

9.4

*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 499 页的『声明』中的信息。

本版本适用于 IBM® MQ V 9 发行版 4 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

<b>故障诊断与支持</b> .....	<b>5</b>
进行初始检查.....	6
在 AIX 上执行初步检查.....	7
在 IBM i 上执行初步检查.....	17
在 Linux 上执行初步检查.....	25
在 Windows 上执行初步检查.....	36
Making initial checks on z/OS.....	43
详细故障诊断.....	57
对 AMQP 问题进行故障诊断.....	57
对 AMS 问题进行故障诊断.....	61
对命令问题进行故障诊断.....	62
对分布式发布/预订问题进行故障诊断.....	65
对分布式队列管理问题进行故障诊断.....	69
对 IBM MQ Console 和 REST API 问题进行故障诊断.....	78
对 IBM MQ Internet Pass-Thru 问题进行故障诊断.....	81
对 IBM MQ MQI client 应用程序问题进行故障诊断.....	84
IBM MQ .NET 问题故障诊断.....	86
对 Java 和 JMS 问题进行故障诊断.....	87
对 Managed File Transfer 问题进行故障诊断.....	110
对消息问题进行故障诊断.....	159
对 MQ Telemetry 问题进行故障诊断.....	159
多点广播问题故障诊断.....	172
对队列管理器问题进行故障诊断.....	174
对队列管理器集群问题进行故障诊断.....	175
对 RDQM 配置问题进行故障诊断.....	194
对安全性问题进行故障诊断.....	208
对 IBM MQ 问题的 WCF 定制通道进行故障诊断.....	220
对 XMS .NET 问题进行故障诊断.....	222
Troubleshooting IBM MQ for z/OS problems.....	224
联系 IBM 支持人员.....	279
为 IBM 支持人员收集故障诊断信息.....	280
使用错误日志.....	367
AIX, Linux, and Windows 上的错误日志.....	369
IBM i 上的错误日志.....	372
Error logs on z/OS.....	375
IBM MQ classes for JMS 中的错误日志.....	375
在 Multiplatforms 上禁止错误日志中的通道错误消息.....	375
First Failure Support Technology (FFST).....	376
FFST: IBM MQ for AIX or Linux.....	377
FFST: IBM MQ for IBM i.....	378
FFST: IBM MQ for Windows.....	380
FFST: IBM MQ classes for JMS.....	382
FFST:WCF XMS First Failure Support Technology.....	386
XMS .NET 应用程序的 FFDC 配置.....	387
跟踪.....	387
AIX and Linux 上的跟踪.....	388
IBM i 上的跟踪.....	393
Windows 上的跟踪.....	402
Tracing on z/OS.....	405
跟踪高级消息排队协议 (AMQP) 服务.....	422
跟踪 IBM MQ Console.....	424
跟踪 IBM MQ Internet Pass-Thru 中的错误.....	427

跟踪 IBM MQ.NET 应用程序.....	428
跟踪 JMS/Jakarta Messaging 和 Java 应用程序.....	433
在多平台上跟踪 Managed File Transfer 资源.....	443
Tracing Managed File Transfer for z/OS resources.....	449
跟踪 REST API.....	463
跟踪 <b>runmqakm</b> .....	464
跟踪 IBM MQ 的 WCF 定制通道.....	465
跟踪 XMS .NET 应用程序.....	465
启用 LDAP 客户机库代码的动态跟踪.....	471
故障后恢复.....	472
磁盘驱动器故障.....	473
损坏的队列管理器对象.....	473
损坏的单个对象.....	473
自动介质恢复故障.....	474
Example recovery procedures on z/OS.....	474
<b>声明.....</b>	<b>499</b>
编程接口信息.....	500
商标.....	500



# 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 279 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 472 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 367 页的『使用错误日志』](#)
- [第 376 页的『First Failure Support Technology \(FFST\)』](#)
- [第 387 页的『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 57 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 279 页的『联系 IBM 支持人员』。

## 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 AIX 上执行初步检查』](#)
-  [第 17 页的『在 IBM i 上执行初步检查』](#)
-  [第 25 页的『在 Linux 上执行初步检查』](#)
-  [第 36 页的『在 Windows 上执行初步检查』](#)
-  [第 43 页的『Making initial checks on z/OS』](#)

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 369 页的『AIX, Linux, and Windows 上的错误日志』](#)
  -  [第 372 页的『IBM i 上的错误日志』](#)
  -  [第 231 页的『Diagnostic information produced on IBM MQ for z/OS』](#)
- 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 Multiplatforms 版上更改 IBM MQ 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 387 页的『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 387 页的『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 367 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 279 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 387 页的『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。

## AIX 在 AIX 上执行初步检查

开始在 AIX 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

当您浏览要考虑的初始问题列表并遵循更多信息的链接时，请记住可能与该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。

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)。有关更多信息，请参阅第 280 页的『为 IBM 支持人员收集故障诊断信息』。

## 过程

### 1. 确定问题的特征

您可以考虑一些初始问题，以帮助您确定问题的原因：

- IBM MQ 之前是否已成功运行？
- 自上次成功运行以来是否进行了任何更改？
- 您是否应用了任何维护更新？
- 应用程序以前运行成功吗？
- 在某些命令的描述性文本中使用特殊字符时，是否会收到错误？
- 是否有任何错误消息或返回码可帮助您确定问题的位置和原因？
- 是否可以重现问题？
- 问题是否影响网络的特定部分？
- 问题是否在一天中的特定时间发生？
- 问题是否为间歇性？

### 2. 调查应用程序，命令和消息的潜在问题

如果迁到有关 IBM MQ 应用程序，命令和消息的问题，那么可以考虑一些问题来帮助您确定问题的原因：

- 消息是否未能到达队列？
- 消息包含意外或损坏的信息吗？
- 使用分布式队列时是否收到意外的消息？
- 您是否未收到来自 PCF 命令的响应？
- 是否只有部分队列失败？

- [问题是否仅影响远程队列?](#)
  - [应用程序或系统运行缓慢吗?](#)
3. 调查与 IBM MQ 资源连接的任何问题，包括 IBM MQ 进程的资源使用情况，与资源不足相关的问题以及资源限制配置。  
有关更多信息，请参阅第 14 页的『[针对 AIX 上的资源问题的其他检查](#)』。
4. 如果需要更多信息来帮助您确定问题的原因，请参阅第 57 页的『[详细故障诊断](#)』。

### 相关任务

第 279 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 17 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 25 页的『[在 Linux 上执行初步检查](#)』

开始在 Linux 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 36 页的『[在 Windows 上执行初步检查](#)』

开始在 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 43 页的『[Making initial checks on z/OS](#)』

Before you start problem determination in detail on z/OS, consider whether there is an obvious cause of the problem, or an area of investigation that is likely to give useful results. This approach to diagnosis can often save a lot of work by highlighting a simple error, or by narrowing down the range of possibilities.

### 相关参考

[消息和原因码](#)

## 在 AIX 上确定问题特征

需要考虑的一些初始问题，以帮助确定问题的原因。

### 关于此任务

使用以下问题作为提示来帮助确定问题原因：

- [IBM MQ 之前是否已成功运行?](#)
- [自上次成功运行以来是否进行了任何更改?](#)
- [您是否应用了任何维护更新?](#)
- [应用程序以前运行成功吗?](#)
- [在某些命令的描述性文本中使用特殊字符时，是否会收到错误?](#)
- [是否有任何错误消息或返回码可帮助您确定问题的位置和原因?](#)
- [是否可以重现问题?](#)
- [问题是否影响网络的特定部分?](#)
- [问题是否在一天中的特定时间发生?](#)
- [问题是否为间歇性?](#)

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。即使您的观察结果没有直接表明原因，但如果您需要进行系统的问题确定练习，那么这些观察结果可能会在稍后有用。

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)。有关更多信息，请参阅第 280 页的『[为 IBM 支持人员收集故障诊断信息](#)』。

## 过程

### 1. 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。有关更多信息，请参阅 [IBM MQ 安装概述](#) 和 [在 AIX 上安装和卸载 IBM MQ](#)。

要运行验证过程，请参阅 [在 AIX 上验证 IBM MQ 安装](#)。另请参阅 [配置 IBM MQ](#) 以获取有关 IBM MQ 安装后配置的信息。

### 2. 自上次成功运行以来是否进行了任何更改？

对 IBM MQ 配置进行的更改或对与 IBM MQ 交互的其他应用程序进行的更改可能是问题的原因。

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？
- 是否更改了操作系统中可能影响 IBM MQ 操作的任何组件？

### 3. 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 支持人员联系以了解任何维护包错误。
- 如果已将维护包应用于任何其他应用程序，请考虑它对 IBM MQ 与其交互的方式可能产生的影响。

### 4. 应用程序以前运行成功吗？

如果问题似乎涉及一个特定应用程序，请考虑该应用程序之前是否已成功运行：

- 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？

在查看代码之前，根据代码所使用的编程语言，检查转换程序或编译器和链接编辑器的输出，以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。有关构建应用程序的信息，请参阅 [开发应用程序](#)。

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



- 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。

5. 在某些命令的描述性文本中使用特殊字符时，是否会收到错误？

某些字符，例如反斜杠 (\) 和双引号 (") 与命令配合使用时，字符具有特殊含义。

在特殊字符前面加上 \，即，如果您需要文本中的 \ 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](#)。

6. 是否有任何错误消息或返回码可帮助您确定问题的位置和原因？

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。有关错误日志的位置和内容的信息，请参阅第 369 页的『[AIX, 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。

7. 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。

8. 问题是否影响网络的特定部分？

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？

您可能能够识别出问题影响的网络特定部分 (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。

9. 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 网络跨多个时区扩展，那么高峰系统装入可能似乎在一天中的其他时间发生。

10. 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。

### 相关概念

第 14 页的『[针对 AIX 上的资源问题的其他检查](#)』

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。

## 相关任务

第 279 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 11 页的『在 AIX 上确定应用程序，命令和消息的问题』

如果迂到有关 IBM MQ 应用程序，命令和消息的问题，那么可以考虑一些问题来帮助您确定问题的原因。

## 相关参考

[消息和原因码](#)

## 在 AIX 上确定应用程序，命令和消息的问题

如果迂到有关 IBM MQ 应用程序，命令和消息的问题，那么可以考虑一些问题来帮助您确定问题的原因。

## 关于此任务

使用以下问题作为提示来帮助确定问题原因：

- [消息是否未能到达队列？](#)
- [消息包含意外或损坏的信息吗？](#)
- [使用分布式队列时是否收到意外的消息？](#)
- [您是否未收到来自 PCF 命令的响应？](#)
- [是否只有部分队列失败？](#)
- [问题是否仅影响远程队列？](#)
- [应用程序或系统运行缓慢吗？](#)

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。即使您的观察结果没有直接表明原因，但如果您需要进行系统的问题确定练习，那么这些观察结果可能会在稍后有用。

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)。有关更多信息，请参阅第 280 页的『为 IBM 支持人员收集故障诊断信息』。

## 过程

### 1. 消息是否未能到达队列？

如果期望消息时消息未到达，请检查消息是否已成功放入队列中：

- 队列正确定义了吗？例如，**MAXMSGL** 是否足够大？
- 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？

还要检查您是否能够从队列中获取任何消息：

- 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。

## 2. 消息包含意外或损坏的信息吗？

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经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。

## 3. 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：

- 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 (**PING QMGR**) 或通道 (**PING CHANNEL**) 使用 MQSC 命令 **PING** 来验证链接是否可操作。
- 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。有关更多信息，请参阅 [使用死信 \(未传递的消息\) 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。有关更多信息，请参阅 [数据转换](#)。

如果无法解决问题，请联系 IBM 支持人员以获取帮助。

## 4. 您是否未收到来自 PCF 命令的响应？

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？使用 **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 **strmqcsv** 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。有关更多信息，请参阅 [MQDLH-死信头](#) 和 [使用死信 \(未传递的消息\) 队列](#)。如果死信队列包含消息，那么可以使用提供的浏览样本应用程序 (`amqsbcg`) 通过 `MQGET` 调用来浏览消息。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？有关更多信息，请参阅第 371 页的『[AIX, Linux, and Windows 上的错误日志目录](#)』。
- 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` 字段以及来自 `MQGET` 的完成代码和原因码的信息。
- 如果您正在使用自己的应用程序将命令放到 `SYSTEM.ADMIN.COMMAND.QUEUE`，是否需要获取同步点？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。如果在重新启动后仍发生此问题，请联系 IBM 支持人员以获取帮助。

#### 5. 是否只有部分队列失败？

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。

使用 `MQSC` 命令 **DISPLAY QUEUE** 来显示有关每个队列的信息。如果 `CURDEPTH` 位于 `MAXDEPTH`，那么表示未处理队列。检查所有应用程序都是正常运行的。

如果 `CURDEPTH` 不在 `MAXDEPTH` 上，请检查以下队列属性以确保它们正确：

- 如果正在使用触发，那么触发器监视器是否正在运行？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。这可能是需要启动应用程序，连接已中断，或者 `MQOPEN` 调用由于某种原因而失败。检查队列属性 `IPPROCS` 和 `OPPROCS`。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。值可能已更改，或者队列可能已打开但现在已关闭。

检查您期望放入或获取消息时的状态。

如果无法解决问题，请联系 IBM 支持人员以获取帮助。

#### 6. 问题是否仅影响远程队列？

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。

#### 7. 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。

导致应用程序性能下降或在队列（通常是传输队列）上构建消息的常见原因是一个或多个应用程序在工作单元外部写入持久消息。有关更多信息，请参阅 [消息持久性](#)。

如果性能问题仍然存在，那么问题可能在于 IBM MQ 自身。如果您怀疑这样做，请联系 IBM 支持人员以获取帮助。

## 相关概念

第 14 页的『[针对 AIX 上的资源问题的其他检查](#)』

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。

## 相关任务

第 279 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 8 页的『[在 AIX 上确定问题特征](#)』

需要考虑的一些初始问题，以帮助确定问题的原因。

## 相关参考

[消息和原因码](#)

## 针对 AIX 上的资源问题的其他检查

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`。

## 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：

– **AIX** IBM MQ 进程数：

```
ps -elf|grep "amq|rqn"|wc -l
```

– 连接数：

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

– 共享内存使用量：

```
ipcs -ma
```

- 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：
  - 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
- 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')。

#### 解决 AIX 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

检查并提高最大用户进程数和堆栈大小用户进程资源限制。

## 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。

## 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
```

## 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。

#### 相关任务

第 8 页的『[在 AIX 上确定问题特征](#)』

需要考虑的一些初始问题，以帮助确定问题的原因。

第 11 页的『[在 AIX 上确定应用程序，命令和消息的问题](#)』

如果迁到有关 IBM MQ 应用程序，命令和消息的问题，那么可以考虑一些问题来帮助确定问题的原因。

第 279 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。

## 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。

如果您无法通过执行初步检查来确定问题的原因，因此需要进行更详细的调查，那么在子主题中还有其他问题供您考虑。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。

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)。有关更多信息，请参阅第 280 页的『为 IBM 支持人员收集故障诊断信息』。

## 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。有关更多信息，请参阅 [IBM MQ 安装概述](#) 和 [在 IBM i 上安装和卸载 IBM MQ](#)。有关运行验证过程的信息，请参阅 [在 IBM i 上验证 IBM MQ 安装](#)。

### 2. IBM MQ 应用程序以前运行成功吗？

#### Yes

继续执行步骤 [第 18 页的『3』](#)。

#### 否

请考虑应用程序是否可能未能编译或链接，如果尝试调用该应用程序，那么会失败。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，或者参阅 [开发应用程序](#) 以获取有关如何构建应用程序的信息。

还要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. 应用程序是否收到意外的 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
- 操作过程中的更改

### 相关任务

[第 279 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 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 23 页的『在 IBM i 上确定应用程序，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。

[第 7 页的『在 AIX 上执行初步检查』](#)

开始在 AIX 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 25 页的『在 Linux 上执行初步检查』](#)

开始在 Linux 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 36 页的『在 Windows 上执行初步检查』](#)

开始在 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 43 页的『Making initial checks on z/OS』](#)

Before you start problem determination in detail on z/OS, consider whether there is an obvious cause of the problem, or an area of investigation that is likely to give useful results. This approach to diagnosis can often save a lot of work by highlighting a simple error, or by narrowing down the range of possibilities.

## 相关参考

消息和原因码

## 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。

### 关于此任务

使用以下问题作为提示来帮助确定问题原因：

- [是否可以重现问题？](#)
- [问题是否为间歇性？](#)
- [在命令中使用特殊字符时是否收到错误？](#)
- [问题是否影响网络的特定部分？](#)
- [问题是否影响网络的特定部分？](#)
- [问题是否在一天中的特定时间发生？](#)
- [问题是否在一天中的特定时间发生？](#)
- [您是否未收到来自命令的响应？](#)

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。即使您的观察结果没有直接表明原因，但如果您需要进行系统的问题确定练习，那么这些观察结果可能会在稍后有用。

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)。有关更多信息，请参阅第 280 页的『[为 IBM 支持人员收集故障诊断信息](#)』。

### 过程

#### 1. 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？

#### 2. 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。

#### 3. 在命令中使用特殊字符时是否收到错误？

要避免特殊字符的潜在问题，请在包含特殊字符时务必小心，例如反斜杠 (\) 和引号 (") 字符，在某些命令的描述性文本中。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。

4. 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？

5. 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。

6. 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](https://www.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_MQ)，以确保应用所有相关的 PTF。

7. 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。）

8. 您是否未收到来自命令的响应？

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 (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？有关更多信息，请参阅第 372 页的『IBM i 上的错误日志』。
- 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 279 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 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 23 页的『[在 IBM i 上确定应用程序，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。

## 相关参考

[消息和原因码](#)

## 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(QMQADM) 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 在 IBM i 上确定应用程序，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。

### 关于此任务

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？](#)
- [问题是否仅影响远程队列？](#)
- [消息是否未能到达队列？](#)
- [消息包含意外或损坏的信息吗？](#)
- [使用分布式队列时是否收到意外的消息？](#)

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。即使您的观察结果没有直接表明原因，但如果您需要进行系统的问题确定练习，那么这些观察结果可能会在稍后有用。

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)。有关更多信息，请参阅第 280 页的『[为 IBM 支持人员收集故障诊断信息](#)』。

### 过程

#### 1. 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。使用 [WRKMQMST](#) 或 [DSPMQM](#) 显示有关此队列的信息，然后使用显示的数据执行以下检查：

- 如果 **CURDEPTH** 位于 **MAXDEPTH**，那么不会处理该队列。检查所有应用程序都是正常运行的。
- 如果 **CURDEPTH** 未处于 **MAXDEPTH**，请检查以下队列属性以确保它们正确：
  - 如果正在使用触发，那么触发器监视器是否正在运行？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？
- 如果没有应用程序进程从队列获取消息，请确定原因。例如，这可能是由于必须启动应用程序，连接已中断，或者由于 MQOPEN 调用由于某种原因而失败。

如果无法解决问题，请联系 IBM 支持人员以获取帮助。

#### 2. 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：

- 确保应该将消息放入远程队列的程序成功运行。
- 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。
- 若有必要，手动启动通道。请参阅[分布式排队和集群](#)。

d. 使用 **PING** 命令检查通道。

### 3. 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？
  - 是否正确定义了队列，例如，**MAXMSGLEN** 是否足够大？
  - 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？如果是这样，请参阅步骤第 24 页的『5』。

### 4. 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**，或者消息格式是否由于数据转换而不正确？

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。

### 5. 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](#)。 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](#)。

### 相关任务

第 279 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 20 页的『[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。

### 相关参考

[消息和原因码](#)

## Linux 在 Linux 上执行初步检查

开始在 Linux 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

当您浏览要考虑的初始问题列表并遵循更多信息的链接时，请记下可能与该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。

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)。有关更多信息，请参阅第 280 页的『[为 IBM 支持人员收集故障诊断信息](#)』。

### 过程

#### 1. 确定问题的特征

您可以考虑一些初始问题，以帮助确定问题的原因：

- [IBM MQ 之前是否已成功运行？](#)
- [自上次成功运行以来是否进行了任何更改？](#)
- [您是否应用了任何维护更新？](#)
- [应用程序以前运行成功吗？](#)
- [在某些命令的描述性文本中使用特殊字符时，是否会收到错误？](#)
- [是否有任何错误消息或返回码可帮助您确定问题的位置和原因？](#)



- [是否可以重现问题?](#)
  - [问题是否影响网络的特定部分?](#)
  - [问题是否在一天中的特定时间发生?](#)
  - [问题是否为间歇性?](#)
2. 调查应用程序，命令和消息的潜在问题
- 如果迁到有关 IBM MQ 应用程序，命令和消息的问题，那么可以考虑一些问题来帮助您确定问题的原因：
- [消息是否未能到达队列?](#)
  - [消息包含意外或损坏的信息吗?](#)
  - [使用分布式队列时是否收到意外的消息?](#)
  - [您是否未收到来自 PCF 命令的响应?](#)
  - [是否只有部分队列失败?](#)
  - [问题是否仅影响远程队列?](#)
  - [应用程序或系统运行缓慢吗?](#)
3. 调查与 IBM MQ 资源连接的任何问题，包括 IBM MQ 进程的资源使用情况，与资源不足相关的问题以及资源限制配置。
- 有关更多信息，请参阅第 32 页的『[针对 Linux 上的资源问题的其他检查](#)』。
4. 如果需要更多信息来帮助您确定问题的原因，请参阅第 57 页的『[详细故障诊断](#)』。

### 相关任务

第 279 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 7 页的『[在 AIX 上执行初步检查](#)』

开始在 AIX 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 17 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 36 页的『[在 Windows 上执行初步检查](#)』

开始在 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 43 页的『[Making initial checks on z/OS](#)』

Before you start problem determination in detail on z/OS, consider whether there is an obvious cause of the problem, or an area of investigation that is likely to give useful results. This approach to diagnosis can often save a lot of work by highlighting a simple error, or by narrowing down the range of possibilities.

### 相关参考

[消息和原因码](#)

## 在 Linux 上确定问题特征

需要考虑的一些初始问题，以帮助确定问题的原因。

### 关于此任务

使用以下问题作为提示来帮助确定问题原因：

- [IBM MQ 之前是否已成功运行?](#)
- [自上次成功运行以来是否进行了任何更改?](#)
- [您是否应用了任何维护更新?](#)

- [应用程序以前运行成功吗？](#)
- [在某些命令的描述性文本中使用特殊字符时，是否会收到错误？](#)
- [是否有任何错误消息或返回码可帮助您确定问题的位置和原因？](#)
- [是否可以重现问题？](#)
- [问题是否影响网络的特定部分？](#)
- [问题是否在一天中的特定时间发生？](#)
- [问题是否为间歇性？](#)

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。即使您的观察结果没有直接表明原因，但如果您需要进行系统的问题确定练习，那么这些观察结果可能会在稍后有用。

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)。有关更多信息，请参阅第 280 页的『[为 IBM 支持人员收集故障诊断信息](#)』。

## 过程

### 1. 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。有关更多信息，请参阅 [IBM MQ 安装概述](#) 和 [在 Linux 上安装和卸载 IBM MQ](#)。

要运行验证过程，请参阅 [在 Linux 上验证 IBM MQ 安装](#)。另请参阅 [配置 IBM MQ](#) 以获取有关 IBM MQ 安装后配置的信息。

### 2. 自上次成功运行以来是否进行了任何更改？

对 IBM MQ 配置进行的更改或对与 IBM MQ 交互的其他应用程序进行的更改可能是问题的原因。

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？
- 是否更改了操作系统中可能影响 IBM MQ 操作的任何组件？

### 3. 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 支持人员联系以了解任何维护包错误。
- 如果已将维护包应用于任何其他应用程序，请考虑它对 IBM MQ 与其交互的方式可能产生的影响。

### 4. 应用程序以前运行成功吗？

如果问题似乎涉及一个特定应用程序，请考虑该应用程序之前是否已成功运行：

- 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？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？

在查看代码之前，根据代码所使用的编程语言，检查转换程序或编译器和链接编辑器的输出，以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。有关构建应用程序的信息，请参阅 [开发应用程序](#)。

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？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。

5. 在某些命令的描述性文本中使用特殊字符时，是否会收到错误？

某些字符，例如反斜杠 (\) 和双引号 (") 与命令配合使用时，字符具有特殊含义。

在特殊字符前面加上 \，即，如果您需要文本中的 \ 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](#)。

6. 是否有任何错误消息或返回码可帮助您确定问题的位置和原因？

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。有关错误日志的位置和内容信息，请参阅 [第 369 页的『AIX, 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。

7. 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。

8. 问题是否影响网络的特定部分？

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？

您可能能够识别出问题影响的网络特定部分 (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。



## 9. 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 网络跨多个时区扩展，那么高峰系统装入可能似乎在一天中的其他时间发生。

## 10. 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。

### 相关概念

第 32 页的『[针对 Linux 上的资源问题的其他检查](#)』

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。

### 相关任务

第 279 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 29 页的『[在 Linux 上确定应用程序，命令和消息的问题](#)』

如果迁到有关 IBM MQ 应用程序，命令和消息的问题，那么可以考虑一些问题来帮助您确定问题的原因。

### 相关参考

[消息和原因码](#)

## Linux 在 Linux 上确定应用程序，命令和消息的问题

如果迁到有关 IBM MQ 应用程序，命令和消息的问题，那么可以考虑一些问题来帮助您确定问题的原因。

### 关于此任务

使用以下问题作为提示来帮助确定问题原因：

- [消息是否未能到达队列？](#)
- [消息包含意外或损坏的信息吗？](#)
- [使用分布式队列时是否收到意外的消息？](#)
- [您是否未收到来自 PCF 命令的响应？](#)
- [是否只有部分队列失败？](#)
- [问题是否仅影响远程队列？](#)
- [应用程序或系统运行缓慢吗？](#)

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。即使您的观察结果没有直接表明原因，但如果您需要进行系统的问题确定练习，那么这些观察结果可能会在稍后有用。

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)。有关更多信息，请参阅第 280 页的『[为 IBM 支持人员收集故障诊断信息](#)』。

### 过程

#### 1. 消息是否未能到达队列？

如果期望消息时消息未到达，请检查消息是否已成功放入队列中：

- 队列正确定义了吗？例如，**MAXMSGL** 是否足够大？
- 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？

还要检查您是否能够从队列中获取任何消息:

- 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。

## 2. 消息包含意外或损坏的信息吗?

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。

## 3. 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:

- 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 (**PING QMGR**) 或通道 (**PING CHANNEL**) 使用 MQSC 命令 **PING** 来验证链接是否可操作。
- 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。有关更多信息，请参阅 [使用死信 \(未传递的消息\) 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。有关更多信息，请参阅 [数据转换](#)。

如果无法解决问题，请联系 IBM 支持人员以获取帮助。

#### 4. 您是否未收到来自 PCF 命令的响应？

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？使用 **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 **strmqcsv** 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。有关更多信息，请参阅 [MQDLH-死信头](#) 和 [使用死信 \(未传递的消息\) 队列](#)。如果死信队列包含消息，那么可以使用提供的浏览样本应用程序 (amqsbcg) 通过 MQGET 调用来浏览消息。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？有关更多信息，请参阅第 371 页的『[AIX, Linux, and Windows 上的错误日志目录](#)』。
- 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* 字段以及来自 MQGET 的完成代码和原因码的信息。
- 如果您正在使用自己的应用程序将命令放到 SYSTEM.ADMIN.COMMAND.QUEUE，是否需要获取同步点？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。如果在重新启动后仍发生此问题，请联系 IBM 支持人员以获取帮助。

#### 5. 是否只有部分队列失败？

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。

使用 MQSC 命令 **DISPLAY QUEUE** 来显示有关每个队列的信息。如果 **CURDEPTH** 位于 **MAXDEPTH**，那么表示未处理队列。检查所有应用程序都是正常运行的。

如果 **CURDEPTH** 不在 **MAXDEPTH** 上，请检查以下队列属性以确保它们正确：

- 如果正在使用触发，那么触发器监视器是否正在运行？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。这可能是需要启动应用程序，连接已中断，或者 MQOPEN 调用由于某种原因而失败。检查队列属性 **IPPROCS** 和 **OPPROCS**。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。值可能已更改，或者队列可能已打开但现在已关闭。

检查您期望放入或获取消息时的状态。

如果无法解决问题，请联系 IBM 支持人员以获取帮助。

#### 6. 问题是否仅影响远程队列？

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。

#### 7. 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。

导致应用程序性能下降或在队列（通常是传输队列）上构建消息的常见原因是一个或多个应用程序在工作单元外部写入持久消息。有关更多信息，请参阅 [消息持久性](#)。

如果性能问题仍然存在，那么问题可能在于 IBM MQ 自身。如果您怀疑这样做，请联系 IBM 支持人员以获取帮助。

### 相关概念

[第 32 页的『针对 Linux 上的资源问题的其他检查』](#)

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。

### 相关任务

[第 26 页的『在 Linux 上确定问题特征』](#)

需要考虑的一些初始问题，以帮助确定问题的原因。

[第 279 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

[消息和原因码](#)

## Linux 针对 Linux 上的资源问题的其他检查

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`。

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：

- IBM MQ 进程数：

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

- 连接数：

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

- 共享内存使用量：

```
ipcs -ma
```

- 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：
  - 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. 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')。

## 解决 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

检查并提高最大用户进程数和堆栈大小用户进程资源限制。

### Linux 上需要其他配置

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。

## 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
```

## 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:
amqmpa 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。

### 相关任务

第 26 页的『[在 Linux 上确定问题特征](#)』

需要考虑的一些初始问题，以帮助确定问题的原因。

第 29 页的『[在 Linux 上确定应用程序，命令和消息的问题](#)』

如果迂到有关 IBM MQ 应用程序，命令和消息的问题，那么可以考虑一些问题来帮助您确定问题的原因。

第 279 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。

## Windows 在 Windows 上执行初步检查

开始在 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

当您浏览要考虑的初始问题列表并遵循更多信息的链接时，请记下可能与该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。

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)。有关更多信息，请参阅第 280 页的『[为 IBM 支持人员收集故障诊断信息](#)』。

### 过程

#### 1. 确定问题的特征

您可以考虑一些初始问题，以帮助您确定问题的原因：

- [IBM MQ 之前是否已成功运行？](#)
- [自上次成功运行以来是否进行了任何更改？](#)
- [您是否应用了任何维护更新？](#)
- [应用程序以前运行成功吗？](#)
- [在某些命令的描述性文本中使用特殊字符时，是否会收到错误？](#)
- [是否有任何错误消息或返回码可帮助您确定问题的位置和原因？](#)
- [是否可以重现问题？](#)
- [问题是否影响网络的特定部分？](#)
- [问题是否在一天中的特定时间发生？](#)
- [问题是否为间歇性？](#)

#### 2. 调查应用程序，命令和消息的潜在问题

如果迂到有关 IBM MQ 应用程序，命令和消息的问题，那么可以考虑一些问题来帮助您确定问题的原因：

- [消息是否未能到达队列？](#)
- [消息包含意外或损坏的信息吗？](#)



- [使用分布式队列时是否收到意外的消息?](#)
- [您是否未收到来自 PCF 命令的响应?](#)
- [是否只有部分队列失败?](#)
- [问题是否仅影响远程队列?](#)
- [在 Windows 上创建或启动队列管理器时是否接收到错误代码?](#)
- [应用程序或系统运行缓慢吗?](#)

3. 如果需要更多信息来帮助您确定问题的原因, 请参阅 [第 57 页的『详细故障诊断』](#)。

### 相关任务

[第 279 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 7 页的『在 AIX 上执行初步检查』](#)

开始在 AIX 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 17 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 25 页的『在 Linux 上执行初步检查』](#)

开始在 Linux 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 43 页的『Making initial checks on z/OS』](#)

Before you start problem determination in detail on z/OS, consider whether there is an obvious cause of the problem, or an area of investigation that is likely to give useful results. This approach to diagnosis can often save a lot of work by highlighting a simple error, or by narrowing down the range of possibilities.

### 相关参考

[消息和原因码](#)

## 在 Windows 上确定问题特征

需要考虑的一些初始问题, 以帮助确定问题的原因。

### 关于此任务

使用以下问题作为提示来帮助确定问题原因:

- [IBM MQ 之前是否已成功运行?](#)
- [自上次成功运行以来是否进行了任何更改?](#)
- [您是否应用了任何维护更新?](#)
- [应用程序以前运行成功吗?](#)
- [在某些命令的描述性文本中使用特殊字符时, 是否会收到错误?](#)
- [是否有任何错误消息或返回码可帮助您确定问题的位置和原因?](#)
- [是否可以重现问题?](#)
- [问题是否影响网络的特定部分?](#)
- [问题是否在一天中的特定时间发生?](#)
- [问题是否为间歇性?](#)

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。即使您的观察结果没有直接表明原因, 但如果您需要进行系统的问题确定练习, 那么这些观察结果可能会在稍后有用。

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)。有关更多信息, 请参阅 [第 280 页的『为 IBM 支持人员收集故障诊断信息』](#)。

## 过程

### 1. 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。有关更多信息，请参阅 [IBM MQ 安装概述](#) 和 [在 Windows 上安装和卸载 IBM MQ](#)。

要运行验证过程，请参阅 [在 Windows 上验证 IBM MQ 安装](#)。另请参阅 [配置 IBM MQ](#) 以获取有关 IBM MQ 安装后配置的信息。

### 2. 自上次成功运行以来是否进行了任何更改？

对 IBM MQ 配置进行的更改或对与 IBM MQ 交互的其他应用程序进行的更改可能是问题的原因。

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？
- 是否更改了操作系统中可能影响 IBM MQ 操作的任何组件？例如，是否已修改 Windows 注册表。

### 3. 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 支持人员联系以了解任何维护包错误。
- 如果已将维护包应用于任何其他应用程序，请考虑它对 IBM MQ 与其交互的方式可能产生的影响。

### 4. 应用程序以前运行成功吗？

如果问题似乎涉及一个特定应用程序，请考虑该应用程序之前是否已成功运行：

- 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些什么，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？

在查看代码之前，根据代码所使用的编程语言，检查转换程序或编译器和链接编辑器的输出，以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。有关构建应用程序的信息，请参阅 [开发应用程序](#)。

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？以下列表中的错误说明在运行 IBM MQ 程序时遇到的问题最常见原因。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。

5. 在某些命令的描述性文本中使用特殊字符时，是否会收到错误？

某些字符，例如反斜杠 (\) 和双引号 (") 与命令配合使用时，字符具有特殊含义。

在特殊字符前面加上 \，即，如果您需要文本中的 \ 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](#)。

6. 是否有任何错误消息或返回码可帮助您确定问题的位置和原因？

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。有关错误日志的位置和内容的信息，请参阅第 369 页的『[AIX, Linux, and Windows 上的错误日志](#)』。

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](#)。

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。

7. 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。

8. 问题是否影响网络的特定部分？

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？

您可能能够识别受问题影响的网络特定部分 (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。

9. 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 网络跨多个时区扩展，那么高峰系统装入可能似乎在一天中的其他时间发生。

10. 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。

## 相关任务

第 40 页的『在 Windows 上确定应用程序，命令和消息的问题』

如果迂到有关 IBM MQ 应用程序，命令和消息的问题，那么可以考虑一些问题来帮助您确定问题的原因。

第 279 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

[消息和原因码](#)

## **Windows** 在 Windows 上确定应用程序，命令和消息的问题

如果迂到有关 IBM MQ 应用程序，命令和消息的问题，那么可以考虑一些问题来帮助您确定问题的原因。

## 关于此任务

使用以下问题作为提示来帮助确定问题原因：

- [消息是否未能到达队列？](#)
- [消息包含意外或损坏的信息吗？](#)
- [使用分布式队列时是否收到意外的消息？](#)
- [您是否未收到来自 PCF 命令的响应？](#)
- [是否只有部分队列失败？](#)
- [问题是否仅影响远程队列？](#)
- [在 Windows 上创建或启动队列管理器时是否接收到错误代码？](#)
- [应用程序或系统运行缓慢吗？](#)

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。即使您的观察结果没有直接表明原因，但如果您需要进行系统的问题确定练习，那么这些观察结果可能会在稍后有用。

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)。有关更多信息，请参阅第 280 页的『为 IBM 支持人员收集故障诊断信息』。

## 过程

### 1. 消息是否未能到达队列？

如果期望消息时消息未到达，请检查消息是否已成功放入队列中：

- 队列正确定义了吗？例如，**MAXMSGL** 是否足够大？
- 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？

还要检查您是否能够从队列中获取任何消息：

- 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。

## 2. 消息包含意外或损坏的信息吗？

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经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。

## 3. 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：

- 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 (**PING QMGR**) 或通道 (**PING CHANNEL**) 使用 MQSC 命令 **PING** 来验证链接是否可操作。
- 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。有关更多信息，请参阅 [使用死信 \(未传递的消息\) 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。有关更多信息，请参阅 [数据转换](#)。

如果无法解决问题，请联系 IBM 支持人员以获取帮助。

## 4. 您是否未收到来自 PCF 命令的响应？

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？使用 **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 **strmqcsv** 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。有关更多信息，请参阅 [MQDLH-死信头](#) 和 [使用死信\(未传递的消息\)队列](#)。如果死信队列包含消息，那么可以使用提供的浏览样本应用程序 (`amqsbcg`) 通过 `MQGET` 调用来浏览消息。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？有关更多信息，请参阅第 371 页的『[AIX, Linux, and Windows 上的错误日志目录](#)』。
- 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` 字段以及来自 `MQGET` 的完成代码和原因码的信息。
- 如果您正在使用自己的应用程序将命令放到 `SYSTEM.ADMIN.COMMAND.QUEUE`，是否需要获取同步点？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。如果在重新启动后仍发生此问题，请联系 IBM 支持人员以获取帮助。

#### 5. 是否只有部分队列失败？

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。

使用 `MQSC` 命令 **DISPLAY QUEUE** 来显示有关每个队列的信息。如果 `CURDEPTH` 位于 `MAXDEPTH`，那么表示未处理队列。检查所有应用程序都是正常运行的。

如果 `CURDEPTH` 不在 `MAXDEPTH` 上，请检查以下队列属性以确保它们正确：

- 如果正在使用触发，那么触发器监视器是否正在运行？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。这可能是需要启动应用程序，连接已中断，或者 `MQOPEN` 调用由于某种原因而失败。检查队列属性 `IPPROCS` 和 `OPPROCS`。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。值可能已更改，或者队列可能已打开但现在已关闭。

检查您期望放入或获取消息时的状态。

如果无法解决问题，请联系 IBM 支持人员以获取帮助。

#### 6. 问题是否仅影响远程队列？

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。

#### 7. 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** 命令未能创建或启动队列管理器 (指示权限问题), 那么可能是因为运行 IBM MQ Windows 服务的用户没有足够的权限。

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, [Prepare IBM MQ Wizard](#) 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。

#### 8. 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。

导致应用程序性能下降或在队列 (通常是传输队列) 上构建消息的常见原因是一个或多个应用程序在工作单元外部写入持久消息。有关更多信息, 请参阅 [消息持久性](#)。

如果性能问题仍然存在, 那么问题可能在于 IBM MQ 自身。如果您怀疑这样做, 请联系 IBM 支持人员以获取帮助。

#### 相关任务

第 37 页的『[在 Windows 上确定问题特征](#)』  
需要考虑的一些初始问题, 以帮助确定问题的原因。

第 279 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

[消息和原因码](#)

## Making initial checks on z/OS

Before you start problem determination in detail on z/OS, consider whether there is an obvious cause of the problem, or an area of investigation that is likely to give useful results. This approach to diagnosis can often save a lot of work by highlighting a simple error, or by narrowing down the range of possibilities.

### About this task

The cause of your problem could be in:

- IBM MQ
- The network
- The application
- Other applications that you have configured to work with IBM MQ

As you go through the list of initial questions to consider and follow the links to more information, make a note of anything that might be relevant to the problem. Even if your observations do not suggest a cause straight away, they might be useful later if you have to carry out a systematic problem determination exercise.

### Procedure

#### 1. Identify characteristics of the problem

There are some initial questions that you can consider to help you to identify the cause of the problem:

- [Has IBM MQ for z/OS run successfully before?](#)
- [Are there any error messages, return codes or other error conditions?](#)
- [Can you reproduce the problem?](#)
- [Have you applied any APARs or PTFs?](#)
- [Have any changes been made since the last successful run?](#)
- [Has the application run successfully before?](#)
- [Does the problem affect specific parts of the network?](#)
- [Does the problem occur at specific times of the day or affect specific users?](#)
- [Is the problem intermittent or does the problem occur with all z/OS, CICS®, or IMS systems?](#)
- [Do you have a program error?](#)

## 2. Look at the problem in more detail

There are some further questions to look at when you have established that no changes have been made to your system, and that there are no problems with your application programs, but the preliminary checks have not enabled you to solve your problem.

- [Have you received some incorrect output?](#)
- [Have you received an unexpected error message or return code?](#)
- [Has there been an abend?](#)
- [Have you received no response from an MQSC command?](#)
- [Is there a problem with the IBM MQ queues?](#)
- [Are some of your queues working?](#)
- [Are the correct queues defined?](#)
- [Does the problem affect only remote or cluster queues?](#)
- [Does the problem affect only shared queues?](#)
- [Is your application or IBM MQ for z/OS running slowly?](#)
- [Has your application or IBM MQ for z/OS stopped processing work?](#)

## 3. If you need further information to help you with identifying the cause of the problem, see [“详细故障诊断” on page 57.](#)

### Related tasks

[“联系 IBM 支持人员” on page 279](#)

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。

[“在 AIX 上执行初步检查” on page 7](#)

开始在 AIX 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。

[“在 IBM i 上执行初步检查” on page 17](#)

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。

[“在 Linux 上执行初步检查” on page 25](#)

开始在 Linux 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。

[“在 Windows 上执行初步检查” on page 36](#)

开始在 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。

### Related reference

[Messages and reason codes](#)

## Identifying characteristics of the problem on z/OS

Some initial questions to consider to help with identifying the cause of the problem.

### About this task

Use the following questions as pointers to help you to identify the cause of the problem:

- Has IBM MQ for z/OS run successfully before?
- Are there any error messages, return codes or other error conditions?
- Can you reproduce the problem?
- Have you applied any APARs or PTFs?
- Have any changes been made since the last successful run?
- Has the application run successfully before?
- Does the problem affect specific parts of the network?
- Does the problem occur at specific times of the day or affect specific users?
- Is the problem intermittent or does the problem occur with all z/OS, CICS, or IMS systems?
- Do you have a program error?

As you go through the list, make a note of anything that might be relevant to the problem. Even if your observations do not suggest a cause straight away, they might be useful later if you need to carry out a systematic problem determination exercise.

### Procedure

1. Has IBM MQ for z/OS run successfully before?

If the answer to this question is **No**, consider the following:

- Check your setup. If IBM MQ has not run successfully on z/OS before, it is likely that you have not yet set it up correctly. See the information about installing and customizing the queue manager in [Installing the IBM MQ for z/OS product](#) for further guidance.
- Verify the installation.
- Check that message `CSQ9022I` was issued in response to the **START QMGR** command (indicating normal completion).
- Ensure that z/OS displays IBM MQ as an installed subsystem. To determine if IBM MQ is an installed subsystem use the z/OS command `D OPDATA`.
- Check that the installation verification program (IVP) ran successfully.
- Use the command **DISPLAY DQM** to check that the channel initiator address space is running, and that the appropriate listeners are started.

2. Are there any error messages, return codes or other error conditions?

Investigate any error messages, return codes, and conditions where the queue manager or channel initiator terminated. The problem might produce the following types of error message or return codes:

- CSQ messages and reason codes

IBM MQ for z/OS error messages have the prefix CSQ. If you receive any messages with this prefix (for example, in the console log, or the CICS log), see [IBM MQ for z/OS messages, completion, and reason codes](#) for an explanation.

- Other messages

For messages with a different prefix, look in the appropriate messages and codes topic for a suggested course of action.

- Unusual messages

Be aware of unusual messages associated with the startup of IBM MQ for z/OS, or issued while the system was running before the error occurred. Any unusual messages might indicate some system problem that prevented your application from running successfully.

- Application MQI return codes

If your application gets a return code indicating that an MQI call has failed, see [Return codes](#) for a description of that return code.

### 3. Can you reproduce the problem?

If you can reproduce the problem, consider the conditions under which you can reproduce it. For example:

- Is it caused by a command? If so, is the command issued from the z/OS console, from CSQUTIL, from a program written to put commands onto the SYSTEM.COMMAND.INPUT queue, or by using the operations and control panels?
- Does the command work if it is entered by another method? If the command works when it is entered at the console, but not otherwise, check that the command server has not stopped, and that the queue definition of the SYSTEM.COMMAND.INPUT queue has not been changed.
- Is the command server running? Issue the command `DIS CMDSERV` to check.
- Is it caused by an application? If so, does it fail in CICS, IMS, TSO, or batch? Does it fail on all IBM MQ systems, or only on some?
- Is an application causing the problem? Can you identify any application that always seems to be running in the system when the problem occurs? If so, examine the application to see if it is in error.

### 4. Have you applied any APARs or PTFs?

APARs and PTFs can occasionally cause unexpected problems with IBM MQ. These fixes can have been applied to IBM MQ or to other z/OS systems.

If an APAR or PTF has been applied to IBM MQ for z/OS, check that no error message was produced. If the installation was successful, check with IBM Support for any APAR or PTF error.

If an APAR or PTF has been applied to any other product, consider the effect it might have on the way IBM MQ interfaces with it.

Ensure that you have followed any instructions in the APAR that affect your system. (For example, you might have to redefine a resource.)

### 5. Have any changes been made since the last successful run?

When you are considering changes that might recently have been made, think about IBM MQ, and also about the other programs it interfaces with, the hardware, and any new applications. Consider also the possibility that a new application that you do not yet know about might have been run on the system.

- Has your initialization procedure been changed? Consider whether that might be the cause of the problem. Have you changed any data sets, or changed a library definition? Has z/OS been initialized with different parameters? In addition, check for error messages sent to the console during initialization.
- Have you changed any queue definitions or security profiles? Consider whether some of your queues have been altered so that they are members of a cluster. This change might mean that messages arrive from different sources (for example, other queue managers or applications).
- Have you changed any definitions in your sysplex that relate to the support and implementation of shared queues? Consider the effect that changes to such definitions as your sysplex couple data set, or Coupling Facility resource management policy. These changes might have on the operation of shared queues. Also, consider the effect of changes to the Db2<sup>®</sup> data sharing environment.
- Has any of the software on your z/OS system been upgraded to a later release? Consider whether there are any necessary post-installation or migration activities that you need to perform.
- Has your z/OS subsystem name table been changed? Changes to levels of corequisite software like z/OS or LE might require additional changes to IBM MQ.

- Do your applications deal with return codes that they might get as a result of any changes you have made? Ensure that your applications deal with any new return codes that you introduce.
6. Has the application run successfully before?

If the problem appears to involve one particular application, consider whether the application has run successfully before.

- Have any changes been made to the application since it last ran successfully? If so, it is likely that the error lies somewhere in the new or modified part of the application. Investigate the changes and see if you can find an obvious reason for the problem.
- Have all the functions of the application been fully exercised before? Did problem occur when part of the application that had never been started before was used for the first time? If so, it is likely that the error lies in that part of the application. Try to find out what the application was doing when it failed, and check the source code in that part of the program for errors. If a program has been run successfully on many previous occasions, check the current queue status and files that were being processed when the error occurred. It is possible that they contain some unusual data value that causes a rarely used path in the program to be invoked.
- Does the application check all return codes? Has your system has been changed, perhaps in a minor way. Check the return codes your application receives as a result of the change. For example:
  - Does your application assume that the queues it accesses can be shared? If a queue has been redefined as exclusive, can your application deal with return codes indicating that it can no longer access that queue?
  - Have any security profiles been altered? An MQOPEN call might fail because of a security violation; can your application recover from the resulting return code?
- Does the application expect particular message formats? If a message with an unexpected message format has been put onto a queue (for example, a message from a queue manager on a different platform), it might require data conversion or another different form of processing.
- Does the application run on other IBM MQ for z/OS systems? Is something different about the way that this queue manager is set up that is causing the problem? For example, have the queues been defined with the same maximum message length, or default priority?
- Does the application use the MQSET call to change queue attributes? Is the application is designed to set a queue to have no trigger, then process some work, then set the queue to have a trigger? The application might have failed before the queue had been reset to have a trigger.
- Does the application handle messages that cause an application to fail? If an application fails because of a corrupted message, the message retrieved is rolled back. The next application might get the same message and fail in the same way. Ensure that applications use the backout count; when the backout count threshold has been reached, the message in question is put onto the backout queue.

If your application has never run successfully before, examine your application carefully to see if you can find any of the following errors:

- Translation and compilation problems

Before you look at the code, examine the output from the translator, the compiler or assembler, and the linkage editor, to see if any errors have been reported. If your application fails to translate, compile/assemble, or link edit into the load library, it also fails to run if you attempt to invoke it. See [Developing applications](#) for information about building your application, and for examples of the job control language (JCL) statements required.

- Batch and TSO programs

For batch and TSO programs, check that the correct stub has been included. There is one batch stub and two RRS stubs. If you are using RRS, check that you are not using the MQCMIT and MQBACK calls with the CSQBRSTB stub. Use the CSQBRRSI stub if you want to continue using these calls with RRS.

- CICS programs

For CICS programs, check that the program, the IBM MQ CICS stub, and the CICS stub have been linked in the correct order. Also, check that your program or transaction is defined to CICS.

- IMS programs

For IMS programs, check that the link includes the program, the IBM MQ stub, and the IMS language interface module. Ensure that the correct entry point has been specified. A program that is loaded dynamically from an IMS program must have the stub and language interface module linked also if it is to use IBM MQ.

- Possible code problems

If the documentation shows that each step was accomplished without error, consider the coding of the application. Do the symptoms of the problem indicate the function that is failing and, therefore, the piece of code in error? See Step “10” on [page 49](#) for some examples of common errors that cause problems with IBM MQ applications.

- Do applications report errors from IBM MQ?

For example, a queue might not be enabled for "gets". It receives a return code specifying this condition but does not report it. Consider where your applications report any errors or problems.

## 7. Does the problem affect specific parts of the network?

You might be able to identify specific parts of the network that are affected by the problem (for example, remote queues). If the link to a remote queue manager is not working, the messages cannot flow to a target queue on the target queue manager.

- Check that the connection between the two systems is available, and that the channel initiator and listener have been started. Use the MQSC **PING CHANNEL** command to check the connection.
- Check that messages are reaching the transmission queue, and check the local queue definition of the transmission queue, and any remote queues. Use the MQSC **BYTSENT** keyword of the **DISPLAY CHSTATUS** command to check that data is flowing along the channel. Use **DISPLAY QLOCAL (XMITQ) CURDEPTH** to check whether there are messages to be sent on the transmission queue. Check for diagnostic messages at both ends of the channel informing you that messages have been sent to the dead-letter queue.
- If you are using IBM MQ clusters, check that the clustering definitions have been set up correctly.

Have you made any network-related changes that might account for the problem? Have you changed any IBM MQ definitions, or any CICS or IMS definitions? Check the triggering attributes of the transmission queue.

## 8. Does the problem occur at specific times of the day or affect specific users?

If the problem occurs at specific times of day, it might be that it is dependent on system loading. Typically, peak system loading is at mid-morning and mid-afternoon, and so these periods are the times when load-dependent problems are most likely to occur. (If your network extends across more than one time zone, peak system loading might seem to occur at some other time of day.) If you think that your IBM MQ for z/OS system has a performance problem, see [“Dealing with performance problems on z/OS” on page 266](#).

If the problem only affects some users, is it because some users do not have the correct security authorization? See [User IDs for security checking](#) for information about user IDs checked by IBM MQ for z/OS.

## 9. Is the problem intermittent or does the problem occur with all z/OS, CICS, or IMS systems?

A problem might be caused by application interaction or be related to other z/OS systems.

An intermittent problem could be caused by failing to take into account the fact that processes can run independently of each other. For example, a program might issue an MQGET call, without specifying WAIT, before an earlier process has completed. You might also encounter this type of problem if your application tries to get a message from a queue while it is in sync point (that is, before it has been committed).



If the problem only occurs when you access a particular z/OS, IMS, or CICS system, consider what is different about this system. Also consider whether any changes have been made to the system that might affect the way it interacts with IBM MQ.

#### 10. Do you have a program error?

The following examples show the most common causes of problems encountered while running IBM MQ programs. Consider the possibility that the problem with your system could be caused by one of these errors.

- Programs issue MQSET to change queue attributes and fail to reset attributes of a queue. For example, setting a queue to NOTRIGGER.
- Making incorrect assumptions about the attributes of a queue. This assumption could include assuming that queues can be opened with MQOPEN when they are MQOPEN-exclusive, and assuming that queues are not part of a cluster when they are.
- Trying to access queues and data without the correct security authorization.
- Linking a program with no stub, or with the wrong stub (for example, a TSO program with the CICS stub). This can cause either a long-running unit of work, or an X'0C4' or other abend.
- Passing incorrect or invalid parameters in an MQI call; if the wrong number of parameters are passed, no attempt can be made to complete the completion code and reason code fields, and the task is abended. (This is an X'0C4' abend.) This problem might occur if you attempt to run an application on an earlier version of MQSeries® than it was written for, where some of the MQI values are invalid.
- Failing to define the IBM MQ modules to z/OS correctly (this error causes an X'0C4' abend in CSQYASCP).
- Failing to check return codes from MQI requests. This problem might occur if you attempt to run an application on a later version of IBM MQ than it was written for, where new return codes have been introduced that are not checked for.
- Failing to open objects with the correct options needed for later MQI calls, for example using the MQOPEN call to open a queue but not specifying the correct options to enable the queue for subsequent MQGET calls.
- Failing to initialize *MsgId* and *CorrelId* correctly. This error is especially true for MQGET.
- Using incorrect addresses.
- Using storage before it has been initialized.
- Passing variables with incorrect lengths specified.
- Passing parameters in the wrong order.
- Failing to define the correct security profiles and classes to RACF®. This might stop the queue manager or prevent you from carrying out any productive work.
- Relying on default MQI options for a ported application. For example, z/OS defaults to MQGET and MQPUT in sync point. The distributed-platform default is out of sync point.
- Relying on default behavior at a normal or abnormal end of a portal application. On z/OS, a normal end does an implicit MQCMIT and an abnormal end does an implicit rollback.

#### Related tasks

“联系 IBM 支持人员” on page 279

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。

“Examining the problem in greater depth on z/OS” on page 50

Further checks to carry out when you have established that no changes have been made to your system, and that there are no problems with your application programs, but the preliminary checks have not enabled you to solve your problem.

#### Related reference

[Messages and reason codes](#)

## Examining the problem in greater depth on z/OS

Further checks to carry out when you have established that no changes have been made to your system, and that there are no problems with your application programs, but the preliminary checks have not enabled you to solve your problem.

### About this task

Use the following questions as pointers to help you to identify the cause of the problem:

- [Have you received some incorrect output?](#)
- [Have you received an unexpected error message or return code?](#)
- [Has there been an abend?](#)
- [Have you received no response from an MQSC command?](#)
- [Is there a problem with the IBM MQ queues?](#)
- [Are some of your queues working?](#)
- [Are the correct queues defined?](#)
- [Does the problem affect only remote or cluster queues?](#)
- [Does the problem affect only shared queues?](#)
- [Is your application or IBM MQ for z/OS running slowly?](#)
- [Has your application or IBM MQ for z/OS stopped processing work?](#)

### Procedure

#### 1. Have you received some incorrect output?

If you have obtained what you believe to be some incorrect output, consider the following:

- When to classify output as incorrect

"Incorrect output" might be regarded as any output that you were not expecting. However, use this term with care in the context of problem determination because it might be a secondary effect of some other type of error. For example, looping could be occurring if you get any repetitive output, even though that output is what you expected.

- Error messages

IBM MQ also responds to many errors it detects by sending error messages. You might regard these messages as "incorrect output", but they are only symptoms of another type of problem. If you have received an error message from IBM MQ that you were not expecting, see [Are there any error messages, return codes or other error conditions?](#) in ["Identifying characteristics of the problem on z/OS"](#) on page 45.

- Unexpected messages

Your application might not have received a message that it was expecting, or has received a message containing unexpected or corrupted information, or has received a message that it was not expecting (for example, one that was destined for a different application). For more information, see ["Dealing with incorrect output on z/OS"](#) on page 272.

#### 2. Have you received an unexpected error message or return code?

If your application has received an unexpected error message, consider whether the error message has originated from IBM MQ or from another program.

- IBM MQ error messages

IBM MQ for z/OS error messages are prefixed with the letters CSQ. If you get an unexpected IBM MQ error message (for example, in the console log, or the CICS log), see [IBM MQ for z/OS messages, completion, and reason codes](#) for an explanation, which might give you enough

information to resolve the problem quickly, or it might redirect you to further information. If you cannot deal with the message, you might have to contact the IBM Support for help.

- Non- IBM MQ error messages

If you get an error message from another IBM program, or from the operating system, look in the appropriate messages and codes documentation for an explanation of what it means. In a queue-sharing environment, look for the following error messages:

- XES (prefixed with the letters IXL)
- Db2 (prefixed with the letters DSN)
- RRS (prefixed with the letters ATR)

- Unexpected return codes

If your application has received an unexpected return code from IBM MQ, see [Return codes](#) for information about how your application can handle IBM MQ return codes.

### 3. Has there been an abend?

If your application has stopped running, this might be caused by an abnormal termination (abend). Abends can be caused by the user ending the task being performed before it terminates normally; for example, if you purge a CICS transaction. Abends can also be caused by an error in an application program.

You are notified of an abend in one of the following places, depending on what type of application you are using:

- For Batch applications, your listing shows the abend.
- For CICS applications, you see a CICS transaction abend message. If your task is a terminal task, this message is displayed on your screen. If your task is not attached to a terminal, the message is displayed on the CICS CSMT log.
- For IMS applications, in all cases, you see a message at the IBM MQ for IMS master terminal and in the listing of the dependent region involved. If an IMS transaction that had been entered from a terminal was being processed, an error message is also sent to that terminal.
- For TSO applications, you might see a TSO message with a return code on your screen. (Whether this message is displayed depends on the way your system is set up, and the type of error.)

For some abends, an address space dump is produced. For CICS transactions, a transaction dump showing the storage areas of interest to the transaction is provided.

- If an application passes some data, the address of which is no longer valid, a dump is sometimes produced in the address space of the user.

**Note:** For a batch dump, the dump is formatted and written to SYSUDUMP. For information about SYSUDUMPs, see [“SYSUDUMP information on z/OS”](#) on page 263. For CICS, a system dump is written to the SYS1.DUMP data sets, as well as a transaction dump being taken.

- If a problem with IBM MQ for z/OS itself causes an abend, an abend code of X'5C6' or X'6C6' is returned, along with an abend reason code. This reason code uniquely describes the cause of the problem. See [“IBM MQ for z/OS abends”](#) on page 228 for information about the abend codes, and see [Return codes](#) for an explanation of the reason code.

If your program has terminated abnormally, see [“Dealing with abends on IBM MQ for z/OS”](#) on page 229.

If your system has terminated abnormally, and you want to analyze the dump produced, see [“IBM MQ for z/OS dumps”](#) on page 246. This section tells you how to format the dump, and how to interpret the data contained in it.

### 4. Have you received no response from an MQSC command?

If you have issued an MQSC command from an application, and not from a z/OS console, but you have not received a response, consider the following questions:

- Is the command server running?

Check that the command server is running, as follows:

- a. Use the **DISPLAY CMDSERV** command at the z/OS console to display the status of the command server.
  - b. If the command server is not running, start it using the **START CMDSERV** command.
  - c. If the command server is running, use the **DISPLAY QUEUE** command with the name of the system-command input queue and the **CURDEPTH** and **MAXDEPTH** attributes to define the data displayed. If these values show that the queue is full, and the command server has been started, the messages are not being read from the queue.
  - d. Try stopping the command server and then restarting it, responding to any error messages that are produced.
  - e. Issue the display command again to see if it is working now.
- Has a reply been sent to the dead-letter queue?

If you do not know the name of the system dead-letter queue, use the **DISPLAY QMGR DEADQ** command to find the name. Use this name in the **DISPLAY QUEUE** command with the **CURDEPTH** attribute to see if there are any messages on the queue. The dead-letter queue message header (dead-letter header structure) contains a reason or feedback code describing the problem. For information about the dead-letter header structure, see [Reason \(MQLONG\)](#).

- Are the queues enabled for PUTs and GETs?

Use the **DISPLAY QUEUE** command from the console to check, for example `DISPLAY QUEUE (SYSTEM.COMMAND.INPUT) PUT GET`.

- Is the **WaitInterval** parameter set to a sufficiently long time?

If your MQGET call has timed out, your application receives completion code of 2 and a reason code of 2033 (MQRC\_NO\_MSG\_AVAILABLE). (See [WaitInterval \(MQLONG\)](#) and [MQGET - Get message for information about the WaitInterval parameter, and completion and reason codes from MQGET.](#))

- Is a sync point required?

If you are using your own application program to put commands onto the system-command input queue, consider whether you must take a sync point. You must take a sync point after putting messages to a queue, and before attempting to receive reply messages, or use `MQPMO_NO_SYNCPOINT` when putting them. Unless you have excluded your request message from sync point, you must take a sync point before attempting to receive reply messages.

- Are the **MaxDepth** and **MaxMsgL** parameters of your queues set sufficiently high?

See [CSQO016E](#) for information about defining the system-command input queue and the reply-to queue.

- Are you using the **CorrelId** and **MsgId** parameters correctly?

You must identify the queue and then display the **CURDEPTH**. Use the **DISPLAY QUEUE** command from the console (for example, `DISPLAY QUEUE (MY.REPLY.QUEUE) CURDEPTH`), to see if there are messages on the reply-to queue that you have not received. Set the values of *MsgId* and *CorrelId* in your application to ensure that you receive all messages from the queue.

The following questions are applicable if you have issued an MQSC command from either a z/OS console (or its equivalent), or an application, but have not received a response:

- Is the queue manager still running, or did your command cause an abend?

Look for error messages indicating an abend, and if one occurred, see [“IBM MQ for z/OS dumps” on page 246](#).

- Were any error messages issued?

Check to see if any error messages were issued that might indicate the nature of the error.

For information about the different methods you can use to enter MQSC commands, see [Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS](#).

5. Is there a problem with the IBM MQ queues?

If you suspect that there is a problem affecting the queues on your subsystem, use the operations and control panels to display the system-command input queue.

- Has the system responded? If the system responds, then at least one queue is working. In this case, continue with Step “6” on page 53.
- Has the system not responded? The problem might be with the whole subsystem. In this instance, try stopping and restarting the queue manager, responding to any error messages that are produced. Check for any messages on the console needing action. Resolve any that might affect IBM MQ, such as a request to mount a tape for an archive log. See if other subsystems or CICS regions are affected. Use the **DISPLAY QMGR COMMANDQ** command to identify the name of the system command input queue.
- Does the problem still occur after restart? Contact IBM Support for help (see “联系 IBM 支持人员” on page 279).

#### 6. Are some of your queues working?

If you suspect that the problem occurs with only a subset of queues, select the name of a local queue that you think is having problems and use the **DISPLAY QUEUE** and **DISPLAY QSTATUS** commands to display information about the queue.

- Is the queue being processed?
  - If **CURDEPTH** is at **MAXDEPTH**, it might indicate that the queue is not being processed. Check that all applications that use the queue are running normally (for example, check that transactions in your CICS system are running or that applications started in response to Queue Depth High events are running).
  - Use the command **DISPLAY QSTATUS(xx) IPPROCS** to see if the queue is open for input. If not, start the application.
  - If **CURDEPTH** is not at **MAXDEPTH**, check the following queue attributes to ensure that they are correct:
    - If triggering is being used, is the trigger monitor running? Is the trigger depth too big? Is the process name correct? Have all the trigger conditions been met?  
Use the command **DISPLAY QSTATUS(xx) IPPROCS** to see if an application has the same queue open for input. In some triggering scenarios, a trigger message is not produced if the queue is open for input. Stop the application to cause the triggering processing to be invoked.
    - Can the queue be shared? If not, another application (batch, IMS, or CICS) might already have it open for input.
    - Is the queue enabled appropriately for GET and PUT?
- Do you have a long-running unit of work?

If **CURDEPTH** is not zero, but when you attempt to **MQGET** a message the queue manager replies that there is no message available, either use the command **DIS QSTATUS(xx) TYPE(HANDLE)** to show you information about applications that have the queue open, or use the command **DIS CONN(xx)** to give you more information about an application that is connected to the queue.

- How many tasks are accessing the queues?

Use the command **DISPLAY QSTATUS(xx) OPROCS IPPROCS** to see how many tasks are putting messages on to, and getting messages from the queue. In a queue-sharing environment, check **OPPROCS** and **IPPROCS** on each queue manager. Alternatively, use the **CMDSCOPE** attribute to check all the queue managers. If there are no application processes getting messages from the queue, determine the reason, which might, for example, be because the applications need to be started, or a connection has been disrupted, or because the **MQOPEN** call has failed for some reason.

- Is this queue a shared queue? Does the problem affect only shared queues?

Check that there is not a problem with the sysplex elements that support shared queues. For example, check that there is not a problem with the IBM MQ-managed Coupling Facility list structure.



Use the command `D XCF, STRUCTURE, STRNAME=ALL` to check that the Coupling Facility structures are accessible.

Use the command `D RRS` to check that RRS is active.

- Is this queue part of a cluster?

Check to see if the queue is part of a cluster (from the **CLUSTER** or **CLUSNL** attribute). If it is, verify that the queue manager that hosts the queue is still active in the cluster.

If you cannot solve the problem, contact IBM Support for help (see [“联系 IBM 支持人员” on page 279](#)).

#### 7. Are the correct queues defined?

IBM MQ requires certain predefined queues. Problems can occur if these queues are not defined correctly.

- Check that the system-command input queue, the system-command reply model queue, and the reply-to queue are correctly defined, and that the `MQOPEN` calls were successful.
- If you are using the system-command reply model queue, check that it was defined correctly.
- If you are using clusters, you need to define the `SYSTEM.CLUSTER.COMMAND.QUEUE` to use commands relating to cluster processing.

#### 8. Does the problem affect only remote or cluster queues?

If the problem affects only remote or cluster queues, check:

- Are the remote queues being accessed? Check that the programs putting messages to the remote queues have run successfully (see [“Dealing with incorrect output on z/OS” on page 272](#)).
- Is the system link active? Use `APPC` or `TCP/IP` commands as appropriate to check whether the link between the two systems is active. Use **PING** or **OPING** for `TCP/IP` or `D NET ID=xxxxx, E` for `APPC`.
- Is triggering working? If you use triggering to start the distributed queuing process, check that the transmission queue has triggering set on and that the queue is get-enabled.
- Is the channel or listener running? If necessary, start the channel or the listener manually, or try stopping and restarting the channel. See [Configuring distributed queuing](#) for more information. Look for error messages on the startup of the channel initiator and listener. See [IBM MQ for z/OS messages, completion, and reason codes](#) and [Configuring distributed queuing](#) to determine the cause.
- What is the channel status? Check the channel status using the **DISPLAY CHSTATUS (channel\_name)** command.
- Are your process and channel definitions correct? Check your process definitions and your channel definitions.

For information about how to use distributed queuing, and for information about how to define channels, see [Configuring distributed queuing](#).

#### 9. Does the problem affect only shared queues?

If the problem affects only queue sharing groups, use the **VERIFY QSG** function of the `CSQ5PQSG` utility. This command verifies that the Db2 setup is consistent in terms of the bitmap allocation fields, and object definition for the Db2 queue manager, structure, and shared queue objects, and reports details of any inconsistency that is discovered.

The following is an example of a **VERIFY QSG** report with errors:

```
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

```

#### 10. Is your application or IBM MQ for z/OS running slowly?

Slow applications can be caused by the application itself or underlying software including IBM MQ.

If your application is running slowly, this could indicate that it is in a loop, or waiting for a resource that is not available.

- Is the problem worse at peak system load times? This could also be caused by a performance problem. Perhaps it is because your system needs tuning, or because it is operating near the limits of its capacity. This type of problem is probably worst at peak system load times, typically at mid-morning and mid-afternoon. If your network extends across more than one time zone, peak system load might seem to you to occur at some other time.
- Does the problem occur when the system is lightly loaded? If you find that degrading performance is not dependent on system loading, but happens sometimes when the system is lightly loaded, a poorly designed application program is probably to blame. This could manifest itself as a problem that only occurs when specific queues are accessed.
- Is IBM MQ for z/OS running slowly? The following symptoms might indicate that IBM MQ for z/OS is running slowly:
  - If your system is slow to respond to commands.
  - If repeated displays of the queue depth indicate that the queue is being processed slowly for an application with which you would expect a large amount of queue activity.

For guidance on dealing with waits and loops, see [“Dealing with applications that are running slowly or have stopped on z/OS” on page 267](#), and on dealing with performance problems, see [“Dealing with performance problems on z/OS” on page 266](#).

#### 11. Has your application or IBM MQ for z/OS stopped processing work?

There are several reasons why your system might unexpectedly stop processing work. The problem areas to check for include:

- Are there any queue manager problems? The queue manager might be shutting down.
- Are there any application problems? An application programming error might mean that the program branches away from its normal processing, or the application might get in a loop. There might also have been an application abend.
- Are there any problems with IBM MQ? Your queues might have become disabled for MQPUT or MQGET calls, the dead-letter queue might be full, or IBM MQ for z/OS might be in a wait state, or a loop.
- Are there any z/OS or other system problems? z/OS might be in a wait state, or CICS or IMS might be in a wait state or a loop. There might be problems at the system or sysplex level that are affecting the queue manager or the channel initiator. For example, excessive paging. It might also indicate DASD problems, or higher priority tasks with high processor usage.
- Are there any Db2 or RRS problems? Check that Db2 and RRS are active.

In all cases, carry out the following checks to determine the cause of the problem:

##### a) Check for error messages.

Use the **DISPLAY THREAD(\*)** command to check if the queue manager is running. If the queue manager has stopped running, look for any messages that might explain the situation. Messages are displayed on the z/OS console, or on your terminal if you are using the operations and control

panels. Use the **DISPLAY DQM** command to see if the channel initiator is working, and the listeners are active. The z/OS command

```
DISPLAY R,L
```

lists messages with outstanding replies. Check to see whether any of these replies are relevant. In some circumstances, for example, when it has used all its active logs, IBM MQ for z/OS waits for operator intervention.

b) If no there are no error messages, issue the following z/OS commands:

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

where *xxxx* is the IBM MQ for z/OS subsystem name.

If you receive a message telling you that the queue manager or channel initiator has not been found, this message indicates that the subsystem has terminated. This condition could be caused by an abend or by operator shutdown of the system.

If the subsystem is running, you receive message IEE105I. This message includes the *CT=nnnn* field, which contains information about the processor time being used by the subsystem. Note the value of this field, and reissue the command.

- If the *CT=* value has not changed, this indicates that the subsystem is not using any processor time. This could indicate that the subsystem is in a wait state (or that it has no work to do). If you can issue a command like **DISPLAY DQM** and you get output back, this indicates there is no work to do rather than a hang condition.
- If the *CT=* value has changed dramatically, and continues to do so over repeated displays, this could indicate that the subsystem is busy or possibly in a loop.
- If the reply indicates that the subsystem is now not found, this indicates that it was in the process of terminating when the first command was issued. If a dump is being taken, the subsystem might take a while to terminate. A message is produced at the console before terminating. To check that the channel initiator is working, issue the **DISPLAY DQM** command. If the response does not show the channel initiator working this could be because it is getting insufficient resources (like the processor). In this case, use the z/OS monitoring tools, such as RMF, to determine if there is a resource problem. If it is not, restart the channel initiator.

c) Check whether the queue manager or channel initiator terminated has abnormally.

Look for any messages saying that the queue manager or channel initiator address space has abnormally terminated. If you get a message for which the system action is to terminate IBM MQ, find out whether a system dump was produced. For more information, see [IBM MQ dumps](#).

d) Check whether IBM MQ for z/OS might still be running.

Consider also that IBM MQ for z/OS might still be running, but only slowly. If it is running slowly, you probably have a performance problem. To confirm this, see Step “10” on page 55. For advice about what to do next, see [Dealing with performance problems](#).

### Related tasks

[“联系 IBM 支持人员” on page 279](#)

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。

[“Identifying characteristics of the problem on z/OS” on page 45](#)

Some initial questions to consider to help with identifying the cause of the problem.

### Related reference

[Messages and reason codes](#)

## 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 367 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 376 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 279 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 387 页的『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。

Windows

Linux

AIX

## 对 AMQP 问题进行故障诊断

用于帮助您解决运行 AMQP 应用程序的问题的故障诊断信息。

### 相关任务

第 422 页的『跟踪高级消息排队协议 (AMQP) 服务』

高级消息排队协议 (AMQP) 服务提供的跟踪工具用于帮助 IBM 支持人员诊断与服务相关的客户问题。

Windows

Linux

AIX

## 重新启动 AMQP 的 IBM MQ 服务

V 9.4.0

V 9.4.0

从 IBM MQ 9.4.0 开始，缺省情况下，如果已安装 AMQP 文件集，那么当创建新的队列管理器时，它将具有以下命令中显示的 SERVICE 定义: **runmqsc**。

## 关于此任务

V 9.4.0

V 9.4.0

```
display service(*) all
AMQ8629I: Display service information details.
SERVICE(SYSTEM.AMQP.SERVICE)          CONTROL(MANUAL)
SERVTYPE(SERVER)
STARTCMD(+MQ_INSTALL_PATH+/bin/amqp.sh)
STARTARG(start -m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+/. " -g "+MQ_DATA_PATH+/.")
STOPCMD(+MQ_INSTALL_PATH+/bin/endmqsd)
STOPARG(-m +QMNAME+ -n "+MQ_SERVICE_NAME+" -t AMQP)
STDOUT(+MQ_Q_MGR_DATA_PATH+/amqp.stdout)
STDERR(+MQ_Q_MGR_DATA_PATH+/amqp.stderr)
DESCR(Manages clients that use the AMQP protocol)
```

V 9.4.0

V 9.4.0

请注意，此服务具有以下属性:

CONTROL(MANUAL)

V 9.4.0

V 9.4.0

这意味着服务不会作为队列管理器启动的一部分启动。如果您希望服务在队列管理器启动时启动，请将 **CONTROL** 属性更改为 **QMGR**。

V 9.4.0

V 9.4.0

从 IBM MQ 9.1.0 到 IBM MQ 9.3.0，即 SYSTEM.AMQP.SERVICE 将其 **CONTROL** 属性设置为 **QMGR**。

要显示服务，请使用 **DISPLAY SVSTATUS** 命令:

```
display svstatus(*)
1 : display svstatus(*)
```

```
AMQ8632I: Display service status details.
SERVICE(SYSTEM.AMQP.SERVICE)          STATUS(RUNNING)
PID(18510)
```

然后发出以下命令以列出服务状态:

```
display svstatus(SYSTEM.AMQP.SERVICE) all
1 : display svstatus(*) all
AMQ8632I: Display service status details.
SERVICE(SYSTEM.AMQP.SERVICE)          STATUS(RUNNING)
PID(26263)                               SERVTYPE(SERVER)
STARTDA(2023-08-22)                     STARTTI(06.25.31)
CONTROL(QMGR)                           STARTCMD(/opt/mqm93//bin/amqp.sh)
STARTARG(start -m AMQP -d "/var/mqm/qmgrs/AMQP/." -g "/var/mqm/." )
STOPCMD(/opt/mqm93//bin/endmqsd)
STOPARG(-m AMQP -n "SYSTEM.AMQP.SERVICE" -t AMQP)
DESCR(Manages clients that use the AMQP protocol)
STDOUT(/var/mqm/qmgrs/AMQP//amqp.stdout)
STDERR(/var/mqm/qmgrs/AMQP//amqp.stderr)
```

## 过程

1. 在 Linux 上, 使用以下命令来标识进程标识 (其中 QMGRNAME 是实际队列管理器的名称):

```
ps -ef | grep QMGRNAME
```

请注意 com.ibm.mq.MQXRService.RunMQXRService 的条目。

2. 在 **runmqsc** 中使用以下命令来停止服务:

```
stop SERVICE(SYSTEM.AMQP.SERVICE)
AMQ8732I: Request to stop Service accepted.
```

3. 等待几秒钟让实际程序结束, 然后重复:

```
ps -ef | grep QMGRNAME
```

请注意, 此时找不到 RunMQXRService 的条目。

4. 在 **runmqsc** 中使用以下命令来启动服务:

```
start SERVICE(SYSTEM.AMQP.SERVICE)
AMQ8733I: Request to start Service accepted.
```

5. 再次发出 `ps -ef | grep QMGRNAME`。

请检查 RunMQXRService 的条目是否再次存在, 这指示 AMQP 服务已重新启动。

**注:** amqp 服务作为 Java 进程运行, 并且如果尝试停止该服务似乎未完成, 那么该进程将保持处于停止状态, 如以下示例中所示:

```
dis svstatus (SYSTEM.AMQP.SERVICE)
27 : dis svstatus (SYSTEM.AMQP.SERVICE)
AMQ8632I: Display service status details.
SERVICE(SYSTEM.AMQP.SERVICE)          STATUS(STOPPING)
```

必须手动结束 Java 进程。

Windows V 9.4.0 V 9.4.0 Linux AIX 为 AMQP 启用 JSON 格式化

## 日志

要启用 JSON 格式错误日志记录, 需要修改 AMQP 配置文件 `amqptraceOn.properties` 和 `amqptraceOff.properties`。

## 关于此任务

您可以将 JSON 格式日志记录配置为与基于文本的错误日志记录一起独占或同时使用。

**注:** 修改配置文件后, 需要重新启动 AMQP 服务以使任何更改生效。



## 过程

1. 要以独占方式启用 JSON 格式日志记录，请修改 `amqptraceOn.properties` 和 `amqptraceOff.properties` 配置文件，并按如下所示更新 **handlers** 属性：

```
handlers= com.ibm.mq.util.logging.MQJSONErrorLogFileHandler
```

2. 要同时启用 JSON 格式日志记录以及基于文本的错误日志记录，请修改 `amqptraceOn.properties` 和 `amqptraceOff.properties` 配置文件并更新 **handlers** 属性，如下所示：

```
handlers= com.ibm.mq.util.logging.MQErrorLogFileHandler,  
com.ibm.mq.util.logging.MQJSONErrorLogFileHandler
```

## 结果

在任一情况下，这都会启用 JSON 格式错误日志记录，并使用以下缺省属性配置记录器：

- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.level`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.filter`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.limit`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.count`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.append`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.permissions`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.formatter`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.pattern`

在功能上，这些属性与 `com.ibm.mq.util.logging.MQErrorLogFileHandler` 配置的基于文本的错误记录器工作方式相同。

## 相关参考

[AMQP 通道日志文件](#)

## Windows Linux AIX AMQP 日志、错误日志和配置文件的位置

查找 AMQP 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

高级消息排队协议 (AMQP) 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 `AMQPn.FDC`。

它还会编写 AMQP 服务日志。日志路径为：

```
WMQ data directory\Qmgrs\qMgrName\errors\amqp.log
```

该日志文件的格式为 `amqp_n.log`。

  (可选) 如果启用了 AMQP 的 JSON 格式日志记录，那么将编写 JSON 日志文件。日志路径为：

```
WMQ data directory\Qmgrs\qMgrName\errors\amqp.json
```

**V 9.4.0** **V 9.4.0** 该日志文件的格式为 amqp\_n.json。

**V 9.4.0** **V 9.4.0** AMQP 服务启动时，会将输出写入：

```
WMQ data directory\Qmgrs\qMgrName\amqp.stdout  
WMQ data directory\Qmgrs\qMgrName\amqp.stdeir
```

## 服务器端的配置文件

### 高级消息排队协议 (AMQP) 服务

从 IBM MQ 9.3.0 起，JAAS.Login 模块文件已从 &MQ\_INSTALL\_DIRECTORY%/mqxr/samples/samples 目录移动到 &MQ\_INSTALL\_DIRECTORY%/mqxr/samples/jaas 目录

AMQP 配置文件名在 Windows 系统上为 amqp\_win.properties 文件，在 AIX 或 Linux 系统上则为 amqp\_unix.properties 文件。属性文件保存在 AMQP 配置目录中：

```
WMQ data directory\Qmgrs\qMgrName\amqp
```

图 1: Windows 上的 AMQP 配置目录

```
/var/mqm/qmgrs/qMgrName/amqp
```

图 2: AIX 或 Linux 上的 AMQP 配置目录

### JVM

将作为自变量传递的 Java 属性设置为 java.properties 文件中的 AMQP 服务。该文件中的属性会直接传递到运行 AMQP 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 AMQP 配置相同的文件夹中查找 java.properties 文件。请参阅 [第 60 页的图 1](#) 和 [第 60 页的图 2](#)。

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

为 AMQP 通道配置 [JAAS](#) 中描述了 JAAS 配置文件，其中包括 IBM MQ AMQP 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 AMQP 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 AMQP 服务定义中已设置的类路径。

通过将 echo set classpath 添加到 runAMQPService.bat 来显示 AMQP 服务正在使用的类路径。输出将发送到 amqp.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\amqp\samples\samples
```

注: `service.env` 不得包含任何变量。请替换 *WMQ Installation Directory* 的实际值。

图 3: *service.env for Windows* 样本

## 跟踪

请参阅 [第 422 页的『跟踪高级消息排队协议 \(AMQP\) 服务』](#)。用于配置跟踪的参数存储在两个文件中:

```
WMQ data directory\Qmgrs\qMgrName\amqp\trace.config  
WMQ data directory\Qmgrs\qMgrName\amqp\amqptraceOn.properties
```

并且存在对应的文件:

```
WMQ data directory\Qmgrs\qMgrName\amqp\amqptraceOff.properties
```

## 客户端日志文件和客户端配置文件

请参阅 [开发 AMQP 客户机应用程序](#)，以获取有关客户端应用程序的信息。

## 对 AMS 问题进行故障诊断

用于帮助您识别和解决与 Advanced Message Security (AMS) 相关的问题的故障诊断信息。

### 关于此任务

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。要获取更多错误日志的信息，请参阅 [第 369 页的『AIX, Linux, and Windows 上的错误日志』](#)。

### 相关概念

[第 416 页的『Enabling internal trace for the AMSM address space』](#)

Trace for the AMSM address space can be enabled using the `_AMS_MSG_LEVEL` variable, which is passed into the AMSM address space through the ENVARS DD card.

### 相关任务

[第 292 页的『收集 AMS 问题的信息』](#)

如果需要 IBM 支持人员的帮助以解决多平台上的 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。

[第 328 页的『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。

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

错误 `com.ibm.security.pkcsutil.PKCSEException: 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.PKCSEException: 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。

如果您要使用的签名算法未在当前安全策略中指定，请针对您的产品版本从以下位置下载正确的 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 \(MCA\) 拦截和 AMS](#)。

**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

当您尝试运行命令时，可能会发生一些常见问题。

### 关于此任务

某些字符，例如反斜杠 (\) 和双引号 (") 与命令配合使用时，字符具有特殊含义。如果在命令的描述性文本中使用特殊字符时接收到错误，请检查是否正确输入了特殊字符。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](#)。

## 解决 MQSC 命令的问题

如果您无法运行 MQSC 命令，请使用本主题中的信息，了解这些常见问题中的任何一个问题是否适用于您。当您阅读命令生成的错误时，产生的问题并不总是很明显。

### 过程

- 如果要从文件重定向输入，请确保使用 < 重定向运算符。  
如果省略此运算符，那么队列管理器会将文件名解释为队列管理器名称，并发出以下错误消息：

```
AMQ8118E: IBM MQ queue manager does not exist.
```

- 如果要将输出重定向到文件，请确保使用 > 重定向运算符。  
缺省情况下，在调用 **runmqsc** 时，将文件放入当前工作目录中。指定标准文件名，以将输出发送至特定文件和目录。
- 通过使用以下命令来显示所有队列管理器，检查是否已创建将运行命令的队列管理器以及队列管理器是否正在运行：

```
dspmq
```

如果队列管理器未在运行，请启动该管理器。有关更多信息，请参阅[启动队列管理器](#)。

- 确保您在 **runmqsc** 命令上指定了队列管理器，或者定义了缺省队列管理器。  
如果尚未定义缺省队列管理器，或者您收到以下错误：

```
AMQ8146E: IBM MQ queue manager not available.
```

- 请确认未尝试将 MQSC 命令指定为 **runmqsc** 命令的参数。不能将 MQSC 命令指定为 **runmqsc** 命令的参数。例如，以下内容无效：

```
runmqsc DEFINE QLOCAL(FRED)
```

- 检查是否已发出 **runmqsc** 命令。不能在发出 **runmqsc** 命令之前输入 MQSC 命令。
- 请确认未尝试从 **runmqsc** 运行控制命令。无法从 **runmqsc** 运行控制命令。例如，当您以交互方式运行 MQSC 命令时，不能发出 **strmqm** 命令来启动队列管理器。如果执行此操作，那么将接收到类似以下消息的错误消息：

```
runmqsc
:
:
Starting MQSC for queue manager jupiter.queue.manager.
1 : strmqm saturn.queue.manager
AMQ8405: Syntax error detected at or near end of cmd segment below:-s
AMQ8426: Valid MQSC commands are:
ALTER
CLEAR
DEFINE
DELETE
DISPLAY
END
PING
REFRESH
RESET
RESOLVE
RESUME
START
STOP
SUSPEND
2 : end
```

### 相关任务

[使用 MQSC 命令管理 IBM MQ](#)

## 使用 **setmqenv** 初始化当前命令环境

如果接收到指示 IBM MQ 命令未知或不可用的错误消息，请使用 **setmqenv** 命令来初始化当前命令环境，以便操作系统可以找到相应的命令。

### 关于此任务

每个操作系统都使用 **PATH** 环境变量中的目录来确定要运行的程序。如果当前 **PATH** 环境变量中没有任何目录包含指定名称的可执行程序，那么这将导致操作系统响应一条消息，指示它找不到命令，如以下示例中所示：

- Linux UNIX 在 UNIX and Linux 上：

```
% dspmqver
ksh: dspmqver: not found.
%
```

- Windows 在 Windows 上：

```
C:\> dspmqver
'dspmqver' is not recognized as an internal or external command,
operable program or batch file.
```

要解决此问题，应使用 **setmqenv** 命令来初始化当前环境，以便操作系统可以找到并执行相应的 IBM MQ 命令。

### 过程

1. 找到 **setmqenv** 命令在可用文件系统上的位置。

- Linux UNIX 在 UNIX and Linux 上：

```
% find / -name setmqenv 2>/dev/null
...
%
```

- Windows 在 Windows 上：

```
C:\> dir /b /s C:\setmqenv.cmd
...
C:\>
```

注：如果存在多个盘符，那么可以包含这些盘符，例如：`C:\> dir /b /s C:\setmqenv.cmd D:\setmqenv.cmd`。

可以忽略带有 **source** 或 **Maint** 的目录，因为它们几乎总是引用在安装修订包期间创建的命令的备份副本。

2. 使用 **setmqenv** 命令来初始化当前环境。

- Linux UNIX 在 UNIX and Linux 上：

```
% . <pathtoexecutable>/setmqenv -s
%
```

在 UNIX and Linux 上，必须对 **setmqenv** 命令进行寻源，以便可以修改当前环境。

- Windows 在 Windows 上：

```
C:\> <pathtoexecutable>\setmqenv.cmd -s
...
C:\>
```

3. 通过使用带有 **-o installation** 选项的 **dspmq** 命令，验证环境是否适合相应的 IBM MQ 安装。



- Linux UNIX 在 UNIX and Linux 上:

```
% dspmq -o installation
...
%
```

- Windows 在 Windows 上:

```
C:\> dspmq -o installation
...
C:\>
```

4. 如果需要, 请使用调整后的参数再次运行 **setmqenv** 命令, 以初始化环境, 从而进行更适当的 IBM MQ 安装。

例如:

- Linux UNIX 在 UNIX and Linux 上:

```
% dspmqver | grep Version
Version:      8.0.0.15

% dspmq -o Inst | sed -e "s/          //g"
QMNAME(V8QM1)  INSTNAME(Installation1) INSTPATH(/opt/mqm) INSTVER(8.0.0.15)
QMNAME(SPIDEY) INSTNAME(Installation4) INSTPATH(/opt/MQ91CD) INSTVER(9.1.5.0)
QMNAME(BOB)   INSTNAME(Installation3) INSTPATH(/opt/MQ92) INSTVER(9.2.2.0)

% . setmqenv -n Installation3

% dspmqver | grep Version
Version:      9.2.2.0
```

**注:** 仍必须对 **setmqenv** 命令进行有源操作, 以便可以修改当前环境。

- Windows 在 Windows 上:

```
C:\> dspmqver | findstr Version
Version:      9.1.0.5

C:\> dspmq -o inst | sed -e "s/          //g"
QMNAME(A)     INSTNAME(Installation2) INSTPATH(C:\IBM\MQ91LTS) INSTVER(9.1.0.5)
QMNAME(B)     INSTNAME(MQ91CD) INSTPATH(C:\IBM\MQ91CD) INSTVER(9.2.2.0)
QMNAME(C)     INSTNAME(MQ92) INSTPATH(C:\IBM\MQ92) INSTVER(9.2.2.0)

C:\> setmqenv -n MQ92

C:\> dspmqver | findstr Version
Version:      9.2.2.0
```

## 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, 帮助您在使用发布/预订集群或层次结构时检测和处理问题。

### 开始之前

如果您的问题通常与集群相关, 而不是与使用集群的发布/预订消息传递相关, 请参阅 [第 175 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 175 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。

- 更改集群中主题的集群路由是对发布/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。



**注意:** 尝试在现有管理主题（也使用 **CLROUTE(TOPICHOST)** 定义）下定义具有 **CLROUTE(TOPICHOST)** 的集群主题会导致报告错误消息 **AMQ8849**。

尽管上述文本中描述的配置似乎会导致一致的路由行为，但 IBM MQ 会防止冲突 (以及集群的远程成员上可能存在冲突的定义)，以消除拓扑中其他位置的不一致定义所导致的意外和潜在的随机路由行为。例如：

- 在新管理主题上方的主题树中附加通配符订户时，将废弃发布。
- 在数字上未创建相同的定义。

## 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消息，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。

每当将 RFH2 转换为消息属性时，还需要转换 <ibm> 文件夹；这将从传递或传递给应用程序的 RFH2 中除去指纹信息。

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。

### 关于此任务

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 71 页的『使用 Ping 测试通信』](#)
- [第 76 页的『重试链接的注意事项』](#)
- [第 73 页的『对通道拒绝运行的问题进行故障诊断』](#)
- [第 77 页的『解决通道停止运行的问题』](#)
- [第 71 页的『使用 dspmqrte 监视消息』](#)
- [第 78 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。

[第 294 页的『收集通道问题的信息』](#)

如果需要 IBM 支持人员提供帮助或在 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。

[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。



### 诊断消息和原因码

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。

当在登台或目标节点中发生此情况时，消息通道代理 (MCA) 可采用四种方式来处理这一情况：

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。

-  **Multi**  
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

## Using SUBSTATE to troubleshoot channels

If your channel is running slowly or appears to have no messages to send, use the SUBSTATE field to help resolve the problem.

## Environment



## Diagnosing the problem

### Sender channel status

```
CHSTATUS(MQ23.TO.MQ24) XMITQ(TO.MQ24) CONNAME(127.0.0.1)
CURRENT CHLTYPE(SDR) STATUS(RUNNING) SUBSTATE(MQGET)
RQMNAME(MQ24)
```

### Receiver channel status

```
CHSTATUS(MQ23.TO.MQ24) CONNAME(127.0.0.1) CURRENT
CHLTYPE(RCVR) STATUS(RUNNING) SUBSTATE(RECEIVE)
RQMNAME(MQ23)
```

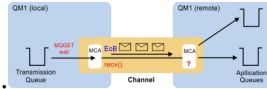
If you see the channel in this state, the channel behaves as it has no messages to send. As an aside, this can be a useful check to make at the same time as checking for uncommitted messages on the transmission queue, when you suggest that the channel is not moving the messages just put.

It is worth noting at this point that the resting state of a server-connection channel is similar to a receiver channel as it spends its time sitting in a network receive waiting for the client to send it an API call to issue.

## Slow channels

The sub status of channels is also useful, when other evidence suggests that your channel is running really slowly. Having described the resting state of channels in the previous section, if you see the sender, rather than the receiver channel, is sitting in a network receive, this can indicate that the line turn around on the network is slow.

This is because the sender channel is waiting on the receiver to process all the messages that have been sent in the batch and then respond to the End of Batch (EoB in the following diagram) notification to say



the batch has been successfully completed.

This state happens at the end of every batch, but it is usually so fleeting, that you are unlikely to see it. If you see it for a prolonged period of time it can be indicative of major network issues, such as:

- Router problems
- Retransmission of dropped packets
- Other issues that cause the network to run slow
- Something else that causes the receiving end to slow down, such as message retry, or issues in a exit

At this point looking at how the receiver channel is doing is the next step.

## Hung channels

When a channel is hung, either stuck in BINDING state for a long time, or even hung in RUNNING state where fields such as Number of Bytes Sent (NUMBYTES) is not increasing, then sub status can help.

When a channel, whether a queue manager channel or a server-connection channel, is starting up there are a number of different operations that it has to do before it can be considered to be RUNNING. All these different operations happen while the channel is in BINDING state.

To help break that down into a more granular view of what the channel is doing, there are sub status values for all these major operations. Some of the more common ones you see are listed below:

### Connecting to the network

When there are problems reaching the host, perhaps because the IP address is not longer in service, this can take a very noticeable number of seconds to timeout and return an error. Until it does, the channel shows SUBSTATE(NETCONNECT).

### Making a Domain Name Server (DNS) request

If the local DNS doesn't know how to resolve your CONNAME, it may be able to ask other related DNS (if your system is configured this way). This can all take considerable time. While this is happening, the channel shows SUBSTATE(NAMESERVER).

### SSL/TLS Handshake

When using SSL or TLS, by specifying a cipher spec on a channel definition in the SSLCIPH field, this means that the channel must complete an SSL handshake before any other data can flow across the channel. While this is happening, the channel shows SUBSTATE(SSLHANDSHAKE).

### In an exit

There are various different exits than can be used on a channel. While the exit is processing, the channel will show a sub state reflecting which exit it is in, for example, SUBSTATE(SENDEXIT). If the exit is taking a long time then you may see this for more than a fleeting moment.

## Resolving the problem

See [DISPLAY CHSTATUS](#) for a summary of all the sub states and when they can be seen.

## 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

如果通道从未运行过，那么问题的一个可能原因是 DQM 和通道可能未正确设置。

如果启动时自动同步由于某种原因而失败，那么通道可能处于不确定状态。不确定通道问题通常会自动解决，但在某些情况下，您可能需要手动再同步通道。

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。

**z/OS** 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 73 页的『对通道拒绝运行的问题进行故障诊断』](#) 的步骤 [第 74 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

### 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。

### **Shared channel recovery on z/OS**

Shared channel recovery is one of the benefits of using queue sharing groups on IBM MQ for z/OS.

The following table shows the types of shared channel failure and how each type is handled:

Type of failure	What happens
Channel initiator communications subsystem failure	The channels dependent on the communications subsystem enter channel retry, and are restarted on an appropriate queue sharing group channel initiator by a load-balanced start command.
Channel initiator failure	The channel initiator fails, but the associated queue manager remains active. The queue manager monitors the failure and initiates recovery processing.
Queue manager failure	The queue manager fails (failing the associated channel initiator). Other queue managers in the queue sharing group monitor the event and initiate peer recovery.
Shared status failure	Channel state information is stored in Db2, so a loss of connectivity to Db2 becomes a failure when a channel state change occurs. Running channels can carry on running without access to these resources. On a failed access to Db2, the channel enters retry.

Shared channel recovery processing on behalf of a failed system requires connectivity to Db2 to be available on the system managing the recovery to retrieve the shared channel status.

### Related concepts

[Preparing IBM MQ for z/OS for DQM with queue sharing groups](#)

## 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。当 IBM 支持人员询问时，您可能还需要配置跟踪。

### 关于此任务

如果在使用 IBM MQ Console 或 REST API 时遇到问题，请检查以下事项。

## 过程

- 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：

```
dspmqweb status
```



### 注意：

在 z/OS 上发出 **setmqweb** 或 **dspmqweb** 命令之前，必须设置 WLP\_USER\_DIR 环境变量，以便该变量指向 mqweb 服务器配置。

要设置 WLP\_USER\_DIR 环境变量，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](#)。



注意: **V 9.4.0** > **Linux**

在独立 IBM MQ Web Server 安装中发出 **setmqweb** 或 **dspmqweb** 命令之前，必须将 **MQ\_OVERRIDE\_DATA\_PATH** 环境变量设置为 IBM MQ Web Server 数据目录。

如果 mqweb 服务器已停止，请使用以下命令来启动该服务器：

```
strmqweb
```

**z/OS** 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](#)。

- 确保存在以下 mqweb 配置文件：

```
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：

– **ALW** *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*。

这些文件采用 UTF-8。要查看文件，可以使用下列其中一种方法：

- 从 z/OS UNIX System Services 命令行使用 **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*。除非发生错误，否则 *STDERR* 不包含任何消息。

- 如果无法从运行 mqweb 服务器的系统以外的主机访问 IBM MQ Console 或 REST API，请检查是否已启用远程连接。

发出以下命令以显示 mqweb 服务器配置：


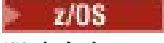
```
dspmqweb properties -a
```


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。通过输入以下命令来启用与 mqweb 服务器的远程连接，其中 `hostname` 指定安装了 IBM MQ 的 IP 地址或域名服务器 (DNS) 主机名：

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

使用星号 \*（用双引号括起）可指定所有可用的网络接口，如下示例中所示：

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

- 如果在 IBM MQ Console 的本地队列管理器窗口小部件中未显示任何队列管理器，请检查您是否将队列管理器与可使用 IBM MQ Console 管理的 mqweb 服务器位于同一主机上。
  -  在 IBM MQ Console 中仅列出 mqweb 服务器所在的安装中的队列管理器。
  -  在 z/OS 上，只有自上次 IPL 以来在与 mqweb 服务器版本相同的版本上启动的队列管理器才会在 IBM MQ Console 中列出。
- 如果在 IBM MQ Console 中浏览队列时截断消息，请使用 `setmqweb properties` 命令调整 `mqConsoleMaxMsgCharsToDisplay` 和 `mqConsoleMaxMsgRequestSize` 属性。
- 如果要将 messaging REST API 与远程队列管理器配合使用，并且无法连接到队列管理器，请检查以下事项：
  - 您要在资源 URL 中指定队列管理器的唯一名称，而不是队列管理器名称。
  - 您正在尝试连接到对 messaging REST API 可视的队列管理器。某些队列管理器可能仅可用于 IBM MQ Console。您可以使用 `dspmweb remote` 命令来验证队列管理器是否配置为对 messaging REST API 可视。查看 `visibility` 参数以检查队列管理器是否对 messaging REST API 可视。
  - 您未将星号作为队列管理器名称的前缀，以指定要连接到队列管理器组。要指定要连接到队列管理器组，必须在远程队列管理器配置中设置 `group` 属性。有关更多信息，请参阅 [设置远程队列管理器组以用于 messaging REST API](#)。
  - 确保 CCDT 文件位于 `dspmweb remote` 命令中的 `ccdtUrl` 参数指定的位置。

-  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 for z/OS UNIX System Services Web Components 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：

与服务器的通信中断，无法与服务器建立通信。

在用于启动 mqweb 服务器的过程中，请检查以下内容：

1. 检查 STEPLIB 库是否处于正确的级别，以及是否已获得 APF 授权。
2. 请检查 INSTDIR，USERDIR，PATH 和 LIBPATH 是否指向正确的路径。

在 z/OS UNIX System Services 中，输入以下命令，其中 `PathPrefix` 是 IBM MQ for z/OS UNIX System Services Components 安装路径：

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

此命令显示类似于以下输出的输出：

```
-rwxr-xr-t a-s- ... /mqm/V9R4M0/web/bin/dspmq
```

检查是否设置了 `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`)

## 下一步做什么

有关收集 IBM MQ Console 和 REST API 的跟踪的更多信息，请参阅第 463 页的『跟踪 REST API』和第 424 页的『跟踪 IBM MQ Console』。

## 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。

### 过程

#### 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. 检查 errors 子目录中的 FFST 报告。

如果在 errors 子目录中找到任何 FFST 报告，那么这意味着已正确安装 MQIPT，但配置可能存在问题。

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. 如果没有 FFST 报告，并且没有跟踪输出，那么这意味着未正确安装 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 报告，但确实有跟踪输出，请配置 MQIPT 连接 (ConnectionLog=true) 并使发送方尝试连接。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 427 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。

第 279 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 319 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。

### 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

如果将 MQIPT 作为 Windows 服务安装，或者作为 Linux 或 AIX System V init 服务安装，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。

请始终先尝试手动启动 MQIPT，然后再将其作为服务安装，以确认是否正确安装和配置了 MQIPT。

有关使用 `mqiptService` 命令自动启动 MQIPT 的信息，请参阅 [启动和停止 MQIPT](#)。

如果 MQIPT 务无法正确启动，请针对您的平台完成以下步骤。

### 在 Windows 系统上

#### Windows

注: 在 Windows 上，MQIPT 服务在 LocalSystem 帐户下运行。无法将服务配置为使用其他用户标识运行。

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运行 `mqiptService -debugevent` 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，请与 IBM 服务代表联系，并提供跟踪文件以及任何 FDC 文件和 `mqiptService -debugevent` 命令的诊断输出。

### 在 AIX and Linux 系统上

#### Linux

#### AIX

#### Linux

在 Linux 上，仅在支持 System V init 的操作系统上支持 MQIPT 服务。在不支持 System V init 的系统上，使用其他方法 (例如 `systemd`) 将 MQIPT 作为服务进行管理。

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。如果需要，请与 IBM 服务代表联系，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。

## Windows MQIPT 无法在 Windows 服务器上启动

IBM MQ Internet Pass-Thru (MQIPT) 通过因特网将客户机应用程序连接到队列管理器，或将队列管理器连接到队列管理器，方法是读取 `mqipt.conf` 文件中的配置数据。

`mqipt.conf` 文件是可在任何编辑器 (包括记事本) 中修改的文本文件。

**要点:** 如果使用记事本，那么在保存文件时必须注意。

如果收到以下错误:

```
MQCPI001 IBM MQ Internet Pass-Thru V2.1.0.1 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPE017 The exception java.io.IOException: MQCPE019 The stanza
[global] was not found before the following :
?[Uxbb][Uxbf][Ux23][Ux23][Ux23][Ux23]
```

MQIPT 找不到必需的配置详细信息。

MQIPT 在 `[global]` 节之前找到一些未知字符，再次在记事本中重新打开 `mqipt.conf` 文件不会显示这些未知字符。

问题在于保存配置文件的方式。

## 解决方案

缺省情况下，记事本使用 UTF-8 编码来保存配置文件，该编码在文件中的实际文本之前添加 "字节顺序标记"。这将导致 MQIPT 失败，并返回先前指定的错误。

要解决此问题，请选择 **文件/另存为** 菜单，并在 **Encoding** 字段中选择下拉列表中的 ANSI。再次保存文件

## 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。您通常只应该在 IBM 支持代表的指导下执行此操作，因为某些诊断设置可能会影响正常 MQIPT 操作。

**MQIPT\_JVM\_OPTIONS** 环境变量可用于通过使用命令行将诊断选项传递到底层 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 握手错误很有用。

### -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 AIX 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
```

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
```

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。

## 对 IBM MQ MQI client 应用程序问题进行故障诊断

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](#)。





### 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 AIX 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。

### 在 IBM MQ MQI client 通道连接上发生 ECONNRESET 错误

在与本地服务器的 IBM MQ MQI client 通道连接上，您正在间歇地从 TCP/IP 接收消息 AMQ9206 或 AMQ9208，这是 ECONNRESET 错误。

由同级重置连接导致 ECONNRESET TCP/IP 错误。当远程计算机由于某种原因而关闭已建立的连接时，会发生此情况。

## 症状

最常见的场景显示错误:

### AMQ9208I:

Error on receive from host <主机名>.

### 说明

An error occurred receiving data from <主机名> over TCP/IP. 这可能是由于通信故障所致。

### 操作

The return code from the TCP/IP receive() call was <xxxxx>. Record these values and tell the systems administrator.

您可能会收到不同的消息，例如:

### AMQ9209I

Connection to host <主机名> for channel <通道名称> closed.

下表显示了错误 *ECONNRESET Connection Reset by Peer:* 的不同操作系统的返回码

| 操作系统    | 小数    | 十六进制  |
|---------|-------|-------|
| AIX     | 73    | x49   |
| IBM i   | 3426  | xD62  |
| Linux   | 104   | x68   |
| Windows | 10054 | x2746 |
| z/OS    | 1121  | x461  |

## 诊断问题

ECONNRESET 错误通常指示 TCP/IP 网络中存在问题。

TCP/IP 发送重置的原因有很多:

- 不有序的连接终止 (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正在侦听的端口和 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 套接字选项进行编码, 因此 IBM MQ 不会导致重置。


- 对于连接无效的 TCP 段, 例如, 错误的应答或序号会导致重置。
- 连接请求超时。TCP 放弃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 放弃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

### 诊断提示和技巧:

请咨询网络管理员, 他们可以使用 TCP/IP 包和嗅探器跟踪来确定发生重置的原因。

 对于 z/OS, 请参阅:

- [有关在 IBM MQ 错误消息上设置 SLIP 并包括 TCP/IP CTRACE 和 TCP/IP PACKET 跟踪的指示信息](#)
- [z/OS UNIX 原因码](#) 表示在 CSQX208E 错误消息中找到的原因码的最后两个字节。

## 解决问题

IBM MQ 不会生成这些类型的错误。IBM MQ 只是通知您网络存在问题。

**要点:** 此问题超出了 "IBM MQ 支持" 的范围, 从 IBM MQ 角度而言, 无法执行任何操作来解决此网络问题。您需要与网络支持团队合作。

请参阅 [自动客户机重新连接](#) 以获取更多信息, 这对于开发团队在 IBM MQ MQI client 应用程序中使用很有用。

## IBM MQ .NET 问题故障诊断

用于帮助您解决运行 IBM MQ .NET 应用程序的问题的故障诊断信息。

### 相关任务

第 428 页的『跟踪 IBM MQ.NET 应用程序』

有几种不同的方法可以对 IBM MQ .NET 应用程序启用跟踪。通常只需要在 IBM 支持人员的请求下使用跟踪工具。

第 279 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 222 页的『对 XMS .NET 问题进行故障诊断』

用于帮助您解决运行 XMS .NET 应用程序的问题的故障诊断信息。

[开发 .NET 应用程序](#)

## 使用 .NET 样本应用程序和错误消息进行故障诊断

您可以使用 .NET 样本应用程序和错误消息来帮助对问题进行故障诊断。

### 过程

- 如果 IBM MQ .NET 应用程序未成功完成, 请运行其中一个 .NET 样本应用程序, 并遵循诊断消息中给出的建议。

[.NET 的样本应用程序](#)中描述了这些样本应用程序。

如果问题仍然存在, 并且您需要联系 IBM 支持团队, 那么可能会要求您开启跟踪工具。有关使用跟踪功能的信息, 请参阅第 428 页的『跟踪 IBM MQ.NET 应用程序』。

- 如果 amqmdnet.dll 或 amqmdxcs.dll 发生未知模块中发生类型为 System.IO.FileNotFoundException 的未处理异常错误, 请执行下列其中一项操作:
  - 请确保两者都已在全局组合件高速缓存中注册。
  - 创建指向 amqmdnet.dll 和 amqmdxcs.dll 组合件的配置文件。

您可以使用作为 .NET Framework 一部分提供的 msconfig.msc 来检查和更改组合件高速缓存的内容。

如果安装 IBM MQ 时 .NET Framework 不可用, 那么可能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。

有关此安装的信息将写入指定的日志文件, 在此示例中为 logfile.txt。

### 相关任务

[安装 IBM MQ classes for .NET](#)

## V9.4.0 IBM MQ .NET 客户机库抛出的公共 SSL 错误代码

从 IBM MQ 9.4.0 开始, IBM MQ .NET 客户机 (amqmdnetstd.dll) 诊断消息的改进为 SSL 相关问题提供了更具体的异常机制。



从 IBM MQ 9.4.0 开始，IBM MQ .NET 客户机库为与 SSL 相关的错误提供有意义的特定错误。MQRC 原因码与其他 .NET 客户机库 (例如 C) 一致。

下表显示了许多场景中的一些场景，其中 IBM MQ .NET 客户机库从 IBM MQ 9.4.0 抛出与其他客户机 .NET 库一致的错误：

| 表 2: 错误消息的比较                                                                        |                               |                                    |
|-------------------------------------------------------------------------------------|-------------------------------|------------------------------------|
| 场景                                                                                  | IBM MQ 9.4.0 之前的错误消息          | 来自 IBM MQ 9.4.0 的错误消息              |
| 为 SSL 密钥存储库 MQC.MQCA_SSL_KEY_REPOSITORY 或 MQEnvironment.SSLKeyRepository 提供的参数不正确时。 | 2059-MQRC_Q_MGR_NOT_AVAILABLE | 2381-MQRC_KEY_REPOSITORY_ERROR     |
| 使用 SSL 连接到 IBM MQ 服务器时，在客户机上设置了无效的密码套件。                                             | 2538-MQRC_HOST_NOT_AVAILABLE  | 2393-MQRC_SSL_INITIALIZATION_ERROR |
| 使用 SSL 连接到 IBM MQ 服务器时在客户机上设置了无效对等名称。                                               | 2059-MQRC_Q_MGR_NOT_AVAILABLE | 2399-MQRC_SSL_PEER_NAME_ERROR      |
| 当对等名称在使用 SSL 连接到 IBM MQ 服务器时不匹配时。                                                   | 2059-MQRC_Q_MGR_NOT_AVAILABLE | 2398-MQRC_SSL_PEER_NAME_MATCH      |
| 在 SSL 握手期间在客户机或服务上设置了无效证书时。                                                         | 2059-MQRC_Q_MGR_NOT_AVAILABLE | 2393-MQRC_SSL_INITIALIZATION_ERROR |
| 当用于在客户机和服务器之间提供 SSL 连接的通道未提供密码套件时。                                                  | 2059-MQRC_Q_MGR_NOT_AVAILABLE | 2393-MQRC_SSL_INITIALIZATION_ERROR |
| 当从客户机设置的密码套件算法与 IBM MQ 服务器不匹配时。                                                     | 2059-MQRC_Q_MGR_NOT_AVAILABLE | 2393-MQRC_SSL_INITIALIZATION_ERROR |
| 当没有对创建 IBM MQ 密钥存储库的文件夹的许可权时。                                                       | 2059-MQRC_Q_MGR_NOT_AVAILABLE | 2393-MQRC_SSL_INITIALIZATION_ERROR |

有关错误的更详细说明，请参阅 `MQ_DATA_DIRECTORY/qmgrs/errors/AMQERR*.log` 中的队列管理器日志。

#### 相关概念

[安装 IBM MQ classes for .NET](#)

## 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

[使用 IBM MQ classes for JMS](#)

[使用 IBM MQ 资源适配器](#)

[使用 IBM MQ classes for Java](#)

#### 相关任务

第 433 页的『[跟踪 JMS/Jakarta Messaging 和 Java 应用程序](#)』

提供了 JMS/Jakarta Messaging 和 Java 应用程序的跟踪工具，以帮助 IBM 支持人员诊断问题。您可以跟踪各种不同的资源。

第 279 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。

## 对 IBM MQ classes for JMS 问题进行故障诊断

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。

### 过程

- 如果应用程序未成功完成，请运行其中一个安装验证程序并遵循诊断消息中给出的建议。  
有关更多信息，请参阅 [The pointto point IVT for IBM MQ classes for JMS](#) 和 [The publish/subscribe IVT for IBM MQ classes for JMS](#)。

### 相关任务

第 433 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。

### 关于此任务

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。

### 过程

- 在 IBM MQ classes for JMS 配置文件中设置属性 com.ibm.msg.client.commonservices.log.outputName，或者在 java 命令中设置为系统属性。  
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](#)）。

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

- 要禁用日志输出，请将属性 com.ibm.msg.client.commonservices.log.status 设置为 **OFF**。此属性的缺省值为 **ON**。

## 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 方式的队列管理器时, 才支持使用 JMS 2.0 API。例如, 如果您正在使用迁移方式进行连接, 那么可能会收到此错误。
- **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 (Jakarta Messaging 3.0)

JM 3.0

```
*****[8/19/16 10:10:48:643 UTC] 000000a2
LocalExceptio E CNTR0020E: EJB threw an unexpected (non-declared) exception during
invocation of method "onMessage" on bean
"BeanId(SibSVTLiteMDB#SibSVTLiteMDBXA_RecoveryEJB_undeployed.jar#QueueReceiver, null)".
异常数据: jakarta.jms.JMSRuntimeException: 未能创建上下文
    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.svtlite.mdb.xa.SVTMDBBase.sendReplyMessage(SVTMDBBase.java:554)
    at sib.test.svtlite.mdb.xa.QueueReceiverBean.onMessage(QueueReceiverBean.java:128)
    at
sib.test.svtlite.mdb.xa.MDBProxyQueueReceiver_37ea5ce9.onMessage(MDBProxyQueueReceiver_37ea5ce9.j
ava)
    位于
com.ibm.mq.jakarta.connector.inbound.MessageEndpointWrapper.onMessage(MessageEndpointWrapper.java:
151)
    at com.ibm.mq.jms.MQSession$FacadeMessageListener.onMessage(MQSession.java:129)
    at com.ibm.msg.client.jms.internal.JmsSessionImpl.run(JmsSessionImpl.java:3236)
    at com.ibm.mq.jms.MQSession.run(MQSession.java:937)
    at com.ibm.mq.jakarta.connector.inbound.ASFWorkImpl.doDelivery(ASFWorkImpl.java:104)
    位于
com.ibm.mq.jakarta.connector.inbound.AbstractWorkImpl.run(AbstractWorkImpl.java:233)
    at com.ibm.ejs.j2c.work.WorkProxy.run(WorkProxy.java:668)
    at com.ibm.ws.util.ThreadPool$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
```

## 顶级池示例 (JMS 2.0)

JMS 2.0

```
*****[8/19/16 10:10:48:643 UTC] 000000a2
LocalExceptio 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.svtlite.mdb.xa.SVTMDBBase.sendReplyMessage(SVTMDBBase.java:554)
    at sib.test.svtlite.mdb.xa.QueueReceiverBean.onMessage(QueueReceiverBean.java:128)
    at
sib.test.svtlite.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:43)
... 12 more
```

## 较低级别的池示例 (Jakarta Messaging 3.0)

JM 3.0

```
*****
[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:
"嵌套异常为: jakarta.jms.JMSRuntimeException: 未能设置新的 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$Worker.run(ThreadPool.java:1892)
原因: [1] --> 消息: jakarta.jms.JMSRuntimeException: 未能设置新的 JMSContext
      类: 类 jakarta.jms.JMSRuntimeException
      Stack :
com.ibm.ejs.jms.JMSCMUtils.mapToJMSRuntimeException(JMSCMUtils.java:522)
      :
com.ibm.ejs.jms.JMSContextHandle.setupInternalContext(JMSContextHandle.java:241)
      :
com.ibm.ejs.jms.JMSManagedConnection.getConnection(JMSManagedConnection.java:783)
      :
com.ibm.ejs.j2c.MCWrapper.getConnection(MCWrapper.java:2336)
      :
com.ibm.ejs.j2c.ConnectionManager.allocateConnection(ConnectionManager.java:1064)
      :
com.ibm.ejs.jms.JMSConnectionFactoryHandle.createContextInternal(JMSConnectionFactoryHandle.java:43)
      :
com.ibm.ejs.jms.JMSConnectionFactoryHandle.createContext(JMSConnectionFactoryHandle.java:335)
      :
sib.test.svt.lite.mdb.xa.SVTMDBBase.sendReplyMessage(SVTMDBBase.java:554)
      :
sib.test.svt.lite.mdb.xa.QueueReceiverBean.onMessage(QueueReceiverBean.java:128)
      :
sib.test.svt.lite.mdb.xa.MDBProxyQueueReceiver_37ea5ce9.onMessage(MDBProxyQueueReceiver_37ea5ce9.java:-1)
      :
com.ibm.mq.jakarta.connector.inbound.MessageEndpointWrapper.onMessage(MessageEndpointWrapper.java:151)
      :
com.ibm.mq.jms.MQSession$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.jakarta.connector.inbound.ASFWorkImpl.doDelivery(ASFWorkImpl.java:104)
      :
com.ibm.mq.jakarta.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: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.jakarta.connector.inbound.MessageEndpointWrapper.onMessage(MessageEndpointWrapper.java:
151)
:
com.ibm.mq.jms.MQSession$FacadeMessageListener.onMessage(MQSession.java:129)
:
com.ibm.msg.client.jms.internal.JmsSessionImpl.run(JmsSessionImpl.java:3236)
: com.ibm.mq.jms.MQSession.run(MQSession.java:937)
:
com.ibm.mq.jakarta.connector.inbound.ASFWorkImpl.doDelivery(ASFWorkImpl.java:104)
:
com.ibm.mq.jakarta.connector.inbound.AbstractWorkImpl.run(AbstractWorkImpl.java:233)
: com.ibm.ejs.j2c.work.WorkProxy.run(WorkProxy.java:668)
: com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1892)

```

## 较低级别的池示例 (JMS 2.0)

### JMS 2.0

```

*****
[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-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.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，以下三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 3：对于激活规范或侦听器端口使用其中消息的目标启用“预读”

### 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。

## 原因 3：对于激活规范或侦听器端口使用其中消息的目标启用“预读”

“预读”运行方式：

- 在同步点之外的非持久消息上
- 在浏览消息时

浏览消息即 IBM MQ 激活规范或 WebSphere Application Server 侦听器端口在以破坏性方式使用消息之前所执行的操作。

因此，与在激活规范浏览线程启动时仅浏览单条消息不同，可一次性浏览（例如）100 条消息，然后缓慢地将消息馈送给线程以供从队列进行破坏性使用，并提供给应用程序消息驱动的 bean 类以进行处理。

因为队列管理器属性“消息标记浏览时间间隔 (MARKINT)”（用于再次浏览前使用已浏览消息）的缺省值为 5 秒，因此，如果有多个 Java Extended Edition 服务器针对同一队列运行激活规范，那么在消息驱动的 bean 有时间处理初始浏览的所有消息前，通常会导致活动规范或其他浏览线程再次浏览同一条消息。

仅第一个破坏性使用线程获取消息。将从队列管理器为所有后续破坏性使用线程指定 MQRC 2033 返回码，此返回码将转换为 JMSCC0108 消息。另外，在返回 2033 返回码前，队列管理器在整个队列中搜索消息，如果队列很深（数千条消息），那么可能花费大量时间。

## 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。

## 相关参考

### 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](#)

## 相关信息

[避免重复传递已浏览过的消息](#)

[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，然后由于超过过时超时而将其销毁，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过时超时设置为 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。第 100 页的『[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.

**Linux** **AIX** 对于 AIX and Linux，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)。缺省情况下，IBM MQ 配置 CHLAUTH 记录，以阻止所有 IBM MQ 管理员作为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](#)。

在 IBM MQ 9.3.0 中，使用客户机传输的 IBM MQ classes for JMS 应用程序所使用的缺省认证方法已从兼容性方式更改为 MQCSP 认证。另外，IBM MQ 9.3.0 中已声明可用于选择特定认证方式的方法以及优先的方法。这可能会导致先前成功连接到队列管理器的应用程序无法连接，原因码为 2035 MQRC\_NOT\_AUTHORIZED，因为当 IBM MQ classes for Java 或 IBM MQ classes for JMS 升级到 IBM MQ 9.3.0 或更高版本时，应用程序使用的认证方法可能会更改。有关 Java 客户机所使用的认证方式的更多信息，请参阅[向 Java 客户机进行连接认证](#)。

## 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：

- **Linux** **AIX** AIX and Linux 系统上的 *mqm*。
- **Windows** Windows 上的 *MUSR\_MQADMIN*。

由于这些用户是管理 IBM MQ 用户，因此缺省情况下会阻止这些用户，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 AIX and Linux 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 `javax/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：

```
JM 3.0 @Resource(name = "jms/MyResourceRef"  
    authenticationType = AuthenticationType.CONTAINER)  
private jakarta.jms.ConnectionFactory myCF
```

```
JMS 2.0 @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：

- JACL

```
wsadmin>set cell [ $AdminConfig getid "/Cell:mycell" ]  
mycell(cells/mycell|cell.xml#Cell_1)  
wsadmin>$AdminTask listWMQConnectionFactories $cell  
MyCF(cells/mycell|resources.xml#MQConnectionFactory_1247500675104)  
wsadmin>$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.listWMQConnectionFactories(cell)  
'MyCF(cells/mycell|resources.xml#MQConnectionFactory_1247500675104)'  
wsadmin>AdminTask.modifyWMQConnectionFactory('MyCF(cells/mycell|resos  
urces.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。 **z/OS** 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，请放宽通道认证记录 (CHLAUTH) 安全性，以便在要使用的服务器连接通道上不阻塞管理连接。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 IBM 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 184 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](#)

## 对 IBM MQ 资源适配器问题进行故障诊断

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：

```
JM 3.0 jakarta.jms.JMSEException: MQJCA0001: JMS 层中发生异常。 See the linked exception for details.
```

```
JMS 2.0 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：

```
JM 3.0 jakarta.jms.JMSEException: MQJMS2005: 无法为 "localhost:ExampleQM" 创建 MQQueueManager
```

```
JMS 2.0 javax.jms.JMSEException: MQJMS2005: 未能为"localhost:ExampleQM"创建 MQQueueManager
```

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: 发生 MQException : 完成代码为 2, 原因码为 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[配置 IBM MQ 资源适配器](#)。

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 AIX, 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。

## 相关任务

第 433 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 436 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 106 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 109 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           |
| 连接选项     | CONNECT_OPTIONS_PROPERTY   |
| 主机名      | 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           |
| 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，并且可能发生变更。

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，将配置文件指定为 Java 调用的一部分。

## 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。

| 表 3: 可以覆盖的属性 |                              |
|--------------|------------------------------|
| 属性           | 属性关键字                        |
| CCSID        | \$CCSID_PROPERTY             |
| 通道           | \$CHANNEL_PROPERTY           |
| 连接选项         | \$CONNECT_OPTIONS_PROPERTY   |
| 主机名          | \$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 **MQQueueManager** 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 **MQQueueManager** 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 **MQQueueManager** 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 104 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 105 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 106 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

使用此信息可帮助您诊断 Managed File Transfer (MFT) 中的错误。

### 相关任务

[第 443 页的『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。

[第 449 页的『Tracing Managed File Transfer for z/OS resources』](#)

The trace facility in Managed File Transfer for z/OS is provided to help IBM Support diagnose your problems and issues. You can trace various different resources.

[第 311 页的『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。

## 第 345 页的『Collecting information for Managed File Transfer for z/OS problems』

If you need assistance from IBM Support to resolve a problem when a Managed File Transfer (MFT) for z/OS agent, logger or command is reporting a problem or failing to work properly you first need to collect troubleshooting information to send to IBM Support to help find a solution. The information that is needed depends on the problem that you are seeing.



### 相关参考

#### MFT 命令

## 使用 MFT 的提示与技巧

以下是一些建议，可帮助您充分利用 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：

-   对于 AIX and Linux 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。

- 您可以设置名为 `BFG_JVM_PROPERTIES` 的操作系统环境变量以处理代理 JVM。您可以使用该环境变量来直接将属性传递给 JVM。但是，如果将该环境变量设置为错误的值，那么它会导致代理的 JVM 初始化失败。

  在 AIX and Linux 上，使用以下命令设置 `BFG_JVM_PROPERTIES`：

```
export BFG_JVM_PROPERTIES = "....."
```

 在 Windows 上，使用以下命令设置 `BFG_JVM_PROPERTIES`：

```
set BFG_JVM_PROPERTIES = "....."
```

注：每个 MFT 命令检查是否已设置 `BFG_JVM_PROPERTIES` 变量，然后在启动代理程序，记录器或 MFT 命令时将该变量的值作为参数传递

- 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 <b>fteCreateTransfer</b> 命令指定 <b>-w</b> 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。这也可能是由于，您向 <b>fteCancelTransfer</b> 命令提供的传输标识不正确。已忽略该取消请求。              |
| 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。                                                                              |



表 4: MFT 返回码 (继续)

| 返回码 | 短名称           | 描述                                                                          |
|-----|---------------|-----------------------------------------------------------------------------|
| 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。                                                |
| 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。                                                                |

表 4: MFT 返回码 (继续)

| 返回码 | 短名称             | 描述                                                                        |
|-----|-----------------|---------------------------------------------------------------------------|
| 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  | 数据库服务器存在问题      | 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。                                                    |
| 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。                                                        |

| 返回码 | 短名称              | 描述                                                               |
|-----|------------------|------------------------------------------------------------------|
| 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：

| 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 116 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 <action> 元素设置为值“progress”的传输日志进度消息中，报告的每个文件都具有带有 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：

| 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

使用以下任务来帮助您解决代理程序状态问题。

### 相关任务

[第 148 页的『对常见 MFT 问题进行故障诊断』](#)

常见 MFT 问题的核对表，通常直接用于定义和修订。

### 相关参考

[第 112 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。

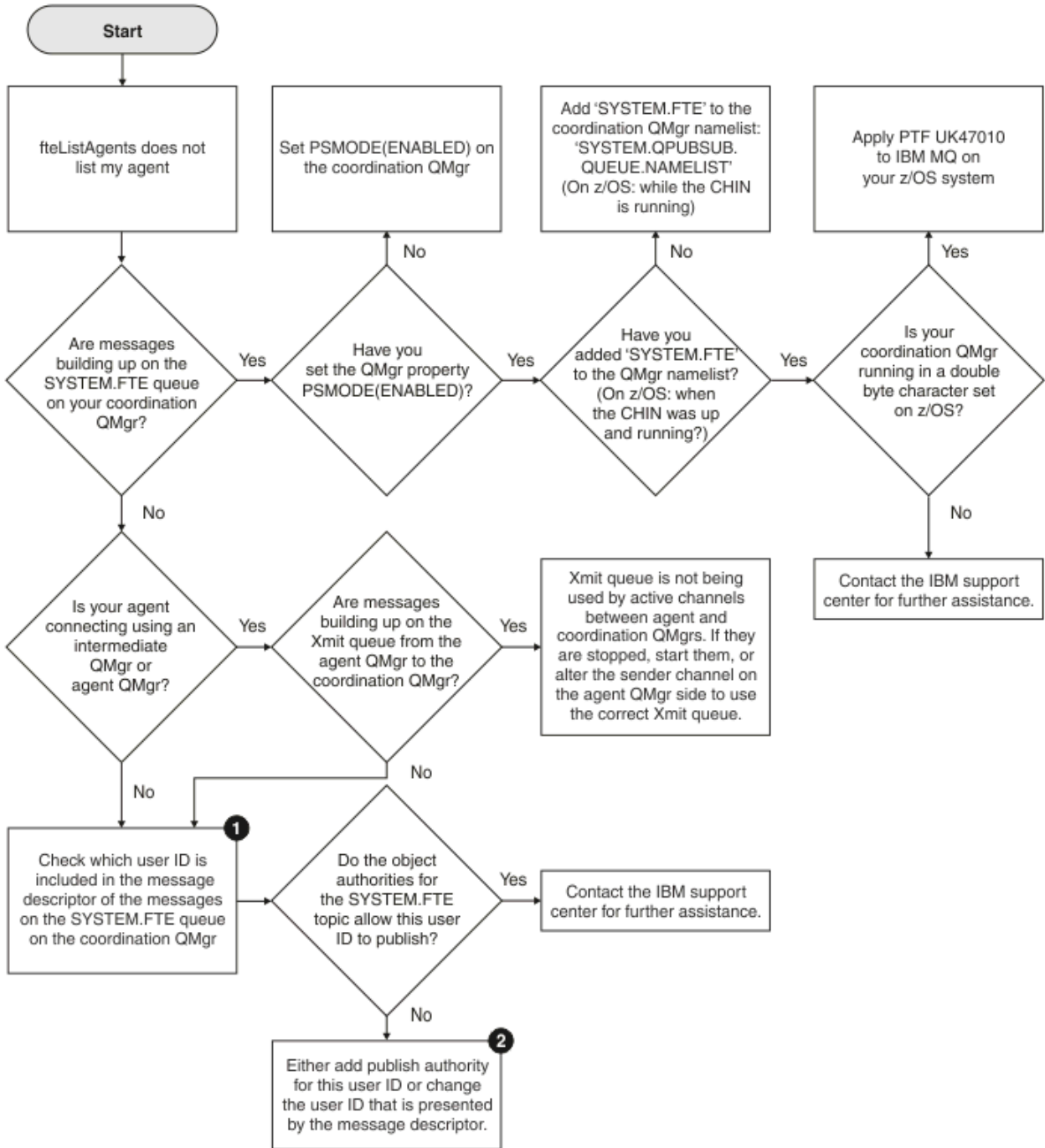
## 对 *fteListAgents* 命令未列出的 MFT 代理程序进行故障诊断

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，请使用流程图来调查原因。

## 关于此任务

使用以下流程图来帮助您诊断问题并决定下一步要执行的操作。

流程图:



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 153 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](#)。



## 对处于 **UNKNOWN** 状态的 **MFT** 代理进行故障诊断

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 插件将代理程序报告为处于 **UNKNOWN** 状态。

### 关于此任务

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。代理程序发布其状态的频率由以下代理程序属性控制：

#### **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) 插件使用这些出版物来确定代理程序的状态。为此，命令和插件执行以下步骤：

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。

具有旧状态消息的任何代理程序都将由 **fteListAgents** 和 **fteShowAgentDetails** 命令以及 IBM MQ Explorer MFT 插件报告为处于 **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。

要确定将代理程序的状态报告为 **UNKNOWN** 的原因，请完成以下步骤：

### 过程

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。再次运行时，请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。如果不是，请将其重新启动，然后使用 **fteListAgents** 或 **fteShowAgentDetails** 命令或 IBM MQ Explorer MFT 插件来查看现在是否正确报告了代理程序状态。

3. 如果代理程序和协调队列管理器正在运行，请在 **fteListAgents** 输出或 IBM MQ Explorer MFT 插件中检查代理程序的 *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，首先要做的是检查代理队列管理器和协调队列管理器的错误日志，以查看是否存在阻止代理发布其状态消息的任何授权问题。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，请检查状态消息是否未卡在 IBM MQ 网络中。验证用于将消息从代理队列管理器路由到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，并且状态消息正在通过 IBM MQ 网络流动，请检查队列管理器的排队发布/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 \(为协调队列管理器设置属性文件和目录\)](#)。

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已排队发布/预订引擎从 SYSTEM.FTE 队列中选取，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。

**Multi** 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 444 页的『自动收集 Managed File Transfer 代理跟踪信息』。

**z/OS** 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 452 页的『Collecting a Managed File Transfer for z/OS agent trace dynamically』。

- 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
```

**Multi** 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 445 页的『在多平台上跟踪 Managed File Transfer 命令』。

**z/OS** 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 455 页的『Tracing Managed File Transfer for z/OS commands』。

收集跟踪后，应将其提供给 IBM 支持人员进行分析。有关更多信息，请参阅第 280 页的『为 IBM 支持人员收集故障诊断信息』。

## 下一步做什么

从命令行查看状态时效。

发布的 **Status Age** 信息将显示为 **fteListAgents** 和 **fteShowAgentDetails** 命令的输出的一部分。

有关更多信息，请参阅 **fteListAgents** 和 **fteShowAgentDetails**。

在 **IBM MQ Explorer** 中查看状态年龄。

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](#)

## 对超时并报告 **BFGCLO214I** 消息的 **ftePingAgent** 进行故障诊断

**ftePingAgent** 是一个有用的命令行实用程序，可用于检查代理程序是否可访问以及它是否能够处理请求。

## 关于此任务

此处详细描述了此命令: **ftePingAgent**。运行时，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。

在消息到达临时队列后，**ftePingAgent** 命令将获取该消息。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：

BFGCL0793I: 在主机 <hostname> 上运行的代理 <agent\_name> 响应 ping 操作 (以 <time> 秒为单位)。

以下两个图显示了这一流程：

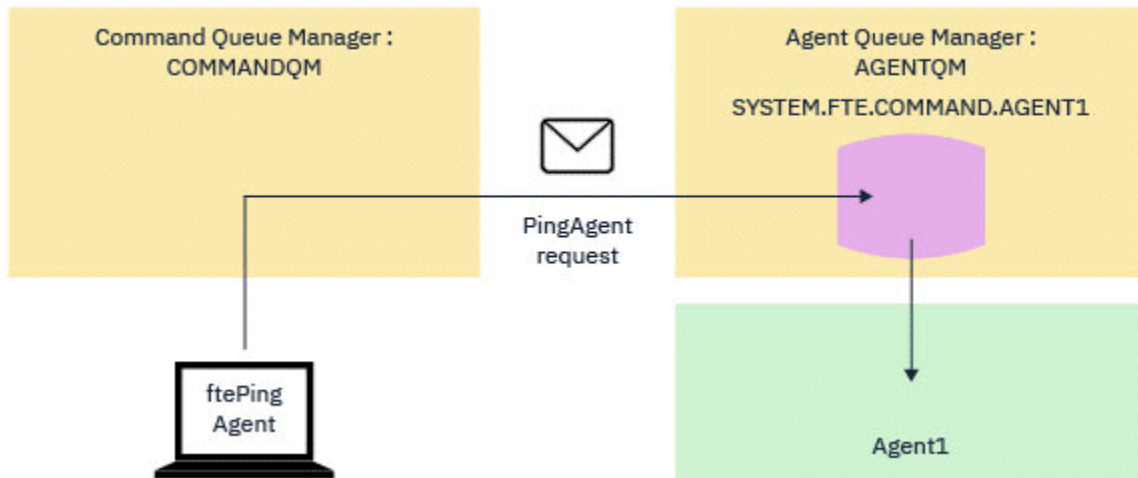


图 4: pingAgent 请求通过命令队列管理器进入代理队列管理器上的 SYSTEM.FTE.COMMAND.agent\_name 队列

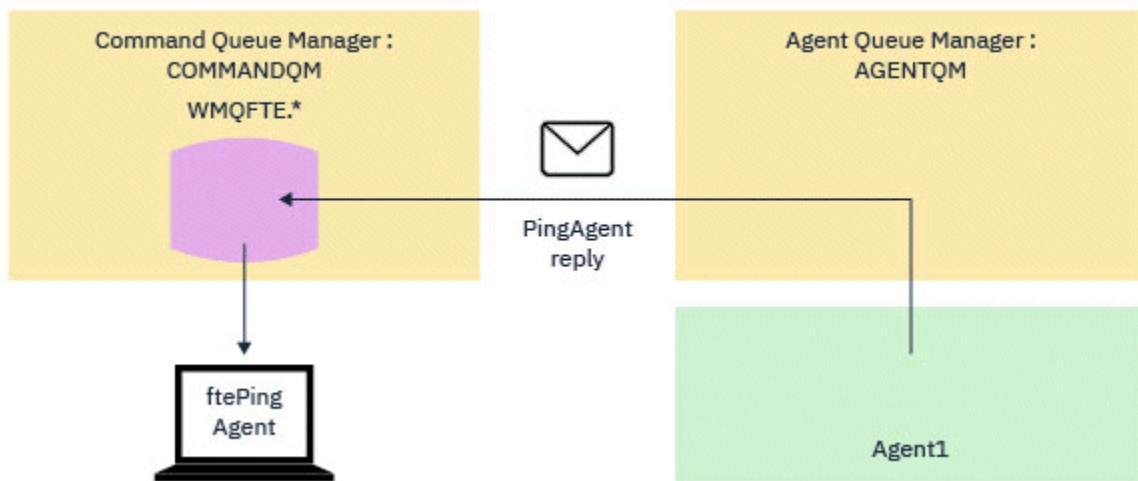


图 5: pingAgent 应答通过代理队列管理器返回到命令队列管理器。

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。

## 过程

1. 调查应答消息未到达的原因。

- 请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正在忙于处理请求，那么获取 Ping MFT 代理程序请求并发送回应答可能需要超过 5 秒的时间。要查看是否存在这种情况，请使用 **-w** 参数指定更长的等待时间间隔来重新运行 **ftePingAgent** 命令。

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
```

- c) 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。
2. 如果该命令仍报告 BFGCL0214I 消息，请在 Ping MFT 代理程序请求和 MFT 代理程序应答消息流经 IBM MQ 网络时跟踪这些消息。

执行此操作以查看 Ping MFT 代理程序消息是否曾经到达 SYSTEM.FTE.COMMAND.agent\_name 队列，并查看代理是否从队列中选取消息并发送回 MFT 代理应答消息。

- a) 在命令和代理队列管理器上启用队列管理器跟踪。
- b) 使用跟踪规范 com.ibm.wmqfte=all 对代理程序动态启用跟踪。

执行此操作的方法取决于正在运行代理程序的平台：

- **Multi** 对于在 IBM MQ for Multiplatforms 上运行的代理程序，请参阅第 444 页的『[自动收集 Managed File Transfer 代理跟踪信息](#)』。
  - **z/OS** 对于在 IBM MQ for z/OS 上运行的代理程序，请参阅第 452 页的『[Collecting a Managed File Transfer for z/OS agent trace dynamically](#)』。
- c) 使用跟踪规范 com.ibm.wmqfte=all 在启用跟踪的情况下运行 **ftePingAgent** 命令。
    - **Multi** 有关在 IBM MQ for Multiplatforms 上跟踪命令的信息，请参阅第 445 页的『[在多平台上跟踪 Managed File Transfer 命令](#)』。
    - **z/OS** 有关在 IBM MQ for z/OS 上跟踪命令的信息，请参阅第 455 页的『[Tracing Managed File Transfer for z/OS commands](#)』。
  3. 当该命令超时时，停止代理程序跟踪和队列管理器跟踪。使代理程序和队列管理器跟踪以及命令中的跟踪可供 IBM 支持人员进行分析。

## 对受管传输问题进行故障诊断

使用以下任务来帮助您解决受管传输问题。

### 相关任务

第 148 页的『[对常见 MFT 问题进行故障诊断](#)』

常见 MFT 问题的核对表，通常直接用于定义和修订。

### 相关参考

第 112 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。

## 对未完成的 MFT 传输进行故障诊断

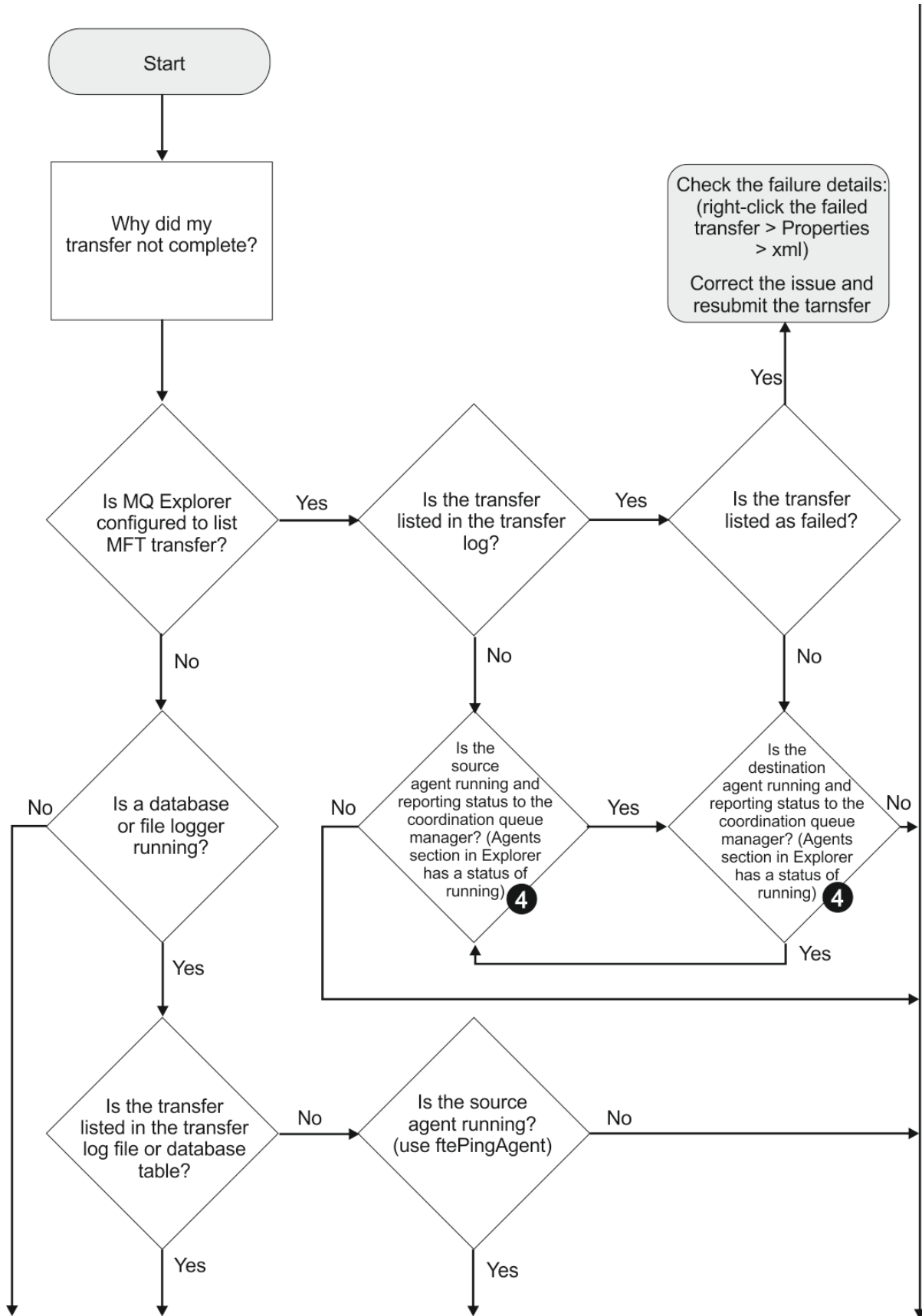
如果传输未完成，请使用流程图来调查原因。

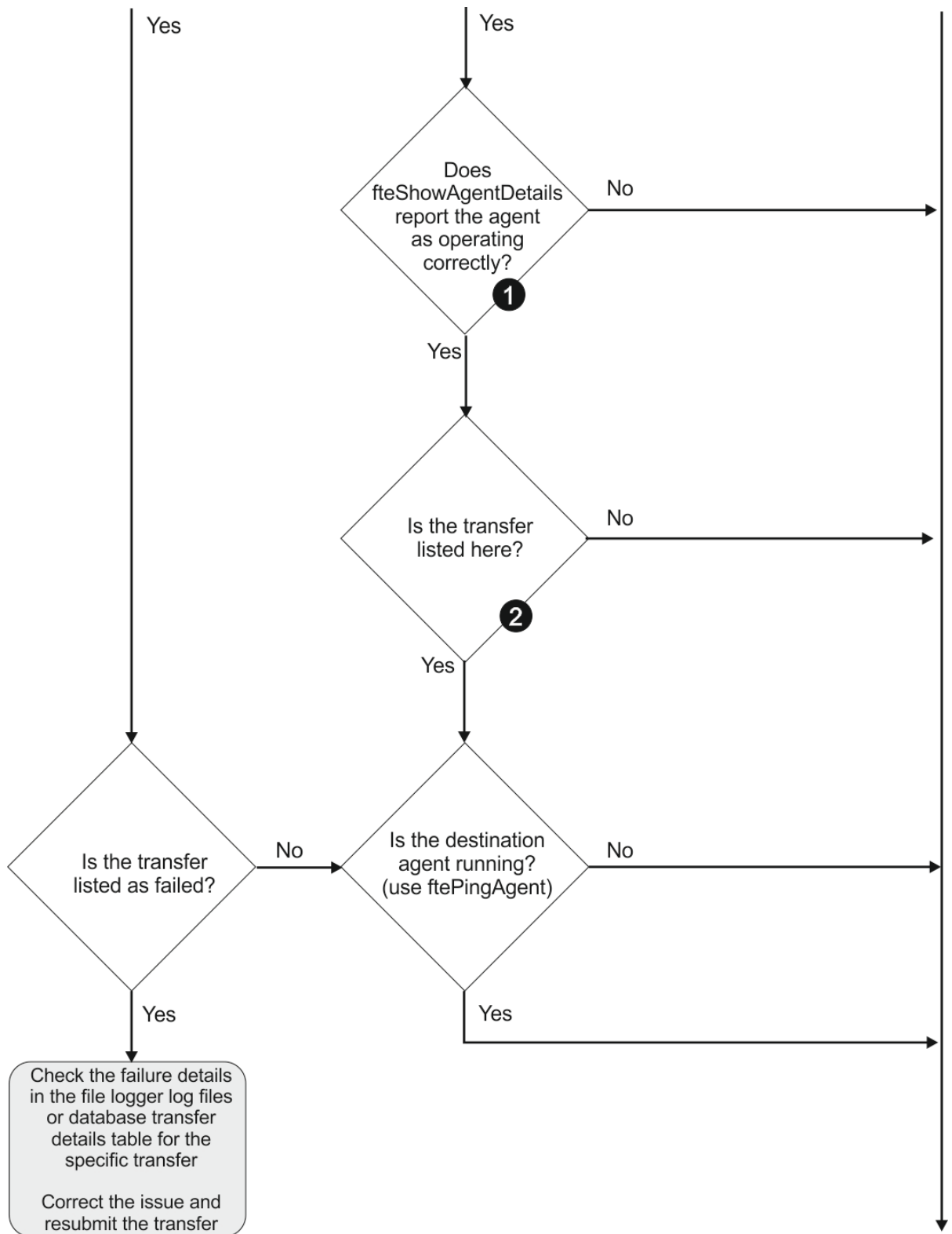
## 关于此任务

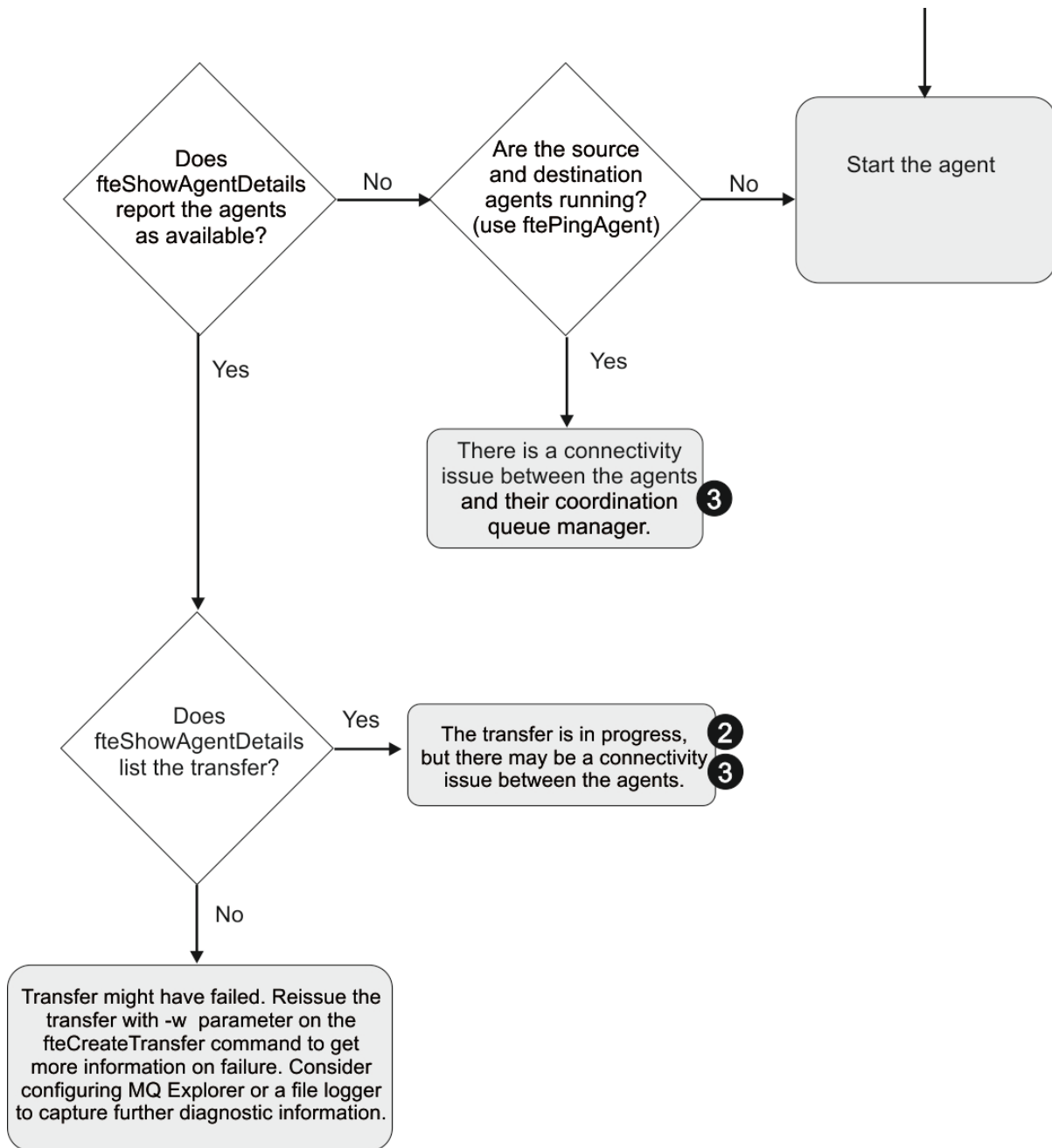
使用以下流程图来帮助您诊断问题并决定下一步要执行的操作。



流程图:







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。

## 对可能卡住的文件传输进行故障诊断

在负载较重的系统上，或者当源代理和目标代理之间存在网络问题时，传输有时会显示为处于排队或恢复状态。

### 关于此任务

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。

要确定问题的原因，请完成以下检查。

### 过程

1. 检查源和目标代理是否处于活动状态并响应新请求。
  - a) 运行 **ftePingAgent** 命令，或者在 IBM MQ Explorer 代理程序 面板中右键单击代理程序名称，然后选择 **Ping**。
  - b) 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。

- a) 运行带有 **-v** (详细) 参数的 **fteShowAgentDetails** 命令，或者在 IBM MQ Explorer 代理程序 面板中右键单击代理程序名称，然后选择 **属性**。
- b) 查看代理的当前传输活动。

如果正在运行的目标传输数达到或接近代理的最大目标传输数，那么这可能会解释为什么源代理的某些传输似乎卡住了。

3. 查看代理日志以查看当前是否存在连接问题。

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。

## 对到集群队列的失败文件传输进行故障诊断

使用 Managed File Transfer 将文件传输到队列时，如果使用的目标是集群队列或集群队列的别名，那么可能会获得原因码 2085 或 2082。如果设置了指向集群队列的远程队列定义，那么将解决此问题。

### 关于此任务

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。

要避免此问题，请完成以下步骤：

### 过程

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
```

## 对未运行或延迟的已调度文件传输进行故障诊断

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。

## 关于此任务

要解决该问题，请使用以下步骤之一：

## 过程

- 配置 agent.properties 文件中的 **maxSchedulerRunDelay** 属性以设置代理等待检查调度传输的最大时间间隔（以分钟为单位）。  
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](#)。
- 使用资源监视器而不是调度的传输。  
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 对传输 IBM i 保存文件时的错误进行故障诊断

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。

## 关于此任务

如果针对 MFT 代理发出多个并发请求以传输同一个 IBM i 保存文件，那么 Managed File Transfer 可能会生成以下一个或两个错误：

```
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"
```

## 过程

- 要多次同时传输同一保存文件，请使用多个源代理。请对每个并发传输使用不同源代理。
- 要使用单个源代理多次传输同一保存文件，请在提交每个新传输请求之前等待上一个传输请求完成。

## 相关任务

[在 IBM i 系统上传输文件](#)

[配置 MFT on IBM i](#)

[在 IBM i 上传输 QSYS.LIB 中的保存文件](#)

## 对因错误而失败的传输进行故障诊断 **BFGIO0341E**

如果受管传输正在将文件传输到受外部进程监视的位置，那么传输可能会失败，并产生以下错误：

**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。这是因为用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。

## 关于此任务

### 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，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，目标代理将删除所有称为 *destination\_filename.partnumber* 的临时文件，因为假定这些文件是由代理在受管传输期间创建的。

### 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。

要避免此问题，请完成以下任一步骤：

## 过程

- 配置外部进程以忽略临时文件。

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。

- 配置目标代理，使其不使用临时文件，并直接写入目标文件。

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。

## 对报告未找到文件的协议网桥代理进行故障诊断

协议网桥代理报告协议网桥连接的 SFTP 或 FTP 服务器返回 File not found 错误消息。这可能意味着发生了多个不同错误案例中的一个。

### 关于此任务

以下可能的错误情况可能导致 SFTP 或 FTP 服务器返回 File not found 错误。对于每个案例，将描述用于解决问题的其他信息和步骤。

## 过程

- **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，那么协议网桥代理会尝试根据用于登录到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 协议网桥代理仅支持允许通过其绝对文件路径访问文件的 SFTP 或 FTP 服务器。协议网桥代理不支持仅允许访问基于当前目录的文件的协议服务器。

### 相关参考

[协议网桥](#)

## 对资源监视器问题进行故障诊断

使用以下任务来帮助您诊断资源监视器的问题。

### 相关任务

第 148 页的『[对常见 MFT 问题进行故障诊断](#)』

常见 MFT 问题的核对表，通常直接用于定义和修订。

### 相关参考

第 112 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。

## MFT 目录资源监视器触发文件时进行故障诊断

目录资源监视器对与触发器规范匹配的文件目录进行轮询。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。在某些情况下，文件不会触发或触发两次。

### 关于此任务

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

在所有这些情况下，目录资源监视器会将触发文件标记为已处理，并忽略该文件，即使传输失败也是如此。

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。

### 示例

您可以运行命令来设置代理程序的所有资源监视器的信息级别输出。

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
```

有关 **logMonitor** 参数的详细信息以及有关如何使用不同选项的示例，请参阅 [fteSetAgentLogLevel](#)。

以下是代理程序的所有资源监视器的信息级别输出示例。

```
=====
[11/01/2022 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
=====
[11/01/2022 11:08:51:842 IST] 00000023    QMON           Monitor Started    Resource Monitor Started
[11/01/2022 11:08:51:844 IST] 00000025    QMON           Start Poll         New poll cycle started
[11/01/2022 11:08:51:924 IST] 00000023    MON1           Monitor Started    Resource Monitor Started
[11/01/2022 11:08:51:925 IST] 00000026    MON1           Start Poll         New poll cycle started
[11/01/2022 11:08:52:029 IST] 00000026    MON1           End Poll           Poll cycle completed in 105
milli seconds. Total items found in this poll [ 50 ]. Total items that matched the trigger condition [ 0 ]
[11/01/2022 11:08:52:055 IST] 00000025    QMON           End Poll           Poll cycle completed in 212
milli seconds. Total items found in this poll [ 72 ]. Total items that matched the trigger condition [ 0 ]
[11/01/2022 11:09:51:840 IST] 00000025    QMON           Start Poll         New poll cycle started
[11/01/2022 11:09:51:875 IST] 00000025    QMON           End Poll           Poll cycle completed in 34
milli seconds. Total items found in this poll [ 72 ]. Total items that matched the trigger condition [ 0 ]
[11/01/2022 11:09:51:924 IST] 00000026    MON1           Start Poll         New poll cycle started
[11/01/2022 11:09:51:969 IST] 00000026    MON1           End Poll           Poll cycle completed in 45
milli seconds. Total items found in this poll [ 50 ]. Total items that matched the trigger condition [ 0 ]
[11/01/2022 11:10:51:840 IST] 00000025    QMON           Start Poll         New poll cycle started
[11/01/2022 11:10:51:924 IST] 00000026    MON1           Start Poll         New poll cycle started
[11/01/2022 11:10:51:962 IST] 00000025    QMON           End Poll           Poll cycle completed in 121
milli seconds. Total items found in this poll [ 72 ]. Total items that matched the trigger condition [ 0 ]
[11/01/2022 11:10:51:963 IST] 00000026    MON1           End Poll           Poll cycle completed in 39
=====
```

```
milli seconds. Total items found in this poll [ 50 ]. Total items that matched the trigger condition [ 0 ]
[11/01/2022 11:10:55:063 IST] 00000041 MON1 Monitor Stopped Resource Monitor Stopped
[11/01/2022 11:10:55:079 IST] 00000041 QMON Monitor Stopped Resource Monitor Stopped
```

## 相关参考

[fteSetAgentLogLevel](#)

## 配置 MFT 资源监视器以避免代理程序超负荷

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。

## 关于此任务

资源监视器轮询目录或队列时，代理将完成以下阶段：

- 查找与触发器模式匹配的所有文件 (例如，目录中的所有 \*.txt 文件)。或者在队列中查找所有完整的消息组。
- 确定哪些文件是新文件或已更改文件，或者确定哪些组是队列中的新组。
- 对与前两个阶段中的条件匹配的文件或组启动传输。
- 添加到已传输的文件和组的列表，以便在更改之前不会再次传输这些文件和组。

对于目录监视器，源目录中的文件越多并且触发模式越广，针对已传输文件列表，代理必须解析和比较的文件列表就越大。

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。

## 过程

- 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。
- 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。

## 下一步做什么

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 \(创建 MFT 资源监视器\)](#)

## 对包含错误数据的队列资源监视器创建的目标文件进行故障诊断

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。

## 关于此任务

此问题由以下事件序列产生：

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。

## 示例

为避免迂到此问题，请使用 `fteCreateTransfer` 命令手动创建传输定义文件，并编辑该文件的 `<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。

在该示例中：

1. 源代理程序（也称为监视代理程序）称为 AGENT\_MON。
2. 目标代理称为 AGENT\_DEST。
3. 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>
```

至

```
<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。

## 对导致多个文件转至单个文件名的变量替换进行故障诊断

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。

## 关于此任务

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>
```

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。

## 示例

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](#)

## 对报告 *BFGDM0107W* 消息的资源监视器进行故障诊断

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`)。

## 关于此任务

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。
```

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，用于将它们们在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。

返回到先前的示例，在此示例中，监视器在单次轮询期间找到 200 条消息，并假定当监视器向代理提交 200 个任务时，所讨论的代理的 **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。

## 示例

可以帮助防止发生警告的一件事是 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。

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

- 在下次轮询期间，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。

假设您将轮询时间间隔增加到 2 分钟。这意味着在一次轮询期间提交的 25 个受管传输将在下次启动时完成，因此代理能够将它们分配给源传输槽，而不必将它们放入其积压中。如以下示例所示：

## 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

## 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

使用以下任务来帮助解决代理程序由于 java.lang.OutOfMemoryErrors 而停止的问题。

### 相关任务

第 148 页的『对常见 MFT 问题进行故障诊断』

常见 MFT 问题的核对表，通常直接用于定义和修订。

## 相关参考

第 112 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。

## 对具有 Java 堆耗尽错误的 MFT 代理程序进行故障诊断

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。

## 关于此任务

发生此问题时，受影响的代理程序 ABEND 并生成三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
```

发生此问题的原因是运行代理程序的 JVM 的 Java 堆内存耗尽。

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。

要帮助降低发生此问题的可能性，请完成以下步骤：

## 过程

### 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，这反过来会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大或更改 Java 堆，请设置 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 服务运行时增大或更改 Java 堆，请使用 **fteModifyAgent** 命令并指定 **-sj** 参数以在 Windows 服务上设置 **-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。

4. 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。

## 相关任务

第 139 页的『对具有本机堆耗尽错误的 MFT 代理程序进行故障诊断』

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。

## 对具有本机堆耗尽错误的 MFT 代理程序进行故障诊断

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。

## 关于此任务

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，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。

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

```
0SECTION      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

```
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:\ProgramData\IBM\MQ\mqft\logs\COORDQM\agents\AGENT1\javacore.20200109.143700.2652.0003.txt
```

由于运行代理程序的系统上的本机堆内存耗尽，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。

要帮助降低发生此问题的可能性，请完成以下步骤：

## 过程

### 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。

- 要在作为正常进程运行代理程序时降低或更改 Java 堆，请设置 `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，请使用 `fteModifyAgent` 命令并指定 `-sj` 参数以在 Windows 服务上设置 `-Xmx` 属性。

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 passwd0rd -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。

### 相关任务

第 137 页的『对具有 Java 堆耗尽错误的 MFT 代理程序进行故障诊断』

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。

## 对记录器问题进行故障诊断

使用以下任务来帮助您解决记录器问题。

### 相关任务

第 148 页的『对常见 MFT 问题进行故障诊断』

常见 MFT 问题的核对表，通常直接用于定义和修订。

### 相关参考

第 112 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。

## 对 Oracle 上 MFT 数据库模式的更新进行故障诊断

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. 在每次出现的文本中，`SELECT FTELOG.sequence_name.nextval` 将文本 `FTELOG` 替换为现有序列所在的模式的名称。
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 记录器错误处理和消息拒绝

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。没有任何内容写入输出日志，因此您必须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** 属性来调整导致常规错误所需的连续每条消息错误数。

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。有关查看拒绝队列内容，查看拒绝消息的原因以及重新处理消息的信息，请参阅 [第 143 页的『重新处理来自拒绝队列的 MFT 消息』](#)。

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](#)。

## 重新处理来自拒绝队列的 MFT 消息

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。您可以查看拒绝队列的内容，了解拒绝消息的原因以及 (在某些情况下) 重新处理消息。

## 开始之前

此任务假定您已阅读 [第 142 页的『MFT 记录器错误处理和消息拒绝』](#)，因此了解以下信息：

- 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。没有任何内容写入输出日志，因此您必须定期检查或持续监视拒绝队列以检测拒绝的消息。
- 如果连续拒绝过多消息，那么会将其视为一般错误。您可以使用 **wmqfte.max.consecutive.reject** 属性来调整导致常规错误所需的连续每条消息错误数。
- 对于独立数据库记录器，输出日志位于目录 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name` 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。

## 关于此任务

产生每条消息错误的消息将移动到拒绝队列中。有时，您可能希望重新处理拒绝队列中的消息。在 [第 142 页的『MFT 记录器错误处理和消息拒绝』](#) 中描述的示例中，通过在数据库中使用两个字符的作业名列，可以在增大数据库列的宽度之后成功处理消息。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。

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](#)。

## 示例

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。您将看到一个名为 **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。

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，记录器创建自己的受管预订时，输入队列由队列管理器定义，并具有类似 `SYSTEM.MANAGED.DURABLE.49998CFF20006204`。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。

## 对无法写入数据库表的 MFT 记录器进行故障诊断

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。

### 关于此任务

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。

要解决此问题，请完成下列步骤：

### 过程

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 8 KB，请增大数据库的页面大小。

创建页大小至少为 8 KB 的表空间和页大小至少为 8 KB 的关联缓冲池。有关如何执行此操作的信息，请参阅数据库的文档。

## 对 Connect:Direct 网桥进行故障诊断

使用以下任务和示例来帮助您诊断从 Connect:Direct 网桥返回的错误。

### 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。

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](#)

## 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 (//) 开头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`。

### 相关任务

第 144 页的『对 Connect:Direct 网桥进行故障诊断』

使用以下任务和示例来帮助您诊断从 Connect:Direct 网桥返回的错误。

[与 Connect:Direct 节点传输数据集](#)

## 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](#)

## 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。

### 关于此任务

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](#)。



## 示例

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](#)。

## 对与 *Connect:Direct* 节点之间的文本传输的数据转换问题进行故障诊断

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。





## 关于此任务

在 MFT 代理程序与 Connect:Direct 节点之间以文本方式传输文件时，将对文本执行代码页和行尾字符转换。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。

## 过程

- 对于网络映射中的每个远程节点，请选择正确的操作系统值。

例如，如果 Connect:Direct 网桥节点位于 Windows 系统上，请确保对于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：

-  **Windows** 如果远程节点位于 Windows 系统上，请选择 Windows。
-  **AIX**  **Linux** 如果远程节点位于 AIX 或 Linux 系统上，请选择 UNIX。
-  **z/OS** 如果远程节点位于 z/OS 系统上，请选择 OS/390。

Connect:Direct 网桥不支持传输到其他操作系统上的远程节点。

- 对于您将文件传输到或从中传输文件的每个远程节点，请在 Connect:Direct 网桥代理配置目录的 `ConnectDirectNodeProperties.xml` 文件中指定远程 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](#)。

## 相关信息

[使用 MFT 传输文本文件](#)

## **Troubleshooting failed transfers to PDS or PDS members through the Connect:Direct bridge**

If the destination of a transfer is a Connect:Direct node on z/OS and is a PDS or PDS member, the transfer fails if the **-de** parameter has not been specified with a value of overwrite.

## Procedure

- If you submitted the transfer by using the **fteCreateTransfer** or **fteCreateTemplate** command, perform the following steps:

- a) Change the command that you submitted to include **-de** overwrite.
- b) Submit the command again.
- If you submitted the transfer by using the IBM MQ Explorer plugin, perform the following steps:
  - a) Specify the source and destination information in the **Create New Managed File Transfer** wizard.
  - b) Select **Overwrite files on the destination file system that have the same name**.
  - c) Submit the command again.

## 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。

## 关于此任务

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。AIX and Linux 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。

## 过程

- 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

- 另外，请更改属于 Connect:Direct 网桥的 Connect:Direct 节点的最大并发 API 连接数。

控制此数字的 Connect:Direct 参数为 **api.max.connects** (如果您的节点在 AIX and Linux 上) 或 **max.api.connects** (如果您的节点在 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 **AIX and Linux** 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**

对于由文件传输调用的 **Connect:Direct** 进程，可以配置 **Connect:Direct** 网桥代理以将日志信息写入 **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

#### **相关任务**

第 144 页的『[对 \*\*Connect:Direct\*\* 网桥进行故障诊断](#)』

使用以下任务和示例来帮助您诊断从 **Connect:Direct** 网桥返回的错误。

#### **相关参考**

[MFT agent.properties 文件](#)

### **对其他 MFT 问题进行故障诊断**

使用以下任务来帮助您诊断 **Managed File Transfer (MFT)** 的常规问题。

#### **相关参考**

第 112 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。

### **对常见 MFT 问题进行故障诊断**

常见 **MFT** 问题的核对表，通常直接用于定义和修订。

## 过程

- 如果您看到 **fteCreateAgent** 命令的以下输出，那么它指示无法联系协调队列管理器，并提供 IBM MQ 原因码以说明原因：

```
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。
```

执行以下检查：

- 请检查协调队列管理器当前是否可用。
- 请检查您是否正确定义了配置。
- 如果您正在使用用户出口例程，并且在调用用户出口时或在调用出口之后发生故障（例如，产品故障或断电），请检查是否未多次调用用户出口。
- 如果您在具有由 DHCP（而不是静态 IP 地址）分配的 IP 地址的系统上具有队列管理器的代理程序，那么和该代理程序使用客户机 TCP/IP 连接来连接到该系统，请在设置了以下系统环境变量的情况下启动该代理程序：

-  在 Windows 上：

```
set BFG_JVM_PROPERTIES="-Dsun.net.inetaddr.ttl=value"
```

-   在 AIX and Linux 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]
```

当代理程序使用 32 位版本的 Java 运行时，如果 LD\_LIBRARY\_PATH 或 LIBPATH 环境变量设置为在 32 位版本之前引用 64 位版本的库，那么会发生此错误。

要解决此问题，请设置 Managed File Transfer 代理属性 `javaLibrary` 路径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`。



- 如果通过在代理属性文件中指定 `authorityChecking=true` 来启用用户权限检查，并且即使用户在相关权限队列上具有必需的权限，所有权限检查也会失败，请确保运行代理的用户在代理队列管理器上具有 ALT\_USER 访问控制。
- 如果已通过在代理属性文件中指定 `authorityChecking=true` 来启用用户权限检查，并且 IBM MQ 错误消息已写入代理 `output0.log` 文件，请执行下列其中一项操作：
  - 忽略消息，因为代理程序不受影响。
  - 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。
- 如果您已编辑代理属性文件，并且代理程序尚未进行编辑，请重新启动代理程序以确保代理程序读取新属性。

### 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，那么这指示尚未在运行配置或管理脚本的环境中设置环境变量 `_BPXK_AUTOCVT=ON`：

FSUM7332 syntax error: got (, expecting Newline

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](#)。

## 使用 JZOS 对 MFT 进行故障诊断

在使用 JZOS 运行 MFT 时迁到问题时要执行的步骤。

### 过程

1. 如果 JZOS 未成功处理：

- 将 `, PARM='+T'` 添加到 JCL。例如：

```
//MQMFT EXEC PGM=JVMLDM86,REGION=0M,PARM='+T'
```

- 将 `set -x` 添加到环境文件。

2. 如果迁到以下任一错误，那么环境文件和 Managed File Transfer 命令存在问题。

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

原因可能是指定的路径无效。

3. 从您的环境文件中找到 **BFG\_PROD** 的值。

- a) 转至 OMVS 并使用 `ls -ltr` 命令。

例如，如果 **BFG\_PROD** 是 `/mqm/V9R2M0/mqft`，请输入以下命令：

```
ls -ltr /mqm/V9R2M0/mqft/bin/fteBatch
```

- b) 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。

- c) 解决任何问题。

4. 如果 JCL 仍然无法正确处理：

- a) 请在 z/OS UNIX System Services 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。

- b) 保存该文件。

- c) 从命令行使用以下命令以允许文件运行：

```
chmod +x myenv
```

- d) 要报告 `myenv` 文件中的任何错误，请运行以下命令：

```
. myenv
```

请注意，上述命令为 *period blank filename*。

- e) 更正 `myenv` 和环境文件中的任何错误。

## 对 MFT 文本数据转换错误 BFGIO0060E 进行故障诊断

如果 Managed File Transfer 文本传输失败并返回错误 BFGIO0060E "文本数据转换已失败"，该怎么办？

### 关于此任务

由于以下两种原因之一，文本传输可能会失败，并产生错误 BFGIO0060E：

- 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。
- 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。

## 过程

- 如果某些字符可接受不转换，请在目标代理上定义替换字符序列，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：输入/输出](#)。
- 传输与源代理不具有相同编码的源文件：
  - 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](#)。
  - 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[fteCreateTransfer：启动新的文件传输](#)。
  - 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](#)。
- 检查您是否为传输选择了正确的源文件编码：
  - a) 将目标文件编码设置为 UTF-8。
  - b) 以文本方式传输文件。
  - c) 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。

Linux

AIX

## 对在不记录诊断信息的情况下消失的 MFT 代理进程进行故障诊断

在 AIX and Linux 平台上，如果代理程序进程已消失，但代理程序日志文件未包含任何解释，那么这可能是由代理程序的启动方式导致的。

## 关于此任务

通过以下方式检查代理程序诊断信息：

## 过程

1. 检查代理程序日志文件是否指出代理程序已停止。
2. 检查代理锁定文件 `agent.lck` 是否仍然存在。
3. 使代理程序运行时间超过调用代理程序的脚本的持续时间。

如果从 shell 脚本 (例如) 启动代理程序，那么在脚本完成时将除去与该脚本相关联的所有子进程 (包括代理进程)。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：

- a) 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，代理程序将继续运行。

## 对不安全的 MFT 代理程序或记录器配置进行故障诊断

如果 Managed File Transfer 进程在启动时检测到配置文件包含敏感信息，是密钥库或信任库文件，并且具有系统范围的读、写或删除许可权，那么该进程不会启动。如果在运行时检测到该条件，那么 Managed File Transfer 将生成一条警告消息并忽略配置文件的内容。这会影响协议网桥和 Connect:Direct 网桥功能，因为如果在代理程序运行期间配置发生更改，那么它们会重新装入配置。

## 关于此任务

要确定问题的原因，请完成以下检查：

## 过程

1. 从生成的错误消息中，标识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。对于协议网桥或 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](#)

## 对在 *SYSTEM.MANAGED.DURABLE* 队列或填充文件系统

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。

## 过程

在 *SYSTEM.MANAGED.DURABLE* 队列。

- 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。

避免在持久队列上构建消息。

- 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:

- a) 选择 **窗口 > 首选项 > IBM MQ Explorer > Managed File Transfer**
- b) 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。

持久预订的名称以显示该预订是由 IBM MQ Explorer MFT 插件，主机名和用户的名称创建的。例如，MQExplorer\_MFT\_Plugin\_HOST\_TJWatson。

## 相关任务

[保留 MFT 日志消息](#)

## 对在 *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。

## 关于此任务

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，请完成下列步骤:

## 过程

1. 检查 `SYSTEM.QPUBSUB.QUEUE.NAMELIST` 名称列表是否已正确设置并包含 `SYSTEM.FTE` 队列的条目。

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)
ALTDATE(2022-03-04)                ALTTIME(14.34.37)
```

2. 如果 `NAMES` 属性不包含 `SYSTEM.FTE`，请使用以下 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

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。如果错误情况停止此发布，那么可以在尝试发布之前检查队列上的消息。

## 关于此任务

要帮助诊断错误情况，请通过完成以下步骤来检查队列上的消息：

## 过程

1. 在协调队列管理器中禁用发布/预订引擎。

使用 `IBM MQ Explorer` 或 `MQSC` 命令完成此步骤。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。

使用 `IBM MQ Explorer` 来浏览协调队列管理器的 `SYSTEM.FTE` 队列。

### IBM MQ Explorer:

- a. 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。
- b. **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。

c. 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，请使用其他程序来浏览队列并显示找到的消息的所有方面。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 JOHNDOE
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)
```

## 配置 *Redistributable Managed File Transfer components* 时可能出现的错误

配置 Redistributable Managed File Transfer components 时，可能会看到错误“无法装入 Windows 的本机库”或“不支持使用绑定方式”。

### 无法装入 Windows 的本机库

**Windows** 必须在系统上安装可从 Microsoft 获取的 Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 库才能使用 Redistributable Managed File Transfer Agent。请参阅 [The latest supported Visual C++ downloads](#)。

对于 Continuous Delivery，Redistributable Managed File Transfer Logger 还需要 Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 库。

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 (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。

在 IBM MQ 9.3.0 中，您可能还会看到 Redistributable Managed File Transfer Logger 的此错误，它仅支持以客户机方式连接到协调队列管理器的 FILE 类型记录器。

## 相关任务

[下载和配置 Redistributable Managed File Transfer components](#)

## 相关参考

[BFGCL0001 - BFGCL9999](#)

[BFGUB0001 - BFGUB9999](#)

## 使用 UAC 和虚拟存储器与 MFT 进行故障诊断

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。

## 关于此任务

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，如果非 IBM MQ 管理员使用标准操作系统工具更改或重建目录结构，那么可能会将新结构转移到虚拟存储器中。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。
- 代理程序可能无法启动，报告消息 BFGCL0315，但未提供支持的原因码。
- 在代理报告的位置找不到任何日志文件。
- 使用 **-F** 参数启动代理程序时，该代理程序可能无法启动，并报告以下消息：

```
The current directory is invalid
```

要更正所有这些情况，请完成以下步骤：

## 过程

- 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** 对作为 Windows 服务运行的 MFT 代理程序或记录器进行故障诊断

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。如果您在使用这些 Windows 服务时遇到问题，那么可以使用服务日志文件和本主题中的步骤来诊断问题。

## 关于此任务

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。

## 过程

- 选择日志记录级别。

使用 **fteCreateAgent**, **fteCreateCDAgent**, **fteCreateBridgeAgent**, **fteModifyAgent**, **fteCreateLogger** 或 **fteModifyLogger** 命令将代理程序或记录器作为 Windows 服务运行时, 可以使用 **-sl** 参数来选择日志记录级别。此参数的可能值为 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。

- 选择日志文件数。

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。

- 授予 "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:

- a) 从开始菜单, 单击管理工具 > 本地安全策略。
- b) 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。
- c) 在策略和安全性设置窗格中, 双击作为服务登录。
- d) 单击添加用户或组, 然后将要运行服务的用户添加到具有 **Log on as a service** 权限的用户列表。

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 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。如果您不想显示密码, 请完成以下步骤:

- a) 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。

- b) 打开 Windows“服务”窗口。  
c) 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 是您指定给独立记录器用作其属性集的协调队列管理器。

有关属性集合的更多信息, 请参阅 [fteStartLogger](#) 和 [fteModifyLogger](#)。

- d) 在**属性**窗口中, 选择**登录**选项卡。  
e) 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。  
f) 单击**确定**。

### 相关任务

[将 MFT 代理作为 Windows 服务启动](#)

### 相关参考

[fteCreateAgent](#)

[fteModifyAgent](#)

[fteCreateCDAgent](#)

[fteCreateBridgeAgent](#)

[fteCreateLogger](#)

[fteModifyLogger](#)

[fteStartLogger](#)

## Windows 如何处理代理程序和记录器 JVM 选项

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



## 当 MFT 未读取 AMS 的密钥库属性时进行故障诊断

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”。

### 关于此任务

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：

### 过程

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。

### 结果

注：如果 Java AMS 正在绑定方式下运行，那么如果密钥库配置文件不在缺省位置中，那么错误 AMQ9062 将显示在队列管理器的错误日志中。

## 除去修订包时避免发生 BFGSS0023E 错误

如果从安装中卸载修订包以返回到产品的先前版本，并且在执行卸载时与安装相关联的代理程序涉及受管传输，那么该代理程序无法启动，并且将报告 BFGSS0023E 错误。您可以通过完成一些步骤来避免在代理程序重新启动时出现 BFGSS0023E 消息，从而避免此错误。

### 关于此任务

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 代理在停止并降级到 IBM MQ 8.0.0 Fix Pack 4 级别时运行了一些正在进行的传输，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。

如果要除去修订包，请完成以下步骤以防止在重新启动 Managed File Transfer 代理程序时出现 BFGSS0023E 消息：

### 过程

1. 确保所有代理都已完成其受管传输。

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

#### 相关任务

[第 297 页的『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。

[第 298 页的『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。

[第 300 页的『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。

## 对未送达的消息进行故障诊断

如果消息未成功送达并因此放入死信队列，那么您可以浏览该队列，并决定如何处理找到的任何消息。

### 关于此任务

由于某些原因而无法送达的消息会放入死信队列。每个队列管理器通常都有一个本地队列用作死信队列，这样就可以存储无法送达正确目标的消息，以供之后检索。

### 过程

1. 使用 MQSC **DISPLAY QUEUE** 命令检查队列是否包含任何消息。
2. 如果队列包含消息，请使用提供的浏览样本应用程序 (**amqsbcbg**)，通过 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。
3. 根据放入队列的消息的原因，决定如何处理在死信队列中找到的任何消息。  
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。

IBM MQ 提供了一个死信队列处理程序，用于指定如何处理或除去在死信队列中找到的消息。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](#)。

#### 相关概念

[死信队列](#)

#### 相关参考

[runmqdlq \(运行死信队列处理程序\)](#)

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

## Windows Linux AIX 遥测日志、错误日志和配置文件的位置

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。

**V 9.4.0** **V 9.4.0** (可选) 如果启用了 MQTT 的 JSON 格式日志记录, 那么将写入 JSON 日志文件。  
日志路径为:

```
WMQ data directory\Qmgrs\qMgrName\errors\mqxr.json
```

**V 9.4.0** **V 9.4.0** 该日志文件的格式为 mqxr\_n.json。

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.stdeerr
```

## 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 6: Windows 上的遥测配置目录

```
/var/mqm/qmgrs/qMgrName/mqxr
```

图 7: 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 160 页的图 6](#) 和 [第 160 页的图 7](#)。

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

从 IBM MQ 9.3.0 起，JAAS.Login 模块文件已从 `&MQ_INSTALL_DIRECTORY&/mqxr/samples/samples` 目录移动到 `&MQ_INSTALL_DIRECTORY&/mqxr/samples/jaas` 目录

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
```

使用以下目录中每个队列管理器的 `service.env` 文件覆盖这些设置：

```
WMQ data directory\Qmgrs\qMgrName\service.env
```

```
CLASSPATH= WMQ Installation Directory\mqxr\samples\jaas
```

图 8: `service.env` for Windows 样本

注：`service.env` 不得包含任何变量。请替换 `WMQ Installation Directory` 的实际值。

## 跟踪

请参阅 [第 162 页的『跟踪遥测 \(MQXR\) 服务』](#)。用于配置跟踪的参数存储在以下文件中：

```
WMQ data directory\Qmgrs\qMgrName\mqxr\trace.config  
WMQ data directory\Qmgrs\qMgrName\mqxr\mqxrtraceOn.properties  
WMQ data directory\Qmgrs\qMgrName\mqxr\mqxrtraceOff.properties
```

## com.ibm.mq.MQXR.GenerateJavaDump 属性

在某些情况下，MQXR 服务不必在“首次故障支持技术”(FFST) 生成过程中生成 Java 转储。

您可以将系统属性 `com.ibm.mq.MQXR.GenerateJavaDump` 设置为指定是否生成 Java 核心转储：

- 将此属性设置为 `true` 表示始终与 FDC 一起生成 Java 核心转储。
- 将此属性设置为 `false` 表示从不与 FDC 一起生成 Java 核心转储。

缺省情况下，`com.ibm.mq.MQXR.GenerateJavaDump` 属性处于禁用状态。此属性位于 `mqinstall/qmgrs/qmgr_name/mqxr/java.properties` 以及 IBM MQ 安装中的 `mqinstall/qmgrs/qmgr_name/amqp/amqp_java.properties` 中。需要将这两个属性都设置为相同的值。

## 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 (mqtvt3app.jar) 和关联的客户机库 (mqtvt3.jar)。它们已在以下目录中提供：
    - ma9b/SDK/clients/java/org.eclipse.paho.sample.mqtvt3app.jar
    - ma9b/SDK/clients/java/org.eclipse.paho.client.mqtvt3.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 (mqtvt3app.jar) 的源。它位于以下目录中：

```
ma9c/clients/java/samples/org/eclipse/paho/sample/mqtvt3app/*.java
```

- 仍然提供了已编译的客户机库。它位于以下目录中：

```
ma9c/clients/java/org.eclipse.paho.client.mqtvt3-1.0.2.jar
```

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)。

您还可以为所选感兴趣区域生成跟踪信息。

- 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。

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：

• 

```
controlMQXRChannel -qmgr=qmgr_name -mode=starttrace [clientid=ClientIdentifier]
```

•  

```
./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：

• 

```
controlMQXRChannel -qmgr=qmgr_name -mode=stoptrace
```

•  

```
./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：

•  *MQ\_DATA\_PATH*\trace.

•   /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。

跟踪文件的名称如下所示：

- 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](#)

## 为 MQTT 启用 JSON 格式的日志

要启用 JSON 格式错误日志记录，需要修改 MQTT 配置文件 *mqxrtrace0n.properties* 和 *mqxrtrace0ff.properties*。



## 关于此任务

您可以将 JSON 格式日志记录配置为与基于文本的错误日志记录一起独占或同时使用。

注: 修改配置文件后, 需要重新启动 MQTT 服务以使任何更改生效。

## 过程

1. 要以独占方式启用 JSON 格式日志记录, 请修改 `mqxrtraceOn.properties` 和 `mqxrtraceOff.properties` 配置文件, 并按如下所示更新 **handlers** 属性:

```
handlers= com.ibm.mq.util.logging.MQJSONErrorLogFileHandler
```

2. 要同时启用 JSON 格式日志记录以及基于文本的错误日志记录, 请修改 `mqxrtraceOn.properties` 和 `mqxrtraceOff.properties` 配置文件并更新 **handlers** 属性, 如下所示:

```
handlers= com.ibm.mq.util.logging.MQErrorLogFileHandler,  
com.ibm.mq.util.logging.MQJSONErrorLogFileHandler
```

## 结果

在任一情况下, 这都会启用 JSON 格式错误日志记录, 并使用以下缺省属性配置记录器:

- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.level`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.filter`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.limit`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.count`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.append`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.permissions`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.formatter`
- `com.ibm.mq.util.logging.MQJSONErrorLogFileHandler.pattern`

在功能上, 这些属性与 `com.ibm.mq.util.logging.MQErrorLogFileHandler` 配置的基于文本的错误记录器工作方式相同。

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 MQTT 客户机编写了自己的 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
<b>REASON_CODE_INVALID_PROTOCOL_VERSION</b>	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。
<b>REASON_CODE_INVALID_CLIENT_ID</b>	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, '._/%
<b>REASON_CODE_SERVER_CONNECT_ERROR</b>	请检查遥测 (MQXR) 服务和队列管理器是否正常运行。使用 <b>netstat</b> 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
<b>REASON_CODE_CLIENT_EXCEPTION</b> <b>REASON_CODE_CLIENT_TIMEOUT</b>	在服务器中查找 FDC 文件；请参阅第 160 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。

使用一个工具（例如，可从 <https://code.google.com/archive/p/tcpmon/> 获取的 **tcpmon**）来检查 TCP/IP 请求

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在客户机工作目录中创建名为 *clientIdentifier-tcpHostNameport* 或 *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 162 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) 服务会将它检测到的错误写入 *mqxr\_n.log*，而队列管理器则会将错误写入 *AMQERR01.LOG*。

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，以获取更多信息。

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 TCP/IP 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 160 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 162 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 160 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 160 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\jaas\LoginModule.java* 以创建自己的认证类 *WMQ installation directory\mqxr\samples\jaas\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 170 页的『示例 JAAS 配置』这一示例是一个样本配置。

从 IBM MQ 9.3.0 起，`JAAS.Login` 模块文件已从 `&MQ_INSTALL_DIRECTORY&/ amqp/samples/samples` 目录移动到 `&MQ_INSTALL_DIRECTORY&/ amqp/samples/jaas` 目录

## 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 160 页的『服务器端日志』，有关日志中列出的异常示例，请参阅第 172 页的图 14。
2. 通过将您的 JAAS 配置与第 170 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 9: *WMQ Installation directory\data\qmgrs\qMgrName\mqxr\jaas.config*

当 `SYSTEM.MQTT.SERVICE` 启动时，它会将第 170 页的图 10 中的路径添加至其类路径。

```
CLASSPATH=C:\WMQTelemetryApps;
```

图 10: *WMQ Installation directory\data\qmgrs\qMgrName\service.env*

第 171 页的图 11 显示了第 170 页的图 10 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 11: *runMQXRService.bat* 的类路径输出

第 171 页的图 12 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/JAASCAUser 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 12: *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 171 页的图 13。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 13: 连接到 *Eclipse Paho* 样本时抛出异常

`mqxr.log` 包含第 171 页的图 13 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 172 页的图 14 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 14: 装入 JAAS 配置时出错

## 多点广播问题故障诊断

使用此信息可帮助您检测和处理 IBM MQ 上的多点广播问题。

### 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
```

收件人：

```
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：

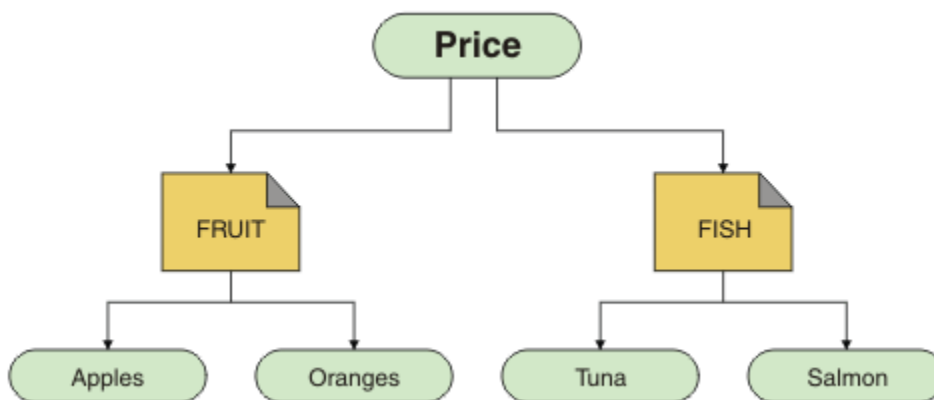
```
DEF COMMINFO(MC1) GRPADDR(
227.20.133.1)

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

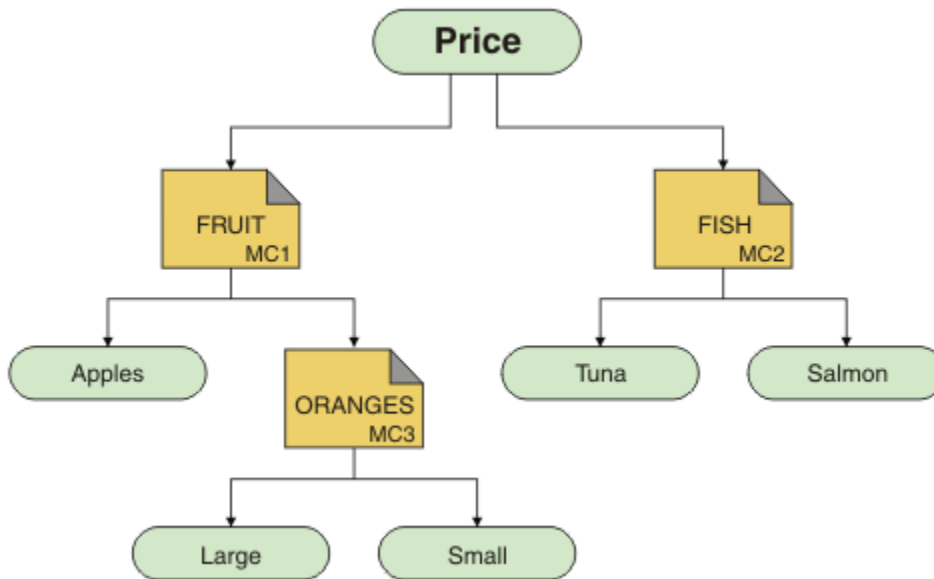
此示例通过添加名为 ORANGES 的另一个主题对象（定义为使用另一个 COMMINFO 主题定义 (MC3)）来扩展前一个示例：

```
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。

## 日志 I/O 操作超过阈值消息

V 9.4.0

- **场景:**您在队列管理器错误日志中接收到 `AMQ6729W` 日志 I/O 操作超过阈值。
- **说明:**IBM MQ 检测到日志读写或输入和输出操作所花费的时间超出预期。这可能是由于操作系统或存储系统存在问题，并可能影响队列管理器性能。
- **解决方案:**使用 `AMQ_IODELAY`，`AMQ_IODELAY_INMS` 和 `AMQ_IODELAY_FFST` 环境变量来帮助诊断存储性能问题，或者增加对此类延迟的容错。有关更多信息，请参阅 `AMQ_IODELAY`，`AMQ_IODELAY_INMS` 和 `AMQ_IODELAY_FFST`。

## 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 66 页的『[针对发布/预订集群的路由：行为注释](#)』。

注：此任务中的步骤要求您运行 MQSC 命令。如何执行此操作因平台而异。请参阅 [使用 MQSC 命令管理 IBM MQ](#)。

### 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 MQSC 命令显示哪些通道正在运行：

```
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。可以使用以下 MQSC 命令重新启动已停止的通道：

```
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。如果自动定义的集群通道是手动停止的，那么队列管理器会记住它是手动停止的，并且不会在将来自动启动它。如果需要停止通道，请记住在方便的时候再次将其重新启动，否则请发出以下 MQSC 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）。

**要点：**如果通道由于已耗尽重试而具有 STOPPED，那么重新启动队列管理器不会使通道脱离其 STOPPED 状态。

- 确认集群中的所有队列管理器均了解所有的完整存储库。

您可以使用以下 MQSC 命令执行此操作：

```
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。

要查看哪些其他队列管理器正在与集群接收方通道进行通信，请使用以下 MQSC 命令：

```
display chstaus(*) 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。

您可以使用以下 MQSC 命令执行此操作：

```
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 MQSC 命令检查传输队列：

```
display ql(*) curdepth where (usage eq xmitq)
```

您可以使用以下 MQSC 命令来检查系统队列：

```
display ql(system*) curdepth
```

## 相关任务

第 296 页的『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。

配置队列管理器集群

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 43 页的『Making initial checks on z/OS』

Before you start problem determination in detail on z/OS, consider whether there is an obvious cause of the problem, or an area of investigation that is likely to give useful results. This approach to diagnosis can often save a lot of work by highlighting a simple error, or by narrowing down the range of possibilities.

第 17 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

### Multi 应用程序均衡故障诊断

使用此部分可帮助您对应用程序均衡问题进行故障诊断

### Multi 应用程序未正确均衡

可以通过各种方式使用 DISPLAY APSTATUS 命令来诊断与应用程序均衡相关的许多症状。

## 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)，其中(可动等于 NO)并调查 IMM 臧 N 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)

此显示命令可用于诊断可能导致应用程序无法按预期进行重新平衡的许多问题。首先，检查 IMMDATE 和 IMMTIME 字段以查看应用程序是否仅临时标记为不可移动。

至于申请未能再获平衡的其他原因，则由 IMMREN 指出。下表显示了所需的各种原因 (IMM 赋) 和操作。请注意，在大多数情况下，需要与相关应用程序开发者或所有者一起复审这些原因

伊姆雷恩	ACTION
NOTCLIENT	应用程序正在使用服务器绑定，因此无法将其移至另一个队列管理器。在大多数情况下，可以修改应用程序以使用客户机连接。这可能需要重新构建应用程序，但这取决于正在使用的语言和库版本。
NOTRECONN	应用程序连接未标记为 "可重新连接"。这可能是应用程序代码中的故意决策，因为其设计要求所有消息都流向单个队列管理器，或者可能指示配置错误或监管 (例如，非常旧的客户机库不支持重新连接)。  请注意，对于要工作的应用程序均衡，RECONNECT_QMGR 是不够的，因为这指示仅允许重新连接到 "同一" 队列管理器实例。要查看应用程序实例中正在使用的连接选项，可以发出 <b>DIS CONN (*) TYPE (CONN) WHERE (CONNTAG eq 'xxx') CONNOPTS</b> ，其中 xxx 是 DIS APSTATUS 输出中的 CONNTAG。
APPNAMECHG	应用程序正在同一 TCP 连接上建立多个连接，但具有不同的应用程序名称。这意味着无法可靠地分隔应用程序实例，因此无法进行重新平衡。如果发生此问题，那么应用程序代码可能显式覆盖 MQCONN 调用中的应用程序名称。
正在移动	这应该是临时状态，因为它指示已识别应用程序实例以进行重新平衡。
INTRANS	应用程序当前在事务中，因此重新平衡可避免中断 (回滚)。如果应用程序开发者或部署者不关心此应用程序的过多回滚，并且希望优先维护应用程序连接的一致平衡，那么可以在应用程序代码或配置设置中忽略此约束；请参阅 <b>BalanceOptions</b> 以获取更多信息。  或者，可以使用 <b>Timeout</b> 字段修改队列管理器允许事务在考虑中断之前继续的时间长度。
REPLY	此应用程序已标记为类型 "请求应答"，并且正在等待对先前分派的请求消息的响应。如果您不希望等待响应，那么标记为 "SIMPLE" 类型将阻止此等待。  或者，您可以使用应用程序请求消息的消息到期或 <b>Timeout</b> 来配置等待时间段的扩展数据块。请注意，通常最好同时进行相应的配置，以便 <b>Timeout</b> 不会意外地截断响应的等待时间。

## 相关参考

[显示状态](#)

## Multi 应用程序持续连接或断开连接

与应用程序关联的症状和解决方案持续连接或断开连接，或者无法达到预期的平衡。

首先，使用第 177 页的『应用程序未正确均衡』中描述的 DIS APSTATUS 命令排除常见原因

### 症状

所有应用程序实例看起来都是可移动的，但实例在不断地重新平衡并且未能达到均衡，并且/或者某些队列管理器没有任何应用程序实例。

可能的原因是未正确配置客户机通道定义表 (CCDT)。统一集群要求客户机使用 CCDT 进行连接，该 CCDT 包含集群中每个单独队列管理器的连接信息。当要求应用程序实例重新连接到另一个队列管理器 (以重新平衡连接) 时，IBM MQ 客户机代码会咨询 CCDT 以建立到新目标的路由。

如果某些队列管理器未包含在 CCDT 中，或者其配置信息中存在错误，那么当应用程序实例尝试重新连接时，它将在移动过程中失败 (通常重新连接回其先前的队列管理器)。在稍后的某个时候，可能会要求它再次尝试重新连接，结果相同。这会导致连接频繁 "跳跃"，而应用程序作为一个整体从未在统一集群中实现均匀分布。

### 解决方案

请确保在提供给客户机应用程序的 CCDT 中始终表示统一集群的所有成员。这应该包括临时不活动的成员，因为不会请求应用程序重新平衡到当前未运行的队列管理器。如果识别 CCDT 中的错误，那么在进行任何更正后，请使用现有机制将文件重新部署到所有客户机。不需要重新启动客户机应用程序，这将在下次尝试重新连接时找到修改后的条目。

如果 CCDT 项显示正确，那么集群的某些成员上的侦听器或 SRVCONN 通道定义可能存在问题，这会导致类似的行为，原因相同。

## 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 **qlocal**。
- 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。

#### **A2001 (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 **CLUSSDR** 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))
CLUSQMGR(QM2) CHANNEL(DEMO.QM2)
```

z/OS

```
CSQM201I +CSQ2 CSQMDRTC DISPLAY CLUSQMGR DETAILS
CLUSQMGR(SYSTEM.TEMPQMGR.<HOSTNAME>(1716))
CLUSQMGR(QM1) 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](#)。

#### 相关参考

[2035 \(07F3\) \(RC2035\): MQRC\\_NOT\\_AUTHORIZED](#)

## 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, 那么您也可能看到此问题。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\\_UNKNOWN\\_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\_UNKNOWN\_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 MQOO\_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 MQOO\_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 MQOO\_BIND\_ON\_OPEN 解析的目标队列管理器的名称。如果别名队列使用 MQOO\_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，发出命令 **REFRESH CLUSTER(DEMO)**。

**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。未完成的等待消息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。

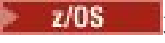

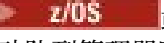
## 存储库发生故障时将发生的情况

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  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。
  -  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  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。

## 原因 3

新的传输队列或托管它的存储介质已填充。

## 解决方案

检查队列和通道状态以确认是否需要管理操作，例如：

- 启动通道以开始移动消息
- 完整远程 (目标) 队列上的可用空间 (如果这导致通道备份)
- 增大传输队列上的 MAXDEPTH 属性

切换过程会连续重试，并在问题解决后完成。

## 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 `runswchl` 和 `-n` 参数 (或在 z/OS 上具有 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。

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。

### 关于此任务

另请参阅用于说明 `rdqmstatus` 命令输出的主题，以帮助进行故障诊断 ([查看 RDQM 和 HA 组状态](#)、[查看 DR RDQM 状态](#)和[查看 DR/HA RDQM 和 HA 组状态](#))。

## 相关任务

第 321 页的『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。

## MQ Adv. Linux RDQM 体系结构

描述复制的数据队列管理器高可用性和灾难恢复 (RDQM HA 和 RDQM DR) 配置的基本体系结构，以帮助进行故障诊断。

- [第 195 页的『资源名称』](#)
- [第 195 页的『HA 体系结构』](#)
- [第 195 页的『DRBD 资源』](#)
- [第 196 页的『DRBD HA 连接』](#)
- [第 197 页的『DRBD keepalive』](#)
- [第 197 页的『Pacemaker』](#)
- [第 198 页的『DR 体系结构』](#)
- [第 198 页的『DRDB DR 连接』](#)

## 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。该名称可在 `/var/mqm/mqs.ini` 文件中找到，这里称为 *qm*。例如，对于名为 TMPQM1 的 RDQM HA 队列管理器，*qm* 将为 tmpqm1。

## HA 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）。请参阅 [https://linbit.com/drbd-user-guide/drbd-guide-9\\_0-en/](https://linbit.com/drbd-user-guide/drbd-guide-9_0-en/) 以获取有关 DRBD 的信息，并参阅 <https://clusterlabs.org/pacemaker/> 以获取有关 Pacemaker 的信息。

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 资源

每个 RDQM HA 队列管理器都有一个为其生成的 DRBD 资源文件，名为 `/etc/drbd.d/qm.res`。例如，当创建名为 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 mqhavam13 kernel: drbd haqm1/0 drbd100 mqhavam15.gamsworthwilliam.com: drbd_sync_handshake:
```

如果存在字符串 `drbd100`，那么表示消息与 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 mqhavam13 kernel: drbd haqm1 mqhavam15.gamsworthwilliam.com: Connection closed
```

## DRBD HA 连接

在正常操作中，每对节点之间都有一对 DRBD 连接：

- 控件或元连接。
- 主数据连接。

通常的模式是，这些连接中的一个由一对中的一个节点建立，另一个由该对中的另一个节点建立。

对于 RDQM HA，在创建队列管理器时，会自动选择队列管理器的 DRBD 侦听器端口。端口号从 7000 开始。您可以通过在相应的 DRBD .res 文件中查找地址行找到队列管理器的端口，例如：

```
address ipv4 10.51.7.58:7000;
```

当您知道相应的端口号时，可以使用 **ss** 命令来查找与该队列管理器相关的 TCP 连接，例如：

```
ss -nt '( src :7000 or dst :7000 )'
```

示例 RDQM HA 配置由以下节点组成：

- node-1, IP 地址 10.51.6.190
- node-2, IP 地址 10.51.6.210
- node-3, IP 地址 10.51.7.19

已配置并运行单个 RDQM HA 队列管理器，并且已连接所有 DRBD 连接。在 node-3 上运行的 **ss** 命令提供以下输出：

State	Recv-Q	Send-Q	Local Address:Port	Peer
Address:Port	Process			
ESTAB	0	0	10.51.7.19:36111	
10.51.6.210:7000				
ESTAB	0	0	10.51.7.19:7000	
10.51.6.210:50901				
ESTAB	0	0	10.51.7.19:7000	
10.51.6.190:42379				
ESTAB	0	0	10.51.7.19:54205	
10.51.6.190:7000				

列出的第一个连接由 node-3 建立到 node-2。

列出的第二个连接由 node-2 建立到 node-3。如上所述，一对节点之间的两个 DRBD 连接通常是在相反方向上建立的。

列出的第三个连接由 node-1 建立到 node-3。

列出的第四个连接由 node-3 建立到 node-1。

将控制连接与活动队列管理器的数据连接区分开来的最简单方法：通过该连接上流动的数据量（通常对于控制连接而言较小）。

要查看流经每个连接的字节数，请使用带有 **i** 选项的 **ss** 命令。在此示例中，运行以下命令：

```
ss -nit '( src :7000 or dst :7000 )'
```

产生下列输出：

State	Recv-Q	Send-Q	Local	Peer
Address:Port	Process	Peer Address:Port	Address:Port	Process
ESTAB	0	0	10.51.7.19:36111	10.51.6.210:7000
cubic wscale:7,7 rto:201 rtt:0.209/0.085 ato:40 mss:1448 pmtu:1500 rcvmss:1448 advmss:1448				
cwnd:10 ssthresh:27 bytes_sent:2587329 bytes_retrans:20 bytes_acked:2587310				
bytes_received:3704817 segs_out:2980 segs_in:1566 data_segs_out:2046 data_segs_in:932 send				
554258373bps lastsnd:229936217 lastrcv:3102096 lastack:3102101 pacing_rate 1106531336bps				
delivery_rate 1672556144bps delivered:2047 busy:131ms retrans:0/1 dsack_dups:1 rcv_rtt:7283.92				
rcv_space:277568 rcv_ssthresh:2662500 minrtt:0.128				
ESTAB	0	0	10.51.7.19:7000	10.51.6.210:50901
cubic wscale:7,7 rto:201 rtt:0.3/0.027 ato:40 mss:1448 pmtu:1500 rcvmss:536 advmss:1448 cwnd:10				
bytes_sent:401848 bytes_acked:401848 bytes_retrans:388128 segs_out:23848 segs_in:46420				
data_segs_out:23343 data_segs_in:23086 send 386133333bps lastsnd:9624 lastrcv:9624 lastack:9624				

```

pacing_rate 770340808bps delivery_rate 176406088bps delivered:23344 app_limited busy:6597ms
rcv_rtt:68387.1 rcv_space:28976 rcv_ssthresh:64336 minrtt:0.097
ESTAB      0          0
10.51.7.19:7000          10.51.6.190:42379
cubic wscale:7,7 rto:201 rtt:0.538/0.059 ato:40 mss:1448 pmtu:1500 rcvmss:536 advmss:1448
cwnd:10 bytes_sent:885176 bytes_acked:885176 bytes_received:415992 segs_out:53022 segs_in:27238
data_segs_out:29765 data_segs_in:23599 send 215315985bps lastsnd:8025 lastrcv:8025 lastack:8025
pacing_rate 430531936bps delivery_rate 132237440bps delivered:29766 app_limited busy:7935ms
rcv_rtt:299060 rcv_space:28960 rcv_ssthresh:68624 minrtt:0.116
ESTAB      0          0
10.51.7.19:54205        10.51.6.190:7000
cubic wscale:7,7 rto:201 rtt:0.245/0.079 ato:40 mss:1448 pmtu:1500 rcvmss:1448 advmss:1448
cwnd:10 ssthresh:28 bytes_sent:2947194 bytes_retrns:20 bytes_acked:2947175
bytes_received:3170253590 segs_out:67287 segs_in:78314 data_segs_out:3556 data_segs_in:76191
send 472816327bps lastsnd:229936217 lastrcv:229936217 lastack:2573657 pacing_rate 942267408bps
delivery_rate 875365232bps delivered:3557 app_limited busy:1187ms retrns:0/1 dsack_dups:1
rcv_rtt:17.234 rcv_space:3043672 rcv_ssthresh:3145728 minrtt:0.078

```

对于每个连接，在输出的第二行中都有一些值，您可以使用这些值决定哪个是控制连接，哪个是数据连接：

- bytes\_sent (仅在 RHEL 8 和 RHEL 9 系统上显示)
- bytes\_acked
- bytes\_received

如果队列管理器处于空闲状态，那么由于每 10 秒发送一次 DRBD ping 以及对它的响应，因此与较大数据流量的连接可能实际上是控制连接。如果您有空闲队列管理器，那么标识控制连接的最可靠方法是使用命令 (例如 **tcpdump src port 36111**) 在其中一个连接上运行 **tcpdump**。如果与指定本地端口的连接是控制连接，那么您应该每 10 秒看到一次活动。

如果没有来自 **tcpdump** 的输出，那么该连接几乎肯定是数据连接。您可以通过将文件复制到队列管理器的 **userdata** 目录验证这一点，这将导致通过数据连接发送数据。

## DRBD keepalive

DRBD 有自己的 keepalive 实现，在控制连接上使用。数据连接依赖于标准的 Linux keepalive 功能部件。

RDQM 使用以下缺省 DRBD 配置：

```

ping-int      10; # seconds, default
ping-timeout  5; # 1/10 seconds, default

```

此配置意味着 DRBD 每 10 秒发送一次 ping 请求，并期望在半秒内得到响应。如果在该时间内未接收到响应，那么您将在系统日志中看到一条消息，指出 PingAck 未及时到达。

## 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**

## DR 体系结构

RDQM DR 的体系结构更简单，因为不涉及 Pacemaker，仅涉及 DRBD。

RDQM DR/HA 的体系结构是 DR 的体系结构与 HA 的体系结构的组合。

## DRDB DR 连接

DR/HA 队列管理器的 DRBD DR 连接位于队列管理器运行所在的节点与当前为 DR 辅助项的 RDQM HA 组中作为 HA 首选位置的节点之间。对于每个队列管理器，HA 组之间仅存在一对连接。DR 辅助 HA 组中的节点负责将其接收到的更新同步复制到 HA 组中的其他两个节点。

DR 复制被描述为异步，但它并非完全异步。异步复制映射到 DRBD 协议 A。如果建立了连接，那么在将数据复制到连接的本地 TCP sendbuffer 后，DRBD 会立即将更新视为已完成。如果未建立连接，那么将记住更新并立即完成更新，或者在 DR/HA 的情况下，在其他 HA 节点上完成更新后立即完成更新。

以下示例 DR/HA 配置包含两个 HA 组，每个组使用不同的 HA 和 DR IP 地址。第一个 HA 组具有以下节点：

节点	HA IP 地址	DR IP 地址
node-a1	10.51.20.185	192.168.198.50
node-a2	10.51.20.192	192.168.198.66
node-a3	10.51.20.203	192.168.199.236

第二个 HA 组具有以下节点：

节点	HA IP 地址	DR IP 地址
node-b1	10.51.7.58	192.168.208.25
node-b2	10.51.7.190	192.168.208.26
node-b3	10.51.15.65	192.168.208.28

队列管理器 QM1 正在 node-a1。The `rdqmstatus -m QM1` 命令上运行将生成以下输出：

```
Node:
node-a1
Queue manager status:      Running
CPU:                       0.00%
Memory:                    181MB
Queue manager file system: 58MB used, 2.9GB allocated [2%]
HA role:                   Primary
HA status:                 Normal
HA control:                Enabled
HA current location:       This node
HA preferred location:     This node
HA blocked location:       None
HA floating IP interface:  None
HA floating IP address:    None
DR role:                   Primary
DR status:                 Normal
DR port:                   8001
DR local IP address:       192.168.198.50
DR remote IP address list: 192.168.208.25,192.168.208.26,192.168.208.28
DR current remote IP address: 192.168.208.25

Node:
node-a2
HA status:                 Normal

Node:
node-a3
HA status:                 Normal
```

DR current remote IP address 的值是 DR 辅助实例当前从 DR 主项接收数据的节点的 DR IP 地址，在本例中为 node-b1。在 node-b1 上运行 **rdqmstatus -m QM1** 会产生以下输出：

```
Node:
node-b1
Queue manager status:      Ended immediately
HA role:                   Primary
HA status:                 Normal
HA control:                Enabled
HA current location:       This node
HA preferred location:     This node
HA blocked location:       None
HA floating IP interface:  None
HA floating IP address:    None
DR role:                   Secondary
DR status:                 Normal
DR port:                   8001
DR local IP address:       192.168.208.25
DR remote IP address list: 192.168.198.50,192.168.198.66,192.168.199.236
DR current remote IP address: 192.168.198.50

Node:
node-b2
HA status:                 Normal

Node:
node-b3
HA status:                 Normal
```

DR port 的值是用于此特定队列管理器的 DR 复制的端口号，如原始 **crtmqm** 命令中所指定。

在此示例中，以下连接在 node-a1: 上使用此端口

```
ss -nt '( src :8001 or dst :8001 )'
State      Recv-Q  Send-Q          Local Address:Port
Peer Address:Port
ESTAB      0        0              192.168.198.50:8001
192.168.208.25:57373
ESTAB      0        0              192.168.198.50:59630
192.168.208.25:8001
```

与 HA 连接一样，DR 连接通常是在相反方向建立的。在这里，第一个连接由 node-b1 建立到 node-a1，第二个连接由 node-a1 建立到 node-b1。其中一个连接是控制连接或元连接，另一个连接是数据连接。以下示例包含有关每个连接的信息以查看接收到的字节数：

```
ss -int '( src :8001 or dst :8001 )'
State      Recv-Q  Send-Q          Local Address:Port
Peer Address:Port
ESTAB      0        0              9.20.198.50:8001
9.20.208.25:57373
    cubic wscale:7,7 rto:202 rtt:1.648/2.188 ato:40 mss:1448 rcvmss:592 advmss:1448 cwnd:10
bytes_acked:733 bytes_received:1181 segs_out:14 segs_in:16 send 70.3Mbps lastsnd:4841379
lastrcv:4841380 lastack:4841373 pacing_rate 140.6Mbps rcv_rtt:6 rcv_space:28960
ESTAB      0        0              9.20.198.50:59630
9.20.208.25:8001
    cubic wscale:7,7 rto:201 rtt:0.61/0.044 ato:40 mss:1448 rcvmss:536 advmss:1448 cwnd:10
bytes_acked:7785 bytes_received:7728 segs_out:970 segs_in:487 send 189.9Mbps lastsnd:1397
lastrcv:1397 lastack:1397 pacing_rate 379.4Mbps rcv_space:29200
```

第一个连接的 bytes\_received 值为 1181，第二个连接的 bytes\_received 值为 7728，这表示第一个连接是控制连接，第二个连接是数据连接，尽管这些值足够相似，可能是由于 DRBD ping 流量而导致最近空闲的数据连接超过了控制连接的实际情况。

## MQ Adv. Linux 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: 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: mqhavam15.gamsworthwilliam.com
HA status: Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.

[midtownjojo@mqhavam14 ~]$ 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 连接都已丢失。

在此情况下，其他两个节点都具有 DRBD 资源的 DRBD 定额，因此 RDQM 正在其他位置运行，在 mqhavam14.gamsworthwilliam.com 上显示为 HA current location 的值。

## 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:                 0KB

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。

查看此问题的一种方法是使用 **drbdadm status** 命令。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
mqhavam15.gamsworthwilliam.com role:Secondary
replication:SyncSource peer-disk:Inconsistent done:90.91

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
```

在此情况下，由于 vm15 上的磁盘为 Inconsistent，因此 RDQM HA 队列管理器 HAQM1 无法移动到 vm15。

done 值是完成的百分比。如果该值未增大，那么您可以尝试断开该副本的连接，然后使用 vm13: 上的以下命令（作为 root 运行）再次连接该副本：

```
drbdadm disconnect haqm1:mqhavam15.gamsworthwilliam.com
drbdadm connect haqm1:mqhavam15.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：

- Corosync 主进程未调度
- RDQM HA 队列管理器未在期望的位置运行

### Pacemaker 方案 1: Corosync 未调度主进程

如果您在 syslog 中看到类似于以下示例的消息，那么表示系统太忙，无法将 CPU 时间安排到主 Corosync 进程，或者更常见的情况是，系统是虚拟机，并且系统管理器未将任何 CPU 时间安排到整个 VM。

```
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 都具有用于检测定额丢失情况的计时器，因此类似以下示例的消息指示节点未运行时间过长，因此会从定额中删除该节点。Corosync 超时为 1.65 秒，1.32 秒的阈值为该值的 80%，因此当主 Corosync 进程的调度延迟达到该超时的 80% 时，将显示此示例中显示的消息。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 `rdqmstatus` 命令。以下示例显示了一切正常时针对该配置给出的响应。在 VM13 上运行这些命令:

```
%rdqmstatus -m HAQM1

Node:                               mqhavam13.gamsworthwilliam.com
Queue manager status:               Running
CPU:                                0.00
Memory:                             123MB
Queue manager file system:         606MB used, 1.0GB allocated [60%]
HA role:                            Primary
HA status:                          Normal
HA control:                         Enabled
HA current location:                This node
HA preferred location:              This node
HA preferred location:              This node
HA blocked location:                None
HA floating IP interface:           eth4
HA floating IP address:             192.0.2.4

%rdqmstatus -m HAQM2

Node:                               mqhavam13.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 blocked location:                None
HA floating IP interface:           eth4
HA floating IP address:             192.0.2.6

%rdqmstatus -m HAQM3

Node:                               mqhavam13.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 blocked location:                None
HA floating IP interface:           eth4
HA floating IP address:             192.0.2.8
```

请注意以下几点:

- 这三个节点均显示 HA 状态为 `Normal`。
- 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` 对于队列管理器 HAQM1 的目录具有无效的值。

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:

```
$ 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:      mqhavm14.gamsworthwilliam.com
HA blocked 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

```

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。检查状态并包含失败的资源操作时将提供以下响应：

```

$ rdqmstatus -m HAQM1 -a

Node:                       mqhavm13.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:     mqhavm14.gamsworthwilliam.com
HA blocked 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

Failed resource action:    Start
Resource type:             Queue manager
Failure node:              mqhavm14.gamsworthwilliam.com
Failure time:              2022-01-01 12:00:00
Failure reason:            Generic error
Blocked location:         mqhavm14.gamsworthwilliam.com

```

请记录已出现的 Failed resource action 部分。

该条目表明，当 Pacemaker 尝试检查 vm14 上 HAQM1 的状态时，由于未配置 HAQM1，发生了错误，这是由于 /var/mqm/mqs.ini 中故意的错误配置导致的。

## 纠正该故障

要纠正该故障，您必须纠正根本问题（在本例中，为 vm14 上 /var/mqm/mqs.ini 中的 HAQM1 恢复正确的目录值）。然后，必须在相应的资源上使用命令 **rdqmclean** 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：

```
$ rdqmclean -m HAQM1
```

然后，再次检查失败的资源操作状态：

```
$ rdqmstatus -m HAQM1 -a
```

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：

```

$ 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 blocked location:       None
HA floating IP interface:   None
HA floating IP address:    None

```

```

Node: mqhavam14.gamsworthwilliam.com
HA status: Normal

Node: mqhavam15.gamsworthwilliam.com
HA status: Normal

```

## MQ Adv. Linux 升级 RDQM 后的问题

如果在升级 RDQM 配置中的一个或多个节点后遇到问题，那么应该检查所安装的 DRBD 内核是否与操作系统内核版本匹配。

RDQM 配置依赖于 DRBD 模块来提供数据复制。升级 RDQM 时，请务必确保为系统上运行的 RHEL 内核版本安装了正确版本的 DRBD 内核模块。如果版本不匹配，那么您可能会遇到不同严重性的问题。可能是队列管理器无法启动，或者它无法在已升级的节点上运行，即使该节点是首选位置。

可以使用 `rdqmstatus` 命令来查看有关已安装版本的信息，请参阅[查看 RDQM 和 HA 组状态](#)、[查看 DR RDQM 状态](#)和[查看 DR/HA RDQM 和 HA 组状态](#)。

有关选择正确的 DRBD 模块进行安装的指导信息，请参阅[安装 RDQM（复制的数据队列管理器）](#)。

以下示例说明了一些潜在的不匹配情况及其影响。

### 示例 1

```

Node: mqhavam57.exampleco.com
OS kernel version: 5.14.0-362.18.1
DRBD OS kernel version: 5.14.0-70.13.1
DRBD version: 9.2.7+ptf.14
DRBD kernel module status: Partially loaded

Queue manager name: QM1
Queue manager status: Running elsewhere
HA status: Unknown
HA current location: mqhavam58.exampleco.com
HA preferred location: This node
HA blocked location: None
DR role: Primary
DR status: Unknown

Queue manager name: QM2
Queue manager status: Running elsewhere
HA status: Unknown
HA current location: mqhavam58.exampleco.com
HA preferred location: This node
HA blocked location: None

Queue manager name: QM3
Queue manager status: Status not available
DR role: Secondary
DR status: Unknown

```

此摘要状态显示已安装 RHEL 9.0 的 DRBD 内核模块，而操作系统内核版本适用于 RHEL 9.3。由于此不匹配情况，将仅装入 DRBD 模块的一部分。QM1 是 HA/DR 队列管理器，并移至另一个节点，其 HA 状态为 unknown，其 DR 状态为 unknown。QM2 是 HA 队列管理器，无法在其首选节点上启动，并且在 Unknown 中处于 HA 状态。QM3 是 DR 队列管理器，并且应该是主实例，但由于 DRBD 内核模块未完全装入，因此它报告为具有 DR 状态 Unknown 的辅助项。要解决这些故障，必须使用正在运行的操作系统内核的版本目标来更新 DRBD 内核模块。

### 示例 2

```

Node: mqhavam07.exampleco.com
OS kernel version: 5.14.0-362.18.1
DRBD OS kernel version: 5.14.0-284.11.1
DRBD version: 9.2.7+ptf.14
DRBD kernel module status: Loaded

Queue manager name: RDQM7
Queue manager status: Running
HA current location: This node

```

HA preferred location:	This node
HA blocked location:	None

此摘要状态显示已安装 RHEL 9.2 的 DRBD 内核模块，而操作系统内核版本适用于 RHEL 9.3。这不是严重的不匹配情况，将装入 DRBD 模块，并且队列管理器 RDQM7 会在此节点上运行。但是，建议您为已安装的 RHEL 安装正确的 DRBD 模块，以确保操作正确。

## 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。

### 相关任务

第 322 页的『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。

第 323 页的『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。

## 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遇到问题，请检查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？

通道提供给队列管理器的地址取决于正在使用的网络适配器。例如，如果用于访问侦听器的 **CONNAME** 是 localhost，那么将 127.0.0.1 作为您的地址；如果它是您计算机的实际 IP 地址，那么该地址是您提供给队列管理器的地址。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(\*) 和其他任何内容。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，如果要在其他平台上指定 CHLAUTH(\*)，那么必须在 z/OS 上指定 CHLAUTH('\*')。

### 队列管理器重新启动期间 SET CHLAUTH 命令的行为

如果是 SYSTEM.CHLAUTH.DATA.QUEUE 已被删除或改变，无法再访问它，即 PUT(DISABLED)，**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。

这意味着虽然 **SET CHLAUTH** 命令实施的规则最初可能可操作，但该命令的作用不会在队列管理器重新启动后持久存在。您应该进行调查，确保队列可访问，然后在循环队列管理器之前使用 ACTION (REPLACE) 重新发出命令。

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' 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'

## 相关信息

[SET CHLAUTH \(创建或修改通道认证记录\)](#)

## 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 AIX, Linux, and Windows 系统上的 TLS 客户机通道指定 DES\_SHA\_EXPORT1024 CipherSpec 并且 AIX, Linux, and 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，某些 TLS 实现 (例如 IBM Global Security Kit (GSKit)，DCM 和 RACF) 只要存在信任锚 (ROOT CA)，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) 证书的方法来控制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](#)

V 9.4.0

Linux

AIX

## 对认证令牌问题进行故障诊断

常见认证令牌问题的列表以及帮助您解决这些问题的信息。

问题列表根据最有可能从故障诊断建议中受益的用户进行划分。

## 针对管理员的建议

### 队列管理器是否配置为接受认证令牌？

V 9.4.0

必须将队列管理器配置为接受认证令牌。有关更多信息，请参阅 [配置队列管理器以使用 JWKS 端点接受认证令牌](#)。

### 队列管理器是否处于支持认证令牌的版本？

您可以将认证令牌与在 AIX 或 Linux 平台上运行 V IBM MQ 9.3.4 或更高版本的队列管理器配合使用。

### 更改令牌认证配置后是否已刷新安全性配置？

对令牌认证配置的更改将在刷新队列管理器的安全性配置或重新启动队列管理器时生效。要刷新队列管理器的安全配置，请发出以下 MQSC 命令：

```
REFRESH SECURITY TYPE(CONNAUTH)
```

### 队列管理器错误日志中是否有消息？

如果认证令牌被拒绝，请检查队列管理器错误日志中的消息。当认证令牌被拒绝时，可能会发出以下消息：

- [AMQ5783E](#)
- [AMQ5784E](#)
- [AMQ5785I](#)
- [AMQ5786E](#)
- [AMQ5788E](#)



这些消息可能包含描述错误原因的错误代码。有关更多信息，请参阅 [令牌认证错误代码](#)。

如果 `qm.ini` 文件的 **AuthToken** 节中的令牌认证配置无效，那么可能会发出以下消息：

- [AMQ7006](#)
- [AMQ7076](#)

#### 令牌签发者的证书还是密钥存储库中的对称密钥？

必须将令牌发布者的公用密钥证书或对称密钥导入到令牌认证密钥存储库中，以便队列管理器验证令牌签名。公用密钥证书不得到期。

#### 访问令牌认证密钥存储库时是否存在问题？

检查 `qm.ini` 文件的 **AuthToken** 节的 **KeyStore** 属性是否引用了正确的密钥存储库。密钥存储库必须是具有文件扩展名 `.kdb` 的 CMS 密钥存储库或具有文件扩展名 `.p12` 的 PKCS#12 密钥存储库。

有关更多信息，请参阅 [配置队列管理器以使用 JWKS 端点接受认证令牌](#)。

#### 密钥存储库密码是否存在问题？

如果迂到用于访问令牌认证密钥存储库的密码问题，请检查以下项：

- **V 9.4.0** 如果使用本地密钥库，加密密钥存储库密码存储在 `qm.ini` 文件的 **AuthToken** 节中引用的文件中，或者存储在隐藏文件中。
  - 如果指定了 `qm.ini` 文件的 **AuthToken** 节的 **KeyStorePwdFile** 属性，那么它必须引用包含加密密钥存储库密码的文件的正确路径。必须使用 `runmqcred` 命令对密钥存储库密码进行加密。密钥存储库密码文件必须仅包含作为单行文本的加密密码。如果在加密密码时指定了初始密钥，那么必须在队列管理器 **INITKEY** 属性中指定相同的初始密钥。有关更多信息，请参阅 [runmqcred \(保护认证令牌密钥库密码\)](#)。
  - 如果未使用 **AuthToken** 节的 **KeyStorePwdFile** 属性指定密钥存储库密码文件，那么必须存在密钥存储库隐藏文件。隐藏文件具有与密钥存储库相同的名称，具有 `.sth` 文件扩展名。隐藏文件由 `runmqakm` 命令创建。
- `mqm` 用户具有对密钥存储库文件，包含密钥存储库密码的文件以及包含密钥存储库的目录的读访问权。

#### 是否未采用令牌中的用户标识？

要在 IBM MQ 中采用令牌用户声明中的用户标识进行后续授权检查，必须配置以下内容：

- 将队列管理器的 **CONNAUTH** 属性所引用的认证信息对象的 **ADOPTCTX** 属性设置为 **YES**。
- 将 `qm.ini` 文件的 **AuthToken** 节的 **UserClaim** 属性设置为包含要采用的用户标识的令牌中声明的名称。

**V 9.4.0** 有关更多信息，请参阅 `qm.ini` 文件的 [AuthToken](#) 节或 [JWKS](#) 节。

#### `qm.ini` Service 节的 SecurityPolicy 属性是否设置为 group？

必须将队列管理器的 `qm.ini` 文件中的 **Service** 节的 **SecurityPolicy** 属性设置为缺省值 `UserExternal`。如果 **SecurityPolicy** 属性显式设置为 `Group`，那么将拒绝认证令牌配置。有关更多信息，请参阅 [SecurityPolicy](#)。

## 针对开发者的建议

#### 应用程序在连接到队列管理器时是否接收到错误代码？

如果应用程序尝试使用认证令牌连接到队列管理器时发生错误，那么会向应用程序返回描述错误的原因码。有关更多信息，请参阅以下原因码的描述：

- [2035 MQRC\\_NOT\\_AUTHORIZED](#)
- [2063 MQRC\\_SECURITY\\_ERROR](#)
- [2064 MQRC\\_TOKEN\\_TIMESTAMP\\_NOT\\_VALID](#)
- [2595 MQRC\\_CSP\\_ERROR](#)
- [2298 MQRC\\_FUNCTION\\_NOT\\_SUPPORTED](#)



### 客户机和队列管理器是否支持认证令牌?

您可以使用 IBM MQ 9.3.4 中的认证令牌。队列管理器必须在 AIX 或 Linux 平台上运行。应用程序必须以客户机方式 (而不是绑定方式) 进行连接的 IBM MQ MQI clients。

### 是否在 Java 应用程序中使用认证令牌?

您可以将认证令牌与 Java 应用程序配合使用, 但只能与使用包含令牌信息的 MQCSP 结构的安全出口配合使用。有关更多信息, 请参阅 [在应用程序中使用认证令牌](#) 和 [Java 类 MQCSP](#)。

确保应用程序对其访问的 IBM MQ 对象具有 INQ 权限。有关更多信息, 请参阅 [JMS 需要非特权用户授权的对象](#)。

### 是否使用 MQCSP 结构将认证令牌传递到队列管理器?

您可以将认证令牌与 IBM MQ MQI clients 配合使用, 这些令牌在 MQCONNX 调用上使用 MQCSO 结构。有关更多信息, 请参阅 [在应用程序中使用认证令牌](#)。

如果应用程序提供的 MQCSP 结构中的任何字段包含无效值, 那么将向应用程序返回原因码 MQRC\_CSP\_ERROR。可能还会将消息 AMQ960E 写入客户机错误日志, 以提供有关错误原因的更多信息。

**V 9.4.0** 对于使用 IBM MQ classes for JMS/Jakarta Messaging 的应用程序, 来自 com.ibm.mq.exits.MQCSP 类的等效消息将写入 IBM MQ classes for JMS 日志。有关更多信息, 请参阅第 375 页的『IBM MQ classes for JMS 中的错误日志』。

### 认证令牌的结构和内容是否正确?

认证令牌是具有三个 base64url 编码元素的结构。这些元素是头, 有效内容和特征符。每个元素由句点 (.) 分隔。解码时, 头和有效内容必须是有效的 JSON 对象。头和有效内容必须包含必需的头参数和声明。有关更多信息, 请参阅 [认证令牌需求](#)。

### 认证令牌是否包含有效的用户声明?

认证令牌可以包含用户声明, 该用户声明指定要用于授权检查以访问 IBM MQ 对象的用户标识。qm.ini 文件的 **AuthToken** 节的 **UserClaim** 属性必须指定包含要采用的用户标识的令牌声明的名称。例如, 如果令牌具有用户声明 "AppUser": "MyUserName", 那么必须在 qm.ini 文件的 **AuthToken** 节中指定 UserClaim=AppUser。有关更多信息, 请参阅 [认证令牌需求](#)。

### 认证令牌是无效还是已到期?

认证令牌必须具有有效的到期 (**exp**) 声明。声明的值必须晚于向队列管理器提供令牌的时间。如果令牌具有 not before (**nbft**) 声明, 那么该令牌的有效时间必须早于将该令牌提供给队列管理器的时间。有关更多信息, 请参阅 [认证令牌需求](#)。

### 是否拒绝有效认证令牌?

如果令牌满足认证令牌的所有需求, 并且队列管理器配置正确, 请检查队列管理器的错误日志中是否存在错误消息。错误消息可能包含描述错误原因的错误代码。有关更多信息, 请参阅 [令牌认证错误代码](#)。

### 应用程序是否尝试使用客户机重新连接?

如果指定了 reconnect 选项, 那么不能使用令牌来连接到 IBM MQ。如果应用程序提供认证令牌, 并在 MQCNO 结构中指定 MQCNO\_RECONNECT 或 MQCNO\_RECONNECT\_Q\_MGR 选项, 那么连接将失败并将原因码 2547 MQRC\_RECONNECT\_不兼容 返回到应用程序。

### 相关概念

[使用认证令牌](#)

[认证令牌的需求](#)

### 相关任务

[配置队列管理器以接受 AuthTokens](#)

[在应用程序中使用认证令牌](#)

### 相关参考

[qm.ini 文件的 AuthToken 节](#)

[MQCSP-安全性参数](#)

## 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: JMSMQ00001: 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.es.e.jmqi.InterceptedJmqiImpl.jmqiConnect(InterceptedJmqiImpl.java:376)
at com.ibm.mq.es.e.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.i(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

用于帮助您解决 IBM MQ 应用程序的 Microsoft Windows Communication Foundation (WCF) 定制通道运行问题的故障诊断信息。

### 相关概念

第 386 页的『[FFST:WCF XMS First Failure Support Technology](#)』

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 465 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 279 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件元数据中。

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 386 页的『FFST:WCF XMS First Failure Support Technology』](#)

## 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)
    {
    }
}
}
```

## 对 XMS .NET 问题进行故障诊断

用于帮助您解决运行 XMS .NET 应用程序的问题的故障诊断信息。

### 相关任务

[第 465 页的『跟踪 XMS .NET 应用程序』](#)

如果您正在使用 IBM MQ classes for XMS .NET Framework，那么可以从应用程序配置文件以及 XMS 环境变量配置跟踪。如果您正在使用 IBM MQ classes for XMS .NET (.NET Standard 和 .NET 6 库)，那么必须从 XMS 环境变量配置跟踪。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。

[第 279 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 86 页的『IBM MQ .NET 问题故障诊断』](#)

用于帮助您解决运行 IBM MQ .NET 应用程序的问题的故障诊断信息。

[开发 XMS .NET 应用程序](#)

## XMS .NET 问题的故障诊断提示

使用以下提示可帮助您对使用 XMS .NET 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 [连接认证](#)。

### 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 AIX, Linux, and Windows、Linux 或 Windows 32 位机器上将 double 属性的值设置为 4.9E-324，而 XMS .NET 在 64 位机器上运行，那么 GetDoubleProperty() 返回的值为 0.0。这是 Microsoft .NET Framework 2.0 的已知问题。

### 相关任务

[开发 XMS .NET 应用程序](#)

## **V 9.4.0** XMS .NET 客户机库抛出的常见 SSL 错误代码

从 IBM MQ 9.4.0 开始，对 XMS .NET 客户机 (amqmxmsstd.dll) 诊断消息的改进为 SSL 相关问题提供了更具体的异常机制。

从 IBM MQ 9.4.0 开始，XMS .NET 客户机库为与 SSL 相关的错误提供有意义的特定错误。MQRC 原因码与其他 .NET 客户机库 (例如 C) 一致。

下表显示了许多场景中的一些场景，其中 XMS .NET 客户机库从 IBM MQ 9.4.0 抛出与其他客户机 .NET 库一致的错误：



场景	IBM MQ 9.4.0 之前的错误消息	来自 IBM MQ 9.4.0 的错误消息
为 SSL 密钥存储库提供的参数时, MQC.MQCA_SSL_KEY_REPOSITORY 或 MQEnvironment.SSLKeyRepository。	2059-MQRC_Q_MGR_NOT_AVAILABLE	2381-MQRC_KEY_REPOSITORY_ERROR
使用 SSL 连接到 IBM MQ 服务器时, 在客户机上设置了无效的密码套件。	2538-MQRC_HOST_NOT_AVAILABLE	2393-MQRC_KEY_REPOSITORY_ERROR
使用 SSL 连接到 IBM MQ 服务器时在客户机上设置了无效对等名称。	2059-MQRC_Q_MGR_NOT_AVAILABLE	2399-MQRC_SSL_PEER_NAME_ERROR
当对等名称在使用 SSL 连接到 IBM MQ 服务器时不匹配时。	2059-MQRC_Q_MGR_NOT_AVAILABLE	2398-MQRC_SSL_PEER_NAME_ERROR
在 SSL 握手期间在客户机或服务上设置了无效证书时。	2059-MQRC_Q_MGR_NOT_AVAILABLE	2393-MQRC_KEY_REPOSITORY_ERROR
当用于在客户机和服务器之间提供 SSL 连接的通道未提供密码套件时。	2059-MQRC_Q_MGR_NOT_AVAILABLE	2393-MQRC_KEY_REPOSITORY_ERROR
当从客户机设置的密码套件算法与 IBM MQ 服务器不匹配时。	2059-MQRC_Q_MGR_NOT_AVAILABLE	2393-MQRC_KEY_REPOSITORY_ERROR
当没有对创建 IBM MQ 密钥存储库的文件夹的许可权时。	2059-MQRC_Q_MGR_NOT_AVAILABLE	2393-MQRC_KEY_REPOSITORY_ERROR

有关错误的更详细说明, 请参阅 `MQ_DATA_DIRECTORY/qmgrs/errors/AMQERR*.log` 中的队列管理器日志。

### 相关概念

[安装 IBM MQ classes for XMS .NET](#)

## Troubleshooting IBM MQ for z/OS problems

IBM MQ for z/OS, CICS, Db2, and IMS produce diagnostic information which can be used for problem determination.

### About this task

This section contains information about the following topics:

- The recovery actions attempted by the queue manager when a problem is detected.
- IBM MQ for z/OS abends, and the information produced when an abend occurs.
- The diagnostic information produced by IBM MQ for z/OS, and additional sources of useful information.

The type of information provided to help with problem determination and application debugging depends on the type of error encountered, and the way your subsystem is set up.

See the following subtopics for more information about problem determination and diagnostic information on IBM MQ for z/OS.

- [“IBM MQ for z/OS performance constraints” on page 225](#)
- [“IBM MQ for z/OS recovery actions” on page 227](#)
- [“IBM MQ for z/OS abends” on page 228](#)
- [“Diagnostic information produced on IBM MQ for z/OS” on page 231](#)
- [“Other sources of problem determination information for IBM MQ for z/OS” on page 233](#)
- [“Diagnostic aids for CICS” on page 235](#)
- [“Diagnostic aids for IMS” on page 246](#)
- [“Diagnostic aids for Db2” on page 246](#)
- [“IBM MQ for z/OS dumps” on page 246](#)
- [“Dealing with performance problems on z/OS” on page 266](#)
- [“Dealing with incorrect output on z/OS” on page 272](#)

### Related concepts

[“使用错误日志” on page 367](#)

有各种错误日志可用于帮助进行问题确定和故障诊断。

[“First Failure Support Technology \(FFST\)” on page 376](#)

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。

### Related tasks

[“IBM MQ 故障诊断和支持” on page 5](#)

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。

[“跟踪” on page 387](#)

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 performance constraints

Use this topic to investigate z/OS resources that can cause performance constraints.

There are a number of decisions to be made when customizing IBM MQ for z/OS that can affect the way your systems perform. These decisions include:

- The size and placement of data sets
- The allocation of buffers
- The distribution of queues among page sets, and Coupling Facility structures
- The number of tasks that you allow to access the queue manager at any one time

### Log buffer pools

Insufficient log buffers can cause applications to wait until a log buffer is available, which can affect IBM MQ performance. RMF reports might show heavy I/O to volumes that hold log data sets.

There are three parameters you can use to tune log buffers. The most important is OUTBUFF. If the log manager statistic QJSTWTB is greater than 0, increase the size of the log buffer. This parameter controls the number of buffers to be filled before they are written to the active log data sets (in the range 1 - 256). Commits and out-of-syncpoint processing of persistent messages cause log buffers to be written out to the log. As a result this parameter might have little effect except when processing large messages, and the number of commits or out of sync point messages is low. These parameters are specified in the CSQ6LOGP macro (see [Using CSQ6LOGP](#) for details), and the significant ones are:

#### OUTBUFF

This parameter controls the size of the output buffer (in the range 40 KB through 4000 KB).

## WRTHRSR

This parameter controls the number of buffers to be filled before they are written to the active log data sets (in the range 1 through 256).

You must also be aware of the LOGLOAD parameter of the CSQ6SYSP macro. This parameter specifies the number of log records that are written between checkpoint records. The range is 200 through 16 000 000 but a typical value for a large system is 500 000. If a value is too small you receive frequent checkpoints, which consume processor time and can cause additional disk I/O.

## Buffer pool size

There is a buffer pool associated with each page set. You can specify the number of buffers in the buffer pool using the `DEFINE BUFFPOOL` command.

Incorrect specification of buffer pool size can adversely affect IBM MQ performance. The smaller the buffer pool, the more frequently physical I/O is required. RMF might show heavy I/O to volumes that hold page sets. For buffer pools with only short-lived messages the buffer manager statistics QPSTSLA, QPSTSOS, and QPSTRIO must typically be zero. For other buffer pools, QPSTSOS and QPSTSLA must be zero.

## Distribution of data sets on available DASD

The distribution of page data sets on DASD can have a significant effect on the performance of IBM MQ.

Place log data sets on low usage volumes with log  $n$  and log  $n+1$  on different volumes. Ensure that dual logs are placed on DASD on different control units and that the volumes are not on the same physical disk.

## Distribution of queues on page sets

The distribution of queues on page sets can affect performance. This change in performance can be indicated by poor response times experienced by transactions using specific queues that reside on heavily used page sets. RMF reports might show heavy I/O to volumes containing the affected page sets.

You can assign queues to specific page sets by defining storage class (STGCLASS) objects specifying a particular page set, and then defining the STGCLASS parameter in the queue definition. It is a good idea to define heavily used queues on different page sets in this way.

## Distribution of queues on Coupling Facility structures

The distribution of queues on Coupling Facility structures can affect performance.

A queue sharing group can connect to up to 64 Coupling Facility structures, one of which must be the administration structure. You can use the remaining 63 Coupling Facility structures for IBM MQ data with each structure holding up to 512 queues. If you need more than one Coupling Facility structure, separate the queues across several structures based on the function of the queue.

There are some steps you can take to maximize efficiency:

- Delete any Coupling Facility structures you no longer require.
- Place all the queues used by an application on the same Coupling Facility to make application processing efficient.
- If work is particularly performance sensitive, choose a faster Coupling Facility structure.

Consider that if you lose a Coupling Facility structure, you lose any non-persistent messages stored in it. The loss of these non-persistent messages can cause consistency problems if queues are spread across various Coupling Facility structures. To use persistent messages, you must define the Coupling Facility structures with at least CFLEVEL(3) and RECOVER(YES).

## Limitation of concurrent threads

The number of tasks accessing the queue manager can also affect performance, particularly if there are other constraints, such as storage, or there are many tasks accessing a few queues. The symptoms can be heavy I/O against one or more page sets, or poor response times from tasks known to access the same queues. The number of threads in IBM MQ is limited to 32767 for both TSO and Batch.

In a CICS environment, you can use CICS MAXTASK to limit concurrent access.

## Using the IBM MQ trace for administration

Although you might have to use specific traces on occasion, using the trace facility has a negative effect on the performance of your systems.

Consider what destination you want your trace information sent to. Using the internal trace table saves I/O, but it is not large enough for traces that produce large volumes of data.

The statistics trace gathers information at intervals. The intervals are controlled by the STATIME parameter of the CSQ6SYSP macro, described in [Using CSQ6SYSP](#). An accounting trace record is produced when the task or channel ends, which might be after many days.

You can limit traces by class, resource manager identifier (RMID), and instrumentation facility identifier (IFCID) to reduce the volume of data collected. See [START TRACE](#) for more information.

## IBM MQ for z/OS recovery actions

Use this topic to understand some of the recovery actions for user detected and queue manager detected errors.

IBM MQ for z/OS can recover from program checks caused by incorrect user data. A completion and reason code are issued to the caller. These codes are documented in [IBM MQ for z/OS messages, completion, and reason codes](#).

## Program errors

Program errors might be associated with user application program code or IBM MQ code, and fall into two categories:

- [User detected errors](#)
- [“Queue manager detected errors” on page 227](#)

## User detected errors

User detected errors are detected by the user (or a user-written application program) when the results of a service request are not as expected (for example, a nonzero completion code). The collection of problem determination data cannot be automated because detection occurs after the IBM MQ function has completed. Rerunning the application with the IBM MQ user parameter trace facility activated can provide the data needed to analyze the problem. The output from this trace is directed to the *generalized trace facility* (GTF).

You can turn the trace on and off using an operator command. See [“Tracing on z/OS” on page 405](#) for more information.

## Queue manager detected errors

The queue manager detects errors such as:

- A program check
- A data set filling up
- An internal consistency error

IBM MQ analyzes the error and takes the following actions:

- If the problem was caused by a user or application error (such as an invalid address being used), the error is reflected back to the application by completion and reason codes.
- If the problem was not caused by a user or application error (for example, all available DASD has been used, or the system detected an internal inconsistency), IBM MQ recovers if possible, either by sending completion and reason codes to the application, or if this is not possible, by stopping the application.
- If IBM MQ cannot recover, it terminates with a specific reason code. An SVC dump is typically taken recording information in the *system diagnostic work area* (SDWA) and *variable recording area* (VRA) portions of the dump, and an entry is made in SYS1.LOGREC.

## IBM MQ for z/OS abends

Abends can occur in WebSphere for z/OS or other z/OS systems. Use this topic to understand the IBM MQ system abend codes and how to investigate abends which occur in CICS, IMS, and z/OS.

IBM MQ for z/OS uses two system abend completion codes, X'5C6' and X'6C6'. These codes identify:

- Internal errors encountered during operation
- Diagnostic information for problem determination
- Actions initiated by the component involved in the error

### X'5C6'

An X'5C6' abend completion code indicates that IBM MQ has detected an internal error and has terminated an internal task (TCB) or a user-connected task abnormally. Errors associated with an X'5C6' abend completion code might be preceded by a z/OS system code, or by internal errors.

Examine the diagnostic material generated by the X'5C6' abend to determine the source of the error that actually resulted in a subsequent task or subsystem termination.

### X'6C6'

An X'6C6' abend completion code indicates that IBM MQ has detected a severe error and has terminated the queue manager abnormally. When an X'6C6' is issued, IBM MQ has determined that continued operation could result in the loss of data integrity. Errors associated with an X'6C6' abend completion code might be preceded by a z/OS system error, one or more X'5C6' abend completion codes, or by error message CSQV086E indicating abnormal termination of IBM MQ.

Table 8 on page 228 summarizes the actions and diagnostic information available to IBM MQ for z/OS when these abend completion codes are issued. Different pieces of this information are relevant in different error situations. The information produced for a particular error depends upon the specific problem. For more information about the z/OS services that provide diagnostic information, see “Diagnostic information produced on IBM MQ for z/OS” on page 231.

	X'5C6'	X'6C6'
Explanation	<ul style="list-style-type: none"> <li>• Error during IBM MQ normal operation</li> </ul>	<ul style="list-style-type: none"> <li>• Severe error; continued operation might jeopardize data integrity</li> </ul>

Table 8. Abend completion codes (continued)		
	X'5C6'	X'6C6'
System action	<ul style="list-style-type: none"> <li>Internal IBM MQ task is abended</li> <li>Connected user task is abended</li> </ul>	<ul style="list-style-type: none"> <li>The entire IBM MQ subsystem is abended</li> <li>User task with an active IBM MQ connection might be abnormally terminated with an X'6C6' code</li> <li>Possible MEMTERM (memory termination) of connected allied address space</li> </ul>
Diagnostic information	<ul style="list-style-type: none"> <li>SVC dump</li> <li>SYS1.LOGREC entry</li> <li>VRA data entries</li> </ul>	<ul style="list-style-type: none"> <li>SYS1.LOGREC</li> <li>VRA data entries</li> </ul>
Associated reason codes	<ul style="list-style-type: none"> <li>IBM MQ abend reason code</li> <li>Associated z/OS system codes</li> </ul>	<ul style="list-style-type: none"> <li>Subsystem termination reason code</li> <li>z/OS system completion codes and X'5C6' codes that precede the X'6C6' abend</li> </ul>
Location of accompanying codes	<ul style="list-style-type: none"> <li>SVC dump title</li> <li>Message CSQW050I</li> <li>Register 15 of SDWA section <i>General Purpose Registers at Time of Error</i></li> <li>SYS1.LOGREC entries</li> <li>VRA data entries</li> </ul>	<ul style="list-style-type: none"> <li>SYS1.LOGREC</li> <li>VRA data entries</li> <li>Message CSQV086E, which is sent to z/OS system operator</li> </ul>

### Related concepts

[“Dealing with abends on IBM MQ for z/OS” on page 229](#)

Abends can occur with applications and other z/OS systems. Use this topic to investigate program abends, batch abends, CICS transaction abends, and IMS transaction abends.

[“CICS, IMS, and z/OS abends” on page 230](#)

Use this topic to investigate abends from CICS, IMS, and z/OS.

[“Diagnostic information produced on IBM MQ for z/OS” on page 231](#)

Use this topic to investigate some of the diagnostic information produced by z/OS that can be useful in problem determination and understand how to investigate error messages, dumps, console logs, job output, symptom strings, and queue output.

[“IBM MQ for z/OS dumps” on page 246](#)

Use this topic for information about the use of dumps in problem determination. It describes the steps you should take when looking at a dump produced by an IBM MQ for z/OS address space.

### Dealing with abends on IBM MQ for z/OS

Abends can occur with applications and other z/OS systems. Use this topic to investigate program abends, batch abends, CICS transaction abends, and IMS transaction abends.

### Types of abend

Program abends can be caused by applications failing to check, and respond to, reason codes from IBM MQ. For example, if a message has not been received, using fields that would have been set up in the message for calculation might cause X'0C4' or X'0C7' abends (ASRA abends in CICS).



The following pieces of information indicate a program abend:

- Error messages from IBM MQ in the console log
- CICS error messages
- CICS transaction dumps
- IMS region dumps
- IMS messages on user or master terminal
- Program dump information in batch or TSO output
- Abend messages in batch job output
- Abend messages on the TSO screen

If you have an abend code, see one of the following documentation for an explanation of the cause of the abend:

- For IBM MQ for z/OS abends (abend codes X' 5C6 ' and X' 6C6 '), see [IBM MQ for z/OS messages, completion, and reason codes](#)
- For batch abends, the [z/OS MVS System Codes](#) manual
- For CICS abends, [CICS Messages](#)
- For IMS abends, [IMS Messages and Codes](#)
- For Db2 abends, [Db2 Messages](#)
- For RRS abends, see ATR messages in [z/OS MVS System Messages, Volume 3](#)
- For XES abends, see IXL messages in [z/OS MVS System Messages, Volume 10](#)

## Batch abends

Batch abends cause an error message containing information about the contents of registers to be displayed in the syslog. TSO abends cause an error message containing similar information to be produced on the TSO screen. A SYSUDUMP is taken if there is a SYSUDUMP DD statement for the step (see [“IBM MQ for z/OS dumps”](#) on page 246 ).

## CICS transaction abends

CICS transaction abends are recorded in the CICS CSMT log, and a message is produced at the terminal (if there is one). A CICS AICA abend indicates a possible loop. See [“Dealing with loops on z/OS”](#) on page 271 for more information. If you have a CICS abend, using CEDF and the CICS trace might help you to find the cause of the problem. See [CICS Troubleshooting](#), formerly the *CICS Problem Determination Guide* for more information.

## IMS transaction abends

IMS transaction abends are recorded on the IMS master terminal, and an error message is produced at the terminal (if there is one). If you have an IMS abend, see [Troubleshooting for IMS](#).

### **CICS, IMS, and z/OS abends**

Use this topic to investigate abends from CICS, IMS, and z/OS.

## CICS abends

A CICS abend message is sent to the terminal, if the application is attached to one, or to the CSMT log. CICS abend codes are explained in [Transaction abend codes](#).

## IMS abends

An IMS application might abend in one of the following circumstances:

- A normal abend.
- An IMS pseudo abend, with an abend code such as U3044 resulting from an error in an ESAF exit program.
- Abend 3051 or 3047, when the REO (region error option) has been specified as "Q" or "A", and an IMS application attempts to reference a non-operational external subsystem, or when resources are unavailable at the time when a thread is created.

An IMS message is sent to the user terminal or job output, and the IMS master terminal. The abend might be accompanied by a region dump.

## **z/OS abends**

During IBM MQ operation, an abend might occur with a z/OS system completion code. If you receive a z/OS abend, see the appropriate z/OS publication.

### **Diagnostic information produced on IBM MQ for z/OS**

Use this topic to investigate some of the diagnostic information produced by z/OS that can be useful in problem determination and understand how to investigate error messages, dumps, console logs, job output, symptom strings, and queue output.

IBM MQ for z/OS functional recovery routines use z/OS services to provide diagnostic information to help you in problem determination.

The following z/OS services provide diagnostic information:

#### **SVC dumps**

The IBM MQ abend completion code X'5C6' uses the z/OS [SDUMP](#) service to create SVC dumps. The content and storage areas associated with these dumps vary, depending on the specific error and the state of the queue manager at the time the error occurred.

#### **SYS1.LOGREC**

Entries are requested in the SYS1.LOGREC data set at the time of the error using the z/OS [SETRP](#) service. The following information is also recorded in SYS1.LOGREC:

- Subsystem abnormal terminations
- Secondary abends occurring in a recovery routine
- Requests from the recovery termination manager

#### **Variable recording area (VRA) data**

Data entries are added to the VRA of the SDWA by using a z/OS VRA defined key. VRA data includes a series of diagnostic data entries common to all IBM MQ for z/OS abend completion codes. Additional information is provided during initial error processing by the invoking component recovery routine, or by the recovery termination manager.

See [SDWA information](#) for more details.

IBM MQ for z/OS provides unique messages that, together with the output of dumps, are aimed at providing sufficient data to allow diagnosis of the problem without having to try to reproduce it. This is known as first failure data capture.

## **Error messages**

IBM MQ produces an error message when a problem is detected. IBM MQ diagnostic messages begin with the prefix CSQ. Each error message generated by IBM MQ is unique; that is, it is generated for one and only one error. Information about the error can be found in [IBM MQ for z/OS messages, completion, and reason codes](#).

The first three characters of the names of IBM MQ modules are also usually CSQ. The exceptions to this are modules for C++ (IMQ), and the header files (CMQ). The fourth character uniquely identifies the component. Characters five through eight are unique within the group identified by the first four characters.

Make sure that you have some documentation on application messages and codes for programs that were written at your installation, as well as viewing [IBM MQ for z/OS messages, completion, and reason codes](#)

There might be some instances when no message is produced, or, if one is produced, it cannot be communicated. In these circumstances, you might have to analyze a dump to isolate the error to a particular module. For more information about the use of dumps, see [“IBM MQ for z/OS dumps” on page 246](#).

## Dumps

Dumps are an important source of detailed information about problems. Whether they are as the result of an abend or a user request, they allow you to see a snapshot of what was happening at the moment the dump was taken. [“IBM MQ for z/OS dumps” on page 246](#) contains guidance about using dumps to locate problems in your IBM MQ system. However, because they only provide a snapshot, you might need to use them with other sources of information that cover a longer period of time, such as logs.

Snap dumps are also produced for specific types of error in handling MQI calls. The dumps are written to the CSQSNAP DD.

## Console logs and job output

You can copy console logs into a permanent data set, or print them as required. If you are only interested in specific events, you can select which parts of the console log to print.

Job output includes output produced from running the job, as well as that from the console. You can copy this output into permanent data sets, or print it as required. You might need to collect output for all associated jobs, for example CICS, IMS, and IBM MQ.

## Symptom strings

Symptom strings display important diagnostic information in a structured format. When a symptom string is produced, it is available in one or more of the following places:

- On the z/OS system console
- In SYS1.LOGREC
- In any dump taken

Figure 15 on page 232 shows an example of a symptom string.

```
PIDS/ 5655R3600 RIDS/CSQMAIN1 AB/S6C6 PRCS/0E30003
```

*Figure 15. Sample symptom string*

The symptom string provides a number of keywords that IBM Support can use to search the software support database. If you report a problem, you are often asked to quote the symptom string.

Although the symptom string is designed to provide keywords for searching the database, it can also give you a lot of information about what was happening at the time the error occurred, and it might suggest an obvious cause or a promising area to start your investigation.

## Queue information

You can display information about the status of queues by using the operations and control panels. Alternatively you can enter the DISPLAY QUEUE and DISPLAY QSTATUS commands from the z/OS console.

**Note:** If the command was issued from the console, the response is copied to the console log, allowing the documentation to be kept together compactly.

### Related concepts

[“Other sources of problem determination information for IBM MQ for z/OS” on page 233](#)

Use this topic to investigate other sources of information for IBM MQ for z/OS problem determination.

[“Diagnostic aids for CICS” on page 235](#)

You can use the CICS diagnostic transactions to display information about queue manager tasks, and MQI calls. Use this topic to investigate these facilities.

[“Diagnostic aids for IMS” on page 246](#)

Use this topic to investigate IMS diagnostic facilities.

[“Diagnostic aids for Db2” on page 246](#)

Use this topic to investigate references for Db2 diagnostic tools.

### Related tasks

[“Tracing on z/OS” on page 405](#)

There are different trace options that can be used for problem determination with IBM MQ. Use this topic to understand the different options and how to control trace.

## Other sources of problem determination information for IBM MQ for z/OS

Use this topic to investigate other sources of information for IBM MQ for z/OS problem determination.

You might find the following items of documentation useful when solving problems with IBM MQ for z/OS.

- [Your own documentation](#)
- [Documentation for the products you are using](#)
- [Source listings and link-edit maps](#)
- [Change log](#)
- [System configuration charts](#)
- [Information from the DISPLAY CONN command](#)

### Your own documentation

Your own documentation is the collection of information produced by your organization about what your system and applications should do, and how they are supposed to do it. How much of this information you need depends on how familiar you are with the system or application in question, and could include:

- Program descriptions or functional specifications
- Flowcharts or other descriptions of the flow of activity in a system
- Change history of a program
- Change history of your installation
- Statistical and monitoring profile showing average inputs, outputs, and response times

### Documentation for the products you are using

The documentation for the product you are using are the InfoCenters in the IBM MQ library, and in the libraries for any other products you use with your application.

Make sure that the level of any documentation you refer to matches the level of the system you are using. Problems often arise through using either obsolete information, or information about a level of a product that is not yet installed.

## Source listings and link-edit maps

Include the source listings of any applications written at your installation with your set of documentation. (They can often be the largest single element of documentation.) Make sure that you include the relevant output from the linkage editor with your source listings to avoid wasting time trying to find your way through a load module with an out-of-date link map. Be sure to include the JCL at the beginning of your listings, to show the libraries that were used and the load library the load module was placed in.

## Change log

The information in the change log can tell you of changes made in the data processing environment that might have caused problems with your application program. To get the most out of your change log, include the data concerning hardware changes, system software (such as z/OS and IBM MQ) changes, application changes, and any modifications made to operating procedures.

## System configuration charts

System configuration charts show what systems are running, where they are running, and how the systems are connected to each other. They also show which IBM MQ, CICS, or IMS systems are test systems and which are production systems.

## Information from the DISPLAY CONN command

The DISPLAY CONN command provides information about which applications are connected to a queue manager, and information to help you to diagnose those that have a long-running unit of work. You could collect this information periodically and check it for any long-running units of work, and display the detailed information about that connection.

### 对 zHyper 链接进行故障诊断

有几种方法可以对 zHyper 链接性能问题进行故障诊断。

发出以下命令：

```
D SMS,DSNAME(dsname),STATS(ZHLWRITE)
```

以显示 zHyper 链路写入统计信息，其中 *dsname* 是用于日志写入的数据集的名称。

```
D SMS,DSNAME,STATS(ZHLWRITE) Start of Report
DATA SET VICY.MQ57.LOGCOPY1.DS01.DATA
STATISTICS Since 04/08/2024 10:51:57.096121
SUMMARY
TOTAL      %SYNC  -----%ASYNC-----
WRITE REQUESTS WRITES  SKIP LNKBSY  -EST  MISC DISABL
      1146    4.36    79.66    0.00    0.08    0.00    4.01
      -----%ASYNC-----
      MISS DELAY  DUAL
      11.86    0.00    0.00
DEVICE STATISTICS
TOTAL      %SYNC  -----%ASYNC-----
SSID DEVNO WRITES WRITES  SKIP LNKBSY  EST  MISC  MISS  DELAY
5144 0619C  2288    3.05  39.90  0.00  0.04  16.21  40.77  0.00
D SMS,DSNAME,STATS(ZHLWRITE) End of Report
```

上述输出显示同步完成的写操作所占的百分比，以及异步完成的写操作所占的百分比（按不使用同步写操作的原因细分）。

输出显示：

## SKIP

由于先前 zHyper 链路故障而未尝试 zHyper 链路的 zHyper 链路写入所占的百分比。

## 行 (LINKBSY)

迁到繁忙链接情况的 zHyper 链接写入的百分比。

## EST

未建立 zHyper 链路写访问权的 zHyper 链路写入的百分比。

## MISC

由于其他原因异步完成的 zHyper 链接写入的百分比。

## 小姐

由于高速缓存不命中而异步完成的 zHyper 链接写入的百分比。

## 延迟

由于操作延迟而异步完成的 zHyper 链接写入的百分比。

以及, 如果存在

## DISABL

未启用 zHyper 链路的 zHyper 链路写入的百分比。

## DUAL

由于双重日志记录请求中的其他数据集而异步完成的 zHyper 链接写入的百分比。

请参阅 [IBM zHyperLink for z/OS 入门](#) 的附录 B, 以获取有关使用 zHyperLink 进行问题确定的更多信息以及各种命令和过程。

## 查找设备的序列号

您可以通过运行以下命令来查找序列号和 SCU 号:

```
DS QD,nnnn
```

其中 *nnnn* 是设备号。例如:

```
DS QD,9200
IEE459I 07.04.45 DEVSERV QDASD 742
UNIT VOLSER SCUTYPE DEVTYPE CYL SSID SCU-SERIAL DEV-SERIAL EFC
09200 CONCD1 2107981 2107900 1113 0230 0175-FCC41 0175-FCC41 *OK
**** 1 DEVICE(S) MET THE SELECTION CRITERIA
**** 0 DEVICE(S) FAILED EXTENDED FUNCTION CHECKING
```

使用此命令的输出来获取显示 zHyper 链接会话数所需的 SCU-SERIAL 号。

## 样本 DS QD, scu-serial 和 ZHL 命令

```
DS QD,MACH=0175-FCC41,ZHL
IEE459I 04.10.26 DEVSERV QDASD 419
UNIT VOLSER SCUTYPE DEVTYPE CYL SSID SCU-SERIAL DEV-SERIAL EFC
09200 CONCD1 2107981 2107900 1113 0230 0175-FCC41 0175-FCC41 *OK
ZHYPERLINK WRITE TOKENS ASSIGNED
TOKEN DATE TIME UNIT DATA SET NAME
00AD 07/07/21 15:10 09C16 DB2BL.D2B1.LOGCOPY2.DS02.DATA
01AE 07/07/21 15:10 09B96 DB2BL.D2B1.LOGCOPY1.DS02.DATA
02AF 07/07/21 15:10 09B96 DB2BL.D2B2.LOGCOPY1.DS02.DATA
03B0 07/07/21 15:10 09639 DB2BL.D2B2.LOGCOPY2.DS02.DATA
**** 4 WRITE TOKEN(S) MET THE SELECTION CRITERIA
**** 1 DEVICE(S) MET THE SELECTION CRITERIA
**** 0 DEVICE(S) FAILED EXTENDED FUNCTION CHECKING
```

每个令牌都表示一个已连接的 zHyper 链接写入会话, 显示哪些数据集正在使用 zHyper 链接。

## Diagnostic aids for CICS

You can use the CICS diagnostic transactions to display information about queue manager tasks, and MQI calls. Use this topic to investigate these facilities.



You can use the CKQC transaction (the CICS adapter control panels) to display information about queue manager tasks, and what state they are in (for example, a GET WAIT). See [Administering IBM MQ for z/OS](#) for more information about CKQC.

The application development environment is the same as for any other CICS application, and so you can use any tools normally used in that environment to develop IBM MQ applications. In particular, the *CICS execution diagnostic facility* (CEDF) traps entry to and exit from the CICS adapter for each MQI call, as well as trapping calls to all CICS API services. Examples of the output produced by this facility are given in [“Examples of CEDF output from MQI calls”](#) on page 236.

The CICS adapter also writes trace entries to the CICS trace. These entries are described in [“CICS adapter trace entries”](#) on page 414.

Additional trace and dump data is available from the CICS region. These entries are as described in [Troubleshooting CICS](#).

### Examples of CEDF output from MQI calls

Examples of the output produced by the CICS execution diagnostic facility (CEDF) when using IBM MQ.

These examples show the data produced on entry to and exit from the following MQI calls, in both hexadecimal and character format. Other MQI calls produce similar data.

#### Related reference

[Function calls](#)

*Example CEDF output for the MQOPEN call*

The parameters for this call are as follows:

<i>Table 9. Parameters for the MQOPEN call</i>	
Parameter	Description
ARG 000	Connection handle
ARG 001	Object descriptor
ARG 002	Options
ARG 003	Object handle
ARG 004	Completion code
ARG 005	Reason code

```

STATUS: ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'000000000000000100000000200004044') AT X'05ECAFD8'
001: ARG 001 (X'D6C440400000000100000001C3C5C4C6') AT X'00144910'
001: ARG 002 (X'000000720000000000000000000000') AT X'001445E8'
001: ARG 003 (X'000000000000007200000000000000') AT X'001445E4'
001: ARG 004 (X'000000000000000000000000000000') AT X'001445EC'
001: ARG 005 (X'000000000000000000000000000000') AT X'001445F0'

```

*Figure 16. Example CEDF output on entry to an MQOPEN call (hexadecimal)*

```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'00000000000000010000000200004044')           AT X'05ECAFD8'
001: ARG 001 (X'D6C440400000000100000001C3C5C4C6')           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'

```

Figure 17. Example CEDF output on exit from an MQOPEN call (hexadecimal)

```

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 ('.....')

```

Figure 18. Example CEDF output on entry to an MQOPEN call (character)

```

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 ('.....')

```

Figure 19. Example CEDF output on exit from an MQOPEN call (character)

### Related reference

[MQOPEN - Open object](#)

*Example CEDF output for the MQCLOSE call*

The parameters for this call are:

<i>Table 10. Parameters for the MQCLOSE call</i>	
Parameter	Description
ARG 000	Connection handle
ARG 001	Object handle
ARG 002	Options
ARG 003	Completion code
ARG 004	Reason code

```

STATUS: ABOUT TO EXECUTE COMMAND
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'00000000000000010000000200004044') AT X'05ECAFD8'
001: ARG 003 (X'00000000000000000000008000000008') AT X'001445EC'
001: ARG 004 (X'00000000000000080000000800000060') AT X'001445F0'

```

Figure 20. Example CEDF output on entry to an MQCLOSE call (hexadecimal)

```

STATUS: COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'00000000000000000000007200000000') AT X'001445E0'
001: ARG 001 (X'00000000000000072000000000000000') AT X'001445E4'
001: ARG 002 (X'00000000000000010000000200004044') AT X'05ECAFD8'
001: ARG 003 (X'00000000000000000000008000000008') AT X'001445EC'
001: ARG 004 (X'00000000000000080000000800000060') AT X'001445F0'

```

Figure 21. Example CEDF output on exit from an MQCLOSE call (hexadecimal)

```

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 ('.....')

```

Figure 22. Example CEDF output on entry to an MQCLOSE call (character)

```

STATUS: COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('.....')
001: ARG 003 ('.....')
001: ARG 004 ('.....')

```

Figure 23. Example CEDF output on exit from an MQCLOSE call (character)

### Related reference

[MQCLOSE - Close object](#)

Example CEDF output for the MQPUT call

The parameters for this call are:

Table 11. Parameters for the MQPUT call

Parameter	Description
ARG 000	Connection handle
ARG 001	Object handle
ARG 002	Message descriptor

Table 11. Parameters for the MQPUT call (continued)

Parameter	Description
ARG 003	Put message options
ARG 004	Buffer length
ARG 005	Message data
ARG 006	Completion code
ARG 007	Reason code

```

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'D4C440400000000100000000000008') AT X'001449B8'
001: ARG 003 (X'D7D4D6400000001000000240000000') AT X'00144B48'
001: ARG 004 (X'0000008000000000000000000040000') AT X'001445F4'
001: ARG 005 (X'5C5CC8C5D3D3D640E6D6D9D3C45C5C') AT X'00144BF8'
001: ARG 006 (X'000000000000000000000080000000') AT X'001445EC'
001: ARG 007 (X'000000000000008000000000000000') AT X'001445F0'
    
```

Figure 24. Example CEDF output on entry to an MQPUT call (hexadecimal)

```

STATUS: COMMAND EXECUTION COMPLETE
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'D4C440400000000100000000000008') AT X'001449B8'
001: ARG 003 (X'D7D4D6400000001000000240000000') AT X'00144B48'
001: ARG 004 (X'0000008000000000000000000040000') AT X'001445F4'
001: ARG 005 (X'5C5CC8C5D3D3D640E6D6D9D3C45C5C') AT X'00144BF8'
001: ARG 006 (X'000000000000000000000080000000') AT X'001445EC'
001: ARG 007 (X'000000000000008000000000000000') AT X'001445F0'
    
```

Figure 25. Example CEDF output on exit from an MQPUT call (hexadecimal)

```

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 ('.....')
    
```

Figure 26. Example CEDF output on entry to an MQPUT call (character)







Table 13. Parameters for the MQGET call (continued)

Parameter	Description
ARG 006	Message length
ARG 007	Completion code
ARG 008	Reason code

```

STATUS: ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'0000000000000000100000072000000000')           AT X'001445E0'
001: ARG 001 (X'00000001000000072000000000000000')           AT X'001445E4'
001: ARG 002 (X'D4C440400000000100000000000000')           AT X'001449B8'
001: ARG 003 (X'C7D4D6400000000100004044FFFFFFFF')           AT X'00144B00'
001: ARG 004 (X'0000008000000000000000000040000')           AT X'001445F4'
001: ARG 005 (X'00000000000000000000000000000000')           AT X'00144C00'
001: ARG 006 (X'00000000000000000000400000000000')           AT X'001445F8'
001: ARG 007 (X'000000000000000000008000000000')           AT X'001445EC'
001: ARG 008 (X'000000000000008000000000000000')           AT X'001445F0'
    
```

Figure 32. Example CEDF output on entry to an MQGET call (hexadecimal)

```

STATUS: COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'0000000000000000100000072000000000')           AT X'001445E0'
001: ARG 001 (X'00000001000000072000000000000000')           AT X'001445E4'
001: ARG 002 (X'D4C44040000000010000000000000008')           AT X'001449B8'
001: ARG 003 (X'C7D4D6400000000100004044FFFFFFFF')           AT X'00144B00'
001: ARG 004 (X'000000800000000080000000000040000')           AT X'001445F4'
001: ARG 005 (X'5C5CC8C5D3D3D640E6D09D3C45C5C5')           AT X'00144C00'
001: ARG 006 (X'00000080000000000000400000000000')           AT X'001445F8'
001: ARG 007 (X'00000000000000000000800000000008')           AT X'001445EC'
001: ARG 008 (X'000000000000008000000080000000')           AT X'001445F0'
    
```

Figure 33. Example CEDF output on exit from an MQGET call (hexadecimal)

```

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 ('.....')
    
```

Figure 34. Example CEDF output on entry to an MQGET call (character)

```

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 ('.....')

```

Figure 35. Example CEDF output on exit from an MQGET call (character)

**Related reference**

[MQGET - Get message](#)

Example CEDF output for the MQINQ call

The parameters for this call are:

Table 14. Parameters for the MQINQ call

Parameter	Description
ARG 000	Connection handle
ARG 001	Object handle
ARG 002	Count of selectors
ARG 003	Array of attribute selectors
ARG 004	Count of integer attributes
ARG 005	Integer attributes
ARG 006	Length of character attributes buffer
ARG 007	Character attributes
ARG 008	Completion code
ARG 009	Reason code

```

STATUS:  ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'000000000000000100000000200004044')           AT X'05ECAFC'
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'000000000000000100000000200004044')           AT X'05ECAFC'
001: ARG 007 (X'00000000000000000000000000000000')           AT X'00144C18'
001: ARG 008 (X'000000000000000000000000800000008')           AT X'001445EC'
001: ARG 009 (X'000000000000000800000008000040000')           AT X'001445F0'

```

Figure 36. Example CEDF output on entry to an MQINQ call (hexadecimal)

```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'00000000000000010000000200004044')          AT X'05ECAFCF'
001: ARG 001 (X'000000010000007200000000000000')          AT X'001445E4'
001: ARG 002 (X'000000020000404485ECA00885ECA220')          AT X'05ECAFD4'
001: ARG 003 (X'0000000D0000000C00400000000000')          AT X'00144C08'
001: ARG 004 (X'000000020000404485ECA00885ECA220')          AT X'05ECAFD4'
001: ARG 005 (X'004000000000000000000000000000')          AT X'00144C10'
001: ARG 006 (X'00000000000000010000000200004044')          AT X'05ECAFCF'
001: ARG 007 (X'000000000000000000000000000000')          AT X'00144C18'
001: ARG 008 (X'00000000000000000000000800000008')          AT X'001445EC'
001: ARG 009 (X'00000000000000080000000800040000')          AT X'001445F0'

```

Figure 37. Example CEDF output on exit from an MQINQ call (hexadecimal)

```

STATUS:  ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('.....e..s.')
001: ARG 003 ('.....')
001: ARG 004 ('.....e..s.')
001: ARG 005 ('.....')
001: ARG 006 ('.....')
001: ARG 007 ('.....')
001: ARG 008 ('.....')
001: ARG 009 ('.....')

```

Figure 38. Example CEDF output on entry to an MQINQ call (character)

```

STATUS:  COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 ('.....')
001: ARG 001 ('.....')
001: ARG 002 ('.....e..s.')
001: ARG 003 ('.....')
001: ARG 004 ('.....e..s.')
001: ARG 005 ('.....')
001: ARG 006 ('.....')
001: ARG 007 ('.....')
001: ARG 008 ('.....')
001: ARG 009 ('.....')

```

Figure 39. Example CEDF output on exit from an MQINQ call (character)

### Related reference

[MQINQ - Inquire object attributes](#)

*Example CEDF output for the MQSET call*

The parameters for this call are:

Parameter	Description
ARG 000	Connection handle
ARG 001	Object handle
ARG 002	Count of selectors

Table 15. Parameters for the MQSET call (continued)

Parameter	Description
ARG 003	Array of attribute selectors
ARG 004	Count of integer attributes
ARG 005	Integer attributes
ARG 006	Length of character attributes buffer
ARG 007	Character attributes
ARG 008	Completion code
ARG 009	Reason code

```

STATUS: ABOUT TO EXECUTE COMMAND
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'000000000000000010000007200000000') AT X'001445E0'
001: ARG 001 (X'00000001000000072000000000000000') AT X'001445E4'
001: ARG 002 (X'00000001000000020000404485ECA008') AT X'05ECAFD8'
001: ARG 003 (X'00000018000007DF000000000000000') AT X'00144C08'
001: ARG 004 (X'00000001000000020000404485ECA008') AT X'05ECAFD8'
001: ARG 005 (X'000000000000000000000000000000') AT X'00144C10'
001: ARG 006 (X'00000000000000010000000200004044') AT X'05ECAFD8'
001: ARG 007 (X'000000000000000000000000000000') AT X'00144C18'
001: ARG 008 (X'00000000000000000000000080000008') AT X'001445EC'
001: ARG 009 (X'00000000000000008000000080000060') AT X'001445F0'
    
```

Figure 40. Example CEDF output on entry to an MQSET call (hexadecimal)

```

STATUS: COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER MQM
001: ARG 000 (X'000000000000000010000007200000000') AT X'001445E0'
001: ARG 001 (X'00000001000000072000000000000000') AT X'001445E4'
001: ARG 002 (X'00000001000000020000404485ECA008') AT X'05ECAFD8'
001: ARG 003 (X'00000018000007DF000000000000000') AT X'00144C08'
001: ARG 004 (X'00000001000000020000404485ECA008') AT X'05ECAFD8'
001: ARG 005 (X'000000000000000000000000000000') AT X'00144C10'
001: ARG 006 (X'00000000000000010000000200004044') AT X'05ECAFD8'
001: ARG 007 (X'000000000000000000000000000000') AT X'00144C18'
001: ARG 008 (X'00000000000000000000000080000008') AT X'001445EC'
001: ARG 009 (X'00000000000000008000000080000060') AT X'001445F0'
    
```

Figure 41. Example CEDF output on exit from an MQSET call (hexadecimal)

```

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 (' .....-')

```

Figure 42. Example CEDF output on entry to an MQSET call (character)

```

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 (' .....-')

```

Figure 43. Example CEDF output on exit from an MQSET call (character)

#### Related reference

[MQSET - Set object attributes](#)

### Diagnostic aids for IMS

Use this topic to investigate IMS diagnostic facilities.

The application development environment is the same as for any other IMS application, and so any tools normally used in that environment can be used to develop IBM MQ applications.

Trace and dump data is available from the IMS region. See [Troubleshooting IMS](#) for more information.

### Diagnostic aids for Db2

Use this topic to investigate references for Db2 diagnostic tools.

See the [Db2 Information Center](#) for help in diagnosing Db2 problems.

### IBM MQ for z/OS dumps

Use this topic for information about the use of dumps in problem determination. It describes the steps you should take when looking at a dump produced by an IBM MQ for z/OS address space.

#### How to use dumps for problem determination

When solving problems with your IBM MQ for z/OS system, you can use dumps in two ways:

- To examine the way IBM MQ processes a request from an application program.  
To do this, you typically need to analyze the whole dump, including control blocks and the internal trace.
- To identify problems with IBM MQ for z/OS itself, under the direction of IBM support center personnel.

Use the instructions in the following topics to get and process a dump:

- [“Getting a dump with IBM MQ for z/OS” on page 247](#)
- [“Using the z/OS DUMP command” on page 248](#)
- [“Processing a dump using the IBM MQ for z/OS dump display panels” on page 250](#)
- [“Processing an IBM MQ for z/OS dump using line mode IPCS” on page 254](#)
- [“Processing an IBM MQ for z/OS dump using IPCS in batch” on page 261](#)

The dump title might provide sufficient information in the abend and reason codes to resolve the problem. You can see the dump title in the console log, or by using the z/OS command `DISPLAY DUMP, TITLE`. The format of the dump title is explained in [“Analyzing the dump and interpreting dump titles on z/OS” on page 262](#). For information about the IBM MQ for z/OS abend codes, see [“IBM MQ for z/OS abends” on page 228](#), and abend reason codes are documented in [IBM MQ for z/OS messages, completion, and reason codes](#).

If there is not enough information about your problem in the dump title, format the dump to display the other information contained in it.

See the following topics for information about different types of dumps:

- [“SYSUDUMP information on z/OS” on page 263](#)
- [“Snap dumps on z/OS” on page 264](#)
- [“SYS1.LOGREC information on z/OS” on page 265](#)
- [“SVC dumps on z/OS” on page 265](#)

### Related concepts

[“IBM MQ for z/OS abends” on page 228](#)

Abends can occur in WebSphere for z/OS or other z/OS systems. Use this topic to understand the IBM MQ system abend codes and how to investigate abends which occur in CICS, IMS, and z/OS.

[“Diagnostic information produced on IBM MQ for z/OS” on page 231](#)

Use this topic to investigate some of the diagnostic information produced by z/OS that can be useful in problem determination and understand how to investigate error messages, dumps, console logs, job output, symptom strings, and queue output.

### Related tasks

[“Tracing on z/OS” on page 405](#)

There are different trace options that can be used for problem determination with IBM MQ. Use this topic to understand the different options and how to control trace.

## **Getting a dump with IBM MQ for z/OS**

Use this topic to understand the different dump types for IBM MQ for z/OS problem determination.

The following table shows information about the types of dump used with IBM MQ for z/OS and how they are initiated. It also shows how the dump is formatted:

Dump type	Data set	Output type	Formatted by	Caused by
SVC	Defined by system	Machine readable	IPCS in conjunction with an IBM MQ for z/OS verb exit	z/OS or IBM MQ for z/OS functional recovery routine detecting error, or the operator entering the z/OS DUMP command
SYSUDUMP	Defined by JCL (SYSOUT=A)	Formatted	Normally SYSOUT=A	An abend condition (only taken if there is a SYSUDUMP DD statement for the step)



Dump type	Data set	Output type	Formatted by	Caused by
Snap	Defined by JCL CSQSNAP (SYSOUT=A)	Formatted	Normally SYSOUT=A	Unexpected MQI call errors reported to adapters, or FFST information from the channel initiator
Stand-alone	Defined by installation (tape or disk)	Machine readable	IPCS in conjunction with an IBM MQ for z/OS verb exit	Operator IPL of the stand-alone dump program

IBM MQ for z/OS recovery routines request SVC dumps for most X'5C6' abends. The exceptions are listed in “SVC dumps on z/OS” on page 265. SVC dumps issued by IBM MQ for z/OS are the primary source of diagnostic information for problems.

If the dump is initiated by the IBM MQ subsystem, information about the dump is put into area called the *summary portion*. This contains information that the dump formatting program can use to identify the key components.

See [SVC dump](#) in the *z/OS MVS Diagnosis: Tools and Service Aids* manual for more information.

### Using the z/OS DUMP command

To resolve a problem, IBM can ask you to create a dump file of the queue manager address space, channel initiator address space, or coupling facilities structures. Use this topic to understand the commands to create these dump files.

You might be asked to create dump file for any or several of the following items for IBM to resolve the problem:

- Main IBM MQ address space
- Channel initiator address space
- Coupling facility application structure
- Coupling facility administration structure for your queue sharing group

Figure 44 on page 248 through to Figure 48 on page 249 show examples of the z/OS commands to do this, assuming a subsystem name of 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
```

Figure 44. Dumping the IBM MQ queue manager and application address spaces

```

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

```

Figure 45. Dumping the IBM MQ queue manager address space

```

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

```

Figure 46. Dumping the channel initiator address space

```

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

```

Figure 47. Dumping the IBM MQ queue manager and channel initiator address spaces

```

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'

```

Figure 48. Dumping a coupling facility structure

See [DUMP command](#) for more information.

## z/OS Processing a dump using the IBM MQ for z/OS dump display panels

You can use commands available through IPCS panels to process dumps. Use this topic to understand the IPCS options.

IBM MQ for z/OS provides a set of panels to help you process dumps. The following section describes how to use these panels:

1. From the IPCS PRIMARY OPTION MENU, select **ANALYSIS - Analyze dump contents** (option 2).

The IPCS MVS ANALYSIS OF DUMP CONTENTS panel is displayed.

2. Select **COMPONENT - MVS component data** (option 6).

The IPCS MVS DUMP COMPONENT DATA ANALYSIS panel is displayed. The appearance of the panel depends on the products installed at your installation, but will be similar to the panel shown in [IPCS MVS Dump Component Data Analysis panel](#):

```
----- 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   IBM MQ dump formatter panel interface
CSQWDMP   IBM MQ dump formatter
CTRACE    Component trace summary
DAEDATA   DAE header data
DIVDATA   Data-in-virtual storage
```

Figure 49. IPCS MVS Dump Component Data Analysis panel

3. Select **CSQMAIN IBM MQ dump formatter panel interface** by typing s next to the line and pressing Enter.

If this option is not available, it is because the member CSQ7IPCS is not present; you should see [Configuring z/OS](#) for more information about installing the IBM MQ for z/OS dump formatting member.

**Note:** If you have already used the dump to do a preliminary analysis, and you want to reexamine it, select **CSQWDMP IBM MQ dump formatter** to display the formatted contents again, using the default options.

4. The IBM MQ for z/OS - DUMP ANALYSIS menu is displayed. Use this menu to specify the action that you want to perform on a system dump.

```

-----IBM 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. Before you can select a particular dump for analysis, the dump you require must be present in the dump inventory. To ensure that this is so, perform the following steps:
  - a. If you do not know the name of the data set containing the dump, specify option 1 - **Display all dump titles xx through xx**.
 

This displays the dump titles of all the dumps contained in the SYS1.DUMP data sets (where xx is a number in the range 00 through 99). You can limit the selection of data sets for display by using the xx fields to specify a range of data set numbers.

If you want to see details of all available dump data sets, set these values to 00 and 99.

Use the information displayed to identify the dump you want to analyze.
  - b. If the dump has not been copied into another data set (that is, it is in one of the SYS1.DUMP data sets), specify option 2 - **Manage the dump inventory**.
 

The dump inventory contains the dump data sets that you have used. Because the SYS1.DUMP data sets are reused, the name of the dump that you identified in step “5.a” on page 251 might be in the list displayed. However, this entry refers to the previous dump that was stored in this data set, so delete it by typing DD next to it and pressing Enter. Then press F3 to return to the DUMP ANALYSIS MENU.
6. Specify option 3 - **Select a dump**, to select the dump that you want to work with. Type the name of the data set containing the dump in the Source field, check that NOPRINT and TERMINAL are specified in the Message Routing field (this is to ensure that the output is directed to the terminal), and press Enter. Press F3 to return to the DUMP ANALYSIS MENU.
7. Having selected a dump to work with, you can now use the other options on the menu to analyze the data in different parts of the dump:
  - To display a list of all address spaces active at the time the dump was taken, select option 4.
  - To display the symptom string, select option 5.
  - To display the symptom string and other serviceability information, including the variable recording area of the system diagnostic work area (SDWA), select option 6.
  - To format and display the data contained in the in-storage LOGREC buffer, select option 7.

It could be that the abend that caused the dump was not the original cause of the error, but was caused by an earlier problem. To determine which LOGREC record relates to the cause of the problem, go to the end of the data set, type FIND ERRORID: PREV, and press Enter. The header of the latest LOGREC record is displayed, for example:

```

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
.
.
.

```

Note the program identifier (if it is not 5655R3600, the problem was not caused by IBM MQ for z/OS and you could be looking at the wrong dump). Also note the value of the TIME field. Repeat the command to find the previous LOGREC record, and note the value of the TIME field again. If the two values are close to each other (say, within about one or two tenths of a second), they could both relate to the same problem.

- To format and display the dump, select option 8. The FORMAT AND DISPLAY THE DUMP panel is displayed:

```

-----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

```

- Use this panel to format your selected system dump. You can choose to display control blocks, data produced by the internal trace, or both, which is the default.

**Note:** You cannot do this for dumps from the channel initiator, or for dumps of coupling facility structures.

- To display the whole of the dump, that is:
  - The dump title
  - The variable recording area (VRA) diagnostic information report
  - The save area trace report
  - The control block summary
  - The trace table
 select option 1.
- To display the information listed for option 1, without the trace table, select option 2.
- To display the information listed for option 1, without the control blocks, select option 3.

You can also use the following options:

– **Use the Summary Dump?**

Use this field to specify whether you want IBM MQ to use the information contained in the summary portion when formatting the selected dump. The default setting is YES.

**Note:** If a summary dump has been taken, it might include data from more than one address space.

– **Subsystem name**

Use this field to identify the subsystem with the dump data you want to display. This is only required if there is no summary data (for example, if the operator requested the dump), or if you have specified NO in the **Use the summary dump?** field.

If you do not know the subsystem name, type `IPCS SELECT ALL` at the command prompt, and press Enter to display a list of all the jobs running at the time of the error. If one of the jobs has the word ERROR against it in the SELECTION CRITERIA column, make a note of the name of that job. The job name is of the form `xxxx MSTR`, where `xxxx` is the subsystem name.

```
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
```

If no job has the word ERROR against it in the SELECTION CRITERIA column, select option 0 - DEFAULTS on the main IPCS Options Menu panel to display the IPCS Default Values panel. Note the address space identifier (ASID) and press F3 to return to the previous panel. Use the ASID to determine the job name; the form is `xxxx MSTR`, where `xxxx` is the subsystem name.

The following command shows which ASIDs are in the dump data set:

```
LDMP DSN('SYS1.DUMPxx') SELECT(DUMPED) NOSUMMARY
```

This shows the storage ranges dumped for each address space.

Press F3 to return to the FORMAT AND DISPLAY THE DUMP panel, and type this name in the **Subsystem name** field.

– **Address space identifier**

Use this field if the data in a dump comes from more than one address space. If you only want to look at data from a particular address space, specify the identifier (ASID) for that address space.

The default value for this field is ALL, which displays information about all the address spaces relevant to the subsystem in the dump. Change this field by typing the 4-character ASID over the value displayed.

**Note:** Because the dump contains storage areas common to all address spaces, the information displayed might not be relevant to your problem if you specify the address space identifier incorrectly. In this case, return to this panel, and enter the correct address space identifier.



## Related concepts

[“Processing an IBM MQ for z/OS dump using line mode IPCS” on page 254](#)

Use the IPCS commands to format a dump.

[“Processing an IBM MQ for z/OS dump using IPCS in batch” on page 261](#)

Use this topic to understand how IBM MQ for z/OS dumps can be formatted by IPCS commands in batch mode.

[“Analyzing the dump and interpreting dump titles on z/OS” on page 262](#)

Use this topic to understand how IBM MQ for z/OS dump titles are formatted, and how to analyze a dump.

## **Processing an IBM MQ for z/OS dump using line mode IPCS**

Use the IPCS commands to format a dump.

To format the dump using line mode IPCS commands, select the dump required by issuing the command:

```
SETDEF DSN('SYS1.DUMP xx')
```

(where SYS1.DUMP *xx* is the name of the data set containing the dump). You can then use IPCS subcommands to display data from the dump.

## Formatting a dump from the channel initiator on z/OS

how to format a channel initiator dump for IBM MQ for z/OS using line mode IPCS commands.

The IPCS VERBEXIT CSQXDPRD enables you to format a channel initiator dump. You can select the data that is formatted by specifying keywords.

Table 17 on page 254 describes the keywords that you can specify with CSQXDPRD.

Keyword	What is formatted
SUBSYS= <i>aaaa</i>	The control blocks of the channel initiator associated with the named subsystem. It is required for all new formatted dumps.
CHST=1, CNAM= <i>channel name</i> , DUMP=S F C	All channel information.  The optional CNAM subparameter allows you to specify the name of a channel, up to a maximum of 20 characters, for which to format details.  The optional DUMP subparameter allows you to control the extent of formatting, as follows: <ul style="list-style-type: none"><li>• Specify DUMP=S (for "short") to format the first line of the hexadecimal dump of the channel data.</li><li>• Specify DUMP=F (for "full") to format all lines of the data.</li><li>• Specify DUMP=C (for "compressed") to suppress the formatting of all duplicate lines in the data containing only 'X'00'. This is the default option</li></ul>
CHST=2, CNAM= <i>channel name</i> ,	A summary of all channels, or of the channel specified by the CNAM keyword.  See CHST=1 for details of the CNAM subparameter.
CHST=3, CNAM= <i>channel name</i> ,	Data provided by CHST=2 and a program trace, line trace and formatted semaphore table print of all channels in the dump.  See CHST=1 for details of the CNAM subparameter.

Keyword	What is formatted
CLUS=1	Cluster report including the cluster repository known on the queue manager.
CLUS=2	Cluster report showing cluster registrations.
CTRACE=S F, DPRO= nnnnnnnn, TCB= nnnnnnn	Select either a short (CTRACE=S) or full (CTRACE=F) CTRACE. The optional DPRO subparameter allows you to specify a CTRACE for the DPRO specified. The optional TCB subparameter allows you to specify a CTRACE for the job specified.
DISP=1, DUMP=S F C	Dispatcher report See CHST=1 for details of the DUMP subparameter.
BUF=1	Buffer report
XSMF=1	Format channel initiator SMF data that is available in a dump.

### Related concepts

[“Processing a dump using the IBM MQ for z/OS dump display panels” on page 250](#)

You can use commands available through IPCS panels to process dumps. Use this topic to understand the IPCS options.

[“Processing an IBM MQ for z/OS dump using IPCS in batch” on page 261](#)

Use this topic to understand how IBM MQ for z/OS dumps can be formatted by IPCS commands in batch mode.

[“Analyzing the dump and interpreting dump titles on z/OS” on page 262](#)

Use this topic to understand how IBM MQ for z/OS dump titles are formatted, and how to analyze a dump.

[“Formatting an IBM MQ for z/OS dump” on page 255](#)

Use this topic to understand how to format a queue manager dump using line mode IPCS commands.

### *Formatting an IBM MQ for z/OS dump*

Use this topic to understand how to format a queue manager dump using line mode IPCS commands.

The IPCS VERBEXIT CSQWDMP invokes the IBM MQ for z/OS dump formatting program (CSQWDPRD), and enables you to format an SVC dump to display IBM MQ data. You can restrict the amount of data that is displayed by specifying parameters.

IBM Service Personnel might require dumps of your coupling facility administration structure and application structures for your queue sharing group, with dumps of queue managers in the queue sharing group, to aid problem diagnosis. For information on formatting a coupling facility list structure, and the STRDATA subcommand, see the [z/OS MVS IPCS Commands](#) manual.

**Note:** This section describes the parameters required to extract the necessary data. Separate operands by commas, not blanks. A blank that follows any operand in the control statement terminates the operand list, and any subsequent operands are ignored. [Table 18 on page 255](#) explains each keyword that you can specify in the control statement for formatting dumps.

Keyword	Description
SUBSYS= <i>aaaa</i>	Use this keyword if the summary dump portion is not available, or not to be used, to give the name of the subsystem to format information for. <i>aaaa</i> is a 1 through 4-character subsystem name.

Keyword	Description
ALL (default)	All control blocks and the trace table.
AA	Data is displayed for all IBM MQ for z/OS control blocks in all address spaces.
DIAG=Y	Print diagnostic information. Use only under guidance from IBM service personnel. DIAG=N (suppresses the formatting of diagnostic information) is the default.
EB= <i>nnnnnnnn</i>	Only the trace points associated with this EB thread are displayed (the format of this keyword is EB= <i>nnnnnnnn</i> where <i>nnnnnnnn</i> is the 8-digit address of an EB thread that is contained in the trace). You must use this in conjunction with the TT keyword.
LG	All control blocks.
PTF=Y, LOAD= <i>load module name</i>	A list of PTFs at the front of the report (from MEPL). PTF=N (suppresses the formatting of such a list) is the default.  The optional load subparameter allows you to specify the name of a load module, up to a maximum of 8 characters, for which to format a PTF report.
SA= <i>hhhh</i>	The control blocks for a specified address space. Use either of the following formats: <ul style="list-style-type: none"> <li>SA= <i>hh</i> or</li> <li>SA= <i>hhhh</i></li> </ul> where <i>h</i> represents a hexadecimal digit.
SG	A subset of system-wide control blocks.
TT ,HANDLES=x ,LOCKS=x ,INSYNCS=x ,URINFO=ALL/LONG	Format trace table  Indicate threads with greater than x handles  Indicate threads with greater than x locks  Indicate threads with greater than x insync operations  Show UR info for ALL threads or for long-running threads

Table 19 on page 256 details the dump formatting keywords that you can use to format the data relating to individual resource managers.

You cannot use these keywords in conjunction with any of the keywords in [Table 18 on page 255](#).

Keyword	What is formatted
BMC=1 BMC=2( <i>buffer pool number</i> ) BMC=3( <i>xx/yyyyyy</i> ) BMC=4( <i>xx/yyyyyy</i> )	Buffer manager data. BMC=1 formats control blocks of all buffers.  BMC=2 formats data relating to the buffer identified in the 2-digit <i>buffer pool number</i> .  BMC=3 and BMC=4 display a page from a page set, if the page is present in a buffer. (The difference between BMC=3 and BMC=4 is the route taken to the page.)
BUFL= <i>nnnnnnnnnnnn</i>	Storage access buffer allocation sz.

Table 19. Resource manager dump formatting keywords (continued)

<b>Keyword</b>	<b>What is formatted</b>
CALLD=Y =W	Show arrow for call depth in TT. and indent trace entry.
CALLTIME=Y	Print call time on exit trace.
CB=(addr/[strmodel])	Format address as IBM MQ block.
CBF=1	CBF report level 1.
CCB=S	Show the Composite Capability Block (CCB) for system EBs in TT.
CFS=1	CFS report level 1.
CFS=2	CFS report level 2.
CHLAUTH=1/2 ONAM=20 chars	CHLAUTH report level. The optional ONAM subparameter allows you to specify the object name, up to a maximum of 20 characters, to limit data printed to objects starting with characters in ONAM.
CLUS=1	Cluster report including the cluster repository known on the queue manager.
CLUS=2	Cluster report showing cluster registrations.
CLXQ=1	Cluster XMITQ report level 1.
CLXQ=2 ONAM=20 chars	Cluster XMITQ report level 2. The optional ONAM subparameter allows you to specify the object name, up to a maximum of 20 characters, to limit data printed to objects starting with characters in ONAM.
CMD=0/1/2	Command trace table display level.
D=1/2/3	Detail level for some reports.
Db2=1	Db2 report level 1.
DMC=1, ONAM=48 chars	DMC report level 1. The optional ONAM subparameter allows you to specify the object name, up to a maximum of 48 characters, to limit data printed to objects starting with characters in ONAM.
DMC=2, ONAM=48 chars	DMC report level 2. The optional ONAM subparameter allows you to limit the objects printed to those with names beginning with the characters specified in ONAM (up to a maximum of 48 characters).
DMC=3, ONAM=48 chars	DMC report level 3. The optional ONAM subparameter allows you to limit the objects printed to those with names beginning with the characters specified in ONAM (up to a maximum of 48 characters).
GR=1	Group indoubt report level 1.
IMS=1	IMS report level 1

Table 20. Resource manager dump formatting keywords (J -P)

Keyword	What is formatted
JOBNAME= xxxxxxxx	Job name
LKM=1	LKM report level 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 report level 2/3. Name (character) Name (Hex) Namespace Lock type Lock qualification LKM report level 3 LKM report level 4
LMC=1	LMC report level 1.
MAXTR= nnnnnnnnn	Max trace entries to format
MHASID= xxxx	Message handle ASID for properties
MMC=1 OBJ=MQLO/MQSH/MQRO/ MQAO/MQMO/MCHL/ MNLS/MSTC/MPRC/ : ' MAUT ONAM	MMC report level 1  Object type The optional ONAM subparameter allows you to limit the objects printed to those with names beginning with the characters specified in ONAM (up to a maximum of 48 characters).
MMC=2 ONAM=48 chars	MMC report level 2 The optional ONAM subparameter allows you to limit the objects printed to those with names beginning with the characters specified in ONAM (up to a maximum of 48 characters).
MSG=nnnnnnnnnnnnnnnn MASID=xxxx LEN=xxxxxxx MSGD=S/D	Format the message at pointer. MASID allows storage in other address spaces. LEN limits amount of storage to format. MSGD controls level of detail.
MSGD=S/D	Message details in DMC=3, BMC=3/4, PSID reports. The parameter controls level of details, S is summary and D is detailed.
MSGH = nnnnnnnnnnnnnnn	Message handle

<i>Table 20. Resource manager dump formatting keywords (J -P) (continued)</i>	
<b>Keyword</b>	<b>What is formatted</b>
MT	Message properties trace
MQVCX	MQCHARVs in hexadecimal format
PROPS= <i>nnnnnnnnnnnnnnnn</i>	Message properties pointer
PSID= <i>nnnnnnnn</i>	Page set to format page
PSTRX	Properties strings in hex format

<i>Table 21. Resource manager dump formatting keywords (R -Z)</i>	
<b>Keyword</b>	<b>What is formatted</b>
RPR= <i>nnnnnnnn</i>	Page or record to format
SHOWDEL	Show deleted records for DMC=3
SMC=1/2/3	Storage manager
TC= * A E O	TT data char format, concatenated print all in suitable character set always print ASCII always print EBCDIC never print either
TFMT=H/M	Time format - human or STCK
THR= <i>nnnnnnnn</i>	Thread address
THR=*/2/3	Set thread report level
TOP=1	TOP report level 1
TOP=2	TOP report level 2
TOP= <i>nnnnnnnnnnnnnnnn</i> /TSTR=48 chars /TSTRX=hex 1208 str	Tnode 64bit address or Topic string (wildcard with % at start or end) <sup>1</sup> This will be converted EBCDIC to ASCII, but only invariant characters Hexadecimal of topic string in 1208 always wildcard character at start.
TOP=3	TOP report level 3
TOP=4	TOP report level 4
TSEG=M(RU)/Q(P64) I(INTERPOLATE) F(WD) D(EBUG)	Search process for 64-bit trace Guess missing TSEG address or addresses Force forward sort Debug search process
TSEG=(M,Q,I,F,D)	Specify multiple TSEG options
W=0/1/2/3	TT width format
XA=1	XA report level 1
ZMH = <i>nnnnnnnnnnnnnnnn</i>	ZST message handle



If the dump is initiated by the operator, there is no information in the summary portion of the dump. [Table 22 on page 260](#) shows additional keywords that you can use in the CSQWDMP control statement.

Keyword	Description
SUBSYS= <i>aaaa</i>	Use this keyword if the summary dump portion is not available, or not to be used, to give the name of the subsystem to format information for. <i>aaaa</i> is a 1 through 4-character subsystem name.
SUMDUMP=NO	Use this keyword if the dump has a summary portion, but you do not want to use it. (You would usually only do this if so directed by your IBM support center.)

The following list shows some examples of how to use these keywords:

- For default formatting of all address spaces, using information from the summary portion of the dump, use:

```
VERBX CSQWDMP
```

- To display the trace table from a dump of subsystem named MQMT, which was initiated by an operator (and so does not have a summary portion) use:

```
VERBX CSQWDMP 'TT,SUBSYS=MQMT'
```

- To display all the control blocks and the trace table from a dump produced by a subsystem abend, for an address space with ASID (address space identifier) 1F, use:

```
VERBX CSQWDMP 'TT,LG,SA=1F'
```

- To display the portion of the trace table from a dump associated with a particular EB thread, use:

```
VERBX CSQWDMP 'TT,EB= nnnnnnnn '
```

- To display message manager 1 report for local non-shared queue objects with a name begins with 'ABC' use:

```
VERBX CSQWDMP 'MMC=1,ONAM=ABC,Obj=MQLO'
```

[Table 23 on page 260](#) shows some other commands that are used frequently for analyzing dumps. For more information about these sub commands, see the [z/OS MVS IPCS Commands](#) manual.

Subcommand	Description
STATUS	To display data usually examined during the initial part of the problem determination process.
STRDATA LISTNUM(ALL) ENTRYPOS(ALL) DETAIL	To format coupling facility structure data.
VERBEXIT LOGDATA	To format the in-storage LOGREC buffer records present before the dump was taken. LOGDATA locates the LOGREC entries that are contained in the LOGREC recording buffer and invokes the EREP program to format and print the LOGREC entries. These entries are formatted in the style of the normal detail edit report.

Table 23. IPCS subcommands used for dump analysis (continued)

Subcommand	Description
VERBEXIT TRACE	To format the system trace entries for all address spaces.
VERBEXIT SYMPTOM	To format the symptom strings contained in the header record of a system dump such as stand-alone dump, SVC dump, or an abend dump requested with a SYSUDUMP DD statement.
VERBEXIT GRSTRACE	To format diagnostic data from the major control blocks for global resource serialization.
VERBEXIT SUMDUMP	To locate and display the summary dump data that an SVC dump provides.
VERBEXIT DAEDATA	To format the dump analysis and elimination (DAE) data for the dumped system.

### Related information

[“Formatting a dump from the channel initiator on z/OS” on page 254](#)

### **Processing an IBM MQ for z/OS dump using IPCS in batch**

Use this topic to understand how IBM MQ for z/OS dumps can be formatted by IPCS commands in batch mode.

To use IPCS in batch, insert the required IPCS statements into your batch job stream (see [Figure 50 on page 261](#)).

Change the data set name (DSN=) on the DUMP00 statement to reflect the dump you want to process, and insert the IPCS subcommands that you want to use.

```

//*****
//*  RUNNING IPCS IN A BATCH JOB          *
//*****
//MQMDMP EXEC PGM=IKJEFT01,REGION=5120K
//STEPLIB DD DSN=mqm.library-name,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//IPCSPRINT 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
/*

```

Figure 50. Sample JCL for printing dumps through IPCS in the z/OS environment

### Related concepts

[“Processing a dump using the IBM MQ for z/OS dump display panels” on page 250](#)

You can use commands available through IPCS panels to process dumps. Use this topic to understand the IPCS options.

[“Processing an IBM MQ for z/OS dump using line mode IPCS” on page 254](#)

Use the IPCS commands to format a dump.

[“Analyzing the dump and interpreting dump titles on z/OS” on page 262](#)

Use this topic to understand how IBM MQ for z/OS dump titles are formatted, and how to analyze a dump.

## **Analyzing the dump and interpreting dump titles on z/OS**

Use this topic to understand how IBM MQ for z/OS dump titles are formatted, and how to analyze a dump.

- [Analyzing the dump](#)
- [Dump title variation with PSW and ASID](#)

### Analyzing the dump

The dump title includes the abend completion and reason codes, the failing load module and CSECT names, and the release identifier. For more information on the dump title see [Dump title variation with PSW and ASID](#)

The formats of SVC dump titles vary slightly, depending on the type of error.

Figure 51 on page 262 shows an example of an SVC dump title. Each field in the title is described after the figure.

```
ssnm, ABN=5C6-00D303F2, U=AUSER, C=R3600. 710. LOCK-CSQL1GET,  
M=CSQGFRCV, LOC=CSQLPLM. CSQL1GET+0246
```

*Figure 51. Sample SVC dump title*

#### **ssnm, ABN=compltn-reason**

- `ssnm` is the name of the subsystem that issued the dump.
- `compltn` is the 3-character hexadecimal abend completion code (in this example, X'5C6'), prefixed by U for user abend codes.
- `reason` is the 4-byte hexadecimal reason code (in this example, X'00D303F2').

**Note:** The abend and reason codes might provide sufficient information to resolve the problem. See the [IBM MQ for z/OS messages, completion, and reason codes](#) for an explanation of the reason code.

#### **U=userid**

- `userid` is the user identifier of the user (in this example, AUSER). This field is not present for channel initiators.

#### **C=compid.release.comp-function**

- `compid` is the last 5 characters of the component identifier. The value R3600 uniquely identifies IBM MQ for z/OS.
- `release` is a 3-digit code indicating the version, release, and modification level of IBM MQ for z/OS (in this example, 710).
- `comp` is an acronym for the component in control at the time of the abend (in this example, LOCK).
- `function` is the name of a function, macro, or routine in control at the time of abend (in this example, CSQL1GET). This field is not always present.

#### **M=module**

- `module` is the name of the FRR or ESTAE recovery routine (in this example, CSQGFRCV). This field is not always present.

**Note:** This is not the name of the module where the abend occurred; that is given by LOC.

#### **LOC=loadmod.csect+csect\_offset**

- `loadmod` is the name of the load module in control at the time of the abend (in this example, CSQLPLM). This might be represented by an asterisk if it is unknown.

- `csect` is the name of the CSECT in control at the time of abend (in this example, `CSQL1GET`).
- `csect_offset` is the offset within the failing CSECT at the time of abend (in this example, `0246`).

**Note:** The value of `csect_offset` might vary if service has been applied to this CSECT, so do not use this value when building a keyword string to search the IBM software support database.

## Dump title variation with PSW and ASID

Some dump titles replace the load module name, CSECT name, and CSECT offset with the PSW (program status word) and ASID (address space identifier). [Figure 52 on page 263](#) illustrates this format.

```
ssnm,ABN=compltn-reason,U=userid,C=compid.release.comp-function,
M=module,PSW=psw_contents,ASID=address_space_id
```

*Figure 52. Dump title with PSW and ASID*

### psw\_contents

- The PSW at the time of the error (for example, `X'077C100000729F9C'`).

### address\_space\_id

- The address space in control at the time of the abend (for example, `X'0011'`). This field is not present for a channel initiator.

### Related concepts

[“Processing a dump using the IBM MQ for z/OS dump display panels” on page 250](#)

You can use commands available through IPCS panels to process dumps. Use this topic to understand the IPCS options.

[“Processing an IBM MQ for z/OS dump using line mode IPCS” on page 254](#)

Use the IPCS commands to format a dump.

[“Processing an IBM MQ for z/OS dump using IPCS in batch” on page 261](#)

Use this topic to understand how IBM MQ for z/OS dumps can be formatted by IPCS commands in batch mode.

## **SYSUDUMP information on z/OS**

The z/OS system can create SYSUDUMPs, which can be used as part of problem determination. This topic shows a sample SYSUDUMP output and gives a reference to the tools for interpreting SYSUDUMPs.

SYSUDUMP dumps provide information useful for debugging batch and TSO application programs. For more information about SYSUDUMP dumps, see the [z/OS MVS Diagnosis: Tools and Service Aids](#) manual.

[Figure 53 on page 264](#) shows a sample of the beginning of a SYSUDUMP dump.

```
JOB MQMBXBA1 STEP TSOUSER 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
```

Figure 53. Sample beginning of a SYSUDUMP

## Snap dumps on z/OS

Snap dump data sets are controlled by z/OS JCL command statements. Use this topic to understand the CSQSNAP DD statement.

Snap dumps are always sent to the data set defined by the CSQSNAP DD statement. They can be issued by the adapters or the channel initiator.

- Snap dumps are issued by the batch, CICS, IMS, or RRS adapter when an unexpected error is returned by the queue manager for an MQI call. A full dump is produced containing information about the program that caused the problem.

For a snap dump to be produced, the CSQSNAP DD statement must be in the batch application JCL, CICS JCL, or IMS dependent region JCL.

- Snap dumps are issued by the channel initiator in specific error conditions instead of a system dump. The dump contains information relating to the error. Message CSQX053E is also issued at the same time.

To produce a snap dump, the CSQSNAP DD statement must be in the channel initiator started-task procedure.

## **SYS1.LOGREC information on z/OS**

Use this topic to understand how the z/OS SYS1.LOGREC information can assist with problem determination.

### **IBM MQ for z/OS and SYS1.LOGREC**

The SYS1.LOGREC data set records various errors that different components of the operating system encounter. For more information about using SYS1.LOGREC records, see the [z/OS MVS Diagnosis: Tools and Service Aids](#) manual.

IBM MQ for z/OS recovery routines write information in the *system diagnostic work area* (SDWA) to the SYS1.LOGREC data set when retry is attempted, or when percolation to the next recovery routine occurs. Multiple SYS1.LOGREC entries can be recorded, because two or more retries or percolations might occur for a single error.

The SYS1.LOGREC entries recorded near the time of abend might provide valuable historical information about the events leading up to the abend.

### **Finding the applicable SYS1.LOGREC information**

To obtain a SYS1.LOGREC listing, either:

- Use the [EREP](#) service aid to format records in the SYS1.LOGREC data set.

Following is some example JCL to get software LOGREC report:

```
//STEP EXEC PGM=IFCEREPI,PARM='CARD'  
//ACCIN DD DSN=EREP.HISTORY,DISP=(OLD,PASS)  
//DIRECTWK DD UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)  
//EREPPT DD SYSOUT=A,DCB=BLKSIZE=133  
//TOURIST DD SYSOUT=A,DCB=BLKSIZE=133  
//SYSIN DD *  
PRINT=PS  
HIST ****don't use this line if record is in SYS1.LOGREC****  
ACC=N  
TYPE=S  
ENDPARM  
/*
```

#### **Notes:**

- PRINT=PS is for Detail and Summary, TYPE=S is for software.  
You can also add a time range like TIME=(2000-2005) to get only the records you need.
- AS for TIME, data can be extracted using DATE as well. See [EREP Selection Parameters, Parameter descriptions](#), and [Syntax rules](#) for information on how you set the selection of the date range.
- Specify the VERBEXIT LOGDATA keyword in IPCS.
- Use option 7 on the DUMP ANALYSIS MENU (refer to [“Processing a dump using the IBM MQ for z/OS dump display panels”](#) on page 250).

Only records available in storage when the dump was requested are included. Each formatted record follows the heading \*\*\*\*\*LOGDATA\*\*\*\*\*.

## **SVC dumps on z/OS**

Use this topic to understand how to suppress SVC dumps on z/OS, and reasons why SVC dumps are not produced.

### **When SVC dumps are not produced**

Under some circumstances, SVC dumps are not produced. Generally, dumps are suppressed because of time or space problems, or security violations. The following list summarizes other reasons why SVC dumps might not be produced:



- The *z/OS serviceability level indication processing (SLIP)* commands suppressed the abend.  
The description of `IEACMD00` in the *z/OS MVS Initialization and Tuning Reference* manual lists the defaults for SLIP commands executed at IPL time.
- The abend reason code was one that does not require a dump to determine the cause of abend.
- `SDWACOMU` or `SDWAEAS` (part of the system diagnostic work area, SDWA) was used to suppress the dump.

## Suppressing IBM MQ for z/OS dumps using z/OS DAE

You can suppress SVC dumps that duplicate previous dumps. The *z/OS MVS Diagnosis: Tools and Service Aids* manual gives details about using *z/OS dump analysis and elimination (DAE)*.

To support DAE, IBM MQ for z/OS defines two *variable recording area (VRA)* keys and a minimum symptom string. The two VRA keys are:

- KEY VRADAE (X'53'). No data is associated with this key.
- KEY VRAMINSC (X'52') DATA (X'08')

IBM MQ for z/OS provides the following data for the minimum symptom string in the *system diagnostic work area (SDWA)*:

- Load module name
- CSECT name
- Abend code
- Recovery routine name
- Failing instruction area
- REG/PSW difference
- Reason code
- Component identifier
- Component subfunction

Dumps are considered duplicates for the purpose of suppressing duplicate dumps if eight (the X'08' from the VRAMINSC key) of the nine symptoms are the same.

## Dealing with performance problems on z/OS

Use this topic to investigate IBM MQ for z/OS performance problems in more detail.

Performance problems are characterized by the following:

- Poor response times in online transactions
- Batch jobs taking a long time to complete
- The transmission of messages is slow

Performance problems can be caused by many factors, from a lack of resource in the z/OS system as a whole, to poor application design.

Remote queuing problems can be due to network congestion and other network problems. They can also be caused by problems at the remote queue manager.

## IBM MQ for z/OS system considerations

The z/OS system is an area that requires examination when investigating performance problems.

You might already be aware that your z/OS system is under stress because these problems affect many subsystems and applications.

You can use the standard monitoring tools such as Resource Monitoring Facility (RMF) to monitor and diagnose these problems. They might include:

- Constraints on storage (paging)
- Constraints on processor cycles
- Constraints on DASD
- Channel path usage

Use normal z/OS tuning techniques to resolve these problems.

## CICS constraints

CICS constraints can also have an adverse effect on IBM MQ for z/OS performance. Use this information for further details about CICS constraints.

Performance of IBM MQ tasks can be affected by CICS constraints. For example, your system might have reached MAXTASK, forcing transactions to wait, or the CICS system might be short on storage. For example, CICS might not be scheduling transactions because the number of concurrent tasks has been reached, or CICS has detected a resource problem. If you suspect that CICS is causing your performance problems (for example because batch and TSO jobs run successfully, but your CICS tasks time out, or have poor response times), see the *CICS Problem Determination Guide* and the *CICS Performance Guide*.

**Note:** CICS I/O to transient data extrapartition data sets uses the z/OS RESERVE command. This could affect I/O to other data sets on the same volume.

### Related concepts

[“Dealing with incorrect output on z/OS” on page 272](#)

Incorrect output can be missing, unexpected, or corrupted information.

[“Dealing with applications that are running slowly or have stopped on z/OS” on page 267](#)

Waits and loops can exhibit similar symptoms. Use the links in this topic to help differentiate between waits and loops on z/OS.

### Related tasks

[“进行初始检查” on page 6](#)

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。

## **Dealing with applications that are running slowly or have stopped on z/OS**

Waits and loops can exhibit similar symptoms. Use the links in this topic to help differentiate between waits and loops on z/OS.

Waits and loops are characterized by unresponsiveness. However, it can be difficult to distinguish between waits, loops, and poor performance.

Any of the following symptoms might be caused by a wait or a loop, or by a badly tuned or overloaded system:

- An application that appears to have stopped running (if IBM MQ for z/OS is still responsive, this problem is probably caused by an application problem)
- An MQSC command that does not produce a response
- Excessive use of processor time

## Distinguishing between waits and loops on z/OS

Waits and loops on IBM MQ for z/OS can present similar symptoms. Use this information to help determine if you are experiencing a wait or a loop.

Because waits and loops can be difficult to distinguish, in some cases you need to carry out a detailed investigation before deciding which classification is appropriate for your problem. The following information gives you guidance about choosing the best classification, and advice on what to do when you have decided on a classification.

## Waits

For problem determination, a wait state is regarded as the state in which the execution of a task has been suspended. That is, the task has started to run, but has been suspended without completing, and has subsequently been unable to resume.

A problem identified as a wait in your system could be caused by any of the following:

- A wait on an MQI call
- A wait on a CICS or IMS call
- A wait for another resource (for example, file I/O)
- An ECB wait
- The CICS or IMS region waiting
- TSO waiting
- IBM MQ for z/OS waiting for work
- An apparent wait, caused by a loop
- Your task is not being dispatched by CICS or MVS due to higher priority work
- Db2 or RRS are inactive

## Loops

A loop is the repeated execution of some code. If you have not planned the loop, or if you have designed it into your application but it does not terminate for some reason, you get a set of symptoms that vary depending on what the code is doing, and how any interfacing components and products react to it. In some cases, at first, a loop might be diagnosed as a wait or performance problem, because the looping task competes for system resources with other tasks that are not involved in the loop. However, a loop consumes resources but a wait does not.

An apparent loop problem in your system could be caused by any of the following:

- An application doing a lot more processing than usual and therefore taking much longer to complete
- A loop in application logic
- A loop with MQI calls
- A loop with CICS or IMS calls
- A loop in CICS or IMS code
- A loop in IBM MQ for z/OS

## Symptoms of waits and loops

Any of the following symptoms could be caused by a wait, a loop, or by a badly tuned or overloaded system:

- Timeouts on MQGET WAITs
- Batch jobs suspended
- TSO session suspended
- CICS task suspended
- Transactions not being started because of resource constraints, for example CICS MAX task
- Queues becoming full, and not being processed
- System commands not accepted, or producing no response

To perform the tests shown in these topics, you need access to the z/OS console, and to be able to issue operator commands.

- [“Dealing with waits on z/OS” on page 269](#)

- [“Dealing with loops on z/OS” on page 271](#)

## Related tasks

[“进行初始检查” on page 6](#)

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。

### *Dealing with waits on z/OS*

Waits can occur in batch or TSO applications, CICS transactions, and other components on IBM MQ for z/OS. Use this topic to determine where waits can occur.

When investigating what appears to be a problem with tasks or subsystems waiting, it is necessary to take into account the environment in which the task or subsystem is running.

It might be that your z/OS system is generally under stress. In this case, there can be many symptoms. If there is not enough real storage, jobs experience waits at paging interrupts or swap-outs. Input/output (I/O) contention or high channel usage can also cause waits.

You can use standard monitoring tools, such as Resource Monitoring Facility (RMF) to diagnose such problems. Use normal z/OS tuning techniques to resolve them.

## Is a batch or TSO program waiting?

Consider the following points:

### **Your program might be waiting on another resource**

For example, a VSAM control interval (CI) that another program is holding for update.

### **Your program might be waiting for a message that has not yet arrived**

This condition might be normal behavior if, for example, it is a server program that constantly monitors a queue.

Alternatively, your program might be waiting for a message that has arrived, but has not yet been committed.

Issue the **DIS CONN(\*) TYPE(HANDLE)** command and examine the queues in use by your program.

If you suspect that your program has issued an MQI call that did not involve an MQGET WAIT, and control has not returned from IBM MQ, take an SVC dump of both the batch or TSO job, and the IBM MQ subsystem before canceling the batch or TSO program.

Also consider that the wait state might be the result of a problem with another program, such as an abnormal termination (see [“Messages do not arrive when expected on z/OS” on page 273](#)), or in IBM MQ itself (see [“Is IBM MQ waiting for z/OS ?” on page 270](#)). See [“IBM MQ for z/OS dumps” on page 246](#) (specifically [Figure 44 on page 248](#)) for information about obtaining a dump.

If the problem persists, see [“联系 IBM 支持人员” on page 279](#) for information about reporting the problem to IBM.

## Is a CICS transaction waiting?

Consider the following points:

### **CICS might be under stress**

This might indicate that the maximum number of tasks allowed (**MAXTASK**) has been reached, or a short on storage (SOS) condition exists. Check the console log for messages that might explain this (for example, SOS messages), or see the *CICS Problem Determination Guide*.

### **The transaction might be waiting for another resource**

For example, this might be file I/O. You can use CEMT INQ TASK to see what the task is waiting for. If the resource type is MQSERIES your transaction is waiting on IBM MQ (either in an MQGET WAIT or a task switch). Otherwise see the *CICS Problem Determination Guide* to determine the reason for the wait.

### The transaction might be waiting for IBM MQ for z/OS

This might be normal, for example, if your program is a server program that waits for messages to arrive on a queue. Otherwise it might be the result of a transaction abend, for example (see [“Messages do not arrive when expected on z/OS” on page 273](#)). If so, the abend is reported in the CSMT log.

### The transaction might be waiting for a remote message

If you are using distributed queuing, the program might be waiting for a message that has not yet been delivered from a remote system (for further information, refer to [“Problems with missing messages when using distributed queuing on z/OS” on page 275](#)).

If you suspect that your program has issued an MQI call that did not involve an MQGET WAIT (that is, it is in a task switch), and control has not returned from IBM MQ, take an SVC dump of both the CICS region, and the IBM MQ subsystem before canceling the CICS transaction. See [“Dealing with loops on z/OS” on page 271](#) for information about waits. Refer to [“IBM MQ for z/OS dumps” on page 246](#) (specifically [Figure 44 on page 248](#)) for information about obtaining a dump.

If the problem persists, refer to [“联系 IBM 支持人员” on page 279](#) for information about reporting the problem to IBM.

### Is Db2 waiting?

If your investigations indicate that Db2 is waiting, check the following:

1. Use the Db2 **-DISPLAY THREAD(\*)** command to determine if any activity is taking place between the queue manager and the Db2 subsystem.
2. Try and determine whether any waits are local to the queue manager subsystems or are across the Db2 subsystems.

### Is RRS active?

- Use the **D RRS** command to determine if RRS is active.

### Is IBM MQ waiting for z/OS ?

If your investigations indicate that IBM MQ itself is waiting, check the following:

1. Use the **DISPLAY THREAD(\*)** command to check if anything is connected to IBM MQ.
2. Use SDSF DA, or the z/OS command **DISPLAY A,xxxxMSTR** to determine whether there is any processor usage as shown in [Has your application or IBM MQ for z/OS stopped processing work? in “Examining the problem in greater depth on z/OS” on page 50](#).
  - If IBM MQ is using some processor time, reconsider other reasons why IBM MQ might be waiting, or consider whether this is actually a performance problem.
  - If there is no processor activity, check whether IBM MQ responds to commands. If you can get a response, reconsider other reasons why IBM MQ might be waiting.
  - If you cannot get a response, check the console log for messages that might explain the wait (for example, IBM MQ might have run out of active log data sets, and be waiting for offload processing).

If you are satisfied that IBM MQ has stalled, use the **STOP QMGR** command in both **QUIESCE** and **FORCE** mode to terminate any programs currently being executed.

If the **STOP QMGR** command fails to respond, cancel the queue manager with a dump, and restart. If the problem recurs, see [“联系 IBM 支持人员” on page 279](#) for further guidance.

### Related concepts

[“Dealing with loops on z/OS” on page 271](#)

Loops can occur in different areas of a z/OS system. Use this topic to help determine where a loop is occurring.

## Related information

[“Distinguishing between waits and loops on z/OS” on page 267](#)

### *Dealing with loops on z/OS*

Loops can occur in different areas of a z/OS system. Use this topic to help determine where a loop is occurring.

The following sections describe the various types of loop that you might encounter, and suggest some responses.

## Is a batch application looping?

If you suspect that a batch or TSO application is looping, use the console to issue the z/OS command `DISPLAY JOBS, A` (for a batch application) or `DISPLAY TS, A` (for a TSO application). Note the CT values from the data displayed, and repeat the command.

If any task shows a significant increase in the CT value, it might be that the task is looping. You could also use `SDSF DA`, which shows you the percentage of processor that each address space is using.

## Is a batch job producing a large amount of output?

An example of this behavior might be an application that browses a queue and prints the messages. If the browse operation has been started with `BROWSE FIRST`, and subsequent calls have not been reset to `BROWSE NEXT`, the application browses, and prints the first message on the queue repeatedly.

You can use `SDSF DA` to look at the output of running jobs if you suspect that it might be causing a problem.

## Does a CICS region show heavy processor activity?

It might be that a CICS application is looping, or that the CICS region itself is in a loop. You might see AICA abends if a transaction goes into a tight (unyielding) loop.

If you suspect that CICS, or a CICS application is looping, see the *CICS Problem Determination Guide*.

## Does an IMS region show heavy processor activity?

It might be that an IMS application is looping. If you suspect this behavior, see *IMS Diagnosis Guide and Reference I*.

## Is the queue manager showing heavy processor activity?

Try to enter an `MQSC DISPLAY` command from the console. If you get no response, it is possible that the queue manager is looping. Follow the procedure shown in [Has your application or IBM MQ for z/OS stopped processing work?](#) in [“Examining the problem in greater depth on z/OS” on page 50](#) to display information about the processor time being used by the queue manager. If this command indicates that the queue manager is in a loop, take a memory dump, cancel the queue manager and restart.

If the problem persists, see [“联系 IBM 支持人员” on page 279](#) for information about reporting the problem to IBM.

## Is a queue, page set, or Coupling Facility structure filling up unexpectedly?

If so, it might indicate that an application is looping, and putting messages on to a queue. (It might be a batch, CICS, or TSO application.)

### Identifying a looping application

In a busy system, it might be difficult to identify which application is causing the problem. If you keep a cross-reference of applications to queues, terminate any programs or transactions that might be



putting messages on to the queue. Investigate these programs or transactions before using them again. (The most likely culprits are new, or changed applications; check your change log to identify them.)

Try issuing a **DISPLAY QSTATUS** command on the queue. This command returns information about the queue that might help to identify which application is looping.

### **Incorrect triggering definitions**

It might be that a getting application has not been triggered because of incorrect object definitions, for example, the queue might be set to NOTRIGGER.

### **Distributed queuing**

Using distributed queuing, a symptom of this problem might be a message in the receiving system indicating that MQPUT calls to the dead-letter queue are failing. This problem might be caused because the dead-letter queue has also filled up. The dead-letter queue message header (dead-letter header structure) contains a reason or feedback code explaining why the message might not be put on to the target queue. See [MQDLH - Dead-letter header](#) for information about the dead-letter header structure.

### **Allocation of queues to page sets**

If a particular page set frequently fills up, there might be a problem with the allocation of queues to page sets. See [IBM MQ for z/OS performance constraints](#) for more information.

### **Shared queues**

Is the Coupling Facility structure full? The z/OS command DISPLAY CF displays information about Coupling Facility storage including the total amount, the total in use, and the total free control and non-control storage. The RMF Coupling Facility Usage Summary Report provides a more permanent copy of this information.

## **Are a task, and IBM MQ for z/OS, showing heavy processor activity?**

In this case, a task might be looping on MQI calls (for example, browsing the same message repeatedly).

### **Related concepts**

[“Dealing with waits on z/OS” on page 269](#)

Waits can occur in batch or TSO applications, CICS transactions, and other components on IBM MQ for z/OS. Use this topic to determine where waits can occur.

### **Related information**

[“Distinguishing between waits and loops on z/OS” on page 267](#)

## **z/OS Dealing with incorrect output on z/OS**

Incorrect output can be missing, unexpected, or corrupted information.

The term "incorrect output" can be interpreted in many different ways. Points to consider if you have obtained what you believe to be some incorrect output are:

- When to classify output as incorrect

"Incorrect output" might be regarded as any output that you were not expecting. However, use this term with care in the context of problem determination because it might be a secondary effect of some other type of error. For example, looping could be occurring if you get any repetitive output, even though that output is what you expected.

- Error messages

IBM MQ also responds to many errors it detects by sending error messages. You might regard these messages as "incorrect output", but they are only symptoms of another type of problem. If you have received an error message from IBM MQ that you were not expecting, see [Are there any error messages, return codes or other error conditions?](#) in [“Identifying characteristics of the problem on z/OS” on page 45](#).

- Unexpected messages

Your application might not have received a message that it was expecting, or has received a message containing unexpected or corrupted information, or has received a message that it was not expecting (for example, one that was destined for a different application).

The subtopics contain information about the problems that you could encounter with your system and classify as incorrect output:

- Application messages that do not arrive when you are expecting them
- Application messages that contain the wrong information, or information that has been corrupted

Additional problems that you might encounter if your application uses distributed queues are also described.

- [“Messages do not arrive when expected on z/OS” on page 273](#)
- [“Problems with missing messages when using distributed queuing on z/OS” on page 275](#)
- [“Problems with getting messages when using message grouping on z/OS” on page 276](#)
- [“Finding messages sent to a cluster queue on z/OS” on page 276](#)
- [“Finding messages sent to the IBM MQ - IMS bridge” on page 277](#)
- [“Messages contain unexpected or corrupted information on z/OS” on page 277](#)

### **Related concepts**

[“Dealing with performance problems on z/OS” on page 266](#)

Use this topic to investigate IBM MQ for z/OS performance problems in more detail.

### **Related tasks**

[“进行初始检查” on page 6](#)

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。

## **Messages do not arrive when expected on z/OS**

Missing messages can have different causes. Use this topic to investigate the causes further.

If messages do not arrive on the queue when you are expecting them, check for the following:

### **Has the message been put onto the queue successfully?**

Did IBM MQ issue a return and reason code for the MQPUT, for example:

- Has the queue been defined correctly, for example is MAXMSGL large enough? (reason code 2030).
- Can applications put messages on to the queue (is the queue enabled for MQPUT calls)? (reason code 2051).
- Is the queue already full? This could mean that an application could not put the required message on to the queue (reason code 2053).

### **Is the queue a shared queue?**

- Have Coupling Facility structures been defined successfully in the CFRM policy data set? Messages held on shared queues are stored inside a Coupling Facility.
- Have you activated the CFRM policy?

### **Is the queue a cluster queue?**

If it is, there might be multiple instances of the queue on different queue managers. This means that the messages could be on a different queue manager.

- Did you want the message to go to a cluster queue?
- Is your application designed to work with cluster queues?
- Did the message get put to a different instance of the queue from that expected?

Check any cluster-workload exit programs to see that they are processing messages as intended.

## Do your gets fail?

- Does the application need to take a syncpoint?

If messages are being put or got within syncpoint, they are not available to other tasks until the unit of recovery has been committed.

- Is the time interval on the MQGET long enough?

If you are using distributed processing, you should allow for reasonable network delays, or problems at the remote end.

- Was the message you are expecting defined as persistent?

If not, and the queue manager has been restarted, the message will have been deleted. Shared queues are an exception because nonpersistent messages survive a queue manager restart.

- Are you waiting for a specific message that is identified by a message or correlation identifier (*MsgId* or *CorrelId*)?

Check that you are waiting for a message with the correct *MsgId* or *CorrelId*. A successful MQGET call sets both these values to that of the message got, so you might need to reset these values to get another message successfully.

Also check if you can get other messages from the queue.

- Can other applications get messages from the queue?

If so, has another application already retrieved the message?

If the queue is a shared queue, check that applications on other queue managers are not getting the messages.

If you cannot find anything wrong with the queue, and the queue manager itself is running, make the following checks on the process that you expected to put the message on to the queue:

- Did the application get started?

If it should have been triggered, check that the correct trigger options were specified.

- Is a trigger monitor running?
- Was the trigger process defined correctly (both to IBM MQ for z/OS and CICS or IMS)?
- Did it complete correctly?

Look for evidence of an abend, for example, in the CICS log.

- Did the application commit its changes, or were they backed out?

Look for messages in the CICS log indicating this.

If multiple transactions are serving the queue, they might occasionally conflict with one another. For example, one transaction might issue an MQGET call with a buffer length of zero to find out the length of the message, and then issue a specific MQGET call specifying the *MsgId* of that message. However, while this is happening, another transaction might have issued a successful MQGET call for that message, so the first application receives a completion code of MQRC\_NO\_MSG\_AVAILABLE. Applications that are expected to run in a multi-server environment must be designed to cope with this situation.

Have any of your systems suffered an outage? For example, if the message you were expecting should have been put on to the queue by a CICS application, and the CICS system went down, the message might be in doubt. This means that the queue manager does not know whether the message should be committed or backed out, and so has locked it until this is resolved when resynchronization takes place.

**Note:** The message is deleted after resynchronization if CICS decides to back it out.

Also consider that the message could have been received, but that your application failed to process it in some way. For example, did an error in the expected format of the message cause your program to reject it? If so, refer to [“Messages contain unexpected or corrupted information on z/OS” on page 277.](#)

Use this topic to understand possible causes of missing messages when using distributed queuing on IBM MQ for z/OS.

If your application uses distributed queuing, consider the following points:

**Has distributed queuing been correctly installed on both the sending and receiving systems?**

Ensure that the instructions about installing the distributed queue management facility in [Configuring z/OS](#) have been followed correctly.

**Are the links available between the two systems?**

Check that both systems are available, and connected to IBM MQ for z/OS. Check that the LU 6.2 or TCP/IP connection between the two systems is active or check the connection definitions on any other systems that you are communicating with.

See [Monitoring and performance](#) for more information about trace-route messaging in a network.

**Is the channel running?**

- Issue the following command for the transmission queue:

```
DISPLAY QUEUE (qname) IPPROCS
```

If the value for IPPROCS is 0, this means that the channel serving this transmission queue is not running.

- Issue the following command for the channel:

```
DISPLAY CHSTATUS (channel-name) STATUS MSGS
```

Use the output produced by this command to check that the channel is serving the correct transmission queue and that it is connected to the correct target machine and port. You can determine whether the channel is running from the STATUS field. You can also see if any messages have been sent on the channel by examining the MSGS field.

If the channel is in RETRYING state, this is probably caused by a problem at the other end. Check that the channel initiator and listener have been started, and that the channel has not been stopped. If somebody has stopped the channel, you need to start it manually.

**Is triggering set on in the sending system?**

Check that the channel initiator is running.

**Does the transmission queue have triggering set on?**

If a channel is stopped under specific circumstances, triggering can be set off for the transmission queue.

**Is the message you are waiting for a reply message from a remote system?**

Check the definitions of the remote system, as previously described, and check that triggering is activated in the remote system. Also check that the LU 6.2 connection between the two systems is not single session (if it is, you cannot receive reply messages).

Check that the queue on the remote queue manager exists, is not full, and accepts the message length. If any of these criteria are not fulfilled, the remote queue manager tries to put the message on the dead-letter queue. If the message length is longer than the maximum length that the channel permits, the sending queue manager tries to put the message on its dead-letter queue.

**Is the queue already full?**

This could mean that an application could not put the required message on to the queue. If this is so, check if the message has been put on to the dead-letter queue.

The dead-letter queue message header (dead-letter header structure) contains a reason or feedback code explaining why the message could not be put on to the target queue. See [MQDLH - Dead-letter header](#) for more information about the dead-letter header structure.

**Is there a mismatch between the sending and receiving queue managers?**

For example, the message length could be longer than the receiving queue manager can handle. Check the console log for error messages.

**Are the channel definitions of the sending and receiving channels compatible?**

For example, a mismatch in the wrap value of the sequence number stops the channel. See [Distributed queuing and clusters](#).

**Has data conversion been performed correctly?**

If a message has come from a different queue manager, are the CCSIDs and encoding the same, or does data conversion need to be performed.

**Has your channel been defined for fast delivery of nonpersistent messages?**

If your channel has been defined with the NPMSPEED attribute set to FAST (the default), and the channel has stopped for some reason and then been restarted, nonpersistent messages might have been lost. See [Nonpersistent message speed \(NPMSPEED\)](#) for more information about fast messages.

**Is a channel exit causing the messages to be processed in an unexpected way?**

For example, a security exit might prevent a channel from starting, or an *ExitResponse* of MQXCC\_CLOSE\_CHANNEL might terminate a channel.

**z/OS Problems with getting messages when using message grouping on z/OS**

Use this topic to understand some of the issues with getting messages when using message grouping on IBM MQ for z/OS.

**Is the application waiting for a complete group of messages?**

Ensure all the messages in the group are on the queue. If you are using distributed queuing, see “Problems with missing messages when using distributed queuing on z/OS” on page 275. Ensure the last message in the group has the appropriate MsgFlags set in the message descriptor to indicate that it is the last message. Ensure the message expiry of the messages in the group is set to a long enough interval that they do not expire before they are retrieved.

If messages from the group have already been retrieved, and the get request is not in logical order, turn off the option to wait for a complete group when retrieving the other group messages.

**If the application issues a get request in logical order for a complete group, and midway through retrieving the group it cannot find a message:**

Ensure that no other applications are running against the queue and getting messages. Ensure that the message expiry of the messages in the group is set to a long enough interval that they do not expire before they are retrieved. Ensure that no one has issued the CLEAR QUEUE command. You can retrieve incomplete groups from a queue by getting the messages by group ID, without specifying the logical order option.

**z/OS Finding messages sent to a cluster queue on z/OS**

Use this topic to understand some of the issues involved with finding messages sent to a cluster queue on IBM MQ for z/OS.

Before you can use the techniques described in these topics to find a message that did not arrive at a cluster queue, you need to determine the queue managers that host the queue to which the message was sent. You can determine this in the following ways:

- You can use the DISPLAY QUEUE command to request information about cluster queues.
- You can use the name of the queue and queue manager that is returned in the MQPMO structure.

If you specified the MQOO\_BIND\_ON\_OPEN option for the message, these fields give the destination of the message. If the message was not bound to a particular queue and queue manager, these fields give the name of the first queue and queue manager to which the message was sent. In this case, it might not be the ultimate destination of the message.

## **Finding messages sent to the IBM MQ - IMS bridge**

Use this topic to understand possible causes for missing messages sent to the IBM MQ - IMS bridge.

If you are using the IBM MQ - IMS bridge, and your message has not arrived as expected, consider the following:

### **Is the IBM MQ - IMS bridge running?**

Issue the following command for the bridge queue:

```
DISPLAY QSTATUS(qname) IPPROCS CURDEPTH
```

The value of IPPROCS should be 1; if it is 0, check the following:

- Is the queue a bridge queue?
- Is IMS running?
- Has OTMA been started?
- Is IBM MQ connected to OTMA?

**Note:** There are two IBM MQ messages that you can use to establish whether you have a connection to OTMA. If message CSQ2010I is present in the job log of the task, but message CSQ2011I is not present, IBM MQ is connected to OTMA. This message also tells you to which IBM MQ system OTMA is connected. For more information about the content of these messages, see [IBM MQ for z/OS messages, completion, and reason codes](#).

Within the queue manager there is a task processing each IMS bridge queue. This task gets from the queue, sends the request to IMS, and then does a commit. If persistent messages are used, then the commit requires disk I/O and so the process takes longer than for non-persistent messages. The time to process the get, send, and commit, limits the rate at which the task can process messages. If the task can keep up with the workload then the current depth is close to zero. If you find that the current depth is often greater than zero you might be able to increase throughput by using two queues instead of one.

Use the IMS command `/DIS OTMA` to check that OTMA is active.

### **If your messages are flowing to IMS, check the following:**

- Use the IMS command `/DIS TMEMBER client TPIPE ALL` to display information about IMS Tpipes. From this you can determine the number of messages enqueued on, and dequeued from, each Tpipe. (Commit mode 1 messages are not usually queued on a Tpipe.)
- Use the IMS command `/DIS A` to show whether there is a dependent region available for the IMS transaction to run in.
- Use the IMS command `/DIS TRAN trancode` to show the number of messages queued for a transaction.
- Use the IMS command `/DIS PROG progname` to show if a program has been stopped.

### **Was the reply message sent to the correct place?**

Issue the following command:

```
DISPLAY QSTATUS(*) CURDEPTH
```

Does the CURDEPTH indicate that there is a reply on a queue that you are not expecting?

## **Messages contain unexpected or corrupted information on z/OS**

Use this topic to understand some of the issues that can cause unexpected or corrupted output on z/OS.

If the information contained in the message is not what your application was expecting, or has been corrupted in some way, consider the following points:



### Has your application, or the application that put the message on to the queue changed?

Ensure that all changes are simultaneously reflected on all systems that need to be aware of the change.

For example, a copybook formatting the message might have been changed, in which case, both applications have to be recompiled to pick up the changes. If one application has not been recompiled, the data will appear corrupt to the other.

Check that no external source of data, such as a VSAM data set, has changed. This could also invalidate your data if any necessary recompilations have not been done. Also check that any CICS maps and TSO panels that you are using for input of message data have not changed.

### Is an application sending messages to the wrong queue?

Check that the messages your application is receiving are not intended for an application servicing a different queue. If necessary, change your security definitions to prevent unauthorized applications from putting messages on to the wrong queues.

If your application has used an alias queue, check that the alias points to the correct queue.

If you altered the queue to make it a cluster queue, it might now contain messages from different application sources.

### Has the trigger information been specified correctly for this queue?

Check that your application should have been started, or should a different application have been started?

### Has data conversion been performed correctly?

If a message has come from a different queue manager, are the CCSIDs and encoding the same, or does data conversion need to be performed.

Check that the *Format* field of the MQMD structure corresponds with the content of the message. If not, the data conversion process might not have been able to deal with the message correctly.

If these checks do not enable you to solve the problem, check your application logic, both for the program sending the message, and for the program receiving it.

## Dealing with issues when capturing SMF data for the channel initiator (CHINIT)

Channel accounting and CHINIT statistics SMF data might not be captured for various reasons.

For more information, see:

### Related concepts

[Layout of SMF records for the channel initiator](#)

## Troubleshooting channel accounting data

Checks to carry out if channel accounting SMF data is not being produced for channels.

### Procedure

1. Check that you have STATCHL set, either at the queue manager or the channel level.
  - A value of OFF at channel level means that data is not collected for this channel.
  - A value of OFF at queue manager level means data is not collected for channels with STATCHL(QMGR).
  - A value of NONE (only applicable at queue manager level) means data is not collected for all channels, regardless of their STATCHL setting.
2. For client channels, check that STATCHL is set at the queue manager level.

3. For automatically defined cluster sender channels, check that the STATACLS queue manager attribute is set.
4. Issue the **DISPLAY TRACE** command. You need TRACE(A) CLASS(4) enabled for channel accounting data to be collected.
5. If the trace is enabled, data is written to SMF when any of the following conditions occur:
  - From IBM MQ for z/OS 9.3.0, a timed interval, depending on the values of the STATIME and ACCTIME queue manager system parameters. Use the **DISPLAY SYSTEM** command to display the value of STATIME and ACCTIME.
  - From IBM MQ for z/OS 9.3.0, the **SET SYSTEM** command is issued to change the value of the STATIME or ACCTIME system parameters.
  - The channel initiator is shut down.
  - The **STOP TRACE(A) CLASS(4)** command is issued.
  - From IBM MQ 9.3.0, if you specify a value between 0 and 1440 for the ACCTIME parameter, that value is used instead of STATIME.
6. SMF might hold the data in memory before writing it out to the SMF data sets or the SMF structure. Issue the MVS™ command **D SMF,0** and note the MAXDORM value. The MAXDORM value is displayed in the format *mmss*, where *mm* is the time in minutes and *ss* is seconds. SMF can keep the data in memory for the MAXDORM period before writing it out.

### Related tasks

[Planning for channel initiator SMF data](#)

[Interpreting IBM MQ performance statistics](#)

### **Troubleshooting CHINIT statistics data**

Checks to carry out if CHINIT statistics SMF data is not being produced.

### Procedure

1. Issue the **DISPLAY TRACE** command. You need TRACE(S) CLASS(4) enabled to gather channel initiator statistics SMF data.
2. If the trace is enabled, data is written to SMF when any of the following conditions occur:
  - On a timed interval, depending on the value of the STATIME queue manager system parameter. A value of zero means that the SMF statistics broadcast is used. Use the **DISPLAY SYSTEM** command to display the value of STATIME.
  - The **SET SYSTEM** command is issued to change the value of the STATIME system parameter.
  - The channel initiator is shut down.
  - The **STOP TRACE(S) CLASS(4)** command is issued.
3. SMF can hold the data in memory before writing it out to the SMF data sets or the SMF structure. Issue the MVS command **D SMF,0** and note the MAXDORM value. The MAXDORM value is displayed in the format *mmss*, where *mm* is the time in minutes and *ss* is seconds. SMF can keep the data in memory for the MAXDORM period before writing it out.


## 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，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

本节说明如何收集 IBM MQ for [Multiplatforms](#) 或 IBM MQ for z/OS 可能迁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 支持人员可能会逐个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 281 页的『使用 runmqras 自动收集故障诊断信息』](#)
  - [第 284 页的『手动收集故障诊断信息』](#)
  - [第 367 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 319 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](#)。

- 

对于容器中的 IBM MQ, 请参阅 [收集使用 IBM MQ Operator 部署的队列管理器的故障诊断信息](#)。

- 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 325 页的『[Collecting troubleshooting information on z/OS](#)』

An overview of how to collect troubleshooting information for IBM MQ for z/OS.

## 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:

-  在 UNIX and Linux 上:

```
sh> PATH="$PATH":/path/to/java/bin (only if needed)
sh> . /opt/mqm/bin/setmqenv -n Installation1
```

-  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
```

-  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 284 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：

-   在 UNIX and Linux 上：

```
sh> runmqras -caseno TS123456789
```

-  在 Windows 上：

```
C:\> runmqras -caseno TS123456789
```

-  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。

-   在 UNIX and Linux 上：

```
sh> runmqras -zipfile TS123456789
```

-  在 Windows 上：

```
C:\> runmqras -zipfile TS123456789
```

-  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：

-   在 UNIX and Linux 上：

```
sh> runmqras -caseno TS123456789 -section defs,cluster,trace
```

-  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 367 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 284 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 367 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:

-   在 UNIX and Linux 上:

```
sh> . /path/to/mqm/bin/setmqenv -n InstallationX
```

-  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 **dspmqrer** 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** **UNIX** 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** **UNIX** 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 **amqsbcg** 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 289 页的『手动打包 UNIX 和 Linux 上的信息』](#)

- **Windows** 第 290 页的『手动打包 Windows 上的信息』
- **IBM i** 第 291 页的『手动打包 IBM i 上的信息』

## 相关任务

第 281 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 367 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 280 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 367 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.cti | 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 280 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 367 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 QPDSPLOG 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(QPDSPLOG) 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(QPJOBLOG) TOFILE(QGPL/JOBLOGS) JOB(135383/QMQM/RUNMQCHL) SPLNBR(1)
====> CPYSPLF FILE(QPJOBLOG) 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(JOBLG) 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 280 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 367 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 支持人员的帮助以解决多平台上的 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 388 页的『AIX and Linux 上的跟踪』
-  第 402 页的『Windows 上的跟踪』

- **IBM i** 第 393 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 `.mq` 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。

**Linux** **UNIX**

在 UNIX 和 Linux 上，使用以下命令：

```
ls -alR ~/.mq
```

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。(如果使用了 Java，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 CMS 或 PKCS #12 密钥库中证书的标签，请运行以下 `runmqakm` 命令：

```
runmqakm -cert -list -db keystorefilename -pw keystorepassword
```

- **V9.4.0** **V9.4.0** 要列出 JKS 密钥库中证书的标签，请运行以下 `runmqktool` 命令：

```
runmqktool -list -keystore keystorefilename
```

- 要显示 CMS 或 PKCS #12 密钥库中所有证书的详细信息，请对每个证书标签运行以下 `runmqakm` 命令：

```
runmqakm -cert -details -db keystorefilename -pw keystorepassword -label labelname
```

- **V9.4.0** **V9.4.0** 要显示 JKS 密钥库中所有证书的详细信息，请对每个证书标签运行以下 `runmqktool` 命令：

```
runmqktool -list -keystore keystorefilename -alias labelname -v
```

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 故障诊断信息。

您可以自动或手动执行此操作。

- 使用 `runmqras` 命令自动收集故障诊断信息，如第 281 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，手动收集故障诊断信息，如第 284 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：**请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 61 页的『[对 AMS 问题进行故障诊断](#)』

用于帮助您识别和解决与 Advanced Message Security (AMS) 相关的问题的故障诊断信息。

## 收集通道问题的信息

如果需要 IBM 支持人员提供帮助或在 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](#)。

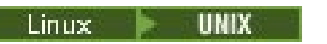
## 过程

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](#)
-  [IBM i](#)
- [Java 和 JMS 客户机](#)
-  [z/OS CHIN 跟踪](#)

3.  在 UNIX 和 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 281 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 284 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：**请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 69 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 388 页的『AIX and Linux 上的跟踪』](#)
-  [第 402 页的『Windows 上的跟踪』](#)
-  [第 393 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 388 页的『AIX and Linux 上的跟踪』](#)
-  [第 402 页的『Windows 上的跟踪』](#)
-  [第 393 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 296 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 281 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 284 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：**请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：

-   [第 388 页的『AIX and Linux 上的跟踪』](#)
-  [第 402 页的『Windows 上的跟踪』](#)
-  [第 393 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：

-   [第 388 页的『AIX and Linux 上的跟踪』](#)

- **Windows** 第 402 页的『Windows 上的跟踪』
- **IBM i** 第 393 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 297 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 281 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 284 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：**请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 175 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：

```
amqsbcg 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：

- Linux UNIX 第 388 页的『AIX and Linux 上的跟踪』
- Windows 第 402 页的『Windows 上的跟踪』
- IBM i 第 393 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：

- Linux UNIX 第 388 页的『AIX and Linux 上的跟踪』
- Windows 第 402 页的『Windows 上的跟踪』
- IBM i 第 393 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 `amqsbcg`) 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：

```
amqsbcg 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：

- Linux UNIX 第 388 页的『AIX and Linux 上的跟踪』
- Windows 第 402 页的『Windows 上的跟踪』
- IBM i 第 393 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 281 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 284 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：**请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 159 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 388 页的『AIX and Linux 上的跟踪』](#)
-  [第 402 页的『Windows 上的跟踪』](#)
-  [第 393 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 `amqsbcbg`) 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：

```
amqsbcbg 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 281 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 284 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：**请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 159 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 388 页的『AIX and Linux 上的跟踪』
-  第 402 页的『Windows 上的跟踪』
-  第 393 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 281 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 284 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:** 请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改？

## 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 队列管理器和应用程序运行 **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 amqzlaa0.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:

- 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。
- 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** [Linux 和 UNIX](#)
- ▶ **Windows** [Windows](#)
- ▶ **IBM i** [IBM i](#)

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** [AIX](#)
- ▶ **Linux** [Linux](#)


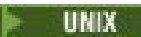
-  Windows
-  IBM i

## 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 304 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 281 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 284 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:** 请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 159 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?
- 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?

从 IBM MQ 9.3.0 开始，IBM MQ Explorer 功能部件不再包含在 Windows 或 Linux x86\_64 上的服务器产品中。它仍可作为这些平台的单独下载。有关更多信息，请参阅 [在 Linux 和 Windows 上作为独立应用程序安装和卸载 IBM MQ Explorer](#)。

### 关于此任务

IBM MQ Explorer 可通过 Fix Central 作为独立安装用于 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 388 页的『AIX and Linux 上的跟踪』
-  第 402 页的『Windows 上的跟踪』
-  第 393 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：



-   要在 UNIX and Linux 上显示网络连接：

```
sh> netstat -an
```

-  要在 Windows 上显示网络连接：

```
C:\> NETSTAT -AN
```

- g) 手动打包 IBM 的文件：

-  第 289 页的『手动打包 UNIX 和 Linux 上的信息』
-  第 290 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: 请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。

收集数据后, 可以将收集的信息发送到 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) 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" ...
```

### d) 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。

- 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。
- 记录 操作系统版本和维护级别。
- 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
```

- Linux UNIX 在 UNIX 和 Linux 系统上，包含 `/etc/opt/mqm/mqinst.ini` 文件 (如果存在)。

- 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)。

- 记录用于启动安装或卸载过程的精确命令。

- Linux 在 Linux 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)。

- 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。

- 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 `amqmsccw.txt` 和 `amqmjpse.txt` (如果存在)。

- 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)。

- 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
```

- 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：



-   [第 289 页的『手动打包 UNIX 和 Linux 上的信息』](#)
-  [第 290 页的『手动打包 Windows 上的信息』](#)
-  [第 291 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:** 请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 159 页的『对消息问题进行故障诊断』](#)

## 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 Message Service 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 388 页的『AIX and Linux 上的跟踪』](#)
-  [第 402 页的『Windows 上的跟踪』](#)
-  [第 393 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 309 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 281 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 284 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:** 请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 388 页的『AIX and Linux 上的跟踪』](#)
-  [第 402 页的『Windows 上的跟踪』](#)
-  [第 393 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：

-   在 UNIX and Linux 上：

```
sh> endmqm -i QMA
sh> dmpmqlog -b -m QMA -f /var/mqm/log/QMA > /tmp/QMA.dmpmqlog.txt
sh> stirmqm QMA
```

-  在 Windows 上：

```
C:\> endmqm -i QMA
C:\> dmpmqlog -b -m QMA -f "C:\ProgramData\IBM\MQ\log\QMA" > %TEMP%\QMA.dmpmqlog.txt
C:\> stirmqm QMA
```

3.  

在 Linux 以及 UNIX 系统上，保存 **mqconfig** 命令的输出。

4. 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 310 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 281 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 284 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：**请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

1. 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 317 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：**请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 110 页的『[对 Managed File Transfer 问题进行故障诊断](#)』

使用此信息可帮助您诊断 Managed File Transfer (MFT) 中的错误。

## **Multi** 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 116 页的『对代理程序状态问题进行故障诊断』](#)
- [第 136 页的『对 java.lang.OutOfMemoryError 问题进行故障诊断』](#)
- [第 144 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 443 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 317 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 281 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 129 页的『对报告未找到文件的协议网桥代理进行故障诊断』](#) 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 317 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 129 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 317 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 122 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 IBM MQ for z/OS 的版本。
  - 目标代理队列管理器的 IBM MQ 或 IBM MQ for z/OS 的版本。
  - 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。
  - 如果源和/或目标代理正在 IBM MQ 9.3 或更高版本上运行 Managed File Transfer 或 IBM MQ for z/OS，那么 VERBOSE 传输日志将涵盖发生问题的时间。有关如何创建传输日志的更多信息，请参阅 [fteSetAgentLogLevel](#) (开启或关闭对某些 MFT 代理操作的文件的日志记录)。
  - 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 443 页的『在多平台上跟踪 Managed File Transfer 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 317 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 281 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。

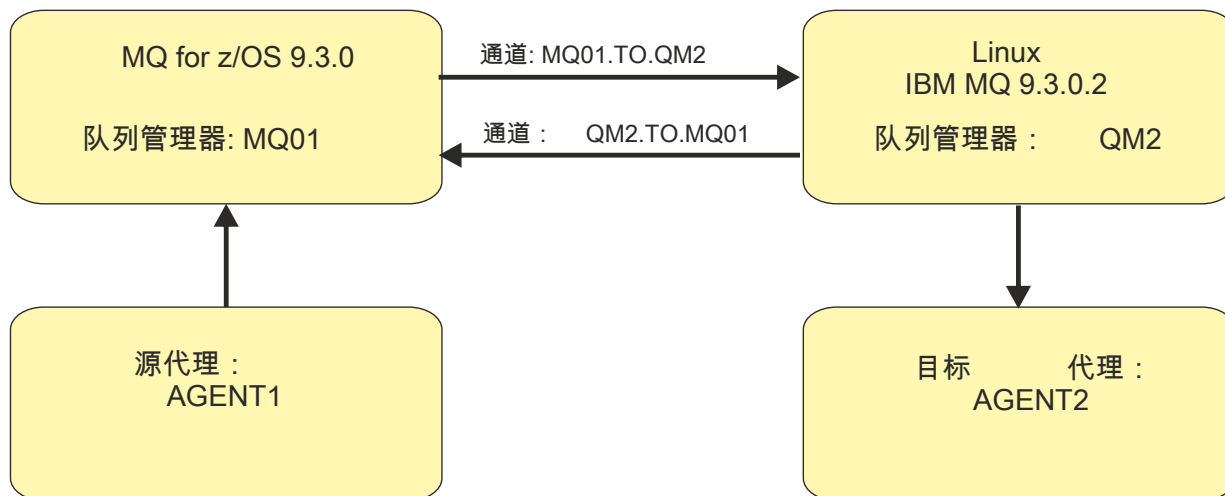


图 54: 显示如何连接源代理 AGENT1 和目标代理 AGENT2 及其代理队列管理器 MQ01 和 QM2 的简单图示例。

### 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 142 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 446 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 317 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 281 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 142 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 446 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遇到问题 (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 317 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 281 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 445 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 317 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。

**V9.4.0** 从 IBM MQ 9.3.4 开始，您可以收集特定 Managed File Transfer 代理程序 (而不是系统上所有代理程序) 的故障诊断信息。通过运行带有 **-agents** 参数的 **fteRAS** 命令来执行此操作。

## 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\COORDQMNAME\*
C:\Program Files\IBM\MQ\mqft\config\COORDQMNAME\*
```

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 上的程序和数据目录位置](#)。

- **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/COORDQMNAME/*
/QIBM/UserData/mqm/mqft/config/COORDQMNAME/*
```

## 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 311 页的『在 Multiplatforms 版上收集 Managed File Transfer 问题的信息』的步骤第 311 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 319 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 281 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 284 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:** 请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 427 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等。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：请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 279 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 81 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 版上报告问题的信息，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 Message Service 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 388 页的『AIX and Linux 上的跟踪』](#)
-  [第 402 页的『Windows 上的跟踪』](#)
-  [第 393 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 320 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 281 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 284 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：**请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. Linux 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 281 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。

### 相关任务

第 194 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。

- **Linux** **UNIX** [第 388 页的『AIX and Linux 上的跟踪』](#)
- **Windows** [第 402 页的『Windows 上的跟踪』](#)
- **IBM i** [第 393 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 281 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 284 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：**请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 208 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`：

-   [第 388 页的『AIX and Linux 上的跟踪』](#)
-  [第 402 页的『Windows 上的跟踪』](#)
-  [第 393 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：

-   [第 388 页的『AIX and Linux 上的跟踪』](#)
-  [第 402 页的『Windows 上的跟踪』](#)
-  [第 393 页的『IBM i 上的跟踪』](#)
- [Java 和 JMS 客户机应用程序](#)
-  [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 281 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 284 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：**请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 208 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。

## 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 388 页的『AIX and Linux 上的跟踪』](#)
-  [第 402 页的『Windows 上的跟踪』](#)
-  [第 393 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 281 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 284 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：**请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 159 页的『对消息问题进行故障诊断』](#)

## Collecting troubleshooting information on z/OS

An overview of how to collect troubleshooting information for IBM MQ for z/OS.

### About this task

**Note:** In addition to the information described in this section, IBM Support might request further information on a case by case basis.

### Procedure

- For information on how to collect troubleshooting and diagnostic information for a specific problem area for IBM MQ for z/OS, see the following topics:
  - [Abend problems](#)



- [MQ Adv. VUE](#) [MQ Adv. z/OS](#) [AMS problems](#)
- [C client problems](#)
- [Channel problems](#)
- [CICS adapter problems](#)
- [CICS bridge problems](#)
- [Cluster problems](#)
- [Data conversion problems](#)
- [Database problems](#)
- [Dead letter queue problems](#)
- [Error messages](#)
- [IBM MQ Explorer remote administration problems](#)
- [IBM WebSphere MQ File Transfer Edition \(FTE\): see Managed File Transfer for z/OS problems](#)
- [IMS OTMA bridge or IMS adapter problems](#)
- [Install problems](#)
- [Java and JMS application problems](#)
- [Managed File Transfer for z/OS problems](#)
- [Performance problems](#)
- [Publish/subscribe problems](#)
- [Security problems](#)
- [Shared channel problems](#)
- [Shared queue problems](#)
- [Shutdown problems](#)
- [Startup problems](#)
- [TLS problems](#)
- [Triggering channels problems](#)
- [Triggering programs problems](#)
- [Wait, hang, or loop problems](#)
- For all other problems, see [Collect troubleshooting data for a general, or unknown problem in WebSphere MQ for z/OS](#).

### Related tasks

“在多平台上收集故障诊断信息” on page 280

有关如何收集 IBM MQ on Multiplatforms 的故障诊断信息的概述。

### [z/OS](#) **Collecting information for abend problems on z/OS**

If you need assistance from IBM Support to resolve an abend problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### Before you begin

Search the [IBM Support site](#) for known problems.

You can search by using symptoms like the message number and error codes.

### Procedure

1. Collect the following troubleshooting (MustGather) information for this problem:
  - a) Collect the following required information:

## Job logs

You can find the IBM MQ for z/OS job logs in the Syslog, MSTR job log, and CHIN job log. The job logs are named *xxxxMSTR* and *xxxxCHIN*, where *xxxx* is the IBM MQ subsystem identifier (SSID). For more information, see [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

## Dumps generated at point of failure

IBM MQ dumps are located in a system dump data set (see Step “4” on page 327).

b) Optionally, also collect the z/OS LOGREC report.

See [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS” on page 265](#).

**Note:** You also need the version and release number of the product. See step “2” on page 327 for information on how you find this information.

2. Record the level of the product for IBM MQ for z/OS and the product at the other end of the channel.

See message `CSQY000I` in the MSTR job log for IBM MQ for z/OS, and for other platforms, see [Displaying the IBM MQ version](#)

3. Use the `AMATERSE` utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.

4. Review the dumps generated at point of failure.

IBM MQ dumps are located in a system dump data set and can be identified by their title. The title for a dump requested by IBM MQ starts with the four-character subsystem name of the queue manager. For example:

```
CSQ1, ABN=5C6-00E20016, U=SYSOPR, C=MQ900.910.DMC  
-CSQIALLC, M=CSQGFRCV, LOC=CSQSLD1 .CSQSVSTK+00000712
```

The dump title might provide sufficient information in the abend and reason codes to resolve the problem. For more information, see [“Analyzing the dump and interpreting dump titles on z/OS” on page 262](#).

For more information about the two system abend completion codes X'5C6' and X'6C6' that IBM MQ for z/OS uses, see [“IBM MQ for z/OS abends” on page 228](#). You can also search for known problems at the [IBM support site](#) by using abend codes, reason codes, and program names listed in the dump.

5. Check the system log (syslog).

Comm dumps might not contain the queue manager name, depending on the comment specified in the dump command. Check the syslog for an `IEA611I` or `IEA911E` message to determine the dump data set name and also to see whether the dump is complete or partial. For example:

```
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)
```

If insufficient disk space is the reason for the problem, there might not be sufficient information in the dump to diagnose the problem.

Dumps might be suppressed by Dump Analysis and Elimination (DAE). In this case, some symptoms might not appear in the system log (syslog) or joblog, but they appear in Logrec (see [“SYS1.LOGREC information on z/OS” on page 265](#)). For more information about management of DAE, see [Generating a suppressed dump](#).

6. Send the information that you have collected to 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。

**Note:** 请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 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 [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。

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？
- 正在使用哪些 IBM MQ 安全策略以及 AMS 正在使用哪些文件？

提供有关如何设置 AMS 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-针对涉及的每个应用程序用户，例如，放置和获取应用程序
- `<AMSUSER>\drq.ams.keyring` 的 RACF 列表
- RACDCERT 标识 (用户标识) `LISTRING(drq.ams.keyring)`
- RACDCERT 标识 (CSQ1AMSM) `LISTRING(drq.ams.keyring)` -将 CSQ1 替换为队列管理器的名称  
有关 RACDCERT 信息，请参阅 [用于外部安全性管理器的 TLS 数字证书命令](#)：
- 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

**注：**您还需要产品的版本和发行版号。请参阅步骤 [第 328 页的『4』](#)，以获取有关如何查找此信息的信息。

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 和放入/获取应用程序地址空间的转储。  
请参阅 [第 246 页的『IBM MQ for z/OS dumps』](#) 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 405 页的『Tracing on z/OS』](#) 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)。  
有关如何捕获 AMS 跟踪的信息，请参阅步骤 [第 329 页的『7』](#)。
- IBM Global Security Kit (GSKit) 跟踪。  
请参阅步骤 [第 329 页的『8』](#)，以获取有关如何捕获 GSKit 跟踪的信息。

4. 记录 IBM MQ for z/OS 的产品级别以及通道另一端的产品级别。

请参阅 IBM MQ for z/OS 的 MSTR 作业日志中的消息 [CSQY000I](#)，对于其他平台，请参阅 [显示 IBM MQ 版本](#)

5. 在上载到 ECUREP 之前，请使用 [AMATERSE](#) 实用程序，并确保指定与数据关联的案例编号。
6. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.

7. 捕获 AMS 跟踪。

有关更多信息，请参阅第 416 页的『[Enabling internal trace for the AMSM address space](#)』。

8. 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 417 页的『[Using GSKit trace for problems related to certificates and keys when using AMS on z/OS](#)』。

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:** 请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 61 页的『[对 AMS 问题进行故障诊断](#)』

用于帮助您识别和解决与 Advanced Message Security (AMS) 相关的问题的故障诊断信息。

## **Collecting information for C client problems on z/OS**

If you need assistance from IBM Support to resolve a client problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

## Before you begin

Before you start this task, check:

- The level of IBM MQ client code that is installed on the client machine.
- That the client connection channel on the client machine is properly defined.

## About this task

If you can reproduce the client problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后，可以将其发送到 IBM。

## Procedure

1. Generate the following traces while the problem is happening:

- a. [Generate a GTF trace](#).
- b. [Generate a MSTR internal trace](#).
- c. [Generate a CHIN trace](#).

2. Optionally, use the following information:

- [Generated dumps](#)
- “[Using the z/OS DUMP command](#)” on page 248
- A LOGREC report, see [A LOGREC report and “SYS1.LOGREC information on z/OS”](#) on page 265.

**Note:** You also need the version and release number of the product. See step “4” on page 330 for information on how you find this information.

3. Collect the IBM MQ data.
4. Record the level of the product for IBM MQ for z/OS and the product at the other end of the channel. See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS, and for other platforms, see [Displaying the IBM MQ version](#)
  - a. Record the [Operating system version and maintenance level](#) on both sides of the channel.
  - b. Provide the client connection and server connection channel definitions.
5. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for channel problems on z/OS**

If you need assistance from IBM Support to resolve a channel problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### **Before you begin**

Before you start this task, answer the following questions about the problem:

- What channel problem did you observe on the system?
- What time did the channel problem start and when did it stop?
- Which queue managers, channels, remote queues and transmission queues are involved?

### **About this task**

If you can reproduce the channel problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后, 可以将其发送到 IBM。

### **Procedure**




1. Generate the following traces while the problem is happening:
  - a. [Generate a GTF trace](#).
  - b. [Generate a MSTR internal trace](#).
  - c. [Generate a CHIN trace](#).

**Note:** You also need the version and release number of the product. See step “5” on page 331 for information on how you find this information.

2. Generate an IBM MQ trace simultaneously at the other end of the channel.

This applies, whether it is a remote queue manager, a native client application, or a JMS or Java client:

-  [Linux and UNIX](#)
-  [Windows](#)

-  IBM i
  - [Java and JMS client](#)
3.  
- On UNIX and Linux systems, save the output from the **mqconfig** command.
  4. Collect the IBM MQ data.
  5. Record the level of the product for IBM MQ for z/OS and the product at the other end of the channel. See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS, and for other platforms, see [Displaying the IBM MQ version](#)
    - a. Record the [Operating system version and maintenance level](#) on both sides of the channel.
    - b. Record the IP addresses and host names of the systems on both sides of the channel.
    - c. If your channel is having difficulty connecting, use your operating system tools to list network connections on both sides immediately before and after the connection attempt:
      - To display the network connections on z/OS, issue the following command, TSO NETSTAT CONN.
      - To display the device links on z/OS, issue the following command, TSO NETSTAT DEVLINK.
      - To display the local interfaces on z/OS, issue the following command, TSO NETSTAT HOME.
  6. Use the AMATERSE utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
  7. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for CICS adapter problems on z/OS**

If you need assistance from IBM Support to resolve a CICS adapter problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### **Before you begin**

The components to connect CICS and IBM MQ for z/OS are integrated with CICS Transaction Server. CICS-IBM MQ Adapter modules and error messages begin with DFHMQ.

Before you start this task, answer the following questions about the problem:

- What is the problem symptom?
- When did the problem first occur?
- Is the problem a one time failure or reoccurring?
- Was software or hardware maintenance applied?
- What are the names of the queue managers, queues, CICS regions and transactions involved in the problem?

### **About this task**

If you can reproduce the CICS-IBM MQ Adapter problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后, 可以将其发送到 IBM。



## Procedure

1. Collect the data requested in the MustGather topic from the CICS documentation.  
See [Collecting CICS troubleshooting data \(CICS MustGather\)](#) for IBM Support.
2. If additional IBM MQ is required, generate the following traces while the problem is happening:
  - a. [Generate a MSTR internal trace](#).
  - b. Optionally, [Generate a CHIN trace](#) if messages are wing put or got from a channel.
  - c. Optionally, [Generate a GTF trace](#) while the problem is happening.

If you need to include IBM MQ and CICS trace entries in GTF, you should update the JOBNAME and USR parameters referenced in those instructions, as follows:

```
JOBNAME=(ssidMSTR,cicsjob,appjob),USR=(5E9,5EA,5EE,F6C)
```

where:

### **ssid**

Is the subsystem id of the queue manager

### **cicsjob**

Is replaced with your CICS job name

### **appjob**

Is any other job batch, IMS, or ssidCHIN job involved in putting or getting the messages.

**Note:** You also need the version and release number of the product. See step “4” on page 332 for information on how you find this information.

3. Collect the IBM MQ data.
4. Record the level of the product for IBM MQ for z/OS and CICS.  
See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS, and message [DFHSI1500](#) in the CICS job log for the CICS version information.
  - a. Record the [Operating system version and maintenance level](#) of IBM MQ and CICS.
  - b. Record the version, release, and maintenance level of any other products involved with the problem.
  - c. Collect the IBM MQ and CICS [joblogs](#).
  - d. Collect the IBM MQ and IBM MQ [dump](#).
5. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for CICS bridge problems on z/OS**

If you need assistance from IBM Support to resolve a CICSbridge problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

## Before you begin

The components to connect CICS and IBM MQ for z/OS are integrated with CICS Transaction Server. CICS-IBM MQ Bridge modules, whose names begin with DFHMQBR, and error messages begin with DFHMQ07.

Before you start this task, answer the following questions about the problem:

- What is the problem symptom?
- When did the problem first occur?
- Is the problem a one time failure or reoccurring?
- Was software or hardware maintenance applied?
- Are you using the 3270 or DPL Bridge?
- Are the fields set correctly in the [MQMD](#) and [MQCIH](#) structures?
- What are the names of the queue managers, queues, CICS regions and transactions involved in the problem?

## About this task

If you can reproduce the CICS-IBM MQ Bridge problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后，可以将其发送到 IBM。

## Procedure

1. Collect the data requested in the MustGather topic from the CICS documentation.  
See [Collecting CICS troubleshooting data \(CICS MustGather\) for IBM Support](#).
2. If additional IBM MQ is required, generate the following traces while the problem is happening:
  - a. [Generate a MSTR internal trace](#).
  - b. Optionally, [Generate a CHIN trace](#) if messages are wing put or got from a channel.
  - c. Optionally, [Generate a GTF trace](#) while the problem is happening.

If you need to include IBM MQ and CICS trace entries in GTF, you should update the JOBNAME and USR parameters referenced in those instructions, as follows:

```
JOBNAME=(ssidMSTR,cicsjob,appjob),USR=(5E9,5EA,5EE,F6C)
```

where:

### **ssid**

Is the subsystem id of the queue manager

### **cicsjob**

Is replaced with your CICS job name

### **appjob**

Is any other job batch, IMS, or ssidCHIN job involved in putting or getting the messages.

**Note:** You also need the version and release number of the product. See step [“4” on page 333](#) for information on how you find this information.

3. Collect the IBM MQ data.
4. Record the level of the product for IBM MQ for z/OS and CICS.  
See message CSQY000I in the MSTR job log for IBM MQ for z/OS, and message DFHSI1500 in the CICS job log for the CICS version information.
  - a. Record the [Operating system version and maintenance level of IBM MQ and CICS](#).
  - b. Record the version, release, and maintenance level of any other products involved with the problem.

- c. Collect the IBM MQ and CICS [joblogs](#).
- d. Collect the IBM MQ and IBM MQ [dump](#).
5. Use the AMATERSE utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for cluster problems on z/OS**

If you need assistance from IBM Support to resolve cluster queues, topics, or channels problems on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### **Before you begin**

Before you start this task, answer the following questions about the problem:

- What cluster problem did you observe on the system?
- What time did the cluster problem start?
- Which queue managers, channels, remote queues, and transmission queues are involved?

### **About this task**

If you can reproduce the cluster problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后, 可以将其发送到 IBM。

### **Procedure**

1. Collect the following required information while the problem is happening:
  - a) Information on the default cluster objects  
See [Default cluster objects](#)
  - b) Dumps generated at point of failure.  
See [“IBM MQ for z/OS dumps” on page 246](#)
  - c) Dumps taken at the point of failure.  
See [“Using the z/OS DUMP command” on page 248](#)

**Note:** You also need the version and release number of the product. See step [“4” on page 334](#) for information on how you find this information.

2. Generate the following optional information while the problem is happening:
  - a. [Generate a GTF trace](#).
  - b. [Generate a MSTR internal trace](#).
  - c. [Generate a CHIN trace](#).
3. Collect the IBM MQ data.
4. Record the level of the product for IBM MQ for z/OS.

See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS, and for other platforms, see [Displaying the IBM MQ version](#)

- a. Record the [Operating system version and maintenance level](#) of all the queue managers involved.
- b. Provide feedback regarding cluster errors, if these occur with sample applications as well.
- c. Provide output from the following commands:

**DISPLAY CLUSQMGR(\*) ALL**

Produces information about queue manager interconnection in a cluster

**DISPLAY QUEUE(\*) CLUSINFO**

Produces information about cluster queues

**DISPLAY CHSTATUS(\*) ALL**

Produces information about channel status

5. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for data conversion problems on z/OS**

If you need assistance from IBM Support to resolve a data conversion problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### **Before you begin**

Before you start this task, answer the following questions about the problem:

- What data conversion problem did you observe on the system?
- What time did the data conversion problem start and when did it stop?
- Which queue managers, channels, remote queues and transmission queues are involved?

Investigate the following:

- An IBM MQ message consists of two parts, the control information in a message descriptor and the application data.

Application data is converted using one of the following methods:

- In an application program when the MQGMO\_CONVERT option is specified on an MQGET call.
- In the channel program by specifying CONVERT(YES) keyword on the sender or server channel.
- The **Format** field in the MQMD structure associated with the message must contain a valid format:
  - MQFMT\_NONE is the initial setting and data conversion does not occur with this setting.
  - The built in format (MQFMT\_STRING) should be used if the message is string data. IBM MQ data conversion programs convert the data.
  - If your message contains numeric data, then you need to have your own format. You also need to write your own exit program to do the data conversion.
  - The built in format (MQFMT\_CICS) can be used with CICS messages, however messages in that format can only be converted on IBM MQ on host systems. When sending messages to a different platform you should configure the sender channel process to do the data conversion. See [RC 2110 \(MQRC\\_FORMAT\\_ERROR\)](#) for more information.

- Conversion of EBCDIC newline characters

If you need to ensure that the data you send from an EBCDIC platform to an ASCII one is identical to the data you receive back again, you must control the conversion of EBCDIC newline characters. This can be done using a platform-dependent switch that forces IBM MQ to use the unmodified conversion tables but you must be aware of the inconsistent behavior that can result.

The problem arises because the EBCDIC newline character is not converted consistently across platforms or conversion tables. As a result, if the data is displayed on an ASCII platform, the formatting can be incorrect. This makes it difficult, for example, to administer an iSeries system remotely from an ASCII platform using RUNMQSC.

For further information about converting EBCDIC-format data to ASCII format, see [ConvEBCDICNewLine](#).

## About this task

If you can reproduce the data conversion problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后，可以将其发送到 IBM。

## Procedure

1. Collect the following required information:

- a) Job logs

You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

- b) A LOGREC report

See [A LOGREC report and “SYS1.LOGREC information on z/OS” on page 265](#).

- c) Gather the following information for the Sending and Receiving queue manager:

### **Sending queue manager**

```
Queue Manager CCSID:  
Putting application setting for MQMD CCSID:  
Putting application setting for MQMD Format:  
Use CSQ4BCG1 to capture the message on the transmission queue:  
What is the character and its Hex representation and offset within the message:
```

### **Receiving queue manager**

```
Queue Manager CCSID:  
Getting Application Setting for MQMD CCSID:  
Use CSQ4BCG1 to capture the message on the destination/local queue:  
What is the character and its Hex representation and offset within the message:
```

**Note:** You also need the version and release number of the product. See step [“4” on page 336](#) for information on how you find this information.

2. Optionally, generate the following traces while the problem is happening:

- a. [Generate a GTF trace](#).
- b. [Generate a MSTR internal trace](#).
- c. [Generate a CHIN trace](#).

3. Collect the IBM MQ data.

4. Record the level of the product for IBM MQ for z/OS and any other product.

See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS

Record the [Operating system version and maintenance level](#) of your system.

5. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for database problems on z/OS**

If you need assistance from IBM Support to resolve a database problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### **Before you begin**

Before you start this task, answer the following questions about the problem:

- What database problem did you observe on the system?
- What time did the database problem start and when did it stop?
- Which queue managers, channels, remote queues and transmission queues are involved?
- What message number and error codes did you receive?

Investigate the following:

- Search the [IBM MQ Support site](#) for known problems.
- Search for the most recent database related APARs.

### **About this task**

If you can reproduce the database problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后, 可以将其发送到 IBM。

### **Procedure**

1. Collect the following required information:

- a) Job logs

You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

- b) A LOGREC report

See [A LOGREC report and “SYS1.LOGREC information on z/OS” on page 265](#).

**Note:** You also need the version and release number of the product. See step [“4” on page 338](#) for information on how you find this information.

2. Optionally, generate the following traces while the problem is happening:

- a. [Generate a GTF trace](#).
- b. [Generate a MSTR internal trace](#).
- c. [Generate a CHIN trace](#).



3. Collect the IBM MQ data.
4. Record the level of the product for IBM MQ for z/OS and any other product.  
See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS  
Record the [Operating system version and maintenance level](#) of your system.
5. Use the `AMATERSE` utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for dead letter queue problems on z/OS**

If you need assistance from IBM Support to resolve a dead letter queue problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### Before you begin

Before you start this task, answer the following questions about the problem:

- What dead letter queue problem (DLQ) did you observe on the system?
- What time did the DLQ problem start and when did it stop?
- Which queue managers, channels, remote queues and transmission queues are involved?
- What message number and error codes did you receive?

Investigate the following:

- Search the [IBM MQ Support site](#) for known problems.
- Determine why a message is on the DLQ:
  1. Browse the message on the dead letter queue by using a program that shows headers, for example, `CSQ4BGCO` in `SCSQLOAD` with sample JCL in `SCSQPROC`.
  2. Locate the dead letter header, which starts with the characters "**DLH**":
 

```
444C 4820 0000 0001 0000 010A 4D59 2E51 'DLH .....MY.Q'
```
  3. Locate the dead letter Reason code, that is the reason the message arrived on the DLQ, in the third word in the dead letter header.  
In the preceding example the dead letter Reason is `x'0000 010A'`.
  4. Convert hexadecimal `x'0000 010A'` to decimal 266.
  5. Use the dead letter reason code to determine why the message was placed on the dead-letter queue. The dead letter reason code is a Reason Code (`MQRC_*`) or a Feedback Code (`MQFB_*`). For example:

```
MQFB_TM_ERROR 266 X'0000010A'
MQRC_Q_FULL 2053 X'00000805'
```

`MQRC_*` lists the reason codes, and `MQFB_*` lists the feedback values.

Additionally, use the Fields information in the `MQDLH` structure to determine why the message was routed to the dead letter queue (DLQ)

## About this task

If you can reproduce the dead letter queue problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后，可以将其发送到 IBM。

## Procedure

1. Collect the following required information:

a) Job logs

You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

b) A LOGREC report

See [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS”](#) on page 265.

**Note:** You also need the version and release number of the product. See step [“4”](#) on page 339 for information on how you find this information.

2. Optionally, generate the following traces while the problem is happening:

a. [Generate a GTF trace](#).

b. [Generate a MSTR internal trace](#).

c. [Generate a CHIN trace](#).

3. Collect the IBM MQ data.

This data includes the reason code or feedback code in the dead letter header.

4. Record the level of the product for IBM MQ for z/OS and any other product.

See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS

Record the [Operating system version and maintenance level](#) of your system.

5. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.

6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for error messages on z/OS**

If you need assistance from IBM Support to resolve an error message problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

## Before you begin

Before you start this task, answer the following questions about the problem:

- What error message did you observe on the system?
- What time did the error message occur?
- Which queue managers, channels, remote queues and transmission queues are involved?
- What message number and error codes did you receive?

Investigate the following:

- Search the [IBM MQ Support site](#) for known problems.
- Search [IBM MQ for z/OS messages, completion, and reason codes](#) for the z/OS information, and messages for any other platform your enterprise is using in the product documentation.

## About this task

If you can reproduce the error message or the message is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后，可以将其发送到 IBM。

## Procedure

1. Collect the following required information:

a) Job logs

You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

b) A LOGREC report

See [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS”](#) on page 265.

**Note:** You also need the version and release number of the product. See step [“4”](#) on page 340 for information on how you find this information.

2. Optionally, generate the following traces while the problem is happening:

a. [Generate a GTF trace](#).

b. [Generate a MSTR internal trace](#).

c. [Generate a CHIN trace](#).

3. Collect the IBM MQ data.

4. Record the level of the product for IBM MQ for z/OS and any other product.

See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS

Record the [Operating system version and maintenance level](#) of your system.

5. Use the AMATERSE utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.

6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for IMS OTMA bridge or IMS adapter problems on z/OS**

If you need assistance from IBM Support to resolve an IMS Open Transaction Manager Access (OTMA) bridge or IMS adapter problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

## Before you begin

Before you start this task, answer the following questions about the problem:

- What IMS problem did you observe on the system?
- What time did the problem start and when did it stop?
- Which queue managers, channels, remote queues and transmission queues are involved?
- What message number and error codes did you receive?

For both IMS OTMA bridge or IMS adapter problems, search the [IBM MQ Support site](#) for known problems.

For IMS OTMA bridge problems, check that:

- The queue is a bridge queue, that is, the queue has been defined with an XCF STGCLASS.
- IMS is running.
- OTMA has been started.
- IBM MQ is connected to OTMA.

and look at:

- The latest IBM MQ IMS technical support information - see [Products and services](#)
- The latest IBM MQ IMS maintenance information - see [IMS service considerations](#)
- [“Finding messages sent to the IBM MQ - IMS bridge” on page 277](#)
- [MP16: Capacity Planning and Tuning for IBM MQ for z/OS](#)

For the IMS Adapter, check that the IMS subsystem is started and look at

- [“IMS-related problems” on page 495.](#)
- [Controlling the IMS trigger monitor.](#)

For security considerations for using IBM MQ with IMS, look at:

- [Security considerations for connecting to IMS](#)
- [Application access control for the IMS bridge](#)
- [Security checking on IMS](#)
- [Security checking done by the IMS bridge](#)
- [Using RACF PassTickets in the IMS header](#)
- [Connection security profiles for IMS connections](#)

## About this task

If you can reproduce the IMS problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后，可以将其发送到 IBM。

## Procedure

1. Collect the following required information:

a) Job logs

You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs.](#)

b) A LOGREC report

See [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS” on page 265.](#)

**Note:** You also need the version and release number of the product. See step “4” on page 342 for information on how you find this information.

2. Optionally, generate the following traces while the problem is happening:
  - a. [Generate a GTF trace.](#)
  - b. [Generate a MSTR internal trace.](#)
  - c. [Generate a CHIN trace.](#)
3. Collect the IBM MQ data.
4. Record the level of the product for IBM MQ for z/OS and any other product.  
See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS  
Record the [Operating system version and maintenance level](#) of your system.
5. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for install problems on z/OS**

If you need assistance from IBM Support to resolve an install problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### **Before you begin**

Review the [IBM MQ for z/OS Migration Guide](#) and also [Migrating IBM MQ on z/OS](#).

If there is still a problem, answer the following questions:

- What install problem did you observe on the system, including any error messages?
- Is there is an error in the Causer section of one of the following SMP/E installation jobs?
  - SMP/E RECEIVE
  - SMP/E APPLY
  - SMP/E ACCEPT
- Did you use the GROUPEXTEND option as recommended in the Program Directory? 有关程序目录的下载链接, 请参阅 [IBM MQ for z/OS 程序目录 PDF 文件](#)。

### **Procedure**

1. Perform the product installation.  
See [Installing IBM MQ for z/OS](#) and the IBM MQ for z/OS Program Directory.
2. Collect the following troubleshooting (MustGather) information for this problem:
  - a) Collect the following required information:

#### **Job logs**

You can find the IBM MQ for z/OS job logs in the Syslog, MSTR job log, and CHIN job log. The job logs are named *xxxxMSTR* and *xxxxCHIN*, where *xxxx* is the IBM MQ subsystem identifier (SSID). For more information, see [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

b) Optionally, also collect the z/OS LOGREC report.

See [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS” on page 265](#)).

**Notes:** You also need the version, release, and modification number of the product:

- You are installing. See step [“4” on page 343](#) for information on how you find this information.
- Already on your system, if you are migrating from an earlier version.

3. Search the [IBM Support site](#) for known problems.

You can search by using symptoms like the message number and error codes.

4. Record the level of the product for IBM MQ for z/OS and any other products related to the problem.

See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS, and for other platforms, see [Displaying the IBM MQ version](#)

5. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.

6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for Java and JMS application problems on z/OS**

If you need assistance from IBM Support to resolve a JMS or Java problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### **Before you begin**

Before you start this task, answer the following questions about the problem:

- What JMS or Java problem did you observe on the system?
- What time did the problem start and when did it stop?
- Were any Java exceptions reported, and did they include a Java call stack?
- Which queue managers, queues and topics does the JMS or Java application use?

Search the [IBM MQ Support site](#) for known problems using, for example, any error messages or return codes.

The majority of IBM MQ for z/OS Java failures are caused by setup or configuration problems. You must configure the path, classpath, and environment variables correctly. See:

- [Setting environment variables for IBM MQ classes for JMS/Jakarta Messaging](#)
- [Environment variables relevant to IBM MQ classes for Java](#)
- [Configuring the Java Native Interface \(JNI\) libraries](#)
- [IBM MQ classes for Java libraries](#)
- [Using WebSphere Application Server with IBM MQ](#)

### **About this task**

If you can reproduce the problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后, 可以将其发送到 IBM。



## Procedure

1. Collect the following required information:

a) Job logs

You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

b) Dumps generated at point of failure.

See [“IBM MQ for z/OS dumps”](#) on page 246

c) Dumps taken at the point of failure.

See [“Using the z/OS DUMP command”](#) on page 248

d) A LOGREC report

See [A LOGREC report](#)

**Note:** You also need the version and release number of the product. See step [“4”](#) on page 344 for information on how you find this information.

2. Optionally, generate the following while the problem is happening:

a. [Generate a GTF trace](#)

b. [Generate a MSTR internal trace](#)

c. [Generate a CHIN trace](#)

d. [“跟踪 IBM MQ classes for JMS 应用程序”](#) on page 433

e. [“跟踪 IBM MQ classes for Java 应用程序”](#) on page 436

f. [Enabling JMS Trace for WebSphere Application Server](#)

3. Collect the IBM MQ data.

4. Record the level of the product for IBM MQ for z/OS and any other product.

See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS

Record the [Operating system version and maintenance level](#) of your system.

5. For standalone applications, record the version of the IBM MQ classes for Java or IBM MQ classes for JMS that are being used.

### Finding the build level for the classes on the current classpath

Issue the following command:

```
java com.ibm.mq.MQJavaLevel
```

or

```
java com.ibm.mq.jms.MQJMSLevel
```

### Finding the build level for a jar files in a specific directory:

Change to the directory containing the jars, and issue the following command:

```
java -cp com.ibm.mq.jar com.ibm.mq.MQJavaLevel
```

or

```
java -cp com.ibm.mqjms.jar:com.ibm.mq.jar com.ibm.mq.jms.MQJMSLevel
```

If you receive *java.lang.NoClassDefFoundError* for **MQJavaLevel** or **MQJMSLevel**, you need provide a path to the class. You can use one of the following methods:

a. Specify the CLASSPATH in the command:

```
java -cp path/com.ibm.mq.jar com.ibm.mq.MQJavaLevel
```

where path is the directory path which contains the jar file of interest.

For example:

```
java -cp ./com.ibm.mq.jar com.ibm.mq.MQJavaLevel
```

gives version information for the `com.ibm.mq.jar` file in the current directory, or

```
java -cp /usr/lpp/mqm/V9R3M0/java/lib/com.ibm.mq.jar com.ibm.mq.MQJavaLevel
```

gives version information for the file `/usr/lpp/mqm/V9R3M0/java/lib/com.ibm.mq.jar`.

**Note:** `java -cp /usr/lpp/mqm/V9R3M0/java/lib/com.ibm.mq.jar com.ibm.mq.MQJavaLevel` must be on one line.

- b. Set up the CLASSPATH and LIBPATH environment variables.

See [Setting environment variables for IBM MQ classes for JMS/Jakarta Messaging and Configuring the Java Native Interface \(JNI\) libraries](#) to match the settings used by the application or WebSphere Application Server.

The "**env**" command in OMVS will show you what your current settings are.

If you need to set the variables for your OMVS session, the following export commands are an example of what to issue from the OMVS command line or add to your `.profile`:

```
export LIBPATH=/usr/lpp/mqm/V9R3M0/java/lib:$LIBPATH
export CLASSPATH=/usr/lpp/mqm/V9R3M0/java/lib/com.ibm.mq.jar:$CLASSPATH
export CLASSPATH=/usr/lpp/mqm/V9R3M0/java/lib/com.ibm.mqjms.jar:$CLASSPATH
```

**Notes:**

- `/usr/lpp` is an install directory that might be different for your environment
- These variables are the minimum needed to display the levels
- The run-time CLASSPATH contains other libraries.

6. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
7. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for Managed File Transfer for z/OS problems**

If you need assistance from IBM Support to resolve a problem when a Managed File Transfer (MFT) for z/OS agent, logger or command is reporting a problem or failing to work properly you first need to collect troubleshooting information to send to IBM Support to help find a solution. The information that is needed depends on the problem that you are seeing.

### Procedure

1. Collect the information that is needed for the type of problem that you are seeing:
  - [Managed File Transfer for z/OS agent problems](#)
  - [Managed File Transfer for z/OS protocol bridge agent problems](#)
  - [Managed File Transfer for z/OS resource monitor problems](#)
  - [Managed File Transfer for z/OS managed transfer problems](#)

- [Managed File Transfer for z/OS database logger problems](#)
  - [Managed File Transfer for z/OS command problems](#)
2. After you have collected the Managed File Transfer data that is needed to investigate the problem, create an archive containing all of the relevant files.  
For more information, see [“Creating an archive of troubleshooting information for MFT for z/OS” on page 351](#).
  3. Send the information that you have collected to 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。

**Note:** 请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。

### Related tasks

[“对 Managed File Transfer 问题进行故障诊断” on page 110](#)  
使用此信息可帮助您诊断 Managed File Transfer (MFT) 中的错误。

### *Collecting information for MFT for z/OS agent problems*

The troubleshooting information that you need to collect and send to IBM if you need assistance from IBM Support with a Managed File Transfer (MFT) agent problem on z/OS.

### About this task

Managed File Transfer agent problems include:

- The agent failing to connect to, or being disconnected from, its agent queue manager.
- The agent hanging.
- The agent stopping unexpectedly.
- The agent going into recovery.
- The **`fteListAgents`** or **`fteShowAgentDetails`** commands, or the IBM MQ Explorer Managed File Transfer plug-in, showing incorrect, or out of date, status information for the agent.
- The agent failing to report any status information.

### Procedure

1. Initially, review the following topics to see whether they help you to resolve the problem:
  - [“对代理程序状态问题进行故障诊断” on page 116](#)
  - [“对 java.lang.OutOfMemoryError 问题进行故障诊断” on page 136](#)
2. If you still require assistance, collect the following information and send it to IBM Support:
  - The name of the agent.
  - The name of the agent queue manager.
  - The version of Managed File Transfer for z/OS that the agent is using.
  - The version of IBM MQ for z/OS for the agent queue manager.
  - Details of how the agent is started (for example, is it running as a started task?).
  - Any error messages that are seen in the agent's event log (`output0.log`) when the issue occurs.
  - An agent trace covering the time of the issue. For more information about how to collect the trace, see [“Tracing Managed File Transfer for z/OS agents” on page 449](#).

- If the agent is experiencing the problem when communicating with its agent queue manager (for example, the agent's event log contains error messages that include an IBM MQ reason code such as 2009 – MQRC\_CONNECTION\_BROKEN), collect the trace using the trace specification =all.
- For all other issues, collect the trace using the trace specification com.ibm.wmqfte=all.
- Three Javacores taken 30 seconds apart, if the agent is hanging. In order to do this, run the **fteSetAgentTraceLevel** command with the -jtc option set, as shown in the following example:

```
fteSetAgentTraceLevel -jtc <agent_name>
```

If the command does not cause the agent to generate a Javacore, then you should send a SIGQUIT signal to the agent process.

- An archive containing the agent's log files, configuration files, trace files and Javacores, if applicable. For more information about how to create the archive, see [“Creating an archive of troubleshooting information for MFT for z/OS” on page 351](#).
- The job logs for the coordination queue manager and agent queue manager.

### *Collecting information for MFT for z/OS protocol bridge agent problems*

The troubleshooting information that you need to collect and send to IBM if you need assistance from IBM Support with a Managed File Transfer (MFT) protocol bridge agent problem on z/OS.

## About this task

Managed File Transfer protocol bridge agent problems include:

- The agent failing to connect to, or being disconnected from, a remote file server.
- Managed transfers to or from a remote file server failing.

## Procedure

1. Initially, review the information in [“对报告未找到文件的协议网桥代理进行故障诊断” on page 129](#) to see whether that helps you to resolve the problem.
2. If you still require assistance, collect the following information and send it to IBM Support:
  - The name of the protocol bridge agent.
  - The name of the protocol bridge agent queue manager.
  - The version of Managed File Transfer for z/OS that the protocol bridge agent is using.
  - The version of IBM MQ for z/OS for the protocol bridge agent queue manager.
  - The hostname of the remote file server system.
  - Product and version information for the remote file server.
  - The protocol that the agent is using to communicate with the remote file server (that is, FTP, FTPS or SFTP).
  - The entry for the remote file server in the protocol bridge agent configuration file (ProtocolBridgeProperties.xml).
  - Any error messages that are seen in the agent's event log (output0.log) when the issue occurs.
  - A protocol bridge agent log file, where the log level for the protocol being used is set to on. For more information about how to set the log level, see [fteSetAgentLogLevel \(Turn on or turn off logging to file of certain MFT agent operations\)](#).
  - An archive that contains the protocol bridge agent's log files and configuration files. For more information about how to create the archive, see [“Creating an archive of troubleshooting information for MFT for z/OS” on page 351](#).

The troubleshooting information that you need to collect and send to IBM if you need assistance from IBM Support with a Managed File Transfer (MFT) resource monitor problem on z/OS.

## About this task

Managed File Transfer resource monitor problems include:

- A resource monitor stops polling.
- A resource monitor is polling, and not triggering on any items (either files or messages).
- A resource monitor is not submitting managed transfer requests to the agent.
- A resource monitor stops unexpectedly.

## Procedure

1. Initially, review the information in [“对资源监视器问题进行故障诊断” on page 129](#) to see whether that helps you to resolve the problem.
2. If you still require assistance, collect the following information and send it to IBM Support:
  - The name of the agent.
  - The name of the agent queue manager.
  - The version of Managed File Transfer for z/OS that the agent is using.
  - The version of IBM MQ for z/OS for the agent queue manager.
  - The name of the resource monitor.
  - The name of the resource (either a queue or directory) that the monitor is polling.
  - The monitor's trigger condition.
  - The monitor's task XML.
  - Details of any items that the monitor is not triggering on.
  - A resource monitor log file (for example, `resmonevent0.log`), where the log level for the resource monitor is set to VERBOSE. For more information about how to create the log file, see [Logging MFT resource monitors](#).

If the monitor is polling, and has not got stuck, then the log file should include entries for at least three polls.

  - An archive containing the agent's configuration files, and the log files for the agent and resource monitor. For more information about how to create the archive, see [“Creating an archive of troubleshooting information for MFT for z/OS” on page 351](#).

The troubleshooting information that you need to collect and send to IBM if you need assistance from IBM Support with a Managed File Transfer (MFT) managed transfer problem on z/OS.

## About this task

Problems related to Managed File Transfer managed transfers include:

- A managed transfer failing unexpectedly.
- A managed transfer going into recovery and not completing.
- A managed transfer getting stuck.

## Procedure

1. Initially, review the information in [“对受管传输问题进行故障诊断” on page 122](#) to see whether that helps you to resolve the problem.

2. If you still require assistance, collect the following information and send it to IBM Support:

- The name of the source agent for the managed transfer.
- The name of the source agent queue manager.
- The version of Managed File Transfer or Managed File Transfer for z/OS that the source agent is using.
- The version of IBM MQ or IBM MQ for z/OS for the source agent queue manager.
- The name of the destination agent for the managed transfer.
- The name of the destination agent queue manager.
- The version of Managed File Transfer or Managed File Transfer for z/OS that the destination agent is using.
- The version of IBM MQ or IBM MQ for z/OS for the destination agent queue manager.
- If the source and destination agent queue managers are different, details of how the queue managers are connected together (that is, through sender/receiver channels or an IBM MQ cluster).
- The transfer identifier for the managed transfer.
- Details of how the managed transfer request was created (that is, was it generated by a resource monitor, the **fteCreateTransfer** command, the IBM MQ Explorer Managed File Transfer plug-in, or something else?).
- Details of any error messages that are related to the managed transfer in either the source agent or destination agent's event log (output0.log).
- If the source and/or destination agents, are running Managed File Transfer or Managed File Transfer for z/OS at IBM MQ 9.3 or later, a VERBOSE transfer log that covers the time when the issue occurred. For more information about how create the transfer log, see [fteSetAgentLogLevel \(Turn on or turn off logging to file of certain MFT agent operations\)](#).
- A trace from both the source and destination agents that covers the time when the issue occurred. For more information about how to collect the trace, see [“在多平台上跟踪 Managed File Transfer 代理”](#) on page 443 or [“Tracing Managed File Transfer for z/OS agents”](#) on page 449. The trace should be collected using the trace specification `com.ibm.wmqfte=all`.
- An archive from the source agent containing the agent's log files and configuration files and an archive from the destination agent, containing the agent's log files and configuration files. For more information about how to collect the archives for the source and destination agents, see [“创建 MFT 故障诊断信息的归档”](#) on page 317 or [“Creating an archive of troubleshooting information for MFT for z/OS”](#) on page 351.
- **runmqras** output for the source agent queue manager and the destination agent queue manager, if they are running on a platform other than z/OS. For more information about how to collect the **runmqras** output for the source agent queue manager and the destination agent queue manager, see [“使用 runmqras 自动收集故障诊断信息”](#) on page 281.

When you are investigating issues related to managed transfers, it is often useful to draw a simple diagram, as shown in the following example, that shows the agents and the agent queue managers. This diagram allows you and IBM Support to see how the agents and the agent queue managers are connected, which can help to identify possible issues within the IBM MQ network that might cause managed transfers to enter recovery or get stuck.



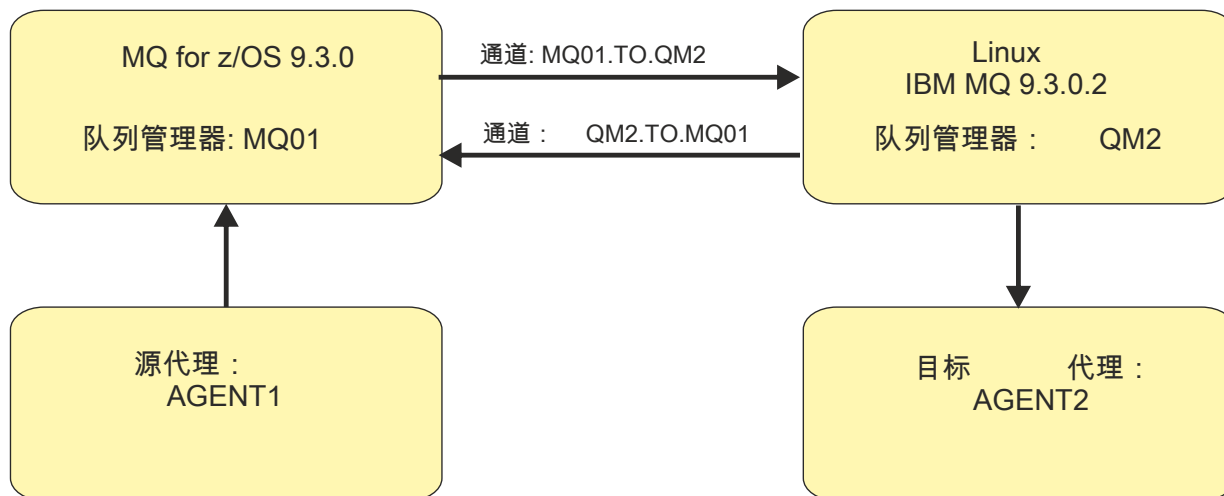


Figure 55. Example of a simple diagram showing how a source agent, AGENT1, and a destination agent, AGENT2, and their agent queue managers, MQ01 and QM2 are connected.

### z/OS Collecting information for MFT for z/OS database logger problems

The troubleshooting information that you need to collect and send to IBM if you need assistance from IBM Support with a Managed File Transfer (MFT) database logger problem on z/OS.

## About this task

Managed File Transfer database logger problems include:

- The database logger fails to connect to the coordination queue manager.
- The database logger fails to connect to the database.
- The database logger doesn't update the database.

## Procedure

1. Initially, review the information in [“对记录器 \(\[记录器\]\) 问题进行故障诊断”](#) on page 142 to see whether that helps you to resolve the problem.
2. If you still require assistance, collect the following information and send it to IBM Support:
  - The name of the database logger.
  - The name of the coordination queue manager that the database logger is connecting to.
  - The version of Managed File Transfer for z/OS that the database logger is using.
  - The version of IBM MQ for z/OS for the coordination queue manager.
  - The type of database that the database logger is using.
  - Details of any error messages that appear in the database logger's event log when the issue occurs.
  - A database logger trace that covers the time of the issue. For more information about how to collect this trace, see [“Tracing Managed File Transfer for z/OS standalone database loggers”](#) on page 457.
    - If the database logger is experiencing the problem when communicating with the coordination queue manager (for example, the database logger's event log contains error messages that include an IBM MQ reason code such as 2009 – MQRC\_CONNECTION\_BROKEN), collect the trace using the trace specification =a11.
    - For all other issues, collect the trace using the trace specification com.ibm.wmqfte=a11.
  - An archive that contains the database logger log files and configuration files. For more information about how to create the archive, see [“Creating an archive of troubleshooting information for MFT for z/OS”](#) on page 351.
  - The job logs for the coordination queue manager.

## Collecting information for MFT for z/OS command problems

The troubleshooting information that you need to collect and send to IBM if you need assistance from IBM Support with a Managed File Transfer (MFT) command problem on z/OS.

### About this task

Problems with Managed File Transfer commands include:

- A command failing to connect to the queue manager.
- A command timing out.
- A command reporting an error.

### Procedure

To investigate these, provide the following information:

- The command that is being run.
- Whether the command is being run from z/OS UNIX System Services (USS) or via JCL.
- The username for the user that is logged in when the command is run.
- The output from the command.
- The version of Managed File Transfer for z/OS that the command is using.
- A trace of the command, covering the time when the issue occurred. For information on how to collect this trace, see [“Tracing Managed File Transfer for z/OS commands”](#) on page 455.
  - If the command is experiencing the problem when communicating with a queue manager (for example, the command reports an error containing an IBM MQ reason code), collect the trace using the trace specification `=all`.
  - For all other issues, collect the trace using the trace specification `com.ibm.wmqfte=all`.
- An archive containing the configuration files on the system where the command is being run. For more information about how to create the archive, see [“Creating an archive of troubleshooting information for MFT for z/OS”](#) on page 351.

## Creating an archive of troubleshooting information for MFT for z/OS

After you have collected the data needed to investigate the Managed File Transfer (MFT) problem that you are seeing on z/OS, you need to create an archive that contains all of the relevant files and send it in to IBM Support. You can either create the archive manually, or by using the **fteRAS** utility.

### About this task



**Warning:** If there are a large number of Managed File Transfer for z/OS agents configured on a system, the **fteRAS** command can take a long time to complete. If that happens, you should create the archive manually by compressing the contents of the Managed File Transfer agent's logs and configuration directories.

#### V9.4.0

From IBM MQ 9.3.4, you can collect troubleshooting information for a specific Managed File Transfer agent or agents instead of for all agents on the system. You do this by running the **fteRAS** command with the **-agents** parameter.

### Procedure

- To archive the Managed File Transfer files automatically using the **fteRAS** command:
  - Copy any trace files that were generated when running a command into the BFG\_DATA directory for your Managed File Transfer for z/OS installation.
  - If you are using z/OS UNIX System Services (USS), run the **fteRAS** command.

- If you are using JCL:
  - Locate the data set containing the JCL for the installation.
  - Submit the BFGRAS member within the data set.
- To archive the Managed File Transfer files manually:
  - Copy the agent and logger javacores, traces, logs, properties and FFST files into a temporary directory. Include any interactive command traces and javacores written to the current directory or to other directories as well as the contents of the following directories:
    - `BFG_DATA/mqft/config/coordination_qmgr_name`
    - `BFG_DATA/mqft/logs/coordination_qmgr_name`
  - Create an archive containing the contents of the temporary directory.

## What to do next

Send the information that you have collected to IBM. For more information, see Step “3” on page 311 of [“Collecting information for Managed File Transfer for z/OS problems” on page 345.](#)

### 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。

请参阅 LOGREC 报告和 [第 265 页的『SYS1.LOGREC information on z/OS』](#)。

3. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
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:** 请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 69 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。

## **Collecting information for publish/subscribe problems on z/OS**

If you need assistance from IBM Support to resolve a publish/ subscribe problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### Before you begin

Before you start this task, answer the following questions about the problem:

- What publish/subscribe problem did you observe on the system?
- What time did the publish/subscribe problem start and has it stopped?
- Which queue managers are involved?
- What message number and error codes did you receive?

Investigate the following:

- Search the [IBM MQ Support site](#) for known problems.
- See [Troubleshooting MQ channels](#) for help in correcting problems with any channels..

### About this task

If you can reproduce the publish/subscribe problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后, 可以将其发送到 IBM。

### Procedure

1. Generate the following traces and job logs while the problem is happening:
  - a. [Generate a GTF trace](#).
  - b. [Generate a CHIN trace](#).
  - c. You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

2. You might require the following information:

a) IBM MQ system dumps

See [“IBM MQ for z/OS dumps” on page 246](#) for more information.

b) A MSTR internal trace

See [Generate a MSTR internal trace](#)

c) A LOGREC report

See [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS” on page 265](#).

**Note:** You also need the version and release number of the product. See step [“4” on page 354](#) for information on how you find this information.

3. Collect the IBM MQ data.

Include the following information:

- Publish/Subscribe status information, by issuing the command **DISPLAY PUBSUB TYPE(ALL) ALL**
- Topic attributes, by issuing the command **DISPLAY TOPIC(\*) TYPE(ALL) ALL**
- Subscription attributes, by issuing the command **DISPLAY SUB(\*) SUBTYPE(ALL) ALL**

4. Record the level of the product for IBM MQ for z/OS and any other product.

See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS

Record the [Operating system version and maintenance level](#) of your system.

5. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.

6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for security problems on z/OS**

If you need assistance from IBM Support to resolve a security problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### **Before you begin**

Before you start this task, answer the following questions about the problem:

- What security problem did you observe on the system?
- What time and date did the security problem start and when did it stop?
- Which queue managers are involved and, if relevant, channels, remote queues and transmission queues?
- What message number and error codes did you receive?

Investigate the following:

- Search the [IBM MQ Support site](#) for known problems.
- See [Troubleshooting MQ channels](#) for help in correcting problems with any channels..

## About this task

If you can reproduce the security problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后，可以将其发送到 IBM。

## Procedure

1. Generate the following traces while the problem is happening:
  - a. [Generate a GTF trace.](#)
  - b. [Generate a MSTR internal trace.](#)
  - c. [Generate a CHIN trace.](#)
2. You might need to collect the following information:
  - a) Job logs  
You require the Syslog, MSTR job log, and CHIN job log.  
The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs.](#)
  - b) A RACF trace  
See the [z/OS Security Server RACF Diagnosis Guide](#) for further information.
  - c) IBM MQ system dumps  
See [“IBM MQ for z/OS dumps” on page 246](#) for more information.
  - d) The output from the command DISPLAY SECURITY ALL  
See [Displaying security status](#) for more information.
  - e) Results of SETROPTS LIST and RACLIST commands  
See [Using SETROPTS RACLIST and SETROPTS GENLIST](#) for more information.
  - f) A LOGREC report  
See [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS” on page 265.](#)

**Note:** You also need the version and release number of the product. See step [“4” on page 355](#) for information on how you find this information.
3. Collect the IBM MQ data.
4. Record the level of the product for IBM MQ for z/OS and any other product.  
See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS  
Record the [Operating system version and maintenance level](#) of your system.
5. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
6. Send the information that you have collected to 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。

**Note:** 请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 上收集共享通道问题的信息

如果需要 IBM 支持人员的帮助以解决 IBM MQ for z/OS 上的共享通道问题，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共享通道问题？
- 共享通道问题何时启动并已停止？
- 涉及哪些队列管理器和通道启动程序？
- 您收到了哪些消息号和错误代码？

请调查以下内容：

- 在 [IBM MQ 支持站点](#) 中搜索已知问题。
- 请参阅 [共享通道](#)，以获取有关更正任何共享通道问题的帮助。
- 请参阅 [DVIPA](#) (动态虚拟 IP 地址) 以获取有关共享入站通道的信息。
- 性能问题：
  - 在通道两端执行 [DISPLAY CHSTATUS](#)，并检查 STATUS，SUBSTATE 和 INDOUBT。
  - 使用监视器或资源管理设施 (RMF) 数据来确认通道启动程序或 TCP/IP 任务未因 WLM 设置或 CPU 约束而遭受分派延迟。
  - 检查发送和接收缓冲区大小，因为 IBM MQ 通道启动程序创建的 TCP/IP 套接字采用 TCP/IP 概要文件中的缺省大小。
- 有关更多信息，请参阅 [分布式排队和队列共享组](#) 和 [使用队列共享组为 IBM MQ for z/OS 设置通信](#)。

## 关于此任务

如果可以重现共享通道问题或者问题正在发生，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作业日志和转储：

- a. 您需要 Syslog，MSTR 作业日志和 CHIN 作业日志。

作业日志名为 xxxxmstr 和 xxxxchln，其中 xxxx 是 IBM MQ 子系统标识 (SSID)。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](#)。

- b. IBM MQ 系统转储，请参阅 [第 246 页的『IBM MQ for z/OS dumps』](#) 以获取更多信息。

2. 您可能需要以下信息：

- a) MSTR 和 CHIN 内部跟踪

请参阅 [生成 MSTR 内部跟踪](#) 和 [生成 CHIN 跟踪](#)

- b) LOGREC 报告

请参阅 [LOGREC 报告](#)

**注：**您还需要产品的版本和发行版号。请参阅步骤 [第 357 页的『4』](#)，以获取有关如何查找此信息的信息。

- c) IBM MQ 命令的输出，其中 cpf 是队列管理器的命令前缀：

**/cpf DISPLAY CHINIT**

这将显示：

- 启动哪些侦听器以及有关这些侦听器的信息。
- 启动了多少分派器，适配器子任务和 SSL 子任务以及请求了多少分派器，适配器子任务和 SSL 子任务。
- TCP 系统名称。
- 当前的通道连接数以及这些连接是处于活动状态，已停止还是正在重试。
- 当前连接的最大数目。

### **/cpf DISPLAY CHSTATUS (xx) CURRENT**

获取当前状态。

### **/cpf 显示 CHSTATUS (xx) SAVED**

获取保存的状态。

### **/cpf 显示通道 (xx)**

获取通道定义。

#### d) 复制 SYSTEM.QSG.CHANNEL.SYNCQ

这是共享通道写入其同步队列信息的位置。有关详细信息和输出示例，请参阅 [在队列管理器运行时将队列复制到数据集](#)。

#### e) 作为共享通道的 ADMIN 结构的转储是序列化应用程序。

序列化信息存储在 QSG 管理结构中。例如，发出以下命令：

```
/dump comm=(title)
/i xx, strlist=(strname=QSGnameStructurename, (listnum=all,
adjunct=capture, entrydata=unser), eventqs, (emcontrols=all))
```

QSGnameStructurename 的示例为 CSQ1CSQ\_ADMIN，其中 CSQ1 是队列管理器子系统名称。

#### f) 对于出站通道，这是在其中定义共享传输队列的结构的转储。

#### g) 共享通道状态 Db2 表的 SPUFI 输出。发出下列命令：

```
SELECT * FROM CSQ.ADMIN_B_SCST ;
SELECT * FROM CSQ.ADMIN_B_SSKT ;
```

看使用 SPUFI 执行 SQL 了解更多信息。

此外，CSQ45STB 在 SCSQPROC 中有示例 JCL，您可以自定义它来执行 Db2SELECT 命令。

#### h) z/OS TCP/IP PKTTRACE。

请参阅 [How to collect Packet Trace and other TCP/IP related trace on z/OS](#) 以获取更多信息。

### 3. 收集 IBM MQ 数据。

包含以下信息：

- 通过发出命令 **DISPLAY PUBSUB TYPE(ALL)** 发布/预订状态信息
- 主题属性，通过发出命令 **DISPLAY TOPIC(\*) TYPE(ALL)**
- 预订属性，通过发出命令 **DISPLAY SUB(\*) SUBTYPE(ALL)**

### 4. 记录 IBM MQ for z/OS 和任何其他产品的产品级别。

请参阅 IBM MQ for z/OS 的 MSTR 作业日志中的消息 [CSQY000I](#)

记录系统的 [操作系统版本和维护级别](#)。

### 5. Use the AMATERSE utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.

### 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：**请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。

## **z/OS** *Collecting information for shared queue problems on z/OS*

If you need assistance from IBM Support to resolve a shared queue problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

## Before you begin

Before you start this task, answer the following questions about the problem:

- What is the name of the shared queue having a problem?
- What is the name of the IBM MQ Coupling Facility structure (CFSTRUCT) associated with the problematic shared queue?
- What is the message id associated with the problem?
- What is the name of the queue sharing group?
- What time did the problem occur?
- Which queue manager in the queue sharing group is involved?

## About this task

If you can reproduce the shared queue problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后，可以将其发送到 IBM。

## Procedure

1. Generate the following traces while the problem is happening:

- a. [Generate a GTF trace.](#)
- b. [Generate a MSTR internal trace.](#)
- c. [Generate a CHIN trace.](#)

2. Collect the data.

The following steps include an example of dumping both application structure and the IBM MQ Administration structure.

a) Record the version, release, and maintenance levels your software:

- IBM MQ: find the version in the CSQY000I message in the MSTR job log.
- The z/OS operating system: find the version in the output of /D IPLINFO in SDSF.
- Any other products involved with the problem: look for the version in the job log for the product.

b) Collect the IBM MQ MSTR and CHIN joblogs and, optionally, collect the syslog.

c) Collect a z/OS LOGREC report.

See [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS”](#) on page 265.

d) Save any [z/OS dumps](#) that you generated from IBM MQ.

IBM MQ dumps are located in a system dump data set and can be identified by their title. The title for a dump requested by IBM MQ starts with the four-character subsystem name of the queue manager. For example:

```
CSQ1,ABN=5C6-00E20016,U=SYSOPR ,C=MQ900.910.DMC
-CSQIALLC,M=CSQGFRCV,LOC=CSQSLD1 .CSQSVSTK+00000712
```

Comm dumps might not contain the queue manager name, depending on the comment specified in the dump command. Check the syslog for an [IEA611I](#) or [IEA911E](#) message to determine the dump data set name and also to see whether the dump is complete or partial. For example:

```
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) Collect Coupling Facility Structure dumps for the application Structure and the IBM MQ Administration Structure:

```
/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
```

where *ssid* is the subsystem ID for the queue manager.

An example of *QSGnameStructurename* is QSG1APPLICATION where QSG1 is the queue sharing group name.

An example of *QSGnameCSQ\_ADMIN* is QSG1CSQ\_ADMIN.

The following example shows dumping the application structure solely:

```
/DUMP COMM=(title)
/R xx,STRLIST=(STRNAME=QSGnameStructurename,(LISTNUM=ALL,
ADJUNCT=CAPTURE,ENTRYDATA=UNSER),EVENTQS,(EMCONTROLS=ALL))
```

An example of *QSGnameStructurename* is QSG1APPLICATION where QSG1 is the queue sharing group name.

3. Use the AMATERSE utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
4. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for shutdown problems on z/OS**

If you need assistance from IBM Support to resolve a shutdown problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### **Before you begin**

Before you start this task, answer the following questions about the problem:

- Was shutdown accompanied by an abend in either the queue manager or channel initiator ?
- Was shutdown associated with an apparent wait or hang ?
- Was any rise in CPU noted or possible loop condition?

Investigate the following:

- Search the [IBM MQ Support site](#) for known problems.
- See [Troubleshooting MQ channels](#) for help in correcting problems with any channels..

### **About this task**

If you can reproduce the shutdown problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后，可以将其发送到 IBM。

## Procedure

1. If shutdown abends, provide the dump, otherwise generate the dumps and job logs while the problem is happening:

See [“IBM MQ for z/OS abends” on page 228](#) for information on the IBM MQ for z/OS abend codes issued.

- a. [“Using the z/OS DUMP command” on page 248](#)

See [“IBM MQ for z/OS dumps” on page 246](#) for additional information.

- b. SLIP dump - see [Collecting documentation to submit to the IBM Support Center](#).
- c. You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

2. You might require the following information:

- a) An operations log

See [OPERLOG](#) for more information.

- b) A LOGREC report

See [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS” on page 265](#).

**Note:** You also need the version and release number of the product. See step [“4” on page 360](#) for information on how you find this information.

3. Collect the IBM MQ data.

4. Record the level of the product for IBM MQ for z/OS and any other product.

See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS

Record the [Operating system version and maintenance level](#) of your system.

5. Use the AMATERSE utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.

6. Send the information that you have collected to 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。

**Note:** 请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。

### **Collecting information for startup problems on z/OS**

If you need assistance from IBM Support to resolve a startup problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

## Before you begin

Before you start this task, answer the following questions about the problem:

- What startup problem did you observe on the system?
- What changes might have been made prior to the problem?
- Was this a new install or migration, and is it within a queue-sharing group?

Investigate the following:

- Search the [IBM MQ Support site](#) for known problems.

## About this task

If you can reproduce the startup problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后，可以将其发送到 IBM。

## Procedure

1. If startup abends, provide the dump, otherwise generate the dumps and job logs while the problem is happening:

See [“IBM MQ for z/OS abends” on page 228](#) for information on the IBM MQ for z/OS abend codes issued.

- a. [“Using the z/OS DUMP command” on page 248](#)

See [“IBM MQ for z/OS dumps” on page 246](#) for additional information.

- b. A LOGREC report, see [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS” on page 265](#).

**Note:** You also need the version and release number of the product. See step [“4” on page 361](#) for information on how you find this information.

- c. You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

2. You might require the following information if you cannot generate any dumps.

- a) A SLIP trap, which will be supplied by IBM Service.

See [Collecting documentation to submit to the IBM Support Center](#) for more information.

3. Collect the IBM MQ data.

4. Record the level of the product for IBM MQ for z/OS and any other product.

See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS

Record the [Operating system version and maintenance level](#) of your system.

5. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.

6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for TLS problems on z/OS**

If you need assistance from IBM Support to resolve a TLS problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

## Before you begin

Before you start this task, answer the following questions about the problem:

- Was the TLS issue associated with a channel ?
- If not associated with a channel, what specific error (message) occurred ?
- Did the issue seem to be associated with the currency of the TLS key ring ?



Investigate the following:

- Search the [IBM MQ Support site](#) for known problems.
- Provide the expected chain-of-trust on both sides of the TLS channel.
- For messages [CSQX633E](#) or [CSQX634E](#) displays of the key ring are useful.

## About this task

If you can reproduce the TLS problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后，可以将其发送到 IBM。

## Procedure

1. Generate the following information while the problem is happening:

a. A CHIN internal trace

See [Generate a CHIN trace](#)

b. A TLS trace

See [Generating System TLS \(formerly SSL\) trace on IBM MQ for z/OS](#)

c. RACDCERT output

See [TLS Digital Certificate Commands for External Security Managers:](#)

2. You might require the following information:

a) You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs.](#)

b) z/OS TCP/IP PKTTRACE.

See [How to collect Packet Traces and other TCP/IP related traces on z/OS](#) for more information.

c) A LOGREC report

See [A LOGREC report](#)

**Note:** You also need the version and release number of the product. See step “4” on [page 362](#) for information on how you find this information.

d) Output from the following commands:

### Displaying the RACF view of the channel initiator key ring:

Issue the following command:

```
RACDCERT ID(ssidCHIN) LISTRING(key ring)
```

### Displaying the RACF view of a specific certificate:

Issue the following command:

```
RACDCERT ID(ssidCHIN) list(LABEL('...'))
```

### Displaying certificate authorities:

Issue the following command:

```
RACDCERT CERTAUTH LIST
```

3. Collect the IBM MQ data.

4. Record the level of the product for IBM MQ for z/OS and any other product.

See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS

Record the [Operating system version and maintenance level](#) of your system.

5. Use the `AMATERSE` utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for triggering channels problems on z/OS**

If you need assistance from IBM Support to resolve a triggering channels problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### **Before you begin**

Before you start this task, answer the following questions about the problem:

- What problem did you observe on the system?
- What time did the problem start and has it stopped?
- Which queue managers, channels, remote queues and transmission queues are involved?
- What message number and error codes did you receive?

Investigate the following:

- Search the [IBM MQ Support site](#) for known problems. using, for example, any error messages or return codes.
- Set a short disconnect interval on the associated channel, The disconnect interval setting stops the channel quickly, with triggering enabled, and make debugging easier.
- For the channel initiator:
  - Make sure that the channel initiator is running as it is the most important piece of the triggering process.
  - On z/OS channel initiator processing is provided by the CHIN started task. You can start the channel initiator using the `START CHINIT` command, for example:

```
START CHINIT INITQ(SYSTEM.CHANNEL.INITQ)
```
  - Make sure that the channel initiator is monitoring the initiation queue, not the transmission queue.
  - Check the channel initiator log for channel error messages.
- Try to start the channel manually. If the channel fails to start, or does not successfully move the message from the transmission queue to the remote queue manager, then this is channel problem. See [“Collecting information for channel problems on z/OS”](#) on page 330 for more information.

### **About this task**

If you can reproduce the problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后, 可以将其发送到 IBM。

### **Procedure**

1. Collect the following required information:
  - a) Job logs

You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

b) Dumps generated at point of failure.

See [“IBM MQ for z/OS dumps” on page 246](#)

c) Dumps taken at the point of failure.

See [“Using the z/OS DUMP command” on page 248](#)

d) A LOGREC report

See [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS” on page 265](#).

**Note:** You also need the version and release number of the product. See step [“4” on page 364](#) for information on how you find this information.

e) A full display of the related object definitions:

- Queue
- Remote Transmission queue
- Initiation queue
- Sender channel
- Process, if a process definition is used, although you do not need the actual object definition.

2. Optionally, generate the following traces while the problem is happening:

a. [Generate a GTF trace](#).

b. [Generate a MSTR internal trace](#).

c. [Generate a CHIN trace](#).

3. Collect the IBM MQ data.

4. Record the level of the product for IBM MQ for z/OS and any other product.

See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS

Record the [Operating system version and maintenance level](#) of your system.

5. Use the AMATERSE utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.

6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for triggering programs problems on z/OS**

If you need assistance from IBM Support to resolve a triggering programs problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### **Before you begin**

Before you start this task, answer the following questions about the problem:

- What problem did you observe on the system?
- What time did the problem start and has it stopped?
- Which queue managers, channels, remote queues and transmission queues are involved?

- What message number and error codes did you receive?

Investigate the following:

- Search the [IBM MQ Support site](#) for known problems. using, for example, any error messages or return codes.
- For the trigger monitor:
  - Make sure that the trigger monitor is running and monitoring the initiation queue, not the transmission queue.
  - Verify that the user ID used to start the trigger monitor has the authority to access the entire path to the executable code.
- Verify that your applications are putting their messages to the trigger queue, not the initiation queue.
- Try to start the trigger program manually using the string specified in the APPLICID property of your process definition. See [DEFINE PROCESS](#) for more information.
- Check for security errors in your External Security Manager, for example, RACF.
- Check logs for your External Security Manager to see if there is a security failure.

## About this task

If you can reproduce the problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后，可以将其发送到 IBM。

## Procedure

1. Collect the following required information:

a) Job logs

You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

b) Dumps generated at point of failure.

See [“IBM MQ for z/OS dumps” on page 246](#)

c) Dumps taken at the point of failure.

See [“Using the z/OS DUMP command” on page 248](#)

d) A LOGREC report

See [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS” on page 265](#).

**Note:** You also need the version and release number of the product. See step [“4” on page 365](#) for information on how you find this information.

e) A full display of the related object definitions:

- Trigger queue
- Initiation queue
- Process

2. Optionally, generate the following traces while the problem is happening:

a. [Generate a GTF trace.](#)

b. [Generate a MSTR internal trace.](#)

c. [Generate a CHIN trace.](#)

3. Collect the IBM MQ data.

4. Record the level of the product for IBM MQ for z/OS and any other product.

See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS

Record the [Operating system version and maintenance level](#) of your system.

5. Use the [AMATERSE](#) utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
6. Send the information that you have collected to 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。

**Note:** 请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。

## **Collecting information for wait, hang, or loop problems on z/OS**

If you need assistance from IBM Support to resolve a wait, hang, or loop problem on IBM MQ for z/OS, you first need to collect troubleshooting information to send to IBM Support to help find a solution.

### **Before you begin**

Before you start this task, answer the following questions about the problem:

- What wait, hang, or loop problem did you observe on the system?
- What time did the problem start and when did it stop?
- Which queue managers, channels, remote queues and transmission queues are involved?
- What message number and error codes did you receive?

Search the [IBM MQ Support site](#) for known problems. using, for example, any error messages or return codes.

### **About this task**

If you can reproduce the problem or the problem is happening right now, you can generate data to provide more information about the problem.

收集故障诊断信息后, 可以将其发送到 IBM。

### **Procedure**

1. Collect the following required information:

- a) Job logs

You require the Syslog, MSTR job log, and CHIN job log.

The job logs are named xxxxMSTR and xxxxCHIN, where xxxx is the IBM MQ subsystem identifier (SSID). See [Creating a print data set containing the JES2 joblog for the IBM MQ for z/OS jobs](#).

- b) Dumps generated at point of failure.

See [“IBM MQ for z/OS dumps” on page 246](#)

- c) Dumps taken at the point of failure.

See [“Using the z/OS DUMP command” on page 248](#)

- d) A LOGREC report

See [A LOGREC report](#) and [“SYS1.LOGREC information on z/OS” on page 265](#).

**Note:** You also need the version and release number of the product. See step [“4” on page 367](#) for information on how you find this information.

2. Optionally, generate the following traces while the problem is happening:

- a. [Generate a GTF trace](#).

- b. [Generate a MSTR internal trace.](#)
- c. [Generate a CHIN trace.](#)
3. Collect the IBM MQ data.
4. Record the level of the product for IBM MQ for z/OS and any other product.  
See message [CSQY000I](#) in the MSTR job log for IBM MQ for z/OS  
Record the [Operating system version and maintenance level](#) of your system.
5. Use the AMATERSE utility before uploading to ECUREP, and ensure you specify the Case number with which the data is associated.
6. Send the information that you have collected to 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。

**Note:** 请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。

## 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 281 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『[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。

 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:

-  第 369 页的『[AIX, Linux, and Windows 上的错误日志](#)』
-  第 372 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，请参阅第 224 页的『[Troubleshooting IBM MQ for z/OS problems](#)』。

有关 IBM MQ classes for JMS 中的错误日志的信息，请参阅第 375 页的『[IBM MQ classes for JMS 中的错误日志](#)』。

## 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 375 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

**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，消息严重性将作为单个大写字母字符追加到消息号，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

**Multi**

消息时间包含在 ISO 8601 格式中，而不是包含在本地时间中。

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

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 376 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 387 页的『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。

[第 279 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。

ALW

## AIX, Linux, and Windows 上的错误日志

在 AIX, Linux, and Windows 上安装 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 AIX 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 371 页的『AIX, 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 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 AIX 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。

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 370 页的图 56 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 56: 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 AIX and Linux 系统下的错误代码

 在 AIX 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 375 页的『Error logs on z/OS』](#)

On z/OS, error messages are written to the z/OS system console and the channel-initiator job log.

[第 376 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 387 页的『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。

### 相关参考

[第 372 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。

[消息和原因码](#)

## ALW AIX, 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 371 页的表 25 中所示。

平台	目录
<b>Linux</b> <b>AIX</b> and Linux 系统	/var/mqm/qmgrs/ <i>qmname</i> /errors
<b>Windows</b> 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 371 页的表 26 中所示。

平台	目录
<b>Linux</b> <b>AIX</b> and Linux 系统	/var/mqm/errors
<b>Windows</b> 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 372 页的表 27 中所示。

表 27: 客户机错误日志目录	
平台	目录
Linux and Linux 系统 AIX AIX	/var/mqm/errors
Windows 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。

有关配置文件的更多信息，请参阅 [更改多平台上的 IBM MQ 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 QMQMADM 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 374 页的图 57 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 57: 来自 IBM MQ 错误日志的摘录

### 相关概念

第 369 页的『[AIX, Linux, and Windows 上的错误日志](#)』

在 AIX, Linux, and Windows 上安装 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 375 页的『[Error logs on z/OS](#)』

On z/OS, error messages are written to the z/OS system console and the channel-initiator job log.

第 376 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 387 页的『[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

[消息和原因码](#)

On z/OS, error messages are written to the z/OS system console and the channel-initiator job log.

If you are using the z/OS message processing facility to suppress messages, the console messages might be suppressed. See [Planning your IBM MQ environment on z/OS](#).

### Related concepts

“Diagnostic information produced on IBM MQ for z/OS” on page 231

Use this topic to investigate some of the diagnostic information produced by z/OS that can be useful in problem determination and understand how to investigate error messages, dumps, console logs, job output, symptom strings, and queue output.

“Other sources of problem determination information for IBM MQ for z/OS” on page 233

Use this topic to investigate other sources of information for IBM MQ for z/OS problem determination.

[IBM MQ for z/OS messages, completion, and reason codes](#)

“AIX, Linux, and Windows 上的错误日志” on page 369

在 AIX, Linux, and Windows 上安装 IBM MQ 时创建的 errors 子目录最多可以包含三个错误日志文件。

### Related reference

“IBM i 上的错误日志” on page 372

使用此信息来了解 IBM MQ for IBM i 错误日志。

## IBM MQ classes for JMS 中的错误日志

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，包含日志的文件称为 `mqjms.log`，并且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 配置文件的更多信息，请参阅 [IBM MQ classes for JMS/Jakarta Messaging 配置文件](#)。有关 `com.ibm.msg.client.commonservices.log.outputName` 属性的有效值的更多信息，请参阅 [第 88 页的『IBM MQ classes for JMS 的日志记录错误』](#)。

### 相关参考

[JMS 异常消息](#)

## 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：

- 在 `qm.ini` 文件的 `QLErrorLog` 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

- 要使用 `qm.ini` 文件中的 `QLErrorLog` 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要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。

– **Linux** **AIX** 在 AIX and Linux 上：

```
export MQ_CHANNEL_SUPPRESS_MSGS=9999,9002,9209
```

– **Windows** 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

[AIX, Linux, and Windows 上的 QMErrorLog 节](#)

### 相关参考

[环境变量描述](#)

[队列管理器属性](#)

## 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 382 页的『FFST: IBM MQ classes for JMS』](#)
-  [第 380 页的『FFST: IBM MQ for Windows』](#)
-   [第 377 页的『FFST: IBM MQ for AIX or Linux』](#)
-  [第 378 页的『FFST: IBM MQ for IBM i』](#)

#### 相关概念

[第 367 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 387 页的『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 279 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

AIX

## FFST: IBM MQ for AIX or Linux

描述 First Failure Support Technology (FFST) 的名称，位置和内容 AIX and Linux 系统的文件。

在 IBM MQ for AIX or Linux 系统上，FFST 信息将记录在 `/var/mqm/errors` 目录内的文件中。

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 AIX 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#fa...
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。



## 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 381 页的图 58](#) 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 58: 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。

## 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。无论您是使用 Jakarta Messaging 3.0 还是 JMS 2.0，都会记录 FFST 信息。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。有关更多信息，请参阅 [IBM MQ classes for JMS/Jakarta Messaging 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\JMCC0007.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
ccl=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
...

WorkQueueManager 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

Detailed JMS System Information
Connections      :
|  Instance          :- com.ibm.msg.client.wmq.internal.WMQXAConnection@bd4b665a
|  connectOptions    :- version:5 options:64 clientConn:
[channelName:'MY.SVRCONN' version:10 channelType:6 transportType:2 desc:'<null>'
qMgrName:'test' xmitQName:'<null>' connectionName:'9.20.124.119(1414)' mcaName:'<null>'
modeName:'<null>' tpName:'<null>' batchSize:50 discInterval:6000 shortRetryCount:10
shortRetryInterval:60 longRetryCount:999999999 longRetryInterval:1200

```





```

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(SimpleWorkQueueItem.java:105)
:
com.ibm.msg.client.commonservices.workqueue.WorkQueueItem.run(WorkQueueItem.java:229)
:
com.ibm.msg.client.commonservices.workqueue.WorkQueueManager.runWorkQueueItem(WorkQueueManager.java:303)
:
com.ibm.msg.client.commonservices.j2se.workqueue.WorkQueueManagerImplementation$ThreadPoolWorker.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。

## FFST:WCF XMS First Failure Support Technology

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。

### 相关任务

[第 465 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 220 页的『对 IBM MQ 问题的 WCF 定制通道进行故障诊断』](#)

用于帮助您解决 IBM MQ 应用程序的 Microsoft Windows Communication Foundation (WCF) 定制通道运行问题的故障诊断信息。

[第 279 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](#)

## 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。

## 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运行跟踪。

- ▶ **Linux** ▶ **AIX** [第 388 页的『AIX and Linux 上的跟踪』](#)
- ▶ **IBM i** [第 393 页的『IBM i 上的跟踪』](#)
- ▶ **Windows** [第 402 页的『Windows 上的跟踪』](#)
- ▶ **z/OS** [第 405 页的『Tracing on z/OS』](#)

- [第 422 页的『跟踪高级消息排队协议 \(AMQP\) 服务』](#)
- [第 424 页的『跟踪 IBM MQ Console』](#)
- [第 427 页的『跟踪 IBM MQ Internet Pass-Thru 中的错误』](#)
- [第 428 页的『跟踪 IBM MQ.NET 应用程序』](#)
- [第 433 页的『跟踪 JMS/Jakarta Messaging 和 Java 应用程序』](#)
- **Multi** [第 443 页的『在多平台上跟踪 Managed File Transfer 资源』](#)
- **z/OS** [第 449 页的『Tracing Managed File Transfer for z/OS resources』](#)
- [第 463 页的『跟踪 REST API』](#)
- [第 464 页的『跟踪 runmqakm』](#)
- [第 465 页的『跟踪 IBM MQ 的 WCF 定制通道』](#)
- [第 465 页的『跟踪 XMS .NET 应用程序』](#)
- [第 471 页的『启用 LDAP 客户机库代码的动态跟踪』](#)

### 相关概念

[第 367 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 376 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 279 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 AIX and Linux 上的跟踪

在 AIX and Linux 上，可以使用 **strmqtrc** 和 **endmqtrc** 命令来启动和结束跟踪，并使用 **dspmqtrc** 来显示跟踪文件。在 AIX 上，除了使用 **strmqtrc** 和 **endmqtrc** 命令外，还可以使用 AIX 系统跟踪。

### 开始之前

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。

### 关于此任务

跟踪可以捕获许多 IBM MQ 问题的根本原因。为了避免影响性能或生成过大的跟踪文件并填充磁盘，限制跟踪处于活动状态的时间很重要。

在 AIX and Linux 上，使用以下命令来控制跟踪工具：

#### strmqtrc

在 AIX and Linux 系统上，使用 **strmqtrc** 控制命令启用或修改跟踪。该命令具有可选参数，用于指定所需的跟踪级别。例如：

- 对于选择性组件跟踪，请使用 **-t** 和 **-x** 参数来控制要记录的跟踪详细信息量。使用 **-t** 参数指定要跟踪的点，或使用 **-x** 参数指定不希望跟踪的点。如果需要，您可以使用不同的参数同时运行多个跟踪。
- 对于选择性进程跟踪，请使用 **-p** 参数将跟踪生成限制为指定的指定进程。
- 如果需要阻止 IBM MQ 在其跟踪中包含任何消息数据，请使用 **-d 0** 参数。

有关更多信息，请参阅 [strmqtrc \(启动跟踪\)](#)。

## endmqtrc

要停止跟踪，请使用 **endmqtrc** 控制命令。**endmqtrc** 命令仅结束由其参数描述的跟踪。使用不带参数的 **endmqtrc** 将结束对所有进程的早期跟踪。有关更多信息，请参阅 [endmqtrc \(结束跟踪\)](#)。

## dspmqtrc filename

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。有关更多信息，请参阅 [dspmqtrc \(显示格式化跟踪\)](#)。

**Linux** 在 Linux x86-64 系统上，您也可以使用 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。

跟踪功能使用几个文件，它们是：

- 要跟踪的每个实体都有一个文件，在该文件中记录跟踪信息。
- 每台机器上的一个附加文件，用于提供用于启动和结束跟踪的共享内存的引用。
- 一个文件，用于标识更新共享内存时使用的信号量。

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。

**注：**确保 `/var/mqm/trace` 目录位于不同于包含队列管理器的文件系统的本地文件系统上。否则，如果 IBM MQ 跟踪填充队列管理器正在使用的文件系统，那么您将面临中断的风险。跟踪目录必须有足够的可用空间，因为部分跟踪可能不包含 IBM 支持人员解决问题所需的信息。

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本示例中显示的更少或更多的数字。
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。

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。

**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`。

## 过程

1. 如果在单个 AIX 或 Linux 系统上具有多个 IBM MQ 安装，请确保为要跟踪的安装设置了您的环境。

例如，如果要跟踪与 `Installation1` 关联的队列管理器或应用程序，那么必须先使用 **setmqenv** 命令切换到 `Installation1`，然后再运行任何命令以启动、停止或格式化跟踪，否则跟踪将不会捕获任何有用信息。

检查 `/etc/opt/mqm/mqinst.ini` 文件以确定系统上 IBM MQ 安装的位置。使用 **setmqenv** 命令为要跟踪的安装设置环境。

例如：

### 使用队列管理器 QMA 为安装设置环境

```
/opt/mq93/bin/setmqenv -m QMA
```

### 设置安装环境 Installation2

```
/opt/mq93/bin/setmqenv -n Installation2
```

2. 使用 **strmqtrc** 命令启动跟踪。

缺省情况下，启用所有跟踪点并生成缺省详细信息跟踪。传递到 **strmqtrc** 的参数可控制哪些进程生成跟踪文件以及它们跟踪的数据类型。例如：

#### 启动队列管理器 QMA 的缺省详细信息跟踪

```
strmqtrc -m QMA
```

#### 仅使用与使用传输层安全性 (TLS) 通道安全性相关联的输出数据启动队列管理器 QMA 的跟踪

```
strmqtrc -m QMA -t ssl
```

#### 在不包含消息数据的情况下启动队列管理器 QMB 的缺省详细信息跟踪

```
strmqtrc -m QMB -d 0
```

#### 对包含所有消息数据的名为 amqsput 的进程启动 API 跟踪

```
strmqtrc -t api -p amqsput -d all
```

#### 使用以 100MB 为包装的文件启动所有内容的高详细跟踪

```
strmqtrc -e -t all -t detail -l 100
```

**AIX** 对于 AIX 上的选择性组件跟踪，请使用环境变量 **MQS\_TRACE\_OPTIONS** 来分别激活高详细信息和参数跟踪功能。由于 **MQS\_TRACE\_OPTIONS** 允许跟踪处于活动状态，而不具有高详细信息和参数跟踪功能，因此当您尝试在启用跟踪的情况下重现问题时，可以使用它来降低对性能和跟踪大小的影响。通常，必须在启动队列管理器的进程中设置 **MQS\_TRACE\_OPTIONS**，并且必须在启动队列管理器之前设置，否则无法识别此队列管理器。在启动跟踪之前设置 **MQS\_TRACE\_OPTIONS**。如果在跟踪启动之后设置该变量，那么其无法识别。

注：仅当 IBM 支持人员指示您设置环境变量 **MQS\_TRACE\_OPTIONS** 时才设置该环境变量。

3. 在跟踪处于活动状态时重现该问题，使每次尝试保持跟踪尽可能短。

要检查跟踪状态，请使用 **strmqtrc** 命令的 **-s** 参数：

```
strmqtrc -s
```

4. 使用 **endmqtrc** 命令停止跟踪。

一旦发生问题，立即停止跟踪。如果您等待停止跟踪，那么 IBM 支持人员需要的数据可能会丢失或被覆盖。您可以一次性停止所有跟踪，也可以单独停止每个跟踪。例如：

#### 停止安装中的所有跟踪

```
endmqtrc -a
```

#### 停止队列管理器 QMA 的跟踪

```
endmqtrc -m QMA
```

#### 停止名为 amqsput 的进程的跟踪

```
endmqtrc -p amqsput
```

5. 使用 **dspmqtrc** 命令格式化跟踪。

在 AIX and Linux 上，IBM MQ 生成的跟踪文件是二进制文件，必须先进行格式化，然后才能读取这些文件。二进制跟踪文件命名为类似于 **AMQ\*.TRC**，如果启动了回绕跟踪，那么某些文件也可能命名为类似于 **AMQ\*.TRS**：

```
cd /var/mqm/trace
dspmqtrc AMQ*.TR?
```

6. 清除跟踪目录。

清除 **/var/mqm/trace** 目录以回收空间，并确保您已准备好在将来生成新的跟踪 (如果需要)。只要您与 IBM 支持人员一起工作，就保存二进制跟踪文件的副本，但不要将二进制跟踪和格式化跟踪都发送到

IBM。在将跟踪文件发送到 IBM 之前，请删除二进制跟踪文件 (AMQ.SSL.TRC 和 AMQ.SSL.TRC.1 除外，如果存在)。

```
cd /var/mqm/trace
rm -f AMQ*?.TR?
```

7. 收集与问题相关的跟踪和任何其他故障诊断信息。

如果您正在使用 **runmqras** 命令来收集 IBM MQ 故障诊断信息，并且需要包含跟踪数据，那么必须在为 **-section** 参数指定的节名称中包含 **trace**。如果跟踪非常大，那么可能需要使用 **-workdirectory** 选项将文件打包到具有大量可用空间的目录中。有关更多信息，请参阅第 281 页的『使用 runmqras 自动收集故障诊断信息』和第 367 页的『将故障诊断信息发送到 IBM』。

8. 将针对相同问题收集的跟踪和任何其他故障诊断信息发送到 IBM。

有关更多信息，请参阅第 367 页的『将故障诊断信息发送到 IBM』。

### 相关概念

第 377 页的『FFST: IBM MQ for AIX or Linux』

描述 First Failure Support Technology (FFST) 的名称，位置和内容 AIX and Linux 系统的文件。

### 相关任务

第 280 页的『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。

## 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。

Linux

AIX

## AIX and Linux 的跟踪数据示例

从 AIX and Linux 的跟踪文件中抽取。

### AIX 的示例

AIX

第 392 页的图 59 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 7.3
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 cbminindex:-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 59: 样本 IBM MQ for AIX 跟踪

### Linux 的示例

Linux

第 393 页的图 60 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 60: 样本 IBM MQ for Linux 跟踪

## 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，而某些选项不适用于独立客户机。

### 关于此任务

在 IBM i 上，服务器和客户机都支持通过调用 **QMQM/STRMQTRC** 和 **QMQM/ENDMQTRC** 程序在 IBM i 命令行上进行跟踪，并且都支持使用 **STRMQTRC**，**ENDMQTRC** 和 **DSPMQTRC** 命令在 IBM i Qshell 上进行跟踪。

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。

### 相关概念

第 378 页的『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。

### 相关任务

第 280 页的『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。

## IBM i 将跟踪与 IBM i 上的 IBM MQ 服务器配合使用

如果在 IBM i 上安装了 IBM MQ 服务器，那么可以使用 **TRCMQM** 命令来启动和停止跟踪，并指定所需的跟踪类型。或者，您可以通过调用 **QMQM/STRMQTRC** 和 **QMQM/ENDMQTRC** 程序或者在 IBM i Qshell 中使用 **STRMQTRC**，**ENDMQTRC** 和 **DSPMQTRC** 命令从 IBM i 命令行控制跟踪。

### 关于此任务

跟踪可以捕获许多 IBM MQ 问题的根本原因。为了避免影响性能或生成过大的跟踪文件并填充磁盘，限制跟踪处于活动状态的时间很重要。

通过早期跟踪，可以跟踪队列管理器的创建和启动。但是，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现的。

如果在 IBM i 上安装了 IBM MQ 服务器，那么 **TRCMQM** 命令是启动和停止跟踪的最常用方法。您可以使用可选参数来指定所需的跟踪级别。例如：

- 要指定所需的详细信息级别，请使用 **TRCLEVEL** 参数。
- 要指定所需的跟踪输出类型，请使用 **OUTPUT** 参数。
- 对于选择性组件跟踪，请使用 **TRCTYPE** 和 **-EXCLUDE** 参数来减少正在保存的跟踪数据量，从而提高运行时性能。使用 **TRCTYPE** 参数指定要跟踪的点，或使用 **-EXCLUDE** 参数指定不希望跟踪的点。如果省略 **TRCTYPE** 参数，那么将启用所有跟踪点。如果省略 **-EXCLUDE** 参数，那么将启用 **TRCTYPE** 中指定的所有跟踪点。
- 要合并跟踪并指定要用于收集的跟踪记录的最大存储器大小，请使用 **MAXSTG** 参数。
- 对于选择性进程跟踪，请使用 **JOB** 参数将跟踪生成限制为指定的指定进程。
- 如果需要阻止 IBM MQ 在其跟踪中包含任何消息数据，请使用 **DATASIZE(\*NONE)** 参数。

有关更多信息，请参阅 [TRCMQM \(跟踪 MQ\)](#)。

注：您可以手动输入 **TRCMQM** 命令（如本任务中的示例所示），也可以输入 **TRCMQM** 并按 F4，后跟 F9 以使用 **TRCMQM** 面板启用跟踪。

您还可以使用以下命令来控制跟踪工具：

#### 启动跟踪

您可以通过调用 **QMQM/STRMQTRC** 程序在 IBM i 命令行上启动跟踪，也可以通过使用 **STRMQTRC** 命令在 IBM i Qshell 上启动跟踪。您可以使用可选参数来指定所需的跟踪级别。例如：

- 对于选择性组件跟踪，请使用 **-t** 和 **-x** 参数来控制要记录的跟踪详细信息量。使用 **-t** 参数指定要跟踪的点，或使用 **-x** 参数指定不希望跟踪的点。如果需要，您可以使用不同的参数同时运行多个跟踪。如果提供多个跟踪类型，那么每个跟踪类型都必须具有自己的 **-t** 或 **-x** 标志。您可以包含任意数目的 **-t** 或 **-x** 标志（如果每个标志都具有与其关联的有效跟踪类型）。在多个 **-t** 或 **-x** 标志上指定相同的跟踪类型不是错误。
- 对于选择性进程跟踪，请使用 **-p** 参数将跟踪生成限制为指定的指定进程。
- 如果需要阻止 IBM MQ 在其跟踪中包含任何消息数据，请使用 **-d 0** 参数。

有关更多信息，请参阅 [strmqtrc \(启动跟踪\)](#)。

#### 结束跟踪

您可以通过调用 **QMQM/ENDMQTRC** 程序在 IBM i 命令行上结束跟踪，也可以通过使用 **ENDMQTRC** 命令在 IBM i Qshell 上结束跟踪。您可以通过不指定参数来结束所有进程的早期跟踪，也可以通过包含相应参数来结束特定类型的跟踪。有关更多信息，请参阅 [endmqtrc \(结束跟踪\)](#)。

#### 格式化跟踪

IBM MQ 生成的跟踪文件是二进制文件。您可以在 IBM i Qshell 中使用 **DSPMQTRC** 命令来格式化跟踪输出，然后再将其发送到 IBM。有关更多信息，请参阅 [dspmqtrc \(显示格式化跟踪\)](#)。

### 过程

1. 启动跟踪。

缺省情况下，启用所有跟踪点并生成缺省详细信息跟踪。要控制哪些进程生成跟踪文件以及它们跟踪的数据类型，请指定相应的参数。

例如：

### 启动缺省详细信息跟踪

- 使用 **TRCMQM** 命令：

```
TRCMQM SET(*ON)
```

- 使用命令行进行安装的步骤如下：

```
CALL PGM(QMQM/STRMQTRC)
```

- 使用 Qshell：

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM
```

### 启动队列管理器 **QMA** 的缺省详细信息跟踪

- 使用 **TRCMQM** 命令：

```
TRCMQM SET(*ON) MQMNAME(QMA)
```

- 使用命令行进行安装的步骤如下：

```
CALL PGM(QMQM/STRMQTRC) PARM('-m' 'QMA')
```

- 使用 Qshell：

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM -m QMA
```

### 在不包含消息数据的情况下启动队列管理器 **QMB** 的缺省详细信息跟踪

- 使用 **TRCMQM** 命令：

```
TRCMQM SET(*ON) MQMNAME(QMB) DATASIZE(*NONE)
```

- 使用命令行进行安装的步骤如下：

```
CALL PGM(QMQM/STRMQTRC) PARM('-m' 'QMB' '-d' '0')
```

- 使用 Qshell：

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM -m QMB -d 0
```

### 使用所有数据启动名为 **AMQSPUT** 的作业的 **API** 跟踪

- 使用 **TRCMQM** 命令：

```
TRCMQM SET(*ON) TRCTYPE(*API) JOB(AMQSPUT) DATASIZE(*ALL)
```

- 使用命令行进行安装的步骤如下：

```
CALL PGM(QMQM/STRMQTRC) PARM('-t' 'api' '-p' 'AMQSPUT' '-d' 'all')
```

- 使用 Qshell：

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM -t api -p AMQSPUT -d all
```

### 使用以 **16MB** 开头的文件启动高详细信息早期跟踪

- 使用 **TRCMQM** 命令：

```
TRCMQM SET(*ON) TRCEARLY(*YES) TRCLEVEL(*DETAIL) MAXSTG(16)
```

- 使用命令行进行安装的步骤如下：

```
CALL PGM(QMQM/STRMQTRC) PARM('-e' '-t' 'all' '-t' 'detail' '-l' '16')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM -e -t all -t detail -l 16
```

2. 在跟踪处于活动状态时重现该问题，使每次尝试保持跟踪尽可能短。

要检查跟踪状态，请使用下列其中一个选项:

- 使用 **TRCMQM** 命令的 **SET** 参数:

```
TRCMQM SET(*STS)  
WRKSPLF SELECT(QMQM)
```

使用选项 5 来查看包含跟踪状态的 **TRCMQM** 假脱机文件。

- 在命令行上，使用 **QMQM/STRMQTRC** 程序的 **-s** 参数:

```
CALL PGM(QMQM/STRMQTRC) PARM('-s')
```

- 在 Qshell 中，使用 **STRMQTRC** 命令的 **-s** 参数:

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM -s
```

3. 使用 **TRCMQM** 命令停止跟踪。

一旦发生问题，立即停止跟踪。如果您等待停止跟踪，那么 IBM 支持人员需要的数据可能会丢失或被覆盖。

例如:

### 停止所有跟踪

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/ENDMQTRC) PARM('-a')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/ENDMQTRC.PGM -a
```

**TRCMQM** 命令中没有等效项。

### 停止缺省详细信息跟踪

- 使用 **TRCMQM** 命令:

```
TRCMQM SET(*OFF)
```

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/ENDMQTRC)
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/ENDMQTRC.PGM
```

使用不带参数的 **endmqtrc** 与 **endmqtrc -e** 具有相同的效果。不能将 **-e** 参数与 **-m** 参数，**-i** 参数或 **-p** 参数一起指定。

### 停止队列管理器 QMA 的跟踪

- 使用 **TRCMQM** 命令:

```
TRCMQM SET(*OFF) MQMNAME(QMA)
```

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/ENDMQTRC) PARM('-m' 'QMA')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/ENDMQTRC.PGM -m QMA
```

### 停止名为 **AMQSPUT** 的作业的跟踪

- 使用 **TRCMQM** 命令:

```
TRCMQM SET(*OFF) JOB(AMQSPUT)
```

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/ENDMQTRC) PARM('-p' 'AMQSPUT')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/ENDMQTRC.PGM -p AMQSPUT
```

### 停止早期跟踪

- 使用 **TRCMQM** 命令:

```
TRCMQM SET(*OFF) TRCEARLY(*YES)
```

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/ENDMQTRC) PARM('-e')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM -e
```

#### 4. 可选: 格式化二进制跟踪文件。

IBM MQ 生成的跟踪文件是二进制文件, 必须先进行格式化, 然后才能读取这些文件。您可以将二进制跟踪发送到 IBM, 也可以在系统未处于高负载时在 Qshell 对其进行格式化。例如:

```
/QSYS.LIB/QMQM.LIB/DSPMQTRC.PGM [-t FormatTemplate] [-h] [-s]
[-o OutputFileName] InputFileName
```

其中:

- *InputFileName* 是必需参数, 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.TRC。
- *-t FormatTemplate* 指定包含如何显示跟踪的详细信息的模板文件的名称。缺省值为 /QIBM/ProdData/mqm/lib/amqtrc.fmt。
- *-h* 省略报告中的标题信息。
- *-s* 抽取跟踪头并将其放入 stdout。
- *-o output\_filename* 指定要将格式化数据写入其中的文件的名称。

还可指定 *dspmqtrc \** 以格式化所有跟踪。

如果自行格式化跟踪, 请先删除二进制跟踪文件, 然后再将文件发送到 IBM:

- 使用命令行来删除二进制跟踪文件:

```
RMVLNK OBJLNK('/QIBM/UserData/mqm/trace/AMQ*.*.TR?')
```

- 使用 Qshell 通过 **DSPMQTRC** 命令显示格式化的跟踪文件, 然后删除二进制跟踪文件:

```
cd /QIBM/UserData/mqm/trace
/QSYS.LIB/QMQM.LIB/DSPMQTRC.PGM AMQ*.*.TR?
rm -f AMQ*.*.TR?
```

#### 5. 收集与问题相关的跟踪和任何其他故障诊断信息。

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 281 页的『使用 runmqras 自动收集故障诊断信息』中所述。如果您正在使用 **runmqras** 命令来收集 IBM MQ 故障诊断信息, 并且需要包含跟踪数据, 那么必须在为 **-section** 参数指定的节名称中包含 **trace**。如果跟踪非常大, 那么可能需要使用 **-workdirectory** 选项将文件打包到具有大量可用空间的目录中。

必须在 Qshell 上运行 **runmqras** 命令, 如以下示例中所示:

```
/QIBM/ProdData/mqm/bin/runmqras -section trace,defs,cluster -caseno TS001234567 -qmlist QMA
```

- 或者, 手动收集数据, 如第 284 页的『手动收集故障诊断信息』中所述。
6. 将针对相同问题收集的跟踪和任何其他故障诊断信息发送到 IBM。  
有关更多信息, 请参阅第 367 页的『将故障诊断信息发送到 IBM』。
  7. 清除跟踪目录以回收空间, 并确保您已准备好在将来生成新的跟踪 (如果需要)。

- 使用命令行进行安装的步骤如下:

```
RMVLNK OBJLNK('/QIBM/UserData/mqm/trace/AMQ*.*')
```

- 使用 Qshell:

```
cd /QIBM/UserData/mqm/trace  
rm -f AMQ*.*
```

## 相关概念

第 378 页的『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。

## 相关任务

第 280 页的『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。

## IBM i 将跟踪与 IBM i 上的 IBM MQ 客户机配合使用

在 IBM i 上, 您可以在 IBM i 命令行上通过调用 **QMQM/STRMQTRC** 和 **QMQM/ENDMQTRC** 程序来控制对 IBM MQ MQI client 的跟踪, 或者在 IBM i Qshell 上使用 **STRMQTRC**, **ENDMQTRC** 和 **DSPMQTRC** 命令来控制对 Qshell 的跟踪。

## 开始之前

IBM i 上的 IBM MQ 服务器安装可以包含客户机, 或者您可以选择在没有 IBM MQ 服务器功能的 IBM i 上安装独立客户机。由于在独立客户机的情况下没有队列管理器, 因此不支持以下参数:

- 启动或结束跟踪命令上的 **-m** 参数。
- **runmqras** 上的 **-qmlist** 参数。

使用独立 IBM MQ 客户机时, 没有用于捕获跟踪的控制语言 (CL) 命令。

要使用 Qshell, 请在 IBM i 命令行中输入 **STRQSH**。您可以随时通过按 F3 键退出并返回到命令行。为了避免输入 **STRMQTRC.PGM** 和 **runmqras** 之类的命令的完整路径, 您可以将 IBM MQ 目录 **/QSYS.LIB/QMQM.LIB** 和 **/QIBM/ProdData/mqm/bin** 添加到 **PATH** 中, 如以下示例所示:

```
export PATH="$PATH:/QSYS.LIB/QMQM.LIB:/QIBM/ProdData/mqm/bin"
```

此任务中的示例提供了所有命令的完整路径名, 但如果更新 **PATH**, 那么无需输入这些名称。

## 关于此任务

跟踪可以捕获许多 IBM MQ 问题的根本原因。为了避免影响性能或生成过大的跟踪文件并填充磁盘，限制跟踪处于活动状态的时间很重要。

在 IBM i 上，使用以下命令来控制 IBM MQ 客户机的跟踪工具：

### 启动跟踪

您可以通过调用 **QMQM/STRMQTRC** 程序在 IBM i 命令行上启动跟踪，也可以通过使用 **STRMQTRC** 命令在 IBM i Qshell 上启动跟踪。您可以使用可选参数来指定所需的跟踪级别。例如：

- 对于选择性组件跟踪，请使用 **-t** 和 **-x** 参数来控制要记录的跟踪详细信息量。使用 **-t** 参数指定要跟踪的点，或使用 **-x** 参数指定不希望跟踪的点。如果需要，您可以使用不同的参数同时运行多个跟踪。如果提供多个跟踪类型，那么每个跟踪类型都必须具有自己的 **-t** 或 **-x** 标志。您可以包含任意数目的 **-t** 或 **-x** 标志 (如果每个标志都具有与其关联的有效跟踪类型)。在多个 **-t** 或 **-x** 标志上指定相同的跟踪类型不是错误。
- 对于选择性进程跟踪，请使用 **-p** 参数将跟踪生成限制为指定的指定进程。
- 如果需要阻止 IBM MQ 在其跟踪中包含任何消息数据，请使用 **-d 0** 参数。

有关更多信息，请参阅 [strmqtrc \(启动跟踪\)](#)。

### 结束跟踪

您可以通过调用 **QMQM/ENDMQTRC** 程序在 IBM i 命令行上结束跟踪，也可以通过使用 **ENDMQTRC** 命令在 IBM i Qshell 上结束跟踪。您可以通过不指定参数来结束所有进程的早期跟踪，也可以通过包含相应参数来结束特定类型的跟踪。有关更多信息，请参阅 [endmqtrc \(结束跟踪\)](#)。

### 格式化跟踪

IBM MQ 生成的跟踪文件是二进制文件。您可以在 IBM i Qshell 中使用 **DSPMQTRC** 命令来格式化跟踪输出，然后再将其发送到 IBM。有关更多信息，请参阅 [dspmqtrc \(显示格式化跟踪\)](#)。

## 过程

1. 通过调用 **QMQM/STRMQTRC** 程序在 IBM i 命令行上启动跟踪，或者通过使用 **STRMQTRC** 命令在 IBM i Qshell 上启动跟踪。

缺省情况下，启用所有跟踪点并生成缺省详细信息跟踪。要控制哪些进程生成跟踪文件以及它们跟踪的数据类型，请指定相应的参数。例如：

#### 启动缺省详细信息跟踪

- 使用命令行进行安装的步骤如下：

```
CALL PGM(QMQM/STRMQTRC)
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM
```

#### 启动队列管理器 QMA 的缺省详细信息跟踪

- 使用命令行进行安装的步骤如下：

```
CALL PGM(QMQM/STRMQTRC) PARM('-m' 'QMA')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM -m QMA
```

#### 在不包含消息数据的情况下启动队列管理器 QMB 的缺省详细信息跟踪

- 使用命令行进行安装的步骤如下：

```
CALL PGM(QMQM/STRMQTRC) PARM('-m' 'QMB' '-d' '0')
```

- 使用 Qshell:



```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM -m QMB -d 0
```

### 对包含所有数据的名为 **AMQSPUT** 的作业启动 **API** 跟踪

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/STRMQTRC) PARM('-t' 'api' '-p' 'AMQSPUT' '-d' 'all')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM -t api -p AMQSPUT -d all
```

### 使用在 **16MB** 处换行的文件启动高详细信息跟踪早期跟踪

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/STRMQTRC) PARM('-e' '-t' 'all' '-t' 'detail' '-l' '16')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM -e -t all -t detail -l 16
```

2. 在跟踪处于活动状态时重现该问题, 使每次尝试保持跟踪尽可能短。

要检查跟踪状态, 请使用 **QMQM/STRMQTRC** 程序或 **STRMQTRC** 命令的 **-s** 参数:

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/STRMQTRC) PARM('-s')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM -s
```

3. 使用 **QMQM/ENDMQTRC** 程序或 **ENDMQTRC** 命令停止跟踪。

一旦发生问题, 立即停止跟踪。如果您等待停止跟踪, 那么 IBM 支持人员需要的数据可能会丢失或被覆盖。您可以一次性停止所有跟踪, 也可以单独停止每个跟踪。例如:

#### 停止所有跟踪

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/ENDMQTRC) PARM('-a')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/ENDMQTRC.PGM -a
```

#### 停止缺省详细信息跟踪

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/ENDMQTRC)
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/ENDMQTRC.PGM
```

使用不带参数的 **endmqtrc** 与 **endmqtrc -e** 具有相同的效果。不能将 **-e** 参数与 **-m** 参数, **-i** 参数或 **-p** 参数一起指定。

#### 停止队列管理器 **QMA** 的跟踪

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/ENDMQTRC) PARM('-m' 'QMA')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/ENDMQTRC.PGM -m QMA
```

### 停止名为 AMQSPUT 的作业的跟踪

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/ENDMQTRC) PARM('-p' 'AMQSPUT')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/ENDMQTRC.PGM -p AMQSPUT
```

### 停止早期跟踪

- 使用命令行进行安装的步骤如下:

```
CALL PGM(QMQM/ENDMQTRC) PARM('-e')
```

- 使用 Qshell:

```
/QSYS.LIB/QMQM.LIB/STRMQTRC.PGM -e
```

#### 4. 可选: 格式化二进制跟踪文件。

IBM MQ 生成的跟踪文件是二进制文件, 必须先进行格式化, 然后才能读取这些文件。您可以将二进制跟踪发送到 IBM, 也可以在系统未处于高负载时在 Qshell 对其进行格式化。如果自行格式化跟踪, 请先删除二进制跟踪文件, 然后再将文件发送到 IBM:

- 使用命令行来删除二进制跟踪文件:

```
RMVLNK OBJLNK('/QIBM/UserData/mqm/trace/AMQ*?.TR?')
```

- 使用 Qshell 通过 **DSPMQTRC** 命令显示格式化的跟踪文件, 然后删除二进制跟踪文件:

```
cd /QIBM/UserData/mqm/trace
/QSYS.LIB/QMQM.LIB/DSPMQTRC.PGM AMQ*.TR?
rm -f AMQ*?.TR?
```

#### 5. 收集与问题相关的跟踪和任何其他故障诊断信息。

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 281 页的『使用 runmqras 自动收集故障诊断信息』中所述。如果您正在使用 **runmqras** 命令来收集 IBM MQ 故障诊断信息, 并且需要包含跟踪数据, 那么必须在为 **-section** 参数指定的节名称中包含 **trace**。如果跟踪非常大, 那么可能需要使用 **-workdirectory** 选项将文件打包到具有大量可用空间的目录中。

必须在 Qshell 上运行 **runmqras** 命令, 如以下示例中所示:

```
/QIBM/ProdData/mqm/bin/runmqras -section trace,defs,cluster -caseno TS001234567 -qmlist
QMA
```

- 或者, 手动收集数据, 如第 284 页的『手动收集故障诊断信息』中所述。

#### 6. 将针对相同问题收集的跟踪和任何其他故障诊断信息发送到 IBM。

有关更多信息, 请参阅第 367 页的『将故障诊断信息发送到 IBM』。

#### 7. 清除跟踪目录以回收空间, 并确保您已准备好在将来生成新的跟踪 (如果需要)。

- 使用命令行进行安装的步骤如下:

```
RMVLNK OBJLNK('/QIBM/UserData/mqm/trace/AMQ*.*')
```

- 使用 Qshell:

```
cd /QIBM/UserData/mqm/trace
rm -f AMQ*.*
```

## 相关概念

第 378 页的『FFST: IBM MQ for IBM i』

描述 First Failure Support Technology (FFST) 的名称, 位置和-content IBM i 系统的文件。

## 相关任务

第 280 页的『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。

## Windows 上的跟踪

在 Windows 上, 可以使用 **strmqtrc** 和 **endmqtrc** 命令来启动和结束跟踪。您还可以使用 IBM MQ Explorer 来启动和结束跟踪。

### 开始之前

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。

### 关于此任务

跟踪可以捕获许多 IBM MQ 问题的根本原因。为了避免影响性能或生成过大的跟踪文件并填充磁盘, 限制跟踪处于活动状态的时间很重要。

在 Windows 上, 使用以下命令来控制跟踪工具:

#### **strmqtrc**

在 Windows 系统上, 使用 **strmqtrc** 控制命令启用或修改跟踪。该命令具有可选参数, 用于指定所需的跟踪级别。例如:

- 对于选择性组件跟踪, 请使用 **-t** 和 **-x** 参数来控制要记录的跟踪详细信息量。使用 **-t** 参数指定要跟踪的点, 或使用 **-x** 参数指定不希望跟踪的点。如果需要, 您可以使用不同的参数同时运行多个跟踪。
- 对于选择性进程跟踪, 请使用 **-p** 参数将跟踪生成限制为指定的指定进程。
- 要防止 IBM MQ 在其跟踪中包含任何消息数据, 请使用 **-d 0** 参数。

有关更多信息, 请参阅 [strmqtrc \(启动跟踪\)](#)。

#### **endmqtrc**

要停止跟踪, 请使用 **endmqtrc** 控制命令。**endmqtrc** 命令仅结束由其参数描述的跟踪。使用不带参数的 **endmqtrc** 将结束对所有进程的早期跟踪。有关更多信息, 请参阅 [endmqtrc \(结束跟踪\)](#)。

在 Windows 系统上, 您也可以使用 IBM MQ Explorer 来启动和停止跟踪, 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**。

将在 `MQ_DATA_PATH/trace` 目录中创建输出文件。

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. 对于运行为受跟踪实体的一部分的每个进程, 都有一个跟踪文件。

SSL 跟踪文件的名称为 `AMQ.SSL.TRC` 和 `AMQ.SSL.TRC.1`。

IBM MQ 在 Windows 上生成的跟踪文件是可读文本文件 (AMQ.SSL.TRC 和 AMQ.SSL.TRC.1 除外, 如果存在)。在 Windows 上, 在将跟踪文件发送到 IBM 支持之前, 不需要对其进行格式化。

## 过程

1. 如果在单个 Windows 系统上具有多个 IBM MQ 安装, 请确保为要跟踪的安装设置了环境。

例如, 如果要跟踪与 Installation1 关联的队列管理器或应用程序, 那么必须先使用 **setmqenv** 命令切换到 Installation1, 然后再运行任何命令以启动, 停止或格式化跟踪, 否则跟踪将不会捕获任何有用信息。

检查 /etc/opt/mqm/mqinst.ini 文件以确定系统上 IBM MQ 安装的位置。使用 **setmqenv** 命令为要跟踪的安装设置环境。

例如:

### 为 Installation1 设置环境

```
"C:\Program Files\IBM\MQ\bin\setmqenv" -n Installation1
```

### 使用队列管理器 QMA 为安装设置环境

```
"C:\Program Files\IBM\MQ\bin\setmqenv" -m QMA
```

跟踪将写入 **MQ\_DATA\_PATH** 环境变量给出的路径下的 **trace** 子目录。运行 **setmqenv** 后, 可以通过运行以下命令来查找跟踪目录的位置:

```
echo "%MQ_DATA_PATH%\trace"
```

2. 使用 **strmqtrc** 命令启动跟踪。

缺省情况下, 启用所有跟踪点并生成缺省详细信息跟踪。传递到 **strmqtrc** 的参数可控制哪些进程生成跟踪文件以及它们跟踪的数据类型。例如:

### 仅跟踪流经通信网络的数据

```
strmqtrc -x all -t comms
```

### 启动队列管理器 QMA 的缺省详细信息跟踪

```
strmqtrc -m QMA
```

### 在不包含消息数据的情况下启动队列管理器 QMB 的缺省详细信息跟踪

```
strmqtrc -m QMB -d 0
```

### 对包含所有消息数据的名为 amqsput 的进程启动 API 跟踪

```
strmqtrc -t api -p amqsput -d all
```

### 使用以 100MB 为包装的文件启动所有内容的高详细跟踪

```
strmqtrc -e -t all -t detail -l 100
```

3. 在跟踪处于活动状态时重现该问题, 使每次尝试保持跟踪尽可能短。

要检查跟踪状态, 请使用 **strmqtrc** 命令的 **-s** 参数:

```
strmqtrc -s
```

4. 使用 **endmqtrc** 命令停止跟踪。

一旦发生问题, 立即停止跟踪。如果您等待停止跟踪, 那么 IBM 支持人员需要的数据可能会丢失或被覆盖。您可以一次性停止所有跟踪, 也可以单独停止每个跟踪。例如:

### 停止安装中的所有跟踪

```
endmqtrc -a
```

## 停止队列管理器 QMA 的跟踪

```
endmqtrc -m QMA
```

## 停止名为 amqsput 的进程的跟踪

```
endmqtrc -p amqsput
```

### 5. 收集与问题相关的跟踪和任何其他故障诊断信息。

如果您正在使用 **runmqras** 命令来收集 IBM MQ 故障诊断信息，并且需要包含跟踪数据，那么必须在为 **-section** 参数指定的节名称中包含 **trace**。如果跟踪非常大，那么可能需要使用 **-workdirectory** 选项将文件打包到具有大量可用空间的目录中。有关更多信息，请参阅第 281 页的『使用 runmqras 自动收集故障诊断信息』。

### 6. 将针对相同问题收集的跟踪和任何其他故障诊断信息发送到 IBM。

有关更多信息，请参阅第 367 页的『将故障诊断信息发送到 IBM』。

### 7. 清除跟踪目录

只要您与 IBM 支持人员一起工作，请保存跟踪文件的副本，但将其保留在跟踪目录以外的位置。清除跟踪目录以回收空间，并确保您已准备好在将来生成新的跟踪 (如果需要)。例如：

```
C:\> CD "%MQ_DATA_PATH%\trace"  
C:\ProgramData\IBM\MQ\trace> DEL AMQ*.TRC AMQ*.TRS
```

## 相关概念

第 380 页的『FFST: IBM MQ for Windows』

描述 First Failure Support Technology (FFST) 的名称，位置和 content Windows 系统的文件。

## 相关任务

第 280 页的『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。

## Windows 的跟踪数据示例

从 IBM MQ for Windows 跟踪文件中抽取。

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=0K)

```

## Tracing on z/OS

There are different trace options that can be used for problem determination with IBM MQ. Use this topic to understand the different options and how to control trace.

### About this task

The trace facilities available with IBM MQ for z/OS are:

- The channel initiator trace
- The CICS adapter trace
- The generalized trace facility (GTF) trace.

If trace data is not produced, check the following:

- Was the GTF started correctly, specifying event identifiers (EIDs) 5E9, 5EA, and 5EE on the USRP option?
- Was the **START TRACE**(GLOBAL) command entered correctly, and were the relevant classes specified?

See [“Using the GTF with IBM MQ global trace”](#) on page 406 for more information.

- The IBM internal trace used by the support center
- The line trace
- System TLS (formerly SSL) trace
- The user parameter (or API) trace

- z/OS traces

For more information, see [“Other types of trace on z/OS” on page 413](#).

See the sub topics listed in [“Collecting troubleshooting information on z/OS” on page 325](#) for more information on what you need to send to IBM support when you have a specific problem that requires resolving.

Use the following links to find out how to collect and interpret the data produced by the user parameter trace, and describes how to produce the IBM internal trace for use by the IBM support center. There is also information about the other trace facilities that you can use with IBM MQ.

### Related tasks

[“为 IBM 支持人员收集故障诊断信息” on page 280](#)

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。

## Using the GTF with IBM MQ global trace

On z/OS, you can obtain information about MQI calls and user parameters passed by some IBM MQ calls on entry to, and exit from, IBM MQ. To do this, use the IBM MQ global trace in conjunction with the z/OS generalized trace facility (GTF).

### Before you begin

To use any of the trace commands, you must have one of the following:

- Authority to issue start and stop trace commands (trace authority).
- Authority to issue the display trace command (display authority).

#### Note:

1. The trace commands can also be entered through the initialization input data sets.
2. The trace information produced will also include details of syncpoint flows - for example PREPARE and COMMIT.

### Procedure

- To start and stop the GTF
  - a) Start the GTF at the console by entering a **START GTF** command.  
When you start the GTF, specify the USRP option. You are prompted to enter a list of event identifiers (EIDs). The EIDs used by IBM MQ are:

#### **5E9**

To collect information about control blocks on entry to IBM MQ

#### **5EA**

To collect information about control blocks on exit from IBM MQ

Sometimes, if an error occurs that you cannot solve yourself, you might be asked by your IBM support center to supply other, internal, trace information for them to analyze. The additional type of trace is:

#### **5EE**

To collect information internal to IBM MQ

You can also use the JOBNAMEP option, specifying the batch, CICS, IMS, or TSO job name, to limit the trace output to specific jobs. The following example shows a sample startup for the GTF, specifying the four EIDs, and a jobname. The lines shown in **bold** are the commands that you enter



at the console; the other lines are prompts and responses. For more information about starting the GTF trace, see [Starting 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

```

where

- xx is the name of the GTF procedure to use (optional)
- yy is an identifier for this occurrence of GTF trace
- xxxx is the name of the queue manager
- zzzzzzzz is a batch job or CICS region name

Up to 5 job names can be listed.

When using GTF, specify the primary job name (CHINIT, CICS, or batch) in addition to the queue manager name (xxxxMSTR).

b) Stop the GTF at the console.

When you enter the stop command for the GTF, include the additional identifier (yy) that you used at startup, as shown in the following example:

```
STOP yy
```

• To use IBM MQ global trace

a) To start writing IBM MQ records to the GTF, use the **START TRACE** command, specifying trace type **GLOBAL**.

You must also specify **dest (GTF)** as shown in the following example:

```
/cpf start trace(GLOBAL)class(2,3)dest(GTF)
```

To define the events that you want to produce trace data for, use one or more of the following classes:

CLASS	Event traced
2	Record the MQI call and MQI parameters when a completion code other than MQRC_NONE is detected.
3	Record the MQI call and MQI parameters on entry to and exit from the queue manager.



**Attention:** Note the following:

- You can use more than one of the classes that are allowed for the type of trace started. You specify a range of classes as m:n, for example, CLASS(1:3).
  - If you do not specify a class, the default is to start class 1.
  - However, if you use the **START TRACE**(STAT) command with no class, the default is to start class 1 and 2.
- b) After the trace has started, use the following MQSC commands to display information about or alter the properties of the trace:
- Use **DISPLAY TRACE** to display a list of active traces.
  - Use **ALTER TRACE** to change the trace events being traced for a particular active queue manager trace. **ALTER TRACE** stops the specified trace, and restarts it with the altered parameters.
- c) To stop tracing, use the **STOP TRACE** command.
- For example:

```
/cpl STOP TRACE(GLOBAL) DEST(GTF) CLASS(*) RMID(*)
```

In this command, /cpl refers to the command prefix for the IBM MQ subsystem.

## **Formatting and identifying the control block information on z/OS**

After capturing a trace, the output must be formatted and the IBM MQ control blocks identified.

- [Formatting the information](#)
- [Identifying the control blocks associated with IBM MQ](#)
- [Identifying the event identifier associated with the control block](#)

### Formatting the information

To format the user parameter data that is collected by the global trace, use either the batch job that is shown in [Figure 61](#) on page 408 or the IPCS GTFTRACE USR( *xxx* ) command, where *xxx* is:

#### 5E9

To format information about control blocks on entry to IBM MQ MQI calls.

#### 5EA

To format information about control blocks on exit from IBM MQ MQI calls.

#### 5EE

To format information about IBM MQ internals.

You can also specify the **JOBNAME**(*jobname*) parameter to limit the formatted output to specific jobs.

```
//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=*
//GTFIN DD DSN=gtf.trace,DISP=SHR
//SYSTSIN DD *
IPCS
SETDEF FILE(GTFIN) NOCONFIRM
GTFTRACE USR(5E9,5EA,5EE)
/*
//STEPLIB DD DSN=thlqual.SCSQAUTH,DISP=SHR
```

Figure 61. Formatting the GTF output in batch

## Identifying the control blocks associated with IBM MQ

The format identifier for the IBM MQ trace is D9. This value appears at the beginning of each formatted control block in the formatted GTF output, in the form:

```
USRD9
```

## Identifying the event identifier associated with the control block

The trace formatter inserts one of the following messages at the start of each control block. These messages indicate whether the data was captured on entry to or exit from IBM MQ:

- CSQW072I ENTRY: MQ user parameter trace
- CSQW073I EXIT: MQ user parameter trace

### Related tasks

[“Using the GTF with IBM MQ global trace” on page 406](#)

On z/OS, you can obtain information about MQI calls and user parameters passed by some IBM MQ calls on entry to, and exit from, IBM MQ. To do this, use the IBM MQ global trace in conjunction with the z/OS generalized trace facility (GTF).

## **Interpreting the trace information on z/OS**

The GTFTRACE produced by IBM MQ can be examined to determine possible errors with invalid addresses, invalid control blocks, and invalid data.

Start the [GTFTRACE subcommand](#) to format generalized trace facility (GTF) records contained in a dump or in a trace data set. For more information on GTF, see [“Using the GTF with IBM MQ global trace” on page 406](#).

When you look at the data produced by the GTFTRACE command, consider the following points:

- If the control block consists completely of zeros, it is possible that an error occurred while copying data from the user's address space. This might be because an invalid address was passed.
- If the first part of the control block contains non-null data, but the rest consists of zeros, it is again possible that an error occurred while copying data from the user's address space, for example, the control block was not placed entirely within valid storage. This might also be due to the control block not being initialized correctly.
- If the error occurred on exit from IBM MQ, it is possible that IBM MQ might not write the data to the user's address space. The data displayed is the version that it was attempting to copy to the user's address space.

The following tables show details of the control blocks that are traced.

[Table 28 on page 409](#) illustrates which control blocks are traced for different MQI calls.

<b>MQI call</b>	<b>Entry</b>	<b>Exit</b>
MQCB	MQCBD, MQMD, MQGMO	MQCBD, MQMD, MQGMO
MQCLOSE	None	None
MQGET	MQMD, MQGMO	MQMD, MQGMO, and the first 256 bytes of message data

Table 28. Control blocks traced for IBM MQ MQI calls (continued)

MQI call	Entry	Exit
MQINQ	Selectors (if <i>SelectorCount</i> is greater than 0)	Selectors (if <i>SelectorCount</i> is greater than 0) Integer attributes (if <i>IntAttrCount</i> is greater than 0) Character attributes (if <i>CharAttrLength</i> is greater than 0)
MQOPEN	MQOD	MQOD
MQPUT	MQMD, MQPMO, and the first 256 bytes of message data	MQMD, MQPMO, and the first 256 bytes of message data
MQPUT1	MQMD, MQOD, MQPMO, and the first 256 bytes of message data	MQMD, MQOD, MQPMO, and the first 256 bytes of message data
MQSET	Selectors (if <i>SelectorCount</i> is greater than 0) Integer attributes (if <i>IntAttrCount</i> is greater than 0) Character attributes (if <i>CharAttrLength</i> is greater than 0)	Selectors (if <i>SelectorCount</i> is greater than 0) Integer attributes (if <i>IntAttrCount</i> is greater than 0) Character attributes (if <i>CharAttrLength</i> is greater than 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

**Note:** In the special case of an MQGET call with the WAIT option, a double entry is seen if there is no message available at the time of the MQGET request, but a message subsequently becomes available before the expiry of any time interval specified.

This is because, although the application has issued a single MQGET call, the adapter is performing the wait on behalf of the application and when a message becomes available it reissues the call. So in the trace it appears as a second MQGET call.

Information about specific fields of the queue request parameter list is also produced in some circumstances. The fields in this list are identified as follows:

Identifier	Description
Action	Requested action
BufferL	Buffer length
CBD	Address of callback descriptor
CompCode	Completion code
CharAttL	Character attributes length
DataL	Data length
Hobj	Object handle
Hsub	Subscription handle

Identifier	Description
IntAttC	Count of integer attributes
pObjDesc	Object descriptor
Oper	Operation
Options	Options
pBuffer	Address of buffer
pCharAtt	Address of character attributes
pCTLO	Address of control callback options
pECB	Address of ECB used in get
pGMO	Address of get message options
pIntAtt	Address of integer attributes
pMsgDesc	Address of message descriptor
pPMO	Address of put message options
pSD	Address of subscription descriptor
pSelect	Address of selectors
pSRQOpt	Address of subscription request options
pSTS	Address of status structure
Reason	Reason code
RSVn	Reserved for IBM
SelectC	Selector count
Thread	Thread
Type	Requested type
UOWInfo	Information about the unit of work
Userid	CICS or IMS user ID, for batch or TSO this is zero

## Using the channel initiator trace

The channel initiator trace can help to determine why a problem is happening in the CHIN address space, whether the cause of the problem is in IBM MQ, the operating system, or the network.

### About this task

See [Figure 46 on page 249](#) for information about how to get a dump of the channel initiator address space. Note that dumps produced by the channel initiator do not include trace data space. The trace data space, which is called CSQXTRDS, contains trace information. You can request this by specifying it on a slip trap or when you use the dump command.

### Procedure

1. To run the trace, use the [START TRACE](#) command.  
You can also set this trace to start automatically using the TRAXSTR queue manager attribute. For more information about how to do this, see [ALTER QMGR](#).
2. To display this trace information, enter the IPCS command:

```
LIST 1000. DSPNAME(CSQXTRDS)
```

3. To format the trace, use the command:

```
CTRACE COMP(CSQX $ssnm$ )
```

where  $ssnm$  is the subsystem name.

### Related tasks

“为 IBM 支持人员收集故障诊断信息” on page 280

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。

## Using line trace

A wrap-around line trace exists for each channel. This trace is kept in a 4 KB buffer for each channel in the channel initiator address space. Trace is produced for each channel, so it is ideal for problems where a channel appears to be hung, because information can be collected about the activity of this channel long after the normal trace has wrapped.

### About this task

The line trace is always active; you cannot turn it off. It is available for both LU 6.2 and TCP channels and should reduce the number of times a communications trace is required.

You can view the trace as unformatted trace that is written to CSQSNAP.

You can display the trace by following these steps:

### Procedure

1. Ensure that the CHIN procedure has a SNAP DD statement.
2. Start a CHIN trace, specifying IFCID 202 as follows:

```
START TRACE(CHINIT) CLASS(4) IFCID(202)
```

3. Display the channel status for those channels for which the line trace is required:

```
DISPLAY CHSTATUS(channel) SAVED
```

This dumps the current line for the selected channels to CSQSNAP. For more information, see [“Snap dumps on z/OS” on page 264](#).

The line trace buffer is unformatted. Each entry starts with a clock, followed by a time stamp, and an indication of whether this is an OUTBOUND or INBOUND flow. Use the time stamp information to find the earliest entry.

#### Notes:

- a. The addresses of the storage dump are incorrect because the CSQXFFST mechanism takes a copy of the storage before writing it to CSQSNAP.
  - b. The dump to CSQSNAP is only produced the first time you run the **DISPLAY CHSTATUS SAVED** command. This is to prevent getting dumps each time you run the command.
4. To obtain another dump of line trace data, stop and restart the current trace.

- a. You can use a selective **STOP TRACE** command to stop just the trace that was started to gather the line trace data. To do this, note the TRACE NUMBER assigned to the trace as shown in this example:

```
+ssid START TRACE(CHINIT) CLASS(4) IFCID(202)
      CSQW130I +ssid 'CHINIT' TRACE STARTED, ASSIGNED TRACE NUMBER 01
```

- b. To stop the trace, issue the following command:

```
+ssid STOP TRACE(CHINIT) TNO(01)
```

- c. You can then enter another **START TRACE** command with a **DISPLAY CHSTATUS SAVED** command to gather more line trace data to CSQSNAP.

### Related tasks


“为 IBM 支持人员收集故障诊断信息” on page 280

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。

## Other types of trace on z/OS

There are other trace facilities available for problem determination. Use this topic to investigate CICS adapter trace, SSL trace, and z/OS trace.

It can be helpful to use the following trace facilities with IBM MQ.

- [The CICS adapter trace](#)
- [System SSL trace](#)
-  [z/OS traces](#)

## The CICS adapter trace

The CICS adapter writes entries to the CICS trace if your trace number is set to a value in the range 0 through 199 (decimal), and if either:

- CICS user tracing is enabled, or
- CICS internal/auxiliary trace is enabled

You can enable CICS tracing in one of two ways:

- Dynamically, using the CICS-supplied transaction [CETR](#)
- By ensuring that the USERTR parameter in the CICS system initialization table (SIT) is set to YES

The CICS trace entry originating from the CICS adapter has a value AP0 000, where 000 is the hexadecimal equivalent of the decimal value of the CICS adapter trace number you specified.

The trace entries are shown in [“CICS adapter trace entries”](#) on page 414.

## System SSL trace

You can collect System SSL trace using the SSL Started Task. The details of how to set up this task are in the *System Secure Sockets Layer Programming* documentation, SC24-5901. A trace file is generated for each SSLTASK running in the CHINIT address space.

## z/OS traces





z/OS traces, which are common to all products operating as formal subsystems of z/OS, are available for use with IBM MQ. For information about using and interpreting this trace facility, see the [z/OS MVS Diagnosis: Tools and Service Aids](#) manual.

## **CICS adapter trace entries**

Use this topic as a reference for CICS adapter trace entries.

The CICS trace entry for these values is APO xxx (where xxx is the hexadecimal equivalent of the trace number you specified when the CICS adapter was enabled). These trace entries are all issued by CSQCTRUE, except CSQCTEST, which is issued by CSQCRST and CSQCDSP.

Name	Description	Trace sequence	Trace data
CSQCABNT	Abnormal termination	Before issuing END_THREAD ABNORMAL to IBM MQ. This is due to the end of the task and therefore an implicit backout could be performed by the application. A ROLLBACK request is included in the END_THREAD call in this case.	Unit of work information. You can use this information when finding out about the status of work. (For example, it can be verified against the output produced by the DISPLAY THREAD command, or the log print utility.)
CSQCAUID	Bridge security	Before validating bridge user password or PassTicket.	User ID.
CSQCBACK	Syncpoint backout	Before issuing BACKOUT to IBM MQ. This is due to an explicit backout request from the application.	Unit of work information.
CSQCCONX	MQCONN	Before issuing MQCONN to IBM MQ.	Connection tag.
CSQCCRC	Completion code and reason code	After unsuccessful return from API call.	Completion code and reason code.
CSQCCOMM	Syncpoint commit	Before issuing COMMIT to IBM MQ. This can be due to a single-phase commit request or the second phase of a two-phase commit request. The request is due to an explicit syncpoint request from the application.	Unit of work information.
CSQDCFF	IBM use only		
CSQDCIN	IBM use only		
CSQDCOT	IBM use only		
CSQCEXER	Execute resolve	Before issuing EXECUTE_RESOLVE to IBM MQ.	The unit of work information of the unit of work issuing the EXECUTE_RESOLVE. This is the last in-doubt unit of work in the resynchronization process.
CSQCGETW	GET wait	Before issuing CICS wait.	Address of the ECB to be waited on.
CSQCGMGD	GET message data	After successful return from MQGET.	Up to 40 bytes of the message data.
CSQCGMGH	GET message handle	Before issuing MQGET to IBM MQ.	Object handle.

Table 29. CICS adapter trace entries (continued)

Name	Description	Trace sequence	Trace data
CSQCGMGI	Get message ID	After successful return from MQGET.	Message ID and correlation ID of the message.
CSQCHCER	Hconn error	Before issuing any MQ verb.	Connection handle.
CSQCINDL	In-doubt list	After successful return from the second INQUIRE_INDOUBT.	The in-doubt units of work list.
CSQCINDO	IBM use only		
CSQCINDS	In-doubt list size	After successful return from the first INQUIRE_INDOUBT and the in-doubt list is not empty.	Length of the list; divided by 64 gives the number of in-doubt units of work.
CSQCINDW	Syncpoint in doubt	During syncpoint processing, CICS is in doubt as to the disposition of the unit of work.	Unit of work information.
CSQCINQH	INQ handle	Before issuing MQINQ to IBM MQ.	Object handle.
CSQCLOSH	CLOSE handle	Before issuing MQCLOSE to IBM MQ.	Object handle.
CSQCLOST	Disposition lost	During the resynchronization process, CICS informs the adapter that it has been cold started so no disposition information regarding the unit of work being resynchronized is available.	Unit of work ID known to CICS for the unit of work being resynchronized.
CSQCNIND	Disposition not in doubt	During the resynchronization process, CICS informs the adapter that the unit of work being resynchronized should not have been in doubt (that is, perhaps it is still running).	Unit of work ID known to CICS for the unit of work being resynchronized.
CSQCNORT	Normal termination	Before issuing END_THREAD NORMAL to IBM MQ. This is due to the end of the task and therefore an implicit syncpoint commit might be performed by the application. A COMMIT request is included in the END_THREAD call in this case.	Unit of work information.
CSQCOPNH	OPEN handle	After successful return from MQOPEN.	Object handle.
CSQCOPNO	OPEN object	Before issuing MQOPEN to IBM MQ.	Object name.
CSQCPMGD	PUT message data	Before issuing MQPUT to IBM MQ.	Up to 40 bytes of the message data.
CSQCPMGH	PUT message handle	Before issuing MQPUT to IBM MQ.	Object handle.
CSQCPMGI	PUT message ID	After successful MQPUT from IBM MQ.	Message ID and correlation ID of the message.

Table 29. CICS adapter trace entries (continued)

Name	Description	Trace sequence	Trace data
CSQCPREP	Syncpoint prepare	Before issuing PREPARE to IBM MQ in the first phase of two-phase commit processing. This call can also be issued from the distributed queuing component as an API call.	Unit of work information.
CSQCP1MD	PUTONE message data	Before issuing MQPUT1 to IBM MQ.	Up to 40 bytes of data of the message.
CSQCP1MI	PUTONE message ID	After successful return from MQPUT1.	Message ID and correlation ID of the message.
CSQCP1ON	PUTONE object name	Before issuing MQPUT1 to IBM MQ.	Object name.
CSQCRBAK	Resolved backout	Before issuing RESOLVE_ROLLBACK to IBM MQ.	Unit of work information.
CSQRCMT	Resolved commit	Before issuing RESOLVE_COMMIT to IBM MQ.	Unit of work information.
CSQCRMIR	RMI response	Before returning to the CICS RMI (resource manager interface) from a specific invocation.	Architected RMI response value. Its meaning depends of the type of the invocation. To determine the type of invocation, look at previous trace entries produced by the CICS RMI component.
CSQCRSYN	Resync	Before the resynchronization process starts for the task.	Unit of work ID known to CICS for the unit of work being resynchronized.
CSQCSETH	SET handle	Before issuing MQSET to IBM MQ.	Object handle.
CSQCTASE	IBM use only		
CSQCTEST	Trace test	Used in EXEC CICS ENTER TRACE call to verify the trace number supplied by the user or the trace status of the connection.	No data.

## Enabling internal trace for the AMSM address space

Trace for the AMSM address space can be enabled using the `_AMS_MSG_LEVEL` variable, which is passed into the AMSM address space through the ENVARS DD card.

A sample data set for the ENVARS DD card is in `th1qua1.SCSQPROC(CSQ40ENV)`.

Trace is written to the SYSOUT of the AMSM address space.

The `_AMS_MSG_LEVEL` variable specifies the subcomponent and message level that is to be logged. An asterisk indicates all subcomponents to be logged; currently there is only one subcomponent.

The severity levels are:

- S - severe messages only
- E - error and severe messages only
- W - warning, error, and severe messages only
- I - informational, warning, error, and severe messages. This is the default value

- D - debug mode, all messages with additional debug diagnostics
- V - verbose mode, all of the preceding, plus buffer dumps



**Attention:** You should only enable debug or verbose mode on the advice of an IBM service representative.

For example, to enable the default for `_AMS_MSG_LEVEL`, issue the following:

```
_AMS_MSG_LEVEL=* .I
```

To enable verbose mode, issue the following:

```
_AMS_MSG_LEVEL=* .V
```

You can also start the internal trace for the AMS address space using the z/OS MODIFY (F) command.

For example, to start debug level trace, issue the following command:

```
/F xxxxAMSM LOG=* .D.
```

To stop debug level trace, and revert to the default information level trace, issue the following command:

```
/F xxxxAMSM LOG=* .I.
```

### Related tasks

[“对 AMS 问题进行故障诊断” on page 61](#)

用于帮助您识别和解决与 Advanced Message Security (AMS) 相关的问题的故障诊断信息。

## Using GSKit trace for problems related to certificates and keys when using AMS on z/OS

Use this topic to understand how to turn on and turn off IBM Global Security Kit (GSKit) tracing when using AMS on z/OS.

### Introduction

In the JCL for the AMS (ssidAMSM) address space, and in sample JCL `h1q.SCSQPROC (CSQ40CFG)` that runs program `CSQOUTIL`, there is an ENVARS DD card that can be used to set environment variables. A sample AMS environment variables file called `h1q.SCSQPROC (CSQ40ENV)` is provided which includes details of how to turn on and turn off GSKit trace.

If you set GSK trace environment variables in the ENVARS DD card in the JCL for the AMS address space, variables are set from the point that the AMS address space is started (that is, as part of queue manager start-up if AMS has been configured). Variables either turn on, or turn off, tracing of all `gsk_*` calls issued by the AMS address space.

If you set GSK trace environment variables in the ENVARS DD card in sample JCL `h1q.SCSQPROC (CSQ40CFG)`, variables are set for the duration of the `CSQ40CFG` job. Variables either turn on, or turn off, tracing of all `gsk_*` calls issued during the processing of AMS commands that, for example, define and display AMS policies.

### Turning on GSKit trace

#### Turn on GSKit trace for the AMS address space

To turn on GSKit trace for the AMS address space, carry out the following procedure:

1. Create JCL job `h1q.SCSQPROC (CSQ40ENV)` with:

```
GSK_TRACE_FILE=/u/<username>/AMStrace/gsktrace/gskssl.%.trc
GSK_TRACE=0xff
```

2. Specify member hlq.SCSQPROC (CSQ40ENV) on the ENVARS DD card in the ssidAMSM JCL for the AMS address space. For example:

```
//ENVARS DD DSN=hlq.SCSQPROC(CSQ40ENV),DISP=SHR
```

3. Start the queue manager (ssidMSTR) and channel initiator (ssidCHIN) address spaces. Remember that the AMS (ssidAMSM) address space is started as part of starting the queue manager.

You can see the environment variable settings in the job log for the AMS address space. For example:

```
-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
```

The `gsk_*` calls issued by the AMS address space to protect or unprotect IBM MQ messages at put and get time respectively, are traced. A trace file is created when the AMS address space is started, to trace all `gsk_*` calls subsequently performed by the address space. The use of the `%` character in the name of the trace file ensures that trace files are named by Unix Systems Services (USS) process identifiers.

4. Issue the following command to list the trace files produced:

```
/u/<username>/AMStrace/gsktrace:>ls
```

For example, you see files like:

```
gskssl.84017302.trc
```

5. To format and view the trace file, issue the following command in USS:

```
/u/<username>/AMStrace/gsktrace:>gsktrace gskssl.84017302.trc
```

which produces output similar to the following:

```
07/01/2022-10:36:41 Thd-0 INFO gsk_svc_init(): System SSL Version 4, Release 4, Service level 0A60573
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 IS08859-1 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,0=IBM,C=UK
07/01/2022-10:36:46 Thd-0 EXIT gsk_dn_to_name(): <--- Exit status 0x00000000 (0)

```

## Turn on GSKit trace for JCL h1q.SCSQPROC (CSQ40CFG)

To turn on GSKit trace for JCL CSQ40CFG, carry out the following procedure:

1. Create JCL job h1q.SCSQPROC (CSQ40CFG) with:

```

GSK_TRACE_FILE=/u/<username>/AMStrace/gsktrace/gskssl.%.trc
GSK_TRACE=0xff

```

2. Set the ENVARS DD card as in the following example. On the **setmqsp1** and the **dspmqsp1** commands, specify parameters specific to your environment:

```

//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=h1q.SCSQANLE, DISP=SHR
//          DD DSN=h1q.SCSQAUTH, DISP=SHR
//ENVARS DD DSN=h1q.SCSQPROC(CSQ40ENV), DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
setmqsp1 -m MQ01 -p BANK.RQ
          -r CN=USERID,0=IBM,C=UK -e AES256
dspmqsp1 -m MQ01 -p BANK.RQ
/*

```

3. Submit the CSQ40CFG job.

The CSQ40CFG job does not give any indication of whether GSKit trace has been enabled or not. However, you can check if trace is enabled or not by looking at the settings in the environment variables file specified for the job, or by checking if a trace file was created for the process under which the CSQ40CFG job ran.

## Turning off GSKit trace

### Turn off GSKit trace for the AMS address space

To turn off GSKit trace for the AMS address space, carry out the following procedure:

1. Stop the queue manager (ssidMSTR) address space. This stops both the channel initiator and the AMS address spaces.
2. Modify the h1q.SCSQPROC (CSQ40ENV) as follows:

```

GSK_TRACE_FILE=/u/<username>/AMStrace/gsktrace/gskssl.%.trc
GSK_TRACE=0x00

```

3. Restart the queue manager (ssidMSTR) and channel initiator (ssidCHIN) address spaces. Remember that the AMS (ssidAMSM) address space is started as part of starting the queue manager.

4. Check the environment variable settings in the job log for the AMS address space to ensure that GSKit trace has been turned off.

### Turn off GSKit for JCL h1q .SCSQPROC (CSQ40CFG)

To turn off GSKit trace for JCL h1q .SCSQPROC (CSQ40CFG), carry out the following procedure:

1. Modify the h1q .SCSQPROC (CSQ40CFG) file as follows:

```
GSK_TRACE_FILE=/u/<username>/AMSttrace/gsktrace/gskssl.%.trc  
GSK_TRACE=0x00
```

2. Submit the h1q .SCSQPROC (CSQ40CFG) job and check that no trace file is produced.

#### Notes:

- In the environment files, coding GSK\_TRACE=0xff turns trace on, and coding GSK\_TRACE=0x00 turns trace off.
- Include the % character in the trace file name to ensure that trace file names produced for different USS processes, that issue gsk\_\* calls, include the process identifier, and hence are kept separate.

#### Related information

[Generating IBM MQ GTF trace on IBM z/OS](#)

### Examples of trace output for z/OS

Use these examples to help interpret trace output on z/OS.

Figure 62 on page 421 shows an example of a trace taken on entry to an MQPUT1 call. The following items have been produced:

- Queue request parameter list
- Object descriptor (MQOD)
- Message descriptor (MQMD)
- Put message options (MQPMO)
- The first 256 bytes of message data

Compare this to [Figure 63 on page 422](#), which illustrates the same control blocks on exit from IBM MQ.



```

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

```

Figure 62. Example trace data from an entry trace of an MQPUT1 request

```

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

```

Figure 63. Example trace data from an exit trace of an MQPUT1 request

## 跟踪高级消息排队协议 (AMQP) 服务

高级消息排队协议 (AMQP) 服务提供的跟踪工具用于帮助 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）。

只能通过指定带有 **amqp** 选项的 **-t** 参数来将跟踪输出限制为 AMQP 服务。

使用不带 **-t** 参数的 **strmqtrc** 命令和 **amqp** 选项来启用跟踪会生成运行 IBM MQ AMQP 服务的整个队列管理器的跟踪信息。此信息包括 IBM MQ AMQP 服务本身以及服务用于与其他队列管理器组件进行通信的底层 Java 消息队列接口 (JMQUI)。

- 运行 **controlAMQPChannel** 命令。通过使用 **controlAMQPChannel** 命令仅跟踪 IBM MQ AMQP 服务来开启跟踪。

如果您不确定要使用的选项，请联系 IBM 支持人员，以获取有关针对您所看到的问题收集跟踪的最佳方法的建议。

## 过程

### 1. 方法 1

- a) 打开命令提示符并浏览至目录:  
`MQ_INSTALLATION_PATH\bin`
- b) 运行 **strmqtrc** 命令来启用跟踪:  
要将跟踪输出限制为仅 AMQP 服务, 请运行以下命令:

```
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。

运行 **strmqtrc** 命令而不指定 `-t` 参数和 `amqp` 选项, 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。

- c) 重现该问题。
- d) 通过运行 **endmqtrc** 命令, 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** **AIX**

```
./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** **AIX**

```
./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** **AIX** `/var/mqm/trace.`

跟踪文件的名称如下所示:

- 包含 AMQP 服务信息的跟踪文件称为 `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称为 `amqpControlMQXRChannel_PPPPP.N.trc`，其中 PPPPP 是 AMQP 服务的进程标识，N 是数字。

JMQI 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。

### 相关任务

第 280 页的『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。

第 57 页的『对 AMQP 问题进行故障诊断』

用于帮助您解决运行 AMQP 应用程序的问题的故障诊断信息。

## 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。

## 跟踪 IBM MQ Console


IBM MQ Console 中提供的跟踪功能旨在帮助 IBM 员工诊断客户问题。可通过各种属性来控制这些功能的行为。


### 开始之前

为 IBM 支持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 包括：

- JavaScript，在浏览器中运行。
- 使用 Java 编写的后端代码，在 mqweb 服务器中运行。

您可以同时跟踪在浏览器中运行的 JavaScript 以及后端代码和 JavaScript。

只有在已启用浏览器跟踪的浏览器中才会输出浏览器跟踪。从 IBM MQ Console 注销后，会自动禁用跟踪。

## 过程

- 要跟踪在 mqweb 服务器中运行的 IBM MQ Console 后端代码：

a) 

在 z/OS 上，确保设置了 WLP\_USER\_DIR 环境变量。如果需要，可以通过输入以下命令来设置变量：

```
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`

b) 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"
```

c) 要在 IBM MQ Web 服务器中启动跟踪：

- a. 启动 mqweb 服务器。
- b. 启动浏览器，并登录到 IBM MQ Console。
- c. 修改用于访问 IBM MQ Console 的 URI，使其包含

```
?trace=true
```

以下示例中所示的字符串：

原始 URI: `https://localhost:9443/ibmmq/console/#/`

新 URI: `https://localhost:9443/ibmmq/console/?trace=true#/`

d) 要关闭跟踪，请修改 URI 并除去以下字符串：

```
?trace=true
```

- 要跟踪在浏览器中运行的 IBM MQ Console JavaScript：

a) 

在 z/OS 上，确保设置了 WLP\_USER\_DIR 环境变量。如果需要，可以通过输入以下命令来设置变量：

```
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`

b) 在命令行上输入以下命令：

```
setmqweb properties -k traceSpec -v "*=info:js.mq*=all"
```

c) 要在 IBM MQ Web 服务器中启动跟踪：

- a. 启动 mqweb 服务器。
- b. 启动浏览器，并登录到 IBM MQ Console。

- c. 修改用于访问 IBM MQ Console 的 URI, 使其包含

```
?trace=true
```

以下示例中所示的字符串:

原始 URI: `https://localhost:9443/ibmmq/console/#/`

新 URI: `https://localhost:9443/ibmmq/console/?trace=true#/`

- d) 要关闭跟踪, 请修改 URI 并除去以下字符串:

```
?trace=true
```

- 要跟踪在浏览器中运行的 IBM MQ Console 后端代码和 JavaScript:

- a) 

在 z/OS 上, 确保设置了 `WLP_USER_DIR` 环境变量。如果需要, 可以通过输入以下命令来设置变量:

```
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`

- b) 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"
```

- c) 要在 IBM MQ Web 服务器中启动跟踪:

- a. 启动 `mqweb` 服务器。
- b. 启动浏览器, 并登录到 IBM MQ Console。
- c. 修改用于访问 IBM MQ Console 的 URI, 使其包含

```
?trace=true
```

以下示例中所示的字符串:

原始 URI: `https://localhost:9443/ibmmq/console/#/`



新 URI: `https://localhost:9443/ibmmq/console/?trace=true#/`

- d) 要关闭跟踪, 请修改 URI 并除去以下字符串:

```
?trace=true
```

## 结果

跟踪将输出到一组文件中。在其中创建跟踪文件的目录为:

-  `MQ_DATA_PATH/web/installations/installationName/servers/mqweb/logs`
-  子目录 `/servers/mqweb/logs` 位于在运行 `crtmqweb` 脚本来创建 `mqweb` 服务器定义时指定的目录下面。缺省情况下, 此目录为 `/var/mqm/web/installation1`。

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](#)。

## 示例

浏览器中运行的 IBM MQ Console JavaScript 代码所生成的跟踪信息的示例在以下文本中显示:

```
[18/08/20 14:34:13:880 BST] 00000050 trace-patcher 1 js.mq.mq.lib.service.trace-patcher patchFn  
DATA [18/08/20 14:34:10:472 BST] User:[mqadmin] (@static) "Patching -> i.componentDidMount"
```

```
[18/08/20 14:34:13:880 BST] 00000050 trace-patcher 1 js.mq.mq.lib.service.trace-patcher patchFn
DATA [18/08/20 14:34:10:472 BST] User:[mqadmin] (@static) "Patching -> i.componentWillUnmount"
[18/08/20 14:34:13:880 BST] 00000050 trace-patcher 1 js.mq.mq.lib.service.trace-patcher patchFn
DATA [18/08/20 14:34:10:472 BST] User:[mqadmin] (@static) "Patching -> i._getTopicsData"
[18/08/20 14:34:13:880 BST] 00000050 trace-patcher 1 js.mq.mq.lib.service.trace-patcher patchFn
DATA [18/08/20 14:34:10:472 BST] User:[mqadmin] (@static) "Patching -> i._showCreateTopic"
[18/08/20 14:34:13:880 BST] 00000050 trace-patcher 1 js.mq.mq.lib.service.trace-patcher patchFn
DATA [18/08/20 14:34:10:472 BST] User:[mqadmin] (@static) "Patching -> i._showDeleteTopic"
[18/08/20 14:34:13:881 BST] 00000050 trace-patcher 1 js.mq.mq.lib.service.trace-patcher patchFn
DATA [18/08/20 14:34:10:472 BST] User:[mqadmin] (@static) "Patching -> i._showTopicDetails"
[18/08/20 14:34:13:881 BST] 00000050 trace-patcher 1 js.mq.mq.lib.service.trace-patcher patchFn
DATA [18/08/20 14:34:10:472 BST] User:[mqadmin] (@static) "Patching -> i._showCreateSubscription"
[18/08/20 14:34:13:881 BST] 00000050 trace-patcher 1 js.mq.mq.lib.service.trace-patcher patchFn
DATA [18/08/20 14:34:10:472 BST] User:[mqadmin] (@static) "Patching -> i.render"
```

## 相关概念

[使用 IBM MQ Console 进行管理](#)

## 相关任务

[第 280 页的『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。

## 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。

### 关于此任务

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.v
```

其中后缀 `v` 是一个整数，表示跟踪文件在循环文件集中的位置。当当前跟踪文件的大小达到最大跟踪文件大小时，会将其关闭并重命名，并打开新的跟踪文件。当前跟踪文件具有后缀 `0`。较旧的跟踪文件具有连续较大的整数后缀。

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
```

## 过程

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。有关更多信息，请参阅 [跟踪](#)。

可以使用 **TraceUserData** 属性来配置由跟踪的路由接收和发送的网络传输中的数据量。有关更多信息，请参阅 [TraceUser 数据](#)。

**V 9.4.0** 从 IBM MQ 9.4.0 开始，可以使用 `mqipt.conf` 文件的 `[global]` 部分中的 **TraceFileCount** 和 **TraceFileSize** 属性来配置跟踪文件的最大大小以及保留的最大跟踪文件数。有关更多信息，请参阅 [TraceFileCount](#) 和 [TraceFileSize](#)。

## 相关任务

[第 81 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。

[第 319 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。

## 跟踪 mqiptAdmin 中的错误

`mqiptAdmin` 命令具有一个执行跟踪工具，可诊断从命令行向 MQIPT 发出管理命令时发生的错误。

### 过程

1. 要对 `mqiptAdmin` 命令启用跟踪，请先设置以下环境变量，然后再运行该命令：

- **Linux** / **AIX** 在 AIX and Linux 系统上，发出以下命令：

```
export MQIPT_JVM_OPTIONS="-Dcom.ibm.mq.ipt.trace.override.path=trace_directory  
-Dcom.ibm.mq.ipt.trace.override.level=5"
```

- **Windows** 在 Windows 系统上，发出以下命令：

```
set MQIPT_JVM_OPTIONS=-Dcom.ibm.mq.ipt.trace.override.path=trace_directory  
-Dcom.ibm.mq.ipt.trace.override.level=5
```

将 `trace_directory` 替换为跟踪文件写入到的目录的名称。

当 `mqiptAdmin` 命令运行时，将在指定的目录中创建跟踪文件。此跟踪文件的名称采用以下格式：

```
AMQyyyyymmddnnnnnnnnn.0.TRC.0
```

2. 要对 `mqiptAdmin` 禁用跟踪，请取消设置环境变量。

## 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(IPTTrace 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

有几种不同的方法可以对 IBM MQ .NET 应用程序启用跟踪。通常只需要在 IBM 支持人员的请求下使用跟踪工具。

## 关于此任务

如果您正在使用 IBM MQ classes for .NET Framework, 那么可以通过以下方式开启跟踪并配置跟踪设置:

- 对于 IBM MQ 的完整安装, 请使用 **strmqtrc** 命令对 IBM MQ classes for .NET Framework 启用跟踪。有关更多信息, 请参阅第 432 页的『使用命令跟踪 IBM MQ classes for .NET Framework 应用程序』。
- 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**。有关更多信息, 请参阅第 431 页的『使用应用程序配置文件跟踪 IBM MQ classes for .NET Framework 客户机』。
- 通过使用 **MQDOTNET\_TRACE\_ON** 环境变量。有关更多信息, 请参阅第 429 页的『使用环境变量跟踪 IBM MQ .NET 应用程序』。
- **V9.4.0** 通过使用 **mqclient.ini** 文件并设置 Trace 节的相应属性。您还可以使用 **mqclient.ini** 文件动态启用和禁用跟踪。有关更多信息, 请参阅第 430 页的『使用 **mqclient.ini** 跟踪 IBM MQ .NET 应用程序』。

.NET (.NET Standard 和 .NET 6 库)不支持使用应用程序配置文件。如果您正在使用 IBM MQ classes for .NET Standard (.NET Standard 和 .NET 6 库), 那么可以通过以下方式配置跟踪:

- 通过使用 **MQDOTNET\_TRACE\_ON** 环境变量。有关更多信息, 请参阅第 429 页的『使用环境变量跟踪 IBM MQ .NET 应用程序』。
- **V9.4.0** 从 IBM MQ 9.3.3 开始, 通过使用 **mqclient.ini** 文件并设置 Trace 节的相应属性。您还可以使用 **mqclient.ini** 文件动态启用和禁用跟踪。有关更多信息, 请参阅第 430 页的『使用 **mqclient.ini** 跟踪 IBM MQ .NET 应用程序』。

注:

对于针对 .NET Framework 构建的 IBM MQ .NET 客户机库, 优先顺序如下所示:

1. App.Config
2. 环境变量
3. **V9.4.0** **mqclient.ini**

对于针对 .NET Standard 和 .NET 6 构建的 IBM MQ .NET 客户机库, 优先顺序如下所示:

1. 环境变量
2. **V9.4.0** **mqclient.ini**

## 相关任务

第 279 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 402 页的『Windows 上的跟踪』

在 Windows 上, 可以使用 **strmqtrc** 和 **endmqtrc** 命令来启动和结束跟踪。您还可以使用 IBM MQ Explorer 来启动和结束跟踪。

第 388 页的『AIX and Linux 上的跟踪』

在 AIX and Linux 上, 可以使用 **strmqtrc** 和 **endmqtrc** 命令来启动和结束跟踪, 并使用 **dspmqrtrc** 来显示跟踪文件。在 AIX 上, 除了使用 **strmqtrc** 和 **endmqtrc** 命令外, 还可以使用 AIX 系统跟踪。

第 465 页的『跟踪 XMS .NET 应用程序』

如果您正在使用 IBM MQ classes for XMS .NET Framework, 那么可以从应用程序配置文件以及 XMS 环境变量配置跟踪。如果您正在使用 IBM MQ classes for XMS .NET (.NET Standard 和 .NET 6 库), 那么必须从 XMS 环境变量配置跟踪。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。

## 使用环境变量跟踪 IBM MQ .NET 应用程序

环境变量 **MQDOTNET\_TRACE\_ON** 用于对可重新分发的客户机启用跟踪。

## 关于此任务

**MQDOTNET\_TRACE\_ON** 环境变量检查 IBM MQ 跟踪目录是否可用。如果跟踪目录可用，那么将在跟踪目录中生成跟踪文件。但是，如果未安装 IBM MQ，那么会将跟踪文件复制到当前工作目录。

用于 IBM MQ classes for .NET Framework 的其他环境变量 (包括 **MQERRORPATH**、**MQLOGLEVEL** 和 **MQSERVER** 等) 可用于 IBM MQ classes for .NET Standard 和 .NET 6，并以相同方式工作。

## 过程

- 要启用跟踪，请设置环境变量以指定所需的跟踪级别。  
等于或小于 0 的值不会启用跟踪，1 会启用缺省跟踪，而大于 1 的值会启用详细信息跟踪。

### 相关概念

[安装 IBM MQ classes for .NET](#)

### 相关参考

[环境变量描述](#)

## V 9.4.0 使用 mqclient.ini 跟踪 IBM MQ .NET 应用程序

从 IBM MQ 9.4.0 开始，可以使用 `mqclient.ini` 文件来启用 IBM MQ .NET 客户机库的跟踪。

## 关于此任务

从 IBM MQ 9.4.0 开始，`mqclient.ini` 文件包含跟踪节。要启用跟踪，必须设置 `Trace` 节的相应属性。如果属性设置为无效值，那么将忽略这些属性。

您还可以动态启用和禁用跟踪。当应用程序正在运行时，如果修改、创建或删除了 `mqclient.ini` 文件，那么 IBM MQ .NET 客户机将再次读取跟踪部分的属性，然后启用或禁用跟踪，以便不需要重新启动应用程序。

注: 有几种不同的方法来启用跟踪。

对于针对 .NET Framework 构建的 IBM MQ .NET 客户机库，优先顺序如下所示:

1. App.Config
2. 环境变量
3. `mqclient.ini`

对于针对 .NET Standard 和 .NET 6 构建的 IBM MQ .NET 客户机库，优先顺序如下所示:

1. 环境变量
2. `mqclient.ini`

## 过程

- 要启动跟踪，请指定 `Trace` 节的以下属性:
  - a) 将 **MQDotnetTraceLevel** 设置为与所需跟踪级别对应的值。  
如果要使用完整详细信息启动跟踪 (建议)，请将此属性设置为 2。如果需要不太详细的跟踪，请将该属性设置为 1。
  - b) 将 **MQDotnetTracePath** 设置为要在其中创建跟踪文件的文件夹的文件路径。  
如果路径留空或未定义 **MQDotnetTracePath** 属性，那么将使用应用程序的当前目录。
  - c) 将 **MQDotnetErrorPath** 设置为要在其中创建错误日志文件的文件夹的文件路径。  
如果路径留空或未定义 **MQDotnetErrorPath** 属性，那么将使用应用程序的当前目录。
- 要停止跟踪，请将 **MQDotnetTraceLevel** 设置为 0。  
这是此属性的缺省值。

## 相关概念

客户机配置文件的跟踪节

## 相关任务

第 432 页的『使用命令跟踪 IBM MQ classes for .NET Framework 应用程序』

对于 IBM MQ 的完整客户机安装，您可以使用 **strmqtrc** 命令对 IBM MQ classes for .NET Framework 启用跟踪。

## 使用应用程序配置文件跟踪 IBM MQ classes for .NET Framework 客户机

如果您正在使用 IBM MQ classes for .NET Framework，那么可以使用应用程序配置文件为 IBM MQ .NET 可再分发受管客户机和独立 IBM MQ .NET 客户机生成跟踪。

## 关于此任务

生成跟踪的这些步骤适用于 IBM MQ .NET 可再分发受管客户机以及独立 IBM MQ .NET 客户机，并且仅适用于 IBM MQ classes for .NET Framework。

您必须使用应用程序配置文件和特定于 IBM MQ 的跟踪配置文件，因为既没有完整的 IBM MQ 客户机安装，也未提供用于启动和停止跟踪的标准工具 **strmqtrc** 和 **endmqtrc**。

应用程序配置文件在 IBM MQ classes for .NET (.NET Standard 和 .NET 6 库) 中不受支持。要对 IBM MQ classes for .NET (.NET Standard 和 .NET 6 库) 启用跟踪，请使用 **MQDOTNET\_TRACE\_ON** 环境变量。值 **-1** 将启用缺省级别跟踪。任何其他大于 0 的整数值将启用详细跟踪。将此环境变量设置为任何其他值都不会启用跟踪。

## 过程

- 要启动和停止跟踪，请使用应用程序配置文件和特定于 IBM MQ 的跟踪配置文件。

### 应用程序配置文件 (**app.config** 或 **web.config**)

应用程序需要在应用程序配置文件（即 **app.config** 或 **web.config** 文件）的 **<appSettings>** 部分下定义 **MQTRACECONFIGFILEPATH** 属性。（应用程序配置文件的实际名称取决于您的应用程序名称。）**MQTRACECONFIGFILEPATH** 属性的值将指定特定于 IBM MQ 的跟踪配置文件 **mqtrace.config** 的位置路径，如以下示例中所示：

```
<appSettings>
<add key="MQTRACECONFIGFILEPATH" value="C:\MQTRACECONFIG" />
</appSettings>
```

如果在所指定应用程序配置文件的路径中找不到 **mqtrace.config** 文件，那么将禁用跟踪。但是，如果应用程序有权写入到当前目录，那么会在应用程序所在目录中创建 First Failure Support Technology (FFST) 和错误日志。

### IBM MQ 特定跟踪配置文件 (**mqtrace.config**)

**mqtrace.config** 文件是一个 XML 文件，其中定义了用于启动和停止跟踪的属性、跟踪文件的路径以及错误日志的路径。下表描述了这些属性。

属性	描述
<b>MQTRACELEVEL</b>	0: 停止跟踪（缺省值）。 1: 启动包含较少详细信息的跟踪。 2: 启动包含完整详细信息的跟踪（推荐）。
<b>MQTRACEPATH</b>	指向将在其中创建跟踪文件的文件夹。如果路径为空白或未定义 <b>MQTRACEPATH</b> 属性，那么将使用应用程序的当前目录。

表 30: mqtrace.config 文件中定义的属性 (继续)	
属性	描述
<b>MQERRORPATH</b>	指向将在其中创建错误日志文件的文件夹。如果路径为空白或未定义 <b>MQERRORPATH</b> 属性, 那么将使用应用程序的当前目录。

以下示例显示了样本 mqtrace.config 文件:

```
<?xml version="1.0" encoding="utf-8"?>
<traceSettings>
  <MQTRACELEVEL>2</MQTRACELEVEL>
  <MQTRACEPATH>C:\MQTRACEPATH</MQTRACEPATH>
  <MQERRORPATH>C:\MQERRORLOGPATH</MQERRORPATH>
</traceSettings>
```

当应用程序正在运行时, 可以通过更改 mqtrace.config 文件中 **MQTRACELEVEL** 属性的值来动态启动和停止跟踪。

运行的应用程序必须具有 **MQTRACELEVEL** 属性所指定的文件夹的创建和写许可权, 才能生成跟踪文件。在 Microsoft Azure PaaS 环境中运行的应用程序还必须确保具有类似的访问许可权, 因为使用在 Microsoft Azure PaaS 中运行的 IBM MQ .NET 组合件的 Web 应用程序可能没有创建和写许可权。如果应用程序不具有指定文件夹的创建和写许可权 (必需), 那么将无法生成跟踪、首次故障数据捕获 (FDC) 和错误日志。

- 要捕获与 SSLStream 类相关的跟踪事件和消息, 请将系统诊断的配置部分添加到应用程序的应用程序配置文件中。

有关更多信息, 请参阅 [为 SSLStream 配置跟踪](#)。

#### 相关概念

[IBM MQ 组件和功能](#)

[可重新分发的客户机](#)

[.NET 应用程序运行时 \(仅限 Windows\)](#)

#### 相关任务

[安装 IBM MQ classes for .NET](#)

[使用 WMQDotnetXAMonitor 应用程序](#)

## 使用命令跟踪 IBM MQ classes for .NET Framework 应用程序

对于 IBM MQ 的完整客户机安装, 您可以使用 **strmqtrc** 命令对 IBM MQ classes for .NET Framework 启用跟踪。

### 关于此任务

通常只需要在 IBM 支持人员的请求下使用跟踪工具。

### 过程

1. 要启动跟踪, 请使用 **strmqtrc** 命令。

有关更多信息, 请参阅 [strmqtrc \(启动跟踪\)](#)。

**注:** **strmqtrc** 命令的 **-i** 和 **-p** 参数 (允许您指定进程和线程标识以及指定的进程) 不会影响 IBM MQ .NET。

2. 要结束跟踪, 请使用 **endmqtrc** 命令。

有关更多信息, 请参阅 [endmqtrc \(结束跟踪\)](#)。

#### 相关任务

第 430 页的『[使用 mqclient.ini 跟踪 IBM MQ .NET 应用程序](#)』

从 IBM MQ 9.4.0 开始, 可以使用 mqclient.ini 文件来启用 IBM MQ .NET 客户机库的跟踪。



## 跟踪 JMS/Jakarta Messaging 和 Java 应用程序

提供了 JMS/Jakarta Messaging 和 Java 应用程序的跟踪工具，以帮助 IBM 支持人员诊断问题。您可以跟踪各种不同的资源。

### 相关任务

第 280 页的『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。

## 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。

### 开始之前

注意:

-  **V 9.4.0**  对于 Long Term Support，将从 IBM MQ 9.4.0 处的产品中除去跟踪控制实用程序。
-  **CD**  对于 Continuous Delivery，将从 IBM MQ 9.3.3 处的产品中除去跟踪控制实用程序。IBM MQ 9.3.2 是交付它的最后一个 Continuous Delivery 发行版。
- 如果需要动态跟踪来诊断问题，那么 IBM 支持人员可根据需要指导您完成收集跟踪的步骤。

### 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 JMS 跟踪。有关更多信息，请参阅第 434 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 435 页的『使用 IBM MQ classes for JMS 配置文件收集 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的错误，First Failure Support Technology (FFST) 信息将记录在名称格式为 JMSCC *xxxx*.FDC（其中 *xxxx* 是一个四位数）的文件中。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 382 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 ...
```

### 相关概念

第 382 页的『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。

### 相关任务

第 280 页的『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。

## 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 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%.cl%u.trc 的文件。
- 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 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%.cl%u.trc 的文件。



- **JM 3.0** **JMS 2.0** 从 IBM MQ 9.3.0 开始，如果应用程序已从可重定位的 JAR 文件 `com.ibm.jakarta.client.jar` (Jakarta Messaging 3.0) 装入 IBM MQ classes for Jakarta Messaging，或从可重定位的 JAR 文件 `com.ibm.mq.allclient.jar` (JMS 2.0) 装入 IBM MQ classes for JMS，那么会将跟踪写入名为 `mjjavaclient_%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 435 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

1. 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](#)。
2. 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。
3. 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。
4. 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 `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.mqjms.jar` 装入 IBM MQ classes for JMS，那么会将跟踪写入名为 `mjjava_%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 `mjjavaclient_%PID%.trc` 的文件。
- 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 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 `mjjava_%PID%.cl%u.trc` 的文件。

- 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 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%.cl%u.trc` 的文件。
-   从 IBM MQ 9.3.0 开始，如果应用程序已从可重定位的 JAR 文件 `com.ibm.jakarta.client.jar` (Jakarta Messaging 3.0) 装入 IBM MQ classes for Jakarta Messaging，或从可重定位的 JAR 文件 `com.ibm.mq.allclient.jar` (JMS 2.0) 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
```

## 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。

### 开始之前

注意：

-   对于 Long Term Support，将从 IBM MQ 9.4.0 处的产品中除去跟踪控制实用程序。
-   对于 Continuous Delivery，将从 IBM MQ 9.3.3 处的产品中除去跟踪控制实用程序。IBM MQ 9.3.2 是交付它的最后一个 Continuous Delivery 发行版。
- 如果需要动态跟踪来诊断问题，那么 IBM 支持人员可根据需要指导您完成收集跟踪的步骤。

### 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 437 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 438 页的『使用 IBM MQ classes for Java 配置文件收集 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 xxxxx.FDC` 的文件名中，其中 `xxxxx` 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 ...
```

## 相关任务

第 280 页的『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。

## 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 `mjqms_%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 `mjava_%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 `mjqjavaclient_%PID%.trc` 的文件。
- 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 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 `mjqjava_%PID%.cl%u.trc` 的文件。
- 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 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 `mjqjavaclient_%PID%.cl%u.trc` 的文件。
- **JM 3.0** 从 IBM MQ 9.3.0 开始，如果应用程序已从可重定位的 JAR 文件 `com.ibm.mq.jakarta.client.jar` (Jakarta Messaging 3.0) 或 `com.ibm.mq.allclient.jar` (JMS 2.0) 装入 IBM MQ classes for Java，那么会将跟踪写入名为 `mjqjavaclient_%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 438 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。
- 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 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%.cl%u.trc` 的文件。
- 从 IBM MQ 9.1.5 和 IBM MQ 9.1.0 Fix Pack 5 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%.cl%u.trc` 的文件。
- **JM 3.0** 从 IBM MQ 9.3.0 开始，如果应用程序已从可重定位的 JAR 文件 `com.ibm.mq.jakarta.client.jar` (Jakarta Messaging 3.0) 或 `com.ibm.mq.allclient.jar` (JMS 2.0) 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
```

## 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。

### 关于此任务

`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 439 页的表 31 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 440 页的表 32 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。



属性的名称	类型	缺省值	描述
logWriterEnabled	字符串	true	启用或禁用将诊断跟踪发送到应用程序服务器提供的 LogWriter 对象的标志。如果值为 true, 那么跟踪发送到 LogWriter 对象。如果值为 false, 那么不会使用应用程序服务器提供的任何 LogWriter 对象。

第 440 页的表 32 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。
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。但是, 您可以通过设置属性在 AIX and Linux 系统上启用诊断跟踪。

## 过程

- 例如, 要在 AIX and Linux 系统上启用诊断跟踪, 您可以设置以下属性:

```
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。您可以使用 **java** 命令上的 **-D** 标志来设置属性, 如以下示例中所示:

```
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。

- 将 WebSphere Application Server traditional 9.0 与 IBM MQ 9.0 资源适配器配合使用时，由于 Java EE 依赖关系注入是一个常见的 Java EE 范例，因此应将标准跟踪字符串更新为包含 `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 property `com.ibm.mq.commonservices` 指定标准 Java 属性文件，该文件包含用于配置诊断输出的多行内容。文件中的每个代码行为自由格式，并以换行符终止。
- System property `com.ibm.mq.commonservices.diagid` 会将跟踪和 FFDC 文件与创建它们的进程相关联。

有关使用 `com.ibm.mq.commonservices` 属性文件来配置诊断信息的内容，请参阅第 441 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 442 页的『[Java 跟踪和 FFDC 文件](#)』。

### 相关概念

第 464 页的『[跟踪 runmqakm](#)』  
如何请求 `runmqakm` 跟踪。

### 相关任务

第 402 页的『[Windows 上的跟踪](#)』

在 Windows 上，可以使用 `strmqtrc` 和 `endmqtrc` 命令来启动和结束跟踪。您还可以使用 IBM MQ Explorer 来启动和结束跟踪。

第 388 页的『[AIX and Linux 上的跟踪](#)』

在 AIX and Linux 上，可以使用 `strmqtrc` 和 `endmqtrc` 命令来启动和结束跟踪，并使用 `dspmqtrc` 来显示跟踪文件。在 AIX 上，除了使用 `strmqtrc` 和 `endmqtrc` 命令外，还可以使用 AIX 系统跟踪。

第 394 页的『[将跟踪与 IBM i 上的 IBM MQ 服务器配合使用](#)』

如果在 IBM i 上安装了 IBM MQ 服务器，那么可以使用 `TRCMQM` 命令来启动和停止跟踪，并指定所需的跟踪类型。或者，您可以通过调用 `QMQM/STRMQTRC` 和 `QMQM/ENDMQTRC` 程序或者在 IBM i Qshell 中使用 `STRMQTRC`，`ENDMQTRC` 和 `DSPMQTRC` 命令从 IBM i 命令行控制跟踪。

第 405 页的『[Tracing on z/OS](#)』

There are different trace options that can be used for problem determination with IBM MQ. Use this topic to understand the different options and how to control trace.

## 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：



### **Diagnostics.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。

### **Diagnostics.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*。

### **Diagnostics.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*。

### **Diagnostics.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*。

### **Diagnostics.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 AIX 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。

### **Diagnostics.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。

### **Diagnostics.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 442 页的图 64 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:\\ffdcdire
#
# Directory for error logging
#
Diagnostics.Java.Errors.Destination.Filename=c:\\errorsdir\\SOAPERRORS.LOG
#
```

图 64: *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 `YYYYMMDDhhmmssmm`。

为 IBM MQ Explorer 生成 Java 跟踪时，会将它写入名称格式为 `AMQYYYYMMDDHHmmssmm.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 `Diagnostics.Java.Errors.Destination.FileName` 为相应的 Java 进程指定的文件。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。

## 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。

### 相关任务

第 280 页的『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。

## 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 443 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 444 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。
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 444 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
```

注: `-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`。

对于名为 AGENT1 的代理程序, 示例命令可能如下所示:

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte=all AGENT1
```

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。
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 AIX

```
./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 447 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 448 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 447 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

使用 `fteSetLoggerTraceLevel` 命令，可以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 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

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。

## Tracing Managed File Transfer for z/OS resources

The trace facility in Managed File Transfer for z/OS is provided to help IBM Support diagnose your problems and issues. You can trace various different resources.

### Procedure

- To trace a Managed File Transfer agent on z/OS, see [“Tracing Managed File Transfer for z/OS agents” on page 449](#).
- To trace a command, see [“Tracing Managed File Transfer for z/OS commands” on page 455](#).
- To trace a Managed File Transfer stand-alone database logger on z/OS see [“Tracing Managed File Transfer for z/OS standalone database loggers” on page 457](#).

### Related tasks

“为 IBM 支持人员收集故障诊断信息” on page 280

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。

## Tracing Managed File Transfer for z/OS agents

The trace facility in Managed File Transfer for z/OS is provided to help IBM Support diagnose your problems and issues. Various commands and properties control the behavior of this facility.

### About this task

If you are asked to provide trace output to investigate an issue with an agent, use one of the following options.

If you are unsure which option to use, contact your IBM Support representative and they will advise you on the best way to collect trace for the issue that you are seeing.

### Procedure

- If it is possible for you to stop an agent for a short period of time, collect a trace of the agent from startup.  
For more information, see [“Collecting a Managed File Transfer for z/OS agent trace from startup” on page 450](#).
- If it is not possible for you to stop an agent, then collect a trace dynamically using the **fteSetAgentTraceLevel** command.  
For more information, see [“Collecting a Managed File Transfer for z/OS agent trace dynamically” on page 452](#).

## **Collecting a Managed File Transfer for z/OS agent trace from startup**

Where it is possible for you to stop an agent for a short period of time, you should collect IBM MQ Managed File Transfer agent trace from startup.

### About this task

The way to collect the trace depends on whether the agent is being administered using z/OS UNIX System Services (z/OS UNIX) or JCL.

If you are unsure which of the following options to use, contact your IBM Support representative and they will advise you on the best way to collect trace for the issue that you are seeing.

### Procedure

- If you are using z/OS UNIX, see [“Collecting an agent trace from startup using z/OS UNIX”](#) on page 450.
- If you are using JCL, see [“Collecting an agent trace from startup using JCL”](#) on page 451.

## **Collecting an agent trace from startup using z/OS UNIX**

To collect a trace of a Managed File Transfer for z/OS agent that is being administered using z/OS UNIX System Services (z/OS UNIX) from startup, you need to set various properties need in the `agent.properties` file for that agent before it is started.

### About this task

To collect a trace from startup, complete the following steps.

### Procedure

1. Locate the `agent.properties` file for the agent that needs to be traced.

The `agent.properties` file can be found in the `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` directory.

2. Edit the files and add entries for the following properties:

- **trace**=*trace specification*

The **trace** property determines the internal classes and packages that are to be traced. Unless otherwise specified by your IBM Support representative, set this property to the value `com.ibm.wmqfte=all`.

- **traceFiles**=*number of trace files to use*
- **traceSize**=*size of each trace file, in MB*

The **traceFiles** and **traceSize** properties are used to control the amount of trace data that is collected. You should set these properties to large values, to collect as much trace data as possible.

For example, to collect 1GB of wrapping trace using the trace specification `com.ibm.wmqfte=all`, add the following lines to the `agent.properties` file:

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

This results in the agent writing trace data to a maximum of 5 files, where each file has a size of 200MB.

For more information on these agent properties, see [The MFT `agent.properties` file](#).

3. Stop the agent that needs to be traced, using the **fteStopAgent** command.
4. Start the agent, by running the **fteStartAgent** command.
5. Reproduce the issue.
6. Stop the agent.

7. Edit the `agent.properties` file for the agent, and remove the entries for the **trace**, **traceFiles**, and **traceSize** properties that you added in step “2” on page 450.

This ensures that trace is not enabled the next time you restart the agent.

## Results

The resultant trace files are written to the `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` directory, where `%PID%` is the process identifier for the agent.

### Collecting an agent trace from startup using JCL

To collect a trace of a Managed File Transfer for z/OS agent that is being administered using JCL from startup, you need to set various properties need in the `agent.properties` file for that agent before it is started.

## About this task

To collect a trace from startup, complete the following steps.

## Procedure

1. Locate the `agent.properties` file for the agent that needs to be traced.

The `agent.properties` file can be found in the `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` directory.

2. Edit the files and add entries for the following properties:

- **trace**=*trace specification*

The **trace** property determines the internal classes and packages that are to be traced. Unless otherwise specified by your IBM Support representative, set this property to the value `com.ibm.wmqfte=all`.

- **traceFiles**=*number of trace files to use*
- **traceSize**=*size of each trace file, in MB*

The **traceFiles** and **traceSize** properties are used to control the amount of trace data that is collected. You should set these properties to large values, to collect as much trace data as possible.

For example, to collect 1GB of wrapping trace using the trace specification `com.ibm.wmqfte=all`, add the following lines to the `agent.properties` file:

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

This results in the agent writing trace data to a maximum of 5 files, where each file has a size of 200MB.

For more information on these agent properties, see [The MFT agent.properties file](#).

3. Locate the data set containing the JCL for the agent that needs to be traced.
4. Submit the BFGAGSP member within the data set to stop the agent.
5. Restart the agent, by submitting the BFGAGST member in the data set .
6. Reproduce the issue.
7. Submit the BFGAGSP member in the data set to stop the agent again.
8. Edit the `agent.properties` file for the agent, and remove the entries for the **trace**, **traceFiles**, and **traceSize** properties that you added in step “2” on page 451.

This ensures that trace is not enabled the next time you restart the agent.

## Results

The resultant trace files are written to the *BFG\_DATA/mqft/logs/coordination\_qmgr\_name/agents/agent\_name/logs/trace%PID%* directory, where %PID% is the process identifier for the agent.

### **Collecting a Managed File Transfer for z/OS agent trace dynamically**

Where it is not possible for you to stop an agent for a short period of time, you should collect Managed File Transfer for z/OS agent trace dynamically.

## About this task

The way to collect the trace depends on whether the agent is being administered using z/OS UNIX System Services (z/OS UNIX) or JCL.

If you are unsure which of the following options to use, contact your IBM Support representative and they will advise you on the best way to collect trace for the issue that you are seeing.

## Procedure

- If you are using:
  - z/OS UNIX, see [“Collecting an agent trace dynamically using z/OS UNIX”](#) on page 452.
  - JCL, see [“Collecting an agent trace dynamically using JCL”](#) on page 453.

### **Collecting an agent trace dynamically using z/OS UNIX**

Under z/OS UNIX System Services (z/OS UNIX), you can use the **fteSetAgentTraceLevel** command to collect trace from a running agent. This can be very useful if IBM Support need to see a trace from an agent that cannot be stopped.

## About this task

To collect a trace from a Managed File Transfer for z/OS agent using the **fteSetAgentTraceLevel** command, complete the following steps.

## Procedure

1. Turn trace on for the agent, by running the following command:

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

The `-traceAgent` parameter determines the internal classes and packages that are to be traced. Unless otherwise specified by your IBM Support representative, set this property to the value `com.ibm.wmqfte=all`.

2. Reproduce the issue.
3. Turn trace off for the agent, by running the following command:

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

4. If an agent is busy, then the trace files might wrap quickly and overwrite the information needed to investigate the issue.

If you can stop the agent for a short period of time, complete the following steps to reduce the amount of trace data that is collected. Otherwise, contact IBM Support and discuss alternative trace specifications to reduce the amount of trace data that is collected.

- a) Schedule some time to stop the agent.
- b) Locate the `agent.properties` file for the agent that needs to be traced.

The `agent.properties` file can be found in the *BFG\_DATA/mqft/config/coordination\_qmgr\_name/agents/agent\_name* directory.

c) Edit the file and add entries for the following properties:

```
traceFiles=number_of_trace_files_to_use  
traceSize=size_of_each_trace_file_in_MB
```

The **traceFiles** and **traceSize** properties are used to control the amount of trace data that is collected.

The default value of the **traceFiles** property is 5, and the **traceSize** property has the default value of 20MB. This means that if you turn on trace dynamically, and you have not set the properties, the agent writes trace information to 5 wrapping trace files, each with a maximum size of 20MB.

You should set these properties to large values, to collect as much trace data as possible.

For example, to collect 1GB of wrapping trace, add the following lines to the `agent.properties` file:

```
traceFiles=5  
traceSize=200
```

This results in the agent writing trace data to a maximum of 5 files, where each file has a size of 200MB.

For more information on these agent properties, see [The MFT agent.properties file](#).

- d) Stop the agent, by running the **fteStopAgent** command.
- e) Start the agent, by running the **fteStartAgent** command.
- f) Turn trace on for the agent, by running the following command:

```
fteSetAgentTraceLevel -traceAgent trace_specification agent_name
```

- g) Reproduce the issue.
- h) Turn trace off for the agent, by running the following command:

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

## Results

The trace files are written to the `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` directory, where `%PID%` is the process identifier for the agent.

### *Collecting an agent trace dynamically using JCL*

You can use the BFGAGTC member within the data set containing the JCL, for the agent that needs to be traced, to collect trace from a running Managed File Transfer for z/OS agent. This can be very useful if IBM Support needs to see a trace from an agent that cannot be stopped.

## About this task

To collect a trace from an agent using the BFGAGTC member, complete the following steps.

## Procedure

1. Locate the data set containing the JCL for the agent that needs to be traced.
2. Edit the BFGAGTC member within the data set, and locate the line that contains the text:

```
-traceAgent
```

The text following this contains the list of internal classes and packages that are to be traced. By default, this list is set to:

```
com.ibm.wmqfte=all
```

Unless otherwise specified by your IBM Support representative, leave this value as is.

3. Submit the BFGAGTC member.
4. Reproduce the issue.
5. Edit the BFGAGTC member again, and set the **-traceAgent** parameter to *=off*, as shown:

```
-traceAgent =off +
```

6. Submit the BFGAGTC member again, to turn trace off.
7. If an agent is busy, then it is possible that the trace files will wrap quickly and overwrite the information needed to investigate the issue.

In this situation there are two options:

a) The first option is to:

- i) Schedule some time to stop the agent.
- ii) Locate the `agent.properties` file for the agent that needs to be traced. The `agent.properties` file can be found in the `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` directory.
- iii) Edit the files and add entries for the following properties:

- **traceFiles**=*number of trace files to use*
- **traceSize**=*size of each trace file, in MB*

The **traceFiles** and **traceSize** properties are used to control the amount of trace data that is collected.

The default value of the **traceFiles** property is 5, and the **traceSize** property has the default value of 20MB. This means that if you turn on trace dynamically, and you have not set the properties, the agent writes trace information to 5 wrapping trace files, each with a maximum size of 20MB.

You should set these properties to large values, to collect as much trace data as possible.

For example, to collect 1GB of wrapping trace, add the following lines to the `agent.properties` file:

```
traceFiles=5  
traceSize=200
```

This results in the agent writing trace data to a maximum of 5 files, where each file has a size of 200MB.

For more information on these agent properties, see [The MFT agent.properties file](#).

- iv) Locate the data set containing the JCL for the agent that needs to be traced.
- v) Submit the BFGAGSP member within the data set to stop the agent.
- vi) Restart the agent, by submitting the BFGAGST member in the data set.
- vii) Edit the BFGAGTC member within the data set, and locate the line that contains the text:

```
-traceAgent
```

The text following this contains the list of internal classes and packages that are to be traced. By default, this list is set to:

```
com.ibm.wmqfte=all
```

Unless otherwise specified by your IBM Support representative, leave this value as is.

- viii) When it is time to enable trace, submit the BFGAGTC member.

ix) Reproduce the issue.

x) Edit the BFGAGTC member again, and set the **-traceAgent** parameter to *=off*, as shown:

```
-traceAgent =off +
```

xi) Submit the BFGAGTC member again, to turn trace off.

b) The second option is to contact your IBM Support representative, if it is not possible to stop the agent for a short period of time.

You can then discuss alternative trace specifications to use, in order to reduce the amount of trace data that is being generated.

## Results

The resultant trace files are written to the *BFG\_DATA/mqft/logs/coordination\_qmgr\_name/agents/agent\_name/logs/trace%PID%* directory, where %PID% is the process identifier for the agent.

## z/OS Tracing Managed File Transfer for z/OS commands

The trace facility in Managed File Transfer for z/OS is provided to help IBM Support diagnose your problems and issues. You can use this facility to trace commands.

### About this task



**Attention:** Tracing a command only collects information about the processing done by the command. It does not trace any activity that an agent might perform while processing that command.

The way to collect the trace depends on whether the command is being run using either z/OS UNIX System Services (z/OS UNIX) or JCL.

If you are unsure which option to use, contact your IBM Support representative and they will advise you on the best way to collect trace for the issue that you are seeing.

### Procedure

- If you are using z/OS UNIX, see [“Collecting an agent trace from startup using z/OS UNIX” on page 450](#).
- If you are using JCL, see [“Collecting an agent trace from startup using JCL” on page 451](#).

## z/OS 使用 z/OS UNIX 收集命令的跟踪信息

要使用 z/OS UNIX System Services (z/OS UNIX) 收集 Managed File Transfer for z/OS 命令的跟踪，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。

## 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 z/OS UNIX System Services 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 456 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 **ftelListMonitors** 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>
//*****
//* ftelListMonitors
//*****
//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=ftelListMonitors -trace com.ibm.wmqfte=all -tracePath /u/trace +
-v -p QM1
/*
//
```

提交该成员后，**ftelListMonitors** 命令会将跟踪信息写入到 z/OS UNIX 目录 /u/trace。

## Tracing Managed File Transfer for z/OS standalone database loggers

The trace facility in Managed File Transfer for z/OS is provided to help IBM Support diagnose your problems and issues. Various commands and properties control the behavior of this facility.

### About this task

If you are asked to provide trace output to investigate an issue with a standalone database logger, use one of the following options.

If you are unsure which option to use, contact your IBM Support representative and they will advise you on the best way to collect trace for the issue that you are seeing.

### Procedure

- If it is possible for you to stop a logger for a short period of time, collect a trace of the logger from startup.  
For more information, see [“Collecting a Managed File Transfer for z/OS standalone database logger trace from startup” on page 458.](#)
- If it is not possible for you to stop a logger, then collect a trace dynamically using the **fteSetLoggerTraceLevel** command.  
For more information, see [“Collecting a Managed File Transfer for z/OS standalone database logger trace dynamically” on page 460.](#)

## **Collecting a Managed File Transfer for z/OS standalone database logger trace from startup**

Where it is possible for you to stop a logger for a short period of time, you should collect IBM MQ Managed File Transfer logger trace from startup.

### **About this task**

The way to collect the trace depends on whether the logger is being administered using z/OS UNIX System Services (z/OS UNIX) or JCL.

If you are unsure which of the following options to use, contact your IBM Support representative and they will advise you on the best way to collect trace for the issue that you are seeing.

### **Procedure**

- If you are using:
  - z/OS UNIX, see [“Collecting a standalone database logger trace from startup using z/OS UNIX” on page 458.](#)
  - JCL, see [“Collecting a standalone database logger trace from startup using JCL” on page 459.](#)

## **Collecting a standalone database logger trace from startup using z/OS UNIX**

To collect a trace of a Managed File Transfer for z/OS logger that is being administered using z/OS UNIX System Services (z/OS UNIX) from startup, you need to set various properties in the `logger.properties` file for that logger before it is started.

### **About this task**

To collect a trace from startup, complete the following steps.

### **Procedure**

1. Locate the `logger.properties` file for the logger that needs to be traced.

The `logger.properties` file can be found in the `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` directory.

2. Edit the file and add entries for the following properties:

- **trace**=*trace specification*

The **trace** property determines the internal classes and packages that are to be traced. Unless otherwise specified by your IBM Support representative, set this property to the value `com.ibm.wmqfte=all`.

- **traceFiles**=*number of trace files to use*
- **traceSize**=*size of each trace file, in MB*

The **traceFiles** and **traceSize** properties are used to control the amount of trace data that is collected. You should set these properties to large values, to collect as much trace data as possible.

For example, to collect 1GB of wrapping trace using the trace specification `com.ibm.wmqfte=all`, add the following lines to the `logger.properties` file:

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

This results in the logger writing trace data to a maximum of 5 files, where each file has a size of 200MB.

For more information on these logger properties, see [The MFT logger.properties file.](#)

3. Stop the logger that needs to be traced, using the **fteStopLogger** command.

4. Start the logger, by running the **fteStartLogger** command.
5. Reproduce the issue.
6. Stop the logger.
7. Edit the `logger.properties` file for the logger, and remove the entries for the **trace**, **traceFiles**, and **traceSize** properties that you added in step “2” on page 458.

This ensures that trace is not enabled the next time you restart the logger.

## Results

The resultant trace files are written to the `BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` directory, where %PID% is the process identifier for the logger.

### *Collecting a standalone database logger trace from startup using JCL*

To collect a trace of a Managed File Transfer for z/OS logger that is being administered using JCL from startup, you need to set various properties in the `logger.properties` file for that logger before it is started.

## About this task

To collect a trace from startup, complete the following steps.

## Procedure

1. Locate the `logger.properties` file for the logger that needs to be traced.

The `logger.properties` file can be found in the `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` directory.

2. Edit the file and add entries for the following properties:

- **trace**=*trace specification*

The **trace** property determines the internal classes and packages that are to be traced. Unless otherwise specified by your IBM Support representative, set this property to the value `com.ibm.wmqfte=all`.

- **traceFiles**=*number of trace files to use*
- **traceSize**=*size of each trace file, in MB*

The **traceFiles** and **traceSize** properties are used to control the amount of trace data that is collected. You should set these properties to large values, to collect as much trace data as possible.

For example, to collect 1GB of wrapping trace using the trace specification `com.ibm.wmqfte=all`, add the following lines to the `logger.properties` file:

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

This results in the logger writing trace data to a maximum of 5 files, where each file has a size of 200MB.

For more information on these logger properties, see [The MFT logger.properties file](#).

3. Locate the data set containing the JCL for the logger that needs to be traced.
4. Submit the BFGLGSP member within the data set to stop the logger.
5. Restart the logger, by submitting the BFGLGST member in the data set.
6. Reproduce the issue.
7. Submit the BFGLGSP member in the data set to stop the logger again.
8. Edit the `logger.properties` file for the logger, and remove the entries for the **trace**, **traceFiles**, and **traceSize** properties that you added in step “2” on page 459.

This ensures that trace is not enabled the next time you restart the logger.

## Results

The resultant trace files are written to the `BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` directory, where `%PID%` is the process identifier for the logger.

## **Collecting a Managed File Transfer for z/OS standalone database logger trace dynamically**

Where it is not possible for you to stop a logger for a short period of time, you should collect Managed File Transfer for z/OS logger trace dynamically.

## About this task

The way to collect the trace depends on whether the logger is being administered using z/OS UNIX System Services (z/OS UNIX) or JCL.

If you are unsure which of the following options to use, contact your IBM Support representative and they will advise you on the best way to collect trace for the issue that you are seeing.

## Procedure

- If you are using:
  - z/OS UNIX, see [“Collecting a standalone database logger trace dynamically using z/OS UNIX” on page 460.](#)
  - JCL, see [“Collecting a standalone database logger trace dynamically using JCL” on page 461.](#)

## **Collecting a standalone database logger trace dynamically using z/OS UNIX**

Under z/OS UNIX System Services (z/OS UNIX), you can use the **fteSetLoggerTraceLevel** command to collect trace from a running logger. This can be very useful if IBM Support need to see a trace from a logger that cannot be stopped.

## About this task

To collect a trace from a Managed File Transfer for z/OS logger using the **fteSetLoggerTraceLevel** command, complete the following steps.

## Procedure

1. Turn trace on for the logger, by running the following command:

```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

The `-traceLogger` parameter determines the internal classes and packages that are to be traced. Unless otherwise specified by your IBM Support representative, set this property to the value `com.ibm.wmqfte=all`.

2. Reproduce the issue.
3. Turn trace off for the logger, by running the following command:

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

4. If a logger is busy, then the trace files might wrap quickly and overwrite the information needed to investigate the issue.

If you can stop the logger for a short period of time, complete the following steps to reduce the amount of trace data that is collected. Otherwise, contact IBM Support and discuss alternative trace specifications to reduce the amount of trace data that is collected.

- a) Schedule some time to stop the logger.
- b) Locate the `logger.properties` file for the logger that needs to be traced.  
The `logger.properties` file can be found in the `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` directory.
- c) Edit the file and add entries for the following properties:

**traceFiles**=*number\_of\_trace\_files\_to\_use*  
**traceSize**=*size\_of\_each\_trace\_file\_in\_MB*

The **traceFiles** and **traceSize** properties are used to control the amount of trace data that is collected.

The default value of the **traceFiles** property is 5, and the **traceSize** property has the default value of 20MB. This means that if you turn on trace dynamically, and you have not set the properties, the agent writes trace information to 5 wrapping trace files, each with a maximum size of 20MB.

You should set these properties to large values, to collect as much trace data as possible.

For example, to collect 1GB of wrapping trace, add the following lines to the `logger.properties` file:

```
traceFiles=5
traceSize=200
```

This results in the logger writing trace data to a maximum of 5 files, where each file has a size of 200MB.

For more information on these logger properties, see [The MFT logger.properties file](#).

- d) Stop the logger, by running the **fteStopLogger** command.
- e) Start the logger, by running the **fteStartLogger** command.
- f) Turn trace on for the logger, by running the following command:

```
fteSetLoggerTraceLevel -traceLogger trace_specification logger_name
```

- g) Reproduce the issue.
- h) Turn trace off for the logger, by running the following command:

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

## Results

The resultant trace files are written to the `BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` directory, where `%PID%` is the process identifier for the logger.

**z/OS** *Collecting a standalone database logger trace dynamically using JCL*

You can use the BFGLGTC member within the dataset containing the JCL, for the logger that needs to be traced, to collect trace from a running Managed File Transfer for z/OS logger. This can be very useful if IBM Support need to see a trace from a logger that cannot be stopped.

## About this task

To collect a trace from a logger using the BFGLGTC member, complete the following steps.

## Procedure

1. Locate the dataset containing the JCL for the logger that needs to be traced.

2. Edit the BFGLGTC member within the dataset, and locate the line that contains the text:

```
-traceLogger
```

The text following this contains the list of internal classes and packages that are to be traced. By default, this list is set to:

```
com.ibm.wmqfte=all
```

Unless otherwise specified by your IBM Support representative, leave this value as is.

3. Submit the BFGLGTC member.
4. Reproduce the issue.
5. Edit the BFGLGTC member again, and set the **-traceLogger** parameter to *=off*, as shown:

```
-traceLogger =off +
```

6. Submit the BFGLGTC member again, to turn trace off.
7. If a logger is busy, then the trace files might wrap quickly and overwrite the information needed to investigate the issue.

If you can stop the logger for a short period of time, complete the following steps to reduce the amount of trace data that is collected. Otherwise, contact IBM Support and discuss alternative trace specifications to reduce the amount of trace data that is collected.

- a) Schedule some time to stop the logger.
- b) Locate the `logger.properties` file for the logger that needs to be traced.

The `logger.properties` file can be found in the `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` directory.

- c) Edit the file and add entries for the following properties:

```
traceFiles=number_of_trace_files_to_use  
traceSize=size_of_each_trace_file_in_MB
```

The **traceFiles** and **traceSize** properties are used to control the amount of trace data that is collected.

The default value of the **traceFiles** property is 5, and the **traceSize** property has the default value of 20MB. This means that if you turn on trace dynamically, and you have not set the properties, the agent writes trace information to 5 wrapping trace files, each with a maximum size of 20MB.

You should set these properties to large values, to collect as much trace data as possible.

For example, to collect 1GB of wrapping trace, add the following lines to the `logger.properties` file:

```
traceFiles=5  
traceSize=200
```

This results in the logger writing trace data to a maximum of 5 files, where each file has a size of 200MB.

For more information on these logger properties, see [The MFT logger.properties file](#).

- d) Locate the data set containing the JCL for the logger that needs to be traced.
- e) Submit the BFGLGSP member within the data set to stop the logger.
- f) Restart the logger, by submitting the BFGLGST member in the data set.
- g) Edit the BFGLGTC member within the data set, and locate the line that contains the following text:

```
-traceLogger
```



The text following this contains the list of internal classes and packages that are to be traced. By default, this list is set to:

```
com.ibm.wmqfte=all
```

Unless otherwise specified by your IBM Support representative, leave this value as is.

- h) When it is time to enable trace, submit the BFGLGTC member.
- i) Reproduce the issue.
- j) Edit the BFGLGTC member again, and set the **-traceLogger** parameter to *=off* by running the following command:

```
-traceLogger =off +
```

- k) Submit the BFGLGTC member again, to turn trace off.

## Results

The trace files are written to the *BFG\_DATA/mqft/logs/coordination\_qmgr\_name/loggers/logger\_name/logs/trace%PID%* directory, where *%PID%* is the process identifier for the logger.

## 跟踪 REST API

提供了 REST API 中的跟踪工具，以帮助 IBM 支持代表诊断问题。可通过各种属性来控制这些功能的行为。

### 开始之前

为 IBM 支持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：

```
ALW MQ_DATA_PATH/web/installations/installationName
```

```
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。
```

### 关于此任务

REST API 包含两个功能区域，每个功能区域都有自己的跟踪机制：

- 在 mqweb 服务器中运行的 REST API 代码的跟踪。
- 在 mqweb 服务器中运行的 REST API 代码的消息传递跟踪。

有关对 IBM MQ Console 启用跟踪的信息，请参阅第 424 页的『跟踪 IBM MQ Console』。

### 过程

1. 为 mqweb 服务器中运行的 REST API 代码启用跟踪

- a) 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。在其中创建跟踪文件的目录为：

- ALW MQ\_DATA\_PATH/web/installations/installationName/servers/mqweb/logs

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

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](#)。

2. 为 mqweb 服务器中运行的 REST API 代码启用消息传递跟踪

- a) 在以下某个目录中创建名为 `jmstrace.config` 的文件：

```
ALW 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=0N
```

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

第 280 页的『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。

## 跟踪 runmqakm

如何请求 **runmqakm** 跟踪。

### runmqakm 跟踪

要请求 **runmqakm** 跟踪，请运行带有以下标志的 **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，如果使用 FTP 将其传输到 IBM 支持，那么必须以二进制传输方式传输该文件。

## 相关任务

第 280 页的『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。

## 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** 设置为环境变量来配置对环境变量的跟踪。
- 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 386 页的『[FFST:WCF XMS First Failure Support Technology](#)』

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 220 页的『[对 IBM MQ 问题的 WCF 定制通道进行故障诊断](#)』

用于帮助您解决 IBM MQ 应用程序的 Microsoft Windows Communication Foundation (WCF) 定制通道运行问题的故障诊断信息。

第 279 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](#)

## 跟踪 XMS .NET 应用程序

如果您正在使用 IBM MQ classes for XMS .NET Framework，那么可以从应用程序配置文件以及 XMS 环境变量配置跟踪。如果您正在使用 IBM MQ classes for XMS .NET (.NET Standard 和 .NET 6 库)，那么必须从 XMS 环境变量配置跟踪。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。

如果您正在使用 IBM MQ classes for XMS .NET Framework，那么可以通过以下方式开启跟踪并配置跟踪设置：

- 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 467 页的『使用应用程序配置文件跟踪 XMS .NET 应用程序』。
- 对 XMS C 或 C++ 应用程序使用 XMS 环境变量。有关更多信息，请参阅第 469 页的『使用 XMS 环境变量跟踪 XMS .NET 应用程序』。
- **V9.4.0** 通过使用 `mqclient.ini` 文件并设置 `Trace` 节的相应属性。您还可以使用 `mqclient.ini` 文件动态启用和禁用跟踪。有关更多信息，请参阅第 470 页的『使用 `mqclient.ini` 跟踪 XMS .NET 应用程序』。

IBM MQ classes for XMS .NET (.NET Standard 和 .NET 6 库)不支持使用应用程序配置文件。如果您正在使用 IBM MQ classes for XMS .NET (.NET Standard 和 .NET 6 库)，那么可以通过以下方式配置跟踪：

- 从 XMS 环境变量。有关更多信息，请参阅第 469 页的『使用 XMS 环境变量跟踪 XMS .NET 应用程序』。
- **V9.4.0** 从 IBM MQ 9.3.3 开始，通过使用 `mqclient.ini` 文件并设置 `Trace` 节的相应属性。您还可以使用 `mqclient.ini` 文件动态启用和禁用跟踪。有关更多信息，请参阅第 470 页的『使用 `mqclient.ini` 跟踪 XMS .NET 应用程序』。

注：

对于针对 .NET Framework 构建的 IBM MQ .NET 客户机库，优先顺序如下所示：

1. App.Config
2. 环境变量
3. `mqclient.ini`

对于针对 .NET Standard 和 .NET 6 构建的 IBM MQ .NET 客户机库，优先顺序如下所示：

1. 环境变量
2. `mqclient.ini`

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

### 相关概念

第 387 页的『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。

### 相关任务

第 279 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。

安装 IBM MQ classes for XMS .NET

从 NuGet 存储库下载 IBM MQ classes for XMS .NET Standard

## 使用应用程序配置文件跟踪 XMS .NET 应用程序

如果您正在使用 IBM MQ classes for XMS .NET Framework，那么可以使用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

IBM MQ classes for XMS .NET Framework 支持使用应用程序文件来配置跟踪。

IBM MQ classes for XMS .NET (.NET Standard 和 .NET 6 库)不支持使用应用程序配置文件。如果您正在使用 IBM MQ classes for XMS .NET (.NET Standard 和 .NET 6 库)，那么必须从 XMS 环境变量配置跟踪。有关更多信息，请参阅第 469 页的『使用 XMS 环境变量跟踪 XMS .NET 应用程序』。 **V9.4.0** 从 IBM MQ 9.3.3 开始，您还可以使用 mqclient.ini 文件。有关更多信息，请参阅第 470 页的『使用 mqclient.ini 跟踪 XMS .NET 应用程序』。

### 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>
```

第 468 页的表 33 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"> <li>- 全部</li> <li>- debug</li> <li>- 事件</li> <li>- EntryExit</li> </ul> <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 环境变量跟踪 XMS .NET 应用程序

您可以使用 XMS 环境变量（例如，**XMS\_TRACE\_ON**）来开启跟踪。

### V 9.4.0

使用 mqclient.ini 跟踪 XMS .NET 应用程序

从 IBM MQ 9.4.0 开始，可以使用 mqclient.ini 文件来启用 XMS .NET 客户机库的跟踪。

## 使用 XMS 环境变量跟踪 XMS .NET 应用程序

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。

如果您正在使用 IBM MQ classes for XMS .NET (.NET Standard 和 .NET 6 库)，那么必须从 XMS 环境变量配置跟踪。IBM MQ classes for XMS .NET (.NET Standard 和 .NET 6 库)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。



环境变量	缺省	设置	含义
XMS_TRACE_SPECIFICATION	不适用	请参阅第 467 页的『使用应用程序配置文件跟踪 XMS .NET 应用程序』(仅限 IBM MQ classes for XMS .NET Framework)	覆盖跟踪规范, 该规范遵循第 467 页的『使用应用程序配置文件跟踪 XMS .NET 应用程序』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跟踪 XMS .NET 应用程序

如果您正在使用 IBM MQ classes for XMS .NET Framework, 那么可以使用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 9.4.0** 使用 mqclient.ini 跟踪 XMS .NET 应用程序

从 IBM MQ 9.4.0 开始, 可以使用 mqclient.ini 文件来启用 XMS .NET 客户机库的跟踪。

### 相关参考

环境变量描述

**V 9.4.0** 使用 mqclient.ini 跟踪 XMS .NET 应用程序

从 IBM MQ 9.4.0 开始, 可以使用 mqclient.ini 文件来启用 XMS .NET 客户机库的跟踪。

### 关于此任务

从 IBM MQ 9.4.0 开始, mqclient.ini 文件包含跟踪节。要启用跟踪, 必须设置 Trace 节的相应属性。如果属性设置为无效值, 那么将忽略这些属性。

您还可以动态启用和禁用跟踪。当应用程序正在运行时, 如果修改, 创建或删除了 mqclient.ini 文件, 那么 XMS .NET 客户机将再次读取跟踪部分的属性, 然后启用或禁用跟踪, 以便不需要重新启动应用程序。

注: 有几种不同的方法来启用跟踪。

对于针对 .NET Framework 构建的 XMS .NET 客户机库, 优先顺序如下所示:

1. App.Config
2. 环境变量
3. mqclient.ini

对于针对 .NET Standard 和 .NET 6 构建的 XMS .NET 客户机库, 优先顺序如下所示:

1. 环境变量
2. mqclient.ini

### 过程

- 要启动跟踪, 请指定 Trace 节的以下属性:
  - a) 将 **XMSDotnetTraceLevel** 设置为与所需跟踪级别对应的值。
    - 如果要使用高级格式启动跟踪, 请将此属性设置为 2。如果要使用基本格式启动跟踪, 请将属性设置为 1。
  - b) 将 **XMSDotnetTraceFilePath** 设置为要在其中创建跟踪文件的文件夹的文件路径。
    - 例如: XMSDotnetTraceFilePath="c:\somepath"。如果路径留空或未定义 **XMSDotnetTraceFilePath** 属性, 那么将使用应用程序的当前目录。
  - c) 指定要与 **XMSDotnetTraceSpecification** 一起包含的跟踪元素。

您可使用“:”（冒号）定界符将多个跟踪元素串在一起。

- *ComponentName* 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，`*=all=enabled` 指定您要跟踪所有类，`IBM.XMS.impl.*=all=enabled` 指定您只需要 API 跟踪。
  - *type* 可以是下列任何跟踪类型：`all`，`debug`，`event` 和 `EntryExit`。
  - *state* 可以启用或禁用。
- d) 使用 **XMSDotnetTraceFileSize** 设置跟踪文件大小。  
缺省最大值为 20 MB，指定为 `XMSDotnetTraceFileSize=20`。
- e) 设置要使用 **XMSDotnetTraceFileNumber** 保留的跟踪文件数。  
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。
- 要停止跟踪，请将 **XMSDotnetTraceLevel** 设置为 0。  
这是此属性的缺省值。

### 相关概念

[客户机配置文件的跟踪节](#)

### 相关任务

[使用应用程序配置文件跟踪 XMS .NET 应用程序](#)

如果您正在使用 IBM MQ classes for XMS .NET Framework，那么可以使用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。

[使用 XMS 环境变量跟踪 XMS .NET 应用程序](#)

您可以使用 XMS 环境变量（例如，**XMS\_TRACE\_ON**）来开启跟踪。


## 启用 LDAP 客户机库代码的动态跟踪

您可以在不停止或启动队列管理器的情况下打开和关闭 LDAP 客户机跟踪。

### 关于此任务

要启用此行为，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。

 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。

### 相关任务

[第 280 页的『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。

## 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 473 页的『磁盘驱动器故障』](#)
- [第 473 页的『损坏的队列管理器对象』](#)
- [第 473 页的『损坏的单个对象』](#)
- [第 474 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 475 页的『Shared queue problems』](#)
- [z/OS 第 475 页的『Active log problems』](#)
- [z/OS 第 481 页的『Archive log problems』](#)
- [z/OS 第 483 页的『BSDS problems』](#)
- [z/OS 第 490 页的『Page set problems』](#)
- [z/OS 第 492 页的『Coupling facility and Db2 problems』](#)
- [z/OS 第 494 页的『Problems with long-running units of work』](#)
- [z/OS 第 495 页的『IMS-related problems』](#)
- [z/OS 第 497 页的『Hardware problems』](#)

### 相关任务

[第 279 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 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。

[备份和复原 IBM MQ](#)

## 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 \* 指示将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

如果队列管理器对象本身已损坏，那么队列管理器将执行先发制人的关闭。

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**，重新启动队列管理器。受损队列管理器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。

## 损坏的单个对象

如果在正常操作期间报告单个对象已损坏，那么对于线性日志记录和复制的日志记录，可以从其介质映像重新创建该对象。但是，对于循环日志记录，您不能重新创建单个对象。

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已过时。

有关从介质映像恢复的信息，请参阅 [恢复受损对象](#)。

## 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** Example recovery procedures on z/OS

Use this topic as a reference for various recovery procedures.

This topic describes procedures for recovering IBM MQ after various error conditions. These error conditions are grouped in the following categories:

Problem category	Problem	Where to look next
Shared queue problems	Conflicting definitions for both private and shared queues.	<a href="#">“Shared queue problems” on page 475</a>
Active log problems	<ul style="list-style-type: none"> <li>Dual logging is lost.</li> <li>Active log has stopped.</li> <li>One or both copies of the active log data set are damaged.</li> <li>Write errors on active log data set.</li> <li>Active log is becoming full or is full.</li> <li>Read errors on active log data set.</li> </ul>	<a href="#">“Active log problems” on page 475</a>
Archive log problems	<ul style="list-style-type: none"> <li>Insufficient DASD space to complete offloading active log data sets.</li> <li>Offload task has terminated abnormally.</li> <li>Archive data set allocation problem. <a href="#">1</a></li> <li>Read I/O errors on the archive data set during restart.</li> </ul>	<a href="#">“Archive log problems” on page 481</a>
BSDS problems	<ul style="list-style-type: none"> <li>Error opening BSDS.</li> <li>Log content does not correspond with BSDS information.</li> <li>Both copies of the BSDS are damaged.</li> <li>Unequal time stamps.</li> <li>Dual BSDS data sets are out of synchronization.</li> <li>I/O error on BSDS.</li> </ul>	<a href="#">“BSDS problems” on page 483</a>
Page set problems	<ul style="list-style-type: none"> <li>Page set full.</li> <li>A page set has an I/O error.</li> </ul>	<a href="#">“Page set problems” on page 490</a>

Table 35. Example recovery procedures (continued)

Problem category	Problem	Where to look next
coupling facility and Db2 problems	<ul style="list-style-type: none"> <li>Storage medium full.</li> <li>Db2 system fails.</li> <li>Db2 data-sharing group fails.</li> <li>Db2 and the coupling facility fail.</li> </ul>	<a href="#">“Coupling facility and Db2 problems” on page 492</a>
Unit of work problems	A long-running unit of work is encountered.	<a href="#">“Problems with long-running units of work” on page 494</a>
IMS problems	<ul style="list-style-type: none"> <li>An IMS application terminates abnormally.</li> <li>The IMS adapter cannot connect to IBM MQ.</li> <li>IMS not operational.</li> </ul>	<a href="#">“IMS-related problems” on page 495</a>
Hardware problems	Media recovery procedures	<a href="#">“Hardware problems” on page 497</a>

## Shared queue problems

Problems occur if IBM MQ discovers that a page set based queue, and a shared queue of the same name are defined.

### Symptoms

IBM MQ issues the following message:

```
CSQI063E +CSQ1 QUEUE queue-name IS BOTH PRIVATE AND SHARED
```

During queue manager restart, IBM MQ discovered that a page set based queue and a shared queue of the same name coexist.

### System action

Once restart processing has completed, any MQOPEN request to that queue name fails, indicating the coexistence problem.

### System programmer action

None.

### Operator action

Delete one version of the queue to allow processing of that queue name. If there are messages on the queue that must be kept, you can use the MOVE QLOCAL command to move them to the other queue.

## Active log problems

Use this topic to resolve different problems with the active logs.

This topic covers the following active log problems:

- [“Dual logging is lost” on page 476](#)
- [“Active log stopped” on page 476](#)
- [“One or both copies of the active log data set are damaged” on page 477](#)
- [“Write I/O errors on an active log data set” on page 477](#)
- [“I/O errors occur while reading the active log” on page 478](#)
- [“Active log is becoming full” on page 479](#)



- Active log is full

## Dual logging is lost

### Symptoms

IBM MQ issues the following message:

```
CSQJ004I +CSQ1 ACTIVE LOG COPY n INACTIVE, LOG IN SINGLE MODE,  
ENDRBA=...
```

Having completed one active log data set, IBM MQ found that the subsequent (COPY *n*) data sets were not offloaded or were marked stopped.

### System action

IBM MQ continues in single mode until offloading has been completed, then returns to dual mode.

### System programmer action

None.

### Operator action

Check that the offload process is proceeding and is not waiting for a tape mount. You might need to run the print log map utility to determine the state of all data sets. You might also need to define additional data sets.

## Active log stopped

### Symptoms

IBM MQ issues the following message:

```
CSQJ030E +CSQ1 RBA RANGE starttrba TO endtrba NOT AVAILABLE IN ACTIVE  
LOG DATA SETS
```

### System action

The active log data sets that contain the RBA range reported in message CSQJ030E are unavailable to IBM MQ. The status of these logs is STOPPED in the BSDS. The queue manager terminates with a dump.

### System programmer action

You must resolve this problem before restarting the queue manager. The log RBA range must be available for IBM MQ to be recoverable. An active log that is marked as STOPPED in the BSDS will never be reused or archived and this creates a hole in the log.

Look for messages that indicate why the log data set has stopped, and follow the instructions for those messages.

Modify the BSDS active log inventory to reset the STOPPED status. To do this, follow this procedure after the queue manager has terminated:

1. Use the print log utility (CSQJU004) to obtain a copy of the BSDS log inventory. This shows the status of the log data sets.
2. Use the DELETE function of the change log inventory utility (CSQJU003) to delete the active log data sets that are marked as STOPPED.
3. Use the NEWLOG function of CSQJU003 to add the active logs back into the BSDS inventory. The starting and ending RBA for each active log data set must be specified on the NEWLOG statement. (The correct values to use can be found from the print log utility report obtained in Step 1.)
4. Rerun CSQJU004. The active log data sets that were marked as STOPPED are now shown as NEW and NOT REUSABLE. These active logs will be archived in due course.



5. Restart the queue manager.

**Note:** If your queue manager is running in dual BSDS mode, you must update both BSDS inventories.

## One or both copies of the active log data set are damaged

### Symptoms

IBM MQ issues the following messages:

```
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
```

### System action

Queue manager startup processing is terminated.

### System programmer action

If one copy of the data set is damaged, carry out these steps:

1. Rename the damaged active log data set and define a replacement data set.
2. Copy the undamaged data set to the replacement data set.
3. Use the change log inventory utility to:
  - Remove information relating to the damaged data set from the BSDS.
  - Add information relating to the replacement data set to the BSDS.
4. Restart the queue manager.

If both copies of the active log data sets are damaged, the current page sets are available, **and the queue manager shut down cleanly**, carry out these steps:

1. Rename the damaged active log data sets and define replacement data sets.
2. Use the change log records utility to:
  - Remove information relating to the damaged data set from the BSDS.
  - Add information relating to the replacement data set to the BSDS.
3. Rename the current page sets and define replacement page sets.
4. Use CSQUTIL (FORMAT and RESETPAGE) to format the replacement page sets and copy the renamed page sets to them. The RESETPAGE function also resets the log information in the replacement page sets.

If the queue manager did not shut down cleanly, you must either restore your system from a previous known point of consistency, or perform a cold start (described in [Reinitializing a queue manager](#)).

### Operator action

None.

## Write I/O errors on an active log data set

### Symptoms

IBM MQ issues the following message:

```
CSQJ105E +CSQ1 csect-name LOG WRITE ERROR DSNAME=... ,  
          LOGRBA=... , ERROR STATUS=ccccffss
```

### System action

IBM MQ carries out these steps:

1. Marks the log data set that has the error as TRUNCATED in the BSDS.
2. Goes on to the next available data set.
3. If dual active logging is used, truncates the other copy at the same point.

The data in the truncated data set is offloaded later, as usual.

The data set will be reused on the next cycle.

### System programmer action

None.

### Operator action

If errors on this data set still exist, shut down the queue manager after the next offload process. Then use Access Method Services (AMS) and the change log inventory utility to add a replacement. (For instructions, see [Changing the BSDS](#).)

## I/O errors occur while reading the active log

### Symptoms

IBM MQ issues the following message:

```
CSQJ106E +CSQ1 LOG READ ERROR DSNAME=..., LOGRBA=...,  
ERROR STATUS=ccccffss
```

### System action

This depends on when the error occurred:

- If the error occurs during the offload process, the process tries to read the RBA range from a second copy.
  - If no second copy exists, the active log data set is stopped.
  - If the second copy also has an error, only the original data set that triggered the offload process is stopped. The archive log data set is then terminated, leaving a gap in the archived log RBA range.
  - This message is issued:

```
CSQJ124E +CSQ1 OFFLOAD OF ACTIVE LOG SUSPENDED FROM  
RBA xxxxxx TO RBA xxxxxx DUE TO I/O ERROR
```

- If the second copy is satisfactory, the first copy is not stopped.
- If the error occurs during recovery, IBM MQ provides data from specific log RBAs requested from another copy or archive. If this is unsuccessful, recovery does not succeed, and the queue manager terminates abnormally.
- If the error occurs during restart, if dual logging is used, IBM MQ continues with the alternative log data set, otherwise the queue manager ends abnormally.

### System programmer action

Look for system messages, such as IEC prefixed messages, and try to resolve the problem using the recommended actions for these messages.

If the active log data set has been stopped, it is not used for logging. The data set is not deallocated; it is still used for reading. Even if the data set is not stopped, an active log data set that gives persistent errors should be replaced.

## Operator action

None.

## Replacing the data set

How you replace the data set depends on whether you are using single or dual active logging.

### ***If you are using dual active logging:***

1. Ensure that the data has been saved.

The data is saved on the other active log and this can be copied to a replacement active log.

2. Stop the queue manager and delete the data set with the error using Access Method Services.
3. Redefine a new log data set using Access Method Services DEFINE so that you can write to it. Use DFDSS or Access Method Services REPRO to copy the good log in to the redefined data set so that you have two consistent, correct logs again.
4. Use the change log inventory utility, CSQJU003, to update the information in the BSDS about the corrupted data set as follows:
  - a. Use the DELETE function to remove information about the corrupted data set.
  - b. Use the NEWLOG function to name the new data set as the new active log data set and give it the RBA range that was successfully copied.

You can run the DELETE and NEWLOG functions in the same job step. Put the DELETE statement before NEWLOG statement in the SYSIN input data set.

5. Restart the queue manager.

### ***If you are using single active logging:***

1. Ensure that the data has been saved.
2. Stop the queue manager.
3. Determine whether the data set with the error has been offloaded:
  - a. Use the CSQJU003 utility to list information about the archive log data sets from the BSDS.
  - b. Search the list for a data set with an RBA range that includes the RBA of the corrupted data set.
4. If the corrupted data set has been offloaded, copy its backup in the archive log to a new data set. Then, skip to step [6](#).
5. If an active log data set is stopped, an RBA is not offloaded. Use DFDSS or Access Method Services REPRO to copy the data from the corrupted data set to a new data set.

If further I/O errors prevent you from copying the entire data set, a gap occurs in the log.

**Note:** Queue manager restart will not be successful if a gap in the log is detected.

6. Use the change log inventory utility, CSQJU003, to update the information in the BSDS about the corrupted data set as follows:
  - a. Use the DELETE function to remove information about the corrupted data set.
  - b. Use the NEWLOG function to name the new data set as the new active log data set and to give it the RBA range that was successfully copied.

The DELETE and NEWLOG functions can be run in the same job step. Put the DELETE statement before NEWLOG statement in the SYSIN input data set.

7. Restart the queue manager.

## Active log is becoming full

The active log can fill up for several reasons, for example, delays in offloading and excessive logging. If an active log runs out of space, this has serious consequences. When the active log becomes full, the queue

manager halts processing until an offload process has been completed. If the offload processing stops when the active log is full, the queue manager can end abnormally. Corrective action is required before the queue manager can be restarted.

### Symptoms

Because of the serious implications of an active log becoming full, the queue manager issues the following warning message when the last available active log data set is 5% full:

```
CSQJ110E +CSQ1 LAST COPYn ACTIVE LOG DATA SET IS nnn PERCENT FULL
```

and reissues the message after each additional 5% of the data set space is filled. Each time the message is issued, the offload process is started.

### System action

Messages are issued and offload processing started. If the active log becomes full, further actions are taken. See [“Active log is full”](#) on page 480

### System programmer action

Use the DEFINE LOG command to dynamically add further active log data sets. This permits IBM MQ to continue its normal operation while the error causing the offload problems is corrected. For more information about the DEFINE LOG command, see [DEFINE LOG](#).

## Active log is full

### Symptoms

When the active log becomes full, the queue manager halts processing until an offload process has been completed. If the offload processing stops when the active log is full, the queue manager can end abnormally. Corrective action is required before the queue manager can be restarted.

IBM MQ issues the following [CSQJ111A](#) message:

```
CSQJ111A +CSQ1 OUT OF SPACE IN ACTIVE LOG DATA SETS
```

and an offload process is started. The queue manager then halts processing until the offload process has been completed.

### System action

IBM MQ waits for an available active log data set before resuming normal IBM MQ processing. Normal shut down, with either QUIESCE or FORCE, is not possible because the shutdown sequence requires log space to record system events related to shut down (for example, checkpoint records). If the offload processing stops when the active log is full, the queue manager stops with an X'6C6' abend; restart in this case requires special attention. For more details, see [“Troubleshooting IBM MQ for z/OS problems”](#) on page 224.

### System programmer action

You can provide additional active log data sets before restarting the queue manager. This permits IBM MQ to continue its normal operation while the error causing the offload process problems is corrected. To add new active log data sets, use the change log inventory utility (CSQJU003) when the queue manager is not active. For more details about adding new active log data sets, see [Changing the BSDS](#).

Consider increasing the number of logs by:

1. Making sure that the queue manager is stopped, then using the Access Method Services DEFINE command to define a new active log data set.
2. Defining the new active log data set in the BSDS, using the change log inventory utility (CSQJU003).

3. Adding additional log data sets dynamically, using the [DEFINE LOG](#) command.

When you restart the queue manager, offloading starts automatically during startup, and any work that was in progress when IBM MQ was forced to stop is recovered.

#### Operator action

Check whether the offload process is waiting for a tape drive. If it is, mount the tape. If you cannot mount the tape, force IBM MQ to stop by using the z/OS CANCEL command.

## Archive log problems

Use this topic to investigate, and resolve problems with the archive logs.

This topic covers the following archive log problems:

- [“Allocation problems” on page 481](#)
- [“Offload task terminated abnormally” on page 481](#)
- [“Insufficient DASD space to complete offload processing” on page 482](#)
- [“Read I/O errors on the archive data set while IBM MQ is restarting” on page 483](#)

## Allocation problems

### Symptoms

IBM MQ issues message: CSQJ103E

```
CSQJ103E +CSQ1 LOG ALLOCATION ERROR DSNAME=dsname,  
        ERROR STATUS=eeeeiii, SMS REASON CODE=sss
```

z/OS dynamic allocation provides the ERROR STATUS. If the allocation was for offload processing, the following message is also displayed: [CSQJ115E](#):

```
CSQJ115E +CSQ1 OFFLOAD FAILED, COULD NOT ALLOCATE AN ARCHIVE  
        DATA SET
```

### System action

The following actions take place:

- If the input is needed for recovery, and recovery is not successful, and the queue manager ends abnormally.
- If the active log had become full and an offload task was scheduled but not completed, the offload task tries again the next time it is triggered. The active log does not reuse a data set that has not yet been archived.

### System programmer action

None.

### Operator action

Check the allocation error code for the cause of the problem, and correct it. Ensure that drives are available, and either restart or wait for the offload task to be retried. Be careful if a DFP/DFSMS ACS user-exit filter has been written for an archive log data set, because this can cause a device allocation error when the queue manager tries to read the archive log data set.

## Offload task terminated abnormally

## Symptoms

No specific IBM MQ message is issued for write I/O errors.

Only a z/OS error recovery program message appears. If you get IBM MQ message [CSQJ128E](#), the offload task has ended abnormally.

## System action

The following actions take place:

- The offload task abandons the output data set; no entry is made in the BSDS.
- The offload task dynamically allocates a new archive and restarts offloading from the point at which it was previously triggered.
- If an error occurs on the new data set:
  - In dual archive mode, message [CSQJ114I](#) is generated and the offload processing changes to single mode:

```
CSQJ114I +CSQ1 ERROR ON ARCHIVE DATA SET, OFFLOAD
          CONTINUING WITH ONLY ONE ARCHIVE DATA SET BEING
          GENERATED
```

- In single archive mode, the output data set is abandoned. Another attempt to process this RBA range is made the next time offload processing is triggered.
- The active log does not wrap around; if there are no more active logs, data is not lost.

## System programmer action

None.

## Operator action

Ensure that offload task is allocated on a reliable drive and control unit.

## Insufficient DASD space to complete offload processing

### Symptoms

While offloading the active log data sets to DASD, the process terminates unexpectedly. IBM MQ issues message [CSQJ128E](#):

```
CSQJ128E +CSQ1 LOG OFF-LOAD TASK FAILED FOR ACTIVE LOG nnnnn
```

The error is preceded by z/OS messages [IEC030I](#), [IEC031I](#), or [IEC032I](#).

### System action

IBM MQ de-allocates the data set on which the error occurred. If IBM MQ is running in dual archive mode, IBM MQ changes to single archive mode and continues the offload task. If the offload task cannot be completed in single archive mode, the active log data sets cannot be offloaded, and the state of the active log data sets remains NOT REUSABLE. Another attempt to process the RBA range of the abandoned active log data sets is made the next time the offload task is triggered.

### System programmer action

The most likely causes of these symptoms are:

- The size of the archive log data set is too small to contain the data from the active log data sets during offload processing. All the secondary space allocations have been used. This condition is normally accompanied by z/OS message [IEC030I](#). The return code in this message might provide further explanations for the cause of these symptoms.

To solve the problem

1. Issue the command `CANCEL queue_manager name` to cancel the queue manager job
2. Increase the primary or secondary allocations (or both) for the archive log data set (in the CSQ6ARVP system parameters).

If the data to be offloaded is large, you can mount another online storage volume or make one available to IBM MQ.

3. Restart the queue manager.

- All available space on the DASD volumes to which the archive data set is being written has been exhausted. This condition is normally accompanied by z/OS message IEC032I.

To solve the problem, make more space available on the DASD volumes, or make another online storage volume available for IBM MQ.

- The primary space allocation for the archive log data set (as specified in the CSQ6ARVP system parameters) is too large to allocate to any available online DASD device. This condition is normally accompanied by z/OS message IEC032I.

To solve the problem, make more space available on the DASD volumes, or make another online storage volume available for IBM MQ. If this is not possible, you must adjust the value of `PRIQTY` in the CSQ6ARVP system parameters to reduce the primary allocation. (For details, see [Using CSQ6ARVP](#).)

**Note:** If you reduce the primary allocation, you might have to increase the size of the secondary space allocation to avoid future abends.

#### Operator action

None.

## Read I/O errors on the archive data set while IBM MQ is restarting

#### Symptoms

No specific IBM MQ message is issued; only the z/OS error recovery program message appears.

#### System action

This depends on whether a second copy exists:

- If a second copy exists, it is allocated and used.
- If a second copy does not exist, restart is not successful.

#### System programmer action

None.

#### Operator action

Try to restart, using a different drive.

## **BSDS problems**

Use this topic to investigate, and resolve problems with BSDS.

For background information about the bootstrap data set (BSDS), see the [Planning your IBM MQ environment on z/OS](#).

This topic describes the following BSDS problems:

- [“Error occurs while opening the BSDS” on page 484](#)
- [“Log content does not agree with the BSDS information” on page 484](#)
- [“Both copies of the BSDS are damaged” on page 485](#)
- [“Unequal time stamps” on page 485](#)
- [“Out of synchronization” on page 486](#)
- [“I/O error” on page 487](#)



- “Log range problems” on page 487

Normally, there are two copies of the BSDS, but if one is damaged, IBM MQ immediately changes to single BSDS mode. However, the damaged copy of the BSDS must be recovered before restart. If you are in single mode and damage the only copy of the BSDS, or if you are in dual mode and damage both copies, use the procedure described in [Recovering the BSDS](#).

This section covers some of the BSDS problems that can occur at startup. Problems not covered here include:

- RECOVER BSDS command errors (messages CSQJ301E - CSQJ307I)
- Change log inventory utility errors (message CSQJ123E)
- Errors in the BSDS backup being dumped by offload processing (message CSQJ125E)

## Error occurs while opening the BSDS

### Symptoms

IBM MQ issues the following message:

```
CSQJ100E +CSQ1 ERROR OPENING BSDSn DSNAME=..., ERROR STATUS=eeii
```

where *eeii* is the VSAM return code. For information about VSAM codes, see the *DFSMS/MVS Macro Instructions for Data Sets* documentation.

### System action

During system initialization, the startup is terminated.

During a RECOVER BSDS command, the system continues in single BSDS mode.

### System programmer action

None.

### Operator action

Carry out these steps:

1. Run the print log map utility on both copies of the BSDS, and compare the lists to determine which copy is accurate or current.
2. Rename the data set that had the problem, and define a replacement for it.
3. Copy the accurate data set to the replacement data set, using Access Method Services.
4. Restart the queue manager.

## Log content does not agree with the BSDS information

### Symptoms

IBM MQ issues the following message:

```
CSQJ102E +CSQ1 LOG RBA CONTENT OF LOG DATA SET DSNAME=...,  
STARTRBA=..., ENDRBA=...,  
DOES NOT AGREE WITH BSDS INFORMATION
```

This message indicates that the change log inventory utility was used incorrectly or that a down-level data set is being used.

**System action**

Queue manager startup processing is terminated.

**System programmer action**

None.

**Operator action**

Run the print log map utility and the change log inventory utility to print and correct the contents of the BSDS.

**Both copies of the BSDS are damaged****Symptoms**

IBM MQ issues the following messages:

```
CSQJ107E +CSQ1 READ ERROR ON BSDS
          DSNAMES=... ERROR STATUS=0874
CSQJ117E +CSQ1 REG8 INITIALIZATION ERROR READING BSDS
          DSNAMES=... ERROR STATUS=0874
CSQJ119E +CSQ1 BOOTSTRAP ACCESS INITIALIZATION PROCESSING FAILED
```

**System action**

Queue manager startup processing is terminated.

**System programmer action**

Carry out these steps:

1. Rename the data set, and define a replacement for it.
2. Locate the BSDS associated with the most recent archive log data set, and copy it to the replacement data set.
3. Use the print log map utility to print the contents of the replacement BSDS.
4. Use the print log records utility to print a summary report of the active log data sets missing from the replacement BSDS, and to establish the RBA range.
5. Use the change log inventory utility to update the missing active log data set inventory in the replacement BSDS.
6. If dual BSDS data sets had been in use, copy the updated BSDS to the second copy of the BSDS.
7. Restart the queue manager.

**Operator action**

None.

**Unequal time stamps****Symptoms**

IBM MQ issues the following message:

```
CSQJ120E +CSQ1 DUAL BSDS DATA SETS HAVE UNEQUAL TIME STAMPS,
          SYSTEM BSDS1=...,BSDS2=...,
          UTILITY BSDS1=...,BSDS2=...
```

The possible causes are:

- One copy of the BSDS has been restored. All information about the restored BSDS is down-level. The down-level BSDS has the earlier time stamp.

- One of the volumes containing the BSDS has been restored. All information about the restored volume is down-level. If the volume contains any active log data sets or IBM MQ data, they are also down-level. The down-level volume has the earlier time stamp.
- Dual logging has degraded to single logging, and you are trying to start without recovering the damaged log.
- The queue manager terminated abnormally after updating one copy of the BSDS but before updating the second copy.

#### **System action**

IBM MQ attempts to resynchronize the BSDS data sets using the more recent copy. If this fails, queue manager startup is terminated.

#### **System programmer action**

None.

#### **Operator action**

If automatic resynchronization fails, carry out these steps:

1. Run the print log map utility on both copies of the BSDS, compare the lists to determine which copy is accurate or current.
2. Rename the down-level data set and define a replacement for it.
3. Copy the good data set to the replacement data set, using Access Method Services.
4. If applicable, determine whether the volume containing the down-level BSDS has been restored. If it has been restored, all data on that volume, such as the active log data, is also down-level.

If the restored volume contains active log data and you were using dual active logs on separate volumes, you need to copy the current version of the active log to the down-level log data set. See [Recovering logs](#) for details of how to do this.

## **Out of synchronization**

### **Symptoms**

IBM MQ issues the following message during queue manager initialization:

```
CSQJ122E +CSQ1 DUAL BSDS DATA SETS ARE OUT OF SYNCHRONIZATION
```

The two input copies of the BSDSs have different time stamps, or contain a record that is inconsistent. Differences can exist if operator errors occurred while the change log inventory utility was being used. (For example, the change log inventory utility was only run on one copy.) The change log inventory utility sets a private time stamp in the BSDS control record when it starts, and a close flag when it ends. IBM MQ checks the change log inventory utility time stamps and, if they are different, or they are the same but one close flag is not set, IBM MQ compares the copies of the BSDSs. If the copies are different, message [CSQJ122E](#) is issued.

This message is also issued by the BSDS conversion utility if two input BSDS are specified and a record is found that differs between the two BSDS copies. This situation can arise if the queue manager terminated abnormally prior to the BSDS conversion utility being run.

#### **System action**

Queue manager startup or the utility is terminated.

#### **System programmer action**

None.

#### **Operator action**

If the error occurred during queue manager initialization, carry out these steps:

1. Run the print log map utility on both copies of the BSDS, and compare the lists to determine which copy is accurate or current.
2. Rename the data set that had the problem, and define a replacement for it.
3. Copy the accurate data set to the replacement data set, using access method services.
4. Restart the queue manager.

If the error occurred when running the BSDS conversion utility, carry out these steps:

1. Attempt to restart the queue manager and shut it down cleanly before attempting to run the BSDS conversion utility again.
2. If this does not solve the problem, run the print log map utility on both copies of the BSDS, and compare the lists to determine which copy is accurate or current.
3. Change the JCL used to invoke the BSDS conversion utility to specify the current BSDS in the SYSUT1 DD statement, and remove the SYSUT2 DD statement, before submitting the job again.

## I/O error

### Symptoms

IBM MQ changes to single BSDS mode and issues the user message:

```
CSQJ126E +CSQ1 BSDS ERROR FORCED SINGLE BSDS MODE
```

This is followed by one of the following messages:

```
CSQJ107E +CSQ1 READ ERROR ON BSDS
          DSNAME=... ERROR STATUS=...

CSQJ108E +CSQ1 WRITE ERROR ON BSDS
          DSNAME=... ERROR STATUS=...
```

### System action

The BSDS mode changes from dual to single.

If the error code for message [CSQJ107E](#) or [CSQJ108E](#) is unexpected, a diagnostic dump might be taken.

### System programmer action

None.

### Operator action

Carry out these steps:

1. Use Access Method Services to rename or delete the damaged BSDS and to define a new BSDS with the same name as the BSDS that had the error. Example control statements can be found in job CSQ4BREC in thlqual.SCSQPROC.
2. Issue the IBM MQ command RECOVER BSDS to make a copy of the good BSDS in the newly allocated data set and reinstate dual BSDS mode. See also [Recovering the BSDS](#).

## Log range problems

### Symptoms

IBM MQ has issued message [CSQJ113E](#) when reading its own log, or message [CSQJ133E](#) or [CSQJ134E](#) when reading the log of a queue manager in the queue sharing group. This can happen when you do not have the archive logs needed to restart the queue manager or recover a CF structure.

## System action

Depending upon what log record is being read and why, the requestor might end abnormally with a reason code of X'00D1032A'.

## System programmer action

Run the print log map utility ([CSQJU004](#)) to determine the cause of the error. When message CSQJ133E or CSQJ134E has been issued, run the utility against the BSDS of the queue manager indicated in the message.

If you have:

- Deleted the entry with the log range (containing the log RBA or LRSN indicated in the message) from the BSDS, and
- Not deleted or reused the data set

you can add the entry back into the BSDS using the following procedure:

1. Identify the data set containing the required RBA or LRSN, by looking at an old copy of the contents of BSDS, or by running [CSQJU004](#) against a backup of the BSDS.
2. Add the data set back into the BSDS using the change log inventory utility ([CSQJU003](#)).
3. Restart the queue manager.

If an archive log data set has been deleted, you will not be able to recover the page set or CF structure that needs the archive logs. Identify the reason that the queue manager needs to read the log record, then take one of the following actions depending on the page set or CF structure affected.

## Page sets

Message CSQJ113E during the recovery phase of queue manager restart indicates that the log is needed to perform media recovery to bring a page set up to date.

Identify the page sets that need the deleted log data set for media recovery, by looking at the media recovery RBA in the [CSQI1049I](#) message issued for each page set during queue manager restart, then perform the following actions.

### • Page set zero

You can recover the objects on page set zero, by using the following procedure.



**Attention:** All data in all other page sets will be lost when you carry out the procedure.

1. Use function SDEFS of the [CSQUTIL](#) utility to produce a file of IBM MQ DEFINE commands.
2. Format page set zero using [CSQUTIL](#), then redefine the other page sets as described in the next section.
3. Restart the queue manager.
4. Use [CSQUTIL](#) to redefine the objects using the DEFINE commands produced by the utility in step 1.

### • Page sets 1-99

Use the following procedure to redefine the page sets.



**Attention:** Any data on the page set is lost when you carry out this operation.

1. If you can access the page set without any I/O errors, reformat the page set using the [CSQUTIL](#) utility with the command `FORMAT TYPE(NEW)`.
2. If I/O errors occurred when accessing the page set, delete the page set and re-create it.

If you want the page set to be the same size as before, use the command `LISTCAT ENT(dsname) ALLOC` to obtain the existing space allocations, and use these in the z/OS [DEFINE CLUSTER](#) command.

Format the new page set using the [CSQUTIL](#) utility with the command `FORMAT TYPE(NEW)`.

3. Restart the queue manager. You might have to take certain actions, such as resetting channels or resolving indoubt channels.

### CF structures

Messages CSQJ113E, CSQJ133E, or CSQJ134E, during the recovery of a CF structure, indicate that the logs needed to recover the structure are not available on at least one member of the queue sharing group.

Take one of the following actions depending on the structure affected:

#### Application CF structure

Issue the command RECOVER CFSTRUCT(*structure-name*) TYPE(PURGE).

This process empties the structure, so any messages on the structure are lost.

#### CSQSYSAPPL structure

Contact your IBM support center.

#### Administration structure

This structure is rebuilt using log data since the last checkpoint on each queue manager, which should be in active logs.

If you get this error during administration structure recovery, contact your IBM support center as this indicates that the active log is not available.

Once you have recovered the page set or CF structure, perform a backup of the logs, BSDS, page sets, and CF structures.

To prevent this problem from occurring again, increase the:

- Archive log retention (ARCRETN) value to be longer, and
- Increase the frequency of the CF structure backups.

## Recovering a CF structure

Conceptually, the data from the previously backed up CF structure is read from the IBM MQ log; the log is read forwards from the backup and any changes are reapplied to the restored structure.

### About this task

The log range to use is found from the latest backup of each structure to be recovered, to the current time. The log range is identified by log range sequence number (LRSN) values.

A LRSN uses the six most significant digits of a 'store clock value'.

Note that the whole log (back to the time the structure was created) is read, if you have not done a backup of the structure.

### Procedure

1. Check that the logs from each queue manager in the queue sharing group (QSG) are read for records in this LRSN range.  
Note that the logs are read backwards.
2. Check that a list of changes for each structure to be recovered is built.
3. Data from the coupling facility (CF) structure backup is read and the data is restored.  
For example, if the backup was done on queue manager A, and the recovery is running on queue manager B, queue manager B reads the logs from queue manager A to restore the structure.  
When the start of the backup of the CF structure is read, an internal task is started to take the restored data for the structure and merge it with the changes read from the log.
4. Check that processing continues for each structure being restored.

## Example

In the following example, the command RECOVER CFSTRUCT(APP3) has been issued, and the following messages produced:

```
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.
```

## Notes:

1. Message CSQE132I is also generated as the result of auto recovery being invoked. For example, "CSQE153I: Auto recovery for structure ABCD has been scheduled" where **RECAUTO** has been set to YES.
2. As part of the System Programmer Response, message CSQE112E directs you to check for the RBA range referenced in message CSQE130I. However, there are certain instances where message CSQE130I is not produced; for example, if no backup has ever been taken before, or if the backup is ignored because of the value of its LRSN.

## Page set problems

Use this topic to investigate, and resolve problems with the page sets.

This topic covers the problems that you might encounter with page sets:

- “Page set I/O errors” on [page 490](#) describes what happens if a page set is damaged.
- “Page set full” on [page 491](#) describes what happens if there is not enough space on the page set for any more MQI operations.

## Page set I/O errors

### Problem

A page set has an I/O error.

### Symptoms

This message is issued:

```
CSQP004E +CSQ1 csect-name I/O ERROR STATUS ret-code
PSID psid RBA rba
```

### System action

The queue manager terminates abnormally.



### System programmer action

None.

### Operator action

Repair the I/O error cause.

If none of the page sets are damaged, restart the queue manager. IBM MQ automatically restores the page set to a consistent state from the logs.

If one or more page sets are damaged:

1. Rename the damaged page sets and define replacement page sets.
2. Copy the most recent backup page sets to the replacement page sets.
3. Restart the queue manager. IBM MQ automatically applies any updates that are necessary from the logs.

You cannot restart the queue manager if page set zero is not available. If one of the other page sets is not available, you can comment out the page set DD statement in the queue manager start-up JCL procedure. This lets you defer recovery of the defective page set, enabling other users to continue accessing IBM MQ.

**When you add the page set back to the JCL procedure, system restart reads the log from the point where the page set was removed from the JCL to the end of the log. This procedure might take a long time if a large amount of data has been logged.**

A reason code of MQRC\_PAGASET\_ERROR is returned to any application that tries to access a queue defined on a page set that is not available.

When you have restored the defective page set, restore its associated DD statement and restart the queue manager.

The operator actions described here are only possible if all log data sets are available. If your log data sets are lost or damaged, see [Restarting if you have lost your log data sets](#).

## Page set full

### Problem

There is not enough space on a page set for one of the following:

- MQPUT or MQPUT1 calls to be completed
- Object manipulation commands to be completed (for example, DEFINE QLOCAL)
- MQOPEN calls for dynamic queues to be completed

### Symptoms

The request fails with reason code MQRC\_STORAGE\_MEDIUM\_FULL. The queue manager cannot complete the request because there is not enough space remaining on the page set.

Reason code MQRC\_STORAGE\_MEDIUM\_FULL can occur even when the page set expand attribute is set to EXPAND(USER). Before the reason code MQRC\_STORAGE\_MEDIUM\_FULL is returned to the application code, the queue manager will attempt to expand the page set and retry the API request. On a heavily loaded system it is possible that the expanded storage can be used by other IO operations before the retry of the API. See [Managing page sets](#).

The cause of this problem could be messages accumulating on a transmission queue because they cannot be sent to another system.

### System action

Further requests that use this page set are blocked until enough messages are removed or objects deleted to make room for the new incoming requests.

### Operator action

Use the IBM MQ command DISPLAY USAGE PSID(\*) to identify which page set is full.

### System programmer action

You can either enlarge the page set involved or reduce the loading on that page set by moving queues to another page set. See [Managing page sets](#) for more information about these tasks. If the cause of the problem is messages accumulating on the transmission queue, consider starting distributed queuing to transmit the messages.

## Coupling facility and Db2 problems

Use this topic to investigate, and resolve problems with the coupling facility, and Db2.

This section covers the problems that you might encounter with the coupling facility and Db2:

- [“Storage medium full” on page 492](#)
- [“A Db2 system fails” on page 492](#)
- [“A Db2 data-sharing group fails” on page 493](#)
- [“Db2 and the coupling facility fail” on page 494](#)

### Storage medium full

#### Problem

A coupling facility structure is full.

#### Symptoms

If a queue structure becomes full, return code MQRC\_STORAGE\_MEDIUM\_FULL is returned to the application.

If the administration structure becomes full, the exact symptoms depend on which processes experience the error, they might range from no responses to CMDSCOPE(GROUP) commands, to queue manager failure as a result of problems during commit processing.

#### System programmer action

You can use IBM MQ to inhibit MQPUT operations to some of the queues in the structure to prevent applications from writing more messages, start more applications to get messages from the queues, or quiesce some of the applications that are putting messages to the queue.

Alternatively you can use XES facilities to alter the structure size in place. The following z/OS command alters the size of the structure:

```
SETXCF START,ALTER,STRNAME= structure-name,SIZE= newsiz
```

where *newsiz* is a value that is less than the value of MAXSIZE specified on the CFRM policy for the structure, but greater than the current coupling facility size.

You can monitor the utilization of a coupling facility structure with the DISPLAY CFSTATUS command.

### A Db2 system fails

If a Db2 subsystem that IBM MQ is connected to fails, IBM MQ attempts to reconnect to the subsystem, and continue working. If you specified a Db2 group attach name in the QSGDATA parameter of the CSQ6SYSP system parameter module, IBM MQ reconnects to another active Db2 that is a member of the same data-sharing group as the failed Db2, if one is available on the same z/OS image.

There are some queue manager operations that do not work while IBM MQ is not connected to Db2. These are:

- Deleting a shared queue or group object definition.

- Altering, or issuing MQSET on, a shared queue or group object definition. The restriction of MQSET on shared queues means that operations such as triggering or the generation of performance events do not work correctly.
- Defining new shared queues or group objects.
- Displaying shared queues or group objects.
- Starting, stopping, or other actions for shared channels.
- Reading the shared queue definition from Db2 the first time that the shared queue is open by issuing an MQOPEN.

Other IBM MQ API operations continue to function as normal for shared queues, and all IBM MQ operations can be performed against the queue manager private versions (COPY objects) built from GROUP objects. Similarly, any shared channels that are running continue normally until they end or have an error, when they go into retry state.

When IBM MQ reconnects to Db2, resynchronization is performed between the queue manager and Db2. This involves notifying the queue manager of new objects that have been defined in Db2 while it was disconnected (other queue managers might have been able to continue working as normal on other z/OS images through other Db2 subsystems), and updating object attributes of shared queues that have changed in Db2. Any shared channels in retry state are recovered.

If a Db2 fails, it might have owned locks on Db2 resources at the time of failure. In some cases, this might make certain IBM MQ objects unavailable to other queue managers that are not otherwise affected. To resolve this, restart the failed Db2 so that it can perform recovery processing and release the locks.

## A Db2 data-sharing group fails

If an entire Db2 data-sharing group fails, recovery might be to the time of failure, or to a previous point in time.

In the case of recovery to the point of failure, IBM MQ reconnects when Db2 has been recovered, the resynchronization process takes places, and normal queue manager function is resumed.

However, if Db2 is recovered to a previous point in time, there might be inconsistencies between the actual queues in the coupling facility structures and the Db2 view of those queues. For example, at the point in time Db2 is recovered to, a queue existed that has since been deleted and its location in the coupling facility structure reused by the definition of a new queue that now contains messages.

If you find yourself in this situation, you must stop all the queue managers in the queue sharing group, clear out the coupling facility structures, and restart the queue managers. You must then use IBM MQ commands to define any missing objects. To do this, use the following procedure:

1. Prevent IBM MQ from reconnecting to Db2 by starting Db2 in utility mode, or by altering security profiles.
2. If you have any important messages on shared queues, you might be able to offload them using the COPY function of the CSQUTIL utility program, but this might not work.
3. Terminate all queue managers.
4. Use the following z/OS command to clear all structures:

```
SETXCF FORCE,STRUCTURE,STRNAME=
```

5. Restore Db2 to a historical point in time.
6. Reestablish queue manager access to Db2.
7. Restart the queue managers.
8. Recover the IBM MQ definitions from backup copies.
9. Reload any offloaded messages to the shared queues.

When the queue managers restart, they attempt to resynchronize local COPY objects with the Db2 GROUP objects. This might cause IBM MQ to attempt to do the following:

- Create COPY objects for old GROUP objects that existed at the point in time Db2 has recovered to.
- Delete COPY objects for GROUP objects that were created since the point in time Db2 has recovered to and so do not exist in the database.

The DELETE of COPY objects is attempted with the NOPURGE option, so it fails for queue managers that still have messages on these COPY queues.

## Db2 and the coupling facility fail

If the coupling facility fails, the queue manager might fail, and Db2 will also fail if it is using this coupling facility.

Recover Db2 using Db2 recovery procedures. When Db2 has been restarted, you can restart the queue managers. The CF administration structure will also have failed, but this is rebuilt by restarting all the queue managers within the queue sharing group.

If a single application structure within the coupling facility suffers a failure, the effect on the queue manager depends on the level of the queue manager and the CFLEVEL of the failed CF structure:

- If the CF application structure is CFLEVEL(3) or higher and RECOVER is set to YES, it will not be usable until you recover the CF structure by issuing an MQSC RECOVER CFSTRUCT command to the queue manager that will do the recovery. You can specify a single CF structure to be recovered, or you can recover several CF structures simultaneously. The queue manager performing the recovery locates the relevant backups on all the other queue managers' logs using the data in Db2 and the bootstrap data sets. The queue manager replays these backups in the correct time sequence across the queue sharing group, from just before the last backup through to the point of failure. If a recoverable application structure has failed, any further application activity is prevented until the structure has been recovered. If the administration structure has also failed, all the queue managers in the queue sharing group must be started before the RECOVER CFSTRUCT command can be issued. All queue managers can continue working with local queues and queues in other CF structures during recovery of a failed CF structure.
- If the CF application structure is CFLEVEL(3) or higher and RECOVER is set to NO, the structure is automatically reallocated by the next MQOPEN request performed on a queue defined in the structure. All messages are lost, as the structure can only contain non-persistent messages.
- If the CF application structure has a CFLEVEL less than 3, the queue manager fails. On queue manager restart, peer recovery attempts to connect to the structure, detect that the structure has failed and allocate a new version of the structure. All messages on shared queues that were in CF structures affected by the coupling facility failure are lost.

Since IBM WebSphere MQ 7.1, queue managers in queue sharing groups have been able to tolerate loss of connectivity to coupling facility structures without failing. If the structure has experienced a connection failure, attempts are made to rebuild the structure in another coupling facility with better connectivity in order to regain access to shared queues as soon as possible.

## Problems with long-running units of work

Use this topic to investigate, and resolve problems with long-running units of work.

This topic explains what to do if you encounter a long-running unit of work during restart. In this context, this means a unit of work that has been active for a long time (possibly days or even weeks) so that the origin RBA of the unit of work is outside the scope of the current active logs. This means that restart could take a long time, because all the log records relating to the unit of work have to be read, which might involve reading archive logs.

## Old unit of work found during restart

### Problem

A unit of work with an origin RBA that predates the oldest active log has been detected during restart.

### Symptoms

IBM MQ issues the following message:

```
CSQR020I +CSQ1 OLD UOW FOUND
```

### System action

Information about the unit of work is displayed, and message CSQR021D is issued, requesting a response from the operator.

### System programmer action

None.

### Operator action

Decide whether to commit the unit of work or not. If you choose not to commit the unit of work, it is handled by normal restart recovery processing. Because the unit of work is old, this is likely to involve using the archive log, and so takes longer to complete.

## IMS-related problems

Use this topic to investigate, and resolve problems with IMS and IBM MQ.

This topic includes plans for the following problems that you might encounter in the IMS environment:

- [“IMS cannot connect to IBM MQ” on page 495](#)
- [“IMS application problem” on page 496](#)
- [“IMS is not operational” on page 496](#)

## IMS cannot connect to IBM MQ

### Problem

The IMS adapter cannot connect to IBM MQ.

### Symptoms

IMS remains operative. The IMS adapter issues these messages for control region connect:

- CSQQ001I
- CSQQ002E
- CSQQ003E
- CSQQ004E
- CSQQ005E
- CSQQ007E

For details, see the [IBM MQ for z/OS 消息, 完成和原因码](#) documentation.

If an IMS application program tries to access IBM MQ while the IMS adapter cannot connect, it can either receive a completion code and reason code, or terminate abnormally. This depends on the value of the REO option in the SSM member of IMS PROCLIB.

### System action

All connection errors are also reported in the IMS message DFS3611.

### System programmer action

None.

## Operator action

Analyze and correct the problem, then restart the connection with the IMS command:

```
/START SUBSYS subsysname
```

IMS requests the adapter to resolve in-doubt units of recovery.

## IMS application problem

### Problem

An IMS application terminates abnormally.

### Symptoms

The following message is sent to the user's terminal:

```
DFS555I TRANSACTION tran-id ABEND abcode  
MSG IN PROCESS: message data:
```

where *tran-id* represents any IMS transaction that is terminating abnormally and *abcode* is the abend code.

### System action

IMS requests the adapter to resolve the unit of recovery. IMS remains connected to IBM MQ.

### System programmer action

None.

### Operator action

As indicated in message DFS554A on the IMS master terminal.

## IMS is not operational

### Problem

IMS is not operational.

### Symptoms

More than one symptom is possible:

- IMS waits or loops

IBM MQ cannot detect a wait or loop in IMS, so you must find the origin of the wait or loop. This can be IMS, IMS applications, or the IMS adapter.

- IMS terminates abnormally.
  - See the manuals *IMS/ESA® Messages and Codes* and *IMS/ESA Failure Analysis Structure Tables* for more information.
  - If threads are connected to IBM MQ when IMS terminates, IBM MQ issues message CSQ3201E. This message indicates that IBM MQ end-of-task (EOT) routines have been run to clean up and disconnect any connected threads.

### System action

IBM MQ detects the IMS error and:

- Backs out in-flight work.
- Saves in-doubt units of recovery to be resolved when IMS is reconnected.

### System programmer action

None.

### Operator action

Resolve and correct the problem that caused IMS to terminate abnormally, then carry out an emergency restart of IMS. The emergency restart:

- Backs out in-flight transactions that changed IMS resources.
- Remembers the transactions with access to IBM MQ that might be in doubt.

You might need to restart the connection to IBM MQ with the IMS command:

```
/START SUBSYS subsysname
```

During startup, IMS requests the adapter to resolve in-doubt units of recovery.

## **Hardware problems**

Use this topic as a starting point to investigate hardware problems.

If a hardware error causes data to be unreadable, IBM MQ can still be recovered by using the *media recovery* technique:

1. To recover the data, you need a backup copy of the data. Use DFDSS or Access Method Services REPRO regularly to make a copy of your data.
2. Reinstate the most recent backup copy.
3. Restart the queue manager.

The more recent your backup copy, the more quickly your subsystem can be made available again.

When the queue manager restarts, it uses the archive logs to reinstate changes made since the backup copy was taken. You must keep sufficient archive logs to enable IBM MQ to reinstate the changes fully. Do not delete archive logs until there is a backup copy that includes all the changes in the log.





# 声明

本信息是为在美国国内供应的产品和服务而编写的。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

**本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区:** International Business Machines Corporation “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗示的）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 可以随时对本资料中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation  
软件互操作性协调员，部门 49XA  
北纬 3605 号公路  
罗切斯特，明尼苏达州 55901  
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含日常商业运作所使用的数据和报表的示例。为了尽可能全面地说明这些数据和报表，这些示例包括个人、公司、品牌和产品的名称。所有这些名字都是虚构的，若现实生活中实际业务企业使用的名字和地址与此相似，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或默示这些程序的可靠性、可维护性或功能。

如果您正在查看本信息的软拷贝，图片和彩色图例可能无法显示。

## 编程接口信息

---

编程接口信息 (如果提供) 旨在帮助您创建用于此程序的应用软件。

本书包含有关允许客户编写程序以获取 IBM MQ 服务的预期编程接口的信息。

但是，该信息还可能包含诊断、修改和调优信息。提供诊断、修改和调优信息是为了帮助您调试您的应用程序软件。

**要点:** 请勿将此诊断，修改和调整信息用作编程接口，因为它可能会发生更改。

## 商标

---

IBM IBM 徽标 ibm.com 是 IBM Corporation 在全球许多管辖区域的商标。当前的 IBM 商标列表可从 Web 上的“Copyright and trademark information”[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) 获取。其他产品和服务名称可能是 IBM 或其他公司的商标。

Microsoft 和 Windows 是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

Linux 是 Linus Torvalds 在美国和/或其他国家或地区的商标。

此产品包含由 Eclipse 项目 (<https://www.eclipse.org/>) 开发的软件。

Java 和所有基于 Java 的商标和徽标是 Oracle 和/或其附属公司的商标或注册商标。





部件号:

(1P) P/N: