

9.1

IBM MQ 故障诊断和支持

IBM

注

在使用本资料及其支持的产品之前，请阅读第 401 页的『声明』中的信息。

本版本适用于 IBM® MQ V 9 发行版 1 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

| | |
|--|----------|
| 故障诊断与支持 | 5 |
| 进行初始检查..... | 6 |
| 在 UNIX, Linux, and Windows 上执行初步检查..... | 7 |
| 在 IBM i 上执行初步检查..... | 18 |
| 在 z/OS 上执行初步检查..... | 26 |
| 详细故障诊断..... | 38 |
| 对 AMS 问题进行故障诊断..... | 38 |
| 对命令问题进行故障诊断..... | 39 |
| 对分布式发布/预订问题进行故障诊断..... | 40 |
| 对分布式队列管理问题进行故障诊断..... | 43 |
| 对 IBM MQ Console 和 REST API 问题进行故障诊断..... | 51 |
| 对 IBM MQ Internet Pass-Thru 问题进行故障诊断..... | 53 |
| 对 IBM MQ MQI client 问题进行故障诊断..... | 55 |
| IBM MQ .NET 问题故障诊断..... | 56 |
| 对 Java 和 JMS 问题进行故障诊断..... | 56 |
| 对 Managed File Transfer 问题进行故障诊断..... | 77 |
| 对消息问题进行故障诊断..... | 124 |
| 对 MQ Telemetry 问题进行故障诊断..... | 124 |
| 多点广播问题故障诊断..... | 136 |
| 对队列管理器问题进行故障诊断..... | 138 |
| 对队列管理器集群问题进行故障诊断..... | 139 |
| 对 RDQM 配置问题进行故障诊断..... | 157 |
| 对安全性问题进行故障诊断..... | 166 |
| 对 IBM MQ 问题的 WCF 定制通道进行故障诊断..... | 176 |
| 故障诊断 XMS .NET 问题..... | 178 |
| 对 IBM MQ for z/OS 问题进行故障诊断..... | 179 |
| 联系 IBM 支持人员..... | 227 |
| 为 IBM 支持人员收集故障诊断信息..... | 227 |
| 使用错误日志..... | 285 |
| UNIX, Linux, and Windows 上的错误日志..... | 286 |
| IBM i 上的错误日志..... | 289 |
| z/OS 上的错误日志..... | 292 |
| 记录 IBM MQ classes for JMS 中的错误..... | 292 |
| 在 Multiplatforms 上禁止错误日志中的通道错误消息..... | 292 |
| First Failure Support Technology (FFST)..... | 293 |
| FFST: IBM MQ classes for JMS..... | 294 |
| FFST: IBM MQ for Windows..... | 298 |
| FFST: IBM MQ for UNIX 和 Linux 系统..... | 301 |
| FFST: IBM MQ for IBM i..... | 302 |
| WCF XMS First Failure Support Technology (FFST)..... | 303 |
| XMS .NET 应用程序的 FFDC 配置..... | 304 |
| 使用跟踪..... | 304 |
| 在 UNIX and Linux 系统上使用跟踪..... | 305 |
| IBM i 上的跟踪..... | 310 |
| 使用 Windows 上的跟踪..... | 315 |
| 在 z/OS 上使用跟踪进行问题确定..... | 318 |
| 跟踪高级消息排队协议 (AMQP) 服务..... | 332 |
| 跟踪 IBM MQ Bridge to Salesforce..... | 334 |
| 跟踪 IBM MQ Bridge to blockchain..... | 335 |
| 跟踪 IBM MQ Console 和 REST API..... | 335 |
| 跟踪 IBM MQ Internet Pass-Thru 中的错误..... | 337 |
| 跟踪 IBM MQ.NET 应用程序..... | 338 |

| | |
|---|------------|
| 跟踪 JMS 和 Java 应用程序..... | 338 |
| 在多平台上跟踪 Managed File Transfer 资源..... | 352 |
| 跟踪 Managed File Transfer for z/OS 资源..... | 358 |
| 跟踪 TLS: runmqakm 、 strmqikm 和 runmqckm 功能..... | 371 |
| 跟踪 IBM MQ 的 WCF 定制通道..... | 372 |
| 为 XMS .NET 应用程序配置跟踪..... | 373 |
| 启用 LDAP 客户机库代码的动态跟踪..... | 376 |
| 故障后恢复..... | 377 |
| 磁盘驱动器故障..... | 378 |
| 损坏的队列管理器对象..... | 379 |
| 损坏的单个对象..... | 379 |
| 自动介质恢复故障..... | 379 |
| z/OS 上的恢复过程示例..... | 379 |
| 声明..... | 401 |
| 编程接口信息..... | 402 |
| 商标..... | 402 |

IBM MQ 故障诊断和支持

如果您的队列管理器网络或 IBM MQ 应用程序存在问题，那么可以使用本信息中描述的方法来帮助您诊断和解决问题。如果需要有关问题的帮助，可以通过 IBM 支持站点联系 IBM 支持人员。

关于此任务

故障诊断是指找出发生问题的原因并加以解决的过程。如果您的 IBM 软件存在问题，那么只要您问自己“发生了什么？”，该问题的故障诊断过程就会立即开始。

高级别的基本故障诊断策略涉及：

1. [记录问题症状](#)
2. [重新创建问题](#)
3. [排除可能的原因](#)

如果您在 IBM MQ 方面遇到问题需要帮助，可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订，故障诊断和其他新闻的通知。有关更多信息，请参阅 [第 227 页的『联系 IBM 支持人员』](#)。

有关问题后恢复的更多信息，请参阅 [第 377 页的『故障后恢复』](#)。

过程

1. 记录问题的症状。

根据问题类型（应用程序问题、服务器问题还是工具问题），您可能会收到表明发生问题的消息。请始终记录看到的错误消息。尽管这听起来很简单，但错误消息有时包含代码，这些代码将在您进一步调查问题时更有意义。您还可能接收多条错误消息，它们看上去相似但有微妙的差别。通过记录每条错误消息的详细信息，就可以更多地了解存在的问题。错误消息的来源包括：

- 问题视图
- 本地错误日志
- Eclipse 日志
- 用户跟踪
- 服务跟踪
- 错误对话框

有关更多信息，请参阅以下主题：

- [第 285 页的『使用错误日志』](#)
- [第 293 页的『First Failure Support Technology \(FFST\)』](#)
- [第 304 页的『使用跟踪』](#)

如果 IBM MQ 组件或命令返回了错误，并且您需要有关写入屏幕或日志的消息的更多信息，请参阅 [消息和原因码](#)。

2. 重新创建问题。

回想一下，您执行的哪些步骤可能导致问题。请重试这些步骤，以查看是否可以轻松地重新创建问题。如果您具有一致可重复的测试用例，那么可以帮助确定需要哪些解决方案。

- 您第一次是如何注意到此问题的？
- 是否执行了其他操作才使您注意到该问题？
- 导致问题的过程是一个新过程，还是以前成功运行过？
- 如果此过程以前运行过，进行了什么更改吗？（更改指的是对系统进行的任何类型的更改，从添加新硬件或软件到重新配置现有软件。）
- 您看到的问题的第一个症状是什么？是否有其他症状在同一时间发生？

- 其他地方是否发生过相同的问题？是只有一台机器有问题还是多台机器有同样的问题？
- 正在生成哪些消息可能会指出问题所在？

有关这些问题类型的更多信息，请参阅第 6 页的『进行初始检查』和第 38 页的『详细故障诊断』。

3. 消除可能的原因。

通过排除不会导致问题的组件以缩小问题的范围。通过使用排除过程，您可以简化问题并避免在不相关方面浪费时间。请参阅本产品文档中的信息以及其他可用资源，以帮助完成消除过程。其他人遇到过此问题吗？是否有可下载的修订？有关更多信息，请参阅第 227 页的『联系 IBM 支持人员』。

相关参考

[故障诊断与支持参考](#)

进行初始检查




您可以进行一些初始检查，这些检查可能会提供您可能遇到的常见问题的答案。

关于此任务

使用子主题中提供的信息和常规建议，帮助针对您的平台执行初步检查并纠正问题。




过程

- 针对您的平台执行初步检查：

-  第 7 页的『在 UNIX, Linux, and Windows 上执行初步检查』
-  第 18 页的『在 IBM i 上执行初步检查』
-  第 26 页的『在 z/OS 上执行初步检查』




面向系统管理员的提示

- 检查错误日志，查找针对您的操作系统的消息：

-  第 286 页的『UNIX, Linux, and Windows 上的错误日志』
-  第 289 页的『IBM i 上的错误日志』
-  第 185 页的『在 IBM MQ for z/OS 上产生的诊断信息』

- 检查 `qm.ini` 的内容，以了解任何配置更改或错误。

有关更改配置信息的更多信息，请参阅：

-  在 UNIX, Linux, and Windows 上更改配置信息
-  在 IBM i 上更改配置信息
-  在 z/OS 上定制队列管理器

- 如果应用程序开发团队报告了意外的情况，可使用跟踪调查这些问题。

有关使用跟踪的信息，请参阅第 304 页的『使用跟踪』。

面向应用程序开发者的提示

- 检查应用程序中 MQI 调用的返回码。

有关原因码的列表，请参阅 [API 完成代码和原因码](#)。使用返回码中提供的信息来确定问题的原因。遵循原因码的程序员响应部分中的步骤以解决问题。

- 如果您不确定应用程序是否按预期运作（例如，不确定传递到 MQI 或传出 MQI 的参数），可使用跟踪来收集有关 MQI 调用的所有输入和输出的信息。

有关使用跟踪的更多信息，请参阅第 304 页的『使用跟踪』。有关处理 MQI 应用程序中的错误的更多信息，请参阅[处理程序错误](#)。

相关概念

第 285 页的『使用错误日志』

有各种错误日志可用于帮助进行问题确定和故障诊断。

相关任务

第 304 页的『使用跟踪』

您可以使用不同类型的跟踪来帮助您确定问题并进行故障诊断。

[故障诊断与支持参考](#)

ULW

在 UNIX, Linux, and Windows 上执行初步检查

开始在 UNIX, Linux, and Windows 上详细确定问题之前，请考虑是否存在引起问题的明显原因，或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

关于此任务

问题可能出在：

- IBM MQ
- 网络
- 应用程序
- 已配置可使用 IBM MQ 的其他应用程序

过程

- 考虑以下问题列表。

在核对该列表的过程中，请记住可能与问题相关的所有信息。即使您的调查没有直接给出原因，但如果您必须完成系统的问题确定练习，它们稍后可能对您是有帮助的。

- [第 8 页的『IBM MQ 以前是否成功运行过？』](#)
- [第 8 页的『自上次成功运行以来是否进行了任何更改？』](#)
- [第 8 页的『是否有任何错误消息或返回码来解释问题？』](#)
- [第 9 页的『是否可以重现问题？』](#)
- [第 9 页的『在 Windows 上创建或启动队列管理器时是否收到错误代码？』](#)
- [第 9 页的『问题是否仅影响远程队列？』](#)
- [第 9 页的『您得到了不正确的输出吗？』](#)
- [第 11 页的『您的一些队列失败了吗？』](#)
- [第 12 页的『您是否未能从 PCF 命令接收响应？』](#)
- [第 12 页的『应用程序以前运行成功吗？』](#)
- [第 13 页的『应用程序或系统运行缓慢吗？』](#)
- [第 14 页的『问题是否影响网络的特定部分？』](#)
- [第 14 页的『问题是否在一天中的特定时间发生？』](#)
- [第 14 页的『问题是否为间歇性？』](#)

相关任务

第 26 页的『在 z/OS 上执行初步检查』

开始在 z/OS 上详细确定问题之前，请考虑是否存在引起问题的明显原因，或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

第 18 页的『在 IBM i 上执行初步检查』

开始在 IBM i 上详细确定问题之前，请考虑是否存在引起问题的明显原因，或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

第 227 页的『联系 IBM 支持人员』

如果您在 IBM MQ 方面遇到问题需要帮助，可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订、故障诊断和其他新闻的通知。

[故障诊断与支持参考](#)

相关参考

[消息和原因码](#)

[PCF 原因码](#)

ULW IBM MQ 以前是否成功运行过？

如果 IBM MQ 之前未成功运行，那么很可能您尚未正确进行设置。请参阅[安装 IBM MQ](#)，并选择贵企业用于检查是否已正确安装产品的一个或多个平台。

要运行验证过程，请参阅针对企业使用的一个或多个平台的[验证 IBM MQ 安装](#)。

另请查看[配置](#)，以获取有关 IBM MQ 安装后配置的信息。

ULW 自上次成功运行以来是否进行了任何更改？

问题的原因可能是对 IBM MQ 配置和维护更新进行了更改，或对其他与 IBM MQ 进行交互的程序进行了更改。

当您考虑最近可能做出的更改时，请考虑 IBM MQ 系统，还要考虑与此系统连接的其他程序、硬件以及任何新应用程序。还要考虑可能已经在系统上运行的您不知道的新的应用程序的可能性。

- 您已经更改、添加或删除了任何队列定义吗？
- 您已经更改或添加了任何通道定义吗？可能对应用程序所需要的 IBM MQ 通道定义或任何底层通信定义进行了更改。
- 您的应用程序会处理那些因为您所做的更改而得到的返回码吗？
- 您是否已更改可能影响 IBM MQ 运行的任何操作系统组件？例如，是否已修改 Windows 注册表。

您是否应用了任何维护更新？

如果已向 IBM MQ 应用维护更新，请检查更新操作是否成功完成，并且未生成任何错误消息。

- 更新有任何特殊的说明吗？
- 运行了验证是否正确和完整的应用了该更新的测试吗？
- 如果 IBM MQ 还原到之前的维护级别，那么问题是否仍存在？
- 如果安装成功，请与 IBM 支持中心核对一下是否存在任何维护包错误。
- 如果已将维护包应用于任何其他程序，请考虑维护包可能对 IBM MQ 与其交互的方式产生的影响。

ULW 是否有任何错误消息或返回码来解释问题？

您可能会找到有助于您确定问题的位置和原因的错误消息或返回码。

IBM MQ 使用错误日志来捕获与其自身的操作有关的消息、启动的任何队列管理器以及来自使用中的通道的错误数据。检查错误日志查看是否已经记录了与问题相关的任何消息。

IBM MQ 还在 Windows 应用程序事件日志中记录了错误。在 Windows 上，检查 Windows 应用程序事件日志是否显示了任何 IBM MQ 错误。要打开日志，请从“计算机管理”面板中展开[事件查看器](#)，并选择[应用程序](#)。

ULW 有关错误日志的位置和内容的信息，请参阅第 286 页的『[UNIX, Linux, and Windows 上的错误日志](#)』

对于每个 IBM MQ 消息队列接口 (MQI) 和 IBM MQ 管理接口 (MQAI) 调用，队列管理器或出口例程都会返回完成代码和原因码，以指示调用是成功还是失败。如果应用程序得到返回码指示消息队列接口 (MQI) 调用失败，请检查原因码找出有关该问题的更多信息。

有关原因码的列表，请参阅 [API 完成代码和原因码](#)。

每个 MQI 调用的描述中均包含有关返回码的详细信息。

相关任务

[故障诊断与支持参考](#)

相关参考

[诊断消息：AMQ4000-9999](#)

[PCF 原因码](#)

[传输层安全性 \(TLS\) 返回码](#)

[WCF 定制通道异常](#)

z/OS [IBM MQ for z/OS 消息、完成代码和原因码](#)

ULW 是否可以重现问题？

如果您可以重现问题，那么考虑重现它的条件：

- 是由命令或等效的管理请求引起的吗？
如果由另一种方法输入，操作是否有效呢？如果在命令行输入时命令有效，但在其他情况下无效，应确保命令服务器未停止并且 SYSTEM.ADMIN.COMMAND.QUEUE 的队列定义未更改。
- 是由程序引起的吗？这是在所有 IBM MQ 系统以及所有队列管理器上失败？还是仅在某些系统或队列管理器上失败？
- 当问题发生时，您可以标识任何总好象在系统中运行的应用程序吗？如果可以，检查应用程序以查看它是否存在错误。

Windows 在 Windows 上创建或启动队列管理器时是否收到错误代码？

如果 IBM MQ Explorer 或 amqmdain 命令无法创建或启动队列管理器，并指出权限问题，那么可能的原因是，运行 IBM MQ Windows 服务的用户的权限不足。

确保配置为用于运行 IBM MQ Windows 服务的用户具有 IBM MQ Windows 服务所需的用户权限中所述的权限。缺省情况下，此服务配置为以 MUSR_MQADMIN 用户身份运行。对于后续安装，“准备 IBM MQ 向导”会创建一个名为 MUSR_MQADMINx 的用户帐户，其中 x 是下一个可用编号，表示不存在的用户标识。

ULW 问题是否仅影响远程队列？

问题仅影响远程队列时需检查的事项。

如果问题仅影响远程队列，请执行以下检查：

- 检查是否已经启动了必需的通道，并且可以触发该通道，以及运行必需的启动程序。
- 检查应该将消息放入远程队列的程序没有报告问题。
- 如果您使用触发来启动分布式排队进程，应检查传输队列是否已将触发设置为打开。还应检查触发器监视器是否正在运行。
- 检查错误日志，查找表明通道错误或问题的消息。
- 若有必要，手动启动通道。

ULW 您得到了不正确的输出吗？

在本部分中，不正确输出指应用程序：未收到您预期接收的消息；正接收包含意外或损坏信息的消息；正接收未预期接收的消息，例如，以另一应用程序为目标的消息。

未到达队列的消息

如果消息未能如预期到达，请检查以下事项：

- 消息成功放入队列了吗？

- 队列正确定义了吗？例如，MAXMSGL 足够大吗？
- 队列启用了放入吗？
- 队列已经满了吗？
- 另一个应用程序得到了对队列的互斥访问权了吗？
- 您能从队列取出任何消息吗？
 - 您需要获取同步点吗？

如果在同步点中放入或检索消息，在落实恢复单元前，它们不可用于其它任务。
 - 您的等待时间间隔足够长吗？

您可以将等待时间间隔设置为 MQGET 调用的一个选项。确保您等了足够长的时间以获得响应。
 - 您在等由消息或相关标识标识的特定消息吗 (*MsgId* 或 *CorrelId*) ？

检查您是否在用正确的 *MsgId* 或 *CorrelId* 等待消息。成功的 MQGET 调用将这两个值都设置为检索的消息的值，因此，您可能需要复位这些值才能成功地取出另一个消息。

同样，检查是否可以从队列取出其他消息。
 - 其他应用程序可以从队列取出消息吗？
 - 您预期的消息是定义为持久的吗？

如不是，并且重新启动了 IBM MQ，那么会丢失此消息。
 - 另一个应用程序得到了对队列的互斥访问权了吗？

如果您找不到队列有什么错误，并且 IBM MQ 正在运行，请针对以下内容检查您预期将消息放入队列的过程：

- 应用程序启动了吗？

如果应该已经触发了该应用程序，请检查是否指定了正确的触发器选项。
- 应用程序停止了吗？
- 是否正在运行触发器监视器？
- 正确定义了触发器进程吗？
- 应用程序正确完成了吗？

查找作业日志中的异常结束证据。
- 应用程序落实其更改了吗？或将它们回退了吗？

如果有多个事务在为此队列提供服务，它们可能会相互冲突。例如，假设有一个事务发出缓冲区长度为零的 MQGET 调用，以查找消息的长度，然后发出指定了那个消息的 *MsgId* 的特定 MQGET 调用。然而，与此同时，另一个事务为该消息发出了成功的 MQGET 调用，因此第一个应用程序收到原因码 MQRC_NO_MSG_AVAILABLE。必须将可能要在多个服务器环境中运行的应用程序设计为能处理上述情况。

考虑可能接收到的消息，但出于某种原因您的应用程序未能处理它。例如，是由于所要的消息格式出错而导致程序拒绝它吗？如果是，请参阅本主题中的后续信息。

包含意外信息或损坏信息的消息

如果在消息中包含的消息不是应用程序所预期的，或已经以某种方式损坏，那么考虑以下各项：

- 您的应用程序或将消息放在队列上的应用程序，被更改了吗？

确保在所有需要注意更改的系统上都同步反映了全部更改。

例如，消息数据的格式可能已经被更改，无论哪种情况，都必须重新编译应用程序来使更改生效。如果一个应用程序还没有被重新编译，那么将对其余应用程序来说显示为数据损坏。
- 应用程序对错误队列发送了消息吗？

检查确保应用程序正在接收的消息并非应该被发送到服务于另一个队列的应用程序。若有必要，请将安全性定义更改为阻止未授权的应用程序将消息放入错误的队列。

如果应用程序使用别名队列，那么检查别名是否指向正确的队列。

- 已经为该队列指定了正确的触发器信息吗？

检查应该已经启动了应用程序；或应该启动了不同的应用程序吗？

如果这些检查没有使您解决问题，那么为发送消息的程序和接收消息的程序检查应用程序逻辑。

在使用分布式队列时不正确的输出问题

如果您的应用程序使用分布式队列，考虑以下要点：

- IBM MQ 已被正确安装在发送和接收系统上了吗？是否已针对分布式排队进行正确配置？
- 在两个系统之间的链接是可用的吗？

检查两个系统是否都可用并连接到 IBM MQ。检查两个系统之间的连接是活动的。

您可以对队列管理器或通道使用 MQSC 命令 PING（即，PING QMGR 或 PING CHANNEL）来验证该链接是可操作的。

- 在发送系统中设置了触发吗？
- 您所等待的消息是从远程系统来的应答消息吗？

检查远程系统中的触发已被激活。

- 队列已经满了吗？

如果是，检查消息是否已经被放入死信队列上。

死信队列头包含原因或反馈代码，解释为什么消息无法被放入目标队列中。有关死信队列头结构的信息，请参阅[使用死信（未送达消息）队列和 MQDLH - 死信头](#)。

- 在发送和接收队列管理器之间有不匹配吗？

例如，消息长度可能超过了接收队列管理器可处理范围的长度。

- 发送和接收通道的通道定义是兼容的吗？

例如，序列号包中的不匹配可能停止分布式排队组件。有关分布式排队的更多信息，请参阅[分布式排队和集群](#)。

- 涉及数据转换吗？如果发送和接收应用程序之间的数据格式不同，那么数据转换是必需的。如果格式被识别为内置格式之一，那么当发出 MQGET 调用时会发生自动转换。

如果数据格式不被转换所识别，那么采用数据转换出口来允许您用自己的例程执行转换。

请参阅[数据转换](#)，以获取有关数据转换的更多信息。

您的一些队列失败了吗？

如果您怀疑问题仅在队列的子集发生，请检查您认为有问题的本地队列。

执行下列检查：

1. 显示有关每个队列的信息。您可以使用 MQSC 命令 DISPLAY QUEUE 来显示信息。
2. 使用所显示的数据来执行以下检查：
 - 如果 CURDEPTH 是在 MAXDEPTH 的，那么没有处理队列。检查所有应用程序都是正常运行的。
 - 如果 CURDEPTH 不是 MAXDEPTH，请检查以下队列属性来确保其正确：
 - 如果正在使用触发：
 - 是否正在运行触发器监视器？
 - 触发器深度太深吗？即，它通常生成足够的触发器事件吗？
 - 进程名正确吗？
 - 进程是可用的和可操作的吗？
 - 队列可以共享吗？如果不可以，那么另一个应用程序可能已经为输入打开了该队列。

- 队列相应地启用了 GET 和 PUT 吗?
- 如果没有应用程序进程从队列处理取出消息，那么确定为什么会这样。可能是因为需要启动应用程序、连接已经中断或由于某种原因 MQOPEN 调用已经失败。

检查队列属性 IPPROCS 和 OPROCS。这些属性表明是否已经为输入和输出打开了队列。如果值是零，那么表明不会发生该类型的操作。可能已经更改了该值；则可能已经打开的队列，但现在被关闭了。

您需要在放入或取出消息时检查状态。

如果无法解决问题，请联系 IBM 支持中心以获取帮助。

您是否未能从 PCF 命令接收响应?

当您发出了命令但没有接收到响应时的注意事项。

如果您发出了命令但没有接收到响应，请考虑以下检查：

- 命令服务器在运行吗?

运用 dspmqcsv 命令来检查命令服务器的状态。

- 如果该命令的响应表明命令服务器没有运行，那么使用 stmqcsv 命令将其启动。
- 如果该命令的响应表明 SYSTEM.ADMIN.COMMAND.QUEUE 不是为 MQGET 请求启用的，那么启用 MQGET 请求的队列。

- 已将应答发送到死信队列了吗?

死信队列头结构包含原因或反馈代码，对该问题进行了描述。有关死信队列头结构 (MQDLH) 的信息，请参阅 [MQDLH - 死信头和使用死信（未送达消息）队列](#)。

如果死信队列包含消息，那么您可以使用所提供的浏览样本应用程序 (amqsbcg) 来使用 MQGET 调用浏览消息。样本应用程序为已命名队列管理器单步遍历已命名队列上的所有消息，为已命名队列上的所有消息显示消息描述符和消息上下文字段。

- 消息被发送到错误日志了吗?

请参阅第 288 页的『UNIX, Linux, and Windows 上的错误日志目录』，以获取进一步的信息。

- 队列启用了放入和取出操作了吗?

- *WaitInterval* 够长了吗?

如果 MQGET 调用超时，将返回完成代码 MQCC_FAILED 和原因码 MQRC_NO_MSG_AVAILABLE。（请参阅 [WaitInterval \(MQLONG\)](#)，以获取有关 *WaitInterval* 字段以及 MQGET 的完成代码和原因码的信息。）

- 如果您使用自己的应用程序将命令放到 SYSTEM.ADMIN.COMMAND.QUEUE 上，您还需要获取同步点吗?

除非已从同步点排除请求消息，否则需要在接收回复消息前获取同步点。

- 队列的 MAXDEPTH 和 MAXMSGL 属性设置的是否足够高?

- 您是否正确使用了 *CorrelId* 和 *MsgId* 字段?

在应用程序中设置 *MsgId* 和 *CorrelId* 的值，以确保接收来自队列的所有消息。

尝试停止命令服务器，然后重新启动，响应所产生的任何错误消息。

如果系统仍未响应，那么可能是队列管理器或整个 IBM MQ 系统出现问题。首先，尝试停止个别的队列管理器来隔离失败的队列管理器。如果此步骤未显示问题，请尝试停止并重新启动 IBM MQ，以响应错误日志中生成的任何消息。

如果问题在重新启动后仍然存在，请联系 IBM 支持中心以获取帮助。

应用程序以前运行成功吗?

使用本主题中的信息来帮助诊断应用程序的常见问题。

如果问题看来好象涉及特定的应用程序，请考虑以前应用程序是否成功运行。

在对此问题回答是之前，考虑以下各项：

- 自从上次成功运行以来，对应用程序做了什么更改吗？

如果做了，错误可能位于应用程序新的或修改过的部分的某一处。检查该更改并看以下您是否可以找到明显的问题原因。有可能重试使用以前级别的应用程序吗？

- 应用程序的所有功能以前都运用过吗？

问题可能是在第一次使用以前从未调用的应用程序的一部分时发生的吗？如果是，那么可能错误是位于应用程序的这一部分的。尝试找出应用程序失败时在做什么，并检查该程序的这一部分的源代码以找出错误。

如果程序已经在以前许多的场合中成功运行，那么检查当前队列状态和错误发生时正在处理的文件。有可能它们包含一些不常见的调用程序中难得使用的路径的数据值。

- 应用程序检查所有返回码吗？

IBM MQ 系统是否略有更改，使得应用程序未检查其接收为更改结果的返回码？例如，您的应用程序是否假设其访问的队列是可以共享的吗？如果队列已被重新定义为独占的，您的应用程序能否处理表明其不再可访问该队列的返回码吗？

- 应用程序是否在其他 IBM MQ 系统上运行？

是否可能是设置此 IBM MQ 系统导致发生问题的方式有所不同？例如，已经用相同的消息长度或优先级定义了队列吗？

在您查看代码之前，根据代码是用哪种编程语言编写的，通过检查转换程序、编译器和链接编辑器的输出来查看是否报告了任何错误。

如果您的应用程序转换、编译或链接编辑到装入库失败，则如果您尝试调用它也会运行失败。请参阅[开发应用程序](#)，以获取有关构建应用程序的信息。

如果文档显示这些步骤中的每一步都是正确完成的，则考虑应用程序的编码逻辑。问题的症状是否指示此功能失败，并因此指示此代码段出错？请参阅以下部分，了解使 IBM MQ 应用程序发生问题的一些常见错误示例。

常见编程错误

以下列表中的错误说明在运行 IBM MQ 程序时遇到的问题的最常见原因。请考虑 IBM MQ 系统问题可能由于以下一项或多项错误所致可能性：

- 假定可以共享队列，而实际上这些队列是独占的。
- 在 MQI 调用中传递了不正确的参数。
- 在 MQI 调用中传递的参数不足。这可能意味着 IBM MQ 无法设置要供应用程序处理的完成码和原因码。
- 无法检查来自 MQI 请求的返回码。
- 传递指定了不正确的长度的变量。
- 以错误的顺序传递参数。
- 无法正确地初始化 *MsgId* 和 *CorrelId*。
- MQRC_TRUNCATED_MSG_ACCEPTED 后的 *Encoding* 和 *CodedCharSetId* 初始化失败。

应用程序或系统运行缓慢吗？

如果您的应用程序运行缓慢，那么可能表明它在循环中，或在等待不可用的资源，或可能存在性能问题。

可能您的系统操作已接近容量极限了。这类问题可能在系统负载的峰值时间最严重，通常在上午的中间时段和下午的中间时段。（如果网络跨越了多个时区，那么系统负载峰值可能会看似发生在其他时间。）

性能问题可能由硬件限制引起。

如果您发现性能降低与系统负载无关，而是在系统负载较轻时发生的，那么可能要归咎于设计不良的应用程序。只有在访问某些队列时才会出现这种问题。

如果性能问题仍然存在，那么问题可能在于 IBM MQ 自身。如果您怀疑这一点，请联系 IBM 支持中心以获取帮助。

应用程序性能变慢或在队列（通常是传输队列）上消息积累的常见原因是，一个或多个应用程序在工作单元外有写入持久消息；有关更多信息，请参阅[消息持久性](#)。

ULW 问题是否影响网络的特定部分？

您可能可以标识受问题影响的网络的特定部分（例如，远程队列）。如果到远程消息队列管理器的链接不起作用，那么消息无法流向远程队列。

检查两个系统之间的连接是否正常，以及是否启动了 IBM MQ 的内部通信组件。

检查该消息正到达传输队列，并检查传输队列和任何远程队列的本地队列定义。

是否已做出任何与网络相关的更改？或是否更改可能会解释此问题的任何 IBM MQ 定义？

ULW 问题是否在一天中的特定时间发生？

如果问题发生在一天中的特定时间，那么可能与系统负载有关。通常，系统负载峰值是在上午的中间时段和下午的中间时段发生的，因此在这此时段很可能发生与负载相关的问题。（如果 IBM MQ 网络跨越了多个时区，那么系统负载峰值可能会看似发生在一天中的其他时间。）

ULW 问题是否为间歇性？

间歇性的问题可能由可以互相独立运行的进程引起。例如，完成较早的进程前，程序可能没有指定等待选项就发出 MQGET 调用。如果应用程序在落实放入消息的调用前尝试从队列取出消息，那么也可能出现间歇性的问题。

Linux UNIX 资源问题

下面介绍了如何确定和解决与 IBM MQ 资源（包括 IBM MQ 进程的资源使用情况）相关的问题，还介绍了如何确定和解决与资源不足以及资源限制配置相关的问题。

用于调查资源问题的有用命令和配置文件

用于显示系统上的当前值或对系统进行临时更改的有用命令：

ulimit -a

显示用户限制

ulimit -Ha

显示用户硬限制

ulimit -Sa

显示用户软限制

ulimit -<paramflag> <value>

其中，**paramflag** 是资源名称的标记，例如，**s** 表示堆栈。

要对系统上的资源限制进行永久更改，请使用 `/etc/security/limits.conf` 或 `/etc/security/limits`。

Linux 您可以从 Linux 上的 `proc` 文件系统获取当前为进程设置的资源限制。例如，`cat /proc/<pid of MQ process>/limits`。

在调整 IBM MQ 或内核参数前执行的基本检查

您需要调查以下项：

- 活动连接数是否在期望的限制范围内。

例如，假设在用户进程数不超过 3000 时，您的系统调整为允许 2000 个连接。如果连接数增加至超过 2000 个，那么用户进程数已增加至超过 3000（因为添加了新的应用程序），或者出现了连接泄漏。

要检查这些问题，请使用以下命令：

– **UNIX** IBM MQ 进程数：

```
ps -elf|egrep "amq|run"|wc -l
```

– **Linux** IBM MQ 进程数：

```
ps -eLf|egrep "amq|run"|wc -l
```

– **Linux** **UNIX** 连接数：

```
echo "dis conn(*) all" | runmqsc <qmgr name>|grep EXTCONN|wc -l
```

– **Linux** **UNIX** 共享内存使用量：

```
ipcs -ma
```

– **Solaris** 共享内存使用量以及项目详细信息：

```
ipcs -mJ
```

- 如果连接数超出期望的限制，请检查连接源。
- 如果共享内存使用量非常高，请检查以下项的数量：
 - 主题
 - 打开的队列句柄
- 必须在 IBM MQ 透视图中检查并调整以下资源：
 - **Linux** 指定的用户进程数所允许的最大线程数。
 - 数据段
 - 堆栈段
 - 文件大小
 - 打开的文件句柄
 - 共享内存限制
 - 线程限制，例如，`threads-max` on Linux
- 可使用 `mqconfig` 命令来检查当前资源使用情况。

注意：

1. 上文中列出的某些资源需要在用户级别进行调整，而某些资源则需要在操作系统级别进行调整。
2. 虽然上面的列表不是完整列表，但足以应对 IBM MQ 报告的最常见的资源问题。
3. **Linux** 由于每个线程都是一个轻量级进程 (LWP)，因此需要在线程级别进行调整。

在 IBM MQ 或应用程序中创建线程或进程时出现的问题

`xcsExecProgram` 和 `xcsCreateThread` 中的故障

探测器标识、错误消息和组成部分

XY348010 来自 `xtmStartTimerThread` 来自 IBM MQ 进程 (例如 `amqz1aa0`) 或应用程序

来自 `amqzxma0` 中 `xcsExecProgram` (含有错误代码 `xecP_E_PROC_LIMIT`) 的 XC037008

XC035040 `xcsCreateThread`

xcsExecProgram 返回 XC037007, 并附带 **xecP_E_NO_RESOURCE**

xcsCreateThread 会失败, 并返回 **xecP_E_NO_RESOURCE**, 后跟故障数据捕获, 例如, 来自 **zlaMain** 的 ZL000066

探测器标识可能会有所不同。请检查错误代码 **xecP_E_PROC_LIMIT** 和 **xecP_E_NO_RESOURCE**。来自 **pthread_create** 的报告 **errno 11** 的错误消息, 例如: 来自 **pthread_create** 的 AMQ6119S: An internal IBM MQ error has occurred ('11 - Resource temporarily unavailable')。

Linux AIX 解决 AIX 和 Linux 上的问题

当 **pthread_create** 或派生再次失败时, IBM MQ 会设置错误代码 **xecP_E_PROC_LIMIT**。

EAGAIN

使用 **ulimit** 命令, 检查并增加每个用户资源限制和堆栈资源限制的进程数。

Linux 其他必要配置

复查并增大 **kernel.pid_max** (**/proc/sys/kernel/kernel.pid_max** 和 **kernel.threads-max** (**/proc/sys/kernel/threads-max**) 内核参数的限制。

您需要增加 **mqm** 用户和用于启动队列管理器和 IBM MQ 应用程序的任何其他任何用户的最大用户进程数 (**nproc**) 和堆栈大小资源限制。

ENOMEM

当 **pthread_create** 或 **fork** 发生 **ENOMEM** 故障时, IBM MQ 会设置错误代码 **xecP_E_NO_resource**。

检查并提高堆栈大小和数据资源限制。

注意:

- 您可以使用 **ulimit** 命令或者通过更改资源限制配置文件来增加用户进程资源限制。
- 使用 **ulimit** 命令所做的更改都是临时更改。修改 **/etc/security/limits** 或 **/etc/security/limits.conf**, 使更改成为永久性更改。操作系统上的实际配置可能有所不同, 因此必须检查该配置。
- 您还应该查看操作系统手册 (例如, **pthread_create** 的联机帮助页) 以获取有关资源问题和调整资源限制的更多详细信息, 并确保对资源限制进行相应配置。
- 您还应该检查系统的资源 (包括内存和 CPU) 是否不足。

Solaris ENOMEM 和 EAGAIN 错误所需的其他配置

使用 **projadd** 或 **projmod** 命令, 检查并提高项目的堆栈 (**process.max-stack-size**) 和数据资源限制。

创建共享内存时出现的问题

错误: **shmget** 失败, 错误编号为 **28(ENOSPC)**

```
| Probe Id          :- XY132002 |
| Component        :- xstCreateExtent |
| ProjectID       :- 0 |
| Probe Description :- AMQ6119: An internal IBM MQ error has occurred |
| (Failed to get memory segment: shmget(0x00000000, 2547712) [rc=-1 |
| errno=28] No space left on device) |
| FDCSequenceNumber :- 0 |
| Arith1          :- 18446744073709551615 (0xffffffffffffffff) |
| Arith2          :- 28 (0x1c) |
| Comment1        :- Failed to get memory segment: shmget(0x00000000, |
| 2547712) [rc=-1 errno=28] No space left on device |
| Comment2        :- No space left on device |
+-----+
MQM Function Stack
ExecCtrlrMain?
xcsAllocateMemBlock
xstExtendSet
xstCreateExtent
xcsFFST
```


shmget 失败, 错误编号为 22(EINVAL)

```
| Operating System  :- SunOS 5.10
| Probe Id         :- XY132002
| Application Name  :- MQM
| Component        :- xstCreateExtent
| Program Name     :- amqzma0
| Major Errorcode  :- xecP_E_NO_RESOURCE
| Probe Description :- AMQ6024: Insufficient resources are available to
|                  complete a system request.
| FDCSequenceNumber :- 0
| Arith1           :- 18446744073709551615 (0xffffffffffffffff)
| Arith2           :- 22 (0x16)
| Comment1        :- Failed to get memory segment: shmget(0x00000000,
|                  9904128) [rc=-1 errno=22] Invalid argument
| Comment2        :- Invalid argument
| Comment3        :- Configure kernel (for example, shmmax) to allow a
|                  shared memory segment of at least 9904128
bytes
+-----+
MQM Function Stack
ExecCtrlrMain
zxcCreateECResources
zutCreateConfig
xcsInitialize
xcsCreateSharedSubpool
xcsCreateSharedMemSet
xstCreateExtent
xcsFFST
```

在 Solaris 上解决问题

您应该:

- 增大 IBM MQ 所使用的项目的共享内存资源限制 (project.max-shm-memory)。
- 使用以下命令, 查找与 IBM MQ 进程和应用程序相关联的项目标识:
 - **ps** 命令:

```
ps -eo user,pid,uid,projid,args|egrep "mq|PROJID"
```

和 **projects -l** 命令, 或

- 故障数据捕获 (FDC) 头中的 **Project Id** 属性以及 **projects -l** 命令, 或
- **ipcs -J** 和 **projects -l** 命令

意外的进程终止和队列管理器崩溃, 或者队列管理器崩溃

进程意外结束, 后跟来自 amqzma0 的 FDC

示例 FDC:

```
Date/Time      :- Mon May 02 2016 01:00:58 CEST
Host Name     :- test.ibm.com
LVLS          :- 8.0.0.4
Product Long Name :- IBM MQ for Linux (x86-64 platform)
Probe Id      :- XC723010
Component     :- xprChildTermHandler
Build Date    :- Oct 17 2015
Build Level   :- p800-004-151017
Program Name  :- amqzma0
Addressing mode :- 64-bit
Major Errorcode :- xecP_E_USER_TERM
Minor Errorcode :- OK
Probe Description :- AMQ6125: An internal IBM MQ error has occurred.
```

可能的原因和解决方案

- 检查用户是否已终止任何进程。
- 检查 IBM MQ 进程是否由于内存异常而终止:

- 通过 FDC Component :- xehExceptionHandler 终止了进程?
- 应用这方面已更正的已知问题的修订。
- 检查操作系统是否因为进程的内存使用量较高而终止了进程:
 - IBM MQ 进程是否消耗了大量内存?
 - 操作系统是否终止了进程?

请查看操作系统日志。例如, Linux 上的 OOM-killer:

```
Jan 2 01:00:57 ibmtest kernel:
amqmpmpa invoked oom-killer: gfp_mask=0x201da, order=0, oom_score_adj=0)
```

- 对已知的内存泄漏问题应用修订。

进程使用的用户限制与已配置的限制的差别

进程使用的用户限制可能与已配置的限制不同。如果进程由不同用户或用户脚本(例如,高可用性脚本)启动,那么可能发生此情况。请务必检查正在启动队列管理器的用户并为此用户设置相应的资源限制。

IBM i 在 IBM i 上执行初步检查

开始在 IBM i 上详细确定问题之前,请考虑是否存在引起问题的明显原因,或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

关于此任务

问题的原因可能在以下几方面:

- 硬件
- 操作系统
- 相关软件,例如语言编译器
- 网络
- IBM MQ 产品
- IBM MQ 应用程序
- 其他应用程序
- 站点操作过程

以下过程列出了一些需要考虑的初步问题。如果执行这些初步检查能够找到问题的原因,那么您可以在需要时使用 IBM MQ 产品文档的其他部分以及其他许可程序库中的信息来帮助解决问题。

如果执行这些初步检查不能确定问题的原因,并因此需要进行更详细的调查,那么子主题中提供了一些需要考虑的其他问题。在浏览问题列表的过程中,请记住可能与问题相关的任何内容。即使您的调查没有直接给出原因,但如果您必须完成系统的问题确定练习,它们稍后可能对您是有帮助的。

过程

- 考虑以下问题。
 - 以下步骤旨在帮助您隔离问题,均从 IBM MQ 应用程序的角度考虑。检查每个阶段的所有建议。
 1. IBM MQ for IBM i 以前是否成功运行过?
 - Yes**
继续执行步骤 [第 18 页的『2』](#)。
 - 否**
很可能您未正确安装或设置 IBM MQ。
 2. IBM MQ 应用程序以前运行成功吗?

Yes

继续执行步骤 [第 19 页的『3』](#)。

否

请考虑以下事项：

- a. 应用程序可能编译或链接失败，在您尝试调用时可能失败。检查来自编译器或链接程序的输出。

请参阅相应的编程语言参考信息，或参阅[开发应用程序](#)，以获取有关如何构建应用程序的信息。

- b. 考虑应用程序的逻辑。例如，问题的症状是否表明函数故障，并因此表明代码出错。

检查以下常见编程错误：

- 假定可以共享队列，而实际上这些队列是独占的。
- 未经授权正确安全权限就尝试访问队列和数据。
- 在 MQI 调用中传递了不正确的参数；如果传递了错误数量的参数，那么无法尝试完成完成代码和原因码字段，任务会异常地结束。
- 无法检查来自 MQI 请求的返回码。
- 使用不正确的地址。
- 传递指定了不正确的长度的变量。
- 以错误的顺序传递参数。
- 无法正确地初始化 *MsgId* 和 *CorrelId*。

3. IBM MQ 应用程序从上次成功运行起是否更改过？

Yes

错误可能位于应用程序新的或修改过的部分中。检查所有更改，看您能否找到明显的问题原因。

- a. 应用程序的所有功能以前都运用过吗？

问题可能是在第一次使用以前从未调用的应用程序的一部分时发生的吗？如果是，那么可能错误是位于应用程序的这一部分的。尝试找出应用程序失败时在做什么，并检查该程序的这一部分的源代码以找出错误。

- b. 如果程序以前成功运行过，请检查当前队列状态和错误发生时正在处理的文件。它们可能包含一些不平常的数据值，导致调用程序中很少使用的路径。

- c. 应用程序收到意外的 MQI 返回码。例如：

- 您的应用程序假设其访问的队列是可以共享的吗？如果队列已被重新定义为独占的，您的应用程序能否处理表明其不再可访问该队列的返回码吗？
- 队列定义或安全概要文件有过更改吗？MQOPEN 调用可能由于违反安全性而失败；您的应用程序可从产生的返回码恢复吗？

请参阅适用于您的编程语言的 [MQI 应用程序参考](#)，以获取每个返回码的描述。

- d. 如果已将任何 PTF 应用于 IBM MQ for IBM i，请检查在安装 PTF 时未收到任何错误消息。

否

确保您排除了前面所有的建议，继续至步骤 [第 19 页的『4』](#)。

4. 服务器系统从上次成功运行起是否未更改过？

Yes

转至第 20 页的 [『在 IBM i 上确定问题特征』](#)。

否

请考虑系统的各个方面，并查看有关更改可能会如何影响 IBM MQ 应用程序的相应文档。例如：

- 与其他应用程序的接口
- 新操作系统或硬件的安装
- PTF 的应用

下一步做什么

相关任务

[第 22 页的『手动应用命令和程序的必需权限』](#)

某些 IBM MQ 命令依靠使用 IBM i 系统命令来创建和管理对象、文件和库，例如，CRTMQM（创建队列管理器）和 DLTMQM（删除队列管理器）。类似地，某些 IBM MQ 程序代码（例如，队列管理器）会使用 IBM i 系统程序。

[第 7 页的『在 UNIX, Linux, and Windows 上执行初步检查』](#)

开始在 UNIX, Linux, and Windows 上详细确定问题之前，请考虑是否存在引起问题的明显原因，或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

[第 26 页的『在 z/OS 上执行初步检查』](#)

开始在 z/OS 上详细确定问题之前，请考虑是否存在引起问题的明显原因，或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

[第 227 页的『联系 IBM 支持人员』](#)

如果您在 IBM MQ 方面遇到问题需要帮助，可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订、故障诊断和其他新闻的通知。

相关参考

[第 24 页的『确定应用程序、命令和消息的问题』](#)

如果遇到 IBM MQ 应用程序、命令和消息的问题，那么您可以考虑大量问题来帮助确定问题原因。

[消息和原因码](#)

[PCF 原因码](#)

[故障诊断与支持参考](#)

IBM i 在 IBM i 上确定问题特征

如果使用初步检查无法确定问题的原因，那么现在应更详细地查看问题的特征。

使用以下问题作为提示来帮助确定问题原因：

- [第 20 页的『是否可以重现问题？』](#)
- [第 21 页的『问题是否为间歇性？』](#)
- [第 21 页的『命令的问题』](#)
- [第 21 页的『问题是否影响到 IBM MQ for IBM i 应用程序的所有用户？』](#)
- [第 21 页的『问题是否影响网络的特定部分？』](#)
- [第 21 页的『问题是否仅出现在 IBM MQ 上？』](#)
- [第 22 页的『问题是否在一天中的特定时间发生？』](#)
- [第 22 页的『未能收到命令的响应吗？』](#)

是否可以重现问题？

如果您可以重现问题，请考虑它重现的条件：

- 是由命令引起的吗？

如果由另一种方法输入，操作是否有效呢？如果在命令行输入，命令仍然有效，但是在其它情况下，命令是无效的，那么检查命令服务器是否已停止。还必须检查未更改 `SYSTEM.ADMIN.COMMAND.QUEUE` 的队列定义。

- 是由程序引起的吗？如果是，它在批处理中失败吗？这是否在所有 IBM MQ for IBM i 系统上失败？或仅在一部分系统上失败？

- 当问题发生时，您可以标识任何总好象在系统中运行的应用程序吗？如果可以，检查应用程序以查看它是否存在错误。
- 问题在任何队列管理器上都发生，还是连接到一个特定队列管理器时才发生？
- 问题在任何队列管理器上的同一类型对象上发生，还是仅在一个特定对象上发生？在清除或重新定义此对象后会发生什么？
- 问题独立于任何消息持久性设置吗？
- 问题仅在使用同步点时发生吗？
- 问题仅在启用一个或多个队列管理器事件时发生吗？

问题是否为间歇性？

间歇出现问题可能是由于未考虑到各个进程可以彼此独立地运行。例如，完成更早的进程前，程序可能没有指定等待选项就发出 MQGET 调用。如果应用程序在放入消息的调用未确定时（也即，在落实或回退前）尝试从队列取出消息，那么您也可能遇到此问题。

命令的问题

使用此信息来避免特殊字符可能出现的问题。在描述性文本中针对某些命令使用特殊字符（例如，反斜杠 (\) 和引号 (") 字符）时，请小心操作。如果在描述性文本中使用了这些字符中的任一字符，请在其前面加一个反斜杠 (\) 字符，例如：

- 如果文本中需要反斜杠 (\) 字符，请输入 \\。
- 如果文本中需要引号 (") 字符，请输入 \"。

队列管理器及其关联的对象名称区分大小写。缺省情况下，IBM i 使用的是大写字符，除非使用撇号 (') 字符将名称括起来。

例如，MYQUEUE 和 myqueue 会转换为 MYQUEUE，而 'myqueue' 会转换为 myqueue。

问题是否影响到 IBM MQ for IBM i 应用程序的所有用户？

如果问题仅影响某些用户，请查找这些用户配置其系统和队列管理器设置时有何差别。

检查库列表和用户概要文件。具有 *ALLOBJ 权限时能否避免问题？

问题是否影响网络的特定部分？

您可能可以标识受问题影响的网络的特定部分（例如，远程队列）。如果到远程消息队列管理器的链接不起作用，那么消息无法流向远程队列。

请检查以下几点：

- 两个系统之间的连接是否正常，以及是否启动了 IBM MQ for IBM i 的内部通信组件？
确保该消息到达传输队列、传输队列的本地队列定义和任何远程队列。
- 是否已做出任何可能会解释此问题的与网络相关的更改？或是否已更改任何 IBM MQ for IBM i 定义？
- 您能分辨通道定义问题和通道消息问题吗？

例如，重新定义通道使用空的传输队列。如果通道正确启动，那么定义配置正确。

问题是否仅出现在 IBM MQ 上？

如果问题仅出现在该版本 IBM MQ 上，请检查 RETAIN 上相应的数据库或 https://www.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_MQ，以确保应用所有相关的 PTF。

问题是否在一天中的特定时间发生？

如果问题发生在一天中的特定时间，那么可能与系统负载有关。通常，系统负载峰值是在上午的中间时段和下午的中间时段发生的，因此在此时段很可能发生与负载相关的问题。（如果 IBM MQ for IBM i 网络跨越了多个时区，那么系统负载峰值可能会看似发生在一天中的其他时间。）

未能收到命令的响应吗？

如果您发出了命令但没有收到响应，考虑以下问题：

- 命令服务器在运行吗？

使用 `DSPMQCSVR` 命令来检查命令服务器的状态。

- 如果该命令的响应表明命令服务器没有运行，请使用 `STRMQCSVR` 命令启动它。
- 如果该命令的响应表明 `SYSTEM.ADMIN.COMMAND.QUEUE` 不是为 `MQGET` 请求启用的，那么启用 `MQGET` 请求的队列。

- 已将应答发送到死信队列了吗？

死信队列头结构包含原因或反馈代码，对该问题进行了描述。请参阅 [MQDLH - 死信消息头](#)，以获取有关死信队列头结构 (`MQDLH`) 的信息。

如果死信队列包含消息，那么您可以使用所提供的浏览样本应用程序 (`amqsbcbg`) 来使用 `MQGET` 调用浏览消息。样本应用程序为已命名队列管理器单步遍历已命名队列上的所有消息，为已命名队列上的所有消息显示消息描述符和消息上下文字段。

- 消息被发送到错误日志了吗？

请参阅第 289 页的『[IBM i 上的错误日志](#)』，以了解更多信息。

- 队列启用了放入和取出操作了吗？

- `WaitInterval` 够长了吗？

如果 `MQGET` 调用超时，将返回完成代码 `MQCC_FAILED` 和原因码 `MQRC_NO_MSG_AVAILABLE`。（请参阅使用 `MQGET` 调用从队列取出消息，以获取有关 `WaitInterval` 字段以及 `MQGET` 的完成代码和原因码的更多信息。）

- 如果您使用自己的应用程序将命令放到 `SYSTEM.ADMIN.COMMAND.QUEUE` 上，您还需要获取同步点吗？

除非已从同步点排除请求消息，否则必须在尝试接收回复消息前获取同步点。

- 队列的 `MAXDEPTH` 和 `MAXMSGL` 属性设置的是否足够高？

- 您是否正确使用了 `CorrelId` 和 `MsgId` 字段？

在应用程序中设置 `MsgId` 和 `CorrelId` 的值，以确保接收来自队列的所有消息。

相关任务

第 5 页的『[IBM MQ 故障诊断和支持](#)』

如果您的队列管理器网络或 IBM MQ 应用程序存在问题，那么可以使用本信息中描述的方法来帮助您诊断和解决问题。如果需要有关问题的帮助，可以通过 IBM 支持站点联系 IBM 支持人员。

第 22 页的『[手动应用命令和程序的必需权限](#)』

某些 IBM MQ 命令依靠使用 IBM i 系统命令来创建和管理对象、文件和库，例如，`CRTMQM`（创建队列管理器）和 `DLTMQM`（删除队列管理器）。类似地，某些 IBM MQ 程序代码（例如，队列管理器）会使用 IBM i 系统程序。

相关参考

第 24 页的『[确定应用程序、命令和消息的问题](#)』

如果遇到 IBM MQ 应用程序、命令和消息的问题，那么您可以考虑大量问题来帮助确定问题原因。

IBM i 手动应用命令和程序的必需权限

某些 IBM MQ 命令依靠使用 IBM i 系统命令来创建和管理对象、文件和库，例如，`CRTMQM`（创建队列管理器）和 `DLTMQM`（删除队列管理器）。类似地，某些 IBM MQ 程序代码（例如，队列管理器）会使用 IBM i 系统程序。

关于此任务

要启用此依赖关系，命令和程序必须对 IBM MQ 用户概要文件 QMQM 和 QMQMADM 具有 *PUBLIC *USE 权限或显式 *USE 权限。

此类权限将在安装过程中自动应用，您无需自己应用。但是，如果遇到问题，您可以按照以下步骤中所述手动设置权限。

过程

1. 使用 GRTOBJAUT 命令和 OBJTYPE(*CMD) 参数设置命令的权限，例如：

```
GRTOBJAUT OBJ(QSYS/ADDLIB) OBJTYPE(*CMD) USER(QMQMADM) AUT(*USE)
```

您可以设置对以下命令的权限：

- QSYS/ADDLIB
- QSYS/ADDPFM
- QSYS/CALL
- QSYS/CHGCURLIB
- QSYS/CHGJOB
- QSYS/CRTJRN
- QSYS/CRTJRNRCV
- QSYS/CRTJOBQ
- QSYS/CRTJOBQ
- QSYS/CRTLIB
- QSYS/CRTMSGQ
- QSYS/CRTPF
- QSYS/CRTPGM
- QSYS/CRTSRCPF
- QSYS/DLTJRN
- QSYS/DLTJRNRCV
- QSYS/DLTLIB
- QSYS/DLTMSGQ
- QSYS/OVRPRTF
- QSYS/RCLACTGRP
- QSYS/RTVJRNE
- QSYS/RCVJRNE
- QSYS/SBMJOB

2. 使用 GRTOBJAUT 命令和 OBJTYPE(*PGM) 参数设置程序的权限，例如：

```
GRTOBJAUT OBJ(QSYS/QWTSETP) OBJTYPE(*PGM) USER(QMQMADM) AUT(*USE)
```

您可以设置对以下程序的权限：

- QSYS/QWTSETP(*PGM)
- QSYS/QSYRLSPH(*PGM)
- QSYS/QSYGETPH(*PGM)

IBM i 确定应用程序、命令和消息的问题

如果遇到 IBM MQ 应用程序、命令和消息的问题，那么您可以考虑大量问题来帮助确定问题原因。

使用以下问题作为提示来帮助确定问题原因：

某些队列是否正常工作？

如果您怀疑问题仅在队列的子集发生，请选择您认为有问题的本地队列的名称。

1. 使用 WRKMQMSTS 或 DSPMQMQ 显示关于此队列的信息。
2. 使用所显示的数据来执行以下检查：
 - 如果 CURDEPTH 是在 MAXDEPTH 的，那么没有处理队列。检查所有应用程序都是正常运行的。
 - 如果 CURDEPTH 不是 MAXDEPTH，请检查以下队列属性来确保其正确：
 - 如果正在使用触发：
 - 是否正在运行触发器监视器？
 - 触发器深度太大吗？
 - 进程名正确吗？
 - 队列可以共享吗？如果不可以，那么另一个应用程序可能已经为输入打开了该队列。
 - 队列相应地启用了 GET 和 PUT 吗？
 - 若无应用程序进程从队列取出消息，请确定原因（例如，由于必须启动应用程序，连接已中断或由于某种原因 MQOPEN 调用已失败）。

如果无法解决该问题，请联系 IBM 支持中心以获取帮助。

问题是否仅影响远程队列？

如果问题仅影响远程队列，请检查后续点：

1. 确保应该将消息放入远程队列的程序成功运行。
2. 如果您使用触发来启动分布式排队进程，应检查传输队列是否已将触发设置为打开。还应检查触发器监视器是否正在运行。
3. 若有必要，手动启动通道。请参阅[分布式排队和集群](#)。
4. 用 PING 命令检查通道。

消息是否未能到达队列？

如果消息未能如预期到达，请检查以下事项：

- 您选择了正确的队列管理器（即缺省队列管理器或已命名队列管理器）吗？
- 消息成功放入队列了吗？
 - 队列正确定义了吗（例如，MAXMSGLEN 是否够大）？
 - 应用程序能否将消息放入队列中（为放入启用队列）？
 - 如果队列已满，那么可能意味着，应用程序无法在队列上放入所需的消息。
- 您可以从队列取出消息吗？
 - 您必须获取同步点吗？

如果在同步点中放入或检索消息，在落实恢复单元前，它们不可用于其它任务。
 - 您的超时间隔足够长吗？
 - 您在等待由消息标识或相关标识 (*MsgId* 或 *CorrelId*) 指示的特定消息吗？

检查您是否在用正确的 *MsgId* 或 *CorrelId* 等待消息。成功的 MQGET 调用将这两个值都设置为检索的消息的值，因此，您可能需要复位这些值才能成功地取出另一个消息。

同样，检查能否从队列获取其他消息。

- 其他应用程序可以从队列取出消息吗？
- 您预期的消息是定义为持久的吗？

如不是，并且重新启动了 IBM MQ for IBM i，那么会丢失此消息。

如果您找不到队列有什么错误，并且队列管理器本身正在运行，那么在您预期将消息放入队列的过程中进行以下检查：

- 应用程序启动了吗？

如果应该已经触发了该应用程序，请检查是否指定了正确的触发器选项。

- 是否正在运行触发器监视器？
- 正确定义了触发器进程吗？
- 它正确完成了吗？

查找作业日志中的异常结束证据。

- 应用程序落实其更改了吗？或将它们回退了吗？

如果多个事务正在服务队列，那么它们偶尔可能彼此冲突。例如，一个事务可能发出带有 0 的缓冲区长度的 MQGET 调用来查找消息的长度，然后发出指定该消息的 *MsgId* 的特定 MQGET 调用。然而，与此同时，另一个事务为该消息发出了成功的 MQGET 调用，因此第一个应用程序接收了完成代码 MQRC_NO_MSG_AVAILABLE。必须将要在多服务器环境中运行的应用程序设计为可应付这样的情况。

请考虑可能已收到的消息，但出于某种原因您的应用程序未能处理它。例如，是由于所要的消息格式出错而导致程序拒绝它吗？如果是，请参阅第 25 页的『使用分布式队列时是否收到意外的消息？』。

消息包含意外或损坏的信息吗？

如果在消息中包含的信息不是应用程序所需的，或出现某种程度的损坏，请考虑以下几点：

- 您的应用程序或将消息放入队列上的应用程序，被更改了吗？

确保在所有需要注意更改的系统上都同步反映了全部更改。

例如，格式化消息的副本文件可能已被更改，无论哪种情况，请都重新编译应用程序来使更改生效。如果一个应用程序还没有被重新编译，那么将对其余应用程序来说显示为数据损坏。

- 应用程序对错误队列发送了消息吗？

检查确保应用程序正在接收的消息并非应该被发送到服务于另一个队列的应用程序。若有必要，请将安全性定义更改为阻止未授权的应用程序将消息放入错误的队列。

如果应用程序使用别名队列，请检查别名是否指向正确的队列。

- 已经为该队列指定了正确的触发器信息吗？

检查应该已经启动了应用程序；或应该启动了不同的应用程序吗？

- 是否正确设置了 CCSID，或因为数据转换而导致消息格式不正确？

如果这些检查没有使您解决问题，那么为发送消息的程序和接收消息的程序检查应用程序逻辑。

使用分布式队列时是否收到意外的消息？

如果您的应用程序使用分布式队列，考虑以下要点：

- 分布式排队已经被正确安装在发送和接收系统上了吗？
- 在两个系统之间的链接是可用的吗？

检查两个系统是否都可用并连接到 IBM MQ for IBM i。检查两个系统之间的连接是活动的。

- 在发送系统中设置了触发吗？
- 您所等待的消息是从远程系统来的应答消息吗？

检查远程系统中的触发已被激活。

- 队列已经满了吗？

如果已满，那么可能意味着，应用程序无法在队列上放入所需的消息。检查消息是否已放入未送达消息的队列上。

死信队列消息头（死信队列头结构）包含原因或反馈代码，解释为什么消息无法被放入目标队列。有关死信头结构的信息，请参阅 [MQDLH - 死信头](#)。对于 IBM i，另请参阅 [IBM i Application Programming Reference \(ILE/RPG\)](#)。

- 在发送和接收队列管理器之间有不匹配吗？

例如，消息长度可能超过了接收队列管理器可处理范围的长度。

- 发送和接收通道的通道定义是兼容的吗？

例如，顺序号包中的不匹配会停止分布式排队组件。请参阅 [分布式排队和集群](#)。

在 z/OS 上执行初步检查

开始在 z/OS 上详细确定问题之前，请考虑是否存在引起问题的明显原因，或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

关于此任务

问题可能出在：

- IBM MQ
- 网络
- 应用程序
- 已配置可使用 IBM MQ 的其他应用程序

过程

- 考虑以下问题列表。在核对该列表的过程中，请记下可能与问题相关的所有信息。即使您的调查没有直接给出原因，但如果您必须完成系统的问题确定练习，它们稍后可能对您是有帮助的。
 - [第 27 页的『IBM MQ for z/OS 以前是否成功运行过？』](#)
 - [第 27 页的『是否已应用任何 APAR 或 PTF？』](#)
 - [第 27 页的『有任何错误消息、返回码或其他错误情况吗？』](#)
 - [第 29 页的『应用程序或 IBM MQ for z/OS 是否已停止处理工作？』](#)
 - [第 29 页的『IBM MQ 队列是否有问题？』](#)
 - [第 30 页的『某些队列是否正常工作？』](#)
 - [第 31 页的『是否定义了正确的队列？』](#)
 - [第 31 页的『问题是否仅影响远程或集群队列？』](#)
 - [第 31 页的『问题是否仅影响共享队列？』](#)
 - [第 32 页的『问题是否影响网络的特定部分？』](#)
 - [第 32 页的『发生在一天中的特定时间或影响特定用户的问题』](#)
 - [第 32 页的『问题是间歇性的，还是发生在所有 z/OS、CICS 或 IMS 系统上？』](#)
 - [第 33 页的『应用程序以前运行成功吗？』](#)
 - [第 34 页的『自上次成功运行以来是否进行了任何更改？』](#)
 - [第 34 页的『是否有程序错误？』](#)
 - [第 35 页的『是否存在异常终止？』](#)
 - [第 36 页的『您得到了不正确的输出吗？』](#)
 - [第 36 页的『是否可以重现问题？』](#)
 - [第 36 页的『您是否未能从 MQSC 命令接收响应？』](#)

相关任务

第 7 页的『在 UNIX, Linux, and Windows 上执行初步检查』

开始在 UNIX, Linux, and Windows 上详细确定问题之前, 请考虑是否存在引起问题的明显原因, 或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

第 18 页的『在 IBM i 上执行初步检查』

开始在 IBM i 上详细确定问题之前, 请考虑是否存在引起问题的明显原因, 或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

第 227 页的『联系 IBM 支持人员』

如果您在 IBM MQ 方面遇到问题需要帮助, 可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订、故障诊断和其他新闻的通知。

[故障诊断与支持参考](#)

相关参考

[消息和原因码](#)

[PCF 原因码](#)

IBM MQ for z/OS 以前是否成功运行过?

了解之前是否成功运行了 IBM MQ for z/OS 可能有助于确定问题, 可执行几项有益的检查。

如果此问题的答案是否, 请考虑以下事项:

- 检查您的设置。

如果 IBM MQ 之前未在 z/OS 上成功运行, 那么您可能尚未正确设置该值。请参阅[安装 IBM MQ for z/OS 产品中有关安装和定制队列管理器的信息](#), 以获取进一步的指导。
- 验证安装。
- 检查是否发出消息 CSQ9022I 来响应 START QMGR 命令 (指示正常完成)。
- 确保 z/OS 将 IBM MQ 显示为安装的子系统。要确定 IBM MQ 是否为安装的子系统, 请使用 z/OS 命令 D OPDATA。
- 检查安装验证程序 (IVP) 是否成功运行。
- 发出 DISPLAY DQM 命令以检查通道启动程序地址空间是否正在运行, 并检查相应的侦听器是否已启动。

是否已应用任何 APAR 或 PTF?

APAR 和 PTF 偶尔可能会导致 IBM MQ 发生意外的问题。这些修订可能已应用于 IBM MQ 或其他 z/OS 系统。

如果 APAR 或 PTF 已应用于 IBM MQ for z/OS, 请检查未生成任何错误消息。如果成功安装, 请与 IBM 支持中心核对一下, 是否存在任何 APAR 或 PTF 错误。

如果 APAR 或 PTF 已应用于任何其他产品, 请考虑它可能对 IBM MQ 与该产品的连接方式的影响。

确保您已遵循 APAR 中影响系统的任何指示信息。(例如, 您可能必须重新定义资源。)

有任何错误消息、返回码或其他错误情况吗?

使用此主题来研究错误消息、返回码以及终止队列管理器或通道启动程序的情况。

该问题可能会生成以下类型的错误消息或返回码:

CSQ 消息和原因码

IBM MQ for z/OS 错误消息具有前缀 CSQ。  如果接收到此前缀的任何消息 (例如, 在控制台日志或 CICS 日志中), 请参阅 [IBM MQ for z/OS 消息, 完成代码和原因码](#) 以获取说明。

其他消息

对于具有不同的前缀的消息, 请查看相应的消息和代码主题, 以了解建议的操作过程。

异常消息

了解与启动 IBM MQ for z/OS 关联的异常消息，或在出错前系统运行时发出的异常消息。任何异常消息均可能指示阻止应用程序成功运行的某一系统问题。

应用程序 MQI 返回码

如果应用程序收到返回码指示 MQI 调用失败，请参阅[返回码](#)以了解该返回码的描述。

您收到意外的错误消息或返回码了吗？

如果应用程序收到意外的错误消息，请考虑此错误消息是源自 IBM MQ，还是来自其他程序。

IBM MQ 错误消息

IBM MQ for z/OS 错误消息以 CSQ 字母为前缀。

如果收到意外的 IBM MQ 错误消息（例如，在控制台日志或 CICS 日志中），请参阅 [IBM MQ for z/OS 消息、完成代码和原因码](#) 以了解相关说明。

IBM MQ for z/OS 消息、完成代码和原因码可能会向您提供足够多的信息来快速解决问题，或使您重定向到另一手册以获得进一步的指导。如果无法处理消息，那么可能必须联系 IBM 支持中心以获取帮助。

非 IBM MQ 错误消息

如果您收到来自其他 IBM 程序或来自操作系统的错误消息，请从相应的库中查看消息和代码手册，以了解其含义的相关说明。

在队列共享环境中，查找以下错误消息：

- XES（以字母 IXL 为前缀）
- Db2（以字母 DSN 为前缀）
- RRS（以字母 ATR 为前缀）

意外返回码

如果应用程序收到来自 IBM MQ 的意外返回码，请参阅[返回码](#)，以获取有关应用程序可如何处理 IBM MQ 返回码的信息。

检查错误消息

发出 DISPLAY THREAD(*) 命令以检查队列管理器是否正在运行。有关此命令的更多信息，请参阅 [DISPLAY THREAD](#)。如果队列管理器停止运行，请查找可能说明此情况的任何消息。如果使用操作和控制面板，消息将在 z/OS 控制台或终端上显示。使用 DISPLAY DQM 命令来查看通道启动程序是否在运行，侦听器是否处于活动状态。z/OS 命令

```
DISPLAY R,L
```

列出具有未完成回复的消息。查看这些回复中的任何回复是否相关。例如，在某些情况下，当已使用所有活动日志后，IBM MQ for z/OS 会等待操作员干预。

未发出任何错误消息

如果未发出任何错误消息，请执行以下过程来确定引起问题的原因：

1. 发出 z/OS 命令

```
DISPLAY A,xxxxMSTR  
DISPLAY A,xxxxCHIN
```

（其中 xxxx 是 IBM MQ for z/OS 子系统名称）。如果收到一条消息称找不到队列管理器或通道启动程序，那么此消息指示子系统已终止。此情况可能由异常终止或操作员关闭系统所致。

2. 如果子系统正在运行，那么您将收到消息 IEE105I。此消息包含 CT=nnnn 字段，此字段中包含有关子系统使用的处理器时间的信息。请注意此字段的值，然后重新发出命令。
- 如果 CT= 值未更改，这指示子系统未使用任何处理器时间。这可能指示子系统正处于等待状态（或无任何要执行的工作）。如果可发出类似 DISPLAY DQM 的命令并恢复输出，这指示无工作要做，而不是挂起状况。
 - 如果大幅度更改了 CT= 值，并在重复显示期间持续这样操作，这可能指示子系统非常繁忙，或可能陷入循环。
 - 如果回复指示当前找不到子系统，这指示在发出第一条命令时，子系统处于终止过程中。如果在获取转储，那么子系统可能会花一点时间才能终止。终止前，控制台上会生成一条消息。
- 要检查通道启动程序是否正在运行，请发出 DISPLAY DQM 命令。如果响应未显示通道启动程序在运行，那么这可能是由于其资源不足（类似处理器）。在这种情况下，请使用 z/OS 监控工具（如 RMF）来确定是否存在资源问题。如不存在，请重新启动通道启动程序。

队列管理器或通道启动程序是否异常终止？

查找说明队列管理器或通道启动程序地址空间已异常终止的任何消息。如果收到系统操作将终止 IBM MQ 的消息，请查明是否生成了系统转储，请参阅 [IBM MQ 转储](#)。

IBM MQ for z/OS 可能仍在运行

还要考虑 IBM MQ for z/OS 可能仍在运行，但只是速度变慢。如果它运行缓慢，那么您可能遇到性能问题。要确认这一点，请参阅[应用程序或 IBM MQ for z/OS 是否运行缓慢](#)。请参阅[处理性能问题](#)以获取有关下一步操作的建议。

应用程序或 IBM MQ for z/OS 是否已停止处理工作？

系统意外地停止处理工作的原因有很多（包括队列管理器、应用程序、z/OS 和数据集等的问题）。

系统意外地停止处理工作的原因有很多。其中包括：

队列管理器问题

队列管理器可能正在关闭。

应用程序问题

应用程序编程错误可能意味着，程序未进行正常处理，或应用程序可能陷入循环。也可能是因为应用程序异常终止。

IBM MQ 问题

可能为 MQPUT 或 MQGET 调用禁用了队列、死信队列可能已满，或 IBM MQ for z/OS 可能处于等待状态或陷入循环。

z/OS 及其他系统问题

z/OS 可能处于等待状态，或者 CICS 或 IMS 可能处于等待状态或陷入循环。系统或综合系统级别上可能发生了影响队列管理器或通道启动程序的问题。例如，过度分页。它也可能指示 DASD 问题，或具有高处理器使用率的高优先级任务。

Db2 和 RRS 问题

检查 Db2 和 RRS 是否处于活动状态。

在所有情况下，都请执行以下检查来确定问题的原因：

IBM MQ 队列是否有问题？

使用此主题来研究 IBM MQ 队列可能存在的问题。

如果您怀疑子系统上发生了影响队列的问题，请使用操作和控制面板来显示系统命令输入队列。

如果系统响应

如果系统响应，那么至少有一个队列在运作。在这种情况下，请遵循第 30 页的『[某些队列是否正常工作？](#)』中的过程。

如果系统未响应

可能是整个子系统发生问题。在此实例中，尝试停止并重新启动队列管理器，响应生成的所有错误消息。

检查控制台上是否有任何消息需要操作。解析可能影响 IBM MQ 的任何消息，如为归档日志安装磁带的请求。查看其他子系统或 CICS 区域是否受到影响。

使用 DISPLAY QMGR COMMANDQ 命令来识别系统命令输入队列的名称。

如果问题在重新启动后仍然存在

请联系 IBM 支持中心以获取帮助（请参阅第 227 页的『联系 IBM 支持人员』）。

相关概念

第 31 页的『是否定义了正确的队列？』

IBM MQ 需要特定的预定义队列。如果未正确定义这些队列，那么可能会发生问题。

第 31 页的『问题是否仅影响远程或集群队列？』

如果问题仅在远程或集群队列上发生，请使用此主题进行进一步研究。

第 31 页的『问题是否仅影响共享队列？』

使用此主题来研究可能存在的队列共享组问题，这些问题可能会引起共享队列问题。

z/OS

某些队列是否正常工作？

使用此主题来研究队列的子集何时发生问题。

如果您怀疑问题仅在队列的子集发生，请选择您认为有问题的本地队列的名称，然后执行以下过程：

显示队列信息

使用 DISPLAY QUEUE 和 DISPLAY QSTATUS 命令来显示有关队列的信息。

是否正在处理队列？

- 如果 CURDEPTH 为 MAXDEPTH，那么它可能表明未在处理队列。请检查使用队列的所有应用程序是否正常运行（例如，检查 CICS 系统中的事务是否正在运行，或为响应队列深度高事件而启动的应用程序是否正在运行）。
- 发出 DISPLAY QSTATUS(xx) IPPROCS，以查看队列是否已打开可进行输入。如未打开，请启动应用程序。
- 如果 CURDEPTH 不是 MAXDEPTH，请检查以下队列属性来确保其正确：
 - 如果正在使用触发：
 - 是否正在运行触发器监视器？
 - 触发器深度太大吗？
 - 进程名正确吗？
 - 是否满足所有触发条件？

发出 DISPLAY QSTATUS(xx) IPPROCS，以查看应用程序是否打开了同一队列以进行输入。在某些触发场景中，如果队列已打开以进行输入，那么将不会生成触发器消息。停止应用程序将导致调用触发处理。

- 队列可以共享吗？如果不可以，那么另一个应用程序（批处理、IMS 或 CICS）可能已将其打开以进行输入。
- 队列相应地启用了 GET 和 PUT 吗？

您是否有长时间运行的工作单元？

如果 CURDEPTH 不是零，但在尝试对队列管理器回复无可用的消息进行 MQGET 处理时，请发出 DIS QSTATUS(xx) TYPE(HANDLE) 来向您显示有关已打开队列的应用程序的信息，或发出 DIS CONN(xx) 来向您提供有关连接到队列的应用程序的更多信息。

有多少任务正在访问队列？

发出 DISPLAY QSTATUS(xx) OPPROCS IPPROCS，以查看有多少任务在将消息放入队列及从队列取出消息。在队列共享环境中，检查每个队列管理器上的 OPPROCS 和 IPPROCS。或者，使用 CMDSCOPE 属

性来检查所有队列管理器。如果无应用程序进程从队列取出消息，请确定原因（例如，由于需要启动应用程序，连接已中断或由于某种原因 MQOPEN 调用已失败）。

此队列是共享队列吗？问题是否仅影响共享队列？

确认支持共享队列的综合系统元素没有问题。例如，确认受 IBM MQ 管理的耦合设施列表结构没有问题。

使用 D XCF, STRUCTURE, STRNAME=ALL 来确认耦合设施结构可供访问。

使用 D RRS 来确认 RRS 处于活动状态。

此队列是集群的一部分吗？

检查队列是否是集群的一部分（通过 CLUSTER 或 CLUSNL 属性）。如果是，请确认托管队列的队列管理器在集群中仍是活动的。

如果无法解决该问题

请联系 IBM 支持中心以获取帮助（请参阅第 227 页的『联系 IBM 支持人员』）。

z/OS 是否定义了正确的队列？

IBM MQ 需要特定的预定义队列。如果未正确定义这些队列，那么可能会发生问题。

检查是否正确定义了系统命令输入队列、系统命令回复模型队列以及应答队列，并检查是否成功调用了 MQOPEN。

如果使用系统命令回复模型队列，请检查其定义是否正确。

如果使用集群，那么需要定义 SYSTEM.CLUSTER.COMMAND.QUEUE，以使用与集群处理相关的命令。

z/OS 问题是否仅影响远程或集群队列？

如果问题仅在远程或集群队列上发生，请使用此主题进行进一步研究。

如果问题仅影响远程或集群队列，请检查：

是否正在访问远程队列？

确保将消息放入远程队列的程序成功运行（请参阅第 221 页的『处理 z/OS 上的不正确输出』）。

系统链接是活动的吗？

根据情况使用 APPC 或 TCP/IP 命令，来检查两个系统间的链接是否处于活动状态。

使用 PING 或 OPING（对于 TCP/IP 或 D NET ID=xxxxx），使用 E（对于 APPC）。

触发是否有效？

如果您使用触发来启动分布式排队进程，请确认传输队列已经将触发设置为打开并启用队列。

通道或侦听器是否正在运行？

必要时，手动启动通道或侦听器，或尝试停止和重新启动通道。请参阅[配置分布式排队](#)以获取更多信息。

启动通道启动程序和侦听器时，查找错误消息。请参阅 [IBM MQ for z/OS 消息、完成代码和原因码和配置分布式排队](#)，以确定原因。

通道状态是什么？

使用 DISPLAY CHSTATUS (channel_name) 命令检查通道状态。

进程和通道定义正确吗？

检查进程定义和通道定义。

请参阅[配置分布式排队](#)，以获取有关如何使用分布式排队的信息，以及如何定义通道的信息。

z/OS 问题是否仅影响共享队列？

使用此主题来研究可能存在的队列共享组问题，这些问题可能会引起共享队列问题。

如果问题仅影响队列共享组，请使用 CSQ5PQSG 实用程序的 VERIFY QSG 功能。此命令可确认 Db2 设置在位图分配字段、Db2 队列管理器的对象定义、结构以及共享队列对象方面保持一致，并报告所发现的任何不一致情况的详细信息。

以下是出错的 VERIFY QSG 报告的一个示例：

```
CSQU501I  VERIFY QSG function requested
CSQU503I  QSG=SQ02, DB2 DSG=DSN710P5, DB2 ssid=DFP5
CSQU517I  XCF group CSQGSQ02 already defined
CSQU520I  Summary information for XCF group CSQGSQ02
CSQU522I  Member=MQ04, state=QUIESCED, system=MV4A
CSQU523I  User data=D4E5F4C15AD4D8F0F4404040C4C5...
CSQU522I  Member=MQ03, state=QUIESCED, system=MV4A
CSQU523I  User data=D4E5F4C15AD4D8F0F3404040C4C6...
CSQU526I  Connected to DB2 DF4A
CSQU572E  Usage map T01_ARRAY_QMGR and DB2 table CSQ.ADMIN_B_QMGR inconsistent
CSQU573E  QMGR MQ04 in table entry 1 not set in usage map
CSQU574E  QMGR 27 in usage map has no entry in table
CSQU572E  Usage map T01_ARRAY_STRUC and DB2 table CSQ.ADMIN_B_STRUCTURE inconsistent
CSQU575E  Structure APPL2 in table entry 4 not set in usage map
CSQU576E  Structure 55 in usage map has no entry in table
CSQU572E  Usage map T03_LH_ARRAY and DB2 table CSQ.OBJ_B_QUEUE inconsistent
CSQU577E  Queue MYSQ in table entry 13 not set in usage map for structure APPL1
CSQU576E  Queue 129 in usage map for structure APPL1 has no entry in table
CSQU528I  Disconnected from DB2 DF4A
CSQU148I  CSQ5PQSG Utility completed, return code=12
```

问题是否影响网络的特定部分？

网络问题可能会导致 MQ for z/OS 出现相关问题。使用此主题来查看网络问题的可能的源。

您可能可以标识受问题影响的网络的特定部分（例如，远程队列）。如果指向远程队列管理器的链接不起作用，那么消息无法流向目标队列管理器上的目标队列。确认两个系统之间的连接正常以及已启动通道启动程序和侦听器。使用 MQSC PING CHANNEL 命令检查连接。

确认消息即将到达传输队列，并检查传输队列和任何远程队列的本地队列定义。使用 DISPLAY CHSTATUS 命令的 MQSC BYTSENT 关键字来检查数据是否正沿通道流动。使用 DISPLAY QLOCAL (XMITQ) CURDEPTH 来检查传输队列上是否发送了消息。检查位于通道两端的诊断消息，通知您已将消息发送至死信队列。

如果使用 IBM MQ 集群，请检查是否正确设置了集群定义。

是否已做出任何可能会导致此问题的与网络相关的更改？

是否已更改任何 IBM MQ 定义或任何 CICS 或 IMS 定义？检查传输队列的触发属性。

发生在一天中的特定时间或影响特定用户的问题

使用此主题来查看发生在一天中的特定时间或发生在特定的用户组上的 IBM MQ 问题。

如果问题发生在一天中的特定时间，那么可能与系统负载有关。通常，系统负载峰值是在上午的中间时段和下午的中间时段发生的，因此在这些时段最有可能发生与负载相关的问题。（如果网络跨越了多个时区，则系统负载峰值可能会看似发生在一天中的其他时间。）

如果您认为 IBM MQ for z/OS 系统存在性能问题，请参阅第 216 页的『处理 z/OS 的性能问题』。

如果问题仅影响某些用户，那么是否是因为这些用户未得到正确的安全授权？请参阅[进行安全性检查的用户标识](#)，以获取有关 IBM MQ for z/OS 所检查的用户标识的信息。

问题是间歇性的，还是发生在所有 z/OS、CICS 或 IMS 系统上？

查看此主题，考虑问题是否由应用程序交互引起，或是否与其他 z/OS 系统相关。

间歇出现问题可能是由于未考虑到各个进程可以彼此独立地运行。例如，完成更早的进程前，程序可能没有指定 WAIT 就发出 MQGET 调用。如果应用程序在位于同步点中时（也即，在落实前）尝试从队列取出消息，那么您也可能遇到此类问题。

如果仅在访问特定的 z/OS、IMS 或 CICS 系统时出现问题，请考虑该系统的不同之处。另请考虑对系统所做的任何更改是否可能会影响系统与 IBM MQ 交互的方式。

z/OS 应用程序以前运行成功吗？

可确定应用程序错误的方式通常有：确定其先前是否已成功运行；或其是否已生成错误消息和意外的返回码。

如果问题看来好象涉及特定的应用程序，请考虑以前应用程序是否成功运行。

在对此问题回答是之前，请考虑以下事项：

自从上次成功运行以来，对应用程序做了什么更改吗？

如果做了，错误可能位于应用程序新的或修改过的部分的某一处。研究所有更改，查看能否找到明显的问题原因。

应用程序的所有功能以前都运用过吗？

问题是在第一次使用以前从未启动的应用程序的一部分时发生的吗？如果是，那么可能错误是位于应用程序的这一部分的。尝试找出应用程序失败时在做什么，并检查该程序的这一部分的源代码以找出错误。

如果程序已经在以前许多的场合中成功运行，请检查当前队列状态和错误发生时正在处理的文件。它们可能包含一些不平常的数据值，导致调用程序中很少使用的路径。

应用程序检查所有返回码吗？

系统是否已略作更改？请检查应用程序接收的更改结果返回码。例如：

- 您的应用程序假设其访问的队列是可以共享的吗？如果队列已被重新定义为独占的，您的应用程序能否处理表明其不再可访问该队列的返回码吗？
- 任何安全概要文件均有过更改吗？MQOPEN 调用由于安全违例可能会失败；应用程序可从产生的返回码恢复吗？

应用程序是否预期特定的消息格式？

如果将具有非预期消息格式的消息放入队列（例如，来自其他平台上的队列管理器的消息），那么可能需要进行数据转换或其他形式的处理过程。

应用程序是否在其他 IBM MQ for z/OS 系统上运行？

导致问题的队列管理器的设置方式是否有所不同？例如，是否使用相同的最大消息长度或缺省优先级定义了队列？

应用程序是否使用 MQSET 调用来更改队列属性？

应用程序是否旨在将队列设置为无触发器，然后处理一部分工作并将队列设置为具有触发器？在队列被重置为具有触发器之前，应用程序可能会失败。

应用程序是否会处理导致其失败的消息？

如果应用程序由于消息损坏而失败，那么将回滚检索到的消息。下一个应用程序可能会获取相同的消息，并以相同的方式失败。确保应用程序使用回退计数；当达到回退计数阈值时，问题消息将被放入回退队列上。

如果应用程序以前从未成功地运行，请认真检查应用程序，查看能否找到以下任一错误：

转换和编译问题

在您查看代码之前，先检查来自转换程序、编译器或汇编程序和链接编辑器的输出，以了解是否报告了任何错误。如果应用程序转换、编译/汇编或链接编辑到装入库失败，那么当尝试调用它时它也会运行失败。请参阅[开发应用程序](#)，以获取有关构建应用程序的信息以及所需作业控制语言 (JCL) 语句示例。

批处理和 TSO 程序

对于批处理和 TSO 程序，检查是否已包含正确的存根。有一个批处理存根和两个 RRS 存根。如果您是使用 RRS，请确认未将 MQCMIT 和 MQBACK 调用用于 CSQBRSTB 存根。如果要继续将这些调用用于 RRS，请使用 CSQBRRSI 存根。

CICS 程序

对于 CICS 程序，请检查是否已按正确的顺序链接程序、IBM MQ CICS 存根和 CICS 存根。此外，还要检查是否针对 CICS 定义了程序或事务。

IMS 程序

对于 IMS 程序，请检查链接是否包含程序、IBM MQ 存根以及 IMS 语言接口模块。确保已指定正确的入口点。如果从 IMS 程序动态装入的程序要使用 IBM MQ，还必须链接存根和语言接口模块。

可能的代码问题

如果文档显示每一个步骤都是正确完成的，请考虑应用程序的编码。问题的症状是否指示此功能失败，并因此指示此代码段出错？请参阅第 34 页的『是否有程序错误？』，了解使 IBM MQ 应用程序发生问题的一些常见错误示例。

应用程序是否报告了 IBM MQ 的错误？

例如，队列可能未针对“gets”启用。它会接收指定此条件的返回码，但不会报告。请考虑应用程序在何处报告任何错误或问题。

自上次成功运行以来是否进行了任何更改？

从上一次成功运行起最近所做的更改通常是意外错误之源。此主题包含有关可作为问题确定过程的一部分研究的某些更改的信息。

当您考虑最近可能做出的更改时，请考虑一下 IBM MQ 及与之连接的其他程序（包括硬件和任何新应用程序）。还要考虑您尚不了解的新应用程序可能已在系统上运行的可能性。

初始化过程已更改了吗？

考虑这是否可能是问题的原因。是否更改了任何数据集？或更改了库定义？是否使用不同的参数初始化了 z/OS？此外，还要检查初始化期间发送给控制台的错误消息。

您已经更改任何队列定义或安全概要文件了吗？

考虑一下是否已更改一部分队列，使其成为集群成员。此更改可能意味着，消息源自不同的源（例如，其他队列管理器或应用程序）。

您已经在综合系统中更改与支持和实施共享队列相关的任何定义了吗？

考虑一下对综合系统耦合数据集或耦合设施资源管理策略等定义所做更改的影响。这些更改可能在执行共享队列的操作时进行。此外，还要考虑对 Db2 数据共享环境所做更改的影响。

z/OS 系统上的所有软件已升级到更高的发行版了吗？

考虑一下是否有任何必要的安装后或迁移活动需要执行。

您已经更改了 z/OS 子系统名称表了吗？

对诸如 z/OS 或 LE 等并存软件级别所做的更改可能需要对 IBM MQ 进行其他更改。

您的应用程序会处理那些因为您所做的更改而得到的返回码吗？

确保应用程序处理收到的任何新返回码。

是否有程序错误？

使用此主题来研究一下是否因为程序错误导致发生 IBM MQ 问题。

以下示例说明在运行 IBM MQ 程序时遇到的最常见的问题原因。考虑一下系统问题可能由于以下一项错误所致的可能性。

- 程序发出 MQSET 来更改队列属性，并且无法重置队列属性。例如，将队列设置为 NOTRIGGER。
- 就队列属性做出不正确的假设。此假设可能包括，在队列排斥 MQOPEN 时，假设可使用 MQOPEN 打开队列，以及假设队列不是其所属集群的一部分。
- 未经授权正确安全权限就尝试访问队列和数据。
- 将程序不与任何存根链接，或与错误的存根链接（例如，具有 CICS 存根的 TSO 程序）。这可能会产生长时间运行的工作单元，或发生 X'0C4' 或其他异常终止。
- 在 MQI 调用中传递了不正确或无效的参数；如果传递了错误数量的参数，那么无法尝试填写完成代码和原因码字段，任务会异常终止。（这是 X'0C4' 异常终止。）

如果您尝试在 MQSeries 的较早版本（早于为其编写应用程序的版本）上运行应用程序，那么可能会发生此问题，此时某些 MQI 值无效。

- 未能正确定义 IBM MQ 模块到 z/OS (此错误导致 X'0C4' 在 CSQYASCP 中异常终止)。
- 无法检查来自 MQI 请求的返回码。

如果您尝试在 IBM MQ 的更高版本（高于为其编写应用程序的版本，且其中引入了未经过检查的新返回码）上运行应用程序，那么可能会发生此问题。

- 无法使用后来调用 MQI 所需的正确选项打开对象，例如，使用 MQOPEN 调用打开队列，但未指定正确的选项来为后续调用 MQGET 启用队列。
- 无法正确地初始化 *MsgId* 和 *CorrelId*。

此错误对于 MQGET 尤其常见。

- 使用不正确的地址。
- 使用尚未初始化的储存。
- 传递指定了不正确的长度的变量。
- 以错误的顺序传递参数。
- 无法向 RACF 定义正确的安全概要文件和类。

这可能会使队列管理器停止运行，或阻止您执行任何生产性工作。

- 针对移植的应用程序使用缺省 MQI 选项。

例如，在同步点中，z/OS 缺省情况下使用 MQGET 和 MQPUT。缺省情况下，分布式平台不在同步点中。

- 在门户网站应用程序正常或异常终止时，使用缺省行为。

在 z/OS 上，正常终止会执行隐式 MQCMIT，异常终止会执行隐式回滚。

是否存在异常终止？

使用此主题来研究异常终止的常见原因，以及可能引起问题的不同类型的异常终止。

如果应用程序已停止运行，那么可能由异常终止所致。

根据您所使用的应用程序类型，系统将通知您以下某一位置出现异常终止：

批处理

您的列表显示异常终止。

CICS

您将看到一条 CICS 事务异常终止消息。如果任务是终端任务，那么此消息将在屏幕上显示。如果任务未连接至终端，那么此消息将在 CICS CSMT 日志上显示。

IMS

在所有情况下，您都将在 IBM MQ for IMS 主终端以及所涉及的从属区域的列表中看到一条消息。如果正在处理从终端输入的 IMS 事务，那么还会将错误消息发送至该终端。

TSO

您可能在屏幕上看到一条 TSO 消息及返回码。（是否显示此消息取决于设置系统的方式以及错误类型。）

异常终止的常见原因

如果用户在所执行任务正常终止前结束任务，那么可能会导致异常终止；例如，如果清除 CICS 事务。异常终止也可能由应用程序中的错误所致。

地址空间转储和事务转储

对于某些异常终止，将生成地址空间转储。对于 CICS 事务，将提供显示事务相关存储区域的事务转储。

- 如果应用程序传递了某些数据且这些数据的地址不再有效，那么有时会在用户地址空间中生成转储。

注：对于批处理转储，此转储将格式化并写入 SYSUDUMP。有关 SYSUDUMP 的信息，请参阅第 214 页的『z/OS 上的 SYSUDUMP 信息』。对于 CICS，系统转储将写入 SYS1.DUMP 数据集，以及正在采用的事务转储。

- 如果 IBM MQ for z/OS 本身的问题导致异常终止，那么将返回异常终止代码 X'5C6' 或 X'6C6' 以及异常终止原因码。此原因码会唯一地描述该问题的原因。请参阅第 182 页的『IBM MQ for z/OS 异常终止』以了解异常终止代码的信息，并参阅[返回码](#)以获取原因码的说明。

程序异常终止

如果程序已异常终止，请参阅第 183 页的『在 IBM MQ for z/OS 上处理异常终止』。

如果系统已异常终止，并且您希望分析所生成的转储，请参阅第 198 页的『IBM MQ for z/OS 转储』。本部分将向您介绍如何对转储进行格式化，以及如何解释其中包含的数据。

z/OS 您得到了不正确的输出吗？

使用此主题来查看已接收的任何不正确的输出。

如果您已获得您认为是一些不正确的输出，请考虑以下操作：

对不正确的输出分类

“不正确的输出”可能会被视为非预期的任何输出。然而，在确定问题时，请小心使用该词，因为它可能会产生某些其他类型的错误的再次影响。例如，如果获得任何重复输出，那么可能会进入循环，即使输出如您所预期也是如此。

错误消息

IBM MQ 还通过发送错误消息来响应其检测到的多个错误。您可能会将这些消息视为“不正确的输出”，但它们仅是另一类问题的症状。如果从 IBM MQ 接收到非预期的错误消息，请参阅第 27 页的『有任何错误消息、返回码或其他错误情况吗？』。

意外的消息

如果应用程序未收到预期的消息，已接收包含意外或损坏的信息的消息，或已接收未预期接收的消息（例如，原本要传至另一应用程序的消息），请参阅第 221 页的『处理 z/OS 上的不正确输出』。

z/OS 是否可以重现问题？

可使用重现问题来帮助为 IBM MQ for z/OS 确定问题。使用此主题来进一步隔离问题重现类型。

如果您可以重现问题，请考虑它可重现的条件。例如：

是由命令引起的吗？

如果是这样，那么是从 z/OS 控制台，从 CSQUTIL，从为将命令放入 SYSTEM.COMMAND.INPUT 队列，还是使用操作和控制面板？

如果由另一种方法输入，命令是否有效呢？

如果在控制台上输入时命令有效，但是在其他情况下，命令是无效的，则检查命令服务器是否已停止并且是否更改了 SYSTEM.COMMAND.INPUT 队列的队列定义。

命令服务器在运行吗？

发出 DIS CMDSERV 命令进行检查。

是由应用程序引起的吗？

如果是，它在 CICS、IMS、TSO 或批处理中会失败吗？

这是否在所有 IBM MQ 系统上失败？或仅在一部分系统上失败？

问题是由应用程序引起的吗？

当问题发生时，您可以标识任何总好象在系统中运行的应用程序吗？如果可以，检查应用程序以查看它是否存在错误。

z/OS 您是否未能从 MQSC 命令接收响应？

使用此主题来研究无法从 MQSC 命令接收响应的问题。

如果已从应用程序（且未从 z/OS 控制台）发出 MQSC 命令，但未收到响应，请考虑以下问题：

命令服务器在运行吗？

检查命令服务器是否正在运行，如下所述：

1. 在 z/OS 控制台上使用 DISPLAY CMDSERV 命令来显示命令服务器的状态。
2. 如果命令服务器未在运行，请使用 START CMDSERV 命令将其启动。
3. 如果命令服务器正在运行，请发出 DISPLAY QUEUE 命令。使用系统命令输入队列名称以及 CURDEPTH 和 MAXDEPTH 属性来定义所显示的数据。

如果这些值显示队列已满，并且已启动命令服务器，那么未从队列中读取消息。

4. 尝试停止命令服务器，然后重新启动，响应所产生的任何错误消息。
5. 重新发出显示命令，以查看它现在是否可运行。

已将应答发送到死信队列了吗？

使用 DISPLAY QMGR DEADQ 命令来查找系统死信队列的名称（如您不知道的话）。

在具有 CURDEPTH 属性的 DISPLAY QUEUE 命令中使用此名称，来查看队列上是否有任何消息。

死信队列消息头（死信头结构）包含原因码或反馈代码，对该问题进行了描述。（请参阅[原因 \(MQLONG\)](#)，以获取有关死信头结构的信息。）

是否针对 PUT 和 GET 启用了队列？

例如，从控制台上使用 DISPLAY QUEUE 命令来检查 DISPLAY QUEUE(SYSTEM.COMMAND.INPUT) PUT GET。

WaitInterval 参数是否设置为足够长的时间？

如果 MQGET 调用超时，那么应用程序会收到完成代码 2 和原因码 2033 (MQRC_NO_MSG_AVAILABLE)。(请参阅[WaitInterval \(MQLONG\)](#) 和 [MQGET-Get message](#)，以获取有关 **WaitInterval** 参数以及来自 MQGET 的完成代码和原因码的信息。)

是否需要同步点？

如果您使用自己的应用程序将命令放到系统命令输入队列上，请考虑是否必须获取同步点。

在尝试接收回复消息前及将消息放入队列后，必须获取同步点；或在放入消息时使用 MQPMO_NO_SYNCPOINT。除非已从同步点排除请求消息，否则必须在尝试接收回复消息前获取同步点。

队列的 MaxDepth 和 MaxMsgL 参数是否设置的足够高？

请参阅 [CSQ0016E](#)，以获取有关定义系统命令输入队列和应答队列的信息。

您是否在正确使用 CorrelId 和 MsgId 参数？

必须识别队列，然后显示 CURDEPTH。从控制台上使用 DISPLAY QUEUE 命令（例如，DISPLAY QUEUE (MY.REPLY.QUEUE) CURDEPTH），来查看应答队列上是否有未收到的消息。

在应用程序中设置 *MsgId* 和 *CorrelId* 的值，以确保接收来自队列的所有消息。

如果已从 z/OS 控制台（或等效设施）或应用程序发出 MQSC 命令，但未收到响应，那么以下问题适用：

队列管理器是否仍在运行？或命令是否引起异常终止？

查找指示异常终止的错误消息，如有，请参阅第 198 页的『[IBM MQ for z/OS 转储](#)』。

是否发出任何错误消息？

检查是否发出了任何可能指示错误性质的错误消息。

请参阅[发出命令](#)，以获取有关可用于输入 MQSC 命令的不同方法的信息。

z/OS 应用程序或 IBM MQ for z/OS 是否运行缓慢？

应用程序运行缓慢可能由于应用程序本身或包含 IBM MQ 在内的底层软件所致。使用此主题对运行缓慢的应用程序进行初始研究。

如果您的应用程序运行缓慢，那么可能表明它在循环中或在等待不可用的资源。

问题在系统负载峰值时间里是否更严重？

这还可能由性能问题所导致。可能是因为您的系统需要调整，或因为它是在接近其容量限制的情况下运作的。这类问题可能在系统负载的峰值时间最严重，通常在上午的中间时段和下午的中间时段。（如果网络跨越了多个时区，那么峰值系统负载对您来说可能会在其他时间出现。）

问题是否在系统负载较轻的情况下发生？

如果您发现性能降低与系统负载无关，且在系统轻负载很轻时偶尔也会发生，那么可能是由不良应用程序设计导致。可能仅在访问特定队列时才会显示该问题。

IBM MQ for z/OS 是否运行缓慢？

以下症状可能指示 IBM MQ for z/OS 正在缓慢运行：

- 如果系统响应命令缓慢。
- 如果队列深度重复显示，表明在为应用程序缓慢处理队列，对该应用程序您预计将要有大量的队列活动。

您可以在第 217 页的『[处理 z/OS 上运行缓慢或者已经停止的应用程序](#)』中找到有关处理等待和循环的指南，在第 216 页的『[处理 z/OS 的性能问题](#)』中找到有关处理性能问题的指南。

详细故障诊断

用于帮助您解决队列管理器网络或 IBM MQ 应用程序问题的故障诊断信息。

相关概念

第 285 页的『[使用错误日志](#)』

有各种错误日志可用于帮助进行问题确定和故障诊断。

第 293 页的『[First Failure Support Technology \(FFST\)](#)』

First Failure Support Technology (FFST) for IBM MQ 可提供有关事件的信息，在发生错误的情况下，这些信息有助于 IBM 支持人员对问题进行诊断。

相关任务

第 6 页的『[进行初始检查](#)』

您可以进行一些初始检查，这些检查可能会提供您可能遇到的常见问题的答案。

第 227 页的『[联系 IBM 支持人员](#)』

如果您在 IBM MQ 方面遇到问题需要帮助，可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订、故障诊断和其他新闻的通知。

第 304 页的『[使用跟踪](#)』

您可以使用不同类型的跟踪来帮助您确定问题并进行故障诊断。

对 AMS 问题进行故障诊断

提供了信息来帮助您找到并解决与 Advanced Message Security 相关的问题。

对于与 Advanced Message Security 相关的问题，请首先检查队列管理器错误日志。

com.ibm.security.pkcsutil.PKCSException: 对 AMS 的内容进行加密时发生错误

错误 com.ibm.security.pkcsutil.PKCSException: Error encrypting contents 暗示 Advanced Message Security 在访问加密算法时出现问题。

如果 Advanced Message Security 返回了以下错误：

```
DRQJP0103E The Advanced Message Security Java interceptor failed to protect message.
com.ibm.security.pkcsutil.PKCSException: Error encrypting contents
(java.security.InvalidKeyException: Illegal key size or default parameters)
```

请验证 `JAVA_HOME/lib/security/local_policy.jar/*.*.policy` 中的 JCE 安全策略是否授予对 MQ AMS 策略中使用的签名算法的访问权限。

如果未在当前安全策略中指定要使用的签名算法，请从以下位置下载适用于您产品版本的正确 Java 策略文件：[IBM Developer Kits](#)。

AMS 的 OSGi 支持

要将 OSGi 捆绑软件与 Advanced Message Security 配合使用，需要其他参数。

在 OSGi 捆绑软件启动期间运行以下参数：

```
-Dorg.osgi.framework.system.packages.extra=com.ibm.security.pkcs7
```

在 keystore.conf 中使用加密密码时，必须在 OSGi 捆绑软件正在运行时添加以下语句：

```
-Dorg.osgi.framework.system.packages.extra=com.ibm.security.pkcs7,com.ibm.misc
```

限制：AMS 支持将只使用 MQ Base Java 类的通信用于 OSGi 捆绑软件中受保护的队列。

将 AMS 与 JMS 配合使用时与打开受保护队列相关的问题。

使用 Advanced Message Security 时，打开受保护队列可能会出现各种问题。

您正在运行 JMS，并且接收到错误 2085 (MQRC_UNKNOWN_OBJECT_NAME) 以及错误 JMSMQ2008。

您已验证是否已设置 AMS，如 [Quick Start Guide for AMS with Java client](#) 中所述。

有许多 IBM MQ 选项不受 Advanced Message Security 支持或者有限制；在 [AMS 的已知限制](#) 中提供了详细信息。

您未设置 AMQ_DISABLE_CLIENT_AMS 环境变量。

解决问题

解决此问题有四种选择：

1. 在受支持的 IBM Java 运行时环境 (JRE) 下启动 JMS 应用程序。
2. 将应用程序移动到正在运行队列管理器的同一台计算机上，并使用绑定方式连接进行连接。
绑定方式连接将使用平台本机库来执行 IBM MQ API 调用。因此，将使用本机 AMS 拦截器来执行 AMS 操作，而不依赖于 JRE 的功能。
3. 请使用 MCA 拦截器，因为这样可以在消息到达队列管理器时立即对消息进行签名和加密，而无需客户机执行任何 AMS 处理。
假定对队列管理器应用了保护，必须使用备用机制来保护从客户机传输到队列管理器的消息。实现此目的最常用的方法是在应用程序使用的服务器连接通道上配置 TLS 加密。
4. 如果您不希望使用 AMS，请设置 AMQ_DISABLE_CLIENT_AMS 环境变量。

有关更多信息，请参阅 [消息通道代理程序 \(MCA\) 拦截](#)。

注：必须为 MCA 拦截器将消息传递到的每个队列制订安全策略。换句话说，目标队列需要制订 AMS 安全策略，其中应包含签署者和接收方的专有名称 (DN)，并且该专有名称与分配给 MCA 拦截器的证书的专有名称匹配。即，由队列管理器使用的 keystore.conf 中的 cms.certificate.channel.SYSTEM.DEF.SVRCONN 属性指定的证书的 DN。

对命令问题进行故障诊断

在描述性文本中使用特殊字符时显示的错误的故障诊断建议。

- **场景：**在某些命令的描述性文本中使用特殊字符时收到错误。
- **说明：**某些字符（例如，反斜杠 (\) 和双引号 (") 字符）在用于命令时具有特殊的含义。
- **解决方案：**在特殊字符前加一个 \，即，如果您在文本中需要 \ 或 "，请输入 \\ 或 \"。并不是所有字符都可用于命令。有关具有特殊含义的字符以及如何对其进行使用的更多信息，请参阅 [具有特殊含义的字符](#)。

对分布式发布/预订问题进行故障诊断

使用子主题中提供的建议，帮助您在发布/预订集群或层次结构时检测和处理问题。

开始之前

如果问题通常与集群相关（而不与使用集群发布/预订消息传递相关），请参阅第 139 页的『对队列管理器集群问题进行故障诊断』。

针对发布/预订集群中的保留发布的设计注意事项中还提供了一些有益的故障诊断提示。

相关概念

[分布式发布/预订系统队列错误](#)

相关任务

[配置发布/预订集群](#)

[设计发布/预订集群](#)

针对发布/预订集群的路由：行为注释

使用此处提供的建议，在使用集群式发布/预订消息传递功能时，帮助您检测路由问题并加以处理。

有关针对任何队列管理器集群进行状态检查和故障诊断的信息，请参阅第 139 页的『对队列管理器集群问题进行故障诊断』。

- 集群中同一已命名主题对象的所有集群定义必须具有相同的 **CLROUTE** 设置。您可以使用以下 MQSC 命令检查集群中所有主机上的所有主题的 **CLROUTE** 设置：

```
display tcluster(*) clroute
```

- 除非主题对象指定了 **CLUSTER** 属性的值，否则 **CLROUTE** 属性没有任何效果。
- 检查主题上的集群名称拼写是否正确。定义集群前，可定义诸如主题等集群对象。因此，当定义集群主题时，由于集群名称可能尚不存在，未对其进行验证。因此，产品未警告您集群名称拼写有误。
- 在设置 **CLROUTE** 属性时，如果队列管理器从另一个具有不同 **CLROUTE** 设置的队列管理器知道同一对象的集群定义，那么系统会生成 **MQRCCF_CLUSTER_TOPIC_CONFLICT** 异常。但是，通过不同的队列管理器上近乎同时的对象定义或与完整存储库建立不稳定的连接，可创建不同的定义。在这种情况下，完整的存储库队列管理器会进行仲裁，接受一个定义，并报告另一个定义的错误。要获取有关冲突的更多信息，请使用以下 MQSC 命令来检查集群中所有队列管理器上所有主题的集群状态：

```
display tcluster(*) clstate
```

无效或暂挂状态（如未立即激活）指出问题。如果检测到无效的主题定义，请识别不正确的主题定义并从集群中除去它。完整存储库具有有关接受和拒绝的定义的信息，创建冲突的队列管理器可在一定程度上指出问题的性质。另请参阅 [DISPLAY TOPIC](#) 中的 **CLSTATE**。

- 在主题树中的某个位置设置 **CLROUTE** 参数会导致其下方的整个分支以这种方式对主题进行路由。您无法更改此分支的子分支的路由行为。出于此原因，为主题树中具有不同 **CLROUTE** 设置的较低或较高节点定义主题对象时，将被拒绝，并出现 **MQRCCF_CLUSTER_TOPIC_CONFLICT** 异常。
- 可使用以下 MQSC 命令来检查主题树中所有主题的主题状态：

```
display tpstatus('#')
```

如果主题树中有大量的分支，那么上一条命令可能会显示不合适的大量主题的状态。在这种情况下，可改为显示可管理的树的一少部分分支，或树中的单个主题。显示的信息包含主题字符串、集群名称和集群路由设置。还包含发布者计数和预订计数（发布者和订户的数量），帮助您判断该主题用户数是否如您所预期。

- 更改集群中主题的路由是对发布/预订拓扑进行的一项重大更改。在对主题对象进行集群（通过设置 **CLUSTER** 属性）之后，无法更改 **CLROUTE** 属性的值。必须先将该对象取消集群（将 **CLUSTER** 设置为 ' '），然后您才能更改此值。对主题取消集群会将主题定义转换成本地主题，这将导致在某个时间段内，不会将发布传递到远程队列管理器上的预订；在执行此更改时应将此考虑在内。请参阅[定义与另一个](#)

队列管理器中集群主题具有相同名称的非集群主题的影响。如果尝试在集群时更改 **CLROUTE** 属性的值，那么系统将生成 **MQRCCF_CLROUTE_NOT_ALTERABLE** 异常。

- 对于主题主机路由，可通过在一系列集群队列管理器上添加和除去相同的集群主题定义，通过集群浏览备选路由。要防止给定的队列管理器充当集群的主题主机，请删除主题对象，或使用 **PUB (DISABLED)** 设置来停顿本主题的消息流量，如针对 **PUB** 参数进行特殊处理中所述。请勿通过将 **CLUSTER** 属性设置为 ' ' 取消集群主题，因为除去集群名称会将主题定义转换为本地主题，并阻止在此队列管理器中使用该主题时该主题的集群行为。请参阅定义与另一个队列管理器中集群主题具有相同名称的非集群主题的影响。
- 如果已将主题树的子分支集群设置为其他集群，并且 **CLROUTE** 设置为 **TOPICHOST**，那么您无法更改该分支的集群。如果在定义时检测到此定义，那么系统会生成 **MQRCCF_CLUSTER_TOPIC_CONFLICT** 异常。类似地，在更高的节点上为其他集群插入新集群化的主题定义时，将生成异常。由于先前描述了集群计时问题，如果以后检测到此不一致的情况，队列管理器会发出错误到队列管理器日志。

相关任务

[配置发布/预订集群](#)

[设计发布/预订集群](#)

检查代理预订位置

代理预订使发布能够在远程队列管理器上流向订户。如果订户未取出在队列管理器网络中的其他地点发布的信息，请检查代理预订是否在您所期望的地点进行。

缺少代理预订可能表明，应用程序未预订正确的主题对象或主题字符串，或主题定义出现问题，或通道未运行或未正确配置通道。

要显示代理预订，请使用以下 **MQSC** 命令：

```
display sub(*) subtype(proxy)
```

代理预订用于所有分布式发布/预订拓扑结构（层次结构和集群）。对于路由主题主机的集群主题，代理预订在该主题的每个主题主机上都存在。对于直接路由的集群主题，代理预订在集群中的每个队列管理器上都存在。通过在主题上设置 **proxysub(force)** 属性，也可以使代理预订存在于网络中的每个队列管理器上。

另请参阅[发布/预订网络中的预订性能](#)。

重新同步代理预订

正常情况下，队列管理器会自动确保系统中的代理预订能正确地反映网络中每个队列管理器上的预订。如果需要的话，可使用 **REFRESH QMGR TYPE (PROXYSUB)** 命令，手动将队列管理器的本地预订与队列管理器在网络中传播的代理预订进行重新同步。但是，只能在特殊情况下这样做。

何时手动重新同步代理预订

当队列管理器接收不得发送的预订，或未接收应接收的预订时，应考虑手动重新同步代理预订。但是，重新同步会在网络上临时创建意外的其他代理预订负载，它源自发出命令的队列管理器。因此，除非 **IBM MQ** 服务、**IBM MQ** 文档或错误日志记录指示您手动重新同步，否则请勿执行此操作。

如果队列管理器将自动进行重新验证，那么您无需手动重新同步代理预订。通常，队列管理器会在以下情况下，通过受影响的直接连接队列管理器重新验证代理预订：

- 形成分层连接时
- 在主题对象上修改 **PUBSCOPE**、**SUBSCOPE** 或 **CLUSTER** 属性时
- 重新启动队列管理器时

有时，配置错误会导致代理预订缺失或不相关：

- 如果使用 **Subscription scope** 设置为队列管理器，或者使用空或不正确的集群名称指定最接近的匹配主题定义，那么可能导致缺少代理预订。请注意，**Publication scope** 不会阻止发送代理预订，但会阻止将发布内容传递到这些预订。

- 如果将最匹配的主题定义指定为 **Proxy subscription behavior** 设置为**强制**，那么可能会导致额外的代理预订。

配置错误引发这些问题时，手动重新同步不会解决问题。在这些情况下，请修订配置。

以下列表描述了应手动重新同步代理预订的异常情况：

- 在发布/预订集群中的队列管理器上发出 **REFRESH CLUSTER** 命令后。
- 在队列管理器错误日志中的消息通知您运行 **REFRESH QMGR TYPE(REPOS)** 命令时。
- 当队列管理器无法正确传播其代理预订时，可能是因为通道已停止，并且所有消息都无法排队等待传输，或者是因为操作员错误导致从 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 队列中错误删除了这些消息。
- 从其他系统队列不正确地删除消息时。
- 在代理预订错误中发出 **DELETE SUB** 命令时。
- 作为灾难恢复的一部分。


如何手动重新同步代理预订

首先改正初始问题（例如，通过重新启动通道），然后在队列管理器上发出以下命令：

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

发出此命令时，队列管理器会向其直接连接的每一个队列管理器发送代理预订应存在的其自身的主题字符串列表。然后，直接连接的队列管理器会更新其保有的代理预订，以便与此列表匹配。之后，直接连接的队列管理器会将代理预订应存在的其自身的主题字符串列表发回至原始队列管理器，此原始队列管理器会相应地更新其保有的代理预订。

重要的用法注意事项：

- 由于未对受影响的预订恢复不正常的代理预订，因此错过发布。
- 重新同步需要队列管理器向其他队列管理器启动通道。如果您正在集群中使用直接路由，或者正在使用主题主机路由，并且在主题主机队列管理器上发出此命令，那么队列管理器会向集群中的所有其他队列管理器启动通道，即使未执行发布/预订操作的通道也会启动。因此，刷新的队列管理器必须具备足够强大的功能，来应对与集群中的所有其他队列管理器通信的情况。
-  如果在未运行 CHINIT 时，在 z/OS 上发出此命令，那么命令会在 CHINIT 启动时入队并进行处理。

相关概念

[针对发布/预订集群的 REFRESH CLUSTER 注意事项](#)

相关任务

[检查是否已完成用于分布式网络的异步命令](#)

分布式发布/预订网络中进行回路检测

在分布式发布/预订网络中，发布内容和代理预订不能形成回路至关重要；这是因为，回路将导致网络浪涌，即，相连接的订户接收到同一原始发布内容的多个副本。

发布/预订网络中的代理预订中描述的代理预订聚集系统不会阻止循环的形成，但它将会阻止代理预订的永久循环。由于发布内容的传播由代理预订的存在确定，因此，它们可能形成永久回路。IBM MQ 使用以下技术来防止发布内容形成永久回路：

当发布内容在发布/预订拓扑中移动时，每个队列管理器都将对消息头添加唯一的指纹。每当发布/预订队列管理器接收到来自另一发布/预订队列管理器的发布内容时，都将检查消息头中的指纹。如果它自己的指纹已存在，那么表明该发布内容已在回路中完整巡回，因此该队列管理器将废弃该消息并在错误日志中添加条目。

注：在回路中，发布内容将沿回路双向传播，回路中的每个队列管理器在始发队列管理器废弃回路的发布内容之前，都将接收到这两份发布内容。这将导致预订应用程序在回路中断前接收到发布内容的重复副本。

回路检测指纹格式

回路检测指纹作为 IBM MQ 8.0 协议的组合部分插入到 RFH2 头或流中。RFH2 程序员需要了解此头原封不动地传递指纹信息。更低版本的 IBM Integration Bus 使用不包含指纹信息的 RFH1 头。

```
<ibm>
  <Rfp>uuid1</Rfp>
  <Rfp>uuid2</Rfp>
  <Rfp>uuid3</Rfp>
  .
</ibm>
```

<ibm> 是一个文件夹的名称，此文件夹存放路由指纹的列表，这些路由指纹包含已访问的每个队列管理器的唯一用户标识 (UUID)。

队列管理器每次发布消息时，都使用 <Rfp> (路由指纹) 标记将其 UUID 添加到 <ibm> 文件夹中。每当收到发布内容时，IBM MQ 都会使用消息属性 API 来迭代 <Rfp> 标记，以查看该特定 uuid 值是否存在。由于使用排队的发布/预订接口时，IBM MQ 的 WebSphere 平台消息传递组件通过通道和 RFH2 预订连接到 IBM Integration Bus 的方式，因此 IBM MQ 在通过该路由接收发布时也会创建指纹。

目标是，如果应用程序不需要任何 RFH2，那么不要仅仅由于已添加指纹信息而将任何 RFH2 传递给应用程序。

只要将 RFH2 转换为消息属性，就还需要转换 <ibm> 文件夹；这将从 RFH2 中除去指纹信息，该信息将传递或送达至已使用 IBM WebSphere MQ 7.0 或更高版本 API 的应用程序。

JMS 应用程序不查看指纹信息，由于 JMS 接口不从 RFH2 中抽取该信息，因此不会将其传递给它的应用程序。

Rfp 消息属性是使用 `propDesc.CopyOptions = MQCOPY_FORWARD` and `MQCOPY_PUBLISH` 创建的。这将对接收并重新发布同一消息的应用程序产生影响。这意味着此类应用程序可以使用 `PutMsgOpts.Action = MQACTP_FORWARD` 继续路由指纹的链，但必须进行适当编码以从链中除去其自己的指纹。缺省情况下，应用程序使用 `PutMsgOpts.Action = MQACTP_NEW` 并启动新链。

对分布式队列管理问题进行故障诊断

故障诊断信息可帮助您解决与分布式队列管理 (DQM) 相关的问题。

描述的某些问题是特定于平台和安装的。如果遇到这类情况，会在文本中加以注明。

IBM MQ 提供名为 **amqldmpa** 的工具来协助进行问题确定。在问题确定的过程中，您的 IBM 服务代表可能会要求您提供工具的输出。

IBM 支持人员将为您提供收集相应诊断信息所需的参数，以及有关如何将记录的数据发送到 IBM 的信息。



注意: 您不应依赖此工具的输出格式，因为格式可能会发生变动而不通知您。

将讨论以下方案的问题确定：

- [第 46 页的『使用 Ping 测试通信』](#)
- [第 45 页的『死信队列注意事项』](#)
- [第 46 页的『对通道拒绝运行的问题进行故障诊断』](#)
- [第 49 页的『重试链接的注意事项』](#)
- [第 50 页的『解决通道停止运行的问题』](#)
- [第 45 页的『使用 dspmqrte 监视消息』](#)
- [第 50 页的『灾难恢复』](#)

相关任务

[第 6 页的『进行初始检查』](#)

您可以进行一些初始检查，这些检查可能会提供您可能迂到的常见问题的答案。

[配置分布式队列](#)

相关参考

[消息和原因码](#)

[z/OS 的通信协议返回码](#)

在何处查找信息以帮助进行故障诊断

根据您遇到的问题类型，可以使用许多可能的信息源来帮助您进行故障诊断。

命令验证问题

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

在创建，变更和删除通道时，将执行大量验证检查，并在适当情况下返回错误消息。在以下情况下可能会发生错误：

- 在创建通道时选择了重复的通道名称
- 在通道参数字段中输入了不能接受的数据
- 要更改的通道不确定，或不存在

问题诊断从解释错误消息开始，并采取纠正行动。

正常通道操作期间的处理问题

在通道正常运行期间发现的问题将通知系统控制台或系统日志。在 Windows 上，会将它们报告给通道日志。问题诊断从从日志中收集所有相关信息开始，然后继续进行分析以确定问题。如果可能，会将确认和错误消息返回到启动命令的终端。

在网络中，问题诊断可能很困难，在此网络中，问题可能出现在对某些消息进行登台的中间系统上。诸如传输队列已满并紧接着死信队列也满的错误情况，会导致到该网站的通道关闭。在此示例中，您在错误日志中接收到的错误消息将指示源自远程站点的问题，但可能无法告诉您有关该站点的错误的任何详细信息。因此，您必须与远程站点的对等方联系，以获取问题的详细信息，并接收有关该通道再次可用的通知。

通道启动协商错误

在通道启动期间，开始端必须指出其位置，并且与对应的通道就通道运行参数达成一致。可能会发生两端无法对参数达成一致的情况，在这种情况下，通道会关闭，并向相应的错误日志发出错误消息。

用户出口问题

通道程序与用户出口程序之间的交互作用具有一些错误检查例程，但此功能只能在用户出口遵循特定规则时成功运行。消息传递通道的通道出口程序中描述了这些规则。出错时，最有可能的结果是通道会停止，通道程序会发出错误消息以及来自用户出口的任何返回码。可通过扫描用户出口本身创建的消息，来确定接口的用户出口一侧检测到的任何错误。

您可能需要使用主机系统的跟踪功能来识别问题。

客户机应用程序问题

客户机应用程序可能会接收到意外错误返回码，例如：

- 队列管理器不可用
- 队列管理器名称错误
- 连接中断

请查看客户机错误日志以获取说明故障原因的消息。另外，错误也可能记录在服务器上，这取决于故障的性质。


注：即使客户机应用程序已终止，其代理进程仍有可能使其队列保持打开状态。通常这将仅持续较短时间，直至通信层通知合作伙伴已离开。

诊断消息和代码

有关有助于对问题进行主要诊断的消息和代码，请参阅 [消息和原因码](#)。

会计和统计数据

IBM MQ 生成记帐和统计数据，您可以使用这些数据来确定利用率和性能的趋势：

-  **Multi** 在多平台上，此信息生成为 PCF 记录，请参阅 [结构数据类型](#)。
-  **z/OS** 在 z/OS 上，此信息生成为 SMF 记录，请参阅 [监视性能和资源使用情况](#)。

数据结构

当在问题诊断期间检查日志和跟踪条目时，需要参考数据结构。

有关更多信息，请参阅 [Channel-exit 调用和数据结构](#) 和 [开发应用程序参考](#)。

相关概念

[通道控制功能](#)

死信队列注意事项

在某些 IBM MQ 实现中，死信队列又被称为未送达消息的队列。

如果通道因任何原因停止运行，应用程序可能会继续将消息放入传输队列，从而可能产生溢出情况。应用程序可以监控传输队列以查找等待发送的消息数量，但这不是通常执行的功能。

如果此情况在消息原始节点中发生，并且本地传输队列已满，那么应用程序的 PUT 操作会失败。

如果此情况在暂存或目标节点中发生，那么 MCA 可通过以下三种方式应对此情况：

1. 调用已定义的消息重试出口。
2. 将所有溢出消息引导至死信队列 (DLQ)，并将异常报告返回给请求这些报告的应用程序。
注：在分布式排队管理中，如果消息对于 DLQ 过大、DLQ 已满或 DLQ 不可用，那么通道会停止，且消息保留在传输队列中。确保您的 DLQ 已定义、可用并且其大小适合于您处理的最大消息。
3. 在前述选项皆不成功的情况下，关闭通道。
4. 将未送达的消息返回至发送端，并将完整报告返回至应答队列（MQRC_EXCEPTION_WITH_FULL_DATA 和 MQRO_DISCARD_MSG）。

如果 MCA 无法在 DLQ 中放置消息：

- 通道停止
- 消息通道两端的系统控制台会发出相应的错误消息。
- 工作单元被退回，消息重新出现在位于通道的发送通道端的传输队列中。
- 触发对于传输队列处于禁用状态

使用 dspmqrte 监视消息

如果消息没有到达其预期目标，那么可以使用 IBM MQ 显示路径应用程序（可通过控制命令 **dspmqrte** 来使用），以确定消息通过队列管理器网络及其最终位置所采用的路径。

可以使用 IBM MQ display route application (**dspmqrte**) 命令通过命令行界面来处理与跟踪路由消息相关的跟踪路由消息和活动信息。

IBM MQ 显示路由应用程序 (**dspmqrte**) 命令可以在除 z/OS 以外的所有平台上运行。通过在发出 **dspmqrte** 命令时指定 **-c** 参数，可以将 IBM MQ 显示路由应用程序作为客户机运行到 IBM MQ for z/OS 队列管理器。

有关更多信息，请参阅 [IBM MQ 显示路由应用程序](#) 和 [dspmqrte \(显示路由信息\)](#)。

使用 Ping 测试通信

Ping 可用于确定通信链路和构成一个消息通道的两个消息通道代理是否在所有界面间工作正常。

关于此任务

Ping 不使用传输队列，但它确实会调用一些用户出口程序。如遇任何错误情况，将发出错误消息。

过程

- 使用 MQSC 命令 [PING CHANNEL](#) 通过将数据作为特殊消息发送到远程队列管理器并检查是否返回了数据来测试通道。
本地队列管理器生成此数据。

 在 z/OS 和 IBM i 上，您还可以使用面板界面来选择此选项。

-  在多平台上，使用 MQSC 命令 [PING QMGR](#) 来测试队列管理器是否响应命令。

相关概念

[使用 Ping 检查链接](#)

对通道拒绝运行的问题进行故障诊断

如果通道拒绝运行，那么有许多潜在原因，例如 DMQ 和通道未正确设置，或者通道处于不确定状态。

关于此任务

通道拒绝运行的其他原因可能是，没有任何一端能够在 ASCII 和 EBCDIC 以及整数格式之间执行必要的消息描述符数据转换。在这种情况下，无法进行通信。

过程

1. 检查 DQM 和通道已正确设置。

如果通道从未运行，那么这可能是问题的根源。原因可能是：

- 发送和接收通道之间的名称不匹配 (请记住，大写字母和小写字母很重要)。
- 指定了不正确的通道类型。
- 序号队列 (如果适用) 不可用或已损坏。
- 死信队列不可用。
- 两个通道定义上的序号合并值不同。
- 队列管理器或通信链路不可用。
- 接收方通道可能处于 STOPPED 状态。
- 可能未正确定义连接。
- 通信软件可能存在问题 (例如，TCP 是否正在运行?)。

有关设置通道的更多信息，请参阅 [配置分布式队列](#)。

2. 检查通道是否处于不确定状态。

如果启动时自动同步由于某种原因而失败，那么可能存在不确定情况。这由系统控制台上的消息指示，并且状态面板可能用于显示有疑问的通道。如果通道处于不确定状态，那么通常在重新启动时自动解析该通道，因此在正常情况下不需要手动解析通道。但是，您可以在必要时手动再同步通道。有关更多信息，请参阅 [处理不确定通道](#)。

需要手动再同步通道的情况的可能响应如下：

- 发出 **RESOLVE CHANNEL** 命令以回退或落实不确定消息。

要确定是否需要回退或落实，请与远程链接主管一起检查以确定上次落实的工作单元标识 (LUWID) 的编号，然后对照链接末尾的最后一个编号来检查此编号。如果远程端已落实一个数字，并且该数字尚未在链路结束时落实，请使用 **RESOLVE CHANNEL** 命令来落实消息。在所有其他情况下，请使用 **RESOLVE CHANNEL** 命令来回退消息。有关更多信息，请参阅 [处理不确定通道](#)。

这些命令的效果是已回退的消息会重新出现在传输队列中并再次进行发送，同时会丢弃已提交的消息。

如果怀疑自己，那么可能以复制已发送消息的可能性来回退可能是更安全的决策。

- 发出 **RESET CHANNEL** 命令。

此命令旨在序列编号生效时使用，并且应当谨慎使用。其目的是重置消息的序号，并且必须仅在使用 **RESOLVE CHANNEL** 命令来解决任何不确定情况之后才使用该序号。

使用顺序编号且发送方通道在重置后启动时，发送方通道会采取两项操作：

- 它会通知接收方通道，自己已重置。
- 它会指定将同时由发送方通道和接收方通道使用的下一个消息序号。

3. 如果通道的接收方端的状态为 STOPPED，请通过启动接收方端将其复位。

注：这并不会启动通道，而仅仅只是重置其状态。通道仍然必须从发送端启动。

相关参考

[RESOLVE CHANNEL \(请求通道解决不确定消息\)](#)

[RESET CHANNEL \(通道的复位消息序号\)](#)


对触发通道进行故障诊断

如果触发的通道拒绝运行，那么该通道可能处于不确定状态。另一种可能是通道已将传输队列上的触发器控制参数设置为 NOTRIGGER。

关于此任务

有关触发的通道无法启动的情况示例如下：

1. 使用触发器类型 FIRST 定义了传输队列。
2. 消息到达传输队列，并生成了触发器消息。
3. 通道已启动，但由于无法与远程系统通信而立即停止。
4. 远程系统变成可用状态。
5. 其他消息到达传输队列
6. 第二个消息未将队列深度从 0 增加到 1，因此没有生成任何触发器消息（除非通道处于重试状态）。如果发生此情况，请手动重新启动通道。

 在 z/OS 上，如果在通道启动程序关闭期间使用 **MODE(FORCE)** 停止队列管理器，那么可能需要在通道启动程序重新启动后手动重新启动某些通道。

过程

1. 检查通道是否处于不确定状态。

如果触发的通道拒绝运行，请调查是否可能出现 [第 46 页的『对通道拒绝运行的问题进行故障诊断』](#) 的步骤 [第 46 页的『2』](#) 中所述的不确定消息。

2. 检查通道是否已将传输队列上的触发器控制参数设置为 NOTRIGGER。

以下情况下会发生此问题：

- 存在通道错误。
- 通道因接收方请求已停止。
- 通道因需要手动干预的发送方问题已停止。

3. 在诊断和修复此问题之后，手动启动通道。

对网络问题进行故障诊断

如果遇到网络问题，那么存在一些需要检查的内容。

过程

- 使用 LU 6.2 时，请确保定义在整个网络内保持一致。
例如，如果在 CICS Transaction Server for z/OS 或 Communications Manager 定义中增加了 RU 大小，但在其定义中有一个具有较小 **MAXDATA** 值的控制器，那么如果尝试通过网络发送大型消息，那么会话可能会失败。此问题的症状可能是成功进行通道协商，但链接在传输消息时失败。
- 使用 TCP 时，如果通道不可靠且连接中断，请尝试为系统或通道设置 **KEEPALIVE** 值。
您可以使用 **SO_KEEPLIVE** 选项来设置系统范围的值。

z/OS 在 z/OS 上，您还有以下选项：

- 使用 "保持活动时间间隔" 通道属性 (**KAINT**) 来设置特定于通道的保持活动值。
- 使用 **RCVTIME** 和 **RCVTMIN** 通道启动程序参数。

有关更多信息，请参阅 [检查通道的另一端是否仍然可用和 保持活动时间间隔 \(KAINT\)](#)。

注：启动组 TCP/IP 侦听器时，将向 DDNS 注册。但在地址可用于网络之前，可能会有延迟。在此期间启动且以新注册的通用名称为目标的通道将发生故障，显示**通信配置错误**消息。在名称可供网络使用之前，将重试此通道。延迟长度取决于所使用的名称服务器配置。

- 如果在通道失去联系后，接收方通道处于 "通信接收" 状态，请检查是否需要用户干预来解决问题。
如果通道失去联系，那么接收方通道可以处于 "通信接收" 状态。重新建立通信时，发送方通道会尝试重新连接。如果远程队列管理器发现接收方通道已在运行，那么它不允许启动同一接收方通道的另一个版本。此问题需要用户干预以纠正问题或使用系统保持活动。

"采用 MCA" 功能可自动解决问题。它使 IBM MQ 能够取消接收方通道并在其位置启动新的接收方通道。

相关概念

[监视 IBM MQ 网络](#)

z/OS 通道故障，返回码为 **ECONNRESET for TCP/IP**

存在通道故障，并且在 z/OS 上接收到以下内容：CSQX208E TRPTYPE=TCP RC=00000461 或 CSQX208E TRPTYPE=TCP RC=00000461 reason=76650446。

原因

根据企业使用的一个或多个平台，当同级重置连接 (ECONNRESET) 时，您将收到以下返回码：

AIX

ECONNRESET 73 (hexadecimalm49)

Linux

ECONNRESET 104 (十六进制 68)

Windows

WSAECONNRESET 10054 (十六进制 2746)

z/OS

10054 或 RC461

此返回码通常是 TCP/IP 网络中问题的结果。TCP/IP 发送重置有各种原因：

- 无序连接终止 (例如，重新引导客户机框) 可能会导致重置。
- 应用程序请求连接到没有服务器侦听的端口和 IP 地址。
- 应用程序关闭套接字，而数据仍在应用程序接收缓冲区中。将重置连接以允许远程合作伙伴知道未传递数据。
- 对于已关闭的连接，到达的任何数据都可能导致重置。

- 应用程序关闭套接字并将 `linger` 套接字选项设置为零。这将通知 TCP/IP 连接不应该存在。
注: IBM MQ 不会对 `linger` 时间 = 0 进行编码, 因此 IBM MQ 本身不会导致重置。
- 对于连接, 无效的 TCP 数据段到达。例如, 错误的应答或序号可能导致重置。
- 连接请求超时。TCP 停止尝试连接到特定端口和 IP 地址并重置连接。
- 如果包不符合防火墙规则和策略, 那么防火墙可以重置连接。例如, 源端口或目标端口或 IP 地址与防火墙规则或策略不匹配。
- 重新传输计时器到期。TCP 停止尝试重新传输包并重置连接。
- 错误的硬件设备可能导致重置。

您需要注意的是, 配置在更高级别的影响 (例如, 通道启动程序分派优先级过低) 可能表现为重置。因此, 在尝试确定发生重置的原因时, 您还应该考虑配置的影响。

诊断问题

使用 [TCP/IP 包跟踪](#) 来确定发生重置的原因。

请参阅 [z/OS UNIX 原因码](#), 以获取在 CSQX208E 错误消息中找到的原因码的最后两个字节。

重试链接的注意事项

如果在正常操作期间发生链路故障, 那么发送方或服务器通道程序本身将启动另一个实例, 前提是满足某些条件。其他错误场景可能更难进行故障诊断, 需要进一步的手动调查。

正常操作期间链路故障

如果在正常操作期间发生链接故障, 发送方或服务器通道程序将自行启动其他实例, 前提是:

1. 初始数据协商和安全交换已完成
2. 通道定义中的重试计数大于零

注: 对于多平台, 要尝试重试, 通道启动程序必须正在运行。对于 IBM MQ for z/OS, 此通道启动程序必须正在监视通道所使用的传输队列中指定的启动队列。

难以识别错误场景

可能发生难以识别的错误场景。例如, 链路和通道可能正常运行, 但在接收端发生的某些情况会导致接收方停止。其他无法预见的情况包括, 接收方系统已耗尽内存并且无法完成事务。

需要注意的是, 可能出现这样的情况, 其特征通常表现为系统看起来很繁忙, 但实际上并没有移动消息。您需要与位于链接最末端的同行合作, 以帮助检测并更正问题。

z/OS 中的共享通道恢复

共享通道恢复是在 IBM MQ for z/OS 上使用队列共享组的优点之一。

下表显示了共享通道故障的类型以及每种类型的处理方式:

| 故障类型 | 发生了什么 |
|---------------|--|
| 通道启动程序通信子系统故障 | 依赖于通信子系统的通道开始通道重试, 并由负载平衡启动命令在适当的队列共享组通道启动程序中重新启动。 |
| 通道启动程序故障 | 通道启动程序失败, 但是关联的队列管理器仍保持活动状态。队列管理器会监控失败情况并启动恢复处理。 |
| 队列管理器故障 | 队列管理器失败 (使关联的通道启动程序失败)。队列共享组中的其他队列管理器监控事件并启动对等恢复。 |

| 表 1: 处理共享通道故障 (继续) | |
|--------------------|--|
| 故障类型 | 发生了什么 |
| 共享状态故障 | 通道状态信息存储在 Db2 中，因此通道状态发生更改时，如果指向 Db2 的连接丢失，那么会发生故障。运行中的通道可在不访问这些资源的情况下继续运行。在对 Db2 的访问失败后，通道开始进行重试。 |

代表故障系统进行共享通道恢复处理要求到 Db2 的连接在管理恢复的系统上可用，以便检索共享通道状态。

相关概念

使用队列共享组为 DQM 准备 IBM MQ for z/OS

解决通道停止运行的问题

通道停止运行问题的两种可能的解决方案是通道切换和连接切换。

关于此任务

通道停止运行问题的两种可能的解决方案是：

通道切换

对于通道切换，为同一传输队列定义了两个消息通道，但具有不同的通信链路。首选一个消息通道，另一个消息通道是在首选通道不可用时使用的替换项。

注：如果这些消息通道需要触发，那么对于每个发送方通道端必须存在关联的进程定义。

连接切换

另一个解决方案是从传输队列切换通信连接。

过程

- 要切换消息通道，请执行以下操作：
 - 如果触发通道，请设置传输队列属性 **NOTRIGGER**。
 - 确保当前通道处于不活动状态。
 - 解析当前通道上的任何不确定消息。
 - 如果触发了通道，请更改传输队列中的进程属性以命名与替换通道关联的进程。
在此上下文中，某些实施允许通道具有空白进程对象定义，在此情况下可以省略此步骤，因为队列管理器将查找并启动相应的进程对象。
 - 重新启动通道，或者如果已触发通道，请设置传输队列属性 **TRIGGER**。
- 要从传输队列切换通信连接，请执行以下操作：
 - 如果触发了发送方通道，请设置传输队列属性 **NOTRIGGER**。
 - 确保通道处于不活动状态。
 - 更改连接字段和概要文件字段以连接到替换通信链路。
 - 确保已定义远程端的对应通道。
 - 重新启动通道，或者如果触发了发送方通道，请设置传输队列属性 **TRIGGER**。

灾难恢复

灾难恢复规划由个别安装负责，所执行的功能可能包括提供安全存储在非现场的常规系统“快照”转储。

这些堆可用于重新生成系统，以防灾难突然降临。如果发生灾难，您需要知道消息会提供怎样的帮助，而后续的描述是为了帮助您开始思考此问题。

首先，系统重新启动时进行重新映射。如果系统由于任何原因而发生故障，那么它可能有一个系统日志，该日志允许通过将系统软件从同步点重放到故障时刻来重新生成在发生故障时运行的应用程序。如果在没有错

误的情况下发生此情况，那么可能发生的最坏情况是，消息通道同步在启动时指向相邻系统可能失败，并且将再次发送各个通道的最后一批消息。持久消息将恢复并再次发送，非持久消息可能会丢失。

如果系统没有用于恢复的系统日志，或者如果系统恢复失败，或者如果调用了灾难恢复过程，那么可能会将通道和传输队列恢复到较早状态，并且在通道发送端和接收端的本地队列上保存的消息可能不一致。

可能丢失了放在本地队列上的消息。此情况的后果取决于特定的 IBM MQ 实现及通道属性。例如，在实施严格消息序列的情况下，接收通道会检测序列号间隔，并且通道会关闭以进行手动干预。然后，恢复取决于应用程序设计，在最坏的情况下，发送应用程序可能需要从较早的消息序号重新启动。

对 IBM MQ Console 和 REST API 问题进行故障诊断

通过查看可用日志来诊断 IBM MQ Console 和 REST API 存在的问题。根据 IBM 工作人员的要求，您可能还需要配置跟踪。

如果遇到 IBM MQ Console 或 REST API 的问题，请执行以下检查：

- 检查 mqweb 服务器的状态。如果 mqweb 服务器已停止，那么无法使用 IBM MQ Console 或 REST API。可以通过使用以下命令来检查该服务器的状态：

```
dspmweb status
```



注意: z/OS V 9.1.0

在 z/OS 上发出 **setmqweb** 或 **dspmweb** 命令之前，必须设置 `WLP_USER_DIR` 环境变量，以便该变量指向 mqweb 服务器配置。

为此，请发出以下命令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，`WLP_user_directory` 是传递到 `crtmqweb` 的目录的名称。例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

有关更多信息，请参阅[创建 mqweb 服务器](#)。

如果 mqweb 服务器已停止，请使用以下命令来启动该服务器：

```
strmqweb
```

在 z/OS 上，检查 mqweb 服务器启动的任务是否正在运行。如果需要，请启动在 [为 mqweb 服务器创建过程中创建的过程](#)。

- 确保必需的 mqweb 配置文件存在：

```
jvm.options
mqwebuser.xml
server.xml
```

使用 `crtmqdir` 命令查找 `MQ_DATA_PATH/web/installations/installationName/servers/mqweb/` 目录中的文件。

要检查包含这些文件的搜索的安装，请使用以下命令：

```
crtmqdir -a
```

如果文件缺失，可使用以下命令重新创建：

```
crtmqdir -s -f
```

- 检查 mqweb 服务器日志文件 `console.log` 和 `messages.log`。可在以下位置中找到这些日志文件：

```
- ULW MQ_DATA_PATH/web/installations/installationName/servers/mqweb/
  logs
```

- **z/OS** 在运行 **crtmqweb** 脚本来创建 mqweb 服务器定义时指定的目录。缺省情况下，此目录为 `/var/mqm/web/installation1/servers/mqweb/logs`。

请注意，这些文件采用 UTF-8 编码。要查看这些文件，可使用以下某种方法：

- 从 UNIX 系统服务命令行中使用 **oedit** 命令。
- 进入 ISPF option 3.17，使用 **va**（查看 ASCII）行命令。

- **z/OS** 在 z/OS 上，检查 mqweb 服务器启动的任务输出中的 STDERR 和 STDOUT。除非发生了错误，否则在 STDERR 中不包含任何消息。
- 如果从除正在运行 mqweb 服务器的系统以外的主机中无法访问 IBM MQ Console 或 REST API，请检查是否已使用 `httpHost` 属性启用远程连接。

发出以下命令以显示 mqweb 服务器配置：

```
dspmqweb properties -a
```

如果 `httpHost` 属性的值为 `localhost`，那么只能从 mqweb 服务器所在的主机中使用 IBM MQ Console 和 REST API。输入以下命令以启用到 mqweb 服务器的远程连接：

```
setmqweb properties -k httpHost -v hostname
```

其中，`hostname` 指定 IP 地址、带有域名后缀的域名服务器 (DNS) 主机名或已安装 IBM MQ 的服务器的 DNS 主机名。使用星号 *（用双引号括起）可指定所有可用的网络接口，如以下示例中所示：

```
setmqweb properties -k httpHost -v "*"
```

- 如果在 IBM MQ Console 中的本地队列管理器窗口小部件中未显示任何队列管理器，请检查在 mqweb 服务器所在的主机上是否存在可由 IBM MQ Console 管理的队列管理器。
 - **ULW** 在 IBM MQ Console 中仅列出 mqweb 服务器所在的安装中的队列管理器。
 - **z/OS** 在 z/OS 中，在 IBM MQ Console 中仅列出自从上次 IPL 后启动的、版本与 mqweb 服务器相同的队列管理器。
- **z/OS** 如果仍遇到问题，那么可能未正确配置 mqweb 服务器启动的任务，或者 IBM MQ Unix 系统服务 Web 组件安装文件可能存在问题。

在 IBM MQ Console 中，您可能会看到以下消息：

```
Lost communication with the server  
Could not establish communication with the server.
```

在用于启动 mqweb 服务器的过程中：

1. 检查 STEPLIB 库的级别是否正确以及是否已获得 APF 授权。
2. 检查 INSTDIR、USERDIR、PATH 和 LIBPATH 指向的路径是否正确。

在 UNIX 系统服务中，输入以下命令：

```
ls -Eltr PathPrefix/web/bin/dspmq
```

其中，`PathPrefix` 是 IBM MQ UNIX 系统服务组件的安装路径。

这将显示如下输出：

```
-rwxr-xr-t a-s- ... /mqm/V9R1M0/web/bin/dspmq
```

检查是否已设置 `t` 和 `a` 标志。如果需要，请使用以下命令：

- `chmod +t PathPrefix/web/bin/dspmq`：用于设置粘性位 (`t`)
- `extattr +a PathPrefix/web/bin/dspmq`：用于设置 APF 授权属性 (`a`)

有关收集 IBM MQ Console 和 REST API 的跟踪的更多信息，请参阅第 335 页的『跟踪 IBM MQ Console 和 REST API』。

对 IBM MQ Internet Pass-Thru 问题进行故障诊断

您可以执行若干步骤来帮助确定在使用 IBM MQ Internet Pass-Thru (MQIPT) 时可能遇到的任何问题的性质。

1. 检查以下常见错误:

- 在直接连接到队列管理器的路由上，**HTTP** 属性设置为 **true**。
- 在直接连接到队列管理器（其未配置为使用 SSL/TLS）的路由上，**SSLClient** 属性设置为 **true**。
- 针对密钥环文件存储的密码区分大小写。

2. 如果您在 errors 子目录中发现任何 FFST 报告，那么表明 MQIPT 已正确安装，但配置可能有问题。

每个 FFST 都会报告导致 MQIPT 或路由终止其启动过程的问题。修复导致每个 FFST 的问题。然后，删除旧的 FFST 并重新启动或刷新 MQIPT。

3. 如果既没有 FFST 也没有跟踪输出，那么表明未正确安装 MQIPT。检查所有文件是否都位于正确的位置。要对此进行检查，请尝试手动启动 MQIPT:

a. 打开命令提示符。转至 bin 子目录并输入:

```
mqipt xxx
```

其中，*xxx* 是 MQIPT 主目录。

b. 当 MQIPT 启动时，请在主目录中查找配置。在 errors 子目录中查找任何错误消息和 FFST 实例。

c. 查看 MQIPT 的文本输出中是否有任何错误消息。检查 FFST 实例。更正所有错误。

注: 如果配置文件的 [global] 部分中存在问题，那么 MQIPT 将不会启动。如果配置文件的 [route] 部分中有问题，那么路由将不会启动。

4. 如果没有 FFST 但有跟踪输出，请配置 MQIPT 连接 (ConnectionLog=true)，并让发送方尝试建立连接。然后检查是否记录了来自该主机的连接。

- 如果记录了来自该主机的连接，那么表明未正确配置发送方。
- 如果没有记录连接，请检查 MQIPT 是否配置为将消息转发到正确的主机和端口。然后，将其视为常规通道问题。

相关任务

第 227 页的『联系 IBM 支持人员』

如果您在 IBM MQ 方面遇到问题需要帮助，可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订、故障诊断和其他新闻的通知。

相关参考

第 337 页的『跟踪 IBM MQ Internet Pass-Thru 中的错误』

IBM MQ Internet Pass-Thru (MQIPT) 提供了由 **Trace** 属性控制的详细执行跟踪工具。

检查端到端连接

如果无法建立连接，请检查连接日志以查看是否正确设置了路由。

创建连接日志：在 mqipt.conf 配置文件中，将 **ConnectionLog** 属性设置为 **true**。启动或刷新 MQIPT，然后尝试建立连接。请参阅连接日志，以获取详细信息。

1. 如果在主目录的 logs 目录下没有创建连接日志，那么表明未正确安装 MQIPT。
2. 如果没有记录连接尝试，那么表明未正确设置发送方。
3. 如果记录了连接尝试，请检查 MQIPT 是否将消息转发到正确的地址。

自动启动 MQIPT

如果安装 MQIPT 作为 Windows 服务或者作为 UNIX 或 Linux **init.d** 系统服务，那么它将在系统启动时启动。如果该服务没有正常启动，请执行本主题中的步骤。

在 Windows 系统上

在安装 MQIPT 作为 Windows 服务之前，请始终尝试手动启动该服务，以确认正确安装该服务。请参阅 [在 Windows 上自动启动 MQIPT](#) 以获取更多详细信息。

如果 MQIPT 服务没有正常启动，请完成以下步骤：

1. 打开 Windows 注册表编辑器，并浏览至 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\MQInternetPassThru 键。检查 **ConfigFilePath** 设置是否包含 `mqipt.conf` 配置文件的正确路径。另外，检查 **ImagePath** 设置是否包含 `mqiptService.exe` 的正确路径。
2. 从管理员命令提示符中运行 **mqiptService -debugevents** 命令，以在 Windows 应用程序事件日志中写入服务启动信息。命令提示符控制台窗口中还会显示其他信息。请检查诊断信息以确定失败原因。
3. 如果失败原因仍不明确，请使用 Windows 文件资源管理器来浏览至 **ConfigFilePath** 中指定的、`mqipt.conf` 所在的目录。请检查 `errors` 子目录的内容以查找包含 FFST 记录的 FDC 文件。
4. 如果失败原因仍不明确，请通过在 `mqipt.conf` 的 `[global]` 部分中将 **Trace** 属性设置为 5 来启用跟踪。重新启动 MQIPT 服务。系统会将一个跟踪文件写入到 MQIPT 的 `errors` 目录中。如有必要，请联系 IBM 软件支持代表，并提供跟踪文件、所有 FDC 文件以及来自 **mqiptService -debugevents** 命令的诊断输出。

在 UNIX and Linux 系统上

在安装 MQIPT 作为服务之前，请始终尝试手动启动该服务，以确认正确安装该服务。请参阅 [在 UNIX 或 Linux 上自动启动 MQIPT](#) 以获取更多详细信息。

如果 MQIPT 服务没有正常启动，请以 root 用户身份完成以下步骤：

1. 检查是否已安装 MQIPT 服务。您可能需要卸载并重新安装此服务。要检查是否已安装此服务，请执行以下操作：
 - 在 AIX 上，运行命令 **lsitab mqipt** 并检查输出是否显示正确的安装目录。以下是通过 `/usr/opt/mqipt` 安装运行的 MQIPT 服务的输出示例：

```
mqipt:2:once:/usr/opt/mqipt/bin/mqipt /usr/opt/mqipt > /dev/console 2>&1
```

检查指定的 MQIPT 可执行文件是否存在以及是否可由 root 用户执行。

- 在 Linux 上，检查是否存在名为 `/etc/init.d/mqipt` 的 MQIPT **init.d** 脚本。该脚本必须存在且必须可由 root 用户执行。
2. 确保安装目录中包含必须可由 root 用户读取的 `mqipt.conf` 文件。
 3. 检查 MQIPT 启动的输出。
 - 在 AIX 上，MQIPT 输出将发送到 `/dev/console`。
 - 在 Linux 上，输出将发送到 MQIPT 安装的 `logs` 目录中名为 `console.log` 的文件。

查找任何 MQIPT 错误并确定原因。如果不存在控制台输出，那么 MQIPT 不是由操作系统启动的。请参考操作系统文档，以获取有关如何诊断服务启动失败的详细信息。

4. 如果故障原因仍不清楚，请浏览至 `mqipt.conf` 所在的 MQIPT 安装目录。请检查 `errors` 子目录的内容以查找包含 FFST 记录的 FDC 文件。
5. 如果失败原因仍不明确，请通过在 `mqipt.conf` 的 `[global]` 部分中将 **Trace** 属性设置为 5 来启用跟踪。重新启动 MQIPT 服务。系统会将一个跟踪文件写入到 MQIPT 的 `errors` 目录中。如有必要，请联系 IBM 软件支持，并提供跟踪文件以及任何 FDC 文件和来自 `/dev/console`（在 AIX 上）或 `console.log`（在 Linux 上）的诊断输出。

使用 JRE 诊断选项

在某些情况下，您可能需要使用 Java runtime environment (JRE) 内置的诊断功能。通常，您只应在 IBM 软件支持代表的指导下完成此操作，因为某些诊断设置可能会影响 MQIPT 的正常运行。

可在命令行中使用 **MQIPT_JVM_OPTIONS** 环境变量来将诊断选项传递至底层的 MQIPT JRE。可以使用对 MQIPT 随附的 IBM JRE 有效的所有命令参数。

可以使用以下两个常用诊断选项：

-Djavax.net.debug=all

此选项支持对 SSL/TLS 和网络吞吐量执行的诊断。设置此选项会导致将内部网络操作的详细日志写入到启动 MQIPT 的控制台中。这对于调试 **SSLClient** 或 **SSLServer** 设置为 true 的路由上的 SSL/TLS 握手错误特别有用。

-Djava.security.debug=access, failure

对于 **SecurityManager** 设置为 true 的 MQIPT 实例，此选项将对 Java security manager 策略启用诊断。设置此选项会导致将安全活动及其所需许可权的详细日志写入到启动 MQIPT 的控制台中。它可用于确定策略文件中缺少的许可权。

下面是在 Windows 平台上同时启用这些设置的示例：

```
set MQIPT_JVM_OPTIONS=-Djavax.net.debug=all -Djava.security.debug=access,failure
```

下面是在 UNIX and Linux 平台上同时启用这些设置的示例：

```
MQIPT_JVM_OPTIONS="-Djavax.net.debug=all -Djava.security.debug=access,failure"  
export MQIPT_JVM_OPTIONS
```

要使这些设置生效，必须从命令提示符中重新启动设置了该环境变量的 MQIPT。

有关 **MQIPT_JVM_OPTIONS** 在诊断问题时的其他用途，请参阅第 337 页的『在 [mqiptKeyman](#) 和 [mqiptKeycmd](#) 中跟踪错误』。

对 IBM MQ MQI client 问题进行故障诊断

此主题集包含有关解决 IBM MQ MQI client 应用程序中的问题的方法的信息。

IBM MQ MQI client 环境中运行的应用程序会以与 IBM MQ 服务器应用程序相同的方式接收 MQRC_* 原因码。然而，对于与 IBM MQ MQI clients 关联的错误情况，还有其他几个原因码。例如：

- 远程机器未响应
- 通信线路错误
- 无效的机器地址

通常在应用程序发出 MQCONN 或 MQCONNX 和接收响应 MQRC_Q_MQR_NOT_AVAILABLE 时最容易发生错误。在客户机错误日志中查找解释故障的消息。另外，错误也可能记录在服务器上，这取决于故障的性质。同样，还要检查 IBM MQ MQI client 上的应用程序是否与正确的库文件相链接。

IBM MQ MQI client 无法进行连接

MQCONN 或 MQCONNX 可能会失败，因为服务器上或在协议检查期间未运行任何侦听器程序。

当 IBM MQ MQI client 向服务器发出 MQCONN 或 MQCONNX 调用时，将在 IBM MQ MQI client 与服务器之间交换套接字和端口信息。要进行任何消息交换，用于在通信线路上侦听任何活动的服务器上必须有程序。如果无程序完成此操作，或者有程序但未进行正确配置，那么 MQCONN 或 MQCONNX 调用将失败，并向 IBM MQ MQI client 应用程序返回相关原因码。

如果成功连接，将交换 IBM MQ 协议消息，并进行进一步的检查。IBM MQ 协议检查阶段，将协调某些方面，而其他方面可能会使连接失败。所有这些检查成功后，MQCONN 或 MQCONNX 调用才会成功。

有关 MQRC_* 原因码的信息，请参阅 [API 完成代码和原因码](#)。

正在停止 IBM MQ MQI clients

即使 IBM MQ MQI client 已停止，服务器的关联进程也仍能将其队列保持打开状态。只有通信层检测到伙伴已离开后，才会关闭队列。

如果已启用共享对话，那么服务器通道始终保持正确状态，以便通信层能够检测到伙伴已离开这一情况。

IBM MQ MQI clients 的错误消息

IBM MQ MQI client 系统出错时，错误消息将被放入 IBM MQ 系统错误文件中。

- 在 UNIX and Linux 系统上，这些文件位于 `/var/mqm/errors` 目录中
- 在 Windows 上，这些文件位于 IBM MQ MQI client 安装的错误子目录中。通常情况下，此目录为 `C:\Program Files\IBM\MQ\errors`。
- 在 IBM i 上，这些文件位于 `/QIBM/UserData/mqm/errors` 目录中

还可以在与客户机连接的服务器关联的 IBM MQ 错误文件中记录特定的客户机错误。

IBM MQ .NET 问题故障诊断

您可以使用 .NET 样本应用程序来帮助您进行问题故障诊断。

使用样本应用程序

如果程序未成功完成，那么运行其中一个 .NET 样本应用程序，然后按照诊断消息中提供的建议操作。[.NET 的样本应用程序](#)中描述了这些样本应用程序。

如果问题继续存在，您需要联系 IBM 服务团队，他们可能会让您启用跟踪功能。有关使用跟踪功能的信息，请参阅第 338 页的[『跟踪 IBM MQ.NET 应用程序』](#)。

错误消息

您可能会看到以下常见错误消息：

在未知模块中出现 `System.IO.FileNotFoundException` 类型的未处理异常

如果针对 `amqmdnet.dll` 或 `amqmdxcs.dll` 出现此错误，请确保两者已在全局组合件高速缓存中注册，或者创建指向 `amqmdnet.dll` 和 `amqmdxcs.dll` 组合件的配置文件。您可以使用 `mscorcfg.msc`（在 .NET 框架中提供）来检查并更改组合件高速缓存的内容。

如果 .NET 框架在安装 IBM MQ 时不可用，那么将不会在全局组合件高速缓存中注册这些类。您可以使用此命令手动重新运行注册过程

```
amqidnet -c MQ_INSTALLATION_PATH\bin\amqidotn.txt -l logfile.txt
```

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

有关此安装的信息写入在指定的日志文件中（在本示例中，为 `logfile.txt`）。

对 Java 和 JMS 问题进行故障诊断

使用此处给出的建议来帮助解决在使用 Java 或 JMS 应用程序时可能发生的常见问题。

相关概念

第 339 页的[『跟踪 IBM MQ classes for JMS 应用程序』](#)

[IBM MQ classes for JMS](#) 中提供了跟踪功能来帮助 IBM 支持人员诊断客户问题。各种属性均可控制此功能的行为。

第 346 页的[『跟踪 IBM MQ 资源适配器』](#)

`ResourceAdapter` 对象封装 IBM MQ 资源适配器的全局属性。要启用 IBM MQ 资源适配器跟踪，需要在 `ResourceAdapter` 对象中定义属性。

第 347 页的[『跟踪其他 IBM MQ Java 组件』](#)

对于 IBM MQ 的 Java 组件，例如 IBM MQ Explorer 和 IBM MQ Transport for SOAP 的 Java 实现，诊断信息是使用标准 IBM MQ 诊断工具或由 Java 诊断类输出的。

相关任务

第 342 页的『跟踪 IBM MQ classes for Java 应用程序』

IBM MQ classes for Java 中提供了跟踪功能来帮助 IBM 支持人员诊断客户问题。各种属性均可控制此功能的行为。

[使用 IBM MQ classes for JMS](#)

[使用 IBM MQ 资源适配器](#)

[使用 IBM MQ classes for Java](#)

使用 IBM MQ classes for JMS 解决问题

您可以运行安装验证程序和使用跟踪和日志功能来研究问题。

如果程序未成功完成，请运行其中一个安装验证程序，如 [The pointto point IVT for IBM MQ classes for JMS](#) 和 [The publish/subscribe IVT for IBM MQ classes for JMS](#) 中所述，并遵循诊断消息中提供的建议。

相关概念

第 339 页的『跟踪 IBM MQ classes for JMS 应用程序』

IBM MQ classes for JMS 中提供了跟踪功能来帮助 IBM 支持人员诊断客户问题。各种属性均可控制此功能的行为。

IBM MQ classes for JMS 的日志记录错误

缺省情况下，日志输出将发送到 `mqjms.log` 文件。您可以将其重定向到特定的文件或目录。

系统提供 IBM MQ classes for JMS 日志功能来报告严重的问题，尤其是可能指出配置错误（而非编程错误）的问题。缺省情况下，日志输出将发送到 JVM 工作目录中的 `mqjms.log` 文件。

您可以通过设置 `com.ibm.msg.client.commonservices.log.outputName` 属性，将日志输出重定向到另一个文件。此属性的值可以是：

- 单一路径名。
- 路径名的逗号分隔列表（所有数据均记录到所有文件中）。

每一个路径名都可以是：

- 绝对的或相对的。
- `stderr` 或 `System.err` 表示标准错误流。
- `stdout` 或 `System.out` 表示标准输出流。

如果该属性的值标识一个目录，那么日志输出将写入到该目录中的 `mqjms.log` 中。如果属性值可识别特定的文件，那么日志输出将写入该文件。

可以在 IBM MQ classes for JMS 配置文件中设置此属性，或在 `java` 命令上将此属性设置为系统属性。在以下示例中，属性将设置为系统属性，并用于识别特定的文件：

```
java -Djava.library.path=library_path
-Dcom.ibm.msg.client.commonservices.log.outputName=/mydir/mylog.txt
MyAppClass
```

在该命令中，`library_path` 是包含 IBM MQ classes for JMS 库的目录的路径（请参阅 [配置 Java 本机接口 \(JNI\) 库](#)）。

可通过将 `com.ibm.msg.client.commonservices.log.status` 属性设置为 `OFF`，来禁用日志输出。此属性的缺省值为 `ON`。

可以设置值 `System.err` 和 `System.out` 以将日志输出发送到 `System.err` 和 `System.out` 流。

JMS 提供程序版本故障诊断

使用此处提供的建议来帮助解决在使用指定提供程序版本连接到队列管理器时可能发生的常见问题。

“此连接不支持 JMS 2.0 功能”错误

- **错误代码:** JMSSC5008
- **场景:** 您收到此连接不支持 JMS 2.0 功能错误。
- **说明:** 仅当连接到使用 IBM MQ 消息传递提供程序版本 8 方式的 IBM MQ 8.0 或更高版本队列管理器时, 才支持使用 JMS 2.0 功能。
- **解决方案:** 将应用程序更改为不使用 JMS 2.0 功能, 或者确保应用程序连接到使用 IBM MQ 消息传递提供程序版本 8 方式的 IBM MQ 8.0 队列管理器。

“此连接不支持 JMS 2.0 API”错误

- **错误代码:** JMSSC5007
- **场景:** 您收到此连接不支持 JMS 2.0 API 错误。
- **说明:** 仅当您连接到使用 IBM MQ 消息传递提供程序正常或版本 8 方式的 IBM WebSphere MQ 7 或 IBM MQ 8 队列管理器时, 才支持使用 JMS 2.0 API。例如, 如果要尝试连接到 IBM WebSphere MQ 6 队列管理器或者要使用迁移方式进行连接, 那么可能会收到此错误。如果指定 SHARECNV(0) 或 PROVIDER_VERSION=6, 那么通常会发生此情况。
- **解决方案:** 将应用程序更改为不使用 JMS 2.0 API, 或者确保应用程序使用 IBM MQ 消息传递提供程序正常或版本 8 方式连接到 IBM WebSphere MQ 7 或 IBM MQ 8 队列管理器。

“队列管理器命令级别与请求的提供程序版本不匹配”错误

- **错误代码:** JMSFMQ0003
- **场景:** 您收到队列管理器命令级别与请求的提供程序版本不匹配错误。
- **说明:** 在连接工厂上的“提供程序版本”属性中指定的队列管理器版本与请求的队列管理器不兼容。例如, 您可能已指定 PROVIDER_VERSION=8, 并尝试以小于 800 的命令级别 (例如 750) 连接到队列管理器。
- **解决方案:** 修改连接工厂以连接到可以支持所需提供程序版本的队列管理器。

有关提供程序版本的更多信息, 请参阅[配置 JMS PROVIDERVERSION 属性](#)。

JMS 中的 PCF 处理

IBM MQ 可编程更改格式 (PCF) 消息是查询和修改队列管理器属性的一种灵活且强大的方式, 而 IBM MQ classes for Java 中提供的 PCF 类可提供在 Java 应用程序中访问其功能的便利方式。功能也可以从 IBM MQ classes for JMS 进行访问, 但是存在潜在问题。

用于处理 JMS 中的 PCF 响应的公共模型

处理 JMS 中 PCF 响应的常用方法是抽取消息的字节有效内容, 将其包装在 `DataInputStream` 中并传递到 `com.ibm.mq.headers.pcf.PCFMessage` 构造函数。

```
Message m = consumer.receive(10000);
//Reconstitute the PCF response.
ByteArrayInputStream bais =
    new ByteArrayInputStream(((BytesMessage)m).getBody(byte[].class));
DataInput di = new DataInputStream(bais);
PCFMessage pcfResponseMessage = new PCFMessage(di);
```

请参阅[使用 IBM MQ 头软件包](#), 获取一些示例。

很遗憾, 此方法并非对所有平台都完全可靠 - 通常该方法适用于大端格式平台, 但不适用于小端格式平台。

问题

问题在于解析消息头过程中, `PCFMessage` 类必须处理数字编码的问题 - 头包含某些大端格式或小端格式编码中的长度字段。

如果将纯 `DataInputStream` 传递到构造函数，那么 `PCFMessage` 类不会明确指示编码，并且必须假设缺省值（很可能不正确）。

如果发生此情况，那么在构造函数中可能显示“MQRCCF_STRUCTURE_TYPE_ERROR”（原因码 3013）：

```
com.ibm.mq.headers.MQDataException: MQJE001: Completion Code '2', Reason '3013'.
  at com.ibm.mq.headers.pcf.PCFParameter.nextParameter(PCFParameter.java:167)
  at com.ibm.mq.headers.pcf.PCFMessage.initialize(PCFMessage.java:854)
  at com.ibm.mq.headers.pcf.PCFMessage.<init>(PCFMessage.java:156)
```

此消息几乎总是意味着已曲解编码。此情况的可能原因是，已读取的已被解释为大端格式数据其实是小端格式数据。

解决方案

避免此问题的方法是向 `PCFMessage` 构造函数传递内容以告知构造函数其正在处理的数据的数字编码。

为此，请从收到的数据执行 `MQMessage`。

以下代码是可能使用的代码的概要示例。



注意：该代码只是概要示例，不包含任何错误处理信息。

```
// get a response into a JMS Message
Message receivedMessage = consumer.receive(10000);
BytesMessage bytesMessage = (BytesMessage) receivedMessage;
byte[] bytesreceived = new byte[(int) bytesMessage.getBodyLength()];
bytesMessage.readBytes(bytesreceived);

// convert to MQMessage then to PCFMessage
MQMessage mqMsg = new MQMessage();
mqMsg.write(bytesreceived);
mqMsg.encoding = receivedMessage.getIntProperty("JMS_IBM_Encoding");
mqMsg.format = receivedMessage.getStringProperty("JMS_IBM_Format");
mqMsg.seek(0);

PCFMessage pcfMsg = new PCFMessage(mqMsg);
```

JMS 连接池错误处理

通过清除策略的各种方法来执行连接池错误处理。

如果在应用程序使用与 JMS 提供程序的 JMS 连接时检测到错误，那么连接池清除策略即生效。连接管理器可以执行以下任一操作：

- 仅关闭遇到问题的连接。这称为 `FailingConnectionOnly` 清除策略，并且是缺省行为。
从工厂创建的任何其他连接（即，正在由其他应用程序使用的连接以及工厂的空闲池中的连接）保持闲置。
- 关闭遇到问题的连接，丢弃工厂的空闲池中的任何连接，并将任何使用中连接标记为旧连接。
使用连接的应用程序下次尝试执行基于连接的操作时，应用程序会接收到 `StaleConnectionException`。对于此行为，请将清除策略设置为 `Entire Pool`。

清除策略 - 仅故障连接

使用 `MDB` 侦听器端口如何使用连接池中描述的示例。两个 `MDB` 部署到应用程序服务器中，每个 `MDB` 使用不同的侦听器端口。侦听器端口均使用 `jms/CF1` 连接工厂。

在 600 秒后，停止第一个侦听器，然后此侦听器端口正在使用的连接返回到连接池。

如果第二个侦听器在轮询 JMS 目标时遇到网络错误，那么侦听器端口会关闭。由于 `jms/CF1` 连接工厂的清除策略设置为 `FailingConnectionOnly`，因此队列管理器仅丢弃第二个侦听器使用的连接。空闲池中的连接保持位置不变。

如果现在重新启动第二个侦听器，那么连接管理器将连接从空闲池传递到侦听器。

清除策略 - 整个池

对于此情况，假设您在应用程序服务器中安装三个 MDB，每个 MDB 使用其自己的侦听器端口。侦听器端口已从 jms/CF1 工厂创建连接。经过一段时间后，停止第一个侦听器，然后将其连接 c1 放入 jms/CF1 空闲池中。

当第二个侦听器检测到网络错误时，它会自行关闭，并且关闭 c2。连接管理器现在关闭空闲池中的连接。但是，正在由第三个侦听器使用的连接将保留。

清除策略的设置目标

如前所述，JMS 连接池的清除策略的缺省值为 `FailingConnectionOnly`。

但是，将清除策略设置为 `EntirePool` 是更好的选项。在大多数情况下，如果应用程序在其与 JMS 提供程序的连接上检测到网络错误，那么从同一连接工厂创建的所有开放连接都可能具有相同问题。

如果清除策略设置为 `FailingConnectionOnly`，那么连接管理器将所有连接都保留在空闲池中。应用程序下次尝试创建与 JMS 提供程序的连接时，如果有可用连接，那么连接管理器会从空闲池中返回连接。但是，当应用程序尝试使用该连接时，它会遇到与第一个应用程序相同的网络问题。

现在，考虑清除策略设置为 `EntirePool` 的相同情况。第一个应用程序遇到网络问题后，连接管理器便会丢弃故障连接并关闭该工厂的空闲池中的所有连接。

在新应用程序启动并尝试从工厂创建连接时，由于空闲池为空，连接管理器会尝试创建新连接。假设已解决网络问题，返回到应用程序的连接有效。

尝试创建 JMS 上下文时发生连接池错误

如果在尝试创建 JMS 上下文时发生错误，那么可根据错误消息来确定是顶级池还是下级池出现问题。

如何将池用于上下文

使用连接和会话时，每种对象类型都有相应的池，而上下文也遵循类似的模型。

使用分布式事务的典型应用程序在同一个事务中会同时涉及消息传递和非消息传递工作负载。

假定当前未在执行任何工作，并且应用程序发出第一个 `createConnection` 方法调用，那么会在连接池（顶级池）的等效项中创建上下文外观或代理。同时会在会话池的等效项中创建另一个对象。第二个对象用于封装底层的 JMS 上下文（下级池）

从概念上讲，池化可允许应用程序进行扩展。多个线程可同时访问有限的资源集。在此示例中，另一个线程将执行 `createContext` 方法调用以从池中获取上下文。如果其他线程仍在执行消息传递工作，那么会扩展顶级池来为请求线程提供另一个上下文。

如果某个线程请求了上下文，并且消息传递工作已完成但非消息传递工作还未完成，因此事务也未完成，那么会扩展下级池。顶级上下文代理会一直分配给该事务直到解析该事务为止，因此它无法分配给其他事务。

如果下级池变满，那么表明非消息传递工作可能需要较长时间才能完成。

如果顶级池变满，那么表明整个消息传递工作需要一会儿才能完成，并且应扩展该池。

识别错误源自哪个池

您可以根据错误消息文本来确定错误源自哪个池：

- 对于顶级池，消息文本为 `Failed to create context`。该消息表明顶级池已由上下文代理对象填满，所有这些对象当前都在运行消息传递事务。
- 对于下级池，消息文本为 `Failed to set up new JMSContext`。该消息表明：虽然连接代理可用，但仍需要等待非消息传递工作完成。

顶级池示例

```
*****[8/19/16 10:10:48:643 UTC] 000000a2
LocalException E CNTR0020E: EJB threw an unexpected (non-declared) exception during
invocation of method "onMessage" on bean
"BeanId(SibSVTLiteMDB#SibSVTLiteMDBXA_RecoveryEJB_undeployed.jar#QueueReceiver, null)".
```

```

Exception data: javax.jms.JMSRuntimeException: Failed to create context
    at com.ibm.ejs.jms.JMSCMUtils.mapToJMSRuntimeException(JMSCMUtils.java:522)
    at
com.ibm.ejs.jms.JMSConnectionFactoryHandle.createContextInternal(JMSConnectionFactoryHandle.java:4
49)
    at
com.ibm.ejs.jms.JMSConnectionFactoryHandle.createContext(JMSConnectionFactoryHandle.java:335)
    at sib.test.svt.lite.mdb.xa.SVTMDBBase.sendReplyMessage(SVTMDBBase.java:554)
    at sib.test.svt.lite.mdb.xa.QueueReceiverBean.onMessage(QueueReceiverBean.java:128)
    at
sib.test.svt.lite.mdb.xa.MDBProxyQueueReceiver_37ea5ce9.onMessage(MDBProxyQueueReceiver_37ea5ce9.j
ava)
    at
com.ibm.mq.connector.inbound.MessageEndpointWrapper.onMessage(MessageEndpointWrapper.java:151)
    at com.ibm.mq.jms.MQSession$FacadeMessageListener.onMessage(MQSession.java:129)
    at com.ibm.msg.client.jms.internal.JmsSessionImpl.run(JmsSessionImpl.java:3236)
    at com.ibm.mq.jms.MQSession.run(MQSession.java:937)
    at com.ibm.mq.connector.inbound.ASFWorkImpl.doDelivery(ASFWorkImpl.java:104)
    at com.ibm.mq.connector.inbound.AbstractWorkImpl.run(AbstractWorkImpl.java:233)
    at com.ibm.ejs.j2c.work.WorkProxy.run(WorkProxy.java:668)
    at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1892)
Caused by: com.ibm.websphere.ce.j2c.ConnectionWaitTimeoutException:
CWTE_NORMAL_J2CA1009
    at com.ibm.ejs.j2c.FreePool.createOrWaitForConnection(FreePool.java:1783)
    at com.ibm.ejs.j2c.PoolManager.reserve(PoolManager.java:3896)
    at com.ibm.ejs.j2c.PoolManager.reserve(PoolManager.java:3116)
    at com.ibm.ejs.j2c.ConnectionManager.allocateMCWrapper(ConnectionManager.java:1548)
    at com.ibm.ejs.j2c.ConnectionManager.allocateConnection(ConnectionManager.java:1031)
    at
com.ibm.ejs.jms.JMSConnectionFactoryHandle.createContextInternal(JMSConnectionFactoryHandle.java:4
43)
    ... 12 more

```

下级池示例

```

*****
[8/19/16 9:44:44:754 UTC] 000000ac SibMessage W   [:] CWSJY0003W: MQJCA4004: Message delivery to
an MDB
'sib.test.svt.lite.mdb.xa.MDBProxyQueueReceiver_37ea5ce9@505d4b68
(BeanId(SibSVTLiteMDB#SibSVTLiteMDBXA_RecoveryEJB_undeployed.jar#QueueReceiver, null))' failed
with exception:
'nested exception is: javax.jms.JMSRuntimeException: Failed to set up new JMSContext'.
^C[root@username-instance-2 server1]# vi SystemOut.log
    :com.ibm.ejs.j2c.work.WorkProxy.run(WorkProxy.java:668)
    : com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1892)
Caused by [1] --> Message : javax.jms.JMSRuntimeException: Failed to set up new
JMSContext
    Class : class javax.jms.JMSRuntimeException
    Stack :
com.ibm.ejs.jms.JMSCMUtils.mapToJMSRuntimeException(JMSCMUtils.java:522)
    :
com.ibm.ejs.jms.JMSContextHandle.setupInternalContext(JMSContextHandle.java:241)
    :
com.ibm.ejs.jms.JMSManagedConnection.getConnection(JMSManagedConnection.java:783)
    :
com.ibm.ejs.j2c.MCWrapper.getConnection(MCWrapper.java:2336)
    :
com.ibm.ejs.j2c.ConnectionManager.allocateConnection(ConnectionManager.java:1064)
    :
com.ibm.ejs.jms.JMSConnectionFactoryHandle.createContextInternal(JMSConnectionFactoryHandle.java:4
43)
    :
com.ibm.ejs.jms.JMSConnectionFactoryHandle.createContext(JMSConnectionFactoryHandle.java:335)
    :
sib.test.svt.lite.mdb.xa.SVTMDBBase.sendReplyMessage(SVTMDBBase.java:554)
    :
sib.test.svt.lite.mdb.xa.QueueReceiverBean.onMessage(QueueReceiverBean.java:128)
    :
sib.test.svt.lite.mdb.xa.MDBProxyQueueReceiver_37ea5ce9.onMessage(MDBProxyQueueReceiver_37ea5ce9.j
ava:-1)
    :
com.ibm.mq.connector.inbound.MessageEndpointWrapper.onMessage(MessageEndpointWrapper.java:151)
    :
com.ibm.mq.jms.MQSession$FacadeMessageListener.onMessage(MQSession.java:129)
    :
com.ibm.msg.client.jms.internal.JmsSessionImpl.run(JmsSessionImpl.java:3236)
    : com.ibm.mq.jms.MQSession.run(MQSession.java:937)
    :
com.ibm.mq.connector.inbound.ASFWorkImpl.doDelivery(ASFWorkImpl.java:104)
    :

```

```

com.ibm.mq.connector.inbound.AbstractWorkImpl.run(AbstractWorkImpl.java:233)
    : com.ibm.ejs.j2c.work.WorkProxy.run(WorkProxy.java:668)
    : com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1892)
Caused by [2] --> Message : com.ibm.websphere.ce.j2c.ConnectionWaitTimeoutException:
CWTE_NORMAL_J2CA1009
    Class : class
com.ibm.websphere.ce.j2c.ConnectionWaitTimeoutException
    Stack : com.ibm.ejs.j2c.FreePool.createOrWaitForConnection(FreePool.java:1783)
    :
com.ibm.ejs.j2c.PoolManager.reserve(PoolManager.java:3840)
    : com.ibm.ejs.j2c.PoolManager.reserve(PoolManager.java:3116)
    :
com.ibm.ejs.j2c.ConnectionManager.allocateMCWrapper(ConnectionManager.java:1548)
    :
com.ibm.ejs.j2c.ConnectionManager.allocateConnection(ConnectionManager.java:1031)
    :
com.ibm.ejs.jms.JMSContextHandle.setupInternalContext(JMSContextHandle.java:222)
    :
com.ibm.ejs.jms.JMSManagedConnection.getConnection(JMSManagedConnection.java:783)
    :
com.ibm.ejs.j2c.MCWrapper.getConnection(MCWrapper.java:2336)
    :
com.ibm.ejs.j2c.ConnectionManager.allocateConnection(ConnectionManager.java:1064)
    :
com.ibm.ejs.jms.JMSConnectionFactoryHandle.createContextInternal(JMSConnectionFactoryHandle.java:443)
    :
com.ibm.ejs.jms.JMSConnectionFactoryHandle.createContext(JMSConnectionFactoryHandle.java:335)
    :
sib.test.svt.light.mdb.xa.SVTMDBBase.sendReplyMessage(SVTMDBBase.java:554)
    :
sib.test.svt.light.mdb.xa.QueueReceiverBean.onMessage(QueueReceiverBean.java:128)
    :
sib.test.svt.light.mdb.xa.MDBProxyQueueReceiver_37ea5ce9.onMessage(MDBProxyQueueReceiver_37ea5ce9.java:-1)
    :
com.ibm.mq.connector.inbound.MessageEndpointWrapper.onMessage(MessageEndpointWrapper.java:151)
    :
com.ibm.mq.jms.MQSession$FacadeMessageListener.onMessage(MQSession.java:129)
    :
com.ibm.msg.client.jms.internal.JmsSessionImpl.run(JmsSessionImpl.java:3236)
    : com.ibm.mq.jms.MQSession.run(MQSession.java:937)
    :
com.ibm.mq.connector.inbound.ASFWorkImpl.doDelivery(ASFWorkImpl.java:104)
    :
com.ibm.mq.connector.inbound.AbstractWorkImpl.run(AbstractWorkImpl.java:233)
    : com.ibm.ejs.j2c.work.WorkProxy.run(WorkProxy.java:668)
    : com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1892)

```

对 JMSSC0108 消息进行故障诊断

在使用以应用程序服务器设施 (ASF) 方式运行的激活规范和 WebSphere Application Server 侦听器端口时，可以执行一些步骤来防止出现 JMSSC0108 消息。

在使用激活规范和 WebSphere Application Server 侦听器端口（以 ASF 方式运行，这是缺省运行方式）时，应用程序服务器日志文件中可能出现以下消息：

```

JMSSC0108: The IBM MQ classes for JMS had detected a message, ready for asynchronous delivery to
an application.
尝试传递时，消息不再可用。

```

使用本主题中的信息，了解出现此消息的原因以及可用于防止出现此消息的可行步骤。

激活规范和侦听器端口如何检测和处理消息

激活规范或 WebSphere Application Server 侦听器端口在启动时会执行以下步骤：

1. 与设为供使用的队列管理器建立连接。
2. 在配置为进行监视的队列管理器上打开 JMS 目标。
3. 浏览此目标以查找消息。

在检测到消息时，激活规范或侦听器端口会执行以下步骤：

1. 构造表示该消息的内部消息引用。
2. 从其内部服务器会话池中获取服务器会话。

3. 通过消息引用装入服务器会话。
4. 通过应用程序服务器工作管理器调度一项工作以运行服务器会话并处理该消息。

然后，激活规范或侦听器端口会返回并再次监视目标，以查找其他待处理的消息。

应用程序服务器工作管理器将运行激活规范或侦听器端口在新的服务器会话线程上提交的工作。该线程在启动时会完成以下操作：

- 根据消息驱动的 bean 是否需要 XA 事务（如消息驱动 bean 的部署描述符中所指定），启动局部或全局 (XA) 事务。
- 通过发出中断性 MQGET API 调用，从目标获取消息。
- 运行消息驱动 bean 的 onMessage() 方法。
- 运行完 onMessage() 方法后，完成局部或全局事务。
- 将服务器会话返还到服务器会话池。

出现 JMSCC0108 消息的原因及其预防方式

主要激活规范或侦听器端口线程将浏览目标上的消息。然后，它会要求工作管理器启动新线程，以中断性方式获取消息并进行处理。这意味着可能出现下列情况：主要激活规范或侦听器端口线程可能在目标上找到某个消息，而在服务器会话线程尝试获取该消息时该消息不再可用。如果发生这种情况，那么服务器会话线程会将以下消息写入应用程序服务器的日志文件：

```
JMSCC0108: The IBM MQ classes for JMS had detected a message, ready for asynchronous delivery to an application.  
尝试传递时，消息不再可用。
```

在服务器会话线程尝试获取消息时，以下两个原因可能导致该消息不再位于目标上：

- [原因 1：另一个应用程序已使用该消息](#)
- [原因 2：该消息已到期](#)

原因 1：另一个应用程序已使用该消息

如果两个或更多个激活规范和/或侦听器端口正在监视同一个目标，那么它们可能检测到同一个消息并进行处理。在发生此情况时：

- 一个激活规范或侦听器端口启动的服务器会话线程会获取该消息并将其传递到消息驱动的 bean 以进行处理。
- 另一个激活规范或侦听器端口启动的服务器会话线程会尝试获取该消息，并发现该消息不再位于目标上。

如果激活规范或侦听器端口使用以下任何方式连接到队列管理器，那么将标记主要激活规范或侦听器端口线程检测到的消息：

- 任何平台上的队列管理器，使用 [IBM MQ 消息传递提供程序正常方式](#)。
- 任何平台上的队列管理器，使用受限制的 [IBM MQ 消息传递提供程序正常方式](#)
- 在 z/OS 上运行的队列管理器，使用 [IBM MQ 消息传递提供程序迁移方式](#)。

标记消息可防止任何其他激活规范或侦听器端口看到此消息并尝试进行处理。

缺省情况下，消息的标记时间为 5 秒。在检测到并标记消息后，5 秒计时器开始计时。在这 5 秒内，必须执行以下步骤：

- 激活规范或侦听器端口必须从服务器会话池中获取服务器会话。
- 必须装入服务器会话以及待处理消息的详细信息。
- 必须调度工作。
- 工作管理器必须处理工作请求并启动服务器会话线程。
- 服务器会话线程需要启动局部或全局事务。
- 服务器会话线程需要以中断性方式获取消息。

在繁忙的系统上，执行这些步骤所用的时间可能不止 5 秒。如果发生此情况，那么将释放消息上的标记。这意味着其他激活规范或侦听器端口现在可以看到消息，并且可以尝试处理该消息，这可能会导致将 JMSCC0108 消息写入应用程序服务器的日志文件。

在这种情况下，您应该考虑以下选项：

- 增大队列管理器属性“消息标记浏览时间间隔 (MARKINT)”的值，为最初检测到该消息的激活规范或侦听器端口提供更多时间来获取该消息。理想情况下，为该属性设置的值应大于消息驱动的 bean 处理消息所用的时间。这意味着，如果主要激活规范或侦听器端口线程因为所有服务器会话都在忙于处理消息而阻止等待服务器会话，那么在服务器会话变为可用时仍应标记消息。请注意，MARKINT 属性是在队列管理器上进行设置，因此适用于在此队列管理器上浏览消息的所有应用程序。
- 增加激活规范或侦听器端口使用的服务器会话池的大小。这意味着有更多的服务器会话可用于处理消息，这应确保在指定的标记时间间隔内可处理完消息。使用此方法时需要注意的是，激活规范或侦听器端口现在能同时处理更多消息，而这会影响应用程序服务器的整体性能。

Multi 如果激活规范或侦听器端口正在使用 IBM MQ 消息传递提供程序迁移方式连接到在 IBM MQ for Multiplatforms 上运行的队列管理器，那么标记功能不可用。这意味着无法防止两个或更多个激活规范和/或侦听器端口检测到同一消息并尝试进行处理。在这种情况下，预期会出现 JMSCC0108 消息。

原因 2：该消息已到期

生成 JMSCC0108 消息的另一个原因是：从激活规范或侦听器端口检测到消息之时起到服务器会话使用该消息这段时间内，该消息已到期。如果发生此情况，那么在服务器会话线程尝试获取消息时，其发现该消息不再存在，因此报告 JMSCC0108 消息。

要解决此问题，请增加激活规范或侦听器端口使用的服务器会话池的大小。增加服务器会话池大小意味着，有更多的服务器会话可用于处理消息，从而有可能在消息到期前处理完消息。值得注意的是，激活规范或侦听器端口现在能同时处理更多消息，而这会影响应用程序服务器的整体性能。

WebSphere Application Server SystemOut.log 文件中的 CWSJY0003W 警告消息

当 MDB 处理来自 IBM MQ 的 JMS 消息时，将在 WebSphere Application Server SystemOut.log 文件中记录 CWSJY0003W 警告消息。

症状

CWSJY0003W: IBM MQ classes for JMS 尝试获取先前通过“带标记浏览”进行标记的消息以便将其传递到消息侦听器，但是该消息不可用。

原因

激活规范和以应用程序服务器工具 (ASF) 方式运行的侦听器端口用于监视 IBM MQ 队列管理器上托管的队列或主题。最初是在队列或主题上浏览消息。当找到消息时将启动一个新线程，该线程以破坏性方式获取该消息，并将该消息传递给消息驱动的 bean 应用程序实例进行处理。

当浏览该消息时，队列管理器将该消息标记了一段时间，因此有效地对其他应用程序服务器实例隐藏了该消息。该消息被标记的时间段由队列管理器属性 MARKINT 确定，缺省情况下该属性设置为 5000 毫秒（5 秒）。这意味着在激活规范或侦听器端口浏览消息之后，队列管理器将等待 5 秒钟以采用破坏性方式获取该消息，之后才允许其他应用程序服务器实例看到该消息并进行处理。

可能发生以下情况：

- 应用程序服务器 1 上运行的激活规范浏览了队列上的消息 A。
- 激活规范启动了一个新线程来处理消息 A。
- 应用程序服务器 1 上发生了某个事件，这意味着 5 秒之后消息 A 仍在该队列上。
- 应用程序服务器 2 上运行的激活规范现在正在浏览消息 A 并启动一个新线程来处理消息 A。
- 应用程序服务器 2 上运行的新线程以破坏性方式获取消息 A，并将其传递给消息驱动的 bean 实例。
- 应用程序服务器 1 上运行的线程尝试获取消息 A，结果却发现消息 A 已不再存在于该队列上。

- 此时，应用程序服务器 A 报告 CWSJY0003W 消息。

解决问题

可以通过以下两种方式来解决此问题：

- 将队列管理器属性 **MARKINT** 的值增大至更大的值。 **MARKINT** 的缺省值是 5000 毫秒（5 秒）。增大此值将允许应用程序服务器有更多的时间在检测到消息后以破坏性方式获取该消息。更改 **MARKINT** 值会影响连接到该队列管理器并在应用程序以破坏性方式获取消息之前浏览消息的所有应用程序。
- 将 WebSphere Application Server 中 **com.ibm.msg.client.wmq.suppressBrowseMarkMessageWarning** 属性的值更改为 *true* 以禁止 CWSJY0003W 警告消息。要在 WebSphere Application Server 中设置变量，请打开管理控制台并浏览至 **服务器 -> 应用程序服务器 -> Java 和进程管理 -> 进程定义 -> Java 虚拟机 -> 定制属性 -> 新建**

```
Name = com.ibm.msg.client.wmq.suppressBrowseMarkMessageWarning
Value = true
```

注：如果激活规范或侦听器端口使用 IBM MQ 消息传递提供者迁移方式来连接到 IBM MQ，那么可以忽略这些消息。此运行方式的设计表示该消息可能在正常运行期间发生。

相关概念

激活规范

[在应用程序服务器工具 \(ASF\) 方式中运行的侦听器端口](#)

[在非应用程序服务器工具（非 ASF）方式中运行的侦听器端口](#)

相关任务

[避免重复传递已浏览过的消息](#)

相关参考

[ALTER QMGR](#)

包含以下错误的 J2CA0027E 消息：The method 'xa_end' has failed with errorCode '100'

WebSphere Application Server SystemOut.log 中显示包含以下错误的 J2CA0027E 消息：The method 'xa_end' has failed with errorCode '100'。

介绍

当使用 WebSphere Application Server IBM MQ 消息传递提供程序的应用程序尝试落实事务时，WebSphere Application Server SystemOut.log 文件中将显示以下错误：

```
J2CA0027E: An exception occurred while invoking end on an XA Resource Adapter from
DataSource JMS_Connection_Factory, within transaction ID Transaction_Identifier:
javax.transaction.xa.XAException: The method 'xa_end' has failed with errorCode '100'.
```

```
J2CA0027E: An exception occurred while invoking rollback on an XA Resource Adapter
from DataSource JMS_Connection_Factory, within transaction ID Transaction_Identifier:
javax.transaction.xa.XAException: The method 'xa_rollback' has failed with errorCode '-7'.
```

原因

这些错误的原因可能是由于 IBM MQ 消息传递提供程序 JMS 连接已被 WebSphere Application Server 关闭，因为该连接的过时超时已到期。

JMS 连接是从 JMS 连接工厂创建的。每个连接工厂都有关联的连接池，而连接池划分为活动池和空闲池两部分。

当应用程序关闭正在使用的 JMS 连接时，该连接将被移入连接工厂连接池的空闲池中，除非已超过连接的过时超时，在这种情况下该连接被销毁。如果 JMS 连接被销毁时仍参与某个活动事务，那么应用程序服务器会向 IBM MQ 传递 `xa_end()`，指示该连接上的所有事务工作均已完成。

如果在使用激活规范或侦听器端口来监视 IBM MQ 队列管理器上的 JMS 目标的事务性消息驱动的 Bean 中创建了 JMS 连接，那么这将导致问题。

在此情况下，会有单个事务使用与 IBM MQ 的两个连接：

- 一个是用于从 IBM MQ 获取消息并将消息传递到消息驱动的 bean 实例进行处理的连接。
- 另一个是在消息驱动的 bean `onMessage()` 方法内创建的连接。

如果消息驱动的 bean 将第二个连接关闭，然后由于超过超时超时而将其销毁，那么会向 IBM MQ 传递 `xa_end()` 以指示所有事务工作均已完成。

当消息驱动的 bean 应用程序完成所分配的消息处理工作时，应用程序服务器需要完成该事务。可以通过将 `xa_end()` 传递至该事务先前涉及的所有资源（包括 IBM MQ）来实现此目的。

但是，IBM MQ 已针对该特定事务收到过 `xa_end()`，因此将 `XA_RBROLLBACK (100)` 错误返回给 WebSphere Application Server，指示该事务已结束，IBM MQ 的所有工作均已回滚。这会导致应用程序服务器报告以下错误：

```
J2CA0027E: An exception occurred while invoking end on an XA Resource Adapter from
DataSource JMS_Connection_Factory, within transaction ID Transaction_Identifier:
javax.transaction.xa.XAException: The method 'xa_end' has failed with errorCode '100'.
```

然后通过将 `xa_rollback()` 传递至事务中征调的所有资源来回滚整个事务。当应用程序服务器将 `xa_rollback()` 传递至 IBM MQ 时，发生以下错误：

```
J2CA0027E: An exception occurred while invoking rollback on an XA Resource Adapter
from DataSource JMS_Connection_Factory, within transaction ID Transaction_Identifier:
javax.transaction.xa.XAException: The method 'xa_rollback' has failed with errorCode '-7'.
```

环境

使用激活规范或侦听器端口来监视 JMS 在 IBM MQ 队列管理器上托管的目标，然后从其 `onMessage()` 方法中使用 JMS 连接工厂创建到 IBM MQ 的新连接的消息驱动的 Bean 应用程序可能会受此问题影响。

解决问题

要解决该问题，请确保应用程序使用的 JMS 连接工厂将连接池属性超时设置为 0。这可防止 JMS 连接在返回到空闲池时被关闭，从而确保能够完成所有未完成的事务工作。

2035 从 WebSphere Application Server 连接到 IBM MQ 时 MQRC_NOT_AUTHORIZED

当应用程序从 WebSphere Application Server 连接到 IBM MQ 时，可能会发生 2035 `MQRC_NOT_AUTHORIZED` 错误。

本主题涵盖了在 WebSphere Application Server 中运行的应用程序在连接到 IBM MQ 时接收到 2035 `MQRC_NOT_AUTHORIZED` 错误的最常见原因。解决问题部分中提供了在开发期间暂时解决 2035 `MQRC_NOT_AUTHORIZED` 错误的快速步骤，同时还提供了在生产环境中实现安全性的注意事项。此外还汇总了容器管理安全性和组件管理安全性的出站场景行为以及侦听器端口和激活规范的进站行为。

问题原因

以下列表描述了 IBM MQ 拒绝连接的最常见原因：

- 通过客户机连接从应用程序服务器传递到 IBM MQ 的用户标识可能存在以下问题：在运行 IBM MQ 队列管理器的服务器上未知；无权连接到 IBM MQ；因超过 12 个字符而被截断。“诊断问题”中提供了有关如何获取和传递此用户标识的更多信息。

Windows 对于正在 Windows 上运行的队列管理器，可能会在此场景的 IBM MQ 错误日志中看到以下错误：`AMQ8075: Authorization failed because the SID for entity 'wasuser' cannot be obtained.`

UNIX 对于 UNIX，IBM MQ 错误日志中未显示任何条目。

- 通过客户机连接从应用程序服务器传递到 IBM MQ 的用户标识是托管 IBM MQ 队列管理器的服务器上 *mqm* 组的成员，并且存在阻止对队列管理器进行管理访问的通道认证记录 (CHLAUTH)。IBM MQ 缺省情况下在 IBM WebSphere MQ 7.1 或更高版本中配置 CHLAUTH 记录，该记录阻止所有 IBM MQ 管理员从客户机连接到队列管理器。在 IBM MQ 错误日志中将显示针对此场景的以下错误：AMQ9777: Channel was blocked。
- 存在 Advanced Message Security 安全策略。

有关 IBM MQ 错误日志的位置，请参阅[错误日志目录](#)。

诊断问题

要了解 2035 MQRC_NOT_AUTHORIZED 原因码的原因，您必须知道 IBM MQ 用于授权应用程序服务器的用户名和密码。

注：本主题提供的内容适用于开发环境，通常需要使用以下某种方法来满足生产环境的安全需求：

- 双向 SSL/TLS 认证

IBM MQ 提供了一些功能以用于使用针对 SSL/TLS 连接提供的数字证书来认证远程连接的客户机。

- 定制或第三方提供的 IBM MQ 安全出口

可以为 IBM MQ 编写安全出口，以用于针对存储库（例如，本地操作系统、IBM MQ 服务器或 LDAP 存储库）执行用户名和密码认证。当使用安全出口进行认证时，请务必继续配置 SSL/TLS 传输安全性，这样可以确保不会以明文形式发送密码。

服务器连接通道上配置的 MCA 用户标识

如果应用程序服务器使用服务器连接通道上配置的 MCA 用户标识进行连接，并且未安装任何安全出口或映射通道认证记录，那么该 MCA 用户标识将覆盖应用程序服务器所提供的用户名。通常，许多客户都会在每个服务器连接通道上设置 MCA 用户标识并仅使用双向 SSL/TLS 认证进行认证。

应用程序服务器未提供凭证时的缺省行为

如果应用程序没有在 **createConnection** 调用时提供凭证，并且未配置组件管理或容器管理安全系统，那么 WebSphere Application Server 会向 IBM MQ 提供空白的用户名。这会导致 IBM MQ 基于用来运行 IBM MQ 侦听器的用户标识来向客户机授权。在大多数情况下，用户标识为 P

-   UNIX 或 Linux 系统上的 *mqm*。
-  Windows 上的 *MUSR_MQADMIN*。

由于这些用户是 IBM MQ 管理用户，缺省情况下会在 IBM WebSphere MQ 7.1 和更高版本中阻止这些用户，并在队列管理器的错误日志中记录 AMQ9777 错误。

出站连接的容器管理安全性

要配置由应用程序服务器针对出站连接传递到 IBM MQ 的用户名和密码，建议使用容器管理安全性。出站连接是指使用连接工厂（而不是使用侦听器端口或激活规范）创建的连接。

应用程序服务器向 IBM MQ 传递了长度不超过 12 个字符的用户名。长度超过 12 个字符的用户名将在授权期间（在 UNIX 上）或在所发送消息的 MQMD 中被截断。容器管理安全性表示应用程序的部署描述符或 EJB 3.0 注释将声明认证类型设置为 Container 的资源引用。然后，当应用程序在 JNDI 中查找连接工厂时，它将通过该资源引用进行间接查找。例如，EJB 2.1 应用程序按如下所示执行 JNDI 查找，其中 *jms/MyResourceRef* 声明为部署描述符中的资源引用：

```
ConnectionFactory myCF = (ConnectionFactory)ctx.lookup("java:comp/env/jms/MyResourceRef")
```

EJB 3.0 应用程序可能按如下所示在 bean 上声明带有注释的对象属性：

```
@Resource(name = "jms/MyResourceRef"
    authenticationType = AuthenticationType.CONTAINER)
private javax.jms.ConnectionFactory myCF
```

当管理员部署应用程序时，他们将此认证别名绑定到已在 JNDI 中创建的实际连接工厂，然后在部署时为其分配 J2C 认证别名。这是此认证别名中包含的用户名和密码，应用程序服务器在应用程序连接时将其传递

到 IBM MQ 或 JMS。此方法使管理员能够控制每个应用程序使用的用户名和密码，防止其他应用程序使用相同的用户名和密码在 JNDI 中直接查找连接工厂。缺省容器管理认证别名在 IBM MQ 连接工厂管理控制台中的配置面板上提供。仅当应用程序使用针对容器管理安全性配置的资源引用，但管理员未在部署期间将其绑定到认证别名时才使用此缺省值。

出站连接的缺省组件管理认证别名

如果将应用程序更改为使用容器管理安全性，或者将其更改为直接在 createConnection 调用中提供用户名和密码这两种操作不切实际，可以提供缺省值。此缺省值称为组件管理认证别名，不能在管理控制台中进行配置（从 WebSphere Application Server 7.0 开始，已将其从 IBM MQ 连接工厂的面板中移除）。以下脚本编制样本显示了如何使用 wsadmin 对其进行配置：

- JAACL

```
wsadmin>set cell [ $AdminConfig getid "/Cell:mycell" ]
mycell(cells/mycell|cell.xml#Cell_1)
wsadmin>$AdminTask listWMQConnectionFactory $cell
MyCF(cells/mycell|resources.xml#MQConnectionFactory_1247500675104)
wsadmin>$AdminTask modifyWMQConnectionFactory MyCF(cells/mycell|
resources.xml#MQConnectionFactory_1247500675104) { -componentAuthAlias myalias }
MyCF(cells/mycell|resources.xml#MQConnectionFactory_1247500675104)
```

- Jython

```
wsadmin>cell = AdminConfig.getid("/Cell:mycell")
wsadmin>AdminTask.listWMQConnectionFactory(cell)
'MyCF(cells/mycell|resources.xml#MQConnectionFactory_1247500675104)'
wsadmin>AdminTask.modifyWMQConnectionFactory('MyCF(cells/mycell|resou
rces.xml#MQConnectionFactory_1247500675104)', "-componentAuthAlias myalias")
'MyCF(cells/mycell|resources.xml#MQConnectionFactory_1247500675104)'
```

使用激活规范的入站 MDB 连接所使用的认证别名

对于使用激活规范的入站连接，认证别名可以在部署应用程序时由管理员指定，也可以在管理控制台中的激活规范上指定缺省认证别名。

使用侦听器端口的入站 MDB 连接所使用的认证别名

对于使用侦听器端口的入站连接，将使用在连接工厂的容器管理认证别名设置中指定的值。在 z/OS 上，首先检查和使用容器管理认证别名（如果已设置），然后才检查和使用组件管理认证别名（如果已设置）。

解决问题

下面是在不需要完全传输安全性的开发环境中解决 2035 MQRC_NOT_AUTHORIZED 错误的最简单步骤：

- 选择想要 WebSphere Application Server 认证的用户。通常，所选用户应具有 WebSphere Application Server 中运行的应用程序所需的操作上下文相关权限，并且无其他权限。例如，mqm 或其他超级用户不适用此情况。
- 如果此用户是 IBM MQ 管理用户，请放宽 IBM WebSphere MQ 7.1 或更高版本中的通道认证记录 (CHLAUTH) 安全性级别，以便不会在要使用的服务器连接通道上阻止管理员进行连接。针对名为 WAS.CLIENTS 为 SET CHLAUTH('WAS.CLIENTS') TYPE(BLOCKUSER) USERLIST(ALLOWANY)。
- 配置服务器连接通道以将 MCA 用户标识 (MCAUSER) 设置为要使用的用户。用于配置服务器连接通道以使用 myuser 作为 MCA 用户标识 ALTER CHL('WAS.CLIENTS') CHLTYPE(SVRCONN) MCAUSER('myuser') 的示例 MQSC 命令。

生产环境的其他重要注意事项

对于需要传输安全性的所有生产环境，必须在应用程序服务器和 IBM MQ 之间配置 SSL/TLS 安全性。

要配置 SSL/TLS 传输安全性，必须在 IBM MQ 队列管理器和 WebSphere Application Server 之间建立适当的信任。应用程序服务器启动 SSL/TLS 握手，并且必须始终配置为信任 IBM MQ 队列管理器提供的证书。如果应用程序服务器配置为向 IBM MQ 队列管理器发送证书，那么该队列管理器也必须配置为信任该证书。如果没有在这两端正确配置信任，那么在连接上启用 SSL/TLS 后将遇到 2393 MQRC_SSL_INITIALIZATION_ERROR 原因码。

如果没有用于执行用户名和密码认证的安全出口，那么应在服务器连接通道上配置双向 SSL/TLS 认证，使队列管理器要求应用程序服务器提供可信证书。要执行此操作，请在 IBM MQ Explorer 或 MQSC 中的 SSLCAUTH(REQUIRED) 中将 SSL 认证 设置为 必需。

如果具有用于执行用户名和密码认证的安全出口并已将其安装在 IBM MQ 服务器中，那么将应用程序配置为提供用户名和密码以供该安全出口验证。先前在 诊断问题 部分中描述了如何配置由应用程序服务器传递到 IBM MQ 的用户名和密码的详细信息。

应当禁用不具有 SSL/TLS 安全性的所有服务器连接通道。用于禁用 `SYSTEM.DEF.SVRCONN` 通道按如下所示提供 (假定名为 ('NOAUTH'), `ALTER CHL(SYSTEM.DEF.SVRCONN) CHLTYPE(SVRCONN) MCAUSER('NOAUTH')` `STOP CHL(SYSTEM.DEF.SVRCONN)` 的 IBM MQ 服务器上不存在任何用户。

有关如何配置 IBM MQ 队列管理器的私有证书和信任以及如何如何在服务器连接通道上启用 SSL 安全性的指示信息，请参阅在队列管理器上配置 SSL 和在通道上配置 SSL。

有关从 WebSphere Application Server 使用 SSL/TLS 以及应用程序服务器是否向 IBM MQ 发送证书以进行认证的信息，请参阅以下信息：

- 要创建或修改 SSL 配置以包含用于连接到 IBM MQ 的相应 SSL/TLS 配置，请参阅 WebSphere Application Server 产品文档中的 SSL 配置。
- IBM MQ 要求您必须在连接的两端指定匹配的 CipherSpec。有关可用于 IBM MQ 的 CipherSpec 和 CipherSuite 的更多信息，请参阅用于 WebSphere® MQ 队列管理器连接的 CipherSuite 和 CipherSpec 名称映射。
- 有关在客户机连接上启用 SSL/TLS 以及选择要使用的 SSL 配置的更多信息，请参阅 WebSphere Application Server 产品文档中的 WebSphere MQ 消息传递提供程序连接工厂设置 和 WebSphere MQ 消息传递提供程序激活规范设置。

相关参考

第 147 页的『返回码 = 2035 MQRC_NOT_AUTHORIZED』

出于各种原因显示 RC2035 原因码，包括打开队列或通道时出错、尝试使用具有管理员权限的用户标识时收到错误、使用 IBM MQ JMS 应用程序和在集群上打开队列时出错。MQS_REPORT_NOAUTH 和 MQSAUTHERRORS 可用于进一步诊断 RC2035。

[2035 \(07F3\) \(RC2035\): MQRC_NOT_AUTHORIZED](#)

IBM MQ 资源适配器的问题确定

使用 IBM MQ 资源适配器时，大多数错误都会导致抛出异常，并将以取决于应用程序服务器的方式向用户报告这些异常。资源适配器会广泛使用链接异常来报告问题。通常，链中的第一个异常是错误的高级描述，链中的后续异常提供诊断问题所需的更详细信息。

例如，如果 IVT 程序无法与 IBM MQ 队列管理器连接，那么可能会抛出以下异常：

```
javax.jms.JMSEException: MQJCA0001: JMS 层中发生异常。  
See the linked exception for details.
```

与此异常相关联的是另一个异常：

```
javax.jms.JMSEException: MQJMS2005: failed to create an MQQueueManager for  
'localhost:ExampleQM'
```

此异常由 IBM MQ classes for JMS 抛出，并具有进一步的链接异常：

```
com.ibm.mq.MQException: MQJE001: An MQException occurred: Completion Code 2,  
Reason 2059
```

此最终异常指示问题的来源。原因码 2059 为 MQRC_Q_MGR_NOT_AVAILABLE，指示可能未启动 ConnectionFactory 对象定义中指定的队列管理器。

如果异常提供的信息不足以诊断问题，那么您可能需要请求诊断跟踪。有关如何启用诊断跟踪的信息，请参阅 [IBM MQ 资源适配器配置](#)。

配置问题通常出现在以下方面：

- 部署资源适配器时
- 部署 MDB 时
- 为出站通信创建连接时

相关任务

[使用 IBM MQ 资源适配器](#)

部署资源适配器时发生的问题

如果资源适配器部署失败，请检查是否正确配置了 Java EE Connector Architecture (JCA) 资源。如果已安装 IBM MQ，请检查类路径中是否包含正确版本 JCA 和 IBM MQ classes for JMS。

部署资源适配器失败通常由于未正确配置 JCA 资源。例如，可能未正确指定 ResourceAdapter 对象属性，或应用程序服务器需要的部署计划可能未正确写入。以下情况可能也会导致失败：应用程序服务器尝试从 JCA 资源定义创建对象并将对象绑定至 Java Naming Directory Interface (JNDI) 名称空间，但未正确指定某些属性或资源定义格式不正确。

资源适配器部署失败还可能是由于：从位于类路径下的 JAR 文件中装入了错误版本的 JCA 或 IBM MQ classes for JMS 类。此类失败情况通常发生在已经安装了 IBM MQ 的系统上。在此系统上，应用程序服务器可能会找到 IBM MQ classes for JMS JAR 文件的现有副本，并优先于 IBM MQ 资源适配器 RAR 文件中提供的类，首先从这些副本中装入类。

相关概念

[为 IBM MQ classes for JMS 安装哪些内容](#)

相关任务

[配置应用程序服务器以使用最新的资源适配器维护级别](#)

部署 MDB 时发生的问题

如果应用程序服务器尝试开始将消息传送到 MDB 时失败，那么原因可能是关联的 ActivationSpec 对象定义错误或者是缺少资源。

在应用程序服务器尝试开始将消息传送到 MDB 时，可能会失败。此类失败通常是由于关联的 ActivationSpec 对象定义错误，或是由于由于定义中引用的资源不可用。例如，队列管理器可能未在运行，或指定的队列可能不存在。

部署 MDB 时，ActivationSpec 对象会尝试验证其属性。如果 ActivationSpec 对象具有任何互斥的属性，或不具有所有必需的属性，那么部署将失败。但是，并非所有与 ActivationSpec 对象属性关联的问题此时都能被检测到。

至于如何向用户报告开始传送消息失败的情况，取决于应用程序服务器。通常，将在应用程序服务器的日志和诊断跟踪中报告这些失败情况。如果启用，那么 IBM MQ 资源适配器的诊断跟踪也会记录这些失败情况。

创建出站通信连接时发生的问题

如果找不到 ConnectionFactory 对象，或如果找到 ConnectionFactory 对象但无法创建连接，那么出站通信可能会失败。这些问题中的任一个问题都有各种原因。

应用程序尝试在 JNDI 名称空间中查找和使用 ConnectionFactory 对象时，出站通信通常会失败。如果此名称空间中找不到 ConnectionFactory 对象，那么将抛出 JNDI 异常。出于以下原因，可能找不到 ConnectionFactory 对象：

- 应用程序为 ConnectionFactory 对象指定了不正确的名称。
- 应用程序服务器无法创建 ConnectionFactory 对象并将其绑定到此名称空间。在此情况下，应用程序服务器的启动日志通常包含失败的相关信息。

如果应用程序成功地从 JNDI 名称空间检索 ConnectionFactory 对象，那么在应用程序调用 ConnectionFactory.createConnection() 方法时，仍可能会抛出异常。此背景下发生异常，指示无法创建与 IBM MQ 队列管理器的连接。以下是可能抛出异常的一些常见的原因：

- 队列管理器不可用，或使用 ConnectionFactory 对象属性无法找到。例如，队列管理器未在运行，或队列管理器的指定主机名、IP 地址或端口号均不正确。
- 用户无权连接到队列管理器。对于客户机连接，如果 createConnection() 调用未指定用户名，并且应用程序服务器未提供用户身份信息，那么 JVM 进程标识将作为用户名传递到队列管理器。要成功连接，此进程标识必须是运行队列管理器的系统中有效的用户名。
- ConnectionFactory 对象具有名为 ccdtURL 的属性和名为 channel 的属性。这两个属性是互斥的。
- 在 TLS 连接上，未正确指定服务器连接通道定义中的与 TLS 相关的特性或与 TLS 相关的属性。

- `sslFipsRequired` 属性对于不同的 JCA 资源有不同的值。有关此限制的更多信息，请参阅 [IBM MQ 资源适配器限制](#)。

相关任务

指定运行时在 MQI 客户机上仅使用经过 FIPS 认证的 CipherSpecs

相关参考

针对 UNIX, Linux, and Windows 的美国联邦信息处理标准 (FIPS)

使用 IBM MQ 连接属性覆盖

通过连接属性覆盖，可以更改客户机应用程序用于连接到队列管理器的详细信息，而不修改源代码。

关于此任务

有时可能无法修改应用程序的源代码，例如，如果该应用程序是旧应用程序，并且源代码不再可用。

在此情况下，如果应用程序在连接到队列管理器时需要指定其他属性，或者需要连接到其他队列管理器，那么可以使用连接覆盖功能来指定新的连接详细信息或队列管理器名称。

连接属性覆盖在两种客户机上均受支持：

- [IBM MQ classes for JMS](#)
- [IBM MQ classes for Java](#)

您可以通过在之后由 IBM MQ classes for JMS 或 IBM MQ classes for Java 在启动时读取的配置文件中定义要更改的属性来覆盖这些属性。

当连接覆盖功能在使用中时，在同一 Java runtime environment 内运行的所有应用程序都选取并使用新属性值。如果使用 IBM MQ classes for JMS 或 IBM MQ classes for Java 的多个应用程序是在同一 Java runtime environment 内运行，那么不能仅覆盖个别应用程序的属性。

要点：只有在无法修改应用程序源代码的情况下才支持此功能。它不得用于其中源代码可用并可更新的应用程序。

相关概念

第 339 页的『跟踪 IBM MQ classes for JMS 应用程序』

IBM MQ classes for JMS 中提供了跟踪功能来帮助 IBM 支持人员诊断客户问题。各种属性均可控制此功能的行为。

相关任务

第 342 页的『跟踪 IBM MQ classes for Java 应用程序』

IBM MQ classes for Java 中提供了跟踪功能来帮助 IBM 支持人员诊断客户问题。各种属性均可控制此功能的行为。

[使用 IBM MQ classes for JMS](#)

[使用 IBM MQ classes for Java](#)

在 IBM MQ classes for JMS 中使用连接属性覆盖

如果连接工厂以编程方式创建，并且无法修改创建该连接工厂的应用程序的源代码，那么在创建连接时，可以使用连接覆盖功能来更改连接工厂使用的属性。但是，不支持将连接覆盖功能与 JNDI 中定义的连接工厂结合使用。

关于此任务

在 IBM MQ classes for JMS 中，有关如何连接到队列管理器的详细信息存储在连接工厂中。连接工厂可以通过管理方式进行定义并存储在 JNDI 存储库中，或者由应用程序使用 Java API 调用以编程方式创建。

如果应用程序以编程方式创建连接工厂，并且无法修改该应用程序的源代码，那么连接覆盖功能允许您短期覆盖连接工厂属性。但是长期而言，必须实施相应的计划，以便能够修改应用程序使用的连接工厂，而不使用连接覆盖功能。

如果由应用程序以编程方式创建的连接工厂定义为使用客户机通道定义表 (CCDT)，那么会优先于已覆盖的属性使用 CCDT 中的信息。如果应用程序使用的连接详细信息需要更改，那么必须创建新版本的 CCDT 并使其对于应用程序可用。

不支持将连接覆盖功能与 JNDI 中定义的连接工厂结合使用。如果应用程序使用 JNDI 中定义的连接工厂，并且该连接工厂的属性需要更改，那么必须在 JNDI 中更新连接工厂的定义。虽然对这些连接工厂应用了连接覆盖功能（并且已覆盖的属性优先于在 JNDI 中查找的连接工厂定义中的属性），那么不支持连接覆盖功能的这一使用。

要点: 连接覆盖功能影响正在 Java runtime environment 内运行的所有应用程序，并且应用于这些应用程序使用的所有连接工厂。无法仅覆盖个别连接工厂或应用程序的属性。

当应用程序使用连接工厂来创建与队列管理器的连接时，IBM MQ classes for JMS 查看已被覆盖的属性，并在创建连接时使用这些属性值，而不是连接工厂中相同属性的值。

例如，假设已定义连接工厂，其中 PORT 属性设置为 1414。如果已使用连接覆盖功能将 PORT 属性设置为 1420，那么在连接工厂用于创建连接时，IBM MQ classes for JMS 对 PORT 属性使用值 1420 而不是 1414。

要修改在从连接工厂创建 JMS 连接时使用的任何连接属性，需要执行以下步骤：

1. 将要覆盖的属性添加到 JMS 配置文件的 IBM MQ 类。
2. 启用连接覆盖功能。
3. 启动应用程序，指定配置文件。

过程

1. 将要覆盖的属性添加到 IBM MQ classes for JMS 配置文件。

- a) 创建包含需要覆盖的标准 Java 属性格式的属性和值的文件。

有关如何创建属性文件的详细信息，请参阅 [IBM MQ classes for JMS 配置文件](#)。

- b) 要覆盖属性，请向属性文件中添加条目。

可以覆盖任何 IBM MQ classes for JMS 连接工厂属性。采用以下格式添加每个必需条目：

```
jmscf.property name=value
```

其中 *property name* 是 JMS 管理属性名称或需要覆盖的属性的 XMSC 常量。有关连接工厂属性的列表，请参阅 IBM MQ classes for JMS 对象的属性。

例如，要设置应用程序应该用于连接到队列管理器的通道的名称，可以向属性文件添加以下条目：

```
jmscf.channel=MY.NEW.SVRCONN
```

2. 启用连接覆盖功能。

要启用连接覆盖，请将 **com.ibm.msg.client.jms.overrideConnectionFactory** 属性设置为 true，以便使用在属性文件中指定的属性来覆盖应用程序中指定的值。可以将额外属性设置为配置文件本身中的另一个属性，或者使用以下命令将属性作为 Java 系统属性传递：

```
-Dcom.ibm.msg.client.jms.overrideConnectionFactory=true
```

3. 启动应用程序，指定配置文件。

通过设置 Java 系统属性，在运行时将所创建的属性文件传递到应用程序：

```
-Dcom.ibm.msg.client.config.location
```

请注意，配置文件的位置必须指定为 URI，例如：

```
-Dcom.ibm.msg.client.config.location=file:///jms/jms.config
```


结果

在启用连接覆盖功能时，只要进行连接，IBM MQ classes for JMS 就会向 JMS 日志中写入条目。日志中的信息显示创建连接时覆盖的连接工厂属性，如以下示例条目所示：

```
Overriding ConnectionFactory properties:
  Overriding property channel:
    Original value = MY.OLD.SVRCONN
    New value      = MY.NEW.SVRCONN
```

相关任务

第 73 页的『在 IBM MQ classes for Java 中使用连接属性覆盖』

在 IBM MQ classes for Java 中，连接详细信息设置为使用不同值的组合的属性。如果无法修改应用程序的源代码，那么可以使用连接覆盖功能来覆盖应用程序使用的连接详细信息。

第 76 页的『覆盖连接属性：IBM MQ classes for JMS 示例』

此示例显示在使用 IBM MQ classes for JMS 时如何覆盖属性。

在 IBM MQ classes for JMS 应用程序中创建并配置连接工厂和目标

在 JNDI 名称空间中配置连接工厂和目标

在 IBM MQ classes for Java 中使用连接属性覆盖

在 IBM MQ classes for Java 中，连接详细信息设置为使用不同值的组合的属性。如果无法修改应用程序的源代码，那么可以使用连接覆盖功能来覆盖应用程序使用的连接详细信息。

关于此任务

用于设置连接属性的不同值是以下各项的组合：

- 将值分配给 **MQEnvironment** 类上的静态字段。
- 设置 **MQEnvironment** 类中属性 **Hashtable** 中的属性值。
- 设置传递到 **MQQueueManager** 构造函数中的 **Hashtable** 中的属性值。

然后，当应用程序构造表示与队列管理器的连接的 **MQQueueManager** 对象时，将使用这些属性。

每个属性都有一个标识（即属性名称），它是一个字符串文字。例如，用于指定 IBM MQ 的主机名的属性由字面值 "hostname" 标识。

要定义应用程序的应用程序名称，您可以在 Java 代码中使用类似于以下内容的代码：

```
Hashtable properties = new Hashtable();
properties.Add("hostname", "localhost");
MQQueueManager qMgr = new MQQueueManager("qmgrname", properties);
```

但是，该字面值是 IBM MQ classes for Java 内部实现的一部分。如果文字部分发生更改（尽管这不太可能），而不是使用文字值，那么应该使用在 **MQConstants** 类中定义的相应常量值。

该常量是 IBM MQ classes for Java 记录的外部接口的一部分，并且不会更改。

对于主机名，此常量为 **HOST_NAME_PROPERTY**，因此首选代码为：

```
Hashtable properties = new Hashtable();
properties.Add( MQConstants.HOST_NAME_PROPERTY, "ExampleApplName" );
MQQueueManager qMgr = new MQQueueManager("qmgrname", properties);
```

可以在程序中设置的完整属性集如下表中所示：

| 属性 | MQConstants 中的常量名称 |
|-------|---------------------------|
| CCSID | CCSID_PROPERTY |
| 通道 | CHANNEL_PROPERTY |

| 属性 | MQConstants 中的常量名称 |
|------------------------|----------------------------|
| 连接选项 | CONNECT_OPTIONS_PROPERTY |
| Hostname | HOST_NAME_PROPERTY |
| SSL 密钥重置 | SSL_RESET_COUNT_PROPERTY |
| 本地地址 | LOCAL_ADDRESS_PROPERTY |
| 密码 | PASSWORD_PROPERTY |
| 端口 | PORT_PROPERTY |
| 密码套件 | SSL_CIPHER_SUITE_PROPERTY |
| 需要 FIPS | SSL_FIPS_REQUIRED_PROPERTY |
| SSL 对等名称 | SSL_PEER_NAME_PROPERTY |
| 用户标识 | USER_ID_PROPERTY |
| V 9.1.2 V 9.1.2 应用程序名称 | APPNAME_PROPERTY |

注: 该表不会列出字面值, 因为如已指出的那样, 这些值是 IBM MQ classes for Java 实现的一部分, 并且可能发生更改。

如果无法修改使用 IBM MQ classes for Java 的应用程序的源代码来指定在创建与队列管理器的连接时必须使用的不同属性, 那么连接覆盖功能允许您短期覆盖连接详细信息。但是长期而言, 必须实施相应的计划, 以便能够修改应用程序使用的连接详细信息, 而不使用连接覆盖功能。

在应用程序创建 **MQQueueManager** 时, IBM MQ classes for Java 会查看已覆盖的属性, 并在创建与队列管理器的连接时使用这些属性值, 而不是下列任何位置中的值:

- MQEnvironment 类上的静态字段
- 存储在 MQEnvironment 类中的属性 Hashtable
- 传递到 **MQQueueManager** 构造函数中的属性 Hashtable

例如, 假定应用程序创建 **MQQueueManager**, 并传递将 CHANNEL 属性设置为 MY.OLD.CHANNEL 的属性散列表。如果已使用连接覆盖功能将 CHANNEL 属性设置为 MY.NEW.CHANNEL, 那么在构造 **MQQueueManager** 时, IBM MQ classes for Java 会尝试使用通道 MY.NEW.CHANNEL 而不是 MY.OLD.CHANNEL 创建与队列管理器的连接。

注: 如果 **MQQueueManager** 配置为使用客户机通道定义表 (CCDT), 那么将优先使用 CCDT 中的信息, 而不使用被覆盖的属性。如果需要更改创建 **MQQueueManager** 的应用程序所使用的连接详细信息, 那么必须创建 CCDT 的新版本并将其提供给该应用程序。

要修改在创建 **MQQueueManager** 时使用的任何连接属性, 您需要执行以下步骤:

1. 创建名为 mqclassesforjava.config 的属性文件。
2. 启用连接属性覆盖功能 (通过将 **OverrideConnectionDetails** 属性设置为 true)。
3. 启动应用程序, 在 Java 调用过程中指定配置文件。

过程

1. 创建名为 mqclassesforjava.config 的属性文件, 其中包含需要覆盖的属性和值。

在作为 **MQQueueManager** 构造函数的一部分连接到队列管理器时, 可以覆盖 IBM MQ classes for Java 所使用的 13 个属性。

| 属性 | 属性关键字 |
|-------|------------------|
| CCSID | \$CCSID_PROPERTY |

| 表 2: 可以覆盖的属性 (继续) | |
|--|------------------------------|
| 属性 | 属性关键字 |
| 通道 | \$CHANNEL_PROPERTY |
| 连接选项 | \$CONNECT_OPTIONS_PROPERTY |
| Hostname | \$HOST_NAME_PROPERTY |
| SSL 密钥重置 | \$SSL_RESET_COUNT_PROPERTY |
| 本地地址 | \$LOCAL_ADDRESS_PROPERTY |
| 队列管理器名称 | qmgr |
| 密码 | \$PASSWORD_PROPERTY |
| 端口 | \$PORT_PROPERTY |
| 密码套件 | \$SSL_CIPHER_SUITE_PROPERTY |
| 需要 FIPS | \$SSL_FIPS_REQUIRED_PROPERTY |
| SSL 对等名称 | \$SSL_PEER_NAME_PROPERTY |
| 用户标识 | \$USER_ID_PROPERTY |
|   应用程序名称 | \$APPNAME_PROPERTY |

注意:

- a. 除队列管理器名称以外，所有属性关键字都以 \$ 字符开头。原因是队列管理器名称作为自变量传递到 **MQueueManager** 构造函数，而不是设置为 MQEnvironment 类上的静态字段或 Hashtable 中的属性，因此在内部需要以与其他属性稍有不同的方式处理此属性。
- b. 以 \$ 字符开头的属性键通过引用 MQConstants.java 中定义的常量值进行处理，如前面的文本中所述。

您可以（但不应该）使用这些常量的字面值，在这种情况下，将省略 \$ 字符

要覆盖属性，请向属性文件中添加以下格式的条目：

```
mqj.property key=value
```

例如，要设置在创建 **MQueueManager** 对象时要使用的通道的名称，您可以将以下条目添加到属性文件：

```
mqj.$CHANNEL_PROPERTY=MY.NEW.CHANNEL
```


要更改 **MQueueManager** 对象所连接到的队列管理器的名称，您可以将以下条目添加到属性文件：

```
mqj.qmgr=MY.OTHER.QMGR
```

2. 通过将 **com.ibm.mq.overrideConnectionDetails** 属性设置为 true 启用连接覆盖功能。

将属性 **com.ibm.mq.overrideConnectionDetails** 设置为 true 意味着将使用属性文件中指定的属性覆盖应用程序中指定的值。可以将额外属性设置为配置文件本身中的另一个属性，或者使用以下命令将属性作为系统属性传递：

```
-Dcom.ibm.mq.overrideConnectionDetails=true
```

 需要使用 IBM MQ 设置特定应用程序名称的应用程序可以通过以下三种方式之一来执行此操作：

- 通过使用上述文本中描述的覆盖机制，定义 `mqj.$APPNAME_PROPERTY` 属性。

`mqj.$APPNAME_PROPERTY` 属性的值指定用于标识与队列管理器的连接的名称，并且仅使用前 28 个字符。例如：

```
mqj.$APPNAME_PROPERTY=ExampleAppName
```

注：您可能会看到使用属性名称的字面值的示例，例如，在较早的文档中。例如，`mqj.APPNAME=ExampleAppName`。

- 您可以将此值传递到 `properties` HashTable 中的 `MQQueueManager` 构造函数，仅使用前 28 个字符。例如：

```
Hashtable properties = new Hashtable();
properties.Add( MQConstants.APPNAME_PROPERTY, "ExampleAppName" );
MQQueueManager qMgr = new MQQueueManager("qmgrname", properties);
```

- 您可以在 `MQEnvironment` 类中设置 `AppName` 属性，仅使用前 28 个字符。例如：

```
MQEnvironment.AppName = "ExampleAppName";
```

3. 启动应用程序。

通过设置 Java 系统属性，在运行时将所创建的属性文件传递到客户机应用程序：

```
-Dcom.ibm.msg.client.config.location
```

请注意，配置文件的位置必须指定为 URI，例如：

```
-Dcom.ibm.msg.client.config.location=file:///classesforjava/mqclassesforjava.config
```

覆盖连接属性：IBM MQ classes for JMS 示例

此示例显示在使用 IBM MQ classes for JMS 时如何覆盖属性。

关于此任务

以下代码示例显示应用程序如何以编程方式创建 `ConnectionFactory`：

```
JmsSampleApp.java
...
JmsFactoryFactory jmsff;
JmsConnectionFactory jmsConnFact;

jmsff = JmsFactoryFactory.getInstance(JmsConstants.WMQ_PROVIDER);
jmsConnFact = jmsff.createConnectionFactory();

jmsConnFact.setStringProperty(WMQConstants.WMQ_HOST_NAME, "127.0.0.1");
jmsConnFact.setIntProperty(WMQConstants.WMQ_PORT, 1414);
jmsConnFact.setStringProperty(WMQConstants.WMQ_QUEUE_MANAGER, "QM_V80");
jmsConnFact.setStringProperty(WMQConstants.WMQ_CHANNEL, "MY.CHANNEL");
jmsConnFact.setIntProperty(WMQConstants.WMQ_CONNECTION_MODE,
                           WMQConstants.WMQ_CM_CLIENT);
...
```

`ConnectionFactory` 配置为使用 `CLIENT` 传输和通道 `MY.CHANNEL` 连接到队列管理器 `QM_V80`。

可以通过使用属性文件来覆盖连接详细信息，并且通过使用以下过程来强制应用程序连接到其他通道。

过程

1. 在 `/userHome` 目录中创建名为 `jms.config` 的 IBM MQ classes for JMS 配置文件（其中 `userHome` 是主目录）。

使用以下内容创建此文件：

```
jmscf.CHANNEL=MY.TLS.CHANNEL
jmscf.SSLCIPHERSUITE=TLS_RSA_WITH_AES_128_CBC_SHA256
```

2. 运行应用程序，将以下 Java 系统属性传递到应用程序运行所在的 Java runtime environment 中：

```
-Dcom.ibm.msg.client.config.location=file:///userHome/jms.config
-Dcom.ibm.msg.client.jms.overrideConnectionFactory=true
```

结果

执行此过程将覆盖应用程序以编程方式创建的 `ConnectionFactory`，因此在应用程序创建连接时，它通过使用通道 `MY.TLS.CHANNEL` 和密码套件 `TLS_RSA_WITH_AES_128_CBC_SHA256` 来尝试连接。

相关任务

第 71 页的『使用 IBM MQ 连接属性覆盖』

通过连接属性覆盖，可以更改客户机应用程序用于连接到队列管理器的详细信息，而不修改源代码。

第 71 页的『在 IBM MQ classes for JMS 中使用连接属性覆盖』

如果连接工厂以编程方式创建，并且无法修改创建该连接工厂的应用程序的源代码，那么在创建连接时，可以使用连接覆盖功能来更改连接工厂使用的属性。但是，不支持将连接覆盖功能与 JNDI 中定义的连接工厂结合使用。

第 73 页的『在 IBM MQ classes for Java 中使用连接属性覆盖』

在 IBM MQ classes for Java 中，连接详细信息设置为使用不同值的组合的属性。如果无法修改应用程序的源代码，那么可以使用连接覆盖功能来覆盖应用程序使用的连接详细信息。

对 Managed File Transfer 问题进行故障诊断

使用以下参考信息可帮助您诊断 Managed File Transfer 中的错误：

使用 MFT 的提示与技巧

以下建议可帮助您最有效地使用 Managed File Transfer。

- 如果更改 `agent.properties` 文件，请停止并重新启动代理以选取更改。
- 如果您启动文件传输，并且看不到传输进度，也未报告任何错误，请检查源代理是否正在运行。如果已显示传输但是无进度，请检查目标代理是否也正在运行。您可在代理日志中检查代理的当前状态或者使用 `ftePingAgent` 命令验证代理是否活动。
- 使用 `fteCancelTransfer` 命令取消个别传输时，可以在 `-agentName` 参数中使用源代理或目标代理。但是，使用 `fteDeleteScheduledTransfer` 命令删除传输调度时，必须在 `-agentName` 参数中使用源代理名称。
- 创建文件传输时，源文件路径和目标文件路径（绝对或相对路径）仅在源代理和目标代理上有效。发出 `fteCreateAgent` 命令的系统和目录与正在传输的文件没有相关性。
- 您的缺省环境设置可能无法充分支持 Managed File Transfer，尤其是如果您正在运行多个并行传输的情况下。如果代理存在错误，指示其内存不足，那么请根据需要检查并更新以下参数：
 - **UNIX** 对于 UNIX 平台：请运行命令：`ulimit -m 1048576`（或者大约 1 GB）。该最大常驻集合大小足以允许最多 25 个并行传输（25 个并行传输是一个代理的缺省最大传输数）。
 - 对于所有平台：按如下所示设置 `BFG_JVM_PROPERTIES` 环境变量：`BFG_JVM_PROPERTIES="-Xmx1024M"`

如果要允许并行传输数大于缺省最大值 25，请针对 `ulimit` 和 `BFG_JVM_PROPERTIES` 使用大于建议值的大小。

注：对于 Connect:Direct 网桥代理，最大并行传输数量的缺省值为 5。

- 使用 Managed File Transfer 以文本方式在不同平台间传输文件时，源平台的缺省文件编码可能不受目标平台的支持。这会导致传输失败，并出现以下错误：

```
BFGI00058E: The transfer source encoding xxx is illegal or for an unsupported character set.
```

您可通过使用环境变量将源编码设置为目标平台支持的编码来解决此问题。按如下所示在源系统上设置 **BFG_JVM_PROPERTIES** 系统环境变量：`BFG_JVM_PROPERTIES="-Dfile.encoding=xxx"`，其中 `xxx` 是目标平台支持的编码。例如，如果您要以文本方式将文件从 Solaris 平台传输至不同的平台，并且源语言环境设置为“ja”，请按如下所示设置 **BFG_JVM_PROPERTIES**：`BFG_JVM_PROPERTIES="-Dfile.encoding=EUC-JP"`。如果源语言环境设置为“ja_JP.PCK”，请按如下所示设置 **BFG_JVM_PROPERTIES**：`BFG_JVM_PROPERTIES="-Dfile.encoding=Shift_JIS"`。

您还可在启动新传输时使用 **-sce** 参数来解决个别传输的该错误。有关更多信息，请参阅 **fteCreateTransfer**：启动新的文件传输。

- 在可能的情况下，请勿将单个代理同时用作同一受管传输的源代理和目标代理。这会给代理带来额外的负载，这可能会影响其参与的其他受管传输，并导致这些传输进入恢复状态。

相关参考

[MFT 的 Java 系统属性](#)

MFT 的返回码

Managed File Transfer 命令、Ant 任务和日志消息均提供了用于指示是否已成功完成功能的返回码。

下表列出了产品返回码及其含义：

| 返回码 | 短名称 | 描述 |
|-----|---------|---|
| 0 | 成功 | 命令成功 |
| 1 | 命令失败 | 命令未成功结束。 |
| 2 | 命令超时 | 代理未在指定的超时内答复命令状态。缺省情况下，该超时对于受管调用和传输命令而言是无限的。例如，当您使用 fteCreateTransfer 命令指定 -w 参数时。缺省情况下，该超时对于其他命令是 5 秒。 |
| 3 | 确认超时 | 代理在指定的超时内未确认接收到命令。缺省情况下，该超时是 5 秒。 |
| 4 | 错误的代理 | 将命令发送到了错误的代理。在命令 XML 中指定的代理不是正在读消息所在命令队列的代理。 |
| 20 | 传输部分成功 | 传输已完成，但只有部分成功，传输了一些文件。 |
| 21 | 传输已停止 | 传输已被一个用户出口停止。 |
| 22 | 取消传输超时 | 代理已收到取消传输的请求，但该取消未能在 30 秒内完成。未取消该传输。 |
| 26 | 未找到取消标识 | 代理已收到取消传输的请求，但未找到该传输。这可能是由于代理处理该取消请求之前，该传输已完成。这也可能是因为，您向 fteCancelTransfer 命令提供的传输标识不正确。已忽略该取消请求。 |

表 3: 返回码 (继续)

| 返回码 | 短名称 | 描述 |
|-----|--------------|---|
| 27 | 正在取消中 | 代理已收到取消传输的请求, 但该传输已处于被取消的过程中。已忽略新的取消传输请求。 |
| 40 | 失败 | 传输失败, 未传输任何指定的文件。 |
| 41 | 已取消 | 已取消该传输。 |
| 42 | 触发失败 | 传输未进行, 因为传输是有条件的并且未满足所需条件。 |
| 43 | 格式不正确的 XML | XML 消息的格式不正确。 |
| 44 | 已超出源代理容量 | 源代理没有足够的容量来执行该传输。 |
| 45 | 已超出目标代理容量 | 目标代理没有足够的容量来执行该传输。 |
| 46 | 已超出源代理最大文件数 | 要传输的文件数已超出源代理的限制。 |
| 47 | 已超出目标代理最大文件数 | 传输的文件数已超出目标代理的限制。 |
| 48 | 无效的日志消息属性 | 日志消息的格式不正确。这是内部错误。如果您收到该返回码, 请与 IBM 支持中心联系以获取进一步的帮助。 |
| 49 | 目标不可达 | 由于出现 IBM MQ 问题, 源代理不能将消息发送到目标代理。例如, 如果源代理队列管理器未正确配置以与目标代理队列管理器进行通信, 将发生该问题。 |
| 50 | 试用版违例 | 试用版代理尝试与非试用版代理进行通信。 |
| 51 | 未准许源传输 | maxSourceTransfers 代理属性已设置为 0。不允许此代理作为任何传输的源。 |
| 52 | 未准许目标传输 | maxDestinationTransfers 代理属性已设置为 0。不允许将此代理程序作为任何传输的目标。 |
| 53 | 未经授权 | 用户无权执行该操作。请参阅随附消息, 以获取进一步的详细信息。 |
| 54 | 权限级别不匹配 | 源代理和目标代理的 authorityChecking 代理属性值不匹配。 |
| 55 | 不支持触发器 | 尝试使用协议网桥代理上的触发器创建传输。该行为不受支持。 |

表 3: 返回码 (继续)

| 返回码 | 短名称 | 描述 |
|-----|-----------------|---|
| 56 | 不支持目标文件进行消息传递 | 目标代理不支持将文件写到目标队列。 |
| 57 | 不支持文件空间 | 目标代理不支持文件空间。 |
| 58 | 已拒绝文件空间 | 目标代理已拒绝文件空间传输。 |
| 59 | 不支持目标消息到文件 | 目标代理不支持消息到文件传输。 |
| 64 | 两个队列都不被允许 | 传输的源和目标是队列。 |
| 65 | 一般数据队列错误 | 访问 Managed File Transfer Agent 数据队列时发生错误。 |
| 66 | 数据队列 put 权限错误 | 访问 Managed File Transfer Agent 数据队列时发生错误。未启用 Advanced Message Security。 |
| 67 | 数据队列 put AMS 错误 | 访问 Managed File Transfer Agent 数据队列时发生权限错误。已启用 Advanced Message Security。 |
| 69 | 传输恢复发生超时 | 在指定的 transferRecoveryTimeout 值后, 恢复传输的操作发生超时。 |
| 70 | 代理已异常结束 | 应用程序发生不可恢复的问题并且被强制终止。 |
| 75 | 队列管理器不可用 | 应用程序无法继续, 因为应用程序的队列管理器不可用。 |
| 78 | 启动配置存在问题 | 由于启动配置数据存在问题, 因此应用程序无法继续。 |
| 85 | | 由于数据库存在问题 (通常仅由记录器返回), 因此应用程序无法继续。 |
| 100 | 监视器替换无效 | 监视器任务 XML 脚本内的变量替换的格式不正确。 |
| 101 | 监视器资源错误 | 监视器资源定义数无效。 |
| 102 | 监视器触发器错误 | 监视器触发器定义数无效。 |
| 103 | 监视器任务错误 | 监视器任务定义数无效。 |
| 104 | 监视器缺失 | 请求的监视器不存在。 |
| 105 | 监视器已存在 | 请求的监视器已存在。 |
| 106 | 监视器用户出口错误 | 监视器用户出口已在资源监视器轮询期间生成错误。 |
| 107 | 监视器用户出口已取消 | 监视器用户出口已请求取消事务。 |
| 108 | 监视器任务失败 | 由于在处理任务时出错, 监视器任务未能完成。 |

表 3: 返回码 (继续)

| 返回码 | 短名称 | 描述 |
|-----|------------------|--|
| 109 | 监视器资源失败 | 无法将监视器资源定义应用到指定资源。 |
| 110 | 监视器任务变量替换失败 | 已在监视器任务中指定变量，但在元数据中未找到任何匹配的名称。因此，无法将该变量替换为值。 |
| 111 | 监视器任务源代理无效 | 监视器传输任务的源代理与资源监视器的代理不匹配。 |
| 112 | 监视器任务源队列管理器无效 | 监视器传输任务的源代理队列管理器与资源监视器的代理队列管理器不匹配。 |
| 113 | 不支持监视器 | 已尝试在协议网桥代理上创建或删除资源监视器。该行为不受支持。 |
| 114 | 监视器资源被拒绝 | 监视器资源扫描的目录拒绝访问。 |
| 115 | 监视器资源队列正在使用中 | 监视器资源队列已打开，但不能使用共享访问权进行输入。 |
| 116 | 未知监视器资源队列 | 监视器的关联队列管理器上不存在此监视器资源队列。 |
| 118 | 监视器资源表达式无效 | 对 XPath 表达式求值时发生错误。对 XPath 表达式求值，以访问消息头中用户定义的属性。消息位于资源监视器监视的队列上。 |
| 119 | 监视器任务源代理队列管理器缺失 | 监视器任务定义中缺少源代理名称或源代理队列管理器名称。 |
| 120 | 监视器队列未启用 | 监视器资源队列未启用。 |
| 121 | 访问监视队列时发生意外错误 | 访问监视资源队列时发生了意外错误。 |
| 122 | 未对上下文标识启用监视器命令队列 | 未对设置上下文标识启用监视器代理命令队列。 |

下表列出了产品中间回复代码及其含义：

表 4: 中间回复代码

| 回复代码 | 短名称 | 描述 |
|------|----------|-----------------------|
| -2 | ACK | 请求已收到，但正在暂挂完成。 |
| -3 | PROGRESS | 该请求用于多个文件，但其中某些仍暂挂完成。 |

注：

仅在生成请求的进程提供了回复队列的情况下，才存在回复代码。这些都是中间回复，Managed File Transfer 命令仅返回最终回复代码。

相关参考

第 82 页的『针对传输中文件的返回码』

传输中的各个文件都具有自己的结果代码，这些代码的含义不同于命令的总体返回码。

针对传输中文件的返回码

传输中的各个文件都具有自己的结果代码，这些代码的含义不同于命令的总体返回码。

在将 <action> 元素设置为“progress”值的传输日志进度消息中，报告的每个文件都有一个带有 resultCode 的 <status> 元素。例如：

```
<action time="2009-11-23T21:28:09.593Z">progress</action>

...
  <status resultCode="1">
    <supplement>BFGI00006E: File &quot;C:\destinationfiles\dest1.doc&quot;
      already exists.</supplement>
  </status>
```

下表描述了 resultCode 的可能值：

| 表 5: 传输中的文件结果代码 | |
|-----------------|---|
| 结果代码值 | 描述 |
| 0 | 成功。文件传输成功。 |
| 1 | 失败。文件传输失败。请参阅 <supplement> 元素，以获取该错误的更多详细信息。 |
| 2 | 警告。文件已传输，但报告了一条警告消息。例如，虽然源处置设置为删除，但无法删除源文件。请参阅 <supplement> 元素，以获取该警告的更多详细信息。 |

对代理程序状态问题进行故障诊断

使用以下参考信息可帮助您解决代理程序状态问题：

相关参考

第 115 页的『MFT 常见问题』

Managed File Transfer 网络中可能出现的常见问题。

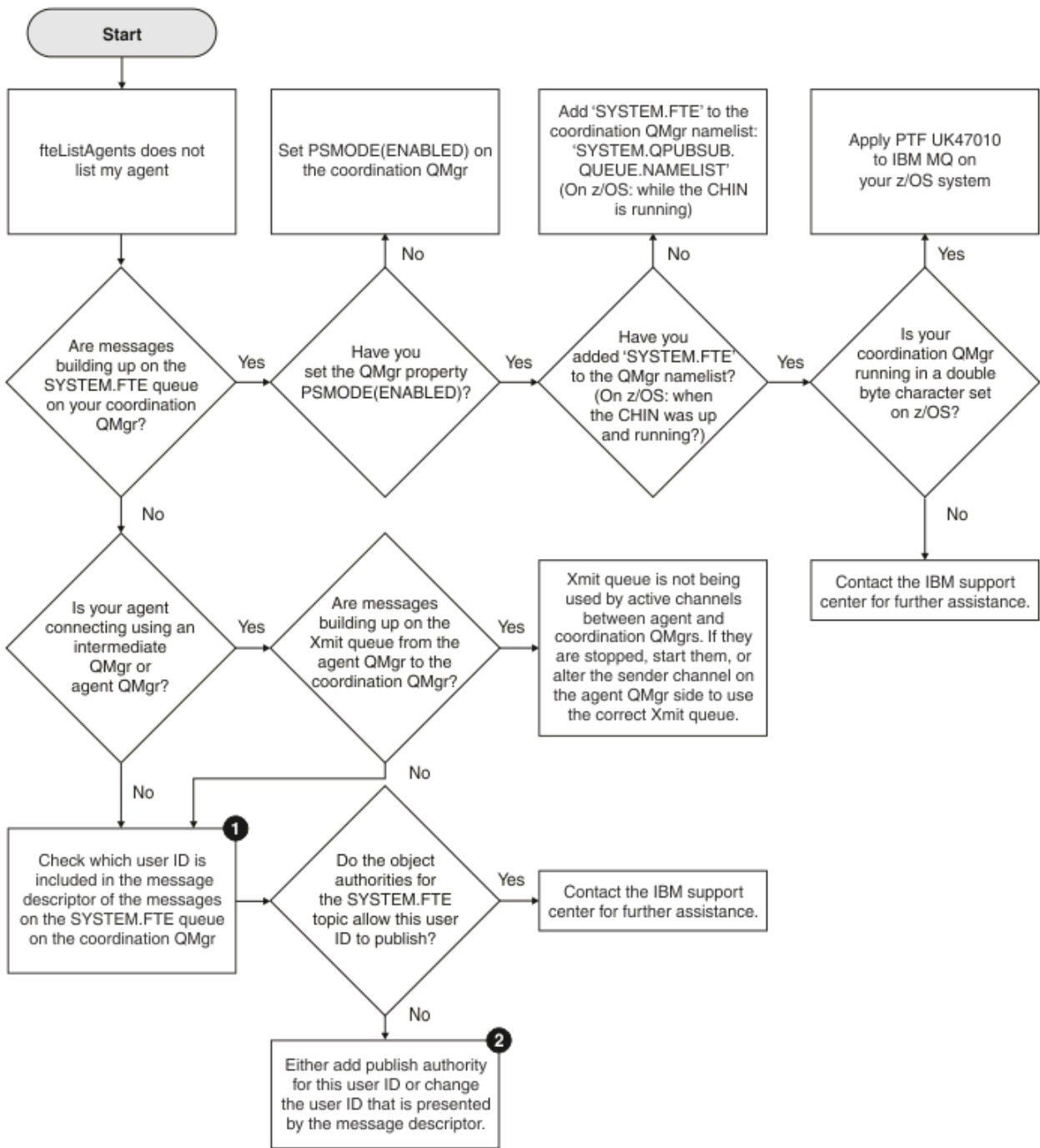
第 78 页的『MFT 的返回码』

Managed File Transfer 命令、Ant 任务和日志消息均提供了用于指示是否已成功完成功能的返回码。

fteListAgents 命令未列出 MFT 代理程序时要执行的操作

如果代理未通过 ***fteListAgents*** 命令列出，或者未显示在 IBM MQ Explorer 中，或者文件传输未显示在 IBM MQ Explorer 的 **传输日志** 中，那么您可以执行许多问题确定步骤来调查原因。

使用以下流程图可帮助您诊断问题并决定接下来采取的操作：



流程图要点：

1. 有关如何检查提供的用户标识的更多信息，请参阅第 119 页的『发布前检查消息』。用户标识必须符合 MQ 用户名的 12 个字符限制。如果用户名多于 12 个字符（例如 Administrator），那么在检查其权限之前将截断该用户名。在使用 Administrator 的示例中，会将以下错误消息添加到队列管理器错误日志中：

AMQ8075: Authorization failed because the SID for entity 'administrato' cannot be obtained.

2. 有关 SYSTEM.FTE 队列，请参阅 [发布 MFT 代理程序日志和状态消息的权限](#)。

如果代理被显示为处于 **UNKNOWN** 状态，该怎么办

您的代理正在运行并成功响应 **ftePingAgent** 命令，且正在正常传输项目。但是，**fteListAgents** 和 **fteShowAgentDetails** 命令以及 IBM MQ Explorer Managed File Transfer 插件报告代理处于 UNKNOWN 状态。

发生此问题的原因

每个代理都会定期将其状态发布到协调队列管理器上的 SYSTEM.FTE 主题。代理发布其状态的频率由以下代理属性控制：

agentStatusPublishRateLimit

由于文件传输状态的变化，代理重新发布其状态的最大速率（以秒为单位）。此属性的缺省值为 30 秒。

agentStatusPublishRateMin

代理发布其状态的最小速率（以秒为单位）。此值必须大于或等于 **agentStatusPublishRateLimit** 属性的值。**agentStatusPublishRateMin** 属性的缺省值为 300 秒（或 5 分钟）。

fteListAgents 和 **fteShowAgentDetails** 命令以及 IBM MQ Explorer Managed File Transfer (MFT) 插件通过这些发布来确定代理的状态。为此，这些命令和插件会执行以下步骤：

1. 连接到协调队列管理器。
2. 预订 SYSTEM.FTE 主题。
3. 接收代理状态发布。
4. 在协调队列管理器上创建临时队列。
5. 将消息放入临时队列，并保存放置时间，以获取协调队列管理器系统上的当前时间。
6. 关闭临时队列。
7. 使用发布中包含的信息以及当前时间来确定代理的状态。
8. 与协调队列管理器断开连接。

如果代理发布状态的时间与当前时间的差值大于以下值，就会认为代理的状态消息是过时的：代理属性 **agentStatusPublishRateMin**（包含在状态消息中）的值加上高级协调队列管理器属性 **agentStatusJitterTolerance** 的值。

缺省情况下，**agentStatusJitterTolerance** 属性的值为 3000 毫秒（3 秒）。

如果将 **agentStatusPublishRateMin** 和 **agentStatusJitterTolerance** 属性设置为其缺省值，那么当代理发布状态的时间与当前时间的差值大于 303 秒（或 5 分 3 秒）时，会将该代理的状态视为已过时。

fteListAgents 和 **fteShowAgentDetails** 命令以及 IBM MQ Explorer MFT 插件会将具有过时状态消息的所有代理都报告为处于 UNKNOWN 状态。

由于下列一个原因，代理的状态发布可能已过时：

1. 运行代理队列管理器的系统与协调队列管理器所在系统之间的系统时间存在显著差异。
2. 代理队列管理器与协调队列管理器之间的通道已停止使用（这会阻止新的状态消息到达协调队列管理器）。
3. 授权问题阻止代理将其状态发布到协调队列管理器上的 SYSTEM.FTE 主题。
4. 代理发生故障。

对问题进行故障诊断

需要执行一些步骤来确定代理的状态为何会被报告为 UNKNOWN：

1. 通过登录代理系统，检查代理是否正在运行。如果代理已停止运行，请调查它不再运行的原因。一旦它再次运行，即检查现在是否正确报告了其状态。
2. 检查协调队列管理器是否正在运行。如果未运行，请将其重新启动，然后使用 **fteListAgents** 或 **fteShowAgentDetails** 命令或者 IBM MQ Explorer MFT 插件来查看现在是否正确报告了代理状态。

3. 如果代理和协调队列管理器正在运行，请在 **fteListAgents** 输出或 IBM MQ Explorer MFT 插件中检查代理的 *Status Age* 值。

此值显示了代理状态消息的发布时间与其处理时间之间的差值。

如果差值：

- 始终比代理属性 **agentStatusPublishRateMin**（包含在状态消息中）的值加上高级协调队列管理器属性 **agentStatusJitterTolerance** 的值略高，请考虑增大 **agentStatusJitterTolerance** 属性的值。这引入了较小的容差，以允许接收和处理状态发布之间存在延迟，并允许代理队列管理器和协调队列管理器系统之间的系统时钟存在差异。
- 比代理属性 **agentStatusPublishRateMin**（包含在状态消息中）的值加上高级协调队列管理器属性 **agentStatusJitterTolerance** 的值超出 10 分钟以上，并且每次检查代理状态时都会持续增加，那么代理的状态消息不会到达协调队列管理器。

在这种情况下，首先要做的就是检查代理队列管理器和协调队列管理器的错误日志，了解是否有任何授权问题阻止代理发布其状态消息。如果日志显示出现授权问题，请确保运行代理进程的用户具有正确的权限，能够将消息发布到协调队列管理器上的 SYSTEM.FTE 主题。

如果队列管理器的错误日志未报告任何授权问题，请确认状态消息未卡在 IBM MQ 网络中。验证用于将消息从代理队列管理器路由至协调队列管理器的所有发送方和接收方通道是否正在运行。

如果这些通道正在运行，请检查与通道相关联的传输队列，以确保状态消息未卡在这些通道上。此外，您还应该检查队列管理器的任何死信队列，确保没有出于某种原因而将状态消息放置在此处。

4. 如果通道正在运行，并且状态消息正在流经 IBM MQ 网络，那么接下来要检查的是队列管理器的已排队发布/预订引擎是否正在提取消息。

fteSetupCoordination 命令 (用于定义协调队列管理器) 为您提供了一些 MQSC 命令，这些命令必须在协调队列管理器上运行，才能配置排队的发布/预订引擎以接收发布。这些命令执行以下步骤：

- 创建 SYSTEM.FTE 主题及其关联的主题字符串。
- 定义名为 SYSTEM.FTE 。
- 通过将队列管理器上的 **PSMODE** 属性设置为 **ENABLED**，启用已排队的发布/预订引擎。
- 修改 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名称列表，由排队的发布/预订引擎使用，以便它包含新 SYSTEM.FTE 队列。

有关这方面的更多信息 (包括需要运行的 MQSC 命令)，请参阅 [fteSetup 协调: 设置协调队列管理器的属性文件和目录](#)。

如果 SYSTEM.FTE 队列，然后您应该检查 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名称列表，并包含该队列的条目。如果缺少该条目，那么排队的发布/预订引擎将不会检测来自代理程序的任何入局状态消息，并且不会处理这些消息。

您还应确保队列管理器上的 **PSMODE** 属性设置为 **ENABLED**，这将打开排队的发布/预订引擎。

5. 如果通道正在运行，并且状态消息正在流经 IBM MQ 网络 并且正在由队列管理器的排队发布/预订引擎从 SYSTEM.FTE 队列中获取，请收集以下跟踪：

- 代理的 IBM MQ MFT 跟踪信息，涵盖的时间段等于代理属性 **agentStatusPublishRateMin** 值的三倍。这会确保在跟踪所涵盖的时间范围内，代理至少发布三条包含其状态的消息。应该使用跟踪规范动态收集跟踪信息：

```
com.ibm.wmqfte.statestore.impl.FTEAgentStatusPublisher,  
com.ibm.wmqfte.utils.AgentStatusDetails,  
com.ibm.wmqfte.wmqiface.AgentPublicationUtils,  
com.ibm.wmqfte.wmqiface.RFHMessageFactory=all
```

注：使用这些字符串输出减少的跟踪量。

有关如何为 IBM MQ for Multiplatforms 上运行的代理启用跟踪的信息，请参阅第 354 页的『[自动收集 Managed File Transfer 代理跟踪信息](#)』。

有关如何为 IBM MQ for z/OS 上运行的代理启用跟踪的信息，请参阅第 361 页的『[动态收集 Managed File Transfer for z/OS 代理跟踪信息](#)』。

- 队列管理器的并发跟踪，用于将状态消息从代理队列管理器路由到协调队列管理器。
- **fteListAgents** 命令的跟踪，涵盖代理被显示为处于 UNKNOWN 状态的时间。应该使用跟踪规范来收集跟踪信息：

```
com.ibm.wmqfte=all
```

有关如何为 IBM MQ for Multiplatforms 上运行的命令启用跟踪的信息，请参阅第 355 页的『在多平台上跟踪 Managed File Transfer 命令』。

有关如何为 IBM MQ for z/OS 上运行的命令启用跟踪的信息，请参阅第 363 页的『跟踪 Managed File Transfer for z/OS 命令』。

收集跟踪后，应将其提供给 IBM 支持人员进行分析。

从命令行查看状态时效

从 IBM MQ 9.1.0 开始，发布的 **Status Age** 信息显示为 **fteListAgents** 和 **fteShowAgentDetails** 命令输出的一部分。

有关更多信息，请参阅 **fteListAgents** 和 **fteShowAgentDetails**。

在 IBM MQ Explorer 中查看状态时效

从 IBM MQ 9.1.0 开始，当您查看代理程序列表并显示个别代理程序属性时，IBM MQ Explorer MFT 插件中提供了 **Status Age** 信息。

相关参考

[fteListAgents](#)

[fteShowAgentDetails](#)

[MFT 代理状态值](#)

[MFT agent.properties 文件](#)

[MFT coordination.properties 文件](#)

当 **ftePingAgent** 超时并报告 **BFGCL0214I** 消息时应执行的操作

ftePingAgent 是随 IBM MQ Managed File Transfer 提供实用命令行实用程序，它使您能够检查代理程序是否可访问以及是否能够对请求作出响应。

命令的工作方式

您可以使用 **ftePingAgent** 命令检查代理程序是否可访问，以及它是否能够处理请求。运行该命令时，它将执行以下步骤：

- 连接到用于 Managed File Transfer (MFT) 拓扑的命令队列管理器。
- 在命令队列管理器上创建一个临时应答队列。
缺省情况下，临时队列有一个以前缀 WMQFTE 开头的名称。但是，您可以通过在 [MFT 命令 .properties 文件](#) 中为安装设置 **dynamicQueuePrefix** 属性更改此值。
- 将 Ping MFT 代理程序请求消息 发送到队列 `SYSTEM.FTE.COMMAND.agent_name`。该请求消息包含临时应答队列的名称。
- 请等待 [MFT 代理程序应答消息](#) 应答消息到达临时应答队列。

代理程序中的其中一个线程是 `CommandHandler`。此线程从代理的 `SYSTEM.FTE.COMMAND.agent_name` 队列获取消息并对其进行处理。

如果此线程接收到包含 Ping MFT 代理请求的消息，那么它会构建 MFT 代理应答消息，并将其发送到命令队列管理器上的临时队列。此消息将通过代理的队列管理器。

一旦消息到达临时队列，**ftePingAgent** 命令就会选取该消息。然后，该命令会在退出之前将类似如下所示的消息写入控制台：

```
BFGCL0213I: agent <agent_name> responded to ping in 0.088 seconds.
```

以下两个图显示了这一流程：

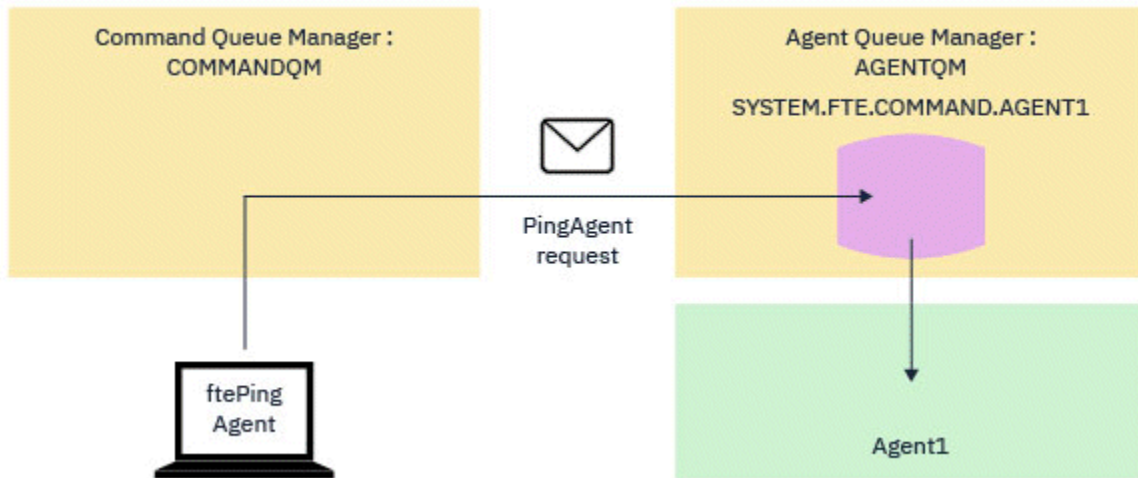


图 1: pingAgent 请求通过命令队列管理器进入代理队列管理器上的 SYSTEM.FTE.COMMAND.agent_name 队列

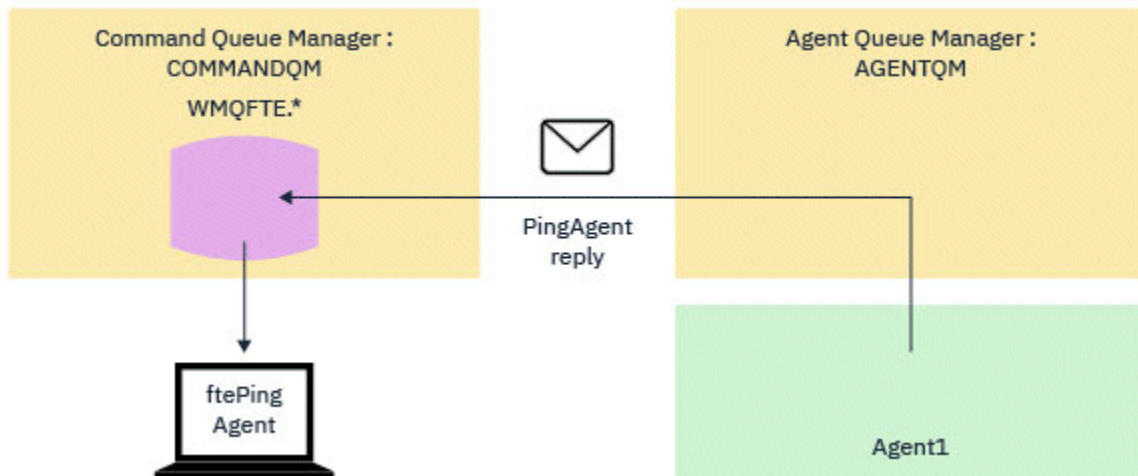


图 2: pingAgent 应答通过代理队列管理器返回到命令队列管理器。

当命令超时时应执行的操作

缺省情况下，**ftePingAgent** 命令会等待 5 秒，以便 MFT 代理程序应答消息到达临时队列。如果应答消息未在 5 秒内到达，那么该命令会将 BFGCL0214I 消息写入控制台。以下消息是一个示例：

```
BFGCL0214I: 代理 AGENT1 在 5 秒后未响应 ping 操作。
```

您可以使用以下步骤来调查应答消息未到达的原因：

- 首先要执行的操作是检查代理程序是否正在运行。如果不是，那么它无法对命令发送的 Ping MFT 代理请求作出响应。
- 如果代理程序正在运行并且忙于处理请求，那么可能需要超过 5 秒才能获取 Ping MFT 代理程序请求并将回复发送回来。

要查看是否应该再次重新运行 **ftePingAgent** 命令，请使用 **-w** 参数指定更长的等待时间间隔。例如，要指定 60 秒的等待时间间隔，请发出以下命令：

```
ftePingAgent -w 60 AGENT1
```

- 如果该命令仍超时，请检查命令队列管理器与代理队列管理器之间通过 IBM MQ 网络的路径。如果路径中的一个或多个通道发生故障，那么 Ping MFT 代理程序请求消息和/或 MFT 代理程序应答消息将停留在某个位置的传输队列上。在此情况下，您应重新启动通道并重新运行 **ftePingAgent** 命令。

如果该命令在您执行上述步骤后仍报告 BFGCL0214I 消息，那么需要跟踪 Ping MFT 代理程序请求和 MFT 代理程序回复消息，因为它们流经 IBM MQ 网络以查看：

- Ping MFT 代理消息是否曾经到达 `SYSTEM.FTE.COMMAND.agent_name` 队列。
- 如果代理从队列中获取消息，并将 MFT 代理应答消息发送回。

要执行此操作，您应该执行以下步骤：

- 在命令和代理队列管理器上启用队列管理器跟踪。
- 使用跟踪规范 `com.ibm.wmqfte=all` 对代理程序动态启用跟踪。

执行此操作的方式取决于运行代理程序的平台。对于正在运行的代理程序：

- IBM MQ for Multiplatforms，请参阅第 354 页的『自动收集 Managed File Transfer 代理跟踪信息』。
- IBM MQ for z/OS，请参阅第 361 页的『动态收集 Managed File Transfer for z/OS 代理跟踪信息』。
- 接下来，使用跟踪规范 `com.ibm.wmqfte=all` 在启用跟踪的情况下运行 **ftePingAgent** 命令。有关跟踪命令的信息：
 - IBM MQ for Multiplatforms，请参阅第 355 页的『在多平台上跟踪 Managed File Transfer 命令』。
 - IBM MQ for z/OS，请参阅第 363 页的『跟踪 Managed File Transfer for z/OS 命令』。

当该命令超时时，停止代理程序跟踪和队列管理器跟踪。然后，应该将代理程序和队列管理器跟踪以及命令中的跟踪提供给 IBM 支持人员进行分析。

对受管传输问题进行故障诊断

使用以下参考信息来帮助您解决受管传输的问题：

相关参考

第 88 页的『当传输没有完成时应如何处理』

如果传输没有完成，您可以执行几个问题确定步骤来调查原因。

第 92 页的『在您认为文件传输卡住时应该怎么办』

在高负荷的系统上或者在源和目标代理之间发生网络问题时，传输可能会偶尔显示为卡住，进入排队或恢复状态。导致此情况的因素有很多。

第 92 页的『如果目标队列是集群队列或集群队列的别名，那么该执行什么操作？』

使用 Managed File Transfer 将文件传输到队列中时，如果使用的目标队列是集群队列或集群队列的别名，那么您将收到原因码 2085 或 2082。从 IBM WebSphere MQ 7.5.0 Fix Pack 4 开始，将属性 `enableClusterQueueInputOutput` 设置为 `true` 可解决此问题。

第 93 页的『当已调度的文件传输未运行或被延迟时该怎么做』

如果您具有到期未运行或被延迟的已调度传输，那么原因可能是代理正在处理其命令队列中的命令。由于代理很繁忙，因此未检查已调度传输，从而未运行已调度传输。

第 93 页的『传输 IBM i 保存文件时可能发生的错误』

如果您使用 Managed File Transfer 多次传输同一个 IBM i 保存文件，那么传输可能失败。

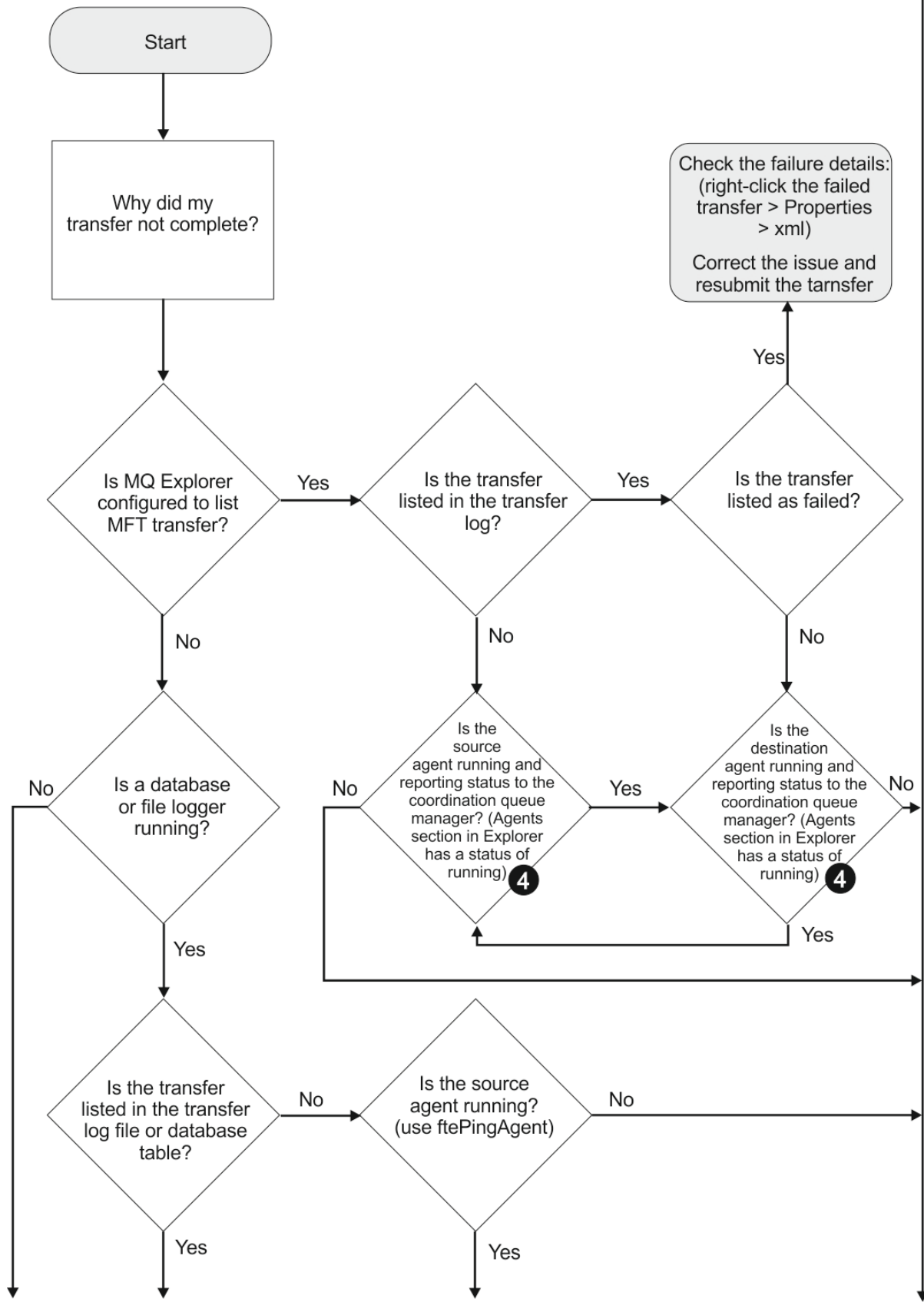
第 93 页的『当受管传输失败并返回 BFGIO0341E 错误时应执行的操作』

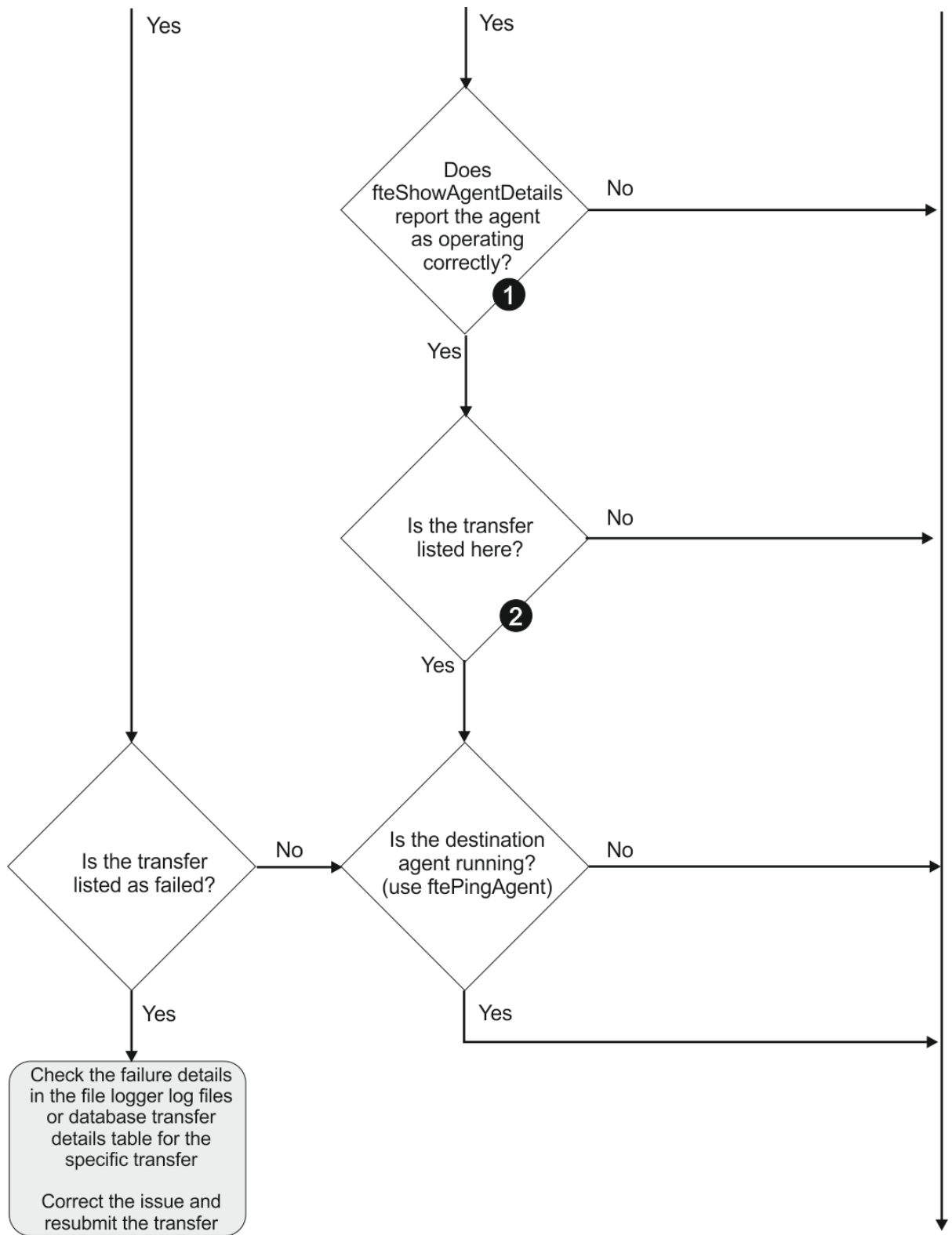
如果受管传输正在将文件传输到由外部进程监视的位置，那么该受管传输可能会失败并返回以下错误：
BFGIO0341E：将临时文件 `destination_filename.part` 重命名为 `destination_filename` 失败，因为该临时文件不存在。这是由于受管传输的目标代理在写入目标文件时使用临时文件的方式造成的。

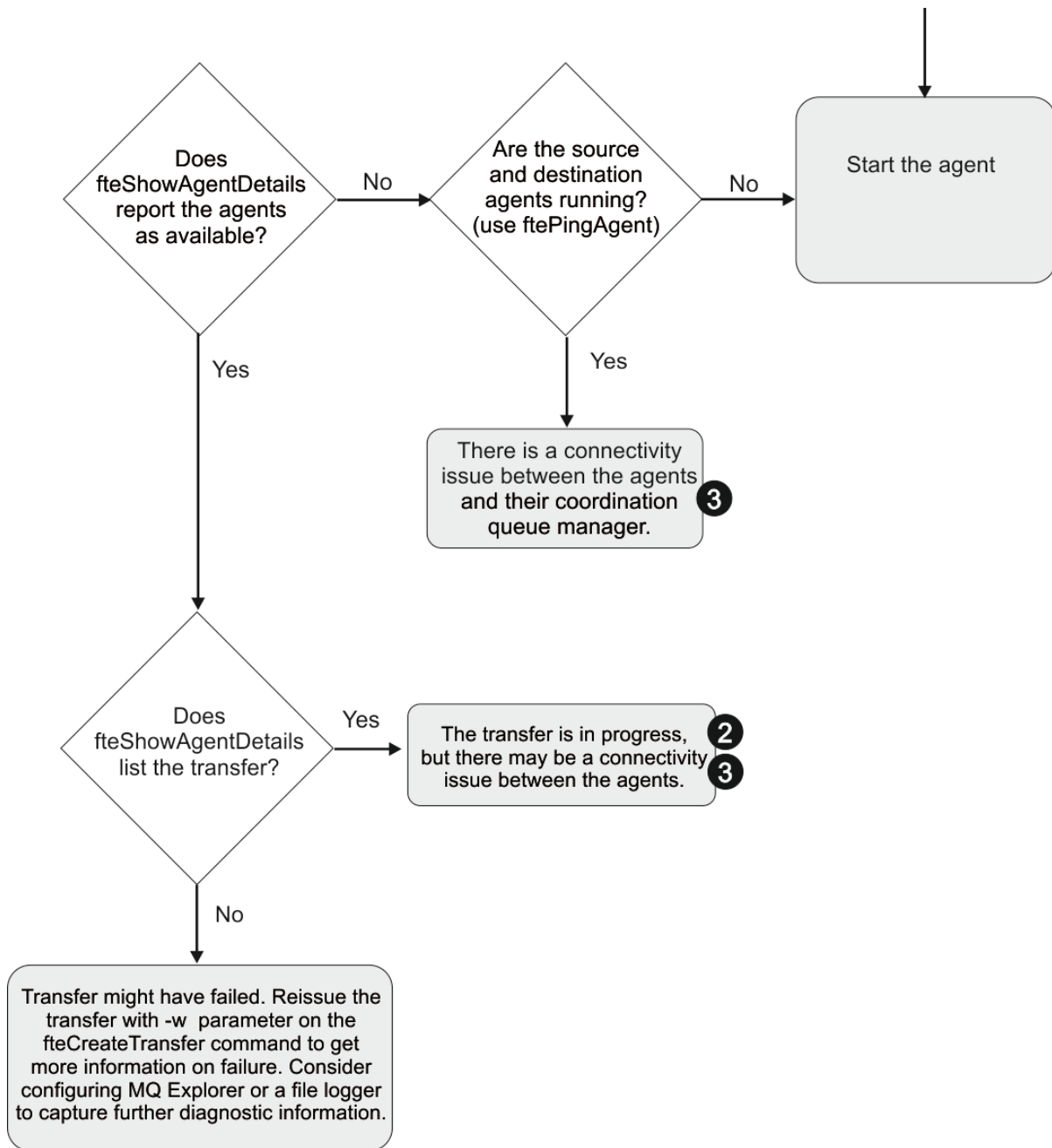
当传输没有完成时应如何处理

如果传输没有完成，您可以执行几个问题确定步骤来调查原因。

使用以下流程图可帮助您诊断问题并决定接下来采取的操作：







流程图要点：

1. 检查代理的 `output0.log` 中是否有错误。如果代理报告自己已成功启动，但 IBM MQ Explorer 或 **fteShowAgentDetails** 都没有报告该代理正在运行，请检查代理队列管理器与协调队列管理器之间的连接。可能是因为队列管理器间通道不可用。
2. 如果源代理将传输标识列示为 `In progress` 传输，但目标代理未列示，那么源队列管理器与目标队列管理器之间可能存在连接问题。在 `command.properties` 文件中，从目标代理机器向使用目标代理队列管理器作为命令队列管理器的源代理使用 **ftePingAgent** 命令。您也可以按相反方向运行此命令，即从源到目标。
3. 如果源和目标代理都将传输标识列示为 `In progress`，那么这表明自传输启动以来，源和目标队列管理器之间存在连接问题。在 `command.properties` 文件中，从目标代理机器向使用目标代理队列管理器作为命令队列管理器的源代理使用 **ftePingAgent** 命令。您也可以按相反方向运行此命令，即从源到目标。
4. 如果已运行过此循环，请检查以下任一声明是否与您的情况相关：

- 源代理和目标代理都报告为 **Running**，但未列出任何传输。传输请求未到达代理命令队列，或者代理虽然报告为 **Running**，但不再监视命令队列。请检查源代理的 `output0.log` 中是否有错误。从发送传输的同一机器向源代理使用 **ftePingAgent** 命令，以验证命令队列管理器和代理队列管理器之间的连接，以及代理是否正在为命令队列提供服务。
- 源和目标代理都报告为 **Running**，并且传输列示为 **In progress, recovering**。在 `command.properties` 文件中，从目标代理机器向使用目标代理队列管理器作为命令队列管理器的源代理使用 **ftePingAgent** 命令。您也可以按相反方向运行此命令，即从源到目标。

在您认为文件传输卡住时应该怎么办

在高负荷的系统上或者在源和目标代理之间发生网络问题时，传输可能会偶尔显示为卡住，进入排队或恢复状态。导致此情况的因素有很多。

请完成以下检查以确定问题原因：

1. 您可以使用 **ftePingAgent** 命令，或者在 IBM MQ Explorer 代理面板中右键单击代理名称并选择 **Ping**，以检查源和目标代理是否处于活动状态并在响应新请求。查看代理日志以查看当前是否存在网络连接问题。
2. 检查目标代理是否以最高能力运行。可能存在大量源代理都在请求将文件传输到相同的目标代理。将 **fteShowAgentDetails** 命令与 **-v** (详细) 参数配合使用，或者在 IBM MQ Explorer 代理程序面板中右键单击代理程序名称并选择 **属性** 以查看代理程序的当前传输活动。如果正在运行的目标传输的数量等于或接近目标传输的代理最大数量，那么这可能是某些源代理传输显示为卡住的原因。
3. 如果联系协议文件服务器时发生问题，那么进出协议网桥代理的传输将进入恢复状态。查看代理日志以查看当前是否存在连接问题。
4. 代理将按优先级顺序处理传输。因此，在装入系统中，低优先级传输可能仍保持排队状态一段时间，而将装入具有较高优先级传输的代理。最后，如果该传输已排了一段时间队，那么将启动低优先级传输，即使存在新的较高优先级的传输。

如果目标队列是集群队列或集群队列的别名，那么该执行什么操作？

使用 Managed File Transfer 将文件传输到队列中时，如果使用的目标队列是集群队列或集群队列的别名，那么您将收到原因码 2085 或 2082。从 IBM WebSphere MQ 7.5.0 Fix Pack 4 开始，将属性 `enableClusterQueueInputOutput` 设置为 `true` 可解决此问题。

发生此问题的原因

当 **-dq** 上无显式队列管理器名称时，会将目标代理的队列管理器名称附加到 **-dq** 参数的队列名称。由于在连接到没有该本地集群队列的集群 MQ `queueManager` 时，无法在 MQOPEN 调用中指定 `queueManager` 对象，由此产生原因码 2085 或 2082。

避免此问题

1. 在队列管理器上创建集群队列。
2. 设置指向集群队列的远程队列定义。

示例

本示例使用远程队列定义。

配置：

- 源代理：SAGENT
- 源代理队列管理器：SQM
- 目标代理：DAGENT
- 目标代理队列管理器：DQM
- 传输的目标队列是队列管理器 SQM 上的 CQ6

要针对 SQM 中的集群队列 CQ6 定义 DQM 上的远程队列定义 Q6_SQM（假设已在 SQM 中定义集群队列 CQ6），请在 DQM 队列管理器上发出 MQSC 命令：

```
define qremote(Q6_SQM) rname(CQ6) rqmname(SQM) xmitq(SQM)
```

注: rname 指向集群队列。

您现在可以传输至队列。例如:

```
fteCreateTransfer -sa SAGENT -sm SQM -da DAGENT -dm DQM -dq Q6_SQM /tmp/single_record.txt
```

当已调度的文件传输未运行或被延迟时该怎么做

如果您具有到期未运行或被延迟的已调度传输,那么原因可能是代理正在处理其命令队列中的命令。由于代理很繁忙,因此未检查已调度传输,从而未运行已调度传输。

要解决该问题,请使用以下步骤之一:

- 在 `agent.properties` 文件中配置 `maxSchedulerRunDelay` 属性,以设置代理等待检查调度传输的最大时间间隔(以分钟计)。设置该属性可确保即使代理很繁忙,代理也能保持检查已调度传输。有关该属性的更多信息,请参阅 `MFT agent.properties` 文件。
- 或者,请使用资源监视器代替已调度传输。资源监视器的工作方式与已调度传输不同,不受繁忙代理的影响。例如,如果要获取目标系统上的最新文件,资源监视器将减少网络流量。这是因为只有当新版本可用时才会传输文件,而不是自动传输文件。但是,在协议网桥代理或 Connect:Direct 网桥代理上不支持资源监视。

有关更多信息,请参阅[监视 MFT 资源](#)。

IBM i 传输 IBM i 保存文件时可能发生的错误

如果您使用 Managed File Transfer 多次传输同一个 IBM i 保存文件,那么传输可能失败。

Managed File Transfer 可能产生以下错误中的一个或全部:

- ```
BFGII0003E: Unable to open file "/qsys.lib/library.lib/SAVF.FILE"
for reading
```
- ```
BFGII0082E: A file open for read failed due to a Java IOException
with message text "Sharing violation occurred"
```

如果向 MFT 代理发出多个请求来传输同一个 IBM i 保存文件,那么可能会发生这些错误。如果希望同时多次传输同一个保存文件,那么必须使用多个源代理。请对每个并发传输使用不同源代理。

要使单个源代理来多次传输同一个保存文件,您必须等到前一个传输请求完成,然后再提交每个新传输请求。

当受管传输失败并返回 BFGI00341E 错误时应执行的操作

如果受管传输正在将文件传输到由外部进程监视的位置,那么该受管传输可能会失败并返回以下错误:

BFGI00341E: 将临时文件 `destination_filename.part` 重命名为 `destination_filename` 失败,因为该临时文件不存在。这是由于受管传输的目标代理在写入目标文件时使用临时文件的方式造成的。

目标代理使用临时文件的方式

缺省情况下,当进行受管文件传输时,目标代理将执行以下步骤:

- 创建名为 `destination_filename.part` 的临时文件。
- 锁定该临时文件。
- 从源代理接收文件数据时,会将这些数据写入到该临时文件。
- 在接收并写入所有文件数据后,解锁该临时文件。
- 将临时文件从 `destination_filename.part` 重命名为 `destination_filename`。

如果受管传输进入恢复状态，那么目标代理可能会创建名为 `destination_filename.partnumber` 的临时文件。然后，目标代理会将文件数据写入此文件，而不是名为 `destination_filename.part` 的文件。

如果已存在临时文件名 `destination_filename.partnumber`，那么目标代理会尝试创建名为 `destination_filename.part(number + 1)` 的新临时文件。如果已存在该文件，那么目标代理会尝试创建名为 `destination_filename.part(number + 2)` 的临时文件，以此类推，直到能够成功创建该文件。在代理尝试创建临时文件 `destination_filename.part1000` 并失败的情况下，它会直接写入目标文件，并且不会使用临时文件。

当受管传输完成时，目标代理会删除名为 `destination_filename.partnumber` 的所有临时文件，因为会假定这些文件是由代理在受管传输期间创建的。

注：如果代理属性 `doNotUseTempOutputFile` 设置为值 `true`，那么目标代理不使用临时文件。而是直接写入到目标文件。有关 `doNotUseTempOutputFile` 属性的更多信息，请参阅 [MFT agent.properties](#) 文件。

发生此问题的原因

如果目标代理尝试重命名临时文件，但发现该文件不再存在，那么会生成 BFGIO0341E 错误。导致此问题的典型场景如下：

- 已在目标文件系统中设置了一个登台目录。
- 外部进程配置为监视该登台目录，并将其找到的任何文件移到新位置。
- 目标代理会在登台目录中创建并锁定临时文件 `destination_filename.part`。
- 目标代理会将文件数据写入到该临时文件。
- 将所有文件数据写入到该临时文件后，目标代理会解锁该文件。
- 外部进程找到该临时文件，并将其移到新位置。
- 目标代理尝试重命名该临时文件，发现该文件已不存在。因此，该传输项被标记为**失败**，并返回 BFGIO0341E 错误。

避免此问题

可以通过两种方法来防止发生 BFGIO0341E 错误：

- 由目标代理编写的临时文件始终以 `.part` 或 `.partnumber` 后缀结尾。如果可以将外部进程配置为忽略这些文件（而不是移动它们），那么当目标代理执行重命名操作时，这些文件仍将存在于目标目录中。
- 也可以配置目标代理，使其不使用临时文件，而是直接写入到目标文件。只有在所有文件数据都已写入到目标文件时，目标文件才会解锁，此时外部进程可以选取目标文件。

要将目标代理配置为直接写入到目标文件，请设置代理属性 `doNotUseTempOutputFile=true`。有关此属性的更多信息，请参阅 [MFT agent.properties](#) 文件。

对协议网桥代理问题进行故障诊断

使用以下参考信息帮助您解决协议网桥代理的问题：

相关参考

第 94 页的『[当协议网桥代理报告找不到某个文件时应如何处理](#)』

当协议网桥代理报告协议网桥所连接的 SFTP 或 FTP 服务器返回 `File not found` 错误消息时，此消息可能表示发生了许多不同的错误情况之一。

当协议网桥代理报告找不到某个文件时应如何处理

当协议网桥代理报告协议网桥所连接的 SFTP 或 FTP 服务器返回 `File not found` 错误消息时，此消息可能表示发生了许多不同的错误情况之一。

以下可能的场景可能导致 SFTP 或 FTP 服务器返回 `File not found` 错误。

- 文件不存在。检查托管 SFTP 或 FTP 服务器的系统上是否存在您尝试传输的文件。

- 文件路径不存在。检查托管 SFTP 或 FTP 服务器的系统上是否存在该文件路径。检查您是否已在传输请求中正确输入文件路径。如有必要，请纠正文件路径，然后重新提交传输请求。
- 文件被另一个应用程序锁定。检查文件是否被另一个应用程序锁定。等待直至文件不再被锁定，然后重新提交传输请求。
- 文件许可权不允许读取文件。检查文件是否具有正确的文件许可权。如有必要，请更改文件许可权，然后重新提交传输请求。
- SFTP 或 FTP 服务器使用虚拟的根路径。如果在传输请求中指定了相对文件路径，那么该协议网桥代理会尝试根据用于登录协议服务器的主目录将相对路径转换为绝对文件路径。Managed File Transfer 协议网桥代理只可以支持允许通过绝对文件路径访问文件的 SFTP 或 FTP 服务器。该协议网桥代理程序不支持这些允许仅根据当前目录访问文件的协议服务器。

相关参考

[协议网桥](#)

对资源监视器问题进行故障诊断

使用以下参考信息来帮助您诊断 Managed File Transfer 资源监视器的问题:

相关概念

[第 99 页的『资源监视器报告 BFGDM0107W 消息时要执行的操作』](#)

配置为轮询目录或队列的资源监视器将查找与指定的触发条件匹配的项，并将受管传输提交到其关联的代理以进行处理。监视器定期将 BFGDM0107W 消息写入代理程序的事件日志 (output0.log)。

相关参考

[第 95 页的『当 MFT 目录资源监视器未触发文件时该怎么做』](#)

目录资源监视器对与触发器规范匹配的文件的目录进行轮询。对于与触发器规范匹配的每个文件，代理都会生成传输请求。提交请求时，会忽略触发文件，直至文件发生更改。

[第 96 页的『用于配置 MFT 资源监视器以避免重载代理的指南』](#)

您可以配置 Managed File Transfer 资源监视器的属性和参数值以减少代理上的负载。减少代理上的负载可提高该代理的性能。有多个设置可供您使用，您可能需要通过反复试验来找到适合您系统配置的最佳设置。

[第 98 页的『如果队列资源监视器启动的传输所创建的目标文件包含错误数据，那么该执行哪些操作?』](#)

您可以创建资源监视器来监视队列，并将队列上的消息或一组消息传输到文件。可以通过使用消息上的 MQMD 消息述符描或组中的第一个消息来指定文件名。如果消息到文件的传输失败，并且消息或组留在队列上，那么下次触发监视器时，可能导致创建的文件中包含错误数据。

[第 98 页的『当变量替换导致多个文件传输到单个文件名时该怎么做』](#)

对于 Managed File Transfer，如果要监视某个目录并将多个文件从源位置传输到目标位置，并且要使用 \${FileName} 变量替换，那么必须测试变量替换的结果。需要对结果进行测试的原因在于：使用变量替换可能导致调用意外的文件传输命令组合。

当 MFT 目录资源监视器未触发文件时该怎么做

目录资源监视器对与触发器规范匹配的文件的目录进行轮询。对于与触发器规范匹配的每个文件，代理都会生成传输请求。提交请求时，会忽略触发文件，直至文件发生更改。

未触发文件的可能原因

1. 目录资源监视器发现一个与触发器规范匹配的文件，但生成的传输请求无效并且代理无法处理请求。可能包含以下原因：
 - 目标代理无效
 - 缺少目标代理
 - 传输由程序调用取消

在所有这些示例中，即使传输失败，目录资源监视器仍将触发文件标记为已处理并忽略文件。
2. 文件超出了资源监视器触发器规范的范围。可能包含以下原因：
 - 触发器模式不正确
 - 正在监控不正确的目录

- 文件许可权不足
- 无法连接至远程文件系统

为什么文件会触发第二次传输

触发器文件可以出于以下原因而生成 Managed File Transfer 传输请求：

- 检测到之前不存在的触发器文件。
- 更新了触发器文件，导致最后修改日期发生变化。

第二次触发的潜在场景包括：

- 移除和替换了文件。
- 文件被一个应用程序锁定，然后被另一个应用程序解锁。
- 监控文件系统发生故障。例如，如果出现网络连接故障，则可能导致文件显示为被移除和替换。
- 文件目录被另一个应用程序更新，导致最后修改日期发生变化。

用于对代理的所有资源监视器设置参考级别输出的命令

V 9.1.0

在此示例中，将监视所有资源监视器，因为您尚未指定特定资源监视器的名称。代理程序的名称为 AGENT1。

```
fteSetAgentLogLevel -logMonitor=info AGENT1
```

请参阅 [fteSetAgentLogLevel](#) 以获取 **logMonitor** 参数的详细信息以及有关如何使用不同选项的示例。

代理的所有资源监视器的参考级别输出示例

V 9.1.0

```
=====
[21/04/2017 11:08:49:367 IST] BFGUT0036I: Resource monitor event log level has changed to "info" for all resource monitors of
this agent.
=====

=====
Date          Time          Thread ID    Monitor Name    Event
Description
=====
[21/04/2017 11:08:51:842 IST] 00000023    QMON           Monitor Started  Resource Monitor Started
[21/04/2017 11:08:51:844 IST] 00000025    QMON           Start Poll      New poll cycle started
[21/04/2017 11:08:51:924 IST] 00000023    MON1          Monitor Started  Resource Monitor Started
[21/04/2017 11:08:51:925 IST] 00000026    MON1          Start Poll      New poll cycle started
[21/04/2017 11:08:52:029 IST] 00000026    MON1          End Poll        Poll cycle completed in 105
milli seconds. Trigger items matched [ 0 ]
[21/04/2017 11:08:52:055 IST] 00000025    QMON           End Poll        Poll cycle completed in 212
milli seconds. Trigger items matched [ 0 ]
[21/04/2017 11:09:51:840 IST] 00000025    QMON           Start Poll      New poll cycle started
[21/04/2017 11:09:51:875 IST] 00000025    QMON           End Poll        Poll cycle completed in 34
milli seconds. Trigger items matched [ 0 ]
[21/04/2017 11:09:51:924 IST] 00000026    MON1          Start Poll      New poll cycle started
[21/04/2017 11:09:51:969 IST] 00000026    MON1          End Poll        Poll cycle completed in 45
milli seconds. Trigger items matched [ 0 ]
[21/04/2017 11:10:51:840 IST] 00000025    QMON           Start Poll      New poll cycle started
[21/04/2017 11:10:51:924 IST] 00000026    MON1          Start Poll      New poll cycle started
[21/04/2017 11:10:51:962 IST] 00000025    QMON           End Poll        Poll cycle completed in 121
milli seconds. Trigger items matched [ 0 ]
[21/04/2017 11:10:51:963 IST] 00000026    MON1          End Poll        Poll cycle completed in 39
milli seconds. Trigger items matched [ 0 ]
[21/04/2017 11:10:55:063 IST] 00000041    MON1          Monitor Stopped Resource Monitor Stopped
[21/04/2017 11:10:55:079 IST] 00000041    QMON          Monitor Stopped Resource Monitor Stopped
=====
```

相关参考

V 9.1.0

[fteSetAgentLogLevel](#)

用于配置 MFT 资源监控器以避免重载代理的指南

您可以配置 Managed File Transfer 资源监视器的属性和参数值以减少代理上的负载。减少代理上的负载可以提高该代理的性能。有多个设置可供您使用，您可能需要通过反复试验来找到适合您系统配置的最佳设置。

资源监视概述

资源监视器轮询目录或队列时，代理将完成以下阶段：

- 查找所有与触发模式匹配的文件（例如，目录中的所有 *.txt 文件）。或查找队列上所有完整的消息组。
- 确定哪些文件是新文件，哪些文件已发生更改，或者确定队列上的哪些组是新组。
- 对与前两个阶段中的条件匹配的文件或组启动传输。
- 添加到已传输文件和组的列表中，这样不会再次传输它们直至其发生更改。

对于目录监视器，源目录中的文件越多，触发模式越广泛，代理必须进行解析并与已传输文件列表进行比较的文件列表也就越长。

对于队列监视器，队列上的组越多，针对已传输组列表，代理必须比较的组列表就越大。

考虑以下关键设置：

- 使用代理属性 **monitorMaxResourcesInPoll** 设置代理每次轮询时包含的最大文件或组数。使用该参数限制轮询时间间隔内的传输数。这还表示，在为该数量的文件或组启动传输之前，代理需要执行较少的解析。目录监视器或队列监视器下次轮询时，代理将包含下一组文件或组。代理属性 **monitorMaxResourcesInPoll** 在 IBM WebSphere MQ File Transfer Edition 7.0.4 及更高版本中可用，对于较早版本的 IBM WebSphere MQ File Transfer Edition，该属性用作 APAR IC78011 的临时修订。
- 创建目录监视器时，请确保所配置的传输定义的源处置为 **delete**。设置该处置意味着，文件传输完成后，会将其从受监视的目录中移除并且代理不再将其保留在内部列表中。
- 创建目录监视器时，请在 **fteCreateMonitor** 命令中使用 **-rl** 参数来限制代理必须递归的目录级别数。使用该参数表示不必扫描较低级别的目录。

创建资源监视器时的更多注意事项

资源监视器轮询过程将耗用代理资源。增大监视器的轮询时间间隔将减少代理上的负载。但是，轮询时间间隔的设置必须均衡，以防止每个轮询时间间隔内生成过多的传输。为资源监视器设置轮询时间间隔时，请考虑以下内容：

- 将文件放入目录或将组放入队列后，需要以多快的速度启动传输。
- 将文件放入目录或将组放入队列的速率。
- 最大代理传输速率。代理必须能够处理监视器生成的所有传输。

使用 **fteCreateMonitor** 命令来创建资源监视器时，通过指定 **-pi**（轮询时间间隔）和 **-pu**（轮询时间间隔单位）参数指定轮询时间间隔。您可能需要试验以确定适合您配置的最佳设置。

可提高运行资源监视器的高负载代理的稳定性的选项是降低代理属性 **maxSourceTransfers** 的值。通过此选项，代理将在资源监视器和传输文件之间分配其处理时间。代理属性 **maxSourceTransfers** 的值越高，传输文件所耗用的处理时间就越多，资源监视器可用的处理时间就越少。如果降低代理属性 **maxSourceTransfers** 的值，那么代理将并行执行较少的传输，但应具有足够的处理时间来轮询其资源监视器。如果降低该代理属性的值，那么应考虑增加代理属性 **maxQueuedTransfers** 的值，因为排队传输数可能会增加。

如果在优化监视器后发现一些传输进入恢复状态，请考虑增加代理超时值。与目标代理协商启动传输时，代理上的重负载可能意味着传输超时。该超时将导致传输进入恢复状态并延迟完成传输。代理属性 **maxTransferNegotiationTime** 指定源代理等待目标代理响应的的时间。如果超出该时间，那么传输将进入恢复状态。该属性的缺省值为 30000 毫秒（30 秒）。增加该属性的值（例如，增加至 300000 毫秒（5 分钟））可以使传输在不超时的情况下继续并避免进入恢复状态。

相关任务

[监视 MFT 资源](#)

[使用传输定义文件](#)

相关参考

fteCreateMonitor: [创建 MFT 资源监视器](#)

如果队列资源监视器启动的传输所创建的目标文件包含错误数据，那么该执行哪些操作？

您可以创建资源监视器来监视队列，并将队列上的消息或一组消息传输到文件。可以通过使用消息上的 MQMD 消息描述符或组中的第一个消息来指定文件名。如果消息到文件的传输失败，并且消息或组留在队列上，那么下次触发监视器时，可能导致创建的文件中包含错误数据。

发生此问题的原因

1. 消息到文件的传输失败，并且消息或组留在队列上。
2. 新的消息或组到达队列。
3. 新的消息或组触发资源监视器。
4. 资源监视器创建一个新传输，此传输使用新消息或组上的 MQMD 消息描述符以及队列上的第一个消息或组中的数据。
5. 创建的文件中包含错误数据。

避免此问题

要避免遇到此问题，必须使用 **fteCreateTransfer** 命令手动创建传输定义文件，并编辑文件的 <queue> 元素，以包含属性 `groupId="${GROUPID}"`。然后使用 **fteCreateMonitor** 命令来提交传输定义文件。

示例

在此示例中：源代理（也是监视代理）名为 AGENT_MON；目标代理名为 AGENT_DEST；目标文件名为 /out/files/\${WMQFTEFileName}。此示例要求消息已设置了 MQMD 消息描述符 WMQFTEFileName。正在监视的队列为 LIVE_QUEUE。

1. 通过运行以下命令来创建传输定义文件：

```
fteCreateTransfer -sa AGENT_MON -da AGENT_DEST -df "/out/files/${WMQFTEFileName}"  
-de error -gt /tmp/TransferDefinition1.xml -sqgi -sq LIVE_QUEUE
```

将生成传输定义文件 /tmp/TransferDefinition1.xml。

2. 编辑 <queue> 元素，以包含属性 `groupId="${GROUPID}"`。将行

```
<queue useGroups="true">LIVE_QUEUE</queue>
```

到

```
<queue useGroups="true" groupId="${GROUPID}">LIVE_QUEUE</queue>
```

该属性是必需的，以便传输从队列中（而非队列上的第一个组或消息中）读取触发传输的组或消息。

3. 通过运行以下命令来创建监视器：

```
fteCreateMonitor -ma AGENT_MON -mq LIVE_QUEUE -mn QueueMon1 -mt /tmp/TransferDefinition1.xml  
-tr completeGroups -dv WMQFTEFileName=UNKNOWN
```

监视器会每隔 60 秒对队列进行一次轮询，以查看是否有新的组或消息到达队列。

当变量替换导致多个文件传输到单个文件名时该怎么做

对于 Managed File Transfer，如果要监视某个目录并将多个文件从源位置传输到目标位置，并且要使用 \${FileName} 变量替换，那么必须测试变量替换的结果。需要对结果进行测试的原因在于：使用变量替换可能导致调用意外的文件传输命令组合。

要确定是否发生此问题，请查看那些看似传输了多个文件但实际只有一个文件到达目标的案例。您可能在文件传输日志中看到一些错误，这些错误表明有多个文件尝试传输到相同目标文件名但传输到相同文件名失败。

发生此问题的原因

当 MFT 目录监视器处理多个文件时，将针对监视器在受监视目录中找到的每个文件都运行任务 XML。如果仅在 XML 任务文件的目标中指定 `${FileName}`，而在源中未指定，那么将针对每个文件多次调用传输（针对每个文件名组合各调用一次）。

例如：

```
<source disposition="delete" recursive="false">
  <file>e:\temp</file>
</source>
<destination exist="overwrite" type="file">
  <file>s:\outdir\${FileName}</file>
</destination>
```

避免此问题

如果要在源和目标中使用 `${FileName}` 变量替换，并且期望相同文件名的变体到达目标，请务必在任务 XML 定义的源和目标中同时指定 `${FileName}`。

以下示例从 `e:\temp\<filename>` 获取文件，并将其传输到 `s:\outdir\<filename>.out`：

```
<source disposition="delete" recursive="false">
  <file>e:\temp\${FileName}</file>
</source>
<destination exist="overwrite" type="file">
  <file>s:\outdir\${FileName}.out</file>
</destination>
```

相关任务

[使用变量替换定制 MFT 任务](#)

相关参考

示例：[变量替换](#)

资源监视器报告 BFGDM0107W 消息时要执行的操作

配置为轮询目录或队列的资源监视器将查找与指定的触发条件匹配的项，并将受管传输提交到其关联的代理以进行处理。监视器定期将 BFGDM0107W 消息写入代理程序的事件日志 (output0.log)。

以下文本显示了典型的 BFGDM0107W 消息：

```
BFGDM0107W: 监视器 MONITOR1 在
轮询时间间隔已超过 maxSource 传输代理属性值的两倍
并且代理属性 monitorMaxResourcesInPoll 设置为其缺省值 -1。
```

发生此警告的原因

每个代理都有许多传输槽，用于保存有关当前正在进行的受管传输和受管调用以及当前积压的受管传输和受管调用请求的详细信息。有关如何使用这些插槽的更多信息，请参阅 [MFT 代理如何将源传输插槽分配给新请求](#)。

缺省情况下，监视器针对轮询期间触发的每个项提交一个任务（即单个受管传输或受管调用请求）。例如，如果资源监视器已配置为轮询源队列以查找完整消息组或不在组中的个别消息，那么如果监视器发现：

- 在轮询期间，队列上的 10 条消息或完成消息组，它会向代理提交 10 个任务（或受管传输请求）。
- 200 条消息或在轮询期间完成队列上的消息组，它会向代理提交 200 个任务（或受管传输请求）。

监视器包含一些逻辑，用于将它们轮询期间提交给代理的任务数与代理具有的源传输槽数（由代理属性 **maxSourceTransfers** 指定）进行比较。如果任务数大于源传输槽数的两倍，那么监视器会将 BFGDM0107W 消息写入代理的事件日志。这让您知道它已向代理程序提交了大量任务，其中一半以上正在处理代理程序的积压任务。

返回到前面的示例，在此示例中，监视器在单次轮询期间找到 200 条消息，并假定在监视器向代理程序提交 200 个任务时，所讨论的代理程序的 **maxSourceTransfers** 属性设置为缺省值 25：

- 分配了 25 个源传输槽，代理将开始直接处理这些传输槽。

- 其余 175 个已分配排队的传输槽; 这些传输槽将转至代理的待办事项, 以便在将来的某个时间进行处理。

在代理的待办事项上进行大量受管传输会占用诸如内存之类的资源, 因此可能会影响代理的性能。因此, 最好尝试将占用排队传输时段的受管传输或受管呼叫的数目保持在尽可能低的数目。

如何防止发生警告

可以帮助您的一点是 BFGMD0107W 消息中提到的 **monitorMaxResourcesInPoll** 属性。这是一个代理属性, 它适用于代理程序中运行的所有资源监视器, 并限制在一次轮询期间监视器触发的项数。该属性的缺省值为 -1, 这意味着监视在轮询中找到的每个项的触发器, 并为每个项提交一个任务。

当该属性设置为 -1 以外的内容时, 监视器在触发该多个项时停止扫描资源。这意味着监视器会将工作以小块的形式发送到代理程序, 而不是一次完成大量工作。

例如, 如果 **monitorMaxResourcesInPoll** 设置为 25, 那么一旦监视器找到与其触发条件匹配的 25 个新项, 它将停止其当前轮询并向代理程序提交 25 个任务。

更改 **monitorMaxResourcesInPoll** 时, 要考虑的另一个事项是增加监视器的轮询时间间隔。理想情况下, 如果资源监视器将某些任务提交给代理程序, 那么应该允许大多数任务 (如果不是全部) 在启动新一轮询之前完成, 并可能使代理程序执行更多工作。这也有助于降低代理程序上的总体负载, 并且可以提高其吞吐量。

示例

假设您有一个资源监视器, 该资源监视器已配置为每分钟监视一个源队列, 以查找完整的消息组或不在组中的个别消息。对于监视器找到的每个消息组或个别消息, 它提交一个任务 (以受管传输请求的形式) 以将该消息或消息组的内容移动到文件中。

运行监视器的代理程序设置了以下代理程序属性:

```
maxQueuedTransfers=1000
maxSourceTransfers=25
monitorMaxResourcesInPoll=25
```

这意味着在每次轮询期间, 监视器都有可能向代理程序提交 25 个任务。假定代理大约需要 2 分钟来处理所有 25 个任务, 那么在轮询时间间隔为 1 分钟的情况下, 将发生以下行为:

分钟 0

- 监视器启动轮询, 扫描源队列并查找 25 条消息 (**monitorMaxResourcesInPoll** 的值)。
- 监视器现在向代理提交 25 个任务 (或受管传输请求), 然后停止其轮询。
- 代理程序选取 25 个受管传输请求, 为每个受管传输请求分配一个源传输槽并开始处理这些请求。

此时, 代理的传输槽如下所示:

| | Used | Free |
|-----------------------|------|------|
| Source transfer slots | 25 | 0 |
| Queued transfer slots | 0 | 1000 |

分钟 1

- 监视器现在开始其第二次轮询。
- 监视器再次扫描源队列, 查找 25 条消息并向代理提交 25 个受管传输请求。
- 投票结束。
- 代理接收这些新的受管传输请求。由于其所有源传输槽都已占用, 因此它将为每个受管传输请求分配一个排队的传输槽, 并将它们放在其待办事项上。

现在, 代理的传输槽如下所示:

| | Used | Free |
|-----------------------|------|------|
| Source transfer slots | 25 | 0 |
| Queued transfer slots | 25 | 975 |

分钟 2

- 此时，所有 25 个受管传输都已完成处理，并且将释放其关联的源传输槽。因此，代理会将 25 个受管传输从排队的传输槽移至源传输槽。

这会使代理的传输槽如下所示：

| | Used | Free |
|-----------------------|------|------|
| Source transfer slots | 25 | 0 |
| Queued transfer slots | 0 | 1000 |

- 监视器执行另一个轮询，查找另一批 25 条消息，并向代理提交 25 个受管传输请求。
- 代理程序会选取这些请求并将其放入其任务列表中

这意味着传输槽现在看起来如下所示：

| | Used | Free |
|-----------------------|------|------|
| Source transfer slots | 25 | 0 |
| Queued transfer slots | 25 | 975 |

分钟 3

- 在下一轮询期间，监视器会发现另外 25 条消息，因此会向代理提交另外 25 条受管传输请求。
- 代理接收这些受管传输请求，并为每个受管传输请求分配一个排队的传输槽。

因此，代理的传输槽现在如下所示：

| | Used | Free |
|-----------------------|------|------|
| Source transfer slots | 25 | 0 |
| Queued transfer slots | 50 | 950 |

等等。

将轮询时间间隔增加到 2 分钟

将监视器的轮询时间间隔增加到两分钟意味着在一次轮询期间提交的 25 个受管传输将在下一次启动时完成。这意味着代理能够将这些受管传输分配给源传输槽，而不必将其放入其任务列表中，如下示例中所示：

分钟 0

- 监视器启动轮询，扫描源队列并查找 25 条消息 (`monitorMaxResourcesInPoll` 的值)。
- 监视器现在向代理提交 25 个受管传输请求，然后停止其轮询。
- 代理程序选取 25 个受管传输请求，为每个受管传输请求分配一个源传输槽并开始处理这些请求。

此时，代理的传输槽如下所示：

| | Used | Free |
|-----------------------|------|------|
| Source transfer slots | 25 | 0 |
| Queued transfer slots | 0 | 1000 |

```
Source transfer slots | 25 | 0
Queued transfer slots | 0 | 1000
```

分钟 2

- 此时，所有 25 个受管传输都已完成处理，并且将释放其关联的源传输槽。

这意味着代理的传输槽如下所示：

```
----- | Used | Free
Source transfer slots | 0 | 25
Queued transfer slots | 0 | 1000
```

- 监视器执行另一个轮询，查找另一批 25 条消息，并向代理提交 25 个受管传输请求。
- 代理程序会选取这些请求，并为每个请求分配一个源传输槽。

这意味着传输槽现在看起来如下所示：

```
----- | Used | Free
Source transfer slots | 25 | 0
Queued transfer slots | 0 | 1000
```

分钟 4

- 两分钟后，监视器提交的 25 个受管传输请求 (以分钟计) 2 已完成，其关联的 "源传输槽" 已释放并释放。

代理的源传输槽现在如下所示：

```
----- | Used | Free
Source transfer slots | 0 | 25
Queued transfer slots | 0 | 1000
```

- 现在，监视器将执行新的轮询，并在队列中找到另外 25 条消息。因此，它会向代理提交 25 个受管传输请求。
- 代理程序会选取受管传输请求。由于它当前未充当任何受管传输的源代理，因此会将 "源传输槽" 分配给每个新请求。

这使其传输槽看起来如下所示：

```
----- | Used | Free
Source transfer slots | 25 | 0
Queued transfer slots | 0 | 1000
```

此方法的优点是，受管传输永远不会进入代理的积压，这会降低代理的总体资源使用率，进而有助于提高性能。

对 java.lang.OutOfMemoryError 问题进行故障诊断

使用以下参考信息来帮助您解决由于 java.lang.OutOfMemoryErrors 而导致代理程序停止的问题

相关概念

[第 103 页的『如果 MFT 代理程序 ABENDS 由于 Java 堆耗尽而产生 java.lang.OutOfMemoryError，该怎么办?』](#)

在处理大量受管传输请求时，例如，文件至文件传输、消息至文件传输或文件至消息传输，代理异常结束（异常终止），报告 java.lang.OutOfMemoryError，并且此时未完全利用全部 RAM 内存。此异常是由 Java 堆耗尽引起的。

[第 105 页的『当由于本机内存耗尽导致 MFT 代理异常终止且显示 java.lang.OutOfMemoryError 时该怎么做』](#)

在处理大量受管传输请求时，例如，文件至文件传输、消息至文件传输或文件至消息传输，代理异常结束（异常终止），报告 `java.lang.OutOfMemoryError`，并且此时未完全利用全部 RAM 内存。此异常是本机内存耗尽所导致的。

如果 MFT 代理程序 ABENDS 由于 Java 堆耗尽而产生 `java.lang.OutOfMemoryError`，该怎么办？

在处理大量受管传输请求时，例如，文件至文件传输、消息至文件传输或文件至消息传输，代理异常结束（异常终止），报告 `java.lang.OutOfMemoryError`，并且此时未完全利用全部 RAM 内存。此异常是由 Java 堆耗尽引起的。

诊断问题

发生此问题时，受影响的代理程序 ABEND 并生成四个文件以提供有关根本原因的详细信息：

- ABEND 文件。此文件的名称符合命名约定 `ABEND.FTE.date_timestamp.identifier.log`。

► **Multi** 在多平台上，该文件将写入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/ffdc` 目录。

► **z/OS** 在 z/OS 上，该文件将写入 z/OS UNIX System Services (z/OS UNIX) 位置 `$BFG_CONFIG/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/ffdc`

- Javacore 文件。此文件的名称具有以下格式：

`javacore.datestamp.timestamp.pid.identifier.txt`

► **Multi** 在多平台上，该文件将写入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name` 目录。

► **z/OS** 在 z/OS 上，该文件将写入 z/OS UNIX 位置 `$BFG_CONFIG/mqft/logs/coordination_qmgr_name/agents/agent_name` 目录。

- Java 快照转储。此文件的名称具有以下格式：`snap.datestamp.timestamp.pid.identifier.txt`

► **Multi** 在多平台上，该文件将写入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name` 目录。

► **z/OS** 在 z/OS 上，该文件将写入 z/OS UNIX 位置 `$BFG_CONFIG/mqft/logs/coordination_qmgr_name/agents/agent_name` 目录。

ABEND 和 Javacore 对包含类似于以下所示示例的信息：

Abend 文件

```
Filename:
C:\ProgramData\IBM\MQ\mqft\logs\QM1\agents\AGENT1\logs\ffdc\ABEND.FTE.20220810102649225.18938124211177445
3.log
Level:      p920-005-220208
Time:       10/08/2022 10:26:49:225 BST
Thread:     45 (FileIOWorker-0:0)
Class:      com.ibm.wmqfte.thread.FTETHread
Instance:   a393304f
Method:     uncaughtException
Probe:      ABEND_001
Cause:      java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
  at java.nio.HeapByteBuffer.<init>(HeapByteBuffer.java:57)
  at java.nio.ByteBuffer.allocate(ByteBuffer.java:335)
  at com.ibm.wmqfte.util.impl.ByteBufferPoolImpl.getBuffer(ByteBufferPoolImpl.java:44)
  at com.ibm.wmqfte.transfer.frame.impl.TransferChunkImpl.getByteBuffer(TransferChunkImpl.java:181)
  at com.ibm.wmqfte.transfer.frame.impl.TransferChunkImpl.<init>(TransferChunkImpl.java:143)
  at
com.ibm.wmqfte.transfer.frame.impl.TransferFrameSenderImpl.requestChunk(TransferFrameSenderImpl.java:636)
  at
com.ibm.wmqfte.transfer.frame.impl.TransferFrameSenderImpl.access$000(TransferFrameSenderImpl.java:100)
  at
com.ibm.wmqfte.transfer.frame.impl.TransferFrameSenderImpl$ChunkRequester.processFileIORequest(TransferFr
ameSenderImpl.java:142)
```

```

at
com.ibm.wmqfte.transfer.frame.impl.TransferFrameIOWorker.doWorkImpl(TransferFrameIOWorker.java:318)
at com.ibm.wmqfte.io.impl.FTEFileIOWorker.doWork(FTEFileIOWorker.java:118)
at com.ibm.wmqfte.io.impl.FTEFileIORequestQueue.run(FTEFileIORequestQueue.java:244)
at java.lang.Thread.run(Thread.java:825)
at com.ibm.wmqfte.thread.FTEThread.run(FTEThread.java:70)

```

Javacore 文件

```

0SECTION      TITLE subcomponent dump routine
NULL          =====
1TICCHARSET   437
1TISIGINF0    Dump Event "systhrow" (00040000) Detail "java/lang/OutOfMemoryError" "Java heap space"
received
1TIDATETIMEUTC Date: 2022/08/10 at 09:26:53:917 (UTC)
1TIDATETIME   Date: 2022/08/10 at 10:26:53:917
1TITIMEZONE   Timezone: (unavailable)
1TINANOTIME   System nanotime: 350635184939400
1TIFILENAME   Javacore filename:
C:\ProgramData\IBM\MQ\mqft\logs\QM1\agents\AGENT1\javacore.20220810.102653.7172.0003.txt

```

发生此问题的原因

由于运行代理程序的 JVM 的 Java 堆内存耗尽，因此发生此问题。

请参阅 [MFT 代理如何使用 Java 堆和本机堆内存](#) 以获取有关 Java 堆内存与本机堆内存之间的区别的更多信息。

避免问题

您可以执行一些操作来帮助降低 MFT 代理程序由于 `java.lang.OutOfMemoryError` 而停止的可能性，这是由于 Java 堆内存耗尽所导致的：

1. 增加运行 MFT 代理程序的 JVM 的 Java 堆大小。

缺省情况下，代理的 Java 堆设置为 512 MB。虽然这对于少量受管传输是令人满意的，但对于类似于生产的工作负载，可能需要将其增加到最多 1024MB (1GB)。



注意：当增大代理程序的 Java 堆大小时，请务必考虑与使用本机堆的代理程序和应用程序在同一系统上运行的其他代理程序和应用程序。

增大代理程序的 Java 堆大小还会增加其本机堆使用率，从而减少可用于其他代理程序和应用程序的本机堆的数量。这意味着代理程序和应用程序迁到本机堆耗尽的可能性增加。

- 要在作为正常进程运行代理程序时增加或更改 Java 堆：

设置 `BFG_JVM_PROPERTIES` 环境变量以将 Java 属性 `-Xmx` 传递到 JVM。例如，在 Windows 上，要将最大堆大小设置为 1024 MB，请在使用 `fteStartAgent` 命令之前运行以下命令：

```
set BFG_JVM_PROPERTIES="-Xmx1024M"
```

有关如何使用 `BFG_JVM_PROPERTIES` 环境变量设置 Java 系统属性的更多信息，请参阅 [MFT 的 Java 系统属性](#)。

- 要在将代理程序作为 Windows 服务运行时增加或更改 Java 堆，请执行以下操作：

使用 `fteModifyAgent` 命令并指定 `-sj` 参数以设置 Windows 服务上的 `-Xmx` 属性。

以下示例使用带有 `-sj` 参数的 `fteModifyAgent` 命令，将运行 Windows 服务配置的代理程序的 JVM 的 Java 堆的最大大小设置为 1GB (1024MB)：

```
fteModifyAgent.cmd -agentName AGENT1 -s -su user1 -sp passw0rd -sj -Xmx1024M
```

您可以在重新启动代理后，通过复查代理的 `output0.log` 文件检查是否已成功设置了此内容。在 *Start Display Current Environment* 部分中，将报告值 1024 MB，如下所示：

```
The maximum amount of memory that the Java virtual machine will attempt to use is: '1024'MB
```


2. 通过减少代理程序的工作负载来限制 Java 堆使用率。

通常，由 Java 堆耗尽导致的 `java.lang.OutOfMemoryErrors` 是代理程序执行过多工作的结果。代理正在处理的每个受管传输和受管调用都使用 Java 堆中的内存，代理积压的受管传输和受管调用也是如此。资源监视器在执行轮询时也使用 Java 堆内存。

这意味着随着代理程序工作负载的增加，它所使用的 Java 堆量也会增加。

减少代理程序的工作负载可在此处提供帮助。要执行此操作：

- 将以下代理程序属性设置为较小的值：
 - `maxQueuedTransfers`
 - `maxSourceTransfers`
 - `maxDestinationTransfers`
- 将代理程序的某些资源监视器移至新代理程序。

这会减少可发生的并行传输的数量，因此降低代理的最大并行工作负载。

3. 启用内存分配检查。

内存分配检查功能确保仅当有足够的 Java 堆内存可供代理程序运行至完成时，代理程序才开始处理新的受管传输。如果内存不足，那么将拒绝受管传输。

缺省情况下，此功能处于关闭状态。要对代理程序启用此功能，请执行以下操作：

- 将以下条目添加到代理程序的 `agent.properties` 文件：

```
enableMemoryAllocationChecking=true
```

- 重新启动代理程序

注：内存分配检查功能使用受管传输所需的最大内存量，这可能超过实际使用的内存量（尤其是用于消息到文件和文件到消息传输的内存量）。这意味着打开它可能会导致代理处理的受管传输较少。

如果代理程序由于 Java 堆耗尽而继续迁到 `java.lang.OutOfMemoryErrors`，请运行 `fteRas` 命令以收集 ABEND 文件，`Javacores`，堆转储文件和快照转储文件（以及有关 MFT 拓扑的其他有用信息），并使输出可供 IBM 支持人员用于分析。

相关概念

第 105 页的『[当由于本机内存耗尽导致 MFT 代理异常终止且显示 `java.lang.OutOfMemoryError` 时该怎么做](#)』

在处理大量受管传输请求时，例如，文件至文件传输、消息至文件传输或文件至消息传输，代理异常结束（异常终止），报告 `java.lang.OutOfMemoryError`，并且此时未完全利用全部 RAM 内存。此异常是本机内存耗尽所导致的。

当由于本机内存耗尽导致 MFT 代理异常终止且显示 `java.lang.OutOfMemoryError` 时该怎么做

在处理大量受管传输请求时，例如，文件至文件传输、消息至文件传输或文件至消息传输，代理异常结束（异常终止），报告 `java.lang.OutOfMemoryError`，并且此时未完全利用全部 RAM 内存。此异常是本机内存耗尽所导致的。

诊断问题

发生此问题时，受影响的代理异常终止并生成两个文件，其中提供有关根本原因的详细信息：

- ABEND 文件。此文件的名称符合命名约定 `ABEND.FTE.date_timestamp.identifier.log`。

Multi 在多平台上，该文件将写入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/ffdc` 目录。

z/OS 在 z/OS 上，此文件将写入 USS 位置 `$BFG_CONFIG/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/ffdc`

- Javacore 文件。此文件的名称具有以下格式:

`javacore.datestamp.timestamp.pid.identifier.txt`

Multi 在多平台上, 该文件将写入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name` 目录。

z/OS 在 z/OS 上, 该文件将写入 USS 位置 `$BFG_CONFIG/mqft/logs/coordination_qmgr_name/agents/agent_name` 目录。

ABEND 和 Javacore 对包含类似于以下所示示例的信息:

示例: 第 1 对

Abend 文件

```
Filename:
C:\ProgramData\IBM\MQ\mqft\logs\COORDQM\agents\AGENT1\logs\ffdc\ABEND.FTE.20200109113518046.1764802189777
906538.log
Level: p900-005-180821
Time: 09/01/2020 11:35:18:046 GMT
Thread: 96 (TransferSender[414d51204d44424b525030372020202045fbd6532ebfaa02])
Class: com.ibm.wmqfte.thread.FTETHread
Instance: 55b455b4
Method: uncaughtException
Probe: ABEND_001
Cause: java.lang.OutOfMemoryError: native memory exhausted

java.lang.OutOfMemoryError: native memory exhausted
at com.ibm.mq.jmqi.local.internal.base.Native.MQPUT(Native Method)
at com.ibm.mq.jmqi.local.LocalMQ.MQPUT(LocalMQ.java)
at com.ibm.wmqfte.wmqiface.WMQQueueImpl.put(WMQQueueImpl.java)
at com.ibm.wmqfte.wmqiface.WMQQueueImpl.put(WMQQueueImpl.java)
at com.ibm.wmqfte.transfer.impl.TransferSenderRunnable.doTransfer(TransferSenderRunnable.java)
at com.ibm.wmqfte.transfer.impl.TransferSenderRunnable.run(TransferSenderRunnable.java)
at java.lang.Thread.run(Thread.java)
at com.ibm.wmqfte.thread.FTETHread.run(FTETHread.java)
```

Javacore 文件

```
NULL -----
OSECTION TITLE subcomponent dump routine
NULL =====
1TISIGINFO Dump Event "systhrow" (00040000) Detail "java/lang/OutOfMemoryError" "native memory
exhausted" received
1TIDATETIME Date: 2020/01/09 at 11:35:18
1TIFILENAME Javacore filename:
C:\ProgramData\IBM\MQ\mqft\logs\COORDQM\agents\AGENT1\javacore.20200109.113518.14148.0002.txt
```

示例: 第 2 对

ABEND 文件

```
Filename:
C:\ProgramData\IBM\MQ\mqft\logs\COORDQM\agents\AGENT1\logs\ffdc\ABEND.FTE.20200109143700286.3177895731698
464509.log
Level: p900-005-180821
Time: 09/01/2020 14:37:00:286 GMT
Thread: 918 (AgentStatusPublisher)
Class: com.ibm.wmqfte.thread.FTETHread
Instance: bc10bc1
Method: uncaughtException
Probe: ABEND_001
Cause: java.lang.OutOfMemoryError: Failed to create a thread: retVal -1073741830, errno 12

java.lang.OutOfMemoryError: Failed to create a thread: retVal -1073741830, errno 12
at java.lang.Thread.startImpl(Native Method)
at java.lang.Thread.start(Thread.java)
```

Javacore 文件

```
NULL -----
0SECTION TITLE subcomponent dump routine
NULL =====
1TISIGINFO Dump Event "systhrow" (00040000) Detail "java/lang/OutOfMemoryError" "Failed to create a
thread: retVal -1073741830, errno 12" received
1TIDATETIME Date: 2020/01/09 at 14:37:00
1TIFILENAME Javacore filename: C
C:\ProgramData\IBM\MQ\mqft\logs\C00RDQM\agents\AGENT1\javacore.20200109.143700.2652.0003.txt
```

发生此问题的原因

由于运行代理的系统上的本机堆内存耗尽而发生此问题。

请参阅 [MFT 代理如何使用 Java 堆和本机堆内存](#) 以获取有关 Java 堆内存与本机堆内存之间的区别的更多信息。

避免问题

您可以执行一些操作，以帮助减少由于本机内存耗尽而导致 MFT 代理程序由于 `java.lang.OutOfMemoryError` 而停止的可能性：

1. 减少运行 MFT 代理程序的 JVM 的 Java 堆大小。

分配的 Java 堆大小越大，可用于本机堆的内存越少。减少代理使用的 Java 堆的大小可释放更多内存用于本机堆。

缺省情况下，代理的 Java 堆设置为 512 MB。如果已将此项更改为更大的值，请考虑减小此项，并使用类似生产的工作负载进行测试。

- 要在将代理作为正常进程运行时降低或更改 Java 堆，请执行以下操作：

设置 `BFG_JVM_PROPERTIES` 环境变量以将选项目录传递到 JVM。例如，在 Windows 上，要将最大堆大小设置为 1024 MB，请在使用 `fteStartAgent` 命令之前运行以下命令：

```
set BFG_JVM_PROPERTIES="-Xmx1024M"
```

有关如何使用 `BFG_JVM_PROPERTIES` 环境变量设置 Java 系统属性的更多信息，请参阅 [MFT 的 Java 系统属性](#)。

- 要在将代理程序作为 Windows 服务运行时降低或更改 Java 堆：

要将选项传递到将代理程序作为 Windows 服务运行的 JVM，请使用 `fteModifyAgent` 命令中指定的 `-sj` 参数修改代理程序。

以下示例将 `fteModifyAgent` 命令与 `-sj` 参数配合使用，以设置运行 Windows 服务配置的代理程序的 JVM 的 Java 堆的最大大小：

```
fteModifyAgent.cmd -agentName AGENT1 -s -su user1 -sp passw0rd -sj -Xmx1024M
```

您可以在重新启动代理后，通过复查代理的 `output0.log` 文件检查是否已成功设置了此内容。在 *Start Display Current Environment* 部分中，将报告值 1024 MB，如下所示：

```
The maximum amount of memory that the Java virtual machine will attempt to use is: '1024'MB
```

2. 限制本机内存使用

通常，如果代理使用 `BINDINGS` 传输连接到其代理队列管理器，那么会看到本机堆耗尽所导致的 `java.lang.OutOfMemoryErrors`。在已将代理配置为使用 `BINDINGS` 传输时，只要需要与队列管理器通信，代理就会调用本机方法。

这意味着，由于队列管理器连接的增加和消息通信的增加，本机内存使用量随代理工作负载的增加而增加。在此情况下，减少工作负载可能很有用。要执行此操作，请将以下代理程序属性设置为小于缺省值 25 的值：

- **maxSourceTransfers**
- **maxDestinationTransfers**

这会减少可发生的并行传输的数量，因此降低代理的最大并行工作负载。

3. 配置代理以在连接到其代理队列管理器时使用 CLIENT 传输。可以通过设置以下代理属性来执行此操作：

- **agentQMgrHost**
- **agentQMgrPort**
- **agentQMgrChannel**

您可以在 [MFT agent.properties](#) 文件主题中查找有关这些属性的信息。

这确保代理与队列管理器之间的所有通信均通过 TCP/IP 进行，而不是本机代码，因此减少了代理所使用的本机内存量。

要点: 执行此操作也会降低性能。使用与本地主机的 TCP/IP 连接，而非本机代码的情况下，当代理查询与队列管理器的交互时，此配置效率不高。

对记录器问题进行故障诊断

使用以下参考信息来帮助您解决记录器问题：

相关参考

第 115 页的『MFT 常见问题』

Managed File Transfer 网络中可能出现的常见问题。

第 78 页的『MFT 的返回码』

Managed File Transfer 命令、Ant 任务和日志消息均提供了用于指示是否已成功完成功能的返回码。

在 Oracle 数据库上更新 MFT 数据库模式期间收到错误时该怎么做

使用 `ftelog_tables_oracle_702_703.sql` 文件将数据库模式更新到最新级别时，可能会收到以下错误消息：ERROR at line 1: ORA-02289: sequence does not exist。出现该错误的原因是表所使用的序列和触发器与这些表未处于同一模式中。

关于此任务

要解决此问题，必须先编辑 `ftelog_tables_oracle_702_703.sql` 的内容，然后再运行。

过程

1. 查明 Managed File Transfer 数据库记录器表所使用的序列和触发器位于哪个模式中。
 - 在 Db2 上，可以使用“控制中心”来查看表和模式。
 - 在 Oracle 上，可以使用 Enterprise Manager 来查看表和模式。
2. 在文本编辑器中打开 `ftelog_tables_oracle_702_703.sql` 文件。
3. 在每次出现的文本 `SELECT FTELOG.sequence_name.nextval` 中，将文本 FTELOG 替换为现有序列所在模式的名称。
4. 在每次出现文本 `CREATE OR REPLACE TRIGGER FTELOG.trigger_name` 之前，插入文本 `DROP TRIGGER schema_name.trigger_name`，其中 `schema_name` 是现有触发器所在的模式的名称。
5. 使用已编辑的 `ftelog_tables_oracle_702_703.sql` 文件来更新数据库表。

MFT 记录器错误处理和拒绝

Managed File Transfer 记录器可识别两种类型的错误：每条消息的错误和通用错误。

每条消息错误可能是由一条或几条个别消息的相关问题导致的。确定为每条消息错误的一些情境示例如下所示：

- 某条消息中缺少结果代码（必需的数据项）
- 某个传输中指定了长度为 3000 个字符且对于关联数据库列过大的作业名

- 接收到有关传输的进度消息，但并没有记录表明传输已启动（可能是因为错误路由或延迟了“传输启动”消息）
- 接收到一条消息，但不是 Managed File Transfer 日志消息

通用错误是每条消息错误之外的所有错误。这些错误的原因可能是配置问题或程序错误。

遇到每条消息的错误时，记录器会将该消息放到拒绝队列上以拒绝该消息。不会向输出日志写入任何内容，因此需要定期检查或持续监视拒绝队列以检测拒绝的消息。

如果连续拒绝了过多消息，而没有任何消息成功写入数据库，那么这种情况将视为通用错误。例如，考虑这样一个站点，它始终使用 10 字符代码作为作业名，但无意间将作业名列重新配置为两个字符宽。尽管过宽的数据通常为每条消息错误，但在此情况下，配置问题是通用的，因此将检测为通用错误。您可以使用 **wmqfte.max.consecutive.reject** 属性调整导致常规错误所需的连续每个消息错误数。

如果检测到通用错误，记录器将回滚尚未提交到队列管理器的任何消息，然后定期重试。如果记录器是使用 **-F** 参数以前台方式启动的，那么会将标识问题的消息写入输出日志和控制台。

记录器的输出日志的位置取决于它是独立数据库记录器还是 JEE 数据库记录器。对于独立数据库记录器，它位于 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name` 目录中。对于 JEE 数据库记录器，它位于应用程序服务器的标准输出日志中。

拒绝队列

产生每条消息错误的消息将移动到拒绝队列中。在每条拒绝的消息上，将设置一个消息属性以指示拒绝该消息的原因。该属性的全名是 **usr.WMQFTE_ReasonForRejection**，尽管是 **usr**。在某些上下文中省略（包括 JMS 和 IBM MQ Explorer）。

如果要使用 IBM MQ Explorer，那么可以通过右键单击拒绝队列并单击**浏览消息**来查看该队列的内容。要查看拒绝消息的原因，请双击该消息以打开其属性对话框，然后选择“**已命名属性**”页面。您将看到名为 **WMQFTE_ReasonForRejection** 的属性。或者，您也可以编写或配置监视工具以自动获取此信息。

有时，您可能希望重新处理拒绝队列中的消息。在本主题先前描述的示例中，对于数据库中的两字符作业名列，在数据库列宽度增加后可以成功处理这些消息。作为另一示例，如果“传输完成”消息由于缺少关联的“传输启动”而被拒绝，可能会在以后接收到“传输启动”消息。这样，重新处理“传输完成”将会成功。

要重新处理消息，请将这些消息从拒绝队列移动到输入队列。在正常安装中（其中记录器创建了自己的受管预订），输入队列由队列管理器定义并具有类似 `SYSTEM.MANAGED.DURABLE.49998CFF20006204` 的名称。您可以通过查看预订 `SYSTEM.FTE.DATABASELogger.AUTO` 的属性中的**目标名称**或使用以下 MQSC 命令来识别输入队列：

```
DISPLAY SUB(SYSTEM.FTE.DATABASELogger.AUTO) DEST
```

在队列之间移动消息的一种方法是使用 [MA01 SupportPac](#)，例如：

```
q -IFTE.REJECT -oSYSTEM.MANAGED.DURABLE.49998CFF20006204
```

拒绝队列可能包含由于各种原因而被拒绝的消息，其中仅有一部分得以解决。在此情况下，您仍可以重新处理所有消息；现在可接受的消息将被使用，而不能接受的消息则再次移动到拒绝队列。

传输日志中的 **Malformed** 日志消息不是由记录器记录的。这些消息不会视为重要消息，因此这些消息将发送到拒绝队列。有关传输日志消息的更多信息，请参阅[文件传输日志消息格式](#)。

在 MFT 记录器已启动但未将任何传输信息记录到数据库时该怎么做


Managed File Transfer 记录器使用的数据库表要求数据库的页面大小为 8KB 或更大。如果数据库的页面大小不足，那么无法正确创建数据库表，并且您将看到 `SQLSTATE=42704` 错误。

如果您正在使用 Java Platform, Enterprise Edition 数据库记录器, 那么可能会在 WebSphere Application Server 系统输出日志中看到以下消息; 如果您正在使用独立数据库记录器, 那么在 output0.log 文件中可能会看到以下错误:

```
DB2 SQL Error: SQLCODE=-204, SQLSTATE=42704
SQLERRMC=FTELOG.TRANSFER_EVENT, DRIVER=3.40.152
```



SQLSTATE 值为 42704 指示记录器预期存在的某个表 (在本例中为 FTELOG.TRANSFER_EVENT) 并不存在。

要修复此问题, 请执行下列步骤:

1. 检查表是否存在且是否完整。有关记录器所使用的表及其列的信息, 请参阅 [MFT 数据库记录器表](#)。
2. 如果表不存在或者不完整, 请检查数据库的页面大小。
3. 如果数据库大小小于 8KB, 请增加数据库页面的大小。
 - 如果数据库位于测试系统上或者数据库中没有数据, 您可以删除表并重新创建一个页面大小大于 8KB 的数据库。
 - 有关如何增加页面大小的信息, 请参阅 [迁移 MFT: 增加 UNIX, Linux, and Windows 上 Db2 的日志数据库页面大小](#)  或 [将 Db2 on z/OS 上的数据库表迁移到 MQ V8.0 或更高版本](#)。

对 Connect:Direct 网桥进行故障诊断

使用以下参考信息和示例, 以帮助您诊断 Connect:Direct 网桥返回的错误。

- [第 110 页的『跟踪 Connect:Direct 网桥』](#)
- [第 111 页的『Connect:Direct 网桥的日志信息』](#)
- [第 111 页的『解决 Connect:Direct 节点的许可权问题』](#)
- [第 111 页的『如果与 Connect:Direct 节点之间的文本传输未正确转换数据, 该如何操作』](#)
-  [第 112 页的『如果通过 Connect:Direct 网桥到 PDS 或 PDS 成员的传输失败, 该如何操作』](#)
-  [第 112 页的『使用双正斜杠指定 Connect:Direct 文件路径』](#)
- [第 113 页的『增加 Connect:Direct 网桥的并行传输数』](#)
- [第 113 页的『调试文件传输调用的 Connect:Direct 进程』](#)

跟踪 Connect:Direct 网桥

您可以从作为 Connect:Direct 网桥组成部分的 Connect:Direct 节点来捕获跟踪, 以帮助确定问题。

关于此任务

要启用跟踪, 请完成以下步骤:

过程

1. 停止 Connect:Direct 网桥代理。
2. 编辑 Connect:Direct 网桥代理属性文件, 以包含以下行:

```
cdTrace=true
```

3. 启动 Connect:Direct 网桥代理。

结果

跟踪信息将写入 Connect:Direct 网桥代理配置目录下的 output0.log 文件中。

相关参考

[MFT agent.properties 文件](#)

Connect:Direct 网桥的日志信息

您可以使用 Connect:Direct 网桥代理，在 MFT 代理和 Connect:Direct 节点之间传输文件。有关这些传输中涉及的 Connect:Direct 节点和进程的日志信息都显示在 IBM MQ Explorer 插件中，并存储在您的日志数据库内。

Connect:Direct 网桥代理必须是 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本。该传输中涉及的其他代理可以是任意版本的 Managed File Transfer。但是，要记录有关 Connect:Direct 节点和进程的信息，传输中涉及的所有 MFT 代理都必须是 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本。要在 IBM MQ Explorer 插件中显示该信息，该插件必须为 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本。要在日志数据库中存储该信息，数据库记录器和数据库模式必须是 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本。

有关文件传输中涉及的 Connect:Direct 节点和 Connect:Direct 进程的日志信息包含在发布到 SYSTEM.FTE 主题。有关更多信息，请参阅[文件传输日志消息格式](#)。

以下信息包含在已发布的消息内：

- Connect:Direct 网桥节点名称
- 主节点 (PNODE) 名称
- 辅助节点 (SNODE) 名称
- 进程名称
- 进程标识号

Connect:Direct 网桥节点与主节点或辅助节点是同一节点。

Connect:Direct 网桥节点名的值是 MFT Connect:Direct 网桥代理所识别的网桥节点的名称。主节点名称和辅助节点名称是用于引用 Connect:Direct 网桥节点的网络映射中的节点的名称。

相关参考

[Connect:Direct 网桥传输日志消息示例](#)

解决 Connect:Direct 节点的许可权问题

如果 Managed File Transfer 和 Connect:Direct 之间的传输失败，并且出现关于没有足够许可权的错误，请使用本主题中的信息。

对于涉及 Connect:Direct 网桥的传输，连接到 Connect:Direct 节点的用户标识由与传输请求相关联的 IBM MQ 消息描述符 (MQMD) 用户标识来确定。您可以将特定的 MQMD 用户标识映射到特定的 Connect:Direct 用户标识。有关更多信息，请参阅[映射 Connect:Direct 的凭证](#)。

您可能看到传输失败，并且出现以下某错误：


- ```
BFGCD0001E: This task was rejected by the Connect:Direct API with the following error message: Connect:Direct Node detected error.
LCCA000I The user has no functional authority to issue the selp command
```
- ```
BFGCD0026I: Connect:Direct messages: The submit of the process succeeded. Process number 1092 (name F35079AE, SNODE MYNODE)
executing. User fteuser does not have permission to override SNODEID.
User fteuser does not have permission to override SNODEID. User
fteuser does not have permission to override SNODEID.
```

如果看到以上任一错误，请确定与传输请求中使用的 MQMD 用户标识相关联的 Connect:Direct 用户标识。该 Connect:Direct 用户标识必须具有执行 Connect:Direct 网桥要求的 Connect:Direct 操作的权限。有关所需的功能权限列表，以及有关如何授予这些权限的指导，请参阅[使用 ConnectDirectCredentials.xml 文件映射 Connect:Direct 的凭证](#)。

如果与 Connect:Direct 节点之间的文本传输未正确转换数据，该如何操作

当以文本方式在 MFT 代理和 Connect:Direct 节点之间传输文件时，会执行代码页转换和行尾字符转换。该传输使用 Connect:Direct 网桥节点的网络映射中的操作系统信息来确定远程节点的行尾字符。如果网络映射中的信息不正确，那么行尾字符转换可能未正确执行。

确保 Connect:Direct 网桥节点和用作传输目标的任何 Connect:Direct 节点的网络映射包含正确的平台描述。


- 如果 Connect:Direct 网桥节点在 Windows 系统上，那么请确保针对网络映射中的每个远程节点，从**操作系统**列表中选择正确的值。
 - 如果远程节点在 Windows 系统上，那么请选择 Windows。
 - 如果远程节点在 UNIX 或 Linux 系统上，请选择 UNIX。
 -  如果远程节点在 z/OS 系统上，那么请选择 OS/390。

Connect:Direct 网桥不支持到其他操作系统上的远程节点的传输。

- 确保对于作为文件传输目标或源的每个远程节点，您在 Connect:Direct 网桥代理配置目录下的 `ConnectDirectNodeProperties.xml` 文件中指定了远程 Connect:Direct 节点的操作系统类型。有关更多信息，请参阅配置 `ConnectDirectNodeProperties.xml` 文件以包含有关远程 Connect:Direct 节点和 Connect:Direct 节点属性文件格式的信息。

相关任务

[在 Connect:Direct 与 MFT 之间传输文本文件](#)

 **如果通过 *Connect:Direct* 网桥到 PDS 或 PDS 成员的传输失败，该如何操作**
如果传输的目标是 z/OS 上的 Connect:Direct 节点，并且是 PDS 或 PDS 成员，那么如果未使用覆盖值指定 `-de` 参数，那么传输将失败。

关于此任务

如果使用 `fteCreateTransfer` 或 `fteCreateTemplate` 命令提交了传输，请执行以下步骤：

过程

1. 更改提交的命令，以包含 `-de overwrite`。
2. 再次提交命令。

使用 *IBM MQ Explorer* 插件

关于此任务

如果使用 IBM MQ Explorer 插件提交了传输，请执行以下步骤：

过程

1. 在“创建新的受管文件传输”向导中指定源和目标信息。
2. 选择覆盖目标文件系统中具有相同名称的文件。
3. 再次提交命令。

使用双正斜杠指定 *Connect:Direct* 文件路径

在文件传输过程中，如果使用以双正斜杠 (//) 开始的文件路径指定位于 Connect:Direct 节点上的文件，那么会将该文件视为数据集。

Connect:Direct 节点上的源和目标都以格式 `cd_node_name:file_path` 指定。如果 `file_path` 以双正斜杠 (//) 开始，那么该源或目标将被视为数据集。即使 Connect:Direct 节点未在 z/OS 上，也是如此。如果意外将文件路径指定为以双正斜杠 (//) 开始并且文件并非数据集，那么会导致传输失败。

如果不希望指定的文件被视为数据集，那么请勿指定以双正斜杠 (//) 开始的 `file_path`。

相关概念

[第 110 页的『对 Connect:Direct 网桥进行故障诊断』](#)

使用以下参考信息和示例，以帮助诊断 Connect:Direct 网桥返回的错误。

相关任务

 [与 Connect:Direct 节点传输数据集](#)

增加 **Connect:Direct** 网桥的并行传输数

要增加 Connect:Direct 网桥代理可以处理的并行传输数，必须更改三个代理属性。您还必须增加 Connect:Direct 节点接受的最大连接数。

Connect:Direct 网桥代理可处理的最大并行传输数取决于某些代理属性的值。**maxSourceTransfers** 和 **maxDestinationTransfers** 代理属性针对 Connect:Direct 网桥代理的传输数缺省值为 5。该缺省值低于针对其他类型的代理的传输数缺省值 (25)。Connect:Direct 网桥 (其中使用缺省值 **maxSourceTransfers** 和 **maxDestinationTransfers** 配置了代理) 在任何时候最多可以处理 10 个传输: 5 个传输 (其中代理是源)，5 个传输 (其中代理是目标)。

这些缺省值可确保 Connect:Direct 网桥代理不超出到 Connect:Direct 节点的最大 API 连接数。使用缺省配置的 Connect:Direct 网桥代理最多使用 10 个到 Connect:Direct 节点的 API 连接。UNIX 上的 Connect:Direct 节点接受的最大连接数由 **api.max.connects** Connect:Direct 参数控制。对于 Windows 上的 Connect:Direct 节点，等效参数为 **max.api.connects**。

如果 Connect:Direct 网桥执行大量文件传输的速率不够，那么可以增加 Connect:Direct 网桥代理处理的并行传输数。针对 Connect:Direct 网桥代理更改以下代理属性：

maxSourceTransfers

将该属性设置为大于 5 且小于等于 25 的值。如果选择大于 25 的值，那么除非增加可用于代理使用的 JVM 的内存量，否则该代理将耗尽内存。

maxDestinationTransfers

将该属性设置为大于 5 且小于等于 25 的值。如果选择大于 25 的值，那么除非增加可用于代理使用的 JVM 的内存量，否则该代理将耗尽内存。

ioThreadPoolSize

ioThreadPoolSize 的缺省值为 10。该属性限制 Connect:Direct 网桥代理充当源代理的传输的 Connect:Direct 节点 API 连接数。这些传输从 Connect:Direct 到 Managed File Transfer。使用以下指导信息来设置该属性的值：

- 如果 **maxSourceTransfers** 的值小于 **maxDestinationTransfers** 的值，请将 **ioThreadPoolSize** 设置为 **maxSourceTransfers** 值的 2 倍或 10 (两者中选用较大值)
- 如果 **maxSourceTransfers** 的值大于 **maxDestinationTransfers** 的值，请将 **ioThreadPoolSize** 设置为 **maxSourceTransfers** 和 **maxDestinationTransfers** 的总和

除这些代理属性外，还必须更改作为 Connect:Direct 网桥的一部分的 Connect:Direct 节点的最大并行 API 连接数。控制此数字的 Connect:Direct 参数为 **api.max.connects** (如果节点在 UNIX 上) 或 **max.api.connects** (如果节点在 Windows 上)。对相应的参数执行以下更改：

api.max.connects (如果 **Connect:Direct** 网桥中的节点在 UNIX 上)

将此参数设置为大于 **maxSourceTransfers** 和 **maxDestinationTransfers** 之和的值。

api.max.connects 参数的缺省值为 16。有关如何设置此参数的更多信息，请参阅 Connect:Direct 文档。

max.api.connects (如果 **Connect:Direct** 网桥中的节点在 Windows 上)

将此参数设置为大于 **maxSourceTransfers** 和 **maxDestinationTransfers** 之和的值。

max.api.connects 参数的缺省值为 10。有关如何设置此参数的更多信息，请参阅 Connect:Direct 文档。

相关任务

[配置 Connect:Direct 网桥](#)

相关参考

[MFT agent.properties 文件](#)

调试文件传输调用的 **Connect:Direct** 进程

您可以配置 Connect:Direct 网桥代理，以将文件传输调用的 Connect:Direct 进程相关日志信息写入 Connect:Direct 网桥代理配置目录中的 `output0.log` 文件。

关于此任务

要配置 Connect:Direct 进程的日志记录，请完成以下步骤：

过程

1. 停止 Connect:Direct 网桥代理。
2. 编辑 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 目录中的 `agent.properties` 文件以包含 `logCDProcess` 属性。
`logCDProcess` 属性可具有以下某个值：
 - None - 未记录任何信息。这是缺省值。
 - Failures - 记录了有关 Connect:Direct 进程失败的信息。
 - All - 记录了有关所有 Connect:Direct 进程的信息。
3. 启动 Connect:Direct 网桥代理。

结果

Connect:Direct 进程相关信息将记录到 Connect:Direct 网桥代理的 `output0.log` 文件中。记录的信息包括：

- MFT 传输标识
- Connect:Direct 进程名称
- Connect:Direct 进程号
- 生成的进程定义
- 如果 Connect:Direct 进程是用户定义的，那么还包含进程模板的文件名

相关概念

[第 110 页的『对 Connect:Direct 网桥进行故障诊断』](#)

使用以下参考信息和示例，以帮助您诊断 Connect:Direct 网桥返回的错误。

相关参考

[MFT agent.properties 文件](#)

MFT 一般故障诊断

使用以下参考信息可帮助您诊断 Managed File Transfer 中的错误：

相关概念

[第 121 页的『用于将 MFT 代理或记录器作为 Windows 服务运行的指南』](#)

您可以将 Managed File Transfer 代理、独立数据库记录器和独立文件记录器作为 Windows 服务运行。如果这些 Windows 服务出现任何问题，您可以使用服务日志文件以及本主题中的信息来诊断问题。

相关参考

[第 115 页的『MFT 常见问题』](#)

Managed File Transfer 网络中可能出现的常见问题。

[第 117 页的『在 MFT 代理进程消失但未记录任何诊断信息时该怎么做』](#)

在 UNIX 平台上，如果代理进程已消失，但代理日志文件不包含任何解释，这可能由于启动此代理的方式所致。

[第 118 页的『当 MFT 代理或记录器配置不安全时该怎么做』](#)

如果某个 Managed File Transfer 进程检测到某个配置文件包含敏感信息，是密钥库或信任密钥库文件，并且具有系统范围的读、写或删除许可权，那么如果是在启动时检测到该情况，该进程将无法启动。如果该情况不是在启动时而是在运行时检测到的，那么 Managed File Transfer 会生成警告消息并忽略该配置文件的内容。这与协议网桥和 Connect:Direct 网桥功能相关，如果在代理运行期间某个配置发生更改，这些功能会重新装入该配置。

[第 118 页的『如果消息在 SYSTEM.MANAGED.DURABLE 队列上聚集或者填充文件系统，那么该执行哪些操作？』](#)

如果 IBM MQ Explorer 插件在协调队列管理器上使用持久预订，那么可以在 SYSTEM.MANAGED.DURABLE 队列。如果您具有大容量 Managed File Transfer 网络，请不要频繁使用 IBM MQ Explorer 插件，否则此消息数据可能会填满本地文件系统。

[第 119 页的『发布前检查消息』](#)

由于代理可连接到 IBM WebSphere MQ 6.0 队列管理器，因此代理不使用 IBM WebSphere MQ 7.0 中引入的直接发布方法。事实上，代理将普通消息发送到包含 MQRFH 头的协调队列管理器。MQRFH 头请求发布消息的有效内容。这些消息会被发送到协调队列管理器上的 SYSTEM.FTE 队列，通常会立即从该队列发布这些消息。如果错误条件阻止此发布，您可以在尝试发布前检查队列上的消息以帮助进行诊断。可通过完成以下步骤来执行此操作：

[第 120 页的『配置 MFT 代理可再分发组件时可能出现的错误』](#)

配置 Redistributable Managed File Transfer Agent 时的错误消息

[第 121 页的『用于将 UAC 和虚拟存储器与 MFT 配合使用的指南』](#)

在 Windows Server 2008 R2 和其他类似操作系统中提供了用户帐户控制 (UAC)。这是一个安全基础结构，其功能之一是在中央 Program Files 目录中存储的用户数据转移到用户位置，这称为虚拟存储器。

[第 123 页的『用于更新代理或记录器 JVM 选项的指南』](#)

如果使用 **fteModifyAgent** 或 **fteModifyLogger** 命令的 **-sj** 参数通过更新，添加或删除 Java 系统属性来修改代理程序或记录器的现有 Windows 服务定义，那么将首先删除现有 Windows 服务，然后再在其位置创建新的服务，并使用新的 Windows 服务的属性更新代理程序或记录器属性文件。新的 Windows 服务定义必须与代理或记录器属性文件中定义并经过更新的 Windows 服务属性保持一致。

[第 123 页的『在 MFT 无法从 AMS 中的密钥库配置文件读取密钥库属性时该怎么做』](#)

如果缺省位置中未提供密钥库配置文件，那么必须由 **MQS_KEYSTORE_CONF** 变量指定该文件的位置，Java AMS 才能够以客户机方式运行。如果未指定此位置，那么 Managed File Transfer Agent 日志将显示以下错误消息：“Failed to read keystore properties from the keystore configuration file”。

[第 123 页的『BFGSS0023E 错误以及如何避免这些错误』](#)

如果从安装中卸载修订包以返回至产品的先前版本，并在发生卸载时受管传输涉及了与安装相关联的代理，那么该代理无法启动，并将报告 BFGSS0023E 错误。要避免此错误，您可以完成一些步骤以防止在重新启动代理时显示 BFGSS0023E 消息。

MFT 常见问题

Managed File Transfer 网络中可能出现的常见问题。

- 如果文本传输失败，并具有以下错误：

BFGI00060E: Text data conversion has failed

发生这种情况有两种原因：

1. 源文件中的一个或多个字符无法从源文件代码页转换到目标文件代码页。如果代码页具有不同字符集并且一些字符无法在这些字符集之间转换，那么可能会出现此问题。

如果您可以接受不对某些字符进行转换，那么可以在目标代理定义替换字符序列，以便传输不会失败。指定代理属性 **textReplacementCharacterSequence** 来定义替换字符序列。有关更多信息，请参阅高级代理属性。

2. 源文件编码与源代理的缺省编码不匹配。在这种情况下，使用缺省设置执行文本传输会损坏字符数据。如果要传输的源文件与源代理的编码不同，请执行以下某个步骤：

- a. 在传输定义文件中指定文件编码。有关更多信息，请参阅[使用传输定义文件](#)。
- b. 将 **-sce** 参数和 **fteCreateTransfer** 命令结合使用来指定文件编码。有关更多信息，请参阅主题 [fteCreateTransfer: 启动新的文件传输](#)。
- c. 在执行 Ant 移动或复制任务的过程中指定文件编码。有关更多信息，请参阅[将 Apache Ant 与 MFT 结合使用](#)。

要检查您是否为传输选择了正确的源文件编码，请执行以下步骤：

1. 将目标文件编码设置为 UTF-8。
2. 以文本方式传输文件。
3. 使用 UTF-8 文件查看器来查看文件的内容。如果文件中所有字符都正确显示，那么源文件编码正确。

- 如果您从 **fteCreateAgent** 命令中看到以下输出：

```
BFGMQ1007I: 无法联系协调队列管理器或已拒绝协调队列管理器
connection attempt.
IBM MQ 原因码是 2058。 将不会发布有关该代理是否存在的信息。
```

它指示协调队列管理器无法访问，并提供 IBM MQ 原因码以说明原因。此参考消息可能指示协调队列管理器当前不可用或者您未正确定义配置。

- 如果您在使用用户出口例程，并且在调用用户出口时或者在已经调用出口之后发生故障（例如，产品故障或断电），那么可能多次调用用户出口。
- 如果您的系统上具有一个包含队列管理器的代理，并且其 IP 地址由 DHCP 分配（而不是静态 IP 地址），并且代理使用客户机 TCP/IP 连接来连接到该系统，那么您必须在设置了以下系统环境变量的情况下启动代理：

- **Windows** 在 Windows 上：

```
set BFG_JVM_PROPERTIES="-Dsun.net.inetaddr.ttl=value"
```

- **UNIX** 在 UNIX 上：

```
export BFG_JVM_PROPERTIES="-Dsun.net.inetaddr.ttl=value"
```

其中，*value* 是指清除 JVM 的高速缓存 DNS 值所经历的时间间隔（单位为秒）。如果队列管理器系统的 IP 地址由于任何原因而重新分配（例如，由于网络中断、IP 租约过期或系统重新引导），那么代理会报告其与队列管理器丢失连接。清除 JVM DNS 高速缓存后，代理可以成功地重新连接。如果未设置环境变量，那么在不重新启动 JVM 的情况下，代理无法在此场景下重新连接。此行为的原因是 JVM 在内部对主机名的 IP 地址进行了高速缓存，并且缺省情况下不会刷新这些 IP 地址。

- 如果运行 **fteStartAgent** 命令并且看到以下错误消息，那么您的环境可能具有与 Managed File Transfer 冲突的附加库路径：

```
BFGCL0001E: An internal error has occurred. 异常为："CC=2;RC=2495;AMQ8568："
找不到本地 JNI 库 "mqjbnd"。[3=mqjbnd]
```

如果 LD_LIBRARY_PATH 或 LIBPATH 环境变量设置为先引用 64 位版本的库，然后引用 32 位版本的库，而同时正在使用 32 位版本的 Java 运行代理（对于大部分平台，都是这种情况），那么会发生此错误。

要解决此问题，请设置 Managed File Transfer 代理属性 javaLibraryPath 以引用库的正确位置。例如，对于 AIX 上的 mqjbnd，请设置为：/usr/mqm/java/lib。对于 Linux 上的 mqjbnd，请设置为：/opt/mqm/java/lib

- 如果您通过在代理属性文件中指定 authorityChecking=true 而启用了用户权限检查，并且所有权限检查都失败（即用户对相关权限队列具有必需权限）：
 - 请确保运行的代理的用户在代理队列管理器上具有 ALT_USER 访问控制。
- 如果您通过在代理属性文件中指定 authorityChecking=true 而启用了用户权限检查，并且 IBM MQ 错误消息写入到代理 output0.log 文件中，请执行以下某项操作：
 - 请忽略这些消息，代理不会受影响。
 - 为运行代理的用户授予 SYSTEM.FTE.AUTH* 队列（属性此代理）上的 GET 权限。
- 如果您已编辑了代理属性文件，并且代理尚未获取这些文件：
 - 重新启动代理，以确保代理能读取新属性。

z/OS

z/OS

- 如果您在 z/OS 上使用代理来传输到 PDS 或 PDSE 数据集并且发生异常终止，那么您的系统可能具有有限的磁盘空间。异常终止可能具有系统完成代码 B14 和返回码 0C，这表示没有任何剩余空间。如果是传输到连续数据集，那么传输会失败并表明空间不足这一情况，但是代理会仍然运行。
- 如果您在 z/OS 上使用代理，并且 WMQFTEP 任务生成了一些 Java 核心转储之后变为无响应，那么请应用 OMVS 系统服务 APAR OA43472。

- 如果在 z/OS 上运行配置或管理脚本时看到以下输出：

```
FSUM7332 syntax error: got (, expecting Newline
```

该输出指示在运行配置或管理脚本的环境中未设置环境变量 `_BPXK_AUTOCVT=ON`。有关此环境变量以及如何设置此环境变量的更多信息，请参阅 [z/OS 上的 MFT 的环境变量](#)。

MFT 中有关 JZOS 的常见问题

下面是在遇到 JZOS 问题时的一些建议。

- 如果 JZOS 无法成功处理，请执行下列操作：

- 在 JCL 中添加“`, PARM= '+T'`”。例如：

```
//MQMFT EXEC PGM=JVMLDM86,REGION=0M,PARM= '+T'
```

- 在环境文件中添加 `set -x`。

- 如果结果为：

```
JVMJZBL1038E 子 shell 进程退出，退出代码为 1
```

```
JVMJZBL1042E JZOS 批处理启动程序失败，返回码=102
```

这表示您的环境文件和 Managed File Transfer 命令有问题。这可能是由于指定的路径无效。

- 从您的环境文件中找到 **BFG_PROD** 的值。

1. 转至 OMVS 并使用 `ls -ltr` 命令。

例如，如果 **BFG_PROD** 为 `/HMF8800/`，请输入以下命令：

```
ls -ltr HMF8800/bin/fteBatch
```

2. 检查此文件是否存在，以及批处理作业是否有权读取此文件。

3. 解决所有问题。

- 如果 JCL 仍然无法正确处理：

1. 请在 USS 中创建一个文件（例如，`myenv`），然后使用编辑器将信息从环境文件复制到此 `myenv` 文件。

2. 保存该文件。

3. 在命令行中，使用命令 `chmod +x myenv` 以允许运行此文件。

4. 发出 `. myenv` 命令。注意，这是（点空格文件名）。

运行此命令将报告 `myenv` 文件中的任何错误。

5. 更正 `myenv` 和环境文件中的任何错误。

UNIX

在 MFT 代理进程消失但未记录任何诊断信息时该怎么做

在 UNIX 平台上，如果代理进程已消失，但代理日志文件不包含任何解释，这可能由于启动此代理的方式所致。

您可通过如下方式查看代理诊断信息：

- 检查代理的日志文件是否声明已停止此代理。
- 检查代理锁定文件 `agent.lck` 是否仍然存在。

例如，如果从 `shell` 脚本启动代理，那么当该脚本完成时（包括代理进程），将移除与该脚本关联的所有子进程。要在调用代理的脚本的持续时间过后使此代理保持运行，请完成以下步骤：

1. 在 `fteStartAgent` 命令前面添加 `nohup` 命令，使 `fteStartAgent` 进程（及所有子进程）从脚本中分离。

将来，当脚本终止时，代理会继续运行。

当 MFT 代理或记录器配置不安全时该怎么做

如果某个 Managed File Transfer 进程检测到某个配置文件包含敏感信息，是密钥库或信任密钥库文件，并且具有系统范围的读、写或删除许可权，那么如果是在启动时检测到该情况，该进程将无法启动。如果该情况不是在启动时而是在运行时检测到的，那么 Managed File Transfer 会生成警告消息并忽略该配置文件的内容。这与协议网桥和 Connect:Direct 网桥功能相关，如果在代理运行期间某个配置发生更改，这些功能会重新装入该配置。

请完成以下检查以确定问题原因：

1. 从提供的错误消息中确定已报告为不安全的配置文件。
2. 确保该文件的访问许可权与所需的需求相匹配。有关更多信息，请参阅 [MFT 用于访问敏感配置信息的许可权](#)。
3. 重新启动代理或记录器。或者，如果是协议网桥或 Connect:Direct 凭证文件，请等待下次重新装入。

示例

在此示例错误消息中，数据库记录器无法启动：

```
BFGDB0066E: The logger encountered a problem accessing its credentials file and will stop.  
Reported error: BFGNV0145E: The 'Everyone' group has access to the file 'C:\mqmftcredentials.xml'.
```

在此示例错误消息中，协议网桥代理无法启动：

```
BFGI00383E: The security permissions defined for credentials file 'C:\ProtocolBridgeCredentials.xml' do  
not meet the  
minimum requirements for a file of this type.  
Reported problem: BFGNV0145E: The 'Everyone' group has access to the file  
C:\ProtocolBridgeCredentials.xml'.
```

相关参考

[用于访问敏感配置信息的 MFT 许可权](#)

如果消息在 *SYSTEM.MANAGED.DURABLE* 队列上聚集或者填充文件系统，那么该执行哪些操作？

如果 IBM MQ Explorer 插件在协调队列管理器上使用持久预订，那么可以在 *SYSTEM.MANAGED.DURABLE* 队列。如果您具有大容量 Managed File Transfer 网络，请不要频繁使用 IBM MQ Explorer 插件，否则此消息数据可能会填满本地文件系统。

要除去 *SYSTEM.MANAGED.DURABLE* 队列上消息的聚集，可执行以下其中一种操作：

- 启动使用持久预订的 IBM MQ Explorer。IBM MQ Explorer 的 Managed File Transfer 插件使用队列中的消息。
- 手动从队列删除消息。

V 9.1.0

您可以通过以下某种方法避免在持久队列上积压消息：

- 指定 IBM MQ Explorer 插件使用对协调队列管理器的非持久预订。在 IBM MQ Explorer 中执行以下步骤：
 1. 选择窗口 > 首选项 > **IBM MQ Explorer > Managed File Transfer**
 2. 在传输日志订阅类型列表中，选择 *NON_DURABLE*。
- 清除 IBM MQ Explorer MFT 插件所创建的来自协调队列管理器的持久预订。

持久预订的名称将作为前缀，以显示该预订是由 IBM MQ Explorer MFT 插件、主机名和用户名称（例如，MQExplorer_MFT_Plugin_HOST_TJWatson）创建的。

相关任务

[保留 MFT 日志消息](#)

在 **SYSTEM.FTE** 队列

IBM MQ Managed File Transfer (MFT) 拓扑的协调队列管理器使用排队的发布/预订来处理状态发布并将其分发给订户。

队列管理器的发布/预订引擎使用发布流来监视 **SYSTEM.FTE** 队列。当它接收到一个时，它会制作它的副本以分发给订户。

在正常操作下，**SYSTEM.FTE** 队列应该为空或仅包含少量消息。如果队列深度继续增长，那么通常意味着发布/预订引擎不再使用发布流。如果最近重新创建了协调队列管理器，那么通常会发生此情况。

要解决此问题，您应该检查 **SYSTEM.QPUBSUB.QUEUE.NAMELIST** 名称列表是否已正确设置并包含 **SYSTEM.FTE** 队列。要执行此操作，请运行以下 MQSC 命令：

```
DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)
```

这将生成类似于以下示例的输出：

```
NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)
NAMCOUNT(3)
NAMES(SYSTEM.BROKER.DEFAULT.STREAM
      ,SYSTEM.BROKER.ADMIN.STREAM
      ,SYSTEM.FTE)
DESCR(A list of queues for the queued Pub/Sub interface to monitor)
ALTDATA(2022-03-04)                ALTTIME(14.34.37)
```

如果 **NAMES** 属性不包含 **SYSTEM.FTE**，那么可以使用以下 MQSC 命令添加该属性：

```
ALTER NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)
NAMES(SYSTEM.BROKER.DEFAULT.STREAM,SYSTEM.BROKER.ADMIN.STREAM,SYSTEM.FTE)
```

发布前检查消息

由于代理可连接到 IBM WebSphere MQ 6.0 队列管理器，因此代理不使用 IBM WebSphere MQ 7.0 中引入的直接发布方法。事实上，代理将普通消息发送到包含 **MQRFH** 头的协调队列管理器。**MQRFH** 头请求发布消息的有效内容。这些消息会被发送到协调队列管理器上的 **SYSTEM.FTE** 队列，通常会立即从该队列发布这些消息。如果错误条件阻止此发布，您可以在尝试发布前检查队列上的消息以帮助进行诊断。可通过完成以下步骤来执行此操作：

1. 在协调队列管理器中禁用发布/预订引擎。

您可以使用 IBM MQ Explorer 或 MQSC 命令来完成该步骤。请注意，如果协调队列管理器同样用于其他目的，那么这将暂时停止队列管理器上的所有发布/预订活动，包括与 Managed File Transfer 无关的活动。

IBM MQ Explorer:

- a. 在“导航器”视图中，右键单击协调队列管理器，然后选择**属性**。
- b. 从**属性**窗格中选择**发布/预订**。
- c. 从**发布/预订方式**列表中选择**兼容性**。

MQSC:

```
ALTER QMGR PSMODE(COMPAT)
```

2. 发送另一消息。

执行存在发布问题的 Managed File Transfer 操作。例如，对于代理注册，任何时候启动代理时都将发送消息（您无需重复删除和创建代理以生成注册消息）。由于已禁用发布/预订引擎，因此将不会进行发布。

3. 浏览协调队列管理器上的 **SYSTEM.FTE** 队列。

您应该使用 IBM MQ Explorer 来浏览协调队列管理器的 **SYSTEM.FTE** 队列。

IBM MQ Explorer:

- 在“导航器”视图中，展开协调队列管理器并单击**队列**。在“内容”视图中，右键单击 SYSTEM.FTE 队列并选择**浏览消息**。将打开**消息浏览器**窗口并显示将发布的消息。
- 用户标识**列显示消息描述符中所含的用户标识。发布失败的常见原因是该用户标识对于 SYSTEM.FTE 主题没有发布权限。
- 您可以通过右键单击消息并选择**属性**，查找有关每条消息（包括将发布的 XML）的更多信息。

没有用于检查消息内容的 MQSC 命令。如果您没有 IBM MQ Explorer，那么必须使用可用于浏览队列并显示所发现消息的所有方面的其他程序。您可以使用 **amqsbcg** 样本程序 (如果已安装)，如以下主题中所述: **浏览队列**。UserIdentifier 行显示用户标识。或者，您可以使用 **dmpmqmsg**；可以在如下行中找到消息的用户标识:

```
A RTM MQ24
A USR JOHND0E
A ACC 1A0FD4D8F2F4C3C8C9D5F1F9C6F7C1C3F3F00019F7AC3000000000000000000
```

该示例中的第二行是该消息的消息描述符用户标识。

4. 重新启用协调队列管理器的发布/预订引擎。

您可以使用 IBM MQ Explorer 或 MQSC 命令来完成该步骤。重新启用协调队列管理器中的发布/预订引擎后，会立即处理 SYSTEM.FTE 队列上的所有消息。

IBM MQ Explorer:

- 在“导航器”视图中，右键单击协调队列管理器，然后选择**属性**。
- 从**属性**窗格中选择**发布/预订**。
- 从**发布/预订方式**列表中选择**已启用**。

MQSC:

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

V 9.1.0 配置 MFT 代理可再分发组件时可能出现的错误

配置 Redistributable Managed File Transfer Agent 时的错误消息

无法装入 Windows 的本机库

Windows 您必须在系统上安装以下 Microsoft 库才能使用 Redistributable Managed File Transfer Agent:

- Microsoft Visual C++ Redistributable 2008
- Microsoft Visual C++ Redistributable 2012

这些库可从 Microsoft 中获取。请参阅 [The latest supported Visual C++ downloads](#)。

如果您在未安装这些库的情况下尝试运行 MFT 命令，将报告以下错误:

- BFGUB0070E: Internal error: Native library for platform Windows 7 (architecture amd64) could not be loaded because mqmft (Not found in java.library.path).
- BFGCL0043I: Specify the '-h' command line parameter to see more usage information.

请检查是否安装了 Microsoft 库。如果未安装这些库，请先安装库然后重新运行此命令。

不支持使用绑定模式

Redistributable Managed File Transfer Agent 只能在客户机模式下连接到 IBM MQ。如果您尝试在绑定模式下运行命令，将报告以下错误:

- BFGCL0408E: Unable to obtain IBM MQ installation information for queue manager 'queue manager name'. Reason Cannot run program "../bin/dspmq": error=2, No such file or directory

发出命令时，您必须提供队列管理器主机、端口、名称和通道名称。

用于将 UAC 和虚拟存储器与 MFT 配合使用的指南

在 Windows Server 2008 R2 和其他类似操作系统中提供了用户帐户控制 (UAC)。这是一个安全基础结构，其功能之一是在中央 Program Files 目录中存储的用户数据转移到用户位置，这称为虚拟存储器。

如果仅使用 Managed File Transfer 工具来管理数据结构，那么 Managed File Transfer 不受 UAC 和虚拟存储器的影响。但是，如果某个非 IBM MQ 管理员使用标准操作系统工具更改或重建了目录结构，那么新结构可能会转移到虚拟存储器。这可能会导致发生以下一种或多种情况：

- 用户（包括 IBM MQ 管理员）无法再在预期位置中看到文件。
- 代理可能无法启动，并报告消息 BFGCL0315，但不提供任何支持性原因码。
- 在代理报告的位置找不到任何日志文件。
- 代理在通过 **-F** 参数启动时可能无法启动，并报告如下消息：

```
The current directory is invalid
```

要更正上述所有情况：

- 以 IBM MQ 管理员身份使用 **fteDeleteAgent** 和 **fteCreateAgent** 命令来重建代理结构。
- 以操作系统管理员身份从受影响用户的虚拟存储器中移除 IBM MQ 条目。例如，在 Windows 上，虚拟存储器的位置如下：`%USERPROFILE%\AppData\Local\VirtualStore\`

相关参考

[fteDeleteAgent](#)

[fteCreateAgent](#)

Windows 用于将 MFT 代理或记录器作为 Windows 服务运行的指南

您可以将 Managed File Transfer 代理、独立数据库记录器和独立文件记录器作为 Windows 服务运行。如果这些 Windows 服务出现任何问题，您可以使用服务日志文件以及本主题中的信息来诊断问题。

要了解有关配置代理、独立记录器或独立文件记录器以作为 Windows 服务运行的更多信息，请参阅[将 MFT 代理作为 Windows 服务启动和 fteModifyLogger：将 MFT 记录器作为 Windows 服务运行](#)。

注：如果可再分发的代理程序将作为 Windows 服务运行，那么需要在系统环境中设置 **BFG_DATA** 环境变量才能使该服务正常工作。

日志文件的位置

使用 **fteCreateAgent**、**fteCreateCDAgent**、**fteCreateBridgeAgent**、**fteModifyAgent**、**fteCreateLogger** 或 **fteModifyLogger** 命令将代理或记录器作为 Windows 服务运行时，可以使用 **-sl** 参数来选择日志记录级别。此参数的可能值是 `error`、`info`、`warn` 和 `debug`。缺省值为 `info`。

Windows 服务的日志文件具有文件名 `servicedate.log`，其中 `date` 是服务的启动日期。代理的文件将写入 `MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\agents\agent_name` 目录中。此目录就是 Managed File Transfer Agent 跟踪文件写入的同一目录。记录器的文件将写入 `MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\loggers\logger_name` 目录中。

如果在启动代理或作为 Windows 服务的独立记录器时遇到问题，请尝试使用 **-sl** 参数将日志记录级别设置为 `debug`。其他信息将写入 `servicedate.log` 文件。

注：当日志记录级别设置为 `debug` 时，您用来运行 Windows 服务的用户帐户和密码将以纯文本形式显示在日志文件中。

日志文件的数目

使用 `fteCreateAgent`, `fteCreateCDAgent`, `fteCreateBridgeAgent`, `fteModifyAgent`, `fteCreateLogger` 或 `fteModifyLogger` 命令将代理程序或独立记录器作为 Windows 服务运行时, 可以使用 `-sj` 参数来选择日志文件数。在命令中指定以下文本以更改日志文件数: `-sj -Dcom.ibm.wmqfte.daemon.windows.windowsServiceLogFiles=number`, 其中 `number` 是您要表示为正整数的日志文件数。如果不指定日志文件的数目, 那么缺省值为 5。

“作为服务登录”权限

您用于运行该服务的 Windows 帐户必须具有 **Log on as a service** 权限。如果您尝试使用 `fteStartAgent`、`fteStartLogger` 命令或 Windows 的 `Sc.exe` 命令启动服务, 但是您使用的用户帐户不具有此权限, 那么这将打开“服务”窗口。假如您希望启动的服务是用于运行代理, 那么此窗口包含以下消息:

```
Unable to start Windows service mqmftAgentAGENT@QMGR.  
System error 1069: The service did not start due to a logon failure.
```

在此消息中, `AGENT` 为代理的名称, `QMGR` 为代理队列管理器名称。如果您尝试作为服务来运行独立记录器, 那么将生成一个类似消息, 此消息引用的是记录器, 而非代理。

要防止发生此错误, 请为用于运行服务的 Windows 帐户授予 **Log on as a service** 权限。例如, 在 Windows 10 上完成以下步骤:

1. 从开始菜单, 单击管理工具 > 本地安全策略。
2. 在安全性设置窗格中, 展开本地策略, 然后单击用户权限分配。
3. 在策略和安全性设置窗格中, 双击作为服务登录。
4. 单击添加用户或组, 然后将您要运行该服务的用户添加到具有 **Log on as a service** 权限的用户列表中。您在运行 `fteCreateAgent`, `fteCreateCDAgent`, `fteCreateBridgeAgent`, `fteModifyAgent`, `fteCreateLogger` 或 `fteModifyLogger` 命令时提供了此用户名。

注: 错误 System error 1069: The service did not start due to a logon failure. 也可能由不正确的密码引起。

隐藏您的 Windows 帐户密码

将代理或独立记录器配置为作为 Windows 服务运行时, 您将指定要使用的用户名和密码。在以下示例中, 将创建代理 `AGENT1` (此代理具有代理队列管理器 `QMGR1`) 并且配置为作为 Windows 服务运行:

```
fteCreateAgent -agentName AGENT1 -agentQMgr QMGR1 -s -su fteuser -sp ftepassword
```

在此示例中, Windows 服务使用用户名 `fteuser` 运行, 该用户名具有关联的密码 `ftepassword`。运行 `fteCreateAgent` 命令或其他一些接受 `-s` 参数的命令时, 您将以纯文本形式为 Windows 帐户指定密码。如果偏好于不显示密码, 请执行以下步骤:

1. 运行命令 (`fteCreateAgent`, `fteCreateCDAgent`, `fteCreateBridgeAgent`, `fteModifyAgent`, `fteCreateLogger` 或 `fteModifyLogger`) 而不指定 `-sp` 参数。例如:

```
fteCreateAgent -agentName AGENT1 -agentQMgr QMGR1 -s -su fteuser
```

注: 此命令会产生一条消息, 警告您必须使用 Windows“服务”工具设置密码, 然后服务才能成功启动。

2. 打开 Windows“服务”窗口。
3. 在服务列表中, 右键单击代理或独立记录器服务, 然后选择属性。代理程序服务显示名称为 `Managed File Transfer Agent AGENT @ QMGR`, 其中 `AGENT` 是代理程序名称, `QMGR` 是代理程序队列管理器名称。记录器服务显示名称为 `Managed File Transfer Logger for property set coordination_qmgr_name`, 其中 `coordination_qmgr_name` 是您为要用作其属性集的独立记录器指定的协调队列管理器。有关属性集合的更多信息, 请参阅 [fteStartLogger](#) 和 [fteModifyLogger](#)。

4. 在属性窗口中，选择登录选项卡。
5. 在密码和确认密码字段中，输入运行服务的用户帐户的密码。输入密码时，密码字符是隐藏的。
6. 单击确定。

已知问题

使用 JAVA_HOME 系统环境变量时发生问题 (仅适用于 IBM WebSphere MQ 7.5.0 Fix Pack 1 或更低版本中的 Managed File Transfer)。

不得设置 JAVA_HOME 系统环境变量，否则代理或记录器 Windows 服务不大可能启动。代理或记录器 Windows 服务必须与 IBM MQ Java 运行时一起运行。

Windows 用于更新代理或记录器 JVM 选项的指南

如果使用 **fteModifyAgent** 或 **fteModifyLogger** 命令的 **-sj** 参数通过更新，添加或删除 Java 系统属性来修改代理程序或记录器的现有 Windows 服务定义，那么将首先删除现有 Windows 服务，然后再在其位置创建新的服务，并使用新的 Windows 服务的属性更新代理程序或记录器属性文件。新的 Windows 服务定义必须与代理或记录器属性文件中定义并经过更新的 Windows 服务属性保持一致。

从 IBM MQ 9.0.0 Fix Pack 4 开始，将在 APAR IT22423 下添加其他检查，以便验证使用 **fteModifyAgent** 或 **fteModifyLogger** 命令的 **-sj** 参数对代理程序或记录器的 JVM 选项进行的任何更新，以确保正确指定了这些选项。如果发现属性无效或者无法验证，那么 **fteModifyAgent** 或 **fteModifyLogger** 命令将失败并显示相应的错误消息。

如果 JVM 属性有效并且成功删除现有 Windows 服务，但在 **fteModifyAgent** 或 **fteModifyLogger** 命令创建新的 Windows 服务时出现故障，那么此命令会尝试从代理或记录器属性文件中移除用于定义替换 Windows 服务的属性。在此情况下，会返回一些错误消息，用于说明无法修改代理或记录器，已删除旧的 Windows 服务，但无法创建新的 Windows 服务，因此代理或记录器将无法作为 Windows 服务运行。然后，您必须手动验证 Windows 服务定义的状态与代理或记录器属性文件中定义的 Windows 服务属性是否一致，并执行相应的操作来纠正任何不一致情况。

相关参考

[fteModifyAgent](#): 将 MFT 代理作为 Windows 服务运行

[fteModifyLogger](#): 将 MFT 记录器作为 Windows 服务运行

在 MFT 无法从 AMS 中的密钥库配置文件读取密钥库属性时该怎么做

如果缺省位置中未提供密钥库配置文件，那么必须由 `MQS_KEYSTORE_CONF` 变量指定该文件的位置，Java AMS 才能够以客户机方式运行。如果未指定此位置，那么 Managed File Transfer Agent 日志将显示以下错误消息：“Failed to read keystore properties from the keystore configuration file”。

密钥库配置文件的缺省位置为 `home_directory/.mqsc/keystore.conf`。如果密钥库配置文件的位置不是缺省位置，请完成以下步骤：

1. 以客户机方式启动 FTE 代理。
2. 将 AMS 安全性应用于 `SYSTEM.FTE.DATA.<agent name>` 队列。如果密钥库配置文件不在此位置中，那么所有传输都将失败，并且不进行任何确认。
3. 针对 **fteStartAgent** 命令，将系统变量 `BFG_JVM_PROPERTIES` 设置为 `BFG_JVM_PROPERTIES=-DMQS_KEYSTORE_CONF=path to keystore_config file`。
4. 针对 **fteStartAgent** 命令，将系统变量 `MQS_KEYSTORE_CONF` 设置为 `MQS_KEYSTORE_CONF=path to keystore_config file`。凡是代理，不管其运行方式如何，必须设置此变量才能确保可运行。

注: 如果 Java AMS 以绑定方式运行，并且密钥库配置文件不在缺省位置中，那么队列管理器的错误日志中将显示错误 AMQ9062。

BFGSS0023E 错误以及如何避免这些错误

如果从安装中卸载修订包以返回至产品的先前版本，并在发生卸载时受管传输涉及了与安装相关联的代理，那么该代理无法启动，并将报告 BFGSS0023E 错误。要避免此错误，您可以完成一些步骤以防止在重新启动代理时显示 BFGSS0023E 消息。

对于当前涉及代理的正在进行中的每个受管传输，在代理的 `SYSTEM.FTE.STATE.agent_name` 队列上有一条消息。此消息在受管传输上存储检查点信息，并且在受管传输进入恢复时使用。在受管传输完成后，将除去 `SYSTEM.FTE.STATE.agent_name` 队列上的相应消息。

每条状态消息都包含一些内部头信息，指示在受管传输运行时代理所使用的 Managed File Transfer 组件版本。版本信息会显示具体的修订包级别，因此，举例来说，如果 IBM MQ 8.0.0 Fix Pack 5 代理在运行受管传输，那么该受管传输的状态消息将包含对 IBM MQ 8.0.0 Fix Pack 5 的引用。

如果从安装中卸载修订包，并且与该安装相关联的代理关联了正在进行的传输，那么该代理无法启动，并将报告以下错误：

```
BFGSS0023E: The agent is configured to use IBM MQ queues that contain data created using a later version of the product. The agent cannot run in this configuration and will end.
```

例如，如果 IBM MQ 8.0.0 Fix Pack 5 代理在停止时运行了一些正在进行的传输，然后降级到 IBM MQ 8.0.0 Fix Pack 4 级别，那么下次启动该代理时，它将检查其 `SYSTEM.FTE.STATE.agent_name` 队列并发现它们是在使用 IBM MQ 8.0.0 Fix Pack 5 时编写的。由于代理现在使用的是 IBM MQ 8.0.0 Fix Pack 4，因此它会报告先前段落中描述的 BFGSS0023E 错误并关闭自身。

作为一般规则，如果您要除去任一 Managed File Transfer 组件的修订包，那么完成以下步骤应该会防止在重新启动代理时显示 BFGSS0023E 消息：

1. 确保其所有代理已完成其受管传输。
2. 停止这些代理。
3. 除去修订包。
4. 重新启动这些代理。

相关任务

[启动 MFT 代理](#)

[在 UNIX 上将队列管理器还原为先前版本](#)

[在 Windows 上将队列管理器还原为先前版本](#)

相关参考

[MFT 代理队列设置](#)

[BFGSS0001 - BFGSS9999](#)

对消息问题进行故障诊断

未送达消息故障诊断

使用此处提供的建议，帮助您在未成功送达消息时解决问题。

- **场景：**消息未如您预期到达队列。
- **说明：**由于某种原因无法送达的消息将放入死信队列中。
- **解决方案：**可通过发出 `MQSC DISPLAY QUEUE` 命令来检查队列是否包含任何消息。

如果队列包含消息，可使用所提供的浏览样本应用程序 (`amqsbcg`) 来使用 `MQGET` 调用浏览队列上的消息。样本应用程序为已命名队列管理器单步遍历已命名队列上的所有消息，为已命名队列上的所有消息显示消息描述符和消息上下文字段。

根据将消息放在队列上的原因，您必须确定如何处置在死信队列上找到的任何消息。如果您没有将死信队列与每个队列管理器关联起来，那么可能会发生问题。

有关死信队列以及处理未送达消息的更多信息，请参阅[处理死信队列](#)。

Windows

Linux

AIX

对 MQ Telemetry 问题进行故障诊断

查找故障诊断任务以帮助解决有关运行 MQ Telemetry 应用程序的问题。

相关概念

[MQ Telemetry](#)

查找 MQ Telemetry 使用的日志、错误日志和配置文件。

注: 这些示例是针对 Windows 系统进行编码的。请更改语法以在 AIX 或 Linux 系统上运行示例。

服务器端日志

遥测 (MQXR) 服务将 FDC 文件写入到 IBM MQ 错误目录:

```
WMQ data directory\errors\AMQ nnn.n.FDC
```

FDC 文件的格式为 MQXRn.FDC。

它还会编写遥测 (MQXR) 服务日志。日志路径为:

```
WMQ data directory\Qmgrs\qMgrName\errors\mqxr.log
```

该日志文件的格式为 mqxr_n.log。

IBM MQ Explorer 创建的 IBM MQ 遥测样本配置使用 *WMQ Telemetry installation directory\bin* 中的命令 **runMQXRService** 启动遥测 (MQXR) 服务。此命令写入到:

```
WMQ data directory\Qmgrs\qMgrName\mqxr.stdout  
WMQ data directory\Qmgrs\qMgrName\mqxr.stderr
```

服务器端的配置文件

遥测通道和遥测 (MQXR) 服务

限制: 在将来的发行版中, 可能会更改遥测通道配置文件的格式、位置、内容和解释。您必须使用 IBM MQ Explorer 或 MQSC 命令来配置遥测通道。

IBM MQ Explorer 会在 Windows 系统上的 `mqxr_win.properties` 文件中保存遥测配置, 并在 AIX 或 Linux 系统上保存 `mqxr_unix.properties` 文件。属性文件保存在遥测配置目录中:

```
WMQ data directory\Qmgrs\qMgrName\mqxr
```

图 3: Windows 上的遥测配置目录

```
/var/mqm/qmgrs/qMgrName/mqxr
```

图 4: AIX 或 Linux 上的遥测配置目录

JVM

将作为参数传递的 Java 属性设置为 `java.properties` 文件中的遥测 (MQXR) 服务。该文件中的属性会直接传递到运行遥测 (MQXR) 服务的 JVM。它们在 Java 命令行上作为附加 JVM 属性进行传递。命令行上设置的属性优先于从 `java.properties` 文件添加到命令行的属性。

在与遥测配置相同的文件夹中查找 `java.properties` 文件。请参阅 [第 125 页的图 3](#) 和 [第 125 页的图 4](#)。

通过将每个属性指定为单独的行来修改 `java.properties`。完全按照需要格式化每个属性, 以将属性作为自变量传递到 JVM。例如:

```
-Xmx1024m  
-Xms1024m
```

JAAS

遥测通道 [JAAS 配置](#) 中描述了 JAAS 配置文件，其中包含 MQ Telemetry 随附的样本 JAAS 配置文件 `JAAS.config`。

如果您配置 JAAS，那么您几乎必然要编写一个类以认证用户，从而替换标准 JAAS 认证过程。

要在遥测 (MQXR) 服务类路径使用的类路径中包含 `Login` 类，请提供 `IBM MQ service.env` 配置文件。

在 `service.env` 中设置 JAAS `LoginModule` 的类路径。您无法在 `service.env` 中使用变量 `%classpath%`。`service.env` 中的类路径已添加至遥测 (MQXR) 服务定义中已设置的类路径。

通过将 `echo set classpath` 添加到 `runMQXRService.bat` 显示遥测 (MQXR) 服务正在使用的类路径。输出将发送到 `mqxr.stdout`。

`service.env` 文件的缺省位置为：

```
WMQ data directory\service.env
```

使用以下位置中每个队列管理器的 `service.env` 文件来覆盖这些设置：

```
WMQ data directory\Qmgrs\qMgrName\service.env
```

```
CLASSPATH= WMQ Installation Directory\mqxr\samples\samples
```

注：`service.env` 不得包含任何变量。请替换 `WMQ Installation Directory` 的实际值。

图 5: `service.env` for Windows 样本

跟踪

请参阅 [第 127 页的『跟踪遥测 \(MQXR\) 服务』](#)。用于配置跟踪的参数存储在两个文件中：

```
WMQ data directory\Qmgrs\qMgrName\mqxr\trace.config  
WMQ data directory\Qmgrs\qMgrName\mqxr\mqxrtraceOn.properties
```

并且存在对应的文件：

```
WMQ data directory\Qmgrs\qMgrName\mqxr\mqxrtraceOff.properties
```

客户端日志文件和客户端配置文件

有关最新信息和下载，请参阅以下资源：

- [Eclipse Paho 项目](#) 和 [MQTT.org](#) 具有可免费下载的最新遥测客户机和一系列编程语言的样本。使用这些站点可帮助您开发样本程序，以便发布和预订 IBM MQ Telemetry Transport，以及添加安全性功能。
- IBM Messaging Telemetry Clients SupportPac 不再可供下载。如果具有先前下载的副本，那么其中包含以下内容：
 - IBM Messaging Telemetry Clients SupportPac 的 MA9B 版本包含了已编译的样本应用程序 (`mqttv3app.jar`) 和关联的客户机库 (`mqttv3.jar`)。它们已在以下目录中提供：
 - `ma9c/SDK/clients/java/org.eclipse.paho.sample.mqttv3app.jar`
 - `ma9c/SDK/clients/java/org.eclipse.paho.client.mqttv3.jar`
 - 在此 SupportPac 的 MA9C 版本中，已除去 `/SDK/` 目录和内容：
 - 仅提供了样本应用程序 (`mqttv3app.jar`) 的源。它位于以下目录中：

```
ma9c/clients/java/samples/org/eclipse/paho/sample/mqttv3app/*.java
```
 - 仍然提供了已编译的客户机库。它位于以下目录中：

Windows Linux AIX 跟踪遥测 (MQXR) 服务

利用 IBM MQ 遥测 (MQXR) 服务所提供的跟踪功能，可以帮助 IBM 支持人员诊断与该服务有关的客户问题。

关于此任务

可使用两种方法来控制 IBM MQ 遥测服务的跟踪：

- 使用 **strmqtrc** 和 **endmqtrc** 命令（用于启动和停止跟踪）。使用 **strmqtrc** 命令启用跟踪后，将生成正在运行 IBM MQ 遥测服务的整个队列管理器的跟踪信息。这包含 IBM MQ 遥测服务自身，以及该服务用于与其他队列管理器组件进行通信的底层 Java 消息排队接口 (JMQUI)。

V 9.1.5 从 IBM MQ 9.1.5 开始，您还可以生成选定的感兴趣区域的跟踪信息。

- 运行 **controlMQXRChannel** 命令。请注意，使用 **controlMQXRChannel** 命令开启跟踪将仅跟踪 IBM MQ 遥测服务。

如果您不确定要使用的选项，请联系 IBM 支持代表，他们将能够提出最佳方式建议，帮助您为所看到的问题收集跟踪信息。

过程

1. 方法 1

- a) 打开命令提示符，浏览至以下目录：

```
MQ_INSTALLATION_PATH\bin
```

- b) 运行 **strmqtrc** 命令来启用跟踪：

对于 Long Term Support 和 IBM MQ 9.1.5 之前的 Continuous Delivery，请运行以下命令：

```
strmqtrc -m qmgr_name
```

其中，*qmgr_name* 是正在运行 IBM MQ MQXR 服务的队列管理器的名称。

V 9.1.5 从 IBM MQ 9.1.5，运行以下命令：

```
strmqtrc -m qmgr_name -t mqxr
```

其中，*qmgr_name* 是正在运行 IBM MQ MQXR 服务的队列管理器的名称，而 **-t mqxr** 将跟踪输出仅限于 MQXR 服务。

- c) 重现该问题。
- d) 通过运行以下命令来停止跟踪：

```
endmqtrc -m qmgr_name
```

2. 方法 2

- a) 打开命令提示符，浏览至以下目录：

```
MQ_INSTALLATION_PATH\mqxr\bin
```

- b) 运行以下命令来启用跟踪：

• **Windows**

```
controlMQXRChannel -qmgr=qmgr_name -mode=starttrace [clientid=ClientIdentifier]
```

• **Linux** **UNIX**

```
./controlMQXRChannel.sh -qmgr=qmgr_name -mode=starttrace [clientid=ClientIdentifier]
```

其中, *qmgr_name* 是正在运行 MQXR 服务的队列管理器的名称。

将 *ClientIdentifier* 设置为 MQTT 客户机的客户机标识。如果指定了 **clientid** 参数, 那么 IBM MQ 遥测服务跟踪将仅捕获具有该客户机标识的 MQTT 客户机的活动。

如果要跟踪多个特定 MQTT 客户机的 IBM MQ 遥测服务活动, 那么可以多次运行该命令, 每次都指定不同的客户机标识。

- c) 重现该问题。
- d) 在发生问题时, 通过运行以下命令来停止跟踪:

• **Windows**

```
controlMQXRChannel -qmgr=qmgr_name -mode=stoptrace
```

• **Linux** **UNIX**

```
./controlMQXRChannel.sh -qmgr=qmgr_name -mode=stoptrace [clientid=ClientIdentifier]
```

其中, *qmgr_name* 是正在运行 MQXR 服务的队列管理器的名称。

结果

要查看跟踪输出, 请切换至以下目录:

- **Windows** MQ_DATA_PATH\trace.
- **Linux** **UNIX** /var/mqm/trace.

包含 MQXR 服务信息的跟踪文件被称为 *mqxr_N.trc*, 其中 *N* 是数字。

V 9.1.5 从 IBM MQ 9.1.5 开始, 将按照以下方式来命名跟踪文件:

- 包含 MQXR 服务信息的跟踪文件被称为 *mqxrRunMQXRService_PPPPP.N.trc*, 其中 *PPPPP* 是 MQXR 服务的进程标识, *N* 是数字。
- 包含来自 **controlMQXRChannel** 命令的信息的跟踪文件被命名为 *mqxrControlMQXRChannel_PPPPP.N.trc* (其中 *PPPPP* 是 MQXR 服务的进程标识, *N* 是编号)。

由 JMQUI 生成的跟踪信息将写入名为 *mqxr_PPPPP.trc* 的跟踪文件, 其中 *PPPPP* 是 MQXR 服务的进程标识。

相关参考

[strmqtrc](#)

Windows **Linux** **AIX** 使用 **controlMQXRChannel** 命令进行的其他诊断

使用 **controlMQXRChannel** 命令来提供有关 MQXR 服务的其他诊断信息。

过程

运行以下命令以提供来自 MQXR 服务的有用诊断信息:

```
<MQ_INSTALLATION_PATH>\mqxr\bin\controlMQXRChannel -qmgr=<QMGR_NAME> -mode=diagnostics  
-diagnosticstype=<number>
```

生成的诊断信息取决于 **-diagnosticstype=<number>** 参数的值:

-diagnosticstype= 0

写入到控制台的线程转储

-diagnosticstype= 1

具有某些内部服务统计信息的 FDC

-diagnosticstype= 2

具有内部统计信息的 FDC, 以及有关当前已连接的客户机的信息

-diagnosticstype= 3

堆转储

-diagnosticstype= 4

Javacore

-diagnosticstype= 5

完整系统转储

-diagnosticstype= 6

有关特定客户机的详细信息。请注意，您还必须提供 **-clientid** 参数来表示该客户机。

Windows Linux AIX 解决问题：MQTT 客户机未连接

解决 MQTT 客户机程序未能连接到遥测 (MQXR) 服务的问题。

开始之前

此问题发生于服务器、客户机还是连接？您是否已使用 C 或 Java IBM MQTT 客户机编写自己的 MQTT v3 协议处理客户机或 MQTT 客户机应用程序？

请参阅验证 [MQ Telemetry 的安装](#) 以获取进一步信息，并检查遥测通道和遥测 (MQXR) 服务是否在正常运行。

关于此任务

有许多原因会造成 MQTT 客户机可能未连接到遥测服务器，或者使您可能推断其尚未连接到遥测服务器。

过程

1. 请考虑可以从遥测 (MQXR) 服务返回至 `MqttClient.Connect` 的原因码中得到哪些推论。它是哪种类型的连接故障？

| 选项 | 描述 |
|---|---|
| REASON_CODE_INVALID_PROTOCOL_VERSION | 请确保套接字地址对应于遥测通道，并且您未对另一个代理使用同一个套接字地址。 |
| REASON_CODE_INVALID_CLIENT_ID | 检查客户机标识是否不超过 23 个字节，并且仅包含范围内的字符: A-Z, a-z, 0-9, '._/% |
| REASON_CODE_SERVER_CONNECT_ERROR | 请检查遥测 (MQXR) 服务和队列管理器是否正常运行。使用 netstat 来检查是否未将套接字地址分配给另一个应用程序。 |

如果您已编写 MQTT 客户机库而不是使用 MQ Telemetry 提供的库之一，请查看 CONNACK 返回码。

从这三个错误中您可以推断客户机已连接至遥测 (MQXR) 服务，但该服务已发现错误。

2. 请考虑当遥测 (MQXR) 服务未作响应时，可以从客户机生成的原因码中得到哪些推论：

| 选项 | 描述 |
|--|---|
| REASON_CODE_CLIENT_EXCEPTION REASON_CODE_CLIENT_TIMEOUT | 在服务器中查找 FDC 文件；请参阅第 125 页的『 服务器端日志 』。当遥测 (MQXR) 服务检测到客户机发生超时时，它会编写一个首次故障数据捕获 (FDC) 文件。每当连接意外断开时，它就会编写一个 FDC 文件。 |

遥测 (MQXR) 服务可能不会对客户机以及客户机到期时的超时情况做出响应。仅在应用程序已设置无限期超时的情况下，MQ Telemetry Java 客户机才会挂起。在由于未经诊断的连接问题而导致为 `MqttClient.Connect` 设置的超时到期之后，客户机就会抛出这些异常之一。

除非您找到了与连接故障相关联的 FDC 文件，否则您无法推断出客户机已尝试连接至服务器：

- a) 确认客户机是否已发送连接请求。
使用工具 (例如 **tcpmon**) 检查 TCPIP 请求, 该工具可从 <https://code.google.com/p/tcpmon/> 获取 (例如)
 - b) 客户机所使用的远程套接字地址与为遥测通道定义的套接字地址相匹配吗?
随 IBM MQ Telemetry 提供的 Java SE MQTT 客户机中的缺省文件持久性类将创建名为 *clientIdentifier-tcphostNameport* 或客户机工作目录中的 *clientIdentifier-sslhostNameport* 的文件夹。此文件夹名称告诉您在连接尝试中所使用的 *hostName* 和 *port*。; 请参阅第 126 页的『客户端日志文件和客户端配置文件』。
 - c) 您可以对远程服务器地址执行 ping 操作吗?
 - d) 服务器上执行的 **netstat** 命令指出遥测通道正在客户机要连接至的端口上运行吗?
3. 请检查遥测 (MQXR) 服务是否发现客户机请求问题。
遥测 (MQXR) 服务会将其检测到的错误写入 *mqxr_n.log* 中, 队列管理器会将相关错误写入 *AMQERR01.LOG* 中; 请参阅
 4. 尝试通过运行另一个客户机来找出问题。
请参阅[验证 MQ Telemetry 的安装](#)以获取进一步信息
在服务器平台上运行样本程序, 以消除网络连接的不确定性, 然后在客户机平台上运行样本。
 5. 要检查的其他事项:
 - a) 是否有数万个 MQTT 客户机同时尝试连接?
遥测通道有一个队列, 用来缓存储备的入局连接。一秒钟处理超过 10,000 个连接。可通过使用 IBM MQ Explorer 中的“遥测通道”向导来配置储备缓冲区的大小。它的缺省大小是 4096。请检查是否尚未将储备配置为一个较小值。
 - b) 遥测 (MQXR) 服务和队列管理器是否仍在运行?
 - c) 客户机已连接至一个已切换其 TCPIP 地址的高可用性队列管理器吗?
 - d) 防火墙将有选择地过滤出站或者返回数据包吗?

Windows Linux AIX 解决问题: MQTT 客户机连接已断开

了解在成功建立连接并且运行较短或较长一段时间之后导致客户机抛出意外的 `ConnectionLost` 异常的原因。

开始之前

MQTT 客户机已成功连接。客户机可能已启动较长一段时间。如果不同客户机之间的启动时间间隔较短, 那么从成功连接到连接断开之间的时间可能较短。

区分“已断开的连接”与“已成功建立连接、但是稍后又断开的连接”并不难。已断开的连接由 MQTT 客户机调用 `MqttCallback.ConnectionLost` 方法来进行定义。仅在已成功建立连接之后才调用此方法。症状不同于 `MqttClient.Connect` 在接收到否定确认信息或者超时之后抛出异常。

如果 MQTT 客户机应用程序未使用 IBM MQ 提供的 MQTT 客户机库, 那么症状取决于客户机。在 MQTT v3 协议中, 症状是缺少对向服务器发出的请求的及时响应, 或者 TCP/IP 连接失败。

关于此任务

MQTT 客户机调用 `MqttCallback.ConnectionLost` 并产生可抛出异常, 以作为对接收肯定连接确认后遇到的任何服务器端问题的响应。当 MQTT 客户机从 `MqttTopic.publish` 和 `MqttClient.subscribe` 返回时, 请求会传输到负责发送和接收消息的 MQTT 客户机线程。通过将可抛出异常传递至 `ConnectionLost` 回调方法, 从而以异步方式报告服务器端错误。

过程

1. 是否已启动使用了同一 `ClientIdentifier` 的另一客户机?

如果使用同一 ClientIdentifier 启动另一客户机，或重新启动同一客户机，那么将断开与第一个客户机的首次连接。

2. 是否客户机访问了一个未授权它发布或预订的主题？

如果遥测服务代表客户机执行的任何操作返回 MQCC_FAIL，那么将导致此服务断开客户机连接。

原因码未返回给客户机。

- 在 mqxr.log 和 AMQERR01.LOG 文件中查找客户机连接到的队列管理器的日志消息；请参阅 [第 125 页的『服务器端日志』](#)。

3. TCP/IP 连接是否已断开？

防火墙可能具有较低的超时设置，以将 TCPIP 连接标记为不活动状态并且断开连接。

- 使用 MqttConnectOptions.setKeepAliveInterval 来缩短不活动 TCPIP 连接时间。

Windows Linux AIX 解决问题：MQTT 应用程序中丢失消息

解决“丢失消息”这一问题。消息是非持久消息、发送至错误的位置还是从未发送？错误编写的客户机程序可能会丢失消息。

开始之前

您有多肯定您所发送的消息已丢失？您可以推断是因为未收到消息而丢失了此消息吗？如果消息是发布，而此消息丢失，那么该消息由发布者发送，还是发送到订户？或者是因为预订丢失了，而代理未将该预订的发布发送至订户？

如果解决方案涉及到使用集群或者发布/预订层次结构的分布式发布/预订，那么有许多配置问题可能会导致丢失消息。

如果已发送具有至少一次或至多一次服务质量的消息，那么您认为丢失的消息有可能没有按预期方式传递。不太可能已经从系统中错误地删除了此消息。有可能未能创建您期望的发布或预订。

在确定“丢失消息”这一问题时，您执行的最重要的步骤是确认此消息是否确实已丢失。重现该场景，丢失更多消息。使用至少一次或至多一次服务质量来消除系统废弃消息的所有情况。

关于此任务

可以通过四条途径来诊断“丢失消息”这一问题。

1. 发出消息之后无需等待应答消息按设计那样工作。系统有时候会废弃发出消息之后无需等待应答消息。
2. 配置：在分布式环境中使用正确的权限来设置发布/预订并不简单。
3. 客户机编程错误：消息传送的责任不仅仅是 IBM 编写的代码的责任。
4. 如果已经穷尽了所有这些可能性，您可能会决定寻求 IBM 支持。

过程

1. 如果丢失的消息具有发出消息之后无需等待应答服务质量，请设置至少一次或至多一次服务质量。尝试再次丢失此消息。
 - 在许多情况下，IBM MQ 会丢弃使用触发和忘记服务质量发送的消息：
 - 通信中断，并且通道已停止。
 - 队列管理器已关闭。
 - 消息数过多。
 - 发出消息之后无需等待应答消息的传递依赖于 TCP/IP 的可靠性。TCP/IP 将继续反复发送数据包，直到传递的数据包获得确认为止。如果 TCP/IP 会话已中断，那么服务质量为发出消息之后无需等待应答的消息就会丢失。当客户机或服务器停机、发生通信问题或者防火墙使会话断开连接时，会话就可能中断。
2. 检查客户机是否正在重新启动先前的会话，以便使用至少一次或至多一次服务质量来再次发送未传递的消息。

- a) 如果客户机应用程序使用的是 Java SE MQTT 客户机，请检查其是否将 `MqttClient.CleanSession` 设置为 `false`
- b) 如果您正在使用不同的客户机库，请检查是否正在正确地重新启动会话。
3. 检查客户机应用程序是否是重新启动同一会话，而不是错误地启动其他会话。
要再次启动同一个会话，`cleanSession = false`、`Mqttclient.clientIdentifier` 和 `MqttClient.serverURI` 必须与前一个会话相同。
4. 如果会话过早关闭，请检查消息在客户机的持久库中是否可用以便再次发送。
 - a) 如果客户机应用程序使用的是 Java SE MQTT 客户机，请检查消息是否保存在持久文件夹中；请参阅第 126 页的『客户端日志文件和客户端配置文件』
 - b) 如果您正在使用不同的客户机库，或者您已经实现了自己的持久性机制，请检查它是否在正常工作。
5. 请检查在传递消息之前是否没有人删除此消息。
等待传递到 MQTT 客户机的未传递消息存储在 `SYSTEM.MQTT.TRANSMIT.QUEUE` 中。等待传送到遥测服务器的消息由客户机持久机制存储；请参阅 [MQTT 客户机中的消息持久性](#)。
6. 请检查客户机是否具有它期望接收的发布的预订。
使用 IBM MQ Explorer 或使用 `runmqsc` 或 PCF 命令列出预订。已命名全部 MQTT 客户机预订。将为其提供以下格式的名称: `ClientIdentifier:Topic name`
7. 请检查发布程序是否具有权限发布以及订户是否具有权限预订发布主题。

```
dspmqaut -m qMgr -n topicName -t topic -p user ID
```

在集群发布/预订系统中，必须授权订户预订与此订户相连的队列管理器上的主题。不需要授权订户预订已发布此发布的队列管理器上的主题。必须正确地授权队列管理器之间的通道传递代理预订和转发发布。

使用 IBM MQ Explorer 创建同一预订并向其进行发布。使用客户机实用程序来模拟应用程序客户机进行发布和预订。从 IBM MQ Explorer 启动实用程序，并将其用户标识更改为与客户机应用程序所采用的用户标识相匹配。

8. 请检查订户是否具有许可权将发布放在 `SYSTEM.MQTT.TRANSMIT.QUEUE` 上。

```
dspmqaut -m qMgr -n queueName -t queue -p user ID
```

9. 检查 IBM MQ 点到点应用程序是否具有权限将其消息放在 `SYSTEM.MQTT.TRANSMIT.QUEUE` 上。

```
dspmqaut -m qMgr -n queueName -t queue -p user ID
```

请参阅[将消息直接发送到客户机](#)。

Windows Linux AIX 解决问题：Telemetry (MQXR) 服务未启动

解决 Telemetry (MQXR) 服务未能启动的问题。请检查 MQ Telemetry 安装，并确认没有文件缺失、被移动或具有错误许可权。检查 Telemetry (MQXR) 服务在查找 Telemetry (MQXR) 服务程序时所使用的路径。

开始之前

已安装 MQ Telemetry 功能部件。IBM MQ Explorer 在 **IBM MQ > 队列管理器 > qMgrName > --> Telemetry** 中具有 Telemetry 文件夹。如果此文件夹不存在，那么表明安装失败。

必须创建 Telemetry (MQXR) 服务才能将其启动。如果尚未创建遥测 (MQXR) 服务，请运行 [定义样本配置 ... Telemetry](#) 文件夹中的向导。

如果之前已启动遥测 (MQXR) 服务，那么会在 Telemetry 文件夹下创建其他 **Channels** 和 **Channel Status** 文件夹。Telemetry 服务 `SYSTEM.MQXR.SERVICE` 位于 **服务** 文件夹中。如果单击了用于显示系统对象的 IBM MQ Explorer 单选按钮，那么该服务可见。

右键单击 `SYSTEM.MQXR.SERVICE`，以启动和停止该服务、显示其状态以及显示您的用户标识是否有权启动该服务。

关于此任务

`SYSTEM.MQXR.SERVICE Telemetry (MQXR)` 服务未能启动。启动清单本身时，会出现两种不同形式的故障：

1. 启动命令立即失败。
2. 启动命令成功，但该服务紧接着就停止了。

过程

1. 启动该服务。

结果

该服务立即停止。此时会出现一个窗口，其中显示错误消息，例如：

```
IBM MQ cannot process the request because the
executable specified cannot be started. (AMQ4160)
```

原因

安装过程中缺少文件，或者错误地设置了对已安装文件的许可权。

`MQ Telemetry` 功能部件仅安装在其中一个高可用性队列管理器（共有两个）上。如果此队列管理器实例切换到备用实例，那么将尝试启动 `SYSTEM.MQXR.SERVICE`。用于启动此服务的命令将失败，因为备用实例上未安装 `Telemetry (MQXR)` 服务。

调查

查看错误日志；请参阅第 125 页的『服务器端日志』。

操作

安装 `MQ Telemetry` 功能部件，或者卸载并重新安装此功能部件。

2. 启动该服务；等待 30 秒；刷新 `IBM MQ Explorer` 并检查服务状态。

结果

该服务启动，然后又停止。

原因

`SYSTEM.MQXR.SERVICE` 启动了 `runMQXRService` 命令，但该命令失败。

调查

查看错误日志；请参阅第 125 页的『服务器端日志』。

查看唯一定义的样本通道是否出现问题。备份并清除 `WMQ data directory\Qmgrs\qMgrName\mqxr\` 目录的内容。运行“样本配置”向导，然后尝试启动该服务。

操作

查找许可权和路径问题。

Windows Linux AIX 解决问题：遥测服务未调用 JAAS 登录模块

了解 `JAAS` 登录模块是否未由遥测 (`MQXR`) 服务调用，并配置 `JAAS` 以更正该问题。

开始之前

您已修改 `WMQ installation directory\mqxr\samples\samples\LoginModule.java` 以创建自己的认证类 `WMQ installation directory\mqxr\samples\samples\LoginModule.class`。或者，您已经编写了自己的 `JAAS` 认证类并且已将它们放入您选择的目录中。在利用遥测 (`MQXR`) 服务进行某些初始测试后，您会认为认证类未由遥测 (`MQXR`) 服务调用。

注: 防止出现以下可能情况: 认证类可能被应用于 IBM MQ 的维护所覆盖。请使用您自己的认证类路径, 而不是 IBM MQ 目录树中的路径。

关于此任务

此任务使用一种方案来说明如何解决此问题。在此方案中, 一个称为 `security.jaas` 的包中包含一个称为 `JAASLogin.class` 的 JAAS 认证类。它存储在路径 `C:\WMQTelemetryApps\security\jaas` 中。请参阅遥测通道 JAAS 配置和 `AuthCallback MQXR` 类, 以获取为 MQ Telemetry 配置 JAAS 方面的相关帮助。第 134 页的『示例 JAAS 配置』这一示例是一个样本配置。

过程

1. 在 `mqxr.log` 中查找由 `javax.security.auth.login.LoginException` 抛出的异常。
有关 `mqxr.log` 的路径, 请参阅第 125 页的『服务器端日志』, 有关日志中列出的异常示例, 请参阅第 136 页的图 11。
2. 通过将您的 JAAS 配置与第 134 页的『示例 JAAS 配置』中已使用的示例进行比较, 从而更正您的 JAAS 配置。
3. 在将 `login` 类重构到认证包中并使用相同的路径部署它之后, 用 `JAASLoginModule` 样本替换该类。将 `loggedIn` 的值在 `true` 与 `false` 之间进行切换。
如果当 `loggedIn` 为 `true` 时并不存在此问题, 而 `loggedIn` 为 `false` 时又出现此问题, 那么说明登录类存在问题。
4. 请检查此问题是否与授权有关, 而不是与认证有关。
 - a) 更改遥测通道定义, 以使用固定的用户标识执行授权检查。选择一个 `mqm` 组的成员的用户标识。
 - b) 重新运行客户机应用程序。

如果不再存在此问题, 那么就要从为授权而传递的用户标识来考虑解决方案。所传递的用户名是什么? 从您的登录模块打印至文件。使用 IBM MQ Explorer 或 `dspmqaauth` 检查其访问许可权。

示例 JAAS 配置

使用 IBM MQ Explorer 中的 "新建遥测通道" 向导来配置遥测通道。

JAAS 配置文件具有名为 `JAASConfig` 的节, 该节将 Java 类命名为 `security.jaas.JAASLogin`, 供 JAAS 用于认证客户机。

```
JAASConfig {
    security.jaas.JAASLogin required debug=true;
};
```

图 6: *WMQ Installation directory\data\qmgrs\qMgrName\mqxr\jaas.config*

当 `SYSTEM.MQTT.SERVICE` 启动时, 它会将第 134 页的图 7 中的路径添加至其类路径。

```
CLASSPATH=C:\WMQTelemetryApps;
```

图 7: *WMQ Installation directory\data\qmgrs\qMgrName\service.env*

第 135 页的图 8 显示了第 134 页的图 7 中添加到为遥测 (MQXR) 服务设置的类路径的其他路径。

```
CLASSPATH=;C:\IBM\MQ\Program\mqxr\bin\..\lib\MQXRListener.jar;
C:\IBM\MQ\Program\mqxr\bin\..\lib\WMQCommonServices.jar;
C:\IBM\MQ\Program\mqxr\bin\..\lib\objectManager.utils.jar;
C:\IBM\MQ\Program\mqxr\bin\..\lib\com.ibm.micro.xr.jar;
C:\IBM\MQ\Program\mqxr\bin\..\lib\com.ibm.mq.jmqi.jar;
C:\IBM\MQ\Program\mqxr\bin\..\lib\com.ibm.mqjms.jar;
C:\IBM\MQ\Program\mqxr\bin\..\lib\com.ibm.mq.jar;
C:\WMQTelemetryApps;
```

图 8: runMQXRService.bat 的类路径输出

第 135 页的图 9 中的输出显示已启动遥测 (MQXR) 服务。

```
21/05/2010 15:32:12 [main] com.ibm.mq.MQXRService.MQXRPropertiesFile
AMQXR2011I: Property com.ibm.mq.MQXR.channel/JAASMCUser value
com.ibm.mq.MQXR.Port=1884;
com.ibm.mq.MQXR.JAASConfig=JAASConfig;
com.ibm.mq.MQXR.UserName=Admin;
com.ibm.mq.MQXR.StartWithMQXRService=true
```

图 9: WMQ Installation directory\data\qmgrs\qMgrName\errors\

当客户机应用程序连接到 JAAS 通道时，如果 com.ibm.mq.MQXR.JAASConfig=JAASWrongConfig 与 jaas.config 文件中 JAAS 节的名称不匹配，那么连接将失败，并且客户机抛出异常（返回码为 0）；请参阅第 135 页的图 10。由于客户机尝试在未连接时断开连接，因此抛出了第二个异常 Client is not connected (32104)。

```
Connecting to tcp://localhost:1883 with client ID SampleJavaV3_publish
reason 5
msg Not authorized to connect
loc Not authorized to connect
cause null
excep Not authorized to connect (5)
Not authorized to connect (5)
    at
    org.eclipse.paho.client.mqttv3.internal.ExceptionHelper.createMqttException(ExceptionHelper.java
:28)
    at
    org.eclipse.paho.client.mqttv3.internal.ClientState.notifyReceivedAck(ClientState.java:885)
    at org.eclipse.paho.client.mqttv3.internal.CommsReceiver.run(CommsReceiver.java:118)
    at java.lang.Thread.run(Thread.java:809)
```

图 10: 连接到 Eclipse Paho 样本时抛出异常

mqxr.log 包含第 135 页的图 10 中所显示的附加输出。

JAAS 检测到该错误，它抛出 javax.security.auth.login.LoginException，原因为 No LoginModules configured for JAAS。如第 136 页的图 11 中所示，这可能由于配置名称错误所致。它也可能是在装入 JAAS 配置时 JAAS 遇到的其他问题导致的。

如果 JAAS 未报告异常，那么 JAAS 已成功装入 JAASConfig 节中指定的 security.jaas.JAASLogin 类。

```
15/06/15 13:49:28.337
AMQXR2050E: Unable to load JAAS config:MQXRWrongConfig.
The following exception occurred javax.security.auth.login.LoginException:
No LoginModules configured for MQXRWrongConfig
```

图 11: 装入 JAAS 配置时出错

多点广播问题故障诊断

以下提示和技巧并非按重要性排序，并且可能会在发布新版本的文档时添加。如果它们是与您执行的工作有关的主题，那么可能会节省您的时间。

在非多点广播网络上测试多点广播应用程序

使用此信息来了解如何以本地方式而不是在多点广播网络上测试 IBM MQ 多点广播应用程序。

在开发或测试多点广播应用程序时，您可能还不具有支持多点广播的网络。要在本地运行应用程序，必须编辑 `mqclient.ini` 文件，如下示例中所示：

编辑 `MQ_DATA_PATH/mqclient.ini` 的 Multicast 节中的 **Interface** 参数：

```
Multicast:
Interface      = 127.0.0.1
```

其中 `MQ_DATA_PATH` 是 IBM MQ 数据目录 (`/var/mqm/mqclient.ini`) 的位置。

多点广播传输现在仅使用本地回送适配器。

为多点广播流量设置相应的网络

开发或测试多点广播应用程序时，在以本地方式对其进行测试之后，您可能想要在支持多点广播的网络上对其进行测试。如果应用程序仅在本地传输，那么您可能必须编辑 `mqclient.ini` 文件，如本节后面所示。如果机器设置正在使用多个网络适配器或虚拟专用网 (VPN) (例如)，那么必须将 `mqclient.ini` 文件中的 **Interface** 参数设置为要使用的网络适配器的地址。

如果 `mqclient.ini` 文件中存在 Multicast 节，请编辑 **Interface** 参数，如下示例中所示：

将：

```
Multicast:
Interface      = 127.0.0.1
```

到：

```
Multicast:
Interface      = IPAddress
```

其中 `IPAddress` 是流动多点广播流量的接口的 IP 地址。

如果 `mqclient.ini` 文件中没有 Multicast 节，请添加以下示例：

```
Multicast:
Interface      = IPAddress
```

其中 `IPAddress` 是流动多点广播流量的接口的 IP 地址。

多点广播应用程序现在在多点广播网络上运行。

多点广播主题字符串太长

如果您的 IBM MQ 多点广播主题字符串由于原因码 MQRC_TOPIC_STRING_ERROR 而遭到拒绝，那么可能是因为该字符串太长。

WebSphereMQ 多点广播对于主题字符串具有 255 字符限制。此限制意味着必须注意树中节点和叶节点的名称；如果节点和叶节点的名称太长，主题字符串可能会超过 255 个字符，并返回 2425 (0979) (RC2425): MQRC_TOPIC_STRING_ERROR 原因码。建议尽可能保持主题字符串简短，因为较长的主题字符串可能对性能造成不利影响。

多点广播主题拓扑问题

使用这些示例来了解不建议使用某些 IBM MQ 多点广播主题拓扑的原因。

如 [IBM MQ 多点广播主题拓扑](#) 中所提及，IBM MQ 多点广播支持要求每棵子树在总体层次结构中具有其自己的多点广播组和数据流。请勿对子树及其父代使用其他多点广播组地址。

有类网络 IP 寻址方案针对多点广播地址指定了地址空间。IP 地址的完整多点广播范围是 224.0.0.0 到 239.255.255.255，但其中一些地址是保留地址。要获取保留地址的列表，请联系您的系统管理员，或访问 <https://www.iana.org/assignments/multicast-addresses> 以获取更多信息。建议您使用 239.0.0.0 到 239.255.255.255 之间的本地作用域多点广播地址。

建议的多点广播主题拓扑

此示例与 [IBM MQ 多点广播主题拓扑](#) 中的示例相同，并且显示两条可能的多点广播数据流。虽然它的表示简单，但是展示了专门设计了 IBM MQ 多点广播的情况，并在此处显示以与 [第二个示例](#) 进行对比：

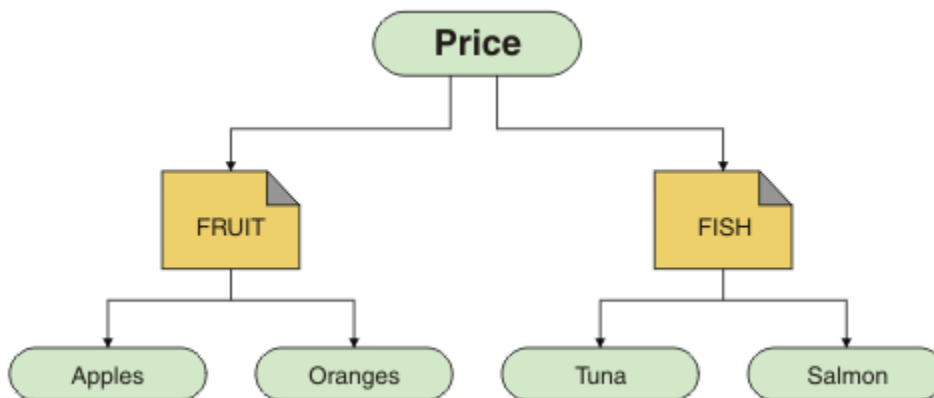
```
DEF COMMINFO(MC1) GRPADDR(
227.20.133.1)

DEF COMMINFO(MC2) GRPADDR(227.20.133.2)
```

其中 227.20.133.1 和 227.20.133.2 是有效的多点广播地址。

这些主题定义用于创建主题树，如下图所示：

```
DEFINE TOPIC(FRUIT) TOPICSTRING('Price/FRUIT') MCAST(ENABLED) COMMINFO(MC1)
DEFINE TOPIC(FISH) TOPICSTRING('Price/FISH') MCAST(ENABLED) COMMINFO(MC2)
```



每个多点广播通信信息 (COMMINFO) 对象都表示一条不同的数据流，因为其组地址不同。在此示例中，主题 FRUIT 定义为使用 COMMINFO 对象 MC1，主题 FISH 定义为使用 COMMINFO 对象 MC2。

IBM MQ 多点广播针对主题字符串具有 255 字符限制。此限制意味着必须注意树中的节点和叶节点的名称；如果节点和叶节点的名称太长，那么主题字符串可能超过 255 个字符并返回 MQRC_TOPIC_STRING_ERROR 原因码。

不建议的多点广播主题拓扑

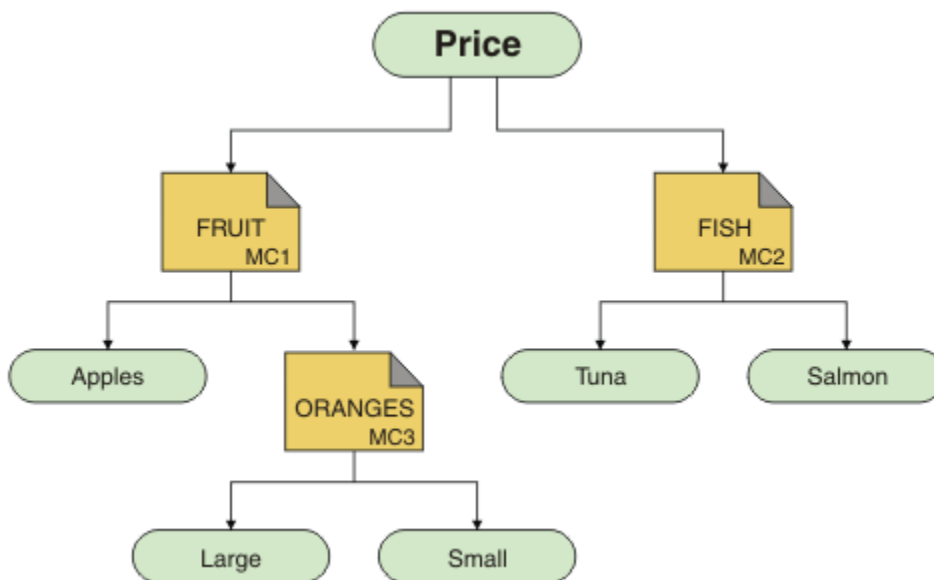
此示例通过添加名为 ORANGES 的另一个主题对象（定义为使用另一个 COMMINFO 主题定义 (MC3)）来扩展前一个示例：

```
DEF COMMINFO(MC1) GRPADDR(227.20.133.1
)
DEF COMMINFO(MC2) GRPADDR(227.20.133.2)
DEF COMMINFO(MC3) GRPADDR(227.20.133.3)
```

其中 227.20.133.1、227.20.133.2 和 227.20.133.3 是有效的多点广播地址。

这些主题定义用于创建主题树，如下图所示：

```
DEFINE TOPIC(FRUIT) TOPICSTRING('Price/FRUIT') MCAST(ENABLED) COMMINFO(MC1)
DEFINE TOPIC(FISH) TOPICSTRING('Price/FISH') MCAST(ENABLED) COMMINFO(MC2)
DEFINE TOPIC(ORANGES) TOPICSTRING('Price/FRUIT/ORANGES') MCAST(ENABLED) COMMINFO(MC3)
```



虽然可以创建此类多点广播拓扑，但是不建议这样做，因为应用程序可能接收不到其期望的数据。

预订 'Price/FRUIT/#' 的应用程序在 COMMINFO MC1 组地址上接收多点广播传输。该应用程序期望在主题树中的该点或该点下方接收有关所有主题的发布。

但是，订户未接收到发布 'Price/FRUIT/ORANGES/Small' 的应用程序所创建的消息，因为消息在 COMMINFO MC3 组地址上进行发送。

对队列管理器问题进行故障诊断

使用此处提供的建议，帮助您解决在使用队列管理器时可能出现的常见问题。

队列管理器不可用错误

- **场景：**您收到队列管理器不可用错误。
- **说明：**配置文件错误通常会阻止找到队列管理器，并导致队列管理器不可用错误。在 Windows 上，当启动队列管理器时，qm.ini 文件中的问题可能会导致队列管理器不可用错误。
- **解决方案：**确保配置文件存在，并且 IBM MQ 配置文件引用正确的队列管理器和日志目录。在 Windows 上，检查 qm.ini 文件中是否存在问题。

IBM MQ 作为资源管理器协调 Db2 时发生错误

- **场景：**您从 IBM MQ Explorer V7,--> 启动队列管理器，协调 Db2 时遇到问题。检查队列管理器错误日志时，发现如以下示例所示的错误：

```
23/09/2008 15:43:54 - Process(5508.1) User(MUSR_MQADMIN) Program(amqzma0.exe)
Host(HOST_1) Installation(Installation1)
VMRF(7.1.0.0) QMgr(A.B.C)
AMQ7604: The XA resource manager 'DB2 MQBankDB database' was not available when called
for xa_open. The queue manager is continuing without this resource manager.
```

- **说明：**运行 IBM MQ 服务进程 amqsvc.exe 的用户标识（缺省名称为 MUSR_MQADMIN）仍在使用不含 DB2USERS 组的组成员资格信息的访问令牌运行。
- **解决方案：**确保 IBM MQ 服务用户标识是 DB2USERS 的成员后，使用以下命令序列：
 1. 停止该服务。
 2. 停止相同用户标识下运行的任何其他进程。
 3. 重新启动这些进程。

重新引导机器将确保完成先前步骤，但这不是必需的。

对队列管理器集群问题进行故障诊断

使用此处提供的核对表以及子主题中提供的建议，帮助您在使用队列管理器集群时检测和处理问题。

开始之前

如果问题与使用集群的发布/预订消息传递相关，而不是与集群相关（这是通常情况），请参阅第 40 页的『针对发布/预订集群的路由：行为注释』。

过程

- 检查所有集群通道是否均已配对。

每个集群发送方通道都连接到名称相同的集群接收方通道。如果无任何本地集群接收方通道的名称与远程队列管理器上的集群发送方通道相同，那么它将无效。

- 检查通道是否正在运行。任何通道都不得永久处于 RETRYING 状态。

使用以下命令来显示运行哪些通道：

```
runmqsc display chstatus(*)
```

如果通道处于 RETRYING 状态，那么可能是通道定义出错，或远程队列管理器可能未运行。当通道处于此状态时，很有可能会在传输队列上构建消息。如果连接到完整存储库的通道处于此状态，那么集群对象（例如，队列和队列管理器）定义将过时且在集群间不再一致。

- 确认无任何通道处于 STOPPED 状态。

手动停止通道时，它将进入 STOPPED 状态。可使用以下命令重新启动停止的通道：

```
runmqsc start channel(xyz)
```

集群的队列管理器将根据需要自动定义到集群中其他队列管理器的集群通道。队列管理器将根据需要自动启动这些自动定义的集群通道，除非先前已手动停止这些通道。如果手动停止了自动定义的集群通道，那么队列管理器会记住其已手动停止，并且不会使其在未来自动启动。如果需要停止通道，请记住在方便的时候再次重新启动，否则请发出以下命令：

```
stop channel(xyz) status(inactive)
```

status(inactive) 选项允许队列管理器在以后重新启动通道（如需要这样做）。

- 确认集群中的所有队列管理器均了解所有的完整存储库。

可使用以下命令完成此操作：

```
runmqsc display clusqmgr(*) qmtype
```

部分存储库可能不了解所有其他部分存储库。所有完整存储库均应了解集群中的所有队列管理器。如果缺少集群队列管理器，那么这可能意味着未正确运行特定的通道。

- 确认集群中的每个队列管理器（完整存储库和部分存储库）均在运行手动定义的集群接收方通道，并在正确的集群中定义。

要查看其他哪些队列管理器正在与集群接收方通道对话，请使用以下命令：

```
runmqsc display chstatus(*) rqmname
```

检查每个手动定义的集群接收方是否都将 **conname** 参数定义为 **ipaddress(port)**。如果没有正确的连接名称，其他队列管理器将不了解往回连接时要使用的连接详细信息。

- 确认每个部分存储库都针对完整存储库运行了手动定义的集群发送方通道，并在正确的集群中定义。

集群发送方通道名称必须与其他队列管理器上的集群接收方通道名称匹配。

- 确认每个完整存储库都针对每个其他完整存储库运行了手动定义的集群发送方通道，并在正确的集群中定义。

集群发送方通道名称必须与其他队列管理器上的集群接收方通道名称匹配。每个完整存储库都未记录集群中的其他完整存储库。它会假定具有手动定义的集群发送方通道的任何队列管理器均为完整存储库。

- 检查死信队列。

队列管理器无法传送的消息将被发送至死信队列。

- 检查是否已为每个部分存储库队列管理器定义单个到完整存储库队列管理器之一的集群发送方通道。

此通道充当部分存储库队列管理器最初连接集群所用的“引导程序”通道。

- 确认预期的完整存储库队列管理器为实际的完整存储库，并在正确的集群中。

可使用以下命令完成此操作：

```
runmqsc display qmgr repos reposnl
```

- 确认未在传输队列或系统队列上构建消息。

您可以使用以下命令检查传输队列：

```
runmqsc display ql(*) curdepth where (usage eq xmitq)
```

您可以使用以下命令检查系统队列：

```
display ql(system*) curdepth
```

相关任务

配置队列管理器集群

[第 7 页的『在 UNIX, Linux, and Windows 上执行初步检查』](#)

开始在 UNIX, Linux, and Windows 上详细确定问题之前，请考虑是否存在引起问题的明显原因，或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

[第 26 页的『在 z/OS 上执行初步检查』](#)

开始在 z/OS 上详细确定问题之前，请考虑是否存在引起问题的明显原因，或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

[第 18 页的『在 IBM i 上执行初步检查』](#)

开始在 IBM i 上详细确定问题之前，请考虑是否存在引起问题的明显原因，或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

相关参考

消息和原因码

V 9.1.3

Multi

应用程序均衡故障诊断

可使用 DISPLAY APSTATUS 命令来列出与应用程序均衡相关联的症状和解决方案。

DIS APSTATUS(X) TYPE(APPL)

症状

未列出预期的应用程序。

解决方案

- 验证是否在代码中或在应用程序启动时正确设置了 APPLTAG 字段。
- 调查 DIS APSTATUS(*) 输出中列出的其他应用程序，以查看是否由于名称格式不正确或设为缺省值而列出了意想不到的应用程序。
- 尝试在统一集群中的每个队列管理器上运行命令 DIS APSTATUS(X) TYPE(LOCAL) where(MOVABLE eq NO)，以查找无法分布在统一集群中的应用程序实例。

症状

未列出预期的应用程序总数。

解决方案

- 验证是否实际启动了预期数量的实例来连接到统一集群。
- 验证统一集群能否正常通信，并验证所有队列管理器是否都使用 DIS APSTATUS(X) TYPE(QMGR) 来报告应用程序计数。

症状

列出了预期的应用程序总数，但某些应用程序被标记为不可移动。

解决方案

在统一集群中的每个队列管理器上，使用 DIS APSTATUS(X) TYPE(LOCAL) where(MOVABLE eq NO) 并调查 IMMREASN 字段。

症状

均衡状态为 UNKNOWN

解决方案

这是一个临时状态，很快就会自行解决。请稍后重试该命令。

症状

均衡状态为 NOTAPPLIC。

解决方案

- 如果此队列管理器不在统一集群中，那么均衡状态始终为 NOTAPPLIC，因为没有可供重新均衡的实例。
- 在统一集群中，这意味着从来没有具有此名称的应用程序连接到集群并可移动。在集群中未分发过此应用程序的相关信息。

使用 DIS APSTATUS(X) TYPE(LOCAL) where(MOVABLE eq NO) 并调查 IMMREASN 字段。

症状

均衡状态为 NO

解决方案

- 监视此输出一段时间。如果应用程序不断连接并断开连接，这可能就是问题的原因，因为无法重新均衡这些实例。
- 使用 DIS APSTATUS(X) TYPE(QMGR) 来调查每个队列管理器上的实例数量，这会指示实例数过剩或不足的队列管理器，然后继续调查这些队列管理器。

DIS APSTATUS(X) TYPE(QMGR)

症状

未列出统一集群中的所有队列管理器。

解决方案

- 验证 BALSTATE 是否不为 NOTAPPLIC，因为这会阻止在统一集群中传播信息。
使用 DIS APSTATUS(X) TYPE(LOCAL) 来查看 IMMREASN 字段。
- 验证所有缺失的队列管理器是否正在运行。
- 验证集群的状态，并验证此队列管理器与缺失的队列管理器之间是否有正在运行的通道。

症状

队列管理器列示为 ACTIVE(NO)

解决方案

- 验证所有缺失的队列管理器是否正在运行
- 验证集群的状态，并验证此队列管理器与不活动的队列管理器之间是否有正在运行的通道。

症状

队列管理器有一些不可移动的应用程序实例。

解决方案

在统一集群中的该队列管理器上，使用 DIS APSTATUS(X) TYPE(LOCAL) where(MOVABLE eq NO) 并调查 IMMREASN 字段。

症状

BALSTATE 并非预期状态。

解决方案

- 监视此情况一段时间，因为 BALSTATE 是队列管理器上次尝试重新均衡应用程序时的状态，这只会定期发生。
- 应用程序是否会不断连接并断开连接？如果是这样，这可能会阻止应用程序重新均衡到稳定状态。
- 如果 BALSTATE 保持未均衡状态，请查看具有 BALSTATE(HIGH) 和 BALSTATE(LOW) 的队列管理器上的错误日志，这些错误日志应指示它们是否正在请求应用程序实例，以及允许移动的应用程序实例数。
- 验证 DIS APSTATUS(X) TYPE(LOCAL) where(IMMCOUNT gt 1)，以查看是否有实例在请求时无法移动。

DIS APSTATUS(X) TYPE(LOCAL)

症状

应用程序实例已标记为 MOVABLE(NO)

解决方案

- IMMREASN 字段是否为 NOTCLIENT。如果是，那么应用程序正在使用服务器绑定，因此无法移动到另一个队列管理器
- IMMREASN 字段是否为 NOTRECONN。如果是，那么应用程序不会作为可重新连接的客户机进行连接，因此无法移动到另一个队列管理器。

使用 DIS CONN(*) TYPE(CONN) WHERE(CONNTAG eq 'xxx') CONNOPTS (其中 xxx 是 DIS APSTATUS 输出中的 CONNTAG)，以了解其连接方式。

- IMMREASN 字段是否为 **V9.1.4** APPNAMECHG。如果是，那么应用程序实例正在同一连接上建立多个对话，但更改了应用程序名称，这会阻止移动特定的应用程序实例。
- IMMREASN 字段是否为 MOVING。如果是，请稍等片刻，当请求移动应用程序实例时，该问题应该会消失。
- 否则，请检查 IMMDATE 和 IMMTIME 字段，以查看应用程序是否只是临时标记为不可移动。

运行 REFRESH CLUSTER 时发现的应用程序问题

发出 **REFRESH CLUSTER** 会破坏集群。它可能会导致集群对象短时间内不可见，直至 **REFRESH CLUSTER** 处理完成为止。这会影响到正在运行的应用程序。以下这些注释描述了您可能看到的部分应用程序问题。

在 MQOPEN、MQPUT 或 MQPUT1 调用中可能会看到的原因码

执行 **REFRESH CLUSTER** 期间，可能会看到以下原因码。在本主题随后的部分中描述了出现以下任一代码的原因。

- 2189 MQRC_CLUSTER_RESOLUTION_ERROR
- 2085 MQRC_UNKNOWN_OBJECT_NAME
- 2041 MQRC_OBJECT_CHANGED
- 2082 MQRC_UNKNOWN_ALIAS_BASE_Q
- 2270 MQRC_NO_DESTINATIONS_AVAILABLE

所有这些原因码都指示在 IBM MQ 代码中的某一级别名称查找失败，如果应用程序在执行 **REFRESH CLUSTER** 操作期间一直在运行，那么预计会出现这些问题。

REFRESH CLUSTER 操作可能会在本地和/或远程执行，并导致这些结果。如果完整存储库非常繁忙，那么出现这些问题的可能性就特别高。如果 **REFRESH CLUSTER** 活动在完整存储库上以本地方式运行，或者在完整存储库负责的一个或多个集群内的其他队列管理器上以远程方式运行，那么会发生此情况。

对于暂时缺失且很快便能恢复的集群队列，所有这些原因码均属于临时可重试状况（但对于 2041 MQRC_OBJECT_CHANGED，可能略微难以确定此状况是否可重试）。如果符合应用程序规则（例如，最大服务次数），那么您可能应重试约一分钟，以便为完成 **REFRESH CLUSTER** 活动提供时间。对于中等规模的集群，完成速度可能要快得多。

如果从 **MQOPEN** 返回以上任一原因码，那么表示未创建任何对象句柄，但稍后重试应可成功创建对象句柄。

如果从 **MQPUT** 返回以上任一原因码，那么不会自动关闭对象句柄，并且重试最终应成功，而无需先关闭对象句柄。但是，如果应用程序使用 bind-on-open 选项打开了句柄，并且因此要求所有消息进入相同通道，那么（与应用程序的期望相反）无法保证重试的 *put* 会进入与之前相同的通道或队列管理器。因此，在此情况下最好关闭该对象句柄并打开新的对象句柄，以重新获取 bind-on-open 语义。

如果从 **MQPUT1** 返回以上任一原因码，那么无法知晓问题是在操作的 *open* 部分还是 *put* 部分中发生的。无论在何处发生问题，均可重试此操作。在此情况下，无需担心 bind-on-open 语义，因为 **MQPUT1** 操作是在一项连续操作中执行的 *open-put-close* 序列。

多中继场景

如果消息流包含了多中继场景（例如，以下示例中显示的多中继场景），那么在应用程序的远程队列管理器上可能会发生由 **REFRESH CLUSTER** 导致的名称查找失败。在此情况下，应用程序会收到成功 (0) 返回

码，但如果发生名称查找失败，它会阻止 **CLUSRCVR** 通道程序将消息路由至任何适当的目标队列。根据消息的持久性，**CLUSRCVR** 通道程序会遵循标准规则将消息写入死信队列。与该操作关联的原因码为：

- 2001 MQRC_ALIAS_BASE_Q_TYPE_ERROR

如果存在持久消息，但没有任何死信队列被定义为接收这些消息，那么您会看到通道将终止。

以下是多中继场景示例：

- 队列管理器 **QM1** 上的 **MQOPEN** 指定 **Q2**。
- 在集群中的远程队列管理器 **QM2** 上，将 **Q2** 定义为别名。
- 消息到达 **QM2**，发现 **Q2** 是 **Q3** 的别名。
- 在集群中的远程队列管理器 **QM3** 上，将 **Q3** 定义为 **qlocal1**。
- 消息到达 **QM3**，并被放入 **Q3**。

测试多中继场景时，您可能会看到以下队列管理器错误日志条目：

- 在发送和接收端，当死信队列就位并且存在持久消息时：
AMQ9544：消息未被放入目标队列
处理“CHLNAME”通道期间，一个或多个消息无法放入目标队列，并且尝试将其放入死信队列。队列的位置为 \$，其中 1 表示本地死信队列，2 表示远程死信队列。
- 在接收端，当死信队列未就位，但存在持久消息时：
AMQ9565：未定义任何死信队列
AMQ9599：程序无法打开队列管理器对象
AMQ9999：通道程序已异常终止
- 在发送端，当死信队列未就位，但存在持久消息时：
AMQ9506：消息接收确认失败
AMQ9780：到远程机器“a.b.c.d(1415)”的通道因出现错误而正在终止
AMQ9999：通道程序已异常终止

有关运行 REFRESH CLUSTER 时为何会显示各原因码的更多详细信息

2189 (088D) (RC2189): MQRC_CLUSTER_RESOLUTION_ERROR

本地队列管理器询问其完整存储库是否存在队列名称。在硬编码的 10 秒超时范围内，没有收到完整存储库的响应。这是因为请求消息或响应消息在队列上进行处理，在适当时机将会清除此状况。在应用程序中，此状况可重试，在这些内部机制完成时将会成功。

2085 (0825) (RC2085): MQRC_UNKNOWN_OBJECT_NAME

本地队列管理器询问（或者之前已询问）其完整存储库是否存在队列名称。完整存储库已响应，表示不知道队列名称。在完整存储库和部分存储库上发生 **REFRESH CLUSTER** 的上下文中，队列所有者可能尚未告知完整存储库有关队列的信息。或者它可能已告知相关信息，但是携带此信息的内部消息在队列上进行处理，在此情况下，在适当时机将会清除此状况。在应用程序中，此状况可重试，在这些内部机制完成时将会成功。

2041 (07F9) (RC2041): MQRC_OBJECT_CHANGED

最常见于 bind-on-open **MQPUT**。本地队列管理器知道存在队列名称，并了解它所在的远程队列管理器。在完整存储库和部分存储库上发生 **REFRESH CLUSTER** 的上下文中，队列管理器记录已被删除，目前正在从完整存储库查询该记录。在应用程序中，要确定此状况是否可重试稍有些复杂。实际上，如果重试 **MQPUT**，在这些内部机制已完成了解远程队列管理器相关信息的作业时将会成功。但是，无法保证将使用相同的队列管理器。更安全的做法是采用接收到 **MQRC_OBJECT_CHANGED** 时通常建议的方法，即关闭对象句柄并重新打开新的对象句柄。

2082 (0822) (RC2082): MQRC_UNKNOWN_ALIAS_BASE_Q

在起因上与 2085 **MQRC_UNKNOWN_OBJECT_NAME** 状况相似，以下情况下会出现此原因码：使用本地别名，并且其 **TARGET** 是因先前针对原因码 2085 所述原因而导致不可访问的集群队列。

2001 (07D1) (RC2001): MQRC_ALIAS_BASE_Q_TYPE_ERROR

此原因码在应用程序上不常见。它只可能出现在队列管理器错误日志中，与尝试向死信队列发送消息有关。**CLUSRCVR** 通道程序接收到来自其伙伴 **CLUSDR** 的消息，并且决定其放置位置。此场景只是先前

针对原因码 2082 和 2085 所述状况的一种变体。在此情况下，当在 MQ 产品中该别名的处理位置不同于在应用程序 **MQPUT** 或 **MQOPEN** 期间该别名的处理位置时，会出现此原因码。

2270 (08DE) (RC2270): MQRC_NO_DESTINATIONS_AVAILABLE

在以下情况下会显示此原因码：应用程序正在使用通过 **MQOO_BIND_NOT_FIXED** 打开的队列，并且目标对象短时间内不可用，直至 **REFRESH CLUSTER** 处理完成为止。

其他备注

如果此环境中存在任何集群发布/预订活动，那么 **REFRESH CLUSTER** 可能会产生其他不需要的影响。例如，暂时丢失订户的预订，然后发现订户未收到消息。请参阅[针对发布/预订集群的 REFRESH CLUSTER 注意事项](#)。

相关概念

[针对发布/预订集群的 REFRESH CLUSTER 注意事项](#)

[集群：使用 REFRESH CLUSTER 最佳实践](#)

相关参考

[MQSC 命令参考：REFRESH CLUSTER](#)

集群发送方通道不断尝试启动

检查队列管理器和侦听器是否正在运行，以及集群发送方和集群接收方通道定义是否正确。

症状

```
1 : display chs(*)
AMQ8417: Display Channel Status details.
CHANNEL(DEMO.QM2) XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
CONNNAME(computer.ibm.com(1414))
CURRENT CHLTYPE(CLUSSDR)
STATUS(RETRYING)
```

原因

1. 远程队列管理器不可用。
2. 为本地手动集群发送方通道或远程集群接收方通道定义了不正确的参数。

解决方案

检查远程队列管理器的可用性是否有问题。

1. 有错误消息吗？
2. 队列管理器是活动的吗？
3. 侦听器在运行吗？
4. 集群发送方通道能启动吗？

如果远程队列管理器可用，那么通道定义是否有问题？检查集群队列管理器的定义类型，以查看通道是否在不断尝试启动，例如：

```
1 : dis clusqmgr(*) deftype where(channel eq DEMO.QM2)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM2) CHANNEL(DEMO.QM2) CLUSTER(DEMO)
DEFTYPE(CLUSSDRA)
```

如果定义类型为 **CLUSSDR**，那么通道正在使用本地人工集群发送方定义。更改本地人工集群发送方定义中任何不正确的参数，并重新启动通道。

如果定义类型为 CLUSSDRA 或 CLUSSDRB，那么通道正在使用自动定义的集群发送方通道。自动定义的集群发送方通道是基于远程集群接收方通道的定义。更改远程集群接收方定义中任何不正确的参数。例如，conname 参数可能不正确：

```
1 : alter chl(demo.qm2) chltype(clusrcvr) conname('newhost(1414)')
AMQ8016: IBM MQ channel changed.
```

对远程集群接收方定义所做的更改会传播到相关的任何集群队列管理器。将相应地更新对应的自动定义的通道。可通过检查更改的参数，可检查是否已正确传播了更新。例如：

```
1 : dis clusqmgr(qm2) conname
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM2) CHANNEL(DEMO.QM2) CLUSTER(DEMO) CONNAME(newhost(1414))
```

现在，如果自动定义的定义正确，请重新启动通道。

DISPLAY CLUSQMGR 显示启动 SYSTEM.TEMP 的 CLUSQMGR 名称。

队列管理器未收到来自手动定义的 CLUSSDR 通道指向的完整存储库队列管理器的任何信息。请检查集群通道定义是否正确。

症状

Multi

```
1 : display clusqmgr(*)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM1) CLUSTER(DEMO)
CHANNEL(DEMO.QM1)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(SYSTEM.TEMPUUID.computer.<yourdomain>(1414))
CLUSTER(DEMO) CHANNEL(DEMO.QM2)
```

z/OS

```
CSQM201I +CSQ2 CSQMDRTC DISPLAY CLUSQMGR DETAILS
CLUSQMGR(SYSTEM.TEMPQMGR.<HOSTNAME>(1716))
CLUSTER(DEMO)
CHANNEL(TO.CSQ1.DEMO)
END CLUSQMGR DETAILS
```

原因

队列管理器未收到来自手动定义的 CLUSSDR 通道指向的完整存储库队列管理器的任何信息。手动定义的 CLUSSDR 通道必须处于运行状态。

解决方案

检查 CLUSRCVR 定义是否同样正确，特别是其 CONNAME 和 CLUSTER 参数。如果定义错误，请更改通道定义。

Multi

您还需要通过发出以下命令来提供 SYSTEM.CLUSTER.TRANSMIT.QUEUE 的正确权限：

```
setmqaut -m <QMGR Name> -n SYSTEM.CLUSTER.TRANSMIT.QUEUE -t q -g mqm +all
```

远程队列管理器尝试进行新一次重新启动，并使用校正的定义启动其通道可能需要一段时间。

返回码 = 2035 MQRC_NOT_AUTHORIZED

出于各种原因显示 RC2035 原因码，包括打开队列或通道时出错、尝试使用具有管理员权限的用户标识时收到错误、使用 IBM MQ JMS 应用程序和在集群上打开队列时出错。MQS_REPORT_NOAUTH 和 MQSAUTHERRORS 可用于进一步诊断 RC2035。

特定的问题

请参阅[生成 RC2035 的特定问题](#)，以获取以下相关信息：

- JMSWMQ2013 安全认证无效
- 队列或通道上出现 MQRC_NOT_AUTHORIZED
- 以管理员身份操作时的 MQRC_NOT_AUTHORIZED（客户机上的 AMQ4036）
- MQS_REPORT_NOAUTH 和 MQSAUTHERRORS 环境变量

在集群中打开队列

此错误的解决方案取决于队列是否在 z/OS 上。在 z/OS 上，使用安全管理器。在其他平台上，针对集群队列创建本地别名，或授权所有用户访问传输队列。

症状

应用程序尝试在集群中打开队列时，会收到返回码 2035 MQRC_NOT_AUTHORIZED。

原因

应用程序尝试在集群中打开队列时，会收到返回码 MQRC_NOT_AUTHORIZED。该队列的权限正确。很可能未授权将应用程序放入集群传输队列。

解决方案

解决方案取决于队列是否在 z/OS 上。请参阅相关信息主题。

当尝试在集群中打开队列时，返回码为 2085 MQRC_UNKNOWN_OBJECT_NAME

症状

应用程序尝试在集群中打开队列时，会收到返回码 2085 MQRC_UNKNOWN_OBJECT_NAME。

原因

对象所在的队列管理器或此队列管理器可能未成功进入集群。

解决方案

确保其中每一个队列管理器均可显示集群中所有的完整存储库。还要确保连接到完整存储库的 CLUSSDR 通道正在尝试启动。

如果队列在集群中，请检查是否使用了相应的打开选项。您无法从远程集群队列获取消息，因此请确保打开选项仅用于输出。

```
1 : display clusqmgr(*) qmtype status
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM1)          CLUSTER(DEMO)
CHANNEL(DEMO.QM1)      QMTYPE(NORMAL)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM2)          CLUSTER(DEMO)
CHANNEL(DEMO.QM2)      QMTYPE(REPOS)
STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM3)          CLUSTER(DEMO)
```

```
CHANNEL (DEMO.QM3)      QMTYPE (REPOS)
STATUS (RUNNING)
```

注: 将 IBM MQ 与 WebSphere Application Server 配合使用时, 如果您具有连接到属于 IBM MQ 集群的 IBM MQ 队列管理器的 JMS 应用程序, 并且 JMS 应用程序尝试访问集群中其他位置的集群队列, 那么您可能会看到此问题。如果应用程序想要打开位于集群中的集群队列, 或者在托管集群队列的集群中指定队列管理器的名称, 那么需要将队列管理器留空。

相关参考

[2085 \(0825\) \(RC2085\): MQRC_UNKNOWN_OBJECT_NAME](#)

尝试打开集群中的队列时返回码 = 2189 MQRC_CLUSTER_RESOLUTION_ERROR

确保连接到完整存储库的 CLUSSDR 通道不会不断尝试启动。

症状

应用程序尝试在集群中打开队列时, 会收到返回码 2189 MQRC_CLUSTER_RESOLUTION_ERROR。

原因

队列第一次打开, 队列管理器无法联系任何完整存储库。

解决方案

确保连接到完整存储库的 CLUSSDR 通道不会不断尝试启动。

```
1 : display clusqmgr(*) qmtype status
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR (QM1)          CLUSTER (DEMO)
CHANNEL (DEMO.QM1)      QMTYPE (NORMAL)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR (QM2)          CLUSTER (DEMO)
CHANNEL (DEMO.QM2)      QMTYPE (REPOS)
STATUS (RUNNING)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR (QM3)          CLUSTER (DEMO)
CHANNEL (DEMO.QM3)      QMTYPE (REPOS)
STATUS (RUNNING)
```

相关参考

[2189 \(088D\) \(RC2189\): MQRC_CLUSTER_RESOLUTION_ERROR](#)

在集群中打开队列时收到返回码 2082 MQRC_UNKNOWN_ALIAS_BASE_Q

应用程序尝试在集群中打开队列时会得到 rc=2082 MQRC_UNKNOWN_ALIAS_BASE_Q。

问题

已发出 MQOPEN 或 MQPUT1 调用, 指定别名队列作为目标, 但是别名队列属性中的 *BaseQName* 未识别为队列名称。

当 *BaseQName* 为无法成功解析的集群队列的名称时, 也可能出现此原因码。

MQRC_UNKNOWN_ALIAS_BASE_Q 可能表明应用程序指定的是其连接到的队列管理器或托管别名队列的队列管理器的 **ObjectQmgrName**。这意味着, 队列管理器将在指定的队列管理器上查找别名目标队列, 并且由于别名目标队列未在本地队列管理器上而失败。

解决方案

将 **ObjectQmgrName** 参数保留为空, 以便集群能够确定路由至哪一个队列管理器。

如果队列在集群中, 请检查是否使用了相应的打开选项。您无法从远程集群队列获取消息, 因此请确保打开选项仅用于输出。

相关参考

2082 (0822) (RC2082): MQRC_UNKOWNN_ALIAS_BASE_Q

消息未到达目标队列

确保相应的集群传输队列为空，并且连接目标队列管理器的通道正在运行。

症状

消息未到达目标队列。

原因

消息可能被困在其源队列管理器上。

解决方案

1. 识别正在将消息发送到目标的传输队列以及通道状态。

```
1 : dis clusqmgr(QM1) CHANNEL(*) STATUS DEFTYPE QMTYPE XMITQ
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM1) CLUSTER(DEMO)
CHANNEL(DEMO.QM1) DEFTYPE(CLUSSDRA)
QMTYPE(NORMAL) STATUS(RUNNING)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.DEMO.QM1)
```

2. 确保集群传输队列为空。

```
1 : display ql(SYSTEM.CLUSTER.TRANSMIT.DEMO.QM1) curdepth
AMQ8409: Display Queue details.
QUEUE(SYSTEM.CLUSTER.TRANSMIT.DEMO.QM1) CURDEPTH(0)
```

放入集群别名队列的消息进入 SYSTEM.DEAD.LETTER.QUEUE

集群别名队列解析到不存在的本地队列。

症状

放入别名队列的消息进入 SYSTEM.DEAD.LETTER.QUEUE，显示原因 MQRC_UNKOWNN_ALIAS_BASE_Q。

原因

消息将路由至已定义集群别名队列的队列管理器。该队列管理器上未定义本地目标队列。由于已使用 MQ00_BIND_ON_OPEN 打开选项放入消息，队列管理器无法将消息重新排队。

使用 MQ00_BIND_ON_OPEN 时，将紧密绑定集群队列别名。解析的名称是目标队列及定义集群队列别名的任何队列管理器的名称。队列管理器名称放在传输队列头中。如果消息将发送至的目标队列在队列管理器上不存在，那么消息将放在死信队列上。将不会重新计算目标，因为传输头包含由 MQ00_BIND_ON_OPEN 解析的目标队列管理器的名称。如果别名队列使用 MQ00_BIND_NOT_FIXED 打开，那么传输队列头将包含空的队列管理器名称，并将重新计算目标。在此情况下，如果在集群中其他位置定义了本地队列，那么将在此处发送消息。

解决方案

1. 更改所有别名队列定义以指定 DEFBIND (NOTFIXED)。
2. 使用 MQ00_BIND_NOT_FIXED 作为打开队列时的打开选项。
3. 如果指定 MQ00_BIND_ON_OPEN，请确保解析到同一队列管理器上定义的本地队列的集群别名为别名。

队列管理器在集群中具有有关队列和通道的过时信息

症状

DISPLAY QCLUSTER 和 DISPLAY CLUSQMGR 显示已过时的对象。

原因

对集群的更新仅通过手动定义的 CLUSSDR 通道在完整存储库间流动。集群形成后，CLUSSDR 通道显示为 DEFTYPE (CLUSSDRB) 通道，因为它们既是手动通道，也是自动通道。必须有足够的 CLUSSDR 通道以在所有完整存储库间形成完整的网络。

解决方案

- 确认对象所在的队列管理器和本地队列管理器仍连接到集群上。
- 确认每个队列管理器均可显示集群中所有的完整存储库。
- 检查连接到完整存储库的 CLUSSDR 通道是否会不断尝试重新启动。
- 检查完整存储库是否定义了足够的 CLUSSDR 通道来将其正确连接在一起。

```
1 : dis clusqmgr(QM1) CHANNEL(*) STATUS DEFTYPE QMTYPE
XMITQ
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM1) CLUSTER(DEMO)
CHANNEL(DEMO.QM1) DEFTYPE(CLUSSDRA)
QMTYPE(NORMAL) STATUS(RUNNING)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.DEMO.QM1)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM2) CLUSTER(DEMO)
CHANNEL(DEMO.QM2) DEFTYPE(CLUSRCVR)
QMTYPE(REPOS)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.DEMO.QM2)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM3) CLUSTER(DEMO)
CHANNEL(DEMO.QM3) DEFTYPE(CLUSSDRB)
QMTYPE(REPOS) STATUS(RUNNING)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.DEMO.QM3)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM4) CLUSTER(DEMO)
CHANNEL(DEMO.QM4) DEFTYPE(CLUSSDRA)
QMTYPE(NORMAL) STATUS(RUNNING)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.DEMO.QM4)
```

本地队列管理器中未反映集群中的更改

存储库管理器进程未在处理存储库命令，可能是因为在命令队列中接收或处理消息时出现问题。

症状

本地队列管理器中未反映集群中的更改。


原因

存储库管理器进程未在处理存储库命令。

解决方案

1. 检查 SYSTEM.CLUSTER.COMMAND.QUEUE 是否为空。

```
1 : display ql(SYSTEM.CLUSTER.COMMAND.QUEUE) curdepth
AMQ8409: Display Queue details.
QUEUE(SYSTEM.CLUSTER.COMMAND.QUEUE) CURDEPTH(0)
```

2.  检查通道启动程序是否正在 z/OS 上运行。
3. 确认错误日志中无错误消息指示队列管理器临时出现资源短缺。

DISPLAY CLUSQMGR 两次显示队列管理器

使用 RESET CLUSTER 命令来除去队列管理器的旧实例的所有跟踪。

```
1 : display clusqmgr(QM1) qmid
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM1)          CLUSTER(DEMO)
CHANNEL(DEMO.QM1)     QMID(QM1_2002-03-04_11.07.01)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(QM1)          CLUSTER(DEMO)
CHANNEL(DEMO.QM1)     QMID(QM1_2002-03-04_11.04.19)
```

集群可正常运作并忽略早期版本的队列管理器。在大约 90 天之后，较早版本的队列管理器的集群信息将到期，并且会自动删除。但是，您可能更倾向于手动删除此信息。

原因

1. 队列管理器可能已被删除，然后重新创建并重新定义。
2. 它可能已在 z/OS 上冷启动，而无需首先遵循此过程来从集群中除去队列管理器。

解决方案

要立即除去队列管理器的所有跟踪，请使用完整存储库队列管理器中的 RESET CLUSTER 命令。该命令会从集群中除去较旧且不需要的队列管理器及其队列。

```
2 : reset cluster(DEMO) qmid('QM1_2002-03-04_11.04.19') action(FORCEREMOVE) queues(yes)
AMQ8559: RESET CLUSTER accepted.
```

使用 RESET CLUSTER 命令可针对受影响的队列管理器停止自动定义的集群发送方通道。必须在完成 RESET CLUSTER 命令后，手动重新启动已停止的所有集群发送方通道。

队列管理器未再加入集群

发出 RESET 或 REFRESH 集群命令后，可停止从队列管理器到集群的通道。检查集群通道状态并重新启动通道。

症状

发出 RESET CLUSTER 和 REFRESH CLUSTER 命令后，队列管理器未再加入集群。

原因

RESET 和 REFRESH 命令的副作用可能是通道会停止。通道会停止，以便在完成 RESET 或 REFRESH 命令时运行正确版本的通道。

解决方案

检查问题队列管理器与完整存储库之间的通道是否在运行，并在必要时使用 START CHANNEL 命令。

相关信息

集群：[使用 REFRESH CLUSTER 最佳实践](#)

集群发送方通道上设置的工作负载均衡不工作

在集群发送方通道上指定的任何工作负载均衡都可能被忽略。相反，请在目标队列管理器的集群接收方通道上指定集群工作负载通道属性。

症状

您已在集群发送方通道上指定一个或多个集群工作负载通道属性。所生成的工作负载均衡不符合您的预期。

原因

在集群发送方通道上指定的任何工作负载均衡都可能被忽略。有关此问题的说明，请参阅[集群通道](#)。请注意，您仍可根据集群缺省值，或根据目标队列管理器的匹配的集群接收方通道上设置的属性，来获取某种形式的工作负载均衡。

解决方案

在目标队列管理器的集群接收方通道上指定集群工作负载通道属性。

相关参考

[CLWLPRTY 通道属性](#)

[CLWLRANK 通道属性](#)

[CLWLWGHT 通道属性](#)

[NETPRTY 通道属性](#)

复原的集群中的过时信息

复原队列管理器后，其集群信息已过时。使用 **REFRESH CLUSTER** 命令刷新集群信息。

问题

在对 QM1 进行映像备份后，已复原 DEMO 集群中的部分存储库，并且其包含的集群信息已过时。

解决方案

在 QM1 上，发出 **REFRESH CLUSTER(DEMO)** 命令。

注: 对于大型集群，当集群正在运行中时，使用 **REFRESH CLUSTER** 命令可能会破坏该集群，并且将在 27 天的时间间隔之后，集群对象才会再次自动向所有相关队列管理器发送状态更新。请参阅[在大型集群中刷新可能会影响集群的性能和可用性](#)。

在 QM1 上运行 **REFRESH CLUSTER(DEMO)** 时，将除去 QM1 具有的有关集群 DEMO 的所有信息，QM1 自身、其所拥有的队列以及如何访问集群中完整存储库的信息除外。然后，QM1 将联系完整存储库，并且告知这些存储库关于自身及其队列的信息。QM1 是部分存储库，因此完整存储库不会立即告知 QM1 有关集群中所有其他部分存储库的信息。相反，QM1 会通过每个其他队列和队列管理器在集群中是下一个活动项时接收的信息来慢慢搭建其他部分存储库的信息。

从完整存储库中错误地强行除去集群队列管理器

通过在从存储库中除去的队列管理器上发出 **REFRESH CLUSTER** 命令，将队列管理器复原至完整存储库。

问题

DEMO 集群中的完整存储库上错误地发出 **RESET CLUSTER(DEMO) QMNAME(QM1) ACTION(FORCEREMOVE)** 命令。

解决方案

在 QM1 上，发出 **REFRESH CLUSTER(DEMO)** 命令。

注: 对于大型集群，当集群正在运行中时，使用 **REFRESH CLUSTER** 命令可能会破坏该集群，并且将在 27 天的时间间隔之后，集群对象才会再次自动向所有相关队列管理器发送状态更新。请参阅[在大型集群中刷新可能会影响集群的性能和可用性](#)。

已删除可能的存储库消息

要传至某个队列管理器的消息将从其他队列管理器中的 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 除去。在受影响的队列管理器上发出 `REFRESH CLUSTER` 命令，来复原信息。

问题

要传至 QM1 的消息将从其他队列管理器中的 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 除去，并且可能是存储库消息。

解决方案

在 QM1 上，发出 `REFRESH CLUSTER (DEMO)` 命令。

注: 对于大型集群，当集群正在运行中时，使用 **REFRESH CLUSTER** 命令可能会破坏该集群，并且将在 27 天的时间间隔之后，集群对象才会再次自动向所有相关队列管理器发送状态更新。请参阅[在大型集群中刷新可能会影响集群的性能和可用性](#)。

QM1 会除去其具有的有关 DEMO 集群的所有信息，与作为集群中的完整存储库的集群队列管理器相关的信息除外。假定此信息仍然正确，QM1 会联系完整存储库。QM1 会通知完整存储库有关其自身及其队列的情况。当存在于集群中其他地点的队列和队列管理器打开时，它会恢复其信息。

同时移动的两个完整存储库

如果同时将两个完整存储库移至新的网络地址，那么将不会自动使用新地址更新集群。请遵循以下过程来传输新的网络地址。一次移动一个存储库，以避免发生问题。

问题

DEMO 集群包含两个完整存储库：QM1 和 QM2。会同时将它们移至网络上的新位置。

解决方案

1. 更改 `CLUSRCVR` 和 `CLUSSDR` 通道中的 `CONNAME`，以指定新的网络地址。
2. 更改其中一个队列管理器（QM1 或 QM2），使其不再是任何集群的完整存储库。
3. 在更改的队列管理器上，发出 `REFRESH CLUSTER (*) REPOS (YES)` 命令。

注: 对于大型集群，当集群正在运行中时，使用 **REFRESH CLUSTER** 命令可能会破坏该集群，并且将在 27 天的时间间隔之后，集群对象才会再次自动向所有相关队列管理器发送状态更新。请参阅[在大型集群中刷新可能会影响集群的性能和可用性](#)。

4. 更改队列管理器，使其充当完整存储库。

建议

可按如下所述执行操作以避免发生问题：

1. 将其中一个队列管理器（例如，QM2）移至其新的网络地址。
2. 更改 QM2 `CLUSRCVR` 通道中的网络地址。
3. 启动 QM2 `CLUSRCVR` 通道。
4. 等待另一完整存储库队列管理器 QM1 获知 QM2 的新地址。
5. 将另一完整存储库队列管理器 QM1 移至其新的网络地址。
6. 更改 QM1 `CLUSRCVR` 通道中的网络地址。
7. 启动 QM1 `CLUSRCVR` 通道。
8. 为了透明起见更改手动定义的 `CLUSSDR` 通道，尽管在此阶段，无需它即可进行正确的集群操作。

此过程会强行使 QM2 重新使用正确的 CLUSSDR 通道的信息，来重新建立与 QM1 的联系，然后重新构建其集群知识。此外，再一次联系 QM1 后，会根据 QM2 CLUSRCVR 定义中的 CONNAME 会其赋予其自己的正确网络地址。

集群的未知状态

通过从集群中所有部分存储库重新构建完整存储库，将所有完整存储库中的集群信息复原为已知状态。

问题

正常情况下，完整存储库会交换集群中有关队列和队列管理器的信息。如果刷新了一个完整存储库，那么将从另一个存储库中恢复集群信息。

问题是如何在集群中完全重置所有系统，使集群复原为已知状态。

解决方案

要停止从完整存储库的未知状态更新集群信息，连接到完整存储库的所有 CLUSRCVR 通道都将停止。CLUSSDR 通道会更改为不活动状态。

刷新完整存储库系统时，所有系统都无法通信，因此它们会从同一清除后状态开始。

刷新部分存储库系统时，它们会再加入集群，并会重新构建到完整的一组队列管理器和队列中。重新构建的完整存储库中的集群信息会复原为已知状态。

注: 对于大型集群，当集群正在运行中时，使用 **REFRESH CLUSTER** 命令可能会破坏该集群，并且将在 27 天的时间间隔之后，集群对象才会再次自动向所有相关队列管理器发送状态更新。请参阅[在大型集群中刷新可能会影响集群的性能和可用性](#)。

1. 在所有完整存储库队列管理器上，执行以下步骤：
 - a. 更改作为完整存储库的队列管理器，使其不再是完整存储库。
 - b. 解析任何不确定的 CLUSSDR 通道。
 - c. 等待 CLUSSDR 通道变为不活动状态。
 - d. 停止 CLUSRCVR 通道。
 - e. 所有完整存储库系统上的所有 CLUSRCVR 通道均停止时，请发出 `REFRESH CLUSTER(DEMO) REPOS(YES)` 命令。
 - f. 更改队列管理器，使其成为完整存储库。
 - g. 启动 CLUSRCVR 通道，以重新启用它进行通信。
2. 在所有部分存储库队列管理器上，执行以下步骤：
 - a. 解析任何不确定的 CLUSSDR 通道。
 - b. 确保队列管理器上的所有 CLUSSDR 通道均已停止或处于不活动状态。
 - c. 发出 `REFRESH CLUSTER(DEMO) REPOS(YES)` 命令。

集群队列管理器发生故障时将发生什么情况

如果某个集群队列管理器发生故障，那么其一些未送达的消息将发送至集群中的其他队列管理器。未完成的将等到重新启动队列管理器为止。使用高可用性机制来自动重新启动队列管理器。

问题

如果将消息批次发送到特定的队列管理器，然后该队列管理器变成不可用，那么发送的队列管理器会发生什么情况？

说明

除 NPMSPEED(FAST) 通道上的非持久消息外，未送达的消息批次将回退到发送的队列管理器上的集群传输队列。在 NPMSPEED(FAST) 通道上，将不对非持久消息进行批处理，并且可能会丢失一条消息。

- 不确定的消息以及绑定到不可用的队列管理器的消息将等到队列管理器变得重新可用为止。
- 其他消息会传递到由工作负载管理例程选择的备用队列管理器。

解决方案

可通过将不可用的集群队列管理器配置为多实例队列管理器，或通过特定于平台的高可用性机制来自动地重新启动此队列管理器。

存储库发生故障时将发生的情况

您如何得知存储库发生故障？如何修复？

问题

1. 会将集群信息发送到名为 `SYSTEM.CLUSTER.COMMAND.QUEUE` 的本地队列上的存储库（无论是完整存储库还是部分存储库）中。如果填满此队列，那么可能由于队列管理器停止工作，集群信息消息将路由至死信队列。
2. 存储库存储空间已用尽。

解决方案

1. 监视队列管理器日志 **z/OS** 或 z/OS 系统控制台上的消息，以检测 `SYSTEM.CLUSTER.COMMAND.QUEUE` 是否已满。如果填满，那么需运行应用程序，以从死信队列检索消息并将其重新路由至正确的目标。
2. 如果存储库队列管理器出错，那么消息会告知您发生了什么错误，以及在尝试重新启动前队列管理器等待的时长。
 - **z/OS** 在 IBM MQ for z/OS 上，将对 MQGET 禁用 `SYSTEM.CLUSTER.COMMAND.QUEUE`。
 - 识别并解决错误后，启用 `SYSTEM.CLUSTER.COMMAND.QUEUE`，以便队列管理器能够成功地重新启动。
3. 在存储库存储空间已用尽这一不太可能发生的情况下，存储器分配错误将被发送到队列管理器日志 **z/OS** 或 z/OS 系统控制台。要修复存储问题，请停止运行，然后重新启动队列管理器。重新启动队列管理器时，系统会自动分配更多存储空间，以保存所有存储库信息。

如果对 MQPUT 禁用集群队列，将发生什么情况

用于工作负载均衡的集群队列的所有实例都可能会针对 MQPUT 而禁用。将消息放入队列的应用程序会收到 `MQRC_CLUSTER_PUT_INHIBITED` 或 `MQRC_PUT_INHIBITED` 返回码。您可能要修改此行为。

问题

针对 MQPUT 禁用集群队列时，其状态会反映在此队列中相关的每个队列管理器的存储库中。工作负载管理算法会尝试将消息发送至为 MQPUT 启用的目标。如果没有为 MQPUT 启用任何目标，并且没有队列的本地实例，那么指定了 `MQOO_BIND_ON_OPEN` 的 MQOPEN 调用会将返回码 `MQRC_CLUSTER_PUT_INHIBITED` 返回到应用程序。如果已指定 `MQOO_BIND_NOT_FIXED`，或存在队列的本地实例，那么将成功进行 MQOPEN 调用，但后续 MQPUT 调用将失败，显示返回码 `MQRC_PUT_INHIBITED`。

解决方案

您可以编写用户出口程序以修改工作负载管理例程，以便消息能够路由至针对 MQPUT 禁用的目标。

消息可以到达针对 MQPUT 禁用的目标。禁用队列时，消息可能未完成，或工作负载出口可能已明确选择了目标。目标队列管理器的工作负载管理例程可采用多种方式处理消息：

- 选择相应的另一目标（如有）。
- 将消息放在死信队列上。

- 将消息返回给发起方（如无死信队列）

切换传输队列时的潜在问题

切换传输队列时可能迂到的一些问题及其原因和最可能的解决方案的列表。

对 z/OS 上的传输队列的访问权不足

症状

z/OS 上的集群发送方通道可能会报告它无权打开其传输队列。

原因

通道正在切换或已切换传输队列，并且通道启动程序未被授予访问新队列的权限。

解决方案

授予通道启动程序针对传输队列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。使用 `DEFCLXQ` 时，`SYSTEM.CLUSTER.TRANSMIT.*` 避免在新队列管理器加入集群时发生此问题。

移动消息失败

症状

消息停止由通道发送，并且它们仍在通道的旧传输队列上排队。

原因

由于发生不可恢复错误，因此队列管理器已停止将消息从旧传输队列移至新传输队列。例如，新的传输队列可能已变满或其后备存储器已耗尽。

解决方案

查看写入队列管理器错误日志（z/OS 上的作业日志）的错误消息，以确定问题并解决其根本原因。解析后，重新启动通道以恢复切换过程，或者停止通道，然后改为使用 `runswch1`（z/OS 上的 `CSQUTIL`）。

交换机未完成

症状

队列管理器反复发出指示其正在移动消息的消息。由于在旧的传输队列上始终存在消息，因此交换机永远不会完成。

原因 1

将通道的消息放入旧传输队列的速度比队列管理器将它们移动到新传输队列的速度要快。这可能是峰值工作负载期间的暂时性问题，因为如果是常见问题，那么通道不太可能能够以足够快的速度通过网络传输消息。

原因 2

旧传输队列上的通道存在未落实的消息。

解决方案

解析任何未落实消息的工作单元，和/或减少或暂挂应用程序工作负载，以允许移动消息阶段完成。

意外删除传输队列

症状 1

由于除去了匹配的 CLCHNAME 值，因此通道意外切换。

症状 2

放入集群队列失败，带有 MQRC_UNKNOWN_XMIT_Q。

症状 3

通道异常结束，因为其传输队列不存在。

症状 4

队列管理器无法移动消息以完成切换操作，因为它无法打开旧的或新的传输队列。

原因

已删除通道当前使用的传输队列或其先前的传输队列 (如果交换机尚未完成)。

解决方案

重新定义传输队列。如果是已删除的旧传输队列，那么管理员也可以使用带有 **-n** 参数的 **runswchl** (或带有 z/OS 上的 MOVEMSGS (NO) 的 CSQUTIL) 来完成切换操作。

请谨慎使用 **-n** 参数，因为如果不适当地使用该参数，那么通道的消息可以完成处理，但不会在旧的传输队列上更新。在此场景中，它是安全的，因为由于队列不存在，因此无法有任何消息来完成和完成处理。

对 RDQM 配置问题进行故障诊断

这些主题提供了可用于对 RDQM 高可用性 (HA) 和灾难恢复 (DR) 配置进行故障诊断的信息。

RDQM HA 体系结构

下面介绍了复制的数据队列管理器高可用性 (RDQM HA) 配置的基本体系结构，以便帮助进行故障诊断。

资源名称

针对每个 RDQM 队列管理器会创建不同的资源，而这些资源的名称基于该队列管理器的目录名称。此名称位于 `/var/mqm/mqs.ini` 文件中，在此处被称为 *qm*。例如，对于名为 TMPQM1 的 RDQM HA 队列管理器，*qm* 将为 tmpqm1。

体系结构

RDQM 高可用性 (HA) 体系结构包括 DRBD (用于数据复制) 和 Pacemaker (用于管理 HA RDQM 队列管理器的运行位置)。

创建 RDQM HA 队列管理器时，将完成以下步骤：

1. 创建一个 DRBD 资源来复制队列管理器的数据。
2. 创建并配置一个队列管理器以将该 DRBD 资源用于其存储。
3. 创建一组 Pacemaker 资源来监视和管理该队列管理器。

DRBD

针对每个 RDQM HA 队列管理器都会生成一个名为 `/etc/drbd.d/qm.res` 的 DRBD 资源文件。例如，在创建名为 HAQM1 的 RDQM HA 队列管理器时，DRBD 资源文件为 `/etc/drbd.d/haqm1.res`。

`.res` 文件中用于故障诊断目的的最重要信息是此特定 DRBD 资源的次设备号。DRBD 记录的许多消息都使用此次设备号。对于示例队列管理器 HAQM1，`.res` 文件包含以下信息：

```
device minor 100;
```

对于此队列管理器，您应该查找类似于以下示例的消息：

```
Jul 31 00:17:24 mqhavm13 kernel: drbd haqm1/0 drbd100 mqhavm15.gamsworthwilliam.com:
drbd_sync_handshake:
```

是否存在字符串 `drbd100` 将指示该消息是否与 HAQM1 相关。并非 DRBD 记录的所有消息都使用次设备号，某些消息会使用 DRBD 资源名称，即 RDQM HA 队列管理器的目录名称。例如：

```
Jul 31 00:17:22 mqhavm13 kernel: drbd haqm1 mqhavm15.gamsworthwilliam.com: Connection closed
```

Pacemaker

针对 RDQM HA 队列管理器会生成大量的 Pacemaker 资源：

qm

这是表示 RDQM HA 队列管理器的主资源。

p_rdqmx_qm

这是一个内部资源。

p_fs_qm

这是标准文件系统资源，用于将队列管理器的卷安装到 `/var/mqm/vols/qm` 上。

ms_drbd_qm

这是 RDQM 的 DRBD 资源的主/从属资源。

p_drbd_qm

这是 RDQM 的 DRBD 资源的原始资源。

如果为 HA RDQM 配置了浮动 IP 地址，那么还会配置其他资源：

p_ip_qm

示例 RDQM HA 配置和错误

下面介绍了一个示例 RDQM HA 配置，其中包含示例错误以及有关如何解决这些错误的信息。

示例 RDQM HA 组由以下三个节点组成：

- `mqhavm13.gamsworthwilliam.com`（称为 `vm13`）。
- `mqhavm14.gamsworthwilliam.com`（称为 `vm14`）。
- `mqhavm15.gamsworthwilliam.com`（称为 `vm15`）。

已创建以下三个 RDQM HA 队列管理器：

- HAQM1（在 `vm13` 上创建）
- HAQM2（在 `vm14` 上创建）
- HAQM3（在 `vm15` 上创建）

初始条件

以下列表中提供了每个节点的初始条件：

vm13

```
[midtownjojo@mqhavm13 ~]$ rdqmstatus -m HAQM1
Node: mqhavm13.gamsworthwilliam.com
Queue manager status: Running
CPU: 0.00%
Memory: 135MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: None
HA floating IP address: None

Node: mqhavm14.gamsworthwilliam.com
HA status: Normal

Node: mqhavm15.gamsworthwilliam.com
HA status: Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.

[midtownjojo@mqhavm13 ~]$ rdqmstatus -m HAQM2
Node: mqhavm13.gamsworthwilliam.com
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: mqhavm14.gamsworthwilliam.com
HA preferred location: mqhavm14.gamsworthwilliam.com
HA floating IP interface: None
HA floating IP address: None

Node: mqhavm14.gamsworthwilliam.com
HA status: Normal

Node: mqhavm15.gamsworthwilliam.com
HA status: Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.

[midtownjojo@mqhavm13 ~]$ rdqmstatus -m HAQM3
Node: mqhavm13.gamsworthwilliam.com
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: mqhavm15.gamsworthwilliam.com
HA preferred location: mqhavm15.gamsworthwilliam.com
HA floating IP interface: None
HA floating IP address: None

Node: mqhavm14.gamsworthwilliam.com
HA status: Normal

Node: mqhavm15.gamsworthwilliam.com
HA status: Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.
```

vm14

```
[midtownjojo@mqhavm14 ~]$ rdqmstatus -m HAQM1
Node: mqhavm14.gamsworthwilliam.com
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: mqhavm13.gamsworthwilliam.com
HA preferred location: mqhavm13.gamsworthwilliam.com
HA floating IP interface: None
HA floating IP address: None

Node: mqhavm13.gamsworthwilliam.com
HA status: Normal

Node: mqhavm15.gamsworthwilliam.com
HA status: Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.

[midtownjojo@mqhavm14 ~]$ rdqmstatus -m HAQM2
```

```

Node: mqhavam14.gamsworthwilliam.com
Queue manager status: Running
CPU: 0.00%
Memory: 135MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: None
HA floating IP address: None

Node: mqhavam13.gamsworthwilliam.com
HA status: Normal

Node: mqhavam15.gamsworthwilliam.com
HA status: Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.

[midtownjojo@mqhavam14 ~]$ rdqmstatus -m HAQM3
Node: mqhavam14.gamsworthwilliam.com
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: mqhavam15.gamsworthwilliam.com
HA preferred location: mqhavam15.gamsworthwilliam.com
HA floating IP interface: None
HA floating IP address: None

Node: mqhavam13.gamsworthwilliam.com
HA status: Normal

Node: mqhavam15.gamsworthwilliam.com
HA status: Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.

```

vm15

```

[midtownjojo@mqhavam15 ~]$ rdqmstatus -m HAQM1
Node: mqhavam15.gamsworthwilliam.com
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: mqhavam13.gamsworthwilliam.com
HA preferred location: mqhavam13.gamsworthwilliam.com
HA floating IP interface: None
HA floating IP address: None

Node: mqhavam13.gamsworthwilliam.com
HA status: Normal

Node: mqhavam14.gamsworthwilliam.com
HA status: Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.

[midtownjojo@mqhavam15 ~]$ rdqmstatus -m HAQM2
Node: mqhavam15.gamsworthwilliam.com
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: mqhavam14.gamsworthwilliam.com
HA preferred location: mqhavam14.gamsworthwilliam.com
HA floating IP interface: None
HA floating IP address: None

Node: mqhavam13.gamsworthwilliam.com
HA status: Normal

Node: mqhavam14.gamsworthwilliam.com
HA status: Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.

[midtownjojo@mqhavam15 ~]$ rdqmstatus -m HAQM3
Node: mqhavam15.gamsworthwilliam.com
Queue manager status: Running
CPU: 0.02%
Memory: 135MB

```



```

Queue manager file system:      51MB used, 1.0GB allocated [5%]
HA role:                        Primary
HA status:                      Normal
HA control:                     Enabled
HA current location:           This node
HA preferred location:         This node
HA floating IP interface:      None
HA floating IP address:        None

Node:                            mqhavam13.gamsworthwilliam.com
HA status:                      Normal

Node:                            mqhavam14.gamsworthwilliam.com
HA status:                      Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.

```

DRBD 场景

RDQM HA 配置将 DRBD 用于数据复制。以下场景展示了 DRBD 可能存在的以下问题：

- 丢失 DRBD 定额
- 单个 DRBD 连接断开
- 同步操作停滞不前

DRBD 场景 1：丢失 DRBD 定额

如果运行 RDQM HA 队列管理器的节点丢失对应于队列管理器的 DRBD 资源的 DRBD 定额，那么 DRBD 会立即开始从 I/O 操作返回错误，这会导致队列管理器开始生成 FDC 并最终停止。

如果其余两个节点具有 DRBD 资源的 DRBD 定额，那么 Pacemaker 会选择两个节点之一来启动队列管理器。因为丢失定额时原始节点上无更新，因此可以安全地从其他位置启动队列管理器。

下面是可用于监视 DRBD 定额丢失情况的两种主要方法：

- 使用 `rdqmstatus` 命令。
- 监视初始运行 RDQM HA 队列管理器的节点的系统日志。

rdqmstatus

如果是使用 `rdqmstatus` 命令，那么当节点 vm13 丢失 HAQM1 的 DRBD 资源的 DRBD 定额时，您可能会看到类似于以下示例的状态：

```

[midtownjojo@mqhavam13 ~]$ rdqmstatus -m HAQM1
Node:                            mqhavam13.gamsworthwilliam.com
Queue manager status:           Running elsewhere
HA role:                        Secondary
HA status:                      Remote unavailable
HA control:                     Enabled
HA current location:           mqhavam14.gamsworthwilliam.com
HA preferred location:         This node
HA floating IP interface:      None
HA floating IP address:        None

Node:                            mqhavam14.gamsworthwilliam.com
HA status:                      Remote unavailable
HA out of sync data:           0KB

Node:                            mqhavam15.gamsworthwilliam.com
HA status:                      Remote unavailable
HA out of sync data:           0KB
Command '/opt/mqm/bin/rdqmstatus' run with sudo.

```

请注意，HA status 已更改为 Remote unavailable，这指示与其他节点的两个 DRBD 连接都已断开。

在此情况下，其他两个节点具有 DRBD 资源的 DRBD 定额，因此 RDQM 正在 mqhavam14.gamsworthwilliam.com 上的其他位置运行，如 HA current location 值所示。

监视系统日志

如果是监视系统日志，那么您将看到 DRBD 在丢失资源定额时会记录一条消息：

```
Jul 30 09:38:36 mqhavam13 kernel: drbd haqm1/0 drbd100: quorum( yes -> no )
```

在定额复原时，系统会记录一条如下消息：

```
Jul 30 10:27:32 mqhavam13 kernel: drbd haqm1/0 drbd100: quorum( no -> yes )
```

DRBD 场景 2：单个 DRBD 连接断开

如果运行 RDQM HA 队列管理器的节点的两个 DRBD 连接中只有一个连接断开，那么队列管理器不会进行移动。

从与第一个场景相同的初始条件开始，在仅阻止一个 DRBD 复制链路后，vm13 上 **rdqmstatus** 报告的状态类似于以下示例：

```
Node:                               mqhavam13.gamsworthwilliam.com
Queue manager status:              Running
CPU:                                0.01%
Memory:                             133MB
Queue manager file system:         52MB used, 1.0GB allocated [5%]
HA role:                             Primary
HA status:                           Mixed
HA control:                          Enabled
HA current location:                 This node
HA preferred location:                This node
HA floating IP interface:            None
HA floating IP address:              None

Node:                               mqhavam14.gamsworthwilliam.com

HA status:                       Remote unavailable
HA out of sync data:             OKB

Node:                               mqhavam15.gamsworthwilliam.com
HA status:                           Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.
```

DRBD 场景 3：同步操作停滞不前

某些版本的 DRBD 都存在“同步操作似乎停滞不前”这个问题，这会在仍在同步到节点时阻止 RDQM HA 队列管理器故障转移到此节点。

可使用 `drbdadm status` 命令来查看此情况。在正常运行的情况下，将输出类似于以下示例的响应：

```
[midtownjojo@mqhavam13 ~]$ drbdadm status
haqm1 role:Primary
disk:UpToDate
mqhavam14.gamsworthwilliam.com role:Secondary
peer-disk:UpToDate
mqhavam15.gamsworthwilliam.com role:Secondary
peer-disk:UpToDate

haqm2 role:Secondary
disk:UpToDate
mqhavam14.gamsworthwilliam.com role:Primary
peer-disk:UpToDate
mqhavam15.gamsworthwilliam.com role:Secondary
peer-disk:UpToDate

haqm3 role:Secondary
disk:UpToDate
mqhavam14.gamsworthwilliam.com role:Secondary
peer-disk:UpToDate
mqhavam15.gamsworthwilliam.com role:Primary
peer-disk:UpToDate
```

如果同步操作停滞不前，那么将输出类似于以下示例的响应：

```
[midtownjojo@mqhavam13 ~]$ drbdadm status
haqm1 role:Primary
disk:UpToDate
mqhavam14.gamsworthwilliam.com role:Secondary
peer-disk:UpToDate
```

```

mqhavm15.gamsworthwilliam.com role:Secondary
  replication:SyncSource peer-disk:Inconsistent done:90.91

haqm2 role:Secondary
  disk:UpToDate
  mqhavm14.gamsworthwilliam.com role:Primary
  peer-disk:UpToDate
  mqhavm15.gamsworthwilliam.com role:Secondary
  peer-disk:UpToDate

haqm3 role:Secondary
  disk:UpToDate
  mqhavm14.gamsworthwilliam.com role:Secondary
  peer-disk:UpToDate
  mqhavm15.gamsworthwilliam.com role:Primary
  peer-disk:UpToDate

```

在此情况下，RDQM HA 队列管理器 HAQM1 无法移至 vm15，因为 vm15 上的磁盘处于 Inconsistent 状态。

done 值是完成百分比。如果此值不增加，那么您可以尝试断开此副本的连接，然后在 vm13 上使用以下命令（以 root 用户身份运行）来重新连接：

```

drbdadm disconnect haqm1:mqhavm15.gamsworthwilliam.com
drbdadm connect haqm1:mqhavm15.gamsworthwilliam.com

```

如果到两个辅助节点的复制操作停滞不前，那么可以执行 **disconnect** 和 **connect** 命令而不指定节点，这会断开两个连接：

```

drbdadm disconnect haqm1
drbdadm connect haqm1

```

Pacemaker 场景

RDQM HA 配置使用 Pacemaker 来确定 RDQM HA 队列管理器的运行位置。以下场景展示了 Pacemaker 可能存在的以下问题：

- 未调度 Corosync 主进程
- RDQM HA 队列管理器未在期望的位置运行

Pacemaker 场景 1：未调度 Corosync 主进程

如果您在系统日志中看到类似于以下示例的消息，这指示系统过于繁忙而无法为 Corosync 进程调度 CPU 时间，或者指示一种更常见的情况，即系统是虚拟机并且系统管理程序没有为整个虚拟机调度任何 CPU 时间。

```

corosync[10800]: [MAIN ] Corosync main process was not scheduled for 2787.0891 ms (threshold
is 1320.0000 ms). Consider token timeout increase.

```

Pacemaker（和 Corosync）和 DRBD 都有用于检测定额丢失情况的计时器，因此类似于此示例的消息指示，节点运行时间尚不足以从定额断开连接。Corosync 超时为 1.65 秒，1.32 秒的阈值是此时间的 80%，因此在主 Corosync 进程的调度延迟达到此超时的 80% 时，将打印示例中显示的消息。在示例中，接近 3 秒钟未调度该进程。必须解决导致此类问题的情况。可帮助解决类似情况的一种方法是降低虚拟机需求（例如，降低所需的 vCPU 数），因为这可使系统管理程序更轻松地调度虚拟机。

Pacemaker 场景 2：RDQM HA 队列管理器未在期望的位置运行

可帮助在此场景中进行故障诊断的主要工具是 **crm status** 命令。以下示例显示了一切正常时针对该配置给出的响应：

```

Stack: corosync
Current DC: mqhavm13.gamsworthwilliam.com (version 1.1.20.linbit-1+20190404+eab6a2092b71.e17.2-
eab6a2092b) - partition with quorum
Last updated: Tue Jul 30 09:11:29 2019
Last change: Tue Jul 30 09:10:34 2019 by root via crm_attribute on mqhavm14.gamsworthwilliam.com

3 nodes configured

```

18 resources configured

Online: [mqhavam13.gamsworthwilliam.com mqhavam14.gamsworthwilliam.com
mqhavam15.gamsworthwilliam.com]

Full list of resources:

```
Master/Slave Set: ms_drbd_haqm1 [p_drbd_haqm1]
  Masters: [ mqhavam13.gamsworthwilliam.com ]
  Slaves: [ mqhavam14.gamsworthwilliam.com mqhavam15.gamsworthwilliam.com ]
p_fs_haqm1 (ocf::heartbeat:Filesystem): Started mqhavam13.gamsworthwilliam.com
p_rdqmx_haqm1 (ocf::ibm:rdqmx): Started mqhavam13.gamsworthwilliam.com
haqm1 (ocf::ibm:rdqm): Started mqhavam13.gamsworthwilliam.com
Master/Slave Set: ms_drbd_haqm2 [p_drbd_haqm2]
  Masters: [ mqhavam14.gamsworthwilliam.com ]
  Slaves: [ mqhavam13.gamsworthwilliam.com mqhavam15.gamsworthwilliam.com ]
p_fs_haqm2 (ocf::heartbeat:Filesystem): Started mqhavam14.gamsworthwilliam.com
p_rdqmx_haqm2 (ocf::ibm:rdqmx): Started mqhavam14.gamsworthwilliam.com
haqm2 (ocf::ibm:rdqm): Started mqhavam14.gamsworthwilliam.com
Master/Slave Set: ms_drbd_haqm3 [p_drbd_haqm3]
  Masters: [ mqhavam15.gamsworthwilliam.com ]
  Slaves: [ mqhavam13.gamsworthwilliam.com mqhavam14.gamsworthwilliam.com ]
p_fs_haqm3 (ocf::heartbeat:Filesystem): Started mqhavam15.gamsworthwilliam.com
p_rdqmx_haqm3 (ocf::ibm:rdqmx): Started mqhavam15.gamsworthwilliam.com
haqm3 (ocf::ibm:rdqm): Started mqhavam15.gamsworthwilliam.com
```

请注意以下几点:

- 所有三个节点都显示为 Online。
- 每个 RDQM HA 队列管理器在创建它的节点上运行, 例如, HAQM1 在 vm13 上运行, 以此类推。

通过阻止 HAQM1 在 vm14 上运行并尝试将 HAQM1 移至 vm14 来构造此场景。HAQM1 无法在 vm14 上运行, 因为 vm14 上的文件 /var/mqm/mqs.ini 包含的值对于队列管理器 HAQM1 的目录无效。

通过在 vm13 上运行以下命令, 将 HAQM1 的首选位置更改为 vm14:

```
rdqmadm -m HAQM1 -n mqhavam14.gamsworthwilliam.com -p
```

此命令通常会导致 HAQM1 移至 vm14, 但是在此情况下, 检查 vm13 的状态将返回以下信息:

```
[midtonjojo@mqhavam13 ~]$ rdqmstatus -m HAQM1
Node: mqhavam13.gamsworthwilliam.com
Queue manager status: Running
CPU: 0.15%
Memory: 133MB
Queue manager file system: 52MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: mqhavam14.gamsworthwilliam.com
HA floating IP interface: None
HA floating IP address: None

Node: mqhavam14.gamsworthwilliam.com
HA status: Normal

Node: mqhavam15.gamsworthwilliam.com
HA status: Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.
```

HAQM1 仍在 vm13 上运行, 它未按请求那样移至 vm14, 并且需要调查原因。检查 Pacemaker 状态将给出以下响应:

```
[midtownjojo@mqhavam13 ~]$ crm status
Stack: corosync
Current DC: mqhavam13.gamsworthwilliam.com (version 1.1.20.linbit-1+20190404+eab6a2092b71.e17.2-
eab6a2092b) - partition with quorum
Last updated: Thu Aug 1 14:16:40 2019
Last change: Thu Aug 1 14:16:35 2019 by hacluster via crmd on mqhavam14.gamsworthwilliam.com

3 nodes configured
18 resources configured

Online: [ mqhavam13.gamsworthwilliam.com mqhavam14.gamsworthwilliam.com
mqhavam15.gamsworthwilliam.com ]
```

Full list of resources:

```
Master/Slave Set: ms_drbd_haqm1 [p_drbd_haqm1]
  Masters: [ mqhavm13.gamsworthwilliam.com ]
  Slaves: [ mqhavm14.gamsworthwilliam.com mqhavm15.gamsworthwilliam.com ]
p_fs_haqm1      (ocf::heartbeat:Filesystem): Started mqhavm13.gamsworthwilliam.com
p_rdqm_haqm1    (ocf::ibm:rdqm): Started mqhavm13.gamsworthwilliam.com
haqm1           (ocf::ibm:rdqm): Started mqhavm13.gamsworthwilliam.com
Master/Slave Set: ms_drbd_haqm2 [p_drbd_haqm2]
  Masters: [ mqhavm14.gamsworthwilliam.com ]
  Slaves: [ mqhavm13.gamsworthwilliam.com mqhavm15.gamsworthwilliam.com ]
p_fs_haqm2      (ocf::heartbeat:Filesystem): Started mqhavm14.gamsworthwilliam.com
p_rdqm_haqm2    (ocf::ibm:rdqm): Started mqhavm14.gamsworthwilliam.com
haqm2           (ocf::ibm:rdqm): Started mqhavm14.gamsworthwilliam.com
Master/Slave Set: ms_drbd_haqm3 [p_drbd_haqm3]
  Masters: [ mqhavm15.gamsworthwilliam.com ]
  Slaves: [ mqhavm13.gamsworthwilliam.com mqhavm14.gamsworthwilliam.com ]
p_fs_haqm3      (ocf::heartbeat:Filesystem): Started mqhavm15.gamsworthwilliam.com
p_rdqm_haqm3    (ocf::ibm:rdqm): Started mqhavm15.gamsworthwilliam.com
haqm3           (ocf::ibm:rdqm): Started mqhavm15.gamsworthwilliam.com
```

Failed Resource Actions:

```
* haqm1_monitor_0 on mqhavm14.gamsworthwilliam.com 'not installed' (5): call=372,
status=complete, exitreason='',
last-rc-change='Thu Aug 1 14:16:37 2019', queued=0ms, exec=17ms
```

记录所显示的 Failed Resource Actions 部分。

操作名称 `haqm1_monitor_0` 告诉我们，这是失败的 RDQM HAQM1 监视操作，其在 `mqhavm14.gamsworthwilliam.com` 上失败，似乎是 Pacemaker 尝试执行我们期望的操作并在 `vm14` 上启动 HAQM1，但由于某些原因而无法这样做。

您可通过查看 `last-rc-change` 参数的值来了解 Pacemaker 何时尝试执行此操作。

了解故障

要了解故障，需要查看发生故障时 `vm14` 的系统日志：

```
Aug 1 14:16:37 mqhavm14 crmd[26377]: notice: Result of probe operation for haqm1 on
mqhavm14.gamsworthwilliam.com: 5 (not installed)
```

此条目显示，在 Pacemaker 尝试检查 `vm14` 上 `haqm1` 的状态时，它收到一条错误，原因是没有配置 `haqm1`，这是因为在 `/var/mqm/mqs.ini` 中有意进行了错误的配置。

纠正该故障

要纠正该故障，必须先解决底层问题（在此情况下，复原 `vm14` 上 `/var/mqm/mqs.ini` 中 `haqm1` 的正确目录值）。然后，必须在相应的资源上使用命令 `crm resource cleanup` 来清除失败的操作，在此情况下为资源 `haqm1`，因为这是在失败操作中提及的资源。例如：

```
[midtownjojo@mqhavm13 ~]$ crm resource cleanup haqm1
Cleaned up haqm1 on mqhavm15.gamsworthwilliam.com
Cleaned up haqm1 on mqhavm14.gamsworthwilliam.com
Cleaned up haqm1 on mqhavm13.gamsworthwilliam.com
```

然后，再次检查 Pacemaker 状态：

```
[midtownjojo@mqhavm13 ~]$ crm status
Stack: corosync
Current DC: mqhavm13.gamsworthwilliam.com (version 1.1.20.linbit-1+20190404+eab6a2092b71.e17.2-
eab6a2092b) - partition with quorum
Last updated: Thu Aug 1 14:23:17 2019
Last change: Thu Aug 1 14:23:03 2019 by hacluster via crmd on mqhavm13.gamsworthwilliam.com

3 nodes configured
18 resources configured

Online: [ mqhavm13.gamsworthwilliam.com mqhavm14.gamsworthwilliam.com
mqhavm15.gamsworthwilliam.com ]

Full list of resources:
```

```

Master/Slave Set: ms_drbd_haqm1 [p_drbd_haqm1]
Masters: [ mqhavm14.gamsworthwilliam.com ]
Slaves: [ mqhavm13.gamsworthwilliam.com mqhavm15.gamsworthwilliam.com ]
p_fs_haqm1 (ocf::heartbeat:Filesystem): Started mqhavm14.gamsworthwilliam.com
p_rdqmx_haqm1 (ocf::ibm:rdqmx): Started mqhavm14.gamsworthwilliam.com
haqm1 (ocf::ibm:rdqm): Started mqhavm14.gamsworthwilliam.com
Master/Slave Set: ms_drbd_haqm2 [p_drbd_haqm2]
Masters: [ mqhavm14.gamsworthwilliam.com ]
Slaves: [ mqhavm13.gamsworthwilliam.com mqhavm15.gamsworthwilliam.com ]
p_fs_haqm2 (ocf::heartbeat:Filesystem): Started mqhavm14.gamsworthwilliam.com
p_rdqmx_haqm2 (ocf::ibm:rdqmx): Started mqhavm14.gamsworthwilliam.com
haqm2 (ocf::ibm:rdqm): Started mqhavm14.gamsworthwilliam.com
Master/Slave Set: ms_drbd_haqm3 [p_drbd_haqm3]
Masters: [ mqhavm15.gamsworthwilliam.com ]
Slaves: [ mqhavm13.gamsworthwilliam.com mqhavm14.gamsworthwilliam.com ]
p_fs_haqm3 (ocf::heartbeat:Filesystem): Started mqhavm15.gamsworthwilliam.com
p_rdqmx_haqm3 (ocf::ibm:rdqmx): Started mqhavm15.gamsworthwilliam.com
haqm3 (ocf::ibm:rdqm): Started mqhavm15.gamsworthwilliam.com

```

失败操作将消失，并且 HAQM1 现在按预期那样在 vm14 上运行。以下示例显示了 RDQM 状态：

```

[midtownjojo@mqhavm13 ~]$ rdqmstatus -m HAQM1
Node: mqhavm13.gamsworthwilliam.com
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: mqhavm14.gamsworthwilliam.com
HA preferred location: mqhavm14.gamsworthwilliam.com
HA floating IP interface: None
HA floating IP address: None

Node: mqhavm14.gamsworthwilliam.com
HA status: Normal

Node: mqhavm15.gamsworthwilliam.com
HA status: Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.

```

对安全性问题进行故障诊断

故障诊断信息可帮助您解决与安全性相关的问题。

对通道认证记录问题进行故障诊断

如果您在使用通道认证记录时发生问题，请检查在以下信息中是否描述了该问题。

您向队列管理器提供的是什么地址？

您的通道向队列管理器提供的地址取决于使用的网络适配器。例如，如果用于访问侦听器的 CONNAME 是“localhost”，那么提供 127.0.0.1 作为地址；如果它是计算机的实际 IP 地址，那么这是向队列管理器提供的地址。您可能会为 127.0.0.1 和您的实际 IP 地址调用不同的认证规则。

将 BLOCKADDR 与通道名称结合使用

如果使用 SET CHLAUTH TYPE(BLOCKADDR)，那么它必须具有通用通道名称 CHLAUTH(*) 而没有其他内容。必须使用任何通道名称阻止来自指定地址的访问。

z/OS 系统上的 CHLAUTH(*)

 在 z/OS 上，包含星号 (*) 的通道名称必须用引号括起。此规则也适用于使用单个星号来匹配所有通道名称。因此，如果在其他平台上指定 CHLAUTH(*)，那么在 z/OS 上必须指定 CHLAUTH('*')。

队列管理器重新启动期间 SET CHLAUTH 命令的行为

如果 SYSTEM.CHLAUTH.DATA.QUEUE 已通过某种方式删除或修改，以致于无法再访问（例如 PUT(DISABLED)），那么 SET CHLAUTH 命令将仅部分成功。在此情况下，SET CHLAUTH 将更新内存中高速缓存，但在固化时将失败。

这意味着尽管由 **SET CHLAUTH** 命令采用的规则可能最初可行，但是该命令的效果在队列管理器重新启动期间将不会持久。用户应进行调查，确保队列可访问，然后在重新启动队列管理器之前重新发出命令（使用 **ACTION(REPLACE)**）。

如果 `SYSTEM.CHLAUTH.DATA.QUEUE` 在队列管理器启动时保持不可访问，那么无法装入已保存的规则的高速缓存，并且所有通道都将受阻，直至队列和规则变为可访问为止。

z/OS 系统上 ADDRESS 和 ADDRLIST 的最大大小



在 z/OS 上，ADDRESS 和 ADDRLIST 字段的最大大小为 48 个字符。某些 IPv6 地址模式的长度可能超过此限制，例如，'0000-ffff:0000-ffff:0000-ffff:0000-ffff:0000-ffff:0000-ffff:0000-ffff'。在这种情况下，可改为使用 '*'。

如果要使用长度超过 48 个字符的模式，请尝试以其他方式表达需求。例如，并非指定

'0001-fffe:0001-fffe:0001-fffe:0001-fffe:0001-fffe:0001-fffe:0001-fffe:0001-fffe:0001-fffe' 作为 USERSRC(MAP) 的地址模式，您可以指定以下三个规则：

- USERSRC(MAP) 表示所有地址 (*)
- USERSRC(NOACCESS) for address '0000:0000:0000:0000:0000:0000:0000:0000'
- USERSRC(NOACCESS) for address 'ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff'

CipherSpec 不匹配

IBM MQ TLS 通道的两端必须使用相同的 CipherSpec。在 TLS 握手期间或在通道启动期间可能会检测到不匹配。

CipherSpec 标识加密算法与散列函数的组合。IBM MQ TLS 通道的两端必须使用相同的 CipherSpec，尽管它们可以以不同的方式指定 CipherSpec。可以在两个阶段检测到不匹配：

在 TLS 握手期间

当 TLS 客户机指定的 CipherSpec 对于连接的 TLS 服务器端的 TLS 支持不可接受时，TLS 握手将失败。TLS 客户机建议 TLS 服务器上的 TLS 供应不支持的 CipherSpec 时，在 TLS 握手期间发生 CipherSpec 故障。例如，当 AIX 上运行的 TLS 客户机向 IBM i 上运行的 TLS 服务器建议 DES_SHA_EXPORT1024 CipherSpec 时。

在通道启动期间

当为通道响应端定义的 CipherSpec 与为通道调用端定义的 CipherSpec 不匹配时，通道启动将失败。仅当通道的一端定义了 CipherSpec 时，通道启动也会失败。

请参阅 [指定 CipherSpecs](#) 以获取更多信息。

注：如果使用全局服务器证书，那么即使在两个通道定义上指定的 CipherSpecs 都匹配，也会在通道启动期间检测到不匹配。

全局服务器证书是一种特殊类型的证书，要求在使用它们的所有通信链路上建立最低级别的加密。如果 IBM MQ 通道配置请求的 CipherSpec 不满足此要求，那么将在 TLS 握手期间重新协商 CipherSpec。这是在 IBM MQ 通道启动期间检测到的故障，因为 CipherSpec 不再与通道上指定的 CipherSpec 匹配。

在这种情况下，请将通道两侧的 CipherSpec 更改为满足全局服务器证书需求的 CipherSpec。要确定已向您发放的证书是否为全局服务器证书，请与发放该证书的认证中心联系。

当 UNIX，Linux 或 Windows 系统上的 TLS 客户机通道指定了 DES_SHA_EXPORT1024 CipherSpec，并且 UNIX，Linux 或 Windows 系统上的相应 TLS 服务器通道正在使用 DES_SHA_EXPORT CipherSpec 时，TLS 服务器不会检测到不匹配情况。在这种情况下，通道正常运行。

TLS 握手期间认证失败

在 TLS 握手期间，有许多常见的认证失败原因。

这些原因包括但不限于以下列表中的原因：

在证书撤销列表或权限撤销列表中找到了证书

您可以根据认证中心发布的撤销列表来检查证书。

认证中心可以通过在证书撤销列表 (CRL) 或权限撤销列表 (ARL) 中发布不再可信的证书来撤销该证书。有关更多信息, 请参阅 [使用已撤销的证书](#)。

OCSP 响应程序已将证书标识为 "已撤销" 或 "未知"

您可以使用 OCSP 检查证书。OCSP 响应程序可以返回 "已撤销" 响应 (指示证书不再有效) 或 "未知" (指示它没有该证书的撤销数据)。有关更多信息, 请参阅 [使用已撤销的证书](#)。

证书已到期或尚未处于活动状态

每个数字证书都有一个有效日期和一个不再有效的日期, 因此尝试使用其生存期之外的证书进行认证失败。

证书已损坏

如果数字证书中的信息不完整或已损坏, 那么认证将失败。

不支持证书

如果证书的格式不受支持, 那么即使证书仍在其生存期内, 认证也会失败。

TLS 客户机没有证书

TLS 服务器始终验证客户机证书 (如果发送了客户机证书)。如果 TLS 客户机未发送证书, 那么在定义充当 TLS 服务器的通道结束时, 认证将失败:

- 将 SSLCAUTH 参数设置为 REQUIRED 或
- 使用 SSLPEER 参数值

没有匹配的 CA 根证书, 或者证书链不完整

每个数字证书都由认证中心 (CA) 发放, 该认证中心还提供包含 CA 公用密钥的根证书。根证书由签发 CA 本身签署。如果正在执行认证的计算机上的密钥存储库不包含发出出局用户证书的 CA 的有效根证书, 那么认证将失败。

认证通常涉及可信证书链。用户证书上的数字签名通过颁发 CA 的证书中的公用密钥进行验证。如果该 CA 证书是根证书, 那么验证过程已完成。如果该 CA 证书是由中间 CA 颁发的, 那么必须验证中间 CA 证书上的数字签名本身。此过程将沿着 CA 证书链继续执行, 直到到达根证书为止。在这种情况下, 必须正确验证链中的所有证书。如果正在执行认证的计算机上的密钥存储库不包含发出出局根证书的 CA 的有效根证书, 那么认证将失败。

但是, 只要信任锚 (ROOT CA) 存在, 某些 TLS 实现 (例如 GSKit, DCM 和 RACF) 就会验证证书, 而某些中间 CA 在信任链中不存在。因此, 确保服务器端证书库包含完整的信任链非常重要。此外, 不得使用选择性除去签署者 (CA) 证书的方法来管理与队列管理器的连接。

有关更多信息, 请参阅 [证书链的工作方式](#)。

有关本主题中使用的术语的更多信息, 请参阅:

- [传输层安全性 \(TLS\) 概念](#)
- [数字证书](#)

对 TLS 问题进行故障诊断

使用下列信息来帮助解决 TLS 系统问题。

概述

对于由在客户机上启用 *FIPS* 的情况下使用非 *FIPS* 密码导致的错误, 您将收到以下错误消息:

JMSCMQ001

IBM MQ 调用失败, 完成代码 2 ('MQCC_FAILED'), 原因 2397 ('MQRC_JSSE_ERROR')

对于本主题中记录的每个其他问题, 您将收到上一条错误消息和/或以下错误消息:

JMSWMQ0018

无法使用连接方式 '*connection_mode*' 和主机名 '*host_name*' 连接到队列管理器 '*queue_manager_name*'

对于本主题中记录的每个问题, 提供了以下信息:

- 来自样本 `SystemOut.log` 或 `Console` 的输出，其中详述异常原因。
- 队列管理器错误日志信息。
- 问题的解决方案。

注:

- 您应始终列出堆栈和第一个异常的原因。
- 是否将错误信息写入 `stdout` 日志文件，取决于应用程序的编写方式以及您正在使用的框架。
- 样本代码包含堆栈和行号。此信息是有用的指导，但是堆栈和行号可能根据修订包而异。您应在堆栈和行号的指导下找到正确的部分，而不是将信息专用于诊断目的。

未在客户机上设置密码套件

Output

原因:

```
com.ibm.mq.jmqi.JmqiException: CC=2;RC=2397;AMQ9641: Remote CipherSpec error for channel
'SYSTEM.DEF.SVRCONN' to host ''. [3=SYSTEM.DEF.SVRCONN]
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.analyseErrorSegment(RemoteConnection.java:4176)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.receiveTSH(RemoteConnection.java:2969)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.initSess(RemoteConnection.java:1180)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.connect(RemoteConnection.java:838)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSessionFromNewConnection
(RemoteConnectionSpecification.java:409)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSession
(RemoteConnectionSpecification.java:305)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionPool.getSession(RemoteConnectionPool.java:146)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1868)
```

队列管理器错误日志

AMQ9639: 远程通道 'SYSTEM.DEF.SVRCONN' 未指定 CipherSpec。

解决方案

在客户机上设置 CipherSuite，以便通道两端均有匹配的 CipherSuite 或 CipherSpec 对。

未在服务器上设置密码套件

Output

原因:

```
com.ibm.mq.jmqi.JmqiException: CC=2;RC=2397;AMQ9641: Remote CipherSpec error
for channel 'SYSTEM.DEF.SVRCONN' to host ''. [3=SYSTEM.DEF.SVRCONN]
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.analyseErrorSegment(RemoteConnection.java:4176)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.receiveTSH(RemoteConnection.java:2969)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.initSess(RemoteConnection.java:1180)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.connect(RemoteConnection.java:838)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSessionFromNewConnection
(RemoteConnectionSpecification.java:409)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSession
(RemoteConnectionSpecification.java:305)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionPool.getSession(RemoteConnectionPool.java:146)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1868)
```

队列管理器错误日志

AMQ9639: 远程通道 'SYSTEM.DEF.SVRCONN' 未指定 CipherSpec。

解决方案

更改通道 `SYSTEM.DEF.SVRCONN` 以指定有效的 CipherSpec。

密码不匹配

Output

原因:

```
com.ibm.mq.jmqi.JmqiException: CC=2;RC=2397;AMQ9641: Remote CipherSpec error
for channel 'SYSTEM.DEF.SVRCONN' to host ''. [3=SYSTEM.DEF.SVRCONN]
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.analyseErrorSegment(RemoteConnection.java:4176)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.receiveTSH(RemoteConnection.java:2969)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.initSess(RemoteConnection.java:1180)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.connect(RemoteConnection.java:838)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSessionFromNewConnection
(RemoteConnectionSpecification.java:409)
```

```
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSession
(RemoteConnectionSpecification.java:305)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionPool.getSession(RemoteConnectionPool.java:146)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1868)
```

队列管理器错误日志

AMQ9631: TLS 握手期间协调的 CipherSpec 与通道 'SYSTEM.DEF.SVRCONN' 的所需 CipherSpec 不匹配。

解决方案

更改服务器连接通道的 SSLCIPH 定义或客户机的密码套件，以便两端均有匹配的 CipherSuite 或 CipherSpec 对。

缺少客户机个人证书

Output

原因:

```
com.ibm.mq.jmqi.JmqiException: CC=2;RC=2059;AMQ9503: Channel negotiation failed. [3=SYSTEM.DEF.SVRCONN]
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.analyseErrorSegment(RemoteConnection.java:4176)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.receiveTSH(RemoteConnection.java:2969)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.initSess(RemoteConnection.java:1180)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.connect(RemoteConnection.java:838)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSessionFromNewConnection
(RemoteConnectionSpecification.java:409)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSession
(RemoteConnectionSpecification.java:305)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionPool.getSession(RemoteConnectionPool.java:146)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1868)
```

队列管理器错误日志

AMQ9637: 通道缺少证书。

解决方案

确保队列管理器的密钥数据库包含来自客户机信任库的已签名的个人证书。

缺少服务器个人证书

Output

原因:

```
com.ibm.mq.jmqi.JmqiException: CC=2;RC=2397;AMQ9771: SSL handshake failed.
[1=javax.net.ssl.SSLHandshakeException[Remote host closed connection during handshake],
3=localhost/127.0.0.1:1418 (localhost),4=SSLSocket.startHandshake,5=default]
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection.protocolConnect(RemoteTCPConnection.java:1173)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.connect(RemoteConnection.java:835)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSessionFromNewConnection
(RemoteConnectionSpecification.java:409)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSession
(RemoteConnectionSpecification.java:305)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionPool.getSession(RemoteConnectionPool.java:146)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1868)
... 12 more
```

原因:

```
javax.net.ssl.SSLHandshakeException: Remote host closed connection during handshake
at com.ibm.jsse2.qc.a(qc.java:158)
at com.ibm.jsse2.qc.h(qc.java:185)
at com.ibm.jsse2.qc.a(qc.java:566)
at com.ibm.jsse2.qc.startHandshake(qc.java:120)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection$6.run(RemoteTCPConnection.java:1142)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection$6.run(RemoteTCPConnection.java:1134)
at java.security.AccessController.doPrivileged(AccessController.java:229)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection.protocolConnect(RemoteTCPConnection.java:1134)
... 17 more
```

原因:

```
java.io.EOFException: SSL peer shut down incorrectly
at com.ibm.jsse2.a.a(a.java:19)
at com.ibm.jsse2.qc.a(qc.java:207)
```

队列管理器错误日志

AMQ9637: 通道缺少证书。

解决方案

确保队列管理器的密钥数据库包含来自客户机信任库的已签名的个人证书。

在客户机上缺少服务器签署者

Output

原因:

```
com.ibm.mq.jmqi.JmqiException: CC=2;RC=2397;AMQ9771: SSL handshake failed.
[1=javax.net.ssl.SSLHandshakeException[com.ibm.jsse2.util.j:
PKIX path validation failed: java.security.cert.CertPathValidatorException:
The certificate issued by CN=JohnDoe, O=COMPANY, L=YOURSITE, C=XX is not trusted; internal cause is:
java.security.cert.CertPathValidatorException: Signature does not match.],3=localhost/127.0.0.1:1418
(localhost),4=SSLSocket.startHandshake,5=default]
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection.protocolConnect(RemoteTCPConnection.java:1173)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.connect(RemoteConnection.java:835)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSessionFromNewConnection
(RemoteConnectionSpecification.java:409)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSession
(RemoteConnectionSpecification.java:305)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionPool.getSession(RemoteConnectionPool.java:146)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1868)
...
```

原因:

```
javax.net.ssl.SSLHandshakeException: com.ibm.jsse2.util.j: PKIX path validation failed:
java.security.cert.CertPathValidatorException:
The certificate issued by CN=JohnDoe, O=COMPANY, L=YOURSITE, C=XX is not trusted;
internal cause is: java.security.cert.CertPathValidatorException: Signature does not match.
...
```

原因:

```
com.ibm.jsse2.util.j: PKIX path validation failed: java.security.cert.CertPathValidatorException:
The certificate issued by CN=JohnDoe, O=COMPANY, L=YOURSITE, C=XX is not trusted;
internal cause is: java.security.cert.CertPathValidatorException: Signature does not match.
at com.ibm.jsse2.util.h.a(h.java:99)
at com.ibm.jsse2.util.h.b(h.java:27)
at com.ibm.jsse2.util.g.a(g.java:14)
at com.ibm.jsse2.yc.a(yc.java:68)
at com.ibm.jsse2.yc.a(yc.java:17)
at com.ibm.jsse2.yc.checkServerTrusted(yc.java:154)
at com.ibm.jsse2.bb.a(bb.java:246)
... 28 more
```

原因:

```
java.security.cert.CertPathValidatorException:
The certificate issued by CN=JohnDoe, O=COMPANY, L=YOURSITE, C=XX is not trusted;
internal cause is: java.security.cert.CertPathValidatorException: Signature does not match.
at com.ibm.security.cert.BasicChecker.(BasicChecker.java:111)
at com.ibm.security.cert.PKIXCertPathValidatorImpl.engineValidate(PKIXCertPathValidatorImpl.java:174)
at java.security.cert.CertPathValidator.validate(CertPathValidator.java:265)
at com.ibm.jsse2.util.h.a(h.java:13)
... 34 more
```

原因:

```
java.security.cert.CertPathValidatorException: Signature does not match.
at com.ibm.security.cert.CertPathUtil.findIssuer(CertPathUtil.java:297)
at com.ibm.security.cert.BasicChecker.(BasicChecker.java:108)
```

队列管理器错误日志

AMQ9665: 通道 '????' 的远端已关闭 SSL 连接。

解决方案

将用于签署队列管理器的个人证书的证书添加到客户机的信任库。

在服务器上缺少客户机签署者

Output

原因:

```
com.ibm.mq.jmqi.JmqiException: CC=2;RC=2397;AMQ9771: SSL handshake failed.
[1=java.net.SocketException[Software caused connection abort: socket write error],
3=localhost/127.0.0.1:1418 (localhost),4=SSLSocket.startHandshake,5=default]
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection.protocolConnect(RemoteTCPConnection.java:1173)
```

```

at com.ibm.mq.jmqi.remote.impl.RemoteConnection.connect(RemoteConnection.java:835)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSessionFromNewConnection
(RemoteConnectionSpecification.java:409)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSession
(RemoteConnectionSpecification.java:305)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionPool.getSession(RemoteConnectionPool.java:146)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1868)
... 12 more

```

原因:

```

java.net.SocketException: Software caused connection abort: socket write error
at java.net.SocketOutputStream.socketWrite(SocketOutputStream.java:120)
at java.net.SocketOutputStream.write(SocketOutputStream.java:164)
at com.ibm.jsse2.c.a(c.java:57)
at com.ibm.jsse2.c.a(c.java:34)
at com.ibm.jsse2.qc.b(qc.java:527)
at com.ibm.jsse2.qc.a(qc.java:635)
at com.ibm.jsse2.qc.a(qc.java:743)
at com.ibm.jsse2.ab.a(ab.java:550)
at com.ibm.jsse2.bb.b(bb.java:194)
at com.ibm.jsse2.bb.a(bb.java:162)
at com.ibm.jsse2.bb.a(bb.java:7)
at com.ibm.jsse2.ab.r(ab.java:529)
at com.ibm.jsse2.ab.a(ab.java:332)
at com.ibm.jsse2.qc.a(qc.java:435)
at com.ibm.jsse2.qc.h(qc.java:185)
at com.ibm.jsse2.qc.a(qc.java:566)
at com.ibm.jsse2.qc.startHandshake(qc.java:120)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection$6.run(RemoteTCPConnection.java:1142)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection$6.run(RemoteTCPConnection.java:1134)
at java.security.AccessController.doPrivileged(AccessController.java:229)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection.protocolConnect(RemoteTCPConnection.java:1134)

```

队列管理器错误日志

AMQ9633: 通道 '????' 的 SSL 证书错误。

解决方案

将用于签署客户机的个人证书的证书添加到队列管理器的密钥数据库。

服务器上设置的 SSLPEER 与证书不匹配

Output

原因:

```

com.ibm.mq.jmqi.JmqiException: CC=2;RC=2397;AMQ9643: Remote SSL peer name error for channel
'SYSTEM.DEF.SVRCONN' on host ''. [3=SYSTEM.DEF.SVRCONN]
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.analyseErrorSegment(RemoteConnection.java:4176)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.receiveTSH(RemoteConnection.java:2969)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.initSess(RemoteConnection.java:1180)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.connect(RemoteConnection.java:838)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSessionFromNewConnection
(RemoteConnectionSpecification.java:409)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSession
(RemoteConnectionSpecification.java:305)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionPool.getSession(RemoteConnectionPool.java:146)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1868)

```

队列管理器错误日志

AMQ9636: SSL 专有名称与对等名称不匹配, 通道 'SYSTEM.DEF.SVRCONN'。

解决方案

确保在服务器连接通道上设置的 SSLPEER 的值与证书的专有名称匹配。

客户机上设置的 SSLPEER 与证书不匹配

Output

原因:

```

com.ibm.mq.jmqi.JmqiException: CC=2;RC=2398;AMQ9636: SSL distinguished name does not match peer name,
channel '?'. [CN=JohnDoe, O=COMPANY, L=YOURSITE, C=XX]
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection.protocolConnect(RemoteTCPConnection.java:1215)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.connect(RemoteConnection.java:835)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSessionFromNewConnection
(RemoteConnectionSpecification.java:409)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSession
(RemoteConnectionSpecification.java:305)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionPool.getSession(RemoteConnectionPool.java:146)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1868)

```

队列管理器错误日志

AMQ9208: 从主机 *host-name (address)* 接收时发生错误。

解决方案

确保客户机中设置的 SSLPEER 的值与证书的专有名称匹配。

在客户机上启用 FIPS 的情况下使用非 FIPS 密码

Output

```
Check the queue manager is started and if running in client mode, check there is a listener running.
Please see the linked exception for more information.
at com.ibm.msg.client.wmq.common.internal.Reason.reasonToException(Reason.java:578)
at com.ibm.msg.client.wmq.common.internal.Reason.createException(Reason.java:214)
at com.ibm.msg.client.wmq.internal.WMQConnection.getConnectOptions(WMQConnection.java:1423)
at com.ibm.msg.client.wmq.internal.WMQConnection.(WMQConnection.java:339)
at com.ibm.msg.client.wmq.factories.WMQConnectionFactory.createV7ProviderConnection
(WMQConnectionFactory.java:6865)
at com.ibm.msg.client.wmq.factories.WMQConnectionFactory.createProviderConnection
(WMQConnectionFactory.java:6221)
at com.ibm.msg.client.jms.admin.JmsConnectionFactoryImpl._createConnection
(JmsConnectionFactoryImpl.java:285)
at com.ibm.msg.client.jms.admin.JmsConnectionFactoryImpl.createConnection
(JmsConnectionFactoryImpl.java:233)
at com.ibm.mq.jms.MQConnectionFactory.createCommonConnection(MQConnectionFactory.java:6016)
at com.ibm.mq.jms.MQConnectionFactory.createConnection(MQConnectionFactory.java:6041)
at tests.SimpleSSLConn.runTest(SimpleSSLConn.java:46)
at tests.SimpleSSLConn.main(SimpleSSLConn.java:26)
```

原因:

```
com.ibm.mq.MQException: JMSCMQ0001: IBM MQ call failed with compcode '2' ('MQCC_FAILED')
reason '2400' ('MQRC_UNSUPPORTED_CIPHER_SUITE').
at com.ibm.msg.client.wmq.common.internal.Reason.createException(Reason.java:202)
```

队列管理器错误日志

不适用。

解决方案

使用支持 FIPS 的密码，或者在客户机上禁用 FIPS。

在队列管理器上启用 FIPS 的情况下使用非 FIPS 密码

Output

原因:

```
com.ibm.mq.jmqi.JmqiException: CC=2;RC=2397;AMQ9771: SSL handshake failed.
[1=javax.net.ssl.SSLHandshakeException[Received fatal alert: handshake_failure],
3=localhost/127.0.0.1:1418 (localhost),4=SSLSocket.startHandshake,5=default]
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection.protocolConnect(RemoteTCPConnection.java:1173)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.connect(RemoteConnection.java:835)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSessionFromNewConnection
(RemoteConnectionSpecification.java:409)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSession
(RemoteConnectionSpecification.java:305)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionPool.getSession(RemoteConnectionPool.java:146)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1868)
... 12 more
```

原因:

```
javax.net.ssl.SSLHandshakeException: Received fatal alert: handshake_failure
at com.ibm.jsse2.j.a(j.java:13)
at com.ibm.jsse2.j.a(j.java:18)
at com.ibm.jsse2.qc.b(qc.java:601)
at com.ibm.jsse2.qc.a(qc.java:100)
at com.ibm.jsse2.qc.h(qc.java:185)
at com.ibm.jsse2.qc.a(qc.java:566)
at com.ibm.jsse2.qc.startHandshake(qc.java:120)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection$6.run(RemoteTCPConnection.java:1142)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection$6.run(RemoteTCPConnection.java:1134)
at java.security.AccessController.doPrivileged(AccessController.java:229)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection.protocolConnect(RemoteTCPConnection.java:1134)
```

队列管理器错误日志

AMQ9616: 未在服务器上启用建议的 CipherSpec。

解决方案

使用支持 FIPS 的密码，或者在队列管理器上禁用 FIPS。

找不到使用 IBM JRE 的客户机密钥库

Output

原因:

```
com.ibm.mq.jmqi.JmqiException: CC=2;RC=2059;AMQ9204: Connection to host 'localhost(1418)' rejected.
[1=com.ibm.mq.jmqi.JmqiException[CC=2;RC=2059;AMQ9503: Channel negotiation failed.
[3=SYSTEM.DEF.SVRCONN]],3=localhost(1418),5=RemoteConnection.analyseErrorSegment]
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:2450)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1396)
at com.ibm.mq.eso.jmqi.InterceptedJmqiImpl.jmqiConnect(InterceptedJmqiImpl.java:376)
at com.ibm.mq.eso.jmqi.ESEJMQI.jmqiConnect(ESEJMQI.java:561)
at com.ibm.msg.client.wmq.internal.WMQConnection.(WMQConnection.java:342)
... 8 more
```

原因:

```
com.ibm.mq.jmqi.JmqiException: CC=2;RC=2059;AMQ9503: Channel negotiation failed. [3=SYSTEM.DEF.SVRCONN]
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.analyseErrorSegment(RemoteConnection.java:4176)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.receiveTSH(RemoteConnection.java:2969)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.initSess(RemoteConnection.java:1180)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.connect(RemoteConnection.java:838)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSessionFromNewConnection
(RemoteConnectionSpecification.java:409)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSession
(RemoteConnectionSpecification.java:305)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionPool.getSession(RemoteConnectionPool.java:146)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1868)
```

队列管理器错误日志

AMQ9637: 通道缺少证书。

解决方案

确保 JVM 属性 `javax.net.ssl.keyStore` 指定有效密钥库的位置。

找不到使用 Oracle JRE 的客户机密钥库

Output

原因:

```
java.security.PrivilegedActionException: java.io.FileNotFoundException:
C:\filepath\wrongkey.jks (The system cannot find the file specified)
at java.security.AccessController.doPrivileged(Native Method)
at sun.security.ssl.SSLContextImpl$DefaultSSLContext.getDefaultKeyManager(Unknown Source)
at sun.security.ssl.SSLContextImpl$DefaultSSLContext.(Unknown Source)
at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
at sun.reflect.NativeConstructorAccessorImpl.newInstance(Unknown Source)
at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(Unknown Source)
at java.lang.reflect.Constructor.newInstance(Unknown Source)
at java.lang.Class.newInstance0(Unknown Source)
at java.lang.Class.newInstance(Unknown Source)
... 28 more
```

原因:

```
java.io.FileNotFoundException: C:\filepath\wrongkey.jks (The system cannot find the file specified)
at java.io.FileInputStream.open(Native Method)
at java.io.FileInputStream.(Unknown Source)
at java.io.FileInputStream.(Unknown Source)
at sun.security.ssl.SSLContextImpl$DefaultSSLContext$2.run(Unknown Source)
at sun.security.ssl.SSLContextImpl$DefaultSSLContext$2.run(Unknown Source)
```

队列管理器错误日志

AMQ9637: 通道缺少证书。

解决方案

确保 JVM 属性 `javax.net.ssl.keyStore` 指定有效密钥库的位置。

密钥库密码错误 - IBM JRE

Output

原因:

```

com.ibm.mq.jmqi.JmqiException: CC=2;RC=2059;AMQ9503: Channel negotiation failed. [3=SYSTEM.DEF.SVRCONN]
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.analyseErrorSegment(RemoteConnection.java:4176)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.receiveTSH(RemoteConnection.java:2969)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.initSess(RemoteConnection.java:1180)
at com.ibm.mq.jmqi.remote.impl.RemoteConnection.connect(RemoteConnection.java:838)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSessionFromNewConnection
(RemoteConnectionSpecification.java:409)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionSpecification.getSession
(RemoteConnectionSpecification.java:305)
at com.ibm.mq.jmqi.remote.impl.RemoteConnectionPool.getSession(RemoteConnectionPool.java:146)
at com.ibm.mq.jmqi.remote.api.RemoteFAP.jmqiConnect(RemoteFAP.java:1868)

```

队列管理器错误日志

AMQ9637: 通道缺少证书。

解决方案

确保 JVM 属性 `javax.net.ssl.keyStorePassword` 的值指定 `javax.net.ssl.keyStore` 所指定的密钥库的密码。

信任库密码错误 - IBM JRE

Output

原因:

```

javax.net.ssl.SSLHandshakeException: java.security.cert.CertificateException:
No X509TrustManager implementation available
at com.ibm.jsse2.j.a(j.java:13)
at com.ibm.jsse2.qc.a(qc.java:204)
at com.ibm.jsse2.ab.a(ab.java:342)
at com.ibm.jsse2.ab.a(ab.java:222)
at com.ibm.jsse2.bb.a(bb.java:157)
at com.ibm.jsse2.bb.a(bb.java:492)
at com.ibm.jsse2.ab.r(ab.java:529)
at com.ibm.jsse2.ab.a(ab.java:332)
at com.ibm.jsse2.qc.a(qc.java:435)
at com.ibm.jsse2.qc.h(qc.java:185)
at com.ibm.jsse2.qc.a(qc.java:566)
at com.ibm.jsse2.qc.startHandshake(qc.java:120)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection$6.run(RemoteTCPConnection.java:1142)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection$6.run(RemoteTCPConnection.java:1134)
at java.security.AccessController.doPrivileged(AccessController.java:229)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection.protocolConnect(RemoteTCPConnection.java:1134)
... 17 more

```

原因:

```

java.security.cert.CertificateException: No X509TrustManager implementation available
at com.ibm.jsse2.xc.checkServerTrusted(xc.java:2)
at com.ibm.jsse2.bb.a(bb.java:246)

```

队列管理器错误日志

AMQ9665: 通道 '????' 的远端已关闭 SSL 连接。

解决方案

确保 JVM 属性 `javax.net.ssl.trustStorePassword` 的值指定 `javax.net.ssl.trustStore` 所指定的密钥库的密码。

找不到或打不开队列管理器密钥数据库

Output

原因:

```

javax.net.ssl.SSLHandshakeException: Remote host closed connection during handshake
at com.ibm.jsse2.qc.a(qc.java:158)
at com.ibm.jsse2.qc.h(qc.java:185)
at com.ibm.jsse2.qc.a(qc.java:566)
at com.ibm.jsse2.qc.startHandshake(qc.java:120)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection$6.run(RemoteTCPConnection.java:1142)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection$6.run(RemoteTCPConnection.java:1134)
at java.security.AccessController.doPrivileged(AccessController.java:229)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection.protocolConnect(RemoteTCPConnection.java:1134)
... 17 more

```

原因:

```
java.io.EOFException: SSL peer shut down incorrectly
at com.ibm.jsse2.a.a(a.java:19)
at com.ibm.jsse2.qc.a(qc.java:207)
```

队列管理器错误日志

AMQ9657: 打不开密钥存储库 (通道 '????')。

解决方案

确保指定的密钥存储库存在，并且其许可权使涉及的 IBM MQ 进程可以从该密钥存储库中进行读取。

找不到或无法使用队列管理器密钥数据库密码隐藏文件

Output

原因:

```
javax.net.ssl.SSLHandshakeException: Remote host closed connection during handshake
at com.ibm.jsse2.qc.a(qc.java:158)
at com.ibm.jsse2.qc.h(qc.java:185)
at com.ibm.jsse2.qc.a(qc.java:566)
at com.ibm.jsse2.qc.startHandshake(qc.java:120)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection$6.run(RemoteTCPConnection.java:1142)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection$6.run(RemoteTCPConnection.java:1134)
at java.security.AccessController.doPrivileged(AccessController.java:229)
at com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection.protocolConnect(RemoteTCPConnection.java:1134)
... 17 more
```

原因:

```
ava.io.EOFException: SSL peer shut down incorrectly
at com.ibm.jsse2.a.a(a.java:19)
at com.ibm.jsse2.qc.a(qc.java:207)
```

队列管理器错误日志

AMQ9660: SSL 证书密钥库: 密码隐藏文件缺失或不可用。

解决方案

确保密码隐藏文件已与同一目录中的密钥数据库文件相关联，并且用于运行 IBM MQ 的用户标识对两个文件均具有读访问权。

对 IBM MQ 问题的 WCF 定制通道进行故障诊断

相关概念

第 303 页的『[WCF XMS First Failure Support Technology \(FFST\)](#)』

您可以通过使用 IBM MQ 跟踪收集有关不同的 IBM MQ 代码部分有何作用的详细信息。XMS FFST 针对 WCF 定制通道具有自己的配置和输出文件。

相关任务

第 372 页的『[跟踪 IBM MQ 的 WCF 定制通道](#)』

您可以使用 IBM MQ 跟踪收集有关不同的 IBM MQ 代码部分有何作用的详细信息。使用 Windows Communication Foundation (WCF) 时，将为与 Microsoft WCF 基础结构跟踪集成的 Microsoft Windows Communication Foundation (WCF) 定制通道跟踪生成单独的跟踪输出。

第 227 页的『[联系 IBM 支持人员](#)』

如果您在 IBM MQ 方面遇到问题需要帮助，可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订、故障诊断和其他新闻的通知。

[使用 IBM MQ 开发 Microsoft Windows Communication Foundation 应用程序](#)

WCF 定制通道异常层次结构

定制通道抛出的异常类型与 WCF 一致，且通常是 `TimeoutException` 或 `CommunicationException`（或 `CommunicationException` 的子类）。使用链接的或内部的异常提供错误条件的更多详细信息（如果可用）。

SOAP/JMS 接口

以下异常是典型示例，通道的体系结构中的每个层都可能导致其他链接异常，例如，`CommunicationsException` 具有链接的 `XMSEException`，后者又具有链接的 `MQException`：

1. `System.ServiceModel.CommunicationsExceptions`
2. `IBM.XMS.XMSEException`
3. `IBM.WMQ.MQException`

在层次结构中最高的 `CommunicationException` 数据集中捕获并提供关键信息。这种数据捕获和供应防止应用程序需要链接到通道体系结构中的每个层，以便询问链接异常以及这些异常可能包含的任何其他信息。定义了以下键名称：

- `IBM.XMS.WCF.ErrorCode`：当前定制通道异常的错误消息代码。
- `IBM.XMS.ErrorCode`：堆栈中第一个 XMS 异常的错误消息。
- `IBM.WMQ.ReasonCode`：底层 IBM MQ 原因码。
- `IBM.WMQ.CompletionCode`：底层 IBM MQ 完成代码。

非 SOAP/非 JMS 接口

以下异常是典型示例，通道的体系结构中的每个层都可能导致其他链接异常，例如，`CommunicationsException` 具有链接的 `MQException`：

1. `System.ServiceModel.CommunicationsExceptions`
2. `IBM.WMQ.MQException`

在层次结构中最高的 `CommunicationException` 数据集中捕获并提供关键信息。这种数据捕获和供应防止应用程序需要链接到通道体系结构中的每个层，以便询问链接异常以及这些异常可能包含的任何其他信息。定义了以下键名称：

- `IBM.WMQ.WCF.ErrorCode`：当前定制通道异常的错误消息代码。
- `IBM.WMQ.ReasonCode`：底层 IBM MQ 原因码。
- `IBM.WMQ.CompletionCode`：底层 IBM MQ 完成代码。

WCF 版本信息

WCF 版本信息有助于确定问题，并且包含在定制通道的组合元数据中。

可使用以下三种方式之一来检索面向 WCF 的 IBM MQ 定制通道版本元数据：

- 使用 IBM MQ 实用程序 `dspmqr`。有关如何使用 `dspmqr` 的信息，请参阅：[dspmqr](#)
- 使用 Windows 资源管理器属性对话框：在 Windows 资源管理器中，右键单击 **IBM.XMS.WCF.dll** > **属性** > **版本**。
- 使用任何通道 FFST 或跟踪文件的头信息。有关 FFST 头信息的更多信息，请参阅：[第 303 页的『WCF XMS First Failure Support Technology \(FFST\)』](#)

WCF 提示和技巧

以下提示和技巧并非按重要性排序，并且可能会在发布新版本的文档时添加。如果这些主题与您正在做的工作相关，那么可能会节省您的时间。

外部化 WCF 服务主机的异常

对于使用 WCF 服务主机托管的服务，缺省情况下，不会外部化服务、WCF 内部或通道堆栈抛出的任何未处理的异常。要获悉这些异常，必须注册错误处理程序。

以下代码提供了定义错误处理程序服务行为的示例，该行为可作为服务的属性加以应用：

```
using System.ServiceModel.Dispatcher;
using System.Collections.ObjectModel;
....
```

```

public class ErrorHandlerBehaviorAttribute : Attribute, IServiceBehavior, IErrorHandler
{
    //
    // IServiceBehavior Interface
    //
    public void AddBindingParameters(ServiceDescription serviceDescription,
        ServiceHostBase serviceHostBase, CollectionServiceEndpoint endpoints,
        BindingParameterCollection bindingParameters)
    {
    }
    public void ApplyDispatchBehavior(ServiceDescription serviceDescription,
        ServiceHostBase serviceHostBase)
    {
        foreach (ChannelDispatcher channelDispatcher in serviceHostBase.ChannelDispatchers)
        {
            channelDispatcher.ErrorHandlers.Add(this);
        }
    }
    public void Validate(ServiceDescription serviceDescription, ServiceHostBase
serviceHostBase)
    {
    }
    //
    // IErrorHandler Interface
    //
    public bool HandleError(Exception e)
    {
        // Process the exception in the required way, in this case just outputting to the
console
        Console.Out.WriteLine(e);

        // Always return false to allow any other error handlers to run
        return false;
    }
    public void ProvideFault(Exception error, MessageVersion version, ref Message fault)
    {
    }
}
}

```

故障诊断 XMS .NET 问题

使用以下技巧可帮助您对使用 XMS 时遇到的问题进行故障诊断。

XMS 应用程序无法连接到队列管理器 (MQRC_NOT_AUTHORIZED)

XMS .NET 客户机的行为可能与 IBM MQ JMS 客户机的行为不同。因此，您可能会发现 XMS 应用程序无法连接到队列管理器，但 JMS 应用程序可以。

- 这个问题的最简单解决方案是尝试使用长度小于 12 个字符且在队列管理器的权限列表中已完全授权的用户标识。如果此解决方案不是很理想，那么另一种较为复杂的方法是使用安全出口。如果您对于此问题需要获得进一步的帮助，请联系 IBM 支持人员以获取帮助。
- 如果设置了连接工厂的 XMSC_USERID 属性，那么此属性必须与已登录用户的用户标识和密码匹配。如果未设置此属性，那么缺省情况下队列管理器将使用已登录用户的用户标识。
- IBM MQ 的用户认证是通过当前已登录用户的详细信息来完成的，而不是通过 XMSC.USERID 和 XMSC.PASSWORD 字段中提供的信息来完成。这旨在与 IBM MQ 保持一致。有关认证的更多信息，请参阅联机 IBM MQ 产品文档中的认证信息。

重定向到消息传递引擎的连接

当连接到 WebSphere Application Server 6.0.2 服务集成总线时，可以将所有连接从原始提供程序端点重定向到总线为该客户机连接选择的消息传递引擎。执行此操作时，始终将连接重定向到由主机名（而不是 IP 地址）指定的主机服务器。因此，如果无法解析主机名，那么您可能会遇到连接问题。

要成功连接到 WebSphere Application Server 6.0.2 服务集成总线，您可能需要在客户机主机上提供主机名与 IP 地址的映射。例如，可以在客户机主机上的本地主机表中指定该映射。

支持 telnet 类型的密码认证

XMS .NET 实时传输协议仅支持简单的 telnet 类型密码认证。XMS .NET 实时传输协议不支持“保护质量”。

设置 double 类型属性的值

在 Windows 64 位平台上，在设置或获取 double 类型属性的值时，如果值小于 Double.Epsilon，那么 SetDoubleProperty() 或 GetDoubleProperty() 方法可能无法正常运行。

例如，如果为 double 类型的属性设置值 4.9E-324，那么 Windows 64 位平台会将其视为 0.0。因此，在分布式消息传递环境中，如果 JMS 或其他应用程序在任何 UNIX 或 Windows 32 位机器上设置 double 属性的值，而 XMS .NET 在 64 位机器上运行，那么 GetDoubleProperty() 返回的值将为 0.0。这是 Microsoft .NET Framework 2.0 Framework 存在的已知问题。

对 IBM MQ for z/OS 问题进行故障诊断

IBM MQ for z/OS、CICS、Db2 和 IMS 产生可用于问题确定的诊断信息。

本节包含有关以下主题的信息：

- 当检测到问题时，队列管理器尝试的恢复操作。
- IBM MQ for z/OS 异常终止，以及发生异常终止时产生的信息。
- IBM MQ for z/OS 产生的诊断信息，以及其他有用信息源。

提供用于帮助进行问题确定和应用程序调试的信息的类型取决于遇到的错误类型以及子系统的设置方式。

请参阅以下子主题以获取有关 IBM MQ for z/OS 上的问题确定和诊断信息的更多信息。

- [第 179 页的『IBM MQ for z/OS 性能约束』](#)
- [第 181 页的『IBM MQ for z/OS 恢复操作』](#)
- [第 182 页的『IBM MQ for z/OS 异常终止』](#)
- [第 185 页的『在 IBM MQ for z/OS 上产生的诊断信息』](#)
- [第 186 页的『IBM MQ for z/OS 的其他问题确定信息来源』](#)
- [第 187 页的『针对 CICS 的诊断辅助』](#)
- [第 197 页的『针对 IMS 的诊断辅助』](#)
- [第 197 页的『针对 Db2 的诊断辅助』](#)
- [第 198 页的『IBM MQ for z/OS 转储』](#)
- [第 216 页的『处理 z/OS 的性能问题』](#)
- [第 221 页的『处理 z/OS 上的不正确输出』](#)

相关概念

[第 285 页的『使用错误日志』](#)

有各种错误日志可用于帮助进行问题确定和故障诊断。

[第 293 页的『First Failure Support Technology \(FFST\)』](#)

First Failure Support Technology (FFST) for IBM MQ 可提供有关事件的信息，在发生错误的情况下，这些信息有助于 IBM 支持人员对问题进行诊断。

相关任务

[第 5 页的『IBM MQ 故障诊断和支持』](#)

如果您的队列管理器网络或 IBM MQ 应用程序存在问题，那么可以使用本信息中描述的方法来帮助您诊断和解决问题。如果需要有关问题的帮助，可以通过 IBM 支持站点联系 IBM 支持人员。

[第 304 页的『使用跟踪』](#)

您可以使用不同类型的跟踪来帮助您确定问题并进行故障诊断。

IBM MQ for z/OS 性能约束

使用本主题来调查可能导致性能约束的 z/OS 资源。

在定制可以影响系统执行方式的 IBM MQ for z/OS 时，要制定许多决策。这些决策包括：

- 数据集的大小和放置
- 缓冲区的分配
- 队列在页集和耦合设施结构中的分布
- 允许任一时间访问队列管理器的任务数

日志缓冲池

日志缓冲区不足会导致应用程序等待直至日志缓冲区可用，这可能会影响 IBM MQ 性能。RMF 报告可能表明对包含日志数据集的卷执行大量 I/O。

有三个参数可用于对日志缓冲区进行调优。最重要的是 OUTBUFF。如果日志管理器统计信息 QJSTWTB 大于 0，请增加日志缓冲区的大小。此参数控制在将缓冲区写入到活动日志数据集之前要填充的缓冲区的数量（范围为 1 到 256）。持久消息的落实和失去同步点处理导致日志缓冲区写出到日志。因此，此参数作用不大，除非处理大型消息，并且落实或失去同步点消息的数量较低。这些参数在 CSQ6LOGP 宏中指定（请参阅使用 CSQ6LOGP 以获取详细信息），并且重要参数为：

OUTBUFF

此参数控制输出缓冲区的大小（大小在 40 KB 到 4000 KB 范围内）。

WRTHRS

此参数控制在将缓冲区写入到活动日志数据集之前要填充的缓冲区的数量（范围为 1 到 256）。

您还必须注意 CSQ6SYSYP 宏的 LOGLOAD 参数。此参数指定在检查点记录之间写入的日志记录数。范围为 200 到 16 000 000，但是大型系统的典型值为 500 000。如果值太小，那么您会接收到频繁检查点，这将耗费处理器时间并可导致其他磁盘 I/O。

缓冲池大小

每个页集都有关联的缓冲池。可以使用 `DEFINE BUFFPOOL` 命令指定缓冲池中的缓冲区的数量。

如果未正确指定缓冲池大小，可能会对 IBM MQ 性能产生负面影响。缓冲池越小，需要的物理 I/O 越频繁。RMF 可能表明对包含页集的卷执行大量 I/O。对于仅包含短期消息的缓冲池，缓冲区管理器统计信息 QPSTSLA、QPSTSOS 和 QPSTRIO 通常必须为零。对于其他缓冲池，QPSTSOS 和 QPSTSTLA 必须为零。

可用 DASD 上的数据集分布

DASD 上的页面数据集分布可能对 IBM MQ 的性能产生重大影响。

将日志数据集放在低使用率卷上，其中日志 n 和日志 $n+1$ 位于不同卷上。确保双日志放在 DASD 上的不同控制单元上，并且卷不在同一物理磁盘上。

页集上的队列分布

页集上的队列分布可能会影响性能。如果使用驻留在使用率高的页集上的特定队列的事务遇到响应时间长的情况，那么可能表明性能发生此更改。RMF 报告可能表明对包含受影响页集的卷执行大量 I/O。

您可以通过定义存储类 (STGCLASS) 对象、指定特定页集，然后在队列定义中定义 STGCLASS 参数，将队列分配到特定页集。理想做法是以此方式在不同页集上定义使用率高的队列。

耦合设施结构上的队列分布

耦合设施结构上的队列分布可能会影响性能。

队列共享组最多可以连接到 64 个耦合设施结构，其中一个结构必须是管理结构。您可以将剩余 63 个耦合设施结构用于 IBM MQ 数据，其中每个结构最多包含 512 个队列。如果需要多个耦合设施结构，请根据队列的功能跨多个结构分隔队列。

可以采取某些步骤，最大限度地提高效率：

- 删除不再需要的任何耦合设施结构。
- 将应用程序使用的所有队列放在同一耦合设施上，以提高应用程序处理效率。
- 如果工作对于性能特别敏感，请选择更快速的耦合设施结构。

假设您丢失耦合设施结构，那么将丢失其中存储的任何非持久消息。如果队列分散于各种耦合设施结构上，那么这些非持久消息的丢失可能会导致一致性问题。要使用持久消息，必须至少使用 CFLEVEL(3) 和 RECOVER(YES) 来定义耦合设施结构。

并发线程的限制

访问队列管理器的任务数也可影响性能，尤其是在有其他约束（如存储）或有许多任务访问少数几个队列的情况下。症状可能是对一个或多个页集执行大量 I/O，或者已知访问相同队列的任务的响应时间长。对于 TSO 和批处理，IBM MQ 中的线程数限制为 32767。

在 CICS 环境中，可以使用 CICS MAXTASK 来限制并行访问。

使用 IBM MQ 跟踪进行管理

虽然您有时可能必须使用特定跟踪，但是使用跟踪功能会对系统性能造成负面影响。

请考虑要将跟踪信息发送到的目标。使用内部跟踪表可节省 I/O，但是它对于产生大量数据的跟踪而言大小不足。

统计信息跟踪以一定的时间间隔收集信息。时间间隔由 CSQ6SYSP 宏的 STATIME 参数控制，如使用 CSQ6SYSP 中所述。当任务或通道结束时，将会产生记帐跟踪记录（可能在许多天之后）。

您可以按类、资源管理器标识 (RMID) 和检测设施标识 (IFCID) 限制跟踪，以减少收集的数据量。请参阅 [START TRACE](#) 以获取更多信息。

IBM MQ for z/OS 恢复操作

使用本主题来了解用户检测的和队列管理器检测的错误的某些恢复操作。

IBM MQ for z/OS 可以从由用户数据不正确所引起的程序检查进行恢复。系统会向调用者发出完成代码和原因码。这些代码记录在 [IBM MQ for z/OS 消息、完成代码和原因码](#) 中。

程序错误

程序错误可能与用户应用程序代码或 IBM MQ 代码相关联，并且分成两个类别：

- [用户检测的错误](#)
- [子系统检测的错误](#)

用户检测的错误

用户检测的错误由用户（或用户编写的应用程序）在服务请求结果并非预期（例如，非零完成代码）时检测。问题确定数据的收集无法自动执行，因为检测发生在 IBM MQ 功能完成之后。在 IBM MQ 用户参数跟踪功能激活的情况下重新运行应用程序可以提供分析问题所需的数据。来自此跟踪的输出会定向到通用跟踪功能 (GTF)。

可以使用操作员命令开启和关闭跟踪。请参阅第 318 页的『[在 z/OS 上使用跟踪进行问题确定](#)』以获取更多信息。

队列管理器检测的错误

队列管理器检测如下错误：

- 程序检查
- 数据集填充
- 内部一致性错误

IBM MQ 分析错误并采取以下操作：

- 如果问题由用户或应用程序错误导致（例如，使用的地址无效），那么错误通过完成代码和原因码传回到应用程序。
- 如果问题不是由用户或应用程序错误导致（例如，所有可用 DASD 都已使用，或者系统检测到内部不一致），那么如果可能，IBM MQ 将通过向应用程序发送完成代码和原因码进行恢复；如果此方法不可能，那么将通过停止应用程序进行恢复。
- 如果 IBM MQ 无法恢复，那么它将终止并返回特定原因码。通常采用 SVC 转储在该转储的系统诊断工作区 (SDWA) 和可变记录区域 (VRA) 部分中记录信息，并在 SYS1.LOGREC 中生成一个条目。

z/OS IBM MQ for z/OS 异常终止

在 WebSphere for z/OS 或其他 z/OS 系统中，可能会发生异常终止。使用本主题来了解 IBM MQ 系统异常终止代码以及如何调查 CICS、IMS 和 z/OS 中发生的异常终止。

IBM MQ for z/OS 使用两个系统异常终止完成代码 X'5C6' 和 X'6C6'。这些代码标识：

- 操作期间遇到的内部错误
- 用于问题确定的诊断信息
- 由错误中涉及的组件启动的操作

X'5C6'

X'5C6' 异常终止完成代码指示 IBM MQ 检测到内部错误，并已异常终止内部任务 (TCB) 或用户连接的任务。与 X'5C6' 异常终止完成代码关联的错误前面可能有 z/OS 系统代码或内部错误。

检查 X'5C6' 异常终止生成的诊断材料，以确定实际导致后续任务或子系统终止的错误的源。

X'6C6'

X'6C6' 异常终止完成代码指示 IBM MQ 检测到严重错误并已异常终止队列管理器。在发出 X'6C6' 时，IBM MQ 确定继续操作可能会导致丢失数据完整性。与 X'6C6' 异常终止完成代码关联的错误前面可能有 z/OS 系统错误、一个或多个 X'5C6' 异常终止完成代码或指示 IBM MQ 异常终止的错误消息 CSQV086E。

第 182 页的表 6 概括在发出这些异常终止完成代码时可用于 IBM MQ for z/OS 的操作和诊断信息。此信息的不同部分与不同错误情况相关。为特定错误产生的信息取决于特定问题。有关提供诊断信息的 z/OS 服务的更多信息，请参阅第 185 页的『在 IBM MQ for z/OS 上产生的诊断信息』。

| | X'5C6' | X'6C6' |
|------|---|---|
| 说明 | <ul style="list-style-type: none"> • 在 IBM MQ 正常操作期间出错 | <ul style="list-style-type: none"> • 严重错误；继续操作可能会危害数据完整性 |
| 系统操作 | <ul style="list-style-type: none"> • 内部 IBM MQ 任务已异常终止 • 连接的用户任务已异常终止 | <ul style="list-style-type: none"> • 整个 IBM MQ 子系统已异常终止 • 具有活动 IBM MQ 连接的用户任务可能异常终止并返回 X'6C6' 代码 • 已连接的联合地址空间的可能 MEMTERM（内存终止） |
| 诊断信息 | <ul style="list-style-type: none"> • SVC 转储 • SYS1.LOGREC 条目 • VRA 数据项 | <ul style="list-style-type: none"> • SYS1.LOGREC • VRA 数据项 |

表 6: 异常终止完成代码 (继续)

| | X'5C6' | X'6C6' |
|---------|---|---|
| 关联的原因码 | <ul style="list-style-type: none"> • IBM MQ 异常终止原因码 • 关联的 z/OS 系统代码 | <ul style="list-style-type: none"> • 子系统终止原因码 • 在 X'6C6' 异常终止之前的 z/OS 系统完成代码和 X'5C6' 代码 |
| 附带代码的位置 | <ul style="list-style-type: none"> • SVC 转储标题 • 消息 CSQW050I • SDWA 部分“出错时的通用寄存器”的寄存器 15 • SYS1.LOGREC 条目 • VRA 数据项 | <ul style="list-style-type: none"> • SYS1.LOGREC • VRA 数据项 • 消息 CSQV086E, 发送到 z/OS 系统操作员 |

相关概念

第 183 页的『在 IBM MQ for z/OS 上处理异常终止』

应用程序和其他 z/OS 系统可能会发生异常终止。使用本主题调查程序异常终止、批处理异常终止、CICS 事务异常终止和 IMS 事务异常终止。

第 184 页的『CICS、IMS 和 z/OS 异常终止』

使用本主题来调查 CICS、IMS 和 z/OS 产生的异常终止。

第 185 页的『在 IBM MQ for z/OS 上产生的诊断信息』

使用本主题来调查由 z/OS 产生的有助于问题确定的一些诊断信息，并且了解如何调查错误消息、转储、控制台日志、作业输出、症状字符串和队列输出。

第 198 页的『IBM MQ for z/OS 转储』

本主题可用于获取有关在问题确定过程中使用转储的信息。它描述您在观察由 IBM MQ for z/OS 地址空间生成的转储时应采取的步骤。

z/OS 在 IBM MQ for z/OS 上处理异常终止

应用程序和其他 z/OS 系统可能会发生异常终止。使用本主题调查程序异常终止、批处理异常终止、CICS 事务异常终止和 IMS 事务异常终止。

异常终止类型

程序异常终止可能是由于应用程序无法检查和响应来自 IBM MQ 的原因码所导致。例如，如果尚未收到消息，那么使用消息中为计算设置的字段可能导致 X'0C4' 或 X'0C7' 异常终止（CICS 中的 ASRA 异常终止）。

以下信息段指示程序异常终止：

- 控制台日志中来自 IBM MQ 的错误消息
- CICS 错误消息
- CICS 事务转储
- IMS 区域转储
- 用户终端或主终端上的 IMS 消息
- 批处理或 TSO 输出中的程序转储信息
- 批处理作业输出中的异常终止消息
- TSO 屏幕上的异常终止消息

如果您具有异常终止代码，请参阅以下手册之一以获取异常终止原因的说明：

- 对于 IBM MQ for z/OS 异常终止（异常终止代码 X'5C6' 和 X'6C6'），请参阅 [IBM MQ for z/OS 消息代码、完成代码和原因码](#)

- 对于批处理异常终止, [z/OS MVS System Codes](#) 手册
- 对于 CICS 异常终止, [CICS 消息](#)
- 对于 IMS 异常终止, [IMS 消息和代码](#)
- 对于 Db2 异常终止, [消息](#)
- Db2
- 对于 RRS 异常终止, [z/OS MVS System Messages , Volume 3](#)
- 对于 XES 异常终止, [z/OS MVS System Messages , Volume 10](#)

批处理异常终止

批处理异常终止导致在系统日志中显示包含有关寄存器内容的信息的错误消息。TSO 异常终止导致在 TSO 屏幕上产生包含类似信息的错误消息。如果针对步骤存在 SYSUDUMP DD 语句, 那么将采用 SYSUDUMP (请参阅第 198 页的『[IBM MQ for z/OS 转储](#)』)。

CICS 事务异常终止

CICS 事务异常终止记录在 CICS CSMT 日志中, 并在终端产生消息 (如果有)。CICS AICA 异常终止指示可能的循环。有关更多信息, 请参阅第 220 页的『[在 z/OS 上处理循环](#)』。如果遇到 CICS 异常终止, 那么使用 CEDF 和 CICS 跟踪可能会帮助查找问题的原因。请参阅 [CICS 故障诊断](#)(以前称为 *CICS Problem Determination Guide*) 以获取更多信息。

IMS 事务异常终止

IMS 事务异常终止记录在 IMS 主终端上, 并在终端产生错误消息 (如果有)。如果存在 IMS 异常终止, 请参阅 [IMS 故障诊断](#)。

CICS、IMS 和 z/OS 异常终止

使用本主题来调查 CICS、IMS 和 z/OS 产生的异常终止。

CICS 异常终止

CICS 异常终止消息将发送到终端 (如果应用程序连接到该终端) 或发送到 CSMT 日志。CICS 异常终止代码在 [CICS 消息和代码](#) 手册中进行了说明。

CICS 适配器发出以字母 Q 开头的异常终止原因码 (例如, QDCL)。这些代码记录在 [IBM MQ for z/OS 消息、完成代码和原因码](#) 中

IMS 异常终止

在下列其中一种情况下, IMS 应用程序可能会异常终止:

- 普通异常终止。
- IMS 伪异常终止, 带有因 ESAF 出口程序出错而产生的异常终止代码 (如 U3044)。
- 异常终止 3051 或 3047, 当 REO (区域错误选项) 已指定为“Q”或“A”, 并且 IMS 应用程序尝试引用不可运行的外部子系统时, 或者如果资源在创建线程时不可用。

系统向用户终端或作业输出以及 IMS 主终端发送 IMS 消息。异常终止可能附带区域转储。

z/OS 异常终止

在 IBM MQ 操作期间, 可能会发生异常终止, 并返回 z/OS 系统完成代码。如果您收到 z/OS 异常终止, 请参阅相应的 z/OS 出版物。

使用本主题来调查由 z/OS 产生的有助于问题确定的一些诊断信息，并且了解如何调查错误消息、转储、控制台日志、作业输出、症状字符串和队列输出。

IBM MQ for z/OS 功能恢复例程使用 z/OS 服务来提供诊断信息，以帮助进行问题确定。

以下 z/OS 服务提供诊断信息：

SVC 转储

IBM MQ 异常终止完成代码 X'5C6' 使用 z/OS SDUMP 服务来创建 SVC 转储。与这些转储相关联的内容和存储区域根据特定错误以及发生错误时的队列管理器状态而异。

SYS1.LOGREC

发生错误时，将使用 z/OS SETRP 服务在 SYS1.LOGREC 数据集中请求条目。SYS1.LOGREC 中还记录了以下信息：

- 子系统异常终止
- 在恢复例程中发生的辅助异常终止
- 来自恢复终止管理器的请求

可变记录区域 (VRA) 数据

数据项通过使用 z/OS VRA 定义的键添加到 SDWA 的 VRA。VRA 数据包括对于所有 IBM MQ for z/OS 异常终止完成代码通用的一系列诊断数据项。其他信息由调用组件恢复例程或由恢复终止管理器在初始错误处理期间提供。

IBM MQ for z/OS 提供唯一消息，这些消息与转储输出一起旨在提供充足数据来诊断问题，而不必尝试重现问题。这称为首次故障数据捕获。

错误消息

在检测到问题时，IBM MQ 产生错误消息。IBM MQ 诊断消息以前缀 CSQ 开头。IBM MQ 生成的每条错误消息都是唯一的；即，针对一个且仅一个错误生成该错误消息。有关错误的信息可在 [IBM MQ for z/OS 消息、完成代码和原因码](#) 中找到。

IBM MQ 模块的名称的前三个字符通常也是 CSQ。此例外情况是 C++ (IMQ) 和头文件 (CMQ) 的模块。第四个字符唯一标识组件。第 5 到 8 个字符在前四个字符标识的组内唯一。

确保您具有有关在安装时编写的程序的应用程序消息和代码的一些文档，并能查看 [IBM MQ for z/OS 消息、完成代码和原因码](#)

可能在有些情况下未产生任何消息，或者如果产生了消息，但其无法传送。在这些情况下，可能必须分析转储以将错误隔离到特定模块。有关转储的使用的更多信息，请参阅 [第 198 页的『IBM MQ for z/OS 转储』](#)。

转储

转储是有关问题的详细信息的重要来源。无论转储是由于异常终止还是用户请求所产生，通过它们都可查看执行转储时发生的情况的快照。第 198 页的『IBM MQ for z/OS 转储』包含有关使用转储来查找 IBM MQ 系统中的问题的指导。但是，由于它们仅提供快照，因此可能需要将其与覆盖更长时间段的其他信息源（如日志）结合使用。

对于处理 MQI 调用时发生的特定类型的错误，也会产生快速转储。转储写入到 CSQSNAP DD。

控制台日志和作业输出

您可以将控制台日志复制到永久数据集中，或者根据需要将其列显。如果您仅对特定事件感兴趣，那么可以选择要列显控制台日志的哪些部分。

作业输出包括从运行作业产生的输出，以及来自控制台的输出。可以将此输出复制到永久数据集中，或者根据需要将其列显。您可能需要收集所有关联作业的输出，例如 CICS、IMS 和 IBM MQ。

症状字符串

症状字符串以结构化格式显示重要诊断信息。产生症状字符串之后，将在以下一个或多个位置提供该症状字符串：

- 在 z/OS 系统控制台上
- 在 SYS1.LOGREC 中
- 在执行的任何转储中

第 186 页的图 12 显示症状字符串的示例。

```
PIDS/ 5655R3600 RIDS/CSQMAIN1 AB/S6C6 PRCS/0E30003
```

图 12: 样本症状字符串

症状字符串提供大量可用于搜索 IBM 软件支持数据库的关键字。如果您对其中一个可选搜索工具具有访问权，那么可以自行搜索数据库。如果您将问题报告给 IBM 支持中心，那么往往会要求您对症状字符串加上引号。

虽然症状字符串旨在为搜索数据库提供关键字，但它也可以提供有关发生错误时的状况的大量信息，并且可能建议明显的原因或有望解决问题的方面来开始您的调查。

队列信息

您可以通过使用操作和控制面板来显示有关队列状态的信息。或者，可以从 z/OS 控制台输入 DISPLAY QUEUE 和 DISPLAY QSTATUS 命令。

注：如果命令是从控制台发出，那么响应会复制到控制台日志，从而允许文档紧实地保存在一起。

相关概念

第 318 页的『[在 z/OS 上使用跟踪进行问题确定](#)』

有不同的跟踪选项可用于对 IBM MQ 进行问题确定。使用本主题来了解不同选项以及如何控制跟踪。

第 186 页的『[IBM MQ for z/OS 的其他问题确定信息来源](#)』

使用本主题来调查用于进行 IBM MQ for z/OS 问题确定的其他信息来源。

第 187 页的『[针对 CICS 的诊断辅助](#)』

您可以使用 CICS 诊断事务来显示有关队列管理器任务和 MQI 调用的信息。本主题可用于调查这些设施。

第 197 页的『[针对 IMS 的诊断辅助](#)』

本主题可用于调查 IMS 诊断设施。

第 197 页的『[针对 Db2 的诊断辅助](#)』

本主题可用于调查 Db2 诊断工具的引用。

z/OS

IBM MQ for z/OS 的其他问题确定信息来源

使用本主题来调查用于进行 IBM MQ for z/OS 问题确定的其他信息来源。

在解决 IBM MQ for z/OS 的问题时，您可能会发现以下文档项有用。

- [您自己的文档](#)
- [您使用的产品的文档](#)
- [源列表和链接编辑映射](#)
- [更改日志](#)
- [系统配置图表](#)
- [来自 DISPLAY CONN 命令的信息](#)

您自己的文档

您自己的文档是贵组织针对系统和应用程序应该做些什么以及应该如何做产生的信息集合。您需要的此信息量取决于您对相关系统或应用程序的熟悉程度，并且可能包括：

- 程序描述或功能规范
- 系统中活动流的流程图或其他描述
- 程序的变更历史记录
- 安装的变更历史记录
- 显示平均输入、输出和响应时间的统计和监控概要文件

您使用的产品的文档

您使用的产品的文档是 IBM MQ 库以及用于应用程序的任何其他产品的库中的 InfoCenter。

请确保您参考的任何文档的级别与您使用的系统的级别相匹配。使用过时信息或有关尚未安装的产品级别的信息，往往会出现问题。

源列表和链接编辑映射

包括在对文档集进行安装时编写的任何应用程序的源列表。（它们通常可以是最大的单一文档元素。）确保将链接编辑器的相关输出包含在源列表中，以避免浪费时间尝试通过带有过时链接映射的装入模块查找方法。务必在列表的开头包含 JCL，以显示所使用的库和装入模块放置所在的装入库。

更改日志

更改日志中的信息可以表明在可能已导致应用程序问题的数据处理环境中进行的更改。要充分利用更改日志，请包含有关对操作过程进行的硬件更改、系统软件（如 z/OS 和 IBM MQ）更改、应用程序更改和任何修改的数据。

系统配置图表

系统配置图表显示正在运行的系统、其运行位置以及系统如何相互连接。它们还显示哪些 IBM MQ、CICS 或 IMS 系统是测试系统，哪些是生产系统。

来自 DISPLAY CONN 命令的信息

DISPLAY CONN 命令提供有关哪些应用程序连接到队列管理器的信息，以及用于帮助诊断具有长时间运行的工作单元的应用程序的信息。您应定期收集此信息并检查其是否有任何长时间运行的工作单元，以及显示有关该连接的详细信息。

针对 CICS 的诊断辅助

您可以使用 CICS 诊断事务来显示有关队列管理器任务和 MQI 调用的信息。本主题可用于调查这些设施。

您可以使用 CKQC 事务（CICS 适配器控制面板）来显示有关队列管理器任务及其所处状态（例如，GET WAIT）的信息。请参阅[管理 IBM MQ for z/OS](#)，以获取有关 CKQC 的更多信息。

应用程序开发环境与任何其他 CICS 应用程序的开发环境相同，因此，您可以使用此环境中的任何常用工具来开发 IBM MQ 应用程序。尤其是，CICS 执行诊断设施 (CEDF) 会为每个 MQI 调用的 CICS 适配器出入口设置陷阱，并为所有 CICS API 服务调用设置陷阱。此设施生成的输出示例在 [第 187 页的『MQI 调用的 CEDF 输出示例』](#) 中提供。

CICS 适配器还会将跟踪条目写入 CICS 跟踪。在 [第 325 页的『CICS 适配器跟踪条目』](#) 中描述了这些条目。

可从 CICS 区域获取其他跟踪和转储数据。在 [CICS Problem Determination Guide](#) 中描述了这些条目。

MQI 调用的 CEDF 输出示例

使用 IBM MQ 时由 CICS 执行诊断工具 (CEDF) 生成的输出的示例。

这些示例以十六进制和字符格式显示在进入和退出以下 MQI 调用时生成的数据。其他 MQI 调用生成类似的数据。

相关参考

函数调用

MQOPEN 调用的示例 CEDF 输出

此调用的参数如下所示:

| 表 7: MQOPEN 调用的参数 | |
|-------------------|-------|
| 参数 | 描述 |
| ARG 000 | 连接句柄 |
| ARG 001 | 对象描述符 |
| ARG 002 | 选项 |
| ARG 003 | 对象句柄 |
| ARG 004 | 完成代码 |
| ARG 005 | 原因码 |

```

STATUS: ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'000000000000000100000000200004044') AT X'05ECAFD8'
001: ARG 001 (X'D6C4404000000001000000001C3C5C4C6') AT X'00144910'
001: ARG 002 (X'00000072000000000000000000000000') AT X'001445E8'
001: ARG 003 (X'00000000000000720000000000000000') AT X'001445E4'
001: ARG 004 (X'00000000000000000000000000000000') AT X'001445EC'
001: ARG 005 (X'00000000000000000000000000000000') AT X'001445F0'
    
```

图 13: 针对 MQOPEN 调用 (十六进制) 的条目的示例 CEDF 输出

```

STATUS: COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'000000000000000100000000200004044') AT X'05ECAFD8'
001: ARG 001 (X'D6C4404000000001000000001C3C5C4C6') AT X'00144910'
001: ARG 002 (X'00000072000000000000000000000000') AT X'001445E8'
001: ARG 003 (X'00000001000000720000000000000000') AT X'001445E4'
001: ARG 004 (X'00000000000000000000000000000000') AT X'001445EC'
001: ARG 005 (X'00000000000000000000000000000000') AT X'001445F0'
    
```

图 14: 从 MQOPEN 调用退出时的示例 CEDF 输出 (十六进制)

```

STATUS: ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('OD .....CEDF')
001: ARG 002 ('.....')
001: ARG 003 ('.....')
001: ARG 004 ('.....')
001: ARG 005 ('.....')
    
```

图 15: 针对 MQOPEN 调用 (字符) 的条目的示例 CEDF 输出

```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('OD .....CEDF')
001: ARG 002 ('.....')
001: ARG 003 ('.....')
001: ARG 004 ('.....')
001: ARG 005 ('.....')

```

图 16: 从 MQOPEN 调用 (字符) 退出时的示例 CEDF 输出

相关参考

[MQOPEN-打开对象](#)

[MQCLOSE 调用的示例 CEDF 输出](#)

此调用的参数为:

表 8: MQCLOSE 调用的参数

| 参数 | 描述 |
|---------|------|
| ARG 000 | 连接句柄 |
| ARG 001 | 对象句柄 |
| ARG 002 | 选项 |
| ARG 003 | 完成代码 |
| ARG 004 | 原因码 |

```

STATUS:  ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'000000000000000010000007200000000')           AT X'001445E0'
001: ARG 001 (X'000000010000000720000000000000000')           AT X'001445E4'
001: ARG 002 (X'00000000000000010000000200004044')           AT X'05ECAFD8'
001: ARG 003 (X'000000000000000000000000800000008')           AT X'001445EC'
001: ARG 004 (X'000000000000000080000000800000060')           AT X'001445F0'

```

图 17: MQCLOSE 调用 (十六进制) 的条目的示例 CEDF 输出

```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'000000000000000000000007200000000')           AT X'001445E0'
001: ARG 001 (X'000000000000000072000000000000000')           AT X'001445E4'
001: ARG 002 (X'00000000000000010000000200004044')           AT X'05ECAFD8'
001: ARG 003 (X'000000000000000000000000800000008')           AT X'001445EC'
001: ARG 004 (X'000000000000000080000000800000060')           AT X'001445F0'

```

图 18: 退出 MQCLOSE 调用时的 CEDF 输出示例 (十六进制)

```

STATUS: ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('.....')
001: ARG 003 ('.....')
001: ARG 004 ('.....')

```

图 19: 针对 MQCLOSE 调用 (字符) 的条目的示例 CEDF 输出

```

STATUS: COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('.....')
001: ARG 003 ('.....')
001: ARG 004 ('.....')

```

图 20: 从 MQCLOSE 调用 (字符) 退出时的示例 CEDF 输出

相关参考

[MQCLOSE-关闭对象](#)

[MQPUT 调用的示例 CEDF 输出](#)

此调用的参数为:

表 9: MQPUT 调用的参数

| 参数 | 描述 |
|---------|---------|
| ARG 000 | 连接句柄 |
| ARG 001 | 对象句柄 |
| ARG 002 | 消息描述符 |
| ARG 003 | 放置消息 选项 |
| ARG 004 | 缓冲区长度 |
| ARG 005 | 消息数据 |
| ARG 006 | 完成代码 |
| ARG 007 | 原因码 |

```

STATUS: ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'00000000000000010000007200000000') AT X'001445E0'
001: ARG 001 (X'00000001000000720000000000000000') AT X'001445E4'
001: ARG 002 (X'D4C44040000000010000000000000008') AT X'001449B8'
001: ARG 003 (X'D7D4D640000000010000002400000000') AT X'00144B48'
001: ARG 004 (X'0000000800000000000000000000004000') AT X'001445F4'
001: ARG 005 (X'5C5CC8C5D3D3D640E6D6D9D3C45C5C5C') AT X'00144BF8'
001: ARG 006 (X'00000000000000000000000000000000') AT X'001445EC'
001: ARG 007 (X'00000000000000080000000000000000') AT X'001445F0'

```

图 21: MQPUT 调用条目的 CEDF 输出示例 (十六进制)

```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'00000000000000010000007200000000') AT X'001445E0'
001: ARG 001 (X'00000000100000072000000000000000') AT X'001445E4'
001: ARG 002 (X'D4C4404000000001000000000000008') AT X'001449B8'
001: ARG 003 (X'D7D4D640000000010000002400000000') AT X'00144B48'
001: ARG 004 (X'000000080000000000000000000040000') AT X'001445F4'
001: ARG 005 (X'5C5CC8C5D3D3D640E6D6D9D3C45C5C5C') AT X'00144BF8'
001: ARG 006 (X'000000000000000000000000800000000') AT X'001445EC'
001: ARG 007 (X'0000000000000000000000000000000') AT X'001445F0'

```

图 22: 退出 MQPUT 调用时的 CEDF 输出示例 (十六进制)

```

STATUS:  ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('MD .....')
001: ARG 003 ('PMO .....')
001: ARG 004 ('.....')
001: ARG 005 ('**HELLO WORLD**')
001: ARG 006 ('.....')
001: ARG 007 ('.....')

```

图 23: 针对 MQPUT 调用 (字符) 的条目的示例 CEDF 输出

```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('MD .....')
001: ARG 003 ('PMO .....')
001: ARG 004 ('.....')
001: ARG 005 ('**HELLO WORLD**')
001: ARG 006 ('.....')
001: ARG 007 ('.....')

```

图 24: 退出 MQPUT 调用时的示例 CEDF 输出 (字符)

相关参考

[MQPUT-放置消息](#)

[MQPUT1 调用的示例 CEDF 输出](#)

此调用的参数为:

| 表 10: MQPUT1 调用的参数 | |
|--------------------|---------|
| 参数 | 描述 |
| ARG 000 | 连接句柄 |
| ARG 001 | 对象描述符 |
| ARG 002 | 消息描述符 |
| ARG 003 | 放置消息 选项 |
| ARG 004 | 缓冲区长度 |
| ARG 005 | 消息数据 |

表 10: MQPUT1 调用的参数 (继续)

| 参数 | 描述 |
|---------|------|
| ARG 006 | 完成代码 |
| ARG 007 | 原因码 |

```

STATUS: ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'000000000000000000000000007200000000') AT X'001445E0'
001: ARG 001 (X'D6C44040000000001000000001C3C5C4C6') AT X'00144910'
001: ARG 002 (X'D4C44040000000001000000000000008') AT X'001449B8'
001: ARG 003 (X'D7D4D64000000000100000002400000000') AT X'00144B48'
001: ARG 004 (X'00000008000000000800000006000040000') AT X'001445F4'
001: ARG 005 (X'5C5CC8C5D3D3D640E6D6D9D3C45C5C5C') AT X'00144BF8'
001: ARG 006 (X'00000000000000000000000800000008') AT X'001445EC'
001: ARG 007 (X'000000000000000008000000080000060') AT X'001445F0'
    
```

图 25: MQPUT1 调用的条目的示例 CEDF 输出 (十六进制)

```

STATUS: COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'000000000000000000000000007200000000') AT X'001445E0'
001: ARG 001 (X'D6C44040000000001000000001C3C5C4C6') AT X'00144910'
001: ARG 002 (X'D4C44040000000001000000000000008') AT X'001449B8'
001: ARG 003 (X'D7D4D64000000000100000002400000000') AT X'00144B48'
001: ARG 004 (X'00000008000000000800000006000040000') AT X'001445F4'
001: ARG 005 (X'5C5CC8C5D3D3D640E6D6D9D3C45C5C5C') AT X'00144BF8'
001: ARG 006 (X'00000000000000000000000800000008') AT X'001445EC'
001: ARG 007 (X'000000000000000008000000080000060') AT X'001445F0'
    
```

图 26: 退出 MQPUT1 调用时的示例 CEDF 输出 (十六进制)

```

STATUS: ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('OD .....CEDF')
001: ARG 002 ('MD .....')
001: ARG 003 ('PMO .....')
001: ARG 004 ('.....')
001: ARG 005 ('**HELLO WORLD**')
001: ARG 006 ('.....')
001: ARG 007 ('.....')
    
```

图 27: 针对 MQPUT1 调用的条目的示例 CEDF 输出 (字符)


```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('OD .....CEDF')
001: ARG 002 ('MD .....')
001: ARG 003 ('PMO .....')
001: ARG 004 ('.....-.....')
001: ARG 005 ('**HELLO WORLD**')
001: ARG 006 ('.....')
001: ARG 007 ('.....-')

```

图 28: 退出 MQPUT1 调用时的示例 CEDF 输出 (字符)

相关参考

[MQPUT1 -放置一条消息](#)

MQGET 调用的示例 CEDF 输出

此调用的参数为:

| 表 11: MQGET 调用的参数 | |
|-------------------|--------|
| 参数 | 描述 |
| ARG 000 | 连接句柄 |
| ARG 001 | 对象句柄 |
| ARG 002 | 消息描述符 |
| ARG 003 | 获取消息选项 |
| ARG 004 | 缓冲区长度 |
| ARG 005 | 消息缓冲区 |
| ARG 006 | 消息长度 |
| ARG 007 | 完成代码 |
| ARG 008 | 原因码 |

```

STATUS:  ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'00000000000000010000007200000000')           AT X'001445E0'
001: ARG 001 (X'000000010000007200000000000000')           AT X'001445E4'
001: ARG 002 (X'D4C440400000000100000000000000')           AT X'001449B8'
001: ARG 003 (X'C7D4D6400000000100004044FFFFFF')           AT X'00144B00'
001: ARG 004 (X'0000008000000000000000000040000')           AT X'001445F4'
001: ARG 005 (X'000000000000000000000000000000')           AT X'00144C00'
001: ARG 006 (X'000000000000000000400000000000')           AT X'001445F8'
001: ARG 007 (X'000000000000000000000080000000')           AT X'001445EC'
001: ARG 008 (X'0000000000000080000000000000')           AT X'001445F0'

```

图 29: MQGET 调用的条目的 CEDF 输出示例 (十六进制)

```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'00000000000000010000007200000000')      AT X'001445E0'
001: ARG 001 (X'00000000100000007200000000000000')      AT X'001445E4'
001: ARG 002 (X'D4C4404000000001000000000000008')      AT X'001449B8'
001: ARG 003 (X'C7D4D6400000000100004044FFFFFF')        AT X'00144B00'
001: ARG 004 (X'000000080000000080000000000040000')    AT X'001445F4'
001: ARG 005 (X'5C5CC8C5D3D3D640E6D6D9D3C45C5C5')      AT X'00144C00'
001: ARG 006 (X'00000008000000000000400000000000')    AT X'001445F8'
001: ARG 007 (X'000000000000000000000000800000008')    AT X'001445EC'
001: ARG 008 (X'00000000000000008000000080000000')    AT X'001445F0'

```

图 30: 退出 MQGET 调用时的 CEDF 输出示例 (十六进制)

```

STATUS:  ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('MD .....')
001: ARG 003 ('GMO .....')
001: ARG 004 ('.....')
001: ARG 005 ('.....')
001: ARG 006 ('.....')
001: ARG 007 ('.....')
001: ARG 008 ('.....')

```

图 31: 针对 MQGET 调用 (字符) 的条目的示例 CEDF 输出

```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('MD .....')
001: ARG 003 ('GMO .....')
001: ARG 004 ('.....')
001: ARG 005 ('**HELLO WORLD**')
001: ARG 006 ('.....')
001: ARG 007 ('.....')
001: ARG 008 ('.....')

```

图 32: 从 MQGET 调用 (字符) 退出时的示例 CEDF 输出

相关参考

[MQGET-获取消息](#)

[MQINQ 调用的示例 CEDF 输出](#)

此调用的参数为:

| 参数 | 描述 |
|---------|----------|
| ARG 000 | 连接句柄 |
| ARG 001 | 对象句柄 |
| ARG 002 | 选择器计数 |
| ARG 003 | 属性选择器的数组 |
| ARG 004 | 整数属性的计数 |

表 12: MQINQ 调用的参数 (继续)

| 参数 | 描述 |
|---------|------------|
| ARG 005 | 整数属性 |
| ARG 006 | 字符属性缓冲区的长度 |
| ARG 007 | 字符属性 |
| ARG 008 | 完成代码 |
| ARG 009 | 原因码 |

```

STATUS: ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'000000000000000010000000200004044') AT X'05ECAFCF'
001: ARG 001 (X'00000001000000720000000000000000') AT X'001445E4'
001: ARG 002 (X'000000020000404485ECA00885ECA220') AT X'05ECAFD4'
001: ARG 003 (X'0000000D0000000C0000000000000000') AT X'00144C08'
001: ARG 004 (X'000000020000404485ECA00885ECA220') AT X'05ECAFD4'
001: ARG 005 (X'00000000000000000000000000000000') AT X'00144C10'
001: ARG 006 (X'000000000000000010000000200004044') AT X'05ECAFCF'
001: ARG 007 (X'00000000000000000000000000000000') AT X'00144C18'
001: ARG 008 (X'00000000000000000000000000800000008') AT X'001445EC'
001: ARG 009 (X'00000000000000000000000000800040000') AT X'001445F0'
    
```

图 33: MQINQ 调用的条目上的示例 CEDF 输出 (十六进制)

```

STATUS: COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'000000000000000010000000200004044') AT X'05ECAFCF'
001: ARG 001 (X'00000001000000720000000000000000') AT X'001445E4'
001: ARG 002 (X'000000020000404485ECA00885ECA220') AT X'05ECAFD4'
001: ARG 003 (X'0000000D0000000C0040000000000000') AT X'00144C08'
001: ARG 004 (X'000000020000404485ECA00885ECA220') AT X'05ECAFD4'
001: ARG 005 (X'00400000000000000000000000000000') AT X'00144C10'
001: ARG 006 (X'000000000000000010000000200004044') AT X'05ECAFCF'
001: ARG 007 (X'00000000000000000000000000000000') AT X'00144C18'
001: ARG 008 (X'00000000000000000000000000800000008') AT X'001445EC'
001: ARG 009 (X'00000000000000000000000000800040000') AT X'001445F0'
    
```

图 34: 从 MQINQ 调用退出时的示例 CEDF 输出 (十六进制)

```

STATUS: ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('.....e...e.s.')
001: ARG 003 ('.....')
001: ARG 004 ('.....e...e.s.')
001: ARG 005 ('.....')
001: ARG 006 ('.....')
001: ARG 007 ('.....')
001: ARG 008 ('.....')
001: ARG 009 ('.....')
    
```

图 35: 针对 MQINQ 调用 (字符) 的条目的示例 CEDF 输出

```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('.....e...e.s.')
001: ARG 003 ('.....')
001: ARG 004 ('.....e...e.s.')
001: ARG 005 ('.....')
001: ARG 006 ('.....')
001: ARG 007 ('.....')
001: ARG 008 ('.....')
001: ARG 009 ('.....')

```

图 36: 从 MQINQ 调用 (字符) 退出时的示例 CEDF 输出

相关参考

[MQINQ-查询对象属性](#)

MQSET 调用的示例 CEDF 输出

此调用的参数为:

表 13: MQSET 调用的参数

| 参数 | 描述 |
|---------|------------|
| ARG 000 | 连接句柄 |
| ARG 001 | 对象句柄 |
| ARG 002 | 选择器计数 |
| ARG 003 | 属性选择器的数组 |
| ARG 004 | 整数属性的计数 |
| ARG 005 | 整数属性 |
| ARG 006 | 字符属性缓冲区的长度 |
| ARG 007 | 字符属性 |
| ARG 008 | 完成代码 |
| ARG 009 | 原因码 |

```

STATUS:  ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'0000000000000000100000007200000000')           AT X'001445E0'
001: ARG 001 (X'00000001000000720000000000000000')           AT X'001445E4'
001: ARG 002 (X'000000010000000020000404485ECA008')           AT X'05ECAFD8'
001: ARG 003 (X'000000180000007DF000000000000000')           AT X'00144C08'
001: ARG 004 (X'000000010000000020000404485ECA008')           AT X'05ECAFD8'
001: ARG 005 (X'00000000000000000000000000000000')           AT X'00144C10'
001: ARG 006 (X'00000000000000010000000200004044')           AT X'05ECAFD8'
001: ARG 007 (X'00000000000000000000000000000000')           AT X'00144C18'
001: ARG 008 (X'00000000000000000000000080000000')           AT X'001445EC'
001: ARG 009 (X'00000000000000008000000080000060')           AT X'001445F0'

```

图 37: MQSET 调用的条目上的 CEDF 输出示例 (十六进制)

```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'00000000000000010000007200000000')      AT X'001445E0'
001: ARG 001 (X'00000000100000007200000000000000')      AT X'001445E4'
001: ARG 002 (X'000000001000000020000404485ECA008')      AT X'05ECAFD8'
001: ARG 003 (X'00000018000007DF000000000000000')      AT X'00144C08'
001: ARG 004 (X'00000001000000020000404485ECA008')      AT X'05ECAFD8'
001: ARG 005 (X'00000000000000000000000000000000')      AT X'00144C10'
001: ARG 006 (X'000000000000000010000000200004044')      AT X'05ECAFD8'
001: ARG 007 (X'00000000000000000000000000000000')      AT X'00144C18'
001: ARG 008 (X'000000000000000000000000800000008')      AT X'001445EC'
001: ARG 009 (X'000000000000000080000000800000060')      AT X'001445F0'

```

图 38: 退出 MQSET 调用时的 CEDF 输出示例 (十六进制)

```

STATUS:  ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('.....e..')
001: ARG 003 ('.....')
001: ARG 004 ('.....e..')
001: ARG 005 ('.....')
001: ARG 006 ('.....')
001: ARG 007 ('.....')
001: ARG 008 ('.....')
001: ARG 009 ('.....-')

```

图 39: 针对 MQSET 调用 (字符) 的条目的示例 CEDF 输出

```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('.....e..')
001: ARG 003 ('.....')
001: ARG 004 ('.....e..')
001: ARG 005 ('.....')
001: ARG 006 ('.....')
001: ARG 007 ('.....')
001: ARG 008 ('.....')
001: ARG 009 ('.....-')

```

图 40: 从 MQSET 调用 (字符) 退出时的示例 CEDF 输出

相关参考

[MQSET-设置对象属性](#)

z/OS 针对 IMS 的诊断辅助

本主题可用于调查 IMS 诊断设施。

应用程序开发环境与任何其他 IMS 应用程序的开发环境相同，因此，该环境中的任何常用工具均可用于开发 IBM MQ 应用程序。

可从 IMS 区域获取跟踪和转储数据。在 *IMS/ESA Diagnosis Guide and Reference* 手册中描述了这些条目。

z/OS 针对 Db2 的诊断辅助

本主题可用于调查 Db2 诊断工具的引用。

请参阅以下手册以获取诊断 Db2 问题的帮助：

- *Db2 for z/OS Diagnosis Guide and Reference*
- *Db2 消息和代码*

z/OS IBM MQ for z/OS 转储

本主题可用于获取有关在问题确定过程中使用转储的信息。它描述您在观察由 IBM MQ for z/OS 地址空间生成的转储时应采取的步骤。

如何将转储用于问题确定

解决 IBM MQ for z/OS 系统问题时，可以通过以下两种方法来使用转储：

- 检查 IBM MQ 处理来自应用程序的请求的方式。
要执行此操作，通常需要分析整个转储，包括控制块和内部跟踪。
- 在 IBM 支持中心人员的指导下，确定 IBM MQ for z/OS 本身的问题。

使用以下主题中的指示信息来获取和处理转储：

- [第 198 页的『使用 IBM MQ for z/OS 获取转储』](#)
- [第 199 页的『使用 z/OS DUMP 命令』](#)
- [第 200 页的『使用 IBM MQ for z/OS 转储显示面板处理转储』](#)
- [第 204 页的『使用行式 IPCS 处理 IBM MQ for z/OS 转储』](#)
- [第 211 页的『以批处理方式使用 IPCS 处理 IBM MQ for z/OS 转储』](#)

转储标题可提供有关异常终止的充足信息以及解决问题的原因码。您可以在控制台日志中或者通过使用 z/OS 命令 DISPLAY DUMP, TITLE 来查看转储标题。在 [第 212 页的『在 z/OS 上分析转储和解释转储标题』](#) 中对转储标题格式进行了说明。有关 IBM MQ for z/OS 异常终止代码的信息，请参阅 [第 182 页的『IBM MQ for z/OS 异常终止』](#)，在 IBM MQ for z/OS 消息代码、完成代码和原因码中记录了异常终止原因码。

如果转储标题中有关问题的信息不足，请格式化转储以显示其中包含的其他信息。

请参阅以下主题以获取有关不同类型的转储的信息：

- [第 214 页的『z/OS 上的 SYSUDUMP 信息』](#)
- [第 214 页的『z/OS 上的快速转储』](#)
- [第 215 页的『z/OS 上的 SYS1.LOGREC 信息』](#)
- [第 215 页的『z/OS 上的 SVC 转储』](#)

相关概念

[第 318 页的『在 z/OS 上使用跟踪进行问题确定』](#)
有不同的跟踪选项可用于对 IBM MQ 进行问题确定。使用本主题来了解不同选项以及如何控制跟踪。

[第 182 页的『IBM MQ for z/OS 异常终止』](#)
在 WebSphere for z/OS 或其他 z/OS 系统中，可能会发生异常终止。使用本主题来了解 IBM MQ 系统异常终止代码以及如何调查 CICS、IMS 和 z/OS 中发生的异常终止。

[第 185 页的『在 IBM MQ for z/OS 上产生的诊断信息』](#)
使用本主题来调查由 z/OS 产生的有助于问题确定的一些诊断信息，并且了解如何调查错误消息、转储、控制台日志、作业输出、症状字符串和队列输出。

z/OS 使用 IBM MQ for z/OS 获取转储

本主题可用于了解用于 IBM MQ for z/OS 问题确定的不同转储类型。

下表显示了有关用于 IBM MQ for z/OS 的转储类型及其启动方式的信息。其中还显示了转储的格式化方式：

| 转储类型 | 数据集 | 输出类型 | 格式化方式 | 原因: |
|----------|-----------------------------|------|--------------------------------|---|
| SVC | 由系统定义 | 机器可读 | 与 IBM MQ for z/OS 动词出口相连的 IPCS | z/OS 或 IBM MQ for z/OS 功能恢复例程检测到错误, 或者操作员输入了 z/OS DUMP 命令 |
| SYSUDUMP | 由 JCL 定义 (SYSOUT=A) | 已格式化 | 正常情况下 SYSOUT=A | 异常终止状况 (仅当针对此步骤存在 SYSUDUMP DD 语句时才采用) |
| 快速 | 由 JCL CSQSNAP 定义 (SYSOUT=A) | 已格式化 | 正常情况下 SYSOUT=A | 向适配器报告意外的 MQI 调用错误, 或者收到来自通道启动程序的 FFST 信息 |
| 独立 | 由安装定义 (磁带或磁盘) | 机器可读 | 与 IBM MQ for z/OS 动词出口相连的 IPCS | 独立转储程序的操作员 IPL |

IBM MQ for z/OS 恢复例程针对大部分 X'5C6' 异常终止都会请求 SVC 转储。在第 215 页的『z/OS 上的 SVC 转储』中列出了异常。由 IBM MQ for z/OS 发出的 SVC 转储是问题诊断信息的主要来源。

如果转储是由 IBM MQ 子系统启动的, 那么会将转储信息放入称为摘要部分的区域。这包含可供转储格式化程序用于识别关键组件的信息。

有关 SVC 转储的更多信息, 请参阅 *z/OS MVS Diagnosis: Tools and Service Aids* 手册。

z/OS 使用 z/OS DUMP 命令

为解决问题, IBM 可能要求您创建队列管理器地址空间、通道启动程序地址空间或耦合设施结构的转储文件。本主题可用于了解创建这些转储文件的命令。

可能要求您为以下任意项或多个项创建转储文件以便 IBM 解决问题:

- 主 IBM MQ 地址空间
- 通道启动程序地址空间
- 耦合设施应用程序结构
- 队列共享组的耦合设施管理结构

第 199 页的图 41 到第 200 页的图 45 显示了执行此操作的 z/OS 命令示例, 假定子系统名称为 CSQ1。

```

DUMP COMM=(MQ QUEUE MANAGER DUMP)
*01 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 01, JOBNAME=(CSQ1MSTR, BATCH), CONT
*02 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
IEE600I REPLY TO 01 IS;JOBNAME=CSQ1MSTR,CONT
R 02, SDATA=(CSA, RGN, PSA, SQA, LSQA, TRT, SUM), END
IEE600I REPLY TO 02 IS;SDATA=(CSA, RGN, PSA, SQA, LSQA, TRT, SUM), END
IEA794I SVC DUMP HAS CAPTURED: 869
DUMPID=001 REQUESTED BY JOB (*MASTER*)
DUMP TITLE=MQ QUEUE MANAGER MAIN DUMP

```

图 41: 转储 IBM MQ 队列管理器和应用程序地址空间

```

DUMP COMM=(MQ QUEUE MANAGER DUMP)
*01 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 01, JOBNAME=(CSQ1MSTR), CONT
*02 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
IEE600I REPLY TO 01 IS;JOBNAME=CSQ1MSTR,CONT
R 02, SDATA=(CSA, RGN, PSA, SQA, LSQA, TRT, SUM), END
IEE600I REPLY TO 02 IS;SDATA=(CSA, RGN, PSA, SQA, LSQA, TRT, SUM), END
IEA794I SVC DUMP HAS CAPTURED: 869
DUMPID=001 REQUESTED BY JOB (*MASTER*)
DUMP TITLE=MQ QUEUE MANAGER DUMP

```

图 42: 转储 IBM MQ 队列管理器地址空间

```

DUMP COMM=(MQ CHIN DUMP)
*01 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 01, JOBNAME=CSQ1CHIN, CONT
*02 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
IEE600I REPLY TO 01 IS;JOBNAME=CSQ1CHIN,CONT
R 02, SDATA=(CSA, RGN, PSA, SQA, LSQA, TRT, SUM), CONT
*03 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
IEE600I REPLY TO 02 IS;SDATA=(CSA, RGN, PSA, SQA, LSQA, TRT, SUM), CONT
R 03, DSPNAME=('CSQ1CHIN'.CSQXTRDS), END
IEE600I REPLY TO 03 IS;DSPNAME='CSQ1CHIN'.CSQXTRDS,END
IEA794I SVC DUMP HAS CAPTURED: 869
DUMPID=001 REQUESTED BY JOB (*MASTER*)
DUMP TITLE=MQ CHIN DUMP

```

图 43: 转储通道启动程序地址空间

```

DUMP COMM=(MQ MSTR & CHIN DUMP)
*01 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 01, JOBNAME=(CSQ1MSTR, CSQ1CHIN), CONT
*02 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
IEE600I REPLY TO 01 IS;JOBNAME=(CSQ1MSTR, CSQ1CHIN), CONT
R 02, SDATA=(CSA, RGN, PSA, SQA, LSQA, TRT, SUM), CONT
*03 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
IEE600I REPLY TO 02 IS;SDATA=(CSA, RGN, PSA, SQA, LSQA, TRT, SUM), CONT
R 03, DSPNAME=('CSQ1CHIN'.CSQXTRDS), END
IEE600I REPLY TO 03 IS;DSPNAME=('CSQ1CHIN'.CSQXTRDS),END
IEA794I SVC DUMP HAS CAPTURED: 869
DUMPID=001 REQUESTED BY JOB (*MASTER*)
DUMP TITLE=MQ MSTR & CHIN DUMP

```

图 44: 转储 IBM MQ 队列管理器和通道启动程序地址空间

```

DUMP COMM=('MQ APPLICATION STRUCTURE 1 DUMP')
01 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 01, STRLIST=(STRNAME=QSG1APPLICATION1, (LISTNUM=ALL, ADJUNCT=CAPTURE, ENTRYDATA=UNSER))
IEE600I REPLY TO 01 IS;STRLIST=(STRNAME=QSG1APPLICATION1, (LISTNUM=
IEA794I SVC DUMP HAS CAPTURED: 677
DUMPID=057 REQUESTED BY JOB (*MASTER*)
DUMP TITLE='MQ APPLICATION STRUCTURE 1 DUMP'

```

图 45: 转储耦合设施结构

使用 IBM MQ for z/OS 转储显示面板处理转储

您可以使用 IPCS 面板提供的命令来处理转储。本主题可用于了解 IPCS 选项。

IBM MQ for z/OS 提供了一组面板以帮助您处理转储。下一节描述了如何使用这些面板：

1. 从 IPCS 主选项菜单中选择 **分析 - 分析转储内容** (选项 2)。

这样会显示“IPCS MVS 转储内容分析”面板。

2. 选择 **组件 - MVS 组件数据** (选项 6)。

这样会显示“IPCS MVS 转储组件数据分析”面板。此面板的外观取决于安装时所安装的产品，但类似于 IPCS MVS 转储组件数据分析面板 中所示的面板：

```
----- IPCS MVS DUMP COMPONENT DATA ANALYSIS -----
OPTION ==>                                SCROLL ==>

To display information, specify "S option name" or enter S to the
left of the option required.  Enter ? to the left of an option to
display help regarding the component support.

Name      Abstract
ALCWAIT   Allocation wait summary
AOMDATA   AOM analysis
ASMCHECK  Auxiliary storage paging activity
ASMDATA   ASM control block analysis
AVMDATA   AVM control block analysis
COMCHECK  Operator communications data
CSQMAIN   WebSphere MQ dump formatter panel interface
CSQWDMP   WebSphere MQ dump formatter
CTRACE    Component trace summary
DAEDATA   DAE header data
DIVDATA   Data-in-virtual storage
```

图 46: IPCS MVS 转储组件数据分析面板

3. 通过在行旁边输入 **s** 并按 **Enter** 键来选择 **CSQMAIN IBM MQ 转储格式化程序** 面板接口。

如果该选项不可用，原因是成员 **CSQ7IPCS** 不存在；您应查看 [配置 z/OS](#) 以获取有关安装 IBM MQ for z/OS 转储格式化成员的更多信息。

注: 如果已使用转储来执行初级分析，并且希望重新检查，请选择 **CSQWDMP IBM MQ 转储格式化程序** 以使用缺省选项重新显示格式化的内容。

4. 这样会显示“IBM MQ for z/OS - 转储分析”菜单。此菜单可用于指定要对系统转储执行的操作。

```
-----IBM WebSphere MQ for z/OS - DUMP ANALYSIS-----
COMMAND ==>

1 Display all dump titles 00 through 99
2 Manage the dump inventory
3 Select a dump

4 Display address spaces active at time of dump
5 Display the symptom string
6 Display the symptom string and other related data
7 Display LOGREC data from the buffer in the dump
8 Format and display the dump

9 Issue IPCS command or CLIST

(c) Copyright IBM Corporation 1993, 2024. All rights reserved.

F1=Help   F3=Exit   F12=Cancel
```

5. 在选择特定转储以进行分析之前，所需转储必须存在于转储库存中。为确保转储存在，请执行以下步骤：

- a. 如果您不知道包含转储的数据集的名称，请指定选项 **1 - 显示 xx 到 xx 的所有转储标题**。

这样会显示 **SYS1.DUMP** 数据集中包含的所有转储的转储标题（其中 **xx** 是范围 **00** 到 **99** 之间的数字）。您可以通过使用 **xx** 字段来指定数据集编号范围，来限制显示的数据集的选择。

如果要查看所有可用转储数据集的详细信息，请将这些值设置为 **00** 和 **99**。

使用显示的信息来识别要分析的转储。

- b. 如果转储尚未复制到其他数据集（即，它位于某一个 SYS1.DUMP 数据集内），请指定选项 2 - **管理转储库存**

转储库存包含您已使用的转储数据集。由于复用 SYS1.DUMP 数据集，因此在步骤 第 201 页的『5.a』中识别的转储名称可能包含在显示的列表中。但是，此条目引用此数据集中存储的先前转储，因此请通过在此条目旁输入 DD 并按 Enter 键来将其删除。然后按 F3 键返回转储分析菜单。

6. 指定选项 3 - **选择转储**以选择要处理的转储。在“源”字段中输入包含转储的数据集的名称，检查在“消息路由”字段中是否指定了 NOPRINT 和 TERMINAL（这是为了确保将输出定向至终端），然后按 Enter 键。按 F3 键返回转储分析菜单。
7. 选择要处理的转储后，可以使用菜单上的其他选项来分析不同转储部分中的数据：
 - 要显示生成转储时处于活动状态的所有地址空间的列表，请选择选项 4。
 - 要显示症状字符串，请选择选项 5。
 - 要显示症状字符串和其他可维护性信息（包括系统诊断工作区 (SDWA) 的可变记录区域），请选择选项 6。
 - 要格式化并显示存储 LOGREC 缓冲区内包含的数据，请选择选项 7。

导致转储的异常终止可能不是导致错误原因，错误是由先前的问题导致的。要确定哪个 LOGREC 记录与问题的原因相关，请转至数据集的末尾，输入 FIND ERRORID: PREV，然后按 Enter 键。这样会显示最新 LOGREC 记录的标题，例如：

```
JOBNAME: NONE-FRR
ERRORID: SEQ=00081 CPU=0040 ASID=0033 TIME=14:42:47.1

SEARCH ARGUMENT ABSTRACT

PIDS/5655R3600 RIDS/CSQRLLM1#L RIDS/CSQRRHSL AB/S05C6
PRCS/00D10231 REGS/0C1F0 RIDS/CSQVEUS2#R

SYMPTOM          DESCRIPTION
-----          -
PIDS/5655R3600   PROGRAM ID: 5655R3600
.
.
.
```

请注意程序标识（如果此标识不是 5655R3600，那么问题不是由 IBM MQ for z/OS 导致的，您可能找错了转储）。并且，请注意 TIME 字段的值。重复此命令以查找前一条 LOGREC 记录，同样注意 TIME 字段的值。如果这两个值彼此接近（如在十分之一或十分之二秒内），那么可能都与同一个问题有关。

- 要格式化并显示转储，请选择选项 8。将显示 FORMAT AND DISPLAY THE DUMP 面板：

```

-----IBM MQ for z/OS - FORMAT AND DISPLAY DUMP-----
COMMAND ==>

1 Display the control blocks and trace
2 Display just the control blocks
3 Display just the trace

Options:

Use the summary dump? . . . . . __ 1 Yes
2 No

Subsystem name (required if summary dump not used) ____

Address space identifier or ALL. . . . . ALL_

F1=Help F3=Exit F12=Cancel

```

- 此面板可用于格式化所选系统转储。您可以选择显示控制块或内部跟踪产生的数据，或者同时显示两者（缺省设置）。

注: 针对来自通道启动程序或者耦合设施结构的转储，不能执行此选择。

– 要显示整个转储，即：

- 转储标题
- 可变记录区域 (VRA) 诊断信息报告
- 保存区域跟踪报告
- 控制块摘要
- 跟踪表

请选择选项 1。

- 要显示针对选项 1 列出的信息（不含跟踪表），请选择选项 2。
- 要显示针对选项 1 列出的信息（不含控制块），请选择选项 3。

您还可以使用以下选项：

– **使用摘要转储？**

该字段可用于指定是否希望 IBM MQ 在格式化所选转储时使用摘要部分中包含的信息。缺省设置为“是”。

注: 如果已生成摘要转储，那么其中可能包含来自多个地址空间的数据。

– **子系统名称**

该字段可用于识别具有要显示的转储数据的子系统。仅当没有摘要数据（例如，如果操作员请求转储），或者您在**是否使用摘要转储？**中指定了 NO 时，才需要执行此操作 字段。

如果您不知道子系统名称，请在命令提示符处输入 `IPCS SELECT ALL`，然后按 `Enter` 键以显示在错误发生时正在运行的所有作业的列表。如果某个作业在 `SELECTION CRITERIA` 列中包含字 `ERROR`，请记录此作业的名称。作业名格式为 `xxxx MSTR`，其中 `xxxx` 是子系统名称。

```

IPCS OUTPUT STREAM -----
COMMAND ==>
ASID JOBNAME ASCBADDR SELECTION CRITERIA
-----
0001 *MASTER* 00FD4D80 ALL
0002 PCAUTH 00F8AB80 ALL
0003 RASP 00F8C100 ALL
0004 TRACE 00F8BE00 ALL
0005 GRS 00F8BC00 ALL
0006 DUMPSRV 00F8DE00 ALL
0008 CONSOLE 00FA7E00 ALL
0009 ALLOCAS 00F8D780 ALL
000A SMF 00FA4A00 ALL
000B VLF 00FA4800 ALL
000C LLA 00FA4600 ALL
000D JESM 00F71E00 ALL
001F MQM1MSTR 00FA0680 ERROR ALL

```

如果没有任何作业在 SELECTION CRITERIA 列中包含字 ERROR，请在主 IPCS 选项菜单面板上选择选项 0 - 缺省值，以显示“IPCS 缺省值”面板。记录地址空间标识 (ASID)，然后按 F3 以返回上一个面板。使用 ASID 来确定作业名，格式为 xxxx MSTR，其中 xxxx 是子系统名称。

以下命令显示了转储数据集中包含的 ASID：

```
LDMP DSN('SYS1.DUMPxx') SELECT(DUMPED) NOSUMMARY
```

这显示了针对每个地址空间转储的存储范围。

按 F3 以返回至“格式化并显示转储”面板，并在子系统名称字段中输入此名称。

– 地址空间标识

如果转储中的数据来自多个地址空间，请使用该字段。如果只需查看来自特定地址空间的数据，请指定该地址空间的标识 (ASID)。

此字段的缺省值为“ALL”，即显示有关与转储中子系统相关的所有地址空间信息。在显示的值上方输入 4 个字符的 ASID 来更改此字段。

注：由于转储包含对所有地址空间公用的存储区域，如果您未正确指定地址空间标识，那么显示的信息可能与您的问题无关。在此情况下，请返回至此面板，并输入正确的地址空间标识。

相关概念

第 204 页的『[使用行式 IPCS 处理 IBM MQ for z/OS 转储](#)』

使用 IPCS 命令来格式化转储。

第 211 页的『[以批处理方式使用 IPCS 处理 IBM MQ for z/OS 转储](#)』

本主题可用于了解如何在批处理方式下通过 IPCS 命令格式化 IBM MQ for z/OS 转储。

第 212 页的『[在 z/OS 上分析转储和解释转储标题](#)』

本主题可用于了解 IBM MQ for z/OS 转储标题的格式化方式以及转储的分析方式。

使用行式 IPCS 处理 IBM MQ for z/OS 转储

使用 IPCS 命令来格式化转储。

要使用行式 IPCS 命令格式化转储，请发出以下命令选择所需转储：

```
SETDEF DSN('SYS1.DUMP xx')
```

(其中 SYS1.DUMP xx 是包含转储的数据集的名称)。然后，您可以使用 IPCS 子命令显示来自转储的数据。

请参阅以下主题以获取有关如何使用 IPCS 命令格式化不同类型的转储的信息：

- 第 205 页的『[格式化 IBM MQ for z/OS 转储](#)』
- 第 210 页的『[在 z/OS 上从通道启动程序格式化转储](#)』

相关概念

第 200 页的『使用 IBM MQ for z/OS 转储显示面板处理转储』

您可以使用 IPCS 面板提供的命令来处理转储。本主题可用于了解 IPCS 选项。

第 211 页的『以批处理方式使用 IPCS 处理 IBM MQ for z/OS 转储』

本主题可用于了解如何在批处理方式下通过 IPCS 命令格式化 IBM MQ for z/OS 转储。

第 212 页的『在 z/OS 上分析转储和解释转储标题』

本主题可用于了解 IBM MQ for z/OS 转储标题的格式化方式以及转储的分析方式。

格式化 IBM MQ for z/OS 转储

本主题用于了解如何使用行式 IPCS 命令格式化队列管理器转储。

IPCS VERBEXIT CSQWDMP 会调用 IBM MQ for z/OS 转储格式化程序 (CSQWDPRD)，然后支持您格式化 SVC 转储以显示 IBM MQ 数据。您可以通过指定参数来限制显示的数据量。

IBM 服务人员可能需要您的耦合设施管理结构和队列共享组应用程序结构的转储以及队列共享组中队列管理器的转储，以帮助进行问题诊断。有关格式化耦合设施列表结构和 STRDATA 子命令的信息，请参阅 [z/OS MVS IPCS Commands](#) 手册。

注：本节描述了抽取必要数据所需的参数。使用逗号而不是空格来分隔操作数。控制语句中任何操作数后跟一个空格会终止操作数列表，并忽略所有后续操作数。第 205 页的表 15 对在控制语句中可针对格式化转储指定的每个关键字进行了解释。

| 关键字 | 描述 |
|--------------------------------------|--|
| SUBSYS= <i>aaaa</i> | 如果摘要转储部分不可用或者将不使用此部分，请使用此关键字来提供要格式化的子系统的名称。 <i>aaaa</i> 是 1 到 4 个字符的子系统名称。 |
| ALL (缺省值) | 所有控制块和跟踪表。 |
| AA | 针对所有地址空间内的所有 IBM MQ for z/OS 控制块显示数据。 |
| DIAG=Y | 列显诊断信息。仅限在 IBM 服务人员的指导下使用。DIAG=N (禁止格式化诊断信息) 为缺省值。 |
| EB= <i>nnnnnnnn</i> | 仅显示与此 EB 线程关联的跟踪点 (此关键字的格式为 EB= <i>nnnnnnnn</i> ，其中 <i>nnnnnnnn</i> 是跟踪中包含的 EB 线程的 8 位数地址)。必须将此关键字与 TT 关键字一起使用。 |
| LG | 所有控制块。 |
| PTF=Y, LOAD= <i>load module name</i> | 位于报告前部 (来自 MEPL) 的 PTF 列表。PTF=N (禁止格式化此类列表) 为缺省值。 可选装入子参数支持您指定要为其格式化 PTF 报告的装入模块的名称 (最多 8 个字符)。 |
| SA= <i>hhhh</i> | 指定地址空间的控制块。使用以下任一格式： <ul style="list-style-type: none">SA= <i>hh</i> 或SA= <i>hhhh</i> 其中 <i>h</i> 表示十六进制数字。 |
| SG | 系统范围控制块的子集。 |

表 15: IBM MQ for z/OS 转储格式化控制语句的关键字 (继续)

| 关键字 | 描述 |
|------------------|-----------------------|
| TT | 格式化跟踪表 |
| ,HANDLES=x | 指示具有超过 x 个句柄的线程 |
| ,LOCKS=x | 指示具有超过 x 个锁定的线程 |
| ,INSYNCS=x | 指示具有超过 x 个同步操作的线程 |
| ,URINFO=ALL/LONG | 显示所有线程或长期运行的线程的 UR 信息 |

第 206 页的表 16 详述了可供您用于格式化个别资源管理器相关数据的转储格式化关键字。

不能将这些关键字与第 205 页的表 15 中的任何关键字一起使用。

表 16: 资源管理器转储格式化关键字

| 关键字 | 格式化的对象 |
|----------------------|--|
| BMC=1 | 缓冲区管理器数据。BMC=1 会格式化所有缓冲区的控制块。 |
| BMC=2(缓冲池编号) | BMC=2 会格式化与 2 位数缓冲池编号中标识的缓冲区相关的数据。 |
| BMC=3(xx/yyyyyy) | BMC=3 和 BMC=4 显示来自页集的某个页面，前提是此页面存在于缓冲区中。（BMC=3 与 BMC=4 之间的差异在于访问页面所采用的路径。） |
| BMC=4(xx/yyyyyy) | |
| BUFL= nnnnnnnnnnn | 存储访问缓冲区分配 sz。 |
| CALLD=Y | 显示 TT 中表示调用深度的箭头。 |
| =W | 和缩进跟踪条目。 |
| CALLTIME=Y | 列显出口跟踪的调用时间。 |
| CB=(addr/[strmodel]) | 将地址格式化为 IBM MQ 块。 |
| CBF=1 | CBF 报告级别 1。 |
| CCB=S | 显示 TT 中系统 EB 的组合功能块 (CCB)。 |
| CFS=1 | CFS 报告级别 1。 |
| CFS=2 | CFS 报告级别 2。 |
| CHLAUTH=1/2 | CHLAUTH 报告级别。 |
| ONAM=20 chars | 可选 ONAM 子参数支持您指定对象名称（最多 20 个字符）以将列显的数据限制为以 ONAM 中的字符开头的对象。 |
| CLUS=1 | 集群报告，包含队列管理器上已知的集群存储库。 |
| CLUS=2 | 显示集群注册的集群报告。 |
| CLXQ=1 | 集群 XMITQ 报告级别 1。 |
| CLXQ=2 | 集群 XMITQ 报告级别 2。 |
| ONAM=20 chars | 可选 ONAM 子参数支持您指定对象名称（最多 20 个字符）以将列显的数据限制为以 ONAM 中的字符开头的对象。 |
| CMD=0/1/2 | 命令跟踪表显示级别。 |
| D=1/2/3 | 部分报告的详细级别。 |
| Db2=1 | Db2 报告级别 1。 |

| 关键字 | 格式化的对象 |
|-------------------------|---|
| DMC=1, ONAM=48 chars | DMC 报告级别 1。 可选 ONAM 子参数支持您指定对象名称 (最多 48 个字符) 以将列显的数据限制为以 ONAM 中的字符开头的对象。 |
| DMC=2, ONAM=48 chars | DMC 报告级别 2。 可选 ONAM 子参数支持您将列显的对象限制为名称以 ONAM 中指定的字符开头的对象 (最多 48 个字符)。 |
| DMC=3, ONAM=48 chars | DMC 报告级别 3。 可选 ONAM 子参数支持您将列显的对象限制为名称以 ONAM 中指定的字符开头的对象 (最多 48 个字符)。 |
| GR=1 | 组不确定报告级别 1。 |
| IMS=1 | IMS 报告级别 1 |

| 关键字 | 格式化的对象 |
|---|---|
| JOBNAME= xxxxxxxx | 作业名 |
| LKM=1 | LKM 报告级别 1。 |
| LKM=2/3, ,NAME=up to 48 chars ,NAMEX= xxxxxxxxxxxxxxxxx ,NAMESP=1/2/3/4/5/6/7/8 ,TYPE=DMCP/QUALNM/TOPIC/ STGCLASS ,QUAL=GET/PUT/CRE/DFXQ/ PGSYNC/CHGCNT/ DELETE/EXPIRE LKM=3 LKM=4 ,JOBNAME= xxxxxxxx ,ASID= xxxx | LKM 报告级别 2/3。 名称 (字符) 名称 (十六进制) 名称空间 锁定类型 锁定资质 LKM 报告级别 3 LKM 报告级别 4 |
| LMC=1 | LMC 报告级别 1。 |
| MAXTR= nnnnnnnnn | 要格式化的最大跟踪条目数 |
| MHASID= xxxx | 属性的消息句柄 ASID |

| 表 17: 资源管理器转储格式化关键字 (J-P) (继续) | |
|---|---|
| 关键字 | 格式化的对象 |
| MMC=1 OBJ=MQLO/MQSH/MQRO/ MQAO/MQMO/MCHL/ MNLS/MSTC/MPRC/ :' MAUT ONAM | MMC 报告级别 1 对象类型 可选 ONAM 子参数支持您将列显的对象限制为名称以 ONAM 中指定的字符开头的对象（最多 48 个字符）。 |
| MMC=2 ONAM=48 chars | MMC 报告级别 2 可选 ONAM 子参数支持您将列显的对象限制为名称以 ONAM 中指定的字符开头的对象（最多 48 个字符）。 |
| MSG=nnnnnnnnnnnnnnnnnn MASID=xxxx LEN=xxxxxxxx MSGD=S/D | 格式化指针处的消息。 MASID 允许使用其他地址空间内的存储。 LEN 用于限制要格式化的存储量。 MSGD 可控制详细级别。 |
| MSGD=S/D | DMC=3、BMC=3/4、PSID 报告中的消息详细信息。 此参数可控制详细级别，S 表示摘要，D 表示详细。 |
| MSGH = nnnnnnnnnnnnnnnnn | 消息句柄 |
| MT | 消息属性跟踪 |
| MQVCX | 十六进制格式的 MQCHARV |
| PROPS= nnnnnnnnnnnnnnnnn | 消息属性指针 |
| PSID= nnnnnnnnn | 用于格式化页面的页集 |
| PSTRX | 十六进制格式的属字符串 |

| 表 18: 资源管理器转储格式化关键字 (R-Z) | |
|---------------------------|--|
| 关键字 | 格式化的对象 |
| RPR= nnnnnnnnn | 要格式化的页面或记录 |
| SHOWDEL | 显示 DMC=3 的删除记录 |
| SMC=1/2/3 | 存储管理器 |
| TC= * A E O | TT 数据字符型格式，已合并 以合适的字符集列显所有内容 始终列显 ASCII 始终列显 EBCDIC 从不列显 |
| TFMT=H/M | 时间格式 - 人员或 STCK |
| THR= nnnnnnnnn | 线程地址 |
| THR=*/2/3 | 设置线程报告级别 |

| 表 18: 资源管理器转储格式化关键字 (R-Z) (继续) | |
|--|--|
| 关键字 | 格式化的对象 |
| TOP=1 | TOP 报告级别 1 |
| TOP=2 | TOP 报告级别 2 |
| TOP= nnnnnnnnnnnnnnnn /TSTR=48 chars /TSTRX=hex 1208 str | Tnode 64 位地址或 主题字符串（以 % 开头或结尾的通配符） 这将 EBCDIC 转换为 ASCII，但仅限不可变字符 1208 中的十六进制主题字符串始终以通配符开头。 |
| TOP=3 | TOP 报告级别 3 |
| TOP=4 | TOP 报告级别 4 |
| TSEG=M(RU)/Q(P64) I(NTERPOLATE) F(WD) D(EBUG) | 搜索 64 位跟踪的进程 猜测缺失的 TSEG 地址 强制转发排序 调试搜索进程 |
| TSEG=(M,Q,I,F,D) | 指定多个 TSEG 选项 |
| W=0/1/2/3 | TT 宽度格式 |
| XA=1 | XA 报告级别 1 |
| ZMH = nnnnnnnnnnnnnnnn | ZST 消息句柄 |

如果转储是由操作员启动的，那么转储的摘要部分中没有任何信息。第 209 页的表 19 显示了可在 CSQWDMP 控制语句中使用的其他关键字。

| 表 19: IBM MQ for z/OS 转储格式化控制语句的摘要转储关键字 | |
|---|---|
| 关键字 | 描述 |
| SUBSYS= aaaa | 如果摘要转储部分不可用或者将不使用此部分，请使用此关键字来提供要格式化其信息的子系统的名称。aaaa 是 1 到 4 个字符的子系统名称。 |
| SUMDUMP=NO | 如果转储具有摘要部分，但您不想使用此部分，请使用此关键字。（仅限根据 IBM 支持中心的指示才能使用此关键字。） |

以下列表显示了如何使用这些关键字的部分示例：

- 要使用来自转储的摘要部分的信息以缺省方式格式化所有地址空间，请使用：

```
VERBX CSQWDMP
```

- 要显示来自操作员启动、不包含摘要部分的名为 MQMT 的子系统的转储的跟踪表，请使用：

```
VERBX CSQWDMP 'TT,SUBSYS=MQMT'
```

- 要显示来自自由子系统异常终止生成的转储的所有控制块和跟踪表，对于 ASID（地址空间标识）为 1F 的地址空间，请使用：

```
VERBX CSQWDMP 'TT,LG,SA=1F'
```

- 要显示来自与特定 EB 线程关联的转储的跟踪表部分，请使用：

```
VERBX CSQWDMP 'TT,EB= nnnnnnnn '
```

- 要显示名称以“ABC”开头的本地非共享队列对象的消息管理器 1 报告，请使用：

```
VERBX CSQWDMP 'MMC=1,ONAM=ABC,Obj=MQLO'
```


第 210 页的表 20 显示了频繁用于分析转储的部分其他命令。有关这些子命令的更多信息，请参阅 [z/OS MVS IPCS Commands](#) 手册。

| 子命令 | 描述 |
|--|--|
| 状态 | 用于显示通常在问题确定过程初始部分期间检查的数据。 |
| STRDATA LISTNUM(ALL) ENTRYPOS(ALL) DETAIL | 用于格式化耦合设施结构数据。 |
| VERBEXIT LOGDATA | 用于格式化在生成转储之前存在的存储 LOGREC 缓冲区记录。LOGDATA 可查找 LOGREC 记录缓冲区中包含的 LOGREC 条目，并调用 EREP 程序来格式化和列显 LOGREC 条目。这些条目根据常规详细编辑报告的样式进行格式化。 |
| VERBEXIT TRACE | 用于格式化所有地址空间的系统跟踪条目。 |
| VERBEXIT SYMPTOM | 用于格式化系统转储的标题记录中包含的症状字符串，例如，独立转储、SVC 转储或使用 SYSUDUMP DD 语句请求的异常终止转储。 |
| VERBEXIT GRSTRACE | 用于格式化来自全局资源序列化的主要控制块的诊断数据。 |
| VERBEXIT SUMDUMP | 用于查找和显示 SVC 转储提供的摘要转储数据。 |
| VERBEXIT DAEDATA | 用于格式化已转储系统的转储分析和消除 (DAE) 数据。 |

相关概念

第 210 页的『[在 z/OS 上从通道启动程序格式化转储](#)』

本主题用于了解如何使用行式 IPCS 命令格式化 IBM MQ for z/OS 的通道启动程序转储。

 在 z/OS 上从通道启动程序格式化转储

本主题用于了解如何使用行式 IPCS 命令格式化 IBM MQ for z/OS 的通道启动程序转储。

IPCS VERBEXIT CSQXDPRD 支持您格式化通道启动程序转储。您可以通过指定关键字选择要格式化的数据。

本部分描述了您可指定的关键字。

第 210 页的表 21 描述了您可通过 CSQXDPRD 指定的关键字。

| 关键字 | 格式化的对象 |
|--------------|--------------------------------------|
| SUBSYS= aaaa | 与指定子系统关联的通道启动程序的控制块。这是所有新格式化的转储所必需的。 |

表 21: IPCS VERBEXIT CSQXDPRD 的关键字 (继续)

| 关键字 | 格式化的对象 |
|---|--|
| CHST=1, CNAM= <i>channel name</i> , DUMP=S F C | 所有通道信息。 可选 CNAM 子参数支持您指定要为其格式化详细信息的通道名称（最多 20 个字符）。 可选 DUMP 子参数支持您控制格式化的范围，如下所示： <ul style="list-style-type: none"> 指定 DUMP=S（表示“简短 (short)”）以格式化通道数据的十六进制转储的第一行。 指定 DUMP=F（表示“完整 (full)”）以格式化数据的所有行。 指定 DUMP=C（表示“压缩 (compressed)”）以禁止格式化仅包含 X'00' 的数据中的所有重复行。这是缺省选项 |
| CHST=2, CNAM= <i>channel name</i> , | 所有通道或者由 CNAM 关键字指定的通道的摘要。 请参阅 CHST=1 以获取有关 CNAM 子参数的详细信息。 |
| CHST=3, CNAM= <i>channel name</i> , | 由 CHST=2 提供的数据以及转储中所有通道的程序跟踪、行跟踪和格式化信标表列显。 请参阅 CHST=1 以获取有关 CNAM 子参数的详细信息。 |
| CLUS=1 | 集群报告，包含队列管理器上已知的集群存储库。 |
| CLUS=2 | 显示集群注册的集群报告。 |
| CTRACE=S F, DPRO= <i>nnnnnnnn</i> , TCB= <i>nnnnnnn</i> | 选择简短 (CTRACE=S) 或完整 (CTRACE=F) CTRACE。 可选 DPRO 子参数支持您为指定的 DPRO 指定 CTRACE。 可选 TCB 子参数支持您为指定的作业指定 CTRACE。 |
| DISP=1, DUMP=S F C | 分派器报告 请参阅 CHST=1 以获取有关 DUMP 子参数的详细信息。 |
| BUF=1 | 缓冲区报告 |
| XSMF=1 | 格式化转储中可用的通道启动程序 SMF 数据。 |

相关概念

第 205 页的『格式化 IBM MQ for z/OS 转储』

本主题用于了解如何使用行式 IPCS 命令格式化队列管理器转储。

以批处理方式使用 IPCS 处理 IBM MQ for z/OS 转储

本主题可用于了解如何在批处理方式下通过 IPCS 命令格式化 IBM MQ for z/OS 转储。

要在批处理中使用 IPCS，请将所需 IPCS 语句插入批处理作业流（请参阅第 212 页的图 47）。

更改 DUMP00 语句上的数据集名称 (DSN=) 以反映要处理的转储，并插入要使用的 IPCS 子命令。

```

//*****
//*  RUNNING IPCS IN A BATCH JOB          *
//*****
//MQMDMP EXEC PGM=IKJEFT01,REGION=5120K
//STEPLIB DD DSN=mqm.library-name,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//IPCSPRNT DD SYSOUT=*
//IPCSDDIR DD DSN=dump.directory-name,DISP=OLD
//DUMP00 DD DSN=dump.name,DISP=SHR
//SYSTSIN DD *
IPCS NOPARM TASKLIB(SCSQLOAD)
SETDEF PRINT TERMINAL DDNAME(DUMP00) NOCONFIRM
*****
* INSERT YOUR IPCS COMMANDS HERE, FOR EXAMPLE: *
VERBEXIT LOGDATA
VERBEXIT SYMPTOM
VERBEXIT CSQWDMP 'TT,SUBSYS=QMGR'
*****

CLOSE ALL
END
/*

```

图 47: 在 z/OS 环境中通过 IPCS 列显转储的样本 JCL

相关概念

第 200 页的『[使用 IBM MQ for z/OS 转储显示面板处理转储](#)』

您可以使用 IPCS 面板提供的命令来处理转储。本主题可用于了解 IPCS 选项。

第 204 页的『[使用行式 IPCS 处理 IBM MQ for z/OS 转储](#)』

使用 IPCS 命令来格式化转储。

第 212 页的『[在 z/OS 上分析转储和解释转储标题](#)』

本主题可用于了解 IBM MQ for z/OS 转储标题的格式化方式以及转储的分析方式。

► z/OS 在 z/OS 上分析转储和解释转储标题

本主题可用于了解 IBM MQ for z/OS 转储标题的格式化方式以及转储的分析方式。

- [分析转储](#)
- [含 PSW 和 ASID 的转储标题变体](#)

分析转储

转储标题包含异常终止完成代码和原因码、失败的装入模块和 CSECT 名称以及发行版标识。有关转储标题的更多信息，请参阅[含 PSW 和 ASID 的转储标题变体](#)

SVC 转储标题的格式根据错误类型而略有不同。

第 212 页的图 48 显示了 SVC 转储标题的示例。图后描述了标题中的每个字段。

```

ssnm,ABN=5C6-00D303F2,U=AUSER,C=R3600. 710.LOCK-CSQL1GET,
M=CSQGFRCV,LOC=CSQLPLM.CSQL1GET+0246

```

图 48: 样本 SVC 转储标题

ssnm,ABN=compltn-reason

- `ssnm` 是发出转储的子系统的名称。
- `compltn` 是 3 个字符的十六进制异常终止完成代码（在此示例中为 X'5C6'），使用前缀 U 表示用户异常终止代码。
- `reason` 是 4 字节的十六进制原因码（在此示例中为 X'00D303F2'）。

注: 异常终止代码和原因码可提供充足信息以解决问题。请参阅 [IBM MQ for z/OS 消息代码、完成代码和原因码](#) 以获取原因码的说明。

U=userid

- `userid` 是用户的用户标识（在此示例中为 `AUSER`）。针对通道启动程序，此字段不显示。

C=compid.release.comp-function

- `compid` 是组件标识的最后 5 个字符。值 `R3600` 唯一地标识 IBM MQ for z/OS。
- `release` 是 3 位数的代码，指示 IBM MQ for z/OS 的版本、发行版和修改级别（在此示例中为 `710`）。
- `comp` 是发生异常终止时控制的组件的首字母缩写词（在此示例中为 `LOCK`）。
- `function` 是发生异常终止时控制的函数、宏或例程的名称（在此示例中为 `CSQL1GET`）。此字段并非始终显示。

M=module

- `module` 是 FRR 或 ESTAE 恢复例程的名称（在此示例中为 `CSQGFRCV`）。此字段并非始终显示。

注: 这并非发生异常终止的模块的名称（此名称由 `LOC` 提供）。

LOC=loadmod.csect+csect_offset

- `loadmod` 是发生异常终止时控制的装入模块的名称（在此示例中为 `CSQLLPLM`）。如果此名称未知，那么可能由星号来表示。
- `csect` 是发生异常终止时控制的 CSECT 的名称（在此示例中为 `CSQL1GET`）。
- `csect_offset` 是发生异常终止时失败的 CSECT 中的偏移量（在此示例中为 `0246`）。

注: 如果已将服务应用于此 CSECT，`csect_offset` 的值可能变化，因此在构建关键字字符串以搜索 IBM 软件支持数据库时，请勿使用此值。

含 PSW 和 ASID 的转储标题变体

部分转储标题会使用 PSW（程序状态字）和 ASID（地址空间标识）来替换装入模块名称、CSECT 名称和 CSECT 偏移量。第 213 页的图 49 展示了此格式。

```
ssnm,ABN=compltn-reason,U=userid,C=compid.release.comp-function,  
M=module,PSW=psw_contents,ASID=address_space_id
```

图 49: 含 PSW 和 ASID 的转储标题

psw_contents

- 发生错误时的 PSW（例如，`X'077C100000729F9C'`）。

address_space_id

- 发生异常终止时控制的地址空间（例如，`X'0011'`）。针对通道启动程序，此字段不显示。

相关概念

第 200 页的『[使用 IBM MQ for z/OS 转储显示面板处理转储](#)』

您可以使用 IPCS 面板提供的命令来处理转储。本主题可用于了解 IPCS 选项。

第 204 页的『[使用行式 IPCS 处理 IBM MQ for z/OS 转储](#)』

使用 IPCS 命令来格式化转储。

第 211 页的『[以批处理方式使用 IPCS 处理 IBM MQ for z/OS 转储](#)』

本主题可用于了解如何在批处理方式下通过 IPCS 命令格式化 IBM MQ for z/OS 转储。

z/OS 上的 SYSUDUMP 信息

z/OS 系统可创建 SYSUDUMP 以用作问题确定的一部分。本主题显示了样本 SYSUDUMP 输出，并为用于解释 SYSUDUMP 的工具提供参考。

SYSUDUMP 转储提供了可用于调试批处理和 TSO 应用程序的实用信息。有关 SYSUDUMP 转储的更多信息，请参阅 *z/OS MVS Diagnosis: Tools and Service Aids* 手册。

第 214 页的图 50 显示了 SYSUDUMP 转储开始的样本。

```
JOB MQMBXBA1 STEP TSUSER TIME 102912 DATE 001019 ID = 000 CPUID = 632202333081
PAGE 00000001

COMPLETION CODE          SYSTEM = 0C1          REASON CODE = 00000001

PSW AT ENTRY TO ABEND  078D1000 000433FC          ILC 2  INTC 000D

PSW LOAD MODULE = BXBAAB01 ADDRESS = 000433FC OFFSET = 0000A7F4

ASCB: 00F56400
+0000 ASCB..... ASCB          FWDP..... 00F60180  BWDP..... 0047800  CMSF..... 019D5A30
SVRB..... 008FE9E0
+0014 SYNC..... 00000D6F  IOSP..... 00000000  TNEW..... 00D18F0  CPUS..... 00000001
ASID..... 0066
+0026 R026..... 0000          LL5..... 00          HLHI..... 01          DPHI..... 00
DP..... 9D
+002C TRQP..... 80F5D381  LDA..... 7FF154E8  RSMF..... 00          R035..... 0000
TRQI..... 42
+0038 CSCB..... 00F4D048  TSB..... 00B61938  EJST..... 00000001  8C257E00

+0048 EWST..... 9CCDE747  76A09480          JSTL..... 00141A4  ECB..... 808FEF78
UBET..... 9CCDE740
.
.
.
ASSB: 01946600
+0000 ASSB..... ASSB          VAFN..... 00000000  EVST..... 00000000  00000000

+0010 VFAT..... 00000000  00000000          RSV..... 000          XMCC..... 0000
XMCT.....00000000
+0020 VSC..... 00000000  NVSC..... 0000004C  ASRR..... 00000000  R02C..... 00000000
00000000 00000000
+0038          00000000  00000000

*** ADDRESS SPACE SWITCH EVENT MASK OFF (ASTESSEM = 0) ***

TCB: 008D18F0
+0000 RBP..... 008FE7D8  PIE..... 00000000  DEB..... 00B1530  TIO..... 008D4000
CMP.....805C6000
+0014 TRN..... 40000000  MSS..... 7FFF7418  PKF..... 80          FLGS..... 01000000  00
+0022 LMP..... FF          DSP..... FE          LLS..... 00D1A88  JLB..... 00011F18
JPQ.....00000000
+0030 GPR0-3... 00001000  008A4000  00000000  00000000
+0040 GPR4-7.. 00FDC730  008A50C8  00000002  80E73F04
+0050 GPR8-11.. 81CC4360  008A6754  008A67B4  00000008
```

图 50: SYSUDUMP 开始的样本

z/OS 上的快速转储

快速转储数据集由 z/OS JCL 命令语句进行控制。本主题可用于了解 CSQSNAP DD 语句。

快速转储始终发送至由 CSQSNAP DD 语句定义的数据集。快速转储可由适配器或通道启动程序发出。

- 当队列管理器针对 MQI 调用返回意外错误时，快速转储由批处理、CICS、IMS 或 RRS 适配器发出。生成完整转储，其中包含有关导致问题的程序的信息。

要生成快速转储，必须在批处理应用程序 JCL、CICS JCL 或 IMS 从属区域 JCL 中包含 CSQSNAP DD 语句。

- 在特定错误情况下，由通道启动程序发出快速转储而不是系统转储。此转储包含错误相关信息。同时还会发出消息 CSQX053E。

要生成快速转储，必须在通道启动程序启动的任务过程中包含 CSQSNAP DD 语句。

z/OS 上的 SYS1.LOGREC 信息

本主题可用于了解 z/OS SYS1.LOGREC 信息如何帮助确定问题。

IBM MQ for z/OS 和 SYS1.LOGREC

SYS1.LOGREC 数据集可记录操作系统的不同组件遇到的各种错误。有关使用 SYS1.LOGREC 记录，请参阅 *z/OS MVS Diagnosis: Tools and Service Aids* 手册。

IBM MQ for z/OS 恢复例程会在尝试重试时或者发生到下一个恢复例程的渗透时，将系统诊断工作区 (SDWA) 中的信息写入 SYS1.LOGREC 数据集。可记录多个 SYS1.LOGREC 条目，因为针对单个错误可能发生多次重试或渗透。

在临近异常终止时记录的 SYS1.LOGREC 条目可提供有关导致异常终止的事件的宝贵历史记录信息。

查找适用的 SYS1.LOGREC 信息

要获取 SYS1.LOGREC 列表，请采用下列其中一种方法：

- 请参阅 *z/OS MVS Diagnosis: Tools and Service Aids* 手册中描述的 [EREP 选择参数](#)，以在 SYS1.LOGREC 数据集。
- 在 IPCS 中指定 VERBEXIT LOGDATA 关键字。
- 使用“转储分析菜单”上的选项 7（请参阅第 200 页的『使用 IBM MQ for z/OS 转储显示面板处理转储』）。

仅包含请求转储时存储中可用的记录。每条格式化的记录都位于标题 *****LOGDATA***** 之后。

z/OS 上的 SVC 转储

本主题可用于了解如何在 z/OS 上禁止 SVC 转储以及不生成 SVC 转储的原因。

不生成 SVC 转储时

在有些情况下，不生成 SVC 转储。通常，因为时间或空间问题或者安全性冲突，将禁止转储。以下列表总结了不生成 SVC 转储的其他原因：

- z/OS 可维护性级别指示处理 (SLIP) 命令禁止异常终止。
z/OS MVS Initialization and Tuning Reference 手册中 [IEACMD00](#) 的描述列出了在 IPL 时执行的 SLIP 命令的缺省值。
- 异常终止原因码是无需转储来确定异常终止原因的原因码。
- SDWACOMU 或 SDWAEAS（系统诊断工作区 (SDWA) 的一部分）用于禁止转储。

使用 z/OS DAE 禁止 IBM MQ for z/OS 转储

您可以禁止重复先前转储的 SVC 转储。*z/OS MVS Diagnosis: Tools and Service Aids* 手册提供了有关使用 z/OS 转储分析和消除 (DAE) 的详细信息。

为支持 DAE，IBM MQ for z/OS 定义了两个可变记录区域 (VRA) 关键字和最短症状字符串。两个 VRA 关键字为：

- KEY VRADAE (X'53')。没有任何数据与此关键字存在关联。
- KEY VRAMINSC (X'52') DATA (X'08')

IBM MQ for z/OS 为系统诊断工作区 (SDWA) 中的最短症状字符串提供了以下数据：

- 装入模块名称
- CSECT 名称
- 异常终止代码

- 恢复例程名称
- 失败的指令区域
- REG/PSW 差异
- 原因码
- 组件标识
- 组件子函数

如果九个症状中的八个症状（VRAMINSC 关键字中的 X'08'）相同，那么转储将被视为重复转储，以禁止重复转储。

z/OS 处理 z/OS 的性能问题

本主题可用于更加详细地调查 IBM MQ for z/OS 性能问题。

性能问题具有以下特征：

- 在线事务响应时间长
- 批处理作业需要较长时间才能完成
- 消息传输缓慢

性能问题可能是由多种因素导致的，包括从 z/OS 系统整体缺少资源到应用程序设计欠佳。

以下主题演示了问题和建议的解决方案，从相对易于诊断的问题（例如，DASD 争用）开始到特定子系统的问题（如 IBM MQ 和 CICS 或 IMS）。

- [第 216 页的『IBM MQ for z/OS 系统注意事项』](#)
- [第 216 页的『CICS 约束』](#)
- [第 217 页的『处理 z/OS 上运行缓慢或者已经停止的应用程序』](#)

远程排队问题可能是由于网络阻塞和其他网络问题所导致的。也可能是由于远程队列管理器的问题而导致的。

相关概念

[第 221 页的『处理 z/OS 上的不正确输出』](#)

错误输出包括缺失信息、意外信息或损坏的信息。阅读本主题以进行进一步调查。

相关任务

[第 6 页的『进行初始检查』](#)

您可以进行一些初始检查，这些检查可能会提供您可能遇到的常见问题的答案。

z/OS IBM MQ for z/OS 系统注意事项

在调查性能问题时，z/OS 系统处在需要进行检查的区域内。

您可能已经注意到自己的 z/OS 系统处于压力之下，因为这些问题影响了许多子系统和应用程序。

您可以使用标准监控工具（如，资源监控设施 (RMF)）来监控和诊断这些问题。这些问题可能包括：

- 存储受到约束（页面调度）
- 处理器周期受到约束
- DASD 受到约束
- 通道路径使用情况

请使用常规 z/OS 调优方法来解决这些问题。

z/OS CICS 约束

CICS 约束还可能对 IBM MQ for z/OS 性能产生负面影响。本主题可用于获取有关 CICS 约束的更多信息。

IBM MQ 任务的性能可能受到 CICS 约束的影响。例如，您的系统可能已达到 MAXTASK、强制事务等待或者 CICS 系统可能缺少存储空间。例如，CICS 可能无法调度事务，因为已达到并行任务最大数量，或者 CICS 检测到资源问题。如果您怀疑 CICS 会导致性能问题（例如，由于批处理作业和 TSO 作业已成功运

行，但是 CICS 任务超时或响应时间较长），请参阅 *CICS Problem Determination Guide* 和 *CICS Performance Guide*。

注：到瞬时数据分区外数据集的 CICS I/O 使用 z/OS RESERVE 命令。这可能影响到相同卷上的其他数据集的 I/O。

z/OS 处理 z/OS 上运行缓慢或者已经停止的应用程序

等待和循环可能表现出类似症状。使用本主题中的链接来帮助在 z/OS 上区分等待和循环。

等待和循环的特征是无响应。但可能难以区分等待、循环和性能欠佳。

以下任何症状都可能是由于等待、循环、错误调整的系统或系统超负荷而导致的：

- 显示为已停止运行的应用程序（如果 IBM MQ for z/OS 仍在响应，那么此问题可能是由应用程序问题导致的）
- MQSC 命令未产生任何响应
- 过度使用处理器时间

要执行这些主题中显示的测试，需要访问 z/OS 控制台并且能够发出操作员命令。

- [第 217 页的『在 z/OS 上区分等待和循环』](#)
- [第 218 页的『在 z/OS 上处理等待』](#)
- [第 220 页的『在 z/OS 上处理循环』](#)

相关任务

[第 6 页的『进行初始检查』](#)

您可以进行一些初始检查，这些检查可能会提供您可能遇到的常见问题的答案。

z/OS 在 z/OS 上区分等待和循环

IBM MQ for z/OS 上的等待和循环可能表现出类似症状。本主题可用于帮助确定遇到的是等待还是循环。

由于等待与循环难以区分，在某些情况下需要执行详细的调查，然后才能确定适合问题的分类。

本部分提供了有关选择最佳分类的指南以及确定分类之后需要采取的操作的建议。

等待

对于问题确定，等待状态被视为已暂挂执行任务的状态。即，任务已开始运行但是在未完成的情况下暂挂，并且随后无法恢复。

系统中被识别为等待的问题可能是由下列任一原因导致的：

- 等待 MQI 调用
- 等待 CICS 或 IMS 调用
- 等待其他资源（例如，文件 I/O）
- ECB 等待
- CICS 或 IMS 区域等待
- TSO 等待
- IBM MQ for z/OS 等待工作
- 明显等待，由循环引起
- 由于存在更高优先级的任务，CICS 或 MVS 未分派您的任务
- Db2 或 RRS 处于不活动状态

循环

循环是重复执行部分代码。如果未计划循环或者如果将其设计到您的应用程序中但是由于某些原因循环未能终止，那么根据代码执行的操作以及任何相连组件和产品如何对它作出反应，可获得一系列症状。在某些情

况下，最初循环可能被诊断为等待或性能问题，因为循环任务会与循环中不涉及的其他任务争用系统资源。但是，循环耗用资源，等待不耗用资源。

系统中的明显循环问题可能是由下列任一原因导致的：

- 应用程序执行的处理量多于往常，因此需要更长时间才能完成
- 应用程序逻辑中存在循环
- MQI 调用循环
- CICS 或 IMS 调用循环
- CICS 或 IMS 代码循环
- IBM MQ for z/OS 中存在循环

等待和循环的症状

以下任何症状都可能是由于等待、循环、错误调整的系统或系统超负荷而导致的：

- MQGET WAIT 超时
- 批处理作业暂挂
- TSO 会话暂挂
- CICS 任务暂挂
- 由于资源约束（例如，CICS MAX 任务）而未能启动事务
- 队列变满，并且未得到处理
- 系统命令未被接受，或者未产生任何响应

相关概念

第 218 页的『在 z/OS 上处理等待』

等待可能发生在批处理作业、TSO 应用程序、CICS 事务和 IBM MQ for z/OS 上的其他组件中。本主题可用于确定等待发生在何处。

第 220 页的『在 z/OS 上处理循环』

循环可能在 z/OS 系统的不同区域内发生。本主题可用于帮助确定发生循环的位置。

在 z/OS 上处理等待

等待可能发生在批处理作业、TSO 应用程序、CICS 事务和 IBM MQ for z/OS 上的其他组件中。本主题可用于确定等待发生在何处。

调查出现任务或子系统等待的问题时，需要考虑任务或子系统运行的环境。

z/OS 系统可能处于压力之下。在此情况下，会出现许多症状。如果实存储器不足，那么在页面调度中断或换出时作业可能会遇到等待。输入/输出 (I/O) 争用或通道使用率过高也可能导致等待。

您可以使用标准监控工具（如，资源监视设施 (RMF)）来诊断这些问题。请使用常规 z/OS 调优方法来解决这些问题。

批处理或 TSO 程序是否正在等待？

请考虑下列几点：

程序可能正在等待其他资源

例如，其他程序暂挂以等待更新的 VSAM 控制区间 (CI)。

程序可能正在等待尚未到达的消息

例如，如果服务器程序对队列进行持续性监控，那么此情况是正常行为。

或者，程序可能正在等待已到达但尚未落实的消息。

发出 DIS CONN(*) TYPE(HANDLE) 命令并检查程序正在使用的队列。

如果怀疑程序已发出不涉及 MQGET WAIT 的 MQI 调用，并且控制尚未从 IBM MQ 返回，请在取消批处理或 TSO 程序之前生成批处理或 TSO 作业与 IBM MQ 子系统的 SVC 转储。

同时，请考虑等待状态可能是由于其他程序问题而导致的，如异常终止（请参阅第 222 页的『z/OS 上消息未按预期到达』）或者 IBM MQ 本身的问题（请参阅第 219 页的『IBM MQ 是否正在等待 z/OS?』）。请参阅第 198 页的『IBM MQ for z/OS 转储』（尤其是第 199 页的图 41）以获取有关获取转储的信息。

如果问题仍然存在，请参阅第 227 页的『联系 IBM 支持人员』以获取有关向 IBM 报告问题的信息。

CICS 事务是否正在等待?

请考虑下列几点：

CICS 可能处于压力之下

这可能指示已达到允许的任务最大数量 (MAXTASK)，或者存在存储空间不足 (SOS) 的情况。请查看控制台日志，以获取可能解释此情况的消息（例如，SOS 消息），或者请参阅 *CICS Problem Determination Guide*。

事务可能正在等待其他资源

例如，这可能是文件 I/O。您可以使用 CEMT INQ TASK 来查看任务正在等待的内容。如果资源类型为 MQSERIES，那么您的事务正在等待 IBM MQ（位于 MQGET WAIT 中或任务切换中）。否则，请参阅 *CICS Problem Determination Guide* 以确定等待原因。

事务可能正在等待 IBM MQ for z/OS

例如，如果程序是服务器程序并且正在等待消息到达队列，那么这是正常情况。否则，这可能是由于事务异常终止而导致的，请参阅第 222 页的『z/OS 上消息未按预期到达』以获取示例。如果情况如此，在 CSMT 日志中报告了异常终止。

事务可能正在等待远程消息

如果使用分布式排队，那么程序可能正在等待尚未从远程系统交付的消息（有关更多信息，请参阅第 223 页的『在 z/OS 上使用分布式排队时缺少消息的问题』）。

如果您怀疑程序发出了不涉及 MQGET WAIT（即，它在任务开关中）的 MQI 调用，并且控制未从 IBM MQ 返回，请先执行 CICS 区域和 IBM MQ 子系统的 SVC 转储，然后再取消 CICS 事务。请参阅第 220 页的『在 z/OS 上处理循环』以获取有关等待的信息。请参阅第 198 页的『IBM MQ for z/OS 转储』（尤其是第 199 页的图 41）以获取有关获取转储的信息。

如果问题仍然存在，请参阅第 227 页的『联系 IBM 支持人员』以获取有关向 IBM 报告问题的信息。

Db2 是否正在等待?

如果调查表明 Db2 正在等待，请检查以下项：

1. 使用 Db2 -DISPLAY THREAD(*) 命令来确定在队列管理器与 Db2 子系统之间是否正在发生任何活动。
2. 尝试并确定任何等待是否发生在队列管理器子系统本地或者发生于 Db2 子系统之间。

RRS 是否处于活动状态?

- 使用 D RRS 命令来确定 RRS 是否处于活动状态。

IBM MQ 是否正在等待 z/OS?

如果调查表明 IBM MQ 本身正在等待，请检查以下项：

1. 使用 DISPLAY THREAD(*) 命令来检查是否与 IBM MQ 存在任何连接。
2. 使用 SDSF DA 或 z/OS 命令 DISPLAY A,xxxMSTR 来确定是否有任何处理器使用情况（如第 29 页的『应用程序或 IBM MQ for z/OS 是否已停止处理工作?』中所示）。
 - 如果 IBM MQ 正在使用部分处理器时间，请重新考虑 IBM MQ 可能正在等待的其他原因，或者考虑这是否确实是性能问题。
 - 如果没有处理器活动，请检查 IBM MQ 是否响应命令。如果可以获取响应，请重新考虑 IBM MQ 可能正在等待的其他原因。

- 如果无法获取响应，请检查控制台日志以获取可能解释此情况的消息（例如，IBM MQ 可能已没有活动的日志数据集，并且正在等待卸载处理）。

如果您确信 IBM MQ 已停滞，请在 QUIESCE 和 FORCE 方式下使用 STOP QMGR 命令来终止当前正在执行的任何程序。

如果 STOP QMGR 命令无法响应，请生成转储并取消队列管理器，然后重新启动。如果问题再次发生，请参阅第 227 页的『联系 IBM 支持人员』，以获取进一步的指导。

相关概念

第 217 页的『在 z/OS 上区分等待和循环』

IBM MQ for z/OS 上的等待和循环可能表现出类似症状。本主题可用于帮助确定遇到的是等待还是循环。

第 220 页的『在 z/OS 上处理循环』

循环可能在 z/OS 系统的不同区域内发生。本主题可用于帮助确定发生循环的位置。

在 z/OS 上处理循环

循环可能在 z/OS 系统的不同区域内发生。本主题可用于帮助确定发生循环的位置。

以下主题描述了您可能遇到的各种类型的循环，并提供了部分响应建议。

批处理应用程序是否正在循环？

如果您怀疑批处理应用程序或 TSO 应用程序正在循环，请使用控制台发出 z/OS 命令 DISPLAY JOBS,A（针对批处理应用程序）或 DISPLAY TS,A（针对 TSO 应用程序）。请注意来自显示的数据的 CT 值，并重复此命令。

如果任何任务显示 CT 值显著增加，那么任务可能正在循环。您还可以使用 SDSF DA 以显示每个地址空间正在使用的处理器的百分比。

批处理作业是否正在生成大量输出？

浏览队列并列显消息的应用程序可作为此行为的示例。如果浏览操作是使用 BROWSE FIRST 启动的，并且后续调用尚未重置为 BROWSE NEXT，那么应用程序会重复浏览并列显队列上的第一条消息。

如果怀疑正在运行的作业可能导致出现问题，那么可以使用 SDSF DA 来查看此作业的输出。

CICS 区域是否显示大量的处理器活动？

CICS 应用程序可能正在循环或者 CICS 区域本身可能正处于循环中。如果某个事务进入紧密（不间断）循环，那么可能会出现 AICA 异常终止。

如果您怀疑 CICS 或 CICS 应用程序正处于循环中，请参阅 *CICS Problem Determination Guide*。

IMS 区域是否显示大量处理器活动？

IMS 应用程序可能正处于循环中。如果您怀疑存在此行为，请参阅 *IMS Diagnosis Guide and Reference*。

队列管理器是否显示大量处理器活动？

尝试从控制台输入 MQSC DISPLAY 命令。如果无法获得任何响应，可能队列管理器正处于循环中。遵循第 29 页的『应用程序或 IBM MQ for z/OS 是否已停止处理工作？』中所示过程以显示有关队列管理器使用的处理器时间的信息。如果此命令表明队列管理器正处于循环中，请生成内存转储、取消队列管理器，然后重新启动。

如果问题仍然存在，请参阅第 227 页的『联系 IBM 支持人员』以获取有关向 IBM 报告问题的信息。

队列、页集或耦合设施结构是否意外写满？

如果是这样，那么表明应用程序正处于循环中并且正在将消息放置到队列上。（它可能是批处理、CICS 或 TSO 应用程序。）

识别处于循环中的应用程序

在繁忙的系统中，可能难以识别出导致问题的应用程序。如果保留着应用程序与队列的交叉引用，请终止可能将消息放置到队列上的任何程序或事务。在重新使用这些程序或事务之前对其进行调查。（最有可能的问题根源是新应用程序或更改后的应用程序；请检查更改日志以识别这些应用程序。）

尝试在队列上发出 DISPLAY QSTATUS 命令。此命令会返回有关队列的信息，这些信息可能有助于识别处于循环中的应用程序。

错误的触发定义

可能由于错误的对象定义（例如，队列可能设置为 NOTRIGGER）导致未触发获取应用程序。

分布式排队

使用分布式排队时，此问题的症状可能是接收系统中出现消息表明对死信队列的 MQPUT 调用失败。此问题可能是由于死信队列已写满而引起的。死信队列消息头（死信消息头结构）包含原因或反馈代码，解释为什么消息无法被放入目标队列。请参阅 [MQDLH - 死信消息头](#)，以获取有关死信消息头结构的信息。

将队列分配到页集

如果特定页集频繁写满，那么将队列分配到页集时可能出现这个问题。有关更多信息，请参阅 [IBM MQ for z/OS 性能约束](#)。

共享队列

耦合设施结构是否已满？z/OS 命令 DISPLAY CF 可显示有关耦合设施存储器的信息，包括总容量、正在使用的总量以及可用控制和非控制存储器总量。RMF 耦合设施使用情况摘要报告提供此信息的更持久副本。

任务和 IBM MQ for z/OS 是否显示大量处理器活动？

在此情况下，MQI 调用上的任务可能处于循环中（例如，重复浏览相同的消息）。

相关概念

第 217 页的『[在 z/OS 上区分等待和循环](#)』

IBM MQ for z/OS 上的等待和循环可能表现出类似症状。本主题可用于帮助确定遇到的是等待还是循环。

第 218 页的『[在 z/OS 上处理等待](#)』

等待可能发生在批处理作业、TSO 应用程序、CICS 事务和 IBM MQ for z/OS 上的其他组件中。本主题可用于确定等待发生在何处。

处理 z/OS 上的不正确输出

错误输出包括缺失信息、意外信息或损坏的信息。阅读本主题以进行进一步调查。

可以多种方式来解释术语“错误输出”，在 [第 36 页的『您得到了不正确的输出吗？』](#) 中解释了“错误输出”对于本产品文档中的问题确定的意义。

以下主题包含有关系统可能遇到的分类为错误输出的问题的信息。

- [期望到达而未到达的应用程序消息](#)
- [包含错误信息或者已损坏的信息的应用程序消息](#)

在此还描述了应用程序使用分布式队列时可能遇到的其他问题。

- [第 222 页的『z/OS 上消息未按预期到达』](#)
- [第 223 页的『在 z/OS 上使用分布式排队时缺少消息的问题』](#)
- [第 224 页的『在 z/OS 上使用消息分组时获取消息的问题』](#)
- [第 224 页的『在 z/OS 上查找发送给集群队列的消息』](#)
- [第 225 页的『查找发送给 IBM MQ - IMS 网桥的消息』](#)

- [第 225 页的『z/OS 上消息包含意外或损坏的信息』](#)

相关概念

[第 216 页的『处理 z/OS 的性能问题』](#)

本主题可用于更加详细地调查 IBM MQ for z/OS 性能问题。

相关任务

[第 6 页的『进行初始检查』](#)

您可以进行一些初始检查，这些检查可能会提供您可能遇到的常见问题的答案。

z/OS 上消息未按预期到达

缺失的消息可能具有不同的原因。本主题可用于进一步调查原因。

如果在期望消息到达时，消息未能到达，请检查以下项：

消息成功放入队列了吗？

IBM MQ 是否针对 MQPUT 发出返回码和原因码，例如：

- 队列正确定义了吗（例如，MAXMSGL 是否够大）？（原因码 2030）。
- 应用程序能否将消息放入队列中（为 MQPUT 调用启用的队列）？（原因码 2051）。
- 队列已经满了吗？这可能意味着应用程序无法将所需消息放入队列（原因码 2053）。

队列是共享队列吗？

- 在 CFRM 策略数据集中是否已成功定义了耦合设施结构？共享队列中保存的消息存储在耦合设施内。
- 是否已激活 CFRM 策略？

队列是集群队列吗？

如果是，那么不同队列管理器上可能有多个队列实例。这意味着消息可能位于不同队列管理器上。

- 是否希望消息进入集群队列？
- 应用程序是否设计为配合集群队列运行？
- 消息放入的队列实例是否不同于期望的队列实例？

检查任何集群工作负载出口程序，查看是否按期望的方式处理消息。

获取消息是否失败？

- 应用程序是否需要生成同步点？

如果在同步点中放入或获取消息，在落实恢复单元前，它们不可用于其他任务。

- MQGET 上的时间间隔足够长吗？

如果使用分布式处理，应考虑合理的网络延迟或远程端的问题。

- 您预期的消息是定义为持久的吗？

如果不是，并且已重新启动队列管理器，那么将删除此消息。共享队列是例外，因为非持久消息在队列管理器重新启动后仍将保留。

- 您在等由消息或相关标识标识的特定消息吗 (*MsgId* 或 *CorrelId*) ？

检查您是否在用正确的 *MsgId* 或 *CorrelId* 等待消息。成功的 MQGET 调用将这两个值都设置为获取的消息的值，因此，您可能需要复位这些值才能成功地获取另一个消息。

同样，检查能否从队列获取其他消息。

- 其他应用程序可以从队列取出消息吗？

如果是这样，那么其他应用程序是否已检索消息？

如果队列是共享队列，请检查其他队列管理器上的应用程序未获取消息。

如果您找不到队列有什么错误，并且队列管理器本身正在运行，那么在您预期将消息放入队列的过程中进行以下检查：

- 应用程序启动了吗？
如果应该已经触发了该应用程序，请检查是否指定了正确的触发器选项。
- 是否正在运行触发器监视器？
- 正确定义了触发器进程吗？（定义 IBM MQ for z/OS 和 CICS 或 IMS）？
- 它正确完成了吗？

在 CICS 日志中查找异常终止证据。

- 应用程序落实其更改了吗？或将它们回退了吗？

查找 CICS 日志中指示此情况的消息。

如果多个事务正在服务队列，那么它们偶尔可能彼此冲突。例如，一个事务可能发出带有 0 的缓冲区长度的 MQGET 调用来查找消息的长度，然后发出指定该消息的 *MsgId* 的特定 MQGET 调用。然而，与此同时，另一个事务为该消息发出了成功的 MQGET 调用，因此第一个应用程序接收了完成代码 MQRC_NO_MSG_AVAILABLE。必须将要在多服务器环境中运行的应用程序设计为可应付这样的情况。

系统是否遇到过中断？例如，如果期望的消息是由 CICS 应用程序放入队列的，并且 CICS 系统发生了中断，那么此消息可能难以确定情况。这意味着队列管理器不确定应落实还是回退此消息，因此将其锁定，直至发生再同步时解决此问题。

注：如果 CICS 决定将此消息回退，那么再同步后会将其删除。

并且试想：可能已接收到消息，但出于某种原因您的应用程序未能处理该消息。例如，是由于所要的消息格式出错而导致程序拒绝它吗？如果此时出现问题，请参阅第 225 页的『z/OS 上消息包含意外或损坏的信息』。

在 z/OS 上使用分布式排队时缺少消息的问题

本主题可用于了解在 IBM MQ for z/OS 上使用分布式排队时缺失消息的可能原因。

如果应用程序使用分布式排队，请考虑以下要点：

分布式排队已经被正确安装在发送和接收系统上了吗？

确保已正确遵循配置 z/OS 中有关安装分布式队列管理设施的指示信息。

在两个系统之间的链接是可用的吗？

检查两个系统是否都可用并连接到 IBM MQ for z/OS。检查两个系统之间的 LU 6.2 或 TCP/IP 连接是否处于活动状态，或者检查正在通信的任何其他系统上的连接定义。

请参阅[监控和性能](#)以获取有关网络中跟踪路由消息传递的更多信息。

通道是否正在运行？

- 针对传输队列发出以下命令：

```
DISPLAY QUEUE (qname) IPPROCS
```

如果 IPPROCS 的值为 0，这表示服务于此传输队列的通道未运行。

- 针对此通道发出以下命令：

```
DISPLAY CHSTATUS (channel-name) STATUS MSGS
```

使用此命令生成的输出来检查通道是否正在服务于正确的传输队列并且是否已连接到正确的目标机器和端口。您可以从 STATUS 字段中确定通道是否正在运行。您还可以通过检查 MSGS 字段来查看在通道上是否已发送任何消息。

如果通道处于 RETRYING 状态，那么这可能是由于另一端的问题所导致的。请检查通道启动程序和侦听器是否已启动，并且通道未停止。如果有人停止了此通道，那么您需要手动将其启动。

在发送系统中设置了触发吗？

检查通道启动程序是否正在运行。

传输队列是否已将触发设置为开启？

如果在特定情况下通道停止，那么可以针对传输队列将触发设置为关闭。

您所等待的消息是从远程系统来的应答消息吗？

检查远程系统定义（如前文所述），并检查远程系统中是否已激活触发。同时检查两个系统之间的 LU 6.2 连接不是单一会话（如果是，那么无法接收回复消息）。

检查远程队列管理器上队列是否存在、未满载并接受消息长度。如果不满足以上任一条件，那么远程队列管理器尝试将消息放入死信队列。如果消息长度超过通道允许的最大长度，那么发送队列管理器尝试将消息放入其死信队列。

队列已经满了吗？

这可能意味着应用程序无法将所需消息放入队列。如果情况如此，请检查消息是否已放入死信队列。

死信队列消息头（死信队列头结构）包含原因或反馈代码，解释为什么消息无法被放入目标队列。请参阅 [MQDLH - 死信消息头](#)，以获取有关死信消息头结构的更多信息。

在发送和接收队列管理器之间有不匹配吗？

例如，消息长度可能超过了接收队列管理器可处理范围的长度。检查控制台日志中的错误消息。

发送和接收通道的通道定义是兼容的吗？

例如，序号复位值不匹配导致通道停止。请参阅 [分布式排队和集群](#)。

是否正确执行了数据转换？

如果消息来自不同队列管理器，那么 CCSID 和编码是否相同，或者是否需要执行数据转换。

是否针对非持久性消息的快速交付对通道进行了定义？

如果定义的通道中已将 NPMSPEED 属性设置为 FAST（缺省设置），并且通道已由于某些原因而停止随后已重新启动，那么非持久性消息可能已丢失。请参阅 [非持久性消息速度 \(NPMSPEED\)](#)，以获取有关快速消息的更多信息。

通道出口是否导致以意外方式来处理消息？

例如，安全出口阻止通道启动，或者 MQXCC_CLOSE_CHANNEL 的 *ExitResponse* 可能终止通道。

在 z/OS 上使用消息分组时获取消息的问题

本主题可用于了解在 IBM MQ for z/OS 上使用消息分组时获取消息的一些问题。

应用程序是否正在等待一组完整消息？

确保组中所有消息都位于队列上。如果使用分布式排队，请参阅第 223 页的『[在 z/OS 上使用分布式排队时缺少消息的问题](#)』。确保组中最后一条消息在消息描述符中设置了相应的 *MsgFlags*，以指示它是最后一条消息。确保组中消息的消息到期设置为足够长的时间间隔，以确保在对消息进行检索之前，消息不会到期。

如果已检索来自该组的消息，并且 *get* 请求未按逻辑顺序排序，那么检索其他组消息时，请关闭该选项以等待完整的组。

如果应用程序针对完整的组按逻辑顺序发出一项 *get* 请求，在检索组的中途无法找到消息：

请确保没有任何其他应用程序正在针对队列运行并获取消息。确保组中消息的消息到期设置为足够长的时间间隔，以确保在对消息进行检索之前，消息不会到期。确保均未发出 *CLEAR QUEUE* 命令。可以通过按组标识获取消息来检索队列中的不完整的组，而无需指定逻辑顺序选项。

在 z/OS 上查找发送给集群队列的消息

本主题可用于了解在 IBM MQ for z/OS 上查找发送给集群队列的消息时所涉及的部分问题。

在使用这些主题中所描述的方法来查找未到达集群队列的消息之前，需要确定托管消息发送到的队列的队列管理器。可以通过以下方法来进行确定：

- 可以使用 *DISPLAY QUEUE* 命令来请求有关集群队列的信息。
- 可以使用 *MQPMO* 结构中返回的队列和队列管理器名称。

如果针对消息指定了 MQOO_BIND_ON_OPEN 选项，那么这些字段可提供消息目标。如果消息未与特定队列和队列管理器绑定，那么这些字段可提供消息发送到的第一个队列和队列管理器的名称。在此情况下，它可能不是消息的最终目标。

z/OS 查找发送给 **IBM MQ - IMS** 网桥的消息

本主题可用于了解缺失发送给 IBM MQ - IMS 网桥的消息的可能原因。

如果使用 IBM MQ - IMS 网桥，并且消息未按预期到达，那么请考虑以下问题：

IBM MQ - IMS 网桥是否正在运行？

针对网桥队列发出以下命令：

```
DISPLAY QSTATUS(qname) IPPROCS CURDEPTH
```

IPPROCS 的值应为 1；如果该值为 0，请检查以下内容：

- 此队列是网桥队列吗？
- IMS 是否正在运行？
- OTMA 是否已启动？
- IBM MQ 是否已连接到 OTMA？

注：存在两条 IBM MQ 消息可供您用于确定是否已连接到 OTMA。如果在任务的作业日志中存在消息 CSQ2010I，但是不存在消息 CSQ2011I，那么 IBM MQ 已连接到 OTMA。此消息还表明 OTMA 连接到哪个 IBM MQ 系统。有关这些消息内容的更多信息，请参阅 [IBM MQ for z/OS 消息代码、完成代码和原因码](#)。

在队列管理器内，存在一个处理每个 IMS 网桥队列的任务。此任务从队列取出消息，将请求发送到 IMS，然后执行落实。如果使用持久消息，那么落实需要磁盘 I/O，因此该过程耗时比非持久性消息更长。处理获取、发送和落实的时间限制了任务处理消息的速度。如果此任务可以满足工作负载的需求，那么当前深度接近零。如果发现当前深度经常大于零，那么可以通过使用两个队列而不是一个队列来增加吞吐量。

使用 IMS 命令 /DIS OTMA 来检查 OTMA 是否处于活动状态。

如果消息正在流向 IMS，请检查以下项：

- 使用 IMS 命令 /DIS TMEMBER client TPIPE ALL 来显示有关 IMS Tpipe 的信息。您可以从中确定每个 TPIPE 上排队和离队的消息数量。（落实方式 1 消息通常不在 TPIPE 上排队。）
- 使用 IMS 命令 /DIS A 来显示是否存在可用于在其中运行 IMS 事务的从属区域。
- 使用 IMS 命令 /DIS TRAN trancode 来显示排队等待事务的消息数。
- 使用 IMS 命令 /DIS PROG progname 来显示程序是否已停止。

回复消息是否已发送至正确位置？

发出以下命令：

```
DISPLAY QSTATUS(*) CURDEPTH
```

CURDEPTH 是否指示在您未预期到的队列上存在回复？

z/OS z/OS 上消息包含意外或损坏的信息

本主题可用于了解 z/OS 上可能导致意外或损坏的输出的部分问题。

如果在消息中包含的信息不是应用程序所需的，或出现某种程度的损坏，请考虑以下几点：

您的应用程序或将消息放入队列上的应用程序，被更改了吗？

确保在所有需要注意更改的系统上都同步反映了全部更改。

例如，格式化此消息的副本可能已经被更改，无论哪种情况，都必须重新编译应用程序来使更改生效。如果一个应用程序还没有被重新编译，那么将对其余应用程序来说显示为数据损坏。

检查确保没有任何外部数据源（例如，VSAM 数据集）发生更改。如果未完成任何必要的再编译，这同样可能导致数据失效。此外，请检查确保用于消息数据输入的任何 CICS 映射和 TSO 面板未发生任何更改。

应用程序对错误队列发送了消息吗？

检查确保应用程序正在接收的消息并非应该被发送到服务于另一个队列的应用程序。若有必要，请将安全性定义更改为阻止未授权的应用程序将消息放入错误的队列。

如果应用程序使用别名队列，请检查别名是否指向正确的队列。

如果更改了队列以使其成为集群队列，那么现在其中可能包含来自不同应用程序源的消息。

已经为该队列指定了正确的触发器信息吗？

检查应该已经启动了应用程序；或应该启动了不同的应用程序吗？

是否正确执行了数据转换？

如果消息来自不同队列管理器，那么 CCSID 和编码是否相同，或者是否需要执行数据转换。

检查 MQMD 结构的 *Format* 字段是否对应于消息内容。如果不对应，那么数据转换过程可能无法正确处理消息。

如果这些检查没有使您解决问题，那么为发送消息的程序和接收消息的程序检查应用程序逻辑。

处理捕获通道启动程序 (CHINIT) 的 SMF 数据时的问题

由于各种原因，可能不会捕获通道记帐和 CHINIT 统计信息 SMF 数据。

有关更多信息，请参阅：

相关概念

[通道启动程序的 SMF 记录布局](#)

通道记帐数据故障诊断

如果通道未在产生通道记帐 SMF 数据，请执行检查。

过程

1. 在队列管理器或通道级别检查是否已设置 STATCHL。
 - 在通道级别值为 OFF 意味着没有为此通道收集数据。
 - 在队列管理器级别值为 OFF 意味着没有为具有 STATCHL(QMGR) 的通道收集数据。
 - 值为 NONE（在队列管理器级别不适用）意味着没有为所有通道收集数据，无论其 STATCHL 设置如何。
2. 对于客户机通道，检查是否在队列管理器级别设置了 STATCHL。
3. 对于自动定义的集群发送方通道，检查是否设置了 STATACLS 队列管理器属性。
4. 发出 **DISPLAY TRACE** 命令。您需要启用 TRACE(A) CLASS(4) 来收集通道记帐数据。
5. 如果启用了跟踪，那么会在发生以下任何情况时将数据写入 SMF：
 - 按照时间间隔，根据 STATIME 队列管理器系统参数的值而定。值为零意味着使用 SMF 统计信息广播。使用 **DISPLAY SYSTEM** 命令来显示 STATIME 的值。
 - 发出 **SET SYSTEM** 命令以更改 STATIME 系统参数的值。
 - 关闭通道启动程序。
 - 发出 **STOP TRACE(A) CLASS(4)** 命令。
6. SMF 可能会将数据保留在内存中，然后再将其写出到 SMF 数据集或 SMF 结构。发出 MVS 命令 **D SMF,0** 并标注 MAXDORM 值。将采用格式 *mmss* 显示 MAXDORM 值，其中，*mm* 是时间（以分钟为单位），*ss* 是秒。SMF 在写出数据之前，可以将其在内存中保留时长 MAXDORM。

相关任务

[规划通道启动程序 SMF 数据](#)

[解释 IBM MQ 性能统计信息](#)

CHINIT 统计数据故障诊断

如果未在产生 CHINIT 统计信息 SMF 数据，请执行检查。

过程



1. 发出 **DISPLAY TRACE** 命令。您需要启用 TRACE(S) CLASS(4) 来收集通道启动程序统计 SMF 数据。
2. 如果启用了跟踪，那么会在发生以下任何情况时将数据写入 SMF：
 - 按照时间间隔，根据 STATIME 队列管理器系统参数的值而定。值为零意味着使用 SMF 统计信息广播。使用 **DISPLAY SYSTEM** 命令来显示 STATIME 的值。
 - 发出 **SET SYSTEM** 命令以更改 STATIME 系统参数的值。
 - 关闭通道启动程序。
 - 发出 **STOP TRACE(S) CLASS(4)** 命令。
3. SMF 可以将数据保留在内存中，然后再将其写出到 SMF 数据集或 SMF 结构。发出 MVS 命令 **D SMF,0** 并标注 MAXDORM 值。将采用格式 *mmss* 显示 MAXDORM 值，其中，*mm* 是时间（以分钟为单位），*ss* 是秒。SMF 在写出数据之前，可以将其在内存中保留时长 MAXDORM。

联系 IBM 支持人员

如果您在 IBM MQ 方面遇到问题需要帮助，可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订、故障诊断和其他新闻的通知。

关于此任务

IBM 支持站点中的 IBM MQ 支持页面包括：

-  [IBM MQ for Multiplatforms 支持 Web 页面](#)
-  [IBM MQ for z/OS 支持 Web 页面](#)

要接收有关 IBM MQ 修订、故障诊断和其他新闻的通知，您可以预订通知。

如果您无法自行解决问题并需要 IBM 支持人员的帮助，那么可以建立案例（请访问 <https://www.ibm.com/mysupport/s/createrecord/NewCase>）。

有关 IBM 支持的更多信息，包括如何注册获取支持，请参阅 [IBM Support Guide](#)。

注：运行 **runmqras** 命令将帮助您收集故障诊断信息，然后再将其发送给 IBM 支持人员。有关更多信息，请参阅 [runmqras \(收集 IBM MQ 故障诊断信息\)](#)。

为 IBM 支持人员收集故障诊断信息

当您使用 IBM 提交案例时，可以包含为帮助调查问题而收集的其他 IBM MQ 故障诊断信息 (MustGather 数据)。除了本部分中描述的信息外，IBM 支持人员可能会逐个案例地请求进一步的信息。

关于此任务

本部分说明如何收集 IBM MQ for [Multiplatforms](#) 或 IBM MQ for [z/OS](#) 可能迂到的多种不同类型问题的故障诊断信息。

在多平台上收集故障诊断信息

有关如何收集 IBM MQ on [Multiplatforms](#) 的故障诊断信息的概述。

关于此任务

注: 除了本部分中描述的信息外, IBM 支持人员可能会逐案例地请求进一步的信息。

过程

- 有关如何收集故障诊断信息并将其发送到 IBM 的常规信息, 请参阅:
 - [第 228 页的『使用 runmqras 自动收集故障诊断信息』](#)
 - [第 231 页的『手动收集故障诊断信息』](#)
 - [第 284 页的『将故障诊断信息发送到 IBM』](#)
- 有关如何收集 IBM MQ for Multiplatforms 的特定问题区域的故障诊断和诊断信息的信息, 请参阅:
 -  [Advanced Message Security \(AMS\)](#)
 - [C, C++, COBOL, .NET, pTAL, RPG 和 Visual Basic 客户机应用程序通道](#)
 - [IBM MQ 集群](#)
 - [数据转换](#)
 - [死信队列消息](#)
 - [错误消息和 FFST 文件](#)
 - [IBM WebSphere MQ File Transfer Edition \(FTE\): 请参阅 Managed File Transfer \(MFT\)](#)
 - [挂起和高 CPU 问题](#)
 - [IBM MQ Explorer](#)
 - [第 266 页的『收集 MQIPT 问题的信息』](#)
 - [安装和卸载](#)
 - [Java 和 JMS](#)
 - [日志记录和恢复](#)
 -  [Managed File Transfer](#)
 - [Microsoft Cluster Service](#)
 - [性能](#)
 - [发布/预订](#)
 -   [复制的数据队列管理器 \(RDQM\)](#)
 - [安全性](#)
 - [TLS 通道 \(以前称为 SSL\)](#)
 - [触发](#)
-  对于 IBM MQ Appliance, 请参阅 [收集 IBM MQ Appliance MustGather 数据以解决问题](#)。
- 有关所有其他问题, 请参阅 [收集 IBM MQ MustGather 数据, 以解决 Linux, UNIX, Windows 和 IBM i 上的所有其他问题](#)。

相关任务

[第 273 页的『在 z/OS 上收集故障诊断信息』](#)
有关如何收集 IBM MQ for z/OS 的故障诊断信息的概述。

使用 *runmqras* 自动收集故障诊断信息

如果需要将 IBM MQ 故障诊断信息发送给 IBM 支持人员, 那么可以使用 `runmqras` 命令将信息一起收集到单个归档中。

开始之前

runmqras 命令是用于收集 IBM MQ 故障诊断信息的 Java 应用程序。如果 IBM MQ 安装包含 Java JRE 组件，那么 **runmqras** 将使用该组件，否则请确保最近的 Java 运行时环境 (JRE) 位于 **PATH** 中，以避免发生以下错误：

AMQ8599E: runmqras 命令找不到 JRE

在启动 **runmqras** 之前，请确保已为 IBM MQ 安装设置环境。例如：

- Linux 在 UNIX and Linux 上：

```
sh> PATH="$PATH":/path/to/java/bin (only if needed)
sh> . /opt/mqm/bin/setmqenv -n Installation1
```

- Windows 在 Windows 上：

```
C:\> SET PATH=%PATH%;C:\path\to\java\bin; (only if needed)
C:\> C:\Program Files\IBM\MQ\bin\setmqenv -n Installation2
```

- IBM i 在 IBM i (Qshell) 上：

```
PATH="$PATH":/QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit (only if needed)
```

(可选) 您可以将 `/QIBM/ProdData/mqm/bin` 目录添加到 **PATH**，以便可以在不输入其完整路径的情况下使用 **runmqras**。要执行此操作，请在 Qshell 中输入下列其中一个命令，或者将其添加到主目录中的 `.profile` 文件，以便每次启动 Qshell 时它都将自动运行：

```
===> . /QIBM/ProdData/mqm/bin/setmqenv -s
```

如果无法使用 **runmqras** 工具自动收集信息 (例如，如果您正在运行较旧版本的 IBM MQ 或由于任何其他原因无法使用 **runmqras**)，那么可以改为手动收集信息，如第 231 页的『[手动收集故障诊断信息](#)』中所述。

提示：在使用 **runmqras** 之前，您可能希望清除 IBM MQ 文件以减少收集的数据量。有关更多信息，请参阅[清除 IBM MQ 文件](#)。

关于此任务

您可以使用 **runmqras** 命令将有关应用程序或 IBM MQ 故障的故障诊断信息收集到单个归档中，当您报告问题时，可以将该归档提交到 IBM。

缺省情况下，**runmqras** 会收集以下信息：

- IBM MQ FDC 文件。
- 错误日志 (来自所有队列管理器以及机器范围的 IBM MQ 错误日志)。
- 产品版本控制，状态信息以及各种其他操作系统命令的输出。

如果 IBM 支持人员要求您提供更详细的信息，那么可以通过使用 **-section** 参数指定必需选项来添加此信息。

过程

1. 要指定输出文件名以案例号开头，请使用 **-caseno** 参数。

例如：

- Linux 在 UNIX and Linux 上：

```
sh> runmqras -caseno TS123456789
```

- Windows 在 Windows 上：

```
C:\> runmqras -caseno TS123456789
```

- ▶ **IBM i** 在 IBM i (Qshell) 上:

```
/QIBM/ProdData/mqm/bin/runmqras -caseno TS123456789
```

如果您正在使用不支持 **-caseno** 参数的较低版本的产品, 请使用 **-zipfile** 选项而不是 **-caseno** 选项, 以使输出文件名以案例编号开头。

- ▶ **Linux** ▶ **UNIX** 在 UNIX and Linux 上:

```
sh> runmqras -zipfile TS123456789
```

- ▶ **Windows** 在 Windows 上:

```
C:\> runmqras -zipfile TS123456789
```

- ▶ **IBM i** 在 IBM i (Qshell) 上:

```
/QIBM/ProdData/mqm/bin/runmqras -zipfile TS123456789
```

2. 选择要为其收集数据的部分。

runmqras 命令使用名为 **isa.xml** 的配置文件, 该文件描述要收集的文件以及要运行的命令。此文件组织成多个部分, 用于确定解决不同类型的问题所需的信息, IBM 会根据需要添加新的部分。

要选择所需部分, 请指定带有相应选项的 **-section** 参数。例如:

- ▶ **Linux** ▶ **UNIX** 在 UNIX and Linux 上:

```
sh> runmqras -caseno TS123456789 -section defs,cluster,trace
```

- ▶ **Windows** 在 Windows 上:

```
C:\> runmqras -caseno TS123456789 -section defs,cluster,trace
```

- ▶ **IBM i** 在 IBM i (Qshell) 上:

```
/QIBM/ProdData/mqm/bin/runmqras -caseno TS123456789 -section defs,cluster,trace
```

3. 选择要为其收集数据的队列管理器。

缺省情况下, **runmqras** 命令尝试收集有关所有队列管理器的信息。使用 **-qmlist** 选项提供当前安装中 **runmqras** 应检查的队列管理器的逗号分隔列表。例如:

- ▶ **Linux** ▶ **UNIX** 在 UNIX and Linux 上:

```
sh> runmqras -caseno TS123456789 -section defs,cluster,trace -qmlist QMA,QMB,QMC
```

- ▶ **Windows** 在 Windows 上:

```
C:\> runmqras -caseno TS123456789 -section defs,cluster,trace -qmlist QMA,QMB,QMC
```

- ▶ **IBM i** 在 IBM i (Qshell) 上:

```
==== /QIBM/ProdData/mqm/bin/runmqras -caseno TS123456789 -section defs,cluster,trace  
-qmlist QMA,QMB,QMC
```

要点: 请勿在 IBM MQ 客户机安装上使用 **-qmlist** 选项。

如果您有多个 IBM MQ 安装, 请不要从一个安装使用 **runmqras** 命令来收集有关另一个安装中的队列管理器的信息。虽然 **runmqras** 命令不会直接失败, 但 **runmqras** 发出的某些命令将失败并产生以下错误:

AMQ6292: 队列管理器与其他安装相关联

而是首先使用 **setmqenv** 命令在安装之间进行切换。然后，在每个安装中，使用 **runmqras** 命令的 **-qmlist** 选项从与该安装关联的队列管理器收集信息。

您选择的队列管理器应该正在运行，否则 **runmqras** 命令发出的某些命令将失败并返回错误

AMQ8146: IBM MQ 队列管理器不可用

但是，如果您有无法启动的队列管理器，那么 **runmqras** 命令仍很有用。

4. 请选择其他目录以处理大型文件。

如果系统有大量要收集的 FDC 或跟踪文件，或者如果您收集 **all** 或 **QMGR** 部分，那么 **runmqras** 命令创建的归档可能非常大。通常，**runmqras** 使用临时目录中的空间来收集和压缩文件。要在具有更多可用空间的文件系统或磁盘上选择其他目录，请使用 **-workdirectory** 选项。指定的目录必须为空。如果它尚不存在，那么 **runmqras** 将创建它。例如：

- Linux UNIX 在 UNIX and Linux 上:

```
sh> runmqras -caseno TS123456789 -section defs,cluster,trace,QMGR -qmlist QMA,QMB,QMC  
-workdirectory /var/bigdata/2019-07-27
```

- Windows 在 Windows 上:

```
C:\> runmqras -caseno TS123456789 -section defs,cluster,trace,QMGR -qmlist QMA,QMB,QMC  
-workdirectory G:\BigData\2019-07-27
```

- IBM i 在 IBM i (Qshell) 上:

```
====> /QIBM/ProdData/mqm/bin/runmqras -caseno TS123456789 -section defs,cluster,trace,QMGR  
-qmlist QMA,QMB,QMC -workdirectory /QIBM/bigdata/2019-07-27
```

5. 将收集的故障诊断信息发送给 IBM 支持人员。

确保 **runmqras** 归档文件以 IBM 案例编号开头，例如 **TS123456789-runmqras.zip**，然后将该文件发送到 IBM。有关更多信息，请参阅第 284 页的『将故障诊断信息发送到 IBM』。

下一步做什么

要点: 将 **runmqras** 归档文件发送到 IBM 后，请保留该文件的副本，直到您的问题得到解决，并且您已测试该解决方案以使您满意为止。

runmqras 命令不会从系统中删除任何文件，既不会删除 IBM MQ 日志，也不会删除 FDC，作业日志，转储或跟踪文件。使用 **runmqras** 收集这些文件后，请考虑将其归档或删除，如清除 IBM MQ 文件中所述。如果您需要稍后再次使用 **runmqras** 收集故障诊断信息，那么新的 **runmqras** 文件将更小，更易于分析，因为它不包含重复文件和旧信息。

相关任务

第 231 页的『手动收集故障诊断信息』

在某些情况下，您可能需要手动收集故障诊断信息，例如，如果您正在运行较旧版本的 IBM MQ，或者无法使用 **runmqras** 命令自动收集故障诊断信息。

第 284 页的『将故障诊断信息发送到 IBM』

生成并收集问题的故障诊断信息后，可以将其发送到 IBM 以帮助确定支持案例的问题。

Multi 手动收集故障诊断信息

在某些情况下，您可能需要手动收集故障诊断信息，例如，如果您正在运行较旧版本的 IBM MQ，或者无法使用 **runmqras** 命令自动收集故障诊断信息。

关于此任务

如果需要为 IBM 支持人员收集故障诊断信息，那么在大多数情况下应使用 **runmqras** 工具，该工具会自动执行收集故障诊断信息的任务，而不是手动收集此信息。

如果您无法使用 **runmqras** 工具自动收集信息 (例如，如果您正在运行较旧版本的 IBM MQ 或由于任何其他原因无法使用 **runmqras**)，那么将提供这些手动指示信息供您使用。

提示: 在打包数据之前, 请考虑清除 IBM MQ 文件, 以减小数据大小并加快将其传输到 IBM 的速度。有关更多信息, 请参阅 [清除 IBM MQ 文件](#)。

过程

1. 如果系统具有多个 IBM MQ 安装, 请先使用 **setmqenv** 命令选择具有问题的安装, 然后再继续:

- ▶ **Linux** ▶ **UNIX** 在 UNIX and Linux 上:

```
sh> ./path/to/mqm/bin/setmqenv -n InstallationX
```

- ▶ **Windows** 在 Windows 上:

```
C:\> "C:\Program Files\IBM\MQ\bin\setmqenv" -n InstallationX
```

2. 记录 IBM MQ 版本和维护级别。

您可以使用 **dspmqr** 命令来显示这些详细信息。有关更多信息, 请参阅 [显示 IBM MQ 版本](#)。如果要收集 [AMS](#), [通道](#), [数据转换](#), [死信队列](#), [错误消息和 FFST](#), [安全性或 TLS 通道](#) 问题的故障诊断信息, 请记录通道两侧的版本和维护级别。或者, 在通道两侧手动收集 IBM MQ 数据。

3. 记录操作系统版本和维护级别。

如果要收集 [AMS](#), [通道](#), [数据转换](#), [死信队列](#), [错误消息和 FFST](#), [安全性或 TLS 通道](#) 问题的故障诊断信息, 请针对通道两侧记录此信息。

4. 如果要收集 [AMS](#), [通道](#), [数据转换](#), [死信队列](#), [错误消息和 FFST](#), [安全性或 TLS 通道](#) 问题的故障诊断信息, 请记录通道两侧系统的 IP 地址和主机名。

5. 保存 IBM MQ 配置信息, 例如注册表键和 .ini 文件。

6. 如果系统具有多个 IBM MQ 安装, 请使用 **dspmqinst** 命令来记录 IBM MQ 安装详细信息:

- ▶ **Linux** ▶ **UNIX** 在 UNIX and Linux 上:

```
sh> dspmqinst > /tmp/dspmqinst.txt
```

- ▶ **Windows** 在 Windows 上:

```
C:\>dspmqinst > %TEMP%/dspmqinst.txt
```

7. 在 IBM MQ 服务器安装上, 使用 **dspmq** 命令来记录队列管理器的状态。

此步骤不适用于 [挂起和高 CPU](#), [发布/预订或触发](#) 问题。

- ▶ **Linux** ▶ **UNIX** 在 UNIX and Linux 上:

```
sh> dspmq -a > /tmp/dspmq.txt
```

- ▶ **Windows** 在 Windows 上:

```
C:\> dspmq -a > %TEMP%/dspmq.txt
```

- ▶ **IBM i** 在 IBM i (命令行) 上:

```
====> WRKMQM
```

- ▶ **IBM i** 在 IBM i (Qshell) 上:

```
====> /QSYS.LIB/QMQM.LIB/DSPMQ.PGM -a > /tmp/dspmq.txt
```

8. 在 IBM MQ 服务器安装上, 记录系统上处于活动状态的 IBM MQ 进程。

此步骤不适用于 [触发](#) 问题。

- **Linux** **UNIX** 在 UNIX and Linux 上:

```
sh> ps -ef | grep mq > /tmp/ps.txt
```

- **Windows** 在 Windows 上:

```
C:\> TASKLIST /V > %TEMP%/tasklist.txt
```

- **IBM i** 在 IBM i (命令行) 上:

```
====> WRKACTJOB SBS(QMQM)
```

- **IBM i** 在 IBM i (Qshell) 上:

```
====> ps -ef | grep mq > /tmp/ps.txt
```

9. **ULW**

仅对于 [日志记录或恢复](#) 问题, 请收集以下故障诊断信息:

a) **ULW**

在 UNIX, Linux, and Windows 上, 列出队列管理器 LogPath 目录的内容。

例如:

- **Linux** **UNIX** 在 UNIX and Linux 上:

```
sh> ls -ltr /var/mqm/log/QMA > /tmp/QMA.logfiles.txt
```

- **Windows** 在 Windows 上:

```
C:\> DIR /s "C:\ProgramData\IBM\MQ\log\QMA" > %TEMP%/QMA.logfiles.txt
```

b) **ULW**

在 UNIX, Linux, and Windows 上, 确保保存日志的文件系统或磁盘未滿。

例如:

- **Linux** **UNIX** 在 UNIX and Linux 上:

```
sh> df -k > /tmp/filesystems.txt
```

- **Windows** 在 Windows 上:

```
C:\> DIR C: > %TEMP%/diskusage.txt
```

c) **ULW**

在 UNIX, Linux, and Windows 上, 对队列管理器运行 **amqldmpa** 程序以收集有关记录器的详细信息。

该命令必须由 IBM MQ 管理员运行, 并且输出文件应该位于队列管理器有权写入的位置。例如:

- **Linux** **UNIX** 在 UNIX and Linux 上:

```
sh> amqldmpa -m QMA -c H -d 8 -f /tmp/QMA.amqldmpa.logger.txt
```

- **Windows** 在 Windows 上:

```
C:\> amqldmpa -m QMA -c H -d 8 -f %TEMP%\QMA.amqldmpa.logger.txt
```

- d) 在所有系统上, 针对队列管理器运行 **amqldmpa** 程序以收集有关持久性层的详细信息。

该命令必须由 IBM MQ 管理员运行, 并且输出文件应该位于队列管理器有权写入的位置。例如:

- **Linux** **UNIX** 在 UNIX and Linux 上:

```
sh> amqldmpa -m QMA -c A -d 8 -f /tmp/QMA.amqldmpa.dap.txt
```

- **Windows** 在 Windows 上:

```
C:\> amqldmpa -m QMA -c A -d 8 -f %TEMP%\QMA.amqldmpa.dap.txt
```

- **IBM i** 在 IBM i (Qshell) 上:

```
====> /QSYS.LIB/QMQM.LIB/AMQLDMPA.PGM -m QMA -c A -d 8 -f /tmp/QMA.amqldmpa.dap.txt
```

e) **ULW**

在 UNIX, Linux, and Windows 上, 收集在队列管理器的 LogPath 的活动子目录中找到的日志文件头 amqhlctl.lfh。

例如:

- **Linux** **UNIX** 在 UNIX and Linux 上:

```
/var/mqm/log/QMA/active/amqhlctl.lfh
```

- **Windows** 在 Windows 上:

```
C:\ProgramData\IBM\MQ\Log\QMA\active\amqhlctl.lfh
```

f) **IBM i**

在 IBM i 上, 从队列管理器的 `qm.ini` 文件中找到 **Library** 属性并显示其内容。有关队列管理器库的更多信息, 请参阅 **IBM i** 上的对象名。

或者, 显示库 **QM*** 并从列表中选择队列管理器以显示其内容。

- 要显示给定队列管理器的库, 例如 QMA:

```
====> WRKLIB LIB(QMQMA)
```

- 要显示所有队列管理器的库:

```
====> WRKLIB LIB(QM*)
```

g) **IBM i**

在 IBM i 上, 使用相同的 **Library** 值来处理队列管理器的日志。

保存输出, 然后使用 F17 显示连接的日志接收器并同时保存这些屏幕的输出。例如, 要显示队列管理器 QMA 的日志和日志接收器:

```
====> WRKJRNA JRN(QMQMA/AMQAJRN)
```

10. 在 IBM MQ 服务器安装上, 使用 **dmpmqcfg** 命令来记录队列管理器配置:

此步骤不适用于 日志记录或恢复 问题。

- **Linux** **UNIX** 在 UNIX and Linux 上:

```
sh> dmpmqcfg -m QMA >/tmp/QMA.config.txt
```

- **Windows** 在 Windows 上:

```
C:\> dmpmqcfg -mQMA>%TEMP%\QMA.config.txt
```

- **IBM i** 在 IBM i (Qshell) 上:

```
====> /QSYS.LIB/QMQM.LIB/DMPMQCFG.PGM -mQMA > /tmp/QMA.config.txt
```

11. 在 IBM MQ 服务器安装上, 使用 **runmqsc** 命令从队列管理器记录状态信息。有关更多信息, 请参阅 [保存 IBM MQ MQSC 输出](#)。

此步骤不适用于 [日志记录或恢复](#) 问题。

如果任何命令返回错误, 请继续执行其他命令:

```
DISPLAY PUBSUB ALL
DISPLAY QMSTATUS ALL
DISPLAY CHSTATUS(*) ALL
DISPLAY LSSTATUS(*) ALL
DISPLAY SVSTATUS(*) ALL
DISPLAY SBSTATUS(*) ALL
DISPLAY CONN(*) TYPE(*) ALL
DISPLAY QSTATUS(*) TYPE(Queue) ALL
DISPLAY QSTATUS(*) TYPE(HANDLE) ALL
DISPLAY TPSTATUS('#') TYPE(PUB) ALL
DISPLAY TPSTATUS('#') TYPE(SUB) ALL
DISPLAY TPSTATUS('#') TYPE(TOPIC) ALL
```

12. 仅针对 [IBM MQ 集群](#) 或 [挂起和高 CPU](#) 问题, 记录有关队列管理器已知的集群对象的信息。

对于 [IBM MQ 集群](#) 问题, 还要转储集群存储库高速缓存的内容。

- a) 使用 **runmqsc** 命令可记录有关队列管理器已知的集群对象的信息。

如果任何命令返回错误, 请继续执行其他命令:

```
DISPLAY CLUSQMGR(*) ALL
DISPLAY QCLUSTER(*) ALL
DISPLAY TCLUSTER(*) ALL
```

- b) 使用 **amqrfdm** 实用程序转储集群存储库高速缓存的内容。

确保对您的平台使用正确的输入文件。例如:

- Linux 要在 UNIX and Linux 上转储队列管理器 QMA 的集群存储库高速缓存, 请执行以下操作:

```
sh> amqrfdm -m QMA < cluster-unix.txt > /tmp/QMA.cluster.txt
```

- Windows 要在 Windows 上转储队列管理器 QMA 的集群存储库高速缓存, 请执行以下操作:

```
C:\> amqrfdm -m QMA < %TEMP%\cluster-win.txt > %TEMP%\QMA.cluster.txt
```

- IBM i 要在 IBM i (Qshell) 上转储队列管理器 QMA 的集群存储库高速缓存, 请执行以下操作:

```
===> /QSYS.LIB/QMQM.LIB/AMQRFDM.PGM -m QMA < cluster-IBMi.txt > /tmp/QMA.cluster.txt
```

13. 仅对于 [发布/预订](#) 问题, 请完成以下步骤:

- a) 在所有系统上, 针对队列管理器运行 **amqldmpa** 程序以收集有关主题的详细信息。

该命令必须由 IBM MQ 管理员运行, 并且输出文件应该位于队列管理器有权写入的位置。例如:

- Linux 在 UNIX and Linux 上:

```
sh> amqldmpa -m QMA -c T -d 8 -f /tmp/QMA.amqldmpa.topic.txt
```

- Windows 在 Windows 上:

```
C:\> amqldmpa -m QMA -c T -d 8 -f %TEMP%\QMA.amqldmpa.topic.txt
```

- IBM i 在 IBM i (Qshell) 上:

```
===> /QSYS.LIB/QMQM.LIB/AMQLDMPA.PGM -m QMA -c T -d 8 -f /tmp/QMA.amqldmpa.topic.txt
```

- b) 如果您的系统已启用已排队的发布/预订, 请使用类似于 **amqsbcbg** 样本的程序来浏览发布预订系统队列。

例如：

```
amqsbcg SYSTEM.PENDING.DATA.QUEUE QMA > QMA.PENDING.DATA.browse.txt
amqsbcg SYSTEM.JMS.ND.SUBSCRIBER.QUEUE QMA > QMA.JMS.ND.SUB.browse.txt
amqsbcg SYSTEM.JMS.ND.CC.SUBSCRIBER.QUEUE QMA > QMA.JMS.ND.CC.SUB.browse.txt
amqsbcg SYSTEM.JMS.D.SUBSCRIBER.QUEUE QMA > QMA.JMS.D.SUB.browse.txt
amqsbcg SYSTEM.JMS.D.CC.SUBSCRIBER.QUEUE QMA > QMA.JMS.D.CC.SUB.browse.txt
```

14. 对于通道或客户机应用程序连接困难的问题，请使用操作系统工具在连接尝试之前和之后立即列出双方的网络连接。

此步骤适用于收集以下类型问题的故障诊断信息：[AMS](#)，[通道](#)，[客户机应用程序](#)，[数据转换](#)，[死信队列](#)，[错误消息](#)和 [FFST](#)，[Java](#) 和 [JMS](#)，[安全性](#)或 [TLS 通道](#)。

- Linux UNIX 要在 UNIX and Linux 上显示网络连接：

```
sh> netstat -an
```

- Windows 要在 Windows 上显示网络连接：

```
C:\>NETSTAT -AN
```

- IBM i 要在 IBM i 命令行上显示 IPv4 和 IPv6 网络连接：

```
====> NETSTAT OPTION(*CNN)
====> NETSTAT OPTION(*CNN6)
```

15. 手动打包 IBM 的文件：

- Linux UNIX [第 236 页的『手动打包 UNIX 和 Linux 上的信息』](#)
- Windows [第 237 页的『手动打包 Windows 上的信息』](#)
- IBM i [第 238 页的『手动打包 IBM i 上的信息』](#)

相关任务

[第 228 页的『使用 runmqras 自动收集故障诊断信息』](#)

如果需要将 IBM MQ 故障诊断信息发送给 IBM 支持人员，那么可以使用 **runmqras** 命令将信息一起收集到单个归档中。

[第 284 页的『将故障诊断信息发送到 IBM』](#)

生成并收集问题的故障诊断信息后，可以将其发送到 IBM 以帮助确定支持案例的问题。

Linux UNIX 手动打包 UNIX 和 Linux 上的信息

在 UNIX 和 Linux 上，您首先选择一个具有足够可用空间的目录，以存放需要收集的所有数据。然后，将所需文件添加到名称以 IBM 案例号开头的压缩文件。

过程

1. 查找具有足够可用空间来存放所有 IBM MQ 数据的目录。

`/var/mqm/errors` 和 `/var/mqm/trace` 目录的内容通常构成大部分 IBM MQ 数据，因此请使用 **du** (磁盘使用情况) 和 **df** (显示文件系统) 命令对照文件系统中的可用空间检查这些目录的磁盘使用情况。例如：

```
sh> du -sk /var/mqm/errors /var/mqm/trace
384      /var/mqm/errors
189496   /var/mqm/trace

sh> df -k
Filesystem      1024-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4         393216         256536   35%      8641    12% /
/dev/hd2        8257536        1072040  88%     70803   21% /usr
/dev/hd9var     393216         126792   68%      6694   16% /var
/dev/hd3       12582912       12441980 99%     5108    2% /tmp
/dev/hd1       1310720         162560   88%       439    2% /home
/proc            -              -        -         -       - /proc
```

```
/dev/hd10opt      7208960    97180    99%    64796    65% /opt
/dev/fslv00      16777216  15405312    9%    12415    1% /var/mqm
```

2. 在您选择的目录中，创建名称以 IBM 大小写编号开头的新 tar 文件，并将 IBM MQ errors 目录的内容添加到该文件中。

例如：

```
sh> tar -cf /tmp/TS001234567-mqdata.tar /var/mqm/errors
```

3. 将 IBM MQ 配置文件添加到 tar 文件。仅当在系统上安装了 IBM WebSphere MQ 7.1 或更高版本时，才包含 mqinst.ini 文件：

```
sh> tar -uf /tmp/TS001234567-mqdata.tar /var/mqm/mqs.ini /etc/opt/mqm/mqinst.ini
```

4. 为队列管理器添加 IBM MQ 配置文件和错误日志。

例如：

```
sh> tar -uf /tmp/TS001234567-mqdata.tar /var/mqm/qmgrs/QMA/qm.ini /var/mqm/qmgrs/QMA/errors/*
.LOG
```

5. 添加任何其他文件，如第 227 页的『在多平台上收集故障诊断信息』中所示，并根据 IBM 支持人员的请求，包括包含来自 IBM MQ 和系统命令的输出的文件。

例如：

```
sh> tar -uf /tmp/TS001234567-mqdata.tar /tmp/ps.txt /tmp/ipcs.txt /tmp/mqconfig.txt
```

6. 如果收集了 IBM MQ 跟踪，请最后添加跟踪文件：

```
sh> tar -uf /tmp/TS001234567-mqdata.tar /var/mqm/trace
```

7. 使用系统上的任何可用压缩工具来压缩 tar 文件。

例如：

- 使用 **compress**：创建 .tar.Z 文件

```
sh> compress /tmp/TS001234567-mqdata.tar
```

- 使用 **gzip**：创建 .tar.gz 文件

```
sh> gzip /tmp/TS001234567-mqdata.tar
```

- 使用 **bzip2**：创建 .tar.bz2 文件

```
sh> bzip2 /tmp/TS001234567-mqdata.tar
```

8. 将数据发送到 IBM 后，如第 284 页的『将故障诊断信息发送到 IBM』中所述，获取要保留的文件备份副本，直到您的案例得到解决，然后从系统中删除该文件以节省空间。

```
sh> rm /tmp/TS001234567-mqdata.*
```

Windows 手动打包 Windows 上的信息

在 Windows 上，您首先选择要在其中打包 IBM MQ 文件的目录。然后，将所需文件添加到名称以 IBM 案例编号开头的压缩文件夹。

关于此任务

有许多第三方实用程序用于在 Windows 上创建归档。如果您愿意，请随意使用其中一个，但请确保在文件名的开头包含案例编号，例如 TS001234567-mqdata.zip。以下指示信息演示如何仅使用 Windows 的功能来打包文件。

过程

1. 打开 Windows Explorer 并浏览到将在其中打包 IBM MQ 文件的目录。

例如，如果要在个人临时目录中执行此操作，可以在 Windows Explorer 位置栏中输入 %TEMP%。

在目录中右键单击，然后选择 **新建 > 压缩 (压缩) 文件夹**。在文件名的开头包含案例编号，例如 TS001234567-mqdata。Windows 会自动添加 .zip 扩展。

2. 打开另一个 "Windows 资源管理器" 窗口，并使用它来查找您希望包含的 Windows 目录和文件。大多数 IBM MQ 文件将位于 "WorkPath" 注册表键所标识的目录下。要确定此目录，请使用 Windows 随附的 `amquregn` 程序，并忽略其返回的路径中的双反斜杠字符：

```
C:\Program Files\IBM\MQ\bin> amquregn amquregn.ctl | FINDSTR WorkPath
.. "WorkPath"="C:\\ProgramData\\IBM\\MQ"
.... "WorkPath"="C:\\ProgramData\\IBM\\MQ"
```

如果系统包含 IBM MQ 8.0 的新安装，那么 WorkPath 可能指向 C:\ProgramData 下的目录，而不是 C:\Program Files (x86)。缺省情况下，Windows 会隐藏 C:\ProgramData 目录，因此您必须在 Windows Explorer 位置栏中输入 %PROGRAMDATA% 以浏览至该目录。或者，您可以在 "控制面板" 中修改个人设置，以便 Windows Explorer 将显示隐藏文件。

3. 通过将目录或文件拖到新的压缩文件夹之上来添加该目录或文件。首先包含顶级 IBM MQ errors 目录。
4. 如果系统仅安装了 IBM WebSphere MQ 7.1 或更高版本，请将 IBM MQ .ini 文件拖至压缩文件夹。
5. 将队列管理器的 IBM MQ 配置文件和错误日志拖到压缩文件夹中。
6. 添加任何其他文件，如第 227 页的『在多平台上收集故障诊断信息』中所示，并根据 IBM 支持人员的请求，包括包含来自 IBM MQ 和系统命令的输出的文件。
7. 如果收集了 IBM MQ 跟踪，请最后添加跟踪文件。
8. 将数据发送到 IBM (如第 284 页的『将故障诊断信息发送到 IBM』中所述) 后，请获取要保留的文件备份副本，直到您的案例得到解决，然后使用 Windows 资源管理器从系统中删除该文件以节省空间。

IBM i 手动打包 IBM i 上的信息

在 IBM i 上，通过在 IBM i 命令行上运行命令来打包 IBM MQ 文件。您需要在每个保存文件名的开头包含 IBM 案例编号。

过程

1. 创建包含顶级 IBM MQ 配置文件和错误目录的保存文件，其中可能包括 IBM MQ FFST 文件，错误日志和作业文件：

```
====> CRTSAVF FILE(QGPL/P12345A) TEXT('Top-level files for PMR 12345,67R,890')
====> SAV DEV('/QSYS.LIB/QGPL.LIB/P12345A.FILE') OBJ((' /QIBM/UserData/mqm/*.ini' *INCLUDE) (' /QIBM/UserData/mqm/errors/*' *INCLUDE)) DTACPR(*MEDIUM)
```

2. 创建一个保存文件，其中包含问题中涉及的任何队列管理器的 qm.ini 文件和错误日志。

例如：

```
====> CRTSAVF FILE(QGPL/P12345B) TEXT('QMB files for PMR 12345,67R,890')
====> SAV DEV('/QSYS.LIB/QGPL.LIB/P12345B.FILE') OBJ((' /QIBM/UserData/mqm/qmgrs/QMB/qm.ini' *INCLUDE) (' /QIBM/UserData/mqm/qmgrs/QMB/errors/*' *INCLUDE))
====> CRTSAVF FILE(QGPL/P12345C) TEXT('QMC files for PMR 12345,67R,890')
====> SAV DEV('/QSYS.LIB/QGPL.LIB/P12345C.FILE') OBJ((' /QIBM/UserData/mqm/qmgrs/QMC/qm.ini' *INCLUDE) (' /QIBM/UserData/mqm/qmgrs/QMC/errors/*' *INCLUDE))
```

3. 创建包含系统历史记录日志的保存文件：

- a) 首先，创建数据库文件：

```
====> CRTPF FILE(QGPL/QHIST) RCDLEN(132) MAXMBRS(*NOMAX) SIZE(10000 1000 100)
```

- b) 显示要显示的时间段的系统历史记录日志。例如：

```
====> DSPLOG PERIOD(('12:00:00' '05/16/2014') ('23:59:59' '05/30/2014')) OUTPUT(*PRINT)
```

- c) 使用假脱机文件来查找 QPDSPLG 历史记录日志信息：

```
====> WRKSPLF
```

d) 将历史记录日志假脱机文件复制到数据库文件中。

例如:

```
====> CPYSPLF FILE(QPDSPL0G) TOFILE(QGPL/QHIST) TOMBR(HISTORY)
```

e) 创建保存文件并将数据库文件保存到该文件中:

```
====> CRTSAVF FILE(QGPL/P12345H) TEXT('History log for PMR 12345,67R,890')
====> SAVOBJ OBJ(QHIST) LIB(QGPL) DEV(*SAVF) SAVF(QGPL/P12345H)
```

4. 创建包含 IBM MQ 作业日志的保存文件:

a) 首先, 创建数据库文件:

```
====> CRTPF FILE(QGPL/JOBLOGS) RCDLEN(132) MAXMBRS(*NOMAX) SIZE(10000 1000 100)
```

b) 使用 QMQM 假脱机文件, 然后按 F11 两次以获取作业日志信息 (按屏幕上的顺序列出的文件 Nbr, 作业, 用户和编号):

```
====> WRKSPLF SELECT(QMQM)
```

c) 将每个作业日志复制到数据库文件中。每个作业日志的 **JOB** 参数应由值 Number/User/Job 组成, 而 **SPLNBR** 参数应仅包含 File Nbr 值。

例如:

```
====> CPYSPLF FILE(QPJ0BLOG) TOFILE(QGPL/JOBLOGS) JOB(135383/QMQM/RUNMQCHL) SPLNBR(1)
====> CPYSPLF FILE(QPJ0BLOG) TOFILE(QGPL/JOBLOGS) JOB(135534/QMQM/AMQZXMA0) SPLNBR(1)
...
```

d) 创建保存文件并将数据库文件保存到该文件。

例如:

```
====> CRTSAVF FILE(QGPL/P12345J) TEXT('Job logs for PMR 12345,67R,890')
====> SAVOBJ OBJ(JOBLOGS) LIB(QGPL) DEV(*SAVF) SAVF(QGPL/P12345J)
```

5. 如果生成了跟踪, 请创建包含跟踪文件的保存文件:

```
====> CRTSAVF FILE(QGPL/P12345T) TEXT('Trace files for PMR 12345,67R,890')
====> SAV DEV('/QSYS.LIB/QGPL.LIB/P12345T.FILE') OBJ('/QIBM/UserData/mqm/trace/*' *INCLUDE)
DTACPR(*MEDIUM)
```

6. 添加任何其他文件, 如第 227 页的『在多平台上收集故障诊断信息』中所示, 并根据 IBM 支持人员的请求, 包括包含来自 IBM MQ 和系统命令的输出的文件。

```
====> CRTSAVF FILE(QGPL/P12345X) TEXT('Extra files for PMR 12345,67R,890')
====> SAV DEV('/QSYS.LIB/QGPL.LIB/P12345X.FILE') OBJ('/tmp/QMA.mqsc.txt' *INCLUDE) ('/tmp/
ipcs.txt' *INCLUDE)
```

7. 按第 284 页的『将故障诊断信息发送到 IBM』中所述将数据发送到 IBM 时, 请确保重命名这些文件, 以使其包含完整的问题记录号, 例如从 P12345A 到 P12345,67R,890A.SAVF 等等。这是必需的, 因为 IBM i 库将名称限制为仅 10 个字符, 但 IBM ECuRep 站点需要完整的 PMR 编号以将文件与问题记录相关联。

8. 将数据发送到 IBM 后, 请备份要保留的保存文件副本, 直到案例得到解决, 然后使用 **WRKOBJ** 选项 4 删除保存文件以节省空间。

```
====> WRKOBJ OBJ(QGPL/P12345*)
```

MQ Adv. Multi 收集 AMS 问题的信息

如果需要 IBM 支持人员的帮助来解决 AMS 问题, 那么首先需要收集故障诊断信息以发送给 IBM 支持人员来帮助查找解决方案。

开始之前

在开始此任务之前, 请回答有关此问题的以下问题:

- 您在系统上观察到什么 AMS 错误?
- 详细的 AMS 消息流是什么?
- 如何在设计中实现 AMS ? (客户端 AMS 或通道, MCA 拦截 AMS)?
- AMS 问题何时开始以及何时停止?
- 涉及哪些特定用户或应用程序以及队列管理器队列? IBM MQ 安全策略, `keystore.conf` 文件和证书密钥库对于 AMS 工作非常重要。提供有关如何设置这些文件的详细信息。
- 提供 IBM MQ 客户机的类型和完整版本。

关于此任务





如果 AMS 问题立即发生, 或者如果您能够重现该问题, 那么可以生成数据以提供有关该问题的更多信息。收集故障诊断信息后, 可以将其发送到 IBM。

过程

生成故障诊断信息。

1. 生成发生安全性问题的队列管理器的跟踪。

如果还实现了客户端 AMS, 那么可能还需要 IBM MQ 客户机跟踪。

-   第 305 页的『在 UNIX and Linux 系统上使用跟踪』
-  第 315 页的『使用 Windows 上的跟踪』
-  第 310 页的『IBM i 上的跟踪』

2. 显示有关所涉及的 AMS 安全策略, `keystore.conf` 文件和密钥库的信息。

- a) 显示 AMS 安全策略。

运行 `dspmqspl` 命令, 如以下示例中所示:

```
dspmqspl -m QMGRNAME
```

其中 `QMGRNAME` 是发生问题的队列管理器的名称。

- b) 提供显示 `keystore.conf` 和证书密钥库的详细文件列表。

`keystore.conf` 文件的缺省位置是用户的主 `.mqsc` 目录。如果 `keystore.conf` 文件位于其他位置, 请显示此位置, 并说明如何指示 IBM MQ 查找 `keystore.conf` 文件。

  在 UNIX 和 Linux 上, 使用以下命令:

```
ls -alR ~/.mqsc
```

- c) 提供 `keystore.conf` 文件的内容。
- d) 提供 IBM MQ 客户机的完整类型和版本。(如果使用 Javais, 请同时提供 Java 版本详细信息。)
- e) 提供所涉及的 AMS 密钥库的证书和证书详细信息列表。

- 要列出密钥库中的证书, 请运行 `runmqakm` 命令, 如以下示例中所示。将列出证书标签。

```
runmqakm -cert -list -db keystorefilename -pw keystorepassword
```

如果密钥库的类型为 `jks`, 请使用 `runmqckm` 命令而不是 `runmqakm` 命令。

- 要显示密钥库中所有证书标签的详细信息, 请对每个标签运行 `runmqakm` 命令, 如以下示例中所示:

```
runmqakm -cert -details -db keystorefilename -pw keystorepassword -label labelname
```

如果密钥库的类型为 `jks`, 请使用 `runmqckm` 命令而不是 `runmqakm` 命令。

更新案例并收集故障诊断信息。

3. 使用 [您对初始问题的回答](#)来更新案例。

将步骤 1 中的输出/信息直接放在顶级 IBM MQ 错误目录中。 **runmqras** 自动化工具和下面的手动收集步骤都会收集在其中找到的文件。

4. 收集 IBM MQ 数据。

您可以自动或手动执行此操作。

- 使用 **runmqras** 命令自动收集数据，如第 228 页的『使用 runmqras 自动收集故障诊断信息』中所述。确保收集 **runmqras** defs, logger 和 trace 部分，并指定案例编号，如以下示例中所示：

```
runmqras -section defs,logger,trace -qmlist QMA -caseno TS123456789
```

- 或者，手动收集数据，如第 231 页的『手动收集故障诊断信息』中所述。

将故障诊断信息发送到 IBM。

5. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据！

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注：始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

相关概念

第 38 页的『对 AMS 问题进行故障诊断』

提供了信息来帮助您找到并解决与 Advanced Message Security 相关的问题。

收集通道问题的信息

如果需要 IBM 支持人员提供帮助以在 IBM MQ 通道报告问题或未能在 Multiplatforms 版上运行时解决问题，那么您首先需要收集故障诊断信息以发送给 IBM 支持人员，以帮助查找解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题：

- 您在系统上观察到什么通道问题？
- 通道问题是在什么时候开始的，什么时候停止的？
- 涉及哪些队列管理器，通道，远程队列和传输队列？

关于此任务


如果现在发生了通道问题，或者如果可以重现该问题，那么可以生成数据以提供有关该问题的更多信息。

收集故障诊断信息后，可以将其发送到 IBM。

有关对通道问题进行故障诊断的更多信息，请参阅 [对 MQ 通道进行故障诊断](#)。

过程

1. 在发生通道问题时生成队列管理器的跟踪：

-  [Linux 和 UNIX](#)

-  [Windows](#)

-  [IBM i](#)

2. 在通道的另一端同时生成 IBM MQ 跟踪，无论是远程队列管理器，本机客户机应用程序还是 JMS 或 Java 客户机：

-  [Linux 和 UNIX](#)

- **Windows** [Windows](#)
- **IBM i** [IBM i](#)
- [Java 和 JMS 客户机](#)
- **z/OS** [z/OS CHIN 跟踪](#)

3. **Linux** **UNIX**

在 UNIX and Linux 系统上，保存 **mqconfig** 命令的输出。

4. 收集 IBM MQ 数据。

您可以自动或手动执行此操作：

- 使用 **runmqras** 命令自动收集数据，如第 228 页的『[使用 runmqras 自动收集故障诊断信息](#)』中所述。请确保收集 **runmqras defs** 和 **trace** (如果跟踪了问题) 部分，并指定您的案例编号，如以下示例中所示：

```
runmqras -section defs,cluster,trace -qmlist QMA -caseno TS001234567
```

- 或者，手动收集数据，如第 231 页的『[手动收集故障诊断信息](#)』中所述。

5. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据！

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注：始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

相关概念

第 43 页的『[对分布式队列管理问题进行故障诊断](#)』

故障诊断信息可帮助您解决与分布式队列管理 (DQM) 相关的问题。

Multi 收集客户机应用程序问题的信息

如果需要 IBM 支持人员的帮助来解决多平台上 IBM MQ C, C++, COBOL, .NET, pTAL, RPG 或 Visual Basic 客户机应用程序的问题，那么首先需要收集故障诊断信息以发送给 IBM 支持人员来帮助查找解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题：

- 您在系统上观察到哪些客户机应用程序问题？
- 客户机应用程序问题何时启动以及何时停止？
- 客户机应用程序名称是什么，它连接到哪个队列管理器？
- 客户机应用程序使用哪些 SVRCONN 通道，队列和其他对象？



关于此任务

如果客户机应用程序问题正在发生，或者如果您能够重现该问题，那么可以生成数据以提供有关该问题的更多信息。

收集故障诊断信息后，可以将其发送到 IBM。

过程

1. 在发生问题时生成客户机应用程序的跟踪：

-   第 305 页的『在 UNIX and Linux 系统上使用跟踪』
 -  第 315 页的『使用 Windows 上的跟踪』
 -  第 310 页的『IBM i 上的跟踪』
2. 如果客户机应用程序从远程队列管理器接收到意外错误，请生成该队列管理器的同时 IBM MQ 跟踪:
-   第 305 页的『在 UNIX and Linux 系统上使用跟踪』
 -  第 315 页的『使用 Windows 上的跟踪』
 -  第 310 页的『IBM i 上的跟踪』
3.  
- 在 Linux 以及 UNIX 系统上，保存 `mqconfig` 命令的输出，并将此 `mqconfig` 数据直接放置在顶级 IBM MQ 错误目录中。
- 步骤 第 243 页的『4』中的自动和手动数据收集过程都会收集在此目录中找到的文件。
4. 收集 IBM MQ 数据。
- 您可以自动或手动执行此操作:
- 使用 `runmqras` 命令自动收集数据，如第 228 页的『使用 runmqras 自动收集故障诊断信息』中所述。确保收集 `runmqras` `defs` 和 `trace` (如果跟踪了问题) 部分，并指定案例编号，如以下示例中所示，用于从队列管理器 QMA 收集输出:

```
runmqras -section defs,trace -qmlist QMA -caseno TS001234567
```

- 或者，手动收集数据，如第 231 页的『手动收集故障诊断信息』中所述。
5. 将收集的信息发送到 IBM。
- 问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!
- 有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。
- 要打开或更新案例，请转至 [IBM My Support](#) 站点。
- 注:** 始终更新案例以指示已发送数据。
- 如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

收集 IBM MQ 集群问题的信息

如果在 IBM MQ 队列管理器与 Multiplatforms 版上的集群队列，主题或通道存在问题时需要 IBM 支持人员提供帮助以解决问题，那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助找到解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题:

- 您在系统上观察到哪些 IBM MQ 集群问题?
- IBM MQ 集群问题何时开始以及何时停止?
- 集群拓扑的外观以及完整存储库的位置是什么?
- 问题涉及哪些集群队列管理器，通道，队列和主题?

关于此任务

如果现在正在发生 IBM MQ 集群问题，或者如果您能够重现该问题，那么可以生成数据以提供有关该问题的更多信息。

收集故障诊断信息后，可以将其发送到 IBM。

过程

1. 在发生 IBM MQ 集群问题时生成队列管理器的跟踪:

- ▶ Linux ▶ UNIX 第 305 页的『在 UNIX and Linux 系统上使用跟踪』
- ▶ Windows 第 315 页的『使用 Windows 上的跟踪』
- ▶ IBM i 第 310 页的『IBM i 上的跟踪』

2. 如果问题涉及集群中的其他队列管理器 (例如集群完整存储库), 请在这些队列管理器上同时生成 IBM MQ 跟踪:

- ▶ Linux ▶ UNIX 第 305 页的『在 UNIX and Linux 系统上使用跟踪』
- ▶ Windows 第 315 页的『使用 Windows 上的跟踪』
- ▶ IBM i 第 310 页的『IBM i 上的跟踪』

3. ▶ Linux ▶ UNIX

在 Linux 以及 UNIX 系统上, 保存 **mqconfig** 命令的输出, 并将此 **mqconfig** 数据直接放置在顶级 IBM MQ 错误目录中。

步骤 第 244 页的『4』中的自动和手动数据收集过程都会收集在此目录中找到的文件。

4. 收集 IBM MQ 数据。

您可以自动或手动执行此操作:

- 使用 **runmqras** 命令自动收集数据, 如第 228 页的『使用 runmqras 自动收集故障诊断信息』中所述。确保收集 **runmqras** 德法斯, 集群和跟踪 (如果跟踪了问题) 部分, 并指定案例编号, 如以下示例中所示, 用于从队列管理器 QMA 和 REPOS1: 收集 **runmqras** 输出

```
runmqras -section defs,cluster,trace -qmlist QMA,REPOS1 -caseno TS001234567
```

runmqras 输出将包含所有集群定义以及集群存储库高速缓存的内容。

- 或者, 手动收集数据, 如第 231 页的『手动收集故障诊断信息』中所述。

5. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

有关 FTP 和电子邮件指示信息, 请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例, 请转至 [IBM My Support](#) 站点。

注: 始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈, 请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈, 可以致电 1-800-IBM-SERV。

相关任务

第 139 页的『对队列管理器集群问题进行故障诊断』

使用此处提供的核对表以及子主题中提供的建议, 帮助您在使用队列管理器集群时检测和处理问题。

▶ Multi 收集数据转换问题的信息

如果需要 IBM 支持人员的帮助来解决 Multiplatforms 版上的数据转换问题, 那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助查找解决方案。

开始之前

在开始此任务之前, 请回答有关此问题的以下问题:

- 您在系统上观察到哪些数据转换问题?
- 什么是消息的 MQMD.Format 及其原始 MQMD.CodedCharSetId (CCSID)?
- 什么是预期的 MQMD.CodedCharSetId ?

- 消息中的哪些特定字符无效，您希望改为看到哪些字符？

关于此任务

如果现在正在发生数据转换问题，或者如果您能够重现该问题，那么可以生成数据以提供有关该问题的更多信息。

收集故障诊断信息后，可以将其发送到 IBM。

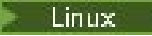

过程

1. 使用样本程序 (例如 `amqsbcbg`) 将消息放入 IBM MQ 队列后，立即浏览该消息。

查看十六进制消息以检查消息数据的 MQMD 头和字节值很重要。例如，要浏览名为 "QMA" 的队列管理器上名为 "Target.Queue" 的队列上的消息，请输入以下命令：

```
amqsbcbg Source.Queue QMA > Source.Queue.browse.txt
```



2. 在应用程序放置消息时生成队列管理器的跟踪：


-   [第 305 页的『在 UNIX and Linux 系统上使用跟踪』](#)

-  [第 315 页的『使用 Windows 上的跟踪』](#)

-  [第 310 页的『IBM i 上的跟踪』](#)

3. 如果在具有 **CONVERT(YES)** 的 IBM MQ 通道上流动时消息内容已损坏，请在消息流经发送通道时生成队列管理器的跟踪：

-   [第 305 页的『在 UNIX and Linux 系统上使用跟踪』](#)

-  [第 315 页的『使用 Windows 上的跟踪』](#)


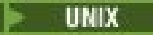
-  [第 310 页的『IBM i 上的跟踪』](#)


4. 在目标应用程序检索消息之前，使用样本程序 (例如 `amqsbcbg`) 来浏览该消息。

例如，要浏览名为 "QMA" 的队列管理器上名为 "Target.Queue" 的队列上的消息，请输入以下命令：

```
amqsbcbg Target.Queue QMA > Target.Queue.browse.txt
```

5. 如果在目标应用程序获取消息时消息内容已损坏，请在应用程序获取消息时生成队列管理器的跟踪：

-   [第 305 页的『在 UNIX and Linux 系统上使用跟踪』](#)

-  [第 315 页的『使用 Windows 上的跟踪』](#)

-  [第 310 页的『IBM i 上的跟踪』](#)

6. 收集 IBM MQ 数据。

您可以自动或手动执行此操作：

- 使用 `runmqras` 命令自动收集数据，如 [第 228 页的『使用 runmqras 自动收集故障诊断信息』](#) 中所述，以收集通道两侧的数据。确保收集 `runmqras` `defs` 和 `trace` 部分，并指定案例编号，如以下示例中所示：

```
runmqras -section defs,cluster,trace -qmlist QMA -caseno TS001234567
```

- 或者，手动收集数据，如 [第 231 页的『手动收集故障诊断信息』](#) 中所述。

7. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据！

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注: 始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈, 请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈, 可以致电 1-800-IBM-SERV。

相关任务

第 124 页的『[对消息问题进行故障诊断](#)』

Multi 收集死信队列问题的信息

如果 IBM MQ 队列管理器将消息放置在其在 Multiplatforms 版上的死信队列 (DLQ) 上, 那么您可以收集故障诊断信息以帮助查找解决方案。

开始之前

在开始此任务之前, 请回答有关此问题的以下问题:

- 您在系统上观察到哪些死信队列问题?
- 死信队列问题是在什么时候开始的, 什么时候停止的?
- 死信消息从哪里来, 他们的预定路线是什么?





关于此任务

如果消息立即进入死信队列, 或者如果您可以重现导致消息进入该队列的问题, 那么可以生成数据以提供有关该问题的更多信息。

收集故障诊断信息后, 可以将其发送到 IBM。

过程

1. 当消息进入死信队列时, 生成队列管理器的跟踪:



-   第 305 页的『[在 UNIX and Linux 系统上使用跟踪](#)』
-  第 315 页的『[使用 Windows 上的跟踪](#)』
-  第 310 页的『[IBM i 上的跟踪](#)』

2. 在目标应用程序检索消息之前, 使用样本程序 (例如 `amqsbcg`) 浏览死信队列上的消息。

例如, 要浏览名为 "QMA" 的队列管理器上名为 "Target.Queue" 的队列上的消息, 请输入以下命令:

```
amqsbcg Target.Queue QMA > Target.Queue.browse.txt
```

将浏览输出文件 (即 `QMA.DLQ.browse.txt`) 直接放在高级错误日志目录中, 即:

-  Linux 上的 `var/mqm/errors`。
-  Windows 上的 `MQ_INSTALLATION_PATH\errors`。

步骤 3 中描述的自动和手动收集过程都会收集在此目录中找到的文件。

3. 收集 IBM MQ 数据。

您可以自动或手动收集此操作:

- 使用 `runmqras` 命令自动收集数据, 如第 228 页的『[使用 runmqras 自动收集故障诊断信息](#)』中所述。确保收集 `runmqras` `defs`, `cluster` 和 `trace` 部分, 并指定案例编号, 如以下示例中所示:

```
runmqras -section defs,cluster,trace -qmlist QMA -caseno TS001234567
```

- 或者, 手动收集数据, 如第 231 页的『[手动收集故障诊断信息](#)』中所述。

4. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注: 始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

相关任务

第 124 页的『[对消息问题进行故障诊断](#)』

Multi 收集错误消息和 FFST 问题的信息

当 IBM MQ 在多平台上记录错误消息或写入 FFST (FDC 文件) 时，如果您需要 IBM 支持人员的帮助来解决问题，那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助找到解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题：

- 您在系统上观察到哪些意外错误消息或 FFST？
- 错误消息或 FFST 何时启动以及何时停止？
- 在问题开始之前是否对系统进行了任何更改？





关于此任务

如果现在正在发生错误消息或 FFST 问题，或者如果您能够重现该错误消息或 FFST 问题，那么可以生成数据以提供有关该问题的更多信息。

收集故障诊断信息后，可以将其发送到 IBM。

过程

1. 在记录错误消息或 FFST 时生成队列管理器的跟踪。如果您有大量磁盘空间，请考虑生成高详细信息跟踪。

-   第 305 页的『[在 UNIX and Linux 系统上使用跟踪](#)』
-  第 315 页的『[使用 Windows 上的跟踪](#)』
-  第 310 页的『[IBM i 上的跟踪](#)』

2. 收集 IBM MQ 数据。

您可以自动或手动执行此操作：

- 使用 `runmqras` 命令自动收集数据，如第 228 页的『[使用 runmqras 自动收集故障诊断信息](#)』中所述。请确保收集 `runmqras defs` 和 `trace` 部分，并指定案例编号，如以下示例中所示，用于从队列管理器 QMA 收集 `runmqras` 输出：

```
runmqras -section defs,cluster,trace -qmlist QMA -caseno TS001234567
```

- 或者，手动收集数据，如第 231 页的『[手动收集故障诊断信息](#)』中所述。

3. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据！

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注: 始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

Multi 收集有关挂起和高 CPU 问题的信息

如果需要 IBM 支持人员的帮助来解决多平台上的 IBM MQ 性能问题，挂起或 CPU 使用率过高的问题，那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助查找解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题：

- 您在系统上观察到哪些性能问题或挂起？
- 问题什么时候开始，什么时候停止？
- 哪些流程涉及性能问题或挂起？
- 在发生问题之前，是否对系统或应用程序进行了任何最新更新？

关于此任务

为了确定问题的原因，必须在发生性能问题或挂起时从系统收集信息，包括来自显示问题的队列管理器和应用程序的堆栈转储和其他调试数据。

收集故障诊断信息后，可以将其发送到 IBM。

过程

1. 从 Managed File Transfer 进程生成数据：

从挂起的 Managed File Transfer 进程生成三个 javacore，在每个进程之间延迟大约 1 分钟。

- a) 使用 **fteSetAgentTraceLevel** 命令生成三个代理程序 javacores，如以下示例中所示：

```
Linux UNIX fteSetAgentTraceLevel -jc AGENTNAME
...
fteSetAgentTraceLevel -jc AGENTNAME
...
fteSetAgentTraceLevel -jc AGENTNAME
```

其中 *AGENTNAME* 是挂起的 Managed File Transfer 代理程序的名称。

- b) 使用 **fteSetLoggerTraceLevel** 命令生成三个记录器 javacores，如以下示例中所示：

```
Linux UNIX fteSetLoggerTraceLevel -jc LOGGERNAME
...
fteSetLoggerTraceLevel -jc LOGGERNAME
...
fteSetLoggerTraceLevel -jc LOGGERNAME
```

其中 *LOGGERNAME* 是挂起的 Managed File Transfer 记录器的名称。

此方法生成的 javacores 根据协调队列管理器名称和代理程序名称存储在 Managed File Transfer 数据目录中。例如：

```
Linux UNIX 在 UNIX 和 Linux 上
/var/mqm/mqft/logs/COORDQMNAME/loggers/LOGGERNAME
/var/mqm/mqft/logs/COORDQMNAME/agents/AGENTNAME
```

```
Windows 在 Windows 上
```

```
C:\Program Files\IBM\MQ\mqft\logs\COORDQMNAME\agents\AGENTNAME
C:\Program Files\IBM\MQ\mqft\logs\COORDQMNAME\loggers\LOGGERNAME
```

此位置可能有所不同，具体取决于您正在使用的 IBM MQ 版本。有关更多信息，请参阅 [Windows 上的程序和数据目录位置](#)。

在这些示例中，*AGENTNAME* 或 *LOGGERNAME* 是挂起的 Managed File Transfer 代理或记录器的名称，*COORDQMNAME* 是协调队列管理器的名称。

2. 对于所有其他 Managed File Transfer 命令，从进程生成三个 javacore，如以下示例中所示。

在这种情况下，javacores 或线程转储通常会写入命令的工作目录。

a) **Linux** **UNIX**

在 UNIX 和 Linux 上，列出正在使用 **ps** 的 Java 虚拟机，并找到正在运行挂起 Managed File Transfer 命令的虚拟机。然后，将 **SIGQUIT** 发送到该进程标识 (PID) 以生成 javacore 或线程转储。**kill -QUIT** 命令不会终止 UNIX 和 Linux 上的 Java 虚拟机，而是使它们创建 javacore 或线程转储。例如：

```
sh> ps -ef | egrep 'PID|StartAgent'
UID PID PPID C STIME TTY          TIME CMD
7001 37789      1  0 Sun03PM ??          3:07.35 java ... com.ibm.wmqfte.api.StartAgent
AGENT1
7001 69177 64373  0  2:35PM ttys003    0:00.00 egrep PID|StartAgent
sh> kill -QUIT 37789
...
sh> kill -QUIT 37789
...
sh> kill -QUIT 37789
```

b) **Windows**

在 Windows 上，从 Windows 命令提示符启动 Managed File Transfer 命令。请确保将 **-F** 选项添加到 **fteStartAgent** 和 **fteStartLogger** 命令，以便它们将在前台运行，而不是在后台运行或作为 Windows 服务运行。然后输入 **Ctrl + Break** 键盘序列以从进程生成 javacore。例如：

```
C:\> fteStartLogger -F LOGGER1
...
Ctrl+Break
...
Ctrl+Break
...
Ctrl+Break
```

c) **IBM i**

在 IBM i 上，使用 **WRKJVMJOB** 选项列出系统中的 Java 虚拟机作业 7，以找到运行挂起的受管文件传输命令的虚拟机作业。然后按 **F3** 以退出并使用作业号，用户和作业名从作业生成 Java 线程转储。例如：

```
====> WRKJVMJOB
Opt  Job Name      User           Number  Function           Status
   QJVACMSRV    QMQM           136365  PGM-StartAgent     THDW
   QYPSJSVR     QYPSJSVR      136415  PGM-jvmStartPa     SIGW
```

使用选项 7 来查找正确的作业，使用 **F3** 以返回到命令行：

```
====> GENJVM DMP JOB(136365/QMQM/QJVACMSRV) TYPE(*JAVA)
```

3. **Linux** **UNIX**

在 UNIX 和 Linux 上，使用 **stackit** 和 **sigdump** 脚本从进程生成调试数据。

a) 下载 IBM **stackit** 和 **sigdump** 脚本。在 Linux 系统上，必须安装 GNU 调试器 (GDB 上称为 **WDB**)，即使临时安装也是如此，**stackit** 才能工作：

- **Linux** [下载堆栈](#)
- **Linux** [下载 GDB for Linux](#)

b) 针对受影响的 IBM MQ 队列管理器 and 应用程序运行 **stackit** 脚本三次，每次运行之间的延迟为一分钟或更少。

例如：

```
sh> stackit -m QMA -m QMB -n myapp -f /var/mqm/errors/stackit-1.txt
sh> sleep 30
sh> stackit -m QMA -m QMB -n myapp -f /var/mqm/errors/stackit-2.txt
```

```
sh> sleep 30
sh> stackit -m QMA -m QMB -n myapp -f /var/mqm/errors/stackit-3.txt
```

- c) 对受影响的 IBM MQ 队列管理器运行一次 `sigdump` 脚本。 `sigdump` 脚本将使每个队列管理器生成诊断 FFST 文件。

例如：

```
sh> sigdump -m QMA -m QMB
```

4. Windows

在 Windows 上，使用调试实用程序从进程生成调试。

- a) 如果系统上没有以下调试实用程序，请从 Microsoft 下载这些实用程序：

- Windows 的最新调试工具版本，可从 [Debug Diagnostic Tool](#) 获取
- [下载 Microsoft PsList](#)
- [下载 Microsoft Handle](#)
- [下载 Microsoft Process Monitor](#)

- b) 显示进程列表：

```
C:\> tasklist -v
```

- c) 显示有关每个进程的其他信息：

```
C:\> pslist -x
```

- d) 通过将每个进程名称的前几个字符传递到句柄程序来显示有关 IBM MQ 进程和任何受影响的应用程序的信息，例如：

```
C:\> handle -a -p amq
C:\> handle -a -p runmq
C:\> handle -a -p myapp
```

- e) 从 IBM MQ 进程和任何受影响的应用程序的挂起 (甚至崩溃) 收集数据，例如：

```
C:\> adplus -hang -pn amqzma0.exe
C:\> adplus -hang -pn amqz1aa0.exe
C:\> adplus -crash -pn runmqchi.exe
```

- f) 使用 Microsoft Process Monitor 工具可提供实时堆栈数据，装入的模块，环境信息，访问的文件，使用的库，访问的注册表键以及更多信息。

此工具可能非常占用 CPU，即使设置了过滤选项也是如此。请参阅所包含的 `procmon.chm` 帮助文件中的 "脚本编制 Process Monitor" 部分，以获取有关在脚本或批处理文件中使用该帮助文件的信息。

5. IBM i

在 IBM i 上，使用 MQSTACK 和 SERVICEDOCS 工具从进程生成调试数据：

- a) 下载并运行 IBM MQSTACK 工具。MQSTACK 将显示所有队列管理器进程的所有线程的状态，但是它不会显示有关非 IBM 进程的信息。
- b) 对于不属于队列管理器的进程 (例如应用程序)，请运行 [SERVICEDOCS](#) 实用程序。SERVICEDOCS 将显示系统上每个进程的主线程的堆栈。


6. 在发生问题时生成 IBM MQ 跟踪：

- [Linux](#) [UNIX](#) 第 305 页的『在 UNIX and Linux 系统上使用跟踪』
- [Windows](#) 第 315 页的『使用 Windows 上的跟踪』
- [IBM i](#) 第 310 页的『IBM i 上的跟踪』



为避免系统性能恶化，请在短时间 (例如，一分钟或更短的时间) 后停止跟踪。

7. 如果在 WebSphere Application Server 中发生挂起或高 CPU 使用率，请完成适用于您的平台的 WebSphere Application Server MustGather 指示信息：

-  [AIX](#)
-  [Linux](#)
-  [Solaris](#)
-  [Windows](#)
-  [IBM i](#)

8.   在 UNIX and Linux 系统上，保存 `mqconfig` 命令的输出。

9. 将以下信息直接放在顶级 IBM MQ errors 目录中：

- 在步骤 1 中收集的调试文件。
-   您在步骤 4 中收集的 `mqconfig` 命令的输出。

步骤 第 251 页的『10』中的自动和手动数据收集过程都会收集在此目录中找到的文件。

10. 收集 IBM MQ 数据。

您可以自动或手动执行此操作：

- 使用 `runmqras` 命令自动收集数据，如第 228 页的『使用 runmqras 自动收集故障诊断信息』中所述。确保收集 `runmqras defs, cluster` 和 `trace` 部分，并指定案例编号，如以下示例中所示：

```
runmqras -section defs,cluster,trace -qmlist QMA -caseno TS001234567
```

- 或者，手动收集数据，如第 231 页的『手动收集故障诊断信息』中所述。

11. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据！

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注：始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

相关任务

第 124 页的『[对消息问题进行故障诊断](#)』

收集 IBM MQ Explorer 问题的信息

如果在管理队列管理器时需要 IBM 支持人员提供帮助以解决 IBM MQ Explorer 问题，那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助查找解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题：

- 您在系统上观察到什么 IBM MQ Explorer 问题？
- IBM MQ Explorer 是 IBM MQ 服务器安装的一部分，还是将其作为 [独立应用程序从 Fix Central](#) 下载？
- 您尝试管理哪些队列管理器，它们位于哪些系统上？
- 远程队列管理器正在运行哪个操作系统版本和 IBM MQ 版本？

关于此任务





IBM MQ Explorer 作为可安装的服务器组件以及通过 Fix Central 作为独立安装提供给 Linux 和 Windows 系统。IBM MQ Explorer 可以管理其安装所在的本地队列管理器以及所有平台上的远程队列管理器。

发生问题时，从 IBM MQ Explorer 收集信息以确定原因很重要。

收集故障诊断信息后，可以将其发送到 IBM。

过程

1. 生成 IBM MQ Explorer 跟踪，这将在您尝试使用 IBM MQ Explorer 来管理队列管理器时显示问题。
2. 在应用程序放置消息时生成队列管理器的跟踪：

-   第 305 页的『在 UNIX and Linux 系统上使用跟踪』
-  第 315 页的『使用 Windows 上的跟踪』
-  第 310 页的『IBM i 上的跟踪』

3. 如果 IBM MQ Explorer 中存在图形问题，请拍摄屏幕快照或使用摄像头来捕获问题的图像。

4. 收集 IBM MQ 数据。

- a) 记录 MQ Explorer 版本和维护级别。
- b) 记录目标队列管理器的 MQ 版本和维护级别。
- c) 记录运行 IBM MQ Explorer 和目标队列管理器的操作系统版本和维护级别。
- d) 如果要使用从 Fix Central 安装的独立 IBM MQ Explorer，请列出其安装目录的内容，例如：

```
Linux sh> ls -alR "/opt/ibm/wmq-explorer"
```

```
Windows C:\> DIR /S "C:\Program Files\IBM\MQ Explorer"
```

注：目录名称是在安装期间选择的，可能与这些示例不同。

- e) 查找 IBM MQ Explorer .log 文件。

当 IBM MQ Explorer 迁到错误时，它可能会创建一个仅名为 .log 的文件，其中包含更多信息。根据 IBM MQ Explorer 安装类型和迁到问题的用户，在相应目录中查找 .log 文件，并收集 .log 文件以及 .metadata 目录中的所有其他文件。在以下示例中，\$HOME 和 %USERPROFILE% 是用于查找文件的特定于用户的环境变量。

- 要在 IBM MQ Explorer 的 .log 文件作为 IBM MQ 服务器安装的一部分时查找该文件，请执行以下操作：

```
Linux sh> ls -al "$HOME"/IBM/WebSphereMQ/workspace-  
InstallationName/.metadata/.log
```

```
Windows C:\> DIR "%USERPROFILE%\IBM\WebSphereMQ\workspace-  
InstallationName\metadata\log"
```

其中 *InstallationName* 表示 IBM MQ 安装的名称。

- 要查找独立 IBM MQ Explorer 的 .log 文件，请执行以下操作：

```
Linux sh> ls -al "$HOME"/IBM/*MQ/workspace/.metadata/.log
```

```
Windows C:\> DIR "%USERPROFILE%\IBM*\MQ\workspace\metadata\log"
```

- f) 如果 IBM MQ Explorer 在连接时迁到困难，请使用操作系统工具在连接尝试之前和之后立即列出双方的网络连接：

- **Linux** **UNIX** 要在 UNIX and Linux 上显示网络连接:

```
sh> netstat -an
```

- **Windows** 要在 Windows 上显示网络连接:

```
C:\>NETSTAT -AN
```

g) 手动打包 IBM 的文件:

- **Linux** **UNIX** 第 236 页的『手动打包 UNIX 和 Linux 上的信息』
- **Windows** 第 237 页的『手动打包 Windows 上的信息』

5. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

有关 FTP 和电子邮件指示信息, 请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例, 请转至 [IBM My Support](#) 站点。

注: 始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈, 请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈, 可以致电 1-800-IBM-SERV。

相关任务

[对 IBM MQ Explorer 的问题进行故障诊断](#)

Multi 收集安装和卸载问题的信息

如果需要 IBM 支持人员的帮助以解决问题 IBM MQ, 或者其某个修订包未能在多平台上正确安装或卸载, 那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助查找解决方案。

开始之前

在开始此任务之前, 请回答有关此问题的以下问题:

- 您正在尝试安装或卸载什么?
- 您使用什么帐户来执行安装或卸载?

关于此任务

在发生安装或卸载问题时, 从系统收集信息以确定原因很有用。

收集故障诊断信息后, 可以将其发送到 IBM。

过程

1. 在启用调试日志记录的情况下运行安装或卸载, 以收集有关失败的更多详细信息:

- a) **AIX**
要在 AIX 上生成调试安装和卸载数据, 请执行以下操作:

- i) 导出环境变量 `INST_DEBUG = YES`, 这将指示 AIX 记录额外的调试信息。然后通过 SMIT 或直接运行 `installp` 命令来运行安装或卸载。例如:

```
sh> export INST_DEBUG=YES  
sh> installp...
```

- ii) 完成时取消设置 `INST_DEBUG` 变量:

```
sh> unset INST_DEBUG
```

位于系统根目录中的 `smit.log` 文件将包含来自安装或卸载尝试的调试信息。

b) **Linux**

要在 Linux 上生成调试安装和卸载数据，请将 `-vv` 选项添加到 `rpm` 命令，并将所有输出 (`stdout` 和 `stderr`) 捕获到文件中。

例如：

```
sh> rpm -vv ... 2>&1 | tee mqinstall.log
```

c) **Solaris**

要在 Solaris 上生成调试安装和卸载数据，请执行以下操作：

i) 使用脚本命令可开始将输出记录到文件中。例如：

```
sh> script mqinstall.log
```

ii) 将 `-v` 选项添加到 Solaris `pkgadd` 或 `pkgrm` 命令。例如：

```
sh> pkgadd -v ...
```

iii) 退出脚本命令以停止日志记录输出。例如：

```
sh> exit
```

d) **Windows**

要在 Windows 上生成调试安装和卸载数据，请使用带有选项 `/l*vx` 的 `msiexec` 命令将调试输出记录到文件中。

要确定要用于使用 `msiexec` 安装或卸载 IBM MQ 的其他参数，请参阅 [使用 msiexec 安装服务器](#)。例如：

```
C:\> msiexec /l*vx "C:\mqinstall.log" ...
```

e) **IBM i**

要在 IBM i 上生成调试安装和卸载数据，请在 `RSTLICPGM` 或 `DLTLICPGM` 命令上指定 **OUTPUT(*PRINT)** 选项以确保作业记录处于假脱机状态。

例如：

```
====> RSTLICPGM ... OUTPUT(*PRINT)
```

然后使用 `WRKSPLF` 选项 5 来显示作业记录。

2. 收集 IBM MQ 数据。

保存安装或卸载过程报告的任何错误的输出。拍摄错误的屏幕快照，或者使用摄像头手机来捕获问题的图像。

a) 记录系统上当前的 MQ 版本和维护级别，或者标识您尝试安装的版本。

b) 记录 操作系统版本和维护级别。

c) 如果系统具有多个 IBM MQ 安装，请记录 IBM MQ 安装详细信息：

- **Linux** **UNIX** 在 UNIX 和 Linux 上：

```
sh> dspmqinst > /tmp/dspmqinst.txt
```

- **Windows** 在 Windows 上：

```
C:\> dspmqinst > %TEMP%\dspmqinst.txt
```

d) **Linux** **UNIX**

在 UNIX 和 Linux 系统上，包含 `/etc/opt/mqm/mqinst.ini` 文件 (如果存在)。

e) **Windows**

在 Windows 系统上, 使用 `amquregn` 程序从 Windows 注册表信息保存 IBM MQ 信息的副本 (如果有 IBM MQ 安装可用于运行该安装)。

f) 记录用于启动安装或卸载过程的精确命令。

Solaris Linux 在 Linux 和 Solaris 系统上, 包含用于重新打包 IBM MQ 的 `crtmqpkg` 命令 (如果您正在处理多个安装)。

g) **AIX**

在 AIX 系统上, 收集在系统根目录中找到的 `smit.log` 和 `smit.script` 文件。

h) **Windows**

在 Windows 系统上, 收集 MSI 安装程序日志文件。如果使用了 `msiexec`, 那么将在命令行上选择文件名。否则, 请包含位于尝试安装或卸载的用户的 `%TEMP%` 目录中的所有名为 `MSI*.*`, `MQ*.*` 和 `amq*.*` 的文件。包含 IBM MQ 数据目录中的文件 `amqmccw.txt` 和 `amqmjpse.txt` (如果存在)。

i) 在所有系统上, 包含 IBM MQ 安装目录中的 `mqpatch.dat` 和 `mqpatch.log` 文件 (如果存在)。

j) 在所有系统上, 列出尝试安装, 更新或删除 IBM MQ 的目录的内容 (如果有)。例如:

• **Linux UNIX** 在 UNIX 和 Linux 上:

```
sh> ls -alR /path/to/mq > mqfiles.txt
```

• **Windows** 在 Windows 上:

```
C:\> DIR /S "C:\Program Files\IBM\MQ" > %TEMP%\mqfile.txt
```

• **IBM i** 在 IBM i Qshell 上:

```
===> ls -alR /QIBM/UserData/mqm /QIBM/ProdData/mqm /QSYS.LIB/QMQM.LIB > /tmp/mqfile.txt
```

k) 手动打包 IBM 的文件, 包括包含步骤 1 和 2 中列出的命令输出的文件。对于新安装, 请跳过系统上尚不存在的任何目录或文件:

• **Linux UNIX** [第 236 页的『手动打包 UNIX 和 Linux 上的信息』](#)

• **Windows** [第 237 页的『手动打包 Windows 上的信息』](#)

• **IBM i** [第 238 页的『手动打包 IBM i 上的信息』](#)

3. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

有关 FTP 和电子邮件指示信息, 请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例, 请转至 [IBM My Support](#) 站点。

注: 始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈, 请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈, 可以致电 1-800-IBM-SERV。

相关任务

[第 124 页的『对消息问题进行故障诊断』](#)

Multi 收集 Java 和 JMS 应用程序问题的信息

如果需要 IBM 支持人员的帮助来解决 Multiplatforms 版上 Java 或 JMS 应用程序的问题, 那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助查找解决方案。

开始之前

IBM 建议在 Java Platform, Enterprise Edition (Java EE) 应用程序服务器 (例如 WebSphere Application Server) 中使用 IBM MQ classes for Java。如果要在 Java EE 环境中使用 IBM MQ classes for Java，请[查看限制及其用法的其他注意事项](#)。

在开始此任务之前，请回答有关此问题的以下问题：

- 您在系统上观察到什么 Java 或 JMS 问题？
- Java 或 JMS 问题开始的时间以及停止的时间？
- 是否报告了任何 Java 异常，以及它们是否包含 Java 调用堆栈？
- Java 或 JMS 应用程序使用哪些队列管理器，队列和主题？

关于此任务

发生 Java 或 JMS 问题时，必须从系统收集信息以确定原因。

收集故障诊断信息后，可以将其发送到 IBM。

过程

1. 根据应用程序是使用 IBM MQ Java 还是 JMS 接口，[生成 IBM MQ classes for Java 跟踪或 IBM Java 消息服务跟踪](#)。
如果应用程序正在 WebSphere Application Server 下运行，请遵循该环境的跟踪指示信息。
2. 在发生问题时生成客户机应用程序的跟踪：
 -   [第 305 页的『在 UNIX and Linux 系统上使用跟踪』](#)
 -  [第 315 页的『使用 Windows 上的跟踪』](#)
 -  [第 310 页的『IBM i 上的跟踪』](#)
3. 收集 Java 或 JMS 应用程序的以下信息：
 - a) 如果应用程序正在 WebSphere Application Server 中运行，请使用其收集器工具来收集有关应用程序服务器及其配置，JNDI 定义，FFDC 文件，日志以及步骤 1 和 2 中生成的任何跟踪的信息：
 - [WebSphere Application Server traditional 9.0.5](#)
 - [WebSphere Application Server 8.5.5](#)
 - b) 如果应用程序正在另一个 Java 应用程序服务器或 Java Platform, Standard Edition (Java SE) 环境中运行，请收集以下文件：
 - 标准输出流数据 (例如，`System.out` 或类似文件)。
 - 标准错误流数据 (例如，`System.err` 或类似文件)。
 - Java 虚拟机日志文件 (例如，`native_stdout.log` 和 `native_stderr.log` 或类似文件)。
 - `mqjms.log` 文件，缺省情况下在应用程序的当前工作目录中找到。
 - 在同一目录中找到针对 Java 虚拟机的进程标识命名的 `mqjms_PID.trc` 文件。
 - 在应用程序当前工作目录的 FFDC 子目录中找到的任何 FFST 文件。
4. 将 Java 或 JMS 跟踪和日志从步骤 1 到 3 以及 (如果适用) WebSphere Application Server 收集器放在顶级 IBM MQ 错误目录中。
步骤 [第 256 页的『5』](#) 中的自动和手动数据收集过程都会收集在此目录中找到的文件。
5. 收集 IBM MQ 数据。
您可以自动或手动执行此操作：

- 使用 **runmqras** 命令自动收集数据，如第 228 页的『使用 runmqras 自动收集故障诊断信息』中所述。确保收集 **runmqras** 跟踪部分以及队列管理器中的 **defs** 和 **主题** 部分，并指定案例编号，如以下示例中所示，以从队列管理器 QMA 收集输出：

```
runmqras -section defs,topic,trace -qmlist QMA -caseno TS001234567
```

要从客户机收集输出，请指定 **trace** 部分和您的案例编号，如以下示例中所示：

```
runmqras -section trace -caseno TS001234567
```

- 或者，手动收集数据，如第 231 页的『手动收集故障诊断信息』中所述。
6. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据！

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注：始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

Multi 收集日志记录和恢复问题的信息

如果您需要 IBM 支持人员的帮助，以解决 IBM MQ 队列管理器在 Multiplatforms 版上报告日志记录数据错误或从其日志中恢复信息的问题，那么您首先需要收集故障诊断信息以发送给 IBM 支持人员，以帮助找到解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题：

- 您在系统上观察到哪些日志记录或恢复问题？
- 日志记录或恢复问题何时开始以及何时停止？
- 您还可以提供哪些其他详细信息来帮助确定问题的原因？

关于此任务





如果现在发生了日志记录或恢复问题，或者如果您能够重现该问题，那么可以生成数据以提供有关该问题的更多信息。


收集故障诊断信息后，可以将其发送到 IBM。

过程

1. 在发生问题时生成队列管理器的跟踪。

如果您有大量磁盘空间，请考虑收集高详细信息跟踪：

-   第 305 页的『在 UNIX and Linux 系统上使用跟踪』
-  第 315 页的『使用 Windows 上的跟踪』
-  第 310 页的『IBM i 上的跟踪』

2. 

在 UNIX, Linux, and Windows 上，转储队列管理器日志的内容。

如果您怀疑要记录的数据量存在问题，那么这特别有用。

注：必须停止有问题的队列管理器才能转储其日志。您还必须提供队列管理器的日志路径。日志路径是使用 `qm.ini` 文件的日志节的 **LogPath** 属性定义的。

以下示例中的命令使用 **dmpmqlog** 命令来转储队列管理器 QMA 的日志内容:

- Linux UNIX 在 UNIX and Linux 上:

```
sh> endmqm -i QMA
sh> dmpmqlog -b -m QMA -f /var/mqm/log/QMA > /tmp/QMA.dmpmqlog.txt
sh> stmqm QMA
```

- Windows 在 Windows 上:

```
C:\> endmqm -i QMA
C:\> dmpmqlog -b -m QMA -f "C:\ProgramData\IBM\MQ\log\QMA" > %TEMP%\QMA.dmpmqlog.txt
C:\> stmqm QMA
```

3. Linux UNIX

在 Linux 以及 UNIX 系统上, 保存 **mqconfig** 命令的输出。

- 将您在步骤 2 和 3 中生成的 **dmpmqlog** 命令和 **mqconfig** 命令的输出放在顶级 IBM MQ 错误目录中。步骤 第 258 页的『5』中的自动和手动数据收集过程都会收集在此目录中找到的文件。
- 收集 IBM MQ 数据。

您可以自动或手动执行此操作:

- 使用 **runmqras** 命令自动收集数据, 如第 228 页的『使用 runmqras 自动收集故障诊断信息』中所述。确保收集 **runmqras** 跟踪部分以及队列管理器中的 **defs** 和 **主题** 部分, 并指定案例编号, 如以下示例中所示, 以从队列管理器 QMA 收集输出:

```
runmqras -section defs,topic,trace -qmlist QMA -caseno TS001234567
```

要从客户机收集输出, 请指定 **trace** 部分和您的案例编号, 如以下示例中所示:

```
runmqras -section trace -caseno TS001234567
```

- 或者, 手动收集数据, 如第 231 页的『手动收集故障诊断信息』中所述。
- 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

有关 FTP 和电子邮件指示信息, 请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例, 请转至 [IBM My Support](#) 站点。

注: 始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈, 请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈, 可以致电 1-800-IBM-SERV。

Multi 在 Multiplatforms 版上收集 Managed File Transfer 问题的信息

如果在 Managed File Transfer (MFT) 代理程序, 记录器或命令报告问题或未能在多平台上正常工作时需要 IBM 支持人员提供帮助来解决问题, 那么首先需要收集故障诊断信息以发送给 IBM 支持人员来帮助查找解决方案。需要的信息取决于您看到的问题。

过程

- 收集您看到的问题类型所需的信息:

- [Managed File Transfer 代理程序问题](#)
- [Managed File Transfer 协议网桥代理问题](#)
- [Managed File Transfer 资源监视器问题](#)
- [Managed File Transfer 受管传输问题](#)
- [Managed File Transfer 数据库记录器问题](#)
- [Managed File Transfer 文件记录器问题](#)

- [Managed File Transfer 命令问题](#)
2. 收集调查问题所需的 Managed File Transfer 数据后，创建包含所有相关文件的归档。
有关更多信息，请参阅第 264 页的『[创建 MFT 故障诊断信息的归档](#)』。
 3. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注: 始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

相关参考

第 77 页的『[对 Managed File Transfer 问题进行故障诊断](#)』

使用以下参考信息可帮助您诊断 Managed File Transfer 中的错误:

收集 MFT 代理程序问题的信息

如果需要 IBM 支持人员针对 Managed File Transfer (MFT) 代理程序问题提供帮助，那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

Managed File Transfer 代理程序问题包括:

- 未能与其代理队列管理器连接或断开连接的代理。
- 代理程序挂起。
- 代理程序意外停止。
- 正在恢复的代理程序。
- **fteListAgents** 或 **fteShowAgentDetails** 命令或 IBM MQ Explorer Managed File Transfer 插件显示了代理程序的不正确状态信息或过期状态信息。
- 代理程序未能报告任何状态信息。

过程

1. 最初，请查看以下主题以了解它们是否帮助您解决问题:
 - 第 82 页的『[对代理程序状态问题进行故障诊断](#)』
 - 第 102 页的『[对 java.lang.OutOfMemoryError 问题进行故障诊断](#)』
 - 第 110 页的『[对 Connect:Direct 网桥进行故障诊断](#)』
2. 如果仍需要帮助，请收集以下信息并将其发送给 IBM 支持人员:
 - 代理的名称。
 - 代理队列管理器的名称。
 - 代理程序正在使用的 Managed File Transfer 版本。
 - 代理队列管理器的 IBM MQ 版本。
 - 代理程序的安装类型 (即，代理程序是从 IBM MQ 产品安装介质安装的，还是通过 Managed File Transfer 可重新分发的代理程序软件包安装的?)。
 - 发生问题时在代理程序的事件日志 (output0.log) 中看到的任何错误消息。
 - 涵盖问题时间的代理程序跟踪。有关如何收集跟踪的更多信息，请参阅第 352 页的『[在多平台上跟踪 Managed File Transfer 代理](#)』。
 - 如果代理在与其代理队列管理器通信时迁到问题 (例如，代理的事件日志包含错误消息，其中包含 IBM MQ 原因码，例如 2009-MQRC_CONNECTION_BROKEN)，请使用跟踪规范 =all 收集跟踪。

- 对于所有其他问题，请使用跟踪规范 `com.ibm.wmqfte=all` 来收集跟踪。
- 如果代理程序挂起，那么三个 Javacore 相隔 30 秒。要执行此操作，请运行带有 `-jtc` 选项集的 **`fteSetAgentTraceLevel`** 命令，如以下示例中所示：

```
fteSetAgentTraceLevel -jtc <agent_name>
```

如果该命令未导致代理程序生成 Javacore，那么应该向代理程序进程发送 SIGQUIT 信号。

- 包含代理程序的日志文件，配置文件，跟踪文件和 Javacores (如果适用) 的归档。有关如何创建归档的更多信息，请参阅第 264 页的『创建 MFT 故障诊断信息的归档』。
- 协调队列管理器和代理队列管理器的 **`runmqras`** 输出。有关如何创建输出的更多信息，请参阅第 228 页的『使用 runmqras 自动收集故障诊断信息』。

Multi 收集 MFT 协议网桥代理问题的信息

如果需要 IBM 支持人员针对 Managed File Transfer (MFT) 协议网桥代理问题提供帮助，那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

Managed File Transfer 协议网桥代理问题包括：

- 代理程序未能连接到远程文件服务器或无法与远程文件服务器断开连接。
- 与远程文件服务器之间的受管传输失败。

过程

1. 最初，请查看第 94 页的『对协议网桥代理问题进行故障诊断』中的信息，以了解这是否有助于您解决问题。
2. 如果仍需要帮助，请收集以下信息并将其发送给 IBM 支持人员：
 - 协议网桥代理的名称。
 - 协议网桥代理队列管理器的名称。
 - 协议网桥代理正在使用的 Managed File Transfer 版本。
 - 协议网桥代理队列管理器的 IBM MQ 版本。
 - 远程文件服务器系统的主机名。
 - 远程文件服务器的产品和版本信息。
 - 代理程序用于与远程文件服务器 (即 FTP，FTPS 或 SFTP) 通信的协议。
 - 协议网桥代理配置文件 (`ProtocolBridgeProperties.xml`) 中远程文件服务器的条目。
 - 发生问题时在代理程序的事件日志 (`output0.log`) 中看到的任何错误消息。
 - 协议网桥代理日志文件，其中正在使用的协议的日志级别设置为 on。有关如何设置日志级别的更多信息，请参阅 `fteSetAgentLog` 级别 (打开或关闭对某些 MFT 代理操作的文件的日志记录)。
 - 包含协议网桥代理的日志文件和配置文件的归档。有关如何创建归档的更多信息，请参阅第 264 页的『创建 MFT 故障诊断信息的归档』。

Multi 收集 MFT 资源监视器问题的信息

如果需要 IBM 支持人员针对 Managed File Transfer (MFT) 资源监视器问题提供帮助，那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

Managed File Transfer 资源监视器问题包括：

- 资源监视器停止轮询。
- 资源监视器正在轮询，并且不会在任何项 (文件或消息) 上触发。

- 资源监视器未向代理提交受管传输请求。
- 资源监视器意外停止。

过程

1. 最初，请查看 [第 95 页的『对资源监视器问题进行故障诊断』](#) 中的信息，以了解这是否有助于您解决问题。
2. 如果仍需要帮助，请收集以下信息并将其发送给 IBM 支持人员：
 - 代理的名称。
 - 代理队列管理器的名称。
 - 代理程序正在使用的 Managed File Transfer 版本。
 - 代理队列管理器的 IBM MQ 版本。
 - 资源监视器的名称。
 - 监视器正在轮询的资源 (队列或目录) 的名称。
 - 监视器的触发条件。
 - 监视器的任务 XML。
 - 监视器未触发的任何项的详细信息。
 - 资源监视器日志文件 (例如，`resmonevent0.log`)，其中资源监视器的日志级别设置为 VERBOSE。有关如何创建日志文件的更多信息，请参阅 [日志记录 MFT 资源监视器](#)。

如果监视器正在轮询，但未卡住，那么日志文件应包含至少三个轮询的条目。

 - 包含代理程序配置文件以及代理程序和资源监视器的日志文件的归档。有关如何创建归档的更多信息，请参阅 [第 264 页的『创建 MFT 故障诊断信息的归档』](#)。

Multi

收集 MFT 受管传输问题的信息

如果需要 IBM 支持人员针对 Managed File Transfer (MFT) 管理的传输问题提供帮助，那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

与 Managed File Transfer 受管传输相关的问题包括：

- 受管传输意外失败。
- 正在进入恢复且未完成的受管传输。
- 受管传输卡住。

过程

1. 最初，请查看 [第 88 页的『对受管传输问题进行故障诊断』](#) 中的信息，以了解这是否有助于您解决问题。
2. 如果仍需要帮助，请收集以下信息并将其发送给 IBM 支持人员：
 - 受管传输的源代理的名称。
 - 源代理队列管理器的名称。
 - 源代理正在使用的 Managed File Transfer 或 Managed File Transfer for z/OS 的版本。
 - 源代理队列管理器的 IBM MQ 或 IBM MQ for z/OS 版本。
 - 受管传输的目标代理的名称。
 - 目标代理队列管理器的名称。
 - 目标代理正在使用的 Managed File Transfer 或 Managed File Transfer for z/OS 版本。
 - 目标代理队列管理器的 IBM MQ 或 IBM MQ for z/OS 版本。

- 如果源和目标代理队列管理器不同，那么将详细说明队列管理器如何连接在一起 (即，通过发送方/接收方通道或 IBM MQ 集群)。
- 受管传输的传输标识。
- 如何创建受管传输请求的详细信息 (即，是由资源监视器， **fteCreateTransfer** 命令， IBM MQ Explorer Managed File Transfer 插件还是其他内容生成的?)。
- 与源代理或目标代理的事件日志 (output0.log) 中的受管传输相关的任何错误消息的详细信息。
- 来自源代理和目标代理的跟踪，涵盖发生问题的时间。有关如何收集跟踪的更多信息，请参阅第 352 页的『在多平台上跟踪 Managed File Transfer 代理』或第 358 页的『跟踪 Managed File Transfer for z/OS 代理』。应该使用跟踪规范 `com.ibm.wmqfte=all` 来收集跟踪。
- 来自源代理程序的归档，其中包含代理程序的日志文件和配置文件，以及来自目标代理程序的归档，其中包含代理程序的日志文件和配置文件。有关如何收集源和目标代理的归档的更多信息，请参阅第 264 页的『创建 MFT 故障诊断信息的归档』。
- 源代理队列管理器和目标代理队列管理器的 **runmqras** 输出。有关如何收集源代理队列管理器和目标代理队列管理器的 **runmqras** 输出的更多信息，请参阅第 228 页的『使用 runmqras 自动收集故障诊断信息』。

在调查与受管传输相关的问题时，绘制一个简单的图通常很有用，如以下示例中所示，该图显示了代理和代理队列管理器。此图允许您和 IBM 支持人员查看如何连接代理和代理队列管理器，这可帮助确定 IBM MQ 网络中可能导致受管传输进入恢复或卡住的问题。

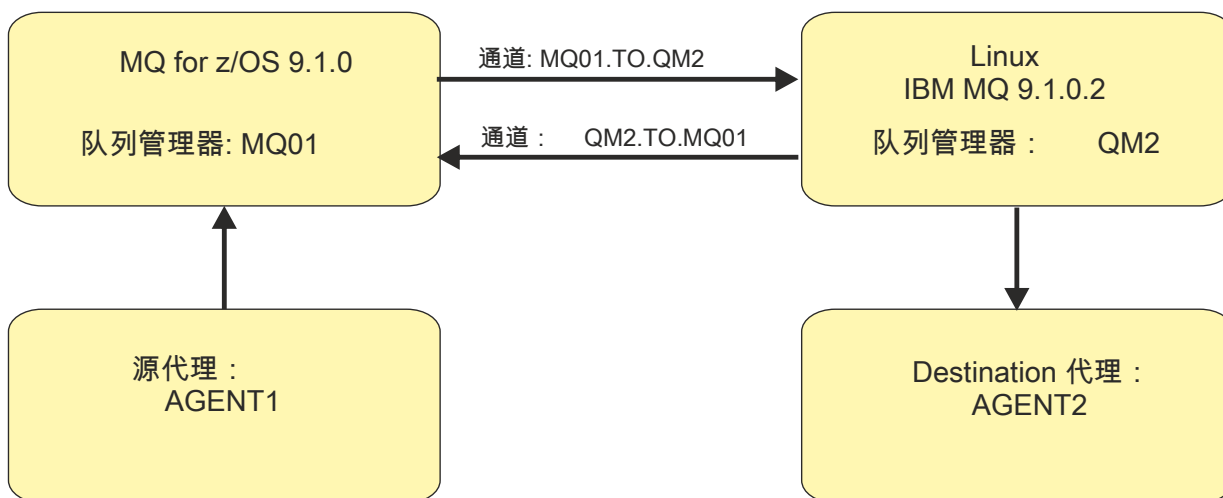


图 51: 显示如何连接源代理 AGENT1 和目标代理 AGENT2 及其代理队列管理器 MQ01 和 QM2 的简单图示例。

Multi 收集 MFT 数据库记录器问题的信息

如果需要 IBM 支持人员针对 Managed File Transfer (MFT) 数据库记录器问题提供帮助，那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

Managed File Transfer 数据库记录器问题包括:

- 数据库记录器无法连接到协调队列管理器。
- 数据库记录器无法连接到数据库。
- 数据库记录器不会更新数据库。

过程

1. 最初，请查看第 108 页的『对记录器问题进行故障诊断』中的信息，以了解这是否有助于您解决问题。
2. 如果仍需要帮助，请收集以下信息并将其发送给 IBM 支持人员:

- 数据库记录器的名称。
- 数据库记录器要连接到的协调队列管理器的名称。
- 数据库记录器正在使用的 Managed File Transfer 版本。
- 协调队列管理器的 IBM MQ 版本。
- 数据库记录器正在使用的数据库类型。
- 发生问题时在数据库记录器的事件日志中显示的任何错误消息的详细信息。
- 涵盖问题时间的数据库记录器跟踪。有关如何收集此跟踪的更多信息，请参阅第 356 页的『在多平台上跟踪 Managed File Transfer 独立记录器』。
 - 如果数据库记录器在与协调队列管理器通信时迁到问题 (例如，数据库记录器的事件日志包含错误消息，其中包含 IBM MQ 原因码，例如 2009-MQRC_CONNECTION_BROKEN)，请使用跟踪规范 =all 收集跟踪。
 - 对于所有其他问题，请使用跟踪规范 com.ibm.wmqfte=all 来收集跟踪。
- 包含数据库记录器日志文件和配置文件的归档。有关如何创建归档的更多信息，请参阅第 264 页的『创建 MFT 故障诊断信息的归档』。
- 协调队列管理器的 **runmqras** 输出。有关如何创建输出的更多信息，请参阅第 228 页的『使用 runmqras 自动收集故障诊断信息』。

Multi 收集 MFT 文件记录器问题的信息

如果需要 IBM 支持人员针对 Managed File Transfer (MFT) 文件记录器问题提供帮助，那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

Managed File Transfer 文件记录器问题包括：

- 文件记录器无法连接到协调队列管理器。
- 文件记录器无法记录任何数据。

过程

1. 最初，请查看第 108 页的『对记录器问题进行故障诊断』中的信息，以了解这是否有助于您解决问题。
2. 如果仍需要帮助，请收集以下信息并将其发送给 IBM 支持人员：
 - 文件记录器的名称。
 - 文件记录器要连接到的协调队列管理器的名称。
 - 文件记录器正在使用的 Managed File Transfer 版本。
 - 协调队列管理器的 IBM MQ 版本。
 - 数据库记录器正在使用的数据库类型。
 - 发生问题时在文件记录器的事件日志中显示的任何错误消息的详细信息。
 - 涵盖问题时间的文件记录器跟踪。有关如何收集此跟踪的更多信息，请参阅第 356 页的『在多平台上跟踪 Managed File Transfer 独立记录器』。
 - 如果文件记录器在与协调队列管理器通信时迁到问题 (例如，文件记录器的事件日志包含错误消息，其中包含 IBM MQ 原因码，例如 2009-MQRC_CONNECTION_BROKEN)，请使用跟踪规范 =all 来收集跟踪。
 - 对于所有其他问题，请使用跟踪规范 com.ibm.wmqfte=all 来收集跟踪。
 - 包含文件记录器日志文件和配置文件的归档。有关如何创建归档的更多信息，请参阅第 264 页的『创建 MFT 故障诊断信息的归档』。
 - 协调队列管理器的 **runmqras** 输出。有关如何创建输出的更多信息，请参阅第 228 页的『使用 runmqras 自动收集故障诊断信息』。

Multi 收集 MFT 命令问题的信息

如果需要 IBM 支持人员针对 Managed File Transfer (MFT) 命令问题提供帮助，那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

Managed File Transfer 命令的问题包括：

- 无法连接到队列管理器的命令。
- 命令超时。
- 报告错误的命令。

过程

要调查这些问题，请提供以下信息：

- 正在运行的命令。
- 运行命令时登录的用户的用户名。
- 命令的输出。
- 命令正在使用的 Managed File Transfer 版本。
- 命令的跟踪，涵盖发生问题的时间。有关如何收集此跟踪的信息，请参阅第 355 页的『在多平台上跟踪 Managed File Transfer 命令』。
 - 如果该命令在与队列管理器通信时迁到问题（例如，该命令报告包含 IBM MQ 原因码的错误），请使用跟踪规范 `=all` 收集跟踪。
 - 对于所有其他问题，请使用跟踪规范 `com.ibm.wmqfte=all` 来收集跟踪。
- 包含正在运行命令的系统上的配置文件的归档。有关如何创建归档的更多信息，请参阅第 264 页的『创建 MFT 故障诊断信息的归档』。

Multi 创建 MFT 故障诊断信息的归档

收集调查所看到的 Managed File Transfer (MFT) 问题所需的数据后，需要创建包含所有相关文件的归档，并将其发送给 IBM 支持人员。您可以手动创建归档，也可以使用 **fteRAS** 实用程序来创建归档。

关于此任务



警告：如果在系统上配置了大量 Managed File Transfer 代理程序，那么 **fteRAS** 命令可能需要很长时间才能完成。如果发生这种情况，您应该通过将 Managed File Transfer 代理程序的日志和配置目录的内容压缩为 zip 文件来手动创建归档。

过程

- 要使用 **fteRAS** 命令自动归档 Managed File Transfer 文件：

- **Linux** **UNIX** 在 UNIX 和 Linux 上，将任何交互式命令跟踪和 javacores 复制到 `/var/mqm/errors` 目录，然后运行 **fteRAS** 命令，如以下示例中所示：

```
sh> fteRAS /var/mqm/errors
...
BFGCL0604I: fteRAS command completed successfully. Output is stored in /var/mqm/errors/
fteRAS.zip
```

- **Windows** 在 Windows 上，将任何交互式命令跟踪和 javacores 复制到顶级 IBM MQ errors 目录。此目录的实际路径名取决于您正在使用的 IBM MQ 版本。有关更多信息，请参阅 [Windows 上的程序和目录位置](#)。使用正确的系统路径名运行 **fteRAS** 命令，例如：

```
C:\> fteRAS "C:\ProgramData\IBM\MQ\errors"
...
```



```
BFGCL0604I: fteRAS command completed successfully. Output is stored in
C:\ProgramData\IBM\MQ\errors\fteRAS.zip
```

- **IBM i** 在 IBM i 上，将您创建的任何交互式命令跟踪和 javacores (即，来自 **GENJVMDMP** 命令的假脱机文件) 复制到 /QIBM/UserData/mqm/errors，然后从 Qshell 运行 **fteRAS** 命令，如以下示例中所示:

```
===> /QIBM/ProdData/mqm/bin/fteRAS /QIBM/UserData/mqm/errors
...
BFGCL0604I: fteRAS command completed successfully. Output is stored in /QIBM/UserData/mqm/
errors/fteRAS.zip
```

- 要手动归档 Managed File Transfer 文件:

- **Linux** **UNIX** 在 UNIX 和 Linux 上，复制代理程序和记录器 javacores，跟踪，日志，属性和 FFST 文件。包括写入当前目录或其他目录的任何交互式命令跟踪和 javacores 以及以下内容:

```
/var/mqm/mqft/logs/COORDQNAME/*
/var/mqm/mqft/config/COORDQNAME/*
```

- **Windows** 在 Windows 上，复制代理程序和记录器 Javacores，跟踪，日志，属性和 FFST 文件。包括写入当前目录或其他目录的任何交互式命令跟踪和 javacores 以及以下目录的内容。

```
C:\Program Files\IBM\MQ\mqft\logs\COORDQNAME\*
C:\Program Files\IBM\MQ\mqft\config\COORDQNAME\*
```

这些目录的实际路径名取决于您正在使用的 IBM MQ 版本。有关更多信息，请参阅 [Windows 上的程序和目录位置](#)。

- **IBM i** 在 IBM i 上，复制代理程序和记录器 JVM 转储，跟踪，日志，属性和 FFST 文件。包含您创建的任何交互式命令跟踪和 javacores (即，来自 **GENJVMDMP** 命令的假脱机文件) 以及内容:

```
/QIBM/UserData/mqm/mqft/logs/COORDQNAME/*
/QIBM/UserData/mqm/mqft/config/COORDQNAME/*
```

下一步做什么

将收集的信息发送到 IBM。有关更多信息，请参阅 [第 258 页的『在 Multiplatforms 版上收集 Managed File Transfer 问题的信息』](#) 的步骤 [第 259 页的『3』](#)。

Windows 收集 *Microsoft Cluster Service* 问题的信息

如果需要 IBM 支持人员的帮助以解决 IBM MQ 队列管理器在 Windows 上的 Microsoft Cluster Service (MSCS) 下未正确故障转移的问题，那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助查找解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题:

- 您在集群上观察到哪些 MSCS 问题?
- MSCS 问题什么时候开始，什么时候停止?
- 集群成员的名称和地址是什么?
- 这是新集群，还是在问题开始之前对任一集群成员进行了任何更改?

关于此任务

必须在发生集群故障时从系统收集数据，以便提供有关问题的更多信息。

收集故障诊断信息后，可以将其发送到 IBM。

过程

1. 发生问题后生成 MSCS 集群日志。

在其中一个集群成员上:

- a) 启动 PowerShell (或在 DOS 提示符中运行 "PowerShell" 命令)。
- b) 转至 IBM MQ 顶级错误目录。

例如:

```
PS C:\> CD $env:ProgramData\IBM\MQ\Errors
```

- c) 运行 Get-ClusterLog cmdlet 以生成集群中节点的集群日志:

```
PS C:\ProgramData\IBM\MQ\Errors> Get-ClusterLog -Destination
```

2. 运行 IBM MQ **amqmsysn** 实用程序以显示有关集群的两个成员上所有 IBM MQ 可执行文件和库的信息。使用 **目标文件** 按钮将此信息保存到文件, 例如 %TEMP%\MQ.exeinfo.txt。
3. 在问题期间生成 **集群的两个成员上的高详细 MQ 跟踪**。例如:

```
C:\> strmqtrc -e -t all -t detail
    ...
C:\> endmqtrc -a
```

4. 显示集群的两个成员上的注册表检查点:

```
C:\> CLUSTER RESOURCE /CHECKPOINTS > %TEMP%\Cluster.checkpoints.txt
```

5. 检查两个集群成员上的注册表检查点。
6. 在集群的两个成员上保存所有三个事件查看器日志 (系统, 应用程序和安全性)。
7. 将来自步骤 1 和 2 的集群日志, 错误日志和命令输出直接放在集群的每个成员上的顶级 IBM MQ 错误目录中。
步骤 第 266 页的『8』中的自动和手动数据收集过程都会收集在此目录中找到的文件。

8. 收集 IBM MQ 数据。

您可以自动或手动执行此操作:

- 在两个集群成员上使用 **runmqras** 命令自动收集数据, 如 第 228 页的『使用 runmqras 自动收集故障诊断信息』中所述。确保收集 **runmqras trace** 部分, 并指定案例编号, 如以下示例中所示, 用于从队列管理器 QMA 收集输出:

```
runmqras -section trace -qmlist QMA -caseno TS001234567
```

- 或者, 手动收集数据, 如 第 231 页的『手动收集故障诊断信息』中所述。

9. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

有关 FTP 和电子邮件指示信息, 请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例, 请转至 [IBM My Support](#) 站点。

注: 始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈, 请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈, 可以致电 1-800-IBM-SERV。

收集 MQIPT 问题的信息

如果需要使用 MQIPT 向 IBM 支持人员报告问题, 请发送有助于更快解决问题的相关信息。

关于此任务

完成以下步骤以获取所需信息。

过程

1. 同步所涉及的所有计算机（包括所有运行 IBM MQ 和 MQIPT 的计算机）上的系统时钟。
此操作有助于匹配不同跟踪文件中的跟踪条目。
2. 将旧跟踪文件移至备份目录，从而使新跟踪文件仅包含与此问题相关的信息。
3. 对受问题影响的所有路由开启跟踪。
有关更多信息，请参阅第 337 页的『跟踪 IBM MQ Internet Pass-Thru 中的错误』。
4. 运行客户机以重现该问题并创建新的跟踪文件。
5. 发送所有 MQIPT .TRC、.FDC 和 .log 文件的副本。
另外发送 IBM MQ 端点之间使用的所有计算机的简单网络图，包括防火墙、路由器、负载均衡器和服务器。对于每台计算机，请包含其名称、IP 地址和相关端口号。
6. 将收集的信息发送到 IBM。
问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据！
有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。
要打开或更新案例，请转至 [IBM My Support](#) 站点。
注：始终更新案例以指示已发送数据。
如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

相关任务

第 227 页的『[联系 IBM 支持人员](#)』

如果您在 IBM MQ 方面遇到问题需要帮助，可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订、故障诊断和其他新闻的通知。

相关参考

第 53 页的『[对 IBM MQ Internet Pass-Thru 问题进行故障诊断](#)』

您可以执行若干步骤来帮助确定在使用 IBM MQ Internet Pass-Thru (MQIPT) 时可能遇到的任何问题的性质。

Multi

收集发布/预订问题的信息

如果您需要 IBM 支持人员的帮助来解决 IBM MQ 发布/预订未正确传递消息或在 Multiplatforms 版上报告问题的话，那么首先需要收集故障诊断信息以发送给 IBM 支持人员，以帮助找到解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题：

- 您在系统上观察到哪些发布预订问题？
- 发布预订问题开始的时间以及停止的时间？
- 问题中涉及哪些特定主题和订户应用程序？

关于此任务

在发生发布/预订问题时，从系统收集信息以确定原因很重要。

收集故障诊断信息后，可以将其发送到 IBM。





过程

生成故障诊断信息。

1. 如果发布预订问题影响 IBM MQ classes for Java 或 IBM MQ classes for JMS 应用程序，请根据需要生成 [IBM MQ classes for Java 跟踪](#)或 [Java 消息服务跟踪](#)。
如果应用程序正在 WebSphere Application Server 下运行，请遵循该环境的跟踪指示信息。

2. 发生发布/预订问题时生成队列管理器的跟踪。

如果要生成 Java 或 JMS 跟踪，请同时执行此操作。

-   第 305 页的『在 UNIX and Linux 系统上使用跟踪』
-  第 315 页的『使用 Windows 上的跟踪』
-  第 310 页的『IBM i 上的跟踪』

3.  

在 Linux 以及 UNIX 系统上，保存 `mqconfig` 命令的输出。

收集故障诊断信息。

4. 将您在步骤 3 中生成的 `mqconfig` 命令的输出放在顶级 IBM MQ 错误目录中。

步骤 第 268 页的『5』中的自动和手动数据收集过程都会收集在此目录中找到的文件。

5. 收集 IBM MQ 数据。

您可以自动或手动执行此操作：

- 使用 `runmqras` 命令自动收集数据，如第 228 页的『使用 runmqras 自动收集故障诊断信息』中所述。确保收集 `runmqras` 跟踪部分以及队列管理器中的 `defs` 和 `主题` 部分，并指定案例编号，如以下示例中所示，以从队列管理器 QMA 收集输出：

```
runmqras -section defs,topic,trace -qmlist QMA -caseno TS001234567
```

- 或者，手动收集数据，如第 231 页的『手动收集故障诊断信息』中所述。

将故障诊断信息发送到 IBM。

6. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据！

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注：始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

收集 RDQM 问题的信息

复制的数据队列管理器 (RDQM) 正在报告问题或未能在 Linux 上正常工作，您需要收集 MustGather 数据以发送给 IBM 支持人员以帮助查找解决方案。

关于此任务

如果需要收集故障诊断信息以在报告 RDQM 问题时发送给 IBM 支持人员，那么可以使用 `runmqras` 命令来收集诊断数据。

这些指示信息适用于 IBM MQ 9.0.0 Fix Pack 4 和 Linux 上的更高版本。

过程

1. 使用 `mqm` 用户从每个 RDQM 节点收集 `runmqras` 输出：

```
sudo runmqras -qmlist rdqmName -section defs,trace -caseno casenumber
```

其中 `rdqmName` 是队列管理器的名称，`casenumber` 是案例编号，例如 TS001234567。

注意：

- `-caseno` 仅在 IBM MQ 9.0.0 Fix Pack 5 或更高版本，IBM MQ 9.1.0 Fix Pack 1 或更高版本，IBM MQ 9.1.1 或更高版本以及 IBM MQ 9.2.0 或更高版本中工作。

- 如果您正在使用 IBM MQ 9.1.5 或更高版本，那么可以从 **-section** 参数中省略 **trace** 属性。

有关使用 **runmqras** 命令的更多信息，请参阅第 228 页的『使用 runmqras 自动收集故障诊断信息』。

2. 从所有三个节点提供 /var/log/messages 文件。

包含可能包含自问题发生之日起的活动的任何已归档 syslog 文件。

3. 从所有三个节点提供 /var/log/pacemaker.log。

包含可能包含自问题发生之日起的活动的任何已归档 pacemaker.log 文件。

注: **-section trace** 选项收集 root-RDQM.log 和 mqm-RDQM.LOG 文件所在的 /var/mqm/trace 文件夹中的文件。DRBD 日志将写入 /var/log/messages (syslog) 文件。

一个小警告是 /var/log/messages 是缺省 syslog 输出的位置。如果将非缺省位置用于 syslog 目标，请在定制位置中找到 syslog。

相关概念

第 157 页的『对 RDQM 配置问题进行故障诊断』

这些主题提供了可用于对 RDQM 高可用性 (HA) 和灾难恢复 (DR) 配置进行故障诊断的信息。

相关参考

RDQM 高可用性

Multi 收集安全问题的信息

如果 IBM MQ 不正确地允许或拒绝对 Multiplatforms 版上的用户或应用程序进行访问，那么您可能需要收集故障诊断信息以帮助查找解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题：

- 您在系统上观察到哪些安全问题？
- 安全问题从什么时候开始，什么时候停止？
- 涉及哪些特定用户或应用程序和队列管理器对象？
- 此系统先前是否正常工作？
- 自工作以来发生了哪些变化？
- 您尝试使用的用户名和密码有多长？

关于此任务





如果现在发生了安全问题，或者您能够重现该问题，那么可以生成数据以提供有关该问题的更多信息。

收集故障诊断信息后，可以将其发送到 IBM。

过程

1. 发生安全性问题时生成队列管理器的跟踪。

如果可能，请在跟踪之前发出 **runmqsc** 命令 **REFRESH SECURITY**，以便跟踪将显示查询操作系统的队列管理器以获取有关用户的详细信息。

-   第 305 页的『在 UNIX and Linux 系统上使用跟踪』
-  第 315 页的『使用 Windows 上的跟踪』
-  第 310 页的『IBM i 上的跟踪』

2. 显示有关用户的信息，尤其是有关用户所属的组的信息。

例如：

- **Linux** **UNIX** 要在 UNIX 和 Linux 上显示用户 watson:

```
sh> id watson > /tmp/watson.id.txt
sh> groups watson > /tmp/watson.groups.txt
```

- **Windows** 要在 Windows 上显示用户 "Thomas Watson":

```
C:\> NET USER "Thomas Watson" > %TEMP%\watson.user.txt
```

- **IBM i** 要在 IBM i 命令行上显示用户 WATSON:

```
====> DSPUSRPRF USER(WATSON) OUTPUT(*PRINT)
```

然后使用 **WRKSPLF** 选项 5 显示 QPUSRPRF 中的作业记录

3. 收集 IBM MQ 数据。

您可以自动或手动收集此操作:

- 使用 **runmqras** 命令自动收集数据, 如第 228 页的『使用 runmqras 自动收集故障诊断信息』中所述。请确保收集 **runmqras** defs 和 trace (如果跟踪了问题) 部分, 并指定您的案例编号, 如以下示例中所示:

```
runmqras -section defs,cluster,trace -qmlist QMA -caseno TS001234567
```

- 或者, 手动收集数据, 如第 231 页的『手动收集故障诊断信息』中所述。

注: 如果此连接的其中一方不是队列管理器, 请收集该客户机的适用日志。

4. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

有关 FTP 和电子邮件指示信息, 请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例, 请转至 [IBM My Support](#) 站点。

注: 始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈, 请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈, 可以致电 1-800-IBM-SERV。

相关任务

第 166 页的『对安全性问题进行故障诊断』
故障诊断信息可帮助您解决与安全性相关的问题。

Multi 收集 TLS 通道问题的信息

如果 IBM MQ 队列管理器或客户机应用程序未能使用 TLS on Multiplatforms 建立安全通道, 那么您可能需要收集故障诊断信息以帮助查找解决方案。

开始之前

在开始此任务之前, 请回答有关此问题的以下问题:

- 您在系统上观察到哪些 TLS 通道问题?
- TLS 通道问题何时开始以及何时停止?
- 问题涉及哪些具体渠道和证书?
- 此通道先前是否使用 TLS 或这是新配置?
- 如果通道先前正在工作, 那么更改了哪些内容?
- 通道是否在没有 TLS 的情况下工作?

提交来自 IBM MQ 连接两侧的输出。以下示例来自基于 KDB 密钥库的系统。对于使用其他格式的客户机, 请参阅文档以获取有关如何列示密钥库的信息的相应格式。

- 密钥库位置和许可权

– **Linux** **UNIX** UNIX 和 Linux 命令行:

```
ls -la <DIRECTORY OF KEYSTORE>
```

– **Windows** Windows Powershell 命令:

```
Get-Acl <DIRECTORY OF KEYSTORE> |
Format-List
```

- **ULW** 密钥库证书列表 UNIX, Linux 和 Windows:

```
runmqakm -cert -list -v -db <KEYSTORE FILE> -stashed
```

- 证书在未来 90 天内到期或到期:

```
runmqakm -cert -list -expiry 90 -db <KEYSTORE NAME> -stashed
```

关于此任务

如果 TLS 通道问题正在发生, 或者您能够重现该问题, 那么可以生成数据以提供有关该问题的更多信息。收集故障诊断信息后, 可以将其发送到 IBM。

过程

1. 发生 TLS 问题时生成队列管理器的跟踪。

除非您的支持代表以不同方式通知您, 否则队列管理器 TLS 跟踪的正确选项为 `-t all -t detail`:

- **Linux** **UNIX** [第 305 页的『在 UNIX and Linux 系统上使用跟踪』](#)

- **Windows** [第 315 页的『使用 Windows 上的跟踪』](#)

- **IBM i** [第 310 页的『IBM i 上的跟踪』](#)

2. 在通道的另一端同时生成 IBM MQ 跟踪, 无论它是另一个队列管理器还是客户机应用程序:

- **Linux** **UNIX** [第 305 页的『在 UNIX and Linux 系统上使用跟踪』](#)

- **Windows** [第 315 页的『使用 Windows 上的跟踪』](#)

- **IBM i** [第 310 页的『IBM i 上的跟踪』](#)

- **z/OS** [z/OS CHIN 跟踪](#)

3. 收集 IBM MQ 数据。

您可以自动或手动收集此操作:

- 使用 `runmqras` 命令自动收集数据, 如 [第 228 页的『使用 runmqras 自动收集故障诊断信息』](#) 中所述。请确保收集 `runmqras defs` 和 `trace` (如果跟踪了问题) 部分, 并指定您的案例编号, 如以下示例中所示:

```
runmqras -section defs,cluster,trace -qmlist QMA -caseno TS001234567
```

- 或者, 手动收集数据, 如 [第 231 页的『手动收集故障诊断信息』](#) 中所述。

注: 如果此连接的其中一方不是队列管理器, 请收集该客户机的适用日志。

4. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

有关 FTP 和电子邮件指示信息, 请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注：始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

相关任务

第 166 页的『[对安全性问题进行故障诊断](#)』故障诊断信息可帮助您解决与安全性相关的问题。

Multi 收集用于触发问题的信息

如果需要 IBM 支持人员提供帮助以解决 IBM MQ 未在多平台上正确触发应用程序或通道的问题，那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助查找解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题：

- 您在系统上观察到什么触发问题？
- 触发问题是什么时候开始的，什么时候停止的？
- 未触发哪个队列，应该已启动哪些通道或进程？





关于此任务

如果现在正在发生触发问题，或者如果您能够重现该问题，那么可以生成数据以提供有关该问题的更多信息。

收集故障诊断信息后，可以将其发送到 IBM。

过程

1. 发生触发问题时生成队列管理器的跟踪：

-   [第 305 页的『在 UNIX and Linux 系统上使用跟踪』](#)
-  [第 315 页的『使用 Windows 上的跟踪』](#)
-  [第 310 页的『IBM i 上的跟踪』](#)

2. 收集 IBM MQ 数据。

您可以自动或手动执行此操作：

- 使用 **runmqras** 命令自动收集数据，如第 228 页的『[使用 runmqras 自动收集故障诊断信息](#)』中所述，以收集通道两侧的数据。确保收集 **runmqras** defs 和 trace 部分，并指定案例编号，如以下示例中所示：

```
runmqras -section defs,trace -qmlist QMA -caseno TS001234567
```

- 或者，手动收集数据，如第 231 页的『[手动收集故障诊断信息](#)』中所述。

3. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据！

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注：始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

相关任务

第 124 页的『对消息问题进行故障诊断』



在 z/OS 上收集故障诊断信息

有关如何收集 IBM MQ for z/OS 的故障诊断信息的概述。

关于此任务

注: 除了本部分中描述的信息外, IBM 支持人员可能会逐个案例地请求进一步的信息。

过程

- 有关如何收集 IBM MQ for z/OS 的特定问题区域的故障诊断和诊断信息的信息, 请参阅以下主题:
 - [ABEND](#)
 -   [Advanced Message Security \(AMS\)](#)
 - [客户机连接](#)
 - [CICS 适配器](#)
 - [CICS 网桥](#)
 - [通道](#)
 - [集群](#)
 - [数据转换](#)
 - [数据库](#)
 - [死信队列消息](#)
 - [错误消息](#)
 - [IBM WebSphere MQ File Transfer Edition \(FTE\): 请参阅 \[Managed File Transfer for z/OS \\(MFT for z/OS\\)\]\(#\)](#)
 - [挂起和高 CPU](#)
 - [IBM MQ Explorer](#)
 - [IMS](#)
 - [安装和卸载](#)
 - [Java 和 JMS](#)
 - [Managed File Transfer for z/OS \(MFT for z/OS\)](#)
 - [性能](#)
 - [发布/预订](#)
 - [安全性](#)
 - [共享通道](#)
 - [共享队列](#)
 - [关闭问题](#)
 - [启动问题](#)
 - [TLS 通道 \(以前称为 SSL\)](#)
 - [触发通道](#)
 - [触发程序](#)
- 有关所有其他问题, 请参阅 [在 WebSphere MQ for z/OS 中收集常规问题或未知问题的故障诊断数据](#)。

相关任务

第 227 页的『在多平台上收集故障诊断信息』

有关如何收集 IBM MQ on Multiplatforms 的故障诊断信息的概述。

收集 z/OS 上的异常终止问题的信息

如果需要 IBM 支持人员的帮助以解决 IBM MQ for z/OS 上的异常终止问题，那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助查找解决方案。

过程

1. 收集以下常规信息:

- IBM MQ 版本, 发行版和维护级别
- 操作系统版本, 发行版和维护级别
- 相关产品版本和发行版级别 (如果适用)

2. 针对此问题收集以下故障诊断 (MustGather) 信息:

a) 收集以下必需信息:

作业日志

您可以在 Syslog, MSTR 作业日志和 CHIN 作业日志中找到 IBM MQ for z/OS 作业日志。作业日志名为 `xxxxxMSTR` 和 `xxxxxCHIN`, 其中 `xxxx` 是 IBM MQ 子系统标识 (SSID)。有关更多信息, 请参阅 [为 IBM MQ for z/OS 作业创建包含 JES2 作业日志的打印数据集](#)。

在发生故障时生成的转储

IBM MQ 转储位于系统转储数据集中 (请参阅步骤 [第 274 页的『4』](#))。

b) (可选) 还收集 z/OS LOGREC 报告 (请参阅 [第 215 页的『z/OS 上的 SYS1.LOGREC 信息』](#))。

3. 在 IBM 支持站点 中搜索已知问题。

您可以使用诸如消息号和错误代码之类的症状进行搜索。

4. 查看在发生故障时生成的转储。

IBM MQ 转储位于系统转储数据集中, 并可通过其标题进行标识。IBM MQ 请求的转储的标题以队列管理器的四字符子系统名称开头。例如:

```
CSQ1,ABN=5C6-00E20016,U=SYSOPR,C=MQ900.910.DMC
-CSQIALLC,M=CSQGFRCV,LOC=CSQSLD1.CSQSVSTK+00000712
```

转储标题可提供有关异常终止的充足信息以及解决问题的原因码。有关更多信息, 请参阅 [第 212 页的『在 z/OS 上分析转储和解释转储标题』](#)。

有关 IBM MQ for z/OS 使用的两个系统异常终止完成代码 X'5C6'和 X'6C6'的更多信息, 请参阅 [第 182 页的『IBM MQ for z/OS 异常终止』](#)。您还可以使用转储中列出的异常终止代码, 原因码和程序名在 [IBM 支持站点](#) 上搜索已知问题。

5. 请检查系统日志 (syslog)。

根据转储命令中指定的注释, 通信转储可能不包含队列管理器名称。检查系统日志以获取 [IEA611I](#) 或 [IEA911E](#) 消息, 以确定转储数据集名称, 并查看转储是完成还是部分完成。例如:

```
IEA611I COMPLETE DUMP ON DUMP.MQT1MSTR.DMP00074
DUMPID=074 REQUESTED BY JOB(MQT1MSTR)
FOR ASID(005E)

IEA911E PARTIAL DUMP ON SYS1.MCEVS4.DMP00039
DUMPID=039 REQUESTED BY JOB(DMSGTODI)
FOR ASID(00D2)
```

如果问题的原因是磁盘空间不足, 那么转储中可能没有足够的信息来诊断问题。

转储可能被转储分析和消除 (DAE) 抑制。在这种情况下, 某些症状可能不会出现在系统日志 (syslog) 或作业日志中, 但会出现在 Logrec 中 (请参阅 [第 215 页的『z/OS 上的 SYS1.LOGREC 信息』](#))。有关 DAE 管理的更多信息, 请参阅 [生成禁止的转储](#)。

6. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

有关 FTP 和电子邮件指示信息, 请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例, 请转至 [IBM My Support](#) 站点。

注: 始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈, 请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈, 可以致电 1-800-IBM-SERV。

MQ Adv. z/OS 收集 z/OS 上的 AMS 问题的信息

如果需要 IBM 支持人员的帮助以解决 IBM MQ for z/OS 上的 Advanced Message Security (AMS) 问题, 那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助查找解决方案。

开始之前

在开始此任务之前, 请回答有关此问题的以下初始问题:

- 您在系统上观察到什么 AMS 错误?
- 详细的 AMS 消息流是什么?
- AMS 问题何时开始以及何时停止?
- 涉及哪些特定用户或应用程序以及队列管理器队列? IBM MQ 安全策略以及 AMS 正在使用的文件。提供有关如何设置这些文件的详细信息。

过程

1. 对于配置问题, 请收集:

- <user>\drq.ams.keyring 的 RACF 列表-针对所涉及的每个应用程序用户, 例如, 放置和获取应用程序
- RACDCERT 标识 (用户标识) LISTRING(drq.ams.keyring)
- <AMSUSER>\drq.ams.keyring 的 RACF 列表
- RACDCERT 标识 (CSQ1AMSM) LISTRING(drq.ams.keyring) -将 CSQ1 替换为队列管理器的名称
- CSQOUTIL 策略列表:
 - **dspmqspl -m "CSQ1"**-将 CSQ1 替换为队列管理器的名称
 - **dspmqspl -m "CSQ1" -p "PROBLEMQ"**-将 CSQ1 替换为队列管理器的名称, 将 PROBLEMQ 替换为队列的名称

2. 对于与 AMS 服务器到服务器消息通道代理程序拦截相关的问题, 请收集通道定义并显示输出。

3. 对于其他错误和/或异常终止故障, 请另外收集:

- 队列管理器, 通道启动程序, AMSM 和放置/获取应用程序地址空间的转储。
请参阅 [第 198 页的『IBM MQ for z/OS 转储』](#) 以获取更多信息。
- 队列管理器, 通道启动程序, AMSM 地址空间和放置/获取应用程序作业 (如果适用) 的作业日志。
- AMS (和/或 IBM MQ) 内部跟踪。
请参阅 [第 318 页的『在 z/OS 上使用跟踪进行问题确定』](#) 以获取更多信息。
- AMS 调试跟踪 (写入 AMSM 地址空间的 SYSOUT 或放入/获取应用程序作业日志)。
请参阅步骤 [第 275 页的『4』](#), 以获取有关如何捕获 AMS 跟踪的信息。
- GSKit 跟踪。

有关如何捕获 GSKit 跟踪的信息, 请参阅步骤 [第 275 页的『5』](#)。

4. 捕获 AMS 跟踪。

有关更多信息, 请参阅 [第 327 页的『为 AMSM 系统启用内部跟踪』](#)。

5. 在系统上捕获 GSKit 跟踪, 以帮助诊断密钥库和证书的问题。

有关更多信息, 请参阅 [第 328 页的『在 z/OS 上使用 AMS 时, 对与证书和密钥相关的问题使用 GSKit 跟踪』](#)。

6. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注：始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

相关概念

第 38 页的『[对 AMS 问题进行故障诊断](#)』

提供了信息来帮助您找到并解决与 Advanced Message Security 相关的问题。

收集 Managed File Transfer for z/OS 问题的信息

如果在 z/OS 代理程序，记录器或命令的 Managed File Transfer (MFT) 报告问题或无法正常工作时需要 IBM 支持人员提供帮助来解决问题，那么您首先需要收集故障诊断信息以发送给 IBM 支持人员来帮助查找解决方案。需要的信息取决于您看到的问题。

过程

1. 收集您看到的问题类型所需的信息：

- [Managed File Transfer for z/OS 代理程序问题](#)
- [Managed File Transfer for z/OS 协议网桥代理问题](#)
- [Managed File Transfer for z/OS 资源监视器问题](#)
- [Managed File Transfer for z/OS 受管传输问题](#)
- [Managed File Transfer for z/OS 数据库记录器问题](#)
- [Managed File Transfer for z/OS 命令问题](#)

2. 收集调查问题所需的 Managed File Transfer 数据后，创建包含所有相关文件的归档。

有关更多信息，请参阅第 281 页的『[为 MFT for z/OS 创建故障诊断信息归档](#)』。

3. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据！

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注：始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

相关参考

第 77 页的『[对 Managed File Transfer 问题进行故障诊断](#)』

使用以下参考信息可帮助您诊断 Managed File Transfer 中的错误：

收集 MFT for z/OS 代理程序问题的信息

如果需要 IBM 支持人员针对 z/OS 上的 Managed File Transfer (MFT) 代理程序问题提供帮助，那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

Managed File Transfer 代理程序问题包括：

- 未能与其代理队列管理器连接或断开连接的代理。
- 代理程序挂起。
- 代理程序意外停止。
- 正在恢复的代理程序。

- **fteListAgents** 或 **fteShowAgentDetails** 命令或 IBM MQ Explorer Managed File Transfer 插件显示了代理程序的不正确状态信息或过期状态信息。
- 代理程序未能报告任何状态信息。

过程

1. 最初，请查看以下主题以了解它们是否帮助您解决问题：

- 第 82 页的『对代理程序状态问题进行故障诊断』
- 第 102 页的『对 java.lang.OutOfMemoryError 问题进行故障诊断』

2. 如果仍需要帮助，请收集以下信息并将其发送给 IBM 支持人员：

- 代理的名称。
- 代理队列管理器的名称。
- 代理程序正在使用的 Managed File Transfer for z/OS 版本。
- 代理队列管理器的 IBM MQ for z/OS 版本。
- 代理程序的启动方式的详细信息 (例如，它是否作为启动式任务运行?)。
- 发生问题时在代理程序的事件日志 (output0.log) 中看到的任何错误消息。
- 涵盖问题时间的代理程序跟踪。有关如何收集跟踪的更多信息，请参阅第 358 页的『跟踪 Managed File Transfer for z/OS 代理』。
 - 如果代理在与其代理队列管理器通信时迁到问题 (例如，代理的事件日志包含错误消息，其中包含 IBM MQ 原因码，例如 2009-MQRC_CONNECTION_BROKEN)，请使用跟踪规范 =all 收集跟踪。
 - 对于所有其他问题，请使用跟踪规范 com.ibm.wmqfte=all 来收集跟踪。
- 如果代理程序挂起，那么三个 Javacore 相隔 30 秒。要执行此操作，请运行带有 -jc 选项集的 **fteSetAgentTraceLevel** 命令，如以下示例中所示：

```
fteSetAgentTraceLevel -jc <agent_name>
```

如果该命令未导致代理程序生成 Javacore，那么应该向代理程序进程发送 SIGQUIT 信号。

- 包含代理程序的日志文件，配置文件，跟踪文件和 Javacores (如果适用) 的归档。有关如何创建归档的更多信息，请参阅第 281 页的『为 MFT for z/OS 创建故障诊断信息归档』。
- 协调队列管理器和代理队列管理器的作业日志。

z/OS 收集 MFT for z/OS 协议网桥代理问题的信息

如果需要 IBM 支持人员针对 z/OS 上的 Managed File Transfer (MFT) 协议网桥代理问题提供帮助，那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

Managed File Transfer 协议网桥代理问题包括：

- 代理程序未能连接到远程文件服务器或无法与远程文件服务器断开连接。
- 与远程文件服务器之间的受管传输失败。

过程

1. 最初，请查看第 94 页的『对协议网桥代理问题进行故障诊断』中的信息，以了解这是否有助于您解决问题。
2. 如果仍需要帮助，请收集以下信息并将其发送给 IBM 支持人员：
 - 协议网桥代理的名称。
 - 协议网桥代理队列管理器的名称。
 - 协议网桥代理正在使用的 Managed File Transfer for z/OS 版本。

- 协议网桥代理队列管理器的 IBM MQ for z/OS 版本。
- 远程文件服务器系统的主机名。
- 远程文件服务器的产品和版本信息。
- 代理程序用于与远程文件服务器 (即 FTP, FTPS 或 SFTP) 通信的协议。
- 协议网桥代理配置文件 (ProtocolBridgeProperties.xml) 中远程文件服务器的条目。
- 发生问题时在代理程序的事件日志 (output0.log) 中看到的任何错误消息。
- 协议网桥代理日志文件, 其中正在使用的协议的日志级别设置为 on。有关如何设置日志级别的更多信息, 请参阅 `fteSetAgentLog` 级别 (打开或关闭对某些 MFT 代理操作的文件的日志记录)。
- 包含协议网桥代理的日志文件和配置文件的归档。有关如何创建归档的更多信息, 请参阅 [第 281 页的『为 MFT for z/OS 创建故障诊断信息归档』](#)。

收集 MFT for z/OS 资源监视器问题的信息

如果需要 IBM 支持人员提供 z/OS 上 Managed File Transfer (MFT) 资源监视器问题的帮助, 那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

Managed File Transfer 资源监视器问题包括:

- 资源监视器停止轮询。
- 资源监视器正在轮询, 并且不会在任何项 (文件或消息) 上触发。
- 资源监视器未向代理提交受管传输请求。
- 资源监视器意外停止。

过程

1. 最初, 请查看 [第 95 页的『对资源监视器问题进行故障诊断』](#) 中的信息, 以了解这是否有助于您解决问题。
2. 如果仍需要帮助, 请收集以下信息并将其发送给 IBM 支持人员:
 - 代理的名称。
 - 代理队列管理器的名称。
 - 代理程序正在使用的 Managed File Transfer for z/OS 版本。
 - 代理队列管理器的 IBM MQ for z/OS 版本。
 - 资源监视器的名称。
 - 监视器正在轮询的资源 (队列或目录) 的名称。
 - 监视器的触发条件。
 - 监视器的任务 XML。
 - 监视器未触发的任何项的详细信息。
 - 资源监视器日志文件 (例如, `resmonevent0.log`), 其中资源监视器的日志级别设置为 VERBOSE。有关如何创建日志文件的更多信息, 请参阅 [日志记录 MFT 资源监视器](#)。

如果监视器正在轮询, 但未卡住, 那么日志文件应包含至少三个轮询的条目。

 - 包含代理程序配置文件以及代理程序和资源监视器的日志文件的归档。有关如何创建归档的更多信息, 请参阅 [第 281 页的『为 MFT for z/OS 创建故障诊断信息归档』](#)。

收集 MFT for z/OS 受管传输问题的信息

如果需要 IBM 支持人员针对 z/OS 上的 Managed File Transfer (MFT) 受管传输问题提供帮助, 那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

与 Managed File Transfer 受管传输相关的问题包括:

- 受管传输意外失败。
- 正在进入恢复且未完成的受管传输。
- 受管传输卡住。

过程

1. 最初, 请查看 [第 88 页的『对受管传输问题进行故障诊断』](#) 中的信息, 以了解这是否有助于您解决问题。
2. 如果仍需要帮助, 请收集以下信息并将其发送给 IBM 支持人员:
 - 受管传输的源代理的名称。
 - 源代理队列管理器的名称。
 - 源代理正在使用的 Managed File Transfer 或 Managed File Transfer for z/OS 的版本。
 - 源代理队列管理器的 IBM MQ 或 IBM MQ for z/OS 版本。
 - 受管传输的目标代理的名称。
 - 目标代理队列管理器的名称。
 - 目标代理正在使用的 Managed File Transfer 或 Managed File Transfer for z/OS 版本。
 - 目标代理队列管理器的 IBM MQ 或 IBM MQ for z/OS 版本。
 - 如果源和目标代理队列管理器不同, 那么将详细说明队列管理器如何连接在一起 (即, 通过发送方/接收方通道或 IBM MQ 集群)。
 - 受管传输的传输标识。
 - 如何创建受管传输请求的详细信息 (即, 是由资源监视器, **fteCreateTransfer** 命令, IBM MQ Explorer Managed File Transfer 插件还是其他内容生成的?)。
 - 与源代理或目标代理的事件日志 (output0.log) 中的受管传输相关的任何错误消息的详细信息。
 - 来自源代理和目标代理的跟踪, 涵盖发生问题的时间。有关如何收集跟踪的更多信息, 请参阅 [第 352 页的『在多平台上跟踪 Managed File Transfer 代理』](#) 或 [第 358 页的『跟踪 Managed File Transfer for z/OS 代理』](#)。应该使用跟踪规范 `com.ibm.wmqfte=all` 来收集跟踪。
 - 来自源代理程序的归档, 其中包含代理程序的日志文件和配置文件, 以及来自目标代理程序的归档, 其中包含代理程序的日志文件和配置文件。有关如何收集源和目标代理的归档的更多信息, 请参阅 [第 264 页的『创建 MFT 故障诊断信息的归档』](#) 或 [第 281 页的『为 MFT for z/OS 创建故障诊断信息归档』](#)。
 - 源代理队列管理器和目标代理队列管理器的 **runmqras** 输出 (如果它们在 z/OS 以外的平台上运行)。有关如何收集源代理队列管理器和目标代理队列管理器的 **runmqras** 输出的更多信息, 请参阅 [第 228 页的『使用 runmqras 自动收集故障诊断信息』](#)。

在调查与受管传输相关的问题时, 绘制一个简单的图通常很有用, 如以下示例中所示, 该图显示了代理和代理队列管理器。此图允许您和 IBM 支持人员查看如何连接代理和代理队列管理器, 这可帮助确定 IBM MQ 网络中可能导致受管传输进入恢复或卡住的问题。

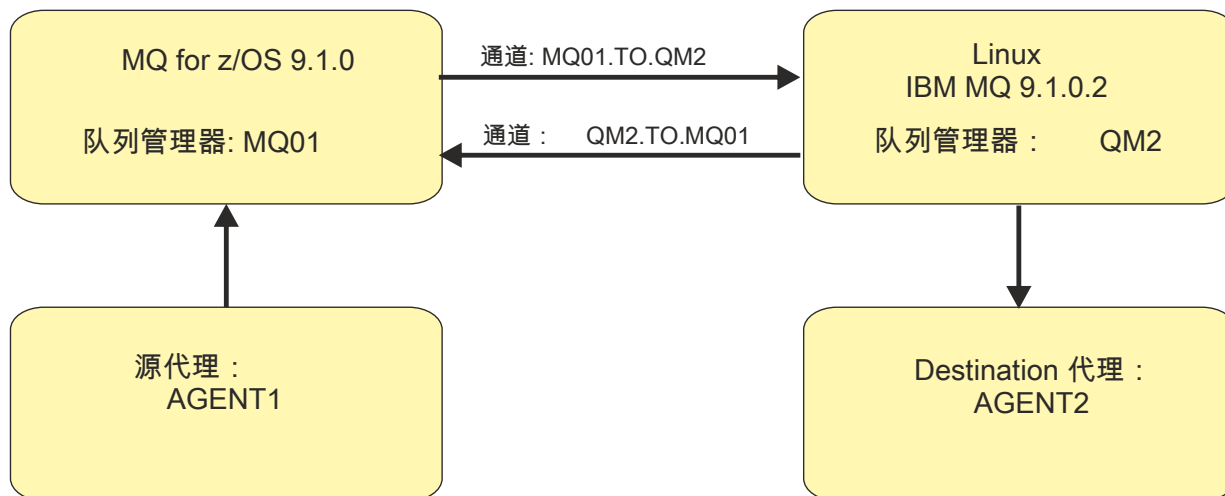


图 52: 显示如何连接源代理 AGENT1 和目标代理 AGENT2 及其代理队列管理器 MQ01 和 QM2 的简单图示例。

z/OS 收集 MFT for z/OS 数据库记录器问题的信息

如果您需要 IBM 支持人员提供帮助以解决 z/OS 上的 Managed File Transfer (MFT) 数据库记录器问题，那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

Managed File Transfer 数据库记录器问题包括:

- 数据库记录器无法连接到协调队列管理器。
- 数据库记录器无法连接到数据库。
- 数据库记录器不会更新数据库。

过程

1. 最初，请查看第 108 页的『对记录器问题进行故障诊断』中的信息，以了解这是否有助于您解决问题。
2. 如果仍需要帮助，请收集以下信息并将其发送给 IBM 支持人员：
 - 数据库记录器的名称。
 - 数据库记录器要连接到的协调队列管理器的名称。
 - 数据库记录器正在使用的 Managed File Transfer for z/OS 版本。
 - 协调队列管理器的 IBM MQ for z/OS 版本。
 - 数据库记录器正在使用的数据库类型。
 - 发生问题时在数据库记录器的事件日志中显示的任何错误消息的详细信息。
 - 涵盖问题时间的数据库记录器跟踪。有关如何收集此跟踪的更多信息，请参阅第 366 页的『跟踪 Managed File Transfer for z/OS 独立数据库记录器』。
 - 如果数据库记录器在与协调队列管理器通信时迁到问题 (例如，数据库记录器的事件日志包含错误消息，其中包含 IBM MQ 原因码，例如 2009-MQRC_CONNECTION_BROKEN)，请使用跟踪规范 =all 收集跟踪。
 - 对于所有其他问题，请使用跟踪规范 com.ibm.wmqfte=all 来收集跟踪。
 - 包含数据库记录器日志文件和配置文件的归档。有关如何创建归档的更多信息，请参阅第 281 页的『为 MFT for z/OS 创建故障诊断信息归档』。
 - 协调队列管理器的作业记录。

收集 MFT for z/OS 命令问题的信息

如果需要 IBM 支持人员针对 z/OS 上的 Managed File Transfer (MFT) 命令问题提供帮助，那么需要收集故障诊断信息并将其发送到 IBM。

关于此任务

Managed File Transfer 命令的问题包括：

- 无法连接到队列管理器的命令。
- 命令超时。
- 报告错误的命令。

过程

要调查这些问题，请提供以下信息：

- 正在运行的命令。
- 命令是从 z/OS UNIX System Services (USS) 运行还是通过 JCL 运行。
- 运行命令时登录的用户的用户名。
- 命令的输出。
- 命令正在使用的 Managed File Transfer for z/OS 版本。
- 命令的跟踪，涵盖发生问题的时间。有关如何收集此跟踪的信息，请参阅 [第 363 页的『跟踪 Managed File Transfer for z/OS 命令』](#)。
 - 如果该命令在与队列管理器通信时迁到问题 (例如，该命令报告包含 IBM MQ 原因码的错误)，请使用跟踪规范 `=all` 收集跟踪。
 - 对于所有其他问题，请使用跟踪规范 `com.ibm.wmqfte=all` 来收集跟踪。
- 包含正在运行命令的系统上的配置文件的归档。有关如何创建归档的更多信息，请参阅 [第 281 页的『为 MFT for z/OS 创建故障诊断信息归档』](#)。

为 MFT for z/OS 创建故障诊断信息归档

收集调查 z/OS 上看到的 Managed File Transfer (MFT) 问题所需的数据后，需要创建包含所有相关文件的归档，并将其发送给 IBM 支持人员。您可以手动创建归档，也可以使用 **fteRAS** 实用程序来创建归档。

关于此任务



警告：如果在系统上配置了大量 Managed File Transfer for z/OS 代理程序，那么 **fteRAS** 命令可能需要很长时间才能完成。如果发生这种情况，您应该通过压缩 Managed File Transfer 代理程序的日志和配置目录的内容来手动创建归档。

过程

- 要使用 **fteRAS** 命令自动归档 Managed File Transfer 文件：
 - 将运行命令时生成的任何跟踪文件复制到 Managed File Transfer for z/OS 安装的 BFG_DATA 目录中。
 - 如果您正在使用 z/OS UNIX System Services (USS)，请运行 **fteRAS** 命令。
 - 如果您正在使用 JCL：
 - 找到包含用于安装的 JCL 的数据集。
 - 提交数据集中的 BFGGRAS 成员。
- 要手动归档 Managed File Transfer 文件：
 - 将代理程序和记录器 `javacores`，跟踪，日志，属性和 FFST 文件复制到临时目录中。包括写入当前目录或其他目录的任何交互式命令跟踪和 `javacores` 以及以下目录的内容：

```
BFG_DATA/mqft/config/coordination_qmgr_name
```

BFG_DATA/mqft/logs/coordination_qmgr_name

- 创建包含临时目录内容的归档。

下一步做什么

将收集的信息发送到 IBM。有关更多信息，请参阅第 276 页的『收集 Managed File Transfer for z/OS 问题的信息』的步骤第 259 页的『3』。

在 z/OS 上收集共享队列问题的信息

如果需要 IBM 支持人员的帮助来解决 IBM MQ for z/OS 上的共享队列问题，那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助查找解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题：

- 共享队列的名称有问题吗？
- 与有问题的共享队列关联的 IBM MQ 耦合设施结构 (CFSTRUCT) 的名称是什么？
- 与问题关联的消息标识是什么？
- 队列共享组的名称是什么？
- 问题是在什么时候发生的？
- 涉及队列共享组中的哪个队列管理器？

关于此任务

如果可以重现共享队列问题或问题正在发生，那么可以生成数据以提供有关该问题的更多信息。

收集故障诊断信息后，可以将其发送到 IBM。

过程

1. 发生问题时生成以下跟踪：

- a. 生成 GTF 跟踪。
- b. 生成 MSTR 内部跟踪。
- c. 生成 CHIN 跟踪。

2. 收集数据。

以下步骤包含同时转储应用程序结构和 IBM MQ 管理结构的示例。

a) 记录软件的版本，发行版和维护级别：

- IBM MQ: 在 MSTR 作业日志中的 CSQY000I 消息中查找版本。
- z/OS 操作系统: 在 SDSF 中的 /D IPLINFO 输出中查找版本。
- 与该问题相关的任何其他产品: 请在作业记录中查找该产品的版本。

b) 收集 IBM MQ MSTR 和 CHIN 作业日志，并 (可选) 收集系统日志。

c) 收集 z/OS LOGREC 报告。

有关更多信息，请参阅第 215 页的『z/OS 上的 SYS1.LOGREC 信息』。

d) 保存从 IBM MQ 生成的任何 z/OS 转储。

IBM MQ 转储位于系统转储数据集中，并可通过其标题进行标识。IBM MQ 请求的转储的标题以队列管理器的四字符子系统名称开头。例如：

```
CSQ1,ABN=5C6-00E20016,U=SYSOPR,C=MQ900.910.DMC  
-CSQIALLC,M=CSQGFRCV,LOC=CSQSLD1.CSQSVSTK+00000712
```

根据转储命令中指定的注释，通信转储可能不包含队列管理器名称。检查系统日志以获取 [IEA611I](#) 或 [IEA911E](#) 消息，以确定转储数据集名称，并查看转储是完成还是部分完成。例如：

```
IEA611I COMPLETE DUMP ON DUMP.MQT1MSTR.DMP00074
DUMPID=074 REQUESTED BY JOB(MQT1MSTR)
FOR ASID(005E)

IEA911E PARTIAL DUMP ON SYS1.MCEVS4.DMP00039
DUMPID=039 REQUESTED BY JOB(DMSGTODI)
FOR ASID(00D2)
```

e) 收集应用程序结构和 IBM MQ 管理结构的耦合设施结构转储：

```
/DUMP COMM=(title)
/R nnn,SDATA=(ALLNUC,LPA,PSA,RGN,SQA,TRT,CSA,XESDATA,COUPLE,GRSQ),CONT
/R nnn,JOBNAME=(ssidMSTR),CONT
/R nnn,STRLIST=(STRNAME=QSGnameStructurename,(LISTNUM=ALL,
ADJUNCT=CAPTURE,ENTRYDATA=UNSER),EVENTQS,(EMCONTROLS=ALL),
/R nnn,STRNAME=QSGnameCSQ_ADMIN,(LISTNUM=ALL,ADJUNCT=CAPTURE,
ENTRYDATA=UNSER),EVENTQS,(EMCONTROLS=ALL)),END
```

其中 *ssid* 是队列管理器的子系统标识。

QSGnameStructurename 的示例是 *QSG1APPLICATION*，其中 *QSG1* 是队列共享组名。

QSGnameCSQ_ADMIN 的示例为 *QSG1CSQ_ADMIN*。

以下示例显示仅转储应用程序结构：

```
/DUMP COMM=(title)
/R xx,STRLIST=(STRNAME=QSGnameStructurename,(LISTNUM=ALL,
ADJUNCT=CAPTURE,ENTRYDATA=UNSER),EVENTQS,(EMCONTROLS=ALL))
```

QSGnameStructurename 的示例是 *QSG1APPLICATION*，其中 *QSG1* 是队列共享组名。

3. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据！

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注：始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

在 z/OS 上收集性能问题的信息

如果需要 IBM 支持人员的帮助以解决 IBM MQ for z/OS 上的性能问题，那么首先需要收集故障诊断信息以发送给 IBM 支持人员以帮助查找解决方案。

开始之前

在开始此任务之前，请回答有关此问题的以下问题：

- 问题有什么影响，比如说高 CPU 或响应延迟？
- 问题最早是在什么时候发生的？
- 是否应用了软件或硬件维护？
- 问题是一次性故障还是再次发生？
- 问题中涉及的队列管理器，队列，通道或其他作业的名称是什么？
- 查看了 [对 MQ 性能问题进行故障诊断中的信息](#)。

关于此任务

The IBM Software Support Handbook states that analyzing performance is one of the activities that often require some form of Advance Support Offering. 如果分析显示产品中存在可疑缺陷，并且您可以重现性能问题或问题正在发生，那么可以生成数据以提供有关问题的更多信息，以便 IBM MQ 支持团队可以诊断问题。

收集故障诊断信息后，可以将其发送到 IBM。

过程

1. 如果可以重现性能问题或问题正在发生，请生成数据以提供有关该问题的更多信息：
 - a. 在发生问题时 [生成 GTF 跟踪](#)。
 - b. 生成 MSTR 内部跟踪，并在发生问题时将其捕获到转储中。
 - c. [生成 CHIN 跟踪](#)，并在发生问题时使用 MSTR 跟踪在同一转储中捕获该跟踪。
 - d. 如果网络性能存在问题，那么在发生问题时，[会在通道的另一端同时生成 z/OS TCP/IP 包跟踪和 MQ CHIN 跟踪](#)。
2. 收集数据。
 - a) 记录软件的版本，发行版和维护级别：
 - IBM MQ: 在 MSTR 作业日志中的 CSQY000I 消息中查找版本。
 - z/OS 操作系统: 在 SDSF 中的 /D IPLINFO 输出中查找版本。
 - 与该问题相关的任何其他产品: 请在作业记录中查找该产品的版本。
 - b) 收集 IBM MQ MSTR 和 CHIN 作业日志，并 (可选) 收集系统日志。
 - c) 收集收集跟踪时生成的 z/OS 转储。
 - d) 收集 z/OS LOGREC 报告。

有关更多信息，请参阅第 215 页的『[z/OS 上的 SYS1.LOGREC 信息](#)』。
3. 将收集的信息发送到 IBM。

问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息](#)。

要打开或更新案例，请转至 [IBM My Support](#) 站点。

注: 始终更新案例以指示已发送数据。

如果需要与 IBM 软件支持人员交谈，请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈，可以致电 1-800-IBM-SERV。

相关概念

第 43 页的『[对分布式队列管理问题进行故障诊断](#)』
故障诊断信息可帮助您解决与分布式队列管理 (DQM) 相关的问题。

将故障诊断信息发送到 IBM

生成并收集问题的故障诊断信息后，可以将其发送到 IBM 以帮助确定支持案例的问题。

关于此任务

发送故障诊断信息时，问题和数据的良好描述是您可以向 IBM 提供的最重要信息。请勿在未提供描述的情况下发送数据!

过程

- 有关 FTP 和电子邮件指示信息，请参阅 [与 IBM 软件支持人员交换信息以确定问题](#)。
- 转至 [IBM 我的支持站点](#) 以打开或更新案例。

注: 请始终更新案例以指示已发送数据。

有关 IBM 支持的更多信息, 包括如何注册获取支持, 请参阅 [IBM Support Guide](#)。

- 如果需要与 IBM 软件支持人员交谈, 请联系您的 [国家或地区代表](#)。如果需要在美国与 IBM 软件支持人员交谈, 可以致电 1-800-IBM-SERV。

相关任务

第 228 页的『[使用 runmqras 自动收集故障诊断信息](#)』

如果需要将 IBM MQ 故障诊断信息发送给 IBM 支持人员, 那么可以使用 **runmqras** 命令将信息一起收集到单个归档中。

第 231 页的『[手动收集故障诊断信息](#)』

在某些情况下, 您可能需要手动收集故障诊断信息, 例如, 如果您正在运行较旧版本的 IBM MQ, 或者无法使用 **runmqras** 命令自动收集故障诊断信息。

使用错误日志

有各种错误日志可用于帮助进行问题确定和故障诊断。

Multi 在 Multiplatforms 上, 使用以下链接了解可用于平台的错误日志及其使用方法:

- **ULW** [第 286 页的『UNIX, Linux, and Windows 上的错误日志』](#)
- **IBM i** [第 289 页的『IBM i 上的错误日志』](#)

z/OS 在 z/OS 上, 错误消息写入到:

- z/OS 系统控制台
- 通道启动程序作业日志

有关 IBM MQ for z/OS 上的错误消息、控制台日志和转储的信息, 请参阅 [z/OS 上的问题确定](#)。

禁止或排除错误日志中的消息

可以禁止或排除 Multiplatforms 和 z/OS 系统上的某些消息:

- **Multi** 有关在 [多平台](#) 上禁止某些消息的信息, 请参阅第 292 页的『[在 Multiplatforms 上禁止错误日志中的通道错误消息](#)』。
- **z/OS** 在 z/OS 上, 如果您是使用 z/OS 消息处理设施来禁止消息, 那么可以禁止控制台消息。有关更多信息, 请参阅 [IBM MQ for z/OS 概念](#)。

AMQ_DIAGNOSTIC_MSG_SEVERITY 环境变量

V 9.1.0 **Multi**

如果为 IBM MQ 进程设置了环境变量 **AMQ_DIAGNOSTIC_MSG_SEVERITY**, 那么当该 IBM MQ 进程将消息写入错误日志或控制台时, 会将消息严重性作为单个大写字母字符追加到消息号中, 如下所示:

| 消息类型 | 字符 |
|--------------|----|
| 参考 (0) | I |
| 警告 (10) | W |
| 错误 (20 或 30) | E |
| 严重 (40) | S |
| 终止 (50) | T |

例如：

```
AMQ5051I: The queue manager task 'LOGGER-IO' has started.
AMQ7075W: Unknown attribute foo at /var/mqm/qmgrs/QM1/qm.ini in
the configuration data.
AMQ9510E: Messages cannot be retrieved from a queue.
AMQ8506S: Command server MQGET failed with reason code 2009.
AMQ8301T: IBM MQ storage monitor job could not be started.
```

注意：

1. 由于队列管理器将写入消息，因此必须在启动队列管理器的环境中设置此环境变量。这在 Windows 上尤其重要（它在其中可能是启动队列管理器的 Windows 服务）。
2. **AMQ_DIAGNOSTIC_MSG_SEVERITY** 还将影响程序显示的消息。

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

请注意，新服务始终添加严重性字符。

ISO 8601 时间

V 9.1.0 Multi

从 IBM MQ 9.1 起，消息时间采用 ISO 8601 格式（而非本地时间）包含在内。

当 IBM MQ 进程将消息写入错误日志时，ISO 8601 格式的消息时间（全球标准时间 (UTC)）将作为 Time() 属性包含在内。

例如，其中 Z 时区指示 UTC：

```
11/04/2017 07:37:59 - Process(1) User(X) Program(amqzmuc0.exe)
Host(JOHNDOE) Installation(MQNI09000200)
VRMF(9.0.2.0) QMgr(QM1)
Time(2017-04-11T07:37:59.976Z)
```

回滚时重命名

V 9.1.0 Multi

在 IBM MQ 9.1 之前，当 AMQERR01.LOG 达到配置的最大大小时，AMQERR02.LOG 将重命名为 AMQERR03.LOG。然后，将 AMQERR01.LOG 的内容复制到 AMQERR02.LOG，并且 AMQERR01.LOG 截断为空。这意味着，对于某些工具而言，在将消息复制到 AMQERR02.LOG 之前，可能丢失工具尚未处理的消息。

从 IBM MQ 9.1 开始，更改了此逻辑，从而将 AMQERR01.LOG 重命名为 AMQERR02.LOG。

相关概念

第 293 页的『[First Failure Support Technology \(FFST\)](#)』

First Failure Support Technology (FFST) for IBM MQ 可提供有关事件的信息，在发生错误的情况下，这些信息有助于 IBM 支持人员对问题进行诊断。

相关任务

第 5 页的『[IBM MQ 故障诊断和支持](#)』

如果您的队列管理器网络或 IBM MQ 应用程序存在问题，那么可以使用本信息中描述的方法来帮助您诊断和解决问题。如果需要有关问题的帮助，可以通过 IBM 支持站点联系 IBM 支持人员。

第 304 页的『[使用跟踪](#)』

您可以使用不同类型的跟踪来帮助您确定问题并进行故障诊断。

ULW UNIX, Linux, and Windows 上的错误日志

安装 IBM MQ 时创建的 errors 子目录最多可以包含三个错误日志文件。

在安装时，将在 UNIX and Linux 系统下的 /var/mqm 文件路径和安装目录 (例如 Windows 系统下的 C:\Program Files\IBM\MQ\ 文件路径) 中创建 errors 子目录。errors 子目录最多可以包含三个错误日志文件，其名称为：

- AMQERR01.LOG
- AMQERR02.LOG
- AMQERR03.LOG

有关存储日志文件的目录的更多信息，请参阅第 288 页的『UNIX, Linux, and Windows 上的错误日志目录』。

创建队列管理器之后，它会在需要时创建三个错误日志文件。这些文件具有与系统错误日志目录中的错误日志文件相同的名称。即 AMQERR01、AMQERR02 和 AMQERR03，并且它们的缺省容量都为 **V9.1.0** 32 MB (33554432 字节)。可以在 Extended 队列管理器属性页面中从 IBM MQ Explorer 更改容量，也可以在 qm.ini 文件的 QMErrorLog 节中更改容量。这些文件放置在您安装 IBM MQ 或创建队列管理器时选择的队列管理器数据目录中的 errors 子目录中。errors 子目录的缺省位置是 UNIX and Linux 系统下的 /var/mqm/qmgrs/ qmname 文件路径和 Windows 系统下的 C:\Program Files\IBM\MQ\qmgrs\ qmname \errors 文件路径。

V9.1.0 系统会生成错误消息，这些消息放在 AMQERR01 中。在 AMQERR01 超过 32 MB 时，其会重命名为 AMQERR02。

因此，最新错误消息始终放在 AMQERR01 中，而其他文件用于维护错误消息的历史记录。

除非队列管理器不可用或者其名称未知，否则与通道相关的所有消息也放在属于队列管理器的相应错误文件中。在任一情况下，与通道相关的消息放在系统错误日志目录中。

要检查任何错误日志文件的内容，请使用常用的系统编辑器。

错误日志示例

第 287 页的图 53 显示 IBM MQ 错误日志中的摘录：

```
17/11/2014 10:32:29 - Process(2132.1) User(USER_1) Program(runmqchi.exe)
Host(HOST_1) Installation(Installation1)
VRMF(8.0.0.0) QMgr (A.B.C)
AMQ9542: Queue manager is ending.

EXPLANATION:
The program will end because the queue manager is quiescing.
ACTION:
None.
----- amqrimna.c : 931 -----
```

图 53: 样本 IBM MQ 错误日志

操作员消息

操作员消息识别一般错误，通常由用户在命令上使用无效的参数等类似操作时直接引起。操作员消息支持本地语言，其中消息目录安装在标准位置。

这些消息写入到关联窗口 (如果有)。此外，某些操作员消息写入到队列管理器目录中的 AMQERR01.LOG 文件，而其他操作员消息写入到系统错误日志目录中的等效文件。

错误日志访问限制

某些错误日志目录和错误日志具有访问限制。

要获取以下访问许可权，用户或应用程序必须是 mqm 组的成员：

- 对所有队列管理器错误日志目录的读写访问权。
- 对所有队列管理器错误日志的读写访问权。

- 对系统错误日志的写访问权。

如果未经授权的用户或应用程序尝试向队列管理器错误日志目录中写入消息，那么该消息重定向到该系统错误日志目录。

忽略 UNIX and Linux 系统下的错误代码

在 UNIX and Linux 系统上，如果不希望将某些错误日志写入到队列管理器错误日志，那么可以使用 QMErrorLog 节指定要忽略的错误代码。

有关更多信息，请参阅[队列管理器错误日志](#)。

忽略 Windows 系统下的错误代码

在 Windows 系统上，错误消息写入到 IBM MQ 错误日志和 Windows 应用程序事件日志。写入到应用程序事件日志的错误消息包括错误严重性、警告严重性和信息严重性的消息。如果不希望将某些错误消息写入到 Windows 应用程序事件日志，那么可以在 Windows 注册表中指定要忽略的错误代码。

使用以下注册表键：

```
HKLM\Software\IBM\WebSphere MQ\Installation\MQ_INSTALLATION_NAME\IgnoredErrorCodes
```

其中 `MQ_INSTALLATION_NAME` 是与 IBM MQ 的特定安装关联的安装名称。

该键设置为的值是以空字符分隔的字符串数组，其中每个字符串值与要从错误日志中忽略的错误代码相关。完整列表以空字符结尾，其类型为 REG_MULTI_SZ。

例如，如果希望 IBM MQ 从 Windows 应用程序事件日志中排除错误代码 AMQ3045、AMQ6055 和 AMQ8079，请将值设置为：

```
AMQ3045\0AMQ6055\0AMQ8079\0\0
```

对于机器上的所有队列消息都将定义要排除的消息的列表。在重新启动每个队列管理器之前，对配置进行的任何更改都不会生效。

相关概念

[第 285 页的『使用错误日志』](#)

有各种错误日志可用于帮助进行问题确定和故障诊断。

[第 179 页的『对 IBM MQ for z/OS 问题进行故障诊断』](#)

IBM MQ for z/OS、CICS、Db2 和 IMS 产生可用于问题确定的诊断信息。

相关任务

[第 5 页的『IBM MQ 故障诊断和支持』](#)

如果您的队列管理器网络或 IBM MQ 应用程序存在问题，那么可以使用本信息中描述的方法来帮助您诊断和解决问题。如果需要有关问题的帮助，可以通过 IBM 支持站点联系 IBM 支持人员。

[第 304 页的『使用跟踪』](#)

您可以使用不同类型的跟踪来帮助您确定问题并进行故障诊断。

相关参考

[第 289 页的『IBM i 上的错误日志』](#)

使用此信息来了解 IBM MQ for IBM i 错误日志。

UNIX, Linux, and Windows 上的错误日志目录

IBM MQ 使用大量错误日志来捕获与 IBM MQ 自己的操作、您启动的任何队列管理器有关的消息以及来自使用中的通道的错误数据。错误日志的位置取决于队列管理器名称是否已知，以及错误是否与客户机关联。

错误日志的存储位置取决于队列管理器名称是否已知，以及错误是否与客户机关联。

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

- 如果队列管理器名称已知，那么错误日志的位置如[第 289 页的表 23](#)中所示。

| 平台 | 目录 |
|--------------------------|--|
| Linux 和 Linux 系统 UNIX | /var/mqm/qmgrs/ <i>qmname</i> /errors |
| Windows 系统 | MQ_INSTALLATION_PATH\QMGRS\ <i>qmname</i> \ERRORS\AMQERR01.LOG |

- 如果队列管理器名称未知，那么错误日志的位置如第 289 页的表 24 中所示。

| 平台 | 目录 |
|--------------------------|--|
| Linux 和 Linux 系统 UNIX | /var/mqm/errors |
| Windows 系统 | MQ_INSTALLATION_PATH\QMGRS\@SYSTEM\ERRORS\AMQERR01.LOG |

- 如果客户机应用程序已发生错误，那么客户机上的错误日志的位置如第 289 页的表 25 中所示。

| 平台 | 目录 |
|--------------------------|----------------------------------|
| Linux 和 Linux 系统 UNIX | /var/mqm/errors |
| Windows 系统 | MQ_DATA_PATH\ERRORS\AMQERR01.LOG |

Windows 在 IBM MQ for Windows 中，错误指示也添加到应用程序日志，可以使用 Windows 系统随附的事件查看器应用程序对其进行检查。

早期错误

存在许多尚未建立这些错误日志即发生错误的特殊情况。IBM MQ 尝试在错误日志中记录任何此类错误。日志的位置取决于已建立的队列管理器的量。

例如，如果由于配置文件损坏，无法确定位置信息，那么会将错误记录到安装时在根目录 (/var/mqm 或 C:\Program Files\IBM\MQ) 上创建的错误目录中。

如果 IBM MQ 可以读取其配置信息，并且可以访问 DefaultPrefix 的值，那么会将错误记录在 DefaultPrefix 属性标识的目录的 errors 子目录中。例如，如果缺省前缀为 C:\Program Files\IBM\MQ，那么会在 C:\Program Files\IBM\MQ\errors 中记录错误。

有关配置文件的进一步信息，请参阅[更改 IBM MQ 和队列管理器配置信息](#)。

注: 启动队列管理器后，将通过消息来通知 Windows 注册表中的错误。

IBM i 上的错误日志

使用此信息来了解 IBM MQ for IBM i 错误日志。

缺省情况下，只有 QMQADM 组成员才能访问错误日志。要使非该组成员的用户有权访问错误日志，请将 **ValidateAuth** 设置为 *No*，并授予这些用户 *PUBLIC 权限。请参阅[文件系统](#)以获取更多信息。

IBM MQ 使用大量错误日志来捕获与 IBM MQ 本身的操作、您启动的任何队列管理器以及来自使用中通道的错误数据有关的消息。

在安装时，将在 IFS 中创建 /QIBM/UserData/mqm/errors 子目录。

IFS 中错误日志的位置取决于队列管理器名称是否已知：

- 如果队列管理器名称已知且队列管理器可用，那么错误日志位于以下目录中：

```
/QIBM/UserData/mqm/qmgrs/qmname/errors
```

- 如果队列管理器不可用，那么错误日志位于以下目录中：

```
/QIBM/UserData/mqm/errors
```

可以使用系统实用程序 EDTF 来浏览错误目录和文件。例如：

```
EDTF '/QIBM/UserData/mqm/errors'
```

或者，可以从 WRKMQM 面板对队列管理器使用选项 23。

errors 子目录最多可以包含三个错误日志文件，其名称为：

- AMQERR01.LOG
- AMQERR02.LOG
- AMQERR03.LOG

创建队列管理器之后，在该队列管理器需要时将创建三个错误日志文件。这些文件的名称与 /QIBM/UserData/mqm/errors 文件相同，即 AMQERR01、AMQERR02 和 AMQERR03，每个文件的容量为 2 MB（2 097 152 字节）。这些文件放置在您创建的每个队列管理器的 errors 子目录中，即 /QIBM/UserData/mqm/qmgrs/qmname/errors。

系统会生成错误消息，这些消息放在 AMQERR01 中。当 AMQERR01 大小超过 2 MB（2 097 152 字节）时，它将复制到 AMQERR02。在复制之前，AMQERR02 复制到 AMQERR03.LOG。系统将废弃 AMQERR03 的先前内容（如果有）。

因此，最新错误消息始终放在 AMQERR01 中，而其他文件用于维护错误消息的历史记录。

除非队列管理器不可用或者其名称未知，否则与通道相关的所有消息也放在队列管理器的相应错误文件中。当队列管理器名称不可用或无法确定其名称时，将与通道相关的消息放置在 /QIBM/UserData/mqm/errors 子目录中。

要检查任何错误日志文件的内容，请使用系统编辑器 EDTF 来查看 IFS 中的流文件。

注：

1. 请勿更改这些错误日志的所有权。
2. 如果删除了任何错误日志文件，那么在记录下一条错误消息时会自动重新创建该错误日志文件。

早期错误

存在许多尚未建立错误日志即发生错误的特殊情况。IBM MQ 尝试在错误日志中记录任何此类错误。日志的位置取决于已建立的队列管理器的量。

例如，如果由于配置文件损坏，无法确定任何位置信息，那么会将错误记录到在安装时创建的 errors 目录中。

如果 IBM MQ 配置文件和 AllQueueManagers 节的 DefaultPrefix 属性均可读，那么会将错误记录在 DefaultPrefix 属性标识的目录的 errors 子目录中。

操作员消息

操作员消息识别一般错误，通常由用户在命令上使用无效的参数等类似操作时直接引起。操作员消息支持本地语言，其中消息目录安装在标准位置。

这些消息写入到作业日志（如果有）。此外，一些操作员消息会写入队列管理器目录中的 AMQERR01.LOG 文件，而其他消息则写入错误日志的 /QIBM/UserData/mqm/errors 目录副本。

IBM MQ 错误日志示例

第 291 页的图 54 显示来自 IBM MQ 错误日志的典型摘录。

```
*****Beginning of data*****
07/19/02 11:15:56 AMQ9411: Repository manager ended normally.

EXPLANATION:
Cause . . . . . : The repository manager ended normally.
Recovery . . . . : None.
Technical Description . . . . . : None.
-----
07/19/02 11:15:57 AMQ9542: Queue manager is ending.

EXPLANATION:
Cause . . . . . : The program will end because the queue manager is quiescing.
Recovery . . . . : None.
Technical Description . . . . . : None.
----- amqrimna.c : 773 -----
07/19/02 11:16:00 AMQ8004: IBM MQ queue manager 'mick' ended.
EXPLANATION:
Cause . . . . . : IBM MQ queue manager 'mick' ended.
Recovery . . . . : None.
Technical Description . . . . . : None.
-----
07/19/02 11:16:48 AMQ7163: IBM MQ job number 18429 started.

EXPLANATION:
Cause . . . . . : This job has started to perform work for Queue Manager
                    mick, The job's PID is 18429 the CCSID is 37. The job name is
                    582775/MQUSER/AMQZXMA0.
Recovery . . . . : None
-----
07/19/02 11:16:49 AMQ7163: IBM MQ job number 18430 started.

EXPLANATION:
Cause . . . . . : This job has started to perform work for Queue Manager
                    mick, The job's PID is 18430 the CCSID is 0. The job name is
                    582776/MQUSER/AMQZFUMA.
Recovery . . . . : None
-----
07/19/02 11:16:49 AMQ7163: IBM MQ job number 18431 started.

EXPLANATION:
Cause . . . . . : This job has started to perform work for Queue Manager
                    mick, The job's PID is 18431 the CCSID is 37. The job name is
                    582777/MQUSER/AMQZXMAX.
Recovery . . . . : None
-----
07/19/02 11:16:50 AMQ7163: IBM MQ job number 18432 started.

EXPLANATION:
Cause . . . . . : This job has started to perform work for Queue Manager
                    mick, The job's PID is 18432 the CCSID is 37. The job name is
                    582778/MQUSER/AMQALMPX.
Recovery . . . . : None
-----
```

图 54: 来自 IBM MQ 错误日志的摘录

相关概念

第 286 页的『[UNIX, Linux, and Windows 上的错误日志](#)』
安装 IBM MQ 时创建的 errors 子目录最多可以包含三个错误日志文件。

第 285 页的『[使用错误日志](#)』
有各种错误日志可用于帮助进行问题确定和故障诊断。

第 179 页的『[对 IBM MQ for z/OS 问题进行故障诊断](#)』
IBM MQ for z/OS、CICS、Db2 和 IMS 产生可用于问题确定的诊断信息。

相关任务

第 304 页的『使用跟踪』

您可以使用不同类型的跟踪来帮助确定问题并进行故障诊断。

z/OS 上的错误日志

错误消息写入到：

- z/OS 系统控制台
- 通道启动程序作业日志

如果您使用 z/OS 消息处理工具来禁止消息，那么可能会禁止控制台消息。请参阅 [在 z/OS 上规划 IBM MQ 环境](#)。

记录 IBM MQ classes for JMS 中的错误

有关可能需要用户采取纠正行动的运行时问题的信息将写入 IBM MQ classes for JMS 日志。

例如，如果应用程序尝试设置连接工厂的属性，但属性的名称无法识别，IBM MQ classes for JMS 会将关于此问题的信息写入其日志。

缺省情况下，包含日志的文件名为 mqjms.log，该文件位于当前工作目录中。但是，您可以通过设置 IBM MQ classes for JMS 配置文件中的 `com.ibm.msg.client.commonservices.log.outputName` 属性来更改日志文件的名称和位置。有关 IBM MQ classes for JMS 配置文件的信息，请参阅 [IBM MQ classes for JMS 配置文件](#)，有关 `com.ibm.msg.client.commonservices.log.outputName` 属性的有效值的更多详细信息，请参阅第 57 页的『IBM MQ classes for JMS 的日志记录错误』。

Multi 在 Multiplatforms 上禁止错误日志中的通道错误消息

您可以在指定的时间间隔内阻止将选定的消息发送到错误日志，例如，在 IBM MQ 系统生成大量参考消息来填充错误日志的情况下。

关于此任务

对于给定的时间间隔，有两种禁止消息的方法：

- 通过在 `qm.ini` 文件的 `QMErrorLog` 节中使用 `SuppressMessage` 和 `SuppressInterval`。此方法使您能够禁止 [诊断消息服务节](#) 中列示的错误消息。
- 通过使用环境变量 `MQ_CHANNEL_SUPPRESS_MSGS` 和 `MQ_CHANNEL_SUPPRESS_INTERVAL`。此方法使您能够禁止任何通道消息。

过程

- 要通过在 `qm.ini` 文件中使用 `QMErrorLog` 节在给定时间间隔内禁止消息，请使用 `SuppressMessage` 指定仅在给定时间间隔内写入队列管理器错误日志一次的消息，并使用 `SuppressInterval` 指定要禁止消息的时间间隔。
例如，要将 AMQ9999、AMQ9002、AMQ9209 消息禁止 30 秒，请在 `qm.ini` 文件的 `QMErrorLog` 节中包含以下信息：

```
SuppressMessage=9001,9002,9202  
SuppressInterval=30
```

Windows **Linux** 或者，您可以使用 IBM MQ Explorer 中的已扩展队列管理器属性页面来排除和禁止消息，而不是直接编辑 `qm.ini` 文件。

- 要使用环境变量 `MQ_CHANNEL_SUPPRESS_MSGS` 和 `MQ_CHANNEL_SUPPRESS_INTERVAL` 在给定时间间隔内禁止消息，请完成以下步骤：
 - a) 使用 `MQ_CHANNEL_SUPPRESS_MSGS` 指定要禁止的消息。

最多可将 20 个通道错误消息代码包含在逗号分隔列表中。没有可以包含在 **MQ_CHANNEL_SUPPRESS_MSGS** 环境变量中的消息标识的限制性列表。然而，消息标识必须是通道消息（即 AMQ9xxx: messages）。

以下示例适用于消息 AMQ9999、AMQ9002 和 AMQ9209。

–   在 UNIX 和 Linux 上:

```
export MQ_CHANNEL_SUPPRESS_MSGS=9999,9002,9209
```

–  在 Windows 上:

```
set MQ_CHANNEL_SUPPRESS_MSGS=9999,9002,9209
```

b) 使用 **MQ_CHANNEL_SUPPRESS_INTERVAL** 指定要禁止消息的时间间隔。

缺省值为 60,5，这意味着在 60 秒时间间隔内给定消息的前五次出现之后，该消息的任何进一步出现都将被禁止，直到该 60 秒时间间隔结束。值 0,0 意味着始终禁止。值 0,n（其中 n > 0）意味着永不禁止。

相关概念

[UNIX 上的 QMErrorLog 节](#)

[IBM i 上的 QMErrorLog 节](#)

[队列管理器属性](#)

相关参考

[环境变量描述](#)

First Failure Support Technology (FFST)

First Failure Support Technology (FFST) for IBM MQ 可提供有关事件的信息，在发生错误的情况下，这些信息有助于 IBM 支持人员对问题进行诊断。

发生内部事件时，首次故障数据捕获 (FFDC) 提供系统环境的自动快照。发生错误时，IBM 支持人员可使用此快照更好地理解发生问题时系统和 IBM MQ 状态。


在 FFST 文件中包含有关事件的信息。在 IBM MQ 中，FFST 文件的文件类型为 FDC。FFST 文件并不始终表示存在错误。FFST 可能只是参考信息。

监控和内务处理

以下是有助于您管理 FFST 事件的一些提示：

- 为系统监控 FFST 事件，并确保发生事件时采取相应且及时的补救操作。在某些情况下，可能收到期望的 FDC 文件，因此可忽略这些文件，例如，当用户终止 IBM MQ 进程时出现的 FFST 事件。通过相应的监控，可以确定哪些事件是期望的事件，以及哪些事件不是期望的事件。
- 针对 IBM MQ 外的事件也会生成 FFST 事件。例如，如果 IO 子系统或网络出现问题，可能在 FDC 类型文件中报告此问题。这些类型的事件在 IBM MQ 可控范围之外，您可能需要与第三方接洽以调查根本原因。
- 确保执行良好的 FFST 文件内务处理。文件必须归档，目录或文件夹必须清除以确保在支持团队需要 FDC 文件时，仅提供最近且相关的 FDC 文件。

使用以下链接中的信息来查找不同平台的 FFST 文件的名称、位置和内容。

- [第 294 页的『FFST: IBM MQ classes for JMS』](#)
- [第 298 页的『FFST: IBM MQ for Windows』](#)
- [第 301 页的『FFST: IBM MQ for UNIX 和 Linux 系统』](#)
-  [第 302 页的『FFST: IBM MQ for IBM i』](#)
-

相关概念

[第 285 页的『使用错误日志』](#)

有各种错误日志可用于帮助进行问题确定和故障诊断。

[第 179 页的『对 IBM MQ for z/OS 问题进行故障诊断』](#)

IBM MQ for z/OS、CICS、Db2 和 IMS 产生可用于问题确定的诊断信息。

相关任务

[第 5 页的『IBM MQ 故障诊断和支持』](#)

如果您的队列管理器网络或 IBM MQ 应用程序存在问题，那么可以使用本信息中描述的方法来帮助您诊断和解决问题。如果需要有关问题的帮助，可以通过 IBM 支持站点联系 IBM 支持人员。

[第 304 页的『使用跟踪』](#)

您可以使用不同类型的跟踪来帮助您确定问题并进行故障诊断。

[第 227 页的『联系 IBM 支持人员』](#)

如果您在 IBM MQ 方面遇到问题需要帮助，可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订、故障诊断和其他新闻的通知。

FFST: IBM MQ classes for JMS

描述 First Failure Support Technology (FFST) 的名称，位置和内容 IBM MQ classes for JMS 生成的文件。

使用 IBM MQ classes for JMS 时，FFST 信息将记录在名为 FFDC 的目录下的文件中，缺省情况下，该目录是生成 FFST 时运行的 IBM MQ classes for JMS 应用程序当前工作目录的子目录。如果已在 IBM MQ classes for JMS 配置文件中设置 `com.ibm.msg.client.commonservices.trace.outputName` 属性，那么 FFDC 目录为该属性所指向的目录的子目录。有关 IBM MQ classes for JMS 的信息，请参阅 [IBM MQ classes for JMS 配置文件](#)。

一份 FFST 文件包含一条 FFST 记录。每一条 FFST 记录均包含有关通常很严重或可能无法恢复的错误的信息。这些记录通常会指出系统配置问题或 IBM MQ classes for JMS 内部错误。

FFST 文件名为 `JMSC nnnn.FDC`，其中 `nnnn` 从 1 开始。如果已存在完整的文件名，那么此值会递增 1，直到找到唯一的 FFST 文件名。

IBM MQ classes for JMS 应用程序实例会将 FFST 信息写入多个 FFST 文件。如果在一次执行应用程序的过程中出现多个错误，那么每一条 FFST 记录都会写入不同的 FFST 文件。

FFST 记录部分

IBM MQ classes for JMS 生成的 FFST 记录包含以下部分：

头

头，指示创建 FFST 记录的时间、运行 IBM MQ classes for JMS 应用程序的平台以及正在调用的内部方法。该头还包含探测器标识，该探测器标识唯一地标识生成 FFST 记录的 IBM MQ classes for JMS 中的位置。

数据

与 FFST 记录关联的一些内部数据。

版本信息

有关生成 FFST 记录的应用程序正在使用的 IBM MQ classes for JMS 版本的信息。

堆栈跟踪

生成 FFST 记录的线程的 Java 堆栈跟踪。

属性库内容

已在运行 IBM MQ classes for JMS 应用程序的 Java 运行时环境上设置的所有 Java 系统属性的列表。

WorkQueueMananger 内容

有关 IBM MQ classes for JMS 所使用的内部线程池的信息。

运行时属性

有关运行 IBM MQ classes for JMS 应用程序的系统上可用的内存量和处理器数的详细信息。

组件管理器内容

有关 IBM MQ classes for JMS 装入的内部组件的一些信息。

特定于提供程序的信息

有关生成 FFST 时运行的 IBM MQ classes for JMS 应用程序当前正在使用的所有活动 JMS 连接， JMS 会话， MessageProducer 和 MessageConsumer 对象的信息。 此信息包含 JMS Connections 和 JMS Sessions 连接到的队列管理器的名称， 以及 MessageProducers 和 MessageConsumers 所使用的 IBM MQ 队列或主题对象的名称。

所有线程信息

有关生成 FFST 记录时 IBM MQ classes for JMS 应用程序正在其中运行的 Java 运行时环境中所有活动线程的状态的详细信息。 将显示每个线程的名称， 以及每个线程的 Java 堆栈跟踪。

FFST 日志文件示例

```
-----START FFST-----
c:\JBoss-6.0.0\bin\FFDC\JMSSC0007.FDC PID:4472

JMS Common Client First Failure Symptom Report

Product      :- IBM MQ classes for JMS
Date/Time    :- Mon Feb 03 14:14:46 GMT 2014
System time  :- 1391436886081
Operating System :- Windows Server 2008
UserID       :- pault
Java Vendor  :- IBM Corporation
Java Version  :- 2.6

Source Class  :- com.ibm.msg.client.commonservices.j2se.wmqsupport.PropertyStoreImpl
Source Method :- getBooleanProperty(String)
ProbeID      :- XS002005
Thread       :- name=pool-1-thread-3 priority=5 group=workmanager-threads
cc1=BaseClassLoader@ef1c3794{vfs:///C:/JBoss-6.0.0/server/default/deploy/basicMDB.ear}

Data
----

| name :- com.ibm.mq.connector.performJavaEEContainerChecks

Version information
-----

Java Message Service Client
7.5.0.2
p750-002-130627
Production

IBM MQ classes for Java Message Service
7.5.0.2
p750-002-130627
Production

IBM MQ JMS Provider
7.5.0.2
p750-002-130627
Production

Common Services for Java Platform, Standard Edition
7.5.0.2
p750-002-130627
Production

Stack trace
-----

Stack trace to show the location of the FFST call
| FFST Location :- java.lang.Exception
|   at com.ibm.msg.client.commonservices.trace.Trace.getCurrentPosition(Trace.java:1972)
|   at com.ibm.msg.client.commonservices.trace.Trace.createFFSTString(Trace.java:1911)
|   at com.ibm.msg.client.commonservices.trace.Trace.ffstInternal(Trace.java:1800)
|   at com.ibm.msg.client.commonservices.trace.Trace.ffst(Trace.java:1624)
|   at
|   com.ibm.msg.client.commonservices.j2se.propertystore.PropertyStoreImpl.getBooleanProperty(
PropertyStoreImpl.java:322)
|   at
|   com.ibm.msg.client.commonservices.propertystore.PropertyStore.getBooleanPropertyObject(Pr
opertyStore.java:302)
```

```

|      at
com.ibm.mq.connector.outbound.ConnectionWrapper.jcaMethodAllowed(ConnectionWrapper.java:510)
|      at
com.ibm.mq.connector.outbound.ConnectionWrapper.setExceptionListener(ConnectionWrapper.java:244)
|      at com.ibm.basicMDB.MDB.onMessage(MDB.java:45)
...

```

Property Store Contents

All currently set properties

```

|  awt.toolkit                :- sun.awt.windows.WToolkit
|  catalina.ext.dirs         :- C:\JBoss-6.0.0\server\default\lib
|  catalina.home             :- C:\JBoss-6.0.0\server\default
|  com.ibm.cpu.endian        :- little
|  com.ibm.jcl.checkClassPath :-
|  com.ibm.mq.connector.performJavaEEContainerChecks :- false
|  com.ibm.oti.configuration :- scar
|  com.ibm.oti.jcl.build     :- 20131013_170512
|  com.ibm.oti.shared.enabled :- false
|  com.ibm.oti.vm.bootstrap.library.path :- C:\Program
Files\IBM\Java70\jre\bin\compressedrefs;C:\Program Files\IBM\Java70\jre\bin
|  com.ibm.oti.vm.library.version :- 26
|  com.ibm.system.agent.path :- C:\Program
Files\IBM\Java70\jre\bin
|  com.ibm.util.extralibs.properties :-
|  com.ibm.vm.bitmode        :- 64
|  com.ibm.zero.version      :- 2
|  console.encoding         :- Cp850
|  file.encoding            :- Cp1252
|  file.encoding.pkg        :- sun.io
...

```

WorkQueueMananger Contents

```

|  Current ThreadPool size   :- 2
|  Maintain ThreadPool size :- false
|  Maximum ThreadPool size  :- -1
|  ThreadPool inactive timeout :- 0

```

Runtime properties

```

|  Available processors      :- 4
|  Free memory in bytes (now) :- 54674936
|  Max memory in bytes     :- 536870912
|  Total memory in bytes (now) :- 235012096

```

Component Manager Contents

Common Services Components:

```

|  CMVC                :- p750-002-130627
|  Class Name          :- class com.ibm.msg.client.commonservices.j2se.J2SEComponent
|  Component Name      :- com.ibm.msg.client.commonservices.j2se
|  Component Title     :- Common Services for Java Platform, Standard Edition
|  Factory Class       :- class com.ibm.msg.client.commonservices.j2se.CommonServicesImplementation
|  Version             :- 7.5.0.2
|  inPreferenceTo[0]  :- com.ibm.msg.client.commonservices.j2me

```

Messaging Provider Components:

```

|  CMVC                :- p750-002-130627
|  Class Name          :- class com.ibm.msg.client.wmq.factories.WMQComponent
|  Component Name      :- com.ibm.msg.client.wmq
|  Component Title     :- IBM MQ JMS Provider
|  Factory Class       :- class com.ibm.msg.client.wmq.factories.WMQFactoryFactory
|  Version             :- 7.5.0.2

```

Provider Specific Information

Overview of JMS System

```

Num. Connections : 3
Num. Sessions    : 3
Num. Consumers    : 0
Num. Producers   : 0

```



```

:
com.ibm.msg.client.commonservices.workqueue.WorkQueueManager.runWorkQueueItem(WorkQueueMa
nager.java:303)
:
com.ibm.msg.client.commonservices.j2se.workqueue.WorkQueueManagerImplementation$ThreadPoo
lWorker.run(WorkQueueManagerImplementation.java:1219)
Name : RcvThread:
com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection@269522111[qmid=test_2014-01-
24_15.55.24,fap=10,channel=MY.SVRCONN,ccsid=850,sharecnv=10,hbint=300,peer=/9.20.124.119(
1414),localport=65243,ssl=no,hConns=0,LastDataSend=1391436871409 (0ms ago)
),LastDataRecv=1391436871409 (0ms ago),]
Priority : 5
ThreadGroup : java.lang.ThreadGroup[name=JMSSCThreadPool,maxpri=10]
ID : 84
State : RUNNABLE
Stack :
java.net.SocketInputStream.socketRead0(SocketInputStream.java:-2)
:
java.net.SocketInputStream.read(SocketInputStream.java:163)
:
java.net.SocketInputStream.read(SocketInputStream.java:133)
:
com.ibm.mq.jmqi.remote.impl.RemoteTCPConnection.receive(RemoteTCPConnection.java:1545)
:
com.ibm.mq.jmqi.remote.impl.RemoteRcvThread.receiveBuffer(RemoteRcvThread.java:794)
:
com.ibm.mq.jmqi.remote.impl.RemoteRcvThread.receiveOneTSH(RemoteRcvThread.java:757)
:
com.ibm.mq.jmqi.remote.impl.RemoteRcvThread.run(RemoteRcvThread.java:150)
:
com.ibm.msg.client.commonservices.workqueue.WorkQueueItem.runTask(WorkQueueItem.java:214)
:
com.ibm.msg.client.commonservices.workqueue.SimpleWorkQueueItem.runItem(SimpleWorkQueueIte
m.java:105)
:
com.ibm.msg.client.commonservices.workqueue.WorkQueueItem.run(WorkQueueItem.java:229)
:
com.ibm.msg.client.commonservices.workqueue.WorkQueueManager.runWorkQueueItem(WorkQueueMan
ager.java:303)
:
com.ibm.msg.client.commonservices.j2se.workqueue.WorkQueueManagerImplementation$ThreadPoo
lWorker.run(WorkQueueManagerImplementation.java:1219)
:
:
First Failure Symptom Report completed at Mon Feb 03 14:14:46 GMT 2014
-----END FFST-----

```

FFST 记录的 "头", "数据" 和 "堆栈跟踪" 部分中的信息由 IBM 用于帮助确定问题。在许多情况下, 在生成 FFST 记录时, 系统管理员都无需很多操作, 但通过 IBM 支持中心提出问题除外。

禁止 FFST 记录

IBM MQ classes for JMS 生成的 FFST 文件包含一条 FFST 记录。如果在 IBM MQ classes for JMS 应用程序执行期间多次出现问题, 那么将生成具有相同探测标识的多个 FFST 文件。这可能不是您想看到的。可使用 `com.ibm.msg.client.commonservices.ffst.suppress` 属性来禁止生成 FFST 文件。必须在应用程序使用的 IBM MQ classes for JMS 配置文件中设置该属性, 并可采用以下值:

- 0: 输出所有 FFDC 文件 (缺省值)。
- 1: 仅为探测标识输出第一份 FFST 文件。
- integer*: 针对探测标识禁止所有 FFST 文件, 数目为该数目倍数的文件除外。

Windows FFST: IBM MQ for Windows

描述 First Failure Support Technology (FFST) 的名称, 位置和内容 Windows 系统的文件。

在 IBM MQ for Windows 中, FFST 信息记录在 `C:\Program Files\IBM\MQ\errors` 目录下的文件中。

FFST 文件包含一条或多条记录。每一条 FFST 记录均包含有关通常很严重或可能无法恢复的错误的信息。这些记录通常指示系统存在配置问题或者存在 IBM MQ 内部错误。

FFST 文件名为 `AMQ nnnnn.mm.FDC`, 其中:

NNNNN

是报告错误的进程的标识

mm

从 0 开始。如果已存在完整的文件名，那么此值会递增 1，直到找到唯一的 FFST 文件名。如果复用进程，那么 FFST 文件名可能已存在。

进程实例将把所有 FFST 信息写入同一个 FFST 文件。如果在一次进程执行期间发生多个错误，那么 FFST 文件可能包含多个记录。

当进程写入一条 FFST 记录时，它还会向事件日志发送一条记录。此记录包含 FFST 文件的名称以帮助进行自动问题跟踪。事件日志条目是在应用程序级别产生的。

典型的 FFST 日志显示在 [第 300 页的图 55](#) 中。

```

+-----+
| WebSphere MQ First Failure Symptom Report
| =====
|
| Date/Time           :- Mon January 28 2008 21:59:06 GMT
| UTC Time/Zone      :- 1201539869.892015 0 GMT
| Host Name          :- 99VXY09 (Windows 7 Build 2600: Service Pack 1)
| PIDS               :- 5724H7200
| LVLS               :- 7.0.0.0
| Product Long Name  :- IBM MQ for Windows
| Vendor             :- IBM
| Probe Id           :- HL010004
| Application Name   :- MQM
| Component          :- hlgReserveLogSpace
| SCCS Info          :- lib/logger/amqhlge0.c, 1.26
| Line Number        :- 246
| Build Date         :- Jan 25 2008
| CMVC level         :- p000-L050202
| Build Type         :- IKAP - (Production)
| UserID             :- IBM User
| Process Name       :- C:\Program Files\IBM\MQ\bin\amqzlaa0.exe |
| Process            :- 00003456
| Thread             :- 00000030
| QueueManager      :- qmgr2
| ConnId(1) IPCC    :- 162
| ConnId(2) QM      :- 45
| Major Errorcode   :- hrcE_LOG_FULL
| Minor Errorcode   :- OK
| Probe Type        :- MSGAMQ6709
| Probe Severity    :- 2
| Probe Description  :- AMQ6709: The log for the Queue manager is full.
| FDCSequenceNumber :- 0
|
+-----+

MQM Function Stack
zlaMainThread
zlaProcessMessage
zlaProcessMQIRequest
zlaMOPUT
zsqMOPUT
kpiMOPUT
kqiPutIt
kqiPutMsgSegments
apiPutMessage
aqmPutMessage
aqhPutMessage
aqqWriteMsg
aqqWriteMsgData
aqlReservePutSpace
almReserveSpace
hlgReserveLogSpace
xcsFFST

MQM Trace History
-----} hlgReserveLogSpace rc=hrcW_LOG_GETTING_VERY_FULL
-----{ xllLongLockRequest
-----} xllLongLockRequest rc=OK

...

```

图 55: 样本 IBM MQ for Windows 首次故障症状报告

函数堆栈和跟踪历史记录是由 IBM 用来辅助进行问题确定的。如果生成了 FFST 记录，在很多时候，系统管理员除了将问题报告给 IBM 支持中心，就无能为力了。

在某些情况下，除了 FFST 文件之外，还会生成一个小的转储文件，它位于 C:\Program Files\IBM\MQ\errors 目录中。转储文件的名称与 FFST 文件的名称相同，格式为 AMQnnnnn.mm.dmp。IBM 可以使用这些文件来帮助进行问题确定。

First Failure Support Technology (FFST) 文件和 Windows 客户机

这些文件在产生时已格式化，它们位于 IBM MQ MQI client 安装目录的 errors 子目录中。

这些错误是正常情况下严重且不可恢复的错误，指示系统存在配置问题或者存在 IBM MQ 内部错误。

文件命名为 AMQnnnnn.mm.FDC，其中：

- nnnnn 是报告错误的进程标识
- mm 是一个序号，通常为 0

当进程创建一个 FFST 时，它还会向系统日志发送一条记录。此记录包含 FFST 文件的名称以帮助进行自动问题跟踪。

系统日志条目是在“user.error”级别产生的。

First Failure Support Technology 在 [First Failure Support Technology \(FFST\)](#) 中进行了详细说明。

Linux UNIX **FFST: IBM MQ for UNIX 和 Linux 系统**

描述 First Failure Support Technology (FFST) 的名称，位置和 content UNIX and Linux 系统的文件。

对于 IBM MQ on UNIX and Linux 系统，FFST 信息记录在 /var/mqm/errors 目录下的文件中。

FFST 文件包含一条或多条记录。每一条 FFST 记录均包含有关通常很严重或可能无法恢复的错误的信息。这些记录指示存在系统配置问题或发生 IBM MQ 内部错误。

FFST 文件名为 AMQ nnnnn.mm.FDC，其中：

NNNNN

是报告错误的进程的标识

mm

从 0 开始。如果已存在完整的文件名，那么此值会递增 1，直到找到唯一的 FFST 文件名。如果复用进程，那么 FFST 文件名可能已存在。

进程实例将把所有 FFST 信息写入同一个 FFST 文件。如果在一次进程执行期间发生多个错误，那么 FFST 文件可能包含多个记录。

要读取 FFST 文件内容，您必须是该文件的创建者或 mqm 组的成员。

当进程写入一条 FFST 记录时，它还会向系统日志发送一条记录。此记录包含 FFST 文件的名称以帮助进行自动问题跟踪。系统日志条目是在 user.error 级别生成的。请参阅有关 syslog.conf 的操作系统文档，以获取有关配置此项的信息。

函数堆栈和跟踪历史记录是由 IBM 用来辅助进行问题确定的。如果生成了 FFST 报告，在很多时候，系统管理员除了将问题报告给 IBM 支持中心，就无能为力了。

然而，有些问题是系统管理员可以解决的。如果 FFST 在调用其中一个 IPC 函数 (例如，se 拖把 或 shmget) 时显示 超出资源 或 超出设备 描述的空间，那么可能已超出相关内核参数限制。

如果 FFST 报告显示 setitimer 有问题，那么可能需要对内核计时器参数进行更改。

要解决这些问题，增加 IPC 限制、重新建立内核并重新启动机器。

First Failure Support Technology (FFST) 文件和 UNIX and Linux 客户机

FFST 日志是在发生严重的 IBM MQ 错误时写入的。它们被写入到 /var/mqm/errors 目录中。

这些错误是正常情况下严重且不可恢复的错误，指示系统存在配置问题或者存在 IBM MQ 内部错误。

文件命名为 AMQnnnnn.mm.FDC，其中：

- nnnnn 是报告错误的进程标识
- mm 是一个序号，通常为 0

当进程创建一个 FFST 时，它还会向系统日志发送一条记录。此记录包含 FFST 文件的名称以帮助进行自动问题跟踪。

系统日志条目是在“user.error”级别产生的。

First Failure Support Technology 在 [First Failure Support Technology \(FFST\)](#) 中进行了详细说明。

IBM i FFST: IBM MQ for IBM i

描述 First Failure Support Technology (FFST) 的名称, 位置和内容 IBM i 系统的文件。

对于 IBM i, 会将 FFST 信息记录在 /QIBM/UserData/mqm/errors 目录下的流文件中。

这些错误是正常情况下严重且不可恢复的错误, 指示系统存在配置问题或者存在 IBM MQ 内部错误。

流文件名为 AMQ nnnnn.mm.FDC, 其中:

- nnnnn 是报告错误的进程的标识。
- mm 是一个序号, 通常为 0。

失败的作业的日志副本将写入与 .FDC 文件同名的文件中。该文件名以 .JOB 结尾。

以下示例中显示了部分典型 FFST 数据。

```
-----  
| IBM MQ First Failure Symptom Report  
| =====  
|  
| Date/Time           :- Mon January 28 2008 21:59:06 GMT  
| UTC Time/Zone       :- 1201539869.892015 0 GMT  
| Host Name           :- WINAS12B.HURSLEY.IBM.COM  
| PIDS                :- 5733A38  
| LVLS                :- 520  
| Product Long Name   :- IBM MQ for IBMi  
| Vendor              :- IBM  
| Probe Id            :- XY353001  
| Application Name    :- MQM  
| Component           :- xehAS400ConditionHandler  
| Build Date          :- Feb 25 2008  
| UserID              :- 00000331 (MAYFCT)  
| Program Name        :- STRMQM_R MAYFCT  
| Job Name             :- 020100/MAYFCT/STRMQM_R  
| Activation Group    :- 101 (QMOM) (QMOM/STRMQM_R)  
| Process              :- 00001689  
| Thread              :- 00000001  
| QueueManager        :- TEST.AS400.OE.P  
| Major Errorcode     :- STOP  
| Minor Errorcode     :- OK  
| Probe Type          :- HALT6109  
| Probe Severity      :- 1  
| Probe Description   :- 0  
| Arith1              :- 1 1  
| Comment1            :- 00d0  
|-----
```

```
MQM Function Stack  
lpiSPIMQConnect  
zstMQConnect  
ziiMQCONN  
ziiClearUpAgent  
xcsTerminate  
xlsThreadInitialization  
xcsConnectSharedMem  
xstConnSetInSPbyHandle  
xstConnSharedMemSet  
xcsFFST
```

```
MQM Trace History  
<-- xcsCheckProcess rc=xecP_E_INVALID_PID  
-->  
xcsCheckProcess  
<-- xcsCheckProcess rc=xecP_E_INVALID_PID  
-->  
xlsThreadInitialization  
-->  
xcsConnectSharedMem  
-->  
xcsRequestThreadMutexSem  
<-- xcsRequestThreadMutexSem rc=OK  
-->  
xihGetConnSPDetailsFromList  
<-- xihGetConnSPDetailsFromList rc=OK
```

```

-->
xstCreateConnExtentList
<-- xstCreateConnExtentList rc=OK
-->
xstConnSetInSPbyHandle
-->
xstSerialiseSPList
-->
xllSpinLockRequest
<-- xllSpinLockRequest rc=OK
<-- xstSerialiseSPList rc=OK
-->
xstGetSetDetailsFromSPbyHandle
<-- xstGetSetDetailsFromSPbyHandle rc=OK
-->
xstConnSharedMemSet
-->
xstConnectExtent
-->
xstAddConnExtentToList
<-- xstAddConnExtentToList rc=OK
<-- xstConnectExtent rc=OK
-->
xcsBuildDumpPtr
-->
xcsGetMem
<-- xcsGetMem rc=OK
<-- xcsBuildDumpPtr rc=OK
-->
xcsBuildDumpPtr
<-- xcsBuildDumpPtr rc=OK
-->
xcsBuildDumpPtr
<-- xcsBuildDumpPtr rc=OK
-->
xcsFFST

Process Control Block
SPP:0000 :1aefSTRMQM_R MAYFCT 020100 :8bba0:0:6d E7C9C8D7 000004E0 00000699 00000000 XIHP...\...r...
SPP:0000 :1aefSTRMQM_R MAYFCT 020100 :8bbb0:1:6d 00000000 00000002 00000000 00000000 .....
SPP:0000 :1aefSTRMQM_R MAYFCT 020100 :8bbc0:2:6d 80000000 00000000 EC161F7C FC002DB0 .....@...¢
SPP:0000 :1aefSTRMQM_R MAYFCT 020100 :8bbd0:3:6d 80000000 00000000 EC161F7C FC002DB0 .....@...¢
SPP:0000 :1aefSTRMQM_R MAYFCT 020100 :8bbe0:4:6d 00000000 00000000 00000000 00000000 .....

Thread Control Block
SPP:0000 :1aefSTRMQM_R MAYFCT 020100 :1db0:20:6d E7C9C8E3 00001320 00000000 00000000 XIHT.....
SPP:0000 :1aefSTRMQM_R MAYFCT 020100 :1dc0:21:6d 00000001 00000000 00000000 00000000 .....
SPP:0000 :1aefSTRMQM_R MAYFCT 020100 :1dd0:22:6d 80000000 00000000 DD13C17B 81001000 .....A#a...
SPP:0000 :1aefSTRMQM_R MAYFCT 020100 :1de0:23:6d 00000000 00000046 00000002 00000001 .....
SPP:0000 :1aefSTRMQM_R MAYFCT 020100 :1df0:24:6d 00000000 00000000 00000000 00000000 .....

RecoveryIndex
SPP:0000 :1aefSTRMQM_R MAYFCT 020100 :2064:128:6d 00000000 .....

```

注:

1. MQM Trace History 部分是 200 个最新函数跟踪语句的日志，记录在 FFST 报告中，而不考虑任何 TRCMQM 设置。
2. 仅记录连接到队列管理器子池的作业的队列管理器详细信息。
3. 如果失败的组件为 xehAS400ConditionHandler，那么其他数据记录在 errors 目录中，以提供来自与异常条件有关的作业日志的摘要。

函数堆栈和跟踪历史记录是由 IBM 用来辅助进行问题确定的。如果生成了 FFST 报告，在很多时候，系统管理员除了将问题报告给 IBM 支持中心，就无能为力了。

WCF XMS First Failure Support Technology (FFST)

您可以通过使用 IBM MQ 跟踪收集有关不同的 IBM MQ 代码部分有何作用的详细信息。XMS FFST 针对 WCF 定制通道具有自己的配置和输出文件。

以前是使用基本名称和进程标识格式 `xmsffdc pid_date.txt`（其中 `pid` 是进程标识，`date` 是时间和日期）来命名 XMS FFST 跟踪文件。

由于仍会随 WCF 定制通道 XMS FFST 文件一起生成 XMS FFST 跟踪文件，因此 WCF 定制通道 XMS FFST 输出文件采用以下格式来避免混淆：`wcf_ffdc_pid_date.txt`（其中 *pid* 是进程标识，*date* 是时间和日期）。

缺省情况下，会在当前工作目录中创建此跟踪输出文件，但如果需要，也可以重新定义该目标。

具有 XMS .NET 跟踪头的 WCF 定制通道与以下示例类似：

```
***** Start Display XMS WCF Environment *****
Product Name :- value
WCF Version :- value
Level :- value
***** End Display XMS WCF Environment *****
```

使用标准方式格式化 FFST 跟踪文件，没有任何特定于该定制通道的格式化。

XMS .NET 应用程序的 FFDC 配置

对于 XMS 的 .NET 实现，将为每个 FFDC 生成一个 FFDC 文件。

采用人类可读的文本文件形式存储首次故障数据捕获 (FFDC) 文件。这些文件的名称格式为 `xmsffdcprocessID_DateTTimestamp.txt`。文件名的示例为 `xmsffdc264_2006.01.06T13.18.52.990955.txt`。时间戳记包含微秒分辨率。

文件以异常发生的日期和时间开头，后跟异常类型。文件包含唯一的短 `probeId`，可用于查找在何处发生此 FFDC。

无需进行任何配置即可开启 FFDC。缺省情况下，会将所有 FFDC 文件写入当前目录中。但是，如果需要，您可以通过在应用程序配置文件的 `Trace` 节中更改 `ffdcDirectory` 来指定其他目录。在以下示例中，所有跟踪文件都记录到 `c:\client\ffdc` 目录中。

```
<IBM.XMS>
  <Trace ffdc=true ffdcDirectory="c:\client\ffdc"/>
</IBM.XMS>
```

您可以通过在应用程序配置文件的 `Trace` 节中将 FFDC 设置为 `false` 来禁用跟踪。

如果您没有使用应用程序配置文件，那么 FFDC 处于开启状态，而跟踪处于关闭状态。

使用跟踪

您可以使用不同类型的跟踪来帮助您确定问题并进行故障诊断。

关于此任务

使用此信息来了解不同类型的跟踪，以及如何针对您的平台运行跟踪。

- **Windows** 第 315 页的『使用 Windows 上的跟踪』
- **Linux** **UNIX** 第 305 页的『在 UNIX and Linux 系统上使用跟踪』
- **IBM i** 第 310 页的『将跟踪与 IBM i 上的 IBM MQ 服务器配合使用』
- **IBM i** 第 313 页的『将跟踪与 IBM i 上的 IBM MQ 客户机配合使用』
- **z/OS** 第 318 页的『在 z/OS 上使用跟踪进行问题确定』
- 第 371 页的『跟踪 TLS: `runmqakm`、`strmqikm` 和 `runmqckm` 功能』
- 第 339 页的『跟踪 IBM MQ classes for JMS 应用程序』
- 第 342 页的『跟踪 IBM MQ classes for Java 应用程序』
- 第 346 页的『跟踪 IBM MQ 资源适配器』

- [第 347 页的『跟踪其他 IBM MQ Java 组件』](#)
- [第 349 页的『使用 IBM MQ classes for Java 和 IBM MQ classes for JMS 控制运行进程中的跟踪』](#)

相关概念

[第 285 页的『使用错误日志』](#)

有各种错误日志可用于帮助进行问题确定和故障诊断。

[第 293 页的『First Failure Support Technology \(FFST\)』](#)

First Failure Support Technology (FFST) for IBM MQ 可提供有关事件的信息，在发生错误的情况下，这些信息有助于 IBM 支持人员对问题进行诊断。

相关任务

[第 5 页的『IBM MQ 故障诊断和支持』](#)

如果您的队列管理器网络或 IBM MQ 应用程序存在问题，那么可以使用本信息中描述的方法来帮助您诊断和解决问题。如果需要有关问题的帮助，可以通过 IBM 支持站点联系 IBM 支持人员。

[第 227 页的『联系 IBM 支持人员』](#)

如果您在 IBM MQ 方面遇到问题需要帮助，可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订、故障诊断和其他新闻的通知。

Linux

UNIX

在 UNIX and Linux 系统上使用跟踪

使用 **strmqtrc** 和 **endmqtrc** 命令来启动和结束跟踪，并使用 **dspmqtrc** 来显示跟踪文件

UNIX and Linux 系统对 IBM MQ MQI client 跟踪功能使用以下命令：

strmqtrc

启动跟踪

endmqtrc

结束跟踪

dspmqtrc filename

用于显示格式化跟踪文件

跟踪功能使用几个文件，它们是：

- 每个被跟踪的实体一个文件，在其中记录跟踪信息
- 每台机器上一个附加文件，提供用于启动和结束跟踪的共享内存的引用
- 一个文件在更新共享内存时标识使用的信号量

与跟踪关联的文件在文件树中的固定位置创建，即为：`/var/mqm/trace`。

全部客户机跟踪将记录在此目录内的文件中。

通过在这个目录上挂装一个临时文件系统，您可以处理大型的跟踪文件。

在 AIX 上，除使用 **strmqtrc** 和 **endmqtrc** 命令以外，还可以使用 AIX 系统跟踪。有关更多信息，请参阅[第 307 页的『使用 AIX 系统跟踪进行跟踪』](#)。

IBM MQ for UNIX 和 Linux 系统上的跟踪文件

在 `/var/mqm/trace` 目录中创建跟踪文件。

注：您可以通过在包含跟踪文件的目录上挂装一个临时文件系统来容纳大型跟踪文件的产生。或者，重命名跟踪目录并创建指向其他目录的符号链接 `/var/mqm/trace`。

跟踪文件名为 `AMQppppp.qq.TRC`，其中变量为：

ppppp

报告错误的进程的标识。

qq

一个序号，从 0 开始。如果存在完整的文件名，那么此值会递增 1，直到找到唯一的跟踪文件名。如果重复使用某个进程，那么可能存在跟踪文件名。

注：

1. 进程标识可以包含比示例中所示更少或更多的位数。
2. 对于运行为受跟踪实体的一部分的每个进程，都有一个跟踪文件。

要格式化或查看跟踪文件，您必须是该跟踪文件的创建者或 mqm 组的成员。

SSL 跟踪文件的名称为 AMQ.SSL.TRC 和 AMQ.SSL.TRC.1。您无法格式化 SSL 跟踪文件；请不要更改这些文件并将其发送给 IBM 支持人员。

如何启动和停止跟踪

在 IBM MQ for UNIX 和 Linux 系统中，使用 **strmqtrc** 控制命令启用或修改跟踪（请参阅 [strmqtrc](#)）。要停止跟踪，请使用 **endmqtrc** 控制命令（请参阅 [endmqtrc](#)）。在 IBM MQ for Linux (x86 和 x86-64 平台) 系统上，也可以使用 IBM MQ Explorer 来启动和停止跟踪。但是，您只能使用提供的函数（相当于使用命令 **strmqtrc -e** 和 **endmqtrc -e**）来跟踪所有内容。

跟踪输出未格式化；请在查看之前使用 **dspmqtrc** 控制命令对跟踪输出进行格式化。例如，要格式化当前目录中的所有跟踪文件，可使用以下命令：

```
dspmqtrc *.TRC
```

有关控制命令 **dspmqtrc** 的详细信息，请参阅 [dspmqtrc](#)。

IBM MQ for UNIX 和 Linux 系统上的选择性组件跟踪

使用 **-t** 和 **-x** 选项来控制要记录的跟踪详细信息量。缺省情况下，启用所有跟踪点。使用 **-x** 选项指定不希望跟踪的点。例如，如果要针对队列管理器 QM1 仅跟踪与使用传输层安全性 (TLS) 通道安全性关联的输出数据，请使用：

```
strmqtrc -m QM1 -t ssl
```

有关跟踪命令的详细信息，请参阅 [strmqtrc](#)。

IBM MQ for AIX 上的选择性组件跟踪

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。

由于 MQS_TRACE_OPTIONS 支持在没有高详细信息功能和参数跟踪功能的情况下激活跟踪，因此可以当尝试在启用跟踪的情况下重现问题时，使用该变量减少对性能和跟踪大小的影响。

请仅在服务人员指导设置环境变量 MQS_TRACE_OPTIONS 的情况下才执行此操作。

通常，MQS_TRACE_OPTIONS 必须设置在启动队列管理器的进程中，并在队列管理器启动之前进行设置，否则无法识别该变量。请在跟踪启动之前设置 MQS_TRACE_OPTIONS。如果在跟踪启动之后设置该变量，那么其无法识别。

IBM MQ for UNIX 和 Linux 系统上的选择性进程跟踪

使用 **strmqtrc** 命令控制的 **-p** 选项将跟踪生成限制到指定的命名进程。例如，要跟踪任何名为 amqxxx 的运行进程产生的所有线程，请使用以下命令：

```
strmqtrc -p amqxxx
```

有关跟踪命令的详细信息，请参阅 [strmqtrc](#)。

相关概念

[第 310 页的『将跟踪与 IBM i 上的 IBM MQ 服务器配合使用』](#)
使用 TRCMQM 命令来启动和停止跟踪并指定所需的跟踪类型。

[第 318 页的『在 z/OS 上使用跟踪进行问题确定』](#)

有不同的跟踪选项可用于对 IBM MQ 进行问题确定。使用本主题来了解不同选项以及如何控制跟踪。

[第 371 页的『跟踪 TLS: runmqakm、strmqikm 和 runmqckm 功能』](#)

如何跟踪传输层安全性 (TLS) 以及请求 **runmqakm** 跟踪与 **strmqikm** (iKeyman) 和 **runmqckm** (iKeycmd) 跟踪。

第 347 页的『跟踪其他 IBM MQ Java 组件』

对于 IBM MQ 的 Java 组件，例如 IBM MQ Explorer 和 IBM MQ Transport for SOAP 的 Java 实现，诊断信息是使用标准 IBM MQ 诊断工具或由 Java 诊断类输出的。

相关参考

第 315 页的『使用 Windows 上的跟踪』

使用 **strmqtrc** 和 **endmqtrc** 命令或 IBM MQ Explorer 界面来启动和结束跟踪。

AIX 使用 AIX 系统跟踪进行跟踪

除 IBM MQ 跟踪以外，IBM MQ for AIX 用户还可以使用标准 AIX 系统跟踪。

注：仅当 IBM 服务人员指示您使用 *aix* 选项时，才应使用此选项。

AIX 系统跟踪是一个三步骤过程：

1. 将 **strmqtrc** 命令上的 **-o** 参数设置为 *aix*。
2. 收集数据，然后运行 **endmqtrc** 命令。
3. 设置结果的格式。

IBM MQ 使用两个跟踪挂钩标识：

X'30D'

此事件由 IBM MQ 在进入或退出子例程时记录。

X'30E'

此事件由 IBM MQ 记录，以跟踪诸如跨通信网络发送或接收的数据。

跟踪提供详细执行跟踪来帮助分析问题。IBM 服务支持人员可能会请求在启用跟踪的情况下重现问题。跟踪产生的文件可能**非常大**，因此尽可能限定跟踪很重要。例如，您可以选择性地按时间和按组件来限定跟踪。

有两种运行跟踪的方法：

1. 交互方式。

以下命令序列对程序 **myprog** 运行交互式跟踪并结束跟踪。

```
trace -j30D,30E -o trace.file
->!myprog
->q
```

2. 异步方式。

以下命令序列对程序 **myprog** 运行异步跟踪并结束跟踪。

```
trace -a -j30D,30E -o trace.file
myprog
trcstop
```

您可以使用以下命令对跟踪文件进行格式化：

```
trcrpt -t MQ_INSTALLATION_PATH/lib/amqtrc.fmt trace.file > report.file
```

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

report.file 是要将格式化跟踪输出放置所在的文件的名称。

注：在跟踪处于活动状态时，机器上的**所有** IBM MQ 活动都被跟踪。

从 UNIX 和 Linux 的跟踪文件中抽取。

AIX 的示例

AIX

第 308 页的图 56 显示了从 IBM MQ for AIX 跟踪中抽取的内容:

```

Timestamp          Process.Thread Trace Ident Trace Data
=====
12:06:32.904335    622742.1      :          Header.v02:9.0:AIX 7.2:64:-1:1:GMT
12:06:32.904427    622742.1      :          Version : 9.0.0.0   Level : p000-L090514
12:06:32.904540    622742.1      :          UTC   Date : 05/15/16   Time :
11:06:32.904302
12:06:32.904594    622742.1      :          Local Date : 05/15/16   Time :
12:06:32.904302    GMT
12:06:32.904697    622742.1      :          PID : 622742 Process : dltmqm_nd (64-bit)
12:06:32.904728    622742.1      :          Host : dynamo
12:06:32.904755    622742.1      :          Operating System : AIX 7.2
12:06:32.904781    622742.1      :          Product Long Name : IBM MQ for AIX
12:06:32.904806    622742.1      :          -----
12:06:32.904832    622742.1      :          xtrNullFd: 3, xihTraceFileNum: 5
12:06:32.904916    622742.1      :          Data: 0x00000000
12:06:32.904952    622742.1      :          Thread stack
12:06:32.904982    622742.1      :          -> InitProcessInitialisation
12:06:32.905007    622742.1      :          { InitProcessInitialisation
12:06:32.905033    622742.1      :          -{ xcsIsEnvironment
12:06:32.905062    622742.1      :          xcsIsEnvironment[AMQ_NO_CS_RELOAD] = FALSE
12:06:32.905088    622742.1      :          -} xcsIsEnvironment rc=OK
12:06:32.905117    622742.1      :          -{ xcsLoadFunction
12:06:32.905145    622742.1      :          LibName(libmqmcs_r.a(shr.o))
LoadType(2097200)
12:06:32.905178    622742.1      :          General, comms, CS, OAM, or WAS
12:06:32.905204    622742.1      :          --{ xcsQueryValueForSubpool
12:06:32.905282    622742.1      :          --{ xcsQueryValueForSubpool rc=OK
12:06:32.905504    622742.1      :          FullPathLibName(/usr/mqm/lib64/
libmqmcs_r.a(shr.o)) loaded with load
12:06:32.905540    622742.1      :          --{ xcsGetMem
12:06:32.905575    622742.1      :          component:24 function:176 length:2088
options:0 cbmindex:-1 *pointer:110011408
12:06:32.905601    622742.1      :          --{ xcsGetMem rc=OK
12:06:32.905638    622742.1      :          Handle(0) Function(0)
FullPathLibName(/usr/mqm/lib64/libmqmcs_r.a(shr.o))
12:06:32.905665    622742.1      :          -} xcsLoadFunction rc=OK

```

图 56: 样本 IBM MQ for AIX 跟踪

Linux 的示例

Linux

第 309 页的图 57 显示了从 IBM MQ for Linux 跟踪中抽取的内容:

```

Timestamp      Process.Thread Trace Ident Trace Data
=====
11:02:23.643879 1239.1      :      Header.v02:9.0:Linux RHEL Server 7
7.2:64:-1:1:GMT
11:02:23.643970 1239.1      :      Version : 9.0.0.0   Level : p000-L090514
11:02:23.644025 1239.1      :      UTC   Date : 05/15/16   Time :
10:02:23.643841
11:02:23.644054 1239.1      :      Local Date : 05/15/16   Time :
11:02:23.643841 GMT
11:02:23.644308 1239.1      :      PID : 1239 Process : dltmqm (64-bit)
11:02:23.644324 1239.1      :      Host : hall
11:02:23.644334 1239.1      :      Operating System : RHEL Server 7 7.2
11:02:23.644344 1239.1      :      Product Long Name : IBM MQ for Linux (x86
platform)
11:02:23.644353 1239.1      :      -----
11:02:23.644363 1239.1      :      xtrNullFd: 3, xihTraceFileNum: 4
11:02:23.644394 1239.1      :      Thread stack
11:02:23.644412 1239.1      :      -> InitProcessInitialisation
11:02:23.644427 1239.1      :      { InitProcessInitialisation
11:02:23.644439 1239.1      :      -{ xcsIsEnvironment
11:02:23.644469 1239.1      :      xcsIsEnvironment[AMQ_NO_CS_RELOAD] = FALSE
11:02:23.644485 1239.1      :      -} xcsIsEnvironment rc=OK
11:02:23.644504 1239.1      :      -{ xcsLoadFunction
11:02:23.644519 1239.1      :      LibName(libmqmcs_r.so) LoadType(2097200)
11:02:23.644537 1239.1      :      General, comms, CS, OAM, or WAS
11:02:23.644558 1239.1      :      --{ xcsQueryValueForSubpool
11:02:23.644579 1239.1      :      --} xcsQueryValueForSubpool rc=OK
11:02:23.644641 1239.1      :      FullPathLibName(/opt/mqm/lib/
libmqmcs_r.so) loaded with dlopen
11:02:23.644652 1239.1      :      --{ xcsGetMem
11:02:23.644675 1239.1      :      component:24 function:176 length:8212
options:0 cbindex:-1 *pointer:0x8065908
11:02:23.644685 1239.1      :      --} xcsGetMem rc=OK
11:02:23.644722 1239.1      :      Handle((nil)) Function((nil))
FullPathLibName(/opt/mqm/lib/libmqmcs_r.so)
11:02:23.644732 1239.1      :      -} xcsLoadFunction rc=OK
11:02:23.644753 1239.1      :      SystemPageSize is 4096.

```

图 57: 样本 IBM MQ for Linux 跟踪

Solaris 的示例

Solaris

第 310 页的图 58 显示了从 IBM MQ for Solaris 跟踪中抽取的内容:

```

Timestamp      Process.Thread Trace Ident Trace Data
=====
11:48:57.905466 7078.1      :      Header.v02:7.0:SunOS 5.9:64:-1:1:GMT
11:48:57.905625 7078.1      :      Version : 7.0.0.0   Level : p000-L090514
11:48:57.905770 7078.1      :      UTC   Date : 05/15/09 Time :
10:48:57.905364
11:48:57.905816 7078.1      :      Local Date : 05/15/09 Time :
11:48:57.905364 GMT
11:48:57.906104 7078.1      :      PID : 7078 Process : dltmqm_nd (64-bit)
11:48:57.906129 7078.1      :      Host : computer.v6.hursley.ibm.com
11:48:57.906148 7078.1      :      Operating System : SunOS 5.9
11:48:57.906167 7078.1      :      Product Long Name : WebSphere MQ for
Solaris (SPARC platform)
11:48:57.906184 7078.1      :      -----
11:48:57.906203 7078.1      :      xtrNullFd: 4, xihTraceFileNum: 5
11:48:57.906276 7078.1      :      Thread stack
11:48:57.906353 7078.1      :      { xcsInitialize
11:48:57.906385 7078.1      :      -{ InitPrivateServices
11:48:57.906439 7078.1      :      --{ xcsGetEnvironmentString
11:48:57.906566 7078.1      :
xcsGetEnvironmentString[MQS_ACTION_ON_EXCEPTION] = NULL
11:48:57.906608 7078.1      :      --{! xcsGetEnvironmentString
rc=xecE_E_ENV_VAR_NOT_FOUND
11:48:57.906709 7078.1      :      --{ xcsIsEnvironment
11:48:57.906738 7078.1      :      xcsIsEnvironment[AMQ_SIGCHLD_SIGACTION] =
FALSE
11:48:57.906755 7078.1      :      --{ xcsIsEnvironment rc=OK
11:48:57.906771 7078.1      :      AMQ_SIGCHLD_SIGACTION is not set
11:48:57.906835 7078.1      :      --{ xcsIsEnvironment
11:48:57.906862 7078.1      :
xcsIsEnvironment[MQS_NO_SYNC_SIGNAL_HANDLING] = FALSE
11:48:57.906878 7078.1      :      --{ xcsIsEnvironment rc=OK
11:48:57.907000 7078.1      :      FPE Handler installed, New=7e0b0f38, Old=0
11:48:57.907035 7078.1      :      SEGV Handler installed, New=7e0b0f38, Old=0
11:48:57.907063 7078.1      :      BUS Handler installed, New=7e0b0f38, Old=0
11:48:57.907091 7078.1      :      ILL Handler installed, New=7e0b0f38, Old=0
11:48:57.907109 7078.1      :      Synchronous Signal Handling Activated

```

图 58: 样本 IBM MQ for Solaris 跟踪

IBM i IBM i 上的跟踪

在 IBM i 上，服务器和客户机安装之间的跟踪几乎完全相同。但是，某些跟踪选项仅在服务器安装上可用。

关于此任务

在 IBM i 上，服务器和客户机都支持在 IBM i 命令行上通过调用 QMQM/STRMQTRC 和 QMQM/ENDMQTRC 程序进行跟踪，并且都支持在 IBM i Qshell 上使用 **STRMQTRC**、**ENDMQTRC** 和 **DSPMQTRC** 命令进行跟踪。

但是，只有 IBM i 的 IBM MQ 服务器安装提供了 **TRCMQM** 命令。此外，独立客户机在启动或结束跟踪命令上不支持 **-m** 参数，因为没有队列管理器。由于同一原因，**runmqras -qmlist** 参数在独立客户机上无效。

IBM i 将跟踪与 IBM i 上的 IBM MQ 服务器配合使用

使用 **TRCMQM** 命令来启动和停止跟踪并指定所需的跟踪类型。

使用跟踪有两个阶段：

1. 决定您是否要早期跟踪。通过早期跟踪，可以跟踪队列管理器的创建和启动。但请注意，早期跟踪很容易生成大量跟踪，因为它通过跟踪所有队列管理器的所有作业来实施。要启用早期跟踪，请使用 **TRCMQM** 并将 **TRCEARLY** 参数设置为 ***YES**。
2. 使用 **TRCMQM *ON** 启动跟踪工作。要停止跟踪，您有两个选项：
 - **TRCMQM *OFF**，用于停止收集一个队列管理器的跟踪记录。跟踪记录写入到 **/QIBM/UserData/mqm/trace** 目录中的文件。

- TRCMQM *END, 用于停止收集所有队列管理器的跟踪记录并禁用早期跟踪。此选项忽略 TRCEARLY 参数的值。

使用设置为以下值之一的 TRCLEVEL 参数来指定所需的详细信息级别:

***DFT**

表示流处理跟踪点的最低详细信息级别。

***DETAIL**

表示流处理跟踪点的高详细信息级别。

***PARMS**

表示流处理跟踪点的缺省详细信息级别。

使用设置为以下值之一的 OUTPUT 参数来指定所需的跟踪输出类型:

***MQM**

收集 TRCDIR 参数指定的目录中的二进制 IBM MQ 跟踪输出。该值为缺省值。

***MQMFMFMT**

收集 TRCDIR 参数指定的目录中的格式化 IBM MQ 跟踪输出。

***PEX**

收集性能探测器 (PEX) 跟踪输出

***ALL**

收集 IBM MQ 未格式化跟踪和 PEX 跟踪输出

选择性跟踪

您可以使用命令 TRCMQM (其中 F4=prompt) 减少保存的跟踪数据量, 从而提高运行时性能, 然后使用 F9 来定制 TRCTYPE 和 EXCLUDE 参数:

TRCTYPE

指定要存储在跟踪文件中的跟踪数据的类型。如果省略此参数, 那么将启用除 EXCLUDE 中指定的跟踪点以外的所有跟踪点。

EXCLUDE

指定要在跟踪文件中省略的跟踪数据的类型。如果省略此参数, 那么将启用 TRCTYPE 中指定的所有跟踪点。

在 TRCTYPE 和 EXCLUDE 上均可用的选项包括:

***ALL (仅 TRCTYPE)**

以下关键字指定的所有跟踪数据都存储在跟踪文件中。

trace-type-list

可以从以下关键字指定多个选项, 但是每个选项只能出现一次。

***API**

与 MQI 和主队列管理器组件相关联的跟踪点的输出数据。

***CMTRY**

与 IBM MQ 组件中的注释相关联的跟踪点的输出数据。

***COMMS**

与流经通信网络的数据相关联的跟踪点的输出数据。

***CSDATA**

与公共服务中的内部数据缓冲区相关联的跟踪点的输出数据。

***CSFLOW**

与公共服务中的处理流相关联的跟踪点的输出数据。

***LQMDATA**

与本地队列管理器中的内部数据缓冲区相关联的跟踪点的输出数据。

***LQMFLOW**

与本地队列管理器中的处理流相关联的跟踪点的输出数据。

***OTHDATA**

与其他组件中的内部数据缓冲区相关联的跟踪点的输出数据。

***OTFLOW**

与其他组件中的处理流相关联的跟踪点的输出数据。

***RMTDATA**

与通信组件中的内部数据缓冲区相关联的跟踪点的输出数据。

***RMTFLOW**

与通信组件中的处理流相关联的跟踪点的输出数据。

***SVCDATA**

与服务组件中的内部数据缓冲区相关联的跟踪点的输出数据。

***SVCFLOW**

与服务组件中的处理流相关联的跟踪点的输出数据。

***VSNDATA**

与正在运行的 IBM MQ 版本相关联的跟踪点的输出数据。

回绕跟踪

使用 MAXSTG 参数回绕跟踪，和指定用于收集的跟踪记录的最大存储器大小。

选项为：

***DFT**

不启用跟踪回绕。对于每个作业，跟踪数据写入到后缀为 .TRC 的文件，直至停止跟踪。

maximum-K-bytes

启用跟踪回绕。当跟踪文件达到其最大大小时，它使用后缀 .TRS 进行重命名，并将打开后缀为 .TRC 的新跟踪文件。系统会删除任何现有 .TRS 文件。请指定范围在 1 到 16 000 内的值。

格式化跟踪输出

要格式化任何跟踪输出，请执行以下操作：

- 输入 QShell
- 输入以下命令

```
/QSYS.LIB/QMQM.LIB/DSPMQTRC.PGM [-t Format] [-h] [-s]  
[-o OutputFileName] InputFileName
```

其中：

InputFileName

必需参数，用于指定包含未格式化跟踪的文件的名称。例如，/QIBM/UserData/mqm/trace/AMQ12345.TRX。

-t FormatTemplate

指定包含如何显示跟踪的详细信息模板文件的名称。缺省值为 /QIBM/ProdData/mqm/lib/amqtrc.fmt。

-h

省略报告的头信息。

-s

抽取跟踪头并放到标准输出。

-o output_filename

格式化的数据写入至的文件名。

还可指定 dspmqtrc * 以格式化所有跟踪。

相关概念

第 305 页的『在 UNIX and Linux 系统上使用跟踪』

使用 **strmqtrc** 和 **endmqtrc** 命令来启动和结束跟踪，并使用 **dspmqtrc** 来显示跟踪文件

第 318 页的『在 z/OS 上使用跟踪进行问题确定』

有不同的跟踪选项可用于对 IBM MQ 进行问题确定。使用本主题来了解不同选项以及如何控制跟踪。

第 371 页的『跟踪 TLS: runmqakm、strmqikm 和 runmqckm 功能』

如何跟踪传输层安全性 (TLS) 以及请求 **runmqakm** 跟踪与 **strmqikm** (iKeyman) 和 **runmqckm** (iKeycmd) 跟踪。

第 347 页的『跟踪其他 IBM MQ Java 组件』

对于 IBM MQ 的 Java 组件，例如 IBM MQ Explorer 和 IBM MQ Transport for SOAP 的 Java 实现，诊断信息是使用标准 IBM MQ 诊断工具或由 Java 诊断类输出的。

相关参考

第 315 页的『使用 Windows 上的跟踪』

使用 **strmqtrc** 和 **endmqtrc** 命令或 IBM MQ Explorer 界面来启动和结束跟踪。

IBM i 将跟踪与 IBM i 上的 IBM MQ 客户机配合使用

在 IBM i 上，当使用独立 IBM MQ MQI client 时，没有用于捕获跟踪的控制语言 (CL) 命令。STRMQTRC 和 ENDMQTRC 程序可以用于启用和禁用跟踪。

启动跟踪的示例：

```
CALL PGM(QMQM/STRMQTRC) PARM('-e' '-t' 'all' '-t' 'detail')
Where -e option requests early tracing of all the process -t option for trace type
```

结束跟踪

```
CALL PGM(QMQM/ENDMQTRC) PARM('-e')
```

• 可选参数：

-t TraceType

要跟踪的点和要记录的跟踪详细信息量。缺省情况下，启用所有跟踪点并生成缺省详细信息跟踪。

或者，您可以在表 1 中提供一个或多个选项。对于您指定的每个 *TraceType* 值（包括 **-t all**），请指定 **-t parms** 或 **-t detail** 以获取适当级别的跟踪详细信息。如果对于任何特定跟踪类型不指定 **-t parms** 或 **-t detail**，那么仅为该跟踪类型生成缺省详细信息跟踪。

如果提供多个跟踪类型，那么每个跟踪类型必须具有其自己的 **-t** 标志。可以包含任意数量的 **-t** 标志，前提是每个标志具有与其关联的有效跟踪类型。

在多个 **-t** 标志上指定同一跟踪类型并非错误。

请参阅下表以获取允许的 *TraceType* 值。

| 值 | 描述 |
|------------|---|
| all | 系统中每个跟踪点的输出数据（缺省）。使用 <i>all</i> 可激活缺省详细信息级别的跟踪。 |
| api | 与消息队列接口 (MQI) 和主队列管理器组件相关联的跟踪点的输出数据。 |
| commentary | 与 IBM MQ 组件中的注释相关联的跟踪点的输出数据。 |
| comms | 与流通信网络的数据相关联的跟踪点的输出数据。 |
| csdata | 与公共服务中的内部数据缓冲区相关联的跟踪点的输出数据。 |
| csflows | 与公共服务中的处理流相关联的跟踪点的输出数据。 |
| 详细信息 | 为流处理跟踪点激活高详细信息级别的跟踪。 |
| lqmdata | 与本地队列管理器中的内部数据缓冲区相关联的跟踪点的输出数据。 |

| 表 26: <i>TraceType</i> 值 (继续) | |
|-------------------------------|-------------------------------|
| 值 | 描述 |
| lqmflo ws | 与本地队列管理器中的处理流相关联的跟踪点的输出数据。 |
| otherd ata | 与其他组件中的内部数据缓冲区相关联的跟踪点的输出数据。 |
| otherfl ows | 与其他组件中的处理流相关联的跟踪点的输出数据。 |
| parms | 为流处理跟踪点激活缺省详细信息级别的跟踪。 |
| remote data | 与通信组件中的内部数据缓冲区相关联的跟踪点的输出数据。 |
| remote flows | 与通信组件中的处理流相关联的跟踪点的输出数据。 |
| service data | 与服务组件中的内部数据缓冲区相关联的跟踪点的输出数据。 |
| service flows | 与服务组件中的处理流相关联的跟踪点的输出数据。 |
| version data | 与正在运行的 IBM MQ 版本相关联的跟踪点的输出数据。 |

-x *TraceType*

不跟踪的点。缺省情况下，启用所有跟踪点并生成缺省详细信息跟踪。可以指定的 *TraceType* 值与表 1 中为 -t 标志列出的值相同。

可以将 -x 标志与 *TraceType* 值结合使用，以排除不希望记录的跟踪点。排除指定的跟踪点有助于减少产生的跟踪量。

如果提供多个跟踪类型，那么每个跟踪类型必须具有其自己的 -x 标志。可以包含任意数量的 -x 标志，前提是每个标志具有与其关联的有效 *TraceType*。

-s

报告当前有效的跟踪选项。必须单独使用此参数，而不带任何其他参数。

有限数量的插槽可用于存储跟踪命令。当所有插槽都在使用中时，除非更换现有插槽，否则无法再接受跟踪命令。插槽号不固定，因此如果移除了插槽号 0 中的命令（例如，通过 **endmqtrc** 命令），那么所有其他插槽都会上移，例如，插槽 1 变为插槽 0。字段中的星号 (*) 意味着未定义任何值，并且等效于星号通配符。

-l *MaxSize*

跟踪文件 (AMQppppp.qq.TRC) 的最大大小 (MB)。例如，如果将 *MaxSize* 指定为 1，那么跟踪的大小限制为 1 MB。

当跟踪文件达到指定的最大值时，它会重命名为 AMQppppp.qq.TRS，并启动新的 AMQppppp.qq.TRC 文件。如果已存在 AMQppppp.qq.TRS 文件的先前副本，那么会将其删除。

MaxSize 可以达到的最高值为 2048 MB。

-e

请求对所有进程进行早期跟踪

有关更多详细信息，请参阅 **strmqtrc** 命令

- 结束跟踪：

```
/QSYS.LIB/QMQM.LIB/ENDMQTRC.PGM [-e] [-a]
```

其中：

-e

结束对所有进程的早期跟踪。

使用不带任何参数的 **endmqtrc** 与 **endmqtrc -e** 效果相同。不能将 **-e** 标志与 **-m** 标志、**-i** 标志或 **-p** 标志一起指定。

-a

结束所有跟踪。

有关更多详细信息，请参阅 [endmqtrc endmqtrc 命令](#)

- 显示格式化跟踪文件：

```
/QSYS.LIB/QMQM.LIB/DSPMQTRC.pgm
```

要检查 First Failure Support Technology (FFST) 文件，请参阅第 302 页的『[FFST: IBM MQ for IBM i](#)』。

相关概念

第 305 页的『[在 UNIX and Linux 系统上使用跟踪](#)』

使用 **strmqtrc** 和 **endmqtrc** 命令来启动和结束跟踪，并使用 **dspmqtrc** 来显示跟踪文件

第 318 页的『[在 z/OS 上使用跟踪进行问题确定](#)』

有不同的跟踪选项可用于对 IBM MQ 进行问题确定。使用本主题来了解不同选项以及如何控制跟踪。

第 371 页的『[跟踪 TLS: runmqakm、strmqikm 和 runmqckm 功能](#)』

如何跟踪传输层安全性 (TLS) 以及请求 **runmqakm** 跟踪与 **strmqikm** (iKeyman) 和 **runmqckm** (iKeycmd) 跟踪。

第 347 页的『[跟踪其他 IBM MQ Java 组件](#)』

对于 IBM MQ 的 Java 组件，例如 IBM MQ Explorer 和 IBM MQ Transport for SOAP 的 Java 实现，诊断信息是使用标准 IBM MQ 诊断工具或由 Java 诊断类输出的。

相关参考

第 315 页的『[使用 Windows 上的跟踪](#)』

使用 **strmqtrc** 和 **endmqtrc** 命令或 IBM MQ Explorer 界面来启动和结束跟踪。

Windows 使用 Windows 上的跟踪

使用 **strmqtrc** 和 **endmqtrc** 命令或 IBM MQ Explorer 界面来启动和结束跟踪。

Windows 对客户机跟踪功能使用以下命令：

strmqtrc

启动跟踪

endmqtrc

结束跟踪

将在 MQ_DATA_PATH/trace 目录中创建输出文件。

IBM MQ for Windows 上的跟踪文件

跟踪文件名为 AMQppppp.qq.TRC，其中变量为：

ppppp

报告错误的进程的标识。

qq

一个序号，从 0 开始。如果存在完整的文件名，那么此值会递增 1，直到找到唯一的跟踪文件名。如果重复使用某个进程，那么可能存在跟踪文件名。

注：

1. 进程标识可以包含比示例中所示更少或更多的位数。
2. 对于运行受跟踪实体的一部分的每个进程，都有一个跟踪文件。

要格式化或查看跟踪文件，您必须是该跟踪文件的创建者或 mqm 组的成员。

SSL 跟踪文件的名称为 AMQ.SSL.TRC 和 AMQ.SSL.TRC.1。您无法格式化 SSL 跟踪文件；请不要更改这些文件并将其发送给 IBM 支持人员。

如何启动和停止跟踪

使用 **strmqtrc** 控制命令启用或修改跟踪 (请参阅 [strmqtrc](#))。要停止跟踪，请使用 **endmqtrc** 控制命令 (请参阅 [endmqtrc](#))。

在 IBM MQ for Windows 系统中，还可以使用 IBM MQ Explorer 启动和停止跟踪，如下所示：

1. 从 **开始** 菜单启动 IBM MQ Explorer。
2. 在导航器视图中，右键单击 **IBM MQ** 树节点，然后选择 **跟踪...**。这会显示“跟踪”对话框。
3. 根据情况，单击 **启动** 或 **停止**。

选择性组件跟踪

使用 **-t** 和 **-x** 选项来控制要记录的跟踪详细信息量。缺省情况下，启用所有跟踪点。可以使用 **-x** 选项指定不希望跟踪的点。因此，例如，如果要仅跟踪流经通信网络的数据，请使用：

```
strmqtrc -x all -t comms
```

有关跟踪命令的详细信息，请参阅 [strmqtrc](#)。

选择性进程跟踪

使用 **strmqtrc** 命令控制的 **-p** 选项将跟踪生成限制到指定的命名进程。例如，要跟踪从名为 amqxxx.exe 的任何正在运行的进程产生的所有线程，请使用以下命令：

```
strmqtrc -p amqxxx.exe
```

有关跟踪命令的详细信息，请参阅 [strmqtrc](#)。

相关概念

第 305 页的『在 UNIX and Linux 系统上使用跟踪』

使用 **strmqtrc** 和 **endmqtrc** 命令来启动和结束跟踪，并使用 **dspmqtrc** 来显示跟踪文件

第 310 页的『将跟踪与 IBM i 上的 IBM MQ 服务器配合使用』

使用 TRCMQM 命令来启动和停止跟踪并指定所需的跟踪类型。

第 318 页的『在 z/OS 上使用跟踪进行问题确定』

有不同的跟踪选项可用于对 IBM MQ 进行问题确定。使用本主题来了解不同选项以及如何控制跟踪。

第 371 页的『跟踪 TLS: runmqakm、strmqikm 和 runmqckm 功能』

如何跟踪传输层安全性 (TLS) 以及请求 **runmqakm** 跟踪与 **strmqikm** (iKeyman) 和 **runmqckm** (iKeycmd) 跟踪。

第 347 页的『跟踪其他 IBM MQ Java 组件』

对于 IBM MQ 的 Java 组件，例如 IBM MQ Explorer 和 IBM MQ Transport for SOAP 的 Java 实现，诊断信息是使用标准 IBM MQ 诊断工具或由 Java 诊断类输出的。

Windows 跟踪数据示例

从 LTS 和 CD 发行版的 IBM MQ for Windows 跟踪文件中抽取。

LTS

LTS 发行版的样本 IBM MQ for Windows 跟踪数据:

| Counter | TimeStamp | PID.TID | Ident | Data |
|----------|-----------------|---------|-------|-----------------------------------|
| 00000EF7 | 16:18:56.381367 | 2512.1 | : | !! - Thread stack |
| 00000EF8 | 16:18:56.381406 | 2512.1 | : | !! - -> InitProcessInitialisation |

```

00000EF9 16:18:56.381429 2512.1 : --{ InitProcessInitialisation
00000EFA 16:18:56.381514 2512.1 : ---{ xcsReleaseThreadMutexSem
00000EFB 16:18:56.381529 2512.1 : ---{ xcsReleaseThreadMutexSem (rc=OK)
00000EFC 16:18:56.381540 2512.1 : ---{ xcsGetEnvironmentString
00000EFD 16:18:56.381574 2512.1 :
xcsGetEnvironmentString[AMQ_REUSE_SHARED_THREAD] = NULL
00000EFE 16:18:56.381587 2512.1 : ---{! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
00000EFF 16:18:56.381612 2512.1 : ---{ xcsGetEnvironmentInteger
00000F00 16:18:56.381622 2512.1 : ----{ xcsGetEnvironmentString
00000F01 16:18:56.381647 2512.1 : xcsGetEnvironmentString[AMQ_AFFINITY_MASK]
= NULL
00000F02 16:18:56.381660 2512.1 : ----{! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
00000F03 16:18:56.381673 2512.1 : ---{! xcsGetEnvironmentInteger
(rc=xecE_E_ENV_VAR_NOT_FOUND)
00000F04 16:18:56.381684 2512.1 : ---{ xcsGetEnvironmentString
00000F05 16:18:56.381708 2512.1 : xcsGetEnvironmentString[AMQ_FFSTINFO] = NULL
00000F06 16:18:56.381747 2512.1 : ---{! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
00000F07 16:18:56.381760 2512.1 : ---{ xcsIsEnvironment
00000F08 16:18:56.381783 2512.1 : xcsIsEnvironment[AMQ_DEBUG_MTIME] = FALSE
00000F09 16:18:56.381793 2512.1 : ---{ xcsIsEnvironment (rc=OK)
00000F0A 16:18:56.381804 2512.1 : ---{ xcsGetEnvironmentInteger
00000F0B 16:18:56.381811 2512.1 : ----{ xcsGetEnvironmentString
00000F0C 16:18:56.381835 2512.1 :
xcsGetEnvironmentString[AMQ_CBM_REUSE_FACTOR] = NULL
00000F0D 16:18:56.381848 2512.1 : ----{! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
00000F0E 16:18:56.381861 2512.1 : ---{! xcsGetEnvironmentInteger
(rc=xecE_E_ENV_VAR_NOT_FOUND)
00000F0F 16:18:56.381874 2512.1 : ---{ xcsGetEnvironmentInteger
00000F10 16:18:56.381885 2512.1 : ----{ xcsGetEnvironmentString
00000F11 16:18:56.381908 2512.1 :
xcsGetEnvironmentString[AMQ_CBM_MAX_CACHEABLE_SIZE] = NULL
00000F12 16:18:56.381919 2512.1 : ----{! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
00000F13 16:18:56.381929 2512.1 : ---{! xcsGetEnvironmentInteger
(rc=xecE_E_ENV_VAR_NOT_FOUND)
00000F14 16:18:56.381941 2512.1 : ---{ xcsGetEnvironmentInteger
00000F15 16:18:56.381952 2512.1 : ----{ xcsGetEnvironmentString
00000F16 16:18:56.381976 2512.1 : xcsGetEnvironmentString[AMQ_CBM_LEN] = NULL
00000F17 16:18:56.381992 2512.1 : ----{! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
00000F18 16:18:56.382003 2512.1 : ---{! xcsGetEnvironmentInteger
(rc=xecE_E_ENV_VAR_NOT_FOUND)
00000F19 16:18:56.382016 2512.1 : --{ InitProcessInitialisation (rc=OK)
00000F1A 16:18:56.383045 2512.1 : --{ DLLMain
00000F1B 16:18:56.383059 2512.1 : ---{ MCSInitCriticalSection
00000F1C 16:18:56.383068 2512.1 : ---{ MCSInitCriticalSection (rc=OK)

```

V 9.1.1 CD 发行版的样本 IBM MQ for Windows 跟踪数据:

```

TimeStamp      PID.TID      Ident      Data
=====
10:55:33.033870 4996.1      :      ---{ zutLookupInitialize
10:55:33.033877 4996.1      :      ----{ xcsCreateThreadMutexSem
10:55:33.033889 4996.1      :      hmtx: 000001DD32A9E0A0, created: TRUE
10:55:33.033896 4996.1      :      ----{ xcsCreateThreadMutexSem (rc=OK)
10:55:33.033903 4996.1      :      ----{ xcsGetMemFn
10:55:33.033911 4996.1      :      Data: 0x000001dd 0x32ab1b30
10:55:33.033923 4996.1      :      component:33 function:431 length:496 options:0
cbmindex:-1 *pointer:000001DD32AB1B30
10:55:33.033932 4996.1      :      ----{ xcsGetMemFn (rc=OK)
10:55:33.033985 4996.1      :      ---{ zutLookupInitialize (rc=OK)
10:55:33.034004 4996.1      :      ---{ xcsGetEnvironmentInteger
10:55:33.034012 4996.1      :      ----{ xcsGetEnvironmentString
10:55:33.034027 4996.1      :      xcsGetEnvironmentString[AMQ_BACKWARDS_TIME_LIMIT] =
NULL
10:55:33.034034 4996.1      :      ----{! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034065 4996.1      :      ---{! xcsGetEnvironmentInteger
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034073 4996.1      :      ---{ xcsReleaseThreadMutexSem
10:55:33.034078 4996.1      :      hmtx: 000001DD32A9DE90
10:55:33.034086 4996.1      :      ---{ xcsReleaseThreadMutexSem (rc=OK)
10:55:33.034089 4996.1      :      ----{ xcsGetEnvironmentString
10:55:33.034099 4996.1      :      xcsGetEnvironmentString[AMQ_REUSE_SHARED_THREAD] =
NULL

```

```

10:55:33.034106 4996.1 : ---}! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034114 4996.1 : ---{ xcsGetEnvironmentInteger
10:55:33.034118 4996.1 : ----{ xcsGetEnvironmentString
10:55:33.034124 4996.1 : xcsGetEnvironmentString[AMQ_AFFINITY_MASK] = NULL
10:55:33.034131 4996.1 : ----}! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034138 4996.1 : ---}! xcsGetEnvironmentInteger
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034146 4996.1 : ---{ xcsGetEnvironmentString
10:55:33.034153 4996.1 : xcsGetEnvironmentString[AMQ_FFSTINFO] = NULL
10:55:33.034160 4996.1 : ---}! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034168 4996.1 : ---{ xcsGetEnvironmentString
10:55:33.034176 4996.1 : xcsGetEnvironmentString[AMQ_CHECK_SEM_OBJECTS] = NULL
10:55:33.034183 4996.1 : ---}! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034191 4996.1 : ---{ xcsGetEnvironmentString
10:55:33.034199 4996.1 :
xcsGetEnvironmentString[AMQ_OVERRIDE_CONVERSION_TABLE] = NULL
10:55:33.034207 4996.1 : ---}! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034215 4996.1 : ---{ xcsGetEnvironmentString
10:55:33.034223 4996.1 : xcsGetEnvironmentString[AMQ_OVERRIDE_CCSID_TABLE] =
NULL
10:55:33.034230 4996.1 : ---}! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034237 4996.1 : ---{ xcsGetEnvironmentInteger
10:55:33.034241 4996.1 : ----{ xcsGetEnvironmentString
10:55:33.034248 4996.1 : xcsGetEnvironmentString[AMQ_CBM_REUSE_FACTOR] = NULL
10:55:33.034255 4996.1 : ----}! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034262 4996.1 : ---}! xcsGetEnvironmentInteger
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034270 4996.1 : ---{ xcsGetEnvironmentInteger
10:55:33.034274 4996.1 : ----{ xcsGetEnvironmentString
10:55:33.034282 4996.1 : xcsGetEnvironmentString[AMQ_CBM_MAX_CACHEABLE_SIZE]
= NULL
10:55:33.034289 4996.1 : ----}! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034296 4996.1 : ---}! xcsGetEnvironmentInteger
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034304 4996.1 : ---{ xcsGetEnvironmentInteger
10:55:33.034308 4996.1 : ----{ xcsGetEnvironmentString
10:55:33.034314 4996.1 : xcsGetEnvironmentString[AMQ_CBM_LEN] = NULL
10:55:33.034322 4996.1 : ----}! xcsGetEnvironmentString
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034330 4996.1 : ---}! xcsGetEnvironmentInteger
(rc=xecE_E_ENV_VAR_NOT_FOUND)
10:55:33.034337 4996.1 : --} InitProcessInitialisation (rc=OK)

```

z/OS 在 z/OS 上使用跟踪进行问题确定

有不同的跟踪选项可用于对 IBM MQ 进行问题确定。使用本主题来了解不同选项以及如何控制跟踪。

可用于 IBM MQ for z/OS 的跟踪功能包括：

- 用户参数（或 API）跟踪
- 支持中心使用的 IBM 内部跟踪
- 通道启动程序跟踪
- 线路跟踪

使用以下链接可了解如何收集和解释用户参数跟踪生成的数据，并描述如何生成 IBM 内部跟踪以供 IBM 支持中心使用。此外，还有关于可用于 IBM MQ 的其他跟踪功能的信息。

- [控制 z/OS 系统的 GTF](#)
- [控制要为其收集数据的每个队列管理器子系统的 IBM MQ 跟踪](#)
- [第 321 页的『在 z/OS 上格式化并识别控制块信息』](#)
- [第 322 页的『在 z/OS 上解释跟踪信息』](#)

如果未产生跟踪数据，请检查以下事项：

- 是否正确启动了 GTF，从而在 USRP 选项上指定 EID 5E9、5EA 和 5EE？

- 是否正确输入了 START TRACE(GLOBAL) 命令，并且指定了相关类？

有关 z/OS 上可用的其他跟踪选项的更多信息，请参阅第 324 页的『z/OS 上其他类型的跟踪』。

相关概念

第 305 页的『在 UNIX and Linux 系统上使用跟踪』

使用 **strmqtrc** 和 **endmqtrc** 命令来启动和结束跟踪，并使用 **dspmqtrc** 来显示跟踪文件

第 310 页的『将跟踪与 IBM i 上的 IBM MQ 服务器配合使用』

使用 TRCMQM 命令来启动和停止跟踪并指定所需的跟踪类型。

第 371 页的『跟踪 TLS: runmqakm、strmqikm 和 runmqckm 功能』

如何跟踪传输层安全性 (TLS) 以及请求 **runmqakm** 跟踪与 **strmqikm** (iKeyman) 和 **runmqckm** (iKeycmd) 跟踪。

第 347 页的『跟踪其他 IBM MQ Java 组件』

对于 IBM MQ 的 Java 组件，例如 IBM MQ Explorer 和 IBM MQ Transport for SOAP 的 Java 实现，诊断信息是使用标准 IBM MQ 诊断工具或由 Java 诊断类输出的。

相关参考

第 315 页的『使用 Windows 上的跟踪』

使用 **strmqtrc** 和 **endmqtrc** 命令或 IBM MQ Explorer 界面来启动和结束跟踪。

z/OS MQI 调用和用户参数以及 z/OS 上的 GTF

使用本主题来了解如何控制 z/OS 通用跟踪设施 (GTF) 和 IBM MQ 跟踪。

您可以在进入或退出 IBM MQ 时，获取有关 MQI 调用以及某些 IBM MQ 调用所传递用户参数的信息。为此，请将全局跟踪与 z/OS 通用跟踪设施 (GTF) 结合使用。

z/OS 启动和停止 GTF

在 z/OS 上，可以使用通用跟踪工具 (GTF) 来记录和诊断系统和程序问题。

关于此任务

您可以在进入或退出 IBM MQ 时，获取有关 MQI 调用以及某些 IBM MQ 调用所传递用户参数的信息。为此，请将全局跟踪与 z/OS 通用跟踪设施 (GTF) 结合使用。

过程

- 通过输入 **START GTF** 命令在控制台上启动 GTF。

启动 GTF 时，请指定 USRP 选项。系统会提示您输入事件标识 (EID) 的列表。IBM MQ 使用的 EID 为：

5E9

在进入 IBM MQ 时收集有关控制块的信息

5EA

在退出 IBM MQ 时收集有关控制块的信息

有时，如果发生您无法自行解决的错误，那么 IBM 支持中心可能会要求您提供其他内部跟踪信息来对错误进行分析。跟踪的其他类型为：

5EE

收集 IBM MQ 的内部信息

您也可以使用 JOBNAMEP 选项，从而指定批处理、CICS、IMS 或 TSO 作业名，以将跟踪输出限制到特定作业。以下示例显示了 GTF 的样本启动，指定了四个 EID 和一个作业名。以 **粗体** 显示的行是您在控制台上输入的命令；其他行是提示和响应。有关启动 GTF 跟踪的更多信息，请参阅 [启动 GTF](#)。

```
START GTFxx.yy
#HASP100 GTFxx.yy ON STCINRDR
#HASP373 GTFxx.yy STARTED
*01 AHL100A SPECIFY TRACE OPTIONS
R 01, TRACE=JOBNAMEP, USRP
TRACE=JOBNAMEP, USRP
```

```

IEE600I REPLY TO 01 IS;TRACE=JOBNAMEP,USRP
*02 ALH101A SPECIFY TRACE EVENT KEYWORDS - JOBNAME=,USR=
R 02, JOBNAME=(xxxxMSTR,xxxxCHIN,zzzzzzzz),USR=(5E9,5EA,5EE)
JOBNAME=(xxxxMSTR,xxxxCHIN,zzzzzzzz),USR=(5E9,5EA,5EE)
IEE600I REPLY TO 02 IS;JOBNAME=(xxxxMSTR,xxxxCHIN,zzzzzzzz),USR=(5E9,5EA,5EE)
*03 ALH102A CONTINUE TRACE DEFINITION OR REPLY END
R 03,END
END
IEE600I REPLY TO 03 IS;END
AHL103I TRACE OPTIONS SELECTED-USR=(5E9,5EA,5EE)
AHL103I JOBNAME=(xxxxMSTR,xxxxCHIN,zzzzzzzz)
*04 AHL125A RESPECIFY TRACE OPTIONS OR REPLY U
R 04,U
U
IEE600I REPLY TO 04 IS;U
AHL031I GTF INITIALIZATION COMPLETE

```

其中:

- xx 是要使用的 GTF 过程的名称 (可选)
- yy 是此 GTF 跟踪实例的标识
- xxxx 是队列管理器的名称
- zzzzzzzz 是批处理作业或 CICS 区域名称

最多可以列出 5 个作业名。

使用 GTF 时, 请指定主作业名 (CHINIT, CICS 或 batch) 以及队列管理器名 (xxxxMSTR)。

- 在控制台停止 GTF。
当您输入 GTF 的 stop 命令时, 请包括您在启动时使用的其他标识 (yy), 如以下示例中所示:

```
STOP yy
```

相关信息

在 IBM z/OS 上生成 IBM MQ GTF 跟踪

控制 *IBM MQ for z/OS* 中的跟踪

IBM MQ for z/OS 跟踪使用 MQSC 命令进行控制。使用本主题来了解如何控制跟踪, 以及输出的跟踪信息的类型。

使用 START TRACE 命令, 指定类型 GLOBAL 以开始将 IBM MQ 记录写入到 GTF。您还必须指定 dest(GTF), 例如在以下命令中:

```
/cpf start trace(G)class(2,3)dest(GTF)
```

要定义您想要为其产生跟踪数据的事件, 请使用以下一个或多个类:

| CLASS | 跟踪的事件 |
|-------|--|
| 2 | 在检测到除 MQRC_NONE 以外的其他完成代码时, 记录 MQI 调用和 MQI 参数。 |
| 3 | 在进入或退出队列管理器时, 记录 MQI 调用和 MQI 参数。 |

在跟踪启动后, 可以使用以下命令显示有关跟踪的信息, 修改跟踪的属性以及停止跟踪:

- DISPLAY TRACE
- ALTER TRACE
- STOP TRACE

要使用任何跟踪命令, 必须具有以下权限之一:

- 用于发出启动和停止跟踪命令的权限 (跟踪权限)
- 用于发出显示跟踪命令的权限 (显示权限)

注:

1. 也可以通过初始化输入数据集输入跟踪命令。
 2. 产生的跟踪信息还将包括同步点流（例如 PREPARE 和 COMMIT）的详细信息。
- 有关这些命令的信息，请参阅 [MQSC 命令](#)。

在 z/OS 上格式化并识别控制块信息

在捕获跟踪后，必须格式化输出并识别 IBM MQ 控制块。

- [格式化信息](#)
- [识别与 IBM MQ 关联的控制块](#)
- [识别与控制块关联的事件标识](#)

格式化信息

要格式化由全局跟踪收集的用户参数数据，请使用 [第 321 页的图 59](#) 中显示的批处理作业或 IPCS GTFTRACE USR(*xxx*) 命令，其中 *xxx* 为:

5E9

用于格式化有关进入 IBM MQ MQI 调用时的控制块的信息。

5EA

用于格式化有关退出 IBM MQ MQI 调用时的控制块的信息。

5EE

用于格式化有关 IBM MQ 内部项的信息。

您还可以指定 **JOBNAME**(*jobname*) 参数以将格式化输出限制到特定作业。

```
//S1 EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=4096K
//IPCSPARM DD DSN=SYS1.PARMLIB,DISP=SHR
//IPCSDDIR DD DSN=thlqual.ipcs.dataset.directory,DISP=SHR
//SYSTSPRT DD SYSOUT=*,DCB=(LRECL=137)
//IPCSTOC DD SYSOUT=*
//GTFFIN DD DSN=gtf.trace,DISP=SHR
//SYSTSIN DD *
IPCS
SETDEF FILE(GTFFIN) NOCONFIRM
GTFTRACE USR(5E9,5EA,5EE)
/*
//STEPLIB DD DSN=thlqual.SCSQAUTH,DISP=SHR
```

图 59: 在批处理中格式化 GTF 输出

识别与 IBM MQ 关联的控制块

IBM MQ 跟踪的格式标识为 D9。该值显示在格式化 GTF 输出中每个格式化控制块的开头，形式为:

```
USRD9
```

识别与控制块关联的事件标识

跟踪格式化程序在每个控制块开头插入以下某条消息。这些消息指示是在进入还是退出 IBM MQ 时捕获数据:

- CSQW072I ENTRY: MQ 用户参数跟踪
- CSQW073I EXIT: MQ 用户参数跟踪

相关任务

[第 319 页的『启动和停止 GTF』](#)

在 z/OS 上，可以使用通用跟踪工具 (GTF) 来记录 and 诊断系统和程序问题。

z/OS 在 z/OS 上解释跟踪信息

可以检查 IBM MQ 产生的 GTFTRACE，以确定地址无效、控制块无效和数据无效等可能的错误。

启动 GTFTRACE 子命令 以格式化包含在转储或跟踪数据集中的通用跟踪设施 (GTF) 记录。有关 GTF 的更多信息，请参阅第 319 页的『启动和停止 GTF』。

查看 GTFTRACE 命令产生的数据时，请考虑以下要点：

- 如果控制块完全由零组成，那么在从用户的地址空间复制数据时可能会发生错误。这可能是因为传递的地址无效。
- 如果控制块的第一个部分包含非空数据，但是其余部分由零组成，那么在从用户的地址空间复制数据时可能再次发生错误，例如，控制块未完全放在有效的存储器中。这也可能是由于控制块未正确初始化。
- 如果在退出 IBM MQ 时发生错误，那么可能是由于 IBM MQ 未将数据写入到用户的地址空间。所显示的数据是其尝试复制到用户的地址空间的版本。

下表显示被跟踪的控制块的详细信息。

第 322 页的表 27 说明针对不同 MQI 调用跟踪的控制块。

| MQI 调用 | 条目 | 退出 |
|---------|--|--|
| MQCB | MQCBD、MQMD、MQGMO | MQCBD、MQMD、MQGMO |
| MQCLOSE | 无 | 无 |
| MQGET | MQMD、MQGMO | MQMD、MQGMO 和消息数据的前 256 字节 |
| MQINQ | 选择器 (如果 <i>SelectorCount</i> 大于 0) | 选择器 (如果 <i>SelectorCount</i> 大于 0) 整数属性 (如果 <i>IntAttrCount</i> 大于 0) 字符属性 (如果 <i>CharAttrLength</i> 大于 0) |
| MQOPEN | MQOD | MQOD |
| MQPUT | MQMD、MQPMO 和消息数据的前 256 字节 | MQMD、MQPMO 和消息数据的前 256 字节 |
| MQPUT1 | MQMD、MQOD、MQPMO 和消息数据的前 256 字节 | MQMD、MQOD、MQPMO 和消息数据的前 256 字节 |
| MQSET | 选择器 (如果 <i>SelectorCount</i> 大于 0) 整数属性 (如果 <i>IntAttrCount</i> 大于 0) 字符属性 (如果 <i>CharAttrLength</i> 大于 0) | 选择器 (如果 <i>SelectorCount</i> 大于 0) 整数属性 (如果 <i>IntAttrCount</i> 大于 0) 字符属性 (如果 <i>CharAttrLength</i> 大于 0) |
| MQSTAT | MQSTS | MQSTS |
| MQSUB | MQSD、MQSD.ObjectString、 MQSD.SubName、 MQSD.SubUserData、 MQSD.SelectionString、 MQSD.ResObjectString | MQSD、MQSD.ObjectString、 MQSD.SubName、 MQSD.SubUserData、 MQSD.SelectionString、 MQSD.ResObjectString |
| MQSUBRQ | MQSRO | MQSRO |

注: 在 MQGET 调用带有 WAIT 选项的特殊情况下, 如果在 MQGET 请求时没有可用消息, 但是消息后来在所指定的任何时间间隔到期之前变为可用, 那么会显示双重入口。

这是因为, 尽管应用程序已发出单个 MQGET 调用, 但是适配器正在代表应用程序执行等待, 并且在消息变为可用时, 它会重新发出该调用。因此在跟踪中, 它显示为第二个 MQGET 调用。


在某些情况下, 还会产生有关队列请求参数列表的特定字段的信息。此列表中的字段标识如下:

| 标识 | 描述 |
|----------|------------------------------------|
| 操作 | 所请求的操作 |
| BufferL | 缓冲区长度 |
| CBD | 回调描述符的地址 |
| CompCode | 完成代码 |
| CharAttL | 字符属性长度 |
| DataL | 数据长度 |
| Hobj | 对象句柄 |
| Hsub | 预订句柄 |
| IntAttC | 整数属性的计数 |
| pObjDesc | 对象描述符 |
| Oper | 操作 |
| 选项 | 选项 |
| pBuffer | 缓冲区的地址 |
| pCharAtt | 字符属性的地址 |
| pCTLO | 控制回调选项的地址 |
| pECB | get 中使用的 ECB 的地址 |
| pGMO | get 消息选项的地址 |
| pIntAtt | 整数属性的地址 |
| pMsgDesc | 消息描述符的地址 |
| pPMO | put 消息选项的地址 |
| pSD | 预订描述符的地址 |
| pSelect | 选择器的地址 |
| pSRQOpt | 预订请求选项的地址 |
| pSTS | 状态结构的地址 |
| 原因 | 原因码 |
| RSVn | 为 IBM 保留 |
| SelectC | 选择器计数 |
| 线程 | 线程 |
| 类型 | 所请求的类型 |
| UOWInfo | 有关工作单元的信息 |
| UserID | CICS 或 IMS 用户标识, 对于批处理或 TSO, 此标识为零 |

z/OS z/OS 上其他类型的跟踪

有其他跟踪功能可用于问题确定。使用本主题来调查通道启动程序跟踪、线路跟踪、CICS 适配器跟踪、SSL 跟踪和 z/OS 跟踪。

将以下跟踪功能与 IBM MQ 结合使用可能有所帮助。

- [通道启动程序跟踪](#)
- [线路跟踪](#)
- [CICS 适配器跟踪](#)
- [系统 SSL 跟踪](#)
-  [z/OS 跟踪](#)

通道启动程序跟踪

请参阅第 200 页的图 43 以获取有关如何获取通道启动程序地址空间转储的信息。请注意，通道启动程序产生的转储不包括跟踪数据空间。跟踪数据空间（称为 CSQXTRDS）包含跟踪信息。您可以通过在 SLIP 陷阱上指定跟踪数据空间或者在使用转储命令时对此进行请求。

可以使用 START TRACE 命令运行跟踪，也可以使用 TRAXSTR 队列管理器属性将此跟踪设置为自动启动。有关如何执行此操作的更多信息，请参阅 [ALTER QMGR](#)。

可以通过输入 IPCS 命令来显示此跟踪信息：

```
LIST 1000. DSPNAME(CSQXTRDS)
```

可以使用以下命令将其格式化：

```
CTRACE COMP(CSQX $ssnm$ )
```

其中， $ssnm$ 是子系统名称。

线路跟踪

对于每条通道存在回绕线路跟踪。针对通道启动程序地址空间中的每条通道，此跟踪保留在 4 KB 缓冲区内。系统为每条通道都产生跟踪，因此它对于解决通道看似处于挂起状态的问题很理想，因为可以在正常跟踪回绕很久之后才收集有关此通道的活动信息。

线路跟踪始终处于活动状态；您无法将其关闭。它对于 LU 6.2 和 TCP 通道均可用，并且应会减少需要通信跟踪的次数。

可以将该跟踪视为写入到 CSQSNAP 的未格式化跟踪。可以通过遵循以下步骤来显示跟踪：

1. 确保 CHIN 过程具有 SNAP DD 语句。
2. 启动 CHIN 跟踪，按如下指定 IFCID 202：

```
START TRACE(CHINIT) CLASS(4) IFCID(202)
```

3. 显示需要线路跟踪的通道状态：

```
DISPLAY CHSTATUS(channel) SAVED
```

这会将所选通道的当前线路转储到 CSQSNAP。请参阅第 214 页的『[z/OS 上的快速转储](#)』，以了解更多信息。

注：

- a. 存储器转储的地址不正确，因为 CSQXFFST 机制在将存储器写入到 CSQSNAP 之前会对其生成副本。
- b. 仅在首次运行 DISPLAY CHSTATUS SAVED 命令时会产生到 CSQSNAP 的转储。这旨在防止每次运行该命令都获取转储。

要获取线路跟踪数据的其他转储，必须停止并重新启动当前跟踪。

- i) 可以使用选择性 STOP TRACE 命令来只停止已启动的跟踪，以收集线路跟踪数据。为此，请注意分配给跟踪的 TRACE NUMBER，如下例所示：

```
+ssid START TRACE(CHINIT) CLASS(4) IFCID(202)
      CSQW130I +ssid 'CHINIT' TRACE STARTED, ASSIGNED TRACE NUMBER 01
```

- ii) 要停止跟踪，请发出以下命令：

```
+ssid STOP TRACE(CHINIT) TNO(01)
```

- iii) 然后，可以随 DISPLAY CHSTATUS SAVED 命令输入另一个 START TRACE 命令，以将更多线路跟踪数据收集到 CSQSNAP。

4. 线路跟踪缓冲区未格式化。每个条目以时钟开头，后跟时间戳记，然后指示这是 OUTBOUND 还是 INBOUND 流。使用时间戳记信息查找最早的条目。

CICS 适配器跟踪

如果跟踪号设置为范围在 0 到 199（十进制）内的值，并且满足以下条件之一，那么 CICS 适配器将条目写入到 CICS 跟踪：

- CICS 用户跟踪已启用，或者
- CICS 内部/辅助跟踪已启用

可以通过以下两种方式之一启用 CICS 跟踪：

- 动态，使用 CICS 提供的事务 CETR
- 通过确保 CICS 系统初始化表 (SIT) 中的 USERTR 参数设置为 YES

有关启用 CICS 跟踪的更多信息，请参阅 *CICS Problem Determination Guide*。

源自 CICS 适配器的 CICS 跟踪条目的值为 AP0 000，其中 000 是与所指定的 CICS 适配器跟踪号的十进制值等效的十六进制数。

跟踪条目显示在 [第 325 页的『CICS 适配器跟踪条目』](#)中。

系统 SSL 跟踪

您可以使用 SSL 启动的任务来收集系统 SSL 跟踪。有关如何设置此任务的详细信息可在 *System Secure Sockets Layer Programming* 文档 (SC24-5901) 中获取。对于在 CHINIT 地址空间中运行的每个 SSLTASK 都会生成跟踪文件。

z/OS 跟踪



z/OS 跟踪（对于作为 z/OS 的正式子系统运行的所有产品通用）可与 IBM MQ 结合使用。有关使用和解释此跟踪工具的信息，请参阅 [z/OS MVS Diagnosis: Tools and Service Aids](#) 手册。

CICS 适配器跟踪条目

使用本主题作为 CICS 适配器跟踪条目的参考。

这些值的 CICS 跟踪条目为 AP0 xxx（其中 xxx 是与启用 CICS 适配器时指定的跟踪号等效的十六进制数）。这些跟踪条目全都由 CSQCTRUE 发出（CSQCTEST 除外，它由 CSQCRST 和 CSQCDSP 发出）。

| 表 28: CICS 适配器跟踪条目 | | | |
|--------------------|-----------|--|--|
| 名称 | 描述 | 跟踪顺序 | 跟踪数据 |
| CSQCABNT | 异常终止 | 在向 IBM MQ 发出 END_THREAD ABNORMAL 之前。这是由于任务结束，因此应用程序可能会执行隐式回退。在此情况下，END_THREAD 调用中包含 ROLLBACK 请求。 | 工作单元信息。在了解工作状态时，可以使用此信息。（例如，可以根据 DISPLAY THREAD 命令产生的输出或日志列显实用程序对其进行验证。） |
| CSQCAUID | 网桥安全性 | 在验证网桥用户密码或通行票之前。 | 用户标识。 |
| CSQCBACK | 同步点回退 | 在向 IBM MQ 发出 BACKOUT 之前。这是由于来自应用程序的显式回退请求。 | 工作单元信息。 |
| CSQC CONX | MQCONN | 在向 IBM MQ 发出 MQCONN 之前。 | 连接标记。 |
| CSQCCRC | 完成代码和原因码 | 在从 API 调用未成功返回之后。 | 完成代码和原因码。 |
| CSQCCOMM | 同步点落实 | 在向 IBM MQ 发出 COMMIT 之前。这可能是由于单阶段落实请求或两阶段落实请求的第二阶段。该请求是由于来自应用程序的显式同步点请求。 | 工作单元信息。 |
| CSQDCFF | 仅供 IBM 使用 | | |
| CSQDCIN | 仅供 IBM 使用 | | |
| CSQDCOT | 仅供 IBM 使用 | | |
| CSQCEXER | 执行解析 | 在向 IBM MQ 发出 EXECUTE_RESOLVE 之前。 | 发出 EXECUTE_RESOLVE 的工作单元的工作单元信息。这是再同步过程中的最后一个不确定的工作单元。 |
| CSQCGETW | GET 等待 | 在发出 CICS 等待之前。 | 要在其上进行等待的 ECB 的地址。 |
| CSQCGMGD | GET 消息数据 | 在从 MQGET 成功返回之后。 | 最多 40 字节的消息数据。 |
| CSQCGMGH | GET 消息句柄 | 在向 IBM MQ 发出 MQGET 之前。 | 对象句柄。 |
| CSQCGMGI | 获取消息标识 | 在从 MQGET 成功返回之后。 | 消息的消息标识和相关标识。 |
| CSQCHCER | Hconn 错误 | 在发出任何 MQ 动词之前。 | 连接句柄。 |
| CSQCINDL | 不确定的列表 | 在从第二个 INQUIRE_INDOUBT 成功返回之后。 | 不确定的工作单元列表。 |
| CSQCINDO | 仅供 IBM 使用 | | |
| CSQCINDS | 不确定的列表大小 | 在从第一个 INQUIRE_INDOUBT 成功返回之后并且不确定的列表非空。 | 列表的长度；除以 64 即得出不确定的工作单元的数量。 |
| CSQCINDW | 不确定的同步点 | 在同步点处理期间，CICS 不能确定工作单元的处置。 | 工作单元信息。 |
| CSQCINQH | INQ 句柄 | 在向 IBM MQ 发出 MQINQ 之前。 | 对象句柄。 |
| CSQCLOSH | CLOSE 句柄 | 在向 IBM MQ 发出 MQCLOSE 之前。 | 对象句柄。 |
| CSQCLOST | 处置丢失 | 在再同步过程中，CICS 通知适配器它已冷启动，因此没有关于进行再同步的工作单元是否可用的处置信息。 | CICS 已知的进行再同步的工作单元的工作单元标识。 |

| 表 28: CICS 适配器跟踪条目 (继续) | | | |
|-------------------------|-------------|---|---|
| 名称 | 描述 | 跟踪顺序 | 跟踪数据 |
| CSQCNIND | 处置可确定 | 在再同步过程中, CICS 通知适配器进行再同步的工作单元不应处于不确定状态 (即, 它可能仍在运行)。 | CICS 已知的进行再同步的工作单元的工作单元标识。 |
| CSQCNORT | 正常终止 | 在向 IBM MQ 发出 END_THREAD NORMAL 之前。这是由于任务结束, 因此应用程序可能会执行隐式同步点落实。在此情况下, END_THREAD 调用中包含 COMMIT 请求。 | 工作单元信息。 |
| CSQCOPNH | OPEN 句柄 | 在从 MQOPEN 成功返回之后。 | 对象句柄。 |
| CSQCOPNO | OPEN 对象 | 在向 IBM MQ 发出 MQOPEN 之前。 | 对象名称。 |
| CSQCPMGD | PUT 消息数据 | 在向 IBM MQ 发出 MQPUT 之前。 | 最多 40 字节的消息数据。 |
| CSQCPMGH | PUT 消息句柄 | 在向 IBM MQ 发出 MQPUT 之前。 | 对象句柄。 |
| CSQCPMGI | PUT 消息标识 | 从 IBM MQ 成功 MQPUT 后。 | 消息的消息标识和相关标识。 |
| CSQCPREP | 同步点准备 | 在两阶段落实处理的第一阶段中向 IBM MQ 发出 PREPARE 之前。也可以从分布式排队组件将此调用作为 API 调用发出。 | 工作单元信息。 |
| CSQCP1MD | PUTONE 消息数据 | 在向 IBM MQ 发出 MQPUT1 之前。 | 最多 40 字节的消息数据。 |
| CSQCP1MI | PUTONE 消息标识 | 在从 MQPUT1 成功返回之后。 | 消息的消息标识和相关标识。 |
| CSQCP1ON | PUTONE 对象名称 | 在向 IBM MQ 发出 MQPUT1 之前。 | 对象名称。 |
| CSQCRBAK | 已解析的回退 | 在向 IBM MQ 发出 RESOLVE_ROLLBACK 之前。 | 工作单元信息。 |
| CSQCRMT | 已解析的落实 | 在向 IBM MQ 发出 RESOLVE_COMMIT 之前。 | 工作单元信息。 |
| CSQCRMIR | RMI 响应 | 在从特定调用返回到 CICS RMI (资源管理器接口) 之前。 | 架构式 RMI 响应值。其含义取决于调用的类型。要确定调用的类型, 请查看 CICS RMI 组件产生的先前跟踪条目。 |
| CSQCRSYN | 再同步 | 在对任务启动再同步过程之前。 | CICS 已知的进行再同步的工作单元的工作单元标识。 |
| CSQCSETH | SET 句柄 | 在向 IBM MQ 发出 MQSET 之前。 | 对象句柄。 |
| CSQCTASE | 仅供 IBM 使用 | | |
| CSQCTEST | 跟踪测试 | 在 EXEC CICS ENTER TRACE 调用中用于验证用户提供的跟踪号和连接的跟踪状态。 | 无数据。 |

为 AMSM 系统启用内部跟踪

可以使用 `_AMS_MSG_LEVEL` 变量启用 AMSM 地址空间跟踪, 该变量会通过 ENVARS DD 卡传递到 AMSM 地址空间中。

ENVARS DD 卡的样本数据集位于 th1qua1.SCSQPROC (CSQ40ENV) 中。

跟踪信息将写入 AMSM 地址空间的 SYSOUT。

`_AMS_MSG_LEVEL` 变量指定要记录的子组件和消息级别。星号指示要记录的所有子组件；当前只有一个子组件。

严重性级别为：

- S - 仅严重消息
- E - 仅错误和严重消息
- W - 仅警告、错误和严重消息
- I - 参考、警告、错误和严重消息。这是缺省值
- D - 调试方式，包含其他调试诊断的所有消息
- V - 详细方式，所有上述内容以及缓冲区转储



注意：只应按照 IBM 服务代表的建议启用调试或详细方式。

例如，要启用 `_AMS_MSG_LEVEL` 的缺省值，请发出以下命令：

```
_AMS_MSG_LEVEL=* .i
```

要启用详细方式，请发出以下命令：

```
_AMS_MSG_LEVEL=* .v
```

在 z/OS 上使用 AMS 时，对与证书和密钥相关的问题使用 GSKit 跟踪

使用本主题来了解在使用 AMS on z/OS 时如何开启和关闭 GSKit 跟踪。

介绍

在 AMS 地址空间的 *AMSD 过程中，以及在运行程序 `csq0util` 的样本作业 `CSQ40CFG` 中，有一个可用于设置环境变量的 ENVARS DD 卡。提供了名为 `CSQ40ENV` 的样本 AMS 环境变量文件，其中包含有关如何打开和关闭 GSKit 跟踪的详细信息。可以在 IBM MQ `h1q.SCSQPROC` PDS 库中找到样本。

如果在 *AMSD 过程的 ENVARS DD 卡中设置 GSK 跟踪环境变量，那么将从启动 AMS 地址空间的位置（即，在配置了 AMS 的情况下作为队列管理器启动的一部分）设置变量。变量可以开启或关闭对 AMS 地址空间发出的所有 `gsk_*` 调用的跟踪。

如果在 `csq40cfg` 作业的 ENVARS DD 卡中设置 GSK 跟踪环境变量，那么将在 `csq40cfg` 作业的持续时间内设置变量。变量可以开启或关闭在处理 AMS 命令期间发出的所有 `gsk_*` 调用的跟踪，例如定义和显示 AMS 策略。

打开 GSKit 跟踪

AMS 地址空间上的 GSKit

要对 AMS 地址空间开启 GSKit 跟踪，请执行以下过程：

1. 使用以下内容定义 `csq40env` 文件：

```
GSK_TRACE_FILE=/u/<username>/AMStrace/gsktrace/gskssl.%.trc  
GSK_TRACE=0x1f
```

在 AMS 地址空间的 *AMSD 过程中的 ENVARS DD 卡上。例如：

```
//ENVARS DD DSN=USERID.JCL(CSQ40ENV),DISP=SHR
```

2. 启动队列管理器，通道启动程序和 AMS 地址空间。

您将在作业记录中看到 AMS 地址空间的环境变量设置。例如：

```
-4.09.18 STC13921 CSQ06091 !MQ07 CSQ0DSRV IBM MQ AMS for z/OS starting V9.2.3, level GA
-4.09.18 STC13921 CSQ06191 !MQ07 CSQ0DSRV AMSPROD=ADVANCEDVUE, recording product usage for MQ z/OS Adv
VUE product id 5555AV9
-4.09.18 STC13921 CSQ06331 !MQ07 CSQ0DSRV AMS environment variables values:
-4.09.18 STC13921 CSQ06341 !MQ07 CSQ0DSRV _CEE_ENVFILE_S=DD:ENVVARS
-4.09.18 STC13921 CSQ06341 !MQ07 CSQ0DSRV _AMS_MSG_LEVEL=*V
-4.09.18 STC13921 CSQ06341 !MQ07 CSQ0DSRV _AMS_MSG_FOLDING=NO
-4.09.18 STC13921 CSQ06341 !MQ07 CSQ0DSRV _AMS_INIT_THREADS=20
-4.09.18 STC13921 CSQ06341 !MQ07 CSQ0DSRV _AMS_MAX_THREADS=100
-4.09.18 STC13921 CSQ06341 !MQ07 CSQ0DSRV TZ=ESTESDT
-4.09.18 STC13921 CSQ06341 !MQ07 CSQ0DSRV GSK_TRACE_FILE=/u/<username>/AMStrace/gsktrace/gskssl.%.trc
-4.09.18 STC13921 CSQ06341 !MQ07 CSQ0DSRV GSK_TRACE=0xff
-4.09.21 STC13921 CSQ06531 !MQ07 CSQ0DLCL CRL checking disabled
-4.09.21 STC13921 CSQ06021 !MQ07 CSQ0DCNS AMS initialization complete
```

将跟踪 AMS 地址空间发出的 `gsk_*` 调用，以分别在放置和获取时间保护或取消保护 IBM MQ 消息。启动 AMS 地址空间时将创建跟踪文件，以跟踪随后由该地址空间执行的所有 `gsk_*` 调用。在跟踪文件的名称中使用 % 字符可确保跟踪文件由 Unix 系统服务 (USS) 进程标识命名。

3. 发出以下命令以列出生成的跟踪文件：

```
/u/<username>/AMStrace/gsktrace:>ls
```

例如，您会看到类似如下的文件：

```
gskssl.84017302.trc
```

4. 要格式化和查看跟踪文件，请在 USS 中发出以下命令：

```
/u/<username>/AMStrace/gsktrace:>gsktrace gskssl.84017302.trc
```

生成类似于以下内容的输出：

```
07/01/2022-10:36:41 Thd-0 INFO gsk_svc_init(): System SSL Version 4, Release 4, Service level OA60573
07/01/2022-10:36:41 Thd-0 INFO gsk_svc_init(): LE runtime level 0x42040000, 31-bit addressing mode
07/01/2022-10:36:41 Thd-0 INFO gsk_svc_init(): STDOUT handle=-1, STDERR handle=-1, TRACE handle=0
07/01/2022-10:36:41 Thd-0 INFO gsk_dll_init_once(): Using variant character table for code set IBM-1047
07/01/2022-10:36:41 Thd-0 INFO gsk_dll_init_once(): Using local code page IBM-1047
07/01/2022-10:36:41 Thd-0 INFO gsk_dll_init_once(): Using ISO8859-1 for TELETEX string
07/01/2022-10:36:41 Thd-0 INFO gsk_dll_init_once(): 64-bit encryption enabled
07/01/2022-10:36:41 Thd-0 INFO gsk_dll_init_once(): 128-bit encryption enabled
07/01/2022-10:36:41 Thd-0 INFO gsk_dll_init_once(): 168-bit encryption enabled
07/01/2022-10:36:41 Thd-0 INFO gsk_dll_init_once(): 256-bit encryption enabled
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): Crypto assist supports strong encryption
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): FIPS mode level 1101
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): SHA-1 crypto assist is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): SHA-224 crypto assist is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): SHA-256 crypto assist is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): SHA-384 crypto assist is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): SHA-512 crypto assist is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): DES crypto assist is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): DES3 crypto assist is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): AES 128-bit crypto assist is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): AES 256-bit crypto assist is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): AES-GCM crypto assist is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): Cryptographic accelerator is not available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): Cryptographic coprocessor is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): Public key hardware support is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): Max RSA key sizes in hardware - signature 4096, encryption
4096, verification 4096
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): Maximum RSA token size 3500
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): ECC clear key support is available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): ECC secure key support is available. Maximum key size 521
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): TKDS is available for the storage of persistent PKCS #11
objects
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): ICSF Secure key PKCS #11 support is not available
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): ICSF FIPS compatibility mode
07/01/2022-10:36:41 Thd-0 INFO crypto_init(): ICSF FMID is HCR77D1
07/01/2022-10:36:41 Thd-0 INFO gsk_dll_init_once(): Job name CSQ40CFG, Process 05020096
07/01/2022-10:36:41 Thd-0 INFO gsk_dll_init_once(): GSKSRVR communication area at 00000000
07/01/2022-10:36:41 Thd-0 ENTRY gsk_dn_to_name(): ---> DN: CN=USER,0=IBM,C=UK
07/01/2022-10:36:41 Thd-0 EXIT gsk_dn_to_name(): <--- Exit status 0x00000000 (0)
07/01/2022-10:36:46 Thd-0 ENTRY gsk_dn_to_name(): ---> DN: CN=USER1,0=IBM,C=UK
07/01/2022-10:36:46 Thd-0 EXIT gsk_dn_to_name(): <--- Exit status 0x00000000 (0)
```

```
07/01/2022-10:36:46 Thd-0 ENTRY gsk_dn_to_name(): ---> DN: CN=USER,O=IBM,C=UK
07/01/2022-10:36:46 Thd-0 EXIT gsk_dn_to_name(): <--- Exit status 0x00000000 (0)
```

csq40cfg 作业的 GSKit

要对 csq40cfg 作业开启 GSKit 跟踪，请设置 ENVARS DD 卡，如以下示例中所示：

```
//CSQ40CFG JOB (ACCOUNT),'DEFAULT JOBCARD',CLASS=C,
//          MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//* Job to define and display an AMS policy on a queue. The policy
//* name is the same as the queue name.
//* Make sure column numbers are not included as otherwise they can
//* interfere with the data in SYSIN.
/*JOBPARM SYSAFF=MVnn
//CSQ40CFG EXEC PGM=CSQ0UTIL,
//          PARM='ENVAR("_CEE_ENVFILE_S=DD:ENVARS")' /'
//STEPLIB DD DSN=hlq.SCSQANLE,DISP=SHR
//          DD DSN=hlq.SCSQAUTH,DISP=SHR
//ENVARS DD DSN=USERID.JCL(CSQ40ENV),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
setmqsp1 -m MQ01 -p BANK.RQ
          -r CN=USERID,O=IBM,C=UK -e AES256
dspmqsp1 -m MQ01 -p BANK.RQ
/*
```

csq40cfg 作业未指示是否已启用 GSKit 跟踪。但是，您可以通过查看为作业指定的环境变量文件中的设置，或者通过检查是否为运行 csq40cfg 作业的进程创建了跟踪文件，来检查是否启用了跟踪。

关闭 GSKit 跟踪

AMS 地址空间上的 GSKit

要关闭 AMS 地址空间的 GSKit 跟踪，请执行以下过程：

1. 停止队列管理器地址空间。这将同时停止通道启动程序和 AMS 地址空间。
2. 按如下所示修改 csq40env 文件：

```
GSK_TRACE_FILE=/u/<username>/AMStrace/gsktrace/gskssl.%.trc
GSK_TRACE=0x00
```

3. 重新启动队列管理器，通道启动程序和 AMS 地址空间。
4. 请检查作业记录中 AMS 地址空间的环境变量设置，以确保已关闭 GSKit 跟踪。

csq40cfg 作业的 GSKit

要对 csq40cfg 作业关闭 GSKit 跟踪，请执行以下过程：

1. 按如下所示修改 csq40env 文件：

```
GSK_TRACE_FILE=/u/<username>/AMStrace/gsktrace/gskssl.%.trc
GSK_TRACE=0x00
```

2. 提交 csq40cfg 作业并检查是否未生成任何跟踪文件。

注意：

- 在环境文件中，编码 GSK_TRACE=0xff 打开跟踪，编码 GSK_TRACE=0x00 关闭跟踪。
- 在跟踪文件名中包含 % 字符，以确保为不同的 USS 进程生成的跟踪文件名，发出 gsk_* 调用，包含进程标识，因此保持独立。

相关信息

[在 IBM z/OS 上生成 IBM MQ GTF 跟踪](#)

z/OS z/OS 的跟踪输出示例

使用本主题作为如何解释跟踪输出的示例。

第 331 页的图 60 显示了对 MQPUT1 调用的条目进行跟踪的示例。已生成以下项:

- 队列请求参数列表
- 对象描述符 (MQOD)
- 消息描述符 (MQMD)
- 放置消息选项 (MQPMO)
- 消息数据的前 256 个字节

将其与第 332 页的图 61 进行比较, 这说明了退出 IBM MQ 时的相同控制块。

```
USRD9 5E9 ASCB 00F87E80          JOBN ECIC330
CSQW072I ENTRY: MQ user parameter trace
PUTONE
  Thread... 004C2B10  Userid... CICSUSER  pObjDesc. 106B2010
  pMsgDesc. 106B20B8  pPMO.... 106B2200
  BufferL.. 00000064  pBuffer.. 106A0578  RSV1..... 00000000
  RSV2..... 00000000  RSV3..... 116BC830
  C9E8C1E8  C5C3C9C3  AA8E8583  76270484  | IYAYECIC..ec...d |
  D4D8E3E3  0000048C  00000000  00000000  | MQTT.....      |
  00000000  1910C7C2  C9C2D4C9  E8C14BC9  | .....GBIBMIYA.I |
  C7C3E2F2  F0F48E85  83762979  00010000  | GCS204.ec..`.... |

          GMT-01/30/05 14:42:08.412320  LOC-01/30/05 14:42:08.412320

USRD9 5E9 ASCB 00F87E80          JOBN ECIC330
CSQW072I ENTRY: MQ user parameter trace
+0000 D6C44040 00000001 00000000 C2404040 | OD .....B      |
+0010 40404040 40404040 40404040 40404040 | |
...
+00A0 00000000 00000000 | .....          |

          GMT-01/30/05 14:42:08.412345  LOC-01/30/05 14:42:08.412345

USRD9 5E9 ASCB 00F87E80          JOBN ECIC330
CSQW072I ENTRY: MQ user parameter trace
+0000 D4C44040 00000001 00000000 00000008 | MD .....      |
...
+0130 40404040 40404040 40404040 40404040 | |
+0140 40404040 | |

          GMT-01/30/05 14:42:08.412370  LOC-01/30/05 14:42:08.412370

USRD9 5E9 ASCB 00F87E80          JOBN ECIC330
CSQW072I ENTRY: MQ user parameter trace
+0000 D7D4D640 00000001 00000000 FFFFFFFF | PMO .....      |
...
+0070 40404040 40404040 40404040 40404040 | |

          GMT-01/30/05 14:42:08.412393  LOC-01/30/05 14:42:08.412393

USRD9 5E9 ASCB 00F87E80          JOBN ECIC330
CSQW072I ENTRY: MQ user parameter trace
+0000 C1C1C1C1 C1C1C1C1 C1404040 40404040 | AAAAAAAAAA      |
...
+0060 40404040 | |

          GMT-01/30/05 14:42:08.412625  LOC-01/30/05 14:42:08.412625
```

图 60: 来自 MQPUT1 请求的条目跟踪的示例跟踪数据

```

USRD9 5EA ASCB 00F87E80          JOBN ECIC330
CSQW073I EXIT: MQ user parameter trace
PUTONE
  Thread... 004C2B10  Userid... CICSUSER  pObjDesc. 106B2010
  pMsgDesc. 106B20B8  pPMO.... 106B2200
  BufferL.. 00000064  pBuffer.. 106A0578  RSV1..... 00000000
  RSV2.... 00000000  RSV3.... 116BC830
  CompCode. 00000002  Reason... 000007FB
  C9E8C1E8  C5C3C9C3  AA8E8583  76270484  | IYAYECIC..ec...d |
  D4D8E3E3  00000048C  00000000  00000000  | MQTT..... |
  00000000  1910C7C2  C9C2D4C9  E8C14BC9  | .....GBIBMIYA.I |
  C7C3E2F2  F0F48E85  83762979  00010000  | GCS204.ec..'.... |
MQRC_OBJECT_TYPE_ERROR

          GMT-01/30/05 14:42:08.412678  LOC-01/30/05 14:42:08.412678

USRD9 5EA ASCB 00F87E80          JOBN ECIC330
CSQW073I EXIT: MQ user parameter trace
+0000 D6C44040 00000001 00000000 C2404040 | OD .....B |
...
+00A0 00000000 00000000 | ..... |

          GMT-01/30/05 14:42:08.412789  LOC-01/30/05 14:42:08.412789

USRD9 5EA ASCB 00F87E80          JOBN ECIC330
CSQW073I EXIT: MQ user parameter trace
+0000 D4C44040 00000001 00000000 00000008 | MD ..... |
...
+0140 40404040 | |

          GMT-01/30/05 14:42:08.412814  LOC-01/30/05 14:42:08.412814

USRD9 5EA ASCB 00F87E80          JOBN ECIC330
CSQW073I EXIT: MQ user parameter trace
+0000 D7D4D640 00000001 00000000 FFFFFFFF | PMO ..... |
...
+0070 40404040 40404040 40404040 40404040 | |

          GMT-01/30/05 14:42:08.412836  LOC-01/30/05 14:42:08.412836

USRD9 5EA ASCB 00F87E80          JOBN ECIC330
CSQW073I EXIT: MQ user parameter trace
+0000 C1C1C1C1 C1C1C1C1 C1404040 40404040 | AAAAAAAAAA |
...
+0060 40404040 | |

          GMT-01/30/05 14:42:08.412858  LOC-01/30/05 14:42:08.412858

```

图 61: 来自 MQPUT1 请求的出口跟踪的跟踪数据示例

跟踪高级消息排队协议 (AMQP) 服务

利用高级消息排队协议 (AMQP) 服务所提供的跟踪功能，可以帮助 IBM 支持人员诊断与服务相关的客户问题。

关于此任务

可使用两种方法来控制 IBM MQ AMQP 服务的跟踪：

- 使用 **strmqtrc** 和 **endmqtrc** 命令（用于启动和停止跟踪）。使用 **strmqtrc** 命令启用跟踪后，将生成正在运行 IBM MQ AMQP 服务的整个队列管理器的跟踪信息。这包含 IBM MQ AMQP 服务自身，以及该服务用于与其他队列管理器组件进行通信的底层 Java 消息排队接口 (JMQUI)。

V 9.1.5 从 IBM MQ 9.1.5 开始，您还可以生成选定的感兴趣区域的跟踪信息。

- 运行 **controlAMQPChannel** 命令。请注意，使用 **controlAMQPChannel** 命令开启跟踪将仅跟踪 IBM MQ AMQP 服务。

如果您不确定要使用的选项，请联系 IBM 支持代表，他们将能够提出最佳方式建议，帮助您为所看到的问题收集跟踪信息。

过程

1. 方法 1

- a) 打开命令提示符，浏览至以下目录：

```
MQ_INSTALLATION_PATH\bin
```

- b) 运行 **strmqtrc** 命令来启用跟踪：

对于 Long Term Support 和 IBM MQ 9.1.5 之前的 Continuous Delivery，请运行以下命令：

```
strmqtrc -m qmgr_name
```

其中，*qmgr_name* 是正在运行 IBM MQ AMQP 服务的队列管理器的名称。

V 9.1.5 从 IBM MQ 9.1.5，运行以下命令：

```
strmqtrc -m qmgr_name -t amqp
```

其中，*qmgr_name* 是正在运行 IBM MQ AMQP 服务的队列管理器的名称，而 **-t amqp** 将跟踪输出仅限于 AMQP 服务。

- c) 重现该问题。

- d) 通过运行以下命令来停止跟踪：

```
endmqtrc -m qmgr_name
```

2. 方法 2

- a) 打开命令提示符，浏览至以下目录：

```
MQ_INSTALLATION_PATH\bin
```

- b) 运行以下命令来启用跟踪：

• **Windows**

```
controlAMQPChannel -qmgr=qmgr_name -mode=starttrace
```

• **Linux** **UNIX**

```
./controlAMQPChannel.sh -qmgr=qmgr_name -mode=starttrace
```

其中，*qmgr_name* 是正在运行 AMQP 服务的队列管理器的名称。

- c) 重现该问题。

- d) 在发生问题时，通过运行以下命令来停止跟踪：

• **Windows**

```
controlAMQPChannel -qmgr=qmgr_name -mode=stoptrace
```

• **Linux** **UNIX**

```
./controlAMQPChannel.sh -qmgr=qmgr_name -mode=stoptrace [clientid=ClientIdentifier]
```

其中，*qmgr_name* 是正在运行 AMQP 服务的队列管理器的名称。

结果

要查看跟踪输出，请切换至以下目录：

• **Windows** `MQ_DATA_PATH\trace.`

• **Linux** **UNIX** `/var/mqm/trace.`

包含来自 AMQP 服务的信息的跟踪文件被命名为 `amqp_N.trc`（其中 N 是编号）。

V 9.1.5 从 IBM MQ 9.1.5 开始，将按照以下方式命名跟踪文件：

- 包含 AMQP 服务信息的跟踪文件被称为 `amqpRunMQXRService_PPPPP.N.trc`，其中 `PPPPP` 是 AMQP 服务的进程标识，`N` 是数字。
- 包含来自 `controlAMQPChannel` 命令的信息的跟踪文件被命名为 `amqpControlMQXRChannel_PPPPP.N.trc`（其中 `PPPPP` 是 AMQP 服务的进程标识，`N` 是编号）。

由 JMQUI 生成的跟踪信息将写入名为 `amqp_PPPPP.trc` 的跟踪文件，其中 `PPPPP` 是 AMQP 服务的进程标识。

Windows Linux AIX 使用 controlAMQPChannel 命令进行的其他诊断

使用 `controlAMQPChannel` 命令来提供有关 AMQP 服务的其他诊断信息。

过程

运行以下命令以提供来自 MQXR 服务的有用诊断信息：

```
<MQ_INSTALLATION_PATH>\amqp\bin\controlAMQPChannel -qmgr=<QMGR_NAME> -mode=diagnostics  
-diagnosticstype=<number>
```

生成的诊断信息取决于 `-diagnosticstype=<number>` 参数的值：

-diagnosticstype= 0

写入到控制台的线程转储

-diagnosticstype= 1

具有某些内部服务统计信息的 FDC

-diagnosticstype= 2

具有内部统计信息的 FDC，以及有关当前已连接的客户机的信息

-diagnosticstype= 3

堆转储

-diagnosticstype= 4

Javacore

-diagnosticstype= 5

完整系统转储

-diagnosticstype= 6

有关特定客户机的详细信息。请注意，您还必须提供 `-clientid` 参数来表示该客户机。

V 9.1.0 Linux 跟踪 IBM MQ Bridge to Salesforce

针对 IBM MQ Bridge to Salesforce 提供了跟踪功能来帮助 IBM 员工诊断客户问题。当您发出 `runmqsfb` 命令以启动网桥时，请对 IBM MQ Bridge to Salesforce 启用跟踪并定义调试级别。

开始之前

注：在 2022 年 11 月 22 日起的所有发行版中不推荐使用 IBM MQ Bridge to Salesforce（请参阅 [美国声明函 222-431](#)）。

过程

1. 设置环境变量 `MQSFB_EXTRA_JAVA_OPTIONS` 以指定 `-D` Java 选项和启动 IBM MQ classes for JMS 跟踪。

```
export MQSFB_EXTRA_JAVA_OPTIONS="-Dcom.ibm.msg.client.commonservices.trace.status=ON"
```

2. 在运行时发出 `runmqsfb` 命令时，将调试级别设置为详细方式 `-d 2`。

```
runmqsfb -f new_config.cfg -r logFile.log -d 2
```

logFile.log 包含可能有助于解决 IBM MQ Bridge to Salesforce 问题的信息。

3. 可选：您可以通过创建 IBM MQ classes for JMS 配置文件来更好地控制精确跟踪。有关更多信息，请参阅第 339 页的『跟踪 IBM MQ classes for JMS 应用程序』，并遵循您的 IBM 服务支持代表提供的建议。

相关任务

[运行 IBM MQ Bridge to Salesforce](#)

[监视 IBM MQ Bridge to Salesforce](#)

相关参考

[runmqsfb \(运行 IBM MQ 网桥到 Salesforce\)](#)

V 9.1.0

Linux

z/OS

MQ Adv. VUE

跟踪 IBM MQ Bridge to blockchain

针对 IBM MQ Bridge to blockchain 提供了跟踪功能来帮助 IBM 员工诊断客户问题。当您发出 `runmqbcb` 命令以启动网桥时，请对 IBM MQ Bridge to blockchain 启用跟踪并定义调试级别。

开始之前

注：在 2022 年 11 月 22 日起的所有发行版中不推荐使用 IBM MQ Bridge to blockchain (请参阅 [美国声明函 222-431](#))。

过程

1. 设置环境变量 `MQBCB_EXTRA_JAVA_OPTIONS` 以指定 `-D` Java 选项和开启 IBM MQ classes for JMS 跟踪。

```
export MQBCB_EXTRA_JAVA_OPTIONS="-Dcom.ibm.msg.client.commonservices.trace.status=ON"
```

2. 在运行时发出 `runmqbcb` 命令时，将调试级别设置为详细方式 `-d 2`。在 z/OS 上，您还可以通过编辑启动的任务 JCL 来实现此目的。

```
./runmqbcb.sh -f new_config.cfg -r logFile.log -d 2
```

logFile.log 包含可能有助于解决 IBM MQ Bridge to blockchain 问题的信息。

3. 可选：您可以通过创建 IBM MQ classes for JMS 配置文件来更好地控制精确跟踪。有关更多信息，请参阅第 339 页的『跟踪 IBM MQ classes for JMS 应用程序』，并遵循您的 IBM 服务支持代表提供的建议。

相关任务

[运行 IBM MQ Bridge to blockchain](#)

相关参考

V 9.1.4

[runmqbcb \(运行 IBM MQ Bridge to Blockchain\)](#)


跟踪 IBM MQ Console 和 REST API

IBM MQ Console 和 REST API 中提供的跟踪功能旨在帮助 IBM 工作人员诊断客户问题。可通过各种属性来控制这些功能的行为。

开始之前

在为 IBM 服务人员收集诊断信息时，请包含以下文件和目录：

- mqweb.xml 文件。
- 包含 mqweb 服务器定义的目录的内容：

–  `MQ_DATA_PATH/web/installations/installationName`

– 

在运行 `crtmqweb` 脚本来创建 mqweb 服务器定义时指定的目录。缺省情况下，此目录为 `/var/mqm/web/installation1`。

关于此任务

IBM MQ Console 和 REST API 一个包含三个功能区域，每个功能区域都有自己的跟踪机制：

- 在浏览器中执行的 [IBM MQ Console JavaScript 代码](#)。
- 在 mqweb 服务器中运行的 [IBM MQ Console 和 REST API 代码](#)。
- 在 mqweb 服务器中运行的 [IBM MQ Classes for JMS 代码](#)。

V 9.1.5 **Multi** 通过 [New Web Console](#) (可从 IBM MQ 9.1.5 用于 Multiplatforms 版上的 Continuous Delivery)，只能跟踪在 mqweb 服务器中运行的后端代码。

V 9.1.5 从 IBM MQ 9.1.5 开始，原先的 Web 控制台被称为 Dashboard Web Console。通过 Dashboard Web Console，可以对后端代码和浏览器启用跟踪。


过程

- 为 IBM MQ Console 启用浏览器跟踪。请完成下列步骤。

注: **V 9.1.5** 此跟踪仅在 Dashboard Web Console 上可用。

仅从在其中启用跟踪的浏览器输出跟踪。从 IBM MQ Console 注销后，会自动禁用跟踪。

a) 登录到 IBM MQ Console。

b) 单击“设置” 图标。

c) 在命令行上输入以下命令：

```
setmqweb properties -k traceSpec -v  
"*=info:com.ibm.mq*=all:com.ibm.mq.rest*=all:js.mq*=all"
```

d) 确保针对[消息传递跟踪](#)和[浏览器跟踪](#)选中了启用。单击保存。

这样会开始跟踪浏览器中执行的操作。此跟踪将定期发送到 mqweb 服务器中运行的 IBM MQ Console 代码，并且在 mqweb 服务器跟踪日志和消息传递跟踪中输出。

- 要为 mqweb 服务器中运行的 IBM MQ Console 和 REST API 代码启用跟踪，请在命令行上输入以下命令：

```
setmqweb properties -k traceSpec -v "*=info:com.ibm.mq*=all:com.ibm.mq.rest*=all:js.mq*=all"
```

如果 mqweb 服务器正在运行，将立即启用跟踪。

跟踪将输出到一组文件中。活动文件名为 `trace.log`。历史跟踪保存在名为 `trace_timestamp.log` 的文件中。可通过设置 `maxTraceFileSize` 和 `maxTraceFiles` 变量来配置这些跟踪文件的大小以及保留的历史文件数。缺省情况下，最大跟踪文件大小为 20 MB，跟踪文件的最大数量为 2。有关更多信息，请参阅[配置日志记录](#)。

- 要对在 mqweb 服务器中运行的 REST API 代码启用消息传递跟踪，请完成以下步骤：

a) 在以下某个目录中创建名为 `jmstrace.config` 的文件：

- **ULW** `MQ_DATA_PATH/web/installations/installationName/servers/mqweb`
- **Z/OS** `WLP_user_directory/servers/mqweb`

其中，`WLP_user_directory` 是在运行 `crtmqweb` 脚本来创建 mqweb 服务器定义时指定的目录。

b) 将以下行添加到 `jmstrace.config` 文件：

```
com.ibm.msg.client.commonservices.trace.outputName=PATH/logs/jmstrace.txt  
com.ibm.msg.client.commonservices.trace.limit=104857600  
com.ibm.msg.client.commonservices.trace.count=10  
com.ibm.msg.client.commonservices.trace.status=ON
```


其中 *PATH* 指定要写入 *jmstrace.txt* 文件的目录的完整路径。

这些行将最大跟踪文件大小设置为 100 MB，并将最大跟踪文件数设置为 10。请确保有足够的磁盘空间可用于保存这些文件。

- c) 在与 *jmstrace.config* 文件相同的目录中，打开或创建 *jvm.options* 文件。
- d) 将以下行添加到 *jvm.options* 文件：

```
-Dcom.ibm.msg.client.commonservices.trace.startup=TRUE  
-Dcom.ibm.msg.client.config.location=CONFIG_PATH/jmstrace.config
```

其中 *CONFIG_PATH* 指定 *jmstrace.config* 文件所在目录的完整路径作为 URL。例如，`file:c:/ProgramData/IBM/MQ/web/installations/Installation2/servers/mqweb/`。

- e) 在命令行上使用以下命令来重新启动 *mqweb* 服务器：

```
endmqweb  
startmqweb
```

相关任务

[使用 Web 控制台进行管理](#)

跟踪 IBM MQ Internet Pass-Thru 中的错误

IBM MQ Internet Pass-Thru (MQIPT) 提供了由 **Trace** 属性控制的详细执行跟踪工具。

跟踪文件将写入 *mqipt_home\errors* 目录（其中 *mqipt_home* 是 MQIPT 主目录，包含 *mqipt.conf*）。生成的每个跟踪文件都具有以下格式的名称：

```
AMQyyyyymmddnnnnnnnnn.n.TRC.n
```

意外的致命错误会作为 FFST 记录写入 *mqipt_home\errors* 目录中的错误日志文件。FFST 文件具有以下格式：

```
AMQyyyyymmddnnnnnnnnn.n.FDC
```

要启用跟踪，请在 *mqipt.conf* 文件的相应部分中添加 **Trace** 配置属性。可以在要跟踪的每个路由的 `[route]` 部分中或者在 `[global]` 部分中指定 **Trace** 属性。未指定 **Trace** 属性的所有路由都会继承 `[global]` 部分中 **Trace** 属性的值。

相关任务

[第 266 页的『收集 MQIPT 问题的信息』](#)

如果需要使用 MQIPT 向 IBM 支持人员报告问题，请发送有助于更快解决问题的相关信息。

相关参考

[第 53 页的『对 IBM MQ Internet Pass-Thru 问题进行故障诊断』](#)

您可以执行若干步骤来帮助确定在使用 IBM MQ Internet Pass-Thru (MQIPT) 时可能遇到的任何问题的性质。

在 *mqiptKeyman* 和 *mqiptKeycmd* 中跟踪错误

mqiptKeycmd 和 *mqiptKeyman* 命令都具有执行跟踪功能，可以诊断证书管理工具中的错误。

要对这些命令启用跟踪，请在运行 *mqiptKeycmd* 或 *mqiptKeyman* 命令之前设置以下环境变量：

- 在 Windows 系统上：

```
set MQIPT_JVM_OPTIONS=-Dkeyman.debug=true -Dkeyman.logging=true
```

- 在 UNIX and Linux 系统上:

```
MQIPT_JVM_OPTIONS="-Dkeyman.debug=true -Dkeyman.logging=true"  
export MQIPT_JVM_OPTIONS
```

这样会在当前工作目录中创建一个跟踪文件。此跟踪文件的名称采用以下格式:

```
debugTrace.n
```

其中, *n* 是从 0 开始的递增数字。

运行证书管理工具的用户必须具有在当前工作目录中创建文件的许可权, 否则该命令将失败并返回错误。

在记录完跟踪日志之后, 请复原该环境变量。

跟踪用户定义的安全出口

要帮助诊断用户定义的安全出口中的问题, 您可以启用跟踪功能, 类似于 MQIPT 所使用的功能。

通过将路由 **Trace** 属性设置为 1 - 5 范围内的值来启用跟踪。请参阅 MQIPT 路由属性中 **Trace** 的条目。

可能会有安全出口的多个实例同时运行, 因此跟踪文件中的个别条目可以使用线程标识符识别。

跟踪功能在安全出口启动时由 MQIPT 初始化; 您所要做的只是选择要跟踪的信息。样本用户出口中有很多跟踪示例。请参阅 [安全出口 \(security exit\)](#)。

进行跟踪至少需要 `entry` 调用、`exit` 调用和您要跟踪的数据。例如:

```
/**  
 * This method is called to initialize the exit (for example, for  
 * loading validation information) and place itself in a ready  
 * state to validate connection requests.  
 */  
public int init(IPTrace t) {  
    final String strMethod = "CustomExit.init";  
  
    // Trace entry into this method  
    t.entry(strMethod);  
  
    // Trace useful information  
    t.data(strMethod, "Starting exit - MQIPT version " + getVersion());  
  
    // Perform initialization and load any data  
    t.data(strMethod, "Ready for work");  
  
    // Trace exit from this method  
    t.exit(strMethod);  
  
    return 0;  
}
```

跟踪 IBM MQ.NET 应用程序

在 IBM MQ .NET 中, 您可以使用 MQI 启动和控制跟踪功能, 就像在 IBM MQ 程序中一样。

但是, `strmqtrc` 命令的 `-i` 和 `-p` 参数使您可以指定进程和线程标识符, 而已命名的进程不会产生任何效果。

您通常只需根据 IBM 服务的请求使用跟踪功能。

请参阅在 [Windows 上使用跟踪](#) 以获取有关跟踪命令的信息。

跟踪 JMS 和 Java 应用程序

提供了 JMS 和 Java 应用程序的跟踪工具, 以帮助 IBM 支持人员诊断问题和问题。您可以跟踪各种不同的资源。

跟踪 IBM MQ classes for JMS 应用程序

IBM MQ classes for JMS 中提供了跟踪功能来帮助 IBM 支持人员诊断客户问题。各种属性均可控制此功能的行为。

如果要求您提供跟踪输出来调查问题，请使用以下提到的选项之一：

- 如果问题很容易重新创建，请使用 Java 系统属性来收集 IBM MQ classes for JMS 跟踪。有关更多信息，请参阅第 340 页的『使用 Java 系统属性收集 IBM MQ classes for JMS 跟踪』。
- 如果应用程序需要运行一段时间才会发生问题，请通过使用 IBM MQ classes for JMS 配置文件来收集 IBM MQ classes for JMS 跟踪信息。有关更多信息，请参阅第 340 页的『使用 IBM MQ classes for JMS 配置文件收集 IBM MQ classes for JMS 跟踪信息』。
- 要从当前正在运行的应用程序生成跟踪，请通过使用 traceControl 实用程序来动态收集 IBM MQ classes for JMS 跟踪信息。有关更多信息，请参阅第 341 页的『使用 traceControl 实用程序动态收集 IBM MQ classes for JMS 跟踪信息』。

如果您不确定要使用的选项，请联系 IBM 支持代表，他们将能够提出最佳方式建议，帮助您为所看到的问题收集跟踪信息。

如果发生严重或不可恢复错误，那么会将 First Failure Support Technology (FFST) 信息记录在格式为 JMSSC *xxxx*.FDC 的文件名中，其中 *xxxx* 是四位数字。此数字会递增以区分 .FDC 文件。

.FDC 文件始终写入到名为 FFDC 的子目录中。子目录位于两个位置之一，具体取决于是否激活跟踪：

激活了跟踪并设置了 *traceOutputName*

创建 FFDC 目录，作为向其写入跟踪文件的目录的子目录。

未激活跟踪，或未设置 *traceOutputName*

创建 FFDC 目录，作为当前工作目录的子目录。

有关 IBM MQ classes for JMS 中的 FFST 的更多信息，请参阅第 294 页的『FFST: IBM MQ classes for JMS』。

JSE 公共服务使用 `java.util.logging` 作为其跟踪和日志记录基础结构。此基础架构的根对象为 `LogManager`。日志管理器包含 `reset` 方法，该方法将关闭所有处理程序，并将日志级别设置为 `null`，它实际上将关闭所有跟踪。如果应用程序或应用程序服务器调用

`java.util.logging.LogManager.getLogManager().reset()`，那么它将关闭所有跟踪，这可能会阻止您诊断任何问题。要避免关闭所有跟踪，请使用无任何作用的覆盖的 `reset()` 方法创建 `LogManager` 类，如下示例中所示：

```
package com.ibm.javaut.tests;
import java.util.logging.LogManager;
public class JmsLogManager extends LogManager {
    // final shutdown hook to ensure that the trace is finally shutdown
    // and that the lock file is cleaned-up
    public class ShutdownHook extends Thread{
        public void run(){
            doReset();
        }
    }
    public JmsLogManager(){
        // add shutdown hook to ensure final cleanup
        Runtime.getRuntime().addShutdownHook(new ShutdownHook());
    }
    public void reset() throws SecurityException {
        // does nothing
    }
    public void doReset(){
        super.reset();
    }
}
```

关闭挂钩是确保在 JVM 完成时正确关闭跟踪所必需的。要使用修改的日志管理器（而不是缺省日志管理器），请将系统属性添加到 JVM 启动中：

```
java -Djava.util.logging.manager=com.mycompany.logging.LogManager ...
```

使用 Java 系统属性收集 IBM MQ classes for JMS 跟踪

对于可以在短时间内重现的问题，应通过在启动应用程序时设置 Java 系统属性来收集 IBM MQ classes for JMS 跟踪。

关于此任务



要使用 Java 系统属性来收集跟踪信息，请完成以下步骤。

过程

- 使用以下命令运行要跟踪的应用程序：

```
java -Dcom.ibm.msg.client.commonservices.trace.status=ON application_name
```

当应用程序启动时，IBM MQ classes for JMS 开始将跟踪信息写入应用程序当前工作目录中的跟踪文件。跟踪文件的名称取决于应用程序的运行环境：

- 对于 IBM MQ classes for JMS for IBM MQ 9.0.0 Fix Pack 1 或更低版本，会将跟踪写入名为 `mqjms_%PID%.trc` 的文件。
- 从 IBM MQ 9.0.0 Fix Pack 2 起，如果应用程序已从 JAR 文件 `com.ibm.mqjms.jar` 装入 IBM MQ classes for JMS，那么会将跟踪写入名为 `mqjava_%PID%.trc` 的文件。
- 从 IBM MQ 9.0.0 Fix Pack 2 起，如果应用程序已从可重定位的 JAR 文件 `com.ibm.mq.allclient.jar` 装入 IBM MQ classes for JMS，那么会将跟踪写入名为 `mqjavaclient_%PID%.trc` 的文件。
-  从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 开始，如果应用程序已从 JAR 文件 `com.ibm.mqjms.jar` 装入 IBM MQ classes for JMS，那么会将跟踪写入名为 `mqjava_%PID%.cl%u.trc` 的文件。
-  从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 开始，如果应用程序已从可重新定位的 JAR 文件 `com.ibm.mq.allclient.jar` 装入 IBM MQ classes for JMS，那么会将跟踪写入名为 `mqjavaclient_%PID%.cl%u.trc` 的文件。

其中，`%PID%` 是正在跟踪的应用程序的进程标识，`%u` 是唯一编号，用于区分在不同 Java 类装入器下运行跟踪的线程之间的文件。

应用程序停止时，即停止向跟踪文件写入信息。

如果应用程序必须长时间运行后才会发生为其收集跟踪信息的问题，那么跟踪文件可能过大。在此情况下，请考虑使用 IBM MQ classes for JMS 配置文件来收集跟踪信息（请参阅第 340 页的『使用 IBM MQ classes for JMS 配置文件收集 IBM MQ classes for JMS 跟踪信息』）。通过这种方式启用跟踪时，可以控制 IBM MQ classes for JMS 生成的跟踪数据量。

使用 IBM MQ classes for JMS 配置文件收集 IBM MQ classes for JMS 跟踪信息

如果应用程序必须长时间运行后才会发生问题，应使用 IBM MQ classes for JMS 配置文件来收集 IBM MQ classes for JMS 跟踪信息。此配置文件允许您指定各种选项来控制收集的跟踪数据量。

关于此任务

要使用 IBM MQ classes for JMS 配置文件来收集跟踪信息，请完成以下步骤。

过程

- 创建 IBM MQ classes for JMS 配置文件。
有关此文件的更多信息，请参阅 [IBM MQ classes for JMS 配置文件](#)。
- 编辑 IBM MQ classes for JMS 配置文件，以便将属性 `com.ibm.msg.client.commonservices.trace.status` 设置为值 `ON`。
- 可选：编辑 IBM MQ classes for JMS 配置文件 Java Standard Edition 跟踪设置中所列的其他属性。
- 使用以下命令来运行 IBM MQ classes for JMS 应用程序：

```
java -Dcom.ibm.msg.client.config.location=config_file_url  
application_name
```





其中, *config_file_url* 是统一资源定位符 (URL), 指定 IBM MQ classes for JMS 配置文件的名称和位置。支持以下类型的 URL: http、file、ftp 和 jar。

以下是 Java 命令的一个示例:

```
java -Dcom.ibm.msg.client.config.location=file:/D:/mydir/myjms.config  
MyAppClass
```

此命令将 IBM MQ classes for JMS 配置文件标识为本地 Windows 系统上的 D:\mydir\myjms.config 文件。

缺省情况下, 当应用程序启动时, IBM MQ classes for JMS 会开始将跟踪信息写入应用程序当前工作目录中的跟踪文件。跟踪文件的名称取决于应用程序的运行环境:

- 对于 IBM MQ classes for JMS for IBM MQ 9.0.0 Fix Pack 1 或更低版本, 会将跟踪写入名为 mqjms_%PID%.trc 的文件。
- 从 IBM MQ 9.0.0 Fix Pack 2 起, 如果应用程序已从 JAR 文件 com.ibm.mqjms.jar 装入 IBM MQ classes for JMS, 那么会将跟踪写入名为 mqjava_%PID%.trc 的文件。
- 从 IBM MQ 9.0.0 Fix Pack 2 起, 如果应用程序已从可重定位的 JAR 文件 com.ibm.mq.allclient.jar 装入 IBM MQ classes for JMS, 那么会将跟踪写入名为 mqjavaclient_%PID%.trc 的文件。
-   从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 开始, 如果应用程序已从 JAR 文件 com.ibm.mqjms.jar 装入 IBM MQ classes for JMS, 那么会将跟踪写入名为 mqjava_%PID%.cl%u.trc 的文件。
-   从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 开始, 如果应用程序已从可重新定位的 JAR 文件 com.ibm.mq.allclient.jar 装入 IBM MQ classes for JMS, 那么会将跟踪写入名为 mqjavaclient_%PID%.cl%u.trc 的文件。

其中, %PID% 是正在跟踪的应用程序的进程标识, %u 是唯一编号, 用于区分在不同 Java 类装入器下运行跟踪的线程之间的文件。

要更改跟踪文件的名称及其写入位置, 请确保应用程序使用的 IBM MQ classes for JMS 配置文件包含属性 **com.ibm.msg.client.commonservices.trace.outputName** 的条目。该属性值可以是以下任一值:

- 在应用程序的工作目录中创建的跟踪文件的名称。
- 跟踪文件的标准名称, 包括创建文件的目录。

例如, 要配置 IBM MQ classes for JMS 以将应用程序的跟踪信息写入名为 C:\Trace\trace.trc 的文件, 那么应用程序使用的 IBM MQ classes for JMS 配置文件需要包含以下条目:

```
com.ibm.msg.client.commonservices.trace.outputName=C:\Trace\trace.trc
```

使用 *traceControl* 实用程序动态收集 *IBM MQ classes for JMS* 跟踪信息

随附于 IBM MQ classes for JMS 的 *traceControl* 实用程序允许从正在运行的应用程序收集跟踪信息。如果 IBM 支持人员需要在发生问题后查看来自应用程序的跟踪信息, 或者需要从不能停止的关键性应用程序收集跟踪信息, 那么此方法十分有用。

关于此任务

要点: 仅 IBM Java runtime environments (JRE) 支持此功能。

有关 *traceControl* 实用程序的更多信息, 请参阅第 349 页的『使用 IBM MQ classes for Java 和 IBM MQ classes for JMS 控制运行进程中的跟踪』。

要使用 *traceControl* 实用程序收集跟踪信息, 请完成以下步骤。

过程

1. 启动命令提示符，并浏览至 `MQ_INSTALLATION_PATH\java\lib` 目录。
2. 运行以下命令：

```
java -jar com.ibm.mq.traceControl.jar -list
```

此命令会显示系统上所有 Java 进程的列表。

3. 识别需要跟踪的 IBM MQ classes for JMS 应用程序的进程标识，并运行以下命令：

```
java -jar com.ibm.mq.traceControl.jar -i processidentifier -enable
```

现在已为应用程序开启跟踪。

启用跟踪后，IBM MQ classes for JMS 会开始将跟踪信息写入应用程序当前工作目录中的跟踪文件。跟踪文件的名称取决于应用程序的运行环境：

- 对于 IBM MQ classes for JMS for IBM MQ 9.0.0 Fix Pack 1 或更低版本，会将跟踪写入名为 `mqjms_%PID%.trc` 的文件。
- 从 IBM MQ 9.0.0 Fix Pack 2 起，如果应用程序已从 JAR 文件 `com.ibm.mqjms.jar` 装入 IBM MQ classes for JMS，那么会将跟踪写入名为 `mqjava_%PID%.trc` 的文件。
- 从 IBM MQ 9.0.0 Fix Pack 2 起，如果应用程序已从可重定位的 JAR 文件 `com.ibm.mq.allclient.jar` 装入 IBM MQ classes for JMS，那么会将跟踪写入名为 `mqjavaclient_%PID%.trc` 的文件。
- **V 9.1.5** **V 9.1.0.5** 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 开始，如果应用程序已从 JAR 文件 `com.ibm.mqjms.jar` 装入 IBM MQ classes for JMS，那么会将跟踪写入名为 `mqjava_%PID%.cl%u.trc` 的文件。
- **V 9.1.5** **V 9.1.0.5** 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 开始，如果应用程序已从重新定位的 JAR 文件 `com.ibm.mq.allclient.jar` 装入 IBM MQ classes for JMS，那么会将跟踪写入名为 `mqjavaclient_%PID%.cl%u.trc` 的文件。

其中，`%PID%` 是正在跟踪的应用程序的进程标识，`%u` 是唯一编号，用于区分在不同 Java 类装入器下运行跟踪的线程之间的文件。

4. 要关闭跟踪，请运行以下命令：

```
java -jar com.ibm.mq.traceControl.jar -i processidentifier -disable
```

跟踪 IBM MQ classes for Java 应用程序

IBM MQ classes for Java 中提供了跟踪功能来帮助 IBM 支持人员诊断客户问题。各种属性均可控制此功能的行为。

关于此任务

如果要求您提供跟踪输出来调查问题，请使用以下提到的选项之一：

- 如果问题很容易重新创建，请使用 Java 系统属性来收集 IBM MQ classes for Java 跟踪。有关更多信息，请参阅第 343 页的『使用 Java 系统属性收集 IBM MQ classes for Java 跟踪』。
- 如果应用程序需要运行一段时间才会发生问题，请通过使用 IBM MQ classes for Java 配置文件来收集 IBM MQ classes for Java 跟踪信息。有关更多信息，请参阅第 344 页的『使用 IBM MQ classes for Java 配置文件收集 IBM MQ classes for Java 跟踪信息』。
- 要从当前正在运行的应用程序生成跟踪，请通过使用 `traceControl` 实用程序来动态收集 IBM MQ classes for Java 跟踪信息。有关更多信息，请参阅第 345 页的『使用 `traceControl` 实用程序动态收集 IBM MQ classes for Java 跟踪信息』。

如果您不确定要使用的选项，请联系 IBM 支持代表，他们将能够提出最佳方式建议，帮助您为所看到的问题收集跟踪信息。

如果发生严重或不可恢复错误，那么会将 First Failure Support Technology (FFST) 信息记录在格式为 JAVACC *xxxx*.FDC 的文件名中，其中 *xxxx* 是四位数字。它会递增以区分 .FDC 文件。

.FDC 文件始终写入到名为 FFDC 的子目录中。子目录位于两个位置之一，具体取决于是否激活跟踪：

激活了跟踪并设置了 **traceOutputName**

创建 FFDC 目录，作为向其写入跟踪文件的目录的子目录。

未激活跟踪，或未设置 **traceOutputName**

创建 FFDC 目录，作为当前工作目录的子目录。

JSE 公共服务使用 `java.util.logging` 作为其跟踪和日志记录基础结构。此基础架构的根对象为 `LogManager`。日志管理器包含 `reset` 方法，该方法将关闭所有处理程序，并将日志级别设置为 `null`，它实际上将关闭所有跟踪。如果应用程序或应用程序服务器调用 `java.util.logging.LogManager.getLogManager().reset()`，那么它将关闭所有跟踪，这可能会阻止您诊断任何问题。要避免关闭所有跟踪，请使用无任何作用的覆盖的 `reset()` 方法创建 `LogManager` 类，如下示例中所示：

```
package com.ibm.javaut.tests;
import java.util.logging.LogManager;
public class JmsLogManager extends LogManager {
    // final shutdown hook to ensure that the trace is finally shutdown
    // and that the lock file is cleaned-up
    public class ShutdownHook extends Thread{
        public void run(){
            doReset();
        }
    }
    public JmsLogManager(){
        // add shutdown hook to ensure final cleanup
        Runtime.getRuntime().addShutdownHook(new ShutdownHook());
    }
    public void reset() throws SecurityException {
        // does nothing
    }
    public void doReset(){
        super.reset();
    }
}
```

关闭挂钩是确保在 JVM 完成时正确关闭跟踪所必需的。要使用修改的日志管理器（而不是缺省日志管理器），请将系统属性添加到 JVM 启动中：

```
java -Djava.util.logging.manager=com.mycompany.logging.LogManager ...
```

使用 Java 系统属性收集 **IBM MQ classes for Java** 跟踪

对于可以在短时间内重现的问题，应通过在启动应用程序时设置 Java 系统属性来收集 IBM MQ classes for Java 跟踪。

关于此任务

要使用 Java 系统属性来收集跟踪信息，请完成以下步骤。



过程

- 使用以下命令运行要跟踪的应用程序：

```
java -Dcom.ibm.msg.client.commonservices.trace.status=ON application_name
```

当应用程序启动时，IBM MQ classes for Java 开始将跟踪信息写入应用程序当前工作目录中的跟踪文件。跟踪文件的名称取决于应用程序的运行环境：

- 对于 IBM MQ classes for Java for IBM MQ 9.0.0 Fix Pack 1 或更低版本，会将跟踪写入名为 `mjms_%PID%.trc` 的文件。

- 从 IBM MQ 9.0.0 Fix Pack 2 起，如果应用程序已从 JAR 文件 `com.ibm.mq.jar` 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mqjava_%PID%.trc` 的文件。
- 从 IBM MQ 9.0.0 Fix Pack 2 起，如果应用程序已从可重定位的 JAR 文件 `com.ibm.mq.allclient.jar` 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mqjavaclient_%PID%.trc` 的文件。
-  从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 开始，如果应用程序已从 JAR 文件 `com.ibm.mq.jar` 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mqjava_%PID%.cl%u.trc` 的文件。
-  从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 开始，如果应用程序已从可重新定位的 JAR 文件 `com.ibm.mq.allclient.jar` 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mqjavaclient_%PID%.cl%u.trc` 的文件。

其中，`%PID%` 是正在跟踪的应用程序的进程标识，`%u` 是唯一编号，用于区分在不同 Java 类装入器下运行跟踪的线程之间的文件。

应用程序停止时，即停止向跟踪文件写入信息。

如果应用程序必须长时间运行后才会发生为其收集跟踪信息的问题，那么跟踪文件可能过大。在此情况下，请考虑使用 IBM MQ classes for Java 配置文件来收集跟踪信息（请参阅第 344 页的『使用 IBM MQ classes for Java 配置文件收集 IBM MQ classes for Java 跟踪信息』）。通过这种方式启用跟踪时，可以控制 IBM MQ classes for Java 生成的跟踪数据量。

使用 IBM MQ classes for Java 配置文件收集 IBM MQ classes for Java 跟踪信息

如果应用程序必须长时间运行后才会发生问题，应使用 IBM MQ classes for Java 配置文件来收集 IBM MQ classes for Java 跟踪信息。此配置文件允许您指定各种选项来控制收集的跟踪数据量。

关于此任务

要使用 IBM MQ classes for Java 配置文件来收集跟踪信息，请完成以下步骤。

过程

1. 创建 IBM MQ classes for Java 配置文件。
有关此文件的更多信息，请参阅 [IBM MQ classes for Java 配置文件](#)。
2. 编辑 IBM MQ classes for Java 配置文件，以便将属性 **`com.ibm.msg.client.commonservices.trace.status`** 设置为值 ON。
3. 可选：编辑 IBM MQ classes for Java 配置文件 Java Standard Edition 跟踪设置中所列的其他属性。
4. 使用以下命令来运行 IBM MQ classes for Java 应用程序：

```
java -Dcom.ibm.msg.client.config.location=config_file_url
application_name
```

其中，`config_file_url` 是统一资源定位符 (URL)，指定 IBM MQ classes for Java 配置文件的名称和位置。支持以下类型的 URL：http、file、ftp 和 jar。

以下是 Java 命令的一个示例：

```
java -Dcom.ibm.msg.client.config.location=file:/D:/mydir/myJava.config
MyAppClass
```

此命令将 IBM MQ classes for Java 配置文件标识为本地 Windows 系统上的 `D:\mydir\myJava.config` 文件。

缺省情况下，当应用程序启动时，IBM MQ classes for Java 会开始将跟踪信息写入应用程序当前工作目录中的跟踪文件。跟踪文件的名称取决于应用程序的运行环境：

- 对于 IBM MQ classes for Java for IBM MQ 9.0.0 Fix Pack 1 或更低版本，会将跟踪写入名为 `mqjms_%PID%.trc` 的文件。

- 从 IBM MQ 9.0.0 Fix Pack 2 起，如果应用程序已从 JAR 文件 `com.ibm.mq.jar` 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mqjava_%PID%.trc` 的文件。
- 从 IBM MQ 9.0.0 Fix Pack 2 起，如果应用程序已从可重定位的 JAR 文件 `com.ibm.mq.allclient.jar` 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mqjavaclient_%PID%.trc` 的文件。
- **V 9.1.5** **V 9.1.0.5** 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 开始，如果应用程序已从 JAR 文件 `com.ibm.mq.jar` 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mqjava_%PID%.cl%u.trc` 的文件。
- **V 9.1.5** **V 9.1.0.5** 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 开始，如果应用程序已从重新定位的 JAR 文件 `com.ibm.mq.allclient.jar` 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mqjavaclient_%PID%.cl%u.trc` 的文件。

其中，`%PID%` 是正在跟踪的应用程序的进程标识，`%u` 是唯一编号，用于区分在不同 Java 类装入器下运行跟踪的线程之间的文件。

要更改跟踪文件的名称及其写入位置，请确保应用程序使用的 IBM MQ classes for Java 配置文件包含属性 `com.ibm.msg.client.commonservices.trace.outputName` 的条目。该属性值可以是以下任一值：

- 在应用程序的工作目录中创建的跟踪文件的名称。
- 跟踪文件的标准名称，包括创建文件的目录。

例如，要配置 IBM MQ classes for Java 以将应用程序的跟踪信息写入名为 `C:\Trace\trace.trc` 的文件，那么应用程序使用的 IBM MQ classes for Java 配置文件需要包含以下条目：

```
com.ibm.msg.client.commonservices.trace.outputName=C:\Trace\trace.trc
```

使用 *traceControl* 实用程序动态收集 *IBM MQ classes for Java* 跟踪信息

随附于 IBM MQ classes for Java 的 *traceControl* 实用程序允许从正在运行的应用程序收集跟踪信息。如果 IBM 支持人员需要在发生问题后查看来自应用程序的跟踪信息，或者需要从不能停止的关键性应用程序收集跟踪信息，那么此方法十分有用。

关于此任务

有关 *traceControl* 实用程序的更多信息，请参阅第 349 页的『使用 IBM MQ classes for Java 和 IBM MQ classes for JMS 控制运行进程中的跟踪』。

要使用 *traceControl* 实用程序收集跟踪信息，请完成以下步骤。

过程

1. 启动命令提示符，并浏览至 `MQ_INSTALLATION_PATH\java\lib` 目录。
2. 运行以下命令：

```
java -jar com.ibm.mq.traceControl.jar ...
```

此命令会显示系统上所有 Java 进程的列表。

3. 识别需要跟踪的 IBM MQ classes for Java 应用程序的进程标识，并运行以下命令：

```
java -jar com.ibm.mq.traceControl -i process identifier -enable
```

现在已为应用程序开启跟踪。

启用跟踪后，IBM MQ classes for Java 会开始将跟踪信息写入应用程序当前工作目录中的跟踪文件。跟踪文件的名称取决于应用程序的运行环境：

- 对于 IBM MQ classes for Java for IBM MQ 9.0.0 Fix Pack 1 或更低版本，会将跟踪写入名为 `mqjms_%PID%.trc` 的文件。

- 从 IBM MQ 9.0.0 Fix Pack 2 起，如果应用程序已从 JAR 文件 `com.ibm.mq.jar` 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mqjava_%PID%.trc` 的文件。
- 从 IBM MQ 9.0.0 Fix Pack 2 起，如果应用程序已从可重定位的 JAR 文件 `com.ibm.mq.allclient.jar` 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mqjavaclient_%PID%.trc` 的文件。
- **V 9.1.5** **V 9.1.0.5** 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 开始，如果应用程序已从 JAR 文件 `com.ibm.mq.jar` 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mqjava_%PID%.cl%u.trc` 的文件。
- **V 9.1.5** **V 9.1.0.5** 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 开始，如果应用程序已从可重新定位的 JAR 文件 `com.ibm.mq.allclient.jar` 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mqjavaclient_%PID%.cl%u.trc` 的文件。

其中，`%PID%` 是正在跟踪的应用程序的进程标识，`%u` 是唯一编号，用于区分在不同 Java 类装入器下运行跟踪的线程之间的文件。

4. 要关闭跟踪，请运行以下命令：

```
java -jar com.ibm.mq.traceControl -i process identifier -disable
```

跟踪 IBM MQ 资源适配器

`ResourceAdapter` 对象封装 IBM MQ 资源适配器的全局属性。要启用 IBM MQ 资源适配器跟踪，需要在 `ResourceAdapter` 对象中定义属性。

`ResourceAdapter` 对象具有两个属性集：

- 与诊断跟踪相关联的属性
- 与资源适配器管理的连接池相关联的属性

定义这些属性的方式取决于应用程序服务器提供的管理接口。

第 346 页的表 29 列出与诊断跟踪相关联的 `ResourceAdapter` 对象的属性。

| 属性的名称 | 类型 | 缺省值 | 描述 |
|-------------------------------|-----|--------------------|---|
| <code>traceEnabled</code> | 字符串 | <code>false</code> | 启用或禁用诊断跟踪的标志。如果值为 <code>false</code> ，那么关闭跟踪。 |
| <code>traceLevel</code> | 字符串 | 3 | 诊断跟踪中的详细信息级别。值可以在范围 0（不产生跟踪）到 10（提供最多详细信息）之间。请参阅第 346 页的表 30 以获取每个级别的描述。如果启用了跟踪，那么应将 <code>traceLevel</code> 设置为值 10，除非 IBM 支持人员另行指定。 |
| <code>logWriterEnabled</code> | 字符串 | <code>true</code> | 启用或禁用将诊断跟踪发送到应用程序服务器提供的 <code>LogWriter</code> 对象的标志。如果值为 <code>true</code> ，那么跟踪发送到 <code>LogWriter</code> 对象。如果值为 <code>false</code> ，那么不会使用应用程序服务器提供的任何 <code>LogWriter</code> 对象。 |

第 346 页的表 30 描述诊断跟踪的详细信息级别。

| 级别号 | 详细信息级别 |
|-----|--------------|
| 0 | 无跟踪。 |
| 1 | 跟踪包含错误消息。 |
| 3 | 跟踪包含错误和警告消息。 |

表 30: 诊断跟踪的详细信息级别 (继续)

| 级别号 | 详细信息级别 |
|-----|-----------------------------------|
| 6 | 跟踪包含错误、警告和参考消息。 |
| 8 | 跟踪包含错误、警告和参考消息，以及方法的入口和出口信息。 |
| 9 | 跟踪包含错误、警告和参考消息、方法的入口和出口信息，以及诊断数据。 |
| 10 | 跟踪包含所有跟踪信息。 |

注: 该表中未包含的任何级别都与次低级别等效。例如，指定跟踪级别 4 等效于指定跟踪级别 3。但是，在 IBM MQ 资源适配器的未来发行版中可能会使用未包含的级别，因此最好避免使用这些级别。

如果关闭诊断跟踪，那么错误和警告消息写入到系统错误流。如果开启诊断跟踪，那么错误消息写入到系统错误流和跟踪目标，但是警告消息仅写入到跟踪目标。不过，仅当跟踪级别为 3 或更高时，跟踪才包含警告消息。缺省情况下，跟踪目标是当前工作目录，但如果设置了 `logWriterEnabled` 属性，那么跟踪发送到应用程序服务器。

一般而言，ResourceAdapter 对象无需管理。不过，要在诸如 UNIX and Linux 的系统上启用诊断跟踪，可以设置以下属性：

```
traceEnabled: true
traceLevel: 10
```

如果还未启动资源适配器，那么这些属性无效（例如，当使用 IBM MQ 资源的应用程序仅在客户机容器中运行时就会发生这种情况）。在此情况下，可以将诊断跟踪的属性设置为 Java virtual machine (JVM) 系统属性。可以通过在 `java` 命令上使用 `-D` 标志来设置属性，如下列示例所示：

```
java ... -DtraceEnabled=true -DtraceLevel=10
```

提示和技巧

无需定义 ResourceAdapter 对象的所有属性。未指定的任何属性都采用其缺省值。

在受管环境中，最好不要混合使用两种指定属性的方式。如果一定要混用，那么 JVM 系统属性优先于 ResourceAdapter 对象的属性。

在将 WebSphere Application Server traditional 9.0 与 IBM MQ 9.0 资源适配器结合使用时，由于 Java EE 依赖注入现在是常见 Java EE 范例，因此标准跟踪字符串应更新为包含 `com.ibm.ws.cdi.jms*=all`。这意味着完整字符串为：

```
*=info:jmsApi=all:Messaging=all:com.ibm.mq.*=all:JMSApi=all:com.ibm.ws.cdi.jms*=all
```

有关将跟踪与 WebSphere Application Server traditional 配合使用的更多信息，请参阅技术说明：[为 WebSphere Application Server 启用 Java 消息服务 \(JMS\) 跟踪](#)。

跟踪其他 IBM MQ Java 组件

对于 IBM MQ 的 Java 组件，例如 IBM MQ Explorer 和 IBM MQ Transport for SOAP 的 Java 实现，诊断信息是使用标准 IBM MQ 诊断工具或由 Java 诊断类输出的。

此上下文中的诊断信息由跟踪、首次故障数据捕获 (FFDC) 和错误消息组成。

您可以根据情况，选择使用 IBM MQ 设施或者 IBM MQ classes for Java 或 IBM MQ classes for JMS 的设施来产生此信息。如果 IBM MQ 诊断设施在本地系统上可用，那么通常使用这些诊断设施。

在以下情况下，可能要使用 Java 诊断：

- 在提供队列管理器的系统上，如果队列管理器与您运行的软件分开管理。
- 要减少 IBM MQ 跟踪的性能影响。

为请求和配置诊断输出，在启动 IBM MQ Java 进程时使用了两个系统属性：

- System p 属性 `com.ibm.mq.commonservices` 指定标准 Java 属性文件，其中包含用于配置诊断输出的多个行。文件中的每个代码行为自由格式，并以换行符终止。
- System p 属性 `com.ibm.mq.commonservices.diagid` 将跟踪和 FFDC 文件与其创建过程相关联。

有关使用 `com.ibm.mq.commonservices` 属性文件来配置诊断信息的信息，请参阅第 348 页的『使用 `com.ibm.mq.commonservices`』。

有关查找跟踪信息和 FFDC 文件的指示信息，请参阅第 349 页的『Java 跟踪和 FFDC 文件』。

相关概念

第 305 页的『在 UNIX and Linux 系统上使用跟踪』

使用 `strmqtrc` 和 `endmqtrc` 命令来启动和结束跟踪，并使用 `dspmqtrc` 来显示跟踪文件

第 310 页的『将跟踪与 IBM i 上的 IBM MQ 服务器配合使用』

使用 TRCMQM 命令来启动和停止跟踪并指定所需的跟踪类型。

第 318 页的『在 z/OS 上使用跟踪进行问题确定』

有不同的跟踪选项可用于对 IBM MQ 进行问题确定。使用本主题来了解不同选项以及如何控制跟踪。

第 371 页的『跟踪 TLS: `runmqakm`、`strmqikm` 和 `runmqckm` 功能』

如何跟踪传输层安全性 (TLS) 以及请求 `runmqakm` 跟踪与 `strmqikm` (iKeyman) 和 `runmqckm` (iKeycmd) 跟踪。

相关参考

第 315 页的『使用 Windows 上的跟踪』

使用 `strmqtrc` 和 `endmqtrc` 命令或 IBM MQ Explorer 界面来启动和结束跟踪。

使用 `com.ibm.mq.commonservices`

`com.ibm.mq.commonservices` 属性文件包含以下与 IBM MQ 的 Java 组件的诊断输出相关的条目。

请注意，在所有这些条目中大小写都至关重要：

Diagnosics.Java= options

使用 Java 跟踪来跟踪哪些组件。选项是一个或多个 `explorer`、`soap` 和 `wmqjavaclasses`（以逗号分隔），其中“`explorer`”是指来自 IBM MQ Explorer 的诊断，“`soap`”是指来自 IBM MQ Transport for SOAP 中正在运行的进程的诊断，“`wmqjavaclasses`”是指来自底层 IBM MQ Java 类的诊断。缺省情况下，不跟踪任何组件。

Diagnosics.Java.Trace.Detail= high/medium/low

Java 跟踪的详细信息级别。`high` 和 `medium` 详细信息级别与 IBM MQ 跟踪中使用的详细信息级别相匹配，但是 `low` 是 Java 跟踪所特有的。如果诊断，那么将忽略此属性。未设置 Java。缺省值为 `medium`。

Diagnosics.Java.Trace.Destination.File= enabled/disabled

是否将 Java 跟踪写入到文件。如果诊断，那么将忽略此属性。未设置 Java。缺省值为 `disabled`。

Diagnosics.Java.Trace.Destination.Console= enabled/disabled

是否将 Java 跟踪写入到系统控制台。如果诊断，那么将忽略此属性。未设置 Java。缺省值为 `disabled`。

Diagnosics.Java.Trace.Destination.Pathname= dirname

Java 跟踪写入到的目录。如果诊断，那么将忽略此属性。未设置 Java 或未设置诊断。Java。Trace.Destination.File= `disabled`。在 UNIX and Linux 系统上，缺省值为 `/var/mqm/trace`（如果存在），否则缺省值为 Java 控制台 (System.err)。在 Windows 上，缺省值为系统控制台。

Diagnosics.Java.FFDC.Destination.Pathname= dirname

Java FFDC 输出写入到的目录。缺省值为当前工作目录。

Diagnosics.Java.Errors.Destination.Filename= filename

Java 错误消息写入到的标准文件名。缺省值是当前工作目录中的 `AMQJAVA.LOG`。

第 349 页的图 62 中给出了 `com.ibm.mq.commonservices` 属性文件的示例。以井号 (#) 开头的行被视为注释。

```

#
# Diagnostics for MQ Explorer are enabled
#
Diagnostics.wmqexplorer
#
# High detail Java trace
#
Diagnostics.Java.Trace.Detail=high
#
# Java trace is written to a file and not to the console.
#
Diagnostics.Java.Trace.Destination.File=enabled
Diagnostics.Java.Trace.Destination.Console=disabled
#
# Directory for Java trace file
#
Diagnostics.Java.Trace.Destination.Pathname=c:\\tracedir
#
# Directory for First Failure Data Capture
#
Diagnostics.Java.FFDC.Destination.Pathname=c:\\ffdcdir
#
# Directory for error logging
#
Diagnostics.Java.Errors.Destination.Filename=c:\\errorsdir\\SOAPERRORS.LOG
#

```

图 62: *com.ibm.mq.commonservices* 属性文件样本

Java 跟踪和 FFDC 文件

Java 跟踪和 FFDC 文件的文件名约定。

为 IBM MQ Transport for SOAP 生成 Java 跟踪时，会将其写入格式为 *AMQ.diagid.counter.TRC* 的文件名。这里的 *diagid* 是与此 Java 进程相关联的系统属性 *com.ibm.mq.commonservices.diagid* 的值，如本节前面所述，而 *counter* 则是大于或等于 0 的整数。名称中的所有字母均为大写，与用于正常 IBM MQ 跟踪的命名约定相匹配。

如果未指定 *com.ibm.mq.commonservices.diagid*，那么 *diagid* 的值为当前时间，格式为 *YYYYMMDDhhmmssmmm*。

为 IBM MQ Explorer 生成 Java 跟踪时，会将它写入名称格式为 *AMQYYYYMMDDHHmmssmmm.TRC.n* 的文件中。每次运行 IBM MQ Explorer 跟踪时，跟踪工具都会通过将文件后缀 *.n* 递增 1 来重命名所有先前的跟踪文件。然后，跟踪工具会创建一个后缀为 *.0* 的新文件，该文件始终是最新的。

IBM MQ Java 类跟踪文件具有基于等效的 IBM MQ Transport for SOAP Java 跟踪文件的名称。该名称的不同之处在于，在 *.TRC* 字符串之前添加了 *.JC* 字符串，提供的格式为 *AMQ.diagid.counter.JC.TRC*。

为 IBM MQ Explorer 或 IBM MQ Transport for SOAP 生成 Java FFDC 时，会将其写入格式为 *AMQ.diagid.counter.FDC* 的文件名，其中 *diagid* 和 *counter* 如 Java 跟踪文件所述。

IBM MQ Explorer 和 IBM MQ Transport for SOAP 的 Java 错误消息输出将写入相应 Java 进程的 *Diagnostics.Java.Errors.Destination.Filename* 指定的文件。这些文件的格式与标准 IBM MQ 错误日志的格式紧密匹配。

进程在将跟踪信息写入到文件时，在该进程的生命周期内，该信息附加到单个跟踪输出文件。同样，在进程的生命周期内使用单个 FFDC 输出文件。

所有跟踪输出都采用 UTF-8 字符集形式。

使用 IBM MQ classes for Java 和 IBM MQ classes for JMS 控制运行进程中的跟踪

IBM MQ classes for Java 和 IBM MQ classes for JMS 注册标准 MBean，从而允许合适的 Java 管理扩展 (JMX) 工具控制客户机进程的某些方面的跟踪行为。

原则

作为 jconsole 等熟知通用工具的替代方法，您可以使用可执行 JAR 文件形式的命令行工具来访问这些工具。

JAR 文件称为 `com.ibm.mq.traceControl.jar`，存储在 IBM MQ 安装的 `java/lib` 子目录中。有关更多详细信息，请参阅 [IBM MQ classes for JMS 的安装组件](#) 和 [IBM MQ classes for Java 的安装目录](#)。

注：根据配置，JMX 工具可以在本地（与进程在同一系统上）使用，也可以远程使用。最初讨论本地使用的情况。

查找进程

要控制进程，您必须与其建立 JMX 连接。要在本地控制进程，必须指定其标识。

要显示正在运行的 Java 进程及其标识的摘要，请使用选项 `-list` 运行可执行 JAR 文件。此选项将生成找到的进程的标识和描述列表。

检查跟踪状态

在找到相关进程的标识后，使用选项 `-i identifier -status` 运行可执行 JAR 文件，其中 `identifier` 是要更改的进程的标识。这些选项显示进程的状态（`enabled` 或 `disabled`），以及有关进程运行位置、跟踪文件的名称和一个树（表示在跟踪中对包的包含与排除）的信息。

启用和禁用跟踪

要对进程启用跟踪，请使用选项 `-i identifier -enable` 运行可执行 JAR 文件。

要对进程禁用跟踪，请使用选项 `-i identifier -disable` 运行可执行 JAR 文件。

注：只能从集 `-status`，`-enable` 和 `-disable` 中选择一个选项。

包含和排除包

要对进程将一个包包含在跟踪中，请使用选项 `-i identifier -ip package_name` 运行可执行 JAR 文件，其中 `package_name` 是包的名称。

要对进程从跟踪中排除一个包，请使用选项 `-i identifier -ep package_name` 运行可执行 JAR 文件。

注：可以使用多个 `-ip` 和 `-ep` 选项。不会检查这些选项的一致性。

当指定包以进行排除或包含时，具有匹配前缀的包的处理不受影响。例如，从跟踪中排除 `com.ibm.mq.jms` 包将不会排除 `com.ibm.mq`、`com.ibm.msq.client.jms` 或 `com.ibm.mq.remote.api`，但会排除 `com.ibm.mq.jms.internal`。

```
C:>java -jar MQ_INSTALLATION_PATH/java/lib/com.ibm.mq.traceControl.jar -list
10008 : 'MQSample'
9004 : 'MQ_INSTALLATION_PATH/java/lib/com.ibm.mq.traceControl.jar -list'

C:>java -jar MQ_INSTALLATION_PATH/java/lib/com.ibm.mq.traceControl.jar -i 10008 -status
Tracing enabled : false
User Directory : C:\Users\IBM_ADMIN\RTCworkspace\sandpit
Trace File Name : mqjms.trc
Package Include/Exclude tree
root - Included

C:>java -jar MQ_INSTALLATION_PATH/java/lib/com.ibm.mq.traceControl.jar -i 10008 -enable
Enabling trace
Tracing enabled : true

C:>java -jar MQ_INSTALLATION_PATH/java/lib/com.ibm.mq.traceControl.jar -i 10008 -status
Tracing enabled : true
User Directory : C:\Users\IBM_ADMIN\RTCworkspace\sandpit
Trace File Name : mqjms_10008.cl0.trc
Package Include/Exclude tree
root - Included
```

```

C:>java -jar MQ_INSTALLATION_PATH/java/lib/com.ibm.mq.traceControl.jar -i 10008 -ip
com.ibm.mq.jms
Adding 'com.ibm.mq.jms' to the list of packages included in trace

C:>java -jar MQ_INSTALLATION_PATH/java/lib/com.ibm.mq.traceControl.jar -i 10008 -status
Tracing enabled : true
User Directory : C:\Users\IBM_ADMIN\RTCworkspace\sandpit
Trace File Name : mqjms_10008.cl0.trc
Package Include/Exclude tree
root - Included
com - Included
ibm - Included
mq - Included
jms - Included

C:>java -jar MQ_INSTALLATION_PATH/java/lib/com.ibm.mq.traceControl.jar -i 10008 -ip
com.acme.banana -ep com.acme.banana.split -ip com.acme.banana.shake
Adding 'com.acme.banana' to the list of packages included in trace
Adding 'com.acme.banana.shake' to the list of packages included in trace
Adding 'com.acme.banana.split' to the list of packages excluded from trace

C:>java -jar MQ_INSTALLATION_PATH/java/lib/com.ibm.mq.traceControl.jar -i 10008 -status
Tracing enabled : true User Directory : C:\Users\IBM_ADMIN\RTCworkspace\sandpit
Trace File Name : mqjms_10008.cl0.trc
Package Include/Exclude tree
root - Included
com - Included
acme - Included
banana - Included
shake - Included
split - Excluded
ibm - Included
mq - Included
jms - Included

```

包包含-排除树

IBM MQ classes for Java 和 IBM MQ classes for JMS 的跟踪机制通过树结构（从根节点开始）来跟踪包的包含和排除。在树结构中，每个节点表示包名称的一个元素（通过包名称元素识别并包含跟踪状态，可以是 Included 或 Excluded）。例如，包 *com.ibm.mq* 将由通过字符串 *com*、*ibm* 和 *mq* 识别的三个节点来表示。

最初，树通常包含条目以包括大多数包，但是 *header* 和 *pcf* 包由于生成大量噪声而排除在外。因此，初始树将如下所示：

```

root - Included
com - Included
ibm - Included
mq - Included
headers - Excluded
pcf - Excluded

```

当确定跟踪功能要包含还是排除包时，它尽可能将包名称的前导部分与树中的节点进行匹配，并且采用最后一个匹配节点的状态。在树的初始状态下，将包含 *com.ibm.msg.client* 和 *com.ibm.mq.jms* 包，因为它们匹配的树中的最后节点（分别为 *com->ibm* 和 *com->ibm->mq*）被标记为已包含。相反，将排除 *com.ibm.headers.internal* 包，因为树中的最后一个匹配节点 (*com->ibm->mq->headers*) 被标记为已排除。

随着使用 *com.ibm.mq.TraceControl.jar* 对该树进行进一步更改，请务必记住，包含或排除只会影响包和子包。因此，鉴于先前显示的初始状态，指定 *-ep com.ibm.mq.jms* 会将树更新为如下所示：

```

root - Included
com - Included
ibm - Included
mq - Included
headers - Excluded
jms - Excluded
pcf - Excluded

```

此更新将排除 `com.ibm.mq.jms` 和 `com.ibm.mq.jms.internal` 包，而不会影响 `com.ibm.mq.jms.*` 层次结构外的包。

如果接下来指定 `-ip com.ibm.mq.jms.admin`，那么树将如下所示：

```
root - Included
com - Included
ibm - Included
mq - Included
headers - Excluded
jms - Excluded
admin - Included
pcf - Excluded
```

此更新仍将排除 `com.ibm.mq.jms` 和 `com.ibm.mq.jms.internal` 包，但现在会在跟踪中包含 `com.ibm.mq.jms.admin` 和 `com.ibm.mq.jms.admin.internal` 包。

远程连接

仅当通过 JMX 代理（已对远程连接启用并使用

`-Dcom.sun.management.jmxremote.port=port_number` 系统设置）启动了进程时，才能远程连接。

使用此系统设置启动后，可以使用选项 `-h host_name -p port_number` 替代 `-i identifier` 选项来运行可执行 JAR 文件，其中 `host_name` 是要连接到的主机的名称，`port_number` 是要使用的端口的名称。

注：您必须确保采取相应的步骤，通过对连接启用 TLS 来最大程度降低安全风险。请参阅有关 JMX 的 Oracle 文档以获取进一步详细信息 <https://www.oracle.com>。

限制

存在以下局限性：

- 对于非 IBM JVM，工具必须通过将 `tools.jar` 添加到其类路径进行启动。这些平台上的命令为：

```
java -cp MQ_INSTALL_DIR/java/lib/com.ibm.mq.traceControl.jar;JAVA_HOME/lib/tools.jar
com.ibm.msg.client.commonservices.trace.TraceController
```

- 本地连接由用户标识控制。必须使用与要受控的进程相同的标识来运行工具。

Multi 在多平台上跟踪 Managed File Transfer 资源

利用 Managed File Transfer 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。您可以跟踪各种不同的资源。

关于此任务

请参阅

- 第 352 页的『[在多平台上跟踪 Managed File Transfer 代理](#)』，以了解有关如何跟踪代理的信息。
- 第 355 页的『[在多平台上跟踪 Managed File Transfer 命令](#)』，以了解有关如何跟踪命令的信息。

Multi 在多平台上跟踪 Managed File Transfer 代理

利用 Managed File Transfer 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。可使用多个命令和属性来控制此功能的行为。

关于此任务

如果要求您提供跟踪输出来调查代理问题，请根据能否在短时间内停止代理，使用以下选项之一。

如果您不确定要使用的选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果可以在短时间内停止代理，请从启动开始收集代理的跟踪信息。
有关更多信息，请参阅第 353 页的『从启动开始收集 Managed File Transfer 代理跟踪信息』。
- 如果无法停止代理，请使用 **fteSetAgentTraceLevel** 命令来动态收集跟踪信息。
有关更多信息，请参阅第 354 页的『自动收集 Managed File Transfer 代理跟踪信息』。

Multi 从启动开始收集 Managed File Transfer 代理跟踪信息

如果可以在短时间内停止代理，那么应从启动开始收集 Managed File Transfer 代理跟踪信息。

开始之前

您需要在 `agent.properties` 文件中为需要跟踪的代理设置各种属性。

关于此任务

要从启动开始收集跟踪信息，请完成以下步骤。

过程

1. 找到需要跟踪的代理的 `agent.properties` 文件。
可以在 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中找到 `agent.properties` 文件。
2. 编辑文件并为以下属性添加条目：
 - **trace**=跟踪规范
trace 属性将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。
 - **traceFiles**=要使用的跟踪文件数
 - **traceSize**=每个跟踪文件的大小 (MB)
traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。例如，要使用跟踪规范 `com.ibm.wmqfte=all` 来收集 1GB 的合并跟踪，请将以下行添加到 `agent.properties` 文件中：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

这会导致代理将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些代理属性的更多信息，请参阅 [MFT agent.properties 文件](#)。

3. 使用 **fteStopAgent** 命令来停止需要跟踪的代理。
4. 通过运行 **fteStartAgent** 命令来启动代理程序。
5. 重现该问题。
6. 停止代理。
7. 编辑代理程序的 `agent.properties` 文件，并除去您在步骤第 353 页的『2』中添加的 **trace**，**traceFiles** 和 **traceSize** 属性的条目。
这样可确保下次重新启动代理时不会启用跟踪。

结果

生成的跟踪文件将写入到 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目录中，其中 `%PID%` 是代理的进程标识。

Multi 自动收集 Managed File Transfer 代理跟踪信息

使用 `fteSetAgentTraceLevel` 命令，可以从正在运行的代理中收集跟踪信息。如果 IBM 支持人员需要查看无法停止的代理的跟踪信息，这会非常有用。

关于此任务

要使用 `fteSetAgentTraceLevel` 命令从代理中收集跟踪信息，请完成以下步骤。

过程

1. 通过运行以下命令来对代理开启跟踪：

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

`-traceAgent` 参数将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

2. 重现该问题。
3. 通过运行以下命令来关闭代理程序的跟踪：

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

如果代理程序繁忙，那么跟踪文件可能会快速合并并覆盖调查问题所需的信息。如果是这样，请安排一些时间来停止代理程序，然后按照以下步骤中的详细信息继续操作。如果无法在短时间内停止代理程序，请与 IBM 支持代表联系并讨论要使用的替代跟踪规范，以减少正在生成的跟踪数据量。

4. 找到需要跟踪的代理的 `agent.properties` 文件。

可以在 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中找到 `agent.properties` 文件。

5. 编辑文件并为以下属性添加条目：

```
traceFiles=number_of_trace_files_to_use  
traceSize=size_of_each_trace_file_in_MB
```

`traceFiles` 和 `traceSize` 属性用于控制要收集的跟踪数据量。

`traceFiles` 属性的缺省值为 5，`traceSize` 属性的缺省值为 20MB。这意味着，如果您动态开启跟踪并且尚未设置属性，那么代理会将跟踪信息写入到 5 个合并跟踪文件，每个文件的最大大小为 20MB。

您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要收集 1GB 的打包跟踪，请将以下行添加到 `agent.properties` 文件中：

```
traceFiles=5  
traceSize=200
```

这会导致代理将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些代理属性的更多信息，请参阅 `MFT agent.properties` 文件。

6. 通过运行 `fteStopAgent` 命令来停止代理程序。
7. 通过运行 `fteStartAgent` 命令来启动代理程序。
8. 通过运行以下命令，对代理程序启用跟踪：

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

除非您的 IBM 支持代表另有指定，否则请将 `-traceAgent` 属性设置为值 `com.ibm.wmqfte=all`。

9. 重现该问题。
10. 通过运行以下命令在代理程序上关闭跟踪:

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

结果

生成的跟踪文件将写入到 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目录中, 其中 `%PID%` 是代理的进程标识。

Multi 在多平台上跟踪 Managed File Transfer 命令

利用 Managed File Transfer 中的跟踪功能, 可以帮助 IBM 支持人员诊断问题。您可以使用此功能来跟踪命令。

关于此任务



注意: 跟踪命令将仅收集命令所执行的处理的相关信息。它不会跟踪在处理该命令时代理可能执行的任何活动。

过程

1. 启动命令提示符, 并浏览至 `MQ_INSTALLATION_PATH\bin` 目录。
2. 运行以下命令:

Linux UNIX

```
./command_name -trace classes=level -tracePath directory_path command_arguments
```

Windows

```
command_name -trace classes=level -tracePath directory_path command_arguments
```

其中:

- `command_name` 是要跟踪的命令的名称。
- `classes=level` 是要使用的跟踪级别, 以及要启用跟踪的类别。除非您的 IBM 支持代表另有指定, 否则请将此项设置为 `com.ibm.wmqfte=all`。
- `directory_path` 是跟踪文件将写入的目录。
- `command_arguments` 是需要传递给命令的自变量, 例如, 用于 `ftePingAgent` 命令的代理的名称。

结果

生成的跟踪文件将写入到 `-tracePath` 参数指定的目录。

跟踪文件称为 `trace%PID%.txt.number`, 其中:

- `%PID%` 是命令的进程标识。
- `number` 是跟踪文件的序号。通常, 命令生成的跟踪信息包含在序号为 0 的单个跟踪文件中。

但是, 命令可能会生成大量跟踪信息。在这种情况下, 会将跟踪信息写入到多个文件中。当前跟踪文件的序号为 0, 下一个最旧的跟踪文件的序号为 1, 依此类推。

命令的跟踪输出最多写入到 5 个合并跟踪文件中。每个跟踪文件的最大大小为 20MB。

注: 如果运行命令的用户无权写入到 `-tracePath` 参数指定的目录, 那么跟踪输出将写入到标准错误。

示例

在此示例中，将跟踪 **fteListAgents** 命令，并将跟踪信息写入到 C:\trace 目录中：

```
fteListAgents -trace com.ibm.wmqfte=all -tracePath C:\trace
```

在此示例中，将跟踪 **fteCreateTransfer** 命令，并将跟踪信息写入到 /tmp 目录中：

```
fteCreateTransfer -trace com.ibm.wmqfte=all -tracePath /tmp -t text -sa AGENT1  
-da AGENT2 -df /import/transferredfile.txt /export/originalfile.txt
```

写入到 /tmp 的跟踪文件仅包含 **fteCreateTransfer** 命令所执行的处理的相关信息，例如，该命令如何构建要发送到代理的传输请求消息，以及它等待多久后代理会发送回表示已收到该请求的确认。该跟踪文件不包含有关传输本身的任何信息。

Multi 在多平台上跟踪 Managed File Transfer 独立记录器

利用 Managed File Transfer 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。可使用多个命令和属性来控制此功能的行为。

关于此任务

如果要求您提供跟踪输出来调查记录器问题，请根据能否在短时间内停止记录器，使用以下选项之一。

如果您不确定要使用的选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果可以在短时间内停止记录器，请从启动开始收集记录器的跟踪信息。
请参阅第 356 页的『从启动开始收集 Managed File Transfer 独立记录器跟踪信息』。
- 如果无法停止记录器，请使用 **fteSetLoggerTraceLevel** 命令动态收集跟踪信息。
请参阅第 357 页的『动态收集 Managed File Transfer 独立记录器跟踪信息』。

Multi 从启动开始收集 Managed File Transfer 独立记录器跟踪信息

如果可以在短时间内停止记录器，那么应从启动开始收集 Managed File Transfer 记录器跟踪信息。

开始之前

您需要在 `logger.properties` 文件中为需要跟踪的记录器设置各种属性。

关于此任务

要从启动开始收集跟踪信息，请完成以下步骤。

过程

1. 找到需要跟踪的记录器的 `logger.properties` 文件。
可以在 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中找到 `logger.properties` 文件。
2. 编辑文件并为以下属性添加条目：
 - **trace**=跟踪规范
trace 属性将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。
 - **traceFiles**=要使用的跟踪文件数
 - **traceSize**=每个跟踪文件的大小 (MB)

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要使用跟踪规范 `com.ibm.wmqfte=all` 来收集 1GB 的合并跟踪，请将以下行添加到 `logger.properties` 文件中：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

这会导致记录器将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些记录器属性的更多信息，请参阅 `MFT logger.properties` 文件。

3. 使用 **fteStopLogger** 命令来停止需要跟踪的记录器。
4. 通过运行 **fteStartLogger** 命令来启动记录器。
5. 重现该问题。
6. 停止记录器。
7. 编辑记录器的 `logger.properties` 文件，并除去您在步骤 第 356 页的『2』中添加的 **trace**、**traceFiles** 和 **traceSize** 属性的条目。
这样可确保下次重新启动记录器时不会启用跟踪。

结果

生成的跟踪文件将写入到 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目录中，其中 `%PID%` 是记录器的进程标识。

Multi

动态收集 *Managed File Transfer* 独立记录器跟踪信息

您可以使用 **fteSetLoggerTraceLevel** 命令从正在运行的记录器收集跟踪。如果 IBM 支持人员需要查看无法停止的记录器的跟踪信息，这会非常有用。

关于此任务

要使用 **fteSetLoggerTraceLevel** 命令从 *Managed File Transfer* 记录器收集跟踪，请完成以下步骤。

过程

1. 通过运行以下命令，打开记录器的跟踪：

```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

`-traceLogger` 参数将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

2. 重现该问题。
3. 通过运行以下命令，关闭记录器的跟踪：

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

4. 如果记录器繁忙，那么跟踪文件可能会快速合并并覆盖调查该问题所需的信息。
如果可以在短时间内停止记录器，请完成以下步骤以减少收集的跟踪数据量。否则，请联系 IBM 支持人员并讨论替代跟踪规范，以减少收集的跟踪数据量。
 - a) 安排一些时间来停止记录器。
 - b) 找到需要跟踪的记录器的 `logger.properties` 文件。
可以在 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中找到 `logger.properties` 文件。
 - c) 编辑文件并为以下属性添加条目：

traceFiles=number_of_trace_files_to_use
traceSize=size_of_each_trace_file_in_MB

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。

traceFiles 属性的缺省值为 5，**traceSize** 属性的缺省值为 20MB。这意味着，如果您动态开启跟踪并且尚未设置属性，那么代理会将跟踪信息写入到 5 个合并跟踪文件，每个文件的最大大小为 20MB。

您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要收集 1GB 的打包跟踪，请将以下行添加到 `logger.properties` 文件中：

```
traceFiles=5  
traceSize=200
```

这会导致记录器将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些记录器属性的更多信息，请参阅 [MFT logger.properties](#) 文件。

- d) 通过运行 **fteStopLogger** 命令来停止记录器。
- e) 通过运行 **fteStartLogger** 命令来启动记录器。
- f) 通过运行以下命令，打开记录器的跟踪。除非您的 IBM 支持代表另有指定，否则请将 **-traceLogger** 属性设置为值 `com.ibm.wmqfte=all`。

```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

- g) 重现该问题。
- h) 通过运行以下命令，关闭记录器的跟踪：

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

结果

跟踪文件将写入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目录，其中 `%PID%` 是记录器的进程标识。

跟踪 Managed File Transfer for z/OS 资源

利用 Managed File Transfer for z/OS 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。您可以跟踪各种不同的资源。

关于此任务

请参阅

- 第 358 页的『跟踪 Managed File Transfer for z/OS 代理』，以了解有关如何跟踪代理的信息。
- 第 363 页的『跟踪 Managed File Transfer for z/OS 命令』，以了解有关如何跟踪命令的信息。

跟踪 Managed File Transfer for z/OS 代理

利用 Managed File Transfer for z/OS 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。可使用多个命令和属性来控制此功能的行为。

关于此任务

如果要求您提供跟踪输出来调查代理问题，请使用以下选项之一。

如果您不确定要使用的选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果可以在短时间内停止代理，请从启动开始收集代理的跟踪信息。
有关更多信息，请参阅第 359 页的『从启动开始收集 Managed File Transfer for z/OS 代理跟踪信息』。
- 如果无法停止代理，请使用 **fteSetAgentTraceLevel** 命令来动态收集跟踪信息。
有关更多信息，请参阅第 361 页的『动态收集 Managed File Transfer for z/OS 代理跟踪信息』。

z/OS 从启动开始收集 Managed File Transfer for z/OS 代理跟踪信息

如果可以在短时间内停止代理，那么应从启动开始收集 IBM MQ Managed File Transfer 代理跟踪信息。

关于此任务

收集跟踪信息的方法取决于使用 Unix 系统服务 (USS) 还是 JCL 来管理代理。

如果您不确定要使用以下哪个选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果使用的是 z/OS UNIX，请参阅第 359 页的『从启动开始使用 USS 收集代理跟踪信息』。
- 如果您正在使用 JCL，请参阅第 360 页的『从启动开始使用 JCL 收集代理跟踪信息』。

z/OS 从启动开始使用 USS 收集代理跟踪信息

要从启动开始收集使用 UNIX 系统服务 (USS) 管理的 Managed File Transfer for z/OS 代理的跟踪信息，您必须先在该代理的 `agent.properties` 文件中设置各种必需属性，然后再启动该代理。

关于此任务

要从启动开始收集跟踪信息，请完成以下步骤。

过程

1. 找到需要跟踪的代理的 `agent.properties` 文件。

可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中找到 `agent.properties` 文件。

2. 编辑文件并为以下属性添加条目：

- **trace**=跟踪规范

trace 属性将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

- **traceFiles**=要使用的跟踪文件数
- **traceSize**=每个跟踪文件的大小 (MB)

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要使用跟踪规范 `com.ibm.wmqfte=all` 来收集 1GB 的合并跟踪，请将以下行添加到 `agent.properties` 文件中：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

这会导致代理将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些代理属性的更多信息，请参阅 `MFT agent.properties` 文件。

3. 使用 **fteStopAgent** 命令来停止需要跟踪的代理。
4. 通过运行 **fteStartAgent** 命令来启动代理程序。

5. 重现该问题。
6. 停止代理。
7. 编辑代理程序的 `agent.properties` 文件，并除去您在步骤 [第 359 页的『2』](#) 中添加的 **trace**、**traceFiles** 和 **traceSize** 属性的条目。
这样可确保下次重新启动代理时不会启用跟踪。

结果

生成的跟踪文件将写入到 `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目录中，其中 `%PID%` 是代理的进程标识。

z/OS 从启动开始使用 JCL 收集代理跟踪信息

要从启动开始收集使用 JCL 管理的 Managed File Transfer for z/OS 代理的跟踪信息，就需要在该代理启动之前在 `agent.properties` 文件中为其设置所需的各种属性。

关于此任务

要从启动开始收集跟踪信息，请完成以下步骤。

过程

1. 找到需要跟踪的代理的 `agent.properties` 文件。
可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中找到 `agent.properties` 文件。
2. 编辑文件并为以下属性添加条目：
 - **trace**=跟踪规范
trace 属性将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。
 - **traceFiles**=要使用的跟踪文件数
 - **traceSize**=每个跟踪文件的大小 (MB)
traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要使用跟踪规范 `com.ibm.wmqfte=all` 来收集 1GB 的合并跟踪，请将以下行添加到 `agent.properties` 文件中：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

这会导致代理将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些代理属性的更多信息，请参阅 [MFT agent.properties](#) 文件。

3. 找到包含需要跟踪的代理的 JCL 的数据集。
4. 提交数据集中的 BFGAGSP 成员以停止代理。
5. 通过提交数据集中的 BFGAGST 成员来重新启动代理。
6. 重现该问题。
7. 提交数据集中的 BFGAGSP 成员以再次停止代理。
8. 编辑代理程序的 `agent.properties` 文件，并除去您在步骤 [第 360 页的『2』](#) 中添加的 **trace**、**traceFiles** 和 **traceSize** 属性的条目。
这样可确保下次重新启动代理时不会启用跟踪。

结果

生成的跟踪文件将写入到 `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目录中，其中 `%PID%` 是代理的进程标识。

z/OS 动态收集 Managed File Transfer for z/OS 代理跟踪信息

如果无法在短时间内停止代理，那么应动态收集 Managed File Transfer for z/OS 代理跟踪信息。

关于此任务

收集跟踪信息的方法取决于使用 Unix 系统服务 (USS) 还是 JCL 来管理代理。

如果您不确定要使用以下哪个选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果您使用：
 - Unix 系统服务 (USS)，请参阅第 361 页的『使用 z/OS UNIX 动态收集代理跟踪信息』。
 - JCL，请参阅第 362 页的『使用 JCL 动态收集代理跟踪信息』。

z/OS 使用 z/OS UNIX 动态收集代理跟踪信息

在 z/OS UNIX System Services (z/OS UNIX) 下，可以使用 `fteSetAgentTraceLevel` 命令从正在运行的代理程序收集跟踪。如果 IBM 支持人员需要查看无法停止的代理的跟踪信息，这会非常有用。

关于此任务

要使用 `fteSetAgentTraceLevel` 命令从 Managed File Transfer for z/OS 代理程序收集跟踪，请完成以下步骤。

过程

1. 通过运行以下命令，打开代理程序的跟踪：

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

`-traceAgent` 参数将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

2. 重现该问题。
3. 通过运行以下命令，关闭代理程序的跟踪：

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

4. 如果代理程序繁忙，那么跟踪文件可能会快速合并并覆盖调查问题所需的信息。如果可以在短时间内停止代理程序，请完成以下步骤以减少收集的跟踪数据量。否则，请联系 IBM 支持人员并讨论替代跟踪规范，以减少收集的跟踪数据量。

- a) 安排一些时间来停止代理。
- b) 找到需要跟踪的代理的 `agent.properties` 文件。

可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中找到 `agent.properties` 文件。

- c) 编辑文件并为以下属性添加条目：

```
traceFiles=number_of_trace_files_to_use  
traceSize=size_of_each_trace_file_in_MB
```

`traceFiles` 和 `traceSize` 属性用于控制要收集的跟踪数据量。

traceFiles 属性的缺省值为 5，**traceSize** 属性的缺省值为 20MB。这意味着，如果您动态开启跟踪并且尚未设置属性，那么代理会将跟踪信息写入到 5 个合并跟踪文件，每个文件的最大大小为 20MB。

您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要收集 1GB 的打包跟踪，请将以下行添加到 `agent.properties` 文件中：

```
traceFiles=5
traceSize=200
```

这会导致代理将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些代理属性的更多信息，请参阅 [MFT agent.properties](#) 文件。

- d) 通过运行 **fteStopAgent** 命令来停止代理程序。
- e) 通过运行 **fteStartAgent** 命令来启动代理程序。
- f) 通过运行以下命令，打开代理程序的跟踪：

```
fteSetAgentTraceLevel -traceAgent trace_specification agent_name
```

- g) 重现该问题。
- h) 通过运行以下命令，关闭代理程序的跟踪：

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

结果

跟踪文件将写入 `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目录，其中 `%PID%` 是代理程序的进程标识。

使用 JCL 动态收集代理跟踪信息

对于需要跟踪的代理，您可以使用包含 JCL 的数据集中的 BFGAGTC 成员，从正在运行的 Managed File Transfer for z/OS 代理中收集跟踪信息。如果 IBM 支持人员需要查看无法停止的代理的跟踪信息，这会非常有用。

关于此任务

要使用 BFGAGTC 成员从代理中收集跟踪信息，请完成以下步骤。

过程

1. 找到包含需要跟踪的代理的 JCL 的数据集。
2. 编辑数据集中的 BFGAGTC 成员，然后找到包含以下文本的行：

```
-traceAgent
```

其下方的文本包含要跟踪的内部类和包的列表。缺省情况下，将此列表设置为：

```
com.ibm.wmqfte=all
```

除非您的 IBM 支持代表另有指定，否则请按原样保留此值。

3. 提交 BFGAGTC 成员。
4. 重现该问题。
5. 再次编辑 BFGAGTC 成员，然后将 **-traceAgent** 参数设置为 `=off`，如下所示：

```
-traceAgent =off +
```

6. 重新提交 BFGAGTC 成员以关闭跟踪。
7. 如果代理繁忙，那么跟踪文件可能会迅速合并和覆盖调查此问题所需的信息。

在此情况下，可以使用以下两个选项：

a) 第一个选项是：

- i) 安排一些时间来停止代理。
- ii) 找到需要跟踪的代理的 `agent.properties` 文件。可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中找到 `agent.properties` 文件。
- iii) 编辑文件并为以下属性添加条目：

- **traceFiles**=要使用的跟踪文件数
- **traceSize**=每个跟踪文件的大小 (MB)

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。

traceFiles 属性的缺省值为 5，**traceSize** 属性的缺省值为 20MB。这意味着，如果您动态开启跟踪并且尚未设置属性，那么代理会将跟踪信息写入到 5 个合并跟踪文件，每个文件的最大大小为 20MB。

您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要收集 1GB 的打包跟踪，请将以下行添加到 `agent.properties` 文件中：

```
traceFiles=5
traceSize=200
```

这会导致代理将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些代理属性的更多信息，请参阅 [MFT agent.properties](#) 文件。

- iv) 找到包含需要跟踪的代理的 JCL 的数据集。
- v) 提交数据集中的 BFGAGSP 成员以停止代理。
- vi) 通过提交数据集中的 BFGAGST 成员来重新启动代理。
- vii) 编辑数据集中的 BFGAGTC 成员，然后找到包含以下文本的行：

```
-traceAgent
```

其下方的文本包含要跟踪的内部类和包的列表。缺省情况下，将此列表设置为：

```
com.ibm.wmqfte=all
```

除非您的 IBM 支持代表另有指定，否则请按原样保留此值。

- viii) 当需要启用跟踪时，请提交 BFGAGTC 成员。
- ix) 重现该问题。
- x) 再次编辑 BFGAGTC 成员，然后将 **-traceAgent** 参数设置为 `=off`，如下所示：

```
-traceAgent =off +
```

- xi) 重新提交 BFGAGTC 成员以关闭跟踪。

b) 第二个选项是联系您的 IBM 支持代表（如果无法在短时间内停止代理）。

然后，您可以讨论要使用的备用跟踪规范，以减少要生成的跟踪数据量。

结果

生成的跟踪文件将写入到 `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目录中，其中 `%PID%` 是代理的进程标识。

跟踪 Managed File Transfer for z/OS 命令

利用 Managed File Transfer for z/OS 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。您可以使用此功能来跟踪命令。

关于此任务



注意: 跟踪命令将仅收集命令所执行的处理的相关信息。它不会跟踪在处理该命令时代理可能执行的任何活动。

收集跟踪信息的方法取决于是使用 Unix 系统服务 (USS) 还是 JCL 来运行命令。

如果您不确定要使用的选项, 请联系 IBM 支持代表, 他们将提出最佳方式建议, 帮助您为所发现的问题收集跟踪信息。

过程

- 如果使用的是 z/OS UNIX, 请参阅第 359 页的『从启动开始使用 USS 收集代理跟踪信息』。
- 如果您正在使用 JCL, 请参阅第 360 页的『从启动开始使用 JCL 收集代理跟踪信息』。

z/OS 使用 USS 收集命令的跟踪信息

要使用 Unix 系统服务 (USS) 收集 Managed File Transfer for z/OS 命令的跟踪信息, 请执行以下过程。

过程

1. 启动命令提示符, 并浏览至 `BFG_PROD/bin` 目录。
2. 运行以下命令:

```
./command_name -trace classes=level -tracePath directory_path command_arguments
```

其中:

- `command_name` 是要跟踪的命令的名称。
- `classes=level` 是要使用的跟踪级别, 以及要启用跟踪的类别。除非您的 IBM 支持代表另有指定, 否则请将此项设置为 `com.ibm.wmqfte=all`。
- `directory_path` 是跟踪文件将写入的目录。
- `command_arguments` 是需要传递给命令的自变量, 例如, 用于 `ftePingAgent` 命令的代理的名称。

结果

生成的跟踪文件将写入到 `-tracePath` 参数指定的目录。

跟踪文件称为 `trace%PID%.txt.number`, 其中:

- `%PID%` 是命令的进程标识。
- `number` 是跟踪文件的序号。通常, 命令生成的跟踪信息包含在序号为 0 的单个跟踪文件中。

但是, 命令可能会生成大量跟踪信息。在这种情况下, 会将跟踪信息写入到多个文件中。当前跟踪文件的序号为 0, 下一个最旧的跟踪文件的序号为 1, 依此类推。

命令的跟踪输出最多写入到 5 个合并跟踪文件中。每个跟踪文件的最大大小为 20MB。

注: 如果运行命令的用户无权写入到 `-tracePath` 参数指定的目录, 那么跟踪输出将写入到标准错误。

示例

在此示例中, 将跟踪 `fteListAgents` 命令, 并将跟踪信息写入到 `/u/fteuser` 目录中:

```
./fteListAgents -trace com.ibm.wmqfte=all -tracePath /u/fteuser
```

在此示例中, 将跟踪 `fteCreateTransfer` 命令, 并将跟踪信息写入到 `/tmp` 目录中:

```
./fteCreateTransfer -trace com.ibm.wmqfte=all -tracePath /tmp -t text -sa AGENT1  
-da AGENT2 -df /tmp/IEEUJV.txt "'SYS1.SAMPLIB(IEEUJV)'"
```

写入到 /tmp 的跟踪文件仅包含 **fteCreateTransfer** 命令所执行的处理的相关信息，例如，该命令如何构建要发送到代理的传输请求消息，以及它等待多久后代理会发送回表示已收到该请求的确认。该跟踪文件不包含有关传输本身的任何信息。

z/OS 使用 JCL 收集命令的跟踪信息

要收集使用 JCL 提交的 Managed File Transfer for z/OS 命令的跟踪信息，需要完成以下步骤。

过程

1. 找到包含需要跟踪的命令的 JCL 的数据集。
2. 在该数据集中，找到该命令的成员。
3. 编辑该成员，并找到包含需要跟踪的命令名称的行。修改以下行，使其包含命令名称后面和 + 符号前面的文本：

```
-trace classes=level -tracePath directory_path
```

其中：

- *classes=level* 是要使用的跟踪级别，以及要启用跟踪的类别。除非您的 IBM 支持代表另有指定，否则请将此项设置为 `com.ibm.wmqfte=all`。
 - *directory_path* 是要将跟踪文件写入到的 USS 目录。
4. 提交该成员。
 5. 重现该问题后，再次编辑此成员，并去掉文本：

```
-trace classes=level -tracePath directory_path
```

这是您在步骤 [第 365 页的『3』](#) 中添加的内容。

结果

生成的跟踪文件将写入到 **-tracePath** 参数指定的目录。

跟踪文件称为 `trace%PID%.txt.number`，其中：

- *%PID%* 是命令的进程标识。
- *number* 是跟踪文件的序号。通常，命令生成的跟踪信息包含在序号为 0 的单个跟踪文件中。

但是，命令可能会生成大量跟踪信息。在这种情况下，会将跟踪信息写入到多个文件中。当前跟踪文件的序号为 0，下一个最旧的跟踪文件的序号为 1，依此类推。

命令的跟踪输出最多写入到 5 个合并跟踪文件中。每个跟踪文件的最大大小为 20MB。

注：如果运行命令的用户无权写入到 **-tracePath** 参数指定的目录，那么跟踪输出将写入到标准错误。

示例

在此示例中，已修改成员 BFGMNL1 以跟踪 **fteListMonitors** 命令：

```
/******  
/* <copyright  
/* notice="lm-source"  
/* pids="5655-MF9"  
/* years="2013,2016"  
/* crc="3927276320" >  
/* Licensed Materials - Property of IBM  
/*  
/* 5655-MF9  
/*  
/* (C) Copyright IBM Corp. 2013, 2022. All Rights Reserved.  
/* </copyright>  
/******  
/* fteListMonitors  
/******  
//BFGCMD EXEC PGM=IKJEFT01,REGION=0M  
//SYSEXEC DD DSN=++LIBRARY++,DISP=SHR
```

```
//SYSTSPRT DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSTSIN  DD *
  %BFGCMD CMD=fteListMonitors -trace com.ibm.wmqfte=all -tracePath /u/trace +
  -v -p QM1
/*
//
```

提交该成员后，**fteListMonitors** 命令会将跟踪信息写入到 USS 目录 /u/trace。

z/OS 跟踪 Managed File Transfer for z/OS 独立数据库记录器

利用 Managed File Transfer for z/OS 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。可使用多个命令和属性来控制此功能的行为。

关于此任务

如果要求您提供跟踪输出来调查独立数据库记录器问题，请使用以下选项之一。

如果您不确定要使用的选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果可以在短时间内停止记录器，请从启动开始收集记录器的跟踪信息。
有关更多信息，请参阅第 366 页的『从启动开始收集 Managed File Transfer for z/OS 独立数据库记录器跟踪信息』。
- 如果无法停止记录器，请使用 **fteSetLoggerTraceLevel** 命令动态收集跟踪信息。
有关更多信息，请参阅第 368 页的『动态收集 Managed File Transfer for z/OS 独立数据库记录器跟踪信息』。

z/OS 从启动开始收集 Managed File Transfer for z/OS 独立数据库记录器跟踪信息

如果可以在短时间内停止记录器，那么应从启动开始收集 IBM MQ Managed File Transfer 记录器跟踪信息。

关于此任务

收集跟踪信息的方法取决于是使用 UNIX 系统服务 (USS) 还是 JCL 来管理记录器。

如果您不确定要使用以下哪个选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果您使用：
 - UNIX 系统服务 (USS)，请参阅第 366 页的『使用 USS 从启动开始收集独立数据库记录器跟踪信息』。
 - JCL，请参阅第 367 页的『使用 JCL 从启动开始收集独立数据库记录器跟踪信息』。

z/OS 使用 USS 从启动开始收集独立数据库记录器跟踪信息

要从启动开始收集使用 UNIX 系统服务 (USS) 管理的 Managed File Transfer for z/OS 记录器的跟踪信息，需要在此记录器的 `logger.properties` 文件中设置各种属性，然后再启动记录器。

关于此任务

要从启动开始收集跟踪信息，请完成以下步骤。

过程

1. 找到需要跟踪的记录器的 `logger.properties` 文件。

可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中找到 `logger.properties` 文件。

2. 编辑文件并为以下属性添加条目：

- **trace**=跟踪规范

trace 属性将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

- **traceFiles**=要使用的跟踪文件数

- **traceSize**=每个跟踪文件的大小 (MB)

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要使用跟踪规范 `com.ibm.wmqfte=all` 来收集 1GB 的合并跟踪，请将以下行添加到 `logger.properties` 文件中：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

这会导致记录器将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些记录器属性的更多信息，请参阅 [MFT logger.properties 文件](#)。

3. 使用 **fteStopLogger** 命令来停止需要跟踪的记录器。
4. 通过运行 **fteStartLogger** 命令来启动记录器。
5. 重现该问题。
6. 停止记录器。
7. 编辑记录器的 `logger.properties` 文件，并除去您在步骤 [第 367 页的『2』](#) 中添加的 **trace**，**traceFiles** 和 **traceSize** 属性的条目。
这样可确保下次重新启动记录器时不会启用跟踪。

结果

生成的跟踪文件将写入到 `BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目录，其中 `%PID%` 是记录器的进程标识。

 使用 JCL 从启动开始收集独立数据库记录器跟踪信息

要从启动开始收集使用 JCL 管理的 Managed File Transfer for z/OS 记录器的跟踪信息，就需要在该记录器启动之前在 `logger.properties` 文件中为其设置各种属性。

关于此任务

要从启动开始收集跟踪信息，请完成以下步骤。

过程

1. 找到需要跟踪的记录器的 `logger.properties` 文件。

可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中找到 `logger.properties` 文件。

2. 编辑文件并为以下属性添加条目：

- **trace**=跟踪规范

trace 属性将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

- **traceFiles**=要使用的跟踪文件数
 - **traceSize**=每个跟踪文件的大小 (MB)
- traceFiles** 和 **traceSize** 属性用于控制要收集的跟踪数据量。您应将这些属性设置为较大的值, 以尽可能多地收集跟踪数据。

例如, 要使用跟踪规范 `com.ibm.wmqfte=all` 来收集 1GB 的合并跟踪, 请将以下行添加到 `logger.properties` 文件中:

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

这会导致记录器将跟踪数据写入到最多 5 个文件中, 每个文件的大小都为 200MB。

有关这些记录器属性的更多信息, 请参阅 [MFT logger.properties](#) 文件。

3. 找到包含需要跟踪的记录器的 JCL 的数据集。
4. 提交数据集中的 BFGLGSP 成员以停止记录器。
5. 通过提交数据集中的 BFGLGST 成员来重新启动记录器。
6. 重现该问题。
7. 提交数据集中的 BFGLGSP 成员以再次停止记录器。
8. 编辑记录器的 `logger.properties` 文件, 并除去您在步骤 [第 367 页的『2』](#) 中添加的 **trace**, **traceFiles** 和 **traceSize** 属性的条目。
这样可确保下次重新启动记录器时不会启用跟踪。

结果

生成的跟踪文件将写入到 `BFQ_DATA/mqft/logs/coordination_qmgr_name/logger/logger_name/logs/trace%PID%` 目录, 其中 `%PID%` 是记录器的进程标识。

动态收集 Managed File Transfer for z/OS 独立数据库记录器跟踪信息

如果无法在短时间内停止记录器, 那么应动态收集 Managed File Transfer for z/OS 记录器跟踪信息。

关于此任务

收集跟踪信息的方法取决于是使用 UNIX 系统服务 (USS) 还是 JCL 来管理记录器。

如果您不确定要使用以下哪个选项, 请联系 IBM 支持代表, 他们将提出最佳方式建议, 帮助您为所发现的问题收集跟踪信息。

过程

- 如果您使用:
 - Unix 系统服务 (USS), 请参阅 [第 368 页的『使用 z/OS UNIX 动态收集独立数据库记录器跟踪信息』](#)。
 - JCL, 请参阅 [第 369 页的『使用 JCL 动态收集独立数据库记录器跟踪信息』](#)。

使用 z/OS UNIX 动态收集独立数据库记录器跟踪信息

在 z/OS UNIX System Services (z/OS UNIX) 下, 您可以使用 **fteSetLoggerTraceLevel** 命令从正在运行的记录器收集跟踪信息。如果 IBM 支持人员需要查看无法停止的记录器的跟踪信息, 这会非常有用。

关于此任务

要使用 **fteSetLoggerTraceLevel** 命令从 Managed File Transfer for z/OS 记录器收集跟踪, 请完成以下步骤。

过程

1. 通过运行以下命令, 打开记录器的跟踪:


```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

-traceLogger 参数将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

2. 重现该问题。
3. 通过运行以下命令，关闭记录器的跟踪：

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

4. 如果记录器繁忙，那么跟踪文件可能会快速合并并覆盖调查该问题所需的信息。

如果可以在短时间内停止记录器，请完成以下步骤以减少收集的跟踪数据量。否则，请联系 IBM 支持人员并讨论替代跟踪规范，以减少收集的跟踪数据量。

- a) 安排一些时间来停止记录器。
- b) 找到需要跟踪的记录器的 `logger.properties` 文件。

可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中找到 `logger.properties` 文件。

- c) 编辑文件并为以下属性添加条目：

traceFiles=number_of_trace_files_to_use

traceSize=size_of_each_trace_file_in_MB

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。

traceFiles 属性的缺省值为 5，**traceSize** 属性的缺省值为 20MB。这意味着，如果您动态开启跟踪并且尚未设置属性，那么代理会将跟踪信息写入到 5 个合并跟踪文件，每个文件的最大大小为 20MB。

您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要收集 1GB 的打包跟踪，请将以下行添加到 `logger.properties` 文件中：

```
traceFiles=5  
traceSize=200
```

这会导致记录器将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些记录器属性的更多信息，请参阅 [MFT logger.properties](#) 文件。

- d) 通过运行 **fteStopLogger** 命令来停止记录器。
- e) 通过运行 **fteStartLogger** 命令来启动记录器。
- f) 通过运行以下命令，打开记录器的跟踪：

```
fteSetLoggerTraceLevel -traceLogger trace_specification logger_name
```

- g) 重现该问题。
- h) 通过运行以下命令，关闭记录器的跟踪：

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

结果

跟踪文件将写入 `BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目录，其中 `%PID%` 是记录器的进程标识。

使用 JCL 动态收集独立数据库记录器跟踪信息

对于需要跟踪的记录器，您可以使用包含 JCL 的数据集中的 BFGLGTC 成员，从正在运行的 Managed File Transfer for z/OS 记录器收集跟踪信息。如果 IBM 支持人员需要查看无法停止的记录器的跟踪信息，这会非常有用。

关于此任务

要使用 BFGLGTC 成员从记录器收集跟踪信息，请完成以下步骤。

过程

1. 找到包含需要跟踪的记录器的 JCL 的数据集。
2. 编辑数据集中的 BFGLGTC 成员，然后找到包含以下文本的行：

```
-traceLogger
```

其下方的文本包含要跟踪的内部类和包的列表。缺省情况下，将此列表设置为：

```
com.ibm.wmqfte=all
```

除非您的 IBM 支持代表另有指定，否则请按原样保留此值。

3. 提交 BFGLGTC 成员。
4. 重现该问题。
5. 再次编辑 BFGLGTC 成员，并将 **-traceLogger** 参数设置为 `=off`，如下所示：

```
-traceLogger =off +
```

6. 重新提交 BFGLGTC 成员以关闭跟踪。
7. 如果记录器繁忙，那么跟踪文件可能会快速合并并覆盖调查该问题所需的信息。

如果可以在短时间内停止记录器，请完成以下步骤以减少收集的跟踪数据量。否则，请联系 IBM 支持人员并讨论替代跟踪规范，以减少收集的跟踪数据量。

- a) 安排一些时间来停止记录器。
- b) 找到需要跟踪的记录器的 `logger.properties` 文件。

可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中找到 `logger.properties` 文件。

- c) 编辑文件并为以下属性添加条目：

```
traceFiles=number_of_trace_files_to_use
```

```
traceSize=size_of_each_trace_file_in_MB
```

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。

traceFiles 属性的缺省值为 5，**traceSize** 属性的缺省值为 20MB。这意味着，如果您动态开启跟踪并且尚未设置属性，那么代理会将跟踪信息写入到 5 个合并跟踪文件，每个文件的最大大小为 20MB。

您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要收集 1GB 的打包跟踪，请将以下行添加到 `logger.properties` 文件中：

```
traceFiles=5  
traceSize=200
```

这会导致记录器将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些记录器属性的更多信息，请参阅 [MFT logger.properties 文件](#)。

- d) 找到包含需要跟踪的记录器的 JCL 的数据集。
- e) 提交数据集中的 BFGLGSP 成员以停止记录器。
- f) 通过提交数据集中的 BFGLGST 成员来重新启动记录器。
- g) 编辑数据集中的 BFGLGTC 成员，并找到包含以下文本的行：

```
-traceLogger
```

其下方的文本包含要跟踪的内部类和包的列表。缺省情况下，将此列表设置为：

```
com.ibm.wmqfte=all
```

除非您的 IBM 支持代表另有指定，否则请按原样保留此值。

- h) 当需要启用跟踪时，请提交 BFGLGTC 成员。
- i) 重现该问题。
- j) 再次编辑 BFGLGTC 成员，并通过运行以下命令将 **-traceLogger** 参数设置为 **= off**：

```
-traceLogger =off +
```

- k) 重新提交 BFGLGTC 成员以关闭跟踪。

结果

跟踪文件将写入 `BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目录，其中 `%PID%` 是记录器的进程标识。

跟踪 TLS: `runmqakm`、`strmqikm` 和 `runmqckm` 功能

如何跟踪传输层安全性 (TLS) 以及请求 `runmqakm` 跟踪与 `strmqikm` (iKeyman) 和 `runmqckm` (iKeycmd) 跟踪。

`strmqikm` 和 `runmqckm` 跟踪

要请求 `strmqikm` 跟踪，请对平台运行带以下 **-D** 标志的 `strmqikm` 命令。

在 UNIX, Linux, and Windows 上：

```
strmqikm -Dkeyman.debug=true -Dkeyman.jnitracng=ON
```

要请求 `runmqckm` 跟踪，请对平台运行带以下 **-D** 标志的 `runmqckm` 命令。

在 UNIX, Linux, and Windows 上：

```
runmqckm -Dkeyman.debug=true -Dkeyman.jnitracng=ON
```

`strmqikm` 和 `runmqckm` 将三个跟踪文件写入从中启动它们的目录，因此请考虑从运行时 TLS 跟踪所写入的跟踪目录启动 iKeyman 或 `runmqckm`：`/var/mqm/trace` on UNIX and Linux systems and `MQ_INSTALLATION_PATH/trace` on Windows。 `MQ_INSTALLATION_PATH` 表示 IBM MQ 安装所在的高级目录。

`strmqikm` 和 `runmqckm` 生成的跟踪文件具有以下格式：

```
debugTrace. n
```

其中，`n` 是从 0 开始的递增数字。

`runmqakm` 跟踪

要请求 `runmqakm` 跟踪，请执行具有以下标志的 `runmqakm` 命令：

```
runmqakm -trace filename
```

其中 `filename` 是要创建的跟踪文件的名称。您无法格式化 `runmqakm` 跟踪文件。请勿更改该文件并将其发送给 IBM 支持人员。`runmqakm` 跟踪文件是二进制文件，如果通过 FTP 将其传输给 IBM 支持人员，那么必须以二进制传输方式进行传输。

运行时 TLS 跟踪

在 UNIX, Linux, and Windows 系统上, 可以独立请求 **strmqikm**、**runmqckm**、运行时 TLS 功能或这些内容组合的跟踪信息。

运行时 TLS 跟踪文件具有名称 AMQ.TLS.TRC 和 AMQ.TLS.TRC.1, 而 TLS 跟踪文件具有名称 AMQ.SSL.TRC 和 AMQ.SSL.TRC.1。您无法格式化任何 TLS 跟踪文件; 请不要更改这些文件并将其发送给 IBM 支持人员。TLS 跟踪文件是二进制文件, 如果通过 FTP 将其传输给 IBM 支持人员, 那么必须以二进制传输方式进行传输。

相关概念

第 305 页的『在 UNIX and Linux 系统上使用跟踪』

使用 **strmqtrc** 和 **endmqtrc** 命令来启动和结束跟踪, 并使用 **dspmqtrc** 来显示跟踪文件

第 310 页的『将跟踪与 IBM i 上的 IBM MQ 服务器配合使用』

使用 TRCMQM 命令来启动和停止跟踪并指定所需的跟踪类型。

第 318 页的『在 z/OS 上使用跟踪进行问题确定』

有不同的跟踪选项可用于对 IBM MQ 进行问题确定。使用本主题来了解不同选项以及如何控制跟踪。

第 347 页的『跟踪其他 IBM MQ Java 组件』

对于 IBM MQ 的 Java 组件, 例如 IBM MQ Explorer 和 IBM MQ Transport for SOAP 的 Java 实现, 诊断信息是使用标准 IBM MQ 诊断工具或由 Java 诊断类输出的。

相关参考

第 315 页的『使用 Windows 上的跟踪』

使用 **strmqtrc** 和 **endmqtrc** 命令或 IBM MQ Explorer 界面来启动和结束跟踪。

跟踪 IBM MQ 的 WCF 定制通道

您可以使用 IBM MQ 跟踪收集有关不同的 IBM MQ 代码部分有何作用的详细信息。使用 Windows Communication Foundation (WCF) 时, 将为与 Microsoft WCF 基础结构跟踪集成的 Microsoft Windows Communication Foundation (WCF) 定制通道跟踪生成单独的跟踪输出。

关于此任务

为 WCF 定制通道完全启用跟踪会生成两个输出文件:

1. 与 Microsoft WCF 基础结构跟踪集成的 WCF 定制通道跟踪。
2. 与 XMS .NET 集成的 WCF 定制通道跟踪。

通过具有两个跟踪输出, 可以使用适当的工具跟踪每个接口的问题, 例如:

- 使用合适 Microsoft 工具的 WCF 问题确定。
- 使用 XMS 跟踪格式的 IBM MQ MQI client 问题。

为了简化跟踪启用, .NET TraceSource 和 XMS .NET 跟踪堆栈都使用单个接口进行控制。

有两个选项可用于为非 SOAP/Non-JMS 接口配置 WCF 跟踪。您可以编程方式或通过环境变量来配置跟踪。

过程

要对非 SOAP/Non-JMS 接口启用 WCF 跟踪, 请选择下列其中一个选项:

- 通过将 **WMQ_TRACE_ON** 设置为环境变量, 通过环境变量配置跟踪。
- 通过将以下代码部分添加到 app.config 文件中的 <system.diagnostics><sources> 部分, 以编程方式配置跟踪

```
<source name="IBM.WMQ.WCF" switchValue="Verbose, ActivityTracing"
xmsTraceSpecification="*=all=enabled"
xmsTraceFileSize="2000000" xmsTraceFileNumber="4"
xmsTraceFormat="advanced">
</source>
```

相关概念

第 303 页的『[WCF XMS First Failure Support Technology \(FFST\)](#)』

您可以通过使用 IBM MQ 跟踪收集有关不同的 IBM MQ 代码部分有何作用的详细信息。XMS FFST 针对 WCF 定制通道具有自己的配置和输出文件。

相关任务

第 176 页的『[对 IBM MQ 问题的 WCF 定制通道进行故障诊断](#)』

第 227 页的『[联系 IBM 支持人员](#)』

如果您在 IBM MQ 方面遇到问题需要帮助，可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订、故障诊断和其他新闻的通知。

[使用 IBM MQ 开发 Microsoft Windows Communication Foundation 应用程序](#)

为 XMS .NET 应用程序配置跟踪

如果使用 IBM MQ classes for XMS .NET Framework，那么可以通过应用程序配置文件以及通过 XMS 环境变量来配置跟踪。如果使用 IBM MQ classes for XMS .NET Standard，那么必须通过 XMS 环境变量来配置跟踪。您可以选择要跟踪的组件。通常在 IBM 支持人员的指导下使用跟踪。

关于此任务

XMS .NET 的跟踪基于标准 .NET 跟踪基础结构。

缺省情况下，将禁用除错误跟踪以外的所有其他跟踪。

如果使用 IBM MQ classes for XMS .NET Framework，那么可以使用以下任一方式来开启跟踪和配置跟踪设置：

- 使用其名称包含相关可执行程序名称且后缀为 `.config` 的应用程序配置文件。例如，`text.exe` 的应用程序配置文件将具有名称 `text.exe.config`。使用应用程序配置文件是为 XMS .NET 应用程序启用跟踪的首选方法。要获取更多详细信息，请参阅第 374 页的『[使用应用程序配置文件配置 XMS .NET 跟踪](#)』。
- 对 XMS C 或 C++ 应用程序使用 XMS 环境变量。要获取更多详细信息，请参阅第 375 页的『[使用 XMS 环境变量配置 XMS .NET 跟踪](#)』。

V9.1.1 如果使用 IBM MQ classes for XMS .NET Standard，那么必须通过 XMS 环境变量来配置跟踪。要获取更多详细信息，请参阅第 375 页的『[使用 XMS 环境变量配置 XMS .NET 跟踪](#)』。IBM MQ classes for XMS .NET Standard 不支持使用应用程序配置文件。

活动跟踪文件的名称格式为 `xms_tracePID.log`，其中 `PID` 表示应用程序的进程标识。缺省情况下，活动跟踪文件的大小限制为 20 MB。达到此限制后，将对该文件进行重命名和归档。归档文件的名称格式为 `xms_tracePID_YY.MM.DD_HH.MM.SS.log`。

缺省情况下，保留的跟踪文件数为 4（1 个活动文件和 3 个归档文件）。这 4 个文件将用作滚动缓冲区直至应用程序停止，最早的文件将被移除并替换为最新的文件。您可以通过在应用程序配置文件中指定不同的数目来更改跟踪文件的数目。但是，必须至少存在 2 个文件（1 个活动文件和 1 个归档文件）。

有两种可用的跟踪文件格式：

- 基本格式的跟踪文件是人类可读的文件，采用 WebSphere Application Server 格式。此格式是缺省跟踪文件格式。基本格式与跟踪分析器工具不兼容。
- 高级格式的跟踪文件与跟踪分析器工具兼容。您必须指定要在应用程序配置文件中采用高级格式生成跟踪文件。

跟踪条目包含以下信息：

- 记录跟踪的日期和时间
- 类名
- 跟踪类型
- 跟踪消息

以下示例显示了来自某个跟踪的摘录：

```
[09/11/2005 14:33:46:914276]    00000004    IBM.XMS.Comms.IoRequest    >    Allocate    Entry
[09/11/2005 14:33:46:914276]    00000004    IBM.XMS.Comms.IoRequest    >    Initialize    Entry
[09/11/2005 14:33:46:914276]    00000004    IBM.XMS.Comms.IoRequest    <    Initialize    Exit
[09/11/2005 14:33:46:914276]    00000004    IBM.XMS.Comms.IoRequest    <    Allocate    Exit
```

在上一个示例中，格式为：

```
[Date Time:Microsecs]    Thread-id    Classname    Trace-type    Methodname    Entry
or Exit
```

其中，Trace-type 为：

- > 表示入口
- < 表示出口
- d 表示调试信息

相关任务

[V 9.1.1 将 XMS 与 Microsoft .NET Core 结合使用](#)

[V 9.1.4 从 NuGet 存储库下载 IBM MQ classes for XMS .NET Standard](#)

使用应用程序配置文件配置 XMS .NET 跟踪

如果您在使用 IBM MQ classes for XMS .NET Framework，那么可以通过应用程序配置文件为 XMS .NET 应用程序配置跟踪。该文件的 trace 节包含用于定义要跟踪内容的参数、跟踪文件位置和最大允许大小、已使用的跟踪文件数以及跟踪文件格式。

关于此任务

[V 9.1.1 IBM MQ classes for XMS .NET Standard 不支持使用应用程序配置文件](#)。如果使用 IBM MQ classes for XMS .NET Standard，那么必须通过 XMS 环境变量来配置跟踪。要获取更多详细信息，请参阅第 375 页的『使用 XMS 环境变量配置 XMS .NET 跟踪』。

过程

- 要使用应用程序配置文件开启跟踪，请将该文件与应用程序的可执行文件放在同一目录中。
可以按组件和跟踪类型来启用跟踪。也可以对整个跟踪组开启跟踪。可以对层次结构中的组件单独或集体开启跟踪。可用的跟踪类型包括：
 - 调试跟踪
 - 异常跟踪
 - 警告、参考消息和错误消息
 - 方法入口和出口跟踪

以下示例显示了在应用程序配置文件的 Trace 节中定义的跟踪设置：

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
    <sectionGroup name="IBM.XMS">
      <section name="Trace"
        type="System.Configuration.SingleTagSectionHandler" />
    </sectionGroup>
  </configSections>

  <IBM.XMS>
    <Trace traceSpecification="*=all=enabled" traceFilePath=""
      traceFileSize="20000000" traceFileNumber="3"
      traceFormat="advanced" />
  </IBM.XMS>
</configuration>
```

```
</IBM.XMS>  
</configuration>
```

第 375 页的表 31 更详细地描述了参数设置。

| 参数 | 描述 |
|--|--|
| <code>traceSpecification=ComponentName=type=state</code> | <p><code>ComponentName</code> 是想要跟踪的类的名称。您可以在该名称中使用 * 通配符。例如, <code>*=all=enabled</code> 指定您要跟踪所有类, <code>IBM.XMS.impl.*=all=enabled</code> 指定您只需要 API 跟踪。</p> <p><code>type</code> 可以是以下任何跟踪类型:</p> <ul style="list-style-type: none">- all- debug- 事件- EntryExit <p><code>state</code> 可以启用或禁用。</p> <p>您可使用“:” (冒号) 定界符将多个跟踪元素串在一起。</p> |
| <code>traceFilePath="filename"</code> | <p>如果您未指定 <code>traceFilePath</code> 或者如果 <code>traceFilePath</code> 存在但包含空字符串, 那么会将跟踪文件放入当前目录中。要将跟踪文件存储在指定目录中, 请在 <code>traceFilePath</code> 中指定该目录名称, 例如:</p> <pre>traceFilePath="c:\somepath"</pre> |
| <code>traceFileSize="size"</code> | <p>允许的跟踪文件的最大大小。当文件达到此大小时, 会将其归档并重命名。缺省最大值为 20 MB, 指定为 <code>traceFileSize="20000000"</code>。</p> |
| <code>traceFileNumber="number"</code> | <p>要保留的跟踪文件的数量。缺省值为 4 (1 个活动文件和 3 个归档文件)。允许的最小数目是 2。</p> |
| <code>traceFormat="format"</code> | <p>缺省跟踪格式为 <code>basic</code>。在指定了 <code>traceFormat="basic"</code> 时、未指定 <code>traceFormat</code> 时或者 <code>traceFormat</code> 存在但包含空字符串时, 都将采用此格式生成跟踪文件。</p> <p>如果需要与跟踪分析器工具兼容的跟踪, 那么必须指定 <code>traceFormat="advanced"</code>。</p> |

应用程序配置文件中的跟踪设置是动态的, 每次保存或替换该文件时都会重新读入跟踪设置。如果在编辑该文件后在其中发现错误, 那么会将跟踪文件设置还原为其缺省值。

相关任务

使用 XMS 环境变量配置 XMS .NET 跟踪

您可以使用 XMS 环境变量 (例如, `XMS_TRACE_ON`) 来开启跟踪。

使用 XMS 环境变量配置 XMS .NET 跟踪

您可以使用 XMS 环境变量 (例如, `XMS_TRACE_ON`) 来开启跟踪。

关于此任务

如果使用 IBM MQ classes for XMS .NET Framework, 那么可以使用 XMS 环境变量 (可代替使用应用程序配置文件) 来开启跟踪。仅当应用程序配置文件中不包含跟踪规范时, 才可以使用这些环境变量。

V 9.1.1 如果使用 IBM MQ classes for XMS .NET Standard, 那么必须通过 XMS 环境变量来配置跟踪。IBM MQ classes for XMS .NET Standard 不支持使用应用程序配置文件。

过程

- 要为 XMS .NET 应用程序配置跟踪, 请在运行该应用程序之前设置以下环境变量:

| 环境变量 | 缺省 | 设置 | 含义 |
|-------------------------|--------|---|---|
| XMS_TRACE_ON | 不适用 | 不适用: 将忽略该变量的值 | 如果设置了 XMS_TRACE_ON, 那么缺省情况下将启用所有跟踪。 |
| XMS_TRACE_FILE_PATH | 当前工作目录 | /dirpath/ | 跟踪和 FFDC 记录将写入的目录路径。 XMS 将在当前工作目录中创建 FFDC 和跟踪文件, 除非您指定了备用位置。您可以通过将环境变量 XMS_TRACE_FILE_PATH 设置为希望 XMS 在其中创建 FFDC 和跟踪文件的目录的标准路径名。必须在启动要跟踪的应用程序之前设置该环境变量。您必须确保应用程序运行时所使用的用户标识有权对 XMS 将在其中创建 FFDC 和跟踪文件的目录执行写操作。 |
| XMS_TRACE_FORMAT | BASIC | BASIC 和 ADVANCED | 指定所需的跟踪格式, 可以是 BASIC 或 ADVANCED。缺省格式为 BASIC。ADVANCED 格式与跟踪分析器工具兼容。 |
| XMS_TRACE_SPECIFICATION | 不适用 | 请参阅第 374 页的『使用应用程序配置文件配置 XMS .NET 跟踪』(仅限 IBM MQ classes for XMS .NET Framework) | 覆盖跟踪规范, 该规范遵循第 374 页的『使用应用程序配置文件配置 XMS .NET 跟踪』中指定的格式。(仅限 IBM MQ classes for XMS .NET Framework) |

相关任务

使用应用程序配置文件配置 XMS .NET 跟踪

如果您在使用 IBM MQ classes for XMS .NET Framework, 那么可以通过应用程序配置文件为 XMS .NET 应用程序配置跟踪。该文件的 trace 节包含用于定义要跟踪内容的参数、跟踪文件位置和最大允许大小、已使用的跟踪文件数以及跟踪文件格式。

V 9.1.0.4 **V 9.1.4** 启用 LDAP 客户机库代码的动态跟踪

从 IBM MQ 9.1.0 Fix Pack 4 和 IBM MQ 9.1.4 开始, 可以开启和关闭 LDAP 客户机跟踪, 而无需停止或启动队列管理器。

关于此任务

在 IBM MQ 9.1.0 Fix Pack 4 和 IBM MQ 9.1.4 之前，必须停止或启动队列管理器，才能开启和关闭 LDAP 客户机跟踪。

从 IBM MQ 9.1.0 Fix Pack 4 和 IBM MQ 9.1.4 开始，可以使用 **strmqtrc** 命令来开启 LDAP 客户机跟踪，并使用 **endmqtrc** 命令来关闭 LDAP 客户机跟踪，而无需停止或启动队列管理器。要启用此行为，还必须将环境变量 **AMQ_LDAP_TRACE** 设置为非空值。

如果 **AMQ_LDAP_TRACE** 设置为非空值并且使用了 LDAP 功能，那么某些队列管理器进程会在 `/var/mqm/trace` 下创建一些长度为零的文件。如果使用 **strmqtrc** 命令开启该跟踪，那么系统会将一些跟踪信息写入到这些文件中。如果随后使用 **endmqtrc** 命令关闭该跟踪，那么会停止将跟踪信息写入到这些文件中，但是这些文件的句柄将保持打开状态，直至队列管理器终止。

UNIX 在 UNIX 平台上，如果只是使用 **rm** 命令取消这些文件的链接，那么无法完全释放文件系统空间。这是由句柄保持打开状态所导致。因此，当需要释放 `/var/mqm/trace` 中的磁盘空间时，应该执行队列管理器结束操作。

过程

- 将环境变量 **AMQ_LDAP_TRACE** 设置为非空值。
- 使用 **strmqtrc** 命令开启该跟踪：

```
strmqtrc -m QMNAME -t servicedata
```

- 使用 **endmqtrc** 命令关闭该跟踪。

故障后恢复

发生严重问题后，请完成一组过程进行恢复。

关于此任务

如果通过使用“故障诊断和支持”部分中描述的诊断技术无法解决底层问题，请使用此处描述的恢复方法。如果使用这些恢复技术仍无法解决您的问题，请联系 IBM 支持中心。

过程

请查看以下链接，以获取有关如何从不同类型的故障中恢复的指示信息：

- [第 378 页的『磁盘驱动器故障』](#)
- [第 379 页的『损坏的队列管理器对象』](#)
- [第 379 页的『损坏的单个对象』](#)
- [第 379 页的『自动介质恢复故障』](#)

z/OS

有关如何从 IBM MQ for z/OS 上不同类型的故障中恢复的指示信息，请查看以下链接：

- **z/OS**
[第 380 页的『共享队列问题』](#)
- **z/OS**
[第 381 页的『活动日志问题』](#)
- **z/OS**
[第 385 页的『归档日志问题』](#)
- **z/OS**
[第 388 页的『BSDS 问题』](#)

-  [第 394 页的『页集问题』](#)
-  [第 395 页的『耦合设施和 Db2 问题』](#)
-  [第 397 页的『长时间运行的工作单元的问题』](#)
-  [第 398 页的『IMS 相关问题』](#)
-  [第 400 页的『硬件问题』](#)

相关任务

[第 227 页的『联系 IBM 支持人员』](#)

如果您在 IBM MQ 方面遇到问题需要帮助，可以通过 IBM 支持站点联系 IBM 支持人员。您还可以预订有关 IBM MQ 修订、故障诊断和其他新闻的通知。


[第 5 页的『IBM MQ 故障诊断和支持』](#)

如果您的队列管理器网络或 IBM MQ 应用程序存在问题，那么可以使用本信息中描述的方法来帮助您诊断和解决问题。如果需要有关问题的帮助，可以通过 IBM 支持站点联系 IBM 支持人员。

[第 7 页的『在 UNIX, Linux, and Windows 上执行初步检查』](#)

开始在 UNIX, Linux, and Windows 上详细确定问题之前，请考虑是否存在引起问题的明显原因，或者是否存在可能得出有用结果的调查范围。这种诊断方法经常可以通过把重点放在一些简单的错误或缩小可能的范围来节省很多工作量。

[备份和复原 IBM MQ](#)

 [z/OS 上的备份和复原规划](#)

磁盘驱动器故障

您的包含了队列管理器数据和/或日志的磁盘驱动器可能会出现问题。这些问题可包括数据丢失或破坏。这三种情况的不同仅限于幸存的数据部分（如果有任何数据幸存的话）。

在所有情况下，请首先检查目录结构是否有任何损坏，并根据需要进行修复。如果您丢失的是队列管理器数据，那么队列管理器目录结构可能已受损。如果是这样，应在重新启动该队列管理器前手动重新创建该目录树。

如果队列管理器数据文件已发生了损坏，但队列管理器日志文件是完好的，那么队列管理器通常能够重新启动。如果队列管理器日志文件发生了任何损坏，那么队列管理器将可能不能重新启动。

检查结构损伤后，您可以执行许多操作，这取决于您所使用的日志类型。

- **如果目录结构受到严重损坏，或对日志有任何损坏**，应除去所有旧文件并回到 QMgrName 级别，包括配置文件、日志和队列管理器目录，复原最近的备份然后重新启动队列管理器。
- **对于具有介质恢复的线性日志记录**，确保该目录结构是完整的，并且重新启动该队列管理器。如果队列管理器重新启动，那么可使用 MQSC 命令（如 DISPLAY QUEUE）检查是否已损坏了任何其他对象。使用 rcrmqobj 命令恢复您找到的受损对象。例如：

```
rcrmqobj -m QMgrName -t all *
```

其中 QMgrName 是要恢复的队列管理器。-t all * 指示要恢复任何类型的所有受损对象。如果只有一个或两个对象报告为受损，那么您可以在此处按名称和类型指定那些对象。

- **对于具有介质恢复和未受损的日志的线性日志记录**，您也许能够恢复队列管理器数据的备份，保留现有日志文件和未更改的日志控制文件。启动队列管理器会应用根据日志所作的更改，把队列管理器置回故障发生时的状态。

此方法基于两个事实：

1. 您必须将检查点文件恢复为队列管理器数据部分。此文件包含一些信息，这些信息确定必须应用日志中的多少数据才能提供一致的队列管理器。
2. 备份时，您必须拥有启动队列管理器所必需的最旧的日志文件和此日志文件目录中所有的后续日志文件。

如果无法执行此操作，那么可复原队列管理器数据和日志的备份，它们是在同一时间备份的。这将使消息的完整性受损。

- **对于循环日志记录**，如果队列管理器日志文件已损坏，将从您拥有的最新备份复原队列管理器。恢复备份后，立即重新启动队列管理器并检查受损对象。但是，由于您没有介质恢复，因此必须寻求重新创建受损对象的其他方法。

如果队列管理器日志文件未损坏，队列管理器通常能够重新启动。在重新启动后，您必须找出所有受损的对象，然后删除并重新进行定义。

损坏的队列管理器对象

如果队列管理器在正常运行期间报告损坏的对象，那么应该如何处理。

根据您使用的日志记录类型，在这些情况下有两种恢复方法：

- **对于线性日志记录**，手动删除包含受损对象的文件并重新启动队列管理器。（您可以使用 `dspmqls` 命令确定受损对象的真实的文件系统名。）受损对象的介质恢复是自动的。
- **对于循环日志记录**，恢复队列管理器数据和日志的最新备份并重新启动队列管理器。

使用循环日志记录时，还另外提供了一个选项。对于损坏的队列或其他对象，删除该对象并重新定义该对象。如果是队列，此选项不允许恢复队列上的任何数据。

注：由于必须关闭队列管理器才能获取队列文件的干净备份，因此从备份复原可能会过时。

损坏的单个对象

如果单个对象在正常操作期间报告为已受损，那么对于线性日志记录，您可以从其介质映像中重新创建该对象。但是，对于循环日志记录，您不能重新创建单个对象。

自动介质恢复故障

如果带有线性日志的队列管理器启动所需的本地队列受损，且自动介质恢复发生故障，那么将恢复该队列管理器数据和日志的最新备份并重新启动该队列管理器。

z/OS 上的恢复过程示例

本主题作为各种恢复过程的参考。

本主题描述在各种错误情况之后恢复 IBM MQ 的过程。这些错误情况分为下列类别：

| 问题类别 | 问题 | 接着阅读何内容 |
|--------|---|------------------|
| 共享队列问题 | 专用和共享队列的冲突定义。 | 第 380 页的『共享队列问题』 |
| 活动日志问题 | <ul style="list-style-type: none"> • 双记录丢失。 • 活动日志停止。 • 活动日志数据集的其中一个副本损坏或者两个副本都损坏。 • 活动日志数据集发生写错误。 • 活动日志即将变满或已满。 • 活动日志数据集发生读错误。 | 第 381 页的『活动日志问题』 |

| 表 33: 恢复过程示例 (继续) | | |
|-------------------|--|---|
| 问题类别 | 问题 | 接着阅读何内容 |
| 归档日志问题 | <ul style="list-style-type: none"> • DASD 空间不足以完成卸载活动日志数据集。 • 卸载任务异常终止。 • 归档数据集分配问题。 1 • 在重新启动期间，归档数据集发生读 I/O 错误。 | 第 385 页的『 归档日志问题 』 |
| BSDS 问题 | <ul style="list-style-type: none"> • 打开 BSDS 时发生错误。 • 日志内容与 BSDS 信息不一致。 • BSDS 的两个副本都已损坏。 • 时间戳记不相同。 • 两个 BSDS 数据集不同步。 • BSDS 发生 I/O 错误。 | 第 388 页的『 BSDS 问题 』 |
| 页集问题 | <ul style="list-style-type: none"> • 页集已满。 • 页集发生 I/O 错误。 | 第 394 页的『 页集问题 』 |
| 耦合设施和 Db2 问题 | <ul style="list-style-type: none"> • 存储介质已满。 • Db2 系统发生故障。 • Db2 数据共享组发生故障。 • Db2 和耦合设施发生故障。 | 第 395 页的『 耦合设施和 Db2 问题 』 |
| 工作单元问题 | 遇到长时间运行的工作单元。 | 第 397 页的『 长时间运行的工作单元的问题 』 |
| IMS 问题 | <ul style="list-style-type: none"> • IMS 应用程序异常终止。 • IMS 适配器无法连接到 IBM MQ。 • IMS 无法运作。 | 第 398 页的『 IMS 相关问题 』 |
| 硬件问题 | 介质恢复过程 | 第 400 页的『 硬件问题 』 |

z/OS 共享队列问题

如果 IBM MQ 发现定义了基于页集的队列和使用相同名称的共享队列，那么将出现问题。

故障现象

IBM MQ 发出以下消息：

```
CSQI063E +CSQ1 QUEUE queue-name IS BOTH PRIVATE AND SHARED
```

在队列管理器重新启动期间，IBM MQ 发现基于页集的队列和使用相同名称的共享队列共存。

系统操作

在重新启动处理完成后，对该队列名发出的任何 MQOPEN 请求都将失败，从而表明发生共存问题。

系统程序员操作

无。

操作员操作

删除该队列的一个版本，以便允许处理该队列名。如果该队列中存在必须保留的消息，那么可以使用 MOVE QLOCAL 命令将其移至另一队列。

本主题用于解决与活动日志相关的不同问题。

本主题包含下列活动日志问题：

- [第 381 页的『双记录丢失』](#)
- [第 381 页的『活动日志停止』](#)
- [第 382 页的『活动日志数据集的其中一个副本损坏或者两个副本都损坏』](#)
- [第 382 页的『活动日志数据集发生写 I/O 错误』](#)
- [第 383 页的『读取活动日志时发生 I/O 错误』](#)
- [第 384 页的『活动日志即将变满』](#)
- [活动日志已满](#)

双记录丢失

故障现象

IBM MQ 发出以下消息：

```
CSQJ004I +CSQ1 ACTIVE LOG COPY n INACTIVE, LOG IN SINGLE MODE,  
ENDRBA=...
```

完成一个活动日志数据集后，IBM MQ 发现未卸载后续 (COPY n) 数据集或未将其标记为停止。

系统操作

IBM MQ 将继续处于单一方式直到完成卸载，然后将返回双重方式。

系统程序员操作

无。

操作员操作

检查卸载过程是否正在进行，并且未等待磁带安装。您可能需要运行输出日志映射实用程序以确定所有数据集的状态。另外，还可能需定义其他数据集。

活动日志停止

故障现象

IBM MQ 发出以下消息：

```
CSQJ030E +CSQ1 RBA RANGE start TO end NOT AVAILABLE IN ACTIVE  
LOG DATA SETS
```

系统操作

包含消息 CSQJ030E 中报告的 RBA 范围的活动日志数据集不可用于 IBM MQ。在 BSDS 中，这些日志的状态是 STOPPED。队列管理器将终止并进行转储。

系统程序员操作

在重新启动队列管理器之前，必须解决此问题。日志 RBA 范围必须可用于 IBM MQ 才能够恢复。在 BSDS 中被标记为 STOPPED 的活动日志将永远不会被复用或归档，这将导致日志中存在间隔。

查找指示了日志数据集停止原因的消息，并按照那些消息的指示信息执行操作。

修改 BSDS 活动日志库存以复位 STOPPED 状态。要完成此任务，请在队列管理器终止后完成以下过程：

1. 使用输出日志实用程序 (CSQJU004) 来获取 BSDS 日志库存的副本。这将显示日志数据集的状态。

2. 使用更改日志库存实用程序 (CSQJU003) 的 DELETE 功能来删除已被标记为 STOPPED 的活动日志数据集。
3. 使用 CSQJU003 的 NEWLOG 功能将活动日志添加回到 BSDS 库存中。在 NEWLOG 语句中, 必须指定每个活动日志数据集的起始 RBA 和结束 RBA。(可从步骤 1 中获取的输出日志实用程序报告中找到要使用的正确值。)
4. 重新运行 CSQJU004。现在, 先前被标记为 STOPPED 的活动日志数据集显示为 NEW 和 NOT REUSABLE。这些活动日志将在到期时进行归档。
5. 重新启动队列管理器。

注: 如果队列管理器正在以双重 BSDS 方式运行, 那么必须更新这两个 BSDS 库存。

活动日志数据集的其中一个副本损坏或者两个副本都损坏

故障现象

IBM MQ 发出以下消息:

```
CSQJ102E +CSQ1 LOG RBA CONTENT OF LOG DATA SET DSNAME=... ,
          STARTRBA=... , ENDRBA=... ,
          DOES NOT AGREE WITH BSDS INFORMATION
CSQJ232E +CSQ1 OUTPUT DATA SET CONTROL INITIALIZATION PROCESS FAILED
```

系统操作

队列管理器的启动处理终止。

系统程序员操作

如果数据集的一个副本损坏, 请执行下列步骤:

1. 将损坏的活动日志数据集重命名并定义替换数据集。
2. 将未损坏的数据集复制到替换数据集。
3. 使用更改日志库存实用程序来执行下列操作:
 - 从 BSDS 中移除关于损坏的数据集的信息。
 - 将关于替换数据集的信息添加到 BSDS。
4. 重新启动队列管理器。

如果活动日志数据集的两个副本都已损坏, 当前页集可用, 并且队列管理器已完全关闭, 请执行下列步骤:

1. 将损坏的活动日志数据集重命名并定义替换数据集。
2. 使用更改日志记录实用程序来执行下列操作:
 - 从 BSDS 中移除关于损坏的数据集的信息。
 - 将关于替换数据集的信息添加到 BSDS。
3. 将当前页集重命名并定义替换页集。
4. 使用 CSQUTIL (FORMAT 和 RESETPAGE) 来格式化替换页集并将重命名的页集复制到替换页集。RESETPAGE 功能还将复位替换页集中的日志信息。

如果未彻底关闭队列管理器, 那么必须从上一个已知一致点恢复您的系统, 或者执行冷启动 (如重新初始化队列管理器中所述)。

操作员操作

无。

活动日志数据集发生写 I/O 错误

故障现象

IBM MQ 发出以下消息：

```
CSQJ105E +CSQ1 csect-name LOG WRITE ERROR DSNAME=...,  
LOGRBA=..., ERROR STATUS=ccccffss
```

系统操作

IBM MQ 执行以下步骤：

1. 在 BSDS 中，将有错误的日志数据集标记为 TRUNCATED。
2. 转到下一个可用的数据集。
3. 如果使用双重活动日志记录功能，那么在同一个点截断另一副本。

稍后，照例会卸载已截断数据集中的数据。

在下一个循环中，将复用该数据集。

系统程序员操作

无。

操作员操作

如果此数据集的错误仍存在，那么将在下个卸载过程后关闭队列管理器。然后，使用访问方法服务 (AMS) 和更改日志库存实用程序来添加替换项。（有关指示信息，请参阅[更改 BSDS](#)。）

读取活动日志时发生 I/O 错误

故障现象

IBM MQ 发出以下消息：

```
CSQJ106E +CSQ1 LOG READ ERROR DSNAME=..., LOGRBA=...,  
ERROR STATUS=ccccffss
```

系统操作

这取决于错误的发生时间：

- 如果在卸载过程中发生错误，该过程会尝试从第二个副本中读取 RBA 范围。
 - 如果不存在第二个副本，那么活动日志数据集将停止。
 - 如果第二个副本同样存在错误，那么将只停止触发该卸载过程的原始数据集。然后，归档日志数据集将终止，从而在已归档的日志 RBA 范围中留下间隔。
 - 将发出以下消息：

```
CSQJ124E +CSQ1 OFFLOAD OF ACTIVE LOG SUSPENDED FROM  
RBA xxxxxx TO RBA xxxxxx DUE TO I/O ERROR
```

- 如果第二个副本令人满意，那么不停止第一个副本。
- 如果在恢复期间发生错误，那么 IBM MQ 将提供从其他副本或归档请求的特定日志 RBA 中的数据。如果此操作不成功，那么恢复将不成功，队列管理器将异常终止。
- 如果在重新启动期间发生错误且使用的是双记录，那么 IBM MQ 将使用备用日志数据集继续操作，否则队列管理器将异常结束。

系统程序员操作

查找系统消息（例如以 IEC 为前缀的消息），并尝试使用这些消息的建议操作来解决问题。

如果活动日志数据集已停止，那么它将不会用于日志记录操作。此数据集不会被取消分配；它仍可供读取。即使未停止此数据集，也应该替换发生持久错误的活动日志数据集。

操作员操作

无。

替换数据集

替换数据集的方式取决于您是正在使用单一还是双重活动日志记录功能。

如果使用的是双重活动日志记录：

1. 确保已保存数据。

数据将保存在另一活动日志中，并且，可以将其复制到替换活动日志。

2. 通过使用访问方法服务，停止队列管理器并删除发生错误的数据集。

3. 使用访问方法服务 DEFINE 来重新定义新的日志数据集，以便可以将内容写入该数据集。使用 DFDSS 或访问方法服务 REPRO 将良好的日志复制到重新定义的数据集，以再次拥有两个一致且正确的日志。

4. 如下所示，使用更改日志库存实用程序 CSQJU003 更新 BSDS 中有关已受损数据集的信息：

a. 使用 DELETE 功能移除有关已受损的数据集的信息。

b. 使用 NEWLOG 功能将新数据集指定为新的活动日志数据集，并为其提供成功复制的 RBA 范围。

可以在同一个作业步骤中运行 DELETE 和 NEWLOG 功能。请将 DELETE 语句置于 SYSIN 输入数据集的 NEWLOG 语句之前。

5. 重新启动队列管理器。

如果使用的是单一活动日志记录：

1. 确保已保存数据。

2. 停止队列管理器。

3. 确定是否卸载了具有错误的数据集：

a. 使用 CSQJU003 实用程序来列出 BSDS 中关于归档日志数据集的信息。

b. 在列表中搜索 RBA 范围包含受损数据集的 RBA 的数据集。

4. 如果已卸载受损的数据集，那么可将其在归档日志中的备份复制到新的数据集。之后，跳转至步骤 6。

5. 如果停止活动日志数据集，那么将不会卸载 RBA。请使用 DFDSS 或访问方法服务 REPRO 将数据从受损数据集复制到新的数据集中。

如果进一步的 I/O 错误导致无法复制整个数据集，那么日志将包含间隔。

注：如果在日志中检测到间隔，那么队列管理器将无法成功地重新启动。

6. 如下所示，使用更改日志库存实用程序 CSQJU003 更新 BSDS 中有关已受损数据集的信息：

a. 使用 DELETE 功能移除有关已受损的数据集的信息。

b. 使用 NEWLOG 功能将新数据集指定为新的活动日志数据集，并为其提供成功复制的 RBA 范围。

可以在同一个作业步骤中运行 DELETE 和 NEWLOG 功能。请将 DELETE 语句置于 SYSIN 输入数据集的 NEWLOG 语句之前。

7. 重新启动队列管理器。

活动日志即将变满

活动日志可能会因多种原因而填满，如卸载延迟和过度记录。如果活动日志耗尽空间，将产生严重后果。如果活动日志变满，那么队列管理器将暂停处理，直到卸载过程完成。如果卸载过程在活动日志已满时停止，那么队列管理器可能异常结束。您必须执行更正操作，然后才能重新启动队列管理器。

故障现象

由于活动日志变满将产生严重影响，因此队列管理器将在最后一个可用活动日志数据集的填满程度达到 5% 时发出以下警告消息：

```
CSQJ110E +CSQ1 LAST COPYn ACTIVE LOG DATA SET IS nnn PERCENT FULL
```

并且，将在数据集空间填满程度每提高 5% 后重新发出此消息。每次发出此消息，都会启动卸载过程。

系统操作

发出消息并启动卸载处理。如果活动日志变满，那么执行进一步的操作。请参阅第 385 页的『[活动日志已满](#)』。

系统程序员操作

使用 DEFINE LOG 命令来动态地添加更多的活动日志数据集。这允许 IBM MQ 在纠正导致卸载问题的错误的同时继续其正常操作。有关 DEFINE LOG 命令的更多信息，请参阅 [DEFINE LOG](#)。

活动日志已满

故障现象

如果活动日志变满，那么队列管理器将暂停处理，直到卸载过程完成。如果卸载过程在活动日志已满时停止，那么队列管理器可能异常结束。您必须执行更正操作，然后才能重新启动队列管理器。

IBM MQ 发出以下 [CSQJ111A](#) 消息：

```
CSQJ111A +CSQ1 OUT OF SPACE IN ACTIVE LOG DATA SETS
```

并且启动卸载过程。之后，队列管理器将暂停处理，直到卸载过程完成。

系统操作

IBM MQ 等待获取可用的活动日志数据集，获取后将恢复正常的 IBM MQ 处理。无法通过 QUIESCE 或 FORCE 正常关闭，因为关闭顺序要求日志空间记录与关闭相关的系统事件（例如，检查点记录）。如果卸载处理在活动日志变满时停止，那么队列管理器将停止并显示 X'6C6' 异常中止；在此情况下重新启动需要特别注意。有关更多详细信息，请参阅第 179 页的『[对 IBM MQ for z/OS 问题进行故障诊断](#)』。

系统程序员操作

在重新启动队列管理器之前，您可以提供其他活动日志数据集。这允许 IBM MQ 在纠正导致卸载处理问题的错误的同时继续其正常操作。要添加新的活动日志数据集，请在队列管理器处于不活动状态时使用更改日志库存实用程序 (CSQJU003)。有关添加新活动日志数据集的更多详细信息，请参阅 [更改 BSDS](#)。

考虑通过下列操作增加日志数：

1. 确保已停止队列管理器，然后使用访问方法服务 DEFINE 命令定义新的活动日志数据集。
2. 使用更改日志库存实用程序 (CSQJU003) 在 BSDS 中定义新的活动日志数据集。
3. 使用 [DEFINE LOG](#) 命令动态添加额外的日志数据集。

重新启动队列管理器时，卸载会在启动期间自动启动，并且在强制停止 IBM MQ 时正在运行的所有工作都将恢复。

操作员操作

检查卸载过程是否在等待磁带机。如果是，请安装磁带。如果无法安装磁带，请使用 z/OS CANCEL 命令强制停止 IBM MQ。

归档日志问题

本主题用于调查和解决与归档日志相关的问题。

本主题包含下列归档日志问题：

- [第 386 页的『分配问题』](#)
- [第 386 页的『卸载任务异常终止』](#)
- [第 387 页的『DASD 空间不足以完成卸载处理』](#)
- [第 388 页的『在 IBM MQ 重新启动期间，归档数据集发生读 I/O 错误。』](#)

分配问题

故障现象

IBM MQ 发出消息：[CSQJ103E](#)

```
CSQJ103E +CSQ1 LOG ALLOCATION ERROR DSNAME=dsname,  
          ERROR STATUS=eeeeiii, SMS REASON CODE=sss
```

z/OS 动态分配提供 ERROR STATUS。如果分配用于卸载处理，那么还将显示以下消息：[CSQJ115E](#)：

```
CSQJ115E +CSQ1 OFFLOAD FAILED, COULD NOT ALLOCATE AN ARCHIVE  
          DATA SET
```

系统操作

发生下列操作：

- 如果恢复需要输入，并且恢复不成功，队列管理器将异常结束。
- 如果活动日志变满，并且卸载任务已安排但未完成，那么会在下次触发时重试该卸载任务。活动日志不复用尚未归档的数据集。

系统程序员操作

无。

操作员操作

检查分配错误码以了解问题原因并更正问题。确保驱动器可用，然后重新启动或等待以重试卸载任务。如果已经为归档日志数据集编写 DFP/DFSMS ACS 用户出口过滤器，请务必小心，其原因在于，这可能导致队列管理器在尝试读取归档日志数据集时发生设备分配错误。

卸载任务异常终止

故障现象

未针对写 I/O 错误发出任何特定 IBM MQ 消息。

仅显示 z/OS 错误恢复程序消息。如果收到 IBM MQ 消息 [CSQJ128E](#)，说明卸载任务已异常结束。

系统操作

发生下列操作：

- 卸载任务废弃了输出数据集；BSDS 中未记录任何条目。
- 卸载任务动态分配了新的归档，并从上次的触发点重新启动了卸载。
- 如果新数据集发生错误：
 - 在双重归档方式下，将生成消息 [CSQJ114I](#)，卸载处理将更改为单一方式：

```
CSQJ114I +CSQ1 ERROR ON ARCHIVE DATA SET, OFFLOAD
CONTINUING WITH ONLY ONE ARCHIVE DATA SET BEING
GENERATED
```

- 在单一归档方式下，将废弃输出数据集。在下次触发卸载处理时，将再次尝试处理此 RBA 范围。
- 活动日志不回绕；即使没有更多的活动日志，也不会丢失数据。

系统程序员操作

无。

操作员操作

确保在可靠的驱动器和控制单元上分配卸载任务。

DASD 空间不足以完成卸载处理

故障现象

将活动日志数据集卸载到 DASD 期间，进程意外终止。IBM MQ 发出消息 [CSQJ128E](#)：

```
CSQJ128E +CSQ1 LOG OFF-LOAD TASK FAILED FOR ACTIVE LOG nnnnn
```

错误前是 z/OS 消息 IEC030I、IEC031I 或 IEC032I。

系统操作

IBM MQ 取消分配出现错误的数据集。如果 IBM MQ 正在以双重归档方式运行，那么 IBM MQ 将更改为单一归档方式并继续执行卸载任务。如果在单一归档方式下无法完成卸载任务，那么将无法卸载活动日志数据集，该活动日志数据集的状态将仍保持为 NOT REUSABLE。下次触发卸载任务时，将再次尝试处理已废弃活动日志数据集的 RBA 范围。

系统程序员操作

这些症状的最有可能的原因是：

- 归档日志数据集的大小太小，在卸载处理期间无法包含活动日志数据集的数据。分配的所有辅助空间已用尽。这种情况通常会伴有 z/OS 消息 IEC030I。此消息中的返回码可能提供对这些症状的原因的进一步解释。

解决该问题

1. 发出命令 `CANCEL queue_manager name` 以取消队列管理器作业
2. 增加归档日志数据集 (在 CSQ6ARVP 系统参数中) 的主分配和/或辅助分配。

如果要卸载的数据非常庞大，那么可以安装其他联机储存卷或向 IBM MQ 提供可用联机储存卷。

3. 重新启动队列管理器。

- 归档数据集写至的 DASD 卷上的所有可用空间均已耗尽。这种情况通常会伴有 z/OS 消息 IEC032I。

为解决此问题，请在 DASD 卷上提供更多可用空间，或者向 IBM MQ 提供其他联机存储卷。

- 归档日志数据集的主空间分配量（在 CSQ6ARVP 系统参数中指定）太大，无法分配给任何可用的联机 DASD 设备。这种情况通常会伴有 z/OS 消息 IEC032I。

为解决此问题，请在 DASD 卷上提供更多可用空间，或者向 IBM MQ 提供其他联机存储卷。如果不可能完成此任务，那么必须在 CSQ6ARVP 系统参数中调整 `PRIQTY` 的值以减少主空间分配量。（有关详细信息，请参阅使用 [CSQ6ARVP](#)。）

注：如果减少主空间分配量，那么可能必须增大辅助空间分配大小以避免将来发生异常结束。

操作员操作

无。

在 IBM MQ 重新启动期间，归档数据集发生读 I/O 错误。

故障现象

未发出任何特定 IBM MQ 消息；仅显示 z/OS 错误恢复程序消息。

系统操作

这取决于是否存在另一个副本：

- 如果存在另一个副本，那么将分配并使用该副本。
- 如果不存在另一个副本，那么重新启动将失败。

系统程序员操作

无。

操作员操作

尝试使用另一个驱动器来重新启动。

z/OS BSDS 问题

本主题用于调查和解决与 BSDS 相关的问题。

有关引导数据集 (BSDS) 的背景信息，请参阅 [在 z/OS 上规划 IBM MQ 环境](#)。

本主题描述以下 BSDS 问题：

- [第 388 页的『打开 BSDS 时发生错误』](#)
- [第 389 页的『日志内容与 BSDS 信息不一致』](#)
- [第 389 页的『BSDS 的两个副本都已损坏』](#)
- [第 390 页的『时间戳记不相同』](#)
- [第 390 页的『不同步』](#)
- [第 391 页的『I/O 错误』](#)
- [第 392 页的『日志范围问题』](#)

通常，BSDS 具有两个副本，但如果其中的一个受损，IBM MQ 会立即更改为单一 BSDS 方式。但是，在重新启动之前，必须恢复已损坏的 BSDS 副本。如果处于单一方式并且损坏了唯一的 BSDS 副本，或者处于双重方式但同时损坏了两个副本，请使用[恢复 BSDS](#)中描述的过程。

本节阐述启动时可能发生的一些 BSDS 问题。此处未阐述的问题包括：

- RECOVER BSDS 命令错误（消息 CSQJ301E - CSQJ307I）
- 更改日志库存实用程序错误（消息 CSQJ123E）
- 通过卸载处理转储的 BSDS 备份中的错误（消息 CSQJ125E）

打开 BSDS 时发生错误

故障现象

IBM MQ 发出以下消息：

```
CSQJ100E +CSQ1 ERROR OPENING BSDSn DSNAME=..., ERROR STATUS=eii
```

其中，*eii* 是 VSAM 返回码。有关 VSAM 代码的信息，请参阅 *DFSMS/MVS* 数据集的宏指示信息文档。

系统操作

在系统初始化期间，启动终止。

在 RECOVER BSDS 命令期间，系统以单一 BSDS 方式继续工作。

系统程序员操作

无。

操作员操作

执行下列步骤：

1. 对 BSDS 的两个副本都运行输出日志映射实用程序，并对列表进行比较以确定哪个副本更准确或更新。
2. 将有问题的数据集重命名，并为其定义替换数据集。
3. 使用访问方法服务将准确的数据集复制到替换数据集。
4. 重新启动队列管理器。

日志内容与 BSDS 信息不一致

故障现象

IBM MQ 发出以下消息：

```
CSQJ102E +CSQ1 LOG RBA CONTENT OF LOG DATA SET DSNAME=...,  
          STARTRBA=..., ENDRBA=...,  
          DOES NOT AGREE WITH BSDS INFORMATION
```

此消息表示您未正确地使用更改日志库存实用程序，或者正在使用低级别的数据集。

系统操作

队列管理器的启动处理终止。

系统程序员操作

无。

操作员操作

运行输出日志映射实用程序和更改日志库存实用程序，以便输出并更正 BSDS 的内容。

BSDS 的两个副本都已损坏

故障现象

IBM MQ 发出以下消息：

```
CSQJ107E +CSQ1 READ ERROR ON BSDS  
          DSNAME=... ERROR STATUS=0874  
CSQJ117E +CSQ1 REG8 INITIALIZATION ERROR READING BSDS  
          DSNAME=... ERROR STATUS=0874  
CSQJ119E +CSQ1 BOOTSTRAP ACCESS INITIALIZATION PROCESSING FAILED
```

系统操作

队列管理器的启动处理终止。

系统程序员操作

执行下列步骤：

1. 将数据集重命名，并为其定义替换数据集。
2. 找到与最新的归档日志数据集相关联的 BSDS，并将其复制到替换数据集。
3. 使用输出日志映射实用程序来输出替换 BSDS 的内容。
4. 使用输出日志记录实用程序来输出替换 BSDS 中缺少的活动日志数据集的摘要报告以及确定 RBA 范围。

5. 使用更改日志库存实用程序来更新替换 BSDS 中缺少的活动日志数据集库存。
6. 如果双 BSDS 数据集已在使用中，请将更新后的 BSDS 复制到 BSDS 的第二个副本。
7. 重新启动队列管理器。

操作员操作

无。

时间戳记不相同

故障现象

IBM MQ 发出以下消息：

```
CSQJ120E +CSQ1 DUAL BSDS DATA SETS HAVE UNEQUAL TIME STAMPS,  
SYSTEM BSDS1=...,BSDS2=...,  
UTILITY BSDS1=...,BSDS2=...
```

可能的原因是：

- 已复原 BSDS 的一个副本。有关已恢复 BSDS 的所有信息都是低级别。低级别 BSDS 具有较早的时间戳记。
- 已复原其中一个包含 BSDS 的卷。有关已恢复卷的所有信息都是低级别。如果该卷包含任何活动日志数据集或 IBM MQ 数据，这些数据/数据也同样是低级别。低级别卷具有较早的时间戳记。
- 双记录已降级为单一记录，且您正在尝试在不恢复已损坏的日志的情况下启动。
- 在更新 BSDS 的一个副本之后但在更新第二个副本之前，队列管理器异常终止。

系统操作

IBM MQ 尝试使用最近的副本再同步 BSDS 数据集。如果此操作失败，那么队列管理器的启动将终止。

系统程序员操作

无。

操作员操作

如果自动再同步失败，请执行下列步骤：

1. 对 BSDS 的两个副本都运行输出日志映射实用程序，并对列表进行比较以确定哪个副本更准确或更新。
2. 将低级别数据集重命名，并为其定义替换数据集。
3. 使用访问方法服务将良好的数据集复制到替换数据集。
4. 适当时，确定是否已复原包含低级别 BSDS 的卷。如果已复原该卷，那么该卷上的所有数据（例如活动日志数据）也是低级别数据。

如果复原的卷包含活动日志数据，并且正在不同的卷上使用双重活动日志，那么需要将当前版本的活动日志复制到低级别日志数据集。有关如何操作的详细信息，请参阅[恢复日志](#)。

不同步

故障现象

IBM MQ 在队列管理器初始化期间发出以下消息：

```
CSQJ122E +CSQ1 DUAL BSDS DATA SETS ARE OUT OF SYNCHRONIZATION
```

V9.1.0 BSDS 的两个输入副本具有不同的时间戳记，或者包含不一致的记录。如果使用更改日志库存实用程序时发生操作员错误，那么可能会存在差别。（例如，只对一个副本运行了更改日志库存实

用程序。) 更改日志库存实用程序启动时, 将在 BSDS 控制记录中设置专用时间戳记, 并在结束时设置关闭标志。IBM MQ 将检查更改日志库存实用程序时间戳记, 如果不相同, 或者相同但一个关闭标志未设置, 那么 IBM MQ 会比较 BSDS 的副本。如果副本不相同, 那么将发出消息 CSQJ122E。

如果指定了两个输入 BSDS, 但发现在两个 BSDS 副本之间不相同的记录, 那么 BSDS 转换实用程序也将发出此消息。如果队列管理器在 BSDS 转换实用程序运行之前异常终止, 那么会出现此情况。

系统操作

队列管理器启动或实用程序终止。

系统程序员操作

无。

操作员操作

如果在队列管理器初始化期间发生错误, 那么可执行以下步骤:

1. 对 BSDS 的两个副本都运行输出日志映射实用程序, 并对列表进行比较以确定哪个副本更准确或更新。
2. 将有问题的数据集重命名, 并为其定义替换数据集。
3. 使用访问方法服务将准确的数据集复制到替换数据集。
4. 重新启动队列管理器。

如果在运行 BSDS 转换实用程序期间发生错误, 那么可执行以下步骤:

1. 尝试重新启动队列管理器, 然后将其彻底关闭, 之后再次尝试运行 BSDS 转换实用程序。
2. 如果此过程无法解决问题, 请针对 BSDS 的两个副本都运行输出日志映射实用程序, 并对列表进行比较以确定哪个副本更准确或更新。
3. 更改用于调用 BSDS 转换实用程序的 JCL, 以在 SYSUT1 DD 语句中指定当前的 BSDS, 移除 SYSUT2 DD 语句, 然后重新提交该作业。

I/O 错误

故障现象

IBM MQ 更改为单一 BSDS 方式并发出用户消息:

```
CSQJ126E +CSQ1 BSDS ERROR FORCED SINGLE BSDS MODE
```

在此消息后面, 跟着下列其中一条消息:

```
CSQJ107E +CSQ1 READ ERROR ON BSDS  
          DSNNAME=... ERROR STATUS=...  
  
CSQJ108E +CSQ1 WRITE ERROR ON BSDS  
          DSNNAME=... ERROR STATUS=...
```

系统操作

BSDS 方式由双重方式切换到单一方式。

系统程序员操作

无。

操作员操作

执行下列步骤:

1. 使用访问方法服务来重命名或删除损坏的 BSDS, 并定义与发生错误的 BSDS 同名的新 BSDS。您可以在 thlqual.SCSQPROC 中的作业 CSQ4BREC 中找到控制语句示例。

2. 发出 IBM MQ 命令 RECOVER BSDS，以在最新分配的数据集中制作完好 BSDS 的副本，然后恢复双重 BSDS 方式。另请参阅[恢复 BSDS](#)。

日志范围问题

症状

IBM MQ 在读取自己的日志时发出消息 [CSQJ113E](#)，或者在读取队列共享组中队列管理器的日志时发出消息 [CSQJ133E](#) 或 [CSQJ134E](#)。如果没有重新启动队列管理器或恢复 CF 结构所需的归档日志，那么将发生此情况。

系统操作

根据读取的日志记录以及读取的原因，请求程序可能异常结束，原因码为 X'00D1032A'。

系统程序员操作

运行输出日志映射实用程序 (CSQJU004) 以确定导致错误的原因。发出消息 [CSQJ133E](#) 或 [CSQJ134E](#) 时，根据消息中指示的队列管理器的 BSDS 运行实用程序。

如果您：

- 从 BSDS 中删除具有日志范围的条目（包含消息中指示的日志 RBA 或 LRSN），并且
- 未删除或复用数据集

那么可以使用以下过程将该条目重新添加回 BSDS：

1. 通过查看 BSDS 内容的原有副本或根据 BSDS 的备份运行 CSQJU004，确定包含所需 RBA 或 LRSN 的数据集。
2. 使用更改日志库存实用程序 ([CSQJU003](#)) 将数据集添加回 BSDS。
3. 重新启动队列管理器。

如果已删除了归档日志数据集，那么将不可恢复需要归档日志的页集或 CF 结构。确定队列管理器需要读取日志记录的原因，然后根据受影响的页集或 CF 结构执行以下操作之一。

页面集

在队列管理器重新启动的恢复阶段出现的消息 [CSQJ113E](#) 指示执行介质恢复以使页集保持最新时需要日志。

在队列管理器重新启动期间，通过查看针对每个页集发出的 [CSQI1049I](#) 消息中的介质恢复 RBA，确定需要已删除的日志数据集才能执行介质恢复的页集，然后执行以下操作。

• 页集零

通过使用以下过程，可以恢复页集零上的对象。



注意：执行该过程时，所有其他页集上的所有数据都将丢失。

1. 使用 [CSQUTIL](#) 实用程序的 SDEFS 功能生成 IBM MQ DEFINE 命令的文件。
2. 使用 [CSQUTIL](#) 格式化页集零，然后按照下一节中的描述重新定义其他页集。
3. 重新启动队列管理器。
4. 使用 [CSQUTIL](#) 利用步骤 1 中的实用程序生成的 DEFINE 命令重新定义对象。

• 页集 1-99

使用以下过程重新定义页集。



注意：执行该操作时，页集上的所有数据都将丢失。

1. 如果可以访问页集且不出现任何 I/O 错误，那么可通过 [CSQUTIL](#) 实用程序利用 `FORMAT TYPE(NEW)` 命令重新格式化该页集。
2. 如果访问页集时发生 I/O 错误，请删除并重新创建该页集。

如果您希望将页面设置为与之前相同的大小，请使用 `LISTCAT ENT(dsname) ALLOC` 命令获取现有空间分配，并在 z/OS [DEFINE CLUSTER](#) 命令中使用这些分配。

通过 CSQUTIL 实用程序利用 FORMAT TYPE(NEW) 命令格式化新的页集。

3. 重新启动队列管理器。您可能必须执行某些操作，如重置通道或解析不确定通道。

CF 结构

恢复 CF 结构期间的消息 [CSQJ113E](#)、[CSQJ133E](#) 或 [CSQJ134E](#)，指示至少有一个队列共享组成员上的恢复结构所需的日志不可用。

根据受影响的结构，执行以下操作之一：

应用程序 CF 结构

发出命令 RECOVER CFSTRUCT(*structure-name*) TYPE(PURGE)。

此过程将清空结构，因此该结构中的所有消息都将丢失。

CSQSYSAPPL 结构

请联系 IBM 支持中心。

管理结构

此结构是使用每个队列管理器最后一个检查点之后的日志数据重新构建的，应当处于活动日志中。

如果在管理结构恢复期间遇到此错误，请联系您的 IBM 支持中心，因为此错误表明活动日志不可用。

在恢复页集或 CF 结构后，立即执行日志、BSDS、页集和 CF 结构的备份。

为防止此问题再次发生，请：

- 将归档日志保留时间 (ARCRETN) 值增大，并且
- 提高备份 CF 结构的频率。

恢复 CF 结构

就概念而言，从 IBM MQ 日志读取先前备份的 CF 结构中的数据；日志从备份向前读取，并且任何更改都重新应用于已复原的结构。

关于此任务

要使用的日志范围介于要恢复的每个结构的最新备份到当前时间之间。日志范围通过日志范围序列号 (LRSN) 值进行识别。

LRSN 使用“存储时钟值”的 6 个最高有效数字。

请注意，如果尚未对结构进行备份，那么将读取整个日志（回到创建结构的时间）。

过程

1. 请检查是否读取了队列共享组 (QSG) 中每个队列管理器的日志来获取此 LRSN 范围内的记录。

请注意，日志向后进行读取。

2. 检查是否已构建要恢复的每个结构的更改列表。
3. 系统将读取耦合设施 (CF) 结构中的数据并复原数据。

例如，如果已在队列管理器 A 上进行备份，并且恢复是在队列管理器 B 上运行，那么队列管理器 B 从队列管理器 A 读取日志以复原结构。

读取 CF 结构备份的开头后，将启动内部任务来采用结构的已复原数据并将其与从日志读取的更改进行合并。

4. 检查是否对正在进行复原的每个结构继续处理过程。

示例

在以下示例中，已发出命令 RECOVER CFSTRUCT(APP3)，并且产生了以下消息：

```
04:00:00 CSQE132I CDL2 CSQERRPB Structure recovery started, using log range from
LRSN=CC56D01026CC
to LRSN=CC56DC368924
```

This is the start of reading the logs backwards from each qmgr in the queue sharing group from the time of failure to the to the structure backup. The LRSN values give the ranges being used. Log records for all structures (just one structure in this example) being recovered are processed at the same time.

```
04:02:00 CSQE133I CDL2 CSQERPLS Structure recovery reading log backwards, LRSN=CC56D0414372
This message is produced periodically to show the process
```

```
04:02:22 CSQE134I CDL2 CSQERRPB Structure recovery reading log completed
The above process of replaying the logs backwards has finished,
```

```
04:02:22 CSQE130I CDL2 CSQERCF2 Recovery of structure APP3 started, using CDL1 log range
from RBA=000EE86D902E to RBA=000EF5E8E4DC
The task to process the data for APP3 has been started. The last backup of CF structure
APP3 was done on CDL1 within the given RBA range, so this log range has to be read.
```

```
04:02:29 CSQE131I CDL2 CSQERCF2 Recovery of structure APP3 completed
The data merge has completed. The structure is recovered.
```

页集问题

本主题用于调查和解决与页集相关的问题。

本主体涵盖您可能遇到的有关页集的问题：

- [第 394 页的『页集 I/O 错误』](#) 描述页集损坏时发生的情况。
- [第 395 页的『页集已满』](#) 描述页集没有足够的空间可用于进一步执行任何 MQI 操作时发生的情况。

页集 I/O 错误

问题

页集发生 I/O 错误。

故障现象

将发出以下消息：

```
CSQP004E +CSQ1 csect-name I/O ERROR STATUS ret-code
PSID psid RBA rba
```

系统操作

队列管理器异常终止。

系统程序员操作

无。

操作员操作

修复 I/O 错误原因。

如果没有页集受损，请重新启动队列管理器。IBM MQ 会通过日志自动将页集恢复为一致状态。

如果一个或多个页集损坏，请执行下列操作：

1. 将损坏的页集重命名并定义替换页集。
2. 将最新的备份页集复制到替换页集。
3. 重新启动队列管理器。IBM MQ 会自动应用日志记录的任何必要的更新。

如果页集零不可用，那么无法重新启动队列管理器。如果另外某个页集不可用，那么您可以将队列管理器启动 JCL 过程中的页集 DD 语句注释掉。这使您能够推迟恢复有缺陷的页集，以使其他用户能够继续访问 IBM MQ。

将该页集添加回到 JCL 过程时，系统将从先前从 JCL 中除去该页集的位置重新开始读取日志，并一直读取到日志末尾。如果已记录了大量数据，那么此过程可能需要很长时间。

对于任何尝试访问不可用页集中定义的队列的应用程序，将返回原因码 MQRC_PAGESET_ERROR。

复原有缺陷的页集之后，请复原与其相关联的 DD 语句并重新启动队列管理器。

仅当所有日志数据集都可用时，才有可能执行此处描述的操作员操作。如果日志数据集丢失或受损，请参阅[如果日志数据集已丢失，请重新启动](#)。

页集已满

问题

在页集中，没有足够的空间来执行下列其中一项操作：

- 要完成的 MQPUT 或 MQPUT1 调用
- 要完成的对象处理命令（例如 DEFINE QLOCAL）
- 要完成的动态队列 MQOPEN 调用

故障现象

请求失败，原因码为 MQRC_STORAGE_MEDIUM_FULL。由于页集中没有足够的剩余空间，因此队列管理器无法完成该请求。

即使页集扩展属性设置为 EXPAND(USER)，也有可能会出现原因码 MQRC_STORAGE_MEDIUM_FULL。在将原因码 MQRC_STORAGE_MEDIUM_FULL 返回给应用程序代码之前，队列管理器将尝试扩展该页集并重试 API 请求。在负载较重的系统上，在重试该 API 之前，扩展的存储器可由其他 IO 操作使用。请参阅[管理页集](#)。

此问题的原因可能是，由于无法将消息发送到另一个系统而导致它们积累在传输队列中。

系统操作

在除去足够的消息或删除对象以便为新的入局请求腾出空间之前，使用此页集的进一步请求将被阻塞。

操作员操作

使用 IBM MQ 命令 DISPLAY USAGE PSID(*) 标识已满的页集。

系统程序员操作

您可以增大所涉及的页集，也可以通过将队列移至另一页集来降低该页集的负载。有关这些任务的更多信息，请参阅[管理页集](#)。如果问题原因是消息积累在传输队列中，请考虑启动分布式排队功能以传输消息。

耦合设施和 Db2 问题

本主题用于调查和解决与耦合设施和 Db2 相关的问题。

本节涵盖您可能遇到的有关耦合设施和 Db2 的问题：

- [第 395 页的『存储介质已满』](#)
- [第 396 页的『Db2 系统失败』](#)
- [第 396 页的『Db2 数据共享组故障』](#)
- [第 397 页的『Db2 和耦合设施故障』](#)

存储介质已满

问题

耦合设施结构已满。

故障现象

如果队列结构变满，那么会将返回码 MQRC_STORAGE_MEDIUM_FULL 返回给应用程序。

管理结构变满时，确切的症状取决于遇到错误的进程，这些症状可能是没有 CMDSCOPE(GROUP) 命令的响应，也可能是队列管理器由于落实处理期间发生问题而导致故障等各种症状。

系统程序员操作

您可以使用 IBM MQ 来禁止对结构中的某些队列执行 MQPUT 操作，以阻止应用程序写入更多消息、启动更多从队列中获取消息的应用程序，或者停顿正在向队列放入消息的部分应用程序。

另外，也可以使用 XES 设施在原位置更改结构大小。以下 z/OS 命令将更改结构的大小：

```
SETXCF START,ALTER,STRNAME= structure-name,SIZE= newsize
```

其中，*newsize* 表示一个值，该值小于结构的 CFRM 策略上指定的 MAXSIZE 值，但大于当前耦合设施的大小值。

您可以通过 DISPLAY CFSTATUS 命令监控耦合设施结构的利用率。

Db2 系统失败

如果 IBM MQ 连接到的 Db2 子系统发生故障，那么 IBM MQ 会尝试重新连接到该子系统，并继续工作。如果在 CSQ6SYSP 系统参数模块的 QSGDATA 参数中指定了 Db2 组连接名称，那么 IBM MQ 请重新连接到与发生故障的 Db2 属于同一数据共享组的另一个活动 Db2 (如果在同一 z/OS 映像上提供了一个)。

在 IBM MQ 未连接到 Db2 时，某些队列管理器操作无法进行。这些字段为：

- 删除共享队列或组对象定义。
- 更改共享队列或组对象定义或者对其发出 MQSET。共享队列的 MQSET 限制意味着，触发或生成性能事件之类的操作无法正确进行。
- 定义新的共享队列或组对象。
- 显示共享队列或组对象。
- 启动或停止共享通道或者对其执行其他操作。
- 通过发出 MQOPEN 第一次打开共享队列后从 Db2 中读取共享队列定义。

其他 IBM MQ API 操作对于共享队列继续正常工作，并且可以针对从 GROUP 对象构建的队列管理器专用版本 (COPY 对象) 执行所有 IBM MQ 操作。同样，任何正在运行的共享通道都将正常地继续运行到结束或发生错误为止，如果发生错误，它们将进入重试状态。

IBM MQ 重新连接到 Db2 后，将在队列管理器和 Db2 之间执行再同步。这涉及向队列管理器通知已在 Db2 中定义的新对象 (其他队列管理器可能已能够通过其他 Db2 子系统在其他 z/OS 映像上继续正常工作)，以及更新已在 Db2 中更改的共享队列的对象属性。将恢复任何处于重试状态的共享通道。

如果 Db2 发生故障，它可能会在故障时锁定 Db2 资源。在某些情况下，这可能会使未受影响的其他队列管理器无法使用某些 IBM MQ 对象。要解决此问题，可重新启动发生故障的 Db2，以便其可执行恢复处理并释放锁定。

Db2 数据共享组故障

如果整个 Db2 数据共享组发生故障，那么恢复可能会恢复至故障时间，或者恢复至上一个时间点。

对于恢复到故障点的情况，IBM MQ 会在 Db2 恢复后重新连接并执行再同步过程，之后将恢复正常的队列管理器功能。

然而，如果将 Db2 恢复到上一个时间点，那么耦合设施结构中的实际队列和这些队列的 Db2 视图之间可能会存在不一致情况。例如，在 Db2 恢复到的时间点，存在已删除的队列，并且其在耦合设施结构中的位置已由现在包含消息的新队列定义复用。

如果发现您处于这种情况，那么必须停止队列共享组中的所有队列管理器，清除耦合设施结构，然后重新启动队列管理器。之后，您必须使用 IBM MQ 命令定义任何缺失的对象。为此，请完成以下过程：

1. 通过以实用程序方式启动 Db2 或更改安全概要文件，阻止 IBM MQ 重新连接到 Db2。

2. 如果您在共享队列上有任何重要的消息，那么可以使用 CSQUTIL 实用程序的 COPY 功能将其卸载，但该操作可能无效。
3. 终止所有队列管理器。
4. 使用以下 z/OS 命令清除所有结构：

```
SETXCF FORCE,STRUCTURE,STRNAME=
```

5. 将 Db2 复原到历史时间点。
6. 重新建立 Db2 的队列管理器访问。
7. 重新启动队列管理器。
8. 通过备份副本恢复 IBM MQ 定义。
9. 将已卸载的所有消息重新装入到共享队列。

在队列管理器重新启动时，它们会尝试与 Db2 GROUP 对象再同步本地 COPY 对象。这可能会导致 IBM MQ 尝试执行以下操作：

- 为在 Db2 恢复到的时间点存在的原有 GROUP 对象创建 COPY 对象。
- 删除在 Db2 恢复到的时间点之后创建的 GROUP 对象的 COPY 对象，以使其不存在于数据库中。

尝试对 COPY 对象执行的 DELETE 操作带有 NOPURGE 选项，因此，对于在这些 COPY 队列中仍有消息的队列管理器，此操作将失败。

Db2 和耦合设施故障

如果耦合设施发生故障，那么队列管理器也可能发生故障，并且如果 Db2 正在使用此耦合设施，那么也将发生故障。

使用 Db2 恢复过程恢复 Db2。重新启动 Db2 后，可以重新启动队列管理器。CF 管理结构也将发生故障，但重新启动队列共享组中的所有队列管理器将重建此结构。

如果耦合设施中的单个应用程序结构发生故障，那么对队列管理器所造成的影响将取决于队列管理器的级别和发生故障的 CF 结构的 CFLEVEL：

- 如果 CF 应用程序结构为 CFLEVEL 为 3 或更高级别，并且 RECOVER 设置为 YES，那么只有在向执行恢复的队列管理器发出 MQSC RECOVER CFSTRUCT 命令以恢复该 CF 结构后，才可以使用该结构。您可以指定只恢复单一 CF 结构，也可以同时恢复多个 CF 结构。执行恢复的队列管理器会使用 Db2 中的数据和引导数据集在所有其他队列管理器的日志上查找相关的备份。此队列管理器以正确的时间顺序跨队列共享组回放这些备份，并正好从上次备份之前一直回放到故障点。如果可恢复的应用程序结构发生故障，那么直到恢复该结构之后，才能执行进一步的应用程序活动。如果管理结构也出现故障，那么必须启动队列共享组中的所有队列管理器，然后才可以发出 RECOVER CFSTRUCT 命令。在发生故障的 CF 结构的恢复期间，所有队列管理器都可以继续使用本地队列以及其他 CF 结构中的队列。
- 如果 CF 应用程序结构的 CFLEVEL 为 3 或更高级别，并且 RECOVER 设置为 NO，那么对该结构中定义的对队列执行的下一个 MQOPEN 请求将自动地重新分配该结构。由于该结构只能包含非持久消息，因此所有消息都将丢失。
- 如果 CF 应用程序结构的 CFLEVEL 小于 3，那么队列管理器将发生故障。队列管理器重新启动时，对等恢复将尝试连接到该结构，这将检测到该结构已发生故障并分配新版本的结构。受耦合设施故障影响的 CF 结构中共享队列的全部消息都将丢失。

自 IBM WebSphere MQ 7.1 起，队列共享组中的队列管理器已容许与耦合设施结构的连接发生中断而不出现故障。如果结构遇到连接故障，那么会尝试在另一个耦合设施中使用更好的连接重建结构，以便能够尽快重新访问共享队列。

长时间运行的工作单元的问题

本主题用于调查和解决长时间运行的工作单元问题。

本主题说明在重新启动时遇到长时间运行的工作单元时应当如何操作。在此上下文中，这表示某工作单元已长时间处于活动状态（可能数天甚至数周），以至于该工作单元的原始 RBA 超出了当前活动日志的范围。这意味着重新启动可能相当耗时，这是因为，必须读取与该工作单元相关的所有日志记录，这可能涉及读取归档日志。

重新启动期间找到旧工作单元

问题

在重新启动期间，检测到某个工作单元的原始 RBA 早于最旧的活动日志。

故障现象

IBM MQ 发出以下消息：

```
CSQR020I +CSQ1 OLD UOW FOUND
```

系统操作

显示关于该工作单元的信息，发出消息 CSQR021D，从而请求操作员作出响应。

系统程序员操作

无。

操作员操作

确定是否落实该工作单元。如果您选择不落实工作单元，那么该工作单元将由正常的重新启动恢复处理所处理。由于此工作单元已旧，有可能需要使用归档日志，因此完成时间将较长。

IMS 相关问题

使用本主题调查并解决与 IMS 和 IBM MQ 相关的问题。

本主题中包含有关您可能在 IMS 环境中遇到的以下问题的计划：

- [第 398 页的『IMS 无法连接到 IBM MQ』](#)
- [第 399 页的『IMS 应用程序问题』](#)
- [第 399 页的『IMS 无法运作』](#)

IMS 无法连接到 IBM MQ

问题

IMS 适配器无法连接到 IBM MQ。

故障现象

IMS 保持运行状态。IMS 适配器针对控制器区域连接发出以下消息：

- CSQQ001I
- CSQQ002E
- CSQQ003E
- CSQQ004E
- CSQQ005E
- CSQQ007E

有关详细信息，请参阅 [IBM MQ for z/OS 消息](#)，[完成和原因码文档](#)。

如果 IMS 应用程序在 IMS 适配器无法连接时尝试访问 IBM MQ，那么它可以接收完成代码和原因码，也可以异常终止。这取决于 IMS PROCLIB 的 SSM 成员中 REO 选项的值。

系统操作

IMS 消息中也会报告所有连接错误 DFS3611.

系统程序员操作

无。

操作员操作

分析并更正问题，然后通过 IMS 命令重新启动连接：

```
/START SUBSYS subsysname
```

IMS 请求适配器解析不确定的恢复单元。

IMS 应用程序问题

问题

IMS 应用程序异常终止。

故障现象

向用户终端发送以下消息：

```
DFS555I TRANSACTION tran-id ABEND abcode  
MSG IN PROCESS: message data:
```

其中 *tran-id* 表示正在异常终止的任何 IMS 事务，*abcode* 表示异常终止代码。

系统操作

IMS 请求适配器解析恢复单元。IMS 保持连接到 IBM MQ。

系统程序员操作

无。

操作员操作

按照 IMS 主终端上 DFS554A 消息中的指示操作。

IMS 无法运作

问题

IMS 无法运作。

故障现象

有可能出现多个症状：

- IMS 等待或循环

IBM MQ 无法在 IMS 中检测到等待或循环，您必须找到等待或循环的源。该源可能是 IMS、IMS 应用程序或 IMS 适配器。

- IMS 异常终止。

- 请参阅 *IMS/ESA Messages and Codes* 和 *IMS/ESA Failure Analysis Structure Tables* 手册，以获取更多信息。
- 如果线程在 IMS 终止时连接到 IBM MQ，那么 IBM MQ 将发出消息 CSQ3201E。此消息表示已运行 IBM MQ 结束任务 (EOT) 例程，以清除和断开任何已连接的线程。

系统操作

IBM MQ 将检测 IMS 错误并：

- 回退未完成的工作。
- 保存重新连接 IMS 时要解析的不确定恢复单元。

系统程序员操作

无。

操作员操作

解析并更正导致 IMS 异常终止的问题，然后执行紧急重新启动 IMS。紧急重新启动将执行下列操作：

- 回退更改了 IMS 资源的未完成事务。

- 记录可能不确定并具有 IBM MQ 访问权的事务。

您可能需要使用 IMS 命令重新启动与 IBM MQ 的连接:

```
/START SUBSYS subsysname
```

在启动期间, IMS 请求适配器解析不确定的恢复单元。

z/OS 硬件问题

本主题是调查硬件问题的着手点。

如果硬件错误导致数据不可读,那么仍可以使用 介质恢复 方法来恢复 IBM MQ :

1. 要恢复数据,需要该数据的备份副本。请定期使用 DFDSS 或访问方法服务 REPRO 来创建数据的副本。
2. 恢复最新的备份副本。
3. 重新启动队列管理器。

备份副本越新,就能够越快地使子系统变为再次可用。

队列管理器重新启动时,它将使用归档日志来恢复创建该备份副本之后所作的更改。您必须保留足够多的归档日志才能使 IBM MQ 完全恢复更改。在存在包含归档日志中的所有更改的备份副本之前,请勿删除该归档日志。

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 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: