

7.5

IBM WebSphere MQ 配置参考

IBM

注

在使用本资料及其支持的产品之前，请阅读第 127 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

配置参考	5
所有平台的示例配置信息.....	5
如何使用通信示例.....	7
示例配置- IBM WebSphere MQ for Windows.....	8
示例配置- IBM WebSphere MQ for AIX.....	16
示例配置- IBM WebSphere MQ for HP-UX.....	21
示例配置- IBM WebSphere MQ for Solaris.....	27
示例配置- IBM WebSphere MQ for Linux.....	32
队列名称.....	38
其他对象名.....	39
队列名称解析.....	40
什么是队列名称解析?.....	41
系统和缺省对象.....	42
Windows 缺省配置对象.....	45
SYSTEM.BASE.TOPIC.....	46
节信息.....	47
用于分布式排队的配置文件节.....	49
通道属性.....	50
通道属性和通道类型.....	50
按字母顺序排列的通道属性.....	53
IBM WebSphere MQ 集群命令.....	76
队列管理器定义命令.....	77
通道定义命令.....	78
队列定义命令.....	80
DISPLAY CLUSQMGR.....	81
暂挂 QMGR 和 RESUME QMGR.....	83
刷新集群.....	83
重置集群: 强制从集群中移除队列管理器.....	84
工作负载均衡.....	85
集群工作负载出口调用和数据结构.....	95
通道程序.....	117
环境变量.....	118
分布式平台的消息通道规划示例.....	122
示例显示的内容.....	122
运行示例.....	125
使用别名来引用 MQ 库.....	125
声明	127
编程接口信息.....	128
商标.....	128

配置参考

使用本部分中的参考信息可帮助您配置 WebSphere MQ。

以下子主题中提供了配置参考信息：

相关任务

[配置](#)

示例配置信息

配置示例描述为建立有效的 WebSphere MQ 网络而执行的任务。这些任务是建立 WebSphere MQ 发送方和接收方通道，以通过所有受支持的协议在平台之间启用双向消息流。

要使用除发送方-接收方以外的通道类型，请参阅 [MQSC 引用中的 DEFINE CHANNEL 命令](#)。

第 5 页的图 1 是单个通道以及与其关联的 WebSphere MQ 对象的概念表示。

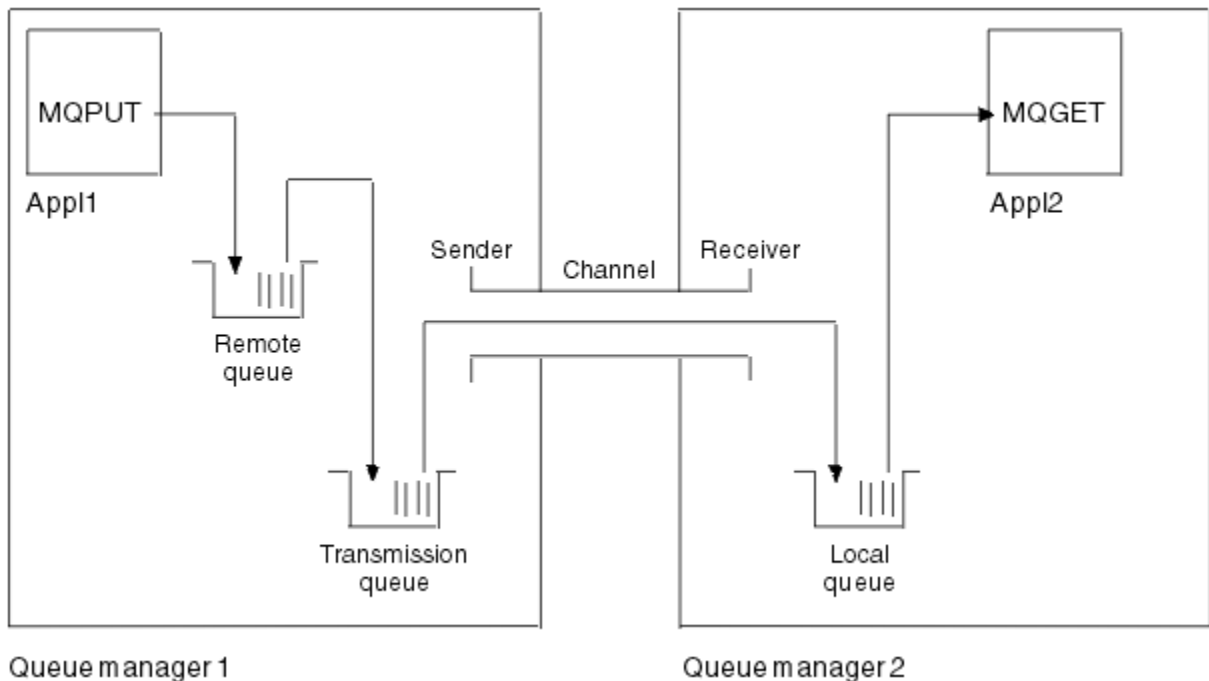


图 1: 要在示例配置中设置的 WebSphere MQ 通道

此示例是一个简单示例，旨在仅引入 WebSphere MQ 网络的基本元素。它不会演示如何使用 [触发通道](#) 中描述的触发。

此网络中的对象为：

- 远程队列
- 传输队列
- 本地队列
- 发送方通道
- 接收方通道

App1 和 App2 都是应用程序；App1 正在放置消息，App2 正在接收消息。

App1 将消息放入远程队列。此远程队列的定义指定目标队列管理器，该队列管理器上的本地队列以及此本地队列管理器上的传输队列的名称。

当队列管理器从 Appl1 接收到将消息放入远程队列的请求时，队列管理器根据队列定义确定目标是远程的。因此，它将消息与传输头直接放在定义中指定的传输队列上。消息将保留在传输队列上，直到通道变为可用为止，这可能会立即发生。

发送方通道在其定义中具有对一个且仅一个传输队列的引用。当通道启动时，在其正常运行期间的其他时间，它会查看此传输队列并将其上的任何消息发送到目标系统。消息在其传输头中包含目标队列和队列管理器的详细信息。

相互通信示例详细描述了针对各种平台组合创建上述每个对象的过程。

在目标队列管理器上，本地队列和通道接收方需要定义。这些对象相互独立操作，因此可以按任何顺序创建。

在本地队列管理器上，远程队列，传输队列和通道的发送方端都需要定义。由于远程队列定义和通道定义都引用了传输队列名称，因此建议先创建传输队列。

示例中的网络基础结构

配置示例假定特定网络基础结构适用于特定平台：

- z/OS 使用连接到令牌环的 3745 网络控制器 (或同等网络控制器) 进行通信
- Solaris 位于还连接到 3745 网络控制器 (或等效) 的相邻局域网 (LAN) 上
- 所有其他平台都连接到令牌环网络

还假定对于 SNA，VTAM 和网络控制程序 (NCP) 中的所有必需定义都已就绪，并且已激活，以便 LAN 连接的平台通过广域网 (WAN) 进行通信。

同样，对于 TCP，假定名称服务器功能可用，可以使用域名服务器，也可以使用本地保存的表 (例如主机文件)。

示例中的通信软件

以下网络软件产品的示例中提供了工作配置：

- SNA
 - IBM Personal Communications for Windows V5.9
 - IBM Communications Server for AIX, V6.3
 - Hewlett-Packard SNAplus2
 - IBM i
 - 数据连接 SNAP-IX V 7 或更高版本
 - OS/390 V 2 发行版 4
- TCP
 - Microsoft Windows
 - AIX V 4 Release 1.4
 - HP-UX V 10.2 或更高版本
 - Sun Solaris Release 2.4 或更高版本
 - IBM i
 - TCP for z/OS
 - HP Tru64 UNIX
- NetBIOS
- SPX

相关任务

[配置](#)

如何使用通信示例

示例配置描述了在单个平台上执行的用于设置与另一个平台的通信的任务。然后，他们描述了建立到该平台的工作通道的任务。

在可能的情况下，目的是使信息尽可能通用。因此，要在不同平台上连接任何两个队列管理器，只需要引用相关的两个部分。因此，将突出显示任何偏差或特殊情况。您还可以连接在同一平台上（在不同机器上或同一机器上）运行的两个队列管理器。在这种情况下，所有的信息都可以从一个部分中得到。

如果您正在使用 Windows，UNIX 或 Linux® 系统，那么在开始遵循针对您的平台的指示信息之前，必须设置各种环境变量。通过输入下列其中一个命令来设置环境变量：

- 在 Windows 上：

```
MQ_INSTALLATION_PATH/bin/setmqenv
```

其中 `MQ_INSTALLATION_PATH` 是指 IBM WebSphere MQ 的安装位置。

- 在 UNIX and Linux 系统上：

```
. MQ_INSTALLATION_PATH/bin/setmqenv
```

其中 `MQ_INSTALLATION_PATH` 是指 IBM WebSphere MQ 的安装位置。此命令为您当前正在使用的 shell 设置环境变量。如果打开另一个 shell，那么必须再次输入该命令。

有一些工作表，您可以在其中找到示例配置中使用的参数。提供了每个参数的简短描述以及有关在系统中查找等效值的位置的一些指导信息。当您有一组自己的值时，请将这些值记录在工作表上的空间中。当您继续完成此部分时，您将在需要时找到对这些值的交叉引用。

这些示例不涵盖如何在使用集群的情况下设置通信。有关在使用集群时设置通信的信息，请参阅 [配置队列管理器集群](#)。此处提供的通信配置值仍然适用。

以下平台有一些示例配置：

- [第 8 页的『示例配置- IBM WebSphere MQ for Windows』](#)
- [第 16 页的『示例配置- IBM WebSphere MQ for AIX』](#)
- [第 21 页的『示例配置- IBM WebSphere MQ for HP-UX』](#)
- [第 27 页的『示例配置- IBM WebSphere MQ for Solaris』](#)
- [第 32 页的『示例配置- IBM WebSphere MQ for Linux』](#)

IT 职责

要了解示例中使用的术语，请将以下准则作为起点。

- 系统管理员：为特定平台安装和配置软件的人员（或人员组）。
- 网络管理员：控制 LAN 连接，LAN 地址分配，网络命名约定和其他网络任务的人员。此人员可以在单独的组中，也可以是系统管理组的一部分。

在大多数 z/OS 安装中，有一个组负责更新 ACF/VTAM，ACF/NCP 和 TCP/IP 软件以支持网络配置。此组中的人员是将任何 WebSphere MQ 平台连接到 WebSphere MQ for z/OS 时所需的主要信息源。它们还会影响或要求 LAN 上的网络命名约定，您必须先验证其控制范围，然后再创建定义。

- 在我们可以更清楚地描述人员职责的情况下，会指示特定类型的管理员（例如 CICS 管理员）。

示例配置部分不会尝试指示谁负责并能够设置每个参数。一般来说，可能涉及到几个不同的人。

相关概念

[第 5 页的『示例配置信息』](#)

配置示例描述为建立有效的 WebSphere MQ 网络而执行的任务。这些任务是建立 WebSphere MQ 发送方和接收方通道，以通过所有受支持的协议在平台之间启用双向消息流。

相关参考

[setmqenv](#)

示例配置- IBM WebSphere MQ for Windows

本部分提供了如何在其他平台上设置从 IBM WebSphere MQ for Windows 到 IBM WebSphere MQ 产品的通信链接的示例。

以下平台上显示了通信链路的设置:

- AIX
- HP Tru64 UNIX
- HP-UX
- Solaris
- Linux
- IBM i
- z/OS
- VSE/ESA

建立连接时, 必须定义一些通道以完成配置。第 11 页的『[IBM WebSphere MQ 配置](#)』中描述了用于配置的示例程序和命令。

请参阅第 5 页的『[示例配置信息](#)』, 以获取有关此部分以及如何使用此部分的背景信息。

建立 LU 6.2 连接

请参阅有关配置 AnyNet SNA over TCP/IP 的信息。

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, SNA Node Operations](#) 和 [Communications Server for Windows](#)

建立 TCP 连接

Windows 系统随附的 TCP 堆栈不包含 *inet* 守护程序或等效守护程序。

用于启动 WebSphere MQ for TCP 侦听器的 WebSphere MQ 命令为:

```
runmqtsr -t tcp
```

必须先显式启动侦听器, 然后才能启动任何通道。它允许接收通道自动启动以响应来自入站发送通道的请求。

接下来怎么做?

建立 TCP/IP 连接时, 您已准备好完成配置。转至第 11 页的『[IBM WebSphere MQ 配置](#)』。

建立 NetBIOS 连接

从队列管理器启动 NetBIOS 连接, 该队列管理器在其通道定义上使用 `ConnectionName` 参数来连接到目标侦听器。

要设置 NetBIOS 连接, 请执行以下步骤:

1. 在通道的每一端指定要由队列管理器配置文件 `qm.ini` 中的 IBM WebSphere MQ 通道进程使用的本地 NetBIOS 名称。例如, 发送端的 Windows 中的 NETBIOS 节可能类似于以下内容:

```
NETBIOS:  
LocalName=WNTNETB1
```

并在接收端:


```
NETBIOS:
LocalName=WNTNETB2
```

每个 IBM WebSphere MQ 进程都必须使用不同的本地 NetBIOS 名称。请勿将系统名称用作 NetBIOS 名称，因为 Windows 已使用该名称。

2. 在通道的每一端，验证系统上正在使用的 LAN 适配器号。逻辑适配器号 0 的 IBM WebSphere MQ for Windows 缺省值是通过 Internet Protocol 网络运行的 NetBIOS。要使用本机 NetBIOS，必须选择逻辑适配器号 1。请参阅 [建立 LAN 适配器号](#)。

在 Windows 注册表的 NETBIOS 节中指定正确的 LAN 适配器号。例如：

```
NETBIOS:
AdapterNum=1
```

3. 因此，发送方通道启动工作，请通过 MQNAME 环境变量指定本地 NetBIOS 名称：

```
SET MQNAME=WNTNETB1I
```

此名称必须唯一。

4. 在发送端，定义一个指定在通道另一端使用的 NetBIOS 名称的通道。例如：

```
DEFINE CHANNEL (WINNT.OS2.NET) CHLTYPE(SDR) +
TRPTYPE(NETBIOS) +
CONNNAME(WNTNETB2) +
XMITQ(OS2) +
MCATYPE(THREAD) +
REPLACE
```

必须指定选项 MCATYPE(THREAD)，因为在 Windows 上，发送方通道必须作为线程运行。

5. 在接收端，定义相应的接收机信道。例如：

```
DEFINE CHANNEL (WINNT.OS2.NET) CHLTYPE(RCVR) +
TRPTYPE(NETBIOS) +
REPLACE
```

6. 启动通道启动程序，因为每个新通道都作为线程启动，而不是作为新进程启动。

```
runmqchi
```

7. 在接收端，启动 IBM WebSphere MQ 侦听器：

```
runmqlsr -t netbios
```

(可选) 可以指定队列管理器名称，NetBIOS 局部名，会话数，名称数和命令数的值。请参阅 [在 Windows 上定义 NetBIOS 连接](#)，以获取有关设置 NetBIOS 连接的更多信息。

建立 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。

本部分包含有关以下内容的信息：

- IPX/SPX 参数
- SPX 寻址
- 在 SPX 上接收

IPX/SPX 参数

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
```

SPX 寻址

WebSphere MQ 使用每台机器的 SPX 地址来建立连接。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。

WebSphere MQ 使用的缺省套接字号为 5E86。您可以通过在 Windows 注册表或队列管理器配置文件 qm.ini 中指定缺省套接字号来更改该套接字号。Windows 注册表中的行可能为：

```
SPX:  
SOCKET=n
```

有关可以在 qm.ini 中设置的值的更多信息，请参阅第 49 页的『用于分布式排队的配置文件节』。

稍后在发送方通道定义的 CONNAME 参数中指定 SPX 地址。如果正在连接的 WebSphere MQ 系统位于同一网络上，那么无需指定网络地址。同样，如果远程系统正在侦听缺省套接字号 (5E86)，那么无需指定该值。CONNAME 参数中的标准 SPX 地址为：

```
CONNAME('network.node(socket)')
```

但是如果系统位于同一网络上并且使用了缺省套接字号，那么参数为：

```
CONNAME(node)
```

第 11 页的『IBM WebSphere MQ 配置』中提供了通道配置参数的详细示例。

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。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。

您应该使用 WebSphere MQ 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
```

(可选) 如果未使用缺省值，那么可以指定队列管理器名称或套接字号。

IBM WebSphere MQ 配置

用于配置的示例程序和命令。

注:

1. 您可以使用样本程序 AMQSBCG 来显示队列中所有消息的内容和头。例如:

```
AMQSBCG q_name qmgr_name
```

显示队列管理器 *qmgr_name* 中定义的队列 *q_name* 的内容。

或者, 可以在 IBM WebSphere MQ Explorer 中使用消息浏览器。

2. 您可以使用命令从命令提示符启动任何通道

```
runmqchl -c channel.name
```

3. 可以在目录 *MQ_INSTALLATION_PATH\qmgs\qmgrname\errors* 和 *MQ_INSTALLATION_PATH\qmgs\@system\errors* 中找到错误日志。在这两种情况下, 最新的消息都位于 *amqerr01.log* 的末尾。

MQ_INSTALLATION_PATH 表示安装 WebSphere MQ 的高级目录。

4. 使用命令解释器 **runmqsc** 输入管理命令时, 行尾的 + 表示下一行是连续行。确保最后一个参数与连续字符之间有空格。

缺省配置

您可以使用 WebSphere MQ Postcard 应用程序来创建缺省配置, 以指导您完成此过程。

有关使用 Postcard 应用程序的信息, 请参阅 [使用 Postcard 应用程序验证安装](#)。

基本配置

您可以从 IBM WebSphere MQ Explorer 或命令提示符创建和启动队列管理器。

. 如果选择命令提示符:

1. 使用以下命令创建队列管理器:

```
crtmqm -u dlqname -q winnt
```

其中:

winnt

是队列管理器的名称

-q

指示这将成为缺省队列管理器

-u dlqname

指定无法传递的消息队列的名称

此命令创建队列管理器 and 一组缺省对象。

2. 使用以下命令启动队列管理器:

```
strmqm winnt
```

其中, *winnt* 是创建队列管理器时提供给该队列管理器的名称。

通道配置

要在 Windows 队列管理器上执行以实现给定通道的示例配置。

以下部分详细描述了要在 Windows 队列管理器上执行以实现 [第 5 页的图 1](#) 中描述的通道的配置。

在每种情况下都会显示 MQSC 命令。从命令提示符启动 **runmqsc** 并依次输入每个命令，或者将命令构建到命令文件中。

提供了用于连接 WebSphere MQ for Windows 和 WebSphere MQ for AIX 的示例。要在另一个平台上连接到 WebSphere MQ，请使用表中的相应值集来代替 Windows 的值集。

注：粗体中的词是用户指定的，反映在这些示例中使用的 WebSphere MQ 对象的名称。如果更改此处使用的名称，请确保在此部分中还更改对这些对象的其他引用。所有其他都是关键字，应按所示输入。

表 1: WebSphere MQ for Windows 的配置工作表				
	参数名称	参考	使用的示例	用户值
本地节点的定义				
A	队列管理器名称		winnt	
B	本地队列名称		WINNT.LOCALQ	
连接到 WebSphere MQ for AIX				
表的此部分中的值必须与第 18 页的表 2 中使用的值匹配，如所指示。				
C	远程队列管理器名称	A	AIX	
D	远程队列名称		AIX。远程队列	
E	远程系统上的队列名称	B	AIX。位置	
F	传输队列的名称		AIX	
G	发送方 (SNA) 通道名称		WINNT.AIX.SNA	
H	发送方 (TCP) 通道名称		WINNT.AIX.TCP	
I	接收方 (SNA) 通道名称	G	AIX.WINNT.SNA	
J	接收方 (TCP) 通道名称	H	AIX.WINNT.TCP	
连接到 WebSphere MQ for HP Tru64 UNIX				
表的此部分中的值必须与 HP Tru64 UNIX 系统中使用的值相匹配。				
C	远程队列管理器名称	A	DECUX	
D	远程队列名称		DECUX.REMOTEQ	
E	远程系统上的队列名称	B	DECUX.LOCALQ	
F	传输队列的名称		DECUX	
H	发送方 (TCP) 通道名称		DECUX.WINNT.TCP	
J	接收方 (TCP) 通道名称	H	WINNT.DECUX.TCP	
连接到 WebSphere MQ for HP-UX				
表的此部分中的值必须与第 23 页的表 3 中使用的值匹配，如所指示。				
C	远程队列管理器名称	A	HPUX	
D	远程队列名称		HPUX.REMOTEQ	
E	远程系统上的队列名称	B	HPUX.LOCALQ	
F	传输队列的名称		HPUX	
G	发送方 (SNA) 通道名称		WINNT.HPUX.SNA	
H	发送方 (TCP) 通道名称		WINNT.HPUX.TCP	

表 1: WebSphere MQ for Windows 的配置工作表 (继续)				
	参数名称	参考	使用的示例	用户值
I	接收方 (SNA) 通道名称	G	HPUX.WINNT.SNA	
J	接收方 (TCP/IP) 通道名称	H	HPUX.WINNT.TCP	
与 WebSphere MQ for Solaris 的连接				
表的此部分中的值必须与 第 29 页的表 4 中使用的值匹配, 如所指示。				
C	远程队列管理器名称	A	Solaris	
D	远程队列名称		SOLARIS.REMOTEQ	
E	远程系统上的队列名称	B	SOLARIS.LOCALQ	
F	传输队列的名称		Solaris	
G	发送方 (SNA) 通道名称		WINNT.SOLARIS.SNA	
H	发送方 (TCP) 通道名称		WINNT.SOLARIS.TCP	
I	接收方 (SNA) 通道名称	G	SOLARIS.WINNT.SNA	
J	接收方 (TCP) 通道名称	H	SOLARIS.WINNT.TCP	
连接到 WebSphere MQ for Linux				
表的此部分中的值必须与 第 35 页的表 5 中使用的值匹配, 如所指示。				
C	远程队列管理器名称	A	Linux	
D	远程队列名称		LINUX.REMOTEQ	
E	远程系统上的队列名称	B	LINUX.LOCALQ	
F	传输队列的名称		Linux	
G	发送方 (SNA) 通道名称		WINNT.LINUX.SNA	
H	发送方 (TCP) 通道名称		WINNT.LINUX.TCP	
I	接收方 (SNA) 通道名称	G	LINUX.WINNT.SNA	
J	接收方 (TCP) 通道名称	H	LINUX.WINNT.TCP	
C	远程队列管理器名称	A	AS400	
D	远程队列名称		AS400.REMOTEQ	
E	远程系统上的队列名称	B	AS400.LOCALQ	
F	传输队列的名称		AS400	
G	发送方 (SNA) 通道名称		WINNT.AS400.SNA	
H	发送方 (TCP) 通道名称		WINNT.AS400.TCP	
I	接收方 (SNA) 通道名称	G	AS400.WINNT.SNA	
C	远程队列管理器名称	A	MVS	
D	远程队列名称		MVS.REMOTEQ	
E	远程系统上的队列名称	B	MVS.LOCALQ	
F	传输队列的名称		MVS	

表 1: WebSphere MQ for Windows 的配置工作表 (继续)				
	参数名称	参考	使用的示例	用户值
G	发送方 (SNA) 通道名称		WINNT.MVS.SNA	
H	发送方 (TCP) 通道名称		WINNT.MVS.TCP	
I	接收方 (SNA) 通道名称	G	MVS.WINNT.SNA	
C	远程队列管理器名称	A	QSG	
D	远程队列名称		QSG.REMOTEQ	
E	远程系统上的队列名称	B	QSG.SHAREDQ	
F	传输队列的名称		QSG	
G	发送方 (SNA) 通道名称		WINNT.QSG.SNA	
H	发送方 (TCP) 通道名称		WINNT.QSG.TCP	
I	接收方 (SNA) 通道名称	G	QSG.WINNT.SNA	
连接到 VSE/ESA 的 MQSeries				
表的此部分中的值必须与 VSE/ESA 系统中使用的值匹配。				
C	远程队列管理器名称	A	VSE	
D	远程队列名称		VSE.REMOTEQ	
E	远程系统上的队列名称	B	VSE.LOCALQ	
F	传输队列的名称		VSE	
G	发送方通道名称		WINNT.VSE.SNA	
I	接收方通道名称	G	VSE.WINNT.SNA	

IBM WebSphere MQ (对于使用 SNA 的 Windows 发送方通道定义)
代码样本。

```

def ql (AIX) +                               F
  usage(xmitq) +
  replace

def qr (AIX.REMOTEQ) +                       D
  rname(AIX.LOCALQ) +                       E
  rqmname(AIX) +                             C
  xmitq(AIX) +                               F
  replace

def chl (WINNT.AIX.SNA) chltype(sdr) +       G
  trptype(lu62) +
  conname(AIXCPIC) +                         18
  xmitq(AIX) +                               F
  replace

```

IBM WebSphere MQ for Windows 接收方-通道定义 (使用 SNA)
代码样本。

```

def ql (WINNT.LOCALQ) replace                B

def chl (AIX.WINNT.SNA) chltype(rcvr) +     I
  trptype(lu62) +
  replace

```

IBM WebSphere MQ for Windows 发送方通道定义 (使用 TCP/IP)
代码样本。

```
def ql (AIX) + usage(xmitq) + replace F
def qr (AIX.REMOTEQ) + rname(AIX.LOCALQ) + rqmname(AIX) + xmitq(AIX) + replace D
E
C
F
def chl (WINNT.AIX.TCP) chltype(sdr) + trptype(tcp) + conname(remote_tcpip_hostname) + xmitq(AIX) + replace H
F
```

IBM WebSphere MQ for Windows 接收方-使用 TCP 的通道定义
代码样本。

```
def ql (WINNT.LOCALQ) replace B
def chl (AIX.WINNT.TCP) chltype(rcvr) + trptype(tcp) + replace J
```

自动启动

WebSphere MQ for Windows 允许您自动启动队列管理器及其通道启动程序，通道，侦听器 and 命令服务器。

使用 IBM WebSphere MQ Services 快照来定义队列管理器的服务。成功完成通信设置的测试后，请在插件中将相关服务设置为 **automatic**。系统启动时，提供的 WebSphere MQ 服务可以读取此文件。

有关更多信息，请参阅[管理 IBM WebSphere MQ](#)。

将通道作为进程或线程运行

WebSphere MQ for Windows 提供了作为 Windows 进程或 Windows 线程运行发送通道的灵活性。这是在发送方通道定义上的 MCATYPE 参数中指定的。

大多数安装将其发送通道作为线程运行，因为支持许多并发通道连接所需的虚拟内存和实内存会减少。但是，对于发送消息通道代理程序，NetBIOS 连接需要单独的进程。

多线程支持-流水线

您可以选择允许消息通道代理 (MCA) 使用多个线程来传输消息。此过程称为 流水线，使 MCA 能够更高效地传输消息，减少等待状态，从而提高通道性能。每个 MCA 限制为最多两个线程。

您可以使用 qm.ini 文件中的 *PipeLineLength* 参数来控制管道传输。此参数将添加到 CHANNELS 节中：

PipeLineLength=1|number

此属性指定通道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。

对于 WebSphere MQ for Windows，请使用 WebSphere MQ Explorer 在注册表中设置 *PipeLineLength* 参数。

注：

1. *PipeLineLength* 仅适用于 V5.2 或更高版本的产品。
2. 流水线仅对 TCP/IP 通道有效。

使用管道时，通道两端的队列管理器必须配置为具有大于 1 的 *PipeLine* 长度。

通道出口注意事项

管道可能导致某些出口程序失败，因为：

- 可能未按顺序调用出口。
- 可以从不同的线程交替调用出口。

在使用管道之前，请检查出口程序的设计：

- 出口必须在其执行的所有阶段都可重入。
- 使用 MQI 调用时，请记住，从不同线程调用出口时，不能使用相同的 MQI 句柄。

请考虑打开队列的消息出口，并将其句柄用于对该出口的所有后续调用的 MQPUT 调用。这在管道方式下失败，因为从不同的线程调用了出口。要避免此故障，请为每个线程保留一个队列句柄，并在每次调用出口时检查线程标识。

示例配置- IBM WebSphere MQ for AIX

本节提供了如何在其他平台上设置从 IBM WebSphere MQ for AIX 到 IBM WebSphere MQ 产品的通信链接的示例。

示例中涵盖了以下平台：

- Windows
- HP Tru64 UNIX
- HP-UX
- Solaris
- Linux
- IBM i
- z/OS
- VSE/ESA

请参阅 [第 5 页的『示例配置信息』](#)，以获取有关此部分以及如何使用此部分的背景信息。

建立 LU 6.2 连接

描述 LU 6.2 连接所需的参数。

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 for AIX](#)。

建立 TCP 连接

必须先显式启动侦听器，然后才能启动任何通道。它允许接收通道自动启动以响应来自入站发送通道的请求。

用于启动 WebSphere MQ for TCP 侦听器的 WebSphere MQ 命令为：

```
runmqclsr -t tcp
```

或者，如果要使用 UNIX 提供的 TCP/IP 侦听器，请完成以下步骤：

1. 编辑文件 /etc/services。

注：要编辑 /etc/services 文件，您必须以超级用户或 root 用户身份登录。如果该文件中没有以下行，请按如下所示添加该文件：

```
MQSeries      1414/tcp      # MQSeries channel listener
```

2. 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，将 `MQ_INSTALLATION_PATH` 替换为安装了 WebSphere MQ 的高级目录：

```
MQSeries stream tcp nowait root MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
[-m queue.manager.name]
```


3. 输入命令 `refresh -s inetd`。

注: 必须将 **root** 添加到 `mqm` 组。您无需将主组设置为 `mqm`。只要 `mqm` 在组集中, 就可以使用这些命令。如果仅运行使用队列管理器的应用程序, 那么您不需要 `mqm` 组权限。

接下来怎么做?

现在已建立连接。您已准备好完成配置。转至第 17 页的『[IBM WebSphere MQ for AIX 配置](#)』。

IBM WebSphere MQ for AIX 配置

定义通道以完成配置。

注:

1. 在开始安装过程之前, 请确保首先创建了 `mqm` 用户和组, 并设置密码。
2. 如果由于文件系统中的空间不足而导致安装失败, 那么可以使用命令 `smit C sna` 来增大大小, 如下所示。(使用 `df` 来显示文件系统的状态。这指示已满的逻辑卷。)

```
-- Physical and Logical Storage
-- File Systems
-- Add / Change / Show / Delete File Systems
-- Journalled File Systems
-- Change/Show Characteristics of a Journalled File System
```

3. 使用以下命令启动任何通道:

```
runmqchl -c channel.name
```

4. 样本程序安装在 `MQ_INSTALLATION_PATH/samp` 中, 其中 `MQ_INSTALLATION_PATH` 表示安装了 WebSphere MQ 的高级目录。
5. 错误日志存储在 `/var/mqm/qmgrs/qmgrname/errors` 中。
6. 在 AIX 上, 可以使用标准 WebSphere MQ 跟踪命令或使用 AIX 系统跟踪来启动 WebSphere MQ 组件的跟踪。有关 WebSphere MQ 跟踪和 AIX 系统跟踪的更多信息, 请参阅 [使用跟踪](#)。
7. 使用命令解释器 `runmqsc` 输入管理命令时, 行尾的 `+` 表示下一行是连续行。确保最后一个参数与连续字符之间有空格。

基本配置

1. 使用以下命令从 AIX 命令行创建队列管理器:

```
crtmqm -u dlqname -q aix
```

其中:

aix

是队列管理器的名称

-q

指示这将成为缺省队列管理器

-u dlqname

指定无法传递的消息队列的名称

此命令创建队列管理器 and 一组缺省对象。

2. 使用以下命令从 AIX 命令行启动队列管理器:

```
strmqm aix
```

其中, *aix* 是创建队列管理器时提供给该队列管理器的名称。

3. 从 AIX 命令行启动 **runmqsc**，并通过输入以下命令使用它来创建无法传递的消息队列：

```
def ql (dlqname)
```

其中，*dlqname* 是创建队列管理器时提供给无法传递的消息队列的名称。

通道配置

包含有关为给定通道和平台配置队列管理器的信息。

以下部分详细说明了要在 AIX 队列管理器上执行以实现 [第 5 页的图 1](#) 中描述的通道的配置。

在每种情况下都会显示 MQSC 命令。从 AIX 命令行启动 **runmqsc** 并依次输入每个命令，或者将这些命令构建到命令文件中。

提供了用于连接 WebSphere MQ for AIX 和 WebSphere MQ for Windows 的示例。要在另一个平台上连接到 WebSphere MQ，请使用表中的相应值集来代替 Windows 的值集。

注：粗体中的词是用户指定的，反映在这些示例中使用的 WebSphere MQ 对象的名称。如果更改此处使用的名称，请确保在此部分中还更改对这些对象的其他引用。所有其他都是关键字，应按所示输入。

标识	参数名称	参考	使用的示例	用户值
表 2: WebSphere MQ for AIX 的配置工作表				
本地节点的定义				
A	队列管理器名称		AIX	
B	本地队列名称		AIX. 位置	
连接到 WebSphere MQ for Windows				
表的此部分中的值必须与 第 12 页的表 1 中使用的值匹配，如所指示。				
C	远程队列管理器名称	A	winnt	
D	远程队列名称		WINNT.REMOTEQ	
E	远程系统上的队列名称	B	WINNT.LOCALQ	
F	传输队列的名称		winnt	
G	发送方 (SNA) 通道名称		AIX.WINNT.SNA	
H	发送方 (TCP/IP) 通道名称		AIX.WINNT.TCP	
I	接收方 (SNA) 通道名称	G	WINNT.AIX.SNA	
J	接收方 (TCP) 通道名称	H	WINNT.AIX.TCP	
连接到 WebSphere MQ for HP Tru64 UNIX				
表的此部分中的值必须与 HP Tru64 UNIX 系统中使用的值相匹配。				
C	远程队列管理器名称	A	DECUX	
D	远程队列名称		DECUX.REMOTEQ	
E	远程系统上的队列名称	B	DECUX.LOCALQ	
F	传输队列的名称		DECUX	
H	发送方 (TCP) 通道名称		DECUX.AIX.TCP	
J	接收方 (TCP) 通道名称	H	AIX.DECUX.TCP	
连接到 WebSphere MQ for HP-UX				
表的此部分中的值必须与 第 23 页的表 3 中使用的值匹配，如所指示。				

表 2: *WebSphere MQ for AIX* 的配置工作表 (继续)

标识	参数名称	参考	使用的示例	用户值
C	远程队列管理器名称	A	HPUX	
D	远程队列名称		HPUX.REMOTEQ	
E	远程系统上的队列名称	B	HPUX.LOCALQ	
F	传输队列的名称		HPUX	
G	发送方 (SNA) 通道名称		AIX.HPUX.SNA	
H	发送方 (TCP) 通道名称		AIX.HPUX.TCP	
I	接收方 (SNA) 通道名称	G	HPUX.AIX.SNA	
J	接收方 (TCP) 通道名称	H	HPUX.AIX.TCP	
与 <i>WebSphere MQ for Solaris</i> 的连接				
表的此部分中的值必须与 第 29 页的表 4 中使用的值匹配, 如所指示。				
C	远程队列管理器名称	A	Solaris	
D	远程队列名称		SOLARIS.REMOTEQ	
E	远程系统上的队列名称	B	SOLARIS.LOCALQ	
F	传输队列的名称		Solaris	
G	发送方 (SNA) 通道名称		AIX.SOLARIS.SNA	
H	发送方 (TCP/IP) 通道名称		AIX.SOLARIS.TCP	
I	接收方 (SNA) 通道名称	G	SOLARIS.AIX.SNA	
J	接收方 (TCP/IP) 通道名称	H	SOLARIS.AIX.TCP	
连接到 <i>WebSphere MQ for Linux</i>				
表的此部分中的值必须与 第 35 页的表 5 中使用的值匹配, 如所指示。				
C	远程队列管理器名称	A	Linux	
D	远程队列名称		LINUX.REMOTEQ	
E	远程系统上的队列名称	B	LINUX.LOCALQ	
F	传输队列的名称		Linux	
G	发送方 (SNA) 通道名称		AIX.LINUX.SNA	
H	发送方 (TCP/IP) 通道名称		AIX.LINUX.TCP	
I	接收方 (SNA) 通道名称	G	LINUX.AIX.SNA	
J	接收方 (TCP/IP) 通道名称	H	LINUX.AIX.TCP	
C	远程队列管理器名称	A	AS400	
D	远程队列名称		AS400.REMOTEQ	
E	远程系统上的队列名称	B	AS400.LOCALQ	
F	传输队列的名称		AS400	
G	发送方 (SNA) 通道名称		AIX. AS400.SNA	
H	发送方 (TCP) 通道名称		AIX. AS400. TCP	

表 2: WebSphere MQ for AIX 的配置工作表 (继续)				
标识	参数名称	参考	使用的示例	用户值
I	接收方 (SNA) 通道名称	G	AS400.AIX.SNA	
C	远程队列管理器名称	A	MVS	
D	远程队列名称		MVS.REMOTEQ	
E	远程系统上的队列名称	B	MVS.LOCALQ	
F	传输队列的名称		MVS	
G	发送方 (SNA) 通道名称		AIX.MVS.SNA	
H	发送方 (TCP) 通道名称		AIX.MVS.TCP	
I	接收方 (SNA) 通道名称	G	MVS。AIX.SNA	
C	远程队列管理器名称	A	QSG	
D	远程队列名称		QSG.REMOTEQ	
E	远程系统上的队列名称	B	QSG.SHAREDQ	
F	传输队列的名称		QSG	
G	发送方 (SNA) 通道名称		AIX.QSG.SNA	
H	发送方 (TCP) 通道名称		AIX.QSG.TCP	
I	接收方 (SNA) 通道名称	G	QSG.AIX.SNA	
连接到 VSE/ESA 的 MQSeries				
表的此部分中的值必须与 VSE/ESA 系统中使用的值匹配。				
C	远程队列管理器名称	A	VSE	
D	远程队列名称		VSE.REMOTEQ	
E	远程系统上的队列名称	B	VSE.LOCALQ	
F	传输队列的名称		VSE	
G	发送方通道名称		AIX.VSE.SNA	
I	接收方通道名称	G	VSE.AIX.SNA	

使用 SNA 的 IBM WebSphere MQ 发送方通道定义示例命令。

```

def ql (WINNT) +                               F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                       D
  rname(WINNT.LOCALQ) +                       E
  rqmname(WINNT) +                             C
  xmitq(WINNT) +                               F
  replace

def chl (AIX.WINNT.SNA) chltype(sdr) +         G
  trptype(lu62) +
  conname('WINNTCPIC') +                     17
  xmitq(WINNT) +                               F
  replace

```

IBM WebSphere MQ for AIX 接收方-使用 SNA 的通道定义
示例命令。

```
def ql (AIX.LOCALQ) replace B
def chl (WINNT.AIX.SNA) chltype(rcvr) + I
  trptype(lu62) +
  replace
```

IBM WebSphere MQ for AIX TPN 设置

当发送方通道启动对话时，确保 SNA 接收方通道正确激活的替代方法。

在 AIX Communications Server 配置过程中，创建了一个 LU 6.2 TPN 概要文件，其中包含 TP 可执行程序
的完整路径。在此示例中，该文件名为 u/interops/AIX.crs6a。您可以选择名称，但请考虑在其中包含队列管
理器的名称。可执行文件的内容必须是：

```
#!/bin/sh
MQ_INSTALLATION_PATH/bin/amqcrs6a -m aix
```

其中，*aix* 是队列管理器名称 (A)，*MQ_INSTALLATION_PATH* 是安装了 WebSphere MQ 的高级目录。创
建此文件后，通过运行以下命令将其启用以执行：

```
chmod 755 /u/interops/AIX.crs6a
```

作为创建可执行文件的替代方法，您可以使用命令行参数在 "添加 LU 6.2 TPN 概要文件" 面板上指定路径。

在这两种方式之一中指定路径可确保当发送方通道启动对话时正确激活 SNA 接收方通道。

IBM WebSphere MQ (对于使用 TCP 的 AIX 发送方通道定义)

示例命令。

```
def ql (WINNT) + F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) + D
  rname(WINNT.LOCALQ) + E
  rqmname(WINNT) + C
  xmitq(WINNT) + F
  replace

def chl (AIX.WINNT.TCP) chltype(sdr) + H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(WINNT) + F
  replace
```

IBM WebSphere MQ for AIX 接收方-使用 TCP 的通道定义
示例命令。

```
def ql (AIX.LOCALQ) replace B
def chl (WINNT.AIX.TCP) chltype(rcvr) + J
  trptype(tcp) +
  replace
```

示例配置- IBM WebSphere MQ for HP-UX

本部分提供了如何在其他平台上设置从 IBM WebSphere MQ for HP-UX 到 IBM WebSphere MQ 产品的通信
链路的示例。

包含以下平台：

- Windows

- AIX
- HP Tru64 UNIX
- Solaris
- Linux
- IBM i
- z/OS
- VSE/ESA

请参阅 [第 5 页的『示例配置信息』](#)，以获取有关此部分以及如何使用此部分的背景信息。

建立 LU 6.2 连接

描述 LU 6.2 连接所需的参数

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](#)，以及以下联机 HP 文档: [HP-UX SNAplus2 安装指南](#)。

建立 TCP 连接

建立连接的替代方法和后续步骤。

必须先显式启动侦听器，然后才能启动任何通道。它允许接收通道自动启动以响应来自入站发送通道的请求。

或者，如果要使用 UNIX 提供的 TCP/IP 侦听器，请完成以下步骤：

1. 编辑文件 `/etc/services`。

注：要编辑 `/etc/services` 文件，您必须以超级用户或 `root` 用户身份登录。如果该文件中没有以下行，请按如下所示添加该文件：

```
MQSeries      1414/tcp      # MQSeries channel listener
```

2. 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，将 `MQ_INSTALLATION_PATH` 替换为安装了 WebSphere MQ 的高级目录。

```
MQSeries stream tcp nowait root MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta
[-m queue.manager.name]
```

3. 使用以下命令查找 `inetd` 的进程标识：

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

4. 运行以下命令：

```
kill -1 inetd processid
```

注：必须将 `root` 添加到 `mqm` 组。不需要将主组设置为 `mqm`。只要 `mqm` 在组集中，就可以使用这些命令。如果您仅运行使用队列管理器的应用程序，那么无需具有 `mqm` 组权限。

接下来怎么做？

现在已建立连接。您已准备好完成配置。转至 [第 22 页的『IBM WebSphere MQ for HP-UX 配置』](#)。

IBM WebSphere MQ for HP-UX 配置

描述如何定义通道以完成配置。

在开始安装过程之前，请确保首先创建了 `mqm` 用户和组，并设置密码。

使用以下命令启动任何通道:

```
runmqchl -c channel.name
```

注:

1. 样本程序安装在 `MQ_INSTALLATION_PATH/samp` 中, 其中 `MQ_INSTALLATION_PATH` 表示安装 WebSphere MQ 的高级目录。
2. 错误日志存储在 `/var/mqm/qmgrs/qmgrname/errors` 中。
3. 使用命令解释器 `runmqsc` 输入管理命令时, 行尾的 `+` 表示下一行是连续行。确保最后一个参数与连续字符之间有空格。

基本配置

1. 使用以下命令从 UNIX 提示符创建队列管理器:

```
crtmqm -u dlqname -q hpuX
```

其中:

hpuX

是队列管理器的名称

-q

指示这将成为缺省队列管理器

-u dlqname

指定无法传递的消息队列的名称

此命令创建队列管理器 and 一组缺省对象。它设置队列管理器的 DEADQ 属性, 但不会创建无法传递的消息队列。

2. 使用以下命令从 UNIX 提示符启动队列管理器:

```
strmqm hpuX
```

其中, `hpuX` 是创建队列管理器时提供给该队列管理器的名称。

通道配置

包含有关为给定通道和平台配置队列管理器的信息。

以下部分详细说明了要在 HP-UX 队列管理器上执行以实现第 5 页的图 1 中描述的通道的配置。

在每种情况下都会显示 MQSC 命令。从 UNIX 提示符启动 `runmqsc` 并依次输入每个命令, 或者将这些命令构建到命令文件中。

提供了用于连接 WebSphere MQ for HP-UX 和 WebSphere MQ for Windows 的示例。要在另一个平台上连接到 WebSphere MQ, 请使用表中的相应值集来代替 Windows 的值集。

注: **粗体** 中的词是用户指定的, 反映在这些示例中使用的 WebSphere MQ 对象的名称。如果更改此处使用的名称, 请确保在此部分中还更改对这些对象的其他引用。所有其他都是关键字, 应按所示输入。

标识	参数名称	参考	使用的示例	用户值
本地节点的定义				
A	队列管理器名称		HPUX	
B	本地队列名称		HPUX.LOCALQ	
连接到 WebSphere MQ for Windows				
表的此部分中的值必须与第 12 页的表 1 中使用的值匹配, 如所指示。				

表 3: *WebSphere MQ for HP-UX* 的配置工作表 (继续)

标识	参数名称	参考	使用的示例	用户值
C	远程队列管理器名称	A	winnt	
D	远程队列名称		WINNT.REMOTEQ	
E	远程系统上的队列名称	B	WINNT.LOCALQ	
F	传输队列的名称		winnt	
G	发送方 (SNA) 通道名称		HPUX.WINNT.SNA	
H	发送方 (TCP/IP) 通道名称		HPUX.WINNT.TCP	
I	接收方 (SNA) 通道名称	G	WINNT.HPUX.SNA	
J	接收方 (TCP) 通道名称	H	WINNT.HPUX.TCP	
连接到 <i>WebSphere MQ for AIX</i>				
表的此部分中的值必须与 第 18 页的表 2 中使用的值匹配, 如所指示。				
C	远程队列管理器名称	A	AIX	
D	远程队列名称		AIX. 远程队列	
E	远程系统上的队列名称	B	AIX. 位置	
F	传输队列的名称		AIX	
G	发送方 (SNA) 通道名称		HPUX.AIX.SNA	
H	发送方 (TCP) 通道名称		HPUX.AIX.TCP	
I	接收方 (SNA) 通道名称	G	AIX.HPUX.SNA	
J	接收方 (TCP) 通道名称	H	AIX.HPUX.TCP	
连接到 <i>WebSphere MQ for HP Tru64 UNIX</i>				
表的此部分中的值必须与 HP Tru64 UNIX 系统中使用的值相匹配。				
C	远程队列管理器名称	A	DECUX	
D	远程队列名称		DECUX.REMOTEQ	
E	远程系统上的队列名称	B	DECUX.LOCALQ	
F	传输队列的名称		DECUX	
H	发送方 (TCP) 通道名称		DECUX.HPUX.TCP	
J	接收方 (TCP) 通道名称	H	HPUX.DECUX.TCP	
与 <i>WebSphere MQ for Solaris</i> 的连接				
表的此部分中的值必须与 第 29 页的表 4 中使用的值匹配, 如所指示。				
C	远程队列管理器名称	A	Solaris	
D	远程队列名称		SOLARIS.REMOTEQ	
E	远程系统上的队列名称	B	SOLARIS.LOCALQ	
F	传输队列的名称		Solaris	
G	发送方 (SNA) 通道名称		HPUX.SOLARIS.SNA	
H	发送方 (TCP/IP) 通道名称		HPUX.SOLARIS.TCP	

表 3: WebSphere MQ for HP-UX 的配置工作表 (继续)				
标识	参数名称	参考	使用的示例	用户值
I	接收方 (SNA) 通道名称	G	SOLARIS.HPUX.SNA	
J	接收方 (TCP/IP) 通道名称	H	SOLARIS.HPUX.TCP	
连接到 WebSphere MQ for Linux				
表的此部分中的值必须与第 35 页的表 5 中使用的值匹配, 如所指示。				
C	远程队列管理器名称	A	Linux	
D	远程队列名称		LINUX.REMOTEQ	
E	远程系统上的队列名称	B	LINUX.LOCALQ	
F	传输队列的名称		Linux	
G	发送方 (SNA) 通道名称		HPUX.LINUX.SNA	
H	发送方 (TCP/IP) 通道名称		HPUX.LINUX.TCP	
I	接收方 (SNA) 通道名称	G	LINUX.HPUX.SNA	
J	接收方 (TCP/IP) 通道名称	H	LINUX.HPUX.TCP	
连接到 AS400				
C	远程队列管理器名称	A	AS400	
D	远程队列名称		AS400.REMOTEQ	
E	远程系统上的队列名称	B	AS400.LOCALQ	
F	传输队列的名称		AS400	
G	发送方 (SNA) 通道名称		HPUX.AS400.SNA	
H	发送方 (TCP/IP) 通道名称		HPUX.AS400.TCP	
I	接收方 (SNA) 通道名称	G	AS400.HPUX.SNA	
连接到 MVS				
C	远程队列管理器名称	A	MVS	
D	远程队列名称		MVS.REMOTEQ	
E	远程系统上的队列名称	B	MVS.LOCALQ	
F	传输队列的名称		MVS	
G	发送方 (SNA) 通道名称		HPUX.MVS.SNA	
H	发送方 (TCP) 通道名称		HPUX.MVS.TCP	
I	接收方 (SNA) 通道名称	G	MVS。 HPUX.SNA	
连接到 VSE/ESA 的 MQSeries				
表的此部分中的值必须与 VSE/ESA 系统中使用的值匹配。				
C	远程队列管理器名称	A	VSE	
D	远程队列名称		VSE.REMOTEQ	
E	远程系统上的队列名称	B	VSE.LOCALQ	
F	传输队列的名称		VSE	
G	发送方通道名称		HPUX.VSE.SNA	

表 3: WebSphere MQ for HP-UX 的配置工作表 (继续)				
标识	参数名称	参考	使用的示例	用户值
I	接收方通道名称	G	VSE.HPUX.SNA	

IBM WebSphere MQ for HP-UX 发送方通道定义 (使用 SNA)
 示例命令。

```
def ql (WINNT) +                                     F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                             D
  rname(WINNT.LOCALQ) +                             E
  rqmname(WINNT) +                                   C
  xmitq(WINNT) +                                     F
  replace

def chl (HPUX.WINNT.SNA) chltype(sdr) +             G
  trptype(lu62) +
  conname('WINNTCPIC') +                             16
  xmitq(WINNT) +                                     F
  replace
```

IBM WebSphere MQ for HP-UX 接收方-使用 SNA 的通道定义
 示例命令。

```
def ql (HPUX.LOCALQ) replace                         B

def chl (WINNT.HPUX.SNA) chltype(rcvr) +           I
  trptype(lu62) +
  replace
```

IBM WebSphere MQ for HP-UX 可调用 TP 设置
 确保当发送方通道启动对话时正确激活 SNA 接收方通道。

对于 HP SNAplus2 Release 6, 这不是必需的。

在 HP SNAplus2 配置过程中, 您创建了可调用的 TP 定义, 该定义指向可执行文件。在此示例中, 该文件称为 /users/interop/HPUX.crs6a。您可以选择调用此文件的内容, 但请考虑在名称中包含队列管理器的名称。可执行文件的内容必须是:

```
#!/bin/sh
MQ_INSTALLATION_PATH/bin/amqcrs6a -m hpux
```

其中, *hpux* 是队列管理器 A 的名称, *MQ_INSTALLATION_PATH* 是安装了 WebSphere MQ 的高级目录。这确保在发送方通道启动对话时正确激活 SNA 接收方通道。

IBM WebSphere MQ for HP-UX 发送方通道定义 (使用 TCP)
 示例命令。

```
def ql (WINNT) +                                     F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                             D
  rname(WINNT.LOCALQ) +                             E
  rqmname(WINNT) +                                   C
  xmitq(WINNT) +                                     F
  replace

def chl (HPUX.WINNT.TCP) chltype(sdr) +           H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
```

```
xmitq(WINNT) +      F
replace
```

IBM WebSphere MQ for HP-UX 接收方-使用 TCP/IP 的通道定义
示例命令。

```
def q1 (HPUX.LOCALQ) replace      B
def chl (WINNT.HPUX.TCP) chltype(rcvr) +      J
trptype(tcp) +
replace
```

示例配置- IBM WebSphere MQ for Solaris

本部分提供了如何在其他平台上设置从 IBM WebSphere MQ for Solaris 到 IBM WebSphere MQ 产品的通信链路的示例。

在以下平台上提供了示例:

- Windows
- AIX
- HP Tru64 UNIX
- HP-UX
- Linux
- IBM i
- z/OS
- VSE/ESA

请参阅 [第 5 页的『示例配置信息』](#)，以获取有关此部分以及如何使用此部分的背景信息。

使用 SNAP-IX 建立 LU 6.2 连接

用于使用 SNAP-IX 配置 LU 6.2 连接的参数。

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](#)，以下联机 MetaSwitch 文档: [SNAP-IX Administration Guide](#)，以及以下联机 Oracle 文档: [配置系统间通信 \(ISC\)](#)

建立 TCP 连接

有关配置 TCP 连接和后续步骤的信息。

要建立 TCP 连接，请执行以下步骤。

1. 编辑文件 `/etc/services`。

注: 要编辑 `/etc/services` 文件，您必须以超级用户或 `root` 用户身份登录。如果该文件中没有以下行，请按如下所示添加该文件:

```
MQSeries      1414/tcp      # MQSeries channel listener
```

2. 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]
```

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

3. 使用以下命令查找 `inetd` 的进程标识:

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

4. 运行相应的命令，如下所示：

- 对于 Solaris 9:

```
kill -1 inetd processid
```

- 对于 Solaris 10 或更高版本:

```
inetconv
```

接下来怎么做？

现在已建立 TCP/IP 连接。您已准备好完成配置。转至第 28 页的『[IBM WebSphere MQ for Solaris 配置](#)』。

IBM WebSphere MQ for Solaris 配置

描述要定义以完成配置的通道。

在开始安装过程之前，请确保首先创建了 *mqm* 用户和组，并设置密码。

使用以下命令启动任何通道：

```
runmqchl -c channel.name
```

注：

1. 样本程序安装在 *MQ_INSTALLATION_PATH/samp* 中。
MQ_INSTALLATION_PATH 表示安装 WebSphere MQ 的高级目录。
2. 错误日志存储在 */var/mqm/qmgrs/qmgrname/errors* 中。
3. 使用命令解释器 **runmqsc** 输入管理命令时，行尾的 + 表示下一行是连续行。确保最后一个参数与连续字符之间有空格。
4. 对于 SNA 或 LU6.2 通道，如果尝试装入通信库时迁到错误，那么可能找不到文件 *liblu62.so*。此问题的可能解决方案是将其位置（可能是 */opt/SUNWlu62*）添加到 *LD_LIBRARY_PATH*。

基本配置

1. 使用以下命令从 UNIX 提示符创建队列管理器：

```
crtmqm -u dlqname -q solaris
```

其中：

solaris

是队列管理器的名称

-q

指示这将成为缺省队列管理器

-u dlqname

指定无法传递的消息队列的名称

此命令创建队列管理器和一组缺省对象。

2. 使用以下命令从 UNIX 提示符启动队列管理器：

```
strmqm solaris
```

其中，*solaris* 是创建队列管理器时提供给该队列管理器的名称。

通道配置

以下部分详细说明了要在 Solaris 队列管理器上执行以实现通道的配置。

描述的配置是实现第 5 页的图 1 中描述的通道。

显示用于创建每个对象的 MQSC 命令。从 UNIX 提示符启动 **runmqsc** 并依次输入每个命令，或者将这些命令构建到命令文件中。

提供了用于连接 WebSphere MQ for Solaris 和 WebSphere MQ for Windows 的示例。要在另一个平台上连接到 WebSphere MQ，请使用表中的相应值集来代替 Windows 的值集。

注：粗体中的词是用户指定的，反映在这些示例中使用的 WebSphere MQ 对象的名称。如果更改此处使用的名称，请确保在此部分中还更改对这些对象的其他引用。所有其他都是关键字，应按所示输入。

表 4: WebSphere MQ for Solaris 的配置工作表				
标识	参数名称	参考	使用的示例	用户值
本地节点的定义				
A	队列管理器名称		Solaris	
B	本地队列名称		SOLARIS.LOCALQ	
连接到 WebSphere MQ for Windows				
表的此部分中的值必须与第 12 页的表 1 中使用的值匹配，如所指示。				
C	远程队列管理器名称	A	winnt	
D	远程队列名称		WINNT.REMOTEQ	
E	远程系统上的队列名称	B	WINNT.LOCALQ	
F	传输队列的名称		winnt	
G	发送方 (SNA) 通道名称		SOLARIS.WINNT.SNA	
H	发送方 (TCP/IP) 通道名称		SOLARIS.WINNT.TCP	
I	接收方 (SNA) 通道名称	G	WINNT.SOLARIS.SNA	
J	接收方 (TCP) 通道名称	H	WINNT.SOLARIS.TCP	
连接到 WebSphere MQ for AIX				
表的此部分中的值必须与第 18 页的表 2 中使用的值匹配，如所指示。				
C	远程队列管理器名称	A	AIX	
D	远程队列名称		AIX。远程队列	
E	远程系统上的队列名称	B	AIX。位置	
F	传输队列的名称		AIX	
G	发送方 (SNA) 通道名称		SOLARIS.AIX.SNA	
H	发送方 (TCP) 通道名称		SOLARIS.AIX.TCP	
I	接收方 (SNA) 通道名称	G	AIX.SOLARIS.SNA	
J	接收方 (TCP) 通道名称	H	AIX.SOLARIS.TCP	
连接到 MQSeries for Compaq Tru64 Unix				
表的此部分中的值必须与 Compaq Tru64 UNIX 系统中使用的值匹配。				
C	远程队列管理器名称	A	DECUX	
D	远程队列名称		DECUX.REMOTEQ	
E	远程系统上的队列名称	B	DECUX.LOCALQ	

表 4: WebSphere MQ for Solaris 的配置工作表 (继续)

标识	参数名称	参考	使用的示例	用户值
F	传输队列的名称		DECUX	
H	发送方 (TCP) 通道名称		DECUX.SOLARIS.TCP	
J	接收方 (TCP) 通道名称	H	SOLARIS.DECUX.TCP	
连接到 WebSphere MQ for HP-UX				
表的此部分中的值必须与 第 23 页的表 3 中使用的值匹配, 如所指示。				
C	远程队列管理器名称	A	HPUX	
D	远程队列名称		HPUX.REMOTEQ	
E	远程系统上的队列名称	B	HPUX.LOCALQ	
F	传输队列的名称		HPUX	
G	发送方 (SNA) 通道名称		SOLARIS.HPUX.SNA	
H	发送方 (TCP) 通道名称		SOLARIS.HPUX.TCP	
I	接收方 (SNA) 通道名称	G	HPUX.SOLARIS.SNA	
J	接收方 (TCP/IP) 通道名称	H	HPUX.SOLARIS.TCP	
连接到 WebSphere MQ for Linux				
表的此部分中的值必须与 第 35 页的表 5 中使用的值匹配, 如所指示。				
C	远程队列管理器名称	A	Linux	
D	远程队列名称		LINUX.REMOTEQ	
E	远程系统上的队列名称	B	LINUX.LOCALQ	
F	传输队列的名称		Linux	
G	发送方 (SNA) 通道名称		SOLARIS.LINUX.SNA	
H	发送方 (TCP/IP) 通道名称		SOLARIS.LINUX.TCP	
I	接收方 (SNA) 通道名称	G	LINUX.SOLARIS.SNA	
J	接收方 (TCP/IP) 通道名称	H	LINUX.SOLARIS.TCP	
C	远程队列管理器名称	A	AS400	
D	远程队列名称		AS400.REMOTEQ	
E	远程系统上的队列名称	B	AS400.LOCALQ	
F	传输队列的名称		AS400	
G	发送方 (SNA) 通道名称		SOLARIS.AS400.SNA	
H	发送方 (TCP) 通道名称		SOLARIS.AS400.TCP	
I	接收方 (SNA) 通道名称	G	AS400.SOLARIS.SNA	
C	远程队列管理器名称	A	MVS	
D	远程队列名称		MVS.REMOTEQ	
E	远程系统上的队列名称	B	MVS.LOCALQ	

表 4: WebSphere MQ for Solaris 的配置工作表 (继续)				
标识	参数名称	参考	使用的示例	用户值
F	传输队列的名称		MVS	
G	发送方 (SNA) 通道名称		SOLARIS.MVS.SNA	
H	发送方 (TCP) 通道名称		SOLARIS.MVS.TCP	
I	接收方 (SNA) 通道名称	G	MVS。SOLARIS.SNA	
连接到 VSE/ESA 的 MQSeries				
表的此部分中的值必须与 VSE/ESA 系统中使用的值匹配。				
C	远程队列管理器名称	A	VSE	
D	远程队列名称		VSE.REMOTEQ	
E	远程系统上的队列名称	B	VSE.LOCALQ	
F	传输队列的名称		VSE	
G	发送方通道名称		SOLARIS.VSE.SNA	
I	接收方通道名称	G	VSE.SOLARIS.SNA	

使用 *SNAP-IX SNA* 的 *Solaris* 发送方通道定义的 *IBM WebSphere MQ* 示例编码。

```

def ql (WINNT) +                               F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                       D
  rname(WINNT.LOCALQ) +                       E
  rqmname(WINNT) +                             C
  xmitq(WINNT) +                               F
  replace

def chl (SOLARIS.WINNT.SNA) chltype(sdr) +    G
  trptype(lu62) +
  conname('NTCPIC') +                         14
  xmitq(WINNT) +                               F
  replace

```

IBM WebSphere MQ for Solaris 接收方-使用 *SNA* 的通道定义示例编码。

```

def ql (SOLARIS.LOCALQ) replace                B

def chl (WINNT.SOLARIS.SNA) chltype(rcvr) +   I
  trptype(lu62) +
  replace

```

IBM WebSphere MQ for Solaris sender-channel 定义 (使用 *TCP*) 示例编码。

```

def ql (WINNT) +                               F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                       D
  rname(WINNT.LOCALQ) +                       E
  rqmname(WINNT) +                             C
  xmitq(WINNT) +                               F
  replace

```

```
def chl (SOLARIS.WINNT.TCP) chltype(sdr) +          H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(WINNT) +                                    F
  replace
```

IBM WebSphere MQ for Solaris 接收方-使用 TCP/IP 的通道定义示例编码。

```
def ql (SOLARIS.LOCALQ) replace                    B

def chl (WINNT.SOLARIS.TCP) chltype(rcvr) +       J
  trptype(tcp) +
  replace
```

示例配置- IBM WebSphere MQ for Linux

本部分提供了如何在其他平台上设置从 IBM WebSphere MQ 到 IBM WebSphere MQ 产品的通信链接的示例。

提供的示例位于以下平台上:

- Windows
- AIX
- Compaq Tru64 UNIX
- HP-UX
- Solaris
- IBM i
- z/OS
- VSE/ESA

请参阅 [第 5 页的『示例配置信息』](#)，以获取有关此部分以及如何使用此部分的背景信息。

建立 LU 6.2 连接

使用此工作表来记录用于配置的值。

注: 本节中的信息仅适用于 WebSphere MQ for Linux (x86 平台)。它不适用于 WebSphere MQ for Linux (x86-64 平台)，WebSphere MQ for Linux (zSeries s390x 平台) 或 WebSphere MQ for Linux (Power 平台)。

For the latest information about configuring SNA over TCP/IP, refer to the Administration Guide for your version of Linux from the following documentation: [Communications Server for Linux 库](#).

在 Linux 上建立 TCP 连接

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 (INETD)。以下指示信息告诉您如何使用 inet 守护程序或扩展 inet 守护程序建立 TCP 连接。

使用 inet 守护程序 (INETD)

MQ_INSTALLATION_PATH 表示安装 WebSphere MQ 的高级目录。

要建立 TCP 连接，请执行以下步骤。

1. 编辑文件 /etc/services。如果文件中没有以下行，请按如下所示进行添加:

```
MQSeries      1414/tcp      # MQSeries channel listener
```

注: 要编辑此文件，您必须以超级用户或 root 用户身份登录。

2. 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]
```

3. 使用以下命令查找 inetd 的进程标识:

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

4. 运行以下命令:

```
kill -1 inetd processid
```

如果系统上有多个队列管理器，因此需要多个服务，那么必须将每个附加队列管理器的行添加到 `/etc/services` 和 `inetd.conf`。

例如:

```
MQSeries1    1414/tcp
MQSeries2    1822/tcp
```

```
MQSeries1 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM1
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
```

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 [使用 TCP 侦听器储备选项](#)。

Linux 上的 `inetd` 进程可以限制 TCP 端口上的入站连接速率。缺省值为 60 秒时间间隔内的 40 个连接。如果需要更高的速率，请指定 60 秒时间间隔内入站连接数的新限制，方法是在 `inetd.conf` 中追加一个句点 (.)，后跟相应服务的 `nowait` 参数的新限制。例如，对于 60 秒时间间隔内使用的 500 个连接的限制:

```
MQSeries stream tcp nowait.500 mqm /MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM1
```

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

使用扩展的 inet 守护程序 (XINETD)

以下指示信息描述了如何在 Red Hat Linux 上实现扩展的 `inet` 守护程序。如果您正在使用不同的 Linux 分发版，那么可能必须调整这些指示信息。

要建立 TCP 连接，请执行以下步骤。

1. 编辑文件 `/etc/services`。如果文件中没有以下行，请按如下所示进行添加:

```
MQSeries    1414/tcp    # MQSeries channel listener
```

注: 要编辑此文件，您必须以超级用户或 `root` 用户身份登录。

2. 在 XINETD 配置目录 `/etc/xinetd.d` 中创建名为 WebSphere MQ 的文件。将以下节添加到文件中:

```
# WebSphere MQ service for XINETD
service MQSeries
{
    disable           = no
    flags             = REUSE
    socket_type       = stream
    wait              = no
    user              = mqm
    server            = MQ_INSTALLATION_PATH/bin/amqcrsta
    server_args       = -m queue.manager.name
    log_on_failure    += USERID
}
```

3. 通过发出以下命令重新启动扩展的 `inet` 守护程序:

```
/etc/rc.d/init.d/xinetd restart
```

如果系统上有多个队列管理器，因此需要多个服务，那么必须为每个额外的队列管理器向 `/etc/services` 添加一行。您可以在 `/etc/xinetd.d` 目录中为每个服务创建一个文件，也可以向先前创建的 WebSphere MQ 文件添加其他节。

Linux 上的 `xinetd` 进程可以限制 TCP 端口上的入站连接速率。缺省值为 10 秒时间间隔内的 50 个连接。如果需要更高的速率，请通过在 `xinetd` 配置文件中指定 "cps" 属性来指定入站连接速率的新限制。例如，对于 60 秒时间间隔内使用的 500 个连接的限制：

```
cps = 500 60
```

接下来怎么做？

现在已建立 TCP/IP 连接。您已准备好完成配置。转至第 34 页的『[IBM WebSphere MQ for Linux 配置](#)』。

IBM WebSphere MQ for Linux 配置

在开始安装过程之前，请确保首先创建了 `mqm` 用户标识和 `mqm` 组，并设置密码。

使用以下命令启动任何通道：

```
runmqchl -c channel.name
```

注：

1. 样本程序安装在 `MQ_INSTALLATION_PATH/samp` 中，其中 `MQ_INSTALLATION_PATH` 表示安装 WebSphere MQ 的高级目录。
2. 错误日志存储在 `/var/mqm/qmgrs/qmgrname/errors` 中。
3. 使用命令解释器 `runmqsc` 输入管理命令时，行尾的 `+` 表示下一行是连续行。确保最后一个参数与连续字符之间有空格。

基本配置

1. 使用以下命令从 UNIX 提示符创建队列管理器：

```
crtmqm -u dlqname -q linux
```

其中：

linux

是队列管理器的名称

-q

指示这将成为缺省队列管理器

-u dlqname

指定死信队列的名称

此命令创建队列管理器 and 一组缺省对象。

2. 使用以下命令从 UNIX 提示符启动队列管理器：

```
strmqm linux
```

其中，`linux` 是创建队列管理器时提供给该队列管理器的名称。

通道配置

以下部分详细说明了要在 Linux 队列管理器上执行以实现 [第 5 页的图 1](#) 中描述的通道的配置。

显示用于创建每个对象的 MQSC 命令。从 UNIX 提示符启动 `runmqsc` 并依次输入每个命令，或者将这些命令构建到命令文件中。

提供了用于连接 WebSphere MQ for Linux 和 WebSphere MQ for HP-UX 的示例。要在另一个平台上连接到 WebSphere MQ，请使用表中的相应值集来代替 HP-UX 的值集。

注：粗体中的词是用户指定的，反映在这些示例中使用的 WebSphere MQ 对象的名称。如果更改此处使用的名称，请确保在此部分中还更改对这些对象的其他引用。所有其他都是关键字，应按所示输入。

表 5: WebSphere MQ for Linux 的配置工作表				
标识	参数名称	参考	使用的示例	用户值
本地节点的定义				
A	队列管理器名称		Linux	
B	本地队列名称		LINUX.LOCALQ	
连接到 WebSphere MQ for Windows				
表的此部分中的值必须与第 12 页的表 1 中使用的值匹配，如所指示。				
C	远程队列管理器名称	A	winnt	
D	远程队列名称		WINNT.REMOTEQ	
E	远程系统上的队列名称	B	WINNT.LOCALQ	
F	传输队列的名称		winnt	
G	发送方 (SNA) 通道名称		LINUX.WINNT.SNA	
H	发送方 (TCP/IP) 通道名称		LINUX.WINNT.TCP	
I	接收方 (SNA) 通道名称	G	WINNT.LINUX.SNA	
J	接收方 (TCP) 通道名称	H	WINNT.LINUX.TCP	
连接到 WebSphere MQ for AIX				
表的此部分中的值必须与第 18 页的表 2 中使用的值匹配，如所指示。				
C	远程队列管理器名称	A	AIX	
D	远程队列名称		AIX。远程队列	
E	远程系统上的队列名称	B	AIX。位置	
F	传输队列的名称		AIX	
G	发送方 (SNA) 通道名称		LINUX.AIX.SNA	
H	发送方 (TCP) 通道名称		LINUX.AIX.TCP	
I	接收方 (SNA) 通道名称	G	AIX.LINUX.SNA	
J	接收方 (TCP) 通道名称	H	AIX.LINUX.TCP	
连接到 MQSeries for Compaq Tru64 UNIX				
表的此部分中的值必须与 Compaq Tru64 UNIX 系统中使用的值匹配。				
C	远程队列管理器名称	A	DECUX	
D	远程队列名称		DECUX.REMOTEQ	
E	远程系统上的队列名称	B	DECUX.LOCALQ	
F	传输队列的名称		DECUX	
H	发送方 (TCP) 通道名称		DECUX.LINUX.TCP	
J	接收方 (TCP) 通道名称	H	LINUX.DECUX.TCP	

表 5: WebSphere MQ for Linux 的配置工作表 (继续)

标识	参数名称	参考	使用的示例	用户值
连接到 WebSphere MQ for HP-UX				
表的此部分中的值必须与 第 23 页的表 3 中使用的值匹配, 如所指示。				
C	远程队列管理器名称	A	HPUX	
D	远程队列名称		HPUX.REMOTEQ	
E	远程系统上的队列名称	B	HPUX.LOCALQ	
F	传输队列的名称		HPUX	
G	发送方 (SNA) 通道名称		LINUX.HPUX.SNA	
H	发送方 (TCP) 通道名称		LINUX.HPUX.TCP	
I	接收方 (SNA) 通道名称	G	HPUX.LINUX.SNA	
J	接收方 (TCP/IP) 通道名称	H	HPUX.LINUX.TCP	
与 WebSphere MQ for Solaris 的连接				
表的此部分中的值必须与 第 29 页的表 4 中使用的值匹配, 如所指示。				
C	远程队列管理器名称	A	Solaris	
D	远程队列名称		SOLARIS.REMOTEQ	
E	远程系统上的队列名称	B	SOLARIS.LOCALQ	
F	传输队列的名称		GIS	
G	发送方 (SNA) 通道名称		LINUX.SOLARIS.SNA	
H	发送方 (TCP/IP) 通道名称		LINUX.SOLARIS.TCP	
I	接收方 (SNA) 通道名称	G	SOLARIS.LINUX.SNA	
J	接收方 (TCP/IP) 通道名称	H	SOLARIS.LINUX.TCP	
C	远程队列管理器名称	A	AS400	
D	远程队列名称		AS400.REMOTEQ	
E	远程系统上的队列名称	B	AS400.LOCALQ	
F	传输队列的名称		AS400	
G	发送方 (SNA) 通道名称		LINUX.AS400.SNA	
H	发送方 (TCP) 通道名称		LINUX.AS400.TCP	
I	接收方 (SNA) 通道名称	G	AS400.LINUX.SNA	
C	远程队列管理器名称	A	MVS	
D	远程队列名称		MVS.REMOTEQ	
E	远程系统上的队列名称	B	MVS.LOCALQ	
F	传输队列的名称		MVS	
G	发送方 (SNA) 通道名称		LINUX.MVS.SNA	
H	发送方 (TCP) 通道名称		LINUX.MVS.TCP	

表 5: WebSphere MQ for Linux 的配置工作表 (继续)				
标识	参数名称	参考	使用的示例	用户值
I	接收方 (SNA) 通道名称	G	MVS。 LINUX.SNA	
连接到 MQSeries for VSE/ESA (仅限 WebSphere MQ for Linux (x86 平台)) 表的此部分中的值必须与 VSE/ESA 系统中使用的值匹配。				
C	远程队列管理器名称	A	VSE	
D	远程队列名称		VSE.REMOTEQ	
E	远程系统上的队列名称	B	VSE.LOCALQ	
F	传输队列的名称		VSE	
G	发送方通道名称		LINUX.VSE.SNA	
I	接收方通道名称	G	VSE.LINUX.SNA	

IBM WebSphere MQ for Linux (x86 平台) 发送方通道定义 (使用 SNA)
 示例编码。

```

def ql (HPUX) +                                     F
  usage(xmitq) +
  replace

def qr (HPUX.REMOTEQ) +                             D
  rname(HPUX.LOCALQ) +                             E
  rqmname(HPUX) +                                   C
  xmitq(HPUX) +                                     F
  replace

def chl (LINUX.HPUX.SNA) chltype(sdr) +             G
  trptype(lu62) +
  conname('HPUXCPIC') +                             14
  xmitq(HPUX) +                                     F
  replace
  
```

IBM WebSphere MQ for Linux (x86 平台) 使用 SNA 的接收方通道定义
 示例编码。

```

def ql (LINUX.LOCALQ) replace                       B

def chl (HPUX.LINUX.SNA) chltype(rcvr) +           I
  trptype(lu62) +
  replace
  
```

使用 TCP 的 Linux 发送方通道定义的 IBM WebSphere MQ
 示例编码。

```

def ql (HPUX) +                                     F
  usage(xmitq) +
  replace

def qr (HPUX.REMOTEQ) +                             D
  rname(HPUX.LOCALQ) +                             E
  rqmname(HPUX) +                                   C
  xmitq(HPUX) +                                     F
  replace

def chl (LINUX.HPUX.TCP) chltype(sdr) +            H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(HPUX) +                                     F
  replace
  
```

IBM WebSphere MQ for Linux 接收方-使用 TCP/IP 的通道定义
示例编码。

```
def ql (LINUX.LOCALQ) replace B
◆
def chl (HPUX.LINUX.TCP) chltype(rcvr) + J
  trptype(tcp) +
  replace
```

队列名称

使用此信息可了解队列名称和保留队列名称的限制。

队列的名称长度最多可以为 48 个字符。

保留队列名称

将为队列管理器定义的队列保留以“SYSTEM.”开头的名称。您可以使用 **ALTER** 或 **DEFINE REPLACE** 命令来更改这些队列定义以适合您的安装。为 IBM WebSphere MQ 定义了以下名称：

队列名称	描述
SYSTEM.ADMIN.ACTIVITY.QUEUE	活动报告的队列
SYSTEM.ADMIN.CHANNEL.EVENT	通道事件的队列
SYSTEM.ADMIN.COMMAND.EVENT	命令事件的队列
SYSTEM.ADMIN.COMMAND.QUEUE	将 PCF 命令消息发送到的队列
SYSTEM.ADMIN.CONFIG.EVENT	配置事件的队列
SYSTEM.ADMIN.PERFM.EVENT	性能事件的队列
SYSTEM.ADMIN.PUBSUB.EVENT	系统发布/预订相关事件队列
SYSTEM.ADMIN.QMGR.EVENT	队列管理器事件的队列
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	跟踪路由应答消息的队列
SYSTEM.AUTH.DATA.QUEUE	用于保存队列管理器的访问控制表的队列。(不适用于 z/OS)
SYSTEM.CHANNEL.INITQ	通道的启动队列
SYSTEM.CHANNEL.SYNCQ	保存通道同步数据的队列
SYSTEM.CHLAUTH.DATA.QUEUE	IBM WebSphere MQ 通道认证数据队列
SYSTEM.CICS.INITIATION.QUEUE	用于触发的队列 (不适用于 z/OS)
SYSTEM.CLUSTER.COMMAND.QUEUE	用于在队列管理器之间通信存储库更改的队列 (仅限 AIX, HP-UX, Linux, IBM i, Solaris, Windows 和 z/OS)
SYSTEM.CLUSTER.HISTORY.QUEUE	此队列用于存储集群状态信息的历史记录以用于服务目的。
SYSTEM.CLUSTER.REPOSITORY.QUEUE	用于保存有关存储库的信息的队列 (仅限 AIX, HP-UX, Linux, IBM i, Solaris, Windows 和 z/OS)
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	该队列用于为每个集群发送方通道创建单独的传输队列。
SYSTEM.CLUSTER.TRANSMIT.QUEUE	集群支持所管理的所有目标的传输队列 (仅限 AIX, HP-UX, Linux, IBM i, Solaris, Windows 和 z/OS)
SYSTEM.COMMAND.INPUT	在 z/OS 上向其发送命令消息的队列

队列名称	描述
SYSTEM.COMMAND.REPLY.MODEL	命令应答的模型队列定义 (针对 z/OS)
SYSTEM.DEAD.LETTER.QUEUE	死信队列 (不适用于 z/OS)
SYSTEM.DEFAULT.ALIAS.QUEUE	缺省别名队列定义
SYSTEM.DEFAULT.INITIATION.QUEUE	用于触发指定进程的队列 (不适用于 z/OS)
SYSTEM.DEFAULT.LOCAL.QUEUE	缺省本地队列定义
SYSTEM.DEFAULT.MODEL.QUEUE	缺省模型队列定义
SYSTEM.DEFAULT.REMOTE.QUEUE	缺省远程队列定义
SYSTEM.DURABLE.SUBSCRIBER.QUEUE	用于在队列管理器中保存持久预订的持久副本的本地队列
SYSTEM.HIERARCHY.STATE	用于保存有关发布/预订层次结构中的队列间管理器关系状态的信息的队列
SYSTEM.JMS.TEMPQ.MODEL	JMS 临时队列的模型
SYSTEM.INTERNAL.REPLY.QUEUE	IBM WebSphere MQ 内部应答队列 (不适用于 z/OS)
SYSTEM.INTER.QMGR.CONTROL	在发布/预订层次结构中用于接收来自远程队列管理器的请求以创建代理预订的队列
SYSTEM.INTER.QMGR.PUBS	在发布/预订层次结构中用于从远程队列管理器接收发布的队列
SYSTEM.INTER.QMGR.FANREQ	在发布/预订层次结构中用于处理请求以在远程队列管理器上创建代理预订的队列
SYSTEM.MQEXPLORER.REPLY.MODEL	IBM WebSphere MQ Explorer 回复的模型队列定义
SYSTEM.MQSC.REPLY.QUEUE	MQSC 命令应答的模型队列定义 (不适用于 z/OS)
SYSTEM.QSG.CHANNEL.SYNCQ	用于存储包含共享通道同步信息的消息的共享本地队列 (仅限 z/OS)
SYSTEM.QSG.TRANSMIT.QUEUE	在同一队列共享组中的队列管理器之间传输消息时, 组内排队代理所使用的共享本地队列 (仅限 z/OS)
SYSTEM.RETAINED.PUB.QUEUE	用于在队列管理器中保存每个保留发布的副本的本地队列。
SYSTEM.SELECTION.EVALUATION.QUEUE	IBM WebSphere MQ 内部选择评估队列 (不适用于 z/OS)
SYSTEM.SELECTION.VALIDATION.QUEUE	IBM WebSphere MQ 内部选择验证队列 (不适用于 z/OS)

其他对象名

进程, 名称列表, 集群, 主题, 服务和认证信息对象的名称长度最多可以为 48 个字符。通道的名称长度最多可以为 20 个字符。存储类的名称长度最多可以为 8 个字符。CF 结构的名称长度最多可以为 12 个字符。

保留对象名

以 SYSTEM 开头的名称。为队列管理器定义的对象保留。可以使用 ALTER 或 DEFINE REPLACE 命令来更改这些对象定义以适合您的安装。为 IBM WebSphere MQ 定义了以下名称:

对象名称	描述
SYSTEM.ADMIN.SVRCONN	用于队列管理器远程管理的服务器连接通道

对象名称	描述
SYSTEM.AUTO.RECEIVER	自动定义的缺省接收方通道 (仅限 Windows 和 UNIX and Linux 系统)
SYSTEM.AUTO.SVRCONN	用于自动定义的缺省服务器连接通道 (仅适用于 IBM i, Windows 和 UNIX and Linux 系统)
SYSTEM.BASE.TOPIC	ASPARENT 解析的基本主题。如果特定管理主题对象没有父管理主题对象, 那么将从此对象继承任何 ASPARENT 属性
SYSTEM.DEF.CLNTCONN	缺省客户机连接通道定义
SYSTEM.DEF.CLUSRCVR	缺省集群接收方通道定义
SYSTEM.DEF.CLUSSDR	缺省集群发送方通道定义
SYSTEM.DEF.RECEIVER	缺省接收方通道定义
SYSTEM.DEF.REQUESTER	缺省请求者通道定义
SYSTEM.DEF.SENDER	缺省发送方通道定义
SYSTEM.DEF.SERVER	缺省服务器通道定义
SYSTEM.DEF.SVRCONN	缺省服务器连接通道定义
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	用于定义类型为 CRLLDAP 的认证信息对象的缺省认证信息对象定义
SYSTEM.DEFAULT.AUTHINFO.OCSP	用于定义 OCSP 类型的认证信息对象的缺省认证信息对象定义
SYSTEM.DEFAULT.LISTENER.LU62	缺省 SNA 侦听器 (仅限 Windows)
SYSTEM.DEFAULT.LISTENER.NETBIOS	缺省 NetBIOS 侦听器 (仅限 Windows)
SYSTEM.DEFAULT.LISTENER.SPX	缺省 SPX 侦听器 (仅限 Windows)
SYSTEM.DEFAULT.LISTENER.TCP	缺省 TCP/IP 侦听器 (仅限 IBM i, Windows 和 UNIX and Linux 系统)
SYSTEM.DEFAULT.NAMELIST	缺省名称列表定义
SYSTEM.DEFAULT.PROCESS	缺省进程定义
SYSTEM.DEFAULT.SERVICE	缺省服务 (仅限 IBM i, Windows 和 UNIX and Linux 系统)
SYSTEM.DEFAULT.TOPIC	缺省主题定义
SYSTEM.QPUBSUB.QUEUE.NAMELIST	要监视的已排队发布/预订接口的队列列表
SYSTEMST	缺省存储类定义 (仅限 z/OS)

队列名称解析

本主题包含有关队列管理器在通道的发送端和接收端执行的队列名称解析的信息。

在更大的网络中, 使用队列管理器比其他形式的通信具有许多优点。这些优点源自 DQM 中的名称解析函数, 主要优点如下:

- 应用程序不需要进行路由决策
- 应用程序不需要知道网络结构
- 网络链接由系统管理员创建
- 网络结构由网络规划者控制

- 可以在节点之间使用多个通道来分区流量

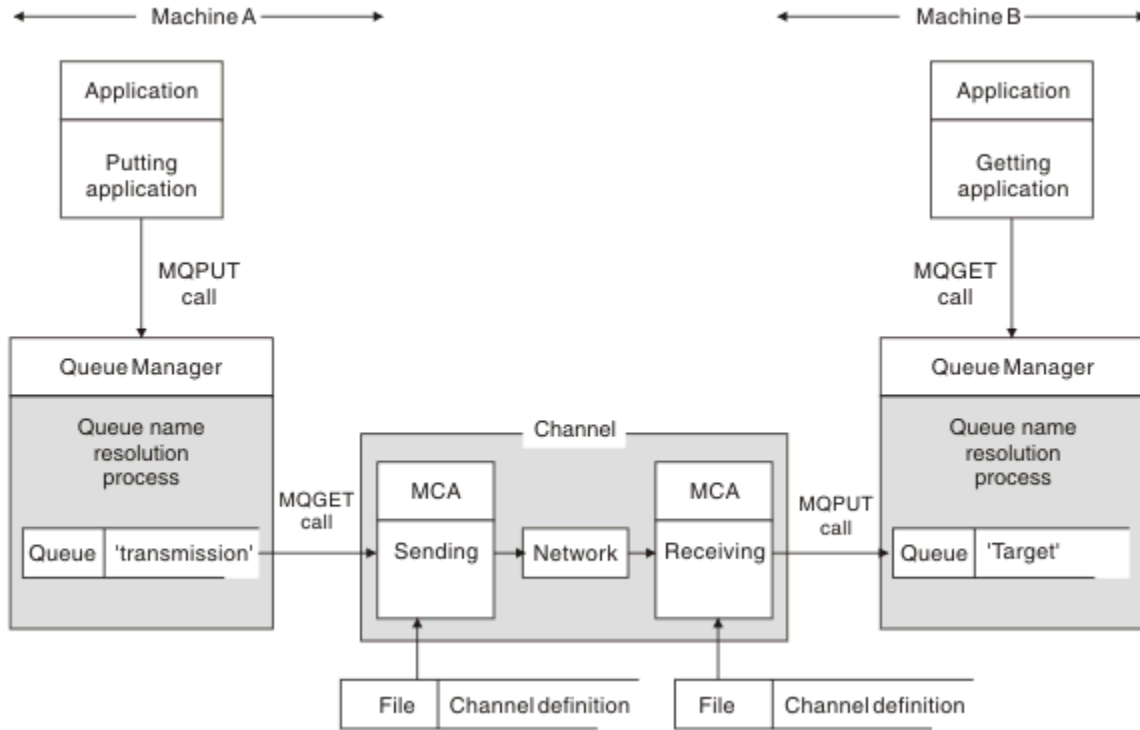


图 2: 名称解析

对于第 41 页的图 2，对于应用程序而言，用于将消息放入远程队列的基本机制与用于将消息放入本地队列的基本机制相同：

- 放入消息的应用程序发出 MQOPEN 和 MQPUT 调用以将消息放入目标队列。
- 获取消息的应用程序发出 MQOPEN 和 MQGET 调用以从目标队列获取消息。

如果两个应用程序都连接到同一个队列管理器，那么不需要队列管理器间通信，并且目标队列对这两个应用程序都描述为本地。

但是，如果应用程序连接到不同的队列管理器，那么在传输中涉及到两个 MCA 及其关联的网络连接，如图所示。在这种情况下，目标队列被视为放置应用程序的远程队列。

事件顺序如下所示：

1. 放置应用程序发出 MQOPEN 和 MQPUT 调用以将消息放置到目标队列。
2. 在 MQOPEN 调用期间，*name resolution* 函数检测到目标队列不是本地队列，并决定适合哪个传输队列。此后，在与 MQOPEN 调用关联的 MQPUT 调用上，所有消息都放置在此传输队列上。
3. 发送 MCA 从传输队列中获取消息，并将它们传递到远程计算机上的接收 MCA。
4. 接收 MCA 将消息放在一个或多个目标队列上。
5. 获取应用程序发出 MQOPEN 和 MQGET 调用以从目标队列获取消息。

注：只有步骤 1 和步骤 5 涉及应用程序代码；步骤 2 到 4 由本地队列管理器和 MCA 程序执行。放置应用程序不知道目标队列的位置，该队列可能位于同一处理器中，也可能位于另一个大陆上的另一个处理器中。

发送 MCA，网络连接和接收 MCA 的组合称为消息通道，本质上是单向设备。正常情况下，需要双向移动消息，为此移动设置了两个通道，每个方向一个通道。

什么是队列名称解析？

队列名称解析对于 DQM 至关重要。它使应用程序不再需要关注队列的物理位置，并使它们与网络的详细信息相隔离。

系统管理员可以将队列从一个队列管理器移至另一个队列管理器，并在无需应用程序了解任何信息的情况下更改队列管理器之间的路由。

为了从应用程序设计中解耦数据传递的确切路径，需要在应用程序引用目标队列时引入应用程序所使用的名称与发生流的通道的命名之间的 `indirection` 级别。此双向是使用队列名称解析机制实现的。

实质上，当应用程序引用队列名称时，解析机制会将该名称映射到传输队列或非传输队列的本地队列。对于到传输队列的映射，需要在目标处进行第二个名称解析，并且将接收到的消息按应用程序设计者的期望放在目标队列上。应用程序仍然不知道用于移动消息的传输队列和通道。

注：队列和通道的定义是系统管理责任，可以由操作员或系统管理实用程序进行更改，而无需更改应用程序。

消息流的系统管理的一个重要要求是需要队列管理器之间提供备用路径。例如，业务需求可能指示通过不同通道将不同的服务类发送到同一目标。此决策是系统管理决策，队列名称解析机制提供了实现此决策的灵活方法。The Application Programming Guide describes this in detail, but the basic idea is to use queue name resolution at the sending queue manager to map the queue name supplied by the application to the appropriate transmission queue for the type of traffic involved. 同样在接收端，队列名称解析将消息描述符中的名称映射到本地（非传输）队列或再次映射到相应的传输队列。

不仅可以将从一个队列管理器到另一个队列管理器的正向路径分区为不同类型的流量，而且发送到出站消息中的应答队列定义的返回消息也可以使用相同的流量分区。队列名称解析满足此要求，并且应用程序设计者无需参与这些流量分区决策。

在发送和接收队列管理器上执行映射的点是名称解析工作方式的一个重要方面。此映射允许将放置应用程序提供的队列名称映射到发送队列管理器上的本地队列或传输队列，并再次重新映射到接收队列管理器上的本地队列或传输队列。

来自接收应用程序或 MCA 的应答消息具有以相同方式执行的名称解析，允许使用路由上所有队列管理器中的队列定义在特定路径上进行返回路由。

系统和缺省对象

列出由 `crtmqm` 命令创建的系统对象和缺省对象。

使用 `crtmqm` 控制命令创建队列管理器时，将自动创建系统对象和缺省对象。

- 系统对象是操作队列管理器或通道所需的那些 IBM WebSphere MQ 对象。
- 缺省对象定义对象的所有属性。创建对象（例如本地队列）时，未显式指定的任何属性都将从缺省对象继承。

下表列出了 `crtmqm` 创建的系统对象和缺省对象：

- 第 42 页的表 6 列出了系统和缺省队列对象。
- 第 44 页的表 7 列出了系统和缺省主题对象。
- 第 44 页的表 8 列出了系统和缺省通道对象。
- 第 44 页的表 9 列出了系统和缺省认证信息对象。
- 第 44 页的表 10 列出了系统和缺省侦听器对象。
- 第 44 页的表 11 列出了系统和缺省名称列表对象。
- 第 45 页的表 12 列出了系统和缺省流程对象。
- 第 45 页的表 13 列出了系统和缺省服务对象。

对象名	描述
SYSTEM.ADMIN.ACCOUNTING.QUEUE	用于保存记帐监视数据的队列。
SYSTEM.ADMIN.ACTIVITY.QUEUE	用于保存返回的活动报告的队列。
SYSTEM.ADMIN.CHANNEL.EVENT	通道的事件队列。

表 6: 系统和缺省对象: 队列 (继续)

对象名	描述
SYSTEM.ADMIN.COMMAND.EVENT	命令事件的事件队列。
SYSTEM.ADMIN.COMMAND.QUEUE	管理命令队列。用于远程 MQSC 命令和 PCF 命令。
SYSTEM.ADMIN.CONFIG.EVENT	配置事件的事件队列。
SYSTEM.ADMIN.PERFM.EVENT	性能事件的事件队列。
SYSTEM.ADMIN.PUBSUB.EVENT	系统发布/预订相关事件队列
SYSTEM.ADMIN.QMGR.EVENT	队列管理器事件的事件队列。
SYSTEM.ADMIN.STATISTICS.QUEUE	用于保存统计信息监视数据的队列。
SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE	显示跟踪活动的队列。
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	用于保存返回的跟踪路由应答消息的队列。
SYSTEM.AUTH.DATA.QUEUE	用于保存队列管理器的访问控制表的队列。
SYSTEM.CHANNEL.INITQ	通道启动队列。
SYSTEM.CHANNEL.SYNCQ	用于保存通道的同步数据的队列。
SYSTEM.CHLAUTH.DATA.QUEUE	IBM WebSphere MQ 通道认证数据队列
SYSTEM.CICS.INITIATION.QUEUE	缺省 CICS 启动队列。
SYSTEM.CLUSTER.COMMAND.QUEUE	用于将消息传递到存储库队列管理器的队列。
SYSTEM.CLUSTER.HISTORY.QUEUE	此队列用于存储集群状态信息的历史记录以用于服务目的。
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	该队列用于为每个集群发送方通道创建单独的传输队列。
SYSTEM.CLUSTER.REPOSITORY.QUEUE	用于存储所有存储库信息的队列。
SYSTEM.CLUSTER.TRANSMIT.QUEUE	所有消息到所有集群的传输队列。
SYSTEM.DEAD.LETTER.QUEUE	死信 (undelivered-message) 队列。
SYSTEM.DEFAULT.ALIAS.QUEUE	缺省别名队列。
SYSTEM.DEFAULT.INITIATION.QUEUE	缺省启动队列。
SYSTEM.DEFAULT.LOCAL.QUEUE	缺省本地队列。
SYSTEM.DEFAULT.MODEL.QUEUE	缺省模型队列。
SYSTEM.DEFAULT.REMOTE.QUEUE	缺省远程队列。
SYSTEM.JMS.TEMPQ.MODEL	JMS 临时队列的模型
SYSTEM.MQEXPLORER.REPLY.MODEL	IBM WebSphere MQ Explorer 应答队列。这是一个模型队列，用于为对 IBM WebSphere MQ Explorer 的应答创建临时动态队列。
SYSTEM.MQSC.REPLY.QUEUE	MQSC 命令应答队列。这是一个模型队列，用于为远程 MQSC 命令的应答创建临时动态队列。
SYSTEM.PENDING.DATA.QUEUE	支持 JMS 中的延迟消息。

表 7: 系统和缺省对象: 主题	
对象名	描述
SYSTEM.BASE.TOPIC	ASPARENT 解析的基本主题。如果特定主题没有父管理主题对象, 或者这些父对象也具有 ASPARENT, 那么将从此对象继承任何剩余的 ASPARENT 属性。
SYSTEM.DEFAULT.TOPIC	缺省主题定义。

表 8: 系统和缺省对象: 通道	
对象名	描述
SYSTEM.AUTO.RECEIVER	动态接收方通道。
SYSTEM.AUTO.SVRCONN	动态服务器连接通道。
SYSTEM.DEF.CLUSRCVR	集群的缺省接收方通道, 用于为在集群中的队列管理器上创建 CLUSRCVR 通道时未指定的任何属性提供缺省值。
SYSTEM.DEF.CLUSSDR	集群的缺省发送方通道, 用于提供在集群中的队列管理器上创建 CLUSSDR 通道时未指定的任何属性的缺省值。
SYSTEM.DEF.RECEIVER	缺省接收方通道。
SYSTEM.DEF.REQUESTER	缺省请求者通道。
SYSTEM.DEF.SENDER	缺省发送方通道。
SYSTEM.DEF.SERVER	缺省服务器通道。
SYSTEM.DEF.SVRCONN	缺省服务器连接通道。
SYSTEM.DEF.CLNTCONN	缺省客户机连接通道。

表 9: 系统对象和缺省对象: 认证信息对象	
对象名	描述
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	用于定义类型为 CRLLDAP 的认证信息对象的缺省认证信息对象。
SYSTEM.DEFAULT.AUTHINFO.OCSP	用于定义 OCSP 类型的认证信息对象的缺省认证信息对象。

表 10: 系统和缺省对象: 侦听器	
对象名	描述
SYSTEM.DEFAULT.LISTENER.TCP	缺省 TCP 侦听器。
SYSTEM.DEFAULT.LISTENER.LU62 ¹	缺省 LU62 侦听器。
SYSTEM.DEFAULT.LISTENER.NETBIOS ¹	缺省 NETBIOS 侦听器。
SYSTEM.DEFAULT.LISTENER.SPX ¹	缺省 SPX 侦听器。

1. 仅适用于 Windows

表 11: 系统对象和缺省对象: 名称列表	
对象名	描述
SYSTEM.DEFAULT.NAMELIST	缺省名称列表。

表 12: 系统和缺省对象: 进程	
对象名	描述
SYSTEM.DEFAULT.PROCESS	缺省进程定义。

表 13: 系统和缺省对象: 服务	
对象名	描述
SYSTEM.DEFAULT.SERVICE	缺省服务。
SYSTEM.BROKER	发布/预订代理程序

Windows 缺省配置对象

在 Windows 系统上, 可以使用 WebSphere MQ Postcard 应用程序来设置缺省配置。

注: 如果计算机上存在其他队列管理器, 那么无法设置缺省配置。

用于 Windows 缺省配置对象的许多名称都涉及使用短 TCP/IP 名称。这是没有域部分的计算机的 TCP/IP 名称; 例如, 计算机 mycomputer.hursley.ibm.com 的短 TCP/IP 名称为 mycomputer。在所有必须截断此名称的情况下, 如果最后一个字符是句点 (.), 那么将除去此名称。

短 TCP/IP 名称中对 WebSphere MQ 对象名 (例如连字符) 无效的任何字符都将替换为下划线字符。

WebSphere MQ 对象名的有效字符包括 :a 到 z, A 到 Z, 0 到 9 以及四个特殊字符 /%。和 _。

Windows 缺省配置的集群名称为 DEFAULT_CLUSTER。

如果队列管理器不是存储库队列管理器, 那么将创建 第 45 页的表 14 中列出的对象。

表 14: 由 Windows 缺省配置应用程序创建的对象	
Object	名称
队列管理器	以字符 QM_ 为前缀的简短 TCP/IP 名称。队列管理器名称的最大长度为 48 个字符。超过此限制的名称将截断为 48 个字符。如果名称的最后一个字符是句点 (.), 那么这将替换为空格 ()。 队列管理器具有与之关联的命令服务器, 通道侦听器 and 通道启动程序。通道侦听器侦听标准 WebSphere MQ 端口, 端口号为 1414。当缺省配置队列管理器仍然存在时, 在此机器上创建的任何其他队列管理器都不得使用端口 1414。
通用集群接收方通道	以字符 TO_QM_ 为前缀的简短 TCP/IP 名称。通用集群接收方名称的最大长度为 20 个字符。超过此限制的名称将截断为 20 个字符。如果名称的最后一个字符是句点 (.), 那么这将替换为空格 ()。
集群发送方通道	集群发送方通道最初是使用名称 TO_ + QMNAME + 创建的。一旦 WebSphere MQ 与缺省配置集群的存储库队列管理器建立了连接, 此名称将替换为缺省配置集群的存储库队列管理器的名称 (以字符 TO_ 作为前缀)。集群发送方通道名称的最大长度为 20 个字符。超过此限制的名称将截断为 20 个字符。如果名称的最后一个字符是句点 (.), 那么这将替换为空格 ()。
本地消息队列	本地消息队列称为 default。
供 WebSphere MQ Postcard 应用程序使用的本地消息队列	供 WebSphere MQ Postcard 应用程序使用的本地消息队列称为 postcard。
服务器连接通道	服务器连接通道允许客户机连接到队列管理器。其名称是短 TCP/IP 名称, 前缀为字符 S_。服务器连接通道名称的最大长度为 20 个字符。超过此限制的名称将截断为 20 个字符。如果名称的最后一个字符是句点 (.), 那么这将替换为空格 ()。

如果队列管理器是存储库队列管理器，那么缺省配置类似于 [第 45 页的表 14](#) 中描述的配置，但存在以下差异：

- 队列管理器定义为缺省配置集群的存储库队列管理器。
- 未定义集群发送方通道。
- 将创建以字符 `clq_default_` 作为前缀的短 TCP/IP 名称的本地集群队列。此名称的最大长度为 48 个字符。超过此长度的名称将在 48 个字符处截断。

如果您请求远程管理设施，那么服务器连接通道 `SYSTEM.ADMIN.SVRCONN`。

SYSTEM.BASE.TOPIC

ASPARENT 解析的基本主题。如果特定主题没有父管理主题对象，或者这些父对象也具有 ASPARENT，那么将从此对象继承任何剩余的 ASPARENT 属性。

表 15: SYSTEM.BASE.TOPIC 的缺省值	
参数	值
TOPICSTR	"
cluster	缺省值为空字符串。
COMMINFO	SYSTEM.DEFAULT.COMMINFO.MULTICAST
DEFPRESP	SYNC
DEFPRTY	0
DEFPSIST	NO
DESCR	"用于解析属性的基本主题"
DURSUB	YES
MCAST	DISABLED
MDURMDL	SYSTEM.DURABLE.MODEL.QUEUE
MNDURMDL	SYSTEM.NDURABLE.MODEL.QUEUE
NPMSGDLV	ALLAVAIL
PMSGDLV	ALLDUR
PROXYSUB	FIRSTUSE
PUB	ENABLED
PUBSCOPE	ALL
SUB	ENABLED
SUBSCOPE	ALL
USEDLQ	YES
WILDCARD	PASSTHRU

如果此对象不存在，那么其缺省值仍由 IBM WebSphere MQ 用于 ASPARENT 属性，而这些属性未由主题树上的父主题解析。

将 `SYSTEM.BASE.TOPIC` 的 `PUB` 或 `SUB` 属性设置为 `DISABLED` 会阻止应用程序发布或预订主题树中的主题，但有两个例外：

1. 主题树中 `PUB` 或 `SUB` 显式设置为 `ENABLE` 的任何主题对象。应用程序可以发布或预订这些主题及其子代。

2. 通过将 SYSTEM.BASE.TOPIC 的 PUB 或 SUB 属性设置为 DISABLED, 不会禁用对 SYSTEM.BROKER.ADMIN.STREAM 的发布和预订。

节信息

以下信息可帮助您在节中配置信息, 并列出了 `mqs.ini`, `qm.ini` 和 `mqclient.ini` 文件的内容。

配置节

使用链接可帮助您在企业中配置一个或多个系统:

- [更改 IBM WebSphere MQ 配置信息](#) 可帮助您配置:

- `AllQueueManagers` 节
- `DefaultQueueManager` 节
- `ExitProperties` 节
- `LogDefaults` 节
- `qm.ini` 文件中的 `Security` 节

- [更改队列管理器配置信息](#) 可帮助您配置:

- `AccessMode` 节 (仅限 Windows)
- `Service` 节-用于可安装服务
- `Log` 节
- `RestrictedMode` 节 (仅限 UNIX and Linux 系统)
- `XAResourceManager` 节
- `TCP`, `LU62` 和 `NETBIOS` 节
- `ExitPath` 节
- `QMErrorLog` 节
- `SSL` 节
- `ExitPropertiesLocal` 节

- [配置服务和组件](#) 可帮助您配置:

- `Service` 节
- `ServiceComponent` 节

并包含有关它们如何用于 UNIX and Linux 和 Windows 平台上的不同服务的链接。

- [配置 API 出口](#) 可帮助您配置以下内容:

- `AllActivityTrace` 节
- `ApplicationTrace` 节

- [配置活动跟踪行为](#) 可帮助您配置:

- `ApiExitCommon` 节
- `ApiExitTemplate` 节
- `ApiExitLocal` 节

- [客户机的配置信息](#) 可帮助您配置:

- `CHANNELS` 节
- `ClientExitPath` 节
- `LU62`, `NETBIOS` 和 `SPX` 节 (仅限 Windows)
- `MessageBuffer` 节
- `SSL` 节

- [TCP](#) 节
 - [第 49 页的『用于分布式排队的配置文件节』](#) 可帮助您配置:
 - [CHANNELS](#) 节
 - [TCP](#) 节
 - [LU62](#) 节
 - [NETBIOS](#)
 - [ExitPath](#) 节
 - [设置已排队的发布/预订消息属性](#) 可帮助您配置:
 - [PersistentPublish](#) 重试 属性
 - [NonPersistentPublishRetry](#) 属性
 - [PublishBatch](#) 大小 属性
 - [PublishRetry](#) 时间间隔 属性
- 在 [Broker](#) 节中。



注意: 如果需要 [Broker](#) 节, 那么必须创建一个节。

配置文件

请参阅:

- [mqs.ini](#) 文件
- [qm.ini](#) 文件
- [mqclient.ini](#) 文件

以获取每个配置文件中可能的节的列表。

[mqs.ini](#) 文件

[UNIX and Linux 系统的 IBM WebSphere MQ 配置文件示例](#) 显示了示例 [mqs.ini](#) 文件。

[mqs.ini](#) 文件可以包含以下节:

- [AllQueue](#) 管理器
- [DefaultQueue](#) 管理器
- [ExitProperties](#)
- [LogDefaults](#)

此外, 每个队列管理器都有一个 [QueueManager](#) 节。

[qm.ini](#) 文件

[针对 UNIX and Linux 系统的 IBM WebSphere MQ 的示例队列管理器配置文件](#) 显示了示例 [qm.ini](#) 文件。

[qm.ini](#) 文件可以包含以下节:

- [ExitPath](#)
- [日志](#)
- [QMErrorLog](#)
- [QueueManager](#)
- [安全性](#)
- [服务](#) 和 [ServiceComponent](#)

要配置 [InstallableServices](#):

- 在 UNIX and Linux 平台上, 使用 [服务](#) 和 [ServiceComponent](#) 节。

- 在 Windows 上, 使用 **regedit**。

- DefaultBind 类型的连接



注意: 如果需要 *Connection* 节, 那么必须创建一个节。

- SSL 和 TLS
- TCP, LU62 和 NETBIOS
- XAResourceManager

此外, 您可以使用 crtmqm 命令来更改以下属性:

- *AccessMode* (仅限 Windows)
- *RestrictedMode* (仅限 UNIX and Linux 系统)

mqclient.ini 文件

mqclient.ini 文件可以包含以下节:

- 通道
- ClientExit 路径
- LU62, NETBIOS 和 SPX
- MessageBuffer
- SSL
- TCP

此外, 您可能需要 PreConnect 节来配置预连接出口。

用于分布式排队的配置文件节

与分布式排队相关的队列管理器配置文件 qm.ini 节的描述。

本主题显示队列管理器配置文件中与分布式排队相关的节。它适用于 Windows 和 UNIX and Linux 系统上 IBM WebSphere MQ 的队列管理器配置文件。该文件在所有平台上都称为 qm.ini。

与分布式排队相关的节为:

- 通道
- TCP
- LU62
- NETBIOS
- SPX (仅限 Windows XP 和 Windows 2003 Server)
- 出口路径

第 50 页的图 3 显示了可以使用这些节设置的值。在定义其中一个节时, 不需要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。

```

CHANNELS:
  MAXCHANNELS=n          ; Maximum number of channels allowed, the
                        ; default value is 100.
  MAXACTIVECHANNELS=n   ; Maximum number of channels allowed to be active at
                        ; any time, the default is the value of MaxChannels.
  MAXINITIATORS=n       ; Maximum number of initiators allowed, the default
                        ; and maximum value is 3.
  MQIBINDTYPE=type1     ; Whether the binding for applications is to be
                        ; "fastpath" or "standard".
                        ; The default is "standard".
  ADOPTNEWMCA=chltype   ; Stops previous process if channel fails to start.
                        ; The default is "NO".
  ADOPTNEWMCATIMEOUT=n ; Specifies the amount of time that the new
                        ; process should wait for the old process to end.
                        ; The default is 60.
  ADOPTNEWMCACHECK=    ; Specifies the type checking required.
                        ; The default is "NAME", "ADDRESS", and "QM".
                        typecheck
  TCP:                  ; TCP entries
  PORT=n               ; Port number, the default is 1414
  KEEPALIVE=Yes        ; Switch TCP/IP KeepAlive on
  LIBRARY2=DLLName2   ; Used if code is in two libraries
  EXITPATH:2          ; Location of user exits (MQSeries for AIX,
                        ; HP-UX, and Solaris only)
  EXITPATHS=          ; String of directory paths.

```

图 3: 用于分布式排队的 *qm.ini* 节

注:

1. MQIBINDTYPE 仅适用于 IBM WebSphere MQ for AIX, IBM WebSphere MQ for HP-UX 和 IBM WebSphere MQ for Solaris。
2. EXITPATH 仅适用于 IBM WebSphere MQ for AIX, IBM WebSphere MQ for HP-UX 和 IBM WebSphere MQ for Solaris。

相关信息

配置

在 Windows, UNIX 和 Linux 系统上更改配置信息

通道属性

本部分描述了在通道定义中保留的通道属性。

此信息是产品敏感的编程接口信息。

您可以选择要针对每个通道的特定环境集最佳的通道属性。但是, 当通道正在运行时, 实际值在启动协商期间可能已更改。请参阅 [准备通道](#)。

许多属性都具有缺省值, 您可以将这些值用于大多数通道。但是, 在缺省值并非最佳的情况下, 请参阅本节以获取有关选择正确值的指导信息。

注: 在 WebSphere MQ for IBM i 中, 可以将大多数属性指定为 *SYSDFTCHL, 这意味着该值取自系统中的系统缺省通道。

通道属性和通道类型

不同类型的通道支持不同的通道属性。

WebSphere MQ 通道属性的通道类型在 [第 50 页的表 16](#) 中列出。

属性字段	MQSC 命令参数	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
更改日期	ALTDATE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
变更时间	ALTTIME	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

表 16: 通道类型的通道属性 (继续)									
属性字段	MQSC 命令参数	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
批处理脉动信号间隔	BATCHHB	Yes	Yes					Yes	Yes
批处理时间间隔	BATCHINT	Yes	Yes					Yes	Yes
批处理大小	BATCHSZ	Yes	Yes	Yes	Yes			Yes	Yes
通道名称	通道	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
通道统计	STATCHL	Yes	Yes	Yes	Yes			Yes	Yes
通道类型	CHLTYPE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
客户机通道权重	CLNTWGHT					Yes			
集群	CLUSTER							Yes	Yes
集群名称列表	CLUSNL							Yes	Yes
集群工作负载优先级	CLWLPRTY							Yes	Yes
集群工作负载列组	CLWLRANK							Yes	Yes
集群工作负载权重	CLWLWGHT							Yes	Yes
连接亲缘关系	AFFINITY					Yes			
连接名称	CONNNAME	Yes	Yes		Yes	Yes		Yes	Yes
转换消息	CONVERT	Yes	Yes					Yes	Yes
数据压缩	COMPMSG	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
描述	DESCR	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
断开连接时间间隔	DISCINT	Yes	Yes					Yes	Yes
头压缩	COMPHDR	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
脉动信号间隔	HBINT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
保持活动时间间隔	KAINT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
本地地址	LOCLADDR	Yes	Yes		Yes	Yes		Yes	Yes
长重试次数	LONGRTY	Yes	Yes					Yes	Yes
长重试时间间隔	LONGTMR	Yes	Yes					Yes	Yes
LU 6.2 方式名	MODENAME	Yes	Yes		Yes	Yes		Yes	Yes
LU 6.2 事务程序名	TPNAME	Yes	Yes		Yes	Yes		Yes	Yes
最大实例数	MAXINST						Yes		
每个客户机的最大实例数	MAXINSTC						Yes		
最大消息长度	MAXMSGL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
消息通道代理程序名称	MCANAME	Yes	Yes		Yes			Yes	Yes
消息通道代理程序类型	MCTYPE	Yes	Yes		Yes			Yes	Yes
消息通道代理程序用户	MCAUSER	Yes	Yes	Yes	Yes		Yes	Yes	Yes
消息出口名	MSGEXIT	Yes	Yes	Yes	Yes			Yes	Yes

表 16: 通道类型的通道属性 (继续)									
属性字段	MQSC 命令参数	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
消息出口用户数据	MSGDATA	Yes	Yes	Yes	Yes			Yes	Yes
消息重试出口名称	MREXIT			Yes	Yes				Yes
消息重试出口用户数据	MRDATA			Yes	Yes				Yes
消息重试次数	MRRTY			Yes	Yes				Yes
消息重试时间间隔	MRTMR			Yes	Yes				Yes
Monitoring	MONCHL	Yes	Yes	Yes	Yes		Yes	Yes	Yes
网络连接优先级	NETPRTY								Yes
非持久消息速度	NPMSPEED	Yes	Yes	Yes	Yes			Yes	Yes
密码	密码	Yes	Yes		Yes	Yes		Yes	
属性控制	PROPCTL	Yes	Yes					Yes	Yes
放入权限	PUTAUT			Yes	Yes				Yes
队列管理器名称	QMNAME					Yes			
接收出口名	RCVEXIT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
接收出口用户数据	RCVDATA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
安全出口名	SCYEXIT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
安全出口用户数据	SCYDATA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
发送出口名	SENDEXIT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
发送出口用户数据	SENDDATA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
序号回绕	SEQWRAP	Yes	Yes	Yes	Yes			Yes	Yes
共享连接	SHARECNV					是	是		
短重试次数	SHORTRTY	Yes	Yes					Yes	Yes
短重试时间间隔	SHORTTMR	Yes	Yes					Yes	Yes
SSL 密码规范	SSLCIPH	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SSL 客户机认证	SSLCAUTH		Yes	Yes	Yes		Yes		Yes
SSL 同级	SSLPEER	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
传输队列的名称	XMITQ	Yes	Yes						
传输类型	TRPTYPE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
使用死信队列	USEDLQ	Yes	Yes	Yes	Yes			Yes	Yes
用户标识	USERID	Yes	Yes		Yes	Yes		Yes	

相关概念

第 53 页的『按字母顺序排列的通道属性』

本节描述通道对象的每个属性及其有效值，并在适当情况下说明其使用情况。

相关参考

[MQSC 引用](#)

按字母顺序排列的通道属性

本节描述通道对象的每个属性及其有效值，并在适当情况下说明其使用情况。

对于某些平台，WebSphere MQ 可能未实现本节中显示的所有属性。在个别属性描述中提及异常和平台差异(如果相关)。

可以在 MQSC 中指定的关键字显示在每个属性的方括号中。

这些属性按字母顺序排列。

变更日期 (ALTDATE)

此属性是上次变更定义 的日期，格式为 yyyy-mm-dd。

此属性对所有通道类型均有效。

变更时间 (ALTTIME)

此属性是上次更改定义的时间，格式为 hh:mm:ss。

此属性对所有通道类型均有效。

批处理脉动信号间隔 (BATCHHB)

此属性允许发送通道在落实一批消息之前验证接收通道是否仍处于活动状态。

因此，批处理脉动信号间隔允许回退批处理，而不是在接收通道未处于活动状态时变为不确定状态。通过回退批处理，这些消息仍可用于处理，例如，可以将它们重定向到另一个通道。

如果发送通道在批处理脉动信号间隔内具有来自接收通道的通信，那么假定接收通道仍处于活动状态，否则将“脉动信号”发送到接收通道以进行检查。

该值以毫秒为单位，必须在范围 0 到 999999 之间。值为零表示不使用批处理心脏跳动。

此属性对以下通道类型有效：

- 发送方
- 服务器
- 集群发送方
- 集群接收方

批处理时间间隔 (BATCHINT)

此属性是一个周期(以毫秒为单位)，在此期间，即使传输队列上没有消息，通道也会保持批处理打开。

您可以指定任何毫秒数，从零到 999 999 999。缺省值为零。

如果未指定批处理时间间隔，那么当 BATCHSZ 中指定的消息数已发送或传输队列变为空时，该批处理将关闭。在负载较轻的通道上，当传输队列频繁变为空时，有效批处理大小可能远小于 BATCHSZ。

您可以使用 BATCHINT 属性通过减少短批处理数来提高通道的效率。但是，请注意，您可以减慢响应时间，因为批处理持续时间更长，而消息未落实的时间更长。

如果指定 BATCHINT，那么仅当满足下列其中一个条件时，批处理才会关闭：

- 已发送 BATCHSZ 中指定的消息数。
- 传输队列上没有更多消息，并且在等待消息时已经过 BATCHINT 时间间隔(自检索到批处理的第一条消息以来)。

注：BATCHINT 指定等待消息所耗用的总时间。它不包括检索已在传输队列上可用的消息所花费的时间，也不包括传输消息所花费的时间。

此属性对以下通道类型有效：

- 发送方

- 服务器
- 集群发送方
- 集群接收方

批处理大小 (BATCSZ)

此属性是在获取同步点之前要发送的最大消息数。

批处理大小不会影响通道传输消息的方式; 消息始终单独传输, 但会作为批处理落实或回退。

为了提高性能, 您可以设置批处理大小以定义要在两个同步点之间传输的最大消息数。当通道启动时, 将协商要使用的批处理大小, 并采用两个通道定义中的较低者。在某些实现上, 将根据两个通道定义中的最低值和两个队列管理器 MAXUMSGS 值来计算批处理大小。批处理的实际大小可以较小; 例如, 当传输队列上没有剩余消息或批处理时间间隔到期时, 批处理完成。

批处理大小的较大值会增加吞吐量, 但恢复时间会增加, 因为有更多消息要回退并再次发送。缺省 BATCSZ 为 50, 建议您先尝试该值。如果通信不可靠, 那么您可以为 BATCSZ 选择较低的值, 从而更有可能需要恢复。

同步点过程需要唯一的逻辑工作单元标识, 以便在每次获取同步点时通过链接进行交换, 以协调批处理落实过程。

如果同步的批处理落实过程中断, 那么可能会出现不确定情况。当消息通道启动时, 将自动解决不确定情境。如果此解决方法不成功, 那么可能需要使用 RESOLVE 命令进行手动干预。

选择批量大小的数字时的一些注意事项:

- 如果该数字过大, 那么链路两端占用的队列空间量将过大。消息在未落实时占用队列空间, 并且在落实之前无法从队列中除去这些消息。
- 如果可能有稳定的消息流, 那么可以通过增加批处理大小来提高通道的性能, 因为传输相同数量的字节需要更少的确认流。
- 如果消息流特征指示消息间歇性到达, 那么具有相对较大的断开连接时间间隔的批处理大小 1 可能会提供更好的性能。
- 该数字可以在 1 到 9999 的范围内。但是, 出于数据完整性原因, 连接到任何当前平台的通道必须指定大于 1 的批处理大小。除了 WebSphere MQ for MVS 之外, 1 的值用于 V1 产品。
- 即使快速通道上的非持久消息不等待同步点, 它们也会影响批量大小计数。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

通道名称 (CHANNEL)

此属性指定通道定义的名称。

该名称最多可以包含 20 个字符, 但由于消息通道的两端必须具有相同的名称, 并且其他实现可能对大小有限制, 因此实际字符数可能必须较小。

在可能的情况下, 通道名称对于互连队列管理器网络中任意两个队列管理器之间的一个通道是唯一的。

该名称必须包含以下列表中的字符:

字母顺序	(A-Z, a-z; 注意大写和小写都很重要)
数字	(0-9)

时期	(.)
正斜杠	(/)
下划线	(_)
百分比符号	(%)

注:

1. 不允许嵌入空白，将忽略前导空白。
2. 在使用 EBCDIC 片假名的系统上，不能使用小写字符。

此属性对所有通道类型均有效。

通道统计信息 (STATCHL)

此属性控制通道的统计数据收集。

可能的值为:

QMGR

此通道的统计信息数据收集基于队列管理器属性 STATCHL 的设置。该值为缺省值。

关闭

已禁用此通道的统计信息数据收集。

低

此通道的统计信息数据收集以较低的数据收集比率启用。

中

此通道的统计信息数据收集已启用，数据收集比率适中。

高

此通道的统计信息数据收集是通过高比率的数据收集来启用的。

有关通道统计信息的更多信息，请参阅 [监控参考](#)。

此属性在 z/OS 上不受支持。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

通道类型 (CHLTYPE)

此属性指定要定义的通道的类型。

可能的通道类型为:

消息通道类型:

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

MQI 通道类型:

- 客户机连接 (仅适用于 Windows 系统和 UNIX 系统的 WebSphere MQ)
注: 还可以在 z/OS 上定义客户机连接通道, 以便在其他平台上使用。
- 服务器连接

通道的两端必须具有相同的名称和兼容类型:

- 具有接收方的发送方
- 具有服务器的请求者
- 具有发送方的请求者 (用于回调)
- 具有接收方的服务器 (服务器用作发送方)
- 客户机连接与服务器连接
- 具有集群接收方的集群发送方

客户机通道权重 (CLNTWGHT)

此属性指定用于影响所使用的客户机连接通道定义的权重。

使用客户机通道权重属性的目的在于当存在多个合适的客户机通道定义时, 可以根据其权重随机地选择这些定义。

当客户机发出 MQCONN 请求与队列管理器组的连接时, 通过指定以星号开头的队列管理器名称 (该名称允许在多个队列管理器之间进行客户机权重均衡), 并且在客户机通道定义表 (CCDT) 中提供了多个合适的通道定义时, 将根据权重随机选择要使用的定义, 首先按字母顺序选择任何适用的 CLNTWGHT (0) 定义。

请指定 0 至 99 范围内的值。缺省值是 0。

零值指示不执行负载均衡并且按字母顺序选择适用的定义。要启用负载均衡, 请指定一个在范围 1 至 99 之间的值, 其中 1 是最小的权重, 99 是最大的权重。两个或多个具有非零权重的通道之间的连接分布与这些权重的比率成正比。例如, 选择了三个 CLNTWGHT 值为 2, 4 和 14 的通道, 大约是时间的 10%, 20% 和 70%。不保证此分布。如果连接的 AFFINITY 属性设置为 PREFERRED, 那么第一个连接会根据客户机权重选择通道定义, 然后后续连接继续使用相同的通道定义。

此属性仅对客户机连接通道类型有效。

集群 (CLUSTER)

此属性是通道所属的集群的名称。

最大长度为 48 个字符, 符合命名 WebSphere MQ 对象的规则。

最多可以有一个 CLUSTER 或 CLUSNL 的结果值为非空白。如果其中一个值为非空白, 那么另一个值必须为空白。

此属性对以下通道类型有效:

- 集群发送方
- 集群接收方

集群名称列表 (CLUSNL)

此属性是名称列表的名称, 用于指定通道所属的集群的列表。

最多可以有一个 CLUSTER 或 CLUSNL 的结果值为非空白。如果其中一个值为非空白, 那么另一个值必须为空白。

此属性对以下通道类型有效:

- 集群发送方
- 集群接收方

集群工作负载优先级 (CLWLPRTY)

此属性指定通道的优先级。

该值必须在范围 0 到 9 中，其中 0 是最低优先级，9 是最高优先级。

此属性对以下通道类型有效：

- 集群发送方
- 集群接收方

集群工作负载列组 (CLWLRANK)

此属性指定通道的排名。

该值必须在范围 0 到 9 中，其中 0 是最低等级，9 是最高等级。

此属性对以下通道类型有效：

- 集群发送方
- 集群接收方

集群工作负载权重 (CLWLWGHT)

此属性将加权因子应用于通道，以便可以控制在该通道下发送的消息的比例。

该值必须在范围 1 到 99 之间，其中 1 是最低权重，而 99 是最高权重。

此属性对以下通道类型有效：

- 集群发送方
- 集群接收方

连接亲缘关系 (AFFINITY)

此属性指定使用同一队列管理器名称多次连接的客户机应用程序是否使用同一客户机通道。

当存在多个适用的通道定义时，请使用此属性。

可能的值为：

PREFERRED

进程中读取客户机通道定义表的第一个连接根据客户机通道权重创建一个适用的定义列表，权重为 0 的任何定义按字母顺序排在前面。进程中的每个连接尝试使用该列表中的第一个定义进行连接。如果连接不成功，那么将使用下一个定义。客户机通道权重值不为 0 的不成功定义将移至列表末尾。客户机通道权重为 0 的定义仍位于列表的前面，每次连接时将首先选择这些定义。

具有相同主机名的每个客户机进程总是创建相同的列表。

对于使用 C，C++ 或 .NET 编程框架 (包括完全受管的 .NET) 编写的客户机应用程序以及使用 IBM WebSphere MQ classes for Java 和 IBM WebSphere MQ classes for JMS 的应用程序，如果自创建此列表以来已修改 CCDT，那么将更新此列表。

该值为缺省值。

无

进程中读取 CCDT 的第一个连接创建适用的定义列表。进程中的所有连接根据客户机通道权重选择适用的定义，首先按字母顺序选择权重为 0 的任何定义。

对于使用 C，C++ 或 .NET 编程框架 (包括完全受管的 .NET) 编写的客户机应用程序以及使用 IBM WebSphere MQ classes for Java 和 IBM WebSphere MQ classes for JMS 的应用程序，如果自创建此列表以来已修改 CCDT，那么将更新此列表。

此属性仅对客户机连接通道类型有效。

连接名称 (CONNNAME)

此属性是通信连接标识。它指定此通道使用的特别通信链路。

它对于服务器通道是可选的，除非触发了服务器通道，在这种情况下，它必须指定连接名称。

指定 CONNNAME 作为有状态 TRPTYPE 的机器的名称的逗号分隔列表。通常只需要一个机器名称。您可以提供多个机器名称来配置具有相同属性的多个连接。通常会采用连接列表中指定的顺序尝试连接，直至成功建立连接。如果提供了 CLNTWGHT 属性，将为客户机修改顺序。如果没有任何连接成功，那么通道会重试连接，具体由该通道的属性确定。通过客户机通道，连接列表提供了除使用队列管理器组之外的替代方法来配置多个连接。通过消息通道，连接列表用于配置与多实例队列管理器的替代地址的连接。

最早在 IBM WebSphere MQ Version 7.0.1 中支持在列表中提供多个连接名称。它更改了 CONNNAME 参数的语法。较早的客户机和队列管理器使用列表中的第一个连接名称进行连接，并且不会读取列表中的剩余连接名称。为了使较早的客户机和队列管理器能够解析新语法，您必须在列表中的第一个连接名称上指定端口号。指定端口号可避免在从运行 IBM WebSphere MQ Version 7.0.1 之前级别的客户机或队列管理器连接到通道时出现问题。

在 AIX, HP-UX, IBM i, Linux, Solaris, and Windows 平台上，集群接收方通道的 TCP/IP 连接名称参数是可选的。如果您将连接名称留空，IBM WebSphere MQ 将为您生成一个连接名称，采用缺省端口并使用系统的当前 IP 地址。您可以覆盖缺省端口号，但仍使用系统的当前 IP 地址。对于每个连接名称，请将 IP 名称留空，并在括号内提供端口号；例如：

```
(1415)
```

生成的 CONNNAME 始终采用点分十进制 (IPv4) 或十六进制 (IPv6) 格式，而不采用字母数字 DNS 主机名格式。

对于 z/OS，名称最多为 48 个字符 (请参阅注释 1)，对于其他平台，名称最多为 264 个字符，并且：

如果传输类型为 TCP

CONNNAME 是远程机器 (或集群接收方通道的本地机器) 的主机名或网络地址。例如，(ABC.EXAMPLE.COM)，(2001:DB8:0:0:0:0:0:0) 或 (127.0.0.1)。它可以包含端口号，例如 (MACHINE(123))。它可以包含 z/OS 动态 DNS 组或 Network Dispatcher 输入端口的 IP_name。

如果在仅支持 IPV4 的网络中使用 IPV6 地址，那么不会解析连接名称。在同时使用 IPV4 和 IPV6 的网络中，连接名称与本地地址交互以确定使用的 IP 堆栈。请参阅第 62 页的『本地地址 (LOCLADDR)』，以了解更多信息。

如果传输类型为 LU 6.2

对于 WebSphere MQ for IBM i，Windows 系统和 UNIX 系统，如果指定了 TPNAME 和 MODENAME，请提供伙伴 LU 的标准名称。对于其他版本，或者如果 TPNAME 和 MODENAME 为空白，请提供特定平台的 CPI-C 端信息对象名。

在 z/OS 上，有两种格式用于指定值：

- 逻辑单元名称

队列管理器的逻辑单元信息包括逻辑单元名、TP 名和可选方式名。可以通过以下三种格式之一指定此名称：

表单	示例
luname	IGY12355
luname/TPname	IGY12345/APING
luname/TPname/modename	IGY12345/APINGD/#INTER

对于第一种格式，必须为 TPNAME 和 MODENAME 属性指定 TP 名称和方式名；否则这些属性必须为空。

注：对于客户机连接通道，只允许第一种形式。

- 符号名称

队列管理器的逻辑单元信息的符号目标名称，在辅助信息数据集中定义。TPNAME 和 MODENAME 属性必须为空。

注: 对于集群接收方通道，辅助信息在集群的其它队列管理器上。或者，在此情况下，它可以是通道自动定义出口可以解析为本地队列管理器的相应逻辑单元信息的名称。

指定或隐含的 LU 名可以是 VTAM 通用资源组的 LU 名。

如果传输协议为 NetBIOS

CONNNAME 是远程机器上定义的 NetBIOS 名称。

如果传输协议是 SPX

CONNNAME 是 SPX 样式的地址，由 4 字节网络地址，6 字节节点地址和 2 字节套接字号组成。以十六进制输入这些值，网络和节点地址之间用句点和方括号中的套接字号分隔。例如：

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

如果省略套接字号，那么将使用缺省 WebSphere MQ SPX 套接字号。缺省值为 X'5E86'。

此属性对以下通道类型有效：

- 发送方
- 服务器
- 请求者
- 客户机连接
- 集群发送方
- 集群接收方

它对于服务器通道是可选的，除非触发了服务器通道，在这种情况下，它必须指定连接名称。

注:

1. 针对 48 个字符限制的变通方法可能是下列其中一项建议：
 - 设置 DNS 服务器，例如，使用 "myserver" 而不是 "myserver.location.company.com" 的主机名，以确保可以使用短主机名。
 - 使用 IP 地址。
2. 传输协议的定义包含在 [第 75 页的『传输类型 \(TRPTYPE\)』](#) 中。

转换消息 (CONVERT)

此属性指定在传输之前必须将消息转换为接收系统所需的格式。

应用程序消息数据通常由接收应用程序转换。但是，如果远程队列管理器位于不支持数据转换的平台上，请使用此通道属性来指定必须将消息转换为接收系统 **前** 传输所需的格式。

可能的值为 yes 和 no。如果指定 yes，那么如果指定了其中一个内置格式名称，或者数据转换出口可用于用户定义的格式，那么将在发送之前转换消息中的应用程序数据 (请参阅 [编写数据转换出口](#))。如果指定 no，那么在发送之前不会转换消息中的应用程序数据。

此属性对以下通道类型有效：

- 发送方
- 服务器
- 集群发送方
- 集群接收方

数据压缩 (COMPMSG)

此属性是通道支持的消息数据压缩技术的列表。

对于发送方，服务器，集群发送方，集群接收方和客户机连接通道，指定的值按优先顺序排列。使用通道的远程端支持的第一种压缩技术。将通道的相互支持的压缩技术传递到发送通道的消息出口，在该出口中，可以根据每条消息来更改所使用的压缩技术。压缩会改变传递到发送和接收出口的数据。请参阅 [第 61 页的『头压缩 \(COMPHDR\)』](#) 以获取消息头的压缩。

可能的值为：

无

不执行消息数据压缩。该值为缺省值。

RLE

使用运行长度编码执行消息数据压缩。

ZLIBFAST

使用 zlib 压缩技术来执行消息数据压缩。推荐使用快速压缩时间。

ZLIBHIGH

使用 zlib 压缩技术来执行消息数据压缩。推荐使用高级压缩。

ANY

允许通道支持队列管理器支持的任何压缩技术。仅支持 "接收方"，"请求者" 和 "服务器连接" 通道。

此属性对所有通道类型均有效。

描述 (DESCR)

此属性描述通道定义，最多包含 64 个字节的文本。

注：如果系统使用双字节字符集(DBCS),那么最大字符数将减少。

使用由队列管理器的编码字符集标识 (CCSID) 标识的字符集中的字符，以确保在将文本发送到另一个队列管理器时正确转换文本。

此属性对所有通道类型均有效。

断开连接时间间隔 (DISCINT)

此属性是通道关闭的时间长度 (如果在该时间段内没有消息到达)。

此属性是服务器，集群发送方，发送方和集群接收方通道的超时属性 (以秒为单位指定)。时间间隔是从批处理结束点 (即达到批处理大小或批处理时间间隔到期且传输队列变为空时) 开始测量的。如果在指定的时间间隔内没有消息到达传输队列，那么通道将关闭。(时间大致。)

通道两端之间的控制数据的关闭交换包括关闭原因的指示。这将确保通道的相应端保持可用状态以再次启动。

您可以指定从零到 999 999 之间的任何秒数，其中零值表示不断开连接; 无限期等待。

对于使用 TCP 协议的服务器连接通道，时间间隔表示以秒为单位指定的客户机不活动断开连接值。如果服务器连接在此持续时间内未接收到来自其伙伴客户机的通信，那么它将终止该连接。

服务器连接不活动时间间隔在来自客户机的 WebSphere MQ API 调用之间适用。

注：具有等待调用的潜在长时间运行的 MQGET 不会归类为不活动，因此，不会因 DISCINT 到期而超时。

此属性对以下通道类型有效：

- 发送方
- 服务器
- 服务器连接
- 集群发送方
- 集群接收方

此属性不适用于使用 TCP 以外的协议的服务器连接通道。

注：性能受为断开连接时间间隔指定的值影响。

低值 (例如几秒) 可能会通过不断启动通道来损害系统性能。较大的值 (超过一个小时) 可能意味着系统资源被不必要地占用。您还可以指定脉动信号间隔, 以便当传输队列上没有消息时, 发送 MCA 向接收 MCA 发送脉动信号流, 从而使接收 MCA 有机会停顿通道, 而无需等待断开连接间隔到期。要使这两个值有效协同工作, 脉动信号间隔值必须显著低于断开连接间隔值。

缺省 DISCONT 值设置为 100 分钟。但是, 几分钟的值通常是合理的使用值, 而不会影响性能或使通道在不必要的长时间内保持运行。如果适合于您的环境, 那么可以在每个单独的通道上或通过更改缺省通道定义 (例如 SYSTEM.DEF.SENDER)。

有关更多信息, 请参阅 [停止和停顿通道](#)。

处置 (QSGDISP)

此属性指定队列共享组中通道的处置。它仅在 z/OS 上有效。

这些值是:

QMGR

在执行该命令的队列管理器的页集上定义通道。此值是缺省值。

GROUP

此通道是在共享存储库中定义的。仅当存在共享队列管理器环境时, 才允许此值。使用 QSGDISP (GROUP) 定义通道时, 将自动生成命令 DEFINE CHANNEL (name) NOREPLACE QSGDISP (COPY) 并将其发送到所有活动队列管理器, 以使它们在页集 0 上生成本地副本。对于未处于活动状态的队列管理器, 或者在以后加入队列共享组的队列管理器, 将在队列管理器启动时生成该命令。

COPY

在执行该命令的队列管理器的页集上定义通道, 从同名的 QSGDISP (GROUP) 通道复制其定义。仅当存在共享队列管理器环境时, 才允许此值。

此属性对所有通道类型均有效。

头压缩 (COMPHDR)

此属性是通道支持的头数据压缩技术的列表。

对于发送方, 服务器, 集群发送方, 集群接收方和客户机连接通道, 指定的值与正在使用的通道的远程端支持的第一种压缩技术按优先顺序排列。将通道的相互支持的压缩技术传递到发送通道的消息出口, 在该出口中, 可以根据每条消息来更改所使用的压缩技术。压缩会改变传递到发送和接收出口的数据。

可能的值为:

无

不执行头数据压缩。该值为缺省值。

SYSTEM

执行头数据压缩。

此属性对所有通道类型均有效。

脉动信号间隔 (HBINT)

此属性指定当传输队列上没有消息时, 要从发送消息通道代理程序 (MCA) 传递的脉动信号流之间的近似时间。

脉动信号流不会阻塞在接收的 MCA, 该 MCA 正在等待消息到达或等待断开连接的时间间隔期满。当取消阻塞接收 MCA 后, 它可以将通道断开连接, 而不需要等待断开连接时间间隔期满。脉动信号流还释放为大型消息分配的任何存储缓冲区, 并关闭在通道接收端保持打开状态的任何队列。

该值以秒为单位, 必须在 0-999 999 范围内。值为零表示不发送脉动信号流。缺省值为 300。为了最有用, 该值必须显著小于断开连接时间间隔值。

对于使用 IBM WebSphere MQ classes for Java, JMS 或 .NET API 的应用程序, 将通过下列其中一种方法来确定 HBINT 值:

- 通过应用程序所使用的 SVRCONN 通道上的值。
- 或者按 CLNTCONN 通道上的值 (如果应用程序已配置为使用 CCDT)。

对于服务器连接和客户端连接通道，脉动信号可以分别从服务器端和客户端流动。如果在脉动信号间隔内未通过通道传输任何数据，那么客户端连接 MQI 代理程序将发送脉动信号流，而服务器连接 MQI 代理程序将使用另一个脉动信号流对其进行响应。发生这种情况与通道的状态无关，例如，无论它是在进行 API 调用时处于不活动状态，还是在等待客户端用户输入时处于不活动状态。无论通道的状态如何，服务器连接 MQI 代理程序也能够再次向客户端启动脉动信号。为防止服务器连接和客户端连接 MQI 代理程序同时相互跳动，在脉动信号间隔加 5 秒内，在通道中未传输任何数据后，服务器脉动信号会流动。

对于在 IBM WebSphere MQ Version 7.0 之前的通道方式下工作的服务器连接和客户端连接通道，仅当服务器 MCA 正在等待指定了 WAIT 选项的 MQGET 命令 (它已代表客户端应用程序发出) 时，脉动信号才会流动。

有关使 MQI 通道在两种方式下工作的更多信息，请参阅 [Sharing Conversations \(MQLONG\)](#)。

相关参考

[DEFINE CHANNEL](#)

[ALTER CHANNEL](#)

保持活动时间间隔 (KAINT)

此属性用于指定通道的超时值。

"保持活动时间间隔" 属性是传递到通信堆栈的值，用于指定通道的保持活动计时。它允许您为每个通道指定不同的保持活动值。

您可以针对每个通道设置通道的 "保持活动时间间隔" (KAINT) 属性。在 z/OS 以外的平台上，您可以访问和修改该参数，但仅存储和转发该参数；没有该参数的功能实现。如果需要 KAINTE 参数提供的功能，请使用脉动信号间隔 (HBINT) 参数，如第 61 页的『脉动信号间隔 (HBINT)』中所述。

要使此属性具有任何效果，必须启用 TCP/IP 保持活动。在 z/OS 上，通过发出 ALTER QMGR TCPKEEP (YES) MQSC 命令来启用保持活动。在其他平台上，当在分布式排队配置文件 qm.ini 的 TCP 节中或通过 IBM WebSphere MQ Explorer 指定 KEEPALIVE=YES 参数时，会发生此情况。还必须使用 TCP 概要文件配置数据集在 TCP/IP 本身内开启保持活动。

该值指示时间 (以秒计)，并且必须在范围 0-99999 内。"保持活动时间间隔" 值 0 指示未对通道启用特定于通道的保持活动，仅使用 TCP/IP 中设置的系统范围保持活动值。您还可以将 KAINTE 设置为值 AUTO (此值是缺省值)。如果 KAINTE 设置为 AUTO，那么 Keepalive 值基于协商脉动信号间隔 (HBINT) 的值，如下所示：

协商的 HBINT	KAINTE
>0	协商的 HBINT + 60 秒
0	0

如果为 KAINTE 指定了 AUTO，并且它是服务器连接通道，那么将使用 TCP INTERVAL 值来代替 keepalive 时间间隔。

此属性对所有通道类型均有效。

对于具有除 TCP 或 SPX 以外的 TransportType (TRPTYPE) 的所有通道，将忽略该值

本地地址 (LOCLADDR)

此属性指定通道的本地通信地址。

仅当传输类型 (TRPTYPE) 为 TCP/IP 时，此属性才适用。对于所有其他传输类型，将忽略该传输类型。

如果指定了 LOCLADDR 值，那么停止然后重新启动的通道将继续使用 LOCLADDR 中指定的 TCP/IP 地址。在恢复方案中，当通道通过防火墙进行通信时，此属性可能很有用。这很有用，因为它将除去由于通道重新启动而导致的问题 (使用它所连接的 TCP/IP 堆栈的 IP 地址)。LOCLADDR 还可以强制通道在双堆栈系统上使用 IPv4 或 IPv6 堆栈，或在单堆栈系统上使用双堆栈。

此属性对以下通道类型有效：

- 发送方

- 服务器
- 请求者
- 客户机连接
- 集群发送方
- 集群接收方

当 LOCLADDR 包含网络地址时，该地址必须是属于运行通道的系统上的网络接口的网络地址。例如，使用以下 MSQC 命令在队列管理器 ALPHA 上定义发送方通道到队列管理器 BETA 时：

```
DEFINE CHANNEL(TO.BETA) CHLTYPE(SDR) CONNAME(192.0.2.0) XMITQ(BETA) LOCLADDR(192.0.2.1)
```

LOCLADDR 地址是 IPv4 地址 192.0.2.1。此发送方通道在队列管理器 ALPHA 的系统上运行，因此 IPv4 地址必须属于其系统的其中一个网络接口。

该值是可选的 IP 地址，以及用于出站 TCP/IP 通信的可选端口或端口范围。此信息的格式如下所示：

```
LOCLADDR([ip-addr][(low-port[,high-port])][,[ip-addr][(low-port[,high-port])]])
```

LOCLADDR 的最大长度（包括多个地址）为 MQ_LOCAL_ADDRESS_LENGTH。

如果省略 LOCLADDR，那么将自动分配本地地址。

请注意，您可以使用客户机通道定义表 (CCDT) 为 C 客户机设置 LOCLADDR。

所有参数都是可选的。省略地址的 ip-addr 部分有助于为 IP 防火墙启用固定端口号配置。省略端口号有助于选择特定网络适配器，而无需识别唯一的本地端口号。TCP/IP 堆栈将生成唯一的端口号。

对每个附加本地地址多次指定 `[,[ip-addr][(low-port[,high-port])]]`。如果要指定本地网络适配器的特定子集，请使用多个本地地址。您还可以使用 `[,[ip-addr][(low-port[,high-port])]]` 来表示多实例队列管理器配置中的不同服务器上的特定本地网络地址。

ip-addr

ip-addr 通过以下三种格式之一来指定：

IPv4 点分十进制

例如 192.0.2.1

IPv6 十六进制表示法

例如 2001:DB8:0:0:0:0:0:0

字母数字主机名格式

例如 WWW.EXAMPLE.COM

low-port and high-port

low-port 和 high-port 是括在括号中的端口号。

下表显示 LOCLADDR 参数的使用方式：

LOCLADDR	含义
9.20.4.98	通道在本地绑定到此地址
9.20.4.98, 9.20.4.99	通道绑定到任一 IP 地址。该地址可能是一个服务器上的两个网络适配器，或者是多实例配置中两个不同服务器上的不同网络适配器。
9.20.4.98(1000)	通道在本地绑定到此地址和端口 1000
9.20.4.98(1000,2000)	通道在本地绑定到此地址并使用 1000 - 2000 范围内的端口
(1000)	通道在本地绑定到端口 1000
(1000,2000)	通道在本地绑定到 1000 - 2000 范围内的端口

启动通道时，为连接名称 (CONNAME) 和本地地址 (LOCLADDR) 指定的值确定用于通信的 IP 堆栈。使用的 IP 堆栈确定如下：

- 如果系统仅配置了 IPv4 堆栈，那么将始终使用 IPv4 堆栈。如果将本地地址 (LOCLADDR) 或连接名称 (CONNAME) 指定为 IPv6 网络地址，那么将生成错误，并且通道无法启动。
- 如果系统仅配置了 IPv6 堆栈，那么将始终使用 IPv6 堆栈。如果将本地地址 (LOCLADDR) 指定为 IPv4 网络地址，那么将生成错误并且通道无法启动。在支持 IPv6 映射寻址的平台上，如果将连接名称 (CONNAME) 指定为 IPv4 网络地址，那么该地址将映射到 IPv6 地址。例如，xxx.xxx.xxx.xxx 映射到 ::ffff:xxx.xxx.xxx.xxx。使用映射地址可能需要协议转换程序。尽可能避免使用映射的地址。
- 如果将本地地址 (LOCLADDR) 指定为通道的 IP 地址，那么将使用该 IP 地址的堆栈。如果将本地地址 (LOCLADDR) 指定为解析为 IPv4 和 IPv6 地址的主机名，那么连接名称 (CONNAME) 将确定使用哪些堆栈。如果将本地地址 (LOCLADDR) 和连接名称 (CONNAME) 同时指定为解析为 IPv4 和 IPv6 地址的主机名，那么使用的堆栈由队列管理器属性 IPADDRV 确定。
- 如果系统配置了双 IPv4 和 IPv6 堆栈，并且没有为通道指定本地地址 (LOCLADDR)，那么为该通道指定的连接名称 (CONNAME) 将确定要使用的 IP 堆栈。如果将连接名称 (CONNAME) 指定为解析为 IPv4 和 IPv6 地址的主机名，那么使用的堆栈由队列管理器属性 IPADDRV 确定。

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。

具有集群发送方通道的本地地址

集群发送方通道始终继承目标队列管理器上定义的相应集群接收方通道的配置。即使存在同名的本地定义集群发送方通道，也是如此，在这种情况下，手动定义仅用于初始通信。

因此，不可能依赖于集群接收方通道中定义的 LOCLADDR，因为创建集群发送方的系统可能不拥有 IP 地址。因此，不应使用集群接收方上的 LOCLADDR，除非有理由仅限制所有潜在集群发送方的端口，而不限制 IP 地址，并且已知这些端口在可能创建集群发送方通道的所有系统上都可用。

如果集群必须使用 LOCLADDR 来获取出站通信通道以绑定到特定 IP 地址，请使用 [通道自动定义出口](#)，或者尽可能使用队列管理器的缺省 LOCLADDR。使用通道出口时，它会将出口中的 LOCLADDR 值强制转换为任何自动定义的 CLUSSDR 通道。

如果通过使用出口或缺省值对集群发送方通道使用非缺省 LOCLADDR，那么任何匹配的手动定义的集群发送方通道 (例如，到完整存储库队列管理器) 也必须设置 LOCLADDR 值以启用通过通道的初始通信。

注: 如果操作系统返回 LOCLADDR 中提供的端口 (或所有端口，如果提供了端口范围) 的绑定错误，那么通道不会启动; 系统会发出错误消息。

相关概念

[使用自动定义的集群发送方通道](#)

长重试计数 (LONGRTY)

此属性指定通道尝试将会话分配给其合作伙伴的最大次数。

如果初始分配尝试失败，那么 *short retry count* 数字将递减，并且通道将重试剩余次数。如果它仍然失败，那么它将重试 *long retry count* 次，每次尝试之间的时间间隔为 *long retry interval*。如果仍未成功，那么通道将关闭。然后，必须使用命令重新启动通道 (通道启动程序不会自动启动该通道)。

(如果失败原因导致重试不可能成功，那么不会尝试重试。)

如果通道启动程序 (在 z/OS 上) 或通道 (在分布式平台上) 在通道进行重试时停止，那么在通道启动程序或通道重新启动时，或者在发送方通道成功放入消息时，将重置短重试计数和长重试计数。但是，如果关闭并重新启动通道启动程序 (在 z/OS 上) 或队列管理器 (在分布式平台上)，那么不会重置短重试计数和长重试计数。通道保留在队列管理器重新启动或放入消息之前的重试计数值。

注: 对于 IBM i, UNIX 系统和 Windows 系统:

1. 当通道从 RETRY 状态进入 RUNNING 状态时，不会立即重置短重试计数和长重试计数。仅当通道进入 RUNNING 状态 (即，本地通道确认发送到另一端的消息数后) 后，第一条消息成功流经通道时，才会重置这些消息。
2. 重新启动通道时，将重置短重试计数和长重试计数。

可以将 *long retry count* 属性从 0 设置为 999 999 999。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 集群发送方
- 集群接收方

注: 对于 UNIX 系统和 Windows 系统, 为了重试, 通道启动程序必须正在运行。通道启动程序必须监视在通道正在使用的传输队列定义中指定的启动队列。

长重试时间间隔 (LONGTMR)

此属性是在长重试方式下, 通道在重试建立连接之前要等待的大致时间间隔 (以秒计)。

如果通道必须等待变为活动状态, 那么可以延长重试之间的时间间隔。

在短重试时间间隔内尝试 *short retry count* 次之后, 通道在此长时间间隔内尝试连接 *long retry count* 次。

此属性可以设置为从零到 999 999 999。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 集群发送方
- 集群接收方

LU 6.2 方式名 (MODENAME)

此属性用于 LU 6.2 连接。执行通信会话分配时, 此属性为连接的会话特征提供额外定义。

将辅助信息用于 SNA 通信时, 方式名在 CPI-C 通信辅助对象或 APPC 辅助信息中定义, 并且此属性必须留空; 否则, 必须将其设置为 SNA 方式名。

名称长度必须为 1 到 8 个字母数字字符。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 请求者
- 客户机连接
- 集群发送方
- 集群接收方

它对于接收方或服务器连接通道无效。

LU 6.2 事务程序名 (TPNAME)

此属性用于 LU 6.2 连接。它是要在链路远端运行的事务程序 (MCA) 的名称或通用名称。

将辅助信息用于 SNA 通信时, 事务程序名是在 CPI-C 通信辅助对象或 APPC 辅助信息中定义的, 此属性必须留空。否则, 发送方通道和请求者通道需要此名称。

名称长度最多为 64 个字符。

必须将该名称设置为 SNA 事务程序名, 除非 CONNAME 包含侧对象名, 在这种情况下必须将其设置为空白。实际名称取自 CPI-C 通信端对象或 APPC 端信息数据集。

此信息在不同平台上以不同方式设置; 请参阅 [使用分布式排队连接应用程序](#) , 以获取有关为平台设置通信的更多信息。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 请求者
- 客户机连接
- 集群发送方
- 集群接收方

最大实例数 (MAXINST)

此属性指定可以启动的服务器连接通道的最大并发实例数。

可以将此属性设置为从 0 到 999 999 999。 值为零时表示在该通道上不允许进行任何客户机连接。 缺省值为 999 999 999。

"客户机连接" 功能部件 (CAF) 是 WebSphere MQ for z/OS 的选项, 支持将客户机连接到 z/OS。 如果未安装 "客户机连接" 功能部件 (CAF), 那么只能在 SYSTEM.ADMIN.SVRCONN 通道。 大于 5 的值将解释为零, 而不安装 CAF。

如果该值减小到低于当前正在运行的服务器连接通道的实例数, 那么正在运行的通道不受影响。 但是, 在已有足够的实例停止运行之前, 新实例无法启动。

此属性仅对服务器连接通道有效。

每个客户机的最大实例数 (MAXINSTC)

此属性指定可以从单个客户机启动的服务器连接通道的最大并发实例数。

可以将此属性设置为从 0 到 999 999 999。 值为零时表示在该通道上不允许进行任何客户机连接。 缺省值为 999 999 999。

"客户机连接" 功能部件 (CAF) 是 WebSphere MQ for z/OS 的选项, 支持将客户机连接到 z/OS。 如果未安装 "客户机连接" 功能部件 (CAF), 那么只能在 SYSTEM.ADMIN.SVRCONN 通道。 大于 5 的值将解释为零, 而不安装 CAF。

如果该值减小到低于当前从各个客户机运行的服务器连接通道的实例数, 那么正在运行的通道不受影响。 但是, 在停止运行足够的现有实例之前, 这些客户机的新实例无法启动。

此属性仅对服务器连接通道有效。

最大消息长度 (MAXMSGL)

此属性指定可以在通道上传输的消息的最大长度。

在 WebSphere MQ for UNIX 系统和 Windows 系统上, 指定大于或等于零且小于或等于队列管理器的最大消息长度的值。 请参阅 [ALTER QMGR](#) 中的 ALTER QMGR 命令的 MAXMSGL 参数, 以获取更多信息。

由于 WebSphere MQ 系统的各种实现存在于不同的平台上, 因此可用于消息处理的大小在某些应用程序中可能受到限制。 此数字必须反映系统能够无压力地处理的大小。 启动通道时, 将采用通道每一端的两个数字中较小的一个。

通过向消息添加数字签名和密钥, [WebSphere MQ Advanced Message Security](#) 会增加消息的长度。

注:

1. 您可以使用最大消息大小 0, 这意味着大小将设置为本地队列管理器最大值。

此属性对所有通道类型均有效。

消息通道代理程序名称 (MCANAME)

此属性是保留的，如果指定了此属性，那么只能设置为空白。

其最大长度为 20 个字符。

消息通道代理程序类型 (MCATYPE)

此属性可以将消息通道代理程序指定为 进程 或 线程。

在 WebSphere MQ for z/OS 上，仅支持通道类型为集群接收方的通道。

作为进程运行的优点包括：

- 每个通道的隔离提供更大的完整性
- 特定于每个通道的作业权限
- 控制作业调度

线程的优点包括：

- 大大减少了对存储的使用
- 通过在命令行上输入更简单的配置
- 执行速度更快-启动线程比指示操作系统启动进程更快

对于发送方，服务器和请求者的通道类型，缺省值为 `process`。对于集群发送方和集群接收方的通道类型，缺省值为 `thread`。这些缺省值可以在安装期间更改。

如果在通道定义上指定 `process`，那么将启动 `RUNMQCHL` 进程。如果指定 `thread`，那么 MCA 将在 `AMQRMPPA` 进程的线程上运行，如果指定了 `MQNOREMPOOL`，那么 MCA 将在 `RUNMQCHI` 进程的线程上运行。在接收站分配的机器上，如果使用 `RUNMSLSR`，那么 MCA 将作为线程运行。如果使用 `inetd`，那么它将作为进程运行。

在 WebSphere MQ for z/OS 上，此属性仅受通道类型为集群接收方的通道支持。在其他平台上，它对以下通道类型有效：

- 发送方
- 服务器
- 请求者
- 集群发送方
- 集群接收方

消息通道代理程序用户标识 (MCAUSER)

此属性是 MCA 用于授权访问 IBM WebSphere MQ 资源的用户标识 (字符串)。

注：为要运行的通道提供用户标识的替代方法是使用通道认证记录。通过通道认证记录，不同的连接可以使用相同的通道，同时使用不同的凭证。如果设置了通道上的两个 MCAUSER，并且通道认证记录用于应用于同一通道，那么通道认证记录优先。仅当通道认证记录使用 `USERSRC (CHANNEL)` 时，才会使用通道定义上的 MCAUSER。

此授权包括 (如果 `PUT` 权限为 `DEF`) 将消息放入接收方或请求者通道的目标队列。

在 IBM WebSphere MQ for Windows 上，可以使用格式 `user@domain` 对用户标识进行域限定，其中 `domain` 必须是本地系统的 Windows 系统域或可信域。

如果此属性为空，那么 MCA 将使用其缺省用户标识。有关更多信息，请参阅 [DEFINE CHANNEL](#)。

此属性对以下通道类型有效：

- 接收方
- 请求者
- 服务器连接
- 集群接收方

相关概念

[通道认证记录](#)

消息出口名称 (MSGEXIT)

此属性指定要由通道消息出口运行的用户出口程序的名称。

此属性可以是要连续运行的程序的名称列表。如果没有通道消息出口生效，请留空。

此属性的格式和最大长度取决于平台，如 [第 71 页的『接收出口名称 \(RCVEXIT\)』](#)。

此属性对以下通道类型有效：

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

消息出口用户数据 (MSGDATA)

此属性指定传递到通道消息出口的用户数据。

您可以运行一系列消息出口。对用户数据长度的限制以及如何为多个出口指定 MSGDATA 的示例如 RCVDATA 所示。请参阅 [第 72 页的『接收出口用户数据 \(RCVDATA\)』](#)。

此属性对以下通道类型有效：

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

消息重试出口名称 (MREXIT)

此属性指定要由消息重试用户出口运行的用户出口程序的名称。

如果没有消息重试出口程序生效，请留空。

名称的格式和最大长度取决于平台，如 [第 71 页的『接收出口名称 \(RCVEXIT\)』](#)。但是，只能指定一个消息重试出口

此属性对以下通道类型有效：

- 接收方
- 请求者
- 集群接收方

消息重试出口用户数据 (MRDATA)

此属性指定在调用时传递到通道消息重试出口的数据。

此属性对以下通道类型有效：

- 接收方
- 请求者
- 集群接收方

消息重试计数 (MRRTY)

此属性指定通道尝试重新传递消息的次数。

仅当消息重试出口名称为空时，此属性才控制 MCA 的操作。如果出口名称不为空，那么会将 MRRTY 的值传递到出口，但所进行的尝试次数(如果有)由出口控制，而不是由此属性控制。

该值必须在范围 0-999 999 999 之间。值为零表示不进行其他尝试。缺省值为 10。

此属性对以下通道类型有效:

- 接收方
- 请求者
- 集群接收方

消息重试时间间隔 (MRTMR)

此属性指定通道可以重试 MQPUT 操作之前必须经过的最短时间间隔。

此时间间隔以毫秒为单位。

仅当消息重试出口名称为空时，此属性才控制 MCA 的操作。如果出口名称不为空，那么会将 MRTMR 的值传递到出口以供出口使用，但重试时间间隔由出口控制，而不是由此属性控制。

该值必须在范围 0-999 999 999 之间。值为零表示尽快执行重试(如果 MRRTY 的值大于零)。缺省值是 1000。

此属性对以下通道类型有效:

- 接收方
- 请求者
- 集群接收方

监视 (MONCHL)

此属性控制联机监视数据的收集。

可能的值为:

QMGR

联机监视数据的集合继承自队列管理器对象中 MONCHL 属性的设置。该值为缺省值。

关闭

关闭此通道的联机监视数据收集。

低

数据收集比率低，对性能的影响最小。但是，显示的监视结果可能不是最新的。

中

数据收集的适度比率，对系统性能的影响有限。

高

数据收集的高比率，可能对性能产生影响。但是，显示的监视结果是最新的。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 接收方
- 请求者
- 服务器连接
- 集群发送方
- 集群接收方

有关监视数据的更多信息，请参阅 [显示队列和通道监视数据](#)。

网络连接优先级 (NETPRTY)

此属性指定网络连接的优先级。

如果有多个可用的路径，分布式排队选择优先级最高的路径。该值必须在 0 到 9 的范围内; 0 是最低优先级。

此属性对以下通道类型有效:

- 集群接收方

非持久消息速度 (NPMSPEED)

此属性指定发送非持久消息的速度。

可能的值为:

正常

通道上的非持久消息在事务中传输。

FAST

通道上的非持久消息不会在事务中传输。

缺省值为 FAST。这样做的好处是可以更快地检索非持久消息。缺点是由于它们不是事务的一部分，如果发生传输故障或当消息正在传输时通道停止，那么可能会丢失消息。请参阅 [消息安全 \(Safety of messages\)](#)。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

密码 (Password)

此属性指定在尝试启动与远程 MCA 的安全 LU 6.2 会话时，MCA 可以使用的密码。

您可以指定最大长度为 12 个字符的密码，尽管仅使用前 10 个字符。

它对发送方，服务器，请求者或客户机连接的通道类型有效。

在 WebSphere MQ for z/OS 上，此属性仅对客户机连接通道有效。在其他平台上，它对以下通道类型有效:

- 发送方
- 服务器
- 请求者
- 客户机连接
- 集群发送方

PUT 权限 (PUTAUT)

此属性指定要由 MCA 执行的安全性处理的类型。

此属性对以下通道类型有效:

- 接收方
- 请求者
- 集群接收方

使用此属性来选择要在执行时由 MCA 执行的安全性处理的类型:

- 针对目标队列的 MQPUT 命令 (对于消息通道)，或者

- MQI 调用 (针对 MQI 通道)。

您可以选择以下选项之一：

进程安全性，也称为缺省权限 (DEF)

将使用缺省用户标识。

在所有平台上，用于检查队列上打开权限的用户标识是在消息通道接收端运行 MCA 的进程或用户的标识。

使用此用户标识和打开选项 MQOO_SET_ALL_CONTEXT 打开队列。

上下文安全性 (CTX)

与消息关联的上下文信息中的用户标识将用作备用用户标识。

消息描述符中的 *UserIdentifier* 将移至对象描述符中的 *AlternateUserId* 字段。将使用打开的选项 MQOO_SET_ALL_CONTEXT 和 MQOO_ALTERNATE_USER_AUTHORITY 打开队列。

在所有平台上，用于检查队列上 MQOO_SET_ALL_CONTEXT 和 MQOO_ALTERNATE_USER_AUTHORITY 的打开权限的用户标识是在消息通道的接收端运行 MCA 的进程或用户的标识。用于检查队列上 MQOO_OUTPUT 的打开权限的用户标识是消息描述符中的 *UserIdentifier* 。

在服务器连接通道上不支持上下文安全性 (CTX)。

可以在 [控制上下文信息](#) 中找到有关上下文字段和打开选项的更多详细信息。

有关安全性的更多信息可以在以下位置找到：

- [安全](#)
- 在 [Windows, UNIX and Linux 系统](#) 上为 WebSphere MQ UNIX 系统和 Windows 系统设置安全性，

队列管理器名称 (QMNAME)

此属性指定 WebSphere MQ MQI 客户机应用程序可请求连接的队列管理器或队列管理器组的名称。

此属性对以下通道类型有效：

- 客户机连接

接收出口名称 (RCVEXIT)

此属性指定要由通道接收用户出口运行的用户出口程序的名称。

此属性可以是要连续运行的程序的名称列表。如果没有有效的通道接收用户出口，那么将其保留为空白。

此属性的格式和最大长度取决于平台：

- 在 z/OS 上，它是最大长度为 8 个字符的装入模块名称，但最大长度为 128 个字符的客户机连接通道除外。
- 在 IBM i 上，格式如下：

```
libname/progname
```

在 CL 命令中指定时。

在 WebSphere MQ 命令 (MQSC) 中指定时，其格式如下：

```
progname libname
```

其中 *progname* 占用前 10 个字符，*libname* 占用后 10 个字符 (如果需要，请将两个字符都填充到右侧)。字符串的最大长度为 20 个字符。

- 在 Windows 上，格式如下：

```
dllname(functionname)
```

其中指定了 *dllname*，但没有后缀 *.DLL*。字符串的最大长度为 40 个字符。

- 在 UNIX 系统上，其格式如下：

```
libraryname(functionname)
```

字符串的最大长度为 40 个字符。

在 z/OS 上的集群发送方通道自动定义期间，通道出口名称将转换为 z/OS 格式。如果要控制出口名称的转换方式，那么可以编写通道自动定义出口。有关更多信息，请参阅 [通道自动定义出口程序](#)。

您可以指定接收，发送或消息出口程序名的列表。这些名称必须以逗号和/或空格分隔。例如：

```
RCVEXIT(exit1 exit2)  
MSGEXIT(exit1,exit2)  
SENDEXIT(exit1, exit2)
```

特定出口类型的出口名称字符串和用户数据字符串的总长度限制为 500 个字符。在 WebSphere MQ for IBM i 中，最多可以列出 10 个出口名称。在 WebSphere MQ for z/OS 中，最多可以列出八个出口名称。

此属性对所有通道类型均有效。

接收出口用户数据 (RCVDATA)

此属性指定传递到接收出口的用户数据。

您可以运行一系列接收出口。一系列出口的用户数据字符串必须以逗号和/或空格分隔。例如：

```
RCVDATA(exit1_data exit2_data)  
MSGDATA(exit1_data,exit2_data)  
SENDDATA(exit1_data, exit2_data)
```

在 WebSphere MQ for UNIX 系统和 Windows 系统中，出口名称字符串和用户数据字符串的长度限制为 500 个字符。在 WebSphere MQ for IBM i 中，最多可以指定 10 个出口名称，每个出口名称的用户数据长度限制为 32 个字符。在 WebSphere MQ for z/OS 中，最多可以指定 8 个用户数据字符串，每个字符串长度为 32 个字符。

此属性对所有通道类型均有效。

安全出口名称 (SCYEXIT)

此属性指定要由通道安全出口运行的出口程序的名称。

如果没有有效的通道安全出口，就将其保留为空白。

名称的格式和最大长度取决于平台，如 [第 71 页的『接收出口名称 \(RCVEXIT\)』](#)。但是，只能指定一个安全出口。

此属性对所有通道类型均有效。

安全出口用户数据 (SCYDATA)

此属性指定传递到安全出口的用户数据。

最大长度为 32 个字符。

此属性对所有通道类型均有效。

发送出口名称 (SENDEXIT)

此属性指定要由通道发送出口运行的出口程序的名称。

此属性可以是要按顺序运行的程序的名称列表。如果没有通道发送出口生效，请保留为空。

此属性的格式和最大长度取决于平台，如 [第 71 页的『接收出口名称 \(RCVEXIT\)』](#)。

此属性对所有通道类型均有效。

发送出口用户数据 (SENDDATA)

此属性指定传递到发送出口的用户数据。

您可以运行一系列发送出口。对用户数据长度的限制以及如何为多个出口指定 SENDDATA 的示例，如 RCVDATA 所示。请参阅 [第 72 页的『接收出口用户数据 \(RCVDATA\)』](#)。

此属性对所有通道类型均有效。

序号换行 (SEQWRAP)

此属性指定消息序号在 1 处重新启动之前达到的最大数目。

该数字的值必须足够高，以避免在先前消息仍在使用该数字时重新发出该数字。通道启动时，通道的两端必须具有相同的序号合并值；否则，将发生错误。

该值可以设置为 100 到 999 999 999 之间的值。

此属性对以下通道类型有效：

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

短重试计数 (SHORTRTY)

此属性指定通道尝试将会话分配给其合作伙伴的最大次数。

如果初始分配尝试失败，那么 *short retry count* 将递减，并且通道将在每次尝试之间以 *short retry interval* 属性中定义的时间间隔重试剩余次数。如果它仍然失败，那么它会重试 *long retry count* 次，每次尝试之间的时间间隔为 *long retry interval*。如果仍不成功，那么通道将终止。

(如果失败原因导致重试不可能成功，那么不会尝试重试。)

如果通道启动程序 (在 z/OS 上) 或通道 (在分布式平台上) 在通道进行重试时停止，那么在通道启动程序或通道重新启动时，或者在发送方通道成功放入消息时，将重置短重试计数和长重试计数。但是，如果关闭并重新启动通道启动程序 (在 z/OS 上) 或队列管理器 (在分布式平台上)，那么不会重置短重试计数和长重试计数。通道保留在队列管理器重新启动或放入消息之前的重试计数值。

注：对于 UNIX 系统和 Windows 系统：

1. 当通道从 RETRY 状态进入 RUNNING 状态时，不会立即重置短重试计数和长重试计数。仅当通道进入 RUNNING 状态 (即，本地通道确认发送到另一端的消息数后) 后，第一条消息成功流经通道时，才会重置这些消息。
2. 重新启动通道时，将重置短重试计数和长重试计数。

可以将此属性设置为从 0 到 999 999 999。

此属性对以下通道类型有效：

- 发送方
- 服务器

- 集群发送方
- 集群接收方

注: 在 UNIX 系统和 Windows 系统上, 为了重试, 必须运行通道启动程序。通道启动程序必须监视在通道正在使用的传输队列定义中指定的启动队列。

短重试时间间隔 (SHORTTMR)

此属性指定在短重试方式下, 通道在重试建立连接之前要等待的大致时间间隔 (以秒计)。

如果通道必须等待变为活动状态, 那么重试之间的时间间隔可能会延长。

此属性可以设置为从零到 999 999 999。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 集群发送方
- 集群接收方

SSL 密码规范 (SSLCIPH)

此属性为 TLS 或 SSL 连接指定单个 CipherSpec。

每个 WebSphere MQ 通道定义都包含 SSLCIPH 属性。该值是最大长度为 32 个字符的字符串。

请注意下列事项:

- SSLCIPH 属性可以包含空白值, 这意味着您未使用 SSL 或 TLS。如果通道的一端具有空白 SSLCIPH 属性, 那么通道的另一端也必须具有空白 SSLCIPH 属性。
- 或者, 如果 SSLCIPH 包含非空白值, 那么通道会尝试使用指定的密码来利用 SSL 或 TLS。同样, 在这种情况下, 通道的两端都必须指定相同的 SSLCIPH 值。

它仅对传输类型 (TRPTYPE) 为 TCP 的通道有效。如果 TRPTYPE 不是 TCP, 那么将忽略该数据并且不发出错误消息。

有关 SSLCIPH 的更多信息, 请参阅 [DEFINE CHANNEL](#) 和 [指定 CipherSpecs](#)。

SSL 客户机认证 (SSLCAUTH)

此属性指定通道是否需要从 SSL 客户机接收和认证 SSL 证书。

可能的值为:

可选

如果对等 SSL 客户机发送证书, 那么将正常处理该证书, 但如果未发送任何证书, 那么认证不会失败。

必需

如果 SSL 客户机未发送证书, 那么认证将失败。

缺省值为 REQUIRED。

可以在非 SSL 通道定义上为 SSLCAUTH 指定值, 在该定义上缺少 SSLCIPH 或为空。

SSLCAUTH 是可选属性。

此属性在所有可以接收通道启动流的通道类型上都有效, 但发送方通道除外。

此属性对以下通道类型有效:

- 服务器
- 接收方
- 请求者
- 服务器连接

- 集群接收方

有关 SSLCAUTH 的更多信息，请参阅 [MQSC 引用](#) 和 [安全性](#)。

SSL 对等 (SSLPEER)

此属性用于从 IBM WebSphere MQ 通道另一端的对等队列管理器或客户机检查证书的专有名称 (DN)。

注: 通过与 SSL 或 TLS 主题专有名称进行匹配来限制通道连接的替代方法是使用通道认证记录。通过通道认证记录，可以将不同的 SSL 或 TLS 主题专有名称模式应用于同一通道。如果通道上的 SSLPEER 和通道认证记录都用于应用于同一通道，那么入站证书必须与这两个模式匹配才能进行连接。

如果从同级接收的 DN 与 SSLPEER 值不匹配，那么通道不会启动。

SSLPEER 是可选属性。如果未指定值，那么启动通道时不会检查对等 DN。

在 z/OS 上，该属性的最大长度为 256 个字节。在所有其他平台上，为 1024 字节。

在 z/OS 上，不会检查所使用的属性值。如果输入不正确的值，那么通道将在启动时失败，并且会将错误消息写入通道两端的错误日志。还会在通道两端生成 "通道 SSL 错误" 事件。在支持 SSLPEER 的平台 (z/OS 除外) 上，首次输入字符串时将检查该字符串的有效性。

您可以在非 SSL 通道定义上指定 SSLPEER 的值，在该定义上缺少 SSLCIPH 或为空。您可以使用此选项来临时禁用 SSL 以进行调试，而不必清除和稍后重新输入 SSL 参数。

有关使用 SSLPEER 的更多信息，请参阅 [MQSC 引用](#) 和 [安全性](#)。

此属性对所有通道类型均有效。

相关概念

[通道认证记录](#)

传输队列名称 (XMITQ)

此属性指定从中检索消息的传输队列的名称。

此属性对于类型为发送方或服务器的通道是必需的，它对于其他通道类型无效。

提供要与此发送方或服务器通道相关联的传输队列的名称，该名称对应于该通道远侧的队列管理器。您可以为传输队列提供与远程端的队列管理器相同的名称。

此属性对以下通道类型有效:

- 发送方
- 服务器

传输类型 (TRPTYPE)

此属性指定要使用的传输类型。

可能的值为:

LU62	LU 6.2
TCP	TCP/IP
NETBIOS	NetBIOS (第 75 页的『1』)
SPX	SPX (第 75 页的『1』)

注意:

1. 在 Windows 上使用。还可以在 z/OS 上用于定义要在 Windows 上使用的客户机连接通道。

此属性对所有通道类型有效，但被响应消息通道代理程序忽略。

使用死信队列 (USEDLQ)

此属性确定当通道无法传递消息时是否使用死信队列 (或未传递的消息队列)。

可能的值为：

否

通道无法传递的消息将被视为失败。根据 NPMSPEED 的设置，通道要么废弃这些消息，要么通道结束。

是 (缺省值)

如果队列管理器 DEADQ 属性提供死信队列的名称，那么将使用该属性，否则行为与 NO 相同。

用户标识 (USERID)

此属性指定尝试启动与远程 MCA 的安全 SNA 会话时 MCA 要使用的用户标识。

您可以指定 20 个字符的任务用户标识。

它对发送方，服务器，请求者或客户机连接的通道类型有效。

此属性不适用于 WebSphere MQ for z/OS (客户机连接通道除外)。

在接收端，如果密码保持为加密格式，并且 LU 6.2 软件正在使用不同的加密方法，那么尝试启动通道将失败，但安全性详细信息无效。可以通过将接收 SNA 配置修改为：

- 关闭密码替换，或者
- 定义安全用户标识和密码。

在 WebSphere MQ for z/OS 上，此属性仅对客户机连接通道有效。在其他平台上，它对以下通道类型有效：

- 发送方
- 服务器
- 请求者
- 客户机连接
- 集群发送方

IBM WebSphere MQ 集群命令

IBM WebSphere MQ 脚本命令 **runmqsc** 命令具有适用于集群的特殊属性和参数。您可以使用其他管理界面来管理集群。

显示 MQSC 命令的方式与系统管理员在命令控制台上输入的方式相同。请记住，您不必以这种方式发出命令。根据您的平台，还有许多其他方法；例如：

- 在 IBM WebSphere MQ for IBM i 上，从 **WRKMQM** 的选项 26 以交互方式运行 MQSC 命令。您还可以使用 CL 命令，或者可以将 MQSC 命令存储在文件中并使用 **STRMQMMQSC CL** 命令。
- 在 z/OS 上，可以使用 **CSQUTIL** 实用程序的 COMMAND 功能，操作和控制面板，也可以使用 z/OS 控制台。
- 在所有其他平台上，可以将命令存储在文件中并使用 **runmqsc**。

在 MQSC 命令中，使用 CLUSTER 属性指定的集群名称最长可以为 48 个字符。

使用 CLUSNL 属性指定的集群名称列表最多可包含 256 个名称。要创建集群名称列表，请使用 DEFINE NAMELIST 命令。

IBM WebSphere MQ 浏览器

Explorer GUI 可以使用 IBM 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中指定其他存储库。对于较早版本的 WebSphere MQ for z/OS，IBM WebSphere MQ Explorer 无法使用存储库队列管理器管理集群。因此，您必须在 IBM WebSphere MQ Explorer 可以管理的系统上指定其他存储库。

在 IBM WebSphere MQ for Windows 和 WebSphere MQ for Linux 上，您还可以使用 IBM WebSphere MQ Explorer 来处理集群。您还可以使用独立 IBM WebSphere MQ Explorer 客户机。

通过使用 IBM WebSphere MQ Explorer，您可以查看集群队列并查询集群发送方通道和集群接收方通道的状态。IBM WebSphere MQ Explorer 包含两个向导，可用于指导您完成以下任务：

- 创建集群
- 将独立队列管理器连接到集群

可编程命令格式 (PCF)

runmqsc 命令	PCF 等效
DISPLAY CLUSQMGR	MQCMD_INQUIRE_CLUSTER_Q_MGR
SUSPEND QMGR	MQCMD_SUSPEND_Q_MGR_CLUSTER
RESUME QMGR	MQCMD_RESUME_Q_MGR_CLUSTER
REFRESH CLUSTER	MQCMD_REFRESH_CLUSTER
RESET CLUSTER	MQCMD_RESET_CLUSTER

相关概念

第 76 页的『IBM WebSphere MQ 集群命令』

IBM WebSphere MQ 脚本命令 **runmqsc** 命令具有适用于集群的特殊属性和参数。您可以使用其他管理界面来管理集群。

队列管理器定义命令

可以在队列管理器定义命令上指定的集群属性。

要指定队列管理器保存集群的完整存储库，请使用指定属性 `REPOS(clustername)` 的 `ALTER QMGR` 命令。要指定多个集群名称的列表，请定义集群名称列表，然后在 `ALTER QMGR` 命令上使用属性 `REPOSNL(namelist)`：

```
DEFINE NAMELIST(CLUSTERLIST)
  DESCR('List of clusters whose repositories I host')
  NAMES(CLUS1, CLUS2, CLUS3)
  ALTER QMGR REPOSNL(CLUSTERLIST)
```

您可以在 `ALTER QMGR` 命令上提供其他集群属性

CLWLEXIT(name)

指定将消息放入集群队列时要调用的用户出口的名称。

CLWLDATA(data)

指定要传递到集群工作负载用户出口的数据。

CLWLLEN(length)

指定要传递到集群工作负载用户出口的最大消息数据量

CLWLMRUC(channels)

指定出站集群通道的最大数量。

`CLWLMRUC` 是未在集群中传播的本地队列管理器属性。它可用于集群工作负载出口和选择消息目标的集群工作负载算法。

CLWLUSEQ(LOCAL | ANY)

指定当目标队列同时具有本地实例和至少一个远程集群实例时 `MQPUT` 的行为。如果放置操作起源于集群通道，那么此属性不适用。可以将 `CLWLUSEQ` 同时指定为队列属性和队列管理器属性。

如果指定 `ANY`，那么本地队列和远程队列都是 `MQPUT` 的可能目标。

如果指定 `LOCAL`，那么本地队列是 `MQPUT` 的唯一目标。

等效的 PCF 为 MQCMD_CHANGE_Q_MGR 和 MQCMD_INQUIRE_Q_MGR。

相关参考

通道定义命令

可在通道定义命令上指定的集群属性。

队列定义命令

可以在队列定义命令上指定的集群属性。

DISPLAY CLUSQMGR

使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

SUSPEND QMGR, RESUME QMGR 和集群

使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如, 在此队列管理器上执行维护之前), 然后将其恢复。

刷新集群

从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。除非在特殊情况下, 否则您不太可能需要使用此命令。

RESET CLUSTER: Forcibly removing a queue manager from a cluster

在特殊情况下, 使用 RESET CLUSTER 命令从集群中强制除去队列管理器。

通道定义命令

可在通道定义命令上指定的集群属性。

DEFINE CHANNEL, ALTER CHANNEL 和 DISPLAY CHANNEL 命令具有针对集群的两个特定 CHLTYPE 参数: CLUSRCVR 和 CLUSSDR。要定义集群接收方通道, 请使用 DEFINE CHANNEL 命令并指定 CHLTYPE (CLUSRCVR)。集群接收方通道定义上的许多属性与接收方或发送方通道定义上的属性相同。要定义集群发送方通道, 请使用 DEFINE CHANNEL 命令, 指定 CHLTYPE (CLUSSDR) 以及用于定义发送方通道的许多相同属性。

在定义集群发送方通道时, 不再需要指定完整存储库队列管理器的名称。如果您知道用于集群中通道的命名约定, 那么可以使用 +QMNAME+ 构造来生成 CLUSSDR 定义。+QMNAME+ 构造在 z/OS 上不受支持。连接后, WebSphere MQ 会更改通道名称, 并替换正确的完整存储库队列管理器名称以代替 +QMNAME+。生成的通道名称将截断为 20 个字符。

有关命名约定的更多信息, 请参阅 [集群命名约定](#)。

仅当命名通道的约定包含队列管理器的名称时, 该技术才有效。例如, 使用名为 CLUSTER1.QM1.ALPHA 的集群接收方通道在名为 CLUSTER1 的集群中定义名为 QM1 的完整存储库队列管理器。其他每个队列管理器都可以使用通道名称 CLUSTER1.+QMNAME+.ALPHA 来定义此队列管理器的集群发送方通道。

如果对所有通道使用相同的命名约定, 请注意一次只能存在一个 +QMNAME+ 定义。

DEFINE CHANNEL 和 ALTER CHANNEL 命令上的以下属性特定于集群通道:

cluster

CLUSTER 属性指定与此通道关联的集群的名称。或者, 使用 CLUSNL 属性。

CLUSNL

CLUSNL 属性指定集群名称的名称列表。

NETPRTY

仅集群接收方。

NETPRTY 属性指定通道的网络优先级。NETPRTY 可帮助工作负载管理例程。如果有多个到目标的可能路径, 那么工作负载管理例程将选择具有最高优先级的路径。

CLWLPRTY

出于工作负载管理目的, CLWLPRTY 参数将优先级因子应用于同一目标的通道。此参数指定用于集群工作负载分发的通道优先级。该值必须在范围 0 到 9 中, 其中 0 是最低优先级, 9 是最高优先级。

CLWLRANK

CLWLRANK 参数将排名因子应用于通道以进行工作负载管理。此参数指定用于集群工作负载分布的通道的列组。该值必须在范围 0 到 9 中, 其中 0 是最低等级, 9 是最高等级。

CLWLWGHT

CLWLWGHT 参数将加权因子应用于通道以进行工作负载管理。CLWLWGHT 会对通道进行加权，以便可以控制在该通道下发送的消息的比例。集群工作负载算法使用 CLWLWGHT 来偏差目标选项，以便可以通过特定通道发送更多消息。缺省情况下，所有通道权重属性都是相同的缺省值。weight 属性允许您在功能强大的 UNIX 机器上分配比小型桌面 PC 上的其他通道更大的权重。权重越大意味着集群工作负载算法比 PC 更频繁地选择 UNIX 机器作为消息的目标。

CONNAME

集群接收方通道定义上指定的 CONNAME 在整个集群中用于标识队列管理器的网络地址。请注意为在整个 WebSphere MQ 集群中解析的 CONNAME 参数选择值。请勿使用通用名称。请记住，在集群接收方通道上指定的值优先于在相应集群发送方通道中指定的任何值。

DEFINE CHANNEL 命令和 ALTER CHANNEL 命令上的这些属性也适用于 DISPLAY CHANNEL 命令。

注: 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，以确保它们与相应集群接收方定义上的属性匹配。例如，请注意，您可以在不指定 CONNAME 参数中的端口号的情况下定义 CLUSRCVR，同时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，并且通道将失败。

注: DISPLAY CHANNEL 命令不会显示自动定义的通道。但是，您可以使用 DISPLAY CLUSQMGR 命令来检查自动定义的集群发送方通道的属性。

使用 DISPLAY CHSTATUS 命令可显示集群发送方或集群接收方通道的状态。此命令提供手动定义的通道和自动定义的通道的状态。

等效的 PCF 为 MQCMD_CHANGE_CHANNEL, MQCMD_COPY_CHANNEL, MQCMD_CREATE_CHANNEL 和 MQCMD_INQUIRE_CHANNEL。

在 CLUSRCVR 定义上省略 CONNAME 值

在某些情况下，可以省略 CLUSRCVR 定义上的 CONNAME 值。不得在 z/OS 上省略 CONNAME 值。

在 AIX, HP-UX, IBM i, Linux, Solaris, and Windows 平台上，集群接收方通道的 TCP/IP 连接名称参数是可选的。如果您将连接名称留空，IBM WebSphere MQ 将为您生成一个连接名称，采用缺省端口并使用系统的当前 IP 地址。您可以覆盖缺省端口号，但仍使用系统的当前 IP 地址。对于每个连接名称，请将 IP 名称留空，并在括号内提供端口号；例如：

(1415)

生成的 CONNAME 始终采用点分十进制 (IPv4) 或十六进制 (IPv6) 格式，而不采用字母数字 DNS 主机名格式。

当您有使用动态主机配置协议 (DHCP) 的机器时，此工具很有用。如果在 CLUSRCVR 通道上未提供 CONNAME 的值，那么不需要更改 CLUSRCVR 定义。DHCP 为您分配新的 IP 地址。

如果对 CLUSRCVR 定义上的 CONNAME 指定空白，那么 WebSphere MQ 将根据系统的 IP 地址生成 CONNAME。仅生成的 CONNAME 存储在存储库中。集群中的其他队列管理器不知道 CONNAME 最初为空。

如果发出 DISPLAY CLUSQMGR 命令，那么您将看到生成的 CONNAME。但是，如果从本地队列管理器发出 DISPLAY CHANNEL 命令，那么会看到 CONNAME 为空。

如果队列管理器由于 DHCP 而停止并使用其他 IP 地址重新启动，那么 WebSphere MQ 将重新生成 CONNAME 并相应地更新存储库。

相关参考

[队列管理器定义命令](#)

可以在队列管理器定义命令上指定的集群属性。

[队列定义命令](#)

可以在队列定义命令上指定的集群属性。

[DISPLAY CLUSQMGR](#)

使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

[SUSPEND QMGR](#), [RESUME QMGR](#) 和 [集群](#)

使用 `SUSPEND QMGR` 和 `RESUME QMGR` 命令来临时减少此队列管理器的入站集群活动 (例如, 在此队列管理器上执行维护之前), 然后将其恢复。

刷新集群

从队列管理器发出 `REFRESH CLUSTER` 命令以废弃有关集群的所有本地保存的信息。除非在特殊情况下, 否则您不太可能需要使用此命令。

RESET CLUSTER: Forcibly removing a queue manager from a cluster

在特殊情况下, 使用 `RESET CLUSTER` 命令从集群中强制除去队列管理器。

队列定义命令

可以在队列定义命令上指定的集群属性。

`DEFINE QLOCAL`, `DEFINE QREMOTE` 和 `DEFINE QALIAS` 命令以及三个等效 `ALTER` 命令上的集群属性为:

cluster

指定队列所属的集群的名称。

CLUSNL

指定集群名称的名称列表。

DefBind

指定当应用程序在 `MQOPEN` 调用上指定 `MQOO_BIND_AS_Q_DEF` 时要使用的绑定。此属性的选项为:

- 指定 `DEFBIND(OPEN)` 以在打开队列时将队列句柄绑定到集群队列的特定实例。 `DEFBIND(OPEN)` 是此属性的缺省值。
- 指定 `DEFBIND(NOTFIXED)`, 以便队列句柄不与集群队列的任何实例绑定。
- 指定 `DEFBIND(GROUP)` 以允许应用程序请求将一组消息全部分配给同一目标实例。

在队列管理器集群中公布多个同名队列时, 应用程序可以选择是否将此应用程序中的所有消息发送到单个实例 (`MQOO_BIND_ON_OPEN`), 以允许工作负载管理算法根据每条消息选择最合适的目标 (`MQOO_BIND_NOT_FIXED`), 或者允许应用程序请求将消息的 "组" 全部分配给同一目标实例 (`MQOO_BIND_ON_GROUP`)。在消息组之间重新驱动工作负载均衡 (无需队列的 `MQCLOSE` 和 `MQOPEN`)。

在队列定义上指定 `DEFBIND` 时, 将使用属性 `MQBND_BIND_ON_OPEN`, `MQBND_BIND_NOT_FIXED` 或 `MQBND_BIND_ON_GROUP` 之一来定义队列。在将组与集群配合使用时, 必须指定 `MQBND_BIND_ON_OPEN` 或 `MQBND_BIND_ON_GROUP`。

建议您在同一集群队列的所有实例上将 `DEFBIND` 属性设置为相同的值。由于 `MQOO_BIND_ON_GROUP` 是 IBM WebSphere MQ Version 7.1 中的新增内容, 因此如果打开此队列的任何应用程序正在连接到 IBM WebSphere MQ Version 7.0.1 或更低版本的队列管理器, 那么不得使用此 `MQOO_BIND_ON_GROUP`。

CLWLRANK

将排名因子应用于队列以进行工作负载管理。 `CLWLRANK` 参数在模型队列上不受支持。集群工作负载算法选择具有最高列组的目标队列。缺省情况下, 所有队列的 `CLWLRANK` 设置为零。

如果最终目标是另一个集群上的队列管理器, 那么可以在相邻集群的交集处设置任何中间网关队列管理器的列组。通过对中间队列管理器进行排名, 集群工作负载算法可以正确选择距离最终目标更近的目标队列管理器。

相同的逻辑适用于别名队列。在检查通道状态之前进行列组选择, 因此即使不可访问的队列管理器也可供选择。这具有允许通过网络路由消息的效果, 而不是让它在两个可能的目标之间进行选择 (就像优先级一样)。因此, 如果通道未启动到列组所指示的位置, 那么消息不会路由到下一个最高列组, 而是等到通道可用于该目标 (消息保留在传输队列上)。

CLWLPRTY

将优先级因子应用于队列以进行工作负载管理。集群工作负载算法选择具有最高优先级的目标队列。缺省情况下, 所有队列的优先级都设置为零。

如果有两个可能的目标队列, 那么可以使用此属性将一个目标故障转移到另一个目标。在检查通道状态后进行优先级选择。将所有消息发送到具有最高优先级的队列, 除非到该目标的通道的状态不如到其他

目标的通道的状态有利。这意味着只有最容易访问的目标可供选择。这将影响所有可用的多个目标之间的优先级划分。

CLWLUSEQ

指定队列的 MQPUT 操作的行为。此参数指定当目标队列具有本地实例和至少一个远程集群实例 (MQPUT 源自集群通道的情况除外) 时 MQPUT 操作的行为。此参数仅对本地队列有效。

可能的值包括: QMGR (行为由队列管理器定义的 CLWLUSEQ 参数指定), ANY (队列管理器将本地队列视为集群队列的另一个实例, 用于工作负载分布), LOCAL (本地队列是 MQPUT 操作的唯一目标, 前提是已启用本地队列)。MQPUT 行为取决于 [集群工作负载管理算法](#)。

DEFINE QLOCAL, DEFINE QREMOTE 和 DEFINE QALIAS 命令上的属性也适用于 DISPLAY QUEUE 命令。

要显示有关集群队列的信息, 请在 DISPLAY QUEUE 命令上指定队列类型 QCLUSTER 或关键字 CLUSINFO, 或者使用命令 DISPLAY QCLUSTER。

DISPLAY QUEUE 或 DISPLAY QCLUSTER 命令返回主管队列的队列管理器的名称 (如果有多个队列实例, 那么返回所有队列管理器的名称)。它还返回主管队列的每个队列管理器的系统名称, 表示的队列类型以及定义可供本地队列管理器使用的日期和时间。使用 CLUSQMGR, QMID, CLUSQT, CLUSDATE 和 CLUSTIME 属性返回此信息。

队列管理器 (QMID) 的系统名称是系统为队列管理器生成的唯一名称。

您可以定义同样是共享队列的集群队列。例如, 在 z/OS 上, 您可以定义:

```
DEFINE QLOCAL(MYQUEUE) CLUSTER(MYCLUSTER) QSGDISP(SHARED) CFSTRUCT(STRUCTURE)
```

等效的 PCF 为 MQCMD_CHANGE_Q, MQCMD_COPY_Q, MQCMD_CREATE_Q 和 MQCMD_INQUIRE_Q。

相关参考

[队列管理器定义命令](#)

可以在队列管理器定义命令上指定的集群属性。

[通道定义命令](#)

可在通道定义命令上指定的集群属性。

DISPLAY CLUSQMGR

使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

[SUSPEND QMGR](#), [RESUME QMGR](#) 和 [集群](#)

使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的进站集群活动 (例如, 在此队列管理器上执行维护之前), 然后将其恢复。

[刷新集群](#)

从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。除非在特殊情况下, 否则您不太可能需要使用此命令。

[RESET CLUSTER: Forcibly removing a queue manager from a cluster](#)

在特殊情况下, 使用 RESET CLUSTER 命令从集群中强制除去队列管理器。

DISPLAY CLUSQMGR

使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

如果从具有完整存储库的队列管理器发出此命令, 那么返回的信息适用于集群中的每个队列管理器。否则, 返回的信息仅适用于其关注的队列管理器。即, 其尝试将消息发送至的每个队列管理器, 以及持有完整存储库的每个队列管理器。

此信息包含适用于集群发送方和集群接收方通道的大多数通道属性。此外, 还可以显示以下属性:

DEFTYPE

队列管理器的定义方式。DEFTYPE 可以是下列其中一个值:

CLUSSDR

已在本地队列管理器上以管理方式定义集群发送方通道, 但目标队列管理器尚未识别该集群发送方通道。要处于此状态, 本地队列管理器已定义手动集群发送方通道, 但接收队列管理器未接受集群

信息。这可能是由于由于可用性或集群发送方配置中的错误 (例如, 发送方与接收方定义之间的 CLUSTER 属性不匹配) 导致从未建立通道。这是暂时性的情况或错误状态, 应该进行调查。

CLUSSDRA

此值表示自动发现的集群队列管理器, 未在本地定义集群发送方通道。这是集群队列管理器的 DEFTYPE, 对于这些集群队列管理器, 本地队列管理器没有本地配置, 但已通知这些集群队列管理器。例如

- 如果本地队列管理器是完整存储库队列管理器, 那么应该是集群中所有部分存储库队列管理器的 DEFTYPE 值。
- 如果本地队列管理器是部分存储库, 那么这可能是正在从此本地队列管理器使用的集群队列的主机, 也可能是已通知此队列管理器使用的第二个完整存储库队列管理器的主机。

如果 DEFTYPE 值为 CLUSSDRA, 并且本地和远程队列管理器都是指定集群的完整存储库, 那么配置不正确, 因为必须定义本地定义的集群发送方通道才能将其转换为 CLUSDRB 的 DEFTYPE。

CLUSDRB

集群发送方通道已在本地队列管理器上以管理方式定义, 并且已被目标队列管理器接受为有效集群通道。这是部分存储库队列管理器的手动配置的完整存储库队列管理器的预期 DEFTYPE。它还应该是集群中从一个完整存储库到另一个完整存储库的任何 CLUSQMGR 的 DEFTYPE。不应将手动集群发送方通道配置为部分存储库或从部分存储库队列管理器配置为多个完整存储库。如果在上述任一情况下看到 CLUSSDRB 的 DEFTYPE, 那么应该对其进行调查和更正。

CLUSRCVR

以管理方式定义为本地队列管理器上的集群接收方通道。这表示集群中的本地队列管理器。

注: 要确定哪些 CLUSQMGRs 是集群的完整存储库队列管理器, 请参阅 QMTYPE 属性。

有关定义集群通道的更多信息, 请参阅 集群通道。

QMTYPE

是保存完整存储库还是仅保存部分存储库。

CLUSDATE

定义可供本地队列管理器使用的日期。

CLUSTIME

定义可供本地队列管理器使用的时间。

状态

此队列管理器的集群发送方通道的状态。

SUSPEND

是否暂挂队列管理器。

cluster

队列管理器的集群。

CHANNEL

队列管理器的集群接收方通道名称。

XMITQ

队列管理器所使用的集群传输队列。此属性仅在 z/OS 以外的平台上可用。

相关参考

队列管理器定义命令

可以在队列管理器定义命令上指定的集群属性。

通道定义命令

可在通道定义命令上指定的集群属性。

队列定义命令

可以在队列定义命令上指定的集群属性。

SUSPEND QMGR, RESUME QMGR 和集群

使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如, 在此队列管理器上执行维护之前), 然后将其恢复。

刷新集群

从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。除非在特殊情况下，否则您不太可能需要使用此命令。

RESET CLUSTER: Forcibly removing a queue manager from a cluster
在特殊情况下，使用 RESET CLUSTER 命令从集群中强制除去队列管理器。

SUSPEND QMGR, RESUME QMGR 和集群

使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如，在此队列管理器上执行维护之前)，然后将其恢复。

当队列管理器从集群中暂挂时，如果集群中的备用队列管理器上存在同名的可用队列，那么它不会在其托管的集群队列上接收消息。但是，明确以此队列管理器为目标的消息，或者目标队列仅在此队列管理器上可用的消息，仍将定向到此队列管理器。

通过停止此集群的集群接收方通道，可以防止在队列管理器暂挂时接收更多入站消息。要停止集群的集群接收方通道，请使用 **SUSPEND QMGR** 命令的 FORCE 方式。

相关任务

[维护队列管理器](#)

相关参考

[队列管理器定义命令](#)

可以在队列管理器定义命令上指定的集群属性。

[通道定义命令](#)

可在通道定义命令上指定的集群属性。

[队列定义命令](#)

可以在队列定义命令上指定的集群属性。

DISPLAY CLUSQMGR

使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

刷新集群

从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。除非在特殊情况下，否则您不太可能需要使用此命令。

RESET CLUSTER: Forcibly removing a queue manager from a cluster
在特殊情况下，使用 RESET CLUSTER 命令从集群中强制除去队列管理器。

[已暂挂的队列管理器](#)

[恢复队列管理器](#)

刷新集群

从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。除非在特殊情况下，否则您不太可能需要使用此命令。

此命令有三种形式:

REFRESH CLUSTER(clustername) REPOS(NO)

缺省值。队列管理器将保留所有本地定义的集群队列管理器和集群队列以及作为完整存储库的所有集群队列管理器的知识。此外，如果队列管理器是集群的完整存储库，那么它还会保留对集群中其他集群队列管理器的了解。其他所有内容都将从存储库的本地副本中除去，并从集群中的其他完整存储库中重新构建。如果使用 REPOS(NO)，那么不会停止集群通道。完整存储库使用其 CLUSSDR 通道来通知集群的其余部分已完成刷新。

REFRESH CLUSTER(clustername) REPOS(YES)

除了缺省行为外，还会刷新表示完整存储库集群队列管理器的对象。如果队列管理器是完整存储库，那么使用此选项无效，如果使用此命令，那么将失败并记录错误 AMQ9406/CSQX406E。如果它是完整存储库，那么必须首先对其进行变更，以使其不是有关集群的完整存储库。将从手动定义的 CLUSSDR 定义中恢复完整存储库位置。发出使用 REPOS(YES) 刷新后，可以更改队列管理器，使其再次成为完整存储库 (如果需要)。

REFRESH CLUSTER(*)

刷新其所属的所有集群中的队列管理器。 如果与 REPOS(YES) REFRESH CLUSTER(*) 配合使用, 那么会产生强制队列管理器从本地 CLUSSDR 定义中的信息重新启动其搜索完整存储库的额外效果。 即使 CLUSSDR 通道将队列管理器连接到多个集群, 也会执行搜索。

注: 使用 **REFRESH CLUSTER** 命令可能会在集群正在进行时对其造成干扰, 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。 出于这种原因, 最好尽可能避免在日常工作中使用该命令, 并使用替代方法来纠正特定的不一致情况。

相关概念

集群 :[REFRESH CLUSTER 和历史记录队列](#)

相关参考

[队列管理器定义命令](#)

可以在队列管理器定义命令上指定的集群属性。

[通道定义命令](#)

可在通道定义命令上指定的集群属性。

[队列定义命令](#)

可以在队列定义命令上指定的集群属性。

DISPLAY CLUSQMGR

使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

SUSPEND QMGR, RESUME QMGR 和集群

使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如, 在此队列管理器上执行维护之前), 然后将其恢复。

RESET CLUSTER: Forcibly removing a queue manager from a cluster

在特殊情况下, 使用 RESET CLUSTER 命令从集群中强制除去队列管理器。

RESET CLUSTER: Forcibly removing a queue manager from a cluster

在特殊情况下, 使用 RESET CLUSTER 命令从集群中强制除去队列管理器。

除非在特殊情况下, 否则您不太可能需要使用此命令。

只能从完整存储库队列管理器发出 RESET CLUSTER 命令。 此命令采用两种格式, 具体取决于您是按名称还是按标识引用队列管理器。

1. `RESET CLUSTER(clustername
) QMNAME(qmname) ACTION(FORCEREMOVE) QUEUES(NO)`
2. `RESET CLUSTER(clustername
) QMID(qmid) ACTION(FORCEREMOVE) QUEUES(NO)`

不能同时指定 QMNAME 和 QMID。 如果使用 QMNAME, 并且集群中有多个具有该名称的队列管理器, 那么不会运行该命令。 使用 QMID 而不是 QMNAME 来确保运行 RESET CLUSTER 命令。

在 RESET CLUSTER 命令上指定 QUEUES(NO) 是缺省值。 指定 QUEUES(YES) 将从集群中除去对队列管理器拥有的集群队列的引用。 除了从集群本身中除去队列管理器外, 还会除去这些引用。

即使集群队列管理器在集群中不可见, 也会除去这些引用; 原因可能是先前在没有 QUEUE 选项的情况下强制除去了这些引用。

例如, 如果已删除队列管理器, 但仍对集群定义了集群接收方通道, 那么可以使用 RESET CLUSTER 命令。 您可以发出 RESET CLUSTER 命令以更快进行整理, 而不是等待 WebSphere MQ 移除这些定义 (它会自动执行此操作)。 然后, 将通知集群中的所有其他队列管理器该队列管理器不再可用。

如果队列管理器暂时损坏, 那么您可能希望在集群中的其他队列管理器尝试向其发送消息之前告知这些队列管理器。 **RESET CLUSTER** 将除去损坏的队列管理器。 稍后, 当损坏的队列管理器再次工作时, 请使用 **REFRESH CLUSTER** 命令来反转 **RESET CLUSTER** 的效果, 并将队列管理器返回到集群。 如果队列管理器位于发布/预订集群中, 那么您需要发出 **REFRESH QMGR TYPE (PROXYSUB)** 命令以恢复任何必需的代理预订。 请参阅针对发布/预订集群的 **REFRESH CLUSTER 注意事项**。

使用 `RESET CLUSTER` 命令是删除自动定义的集群发送方通道的唯一方法。在正常情况下，您不太可能需要此命令。IBM 支持中心可能会建议您发出命令，以整理集群队列管理器所保存的集群信息。请勿将此命令用作从集群中除去队列管理器的简短命令。[从集群中除去队列管理器](#) 中描述了从集群中除去队列管理器的正确方法。

由于存储库仅保留信息 90 天，因此在该时间之后，被强制除去的队列管理器可以重新连接到集群。它会自动重新连接，除非已将其删除。如果要阻止队列管理器重新加入集群，那么需要采取相应的安全措施。

所有集群命令 (`DISPLAY CLUSQMGR` 除外) 都以异步方式工作。用于更改涉及集群的对象属性的命令将更新对象并向存储库处理器发送请求。检查用于处理集群的命令的语法，并将请求发送到存储库处理器。

将异步处理发送到存储库处理器的请求以及从集群的其他成员接收的集群请求。如果必须将它们传播到整个集群以确定它们是否成功，那么处理可能需要相当长的时间。

相关参考

[队列管理器定义命令](#)

可以在队列管理器定义命令上指定的集群属性。

[通道定义命令](#)

可在通道定义命令上指定的集群属性。

[队列定义命令](#)

可以在队列定义命令上指定的集群属性。

[DISPLAY CLUSQMGR](#)

使用 `DISPLAY CLUSQMGR` 命令可显示有关集群中队列管理器的集群信息。

[SUSPEND QMGR](#)、[RESUME QMGR](#) 和 [集群](#)

使用 `SUSPEND QMGR` 和 `RESUME QMGR` 命令来临时减少此队列管理器的进站集群活动 (例如，在此队列管理器上执行维护之前)，然后将其恢复。

[刷新集群](#)

从队列管理器发出 `REFRESH CLUSTER` 命令以废弃有关集群的所有本地保存的信息。除非在特殊情况下，否则您不太可能需要使用此命令。

[RESET CLUSTER \(重置集群\)](#)

工作负载均衡

如果集群包含同一队列的多个实例，那么 WebSphere MQ 会选择要将消息路由到的队列管理器。它使用集群工作负载管理算法来确定要使用的最佳队列管理器。您可以提供工作负载均衡算法，以通过编写集群工作负载出口程序来选择队列管理器。

将根据队列管理器和队列的可用性以及与队列管理器，队列和通道相关联的许多特定于集群工作负载的属性来选择合适的目标。

如果工作负载均衡算法的结果不满足您的需求，那么您可以编写集群工作负载用户出口程序。使用出口将消息路由到集群中您选择的队列。

相关概念

[集群工作负载出口调用和数据结构](#)

本部分提供了集群工作负载出口和数据结构的参考信息。这是通用编程接口信息。

集群工作负载管理算法

工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

本部分列出了在确定要放入集群队列的消息的最终目标时使用的工作负载管理算法。这些规则受应用于队列，队列管理器和通道的以下属性的设置影响：

表 20: 集群工作负载管理的属性

队列	队列管理器	通道
<ul style="list-style-type: none"> • CLWLPRTY • CLWLRANK • CLWLUSEQ 	<ul style="list-style-type: none"> • CLWLUSEQ • CLWLMRUC 	<ul style="list-style-type: none"> • CLWLPRTY • CLWLRANK • CLWLWGHT • NETPRTY

最初，队列管理器从两个过程构建可能的目标列表：

- 将目标 `ObjectName` 和 `ObjectQmgrName` 与队列管理器所在集群中共享的队列管理器别名定义进行匹配。
- 查找到队列管理器的唯一路由（换言之，通道），该队列管理器托管名为 `ObjectName` 的队列，并且位于队列管理器所属的某个集群中。

以下算法的步骤将从可能的目标列表中除去目标。

1. 如果指定了队列名称：
 - a. 未启用放入的队列将作为可能的目标消除。
 - b. 将消除未与本地队列管理器共享集群的队列的远程实例。
 - c. 将消除与队列不在同一集群中的远程 `CLUSRCVR` 通道。
2. 如果指定了队列管理器名称，
 - a. 将消除未启用的队列管理器别名。
 - b. 将消除与本地队列管理器不在同一集群中的远程 `CLUSRCVR` 通道。
3. 如果生成的队列集包含队列的本地实例，那么通常使用队列的本地实例。如果满足以下三个条件之一，那么将使用队列的本地实例：
 - 队列 `CLWLUSEQ` 的 `use-queue` 属性设置为 `LOCAL`。
 - 以下两种情况均成立：
 - a. 队列 `CLWLUSEQ` 的 `use-queue` 属性设置为 `QMGR`。
 - b. 队列管理器 `CLWLUSEQ` 的 `use-queue` 属性设置为 `LOCAL`。
 - 通过集群通道接收消息，而不是由本地应用程序放入消息。

注：如果未设置 `MQWXP_PUT_BY_CLUSTER_CH` 和 `MQQF_CLWL_USEQ_ANY` 标志，那么可以在用户出口中检测来自集群通道的消息：

 - `MQWXP`。标志 `MQWXP_PUT_BY_CLUSTER_CH`。
 - `MQWQR`。标志 `MQQF_CLWL_USEQ_ANY`。
4. 如果消息是集群 `PCF` 消息，那么将消除已向其发送发布或预订的任何队列管理器。
5. 将消除 `CLWLRANK` 小于所有剩余通道或队列管理器别名的最大排名的队列管理器或队列管理器别名的所有通道。
6. 将消除 `CLWLRANK` 小于所有剩余队列的最大列组的所有队列（非队列管理器别名）。
7. 如果仅保留队列的远程实例，那么将优先选择恢复的队列管理器，而不是暂挂的队列管理器。
8. 如果一个队列的多个远程实例仍然存在，那么将包括所有处于不活动状态或正在运行的通道。列出了状态常量：
 - `MQCHS_INACTIVE`
 - `MQCHS_RUNNING`
9. 如果没有保留队列的远程实例，那么将包括处于绑定，初始化，启动或停止状态的所有通道。列出了状态常量：
 - `MQCHS_BINDING`

- MQCHS_INITIALIZING
 - MQCHS_STARTING
 - MQCHS_STOPPING
10. 如果没有保留队列的远程实例，那么将包括正在重试的所有通道 MQCHS_RETRYING。
 11. 如果没有保留队列的远程实例，那么将包括处于 "正在请求"，"已暂停" 或 "已停止" 状态的所有通道。列出了状态常量：
 - MQCHS_REQUESTING
 - MQCHS_PAUSED
 - MQCHS_STOPPED
 12. 如果保留了队列的多个远程实例，并且该消息是集群 PCF 消息，那么将选择本地定义的 CLUSSDR 通道。
 13. 如果保留到任何队列管理器的队列的多个远程实例，那么将为每个队列管理器选择具有最高 NETPRTY 值的通道。
 14. 如果正在选择队列管理器：
 - 将消除除具有最高优先级 CLWLPRTY 的通道和别名以外的所有剩余通道和队列管理器别名。如果保留任何队列管理器别名，那么将保留到队列管理器的通道。
 15. 如果正在选择队列：
 - 将消除除具有最高优先级的队列 CLWLPRTY 以外的所有队列，并保留通道。
 16. 将消除所有通道，但 MQWDR 中具有最高值的多个通道除外。DestSeqNumber。当剩余通道数不大于允许的最近使用的最大通道数 CLWLMRUC 时，将停止消除。
 17. 如果一个队列的多个远程实例仍然存在，那么将选择最近最少使用的通道。最近最少使用的通道具有最低值 MQWDR。DestSeqFactor。
 - 如果存在多个具有最低值的通道，那么将选择 MQWDR 中具有最低值的通道之一。DestSeqNumber。
 - 此选项的目标序列因子由队列管理器增加大约 1000/CLWLWGHT。

注：

- a. 如果更改了可用 CLUSRCVR 通道的集群工作负载属性，那么会将所有目标的目标序列因子重置为零。如果新的 CLUSRCVR 通道变为可用，那么序列因子为零。
- b. 对手动定义的 CLUSSDR 通道的工作负载属性的修改不会重置目标序列因子。

用户消息的分发并不总是准确，因为集群的管理和维护会导致消息在通道之间流动。结果是用户消息分布不均，可能需要一些时间才能稳定下来。由于管理消息和用户消息混合在一起，因此在工作负载均衡期间不依赖消息的精确分布。

相关参考

CLWLPRTY 队列属性

CLWLPRTY 队列属性指定集群工作负载分布的本地，远程或别名队列的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 队列属性

CLWLRANK 队列属性指定用于集群工作负载分发的本地队列，远程队列或别名队列的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLUSEQ 队列属性

CLWLUSEQ 队列属性指定是否将队列的本地实例作为目标优先于集群中的其他实例。

CLWLUSEQ 队列管理器属性

CLWLUSEQ 队列管理器属性指定是否将队列的本地实例作为目标优先于集群中队列的其他实例。如果 CLWLUSEQ 队列属性设置为 QMGR，那么此属性适用。

CLWLMRUC 队列管理器属性

CLWLMRUC 队列管理器属性设置最近选择的通道数。集群工作负载管理算法使用 CLWLMRUC 来限制活动出站集群通道数。该值必须在范围 1-999 999 999 之间。

CLWLPRTY 通道属性

CLWLPRTY 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 通道属性

CLWLRANK 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLWGHT 通道属性

CLWLWGHT 通道属性指定应用于集群工作负载分布的 CLUSSDR 和 CLUSRCVR 通道的权重。该值必须在范围 1-99 内，其中 1 是最低权重，而 99 是最高权重。

NETPRTY 通道属性

NETPRTY 通道属性指定 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLPRTY 队列属性

CLWLPRTY 队列属性指定集群工作负载分布的本地，远程或别名队列的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

使用 CLWLPRTY 队列属性来设置目标队列的首选项。WebSphere MQ 先选择具有最高优先级的目标，然后再选择具有最低集群目标优先级的目标。如果存在多个具有相同优先级的目标，那么将选择最近最少使用的目标。

如果有两个可能的目标，那么可以使用此属性来允许故障转移。优先级最高的队列管理器接收请求，优先级较低的队列管理器充当保留。如果最高优先级队列管理器失败，那么可用的下一个最高优先级队列管理器将接管。

WebSphere MQ 在检查通道状态后获取队列管理器的优先级。只有可用的队列管理器才是可供选择的候选者。

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。

相关概念

集群工作负载管理算法

工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

相关参考

CLWLRANK 队列属性

CLWLRANK 队列属性指定用于集群工作负载分发的本地队列，远程队列或别名队列的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLUSEQ 队列属性

CLWLUSEQ 队列属性指定是否将队列的本地实例作为目标优先于集群中的其他实例。

CLWLUSEQ 队列管理器属性

CLWLUSEQ 队列管理器属性指定是否将队列的本地实例作为目标优先于集群中队列的其他实例。如果 CLWLUSEQ 队列属性设置为 QMGR，那么此属性适用。

CLWLMRUC 队列管理器属性

CLWLMRUC 队列管理器属性设置最近选择的通道数。集群工作负载管理算法使用 CLWLMRUC 来限制活动出站集群通道数。该值必须在范围 1-999 999 999 之间。

CLWLPRTY 通道属性

CLWLPRTY 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 通道属性

CLWLRANK 通道属性指定用于集群工作负载分发的 **CLUSDR** 或 **CLUSRCVR** 通道的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLWGHT 通道属性

CLWLWGHT 通道属性指定应用于集群工作负载分布的 **CLUSDR** 和 **CLUSRCVR** 通道的权重。该值必须在范围 1-99 内，其中 1 是最低权重，而 99 是最高权重。

NETPRTY 通道属性

NETPRTY 通道属性指定 **CLUSRCVR** 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 队列属性

CLWLRANK 队列属性指定用于集群工作负载分发的本地队列，远程队列或别名队列的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

如果要控制发送到另一个集群中的队列管理器的消息的最终目标，请使用 **CLWLRANK** 队列属性。设置 **CLWLRANK** 时，消息将通过相互连接的集群采用指定的路由，以到达更高排名的目标。

例如，您可能定义了两个配置完全相同的网关队列管理器，以提高网关的可用性。假设您已在网关上为集群中定义的本地队列定义了集群别名队列。如果本地队列变为不可用，那么您打算将消息保留在其中一个网关上，以等待队列再次变为可用。要保留网关上的队列，必须定义具有高于网关上的集群别名队列的列组的本地队列。

如果使用与队列别名相同的列组定义本地队列，并且本地队列不可用，那么消息将在网关之间传递。当发现本地队列不可用时，第一个网关队列管理器会将消息路由到另一个网关。另一个网关尝试再次将消息传递到目标本地队列。如果本地队列仍然不可用，那么它会将消息路由回第一个网关。消息将在网关之间来回移动，直到目标本地队列再次可用为止。通过为本地队列提供更高的列组，即使队列不可用，也不会将消息重新路由到较低列组的目标。

WebSphere MQ 在检查通道状态之前获取队列的列组。在检查通道状态之前获取列组意味着即使不可访问的队列也可供选择。它允许通过网络路由消息，即使最终目标不可用也是如此。

如果使用了优先级属性 **WebSphere MQ**，请在可用目标之间进行选择。如果通道对具有最高列组的目标不可用，那么消息将保留在传输队列上。当通道变为可用时，将释放此消息。消息不会按列组顺序发送到下一个可用目标。

相关概念

集群工作负载管理算法

工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

相关参考

CLWLPRTY 队列属性

CLWLPRTY 队列属性指定集群工作负载分布的本地，远程或别名队列的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLUSEQ 队列属性

CLWLUSEQ 队列属性指定是否将队列的本地实例作为目标优先于集群中的其他实例。

CLWLUSEQ 队列管理器属性

CLWLUSEQ 队列管理器属性指定是否将队列的本地实例作为目标优先于集群中队列的其他实例。如果 **CLWLUSEQ** 队列属性设置为 **QMGR**，那么此属性适用。

CLWLMRUC 队列管理器属性

CLWLMRUC 队列管理器属性设置最近选择的通道数。集群工作负载管理算法使用 **CLWLMRUC** 来限制活动出站集群通道数。该值必须在范围 1-999 999 999 之间。

CLWLPRTY 通道属性

CLWLPRTY 通道属性指定用于集群工作负载分发的 **CLUSDR** 或 **CLUSRCVR** 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 通道属性

CLWLRANK 通道属性指定用于集群工作负载分发的 **CLUSDR** 或 **CLUSRCVR** 通道的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLWGHT 通道属性

CLWLWGHT 通道属性指定应用于集群工作负载分布的 CLUSSDR 和 CLUSRCVR 通道的权重。该值必须在范围 1-99 内，其中 1 是最低权重，而 99 是最高权重。

NETPRTY 通道属性

NETPRTY 通道属性指定 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLUSEQ 队列属性

CLWLUSEQ 队列属性指定是否将队列的本地实例作为目标优先于集群中的其他实例。

CLWLUSEQ 队列属性仅对本地队列有效。仅当消息由应用程序或不是集群通道的通道放入时才适用。

Local

本地队列是 MQPUT 的唯一目标，前提是已启用本地队列。MQPUT 行为取决于 [集群工作负载管理](#)。

QMGR

行为由 CLWLUSEQ 队列管理器属性指定。

ANY

MQPUT 将本地队列视为与集群中用于工作负载分发的任何其他队列实例相同。

相关概念

集群工作负载管理算法

工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

相关参考

CLWLPRTY 队列属性

CLWLPRTY 队列属性指定集群工作负载分布的本地，远程或别名队列的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 队列属性

CLWLRANK 队列属性指定用于集群工作负载分发的本地队列，远程队列或别名队列的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLUSEQ 队列管理器属性

CLWLUSEQ 队列管理器属性指定是否将队列的本地实例作为目标优先于集群中队列的其他实例。如果 CLWLUSEQ 队列属性设置为 QMGR，那么此属性适用。

CLWLMRUC 队列管理器属性

CLWLMRUC 队列管理器属性设置最近选择的通道数。集群工作负载管理算法使用 CLWLMRUC 来限制活动出站集群通道数。该值必须在范围 1-999 999 999 之间。

CLWLPRTY 通道属性

CLWLPRTY 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 通道属性

CLWLRANK 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLWGHT 通道属性

CLWLWGHT 通道属性指定应用于集群工作负载分布的 CLUSSDR 和 CLUSRCVR 通道的权重。该值必须在范围 1-99 内，其中 1 是最低权重，而 99 是最高权重。

NETPRTY 通道属性

NETPRTY 通道属性指定 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLUSEQ 队列管理器属性

CLWLUSEQ 队列管理器属性指定是否将队列的本地实例作为目标优先于集群中队列的其他实例。如果 CLWLUSEQ 队列属性设置为 QMGR，那么此属性适用。

CLWLUSEQ 队列属性仅对本地队列有效。仅当消息由应用程序或不是集群通道的通道放入时才适用。

Local

本地队列是 MQPUT 的唯一目标。LOCAL 是缺省值。

ANY

MQPUT 将本地队列视为与集群中用于工作负载分发的任何其他队列实例相同。

相关概念

[集群工作负载管理算法](#)

工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

相关参考

[CLWLPRTY 队列属性](#)

CLWLPRTY 队列属性指定集群工作负载分布的本地、远程或别名队列的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

[CLWLRANK 队列属性](#)

CLWLRANK 队列属性指定用于集群工作负载分发的本地队列、远程队列或别名队列的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

[CLWLUSEQ 队列属性](#)

CLWLUSEQ 队列属性指定是否将队列的本地实例作为目标优先于集群中的其他实例。

[CLWLMRUC 队列管理器属性](#)

CLWLMRUC 队列管理器属性设置最近选择的通道数。集群工作负载管理算法使用 CLWLMRUC 来限制活动出站集群通道数。该值必须在范围 1-999 999 999 之间。

[CLWLPRTY 通道属性](#)

CLWLPRTY 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

[CLWLRANK 通道属性](#)

CLWLRANK 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

[CLWLWGHT 通道属性](#)

CLWLWGHT 通道属性指定应用于集群工作负载分布的 CLUSSDR 和 CLUSRCVR 通道的权重。该值必须在范围 1-99 内，其中 1 是最低权重，而 99 是最高权重。

[NETPRTY 通道属性](#)

NETPRTY 通道属性指定 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLMRUC 队列管理器属性

CLWLMRUC 队列管理器属性设置最近选择的通道数。集群工作负载管理算法使用 CLWLMRUC 来限制活动出站集群通道数。该值必须在范围 1-999 999 999 之间。

初始缺省值为 999 999 999 999。

相关概念

[集群工作负载管理算法](#)

工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

相关参考

[CLWLPRTY 队列属性](#)

CLWLPRTY 队列属性指定集群工作负载分布的本地、远程或别名队列的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

[CLWLRANK 队列属性](#)

CLWLRANK 队列属性指定用于集群工作负载分发的本地队列、远程队列或别名队列的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

[CLWLUSEQ 队列属性](#)

CLWLUSEQ 队列属性指定是否将队列的本地实例作为目标优先于集群中的其他实例。

[CLWLUSEQ 队列管理器属性](#)

CLWLUSEQ 队列管理器属性指定是否将队列的本地实例作为目标优先于集群中队列的其他实例。如果 CLWLUSEQ 队列属性设置为 QMGR，那么此属性适用。

CLWLPRTY 通道属性

CLWLPRTY 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 通道属性

CLWLRANK 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLWGHT 通道属性

CLWLWGHT 通道属性指定应用于集群工作负载分布的 CLUSSDR 和 CLUSRCVR 通道的权重。该值必须在范围 1-99 内，其中 1 是最低权重，而 99 是最高权重。

NETPRTY 通道属性

NETPRTY 通道属性指定 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLPRTY 通道属性

CLWLPRTY 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

使用 CLWLPRTY 通道属性来设置 CLUSSDR 或 CLUSRCVR 通道的首选目标。IBM WebSphere MQ 先选择具有最高优先级的目标，然后再选择具有最低集群目标优先级的目标。如果存在多个具有相同优先级的目标，那么将选择最近最少使用的目标。

如果有两个可能的目标，那么可以使用此属性来允许故障转移。消息转至具有最高优先级通道的队列管理器。如果它变为不可用，那么消息将转至下一个优先级最高的队列管理器。优先级较低的队列管理器充当保留。

WebSphere MQ 在检查通道状态后获取通道的优先级。只有可用的队列管理器才是可供选择的候选者。

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。

相关概念

集群工作负载管理算法

工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

相关参考

CLWLPRTY 队列属性

CLWLPRTY 队列属性指定集群工作负载分布的本地、远程或别名队列的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 队列属性

CLWLRANK 队列属性指定用于集群工作负载分发的本地队列、远程队列或别名队列的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLUSEQ 队列属性

CLWLUSEQ 队列属性指定是否将队列的本地实例作为目标优先于集群中的其他实例。

CLWLUSEQ 队列管理器属性

CLWLUSEQ 队列管理器属性指定是否将队列的本地实例作为目标优先于集群中队列的其他实例。如果 CLWLUSEQ 队列属性设置为 QMGR，那么此属性适用。

CLWLMRUC 队列管理器属性

CLWLMRUC 队列管理器属性设置最近选择的通道数。集群工作负载管理算法使用 CLWLMRUC 来限制活动出站集群通道数。该值必须在范围 1-999 999 999 之间。

CLWLRANK 通道属性

CLWLRANK 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLWGHT 通道属性

CLWLWGHT 通道属性指定应用于集群工作负载分布的 CLUSSDR 和 CLUSRCVR 通道的权重。该值必须在范围 1-99 内，其中 1 是最低权重，而 99 是最高权重。

NETPRTY 通道属性

NETPRTY 通道属性指定 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 通道属性

CLWLRANK 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

如果要控制发送到另一个集群中的队列管理器的消息的最终目标，请使用 CLWLRANK 通道属性。通过设置将队列管理器连接到集群交集处的网关队列管理器的通道的列组来控制最终目标的选择。设置 CLWLRANK 时，消息将通过相互连接的集群采用指定的路由，以到达更高排名的目标。例如，消息到达网关队列管理器，该网关队列管理器可以使用排名 1 和 2 的通道将它们发送到两个队列管理器中的任何一个。它们将自动发送到由具有最高列组的通道连接的队列管理器，在这种情况下，会将该通道发送到排名 2 的队列管理器。

WebSphere MQ 在检查通道状态之前获取通道的列组。在检查通道状态之前获取列组意味着即使不可访问的通道也可供选择。它允许通过网络路由消息，即使最终目标不可用也是如此。

如果使用了优先级属性 WebSphere MQ，请在可用目标之间进行选择。如果通道对具有最高列组的目标不可用，那么消息将保留在传输队列上。当通道变为可用时，将释放此消息。消息不会按列组顺序发送到下一个可用目标。

相关概念

集群工作负载管理算法

工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

相关参考

CLWLPRTY 队列属性

CLWLPRTY 队列属性指定集群工作负载分布的本地、远程或别名队列的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 队列属性

CLWLRANK 队列属性指定用于集群工作负载分发的本地队列、远程队列或别名队列的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLUSEQ 队列属性

CLWLUSEQ 队列属性指定是否将队列的本地实例作为目标优先于集群中的其他实例。

CLWLUSEQ 队列管理器属性

CLWLUSEQ 队列管理器属性指定是否将队列的本地实例作为目标优先于集群中队列的其他实例。如果 CLWLUSEQ 队列属性设置为 QMGR，那么此属性适用。

CLWLMRUC 队列管理器属性

CLWLMRUC 队列管理器属性设置最近选择的通道数。集群工作负载管理算法使用 CLWLMRUC 来限制活动出站集群通道数。该值必须在范围 1-999 999 999 之间。

CLWLPRTY 通道属性

CLWLPRTY 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLWGHT 通道属性

CLWLWGHT 通道属性指定应用于集群工作负载分布的 CLUSSDR 和 CLUSRCVR 通道的权重。该值必须在范围 1-99 内，其中 1 是最低权重，而 99 是最高权重。

NETPRTY 通道属性

NETPRTY 通道属性指定 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLWGHT 通道属性

CLWLWGHT 通道属性指定应用于集群工作负载分布的 CLUSSDR 和 CLUSRCVR 通道的权重。该值必须在范围 1-99 内，其中 1 是最低权重，而 99 是最高权重。

使用 CLWLWGHT 发送具有更多处理能力的服务器消息。通道权重越高，通过该通道发送的消息越多。

相关概念

集群工作负载管理算法

工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

相关参考

CLWLPRTY 队列属性

CLWLPRTY 队列属性指定集群工作负载分布的本地，远程或别名队列的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 队列属性

CLWLRANK 队列属性指定用于集群工作负载分发的本地队列，远程队列或别名队列的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLUSEQ 队列属性

CLWLUSEQ 队列属性指定是否将队列的本地实例作为目标优先于集群中的其他实例。

CLWLUSEQ 队列管理器属性

CLWLUSEQ 队列管理器属性指定是否将队列的本地实例作为目标优先于集群中队列的其他实例。如果 CLWLUSEQ 队列属性设置为 QMGR，那么此属性适用。

CLWLMRUC 队列管理器属性

CLWLMRUC 队列管理器属性设置最近选择的通道数。集群工作负载管理算法使用 CLWLMRUC 来限制活动出站集群通道数。该值必须在范围 1-999 999 999 之间。

CLWLPRTY 通道属性

CLWLPRTY 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 通道属性

CLWLRANK 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

NETPRTY 通道属性

NETPRTY 通道属性指定 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

NETPRTY 通道属性

NETPRTY 通道属性指定 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

使用 NETPRTY 属性使一个网络成为主网络，另一个网络成为备份网络。给定一组排名相同的通道，集群会在多个路径可用时选择具有最高优先级的路径。

使用 NETPRTY 通道属性的典型示例是区分具有不同成本或速度并连接相同目标的网络。

相关概念

集群工作负载管理算法

工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

相关参考

CLWLPRTY 队列属性

CLWLPRTY 队列属性指定集群工作负载分布的本地，远程或别名队列的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 队列属性

CLWLRANK 队列属性指定用于集群工作负载分发的本地队列，远程队列或别名队列的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLUSEQ 队列属性

CLWLUSEQ 队列属性指定是否将队列的本地实例作为目标优先于集群中的其他实例。

CLWLUSEQ 队列管理器属性

CLWLUSEQ 队列管理器属性指定是否将队列的本地实例作为目标优先于集群中队列的其他实例。如果 CLWLUSEQ 队列属性设置为 QMGR，那么此属性适用。

CLWLMRUC 队列管理器属性

CLWLMRUC 队列管理器属性设置最近选择的通道数。集群工作负载管理算法使用 CLWLMRUC 来限制活动出站集群通道数。该值必须在范围 1-999 999 999 之间。

CLWLPRTY 通道属性

CLWLPRTY 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的优先级。该值必须在 0-9 范围内，其中 0 是最低优先级，9 是最高优先级。

CLWLRANK 通道属性

CLWLRANK 通道属性指定用于集群工作负载分发的 CLUSSDR 或 CLUSRCVR 通道的列组。该值必须在 0-9 范围内，其中 0 是最低等级，9 是最高等级。

CLWLWGHT 通道属性

CLWLWGHT 通道属性指定应用于集群工作负载分布的 CLUSSDR 和 CLUSRCVR 通道的权重。该值必须在范围 1-99 内，其中 1 是最低权重，而 99 是最高权重。

集群工作负载出口调用和数据结构

本部分提供了集群工作负载出口和数据结构的参考信息。这是通用编程接口信息。

您可以使用以下编程语言编写集群工作负载出口：

- C
- System/390 汇编程序 (WebSphere MQ for z/OS)

在以下内容中描述了此调用：

- [第 96 页的『MQ_CLUSTER_WORKLOAD_EXIT -调用描述』](#)

出口使用的结构数据类型描述如下：

- [第 97 页的『MQXCLWLN -浏览集群工作负载记录』](#)
- [第 100 页的『MQWXP -集群工作负载出口参数结构』](#)
- [第 107 页的『MQWDR -集群工作负载目标记录结构』](#)
- [第 111 页的『MQWQR -集群工作负载队列记录结构』](#)
- [第 116 页的『MQWCR -集群工作负载集群记录结构』](#)

在此部分中，队列管理器属性和队列属性全部显示。下面显示了 MQSC 命令书中使用的等效名称。有关 MQSC 命令的详细信息，请参阅 [MQSC 参考](#)。

全名	MQSC 中使用的名称
<i>ClusterWorkloadData</i>	CLWLDATA
<i>ClusterWorkloadExit</i>	CLWLEXIT
<i>ClusterWorkloadLength</i>	CLWLLEN

全名	MQSC 中使用的名称
<i>DefBind</i>	DEFBIND
<i>DefPersistence</i>	DEFPSIST

表 22: 队列属性 (继续)	
全名	MQSC 中使用的名称
<i>DefPriority</i>	DEFPRTY
<i>InhibitPut</i>	PUT
<i>QDesc</i>	DESCR

相关概念

工作负载均衡

如果集群包含同一队列的多个实例，那么 WebSphere MQ 会选择要将消息路由到的队列管理器。它使用集群工作负载管理算法来确定要使用的最佳队列管理器。您可以提供工作负载均衡算法，以通过编写集群工作负载出口程序来选择队列管理器。

MQ_CLUSTER_WORKLOAD_EXIT -调用描述

队列管理器调用集群工作负载出口以将消息路由到可用队列管理器。

注: 队列管理器未提供名为 MQ_CLUSTER_WORKLOAD_EXIT 的入口点。而是由 ClusterWorkloadExit 队列管理器属性定义集群工作负载出口的名称。

MQ_CLUSTER_WORKLOAD_EXIT 出口在所有平台上都受支持。

语法

MQ_CLUSTER_WORKLOAD_EXIT (*ExitParms*)

相关参考

[MQXCLWLN -浏览集群工作负载记录](#)

MQXCLWLN 调用用于浏览存储在集群高速缓存中的 MQWDR, MQWQR 和 MQWCR 记录的链。

[MQWXP -集群工作负载出口参数结构](#)

下表汇总了 MQWXP -集群工作负载出口参数结构中的字段。

[MQWDR-集群工作负载目标记录结构](#)

下表汇总了 MQWDR -集群工作负载目标记录结构中的字段。

[MQWQR -集群工作负载队列记录结构](#)

下表汇总了 MQWQR -集群工作负载队列记录结构中的字段。

[MQWCR -集群工作负载集群记录结构](#)

下表汇总了 MQWCR 集群工作负载记录结构中的字段。

参数 MQ_CLUSTER_WORKLOAD_EXIT

MQ_CLUSTER_WORKLOAD_EXIT 调用中参数的描述。

ExitParms(MQWXP)-输入/输出

出口参数块。

- 出口设置 MQWXP 中的信息以指示如何管理工作负载。

相关参考

[使用说明](#)

集群工作负载出口所执行的功能由该出口的提供程序定义。但是，出口必须符合关联控制块 MQWXP 中定义的规则。

[MQ_CLUSTER_WORKLOAD_EXIT 的语言调用](#)

MQ_CLUSTER_WORKLOAD_EXIT 支持两种语言 :C 和 High Level Assembler。

使用说明

集群工作负载出口所执行的功能由该出口的提供程序定义。但是，出口必须符合关联控制块 MQWXP 中定义的规则。

队列管理器未提供名为 MQ_CLUSTER_WORKLOAD_EXIT 的入口点。但是，在 C 编程语言中为名称 MQ_CLUSTER_WORKLOAD_EXIT 提供了 typedef。使用 typedef 来声明用户编写的出口，以确保参数正确。

相关参考

[参数 MQ_CLUSTER_WORKLOAD_EXIT](#)

MQ_CLUSTER_WORKLOAD_EXIT 调用中参数的描述。

[MQ_CLUSTER_WORKLOAD_EXIT 的语言调用](#)

MQ_CLUSTER_WORKLOAD_EXIT 支持两种语言 :C 和 High Level Assembler。

MQ_CLUSTER_WORKLOAD_EXIT 的语言调用

MQ_CLUSTER_WORKLOAD_EXIT 支持两种语言 :C 和 High Level Assembler。

C 调用

```
MQ_CLUSTER_WORKLOAD_EXIT (&ExitParms);
```

将 MQ_CLUSTER_WORKLOAD_EXIT 替换为集群工作负载出口函数的名称。

按如下所示声明 MQ_CLUSTER_WORKLOAD_EXIT 参数:

```
MQWXP ExitParms; /* Exit parameter block */
```

高级汇编程序调用

```
CALL EXITNAME, (EXITPARMS)
```

按如下所示声明参数:

```
EXITPARMS          CMQWXP          Exit parameter block
```

相关参考

[参数 MQ_CLUSTER_WORKLOAD_EXIT](#)

MQ_CLUSTER_WORKLOAD_EXIT 调用中参数的描述。

使用说明

集群工作负载出口所执行的功能由该出口的提供程序定义。但是，出口必须符合关联控制块 MQWXP 中定义的规则。

MQXCLWLN -浏览集群工作负载记录

MQXCLWLN 调用用于浏览存储在集群高速缓存中的 MQWDR, MQWQR 和 MQWCR 记录的链。

集群高速缓存是一个主存储器区域，用于存储与集群相关的信息。

如果集群高速缓存是静态的，那么它具有固定大小。如果将其设置为动态，那么可以根据需要扩展集群高速缓存。

使用系统参数或宏将集群高速缓存类型设置为 STATIC 或 DYNAMIC。

- 除 z/OS 以外的平台上的系统参数 ClusterCacheType
- z/OS 上的 CSQ6SYSP 宏中的 CLCACHE 参数。

语法

```
MQXCLWLN (ExitParms, CurrentRecord, NextOffset, NextRecord, Compcode, Reason)
```

相关参考

MQ_CLUSTER_WORKLOAD_EXIT -调用描述

队列管理器调用集群工作负载出口以将消息路由到可用队列管理器。

MQWXP -集群工作负载出口参数结构

下表汇总了 MQWXP -集群工作负载出口参数结构中的字段。

MQWDR -集群工作负载目标记录结构

下表汇总了 MQWDR -集群工作负载目标记录结构中的字段。

MQWQR -集群工作负载队列记录结构

下表汇总了 MQWQR -集群工作负载队列记录结构中的字段。

MQWCR -集群工作负载集群记录结构

下表汇总了 MQWCR 集群工作负载记录结构中的字段。

MQXCLWLN 的参数-浏览集群工作负载记录

MQXCLWLN 调用中参数的描述。

ExitParms (MQWXP)-输入/输出

出口参数块。

此结构包含与出口调用相关的信息。出口设置此结构中的信息以指示如何管理工作负载。

CurrentRecord (MQPTR)-输入

当前记录的地址。

此结构包含与出口当前正在检查的记录的地址相关的信息。该记录必须是下列其中一种类型:

- 集群工作负载目标记录 (MQWDR)
- 集群工作负载队列记录 (MQWQR)
- 集群工作负载集群记录 (MQWCR)

NextOffset (MQLONG)-输入

下一条记录的偏移量。

此结构包含与下一个记录或结构的偏移量相关的信息。 *NextOffset* 是当前记录中相应偏移量字段的值，并且必须是下列其中一个字段:

- MQWDR 中的 ChannelDef 偏移量 字段
- MQWDR 中的 ClusterRec 偏移量 字段
- MQWQR 中的 ClusterRec 偏移量 字段
- MQWCR 中的 ClusterRec 偏移量 字段

NextRecord (MQPTR)-输出

下一个记录或结构的地址。

此结构包含与下一个记录或结构的地址相关的信息。如果 *CurrentRecord* 是 MQWDR 的地址，而 *NextOffset* 是 ChannelDefOffset 字段的值，那么 *NextRecord* 是通道定义结构 (MQCD) 的地址。

如果没有下一个记录或结构，那么队列管理器会将 *NextRecord* 设置为空指针，并且调用会返回完成代码 MQCC_WARNING 和原因码 MQRC_NO_RECORD_AVAILABLE。

CompCode (MQLONG)-输出

完成代码。

完成代码具有下列其中一个值:

MQCC_OK

成功完成。

MQCC_WARNING

警告（部分完成）。

MQCC_FAILED

调用失败。

Reason (MQLONG)-输出

原因码限定 CompCode

如果 CompCode 为 MQCC_OK:

MQRC_NONE

(0, X'0000')

没有报告的理由。

如果 CompCode 为 MQCC_WARNING:

MQRC_NO_RECORD_AVAILABLE

(2359, X'0937')

没有可用的记录。从集群工作负载出口发出了 MQXCLWLN 调用，以获取链中下一条记录的地址。当前记录是链中的最后一条记录。更正操作: 无。

如果 CompCode 为 MQCC_FAILED:

MQRC_CURRENT_RECORD_ERROR

(2357, X'0935')

CurrentRecord 参数无效。从集群工作负载出口发出了 MQXCLWLN 调用，以获取链中下一条记录的地址。*CurrentRecord* 参数指定的地址不是有效记录的地址。

CurrentRecord 必须是驻留在集群高速缓存中的目标记录 MQWDR，队列记录 (MQWQR) 或集群记录 (MQWCR) 的地址。更正操作: 确保集群工作负载出口传递驻留在集群高速缓存中的有效记录的地址。

MQRC_ENVIRONMENT_ERROR

(2012, X'07DC')

调用在环境中无效。已发出 MQXCLWLN 调用，但未从集群工作负载出口发出。

MQRC_NEXT_OFFSET_ERROR

(2358, X'0936')

NextOffset 参数无效。从集群工作负载出口发出了 MQXCLWLN 调用，以获取链中下一条记录的地址。*NextOffset* 参数指定的偏移量无效。*NextOffset* 必须是以下字段之一的值:

- MQWDR 中的 ChannelDef 偏移量 字段
- MQWDR 中的 ClusterRec 偏移量 字段
- MQWQR 中的 ClusterRec 偏移量 字段
- MQWCR 中的 ClusterRec 偏移量 字段

更正操作: 确保为 *NextOffset* 参数指定的值是先前列示的其中一个字段的值。

MQRC_NEXT_RECORD_ERROR

(2361, X'0939')

NextRecord 参数无效。

MQRC_WXP_ERROR

(2356, X'0934')

工作负载出口参数结构无效。从集群工作负载出口发出了 MQXCLWLN 调用，以获取链中下一条记录的地址。由于下列其中一个原因，工作负载出口参数结构 *ExitParms* 无效:

- 参数指针无效。并非总是能够检测到无效的参数指针; 如果未检测到，那么会发生不可预测的结果。
- StrucId 字段不是 MQWXP_STRUC_ID。
- 版本 字段不是 MQWXP_VERSION_2。
- 上下文 字段不包含队列管理器传递到出口的值。

更正操作: 确保为 *ExitParms* 指定的参数是调用出口时传递到出口的 MQWXP 结构。

相关参考

[MQXCLWLN 的使用说明-浏览集群工作负载记录](#)

[使用 MQXCLWLN 浏览集群记录，即使高速缓存是静态的也是如此。](#)

[MQXCLWLN 的语言调用](#)

MQXCLWLN 支持两种语言 :C 和 High Level Assembler。

MQXCLWLN 的使用说明-浏览集群工作负载记录

使用 MQXCLWLN 浏览集群记录，即使高速缓存是静态的也是如此。

如果集群高速缓存是动态的，那么必须使用 MQXCLWLN 调用来浏览记录。如果使用简单的指针和偏移算术来浏览记录，那么出口将异常结束。

如果集群高速缓存是静态的，那么无需使用 MQXCLWLN 来浏览记录。通常，即使高速缓存是静态的，也会使用 MQXCLWLN。然后，可以将集群高速缓存更改为动态高速缓存，而无需更改工作负载出口。

相关参考

[MQXCLWLN 的参数-浏览集群工作负载记录](#)

MQXCLWLN 调用中参数的描述。

MQXCLWLN 的语言调用

MQXCLWLN 支持两种语言 :C 和 High Level Assembler。

MQXCLWLN 的语言调用

MQXCLWLN 支持两种语言 :C 和 High Level Assembler。

C 调用

```
MQXCLWLN (&ExitParms, CurrentRecord, NextOffset, &NextRecord, &CompCode, &Reason) ;
```

按如下所示声明参数：

```
Typedef struct tagMQXCLWLN {
    MQWXP      ExitParms;           /* Exit parameter block */
    MQPTR      CurrentRecord;       /* Address of current record*/
    MQLONG     NextOffset;         /* Offset of next record */
    MQPTR      NextRecord;         /* Address of next record or structure */
    MQLONG     CompCode;           /* Completion code */
    MQLONG     Reason;             /* Reason code qualifying CompCode */
}
```

高级汇编程序调用

```
CALL MQXCLWLN, (CLWLEXITPARMS, CURRENTRECORD, NEXTOFFSET, NEXTRECORD, COMPCODE, REASON)
```

按如下所示声明参数：

CLWLEXITPARMS	CMQWXP,	Cluster workload exit parameter block
CURRENTRECORD	CMQWDRA,	Current record
NEXTOFFSET	DS F	Next offset
NEXTRECORD	DS F	Next record
COMPCODE	DS F	Completion code
REASON	DS F	Reason code qualifying COMPCODE

相关参考

[MQXCLWLN 的参数-浏览集群工作负载记录](#)

MQXCLWLN 调用中参数的描述。

MQXCLWLN 的使用说明-浏览集群工作负载记录

使用 MQXCLWLN 浏览集群记录，即使高速缓存是静态的也是如此。

MQWXP - 集群工作负载出口参数结构

下表汇总了 MQWXP - 集群工作负载出口参数结构中的字段。

表 23: MQWXP 中的字段		
字段	描述	页面
<i>StrucId</i>	结构标识	StrucId
<i>Version</i>	结构版本号	版本

表 23: MQWXP 中的字段 (继续)

字段	描述	页面
<i>ExitId</i>	出口类型	ExitId
<i>ExitReason</i>	调用出口的原因	ExitReason
<i>ExitResponse</i>	来自出口的响应	ExitResponse
<i>ExitResponse2</i>	来自出口的辅助响应	ExitResponse2
<i>Feedback</i>	反馈代码	反馈
<i>Flags</i>	标记值。这些位标志用于指示有关要放入的消息的信息	标记
<i>ExitUserArea</i>	出口用户区域	ExitUserArea
<i>ExitData</i>	出口数据	ExitData
<i>MsgDescPtr</i>	消息描述符 (MQMD) 的地址	MsgDescPtr
<i>MsgBufferPtr</i>	包含部分或全部消息数据的缓冲区的地址	MsgBufferPtr
<i>MsgBufferLength</i>	包含消息数据的缓冲区的长度	MsgBuffer 长度
<i>MsgLength</i>	完整消息的长度	MsgLength
<i>QName</i>	队列名称	QName
<i>QMgrName</i>	本地队列管理器的名称	QMgrName
<i>DestinationCount</i>	可能的目标数	DestinationCount
<i>DestinationChosen</i>	已选择目标	DestinationChosen
<i>DestinationArrayPtr</i>	指向目标记录的指针数组的地址 (MQWDR)	DestinationArrayPtr
<i>QArrayPtr</i>	指向队列记录的指针数组的地址 (MQWQR)	QArrayPtr
注: 如果版本小于 MQWXP_VERSION_2, 那么将忽略其余字段。		
<i>CacheContext</i>	上下文信息	CacheContext
<i>CacheType</i>	集群高速缓存的类型	CacheType
注: 如果版本小于 MQWXP_VERSION_3, 那么将忽略其余字段。		
<i>CLWLMRUChannels</i>	允许的最大活动出站集群通道数	CLWLMRUChannels
注: 如果版本小于 MQWXP_VERSION_4, 那么将忽略其余字段。		
<i>pEntryPoints</i>	允许进行 MQI 和 DCI 调用的 MQIEP 结构的地址	pEntry 点

集群工作负载出口参数结构描述传递到集群工作负载出口的信息。

集群工作负载出口参数结构在所有平台上都受支持

此外, MQWXP1, MQWXP2 和 MQWXP3 结构可用于向后兼容性。

相关参考

[MQ_CLUSTER_WORKLOAD_EXIT](#) -调用描述

队列管理器调用集群工作负载出口以将消息路由到可用队列管理器。

[MQXCLWLN](#) -浏览集群工作负载记录

[MQXCLWLN](#) 调用用于浏览存储在集群高速缓存中的 MQWDR, MQWQR 和 MQWCR 记录的链。

[MQWDR](#)-集群工作负载目标记录结构

下表汇总了 MQWDR -集群工作负载目标记录结构中的字段。

MQWQR -集群工作负载队列记录结构

下表汇总了 MQWQR -集群工作负载队列记录结构中的字段。

MQWCR -集群工作负载集群记录结构

下表汇总了 MQWCR 集群工作负载记录结构中的字段。

MQWXP 中的字段-集群工作负载出口参数结构

MQWXP -集群工作负载出口参数结构中字段的描述

StrucId (MQCHAR4)-输入

集群工作负载出口参数结构的结构标识。

- StrucId 值为 MQWXP_STRUC_ID。
- 对于 C 编程语言，还定义了常量 MQWXP_STRUC_ID_ARRAY。它具有与 MQWXP_STRUC_ID 相同的值。它是一个字符数组，而不是字符串。

V (MQLONG)-输入

指示结构版本号。Version 采用下列其中一个值：

MQWXP_VERSION_1

Version-1 集群工作负载出口参数结构。

MQWXP_VERSION_1 在所有环境中都受支持。

MQWXP_VERSION_2

Version-2 集群工作负载出口参数结构。

MQWXP_VERSION_2 在以下环境中受支持: AIX, HP-UX, Linux, IBM i, Solaris 和 Windows。

MQWXP_VERSION_3

Version-3 集群工作负载出口参数结构。

MQWXP_VERSION_3 在以下环境中受支持: AIX, HP-UX, Linux, IBM i, Solaris 和 Windows。

MQWXP_VERSION_4

Version-4 集群工作负载出口参数结构。

MQWXP_VERSION_4 在以下环境中受支持: AIX, HP-UX, Linux, IBM i, Solaris 和 Windows。

MQWXP_CURRENT_VERSION

集群工作负载出口参数结构的当前版本。

ExitId (MQLONG)-输入

指示正在调用的出口的类型。集群工作负载出口是唯一受支持的出口。

- ExitId 值必须为 MQXT_CLUSTER_WORKLOAD_EXIT

ExitReason (MQLONG)-输入

指示调用集群工作负载出口的原因。ExitReason 采用下列其中一个值：

MQXR_INIT

指示正在首次调用出口。

获取并初始化出口可能需要的任何资源，例如主存储器。

MQXR_TERM

指示出口即将终止。

释放出口自初始化以来可能已获取的任何资源，例如主存储器。

MQXR_CLWL_OPEN

由 MQOPEN 调用。

MQXR_CLWL_PUT

由 MQPUT 或 MQPUT1 调用。

MQXR_CLWL_MOVE

当通道状态已更改时由 MCA 调用。

MQXR_CLWL_REPOS

由 MQPUT 或 MQPUT1 针对 repository-manager PCF 消息调用。

MQXR_CLWL_REPOS_MOVE

由 MCA 针对存储库管理器 PCF 消息调用 (如果通道状态已更改)。

ExitResponse (MQLONG)-输出

设置 ExitResponse 以指示是否继续处理消息。它必须是下列其中一个值：

MQXCC_OK

继续正常处理消息。

- DestinationChosen 标识要将消息发送到的目标。

MQXCC_SUPPRESS_FUNCTION

停止处理消息。

- 队列管理器执行的操作取决于调用出口的原因：

表 24: 队列管理器执行的操作	
ExitReason	执行的操作
<ul style="list-style-type: none"> - MQXR_CLWL_OPEN - MQXR_CLWL_REPOS - MQXR_CLWL_PUT 	MQOPEN, MQPUT 或 MQPUT1 调用失败, 完成代码为 MQCC_FAILED, 原因码为 MQRC_STOPPED_BY_CLUSTER_EXIT。
<ul style="list-style-type: none"> - MQXR_CLWL_MOVE - MQXR_CLWL_REPOS_MOVE 	消息放置在死信队列上。

MQXCC_SUPPRESS_EXIT

继续正常处理当前消息。在队列管理器关闭之前, 请勿再次调用出口。

队列管理器会处理后续消息, 就像 ClusterWorkloadExit 队列管理器属性为空一样。DestinationChosen 标识将当前消息发送到的目标。

任何其他值

如同指定了 MQXCC_SUPPRESS_FUNCTION 一样处理消息。

ExitResponse2 (MQLONG)-输入/输出

设置 ExitResponse2 以向队列管理器提供更多信息。

- MQXR2_STATIC_CACHE 是缺省值, 在进入出口时设置。
- 当 ExitReason 具有值 MQXR_INIT 时, 出口可以在 ExitResponse2 中设置下列其中一个值:

MQXR2_STATIC_CACHE

出口需要静态集群高速缓存。

- 如果集群高速缓存是静态的, 那么出口无需使用 MQXCLWLN 调用来浏览集群高速缓存中的记录链。
- 如果集群高速缓存是动态的, 那么出口无法正确浏览高速缓存中的记录。

注: 队列管理器处理来自 MQXR_INIT 调用的返回, 就像出口在 ExitResponse 字段中返回了 MQXCC_SUPPRESS_EXIT 一样。

MQXR2_DYNAMIC_CACHE

该出口可以使用静态或动态高速缓存进行操作。

- 如果该出口返回此值, 那么该出口必须使用 MQXCLWLN 调用来浏览集群高速缓存中的记录链。

反馈 (MQLONG)-输入

保留字段。值为零。

标志 (MQLONG)-输入

指示有关正在放入的消息的信息。

- 标志 的值为 MQWXP_PUT_BY_CLUSTER_CHL。消息源自集群通道, 而不是本地或非集群通道。换言之, 消息来自另一个集群队列管理器。

保留 (MQLONG)-输入

保留字段。 值为零。

ExitUser 区域 (MQBYTE16)-输入/输出

设置 ExitUser 区域 以在对出口的调用之间进行通信。

- ExitUser 区域 在第一次调用出口之前初始化为二进制零。 对于在 MQCONN 调用和匹配的 MQDISC 调用之间进行的出口调用，将保留该出口对此字段所作的任何更改。 发生 MQDISC 调用时，该字段将重置为二进制零。
- 出口的第一次调用由具有值 MQXR_INIT 的 ExitReason 字段指示。
- 定义了以下常量：

MQXUA_NONE -字符串

MQXUA_NONE_ARRAY -字符数组

无用户信息。 对于字段的长度，两个常量都是二进制零。

MQ_EXIT_USER_AREA_LENGTH

ExitUser 区域的长度。

ExitData (MQCHAR32)-输入

ClusterWorkload 数据 队列管理器属性的值。 如果未为该属性定义值，那么此字段全部是空白。

- ExitData 的长度由 MQ_EXIT_DATA_LENGTH 给出。

MsgDescPtr (PMQMD)-输入

正在处理的消息的消息描述符 (MQMD) 副本的地址。

- 队列管理器将忽略出口对消息描述符所作的任何更改。
- 如果 ExitReason 具有下列其中一个值 MsgDescPtr 设置为空指针，并且没有消息描述符传递到出口：
 - MQXR_INIT
 - MQXR_TERM
 - MQXR_CLWL_OPEN

MsgBufferPtr (PMQVOID)-输入

包含消息数据的第一个 MsgBufferLength 字节的副本的缓冲区地址。

- 队列管理器将忽略出口对消息数据所作的任何更改。
- 在以下情况下，不会将任何消息数据传递到出口：
 - MsgDescPtr 是空指针。
 - 消息没有数据。
 - ClusterWorkloadLength 队列管理器属性为零。

在这些情况下，MsgBufferPtr 是空指针。

MsgBuffer 长度 (MQLONG)-输入

包含传递到出口的消息数据的缓冲区的长度。

- 长度由 ClusterWorkloadLength 队列管理器属性控制。
- 长度可能小于完整消息的长度，请参阅 MsgLength。

MsgLength (MQLONG)-输入

传递到出口的完整消息的长度。

- MsgBufferLength 可能小于完整消息的长度。
- 如果 ExitReason 为 MQXR_INIT，MQXR_TERM 或 MQXR_CLWL_OPEN，那么 MsgLength 为零。

QName (MQCHAR48)-输入

这是目标队列的名称。 该队列是集群队列。

- QName 的长度为 MQ_Q_NAME_LENGTH。

QMgrName (MQCHAR48)-输入

已调用集群工作负载出口的本地队列管理器的名称。

- QMgrName 的长度为 MQ_Q_MGR_NAME_LENGTH。

DestinationCount (MQLONG)-输入

可能的目标数。目标是目标队列的实例，由目标记录描述。

- 目标记录是 MQWDR 结构。队列的每个实例的每个可能路径都有一个结构。
- MQWDR 结构由指针数组寻址，请参阅 DestinationArrayPtr。

DestinationChosen (MQLONG)-输入/输出

所选目标。

- MQWDR 结构的编号，用于标识要在其中发送消息的路由和队列实例。
- 该值在范围 1- DestinationCount 内。
- 在输入到出口时，DestinationChosen 指示队列管理器已选择的路由和队列实例。出口可以接受此选项，也可以选择其他路由和队列实例。
- 出口设置的值必须在范围 1- DestinationCount 内。如果返回任何其他值，那么队列管理器将在输入到出口时使用 DestinationChosen 的值。

DestinationArrayPtr (PPMQWDR)-输入

指向目标记录的指针数组的地址 (MQWDR)。

- 存在 DestinationCount 个目标记录。

QArrayPtr (PPMQQR)-输入

指向队列记录的指针数组的地址 (MQWQR)。

- 如果队列记录可用，那么其中存在 DestinationCount 个记录。
- 如果没有可用的队列记录，那么 QArrayPtr 是空指针。

注: QArrayPtr 可以是空指针，即使 DestinationCount 大于零也是如此。

CacheContext (MQPTR): V 2-输入

CacheContext 字段保留供队列管理器使用。出口不得改变此字段的值。

CacheType (MQLONG): V 2-输入

集群高速缓存具有下列其中一种类型:

MQCLCT_STATIC

高速缓存是静态的。

- 高速缓存的大小是固定的，不能随着队列管理器的运行而增大。
- 您不需要使用 MQXCLWLN 调用来浏览此类型高速缓存中的记录。

MQCLCT_DYNAMIC

高速缓存是动态的。

- 为了适应不同的集群信息，可以增大高速缓存的大小。
- 您必须使用 MQXCLWLN 调用来浏览此类型高速缓存中的记录。

CLWLMRUChannels (MQLONG): V 3-输入

指示要考虑由集群工作负载选择算法使用的最大活动出站集群通道数。

- CLWLMRUChannels 是值 1-999 999 999。

pEntryPoints (PMQIEP): V 4

MQIEP 结构的地址，可通过该结构进行 MQI 和 DCI 调用。

MQWXP 的初始值和语言声明

MQWXP -集群工作负载出口参数结构的初始值以及 C 和 High Level Assembler 语言声明。

表 25: MQWXP 中字段的初始值

字段名称	常量的名称	常量值
<i>StrucId</i>	MQWXP_STRUC_ID	'WXP? '
<i>Version</i>	MQWXP_VERSION_2	2
<i>ExitId</i>	None	0
<i>ExitReason</i>	MQXCC_OK	0
<i>ExitResponse</i>	None	0
<i>ExitResponse2</i>	None	0
<i>Flags</i>	None	0
<i>ExitUserArea</i>	{MQXUA_NONE_ARRAY}	0
<i>ExitData</i>	None	""
<i>MsgDescPtr</i>	None	NULL
<i>MsgBufferPtr</i>	None	NULL
<i>MsgBufferLength</i>	None	0
<i>MsgBufferPtr</i>	None	0
<i>QName</i>	None	""
<i>QMgrName</i>	None	""
<i>DestinationCount</i>	None	0
<i>DestinationChosen</i>	None	0
<i>DestinationArrayPtr</i>	None	NULL
<i>QArrayPtr</i>	None	NULL
<i>CacheContext</i>	None	NULL
<i>CacheType</i>	MQCLCT_DYNAMIC	1
<i>CLWLMRUChannels</i>	None	0
<i>pEntryPoints</i>	None	NULL

注意:

1. 符号 ? 表示单个空白字符。
2. 在 C 编程语言中, 宏变量 MQWXP_DEFAULT 包含缺省值。通过以下方式使用它来为结构中的字段提供初始值:

```
MQWDR MyWXP = {MQWXP_DEFAULT};
```

C 声明

```
typedef struct tagMQWXP {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    ExitId;           /* Type of exit */
    MQLONG    ExitReason;       /* Reason for invoking exit */
    MQLONG    ExitResponse;     /* Response from exit */
    MQLONG    ExitResponse2;    /* Reserved */
    MQLONG    Feedback;        /* Reserved */
};
```

```

MQLONG    Flags;                /* Flags */
MQBYTE16  ExitUserArea;        /* Exit user area */
MQCHAR32  ExitData;            /* Exit data */
PMQMD     MsgDescPtr;          /* Address of message descriptor */
PMQVOID   MsgBufferPtr;        /* Address of buffer containing some
                                or all of the message data */

MQLONG    MsgBufferLength;     /* Length of buffer containing message
                                data */

MQLONG    MsgLength;           /* Length of complete message */
MQCHAR48  QName;               /* Queue name */
MQCHAR48  QMgrName;           /* Name of local queue manager */
MQLONG    DestinationCount;    /* Number of possible destinations */
MQLONG    DestinationChosen;   /* Destination chosen */
PPMQWDR   DestinationArrayPtr; /* Address of an array of pointers to
                                destination records */

PPMQWQR   QArrayPtr;          /* Address of an array of pointers to
                                queue records */

/* version 1 */
MQPTR     CacheContext;        /* Context information */
MQLONG    CacheType;           /* Type of cluster cache */
/* version 2 */
MQLONG    CLWLMRUChannels;     /* Maximum number of most recently
                                used cluster channels */

/* version 3 */
PMQIEP    pEntryPoints;        /* Address of the MQIEP structure */
/* version 4 */
};

```

High Level Assembler

```

MQWXP                                DSECT
MQWXP_STRUCID                        DS    CL4      Structure identifier
MQWXP_VERSION                        DS    F        Structure version number
MQWXP_EXITID                         DS    F        Type of exit
MQWXP_EXITREASON                     DS    F        Reason for invoking exit
MQWXP_EXITRESPONSE                   DS    F        Response from exit
MQWXP_EXITRESPONSE2                 DS    F        Reserved
MQWXP_FEEDBACK                       DS    F        Reserved
MQWXP_RESERVED                       DS    F        Reserved
MQWXP_EXITUSERAREA                   DS    XL16     Exit user area
MQWXP_EXITDATA                       DS    CL32     Exit data
MQWXP_MSGDESCPTR                     DS    F        Address of message
*                                     descriptor
MQWXP_MSGBUFFERPTR                   DS    F        Address of buffer containing
*                                     some or all of the message
*                                     data
MQWXP_MSGBUFFERLENGTH                DS    F        Length of buffer containing
*                                     message data
MQWXP_MSGLength                       DS    F        Length of complete message
MQWXP_QNAME                           DS    CL48     Queue name
MQWXP_QMGRNAME                       DS    CL48     Name of local queue manager
MQWXP_DESTINATIONCOUNT              DS    F        Number of possible
*                                     destinations
MQWXP_DESTINATIONCHOSEN               DS    F        Destination chosen
MQWXP_DESTINATIONARRAYPTR            DS    F        Address of an array of
*                                     pointers to destination
*                                     records
MQWXP_QARRAYPTR                      DS    F        Address of an array of
*                                     pointers to queue records
MQWXP_CACHECONTEXT                   DS    F        Context information
MQWXP_CACHETYPE                       DS    F        Type of cluster cache
MQWXP_CLWLMRUCHANNELS                DS    F        Number of most recently used
*                                     channels for workload balancing

MQWXP_LENGTH                          EQU    *-MQWXP Length of structure
MQWXP_AREA                             ORG    MQWXP
                                         DS    CL(MQWXP_LENGTH)

```

MQWDR-集群工作负载目标记录结构

下表汇总了 MQWDR -集群工作负载目标记录结构中的字段。

表 26: MQWDR 中的字段		
字段	描述	页面
<i>StrucId</i>	结构标识	StrucId
<i>Version</i>	结构版本号	版本
<i>StrucLength</i>	MQWDR 结构的长度	StrucLength
<i>QMgrFlags</i>	队列管理器标志	QMgrFlags
<i>QMgrIdentifier</i>	队列管理器标识	QMgrIdentifier
<i>QMgrName</i>	队列管理器名称	QMgrName
<i>ClusterRecOffset</i>	第一个集群记录的逻辑偏移量 (MQWCR)	ClusterRec 偏移量
<i>ChannelState</i>	通道状态	ChannelState
<i>ChannelDefOffset</i>	通道定义结构的逻辑偏移量 (MQCD)	ChannelDef 偏移量
注: 如果版本小于 MQWDR_VERSION_2, 那么将忽略其余字段。		
<i>DestSeqNumber</i>	通道目标序号	DestSeq 编号
<i>DestSeqFactor</i>	用于加权的通道目标序列因子	DestSeq 因子

集群工作负载目标记录结构包含与消息的可能目标之一相关的信息。目标队列的每个实例都有一个集群工作负载目标记录结构。

集群工作负载目标记录结构在所有环境中都受支持。

此外, MQWDR1 和 MQWDR2 结构可用于向后兼容性。

相关参考

[MQ_CLUSTER_WORKLOAD_EXIT](#) -调用描述

队列管理器调用集群工作负载出口以将消息路由到可用队列管理器。

[MQXCLWLN](#) -浏览集群工作负载记录

[MQXCLWLN](#) 调用用于浏览存储在集群高速缓存中的 MQWDR, MQWQR 和 MQWCR 记录的链。

[MQWXP](#) -集群工作负载出口参数结构

下表汇总了 [MQWXP](#) -集群工作负载出口参数结构中的字段。

[MQWQR](#) -集群工作负载队列记录结构

下表汇总了 [MQWQR](#) -集群工作负载队列记录结构中的字段。

[MQWCR](#) -集群工作负载集群记录结构

下表汇总了 [MQWCR](#) 集群工作负载记录结构中的字段。

MQWDR-集群工作负载目标记录结构中的字段

MQWDR -集群工作负载目标记录结构中参数的描述。

StrucId (MQCHAR4)-输入

集群工作负载目标记录结构的结构标识。

- StrucId 值为 MQWDR_STRUC_ID。
- 对于 C 编程语言, 还定义了常量 MQWDR_STRUC_ID_ARRAY。它具有与 MQWDR_STRUC_ID 相同的值。它是一个字符数组, 而不是字符串。

V (MQLONG)-输入

结构版本号。Version 采用下列其中一个值:

MQWDR_VERSION_1

Version-1 集群工作负载目标记录。

MQWDR_VERSION_2

Version-2 集群工作负载目标记录。

MQWDR_CURRENT_VERSION

集群工作负载目标记录的 当前版本。

StrucLength (MQLONG)-输入

MQWDR 结构的长度。StrucLength 采用下列其中一个值:

MQWDR_LENGTH_1

version-1 集群工作负载目标记录的长度。

MQWDR_LENGTH_2

version-2 集群工作负载目标记录的长度。

MQWDR_CURRENT_LENGTH

集群工作负载目标记录的当前版本的长度。

QMgrFlags (MQLONG)-输入

队列管理器标志, 指示托管 MQWDR 结构所描述的目标队列实例的队列管理器的属性。定义了以下标志:

MQQMF_REPOSITORY_Q_MGR

目标是完整的存储库队列管理器。

MQQMF_CLUSSDR_USER_DEFINED

已手动定义集群发送方通道。

MQQMF_CLUSSDR_AUTO_DEFINED

已自动定义集群发送方通道。

MQQMF_AVAILABLE

目标队列管理器可用于接收消息。

其他值

此字段中的其他标志可能由队列管理器设置以用于内部目的。

QMgrIdentifier (MQCHAR48)-输入

队列管理器标识是用于托管由 MQWDR 结构描述的目标队列实例的队列管理器的唯一标识。

- 此标识由队列管理器生成。
- QMgrIdentifier 的长度为 MQ_Q_MGR_IDENTIFIER_LENGTH。

QMgrName (MQCHAR48)-输入

托管 MQWDR 结构所描述的目标队列实例的队列管理器的名称。

- QMgrName 可以是本地队列管理器的名称, 也可以是集群中的另一个队列管理器的名称。
- QMgrName 的长度为 MQ_Q_MGR_NAME_LENGTH。

ClusterRec 偏移量 (MQLONG)-输入

属于 MQWDR 结构的第一个 MQWCR 结构的逻辑偏移量。

- 对于静态高速缓存, ClusterRecOffset 是属于 MQWDR 结构的第一个 MQWCR 结构的偏移量。
- 偏移量以从 MQWDR 结构开始的字节为单位进行测量。
- 请勿将逻辑偏移用于具有动态高速缓存的指针算术。要获取下一条记录的地址, 必须使用 MQXCLWLN 调用。

ChannelState (MQLONG)-输入

将本地队列管理器链接到由 MQWDR 结构标识的队列管理器的通道的状态。可能的值如下所示:

MQCHS_BINDING

通道正在与合作伙伴协商。

MQCHS_INACTIVE

通道未处于活动状态。

MQCHS_INITIALIZING

通道正在初始化。

MQCHS_PAUSED

通道已暂停。

MQCHS_REQUESTING

请求者通道正在请求连接。

MQCHS_RETRYING

通道正在重新尝试建立连接。

MQCHS_RUNNING

通道正在传输或等待消息。

MQCHS_STARTING

通道正在等待变为活动状态。

MQCHS_STOPPING

通道正在停止。

MQCHS_STOPPED

通道已停止。

ChannelDefOffset (MQLONG)-输入

用于将本地队列管理器链接到由 MQWDR 结构标识的队列管理器的通道的通道定义 (MQCD) 的逻辑偏移量。

- ChannelDefOffset 类似于 ClusterRecOffset
- 不能在指针算术中使用的逻辑偏移。要获取下一条记录的地址，必须使用 MQXCLWLN 调用。

DestSeqFactor (MQLONG)-输入

允许根据权重选择通道的目标序列因子。

- 在队列管理器对其进行更改之前，将使用 DestSeqFactor。
- 工作负载管理器会增加 DestSeqFactor，以确保根据消息的权重来分发消息。

DestSeqNumber (MQLONG)-输入

队列管理器更改集群通道目标值之前的集群通道目标值。

- 每次将消息放入该通道时，工作负载管理器都会增加 DestSeqNumber。
- 工作负载出口可以使用 DestSeqNumber 来决定将消息放入哪个通道。

MQWDR 的初始值和语言声明

MQWDR - 集群工作负载目标记录的初始值以及 C 和 High Level Assembler 语言声明。

字段名称	常量的名称	常量值
<i>StrucId</i>	MQWDR_STRUC_ID	'WDR? '
<i>Version</i>	MQWDR_VERSION_1	1
<i>StrucLength</i>	MQWDR_CURRENT_LENGTH ³	136
<i>QMgrFlags</i>	MQWDR_NONE	0
<i>QMgrIdentifier</i>	None	""
<i>QMgrName</i>	None	""
<i>ClusterRecOffset</i>	None	0
<i>ChannelState</i>	None	0
<i>ChannelDefOffset</i>	None	0
<i>DestSeqNumber</i>	None	0
<i>DestSeqFactor</i>	None	0

表 27: MQWDR 中字段的初始值 (继续)

字段名称	常量的名称	常量值
注意:		
1. 符号 ? 表示单个空白字符。		
2. 在 C 编程语言中, 宏变量 MQWDR_DEFAULT 包含缺省值。通过以下方式使用它来为结构中的字段提供初始值:		
<pre>MQWDR MyWDR = {MQWDR_DEFAULT};</pre>		
3. 初始值有意将结构的长度设置为当前版本的长度, 而不是结构的 1 版本。		

High Level Assembler

```
MQWDR          DSECT
MQWDR_STRUCID DS   CL4      Structure identifier
MQWDR_VERSION DS    F       Structure version number
MQWDR_STRUCLength DS   F       Length of MQWDR structure
MQWDR_QMGRFlags DS    F       Queue-manager flags
MQWDR_QMGRIDentifier DS  CL48   Queue-manager identifier
MQWDR_QMGRName DS   CL48   Queue-manager name
MQWDR_CLUSTERRECOFFSET DS    F   Offset of first cluster
* record
MQWDR_CHANNELSTATE DS    F     Channel state
MQWDR_CHANNELDEFOffset DS    F   Offset of channel definition
* structure
MQWDR_LENGTH EQU  *-MQWDR   Length of structure
ORG  MQWDR
MQWDR_AREA DS    CL(MQWDR_LENGTH)
```

C 声明

```
typedef struct tagMQWDR {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    StrucLength;      /* Length of MQWDR structure */
    MQLONG    QMgrFlags;        /* Queue-manager flags */
    MQCHAR48  QMgrIdentifier;    /* Queue-manager identifier */
    MQCHAR48  QMgrName;         /* Queue-manager name */
    MQLONG    ClusterRecOffset; /* Offset of first cluster record */
    MQLONG    ChannelState;     /* Channel state */
    MQLONG    ChannelDefOffset; /* Offset of channel definition structure */
    /* Ver:1 */
    MQLONG    DestSeqNumber;     /* Cluster channel destination sequence number */
    MQINT64   DestSeqFactor;     /* Cluster channel factor sequence number */
    /* Ver:2 */
};
```

MQWQR - 集群工作负载队列记录结构

下表汇总了 MQWQR - 集群工作负载队列记录结构中的字段。

表 28: MQWQR 中的字段		
字段	描述	页面
<i>StrucId</i>	结构标识	StrucId
<i>Version</i>	结构版本号	版本
<i>StrucLength</i>	MQWQR 结构的长度	StrucLength
<i>QFlags</i>	队列标志	QFlags
<i>QName</i>	队列名称	QName

表 28: MQWQR 中的字段 (继续)

字段	描述	页面
<i>QMgrIdentifier</i>	队列管理器标识	QMgrIdentifier
<i>ClusterRecOffset</i>	第一个集群记录 (MQWCR) 的偏移量	ClusterRec 偏移量
<i>QType</i>	队列类型	QTYPE
<i>QDesc</i>	队列描述	QDesc
<i>DefBind</i>	缺省绑定	DefBind
<i>DefPersistence</i>	缺省消息持久性	DefPersistence
<i>DefPriority</i>	缺省消息优先级	DefPriority
<i>InhibitPut</i>	是否允许对队列执行放置操作	InhibitPut
注: 如果版本小于 MQWQR_VERSION_2, 那么将忽略其余字段。		
<i>CWLQueuePriority</i>	表示队列优先级的值 0-9	CLWLQueuePriority
<i>CLWLQueueRank</i>	表示队列列组的值 0-9	CLWLQueueRank
注: 如果版本小于 MQWQR_VERSION_3, 那么将忽略其余字段。		
<i>DefPutResponse</i>	缺省放置响应	DefPut 响应

集群工作负载队列记录结构包含与消息的可能目标之一相关的信息。目标队列的每个实例都有一个集群工作负载队列记录结构。

集群工作负载队列记录结构在所有环境中都受支持。

此外, MQWQR1 和 MQWQR2 结构可用于向后兼容性。

相关参考

[MQ_CLUSTER_WORKLOAD_EXIT](#) -调用描述

队列管理器调用集群工作负载出口以将消息路由到可用队列管理器。

[MQXCLWLN](#) -浏览集群工作负载记录

[MQXCLWLN](#) 调用用于浏览存储在集群高速缓存中的 MQWDR, MQWQR 和 MQWCR 记录的链。

[MQWXP](#) -集群工作负载出口参数结构

下表汇总了 [MQWXP](#) -集群工作负载出口参数结构中的字段。

[MQWDR](#) -集群工作负载目标记录结构

下表汇总了 [MQWDR](#) -集群工作负载目标记录结构中的字段。

[MQWCR](#) -集群工作负载集群记录结构

下表汇总了 [MQWCR](#) 集群工作负载记录结构中的字段。

MQWQR 中的字段-集群工作负载队列记录结构

[MQWQR](#) -集群工作负载队列记录结构中字段的描述。

StrucId (MQCHAR4)-输入

集群工作负载队列记录结构的结构标识。

- StrucId 值为 MQWQR_STRUC_ID。
- 对于 C 编程语言, 还定义了常量 MQWQR_STRUC_ID_ARRAY。它具有与 MQWQR_STRUC_ID 相同的值。它是一个字符数组, 而不是字符串。

V (MQLONG)-输入

结构版本号。Version 采用下列其中一个值:

MQWQR_VERSION_1

Version-1 集群工作负载队列记录。

MQWQR_VERSION_2

Version-2 集群工作负载队列记录。

MQWQR_VERSION_3

Version-3 集群工作负载队列记录。

MQWQR_CURRENT_VERSION

集群工作负载队列记录的当前版本。

StrucLength (MQLONG)-输入

MQWQR 结构的长度。 StrucLength 采用下列其中一个值:

MQWQR_LENGTH_1

version-1 集群工作负载队列记录的长度。

MQWQR_LENGTH_2

version-2 集群工作负载队列记录的长度。

MQWQR_LENGTH_3

version-3 集群工作负载队列记录的长度。

MQWQR_CURRENT_LENGTH

集群工作负载队列记录的当前版本的长度。

QFlags (MQLONG)-输入

队列标志指示队列的属性。 定义了以下标志:

MQQF_LOCAL_Q

目标是本地队列。

MQQF_CLWL_USEQ_ANY

允许在 put 中使用本地和远程队列。

MQQF_CLWL_USEQ_LOCAL

仅允许本地队列放置。

其他值

此字段中的其他标志可能由队列管理器设置以用于内部目的。

QName (MQCHAR48)-输入

作为消息的可能目标之一的队列的名称。

- QName 的长度为 MQ_Q_NAME_LENGTH。

QMgrIdentifier (MQCHAR48)-输入

队列管理器标识是用于托管由 MQWQR 结构描述的队列实例的队列管理器的唯一标识。

- 此标识由队列管理器生成。
- QMgrIdentifier 的长度为 MQ_Q_MGR_IDENTIFIER_LENGTH。

ClusterRec 偏移量 (MQLONG)-输入

属于 MQWQR 结构的第一个 MQWCR 结构的逻辑偏移量。

- 对于静态高速缓存, ClusterRecOffset 是属于 MQWQR 结构的第一个 MQWCR 结构的偏移量。
- 偏移量以从 MQWQR 结构开始的字节为单位进行测量。
- 请勿将逻辑偏移用于具有动态高速缓存的指针算术。 要获取下一条记录的地址, 必须使用 MQXCLWLN 调用。

QType (MQLONG)-输入

目标队列的队列类型。 可能的值如下所示:

MQCQT_LOCAL_Q

本地队列。

MQCQT_ALIAS_Q

别名队列。

MQCQT_REMOTE_Q

远程队列。

MQCQT_Q_MGR_ALIAS

队列管理器别名。

QDesc (MQCHAR64)-输入

在主管由 MQWQR 结构描述的目标队列实例的队列管理器上定义的队列描述队列属性。

- QDesc 的长度为 MQ_Q_DESC_LENGTH。

DefBind (MQLONG)-输入

在托管 MQWQR 结构所描述的目标队列实例的队列管理器上定义的缺省绑定队列属性。在将组与集群配合使用时，必须指定 MQBND_BIND_ON_OPEN 或 MQBND_BIND_ON_GROUP。可能的值如下所示：

MQBND_BIND_ON_OPEN

由 MQOPEN 调用修复的绑定。

MQBND_BIND_NOT_FIXED

绑定未固定。

MQBND_BIND_ON_GROUP

允许应用程序请求将一组消息全部分配给同一目标实例。

DefPersistence (MQLONG)-输入

在托管 MQWQR 结构所描述的目标队列实例的队列管理器上定义的缺省消息持久性队列属性。可能的值如下所示：

MQPER_PERSISTENT

消息是持久消息。

MQPER_NOT_PERSISTENT

消息不是持久消息。

DefPriority (MQLONG)-输入

在托管 MQWQR 结构所描述的目标队列实例的队列管理器上定义的缺省消息优先级队列属性。优先级范围为 0- MaxPriority。

- 0 是最低优先级。
- MaxPriority 是主管此目标队列实例的队列管理器的队列管理器属性。

InhibitPut (MQLONG)-输入

在托管 MQWQR 结构所描述的目标队列实例的队列管理器上定义的禁止放入的队列属性。可能的值如下所示：

MQQA_PUT_INHIBITED

禁止执行放置操作。

MQQA_PUT_ALLOWED

允许执行放置操作。

CLWLQueuePriority (MQLONG)-输入

在托管 MQWQR 结构所描述的目标队列实例的队列管理器上定义的集群工作负载队列优先级属性。

CLWLQueueRank (MQLONG)-输入

在托管 MQWQR 结构所描述的目标队列实例的队列管理器上定义的集群工作负载队列列组。

DefPut 响应 (MQLONG)-输入

在托管 MQWQR 结构所描述的目标队列实例的队列管理器上定义的缺省 put 响应队列属性。可能的值如下所示：

MQPRT_SYNC_RESPONSE

对 MQPUT 或 MQPUT1 调用的同步响应。

MQPRT_ASYNC_RESPONSE

对 MQPUT 或 MQPUT1 调用的异步响应。

MQWQR 的初始值和语言声明

MQWQR -集群工作负载队列记录的初始值以及 C 和 High Level Assembler 语言声明。

表 29: MQWQR 中字段的初始值

字段名称	常量的名称	常量值
<i>StrucId</i>	MQWQR_STRUC_ID_ARRAY	'WQR? '
<i>Version</i>	MQWQR_VERSION_1	1
<i>StrucLength</i>	MQWQR_CURRENT_LENGTH ³	212
<i>QFlags</i>	None	0
<i>QName</i>	None	""
<i>QMgrIdentifier</i>	None	""
<i>ClusterRecOffset</i>	None	0
<i>QType</i>	None	0
<i>QDesc</i>	None	""
<i>DefBind</i>	None	0
<i>DefPersistence</i>	None	0
<i>DefPriority</i>	None	0
<i>InhibitPut</i>	None	0
<i>CLWLQueuePriority</i>	None	0
<i>CLWLQueueRank</i>	None	0
<i>DefPutResponse</i>	None	1

注意:

1. 符号 ? 表示单个空白字符。
2. 在 C 编程语言中, 宏变量 MQWQR_DEFAULT 包含缺省值。通过以下方式使用它来为结构中的字段提供初始值:

```
MQWQR MyWQR = {MQWQR_DEFAULT};
```

3. 初始值有意将结构的长度设置为当前版本的长度, 而不是结构的 1 版本。

C 声明

```
typedef struct tagMQWQR {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    StrucLength;      /* Length of MQWQR structure */
    MQLONG    QFlags;          /* Queue flags */
    MQCHAR48  QName;           /* Queue name */
    MQCHAR48  QMgrIdentifier;   /* Queue-manager identifier */
    MQLONG    ClusterRecOffset; /* Offset of first cluster record */
    MQLONG    QType;           /* Queue type */
    MQCHAR64  QDesc;           /* Queue description */
    MQLONG    DefBind;         /* Default binding */
    MQLONG    DefPersistence;   /* Default message persistence */
    MQLONG    DefPriority;      /* Default message priority */
    MQLONG    InhibitPut;      /* Whether put operations on the queue
                               are allowed */

    /* version 2 */
    MQLONG    CLWLQueuePriority; /* Queue priority */
    MQLONG    CLWLQueueRank;    /* Queue rank */
    /* version 3 */
    MQLONG    DefPutResponse;   /* Default put response */
};
```

High Level Assembler

```

MQWQR                                DSECT
MQWQR_STRUCID                        DS    CL4      Structure identifier
MQWQR_VERSION                        DS    F        Structure version number
MQWQR_STRUCLength                    DS    F        Length of MQWQR structure
MQWQR_QFLAGS                         DS    F        Queue flags
MQWQR_QNAME                          DS    CL48     Queue name
MQWQR_QMGRIDENTIFIER                 DS    CL48     Queue-manager identifier
MQWQR_CLUSTERRECOFFSET              DS    F        Offset of first cluster
*
MQWQR_QTYPE                          DS    F        Queue type
MQWQR_QDESC                          DS    CL64     Queue description
MQWQR_DEFBIND                        DS    F        Default binding
MQWQR_DEFPERSISTENCE                 DS    F        Default message persistence
MQWQR_DEFPRIORITY                    DS    F        Default message priority
MQWQR_INHIBITPUT                     DS    F        Whether put operations on
*
MQWQR_DEFPUTRESPONSE                 DS    F        the queue are allowed
MQWQR_LENGTH                         EQU    *-MQWQR  Default put response
*
MQWQR_AREA                           DS    CL(MQWQR_LENGTH)  Length of structure

```

MQWCR - 集群工作负载集群记录结构

下表汇总了 MQWCR 集群工作负载记录结构中的字段。

表 30: MQWCR 中的字段		
字段	描述	页面
<i>ClusterName</i>	集群的名称	ClusterName
<i>ClusterRecOffset</i>	下一个集群记录的偏移量 (MQWCR)	ClusterRec 偏移量
<i>ClusterFlags</i>	集群标志	ClusterFlags

集群工作负载集群记录结构包含有关集群的信息。对于目标队列所属的每个集群，都有一个集群工作负载集群记录结构。

集群工作负载集群记录结构在所有环境中都受支持。

相关参考

[MQ_CLUSTER_WORKLOAD_EXIT](#) - 调用描述

队列管理器调用集群工作负载出口以将消息路由到可用队列管理器。

[MQXCLWLN](#) - 浏览集群工作负载记录

[MQXCLWLN](#) 调用用于浏览存储在集群高速缓存中的 MQWDR, MQWQR 和 MQWCR 记录的链。

[MQWXP](#) - 集群工作负载出口参数结构

下表汇总了 MQWXP - 集群工作负载出口参数结构中的字段。

[MQWDR](#) - 集群工作负载目标记录结构

下表汇总了 MQWDR - 集群工作负载目标记录结构中的字段。

[MQWQR](#) - 集群工作负载队列记录结构

下表汇总了 MQWQR - 集群工作负载队列记录结构中的字段。

MQWCR - 集群工作负载集群记录结构中的字段。

MQWCR - 集群工作负载集群记录结构中字段的描述。

ClusterName (MQCHAR48)-输入

拥有 MQWCR 结构的目标队列实例所属的集群的名称。目标队列实例由 MQWDR 结构描述。

- ClusterName 的长度为 MQ_CLUSTER_NAME_LENGTH。

ClusterRec 偏移量 (MQLONG)-输入

下一个 MQWCR 结构的逻辑偏移量。

- 如果没有更多 MQWCR 结构，那么 ClusterRecOffset 为零。

- 偏移量以从 MQWCR 结构开始的字节为单位进行测量。

ClusterFlags (MQLONG)-输入

集群标志指示由 MQWCR 结构标识的队列管理器的属性。定义了以下标志:

MQQMF_REPOSITORY_Q_MGR

目标是完整的存储库队列管理器。

MQQMF_CLUSSDR_USER_DEFINED

已手动定义集群发送方通道。

MQQMF_CLUSSDR_AUTO_DEFINED

已自动定义集群发送方通道。

MQQMF_AVAILABLE

目标队列管理器可用于接收消息。

其他值

此字段中的其他标志可能由队列管理器设置以用于内部目的。

相关参考

MQWCR 的初始值和语言声明

MQWCR -集群工作负载集群记录结构的初始值以及 C 和 High Level Assembler 语言声明。

MQWCR 的初始值和语言声明

MQWCR -集群工作负载集群记录结构的初始值以及 C 和 High Level Assembler 语言声明。

表 31: MQWCR 中字段的初始值

字段名称	常量的名称	常量值
<i>ClusterName</i>	None	" "
<i>ClusterRecOffset</i>	None	0
<i>ClusterFlags</i>	None	0

C 声明

```
typedef struct tagMQWCR {
    MQCHAR48 ClusterName; /* Cluster name */
    MQLONG ClusterRecOffset; /* Offset of next cluster record */
    MQLONG ClusterFlags; /* Cluster flags */
};
```

High Level Assembler

```
MQWCR          DSECT
MQWCR_CLUSTERNAME DS CL48 Cluster name
MQWCR_CLUSTERRECOFFSET DS F Offset of next cluster
* record
MQWCR_CLUSTERFLAGS DS F Cluster flags
MQWCR_LENGTH EQU *-MQWCR Length of structure
MQWCR_AREA      DS CL(MQWCR_LENGTH)
```

相关参考

MQWCR -集群工作负载集群记录结构中的字段。

MQWCR -集群工作负载集群记录结构中字段的描述。

通道程序

此部分查看可用于通道的不同类型的通道程序 (MCA)。

下表显示了 MCA 的名称。

表 32: Windows UNIX and Linux 系统的通道程序

程序名	连接方向	通信
阿姆克姆帕		任何
运行 mqlsr	入站	任何
amqcrs6a	入站	LU 6.2
阿姆克斯塔	入站	TCP
鲁姆奇尔	出站	任何
鲁姆基	出站	任何

runmqlsr (运行 WebSphere MQ 侦听器), runmqchl (运行 WebSphere MQ 通道) 和 runmqchi (运行 WebSphere MQ 通道启动程序) 是在命令行中输入的控制命令。

使用 inetd 对 UNIX and Linux 系统上的 TCP 通道调用 amqcrsta, 其中不启动任何侦听器。

使用 LU6.2 时, 会将 amqcrs6a 作为事务程序调用

环境变量

供客户使用的服务器和客户机环境变量的列表。

使用范例

- 在 UNIX and Linux 系统上, 使用: `export [environment variable]=filename`。
- 在 Windows 系统上, 使用: `Set [environment variable]=filename`。
-

AMQ_MQS_INI_LOCATION

在 UNIX and Linux 系统上, 可以通过在此变量中设置 `mqs.ini` 文件的位置来变更用于 `mqs.ini` 文件的位置。必须在系统级别设置此变量。

AMQ_NO_IPV6

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

AMQ_SSL_ALLOW_DEFAULT_CERT

如果未设置 `AMQ_SSL_ALLOW_DEFAULT_CERT` 环境变量, 那么仅当证书包含 `ibmwebspheremq<userid>` 的标签名称时, 应用程序才能连接到客户机密钥库中具有个人证书的队列管理器。设置 `AMQ_SSL_ALLOW_DEFAULT_CERT` 环境变量时, 证书不需要标签名称 `ibmwebspheremq<userid>`。即, 用于连接到队列管理器的证书可以是缺省证书, 前提是密钥存储库中存在缺省证书, 并且密钥存储库不包含前缀为 `ibmwebspheremq<userid>` 的个人证书。有关更多信息, 请参阅技术说明 [在 MQ 客户机的 SSL 证书标签中指定用户标识](#)。

值 1 允许使用缺省证书。

V 7.5.0.9

AMQ_SSL_LDAP_SERVER_VERSION

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

在用于启动队列管理器或通道的环境中, 将该变量设置为相应的值。要请求使用 LDAP v2, 请设置 `AMQ_SSL_LDAP_SERVER_VERSION=2`。要请求使用 LDAP v3, 请设置 `AMQ_SSL_LDAP_SERVER_VERSION=3`。

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

GMQ_MQ_LIB

当系统上安装了 IBM WebSphere MQ MQI 客户机和 IBM WebSphere MQ 服务器时, 缺省情况下, MQAX 应用程序将针对服务器运行。要对客户机运行 MQAX, 必须在 `GMQ_MQ_LIB` 环境变量中指定客

户机绑定库，例如，设置 GMQ_MQ_LIB=mqic.dll。仅对于客户机安装，不需要设置 GMQ_MQ_LIB 环境变量。如果未设置此变量，那么 WebSphere MQ 将尝试装入 amqzst.dll。如果此 DLL 不存在(如仅在客户机安装中一样)，那么 WebSphere MQ 会尝试装入 mqic.dll。

主

此变量包含搜索 mqclient.ini 文件的目录的名称。此文件包含 UNIX and Linux 系统上的 IBM WebSphere MQ MQI 客户机使用的配置信息。

HOMEDRIVE 和 HOMEPATH

要同时使用这两个变量，必须设置。它们用于包含搜索 mqclient.ini 文件的目录的名称。此文件包含 Windows 系统上的 IBM WebSphere MQ MQI 客户机使用的配置信息。

LDAP 基本 DN

运行 LDAP 样本程序所需的环境变量。它指定目录搜索的基本专有名称。

LDAP_HOST

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

LDAP 版本

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

MQAPI_TRACE_LOGFILE

样本 API 出口程序使用 MQAPI_TRACE_LOGFILE 环境变量中定义的前缀对用户指定的文件生成 MQI 跟踪。

MQCCSID

指定要使用的编码字符集号并覆盖应用程序的本机 CCSID。

MQCERTVPOL

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。

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。

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。

MQC_IPC_HOST

共享 IBM WebSphere MQ 文件和生成的 myHostName 值时，将使用环境变量 MQC_IPC_HOST 创建问题集 myHostName

MQCLNTCF

使用此环境变量可修改 mqclient.ini 文件路径。

MQ_CHANNEL_SUPPRESS_INTERVAL

指定要禁止使用 MQ_CHANNEL_SUPPRESS_MSGS 定义的消息写入错误日志的时间间隔(以秒计)，以及在指定的时间间隔内禁止消息出现的次数。缺省值为 60,5，这意味着在该消息的前五次出现之后，将在 60 秒时间间隔内禁止给定消息的任何其他出现。有关更多信息，请参阅 [禁止来自错误日志的通道错误消息](#)。

环境变量 MQ_CHANNEL_SUPPRESS_INTERVAL 与 [qm.ini](#) 文件中的 [SuppressInterval](#) 相当。

MQ_CHANNEL_SUPPRESS_MSGS

指定在 MQ_CHANNEL_SUPPRESS_INTERVAL 中定义的时间间隔内直到该时间间隔到期之前，仅允许在指定次数内将 IBM WebSphere MQ 通道错误消息写入错误日志。有关更多信息，请参阅 [禁止来自错误日志的通道错误消息](#)。

环境变量 MQ_CHANNEL_SUPPRESS_MSGS 与 `qm.ini` 文件中的 `SuppressMessage` 相当，但其指定方式不同。

MQ_CONNECT_TYPE

在 IBM WebSphere MQ for Windows UNIX and Linux 系统上，将此环境变量与 MQCONNX 调用上使用的 MQCNO 结构的 "选项" 字段中指定的绑定类型结合使用。请参阅 [MQCONNX 环境变量](#)

MQ_FILE_PATH

在 Windows 平台上安装运行时包期间，将配置名为 MQ_FILE_PATH 的新环境变量。此环境变量包含与 Windows 注册表中的以下键相同的数据：

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

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"。

MQ_JAVA_DATA_PATH

指定日志和跟踪输出的目录。

MQ_JAVA_INSTALL_PATH

指定 IBM WebSphere MQ classes for Java 的安装目录，如 IBM WebSphere MQ classes for Java 安装目录中所示。

MQ_JAVA_LIB_PATH

指定存储 IBM WebSphere MQ classes for Java 库的目录。IBM WebSphere MQ classes for Java 提供的某些脚本（例如，IVTRun）使用此环境变量。

MQNAME

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

MQNOREMPOOL

设置此变量时，它将关闭通道池并使通道作为侦听器的线程运行。

MQPSE_TRACE_LOGFILE

发布出口样本程序时使用。在要跟踪的应用程序进程中，此环境变量描述必须将跟踪文件写入的位置。请参阅 ["发布出口" 样本程序](#)

MQSERVER

MQSERVER 环境变量用于定义最小通道。不能使用 MQSERVER 来定义 SSL 通道或具有通道出口的通道。MQSERVER 指定 WebSphere MQ 服务器的位置以及要使用的通信方法。

MQ_SET_NODELAYACK

设置此变量时，它会关闭 TCP 延迟应答

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

MQSNOAUT

MQSNOAUT 禁用对象权限管理器 (OAM) 并阻止任何安全性检查。MQSNOAUT 变量仅在创建队列管理器时生效。



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

MQSPREFIX

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`。

MQSSLCRYP

MQSSLCRYP 包含可用于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。

MQSSLFIPS

MQSSLFIPS 指定在 IBM 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。

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。

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。

MQSSLRESET

MQSSLRESET 表示在重新协商 SSL 密钥之前在 SSL 通道上发送和接收的未加密字节数。

MQS_TRACE_OPTIONS

使用环境变量 MQS_TRACE_OPTIONS 来分别激活高详细信息功能和参数跟踪功能。

MQTCPTIMEOUT

此变量指定 IBM WebSphere MQ 等待 TCP 连接调用的时间长度。

MQSUITEB

此变量指定是否使用符合 Suite B 的密码术。在使用 Suite B 密码术的实例中，可以通过将 MQSUITEB 设置为下列其中一项来指定密码术的强度：

- 无
- 128_BIT, 192_BIT
- 128_BIT
- 192_BIT

ODQ_MSG

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

ODQ_TRACE

如果使用不同于 RUNMQDLQ 的死信队列处理程序，那么样本的源可供您用作基础。该样本类似于产品中提供的死信处理程序，但跟踪和错误报告不同。将 ODQ_TRACE 环境变量设置为 YES 或 yes 以开启跟踪

OMQ_PATH

如果 ActiveX 脚本的 IBM WebSphere MQ 自动化类失败，那么您可以在此环境变量中找到 "首次故障症状" 报告。

OMQ_TRACE

MQAX 包含跟踪工具，用于帮助服务组织识别发生问题时发生的情况。它显示运行 MQAX 脚本时采用的路径。除非您迁到问题，否则请在设置跟踪的情况下运行，以避免不必要使用系统资源。OMQ_TRACE 是设置为控制跟踪的三个环境变量之一。为 OMQ_TRACE 指定任何值将打开跟踪工具。即使将 OMQ_TRACE 设置为 OFF，跟踪仍处于活动状态。请参阅 [使用跟踪](#)

OMQ_TRACE_PATH

为控制跟踪而设置的三个环境变量之一。请参阅 [使用跟踪](#)

OMQ_TRACE_LEVEL

为控制跟踪而设置的三个环境变量之一。请参阅 [使用跟踪](#)

ONCONFIG

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

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

在 Windows 系统上，使用：

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

WCF_TRACE_ON

有两种不同的跟踪方法可用于 WCF 定制通道，这两种跟踪方法独立或一起激活。每种方法都会生成各自的跟踪文件，所以当两种跟踪方法都激活时，会生成两个输出文件。可以使用四种组合来启用和禁用 2 种不同的跟踪方法。除了这些用于启用 WCF 跟踪的组合外，还可以使用 WCF_TRACE_ON 环境变量来启用 XMS .NET 跟踪。请参阅 [WCF 跟踪配置和跟踪文件名](#)

WMQSOAP_HOME

在 IBM WebSphere MQ 中正确安装和配置 .NET SOAP over JMS 服务托管环境之后，在执行其他配置步骤时使用。可从本地队列管理器进行访问。请参阅 [WCF 客户机到由 WebSphere MQ 样本托管的 .NET 服务](#)，以及 [WCF 客户机到由 WebSphere MQ 样本托管的 Axis Java 服务](#)

还在安装 WebSphere MQ Web Transport for SOAP 时使用。请参阅 [安装 WebSphere MQ Web Transport for SOAP](#)

分布式平台的消息通道规划示例

此部分提供了一个详细示例，说明如何将两个队列管理器连接在一起，以便可以在它们之间发送消息。

此示例说明了使应用程序能够使用队列管理器 QM1 将消息放在队列管理器 QM2 上的队列上所需的准备工作。在 QM2 上运行的应用程序可以检索这些消息，并将响应发送到 QM1 上的应答队列。

此示例说明如何使用 TCP/IP 连接。此示例假定将触发通道以在第一条消息到达他们正在服务的传输队列时启动。必须启动通道启动程序才能触发工作。

此示例使用 SYSTEM.CHANNEL.INITQ 作为启动队列。此队列已由 WebSphere MQ 定义。您可以使用其他启动队列，但必须自行定义该队列，并在启动通道启动程序时指定队列的名称。

示例显示的内容

此示例显示了您可以使用的 WebSphere MQ 命令 (MQSC)。

在所有示例中，MQSC 命令将显示为显示在命令文件中，并且将在命令行中输入这些命令。这两种方法看起来完全相同，但是，要在命令行上发出命令，必须首先为缺省队列管理器输入 `runmqsc`，或者输入 `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。

另一种方法是创建包含这些命令的文件。然后，命令中的任何错误都很容易更正。如果您调用了文件 `mqsc.in`，那么要在队列管理器 QMNAME 上运行该文件，请使用：

```
runmqsc QMNAME < mqsc.in > mqsc.out
```

您可以先验证文件中的命令，然后再使用以下命令运行该文件：

```
runmqsc -v QMNAME < mqsc.in > mqsc.out
```

对于可移植性，您应该将命令的行长度限制为 72 个字符。使用并置字符在有多行上继续。在 Windows 上，使用 Ctrl-z 在命令行结束输入。在 UNIX and Linux 系统上，使用 Ctrl-d。或者，使用 **end** 命令。

第 122 页的图 4 显示了示例方案。

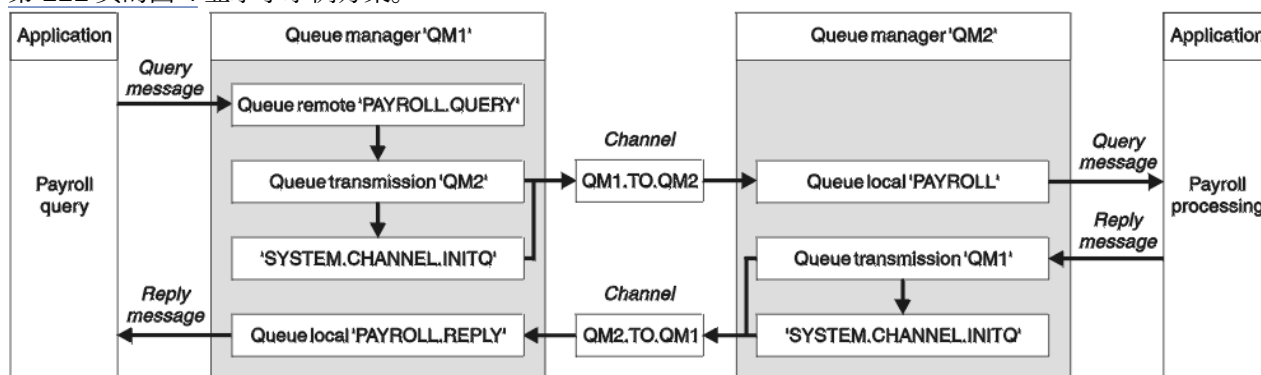


图 4: Windows UNIX and Linux 系统的消息通道示例

此示例涉及连接到队列管理器 QM1 的薪资查询应用程序，该应用程序将薪资查询消息发送到在队列管理器 QM2 上运行的薪资处理应用程序。工资单查询应用程序需要对其发送回 QM1 的查询的应答。薪资查询消息在名为 QM1.TO.QM2 的发送方/接收方通道上从 QM1 发送到 QM2，应答消息在另一个名为 QM2.TO.QM1

的发送方/接收方通道上从 QM2 发送回 QM1。一旦这两个通道有消息要发送到另一个队列管理器，就会触发这两个通道启动。

工资单查询应用程序将查询消息放入 QM1 上定义的远程队列“PAYROLL.QUERY”。此远程队列定义解析为 QM2 上的本地队列“PAYROLL”。此外，工资单查询应用程序指定将对查询的应答发送到 QM1 上的本地队列“PAYROLL.REPLY”。薪资处理应用程序从 QM2 上的本地队列“工资单”获取消息，并将回复发送到所需的任何位置；在这种情况下，本地队列“PAYROLL.REPLY”位于 QM1 上。

在 TCP/IP 的示例定义中，QM1 的主机地址为 192.0.2.0，并且正在侦听端口 1411，而 QM2 的主机地址为 192.0.2.1，并且正在侦听端口 1412。此示例假定这些内容已在您的系统上定义并且可供使用。

需要在 QM1 上创建的对象定义包括：

- 远程队列定义，PAYROLL.QUERY
- 传输队列定义，QM2 (缺省值为远程队列管理器名称)
- 发送方通道定义，QM1.TO.QM2
- 接收方通道定义，QM2.TO.QM1
- 应答队列定义，PAYROLL.REPLY

需要在 QM2 上创建的对象定义包括：

- 本地队列定义，PAYROLL
- 传输队列定义，QM1 (缺省值 = 远程队列管理器名称)
- 发送方通道定义，QM2.TO.QM1
- 接收方通道定义，QM1.TO.QM2

连接详细信息在发送方通道定义的 CONNAME 属性中提供。

您可以在 [第 122 页的图 4](#) 中看到排列图。

队列管理器 QM1 示例

以下对象定义允许连接到队列管理器 QM1 的应用程序将请求消息发送到 QM2 上名为 PAYROLL 的队列，并在 QM1 上名为 PAYROLL.REPLY 的队列上接收应答。

已向所有对象定义提供 DESCR 和 REPLACE 属性。提供的其他属性是使示例工作所需的最小值。未提供的属性采用队列管理器 QM1 的缺省值。

在队列管理器 QM1 上运行以下命令。

远程队列定义

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QM2') REPLACE +  
PUT(ENABLED) XMITQ(QM2) RNAME(PAYROLL) RQNAME(QM2)
```

注：远程队列定义不是物理队列，而是将消息定向到传输队列 QM2 的一种方法，以便可以将这些消息发送到队列管理器 QM2。

传输队列定义

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') REPLACE +  
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +  
INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(QM1.TO.QM2.PROCESS)
```

将第一条消息放入此传输队列时，会将触发器消息发送到启动队列 SYSTEM.CHANNEL.INITQ。通道启动程序从启动队列中获取消息，并启动在指定进程中标识的通道。

发送方通道定义

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) +  
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +  
CONNNAME('192.0.2.1(1412)')
```

接收方通道定义

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QM2')
```

应答队列定义

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Reply queue for replies to query messages sent to QM2')
```

应答队列定义为 PUT (ENABLED)。这将确保可以将应答消息放入队列。如果无法将应答放入应答队列，那么会将其发送至 QM1 上的死信队列，如果此队列不可用，请将其保留在队列管理器 QM2 上的传输队列 QM1 上。队列已定义为 GET (ENABLED)，以允许检索应答消息。

队列管理器 QM2 示例

以下对象定义允许连接到队列管理器 QM2 的应用程序从名为 PAYROLL 的本地队列检索请求消息，并将对这些请求消息的应答放入队列管理器 QM1 上名为 PAYROLL.REPLY 的队列。

您不需要提供远程队列定义以允许将应答返回到 QM1。从本地队列 PAYROLL 中检索的消息的消息描述符包含应答队列和应答队列管理器名称。因此，只要 QM2 可以将应答队列管理器名称解析为队列管理器 QM2 上传输队列的名称，就可以发送应答消息。在此示例中，应答队列管理器名称为 QM1，因此队列管理器 QM2 需要同名的传输队列。

所有对象定义都已随 DESCR 和 REPLACE 属性一起提供，并且是使示例起作用的最低要求。未提供的属性采用队列管理器 QM2 的缺省值。

在队列管理器 QM2 上运行以下命令。

本地队列定义

```
DEFINE QLOCAL(PAYROLL) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Local queue for QM1 payroll details')
```

由于与队列管理器 QM1 上的应答队列定义相同的原因，此队列定义为 PUT (ENABLED) 和 GET (ENABLED)。

传输队列定义

```
DEFINE QLOCAL(QM1) DESCR('Transmission queue to QM1') REPLACE +  
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +  
INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(QM2.TO.QM1.PROCESS)
```

将第一条消息放入此传输队列时，会将触发器消息发送到启动队列 SYSTEM.CHANNEL.INITQ。通道启动程序从启动队列中获取消息，并启动在指定进程中标识的通道。

发送方通道定义

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(TCP) +
```



```
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +  
CONNAME('192.0.2.0(1411)')
```

接收方通道定义

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QM1')
```

运行示例

有关启动通道启动程序和侦听器的信息以及有关在此场景中展开的建议。

一旦创建了这些定义，您就需要：

- 在每个队列管理器上启动通道启动程序。
- 启动每个队列管理器的侦听器。

有关启动通道启动程序和侦听器的信息，请参阅 [设置 Windows 的通信](#) 和 [在 UNIX and Linux 系统上设置通信](#)。

展开此示例

此简单示例可使用以下内容进行扩展：

- 使用 LU 6.2 通信与 CICS 系统互连以及事务处理。
- 添加更多队列，进程和通道定义，以允许其他应用程序在两个队列管理器之间发送消息。
- 在通道上添加用户出口程序以允许进行链接加密，安全检查或其他消息处理。
- 使用队列管理器别名和应答队列别名来了解有关如何在队列管理器网络的组织中使用这些别名的更多信息。

使用别名来引用 MQ 库

您可以定义别名以引用 JCL 中的 MQ 库，而不是直接使用 MQ 库的名称。然后，如果 MQ 库的名称发生更改，那么您只能删除并重新定义别名。

示例

以下示例定义别名 MQM.SCSQANLE 以引用 MQ 库 MQM.V600.SCSQANLE：

```
//STEP1 EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
DEFINE ALIAS (NAME(MQM.SCSQANLE) RELATE(MQM.V600.SCSQANLE))  
/*
```

然后，引用 MQM.V600.SCSQANLE 库使用别名 MQM.SCSQANLE。

注：库名和别名必须位于同一目录中，因此请对两者使用相同的高级限定符；在此示例中，高级限定符为 MQM。

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: