持未使用状态的时间长度 (以分钟计)。

请指定以下某项:

- 时间 (以分钟为单位) ，范围在 0 到 1440 之间
- 无限制

指定 1440 或 NOLIMIT 意味着从不取消分配磁带机。

缺省值为 0。

从磁带读取归档日志数据时，建议您将此值设置为足够高，以允许 IBM MQ 针对多个读应用程序优化磁带处理。

INBUFF
指定用于在恢复期间读取活动日志和归档日志的输入缓冲区的大小 (以千字节计)。 使用范围在 28 到 60 之间的十进制数。 指定的值向上舍入为 4 的倍数。

缺省值为 60 KB。

建议设置:

测试系统	28 KB
生产系统	60 KB 将此值设置为最佳日志读取性能的最大值。

MAXARCH
指定可以在 BSDS 中记录的最大归档日志卷数。 超过此数字时，将在 BSDS 开始时再次开始记录。

使用范围在 10 到 1000 之间的十进制数。

缺省值为 500。

建议设置:

测试系统	500 (缺省值)

生产系统 1 000

将此值设置为最大值，以便 BSDS 可以记录尽可能多的日志。

有关日志和 BSDS 的信息，请参阅 管理 IBM MQ 资源。

MAXCNOFF
指定可以并行运行的 CSQJOFF7 卸载任务数。

这允许对队列管理器或队列管理器进行调整，以使它们不会使用所有可用的磁带机。

相反，队列管理器会等待 CSQJOFF7 卸载任务完成，然后再尝试分配任何新的归档数据集。

如果队列管理器正在归档至磁带，请设置此参数，以便并发磁带请求数不应等于或超过可用磁带单元数，否则系统可能会挂起。

请注意，如果正在使用双重归档，那么每个卸载任务都将执行这两个归档，因此需要相应地设置参数。例如，如果队列管理器对磁带进行双重归档，那么值 MAXCNOFF=2 将允许最多两个活动日志同时对四个磁带进行归档。

如果多个队列管理器共享磁带机，那么应该相应地为每个队列管理器设置 MAXCNOFF。

缺省值为 31。

指定 1 到 31 范围内的值。

MAXRTU
指定可同时分配给读取归档日志磁带卷的专用磁带机的最大数量。

此参数和 DEALLCT 参数允许 IBM MQ 优化从磁带设备读取的归档日志。

指定 1 到 99 范围内的值。

缺省值为 2。

建议您将该值设置为至少比可用于 IBM MQ 的磁带机数少一个。 否则，卸载过程可能会延迟，这可能会影响系统的性能。 对于归档日志处理期间的最大吞吐量，请为此选项指定可能的最大值，请记住至少需要一个磁带机进行卸载处理。

OFFLOAD
指定归档是“打开”还是“关闭”。

指定下列其中一项:

YES
归档已开启

否
已关闭归档

缺省值为 YES。

注意: 请勿 关闭归档，除非您正在测试环境中工作。 如果您将其关闭，那么无法保证在发生系统或事务故障时将恢复数据。

OUTBUFF
指定要由 IBM MQ 用于写入活动日志数据集和归档日志数据集的输出缓冲区的存储器的总大小 (以千字节计)。 每个输出缓冲区为 4 KB。

参数必须在 128 到 4000 范围内。 指定的值向上舍入为 4 的倍数。 出于兼容性原因，将接受 40 到 128 之间的值，并将其视为值 128。

缺省值为 4000 KB。

建议设置:

测试系统 400 KB

生产系统	4 000 KB 将此值设置为最大值以避免日志输出缓冲区耗尽。

TWOACTV
指定单项或双项活动日志记录。

指定下列其中一项:

否
单个活动日志

YES
双活动日志

缺省值为 YES。

有关使用单日志记录和双日志记录的更多信息，请参阅 管理 IBM MQ 资源。

TWOARCH
指定 IBM MQ 在卸载活动日志时生成的归档日志数。

指定下列其中一项:

否
单个归档日志

YES
双重归档日志

缺省值为 YES。

建议设置:

测试系统	否
生产系统	是（缺省值）

有关使用单日志记录和双日志记录的更多信息，请参阅 管理 IBM MQ 资源。

TWOBSDS
指定引导数据集的数量。

指定下列其中一项:

否
单个 BSDS

YES
双 BSDS

缺省值为 YES。

有关使用单日志记录和双日志记录的更多信息，请参阅 管理 IBM MQ 资源。

WRTHRSH
指定要在将 4 KB 输出缓冲区写入活动日志数据集之前填充这些缓冲区的数目。

缓冲区数量越大，执行写操作的频率越低，这将提高 IBM MQ 的性能。 如果发生重要事件 (例如落实点) ，那么可能会在达到此数目之前写入缓冲区。

指定 1 到 256 范围内的缓冲区数。

缺省值为 20。

V 9.2.0 **ZHYWRITE**
指定是否在启用 zHyperWrite 的情况下对活动日志执行写操作。 活动日志数据集必须位于支持 zHyperWrite 的卷上，才能启用 zHyperWrite。

有关对活动日志启用 zHyperWrite 的更多信息，请参阅对 IBM MQ 活动日志使用 zHyperWrite。

值可以是：

否
不启用 zHyperWrite。

YES
启用 zHyperWrite。

相关参考

第 754 页的『使用 CSQ6SYSP』
使用本主题作为如何使用 CSQ6SYSP 设置系统参数的参考。

第 766 页的『使用 CSQ6ARVP』
使用本主题作为如何使用 CSQ6ARVP 指定归档环境的参考

z/OS 使用 *CSQ6ARVP*
使用本主题作为如何使用 CSQ6ARVP 指定归档环境的参考

使用 CSQ6ARVP 来建立归档环境。

第 766 页的表 79 中显示了 CSQ6ARVP 的缺省参数以及是否可以使用 SET ARCHIVE 命令更改每个参数。 如果需要更改其中任何值，请参阅参数的详细描述。 有关规划存储器的更多信息，请参阅 在 z/OS 上规划存储器和性能需求。

表 *79: CSQ6ARVP* 参数的缺省值

参数	描述	缺省值	SET 命令
ALCUNIT	进行主空间和辅助空间分配的单元。	BLK (块)	X
ARCPFX1	第一个归档日志数据集名称的前缀。	CSQARC1	X
ARCPFX2	第二个归档日志数据集名称的前缀。	CSQARC2	X
ARCRETN	归档日志数据集的保留期 (以天计)。	9999	X
ARCWRTC	向操作员发送有关归档日志数据集的消息的路由代码列表。	1,3,4	X
ARCWTOR	是否向操作员发送消息并等待应答，然后再尝试安装归档日志数据集。	YES	X
blksize	归档日志数据集的块大小。	28 672	X
Catalog	是否在 ICF 中对归档日志数据集进行编目。	否	X
精简安装	是否应压缩归档日志数据集。	否	X
PRIQTY	DASD 数据集的主空间分配。	25 715	X
PROTECT	创建数据集时，归档日志数据集是否受 ESM 概要文件保护。	否	X
QUIESCE	指定了 MODE (QUIESCE) 的 ARCHIVE LOG 时允许停顿的最长时间 (以秒计)。	5	X
SECQTY	DASD 数据集的辅助空间分配。 请参阅 ALCUNIT 参数以了解要使用的单元。	540	X
TSTAMP	归档数据集名称是否应包含时间戳记。	否	X
UNIT	存储归档日志数据集的第一个副本的设备类型或单元名称。	磁带	X
UNIT2	存储归档日志数据集的第二个副本的设备类型或单元名称。	Blank	X

ALCUNIT
指定分配主要及次要空间的单位。

指定以下一项:

CYL
柱面

TRK
跟踪

BLK
块

建议您使用 BLK，因为它与设备类型无关。

缺省值为 BLK。

如果归档 DASD 卷上的可用空间可能已分段，那么建议您指定较小的主扩展数据块并允许扩展至辅助扩展数据块。 有关活动日志的空间分配的更多信息，请参阅 规划日志归档存储器。

ARCPFX1
指定第一个归档日志数据集名称的前缀。

请参阅 TSTAMP 参数，以获取有关如何命名数据集的描述以及对 ARCPFX1 长度的限制。

此参数不能留空。

缺省值为 CSQARC1。

您可能需要对与 IBM MQ 队列管理器地址空间相关联的用户标识进行授权，以创建具有此前缀的归档日志。

ARCPFX2
指定第二个归档日志数据集名称的前缀。

请参阅 TSTAMP 参数，以获取有关如何命名数据集的描述以及对 ARCPFX2 长度的限制。

即使将 TWOARCH 参数指定为 NO，此参数也不能为空。

缺省值为 CSQARC2。

您可能需要对与 IBM MQ 队列管理器地址空间相关联的用户标识进行授权，以创建具有此前缀的归档日志。

ARCRETN
指定创建归档日志数据集时要使用的保留期 (以天计)。

参数必须在范围 0 到 9999 之间。

缺省值是 9999。

建议设置:

测试系统	3 在测试系统中，可能不需要长时间归档日志。
生产系统	9 999 (缺省值) 将此值设置为高，可有效关闭自动归档日志删除。

有关废弃归档日志数据集的更多信息，请参阅 废弃归档日志数据集。

ARCWRTC
指定向操作员发送有关归档日志数据集的消息的 z/OS 路由代码列表。 如果 ARCWTOR 设置为 NO，那么将忽略此字段。

最多指定 14 个路由代码，每个路由代码的值在 1 到 16 的范围内。 必须至少指定一个代码。 用逗号而不是空格分隔列表中的代码。

缺省值为值列表: 1,3,4。

有关 z/OS 路由代码的更多信息，请参阅 *z/OS MVS System Messages* 手册的其中一个卷中的 消息描述 中的 路由代码 。

ARCWTOR
指定是否先发送一条消息给操作员并接收回复，再尝试装载归档日志数据集。

可能会强制其他 IBM MQ 用户等待，直到安装了数据集，但在 IBM MQ 等待此消息的应答时，这些用户不会受影响。

指定下列其中一项:

YES
设备需要很长时间来安装归档日志数据集。 例如，磁带机。

否
设备没有长时间延迟。 例如， DASD。

缺省值为 YES。

建议设置:

测试系统	否
生产系统	是（缺省值） 这取决于操作过程。 如果使用磁带机器人，那么 NO 可能更合适。

BLKSIZE
指定归档日志数据集的块大小。 您指定的块大小必须与在 UNIT 参数中指定的设备类型兼容。

参数必须在 4 097 到 28 672 范围内。 您指定的值向上舍入为 4 096 的倍数。

缺省值为 28 672。

此参数被存储管理子系统 (SMS) 数据类块大小覆盖 (如果提供)。

如果将归档日志数据集写入 DASD ，那么建议您选择允许每个磁道有两个块的最大块大小。 例如，对于 3390 设备，应使用块大小 24 576。

如果将归档日志数据集写入磁带，那么指定可能的最大块大小将提高读取归档日志的速度。 您应该使用 28 672 的块大小。

建议设置:

测试系统	根据用于归档日志的介质，使用块大小建议。 即，对于磁盘 24 576 和磁带 28 672。
生产系统	根据用于归档日志的介质，使用块大小建议。 即，对于磁盘 24 576 和磁带 28 672。

CATALOG
指定是否在主要的集成目录设施 (ICF) 目录中编目归档日志数据集。

指定下列其中一项:

否
未对归档日志数据集进行编目

YES
归档日志数据集已编目

缺省值为 NO。

必须对 DASD 上分配的所有归档日志数据集进行编目。 如果在将 CATALOG 参数设置为 NO 的情况下归档到 DASD ，那么每次分配归档日志数据集时都会显示消息 CSQJ072E ，并 IBM MQ 对数据集进行编目。

建议设置:

测试系统	YES
生产系统	YES，在 DASD 上分配归档时

COMPACT
指定是否对写入归档日志的数据进行压缩。 此选项仅适用于带有增强的数据记录能力（IDRC）功能部件的 3480 设备或 3490 设备。 当打开此功能部件时，磁带控制单元中的硬件以比正常密度高很多的密度写入数据，这就使每卷能容纳更多数据。 如果不将 3480 设备与 IDRC 功能部件或 3490 基本型号 (3490E 除外) 配合使用，请指定 NO。 如果要压缩数据，请指定 YES。

指定下列其中一项:

否
请勿压缩数据集

YES
压缩数据集

缺省值为 NO。

指定 YES 将对性能产生负面影响。 还要注意，压缩到磁带的数据只能使用支持 IDRC 功能的设备进行读取。 如果必须将归档磁带发送到另一个站点以进行远程恢复，那么这可能是一个问题。

建议设置:

测试系统	不适用
生产系统	NO (缺省值) 这仅适用于 3480 和 3490 IDR 压缩。 将此值设置为 YES 可能会在恢复和重新启动期间降低归档日志读取性能; 但是，这不会影响对磁带的写入。

PRIQTY
指定 ALCUNITs 中 DASD 数据集的主空间分配。

此值必须大于零。

缺省值为 25 715。

此值必须足以用于日志数据集或其相应 BSDS (以较大者为准) 的副本。 要确定必需的值，请遵循以下过程:

1. 确定分配的活动日志记录数 (c) ，如 第 751 页的『创建引导程序和日志数据集』中所述。
2. 确定每个归档日志块中的 4096 字节块数:

```
d = BLKSIZE / 4096
```

   其中 BLKSIZE 是向上取整的值。
3. 如果 ALCUNIT = BLK:

```
PRIQTY = INT(c / d) + 1
```

   其中 INT 表示向下舍入为整数。

   如果 ALCUNIT = TRK:

```
PRIQTY = INT(c / (d * INT(e/BLKSIZE))) + 1
```

   其中 e 是每个磁道的字节数 (对于 3390 设备为 56664) ， INT 表示向下舍入为整数。

如果 ALCUNIT = CYL:

```
PRIQTY = INT(c / (d * INT(e/BLKSIZE) * f)) + 1
```

其中 f 是每个柱面的磁道数 (对于 3390 设备为 15) ， INT 表示向下舍入为整数。

有关使日志和归档数据集的大小的信息，请参阅 第 751 页的『创建引导程序和日志数据集』 和 第 752 页的『定义页面集』。

建议设置:

测试系统

1 680

足以保存整个活动日志，即:

```
10  080 / 6 = 1  680 blocks
```

生产系统

归档到磁带时不适用。

如果归档 DASD 卷上的可用空间可能已分段，那么建议您指定较小的主扩展数据块并允许扩展至辅助扩展数据块。 有关活动日志的空间分配的更多信息，请参阅 在 z/OS 上规划。

PROTECT

指定创建归档日志数据集时，是否要通过独立的 ESM（外部安全性管理器）配置文件保护这些数据集。

指定下列其中一项:

否

未创建概要文件。

YES

在卸载日志时创建离散数据集概要文件。 如果指定 YES:

- 对于 IBM MQ， ESM 保护必须处于活动状态。
- 与 IBM MQ 队列管理器地址空间相关联的用户标识必须有权创建这些概要文件。
- 如果要归档到磁带，那么 TAPEVOL 类必须处于活动状态。

否则，卸载将失败。

缺省值为 NO。

QUIESCE

指定在指定了 MODE (QUIESCE) 的情况下发出 ARCHIVE LOG 命令时允许停顿的最长时间 (以秒计)。

该参数必须在 1 到 999 范围内。

缺省值为 5。

SECQTY

指定 ALCUNITs 中 DASD 数据集的辅助空间分配。 辅助扩展数据块最多可分配 15 次; 请参阅 *z/OS MVS JCL Reference* 和 *z/OS MVS JCL User 's Guide* 以获取详细信息。

参数必须大于零。

缺省值为 540。

TSTAMP

指定归档日志数据集名称中是否包含时间戳记。

指定下列其中一项:

否

名称不包含时间戳记。 归档日志数据集命名为:

```
arcpfxi.A nnnnnnn
```

其中 *arcpfxi* 是由 ARCPFX1 或 ARCPFX2 指定的数据集名称前缀。 *arcpfxi* 最多可以有 35 个字符。

YES
名称包含时间戳记。 归档日志数据集命名为:

```
arcpfxi.cyyddd.T hhmmsst.A nnnnnnn
```

其中， *c* 是 "D" 表示直至 (包括 1999 年) 的年份或 "E" 表示 2000 年及以后的年份， *arcpfxi* 是由 ARCPFX1 或 ARCPFX2 指定的数据集名称前缀。 *arcpfxi* 最多可以有 19 个字符。

EXT
名称包含时间戳记。 归档日志数据集命名为:

```
arcpfxi.D yyyyddd.T hhmmsst.A nnnnnnn
```

其中 *arcpfxi* 是由 ARCPFX1 或 ARCPFX2 指定的数据集名称前缀。 *arcpfxi* 最多可以有 17 个字符。

缺省值为 NO。

UNIT
指定用于存储归档日志数据集的第一个副本的设备的设备类型或单元名称。

通过 8 字母数字字符指定 1 的设备类型或单元名称。 第一个字符必须是字母。

此参数不能为空。

缺省值为 TAPE。

如果归档到 DASD ，那么可以指定具有有限卷范围的通用设备类型，例如 UNIT=3390。

如果归档到 DASD ， 请确保:

- 主空间分配足够大，可以包含来自活动日志数据集的所有数据。
- 归档日志数据集目录选项 (CATALOG) 设置为 YES。
- 您对 BLKSIZE 使用了正确的值。

如果归档到 TAPE ，那么 IBM MQ 最多可扩展至 20 个卷。

建议设置:

测试系统	DASD
生产系统	磁带

有关选择归档日志位置的更多信息，请参阅 在 z/OS 上规划。

UNIT2
指定用于存储归档日志数据集的第二个副本的设备的设备类型或单元名称。

通过 8 字母数字字符指定 1 的设备类型或单元名称。 第一个字符必须是字母。 如果此参数为空，那么将使用为 UNIT 参数设置的值。

缺省值为空白。

相关参考
第 754 页的『使用 CSQ6SYSP』
使用本主题作为如何使用 CSQ6SYSP 设置系统参数的参考。

第 762 页的『使用 CSQ6LOGP』
使用本主题作为如何使用 CSQ6LOGP 指定日志记录选项的参考。

z/OS 使用 *CSQ6USGP*
使用本主题作为如何使用 CSQ6USGP 设置系统参数的参考

使用 CSQ6USGP 来控制产品使用情况记录。

CSQ6USGP 的缺省参数显示在 第 772 页的表 80 中。 如果需要更改其中任何值，请参阅参数的详细描述。

注意: 不能使用 SET SYSTEM 命令更改这些参数中的任何一个。

表 *80: CSQ6USGP* 参数的缺省值

参数	描述	缺省值
QMGRPROD	要针对其记录队列管理器使用情况的产品	Blank
AMSPROD	要针对其记录 Advanced Message Security (AMS) 使用情况的产品	Blank

QMGRPROD
指定要记录队列管理器使用情况的产品。

指定以下一项:

MQ
队列管理器使用情况记录为独立 IBM MQ for z/OS 产品，产品标识为 5655-MQ9。

VUE
队列管理器使用情况记录为独立 IBM MQ for z/OS Value Unit Edition (VUE) 产品，产品标识为 5655-VU9。

ADVANCEDVUE
队列管理器使用情况记录为 IBM MQ Advanced for z/OS Value Unit Edition 产品的一部分，产品标识为 5655-AV1。

AMSPROD
V 9.2.0 如果未设置此参数，那么 AMS 地址空间将不会启动，并且将输出消息 CSQY024I 。

指定要对其记录 Advanced Message Security 使用情况 (如果使用) 的产品。

指定以下一项:

AMS
AMS 使用情况记录为独立 Advanced Message Security for z/OS 产品，产品标识为 5655-AM9。

ADVANCED
AMS 使用情况记录为 IBM MQ Advanced for z/OS 产品的一部分，产品标识为 5655-AV9。

ADVANCEDVUE
AMS 使用情况记录为 IBM MQ Advanced for z/OS Value Unit Edition 产品的一部分，产品标识为 5655-AV1。

请参阅 报告产品信息 ，以获取有关产品使用情况记录的更多信息。

相关参考
第 754 页的『使用 CSQ6SYSP』
使用本主题作为如何使用 CSQ6SYSP 设置系统参数的参考。

第 762 页的『使用 CSQ6LOGP』
使用本主题作为如何使用 CSQ6LOGP 指定日志记录选项的参考。

z/OS 定制通道启动程序参数

使用 ALTER QMGR 来定制通道启动程序以满足您的需求。

- 根据需要对每个 *IBM MQ* 队列管理器重复此任务。
- 从先前版本迁移时，必须执行此任务。

许多队列管理器属性控制分布式排队的操作方式。 使用 MQSC 命令 ALTER QMGR 来设置这些属性。 初始化数据集样本 thlqual.SCSQPROC(CSQ4INYG) 包含一些可以定制的设置。 有关更多信息，请参阅 ALTER QMGR。

每次启动通道启动程序时，这些参数的值都显示为一系列消息。

适配器，分派器和最大通道数之间的关系

ALTER QMGR 参数 CHIADAPS 和 CHIDISPS 定义通道启动程序使用的任务控制块 (TCB) 数。 CHIADAPS (适配器) TCB 用于对队列管理器进行 IBM MQ API 调用。 CHIDISPS (分派器) TCB 用于对通信网络进行呼叫。

ALTER QMGR 参数 MAXCHL 会影响通道在分派器 TCB 上的分布。

CHIDISPS

如果您有少量通道，请使用缺省值。

每个处理器一个任务可优化系统性能。 由于分派器任务是 CPU 密集型任务，因此原则是尽量减少任务的忙碌，从而将查找和启动线程所花费的时间降至最低。

CHIDISPS (20) 适用于具有 100 个以上信道的系统。 在有 CHIDISPS (20) 的情况下，如果这是比需要更多的分派器 TCB ，则不大可能有任何重大的劣势。

作为准则，如果您有超过 1000 个通道，那么允许每 50 个当前通道有一个分派器。 例如，指定 CHIDISPS (40) 以处理最多 2000 个活动通道。

如果您正在使用 TCP/IP ，那么用于 TCP/IP 通道的最大分派器数为 100 ，即使您在 CHIDISPS 中指定了较大的值也是如此。

CHIADAPS

对队列管理器的每个 IBM MQ API 调用都独立于任何其他 API 调用，并且可以在任何适配器 TCB 上进行。 由于日志 I/O ，使用持久消息的调用可能比使用非持久消息的调用耗时长得多。 因此，在许多通道中处理大量持久消息的通道启动程序可能需要比缺省 8 适配器 TCB 更多的资源以实现最佳性能。 在实现的 batchsize 较小的情况下尤其如此，因为批处理结束还需要日志 I/O ，并且使用瘦客户机通道。

生产环境的建议值为 CHIADAPS (30)。 使用超过此值不太可能带来任何显着的额外好处，如果这是比需要更多的适配器 TCB ，那么在具有 CHIADAPS (30) 方面也不太可能有任何重大的缺点。

MAXCHL

每个通道在通道启动时都与特定分派器 TCB 相关联，并保持与该 TCB 相关联，直到通道停止为止。 许多通道可以共享每个 TCB。 MAXCHL 用于跨可用分派器 TCB 传播通道。 第一个 (MIN ((MAXCHL/CHIDISPS)) ， 10)) 要启动的通道与第一个分派器 TCB 相关联，依此类推，直到所有分派器 TCB 都在使用中为止。

这对于少数通道和大型 MAXCHL 的影响是通道在分派器之间分布不均匀。 例如，如果设置 CHIDISPS (10) 并将 MAXCHL 保留为其缺省值 200 ，但只有 50 个通道，那么 5 个分派器将与每个通道 10 个通道相关联， 5 个分派器将未使用。 我们建议将 MAXCHL 设置为实际要使用的通道数 (如果这是一个小的固定数目)。

如果更改此队列管理器属性，那么还必须查看 ACTCHL ， LU62CHL 和 TCPCHL 队列管理器属性以确保这些值兼容。 请参阅 队列管理器参数 ，以获取这些属性及其关系的完整描述。

为通道启动程序设置 z/OS UNIX System Services 环境

通道启动程序 (CHINIT) 使用 OMVS 线程。 在创建新的 CHINIT 或修改分派器数或 SSLTASKS 之前，请查看 OMVS 配置参数。

每个 CHINIT 都使用 3 + CHIDISP + SSLTASKS OMVS 线程。 这些占 LPAR 中使用的 OMVS 线程总数的比例，占 CHINIT 启动式任务用户标识使用的线程数的比例。

您可以使用 **D OMVS,L** 并查看 MAXPROCSYS 的当前使用情况，高水位使用情况和系统限制 (系统允许的最大进程数)。

如果要添加新的 CHINIT 或增加 CHIDISPS 或 SSLTASKS 的值，那么必须计算线程数的增加，并查看对 MAXPROCSYS 值的影响。 您可以使用 **SETOMVS** 命令动态更改 MAXPROCSYS 和/或更新 BPXPRCxx 参数库值。

OMVS 参数 MAXPROCUSER 是具有相同 UID 的单个 OMVS 用户可拥有的 OMVS 线程数。 线程计入此值。 因此，如果您具有具有相同启动式任务用户标识的 2 CHINITS ，并且每个具有 10 个分派器和 3 SSLTASKS ，那么对于 OMVS uid ，将有 2 * (3 +10 + 3) = 32 个线程。

您可以通过发出 **D OMVS,O** 命令来显示缺省 MAXPROCUSER，并且可以使用 **SETOMVS** 命令动态更改 MAXPROCUSER 和/或更新 BPXPRCxx parmlib 值。

您可以使用 RACF 命令 **ALTUSER userid OMVS(PROCUSERMAX(nnnn))** 或等效命令来覆盖每个用户的此值。

要启动通道启动程序，请发出以下命令:

```
START CHINIT
```

要确保通道启动器已成功启动，请检查 xxxxCHIN(ssidCHIN) 工作日志中是否有 ICH408I 错误。

相关概念

第 774 页的『设置批处理， TSO 和 RRS 适配器』
通过将库添加到相应的 STEPLIB 并置，使适配器可供应用程序使用。 为了满足适配器发出的 SNAP 转储，请分配 CSQSNAP DDname。 考虑使用 CSQBDEFV 来提高应用程序的可移植性

通道启动程序统计数据记录

z/OS 设置批处理， *TSO* 和 *RRS* 适配器

通过将库添加到相应的 STEPLIB 并置，使适配器可供应用程序使用。 为了满足适配器发出的 SNAP 转储，请分配 CSQSNAP DDname。 考虑使用 CSQBDEFV 来提高应用程序的可移植性

- 根据需要对每个 *IBM MQ* 队列管理器重复此任务。
- 从先前版本进行迁移时，可能需要执行此任务。

要使适配器可用于使用批处理连接的批处理和其他应用程序，请将以下 IBM MQ 库添加到批处理应用程序的 STEPLIB 并置:

- thlqual.SCSQANL *x*
- thlqual.SCSQAUTH

其中 *x* 是本地语言的语言字母。 (如果库位于 LPA 或链接列表中，那么不需要执行此操作。)

对于 TSO 应用程序，请将库添加到 TSO 登录过程中的 STEPLIB 并置，或者使用 TSO 命令 TSOLIB 激活这些库。

如果适配器检测到意外的 IBM MQ 错误，那么它会向 DDname CSQSNAP 发出 z/OS SNAP 转储，并向应用程序发出原因码 MQRC_UNEXPECTED_ERROR 。如果 CSQSNAP DD 语句不在应用程序 JCL 中，或者未将 CSQSNAP 分配给 TSO 下的数据集，那么不会执行转储。 如果发生这种情况，那么可以在应用程序 JCL 中包含 CSQSNAP DD 语句，或者将 CSQSNAP 分配到 TSO 下的数据集，然后重新运行应用程序。 但是，由于某些问题是间歇性的，因此建议您在应用程序 JCL 中包含 CSQSNAP 语句，或者将 CSQSNAP 分配给 TSO 登录过程中的数据集，以捕获发生故障的原因。

提供的程序 CSQBDEFV 可提高应用程序的可移植性。 在 CSQBDEFV 中，可以指定要连接到的队列管理器或队列共享组的名称，而不是在应用程序中的 MQCONN 或 MQCONNX 调用中指定该名称。 您可以为每个队列管理器或队列共享组创建新版本的 CSQBDEFV。 为此，请完成下列步骤:

1. 将 IBM MQ 汇编程序 CSQBDEFV 从 thlqual.SCSQASMS 复制到用户库。
2. 提供的程序包含缺省子系统名称 CSQ1。 您可以保留此名称以进行测试和安装验证。 对于生产子系统，可以将 NAME=CSQ1 更改为 1 到 4 个字符的子系统名称，或者使用 CSQ1。

   如果您正在使用队列共享组，那么可以指定队列共享组名，而不是指定 CSQ1。 如果执行此操作，那么程序会向该组中的活动队列管理器发出连接请求。
3. 组装并链接编辑程序以生成 CSQBDEFV 装入模块。 对于组合件，请在 SYSLIB 并置中包含库 thlqual.SCSQMACS ; 使用链接编辑参数 RENT,AMODE=31,RMODE=ANY。 这在 thlqual.SCSQPROC(CSQ4DEFV) 中的样本 JCL 中显示。 然后将装入库包含在 z/OS Batch 或 TSO STEPLIB 中，先于 thlqual.SCSQAUTH。

相关概念

第 775 页的『设置操作和控制面板』

要设置操作和控制面板，必须首先设置包含所需面板， EXECs ，消息和表的库。 要执行此操作，必须考虑要用于面板的本地语言功能部件。 完成此操作后，可以选择更新 IBM MQ 操作和控制面板的主 ISPF 菜单，并更改功能键设置。

z/OS 设置操作和控制面板

要设置操作和控制面板，必须首先设置包含所需面板， EXECs ，消息和表的库。 要执行此操作，必须考虑要用于面板的本地语言功能部件。 完成此操作后，可以选择更新 IBM MQ 操作和控制面板的主 ISPF 菜单，并更改功能键设置。

- 您需要对要运行 *IBM MQ* 的每个 *z/OS* 系统执行一次此任务。
- 从先前版本进行迁移时，可能需要执行此任务。

设置库

执行以下步骤以设置 IBM MQ 操作和控制面板:

1. 确保并置中包含的所有库都采用相同的格式 (F ， FB ， V ， VB) 并且具有相同的块大小，或者按块大小递减的顺序。 否则，尝试使用这些面板时可能会迁到问题。
2. 将库 thlqual.SCSQEXEC 包含在 SYSEXEC 或 SYSPROC 并置中，或者使用 TSO ALTLIB 命令将其激活。 在安装期间使用固定块 80 记录格式分配的此库包含必需的 EXECs。

   最好将库放入 SYSEXEC 并置中。 但是，如果要将其放入 SYSPROC 中，那么库必须具有 80 字节的记录长度。
3. 将 thlqual.SCSQAUTH 和 thlqual.SCSQANLx 添加到 TSO 登录过程 STEPLIB 或使用 TSO TSOLIB 命令将其激活 (如果它不在链接列表或 LPA 中)。
4. 您可以将 IBM MQ 面板库永久添加到 ISPF 库设置，也可以允许在使用这些面板时动态设置这些面板库。 对于前一种选择，您需要执行以下操作:

   a. 在 ISPPLIB 并置中包含包含操作和控制面板定义的库。 名称为 thlqual.SCSQPNLx，其中 x 是本地语言的语言字母。

   b. 将包含所需表的库包含在 ISPTLIB 并置中。 名称为 thlqual.SCSQTBLx，其中 x 是本地语言的语言字母。

   c. 将包含所需消息的库包含在 ISPMLIB 并置中。 名称为 thlqual.SCSQMSGx，其中 x 是本地语言的语言字母。

   d. 将包含所需装入模块的库包含在 ISPLLIB 并置中。 此库的名称为 thlqual.SCSQAUTH。

   对于后一个选项，请使用 z/OS LIBDEF 命令。 请参阅 示例 ，以获取指向您可以使用的各种关键字的链接。
5. 测试您是否可以从 "TSO 命令处理器" 面板访问 IBM MQ 面板。 这通常是 ISPF/PDF 主选项菜单上的选项 6。 您运行的 EXEC 的名称为 CSQOREXX。如果在 ISPF 设置中永久放置 IBM MQ 库 (如步骤 4 中所示) ，那么没有要指定的参数。 如果没有，请使用以下命令:

   ```
   CSQOREXX thlqual langletter
   ```

   其中 `langletter` 是一个字母，用于标识要使用的本地语言:

   C
   简体中文

   E
   U.S. 英语 (混合大小写)

   F
   法语

   K
   日语

U
U.S. 英语 (大写)

更新 ISPF 菜单

您可以更新 ISPF 主菜单，以允许从 ISPF 访问 IBM MQ 操作和控制面板。 The required setting for &ZSEL is:

```
CMD(%CSQOREXX thlqual langletter)
```

有关 `thlqual` 和 `langletter` 的信息，请参阅步骤 第 775 页的『5』。

有关更多详细信息，请参阅 *z/OS: ISPF Dialog Developer 's Guide and Reference* 手册。

更新功能键和命令设置

您可以使用常规 ISPF 过程来更改面板所使用的功能键和命令设置。 应用程序标识为 CSQO。

但是，建议不要执行此操作，因为未更新帮助信息以反映您所做的任何更改。

相关概念
第 776 页的『包含 IBM MQ 转储格式化成员』
要能够使用交互式问题控制系统 (IPCS) 格式化 IBM MQ 转储，必须更新一些系统库。

z/OS 包含 *IBM MQ* 转储格式化成员

要能够使用交互式问题控制系统 (IPCS) 格式化 IBM MQ 转储，必须更新一些系统库。

- 您需要对要运行 *IBM MQ* 的每个 *z/OS* 系统执行一次此任务。
- 从先前版本迁移时，需要执行此任务。

为了能够使用交互式问题控制系统 (IPCS) 格式化 IBM MQ 转储，请将数据集 thlqual.SCSQPROC(CSQ7IPCS) 复制到 SYS1.PARMLIB。 您不应该需要编辑此数据集。

如果已为 IPCS 定制了 TSO 过程，那么可以将 thlqual.SCSQPROC(CSQ7IPCS) 复制到 IPCSPARM 定义中的任何库中。 有关 IPCSPARM 的详细信息，请参阅 *z/OS MVS IPCS Customization* 手册。

还必须在 ISPPLIB 并置中包含库 thlqual.SCSQPNLA 。

要使转储格式化程序可用于 TSO 会话或 IPCS 作业，还必须在 STEPLIB 并置中包含库 thlqual.SCSQAUTH ，或者使用 TSO TSOLIB 命令将其激活 (即使它已在链接列表或 LPA 中)。

相关概念
第 776 页的『禁止参考消息』
IBM MQ 系统可能会生成大量参考消息。 您可以阻止将所选消息发送到控制台或硬拷贝日志。

z/OS 禁止参考消息

IBM MQ 系统可能会生成大量参考消息。 您可以阻止将所选消息发送到控制台或硬拷贝日志。

- 您需要对要运行 *IBM MQ* 的每个 *z/OS* 系统执行一次此任务。
- 从先前版本迁移时，不需要执行此任务。

如果大量使用 IBM MQ 系统，并且许多通道正在停止和启动，那么会将大量参考消息发送到 z/OS 控制台和硬拷贝日志。 IBM MQ - IMS 网桥和缓冲区管理器还可能生成大量参考消息。

如果需要，您可以使用由 SYS1.PARMLIB 的 MPFLSTxx 成员指定的 z/OS 消息处理设施列表来禁止其中一些控制台消息。 您指定的消息仍显示在硬拷贝日志上，但不显示在控制台上。

样本 thlqual.SCSQPROC(CSQ4MPFL) 显示了 MPFLSTxx 的建议设置。 请参阅 *z/OS MVS Initialization and Tuning Reference* 手册，以获取有关 MPFLSTxx 的更多信息。

如果要在硬拷贝日志上禁止所选参考消息，那么可以使用 z/OS 安装出口 IEAVMXIT。 您可以为所需消息设置以下位开关为 ON:

CTXTRDTM
删除消息。

此消息不会显示在控制台上，也不会以硬拷贝方式登录。

CTXTESJL
从作业记录中禁止。

消息不会进入 JES 作业日志。

CTXTNWTP
不执行 WTP 处理。

消息不会发送到 TSO 终端或批处理作业的系统消息数据集。

注:

1. 有关其他参数的完整详细信息，请参阅 MVS 安装出口 文档。
2. 建议不要禁止除建议黑名单 CSQ4MPFL 中的消息以外的消息。

此外，还可以指定额外的参数:

EXCLMSG
指定要从任何日志中排除的消息的列表。

此列表中的消息不会发送到 z/OS 控制台和硬拷贝日志。 请参阅 第 754 页的『使用 CSQ6SYSP』 中的 EXCLMSG 以获取更多信息。

相关任务

第 790 页的『在 z/OS 上测试队列管理器』
定制或迁移队列管理器后，可以通过运行安装验证程序以及 IBM MQ for z/OS 随附的一些样本应用程序来对其进行测试。

z/OS 配置队列共享组

如果要将共享队列用于高可用性，请将这些主题用作配置队列共享组的逐步指南。

完成设置 IBM MQ for z/OS 系统的过程的此部分中的步骤后，应 第 752 页的『定制系统参数模块』 添加队列共享组数据。 您需要修改 CSQ6SYSP 以指定 QSGDATA 参数。

z/OS 设置 *Db2* 环境

如果您正在使用队列共享组，那么必须通过定制和运行多个样本作业来创建所需的 Db2 对象。

设置 Db2 环境

您必须通过定制和运行多个样本作业来创建和绑定必需的 Db2 对象。

- 对每个 Db2 数据共享组重复此任务。
- 从先前版本迁移时，需要执行 `bind` 和 `grant` 步骤。
- 如果未使用队列共享组，请省略此任务。

  如果以后要使用队列共享组，请在该时间执行此任务。

IBM MQ 提供了两组等效作业。 具有 CSQ45 前缀的那些前缀用于与较早版本的 IBM MQ 兼容以及用于 Db2 V11 和更低版本。 如果要使用 Db2 V12 或更高版本设置新的数据共享组，那么建议您使用具有 CSQ4X 前缀的作业，因为这些作业会利用最新的 Db2 功能来进行动态大小调整和通用表空间 (UTS)。

必须对每个新的 Db2 数据共享组执行以下步骤。 所有样本 JCL 都在 thlqual.SCSQPROC 中。

1. V 9.2.0 定制并执行样本 JCL CSQ4XCSG 以创建要用于 IBM MQ 数据库，表空间和表的存储器组。

2. V 9.2.0 定制并执行样本 JCL CSQ4XCDB，以创建要由连接到此 Db2 数据共享组的所有队列管理器使用的数据库。
3. V 9.2.0 定制并执行样本 JCL CSQ4XCTS 以创建包含用于队列共享组的队列管理器和通道启动程序表的表空间。
4. V 9.2.0 定制并执行样本 JCL CSQ4XCTB 以创建 15 个 Db2 表和关联的索引。 请勿更改任何行名称或属性。
5. 定制并执行样本 JCL CSQ45BPL 以绑定队列管理器，实用程序和通道启动程序的 Db2 计划。
6. 定制并执行样本 JCL CSQ45GEX，以授予队列管理器，实用程序和通道启动程序所使用的用户标识的计划执行权限。 队列管理器和通道启动程序的用户标识是运行启动式任务过程的用户标识。 实用程序的用户标识是可提交批处理作业的用户标识。

   下表显示了相应计划的名称:

   - LTS LTS 列中的 Long Term Support 版本。
   - CD CD 列中的 Continuous Delivery 版本，其中 n 表示 CD 发行版。

用户	计划 (LTS)	计划 (CD)
队列管理器	CSQ5A 920 CSQ5C 920 CSQ5D 920 CSQ5K 920 CSQ5L 920 CSQ5M 920 CSQ5P 920 CSQ5R 920 CSQ5S 920 CSQ5T 920 CSQ5U 920 CSQ5W 920	V 9.2.0 CSQ5A 9X0, CSQ5C 9X0, CSQ5D 9X0, CSQ5K 9X0, CSQ5L 9X0, CSQ5M 9X0, CSQ5P 9X0, CSQ5R 9X0, CSQ5S 9X0, CSQ5T 9X0, CSQ5U 9X0, CSQ5W 9X0
CSQUTIL 批处理实用程序的 SDEFS 函数	CSQ52 920	CSQ52 9X0
CSQ5PQSG 和 CSQJUCNV 批处理实用程序	CSQ5B 920	CSQ5B 9X0
CSQUZAP 服务实用程序	CSQ5Z 920	CSQ5Z 9X0

V 9.2.0 如果在 Db2 设置期间发生故障，那么可以定制并执行以下作业:

- CSQ45DTB，用于删除表和索引。
- CSQ4XDTS 以删除表空间。
- CSQ4XDDB 以删除数据库。
- CSQ4XDSG 以删除存储器组。

注: 如果这些作业由于 Db2 锁定问题而失败，那么可能是由于争用 Db2 资源所致，尤其是在系统大量使用的情况下。 稍后重新提交作业。 当系统被轻度使用或停顿时，最好运行这些作业。

请参阅 *Db2 (针对 z/OS 12.0.0)* 中的 Db2 管理，以获取有关设置 Db2 的更多信息。

有关 Db2 表大小的信息，请参阅 在 z/OS 上规划 。

相关概念

第 779 页的『设置耦合设施』

如果您正在使用队列共享组，请使用 IXCMIAPU 在耦合设施资源管理 (CFRM) 策略数据集中定义队列共享组 (QSG) 中的队列管理器所使用的耦合设施结构。

z/OS 设置耦合设施

如果您正在使用队列共享组，请使用 IXCMIAPU 在耦合设施资源管理 (CFRM) 策略数据集中定义队列共享组 (QSG) 中的队列管理器所使用的耦合设施结构。

- 对每个队列共享组重复此任务。
- 从先前版本迁移时，可能需要执行此任务。
- 如果未使用队列共享组，请省略此任务。

  如果以后要使用队列共享组，请在该时间执行此任务。

队列共享组的所有结构都以队列共享组的名称开头。 定义以下结构:

- 称为 *qsg-name* CSQ_ADMIN 的管理结构。 此结构由 IBM MQ 本身使用，并且不包含任何用户数据。
- 称为 *qsg-name* CSQSYSAPPL 的系统应用程序结构。 此结构由 IBM MQ 系统队列用于存储状态信息。
- 用于保存共享队列的消息的一个或多个结构。 这些名称可以具有您选择的任何名称，长度最多为 16 个字符。
  - 前四个字符必须是队列共享组名。 (如果队列共享组名长度小于 4 个字符，那么必须使用 @ 符号将其填充为 4 个字符。)
  - 第五个字符必须是字母，后续字符可以是字母或数字。 此部分名称 (没有队列共享组名) 是您在定义共享队列或 CF 结构对象时为 CFSTRUCT 名称指定的名称。

  在用于保存共享队列的消息的结构的名称中只能使用字母和数字字符，不能使用任何其他字符 (例如，在管理结构的名称中使用的 _ 字符)。

IXCMIAPU 的样本控制语句位于数据集 thlqual.SCSQPROC(CSQ4CFRM) 中。 定制这些元素并将其添加到耦合设施的 IXCMIAPU 作业中，然后运行该作业。

成功定义结构后，请激活正在使用的 CFRM 策略。 要执行此操作，请发出以下 z/OS 命令:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME= policy-name
```

有关规划 CF 结构及其大小的信息，请参阅 定义耦合设施资源。

相关概念

第 746 页的『实施 ESM 安全控制』
对队列管理器和通道启动程序实施安全控制。

z/OS 设置 *SMDS* 环境

如果要使用 SMDS 来卸载共享队列上的消息，请设置 SMDS 卸载存储环境。

- 对要配置以将数据卸载到 *SMDS* 的队列共享组中的每个队列管理器和结构执行此任务。
- 如果要配置其他结构以稍后将数据卸载到 *SMDS* ，那么可以在此时再次执行此任务。
- 如果未使用队列共享组，请省略此任务。

  如果您以后想要使用队列共享组，请在该时间执行此任务。

设置 SMDS 环境

1. 估算结构和数据集空间需求。 请参阅 共享消息数据集容量注意事项。
2. 分配和预先格式化数据集。 请参阅 创建共享消息数据集。
3. 将 CF 结构定义到 IBM MQ 时，请确保使用 CFLEVEL (5) 和 OFFLOAD (SMDS) 定义 CFSTRUCT。

相关概念

第 779 页的『设置耦合设施』

如果您正在使用队列共享组，请使用 IXCMIAPU 在耦合设施资源管理 (CFRM) 策略数据集中定义队列共享组 (QSG) 中的队列管理器所使用的耦合设施结构。

z/OS 将 *IBM MQ* 条目添加到 *Db2* 表

如果您正在使用队列共享组，请运行 CSQ5PQSG 实用程序以将队列共享组和队列管理器条目添加到 Db2 数据共享组中的 IBM MQ 表。

- 对每个 *IBM MQ* 队列共享组和每个队列管理器重复此任务。
- 从先前版本进行迁移时，可能需要执行此任务。
- 如果未使用队列共享组，请省略此任务。

  如果您以后想要使用队列共享组，请在该时间执行此任务。

对将成为队列共享组成员的每个队列共享组和每个队列管理器运行 CSQ5PQSG 。（管理 IBM MQ for z/OS 中描述了 CSQ5PQSG 。）

按指定顺序执行以下操作:

1. 使用 CSQ5PQSG 程序的 ADD QSG 函数将队列共享组条目添加到 IBM MQ Db2 表中。thlqual.SCSQPROC(CSQ45AQS) 中提供了样本。

   对 Db2 数据共享组中定义的每个队列共享组执行一次此功能。 在添加引用队列共享组的任何队列管理器条目之前，该队列共享组条目必须存在。
2. 使用 CSQ5PQSG 程序的 ADD QMGR 函数将队列管理器条目添加到 IBM MQ Db2 表中。thlqual.SCSQPROC(CSQ45AQM) 中提供了样本。

   对要作为队列共享组成员的每个队列管理器执行此功能。

   注:

   a. 队列管理器只能是一个队列共享组的成员。

   b. 必须运行 RRS 才能使用队列共享组。

相关概念

第 752 页的『定制系统参数模块』

IBM MQ 系统参数模块控制 IBM MQ 在其操作中使用的日志记录，归档，跟踪和连接环境。 提供了缺省模块。 您应该创建自己的系统参数模块，因为某些参数 (例如数据集名称) 通常是特定于站点的。

z/OS 为队列共享组实施 *ESM* 安全控制

对队列共享组中的所有队列管理器实施安全性控制，以访问 Db2 和耦合设施列表结构。

- 对队列共享组中的每个 *IBM MQ* 队列管理器重复此任务。
- 从先前版本进行迁移时，可能需要执行此任务。

确保与队列管理器，通道启动程序和实用程序相关联的用户标识有权与要与之建立连接的每个 Db2 子系统建立 RRSAF 连接。 队列管理器和通道启动程序的用户标识是运行启动式任务过程的用户标识。

实用程序的用户标识是可提交批处理作业的用户标识。 用户标识需要读访问权的 RACF 概要文件在 DSNR 资源类中为 `Db2ssid.RRSAF`

需要授予与队列共享组中每个队列管理器相关联的用户标识对耦合设施列表结构的相应访问级别。 RACF 类为 FACILITY。

以下用户标识需要 ALTER 访问权:

- `IXLSTR.structure-name` 概要文件的队列管理器标识
- 运行 CSQ5PQSG 的用户标识

相关概念

第 746 页的『实施 ESM 安全控制』

对队列管理器和通道启动程序实施安全控制。

z/OS 配置 Advanced Message Security for z/OS

使用这些主题作为配置 Advanced Message Security (AMS) 的逐步指南。

开始之前

在开始配置 AMS 之前，请确保已执行以下队列管理器配置步骤:

1. 将 CSQ0DRTM 模块添加到 LPA，如 第 736 页的『更新 z/OS 链接列表和 LPA』中所述。
2. 将 CSQ0DSRV 的条目添加到 z/OS 程序属性表 (PPT)，如 第 740 页的『更新 z/OS 程序属性表』中所述。
3. 将 CSQ4INSM 成员包含在队列管理器启动式任务过程的 CSQINP2 并置中，如 第 747 页的『定制初始化输入数据集』中所述。
4. V 9.2.0

   使用 AMSPROD 属性启用 AMS。 请参阅 IBM MQ for z/OS 产品的产品使用记录以获取更多详细信息。

下一步做什么

配置受 AMS 保护的队列的策略。 管理 Advanced Message Security 安全策略中描述了安全策略。

z/OS 上的示例配置中有 AMS 配置的示例。

z/OS 为 *Advanced Message Security* 创建过程

要配置为使用 Advanced Message Security (AMS) 的每个 IBM MQ 子系统都需要一个编目过程来启动 AMS 地址空间。 您可以创建自己的或使用 IBM 提供的过程库。

过程

1. 将样本启动式任务过程 *thlqual*.SCSQPROC (CSQ4AMSM) 复制到 SYS1.PROCLIB 或 (如果未使用 SYS1.PROCLIB，过程库。 将过程命名为 xxxxAMSM，其中 xxxx 是 IBM MQ 子系统的名称。 例如，CSQ1AMSM 将是队列管理器 CSQ1 的 AMS 启动式任务过程。
2. 为要使用的每个 IBM MQ 子系统创建一个副本。
3. 使用样本过程 CSQ4AMSM 中的指示信息，根据您的需求定制过程。 您还可以在 JCL 中使用符号参数，以允许在过程启动时对其进行修改。
4. 查看并 (可选) 更改使用 Language Environment ® _CEE_ENVFILE 文件传递到 AMS 任务的参数。 样本 thlqual.SCSQPROC(CSQ40ENV) 列出了受支持的参数。
5. 对每个 IBM MQ 队列管理器重复步骤 1 到 4。

下一步做什么

第 781 页的『设置 Advanced Message Security 启动式任务用户标识』

z/OS 设置 *Advanced Message Security* 启动式任务用户标识

Advanced Message Security (AMS) 任务需要允许其称为 z/OS UNIX System Services (z/OS UNIX) 进程的用户标识。

关于此任务

此外，该任务所代表的用户还必须具有 UNIX UID (用户标识) 和 GID (组标识) 的相应定义，因此这些用户称为 z/OS UNIX System Services 用户。 有关定义 z/OS UNIX System Services UID 和 GID 的更多信息，请参阅 *z/OS: Security Server RACF Security Administrator 's Guide*。

z/OS: z/OS UNIX System Services 规划 将传统 UNIX 安全性与 z/OS 安全性进行比较。 传统 UNIX 安全性与 z/OS 安全性之间的主要区别在于内核服务支持两个级别的相应特权: UNIX 级别和 z/OS UNIX 级别。

根据安装的安全策略， Advanced Message Security 任务可以使用超级用户权限 (uid (0)) 运行，也可以使用其 RACF 身份允许 RACF FACILITY 类 BPX.DAEMON 和 BPX.SERVER 概要文件，因为此任务必须能够采用其用户的 RACF 身份。

如果使用后一种方法，或者您已激活 BPX.DAEMON 或 BPX.SERVER 概要文件， Advanced Message Security 任务程序 (thlqual.SCSQAUTH(CSQ0DSRV)) 必须位于 RACF 程序控制的库中。

查看 *z/OS: z/OS UNIX System Services* 规划 以确保您了解传统 UNIX 安全性与 z/OS UNIX 安全性之间的安全性差异。 这允许您根据安装的安全策略来管理 Advanced Message Security 任务，以部署和运行特权 z/OS UNIX System Services 进程。

供参考，对本次审查有用的出版物有:

- *z/OS: z/OS UNIX System Services* 规划
- *z/OS: Security Server RACF Security Administrator 's Guide*

注: 请仔细选择此任务的用户标识，因为 Advanced Message Security 收件人证书将装入到与此用户标识关联的密钥环中。 在 z/OS 上使用证书 中讨论了此注意事项。

此处显示的步骤描述了如何设置 Advanced Message Security 启动式任务用户。 这些步骤使用 RACF 命令作为示例。 如果您正在使用其他安全管理器，那么应使用等效命令。

注: 本节中的示例假定您已为 RACF STARTED， FACILITY 和 SURROGAT 类和通用概要文件检查激活通用概要文件命令处理。 有关 RACF 如何处理通用概要文件的更多信息，请参阅 *z/OS: Security Server RACF Command Language Reference*。

过程

1. 将 Advanced Message Security 启动式任务用户定义到 RACF。 本节中的示例使用用户标识 WMQAMSM。

   ```
   ADDUSER WMQAMSM NAME('AMS user') OMVS (UID(0)) DFLTGRP(group)
   ```

   根据您的安装标准选择缺省 "组"。

   注: 如果您不希望授予 z/OS UNIX 超级用户权限 (UID (0))，那么必须允许将 Advanced Message Security 用户标识授予 BPX.DAEMON 和 BPX.SERVER 设施类概要文件:

   ```
   PERMIT BPX.DAEMON CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
   ```

   并且 Advanced Message Security 任务程序 (*thlqual*.SCSQAUTH (CSQ0DSRV)) 必须位于 RACF 程序控制的库中。

   要使 SCSQAUTH 库程序受控，可以使用以下命令:

   ```
   RALTER PROGRAM * ADDMEM('thlqual.SCSQAUTH'//NOPADCHK) -or-
   RALTER PROGRAM ** ADDMEM('thlqual.SCSQAUTH'//NOPADCHK)
   SETROPTS WHEN(PROGRAM) REFRESH
   ```

   您还必须对 Advanced Message Security 任务所使用的本地语言库 (*thlqual*.SCSQANLx) 启用程序控制。
2. 确定 RACF STARTED 类是否处于活动状态。 如果不是，请激活 RACF STARTED 类:

   ```
   SETROPTS CLASSACT(STARTED)
   ```
3. 为 Advanced Message Security 任务定义已启动的类概要文件，并指定您在步骤 1 中选择或创建的用户标识:

   ```
   RDEFINE STARTED qmgrAMSM.* STDATA(USER(WMQAMSM))
   ```

   其中 *qmgr* 是启动式任务名称的前缀。 例如，启动式任务可以命名为 CSQ1AMSM。 在这种情况下，可以将 *qmgr*AMSM.* 替换为 CSQ1AMSM.*。

   AMS 启动式任务必须命名为 *qmgr*AMSM。
4. 使用 **SETROPTS** RACF 命令刷新存储器中的 RACLISTed STARTED 类概要文件:

```
SETROPTS RACLIST(STARTED) REFRESH
```

5. Advanced Message Security 任务在 IBM MQ 消息的保护处理期间临时采用请求者的主机用户标识的身份。 因此，需要在 SURROGAT 类中为每个可以发出请求的用户标识定义概要文件。

   如果 RACF SURROGAT 类处于活动状态，那么定义单个通用概要文件将允许 Advanced Message Security 任务采用任何用户的身份。 如果 SURROGAT 类未处于活动状态，那么将忽略此检查。 *z/OS* 中描述了所需的 *SURROGAT* 概要文件*: z/OS UNIX System Services Planning*。

   要在 SURROGAT 类中定义概要文件:

   a) 使用 RACF SETROPTS 命令激活 RACF SURROGAT 类:

   ```
   SETROPTS CLASSACT(SURROGAT)
   ```

   b) 激活 RACF SURROGAT 类的通用概要文件处理:

   ```
   SETROPTS GENERIC(SURROGAT)
   ```

   c) 激活 RACF SURROGAT 类的通用概要文件命令处理:

   ```
   SETROPTS GENCMD(SURROGAT)
   ```

   d) 在 SURROGAT 类中定义通用概要文件:

   ```
   RDEFINE SURROGAT BPX.SRV.* UACC(NONE)
   ```

   e) 允许 Advanced Message Security 用户标识使用通用 SURROGAT 类概要文件:

   ```
   PERMIT BPX.SRV.* CLASS(SURROGAT) ID(WMQAMSM) ACCESS(READ)
   ```

   注: 如果要将特定用户限制为由 Advanced Message Security 任务处理，那么可以定义更具体的概要文件，如 *z/OS* 中所述*: z/OS UNIX System Services* 规划。

   例如，名为 BPX.SRV.MQUSER1 的概要文件控制 AMS 任务是否可以采用用户标识 MQUSER1 的身份。

   f) 允许 Advanced Message Security 用户标识使用 BPX.SERVER 工具 (如果尚未在 创建证书和密钥环 中完成):

   ```
   PERMIT BPX.SERVER CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
   ```

   g) 使用 **SETROPTS** RACF 命令来刷新存储器中 RACLISTed 启动的类概要文件:

   ```
   SETROPTS RACLIST(SURROGAT) REFRESH
   SETROPTS RACLIST(FACILITY) REFRESH
   ```

6. Advanced Message Security 任务使用 z/OS System SSL 服务提供的工具来打开 SAF 管理的密钥环。 访问密钥环内容的底层系统授权工具 (SAF) 由 RACF 或同等安全管理器控制。

   此服务是 IRRSDL00 (R_datalib) 可调用服务。 此可调用服务使用用于保护对 RACF FACILITY 类定义的 RACF RACDCERT 命令的相同概要文件进行保护。 因此，必须使用以下命令允许 Advanced Message Security 用户标识用于概要文件:

   a) 如果尚未这样做，请将 RACF 通用概要文件定义到保护 RACDCERT 命令和 IRRSDL00 可调用服务的 RACF FACILITY 类:

   ```
   RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)
   SETROPTS RACLIST(FACILITY) REFRESH
   ```

   b) 将对启动式任务用户标识的权限授予 RACF 通用概要文件:

   ```
   PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(WMQAMSM) ACC(READ)
   ```

   或者，您可以授予对数据服务任务用户在 RDATALIB 类中的密钥环的 READ 访问权，如下所示:

   ```
   PERMIT WMQASMD.DRQ.AMS.KEYRING.LST CLASS(RDATALIB) ID(WMQAMSM) ACC(READ)
   ```

7. 配置资源安全性:

a) Advanced Message Security 启动式任务用户需要权限才能作为批处理应用程序连接到队列管理器。

如果队列管理器已启用连接安全性，请使用以下命令授予 AMS 任务权限以连接到队列管理器:

```
PERMIT hlq.BATCH CLASS(MQCONN) ID(WMQAMSM) ACC(READ)
```

其中 *hlq* 可以是队列管理器名称队列共享组名。

有关更多信息，请参阅 批处理连接的连接安全概要文件。

b) Advanced Message Security 启动式任务用户需要具有浏览 SYSTEM.PROTECTION.POLICY.QUEUE。

如果队列安全性在队列管理器上处于活动状态，请使用以下命令授予 AMS 用户访问队列的权限:

```
RDEFINE MQQUEUE hlq.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
PERMIT hlq.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE) ID(WMQAMSM) ACCESS(READ)
```

其中 *hlq* 可以是队列管理器名称队列共享组名。

如果队列管理器正在使用混合大小写概要文件，请改为在 MXQUEUE 类中定义概要文件。

要使用 CSQ0UTIL 实用程序管理 AMS 安全策略，管理员需要具有将消息放入 SYSTEM.PROTECTION.POLICY.QUEUE。 这是通过授予对保护队列的概要文件的 UPDATE 访问权来执行的。

有关更多信息，请参阅 队列安全性的概要文件。

下一步做什么

第 784 页的『将 RACDCERT 许可权授予 Advanced Message Security 的安全性管理员』

z/OS 将 *RACDCERT* 许可权授予 *Advanced Message Security* 的安全性管理员

Advanced Message Security 安全性管理员需要权限才能使用 RACDCERT 命令来创建和管理数字证书。

过程

- 标识此角色的相应用户标识，并授予使用 RACDCERT 命令的许可权。 例如:

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(admin) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

其中， `admin` 是 Advanced Message Security 安全性管理员的用户标识。

下一步做什么

第 784 页的『授予用户对 Advanced Message Security 的资源许可权』

z/OS 授予用户对 *Advanced Message Security* 的资源许可权

Advanced Message Security 用户需要相关资源许可权。

关于此任务

Advanced Message Security 用户 (即正在放入或获取 Advanced Message Security 受保护消息的用户) 需要:

- 与用户标识关联的 OMVS 段
- IRR.DIGTCERT.LISTRING 或 RDATALIB
- ICSF 类 CSFSERV 和 CSFKEYS 概要文件的许可权
- 用于放入 SYSTEM.PROTECTION.ERROR.QUEUE

Advanced Message Security 任务临时采用其客户机的身份; 即，在将 IBM MQ 消息处理到受 Advanced Message Security 保护的队列期间，该任务充当 Advanced Message Security 用户的 z/OS 用户标识的代理。

为了使任务具有用户的 z/OS 身份，客户机 z/OS 用户标识必须具有与其用户概要文件相关联的已定义 OMVS 段。

作为管理辅助， RACF 提供了定义可能与 RACF 用户和组概要文件相关联的缺省 OMVS 段的能力。 如果 z/OS 用户标识或组概要文件未显式定义 OMVS 段，那么将使用此缺省值。 如果计划有大量用户使用 Advanced Message Security，那么可以选择使用此缺省值，而不是为每个用户显式定义 OMVS 段。

z/OS: Security Server RACF Security Administrator 's Guide 包含定义缺省 OMVS 段的详细过程。 请查看本出版物中概述的过程，以确定 RACF 用户和组概要文件中缺省 OMVS 段的定义是否适合您的安装。

过程

1. 将 READ 许可权授予 IRR.DIGTCERT.LISTRING 概要文件:
   - 授予对 IRR.DIGTCERT.LISTRING 概要文件向所有用户发出以下命令:

     ```
     RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(READ)
     ```

   - 授予对 IRR.DIGTCERT.LISTRING 概要文件发出以下命令:

     ```
     PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACCESS(READ)
     ```

     其中， userid 是 Advanced Message Security 用户的名称。
   - 或者，使用 RDATALIB 类来授予对特定密钥环的访问权。 RDATALIB 许可权优先于 IRR.DIGTCERT.LISTRING 许可权。 例如：

     ```
     PERMIT user.DRQ.AMS.KEYRING.LST CLASS(RDATALIB) ID(user) ACC(READ)
     ```

2. 如果您正在使用 ICSF 管理的证书和专用密钥，那么 Advanced Message Security 用户需要访问某些类 CSFSERV 和 CSFKEYS 概要文件。 下表详细介绍了此访问权:

表 *81:* 对类 *CSFSERV* 和 *CSFKEYS* 概要文件的必需用户访问权		
类	**概要文件**	**许可权**
CSFSERV	CSFDSG	读取
CSFSERV	CSFPKE	读取
CSFSERV	CSFPKD	读取
CSFSERV	CSFDSV	读取
CSFKEYS	ICSF PKDS 标签	读取

3. 在定义了 AMS 策略的队列上执行操作的应用程序需要具有将消息放入 SYSTEM.PROTECTION.ERROR.QUEUE。 使用以下命令授予对队列的放置访问权:

   ```
   RDEFINE MQQUEUE hlq.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
   PERMIT hlq.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE) ID(userId) ACCESS(UPDATE)
   ```

   其中 *hlq* 可以是队列管理器名称队列共享组名，而 *userID* 是应用程序用户标识。

下一步做什么

第 785 页的『为 Advanced Message Security 创建密钥环』

z/OS 为 *Advanced Message Security* 创建密钥环

Advanced Message Security (AMS) 用于签名和加密的证书存储在 z/OS SAF 密钥环中。 您需要先创建这些密钥环和证书，然后才能使用 AMS。

关于此任务

Advanced Message Security 访问以下密钥环中的证书:

- AMS 地址空间用户拥有的单个密钥环。
- 由个别用户拥有的密钥环，这些用户在定义了 AMS 策略的队列上发送或接收消息。

这些密钥环必须全部命名为 `drq.ams.keyring`。

在 z/OS 上使用证书中提供了有关 AMS 所使用的密钥环和证书的更多信息以及示例方案。

执行以下步骤以创建 AMS 所需的密钥环，并将证书连接到密钥环。 在启动 AMS 之前，必须创建 AMS 地址空间用户拥有的密钥环。 您可以创建由随时发送或接收消息的用户拥有的密钥环。

过程

1. 发出以下命令以创建 AMS 地址空间用户拥有的密钥环:

   ```
   RACDCERT ID(amsUser) ADDRING(drq.ams.keyring)
   ```

   其中 *amsUser* 是 AMS 地址空间的用户标识。
2. 通过在步骤 1 中针对每个用户标识发出命令，为发送或接收受 AMS 保护的消息的每个用户创建密钥环。
3. 将用户证书颁发者的认证中心 (CA) 证书连接到 AMS 地址空间用户标识所拥有的密钥环。 发出以下命令:

   ```
   RACDCERT ID(amsUser) CONNECT(CERTAUTH LABEL('caLabel') RING(drq.ams.keyring))
   ```

   其中 *amsUser* 是 AMS 地址空间的用户标识， *caLabel* 是 CA 证书的标签。

   如果使用 RACF 作为 CA，并且需要创建认证中心证书，请遵循 定义本地认证中心证书中的示例。
4. 如果要使用隐私或机密性安全策略对受 AMS 保护的队列上的消息进行加密，请将消息接收方的证书连接到 AMS 地址空间用户标识所拥有的密钥环。 发出以下命令:

   ```
   RACDCERT ID(amsUser) CONNECT(ID(userId) LABEL('certLabel')
   RING(drq.ams.keyring) USAGE(SITE))
   ```

   其中 *amsUser* 是 AMS 地址空间的用户标识， *userId* 是消息收件人， *certLabel* 是用户证书的标签。

   USAGE(SITE) 属性阻止在密钥环中访问专用密钥。

   如果要使用 RACF 创建自己的证书，请遵循 使用专用密钥创建数字证书 中的示例来创建证书。
5. 将发送或接收受 AMS 保护的消息的每个用户的证书连接到该用户拥有的密钥环。 该证书必须作为密钥环中的缺省证书进行连接。 发出以下命令:

   ```
   RACDCERT ID(userId) CONNECT(ID(userId) LABEL('certLabel')
   RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
   ```

   其中 *userId* 是正在发送或接收消息的用户， *certLabel* 是用户证书的标签。

   注: 如果应用程序仅为输出打开队列并将消息发送到受 AMS 机密性策略保护的队列，那么不需要步骤 第 786 页的『2』 和 第 786 页的『5』 。

下一步做什么

第 786 页的『启用 Advanced Message Security』

z/OS 启用 *Advanced Message Security*

队列管理器的安全策略功能由系统参数模块中的 SPLCAP 参数控制。

关于此任务

遵循以下步骤对单个队列管理器启用 Advanced Message Security (AMS)。

此任务要求您对系统参数模块进行更改。 有关创建和定制系统参数模块的更多信息，请参阅 第 752 页的『定制系统参数模块』 。

过程

1. 在 CSQ6SYSP 中将 **SPLCAP** 设置为 YES 。 有关 CSQ6SYSP 宏的更多信息，请参阅 第 754 页的『使用 CSQ6SYSP』 。
2. V 9.2.0
   根据您的许可权利，将 **AMSPROD** 设置为 AMS， ADVANCED 或 ADVANCEDVUE。 请参阅 使用 CSQ6USGP ，以获取有关 CSQ6USGP 宏的更多信息。
3. 重新编译系统参数模块。
4. 使用更新后的系统参数模块重新启动队列管理器。 当队列管理器启动时，将自动启动 AMS 地址空间。

z/OS 配置 mqweb 服务器

使用这些主题作为配置 mqweb 服务器的逐步指南。

相关任务

第 674 页的『配置 IBM MQ Console 和 REST API』
为托管 IBM MQ Console 和 REST API 的 mqweb 服务器提供了缺省配置。 必须完成一些配置任务（例如，配置安全性以允许用户登录），才能使用这些组件。 本主题描述了所有可用的配置选项。

z/OS 创建 *mqweb* 服务器

如果安装了 IBM MQ for z/OS UNIX System Services Web Components，并且想要使用 MQ Console 或 REST API，那么需要创建和定制 mqweb 服务器。

开始之前

在运行 **crtmqweb** 脚本以创建 mqweb 服务器之前，必须设置 JAVA_HOME 环境变量以在系统上引用 64 位版本的 Java 。

您需要创建 SYSTEM.REST.REPLY.QUEUE 以使用 Liberty 服务器。 通过使用 第 747 页的『定制初始化输入数据集』中的最新 **CSQ4INSG** 样本来执行此操作。

注意: 启动 mqweb 服务器时，如果迁到错误消息 CWWKG0014E，如以下输出中所示:

```
     Launching mqweb (MQM MVS/ESA V9 R2.0/wlp...) (en_US)
               ÝAUDIT   ¨ CWWKE0001I: The server mqweb has been
launched.
               ÝWARNING ¨ CWWKF0009W: The server has not been configured to install any
features.
               ÝAUDIT   ¨ CWWKF0011I: The mqweb server is ready to run a smarter planet.
The mqweb server started in 6.348 seconds.
               ÝERROR   ¨ CWWKG0014E: The configuration parser detected an XML syntax
error while parsing the root of the configuration and the referenced configuration
documents.
                                Error: An invalid XML character (Unicode: 0x4c) was found
in the prolog of the document.
                                 File: file:<your filepath>/servers/mqweb/server.xml Line:
1 Column: 1
```

您应该检查 AUTOCVT 的 z/OS 设置 (自动将文件从一个代码集转换为另一个代码集)，并根据需要通过执行下列其中一项操作来调整值。

在 USS 终端中:

发出命令 :echo **$_BPXK_AUTOCVT** 以显示此环境变量的值。 如果未定义环境变量，那么不会显示任何值。

要设置环境变量，请参阅 _BPXK 环境变量。

系统范围:

显示 z/OS UNIX System Services (OMVS) 状态的示例 6 显示了如何在 BPXPRMxx 中显示系统范围 AUTOCVT 语句的值。

要在系统范围内设置环境变量，请使用 BPXPRMxx 中的 AUTOCVT 语句。

如果在 USS 终端中设置了环境变量 _BPXX_AUTOCVT ，那么它将覆盖 BPXPRMxx 中语句 AUTOCVT 的系统范围设置。

关于此任务

- 您需要对要运行 MQ Console 或 REST API 的每个 z/OS 系统执行一次此任务。
- 对于正在运行的每个版本的 IBM MQ，您都需要一个 mqweb 服务器。例如，如果您正在运行 IBM MQ 9.2.0， 9.1.5 和 9.1.0，那么需要三个不同的 mqweb 服务器。
- 从先前版本迁移时，可能需要刷新或修改服务器配置。

MQ Console 和 REST API 需要创建名为 mqweb 的单个 Liberty 服务器。

服务器配置和日志文件都存储在 Liberty 用户目录下。

V 9.2.0 需要使用将在其中运行的产品标识 (PID) 来配置 mqweb 服务器。这在创建时决定，并且必须是要用于运行 mqweb 服务器所连接的本地队列管理器的 PID。

注: 如果本地队列管理器在多个不同的 PID 下运行，请选择其中一个 PID 来运行 mqweb 服务器。

V 9.2.0 请参阅 随 IBM MQ for z/OS 产品一起记录产品使用情况，以获取有关 PID 及其在 z/OS 上的使用方式的更多信息。

V 9.2.0 在创建 mqweb 服务器之后，可以使用 setmqweb 命令来更改该服务器运行所使用的 PID。

执行以下过程以创建 mqweb 服务器:

过程

1. 决定 mqweb 服务器将在哪个 PID 下运行。
2. 为 Liberty 用户目录选择合适的位置。

   在其中运行 mqweb 服务器的用户标识需要对此用户目录及其内容的读写访问权。由于此用户目录将包含日志文件以及服务器配置，因此您应该在单独的文件系统中创建此目录。

   注: mqweb 服务器启动时存在大量磁盘 I/O。要减少启动 mqweb 服务器所花费的时间，请确保 IBM MQ 安装 z/OS UNIX 文件系统和 Liberty 用户目录文件系统都具有 sysplex-感知功能，或者在运行 mqweb 服务器的系统上本地安装。
3. 在 z/OS UNIX System Services 中，通过发出以下命令来确保当前目录为 `PathPrefix/web/bin`:

   ```
   cd PathPrefix/web/bin
   ```

   `PathPrefix` 是 IBM MQ for z/OS UNIX System Services Components 安装路径。
4. 通过运行 **crtmqweb** 脚本，创建包含模板 mqweb 服务器定义的 Liberty 用户目录。

   V 9.2.0 **crtmqweb** 命令的格式为:

   ```
   crtmqweb user_directory -p pid_value
   ```

   其中:

   user_directory
   是在步骤 第 788 页的『2』中决定的可选 Liberty 用户目录。如果未提供此参数，那么将使用缺省 Liberty 用户目录 `/var/mqm/web/installation1`。

   pid_value
   指示 mqweb 服务器将在其中运行的 PID。这取决于您在步骤 第 788 页的『1』中决定的 PID。

   如果选择 IBM MQ for z/OS (5655-MQ9)，那么 *pid_value* 为 MQ。

   如果选择了 IBM MQ for z/OS Value Unit Edition (VUE) (5655-VU9)，那么 *pid_value* 是 VUE。

   如果选择 IBM MQ Advanced for z/OS VUE (5655-AV1)，那么 *pid_value* 是 ADVANCEDVUE。

   例如，要使用 Liberty 用户目录 `/usr/mqweb` 和 PID IBM MQ Advanced for z/OS VUE (5655-AV1) 创建 mqweb 服务器，请运行以下命令:

   ```
   ./crtmqweb /usr/mqweb -p ADVANCEDVUE
   ```

5. 使用以下命令更改 Liberty 用户目录中目录和文件的所有权，以便它们属于 mqweb 服务器运行所使用的用户标识和组:

```
chown -R userid:group path
```

要授予组对路径的写访问权，请发出以下命令:

```
chmod -R 770 path
```

下一步做什么

第 789 页的『为 mqweb 服务器创建过程』

相关任务

第 674 页的『配置 IBM MQ Console 和 REST API』

为托管 IBM MQ Console 和 REST API 的 mqweb 服务器提供了缺省配置。 必须完成一些配置任务（例如，配置安全性以允许用户登录），才能使用这些组件。 本主题描述了所有可用的配置选项。

z/OS 为 *mqweb* 服务器创建过程

如果安装了 IBM MQ for z/OS UNIX System Services Web Components，并且想要使用 MQ Console 或 REST API，那么需要创建一个编目过程来启动 mqweb 服务器。 mqweb 服务器是托管 MQ Console 和 REST API 的 Liberty 服务器。

- 您需要对要运行 MQ Console 或 REST API 的每个 z/OS 系统执行一次此任务。
- 对于正在运行的每个版本的 IBM MQ，都需要一个 mqweb 服务器。 例如，对于位于 IBM MQ for z/OS 9.1.0 的队列管理器，名为 MQWB0910 的启动式任务，对于位于 IBM MQ for z/OS 9.0.5 的队列管理器，名为 MQWB0905 的启动式任务。

  如果 z/OS 系统上只有一个队列管理器，那么可以运行单个 Liberty 服务器启动式任务，并在迁移队列管理器时更改其使用的库。
- 从先前版本进行迁移时，可能需要修改编目过程。

执行以下过程以创建编目过程:

1. 将样本启动式任务过程 `thlqual.SCSQPROC(CSQ4WEBS)` 复制到过程库。

   根据企业的标准对过程进行命名。

   例如 `MQWB0910`，指示这是 IBM MQ for z/OS 9.1.0 mqweb 服务器的编目过程。
2. 使用样本过程 CSQ4WEBS 中的指示信息，根据您的需求定制该过程。

   请注意， Liberty 用户目录是运行 **crtmqweb** 脚本以创建 mqweb 服务器定义时指定的目录。

   请参阅 第 787 页的『创建 mqweb 服务器』 以了解详细信息。

   注: 确保在编辑成员时指定 **Caps off**，因为该文件具有小写数据。
3. 授权该过程在外部安全管理器下运行。
4. 使用 IBM 工作负载管理器 (WLM) 对此地址空间进行分类。

   mqweb 服务器是 IBM MQ 应用程序，用户与此应用程序进行交互。 应用程序在 WLM 中不需要具有很高的重要性，并且服务类 **STCUSER** 可能适用。

后续操作

遵循 第 675 页的『mqweb 服务器的基本配置』 中的步骤来完成配置 mqweb 服务器。

相关任务

第 674 页的『配置 IBM MQ Console 和 REST API』

为托管 IBM MQ Console 和 REST API 的 mqweb 服务器提供了缺省配置。 必须完成一些配置任务（例如，配置安全性以允许用户登录），才能使用这些组件。 本主题描述了所有可用的配置选项。

z/OS 在 z/OS 上测试队列管理器

定制或迁移队列管理器后，可以通过运行安装验证程序以及 IBM MQ for z/OS 随附的一些样本应用程序来对其进行测试。

关于此任务

安装并定制 IBM MQ for z/OS 后，可以使用提供的安装验证程序 CSQ4IVP1 来确认 IBM MQ for z/OS 可操作。

基本安装验证程序 CSQ4IVP1 测试非共享队列，并在不使用 C， COBOL 或 CICS 样本的情况下验证基本 IBM MQ 。

运行基本安装验证后，您可以使用具有不同队列的 CSQ4IVP1 来测试共享队列，还可以测试是否正确设置了 Db2 和耦合设施。 要确认分布式排队可操作，您可以使用提供的安装验证程序 CSQ4IVPX。

CSQ4IVP1 作为装入模块提供，并提供一组过程样本应用程序作为源模块，用于演示消息队列接口 (MQI) 的典型使用。 您可以使用这些源模块来测试不同的编程语言环境。 您可以使用提供的样本 JCL 来编译和链接编辑适合您安装的任何其他样本。

过程

- 有关如何在 z/OS 上测试队列管理器的信息，请参阅以下子主题:
  - 第 790 页的『运行基本安装验证程序』
  - 第 793 页的『测试队列共享组』
  - 第 794 页的『分布式排队测试』
  - 第 797 页的『使用 IBM MQ for z/OS 测试 C， C++， COBOL， PL/I 和 CICS 程序』

相关概念

IBM MQ for z/OS 概念

相关任务

在 z/OS 上规划 IBM MQ 环境

第 730 页的『在 z/OS 上配置队列管理器』

使用以下指示信息在 IBM MQ for z/OS 上配置队列管理器。

管理 IBM MQ for z/OS

z/OS 运行基本安装验证程序

安装并定制 IBM MQ 后，可以使用提供的安装验证程序 CSQ4IVP1 来确认 IBM MQ 可操作。

基本安装验证程序是批处理汇编程序 IVP，用于在不使用 C， COBOL 或 CICS 样本的情况下验证基本 IBM MQ 。

批处理汇编程序 IVP 由 SMP/E 进行链接编辑，装入模块在库 thlqual.SCSQLOAD 中提供。

在完成 SMP/E APPLY 步骤和定制步骤之后，运行 Batch Assembler IVP。

请参阅以下部分以获取更多详细信息:

- CSQ4IVP1 应用程序概述
- 准备运行 CSQ4IVP1
- 运行 CSQ4IVP1
- 检查 CSQ4IVP1 的结果

CSQ4IVP1 应用程序概述

CSQ4IVP1 是一个批处理应用程序，用于连接到 IBM MQ 子系统并执行以下基本功能:

- 问题 IBM MQ 调用
- 与命令服务器通信
- 验证触发是否处于活动状态
- 生成和删除动态队列
- 验证消息到期处理
- 验证消息落实处理

准备运行 CSQ4IVP1

运行 CSQ4IVP1:

1. 检查 IVP 条目是否位于队列管理器启动程序中的 CSQINP2 数据集并置中。 在成员 thlqual.SCSQPROC(CSQ4IVPQ) 中提供了 IVP 条目。 否则，将 thlqual.SCSQPROC(CSQ4IVPQ) 中提供的定义添加到 CSQINP2 并置。 如果队列管理器当前正在运行，那么需要将其重新启动以使这些定义生效。
2. 运行安装验证程序所需的样本 JCL CSQ4IVPR 位于库 thlqual.SCSQPROC 中。

   使用 IBM MQ 库的高级限定符，要使用的本地语言，四字符 IBM MQ 队列管理器名称以及作业输出的目标来定制 CSQ4IVPR JCL。
3. 更新 RACF 以允许 CSQ4IVP1 在 IBM MQ 安全性处于活动状态时访问其资源。

   要在启用 IBM MQ 安全性时运行 CSQ4IVP1 ，您需要具有访问对象的权限的 RACF 用户标识。 有关向 RACF 定义资源的详细信息，请参阅 在 z/OS 上设置安全性 。 运行 IVP 的用户标识必须具有以下访问权限:

权限	概要文件	类
读取	ssid.DISPLAY.PROCESS	MQCMDS
更新	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
更新	ssid.SYSTEM.COMMAND.REPLY.MODEL	MQQUEUE
更新	ssid.CSQ4IVP1.**	MQQUEUE
读取	ssid.BATCH	MQCONN

这些需求假定所有 IBM MQ 安全性都处于活动状态。 第 791 页的图 103 中显示了用于激活 IBM MQ 安全性的 RACF 命令。 此示例假定队列管理器名称为 CSQ1 ，并且运行样本 CSQ4IVP1 的人员的用户标识为 TS101。

```
RDEFINE MQCMDS CSQ1.DISPLAY.PROCESS
PERMIT CSQ1.DISPLAY.PROCESS CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.REPLY.MODEL
PERMIT CSQ1.SYSTEM.COMMAND.REPLY.MODEL CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.CSQ4IVP1.**
PERMIT CSQ1.CSQ4IVP1.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)
```

图 *103: RACF* 用于 *CSQ4IVP1*

运行 CSQ4IVP1

完成这些步骤后，启动队列管理器。 如果队列管理器已在运行，并且您已更改 CSQINP2，那么必须停止该队列管理器并将其重新启动。

IVP 作为批处理作业运行。 定制作业卡以满足安装的提交需求。

检查 CSQ4IVP1 的结果

IVP 分为 10 个阶段; 每个阶段必须在运行下一个阶段之前完成一个零完成代码。 IVP 生成报告，列出:

- 要连接到的队列管理器的名称。
- 显示完成代码和从每个阶段返回的原因码的单行消息。
- 一条单行参考消息 (如果适用)。

第 793 页的图 104 中提供了样本报告

z/OS 有关完成代码和原因码的说明，请参阅 IBM MQ for z/OS 消息，完成代码和原因码。

某些阶段具有多个 IBM MQ 调用，如果发生故障，将发出一条消息，指示返回故障的特定 IBM MQ 调用。 此外，对于某些阶段， IVP 会将解释和诊断信息放入注释字段中。

IVP 作业请求对某些队列管理器对象进行独占控制，因此应该通过系统进行单线程控制。 但是，可以对队列管理器运行 IVP 的次数没有限制。

每个阶段执行的功能包括:

第 1 阶段
通过发出 MQCONN API 调用来连接到队列管理器。

第 2 阶段
确定命令服务器用于检索请求消息的系统命令输入队列的名称。 此队列接收来自阶段 5 的显示请求。

为此，调用顺序为:

1. 发出 MQOPEN 调用并指定队列管理器名称，以打开队列管理器对象。
2. 发出 MQINQ 调用以找出系统命令输入队列的名称。
3. 发出 MQINQ 调用以了解有关各种队列管理器事件交换机的信息。
4. 发出 MQCLOSE 调用以关闭队列管理器对象。

成功完成此阶段后，系统命令输入队列的名称将显示在注释字段中。

第 3 阶段
使用 **MQOPEN** 调用打开启动队列。

在此阶段，将打开此队列以预测触发器消息，此消息是由于命令服务器应答来自阶段 5 的请求而到达的。 必须打开队列以进行输入以满足触发条件。

第 4 阶段
使用 CSQ4IVP1.MODEL 队列作为模型。 动态队列具有与从中创建该动态队列的模型相同的属性。 这意味着在将来自阶段 5 中的命令服务器请求的应答写入此队列时，会将触发器消息写入在阶段 3 中打开的启动队列。

成功完成此阶段后，将在注释字段中指示永久动态队列的名称。

第 5 阶段:
向命令服务器命令队列发出 MQPUT1 请求。

类型为 MQMT_REQUEST 的消息将写入系统命令输入队列，请求显示进程 CSQ4IVP1。 消息的消息描述符指定在阶段 4 中创建的永久动态队列作为命令服务器响应的应答队列。

阶段 6
从启动队列发出 **MQGET** 请求。 在此阶段，将针对在阶段 3 中打开的启动队列发出时间间隔为 1 分钟的 GET WAIT。 返回的消息应该是由正在写入应答队列的命令服务器的响应消息生成的触发器消息。

阶段 7
删除在阶段 4 中创建的永久动态队列。 由于队列上仍有消息，因此将使用 MQCO_PURGE_DELETE 选项。

阶段 8

1. 打开动态队列。
2. 设置了到期时间间隔的 MQPUT 消息。
3. 等待消息到期。
4. 尝试 MQGET 到期消息。
5. MQCLOSE 队列。

阶段 9

1. 打开动态队列。
2. MQPUT 消息。
3. 发出 MQCMIT 以落实当前工作单元。
4. MQGET 消息。
5. 发出 MQBACK 以回退消息。
6. MQGET 相同消息，并确保回退计数设置为 1。
7. 发出 MQCLOSE 以关闭队列。

阶段 10
使用 **MQDISC** 与队列管理器断开连接。

运行 IVP 后，可以删除不再需要的任何对象。

如果 IVP 未成功运行，请手动尝试每个步骤以找出哪个功能失败。

```
DATE : 2005.035                 IBM MQ for z/OS - V6            PAGE : 0001
INSTALLATION VERIFICATION PROGRAM
PARAMETERS ACCEPTED. PROGRAM WILL CONNECT TO : CSQ1
,OBJECT QUALIFER : CSQ4IVP1
INSTALLATION VERIFICATION BEGINS :
STAGE 01 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR BRIDGE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS EXCP FOR CHANNEL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR SSL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR INHIBITED EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR LOCAL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR PERFORMENCE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR REMOTE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR START/STOP EVENTS
STAGE 02 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 SYSTEM.COMMAND.INPUT
STAGE 03 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 04 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9810EFEAC8980
STAGE 05 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 06 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 07 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 08 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9810F0070E645
STAGE 09 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9812BA8706803
STAGE 10 COMPLETE. COMPCODE : 0000 REASON CODE : 0000>>>>>>>>>>   END OF REPORT   <<<<<<<<<<
```

图 *104:* 来自 *CSQ4IVP1* 的样本报告

z/OS 测试队列共享组

基本安装验证程序 CSQ4IVP1 测试非共享队列。

无论队列管理器是否是队列共享组的成员，都可以使用 CSQ4IVP1 。 运行基本 IVP 后，可以使用具有不同队列的 CSQ4IVP1 安装验证程序来测试共享队列。 此外，还会测试是否正确设置了 Db2 和耦合设施。

准备对队列共享组运行 CSQ4IVP1

运行 CSQ4IVP1:

1. 将 IVP 使用的耦合设施结构添加到 CFRM 策略数据集，如 第 779 页的『设置耦合设施』中所述。 提供的样本使用名为 APPLICATION1 的结构，但您可以根据需要更改此结构。
2. 检查 IVP 条目是否位于队列管理器启动程序中的 CSQINP2 数据集并置中。 在成员 thlqual.SCSQPROC(CSQ4IVPG) 中提供了 IVP 条目。 如果不是，请将 thlqual.SCSQPROC(CSQ4IVPG) 中提供的定义添加到 CSQINP2 并置。 如果队列管理器当前正在运行，那么需要将其重新启动以使这些定义生效。
3. 必要时更改 thlqual.SCSQPROC(CSQ4IVPG) 中使用的耦合设施结构的名称。
4. 运行队列共享组的安装验证程序所需的样本 JCL CSQ4IVPS 位于库 thlqual.SCSQPROC 中。

   使用 IBM MQ 库的高级限定符，要使用的本地语言，四字符 IBM MQ 队列管理器名称以及作业输出的目标来定制 CSQ4IVPS JCL。
5. 更新 RACF 以允许 CSQ4IVP1 在 IBM MQ 安全性处于活动状态时访问其资源。

   要在启用 IBM MQ 安全性时运行 CSQ4IVP1 ，您需要具有访问对象的权限的 RACF 用户标识。 有关向 RACF 定义资源的详细信息，请参阅 在 z/OS 上设置安全性 。 除了运行基本 IVP 所需的访问权限外，运行 IVP 的用户标识还必须具有以下访问权限:

权限	概要文件	类
更新	ssid.CSQ4IVPG.**	MQQUEUE

   这些需求假定所有 IBM MQ 安全性都处于活动状态。 第 794 页的图 105 中显示了用于激活 IBM MQ 安全性的 RACF 命令。 此示例假定队列管理器名称为 CSQ1 ，并且运行样本 CSQ4IVP1 的人员的用户标识为 TS101。

```
RDEFINE MQQUEUE CSQ1.CSQ4IVPG.**
PERMIT CSQ1.CSQ4IVPG.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)
```

图 *105:* 针对队列共享组的 *CSQ4IVP1* 的 *RACF* 命令

对队列共享组运行 CSQ4IVP1

完成这些步骤后，启动队列管理器。 如果队列管理器已在运行，并且您已更改 CSQINP2，那么必须停止该队列管理器并将其重新启动。

IVP 作为批处理作业运行。 定制作业卡以满足安装的提交需求。

检查队列共享组的 CSQ4IVP1 结果

队列共享组的 IVP 工作方式与基本 IVP 相同，但创建的队列称为 CSQIVPG。 *xx*。 遵循 第 792 页的『检查 CSQ4IVP1 的结果』中提供的指示信息来检查队列共享组的 IVP 结果。

z/OS 分布式排队测试

您可以使用提供的安装验证程序 CSQ4IVPX 来确认分布式排队可操作。

CSQ4IVPX 作业概述

CSQ4IVPX 是一个批处理作业，用于启动通道启动程序并发出 IBM MQ DISPLAY CHINIT 命令。 这验证了分布式排队的所有主要方面都是可操作的，同时避免了需要设置通道和网络定义。

准备运行 CSQ4IVPX

在运行 CSQ4IVPX:

1. 运行安装验证程序所需的样本 JCL CSQ4IVPX 位于库 thlqual.SCSQPROC 中。

使用 IBM MQ 库的高级限定符，要使用的本地语言，四字符队列管理器名称以及作业输出的目标来定制 CSQ4IVPX JCL。

2. 更新 RACF 以允许 CSQ4IVPX 在 IBM MQ 安全性处于活动状态时访问其资源。 要在启用 IBM MQ 安全性时运行 CSQ4IVPX，您需要具有访问对象的权限的 RACF 用户标识。 有关向 RACF 定义资源的详细信息，请参阅 在 z/OS 上设置安全性。 运行 IVP 的用户标识必须具有以下访问权限:

权限	概要文件	类
CONTROL	ssid.START.CHINIT 和 ssid.STOP.CHINIT	MQCMDS
更新	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
更新	ssid.SYSTEM.CSQUTIL.*	MQQUEUE
读取	ssid.BATCH	MQCONN
读取	ssid.DISPLAY.CHINIT	MQCMDS

这些需求假定已定义连接安全概要文件 ssid.CHIN (如 通道启动程序的连接安全概要文件 中所示)，并且所有 IBM MQ 安全性都处于活动状态。 第 796 页的图 106 中显示了用于执行此操作的 RACF 命令。 本示例假定:

- 队列管理器名称为 CSQ1
- 运行样本 CSQ4IVPX 的人员的用户标识为 TS101
- 通道启动程序地址空间正在用户标识 CSQ1MSTR 下运行

3. 更新 RACF 以允许通道启动程序地址空间具有以下访问权限:

权限	概要文件	类
读取	ssid.CHIN	MQCONN
更新	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
更新	ssid.SYSTEM.CHANNEL.INITQ	MQQUEUE
更新	ssid.SYSTEM.CHANNEL.SYNCQ	MQQUEUE
更改	ssid.SYSTEM.CLUSTER.COMMAND.QUEUE	MQQUEUE
更新	ssid.SYSTEM.CLUSTER.TRANSMIT.QUEUE	MQQUEUE
更改	ssid.SYSTEM.CLUSTER.REPOSITORY.QUEUE	MQQUEUE
CONTROL	ssid.CONTEXT.**	MQADMIN

第 796 页的图 106 中还显示了用于执行此操作的 RACF 命令。

```
RDEFINE MQCMDS CSQ1.DISPLAY.DQM
PERMIT CSQ1.DISPLAY.DQM CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQCMDS CSQ1.START.CHINIT
PERMIT CSQ1.START.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQCMDS CSQ1.STOP.CHINIT
PERMIT CSQ1.STOP.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101,CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CSQUTIL.*
PERMIT CSQ1.SYSTEM.CSQUTIL.* CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)

RDEFINE MQCONN CSQ1.CHIN
PERMIT CSQ1.CHIN CLASS(MQCONN) ID(CSQ1MSTR) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.SYNCQ
PERMIT CSQ1.SYSTEM.CHANNEL.SYNCQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.INITQ
PERMIT CSQ1.SYSTEM.CHANNEL.INITQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQADMIN CSQ1.CONTEXT.**
PERMIT CSQ1.CONTEXT.** CLASS(MQADMIN) ID(CSQ1MSTR) ACCESS(CONTROL)
```

图 *106: RACF* 用于 *CSQ4IVPX*

运行 CSQ4IVPX

完成这些步骤后，启动队列管理器。

IVP 作为批处理作业运行。 定制作业卡以满足安装的提交需求。

检查 CSQ4IVPX 的结果

CSQ4IVPX 运行 CSQUTIL IBM MQ 实用程序以发出三个 MQSC 命令。 SYSPRINT 输出数据集应该类似于 第 797 页的图 107，尽管详细信息可能因队列管理器属性而异。

- 您应该会看到每个命令 **(1)** 后跟多条消息。
- 每个命令的最后一条消息应该是 “CSQ9022I ... 正常完成” **(2)**。
- 整个作业应完成，返回码为零 **(3)**。

```
CSQU000I CSQUTIL IBM MQ for z/OS - V6
CSQU001I CSQUTIL Queue Manager Utility - 2005-05-09 09:06:48
COMMAND
CSQU127I CSQUTIL Executing COMMAND using input from CSQUCMD data set
CSQU120I CSQUTIL Connecting to queue manager CSQ1
CSQU121I CSQUTIL Connected to queue manager CSQ1
CSQU055I CSQUTIL Target queue manager is CSQ1
START CHINIT
(1)
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQM138I +CSQ1 CSQMSCHI CHANNEL INITIATOR STARTING
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' START CHINIT' NORMAL COMPLETION
(2)
DISPLAY CHINIT
(1)
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMDDQM DISPLAY CHINIT COMMAND ACCEPTED
CSQN205I  COUNT=   12, RETURN=00000000, REASON=00000000
CSQX830I +CSQ1 CSQXRDQM Channel initiator active
CSQX002I +CSQ1 CSQXRDQM Queue sharing group is QSG1
CSQX831I +CSQ1 CSQXRDQM 8 adapter subtasks started, 8 requested
CSQX832I +CSQ1 CSQXRDQM 5 dispatchers started, 5 requested
CSQX833I +CSQ1 CSQXRDQM 0 SSL server subtasks started, 0 requested
CSQX840I +CSQ1 CSQXRDQM 0 channel connections current, maximum 200
CSQX841I +CSQ1 CSQXRDQM 0 channel connections active, maximum 200,
including 0 paused
CSQX842I +CSQ1 CSQXRDQM 0 channel connections starting,
0 stopped, 0 retrying
CSQX836I +CSQ1 Maximum channels - TCP/IP 200, LU 6.2 200
CSQX845I +CSQ1 CSQXRDQM TCP/IP system name is TCPIP
CSQX848I +CSQ1 CSQXRDQM TCP/IP listener INDISP=QMGR not started
CSQX848I +CSQ1 CSQXRDQM TCP/IP listener INDISP=GROUP not started
CSQX849I +CSQ1 CSQXRDQM LU 6.2 listener INDISP=QMGR not started
CSQX849I +CSQ1 CSQXRDQM LU 6.2 listener INDISP=GROUP not started
CSQ9022I +CSQ1 CSQXCRPS ' DISPLAY CHINIT' NORMAL COMPLETION
(2)
STOP CHINIT
(1)
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMTCHI STOP CHINIT COMMAND ACCEPTED
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' STOP CHINIT' NORMAL COMPLETION
(2)
CSQU057I CSQUCMDS 3 commands read
CSQU058I CSQUCMDS 3 commands issued and responses received, 0 failed
CSQU143I CSQUTIL 1 COMMAND statements attempted
CSQU144I CSQUTIL 1 COMMAND statements executed successfully
CSQU148I CSQUTIL Utility completed, return code=0
(3)
```

图 *107: CSQ4IVPX* 的示例输出

z/OS 使用 IBM MQ for z/OS 测试 C， C++， COBOL， PL/I 和 CICS 程序

您可以使用随 IBM MQ 提供的样本应用程序来测试 C， C++， COBOL， PL/I 或 CICS。

IVP (CSQ4IVP1) 作为装入模块提供，并作为源模块提供样本。 您可以使用这些源模块来测试不同的编程语言环境。

有关样本应用程序的更多信息，请参阅 IBM MQ for z/OS 的样本程序 。

z/OS 在 z/OS 上设置与其他队列管理器的通信

本部分描述了在开始使用分布式排队之前需要进行的 IBM MQ for z/OS 准备工作。

关于此任务

要定义分布式排队需求，需要定义以下项:

- 通道启动程序过程和数据集

- 通道定义
- 队列和其他对象
- 访问安全性

如果您正在使用队列共享组，请参阅 分布式排队和队列共享组。

有关准备使用 IBM MQ for z/OS 设置分布式排队时要考虑的其他要点，请参阅 第 798 页的『在 z/OS 上使用分布式排队的注意事项』。

过程

要启用分布式排队，请完成以下步骤:

- 定制分布式排队设施并定义所需的 IBM MQ 对象，如 定义系统对象 和 第 731 页的『准备在 z/OS 上定制队列管理器』 中所述。
- 定义访问安全性，如 z/OS 上通道启动程序的安全性注意事项中所述。
- 设置通信，如 第 815 页的『为 z/OS 设置通信』 中所述。

相关概念

第 734 页的『设置 IBM MQ for z/OS』
使用本主题作为定制 IBM MQ for z/OS 系统的逐步指南。

相关任务

第 152 页的『配置分布式队列』
此部分提供有关 IBM MQ 安装之间的相互通信的更详细信息，包括队列定义，通道定义，触发和同步点过程

z/OS 在 z/OS 上使用分布式排队的注意事项

准备在 z/OS 上使用分布式排队时要考虑的点。

如果您正在使用队列共享组，请参阅 分布式排队和队列共享组。

操作员消息

由于通道启动程序使用大量异步操作的分派器，因此操作员消息可能会在从时间序列中注销时出现。

通道操作命令

通道操作命令通常涉及两个阶段。 检查命令语法并验证通道是否存在时，将向通道启动程序发送请求。 向命令发布者发送消息 CSQM134I 或 CSQM137I 以指示第一阶段的完成。 当通道启动程序已处理该命令时，将向命令发布者发送指示其成功与否的更多消息以及消息 CSQ9022I 或 CSQ9023E。 生成的任何错误消息也可以发送到 z/OS 控制台。

但是，除 **DISPLAY CLUSQMGR** 以外的所有集群命令都以异步方式工作。 用于更改对象属性的命令将更新对象并向通道启动程序发送请求。 检查用于处理集群的命令的语法，并向通道启动程序发送请求。 在这两种情况下，都会向命令发布者发送消息 CSQM130I，指示已发送请求。 此消息后跟消息 CSQ9022I，以指示命令已成功完成，因为已发送请求。 这不指示集群请求已成功完成。 将异步处理发送到通道启动程序的请求以及从集群的其他成员接收的集群请求。 在某些情况下，必须将这些请求发送到整个集群以确定它们是否成功。 将向运行通道启动程序的系统上的 z/OS 报告任何错误。 它们不会发送到命令签发者。

未送达消息的队列

IBM MQ for z/OS 随附了死信处理程序。 有关更多信息，请参阅 死信队列处理程序实用程序 (CSQUDLQH)。

正在使用的队列

即使未传输消息，接收方通道的 MCA 也可以使目标队列保持打开状态。 此行为会导致队列显示为 "正在使用中"。

安全性更改

如果更改用户标识的安全访问权，那么更改可能不会立即生效。 有关更多信息，请参阅 z/OS 上通道启动程序的安全性注意事项， 队列安全性的概要文件和 第 746 页的『实施 ESM 安全控制』 。

通信已停止-TCP

如果 TCP 由于某种原因而停止，然后重新启动，那么将停止在 TCP 端口上等待的 IBM MQ for z/OS TCP 侦听器。

自动通道重新连接允许通道启动程序检测到 TCP/IP 不可用，并在 TCP/IP 返回时自动重新启动 TCP/IP 侦听器。 此自动重新启动可减少操作人员注意到 TCP/IP 问题并手动重新启动侦听器的需求。 当侦听器未执行操作时，还可以使用通道启动程序以 LSTRTMR 指定的时间间隔重试该侦听器。 这些尝试可以继续，直到 TCP/IP 返回并且侦听器成功自动重新启动为止。 有关 LSTRTMR 的更多信息，请参阅 ALTER QMGR 和 分布式排队消息 (CSQX ...)。

通信已停止- LU6.2

如果 APPC 已停止，那么侦听器也将停止。 同样，在这种情况下，侦听器会在 LSTRTMR 时间间隔内自动重试，以便在 APPC 重新启动时，侦听器也可以重新启动。

如果 Db2 失败，那么已在运行的共享通道将继续运行，但任何新的通道启动请求都将失败。 当 Db2 复原时，新的请求能够完成。

z/OS 自动重新启动管理 (Automatic Restart Management ， ARM)

自动重新启动管理 (ARM) 是一种 z/OS 恢复功能，可提高特定批处理作业或启动式任务 (例如，子系统) 的可用性。 因此，可以更快地恢复生产性工作。

要使用 ARM ，必须以特定方式设置队列管理器和通道启动程序，以使它们自动重新启动。 有关更多信息，请参阅 使用 z/OS 自动重新启动管理器 (ARM)。

相关概念

第 734 页的『设置 IBM MQ for z/OS』
使用本主题作为定制 IBM MQ for z/OS 系统的逐步指南。

相关任务

第 152 页的『配置分布式队列』
此部分提供有关 IBM MQ 安装之间的相互通信的更详细信息，包括队列定义，通道定义，触发和同步点过程

z/OS 定义 IBM MQ 对象

使用其中一种 IBM MQ 命令输入方法来定义 IBM MQ 对象。 请参阅本主题中的信息，以获取有关定义这些对象的更多详细信息。

有关定义对象的信息，请参阅 第 800 页的『监视和控制 z/OS 上的通道』 。

传输队列和触发通道

定义以下内容:

- 每个发送消息通道使用 XMITQ 的本地队列。
- 远程队列定义。

  根据指定名称和内容的方式，远程队列对象有三种不同的用途:

  – 远程队列定义
  – 队列管理器别名定义
  – 应答队列别名定义

  使用远程队列定义对象的三种方法中显示了这三种方法。

使用传输队列上的 TRIGDATA 字段来触发指定的通道。 例如:

```
DEFINE QLOCAL(MYXMITQ) USAGE(XMITQ) TRIGGER +
INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(MYCHANNEL)
DEFINE CHL(MYCHANNEL) CHLTYPE(SDR) TRPTYPE(TCP) +
XMITQ(MYXMITQ) CONNAME('9.20.9.30(1555)')
```

提供的样本 CSQ4INYD 提供了必需定义的其他示例。

z/OS 与定义了共享通道的同步队列的 CF 结构的连接中断或类似问题可能会暂时阻止通道启动。 解决问题后，如果您正在使用触发器类型 FIRST，并且通道在触发时无法启动，那么必须手动启动通道。 如果要在问题解决后自动启动触发的通道，请考虑将队列管理器 TRIGINT 属性设置为缺省值以外的值。 将 TRIGINT 属性设置为非缺省值会导致通道启动程序在传输队列上有消息时定期重试启动通道。

同步队列

DQM 需要队列以与序号和逻辑工作单元标识 (LUWID) 配合使用。 必须确保名称为 SYSTEM.CHANNEL.SYNCQ (请参阅 在 z/OS 上规划)。 此队列必须可用，否则通道启动程序无法启动。

确保使用 INDXTYPE (MSGID) 定义此队列。 此属性可提高访问它们的速度。

通道命令队列

您需要确保名为 SYSTEM.CHANNEL.INITQ。

如果通道启动程序检测到 SYSTEM.CHANNEL.INITQ，在更正问题之前无法正常继续。 问题可能是下列其中一项:

- 队列已满
- 未对放置启用队列
- 队列上的页集已满
- 通道启动程序对队列没有正确的安全权限

如果在通道启动程序运行时将队列定义更改为 GET (DISABLED)，那么启动程序无法从队列获取消息并终止。

启动通道启动程序

使用通道启动程序实现触发。 在 IBM MQ for z/OS 上，将使用 MQSC 命令 START CHINIT 启动发起方。

停止通道启动程序

当您停止队列管理器时，将自动停止通道启动程序。 如果需要停止通道启动程序而不是队列管理器，那么可以使用 MQSC 命令 STOP CHINIT。

z/OS 监视和控制 z/OS 上的通道

使用 DQM 命令和面板来创建，监视和控制到远程队列管理器的通道。

每个 z/OS 队列管理器都有一个 DQM 程序 (通道启动程序) 用于控制与使用本机 z/OS 工具的远程队列管理器的相互连接。

z/OS 上这些面板和命令的实现集成到操作和控制面板以及 MQSC 命令中。 在这两组面板和命令的组织中没有做出区分。

您还可以使用可编程命令格式 (PCF) 命令输入命令。 有关使用这些命令的信息，请参阅 自动化管理任务 。

此部分中的信息适用于通道启动程序用于分布式排队的所有情况。 无论您使用的是队列共享组还是组内排队，都适用。

DQM 通道控制功能

有关分布式队列管理模型的概述，请参阅 第 171 页的『消息发送和接收』。

通道控制功能由面板，命令和程序，两个同步队列，通道命令队列和通道定义组成。 本主题简要描述了通道控制功能的组件。

- 通道定义作为对象保存在页集 0 或 Db2 中，与 z/OS 中的其他 IBM MQ 对象一样。
- 您可以使用操作和控制面板， MQSC 命令或 PCF 命令来执行以下操作:
  - 创建，复制，显示，改变和删除通道定义
  - 启动和停止通道启动程序和侦听器
  - 启动，停止和 ping 通道，重置通道序号以及在无法重新建立链接时解析不确定消息
  - 显示有关通道的状态信息
  - 显示有关 DQM 的信息

  尤其是，您可以使用 CSQINPX 初始化输入数据集来发出 MQSC 命令。 每次启动通道启动程序时，都可以处理此集合。 有关更多信息，请参阅 初始化命令。
- 有两个队列 (SYSTEM.CHANNEL.SYNCQ 和 SYSTEM.QSG.CHANNEL.SYNCQ) 用于通道再同步目的。 由于性能原因，请使用 INDXTYPE (MSGID) 定义这些队列。
- 通道命令队列 (SYSTEM.CHANNEL.INITQ) 用于保存通道启动程序，通道和侦听器的命令。
- 通道控制功能程序在其自己的地址空间中运行，独立于队列管理器，并包含通道启动程序，侦听器，MCA ，触发器监视器和命令处理程序。
- 对于队列共享组和共享通道，请参阅 共享队列和队列共享组。
- 对于组内排队，请参阅 组内排队

在 z/OS 上管理通道

使用下表中的链接以获取有关如何管理通道，通道发起程序和侦听器的信息:

表 *82:* 通道任务

要执行的任务	**MQSC 命令**
定义通道	DEFINE CHANNEL
变更通道定义	ALTER CHANNEL
显示通道定义	显示通道
删除通道定义	删除通道
启动通道启动程序	START CHINIT
停止通道启动程序	STOP CHINIT
显示通道启动程序信息	显示中国
启动通道侦听器	启动侦听器
停止通道侦听器	停止侦听器
启动通道	启动通道
测试通道	Ping 通道
重置通道的消息序号	重置通道
解析通道上的不确定消息	解析通道
停止通道	停止通道
显示通道状态	DISPLAY CHSTATUS
显示集群通道	DISPLAY CLUSQMGR

相关概念

第 802 页的『使用面板和命令』
您可以使用 MQSC 命令， PCF 命令或操作和控制面板来管理 DQM。

第 734 页的『设置 IBM MQ for z/OS』
使用本主题作为定制 IBM MQ for z/OS 系统的逐步指南。

第 815 页的『为 z/OS 设置通信』
当启动分布式排队管理通道时，它会尝试使用通道定义中指定的连接。 要成功，需要定义连接并使其可用。 本节说明如何定义连接。

第 819 页的『为具有队列共享组的 DQM 准备 IBM MQ for z/OS』
使用本节中的指示信息在 IBM MQ for z/OS 上配置具有队列共享组的分布式排队。

第 822 页的『使用队列共享组为 IBM MQ for z/OS 设置通信』
当分布式排队管理通道启动时，它会尝试使用通道定义中指定的连接。 要使此尝试成功，需要定义连接并使其可用。

相关任务

第 152 页的『配置分布式队列』
此部分提供有关 IBM MQ 安装之间的相互通信的更详细信息，包括队列定义，通道定义，触发和同步点过程

第 797 页的『在 z/OS 上设置与其他队列管理器的通信』
本部分描述了在开始使用分布式排队之前需要进行的 IBM MQ for z/OS 准备工作。

z/OS 使用面板和命令

您可以使用 MQSC 命令， PCF 命令或操作和控制面板来管理 DQM。

有关 MQSC 命令的信息，请参阅 使用 MQSC 命令进行管理。 有关 PCF 命令的信息，请参阅 使用可编程命令格式命令自动管理。

使用初始面板

有关使用功能键调用操作和控制面板以及获取帮助的简介，请参阅 管理 IBM MQ for z/OS。

注: 要使用操作和控制面板，您必须具有正确的安全权限; 请参阅 管理 IBM MQ for z/OS 和子主题以获取更多信息。 第 802 页的图 108 显示了启动面板会话时显示的面板。 面板后面的文本说明了您在此面板中执行的操作。

```
IBM MQ for z/OS - Main Menu

Complete fields. Then press Enter.

Action . . . . . . . . . . . 1   0. List with filter  4. Manage
1. List or Display  5. Perform
2. Define like    6. Start
3. Alter       7. Stop
8. Command
Object type . . . . . . . . CHANNEL    +
Name . . . . . . . . . . . . *
Disposition . . . . . . . . A Q=Qmgr, C=Copy, P=Private, G=Group,
S=Shared, A=All

Connect name . . . . . . . MQ25 - local queue manager or group
Target queue manager . . . MQ25
- connected or remote queue manager for command input
Action queue manager . . . MQ25 - command scope in group
Response wait time . . . . 10  5 - 999 seconds

(C) Copyright IBM Corporation 1993, 2025. All rights reserved.

Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F4=Prompt  F9=SwapNext F10=Messages
F12=Cancel
```

图 *108:* 操作和控制初始面板

在此面板中，您可以执行以下操作:

- 通过在 **操作** 字段中输入相应的数字，选择要执行的操作。
- 指定要使用的对象类型。 如果您不确定对象类型的列表，请按 F4 以获取这些对象类型的列表。
- 显示指定类型的对象列表。 在 **名称** 字段中输入星号 (*)，然后按 Enter 键以显示已在此子系统上定义的对象 (指定类型) 的列表。 然后，可以按顺序选择一个或多个要使用的对象。 第 803 页的图 109 显示了以此方式生成的通道的列表。
- 在 **处置** 字段中指定要使用的对象的队列共享组中的处置。 处置确定保留对象的位置以及对象的行为方式。
- 在 **连接名称** 字段中选择要连接的本地队列管理器或队列共享组。 如果希望在远程队列管理器上发出命令，请选择 **目标队列管理器** 字段或 **操作队列管理器** 字段，具体取决于远程队列管理器是否是队列共享组的成员。 如果远程队列管理器不是队列共享组的成员，请选择 **目标队列管理器** 字段。 如果远程队列管理器是队列共享组的成员，请选择 **操作队列管理器** 字段。
- 在 **响应等待时间** 字段中选择要接收的响应的等待时间。

```
List Channels - MQ25          Row 1 of 8

Type action codes, then press Enter. Press F11 to display connection status.
1=Display  2=Define like  3=Alter  4=Manage  5=Perform
6=Start   7=Stop

Name         Type     Disposition  Status
<> *           CHANNEL    ALL   MQ25
_  SYSTEM.DEF.CLNTCONN  CLNTCONN   QMGR  MQ25
_  SYSTEM.DEF.CLUSRCVR  CLUSRCVR   QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.CLUSSDR  CLUSSDR    QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.RECEIVER  RECEIVER   QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.REQUESTER REQUESTER   QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.SENDER   SENDER    QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.SERVER   SERVER    QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.SVRCONN  SVRCONN    QMGR  MQ25 INACTIVE
******** End of list ********



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F4=Filter  F5=Refresh  F7=Bkwd
F8=Fwd    F9=SwapNext F10=Messages F11=Status  F12=Cancel
```

图 *109:* 列出通道

z/OS 在 *z/OS* 上定义通道

在 z/OS 上，可以使用 MQSC 命令或使用操作和控制面板来定义通道。

要使用 MQSC 命令定义通道，请使用 DEFINE CHANNEL。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	2 (定义相似项)
对象类型	通道类型 (例如 SENDER) 或 CHANNEL
名称	
处置	新对象的位置。

将向您显示一些面板，以完成有关您要定义的通道的名称和属性的信息。 它们使用缺省属性值进行初始化。 在按 Enter 键之前更改您想要的任何项。

注: 如果在 **object type** 字段中输入了 CHANNEL，那么将首先显示 "选择有效通道类型" 面板。

如果要定义与现有通道具有相同属性的通道，请将要复制的通道的名称放在初始面板上的 **Name** 字段中。 面板使用现有对象的属性进行初始化。

有关通道属性的信息，请参阅 通道属性

注:

1. 对网络中的所有通道进行唯一命名。 如 显示所有通道的网络图中所示，包括通道名称中的源队列管理器名称和目标队列管理器名称是执行此命名的好方法。

定义通道后，必须保护通道，请参阅 第 805 页的『保护通道』

z/OS 更改通道定义

您可以使用 MQSC 命令或使用操作和控制面板来变更通道定义。

要使用 MQSC 命令变更通道定义，请使用 ALTER CHANNEL。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	3 (变更)
对象类型	通道类型 (例如 SENDER) 或 CHANNEL
名称	CHANNEL.TO.ALTER
处置	存储对象的位置。

将向您显示一些面板，其中包含有关通道当前属性的信息。 通过过度输入新值来更改您想要的任何不受保护的字段，然后按 Enter 键以更改通道定义。

有关通道属性的信息，请参阅 通道属性。

z/OS 显示通道定义

您可以使用 MQSC 命令或使用操作和控制面板来显示通道定义。

要使用 MQSC 命令显示通道定义，请使用 DISPLAY CHANNEL。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	1 (列表或显示)
对象类型	通道类型 (例如 SENDER) 或 CHANNEL
名称	CHANNEL.TO.DISPLAY
处置	对象的位置。

将向您显示一些面板，其中显示有关通道当前属性的信息。

有关通道属性的信息，请参阅 通道属性。

z/OS 删除通道定义

您可以使用 MQSC 命令或使用操作和控制面板来删除通道定义。

要使用 MQSC 命令删除通道定义，请使用 DELETE CHANNEL。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	4 (管理)
对象类型	通道类型 (例如 SENDER) 或 CHANNEL

字段	值
名称	CHANNEL.TO.DELETE
处置	对象的位置。

您将看到另一个面板。 在此面板上选择函数类型 1。

按 Enter 键以删除通道定义; 再次按 Enter 键将要求您确认是否要删除通道定义。

注: 必须先运行通道启动程序，然后才能删除通道定义 (客户机连接通道除外)。

z/OS 显示有关通道启动程序的信息

您可以使用 MQSC 命令或使用操作和控制面板来显示有关通道启动程序的信息。

要使用 MQSC 命令显示有关通道启动程序的信息，请使用 DISPLAY CHINIT。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	1 (显示)
对象类型	SYSTEM
名称	Blank

您将看到另一个面板。 在此面板上选择函数类型 1。

注:

1. 如果您有许多通道，那么显示分布式排队信息可能需要一些时间。
2. 必须先运行通道启动程序，然后才能显示有关分布式排队的信息。

z/OS 保护通道

您可以使用 MQSC 命令或使用操作和控制面板来保护通道。

要使用 MQSC 命令保护通道，请使用 SET CHLAUTH。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	8

提供了一个编辑器，您可以在该编辑器中提供 MQSC 命令 (在本例中为 CHLAUTH 命令)，请参阅 第 805 页的图 110。 在命令中完成输入后，需要加号 (+)。 输入 PF3 以退出编辑器并将命令提交到命令服务器。

```
****************************** Top of Data ******************************
000001 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000002 TYPE(SSLPEERMAP) +
000003 SSLPEER('CN="John Smith"') +
000004 MCAUSER('PUBLIC')
****** **************************** Bottom of Data ****************************

Command ===>                           Scroll ===> PAGE
F1=Help   F3=Exit   F4=LineEdit F12=Cancel
```

图 *110:* 命令条目

然后，将向您显示该命令的输出，请参阅 第 806 页的图 111

```
****** ***************************** Top of Data ******************************
000001 CSQU000I CSQUTIL IBM MQ for z/OS V7.1.0
000002 CSQU001I CSQUTIL Queue Manager Utility - 2011-04-20 14:42:58
000003 COMMAND TGTQMGR(MQ23) RESPTIME(30)
000004 CSQU127I Executing COMMAND using input from CSQUCMD data set
000005 CSQU120I Connecting to MQ23
000006 CSQU121I Connected to queue manager MQ23
000007 CSQU055I Target queue manager is MQ23
000008 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000009 TYPE(SSLPEERMAP) +
000010 SSLPEER('CN="John Smith"') +
000011 MCAUSER('PUBLIC')
000012 CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000000
000013 CSQ9022I !MQ23 CSQMSCA ' SET CHLAUTH' NORMAL COMPLETION
000014 CSQU057I 1 commands read
000015 CSQU058I 1 commands issued and responses received, 0 failed
000016 CSQU143I 1 COMMAND statements attempted
000017 CSQU144I 1 COMMAND statements executed successfully
000018 CSQU148I CSQUTIL Utility completed, return code=0
Command ===>                          Scroll ===> PAGE
F1=Help   F3=Exit   F5=Rfind   F6=Rchange  F9=SwapNext F12=Cancel
```

图 *111:* 命令输出

z/OS 启动通道启动程序

您可以使用 MQSC 命令或使用操作和控制面板来启动通道启动程序。

要使用 MQSC 命令启动通道启动程序，请使用 START CHINIT。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	6 (启动)
对象类型	SYSTEM
名称	Blank

将显示 "启动系统功能" 面板。 以下面板后面的文本说明了要执行的操作:

```
Start a System Function

Select function type, complete fields, then press Enter to start system
function.

Function type . . . . . _   1. Channel initiator
2. Channel listener
Action queue manager . . . : MQ25

Channel initiator
JCL substitution . . . . . ________________________________________________
------------------------------------------------

Channel listener
Inbound disposition . . . Q G=Group, Q=Qmgr
Transport type . . . . . _ L=LU6.2, T=TCP/IP
LU name (LU6.2) . . . . . ________________
Port number (TCP/IP) . . . 1414
IP address (TCP/IP) . . . ________________________________________________



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

图 *112:* 启动系统功能

选择功能类型 1 (通道启动程序) ，然后按 Enter 键。

z/OS 停止通道启动程序

您可以使用 MQSC 命令或使用操作和控制面板来停止通道启动程序。

要使用 MQSC 命令停止通道启动程序，请使用 STOP CHINIT。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	7 (停止)
对象类型	SYSTEM
名称	Blank

将显示 "停止系统功能" 面板。 面板后面的文本说明了如何使用此面板:

```
Stop a System Function

Select function type, complete fields, then press Enter to stop system
function.

Function type . . . . . _   1. Channel initiator
2. Channel listener
Action queue manager . . . : MQ25

Channel initiator
Restart shared channels  Y Y=Yes, N=No

Channel listener
Inbound disposition . . . Q G=Group, Q=Qmgr
Transport type . . . . . _ L=LU6.2, T=TCP/IP

Port number (TCP/IP) . . . _____
IP address (TCP/IP) . . . ________________________________________________



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

图 *113:* 停止功能控件

选择功能类型 1 (通道启动程序) 并按 Enter 键。

通道启动程序在停止之前等待所有正在运行的通道以停顿方式停止。

注: 如果某些通道是正在运行但未处于活动状态的接收方或请求者通道，那么向接收方或发送方通道启动程序发出的停止请求会导致其立即停止。

但是，如果消息正在流动，那么通道启动程序会等待当前消息批次完成，然后再停止。

z/OS 启动通道侦听器

您可以使用 MQSC 命令或使用操作和控制面板来启动通道侦听器。

要使用 MQSC 命令启动通道侦听器，请使用 START LISTENER。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	6 (启动)
对象类型	SYSTEM
名称	Blank

将显示 "启动系统功能" 面板 (请参阅 第 806 页的图 112)。

选择函数类型 2 (通道侦听器)。 选择入站处置。 选择传输类型。 如果传输类型为 L ，请选择 LU 名。 如果 "传输" 类型为 T ，请选择 "端口号" 和 (可选) IP 地址。 按 Enter 键。

注: 对于 TCP/IP 侦听器，可以启动端口和 IP 地址的多个组合。

z/OS 停止通道侦听器

您可以使用 MQSC 命令或使用操作和控制面板来停止通道侦听器。

要使用 MQSC 命令停止通道侦听器，请使用 STOP LISTENER。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	7 (停止)
对象类型	SYSTEM
名称	Blank

将显示 "停止系统功能" 面板 (请参阅 第 807 页的图 113)。

选择函数类型 2 (通道侦听器)。 选择入站处置。 选择传输类型。 如果传输类型为 "T" ，请选择 "端口号" 和 (可选) IP 地址。 按 Enter 键。

注: 对于 TCP/IP 侦听器，可以停止端口和 IP 地址的特定组合，也可以停止所有组合。

z/OS 启动通道

您可以使用 MQSC 命令或使用操作和控制面板来启动通道。

要使用 MQSC 命令启动通道，请使用 START CHANNEL。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	6 (启动)
对象类型	通道类型 (例如 SENDER) 或 CHANNEL
名称	CHANNEL.TO.USE
处置	对象的处置。

将显示 "启动通道" 面板。 面板后面的文本说明了如何使用面板:

```
Start a Channel

Select disposition, then press Enter to start channel.


Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE


Disposition . . . . . P   P=Private on MQ25
S=Shared on MQ25
A=Shared on any queue manager









Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

图 *114:* 启动通道

选择通道实例的处置以及要在其上启动该通道实例的队列管理器。

按 Enter 键以启动通道。

z/OS 启动共享通道

要启动共享通道并将其保留在指定的通道启动程序上，请使用处置 = S (在 START CHANNEL 命令上，指定 CHLDISP (FIXSHARED))。

一次只能有一个共享通道实例在运行。 尝试启动通道的第二个实例失败。

以此方式启动通道时，以下规则适用于该通道:

- 您可以从队列共享组中的任何队列管理器停止通道。 即使在发出 stop-channel 请求时启动它的通道启动程序未在运行，您也可以将其停止。 当通道已停止时，您可以通过在同一通道启动程序或另一通道启动程序上指定处置 = S (CHLDISP (FIXSHARED)) 来重新启动该通道。 您还可以通过指定处置 = A (CHLDISP (SHARED)) 来将其启动。
- 如果通道处于正在启动或重试状态，那么可以通过在同一通道启动程序或其他通道启动程序上指定处置 = S (CHLDISP (FIXSHARED)) 来重新启动该通道。 您还可以通过指定处置 = A (CHLDISP (SHARED)) 来将其启动。
- 当通道进入不活动状态时，可以启动该通道。 启动触发器的共享通道始终具有共享处置 (CHLDISP (SHARED))。
- 当通道进入不活动状态时，可以在任何通道启动程序上使用 CHLDISP (FIXSHARED) 启动该通道。 您还可以通过指定处置 = A (CHLDISP (SHARED)) 来将其启动。
- 当使用 SHARED (RESTART) 停止启动通道的通道启动程序时，或者当通道启动程序异常终止时，队列共享组中的任何其他活动通道启动程序都不会恢复该通道。 仅当下次重新启动启动通道的通道启动程序时，才会恢复该通道。 这将停止将失败的通道恢复尝试传递到队列共享组中的其他通道启动器，这将增加它们的工作负载。

z/OS 测试通道

您可以使用 MQSC 命令或使用操作和控制面板来测试通道。

要使用 MQSC 命令测试通道，请使用 PING CHANNEL。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	5 (执行)
对象类型	SENDER， SERVER 或 CHANNEL
名称	CHANNEL.TO.USE
处置	通道对象的处置。

将显示 "执行通道功能" 面板。 面板后面的文本说明了如何使用面板:

```
Perform a Channel Function

Select function type, complete fields, then press Enter.


Function type . . . . . . . . _   1. Reset  3. Resolve with commit
2. Ping  4. Resolve with backout

Channel name . . . . . . . : CHANNEL.TO.USE
Channel type . . . . . . . : SENDER
Description . . . . . . . . : Description of CHANNEL.TO.USE


Disposition . . . . . . . . . P   P=Private on MQ25
S=Shared on MQ25
A=Shared on any queue manager

Sequence number for reset . . 1     1 - 999999999
Data length for ping . . . . 16   16 - 32768



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

图 *115:* 测试通道

选择函数类型 2 (ping)。

选择要对其执行测试的通道的处置以及要对其进行测试的队列管理器。

数据长度最初设置为 16。 如果需要，请进行更改，然后按 Enter 键。

z/OS 重置通道的消息序号

您可以使用 MQSC 命令或使用操作和控制面板来重置通道的消息序号。

要使用 MQSC 命令重置通道序号，请使用 RESET CHANNEL。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	5 (执行)
对象类型	通道类型 (例如 SENDER) 或 CHANNEL
名称	CHANNEL.TO.USE
处置	通道对象的处置。

将显示 "执行通道功能" 面板 (请参阅 第 810 页的图 115)。

选择功能类型 1 (重置)。

选择要对其执行重置的通道的处置以及要对其执行重置的队列管理器。

序号 字段最初设置为 1。 如果需要，请更改此值，然后按 Enter 键。

z/OS 在通道上解析不确定消息

您可以使用 MQSC 命令或使用操作和控制面板来解析通道上的不确定消息。

要使用 MQSC 命令解析通道上的不确定消息，请使用 RESOLVE CHANNEL。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	5 (执行)
对象类型	SENDER， SERVER 或 CHANNEL
名称	CHANNEL.TO.USE
处置	对象的处置。

将显示 "执行通道功能" 面板 (请参阅 第 810 页的图 115)。

选择函数类型 3 或 4 (使用落实或回退进行解析)。 (请参阅 第 188 页的『处理不确定通道』 以获取更多信息。)

选择要对其执行解析的通道的处置以及要对其执行解析的队列管理器。 按 Enter 键。

z/OS 停止通道

您可以使用 MQSC 命令或使用操作和控制面板来停止通道。

要使用 MQSC 命令停止通道，请使用 STOP CHANNEL。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	值
操作	7 (停止)
对象类型	通道类型 (例如 SENDER) 或 CHANNEL
名称	CHANNEL.TO.USE
处置	对象的处置。

这样会显示 "停止通道" 面板。 面板后面的文本说明了如何使用面板:

```
Stop a Channel

Complete fields, then press Enter to stop channel.


Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE


Disposition . . . . . P   P=Private on MQ25
A=Shared on any queue manager

Stop mode . . . . . 1   1. Quiesce  2. Force
Stop status . . . . . 1   1. Stopped  2. Inactive

Queue manager . . . . . _________________________________________________
Connection name . . . . . _________________________________________________



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

图 *116:* 停止通道

选择要对其执行停止的通道的处置以及要在哪个队列管理器上停止该通道。

选择所需的停止方式:

停顿
当当前消息完成并随后结束批处理时，通道将停止，即使尚未达到批处理大小值并且有消息已在传输队列上等待也是如此。 未启动任何新批处理。 此方式是缺省方式。

Force
通道立即停止。 如果正在处理一批消息，那么可能会产生 "不确定" 情境。

选择要停止的通道的队列管理器和连接名称。

选择所需的状态:

已停止
通道不会自动重新启动，必须手动重新启动。 如果未指定队列管理器或连接名称，那么此方式是缺省方式。 如果指定了名称，那么不允许该名称。

不活动
需要时将自动重新启动通道。 如果指定了队列管理器或连接名称，那么此方式是缺省方式。

按 Enter 键以停止通道。

请参阅 第 186 页的『停止和停顿通道』 以获取更多信息。 有关重新启动已停止的通道的信息，请参阅 第 188 页的『重新启动已停止的通道』。

注: 如果共享通道处于重试状态，并且启动该通道的通道启动程序未运行，那么将在输入该命令的队列管理器上发出针对该通道的 STOP 请求。

z/OS 显示通道状态

您可以使用 MQSC 命令或使用操作和控制面板来显示通道状态。

要使用 MQSC 命令显示通道或一组通道的状态，请使用 DISPLAY CHSTATUS。

注: 如果您有大量通道，那么显示通道状态信息可能需要一些时间。

通过使用 "列示通道" 面板上的操作和控制面板 (请参阅 第 803 页的图 109)，将显示每个通道的通道状态摘要，如下所示:

不活动的	没有处于活动状态的连接
状态	一个连接处于活动状态
nnn 状态	多个连接是当前连接，所有当前连接都具有相同状态
nnn 当前	多个连接是当前连接，并且当前连接并非都具有相同状态
Blank	IBM MQ 无法确定有多少连接处于活动状态 (例如，由于通道启动程序未在运行) **注:** 对于具有处置 GROUP 的通道对象，不会显示任何状态。

其中 *nnn* 是活动连接数， *status* 是下列其中一项:

初始化	正在初始化
BIND	BINDING
START	正在启动
RUN	正在运行
结束	STOPPED 或 STOPPED
重试	RETRYING
REQST	REQUESTING

要显示有关通道状态的更多信息，请按 "列表通道" 或 "显示" 或 "变更通道" 面板上的 "状态" 键 (F11) 以显示 "列表通道-当前状态" 面板 (请参阅 第 813 页的图 117)。

```
List Channels - Current Status - MQ25      Row 1 of 16

Type action codes, then press Enter. Press F11 to display saved status.
1=Display current status

Channel name     Connection name                 State
Start time      Messages Last message time  Type   Disposition
<> *                          CHANNEL  ALL   MQ25

_ RMA0.CIRCUIT.ACL.F  RMA1                        STOP
_  2005-03-21 10.22.36 557735  2005-03-24 09.51.11 SENDER  PRIVATE MQ25
_ RMA0.CIRCUIT.ACL.N  RMA1
_  2005-03-21 10.23.09 378675  2005-03-24 09.51.10 SENDER  PRIVATE MQ25
_ RMA0.CIRCUIT.CL.F  RMA2
_  2005-03-24 01.12.51 45544   2005-03-24 09.51.08 SENDER  PRIVATE MQ25
_ RMA0.CIRCUIT.CL.N  RMA2
_  2005-03-24 01.13.55 45560   2005-03-24 09.51.11 SENDER  PRIVATE MQ25
_ RMA1.CIRCUIT.CL.F  RMA1
_  2005-03-21 10.24.12 360757  2005-03-24 09.51.11 RECEIVER PRIVATE MQ25
_ RMA1.CIRCUIT.CL.N  RMA1
_  2005-03-21 10.23.40 302870  2005-03-24 09.51.09 RECEIVER PRIVATE MQ25
******** End of list ********
Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F4=Filter  F5=Refresh  F7=Bkwd
F8=Fwd    F9=SwapNext F10=Messages F11=Saved  F12=Cancel
```

图 *117:* 列出通道连接

状态值如下所示:

初始化	正在初始化
BIND	BINDING
START	正在启动
RUN	正在运行
结束	STOPPED 或 STOPPED
重试	RETRYING
REQST	REQUESTING
不确定	STOPPED 和 INDOUBT (YES)

请参阅 第 179 页的『通道状态』 以获取更多信息。

您可以按 F11 以查看具有已保存状态的通道连接的类似列表; 按 F11 以返回到当前列表。 在通道上至少传输了一批消息之后，保存的状态才适用。

使用操作码 1 或斜杠 (/) 来选择连接并按 Enter 键。 将显示 "显示通道连接当前状态" 面板。

z/OS 显示集群通道

您可以使用 MQSC 命令或使用操作和控制面板来显示集群通道。

要显示已定义的所有集群通道 (显式或使用自动定义) ，请使用 MQSC 命令 DISPLAY CLUSQMGR。

使用操作和控制面板，从初始面板开始，填写以下字段并按 Enter 键:

字段	**值**
操作	1 (列表或显示)
对象类型	CLUSCHL

字段	值
名称	*

将显示一个类似图 第 814 页的图 118 的面板，其中每个集群通道的信息占用三行，并包含其通道，集群和队列管理器名称。 对于集群发送方通道，将显示总体状态。

```
List Cluster queue manager Channels - MQ25    Row 1 of 9

Type action codes, then press Enter. Press F11 to display connection status.
1=Display  5=Perform  6=Start  7=Stop

Channel name     Connection name                 State
Type     Cluster name                   Suspended
Cluster queue manager name            Disposition
<>   *                          -    MQ25
_ TO.MQ90.T      HURSLEY.MACH90.COM(1590)
_  CLUSRCVR   VJH01T                        N
_   MQ90                        -    MQ25
_ TO.MQ95.T      HURSLEY.MACH95.COM(1595)             RUN
_  CLUSSDRA   VJH01T                        N
_   MQ95                        -    MQ25
_ TO.MQ96.T      HURSLEY.MACH96.COM(1596)             RUN
_  CLUSSDRB   VJH01T                        N
_   MQ96                        -    MQ25
******** End of list ********


Command ===> ____________________________________________________________________
F1=Help   F2=Split   F3=Exit   F4=Filter  F5=Refresh  F7=Bkwd
F8=Fwd    F9=SwapNext F10=Messages F11=Status  F12=Cancel
```

图 *118:* 列出集群通道

要显示有关一个或多个通道的完整信息，请针对其名称输入操作码 1 ，然后按 Enter 键。 使用操作码 5 ， 6 或 7 来执行功能 (例如 ping ，解析和重置) 以及启动或停止集群通道。

要显示有关通道状态的更多信息，请按 "状态" 键 (F11)。

z/OS 准备 *IBM MQ for z/OS* 以使用 *zEnterprise Data Compression Express* 工具

zEnterprise Data Compression (zEDC) Express 工具可用于特定型号的 IBM Z 机器，从 IBM zEC12 GA2 开始，使用最低 z/OS 级别 z/OS 2.1。

请参阅 zEnterprise 数据压缩 (zEDC) 以获取更多信息。

先决条件

对于 IBM z15 和更高版本， zEnterprise Data Compression (zEDC) Express 设施已从硬件系统的 PCIe I/O 抽屉中的可选功能部件移至片上作为 zEDC 的集成加速器。 通过此更改，将更新配置先决条件，并且这些先决条件取决于您的硬件系统。

IBM z15 或更高版本

根据您的 z/OS 级别，应用下列其中一个 PTF:

- z/OS 2.4: UJ00636
- z/OS 2.3: UJ00635
- z/OS 2.2: UJ00638
- z/OS 2.1: UJ00639

z15 或更高版本的系统没有硬件需求。 这些系统中的 Integrated Accelerator for zEDC 解决方案提供了内置的数据加速功能，因此不再需要单独的适配器。

IBM zEC12 GA2 到 IBM z14

系统还必须具有下列需求：

- 安装在硬件系统的 PCIe I/O 抽屉中的 zEDC Express ® 适配器。
- 必须在 IFAPRDxx parmlib 成员中启用 zEDC 软件功能 (可选的付费功能)。

过程

IBM zEC12 GA2 到 IBM z14

确保通道启动程序用户标识对 RACF FACILITY CLASS 中的 FPZ.ACCELERATOR.COMPRESSION 概要文件具有 READ 权限，或者对企业使用的外部安全管理器 (ESM) 中的等效概要文件具有 READ 权限。

注意: IBM z15 或更高版本不需要。

IBM zEnterprise zEC12 GA2 或更高版本

在发送端和接收端配置具有 COMPMSG (ZLIBFAST) 的通道。 配置后，将使用 zlib 压缩来压缩和解压缩流经通道的消息。

当要压缩的数据大小超过最小阈值时，将在 zEDC 中执行压缩。 阈值取决于正在使用的 IBM z 硬件

- IBM zEC12 GA2 到 IBM z14 的最小阈值为 4KB
- IBM z15 或更高版本的最小阈值为 1KB

对于低于阈值大小的消息，将在软件中执行压缩或通货膨胀。

z/OS 为 z/OS 设置通信

当启动分布式排队管理通道时，它会尝试使用通道定义中指定的连接。 要成功，需要定义连接并使其可用。本节说明如何定义连接。

DQM 是 IBM MQ 的远程排队工具。 它为构成通信链路接口的队列管理器提供通道控制程序。 这些链路由系统操作员控制。 分布式排队管理所持有的通道定义使用这些连接。

从可用于 z/OS 的两种形式的通信协议中选择一种:

- 第 816 页的『在 z/OS 上定义 TCP 连接』
- 第 818 页的『使用 APPC/MVS 为 z/OS 定义 LU6.2 连接』

CD MQ Adv. V 9.2.0 使用 TCP/IP 的消息通道可以指向 IBM Aspera fasp.io Gateway，它提供了可显着提高网络吞吐量的快速 TCP/IP 隧道。 在任何授权平台上运行的队列管理器都可以通过 Aspera gateway 进行连接。 网关本身部署在 Red Hat 或 Ubuntu Linux 或 Windows 上。 请参阅 在 Linux 或 Windows 上定义 Aspera gateway 连接。

每个通道定义必须仅指定一个协议作为传输协议 (传输类型) 属性。 队列管理器可以使用多个协议进行通信。

您可能还会发现请参阅 示例配置- IBM MQ for z/OS 。如果您正在使用队列共享组，请参阅 第 822 页的『使用队列共享组为 IBM MQ for z/OS 设置通信』。

相关概念

第 802 页的『使用面板和命令』
您可以使用 MQSC 命令， PCF 命令或操作和控制面板来管理 DQM。

第 734 页的『设置 IBM MQ for z/OS』
使用本主题作为定制 IBM MQ for z/OS 系统的逐步指南。

第 800 页的『监视和控制 z/OS 上的通道』
使用 DQM 命令和面板来创建，监视和控制到远程队列管理器的通道。

第 819 页的『为具有队列共享组的 DQM 准备 IBM MQ for z/OS』
使用本节中的指示信息在 IBM MQ for z/OS 上配置具有队列共享组的分布式排队。

第 822 页的『使用队列共享组为 IBM MQ for z/OS 设置通信』

当分布式排队管理通道启动时，它会尝试使用通道定义中指定的连接。 要使此尝试成功，需要定义连接并使其可用。

相关任务

第 152 页的『配置分布式队列』
此部分提供有关 IBM MQ 安装之间的相互通信的更详细信息，包括队列定义，通道定义，触发和同步点过程

第 797 页的『在 z/OS 上设置与其他队列管理器的通信』
本部分描述了在开始使用分布式排队之前需要进行的 IBM MQ for z/OS 准备工作。

z/OS 在 *z/OS* 上定义 *TCP* 连接

要定义 TCP 连接，需要配置许多设置。

必须在 TCP 系统参数数据集 *tcpip*.TCPIP.DATA。 在数据集中，必须包含 “TCPIPJOBNAME *TCPIP_proc*” 语句。

如果您正在使用防火墙，那么需要配置 `allow` 从通道启动程序到通道中地址的连接，以及从远程连接到队列管理器的连接。

通常，防火墙的定义会配置将 IP 地址和端口发送到目标 IP 地址和端口:

- z/OS 映像可以具有多个主机名，您可能需要使用多个主机地址作为源地址来配置防火墙。

  可以使用 NETSTAT HOME 命令来显示这些名称和地址。
- 通道启动程序可以在不同端口上具有多个侦听器，因此您需要配置这些端口。
- 如果要将共享端口用于队列共享组，那么还必须配置该共享端口。

通道启动程序地址空间必须有权读取数据集。 可以使用以下方法来访问 TCPIP.DATA 数据集，取决于您正在使用的 TCP/IP 产品和接口:

- 环境变量， RESOLVER_CONFIG
- 文件系统上的 /etc/resolv.conf
- // SYSTCPD DD 语句
- // SYSTCPDD DD 语句
- *jobname/userid*.TCPIP.DATA
- SYS1.TCPPARMS(TCPDATA)
- *zapname*.TCPIP.DATA

您还必须注意正确指定 TCP/IP 的高级限定符。

您需要一个适当配置的域名系统 (DNS) 服务器，能够同时进行 "名称到 IP 地址" 转换和 "IP 地址到名称" 转换。

注: 对解析器配置的某些更改需要重新启动使用它的应用程序，例如， IBM MQ。

要获取更多信息，请参阅以下内容:

- 基本 TCP/IP 系统
- z/OS UNIX System Services.

每个 TCP 通道在启动时都使用 TCP 资源; 您可能需要在 PROFILE.TCPIP 配置数据集:

ACBPOOLSIZE
每个启动的 TCP 通道添加一个，外加一个

CCBPOOLSIZE
每个已启动的 TCP 通道添加一个，每个 DQM 分派器添加一个，外加一个

DATABUFFERPOOLSIZE
每个启动的 TCP 通道添加两个，外加一个

MAXFILEPROC
控制通道启动程序中每个分派器可以处理的通道数。

此参数在 SYSI.PARMLIB。 请确保指定足够大的值以满足您的需要。

缺省情况下，通道启动程序只能绑定到与 TCPNAME 队列管理器属性中指定的堆栈相关联的 IP 地址。 要允许通道启动程序使用系统上的其他 TCP/IP 堆栈进行通信，请将 TCPSTACK 队列管理器属性更改为 MULTIPLE。

CD MQ Adv. V 9.2.0 使用 TCP/IP 的消息通道可以指向 IBM Aspera fasp.io Gateway，它提供了可显着提高网络吞吐量的快速 TCP/IP 隧道。 在任何授权平台上运行的队列管理器都可以通过 Aspera gateway 进行连接。 网关本身部署在 Red Hat 或 Ubuntu Linux 或 Windows 上。 请参阅 在 Linux 或 Windows 上定义 Aspera gateway 连接。

相关概念

第 817 页的『发送结束』
在 TCP/IP 连接的发送端，有许多要配置的设置。

第 817 页的『在 TCP 上接收』
在 TCP/IP 连接的接收端，有许多要配置的设置。

第 817 页的『在 z/OS 上使用 TCP 侦听器储备选项』
在 TCP/IP 上接收时，将设置最大未完成的连接请求数。 这些未完成的请求可被视为在 TCP/IP 端口上等待侦听器接受请求的 积压 请求。

z/OS 发送结束

在 TCP/IP 连接的发送端，有许多要配置的设置。

通道定义中的连接名称 (CONNAME) 字段必须设置为主机名 (例如 MVSHUR1) 或目标的 TCP 网络地址。 TCP 网络地址可以采用 IPv4 点分十进制格式 (例如 127.0.0.1) 或 IPv6 十六进制格式 (例如 2001:DB8:0:0:0:0:0:0)。 如果连接名称是主机名，那么需要 TCP 名称服务器将主机名转换为 TCP 主机地址。 (此需求是 TCP 的函数，而不是 IBM MQ。)

在连接的起始端 (发送方，请求者和服务器通道类型) 上，可以提供连接的可选端口号，例如:

连接名称
192.0.2.0(1555)

在这种情况下，发起端尝试连接到侦听端口 1555 的接收程序。

注: 如果未指定可选端口号，那么将使用缺省端口号 1414。

通道启动程序可以使用任何处于活动状态且可用的 TCP/IP 堆栈。 缺省情况下，通道启动程序将其出站通道与 TCPNAME 队列管理器属性中指定的 TCP/IP 堆栈的缺省 IP 地址绑定。 要通过其他堆栈进行连接，需要在通道的 LOCLADDR 属性中指定堆栈的主机名或 IP 地址。

z/OS 在 *TCP* 上接收

在 TCP/IP 连接的接收端，有许多要配置的设置。

接收通道程序是响应来自发送通道的启动请求而启动的。 为此，必须启动侦听器程序以检测入局网络请求并启动关联通道。 可使用 START LISTENER 命令或使用操作和控制面板来启动此侦听器程序。

缺省情况下:

- TCP 侦听器程序使用端口 1414 并侦听可用于 TCP 堆栈的所有地址。
- TCP/IP 侦听器只能绑定到与 TCPNAME 队列管理器属性中指定的 TCP/IP 堆栈相关联的地址。

要启动其他地址或所有可用 TCP 堆栈的侦听器，请将 TCPSTACK 队列管理器属性设置为 "MULTIPLE"。

通过在 START LISTENER 命令中指定 IPADDR ，可以启动 TCP 侦听器程序以仅侦听特定地址或主机名。 有关更多信息，请参阅侦听器。

z/OS 在 *z/OS* 上使用 *TCP* 侦听器储备选项

在 TCP/IP 上接收时，将设置最大未完成的连接请求数。 这些未完成的请求可被视为在 TCP/IP 端口上等待侦听器接受请求的 积压 请求。

z/OS 上的缺省侦听器任务列表值为 10000。 如果积压达到此值，那么将拒绝 TCP/IP 连接，并且通道无法启动。

对于 MCA 通道，这会导致通道进入 RETRY 状态，并在稍后重试连接。

对于客户机连接，客户机从 MQCONN 接收到 MQRC_Q_MGR_NOT_AVAILABLE 原因码，并且可以稍后重试连接。

相关任务

在 AIX and Linux 上使用 TCP 侦听器储备选项

z/OS 使用 *APPC/MVS* 为 *z/OS* 定义 *LU6.2* 连接

要定义 LU6.2 连接，需要配置许多设置。

APPC/MVS 设置

通道启动程序的每个实例都必须具有要在 SYS1.PARMLIB，如以下示例中所示:

```
LUADD ACBNAME( luname ) NOSCHED TPDATA(CSQ.APPCTP)
```

luname 是要使用的逻辑单元的名称。 NOSCHED 是必需的; 不使用 TPDATA 。 不需要对 ASCHPMxx 成员或 APPC/MVS TP 概要文件数据集进行添加。

必须扩展辅助信息数据集以定义 DQM 所使用的连接。 请参阅提供的样本 CSQ4SIDE ， 以获取有关如何使用 APPC 实用程序 ATBSDFMU 执行此操作的详细信息。 有关要使用的 TPNAME 值的详细信息，请参阅下表以获取信息:

表 *83:* 远程队列管理器平台的本地 *z/OS* 系统上的设置

远程平台	TPNAME
z/OS 或 MVS	与有关远程队列管理器的相应辅助信息中的 TPNAME 相同。
IBM i	与 IBM i 系统上的路由条目中的比较值相同。
AIX and Linux 系统	与有关远程队列管理器的相应辅助信息中的 TPNAME 相同。
Windows	如 Windows Run Listener 命令或使用 Windows 上的 TpSetup 定义的可调用事务程序中所指定。

如果同一机器上有多个队列管理器，请确保通道定义中的 TPname 是唯一的。

另请参阅 多平台 *APPC* 配置指南 ， 以获取有关可能需要的 VTAM 定义的信息。

在队列管理器使用 APPC 与同一或另一 z/OS 系统上的队列管理器进行通信的环境中，请确保通信 LU 的 VTAM 定义指定 SECACPT (ALREADYV) ， 或者为 LU 之间的连接提供 RACF APPCLU 概要文件，该概要文件指定 CONVSEC (ALREADYV)。

在尝试启动入站或出站通信之前，必须对基本 LU 和侦听器 LU 发出 z/OS 命令 VARY ACTIVE。

注意: 除了 APPC 设置外，还必须发出以下命令:

```
ALTER QMGR LUNAME(luname)
```

并重新启动通道启动程序。

请参阅 LUNAME 以获取更多信息。

相关概念

第 818 页的『连接到 LU 6.2』
要连接到 LU 6.2，需要配置许多设置。

第 819 页的『在 LU 6.2 上接收』
要在 LU 6.2 上接收，需要配置许多设置。

z/OS 连接到 *LU 6.2*

要连接到 LU 6.2，需要配置许多设置。

通道定义中的连接名称 (CONNAME) 字段必须设置为符号目标名称，如 APPC/MVS 的辅助信息数据集中所指定。

还必须在通道启动程序参数中指定要使用的 LU 名 (如前所述对 APPC/MVS 定义)。 它必须设置为侦听器用于接收的同一 LU。

通道启动程序使用 “SECURITY (SAME)” APPC/MVS 选项，因此它是用于出站传输的通道启动程序地址空间的用户标识，并提供给接收方。

z/OS 在 *LU 6.2* 上接收

要在 LU 6.2 上接收，需要配置许多设置。

启动接收 MCA 以响应来自发送通道的启动请求。 为此，必须启动侦听器程序以检测入局网络请求并启动关联通道。 侦听器程序是 APPC/MVS 服务器。 可以使用 START LISTENER 命令或使用操作和控制面板将其启动。 必须指定要与辅助信息数据集中定义的符号目标名称配合使用的 LU 名。 如此标识的本地 LU 必须与用于出站传输的 LU 相同，如在通道启动程序参数中所设置。

z/OS 为具有队列共享组的 DQM 准备 IBM MQ for z/OS

使用本节中的指示信息在 IBM MQ for z/OS 上配置具有队列共享组的分布式排队。

有关使用队列共享组的示例配置，请参阅 示例配置- IBM MQ for z/OS 使用队列共享组。 有关使用队列共享组的消息通道规划示例，请参阅 使用队列共享组的 z/OS 消息通道规划示例。

您需要创建并配置以下组件，以允许使用队列共享组进行分布式排队:

- LU 6.2 和 TCP/IP 侦听器
- 传输队列和触发
- 消息通道代理程序
- 同步队列

创建需要设置通信的组件后，请参阅 第 822 页的『使用队列共享组为 IBM MQ for z/OS 设置通信』。

有关如何在使用队列共享组时监视和控制通道的信息，请参阅 第 800 页的『监视和控制 z/OS 上的通道』。

请参阅以下部分以了解队列共享组概念和优点。

服务等级

共享队列是提供不同服务类的本地队列类型。 共享队列上的消息存储在耦合设施 (CF) 中，这允许队列共享组中的所有队列管理器访问这些消息。 共享队列上的消息必须是长度不超过 100 MB 的消息。

通用接口

队列共享组具有一个通用接口，该接口允许网络将该组作为单个实体查看。 此视图通过具有可用于连接到组中的任何队列管理器的单个通用地址来实现。

队列共享组中的每个队列管理器都侦听与通用地址逻辑相关的地址上的入站会话请求。 有关更多信息，请参阅第 820 页的『用于队列共享组的 LU 6.2 和 TCP/IP 侦听器』。

负载均衡通道启动

共享传输队列可由队列共享组中任何通道启动程序上运行的出站通道提供服务。 负载均衡通道启动确定启动通道命令的目标位置。 选择了可访问必需通信子系统的相应通道启动程序。 例如，使用 TRPTYPE (LU6.2) 定义的通道不能在只能访问 TCP/IP 子系统的通道启动程序上启动。

通道启动程序的选择取决于通道负载和通道启动程序的头空间。 通道负载是活动通道数占通道启动程序参数中定义的允许的最大活动通道数的百分比。 headroom 是活动通道数与允许的最大数目之间的差值。

可以使用通用地址在队列共享组中对入站共享通道进行负载均衡，如 第 820 页的『用于队列共享组的 LU 6.2 和 TCP/IP 侦听器』中所述。

共享通道恢复

下表显示共享通道故障的类型以及每个类型的处理方式。

故障类型：	处理方式：
通道启动程序通信子系统故障	依赖于通信子系统的通道开始通道重试，并由负载平衡启动命令在适当的队列共享组通道启动程序中重新启动。
通道启动程序故障	通道启动程序失败，但是关联的队列管理器仍保持活动状态。 队列管理器会监控失败情况并启动恢复处理。
队列管理器故障	队列管理器失败（使关联的通道启动程序失败）。 队列共享组中的其他队列管理器监控事件并启动对等恢复。
共享状态故障	通道状态信息存储在 Db2 中，因此通道状态发生更改时，如果指向 Db2 的连接丢失，那么会发生故障。 运行中的通道可在不访问这些资源的情况下继续运行。 在对 Db2 的访问失败后，通道开始进行重试。

代表故障系统进行共享通道恢复处理要求到 Db2 的连接在管理恢复的系统上可用，以便检索共享通道状态。

客户机通道

客户机连接通道可以从连接到通用接口的队列共享组中的消息的高可用性 (而不是连接到特定队列管理器) 中获益。 有关更多信息，请参阅 客户机连接通道。

相关概念

共享队列和队列共享组

第 734 页的『设置 IBM MQ for z/OS』
使用本主题作为定制 IBM MQ for z/OS 系统的逐步指南。

第 822 页的『集群和队列共享组』
您可以使共享队列在单个定义中可供集群使用。 要执行此操作，请在定义共享队列时指定集群的名称。

第 822 页的『通道和序列化』
在共享队列对等恢复期间，处理共享队列上的消息的消息通道代理将其对队列的访问序列化。

组内排队

相关任务

第 152 页的『配置分布式队列』
此部分提供有关 IBM MQ 安装之间的相互通信的更详细信息，包括队列定义，通道定义，触发和同步点过程

第 797 页的『在 z/OS 上设置与其他队列管理器的通信』
本部分描述了在开始使用分布式排队之前需要进行的 IBM MQ for z/OS 准备工作。

z/OS 用于队列共享组的 *LU 6.2* 和 *TCP/IP* 侦听器

组 LU 6.2 和 TCP/IP 侦听器侦听逻辑连接到通用地址的地址。

对于 LU 6.2 侦听器，指定的 LUGROUP 将映射到与队列共享组相关联的 VTAM 通用资源。 有关设置此技术的示例，请参阅 第 818 页的『使用 APPC/MVS 为 z/OS 定义 LU6.2 连接』。

对于 TCP/IP 侦听器，可以通过下列其中一种方式将指定的端口连接到通用地址:

- 对于前端路由器 (例如 IBM Network Dispatcher) ，入站连接请求将从路由器转发到队列共享组的成员。
- 对于 TCP/IP 综合系统分发器，正在运行且正在侦听设置为分布式 DVIPA 的特定地址的每个侦听器都将分配一部分入局请求。 有关设置此技术的示例，请参阅 使用综合系统分发器

z/OS 队列共享组的传输队列和触发

共享传输队列用于在将消息从队列共享组移动到目标之前存储这些消息。

它是共享队列，可供队列共享组中的所有队列管理器访问。

触发

触发的共享队列可以为满足的触发条件生成多条触发消息。 为与触发的共享队列相关联的队列共享组中的队列管理器上定义的每个本地启动队列生成一条触发器消息。

对于分布式排队，每个通道启动程序接收满足的共享传输队列触发条件的触发消息。 但是，只有一个通道启动程序实际处理触发的启动，其他通道启动程序安全失败。 然后，通过负载均衡启动来启动触发的通道 (请参阅 第 819 页的『为具有队列共享组的 DQM 准备 IBM MQ for z/OS』) 触发以启动通道 QSG.TO.QM2。 要创建共享传输队列，请使用 IBM MQ 命令 (MQSC) ，如以下示例中所示:

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') +
USAGE(XMITQ) QSGDISP(SHARED) +
CFSTRUCT(APPLICATION1) INITQ(SYSTEM.CHANNEL.INITQ) +
TRIGGER TRIGDATA(QSG.TO.QM2)
```

注: 如果设置了共享队列以触发并连接到托管共享队列的耦合设施，那么可能会生成触发器事件并将消息放入启动队列。 即使未将任何消息放入原始共享队列设置以进行触发，也会发生此情况。 这是由 IXLVECTR 宏 (如 The List Notification Vector 中所述) 过度指示位所导致的。

z/OS 队列共享组的消息通道代理程序

仅当通道具有对具有该名称的通道的通道定义的访问权时，才能在通道启动程序上启动该通道。

消息通道代理程序是一个 IBM MQ 程序，用于控制消息的发送和接收。 消息通道代理程序将消息从一个队列管理器移动到另一个队列管理器; 通道的每一端都有一个消息通道代理程序。

可以将通道定义定义为队列管理器专用，也可以将其存储在共享存储库中，并且在任何位置都可用 (组定义)。 这意味着组定义的通道在队列共享组中的任何通道启动程序上都可用。

注: 可以更改或删除组定义的专用副本。

要创建组通道定义，请使用 IBM MQ 命令 (MQSC) ，如以下示例中所示:

```
DEFINE CHL(QSG.TO.QM2) CHLTYPE(SDR) +
TRPTYPE(TCP) CONNAME(QM2.MACH.IBM.COM) +
XMITQ(QM2) QSGDISP(GROUP)
```

```
DEFINE CHL(QM2.TO.QSG) CHLTYPE(RCVR) TRPTYPE(TCP) +
QSGDISP(GROUP)
```

可以从两个透视图来查看用于与队列共享组进行分布式排队的消息通道代理程序:

入站

如果入站通道通过组侦听器连接到队列管理器，那么入站通道是共享通道。 它通过通用接口连接到队列共享组，然后定向到组中的队列管理器，或者以特定队列管理器的组端口或组侦听器所使用的 luname 为目标。

出库

如果出站通道从共享传输队列中移动消息，那么该出站通道是共享通道。 在示例命令中，发送方通道 QSG.TO.QM2 是共享通道，因为其传输队列 QM2 是使用 QSGDISP (SHARED) 定义的。

z/OS 队列共享组的同步队列

共享通道有自己的称为 SYSTEM.QSG.CHANNEL.SYNCQ。

此同步队列可供队列共享组的任何成员访问。 (专用通道继续使用专用同步队列。 请参阅 第 799 页的『定义 IBM MQ 对象』)。 这意味着在通信子系统，通道启动程序或队列管理器发生故障时，可以在队列共享组中的其他队列管理器和通道启动程序实例上重新启动通道。 有关更多信息，请参阅第 819 页的『为具有队列共享组的 DQM 准备 IBM MQ for z/OS』。

具有队列共享组的 DQM 要求共享队列具有名称 SYSTEM.QSG.CHANNEL.SYNCQ。 此队列必须可用，这样组侦听器才能成功启动。

如果由于队列不可用而导致组侦听器失败，那么可以定义该队列，并且可以在不重新启动通道启动程序的情况下重新启动该侦听器。 非共享通道不受影响。

确保使用 INDXTYPE (MSGID) 定义此队列。 此定义可提高可访问队列上的消息的速度。

z/OS 集群和队列共享组

您可以使共享队列在单个定义中可供集群使用。 要执行此操作，请在定义共享队列时指定集群的名称。

网络中的用户将共享队列视为由队列共享组中的每个队列管理器托管。 (未将共享队列通告为由队列共享组主管)。 客户机可以与队列共享组的所有成员启动会话，以将消息放入同一共享队列。

有关更多信息，请参阅第 235 页的『配置队列管理器集群』。

z/OS 通道和序列化

在共享队列对等恢复期间，处理共享队列上的消息的消息通道代理将其对队列的访问序列化。

如果队列共享组中的队列管理器在消息通道代理程序处理一个或多个共享队列上未落实的消息时失败，那么通道和关联的通道启动程序将结束，并且将对队列管理器进行共享队列对等恢复。

由于共享队列对等恢复是一项异步活动，因此在共享队列对等恢复完成之前，对等通道恢复可能会尝试在队列共享组的另一部分中同时重新启动该通道。 如果发生此事件，那么可能会在仍在恢复的消息之前处理已落实的消息。 为了确保以这种方式不按顺序处理消息，在共享队列上处理消息的消息通道代理程序将其对这些队列的访问序列化。

尝试启动共享队列对等恢复仍在进行的通道可能会导致失败。 将发出一条错误消息，指示正在进行恢复，并且通道将进入重试状态。 队列管理器对等恢复完成后，通道可以在下次重试时重新启动。

尝试解析， PING 或 DELETE 通道可能由于相同的原因而失败。

z/OS 使用队列共享组为 *IBM MQ for z/OS* 设置通信

当分布式排队管理通道启动时，它会尝试使用通道定义中指定的连接。 要使此尝试成功，需要定义连接并使其可用。

从可使用的两种通信协议形式中选择一种:

- TCP
- LU 6.2 到 APPC/MVS

您可能会发现参考 示例配置- IBM MQ for z/OS 使用队列共享组很有用。

z/OS 定义队列共享组的 *TCP* 连接

要为队列共享组定义 TCP 连接，必须配置发送端和接收端上的某些属性。

有关设置 TCP 的信息，请参阅 第 816 页的『在 z/OS 上定义 TCP 连接』。

发送结束

通道定义中用于连接到队列共享组的连接名称 (CONNAME) 字段必须设置为队列共享组的通用接口 (请参阅 队列共享组)。 有关更多详细信息，请参阅 使用综合系统分发器。

使用队列共享组在 TCP 上接收

接收共享通道程序是响应来自发送通道的启动请求而启动的。 为此，必须启动侦听器以检测入局网络请求并启动关联通道。 可以使用 START LISTENER 命令，使用组的入站处置或使用操作和控制面板来启动此侦听器程序。

队列共享组中的所有组侦听器都必须在同一端口上侦听。 如果在单个 MVS 映像上运行了多个通道启动程序，那么可以通过在 START LISTENER 命令中指定 IPADDR 来定义虚拟 IP 地址并启动 TCP 侦听器程序以仅侦听特定地址或主机名。 (有关更多信息，请参阅 START LISTENER。)

z/OS 在 *z/OS* 上定义 *LU 6.2* 连接

要为队列共享组定义 LU 6.2 连接，必须配置发送端和接收端上的某些属性。

有关设置 APPC/MVS 的信息，请参阅 设置 z/OS 的通信 。

连接到 APPC/MVS (LU 6.2)

用于连接到队列共享组的通道定义中的连接名称 (CONNAME) 字段必须设置为符号目标名称，如 APPC/MVS 的辅助信息数据集中所指定。 在此符号目标中指定的伙伴 LU 必须是通用资源名称。 有关更多详细信息，请参阅 使用通用资源定义自己到网络。

使用通用接口在 LU 6.2 上接收

接收共享 MCA 是响应来自发送通道的启动请求而启动的。 要执行此操作，必须启动组侦听器程序以检测入局网络请求并启动关联通道。 侦听器程序是 APPC/MVS 服务器。 您可以使用 START LISTENER 命令，使用入站处置组或使用操作和控制面板将其启动。 必须指定 LU 名以使用在辅助信息数据集中定义的符号目标名称。 有关更多详细信息，请参阅 使用通用资源定义自己到网络。

z/OS 将 IBM MQ 用于 IMS

IBM MQ -IMS 适配器和 IBM MQ - IMS 网桥是允许 IBM MQ 与 IMS 进行交互的两个组件。

要配置 IBM MQ 和 IMS 以协同工作，必须完成以下任务:

- 第 823 页的『设置 IMS 适配器』
- 第 828 页的『设置 IMS 网桥』

相关概念

IBM MQ 和 IMS

第 829 页的『将 IBM MQ 与 CICS 结合使用』
要将 IBM MQ 与 CICS 配合使用，必须配置 IBM MQ CICS 适配器以及 (可选) IBM MQ CICS bridge 组件。

第 832 页的『在 IMS 中使用 OTMA 出口』
如果要将 IMS Open Transaction Manager Access 出口与 IBM MQ for z/OS 配合使用，请使用本主题。

IBM MQ for z/OS 上的 IMS 和 IMS 网桥应用程序

相关任务

第 730 页的『在 z/OS 上配置队列管理器』
使用以下指示信息在 IBM MQ for z/OS 上配置队列管理器。

相关参考

第 830 页的『将服务升级并应用于 Language Environment 或 z/OS Callable Services』
您必须执行的操作根据您使用 CALLLIBS 还是 LINK 以及 SMP/E 的版本而有所不同。

z/OS 设置 IMS 适配器

要在 IMS 中使用 IBM MQ，需要 IBM MQ - IMS 适配器 (通常称为 IMS 适配器)。

本主题说明如何使 IMS 适配器可供 IMS 子系统使用。 如果您不熟悉定制 IMS 子系统，请参阅 IBM Documentation 中的 IMS 信息。

要使 IMS 适配器可供 IMS 应用程序使用，请执行以下步骤:

1. 使用 IMS 外部子系统连接设施 (ESAF) 将 IBM MQ 定义为 IMS 外部子系统。

   请参阅 第 825 页的『将 IBM MQ 定义到 IMS』。
2. 将 IBM MQ 装入库 thlqual.SCSQAUTH 包含在 JCL 中的 JOBLIB 或 STEPLIB 并置中，以用于 IMS 控制区域以及连接到 IBM MQ 的任何从属区域 (如果它不在 LPA 或链接列表中)。 如果未授权 JOBLIB 或 STEPLIB，请在包含 IMS 模块 (通常为 IMS RESLIB) 的库之后将其包含在 DFSESL 并置中。

   还包括 thlqual.SCSQANLx (其中 x 是语言字母)。

   如果存在 DFSESL，那么需要将 SCSQAUTH 和 SCSQANLx 包含在并置中或添加到 LNKLIST 中。 在 JCL 中添加到 STEPLIB 或 JOBLIB 并置是不够的。
3. 将 IBM MQ 汇编程序 CSQQDEFV 从 thlqual.SCSQASMS 复制到用户库。
4. 提供的程序 CSQQDEFV 包含一个子系统名称 CSQ1，该子系统名称通过 IMS 语言接口令牌 (LIT) MQM1 标识为缺省值。 您可以保留此名称以进行测试和安装验证。

对于生产子系统，将 NAME=CSQ1 更改为您自己的子系统名称，或者使用 CSQ1。 您可以根据需要添加更多子系统定义。 请参阅 第 827 页的『向 IMS 适配器定义 IBM MQ 队列管理器』，以获取有关 LIT 的更多信息。

5. 组装并链接编辑程序以生成 CSQQDEFV 装入模块。 对于组合件，请在 SYSLIB 并置中包含库 thlqual.SCSQMACS ; 请使用 link-edit 参数 RENT。 这在 thlqual.SCSQPROC(CSQ4DEFV) 中的样本 JCL 中显示。
6. 包含包含您在 JCL 中的 JOBLIB 或 STEPLIB 并置中为连接到 IBM MQ 的任何从属区域创建的模块 CSQQDEFV 的用户库。 将此库放在 SCSQAUTH 之前，因为 SCSQAUTH 具有缺省装入模块。 如果不执行此操作，那么将从 IMS 接收到用户 3041 异常终止。
7. 如果 IMS 适配器检测到意外 IBM MQ 错误，那么它会向 DD 名称 CSQSNAP 发出 z/OS SNAP 转储，并向应用程序发出原因码 MQRC_UNEXPECTED_ERROR。 如果 CSQSNAP DD 语句不在 IMS 从属区域 JCL 中，那么不会执行转储。 如果发生这种情况，那么可以在 JCL 中包含 CSQSNAP DD 语句，然后重新运行应用程序。 但是，由于某些问题可能是间歇性的，因此建议您包含 CSQSNAP DD 语句以捕获发生故障的原因。
8. 如果要使用动态 IBM MQ 调用 (如 动态调用 IBM MQ 存根 中所述)，请构建动态存根，如 第 824 页的 图 119 中所示。
9. 如果要使用 IMS 触发器监视器，请定义 IMS 触发器监视器应用程序 CSQQTRMN，然后执行 PSBGEN 和 ACBGEN。 请参阅第 828 页的『设置 IMS 触发器监视器』。
10. 如果要使用 RACF 来保护 OPERCMDS 类中的资源，请确保与 IBM MQ 队列管理器地址空间相关联的用户标识有权向其可能连接到的任何 IMS 系统发出 MODIFY 命令。

```
//DYNSTUB EXEC PGM=IEWL,PARM='RENT,REUS,MAP,XREF'
//SYSPRINT DD  SYSOUT=*
//ACSQMOD  DD  DISP=SHR,DSN=thlqual.SCSQLOAD
//IMSLIB   DD  DISP=SHR,DSN=ims.reslib
//SYSLMOD  DD  DISP=SHR,DSN=private.load¹
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,1)
//SYSLIN   DD  *
  INCLUDE ACSQMOD(CSQQSTUB)
  INCLUDE IMSLIB(DFSLI000)
  ALIAS MQCONN,MQCONNX,MQDISC             MQI entry points
  ALIAS MQGET,MQPUT,MQPUT1                MQI entry points
  ALIAS MQOPEN,MQCLOSE                    MQI entry points
  ALIAS MQBACK,MQCMIT                     MQI entry points
  ALIAS CSQBBAK,CSQBCMT                   MQI entry points
  ALIAS MQINQ,MQSET                       MQI entry points
  ALIAS DFSPLI,PLITDLI                    IMS entry points
  ALIAS DFSCOBOL,CBLTDLI                  IMS entry points
  ALIAS DFSFOR,FORTDLI                    IMS entry points
  ALIAS DFSASM,ASMTDLI                    IMS entry points
  ALIAS DFSPASCL,PASTDLI                  IMS entry points
  ALIAS DFHEI01,DFHEI1                    IMS entry points
  ALIAS DFSAIBLI,AIBTDLI                  IMS entry points
  ALIAS DFSESS,DSNWLI,DSNHLI              IMS entry points
  ALIAS MQCRTMH,MQDLTMH,MQDLTMP           IMS entry points
  ALIAS MQINQMP,MQSETMP,MQMHBUF,MQBUFMH   IMS entry points
  MODE AMODE(31),RMODE(24)                Note RMODE setting
  NAME CSQQDYNS(R)
/*

¹Specify the name of a library accessible to IMS applications that
want to make dynamic calls to IBM MQ.
```

图 *119:* 用于链接编辑动态调用存根的样本 *JCL*

相关概念

IBM MQ 和 IMS

第 828 页的『设置 IMS 网桥』

IBM MQ - IMS 网桥是一个可选组件，它使 IBM MQ 能够在未启用 IBM MQ 的现有程序和事务之间进行输入和输出。

IBM MQ for z/OS 上的 IMS 和 IMS 网桥应用程序

z/OS 将 *IBM MQ* 定义到 *IMS*

必须将 IBM MQ 定义到 IMS 控制区域以及访问该 IBM MQ 队列管理器的每个从属区域。 要执行此操作，必须在 IMS 中创建子系统成员 (SSM)。PROCLIB 库，并向适用的 IMS 区域标识 SSM。

将子系统成员条目放在 IMS 中。PROCLIB

IMS 中的每个 SSM 条目。PROCLIB 定义从 IMS 区域到其他队列管理器的连接。

要命名 SSM，请将 IMS IMSCTRL 宏的 IMS 标识字段的值 (1 到 4 个字母数字字符) 与站点定义的任何名称 (1 到 4 个字母数字字符) 并置。

一个 SSM 可以由所有 IMS 区域共享，也可以为每个区域定义特定成员。 此成员包含与外部子系统连接的条目数。 每个条目都是 80 个字符的记录。

位置参数

此条目中的字段为:

```
SSN,LIT,ESMT,RTT,REO,CRC
```

其中:

社保号
指定 IBM MQ 队列管理器名称。 它是必需的，并且必须包含 1 到 4 个字符。

lit
指定提供给 IMS 的语言接口令牌 (LIT)。 此字段是必需的，其值必须与 CSQQDEFV 模块中的值匹配。

ESMT
指定外部子系统模块表 (ESMT)。 此表指定 IMS 必须装入哪些连接模块。 CSQQESMT 是此字段的必需值。

RTT
IBM MQ 不支持此选项。

REO
指定在 IMS 应用程序引用不可操作的外部子系统时或在创建线程时资源不可用时要使用的区域错误选项 (REO)。 此字段是可选的，包含单个字符，可以是:

R
将返回码传递到应用程序，指示针对 IBM MQ 服务的请求失败。

Q
使用异常终止代码 U3051 结束应用程序，将活动回退到最后一个落实点，执行事务的 PSTOP，然后重新排队输入消息。 仅当 IMS 应用程序尝试引用不可操作的外部子系统时，或者当资源在创建线程时不可用时，此选项才适用。

如果在 IBM MQ 处理请求时发生 IBM MQ 问题，那么 IBM MQ 完成代码和原因码将返回到应用程序; 即，在适配器将请求传递到 IBM MQ 之后。

A
以异常终止代码 U3047 结束应用程序，并废弃输入消息。 仅当 IMS 应用程序引用不可操作的外部子系统时，或者如果资源在创建线程时不可用，那么此选项才适用。

如果在 IBM MQ 处理请求时发生 IBM MQ 问题，那么 IBM MQ 完成代码和原因码将返回到应用程序; 即，在适配器将请求传递到 IBM MQ 之后。

CRC
可以指定此选项，但 IBM MQ 不使用此选项。

注: 有关所有位置参数的完整详细信息，请参阅 如何向 IMS 指定外部子系统。

示例 SSM 条目为:

```
CSQ1,MQM1,CSQQESMT,,R,
```

其中：

CSQ1	随 IBM MQ 提供的缺省子系统名称。 您可以更改此设置以适合您的安装。
MQM1	CSQQDEFV 中提供的缺省 LIT。
CSQQESMT	外部子系统模块名称。 必须使用此值。
R	REO 选项。

关键字参数

可以使用关键字格式指定 IBM MQ 参数。 SST 参数的值可以是 DB2 或 MQ。 在 IMS 14 中添加了对 MQ 值的支持。 使用 MQ 可提高清晰度， IMS 子系统命令现在包含 SST 值，但否则不会产生任何显着影响。 如果需要，仍可以使用值 DB2 。 其他参数如 位置参数中所述，如以下示例所示：

```
SST=MQ,SSN=SYS3,LIT=MQM3,ESMT=CSQQESMT
```

其中：

SYS3	子系统名称
MQM3	CSQQDEFV 中提供的 LIT
CSQQESMT	外部子系统模块名称

指定 SSM EXEC 参数

在 IMS 控制区域的启动过程中指定 SSM EXEC 参数。 此参数指定一个字符到四个字符的子系统成员名 (SSM)。

如果为 IMS 控制区域指定 SSM ，那么在控制区域下运行的任何从属区域都可以连接到 IMS 中指定的 IBM MQ 队列管理器。由 SSM 参数指定的 PROCLIB 成员。 IMS。PROCLIB 成员名是与 SSM EXEC 参数中指定的 1 到 4 个字符并置的 IMS 标识 (IMSID= *xxxx*)。 IMS 标识是 IMSCTRL 生成宏的 IMS 标识参数。

IMS 允许您根据需要定义任意数量的外部子系统连接。 可以为不同的 IBM MQ 队列管理器定义多个连接。 所有 IBM MQ 连接必须位于同一 z/OS 系统中。 对于从属区域，可以指定从属区域 SSM ，也可以使用为控制区域指定的从属区域 SSM。 可以在从属区域 SSM 和控制区域 SSM 中指定不同的区域错误选项 (REO)。 第 826 页的表 84 显示了 SSM 规范的不同可能性。

表 *84: SSM* 规范选项

用于控制区域的 SMM	从属区域的 SMM	操作	注释
否	否	None	无法连接外部子系统。
否	Yes	None	无法连接外部子系统。
Yes	否	使用控制区域 SSM	在区域中调度的应用程序可以访问控制区域 SSM 中标识的外部子系统。 每个附件的出口和控制块将装入到控制区域和从属区域地址空间中。
Yes	是 (空)	没有 SSM 用于从属区域	此区域中调度的应用程序只能访问 DL/I 数据库。 将每个附件的出口和控制块装入到控制区域地址空间中。
Yes	是 (非空)	使用控制区域 SSM 检查从属区域 SSM	此区域中调度的应用程序只能访问两个 SSM 中标识的外部子系统。 每个附件的出口和控制块将装入到控制区域和从属区域地址空间中。

没有特定参数来控制 SSM 规范可能性的最大数目。

预装入 IMS 适配器

如果由 IMS 预装入 IMS 适配器，那么可以提高该适配器的性能。 预装入由 IMS 的 DFSMPLxx 成员控制。 PROCLIB: 请参阅 " IMS Administration Guide: System " 以获取更多信息。 要指定的 IBM MQ 模块名称为:

CSQACLST	CSQAMLST	CSQAPRH	CSQAVICM	CSQFSALM	CSQQDEFV
CSQQCONN	CSQQDISC	CSQQTERM	CSQQINIT	CSQQBACK	CSQQCMMT
CSQQESMT	CSQQPREP	CSQQTTHD	CSQQWAIT	CSQQNORM	CSQQSSOF
CSQQSSON	CSQFSTAB	CSQQRESV	CSQQSNOP	CSQQCMND	CSQQCVER
CSQQTMID	CSQQTRGI	CSQQCON2	CSQBPAPI	CSQBCRMH	CSQBAPPL

有关使用 IBM MQ classes for JMS 的更多信息，请参阅 在 IMS 中使用 IBM MQ classes for JMS。

IMS 的当前发行版仅支持从 MPP， BMP， IFP， JMP 和 JBP 区域中的 PDS-E 格式库预装入 IBM MQ 模块。 任何其他类型的 IMS 区域都不支持从 PDS-E 库预装入。 如果任何其他类型的区域都需要预装入，那么必须将提供的 IBM MQ 模块复制到 PDS 格式库。

z/OS 向 *IMS* 适配器定义 *IBM MQ* 队列管理器

必须在队列管理器定义表中定义 IBM MQ 队列管理器及其相应语言接口令牌 (LIT) 的名称。

使用提供的 CSQQDEFX 宏来创建 CSQQDEFV 装入模块。 第 827 页的图 120 显示了此汇编程序宏的语法。

```
CSQQDEFX TYPE=ENTRY|DEFAULT,NAME=qmgr-name,LIT=token
or
CSQQDEFX TYPE=END
```

图 *120: CSQQDEFX* 宏语法

参数

类型 = 条目 | 缺省值
指定 TYPE=ENTRY 或 TYPE=DEFAULT， 如下所示:

TYPE=ENTRY
指定要生成描述可用于 IMS 应用程序的 IBM MQ 队列管理器的表条目。 如果这是第一个条目，那么还会生成表头，包括 CSQQDEFV CSECT 语句。

类型 = 缺省值
至于 TYPE=ENTRY。 当 MQCONN 或 MQCONNX 指定全为空白的名称时，指定的队列管理器是要使用的缺省队列管理器。 表中必须只有一个此类条目。

名称 = *qmgr*-名称
指定队列管理器的名称，如 **MQCONN** 或 **MQCONNX** 所指定。

LIT = 令牌
指定 IMS 用于标识队列管理器的语言接口令牌 (LIT) 的名称。

MQCONN 或 MQCONNX 调用将 *name* 输入参数和 *hconn* 输出参数与名称标签相关联，因此与 CSQQDEFV 条目中的 LIT 相关联。 传递 *hconn* 参数的其他 IBM MQ 调用使用 MQCONN 或 MQCONNX 调用中标识的 CSQQDEFV 条目中的 LIT， 将调用定向到具有相同 LIT 的 IMS SSM PROCLIB 成员中定义的 IBM MQ 队列管理器。

总之， MQCONN 或 MQCONNX 调用上的 **name** 参数标识 CSQQDEFV 中的 LIT， 而 SSM 成员中的相同 LIT 标识 IBM MQ 队列管理器。 (有关 MQCONN 调用的信息，请参阅 MQCONN-Connect 队列管理器。 有关 MQCONNX 调用的信息，请参阅 MQCONNX-Connect 队列管理器 (扩展)。)

类型 = 结束
指定表已完成。 如果省略此参数，那么将采用 TYPE=ENTRY。

使用 CSQQDEFX 宏

第 828 页的图 121 显示了队列管理器定义表的常规布局。

```
CSQQDEFX NAME=subsystem1,LIT=token1
CSQQDEFX NAME=subsystem2,LIT=token2,TYPE=DEFAULT
CSQQDEFX NAME=subsystem3,LIT=token3
...
CSQQDEFX NAME=subsystemN,LIT=tokenN
CSQQDEFX TYPE=END
END
```

图 *121:* 队列管理器定义表的布局

z/OS 设置 *IMS* 触发器监视器

您可以设置面向 IMS 批处理的程序以监视 IBM MQ 启动队列。

使用 thlqual.SCSQPROC 库中的模型 CSQQTAPL 将应用程序定义到 IMS (请参阅 CSQQTRMN 的事务定义示例)。

使用 thlqual.SCSQPROC 库中的模型 CSQQTPSB 生成 PSB 和 ACB (请参阅 CSQQTRMN 的示例 PSB 定义)。

```
*  This is the application definition *
*  for the IMS Trigger Monitor BMP    *

APPLCTN PSB=CSQQTRMN,
PGMTYPE=BATCH,
SCHDTYP=PARALLEL
```

图 *122: CSQQTRMN* 的事务定义示例

```
PCB  TYPE=TP,      ALTPCB for transaction messages
MODIFY=YES,        To "triggered" IMS transaction
PCBNAME=CSQQTRMN
PCB  TYPE=TP,      ALTPCB for diagnostic messages
MODIFY=YES,        To LTERM specified or "MASTER"
PCBNAME=CSQQTRMG,
EXPRESS=YES
PSBGEN LANG=ASSEM,
PSBNAME=CSQQTRMN,  Runs program CSQQTRMN
CMPAT=YES
```

图 *123: CSQQTRMN* 的示例 *PSB* 定义

有关启动和停止 IMS 触发器监视器的更多信息，请参阅 控制 IMS 触发器监视器。

z/OS 设置 IMS 网桥

IBM MQ - IMS 网桥是一个可选组件，它使 IBM MQ 能够在未启用 IBM MQ 的现有程序和事务之间进行输入和输出。

本主题描述了定制 IBM MQ - IMS 网桥必须执行的操作。

定义 IBM MQ 的 XCF 和 OTMA 参数。
此步骤定义 IBM MQ 系统的 XCF 组和成员名以及其他 OTMA 参数。 IBM MQ 和 IMS 必须属于同一 XCF 组。 使用 CSQ6SYSP 宏的 OTMACON 关键字来定制系统参数装入模块中的这些参数。

请参阅 使用 CSQ6SYSP 以获取更多信息。

将 XCF 和 OTMA 参数定义到 IMS。
此步骤定义 IMS 系统的 XCF 组和成员名。 IMS 和 IBM MQ 必须属于同一 XCF 组。

将以下参数添加到 IMS 参数列表中 (在 JCL 中或在 IMS PROCLIB 中的成员 DFSPBxxx 中):

OTMA=Y
这将在启动 IMS 时自动启动 OTMA。 (可选，如果指定 OTMA=N ，那么还可以通过发出 IMS 命令 /START OTMA 来启动 OTMA。)

名称 =
此参数提供 XCF 组名。

它与存储类定义 (请参阅下一步) 和 CSQ6SYSP 宏的 OTMACON 关键字的 **Group** 参数中指定的组名相同。

OTMANM=
此参数提供 IMS 系统的 XCF 成员名。

这与存储类定义中指定的成员名相同 (请参阅下一步)。

告诉 IBM MQ IMS 系统的 XCF 组和成员名。
这是由队列的存储类指定的。 如果要通过 IBM MQ - IMS 网桥发送消息，那么在定义队列的存储类时必须指定此项。 在存储类中，必须定义目标 IMS 系统的 XCF 组和成员名。 要执行此操作，请使用 IBM MQ 操作和控制面板，或者使用 IBM MQ 命令，如 可编程命令格式简介中所述。

设置所需的安全性。
/SECURE OTMA IMS 命令确定要应用于通过 OTMA 连接到 IMS 的 **每** IBM MQ 队列管理器的安全性级别。 请参阅 将 IBM MQ 与 IMS 配合使用的安全注意事项 以获取更多信息。

将其他 IMS 连接添加到同一队列管理器

要将 IMS 连接添加到同一队列管理器，必须定义第二个存储类 (STGCLASS) 以指向新的 IMS; 请参阅 DEFINE STGCLASS 以获取更多信息。

要点:

- 一个本地队列不能指向两个存储类。
- 一个存储类不能指向两个 IMS 网桥。
- IBM MQ 和 IMS 必须属于同一 XCF 组。 使用 CSQ6SYSP 宏的 OTMACON 关键字来定制系统参数装入模块中的这些参数。

请参阅 使用 CSQ6SYSP 以获取更多信息。

相关概念
IBM MQ 和 IMS
第 823 页的『设置 IMS 适配器』
要在 IMS 中使用 IBM MQ ，需要 IBM MQ - IMS 适配器 (通常称为 IMS 适配器)。

IBM MQ for z/OS 上的 IMS 和 IMS 网桥应用程序

z/OS 将 IBM MQ 与 CICS 结合使用

要将 IBM MQ 与 CICS 配合使用，必须配置 IBM MQ CICS 适配器以及 (可选) IBM MQ CICS bridge 组件。

有关配置 IBM MQ CICS 适配器和 IBM MQ CICS bridge 组件的更多信息，请参阅 CICS 文档的 配置与 MQ 的连接 部分。

相关概念
IBM MQ 和 CICS
第 823 页的『将 IBM MQ 用于 IMS』
IBM MQ -IMS 适配器和 IBM MQ - IMS 网桥是允许 IBM MQ 与 IMS 进行交互的两个组件。

相关参考
第 830 页的『将服务升级并应用于 Language Environment 或 z/OS Callable Services』
您必须执行的操作根据您使用 CALLLIBS 还是 LINK 以及 SMP/E 的版本而有所不同。

z/OS 将服务升级并应用于 Language Environment 或 z/OS Callable Services

您必须执行的操作根据您使用 CALLLIBS 还是 LINK 以及 SMP/E 的版本而有所不同。

下表显示了在升级以下产品的级别或将服务应用于这些产品时需要对 IBM MQ for z/OS 执行的操作:

- 语言环境
- z/OS 可调用服务 (例如， APPC 和 RRS)

表 *85:* 服务已应用或产品已升级到新发行版

产品	**使用 CALLLIBS 和 SMP/E V3r2 或更高版本时的操作** **注: 您不需要为 Language Environment 和 Callable 服务运行单独的作业。 一份工作就够了**	**使用 LINK 时的操作**
语言环境	1. 将 SMP/E 作业上的边界设置为目标区域。 2. 在 SMPCNTL 卡上，指定 LINK LMODS CALLLIBS。 您还可以指定其他参数，例如 CHECK， RETRY (YES) 和 RC。 请参阅 *SMP/E for z/OS: Commands* 以获取更多信息。 3. 运行 SMP/E 作业。	如果设置了 SMP/E 区域以进行自动重新链接，并且已运行 CSQ8SLDQ 作业，那么无需执行任何操作。
可调用服务	1. 将 SMP/E 作业上的边界设置为目标区域。 2. 在 SMPCNTL 卡上，指定 LINK LMODS CALLLIBS。 您还可以指定其他参数，例如 CHECK， RETRY (YES) 和 RC。 请参阅 *SMP/E for z/OS: Commands* 以获取更多信息。 3. 运行 SMP/E 作业。	如果设置了 SMP/E 区域以进行自动重新链接，并且已运行 CSQ8SLDQ 作业，那么无需执行任何操作。

表 86: 其中一个产品已更新为新的 *SMP/E* 环境和库中的新发行版

产品	使用 CALLLIBS 和 SMP/E V3r2 或更高版本时的操作 注: 您不需要为 Language Environment 和 Callable 服务运行三个单独的作业。 一项工作对这两种产品都足够了。	使用 LINK 时的操作
语言环境	1. 更改 SCEELKED 和 SCEESPC 的 DDDEF 以指向新库。 2. 将 SMP/E 作业上的边界设置为目标区域。 3. 在 SMPCNTL 卡上，指定 LINK LMODS CALLLIBS。 您还可以指定其他参数，例如 CHECK， RETRY (YES) 和 RC。 请参阅 *SMP/E for z/OS: Commands* 以获取更多信息。 4. 运行 SMP/E 作业。	1. 删除 IBM MQ for z/OS 目标区域中以下 LMOD 条目的 XZMOD 子条目: CSMQXDCST， CMQXRCTL， CMQXSUPR，CSQCBE00， CSQCBE30， CSQCBP00，CSQCBP10， CSQCBR00， CSQUCVX，CSQUDLQH， CSQVXPCB， CSQVXSPT，CSQXDCST， CSQXRCTL， CSQXSUPR，CSQXTQCMI， CSQXTCP， CSQXTNSV，CSQ7DRPS，CSQCSQCSQCSQXQHRQCSQCSQCQCCQCQCQCQCQCQCQCQCQCQCQCSQCQCQCSQCSQ 2. 在 IBM MQ 区域和语言环境区域之间设置相应的 ZONEINDEXs。 3. 定制 CSQ8SLDQ 以引用 LINK 命令的 FROMZONE 参数上的新区域。 可以在 SCSQINST 库中找到 CSQ8SLDQ 。 4. 运行 CSQ8SLDQ。
可调用服务	1. 更改 CSSLIB 的 DDDEF 以指向新库 2. 将 SMP/E 作业上的边界设置为目标区域。 3. 在 SMPCNTL 卡上，指定 LINK LMODS CALLLIBS。 您还可以指定其他参数，例如 CHECK， RETRY (YES) 和 RC。 请参阅 *SMP/E for z/OS: Commands* 以获取更多信息。 4. 运行 SMP/E 作业。	1. 删除 IBM MQ for z/OS 目标区域中以下 LMOD 条目的 XZMOD 子条目: CMQXRCTL， CMQXSUPR， CSQBSRV，CSQILPLM， CSQXJST， CSQXRCTL，CSQXSUPR， CSQ3AMGP， CSQ3EPX，CSQ3REPL 2. 在 IBM MQ 专区和可调用服务专区之间设置相应的 ZONEINDEX。 3. 定制 CSQ8SLDQ 以引用 LINK 命令的 FROMZONE 参数上的新区域。 可以在 SCSQINST 库中找到 CSQ8SLDQ 。 4. 运行 CSQ8SLDQ。

有关使用 CALLLIBS 时用于重新链接模块的作业的示例，请参阅 第 831 页的『运行 LINK CALLLIBS 作业』。

z/OS 运行 LINK CALLLIBS 作业

使用 CALLLIBS 时用于重新链接模块的示例作业。

以下是在 SMP/E V3r2 系统上使用 CALLLIBs 时用于重新链接模块的作业示例。 必须提供 JOBCARD 以及包含 IBM MQ for z/OS 的 SMP/E CSI 的数据集名称。

```
//*************************************************************
//* RUN LINK CALLLIBS.
//*************************************************************
//CALLLIBS EXEC PGM=GIMSMP,REGION=4096K
//SMPCSI  DD DSN=your.csi
//       DISP=SHR
//SYSPRINT DD SYSOUT=*
//SMPCNTL DD *
SET BDY(TZONE).
LINK LMODS CALLLIBS .
/*
```

图 *124:* 示例 *SMP/E LINK CALLLIBS* 作业

z/OS 在 IMS 中使用 OTMA 出口

如果要将 IMS Open Transaction Manager Access 出口与 IBM MQ for z/OS 配合使用，请使用本主题。

如果要将输出从 IMS 事务发送到 IBM MQ，并且该事务并非源自 IBM MQ，那么需要对一个或多个 IMS OTMA 出口进行编码。

同样，如果要将输出发送到非 OTMA 目标，并且事务确实源自 IBM MQ，那么还需要对一个或多个 IMS OTMA 出口进行编码。

IMS 中提供了以下出口，使您能够在 IMS 和 IBM MQ 之间定制处理:

- OTMA 预路由出口
- 目标解析用户 (DRU) 出口

OTMA 出口名称

必须将预路由出口命名为 DFSYPRX0。 只要 DRU 出口与 IMS 中已有的模块名称不冲突，就可以对其进行任何命名。

指定目标解析用户出口名称

可以使用 CSQ6SYSP 宏的 OTMACON 关键字的 *Druexit* 参数来指定要由 IMS 运行的 OTMA DRU 出口的名称。

要简化对象标识，请考虑采用命名约定 DRU0xxxx，其中 xxxx 是 IBM MQ 队列管理器的名称。

如果未在 OTMACON 参数中指定 DRU 出口的名称，那么缺省值为 DFSYDRU0。 请参阅 DFSYDRU0 以获取更多信息。

IMS 目标的命名约定

您需要将 IMS 程序的输出发送到的目标的命名约定。 这是在 IMS 应用程序的 CHNG 调用中设置的目标，或者在 IMS PSB 中预设的目标。

OTMA 出口的样本方案

对于 IMS 的预路由出口和目标路由出口的示例，请使用以下主题:

- 第 833 页的『预路由出口 DFSYPRX0』
- 第 834 页的『目标解析用户出口』

要简化标识，请使 OTMA 目标名称类似于 IBM MQ 队列管理器名称，例如 IBM MQ 队列管理器名称重复。 在此情况下，如果 IBM MQ 队列管理器名称为 " **VCPE** "， CHNG 调用设置的目标为" **VCPEVCPE** "。

相关概念

IBM MQ 和 IMS

第 823 页的『将 IBM MQ 用于 IMS』
IBM MQ -IMS 适配器和 IBM MQ - IMS 网桥是允许 IBM MQ 与 IMS 进行交互的两个组件。

IBM MQ for z/OS 上的 IMS 和 IMS 网桥应用程序

z/OS 预路由出口 DFSYPRX0

本主题包含 IMS 中 OTMA 的样本预路由出口。

必须首先对预路由出口 DFSYPRX0 进行编码。 请参阅 OTMA 目标解析用户出口 (DFSYPRX0 和其他 OTMAYPRX 类型出口)，以获取 IMS 传递到此例程的参数。

此出口测试消息是否用于已知 OTMA 目标 (在我们的示例 VCPEVCPE 中)。 如果是，那么出口必须检查发送消息的事务是否源自 OTMA。 如果消息源自 OTMA，那么它将具有 OTMA 头，因此您应该退出 DFSYPRX0，并将寄存器 15 设置为零。

- 如果发送消息的事务并非源自 OTMA，那么必须将客户机名称设置为有效的 OTMA 客户机。 这是要向其发送消息的 IBM MQ 队列管理器的 XCF 成员名。 您应该将客户机名称 (在 CSQ6SYSP 宏的 OTMACON 参数中) 设置为队列管理器名称。 这是缺省值。 然后，应该退出 DFSYPRX0，将寄存器 15 设置为 4。
- 如果发送消息的事务源自 OTMA，并且目标是非 OTMA，那么应将注册 15 设置为 8 并退出。
- 在所有其他情况下，应将寄存器 15 设置为零。

如果将 OTMA 客户机名称设置为 IMS 未知的名称，那么应用程序 CHNG 或 ISRT 调用将返回 A1 状态码。

对于与多个 IBM MQ 队列管理器进行通信的 IMS 系统，应该对每个 IBM MQ 队列管理器重复该逻辑。

第 834 页的图 125 中显示了样本汇编程序代码:

```
TITLE 'DFSYPRX0: OTMA PRE-ROUTING USER EXIT'
DFSYPRX0 CSECT
DFSYPRX0 AMODE 31
DFSYPRX0 RMODE ANY
*
SAVE (14,12),,DFSYPRX0&SYSDATE&SYSTIME
SPACE 2
LR  R12,R15        MODULE ADDRESSABILITY
USING DFSYPRX0,R12
*
L   R2,12(,R1)      R2 -> OTMA PREROUTE PARMS
*
LA  R3,48(,R2)      R3 AT ORIGINAL OTMA CLIENT (IF ANY)
CLC  0(16,R3),=XL16'00'  OTMA ORIG?
BNE  OTMAIN        YES, GO TO THAT CODE
*
NOOTMAIN DS 0H            NOT OTMA INPUT
LA  R5,8(,R2)       R5 IS AT THE DESTINATION NAME
CLC  0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE  EXIT0         NO, NORMAL PROCESSING
*
L   R4,80(,R2)      R4 AT ADDR OF OTMA CLIENT
MVC  0(16,R4),=CL16'VCPE' CLIENT OVERRIDE
B   EXIT4         AND EXIT
*
OTMAIN  DS 0H            OTMA INPUT
LA  R5,8(,R2)       R5 IS AT THE DESTINATION NAME
CLC  0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE  EXIT8         NO, NORMAL PROCESSING
```

```
*
EXIT0  DS 0H
LA  R15,0         RC = 0
B   BYEBYE
*
EXIT4  DS 0H
LA  R15,4         RC = 4
B   BYEBYE
*
EXIT8  DS 0H
LA  R15,8         RC = 8
B   BYEBYE
*
BYEBYE  DS 0H
RETURN (14,12),,RC=(15)   RETURN WITH RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
END
```

图 *125: OTMA* 预路由出口汇编程序样本

z/OS 目标解析用户出口

本主题包含 IMS 的样本目标解析用户出口。

如果在 DFSYPRX0 中将寄存器 15 设置为 4，或者如果事务的源是 OTMA **和**，那么将寄存器 15 设置为零，那么将调用 DRU 出口。 在此示例中， DRU 出口名称为 DRU0VCPE。

DRU 出口检查目标是否为 VCPEVCPE。 如果是，那么将按如下所示设置 OTMA 用户数据 (在 OTMA 前缀中):

偏移量
OTMA 用户数据

(十进制)

0
OTMA 用户数据长度 (在此示例中为 334)

2
MQMD

326
回复格式

这些偏移是 IBM MQ - IMS 网桥期望找到此信息的位置。

DRU 出口应尽可能简单。 因此，在此样本中，源自特定 IBM MQ 队列管理器的 IMS 的所有消息都将放入同一 IBM MQ 队列。

如果消息需要持久，那么 IMS 必须使用同步事务管道。 为此， DRU 出口必须设置 OUTPUT 标志。 有关更多信息，请参阅 指定 IBM MQ 的同步 tpipe 。

编写 IBM MQ 应用程序以处理此队列，并使用 MQMD 结构， MQIIH 结构 (如果存在) 或用户数据中的信息将每条消息路由到其目标。

第 835 页的图 126 中显示了样本汇编程序 DRU 出口。

```
TITLE 'DRU0VCPE: OTMA DESTINATION RESOLUTION USER EXIT'
DRU0VCPE CSECT
DRU0VCPE AMODE 31
DRU0VCPE RMODE ANY
*
SAVE (14,12),,DRU0VCPE&SYSDATE&SYSTIME
SPACE 2
LR  R12,R15            MODULE ADDRESSABILITY
USING DRU0VCPE,R12
*
L   R2,12(,R1)           R2 -> OTMA DRU PARMS
*
L   R5,88(,R2)           R5 ADDR OF OTMA USERDATA
LA  R6,2(,R5)           R6 ADDR OF MQMD
USING MQMD,R6            AS A BASE
*
LA  R4,MQMD_LENGTH+10       SET THE OTMA USERDATA LEN
STH  R4,0(,R5)           = LL + MQMD + 8
*                       CLEAR REST OF USERDATA
MVI  0(R6),X'00'          ...NULL FIRST BYTE
MVC  1(255,R6),0(R6)        ...AND PROPAGATE IT
MVC  256(MQMD_LENGTH-256+8,R6),255(R6) ...AND PROPAGATE IT
*
VCPE   DS  0H
CLC  44(16,R2),=CL16'VCPE'     IS DESTINATION VCPE?
BNE  EXIT4             NO, THEN DEST IS NON-OTMA
MVC  MQMD_REPLYTOQ,=CL48'IMS.BRIDGE.UNSOLICITED.QUEUE'
MVC  MQMD_REPLYTOQMGR,=CL48'VCPE'  SET QNAME AND QMGRNAME
MVC  MQMD_FORMAT,MQFMT_IMS     SET MQMD FORMAT NAME
MVC  MQMD_LENGTH(8,R6),MQFMT_IMS_VAR_STRING
*                       SET REPLYTO FORMAT NAME
B   EXIT0
*
EXIT0  DS  0H
LA  R15,0             SET RC TO OTMA PROCESS
B   BYEBYE             AND EXIT
*
EXIT4  DS  0H
LA  R15,4             SET RC TO NON-OTMA
B   BYEBYE             AND EXIT
*
BYEBYE  DS  0H
RETURN (14,12),,RC=(15)       RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
CMQA  EQUONLY=NO
CMQMDA DSECT=YES
SPACE 2
END
```

图 *126:* 样本汇编程序 *DRU* 出口

z/OS 使用 IBM z/OSMF 自动执行 IBM MQ

IBM z/OS Management Facility (z/OSMF) 在具有集成用户帮助的面向任务的基于 Web 浏览器的用户界面中提供系统管理功能，以便您可以更轻松地管理大型机 z/OS 系统的日常操作和管理。

通过精简某些传统任务并使其他任务自动化， z/OSMF 可以帮助简化 z/OS 系统管理的某些方面。

可以在单击按钮时从用户提供的门户网站供应或取消供应资源。 z/OSMF 提供 REST API 以帮助执行此任务。

z/OSMF 随附的样本市场门户网站也可用于供应和取消供应资源。 或者，更有经验的用户可以使用 z/OSMF Web 用户界面 (WUI)。

本部分假定您了解 z/OSMF，但如果您不熟悉 z/OSMF ，那么应阅读 z/OSMF 入门。 或者，您可以从 z/OSMF WUI 联机帮助访问此部分。

您应该熟悉 z/OS Cloud 配置，即:

- 云供应- 资源管理服务
- 工作负载管理-请参阅 IBM z/OS Management Facility Programming Guide 以获取更多信息。
- 入门-请参阅 入门教程-云

z/OSMF 2.2 引入了基于角色的活动和任务，因此了解以下概念很重要:

域
管理人员
approvers
租户
模板
个实例
工作流程

等等。

提供了样本 IBM MQ z/OSMF 工作流程和关联文件，可以将其作为 IBM MQ for z/OS UNIX System Services Components 功能部件的一部分进行安装。 IBM MQ for z/OS 程序目录 有关程序目录的下载链接，请参阅 IBM MQ for z/OS 程序目录 PDF 文件。 中描述了此功能部件的安装过程以及目录和文件结构。

样本工作流程以 XML 编写，演示如何自动供应 (创建) 或取消供应 (破坏) IBM MQ 队列管理器，通道启动程序和本地队列，以及如何对供应的 IBM MQ 资源执行操作。 工作流程中的步骤将提交作业 (JCL) ，运行 REXX execs ，处理 Shell 脚本或发出 REST API 调用。

这些样本旨在说明可使用 z/OSMF 实现的函数类型。 预计 z/OSMF 工作流程通常将用于供应资源，而诸如 put 或 get 消息之类的操作实质上将使用 IBM MQ 应用程序来执行。

您可以按提供的方式运行样本工作流程，前提是已设置工作流程变量属性 (如以下部分中所述) ，或者您可以根据需要对其进行定制。 您可能更愿意编写自己的工作流程以执行其他功能。 在运行样本工作流程之前，请参阅:

- 第 836 页的『z/OSMF 的先决条件』
- 第 838 页的『安全设置』
- 第 840 页的『限制』

向以下用户提供了样本工作流程应用程序:

- 第 841 页的『自动供应或取消供应 IBM MQ 队列管理器，并对供应的队列管理器执行操作』
- 第 842 页的『自动供应或取消供应 IBM MQ 本地队列，并对供应的队列执行操作』.

相关概念
第 734 页的『设置 IBM MQ for z/OS』
使用本主题作为定制 IBM MQ for z/OS 系统的逐步指南。

z/OS z/OSMF 的先决条件

使用 IBM MQ 运行 IBM z/OS Management Facility (z/OSMF) 所需的先决条件

IBM MQ for z/OS 9.1.0 中提供的工作流程利用 z/OSMF 中的新功能，此功能是通过 z/OS 2.1 和 2.2 上的 APAR 提供的。 以下文本提供了更多详细信息。

1. 您已正确安装和配置 IBM z/OS Management Facility 2.2 。 如果在启用安全性的情况下运行，请确保已配置 z/OSMF 记录的所有安全性设置。
2. 您已安装以下 APAR:

   z/OS 2.1

   - PI71068
   - PI71079
   - PI71082
   - PI71084
   - OA50130

   z/OS 2.2

   - PI70526
   - PI70521
   - PI70527
   - PI67839
   - PI70767
   - PI46315
   - OA49081
   - OA49802
   - OA50130
3. 已配置 z/OSMF Angel 进程 (如果需要) 和服务器进程。
4. 已配置 z/OS 云环境 (如上文简要讨论并由 z/OSMF 记录)
5. IBM MQ for z/OS 9.0.1 已安装，并且产品装入库可用。
6. 已执行以下 IBM MQ 队列管理器定制任务:

任务	描述
1	标识 z/OS 系统参数
2	APF 授权 IBM MQ 装入库
3	更新 z/OS 链接列表和 LPA
4	更新 z/OS 程序属性表

7. 样本工作流程和关联文件安装在合适的 z/OS UNIX System Services (z/OS UNIX) 目录中。
8. `/tmp` z/OS UNIX 目录可用，因为 provision.xml 工作流程可能会在此目录中创建临时文件。 如果创建了文件，那么工作流程通常会在使用后删除该文件。
9. `deprovision.xml` 文件中包含用于调用 CSQ4ZWS1.rexx 和 CSQ4ZWS2.rexx REXX execs 的步骤。 这些 execs 等待队列管理器和通道启动程序子系统停止; execs 调用 z/OS UNIX **SLEEP** 命令作为系统调用。

   根据您的 z/OS UNIX 配置，您可能会发现 **SLEEP** 命令无法以编码方式工作。 如果在处理期间迁到指示找不到 **SLEEP** 命令的错误，那么可以尝试在 execs CSQ4ZWS1.rexx 和 CSQ4ZWS2.rexx:

```
CALL SYSCALLS('ON')            /* Enable z/OS UNIX calls    */
ADDRESS SYSCALL
"SLEEP" 10                          /* Sleep for 10 seconds */
CALL SYSCALLS 'OFF'            /* Disable z/OS UNIX calls    */
```

   替换为

```
'sleep' 10
```

然后，发出 Open MVS (OMVS) **env** 命令以检查 PATH 环境变量设置。 确保将包含 **sleep** 命令的目录定义到 PATH。 请注意， **sleep** 命令通常位于 `/bin` 目录中。

10. 确保已启动 z/OSMF。

    必须启动 Angel 进程和服务器 z/OSMF 进程，并且 z/OSMF Web 用户界面 (WUI) 已启动并正在运行。 有关更多详细信息，请参阅 Liberty 概要文件: z/OS 上的进程类型。

    即使您打算使用 REST API 来驱动工作流程，也需要启动 z/OSMF WUI。 z/OSMF WUI 对于监视工作流程的创建和执行很有用。

相关概念

第 835 页的『使用 IBM z/OSMF 自动执行 IBM MQ』

IBM z/OS Management Facility (z/OSMF) 在具有集成用户帮助的面向任务的基于 Web 浏览器的用户界面中提供系统管理功能，以便您可以更轻松地管理大型机 z/OS 系统的日常操作和管理。

z/OS 安全设置

运行 z/OSMF 所需的安全设置。

在属性文件中定义了以下用户标识变量属性。 有关更多详细信息，请参阅 第 844 页的『运行工作流程』。

用户标识属性	描述
CSQ_USERID	用于运行工作流程步骤的用户标识。 但是，请注意，根据以下文本中列出的 **CSQ_ADMIN_*** 用户标识的设置，所选步骤 (通常需要提升权限级别) 将使用不同的用户标识来运行。 正在使用的用户标识由工作流程中相应步骤的 **runAsUser** 属性标识。
CSQ_ADMIN_APF_USERID	APF 授权包含队列管理器系统参数模块的装入库时要使用的用户标识。
CSQ_APF_APPROVAL_ID	用于允许用户以用户 CSQ_ADMIN_APF_USERID 身份运行数据集 APF 授权步骤的核准标识。
CSQ_ADMIN_CONSOLE_USERID	在发出 z/OS 控制台命令的情况下运行步骤时使用的用户标识。 **注意:** 需要允许此用户标识对启动式任务概要文件 (MVS.START.STC.*) 进行 UPDATE 访问 在 OPERCMDS 类中。 请参阅 z/OS 文档中的控制操作程序命令的使用 以获取更多信息。
CSQ_CONSOLE_APPROVAL_ID	用于允许用户以用户 CSQ_ADMIN_CONSOLE_USERID 身份运行在运行下发出 z/OS 控制台命令的步骤的核准标识。
CSQ_ADMIN_SAF_USERID	发出 SAF 命令时要使用的用户标识。
CSQ_SAF_APPROVAL_ID	用于允许用户以用户 CSQ_ADMIN_SAF_USERID 身份运行 SAF 命令步骤的核准标识。
CSQ_ADMIN_SSI_USERID	发出 SETSSI 命令以标识要供应给 z/OS 的子系统时要使用的用户标识。
CSQ_SSI_APPROVAL_ID	用于允许用户以用户 CSQ_ADMIN_SSI_USERID 身份在运行下运行 SETSSI 命令步骤的核准标识。

注: 用于运行供应和取消供应工作流程的用户标识需要具有以下列出的足够权限:

1. 队列管理器供应和取消供应工作流程使用 SETPROG 命令对数据集进行 APF 授权。 用户标识是在属性 CSQ_ADMIN_APF_USERID 中设置的，或者需要允许用于运行工作流程的用户标识发出此命令。 可以通过发出以下命令来实现此目的:

```
PERMIT MVS.SETPROG CLASS(OPERCMDS) ID(value of CSQ_ADMIN_APF_USERID) ACCESS(UPDATE)
```

注: SETPROG 命令可能不会在 z/OS 系统的 IPL 期间持久存储，因此可能需要在 IPL 之后手动发出以下 SETPROG 命令:

```
SETPROG APF,ADD,DSN=value of CSQ_AUTH_LIB_HLQ.value of CSQ_SSID.APF.LOAD,SMS
```

有关 SETPROG 命令的更多详细信息，请参阅 使用 RACF 来控制 APF 列表。

此外，您可能已启用 FACILITY 类来控制哪些库可以获得 APF 授权，因此您可能需要发出以下命令:

```
PERMIT CSVAPF.libname CLASS(FACILITY) ID(value of CSQ_ADMIN_APF_USERID)
       ACCESS(UPDATE)
```

2. 队列管理器供应工作流程中的步骤发出 SETSSI 命令以向 z/OS 标识 IBM MQ 子系统。 需要允许属性 CSQ_ADMIN_SSI_USERID 中设置的用户标识使用此命令。 可以通过发出以下命令来实现此目的:

```
PERMIT MVS.SETSSI.ADD CLASS(OPERCMDS) ID(value of CSQ_ADMIN_SSI_USERID)
        ACCESS(CONTROL)
```

注: 已通过 SETSSI 命令向 z/OS 标识的子系统不会在 z/OS 系统的 IPL 期间持久存储。 因此，可能需要在 IPL 之后手动发出以下 SETSSI 命令:

```
SETSSI  ADD,S='value of CSQ_SSID',I=CSQ3INI,
       P='CSQ3EPX,value of CSQ_CMD_PFX,S'
```

有关 SETSSI 命令的更多详细信息，请参阅: SETSSI 命令。

3. 工作流程会发出队列管理器命令，因此如果您计划启用安全性，那么需要向属性 CSQ_ADMIN_RACF_USERID 中设置的用户标识 (或用于运行工作流程的用户标识) 授予对 MQADMIN 或 MXADMIN 类的 CLAUTH (客户机认证) 权限 (取决于所使用的类)。 这是为了允许此用户标识对这些类定义安全概要文件。 可以通过发出以下命令来实现此目的:

```
ALTUSR value of CSQ_ADMIN_RACF_USERID CLAUTH(MQADMIN)
```

有关 **CLAUTH** 的更多详细信息，请参阅 CLAUTH (类权限) 属性。

4. deprovision.xml 工作流程发出 z/OS 命令，例如 DISPLAY ACTIVE 作业， CANCEL 或 FORCE 子系统，因此属性 CSQ_ADMIN_CONSOLE_USERID 中设置的用户标识 (或用于运行工作流程的用户标识) 需要具有发出此类命令的适当权限。
5. 使用 "软件服务" 任务的模板表请求队列管理器实例的用户必须有权访问 z/OSMF 和配置助手 (由 z/OSMF 定义)。
6. 供应队列管理器的使用者的用户标识需要权限才能从使用变量 CSQ_PROC_LIB 定义的 PROCLIB 数据集中添加和删除成员。
7. 必须先供应队列管理器，再供应队列。
8. 要使用 `queueLoad.xml` 和 `queueOffload.xml` 工作流程，需要提前定义所使用的数据集。 此外，需要向用于运行这些工作流程的用户标识授予对数据集的 UPDATE 权限。
9. 队列管理器 `provision.xml` 工作流程中的步骤当前禁用子系统安全性。 您可以修改作业 `csq4znse.jcl` 以通过添加用于保护 IBM MQ 资源的相应安全性命令来启用子系统安全性。 但是，请注意，如果添加其他命令，那么还需要添加命令以删除 `csq4dse.jcl` 中的安全许可权 (由 deprovision.xml 工作流程提交)。

   注: 此步骤会发出 RACF 安全性命令。 如果您正在使用备用安全产品，那么需要修改此步骤以针对安全产品发出相应的命令。

网络需求

添加队列管理器模板以及该模板的资源时，需要单击 **创建网络资源池**。 这将创建具有此模板的网络资源的资源池。

通过使用配置助手，网络管理员需要通过定义要为此模板分配的端口数的限制来完成此网络资源池定义。

对于每个模板实例， `provision.xml` 工作流程会在该范围内分配一个端口，并启动侦听器以侦听该端口。

使用 IBM 工作负载管理器进行分类

如果要使用 WLM 对队列管理器和通道启动程序地址空间进行分类，那么在添加用于供应队列管理器的模板时需要指定此项。

是否分类由文件 `workflow_variables.properties` 中设置的标志 **CSQ_DEFINE_MSTR_WLM_RULE** 和 **CSQ_DEFINE_CHIN_WLM_RULE** 控制。

有关使用 WLM 进行分类的更多信息，请参阅 *z/OSMF* 配置指南。

相关概念

第 836 页的『z/OSMF 的先决条件』

使用 IBM MQ 运行 IBM z/OS Management Facility (z/OSMF) 所需的先决条件

z/OS 限制

将 z/OSMF 与 IBM MQ 配合使用时的限制。

1. `provision.xml` 工作流程当前自动执行以下突出显示的队列管理器定制任务:

任务	描述
1	标识 z/OS 系统参数
2	APF 授权 IBM MQ 装入库 **(provision.xml 执行 APF 授权某些库)**
3	更新 z/OS 链接列表和 LPA
4	更新 z/OS 程序属性表
5	**将 IBM MQ 子系统定义到 z/OS**
6	**为 IBM MQ 队列管理器创建过程**
7	**为通道启动程序创建过程**
8	**将 IBM MQ 子系统定义到 z/OS WLM 服务类**
9	选择并设置耦合设施卸载存储环境
10	设置耦合设施
11	实施 ESM 安全控制
12	更新 SYS1.PARMLIB 成员
13	**定制初始化输入数据集**
14	**创建引导程序和日志数据集**
15	**定义页面集**
16	将 IBM MQ 条目添加到 Db2 数据共享组
17	**定制系统参数模块 (某些)**
18	**定制通道启动程序参数 (部分)**
19	设置批处理， TSO 和 RRS 适配器
20	设置操作和控制面板
21	包含 IBM MQ 转储格式化成员
22	禁止参考消息

任务	描述
23	更新 Advanced Message Security 的系统 DIAG 成员
24	为 Advanced Message Security 创建过程
25	设置启动式任务用户 "高级消息安全性"
26	将 RACDCERT 许可权授予 Advanced Message Security 的安全性管理员
27	授予用户对 Advanced Message Security 的资源许可权

2. 需要手动执行未以粗体文本突出显示的定制任务 (如果需要)。
3. 样本 INP1 和 INP2 成员当前按现有方式使用。 如果需要，可以定义其他属性来控制这些成员定义的资源。
4. 与属性文件中列出的特定属性相关的注释指示使用这些属性的任何限制。 有关更多详细信息，请参阅 第 844 页的『运行工作流程』。

相关概念

第 838 页的『安全设置』
运行 z/OSMF 所需的安全设置。

z/OS 自动供应 IBM MQ 对象

提供了样本以自动供应队列管理器和本地队列。

自动供应或取消供应 IBM MQ 队列管理器，并对供应的队列管理器执行操作

提供了以下特定于队列管理器的样本 z/OSMF 工作流程:

工作流程名称	描述
provision.xml	供应 IBM MQ for z/OS 队列管理器 此样本工作流程: • 供应队列管理器所需的系统资源。 • 为通道启动程序供应必需的系统资源。 • 启动队列管理器 (同时启动通道启动程序和 TCP/IP 侦听器) • 运行样本队列管理器安装验证程序。 可以设置环境属性以控制具有不同特征的队列管理器的供应。 有关更多信息，请参阅第 844 页的『运行工作流程』。 **注:** 提供了清单文件 (`provision.mf`) 以帮助为此工作流程添加模板。 此文件包含对包含其他信息的 **qaas_readme.pdf** 文件的引用。 添加模板后，您可以通过链接访问该文件。
deprovision.xml	取消供应 IBM MQ for z/OS 队列管理器 此样本工作流程: • 停止通道启动程序 (也停止 TCP/IP 侦听器) 和队列管理器。 • 等待子系统停止 • 取消供应所有通道启动程序和队列管理器系统资源。
startQMgr.xml	启动 IBM MQ for z/OS 队列管理器 此样本工作流程会启动队列管理器 (也会启动通道启动程序和 TCP/IP 侦听器)。
stopQMgr.xml	停止 IBM MQ for z/OS 队列管理器

工作流程名称	描述
	此样本工作流程会停止通道启动程序 (也会停止 TCP/IP 侦听器) 和队列管理器。

每个工作流程都执行一个或多个步骤。 工作流程中的注释说明了每个步骤所执行的功能。 某些步骤仅请求数据输入，而某些步骤提交 JCL ， 调用 REXX execs ， Shell 脚本或发出 REST API 调用以完成指定的函数。

请参阅每个步骤以获取 JCL 或 REXX exec 文件的确切名称。 工作流程和关联的 JCL 或 REXX 可执行文件引用在一个或多个变量 XML 文件中声明的变量。 有关更多详细信息，请参阅 第 844 页的『工作流程变量声明文件』。

可以对供应的 IBM MQ for z/OS 队列管理器执行 **deprovision**， **startQMgr** 和 **stopQMgr** 操作。

自动供应或取消供应 IBM MQ 本地队列，并对供应的队列执行操作

提供了以下特定于队列的样本 z/OSMF 工作流程:

工作流程名称	描述
defineQueue.xml	定义本地队列 此样本工作流程演示了如何使用 z/OSMF 工作流程根据属性设置来定义小型，中型或大型队列。 **注:** 提供了清单文件 (`provision.mf`) 以帮助为此工作流程添加模板。 此文件包含对包含其他信息的 **qaas_readme.pdf** 文件的引用。 添加模板后，您可以通过链接访问该文件。
displayQueue.xml	显示本地队列的所选属性 此样本工作流程显示本地队列的所选属性。 将在 z/OSMF 变量中返回这些属性 (请参阅工作流程中的步骤以获取变量的名称) ，然后显示这些属性。 如果需要，可以使用 REST API 来访问变量的内容。 有关更多详细信息，请参阅 云供应 REST API，另请参阅 z/OSMF 工作流程服务。
deleteQueue.xml	删除本地队列 此样本工作流程将删除指定队列管理器上的本地队列。
putQueue.xml	将一条或多条消息放入本地队列。 此样本工作流程将一条或多条消息放入本地队列。 可以指定消息文本，但如果同时将多条消息放入本地队列，那么将使用相同的消息文本。
getQueue.xml	从本地队列获取一条或多条消息。 此样本工作流程从本地队列获取一条或多条消息。 将在 z/OSMF 变量中返回消息 (请参阅工作流程中的步骤以获取变量的名称) ，然后显示这些消息。 如果需要，可以使用 REST API 访问变量的内容。 有关更多详细信息，请参阅 云供应 REST API，另请参阅 z/OSMF 工作流程服务。
loadQueue.xml	将消息从数据集装入到本地队列。 此样本工作流程将消息从数据集装入到本地队列。 通过设置属性来指定数据集的缺省名称。 有关更多详细信息，请参阅 第 844 页的『运行工作流程』。
offloadQueue.xml	将消息从本地队列卸载到数据集。

工作流程名称	描述
	此样本工作流程将消息从本地队列卸载到数据集。 通过设置属性来指定数据集的缺省名称。 有关更多详细信息，请参阅 第 844 页的『运行工作流程』。
clearQueue.xml	清除本地队列上的消息。 此样本工作流程清除 (删除) 本地队列上的所有消息。

注意:

1. **放入队列** 操作允许您输入一些消息数据并将一条或多条消息放入队列中。 如果要在给定请求期间将多条消息放置到队列上，那么将使用相同的消息数据。
2. loadQueue.xml 和 offloadQueue.xml 工作流程使用别名 QLOAD 调用 SCSQLOAD 库中的可执行模块 CSQUDMSG。 这等同于随 IBM MQ for Multiplatforms 提供的 **dmpmqmsg** 实用程序。 因此，从数据集装入到队列或从队列装入到数据集的消息应该采用 **dmpmqmsg** 格式。

   样本 JCL 也作为 SCSQPROC 中的成员 CSQ4QLOD 提供。

   尝试 loadQueue 和 offloadQueue 操作的最简单方法是执行以下操作:

   a. 发出 **putQueue** 几次以将某些消息放入队列。
   b. 使用 **offloadQueue** 将消息从队列卸载到数据集。
   c. 如果需要，请发出 **clearQueue** 以从队列中除去所有消息。
   d. 使用 **loadQueue** 将消息从数据集装入到同一队列或其他队列中。

   如果您对 **dmpmqmsg** 格式感兴趣，可以在发出卸载请求后浏览数据集的内容。
3. 您可以对供应的 IBM MQ for z/OS 本地队列执行 **displayQueue**， **deleteQueue**， **putQueue**， **getQueue**， **loadQueue**， **offloadQueue** 和 **clearQueue** 作为操作。 有关操作和操作文件的更多详细信息，请参阅 *z/OSMF Programming Guide*。
4. 缺省情况下，将删除所有与操作相关的工作流程。 这样做的原因是为了最大限度减少用户清除工作流程的需求。

   但是，这方面的问题在于，如果某个操作导致某些输出。 例如， **displayQueue** 和 **getQueue** 操作都将生成输出。

   无法看到输出，因为一旦执行了操作，就会删除相关工作流程。 因此，如果您从 z/OS WUI 驱动工作流程操作，那么需要在要查看其输出的每个操作的 **<workflow>** 标记上将 **cleanAfterComplete** 标志设置为 假 。

   例如，要查看 **displayQueue** 的输出，请按如下所示设置标志:

```
<action name="displayQueue">
  <workflow cleanAfterComplete="false">
  ...

  ...
  </workflow>
</action>
```

   但是，这意味着您必须手动清除与操作相关的工作流程。

每个样本 z/OSMF 工作流程执行一个或多个步骤。 工作流程中的注释说明了每个步骤所执行的功能。 某些步骤仅请求数据输入，而某些步骤提交 JCL ，而其他步骤调用 REXX execs 以完成所声明的功能。

请参阅每个步骤以获取 JCL 或 REXX exec 文件的确切名称。 工作流程和关联的 JCL 或 REXX 可执行文件引用在一个或多个 第 844 页的『工作流程变量声明文件』中声明的变量。

相关概念

第 840 页的『限制』
将 z/OSMF 与 IBM MQ 配合使用时的限制。

z/OS 运行工作流程

样本 " z/OSMF " 工作流程所引用的文件的描述以及如何运行工作流程。

工作流程变量声明文件

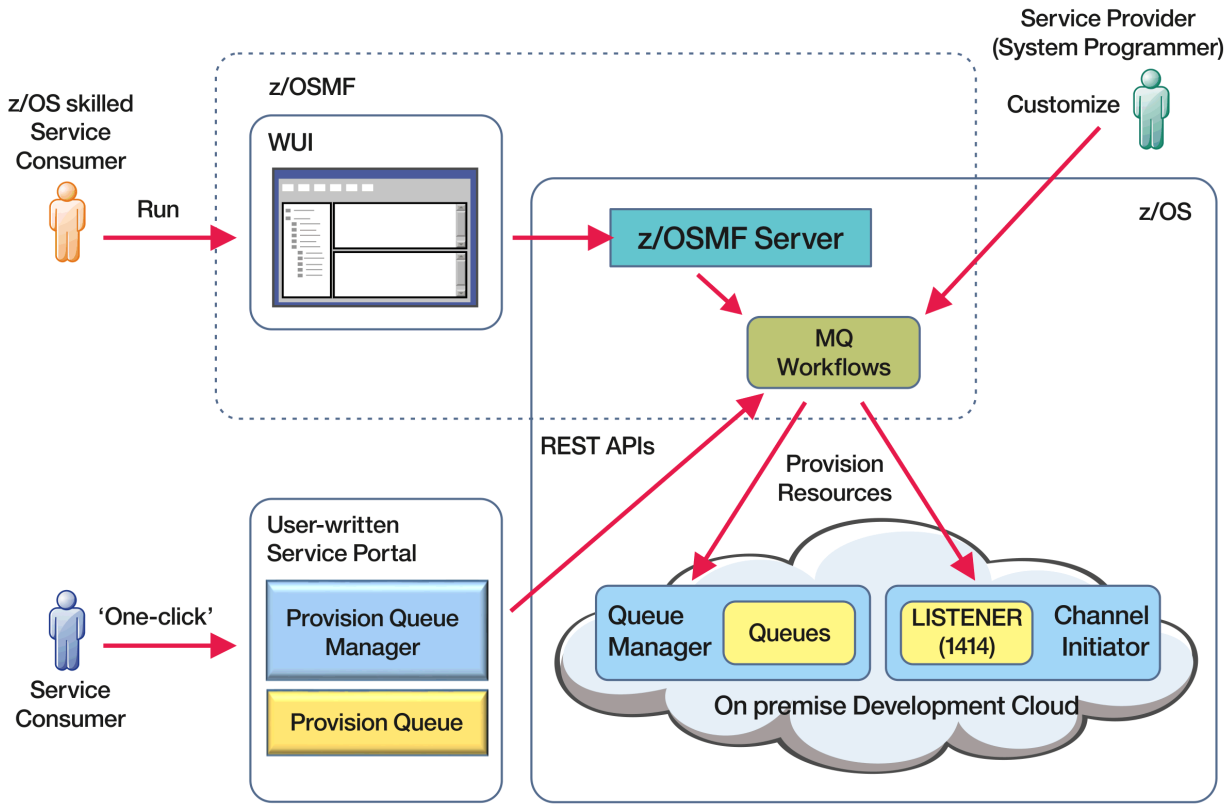
以下文件声明样本 z/OSMF 工作流程以及关联的 JCL 或 REXX 可执行文件所引用的变量:

工作流程变量声明文件名	描述
common_variables.xml	队列管理器 (外加通道启动程序) 和队列工作流程的公共变量。
qmgr_variables.xml	特定于队列管理器 (以及通道启动程序) 工作流程的变量。
queue_variables.xml	特定于队列工作流程的变量。
tcpip_variables.xml	特定于队列管理器 (以及通道启动程序) 工作流程的变量，用于标识 TCP/IP 资源。

注: 变量的缺省可视性为 *private*。 为了允许使用 z/OSMF REST API 查询变量，所选变量已标记为 *public*。 但是，如果需要，您可以更改给定变量的可视性。

运行工作流程

图 *127: "一键式" 供应 IBM MQ for z/OS 资源*



在可以运行工作流程之前，需要在以下文件中设置某些属性:

工作流程变量属性文件名	描述
workflow_variables.properties	工作流程变量的初始属性。 文件中的注释指示每个属性的用途。

工作流程变量属性文件名	描述
	• Properties within meta-brackets (< >) need to be set to user specific values. • 可以将环境属性设置为为开发 (DEV)，测试 (TEST)，质量保证 (QA) 或生产 (PROD) 环境供应队列管理器。 其他属性设置控制要为每个环境供应的队列管理器的特征。 例如，可以更改每种环境类型的活动日志数或页集数。 • 其他属性设置为 IBM MQ 缺省值，但可以进行修改以满足本地约定 (如果需要)。

通常，一旦设置了属性，就可以按现在的方式运行工作流程。 但是，如果需要，您可以定制工作流程以修改或除去现有步骤，或者添加新步骤。

可以运行工作流程:

- 从 z/OSMF WUI。

  从云供应-> WUI 中的软件服务，可以自动或手动方式运行工作流程。 手动方式在测试时很有用，在这两种方式下，都可以监视工作流程中每个步骤的进度。

  有关更多详细信息，请参阅 云供应服务 和 创建工作流程。
- 使用 z/OSMF REST Workflow Services。

  REST Workflow Services 可用于通过 REST API 运行工作流程。 此方式对于从用户编写的门户网站创建一键操作很有用。

  有关更多详细信息，请参阅 云供应 REST API，另请参阅 z/OSMF 工作流程服务。
- 使用 z/OSMF 随附的样本市场门户网站。

相关概念
第 841 页的『自动供应 IBM MQ 对象』
提供了样本以自动供应队列管理器和本地队列。

z/OS MQ Adv. VUE 配置 IBM MQ Advanced for z/OS VUE

使用此信息来配置随 IBM MQ Advanced for z/OS VUE 权利提供的功能部件。

关于此任务

您可以启用 Managed File Transfer 代理程序与 IBM MQ Advanced for z/OS Value Unit Edition 的远程连接。

您可以使用从 IBM MQ Advanced for z/OS Value Unit Edition 队列管理器到 IBM Cloud 中的 IBM Blockchain 服务的连接。

您可以使用客户机连接将 IBM MQ classes for JMS 或 IBM MQ classes for Java 应用程序连接到 z/OS 上具有 **ADVCAP(ENABLED)** 属性的队列管理器。 有关更多信息，请参阅 Java 和 JMS 客户机与 z/OS 队列管理器的连接。

过程

1. 启用与 IBM MQ Advanced for z/OS Value Unit Edition 的 Managed File Transfer 代理程序远程连接。
2. 配置 IBM MQ Advanced for z/OS VUE 以与 IBM Cloud 中的 IBM Blockchain 服务配合使用。

z/OS MQ Adv. VUE 启用与远程 z/OS 队列管理器的 MFT 代理程序连接

LTS 在某些情况下， z/OS 上的 Managed File Transfer 代理程序可以使用客户机连接来连接到 z/OS 上的远程队列管理器。 这可能导致更简单的 IBM MQ 拓扑。

LTS

在以下情况下，支持与远程 z/OS 队列管理器的客户机连接:

- MFT 代理程序位于 IBM MQ 9.2.0 长期支持，应用了 APAR PH56722 ，并且已与 IBM MQ Advanced for z/OS Value Unit Edition 或 IBM MQ Advanced for z/OS 的产品标识 (PID) 相关联
- MFT 代理程序位于 IBM MQ 9.2.0 ，并且已与 IBM MQ Advanced for z/OS Value Unit Edition 的 PID 相关联

请参阅 IBM MQ 产品标识和导出信息 ，以获取 IBM MQ 产品，其关联的 PID 值和导出分类的详细信息。

有关如何设置与 MFT 安装关联的 PID 的信息，请参阅 **fteSetProductID** 。

代理程序正在其下运行的 PID 在代理程序启动时显示在日志中。

注: 在任何其他 PID 下运行的 z/OS 上的 MFT 代理程序只能使用绑定方式连接来连接到本地队列管理器。

如果代理程序尝试连接到未在 z/OS 上运行的队列管理器，那么将发出消息 BFGQM1044E ，并结束代理程序启动。

相关任务

在 z/OS 上启动 MFT 代理

z/OS Deprecated MQAdv.VUE 配置 IBM MQ Advanced for z/OS VUE 以用于区块链

设置并运行 IBM MQ Bridge to blockchain 以安全地连接 z/OS 队列管理器上的 IBM MQ 和 IBM Blockchain。 通过使用连接到 IBM MQ Advanced for z/OS VUE 队列管理器的消息传递应用程序，使用网桥以异步方式连接到，查找和更新区块链中资源的状态。

开始之前

注意:

- Deprecated 从 2022 年 11 月 22 日起，在所有发行版中不推荐使用 IBM MQ Bridge to blockchain (请参阅 美国声明函 222-341)。
- V 9.2.0.21 Removed 对于 Long Term Support，将在 IBM MQ 9.2.0 CSU 21 中移除 IBM MQ Bridge to blockchain 。 如果您有将受此更改影响的应用程序，请联系 IBM 支持人员。
- IBM MQ Bridge to blockchain 作为 IBM MQ Advanced for z/OS Value Unit Edition 9.1.0 上连接器包的一部分提供。 您可以连接到在同一命令级别或更高级别上运行的 IBM MQ Advanced for z/OS VUE 队列管理器。
- 支持将 IBM MQ Bridge to blockchain 与基于 Hyperledger Fabric 构建的 Hyperledger Composer 的区块链网络配合使用。
- IBM MQ Bridge to blockchain 必须安装在 z/OS UNIX System Services 环境中，并且需要来自 IBM 的 Java runtime environment 版本 8。

关于此任务

Blockchain 是一个共享的分布式数字分类帐，由表示网络中同级之间的商定交易的区块链组成。 链中的每个块都链接到前一个块，以此类推，返回到第一个事务。

IBM Blockchain 是基于 Hyperledger Fabric 和 Hyperledger Composer 构建的。 您可以在本地使用 Docker 进行开发，也可以在 IBM Cloud 中的容器集群中进行开发。 您还可以在生产环境中激活和使用 IBM Blockchain 网络，以构建和管理具有高级别安全性，隐私性和性能的业务网络。 有关更多信息，请参阅 IBM Blockchain Platform。

Hyperledger Fabric 和 Hyperledger Composer 是由 Hyperledger Project 成员协作开发的开放式源代码企业区块链框架，包括作为初始代码添加程序的 IBM 。 Hyperledger Project 或 Hyperledger，是一个 Linux

Foundation 开源的全球协作计划，旨在推进跨行业区块链技术。 有关更多信息，请参阅 IBM Blockchain，Hyperledger 项目， Hyperledger Fabric 和 Hyperledger Composer。

如果您已在使用 IBM MQ Advanced for z/OS VUE 和 IBM Blockchain，那么可以使用 IBM MQ Bridge to blockchain 通过 Hyperledger Composer REST 接口来驱动 Hyperledger Composer 业务模型，从而允许您更新或查询区块链中的状态，并从区块链网络接收回复。 通过这种方式，您可以将内部部署 IBM 软件与云区块链服务或本地管理的内部部署解决方案集成。

可以在 图 1 中查看网桥操作过程的简要概述。 用户应用程序将 JSON 格式的消息放在 z/OS 队列管理器上的输入/请求队列上。 通过使用 Hyperledger Composer REST 服务器，网桥将连接到队列管理器，从输入/请求队列获取消息，检查 JSON 是否正确格式化，然后向区块链发出 REST 请求。 区块链返回的数据由网桥解析并放置在应答队列上，如原始 IBM MQ 请求消息中所定义。 用户应用程序可以连接到队列管理器，从应答队列获取响应消息，并使用该信息。

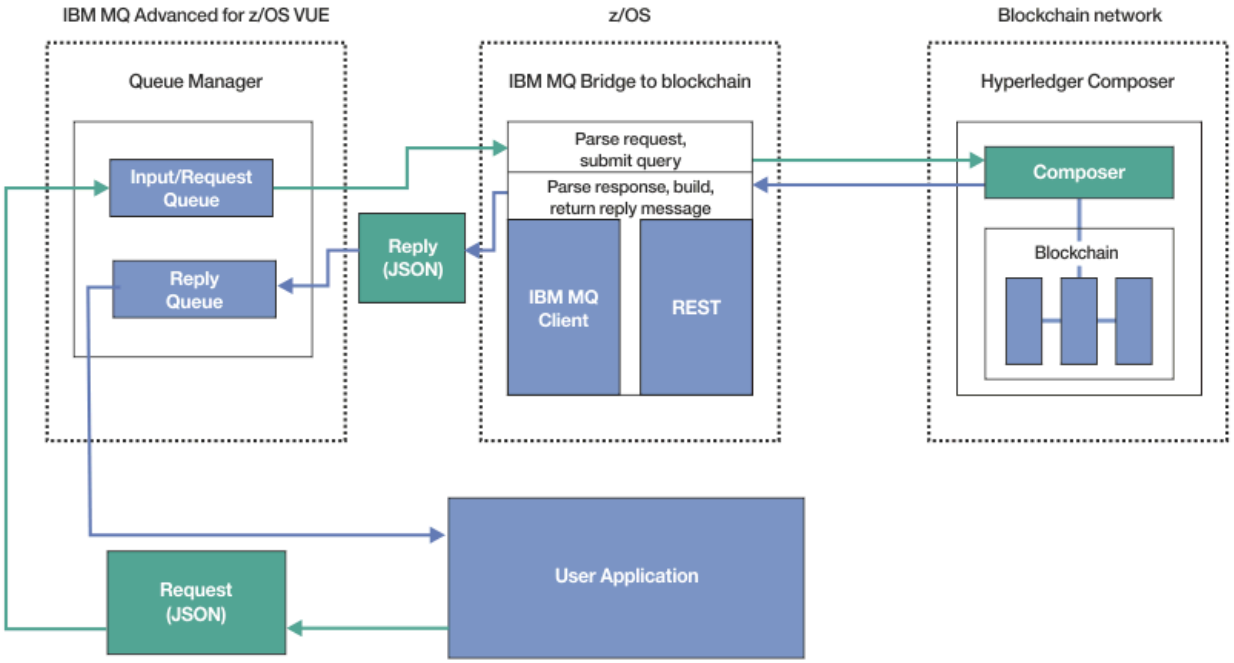


图 *128: IBM MQ Bridge to blockchain*

必须配置 IBM MQ Bridge to blockchain 以连接到 Hyperledger Composer REST 服务器，而不是直接连接到底层 Hyperledger Fabric 层。 当网桥正在运行时，消息传递应用程序请求网桥根据用户定义的业务网络模型来驱动 Hyperledger Composer REST API，进而驱动可查询或更新资源状态的底层链代码例程，并使用 Hyperledger Composer REST 服务器将结果作为响应返回到消息传递应用程序。

过程

通过在 `thlqual.SCSQPROC(CSQ4BCBQ)`中定制并提交样本 JCL，为网桥创建队列。
为用于下列各项的缺省指定队列提供了样本网桥队列定义:

- 网桥的消息输入: `SYSTEM.BLOCKCHAIN.INPUT.QUEUE` 和 `APPL1.BLOCKCHAIN.INPUT.QUEUE`
- 来自区块链的回复: `APPL1.BLOCKCHAIN.REPLY.QUEUE`

不同的应用程序可以使用相同的输入队列，但您可以指定多个应答队列，每个应用程序一个应答队列。 您不必使用定义的应答队列。 如果要将动态队列用于应答，那么必须考虑其安全性配置。

结果

您已创建网桥处理来自 IBM MQ 和区块链网络的消息所需的队列。

下一步做什么

使用队列管理器的信息以及来自区块链网络的凭证，为 IBM MQ Bridge to blockchain 创建配置文件。

z/OS Deprecated MQ Adv. VUE 为 IBM MQ Bridge to blockchain 创建配置文件

输入队列管理器和区块链网络参数，为 IBM MQ Bridge to blockchain 创建配置文件以连接到 IBM MQ 和 IBM Blockchain 网络。

开始之前

注意:

- Deprecated 从 2022 年 11 月 22 日起，在所有发行版中不推荐使用 IBM MQ Bridge to blockchain (请参阅 美国声明函 222-341)。
- V 9.2.0.21 Removed 对于 Long Term Support，将在 IBM MQ 9.2.0 CSU 21 中移除 IBM MQ Bridge to blockchain 。 如果您有将受此更改影响的应用程序，请联系 IBM 支持人员。
- 您已创建并配置 Hyperledger Composer 区块链网络。
- 您已在 z/OS 环境中安装 IBM MQ Bridge to blockchain 。
- 您已启动 IBM MQ Advanced for z/OS VUE 队列管理器。

关于此任务

此任务将引导您完成创建 IBM MQ Bridge to blockchain 配置文件并成功连接到 IBM Blockchain 和 IBM MQ 网络所需的最低设置。

您可以使用网桥连接到基于 Hyperledger Composer 的区块链网络。 要使用网桥，您需要来自区块链网络的配置信息。 在此任务中的每个步骤中，您可以找到基于两个不同配置的区块链网络的示例配置详细信息:

- 在 Docker 中运行的 Hyperledger Composer 网络。 有关更多信息，请参阅 安装 Hyperledger Composer 和 生成 REST API。
- 在 IBM Cloud 中的 Kubernetes 集群中运行的 Hyperledger Composer 网络。 有关详细信息，请参阅在 IBM Blockchain Platform 的云沙箱中进行开发。

有关所有 IBM MQ Bridge to blockchain 参数的含义和选项的更多信息，请参阅 runmqbcb 命令。 您必须考虑自己的安全需求，并定制适合于您的部署的参数。

过程

1. 在 z/OS UNIX System Services (z/OS UNIX) 环境中运行网桥以创建配置文件。

   您需要 Hyperledger Composer 安全性信息和 IBM MQ Advanced for z/OS VUE 队列管理器中的参数。

   从安装了 IBM MQ 的 z/OS UNIX 中的位置的 `mqbc/bin` 目录运行网桥脚本。

   ```
   ./runmqbcb -o config_file_name.cfg
   ```

   如以下示例所示，现有值显示在方括号内。 按 `Enter` 以接受现有值，按 `Space` 然后 `Enter` 以清除值，然后在方括号内输入，然后按 `Enter` 以添加新值。 您可以使用逗号或在新行上输入每个值来分隔值列表(例如同级)。 空白行结束列表。

   注: 无法编辑现有值。 您可以保留，替换或清除这些信息。
2. 输入与 IBM MQ Advanced for z/OS VUE 队列管理器的连接的值。

   连接所需的最小值是您定义的队列管理器名称和网桥输入队列的名称。 对于与远程 IBM MQ Advanced for z/OS VUE 队列管理器的连接，您还需要 **MQ Channel** 和 **MQ Conname** (运行队列管理器的主机地址和端口)。

   要使用 TLS 在步骤 第 849 页的『5』 中连接到 IBM MQ ，必须使用 JNDI 或 CCDT 并相应地指定 **MQ CCDT URL** 或 **JNDI implementation class** 和 **JNDI provider URL** 。

   注: MQ CCDT 或 **JNDI** 值优先于值重叠的配置文件。

```
Connection to Queue Manager
---------------------------
Queue Manager                       : [z/OS_ADV_VUE_qmgr_name]
Bridge Input Queue                  : [APPL1.BLOCKCHAIN.INPUT.QUEUE]
MQ Channel                          : []
MQ Conname                          : []
MQ CCDT URL                         : []
JNDI implementation class           : []
JNDI provider URL                   : []
MQ Userid                           : []
MQ Password                         : []
```

3. 输入与区块链网络关联的 Hyperledger Composer REST 服务器的凭证 (如果已配置)。

   在以下示例中， Hyperledger Composer REST 服务器已使用 **passport-ldapauth NodeJS** 模块配置了 LDAP 凭证库。 请注意，您可以使用以这种方式提供基本用户和密码样式凭证的任何 **passport-*** 模块。 有关更多信息，请参阅 对 REST 服务器启用认证。

```
User Identification
--------------------------------
Userid                            : []admin
Password                          : []******
API path for Login                : auth/ldap
```

4. 输入 Hyperledger Composer REST 服务器的地址。

   请注意，此属性中不需要任何协议 (即 `http` 或 `https`) ，并且端口号是必需的。 是否使用 HTTP 或 HTTPS 协议取决于 REST 服务器的安全配置。 如果向 REST 服务器提供了证书和专用密钥对，那么将使用 HTTPS。 使用 HTTPS。 否则，将使用 HTTP。 有关如何指定证书和专用密钥对的信息，请参阅步骤 第 849 页的『5』。

```
REST Server
---------------------------
Address for Composer REST server     : [composer-rest-server-ip-address:3000]
```

5. 输入 TLS 连接的证书库值。

   网桥充当正在连接到队列管理器的 IBM MQ JMS 客户机，这意味着可以将其配置为使用 TLS 安全性来以与任何其他 IBM MQ JMS 客户机相同的方式进行安全连接。 仅当您在步骤 第 848 页的『2』 中指定 JNDI 或 CCDT 信息后，才会公开 TLS 连接详细信息的配置。

   证书库用于 Hyperledger Composer 以及 IBM MQ Advanced for z/OS VUE 队列管理器。 如果指定了证书库，那么网桥始终尝试使用 HTTPS 连接到 Hyperledger REST 服务器。 但是，可以对 IBM MQ 连接禁用 TLS ，但仍使用以下选项将 TLS 用于 Hyperledger Composer 。

```
Certificate stores for TLS connections
--------------------------------------
Personal keystore                   : []
Keystore password                   : []
Trusted store for signer certs      : []
Trusted store password              : []
Use TLS for MQ connection           : [N]
Timeout for Blockchain operations   : [12]
```

   有关更多信息，请参阅 使用 HTTPS 和 TLS 保护 REST 服务器。

6. 可选： 输入 IBM MQ Bridge to blockchain 的日志文件的位置。

   可以在配置文件或命令行中指定日志文件名和位置。

```
Behavior of bridge program
--------------------------
Runtime logfile for copy of stdout/stderr : [/var/mqm/errors/runmqbcb.log]
Done.
```

结果

您已创建 IBM MQ Bridge to blockchain 用于连接到 IBM Blockchain 网络和 IBM MQ Advanced for z/OS VUE 队列管理器的配置文件。

下一步做什么

完成 第 850 页的『运行 IBM MQ Bridge to blockchain』 的步骤

相关参考

runmqbcb (运行 IBM MQ Bridge to blockchain)

z/OS Deprecated IBM MQ Bridge to blockchain 的 IBM MQ 安全性配置

使用 IBM MQ Bridge to blockchain 设置 IBM MQ 安全性的注意事项。

注意:

- Deprecated 从 2022 年 11 月 22 日起，在所有发行版中不推荐使用 IBM MQ Bridge to blockchain (请参阅 美国声明函 222-341)。
- V 9.2.0.21 Removed 对于 Long Term Support，将在 IBM MQ 9.2.0 CSU 21 中移除 IBM MQ Bridge to blockchain 。 如果您有将受此更改影响的应用程序，请联系 IBM 支持人员。

以下示例显示可用于授予 IBM MQ Bridge to blockchain 对其需要的队列的访问权的 RACF 定义。 这些定义假定网桥正在 MQBCBUSR 用户标识下运行。

此外，需要授予 IBM MQ Bridge to blockchain 访问权以连接到队列管理器，或者:

- 直接使用绑定方式; 请参阅 批处理连接的连接安全概要文件，或者
- 通过 CHINIT 使用客户机方式; 请参阅 客户机 MQI 请求

IBM MQ Bridge to blockchain 请求队列的授权

发出以下 RACF 命令以授予 MQBCBUSR 用户标识访问权，从而从缺省 SYSTEM.BLOCKCHAIN.INPUT.QUEUE 请求队列:

```
RDEFINE MQQUEUE SYSTEM.BLOCKCHAIN.INPUT.QUEUE UACC(NONE)
PERMIT SYSTEM.BLOCKCHAIN.INPUT.QUEUE CLASS(MQQUEUE) ID(MQBCBUSR) ACCESS(UPDATE)
```

IBM MQ Bridge to blockchain 响应队列的授权

发出以下 RACF 命令以授予 MQBCBUSR 用户标识访问权，从而将消息发送到 APPL1.BLOCKCHAIN.REPLY.QUEUE。 此队列名称是在对请求消息的队列名称的应答上指定的:

```
RDEFINE MQQUEUE APPL1.BLOCKCHAIN.REPLY.QUEUE UACC(NONE)
PERMIT APPL1.BLOCKCHAIN.REPLY.QUEUE CLASS(MQQUEUE) ID(MQBCBUSR) ACCESS(UPDATE)
PERMIT CONTEXT.APPL1.BLOCKCHAIN.REPLY.QUEUE CLASS(MQADMIN) ID(MQBCBUSR) ACCESS(UPDATE)
```

相关任务

第 727 页的『运行 IBM MQ Bridge to blockchain 客户机样本』

您可以使用随 IBM MQ Bridge to blockchain 提供的 JMS 客户机样本，将消息放入输入队列中，以供区块链网桥检查并查看接收到的应答。 此样本基于使用 IBM MQ Bridge to blockchain 与 Hyperledger Composer Trader 网络示例集成。

相关参考

队列安全性的概要文件

API-资源安全访问快速参考

z/OS Deprecated MQ Adv. VUE 运行 IBM MQ Bridge to blockchain

运行 IBM MQ Bridge to blockchain 以连接到 IBM Blockchain 和 IBM MQ。 连接后，网桥已准备好处理请求消息，将其发送到 Hyperledger Composer 区块链网络，并接收和处理回复。

开始之前

注意:

- Deprecated 从 2022 年 11 月 22 日起，在所有发行版中不推荐使用 IBM MQ Bridge to blockchain (请参阅 美国声明函 222-341)。
- V 9.2.0.21 Removed 对于 Long Term Support，将在 IBM MQ 9.2.0 CSU 21 中移除 IBM MQ Bridge to blockchain 。 如果您有将受此更改影响的应用程序，请联系 IBM 支持人员。

关于此任务

使用在前一任务中创建的配置文件来运行 IBM MQ Bridge to blockchain。

过程

1. 启动要用于网桥的 IBM MQ Advanced for z/OS VUE 队列管理器。
2. 启动 IBM MQ Bridge to blockchain 以连接到区块链网络和 IBM MQ Advanced for z/OS VUE 队列管理器。

   请完成下面任意一项任务:

   a) 从安装了 IBM MQ 的 z/OS UNIX 位置中的 `mqbc/bin` 目录直接在 z/OS UNIX System Services (z/OS UNIX) 中运行网桥。

   ```
   ./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
   ```

   或

   b) b. 使用 `thlqual.SCSQPROC(CSQ4BCB)` 中提供的样本 JCL 在 z/OS 系统上运行网桥。

   您必须对 JCL 进行一些特定于您的环境的更新:

   - 将 ++THLQUAL++ 替换为 IBM MQ 目标库数据集的高级限定符。
   - 将 ++LANGLETTER++ 替换为要显示消息的语言的字母。
   - 将 ++PATHPREFIX++ 替换为 z/OS UNIX 组件安装路径。
   - 将 ++CONFIGFILE++ 替换为使用 z/OS UNIX 中的 `runmqbc -o <file>` 命令创建的配置文件的路径。
   - 将 ++JAVAHOME++ 替换为在 Java 8 或更高版本上运行的 64 位 Java 虚拟机 (JVM) 的位置。

   连接网桥时，将返回类似于以下内容的输出:

   ```
   2018-05-17 14:28:16.866 BST IBM MQ Bridge to Blockchain
   5724-H72 (C) Copyright IBM Corp. 2017, 2025.

   2018-05-17 14:28:19.331 BST Ready to process input messages.
   ```

3. 可选： 对 IBM MQ Advanced for z/OS VUE 队列管理器和区块链网络的连接进行故障诊断，前提是运行网桥后返回的消息指示连接不成功。

   a) 在调试方式下使用调试选项 1 发出该命令。

   ```
   ./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/
   logFile.log -d 1
   ```

   网桥将逐步完成连接设置并以简洁方式显示处理消息。

   b) 在调试方式下使用调试选项 2 发出该命令。

   ```
   ./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/
   logFile.log -d 2
   ```

   网桥将逐步完成连接设置并以详细方式显示处理消息。 完整输出将写入日志文件。

   请注意， (可选) 您还可以通过将 "`-d 0`" 更改为 "`-d 1`" 或 "`-d 2`" 来指定 JCL 中的调试方式选项。

结果

您已启动 IBM MQ Bridge to blockchain 并连接到队列管理器和区块链网络。

下一步做什么

- 请遵循 第 727 页的『运行 IBM MQ Bridge to blockchain 客户机样本』 中的步骤进行格式化，并向区块链网络发送查询或更新消息。
- 使用 *MQBCB_EXTRA_JAVA_OPTIONS* 变量传入 JVM 属性，例如启用 IBM MQ 跟踪。 有关更多信息，请参阅 跟踪 IBM MQ Bridge to blockchain。

z/OS Deprecated *IBM MQ 9.2.0 之前的 IBM MQ Bridge to blockchain 的消息格式*

有关 IBM MQ Bridge to blockchain 发送和接收的消息格式的信息。

V 9.2.0

注意: 消息格式的现有格式已过时。 从 IBM MQ 9.2.0 开始，如果您有 Hyperledger Fabric 网络，请使用 第 726 页的『来自 IBM MQ 9.2.0 的 IBM MQ Bridge to blockchain 的消息格式』 中描述的消息格式。

应用程序请求 IBM MQ Bridge to blockchain 驱动 Hyperledger Composer 定义的 REST API 对区块链上保存的信息执行操作。 应用程序通过在网桥请求队列上放置请求消息来执行此操作。 REST 请求的结果由网桥格式化为应答消息。 网桥使用请求消息的 MQMD 中的 **ReplyToQ** 和 **ReplyToQMgr** 字段中包含的信息作为应答消息的目标。

请求和应答消息是 JSON 格式的文本 (MQSTR) 消息。

请求消息格式

请求消息包含三个属性:

方法 (method)
用于调用 Hyperledger Composer REST API (例如 POST， DELETE 或 GET) 的 REST 动词

path
Hyperledger Composer REST API 的路径。 这将添加到基本服务器 URL。 路径应以 "api/" 开头。

主体
特定于方法的内容。 这通常是 JSON 结构。

以下示例使用 POST 方法来路径 api/Trader，以创建新的 Trader 对象。 主体指定由用户的 Hyperledger Composer 模型定义的贸易商类，并且还指定在区块链网络中创建新的 Trader 对象所需的其他值。

```
{ "method": "POST",
  "path": "api/Trader",
  "body": {
    "$class" : "org.example.trading",
    "tradeId" : "Trader2",
    "firstName": "Jane",
    "lastName" : "Doe"
```

应答消息格式

应答消息将其相关标识设置为入站消息的消息标识。 任何用户定义的属性都将从请求消息复制到应答消息。 应答中的用户标识设置为发起方的用户标识。

statusCode 是 HTTP 状态码。 如果错误来自 IBM MQ 或网桥，那么将使用相应的 **statusCode** 。

statusType 是字符串 成功 或 失败。

对于成功的请求，应答消息中的 "**data**" 元素包含来自调用的 Hyperledger Composer REST API 的响应。

成功处理的示例:

```
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "data": [
      {
         "$class": "org.example.trading",
         "firstName": "John",
```

```
         "lastName": "Doe",
         "tradeId": "Trader1"
      },
      {
         "$class": "org.example.trading",
         "firstName": "Jane",
         "lastName": "Doe",
         "tradeId": "Trader2"
      }
   ]
}
```

所有错误响应都具有相同的字段，无论这些字段是由网桥本身生成，从对 Hyperledger Composer REST 服务器的调用，区块链还是从链代码调用生成。 例如：

- JSON 输入消息错误

```
{
   "statusCode": 400,
   "statusType": "FAILURE",
   "message": "[AMQBC021E] Error: Cannot parse input message or there are
    missing fields in the message. Missing fields appear to be: "method".”
}
```

- 未能由 Hyperledger Composer REST 服务器处理的请求

```
{
   "statusCode": 500,
   "statusType": "FAILURE",
   "message": "Error trying to invoke business network. Error: No valid responses
    from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
    error (status: 500, message: Error: Failed to add object with ID 'Trader1'
    as the object already exists)"
}
```

应用程序可以通过查看 **statusType** 字符串或数据字段的存在来判断请求是成功还是失败。 当处理输入消息时发生错误，并且网桥未将其发送到区块链时，从网桥返回的值为 MQRC 值，通常为 **MQRC_FORMAT_ERROR**。

z/OS Deprecated 运行 *IBM MQ Bridge to blockchain* 客户机样本

您可以使用随 IBM MQ Bridge to blockchain 提供的 JMS 客户机样本，将消息放入输入队列中，以供区块链网桥检查并查看接收到的应答。 此样本基于使用 IBM MQ Bridge to blockchain 与 Hyperledger Composer Trader 网络示例集成。

开始之前

请参阅 /trade_network 以获取更多信息

IBM MQ Bridge to blockchain 正在运行，并且已连接到 IBM MQ Advanced 或 IBM MQ Advanced for z/OS VUE 队列管理器以及区块链网络。

关于此任务

在 IBM MQ Bridge to blockchain 的 samp 目录中查找 JMS 样本应用程序 (ComposerBCBSamp.java)。

例如: <MQ_INSTALL_ROOT>/mqbc/samp/ComposerBCBSamp.java，其中 <MQ_INSTALL_ROOT> 是:

- Linux IBM MQ 的安装目录
- z/OS 安装 IBM MQ 的 z/OS UNIX 组件的 z/OS UNIX System Services 目录

.

过程

1. 编辑客户机样本 Java 源文件。

遵循样本中的指示信息以将其配置为与 IBM MQ 环境和区块链网络相匹配。

样本中的以下代码定义了要发送到网桥的三条 JSON 请求消息:

a. 首先，要除去现有 "commodity"

b. 其次，要创建新的 "commodity"， "owner" 和关联值,

c. 最后，在前两条请求消息之后显示有关 "commodity" 的新信息

```
private static JSONObject[] createMessageBodies() {
    JSONObject[] msgs = new JSONObject[3]; // This method creates 3 messages
    JSONObject m, m2;
    String commodityName = "BC";

    // Clean out the commodity in case it's already there. If
    // it's not there, there will be an error returned from Composer.
    m = new JSONObject();
    m.put("method", "DELETE");
    m.put("path", "api/Commodity/" + commodityName);
    msgs[0] = m;

    // To add the item to the table, the
    // operation looks like this:
    //
    // { "method": "POST",
    //   "path": "api/Commodity",
    //   "body" : {
    //     "$class": "org.example.trading.Commodity",
    //     "tradingSymbol" : "BC",
    //     "description" : "BC",
    //     "mainExchange" : "HERE",
    //     "owner" : "Me",
    //     "quantity" : 100
    //   }
    // }
    // You can see this structure in the API Explorer
    m = new JSONObject();
    m.put("method", "POST");
    m.put("path", "api/Commodity");
    m2 = new JSONObject();
    m2.put("$class", " org.example.trading.Commodity");
    m2.put("tradingSymbol", commodityName);
    m2.put("description", "Blockchain Sample Description");
    m2.put("mainExchange", "My Exchange");
    m2.put("owner", "Me");
    m2.put("quantity", 100);
    m.put("body", m2);
    msgs[1] = m;

    // And list all items that have been created
    m = new JSONObject();
    m.put("method", "GET");
    m.put("path", "api/Commodity");
    msgs[2] = m;

    return msgs;
  }
```

2. 编译样本。

   指向网桥目录中提供的 IBM MQ 客户机类和 JSON4J.jar 文件。

```
javac -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar
ComposerBCBClient.java
```

3. 运行已编译的类。

```
java -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar:.
ComposerBCBClient
```

```
Starting Simple MQ Blockchain Bridge Client
Starting the connection.
Sent message:
  {"method":"DELETE"," path ":"api\/Commodity\/BC"}
Response text:
```

```
{
   "statusCode": 204,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": ""
}
SUCCESS
Sent message:
{"body":
{"$class":"org.example.trading.Commodity","owner":"Me","quantity":100,"description":"Blockcha
in Sample Description","mainExchange":"My
Exchange","tradingSymbol":"BC"},"operation":"POST","url":"Commodity"}
Response text:
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": {
      "$class": "org.example.trading.Commodity",
      "description": "Blockchain Sample Description",
      "mainExchange": "My Exchange",
      "owner": "Me",
      "quantity": 100,
      "tradingSymbol": "BC"
   }
}
SUCCESS
Sent message:
  {"method":"GET","path":"api\/Commodity"}
Response text:
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": [
      {
         "$class": "org.example.trading.Commodity",
         "description": "Blockchain Sample Description",
         "mainExchange": "My Exchange",
         "owner": "resource:org.example.trading.Trader#Me",
         "quantity": 100,
         "tradingSymbol": "BC"
      }
   ]
}
SUCCESS
```

message 字段包含成功处理的消息的 "OK"，或者在请求失败的情况下，包含有关失败原因的信息。

如果客户机接收到等待响应的超时错误，请检查网桥是否正在运行。

配置 IBM MQ Internet Pass-Thru

本部分描述了 IBM MQ Internet Pass-Thru (MQIPT) 支持的各种功能以及如何配置这些功能。

通过对配置文件 `mqipt.conf` 进行更改来配置 MQIPT。 MQIPT 配置文件的结构以及可以指定的属性在 IBM MQ Internet Pass-Thru 配置参考中进行了描述。

注: 您应该在 `mqipt.conf` 文件所在的目录上设置安全文件许可权，以防止未经授权的用户看到任何存储的密码或更改配置。 遵循 第 889 页的『对 MQIPT 中存储的密码进行加密』中的过程来保护配置文件中指定的所有密码。

对配置文件的更改将在 MQIPT 启动或刷新时生效。 刷新 MQIPT 的活动实例将使配置更改生效，而无需重新启动 MQIPT。 刷新 MQIPT 时，将重新读取 `mqipt.conf` 配置文件， MQIPT 将执行以下操作:

- 任何标记为不活动或在配置文件中不再指定的活动路由都将关闭，并且不再接受入局连接。
- 将启动在配置文件中标记为活动且当前未在运行的任何路由。
- 将应用对活动路由的配置参数所作的任何更改。 如果可能，这些更改将在不中断活动连接的情况下生效。对于某些参数更改 (例如对路由目标的更改)，将在应用更改并重新启动路由之前关闭所有连接。

要刷新 MQIPT，请使用 **mqiptAdmin** 命令。 有关使用 **mqiptAdmin** 命令管理 MQIPT 的更多信息，请参阅使用命令行管理 MQIPT。

MQIPT 中的 HTTP 支持

MQIPT 支持 HTTP 隧道传送。 可以配置 MQIPT 以使它转发的数据包编码为 HTTP 请求。

IBM MQ 通道不接受 HTTP 请求。 因此，需要第二个 MQIPT 来接收 HTTP 请求并将其转换回 IBM MQ 协议包。 第二个 MQIPT 除去 HTTP 头以将入局包转换回标准 IBM MQ 协议包，然后再将其传递到目标队列管理器。

在两个 MQIPT 实例之间使用 HTTP 时，HTTP 请求和应答流动所在的 TCP/IP 连接是持久的，在消息通道生命周期内将保持打开状态。 MQIPT 不关闭请求/应答对之间的 TCP/IP 连接。

如果两个 MQIPT 实例通过 HTTP 通信，那么 HTTP 请求可能较长时间内未处理。 例如，在请求方/服务器通道中，服务器端等待新消息到达其传输队列。 IBM MQ 通道协议提供脉动信号机制，这需要定期发出“等待结束”信号，以便将脉动信号消息发送到其合作伙伴。 缺省通道脉动信号周期为 5 分钟。 MQIPT 使用此脉动信号作为 HTTP 应答。 不要禁用此通道脉动信号，或者将其设为过高的值，以避免某些防火墙中引起超时问题。

MQIPT 以分块格式接受 HTTP 代理或服务器生成的 HTTP 流量。

有关在 MQIPT 中使用 HTTP 的示例，请参阅 配置 HTTP 隧道。

HTTP 代理

可以在 MQIPT 的两个实例之间放置 HTTP 代理。 HTTP 代理必须满足以下需求:

- 代理必须支持 HTTP 1.1 协议。
- 代理必须采用 MQIPT 设置的 **Connection** 或 **Proxy-Connection** HTTP 头。 这允许在消息通道的生存期内保持 MQIPT 的两个实例之间的连接处于打开状态。
- 必须在整个代理中维护持续连接的一对一映射。 这将确保不使用从代理到目标 MQIPT 的 TCP/IP 连接来传输多个消息通道的数据。

您可以设置属性以配置如何在某些 HTTP 代理上管理持久连接。 例如，您可能能够设置可以在持久连接上发出的最大请求数。 应设置以下属性:

- 应启用持久连接。
- 应该禁用多个 HTTP 会话复用从代理到 MQIPT 的 TCP/IP 连接，以维护整个代理中持久连接的一对一映射。
- 应该将代理请求的超时设置为高值。 例如，12 小时。
- 应该将持久连接上可以发出的最大请求数设置为高值。 例如， 5000。

MQIPT 使用 HTTP POST 请求在 MQIPT 的两个实例之间发送数据。 如果 MQIPT 配置使用 **HTTPProxy** 属性指定代理的主机名，那么 MQIPT 将连接到代理并使用 HTTP CONNECT 方法来请求代理建立到目标 MQIPT 的隧道。 这允许 HTTPS 连接通过代理，而不终止代理中的 TLS 会话。

如果在 MQIPT 实例之间放置了负载均衡器，那么必须将其配置为使用 *MQIPTSessionId* HTTP cookie 的值，以确保将每个会话的所有请求转发到同一目标。

MQIPT 中的 HTTPS

通过在发出客户机连接的 MQIPT 上启用 **HTTPS** 和 **SSLClient** 路由属性，可以在 HTTP 连接上使用 HTTPS。

MQIPT 必须具有用于认证目标 HTTP 代理/服务器的信任 CA 证书的访问权。 **SSLClientCAKeyring** 属性可用于定义包含可信 CA 证书的密钥环文件。

HTTPS 的通用设置将使用本地 HTTP 代理通过隧道越过防火墙，连接到远程 HTTP 服务器（或其他代理），远程 HTTP 服务器（或其他代理）又连接到远程 MQIPT。 连接服务器端的这个 MQIPT 不需要任何特定配置，因为连接请求被视为正常的 HTTP 连接。

MQIPT 使用 **HTTPProxy** 和 **HTTPServer** 属性来区分本地和远程代理。 设置了 **HTTPProxy** 属性的 MQIPT 路由被视为本地 HTTP 代理，设置了 **HTTPServer** 属性的 MQIPT 路由是远程服务器 (或代理)。

HTTPS 连接通常与 HTTP 代理/服务器上的侦听器端口地址 443 建立，但可以使用 **HTTPProxyPort** 和 **HTTPServerPort** 属性来覆盖此缺省值。

MQIPT 中的 SOCKS 支持

SOCKS 代理是一种网络服务，用作通过防火墙的退出控制点。 在防火墙内运行且已启用 SOCKS 的应用程序可以使用 SOCKS 代理连接到远程应用程序。

MQIPT 可以通过启用 **SocksServer** 属性来充当 SOCKS 代理，从而允许支持 SOCKS 的 IBM MQ 应用程序通过 MQIPT 连接到远程 IBM MQ 队列管理器。 使用此功能时，将在 SOCKS 握手过程中获取目标目标和目标端口地址，因此将覆盖 **Destination** 和 **DestinationPort** 路由属性。 这是支持 IBM MQ 集群的关键功能。

MQIPT 也可以用作 SOCKS 客户机，以代表尚未启用 SOCKS 的本地 IBM MQ 应用程序。 在使用仅允许通过 SOCKS 代理进行出站连接的防火墙时，该功能非常有用。 各个 MQIPT 路径可以配置为与不同的 SOCKS 代理通信。

有关如何使用 SOCKS 的示例，请参阅 配置 SOCKS 代理 。

MQIPT 中的集群

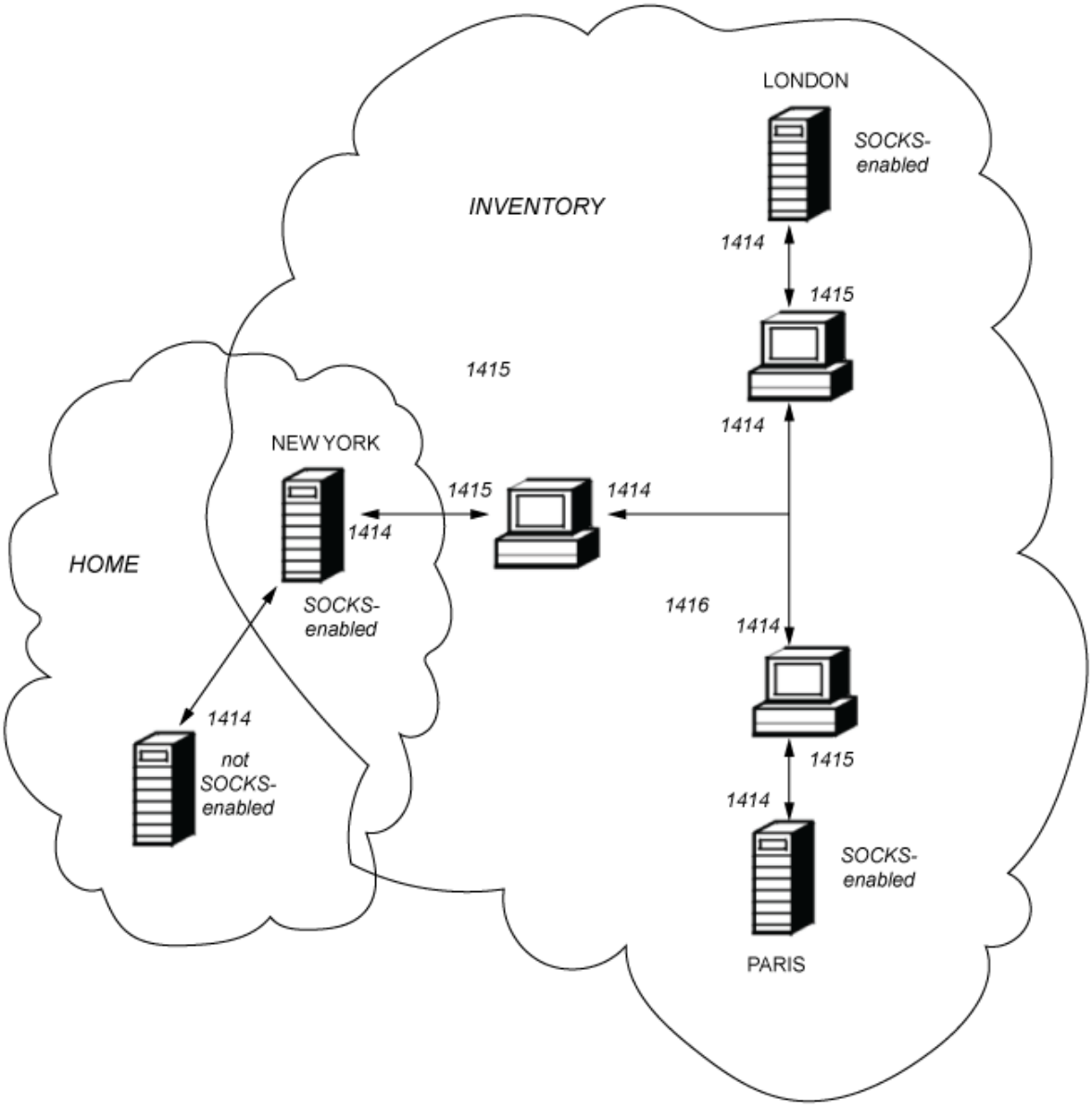
通过 SOCKS-启用跨因特网的集群中的每个队列管理器，以及通过启用 MQIPT 作为 SOCKS 代理，可以将 IBM MQ 集群与 MQIPT 配合使用。

在下图中，NEWYORK 和 CHICAGO 位于名为 HOME 的集群中，并且都拥有完整存储库。 NEWYORK、LONDON 和 PARIS 位于另一个集群（称为 INVENTORY）中。 请注意，CHICAGO 不需要启用 SOCKS，因为它所在的集群不需要使用 MQIPT。

INVENTORY 集群中的每个队列管理器都有效地“隐藏”在 MQIPT 后面。 由于队列管理器已经启用 SOCKS，所以在启动集群发送方通道时，将使用 MQIPT 作为 SOCKS 代理来将请求向外发送到其目的地。 通常，集群接收方通道中的 CONNAME 用于识别本地队列管理器，但与 MQIPT 配合使用时，CONNAME 必须识别本地 MQIPT 及其传入侦听器端口。 在下图中，所有传入侦听器端口都是 1414，传出侦听器端口地址都是 1415。

可使用两种方式来运行已启用 SOCKS 的队列管理器。 第一种方式是在运行队列管理器的整个计算机中启用 SOCKS。 第二种方式是仅在队列管理器中启用 SOCKS。 无论使用哪种方法，都必须配置 SOCKS 客户机，使其仅通过将 MQIPT 用作 SOCKS 代理来建立远程连接并禁用用户认证。 市场上有大量产品可用于提供 SOCKS 支持。 必须选择支持 SOCKS V5 协议的产品。

有关如何配置集群网络的示例，请参阅 配置 MQIPT 集群支持 。



MQIPT 中的 SSL/TLS 支持

可以使用安全套接字确保通信隐私、通信完整性和认证。

通信隐私
连接可以成为专用。 客户机和服务器之间交换的数据可以加密，只有发送方和接收方可以获得有用数据。 这意味着可以安全传输私人信息，例如，信用卡号码。

通信完整性
连接是可靠的。 消息传输包含基于安全散列函数的消息完整性检查。

认证
客户机可以认证服务器，已认证的服务器可以认证客户机。 这意味着可以保证信息仅在既定方之间交换。 认证机制基于数字证书（X.509v3 证书）的交换。

安全套接字协议

在 MQIPT 中，使用传输层安全性 (TLS) 和安全套接字层 (SSL) 协议提供安全套接字。 两个安全套接字协议是类似的，但是不协同操作。 在本文档中，除非注意到特定差异，否则可互换使用 SSL 和 TLS 术语。

MQIPT 支持提供的 Java runtime environment (JRE) 提供的 SSL 3.0， TLS 1.0， TLS 1.1 和 TLS 1.2 。 V 9.2.5 从 IBM MQ 9.2.5 开始， MQIPT 还支持 TLS 1.3。 远程通道的 IBM MQ CipherSpec 确定 MQIPT 使用哪个协议。

SSL 3.0 不安全，因此缺省情况下在 MQIPT 中禁用。 V 9.2.0 缺省情况下，在 MQIPT from IBM MQ 9.1.4 中也会禁用 TLS 1.0 和 TLS 1.1 。 如果需要使用其中任何已禁用的协议，那么可以通过执行 第 878 页的『在 MQIPT 中启用不推荐的协议和 CipherSuites』中的过程来重新启用这些协议。

SSL/TLS 协议可以使用不同的数字签名算法对通信方进行认证。 在 SSL/TLS 中使用的加密操作，用于数据机密性的加密以及用于消息完整性的安全散列，都依赖于客户机与服务器之间的密钥共享。 SSL/TLS 提供各种密钥交换机制，允许共享密钥。 SSL/TLS 可以利用各种算法进行加密和散列。

在 MQIPT 中启用 FIPS 方式

JRE 的 SSL/TLS 加密组件包含以下经认证符合 FIPS 140-$tag1 标准的安全提供程序:

- V 9.2.0.5 V 9.2.4 从 IBM MQ 9.2.0 Fix Pack 5 或 IBM MQ 9.2.4， `IBMJCEPlusFIPS` 安全提供程序。
- 在 IBM MQ 9.2.0 Fix Pack 5 或 IBM MQ 9.2.4 之前的版本中， `IBMJSSEFIPS` 和 `IBMJCEFIPS` 安全提供程序。

支持各种密码算法; 请使用 SSL/TLS CipherSuites 来指定这些算法。 并非所有 CipherSuite 都是经过 FIPS 140-2 认证的。

如果要仅在 MQIPT 中使用 FIPS 认证的密码术，请在启动 MQIPT 时通过设置以下 Java 系统属性在 `IBMJSSE2` 提供程序中启用 FIPS 方式:

- `com.ibm.jsse2.usefipsprovider=true`
- V 9.2.0.5 V 9.2.4 从 IBM MQ 9.2.0 Fix Pack 5 或 IBM MQ 9.2.4, `com.ibm.jsse2.usefipsProviderName=IBMJCEPlusFIPS`

使用 **MQIPT_JVM_OPTIONS** 环境变量启动 MQIPT 时，可以设置 Java 系统属性。 例如，在 Linux 上，发出以下命令以设置环境变量，然后发出命令以启动 MQIPT:

```
export MQIPT_JVM_OPTIONS="-Dcom.ibm.jsse2.usefipsprovider=true
-Dcom.ibm.jsse2.usefipsProviderName=IBMJCEPlusFIPS"
```

有关启用 FIPS 模式的更多信息，请参阅在 FIPS 模式下运行 IBMJSSE2。

SSL/TLS 桥接方式

当路由同时设置了 SSLServer 和 SSLClient 时， MQIPT 接受一个入局 SSL/TLS 安全连接，并与另一个 MQIPT 或目标队列管理器建立第二个 SSL/TLS 安全连接。 IBM MQ 通道信息将在这两个 SSL/TLS 连接之间进行解密和重新加密。 SSL/TLS 桥接也称为 *SSL/TLS* 终止代理。

IBM MQ 支持使用 MQIPT 进行 SSL/TLS 桥接。 如果代理组合或重构的 SSL/TLS 记录大小与 IBM MQ 发送的 SSL/TLS 记录大小不同，那么已观察到具有 IBM MQ 的其他 SSL/TLS 终止代理会导致连接中断。 这是由于队列管理器为入局 IBM MQ 网络数据分配和管理内存的方式与将 IBM MQ 网络数据打包到 SSL/TLS 记录的方式之间的交互。

MQIPT 保留 SSL/TLS 记录中 IBM MQ 网络数据的打包，而不拆分或组合这些数据。 如果其他 SSL/TLS 网桥未完全保留 SSL/TLS 记录，那么它们可能会导致 IBM MQ 通道失败并产生错误消息:

```
AMQ9638: SSL communications error for channel
AMQ9208: Error on receive from host
```

SSL/TLS 代理方式

可以在 SSL/TLS 代理方式下配置 MQIPT 路由，作为 SSL/TLS 桥接的替代方法。 在此方式下，路由仅在两个 IBM MQ 端点之间转发 SSL/TLS 数据; 它不参与 SSL/TLS 握手，也不需要任何数字证书。

如果已针对 SSL/TLS 通信配置了通过 MQIPT 进行通信的 IBM MQ 通道，并且要将 MQIPT 用于其他用途 (例如，通过防火墙路由连接或通过安全出口限制允许的连接集)，那么可以使用 SSL/TLS 代理方式。 以 SSL/TLS 代理方式运行时， MQIPT 会先检查从新连接接收的初始 SSL/TLS 包是否有效，然后再将这些包转发到目标。

IBM MQ 支持使用 MQIPT 或任何其他 SSL/TLS 代理的 SSL/TLS 代理方式

具有 MQIPT 的 IBM MQ 多个证书支持

IBM MQ 8.0 和更高版本支持在同一队列管理器上使用多个证书，使用每个通道的证书标签 (使用通道定义上的 **CERTLABL** 属性指定)。 队列管理器的入站通道 (例如，服务器连接或接收方) 依赖于使用 TLS 服务器名称指示 (SNI) 来检测通道名称，以便提供来自队列管理器的正确证书。 有关在队列管理器上使用多个证书的更多信息，请参阅 如何 IBM MQ 提供多个证书功能。

如果通道通过 MQIPT 连接到目标队列管理器，并且 MQIPT 路由同时设置了 **SSLServer** 和 **SSLClient**，那么端点。 在低于 IBM MQ 9.2.5 的版本中， SNI 数据不会流经会话中断。 这将阻止在目标队列管理器上针对 MQIPT 与队列管理器之间的 TLS 连接使用每个通道的证书。 要在目标队列管理器上使用每个通道的证书，对于通过 MQIPT 且版本低于 IBM MQ 9.2.5 的 TLS 连接， MQIPT 路由必须使用 SSL/TLS 代理方式，这将完整地转发所有 TLS 控制流，包括 SNI 名称。

V 9.2.5 可以使用 SSLClientOutboundSNI 路由属性对 "IBM MQ 9.2.5、"MQIPT 进行配置，将 TLS 连接的 SNI 设置为特定值，或将入站连接收到的 SNI 传递给路由。 要允许在目标队列管理器上使用每个通道的证书，必须将路由配置为将 SNI 设置为 IBM MQ 通道名称，或者将在入站连接上接收到的 SNI 传递到路由。 如果 MQIPT 配置为通过 SNI，那么连接到 MQIPT 的队列管理器或客户机必须将 SNI 设置为通道名称。

可以针对每个路由单独配置用于由 MQIPT 终止或启动的 TLS 连接的证书，例如，使用 **SSLServerSiteLabel** 和 **SSLClientSiteLabel** 路由属性。

MQIPT 支持的 CipherSuites

下表显示 MQIPT 支持的 CipherSuite 以及缺省情况下启用的 CipherSuite。

缺省情况下，仅启用 CipherSuite 子集。 JRE 会禁用基于若干被视为不安全的算法的 CipherSuites 。 如果您已经意识到潜在的危险，但仍需要使用这些 CipherSuites,之一，您可以按照 "第 878 页的『在 MQIPT 中启用不推荐的协议和 CipherSuites』 中的步骤添加对禁用 CipherSuite 支持。

表 *87:* 可用于 *MQIPT* 的 *CipherSuites*

CipherSuite	**在缺省情况下启用**
CipherSuites for TLS 1.3	
V 9.2.5 V 9.2.5 TLS_AES_128_GCM_SHA256	Yes
V 9.2.5 V 9.2.5 TLS_AES_256_GCM_SHA384	Yes
V 9.2.5 V 9.2.5 TLS_CHACHA20_POLY1305_SHA256	Yes
CipherSuites for SSL 3.0， TLS 1.0， TLS 1.1 和 TLS 1.2	
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA	
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5	
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	
SSL_DH_anon_WITH_AES_128_CBC_SHA	
SSL_DH_anon_WITH_AES_128_CBC_SHA256	
SSL_DH_anon_WITH_AES_128_GCM_SHA256	
SSL_DH_anon_WITH_AES_256_CBC_SHA	

表 87: 可用于 *MQIPT* 的 *CipherSuites (继续)*	
CipherSuite	**在缺省情况下启用**
SSL_DH_anon_WITH_AES_256_CBC_SHA256	
SSL_DH_anon_WITH_AES_256_GCM_SHA384	
SSL_DH_anon_WITH_DES_CBC_SHA	
SSL_DH_anon_WITH_RC4_128_MD5	
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	
SSL_DHE_DSS_WITH_AES_128_CBC_SHA	V 9.2.0 是
SSL_DHE_DSS_WITH_AES_128_CBC_SHA256	V 9.2.0 是
SSL_DHE_DSS_WITH_AES_128_GCM_SHA256	V 9.2.0 是
SSL_DHE_DSS_WITH_AES_256_CBC_SHA	V 9.2.0 是
SSL_DHE_DSS_WITH_AES_256_CBC_SHA256	V 9.2.0 是
SSL_DHE_DSS_WITH_AES_256_GCM_SHA384	V 9.2.0 是
SSL_DHE_DSS_WITH_DES_CBC_SHA	
SSL_DHE_DSS_WITH_RC4_128_SHA	
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_DHE_RSA_WITH_AES_128_CBC_SHA	V 9.2.0 是
SSL_DHE_RSA_WITH_AES_128_CBC_SHA256	V 9.2.0 是
SSL_DHE_RSA_WITH_AES_128_GCM_SHA256	V 9.2.0 是
SSL_DHE_RSA_WITH_AES_256_CBC_SHA	V 9.2.0 是
SSL_DHE_RSA_WITH_AES_256_CBC_SHA256	V 9.2.0 是
SSL_DHE_RSA_WITH_AES_256_GCM_SHA384	V 9.2.0 是
SSL_DHE_RSA_WITH_DES_CBC_SHA	
SSL_ECDH_anon_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_anon_WITH_AES_128_CBC_SHA	
SSL_ECDH_anon_WITH_AES_256_CBC_SHA	
SSL_ECDH_anon_WITH_NULL_SHA	
SSL_ECDH_anon_WITH_RC4_128_SHA	
SSL_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA	Yes
SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	Yes

表 87: 可用于 *MQIPT* 的 *CipherSuites (*继续*)*

CipherSuite	**在缺省情况下启用**
SSL_ECDH_ECDSA_WITH_AES_128_GCM_SHA256	Yes
SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA	Yes
SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA384	Yes
SSL_ECDH_ECDSA_WITH_AES_256_GCM_SHA384	Yes
SSL_ECDH_ECDSA_WITH_NULL_SHA	
SSL_ECDH_ECDSA_WITH_RC4_128_SHA	
SSL_ECDH_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_RSA_WITH_AES_128_CBC_SHA	Yes
SSL_ECDH_RSA_WITH_AES_128_CBC_SHA256	Yes
SSL_ECDH_RSA_WITH_AES_128_GCM_SHA256	Yes
SSL_ECDH_RSA_WITH_AES_256_CBC_SHA	Yes
SSL_ECDH_RSA_WITH_AES_256_CBC_SHA384	Yes
SSL_ECDH_RSA_WITH_AES_256_GCM_SHA384	Yes
SSL_ECDH_RSA_WITH_NULL_SHA	
SSL_ECDH_RSA_WITH_RC4_128_SHA	
SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	Yes
SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	Yes
SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	Yes
SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	Yes
SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	Yes
SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	Yes
SSL_ECDHE_ECDSA_WITH_NULL_SHA	
SSL_ECDHE_ECDSA_WITH_RC4_128_SHA	
SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA	Yes
SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256	Yes
SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256	Yes
SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA	Yes
SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384	Yes
SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384	Yes
SSL_ECDHE_RSA_WITH_NULL_SHA	
SSL_ECDHE_RSA_WITH_RC4_128_SHA	
SSL_KRB5_EXPORT_WITH_DES_CBC_40_MD5	

表 87: 可用于 *MQIPT* 的 *CipherSuites (*继续*)*	
CipherSuite	**在缺省情况下启用**
SSL_KRB5_EXPORT_WITH_DES_CBC_40_SHA	
SSL_KRB5_EXPORT_WITH_RC4_40_MD5	
SSL_KRB5_EXPORT_WITH_RC4_40_SHA	
SSL_KRB5_WITH_3DES_EDE_CBC_MD5	
SSL_KRB5_WITH_3DES_EDE_CBC_SHA	
SSL_KRB5_WITH_DES_CBC_MD5	
SSL_KRB5_WITH_DES_CBC_SHA	
SSL_KRB5_WITH_RC4_128_MD5	
SSL_KRB5_WITH_RC4_128_SHA	
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	
SSL_RSA_EXPORT_WITH_RC4_40_MD5	
SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA（注释 1）	
SSL_RSA_FIPS_WITH_DES_CBC_SHA（注释 1）	
SSL_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_RSA_WITH_AES_128_CBC_SHA	Yes
SSL_RSA_WITH_AES_128_CBC_SHA256	Yes
SSL_RSA_WITH_AES_128_GCM_SHA256	Yes
SSL_RSA_WITH_AES_256_CBC_SHA	Yes
SSL_RSA_WITH_AES_256_CBC_SHA256	Yes
SSL_RSA_WITH_AES_256_GCM_SHA384	Yes
SSL_RSA_WITH_DES_CBC_SHA	
SSL_RSA_WITH_NULL_MD5	
SSL_RSA_WITH_NULL_SHA	
SSL_RSA_WITH_NULL_SHA256	
SSL_RSA_WITH_RC4_128_MD5	Yes
SSL_RSA_WITH_RC4_128_SHA	
V 9.2.2 TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	
V 9.2.2 TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	
V 9.2.2 TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	

注:

1. V 9.2.2 此 CipherSuite 在 MQIPT 中不受 IBM MQ 9.2.2 支持。 虽然此 CipherSuite 在 MQIPT 版本低于 IBM MQ 9.2.2 的版本中受支持，以便与先前版本兼容，但它不再符合 FIPS 标准，应该避免使用它。

CipherSpecs 和 MQIPT CipherSuites

下表显示 IBM MQ 支持的 CipherSpec 和 MQIPT 支持的 CipherSuite 之间的关系。

该表还显示 IBM MQ 期望每个 CipherSpec 使用的协议版本。

IBM MQ CipherSpec 将唯一地确定加密算法和要使用的安全套接字协议版本。 有些 IBM MQ CipherSpec 仅协议版本不同，所以不足以单独配置 CipherSuite。 SSL/TLS 握手协商双方支持的最高安全套接字协议版本，然后从相互启用的密码集中选择 CipherSuite 。

例如，`SSLClientCipherSuites=SSL_RSA_WITH_3DES_EDE_CBC_SHA` 的 SSLClient 路由可以与远程队列管理器协商 TLS_RSA_WITH_3DES_EDE_CBC_SHA (TLS 1.0) 或 TRIPLE_DES_SHA_US (SSL 3.0)。 事实上，可以通过 TLS 1.2 协商此 CipherSuite，但是 IBM MQ 不支持通过 TLS 1.2 使用此 CipherSuite。 鉴于此，SSLClient 路由很可能引发队列管理器的 AMQ9616 或 AMQ9631 错误。

要避免 SSLClient 路由上出现此类错误，将 **SSLClientProtocols** 路由属性设置为预期的 CipherSpec 的相应值。 在某些情况下，还有必要使用 **SSLServerProtocols** 路由属性限制服务器端的协议集。 使用表中显示的协议版本确定这些路由属性的正确设置。

此问题尤其影响 SSLClient 路由的以下 CipherSuite 和 CipherSpec：

- SSL_RSA_WITH_3DES_EDE_CBC_SHA，其对应于：
  - SSL 3.0：MQ CipherSpec TRIPLE_DES_SHA_US
  - TLS 1.0：MQ CipherSpec TLS_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_DES_CBC_SHA，其对应于：
  - SSL 3.0：MQ CipherSpec DES_SHA_EXPORT
  - TLS 1.0：MQ CipherSpec TLS_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_RC4_128_SHA，其对应于：
  - SSL 3.0：MQ CipherSpec RC4_SHA_US
  - TLS 1.2: MQ CipherSpec TLS_RSA_WITH_RC4_128_SHA256

如果想要使用单个 MQIPT SSLClient 路由通过隧道传送多个使用不同 CipherSpec 的 IBM MQ 通道，请确保所有通道彼此具有使用相同安全套接字协议版本的 CipherSpec，并且将 **SSLClientProtocols** 设置为使用此单个协议版本。

有关 IBM MQ CipherSpec 的更多信息，请参阅启用 CipherSpec。

IBM MQ CipherSpec	MQIPT CipherSuite	协议版本
DES_SHA_EXPORT	SSL_RSA_WITH_DES_CBC_SHA	SSLv3
DES_SHA_EXPORT1024	不适用	不适用
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	TLSv1.2
ECDHE_ECDSA_AES_128_CBC_SHA256	SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	TLSv1.2
ECDHE_ECDSA_AES_128_GCM_SHA256	SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	TLSv1.2
ECDHE_ECDSA_AES_256_CBC_SHA384	SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	TLSv1.2
ECDHE_ECDSA_AES_256_GCM_SHA384	SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLSv1.2
ECDHE_ECDSA_NULL_SHA256	SSL_ECDHE_ECDSA_WITH_NULL_SHA	TLSv1.2
ECDHE_ECDSA_RC4_128_SHA256	SSL_ECDHE_ECDSA_WITH_RC4_128_SHA	TLSv1.2
ECDHE_RSA_3DES_EDE_CBC_SHA256	SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1.2

IBM MQ CipherSpec	MQIPT CipherSuite	协议版本
ECDHE_RSA_AES_128_CBC_SHA256	SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLSv1.2
ECDHE_RSA_AES_128_GCM_SHA256	SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256	TLSv1.2
ECDHE_RSA_AES_256_CBC_SHA384	SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384	TLSv1.2
ECDHE_RSA_AES_256_GCM_SHA384	SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384	TLSv1.2
ECDHE_RSA_NULL_SHA256	SSL_ECDHE_RSA_WITH_NULL_SHA	TLSv1.2
ECDHE_RSA_RC4_128_SHA256	SSL_ECDHE_RSA_WITH_RC4_128_SHA	TLSv1.2
FIPS_WITH_3DES_EDE_CBC_SHA (Note 1)	SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA	SSLv3
FIPS_WITH_DES_CBC_SHA (Note 1)	SSL_RSA_FIPS_WITH_DES_CBC_SHA	SSLv3
NULL_MD5	SSL_RSA_WITH_NULL_MD5	SSLv3
NULL_SHA	SSL_RSA_WITH_NULL_SHA	SSLv3
RC2_MD5_EXPORT	不适用	不适用
RC4_56_SHA_EXPORT1024	不适用	不适用
RC4_MD5_EXPORT	SSL_RSA_EXPORT_WITH_RC4_40_MD5	SSLv3
RC4_MD5_US	SSL_RSA_WITH_RC4_128_MD5	SSLv3
RC4_SHA_US	SSL_RSA_WITH_RC4_128_SHA	SSLv3
V 9.2.5 V 9.2.5 TLS_AES_128_GCM_SHA256	TLS_AES_128_GCM_SHA256	TLSv1.3
V 9.2.5 V 9.2.5 TLS_AES_256_GCM_SHA384	TLS_AES_256_GCM_SHA384	TLSv1.3
V 9.2.5 V 9.2.5 TLS_CHACHA20_POLY1305_SHA256	TLS_CHACHA20_POLY1305_SHA256	TLSv1.3
TLS_RSA_WITH_3DES_EDE_CBC_SHA	SSL_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA	SSL_RSA_WITH_AES_128_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA256	SSL_RSA_WITH_AES_128_CBC_SHA256	TLSv1.2
TLS_RSA_WITH_AES_128_GCM_SHA256	SSL_RSA_WITH_AES_128_GCM_SHA256	TLSv1.2
TLS_RSA_WITH_AES_256_CBC_SHA	SSL_RSA_WITH_AES_256_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_256_CBC_SHA256	SSL_RSA_WITH_AES_256_CBC_SHA256	TLSv1.2
TLS_RSA_WITH_AES_256_GCM_SHA384	SSL_RSA_WITH_AES_256_GCM_SHA384	TLSv1.2
TLS_RSA_WITH_DES_CBC_SHA	SSL_RSA_WITH_DES_CBC_SHA	TLSv1
TLS_RSA_WITH_NULL_NULL	不适用	不适用
TLS_RSA_WITH_NULL_SHA256	SSL_RSA_WITH_NULL_SHA256	TLSv1.2
TLS_RSA_WITH_RC4_128_SHA256	SSL_RSA_WITH_RC4_128_SHA	TLSv1.2
TRIPLE_DES_SHA_US	SSL_RSA_WITH_3DES_EDE_CBC_SHA	SSLv3

注:

1. V 9.2.2 此 CipherSuite 在 MQIPT 中不受 IBM MQ 9.2.2 支持。 虽然此 CipherSuite 在 MQIPT 版本低于 IBM MQ 9.2.2 的版本中受支持，以便与先前版本兼容，但它不再符合 FIPS 标准，应该避免使用它。

MQIPT 中的 SSL/TLS 握手

在执行 CipherSuites 的认证和协商时， SSL/TLS 客户机与服务器之间的初始连接请求期间发生 SSL/TLS 握手过程。

所有受支持的 SSL/TLS CipherSuites (请参阅 第 858 页的『MQIPT 中的 SSL/TLS 支持』) (匿名 CipherSuites 除外) 都需要服务器认证并允许客户机认证; 可以将服务器配置为请求客户机认证。 您应该避免使用匿名 CipherSuite，因为它们不保证远程同级的身份。 中间人攻击可能在您不知情的情况下拦截匿名 SSL/TLS 连接。 只在值得信任的内部网络上并且您准备好接受数据拦截风险时，使用匿名 CipherSuite。

SSL/TLS 中的通信对等认证基于公用密钥密码术和 X.509v3 数字证书。 应该在 SSL/TLS 协议中进行认证的站点需要专用密钥和数字证书 (其中包含相应的公用密钥以及有关站点身份的信息)，证书的有效期。 证书由证书颁发机构签署，此类机构的证书称为签署者证书。 证书后面跟着一个或多个签署者证书，则构成证书链。 证书链的特点是，从第一个证书（站点证书）开始，可以使用下一个签署者证书中包含的公用密钥，验证链中每一个证书的签署。

正在建立需要服务器认证的安全连接时，服务器将向客户机发送证书链以证明其身份。 仅当 SSL/TLS 客户机可以对服务器进行认证 (例如，验证服务器站点证书的签名) 时，它才会继续建立与服务器的连接。 为了验证签名， SSL/TLS 客户机需要信任服务器站点本身或服务器提供的证书链中的至少一个签署者。 必须在客户机端维护信任站点和签署者的证书以执行此验证。

SSL/TLS 客户机从站点证书开始检查服务器的证书链。 在以下情况下，客户机认为站点证书的签名有效:

- 站点证书在信任站点或签署者证书的存储库中
- 可以根据信任签署者证书的存储库验证链中的签署者证书

在后一种情况下， SSL/TLS 客户机会检查证书链是否确实正确签名，从可信签署者证书到服务器的站点证书。 还会检查此过程中每个证书的格式是否正确以及日期的有效性。 如果其中任何一项检查没有通过，那么服务器连接将被拒绝。 验证服务器证书后，客户机将在 SSL/TLS 协议的后续步骤中使用该证书中嵌入的公用密钥。 仅当服务器确实具有相应的专用密钥时，才能建立 SSL/TLS 连接。

客户机认证遵循相同的过程: 如果 SSL/TLS 服务器需要客户机认证，那么客户机会向服务器发送证书链以证明其身份。 服务器根据信任站点和签署者证书的存储库验证链。 验证客户机的证书后，服务器将在 SSL/TLS 协议的后续步骤中使用该证书中嵌入的公用密钥。 仅当客户机确实具有相应的专用密钥时，才能建立 SSL/TLS 连接。

TLS 协议的最新版本提供高安全性通信 (SSL 和较旧的 TLS 协议被视为不安全)。 然而，协议根据应用程序提供的信息操作。 只有当该信息库也得到安全维护，才能实现安全通信的总体目标。 例如，如果您的信任站点和签署者证书的存储库陷入危险，您可能与非常不安全的通信合作伙伴建立了安全连接。

SSL/TLS 的 MQIPT 实现

SSL 3.0， TLS 1.0， TLS 1.1， TLS 1.2 和 TLS 1.3 是使用存储在密钥环文件 (文件类型为 `.p12` 或 `.pfx`) 中的公用密钥密码术标准 (PKCS) #12 令牌实现的，其中包含 X509.V3 证书。 V 9.2.0 MQIPT 还可以使用支持 PKCS#11 Cryptographic Token Interface 标准的加密硬件密钥库。 MQIPT 使用 IBM Java Secure Socket Extension (JSSE) 软件包。

MQIPT 可以充当 SSL/TLS 客户机或 SSL/TLS 服务器，具体取决于哪个端启动连接。 客户机启动连接，服务器接受连接请求。 MQIPT 路由可以同时充当客户机和服务器。 在这种情况下，使用 SSL/TLS 代理方式功能通常会提供更好的性能。

当针对 SSL/TLS 代理方式配置 MQIPT 时，它仅在两个端点之间转发 SSL/TLS 数据; 它不参与 SSL/TLS 握手，并且不需要任何数字证书。

在 IBM MQ 9.2.5 之前的版本中， MQIPT 不会将入站 TLS 连接上接收到的 TLS 服务器名称指示 (SNI) 数据传递到出站 TLS 连接。 这意味着使用 **CERTLABL** 通道属性指定的每个通道证书不能用于 MQIPT 与目标队列管理器之间的 TLS 连接。 要在目标队列管理器上使用每个通道的证书，对于通过 MQIPT 版本低于 IBM MQ 9.2.5， 的 TLS 连接， MQIPT 路由必须使用 SSL/TLS 代理方式，这将完整地转发所有 TLS 控制流，包括

SNI 名称。 V 9.2.5 从 IBM MQ 9.2.5 开始， MQIPT 可以配置为将 TLS 连接的 SNI 设置为特定值，或者通过入站连接上接收到的 SNI 传递到路由。 有关将队列管理器上的多个证书与 MQIPT 配合使用的更多信息，请参阅 第 860 页的『具有 MQIPT 的 IBM MQ 多个证书支持』。

可以使用其自己的一组 SSL/TLS 属性来独立配置每个 MQIPT 路由。 请参阅 MQIPT 路由属性 以获取更多详细信息。

在 MQIPT 中加密密钥环密码

使用 **mqiptPW** 命令对用于打开密钥环文件或用于访问 MQIPT 所使用的加密硬件的密码进行加密。 以下任何属性都可以使用加密密码: **SSLClientKeyRingPW**， **SSLClientCAKeyRingPW**, **SSLServerKeyRingPW**， **SSLServerCAKeyRingPW** 和 **SSLCommandPortKeyRingPW**。 本主题描述了存储密钥环密码以供 MQIPT 使用的正确方法。

MQIPT 不支持 **mqiptkeyman** (iKeyman) 隐藏文件工具。 作为使用存储文件的替代做法，您必须使用 **mqiptPW** 命令存储加密后的密码。

在 IBM MQ 9.1.5 之前的版本中， MQIPT 使用的密钥环密码存储在任何 **SSL*KeyRingPW** 属性引用的文件中。

V 9.2.0 从 IBM MQ 9.1.5 开始，使用 **mqiptPW** 命令对供 MQIPT 使用的密钥环密码进行加密，并将 **SSL*KeyRingPW** 属性的值设置为加密密码。 MQIPT 能够区分属性值中的加密密码和文件名，以便与 IBM MQ 9.1.5 之前创建的配置兼容。

不推荐使用在低于 IBM MQ 9.1.5 的 MQIPT 版本中可用的密钥库密码加密方法，但仍可使用此方法。 为提高对密钥环密码的保护，请使用最新的保护方法对先前已加密的任何密钥环密码进行重新加密。

要加密密钥环密码以供 MQIPT 使用，请执行 加密存储的密码中的步骤。

必须使用密码 `mqiptSample` 来打开 MQIPT 安装目录的 `samples/ssl` 子目录中提供的任何样本密钥环文件。

从 MQIPT 中的密钥环文件选择证书

可以将多个个人证书存储在同一密钥环文件或加密硬件令牌中。 可以在客户机端使用 **SSLClientSite*** 属性来选择要发送到服务器以进行认证的证书，可以在服务器端使用 **SSLServerSite*** 属性来选择要发送到客户机以进行认证的证书。

通过使用这些属性，可以根据专有名称 (DN) 来选择证书。 或者，可通过 **SSLServerSiteLabel** 和 **SSLClientSiteLabel** 属性使用证书标签来选择证书。

V 9.2.0 V 9.2.0 要选择 TLS 命令端口所使用的服务器证书，请使用 **SSLCommandPortSiteLabel** 属性来指定证书的标签名称。

MQIPT 中的信任设置

密钥环包含包含签署者证书或签署者证书链的个人证书。

MQIPT 使用两种类型的密钥环:

认证中心 (CA) 密钥环

此密钥环包含用于验证属于远程同级的证书的可信 CA 证书。 这些 CA 证书有助于确定远程同级是否可信。 MQIPT 支持 PKCS #12 格式密钥环文件和支持 PKCS #11 接口的加密硬件密钥库，用于存储 CA 证书。 MQIPT CA 密钥环文件由 **SSLClientCAKeyRing** 和 **SSLServerCAKeyRing** 路由属性标识。 通过设置 **SSLClientCAKeyRingUseCryptoHardware** 和 **SSLServerCAKeyRingUseCryptoHardware** 属性，可以使用加密硬件来访问 CA 证书。

SSL/TLS 客户机端上的 CA 密钥环应包含将用于认证从服务器发送的证书的可信 CA 证书列表。 如果为客户机认证配置了 SSL 服务器路由，那么 SSL/TLS 服务器端的 CA 密钥环应包含将用于认证从客户机发送的证书的可信 CA 证书列表。

个人证书密钥环

此密钥环包含 MQIPT 用于向远程同级标识自身的个人证书。 生成自签名证书或请求 CA 签名的证书时，应使用个人证书密钥环来执行此操作。 MQIPT 支持 PKCS #12 格式密钥环文件和支持 PKCS #11 接口的

加密硬件密钥库，用于存储个人证书。 在 MQIPT 中，个人证书密钥环文件由 **SSLClientKeyRing** 和 **SSLServerKeyRing** 路由属性标识。 通过设置 **SSLClientKeyRingUseCryptoHardware** 和 **SSLServerKeyRingUseCryptoHardware** 属性，可以使用加密硬件来访问个人证书。

SSL/TLS 服务器端的密钥环应包含 MQIPT 服务器的个人证书。 如果需要在 SSL 客户机路由上进行客户机认证，那么 SSL/TLS 客户机端上的密钥环应包含客户机的个人证书。

如果需要客户机认证，那么必须在服务器端启用 **SSLServerAskClientAuth** 属性。 客户端上的密钥环应包含客户端的个人证书。 服务器端的 MQIPT 密钥环 (由 **SSLServerCAKeyRing** 属性标识) 应包含将用于认证客户机的可信 CA 证书的列表。

如果没有为路由配置 CA 密钥环，那么 MQIPT 将改为在个人证书密钥环中搜索 CA 证书 (如果已配置)。 例如，如果没有为 **SSLServerCAKeyRing** 设置任何值，那么 MQIPT 将在由 **SSLServerKeyRing** 标识的密钥环中搜索 CA 证书。

作为使用受信 CA 签署的证书的替代方式，您可以使用自签名证书。 您可以在 `samples/ssl` 子目录中随 MQIPT 提供的 `sslSample.pfx` 样本密钥环文件中找到自签名证书的示例。 要打开样本 PKCS#12 密钥环文件，必须使用密码 `mqiptSample`。

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 SSL/TLS 连接，而无需为证书支付 CA 费用。 但是，不应在生产环境中使用自签名证书。 要创建 CA 签名的证书，请参阅 创建密钥环文件。

您可以使用 MQIPT 随附的名为 **mqiptkeyman** 的实用程序来管理数字证书和密钥库。 请参阅 第 870 页的『MQIPT 中的 mqiptKeyman 和 mqiptKeycmd』 获取安装指示信息和更多信息。

您必须使用操作系统的安全功能来保护任何密钥环文件和密码文件，以防止对它们进行未经授权的访问。

在 MQIPT 中测试 SSL/TLS

您可以使用本文档中提供的示例来测试 SSL/TLS 连接。

请参阅 IBM MQ Internet Pass-Thru 入门，以获取各种方案的描述。 尤其要查看以下任务：

- 认证 SSL/TLS 服务器
- 认证 SSL/TLS 客户机
- 以 SSL/TLS 代理方式运行 MQIPT
- 使用安全管理器以 SSL/TLS 代理方式运行 MQIPT

要测试 SSL/TLS 配置是否正常工作，可以使用自签名证书。 自签名证书在测试场景中很有用，因此您可以确保 SSL/TLS 连接，而无需向认证中心 (CA) 支付证书费用。 请参阅 创建测试证书 以获取详细信息。

您可以在 `samples/ssl` 子目录中随 MQIPT 提供的 `sslSample.pfx` 样本密钥环文件中找到自签名证书的示例。 要打开样本 PKCS #12 密钥环文件，必须使用密码 `mqiptSample`。 为方便您在测试期间提供样本证书。 但是，样本证书的专用密钥对于所有 MQIPT 用户都是已知的。 这意味着它不安全，应该仅在测试环境中使用。

您不能在生产环境中使用任何自签名证书，无论这些证书是否是样本证书。 相反，应从受信 CA 获取 CA 签名的证书。 要创建 CA 签名的证书，请参阅 创建密钥环文件。

在创建或请求证书时，您应考虑哪些密钥类型、密钥大小和数字签名算法适用于您的安全需求。 请参阅第 872 页的『MQIPT 的数字证书注意事项』，以了解更多信息。

证书和证书管理技术可从众多第三方供应商处获取。

MQIPT 中的 SSL/TLS 错误消息

握手失败以 JSSE 异常形式记录到 MQIPT 连接日志中。

有关更多信息，请参阅第 891 页的『MQIPT 中的连接日志』。 下表描述了不同的异常、可能的原因和解决此失败的相应操作。

证书异常通常与连接的远程端的证书相关。

如果错误与 IBM MQ 客户机或队列管理器的证书相关，那么术语 密钥环文件 包含远程合作伙伴的 IBM MQ 密钥存储库。

在 MQIPT 中， CA 证书存储在 CA 密钥环文件中，该文件由 **SSLClientCAKeyRing** 和 **SSLServerCAKeyRing** 路由属性标识。 如果未设置 CA 密钥环路由属性，那么将改为搜索相应的个人密钥环文件 (由 **SSLClientKeyRing** 或 **SSLServerKeyRing** 属性引用) 以查找 CA 证书。

异常	原因	操作
CertificateException	该证书不受信任，因为它是由不在 CA 密钥环中的 CA 签署的。	请检查 CA 密钥环文件中是否存在所有必需的 CA 证书。 使用 MQIPT 随附的 IBM 密钥管理工具来添加任何缺少的 CA 证书，请注意从值得信任的源获取每个 CA 证书的副本。
CertificateExpiredException	1. 证书已到期：已超过 **notAfter** 日期。 2. 系统时钟设置不正确。	1. 获取新证书并将其插入到密钥环文件中。 如果证书属于认证中心，请将新证书放入 CA 密钥环文件中。 2. 请检查 UTC 系统时钟是否设置为正确时间。
CertificateNotYetValidException	1. 过早使用此证书：尚未到达 **notBefore** 日期。 2. 系统时钟设置不正确。	1. 请检查证书的生成和签署是否正确。 如果您的组织使用自己的 CA 操作，可能是 CA 的 UTC 系统时钟不正确。 2. 请检查 UTC 系统时钟是否设置为正确时间。
CertificateParsingException	1. 证书包含无效 DER 数据。 2. 此证书使用了不支持的 DER 功能。	确保已正确生成证书，并且可以在 IBM 随 MQIPT 提供的密钥管理工具中查看该证书。 考虑获取带有较少证书扩展名的新证书。
CertificateRevokedException	证书撤销检查已启用并发现此证书被撤销。	所提及的证书不能信任。 获取替换证书，并确保新证书及其专用密钥存在于密钥环文件中。
CertPathBuilderException	认证中心未对证书链进行签名。	1. 如果您正在使用 CA 签名的证书，请检查 CA 密钥环文件中是否存在所有根 CA 证书和中间 CA 证书。 2. 如果您正在使用自签名证书，请确保已抽取远程证书的公用部分的副本并将其添加到 CA 密钥环文件中。 避免在生产环境中使用自签名证书。

异常	原因	操作
CertStoreException KeyStoreException	由于下列其中一个原因，从密钥环读取证书时发生错误: 1. 密钥环文件已损坏。 2. 缺少密钥环文件。 3. 存储的密码与密钥环文件密码不匹配。 4. 如果路由配置为使用加密硬件，那么 MQIPT 无法连接到加密硬件。	1. 确保可以读取密钥环文件，并且可以使用 IBM 密钥管理工具来查看所有证书。 2. 请检查所有密钥环路由属性是否引用了正确的文件名。 3. 请检查存储的密钥环文件密码是否正确。 使用 **mqiptPW** 工具存储正确的密码。 4. 如果路由配置为使用加密硬件，请检查以下内容: • Java 安全属性文件指定安装 IBMPKCS11Impl 安全提供程序。 • Java 安全属性文件包含用于初始化 IBMPKCS11Impl 安全提供程序的配置文件的标准名称。 • 用于初始化 IBMPKCS11Impl 安全提供程序的配置文件有效。
SSLException: 没有与启用的 SSL 密码套件对应的可用证书或密钥。	您必须有个人证书，且该证书的密钥类型对于您正在使用的密码套件正确无误。 例如，名称以 SSL_ECDH_ECDSA_ 开头的密码套件需要带有椭圆曲线公用密钥的证书。 最常用的密码套件需要带有 RSA 公用密钥的证书。	使用 IBM Key Management 工具打开密钥环文件。 在“个人证书”视图下，依次选择各个证书并进行查看。 单击**查看详细信息**并导航到“主题公用密钥”部分以查看公用密钥类型。 然后检查 MQIPT **SSLClientCipherSuites** 和 **SSLServerCipherSuites** 路由属性以确保已启用适当的密码套件。
SSLException: 无公共密码套件 SSLHandshakeException: 无公共密码套件	握手未能同意密码套件，因为连接两端的已启用密码套件的设置之间没有重叠。 特别是出站 IBM MQ 连接只启用了一个密码，因此 SSLServer MQIPT 路由极有可能会遇到此错误。 当下列所有三个条件都为 true 时，也会发生此错误: • 在路由上未指定 CipherSuite • 在为路由配置的密钥环中找不到合适的站点证书 • 已禁用匿名 CipherSuites	请检查 MQIPT **SSLClientCipherSuites** 和 **SSLServerCipherSuites** 路由属性中已启用密码套件的列表。 考虑启用其他密码套件。 请参考所提供的表来确定要为各个 IBM MQ 通道 CipherSpec 值启用的正确密码套件。 如果未在路由上指定 CipherSuite，请检查密钥环路由属性是否引用了正确的密钥环文件，以及密钥环是否包含 MQIPT 可以使用的个人证书。 如果路由配置为使用加密硬件，请检查用于初始化 IBMPKCS11Impl 安全提供程序的配置文件中的 **tokenlabel** 属性是否指定了正确的加密设备令牌标签。

MQIPT 中的 mqiptKeyman 和 mqiptKeycmd

mqiptKeyman (iKeyman) 是 IBM MQ 用户已经熟悉的证书和密钥管理应用程序。 **mqiptKeyman** 和 **mqiptKeycmd** 命令可用于管理 IBM MQ Internet Pass-Thru 所使用的密钥环文件中的对称密钥，非对称密钥，数字证书和证书请求。 这些文件还可用于管理密钥环文件本身。

mqiptKeyman 和 **mqiptKeycmd** 使用术语 密钥数据库 来引用密钥环文件; 这些术语是同义词。

iKeyman 可采用以下两种方式运行：图形用户界面 (GUI) 和命令行界面 (CLI)。 使用 **mqiptKeyman** 命令来启动 GUI，使用 **mqiptKeycmd** 命令来运行 CLI。

用于在 IBM MQ 中管理证书的等效命令是 **strmqikm** 用于启动 GUI， **runmqckm** 用于运行 CLI。 使用 **runmqckm**， **runmqakm** 和 **strmqikm** 来管理数字证书中描述了 IBM MQ 命令。

MQIPT 所需的密钥环文件格式

创建密钥环文件以在 MQIPT 中使用时，必须使用 PKCS #12 文件格式:

- 在 UI 中，创建密钥环文件时，在 **密钥数据库类型** 字段中选择 PKCS#12 。
- 在 CLI 中，在 `mqiptKeycmd -keydb -create` 命令上包含 `-type pkcs12` 参数。

V 9.2.0 MQIPT 还可以访问存储在支持 PKCS #11 接口的加密硬件中的证书。 该接口还可用于管理 PKCS #11 硬件上的证书。 有关更多信息，请参阅 第 879 页的『在 MQIPT 中使用 PKCS #11 加密硬件』。

对 MQIPT 的密钥环密码进行加密

创建密钥环文件后，必须以 MQIPT 可用于访问该文件的格式对密钥环密码进行加密。 有关信息，请参阅第 867 页的『在 MQIPT 中加密密钥环密码』。

请注意， MQIPT 不支持隐藏文件工具。 必须使用 **mqiptPW** 命令来加密密钥环密码，而不是使用隐藏文件。

命令行示例

CLI 使用与 IBM MQ **runmqckm** 命令相同的语法。 将所需的参数附加到 **mqiptKeycmd** 命令中，如以下示例所示:

- 要创建 PKCS#12 文件:

```
mqiptKeycmd -keydb -create -db key.p12 -pw password -type pkcs12
```

- 要创建用于测试目的的自签名个人证书:

```
mqiptKeycmd -cert -create -db key.p12 -pw password -type pkcs12
-label mqipt -dn "CN=Test Certificate,OU=Sales,O=Example,C=US"
-sig_alg SHA256WithRSA -size 2048
```

  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。 请参阅 第 872 页的『MQIPT 的数字证书注意事项』 以获取更多信息。

  此示例使用适用于测试目的的自签名证书。 但是，在生产环境中应使用由认证中心签署的证书。

  请注意， MQIPT V2.0 和更低版本不支持 SHA-2 数字签名，因此该证书不适用于建立与先前 MQIPT 发行版的安全套接字连接；将需要使用较低版本的签名算法，例如 SHA1WithRSA。

- 要因生产目的而创建针对认证中心签名证书的证书请求:

```
mqiptKeycmd -certreq -create -db key.p12 -pw password -type pkcs12 -file cert.req
-label mqipt -dn "CN=Test Certificate,OU=Sales,O=Example,C=US"
-sig_alg SHA256WithRSA -size 2048
```

  该命令将创建使用 2048 位 RSA 公用密钥的数字证书请求以及将 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。 请参阅 第 872 页的『MQIPT 的数字证书注意事项』 以获取更多信息。

- 要将 CA 签名的个人证书文件 `cert.crt` 接收到密钥环文件中:

```
mqiptKeycmd -cert -receive -db key.p12 -pw password -type pkcs12 -file cert.crt
```

  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，例如:

```
mqiptKeycmd -cert -add -db key.p12 -pw password -type pkcs12 -file ca.crt -label rootCA
```

MQIPT 的数字证书注意事项

要考虑的点包括证书密钥大小，选择相应的证书数字签名算法和数字证书以及 CipherSuite compatibilityDigital 证书和 CipherSuite 兼容性。

MQIPT 的证书密钥大小注意事项

公用密钥大小取决于组织的安全策略，并且取决于所使用的加密算法。 一般而言，密钥大小越大，就越安全。 下表列出了应该使用的最小密钥大小：

算法	最小密钥大小（位）
Elliptic Curve	256
RSA	2048

在创建证书或证书请求时，应指定证书的密钥大小。

- 使用 **mqiptKeycmd** CLI 命令时，可使用 **-size** 参数来指定密钥大小。
- 使用 **mqiptKeyman** GUI 时， "证书创建" 窗口中的 **密钥大小** 字段指定密钥大小。

选择合适的证书数字签名算法

为防止伪造数字证书，请务必使用强大的数字签名算法。 创建或请求证书时，请谨慎选择合适的算法。

应避免使用基于 MD5 或 SHA-1 的旧数字签名算法，因为这些算法对于新式应用而言不够安全。 如果可以，请使用基于 SHA-2 的最新数字签名算法，例如，具备 RSA 功能的 SHA-256 算法 (SHA256WithRSA)。

但是，低于 V2.1 的 MQIPT 版本不支持 SHA-2 数字签名，因此，为了实现与先前 MQIPT 发行版的互操作性，请使用 SHA1WithRSA 数字签名算法。 但是，您应计划升级较低版本的 MQIPT，并分阶段停止使用 MD5 和 SHA-1 数字签名。

- 使用 **mqiptKeycmd** CLI 命令时，可使用 **-sig_alg** 参数来指定数字签名算法。
- 使用 **mqiptKeyman** GUI 时， "证书创建" 窗口的 **签名算法** 字段指定数字签名算法。

MQIPT 中的数字证书和 CipherSuite 兼容性

并非任何密码套件都可用于所有数字证书。 存在多种类型的密码套件（按密码套件名称前缀分组）。 每种类型的密码套件都对可用的数字证书类型施加了不同的限制。 这些限制适用于所有 MQIPT SSL/TLS 连接，但与椭圆曲线密码术的用户特别相关。 执行安全套接字握手时，MQIPT 会自动选择个人证书来标识该证书适用于所协商的密码套件。 在大多数情况下，MQIPT 会自动与远程同级进行互操作。 但是，在某些情况下，可能需要使用特定的 MQIPT 密码套件来与远程 IBM MQ 系统进行互操作。 MQIPT 随附的 **mqiptKeyman** 应用程序只能使用 DSA 和 RSA 公用密钥创建证书和证书请求。 此外，IBM MQ **runmqakm** 实用程序可以创建使用 Elliptic Curve 公用密钥的证书和证书请求。 请咨询认证中心，以获取有关创建其他类型证书的建议。

要使用的数字证书类型取决于您所使用的密码套件类型：

- 名称以 SSL_ECDH_ECDSA_ 和 SSL_ECDHE_ECDSA_ 开头的密码套件要求提供使用 Elliptic Curve 公用密钥的数字证书。
- 名称中包含 *anon* 的密码套件是匿名密码套件，不需要数字证书即可识别远程同级。 此类密码套件可避免在使用备用认证方法的网络中产生证书生命周期管理方面的开销，但通常会因为缺少认证而不使用这类密码套件。
- 其他密码套件要求提供使用 RSA 公用密钥的数字证书。

注: **mqiptKeyman** 和 **mqiptKeycmd** 工具无法创建使用 Elliptic Curve 公用密钥的证书或证书请求。 您可以使用 IBM MQ 随附的 **runmqakm** 命令来实现此目的。 使用 **runmqckm**， **runmqakm** 和 **strmqikm** 来管理数字证书中描述了 **runmqakm** 命令。

MQIPT 中的证书出口

证书出口的用途是验证 MQIPT 接收的 SSL/TLS 对等证书。

您可以将 MQIPT 路由配置为在建立新连接时充当 SSL/TLS 客户机，并在接收连接请求时充当 SSL/TLS 服务器。 在 SSL/TLS 握手过程中， SSL/TLS 客户机从服务器接收对等证书，该证书可用于认证服务器。 SSL/TLS 服务器还可以从客户机接收对等证书，该证书可用于认证客户机。

MQIPT 接收到同级证书时调用证书出口，允许您执行进一步验证。 出口捕获的任何异常也由 MQIPT 捕获，连接请求终止。 因此，最佳做法是，出口捕获所有异常并将相应的返回码传递回 MQIPT。

提供了样本以显示可实现的证书出口，有关更多信息，请参阅 使用证书出口来认证 SSL/TLS 服务器。

注: MQIPT 在单个 Java virtual machine 中运行，因此用户定义的证书出口可能会以下列其中一种方式危及 MQIPT 的正常操作:

- 影响系统资源
- 生成瓶颈
- 降低性能

在证书出口实施于生产环境之前，应对证书出口的影响进行广泛测试。

MQIPT 中的 *com.ibm.mq.ipt.exit.CertificateExit* 类

com.ibm.mq.ipt.exit.CertificateExit 类是一个抽象类，必须由使用 SSLExitName 属性定义的类实现。

该类包含运行出口和一些公共方法（您可以根据自己的需求选择性地改写）的缺省实施。 下面是所支持的方法的完整列表:

方法

public int init(IPTTrace)

当 MQIPT 装入出口并且可以实施该出口以执行出口的任何初始化时， MQIPT 调用此 ini 方法；例如，装入验证过程使用的数据。 缺省实施不执行任何操作。

public int refresh(IPTTrace)

实施此刷新方法以执行任何数据的刷新；例如，重新装入验证过程期间使用的任何磁盘数据。 MQIPT 管理员发出刷新命令时调用此方法。 缺省实施不执行任何操作。

public void close(IPTTrace)

当路由将要停止或者 MQIPT 正关闭时，实施此关闭方法以执行任何管理。 缺省实施不执行任何操作。

public CertificateExitResponse validate(IPTTrace)

执行同级证书验证时调用此验证方法。 可使用返回对象将信息传递回 MQIPT；例如，可以添加到连接日志的返回码和一些文本。 缺省实施返回 CertificateExitResponse 为 CertificateExitResponse.OK。

支持的获取属性方法:

public int getListenerPort()
检索路由侦听器端口 - 由 ListenerPort 属性定义

public String getDestination()
检索目标地址 - 由 Destination 属性定义

public int getDestinationPort()
检索目标侦听器端口地址 - 由 DestinationPort 属性定义

public String getClientIPAddress()
检索发出连接请求的客户机的 IP 地址。

public int getClientPortAddress()
检索客户机发出连接请求使用的端口地址

public boolean isSSLClient()
用于确定是否将出口作为 SSL/TLS 客户机或 SSL/TLS 服务器调用。 如果返回 true，出口在连接的客户机端，验证从服务器获取的证书。 如果返回 false，出口在连接的服务器端，验证客户机发送的证书。 它对于同时充当 SSL/TLS 服务器和 SSL/TLS 客户机，解密和重新加密流量的路由有效。 在这种情况下，尽管存在单个出口类，类的某些实例将作为客户机调用，某些实例作为服务器调用。 您可以使用 isSSLClient 确定给定实例的情况。

public int getConnThreadID()
用于检索处理连接请求的工作程序线程的标识，该标识可用于调试。

public String getChannelName()
检索连接请求中使用的 IBM MQ 通道名称。 仅当入局请求未使用 SSL/TLS 并且 MQIPT 充当 SSL/TLS 客户机时，此功能才可用。

public String getQMName()
检索连接请求中使用的 IBM MQ 队列管理器的名称。 仅当客户机请求未使用 SSL/TLS 并且 MQIPT 充当 SSL/TLS 客户机时，此功能才可用。

public boolean getTimedout()
出口用于确定超时是否到期。

public IPTCertificate getCertificate()
检索需要验证的 SSL/TLS 证书。

public String getExitData()
检索 SSLExitData 属性所定义的出口数据。

public String getExitName()
检索 SSLExitName 属性所定义的出口名称。

MQIPT 中的 *com.ibm.mq.ipt.exit.CertificateExitResponse* 类

此类用于在验证证书后将信息传递回 MQIPT。

构造函数

public CertificateExitResponse(*int rc*, string message)

此构造函数可用于往回传递返回码和一些消息文本。 可能的原因码：

- ExitRc.OK
- ExitRc.VALIDATE_ERROR
- ExitRc.VALIDATE_REJECTED

public CertificateExitResponse(*int rc*)

此构造函数可用于往回传递返回码，不传递消息文本。 可能的原因码：

- ExitRc.OK
- ExitRc.VALIDATE_ERROR
- ExitRc.VALIDATE_REJECTED

public CertificateExitResponse()

此构造函数可用于往回传递返回码 ExitRc.OK，不传递消息文本。

方法

public String getVersion()

此方法返回此类的版本。

public String toString

此方法返回响应的字符串表示，例如“Reason code: 4, Message: Failed CRL check”。

MQIPT 中的 *com.ibm.mq.ipt.exit.IPTCertificate* 类

此类包含要验证的 SSL/TLS 证书。

方法

public int getVersion()

此方法返回此类的版本。

public byte [] getDerEncoding()

此方法返回 X.509 证书的 ASN.1/DER 编码或 NULL（如果存在错误）。

public byte [] getPemEncoding()

此方法返回 X.509 证书的 PEM (BASE64) 编码或 NULL（如果存在错误）。

public String getLabel()

此方法返回证书标签或 NULL（如果存在错误）。

public String getName()

此方法返回证书的专有名称或 NULL（如果不可用）。 例如：

```
CN=Test Queue Manager,OU=Sales,O=Example,L=London,C=GB
```

public String getIssuerName()

此方法返回证书发行者的专有名称或 NULL（如果不可用）。 例如：

```
CN=Certificate Authority,OU=Security,O=Example,L=New York,C=US
```

public IPTCertificate getSigner()

此方法返回签署者证书或 NULL（如果不可用）。 对于自签名证书，将返回对其自身的引用。

public String toString()

此方法返回证书的字符串表示。

MQIPT 中的 *com.ibm.mq.ipt.exit.IPTTrace* 类

MQIPT 跟踪函数提供了入口和出口调用，可以在方法的入口和出口上使用这些调用。 还有各种用于跟踪有用信息的数据调用。

方法

public void entry(String *fid*)

其中 *fid* 用于标识进行调用的位置，例如类和方法名称。

此方法将以相应的缩进级别将入口写入跟踪输出文件以记录控制流进入方法的点。 此调用是可选的，但是一旦使用此调用，同一方法中还必须使用对“exit(String)”的匹配调用。

public void exit(String *fid*)

其中 *fid* 用于标识进行调用的位置，例如类和方法名称。

此方法将以相应的缩进级别将出口写入跟踪输出文件以记录控制流离开方法的点。 仅在同一方法内先前已使用“entry(String)”调用的情况下才会使用此方法。

public void exit(String *fid*, int *rc*)

其中 *fid* 用于标识进行调用的位置（例如类和方法名称），*rc* 是来自此方法的数字返回码。 此跟踪方法应用于记录可返回整数的方法的出口。

此方法将以相应的缩进级别将出口写入跟踪输出文件以记录控制流离开方法的点和该方法返回的数字返回码。 仅在同一方法内先前已使用“entry(String)”调用的情况下才会使用此方法。

public void exit(String *fid*, boolean *rc*)

其中 *fid* 用于标识进行调用的位置（例如类和方法名称），*rc* 是来自此方法的布尔值返回码。 此跟踪方法应用于记录可返回布尔值的方法的出口。

此方法将以相应的缩进级别将出口写入跟踪输出文件以记录控制流离开方法的点和该方法返回的布尔值返回码。 仅在同一方法内先前已使用“entry(String)”调用的情况下才会使用此方法。

public void data(String *fid*, String data)

其中 *fid* 用于标识进行调用的位置，例如类和方法名称。

此方法可将某些字符串数据写入跟踪输出文件。

public void data(String *fid*, int data)

其中 *fid* 用于标识进行调用的位置，例如类和方法名称。

此方法可将某些整数数据写入跟踪输出文件。

public void data(String *fid*, byte[])

其中 *fid* 用于标识进行调用的位置，例如类和方法名称。

此方法可将某些二进制数据写入跟踪输出文件。

样本跟踪

为了帮助诊断出口问题，您可以使用与 MQIPT 相同的跟踪功能，也可以实施您自己的跟踪功能。 如果您决定使用 MQIPT 跟踪功能，那么有一些可用于进入和退出方法的入口和出口调用。 如以下示例所示，还有各种数据调用可以跟踪有用信息。

```
/**
* This method is called to initialize the exit (for example, for
* loading validation information) and place itself in a ready
* state to validate connection requests.
*/
 public int init(IPTTrace t) {
 final String fid = "MyExit.init";

 // Trace entry into this method
 t.entry(fid);

 // Trace useful information
 t.data(fid, "Starting exit - MQIPT version " + getVersion());

 // Perform initialization and load any data
 t.data(fid, "Ready for work");

 // Trace exit from this method
 t.exit(fid);

 return ExitRc.OK;
}
```

此方法将按照以下示例所示的格式生成跟踪：

```
16:36:48.625    14   5000-1s    ------{ ConnectionThread.setCertificateExit()
16:36:48.625    14   5000-1s     Creating instance of certificate exit
16:36:48.625    14   5000-1s     Calling init() of certificate exit
16:36:48.625    14   5000-1s    -------{ MyExit.init()
16:36:48.625    14   5000-1s     Starting exit - MQIPT version 2.1.0.0
16:36:48.625    14   5000-1s    Ready for work
16:36:48.625    14   5000-1s    -------} MyExit.init() rc=0
16:36:48.625    14   5000-1s    ------} ConnectionThread.setCertificateExit() rc=0
```

MQIPT 中的证书出口返回码

MQIPT 在许多不同情况下调用证书出口时识别的返回码。

在以下情况下调用证书出口时，MQIPT 识别以下返回码：

返回码	描述	初始化	Validate	刷新
ExitRc.OK	请求已成功完成。	yes	yes	yes
ExitRc.INIT_ERROR	初始化请求失败，将禁用路由。	yes		
ExitRc.REFRESH_ERROR	刷新请求失败，将禁用路由。			yes
ExitRc.VALIDATE_ERROR	验证过程失败，连接请求被拒绝。		yes	
ExitRc.VALIDATE_REJECTED	验证请求被拒绝，连接请求被拒绝。		yes	

MQIPT 中的 LDAP 和 CRL

MQIPT 支持使用轻量级目录访问协议 (LDAP) 服务器对数字证书执行证书撤销列表 (CRL) 认证。

LDAP 支持已通过类似方式实施到 IBM MQ，因为同一 LDAP 服务器可用于 IBM MQ 和 MQIPT。

在 SSL/TLS 握手期间，通信伙伴使用数字证书相互认证。 认证过程中，可以检查所接收到的证书是否仍然可信。 认证机构 (CA) 会因为以下各种原因撤销证书：

- 所有者已经移到不同的组织。
- 专用密钥不再加密。

CA 在证书撤销列表 (CRL) 中发布已撤销的个人证书。 已经被撤销的 CA 证书发布在权限撤销列表 (ARL) 中。 请注意，后续对 CRL 的引用也适用于 ARL。

有关将 LDAP 服务器与 IBM MQ 配合使用以及有关 CRL 和 ARL 管理的更多信息，请参阅 使用证书撤销列表和权限撤销列表。

MQIPT 最多支持每个路由上有两个 LDAP 服务器。 第一个 LDAP 服务器将被视为主要服务器，第二个 LDAP 服务器将保留为备份服务器。 仅在无法连接主服务器时才会使用第二个服务器。 备份服务器应该是主服务器的镜像映像。

通过使用 LDAP 用户标识和密码属性，可以使用用户标识和密码来保护对存储在 LDAP 服务器上的信息的访问。 V 9.2.0 可以在 MQIPT 配置中从 IBM MQ 9.1.5 对 LDAP 服务器密码进行加密。 有关对 MQIPT 要使用的密码进行加密的更多信息，请参阅 对存储的密码进行加密。

当 MQIPT 从密钥环文件装入 PKCS #12 令牌时，将检查任何 CA 证书的 CRL 有效性。 如果 CA 证书有附加的 CRL，将检查此 CRL 是否到期，如果到期，将从 LDAP 服务器检索更新的 CRL。 任何检索到的 CRL 将加载到当前令牌并附加到其 CA 证书。

如果向主 LDAP 服务器发送查询时没有与指定 CA 匹配的条目，则认为没有用于此 CA 的 CRL 且不会使用备份服务器。 但是，如果无法连接主 LDAP 服务器或者未在指定时间范围内返回，将使用备份服务器。 来自备份服务器的任何错误都会导致客户机连接终止。 通过将 **LDAPIgnoreErrors** 属性设置为 `true` 可以覆盖此操作。

MQIPT 检索到的任何 CRL 均保留在缓存中并由该路由上的所有连接共享。 如果缓存的 CRL 已到期，那么此 CRL 将从缓存中移除并从 LDAP 服务器中检索一个新的 CRL。 如果新的 CRL 不可用，那么仍会拒绝连接。

也会对从 LDAP 服务器检索到的 CRL 检查是否到期并将显示一则警告消息 (MQCPW001)。 到期的 CRL 仍会加载到系统中，但引用此 CRL 的任何连接请求会被拒绝。 您应该使用当前 CRL 替换 LDAP 服务器中到期的 CRL。

LDAPCacheTimeout 属性可用于控制清除 CRL 缓存的频率。 缺省值是 1 天。 此值设置为 0 表示路由重新启动之前不会清除缓存条目。

到期的 CRL 可以存储在密钥环文件中，也可以存储在 LDAP 服务器上。 如果新的 CRL 尚未发布，将拒绝后续连接请求。 通过启用 **IgnoreExpiredCRLs** 属性可忽略到期的 CRL。

注: 如果启用 **LDAPIgnoreErrors** 属性或 **IgnoreExpiredCRLs** 属性，那么可使用已撤销的证书来建立 SSL/TLS 连接。

MQIPT 中的多值证书专有名称 OU 属性

您可以匹配证书专有名称中的多个组织单位 (OU) 值。

下面的路由属性现在支持匹配多个 OU 值：

- **SSLClientDN_OU**
- **SSLClientSiteDN_OU**
- **SSLServerDN_OU**
- **SSLServerSiteDN_OU**

要匹配多个 OU 值，在路由属性值中使用逗号作为分隔符。 例如：

```
SSLClientDN_OU=Sales, Europe
```

这表示匹配 OU=Sales 和 OU=Europe 的证书。 OU 值的匹配顺序与 IBM MQ SSLPEER 过滤器中的多个 OU 值顺序相同。

在 [route] 部分中，请勿多次指定相同的路由属性。 如前面的示例所示，匹配多个 OU 值的正确方式是指定一次属性。 如果在同一 mqipt.conf 部分中多次输入相同属性，那么最后一个值生效。 例如，下面的输入将生成唯一匹配 Europe，因为第二行覆盖第一行：

```
SSLClientDN_OU=Sales
SSLClientDN_OU=Europe
```

如果必须匹配 OU 值内的文字逗号，那么紧贴逗号前面插入反斜杠 (\) 作为转义字符。 例如：

```
SSLClientDN_OU=Sales\, Europe
```

这样匹配单一值：OU=Sales, Europe。 后面没有紧接逗号的反斜杠匹配文字反斜杠。

如果从先前版本的 MQIPT 升级并信任匹配 OU 值中的逗号功能，那么必须将反斜杠转义字符插入 OU 路由属性，以保存先前行为。。

在 MQIPT 中启用不推荐的协议和 CipherSuites

缺省情况下，在 MQIPT 随附的 Java runtime environment (JRE) 中禁用被视为不安全的安全套接字协议和密码套件。 必须先启用这些不推荐使用的协议和密码套件，然后才能使用这些协议和密码套件。

关于此任务

如果您知道潜在危险，但仍需要使用在 MQIPT 中被视为不安全的某个协议或密码套件，请遵循此过程来启用需要使用的协议或密码套件。

注: V 9.2.0 V 9.2.0 不推荐使用的协议和密码套件不能与 TLS 命令端口配合使用。

过程

1. 编辑在 *mqipt_path*/java/jre/lib/security 目录中找到的 java.security 文件，其中 *mqipt_path* 是 MQIPT 的安装位置。
2. 通过从 jdk.tls.disabledAlgorithms 属性的已禁用算法列表中除去相应条目，向 JRE 添加对协议或算法的支持。
   - V 9.2.0 要添加对协议的支持，请从禁用算法列表中除去该协议。 例如，要添加对 TLS 1.0 的支持，请从列表中除去 TLSv1 。
   - 要添加对密码套件的支持，请从禁用算法列表中除去相应的算法。 例如，要添加对 SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA 密码套件的支持，请从列表中除去 3DES_EDE_CBC 和 DESede 。
3. V 9.2.0
   要在 JRE 中启用 SSL 3.0 ，还需要设置系统属性 com.ibm.jsse2.disableSSLv3=false。

如果是使用 **mqipt** 命令从命令行启动 MQIPT，那么可以使用环境变量 **MQIPT_JVM_OPTIONS** 来设置属性。 例如：

```
set MQIPT_JVM_OPTIONS=-Dcom.ibm.jsse2.disableSSLv3=false
```

Windows V 9.2.0.4 V 9.2.4 如果 MQIPT 安装为 Windows 服务，那么您可以通过在 Windows 注册表中的 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MQInternetPassThru 键下定义字符串值来设置该属性。 该值应该具有以下属性：

名称
MqiptJvm 选项

值数据
-Dcom.ibm.jsse2.disableSSLv3=false

4. 要在 MQIPT 路由上启用 SSL 3.0， TLS 1.0 或 TLS 1.1，请将相应协议添加到 **SSLServerProtocols** 或 **SSLClientProtocols** 路由属性。
5. 重新启动 MQIPT 以使对 JRE 属性的更改生效。

V 9.2.0 在 MQIPT 中使用 PKCS #11 加密硬件

MQIPT 可以访问存储在支持 PKCS #11 接口的加密硬件中的数字证书。

开始之前

在开始配置 MQIPT 以使用加密硬件之前，请确保加密卡，卡驱动程序和任何关联的支持软件已安装并正常运行。

对 MQIPT 中 PKCS #11 加密硬件的支持由 IBM Java PKCS11 加密提供程序 (IBMPKCS11Impl 提供程序) 提供。 有关 IBMPKCS11Impl 提供程序以及 Java 8 支持的加密卡列表的更多信息，请参阅 IBM PKCS11 加密提供程序。

关于此任务

您可以将 MQIPT 访问的个人证书和 CA 证书存储在加密硬件密钥库中。 但是，由于 PKCS #11 设备通常没有足够的空间来存储大量签署者证书，因此您可能希望对 CA 证书使用单独的基于文件的密钥库。

遵循此过程以将 MQIPT 配置为使用加密硬件密钥库中的证书。

注: 将加密硬件与 MQIPT 配合使用是 IBM MQ Advanced 功能。 要使用此功能，还需要使用 MQIPT 路由连接的本地队列管理器具有 IBM MQ Advanced， IBM MQ Appliance， IBM MQ Advanced for z/OS VUE 或 IBM MQ Advanced for z/OS 权利。

过程

1. 创建在初始化 IBMPKCS11Impl 提供程序时使用的配置文件。

   下载 IBMPKCS11Impl 提供程序支持的每个硬件加密卡的样本配置文件，并为系统配置样本。 可以从 Java 的 IBM Documentation 中的以下主题下载样本: 配置文件。

   配置文件是文本文件，应该至少包含以下属性：

   名称
   提供程序实例的名称后缀。

   库
   随加密硬件提供的 PKCS #11 库的标准名称。

   记号标签
   PKCS #11 加密设备令牌标签。

   例如，配置文件可能包含以下条目：

```
name = IPTPKCS11Provider
library = /usr/lib64/pkcs11/PKCS11_API.so
tokenlabel = icatoken
```

2. 编辑位于 MQIPT 安装目录的 `java/jre/lib/security` 子目录中的 Java 安全属性文件 `java.security`。
   a) 如果该文件中尚不存在，请添加 IBMPKCS11Impl 安全提供程序。
   例如，通过添加以下行:

   ```
   security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
   ```

   b) 在提供程序名称之后添加配置文件的标准名称。
   例如，如果您在步骤 第 879 页的『1』 中创建的配置文件名为 `/opt/mqipt/pkcs11.cfg`，那么应该将此路径添加到与安全提供程序相同的行中:

   ```
   security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl /opt/mqipt/
   pkcs11.cfg
   ```

3. 如果要将密钥环文件用于 CA 证书，而不是将 CA 证书存储在加密硬件中，请创建 PKCS #12 格式的 CA 密钥环文件。
   您可以使用 **mqiptKeyman** 图形用户界面 (GUI) 或 **mqiptKeycmd** 命令行界面 (CLI) 来创建 CA 密钥环文件。
   - 要使用 CLI，请输入以下命令:

     ```
     mqiptKeycmd -keydb -create -db filename -pw password -type pkcs12
     ```

     其中 *filename* 是要创建的密钥环文件的名称， *password* 是密钥环密码。
   - 要使用 GUI，请执行以下步骤:
     a. 通过输入命令 **mqiptKeyman** 来启动 GUI。
     b. 单击 **密钥数据库文件** > **打开**。
     c. 单击 **密钥数据库类型**，然后选择 **PKCS11Config**。
     d. 单击**确定**。 "打开密码令牌" 窗口将打开。
     e. 选择要用于存储证书的加密设备令牌标签。
     f. 在 **加密令牌密码** 字段中，输入访问加密硬件所需的密码。
     g. 要创建新的 CA 密钥环文件，请选择 **创建新的辅助密钥数据库文件**。
     h. 单击 **密钥数据库类型**，然后选择 **PKCS12**。
     i. 在 **文件名** 字段中，输入 CA 密钥环的文件名。
     j. 在 **位置** 字段中，输入 CA 密钥环文件的完整路径。
     k. 单击**确定**。 这样会打开“密码提示”窗口。
     l. 在 **密码** 字段中输入 CA 密钥环的密码，然后在 **确认密码** 字段中再次输入该密码。
     m. 单击**确定**。

4. 使用 **mqiptKeycmd** 或 **mqiptKeyman**，请求加密硬件的个人证书。
   - 要使用 CLI，请输入以下命令:

     ```
     mqiptKeycmd -certreq -create -crypto module_name -tokenlabel hardware_token
                 -pw password -label label -size key_size
                 -sig_alg algorithm -dn distinguished_name -file filename
     ```

     其中:

     -crypto *module_name*
     指定随加密硬件一起提供的 PKCS #11 库的标准名称。

     -tokenlabel *token_label*
     指定 PKCS #11 加密设备令牌标签。

-pw *password*
指定用于访问加密硬件的密码。

-label *label*
指定证书标签。

-size 键大小
指定密钥大小。 该值可以是 512， 1024， 2048 或 4096。

-sig_alg 算法
指定用于创建条目的密钥对的非对称签名算法。 该值可以是 MD2_WITH_RSA， MD2WithRSA， MD5_WITH_RSA， MD5WithRSA， SHA1WithDSA， SHA1WithECDSA， SHA1WithRSA， SHA2/ECDSA， SHA224WithECDSA， SHA256_WITH_RSA。 SHA256WithECDSA， SHA256WithRSA， SHA2WithECDSA， SHA3/ECDSA， SHA384_WITH_RSA， SHA384WithECDSA， SHA384WithRSA， SHA3WithECDSA， SHA5/ECDSA， SHA512_WITH_RSA， SHA512WithECDSA， SHA512WithRSA， SHA5WithECDSA， SHA_WITH_DSA， SHA_WITH_RSA 或 SHAWithDSA。 缺省值为 SHA256WithRSA。

-dn 区分名称
指定用双引号括起的 X.500 专有名称。

-file 文件名
指定证书请求的文件名。

- 要使用 GUI，请执行以下步骤:
  a. 从 **创建** 菜单中，单击 **新建证书请求**。
  b. 在 **密钥标签** 字段中，输入证书标签。
  c. 选择所需的 **密钥大小** 和 **签名算法** 。
  d. 输入 **公共名称** 和 **组织**的值，然后选择 **国家或地区**。 对于其余可选字段，请接受缺省值，或者输入或选择新值。
  e. 在 **输入要在其中存储证书请求的文件的名称** 字段中，接受缺省值 certreq.arm，或者输入具有完整路径的新值。
  f. 单击**确定**。
  g. **个人证书请求** 列表显示您创建的新个人证书请求的标签。 证书请求存储在您选择的文件中。

5. 在 CA 向您发送个人证书之后，将 CA 证书添加到密钥库或 CA 密钥环文件 (如果尚未存在)。
   - 要使用 CLI 将 CA 证书添加到 CA 密钥环文件，请输入以下命令:

```
mqiptKeycmd -cert -add -db filename -pw password -type pkcs12
            -label label -file cert_filename
```

   其中 *filename* 是 CA 密钥环文件的名称， *password* 是 CA 密钥环密码， *label* 是附加到证书的标签， *cert_filename* 是包含 CA 证书的文件的名称。
   - 要使用 CLI 将 CA 证书添加到加密硬件，请输入以下命令:

```
mqiptKeycmd -cert -add -crypto module_name -tokenlabel hardware_token
            -pw password -label label -file cert_filename
```

   其中 *module_name* 是随加密硬件提供的 PKCS #11 库的标准名称， *hardware_token* 是 PKCS #11 加密设备令牌标签， *password* 是用于访问加密硬件的密码， *label* 是附加到证书的标签， *cert_filename* 是包含 CA 证书的文件的名称。
   - 要使用 GUI，请执行以下步骤:
     a. 在**密钥数据库内容**字段中，选择**签署者证书**。
     b. 单击**添加**。 这样会打开“从文件添加 CA 证书”窗口。
     c. 输入证书文件名和用于存储证书的位置，或单击**浏览**以选择名称和位置。
     d. 单击**确定**。 这样会打开“输入标签”窗口。
     e. 在“输入标签”窗口中，输入证书名称。
     f. 单击**确定**。 这样会将该证书添加至密钥数据库中。

6. 将 CA 提供的个人证书接收到加密硬件密钥库中。
   - 要使用 CLI，请输入以下命令:

     ```
     mqiptKeycmd -cert -receive -file filename -crypto module_name
                 -tokenlabel hardware_token -pw password
     ```

     其中 *filename* 是包含要接收的证书的文件的名称， *module_name* 是随加密硬件提供的 PKCS #11 库的标准名称， *hardware_token* 是 PKCS #11 加密设备令牌标签， *password* 是用于访问加密硬件的密码。

     如果 CA 证书存储在 CA 密钥环中，而不是存储在加密硬件中，那么您将收到一条警告，指示无法验证证书链，因为在将个人证书接收到密钥库中时， **mqiptKeycmd** 命令无法访问 CA 密钥环。
   - 要使用 GUI，请执行以下步骤:
     a. 单击 **接收**。 "从文件接收证书" 窗口将打开。
     b. 输入新个人证书的证书文件名和位置，或者单击 **浏览** 以选择名称和位置。
     c. 单击**确定**。 **个人证书** 字段显示您添加的新个人证书的标签。
7. 使用 **mqiptPW** 命令对密码进行加密以访问加密硬件。

   V 9.2.0 输入以下命令:

   ```
   mqiptPW -sf encryption_key_file
   ```

   其中 *encryption_key_file* 是包含 MQIPT 安装的密码加密密钥的文件的名称。 如果 MQIPT 安装正在使用缺省密码加密密钥，那么不需要指定 **-sf** 参数。 输入密码以在提示时访问要加密的加密硬件。

   有关加密密钥库密码的更多信息，请参阅 第 867 页的『在 MQIPT 中加密密钥环密码』。
8. 如果在步骤 第 880 页的『3』中创建了 CA 密钥环文件，请遵循步骤 第 882 页的『7』中的指示信息对 CA 密钥环文件的密码进行加密。
9. 编辑 `mqipt.conf` 配置文件。
   a) 通过将 **EnableAdvancedCapabilities** 全局属性设置为 `true`，确认您具有使用此 IBM MQ Advanced 功能的相应权利。
   b) 通过将一个或多个 **SSLServerKeyRingUseCryptoHardware**，**SSLServerCAKeyRingUseCryptoHardware**， **SSLServerKeyRingUseCryptoHardware** 或 **SSLServerKeyRingUseCryptoHardware** 属性设置为 `true`，允许在路由上使用加密硬件密钥库。

   有关允许在路由上使用加密硬件的属性的更多信息，请参阅 MQIPT 路由属性。

   V 9.2.0 V 9.2.0 从 IBM MQ 9.2.0 开始，您还可以通过将 **SSLCommandPortKeyRingUseCryptoHardware** 属性设置为 `true`，将加密硬件与 TLS 命令端口配合使用。
   c) 如果要将密钥环文件用于 CA 证书，请通过设置一个或多个 **SSLServerCAKeyRing** 或 **SSLServerCAKeyRing** 属性来指定 CA 密钥环的位置。

   如果已配置路由以将加密硬件用于站点证书，并且未指定 CA 密钥环文件，那么将使用加密硬件密钥库作为 CA 密钥库。
   d) 使用 **SSLServerKeyRingPW**， **SSLServerCAKeyRingPW**， **SSLClientKeyRingPW**，**SSLClientCAKeyRingPW** 或 V 9.2.0 V 9.2.0 **SSLCommandPortKeyRingPW** 属性指定用于访问加密硬件和 CA 密钥环的加密密码。

   V 9.2.0 将 **SSL*KeyRingPW** 属性的值设置为 **mqiptPW** 命令输出的加密密码。
   e) 如果加密硬件包含多个个人证书，请指定 MQIPT 应选择哪个证书以发送到 SSL/TLS 服务器或客户机进行认证。

   您可以通过为 SSL/TLS 客户机路由设置一个或多个 **SSLClientSite*** 属性或者为 SSL/TLS 服务器路由设置一个或多个 **SSLServerSite*** 属性来指定应该选择的证书。

V 9.2.0 V 9.2.0 通过使用 **SSLCommandPortSiteLabel** 属性来指定证书标签名称，可以指定 TLS 命令端口应使用的证书。

有关从密钥环中选择证书的更多信息，请参阅 第 867 页的『从 MQIPT 中的密钥环文件选择证书』。 MQIPT 路由属性中描述了用于从密钥环中选择证书的属性。

V 9.2.0 例如，要将加密硬件密钥库用于 TLS 服务器路由上的站点证书，并使用密钥环文件来存储同一路由的 CA 证书，请将以下属性添加到路由定义中:

```
SSLServerKeyRingUseCryptoHardware=true
SSLServerKeyRingPW=<mqiptPW>1!gORdM4wft5d1rCgNMDEGag==!dZxhgQD2A8Ea0yeqawQvPg==
SSLServerCAKeyRing=/opt/mqipt/ssl/ca.pfx
SSLServerCAKeyRingPW=<mqiptPW>1!3Vdrpiu6kMwn0sWRCVgT5g==!LHltGLEg3OFvN8+02Re0YA==
SSLServerSiteLabel=mqiptsite
```

10. 重新启动 MQIPT。

Java security manager 在 MQIPT 中

Java security manager 可以与任何 MQIPT 功能部件配合使用，以提供更高级别的安全性。

MQIPT 使用 `java.lang.SecurityManager` 类中定义的缺省 Java security manager 。 可以使用全局属性 **SecurityManager** 来启用或禁用 MQIPT 中的 Java security manager 功能部件。 请参阅 MQIPT 全局属性 以获取更多信息。

Java security manager 使用两个缺省策略文件:

- 名为 *$MQIPT_PATH*`/java/jre/lib/security/java.policy`（其中 *$MQIPT_PATH* 是 MQIPT 的安装目录）的全局系统策略文件，供主机上虚拟机的所有实例使用。
- 名为 `.java.policy` 的特定于用户的策略文件，可存在于用户的主目录中。

还可以使用其他 MQIPT 策略文件。 您应该使用 MQIPT 策略文件，而不是之前描述的缺省策略文件。 请参阅 MQIPT 全局属性 中的 **SecurityManagerPolicy** 以获取更多信息。

尽管策略文件的语法可以使用文本编辑器进行更改，但这些语法相当复杂，通常使用随 Java 一起提供的策略工具实用程序进行更改会更加容易。 策略工具实用程序可从 *$MQIPT_PATH*`/java/jre/bin` 目录中找到并在 Java 文档内有完整记录。

MQIPT 随附了样本策略文件 (`mqiptSample.policy`)，向您显示必须设置哪些许可权才能运行 MQIPT。

您必须编辑样本策略文件以与您的配置相匹配。 请特别注意，包含 `mqipt.conf` 配置文件的 MQIPT 主目录可能与 MQIPT 安装目录不同，因此在安全策略中配置 **FilePermission** 条目时，请注意指定正确的目录。

您必须更改以下条目:

- 授予对 `errors` 目录的 **读** 和 **写** 访问权的 **java.io.FilePermission** 条目。 此条目中的文件路径必须引用 MQIPT 主目录，因为这是 `errors` 目录所在的位置。 MQIPT 在 `errors` 目录中创建 FFST 故障数据捕获文件 (`AMQ*.FDC`) 和跟踪文件 (`AMQ*.TRC*`)。 您必须确保 MQIPT 有权在 `errors` 目录中创建跟踪文件和 FFST 文件，以便可以进行故障诊断。
- 授予对 `logs` 目录的 **读** 和 **写** 访问权的 **java.io.FilePermission** 条目。 此条目中的文件路径必须引用 MQIPT 主目录，因为这是 `logs` 目录所在的位置。 如果启用了 **ConnectionLog** 全局属性，那么 MQIPT 将在 `logs` 目录中创建连接日志文件 (`mqipt*.log`)。
- 授予 `read` 和 `execute` 访问 MQIPT 安装目录中的任何目录 (例如 `bin`， `exits`， `lib` 和 `ssl` 目录) 的权限的 **java.io.FilePermission** 条目。 这些条目中的文件路径必须更改为引用 MQIPT 安装目录。 可以省略其中某些条目（如果不需要）。
- 必须修改 **java.net.SocketPermission** 条目以控制指向各个侦听 MQIPT 路由的连接。 每个 MQIPT 路由的侦听器端口和侦听器地址都需要 `listen` 和 `accept` 许可权。
- 必须修改 **java.net.SocketPermission** 条目以控制各个 MQIPT 路由发出的连接。 对于 MQIPT 路由连接到的任何路由目标，代理服务器或 LDAP 服务器，都需要 `connect` 许可权。 使用主机名而不是 IP 地址指定目标时，需要 `resolve` 许可权。

根据您的配置，您可能还需要添加以下条目:

- **java.io.FilePermission** 条目，用于授予对 mqipt.conf 配置文件或包含 mqipt.conf 的 MQIPT 主目录的读访问权。
- **java.io.FilePermission** 条目，授予对安全策略文件自身的读访问权。 如果 MQIPT 刷新会导致重新读取安全策略文件，那么此条目非常有用。
- 某些 **java.io.FilePermission** 条目授予对任何 SSL/TLS 密钥环文件和密钥环密码文件的读访问权。 仅当使用已启用 **SSLClient** 或 **SSLServer** 属性的路由时，或者配置 TLS 命令端口时，才需要执行此操作。
- 某些 **java.io.FilePermission** 条目授予对任何 MQIPT 出口类的 read 或 execute 访问权。 仅在启用 MQIPT 出口时才需要这些条目。 如果出口需要，您可能需要授予其他许可权。

注: Windows **java.io.FilePermission** 条目必须对路径中的每个反斜杠使用两个反斜杠字符 (\\)。 这是因为单个反斜杠用作转义字符。

样本文件假定 MQIPT 已安装在 C:\Program Files\IBM\MQ Internet Pass-Thru 中的 Windows 系统上。 同时假设 MQIPT 主目录（mqipt.conf 文件的位置）与 MQIPT 安装目录相同。

如果已在其他位置安装 MQIPT ，那么必须更改 **codeBase** 定义中的目录以引用 MQIPT 安装目录。 请注意包含正确的前缀 (file:/) 和正确的文件后缀 (/lib/com.ibm.mq.ipt.jar)。 在 AIX and Linux 系统上，典型的 **codeBase** URL 可能是 file:/opt/mqipt/lib/com.ibm.mq.ipt.jar，假定 MQIPT 安装在 /opt/mqipt 中。

许可权通常通过三个属性定义。 为了控制套接字连接，其值为:

类权限
java.net.SocketPermission

要控制的名称
以 hostname:port 格式构成，其中名称的各个组成部分可以使用通配符指定。 主机名可以是域名或 IP 地址。 主机名的最左边位置可以由星号 (*) 指定。 例如， harry.company1.com 将由以下每个字符串匹配:

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

名称的端口组成部分可以指定为单个端口地址或一定范围内的端口地址，例如:

1414
只有端口 1414

1414-
所有大于或等于 1414 的端口地址

-1414
所有小于或等于 1414 的端口地址

1-1414
1 到 1414 之间的所有端口地址（包括 1 和 1414）

允许的操作
java.net.SocketPermission 使用的操作有:

接受
允许接受来自指定目标的连接

connect
允许与指定目标的连接

侦听
允许应用程序在指定的一个或多个端口上侦听连接请求

解决
允许使用 DNS 将域名解析为 IP 地址

还可以通过 java.security.manager 和 java.security.policy Java 系统属性来控制 Java security manager，但建议您使用 **SecurityManager** 和 **SecurityManagerPolicy** 属性来控制 MQIPT。

要在跟踪和 FFST 记录中包含诊断信息，MQIPT 必须访问特定的 MQIPT 系统属性和环境变量。必须始终在 Java 安全策略中包含以下属性：

V 9.2.0 V 9.2.0

```
permission java.util.PropertyPermission "java.home", "read";
permission java.util.PropertyPermission "java.version", "read";
permission java.util.PropertyPermission "java.runtime.version", "read";
permission java.util.PropertyPermission "java.vm.info", "read";
permission java.util.PropertyPermission "java.vm.vendor", "read";
permission java.util.PropertyPermission "os.arch", "read";
permission java.util.PropertyPermission "os.name", "read";
permission java.util.PropertyPermission "os.version", "read";
permission java.lang.RuntimePermission "getenv.MQIPT_PATH";
permission java.lang.RuntimePermission "getStackTrace";
permission javax.management.MBeanServerPermission "createMBeanServer";
permission javax.management.MBeanPermission "com.ibm.mq.ipt.IPTManager#-
[com.ibm.mq.ipt:type=IPTManager]", "registerMBean";
permission javax.management.MBeanPermission "com.ibm.mq.ipt.IPTManager#-
[com.ibm.mq.ipt:type=IPTManager]", "unregisterMBean";
permission javax.management.MBeanTrustPermission "register";
```

如果未包含所有这些属性，MQIPT 将无法正常运行，并且会影响问题诊断。

MQIPT 中的安全出口

使用安全出口控制对 **Destination** 路由属性所定义的目标的访问。当 MQIPT 收到来自客户机的连接请求时，连接目标之前会调用安全出口。

根据初始连接属性，安全出口将决定是否允许连接完成。

路由启动时，将调用安全出口进行初始化并使自身准备好处理连接请求。初始化处理应用于加载所有用户数据并准备好数据以便快速轻松地进行访问，从而最大程度地减少处理连接请求所需的时间。

每个路由都可以有自己的安全出口。

- **SecurityExit** 属性用于启用/禁用用户定义的安全出口。
- **SecurityExitName** 属性用于定义用户定义的安全出口的类名。
- **SecurityExitPath** 属性用于定义包含类文件的目录名称。如果不设置此属性，会认为类文件将在 exits 子目录中找到。**SecurityExitPath** 还可以定义包含用户定义的安全出口的 JAR 文件的名称。
- **SecurityExitTimeout** 属性由 MQIPT 用以确定验证连接请求时等待安全出口响应的时间。

有关安全出口属性的详细信息，请参阅 MQIPT 路由属性 。

MQIPT 使用 SecurityExit 类调用用户定义的安全出口。该类必须由用户定义的安全出口进行扩展且其大部分方法将被覆盖以提供所需的功能。SecurityExitResponse 对象用于将数据传回 MQIPT，且这些数据将由 MQIPT 用以决定此连接请求应该接受还是应该拒绝。SecurityExitResponse 对象还可以包含新的目标和目标端口地址，用于覆盖安全出口属性定义的路由。

提供了三个样本安全出口来向您展示如何实施安全出口。

- SampleSecurityExit 显示了如何根据 IBM MQ 通道的名称控制对 IBM MQ 队列管理器的访问。此安全出口只允许通道名称以字符串“MQIPT”开头的连接。请参阅 使用安全出口 以获取更多信息。
- SampleRoutingExit 允许将客户机连接请求动态路由到已定义的 IBM MQ 服务器池中，每个服务器托管相同名称和相同属性的队列管理器。此样本包括的配置文件中包含服务器名称列表。请参阅 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 以获取更多信息。
- SampleOneRouteExit 允许动态路由到 IBM MQ 队列管理器，该队列管理器源自连接请求中使用的 IBM MQ 通道名称。此样本包括的配置文件中包含队列管理器名称到服务器名称的映射。请参阅 动态路由客户机连接请求 以获取更多信息。

注: MQIPT 在单个 JVM 中运行，所以用户定义的安全出口可能通过以下某个方式影响 MQIPT 的正常操作：

- 影响系统资源
- 生成瓶颈

- 降低性能

在安全出口实施于生产环境之前，应对安全出口的影响进行广泛测试。

MQIPT 中的 com.ibm.mq.ipt.exit.SecurityExit 类

此类及其公共方法必须通过用户定义的安全出口进行扩展才能访问某些通用数据并允许某些 MQIPT 初始化发生。

在 MQIPT 调用各个方法前，将提供某些属性供方法使用。 可通过使用此类中定义的相应的 get 方法检索它们的值。

方法

public int init(IPTTrace)

以下属性可用：

- 侦听器端口
- destination
- 目标端口
- 版本

`init` 方法由 MQIPT 在启动路由时调用。 在此方法返回时，安全出口必须准备好验证连接请求。 有效返回码为 `ExitRc.OK` 或 `ExitRc.INIT_ERROR`。

public int refresh(IPTTrace)

以下属性可用：

- 侦听器端口
- destination
- 目标端口

刷新 MQIPT 配置时， MQIPT 将调用 `refresh` 方法。 当配置文件中的属性已更改时，通常会执行此操作。 MQIPT 从配置文件重新装入所有属性，以确定哪些属性已更改以及是否需要重新启动路由。

此方法应执行其使用的任何外部数据的重新装入; 即，由 **init** 方法装入的数据。 有效返回码为 `ExitRc.OK` 或 `ExitRc.REFRESH_ERROR`。

public void close(IPTTrace)

以下属性可用：

- 侦听器端口
- destination
- 目标端口

`close` 方法在停止时由 MQIPT 调用。 此方法应释放出口在其操作期间获取的任何系统资源。 MQIPT 在关闭之前等待此方法完成。

如果先前已启用安全出口，但现在已在配置文件中禁用，那么也会调用此方法。

public SecurityExitResponse validate(IPTTrace)

以下属性可用：

- 侦听器端口
- destination
- 目标端口
- timeout
- 客户机 IP 地址
- 客户机端口地址
- 渠道名称

- 队列管理器名称

  validate 方法由 MQIPT 在接收到要验证的连接请求时调用。 如果已启用 **SSLProxyMode** 属性，那么通道名称和队列管理器名称将不可用，因为此功能仅用于隧道 TLS 数据，因此通常从初始数据流获取的数据不可读。

安全出口必须返回包含以下信息的 SecurityExitResponse 对象:

- 原因码（必须设置）
- 新的目标地址（可选）
- 新目标侦听器端口地址（可选）
- 消息（可选）

原因码确定 MQIPT 是接受还是拒绝连接。 可以选择设置 newDestination 和 newDestinationPort 字段以定义新的目标队列管理器。 如果未设置这些属性，那么将使用配置文件中定义的路径 **Destination** 和 **DestinationPort** 属性。 返回的任何消息都将附加到连接日志文件条目。

支持以下方法来获取 MQIPT 配置属性的值:

public int getListenerPort()
检索路由侦听器端口-由 **ListenerPort** 属性定义

public String getDestination()
检索目标地址-由 **Destination** 属性定义

public int getDestinationPort()
检索目标侦听器端口地址-由 **DestinationPort** 属性定义

public String getClientIPAddress()
检索发出连接请求的客户机的 IP 地址。

public int getClientPortAddress()
检索客户机发出连接请求使用的端口地址

public int getTimeout()
检索超时值。 MQIPT 将等待安全出口验证请求-由 **SecurityExitTimeout** 属性定义

public int getConnThreadID()
检索处理连接请求的连接线程标识，这对于调试目的很有用

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

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

public boolean getTimedout()
可由安全出口用以确定超时是否已到期

MQIPT 中的 com.ibm.mq.ipt.exit.SecurityExitResponse 类

此类用于将响应从用户定义的安全出口传回 MQIPT，并用以决定此连接请求应该接受还是应该拒绝。

此类型的对象只会在验证方法（请参阅 第 886 页的『MQIPT 中的 com.ibm.mq.ipt.exit.SecurityExit 类』）中创建。 下面是便于创建这些对象的构造函数以及用于各个属性的方法。 请参阅样本安全出口以获取更多信息。

创建缺省 SecurityExitResponse 对象将拒绝连接请求。

构造函数

- **public SecurityExitResponse (String dest, int destPort, int rc, String msg)**

  其中:

  - dest 是新目标
  - destPort 是新目标端口地址

- rc 是原因码
- msg 是将添加到连接日志条目中的消息

- **public SecurityExitResponse (String dest, int destPort, int rc)**
- **public SecurityExitResponse (int rc, String msg)**
- **public SecurityExitResponse (int rc)**

方法

public void setDestination(String dest)
为连接请求设置新目标地址

public void setDestinationPort(int port) throws IPTException
为连接请求设置新目标侦听器端口地址 - 为无效端口地址抛出 IPTException

public void setMessage(String msg)
向连接日志记录添加消息

public void setReasonCode(int rc)
设置连接请求的原因码。

MQIPT 中的安全出口返回码

MQIPT 在许多不同情况下调用安全出口时识别的返回码。

在以下情景中调用安全出口时，MQIPT 识别以下返回码：

返回码	描述	初始化	Validate	刷新
ExitRc.OK	请求已成功完成。	yes	yes	yes
ExitRc.INIT_ERROR	初始化请求失败，将禁用路由。	yes		
ExitRc.REFRESH_ERROR	刷新请求失败。			yes
ExitRc.NOT_AUTHORIZED	验证过程失败，连接请求被拒绝。		yes	
ExitRc.DISABLE_SSL	验证请求成功，目标的连接将不使用 SSL 或 TLS。		yes	

MQIPT 中的端口号控制

在使用 MQIPT 时，可以限制进行出局连接时使用的本地端口号范围。

在路由上设置 **OutgoingPort** 属性以指定初始本地端口号，设置 **MaxConnectionThreads** 来指定要使用的端口数。 例如，如果将 **OutgoingPort** 设置为 1600，将 **MaxConnectionThreads** 设置为 20，那么此路由的本地端口号的范围为 1600 - 1619。

MQIPT 管理员要负责确保这些路由之间的端口号没有冲突。

如果没有定义 **OutgoingPort**，缺省值 0 表示对每个连接使用系统分配的端口号。

在使用 HTTP 时，出局端口数是不使用 HTTP 时的两倍。 在上述示例中，如果路由使用 HTTP，那么端口号范围应该为 1600 - 1639。

请参阅 分配端口号 以获取更多信息。

多宿主系统

在使用多宿主系统时，可通过使用 **LocalAddress** 属性来指定出局连接将绑定到的 IP 地址。 此属性不支持主机名称。

V 9.2.0 对 MQIPT 中存储的密码进行加密

MQIPT 配置可能包含用于访问各种资源的密码，以及用于使用命令端口访问 MQIPT 的密码。 从 IBM MQ 9.2.0 开始，应该通过加密来保护所有这些密码。

关于此任务

在低于 IBM MQ 9.2.0 (或低于 IBM MQ 9.1.4 for Continuous Delivery) 的版本中，只能对 MQIPT 用于访问密钥环或加密硬件密钥库的密码进行加密。 加密密码存储在任何 **SSL*KeyRingPW** 属性引用的文件中。 LDAP 服务器的其他密码和 MQIPT 访问密码以明文形式存储在 `mqipt.conf` 配置文件中。

从 IBM MQ 9.2.0 (或从 IBM MQ 9.1.5 for Continuous Delivery) 开始，应该通过使用 **mqiptPW** 命令对密码进行加密来保护供 MQIPT 使用的所有存储密码。 加密密码作为属性值存储在 `mqipt.conf` 配置文件中。 MQIPT 能够区分加密密码，纯文本密码和属性值中的文件名。 您应该以这种方式对存储供 MQIPT 使用的所有密码进行加密，因为这是最安全的保护方法。

不推荐使用在 MQIPT IBM MQ 9.2.0 之前 (或 IBM MQ 9.1.4 for Continuous Delivery 之前) 使用的密钥库密码加密方法，但仍可用于 IBM MQ 9.2.0 之前可用的配置属性。 为提高对密钥环密码的保护，请使用最新的保护方法对先前已加密的任何密钥环密码进行重新加密。

注: V 9.2.0 V 9.2.0 `mqipt.conf` 配置文件中的 **SSLCommandPortKeyRingPW** 属性和 **mqiptAdmin** 属性文件中的 **SSLClientCAKeyRingPW** 属性不能引用密码文件。 这些属性的值必须设置为 **mqiptPW** 命令输出的加密密码字符串。

如果 MQIPT 配置中存在纯文本或受弱保护的密码，那么将在 MQIPT 启动时或路由启动时发出警告消息。

使用此过程来加密要存储的密码，以供 MQIPT 使用最新的保护方法来使用。 要在 MQIPT IBM MQ 9.2.0 之前 (或者针对 Continuous Delivery 的 IBM MQ 9.1.4 或更低版本) 对密钥环密码进行加密，请执行 第 890 页的『在 IBM MQ 9.2.0 中加密 MQIPT 之前的密钥环密码』中的步骤。

过程

1. 可选： 创建包含密码加密密钥的文件 (如果您还没有密码加密密钥)。

   MQIPT 使用加密密钥对密码进行加密。 您可以在文件中指定自己的加密密钥。 该文件必须至少包含一个字符，并且只能包含一行文本。

   同一密码加密密钥用于对 MQIPT 实例的所有已存储密码进行加密和解密。 因此，对于每个 MQIPT 安装，您只需要单个密码加密密钥文件。

   V 9.2.0 V 9.2.0 您可以使用不同的密码加密密钥来加密存储在 **mqiptAdmin** 属性文件中的密码，而不是用于加密 MQIPT 配置中的密码的加密密钥。

   如果计划将 MQIPT 作为自动启动的服务运行，那么必须创建缺省名称为 `mqipt_cred.key` 的密码加密密钥文件，并将其放在 MQIPT 主目录中。

   您不必指定密码加密密钥，但这样做更安全。 如果未指定您自己的加密密钥，那么将使用缺省加密密钥。

   注: 您必须确保对密码加密密钥文件设置相应的文件许可权，以防止任何未经授权的用户读取加密密钥。 只有运行 **mqiptPW** 命令的用户和运行 MQIPT 的用户需要读取密码加密密钥的权限。
2. 使用 **mqiptPW** 命令对密码进行加密。

   mqiptPW (加密存储的密码)中描述了 **mqiptPW** 命令的语法。

   如果在步骤 第 889 页的『1』中创建了密码加密密钥文件，请使用 **-sf** 参数将文件名指定为 **mqiptPW**。 例如，可以发出以下命令以使用 **-sf** 参数指定的文件中的加密密钥对密码进行加密:

   ```
   mqiptPW -sf /opt/mqipt/mqipt_password.key
   ```
3. 输入提示时要加密的密码。

   加密密码将由 **mqiptPW** 输出。
4. 将加密密码复制到 `mqipt.conf` 配置文件或 **mqiptAdmin** 属性文件中的相应属性。

例如，以下行指定 MQIPT 访问密码的加密密码:

```
AccessPW=<mqiptPW>1!QL+2Jvj/tigKKlD7Nz8Oqw==!AMDBefOUrmPf5i1OuqV5MA==
```

5. 启动 MQIPT。 如果在步骤 第 889 页的『1』 中创建了具有非缺省名称的密码加密密钥文件，请在启动 MQIPT 时指定加密密钥文件的名称。

   您可以在启动 MQIPT 时使用 **-sf** 参数指定密码加密密钥文件的名称。 例如，发出以下命令以使用 **-sf** 参数指定的文件中的加密密钥启动 MQIPT :

   ```
   mqipt /opt/mqipt -sf /opt/mqipt/mqipt_password.key
   ```

   有关在启动 MQIPT 时指定密码加密密钥文件名的其他方法的信息，请参阅 指定密码加密密钥。

   V 9.2.0 V 9.2.0 您可以使用 **mqiptAdmin** 属性文件中的 **PasswordProtectionKeyFile** 属性为 **mqiptAdmin** 命令指定密码加密密钥文件的名称。

在 IBM MQ 9.2.0 中加密 MQIPT 之前的密钥环密码

在 IBM MQ 9.2.0 (以及 Continuous Delivery 的 IBM MQ 9.1.4 和更低版本) 之前，用于访问 MQIPT 所使用的密钥环的加密密码存储在文件中。

关于此任务

请遵循本任务中的过程对密钥环密码进行加密，以便 MQIPT 在 IBM MQ 9.2.0 之前使用，或者在 IBM MQ 9.1.4 或更低版本中对 Continuous Delivery 进行 MQIPT 加密。 从 IBM MQ 9.2.0 中的 MQIPT for Long Term Support 或 IBM MQ 9.1.5 for Continuous Delivery，改为使用 第 889 页的『对 MQIPT 中存储的密码进行加密』 中描述的更安全的保护方法。

过程

1. 使用 **mqiptPW** 命令加密密钥环密码。

   输入以下命令以加密密码:

   ```
   mqiptPW password filename
   ```

   其中:

   密码
   是访问密钥环所需的明文密码

   文件名
   是要创建的密码文件的名称

   mqiptPW (加密存储的密码)中描述了 **mqiptPW** 命令的语法。
2. 将相应的路由属性设置为包含步骤 第 890 页的『1』 中创建的加密密码的文件的名称。

   例如，要指定包含 MQIPT TLS 服务器证书的密钥环的密码文件，请将以下行添加到 `mqipt.conf` 配置文件:

   ```
   SSLServerKeyRingPW=filename
   ```

MQIPT 的其他安全注意事项

MQIPT 有几个其他功能可以帮助设计人员构建安全的解决方案。

- 如果内部网络中的多个客户机都在尝试进行出局连接，那么它们都要经过位于防火墙内的 MQIPT。 防火墙管理员只能授予对 MQIPT 计算机的外部访问权。
- MQIPT 只能连接到在配置文件中已显式配置的队列管理器，除非 MQIPT 正在充当 SOCKS 代理或正在使用安全出口。
- MQIPT 会验证其收到和传输的消息是否有效，以及是否符合 IBM MQ 协议。 这有助于防止 MQIPT 用于 IBM MQ 协议之外的安全攻击。 如果 MQIPT 充当 SSL/TLS 代理，那么当所有 IBM MQ 数据和协议都已加密时， MQIPT 只能保证初始 SSL/TLS 握手。 在此情况下，请使用 Java security manager。

- MQIPT 允许通道出口运行自己的端到端安全协议。
- 通过设置 MaxConnectionThreads 属性可以限制入局连接的总数。 这有助于保护脆弱的内部队列管理器免受拒绝服务攻击。

配置文件

您必须保护 MQIPT 配置文件 mqipt.conf 不被未经授权的用户读取，因为它可能包含敏感信息，例如用于控制对 MQIPT 的远程管理访问的 **AccessPW** 密码。 遵循 第 889 页的『对 MQIPT 中存储的密码进行加密』中的过程来保护配置文件中指定的所有密码。 此外，请确保保护 mqipt.conf 不会受到未经授权的修改。 为 mqipt.conf 设置操作系统文件许可权，以使只有运行 MQIPT 的用户帐户可以读取或更新此文件。

命令端口

MQIPT 命令端口接受通过网络由 **mqiptAdmin** 命令向 MQIPT 的远程实例发出的管理命令。

V 9.2.0 V 9.2.0 从 IBM MQ 9.2.0 开始，可以使用一个不受保护的命令端口和一个受 TLS 保护的命令端口来配置 MQIPT 。 与不安全命令端口的连接未加密。

注: 通过网络发送到不受保护的命令端口的数据 (包括 MQIPT 访问密码) 可以对网络上的其他用户可见。

V 9.2.0 V 9.2.0 在启用不受保护的命令端口或 TLS 命令端口之前，您必须考虑是否需要启用命令端口，并评估允许远程管理 MQIPT 的风险。 从 IBM MQ 9.2.0 开始， **mqiptAdmin** 命令可以管理在与 **mqiptAdmin** 命令相同的用户下运行的 MQIPT 本地实例，而无需使用命令端口。 因此，您可能不需要启用命令端口来管理 MQIPT 的本地实例。

如果启用了不安全或 TLS 命令端口，那么必须防止对命令端口进行未经授权的访问。 例如，在保护对命令端口的访问时，应考虑以下几点:

- 使用防火墙来限制可连接到 MQIPT 命令端口的计算机集。
- V 9.2.0 V 9.2.0 使用 **AccessPW** 和 **RemoteCommandAuthentication** 属性在命令端口上启用认证。 有关启用命令端口认证的更多信息，请参阅 命令端口认证。
- 请考虑使用 **RemoteShutdown** 属性来禁用远程关闭。
- 请考虑使用 **CommandPortListenerAddress** 和 **SSLCommandPortListenerAddress** 属性来配置要在特定网络接口上侦听的命令端口。

有关使用 **mqiptAdmin** 命令管理 MQIPT 的更多信息，请参阅 使用命令行管理 MQIPT。

MQIPT 中的连接日志

MQIPT 提供了连接日志功能，此日志中包含所有成功和失败的连接尝试的列表。

对于 MQIPT 路由接收或建立的每个连接，以及对于 MQIPT 接收的每个管理命令，将一个条目写入连接日志。 通过使用 **ConnectionLog** 和 **MaxLogFileSize** 属性来控制连接日志。 请参阅 MQIPT 全局属性 以获取更多信息。

每次启动 MQIPT 时，都会创建新的连接日志。 为了便于识别，其文件名中包含当前时间戳记，例如:

```
mqiptYYYYMMDDHHmmSS.log
```

其中:

YYYY 表示年份
MM 表示月份
DD 表示日期
HH 表示小时
mm 表示分钟
SS 表示秒

当连接日志达到 **MaxLogFileSize** 属性确定的最大大小时，将会创建 mqipt001.log 备份文件。 最多维护两个备份文件（mqipt001.log 和 mqipt002.log）。

连接日志中的每个条目都会表示连接请求的各个部分。 MQIPT 收到的连接请求和 MQIPT 与目的地地址建立的新连接将会显示为两个日志条目，并且随后在每个连接结束时会再生成两个条目。

下面是成功连接请求的连接日志：

```
Wed May 15 13:13:51 BST 2013 conn accept 127.0.0.1(3842) 127.0.0.1(5000) OK 5000-0
Wed May 15 13:13:51 BST 2013 conn conn   127.0.0.1(3843) localhost(3500) OK 5000-0
Wed May 15 13:13:52 BST 2013 conn close  127.0.0.1(3842) 127.0.0.1(5000) OK 5000-0
Wed May 15 13:13:52 BST 2013 conn close  127.0.0.1(3843) localhost(3500) OK 5000-0
```

下面是失败连接请求的连接日志：

```
Wed May 15 14:56:40 BST 2013 conn accept 127.0.0.1(4138) 127.0.0.1(7000) OK    7000-0
Wed May 15 14:56:40 BST 2013 conn close  127.0.0.1(4138) 127.0.0.1(7000) ERROR 7000-0
 Unrecognized SSL handshake request '54'
```

连接日志条目

每个连接日志条目都包含以下信息：

- 创建条目的时间。
- 条目的类型。 该值可以是下列任一值：

  admin
  管理命令

  连接
  路由连接
- 发生的事件。 该值可以是下列任一值：

  接受
  接收到连接请求

  关闭
  连接已关闭

  连接
  路由目标的连接请求

  德斯皮特
  接收到显示 MQIPT 命令

  nodata
  未从调用者接收到任何数据

  ping
  接收到 ping 请求

  状态
  接收到显示状态命令

  雷夫尔
  接收到刷新命令

  停止
  接收到停止命令
- 源网络地址和端口号。 对于在不使用命令端口的情况下在本地发出的管理命令，将显示值 LOCAL 。
- 目标网络地址和端口号。 对于在未使用命令端口的情况下在本地发出的管理命令，不会显示此命令。
- 完成代码。 值可以是 OK 或 ERROR。
- MQIPT 线程标识。
- 可选错误消息。

使用容器配置 IBM MQ Internet Pass-Thru

您可以在容器中运行 IBM MQ Internet Pass-Thru (MQIPT)。 容器映像使用的基本映像必须使用受支持的 Linux 操作系统。

过程

- mq-container GitHub 存储库中提供了样本 MQIPT Docker 映像。 要构建和运行容器，请遵循 Docker 上的 IBM MQ Internet Pass-Thru 中的指示信息。

下一步做什么

您可以使用 **docker ps** 命令来查看正在运行的容器。 要查看在 Docker 容器中运行的 MQIPT 的控制台输出，请使用 **docker logs ${CONTAINER_ID}** 命令。

V 9.2.3 Multi 配置流式队列

流式队列功能允许您将每条消息的副本放入队列，并传递到第二个队列。 按队列对流式队列进行配置。

本地队列和模型队列有两个与流式队列相关的新属性:

流
这是应将流式消息传递到的队列的名称。 您应该将 **STREAMQ** 属性设置为另一个队列的名称。

对于哪些队列可以配置为将消息流式传输到其他队列，存在一些限制，哪些队列可以设置为流式消息的目标。 有关消息流限制的信息，请参阅 流队列限制。

STRMQOS
这是交付流式消息时要使用的服务质量。

您可以将 **STRMQOS** 属性设置为以下两个值之一:

贝 STEF
尽力而为，这是缺省值。

队列管理器尝试将每条消息的副本传递到 **STREAMQ** 属性中指定的队列。 如果传递流式消息时发生问题，那么这不会影响原始消息的传递。

MUSTDUP
队列管理器尝试将每条消息的副本传递到流式队列。

如果传递流消息时发生问题，那么原始消息不会传递到其队列，并且应用程序会接收 MQCC_FAILED 以及相应的原因码。

请参阅 ALTER 队列， DEFINE 队列和 DISPLAY QUEUE MQSC 命令以及 更改，复制和创建队列，查询队列和查询队列 (响应) PCF 命令以获取更多详细信息。

如果需要每条消息的多个副本，那么可以配置 **STREAMQ** 属性以引用其目标引用了 IBM MQ 主题的 IBM MQ 别名队列的名称。 将消息放入原始队列时，会将消息的副本发布到指定的主题。

当每个预订接收到消息的副本时，必须确保您具有针对主题对象的 API 或受管预订。 传递给订户的消息遵循与其他发布/预订消息相同的规则。 例如，每条消息都具有新的消息标识，并且 MQMD 的上下文字段与原始消息中的上下文字段不同。 有关原始消息与流式消息之间的相似性和差异的更多信息，请参阅 流式消息。

要点: 改变 **STREAMQ** 或 **STRMQOS** 属性是动态的，将影响现有应用程序 (即，它们可能开始流式方法，或者可能开始流式方法到其他队列)。

示例

最佳努力示例

在以下示例中，本地队列为 ORDERS.QUEUE 已更改为将流式消息放入第二个队列 ANALYTICS.QUEUE。 BESTEF 服务质量用于确保在将流式消息放入 ANALYTICS.QUEUE，例如，如果 ANALYTICS.QUEUE 已满，原始消息仍可放入 ORDERS.QUEUE。

这种配置可用于对正在接收的订单执行分析，方法是分析流式消息，同时将原始消息放入订单队列并进行处理。 流式队列功能的一个优点是您可以将流式消息保留在 ANALYTICS.QUEUE 正在等待处理，但不会影响业务满足的实际订单。

```
DEFINE QLOCAL(ANALYTICS.QUEUE)

ALTER QLOCAL(ORDERS.QUEUE) STRMQOS(BESTEF) STREAMQ(ANALYTICS.QUEUE)
```

注: 在示例中， **STRMQOS** 已设置为 BESTEF，尽管您可以将此属性保留在 **ALTER** 命令之外，因为 BESTEF 是缺省服务质量。

必须复制示例

在此示例中，本地队列 PAYMENTS.QUEUE 已更改为将每条消息的流式副本放入另一个本地队列 AUDIT.QUEUE。 重要的是，放入支付队列的每条消息都流到审计队列中，因此会使用 MUSTDUP 服务质量。

如果将流式消息传递到其队列时发生问题，那么也不会传递原始消息，并且应用程序会接收到适当的完成代码和原因码。 如果只涉及单个队列，那么应用程序必须以相同的方式重试该放入操作。

```
DEFINE QLOCAL(AUDIT.QUEUE)

ALTER QLOCAL(PAYMENTS.QUEUE) STRMQOS(MUSTDUP) STREAMQ(AUDIT.QUEUE)
```

注意:

1. 更改原始队列时，不需要流式队列存在。 但是，需要注意的是，由于所使用的服务质量是 MUSTDUP ，因此尝试将消息放入原始队列失败，直到您定义了流式队列。
2. 将队列别名与主题对象的目标配合使用时，如果没有订户，那么流消息的传递仍被视为成功，并且原始消息将传递到其队列。
3. 如果流消息无法传递到其队列，那么队列管理器不会尝试将其传递到死信队列。 但是，如果将流式消息发送到远程队列，那么当该消息通过通道传输到另一个队列管理器时，可以按照现有死信规则将其传递到死信队列。

配置流式队列

无需在流式队列上执行其他配置。 它接收来自将其命名为流式队列的任何队列的消息。 但是，可能需要考虑在流式队列上配置的属性值。

例如，如果原始队列的最大深度为 100,000 ，而流式队列的最大深度仅为 5000 ，那么如果 **STRMQOS** 设置为 BESTEF，那么流式消息可能会丢失; 如果 **STRMQOS** 设置为 MUSTDUP，但发生 MQRC_Q_FULL 错误，那么即使原始队列上留有大量空间，流式消息也会在放置时失败。

根据原始队列的配置方式，考虑可能需要将流式队列上的哪些属性更改为具有相应的值。

相关参考

流式队列

声明

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能。 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默示只能使用 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、程序或服务的操作，由用户自行负责。

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

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：

知识产权许可
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 063-8506 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 可以随时对本出版物中描述的产品和/或程序进行改进和/或更改，而不另行通知。

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

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

本程序的被许可方如果要了解有关程序的信息以达到如下目的：(i) 允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及 (ii) 允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM 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。 所有这些名称都是虚构的，如与实际商业企业所使用的名称和地址有任何雷同，纯属巧合。

版权许可：

本信息包含源语言形式的样本应用程序，用以阐明在不同操作平台上的编程技术。 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 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: