

7.5

配置 *IBM WebSphere MQ*

IBM

注

在使用本资料及其支持的产品之前，请阅读第 377 页的『[声明](#)』中的信息。

此版本适用于 IBM® WebSphere MQ V 7 发行版 5 以及所有后续发行版和修订版，直到在新版本中另有声明为止。

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

© Copyright International Business Machines Corporation 2007, 2024.

内容

配置	5
在 UNIX , Linux 和 Windows 上配置多个安装.....	5
在多安装环境中连接应用程序.....	5
更改主安装.....	12
使队列管理器与安装相关联.....	14
在系统上查找 IBM WebSphere MQ 的安装.....	15
创建和管理队列管理器.....	16
创建缺省队列管理器.....	18
使现有队列管理器成为缺省队列管理器.....	19
创建队列管理器后备份配置文件.....	20
启动队列管理器.....	20
停止队列管理器.....	20
重新启动队列管理器.....	22
删除队列管理器.....	22
使用分布式排队连接应用程序.....	23
IBM WebSphere MQ 分布式消息传递技术.....	23
分布式队列管理简介.....	40
在 UNIX , Linux 和 Windows 上监视和控制通道.....	62
配置服务器与客户机之间的连接.....	83
确定要使用的通信类型.....	84
配置扩展事务客户机.....	86
定义 MQI 通道.....	94
在不同平台上创建服务器连接和客户机连接定义.....	95
在服务器上创建服务器连接和客户机连接定义.....	97
通道出口.....	102
将客户机连接到队列共享组.....	105
使用配置文件配置客户机.....	106
使用环境变量配置客户机.....	119
控制排队的发布/预订.....	126
设置排队的发布/预订消息属性.....	126
正在启动排队的发布/预订.....	127
正在停止已排队的发布/预订.....	128
添加流.....	128
删除流.....	129
添加预订点.....	130
将队列管理器连接到层次结构.....	131
将队列管理器与层次结构断开连接.....	132
配置队列管理器集群.....	132
访问控制和多个集群传输队列.....	134
与分布式排队比较.....	134
集群的组件.....	136
如何选择集群队列管理器以保存完整存储库.....	147
组织集群.....	148
集群命名约定.....	148
重叠集群.....	149
集群提示.....	150
在集群中建立通信.....	151
存储库信息的保留.....	153
管理 IBM WebSphere MQ 集群.....	153
将消息路由到集群或从集群路由消息.....	208
使用集群进行工作负载管理.....	220
集群: 最佳实践.....	233

可用性、恢复和重新启动.....	259
自动客户机重新连接.....	259
控制台消息监视.....	264
高可用性配置.....	264
确保消息不丢失 (日志记录).....	332
备份和复原 IBM WebSphere MQ 队列管理器数据.....	343
更改配置信息.....	347
在 UNIX ， Linux 和 Windows 系统上更改配置信息.....	348
用于更改 IBM WebSphere MQ 配置信息的属性.....	353
更改队列管理器配置信息.....	358
配置 HP Integrity NonStop Server.....	372
网关进程概述.....	373
配置 Gateway 以在 Pathway 下运行.....	373
配置客户机初始化文件.....	374
向通道授予许可权.....	374
声明.....	377
编程接口信息.....	378
商标.....	378

配置

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。

在配置 IBM WebSphere MQ 之前，请阅读 [IBM WebSphere MQ 技术概述](#) 中的 IBM WebSphere MQ 概念。了解如何在 [规划](#) 中规划 IBM WebSphere MQ 环境。

您可以使用多种不同的方法在 IBM WebSphere 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 WebSphere 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 16 页的『创建和管理队列管理器』](#)。

有关如何创建将 IBM WebSphere 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 23 页的『使用分布式排队连接应用程序』](#)。

有关如何使用不同方法将客户机连接到 IBM WebSphere MQ 服务器的指示信息，请参阅 [第 83 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 132 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 WebSphere MQ 或队列管理器的行为。有关更多信息，请参阅 [第 347 页的『更改 IBM WebSphere MQ 和队列管理器配置信息』](#)。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。

相关概念

[WebSphere MQ 技术概述](#)

相关任务

[管理本地 WebSphere MQ 对象](#)

[管理远程 WebSphere MQ 对象](#)

[规划](#)

在 UNIX, Linux, and Windows 上配置多个安装

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。

此信息适用于 UNIX, Linux®, and Windows。

使用以下链接中的信息来配置安装：

- [第 12 页的『更改主安装』](#)
- [第 14 页的『使队列管理器与安装相关联』](#)
- [第 5 页的『在多安装环境中连接应用程序』](#)

相关概念

[多个安装](#)

相关任务

[选择主安装](#)

[选择安装名称](#)

在多安装环境中连接应用程序

在 UNIX, Linux, and Windows 系统上，如果装入了 IBM WebSphere MQ Version 7.1 或更高版本的库，那么 IBM WebSphere MQ 会自动使用相应的库，而无需执行任何进一步的操作。IBM WebSphere MQ 使用与应用程序连接到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 WebSphere MQ 的方式：

链接

编译应用程序时，应用程序将链接到 IBM WebSphere 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 WebSphere 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 7 页的『正在装入 IBM WebSphere MQ Version 7.1 或更高版本的库』。

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 WebSphere 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 WebSphere 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 Version 7.1 或更高版本的库，那么 IBM WebSphere 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 WebSphere MQ 会从与队列管理器关联的安装装入库。

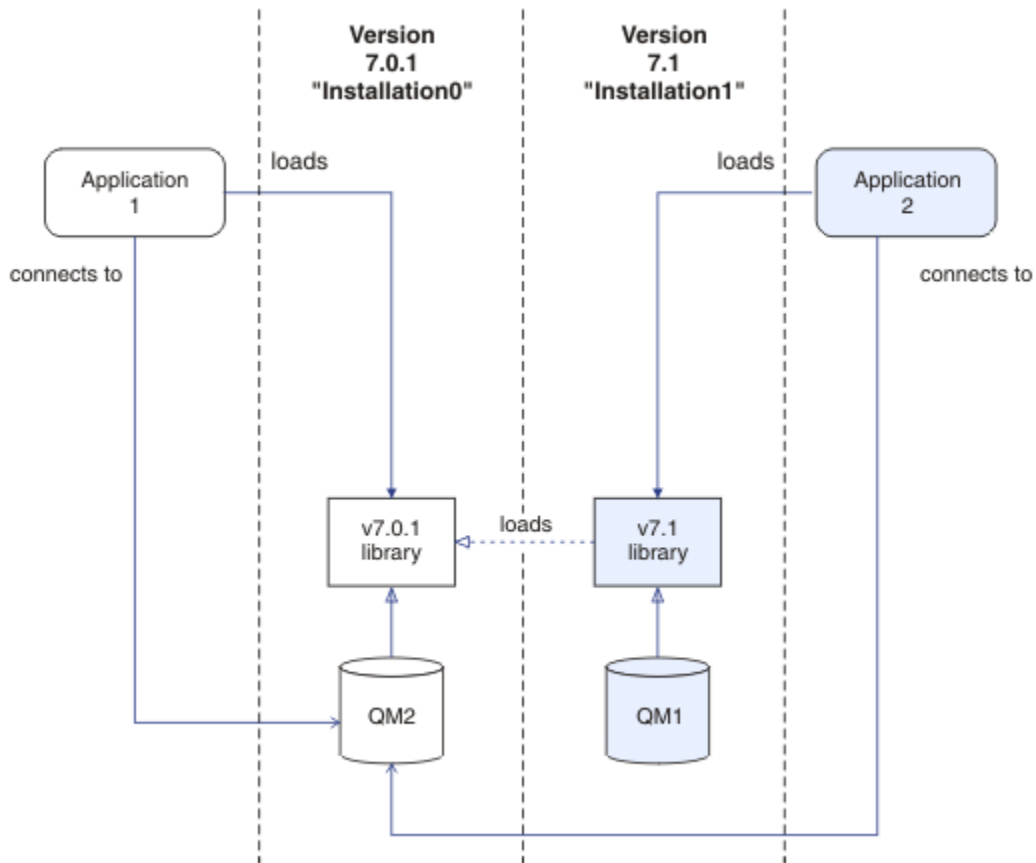


图 1: 在多安装环境中连接应用程序

例如，第 6 页的图 1 显示了具有 V 7.0.1 安装 (Installation0) 和 V 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 V 7.0.1 库。当 application 1 连接到 QM2 时，将使用 V 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，因为版本

7.0.1 库无法装入其他库版本。即，如果直接装入 V 7.0.1 库，那么无法使用与更高版本的 IBM WebSphere MQ 的安装相关联的队列管理器。

Application 2 直接装入 V 7.1 库。当 application 2 连接到 QM2 时，V 7.1 库将装入并使用 V 7.0.1 库。如果 application 2 连接到 QM1，或者如果 QM2 与 Installation1 关联，那么将装入 V 7.1 库，并且应用程序将按预期工作。

在 UNIX, Linux 和 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 Version 7.1 库的更多信息，请参阅第 7 页的『正在装入 IBM WebSphere MQ Version 7.1 或更高版本的库』。

支持和限制

如果找到并装入了以下任何版本 7.1 或更高版本的库，那么 IBM WebSphere MQ 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
- 非受管方式下的 .NET

IBM WebSphere 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 10 页的『使用多个安装的应用程序的限制』。

相关概念

第 14 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 UNIX, 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。

第 10 页的『使用多个安装的应用程序的限制』

在多个安装环境中使用 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 7 页的『正在装入 IBM WebSphere MQ Version 7.1 或更高版本的库』

在决定如何装入 IBM WebSphere 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 WebSphere MQ 的位置以及 IBM WebSphere MQ 的位置是否可能更改。

相关任务

选择主安装

第 12 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。

正在装入 IBM WebSphere MQ Version 7.1 或更高版本的库

在决定如何装入 IBM WebSphere 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 WebSphere MQ 的位置以及 IBM WebSphere MQ 的位置是否可能更改。

IBM WebSphere MQ Version 7.1 库的定位和装入方式取决于您的安装环境：

- 在 UNIX and Linux 系统上，如果 IBM WebSphere MQ Version 7.1 的副本安装在缺省位置，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，那么必须选择版本 7.1 安装作为主安装，或者手动创建符号链接。
- 如果 IBM WebSphere MQ Version 7.1 安装在非缺省位置 (如果还安装了 IBM WebSphere MQ Version 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 WebSphere MQ Version 7.1 或更高版本的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 9 页的『操作系统库装入机制』。

最好是确保队列管理器与操作系统装入的 IBM WebSphere MQ 库相关联。

用于装入 IBM WebSphere MQ 库的方法因平台而异，并且每种方法都具有优点和缺点。

平台	选项	优势	缺陷
UNIX and Linux 系统	<p>设置或更改应用程序的嵌入式运行时搜索路径 (RPath)。</p> <p>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 WebSphere MQ 应用程序。</p>	<ul style="list-style-type: none"> 变化的范围是明确的。 	<ul style="list-style-type: none"> 您必须能够重新编译并链接应用程序。 如果 IBM WebSphere MQ 的位置发生更改，那么必须更改 RPath。
UNIX and Linux 系统	<p>使用带有 -k 或 -l 选项的 <code>setmqenv</code> 或 <code>crtmqenv</code> 来设置 <code>LD_LIBRARY_PATH</code> 环境变量 (AIX 上的 <code>LIBPATH</code>)。</p>	<ul style="list-style-type: none"> 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 WebSphere MQ 的位置发生更改)。 	<ul style="list-style-type: none"> 出于安全原因，<code>setuid</code> 和 <code>setgid</code> 应用程序或以其他方式构建的应用程序可能会忽略 <code>LD_LIBRARY_PATH</code>。 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 <code>LD_LIBRARY_PATH</code> 的其他应用程序的可能影响。 HP-UX: 编译应用程序时使用的选项可能会禁用 <code>LD_LIBRARY_PATH</code>。有关更多信息，请参阅 HP-UX 的运行时链接注意事项。 Linux: 用于构建应用程序的编译器可能会禁用 <code>LD_LIBRARY_PATH</code>。有关更多信息，请参阅 Linux 的运行时链接注意事项。
Windows 系统	<p>使用 <code>setmqenv</code> 或 <code>crtmqenv</code> 设置 <code>PATH</code> 变量。</p>	<ul style="list-style-type: none"> 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 WebSphere MQ 的位置发生更改)。 	<ul style="list-style-type: none"> 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。
UNIX, Linux, and Windows 系统	<p>将主安装设置为 V 7.1 或更高版本的安装。请参阅第 12 页的『更改主安装』。</p> <p>有关主安装的更多信息，请参阅 选择主安装。</p>	<ul style="list-style-type: none"> 不需要对现有应用程序进行任何更改。 如果 IBM WebSphere 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 WebSphere MQ 相似的行为。 	<ul style="list-style-type: none"> 安装 WebSphere MQ V 7.0.1 时，无法将主安装设置为 V 7.1 或更高版本。 UNIX and Linux: 如果 <code>/usr/lib</code> 不在缺省搜索路径中，那么不起作用。

HP-UX 的库装入注意事项

先前版本的 IBM WebSphere MQ 的产品文档中的样本编译命令包含 64 位应用程序的 `-W1, +noenvvar` 链接选项。此选项禁止使用 `LD_LIBRARY_PATH` 来装入共享库。如果您希望应用程序从 RPath 中指定的位置以外的位置装入 IBM WebSphere MQ 库，那么必须更新应用程序。您可以通过在不使用 `-W1, +noenvvar` 链接选项的情况下重新编译和链接，或者使用 `chatr` 命令来更新应用程序。

要了解应用程序当前如何装入库，请参阅 [第 9 页的『操作系统库装入机制』](#)。

Linux 的库装入注意事项

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。

Solaris 的库装入注意事项

先前版本的 IBM WebSphere MQ 的产品文档中的样本编译命令包含 `-lmqmcs -lmqzse` 链接选项。这些库的相应版本现在由 IBM WebSphere MQ 自动装入。如果 IBM WebSphere MQ 安装在非缺省位置，或者如果系统上有多个安装，那么必须更新应用程序。您可以通过在不使用 `-lmqmcs -lmqzse` 链接选项的情况下重新编译和链接来更新应用程序。

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。

在 UNIX 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 `LIBPATH`，HP-UX 上的 `SHLIB_PATH`)。如果设置了此变量，那么它将定义一组要搜索所需的 WebSphere 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 WebSphere 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，但您无法重新编译和链接该应用程序，那么必须在缺省位置安装 IBM WebSphere MQ Version 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 WebSphere 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: `dump`
- HP-UX: `chatr`
- Linux: `readelf`
- Solaris: `elfdump`

相关概念

[第 14 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 UNIX, 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。

[第 10 页的『使用多个安装的应用程序的限制』](#)

在多个安装环境中使用 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 5 页的『在多安装环境中连接应用程序』](#)

在 UNIX, Linux, and Windows 系统上, 如果装入了 IBM WebSphere MQ Version 7.1 或更高版本的库, 那么 IBM WebSphere MQ 会自动使用相应的库, 而无需执行任何进一步的操作。IBM WebSphere MQ 使用与应用程序连接到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 12 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。

使用多个安装的应用程序的限制

在多个安装环境中使用 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 WebSphere 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 Version 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 WebSphere MQ 库, 并且不能链接到 IBM WebSphere MQ 库。有关更多信息, 请参阅 [编写和编译出口和可安装服务](#)。

快速路径

在具有多个安装的服务器上, 使用与 IBM WebSphere MQ Version 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 WebSphere 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 WebSphere 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. 以非快速路径连接与 Version 7.1 或更高版本安装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. 您无法既连接与 Version 7.0.1 安装关联的队列管理器, 又以快速路径连接与 Version 7.1 或更高版本安装关联的队列管理器。

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 WebSphere 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

[MQCONN-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](#)

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，但这些组合件未向 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 WebSphere 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 11 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 12 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：

	使用较低版本的 IBM WebSphere MQ 编译的应用程序	使用更高版本的 IBM WebSphere MQ 编译的应用程序
运行具有更高版本的 IBM WebSphere MQ 主安装的应用程序。(GAC 中的更高版本的组合件):	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 WebSphere MQ 主安装的应用程序。(GAC 中的较低版本组合件):	不需要进行任何更改	在应用程序配置文件中: <ul style="list-style-type: none"> • 使用 <code><bindingRedirect></code> 标记来指示使用 GAC 中的较低版本的组合件
使用更高版本的 IBM WebSphere MQ 非主安装来运行应用程序。(安装文件夹中的更高版本的组合件):	在应用程序配置文件中: <ul style="list-style-type: none"> • 使用 <code><codebase></code> 标记来指向更高版本组合件的位置 • 使用 <code><bindingRedirect></code> 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: <ul style="list-style-type: none"> • 使用 <code><codebase></code> 标记来指向更高版本组合件的位置

表 2: 配置应用程序以使用特定组合件 (继续)

	使用较低版本的 IBM WebSphere MQ 编译的应用程序	使用更高版本的 IBM WebSphere MQ 编译的应用程序
运行具有较低版本的 IBM WebSphere MQ 非主安装的应用程序。(安装文件夹中的较低版本组合件):	在应用程序配置文件中: <ul style="list-style-type: none"> 使用 <code><codebase></code> 标记来指向较低版本组合件的位置 包含标记 <code><publisherpolicy Apply=no></code> 	在应用程序配置文件中: <ul style="list-style-type: none"> 使用 <code><codebase></code> 标记来指向较低版本组合件的位置 使用 <code><bindingRedirect></code> 标记来指示使用较低版本的组合件 包含标记 <code><publisherpolicy Apply=no></code>

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 WebSphere 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 5 页的『在多安装环境中连接应用程序』

在 UNIX, Linux, and Windows 系统上，如果装入了 IBM WebSphere MQ Version 7.1 或更高版本的库，那么 IBM WebSphere MQ 会自动使用相应的库，而无需执行任何进一步的操作。IBM WebSphere MQ 使用与应用程序连接到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](#)

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 UNIX, 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 Version 7.1 或更高版本的安装与 IBM WebSphere MQ Version 7.0.1 的安装共存，那么 IBM WebSphere MQ Version 7.0.1 安装必须是主安装。当安装了 IBM WebSphere MQ Version 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 Version 7.1 或更高版本的安装成为主安装。

在 Windows 上的安装过程中，可以指定安装将作为主安装。在 UNIX 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。

第 13 页的『设置主安装』。

第 13 页的『取消设置主安装』。

设置主安装

过程

要将安装设置为主安装：

1. 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 Version 7.1 或更高版本安装的安装路径。

2. 如果将现有 IBM WebSphere MQ Version 7.1 或更高版本安装设置为主安装，请遵循第 13 页的『取消设置主安装』中的指示信息取消设置。如果 IBM WebSphere MQ Version 7.0.1 安装在系统上，那么无法更改主安装。
3. 作为 UNIX and Linux 系统上的 root 用户或 Windows 系统上的 Administrators 组的成员，请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
```

4. 在 Windows 系统上，重新启动系统。

取消设置主安装

过程

要将安装取消设置为主安装，请执行以下操作：

1. 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 Version 7.1 或更高版本安装的安装路径。

如果 IBM WebSphere MQ Version 7.0.1 是主安装，那么无法取消设置主安装。

2. 作为 UNIX and Linux 系统上的 root 用户或 Windows 系统上的 Administrators 组的成员，请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
```

相关概念

[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 UNIX 和 Linux 上的主安装](#)

相关任务

[卸载，升级和维护主安装](#)

选择安装名称

相关参考

塞特姆因斯特

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 UNIX, 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。

您可以通过以下方式使用 **setmqm** 命令：

- 在等效版本的 WebSphere 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 WebSphere MQ 迁移到较高版本的 WebSphere 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 [迁移和升级 WebSphere MQ](#)。

要将队列管理器与安装相关联，请执行以下操作：

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。

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 V7.1 或更高版本的安装上启动 V7.0.1 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。

HA 环境中的队列管理器关联

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。

与已删除的安装关联的队列管理器

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 15 页的『在系统上查找 IBM WebSphere MQ 的安装』

如果系统上存在多个 IBM WebSphere MQ 安装，那么可以检查安装的版本及其位置。

第 349 页的『IBM WebSphere MQ 配置文件 `mqs.ini`』

IBM WebSphere 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

[塞特姆](#)

[strmqm](#)

[dspmq](#)

[长石](#)

在系统上查找 IBM WebSphere MQ 的安装

如果系统上存在多个 IBM WebSphere 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 WebSphere MQ 安装：

- 使用 **dspmqver** 命令。如果是从 Version 7.0.1 安装发出，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 WebSphere MQ 的安装位置。然后，从 Version 7.1 或更高版本的安装使用 **dspmqver** 命令。以下命令是可用于查询已安装 IBM WebSphere MQ 的位置的命令示例：
 - 在 AIX 系统上，可以使用 **lslpp** 命令：

```
lslpp -R ALL -l mqm.base.runtime
```

- 在 HP-UX 系统上，可以使用 **swlist** 命令：

```
swlist -a location -a revision -l product MQSERIES
```

- 在 Linux 系统上，可以使用 **rpm** 命令：

```
rpm -qa --qf "%{NAME}-%{VERSION}-%{RELEASE}\t%{INSTPREFIXES}\n" | grep MQSeriesRuntime
```

- 在 Solaris 系统上，可以使用 **pkginfo** 和 **pkgparam** 命令：

1. 通过输入以下命令列出已安装的软件包：

```
pkginfo | grep -w mqm
```

2. 对于列出的每个软件包，请输入以下命令：

```
pkgparam pkgname BASEDIR
```

- 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
```

- 在 UNIX and Linux 系统上，发出以下命令以了解 IBM WebSphere MQ 的安装位置：

```
cat /etc/opt/mqm/mqinst.ini
```

然后从 Version 7.1 或更高版本安装使用 **dspmqver** 命令。

- 要在 32 位 Windows 上显示系统上安装的详细信息，请发出以下命令：

```
reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation" /s
```

- 在 64 位 Windows 上，发出以下命令：

```
reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IBM\WebSphere MQ\Installation" /s
```

注：**reg.exe** 命令将仅显示 Version 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](#)

创建和管理队列管理器

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。

创建队列管理器

队列管理器管理与其关联的资源，特别是其拥有的队列。它为消息队列接口 (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 WebSphere MQ 对象。

要创建队列管理器，请使用 IBM WebSphere MQ 控制命令 `crtmqm` (在 `crtmqm` 中描述)。`crtmqm` 命令会自动创建所需的缺省对象和系统对象 (如系统缺省对象中所述)。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。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。

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

打开

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

与队列管理器关联的安装

`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 UNIX，Linux 和 Windows 系统上管理 IBM WebSphere 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，请确保网络中没有其他队列管理器具有相同的名称 *anywhere*。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。

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，您可以省略此项，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 WebSphere 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。这在例子中是做不到的，因为这让它们更难以为继。

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 WebSphere 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。当您在未指定队列管理器名称的情况下调用 `runmqsc` 命令时，处理 `MQSC` 命令的也是队列管理器。

将队列管理器指定为缺省 替换 节点的任何现有缺省队列管理器规范。

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 18 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，使用 `-u` 标志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 [使用队列管理器](#)，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 259 页的『可用性、恢复和重新启动』](#)。

仅适用于 IBM WebSphere 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，那么创建将失败并显示首次故障支持技术 (FFST) 消息。如果该目录不为空，那么将创建新目录。

相关概念

第 5 页的『[配置](#)』

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。

第 20 页的『[创建队列管理器后备份配置文件](#)』

IBM WebSphere MQ 配置信息存储在 Windows UNIX and Linux 系统上的配置文件中。

第 20 页的『[启动队列管理器](#)』

创建队列管理器时，必须将其启动以使其能够处理命令或 MQI 调用。

第 20 页的『[停止队列管理器](#)』

有三种方法可以停止队列管理器：停顿关闭和立即关闭以及先发制人关闭。

第 22 页的『重新启动队列管理器』

您可以使用 **strmqm** 命令来重新启动队列管理器，或者在 IBM WebSphere MQ for Windows 和 IBM WebSphere MQ for Linux (x86 和 x86-64 平台) 系统上，从 IBM WebSphere MQ Explorer 重新启动队列管理器。

第 347 页的『更改 IBM WebSphere MQ 和队列管理器配置信息』

更改 IBM WebSphere MQ 或单个队列管理器的行为以满足安装需要。

系统和缺省对象

相关任务

第 19 页的『使现有队列管理器成为缺省队列管理器』

您可以使现有队列管理器成为缺省队列管理器。您执行此操作的方式取决于您正在使用的平台。

第 22 页的『删除队列管理器』

您可以使用 **dlmqm** 命令或 WebSphere MQ Explorer 来删除队列管理器。

distributed 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 WebSphere 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 16 页的『创建和管理队列管理器』中描述的注意事项。

UNIX 当您使用 **crtmqm** 在 UNIX 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中，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 WebSphere 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 WebSphere 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 [启动队列管理器](#)。

相关概念

[第 20 页的『创建队列管理器后备份配置文件』](#)

IBM WebSphere MQ 配置信息存储在 Windows UNIX and Linux 系统上的配置文件中。

[使用队列管理器](#)

[使用本地队列](#)

相关参考

[系统和缺省对象](#)

使现有队列管理器成为缺省队列管理器

您可以使现有队列管理器成为缺省队列管理器。 您执行此操作的方式取决于您正在使用的平台。

WebSphere MQ for Windows 和 WebSphere MQ for Linux (x86 和 x86-64 平台) 系统

关于此任务

使用以下指示信息使现有队列管理器成为 WebSphere MQ for Windows 和 WebSphere MQ for Linux (x86 和 x86-64 平台) 系统上的缺省队列管理器:

过程

1. 打开 IBM WebSphere MQ Explorer。
2. 右键单击 IBM WebSphere MQ, 然后选择 Properties...。 此时将显示 "WebSphere MQ 的属性" 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”。

UNIX and Linux 系统

关于此任务

创建缺省队列管理器时, 其名称将插入到 WebSphere MQ 配置文件 (mq5.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。
- 如果意外地将另一个队列管理器设置为缺省队列管理器, 并且想要还原为原始缺省队列管理器, 请编辑 mq5.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。

下一步做什么

有关配置文件的信息，请参阅 [第 347 页的『更改 IBM WebSphere MQ 和队列管理器配置信息』](#)。

创建队列管理器后备份配置文件

IBM WebSphere MQ 配置信息存储在 Windows UNIX and Linux 系统上的配置文件中。

在 Windows 和 Linux (x86 和 x86-64) 系统上，使用 IBM WebSphere MQ Explorer 对配置文件进行更改。

在 Windows 系统上，您还可以使用 `amqmdain` 命令对配置文件进行更改。请参阅 [amqmdain](#)

有两种类型的配置文件：

- 安装产品时，将创建 IBM WebSphere 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。

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。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。

有关配置文件的更多信息，请参阅 [第 347 页的『更改 IBM WebSphere MQ 和队列管理器配置信息』](#)。

启动队列管理器

创建队列管理器时，必须将其启动以使其能够处理命令或 MQI 调用。

要启动队列管理器，请使用 `strmqm` 命令。

注：必须从与您正在使用的队列管理器相关联的安装中使用 `strmqm` 命令。您可以使用 `dspmqs -o installation` 命令来查明队列管理器与之关联的安装。

例如，要启动队列管理器 QMB，请输入以下命令：

```
strmqm QMB
```

在 WebSphere MQ for Windows 和 WebSphere MQ for Linux (x86 和 x86-64 平台) 系统上，可以启动队列管理器，如下所示：

1. 打开 IBM WebSphere MQ Explorer。
2. 从 "Navigator" 视图中选择队列管理器。
3. 单击 Start。队列管理器将启动。

如果队列管理器启动时间超过几秒，那么 WebSphere MQ 会间歇性地发出参考消息，详细说明启动进度。

在队列管理器启动并准备好接受连接请求之前，`strmqm` 命令不会返回控制权。

自动启动队列管理器

在 WebSphere MQ for Windows 中，可以在系统使用 IBM WebSphere MQ Explorer 启动时自动启动队列管理器。有关更多信息，请参阅 [使用 IBM WebSphere MQ Explorer 进行管理](#)。

停止队列管理器

有三种方法可以停止队列管理器：停顿关闭和立即关闭以及先发制人关闭。

使用 `endmqm` 命令可停止队列管理器。

注：必须从与您正在使用的队列管理器相关联的安装中使用 `endmqm` 命令。您可以使用 `dspmqs -o installation` 命令来查明队列管理器与之关联的安装。

例如，要停止名为 QMB 的队列管理器，请输入以下命令：

```
endmqm QMB
```

在 WebSphere MQ for Windows 和 WebSphere MQ for Linux (x86 和 x86-64 平台) 系统上, 可以按如下所示停止队列管理器:

1. 打开 IBM WebSphere MQ Explorer。
2. 从 "Navigator" 视图中选择队列管理器。
3. 单击 Stop...。将显示 "结束队列管理器" 面板。
4. 选择 "受控" 或 "立即"。
5. 单击 OK。队列管理器停止。

停顿关闭 (quiesced shutdown)

缺省情况下, **endmqm** 命令执行指定队列管理器的停顿关闭。这可能需要一段时间才能完成。停顿的关闭将等待所有已连接的应用程序断开连接。

使用此类型的关闭来通知应用程序停止。如果您发出:

```
endmqm -c QMB
```

未在所有应用程序停止时告知您。(endmqm -c QMB 命令等同于 endmqm QMB 命令。)

但是, 如果您发出:

```
endmqm -w QMB
```

该命令将一直等到所有应用程序都已停止并且队列管理器已结束。

立即关闭 (immediate shutdown)

对于立即关闭, 将允许任何当前 MQI 调用完成, 但任何新调用都将失败。此类型的关闭不会等待应用程序与队列管理器断开连接。

对于立即关闭, 请输入:

```
endmqm -i QMB
```

抢先关闭 (preemptive shutdown)

注: 除非使用 **endmqm** 命令停止队列管理器的所有其他尝试都失败, 否则请勿使用此方法。此方法可能会对已连接的应用程序产生不可预测的后果。

如果立即关闭不起作用, 那么必须使用 *preemptive* 关闭, 并指定 -p 标志。例如:

```
endmqm -p QMB
```

这将立即停止队列管理器。如果此方法仍然不起作用, 请参阅 [手动停止队列管理器](#) 以获取备用解决方案。

有关 **endmqm** 命令及其选项的详细描述, 请参阅 [endmqm](#)。

如果您在关闭队列管理器时遇到问题

关闭队列管理器时出现问题通常是由应用程序引起的。例如, 当应用程序:

- 不正确检查 MQI 返回码
- 不请求通知停顿
- 终止而不断开与队列管理器的连接 (通过发出 MQDISC 调用)

如果在停止队列管理器时发生问题, 那么可以使用 Ctrl-C 中断 **endmqm** 命令。然后, 您可以发出另一个 **endmqm** 命令, 但这次使用指定所需关闭类型的标志。

重新启动队列管理器

您可以使用 `stmqm` 命令来重新启动队列管理器，或者在 IBM WebSphere MQ for Windows 和 IBM WebSphere MQ for Linux (x86 和 x86-64 平台) 系统上，从 IBM WebSphere MQ Explorer 重新启动队列管理器。

要重新启动队列管理器，请输入：

```
stmqm saturn.queue.manager
```

在 IBM WebSphere MQ for Windows 和 IBM WebSphere MQ for Linux (x86 和 x86-64 平台) 系统上，可以像启动队列管理器一样重新启动队列管理器，如下所示：

1. 打开 IBM WebSphere MQ Explorer。
2. 从 "Navigator" 视图中选择队列管理器。
3. 单击 Start。队列管理器将重新启动。

如果队列管理器重新启动需要超过几秒的时间，那么 IBM WebSphere MQ 会间歇性地发出参考消息，详细说明启动进度。

删除队列管理器

您可以使用 `dltmqm` 命令或 WebSphere MQ Explorer 来删除队列管理器。

开始之前

停止队列管理器。

过程

- 发出以下命令：`dltmqm QMB`

注：必须从与您正在使用的队列管理器相关联的安装中使用 `dltmqm` 命令。您可以使用 `dspmqr -o installation` 命令来查明队列管理器与之关联的安装。

删除队列管理器的步骤

关于此任务

在 WebSphere MQ for Windows 和 WebSphere MQ for Linux (x86 和 x86-64 平台) 系统上，可以删除队列管理器，如下所示：

过程

1. 打开 WebSphere MQ Explorer。
2. 在 "Navigator" 视图中，选择队列管理器。
3. 如果未停止队列管理器，请将其停止。
 - a) 右键单击队列管理器。
 - b) 单击**停止**。
4. 右键单击队列管理器。
5. 单击**删除**。

结果

将删除队列管理器。



注意：

- 删除队列管理器是一个激烈的步骤，因为您还会删除与队列管理器关联的所有资源，包括所有队列及其消息和所有对象定义。如果使用 **dltmqm** 命令，那么不会显示允许您改变主意的提示；当您按 Enter 键时，所有关联资源都将丢失。
- 在 WebSphere MQ for Windows 中，删除队列管理器还会从自动启动列表中除去队列管理器 (如 [第 20 页的『启动队列管理器』](#) 中所述)。命令完成后，将显示 WebSphere MQ queue manager ending 消息；不会告知您已删除队列管理器。
- 删除集群队列管理器不会将其从集群中除去。请参阅 **dltmqm** 的描述中的注释以获取更多信息。


有关 **dltmqm** 命令及其选项的描述，请参阅 [dltmqm](#)。确保只有可信管理员才有使用此命令。(有关安全性的信息，请参阅 [在 Windows UNIX and Linux 系统上设置安全性](#)。)

使用分布式排队连接应用程序

本部分提供有关 WebSphere 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

在阅读本部分之前，了解 [相互通信概念](#) 中引入的通道，队列和其他概念很有用。

使用以下链接中的信息通过分布式排队连接应用程序：

- [第 42 页的『如何将消息发送到另一个队列管理器』](#)
- [第 57 页的『触发通道』](#)
- [第 55 页的『消息的安全性』](#)
- [第 23 页的『IBM WebSphere MQ 分布式消息传递技术』](#)
- [第 40 页的『分布式队列管理简介』](#)
-  [第 62 页的『监视和控制 UNIX, Linux, and Windows 上的通道』](#)

相关概念

[第 83 页的『配置客户机与服务器之间的连接』](#)

要配置 WebSphere MQ MQI 客户机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 347 页的『更改 IBM WebSphere MQ 和队列管理器配置信息』](#)

更改 IBM WebSphere MQ 或单个队列管理器的行为以满足安装需要。

相关任务

[第 132 页的『配置队列管理器集群』](#)

使用本主题中的链接可了解集群如何工作，如何设计集群配置以及如何设置简单集群的示例。

IBM WebSphere MQ 分布式消息传递技术

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：

- [分布式平台的消息通道规划示例](#)

相关概念

[第 23 页的『使用分布式排队连接应用程序』](#)

本部分提供有关 WebSphere 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

[通道](#)

[消息排队简介](#)

[互通的概念](#)

[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 23 页的『使用分布式排队连接应用程序』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 24 页的『消息流控制 (message flow control)』中所述。

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[相互通信概念 \(Concepts of intercommunication\)](#)。

相关概念

第 24 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 24 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 25 页的表 3 说明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。

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	-
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	-
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	-
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 26 页的图 2 显示将消息放置在名为 "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。

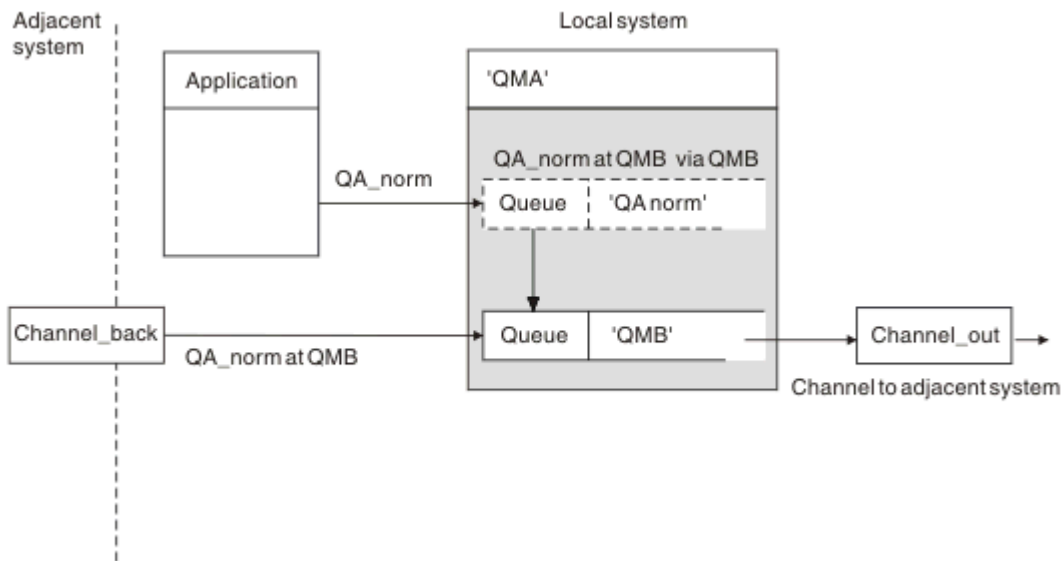


图 2: 远程队列定义用于将队列名解析为传输队列到相邻队列管理器

如果您是 WebSphere MQ 系统管理员，那么必须执行以下操作：

- 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。有关信息，请参阅 [集群](#)。

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。

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。

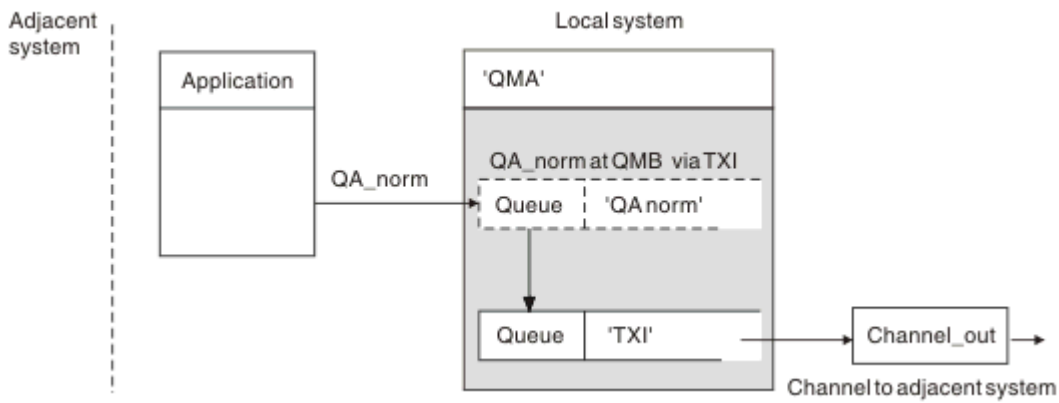


图 3: 远程队列定义允许使用其他传输队列

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，请使用与第 25 页的『将消息放在远程队列上』中的第 26 页的图 2 中所示相同的系统配置。本主题中的第 27 页的图 3 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 27 页的图 3 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 138 页的『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。

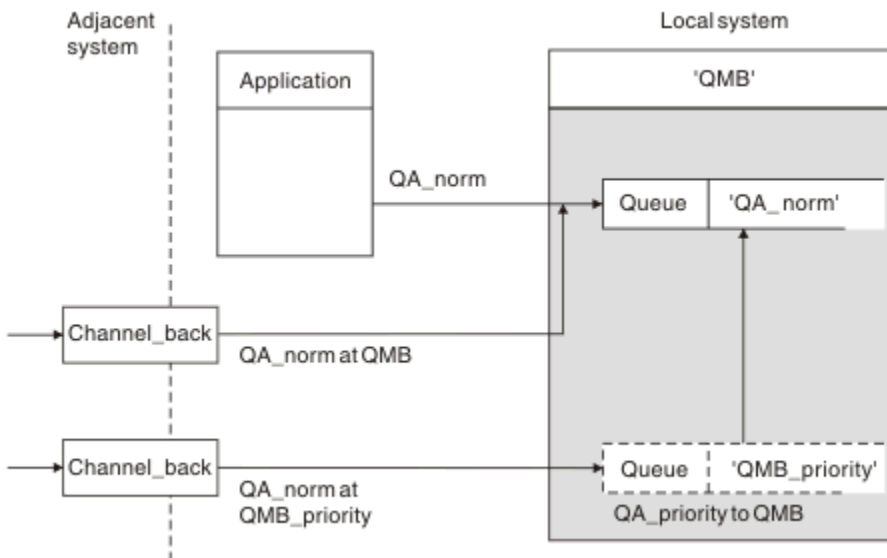


图 4: 直接接收消息并解析别名队列管理器名称

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 27 页的图 4 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。

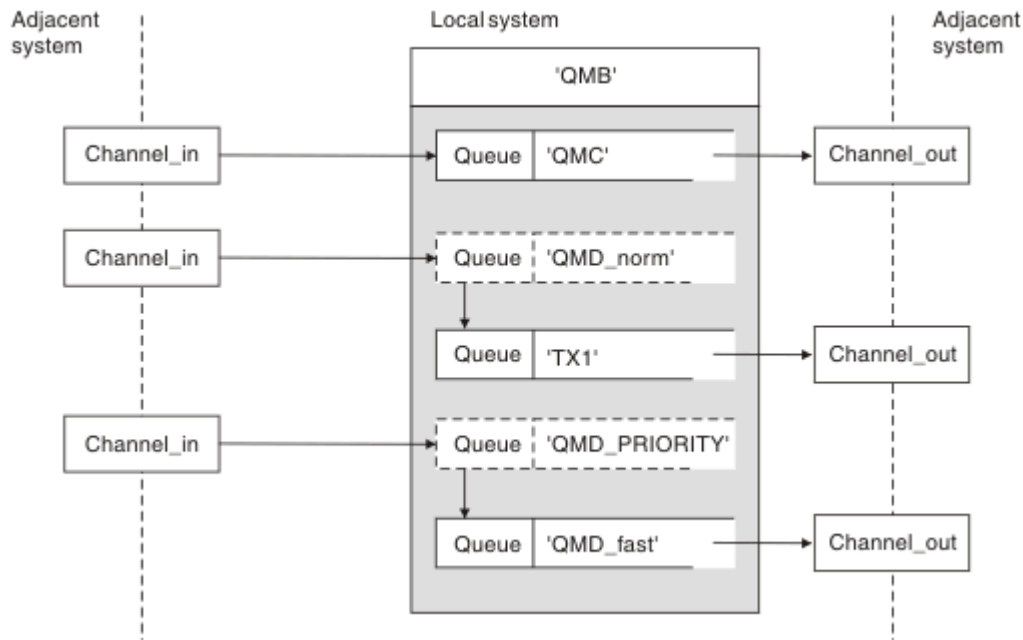


图 5: 通过系统传递消息的三种方法

第 27 页的『接收消息』中的第 27 页的图 4 中显示的方法显示了如何捕获别名流。第 28 页的图 5 说明了通过整合先前描述的技术来构建网络的方式。

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，然后将消息存储在保留队列中，直到调度为止。

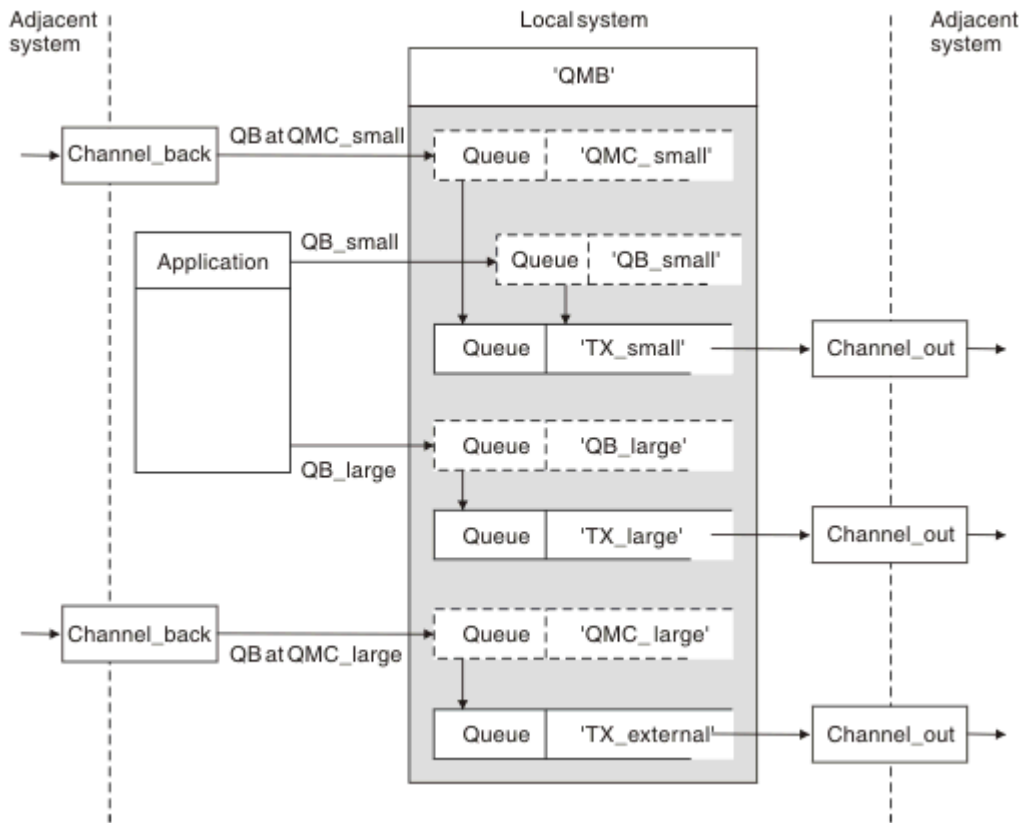


图 6: 分隔消息流

在第 30 页的图 6 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。

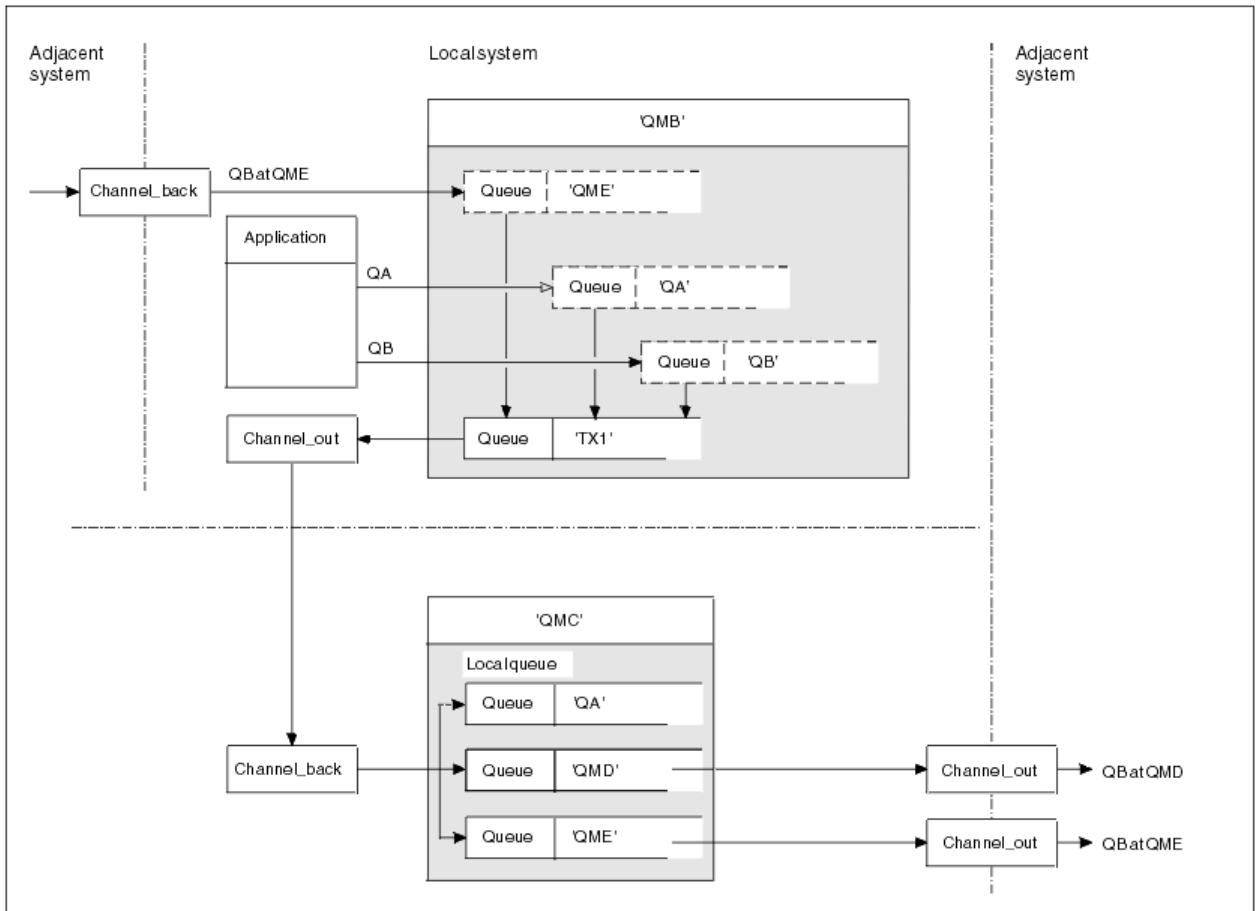


图 7: 将消息流组合到通道

第 31 页的图 7 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。

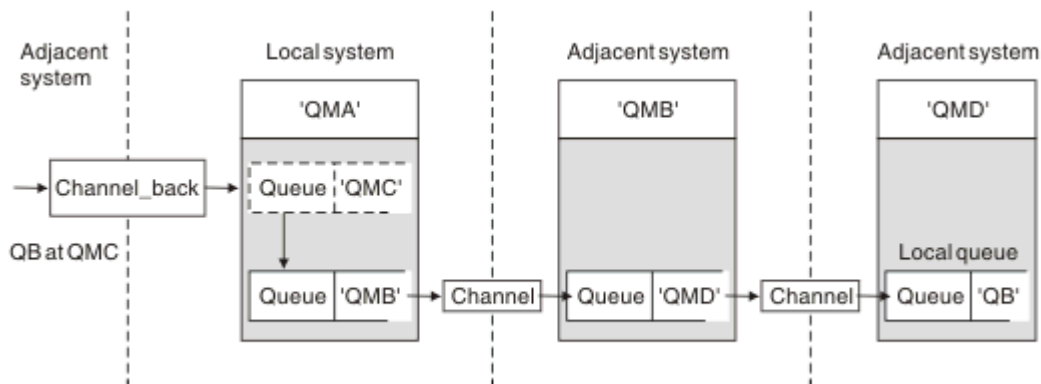


图 8: 将消息流转移到另一个目标

第 32 页的图 8 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 218 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 WebSphere MQ 中，应用程序可以使用单个 MQPUT 调用向多个目标发送消息。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。

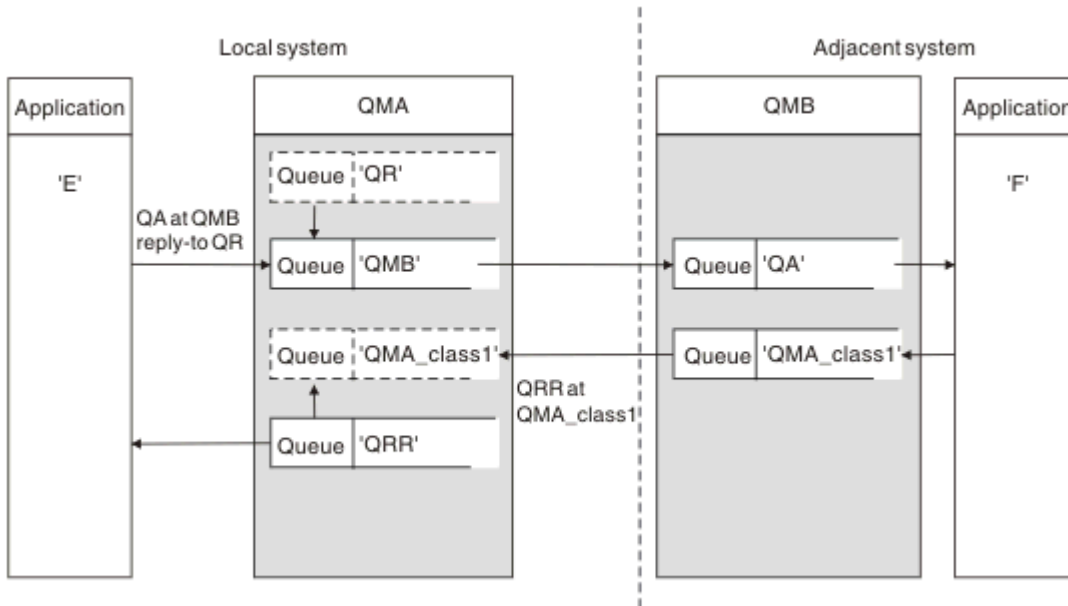


图 9: PUT 调用期间的应答队列名称替换

第 33 页的图 9 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 39 页的表 7 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隔离流。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 24 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 34 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 36 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 36 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 34 页的图 10 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。

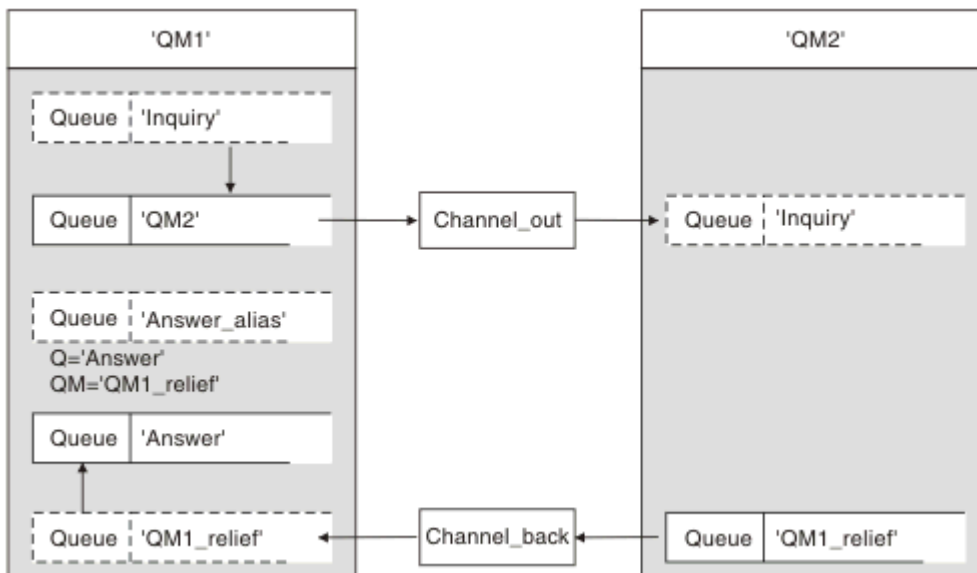


图 10: 应答队列别名示例

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

位于 QM1 的 WebSphere MQ 系统管理员必须确保创建应答队列 "Answer" 以及其他对象。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

位于 QM2 的 WebSphere MQ 系统管理员必须确保本地队列适用于入局消息，并且正确命名的传输队列可用于应答消息。

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。

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。

- 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

- 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。
- 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"。
- 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。
- 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。

- 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。
- 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，请参阅产品文档的 [集群如何工作](#) 部分中的相应主题。

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 38 页的表 5，第 38 页的表 6 和第 39 页的表 7 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。

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

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)
(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

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)	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

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

通道使用序号来确保传递消息，不重复传递消息，并以从传输队列中获取消息的相同顺序存储这些消息。

序号在通道发送端生成，在使用前按 1 递增，这意味着当前序号是发送的最后一消息的编号。可使用 DISPLAY CHSTATUS 显示此信息 (请参阅 MQSC 引用)。对于批处理中传输的最后一消息，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

如果应用程序将消息序列放入同一目标队列，那么在满足以下条件的情况下，单个应用程序可以使用 MQGET 操作按顺序检索这些消息：

- 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。

您可以使用通过控制命令 `dspmqrte` 提供的 WebSphere MQ 显示路由应用程序，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 and 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:

-  [第 62 页的『监视和控制 UNIX, Linux, and Windows 上的通道』](#)

有关消息通道规划示例，请参阅:

-  [分布式平台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 41 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 45 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 55 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 59 页的『初始化和配置文件』

通道初始化数据的处理取决于 WebSphere MQ 平台。

第 60 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 WebSphere MQ 消息时，可能需要进行数据转换。

第 60 页的『编写您自己的消息通道代理程序』

WebSphere 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一个程序。

第 60 页的『分布式队列管理要考虑的其他事项』

准备 WebSphere 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。

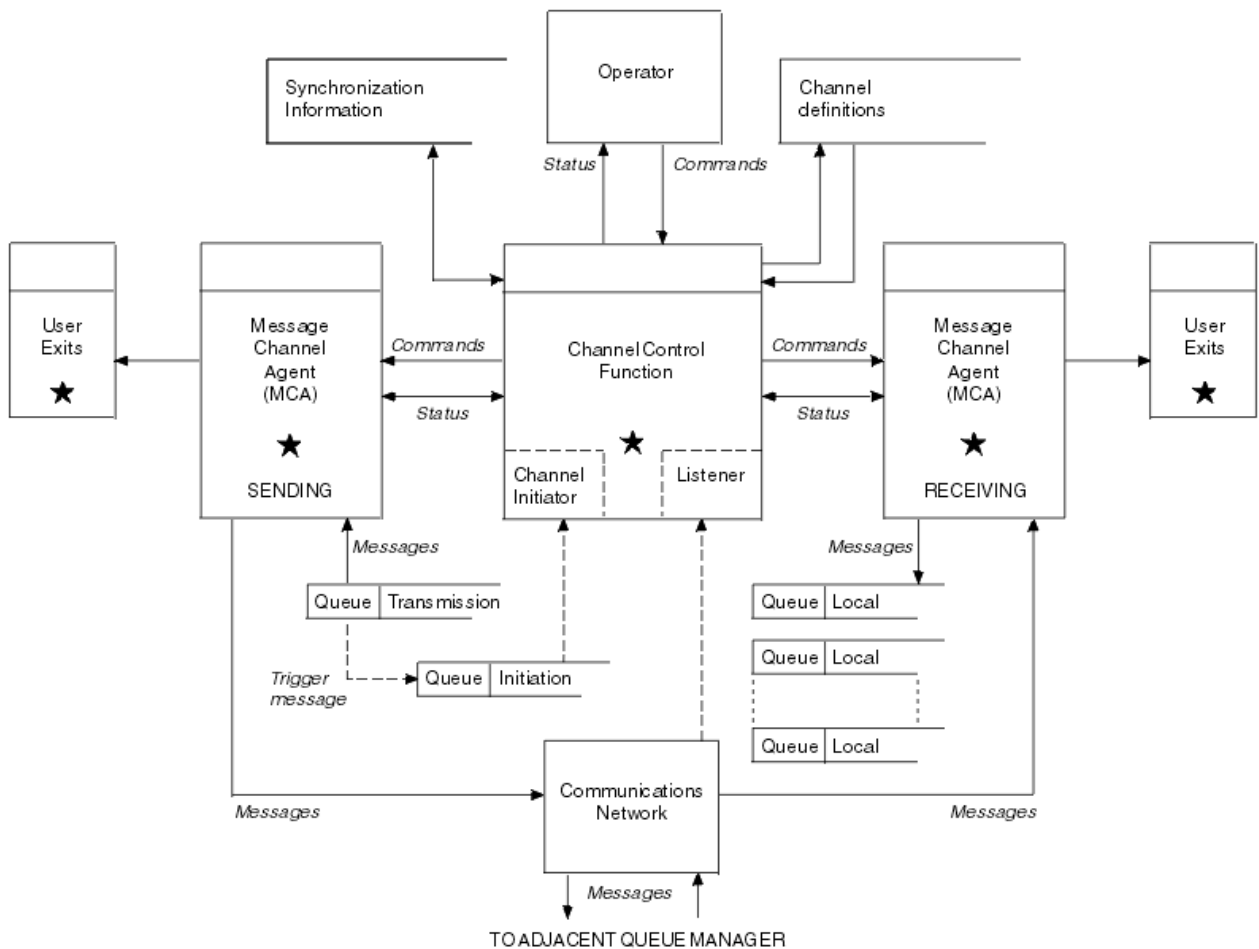


图 11: 分布式队列管理模型

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 WebSphere MQ 安全性授权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:
 - 发送通道
 - 远程队列定义
 - 启动队列 (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 WebSphere MQ 平台：

- 在所有平台上，都可以使用 MQSC 命令中描述的 WebSphere MQ 脚本命令 (MQSC)，[自动化管理任务](#)中描述的可编程命令格式 (PCF) 命令或 WebSphere MQ Explorer。

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 16 页的『创建和管理队列管理器』](#)

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。

[第 23 页的『IBM WebSphere MQ 分布式消息传递技术』](#)

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 40 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 57 页的『触发通道』](#)

WebSphere MQ 提供了一种工具，用于在满足队列上的特定条件时自动启动应用程序。此工具称为触发。

[第 55 页的『消息的安全性』](#)

除了 WebSphere 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。

[第 62 页的『监视和控制 UNIX, 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 WebSphere 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 83 页的『配置客户机与服务器之间的连接』](#)

要配置 WebSphere MQ MQI 客户机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。

相关任务

[第 132 页的『配置队列管理器集群』](#)

使用本主题中的[链接](#)可了解集群如何工作，如何设计集群配置以及如何设置简单集群的示例。

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 WebSphere 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 WebSphere 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。

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。

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

在 z/OS 上是必需的，在其他平台上是可选的。考虑命名启动队列 SYSTEM.CHANNEL.INITQ。在其他平台上。

- 传输队列定义

将 USAGE 属性设置为 XMITQ 的本地队列。

- 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 44 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。

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 WebSphere MQ 命令 (MQSC)，如以下示例中所示：

创建传输队列示例

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

创建远程队列示例

```
DEFINE QREMOTE(PAYROLL) DESC('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。

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 62 页的『[监视和控制 UNIX, Linux, and Windows 上的通道](#)』中详细描述的那些 WebSphere 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

有关定义，控制和监视通道的更多信息，请参阅以下子主题:

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 WebSphere 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 WebSphere 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 23 页的『使用分布式排队连接应用程序』 中描述的方案。

有关定义 MQI 通道的信息，请参阅 第 94 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 53 页的『通道的数量』。

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 47 页的图 12 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 48 页的图 13 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。

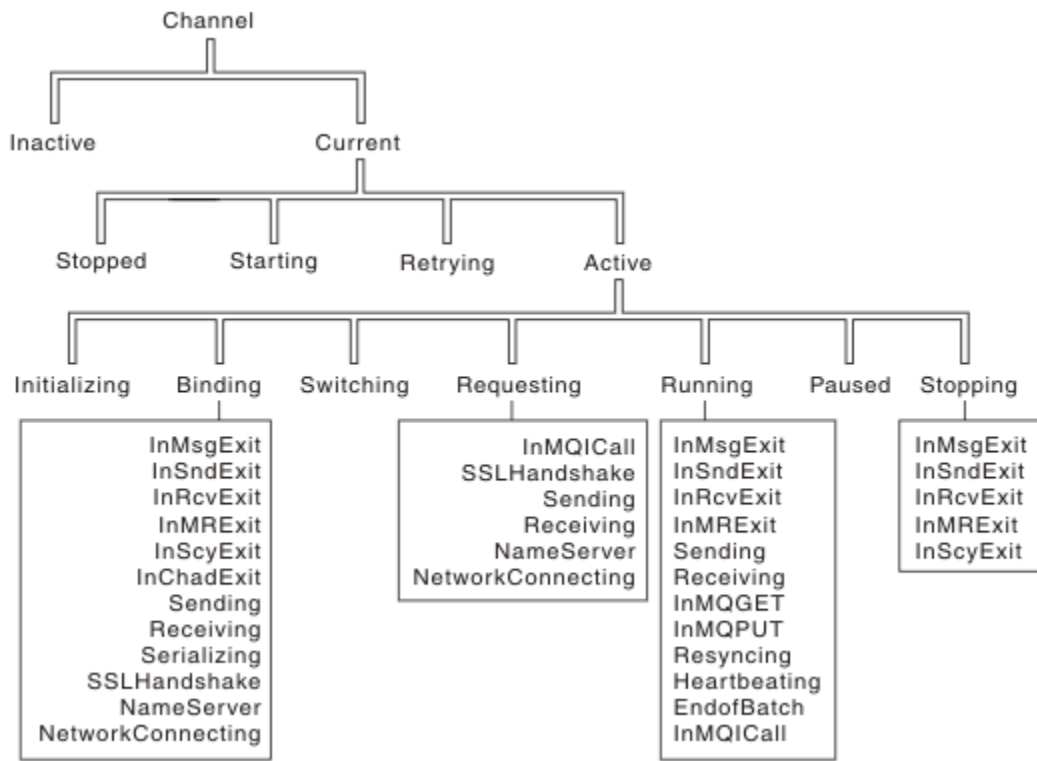


图 12: 通道状态和子状态

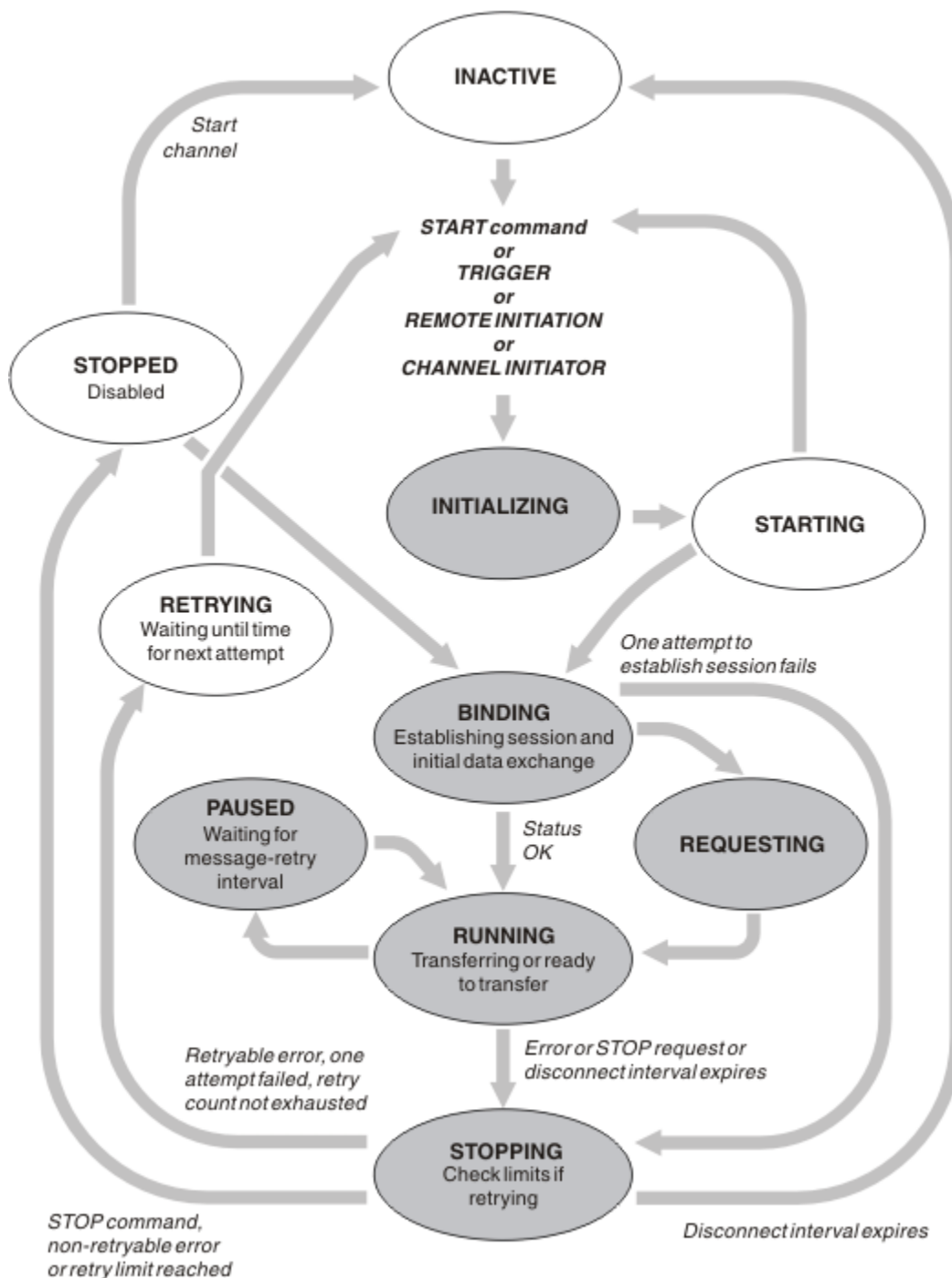


图 13: 通道状态之间的流

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。第 48 页的图 13 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 47 页的图 12 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。

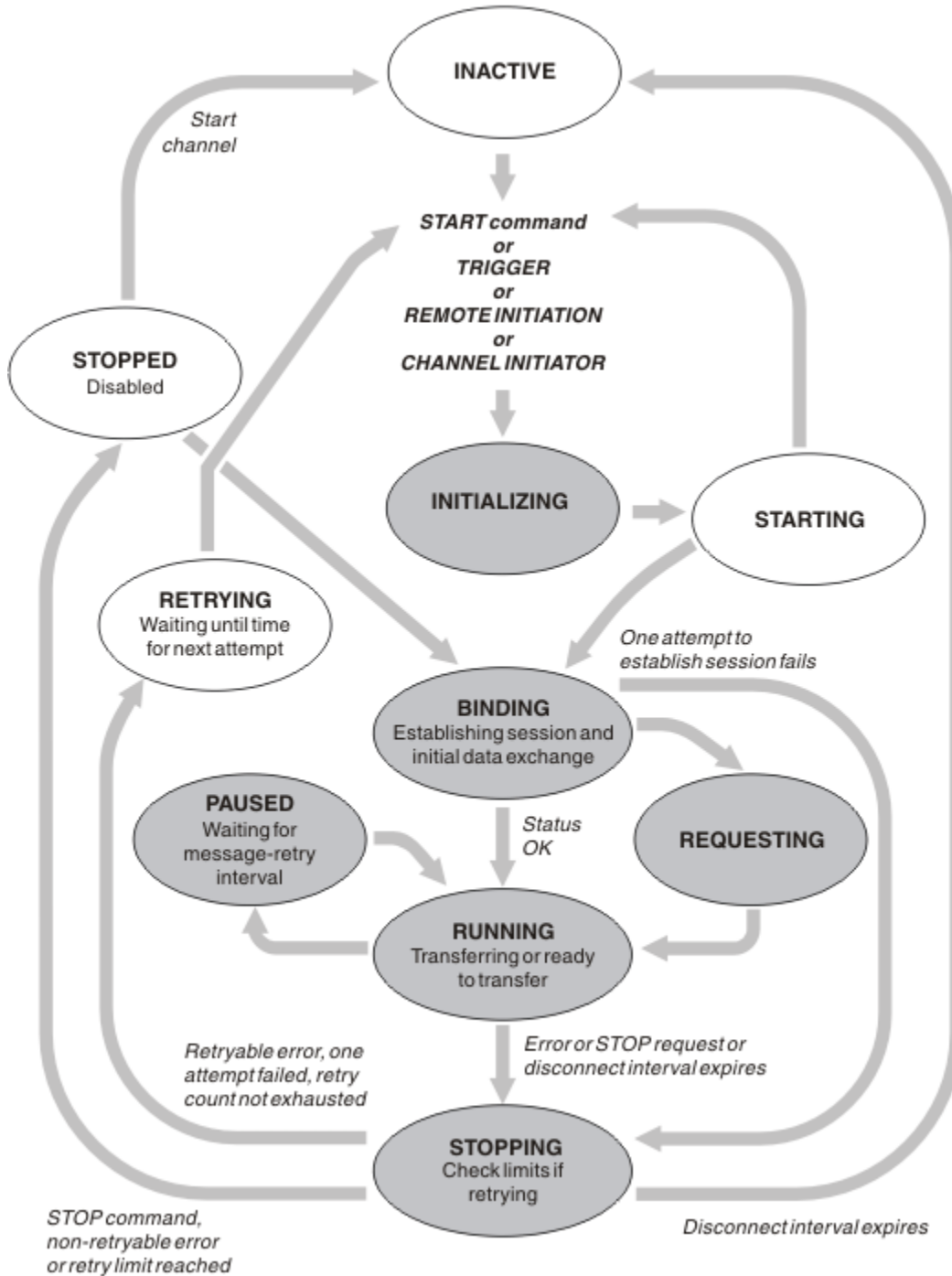


图 14: 通道状态之间的流

注:

1. When a channel is in one of the six states highlighted in 第 49 页的图 14 (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。使用 UNIX and Linux 系统的队列管理器配置文件或 WebSphere MQ Explorer 来指定此值。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 [管理 IBM WebSphere MQ for WebSphere MQ for UNIX and Linux systems, and Windows systems](#)。

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。

每当通道 (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 [管理 IBM WebSphere MQ for WebSphere MQ for UNIX and Linux systems, and Windows systems](#)。

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，然后才能重新启动 (请参阅第 53 页的『重新启动已停止的通道』)。

注: 对于 UNIX, Linux 和 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。

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。但是，如果队列管理器在通道处于 STOPPED 状态时停止，那么将重置 SVRCONN 通道类型的通道状态。

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耗尽队列管理器通道资源 **MAXINST**，并防止单个客户机应用程序耗尽服务器连接通道容量 **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。

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 WebSphere MQ for UNIX, Linux 和 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 WebSphere MQ for UNIX, Linux 和 Windows 系统中, 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. 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。

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 WebSphere MQ V 7 客户机通道中, 脉动信号可以从服务器和客户机端传递。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。

采用 MCA

"采用 MCA" 功能使 IBM WebSphere MQ Explorer 能够取消接收方通道并在其位置启动新的接收方通道。

如果通道发生通信故障, 那么接收方通道可能处于 "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 WebSphere MQ Explorer 能够取消接收方通道并在其位置启动新的接收方通道。

该功能可以使用各种选项进行设置。 **distributed** 对于分布式平台, 请参阅 [管理](#)。

停止和停顿通道

本主题说明如何在断开连接时间间隔到期之前停止和停顿通道。

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 WebSphere MQ 资源管理器

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。

TERMINATE

TERMINATE 选项尝试立即停止通道，并终止该通道的线程或进程。

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。

要重新启动通道，请发出下列其中一个命令：

- START CHANNEL MQSC 命令
- "启动通道 PCF" 命令
- IBM WebSphere MQ 资源管理器

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。

如果队列管理器（在分布式平台上）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。但是，如果队列管理器在通道处于 STOPPED 状态时停止，那么将重置 SVRCONN 通道类型的通道状态。

通道的数量

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 WebSphere 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 WebSphere MQ for IBM i 上，可以使用 STRMQMMQSC 命令或 "使用 MQM 通道状态" CL 命令 WRKMQMCHST 从文件中执行 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)
```

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。

问题确定

问题确定有两个不同的方面：在提交命令时发现的问题以及在通道操作期间发现的问题。

命令验证

在接受命令和面板数据以进行处理之前，这些命令和面板数据必须没有错误。验证发现的任何错误都会立即通过错误消息通知用户。

问题诊断从解释这些错误消息开始，并采取纠正行动。

处理问题

在通道正常运行期间发现的问题将通知系统控制台或系统日志。问题诊断从从日志中收集所有相关信息开始，然后继续进行分析以确定问题。

如果可能，会将确认和错误消息返回到启动命令的终端。

WebSphere MQ 生成记帐和统计数据，可用于确定利用率和性能的趋势。 **distributed** 在分布式平台上，此信息生成为 PCF 记录，请参阅 [结构数据类型](#) 以获取详细信息。

消息和代码

要获取有助于对问题进行主诊断的消息和代码，请参阅 [诊断消息和原因码](#)。

消息的安全性

除了 WebSphere 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 55 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 56 页的图 15](#) 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。）

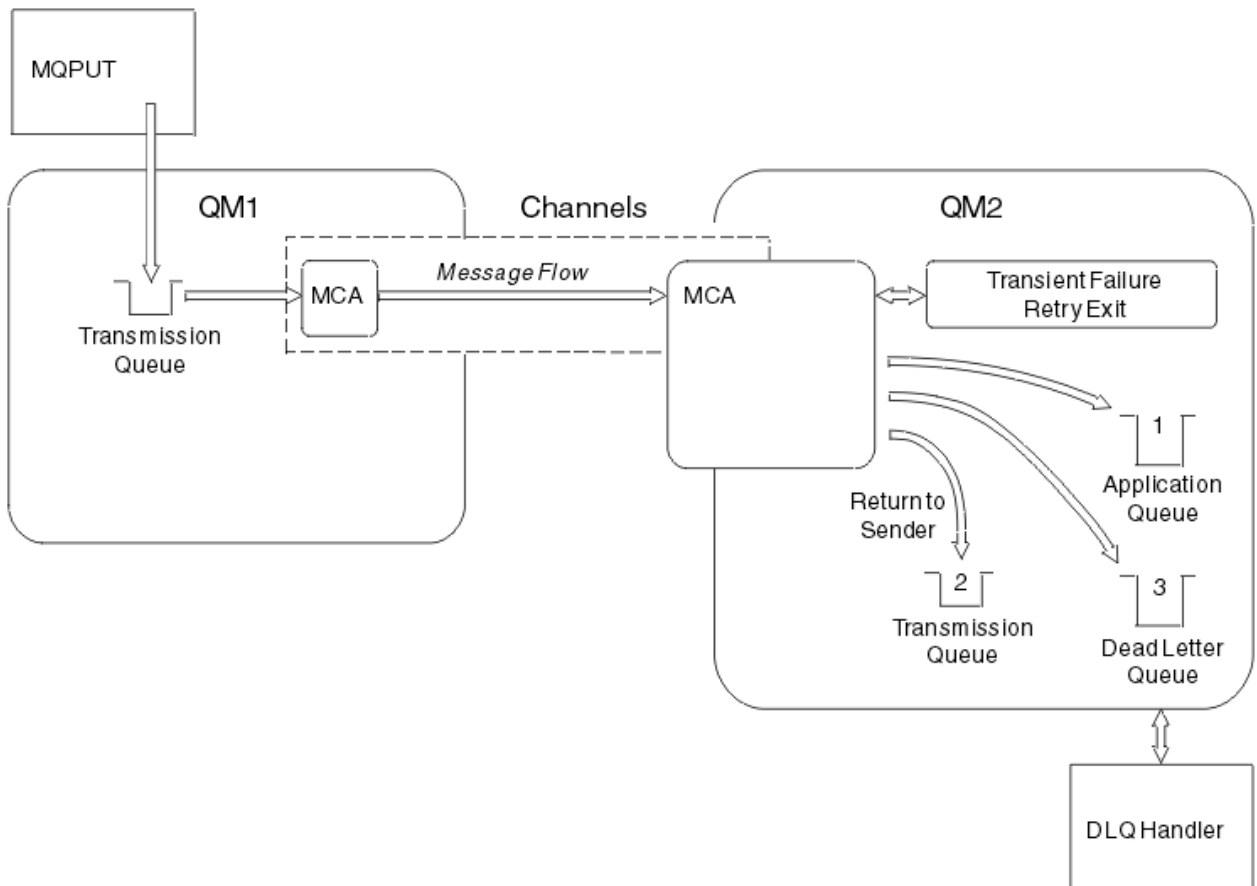


图 15: 无法传递消息时发生的情况

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 [使用 WebSphere MQ dead-letter queue handler for IBM WebSphere MQ for UNIX, Linux 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 Version 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

WebSphere MQ 提供了一种工具，用于在满足队列上的特定条件时自动启动应用程序。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 WebSphere MQ 应用程序](#)。

有关特定于平台的信息，请参阅以下内容：

- 对于 Windows，请参阅 UNIX and Linux 系统，第 58 页的『在 UNIX，Linux 和 Windows 系统上触发通道。』

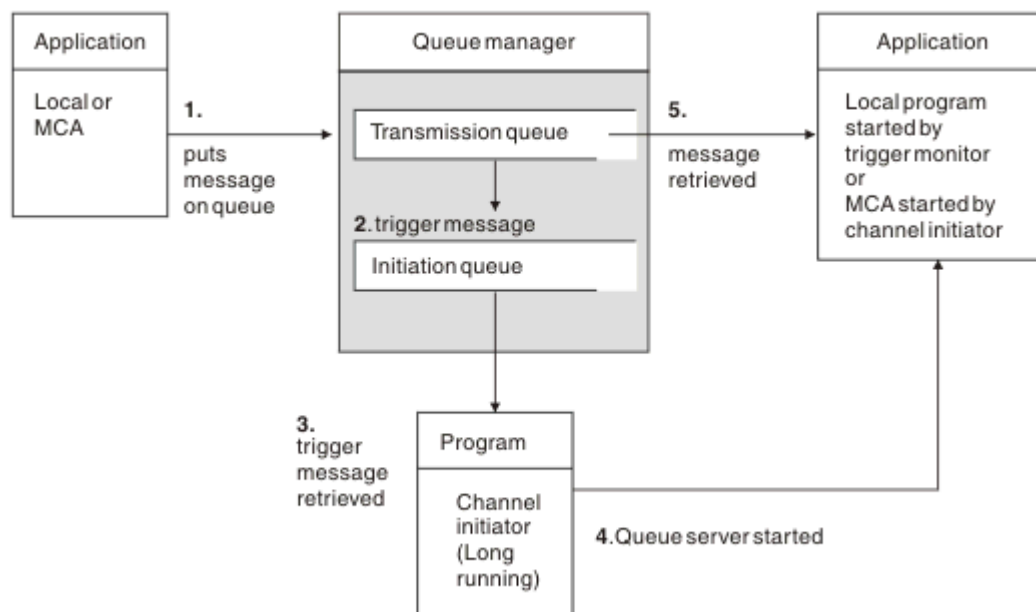


图 16: 触发的概念

第 57 页的图 16 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。
- (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。WebSphere MQ for UNIX，

Linux 和 Windows 系统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。

在 UNIX，Linux 和 Windows 系统上触发通道。

您可以在 WebSphere MQ 中创建进程定义，以定义要触发的进程。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 WebSphere 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。

相关概念

第 58 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。

第 23 页的『[使用分布式排队连接应用程序](#)』

本部分提供有关 WebSphere 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

[UNIX，Linux 和 Windows 系统上的通道程序](#)

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 Linux 和 Windows 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 UNIX 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 [WebSphere 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 WebSphere MQ 平台。

Windows 和 UNIX and Linux 系统

在 WebSphere MQ for Windows, UNIX and Linux 系统中，有配置文件用于保存有关 WebSphere 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。

WebSphere MQ 配置文件

此文件保存与 WebSphere MQ 系统上的所有队列管理器相关的信息。该文件称为 mqs.ini。在 [管理 for WebSphere MQ for Windows, UNIX and Linux systems](#) 中对其进行了全面描述。

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。该文件中包含的信息包括日志配置与 WebSphere 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 UNIX and Linux 系统：

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

以下是 qm.ini 文件的摘录。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将为 200，活动通道的最大数目将为 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。以下示例显示指定通道范围的 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。

对于 Windows 系统:

```
C:\Program Files\IBM\WebSphere MQ\qmgrs\QMNAME\qm.ini
```

有关 qm.ini 文件的更多信息，请参阅 [分布式排队的配置文件节](#)。

消息的数据转换

在不同队列管理器上的队列之间发送 WebSphere MQ 消息时，可能需要进行数据转换。

WebSphere 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

WebSphere 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一个程序。

您可能希望编写自己的 MCA 程序，以使 WebSphere MQ 通过您自己的专有通信协议进行互操作，或者通过 WebSphere MQ 不支持的协议发送消息。(不能编写自己的 MCA 以在另一端与 WebSphere MQ 提供的 MCA 进行互操作。)

如果您决定使用 WebSphere 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 WebSphere 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 WebSphere MQ 所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 WebSphere 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。在 UNIX and Linux 系统上随 WebSphere 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](#)。

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

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 WebSphere MQ 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。

监视和控制 UNIX, 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 WebSphere MQ Explorer，通道定义的文件以及同步信息的存储区来控制通道。

您可以使用以下类型的命令：

IBM WebSphere MQ 命令 (MQSC)

您可以在 Windows UNIX and Linux 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 WebSphere 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 [控制命令](#)。

可编程命令格式命令

有关详细信息，请参阅 [PCF 命令](#)。

IBM WebSphere MQ Explorer

在 UNIX，Linux 和 Windows 系统上，可以使用 IBM WebSphere 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位 (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 63 页的表 8](#)。

相关概念

[第 65 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 70 页的『为 Windows 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，需要定义连接并使其可用。本节说明如何使用 WebSphere MQ for Windows 系统的四种通信形式之一来执行此操作。

[第 78 页的『在 UNIX and Linux 系统上设置通信』](#)

DQM 是 IBM WebSphere 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，由系统操作员控制。由分布式排队管理持有的通道定义使用这些连接。

相关参考

[UNIX，Linux 和 Windows 系统上的通道程序](#)

[分布式平台的消息通道规划示例](#)

[示例配置信息](#)

[通道属性](#)

设置和控制通道所需的功能

可能需要许多 IBM WebSphere 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 WebSphere 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 WebSphere 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 65 页的『对象入门』](#)
- [第 65 页的『创建关联对象』](#)
- [第 65 页的『创建缺省对象』](#)
- [第 65 页的『创建通道』](#)
- [第 66 页的『显示通道』](#)
- [第 66 页的『显示通道状态』](#)
- [第 67 页的『使用 Ping 检查链接』](#)
- [第 67 页的『启动通道』](#)
- [第 68 页的『停止通道』](#)
- [第 69 页的『重命名通道』](#)
- [第 69 页的『重置通道』](#)
- [第 70 页的『在通道上解析不确定消息』](#)

第 63 页的表 8 显示了您可能需要的 IBM WebSphere MQ 函数的完整列表。

表 8: UNIX, Linux, and Windows 系统中所需的功能			
函数	控制命令	MQSC	WebSphere MQ Explorer 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	dlmqm		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 8: UNIX, Linux, and Windows 系统中所需的功能 (继续)

函数	控制命令	MQSC	WebSphere MQ Explorer 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 (仅限 Windows 系统, AIX, HP-UX 和 Solaris)		否

注:

1. 当队列管理器启动时, 可能会自动启动侦听器。

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 WebSphere MQ 命令 (MQSC) 或 IBM WebSphere 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:

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 [分布式平台的消息通道规划示例](#)。

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 [控制命令](#)。

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。

创建通道

创建 **两个** 通道定义, 一个在连接的每一端。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 **同一** 通道名称来定义两端。两端必须具有 **兼容** 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') +
CONNNAME(QM2) TRPTYPE(TCP) XMITQ(QM2) CONVERT(YES)
```

在 MQSC 的所有示例中，该命令显示为显示在命令文件中，并显示为在 Windows，UNIX 或 Linux 系统中输入。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。在 Windows 上，使用 Ctrl-z 在命令行结束条目。在 UNIX and Linux 系统上，使用 Ctrl-d。或者，在 UNIX，Linux 或 Windows 系统上，使用 **end** 命令。

显示通道

使用 MQSC 命令 DISPLAY CHANNEL 来显示通道的属性。

缺省情况下，如果未请求任何特定属性并且指定的通道名称不是通用的，那么将采用 DISPLAY CHANNEL 命令的 ALL 参数。

这些属性在 [通道属性](#) 中进行了描述。

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
```

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

- 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。

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。

它仅可从发送方和服务器通道获取。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 WebSphere MQ 对其进行加密。然后在流经对话之前对其进行解密。

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：

- 对于 UNIX and Linux 系统上的 TCP，可以将 inetd 配置为启动通道。inetd 作为单独的进程启动。
- 对于 UNIX and Linux 系统中的 LU 6.2，配置 SNA 产品以启动 LU 6.2 响应程序进程。
- 对于 Windows 系统中的 LU 6.2，使用 SNA Server 可以使用 TpStart (随 SNA Server 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

`MQ_INSTALLATION_PATH\mqgrs\qmname\errors\AMQERR01.LOG` (对于名为 qmname 的每个队列管理器)

`MQ_INSTALLATION_PATH\mqgrs\@SYSTEM\errors\AMQERR01.LOG` (对于一般错误)

`MQ_INSTALLATION_PATH` 表示安装 WebSphere MQ 的高级目录。

注：在 Windows 系统上，您还会在 Windows 系统应用程序事件日志中收到一条消息。

UNIX and Linux 系统

`/var/mqm/mqgrs/qmname/errors/AMQERR01.LOG` (对于名为 qmname 的每个队列管理器)

`/var/mqm/mqgrs/@SYSTEM/errors/AMQERR01.LOG` (对于一般错误)

在 Windows UNIX and Linux 系统上，使用 `runmqclsr` 命令启动 WebSphere 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。

```
runmqclsr -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。

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通道的信息，请参阅第 53 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。

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，一个通道具有 **两个** 通道定义，每端各有一个。确保同时在两端重命名通道。

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 65 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 45 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 23 页的『使用分布式排队连接应用程序』](#)

本部分提供有关 WebSphere 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](#)

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或服务器
- 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 65 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 45 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 23 页的『使用分布式排队连接应用程序』](#)

本部分提供有关 WebSphere 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](#)

为 Windows 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，需要定义连接并使其可用。本节说明如何使用 WebSphere MQ for Windows 系统的四种通信形式之一来执行此操作。

您可能会发现，请参阅 [示例配置- IBM WebSphere MQ for Windows](#)。

对于 UNIX and Linux 系统，请参阅 [第 78 页的『在 UNIX and Linux 系统上设置通信』](#)。

决定连接

从 WebSphere MQ for Windows 系统的以下四种通信形式中进行选择：

- [第 71 页的『在 Windows 上定义 TCP 连接』](#)
- [第 72 页的『在 Windows 上定义 LU 6.2 连接』](#)
- [第 73 页的『在 Windows 上定义 NetBIOS 连接』](#)
- [第 76 页的『在 Windows 上定义 SPX 连接』](#) (仅限 Windows XP 和 Windows 2003 Server)

每个通道定义都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一个或多个协议。

对于 WebSphere MQ 客户机，具有使用不同传输协议的备用通道可能很有用。有关 WebSphere MQ 客户机的更多信息，请参阅 [客户机概述](#)。

相关概念

[第 23 页的『使用分布式排队连接应用程序』](#)

本部分提供有关 WebSphere 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 62 页的『[监视和控制 UNIX, 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 WebSphere 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 83 页的『[配置客户机与服务器之间的连接](#)』

要配置 WebSphere MQ MQI 客户机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。

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。

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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere MQ 侦听器。

要运行 IBM WebSphere MQ 随附的侦听器 (将新通道作为线程启动)，请使用 `runmqtsr` 命令。

使用 `runmqtsr` 命令的基本示例：

```
runmqtsr -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 WebSphere 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 WebSphere MQ 的号码。

为了获得最佳性能, 请将 IBM WebSphere MQ 侦听器作为可信应用程序运行, 如第 61 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 51 页的『正在检查通道的另一端是否仍然可用』。

在 Windows 上, Windows KeepAlive 时间选项的 HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters 注册表值控制在检查连接之前经过的时间间隔。缺省值为两小时。

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。

远程平台	TPNAME	主题方案路径
没有 CICS 的 z/OS 或 MVS/ESA	与有关远程队列管理器的相应辅助信息中的相同。	-
使用 CICS 的 z/OS 或 MVS/ESA	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。	-
UNIX 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。	<code>MQ_INSTALLATION_PATH/bin/amqcrs6a</code>
Windows	如 Windows "运行侦听器" 命令或 Windows 上使用 TpSetup 定义的可调用事务程序中所指定。	<code>MQ_INSTALLATION_PATH\bin\amqcrs6a</code>

`MQ_INSTALLATION_PATH` 表示安装了 WebSphere MQ 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 73 页的『在 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 73 页的『在 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。

在 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      OS2R0G2
Partner TP Name     recv
Mode Name           #INTER
```

在 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, 您可以在 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 是在另一个 (发送) 端指定为 "TpName 以在远程端启动" 的 TpName。方括号中的最后一部分是可选的, 缺省队列管理器不需要此部分。

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 WebSphere 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 WebSphere 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。请参阅 [微软公司 SNA Server APPC 程序员指南](#) 或 [微软公司 SNA Server CPI-C 程序员指南](#) 以获取更多信息。

如果未指定队列管理器名称, 那么将采用缺省队列管理器。

在 Windows 上定义 NetBIOS 连接

WebSphere MQ 在与其他 WebSphere 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 WebSphere 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 WebSphere 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 74 页的『定义 IBM WebSphere MQ 本地 NetBIOS 名称』](#)

可以通过三种方式指定 IBM WebSphere MQ 通道进程使用的本地 NetBIOS 名称。

[第 74 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 75 页的『建立 LAN 适配器号』](#)

要使通道在 NetBIOS 之间成功工作，每个端的适配器支持必须兼容。IBM WebSphere MQ 允许您通过在 qm.ini 文件的 NETBIOS 节中使用 AdapterNum 值并通过在 runmqclsr 命令上指定 -a 参数来控制 LAN 适配器 (LANA) 编号的选择。

[第 75 页的『启动 NetBIOS 连接』](#)

定义启动连接所需的步骤。

[第 75 页的『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。

定义 IBM WebSphere MQ 本地 NetBIOS 名称

可以通过三种方式指定 IBM WebSphere MQ 通道进程使用的本地 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 WebSphere 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。如果在建立 NetBIOS 通道时迁到问题，并且队列管理器错误日志中有错误消息显示 NetBIOS 返回码 X'15'，请查看您使用的 NetBIOS 名称。
2. 在 Windows 上，无法将机器名称用作 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。

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
```

如果未在命令中指定 `-m` 操作数，那么值仅适用于缺省队列管理器。

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
```

建立 LAN 适配器号

要使通道在 NetBIOS 之间成功工作，每个端的适配器支持必须兼容。IBM WebSphere MQ 允许您通过在 `qm.ini` 文件的 NETBIOS 节中使用 `AdapterNum` 值并通过在 `runmqcls` 命令上指定 `-a` 参数来控制 LAN 适配器 (LANA) 编号的选择。

IBM WebSphere 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 适配器号的更多信息，请参阅 Microsoft 知识库文章 [138037 HOWTO: Use LANA Numbers in a 32-bit Environment](#)。

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。

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
```

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 WebSphere MQ 侦听器程序以建立该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 WebSphere MQ 侦听器作为可信应用程序运行，如第 61 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 WebSphere 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。

在 Windows 上定义 SPX 连接

SPX 连接仅适用于运行 Windows XP 和 Windows 2003 Server 的客户机和服务器。

发送端的通道定义指定目标的地址。必须在接收端运行侦听器程序。

相关概念

第 76 页的『在 SPX 上发送结束』

如果目标机器是远程机器，请在通道定义的“连接名称”字段中指定目标机器的 SPX 地址。

第 77 页的『在 SPX 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。

第 78 页的『IPX/SPX 参数』

在大多数情况下，IPX/SPX 参数的缺省设置将适合您的需要。但是，您可能需要在环境中修改其中部分内容，以调整其用于 WebSphere MQ。

在 SPX 上发送结束

如果目标机器是远程机器，请在通道定义的“连接名称”字段中指定目标机器的 SPX 地址。

SPX 地址按以下格式指定：

```
network.node(socket)
```

其中：

network

是远程机器所在网络的 4 字节网络地址，

node

是 6 字节节点地址，这是远程机器中 LAN 适配器的 LAN 地址

socket

是远程机器侦听的 2 字节套接字号。

如果本地机器和远程机器位于同一网络上，那么无需指定网络地址。如果远程端正在侦听缺省套接字 (5E86)，那么无需指定套接字。

MQSC 命令的 CONNAME 参数中指定的完全指定 SPX 地址的示例为：

```
CONNAME('00000001.08005A7161E5(5E87)')
```

在缺省情况下，如果机器都位于同一网络上，那么这将变为：


```
CONNNAME (08005A7161E5)
```

可以通过在队列管理器配置文件 (qm.ini) 中指定缺省套接字号来对其进行更改:

```
SPX:  
Socket=5E87
```

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 SPX 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通道。

使用 WebSphere MQ 侦听器。

使用 SPX 侦听器储备选项

在 SPX 上接收时, 将设置最大未完成的连接请求数。这可被视为在 SPX 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 77 页的表 10](#) 中。

平台	缺省侦听器任务列表值
Windows 服务器	5
Windows 工作站	5

如果积压达到 [第 77 页的表 10](#) 中的值, 那么尝试使用 MQCONN 或 MQCONNX 连接到队列管理器时, 将收到原因码 MQRC_Q_MGR_NOT_AVAILABLE。如果发生这种情况, 可以再次尝试连接。

但是, 为了避免此错误, 您可以在 qm.ini 文件或 Windows 注册表中添加一个条目:

```
SPX:  
ListenerBacklog = n
```

这将覆盖 SPX 侦听器的缺省最大未完成请求数 (请参阅 [第 77 页的表 10](#))。

注: 某些操作系统支持大于缺省值的值。如果需要, 可以使用此选项来避免达到连接限制。

要在打开 **储备** 选项的情况下运行侦听器, 请执行以下任一操作:

- 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](#)。

使用 WebSphere MQ 侦听器

要运行随 WebSphere 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 spx [-m QMNAME] [-x 5E87]
```

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 (5E86), 那么不需要套接字号。

为了获得最佳性能, 请将 WebSphere MQ 侦听器作为可信应用程序运行, 如 [第 61 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更多信息。

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 WebSphere 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。

IPX/SPX 参数

在大多数情况下, IPX/SPX 参数的缺省设置将适合您的需要。但是, 您可能需要在环境中修改其中部分内容, 以调整其用于 WebSphere MQ。

根据 SPX 通信支持的平台和提供商, 实际参数和更改参数的方法各不相同。此示例部分描述了其中一些参数, 尤其是那些可能影响 WebSphere MQ 通道和客户机连接的操作的参数。

Windows 系统

有关 NWLink IPX 和 SPX 参数的使用和设置的完整详细信息, 请参阅 Microsoft 文档。IPX/SPX 参数位于注册表中的以下路径中:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service\NWLinkSPX\Parameters  
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service\NWLinkIPX\Parameters
```

在 UNIX and Linux 系统上设置通信

DQM 是 IBM WebSphere 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, 由系统操作员控制。由分布式排队管理持有的通道定义使用这些连接。

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。您可能还会发现, 请参阅以下部分很有用:

- [示例配置- IBM WebSphere MQ for AIX](#)
- [示例配置- IBM WebSphere MQ for HP-UX](#)
- [示例配置- IBM WebSphere MQ for Solaris](#)
- [示例配置- IBM WebSphere MQ for Linux](#)

对于 Windows, 请参阅第 70 页的『为 Windows 设置通信』。

您可以在 UNIX and Linux 系统上的 WebSphere MQ 的两种通信形式之间进行选择:

- [第 78 页的『在 UNIX and Linux 上定义 TCP 连接』](#)
- [第 81 页的『在 UNIX and Linux 上定义 LU 6.2 连接』](#)

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 WebSphere MQ Explorer MQI 客户机, 具有使用不同传输协议的备用通道可能很有用。有关 IBM WebSphere MQ Explorer MQI 客户机的更多信息, 请参阅 [IBM WebSphere MQ MQI 客户机概述](#)。

相关概念

[第 23 页的『使用分布式排队连接应用程序』](#)

本部分提供有关 WebSphere 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 62 页的『监视和控制 UNIX, 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 WebSphere 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 83 页的『配置客户机与服务器之间的连接』](#)

要配置 WebSphere MQ MQI 客户机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。

在 UNIX 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。

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 WebSphere 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 WebSphere 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 79 页的『使用 TCP/IP 侦听器』](#)

要在 UNIX and Linux 上启动通道，必须编辑 /etc/services 文件和 inetd.conf 文件

[第 80 页的『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 81 页的『使用 WebSphere MQ 侦听器』](#)

要运行随 WebSphere MQ 提供的侦听器 (将新通道作为线程启动)，请使用 runmqclsr 命令。

[第 81 页的『使用 TCP/IP SO_KEEPALIVE 选项』](#)

在某些 UNIX 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。

使用 TCP/IP 侦听器

要在 UNIX 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 WebSphere 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 WebSphere 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：

- 在 AIX 上:

```
refresh -s inetd
```

- 在 HP-UX 上, 从 mqm 用户标识:

```
inetd -c
```

- 在 Solaris 10 或更高版本上:

```
inetconv
```

- 在其他 UNIX and Linux 系统 (包括 Solaris 9) 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 WebSphere 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 WebSphere 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 表示安装了 WebSphere MQ 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 80 页的『使用 TCP 侦听器储备选项』](#)。

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。

缺省侦听器任务列表值显示在 [第 80 页的表 11](#) 中。

表 11: 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
HP-UX	20
Linux	100
IBM i	255
Solaris	100
Windows 服务器	100

表 11: 在 TCP/IP 端口上排队的最大未完成连接请求数 (继续)	
服务器平台	最大连接请求数
Windows 工作站	100
z/OS	255

如果积压达到 [第 80 页的表 11](#) 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 文件中添加一个条目：

```
TCP:
ListenerBacklog = n
```

这将覆盖 TCP/IP 侦听器的缺省最大未完成请求数 (请参阅 [第 80 页的表 11](#))。

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。

要在打开 **储备** 选项的情况下运行侦听器，请执行以下任一操作：

- 使用 `runmqtsr -b` 命令，或者
- 使用 MQSC 命令 **DEFINE LISTENER**，将 BACKLOG 属性设置为必需值。

有关 `runmqtsr` 命令的信息，请参阅 [runmqtsr](#)。有关 DEFINE LISTENER 命令的信息，请参阅 [定义侦听器](#)。

使用 WebSphere MQ 侦听器

要运行随 WebSphere MQ 提供的侦听器 (将新通道作为线程启动)，请使用 `runmqtsr` 命令。

例如：

```
runmqtsr -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 WebSphere MQ 侦听器作为可信应用程序运行，如 [第 61 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 WebSphere MQ 侦听器：

```
endmqtsr [-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。

使用 TCP/IP SO_KEEPALIVE 选项

在某些 UNIX 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 51 页的『正在检查通道的另一端是否仍然可用』](#))，那么必须将以下条目添加到队列管理器配置文件 (qm.ini)：

```
TCP:
KeepAlive=yes
```

请参阅 UNIX and Linux 系统的文档以获取更多信息。

在 UNIX 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 12: 远程队列管理器平台的本地 UNIX and Linux 系统上的设置		
远程平台	TPNAME	主题方案路径
没有 CICS 的 z/OS	与远程队列管理器的辅助信息中的相应 TPName 相同。	-
使用 CICS 的 z/OS	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。	-
UNIX and Linux 系统	与远程队列管理器的辅助信息中的相应 TPName 相同。	<code>MQ_INSTALLATION_PATH/bin/amqcrs6a</code>
Windows	如 Windows "运行侦听器" 命令或 Windows 上使用 TpSetup 定义的可调用事务程序中所指定。	<code>MQ_INSTALLATION_PATH\bin\amqcrs6a</code>

`MQ_INSTALLATION_PATH` 表示安装 WebSphere MQ 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 82 页的『发送结束』

在 UNIX 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 82 页的『在 LU 6.2 上接收』

在 UNIX 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。

发送结束

在 UNIX 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
```

在 HP-UX 上，使用 APPCLLU 环境变量来命名发送方应该使用的本地 LU。在 Solaris 上，将 APPC_LOCAL_LU 环境变量设置为本地 LU 名称。

当 WebSphere MQ 尝试建立 SNA 会话时，将在 CPI-C 支持的情况下使用 SECURITY PROGRAM。

在 LU 6.2 上接收

在 UNIX 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` 表示安装 WebSphere MQ 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。在 AIX, Solaris 和 HP-UX 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。

配置客户机与服务器之间的连接

要配置 WebSphere MQ MQI 客户机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。

在 WebSphere MQ 中，对象之间的逻辑通信链路称为通道。用于将 WebSphere MQ MQI 客户机连接到服务器的通道称为 MQI 通道。在链路的每一端设置通道定义，以便 WebSphere MQ MQI 客户机上的 WebSphere MQ 应用程序可以与服务器上的队列管理器进行通信。有关如何执行此操作的详细描述，请参阅 [用户定义的通道](#)。

在定义 MQI 通道之前，必须：

1. 决定您要使用的通信形式。请参阅第 83 页的『要使用的通信类型』。
2. 定义通道两端的连接：

要定义连接，必须执行以下操作：

- 配置连接。
- 记录通道定义所需的参数值。
- 通过启动 侦听器，使服务器能够检测来自 WebSphere MQ MQI 客户机的入局网络请求。

要使用的通信类型

不同的平台支持不同的传输协议。您选择的传输协议取决于 WebSphere MQ MQI 客户机和服务器平台的组合。

根据您的客户机和服务器平台，最多有四种类型的 MQI 通道传输协议：

- LU 6.2
- NetBIOS
- SPX
- TCP/IP

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 WebSphere MQ MQI 客户机，具有使用不同传输协议的备用 MQI 通道可能很有用。

您选择的传输协议可能受到特定 WebSphere MQ MQI 客户机和服务器平台组合的限制。下表中显示了可能的组合。

传输协议	WebSphere MQ MQI 客户机	WebSphere MQ 服务器
TCP/IP	UNIX 系统 Windows	UNIX 系统 Windows z/OS
LU 6.2	UNIX 系统 ¹ Windows	UNIX 系统 ¹ Windows
NetBIOS	Windows	Windows

表 13: 传输协议- WebSphere MQ MQI 客户机和服务器平台的组合 (继续)

传输协议	WebSphere MQ MQI 客户机	WebSphere MQ 服务器
SPX	Windows	Windows
注: 1. Linux for Power Systems 除外		

有关设置不同类型的连接的更多信息，请参阅以下链接：

- [第 71 页的『在 Windows 上定义 TCP 连接』](#)
- [第 78 页的『在 UNIX and Linux 上定义 TCP 连接』](#)
- [第 86 页的『TCP/IP 连接限制』](#)
- [第 72 页的『在 Windows 上定义 LU 6.2 连接』](#)
- [第 81 页的『在 UNIX and Linux 上定义 LU 6.2 连接』](#)
- [第 73 页的『在 Windows 上定义 NetBIOS 连接』](#)
- [第 76 页的『在 Windows 上定义 SPX 连接』](#)

相关概念

[第 86 页的『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。

[第 94 页的『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*。

[第 95 页的『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。对于如何在客户端计算机上创建通道定义有一些限制。

[第 97 页的『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。

[第 102 页的『MQI 通道的通道出口程序』](#)

三种类型的通道出口可用于 UNIX，Linux 和 Windows 系统上的 WebSphere MQ MQI 客户机环境。

[第 105 页的『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。

[第 106 页的『使用配置文件配置客户机』](#)

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

[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 WebSphere MQ MQI 客户机和服务器平台的组合。

不同平台上的 MQI 通道有四种类型的通信：

- LU 6.2

- NetBIOS
- SPX
- TCP/IP

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 WebSphere MQ MQI 客户机，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 WebSphere 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。

传输协议	WebSphere MQ MQI 客户机	WebSphere MQ 服务器
TCP/IP	UNIX 系统 Windows	UNIX 系统 Windows
LU 6.2	UNIX 系统 ¹ Windows	UNIX 系统 ¹ Windows
NetBIOS	Windows	Windows
SPX	Windows	Windows

注:

1. Linux (POWER 平台) 除外

相关概念

第 71 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 78 页的『在 UNIX 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 72 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 81 页的『在 UNIX 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 73 页的『在 Windows 上定义 NetBIOS 连接』

WebSphere MQ 在与其他 WebSphere 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。

第 76 页的『在 Windows 上定义 SPX 连接』

SPX 连接仅适用于运行 Windows XP 和 Windows 2003 Server 的客户机和服务器。

相关参考

第 86 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。

定义 TCP/IP 连接

在客户机端的通道定义上指定传输类型 TCP。在服务器上启动侦听器程序。

通过在通道定义上指定传输类型 TCP，在客户机上指定 TCP/IP 连接。

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。启动侦听器程序的过程取决于服务器平台。

请参阅客户机和服务器平台的相关主题。

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 WebSphere 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：

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

如果达到连接限制，那么客户机从 MQCONN 调用接收到 MQRC_HOST_NOT_AVAILABLE 返回码，并在客户机错误日志中接收到 AMQ9202 错误 (UNIX and Linux 系统上的 /var/mqm/errors/AMQERR0n.LOG 或 Windows 上的 IBM WebSphere MQ 客户机安装 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。

定义 NetBIOS 或 SPX 连接

NetBIOS 和 SPX 连接仅适用于 Windows 系统。

NetBIOS 连接仅适用于运行 Windows 的客户机和服务器。请参阅 [定义 NetBIOS 连接](#)。

SPX 连接仅适用于运行 Windows XP 或 Windows 2003 Server 的客户机和服务器。请参阅 [定义 SPX 连接](#)。

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。

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 WebSphere 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

较早版本的 WebSphere MQ 支持 WebSphere Application Server V 4 或 V 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 V 6 和更高版本包含 WebSphere MQ 消息传递提供程序，因此您不需要使用扩展事务客户机。

相关概念

第 87 页的『[配置符合 XA 的事务管理器](#)』

首先配置 WebSphere 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。

第 93 页的『[Microsoft Transaction Server](#)』

在可以将 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。

配置符合 XA 的事务管理器

首先配置 WebSphere 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 WebSphere MQ 基本客户机，如安装 IBM WebSphere MQ 客户机中所述。通过使用本部分中的信息，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 MQCONNEX 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 MQCONNEX，那么 MQCONN 或 MQCONNEX 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 WebSphere 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 88 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 91 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 92 页的『[为 CICS 配置扩展事务客户机](#)』
- 第 93 页的『[为 Tuxedo 配置扩展事务客户机](#)』

相关概念

第 89 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 90 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 91 页的『[将扩展事务客户机与 SSL 通道配合使用](#)』

不能使用 `xa_open` 字符串设置 SSL 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 90 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 90 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。

`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，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。

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 QSG 名称，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。

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。

以下是 xa_open 字符串的示例:

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

xa_open 字符串的 CHANNEL，TRPTYPE，CONNNAME 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。

如果在 xa_open 字符串中未提供 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由 QMNAME 参数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。

如果应用程序客户机正在连接到 V7.0.1 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。

如果应用程序在 QMNAME 参数字段中使用 QSG 名称，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。

如果应用程序连接到版本低于 V7.0.1 的队列管理器，那么 xa_open 调用成功，但事务具有 QMGR 恢复单元处置。确保需要 GROUP 恢复单元处置的应用程序仅连接到 V7.0.1 或更高版本的队列管理器。

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。扩展事务客户机会检查 MQCONN 或 MQCONNX 调用上 QMgrName 参数的值是否是 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。扩展事务客户机检查 *xa_open* 字符串中 QMNAME 参数的值是否是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 90 页的表 16 以获取 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	HP-UX	/opt/encina/lib/libEncServer.sl
CICS	Solaris	/opt/encina/lib/libEncServer.so
CICS	Windows 系统	libEnc 服务器
Tuxedo	AIX	/usr/lpp/tuxedo/lib/libtux.a(libtux.so.60)
Tuxedo	HP-UX	/opt/tuxedo/lib/libtux.sl
Tuxedo	Solaris	/opt/tuxedo/lib/libtux.so.60
Tuxedo	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, 如果您需要更改任何配置信息, 那么必须确保在进行更改之前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:

MQRMIASwitch

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。

MQRMIASwitchDynamic

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 91 页的表 17](#) 中显示的库中。

平台	包含 XA 开关结构的库
AIX HP-UX Linux Solaris	<code>MQ_INSTALLATION_PATH/lib/libmqcxa</code>
Windows 系统	<code>MQ_INSTALLATION_PATH\bin\mqcxa.dll</code> ¹
<i>MQ_INSTALLATION_PATH</i> 表示安装 WebSphere MQ 的高级目录。	

每个交换机结构中的 WebSphere 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相同的交换机结构。

相关概念

[第 91 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 SSL 通道配合使用

不能使用 xa_open 字符串设置 SSL 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 SSL 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 SSL 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 SSL 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 资源定义联机 (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, 供以下每个平台上的 CICS 使用: AIX, HP-UX, Solaris 和 Windows 系统。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 92 页的表 18 以获取每个交换机装入文件的标准路径名。

平台	切换装入文件
AIX HP-UX Linux Solaris	MQ_INSTALLATION_PATH/lib/amqczsc
Windows 系统	MQ_INSTALLATION_PATH\bin\mqcc4swi.dll ¹

MQ_INSTALLATION_PATH 表示安装 WebSphere MQ 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="WebSphere MQ queue manager MARS" \
  XAOpen="channel=MARS.SVR,trptype=tcp,connname=MARS(1415),qmname=MARS,tpm=cics" \
  SwitchLoadFile="C:\Program Files\IBM\WebSphere MQ\bin\mqcc4swi.dll"
```

有关向 CICS 区域添加 XAD 资源定义的更多信息, 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 WebSphere MQ 的一个 XAD 资源定义添加到 CICS 区域。此表示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 WebSphere MQ MQI 客户机应用程序都使用相同的 MQI 通道。
- 当 MQI 通道启动, 并且该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。在特定情况下, 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, 那么必须确保它有权访问它需要访问的所有资源。

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 [安全性](#)。

- 第 93 页的表 19 中列出了提供用于 WebSphere MQ 客户机系统的 CICS 任务终止出口。配置这些出口的方式与配置 WebSphere 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](#)。

平台	来源	库
AIX HP-UX Linux Solaris	amqzscgx.c	amqczscg
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。在 AIX，HP-UX 和 Solaris 上，文件名为 ubbstxcx.cfg，在 Windows 系统上，文件名为 ubbstxcn.cfg。

- 在 Tuxedo 资源管理器表中的队列管理器条目中：
 - udataobj/RM (AIX，HP-UX 和 Solaris)
 - udataobj\rm (Windows 系统)

指定 XA 开关结构的名称以及包含该结构的库的标准路径名。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。

Microsoft Transaction Server

在可以将 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在 WebSphere MQ MQI 客户机上授予 WebSphere MQ 应用程序对通道的访问权。

详细描述了这两种方法：

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

这适用于 WebSphere MQ MQI 客户机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 95 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 WebSphere MQ MQI 客户机时使用此方法。

有关如何使用此方法的详细信息，请参阅第 97 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 WebSphere MQ 产品包含一个功能部件，该功能部件可以在服务器上自动创建通道定义 (如果不存在)。

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 WebSphere 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。

有关自动创建通道定义的更多信息，请参阅 [自动定义接收方和服务器连接通道](#)。

相关概念

[第 94 页的『自动定义的通道』](#)

除 z/OS 以外的平台上的 WebSphere MQ 产品包含一个功能部件，该功能部件可以在服务器上自动创建通道定义 (如果不存在)。

[第 95 页的『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在 WebSphere MQ MQI 客户机上授予 WebSphere MQ 应用程序对通道的访问权。

[第 45 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。

自动定义的通道

除 z/OS 以外的平台上的 WebSphere MQ 产品包含一个功能部件，该功能部件可以在服务器上自动创建通道定义 (如果不存在)。

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 WebSphere 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。

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在 WebSphere MQ MQI 客户机上授予 WebSphere MQ 应用程序对通道的访问权。

详细描述了这两种方法：

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

这适用于 WebSphere MQ MQI 客户机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 95 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 WebSphere MQ MQI 客户机时使用此方法。

有关如何使用此方法的详细信息，请参阅 [第 97 页的『在服务器上创建服务器连接和客户机连接定义』](#)。

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。对于如何在客户端计算机上创建通道定义有一些限制。

在所有平台上，都可以使用 WebSphere MQ Script (MQSC) 命令，可编程命令格式 (PCF) 命令或 IBM WebSphere MQ Explorer 来定义服务器上的服务器连接通道。

由于 MQSC 命令在仅作为 WebSphere MQ MQI 客户机安装了 WebSphere MQ 的机器上不可用，因此必须使用不同的方法在客户机上定义客户机连接通道。

相关概念

[第 96 页的『在 IBM WebSphere MQ MQI 客户机上创建客户机连接通道』](#)

您可以在客户机工作站上使用 MQSERVER 或在 MQCONNX 调用上使用 MQCNO 结构定义客户机连接通道。

相关任务

[第 95 页的『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。

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 SSL 或 TLS 主题专有名称。有关更多信息, 请参阅 [通道认证记录](#)

在 IBM WebSphere MQ MQI 客户机上创建客户机连接通道

您可以在客户机工作站上使用 MQSERVER 或在 MQCONN 调用上使用 MQCNO 结构定义客户机连接通道。

使用 MQSERVER

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。

- 在 Windows 上指定简单通道定义, 如下所示:

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

- 在 UNIX and Linux 系统上指定简单通道定义, 如下所示:

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

其中:

- ChannelName 必须与服务器上定义的名称相同。它不能包含正斜杠。
- TransportType 可以是下列其中一个值, 具体取决于 IBM WebSphere MQ MQI 客户机平台:
 - LU62
 - TCP
 - NETBIOS
 - SPX

注: 在 UNIX and Linux 系统上, TransportType 区分大小写, 并且必须为大写。如果无法识别 TransportType, 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。

例如, 在 Windows 上:

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

或者, 在 UNIX and Linux 系统上:

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

注: 要更改 TCP/IP 端口号, 请参阅 [第 123 页的『MQSERVER』](#)。

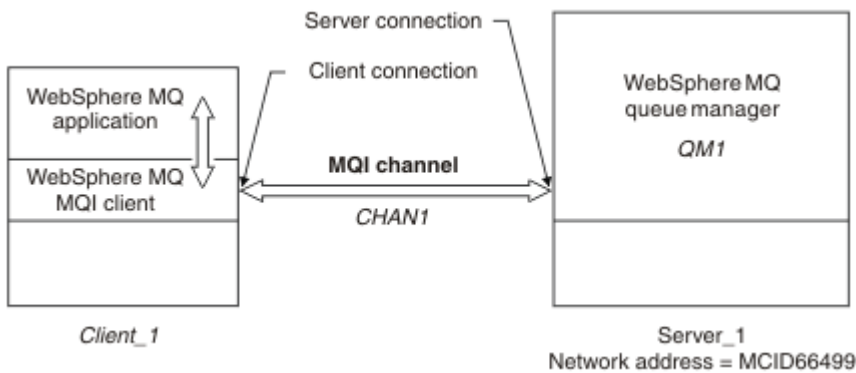


图 17: 简单通道定义

以下是简单通道定义的更多示例:

- 在 Windows 上:

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

- 在 UNIX and Linux 系统上:

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

其中 BOX99 是 LU 6.2 ConnectionName。

在 IBM WebSphere MQ MQI 客户机上, 所有 **MQCONN** 或 **MQCONNX** 请求都会尝试使用您定义的通道, 除非在提供给 **MQCONNX** 的 MQCNO 结构中引用的 MQCD 结构中覆盖该通道。

注: 有关 **MQSERVER** 环境变量的更多信息, 请参阅第 123 页的『MQSERVER』。

在 MQCONNX 调用上使用 MQCNO 结构

IBM WebSphere MQ MQI 客户机应用程序可以在 **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。

如果通道使用安全套接字层 (SSL) 或传输层安全性 (TLS), 那么客户机应用程序还可以在 MQCD 结构中提供与 SSL 或 TLS 相关的信息。可以在 SSL 或 TLS 配置选项结构 MQSCO 中提供与 SSL 或 TLS 相关的其他信息, MQSCO 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。

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。

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 WebSphere MQ Script (MQSC) 命令, 可编程命令格式 (PCF) 命令或 IBM WebSphere MQ Explorer 在服务器上定义服务器连接通道。

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。

相关概念

第 98 页的『[客户机通道定义表](#)』

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 z/OS 以外的平台上，将自动创建 CCDT。然后，必须使其可供客户机应用程序使用。

相关任务

第 99 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 100 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 101 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，使客户机通道定义表 (CCDT) 可供客户机应用程序使用，然后在客户机上指定其位置和名称。

客户机通道定义表

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 z/OS 以外的平台上，将自动创建 CCDT。然后，必须使其可供客户机应用程序使用。

客户机通道定义表 (CCDT) 的用途是确定客户机应用程序用于连接到队列管理器的通道定义。通道定义还指定应用于连接的认证信息。

CCDT 是一个二进制文件。它由队列管理器生成。队列管理器未读取 CCDT 文件。

在 z/OS 以外的平台上，创建队列管理器时会创建 CCDT。当您使用 **DEFINE CHANNEL** 命令时，会将客户机连接通道添加到表中，并且在您发出 **ALTER CHANNEL** 命令时更改了它们的定义。

您可以使用 CCDT 向客户机提供用于检查 SSL 证书撤销的认证信息。定义包含认证信息对象的名称列表，并将队列管理器属性 **SSLCRLNameList** 设置为名称列表的名称。

客户机应用程序可通过多种方法使用 CCDT。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。

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。

除 z/OS 以外的平台

创建队列管理器时，将创建名为 AMQCLCHL.TAB 的缺省 CCDT。

缺省情况下，AMQCLCHL.TAB 位于服务器上的以下目录中：

-   在 UNIX and Linux 系统上：

```
/prefix/qmgrs/QUEUEMANAGERNAME/@ipcc
```

在 UNIX and Linux 系统上，*QUEUEMANAGERNAME* 引用的目录的名称区分大小写。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。

-  在 Windows 上：

```
MQ_INSTALLATION_PATH\data\qmgrs\QUEUEMANAGERNAME\@ipcc
```

MQ_INSTALLATION_PATH 表示安装 IBM WebSphere MQ 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。

可通过设置 MQCHLLIB 来更改 CCDT 的路径。如果设置了 MQCHLLIB，请注意，如果在同一服务器上具有多个队列管理器，那么它们共享同一个 CCDT 位置。

CCDT 是在创建队列管理器时创建的。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。

如何在客户机上指定 CCDT 的位置

在客户机系统上，可以通过两种方式指定 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。

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。

相关参考

[第 121 页的『MQCHLLIB』](#)

MQCHLLIB 指定包含客户机通道定义表 (CCDT) 的文件的目录路径。该文件在服务器上创建，但可以复制到 WebSphere MQ MQI 客户机工作站。

相关信息

[使用已撤销证书](#)

迁移和客户机通道定义表 (CCDT)

通常，客户机通道定义表的内部格式可能从 IBM WebSphere MQ 的一个发行版级别更改为下一个发行版级别。因此，仅当 IBM WebSphere MQ MQI 客户机已由与客户机处于相同发行版级别或较低发行版级别的服务器队列管理器准备时，才可以使用客户机通道定义表。

V 7.1 IBM WebSphere MQ MQI 客户机可以使用由 V 6.0 队列管理器准备的客户机通道定义表。但是，V 6.0 客户机无法使用已由 V 7.1 队列管理器准备的客户机通道定义表。

Active Directory 中的客户机连接通道

在支持 Active Directory 的 Windows 系统上，IBM WebSphere 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，会将这些对象写入客户机通道定义文件，缺省情况下称为 AMQCLCHL.TAB。如果客户机连接通道使用 TCP/IP 协议，那么 IBM WebSphere MQ 服务器还会在 Active Directory 中发布这些协议。当 IBM WebSphere 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 WebSphere 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 SSL 或 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。

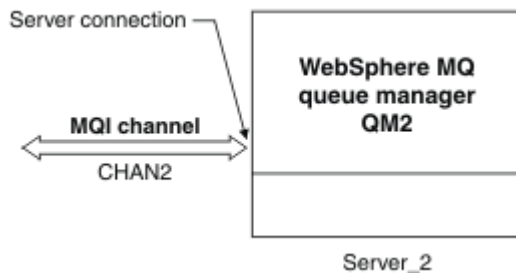


图 18: 定义服务器连接通道

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 WebSphere 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 SSL 或 TLS 主题专有名称。有关更多信息, 请参阅 [通道认证记录](#)

结果

在 z/OS 以外的平台上，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 98 页的『客户机通道定义表』](#)。

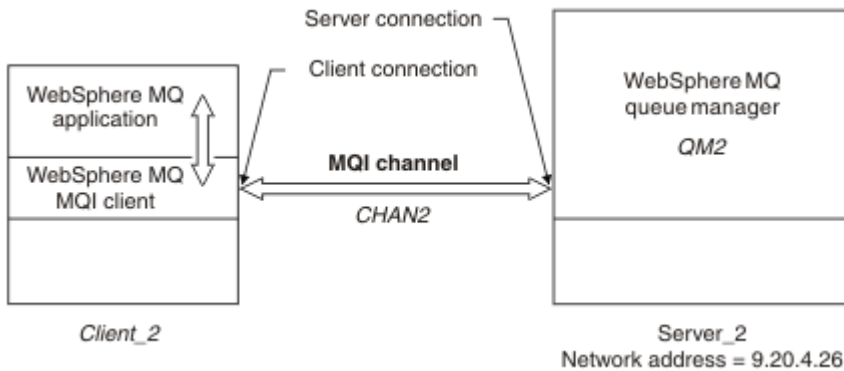


图 19: 定义客户机连接通道

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，使客户机通道定义表 (CCDT) 可供客户机应用程序使用，然后在客户机上指定其位置和名称。

开始之前

您已定义所需的客户机连接通道。

在 z/OS 上，您已创建 CCDT。在其他平台上，将自动创建和更新 CCDT。

关于此任务

要让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

过程

1. 通过以下三种方法之一使 CCDT 可供客户机应用程序使用：

- 可选：将 CCDT 复制到客户端计算机。
- 可选：将 CCDT 复制到由多个客户机共享的位置。
- 可选：将 CCDT 保留在服务器上，但使其可由客户机共享。

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：

- 可选：使用客户机配置文件的 CHANNELS 节。有关更多信息，请参阅 [第 112 页的『客户机配置文件的 CHANNELS 节』](#)。
- 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 HP Integrity NonStop Server 和 UNIX and Linux 系统上：

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

其中 `MQ_INSTALLATION_PATH` 表示安装 WebSphere MQ 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。如果设置了 `MQSERVER` 环境变量，那么 WebSphere MQ 客户机将使用 `MQSERVER` 指定的客户机连接通道定义，而不是使用客户机通道定义表中的任何定义。

MQI 通道的通道出口程序

三种类型的通道出口可用于 UNIX, Linux 和 Windows 系统上的 WebSphere MQ MQI 客户机环境。

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 WebSphere MQ 的一部分提供, 但您可能希望使用不同的压缩技术)
- 对用户数据进行加密和解密 (此功能是作为 WebSphere MQ 的一部分提供的, 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 WebSphere 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 103 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 WebSphere MQ 可安装服务](#)

相关任务

[扩展队列管理器设施](#)

相关参考

第 102 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 104 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。

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 UNIX, Linux 和 Windows 系统上, 将在安装 WebSphere MQ MQI 客户机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 ClientExit 路径节, 并装入在通道定义中指定的任何通道出口。

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 104 页的图 20 说明了如何在客户机连接中使用安全出口，使用 WebSphere 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。

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，此功能可用于更改值，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 20: 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 WebSphere MQ v7.1 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](#)。

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。

API 调用	请求的字节 10 的值	应答的字节 10 的值
MQCONN 第 105 页的『1』 , 第 105 页的『2』	X'81'	X'91'
MQ 描述 第 105 页的『1』	X'82'	X'92'
MQ 打开 第 105 页的『3』	X'83'	X'93'
MQCLOSE	X'84'	X'94'
MQGET 第 105 页的『4』	X'85'	X'95'
MQPUT 第 105 页的『4』	X'86'	X'96'

表 20: 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
MQPUT1 请求 第 105 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: <ol style="list-style-type: none"> 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，MQCONN 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。 		

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。

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。

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。

通过直接连接到队列共享组中的队列管理器，您可以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 WLM/DNS 组名或 VTAM 通用资源名称，也可以是队列共享组的另一个公共接口。

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。为此，您需要在组中的每个队列管理器上具有相同的定义。

定义 SVRCONN 通道，如下所示：

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') 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。

定义 CLNTCONN 通道，如下所示：

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

因为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，所以您现在可以连接到该组中的任何队列管理器，并放入该组拥有的共享队列。

相关概念

[第 106 页的『创建通道定义』](#)

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。为此，您需要在组中的每个队列管理器上具有相同的定义。

创建通道定义

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。为此，您需要在组中的每个队列管理器上具有相同的定义。

定义 SVRCONN 通道，如下所示：

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') 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。

定义 CLNTCONN 通道，如下所示：

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

因为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，所以您现在可以连接到该组中的任何队列管理器，并放入该组拥有的共享队列。

使用配置文件配置客户机

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。

使用类似于 UNIX and Linux 平台上使用的队列管理器配置文件 qm.ini 的文本文件来配置 IBM WebSphere MQ MQI client。该文件包含多个节，每个节都包含许多格式为 **attribute-name=value** 的行。

在本文档中，此文件称为 *WebSphere MQ MQI* 客户机配置文件；其文件名通常为 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所有平台以及使用 MQI，IBM WebSphere MQ classes for Java，IBM WebSphere MQ classes for JMS，IBM WebSphere MQ classes for .NET 和 XMS 的客户机。

虽然 IBM WebSphere MQ MQI client 配置文件中的属性适用于大多数 IBM WebSphere 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 WebSphere MQ classes for Java 或 IBM WebSphere MQ classes for JMS 的客户机读取。有关更多信息，请参阅第 108 页的『[哪些 IBM WebSphere 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。

IBM WebSphere MQ 版本低于 7.0 的发行版中支持的环境变量继续受支持，如果此类环境变量与客户机配置文件中的等效值匹配，那么环境变量将覆盖客户机配置文件值。

对于使用 IBM WebSphere 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：

- 在 JMS 配置文件中设置属性
- 设置 Java 系统属性，这也会覆盖 JMS 配置文件

对于 .NET 客户机，还可以使用 .NET 应用程序配置文件覆盖客户机配置文件和等效环境变量。

请注意，不能使用客户机配置文件来设置多个通道连接。

示例客户机配置文件

```

#* Module Name: mqclient.ini                                     *#
#* Type       : WebSphere 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=32768
  ClntRcvBuffSize=32768
  Connect_Timeout=0

MessageBuffer:
  MaximumSize=-1
  Updatepercentage=-1
  PurgeTime=0

LU62:
  TPName
  Library1=DLLName1
  Library2=DLLName2

PreConnect:
  Module=amqldapi
  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)

```

相关参考

[第 108 页的『客户机配置文件的位置』](#)

IBM WebSphere MQ MQI 客户机配置文件可以保存在多个位置。

[第 112 页的『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。

[第 114 页的『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。

[第 114 页的『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。

第 115 页的『[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。

第 116 页的『[客户机配置文件的 SSL 节](#)』
使用 SSL 节来指定有关使用 SSL 或 TLS 的信息。

第 118 页的『[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。

第 119 页的『[使用 WebSphere MQ 环境变量](#)』
本部分描述了可用于 WebSphere MQ MQI 客户机应用程序的环境变量。

第 358 页的『[更改队列管理器配置信息](#)』
此处描述的属性用于修改单个队列管理器的配置。它们覆盖 WebSphere MQ 的任何设置。

客户机配置文件的位置

IBM WebSphere MQ MQI 客户机配置文件可以保存在多个位置。

客户机应用程序使用以下搜索路径来查找 IBM WebSphere MQ MQI 客户机配置文件:

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。

请注意下列事项:

- 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. Windows UNIX and Linux 系统的 IBM WebSphere MQ 数据目录中名为 `mqclient.ini` 的文件。

请注意下列事项:

- IBM WebSphere MQ 数据目录在某些平台 (例如, IBM i 和 z/OS) 上不存在, 或者在客户机已随其他产品一起提供的情况下不存在。
- 在 UNIX and Linux 系统上, 目录为 `/var/mqm`
- 在 Windows 平台上, 在安装期间配置环境变量 `MQ_FILE_PATH` 以指向数据目录。通常为 `C:\Program Files\IBM\WebSphere MQ`
- 要允许不支持读取环境变量的 Java 1.4.2 JRE, 可以使用 `MQ_FILE_PATH` Java 系统属性手动覆盖 `MQ_FILE_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。
- 对于 UNIX and Linux 平台上的 C 客户机, 这是 `HOME` 环境变量的值。
- 对于 Windows 上的 C 客户机, 这是 `HOMEDRIVE` 和 `HOMEPATH` 环境变量的并置值。

注: 对于 HP Integrity NonStop Server 的 IBM WebSphere MQ 客户机, `mqclient.ini` 文件必须位于 OSS 文件系统中。守护程序应用程序必须将 `mqclient.ini` 文件放在 IBM WebSphere MQ 数据目录中, 或者将 MQCLNTCF 环境变量设置为 OSS 文件系统中的某个位置。

哪些 IBM WebSphere MQ 客户机可以读取每个属性

IBM WebSphere 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 WebSphere MQ classes for Java 或 IBM WebSphere MQ classes for JMS 的客户机读取。

表 21: 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						
<u>CCSID</u>	要使用的编码字符集编号。	Yes	否	否	Yes	Yes
<u>ChannelDefinitionDirectory</u>	包含客户机通道定义表的文件的目录路径。	Yes	否	否	Yes	Yes
<u>ChannelDefinitionFile</u>	包含客户机通道定义表的文件的名称。	Yes	否	否	Yes	Yes
<u>ReconDelay</u>	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
<u>DefRecon</u>	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
<u>MQReconnectTimeout</u>	重新连接到客户机的超时 (以秒计)。	Yes	否	否	Yes	否
<u>ServerConnectionParms</u>	IBM WebSphere MQ 服务器的位置以及要使用的通信方法。	Yes	否	否	Yes	Yes
<u>Put1DefaultAlwaysSync</u>	通过选项 MQPMO_RESPONSE_AS_Q_DEF 控制 MQPUT1 函数调用的行为。	Yes	Yes	Yes	Yes	Yes
ClientExit 路径节						
<u>ExitsDefaultPath</u>	指定客户机的 32 位通道出口的位置。	Yes	否	否	Yes	Yes


表 21: 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
ExitsDefaultPath64	指定客户机的 64 位通道出口的位置。	Yes	否	否	Yes	Yes
JavaExitsClassPath	运行 Java 出口时要添加到类路径的值。	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
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
SSL 节						
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
CertificateValidationStrategy	确定所使用的证书验证类型。	Yes	否	否	否	否
ClientRevocationCheck	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

表 21: 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
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
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 WebSphere MQ 的行为。	Yes	否	否	否	否
OCSPCheckExtensions	控制 IBM WebSphere MQ 是否对 AuthorityInfo 访问证书扩展执行操作。	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 WebSphere MQ 中执行密码术时是否仅使用 FIPS 认证的算法。	Yes	否	否	否	否
SSLHTTPProxyName	该字符串是供 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
SSLKeyResetCount	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

表 21: 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
TCP 节						
ClntRcvBufferSize	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
ClntSndBufferSize	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
 Library1	(仅在 Windows 上) TCP/IP 套接字 DLL 的名称。	Yes	否	否	否	否

对于 HP Integrity NonStop Server, 可以使用 [TMF](#) 和 [TmfGateway](#) 节来与 TMF/Gateway 进行通信。

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。

CHANNELS 节中可以包含以下属性:

CCSID=number

要使用的编码字符集编号。

CCSID 号等同于 MQCCSID 环境参数。

ChannelDefinitionDirectory=路径

包含客户机通道定义表的文件的目录路径。

在 Windows 系统上, 缺省值为 IBM WebSphere MQ 安装目录, 通常为 C:\Program Files\IBM\WebSphere MQ。在 UNIX and Linux 系统上, 缺省值为 /var/mqm。

ChannelDefinition 目录路径等同于 MQCHLLIB 环境参数。

ChannelDefinitionFile=文件名| AMQCLCHL.TAB

包含客户机通道定义表的文件的名称。

客户机通道定义表等同于 MQCHLTAB 环境参数。

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。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。

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。

IBM WebSphere MQ Java 类不支持自动客户机重新连接。

否

除非被 MQCONNX 覆盖，否则不会自动重新连接客户机。

是

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 22: 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
否	YES	QMGR	否	否
是	YES	QMGR	YES	否
QMGR	YES	QMGR	QMGR	否
DISABLED	否	否	否	否

MQReconnectTimeout

重新连接到客户机的超时 (以秒计)。缺省值是 1800 秒 (30 分钟)。

IBM WebSphere MQ classes for XMS .NET 客户机可以使用属性 XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT。该属性的缺省值是 1800 秒 (30 分钟)。

ServerConnectionParms

ServerConnection 参数等同于 MQSERVER 环境参数，用于指定 IBM WebSphere MQ 服务器的位置以及要使用的通信方法。ServerConnectionParms 属性仅定义简单通道；不能使用它来定义 SSL 通道或具有通道出口的通道。它是格式为 *ChannelName/TransportType/ConnectionName* 的字符串，*ConnectionName* 必须是标准网络名。*ChannelName* 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 ServerConnection 参数来定义客户机通道时，将使用最大消息长度 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|YES

通过选项 MQPMO_RESPONSE_AS_Q_DEF 控制 MQPUT1 函数调用的行为。

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。

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。

以下属性可以包含在 ClientExitPath 节中：

ExitsDefaultPath=字符串

指定客户机的 32 位通道出口的位置。

ExitsDefaultPath64=字符串

指定客户机的 64 位通道出口的位置。

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。

在 JMS 配置文件中，将为 JavaExitsClassPath 名称提供标准 com.ibm.mq.cfg。前缀和此全名也用于 Websphere MQ V7.0 系统属性。在 V 6.0，此属性是使用系统属性 com.ibm.mq.exitClasspath 指定的，此属性记录在 V 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，那么仍在 Websphere MQ V7.0 中使用。

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

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

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 环境参数。

NumCmds=number| 1

要分配的命令数。

NumSess=number| 1

要分配的会话数。

SPX

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。

Socket=number| 5E86

以十六进制表示法表示的 SPX 套接字号。

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。

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。

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)。

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=integer|_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 指示客户机确定相应的值。

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：

$$T = \text{Upper} - \text{Lower}$$

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：

$$\text{Lower} = \text{Upper} \times \text{UpdatePercentage} / 100$$

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

$$T = \text{Upper} - \text{Lower}$$

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

$$T = \text{Upper} - \text{Lower}$$

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。

客户机配置文件的 SSL 节

使用 SSL 节来指定有关使用 SSL 或 TLS 的信息。

以下属性可以包含在 SSL 节中：

CDPCheckExtensions=YES|NO

CDPCheckExtensions 指定此队列管理器上的 SSL 或 TLS 通道是否尝试检查在 *CrlDistribution* 点证书扩展中指定的 CDP 服务器。

此属性具有以下可能的值：

- YES: SSL 或 TLS 通道尝试检查 CDP 服务器以确定是否撤销数字证书。
- NO: SSL 或 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。

CertificateValPolicy=字符串

确定所使用的证书验证类型。

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 = REQUIRED | OPTIONAL | DISABLED


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。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。

 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。

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。可能的值为：

无

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。

OCSPAuthentication=OPTIONAL | REQUIRED | WARN

定义在启用 OCSP 且 OCSP 撤销检查无法确定证书撤销状态时 WebSphere MQ 的行为。共有三个可能的值：

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 MQRQ_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 MQRQ_CHANNEL_SSL_WARNING 消息。允许连接继续

OCSPCheckExtensions=YES|NO

控制 WebSphere MQ 是否对 AuthorityInfo 访问证书扩展执行操作。如果该值设置为 NO，那么 WebSphere MQ 将忽略 AuthorityInfo 访问证书扩展，并且不会尝试 OCSP 安全性检查。缺省值为 YES。

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。

指定以下格式的字符串: GSK_PKCS11=*driver path and filename; token label; token password*; 对称密码设置;

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:

SYMMETRIC_CIPHER_OFF

请勿访问对称密码操作。此设置为缺省设置。

SYMMETRIC_CIPHER_ON

访问对称密码操作。

字符串的最大长度为 256 个字符。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=YES|_NO

指定在 WebSphere MQ 中执行密码术时是否仅使用 FIPS 认证的算法。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

该字符串是供 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。在 HP-UX PA-RISC 和 Sun Solaris SPARC 平台上，以及对于 AIX 上的 32 位客户机，网络地址只能是 IPv4 地址;在其他平台上，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。

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。

SSLKeyResetCount=*integer*|_0

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。

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 SSL 或 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。

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。

以下属性可以包含在 TCP 节中:

ClntRcvBuffSize=number| 32768

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)。值为零表示操作系统将管理缓冲区大小,而不是由 WebSphere MQ 固定缓冲区大小。

ClntSndBuffSize=number| 32768

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)。值为零表示操作系统将管理缓冲区大小,而不是由 WebSphere MQ 固定缓冲区大小。

Connect_Timeout=number

尝试连接套接字超时之前的秒数;缺省值为 0,除非已使用非零客户机通道权重配置通道,在这种情况下,缺省值为 5。

IPAddressVersion=MQIPADDR_IPV4|MQIPADDR_IPV6

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 具有相同的含义。

KeepAlive=YES|NO

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。

Windows Library1=DLLName| WSOCK32

(仅限 Windows) TCP/IP 套接字 DLL 的名称。

TMF 和 TMF/Gateway 节

IBM WebSphere MQ 提供的 TMF/Gateway 在 Pathway 环境中运行。使用 TMF 和 TMF/Gateway 节来指定 IBM WebSphere MQ Client for HP Integrity NonStop Server 与 TMF/Gateway 通信所需的配置参数。

如果要使用 TMF,那么必须为要与之通信的每个队列管理器定义一个 TMF 节和一个 TmfGateway 节。所有值都派生自您的配置。

TMF 节

PathMon=名称

定义 TMF/Gateway 的服务器类的已定义 Pathmon 进程的名称。

TmfGateway 节

以下属性可以包含在此节中:

QManager=名称

队列管理器的名称。

Server=名称

为该队列管理器配置的 TMF/Gateway 的服务器类名。

示例

以下是使用两个 TmfGateway 节为不同服务器上的两个不同队列管理器定义的 TMF 节的示例:

```
TMF:
  PathMon=$PSD1P

TmfGateway:
  QManager=MQ5B
  Server=MQ-MQ5B

TmfGateway:
  QManager=MQ5C
  Server=MQ-MQ5C
```

使用 WebSphere MQ 环境变量

本部分描述了可用于 WebSphere MQ MQI 客户机应用程序的环境变量。

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

WebSphere MQ 将缺省值用于尚未设置的变量。

除非另有说明, 否则所有 WebSphere MQ MQI 客户机平台上都提供了命令。

对于每个环境变量, 请使用与您的平台相关的命令来显示当前设置或重置变量的值。

例如:

设置或重置环境变量的值		
结果	命令	
	Windows	UNIX and Linux 系统
除去变量	set MQSERVER=	unset MQSERVER
显示当前设置	设置 MQSERVER	echo \$MQSERVER
显示会话的所有环境变量	set	set

有关各个变量的信息, 请参阅以下子主题:

相关概念

第 106 页的『使用配置文件配置客户机』

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。

相关参考

[环境变量](#)

MQCCSID

MQCCSID 指定要使用的编码字符集号, 并覆盖已配置服务器的 CCSID 值。

请参阅 [选择客户机或服务器编码字符集标识 \(CCSID\)](#) 以获取更多信息。

要设置此变量, 请使用下列其中一个命令:

- 对于 Windows:

```
SET MQCCSID=number
```

- 对于 UNIX and Linux 系统:

```
export MQCCSID=number
```

MQCERTVPOL

MQCERTVPOL 指定使用的证书验证策略。

有关 WebSphere MQ 中的证书验证策略的更多信息, 请参阅 [WebSphere MQ 中的证书验证策略](#)。

此环境变量将覆盖客户机 ini 文件的 SSL 节中的 *CertificateValPolicy* 设置。该变量可以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。

RFC5280

仅使用符合 RFC 5280 标准的证书验证。

要设置此变量，请使用下列其中一个命令：

- 对于 Windows：

```
SET MQCERTVPOL=value
```

- 对于 UNIX and Linux 系统：

```
export MQCERTVPOL=value
```

MQCHLLIB

MQCHLLIB 指定包含客户机通道定义表 (CCDT) 的文件的目录路径。该文件在服务器上创建，但可以复制到 WebSphere MQ MQI 客户机工作站。

如果未设置 MQCHLLIB，那么客户机的路径缺省为：

- **Windows** 对于 Windows: `MQ_INSTALLATION_PATH`
- **UNIX Linux** 对于 UNIX and Linux 系统: `/var/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。

- **Windows** 对于 Windows: `datapath\@ipcc`
- **UNIX Linux** 对于 UNIX 和 Linux 系统: `datapath/@ipcc`

或者：

- **Windows** 对于 Windows: `MQ_INSTALLATION_PATH\data\qmgrs\qmgrname\@ipcc`
- **UNIX Linux** 对于 UNIX and Linux 系统: `/prefix/qmgrs/qmgrname/@ipcc`

其中：

- `MQ_INSTALLATION_PATH` 表示安装 IBM WebSphere MQ 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 `/var/mqm` 在 UNIX 和 Linux 平台上。
- `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。
- 队列管理器节在 UNIX 和 Linux 上的 `mqsc.ini` 文件以及 Windows 上的注册表中定义

注意：

1. 如果设置了此属性，那么 MQCHLLIB 将覆盖用于查找 CCDT 的路径。
2. 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。
3. 如果在服务器上设置 MQCHLLIB 系统范围，那么它将为服务器上的所有队列管理器设置相同的 CCDT 文件路径。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)。
4. 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`。此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。
5. 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。将替换先前创建但并非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。

6. `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。
7. 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。
8. MQCHLLIB 的路径不需要引号。

示例

要设置此变量，请使用下列其中一个命令：

- **Windows** 对于 Windows:

```
SET MQCHLLIB=pathname
```

例如：

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

- **UNIX Linux** 对于 UNIX and Linux 系统:

```
export MQCHLLIB=pathname
```

MQCHLTAB

MQCHLTAB 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。

有关客户机通道定义表在服务器上的位置的信息，请参阅 [第 98 页的『客户机通道定义表』](#)。

要设置此变量，请使用下列其中一个命令：

- 在 Windows 上：

```
SET MQCHLTAB=filename
```

- 在 UNIX and Linux 系统上：

```
export MQCHLTAB=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。

MQIPADDRV

MQIPADDRV 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 具有相同的含义。如果未设置，那么将采用 "MQIPADDR_IPV4"。

要设置此变量，请使用下列其中一个命令：

- 对于 Windows:

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

- 对于 UNIX and Linux 系统:

```
export MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6
```

MQNAME

MQNAME 指定 WebSphere MQ 进程可以使用的本地 NetBIOS 名称。

请参阅第 73 页的『在 Windows 上定义 NetBIOS 连接』以获取完整描述以及客户机和服务器上的优先顺序规则。

要设置此变量，请使用以下命令：

```
SET MQNAME=Your_env_Name
```

例如：

```
SET MQNAME=CLIENT1
```

如果在 WebSphere MQ MQI 客户机上同时运行多个 WebSphere MQ 应用程序，那么某些平台上的 NetBIOS 需要对每个应用程序使用不同的名称（由 MQNAME 设置）。

MQSERVER

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

不能使用 MQSERVER 来定义 SSL 通道或具有通道出口的通道。有关如何定义 SSL 通道的详细信息，请参阅[使用 SSL 保护通道](#)。

ConnectionName 必须是标准网络名。*ChannelName* 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。

要设置此变量，请使用下列其中一个命令：

- 对于 Windows：

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

- 对于 UNIX and Linux 系统：

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

TransportType 可以是下列其中一个值，具体取决于 IBM WebSphere MQ 客户机平台：

- LU62
- TCP
- NETBIOS
- SPX

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。*ConnectionName* 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 *ConnectionName* 列表来指定不同的队列管理器实例。

TCP/IP 缺省端口

缺省情况下，对于 TCP/IP，WebSphere 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：

- 将方括号中的端口号添加为 *ConnectionName* 的最后一部分：
 - 对于 Windows：

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

- 对于 UNIX and Linux 系统:

```
export 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 WebSphere MQ 添加到服务文件, 如 [第 79 页的『使用 TCP/IP 侦听器』](#) 中所述。

SPX 缺省套接字

缺省情况下, 对于 SPX, WebSphere 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:

- 将方括号中的套接字号添加为 `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 WebSphere 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 MQSERVER

如果使用 `MQSERVER` 环境变量来定义 WebSphere MQ MQI 客户机与服务器之间的通道, 那么这是唯一可供应用程序使用的通道, 并且不会引用客户机通道定义表 (CCDT)。

在此情况下, 您在服务器上运行的侦听器程序将确定应用程序将连接到的队列管理器。它将与侦听器程序连接到的队列管理器相同。

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 `MQSERVER` 参数 `TransportType` 无法识别, 那么 `MQCONN` 或 `MQCONNX` 请求将失败, 返回码为 `MQRC_Q_MGR_NAME_ERROR`。

在 UNIX and Linux 系统上, 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` 指向的任何客户机通道定义。

取消 MQSERVER

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:

- 在 Windows 上:

```
SET MQSERVER=
```


- 在 UNIX and Linux 系统上:

```
unset MQSERVER
```

MQSSLCRYP

MQSSLCRYP 包含允许您配置系统上存在的加密硬件的参数字符串。允许的值与 ALTER QMGR 命令的 SSLCRYP 参数相同。

要设置此变量，请使用下列其中一个命令:

- 在 Windows 系统上:

```
SET MQSSLCRYP=string
```

- 在 UNIX and Linux 系统上:

```
export MQSSLCRYP=string
```

相关参考

[ALTER QMGR command SSLCRYP 参数](#)

MQSSLFIPS

MQSSLFIPS 指定在 WebSphere MQ 中执行密码术时是否仅使用 FIPS 认证的算法。这些值与 ALTER QMGR 命令的 SSLFIPS 参数相同。

使用加密硬件会影响 FIPS 认证算法的使用，请参阅 [指定在 MQI 客户机上运行时仅使用 FIPS 认证的 CipherSpecs](#)。

要设置此变量，请使用下列其中一个命令:

- 在 Windows 系统上:

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

- 在 UNIX and Linux 系统上:

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

缺省值为 NO。

MQSSLKEYR

MQSSLKEYR 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。有关完整详细信息，请参阅 ALTER QMGR 命令的 SSLKEYR 参数。

要设置此变量，请使用下列其中一个命令:

- 在 Windows 系统上:

```
SET MQSSLKEYR=pathname
```

- 在 UNIX and Linux 系统上:

```
export MQSSLKEYR=pathname
```

无缺省值。

MQSSLPROXY

MQSSLPROXY 指定 GSKit 用于 OCSP 检查的 HTTP 代理服务器的主机名和端口号。

要设置此变量，请使用下列其中一个命令：

- 在 Windows 系统上：

```
SET MQSSLPROXY=string
```

- 在 UNIX and Linux 系统上：

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

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。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。

例如，在 UNIX 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"
```

MQSSLRESET

MQSSLRESET 表示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。

请参阅 [重置 SSL 和 TLS 密钥](#)，以获取有关密钥重新协商的更多信息。

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 SSL 或 TLS 密钥重置计数，那么 SSL 或 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 SSL 或 TLS 密钥重置值进行过多的密钥重置。

要设置此变量，请使用下列其中一个命令：

- 在 Windows 系统上：

```
SET MQSSLRESET=integer
```

- 在 UNIX and Linux 系统上：

```
export MQSSLRESET=integer
```

控制排队的发布/预订

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。

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：

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](#)

描述	MQSC 参数名称
命令消息重试计数	PSRTYCNT

表 23: 发布/预订配置参数 (继续)	
描述	MQSC 参数名称
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 WebSphere 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

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

注: 如果您已从 Version 6.0 迁移, 那么在使用已升级的队列管理器时, 必须使用 **strmqbrk** 来迁移 Version 6.0 发布/预订代理状态。这不适用于 z/OS。

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 WebSphere MQ Explorer 或 **runmqsc** 命令来启用该接口。

示例

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

下一步做什么

WebSphere MQ 进程已排队的发布/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

开始之前

不推荐使用已排队的发布/预订。

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 WebSphere 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, 以便这些流与从 Version 6.0 队列管理器迁移的流共存。

开始之前

通过阅读主题 [Streams 和主题](#)，熟悉发布/预订流的操作方式。

关于此任务

使用 PCF 命令，**runmqsc** 或 IBM WebSphere 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 Version 6.0 流同名的本地队列。
2. 定义与 Version 6.0 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 Version 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，Version 6.0 和 Version 7.1 队列管理器在同一发布/预订层次结构中工作。Version 6.0 队列管理器共享名为 'Sport' 的流。此示例显示如何在名为 'Sport' 的 Version 7.1 队列管理器上创建队列和主题，并使用与 V 6 stream 'Sport' 共享的主题字符串 'Sport'。

使用主题字符串 'Soccer/Results' 发布到主题 'Sport' 的 Version 7.1 发布应用程序将创建生成的主题字符串 'Sport/Soccer/Results'。在 Version 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 Version 6.0 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, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
define qlocal('Sport')
  1 : define qlocal('Sport')
AMQ8006: WebSphere MQ queue created.
define topic('Sport') topicstr('Sport')
  2 : define topic('Sport') topicstr('Sport')
AMQ8690: WebSphere 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: WebSphere 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。

如果代理是 Version 6.0，请将其作为 Version 6.0 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`。

如果代理是 Version 7.1，那么必须手动配置层次结构中的每个 Version 7.1 队列管理器。

删除流

您可以从 IBM WebSphere MQ Version 7.1 或更高版本的队列管理器中删除流。

开始之前

在 IBM WebSphere MQ Version 7.1 中不推荐使用已排队的发布/预订。

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。

关于此任务

有关从与其连接的任何 Version 6.0 队列管理器中删除流的指示信息，请参阅 [删除流 \(v6.0 文档中的 ps11870_.htm\)](#)。

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

1. 对托管流的所有其他已连接 Version 7.1 或更高版本的队列管理器重复步骤 3 到 5。
2. 从所有其他已连接的 Version 6.0 或更早版本的队列管理器中除去流。

添加预订点

如何添加未通过 **migmbbrk** 从 IBM WebSphere MQ Event Broker 或 IBM WebSphere MQ Message Broker 迁移的预订点。使用新的预订点扩展已从 IBM WebSphere MQ Event Broker 或 IBM WebSphere MQ Message Broker 迁移的现有排队发布/预订应用程序。

开始之前

1. 完成从 IBM WebSphere MQ Event Broker 和 IBM WebSphere MQ Message Broker Version 6.0 到 IBM WebSphere MQ Version 7.1 的迁移。
2. 请检查是否尚未在 `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST` 中定义预订点。
3. 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

现有 IBM WebSphere MQ Event Broker 应用程序使用预订点。新的 IBM WebSphere MQ Version 7.1 应用程序不使用预订点，但可以使用预订点迁移机制与现有应用程序进行互操作。

如果预订点在迁移时未被使用，那么 **migmbbrk** 可能尚未迁移该预订点。

您可能希望将预订点添加到从 IBM WebSphere MQ Event Broker 迁移的现有排队发布/预订程序。

预订点不适用于使用 MQRFH1 头 (已从 IBM WebSphere MQ Version 6.0 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 Version 7.1 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。

将队列管理器连接到代理层次结构

您可以将本地队列管理器连接到父队列管理器以修改代理层次结构。

开始之前

1. 启用排队的发布/预订方式。 请参阅 [启动排队的发布/预订](#)。
2. 此更改将使用 IBM WebSphere MQ 连接传播到父队列管理器。 有两种方法可以建立连接。
 - 将队列管理器连接到 IBM WebSphere MQ 集群，请参阅 [将队列管理器添加到集群](#)
 - 使用与父队列管理器同名的传输队列或队列管理器别名来建立点到点通道连接。 有关如何建立点到点通道连接的更多信息，请参阅 [WebSphere MQ 分布式消息传递技术](#)。

关于此任务

使用 **ALTER QMGR PARENT (PARENT_NAME) runmqsc** 命令将子代连接到父代。

分布式发布/预订通过使用队列管理器集群和集群主题定义来实现。 对于与 IBM WebSphere MQ Version 6.0 和 WebSphere Message Broker Version 6.1 和 WebSphere Event Broker Version 6.1 及更低版本的互操作性，只要启用了排队发布/预订方式，您还可以将 Version 7.1 或更高版本的队列管理器连接到代理层次结构。

过程

ALTER QMGR PARENT (PARENT)

示例

第一个示例显示了如何将 QM2 作为 QM1 的子代进行连接，然后查询 QM2 以获取其连接:

```
C:>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2
alter qmgr parent(QM1)
  1 : alter qmgr parent(QM1)
AMQ8005: WebSphere 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, 2024. 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)
```

下一步做什么

您可以在一个代理或队列管理器上定义可供已连接队列管理器上的发布者和订户使用的主题。有关更多信息, 请参阅 [定义管理主题](#)

相关概念

[流和主题](#)

[WebSphere MQ 发布/预订消息传递简介](#)

相关参考

[显示发布预订](#)

断开队列管理器与代理层次结构的连接

将子队列管理器与代理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, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2.
  1 : alter qmgr parent('')
AMQ8005: WebSphere MQ queue manager changed.
  2 : display pubsub type(child)
AMQ8147: WebSphere MQ object not found.
display pubsub type(parent)
  3 : display pubsub type(parent)
AMQ8147: WebSphere 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。

配置队列管理器集群

使用本主题中的链接可了解集群如何工作, 如何设计集群配置以及如何设置简单集群的示例。

开始之前

有关集群概念的简介, 请参阅以下主题:

- [集群工作方式](#)

- [第 134 页的『集群与分布式排队的比较』](#)
- [第 136 页的『集群的组件』](#)

在设计队列管理器集群时，必须做出一些决策。您必须首先决定集群中的哪些队列管理器将保存集群信息的完整存储库。您创建的任何队列管理器都可以在集群中工作。您可以为此目的选择任意数量的队列管理器，但理想数量为两个。有关选择队列管理器以保存完整存储库的信息，请参阅 [第 147 页的『如何选择集群队列管理器以保存完整存储库』](#)。

请参阅以下主题以获取有关设计集群的更多信息：

- [第 148 页的『组织集群』](#)
- [第 148 页的『集群命名约定』](#)
- [第 149 页的『重叠集群』](#)

示例

最小的可能集群仅包含两个队列管理器。在这种情况下，两个队列管理器都包含完整存储库。您只需要几个定义即可设置集群，然而每个队列管理器都有高度的自主性。

第 133 页的图 21 显示了一个名为 DEMOCLSTR 的集群，其中有两个名为 QM1 和 QM2 的队列管理器。

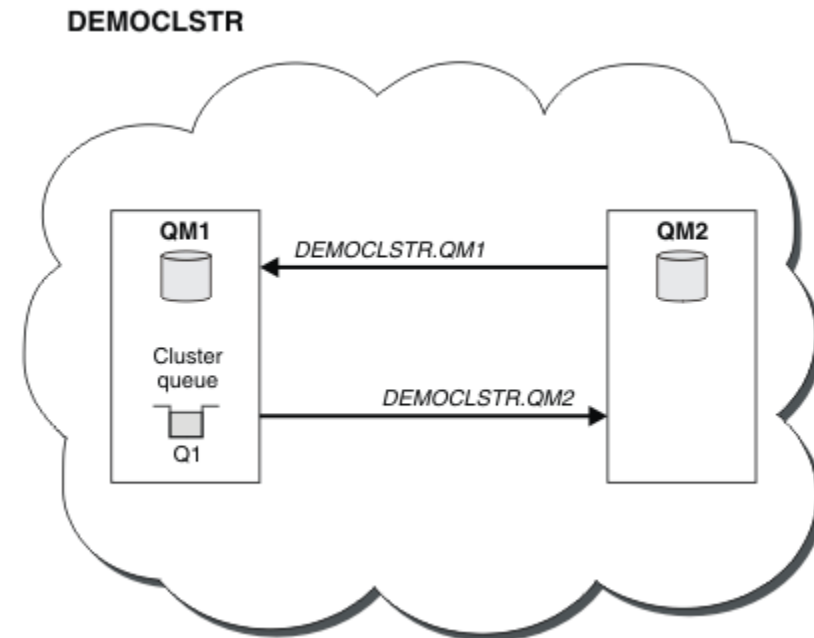


图 21: 两个队列管理器的小型集群

- 队列管理器具有长名称，例如 LONDON 和 NEWYORK。在高级任务和工作负载均衡任务中使用相同的名称。在 IBM WebSphere MQ for z/OS 上，队列管理器名称限制为四个字符。
- 队列管理器的名称意味着每个队列管理器都位于单独的机器上。您可以对同一机器上的所有队列管理器执行这些任务。
- 这些任务使用 IBM WebSphere MQ 脚本命令，因为系统管理员将使用 **MQSC** 命令输入这些命令。还有其他输入命令的方法，包括使用更简单的 IBM WebSphere MQ Explorer。使用 WebSphere MQ 脚本命令的意义在于演示任务中使用的 IBM WebSphere MQ 命令。

有关设置类似示例集群的指示信息，请参阅 [第 153 页的『设置新集群』](#)。

下一步做什么

有关配置和使用集群的更多信息，请参阅以下主题：

- [第 151 页的『在集群中建立通信』](#)

- [第 153 页的『管理 IBM WebSphere MQ 集群』](#)
- [第 208 页的『将消息路由到集群或从集群路由消息』](#)
- [第 220 页的『使用集群进行工作负载管理』](#)

有关帮助您配置集群的更多信息，请参阅 [第 150 页的『集群提示』](#)。

相关概念

[集群](#)

访问控制和多个集群传输队列

在应用程序将消息放入远程集群队列时选择三种检查方式。这些方式是针对集群队列进行远程检查，针对 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 进行本地检查，或者针对集群队列或集群队列管理器的本地概要文件进行检查。

IBM WebSphere MQ 允许您选择在本地或本地和远程检查用户是否有权将消息放入远程队列。典型的 IBM WebSphere MQ 应用程序仅使用本地检查，并且依赖于远程队列管理器信任对本地队列管理器进行的访问检查。如果未使用远程检查，那么将使用远程消息通道进程的权限将消息放入目标队列。要使用远程检查，必须将接收通道的放置权限设置为上下文安全性。

将针对应用程序打开的队列进行本地检查。在分布式队列中，应用程序通常会打开远程队列定义，并且会针对远程队列定义进行访问检查。如果将消息与完整路由头放在一起，那么将针对传输队列进行检查。如果应用程序打开不在本地队列管理器上的集群队列，那么没有要检查的本地对象。将针对集群传输队列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 执行访问控制检查。即使有多个集群传输队列，从 Version 7.5 开始，也会针对 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 进行远程集群队列的本地访问控制检查。

选择本地或远程检查是两个极端之间的选择。远程检查是细颗粒度的。每个用户都必须在集群中的每个队列管理器上具有访问控制概要文件才能放入任何集群队列。本地检查是粗颗粒度的。每个用户在其连接到的队列管理器上只需要一个集群传输队列的访问控制概要文件。通过该概要文件，他们可以将消息放入任何集群中任何队列管理器上的任何集群队列。

从 Version 7.1 开始，管理员有另一种方法来设置集群队列的访问控制。您可以使用 `setmqaut` 命令在集群中的任何队列管理器上为集群队列创建安全概要文件。如果在本地打开远程集群队列（仅指定队列名称），那么概要文件将生效。您还可以为远程队列管理器设置概要文件。如果执行此操作，那么队列管理器可以通过提供标准名称来检查打开集群队列的用户的概要文件。

仅当将队列管理器节 `ClusterQueueAccessControl` 更改为 `RQMName` 时，新概要文件才有效。缺省值为 `Xmitq`。您必须为现有应用程序使用集群队列的所有集群队列创建概要文件。如果将节更改为 `RQMName` 而不创建概要文件，那么应用程序可能会失败。

提示: 对访问检入 Version 7.1 的集群队列所作的更改不适用于远程队列。仍会针对本地定义进行访问检查。这些更改意味着您可以遵循相同的方法来配置对集群队列和集群主题的访问检查。

相关概念

[第 237 页的『集群: 使用多个集群传输队列进行应用程序隔离』](#)

您可以在集群中的队列管理器之间隔离消息流。您可以将由不同集群发送方通道传输的消息放入不同的集群传输队列中。您可以在单个集群中使用此方法，也可以将此方法与重叠的集群配合使用。本主题提供了一些示例和一些最佳实践，以指导您选择使用方法。

集群与分布式排队的比较

比较需要定义的组件，以使用分布式排队和集群来连接队列管理器。

如果不使用集群，那么队列管理器是独立的，并使用分布式排队进行通信。如果一个队列管理器需要将消息发送到另一个队列管理器，那么必须定义：

- 传输队列
- 到远程队列管理器的通道

[第 135 页的图 22](#) 显示了分布式排队所需的组件。

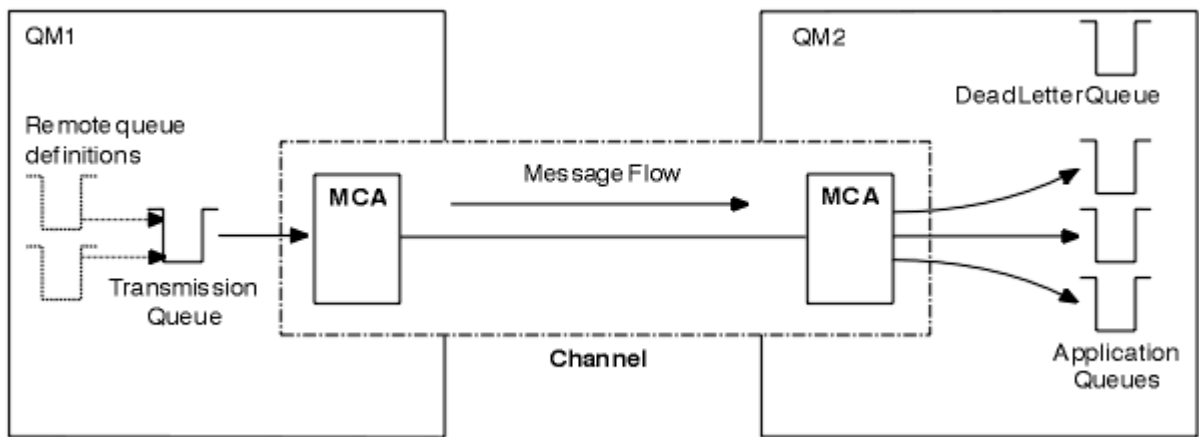


图 22: 分布式 排队

如果对集群中的队列管理器进行分组，那么任何队列管理器上的队列都可供集群中的任何其他队列管理器使用。任何队列管理器都可以在没有显式定义的情况下向同一集群中的任何其他队列管理器发送消息。您不会为每个目标提供通道定义，远程队列定义或传输队列。集群中的每个队列管理器都有一个传输队列，可以从中将消息传输到集群中的任何其他队列管理器。集群中的每个队列管理器仅需要定义：

- 要在其上接收消息的一个集群接收方通道
- 一个集群发送方通道，通过该通道引入自身并了解集群

用于设置集群与分布式排队的定义

查看 第 135 页的图 23，其中显示了四个队列管理器，每个队列管理器具有两个队列。请考虑使用分布式排队连接这些队列管理器所需的定义数。比较将同一网络设置为集群所需的定义数。

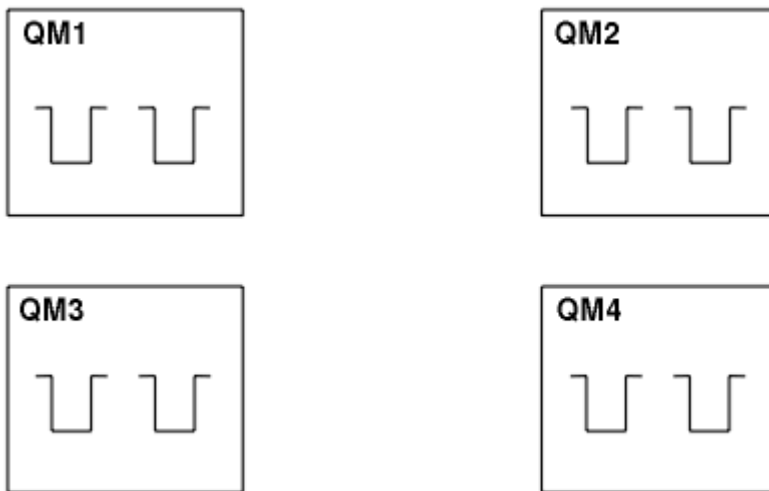


图 23: 由四个队列管理器组成的网络

用于使用分布式排队来设置网络的定义

要使用分布式排队来设置 第 135 页的图 22 中显示的网络，您可能具有以下定义：

表 24: 分布式排队的定义		
描述	每个队列管理器的数目	总数
要将消息发送到其他每个队列管理器的通道的发送方通道定义	3	12
要在其上从每个其他队列管理器接收消息的通道的接收方通道定义	3	12

表 24: 分布式排队的定义 (继续)		
描述	每个队列管理器的数目	总数
传输队列到每个其他队列管理器的传输队列定义	3	12
每个本地队列的本地队列定义	2	8
此队列管理器要将消息放入的每个远程队列的远程队列定义	6	24

您可以使用通用接收方通道定义来减少此数目的定义。每个队列管理器上的最大定义数可能多达 17 个，此网络的定义总数为 68 个。

用于使用集群设置网络的定义

要使用集群设置 [第 135 页](#) 的图 22 中显示的网络，您需要以下定义：

表 25: 集群的定义		
描述	每个队列管理器的数目	总数
要将消息发送到存储库队列管理器的通道的集群发送方通道定义	1	4
要在其上从集群中的其他队列管理器接收消息的通道的集群接收方通道定义	1	4
每个本地队列的本地队列定义	2	8

要设置此队列管理器集群 (具有两个完整存储库)，每个队列管理器上需要四个定义，总共需要十六个定义。您还需要更改两个队列管理器的队列管理器定义，以使它们成为集群的完整存储库队列管理器。

仅需要一个 CLUSSDR 和一个 CLUSRCVR 通道定义。定义集群时，您可以添加或删除队列管理器 (存储库队列管理器除外)，而不会对其他队列管理器造成任何干扰。

使用集群可减少设置包含许多队列管理器的网络所需的定义数。

定义较少，因此发生错误的风险较低：

- 对象名始终匹配，例如发送方/接收方对中的通道名称。
- 通道定义中指定的传输队列名称始终与正确的传输队列定义或远程队列定义中指定的传输队列名称相匹配。
- QREMOTE 定义始终指向远程队列管理器上的正确队列。

设置集群后，您可以将集群队列从一个队列管理器移至集群中的另一个队列管理器，而不必对任何其他队列管理器执行任何系统管理工作。没有可能忘记删除或修改通道，远程队列或传输队列定义。您可以将新的队列管理器添加到集群中，而不会对现有网络造成任何干扰。

集群的组件

集群由队列管理器，集群存储库，集群通道和集群队列组成。

请参阅以下子主题以获取有关每个集群组件的信息：

相关概念

集群

[第 134 页](#) 的『[集群与分布式排队的比较](#)』

比较需要定义的组件，以使用分布式排队和集群来连接队列管理器。

[第 153 页](#) 的『[管理 IBM WebSphere MQ 集群](#)』

您可以创建，扩展和维护 IBM WebSphere MQ 集群。

相关任务

第 132 页的『配置队列管理器集群』

使用本主题中的链接可了解集群如何工作，如何设计集群配置以及如何设置简单集群的示例。

第 153 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。

集群存储库

存储库是有关作为集群成员的队列管理器的信息集合。

存储库信息包括队列管理器名称，它们的位置，它们的通道，它们托管的队列以及其他信息。该信息以消息形式存储在名为 `SYSTEM.CLUSTER.REPOSITORY.QUEUE` 的队列上。队列是其中一个缺省对象。它是在您创建 WebSphere MQ 队列管理器时定义的，但在 WebSphere MQ for z/OS 上除外。

通常，集群中的两个队列管理器保存完整存储库。其余队列管理器都持有部分存储库。

完整存储库和部分存储库

托管集群中每个队列管理器的完整信息集的队列管理器具有完整的存储库。集群中的其他队列管理器具有部分存储库，其中包含完整存储库中的一部分信息。

部分存储库仅包含有关队列管理器需要与之交换消息的那些队列管理器的信息。队列管理器请求更新它们需要的信息，因此如果它发生更改，那么完整存储库队列管理器将向它们发送新信息。在大部分时间内，部分存储库包含队列管理器在集群中需要执行的所有信息。当队列管理器需要某些其他信息时，它可查询完整存储库并更新其部分存储库。队列管理器使用名为 `SYSTEM.CLUSTER.COMMAND.QUEUE` 的队列来请求和接收对存储库的更新。此队列是其中一个缺省对象除外。

集群队列管理器

集群队列管理器是属于集群的队列管理器。

一个队列管理器可以隶属于多个集群。每个集群队列管理器都必须具有在其所属的所有集群中唯一的名称。

集群队列管理器可以托管队列，它会将这些队列通告给集群中的其他队列管理器。集群队列管理器不必托管或宣传任何队列。它可以将消息馈送到集群中，并且只接收显式定向到集群的响应，而不接收已发布的队列的响应。

在 WebSphere MQ for z/OS 中，集群队列管理器可以是队列共享组的成员。在这种情况下，它与同一队列共享组中的其他队列管理器共享其队列定义。

集群队列管理器是自主的。他们可以完全控制他们定义的队列和通道。其他队列管理器 (同一队列共享组中的队列管理器除外) 无法修改其定义。存储库队列管理器不控制集群中其他队列管理器中的定义。它们包含所有定义的完整集合，以便在需要时使用。集群是队列管理器的联合。

在集群队列管理器上创建或更改定义后，会将信息发送到完整存储库队列管理器。稍后将更新集群中的其他存储库。

完整存储库队列管理器

完整存储库队列管理器是一个集群队列管理器，用于保存集群资源的完整表示。要确保可用性，请在每个集群中设置两个或更多完整存储库队列管理器。完整存储库队列管理器接收集群中其他队列管理器发送的信息并更新其存储库。它们相互发送消息以确保它们都与有关集群的新信息保持一致。

队列管理器和存储库

每个集群都有至少一个 (最好是两个) 队列管理器，这些队列管理器保存有关集群中的队列管理器，队列和通道的完整信息存储库。这些存储库还包含来自集群中其他队列管理器的请求，用于更新信息。

其他队列管理器各自保存部分存储库，其中包含有关需要与其通信的队列和队列管理器的子集的信息。队列管理器通过在首次需要访问另一个队列或队列管理器时进行查询来构建其部分存储库。他们请求将有关该队列或队列管理器的任何新信息通知他们。

每个队列管理器都将其存储库信息存储在名为 `SYSTEM.CLUSTER.REPOSITORY.QUEUE` 的队列上的消息中。队列管理器在名为 `SYSTEM.CLUSTER.COMMAND.QUEUE` 的队列上的消息中交换存储库信息。

连接集群的每个队列管理器都定义了一个集群发送方 `CLUSDR` 通道，用于连接到其中一个存储库。它会立即了解集群中的哪些其他队列管理器保存完整存储库。从此，队列管理器可以从任何存储库请求信息。当队列管理器将信息发送到所选存储库时，它还会将信息发送到另一个存储库(如果有)。

当托管完整存储库的队列管理器从与其链接的其中一个队列管理器接收新信息时，将更新完整存储库。还会将新信息发送到另一个存储库，以降低在存储库队列管理器无法使用时将其延迟的风险。由于将两次发送所有信息，因此存储库必须废弃重复项。每个信息项都带有一个序号，存储库使用该序号来标识重复项。通过交换消息使所有存储库保持一致。

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。

集群队列可以是 IBM WebSphere 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

除了在 z/OS 上，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`，那么对于每个集群发送方通道，将通过 `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` 自动创建一个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` 与从较低版本的产品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 之外的每个队列管理器上创建单个 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 队列的定义。

在 z/OS 以外的平台上，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。

要手动创建集群传输队列，请创建本地队列，将 `USAGE` 属性设置为 `XMITQ`，并将 `CLCHNAME` 属性设置为解析为一个或多个集群发送方通道的通用通道名称；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。WebSphere 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
```

相关参考

[ClusterChannel 名称 \(MQCHAR20\)](#)

集群通道

必须为集群中的队列管理器定义集群接收方和集群发送方通道。特殊注意事项适用于完整存储库。

在集群中，消息分布在需要集群接收方通道定义和集群发送方通道定义的特殊类型通道上的集群队列管理器之间。

集群发送方通道: CLUSSDR

在集群中的每个队列管理器上手动定义到完整存储库的集群发送方通道。集群发送方定义使队列管理器能够与集群进行初始联系。它指定队列管理器优先选择向其发送集群信息的完整存储库队列管理器。集群发送

方通道用于通知存储库对队列管理器状态的任何更改。例如，如果添加或删除了队列。它还用于传输消息。

完整存储库队列管理器自身拥有指向彼此的集群发送方通道。它们使用这些通道以就集群状态的更改进行相互通信。

CLUSSDR 通道定义指向的完整存储库不重要。完成初始联系后，将根据需要自动定义其他集群队列管理器对象。队列管理器可以将集群信息发送到每个完整存储库，并将消息发送到每个队列管理器。

在完整存储库队列管理器上创建的 CLUSSDR 定义是特殊定义。由完整存储库交换的所有更新仅在这些通道上流动。管理员显式控制完整存储库的网络。管理员必须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。管理员必须手动在完整存储库队列管理器上创建 CLUSSDR 定义，而不能将其保留为自动定义。

必须仅定义集群发送方通道以将部分存储库连接到完整存储库，或者将两个完整存储库连接到一起。手动配置针对部分存储库或不在集群中的队列管理器的 CLUSSDR 通道会导致发出错误消息，例如 AMQ9427 和 AMQ9428。

虽然作为临时情况 (例如修改完整存储库的位置)，这有时是不可避免的，但应该尽快删除不正确的定义，以停止发出这些错误。

集群接收方通道: CLUSRCVR

集群接收方通道定义定义了集群队列管理器可从集群中的其他队列管理器接收消息的通道的结束。

集群接收方通道还可以传递有关以本地存储库为目标的集群信息的信息。通过定义集群接收方通道，队列管理器将显示它可用于接收消息的其他集群队列管理器。每个集群队列管理器至少需要有一个集群接收方通道。

CLUSRCVR 定义使其他队列管理器能够自动定义相应的 CLUSSDR 通道定义。

相关概念

第 140 页的『集群通道的自动定义』

队列管理器必须具有集群发送方通道的定义，然后才能将消息发送到远程目标。通过使其初始 CLUSSDR 和 CLUSRCVR 定义向集群引入队列管理器后，WebSphere MQ 会在需要时自动生成集群发送方通道定义。无法修改自动定义的集群发送方通道。您可以使用通道自动定义出口来修改其行为。

集群通道的自动定义

队列管理器必须具有集群发送方通道的定义，然后才能将消息发送到远程目标。通过使其初始 CLUSSDR 和 CLUSRCVR 定义向集群引入队列管理器后，WebSphere MQ 会在需要时自动生成集群发送方通道定义。无法修改自动定义的集群发送方通道。您可以使用通道自动定义出口来修改其行为。

当定义了通道的集群发送方端和集群接收方端时，将启动该通道。自动定义的通道将保持活动状态，直到不再需要该通道并使用正常断开连接时间间隔规则将其关闭为止。

自动定义的集群发送方通道从接收队列管理器上的相应集群接收方通道定义中获取其属性。即使存在手动定义的集群发送方通道，也会自动修改其属性以确保它们与相应的集群接收方定义匹配。例如，假设您定义 CLUSRCVR 而不在 CONNAME 参数中指定端口号，并手动定义指定端口号的 CLUSSDR。当自动定义的 CLUSSDR 替换手动定义的端口号时，端口号 (取自 CLUSRCVR) 将变为空白。将使用缺省端口号，并且通道将发生故障。

不能修改自动定义的集群发送方定义。

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)
```

要显示与您创建的 CLUSRCVR 通道定义对应的自动定义的 CLUSSDR 通道的状态，请使用以下命令：

```
DISPLAY CHSTATUS(channelName)
```

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 WebSphere MQ 通道自动定义出口。可以在集群环境中使用通道自动定义出口来执行以下操作：

- 定制通信定义，即 SNA LU6.2 名称
- 添加或删除其他出口，例如，安全出口
- 更改通道出口的名称。必须更改 CLUSSDR 通道出口的名称，因为 CLUSSDR 通道出口名称是从 CLUSRCVR 通道定义自动生成的。自动生成的名称可能是错误的，如果通道的两端位于不同的平台上，那么几乎肯定是错误的。在不同的平台上，出口名称的格式不同。例如，在 Windows 上为 SCYEXIT('drive:\path\library(secexit)')。

除 z/OS 以外的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，将使用截断为 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

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将特定网卡用于出站通信，那么 LOCLADDR 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，那么必须编写通道自动定义出口，以便将 LOCLADDR 值强制转换为其自动定义的任何 CLUSSDR 通道，并且必须在手动定义的 CLUSSDR 通道中指定该值。

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。

将端口号或端口范围放在 CLUSRCVR 通道的 LOCLADDR 中 (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)

distributed 在分布式平台上，可以设置缺省本地地址值，该值将用于所有未定义本地地址的发送方通道。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。

自动定义的集群发送方通道定义不是实际通道对象。在 z/OS 以外的平台上，OAM (对象权限管理器) 不知道它们是否存在。如果尝试在自动定义的集群发送方通道上发出启动，停止，ping，重置或解析命令，那么 OAM 会检查您是否有权在集群的集群接收方通道上执行相同的操作。

如果集群需要使用 PROPCTL 从 WebSphere MQ Version 7 队列管理器到较低级别的 WebSphere MQ 上的队列管理器的消息中除去应用程序头 (例如 RFH2)，那么必须编写通道自动定义出口，将 PROPCTL 强制设置为值 NONE。该出口是必需的，因为集群发送方通道具有基于相应集群接收方通道的定义。由于较早级别的集群接收方通道没有 PROPCTL 属性，因此自动集群发送方通道会将该属性设置为 COMPAT。无论在手动集群发送方通道上设置了什么，该属性都将设置为 COMPAT。

相关参考

[本地地址 \(LOCLADDR\)](#)

缺省集群对象

使用 WebSphere MQ 集群时创建缺省集群对象。它们包括在定义队列管理器上除外。

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。

SYSTEM.CLUSTER.COMMAND.QUEUE

集群中的每个队列管理器都有一个名为 SYSTEM.CLUSTER.COMMAND.QUEUE 的本地队列，用于将消息传输到完整存储库。此消息包含有关队列管理器的任何新信息或已更改的信息，或者有关其他队列管理器的任何信息请求。SYSTEM.CLUSTER.COMMAND.QUEUE 通常为空。

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.REPOSITORY.QUEUE

集群中的每个队列管理器都有一个名为 `SYSTEM.CLUSTER.REPOSITORY.QUEUE` 的本地队列。此队列用于存储所有完整存储库信息。此队列通常不为空。

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` (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`。

SYSTEM.DEF.CLUSRCVR

每个集群都有一个名为 `SYSTEM.DEF.CLUSRCVR` 的缺省 `CLUSRCVR` 通道定义。`SYSTEM.DEF.CLUSRCVR` 用于为您在集群中的队列管理器上创建集群接收方通道时未指定的任何属性提供缺省值。

SYSTEM.DEF.CLUSSDR

每个集群都有一个名为 `SYSTEM.DEF.CLUSSDR` 的缺省 `CLUSSDR` 通道定义。`SYSTEM.DEF.CLUSSDR` 用于为您在集群中的队列管理器上创建集群发送方通道时未指定的任何属性提供缺省值。

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。

显示集群发送方通道时，您会看到它与传输队列相关联。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。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)
CONNNAME(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 143 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 143 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 144 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。直到 Version 7.5 集群连接共享了单集群传输队列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。从 Version 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。请参阅第 240 页的『集群：规划如何配置集群传输队列』和第 171 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 143 页的图 24 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。

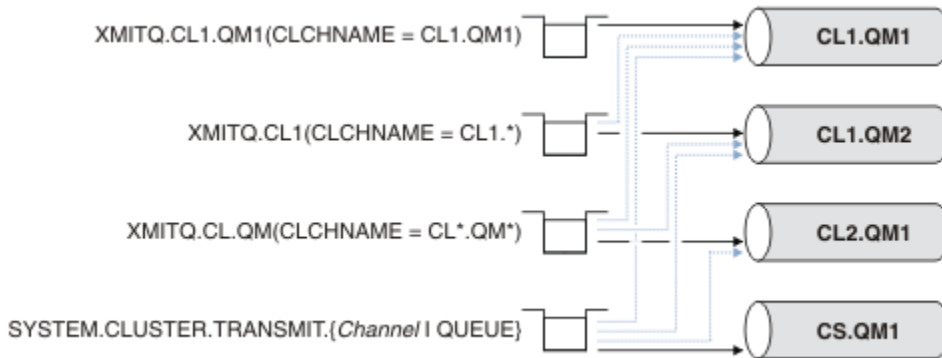


图 24: 传输队列/集群发送方通道优先顺序

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 143 页的图 24 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 143 页的图 24 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。通过一个队列，您有一组要设置和查询的集群传输队列属性，通过多个队列，您有多个集合。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，请参阅第 134 页的『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。其中，选择连接到目标的集群发送方通道，工作负载均衡算法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 146 页的图 25 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。

如果您打算分隔消息流量，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分隔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 146 页的图 25 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。

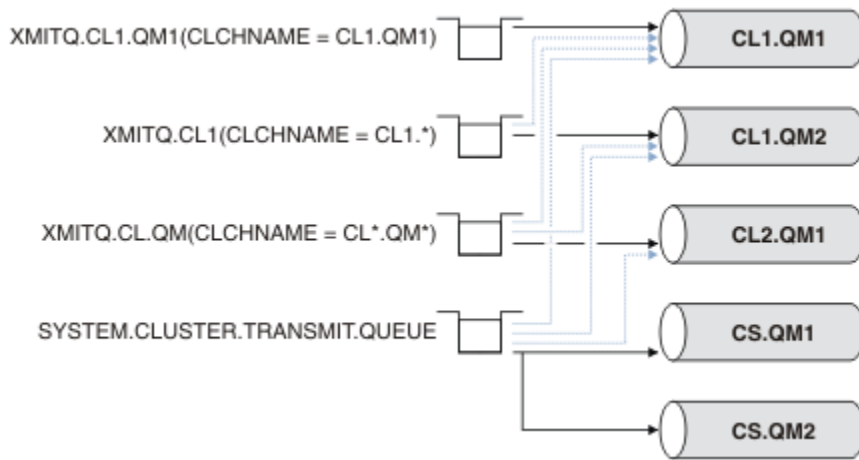


图 25: 多个集群发送方通道

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 144 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

[第 237 页的『集群: 使用多个集群传输队列进行应用程序隔离』](#)

您可以在集群中的队列管理器之间隔离消息流。您可以将由不同集群发送方通道传输的消息放入不同的集群传输队列中。您可以在单个集群中使用此方法，也可以将此方法与重叠的集群配合使用。本主题提供了一些示例和一些最佳实践，以指导您选择使用方法。

[第 335 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

[第 240 页的『集群: 规划如何配置集群传输队列』](#)

将指导您选择集群传输队列。您可以配置一个公共缺省队列，单独的缺省队列或手动定义的队列。配置多个集群传输队列适用于除 z/OS 以外的平台。

[第 181 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 164 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 168 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 171 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。

如何选择集群队列管理器以保存完整存储库

在每个集群中，必须至少选择一个队列管理器，最好选择两个队列管理器以保存完整存储库。除了最例外的情况外，两个完整的存储库足以应对所有情况。如果可能，请选择在稳健且永久连接的平台托管的队列管理器，这些队列管理器不会发生重合的中断，并且在地理位置上处于中心位置。另外，请考虑将系统专用为完整存储库主机，而不要将这些系统用于任何其他任务。

完整存储库是保持集群状态完整的队列管理器。要共享此信息，每个完整存储库都由 CLUSSDR 通道 (及其相应的 CLUSRCVR 定义) 连接到集群中的每个其他完整存储库。您必须手动定义这些通道。



图 26: 两个已连接的完整存储库。

集群中的每个其他队列管理器都会在部分存储库中维护其当前了解的集群状态的图片。这些队列管理器使用任何两个可用的完整存储库发布有关自身的信息，并请求有关其他队列管理器的信息。如果所选完整存储库不可用，那么将使用另一个完整存储库。当所选的完整存储库再次变为可用时，它将从其他存储库收集最新的新信息和已更改的信息，以便它们保持同步。如果所有完整存储库都不起作用，那么其他队列管理器将使用其部分存储库中的信息。但是，它们仅限于使用它们所拥有的信息；无法处理新信息和更新请求。当完整存储库重新连接到网络时，将交换消息以更新所有存储库 (包括完整存储库和部分存储库)。

规划完整存储库的分配时，请包括以下注意事项：

- 选择用于保存完整存储库的队列管理器需要可靠且受管。选择在健壮且永久连接的平台托管的队列管理器。
- 请考虑托管完整存储库的系统的计划中断，并确保这些系统不会发生重合的中断。
- 考虑网络性能：选择在地理位置上处于中心位置的队列管理器，或者选择与集群中的其他队列管理器共享同一系统的队列管理器。
- 请考虑队列管理器是否是多个集群的成员。使用同一队列管理器来托管多个集群的完整存储库在管理上可能很方便，前提是此优势与您期望队列管理器的繁忙程度相平衡。
- 请考虑将某些系统专用于仅包含完整存储库，而不将这些系统用于任何其他任务。这将确保这些系统只需要维护队列管理器配置，并且不会从维护其他业务应用程序的服务中除去。它还确保维护存储库的任务不会与系统资源的应用程序竞争。这在大型集群 (例如，包含超过 1000 个队列管理器的集群) 中特别有用，在这些集群中，完整存储库在维护集群状态方面的工作负载要高得多。

可能有两个以上完整存储库，但很少建议使用。虽然对象定义 (即，队列，主题和通道) 流向所有可用的完整存储库，但请求仅从部分存储库流向最多两个完整存储库。这意味着当定义了两个以上的完整存储库，并且任何两个完整存储库变为不可用时，某些部分存储库可能不会收到它们期望的更新。请参阅 [MQ 集群：为什么只有两个完整存储库？](#)

您可能发现定义两个以上完整存储库很有用的一种情况是在将现有完整存储库迁移到新硬件或新队列管理器时。在这种情况下，您应该先引入替换完整存储库，并确认它们已完全填充，然后再移除先前的完整存储库。每当添加完整存储库时，请记住必须使用 CLUSSDR 通道将其直接连接到其他所有完整存储库。

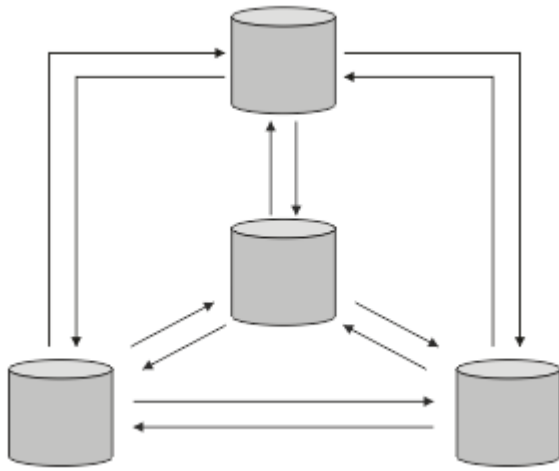


图 27: 两个以上已连接的完整存储库

相关信息

[MQ 集群: 为什么只有两个完整存储库?](#)

[MQ 集群有多大?](#)

组织集群

选择要链接到哪个完整存储库的队列管理器。请考虑性能影响，队列管理器版本以及是否需要多个 CLUSSDR 通道。

在选择队列管理器以保存完整存储库之后，您需要决定要链接到哪个完整存储库的队列管理器。CLUSSDR 通道定义将队列管理器链接到完整存储库，从中可以发现集群中的其他完整存储库。从此，队列管理器将消息发送到任意两个完整存储库。它始终首先尝试使用其具有 CLUSSDR 通道定义的通道。您可以选择将队列管理器链接到任一完整存储库。在选择时，请考虑配置的拓扑以及队列管理器的物理或地理位置。

由于所有集群信息都将发送到两个完整存储库，因此可能存在您想要生成第二个 CLUSSDR 通道定义的情况。您可以在集群中定义第二个 CLUSSDR 通道，该集群具有分布在广泛区域中的许多完整存储库。然后，您可以控制将信息发送到哪个两个完整存储库。

集群命名约定

请考虑使用用于标识队列管理器所属集群的命名约定来命名同一集群中的队列管理器。对通道名称使用类似的命名约定，并对其进行扩展以描述通道特征。

命名 MQ 集群时的最佳实践

虽然集群名称最多可以有 48 个字符，但在将命名约定应用于其他对象时，相对较短的集群名称很有用。请参阅第 149 页的『选择集群通道名称时的最佳实践』。

在选择集群名称时，通常有助于表示集群的“用途”（可能是长期存在）而不是“内容”。例如，“B2BPROD”或“ACTTEST”而不是“QM1_QM2_QM3_CLUS”。

选择集群队列管理器名称时的最佳实践

如果要从头开始创建新集群及其成员，请考虑队列管理器的命名约定，以反映其集群使用情况。每个队列管理器都必须具有不同的名称。但是，您可以为集群中的队列管理器提供一组类似的名称，以帮助识别和记住逻辑分组（例如，“ACTTQM1，ACTTQM2”）。

相对较短的队列管理器名称（例如，少于 8 个字符）可帮助您选择对通道名称使用下一节中描述的约定或类似的约定。

选择集群通道名称时的最佳实践

由于队列管理器和集群的名称最多可包含 48 个字符，并且通道名称限制为 20 个字符，因此在第一个命名对象时请注意避免在项目中途更改命名约定 (请参阅先前部分)。

定义通道时，请记住，在集群中的任何队列管理器上自动创建的集群发送方通道会从集群中的接收队列管理器上配置的相应集群接收方通道中获取其名称，因此这些通道必须是唯一的，并且在集群中的远程队列管理器上有意义。

一种常见方法是使用以集群名称开头的队列管理器名称。例如，如果集群名称为 CLUSTER1，队列管理器为 QM1，QM2，那么集群接收方通道为 CLUSTER1.QM1，CLUSTER1.QM2。

如果通道具有不同的优先级或使用不同的协议，那么可以扩展此约定。例如：

- CLUSTER1.QM1.S1
- CLUSTER1.QM1.N3
- CLUSTER1.QM1.T4

在此示例中，S1 可能是第一个 SNA 通道，N3 可能是网络优先级为 3 的 NetBIOS 通道，T4 可能是使用 IPV4 网络的 TCP IP。

命名共享通道定义

可以跨多个集群共享单个通道定义，在这种情况下，此处建议的命名约定需要修改。但是，如 [管理通道定义](#) 中所述，在任何情况下，通常最好为每个集群定义离散通道。

较旧的通道命名约定

在集群环境之外，使用 "FROMQM.TO.TARGETQM" 命名约定在历史上是常见的，因此您可能会发现现有集群使用了类似的内容 (例如 CLUSTER.TO.TARGET)。建议不要将此作为新集群命名方案的一部分，因为这将进一步减少可用字符以在通道名称中传递 "有用" 信息。

重叠集群

重叠的集群提供了额外的管理功能。使用名称列表来减少管理重叠集群所需的命令数。

您可以创建重叠的集群。您可以定义重叠集群的原因有很多；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。

在第 150 页的图 28 中，队列管理器 STF2 是这两个集群的成员。当队列管理器是多个集群的成员时，您可以利用名称列表来减少所需的定义数。名称列表包含名称列表，例如，集群名称。您可以创建命名集群的名称列表。在 STF2 的 ALTER QMGR 命令上指定名称列表，以使其成为两个集群的完整存储库队列管理器。

如果网络中有多个集群，那么必须为它们提供不同的名称。如果两个具有相同名称的集群曾经合并，那么无法再次将它们分开。给集群和通道不同的名称也是一个好主意。当您查看 DISPLAY 命令的输出时，可以更轻松地对它们进行区分。队列管理器名称在集群中必须是唯一的，才能使其正常工作。

定义服务类

想象一下，有一个大学，每个工作人员和每个学生都有一个队列管理器。工作人员之间的消息是在具有高优先级和高带宽的通道上传输。学生之间的信息是在更便宜，更慢的渠道上旅行。您可以使用传统的分布式排队技术来设置此网络。WebSphere MQ 通过查看目标队列名称和队列管理器名称来选择要使用的通道。

要明确区分职员和学生，您可以将他们的队列管理器分组到两个集群中，如第 150 页的图 28 中所示。WebSphere MQ 仅通过人员集群中定义的通道将消息移至该集群中的会议队列。学生集群中 gossip 队列的消息将通过该集群中定义的通道，并接收相应的服务类。

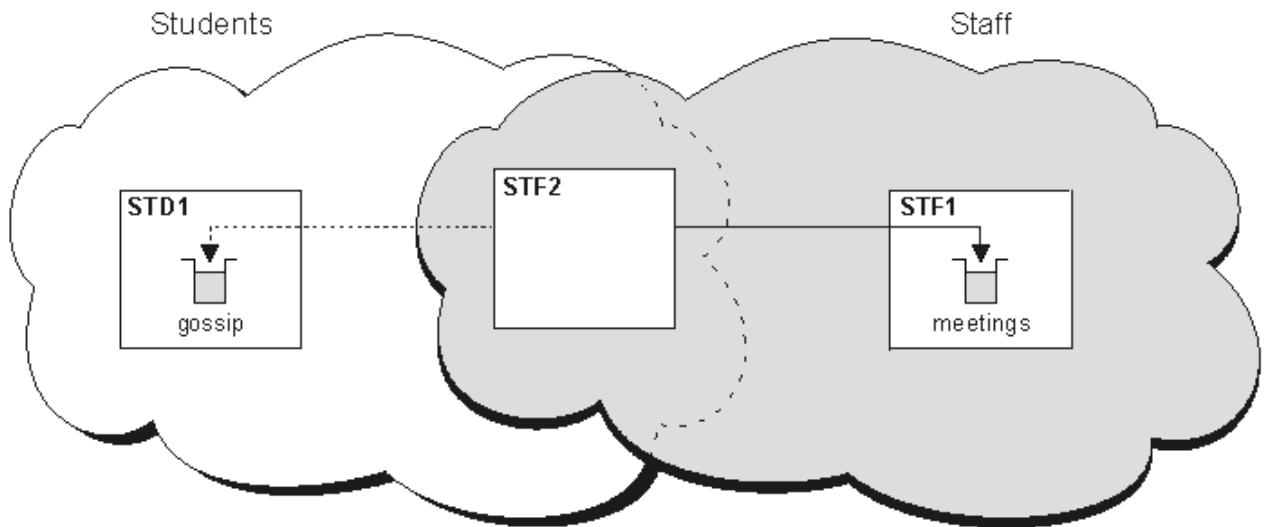


图 28: 服务等级

集群提示

在使用集群之前，您可能需要对系统或应用程序进行一些更改。与分布式排队的行为既有相似之处，也有不同之处。

- WebSphere MQ Explorer 无法直接管理版本低于 6.0 的 WebSphere MQ for z/OS 队列管理器。
- 必须将手动配置定义添加到集群外部的队列管理器，以便它们访问集群队列。
- 如果合并具有相同名称的两个集群，那么不能再次将其分隔开。因此，建议为所有集群提供唯一名称。
- 如果消息到达队列管理器，但那里没有要接收该消息的队列，那么会将该消息放在死信队列上。如果没有死信队列，那么通道将失败并重试。死信队列的使用与分布式排队相同。
- 维护持久消息的完整性。消息不会因使用集群而重复或丢失。
- 使用集群可减少系统管理。集群使您能够轻松地将更大的网络与更多的队列管理器连接起来，而不需要您考虑使用分布式排队。如果尝试在集群中的每个队列管理器之间启用通信，那么可能会消耗过多的网络资源。
- 如果使用 WebSphere MQ Explorer (它以树结构显示队列管理器)，那么大型集群的视图可能很繁琐。
- WebSphere MQ Explorer 可以使用 WebSphere MQ for z/OS V 6 或更高版本上的存储库队列管理器来管理集群。您无需在单独的系统上指定其他存储库。对于较早版本的 WebSphere MQ on z/OS，WebSphere MQ Explorer 无法管理具有存储库队列管理器的集群。您必须在 WebSphere MQ Explorer 可以管理的系统上指定其他存储库。
- 分发列表的目的是使用单个 MQPUT 命令将同一消息发送到多个目标。在 WebSphere MQ for AIX, IBM i, HP-UX, Solaris, Linux 和 Windows 上支持分发列表。您可以将分发列表与队列管理器集群配合使用。在集群中，所有消息在 MQPUT 时展开。在网络流量方面的优势，并不像在非集群环境中那么大。分发列表的优点是不需要手动定义众多通道和传输队列。
- 如果要使用集群来平衡工作负载，请检查应用程序。查看它们是否要求消息由特定队列管理器处理还是按特定顺序处理。此类应用程序据说具有消息亲缘关系。您可能需要先修改应用程序，然后才能在复杂集群中使用这些应用程序。
- 您可以选择在 MQOPEN 上使用 MQOO_BIND_ON_OPEN 选项，以强制将消息发送到特定目标。如果目标队列管理器不可用，那么直到队列管理器再次可用之后才会传递消息。由于存在重复的风险，因此不会将消息路由到另一个队列管理器。
- 如果队列管理器要托管集群存储库，那么您需要知道其主机名或 IP 地址。在加入集群的其他队列管理器上创建 CLUSSDR 定义时，必须在 CONNAME 参数中指定此信息。如果使用 DHCP，那么 IP 地址可能会更改，因为 DHCP 可以在每次重新启动系统时分配新的 IP 地址。因此，不得在 CLUSSDR 定义中指定 IP 地址。即使所有 CLUSSDR 定义都指定了主机名而不是 IP 地址，这些定义仍然不可靠。DHCP 不一定使用新地址更新主机的 DNS 目录条目。如果必须将队列管理器指定为使用 DHCP 的系统上的完整存储库，请安装保证 DNS 目录保持最新的软件。

- 请勿使用通用名称，例如 VTAM 通用资源或动态域 名称 服务器 (DDNS) 通用名称作为通道的连接名称。如果您这样做，那么您的通道可能会连接到与预期不同的队列管理器。
- 只能从本地集群队列获取消息，但可以将消息放入集群中的任何队列。如果打开队列以使用 MQGET 命令，那么队列管理器将打开本地队列。
- 如果设置了简单的 WebSphere MQ 集群，那么不需要更改任何应用程序。应用程序可以在 MQOPEN 调用上命名目标队列，并且不需要了解队列管理器的位置。如果为工作负载管理设置集群，那么必须复审应用程序并根据需要对其进行修改。
- 您可以使用 DISPLAY CHSTATUS 和 DISPLAY QSTATUS **runmqsc** 命令来查看通道或队列的当前监视和状态数据。监控信息可用于帮助评估系统的性能和运行状况。监视由队列管理器，队列和通道属性控制。可以使用 MONACLS 队列管理器属性来监视自动定义的集群发送方通道。

相关概念

集群

集群工作方式

第 134 页的『[集群与分布式排队的比较](#)』

比较需要定义的组件，以使用分布式排队和集群来连接队列管理器。

第 136 页的『[集群的组件](#)』

集群由队列管理器，集群存储库，集群通道和集群队列组成。

第 153 页的『[管理 IBM WebSphere MQ 集群](#)』

您可以创建，扩展和维护 IBM WebSphere MQ 集群。

相关任务

第 132 页的『[配置队列管理器集群](#)』

使用本主题中的链接可了解集群如何工作，如何设计集群配置以及如何设置简单集群的示例。

第 153 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。

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，以及 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 WebSphere MQ 平台上，可使用 START LISTENER 命令启动侦听器。在 WebSphere MQ for IBM i，Windows 和 UNIX and Linux 系统上，可以在队列管理器的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。

过程

1. 启动通道启动程序。



IBM WebSphere MQ (对于 Windows 和 UNIX and Linux 系统)

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。

Windows

IBM WebSphere MQ (对于 Windows)

使用 WebSphere MQ 提供的通道侦听器程序或操作系统提供的工具。

要启动 WebSphere MQ 通道侦听器，请使用 `RUNMQLSR` 命令。例如：

```
RUNMQLSR -t tcp -p 1414 -m QM1
```

UNIX Linux

UNIX and Linux 系统上的 IBM WebSphere MQ

使用 WebSphere MQ 提供的通道侦听器程序或操作系统提供的工具；例如，用于 TCP 通信的 `inetd`。

要启动 WebSphere MQ 通道侦听器，请使用 `runmqclsr` 命令。例如：

```
runmqclsr -t tcp -p 1414 -m QM1
```

要使用 `inetd` 来启动通道，请配置两个文件：

- 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 WebSphere 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。

- 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 WebSphere 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：

在 AIX 上：

```
refresh -s inetd
```

在 HP-UX 上：

```
inetd -c
```

在 Solaris 或 Linux 上：

- 使用以下命令查找 `inetd` 的进程标识：

```
ps -ef | grep inetd
```

- 运行相应的命令，如下所示：

- 对于 Solaris 9 和 Linux：

```
kill -1 inetd processid
```

- 对于 Solaris 10 或更高版本：

```
inetconv
```

队列管理器存储库保留信息的时间长度？

队列管理器存储库将信息保留 30 天。自动进程高效地刷新正在使用的信息。

当队列管理器发出有关自身的一些信息时，完整和部分存储库队列管理器会将该信息存储 30 天。例如，当队列管理器发布创建新队列的广告时，将发送信息。为了防止此信息到期，队列管理器将在 27 天后自动重新发送有关自身的所有信息。如果部分存储库在 30 天生存期内发送新的信息部分请求，那么到期时间仍为原来的 30 天。

当信息到期时，不会立即将其从存储库中除去。而是举行 60 天的宽限期。如果在宽限期内未收到任何更新，那么将除去该信息。宽限期允许队列管理器可能在到期日期暂时停止服务。如果队列管理器与集群断开连接超过 90 天，那么它将停止成为集群的一部分。但是，如果它重新连接到网络，那么它将再次成为集群的一部分。完整存储库不使用已到期的信息来满足来自其他队列管理器的新请求。

同样，当队列管理器从完整存储库发送对最新信息的请求时，该请求将持续 30 天。27 天后，IBM WebSphere MQ 将检查请求。如果在 27 天内引用了该参数，那么将自动刷新该参数。如果没有，那么它将保留为到期，并由队列管理器刷新(如果再次需要)。到期请求会阻止来自休眠队列管理器的信息请求的累积。

注：对于大型集群，如果许多队列管理器同时自动重新发送有关其自身的所有信息，那么这可能具有破坏性。请参阅第 256 页的『在大型集群中刷新可能会影响集群的性能和可用性』。

相关概念

第 256 页的『[集群：使用 REFRESH CLUSTER 最佳实践](#)』

使用 **REFRESH CLUSTER** 命令可废弃有关集群的所有本地保存的信息，并从集群中的完整存储库重建该信息。您不应该使用此命令，除非在特殊情况下。如果确实需要使用它，那么有关如何使用它有特殊注意事项。此信息是基于测试和客户反馈的指南。

管理 IBM WebSphere MQ 集群

您可以创建，扩展和维护 IBM WebSphere MQ 集群。

有关如何管理 IBM WebSphere MQ 集群的详细信息，请参阅以下子主题：

相关概念

集群

集群工作方式

第 134 页的『[集群与分布式排队的比较](#)』

比较需要定义的组件，以使用分布式排队和集群来连接队列管理器。

第 136 页的『[集群的组件](#)』

集群由队列管理器，集群存储库，集群通道和集群队列组成。

相关任务

第 132 页的『[配置队列管理器集群](#)』

使用本主题中的链接可了解集群如何工作，如何设计集群配置以及如何设置简单集群的示例。

第 153 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。

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 WebSphere 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。
- 有关可帮助您了解设置集群所执行步骤的背景信息，请参阅第 138 页的『[集群队列](#)』，[通道](#) 和 [侦听器](#)。

关于此任务

您正在为链式商店设置新的 IBM WebSphere 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。

第 154 页的图 29 显示了此集群的外观。

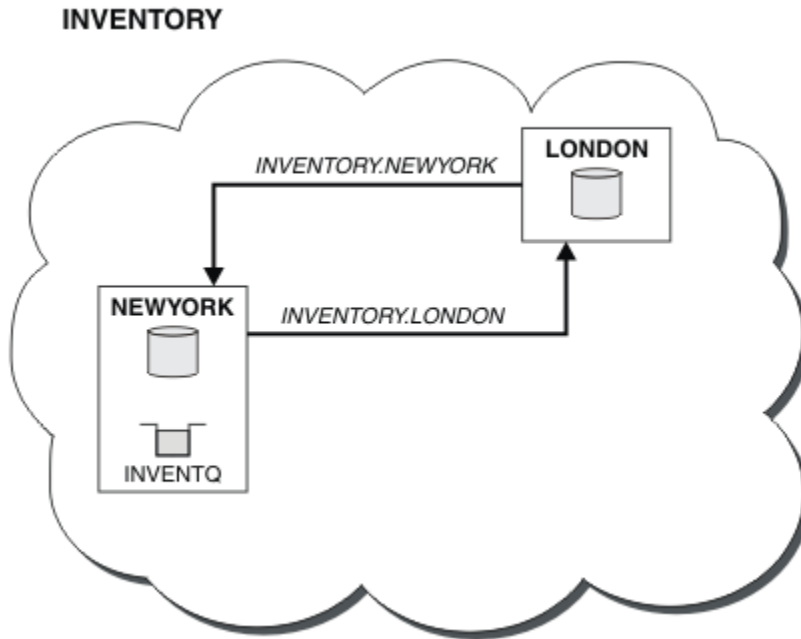


图 29: 具有两个队列管理器的 *INVENTORY* 集群

您可以配置集群中不在 z/OS 上的每个队列管理器，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。

过程

- [第 155 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 157 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 159 页的『在 z/OS 上使用 LU 6.2 设置集群』](#)
- [第 161 页的『验证集群』](#)

结果

第 154 页的图 29 显示了此任务的 *INVENTORY* 集群设置。

显然，*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。

相关概念

集群

集群工作方式

[第 134 页的『集群与分布式排队的比较』](#)

比较需要定义的组件，以使用分布式排队和集群来连接队列管理器。

第 136 页的『集群的组件』

集群由队列管理器，集群存储库，集群通道和集群队列组成。

第 153 页的『管理 IBM WebSphere MQ 集群』

您可以创建，扩展和维护 IBM WebSphere MQ 集群。

相关任务

第 132 页的『配置队列管理器集群』

使用本主题中的链接可了解集群如何工作，如何设计集群配置以及如何设置简单集群的示例。

使用 TCP/IP 设置集群，每个队列管理器具有单个传输队列

开始之前

- 在 AIX, HP-UX, IBM i, Linux, Solaris, and Windows 上，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 AIX, HP-UX, IBM i, Linux, Solaris, and Windows 上设置集群。

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 162 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 WebSphere 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: Websphere 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)
```

注：定义侦听器时，如果在 CONNAME 字段中使用 IP 地址，并且端口号不是缺省端口 (1414)，那么应定义端口号。例如：

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(1415)
```

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，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。CLUSRCVR 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 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)，并且集群未包含 z/OS 上的队列管理器，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。

在集群中的每个队列管理器上，定义一个集群发送方通道。队列管理器将消息发送到集群发送方通道上的其中一个完整存储库队列管理器。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。它们都必须具有指向另一个队列管理器上定义的 CLUSRCVR 通道的 CLUSSDR 定义。

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。一旦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，就会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。

使用每个队列管理器的多个传输队列在 TCP/IP 上设置集群

关于此任务

执行以下步骤以使用传输协议 TCP/IP 在 AIX, HP-UX, IBM i, Linux, Solaris, and Windows 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 164 页的『将队列管理器添加到集群: 单独的传输队列』。无法在 z/OS 上设置队列管理器以使用单独的集群传输队列。

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 162 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 WebSphere 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: Websphere 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 162 页的『将队列管理器添加至集群』和 第 164 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)
```

注：定义侦听器时，如果在 CONNAME 字段中使用 IP 地址，并且端口号不是缺省端口 (1414)，那么应定义端口号。例如：

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(1415)
```

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，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。CLUSRCVR 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 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), 并且集群未包含 z/OS 上的队列管理器, 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。

在集群中的每个队列管理器上, 定义一个集群发送方通道。队列管理器将消息发送到集群发送方通道上的其中一个完整存储库队列管理器。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。它们都必须具有指向另一个队列管理器上定义的 CLUSRCVR 通道的 CLUSSDR 定义。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。一旦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, 就会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。

在 z/OS 上使用 LU 6.2 设置集群

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 162 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: Websphere 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。

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，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。CLUSRCVR 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)
CONNNAME(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)
CONNNAME(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)
CONNNAME(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。

在集群中的每个队列管理器上，定义一个集群发送方通道。队列管理器将消息发送到集群发送方通道上的其中一个完整存储库队列管理器。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。它们都必须具有指向另一个队列管理器上定义的 CLUSRCVR 通道的 CLUSSDR 定义。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。一旦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，就会启动集群发送方通道。


```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNNAME(CPIC) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNNAME(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)
CONNNAME(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。

验证集群

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( )
CONNNAME(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)
CONNNAME(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) 将消息描述符中的 *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) 从消息描述符检索 *ReplyToQ* 名称。
 - e) 将 *ReplyToQ* 名称放在对象描述符的 *ObjectName* 字段中。
 - 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。

方案:

- 如第 153 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)
CONNNAME(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。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必须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.LONDON 的通道进行以下定义。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

4. 可选：如果此队列管理器正在重新加入集群，请完成一些额外的步骤。

- a) 如果要将队列管理器添加到先前已从同一集群中除去的集群，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：

- i) 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，当集群正在运行中时，使用 **REFRESH CLUSTER** 命令可能会破坏该集群，并且将在 27 天的时间间隔之后，集群对象才会再次自动向所有相关队列管理器发送状态更新。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。

- ii) 重新启动 CLUSSDR 通道 (例如，使用 `START CHANNEL` 命令)。

- iii) 重新启动 CLUSRCVR 通道。

- b) 如果集群是发布/预订集群，并且重新连接的队列管理器具有预订，请发出以下命令以确保在集群中正确同步代理预订：

```
REFRESH QMGR TYPE(PROXYSUB)
```

结果

下图显示了此任务设置的集群。

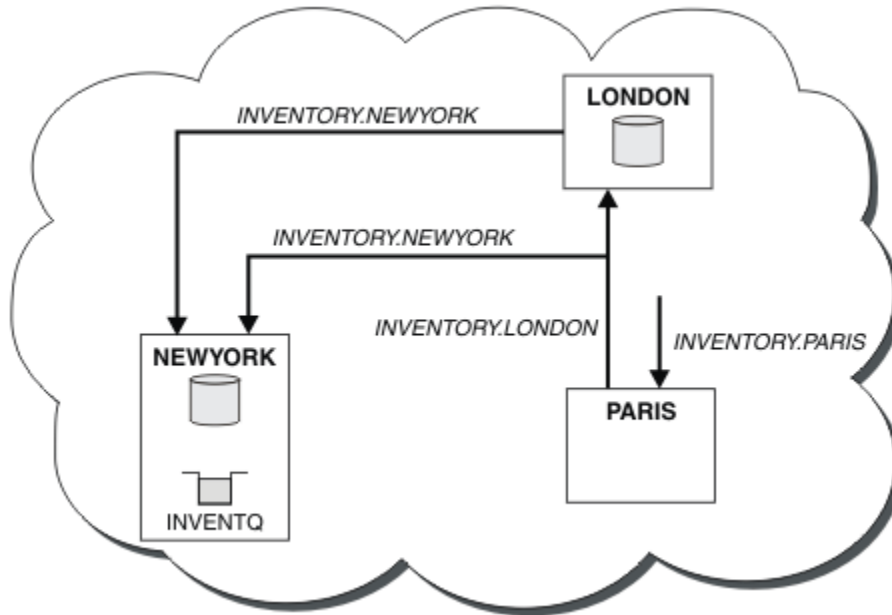


图 30: 具有三个队列管理器的 *INVENTORY* 集群

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

- 队列管理器是在 z/OS 以外的平台上定义的。
- 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 153 页的『设置新集群』。

关于此任务

此任务是第 162 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` 与从较低版本的产品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。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。

4. 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必须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.LONDON 的通道进行以下定义。

```
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。

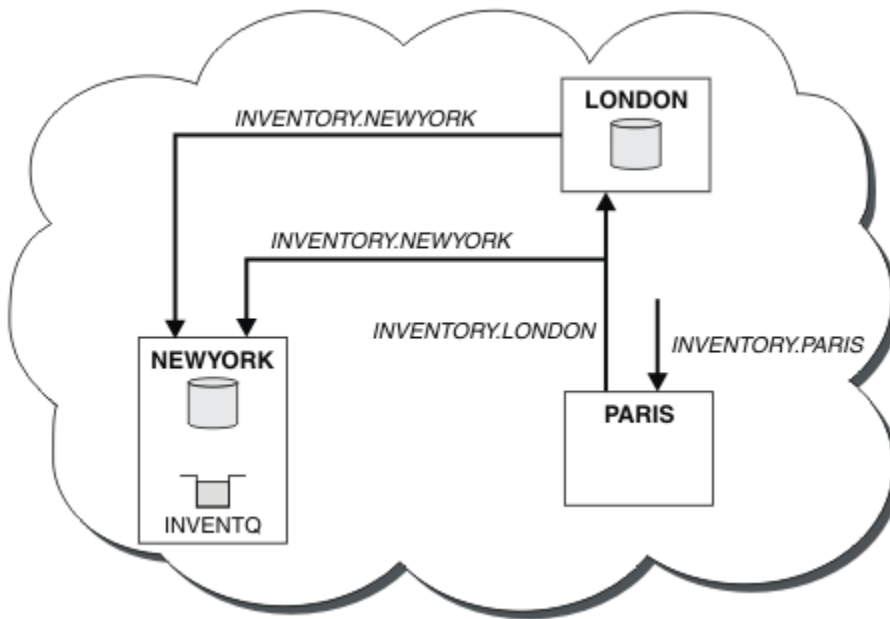


图 31: 具有三个队列管理器的 INVENTORY 集群

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。

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 [第 181 页的『使用网关队列管理器创建两个重叠的集群』](#) 中的 [第 181 页的图 37](#) 中显示的重叠集群。

关于此任务

该解决方案使用分布式队列来分隔 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。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 167 页的图 32](#) 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 168 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。

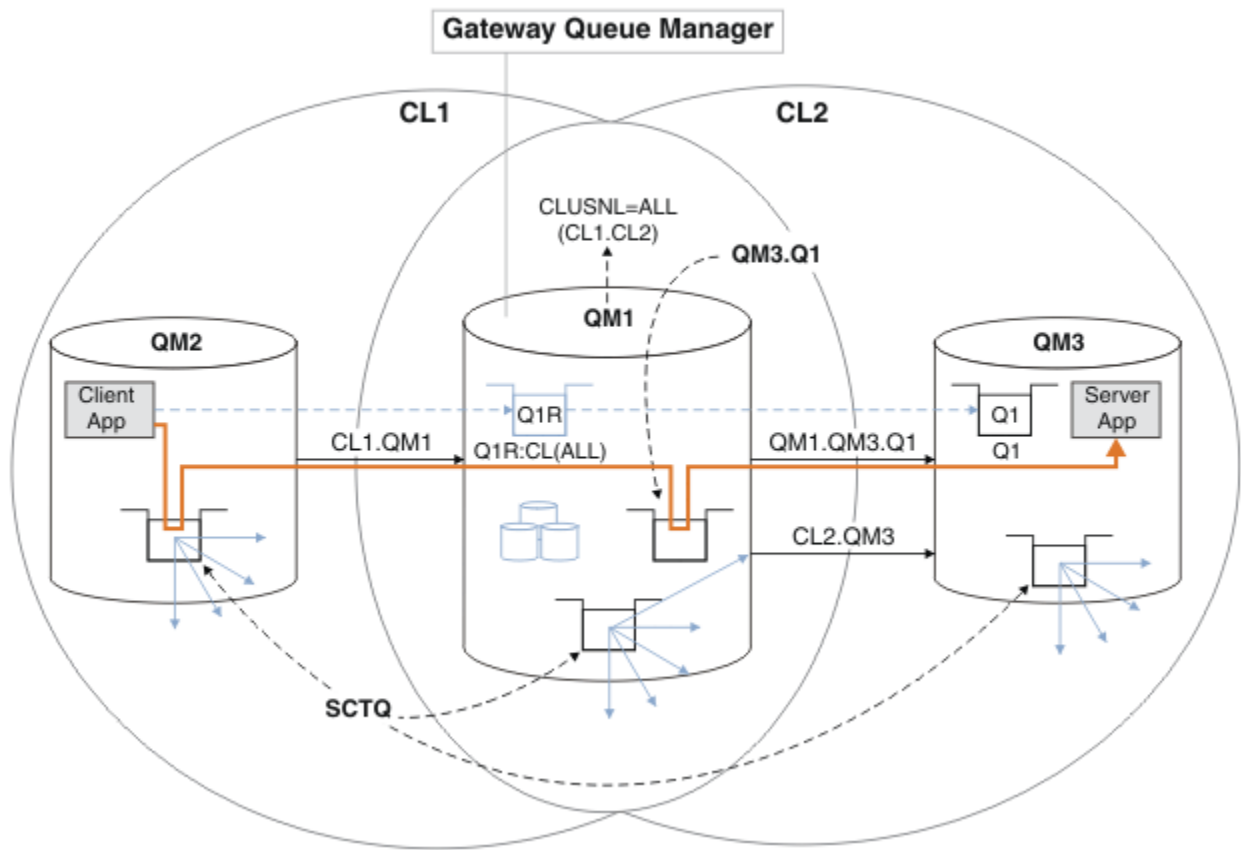


图 32: 使用远程队列定义部署到中心和辐射群体系统结构的客户机/服务器应用程序

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

[第 134 页的『访问控制和多个集群传输队列』](#)

在应用程序将消息放入远程集群队列时选择三种检查方式。这些方式是针对集群队列进行远程检查，针对 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 进行本地检查，或者针对集群队列或集群队列管理器的本地概要文件进行检查。

[第 237 页的『集群: 使用多个集群传输队列进行应用程序隔离』](#)

您可以在集群中的队列管理器之间隔离消息流。您可以将由不同集群发送方通道传输的消息放入不同的集群传输队列中。您可以在单个集群中使用此方法，也可以将此方法与重叠的集群配合使用。本主题提供了一些示例和一些最佳实践，以指导您选择使用方法。

相关任务

[第 164 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 181 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 166 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』](#)

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。

[第 186 页的『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。

[第 168 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 171 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。

[第 240 页的『集群: 规划如何配置集群传输队列』](#)

将指导您选择集群传输队列。您可以配置一个公共缺省队列，单独的缺省队列或手动定义的队列。配置多个集群传输队列适用于除 z/OS 以外的平台。

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 Version 7.5 或更高版本上，并且必须在除 z/OS 以外的平台上。

2. 通过执行该任务中的步骤，构造第 181 页的『使用网关队列管理器创建两个重叠的集群』中的第 181 页的图 37 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 上的集群接收方通道的名称；请参阅第 169 页的图 33。

与第 166 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。

在第 169 页的图 33 中隔离集群消息流量的配置不会导致与在第 166 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 169 页的图 33 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 171 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』。

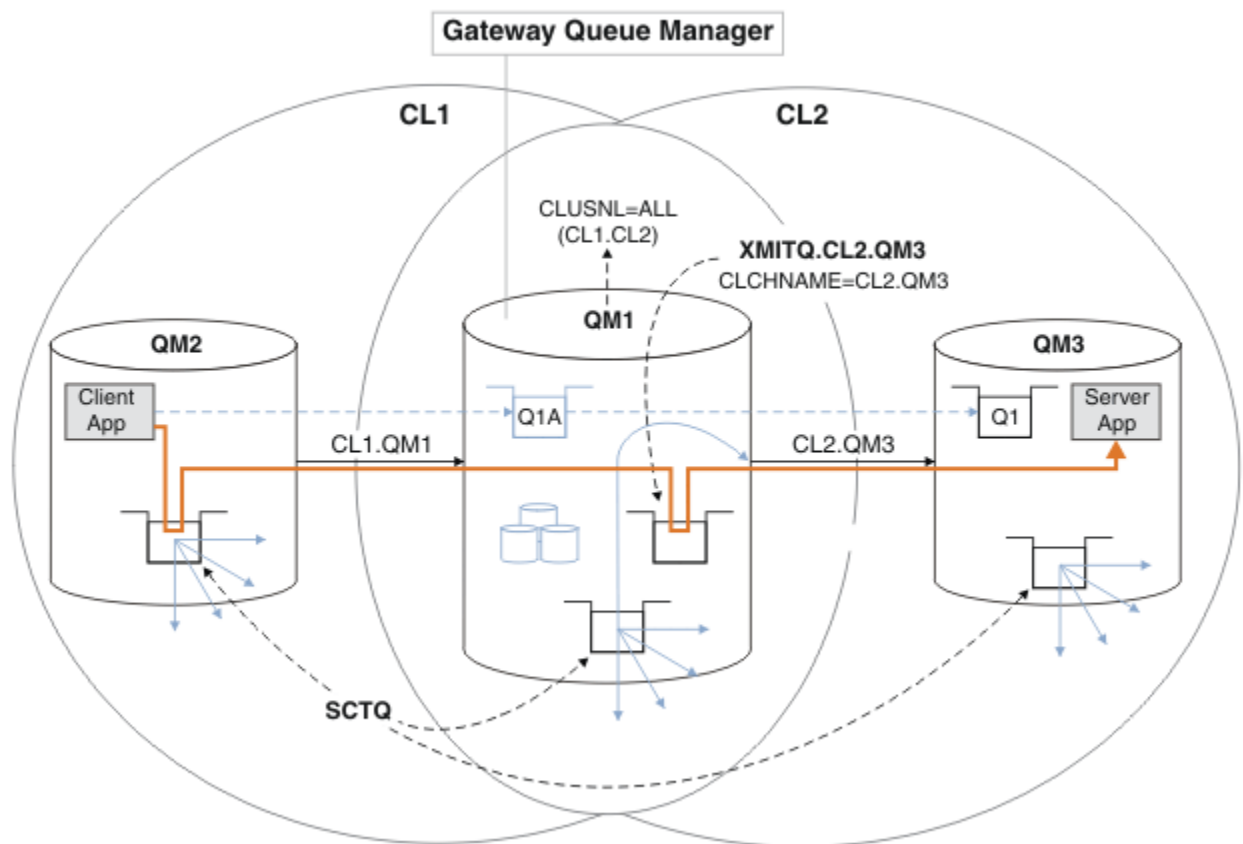


图 33: 使用其他集群传输队列部署到中心和辐射体系结构的客户机/服务器应用程序。

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 WebSphere 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)
CONNNAME(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 WebSphere 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)
CONNNAME(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

[第 134 页的『访问控制和多个集群传输队列』](#)

在应用程序将消息放入远程集群队列时选择三种检查方式。这些方式是针对集群队列进行远程检查，针对 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 进行本地检查，或者针对集群队列或集群队列管理器的本地概要文件进行检查。

[第 237 页的『集群: 使用多个集群传输队列进行应用程序隔离』](#)

您可以在集群中的队列管理器之间隔离消息流。您可以将由不同集群发送方通道传输的消息放入不同的集群传输队列中。您可以在单个集群中使用此方法，也可以将此方法与重叠的集群配合使用。本主题提供了一些示例和一些最佳实践，以指导您选择使用方法。

[第 142 页的『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。

相关任务

[第 164 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 181 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 166 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』](#)

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。

[第 186 页的『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。

[第 168 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 171 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。

[第 240 页的『集群: 规划如何配置集群传输队列』](#)

将指导您选择集群传输队列。您可以配置一个公共缺省队列，单独的缺省队列或手动定义的队列。配置多个集群传输队列适用于除 z/OS 以外的平台。

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 169 页的图 33](#) 中说明的配置。

1. 网关队列管理器必须在 **Version 7.5** 或更高版本上，并且必须在除 z/OS 以外的平台上。
2. 通过执行该任务中的步骤，构造 [第 181 页的『使用网关队列管理器创建两个重叠的集群』](#) 中的 [第 181 页的图 37](#) 中显示的重叠集群。
3. 在 [第 168 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』](#) 中执行 [第 169 页的图 33](#) 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 168 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，你还有两个选择。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 168 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 173 页的图 34](#) 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 WebSphere MQ 无法自动将消息从您在 [第 168 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。

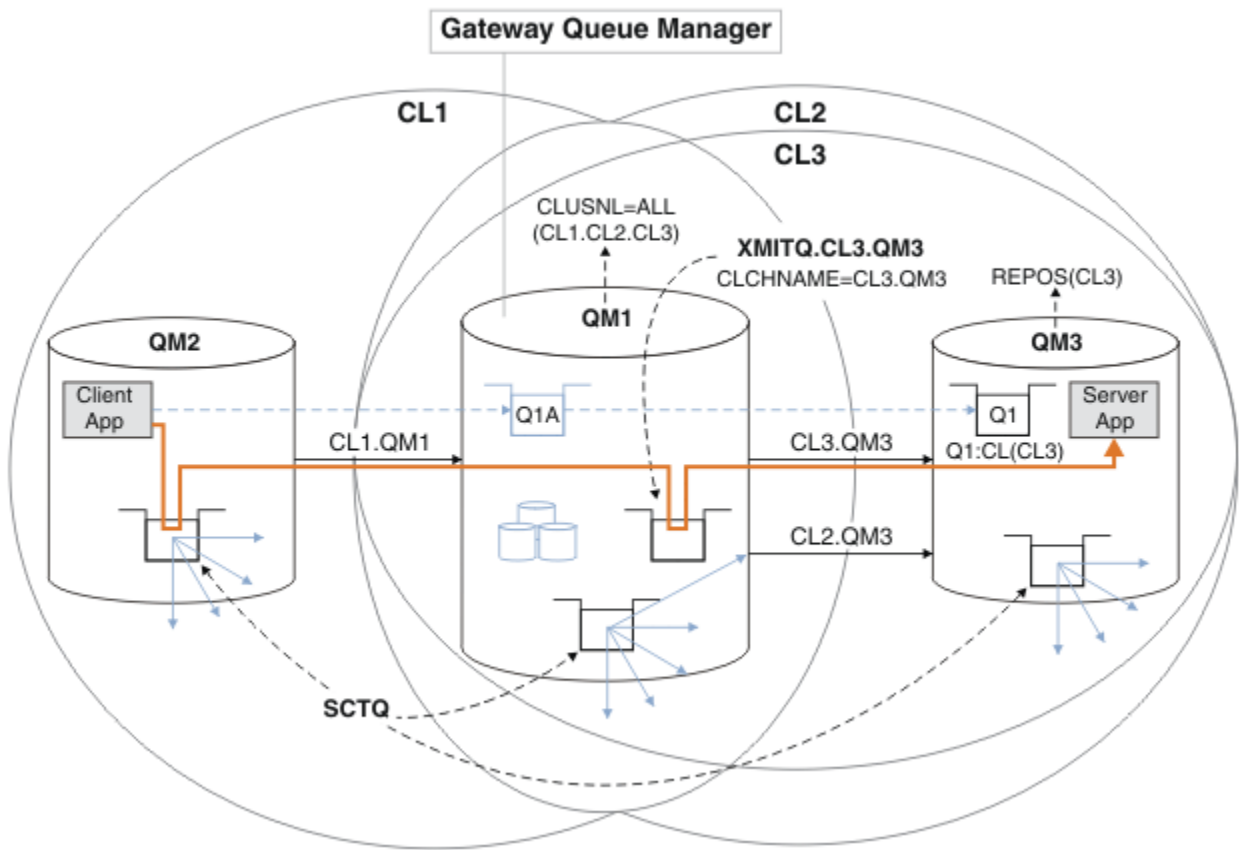


图 34: 使用其他集群来分隔网关队列管理器中的消息流量，该消息流量转至同一队列管理器上的多个集群队列中的一个

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 NAMLIST(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 NAMLIST(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 WebSphere 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 WebSphere 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)
CONNNAME(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 WebSphere 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)
CONNNAME(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.传输。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

第 134 页的『[访问控制和多个集群传输队列](#)』

在应用程序将消息放入远程集群队列时选择三种检查方式。这些方式是针对集群队列进行远程检查，针对 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 进行本地检查，或者针对集群队列或集群队列管理器的本地概要文件进行检查。

第 237 页的『[集群: 使用多个集群传输队列进行应用程序隔离](#)』

您可以在集群中的队列管理器之间隔离消息流。您可以将由不同集群发送方通道传输的消息放入不同的集群传输队列中。您可以在单个集群中使用此方法，也可以将此方法与重叠的集群配合使用。本主题提供了一些示例和一些最佳实践，以指导您选择使用方法。

第 142 页的『[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。

相关任务

第 164 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 181 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 166 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](#)』

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。

第 186 页的『[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。

第 168 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 171 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。

第 240 页的『[集群: 规划如何配置集群传输队列](#)』

将指导您选择集群传输队列。您可以配置一个公共缺省队列，单独的缺省队列或手动定义的队列。配置多个集群传输队列适用于除 z/OS 以外的平台。

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 `CLUSDR` 定义上使用 `+QMNAME+` 的能力。

z/OS 上未提供任何功能部件。

方案:

- 已按第 153 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 WebSphere MQ 从系统中查找连接名称。IBM WebSphere 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：如果此队列管理器正在重新加入集群，请完成一些额外的步骤。

- a) 如果要将队列管理器添加到先前已从同一集群中除去的集群，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：
 - i) 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，当集群正在运行中时，使用 **REFRESH CLUSTER** 命令可能会破坏该集群，并且将在 27 天的时间间隔之后，集群对象才会再次自动向所有相关队列管理器发送状态更新。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](#)。

- ii) 重新启动 CLUSSDR 通道 (例如, 使用 `START CHANNEL` 命令)。
- iii) 重新启动 CLUSRCVR 通道。
- b) 如果集群是发布/预订集群, 并且重新连接的队列管理器具有预订, 请发出以下命令以确保在集群中正确同步代理预订:

```
REFRESH QMGR TYPE(PROXYSUB)
```

结果

此任务设置的集群与 第 162 页的『将队列管理器添加至集群』相同:

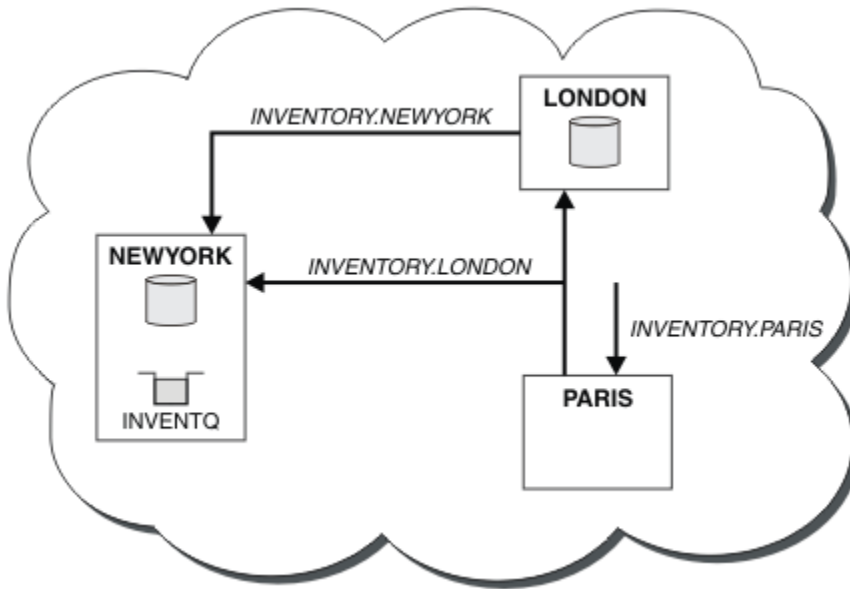


图 35: 具有三个队列管理器的 *INVENTORY* 集群

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 WebSphere MQ 上, 将 +QMNAME+ 解析为队列管理器名称 (LONDON)。然后, IBM WebSphere MQ 将名为 INVENTORY.LONDON 的通道定义与相应的 CLUSRCVR 定义相匹配。

WebSphere 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 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。

方案:

- 已按第 162 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 192 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: 如果此队列管理器正在重新加入集群，请完成一些额外的步骤。

- a) 如果要将队列管理器添加到先前已从同一集群中除去的集群，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:

- i) 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, 当集群正在运行中时, 使用 **REFRESH CLUSTER** 命令可能会破坏该集群, 并且将在 27 天的时间间隔之后, 集群对象才会再次自动向所有相关队列管理器发送状态更新。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。

ii) 重新启动 CLUSSDR 通道 (例如, 使用 `START CHANNEL` 命令)。

iii) 重新启动 CLUSRCVR 通道。

b) 如果集群是发布/预订集群, 并且重新连接的队列管理器具有预订, 请发出以下命令以确保在集群中正确同步代理预订:

```
REFRESH QMGR TYPE(PROXYSUB)
```

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 180 页的图 36 显示了此任务设置的 INVENTORY 集群。

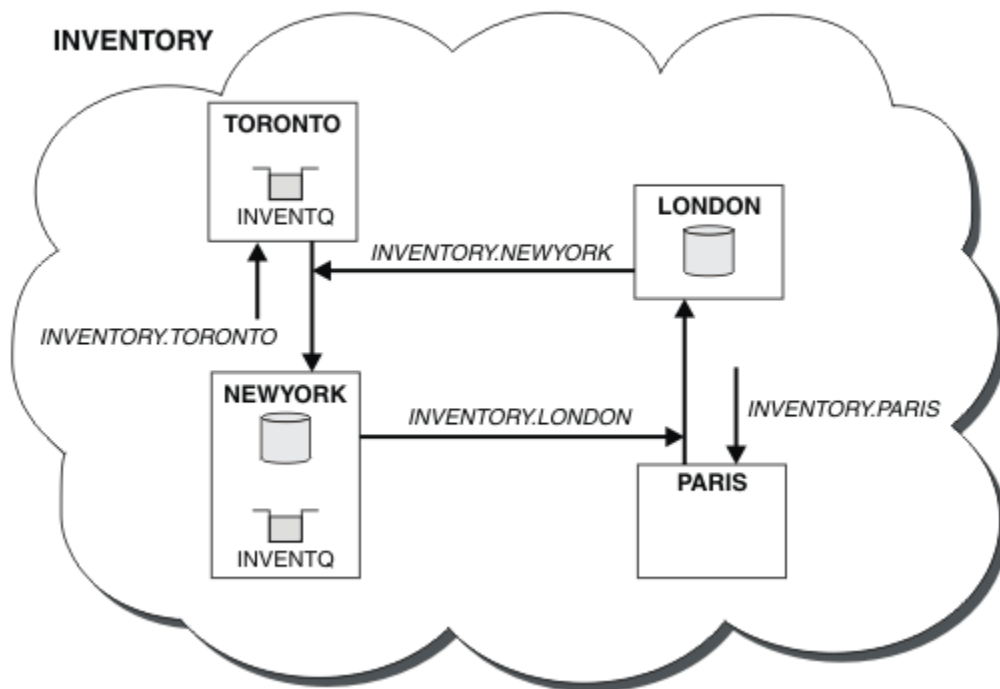


图 36: 具有四个队列管理器的 INVENTORY 集群

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，如果尚未执行此操作，请在 IBM WebSphere 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。

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

第 181 页的图 37 中显示了用于说明隔离集群消息流量的示例集群配置。第 237 页的『[集群: 使用多个集群传输队列进行应用程序隔离](#)』中描述了此示例。

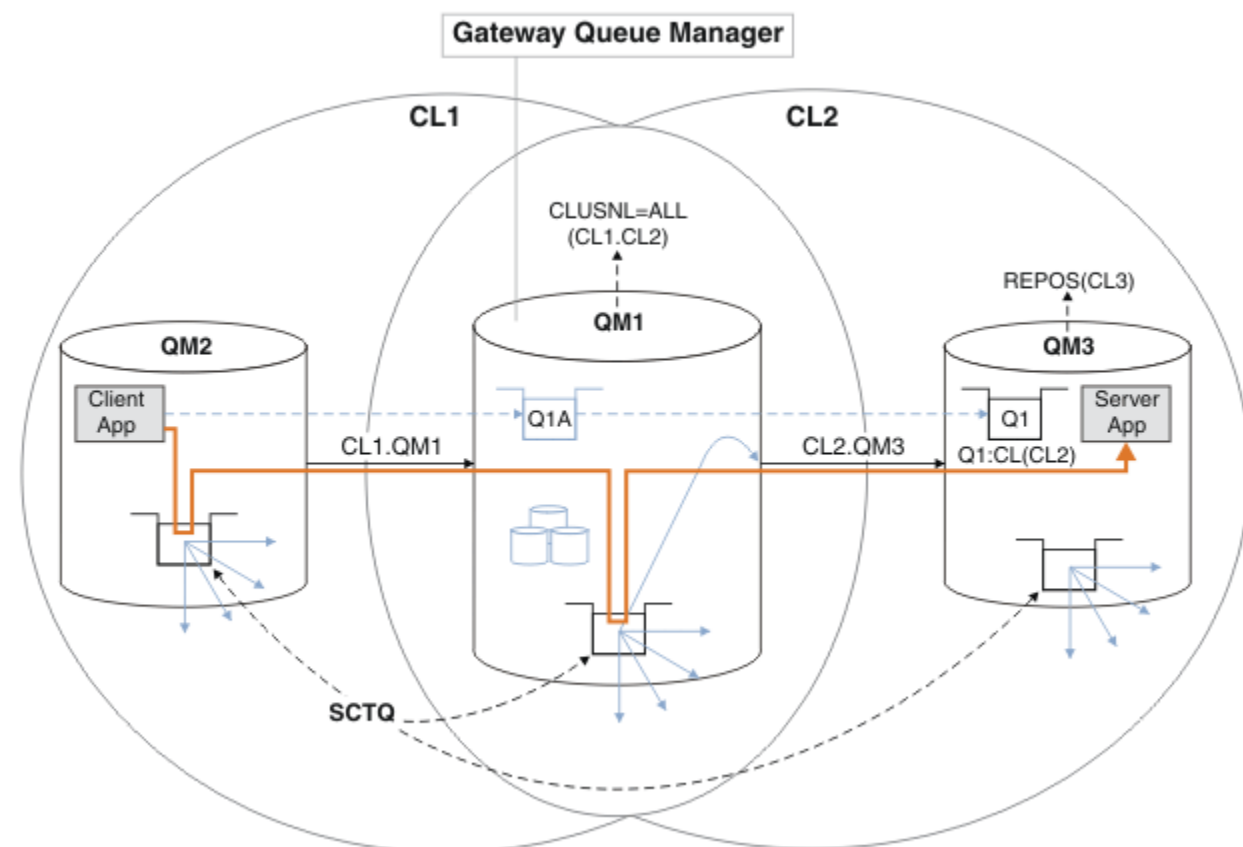


图 37: 使用 IBM WebSphere MQ 集群将客户机/服务器应用程序部署到中心和辐射体系结构

要使构造示例的步骤数尽可能少，配置保持简单，而不是现实。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，请参阅第 240 页的『[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。

这些指示信息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。由于性能原因，网关队列管理器未用作存储库。

过程

1. 创建并启动队列管理器 QM1, QM2, QM3, QM4 和 QM5。

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QMn
strmqm QmgrName
```

注: QM4 和 QM5 是集群的备份完整存储库。

2. 定义并启动每个队列管理器的侦听器。

```
*... On QMn
DEFINE LISTENER(TCP141n) TRPTYPE(TCP) IPADDR(hostname) PORT(141n) CONTROL(QMGR) REPLACE
START LISTENER(TCP141n)
```

3. 为所有集群创建集群名称列表。

```
*... On 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 182 页的表 26](#) 中显示的值:

队列管理器	集群 <i>c</i>	其他存储库 <i>n</i>	此存储库 <i>m</i>
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 添加到每个集群。

```
*... On 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. 将集群队列管理器别名 Q1A 添加到网关队列管理器。

```
*... On QM1
DEFINE QALIAS(Q1A) CLUSNL(ALL) TARGET(Q1) TARGTYPE(Queue) DEFBIND(NOTFIXED) REPLACE
```

注: 在任何其他队列管理器上使用队列管理器别名但 QM1 的应用程序在打开别名队列时必须指定 DEFBIND(NOTFIXED)。 **DEFBIND** 指定当应用程序打开队列时是否固定消息头中的路由信息。如果设置为缺省值 OPEN, 那么会将消息路由到 Q1@QM1。Q1@QM1 不存在, 因此来自其他队列管理器的消息将在死信队列上结束。通过将队列属性设置为 DEFBIND(NOTFIXED), 缺省为 **DEFBIND** 的队列设置的应用程序 (例如 **amqspu**t) 的行为方式正确。

9. 将所有集群队列管理器的集群队列管理器别名定义添加到网关队列管理器 QM1。

```
*... On QM1
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSNL(ALL) REPLACE
```

提示: 网关队列管理器上的队列管理器别名定义传输消息, 这些消息引用另一个集群中的队列管理器; 请参阅 [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 **amqspu**t 以放置消息。

```
C:\IBM\MQ>amqspu 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 **amqsge**t 以从 QM3 上的 Q1 获取消息

```
C:\IBM\MQ>amqsge 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 在 QM1 上定义为集群队列管理器别名。QM3 将应答消息路由到 QM1。QM1 将消息路由到 QM2。

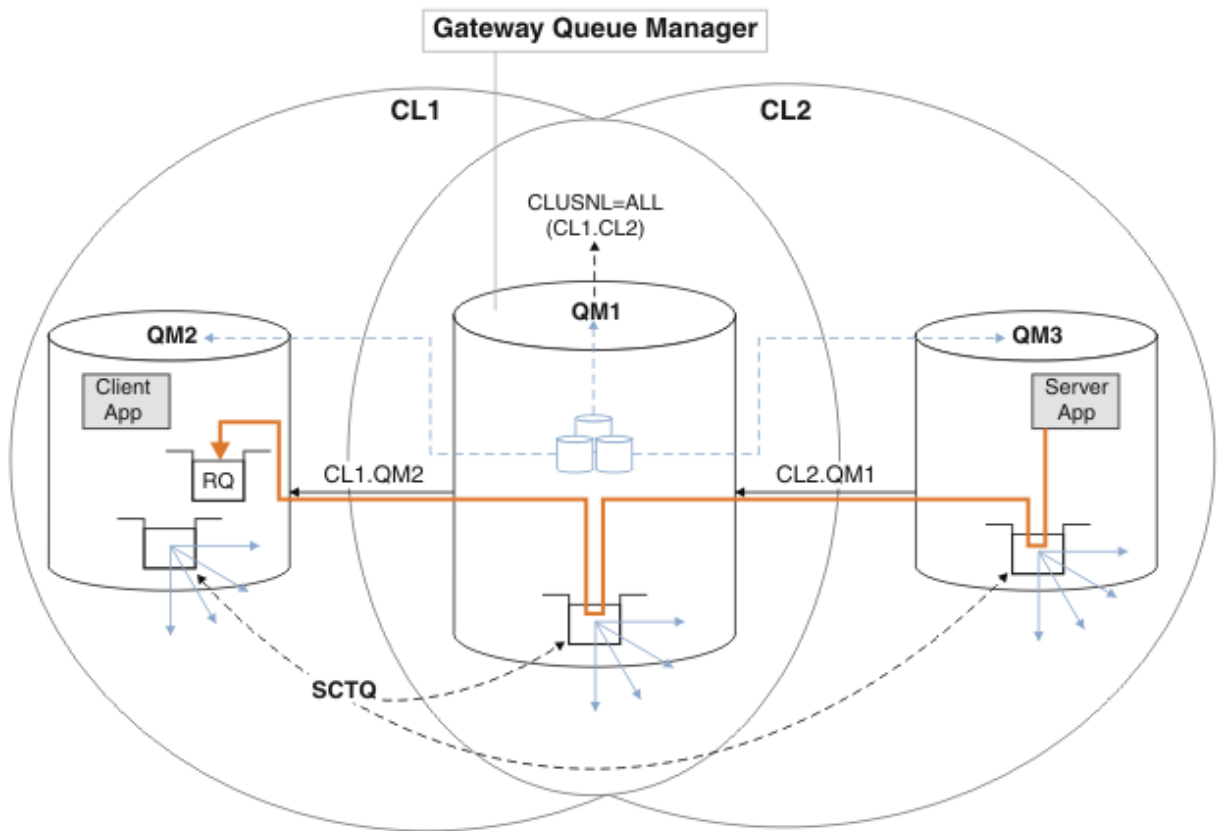


图 38: 使用队列管理器别名将应答消息返回到其他集群

路由工作方式如下。每个集群中的每个队列管理器都在 QM1 上具有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。消息将路由到 QM1，因为在集群 CL2 中的 QM1 上定义了队列管理器别名定义 QM2，而队列管理器 QM2 不在集群 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。

QM1 将消息放在 QM1 上的集群传输队列上，以便传输到 QM2。QM1 将消息路由到 QM2，因为 QM1 for QM2 上的队列管理器别名定义将 QM2 定义为实际目标队列管理器。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。实际定义由 QM1 解析，因为 QM1 和 QM2 都位于同一集群 CL1 中。QM1 从 CL1 的存储库中查找 QM2 的连接信息，并将消息路由到 QM2。对于要由 QM1 重新路由的消息，服务器应用程序必须已在选项 DEFBIND 设置为 MQBND_BIND_NOT_FIXED 的情况下打开应答队列。如果服务器应用程序已使用选项 MQBND_BIND_ON_OPEN 打开应答队列，那么不会重新路由该消息并将其结束于死信队列。

- 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 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, 2024. ALL RIGHTS RESERVED.  
01/02/2012 16:17:15: WebSphere 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

[第 134 页的『访问控制和多个集群传输队列』](#)

在应用程序将消息放入远程集群队列时选择三种检查方式。这些方式是针对集群队列进行远程检查，针对 SYSTEM.CLUSTER.TRANSMIT.QUEUE 进行本地检查，或者针对集群队列或集群队列管理器的本地概要文件进行检查。

[第 237 页的『集群: 使用多个集群传输队列进行应用程序隔离』](#)

您可以在集群中的队列管理器之间隔离消息流。您可以将由不同集群发送方通道传输的消息放入不同的集群传输队列中。您可以在单个集群中使用此方法，也可以将此方法与重叠的集群配合使用。本主题提供了一些示例和一些最佳实践，以指导您选择使用方法。

相关任务

[第 164 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 181 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 166 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』](#)

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。

[第 186 页的『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。

[第 168 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 171 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。

[第 240 页的『集群: 规划如何配置集群传输队列』](#)

将指导您选择集群传输队列。您可以配置一个公共缺省队列，单独的缺省队列或手动定义的队列。配置多个集群传输队列适用于除 z/OS 以外的平台。

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 Version 7.5 或更高版本上，并且必须在除 z/OS 以外的平台上。
2. 通过执行该任务中的步骤，构造 [第 181 页的『使用网关队列管理器创建两个重叠的集群』](#) 中的 [第 181 页的图 37](#) 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 Version 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 187 页的图 39](#)。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 164 页的『将队列管理器添加到集群: 单独的传输队列』](#)。

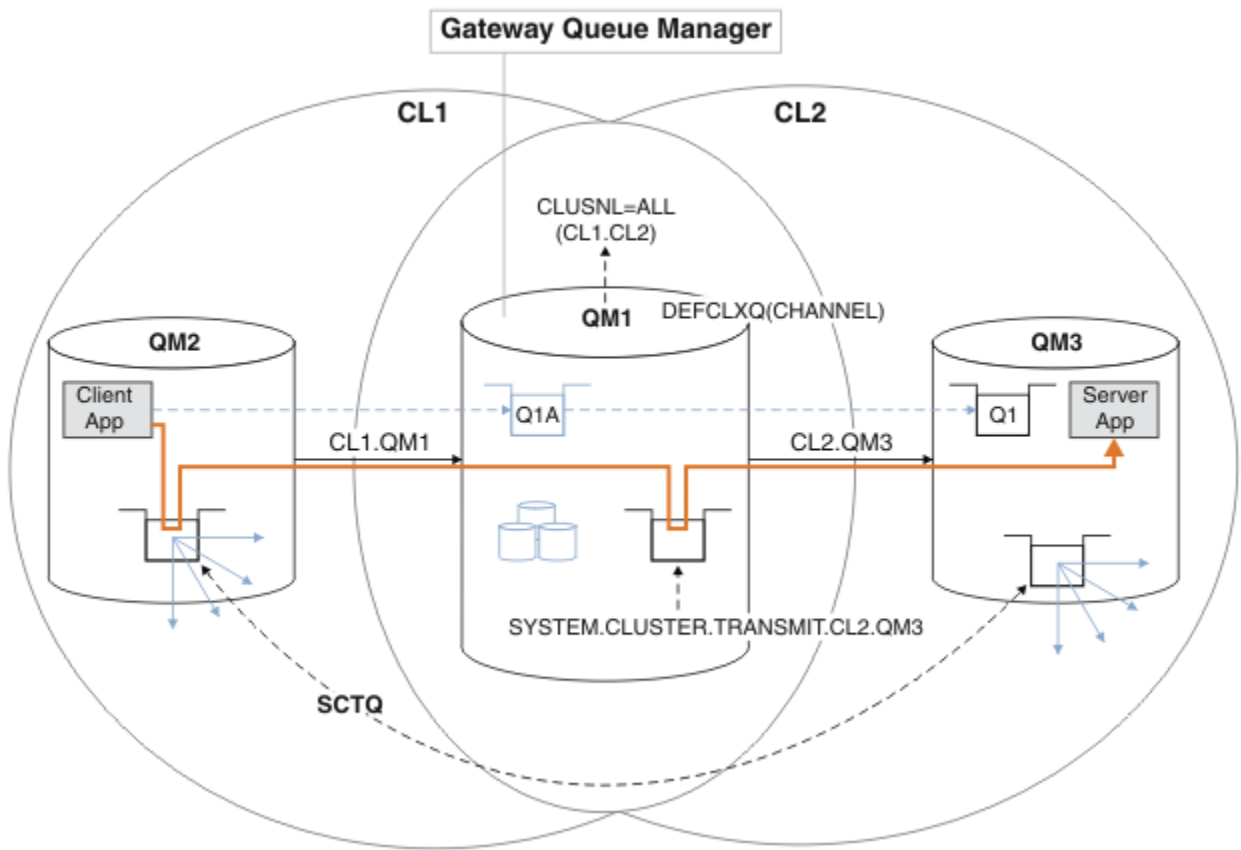


图 39: 客户机/服务器应用程序已部署到具有网关队列管理器上的独立集群传输队列的中心和辐射体系结构。

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)
CONNNAME (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)
CONNNAME (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 WebSphere 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)
CONNNAME (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)
CONNNAME (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)
CONNNAME (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)
CONNNAME (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 WebSphere 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.传输。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 **amqspout** 以放置消息。

```
C:\IBM\MQ>amqspout 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

[第 134 页的『访问控制和多个集群传输队列』](#)

在应用程序将消息放入远程集群队列时选择三种检查方式。这些方式是针对集群队列进行远程检查，针对 SYSTEM.CLUSTER.TRANSMIT.QUEUE 进行本地检查，或者针对集群队列或集群队列管理器的本地概要文件进行检查。

[第 237 页的『集群: 使用多个集群传输队列进行应用程序隔离』](#)

您可以在集群中的队列管理器之间隔离消息流。您可以将由不同集群发送方通道传输的消息放入不同的集群传输队列中。您可以在单个集群中使用此方法，也可以将此方法与重叠的集群配合使用。本主题提供了一些示例和一些最佳实践，以指导您选择使用方法。

相关任务

[第 164 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 181 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 166 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』](#)

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。

[第 186 页的『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。

[第 168 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 171 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。

[第 240 页的『集群: 规划如何配置集群传输队列』](#)

将指导您选择集群传输队列。您可以配置一个公共缺省队列，单独的缺省队列或手动定义的队列。配置多个集群传输队列适用于除 z/OS 以外的平台。

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。

方案:

- 已按第 179 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, OPROCS 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

此任务所设置的集群类似于先前任务所设置的集群。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)
```

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。

方案：

- 已按第 162 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)
CONNNAME(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)
CONNNAME(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)
```

- 将集群中指向 LONDON 的所有其他集群发送方通道替换为指向 NEWYORK 或 PARIS 的通道。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 194 页的图 40 显示此任务设置的集群。

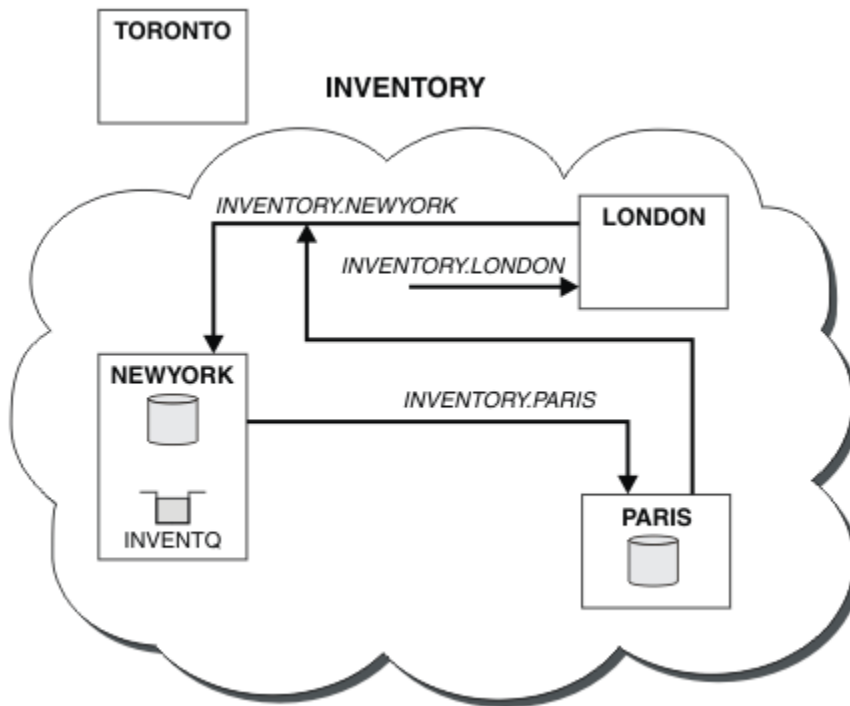


图 40: 将完整存储库移至 PARIS 的 INVENTORY 集群

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 153 页的『设置新集群』到第 192 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。

方案：

- IBM WebSphere MQ 网络已就绪，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 195 页的图 41 中进行了说明。

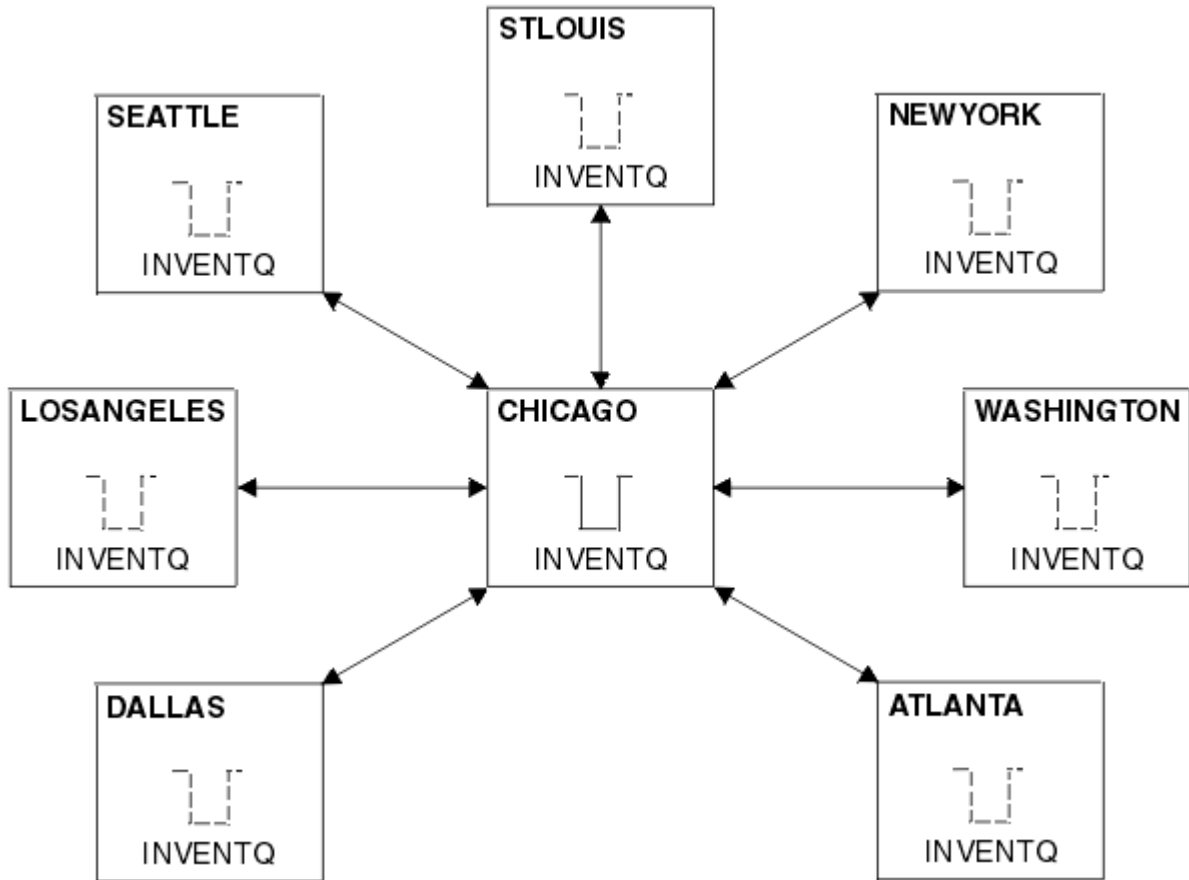


图 41: 集线器和辐射网络

- 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` 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 232 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)  
CONNNAME(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)  
CONNNAME(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。

方案:

- 已设置 WebSphere MQ 集群，如第 194 页的『将现有网络转换为集群』中所述。
- 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 200 页的图 42](#) 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。

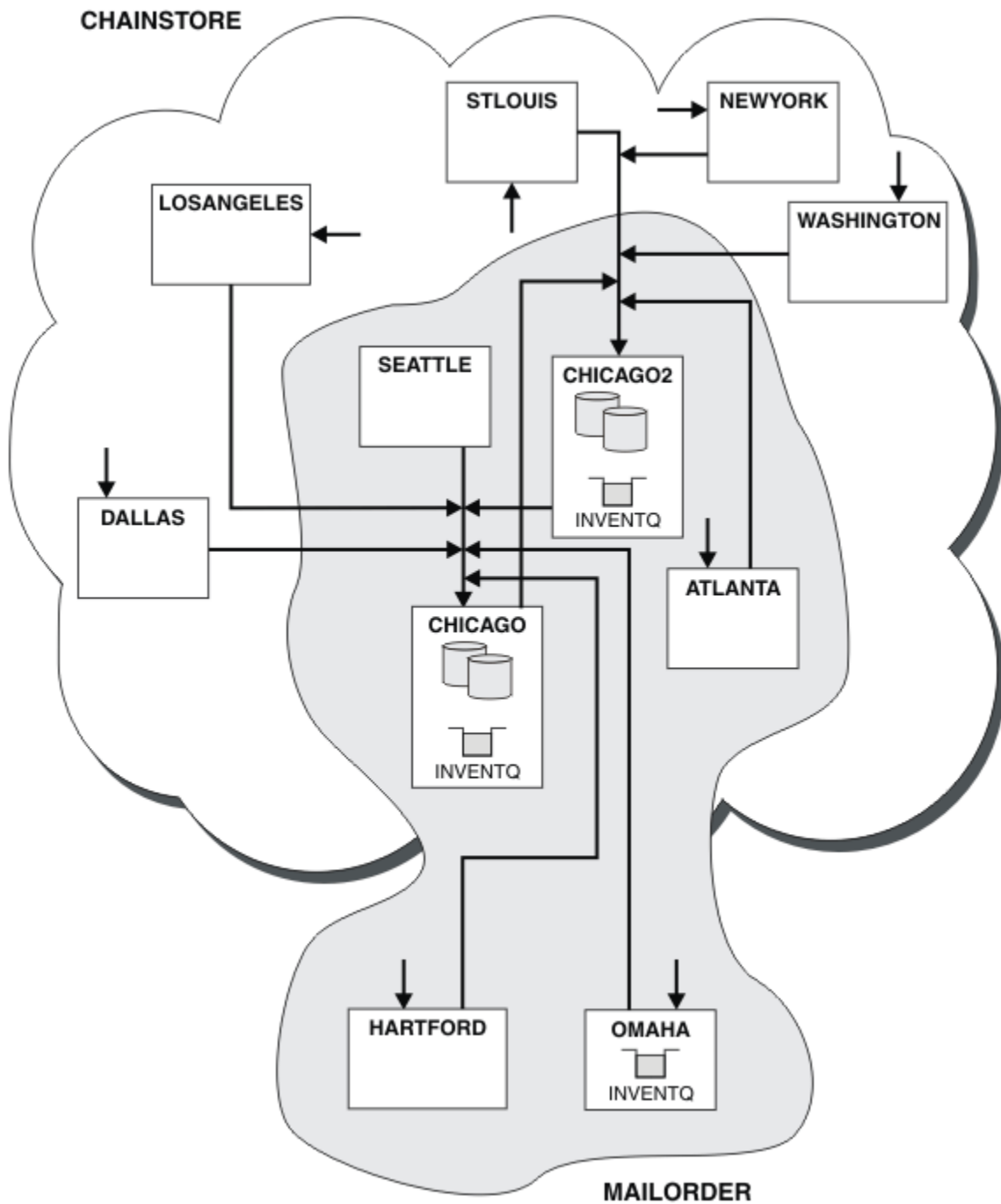


图 42: 互连集群

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。

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。

方案：

- 已按第 194 页的『将现有网络转换为集群』中所述设置 IBM WebSphere 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 205 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。

从集群除去队列管理器

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 204 页的『[从集群中除去队列管理器：备用方法](#)』。

在从集群中除去队列管理器之前，必须确保该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 192 页的『[将完整存储库移至另一个队列管理器](#)』完成步骤 1-4。
- 如果队列管理器托管集群队列，请从第 190 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。

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。

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 162 页的『[将队列管理器添加至集群](#)』中所述进行设置，并按第 190 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(' ')
```

2. 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。

3. 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.LONDON) 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)
```

稍后可以将除去的队列管理器重新添加到集群中，如第 162 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 205 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。

从集群中除去队列管理器: 备用方法

从集群中除去队列管理器，在某些情况下，由于重大系统或配置问题，队列管理器无法与集群中的任何完整存储库进行通信。

开始之前

此手动从集群中除去队列管理器的备用方法将停止并删除所有将已除去的队列管理器链接到集群的集群通道，并强制从集群中除去该队列管理器。此方法用于要除去的队列管理器无法与任何完整存储库进行通信的场景。这可能是由于(例如)队列管理器已停止工作，或者由于队列管理器与集群之间发生了长时间的通信故障。否则，请使用最常见的方法: 第 202 页的『从集群除去队列管理器』。

在从集群中除去队列管理器之前，必须确保该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 192 页的『将完整存储库移至另一个队列管理器』完成步骤 1-4。
- 如果队列管理器托管集群队列，请从第 190 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。

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。

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 162 页的『将队列管理器添加至集群』中所述进行设置，并按第 190 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。

当无法等待队列管理器正常离开集群时，此过程可能适用于紧急情况。

过程

1. 停止用于与集群中的其他队列管理器通信的所有通道。使用 `MODE(FORCE)` 在队列管理器 LONDON 上停止 `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)
```

2. 在队列管理器 LONDON 上监视通道状态，直到通道停止为止:

```
DISPLAY CHSTATUS(INVENTORY.LONDON)
DISPLAY CHSTATUS(INVENTORY.TORONTO)
DISPLAY CHSTATUS(INVENTORY.PARIS)
DISPLAY CHSTATUS(INVENTORY.NEWYORK)
```

在通道停止后，不会向集群中的其他队列管理器发送更多应用程序消息。

3. 在队列管理器 LONDON 上删除手动定义的集群通道:

```
DELETE CHANNEL(INVENTORY.NEWYORK)
DELETE CHANNEL(INVENTORY.TORONTO)
```

4. 集群中的其余队列管理器仍保留已除去队列管理器的知识，并且可能继续向其发送消息。要从其余队列管理器中清除知识，请在其中一个完整存储库上重置从集群中除去的队列管理器：

```
RESET CLUSTER(INVENTORY) ACTION(FORCEREMOVE) QMNAME(LONDON) QUEUES(YES)
```

如果集群中可能有另一个队列管理器与除去的队列管理器同名，请指定除去的队列管理器的 **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)
```

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，请删除 LONDON 上的 CLUSRCVR 通道：

```
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 162 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 205 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。

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。

要点：如果从集群中除去队列管理器并使用 REPOS (YES) 对其进行刷新，那么仅通过更改其 CLUSRCVR 的 CLUSTER 属性将无法再次将其重新添加。将其 CLUSRCVR 的 CLUSTER 属性更改为非空白 (即，clustername) 之后，您还需要使用 REPOS (NO) 发出刷新集群，此时 CLUSRCVR 上的内部序号将更新。然后，队列管理器将成功地将自身重新引入到完整存储库和其余集群成员中。(请注意，必须在为 CLUSRCVR 通道指定正确的集群名称之后运行命令的 REPOS (NO) 版本。)

此限制仅适用于 IBM WebSphere MQ Version 7.5。

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 202 页的『[从集群除去队列管理器](#)』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

第 134 页的『[集群与分布式排队的比较](#)』
比较需要定义的组件，以使用分布式排队和集群来连接队列管理器。

第 136 页的『[集群的组件](#)』
集群由队列管理器，集群存储库，集群通道和集群队列组成。

第 153 页的『[管理 IBM WebSphere MQ 集群](#)』
您可以创建，扩展和维护 IBM WebSphere MQ 集群。

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。

WebSphere 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进站通道。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。

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 未变满。

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。请参阅第 256 页的『[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 WebSphere MQ 确保不会从队列中丢失任何数据。

相关概念

第 256 页的『[集群：使用 REFRESH CLUSTER 最佳实践](#)』

使用 **REFRESH CLUSTER** 命令可废弃有关集群的所有本地保存的信息，并从集群中的完整存储库重建该信息。您不应该使用此命令，除非在特殊情况下。如果确实需要使用它，那么有关如何使用它有特殊注意事项。此信息是基于测试和客户反馈的指南。

恢复队列管理器

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，当集群正在运行中时，使用 `REFRESH CLUSTER` 命令可能会破坏该集群，并且将在 27 天的时间间隔之后，集群对象才会再次自动向所有相关队列管理器发送状态更新。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。

相关概念

第 256 页的『[集群：使用 REFRESH CLUSTER 最佳实践](#)』

使用 `REFRESH CLUSTER` 命令可废弃有关集群的所有本地保存的信息，并从集群中的完整存储库重建该信息。您不应该使用此命令，除非在特殊情况下。如果确实需要使用它，那么有关如何使用它有特殊注意事项。此信息是基于测试和客户反馈的指南。

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。

在 z/OS 以外的平台上

使用 `qm.ini` 文件或 Windows NT 注册表中的 `AdoptNewMCA`，`AdoptNewMCATimeout` 和 `AdoptNewMCACheck` 属性来控制通道的集群接收方端处于活动状态的问题。

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

集群工作方式

[第 134 页的『集群与分布式排队的比较』](#)

比较需要定义的组件，以使用分布式排队和集群来连接队列管理器。

[第 136 页的『集群的组件』](#)

集群由队列管理器，集群存储库，集群通道和集群队列组成。

[第 153 页的『管理 IBM WebSphere MQ 集群』](#)

您可以创建，扩展和维护 IBM WebSphere MQ 集群。

[第 218 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，并将消息发送到集群的工作负载均衡。

[第 220 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 220 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 132 页的『配置队列管理器集群』](#)

使用本主题中的链接可了解集群如何工作，如何设计集群配置以及如何设置简单集群的示例。

[第 153 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 210 页的图 43](#) 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 可以是不支持集群的 WebSphere MQ 产品上的队列管理器。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)
```

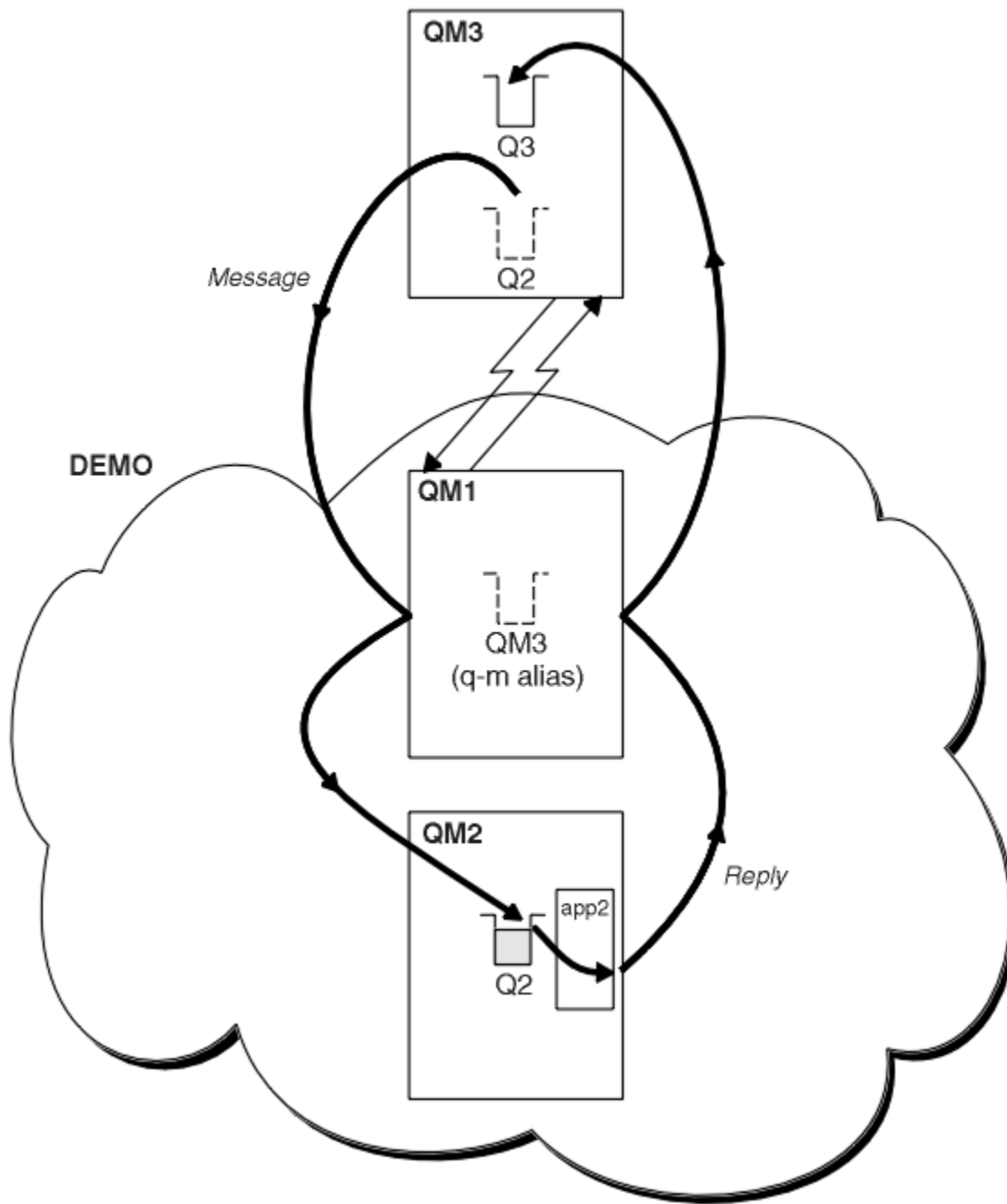


图 43: 从集群外部的队列管理器放入

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 210 页的图 43 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 为集群外部的队列管理器 QM3 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。WebSphere 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。

相关任务

第 212 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 212 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 209 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 213 页的图 44](#)，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(' ')
```

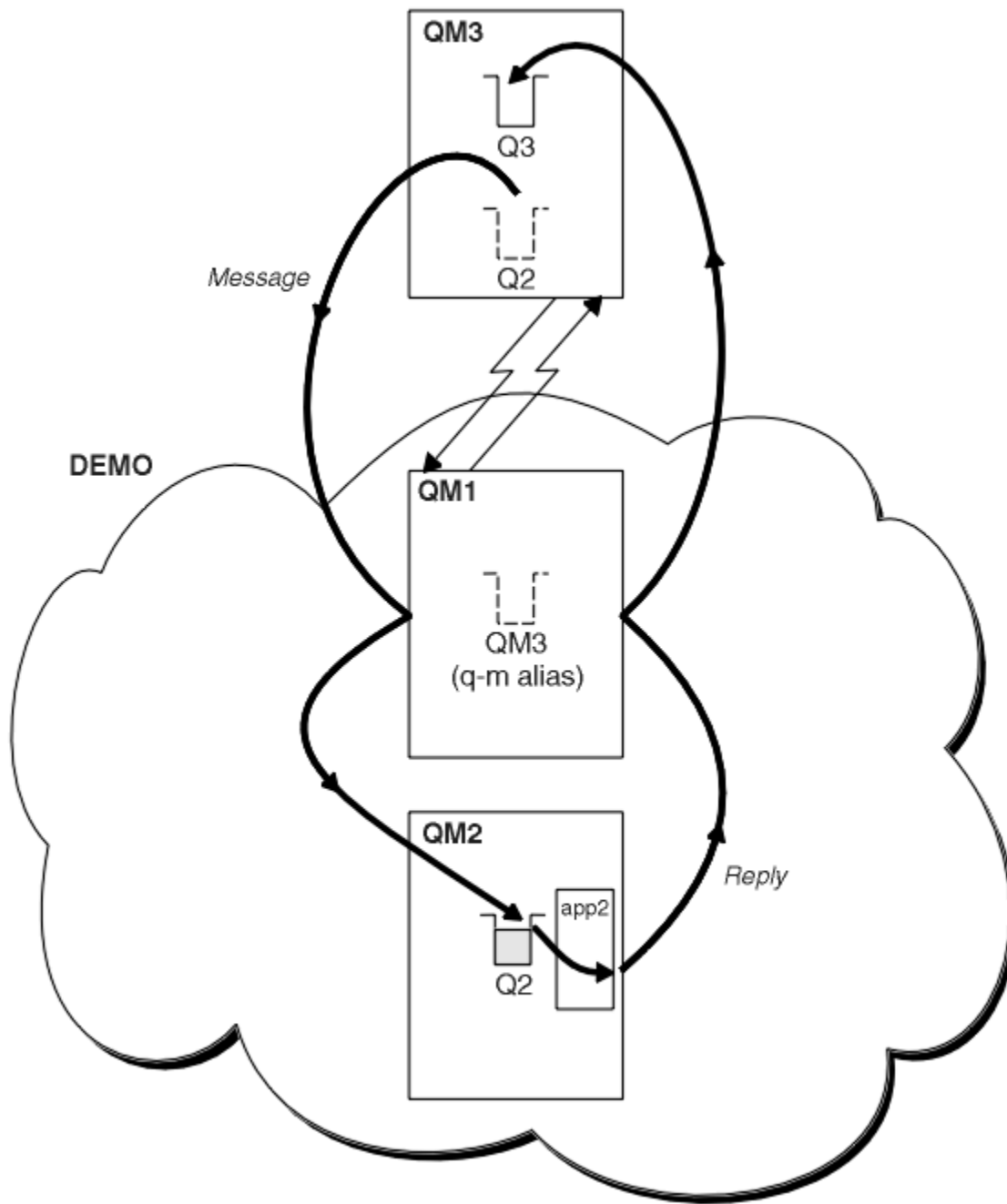


图 44: 从集群外部的队列管理器放入

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)
```

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 214 页的图 45 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。

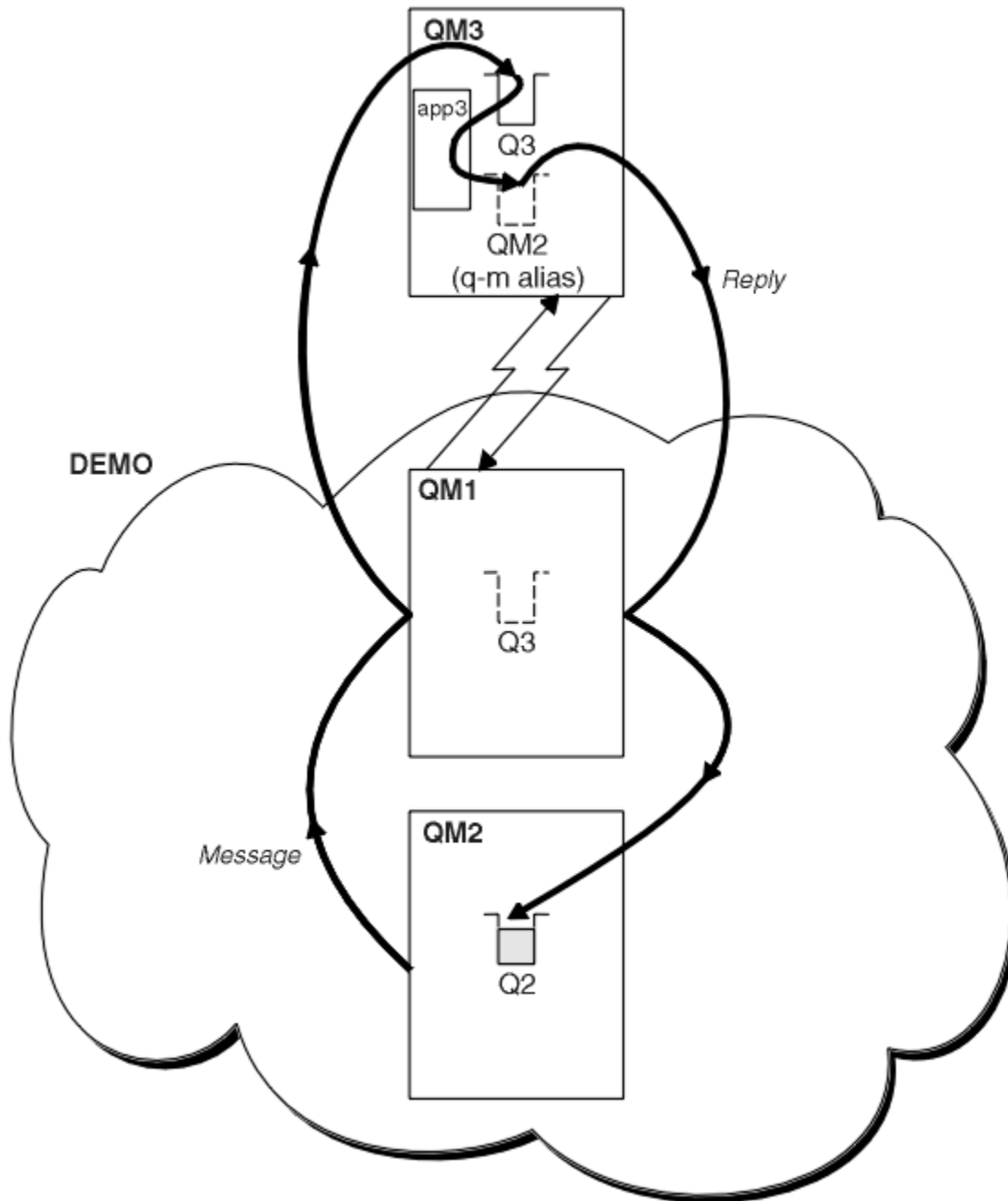


图 45: 放入集群外部的队列管理器

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。第 214 页的图 45 中未显示这些通道。

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。

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 209 页的『配置集群的请求/应答』中的第 210 页的图 43 中所示。

关于此任务

在此场景中，集群外部的队列管理器 QM3 在第 216 页的图 46 中向队列 Q2 发送请求。Q2 托管在集群 DEMO 中的两个队列管理器上以使用工作负载均衡。在队列管理器 QM2 和 QM4 上定义了名为 Q2 的队列，但未在网关队列管理器 QM1 上定义该队列。来自 QM3(集群外部的队列管理器)的请求将发送到 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 216 页的图 46 中未显式显示通道和传输队列。

该过程扩展了第 209 页的『配置集群的请求/应答』中的第 210 页的图 43 中的示例。

过程

1. 在每个 QM2 和 QM4 上定义名为 Q2 的本地队列。

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO) DEFBIND(NOTFIXED)
```

2. 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。

3. 在 QM3 上创建队列管理器别名 Q3。

```
DEFINE QREMOTE(Q3) RNAME(' ') RQMNAME(' ') CLUSTER(DEMO) DEFBIND(NOTFIXED)
```

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，' '。

4. QM1 是网关队列管理器，没有特殊定义。

结果

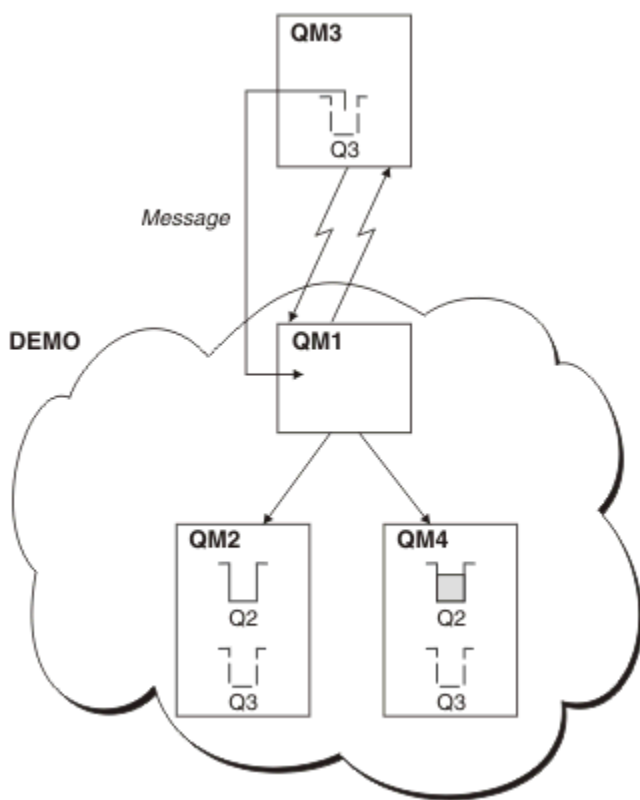


图 46: 从集群外部的队列管理器放入

当 QM3 上的应用程序发出 MQPUT 调用以将消息放入 Q2 时，QREMOTE 定义会导致消息通过网关队列管理器 QM1 进行路由。QM1 使用工作负载均衡在具有 Q3 的集群队列管理器别名的两个队列管理器 QM2 和 QM4 上的名为 Q2 的队列之间分发以 Q2 为目标的消息。

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。

关于此任务

可以有許多較小的集群，而不是將所有隊列管理器分組到一個大型集群中。每個集群都有一個或多個隊列管理器充當網橋。這樣做的優點是可以限制集群中隊列和隊列管理器名稱的可視性。請參閱第 149 頁的『[重疊集群](#)』。使用別名來更改隊列和隊列管理器的名稱，以避免名稱衝突或遵守本地命名約定。

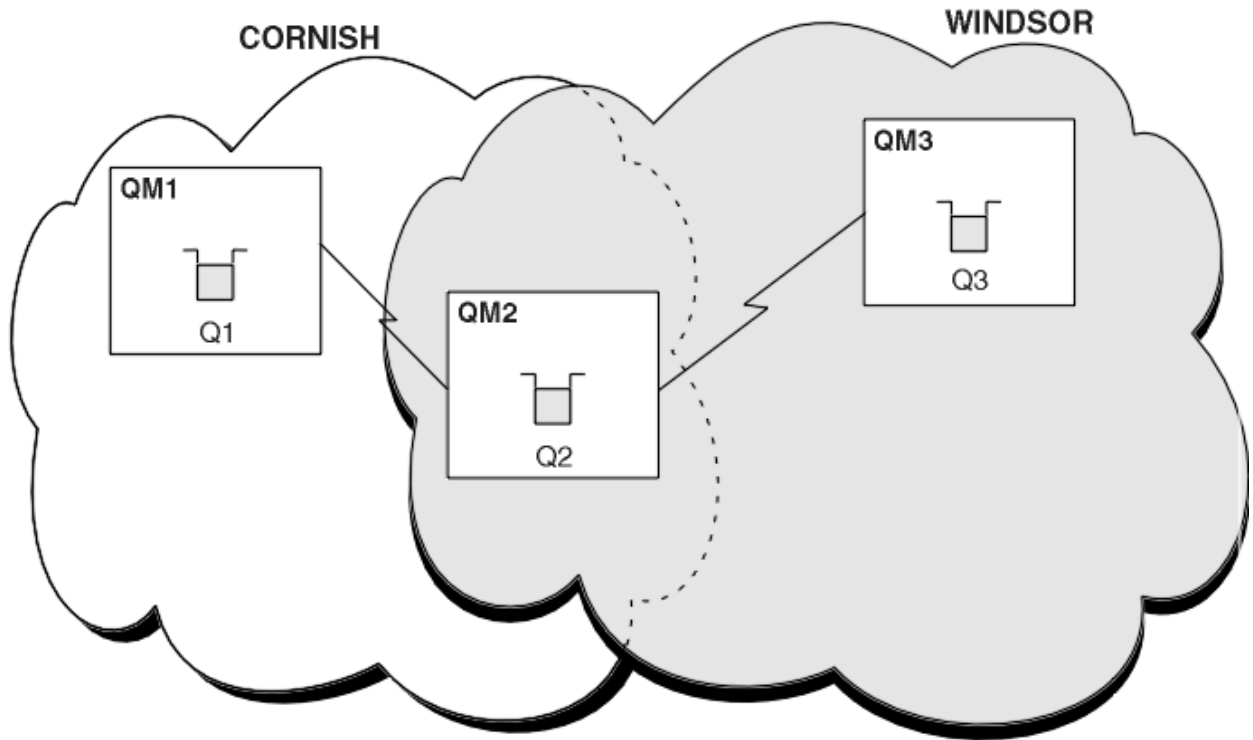


图 47: 跨集群的桥接

第 217 页的图 47 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 NAMLIST(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 217 页的图 47 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，都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。

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，并将消息发送到集群的工作负载均衡。

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 的队列中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。如果找到一个队列管理器，那么它会将来自队列管理器别名定义的 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 181 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 在 QM1 上定义为集群队列管理器别名。QM3 将应答消息路由到 QM1。QM1 将消息路由到 QM2。

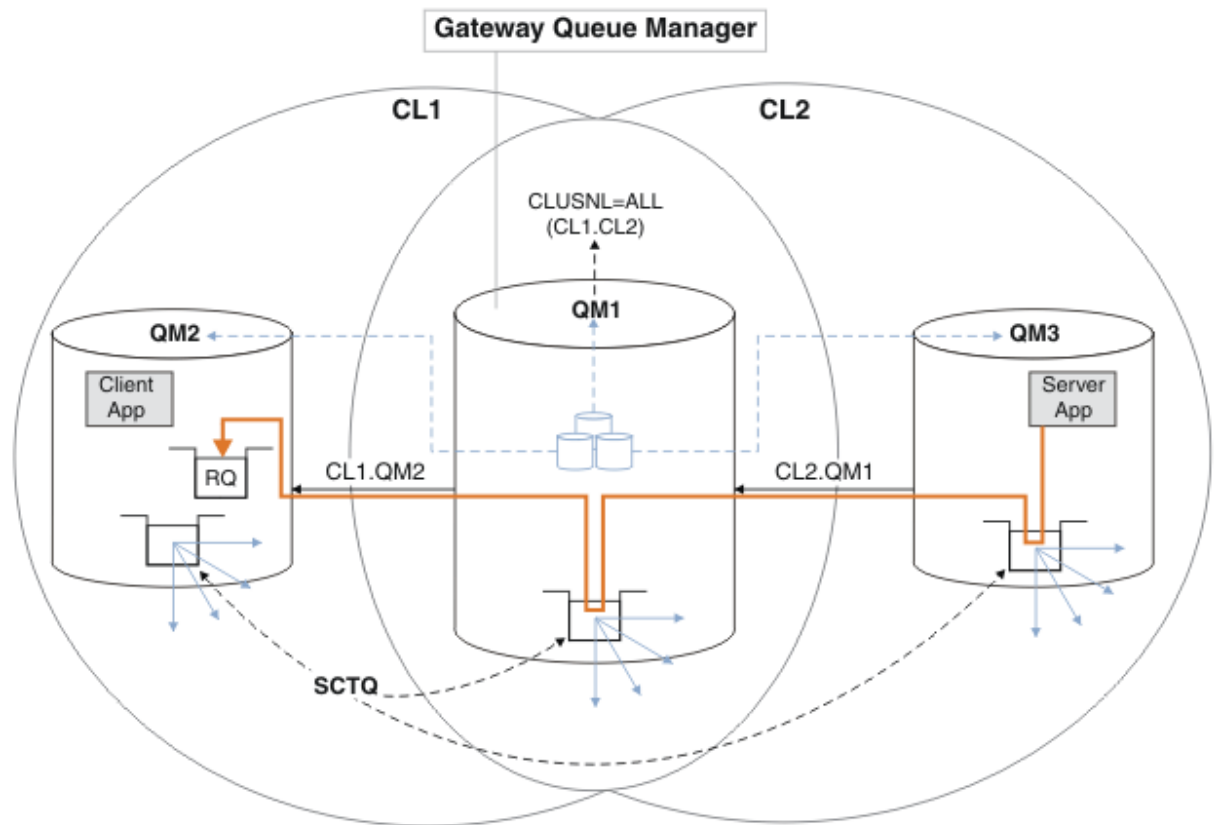


图 48: 使用队列管理器别名将应答消息返回到其他集群

路由工作方式如下。每个集群中的每个队列管理器都在 QM1 上具有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。消息将路由到 QM1，因为在集群 CL2 中的 QM1 上定义了队列管理器别名定义 QM2，而队列管理器 QM2 不在集群 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。

QM1 将消息放在 QM1 上的集群传输队列上，以便传输到 QM2。QM1 将消息路由到 QM2，因为 QM1 for QM2 上的队列管理器别名定义将 QM2 定义为实际目标队列管理器。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。实际定义由 QM1 解析，因为 QM1 和 QM2 都位于同一集群 CL1 中。QM1 从 CL1 的存储库中查找 QM2 的连接信息，并将消息路由到 QM2。对于要由 QM1 重新路由的消息，服务器应用程序必须已在选项 DEFBIND 设置为 MQBND_BIND_NOT_FIXED 的情况下打开应答队列。如果服务器应用程序已使用选项 MQBND_BIND_ON_OPEN 打开应答队列，那么不会重新路由该消息并将其结束于死信队列。

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 215 页的『从集群外部配置工作负载均衡』

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。

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。

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。

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。您可以通过具有应用程序的多个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相互独立运行。

以这种方式使用集群的优点包括：

- 提高队列和应用程序的可用性
- 更快的消息吞吐量
- 更平均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。应用程序在发送消息时不命名队列管理器。工作负载管理算法确定哪个队列管理器处理消息。

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](#)

[集群工作方式](#)

[第 134 页的『集群与分布式排队的比较』](#)

比较需要定义的组件，以使用分布式排队和集群来连接队列管理器。

[第 136 页的『集群的组件』](#)

集群由队列管理器，集群存储库，集群通道和集群队列组成。

[第 153 页的『管理 IBM WebSphere MQ 集群』](#)

您可以创建，扩展和维护 IBM WebSphere MQ 集群。

[第 208 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

[第 132 页的『配置队列管理器集群』](#)

使用本主题中的链接可了解集群如何工作，如何设计集群配置以及如何设置简单集群的示例。

[第 153 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。

[编写和编译集群工作负载出口](#)

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 222 页的图 49](#) 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。

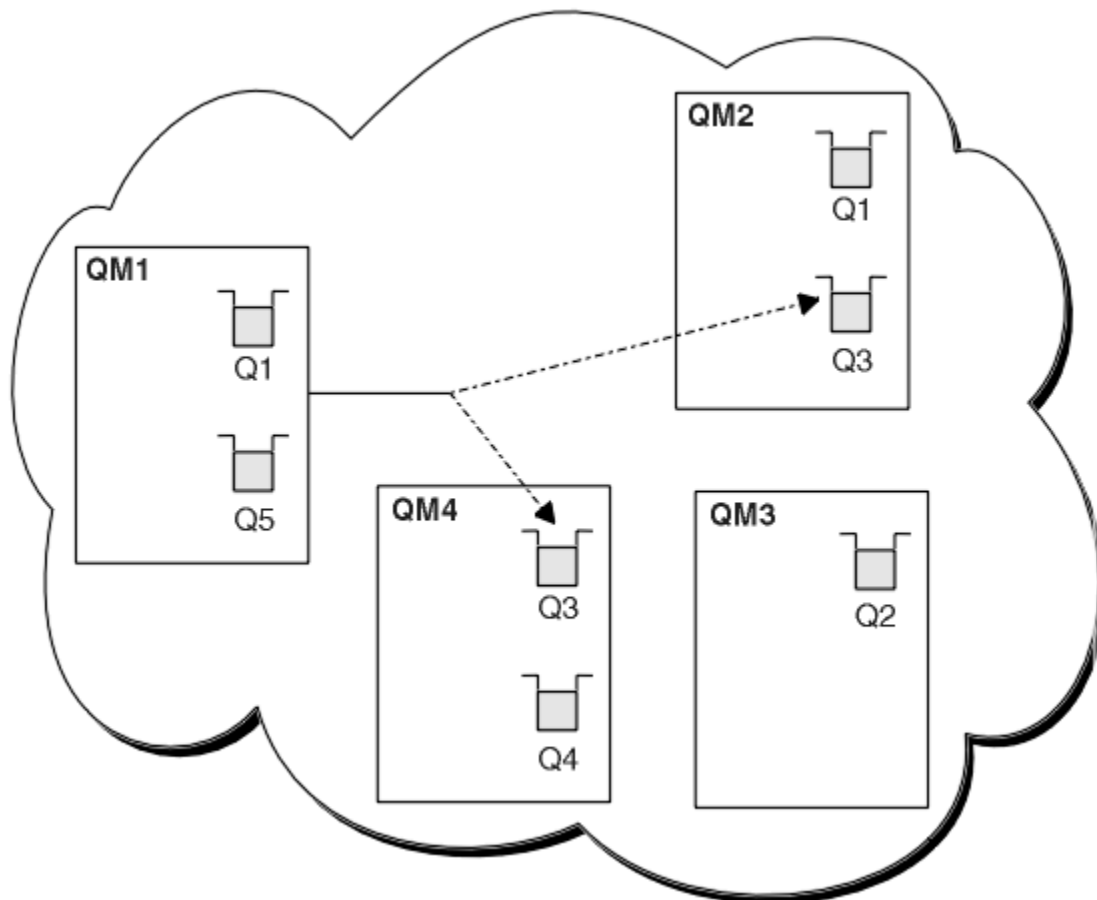


图 49: 具有同一队列的多个实例的集群

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。

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，请参阅第 232 页的『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 WebSphere 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 223 页的图 50 显示此任务设置的集群。

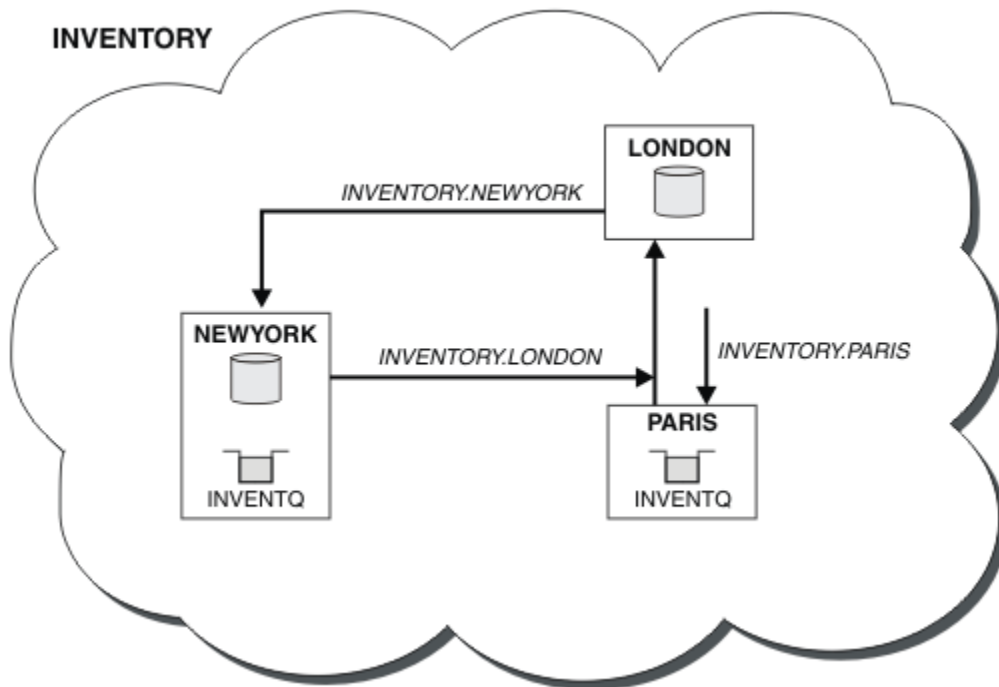


图 50: 具有三个队列管理器的 INVENTORY 集群

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。

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，如果尚未在 WebSphere MQ for z/OS 上启动通道启动程序，请执行此操作。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 225 页的图 51 显示此任务设置的集群。

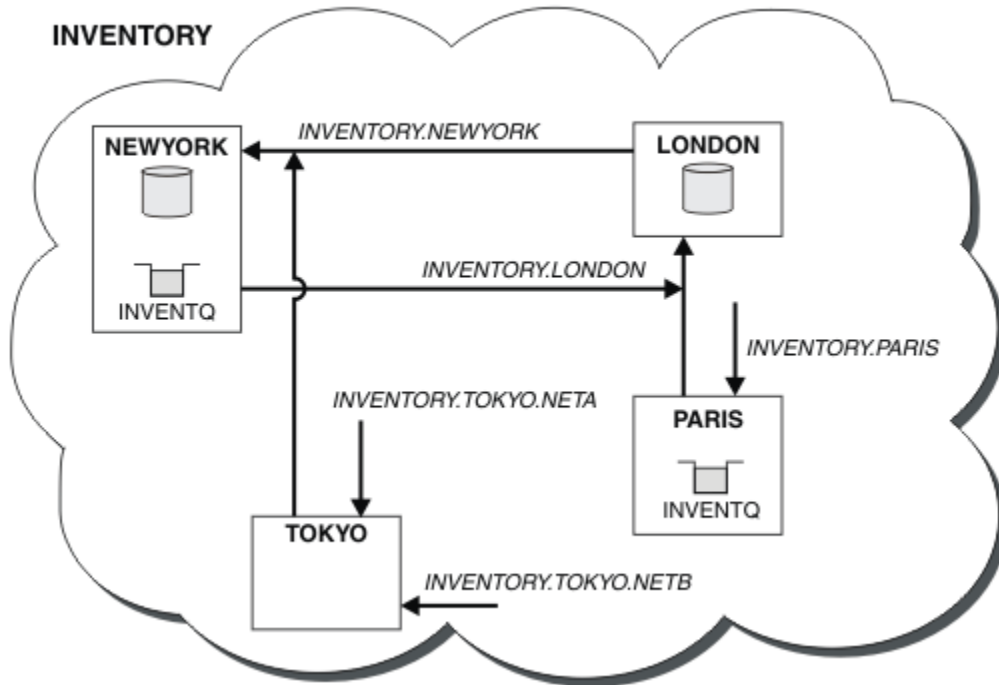


图 51: INVENTORY 集群，具有四个队列管理器

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。

相关任务

第 162 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 224 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。

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 162 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，如果尚未在 WebSphere MQ for z/OS 上启动通道启动程序，请执行此操作。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 227 页的图 52 显示此任务设置的集群。

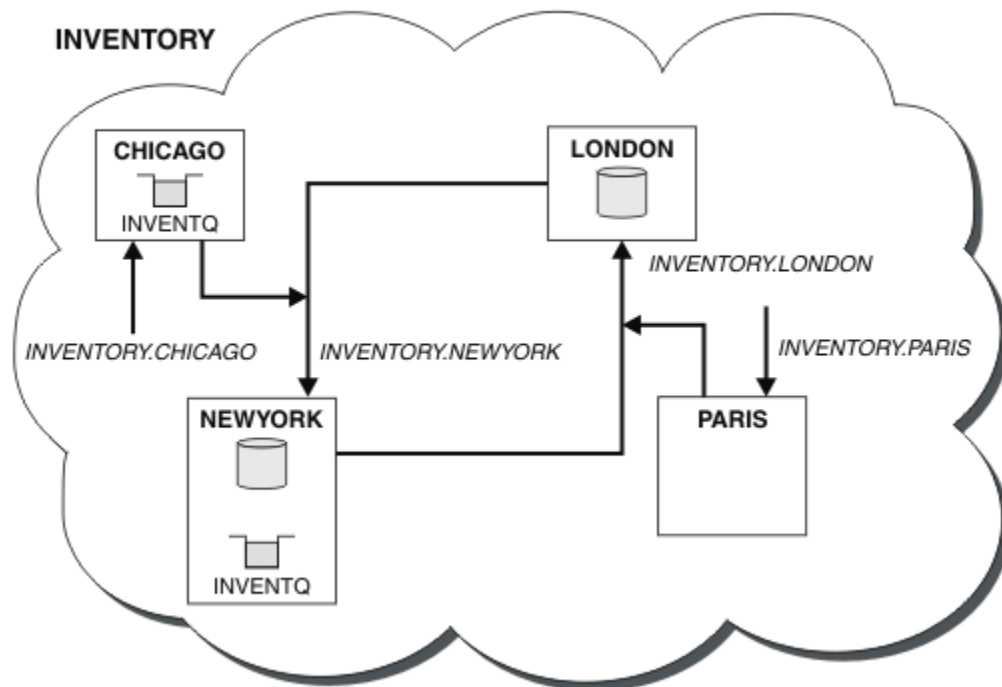


图 52: 具有四个队列管理器的 INVENTORY 集群

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，其中一些状态对集群工作负载管理算法的优先级较低。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。

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)  
CONNNAME(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)  
CONNNAME(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 WebSphere 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 229 页的图 53 显示此任务设置的集群。

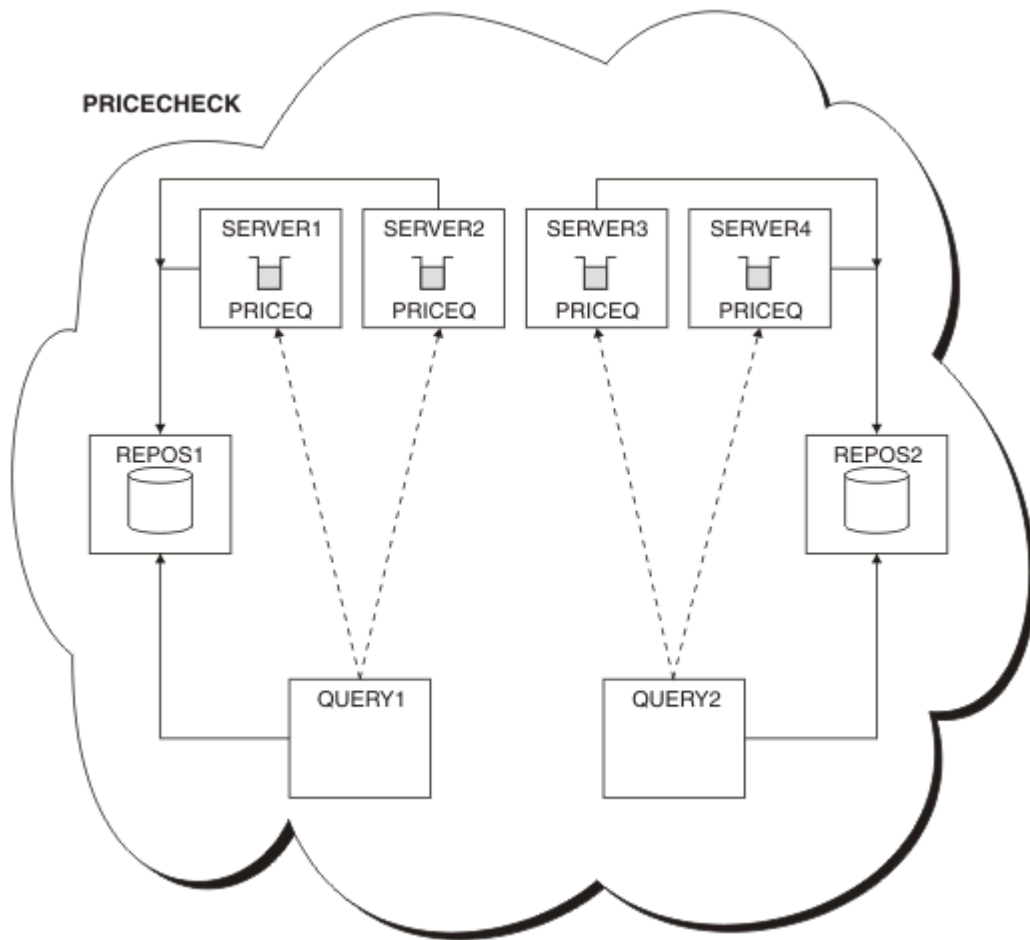


图 53: PRICECHECK 集群，具有四个服务器队列管理器，两个存储库和两个查询队列管理器

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。

下一步做什么

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。

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。

方案:

- 已按第 162 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，如果尚未在 WebSphere MQ for z/OS 上启动通道启动程序，请执行此操作。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 229 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。

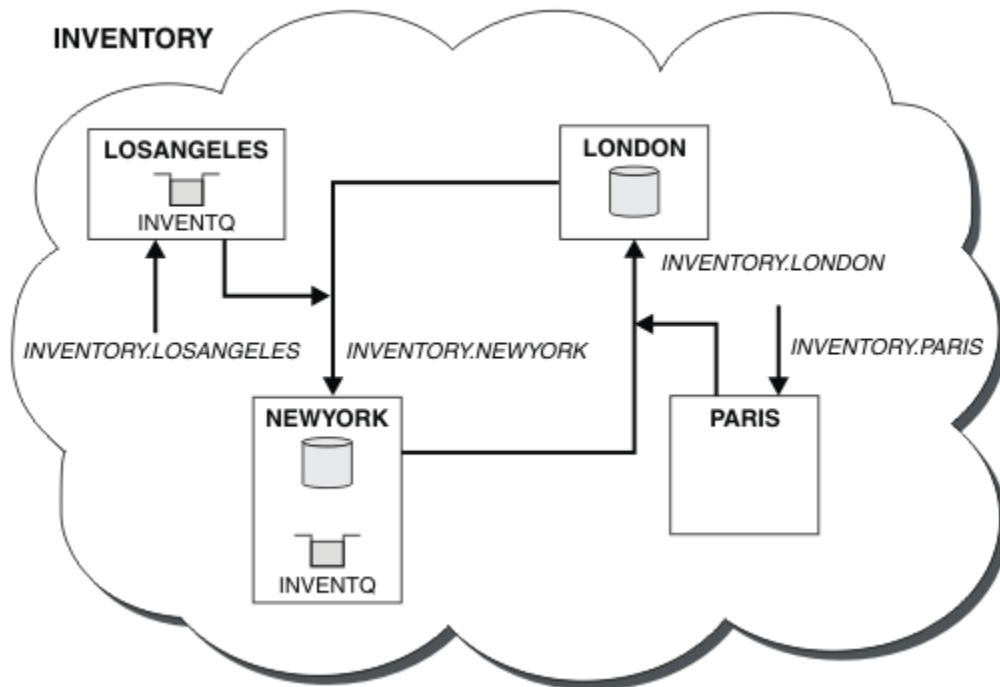


图 54: 具有四个队列管理器的 INVENTORY 集群

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。

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)。

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。

集群: 最佳实践

集群提供了一种用于互连队列管理器的机制。本节中描述的最佳实践基于客户的测试和反馈。

成功的集群设置依赖于良好的规划和对 IBM WebSphere MQ 基础知识 (例如，良好的应用程序管理和网络设计) 的全面了解。请确保您熟悉下面列出的相关主题中的信息，然后再继续。

相关概念

[集群](#)

集群: 重叠集群的特殊注意事项

本主题提供有关规划和管理 IBM WebSphere MQ 集群的指导。此信息是基于测试和客户反馈的指南。

集群所有权

请先熟悉重叠的集群, 然后再阅读以下信息。请参阅 [第 149 页的『重叠集群』](#) 和 [第 216 页的『配置集群之间的消息路径』](#) 以获取必要的信息。

在配置和管理由重叠集群组成的系统时, 最好遵循以下内容:

- 尽管如前所述, IBM WebSphere MQ 集群 "松散耦合", 但将集群视为单个管理单元很有用。使用此概念是因为各个队列管理器上的定义之间的交互对于集群的顺利运行至关重要。例如: 使用工作负载均衡的集群队列时, 单个管理员或团队必须了解消息的完整可能目标集, 这取决于分布在整个集群中的定义。更重要的是, 集群发送方/接收方通道对必须自始至终兼容。
- 考虑到先前的这一概念; 当多个集群满足 (将由单独的团队/个人管理) 时, 重要的是要有明确的策略来控制网关队列管理器的管理。
- 将重叠集群视为单个名称空间很有用: 通道名称和队列管理器名称在单个集群中必须唯一。在整个拓扑中唯一时, 管理更容易。最好遵循合适的命名约定, [第 148 页的『集群命名约定』](#) 中描述了可能的约定。
- 有时, 行政和系统管理合作是不可避免的/不可缺少的: 例如, 拥有需要重叠的不同集群的组织之间的合作。明确了解谁拥有哪些内容和可执行的规则/约定有助于集群在重叠集群时平稳运行。

重叠集群: 网关

通常, 单个集群比多个集群更易于管理。因此, 创建大量小型集群 (例如, 针对每个应用程序创建一个集群) 通常是要避免的。

但是, 要提供服务类, 您可以实现重叠集群。例如:

- 如果您有同心集群, 其中较小的集群用于发布/预订。请参阅 [如何调整系统大小](#) 以获取更多信息。
- 如果某些队列管理器将由不同的团队管理。请参阅上一部分 [第 234 页的『集群所有权』](#) 以获取更多信息。
- 如果从组织或地理角度讲是有意义的。
- 如果等效集群使用名称解析, 例如, 在现有集群中实现 SSL 或 TLS 时。

重叠集群没有安全优势; 允许由两个不同团队管理的集群重叠, 有效地加入团队以及拓扑。任何:

- 在此类集群中公布的名称可供其他集群访问。
- 在一个集群中公布的名称可以在另一个集群中公布, 以提取符合条件的消息。
- 可以从网关所属的任何集群解析与网关相邻的队列管理器上的非广告对象。

名称空间是两个集群的并集, 必须视为单个名称空间。因此, 重叠集群的所有权在两个集群的所有管理员之间共享。

当系统包含多个集群时, 可能需要将消息从一个集群中的队列管理器路由到另一个集群中的队列管理器上的队列。在此情况下, 必须以某种方式互连多个集群: 要遵循的良好模式是在集群之间使用网关队列管理器。这种安排避免了建立起一个难以管理的点对点通道网络, 并为管理安全策略等问题提供了一个好的场所。实现这一安排有两种不同的方法:

1. 使用第二个集群接收方定义在两个集群中放置一个 (或多个) 队列管理器。这种安排涉及较少的管理定义, 但如前所述, 意味着重叠集群的所有权由两个集群的所有管理员共享。
2. 使用传统的点到点通道将集群 1 中的队列管理器与集群 teo 中的队列管理器配对。

在任何一种情况下, 都可以使用各种工具来适当路由流量。特别是, 可以使用队列或队列管理器别名来路由到其他集群, 并且具有空白 **QMNAME** 属性的队列管理器别名会在需要时重新驱动工作负载均衡。

相关概念

第 148 页的『[集群命名约定](#)』

请考虑使用用于标识队列管理器所属集群的命名约定来命名同一集群中的队列管理器。对通道名称使用类似的命名约定，并对其进行扩展以描述通道特征。

集群: 拓扑设计注意事项

本主题提供有关规划和管理 IBM WebSphere MQ 集群的指导。此信息是基于测试和客户反馈的指南。

通过提前思考用户应用程序和内部管理流程的位置，可以避免许多问题，或者在稍后的日期将其最小化。本主题包含有关设计决策的信息，这些决策可以提高性能，并随着集群的扩展而简化维护任务。

- [第 235 页的『集群基础结构的性能』](#)
- [第 235 页的『完整存储库』](#)
- [第 236 页的『应用程序是否应该在完整存储库上使用队列?』](#)
- [第 236 页的『管理通道定义』](#)
- [第 237 页的『多个通道上的工作负载均衡』](#)

集群基础结构的性能

当应用程序尝试在集群中的队列管理器上打开队列时，队列管理器会向该队列的完整存储库注册其兴趣，以便它可以了解该队列在集群中的位置。队列位置或配置的任何更新都将由完整存储库自动发送到相关队列管理器。此相关注册在内部称为预订 (这些预订与 IBM WebSphere MQ 中用于发布/预订消息传递的 IBM WebSphere MQ 预订不同)

有关集群的所有信息都将通过每个完整存储库。因此，完整存储库始终在集群中用于管理消息流量。在管理这些预订时，系统资源的高使用率，以及这些预订的传输和生成的配置消息，都可能对集群基础结构造成相当大的负载。在确保尽可能了解并最小化此负载时，需要考虑许多事项：

- 使用集群队列的单个队列管理器越多，系统中的预订就越多，因此发生更改时的管理开销越大，需要通知感兴趣的订户，尤其是在完整存储库队列管理器上。将不必要的流量和完整存储库负载最小化的一种方法是将类似的应用程序 (即，那些使用相同队列的应用程序) 连接到较少数量的队列管理器。
- 除了系统中影响性能的预订数之外，集群对象配置的更改率也会影响性能，例如频繁更改集群队列配置。
- 当队列管理器是多个集群的成员 (即，它是重叠集群系统的一部分) 时，队列中的任何兴趣都会导致对它是其成员的每个集群进行预订，即使相同的队列管理器是多个集群的完整存储库也是如此。此安排会增加系统上的负载，并且是考虑是否需要多个重叠集群 (而不是单个集群) 的一个原因。
- 应用程序消息流量 (即，IBM WebSphere MQ 应用程序发送到集群队列的消息) 不会通过完整存储库到达目标队列管理器。此消息流量直接在消息进入集群的队列管理器与集群队列所在的队列管理器之间发送。因此，不必适应与完整存储库队列管理器相关的应用程序消息流量的高速率，除非完整存储库队列管理器恰好是提及的这两个队列管理器中的任一个。因此，建议不要将完整存储库队列管理器用于集群基础结构负载较大的集群中的应用程序消息流量。

完整存储库

存储库是有关作为集群成员的队列管理器的信息集合。托管集群中每个队列管理器的完整信息集的队列管理器具有完整的存储库。有关完整存储库和部分存储库的更多信息，请参阅 [第 137 页的『集群存储库』](#)。

完整存储库必须保存在可靠且尽可能高可用性的服务器上，并且必须避免单点故障。集群设计必须始终具有两个完整存储库。如果完整存储库发生故障，那么集群仍可运行。

集群中的队列管理器对集群资源进行的任何更新的详细信息；例如，集群队列从该队列管理器发送到该集群中最多两个完整存储库 (如果集群中只有一个完整存储库队列管理器，那么发送到一个完整存储库)。这些完整存储库保存信息，并将其传播到集群中对其感兴趣的任何队列管理器 (即，他们预订该信息)。要确保集群的每个成员都具有其中的集群资源的最新视图，每个队列管理器必须能够在任何时候与至少一个完整存储库队列管理器进行通信。

如果由于任何原因，队列管理器无法与任何完整存储库通信，那么它可以根据其已高速缓存的信息级别在一段时间内继续在集群中运行，但没有新的更新或对先前未使用的集群资源的访问权。

因此，您必须始终保持这两个完整存储库可用。但是，这一安排并不意味着必须采取极端措施，因为集群在没有完整存储库的短时间内充分运作。

还有一个原因是集群必须具有两个完整存储库队列管理器(集群信息的可用性除外):此原因是为了确保保存在完整存储库高速缓存中的集群信息存在于两个位置以进行恢复。如果只有一个完整存储库，并且它丢失了有关集群的信息，那么需要对集群中的所有队列管理器进行手动干预才能使集群再次工作。但是，如果有两个完整存储库，那么由于始终将信息发布到两个完整存储库并从两个完整存储库预订信息，因此可以尽最大努力恢复失败的完整存储库。

- 可以在两个完整存储库集群设计中对完整存储库队列管理器执行维护，而不影响该集群的用户:该集群仅使用一个存储库继续运行，因此在可能的情况下，请关闭存储库，应用维护，然后一次备份一个存储库。即使在第二个完整存储库上发生中断，运行中的应用程序也至少不受影响三天。
- 除非有充分的理由使用第三个存储库，例如出于地理原因使用地理上的本地完整存储库，否则请使用两个存储库设计。具有三个完整存储库意味着您永远不知道当前正在使用的两个存储库，并且可能存在由多个工作负载管理参数之间的交互所导致的管理问题。建议不要有两个以上的完整存储库。
- 如果仍需要更好的可用性，请考虑将完整存储库队列管理器作为多实例队列管理器托管，或者使用特定于平台的高可用性支持来提高其可用性。
- 您必须将所有完整存储库队列管理器与手动定义的集群发送方通道完全互连。当由于某种合理原因，集群具有两个以上完整存储库时，必须特别小心。在这种情况下往往可能错过一个或多个渠道，为的是不至于立竿见影。当完全互联不发生时，往往会出现难以诊断的问题。它们很难诊断，因为某些完整存储库未保存所有存储库数据，因此导致集群中的队列管理器具有不同的集群视图，具体取决于它们所连接的完整存储库。

应用程序是否应该在完整存储库上使用队列?

完整存储库在大多数方面与任何其他队列管理器完全相同，因此可以在完整存储库上托管应用程序队列，并将应用程序直接连接到这些队列管理器。应用程序是否应该在完整存储库上使用队列?

普遍接受的答案是“不”。虽然可以进行此配置，但许多客户希望保留这些队列管理器以专用于维护完整的存储库集群高速缓存。此处描述了在决定任一选项时要考虑的点，但最终集群体系结构必须适合环境的特定需求。

- 升级:通常，为了在 IBM WebSphere MQ 的新发行版中使用新的集群功能，必须首先升级该集群的完整存储库队列管理器。当集群中的应用程序想要使用新功能时，能够更新完整存储库(以及部分存储库的部分子集)而无需测试多个共存的应用程序可能很有用。
- 维护:以类似方式，如果必须对完整存储库应用紧急维护，那么可以使用 **REFRESH** 命令重新启动或刷新这些存储库，而无需接触应用程序。
- 性能:随着集群的增长以及对完整存储库集群高速缓存维护的需求越来越大，使应用程序保持独立会降低通过争用系统资源而影响应用程序性能的风险。
- 硬件需求:通常，完整存储库不需要强大;例如，具有良好可用性期望的简单 UNIX 服务器就足够了。或者，对于非常大或不断变化的集群，必须考虑完整存储库计算机的性能。
- 软件需求:需求通常是选择在完整存储库上托管应用程序队列的主要原因。在小型集群中，并置可能意味着需要减少所有队列管理器/服务器的数量。

管理通道定义

即使在单个集群中，也可以存在多个通道定义，从而在两个队列管理器之间提供多个路由。

有时在单个集群中具有并行通道的优势，但必须彻底考虑此设计决策;除了增加复杂性外，此设计还可能导致通道利用率不足，从而降低性能。发生此情况是因为测试通常涉及以恒定速率发送大量消息，因此完全使用并行通道。但在非恒定消息流的现实世界条件下，随着消息流从通道切换到通道，工作负载均衡算法会导致性能下降。

当队列管理器是多个集群的成员时，存在将单个通道定义与集群名称列表配合使用的选项，而不是为每个集群定义单独的 CLUSRCVR 通道。但是，此设置可能会在以后导致管理困难;例如，考虑将 SSL 应用于一个集群而不应用于另一个集群的情况。因此，最好创建单独的定义，并且第 148 页的『[集群命名约定](#)』中建议的命名约定支持此操作。

多个通道上的工作负载均衡

此信息旨在作为对主题的高级理解。有关此主题的基本说明 (在使用此处的信息之前必须了解), 请参阅 [第 220 页的『使用集群进行工作负载管理』](#), [工作负载均衡](#) 和 [集群工作负载管理算法](#)。

集群工作负载管理算法提供了一组庞大的工具, 但如果不充分了解它们的工作和交互方式, 它们就不能全部相互配合使用。对于工作负载均衡过程, 通道的重要性可能并不是很明显: 工作负载管理循环法算法的行为就像拥有集群队列的队列管理器的多个集群通道被视为该队列的多个实例一样。以下示例更详细地说明了此过程:

1. 集群中有两个托管队列的队列管理器: QM1 和 QM2。
2. 有五个集群接收方通道到 QM1。
3. 只有一个集群接收方通道到 QM2。
4. 当 QM3 上的 MQPUT 或 MQOPEN 选择实例时, 算法将消息发送到 QM1 的可能性比发送到 QM2 的可能性高五倍。
5. 发生步骤 4 中的情况是因为算法看到六个选项可供选择 (5 + 1), 并在所有五个通道之间循环至 QM1 和单个通道至 QM2。

另一个微妙的行为是, 即使将消息放入正好在本地队列管理器上配置了一个实例的集群队列, IBM WebSphere MQ 也会使用本地集群接收方通道的状态来决定是将消息放入队列的本地实例还是队列的远程实例。在此场景中:

1. 在放置消息时, 工作负载管理算法不会查看个别集群队列, 而是会查看可到达这些目标的集群通道。
2. 要到达本地目标, 本地接收方通道将包括在此列表中 (尽管它们不用于发送消息)。
3. 当本地接收方通道停止时, 工作负载管理算法在缺省情况下首选备用实例 (如果其 CLUSRCVR 未停止)。如果目标有多个本地 CLUSRCVR 实例, 并且至少有一个实例未停止, 那么本地实例仍符合条件。

集群: 使用多个集群传输队列进行应用程序隔离

您可以在集群中的队列管理器之间隔离消息流。您可以将由不同集群发送方通道传输的消息放入不同的集群传输队列中。您可以在单个集群中使用此方法, 也可以将此方法与重叠的集群配合使用。本主题提供了一些示例和一些最佳实践, 以指导您选择使用方法。

部署应用程序时, 您可以选择它与其他应用程序共享哪些 IBM WebSphere MQ 资源以及它不共享哪些资源。可以共享多种类型的资源, 主要资源是服务器本身, 队列管理器, 通道和队列。您可以选择配置共享资源较少的应用程序; 将单独的队列, 通道, 队列管理器甚至服务器分配给各个应用程序。如果执行此操作, 那么整体系统配置将变得更大且更复杂。使用 IBM WebSphere MQ 集群可降低管理更多服务器, 队列管理器, 队列和通道的复杂性, 但它会引入另一个共享资源, 即集群传输队列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。

[第 238 页的图 55](#) 是大型 IBM WebSphere MQ 部署中的一个切片, 用于说明共享 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 的重要性。在图中, 应用程序 Client App 已连接到集群 CL1 中的队列管理器 QM2。来自 Client App 的消息由应用程序 Server App 处理。此消息由 Server App 从 CLUSTER2 中队列管理器 QM3 上的集群队列 Q1 中检索。由于客户机和服务器应用程序不在同一集群中, 因此消息由网关队列管理器 QM1 传输。

配置集群网关的正常方法是使网关队列管理器成为所有集群的成员。在网关队列管理器上, 为所有集群中的集群队列定义了集群别名队列。集群队列别名在所有集群中都可用。放入集群队列别名的消息通过网关队列管理器路由到其正确目标。网关队列管理器将发送到集群别名队列的消息放到 QM1 上的公共 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上。

中心和辐射体系结构要求集群之间的所有消息都通过网关队列管理器。结果是所有消息都流经 QM1 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上的单集群传输队列。

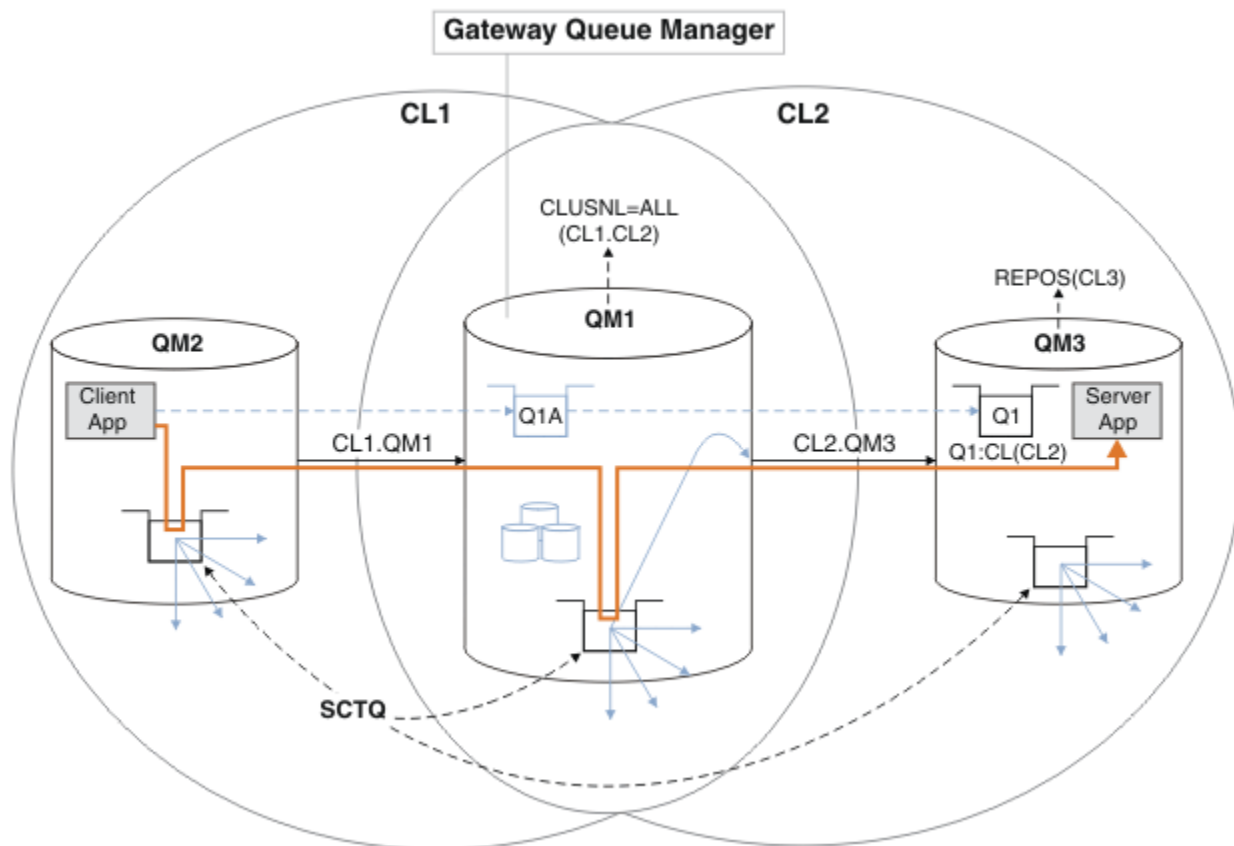
从性能角度来看, 单个队列不是问题。公共传输队列通常不表示性能瓶颈。网关上的消息吞吐量在很大程度上取决于与其连接的通道的性能。吞吐量通常不受队列数或使用通道的队列上的消息数的影响。

从其他一些角度来看, 将单个传输队列用于多个应用程序有缺点:

- 不能将消息流到一个目标与消息流到另一个目标隔离。即使目标位于不同队列管理器上的不同集群中, 也无法在转发消息之前将其存储分开。

如果一个集群目标变为不可用，那么将在单个传输队列中构建该目标的消息，并最终填充该消息。一旦传输队列已满，它就会停止将消息放置到任何集群目标的传输队列上。

- 监视消息到不同集群目标的传输并不容易。所有消息都在单个传输队列上。显示传输队列的深度几乎不会指示是否将消息传输到所有目标。



注: 第 238 页的图 55 和下图中的箭头具有不同类型。实心箭头表示消息流。实线箭头上的标签是消息通道名称。灰色实线箭头是从 SYSTEM.CLUSTER.TRANSMIT.QUEUE 到集群发送方通道的潜在消息流。黑色虚线将标签连接到其目标。灰色虚线箭头是引用; 例如, 从 MQOPEN 调用 Client App 到集群别名队列定义 Q1A。

图 55: 使用 IBM WebSphere 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

在第 238 页的图 55 中, Server App 的客户机打开队列 Q1A。消息将放置到 QM2 上的 SYSTEM.CLUSTER.TRANSMIT.QUEUE, 传输到 QM1 上的 SYSTEM.CLUSTER.TRANSMIT.QUEUE, 然后传输到 QM3 上的 Q1, Server App 应用程序将在此接收到这些消息。

来自 Client App 的消息通过 QM2 和 QM1 上的系统集群传输队列传递。在第 238 页的图 55 中, 目标是将网关队列管理器上的消息流与客户机应用程序隔离, 以便其消息不会存储在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上。您可以隔离任何其他集群队列管理器上的流。您还可以将其他方向的流隔离回客户机。为了使解决方案的描述简短, 这些描述仅考虑来自客户机应用程序的单个流。

用于隔离集群网关队列管理器上的集群消息流量的解决方案

解决问题的一种方法是使用队列管理器别名或远程队列定义在集群之间进行桥接。创建集群远程队列定义, 传输队列和通道, 以分隔网关队列管理器上的每个消息流; 请参阅第 166 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』。

从 Version 7.5 开始, 集群队列管理器不限于单个集群传输队列。您有两种选择:

1. 手动定义其他集群传输队列, 并定义哪些集群发送方通道从每个传输队列传输消息; 请参阅第 168 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』。

2. 允许队列管理器自动创建和管理其他集群传输队列。它为每个集群发送方通道定义不同的集群传输队列；请参阅第 186 页的『[将缺省值更改为单独的集群传输队列以隔离消息流量](#)』

您可以将某些集群发送方通道的手动定义的集群传输队列与队列管理器一起管理其余的集群传输队列。传输队列的组合是第 168 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](#)』中采用的方法。在该解决方案中，集群之间的大多数消息都使用公共 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。一个应用程序是关键的，它的所有消息流都通过使用一个手动定义的集群传输队列与其他流隔离。

第 168 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](#)』中的配置受限。它不会将消息流量与另一个集群队列分隔到同一集群中同一队列管理器上的集群队列。您可以使用属于分布式队列的远程队列定义来分隔到各个队列的消息流量。借助集群，通过使用多个集群传输队列，您可以将进入不同集群发送方通道的消息流量分开。同一集群中的多个集群队列在同一队列管理器上共享一个集群发送方通道。这些队列的消息存储在同一传输队列上，然后再从网关队列管理器转发。在第 171 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](#)』中的配置中，通过添加另一个集群并使队列管理器和集群队列成为新集群的成员来规避此限制。新的队列管理器可能是集群中唯一的队列管理器。您可以向集群添加更多队列管理器，并使用同一集群来隔离这些队列管理器上的集群队列。

相关概念

第 134 页的『[访问控制和多个集群传输队列](#)』

在应用程序将消息放入远程集群队列时选择三种检查方式。这些方式是针对集群队列进行远程检查，针对 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 进行本地检查，或者针对集群队列或集群队列管理器的本地概要文件进行检查。

第 234 页的『[集群: 重叠集群的特殊注意事项](#)』

本主题提供有关规划和管理 IBM WebSphere MQ 集群的指导。此信息是基于测试和客户反馈的指南。

第 142 页的『[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。

第 149 页的『[重叠集群](#)』

重叠的集群提供了额外的管理功能。使用名称列表来减少管理重叠集群所需的命令数。

相关任务

授权将消息放入远程集群队列

第 166 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](#)』

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。

第 168 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 171 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。

第 186 页的『[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。

第 181 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 216 页的『[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 240 页的『[集群: 规划如何配置集群传输队列](#)』

将指导您选择集群传输队列。您可以配置一个公共缺省队列，单独的缺省队列或手动定义的队列。配置多个集群传输队列适用于除 z/OS 以外的平台。

相关参考

第 358 页的『安全性』
使用 `qm.ini` 文件中的 `Security` 节来指定对象权限管理器 (OAM) 的选项。

[塞特 MQaut](#)

集群: 规划如何配置集群传输队列

将指导您选择集群传输队列。您可以配置一个公共缺省队列, 单独的缺省队列或手动定义的队列。配置多个集群传输队列适用于除 z/OS 以外的平台。

开始之前

复审 [第 243 页的『如何选择要使用的集群传输队列类型』](#)。

关于此任务

当您计划如何配置队列管理器以选择集群传输队列时, 可以选择一些选项。

1. 什么是集群消息传输的缺省集群传输队列?
 - a. 公共集群传输队列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。
 - b. 单独的集群传输队列。队列管理器管理单独的集群传输队列。它会将它们创建为模型队列 `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` 中的永久动态队列。它为其使用的每个集群发送方通道创建一个集群传输队列。
2. 对于您决定手动创建的集群传输队列, 还有两个选项:
 - a. 为您决定手动配置的每个集群发送方通道定义单独的传输队列。在这种情况下, 请将传输队列的 **CLCHNAME** 队列属性设置为集群发送方通道的名称。选择要从此传输队列传输消息的集群发送方通道。
 - b. 将一组集群发送方通道的消息流量组合到同一集群传输队列上; 请参阅 [第 241 页的图 56](#)。在这种情况下, 请将每个公共传输队列的 **CLCHNAME** 队列属性设置为通用集群发送方通道名称。通用集群发送方通道名称是用于对集群发送方通道名称进行分组的过滤器。例如, `SALES.*` 对名称以 `SALES.` 开头的所有集群发送方通道进行分组。可以在 `filter-string` 中的任何位置放置多个通配符。通配符为星号 `*`。它表示从零到任意数目的字符。

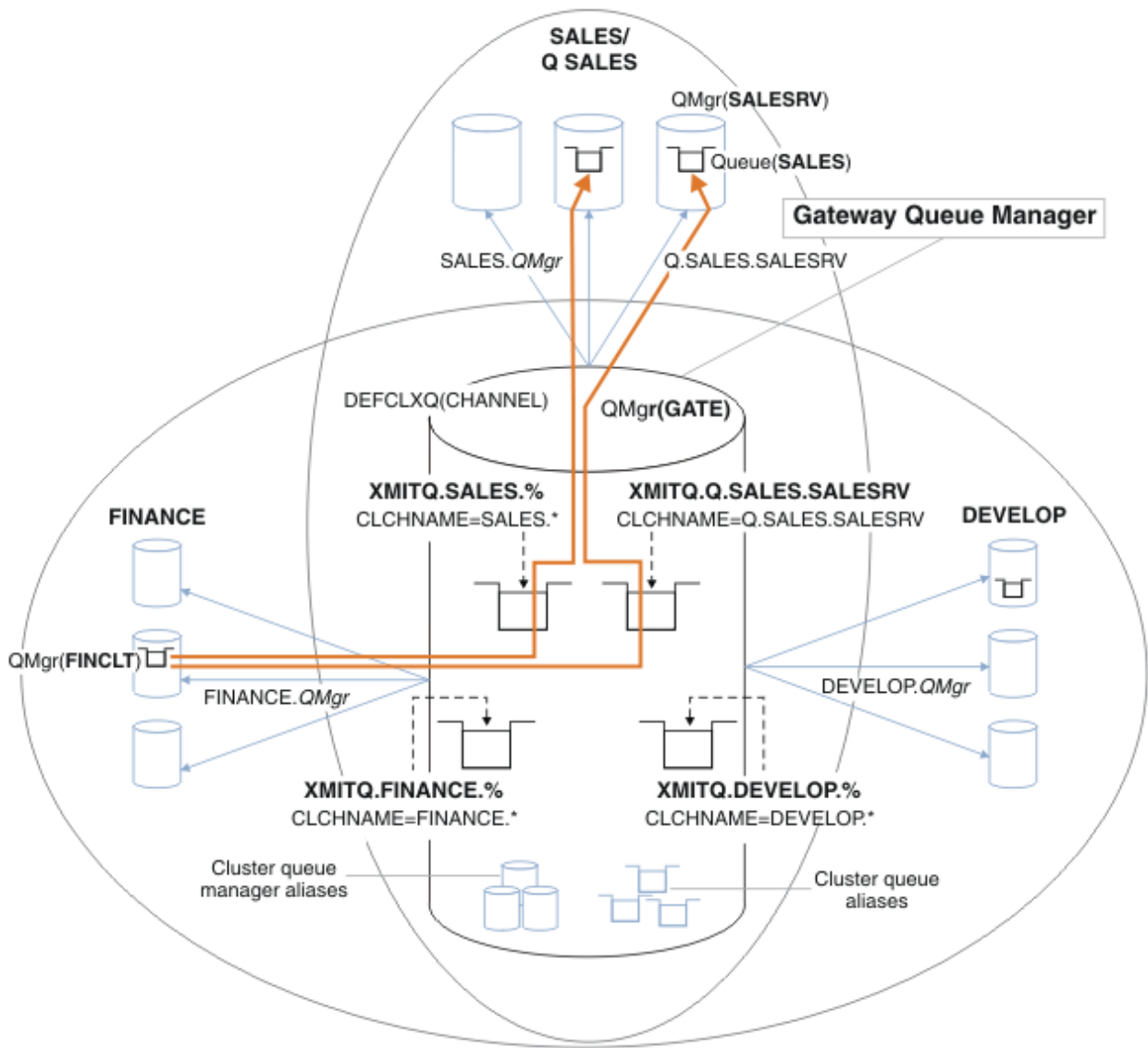


图 56: 不同部门 IBM WebSphere MQ 集群的特定传输队列示例

过程

1. 选择要使用的缺省集群传输队列类型。

- 选择单个集群传输队列，或者为每个集群连接选择单独的队列。

保留缺省设置或运行 **MQSC** 命令：

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. 隔离不得与其他流共享集群传输队列的任何消息流。

- 请参阅第 244 页的『集群: 多个集群传输队列的示例配置』。在此示例中，必须隔离的 SALES 队列是 SALESRV 上 SALES 集群的成员。要隔离 SALES 队列，请创建新集群 Q.SALES，使 SALESRV 队列管理器成为成员，并修改 SALES 队列以属于 Q.SALES。
- 向 SALES 发送消息的队列管理器也必须是新集群的成员。如果使用集群队列别名和网关队列管理器，例如，在许多情况下，可以将更改限制为使网关队列管理器成为新集群的成员。
- 但是，将流从网关分离到目标不会将流从源队列管理器分离到网关。但有时事实证明，这足以将流与网关分开，而不是流向网关。如果不够，请将源队列管理器添加到新集群中。如果希望消息通过网关传递，请将集群别名移动到新集群，并继续将消息发送到网关上的集群别名，而不是直接发送到目标队列管理器。

执行以下步骤以隔离消息流:

- a) 配置流的目标, 以便每个目标队列都是该队列管理器上特定集群中的唯一队列。
 - b) 为遵循系统命名约定创建的任何新集群创建集群发送方通道和集群接收方通道。
 - 请参阅第 234 页的『[集群: 重叠集群的特殊注意事项](#)』。
 - c) 为每个将消息发送到目标队列的队列管理器上的每个隔离目标定义集群传输队列。
 - 集群传输队列的命名约定是使用以 XMITQ. 为前缀的集群通道名称属性 CLCHNAME 的值
3. 创建集群传输队列以满足监管或监视需求。
- 典型的监管和监视需求会导致每个集群的传输队列或每个队列管理器的传输队列。如果遵循集群通道 `ClusterName.QueueManagerName` 的命名约定, 那么可以轻松创建用于选择队列管理器集群或队列管理器所属的所有集群的通用通道名称; 请参阅第 244 页的『[集群: 多个集群传输队列的示例配置](#)』。
 - 通过将星号替换为百分号, 扩展集群传输队列的命名约定以迎合通用通道名称。例如,

```
DEFINE QLOCAL(XMITQ.SALES.%) USAGE(XMITQ) CLCHNAME(SALES.*)
```

相关概念

[第 142 页的『\[集群传输队列和集群发送方通道\]\(#\)』](#)

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。

[第 237 页的『\[集群: 使用多个集群传输队列进行应用程序隔离\]\(#\)』](#)

您可以在集群中的队列管理器之间隔离消息流。您可以将由不同集群发送方通道传输的消息放入不同的集群传输队列中。您可以在单个集群中使用此方法, 也可以将此方法与重叠的集群配合使用。本主题提供了一些示例和一些最佳实践, 以指导您选择使用方法。

[第 134 页的『\[访问控制和多个集群传输队列\]\(#\)』](#)

在应用程序将消息放入远程集群队列时选择三种检查方式。这些方式是针对集群队列进行远程检查, 针对 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 进行本地检查, 或者针对集群队列或集群队列管理器的本地概要文件进行检查。

[第 234 页的『\[集群: 重叠集群的特殊注意事项\]\(#\)』](#)

本主题提供有关规划和管理 IBM WebSphere MQ 集群的指导。此信息是基于测试和客户反馈的指南。

[第 149 页的『\[重叠集群\]\(#\)』](#)

重叠的集群提供了额外的管理功能。使用名称列表来减少管理重叠集群所需的命令数。

相关任务

[第 166 页的『\[添加远程队列定义以隔离从网关队列管理器发送的消息\]\(#\)』](#)

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。

[第 168 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 171 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。

[第 186 页的『\[将缺省值更改为单独的集群传输队列以隔离消息流量\]\(#\)』](#)

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。

[第 181 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 216 页的『\[配置集群之间的消息路径\]\(#\)』](#)

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。

如何选择要使用的集群传输队列类型

如何在不同的集群传输队列配置选项之间进行选择。

从 Version 7.5 开始，您可以选择哪个集群传输队列与集群发送方通道相关联。

1. 您可以使所有集群发送方通道与单个缺省集群传输队列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 相关联。此选项是缺省值，并且是运行版本 Version 7.1 或更低版本的队列管理器的唯一选项。
2. 您可以将所有集群发送方通道设置为自动与单独的集群传输队列相关联。队列由队列管理器从模型队列 `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` 创建，命名为 `SYSTEM.CLUSTER.TRANSMIT.ChannelName`。如果队列管理器属性 `DEFCLXQ` 设置为 `CHANNEL`，那么通道将使用其唯一命名的集群传输队列。



注意: 如果要专用 `SYSTEM.CLUSTER.TRANSMIT.QUEUES` 与从较低版本的产品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`。

3. 您可以设置要由单个集群传输队列提供服务的特定集群发送方通道。通过创建传输队列并将其 `CLCHNAME` 属性设置为集群发送方通道的名称来选择此选项。
4. 您可以选择要由单个集群传输队列提供服务的集群发送方通道组。通过创建传输队列并将其 `CLCHNAME` 属性设置为通用通道名称 (例如 `ClusterName.*`) 来选择此选项。如果通过遵循第 234 页的『[集群: 重叠集群的特殊注意事项](#)』中的命名约定来命名集群通道，那么此名称将选择连接到集群 `ClusterName` 中的队列管理器的所有集群通道。

您可以将某些集群发送方通道的任一缺省集群传输队列选项与任意数量的特定和通用集群传输队列配置组合在一起。

最佳实践

在大多数情况下，对于现有 IBM WebSphere MQ 安装，缺省配置是最佳选择。集群队列管理器将集群消息存储在单个集群传输队列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上。您可以选择将缺省值更改为将不同队列管理器和不同集群的消息存储在不同的传输队列上，或者定义您自己的传输队列。

在大多数情况下，对于新的 IBM WebSphere MQ 安装，缺省配置也是最佳选择。从缺省配置切换到针对每个集群发送方通道具有一个传输队列的备用缺省值的过程是自动的。切换回也是自动的。一个或另一个的选择并不关键，你可以逆转它。

选择其他配置的原因更多是与监管和管理有关，而不是与功能或性能有关。如果有几个异常，那么配置多个集群传输队列对队列管理器的行为没有好处。这将生成更多队列，并要求您修改已设置的监视和管理过程，这些过程将引用单个传输队列。这就是为什么在平衡上，保留缺省配置是最佳选择，除非您有强大的治理或管理原因进行不同的选择。

这些异常都与 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上存储的消息数增加时发生的情况相关。如果您执行每个步骤来将一个目标的消息与另一个目标的消息分开，那么一个目标的通道和传递问题不应影响到另一个目标的传递。但是，由于无法以足够快的速度将消息传递到一个目标，因此存储在 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上的消息数可能会增加。`SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上针对一个目标的消息数可能会影响将消息传递到其他目标。

要避免由于填充单个传输队列而导致的问题，请将足够的容量构建到配置中。然后，如果目标失败并且消息积压开始构建，那么您有时间来解决该问题。

如果通过中心队列管理器 (例如集群网关) 路由消息，那么它们共享公共传输队列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。如果网关队列管理器上存储在 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上的消息数达到其最大深度，那么队列管理器将开始拒绝传输队列的新消息，直到其深度减小为止。拥塞会影响通过网关路由的所有目标的消息。消息备份正在向网关发送消息的其他队列管理器的传输队列。该问题表现在写入队列管理器错误日志的消息中，消息吞吐量下降，以及从发送消息到消息到达其目标的时间之间的耗用时间较长。

拥堵对单个传输队列的影响会变得明显，即使在它已满之前也是如此。如果您有混合的消息流量，有一些大的非持久消息和一些小的消息，那么随着传输队列的填满，传递小消息的时间会增加。延迟是由于将通常不会写入磁盘的大型非持久消息写入磁盘。如果您具有时间关键消息流，并且与其他混合消息流共享集群传输队列，那么可能需要配置特殊消息路径以将其与其他消息流隔离; 请参阅第 171 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](#)』。

配置单独的集群传输队列的其他原因是为了满足监管要求，或者为了简化发送到不同集群目标的监视消息。例如，您可能必须证明一个目标的消息从不与另一个目标的消息共享传输队列。

更改用于控制缺省集群传输队列的队列管理器属性 **DEFCLXQ**，以便为每个集群发送方通道创建不同的集群传输队列。多个目标可以共享一个集群发送方通道，因此您必须规划集群以完全满足此目标。系统地将方法第 171 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』应用于所有集群队列。您的目标结果是没有集群目标与另一个集群目标共享集群发送方通道。因此，集群目标的任何消息都不会与另一个目标的消息共享其集群传输队列。

为某些特定消息流创建单独的集群传输队列，可以轻松监视到该目标的消息流。要使用新的集群传输队列，请定义该队列，将其与集群发送方通道相关联，然后停止并启动该通道。此更改不必是永久性的。您可以将消息流隔离一段时间，以监视传输队列，然后再次还原为使用缺省传输队列。

相关任务

集群: 多个集群传输队列的示例配置

在此任务中，您将应用步骤以将多个集群传输队列规划到三个重叠的集群。这些要求是将消息流与所有其他消息流分离到一个集群队列，并将不同集群的消息存储在不同的集群传输队列上。

集群: 切换集群传输队列

规划如何使对现有生产队列管理器的集群传输队列所作的更改生效。

集群: 多个集群传输队列的示例配置

在此任务中，您将应用步骤以将多个集群传输队列规划到三个重叠的集群。这些要求是将消息流与所有其他消息流分离到一个集群队列，并将不同集群的消息存储在不同的集群传输队列上。

关于此任务

此任务中的步骤显示如何应用第 240 页的『集群: 规划如何配置集群传输队列』中的过程并得出第 245 页的图 57 中显示的配置。它是具有网关队列管理器的三个重叠集群的示例，该集群配置有单独的集群传输队列。第 247 页的『创建示例集群』中描述了用于定义集群的 MQSC 命令。

对于此示例，有两个需求。一是将消息流从网关队列管理器分离到记录销售的销售应用程序。二是查询有多少消息等待在任何时间点发送到不同的部门地区。已定义 SALES, FINANCE 和 DEVELOP 集群。集群消息当前从 SYSTEM.CLUSTER.TRANSMIT.QUEUE 转发。

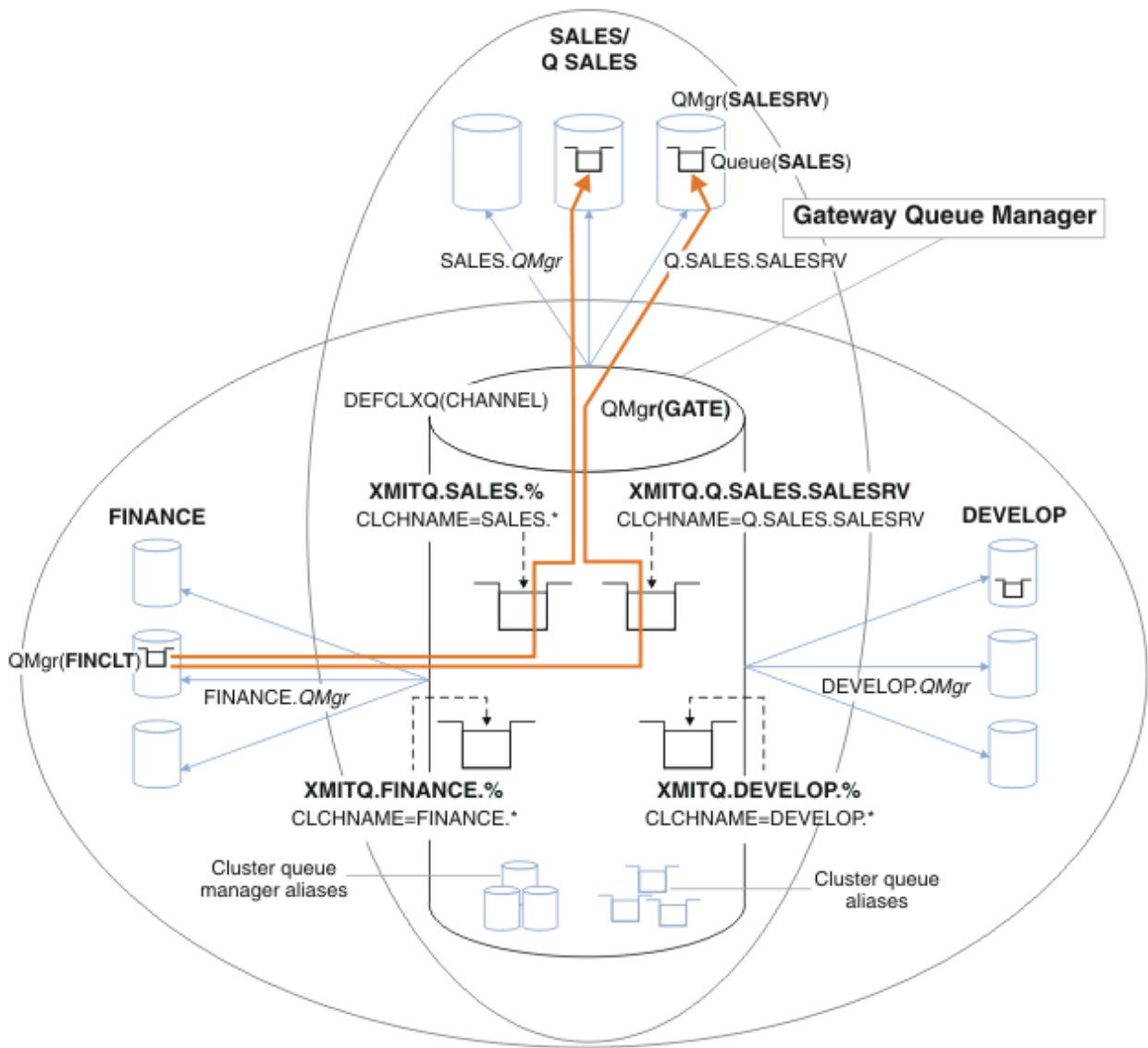


图 57: 不同部门 IBM WebSphere MQ 集群的特定传输队列示例

修改集群的步骤如下所示: 请参阅 [更改以隔离新集群中的销售队列并分隔网关集群传输队列](#) 以获取定义。

过程

1. 第一个配置步骤是“选择要使用的缺省集群传输队列类型”。

决定是通过在 GATE 队列管理器上运行以下 **MQSC** 命令来创建单独的缺省集群传输队列。

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

没有强烈的理由选择此缺省值, 因为目的是手动定义集群传输队列。该选项确实具有弱诊断值。如果手动定义错误, 并且消息在缺省集群传输队列中流下, 那么会在创建永久动态集群传输队列时显示该消息。

2. 第二个配置步骤是“隔离不得与其他流共享集群传输队列的任何消息流”。

在这种情况下, 从 SALESRV 上的队列 SALES 接收消息的销售应用程序需要隔离。仅需要隔离来自网关队列管理器的消息。三个分步骤是:

- a) “配置流的目标, 以便每个目标队列都是该队列管理器上特定集群中的唯一队列”。

此示例需要将队列管理器 SALESRV 添加到销售部门中的新集群。如果您有几个需要隔离的队列, 那么可以决定为 SALES 队列创建特定集群。集群名称的可能命名约定是将此类集群命名为 Q。

QueueName, 例如 Q.SALES。如果要隔离大量队列, 那么可能更实际的替代方法是在需要时创建隔离队列的集群。集群名称可能为 QUEUES.n。

在此示例中, 新集群称为 Q.SALES。要添加新集群, 请参阅 [更改以隔离新集群中的销售队列并分隔网关集群传输队列中的定义](#)。定义更改的摘要如下:

- i) 将 Q.SALES 添加到存储库队列管理器上的集群的名称列表。在队列管理器 **REPOSNL** 参数中引用了名称列表。
- ii) 将 Q.SALES 添加到网关队列管理器上的集群的名称列表。在网关队列管理器上的所有集群队列别名和集群队列管理器别名定义中都引用了名称列表。
- iii) 在队列管理器 SALESRV 上为其所属的两个集群创建名称列表, 并更改 SALES 队列的集群成员资格:

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES) REPLACE
ALTER QLOCAL(SALES) CLUSTER(' ') CLUSNL(SALESRV.CLUSTERS)
```

SALES 队列是两个集群的成员, 仅用于转换。新配置运行后, 将从 SALES 集群中除去 SALES 队列; 请参阅 [第 249 页的图 58](#)。

- b) “为遵循系统命名约定创建的任何新集群创建集群发送方通道和集群接收方通道”。

- i) 将集群接收方通道 Q.SALES.RepositoryQMGr 添加到每个存储库队列管理器
- ii) 将集群发送方通道 Q.SALES.OtherRepositoryQMGr 添加到每个存储库队列管理器, 以连接到其他存储库管理器。启动这些通道。
- iii) 将集群接收方通道 Q.SALES.SALESRV 和 Q.SALES.GATE 添加到正在运行的任一存储库队列管理器。
- iv) 将集群发送方通道 Q.SALES.SALESRV 和 Q.SALES.GATE 添加到 SALESRV 和 GATE 队列管理器。将集群发送方通道连接到您在其中创建集群接收方通道的存储库队列管理器。

- c) “为每个将消息发送到目标队列的队列管理器上的每个隔离目标定义集群传输队列”。

在网关队列管理器上, 为 Q.SALES.SALESRV 集群发送方通道定义集群传输队列 XMITQ.Q.SALES.SALESRV:

```
DEFINE QLOCAL(XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
```

3. 第三个配置步骤是“创建集群传输队列以满足监管或监视需求”。

在网关队列管理器上定义集群传输队列:

```
DEFINE QLOCAL(XMITQ.SALES) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
DEFINE QLOCAL(XMITQ.DEVELOP) USAGE(XMITQ) CLCHNAME(DEVELOP.*) REPLACE
DEFINE QLOCAL(XMITQ.FINANCE) USAGE(XMITQ) FINANCE(SALES.*) REPLACE
```

下一步做什么

切换到网关队列管理器上的新配置。

通过启动新通道并重新启动现在与不同传输队列相关联的通道来触发交换机。或者, 可以停止并启动网关队列管理器。

1. 停止网关队列管理器上的以下通道:

```
SALES.Qmgr
DEVELOP.Qmgr
FINANCE.Qmgr
```

2. 在网关队列管理器上启动以下通道:

```
SALES.Qmgr
DEVELOP.Qmgr
```

交换机完成后，从 SALES 集群中除去 SALES 队列；请参阅 [第 249 页的图 58](#)。

相关概念

如何选择要使用的集群传输队列类型
如何在不同的集群传输队列配置选项之间进行选择。

相关任务

集群: 切换集群传输队列
规划如何使对现有生产队列管理器的集群传输队列所作的更改生效。

创建示例集群

用于创建示例集群并对其进行修改以隔离 SALES 队列和网关队列管理器上的单独消息的定义和指示信息。

关于此任务

用于创建 FINANCE, SALES 和 Q.SALES 集群的完整 MQSC 命令在 [基本集群的定义](#)，用于隔离新集群中的销售队列并分隔网关集群传输队列的更改和 [从销售集群中除去队列管理器 SALESRV 上的销售队列](#) 中提供。定义中省略了 DEVELOP 集群，以保持定义较短。

过程

1. 创建 SALES 和 FINANCE 集群以及网关队列管理器。

a) 创建队列管理器。

针对 [第 247 页的表 27](#) 中的每个队列管理器名称运行命令: `crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QmgrName`。

描述	队列管理器名称	端口号
财务存储库	FINR1	1414
财务存储库	FINR2	1415
财务客户	FINCLT	1418
销售存储库	SALER1	1416
销售存储库	SALER2	1417
销售服务器	SALESRV	1419
网关	GATE	1420

b) 启动所有队列管理器

对 [第 247 页的表 27](#) 中的每个队列管理器名称运行命令 `strmqm QmgrName`。

c) 为每个队列管理器创建定义

Run the command: `runmqsc QmgrName <filename` where the files are listed in [基本集群的定义](#), and the file name matches the queue manager name.

基本集群的定义

finr1.txt

```
DEFINE LISTENER(1414) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1414) REPLACE
START LISTENER(1414)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
```

finr2.txt

```
DEFINE LISTENER(1415) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1415) REPLACE
START LISTENER(1415)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
```

finclt.txt

```
DEFINE LISTENER(1418) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1418) REPLACE
START LISTENER(1418)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINCLT) CHLTYPE(CLUSRCVR) CONNAME('localhost(1418)')
CLUSTER(FINANCE) REPLACE
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

saler1.txt

```
DEFINE LISTENER(1416) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1416) REPLACE
START LISTENER(1416)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
```

saler2.txt

```
DEFINE LISTENER(1417) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1417) REPLACE
START LISTENER(1417)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
```

salesrv.txt

```
DEFINE LISTENER(1419) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1419) REPLACE
START LISTENER(1419)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(SALES) REPLACE
DEFINE QLOCAL(SALES) CLUSTER(SALES) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

gate.txt

```
DEFINE LISTENER(1420) TRPTYPE(TCP) IPADDR(LOCALHOST) CONTROL(QMGR) PORT(1420) REPLACE
START LISTENER(1420)
DEFINE NAMELIST(ALL) NAMES(SALES, FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(SALES) REPLACE
DEFINE QALIAS(A.SALES) CLUSNL(ALL) TARGET(SALES) TARGTYPE(QUEUE) DEFBIND(NOTFIXED)
REPLACE
DEFINE QREMOTE(FINCLT) RNAME(' ') RQMNAME(FINCLT) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(SALESRV) RNAME(' ') RQMNAME(SALESRV) CLUSNL(ALL) REPLACE
```

2. 通过运行样本请求程序来测试配置。

a) 在 SALESRV 队列管理器上启动触发器监视器程序

在 Windows 上, 打开命令窗口并运行命令 `runmqtrm -m SALESRV`

b) 运行样本请求程序，然后发送请求。

在 Windows 上，打开命令窗口并运行命令 `amqsreq A.SALES FINCLT`

将回传请求消息，并在 15 秒后完成样本程序。

3. 创建定义以隔离 Q.SALES 集群中的 SALES 队列，并在网关队列管理器上为 SALES 和 FINANCE 集群单独创建集群消息。

Run the command: `runmqsc QmgrName < filename` where the files are listed in the following list, and the file name almost matches the queue manager name.

用于隔离新集群中的销售队列并分隔网关集群传输队列的更改

chgsaler1.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
```

chgsaler2.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
```

chgsalesrv.txt

```
DEFINE NAMELIST (CLUSTERS) NAMES(SALES, Q.SALES)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SAVESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(Q.SALES) REPLACE
ALTER QLOCAL (SALES) CLUSTER(' ') CLUSNL(CLUSTERS)
```

chggate.txt

```
ALTER NAMELIST(ALL) NAMES(SALES, FINANCE, Q.SALES)
ALTER QMGR DEFCLXQ(CHANNEL)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.GATE) CHLTYPE(CLUSRCVR) CONNAME('localhost(1420)')
CLUSTER(Q.SALES) REPLACE
DEFINE QLOCAL (XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
DEFINE QLOCAL (XMITQ.SALES) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
DEFINE QLOCAL (XMITQ.FINANCE) USAGE(XMITQ) CLCHNAME(FINANCE.*) REPLACE
```

4. 从 SALES 集群中除去 SALES 队列。

在 [第 249 页的图 58](#) 中运行 **MQSC** 命令:

```
ALTER QLOCAL(SALES) CLUSTER('Q.SALES') CLUSNL(' ')
```

图 58: 从销售集群中除去队列管理器 SALESRV 上的销售队列

5. 将通道切换到新的传输队列。

需要停止并启动 GATE 队列管理器正在使用的所有通道。要使用最少数量的命令执行此操作，请停止并启动队列管理器

```
endmqm -i GATE
strmqm GATE
```

下一步做什么

1. 重新运行样本请求程序以验证新配置是否有效; 请参阅步骤 [第 248 页的『2』](#)

2. 监视流经 GATE 队列管理器上所有集群传输队列的消息:

- a. 更改要开启队列监视的每个集群传输队列的定义。

```
ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.  
name) STATQ(ON)
```

- b. 检查队列管理器统计信息监视是否为 OFF，以最小化输出，并将监视时间间隔设置为较低的值以方便地执行多个测试。

```
ALTER QMGR STATINT(60) STATCHL(OFF) STATQ(OFF) STATMQI(OFF) STATACLS(OFF)
```

- c. 重新启动 GATE 队列管理器。

- d. 运行样本请求程序几次以验证是否有相同数量的消息流经 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV 和 SYSTEM.CLUSTER.TRANSMIT.QUEUE。请求流经 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV，应答流经 SYSTEM.CLUSTER.TRANSMIT.QUEUE。

```
amqsmon -m GATE -t statistics
```

- e. 几个时间间隔内的结果如下所示:

```
C:\Documents and Settings\Admin>amqsmon -m GATE -t statistics  
MonitoringType: QueueStatistics  
QueueManager: 'GATE'  
IntervalStartDate: '2012-02-27'  
IntervalStartTime: '14.59.20'  
IntervalEndDate: '2012-02-27'  
IntervalEndTime: '15.00.20'  
CommandLevel: 700  
ObjectCount: 2  
QueueStatistics: 0  
  QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'  
  CreateDate: '2012-02-24'  
  CreateTime: '15.58.15'  
  ...  
  Put1Count: [0, 0]  
  Put1FailCount: 0  
  PutBytes: [435, 0]  
  GetCount: [1, 0]  
  GetBytes: [435, 0]  
  ...  
QueueStatistics: 1  
  QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'  
  CreateDate: '2012-02-24'  
  CreateTime: '16.37.43'  
  ...  
  PutCount: [1, 0]  
  PutFailCount: 0  
  Put1Count: [0, 0]  
  Put1FailCount: 0  
  PutBytes: [435, 0]  
  GetCount: [1, 0]  
  GetBytes: [435, 0]  
  ...  
MonitoringType: QueueStatistics  
QueueManager: 'GATE'  
IntervalStartDate: '2012-02-27'
```



```

IntervalStartTime: '15.00.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.01.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
  QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
  CreateDate: '2012-02-24'
  CreateTime: '15.58.15'
  ...
  PutCount: [2, 0]
  PutFailCount: 0
  Put1Count: [0, 0]
  Put1FailCount: 0
  PutBytes: [863, 0]
  GetCount: [2, 0]
  GetBytes: [863, 0]
  ...
QueueStatistics: 1
  QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
  CreateDate: '2012-02-24'
  CreateTime: '16.37.43'
  ...
  PutCount: [2, 0]
  PutFailCount: 0
  Put1Count: [0, 0]
  Put1FailCount: 0
  PutBytes: [863, 0]
  GetCount: [2, 0]
  GetBytes: [863, 0]
  ...
2 Records Processed.

```

在第一个时间间隔内发送了一个请求和应答消息，在第二个时间间隔内发送了两个请求和应答消息。您可以推断请求消息已放置在 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV 上，应答消息已放置在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上。

集群: 切换集群传输队列

规划如何使对现有生产队列管理器的集群传输队列所作的更改生效。

开始之前

如果减少切换过程必须传输到新传输队列的消息数，那么切换将更快地完成。请阅读第 144 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](#)』，以了解在继续任何操作之前尝试清空传输队列的原因。

关于此任务

您可以选择两种方法使对集群传输队列的更改生效。

1. 让队列管理器自动进行更改。这是缺省值。当集群发送方通道下次启动时，队列管理器会切换具有暂挂传输队列更改的集群发送方通道。
2. 手动进行更改。您可以在集群发送方通道停止时对其进行更改。在集群发送方通道启动之前，可以将其从一个集群传输队列切换到另一个集群传输队列。

在决定选择两个选项中的哪一个时，您会考虑哪些因素，以及如何管理交换机？

过程

- 选项 1: 让队列管理器自动进行更改; 请参阅 [第 252 页的『将活动集群发送方通道切换到另一组集群传输队列』](#)。

如果希望队列管理器为您进行切换, 请选择此选项。

描述此选项的另一种方法是, 队列管理器在不强制通道停止的情况下切换集群发送方通道。您可以选择强制通道停止, 然后启动通道, 以使交换机更快发生。交换机在通道启动时启动, 并在通道运行时运行, 这与选项 2 不同。在选项 2 中, 交换机在通道停止时发生。

如果通过让交换机自动发生来选择此选项, 那么切换过程将在集群发送方通道启动时启动。如果通道未停止, 那么它将在变为不活动状态后启动 (如果有要处理的消息)。如果通道已停止, 请使用 `START CHANNEL` 命令将其启动。

一旦在通道所服务的传输队列上没有针对集群发送方通道的剩余消息, 交换机进程就会完成。一旦出现这种情况, 集群发送方通道的新到达消息将直接存储在新的传输队列上。在此之前, 消息存储在旧的传输队列上, 切换过程将消息从旧的传输队列传输到新的传输队列。集群发送方通道在整个切换过程中转发来自新集群传输队列的消息。

交换机进程何时完成取决于系统的状态。如果要在维护窗口中进行更改, 请事先评估切换过程是否会及时完成。它是否会及时完成取决于等待从旧传输队列传输的消息数是否达到零。

第一种方法的优点是它是自动的。一个缺点是, 如果进行配置更改的时间仅限于维护窗口, 那么您必须确信可以控制系统在维护窗口内完成切换过程。如果无法确定, 那么选项 2 可能是更好的选择。

- 选项 2: 手动进行更改; 请参阅 [第 253 页的『将已停止的集群发送方通道切换到另一个集群传输队列』](#)。

如果要手动控制整个切换过程, 或者要切换已停止或不活动的通道, 请选择此选项。如果要切换几个集群发送方通道, 并且要在维护窗口期间执行切换, 那么这是一个不错的选择。

此选项的替代描述是表示在集群发送方通道停止时切换集群发送方通道。

如果选择此选项, 那么您可以完全控制交换机的发生时间。

您可以确定在固定时间内, 在维护窗口内完成切换过程。切换所花费的时间取决于必须从一个传输队列传输到另一个传输队列的消息数。如果消息不断到达, 那么进程可能需要一段时间才能传输所有消息。

您可以选择在不从旧传输队列传输消息的情况下切换通道。交换机为“即时”。

当您重新启动集群发送方通道时, 它将开始处理新分配给它的传输队列上的消息。

第二种方法的优点是您可以控制切换过程。缺点是您必须识别要切换的集群发送方通道, 运行必要的命令, 并解决可能阻止集群发送方通道停止的任何不确定通道。

相关概念

[如何选择要使用的集群传输队列类型](#)

[如何在不同的集群传输队列配置选项之间进行选择。](#)

相关任务

[集群: 多个集群传输队列的示例配置](#)

在此任务中, 您将应用步骤以将多个集群传输队列规划到三个重叠的集群。这些要求是将消息流与所有其他消息流分离到一个集群队列, 并将不同集群的消息存储在不同的集群传输队列上。

[第 252 页的『将活动集群发送方通道切换到另一组集群传输队列』](#)

此任务为您提供了三个用于切换活动集群发送方通道的选项。一个选项是让队列管理器自动进行切换, 这不会影响正在运行的应用程序。其他选项包括手动停止和启动通道, 或者重新启动队列管理器。

[第 253 页的『将已停止的集群发送方通道切换到另一个集群传输队列』](#)

相关信息

[第 144 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』](#)

[将活动集群发送方通道切换到另一组集群传输队列](#)

此任务为您提供了三个用于切换活动集群发送方通道的选项。一个选项是让队列管理器自动进行切换, 这不会影响正在运行的应用程序。其他选项包括手动停止和启动通道, 或者重新启动队列管理器。

开始之前

更改集群传输队列配置。您可以更改 **DEFCLXQ** 队列管理器属性，或者添加或修改传输队列的 **CLCHNAME** 属性。

如果减少切换过程必须传输到新传输队列的消息数，那么切换将更快地完成。请阅读第 144 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』，以了解在继续任何操作之前尝试清空传输队列的原因。

关于此任务

使用任务中的步骤作为制定您自己的计划以进行集群传输队列配置更改的基础。

过程

1. 可选：记录当前通道状态

记录正在为集群传输队列提供服务的当前通道和已保存通道的状态。以下命令显示与系统集群传输队列关联的状态。添加您自己的命令以显示与已定义的集群传输队列相关联的状态。使用约定 (例如 `XMITQ.ChannelName`) 来命名您定义的集群传输队列，以便于显示这些传输队列的通道状态。

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

2. 交换机传输队列。

- 不执行任何操作。当集群发送方通道在停止或不活动后重新启动时，队列管理器会切换这些集群发送方通道。

如果您没有更改队列管理器配置的规则或顾虑，请选择此选项。正在运行的应用程序不受更改影响。

- 重新启动队列管理器。将根据需要停止并自动重新启动所有集群发送方通道。

选择此选项以立即启动所有更改。在队列管理器关闭并重新启动时，正在运行的应用程序将被队列管理器中断。

- 停止各个集群发送方通道并将其重新启动。

选择此选项可立即更改几个通道。正在运行的应用程序在停止消息通道和再次启动消息通道之间的消息传输中迁到短暂延迟。集群发送方通道保持运行，但在您停止该通道期间除外。在切换过程中，将消息传递到旧的传输队列，通过切换过程传输到新的传输队列，并通过集群发送方通道从新的传输队列转发。

3. 可选：在通道切换时监视通道

显示交换机期间的通道状态和传输队列深度。以下示例显示系统集群传输队列的状态。

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

- 4. 可选：监视消息“AMQ7341 通道 `ChannelName` 的传输队列已从队列 `QueueName` 切换到写入队列管理器错误日志的 `QueueName`”。

将已停止的集群发送方通道切换到另一个集群传输队列

开始之前

您可以进行一些配置更改，现在希望使这些更改生效，而不启动受影响的集群发送方通道。或者，作为任务中的其中一个步骤进行所需的配置更改。

如果减少切换过程必须传输到新传输队列的消息数，那么切换将更快地完成。请阅读第 144 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』，以了解在继续任何操作之前尝试清空传输队列的原因。

关于此任务

此任务切换由已停止或不活动的集群发送方通道提供服务的传输队列。您可以执行此任务，因为集群发送方通道已停止，并且您希望立即切换其传输队列。例如，由于某种原因，集群发送方通道未启动，或者存在一些其他配置问题。要解决此问题，您决定创建集群发送方通道，并将旧集群发送方通道的传输队列与您定义的新集群发送方通道相关联。

更有可能的情况是，您希望控制何时执行集群传输队列的重新配置。要完全控制重新配置，请停止通道，更改配置，然后切换传输队列。

过程

1. 停止要切换的通道

- a) 停止要切换的任何正在运行或不活动的通道。停止不活动的集群发送方通道会阻止它在您进行配置更改时启动。

```
STOP CHANNEL(ChannelName) MODE(QUIESCSE) STATUS(STOPPED)
```

2. 可选：进行配置更改。

例如，请参阅第 244 页的『[集群: 多个集群传输队列的示例配置](#)』。

3. 将集群发送方通道切换到新的集群传输队列。

```
runswchl -m QmgrName -c ChannelName
```

runswchl 命令将旧传输队列上的任何消息传输到新的传输队列。当此通道的旧传输队列上的消息数达到零时，交换机将完成。该命令是同步的。此命令在切换过程中将进度消息写入窗口。

在传输阶段，将以集群发送方通道为目标的现有消息和新消息传输到新的传输队列。

由于集群发送方通道已停止，因此消息会在新的传输队列上构建。将已停止的集群发送方通道与第 252 页的『[将活动集群发送方通道切换到另一组集群传输队列](#)』中的步骤第 253 页的『2』进行对比。在该步骤中，集群发送方通道正在运行，因此消息不一定会在新的传输队列上构建。

4. 可选：在通道切换时监视通道

在另一个命令窗口中，显示切换期间的传输队列深度。以下示例显示系统集群传输队列的状态。

```
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

5. 可选：监视消息“AMQ7341 通道 *ChannelName* 的传输队列已从队列 *QueueName* 切换到写入队列管理器错误日志的 *QueueName*”。

6. 重新启动已停止的集群发送方通道。

通道不会自动启动，因为您已停止这些通道，并将其置于 STOPPED 状态。

```
START CHANNEL(ChannelName)
```

相关参考

[伦斯湖](#)

[解析通道](#)

[停止通道](#)

集群: 迁移和修改最佳实践

本主题提供有关规划和管理 IBM WebSphere MQ 集群的指导。此信息是基于测试和客户反馈的指南。

1. 第 255 页的『[在集群中移动对象](#)』(在集群内移动对象的最佳实践，而不安装任何修订包或新版本的 IBM WebSphere MQ)。
2. 第 255 页的『[升级和维护安装](#)』(在应用维护或升级以及测试新体系结构时保持工作集群体系结构正常运行的最佳实践)。

在集群中移动对象

应用程序及其队列

当您必须将托管在一个队列管理器上的队列实例移动到另一个队列管理器上时，可以使用工作负载均衡参数来确保平稳过渡。

创建要在其中进行新托管的队列实例，但使用集群工作负载均衡设置继续向原始实例发送消息，直到应用程序准备好切换为止。这是通过以下步骤实现的：

1. 将现有队列的 **CLWLRANK** 属性设置为高值，例如 5。
2. 创建队列的新实例，并将其 **CLWLRANK** 属性设置为零。
3. 完成新系统的任何进一步配置，例如，针对队列的新实例部署和启动使用应用程序。
4. 将新队列实例的 **CLWLRANK** 属性设置为高于原始实例，例如 9。
5. 允许原始队列实例处理系统中的任何排队消息，然后删除该队列。

移动整个队列管理器

如果队列管理器位于同一主机上，但 IP 地址正在更改，那么此过程如下所示：

- 当正确使用 DNS 时，可以帮助简化该过程。有关通过设置 连接名称 (CONNAME) 通道属性使用 DNS 的信息，请参阅 ALTER CHANNEL。
- 如果移动完整存储库，请确保在进行更改之前至少有一个其他完整存储库正在平稳运行 (例如，通道状态没有问题)。
- 使用 SUSPEND QMGR 命令暂挂队列管理器以避免流量堆积。
- 修改计算机的 IP 地址。如果 CLUSRCVR 通道定义使用 CONNAME 字段中的 IP 地址，请修改此 IP 地址项。可能需要清空 DNS 高速缓存以确保到处都有可用的更新。
- 当队列管理器重新连接到完整存储库时，通道自动定义会自动解析自身。
- 如果队列管理器托管了完整存储库，并且 IP 地址发生更改，请务必确保尽快切换分区以将任何手动定义的 CLUSSDR 通道指向新位置。在执行此切换之前，这些队列管理器可能只能与剩余的 (未更改的) 完整存储库联系，并且可能会看到有关不正确通道定义警告消息。
- 使用 RESUME QMGR 命令恢复队列管理器。

如果必须将队列管理器移至新主机，那么可以复制队列管理器数据并从备份复原。但是，除非没有其他选项，否则建议不要执行此过程；最好在新机器上创建队列管理器，并如上一节中所述复制队列和应用程序。这种情况提供了一种平滑的翻转/回滚机制。

如果您决心使用备份来移动完整的队列管理器，请遵循以下最佳实践：

- 将整个进程视为来自备份的队列管理器复原，应用您通常用于操作系统环境的系统恢复的任何进程。
- 迁移后使用 **REFRESH CLUSTER** 命令来废弃所有本地持有的集群信息 (包括任何不确定的自动定义通道)，并强制对其进行重建。

注：对于大型集群，当集群正在运行中时，使用 **REFRESH CLUSTER** 命令可能会破坏该集群，并且将在 27 天的时间间隔之后，集群对象才会再次自动向所有相关队列管理器发送状态更新。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。

创建队列管理器并从集群中的现有队列管理器复制设置 (如本主题中所述) 时，切勿将两个不同的队列管理器视为实际相同。特别是，不要为新队列管理器提供相同的队列管理器名称和 IP 地址。尝试 "删除" 替换队列管理器是导致 IBM WebSphere MQ 集群中出现问题的常见原因。高速缓存期望接收包括 **QMID** 属性在内的更新，并且状态可能已损坏。

如果意外创建了两个同名的不同队列管理器，那么建议使用 RESET CLUSTER QMID 命令从集群中弹出不正确的条目。

升级和维护安装

避免 "大爆炸场景" (例如，停止所有集群和队列管理器活动，将所有升级和维护应用于所有队列管理器，然后同时启动所有内容)：集群设计为仍与多个版本的队列管理器共存工作，因此建议采用精心规划的分阶段维护方法。

具有备份计划:

- 在 z/OS 上, 是否应用了向后迁移 PTF?
- 您是否进行了备份?
- 避免立即使用新的集群功能: 请等待您确定所有队列管理器都已升级到新级别, 并确定您不会将它们中的任何一个回滚。在某些队列管理器仍处于较早级别的集群中使用新的集群功能可能会导致未定义的行为。例如, 在从 IBM WebSphere MQ Version 6.0 移至 IBM WebSphere MQ Version 7.1 时, 如果队列管理器定义了集群主题, 那么 IBM WebSphere MQ Version 6.0 队列管理器将无法理解该定义或无法发布此主题。

首先迁移完整存储库。虽然他们可以转发他们不了解的信息, 但他们不能坚持下去, 因此除非绝对必要, 否则并不是建议的方法。有关更多信息, 请参阅 [队列管理器集群迁移](#)。

相关概念

第 256 页的『[集群: 使用 REFRESH CLUSTER 最佳实践](#)』

使用 **REFRESH CLUSTER** 命令可废弃有关集群的所有本地保存的信息, 并从集群中的完整存储库重建该信息。您不应该使用此命令, 除非在特殊情况下。如果确实需要使用它, 那么有关如何使用它有特殊注意事项。此信息是基于测试和客户反馈的指南。

集群: 使用 REFRESH CLUSTER 最佳实践

使用 **REFRESH CLUSTER** 命令可废弃有关集群的所有本地保存的信息, 并从集群中的完整存储库重建该信息。您不应该使用此命令, 除非在特殊情况下。如果确实需要使用它, 那么有关如何使用它有特殊注意事项。此信息是基于测试和客户反馈的指南。

仅在确实需要时运行 REFRESH CLUSTER

IBM WebSphere MQ 集群技术可确保对集群配置的任何更改 (例如对集群队列的更改) 自动为需要了解信息的集群成员所知。没有必要采取进一步的行政步骤来实现这种信息传播。

如果此类信息未到达需要该信息的集群中的队列管理器, 例如, 当应用程序首次尝试将其打开时, 集群中的另一个队列管理器不知道集群队列, 那么意味着集群基础结构中存在问题。例如, 可能无法在队列管理器与完整存储库队列管理器之间启动通道。因此, 必须对观察到的任何不一致情况进行调查。如果可能, 请在不使用 **REFRESH CLUSTER** 命令的情况下解决此情况。

在本产品文档中其他地方记录的极少数情况下, 或者在 IBM 支持人员请求时, 您可以使用 **REFRESH CLUSTER** 命令来废弃有关集群的所有本地保存的信息, 并从集群中的完整存储库重建该信息。

在大型集群中刷新可能会影响集群的性能和可用性

使用 **REFRESH CLUSTER** 命令可能会对正在进行的集群造成干扰, 例如, 在处理队列管理器集群资源的重新传播时, 为完整存储库创建突然增加的工作。如果在大型集群 (即, 数百个队列管理器) 中刷新, 那么应避免在日常工作中使用该命令 (如果可能), 并使用替代方法来更正特定不一致情况。例如, 如果未在集群中正确传播集群队列, 那么更新集群队列定义的初始调查技术 (例如, 更改其描述) 将在集群中重新传播队列配置。此过程可帮助确定问题并可能解决临时不一致问题。

如果无法使用替代方法, 并且必须在大型集群中运行 **REFRESH CLUSTER**, 那么应在非高峰时间或维护时段执行此操作, 以避免影响用户工作负载。您还应避免在单个批处理中刷新大型集群, 而是按 [第 256 页的『避免集群对象发送自动更新时出现性能和可用性问题』](#) 中所述错开活动。

避免集群对象发送自动更新时出现性能和可用性问题

在队列管理器上定义新集群对象之后, 将从定义开始每 27 天生成此对象的更新, 并将此更新发送到集群中的每个完整存储库, 然后再发送到任何其他相关队列管理器。向队列管理器发出 **REFRESH CLUSTER** 命令时, 将在指定集群中本地定义的所有对象上重置此自动更新的时钟。

如果在单个批处理中刷新大型集群 (即, 数百个队列管理器), 或者在其他情况下 (例如, 从配置备份重新创建系统), 那么在 27 天后, 所有这些队列管理器都将同时向完整存储库重新发布其所有对象定义。这可能再次导致系统运行速度显着降低, 甚至变为不可用, 直到所有更新都完成为止。因此, 当您必须在大型集群中刷新或重新创建多个队列管理器时, 应该将活动错开几个小时或几天, 以便后续自动更新不会定期影响系统性能。

系统集群历史记录队列

执行 **REFRESH CLUSTER** 时，队列管理器将在刷新之前生成集群状态的快照，并将其存储在 `SYSTEM.CLUSTER.HISTORY.QUEUE (SCHQ)` (如果在队列管理器上定义) 上。此快照仅用于 IBM 服务目的，以防以后发生系统问题。缺省情况下，在启动时在分布式队列管理器上定义 `SCHQ`。对于 z/OS 迁移，必须手动定义 `SCHQ`。`SCHQ` 上的消息将在三个月后到期。

相关概念

[运行 REFRESH CLUSTER 时发现的应用程序问题](#)

[针对发布/预订集群的 REFRESH CLUSTER 注意事项](#)

相关参考

[MQSC 命令参考: REFRESH CLUSTER](#)

集群: 可用性, 多实例和灾难恢复

本主题提供有关规划和管理 IBM WebSphere MQ 集群的指导。此信息是基于测试和客户反馈的指南。

IBM WebSphere MQ 集群本身不是高可用性解决方案，但在某些情况下，可以使用它来提高使用 IBM WebSphere MQ 的服务的可用性，例如，在不同的队列管理器上具有多个队列实例。本部分提供了有关确保 IBM WebSphere MQ 基础结构尽可能高可用性的指导信息，以便可以在此类体系结构中使用该基础结构。

集群资源的可用性

通常建议维护两个完整存储库的原因是，丢失一个存储库对于集群的顺利运行并不重要。即使两者都变得不可用，对于部分存储库持有的现有知识也有 60 天的宽限期，尽管在此事件中没有新的或先前未访问的资源 (例如，队列) 可用。

使用集群来提高应用程序可用性

通过使用队列和应用程序的多个实例，集群可以帮助设计高可用性应用程序 (例如请求/响应类型服务器应用程序)。如果需要，优先级属性可以优先使用 "活动" 应用程序，除非例如队列管理器或通道变为不可用。这对于在发生问题时快速切换以继续处理新消息非常强大。

但是，传递到集群中特定队列管理器的消息仅保留在该队列实例上，并且在恢复该队列管理器之前不可用于处理。因此，为了实现真正的数据高可用性，您可能需要考虑其他技术，例如多实例队列管理器。

多实例队列管理器

软件高可用性 (多实例) 是保持现有消息可用的最佳内置产品。请参阅第 264 页的『将 WebSphere MQ 与高可用性配置配合使用』，第 288 页的『创建多实例队列管理器』和以下部分以获取更多信息。只要集群中的所有队列管理器至少正在运行 IBM WebSphere MQ Version 7.0.1，就可以使用此技术使集群中的任何队列管理器具有高可用性。如果集群中的任何队列管理器处于先前级别，那么当它们故障转移到辅助 IP 时，可能会丢失与多实例队列管理器的连接。

正如本主题中先前所讨论的那样，只要配置了两个完整的存储库，它们几乎按其性质高度可用。如果需要，可以将 IBM WebSphere MQ 软件高可用性/多实例队列管理器用于完整存储库。没有强烈的理由使用这些方法，事实上，对于临时中断，这些方法可能会在故障转移期间造成额外的性能成本。建议不要使用软件 HA 而不是运行两个完整存储库，因为例如，在发生单个通道中断时，它不一定会故障转移，但可能会导致部分存储库无法查询集群资源。

灾难恢复

灾难恢复 (例如，从存储队列管理器数据的磁盘损坏时进行恢复) 很难很好地执行; IBM WebSphere MQ 可以提供帮助，但它无法自动执行此操作。IBM WebSphere MQ 中唯一的 "true" 灾难恢复选项 (不包括任何操作系统或其他底层复制技术) 是从备份复原。在以下情况下，需要考虑一些特定于集群的点:

- 测试灾难恢复方案时请小心。例如，如果测试备份队列管理器的操作，那么在使它们在同一网络中联机时要小心，因为有可能意外地加入实时集群，并通过托管与实时集群队列管理器中相同的指定队列来开始 "窃取" 消息。
- 灾难恢复测试不得干扰正在运行的实时集群。避免干扰的方法包括:
 - 在防火墙级别完成网络分离或分离。
 - 在发生实际灾难恢复方案之前，或者除非发生实际灾难恢复方案，否则不会向灾难恢复系统发出实时 SSL 证书。

- 在集群中复原队列管理器的备份时，备份可能与集群的其余部分不同步。**REFRESH CLUSTER** 命令可以解析更新并与集群同步，但 **REFRESH CLUSTER** 命令必须用作最后的手段。请参阅第 256 页的『[集群：使用 REFRESH CLUSTER 最佳实践](#)』。查看任何内部流程文档和 IBM WebSphere MQ 文档，以了解在使用该命令之前是否错过了简单的步骤。
- 对于任何恢复，应用程序都必须处理重放和数据丢失。必须决定是将队列清除到已知状态，还是在其他地方有足够的信息来管理重放。

集群: 监视

本主题提供有关规划和管理 IBM WebSphere MQ 集群的指导。此信息是基于测试和客户反馈的指南。

监视集群中的应用程序消息

通常，离开队列管理器的所有集群消息都通过 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 传递，而不考虑使用哪个集群发送方通道来传输消息。每个通道都在与所有其他集群发送方通道并行地排出针对该通道的消息。此队列上不断增加的消息可能指示一个或多个通道存在问题，必须对其进行调查：

- 必须针对集群设计相应地监视队列的深度。
- 以下命令将返回具有多个正在传输队列上等待的消息的所有通道：

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

对于单个队列上的所有集群消息，在开始填充时并不总是容易看到哪个通道存在问题。使用此命令是查看哪个通道负责的简单方法。

您可以将集群队列管理器配置为具有多个传输队列。如果将队列管理器属性 `DEFCLXQ` 更改为 `CHANNEL`，那么每个集群发送方通道都与不同的集群传输队列相关联。或者，您可以手动配置单独的传输队列。要显示与集群发送方通道关联的所有集群传输队列，请运行以下命令：

```
DISPLAY CLUSQMGR (qmgrName) XMITQ
```

定义集群传输队列，以便它们遵循左侧具有队列名称的固定主干的模式。然后，可以使用通用队列名称来查询 **DISPLAY CLUSMGR** 命令返回的所有集群传输队列的深度：

```
DISPLAY QUEUE (qname*) CURDEPTH
```

监视集群中的控制消息

`SYSTEM.CLUSTER.COMMAND.QUEUE` 队列用于处理队列管理器的所有集群控制消息，这些消息由本地队列管理器生成或从集群中的其他队列管理器发送到此队列管理器。当队列管理器正确维护其集群状态时，此队列趋向于零。但是，在某些情况下，此队列上的消息深度可能会临时增大：

- 队列中有大量消息表示流失率处于集群状态。
- 进行重大更改时，允许队列在这些更改之间进行结算。例如，移动存储库时，允许队列在移动第二个存储库之前达到零。

当此队列上存在积压的消息时，不会处理对集群状态或与集群相关的命令的更新。如果长时间未从此队列中除去消息，那么需要进一步调查，最初是通过检查队列管理器错误日志，这可能说明导致此情况的进程。

`SYSTEM.CLUSTER.REPOSITORY.QUEUE` 将集群存储库高速缓存信息作为大量消息保存。通常，消息会始终存在于此队列中，而对于更大的集群则会存在更多消息。因此，此队列上的消息深度不是值得关注的问题。

监视日志

由于信息高速缓存和集群的分布式性质，在最初发生问题后的许多天（甚至几个月）内，集群中发生的问题可能不会向应用程序显示外部症状。但是，原始问题通常在 IBM WebSphere MQ 错误日志中报告。因此，主动监视这些日志以获取与集群相关的任何消息至关重要。必须阅读和理解这些消息，并在必要时采取任何行动。

例如: 中断与集群中的队列管理器的通信可能会导致了解由于集群通过重新发布信息定期重新验证集群资源的方式而正在删除的某些集群资源。可能会发生此类事件的警告由消息 [AMQ9465](#) 报告。此消息指示需要调查问题。

负载均衡的特殊注意事项

当集群在队列的两个或多个实例之间进行负载均衡时, 使用应用程序必须处理每个实例上的消息。如果一个或多个使用应用程序的应用程序终止或停止处理消息, 那么集群可能会继续向这些队列实例发送消息。在这种情况下, 直到应用程序再次正常工作时, 才会处理这些消息。因此, 对应用程序的监视是解决方案的重要组成部分, 在这种情况下, 必须采取行动来重新路由消息。可在以下样本中找到用于自动执行此类监视的机制的示例: [集群队列监视样本程序 \(AMQSCLM\)](#)。

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, 并在服务器或存储器发生故障后恢复消息, 使应用程序具有高可用性。

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, 或在队列管理器发生故障后切换到多实例队列管理器的新活动实例, 提高客户机应用程序可用性。WebSphere MQ classes for Java 不支持客户机自动重新连接。

在 Windows 上, UNIX, Linux 和 IBM i 平台将服务器应用程序部署到多实例队列管理器, 该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; 如果运行活动实例的服务器失败, 那么执行会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。

您可以将 WebSphere MQ 配置为特定于平台的集群解决方案 (例如 微软公司 Cluster Server, 或 PowerHA for AIX (先前为 AIX 上的 HACMP) 和其他 UNIX and Linux 集群解决方案) 的一部分。

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。

消息传递系统确保将输入到系统中的消息传递到其目标。WebSphere 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。

WebSphere MQ 通过维护处理消息接收, 传输和传递的队列管理器活动的恢复日志, 确保消息不会丢失。它将 这些日志用于三种类型的恢复:

1. 以计划方式停止 WebSphere MQ 时的重新启动恢复。
2. 故障恢复, 当故障停止时 WebSphere 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。

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。

相反, 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 260 页的表 28 列出了必须安装在客户机工作站上的 IBM WebSphere 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 260 页的表 29 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 JMS 客户机，请在连接工厂配置中设置重新连接选项。使用 WebSphere MQ 资源适配器或集成在 Java EE 环境中的 JMS 客户机时，自动客户机重新连接可能不可用。某些受管环境中存在一些限制，有关更多信息，请参阅在 Java SE 和 Java EE 环境中使用自动客户机重新连接。

注：WebSphere MQ Java 类不支持自动客户机重新连接。

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 (在 Windows 上为非受管 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
	WebSphere MQ Java 类	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。

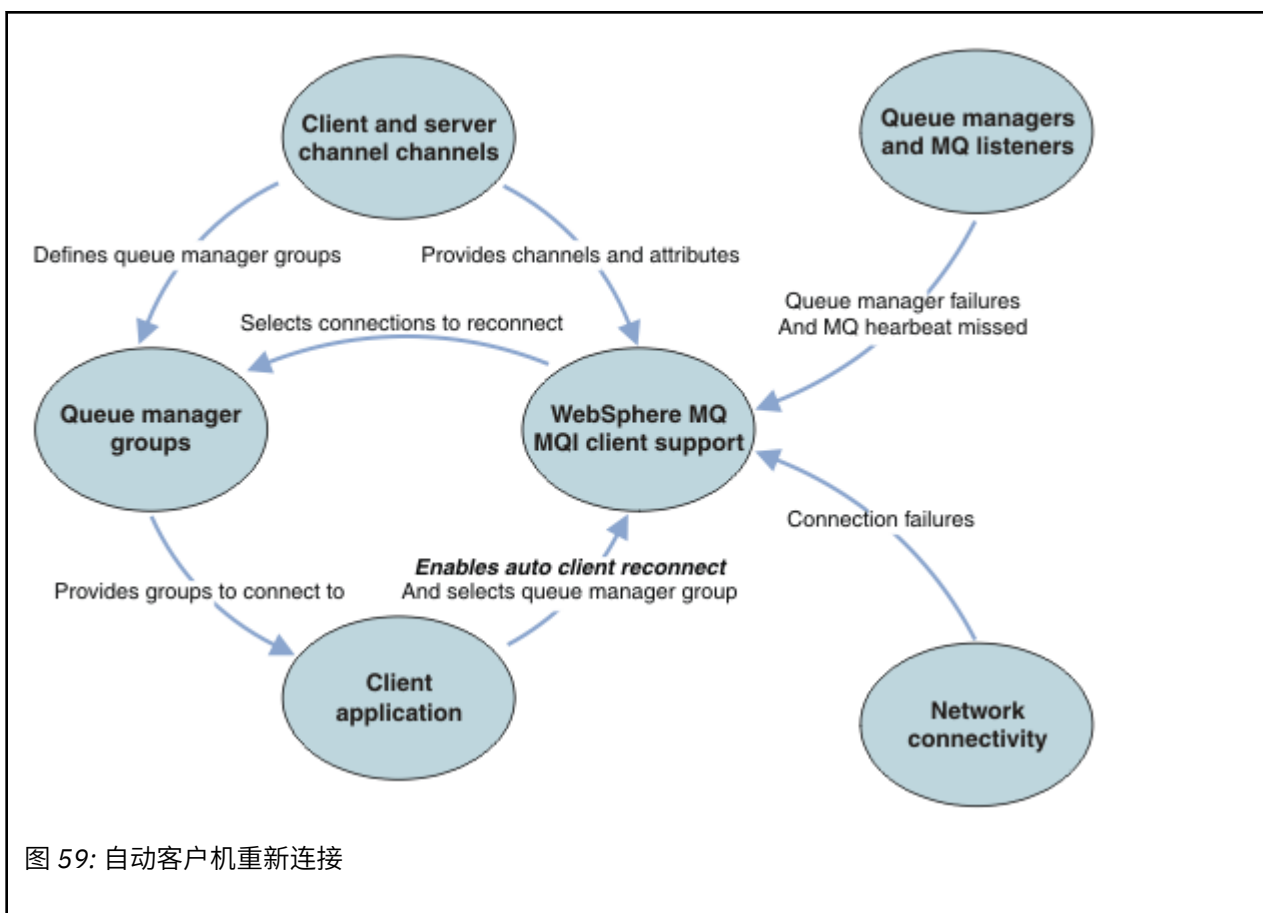
自动重新连接具有以下配置需求：

组件	需求	不符合要求的影响
WebSphere MQ MQI 客户机安装	请参阅 第 260 页的表 28	MQRC_OPTIONS_ERROR
WebSphere MQ Server 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

表 29: 自动重新连接配置需求 (继续)

组件	需求	不符合要求的影响
MQI	值为下列其中一项: <ul style="list-style-type: none"> MQCONN, 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 261 页的图 59 显示了客户机重新连接中涉及的组件之间的主要交互。



客户机应用程序

客户机应用程序是 IBM WebSphere MQ MQI 客户机。

- 缺省情况下, 不会自动重新连接客户机。通过设置 MQCONN 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。

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 327 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

如果设置了重新连接选项，那么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。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。

多实例队列管理器

通过使用自动客户机重新连接，在多实例队列管理器激活其备用实例后，简化重新启动 WebSphere MQ MQI 客户机应用程序的过程。

多实例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。

通常，WebSphere MQ MQI 客户机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 WebSphere MQ MQI 客户机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 WebSphere MQ MQI 客户机重新连接到其他队列管理器，那么该客户机将失败。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。

队列管理器组

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具有相同 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 *name* 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](#)。

队列共享组

客户机与 z/OS 队列共享组的自动重新连接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。

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。

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 WebSphere 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

只有网络传递到 WebSphere MQ MQI 客户机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 WebSphere MQ 不可见。
- 设置 HBINT 有助于处理对 WebSphere MQ 不可见的网络故障。

队列管理器和 WebSphere 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。

WebSphere MQ MQI 客户机自动重新连接支持

如果在 WebSphere MQ MQI 客户机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 调用来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`，重新连接和选项
- `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 328 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。

虽然重新连接需要的时间不超过一分钟，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 WebSphere 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。

控制台消息监视

队列管理器或通道启动程序发出的若干参考消息应视为特别重要。这些消息本身并不表示存在问题，但可能有助于跟踪，因为它们确实指示可能需要解决的潜在问题。

此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，这可能是更大问题的症状：

- PUT 的消息 (例如不受控制的循环) 的用户应用程序存在问题。
- 来自队列的 GET 消息不再起作用的用户应用程序。

将 WebSphere MQ 与高可用性配置配合使用

如果要在高可用性 (HA) 配置中操作 WebSphere MQ 队列管理器，那么可以将队列管理器设置为使用高可用性管理器 (例如 PowerHA for AIX (以前称为 HACMP) 或 Microsoft 集群服务 (MSCS)，或使用 WebSphere MQ 多实例队列管理器。

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。

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 (UNIX) 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 265 页的『[HA 集群与队列管理器集群的关系](#)』中描述了 IBM WebSphere MQ 集群与 HA 集群之间的关系。

多实例队列管理器

在两台或更多台计算机上配置的同一样列管理器的实例。通过启动多个实例，一个实例将成为活动实例，其他实例将成为备用实例。如果活动实例发生故障，那么在另一台计算机上运行的备用实例将自动接管。您可以使用多实例队列管理器根据 WebSphere MQ 来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。

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 WebSphere 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 WebSphere 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。

将 WebSphere MQ 与 UNIX and Linux 上的高可用性集群配合使用

您可以将 WebSphere MQ 与 UNIX 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 WebSphere MQ V 7.0.1 之前，提供了 SupportPac MC91 以帮助配置 HA 集群。与先前版本相比，WebSphere MQ V 7.0.1 提供了对队列管理器存储其数据的更大程度的控制。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 266 页的『HA 集群配置』，HA 集群与队列管理器集群的关系，第 266 页的『WebSphere MQ 客户机』和第 266 页的『在 HA 集群中运行的 WebSphere MQ』，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 WebSphere MQ 来帮助设置备用配置或接管配置，包括所有集群节点运行 WebSphere 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。

WebSphere MQ 客户机

与队列管理器通信的 WebSphere MQ 客户机(可能需要重新启动或接管)必须写入以容许中断的连接，并且必须反复尝试重新连接。WebSphere MQ Version 7 引入了客户机通道定义表(CCDT)处理中的功能，这些功能有助于实现连接可用性和工作负载均衡；但是，在使用故障转移系统时，这些功能并不直接相关。

允许 WebSphere MQ MQI 客户机参与两阶段事务的扩展事务客户机(ETC)必须始终连接到同一队列管理器。ETC 无法使用诸如 IP 负载均衡器之类的方法从队列管理器列表中进行选择。当您使用 HA 产品时，队列管理器会保留其身份(名称和地址)，无论它在哪个节点上运行，因此 ETC 可以与受 HA 控制的队列管理器配合使用。

在 HA 集群中运行的 WebSphere MQ

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。

WebSphere 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 WebSphere MQ 资源(例如，侦听器或触发器监视器)作为单独的资源包含在同一资源组中，或者包含在队列管理器本身的控制下。

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。

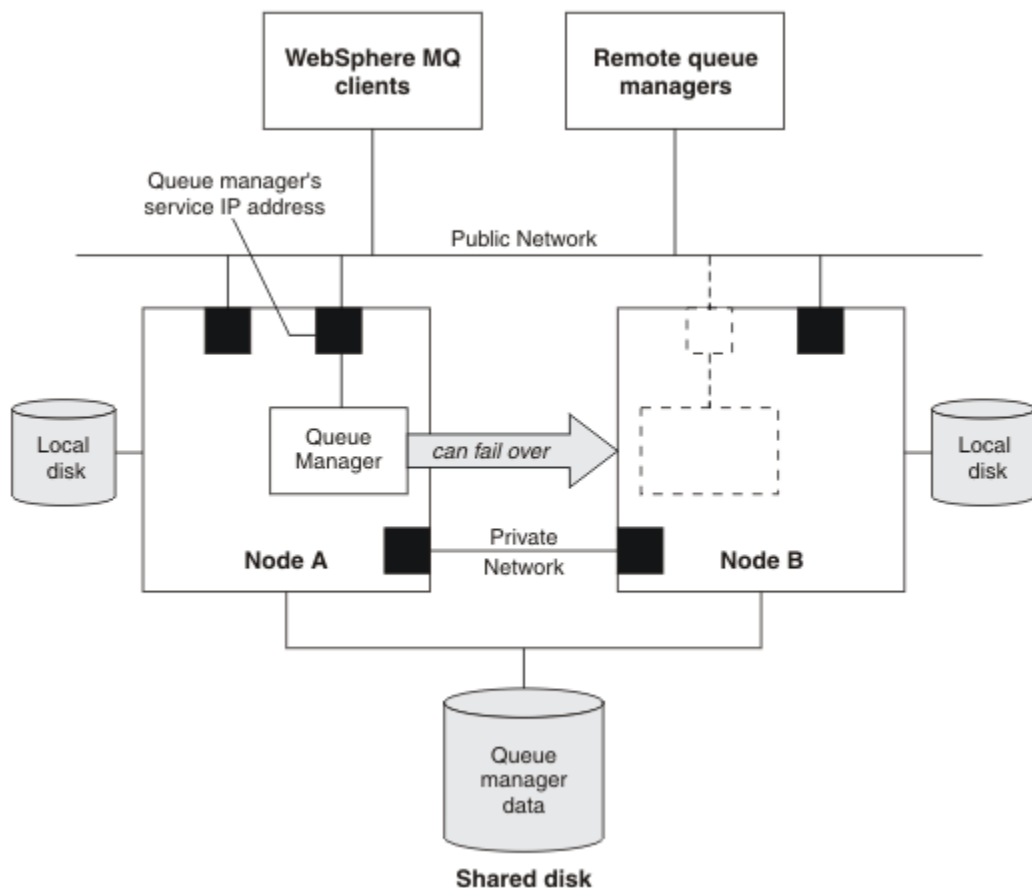


图 60: HA 集群

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。WebSphere MQ 以 mqm 用户身份运行。WebSphere 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 WebSphere 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 WebSphere MQ；不能共享产品可执行文件。

配置共享磁盘

HA 集群中的 WebSphere 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。

要配置共享磁盘，请完成以下步骤：

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。

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。

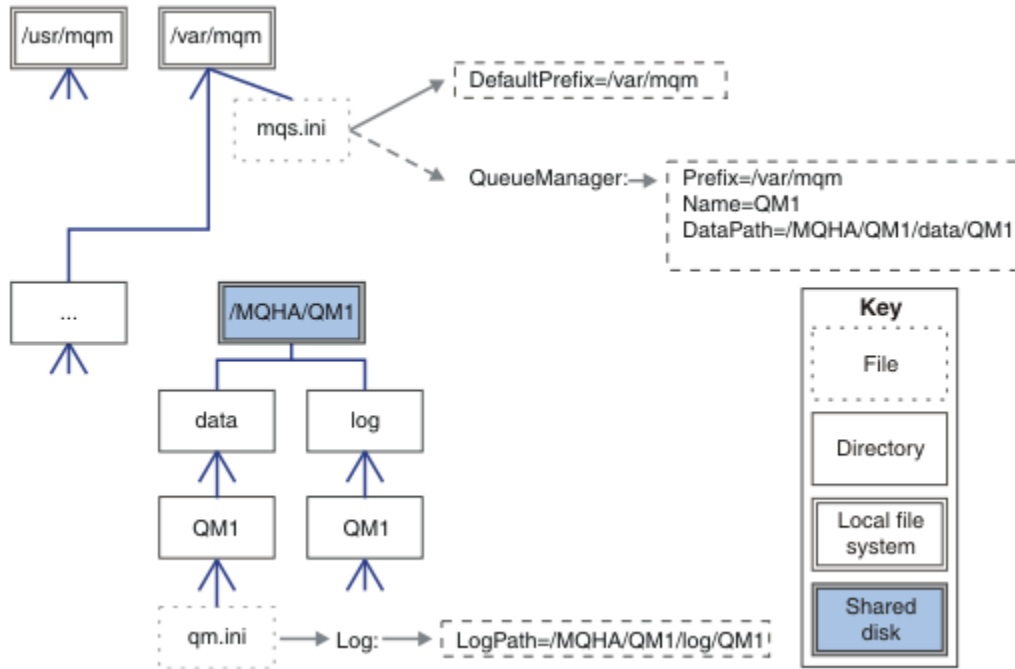


图 61: 共享名为 data 和 log 的目录

创建队列管理器以在高可用性 (HA) 集群中使用

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。

要创建要在 HA 集群中使用的队列管理器，请选择集群中要在其上创建队列管理器的其中一个节点。在此节点上，完成以下步骤：

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 **dspmqrinf** 命令可显示可在后续任务中使用的 **addmqinf** 命令，该任务记录在 [第 269 页的『将队列管理器配置信息添加到高可用性 \(HA\) 集群中的其他节点』](#) 中：

```
dspmqrinf -o command qmgrname
```

其中，qmgrname 是队列管理器的名称。

7. 显示的 **addmqinf** 命令将类似于以下示例：

```
addmqinf -sQueueManager -vName=qmgrname -vDirectory=qmgrname \
-vPrefix=/var/mqm -vDataPath=/MQHA/qmgrname/data/qmgrname
```

请仔细记录显示的命令。

8. 卸载队列管理器的文件系统。

现在，您已准备好完成第 269 页的『将队列管理器配置信息添加到高可用性 (HA) 集群中的其他节点』中描述的步骤。

将队列管理器配置信息添加到高可用性 (HA) 集群中的其他节点

必须将队列管理器配置添加到 HA 集群中的其他节点。

在完成此任务之前，必须已完成第 268 页的『创建队列管理器以在高可用性 (HA) 集群中使用』中的步骤。

要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，请在每个其他节点上完成以下步骤：

1. 安装队列管理器文件系统。
2. 通过直接编辑 `/var/mqm/mqs.ini` 或发出 `dspmqinf` 命令在步骤 6 和第 268 页的『创建队列管理器以在高可用性 (HA) 集群中使用』中的 7 中显示的 `addmqinf` 命令，将队列管理器配置信息添加到节点。
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 WebSphere MQ 安装发出用于启动和停止队列管理器的命令。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](#)。

4. 卸载队列管理器文件系统。

在高可用性 (HA) 集群的控制下启动队列管理器

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 amqzma0 amqfcxba amqfqpud amqpcsea amqzlaa0 \
  amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqimppa \
  amqzfuma amqzmaa 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 stirmqm command does not use the -x flag.
stirmqm ${QM}
```

您可以修改脚本以启动其他相关程序。

在高可用性 (HA) 集群的控制下停止队列管理器

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。

以下脚本是一个示例，说明如何在不假设队列管理器当前状态的情况下立即停止。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 269 页的『在高可用性 \(HA\) 集群的控制下启动队列管理器』](#)中提供了 shell 脚本示例)：

```
#!/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 amqfcpub amqpcsea amqzlaa0 \
            amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqzmpa \
            amqzfuma amqzdmaa 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 amqzma0 | 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

```

监视队列管理器

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 WebSphere MQ V 7.1 开始，可以在系统上共存多个 WebSphere 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 V 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。如果您有主安装，或者您正在使用版本低于 7.1 的脚本，那么无需指定 `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 V 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 WebSphere 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 269 页](#)的『在高可用性 (HA) 集群的控制下启动队列管理器』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 stirmqm | 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

```

将队列管理器置于高可用性 (HA) 集群的控制之下

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。

要定义资源组以包含队列管理器及其所有关联资源，请完成以下步骤：

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 集群是否正确地在集群节点之间切换资源并准备好控制队列管理器。

从高可用性 (HA) 集群节点中删除队列管理器

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。

要从 HA 集群中的节点除去队列管理器，请完成以下步骤：

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
```

要完全删除队列管理器，请使用 `dltmqm` 命令。但是，请注意，这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。

支持 Microsoft 集群服务 (MSCS)

引入并设置 MSCS 以支持虚拟服务器的故障转移。

此信息仅适用于 **WebSphere MQ for Windows**。

Microsoft 集群服务 (MSCS) 使您能够将服务器连接到集群中，从而提供更高的数据和应用程序可用性，并使管理系统更容易。MSCS 可自动检测服务器或应用程序故障并从中恢复。

MSCS 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 通过此过程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，然后在该计算机上重新启动应用程序。

本部分介绍了 MSCS 集群，并在以下部分中描述了如何设置 MSCS 支持：

- [第 273 页的『介绍 MSCS 集群』](#)
- [第 274 页的『为 MSCS 集群设置 IBM WebSphere MQ』](#)

然后，在以下部分中告诉您如何为 MSCS 集群配置 WebSphere MQ：

- [第 275 页的『创建队列管理器以用于 MSCS』](#)
- [第 276 页的『将队列管理器移至 MSCS 存储器』](#)
- [第 277 页的『将队列管理器置于 MSCS 控制之下』](#)
- [第 282 页的『从 MSCS 控件中除去队列管理器』](#)

然后，提供有关将 MSCS 与 WebSphere MQ 配合使用的一些有用提示，并在以下部分中详细说明 WebSphere MQ MSCS 支持实用程序：

- [第 283 页的『有关使用 MSCS 的提示和技巧』](#)
- [第 285 页的『IBM WebSphere MQ MSCS 支持实用程序』](#)

介绍 MSCS 集群

MSCS 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配置：如果其中一台计算机发生故障，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传输到集群中的另一台计算机，然后在该计算机上重新启动其操作。

[第 264 页的『将 WebSphere MQ 与高可用性配置配合使用』](#) 包含 MSCS 集群，多实例队列管理器和 WebSphere 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 WebSphere 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。另外还有另一个共享磁盘，必须是独立磁盘的冗余阵列 (RAID) 级别 1，供 MSCS 独占使用；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 274 页的图 62](#) 所示。

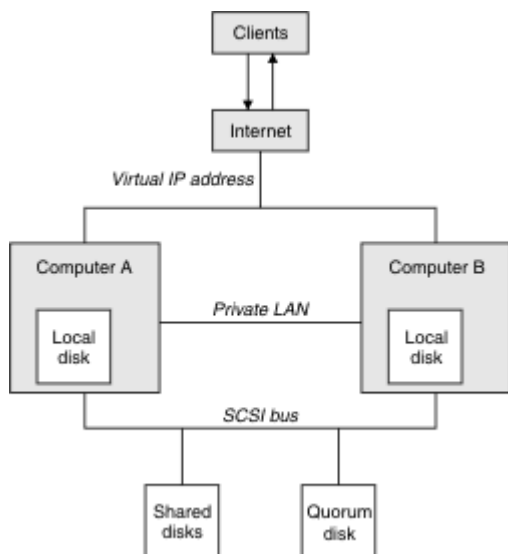


图 62: 双计算机 MSCS 集群

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。

为 MSCS 集群设置 IBM WebSphere 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 WebSphere 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。

要为此设置系统，请首先在集群中的每台计算机上安装 IBM WebSphere MQ。

由于队列管理器与 IBM WebSphere MQ 安装名称相关联，因此集群中所有计算机上的 IBM WebSphere 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 WebSphere MQ 随附的工具对其进行迁移；请参阅第 276 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 275 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 277 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 282 页的『从 MSCS 控件中除去队列管理器』中描述的过程。

设置对称性

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。尤其是，请确保安装在两台计算机上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 274 页的『MSCS 安全性』中所述。

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 WebSphere MQ 相关的帐户是域帐户，那么会自动发生此情况。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，WebSphere 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 WebSphere MQ 服务。确保它存在于集群中每台计算机上的本地 mqm 组中。

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。

集群方式

您可以使用 WebSphere MQ 以两种方式运行集群系统：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 WebSphere 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的计算机的全部电源。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。

创建队列管理器以用于 MSCS

此过程确保以适合在 MSCS 控制下进行准备和放置的方式创建新的队列管理器。

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 WebSphere 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 WebSphere 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:\WebSphere 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。如果未设置该变量，那么将在队列管理器的 WebSphere 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 276 页的『将队列管理器移至 MSCS 存储器』。

使用 WebSphere MQ Explorer 创建队列管理器

执行以下步骤以使用 IBM WebSphere 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 WebSphere MQ Explorer。
2. 在 "Navigator" 视图中，展开树节点以查找 Queue Managers 树节点。
3. 右键单击 Queue Managers 树节点，然后选择 New->Queue Manager。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 Next>。
5. 完成对话框 (步骤 2)，然后单击 Next>。
6. 完成对话框 (步骤 3)，确保未选择 Start Queue Manager 和 Create Server Connection Channel，然后单击 Next>。
7. 完成对话框 (步骤 4)，然后单击 Finish。
8. 转至第 276 页的『将队列管理器移至 MSCS 存储器』。

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 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\不要 尝试手动移动文件；使用作为 WebSphere 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 SSL 连接，并且 SSL 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 SSL 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` 表示安装了 WebSphere MQ 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 WebSphere MQ Explorer 或 MQSC 命令 ALTER QMGR 修改队列管理器属性 SSLKEYR，以指向新的 SSL 密钥存储库文件。

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 277 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 284 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:\
WebSphere MQ" /ld "e:\
WebSphere 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 *WebSphere MQ*。如果这些目录尚不存在，那么将创建这些目录。

8. 使用 IBM WebSphere 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 Start。队列管理器将启动。
 - b. 右键单击 Queues 树节点，然后选择 New->Local Queue...，并为队列提供名称。
 - c. 单击 Finish。
 - d. 右键单击队列，然后选择 Put Test Message...。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 Put Test Message，然后关闭面板。
 - f. 右键单击队列，然后选择 Browse Messages...。将显示 "消息浏览器" 面板。
 - g. 确保消息位于队列上，然后单击 Close。"消息浏览器" 面板将关闭。
 - h. 右键单击队列，然后选择 Clear Messages...。将清除队列上的消息。
 - i. 右键单击队列，然后选择 Delete...。将显示确认面板，单击 OK。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 Stop...。将显示 "结束队列管理器" 面板。
 - k. 单击 OK。队列管理器停止。
9. 作为 WebSphere MQ 管理员，请确保将队列管理器的启动属性设置为手动。在 IBM WebSphere MQ Explorer 的 "队列管理器属性" 面板中，将 "启动" 字段设置为 manual。
10. 转至 [第 277 页的『将队列管理器置于 MSCS 控制之下』](#)。

将队列管理器置于 MSCS 控制之下

将队列管理器置于 MSCS 控制下所涉及的任务，包括先决条件任务。

在将队列管理器置于 MSCS 控制之下之前

在将队列管理器置于 MSCS 控制之下之前，请执行以下任务：

1. 确保 IBM WebSphere MQ 及其 MSCS 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 274 页的『为 MSCS 集群设置 IBM WebSphere MQ』](#) 中所述。
2. 使用 **haregtyp** 实用程序在所有集群节点上将 WebSphere MQ 注册为 MSCS 资源类型。要获取更多信息，请参阅 [第 285 页的『IBM WebSphere MQ MSCS 支持实用程序』](#)。
3. 如果尚未创建队列管理器，请参阅 [第 275 页的『创建队列管理器以用于 MSCS』](#)。
4. 如果您已创建队列管理器，或者该队列管理器已存在，请确保您已执行 [第 276 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 WebSphere MQ Explorer 来停止该队列管理器。
6. 在继续执行本主题中的以下任一 Windows 过程之前，请测试共享驱动器的 MSCS 操作。

Windows Server 2012



注意: MSCS 支持在 WebSphere MQ 7.5 中使用 32 位 DLL 提供。由于 Windows 2012 中的限制, IBM WebSphere MQ 队列管理器不会在重新启动后进行故障转移。

Microsoft 已不推荐将 32 位 DLL 与 Windows 2012 配合使用, 因此当前没有可用于此问题的操作系统修订。IBM 没有为 IBM WebSphere MQ 7.5 提供 64 位库。

从 IBM MQ 8.0 提供了 64 位库, 因此必须将此版本的产品用于 Windows 2012 和更高版本的完整 MSCS 功能。

要将队列管理器置于 Windows Server 2012 上的 MSCS 控制之下, 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. 与先前版本的 Windows 上 MSCS 集群管理员中使用的组方案不同, 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 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 **IBM MQSeries MSCS** 资源。
 - f. 完成向导中的其余步骤。
5. 在使资源联机之前, IBM 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 IBM 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; 以实现更好的隔离
 - 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 WebSphere 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 WebSphere 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 285 页的『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 285 页的『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 282 页的『looksAlive 和 isAlive 轮询的摘要』。

- d. (可选) 设置首选节点 (但请注意第 285 页的『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 WebSphere MQ Explorer。例如:
 - a. 右键单击 Queues 树节点, 然后选择 New->Local Queue..., 并为队列提供名称。
 - b. 单击 Finish。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 Put Test Message...。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 Put Test Message, 然后关闭面板。
 - e. 右键单击队列, 然后选择 Browse Messages...。将显示 "消息浏览器" 面板。
 - f. 确保消息位于队列上, 然后单击 Close。"消息浏览器" 面板将关闭。
 - g. 右键单击队列, 然后选择 Clear Messages...。将清除队列上的消息。
 - h. 右键单击队列, 然后选择 Delete...。将显示确认面板, 单击 OK。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 276 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

要在 Windows Server 2008 上将队列管理器置于 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. 与先前版本的 Windows 上 MSCS 集群管理员中使用的组方案不同, 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 **IBM MQSeries MSCS** 资源。
 - f. 完成向导中的其余步骤。
5. 在使资源联机之前，IBM 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 IBM 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; 以实现更好的隔离
 - 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 WebSphere 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 WebSphere 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 285 页的『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 285 页的『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 282 页的『looksAlive 和 isAlive 轮询的摘要』。
 - d. (可选) 设置首选节点 (但请注意第 285 页的『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 WebSphere MQ Explorer。例如:
- a. 右键单击 Queues 树节点，然后选择 New->Local Queue...，并为队列提供名称。
 - b. 单击 Finish。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 Put Test Message...。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 Put Test Message，然后关闭面板。
 - e. 右键单击队列，然后选择 Browse Messages...。将显示 "消息浏览器" 面板。
 - f. 确保消息位于队列上，然后单击 Close。"消息浏览器" 面板将关闭。
 - g. 右键单击队列，然后选择 Clear Messages...。将清除队列上的消息。
 - h. 右键单击队列，然后选择 Delete...。将显示确认面板，单击 OK。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。这可能需要在一些时间。(如果要在其他时间将队列管理器快速移动到其它节点，请遵循第 276 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 275 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 WebSphere 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; 以实现更好的隔离
- 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 WebSphere 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 WebSphere 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 285 页的『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 285 页的『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 282 页的『looksAlive 和 isAlive 轮询的摘要』。

8. (可选) 设置首选节点 (但请注意 第 285 页的『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 WebSphere MQ Explorer。例如:
 - a. 右键单击 Queues 树节点, 然后选择 New->Local Queue..., 并为队列提供名称。
 - b. 单击 Finish。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 Put Test Message...。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 Put Test Message, 然后关闭面板。
 - e. 右键单击队列, 然后选择 Browse Messages...。将显示 "消息浏览器" 面板。
 - f. 确保消息位于队列上, 然后单击 Close。"消息浏览器" 面板将关闭。
 - g. 右键单击队列, 然后选择 Clear Messages...。将清除队列上的消息。
 - h. 右键单击队列, 然后选择 Delete...。将显示确认面板, 单击 OK。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。(如果您希望在其他时间将队列管理器快速移动到其它节点, 请遵循 第 276 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。

looksAlive 和 isAlive 轮询的摘要

looksAlive 和 isAlive 是 MSCS 重新调用资源类型提供的库代码以及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。

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 将根据其他高级 MSCS 选项, 针对资源和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](#)。

从 MSCS 控件中除去队列管理器

您可以从 MSCS 控制中除去队列管理器, 并将其返回到手动管理。

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。

如果要除去的队列管理器使用 SSL 连接, 那么必须使用 WebSphere 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 SSL 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 283 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 283 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. 选择 Groups，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 WebSphere 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\hamvmqm 实用程序：

1. 关闭队列管理器，并检查是否没有错误。
2. 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 284 页的『队列管理器日志文件』以了解这很重要的原因）。
3. 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。
4. 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。
5. 运行实用程序以移动队列管理器，如下所示：

```
hamvmqm /m qmname /dd "c:\
WebSphere MQ" /ld "c:\
WebSphere 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 *WebSphere MQ*（如果这些目录尚不存在，那么将创建这些目录）。

6. 测试队列管理器以确保其工作（如第 276 页的『将队列管理器移至 MSCS 存储器』中所述）。

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 WebSphere MQ 对 MSCS 的支持。

本部分包含一些常规信息，可帮助您有效使用 WebSphere MQ 对 MSCS 的支持。

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。

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 275 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它节点，并且可以使该资源脱机。

手动启动

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 WebSphere MQ MSCS 支持可以重新启动 IBM MQSeries 服务，而无需立即启动队列管理器。

WebSphere 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，但它本身必须控制哪些队列管理器正在运行以及哪些机器上运行。请参阅第 276 页的『将队列管理器移至 MSCS 存储器』以获取更多信息。

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 WebSphere MQ 配合工作，节点 A 上的每个节点都需要一个相同的队列管理器。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 276 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; WebSphere 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 285 页的『IBM WebSphere MQ MSCS 支持实用程序』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 274 页的图 62)。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 275 页的『创建队列管理器以用于 MSCS』
- 第 276 页的『将队列管理器移至 MSCS 存储器』
- 第 277 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

将队列管理器移动到 MSCS 存储器时，会将其日志和数据文件移动到共享磁盘(有关示例，请参阅第 276 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

WebSphere 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。

始终使用 MSCS 来管理集群

请勿尝试使用控制命令或 IBM WebSphere 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。

在主动/主动方式下工作

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 WebSphere MQ 与 MSCS 和 COM + 配合使用:

- 使用 WebSphere MQ 的 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。

PostOnline 命令和 PreOffline 命令

使用这些命令可将 WebSphere MQ MSCS 支持与其他系统集成。您可以使用它们来发出 WebSphere MQ 命令, 但有一些限制。

在 "参数" 中对类型为 IBM WebSphere MQ MSCS 的资源指定这些命令。您可以使用它们将 WebSphere 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。调用时, 缺省情况下将从 Windows 系统目录运行这些命令。由于 WebSphere MQ 使用 32 位资源监视器进程, 因此在 Windows 64 位系统上, 这是 \Windows\SysWOW64 目录而不是 \Windows\system32。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; WebSphere MQ MSCS 支持不会等待它们完成后再继续。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 WebSphere 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。

使用首选节点

当使用 "主动/主动" 方式为每个队列管理器配置 首选节点时, 它很有用。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。

如果应用程序事件日志中发生 COM + 错误

在新安装的 MSCS 集群上安装 WebSphere MQ 时, 可能会发现 "应用程序事件" 日志中报告的 "源 COM +" 和 "事件标识 4691" 错误。

这意味着当 Microsoft 分布式事务协调程序 (MSDTC) 尚未配置为在 Microsoft Cluster Server (MSCS) 环境中运行时, 您将尝试在该环境中运行 WebSphere MQ。有关在集群环境中配置 MSDTC 的信息, 请参阅 Microsoft 文档。

IBM WebSphere MQ MSCS 支持实用程序

可以在命令提示符处运行的 MSCS 实用程序的 IBM WebSphere MQ 支持的列表。

对 MSCS 的 IBM WebSphere 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 WebSphere 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 283 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 WebSphere 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 276 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 WebSphere 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 WebSphere 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。

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。

第 287 页的图 63 显示了 QM1 的多实例配置。IBM WebSphere 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。

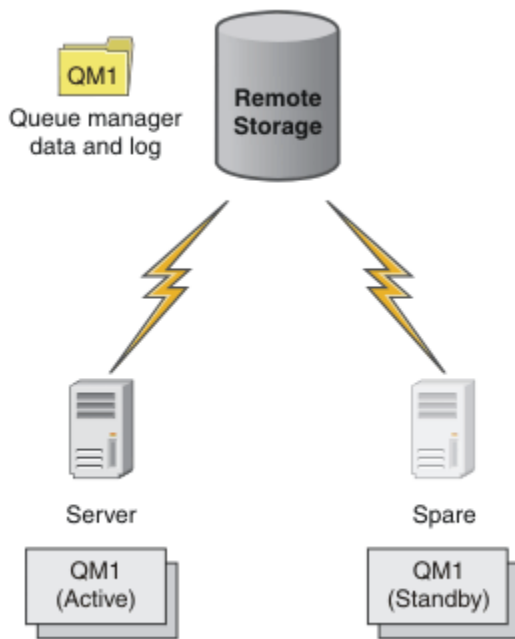


图 63: 多实例队列管理器

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。
- 队列管理器进程的无响应能力 (由 WebSphere MQ 检测到)，然后将关闭队列管理器。

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。

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，以将 WebSphere MQ 连接传输到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。您尤其需要密切关注事务性应用程序的行为，以及浏览 WebSphere MQ 队列的应用程序。
- 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。

WebSphere MQ MQI 客户机和通道在变为活动时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 WebSphere MQ Java 类不支持客户机自动重新连接。

支持的平台

您可以在任何非 z/OS 平台上从 V 7.0.1 创建多实例队列管理器。

从 V 7.0.1 开始，MQI 客户机支持自动客户机重新连接。

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 WebSphere 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 UNIX 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 **dspmqlnf** 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 UNIX and Linux 上，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 310 页的『在 Windows 上保护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 290 页的『在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夹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 302 页的『在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夹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 WebSphere 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 WebSphere MQ。**dspmqlnf** 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 *mqm.ini* 和 *qm.ini* 文件。主题第 320 页的『在 Linux 上创建多实例队列管理器』，第 290 页的『在域工作站或服务器上创建多实例队列管理器』和 第 302 页的『在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 Windows UNIX and Linux 系统上，您可以通过将单个 `mqm.ini` 文件放在网络共享上并将 `AMQ_QMS_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 WebSphere 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 WebSphere 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。
5. 在 UNIX 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。

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。

在配置多实例队列管理器之前，请阅读第 313 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 310 页的『[在 Windows 上保护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 290 页的『[在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 Version 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。在 Version 7.1 之前，多实例队列管理器仅在域控制器上运行；请参阅第 290 页的『[在域控制器上运行多实例队列管理器](#)』。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另一队列管理器。

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。其他安全组在 `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。

创建多实例队列管理器的任务与 Version 7.0.1 中具有一个更改的任务相同。必须将其他安全组名添加到 `crtmqm` 命令的参数中。此任务在第 290 页的『[在域工作站或服务器上创建多实例队列管理器](#)』中进行了描述。

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 313 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 314 页的『[读取和写入本地 mqm 组授权的数据和日志文件](#)』和第 317 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 310 页的『[在 Windows 上保护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 WebSphere 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 292 页的『[为 IBM WebSphere MQ 创建 Active Directory 和 DNS 域](#)』。
2. 第 295 页的『[在 Windows 域中的服务器或工作站上安装 IBM WebSphere MQ](#)』。

3. [第 297 页的『为队列管理器数据和日志文件创建共享目录』](#)。
4. [第 300 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 290 页的『在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

在 Version 7.0.1 中，多实例队列管理器仅在域控制器上运行。可以使用域 `mqm` 组来保护队列管理器数据。如 [第 310 页的『在 Windows 上保护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 WebSphere 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 302 页的『在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

[Windows 2000，Windows Server 2003 和 Windows Server 2008 集群节点作为域控制器](#)

在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 WebSphere 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 WebSphere 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 UNIX 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 WebSphere 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 292 页的『为 IBM WebSphere MQ 创建 Active Directory 和 DNS 域』](#)。
2. [第 295 页的『在 Windows 域中的服务器或工作站上安装 IBM WebSphere MQ』](#)。
3. [第 297 页的『为队列管理器数据和日志文件创建共享目录』](#)。
4. [第 300 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 WebSphere MQ for Windows。

- a) 执行任务 第 295 页的『在 Windows 域中的服务器或工作站上安装 IBM WebSphere 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 WebSphere 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 WebSphere 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 300 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 **dspmqlinf** 命令：

```
dspmqlinf QMGR
```

系统响应：

```
QueueManager:
  Name=QMGR
  Directory=QMGR
  Prefix=C:\Program Files\IBM\WebSphere MQ
  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 **dspmqlinf** 命令。

```
dspmqlinf -o command QMGR
```

系统响应：

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"
-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="C:\Program Files\IBM\WebSphere MQ"
-v DataPath=\\sun\wmq\data\QMGR
```

WebSphere 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>dspmqr -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:

```
WebSphere 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'.
WebSphere 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:

```
WebSphere 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:

```
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ended, permitting switchover to
a standby instance.
```

在 *mars* 上:

```
dspmqr
QMNAME(QMGR) STATUS(Running as standby)
C:\Users\wmquser2>dspmqr
QMNAME(QMGR) STATUS(Running as standby)
C:\Users\wmquser2>dspmqr
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 308 页的『在 Windows 上验证多实例队列管理器』](#)。

为 *IBM WebSphere MQ* 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 WebSphere 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 WebSphere 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 WebSphere 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 289 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 WebSphere 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 会自动安装这些文件。
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 WebSphere 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\Administrator* 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的 "准备 IBM WebSphere MQ" 向导的行为。这将导致 "准备 IBM WebSphere MQ" 向导自动将组 Domain mqm 添加到域中 IBM WebSphere 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。如果要这样做，那么必须定义一个与 Domain mqm 组具有相同属性的组。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 WebSphere 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 WebSphere 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对列管理器。
- 在域中的工作站或服务器上安装 IBM WebSphere MQ 时指定的域用户必须是 Domain mqm 组的成员，或者是使用与 Domain mqm 组相同的属性定义的备用组的成员。

- 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，选择列表底部的 **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。

一个用户 (示例中的 *wmquser1*) 运行 IBM WebSphere 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对列管理器。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。"准备 IBM WebSphere MQ" 向导会自动将 Domain mqm 配置为运行向导的本地 mqm 组的成员。

必须提供不同的用户才能在单台计算机上针对 IBM WebSphere MQ 的每个安装运行 IBM WebSphere 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 WebSphere MQ 作为服务运行。

如果需要将 IBM WebSphere MQ 作为服务运行, 然后向域用户 (从域管理员处获取) 授予作为服务运行的权限, 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 Vista 和 Windows Server 2008 上, 缺省情况下已启用用户帐户控制 (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 295 页的『在 Windows 域中的服务器或工作站上安装 IBM WebSphere MQ』](#)。

在 Windows 域中的服务器或工作站上安装 IBM WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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. 执行第 292 页的『为 IBM WebSphere MQ 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，以了解您可以在其上运行 IBM WebSphere 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 WebSphere 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 289 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 WebSphere 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 "本地区域连接属性" 窗口的项列表中选择 **Internet Protocol 版本 4** 或 **Internet Protocol 版本 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 WebSphere MQ for Windows。

有关运行 IBM WebSphere MQ for Windows 安装向导的更多信息; 请参阅 [在 Windows 上安装 IBM WebSphere MQ 服务器](#)。

 - a) 在 *mars* 上，以本地管理员 *mars\Administrator* 身份登录。

b) 在 IBM WebSphere MQ for Windows 安装介质上运行 **Setup** 命令。

IBM WebSphere 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 292 页的『为 IBM WebSphere MQ 创建 Active Directory 和 DNS 域』为这组任务配置域用户标识。

e) 单击 **WebSphere MQ 安装**，选择安装语言，然后单击 "启动 IBM WebSphere 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 WebSphere MQ 已安装，安装程序将启动 "准备 IBM WebSphere MQ" 向导。

要点: 请不要运行向导。

5. 使用 **作为服务运行** 权限配置要运行 IBM WebSphere 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 WebSphere 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“准备 IBM WebSphere MQ”向导。

有关运行“准备 IBM WebSphere MQ”向导的更多信息; 请参阅 [使用“准备 WebSphere MQ”向导配置 WebSphere MQ](#)。

a) IBM WebSphere MQ 安装程序会自动运行“准备 IBM WebSphere MQ”。

要手动启动向导，请在 **启动 > 所有程序 > IBM WebSphere MQ** 文件夹中找到“准备 IBM WebSphere MQ”的快捷方式。选择与多安装配置中的 IBM WebSphere 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 WebSphere 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 WebSphere 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**。

此向导使用 `wmquser1` 配置并启动 IBM WebSphere 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 314 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 297 页的『[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。

相关概念

[WebSphere MQ Windows 服务所需的用户权限](#)

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 WebSphere 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 WebSphere 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 292 页的『为 IBM WebSphere MQ 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 289 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 WebSphere 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 WebSphere MQ 服务器上本地 `mqm` 组的成员。因此，*wmquser1* 和 *wmquser2* 是两个 IBM WebSphere MQ 服务器上本地 `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 WebSphere MQ 服务器读取文件并将其写入共享目录。检查 IBM WebSphere 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 300 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』](#)。

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 WebSphere 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 WebSphere 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 WebSphere 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 292 页的『为 IBM WebSphere MQ 创建 Active Directory 和 DNS 域』。
2. 第 295 页的『在 Windows 域中的服务器或工作站上安装 IBM WebSphere MQ』。
3. 第 297 页的『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 289 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 WebSphere 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, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

并：

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. 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:

```
WebSphere 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:

```
WebSphere 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'.
WebSphere 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, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)
AMQ8006: WebSphere 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 **amqspout** 放置测试消息。

```
echo 'A test message' | amqspout 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:

```
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ended.
```

6. 删除队列管理器。

```
dltmqm QMGR
```

系统响应:

```
WebSphere 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
```

在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”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 290 页的『在域工作站或服务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 WebSphere 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 UNIX and Linux。在 Windows 上, 检查始终成功。
2. 执行第 292 页的『为 IBM WebSphere MQ 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 305 页的『将第二个域控制器添加到 `wmq.example.com` 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 WebSphere MQ for Windows，并验证安装。
4. 执行第 307 页的『在 `wmq.example.com` 域中的域控制器上安装 IBM WebSphere MQ』中的步骤以在两个域控制器上安装 IBM WebSphere 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：

```
WebSphere 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.
```

- 在 *sun* 上启动队列管理器，允许备用实例。

```
strmqm -x QMGR
```

系统响应:

```
WebSphere 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'.  
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

- 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=C:\Program Files\IBM\WebSphere MQ  
  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="C:\Program Files\IBM\WebSphere MQ"  
-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
```

WebSphere MQ configuration information added.

- 在 *earth* 上启动队列管理器的备用实例。

```
strmqm -x QMGR
```

系统响应:

```
WebSphere 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:

```
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere 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

要使用本程序验证多实例队列管理器, 请参阅 [第 308 页的『在 Windows 上验证多实例队列管理器』](#)。

相关任务

[第 305 页的『将第二个域控制器添加到 *wmq.example.com* 域』](#)

[第 307 页的『在 *wmq.example.com* 域中的域控制器上安装 IBM WebSphere MQ』](#)

相关信息

[Windows 2000, Windows Server 2003 和 Windows Server 2008 集群节点作为域控制器](#)

将第二个域控制器添加到 *wmq.example.com* 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 WebSphere 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 292 页的『为 IBM WebSphere MQ 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 289 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 "本地区域连接属性" 窗口的项列表中选择 **Internet Protocol 版本 4** 或 **Internet Protocol 版本 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 会自动安装这些文件。
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 WebSphere MQ; 请参阅 [第 307 页的『在 `wmq.example.com` 域中的域控制器上安装 IBM WebSphere MQ』](#)。

相关任务

[第 292 页的『为 IBM WebSphere MQ 创建 Active Directory 和 DNS 域』](#)

在 *wmq.example.com* 域中的域控制器上安装 *IBM WebSphere MQ*

在 *wmq.example.com* 域中的两个域控制器上安装和配置 IBM WebSphere MQ 的安装。
在此处放置简短描述; 用于第一段和摘要。

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 WebSphere 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 292 页的『为 IBM WebSphere MQ 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 305 页的『将第二个域控制器添加到 `wmq.example.com` 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 WebSphere 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 WebSphere MQ 的安装。

过程

1. 在 *sun* 和 *earth* 上安装 IBM WebSphere MQ。

有关运行 IBM WebSphere MQ for Windows 安装向导的更多信息; 请参阅 [在 Windows 上安装 IBM WebSphere MQ 服务器](#)。

- a) 在 *sun* 和 *earth* 上，以域管理员 *wmq\Administrator* 身份登录。
- b) 在 IBM WebSphere MQ for Windows 安装介质上运行 **Setup** 命令。

IBM WebSphere 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 **WebSphere MQ 安装**，选择安装语言，然后单击 "启动 IBM WebSphere 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 WebSphere MQ 已安装，安装程序将启动 "准备 IBM WebSphere MQ" 向导。

IBM WebSphere 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 "准备 IBM WebSphere MQ" 向导。

有关运行 "准备 IBM WebSphere MQ" 向导的更多信息，请参阅 [使用 "准备 WebSphere MQ" 向导配置 WebSphere MQ](#)。

- a) IBM WebSphere MQ 安装程序会自动运行 "准备 IBM WebSphere MQ"。

要手动启动向导，请在 **启动 > 所有程序 > IBM WebSphere MQ** 文件夹中找到 "准备 IBM WebSphere MQ" 的快捷方式。选择与多安装配置中的 IBM WebSphere 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**。

"准备 IBM WebSphere MQ" 向导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 WebSphere 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 ...** 并输入 `mqm`；单击 **检查名称 > 确定 > 确定**

结果

1. 检查 "准备 IBM WebSphere MQ" 是否创建了域用户 `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 **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 314 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 302 页的『在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。

相关概念

[WebSphere MQ Windows 服务所需的用户权限](#)

在 *Windows* 上验证多实例队列管理器

使用样本程序 `amqsgshac`，`amqspshac` 和 `amqsmshac` 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。

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 WebSphere MQ 更简单，就像它不在域中一样。

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 302 页的『在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 XP Service Pack 2 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 WebSphere 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 WebSphere 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: 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 TCP/IP 侦听器正在运行且具有 `Control = Queue Manager` 的 WebSphere 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 WebSphere 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>
```

在 Windows 上保护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 WebSphere 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 WebSphere MQ 服务的用户标识以访问这些目录。

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。即使您使用该用户成为域用户，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 311 页的图 64。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。使用 `crtmqm` 命令上的 `-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。名称在 `qm.ini` 文件的 **AccessMode** 节中作为 `SecurityGroup` 进行传输；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。

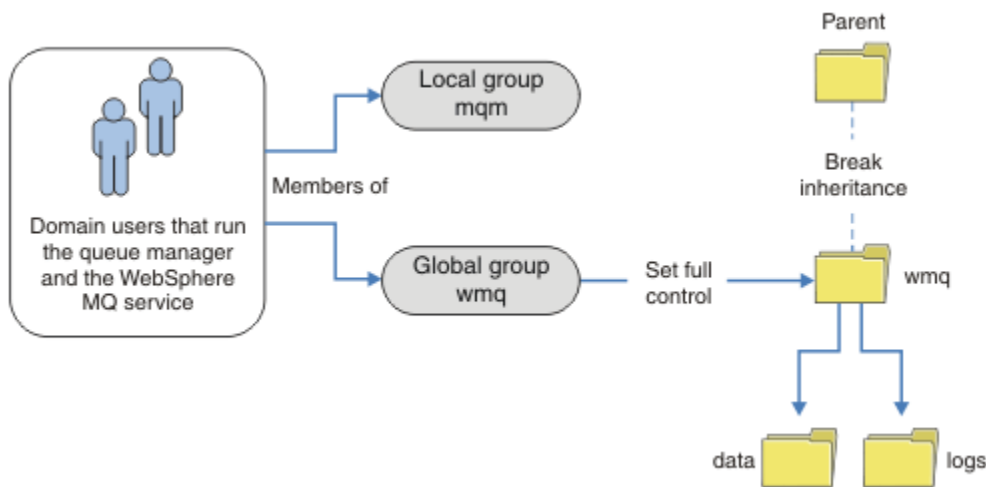


图 64: 使用备用全局安全组 (1) 保护队列管理器数据和日志

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 311 页的图 64 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 WebSphere MQ 服务器，那么第 311 页的图 64 中的用户分组不方便。您需要重复将用户添加到每个 IBM WebSphere 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 WebSphere MQ 成员的用户成为 Domain `mqm` 组的成员；请参阅第 312 页的图 65。将 IBM WebSphere MQ 作为域安装安装时，“准备 IBM WebSphere MQ”向导会自动使 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: 相同用户可以在不同服务器上运行 IBM WebSphere MQ，但在单个服务器上，您必须具有不同的用户才能将 IBM WebSphere 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。

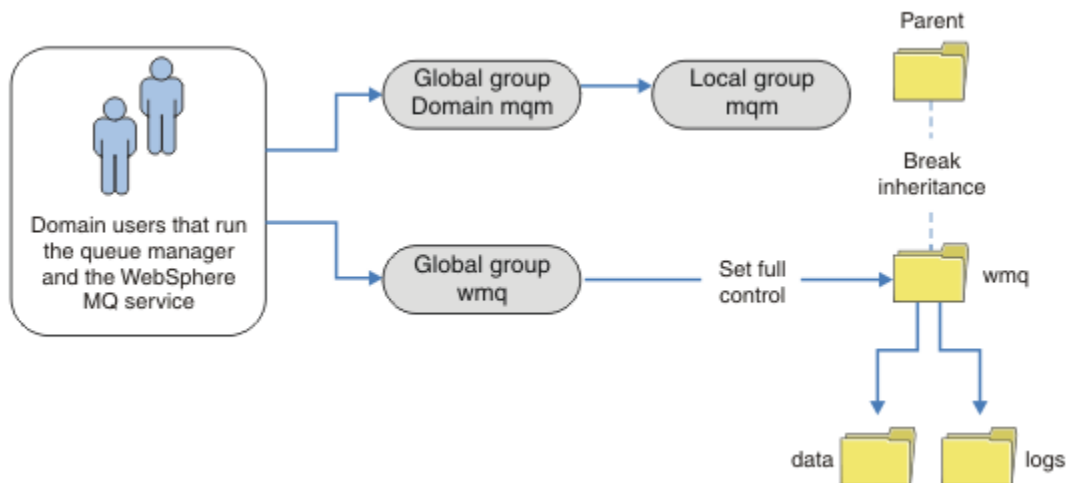


图 65: 使用备用全局安全组 (2) 保护队列管理器数据和日志

第 312 页的图 65 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 312 页的图 66。

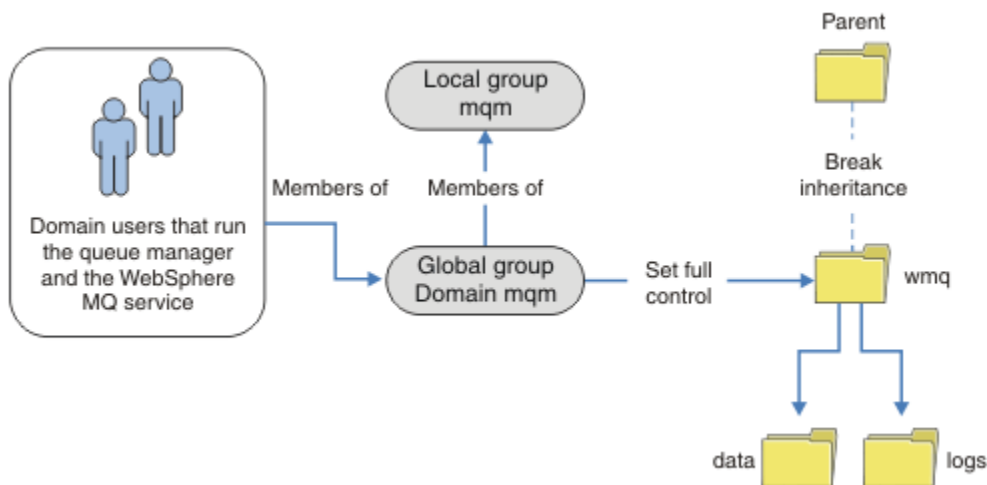


图 66: 使用备用全局安全组 (3) 保护队列管理器数据和日志

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 313 页的图 67。在 第 313 页的图 67 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 WebSphere 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 WebSphere MQ 的安装将 Domain mqm 传播到 第 312 页的图 66 中的本地 mqm 组，因为有两组用户。

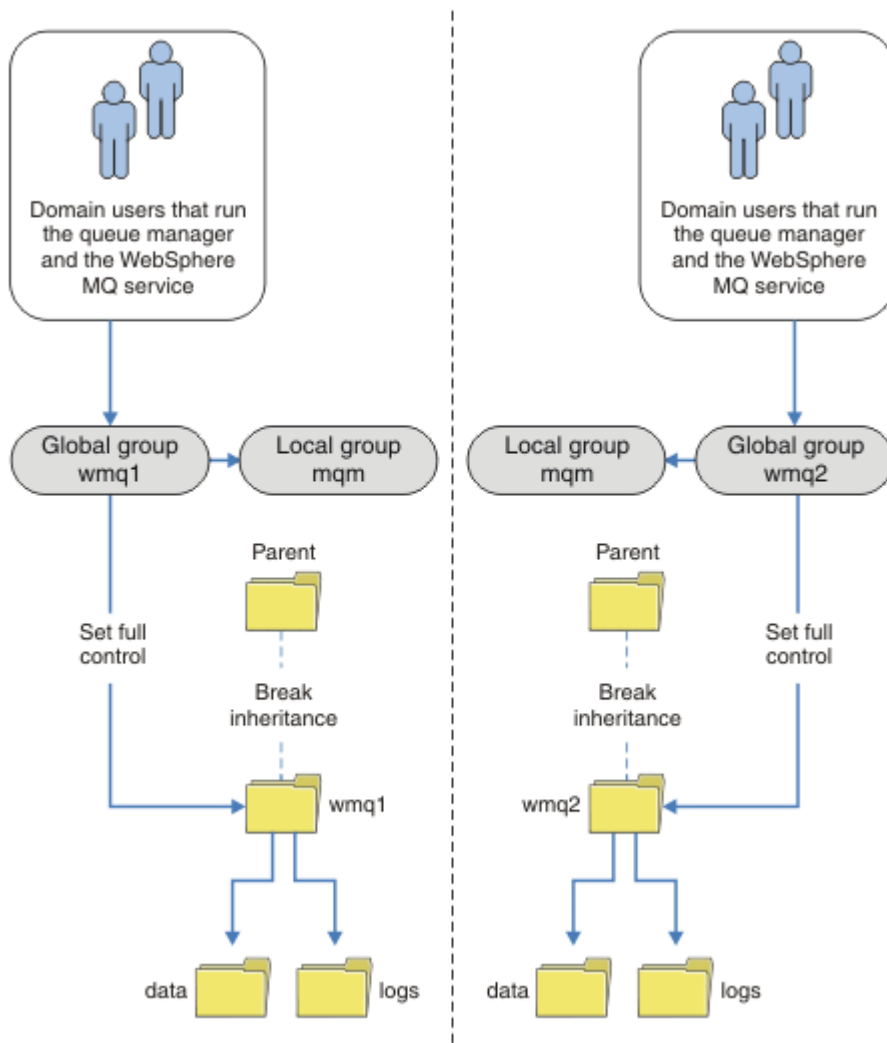


图 67: 使用备用全局安全性主体 (4) 保护队列管理器数据和日志

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 312 页](#)的图 66 中显示的更简单的模型。

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 WebSphere 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 WebSphere MQ 服务的用户标识以访问这些目录。

两个示例演示如何配置 IBM WebSphere MQ 的访问控制。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 314 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 317 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 310 页](#)的『在 Windows 上保护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 WebSphere 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 WebSphere 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 314 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 314 页的图 68](#) 的步骤。本地组 `wmq` 是备用安全组的示例。

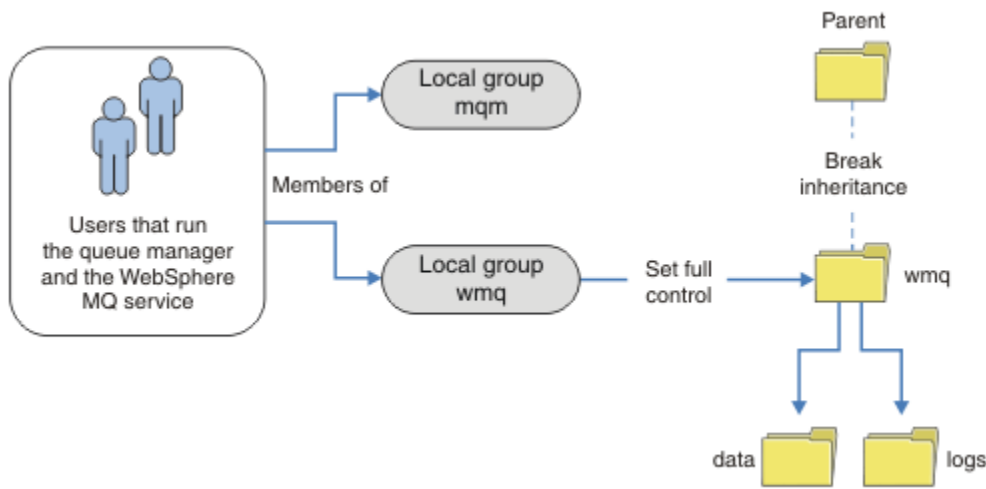


图 68: 使用备用本地安全组 `wmq` 保护队列管理器数据和日志

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 310 页的『在 Windows 上保护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。

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 WebSphere MQ for Windows 作为主安装。

2. 运行“准备 IBM WebSphere MQ”向导。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 289 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 289 页的『Windows 域和多实例队列管理器』一起执行。

在 Windows 上，可以在您选择的任何目录中创建 IBM WebSphere MQ for Windows 的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 WebSphere 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 WebSphere MQ 安装配置为域的一部分，请用户标识成为 Domain mqm 组的成员。

“准备 IBM WebSphere 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：

```
WebSphere 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:

```
WebSphere 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'.  
WebSphere 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, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: WebSphere 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:

```
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere MQ queue manager 'QMGR' ended.
```

6. 删除队列管理器。

```
dltmqm QMGR
```

系统响应:

```
WebSphere 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 289 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

第 317 页的『[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 300 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 290 页的『[在域工作站或服务器上创建多实例队列管理器](#)』

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 WebSphere MQ for Windows 作为主安装。
2. 运行“准备 IBM WebSphere MQ”向导。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 289 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 289 页的『[Windows 域和多实例队列管理器](#)』一起执行。

在 Windows 上, 可以在您选择的任何目录中创建 IBM WebSphere MQ for Windows 的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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere MQ 作为域的一部分安装，并且要执行第 290 页的『在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 292 页的『为 IBM WebSphere MQ 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 WebSphere 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 WebSphere 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, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

并：

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. 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 WebSphere MQ 安装配置为域的一部分, 请用户标识成为 `Domain mqm` 组的成员。

“准备 IBM WebSphere 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:

```
WebSphere 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:

```
WebSphere 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'.
WebSphere 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, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QMGR.

```
1 : define qlocal(QTEST)
AMQ8006: WebSphere 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:

```
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ended.
```

6. 删除队列管理器。

```
dltmqm QMGR
```

系统响应:

```
WebSphere 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
```

在 Linux 上创建多实例队列管理器

提供了用于说明如何在 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。此示例基于 Linux Red Hat Enterprise 5。这些步骤在其他 UNIX 平台上有所不同。

此示例是在运行 Windows XP Service Pack 2 的 3 GB RAM 的 2 GHz 笔记本电脑上设置的。两个 VMware 虚拟机 (Server1 和 Server2) 在 640 MB 映像中运行 Linux Red Hat Enterprise 5。Server1 托管网络文件系统 (NFS), 队列管理器日志和 HA 实例。通常, NFS 服务器也会托管其中一个队列管理器实例; 这是为了简化此示例。Server2 将 Server1 的队列管理器日志与备用实例一起安装。WebSphere MQ MQI 客户机安装在

额外的 400 MB VMware 映像上，该映像运行 Windows XP Service Pack 2 并运行样本高可用性应用程序。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。

示例

表 30: Linux 上的说明性多实例队列管理器配置	
Server1	Server2
以 root 用户身份登录	
遵循 安装 IBM WebSphere MQ 中的指示信息来安装 WebSphere MQ，创建 mqm 用户和组 (如果这些用户和组不存在)，并定义 /var/mqm。 检查第一台机器上 /etc/passwd 中针对 mqm 显示的 uid 和 gid；例如， mqm:x:501:100:MQ User:/var/mqm:/bin/bash 与第二台机器上 /etc/passwd 中的 mqm 的 uid 和 gid 匹配，以确保这些值完全相同。如果必须更改值，请重新引导此机器。	
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。	
在将要共享的公共文件夹 /MQHA 中创建日志和数据目录。例如： 1. mkdir /MQHA 2. mkdir /MQHA/logs 3. mkdir /MQHA/qmgrs	创建用于安装共享文件系统的文件夹 /MQHA。保持路径与 Server1；例如： 1. mkdir /MQHA
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 1. chown -R mqm:mqm /MQHA 2. chmod -R ug+rwX /MQHA	
创建队列管理器： crtmqm -ld /MQHA/logs -md /MQHA/qmgrs QM1	
添加 ² /MQHA *(rw, sync, no_wdelay, fsid=0) 添加到 /etc/exports	

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。

表 30: Linux 上的说明性多实例队列管理器配置 (继续)	
Server1	Server2
启动 NFS 守护程序: <code>/etc/init.d/nfs start</code>	安装所导出的文件系统 /MQHA: <code>mount -t nfs4 -o hard,intr Server1:/ /MQHA</code>
从 Server1: <code>dspmqinf -o command QM1</code> 并将结果复制到剪贴板: <code>addmqinf -s QueueManager -v Name=QM1 -v Directory=QM1 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM1</code>	将队列管理器配置命令粘贴到 Server2: <code>addmqinf -s QueueManager -v Name=QM1 -v Directory=QM1 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM1</code>
使用 x 参数以任意顺序启动队列管理器实例: <code>strmqm -x QM1</code> 必须从与 <code>addmqinf</code> 命令相同的 IBM WebSphere 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 <code>setmqm</code> 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 。	

在 Linux 上验证多实例队列管理器

使用样本程序 `amqsgshac`, `amqspshac` 和 `amqsmshac` 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 320 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 XP Service Pack 2 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 WebSphere 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 WebSphere 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, 那么您可能会遇到访问许可权问题。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) +
CONNNAME('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 WebSphere 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 WebSphere 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:\>amqspshac 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:\>amqsgshac 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>
```

删除多实例队列管理器

要完全删除多实例队列管理器，您需要使用 **dltmqm** 命令来删除队列管理器，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。

在 UNIX 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, 或者
```

启动和停止多实例队列管理器

启动和停止配置为单个实例或多实例队列管理器的队列管理器。

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 WebSphere 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。

共享文件系统

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 WebSphere 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。

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。

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, 可重新连接的客户端将自动重新连接。您不需要在 **endmqm** 命令上包含 **-r** 标志来请求客户端重新连接。WebSphere MQ classes for Java 不支持客户端自动重新连接。

如果发现无法重新启动发生故障的实例, 即使发生了故障转移并且备用实例已变为活动状态, 请检查本地连接到发生故障的实例的应用程序是否未与发生故障的实例断开连接。本地连接的应用程序结束或断开与失败的队列管理器实例的连接, 以确保可以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。

通道和客户端重新连接

在备用队列管理器实例处于活动状态后, 通道和客户端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 WebSphere MQ 通道和客户端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, 客户端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。WebSphere MQ classes for Java 不支持客户端自动重新连接。

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。WebSphere 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端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端连接。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。

客户端连接

客户端连接可以使用连接列表或队列管理器组来选择备用连接。有关客户端重新连接到多实例队列管理器的更多信息, 请参阅第 259 页的『自动客户端重新连接』。需要编译客户端机以使用 WebSphere MQ V 7.0.1 客户端库或更高版本运行。它们必须至少连接到 V 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端必须检测连接故障, 并启动新的客户端连接。如果新客户机连接选择新活动的备用队列管理器, 那么客户端机将重新连接到同一队列管理器。

如果客户端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。

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在 WebSphere MQ MQI 客户机与队列管理器之间共享并存储在队列管理器中的上下文。WebSphere MQ MQI 客户机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 WebSphere 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接收到 MQRC_BACKED_OUT 原因码以指示事务已失败，并且需要通过发出 MQBACK 来回滚事务。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。WebSphere MQ classes for Java 不支持客户机自动重新连接。

通过 WebSphere MQ MQI 客户机应用程序，您可以设置用于自动重新连接客户机的连接选项。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 WebSphere MQ 服务器程序。

您可以通过两种方式自动执行 WebSphere 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. WebSphere MQ MQI 客户机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: 如果它通过发出 MQBACK 并重新启动工作单元, 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。其他案例记录在 [第 328 页的『自动重新连接的客户机的恢复』](#)中。

在故障转移之后, 您有三种针对 WebSphere MQ MQI 客户机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 WebSphere 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 WebSphere 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, 提供了一个示例 [第 328 页的『2』](#), 它是使用本地事务来协调 MQGET 和 MQPUT 的 WebSphere MQ MQI 客户机。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。

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 WebSphere MQ MQI 客户机的状态, 以便将客户机应用程序设计为在队列管理器故障转移之后仍然存在。

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 WebSphere 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。

WebSphere MQ MQI 客户机环境了解故障转移本身, 并在重新连接后复原尽可能多的上下文, 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 调用以复原其 WebSphere 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。错误处理程序可能与 WebSphere MQ 断开连接，然后重新连接，甚至完全停止该程序。要提高连续性，可能需要以不同的方式处理某些错误。

2. 非持久消息可能会丢失。
3. 将回滚事务。
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 WebSphere MQ MQI 客户机不知道消息是否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。
- 如果 WebSphere MQ MQI 客户机未收到来自队列管理器的响应以指示成功或失败，那么将返回 MQRC_CALL_INTERRUPTED。
 - 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该队列），那么当发生重新连接时，WebSphere MQ MQI 客户机无法重新创建临时动态队列，因为它没有要从中创建该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。

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 WebSphere 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 WebSphere 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。

- WebSphere 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 WebSphere MQ（例如 Windows、UNIX 和 Linux 系统上的 `mqs.ini` 文件）或您自己的部分配置管理（例如 `MQSC` 配置脚本）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。UNIX 和 Linux 系统上的缺省数据目录为 `/var/mqm`。在 Windows 上，缺省数据目录是 WebSphere 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 WebSphere MQ 可用性解决方案

应用程序正在使用其他 IBM WebSphere 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 WebSphere 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

仅在 z/OS 上提供的队列共享组允许一组队列管理器共享对队列的服务。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。

WebSphere MQ 客户机提高应用程序可用性

WebSphere MQ MQI 客户机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。WebSphere 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。

确保消息不丢失 (日志记录)

WebSphere MQ 记录从队列管理器故障中恢复所需的所有信息。

WebSphere MQ 在恢复日志中记录队列管理器控制的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，WebSphere MQ 还依赖于托管其文件的磁盘系统。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。

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。

WebSphere 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。

有许多日志文件包含正在记录的数据。您可以定义数字和大小 (如 [第 347 页的『更改 IBM WebSphere MQ 和队列管理器配置信息』](#) 中所述)，或者采用三个文件的系统缺省值。

在 WebSphere MQ for Windows 中，三个文件中的每个文件缺省为 1 MB。在 WebSphere MQ for UNIX and Linux 系统中，三个文件中的每个文件缺省为 4 MB。

创建队列管理器时，定义的日志文件数是分配的主日志文件数。如果未指定数字，那么将使用缺省值。

在 WebSphere 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文件：

```
C:\Program Files\IBM\WebSphere MQ\log\<QMgrName>
```

在 WebSphere MQ for UNIX and Linux 系统中，如果未更改日志路径，那么将在以下目录下创建日志文件：

```
/var/mqm/log/<QMgrName>
```

WebSphere MQ 从这些主日志文件开始，但是如果主日志空间不足，那么它会分配辅助日志文件。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文件。您可以更改此缺省分配，如 [第 347 页的『更改 IBM WebSphere MQ 和队列管理器配置信息』](#) 中所述。

日志控制文件

日志控制文件包含控制日志文件的使用所需的信息，例如它们的大小和位置以及下一个可用文件的名称。

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。

注：确保在启动队列管理器时创建的日志足够大，以容纳应用程序将处理的消息大小和数量。您可能需要更改缺省日志编号和大小以满足您的需求。有关更多信息，请参阅 [第 335 页的『计算日志大小』](#)。

日志记录类型

在 WebSphere MQ 中，日志记录所需的文件数取决于文件大小，已接收的消息数以及消息长度。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。

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。

WebSphere MQ 会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 333 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存在一系列连续的文件中。由于空间不能重复使用，所以您总是可以检索到尚未删除的任何日志扩展数据块中记录的任何记录。

由于磁盘空间有限，您可能需要考虑某种形式的归档。这是一项管理任务，用于管理日志的磁盘空间，根据需要复用或扩展现有空间。

用于线性日志记录的日志文件数可能非常大，具体取决于您的消息流和队列管理器的存在时间。但是，有许多文件被认为是活动的文件。活动文件包含重新启动队列管理器所需的日志条目。活动日志文件统称为活动日志。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 335 页的『计算日志大小』以获取有关定义数字的信息。)

控制日志文件是否被称为活动的关键事件是检查点。WebSphere MQ 检查点是恢复日志和对象文件之间的一致性点。检查点确定执行重新启动恢复所需的日志文件集。未处于活动状态的日志文件不是重新启动恢复所必需的，并且被称为不活动的日志文件。在某些情况下，介质恢复需要不活动的日志文件。(请参阅第 333 页的『使用检查点来确保完全恢复』以获取有关检查点的更多信息。)

可以归档不活动日志文件，因为重新启动恢复不需要这些文件。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。请参阅第 336 页的『管理日志』，以获取有关日志文件处置的更多信息。

如果在第二个或更高版本的主日志文件中记录了新的检查点，那么第一个文件可能会变为不活动状态，并且会格式化新的主文件并将其添加到主池的末尾，从而恢复可用于日志记录的主文件数。这样，可以看到主日志文件池是一个不断扩展的日志文件列表中的当前文件集。同样，根据您的操作需求来管理不活动文件也是一项管理任务。

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。

如果可用辅助文件数已用完，那么将拒绝需要日志活动的大多数进一步操作的请求，并将 MQRC_RESOURCE_PROBLEM 返回码返回到应用程序。

这两种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

检查点将同步队列管理器数据和日志文件，并标记可废弃日志记录的一致性点。频繁的检查点使恢复更快。

对消息队列的持久更新分两个阶段进行。首先，将表示更新的记录写入日志，然后更新队列文件。因此，日志文件可能比队列文件更新更新。为确保重新启动处理从一致点开始，WebSphere MQ 使用检查点。检查点是日志中描述的记录与队列中的记录相同的时间点。检查点本身由重新启动队列管理器所需的一系列日志记录组成；例如，检查点时处于活动状态的所有事务 (工作单元) 的状态。

WebSphere MQ 会自动生成检查点。当队列管理器启动时，关闭时，日志记录空间运行不足时，以及在记录每 10 000 个操作之后，将会执行这些操作。

当队列处理更多消息时，检查点记录将与队列的当前状态不一致。

当 WebSphere 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。检查点记录表示日志与数据之间的最新一致性点。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物的列表。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哪些操作，WebSphere MQ 会访问与正在进行的事务相关联的每个活动日志记录。这称为恢复阶段。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。将发出消息 AMQ7233 以指示解决阶段的进度。

WebSphere MQ 维护指向日志头和尾的内部指针。它将头指针移动到与恢复消息数据一致的最近检查点。

检查点用于提高恢复效率，并控制主日志文件和辅助日志文件的复用。

在第 334 页的图 69 中，WebSphere 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。

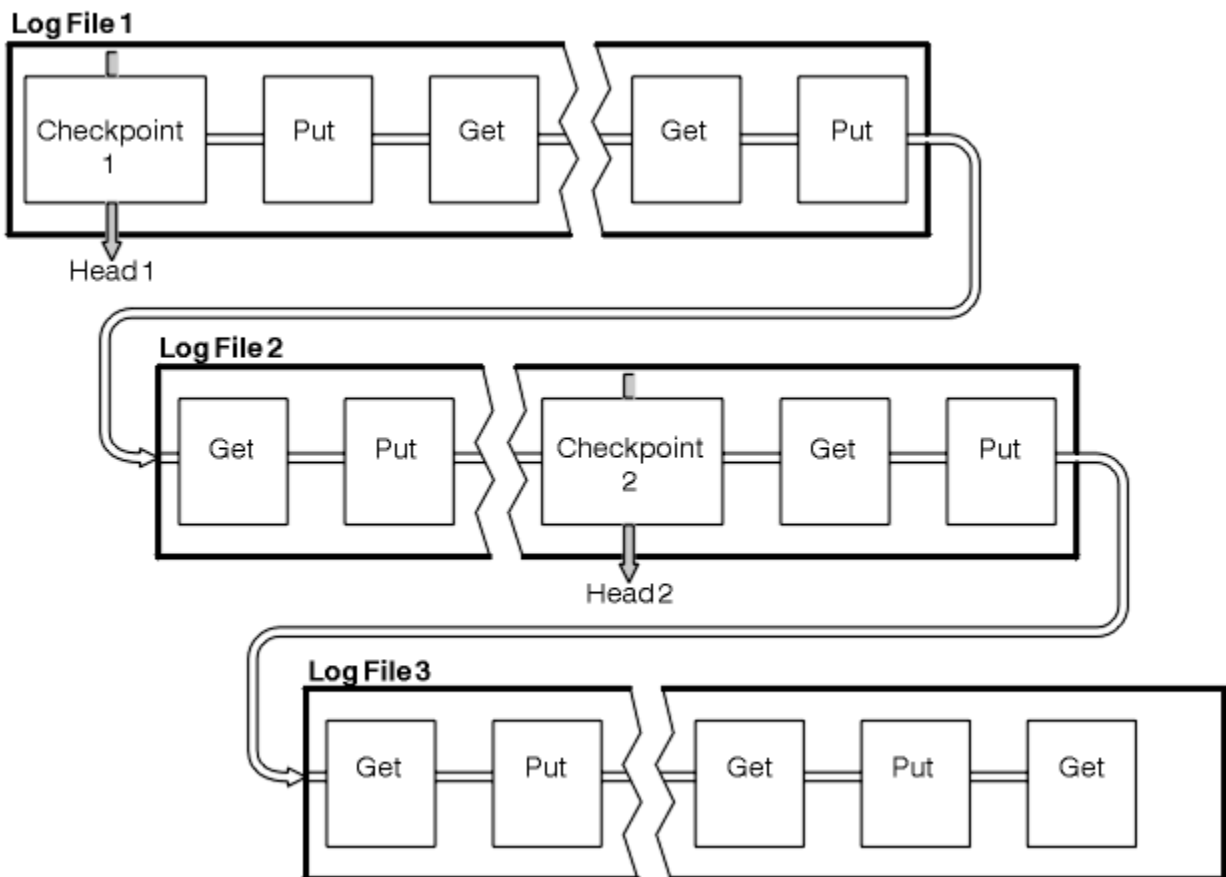


图 69: 检查点

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 335 页的图 70 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。

长时间运行的事务完成后，在 LR 2 上，日志的头将移至 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，那么将使用辅助日志文件来避免日志已满。

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。

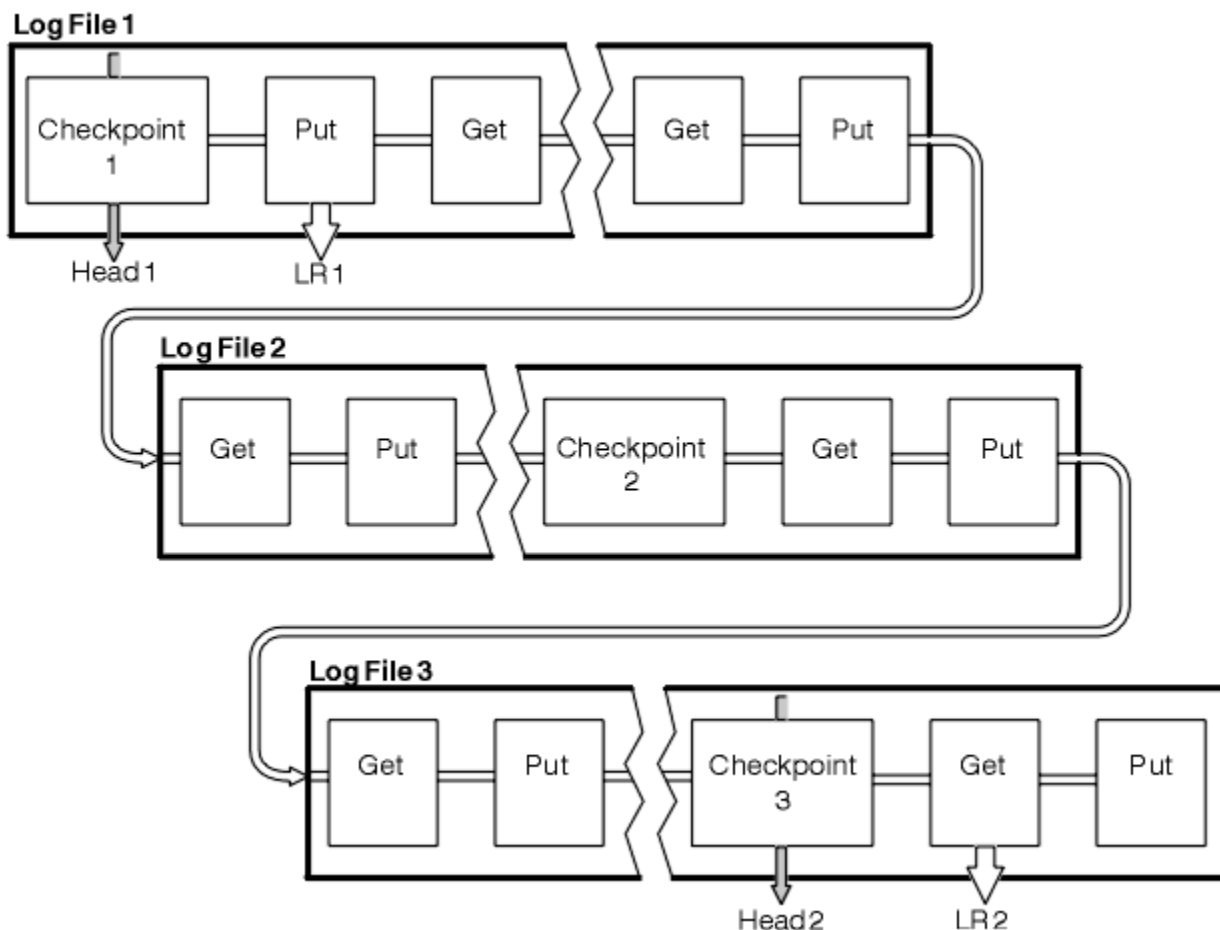


图 70: 使用长时间运行的事务进行检查点

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需要的日志大小。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

可以创建以在主日志文件已满时使用的辅助日志文件数

第 336 页的表 31 显示队列管理器针对各种操作记录的数据量。大多数队列管理器操作需要最小数量的日志空间。但是，将持久消息放入队列时，必须将 **所有** 消息数据写入日志才能恢复消息。日志大小通常取决于队列管理器需要处理的持久消息数和大小。

表 31: 日志条目大小 (所有值都是近似的)

操作	大小
放置持久消息	750 字节 + 消息长度 如果消息很大, 那么会将其划分为 261844 个字节的段, 每个段添加额外的 300 个字节。
获取消息	260 字节
同步点, 落实	750 字节
同步点, 回滚	每次获取或放入要回滚的 1000 字节 + 12 字节
创建对象	1500 字节
删除对象	300 字节
变更属性	1024 字节
记录媒体图像	800 字节 + 图像 图像分为 260 000 个字节的段, 每个段添加额外的 300 个字节。
检查点	750 字节 + 200 字节 (对于每个活动工作单元) + 380 字节 (对于每个集群发送方通道) (如果每个队列管理器使用多个集群传输队列)。 对于任何未落实的放置或由于性能原因而缓冲的获取, 可能会记录其他数据。 如果您有集群发送方通道, 那么在每个检查点, 每个集群发送方通道会向日志写入额外的 380 字节。

注:

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。
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 UNIX 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。这意味着队列管理器的日志可能小于您估计队列管理器需要记录的数据量。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。
6. 主日志文件可供在检查点期间复用。由于日志空间量较低, 因此在获取检查点之前, 队列管理器会同时考虑主日志空间和辅助日志空间。

如果未定义比辅助日志文件更多的主日志文件, 那么队列管理器可能会在执行检查点之前分配辅助日志文件。这将使主日志文件可供复用。

管理日志

日志几乎是自我管理, 但有时需要进行管理以解决空间问题。

随着时间的推移, 写入的某些日志记录将不需要重新启动队列管理器。如果您正在使用循环日志记录, 那么队列管理器会回收日志文件中的已释放空间。对于用户而言, 此活动并不明显, 您通常不会看到已使用的磁盘空间量减少, 因为已分配的空间会快速复用。

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。因此, 如果长时间未使用检查点, 或者如果长时间运行的事务在很久之前写入日志记录, 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。

当长时间运行的事务填充日志时，尝试写入日志记录失败，并且某些 MQI 调用返回 MQRC_RESOURCE_PROBLEM。（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。）

队列管理器回滚占用过多日志空间的事务。以这种方式回滚具有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，将释放其日志空间，并且队列管理器将继续正常运行。

如果日志已满，那么将发出消息 AMQ7463。此外，如果由于长时间运行的事务已阻止释放空间而导致日志填充，那么将发出消息 AMQ7465。

最后，如果记录写入日志的速度超过日志可以处理的速度，那么将发出消息 AMQ7466。如果看到此消息，请增加日志文件数或减少队列管理器正在处理的数据量。

磁盘变满时发生的情况

队列管理器日志记录组件可以处理完整磁盘和完整日志文件。如果包含日志的磁盘已满，那么队列管理器将发出消息 AMQ6708 并生成错误记录。

将以最大大小创建日志文件，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 WebSphere MQ 只能在创建新文件时耗尽磁盘空间；在将记录写入日志时，不能耗尽空间。WebSphere MQ 始终知道现有日志文件中的可用空间量，并相应地管理这些文件中的空间。

如果填充包含日志文件的驱动器，那么您可能可以释放一些磁盘空间。如果您正在使用线性日志，那么日志目录中可能有一些不活动的日志文件，您可以将这些文件复制到另一个驱动器或设备。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。队列管理器假定所有日志文件大小相同。

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。

如果要使用循环日志记录，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 355 页的『IBM WebSphere MQ 的日志缺省值』和第 361 页的『队列管理器日志』）。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。

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的推移而增加。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。以下是确定仍需要哪些日志文件的方法：

记录器事件消息

启用时，当队列管理器开始将日志记录写入新的日志文件时，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 [脚本 \(MQSC\) 命令](#)，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 AMQ7467 给出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 AMQ7468 提供了介质恢复所需的最旧日志文件的名称。

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。

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。

如果找不到需要的任何日志文件，那么将发出操作员消息 AMQ6767。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。

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。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：

在 Windows 上

```
rcdmqimg -m QMNAME -t all *
```

在 UNIX 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 340 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**。

注: 在运行 **rcdmqimg** 命令时，还可以发出消息 AMQ7467 和 AMQ7468。

确定多余的日志文件

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。

要确定重新启动队列管理器所需的最早日志文件，请发出命令 **DISPLAY QMSTATUS RECLOG**。

要确定执行介质恢复所需的最早日志文件，请发出命令 **DISPLAY QMSTATUS MEDIALOG**。

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 (10 年每天 3000 个日志文件的顺序)，否则您不需要满足 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。

但是，如果您的日志文件更替率非常高，或者希望有信心应对一般情况，那么通常可以使用以下算法：

Let S == 重新启动日志文件号

(来自 **DISPLAY QMSTATUS RECLOG**)。

Let M == 介质恢复日志文件号

(来自 **DISPLAY QMSTATUS MEDIALOG**)。

让 L == 具有删除或归档资格的日志文件号

需要确定。

```
函数 minlog (a , b) {  
  if (abs (a - b) < 5000000)  
    返回 min (a , b); # 未换行。  
  else  
    返回最大 (a , b); # 已包装。}
```

在以下情况下可以删除日志文件 L:

(L != S && L != M && minlog (L, minlog (S, M)) == L).

在以下情况下，可以归档日志文件 L:

(L != S && minlog (L, S) == L).

日志文件位置

为日志文件选择位置时，请记住，如果 WebSphere 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。

理想情况下，将日志文件放在与队列管理器数据不同的磁盘驱动器上。这在性能方面有好处。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 WebSphere MQ 系统的最后一个备份。

使用日志进行恢复

使用日志从故障中恢复。

有几种方法会损坏您的数据。WebSphere 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

WebSphere MQ 可以从通信故障和断电中恢复。此外，它有时还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，消息将保留在队列上，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，WebSphere 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 WebSphere 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 WebSphere 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

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；它是该对象的最新介质映像的开头。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 `rcdmqing` 命令手动记录介质映像，如 `rcdmqing` 中所述。此命令将写入 IBM WebSphere MQ 对象的介质映像。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 WebSphere 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 WebSphere 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。

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 WebSphere 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 WebSphere 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，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)。

保护 IBM WebSphere 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 WebSphere 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 WebSphere 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 WebSphere MQ 可执行文件)。因此，它们可能会以 IBM WebSphere 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` 命令来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。必须指定从先前 `dmpmqlog` 输出获取的有效 *LSN* 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。

- 仅对于线性日志，您可以指示 `dmpmqlog` 从任何给定的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` 期望在与活动日志文件相同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：

<i>logactive</i>	主日志扩展数据块数。
<i>loginactive</i>	辅助日志扩展数据块的数量。
日志大小	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 <i>LSN</i> 。
<i>nextlsn</i>	要写入的下一个日志记录的 <i>LSN</i> 。
标题	日志头的日志记录的 <i>LSN</i> 。
尾数	标识日志尾部位置的 <i>LSN</i> 。
<i>hflag1</i>	日志是循环还是 LOG RETAIN (线性)。
<i>HeadExtent</i> 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：

<i>LSN</i>	日志序号。
<i>LogRecd</i> 类型	日志记录的类型。
<i>XTranid</i>	与此日志记录关联的事务标识 (如果有)。 MQI 的 <i>TranType</i> 指示仅 WebSphere MQ 事务。其他资源管理器涉及 XA 的 <i>TranType</i> 。同一工作单元中涉及的更新具有相同的 <i>XTranid</i> 。
<i>QueueName</i>	与此日志记录关联的队列 (如果有)。
<i>Qid</i>	队列的唯一内部标识。
<i>PrevLSN</i>	同一事务 (如果有) 中先前日志记录的 <i>LSN</i> 。

启动队列管理器

这将记录队列管理器已启动的日志。

<i>StartDate</i>	队列管理器的启动日期。
<i>StartTime</i>	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。

<i>StopDate</i>	队列管理器停止的日期。
<i>StopTime</i>	队列管理器停止的时间。
<i>ForceFlag</i>	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。

<i>ChkPtLSN</i>	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：

<i>MapIndex</i>	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 <i>QueueName</i> 和 <i>MapIndex</i> 的后续 <i>Get Message</i> 日志记录。此时，可以复用 <i>MapIndex</i> 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) 和消息数据本身。

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：

<i>MapIndex</i>	标识从队列中检索的消息。包含相同 <i>QueueName</i> 和 <i>MapIndex</i> 的最新 <i>Put Message</i> 日志记录标识检索的消息。
<i>QPriority</i>	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 WebSphere 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：

<i>XTranid</i>	事务标识。
<i>FirstLSN</i>	与事务关联的第一个日志记录的 LSN。
<i>LastLSN</i>	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：

<i>RMName</i>	资源管理器的名称。
<i>RMID</i>	资源管理器标识。这还会记录在后续 <i>Transaction Prepared</i> 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。
<i>SwitchFile</i>	此资源管理器的交换机装入文件。
<i>XAOpenString</i>	此资源管理器的 XA 打开字符串。
<i>XACloseString</i>	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 WebSphere MQ 对象的创建。

<i>ObjName</i>	创建的对象名称。
<i>UserId</i>	执行创建的用户标识。

删除对象

这将记录 WebSphere MQ 对象的删除。

<i>ObjName</i>	已删除的对象名称。
----------------	-----------

备份和复原 IBM WebSphere MQ 队列管理器数据

正在备份队列管理器和队列管理器数据。

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。

要备份和复原队列管理器数据，请参阅：

- [第 344 页的『备份队列管理器数据』](#)。
- [第 345 页的『复原队列管理器数据』](#)。

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，其中包含来自不可恢复队列管理器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。

要创建并激活备份队列管理器，请参阅：

- [第 346 页的『创建备份队列管理器』](#)。
- [第 346 页的『启动备份队列管理器』](#)。

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。如果由于硬件故障而丢失了队列管理器配置和日志数据，那么队列管理器将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，那么您将能够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。

要备份和复原队列管理器配置，请参阅：

- [第 347 页的『备份队列管理器配置』](#)
- [第 347 页的『复原队列管理器配置』](#)

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

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，仅当该命令失败时，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：

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 347 页的『更改 IBM WebSphere MQ 和队列管理器配置信息』](#)。

注：您可能难以理解目录中显示的名称。这些名称将进行变换，以确保它们与您使用 WebSphere MQ 的平台兼容。有关名称变换的更多信息，请参阅[了解 WebSphere 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 332 页的『日志的外观』](#) 中所述) 和配置文件 (如 [第 59 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。

3. 保留文件的所有权。对于 WebSphere MQ for UNIX and Linux 系统，可以使用 `tar` 命令来执行此操作。(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 WebSphere MQ V 7.5 及更高版本时，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 WebSphere MQ。

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

确保队列管理器未在运行。

关于此任务

要复原队列管理器数据的备份，请执行以下操作：

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。
4. 更新配置信息文件。

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 WebSphere 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 [UNIX and Linux 系统上的目录内容](#)。

确保您有日志控制文件以及日志文件。还要检查 IBM WebSphere MQ 和队列管理器配置文件是否一致，以便 WebSphere MQ 可以在正确的位置查找复原的数据。

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。

注：升级到 WebSphere MQ V 7.5 及更高版本时，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 WebSphere MQ。

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。

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。

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，您才应该复制日志文件。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。一旦激活了备份队列管理器，就无法再对其进行更新。

有关如何创建，更新和启动备份队列管理器的信息，请参阅以下主题：

- [第 346 页的『创建备份队列管理器』](#)
- [第 346 页的『更新备份队列管理器』](#)
- [第 346 页的『启动备份队列管理器』](#)

创建备份队列管理器

使用线性日志记录时，只能使用备份队列管理器。

要为现有队列管理器创建备份队列管理器，请执行以下操作：

1. 使用控制命令 `strmqm` 为现有队列管理器创建备份队列管理器。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。
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 344 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执行以下控制命令：

```
strmqm -r BackupQMName
```

这会将队列管理器标记为 WebSphere 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。

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。

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。无需停止要备份的队列管理器。

要更新备份队列管理器，请执行以下操作：

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，直至 (但不包括) 步骤 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发出以下控制命令：

```
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。

警告：如果将 `non-contiguous` 日志集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少日志的位置的日志。

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。

要进行此操作，请执行以下步骤：

1. 执行以下控制命令以激活备份队列管理器：


```
stimqm -a BackupQMName
```

已激活备份队列管理器。现在处于活动状态，无法再更新备份队列管理器。

2. 执行以下控制命令以启动备份队列管理器:

```
stimqm BackupQMName
```

WebSphere MQ 将此视为重新启动恢复，并利用备份队列管理器中的日志。在上次更新备份队列管理器重放期间，将仅回滚上次记录的检查点中的活动事务。

当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。

3. 重新启动所有通道。

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 WebSphere MQ 目录和子目录的更多信息，请参阅 [规划文件系统支持](#)。

确保您有日志控制文件以及日志文件。还要检查 WebSphere MQ 和队列管理器配置文件是否一致，以便 WebSphere MQ 可以在正确的位置查找复原的数据。

如果正确备份和复原了数据，那么队列管理器现在将启动。

注: 即使队列管理器数据和日志文件保存在不同的目录中，也会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。如果它确实启动，那么您的数据很可能已损坏。

备份队列管理器配置

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。

要获取队列管理器配置的备份副本，请执行以下操作:

1. 确保队列管理器正在运行。
2. a. 在 AIX, HP-UX, Linux, Solaris 或 Windows 上: 使用缺省格式化选项 (-f mqsc) MQSC 和所有属性 (-a) 执行转储 MQ 配置命令 (dmpmqcfg)，使用标准输出重定向将定义存储到文件中，例如:

```
dmpmqcfg -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
```

复原队列管理器配置

执行以下步骤以复原队列管理器配置的备份。

要复原队列管理器配置的备份，请执行以下操作:

1. 确保队列管理器正在运行。请注意，如果通过其他方法无法恢复对数据和日志的损坏，那么可能已重新创建队列管理器。
2. 根据您的平台，执行下列其中一个命令:
 - a. 在 AIX, HP-UX, Linux, Solaris 或 Windows 上: 对队列管理器执行 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，例如:

```
runmqsc MYQMGR < /mq/backups/MYQMGR.mqsc
```

相关参考

[dmpmqcfg](#)

更改 IBM WebSphere MQ 和队列管理器配置信息

更改 IBM WebSphere MQ 或单个队列管理器的行为以满足安装需要。

您可以通过更改一组用于管理 IBM WebSphere MQ 的配置属性 (或参数) 上指定的值来更改 IBM WebSphere MQ 配置信息。

通过编辑 IBM WebSphere MQ 配置文件来更改属性信息。在 IBM WebSphere MQ for Windows 和 Linux (x86 和 x86-64 平台) 上, 可以使用 IBM WebSphere MQ Explorer 来编辑 IBM WebSphere MQ 配置文件。

在 Windows 系统上, 您还可以使用 amqmdain 来更改配置信息, 如 [amqmdain](#) 中所述

要了解有关为平台配置 IBM WebSphere MQ 和队列管理器的更多信息, 请参阅以下子主题:

相关概念

[第 5 页的『配置』](#)

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。

相关任务

[规划](#)

[管理 WebSphere MQ](#)

更改 UNIX, Linux, and Windows 系统上的配置信息

配置属性保存在配置文件中的节点级别和队列管理器级别。

在 Windows UNIX and Linux 平台上, 您可以在以下范围内更改 IBM WebSphere MQ 配置属性:

- IBM WebSphere MQ 配置文件 (**mqs.ini**), 用于在整个节点上影响 IBM WebSphere MQ 的更改。每个节点都有一个 mqs.ini 文件。
- 队列管理器配置文件 (**qm.ini**), 用于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。

客户机配置选项单独保存在客户机配置文件中。

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) 的行组。

由于 IBM WebSphere 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 WebSphere MQ 配置文件中定义的队列管理器。

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。

在 Windows 和 Linux (x86 和 x86-64 平台) 系统上, 可以从 IBM WebSphere MQ Explorer 编辑配置信息。

在 Windows 系统上, 您还可以使用 amqmdain 命令来编辑配置文件。

有关 Windows UNIX and Linux 系统上的配置选项的更多信息, 请参阅以下子主题:

相关概念

[第 5 页的『配置』](#)

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。

[第 347 页的『更改 IBM WebSphere MQ 和队列管理器配置信息』](#)

更改 IBM WebSphere MQ 或单个队列管理器的行为以满足安装需要。

相关任务

[规划](#)

[管理 WebSphere MQ](#)

相关参考

[第 353 页的『用于更改 IBM WebSphere MQ 配置信息的属性』](#)

在 IBM WebSphere MQ for Windows 系统和 IBM WebSphere MQ for Linux (x86 和 x86-64 平台) 系统上, 使用 IBM WebSphere MQ Explorer 修改配置信息。在其他系统上, 通过编辑 mqs.ini 配置文件来修改信息。

[第 358 页的『更改队列管理器配置信息』](#)

此处描述的属性用于修改单个队列管理器的配置。它们覆盖 WebSphere MQ 的任何设置。

编辑配置文件

使用命令或标准文本编辑器编辑配置文件。

在编辑配置文件之前首先进行备份，这样您就可以有了一个副本，可在需要进行恢复。

您可以用以下方法之一编辑配置文件：

- 自动，在节点上使用更改队列管理器配置的命令
- 手动，使用标准文本编辑器

您可以在安装后编辑 WebSphere MQ 配置文件中的缺省值。

如果您在配置文件属性上设置了不正确的值，此值将被忽略，并会发出操作员消息以指出问题。（这个效果和完全缺少属性是一样的。）

创建新的队列管理器时：

- 备份 WebSphere MQ 配置文件
- 备份新的队列管理器配置文件

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，并且该字符将用作配置数据的一部分。

何时需要编辑配置文件？

编辑配置文件以从备份恢复，移动队列管理器，更改缺省队列管理器或帮助 IBM 支持。

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 支持中心建议您执行此操作。

配置文件优先级

属性的值是在多个位置定义的。命令中设置的属性优先于配置文件中的属性。

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

IBM WebSphere MQ 配置文件 mqs.ini

IBM WebSphere 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。

在 IBM WebSphere MQ for UNIX and Linux 产品上，数据目录和日志目录始终分别为 /var/mqm 和 /var/mqm/log。

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。

此外，在 Windows 系统上，安装配置信息（包含在 IBM WebSphere MQ for UNIX and Linux 系统上的 mqinst.ini 中）位于注册表中，因为 Windows 上没有 mqinst.ini 文件。

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

- 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

为新 IBM WebSphere 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。对于 [第 350 页的图 71](#) 中的 LogDefaults 节，将显示缺省值。LogBufferPages 属性的值为零表示 512。

如果需要非缺省值，那么必须在 LogDefaults 节中显式指定该值。

```
#####  
#* Module Name: mqs.ini                               *#  
#* Type       : WebSphere MQ Machine-wide Configuration File   *#  
#* Function   : Define WebSphere 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
```

图 71: UNIX 系统的 IBM WebSphere MQ 配置文件示例

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。

V 7.5.0.9 从 IBM WebSphere MQ Version 7.5.0 修订包 9 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, 这使您更容易查看错误内容, 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



在 UNIX and Linux 系统上, **qm.ini** 文件保存在队列管理器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
```

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, 队列管理器 **QMNAME** 的配置文件的 **路径和名称**是:

```
C:\Program Files\IBM\WebSphere MQ\qmgrs\QMNAME\qm.ini
```

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 [了解 WebSphere MQ 文件名](#)。

示例 qm.ini 文件



以下示例显示了如何在 IBM WebSphere MQ 中针对 UNIX and Linux 系统的队列管理器配置文件中安排属性组。

```
## Module Name: qm.ini                ##
## Type       : WebSphere 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  ##
##          ##
##*****#

ExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64

Service:
  Name=AuthorizationService
  EntryPoints=13

ServiceComponent:
  Service=AuthorizationService
  Name=MQSeries.UNIX.auth.service
  Module=opt/mqm/bin/amqzfu
  ComponentDataSize=0

Log:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=01
  LogPath=/var/mqm/log/saturn!queue!manager/

AccessMode:
  SecurityGroup=wmq\wmq

XAResourceManager:
  Name=DB2 Resource Manager Bank
  SwitchFile=/usr/bin/db2swit
  XAOpenString=MQBankDB
  XACloseString=
  ThreadOfControl=THREAD
```

```
Channels: 2
  MaxChannels=200
  MaxActiveChannels=100
  MQIBindType=STANDARD

AccessMode:
  SecurityGroup=wmq\wmq
TCP:
  KeepAlive = Yes
  SvrSndBuffSize=32768
  SvrRcvBuffSize=32768
  Connect_Timeout=0

QMErrorLog:
  ErrorLogSize=262144
  ExcludeMessage=7234
  SuppressMessage=9001,9002,9202
  SuppressInterval=30

ApiExitLocal:
  Name=ClientApplicationAPIchecker
  Sequence=3
  Function=EntryPoint
  Module=/usr/Dev/ClientAppChecker
  Data=9.20.176.20
```

注意:

1. LogBufferPages 的值为零表示值为 512。
2. 有关 Channel 节的更多信息，请参阅第 59 页的『初始化和配置文件』。
3. 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。

WebSphere MQ 在 Unix 中使用文件扩展名为 .ini 的配置文件，例如 qm.ini。WebSphereMQ 中有一些实用程序 (例如 **setmqm**) 将生成文件的临时备份副本。例如，文件 qm.ini 将创建名为 qm.ini.bak 的备份副本。实用程序将修改 qm.ini 文件，存储更新后的文件，然后删除 qm.ini.bak 文件。如果实用程序无法存储 qm.ini 文件，那么它将从备份文件 qm.ini.bak 复原 qm.ini 的内容，然后删除 qm.ini.bak 文件。

如果存在现有的 qm.ini.bak 文件，那么实用程序会将 qm.ini 文件还原为 qm.ini.bak 的内容，并删除 qm.ini.bak 文件。因此，不应使用文件扩展名 .bak 创建 *.ini 文件的备份副本，因为 WebSphere MQ 实用程序可能会删除这些备份文件。

请参阅第 348 页的『更改 UNIX, Linux, and Windows 系统上的配置信息』，以获取有关更改何时生效的信息。

安装配置文件，mqinst.ini

UNIX and Linux 系统

安装配置文件 mqinst.ini 包含有关 UNIX 或 Linux 系统上所有 IBM WebSphere MQ 安装的信息。

mqinst.ini 文件位于 UNIX 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

不能直接编辑或引用此文件，因为其格式不固定，并且可能会更改。而是使用以下命令来创建，删除，查询和修改 mqinst.ini 文件中的值:

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。

安装标识 (仅供内部使用) 是自动设置的, 不得更改。

Windows 系统

安装配置信息保存在 Windows 系统上的以下密钥中:

```
HKLM\SOFTWARE\IBM\WebSphere MQ\Installation\
```

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。请改为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。

用于更改 IBM WebSphere MQ 配置信息的属性

在 IBM WebSphere MQ for Windows 系统和 IBM WebSphere MQ for Linux (x86 和 x86-64 平台) 系统上, 使用 IBM WebSphere MQ Explorer 修改配置信息。在其他系统上, 通过编辑 mqs.ini 配置文件来修改信息。

请参阅以下子主题以了解特定组件的属性:

相关概念

[第 5 页的『配置』](#)

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。

[第 347 页的『更改 IBM WebSphere MQ 和队列管理器配置信息』](#)

更改 IBM WebSphere MQ 或单个队列管理器的行为以满足安装需要。

相关任务

[规划](#)

[管理 WebSphere MQ](#)

相关参考

[第 358 页的『更改队列管理器配置信息』](#)

此处描述的属性用于修改单个队列管理器的配置。它们覆盖 WebSphere MQ 的任何设置。

所有队列管理器

使用 IBM WebSphere MQ Explorer 中的 General 和 Extended WebSphere MQ 属性页面或 mqs.ini 文件中的 AllQueueManagers 节来指定有关所有队列管理器的以下信息。

DefaultPrefix=目录名称

此属性指定保留队列管理器数据的 qmgrs 目录的路径。

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 WebSphere MQ, 并仅在将结构移至新位置并更改缺省前缀后重新启动 WebSphere MQ。

注: 请勿删除 UNIX 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。

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 属性来指定 WebSphere MQ 如何将 EBCDIC NL 字符转换为 ASCII 格式。

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。

表

根据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 354 页的表 32 中显示了可能的 ISO CCSID。

表 32: 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。

缺省队列管理器

使用 IBM WebSphere MQ Explorer 中的 General WebSphere MQ 属性页面或 mqs.ini 文件中的 DefaultQueueManager 节来指定缺省队列管理器。

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。

出口属性

使用 IBM WebSphere MQ Explorer 中的 Extended IBM WebSphere MQ 属性页面或 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。

CLWLMODE=SAFE|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, 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。

SAFE

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 (amqzlw0) 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。

IBM WebSphere MQ 的日志缺省值

使用 IBM WebSphere MQ Explorer 中的 Default log settings IBM WebSphere MQ 属性页面或 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。

如果该节不存在，那么将使用 MQ 缺省值。在创建队列管理器时，日志属性将用作缺省值，但如果在 `crtmqm` 命令中指定日志属性，那么可以覆盖这些属性。请参阅 [crtmqm](#) 以获取此命令的详细信息。

创建队列管理器后，将从 [第 361 页](#) 的『队列管理器日志』中描述的设置中获取该队列管理器的日志属性。

缺省前缀(在 [第 353 页](#) 的『所有队列管理器』中指定)和为特定队列管理器(在 [第 361 页](#) 的『队列管理器日志』中指定)指定的日志路径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 335 页](#) 的『计算日志大小』。

注: 以下参数列表中给出的限制是由 WebSphere 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|2-254 (Windows) |2-510 (UNIX 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 UNIX 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 UNIX 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|1-253 (Windows) |1-509 (UNIX 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 UNIX 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 UNIX 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。

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 UNIX 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| LINEAR

要使用的日志的类型。缺省值为循环。

CIRCULAR

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 333 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 333 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。创建队列管理器后, 无法更改记录方法。

LogBufferPages=0|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。在 WebSphere MQ V 7.0 中, 这是 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:

- <DefaultPrefix>\log for WebSphere MQ for Windows, 其中 <DefaultPrefix> 是在 All Queue Managers WebSphere MQ 属性页面上的 DefaultPrefix 属性上指定的值。此值是在安装时设置的。
- /var/mqm/log for WebSphere MQ for UNIX and Linux 系统

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。

LogWrite 完整性 =SingleWrite|DoubleWrite| TripleWrite

记录器用于可靠地写入日志记录的方法。

TripleWrite

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

仅当文件系统或托管 WebSphere MQ 恢复日志的设备明确保证 4 KB 原子性写入时, 您才能使用 **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。

高级配置和电源接口 (Advanced Configuration and Power Interface , ACPI)

使用 IBM WebSphere MQ Explorer 中的 ACPI WebSphere MQ 属性页面来指定当系统接收到暂挂请求时 WebSphere MQ 的行为方式。

Windows 支持高级配置和电源接口 (ACPI) 标准。这使启用了 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。

请注意, 仅当 "警报监视器" 正在运行时, 才会应用 ACPI WebSphere 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	如果没有任何通道正在运行, 请接受暂挂请求; 如果这些通道正在显示对话框 (请参阅 注意 ; 接受请求)。
Y	Y	N	显示对话框 (注意 ; 拒绝暂挂请求)。
Y	Y	Y	如果没有通道正在运行, 请接受暂挂请求; 如果这些通道正在显示对话框 (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。

API 出口

使用 IBM WebSphere MQ Explorer 或 amqmdain 命令来更改 API 出口的条目。

使用 IBM WebSphere MQ Explorer 中的 Exits IBM WebSphere MQ 属性页面或 mqs.ini 文件中的 ApiExitTemplate 和 ApiExitCommon 节来标识所有队列管理器的 API 出口例程。在 Windows 系统上, 您还可以使用 amqmdain 命令来更改 API 出口的条目。(要识别各个队列管理器的 API 出口例程, 请使用 ApiExitLocal 节, 如 [第 369 页](#)的『API 出口』中所述。)

有关这些节的属性的完整描述, 请参阅 [配置 API 出口](#)。

队列管理器

每个队列管理器都有一个 QueueManager 节。使用此节来指定队列管理器目录的位置。

在 Windows UNIX and Linux 系统上, 每个队列管理器都有一个 QueueManager 节。这些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 WebSphere MQ 文件名](#)。

名称 =*queue_manager_name*

队列管理器的名称。

前缀 =前缀

存储队列管理器文件的位置。缺省情况下，此值与所有队列管理器信息的 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，将覆盖前缀和目录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 WebSphere 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命令。如果不存在 InstallationName 值，那么队列管理器将与低于 V 7.1 的 WebSphere MQ 安装相关联。

相关概念

第 14 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 UNIX, 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。

安全性

使用 **qm.ini** 文件中的 Security 节来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。

GroupModel=GlobalGroups

此属性确定在 Windows 上确定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 ClusterQueueAccessControl=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 ClusterQueueAccessControl=Xmitq 或将定制授权服务升级到 MQZAS_VERSION_6 或更高版本。

更改队列管理器配置信息

此处描述的属性用于修改单个队列管理器的配置。它们覆盖 WebSphere MQ 的任何设置。

在 UNIX and Linux 系统上，通过编辑 **qm.ini** 配置文件来修改队列管理器配置信息。在 **qm.ini** 中定义节时，无需在新行上启动每个项。可以使用分号 (;) 或散列字符 (#) 来指示注释。

在 Windows 和 Linux (x86 和 x86-64 平台) 系统上，可以使用 IBM WebSphere MQ Explorer 来修改某些配置信息。但是，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 WebSphere MQ

Explorer 中处于只读状态。因此，您必须通过在 Windows 上使用 **regedit** 并通过在 UNIX and Linux 上编辑 `qm.ini` 文件来对可安装服务进行任何更改。

有关更改队列管理器配置信息的更多详细信息，请参阅以下子主题：

相关概念

[第 5 页的『配置』](#)

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。

[第 347 页的『更改 IBM WebSphere MQ 和队列管理器配置信息』](#)

更改 IBM WebSphere MQ 或单个队列管理器的行为以满足安装需要。

相关任务

[规划](#)

[管理 WebSphere MQ](#)

相关参考

[第 353 页的『用于更改 IBM WebSphere MQ 配置信息的属性』](#)

在 IBM WebSphere MQ for Windows 系统和 IBM WebSphere MQ for Linux (x86 和 x86-64 平台) 系统上，使用 IBM WebSphere MQ Explorer 修改配置信息。在其他系统上，通过编辑 `mqs.ini` 配置文件来修改信息。

访问方式

Access Mode 仅适用于 Windows 服务器。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

在使用 `-a [r]` 选项运行 `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。如果您将队列管理器作为服务启动，那么 Windows 系统检查的用户标识是您使用 **准备 IBM WebSphere MQ** 向导配置的用户标识。如果是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 `stmqm` 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。

相关概念

[第 313 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 310 页的『在 Windows 上保护共享队列管理器数据和日志目录及文件』](#)

相关任务

[第 290 页的『在域工作站或服务器上创建多实例队列管理器』](#)

相关参考

[克特姆](#)

可安装服务

您可以使用 **regedit** 在 Windows 上更改可安装服务，并使用 `qm.ini` 文件中的 `Service` 节在 UNIX and Linux 上更改可安装服务。

注: 更改可安装服务及其组件有重大影响。因此，可安装服务在 WebSphere MQ Explorer 中是只读的。

要在 Windows 系统上更改可安装服务，请使用 **regedit**，或者在 UNIX and Linux 系统上使用 `qm.ini` 文件中的 `Service` 节。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。在 UNIX and Linux 系统上，请使用 `ServiceComponent` 节来执行此操作。

名称=**AuthorizationService|NameService**

必需服务的名称。

AuthorizationService

对于 WebSphere 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。

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。

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。

SecurityPolicy=**Default|NTSIDsRequired** (仅限 WebSphere MQ for Windows)

仅当指定的服务是缺省授权服务 (即 OAM) 时，`SecurityPolicy` 属性才适用。`SecurityPolicy` 属性允许您指定每个队列管理器的安全策略。可能的值为：

Default

使用缺省安全策略以使其生效。如果未将特定用户标识的 Windows 安全标识 (NT SID) 传递到 OAM，那么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。

请参阅 [Windows 安全标识 \(SID\)](#) 以获取更多信息。

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：

Default

`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` 属性仅用于与快速路径绑定的关系。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：

Default

`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:

Default

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 [适用于 UNIX, Linux 和 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 Windows UNIX and Linux 系统上设置安全性](#)。

相关参考

[可安装服务参考信息](#)

服务组件

添加新的可安装服务时, 必须指定服务组件信息。在 Windows 系统上, 使用 `regedit`, 在 UNIX and Linux 系统上, 使用 `qm.ini` 文件中的 `ServiceComponent` 节。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。

按如下所示指定服务组件:

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 WebSphere MQ 对象的名称有效的字符 (例如, 队列名称)。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。

ComponentData 大小 = 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。

有关可安装服务和组件的更多信息, 请参阅 [适用于 UNIX, Linux 和 Windows 的可安装服务和组件](#)。

队列管理器日志

使用 IBM WebSphere MQ Explorer 中的 Log 队列管理器属性页面或 `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。

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 355 页的『IBM WebSphere MQ 的日志缺省值』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 335 页的『计算日志大小』。

注: 以下参数列表中给出的限制由 WebSphere MQ 设置。操作系统限制可能会减小可能的最大日志大小。

LogPrimary 文件 = 3 | 2-254 (Windows) | 2-510 (UNIX 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 UNIX 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 UNIX 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。

LogSecondary 文件 =2 |1-253 (Windows) |1-509 (UNIX 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 UNIX 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 UNIX 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。

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 UNIX 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 | LINEAR

队列管理器要使用的日志记录类型。创建队列管理器后，无法更改要使用的日志记录类型。请参阅第 355 页的『IBM WebSphere MQ 的日志缺省值』中 LogType 属性的描述，以获取有关使用所需日志记录类型创建队列管理器的信息。

CIRCULAR

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 333 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 333 页的『日志记录类型』，以获取更全面的线性日志记录说明。

LogBuffer 页面数 =0 |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。在 WebSphere MQ V 7.0 中，这是 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。

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=directory_name

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：

- WebSphere MQ for Windows 中的 C:\Program Files\IBM\WebSphere MQ\log。
- WebSphere MQ for UNIX 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 WebSphere MQ for UNIX and 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。

LogWrite 完整性 =SingleWrite|DoubleWrite| TripleWrite

记录器用于可靠地写入日志记录的方法。

TripleWrite

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

仅当文件系统或托管 WebSphere MQ 恢复日志的设备明确保证 4 KB 原子性写入时，您才能使用 **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。

受限方式

此选项仅适用于 UNIX and Linux 系统。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。

在某些目录下，IBM WebSphere 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 WebSphere MQ 应用程序的所有用户都必须位于应用程序组中。

由不在应用程序组中的用户发出的任何 `MQCONN` 或 `MQCONNX` 调用都失败，原因码为 `MQRC_Q_MGR_NOT_AVAILABLE`。

受限方式使用 IBM WebSphere MQ 授权服务进行操作。因此，您还必须授予用户连接到 IBM WebSphere MQ 的权限，并使用 IBM WebSphere MQ 授权服务访问用户所需的资源。

 可以在 [在 Windows UNIX and Linux 系统上设置安全性](#) 中找到有关配置 IBM WebSphere 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 WebSphere MQ 受限方式。

XA 资源管理器

使用 IBM WebSphere MQ Explorer 中的 XA resource manager 队列管理器属性页面或 qm.ini 文件中的 XAResourceManager 节来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以下信息。

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。

WebSphere 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=name (必需)

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 指定的路径装入到应用程序中。

64 位文件将从 ExitsDefaultPath64 指定的路径装入到队列管理器中。

XAOpenString=string (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=string (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

对于 WebSphere MQ for Windows，此属性是必需的。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 WebSphere 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 WebSphere 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。

通道节的属性

这些属性确定通道的配置。

此信息不适用于 z/OS 平台的 WebSphere MQ。

使用 WebSphere MQ Explorer 中的 Channels 队列管理器属性页面或 qm.ini 文件中的 CHANNELS 节来指定有关通道的信息。

MaxChannels=100 |编号

允许的当前通道的最大数量。

该值必须在 1-65535 范围内。缺省值是 100。

MaxActive 通道数 =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 |number

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。

PipeLine 长度 =1 |数字

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 PipeLine 长度。

注: 管道传输仅对 TCP/IP 通道有效。

AdoptNewMCA=NO| SVR | SDR | RCVR | CLUSRCVR | ALL | FASTPATH

如果 WebSphere MQ 接收到启动通道的请求，但发现该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. 如果在 AdoptNewMCATimeout 等待时间间隔到期之前，先前的通道服务器未响应此请求，那么将结束先前通道服务器的线程或进程。
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，WebSphere 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。

ALL

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! : 对于 FASTPATH 通道，`AdoptNewMCA` 属性的行为可能是不可预测的。对 FASTPATH 通道启用 `AdoptNewMCA` 属性时请务必谨慎。

AdoptNewMCATimeout=60 | 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。如果这与 `hosta` 不同，那么 `AdoptNewMCACheck=ADDRESS` 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。

ALL

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`。

相关概念

第 47 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。

TCP，LU62，NETBIOS 和 SPX

使用这些队列管理器属性页面或 `qm.ini` 文件中的节来指定网络协议配置参数。它们将覆盖通道的缺省属性。

TCP

使用 IBM WebSphere MQ Explorer 中的 TCP 队列管理器属性页面或 `qm.ini` 文件中的 TCP 节来指定传输控制协议/Internet Protocol (TCP/IP) 配置参数。

端口 =1414|port_number

TCP/IP 会话的缺省端口号 (十进制表示法)。WebSphere MQ 的知名端口号为 1414。

库 1 =DLLName1 (仅适用于 Windows 的 WebSphere MQ)

TCP/IP 套接字 DLL 的名称。

缺省值是 WSOCK32。

KeepAlive=NO|YES

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。

ListenerBacklog= 数字

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。这可以视为在 TCP/IP 端口上等待侦听器接受请求的积压请求。缺省侦听器任务列表值显示在 第 367 页的表 33 中。

平台	缺省 ListenerBacklog 值
Windows 服务器	100
Windows 工作站	5
Linux	100
Solaris	100
HP-UX	20
AIX V4.2 或更高版本	100
AIX V4.1 或更低版本	10

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 367 页的表 33 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。

SvrSndBuffSize=32768|number

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=32768|number

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)。

Connect_Timeout=0|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。

LU62 (仅限 WebSphere MQ for Windows)

使用 IBM WebSphere MQ Explorer 中的 LU6.2 队列管理器属性页面或 qm.ini 文件中的 LU62 节来指定 SNA LU 6.2 协议配置参数。

TPName

要在远程站点上启动的 TP 名称。

库 1 =DLLName 1

APPC DLL 的名称。

缺省值为 WCPIC32。

库 2 =DLLName2

与 Library1 相同，如果代码存储在两个单独的库中。

缺省值为 WCPIC32。

NETBIOS (仅适用于 Windows 的 WebSphere MQ)

使用 IBM WebSphere MQ Explorer 中的 Netbios 队列管理器属性页面或 qm.ini 文件中的 NETBIOS 节来指定 NetBIOS 协议配置参数。

LocalName=名称

LAN 上用于识别此机器的名称。

AdapterNum=0|adapter_number

LAN 适配器的编号。缺省值为适配器 0。

NumSess=1|number_of_sessions

要分配的会话数。缺省值是 1。

NumCmds=1|number_of_commands

要分配的命令数。缺省值是 1。

NumNames=1|number_of_names

要分配的名称数。缺省值是 1。

库 1 =DLLName1

NetBIOS DLL 的名称。

缺省值为 NETAPI32。

SPX (仅限 WebSphere MQ for Windows)

使用 IBM WebSphere MQ Explorer 中的 SPX 队列管理器属性页面或 qm.ini 文件中的 SPX 节来指定 SPX 协议配置参数。

套接字 =5E86|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|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。

库 1 =DLLName1

SPX DLL 的名称。

缺省值为 WSOCK32.DLL。

库 2 =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。这可被视为在 SPX 套接字上等待侦听器接受请求的积压请求。缺省侦听器任务列表值显示在 [第 368 页的表 34](#) 中。

表 34: 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 368 页的表 34](#) 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。

出口路径

使用 IBM WebSphere MQ Explorer 中的 Exits 队列管理器属性页面或 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。

ExitsDefault 路径 =字符串

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

API 出口

对于服务器, 请使用 IBM WebSphere MQ Explorer 中的 Exits 队列管理器属性页面或 `qm.ini` 文件中的 `ApiExitLocal` 节来标识队列管理器的 API 出口例程。对于客户机, 请修改 `mqclient.ini` 文件中的 `ApiExitLocal` 节, 以标识队列管理器的 API 出口例程。

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 出口例程, 请使用 `ApiExitCommon` 和 `ApiExitTemplate` 节, 如第 357 页的『API 出口』中所述。)

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。

有关这些节的属性的完整描述, 请参阅 [配置 API 出口](#)。

UNIX, Linux, and Windows 上的 QMErrorLog 节

使用 WebSphere MQ Explorer 中的 Extended 队列管理器属性页面或 `qm.ini` 文件中的 `QMErrorLog` 节来定制队列管理器错误日志的操作和内容。



注意: 仅当您在 Windows 平台上使用本地队列管理器时, 才能使用 WebSphere MQ Explorer 进行更改。

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 2097152 字节 (2 MB)。

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 WebSphere 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。WebSphere MQ-IMS 网桥和缓冲区管理器也可能生成大量参考消息, 因此排除消息会阻止您在需要时接收大量消息。 *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
- 7234 - 装入的消息数
- 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
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 WebSphere 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。WebSphere MQ-IMS 网桥和缓冲区管理器也可能生成大量参考消息, 因此禁止消息会阻止您在需要时接收大量重复消息。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
7234 - 装入的消息数
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
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。

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。

队列管理器缺省绑定类型

使用 IBM WebSphere MQ Explorer 中的 Extended 队列管理器属性页面或 qm.ini 文件中的 Connection 节来指定缺省绑定类型。

DefaultBind 类型 =SHARED|ISOLATED

如果将 DefaultBind 类型设置为隔离, 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 DefaultBind 类型设置为 SHARED, 那么应用程序和队列管理器将在单独的进程中运行, 但某些资源将在它们之间共享。

缺省值为 shared。

队列管理器配置文件的 SSL 和 TLS 节

使用队列管理器配置文件的 SSL 节在队列管理器上配置 SSL 或 TLS 通道。

联机证书状态协议 (OCSP)

证书可以包含 AuthorityInfo 访问扩展。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 AuthorityInfo 访问扩展，请确保其中指定的 OCSP 服务器可用，已正确配置并且可通过网络访问。有关更多信息，请参阅 [使用已撤销的证书](#)。

CrlDistribution 点 (CDP)

证书可以包含 CrlDistribution 点扩展。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 CrlDistribution 点扩展，请确保其中指定的 CDP 服务器可用，配置正确并且可通过网络访问。

SSL 节

使用 `qm.ini` 文件中的 SSL 节来配置队列管理器上的 SSL 或 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。

CDPCheckExtensions=YES|NO

CDPCheckExtensions 指定此队列管理器上的 SSL 或 TLS 通道是否尝试检查在 CrlDistribution 点证书扩展中指定的 CDP 服务器。

- YES: SSL 或 TLS 通道尝试检查 CDP 服务器以确定是否撤销数字证书。
- NO: SSL 或 TLS 通道不会尝试检查 CDP 服务器。此值是缺省值。

OCSPAAuthentication=REQUIRED|WARN|OPTIONAL

OCSPAAuthentication 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 SSL 或 TLS 通道程序会尝试联系 OCSP 服务器。

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 OCSPAAuthentication 参数的值。

- REQUIRED: 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=YES|NO

OCSPCheckExtensions 指定此队列管理器上的 SSL 和 TLS 通道是否尝试检查在 AuthorityInfo 访问证书扩展中指定的 OCSP 服务器。

- YES: SSL 和 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: SSL 和 TLS 通道不会尝试检查 OCSP 服务器。

SSLHTTPProxyName=字符串

该字符串是供 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。在 HP-UX PA-RISC 和 Sun Solaris SPARC 平台上，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，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。

出口属性

使用 IBM WebSphere MQ Explorer 中的 "集群队列管理器属性" 页面或 qm.ini 文件中的 ExitPropertiesLocal 节来指定有关队列管理器上的出口属性的信息。或者，可以使用 **amqmdain** 命令对其进行设置。

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 354 页的『出口属性』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。

CLWLMode=SAFE|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, 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。

SAFE

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 (amqzlwao) 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。

子池

此节由 WebSphere MQ 创建。请勿对其进行更改。

该节中的子池和属性 ShortSubpoolName 由 WebSphere MQ 在创建队列管理器时自动写入。WebSphere MQ 选择 ShortSubpool 名称的值。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，WebSphere MQ 用于在其运行的进程之间进行内部通信。

配置 HP Integrity NonStop Server

使用此信息可帮助您配置 IBM WebSphere MQ 客户机以进行 HP Integrity NonStop Server 安装。

有关使用配置文件配置客户机的详细信息，请参阅第 106 页的『使用配置文件配置客户机』。

有关使用环境变量配置客户机的详细信息，请参阅第 119 页的『使用 WebSphere MQ 环境变量』。

如果要在 TMF/Gateway 下执行 IBM WebSphere MQ Client for HP Integrity NonStop Server 操作，请参阅子主题以获取有关如何配置 TMF/Gateway 的信息。其中包括 Gateway 进程概述，将 Gateway 配置为在 Pathway 下运行，以及配置客户机初始化文件以使 IBM WebSphere MQ Client for HP Integrity NonStop Server 能够访问 TMF Gateway。

本部分还包含 IBM WebSphere MQ Client for HP Integrity NonStop Server 有关授予通道许可权的特定信息。

网关进程概述

HP NonStop Transaction Management Facility (TMF) 提供支持网关进程注册为资源管理器的服务。IBM WebSphere MQ 提供的 TMF/Gateway 进程在 Pathway 下运行。

IBM WebSphere MQ 为由 TMF 协调的每个队列管理器注册单个网关进程，因此必须为要参与 TMF 协调工作单元的每个队列管理器配置单独的 TMF/Gateway。此注册使每个队列管理器都是独立的资源管理器，出于管理目的，向 HP NonStop TMF 注册每个队列管理器一次会生成易于理解的映射。

对于 IBM WebSphere MQ 的多个安装，必须为要由 TMF 协调的每个队列管理器指定其中一个安装的单个网关进程。

Gateway 进程接口支持任何相同或较早版本的客户机。

有关管理网关进程的更多信息，请参阅 [管理 HP Integrity NonStop Server](#)。

配置 Gateway 以在 Pathway 下运行

TMF/Gateway 是 HP NonStop 事务管理设施 (TMF) 与 IBM WebSphere MQ 之间的接口，使 TMF 能够成为 IBM WebSphere MQ 事务的事务协调程序。

IBM WebSphere MQ 提供的 TMF/Gateway 将事务从 TMF 协调转换为 eXtended 体系结构 (XA) 事务协调，以及与远程队列管理器进行通信。

每个队列管理器必须有一个需要协调的 TMF/Gateway，并且需要客户机配置，以便客户机可以连接到正确的 Gateway。

TMF/Gateway 可以使用客户机可用于与队列管理器通信的所有机制。以针对其他应用程序的方式配置 TMF/Gateway。

TMF/Gateway 不是 HP Integrity NonStop Server 进程对，并且设计为在 Pathway 环境中运行。TMF/Gateway 在 TMF 中创建永久资源，并在后续运行时复用这些资源，因此 TMF/Gateway 必须始终在同一用户权限下运行。

定义 serverclass

TMF/Gateway 在 Pathway 环境中作为服务器类进行托管。要定义 serverclass，必须设置以下服务器属性：

PROCESSTYPE=OSS

指定服务器类中的服务器类型。Gateway 进程是多线程 OSS 程序。此属性是必需的，并且必须设置为 OSS。

MAXSERVERS=1

指定此服务器类中可以同时运行的最大服务器进程数。对于任何队列管理器，只能有一个网关进程。此属性是必需的，必须设置为 1。

NUMSTATIC=1

指定此服务器类中的最大静态服务器数。网关进程必须作为静态服务器运行。此属性是必需的，必须设置为 1。

TMF=ON

指定此服务器类中的服务器是否可以锁定和更新由 TMF 子系统审计的数据文件。Gateway 进程参与 IBM WebSphere MQ 客户机应用程序的 TMF 事务，因此此属性必须设置为 ON。

PROGRAM=<OSS installation path>/opt/mqm/bin/runmqtmf

对于 IBM WebSphere MQ client for IBM WebSphere MQ，此属性必须为 runmqtmf。此属性必须是绝对 OSS 路径名。案例很重要。

ARGLIST=-m<QMgr name>[-c<channel name>][,-p<port>][,-h<host name>][,-n<max threads>]

这些属性为 Gateway 进程提供参数，其中：

- **QMgrName** 是此网关进程的队列管理器的名称。如果您正在使用队列共享组（或其他端口分配技术），则此参数必须以特定队列管理器为目标。该参数是必需的。
- **channel name** 是队列管理器上要由 Gateway 进程使用的服务器通道的名称。此参数是可选的。
- **port** 是队列管理器的 TCP/IP 端口。此参数是可选的。

- `host name` 是队列管理器的主机名。此参数是可选的。
- **最大线程数** 是 Gateway 进程创建的最大工作程序线程数。此参数的值可以等于或大于 10。即使指定了小于 10 的值，使用的最小值也为 10。如果没有提供值，则 Gateway 进程最多可创建 50 个线程。

使用 `-c`、`-p` 和 `-h` 属性作为向网关提供连接信息的替代方法，以及第 374 页的『使用环境变量配置 TMF/Gateway』中描述的方法。如果指定一个或多个 `-c`、`-p` 和 `-h` 属性，但未指定缺省值为以下值：

- **通道名称** 缺省为 `SYSTEM.DEF.SVRCONN`
- `host name` 缺省为 `localhost`
- `port` 缺省为 `1414`

如果您提供的任何参数无效，那么 TMF/Gateway 会向错误日志发出诊断消息 [AMQ5379](#) 并终止。

OWNER=ID

网关运行所使用的用户标识，必须向该用户标识授予对队列管理器的连接权限。

SECURITY="value"

指定可以从 IBM WebSphere MQ 客户机应用程序访问 Gateway 的用户 (与 `Owner` 属性相关)。

必须为 `LINKDEPTH` 和 `MAXLINKS` 配置相应的值，以满足可能希望与 Gateway 并发通信的 IBM WebSphere MQ 客户机应用程序的预期数目。如果这些值设置过低，那么您可能会看到从客户机应用程序发出的错误消息 [AMQ5399](#) 出现。

有关这些服务器属性的更多信息，请参阅 *HP NonStop TS/MP 2.5 系统管理手册*。

使用环境变量配置 TMF/Gateway

定义 TMF/Gateway 最常用的方法之一是设置 `MQSERVER` 环境变量，例如：

```
ENV MQSERVER=<channel name>/<transport>/<host name>(<listener port>)
```

命令开头的 ENV 是 Pathway 表示法。

配置客户机初始化文件

如果您正在使用 HP NonStop 事务管理设施 (TMF)，那么必须具有 IBM WebSphere MQ 客户机初始化文件，以使 IBM WebSphere MQ 客户机能够使 HP Integrity NonStop Server 访问 TMF 网关。

HP Integrity NonStop Server 的 IBM WebSphere MQ 客户机初始化文件可以保存在多个位置，有关更多信息，请参阅第 108 页的『客户机配置文件的位置』。

有关配置文件内容的详细信息以及示例，请参阅第 106 页的『使用配置文件配置客户机』。使用 TMF 节来指定 TMF 队列管理器和服务器详细信息，有关更多信息，请参阅第 119 页的『TMF 和 TMF/Gateway 节』。

HP Integrity NonStop Server 的 IBM WebSphere MQ 客户机的条目示例如下：

```
TMF:
  PathMon=$PSD1P

TmfGateway:
  QManager=MQ5B
  Server=MQ-MQ5B

TmfGateway:
  QManager=MQ5C
  Server=MQ-MQ5C
```

有关使用环境变量配置客户机的更多信息，请参阅第 119 页的『使用 WebSphere MQ 环境变量』。

向通道授予许可权

授予对 IBM WebSphere MQ Client for HP Integrity NonStop Server 上的通道的许可权与其他操作系统相同，但是您必须知道正在运行网关的所有者的标识。

然后，可以使用网关所有者的标识来授予相应的许可权。重要的区别是，授予对队列管理器通道的许可权不受任何应用程序的权限。

使用 **setmqaut** 命令来授予授权 (即，授予 IBM WebSphere MQ 主体或用户组执行操作的许可权) 和撤销授权 (即，除去执行操作的许可权)。

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示的）保证，包括但不限于暗示的有关非侵权，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示的保证。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。

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 IBM 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 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 (<http://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: