

8.0

Managed File Transfer



注

在使用本资料及其支持的产品之前，请阅读第 1045 页的『声明』中的信息。

本版本适用于 IBM® MQ V 8 发行版 0 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, 2023.

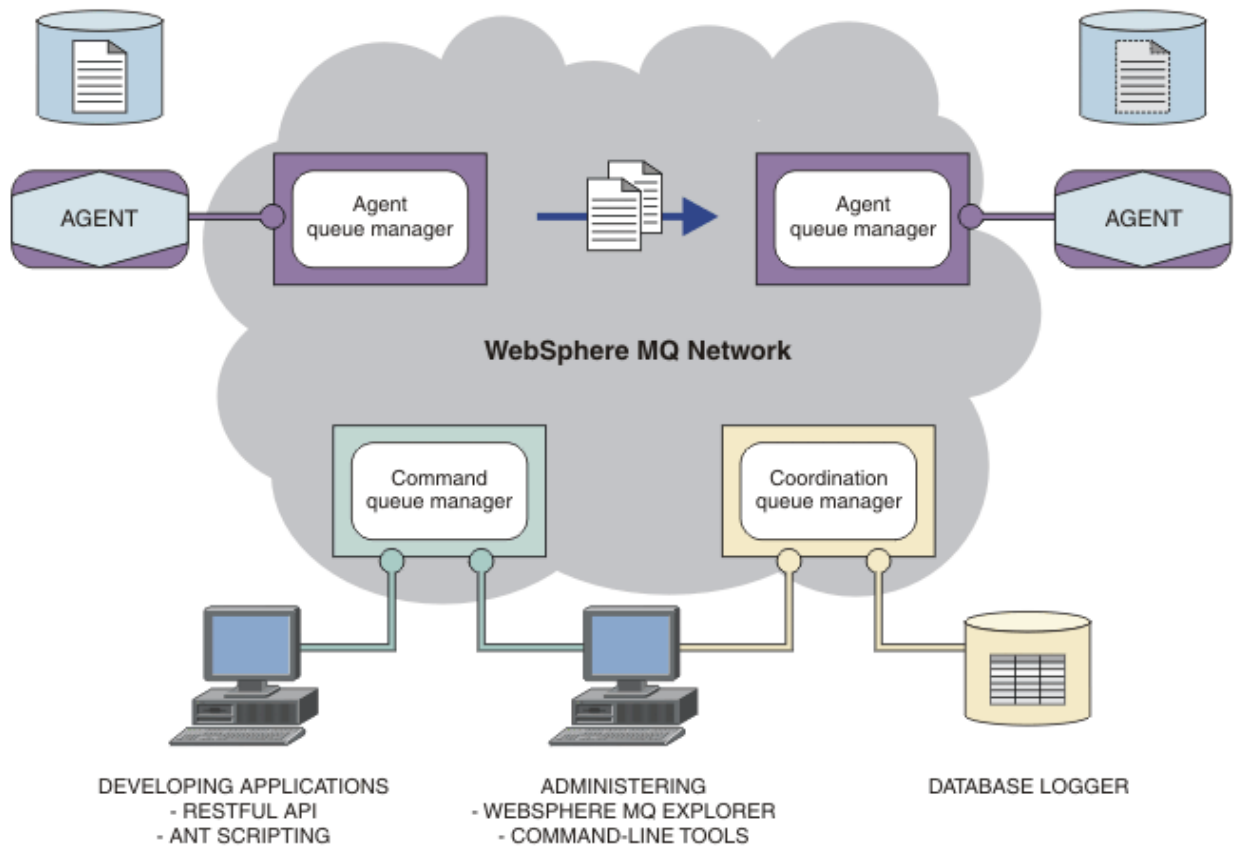
IBM MQ Managed File Transfer 介绍.....	5
产品概述.....	7
IBM MQ Managed File Transfer 介绍.....	7
IBM MQ Managed File Transfer 产品选项.....	9
IBM MQ Managed File Transfer 拓扑概述.....	14
V8.0.0.0 中的新增功能和变更功能.....	15
场景概述.....	15
常见拓扑.....	15
配置基本服务器.....	18
安装 IBM MQ Managed File Transfer.....	25
WebSphere MQ File Transfer Edition V7.0.4 或更低版本与 IBM WebSphere MQ V7.5 或更高版本之间的更改.....	26
在零售环境中使用 IBM MQ Managed File Transfer.....	38
零售环境中的场景.....	38
准备在 IBM 4690 系统上安装 IBM MQ Managed File Transfer.....	60
卸载 4690 系统中的 IBM MQ Managed File Transfer.....	65
在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置.....	66
在 4690 OS 配置捆绑软件中定制代理名称.....	67
在 4690 OS 配置束中定制代理属性.....	69
IBM 4690 系统的配置捆绑软件样本.....	76
在主/备份 4690 OS 控制器设置中配置 IBM MQ Managed File Transfer.....	77
在 4690 OS 控制器设置中配置多个 IBM MQ Managed File Transfer 代理.....	78
在 4690 OS 系统上启动代理.....	78
在 4690 OS 系统上运行时的限制.....	81
文件分发属性.....	82
在 IBM 4690 系统上的沙箱中工作.....	84
零售环境中使用的 IBM MQ Managed File Transfer 命令的摘要.....	84
对 IBM 4690 系统进行故障诊断.....	95
IBM MQ Managed File Transfer 的安全性概述.....	96
IBM MQ Managed File Transfer 和 IBM MQ 连接认证.....	97
沙箱.....	99
为 IBM MQ Managed File Transfer 配置 SSL 或 TLS 加密.....	103
以客户机方式使用通道认证来连接至 WebSphere MQ V7.1 或更高版本队列管理器.....	104
将 IBM MQ Advanced Message Security 与 IBM MQ Managed File Transfer 一起使用.....	105
保证 Web 网关的安全.....	106
配置 Connect:Direct 网桥代理和 Connect:Direct 节点之间的 SSL 或 TLS 连接.....	113
配置 IBM MQ Managed File Transfer.....	115
分布式平台上的配置选项.....	116
z/OS 上的配置选项.....	118
创建 IBM MQ Managed File Transfer 代理或记录器命令数据集.....	118
针对 IBM MQ Managed File Transfer for z/OS 的配置任务.....	119
在 IBM i 系统上安装 IBM MQ Managed File Transfer 后进行配置.....	140
为首次使用配置 IBM MQ Managed File Transfer.....	141
配置 Managed File Transfer 记录器.....	151
配置 Web 网关.....	181
配置 Connect:Direct 网桥.....	204
通过 MSCS 配置 IBM MQ Managed File Transfer 代理.....	213
管理 IBM MQ Managed File Transfer.....	214
启动 IBM MQ Managed File Transfer 代理.....	215
启动新的文件传输.....	221
创建调度的文件传输.....	224
从 IBM MQ Explorer 处理暂挂传输.....	225

触发文件传输.....	225
从 IBM MQ Explorer 监视正在进行的文件传输.....	226
使用传输日志来查看文件传输的状态.....	228
资源监控.....	229
使用传输模板.....	248
将数据从文件传输至消息.....	250
将数据从消息传输到文件.....	263
列出 IBM MQ Managed File Transfer 代理.....	273
停止 IBM MQ Managed File Transfer 代理.....	273
协议网桥.....	275
Connect:Direct 网桥.....	288
使用 IBM Integration Bus.....	302
IBM MQ Managed File Transfer 的恢复和重新启动.....	302
开发应用程序.....	303
指定要运行的程序.....	303
IBM MQ Managed File Transfer Web 网关.....	304
将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用.....	354
使用用户出口例程定制 IBM MQ Managed File Transfer.....	358
通过将消息放置在代理命令队列中来控制 IBM MQ Managed File Transfer.....	370
IBM MQ Managed File Transfer 故障诊断.....	371
一般故障诊断.....	371
对 Web 网关进行故障诊断.....	412
对 Connect:Direct 网桥进行故障诊断.....	423
参考.....	428
产品概述.....	428
安装.....	430
安全性.....	431
MFT 命令和进程与队列管理器的连接关系.....	445
IBM MQ Managed File Transfer 命令汇总.....	447
配置.....	580
管理.....	700
开发应用程序.....	906
MFT 诊断消息.....	1044
声明.....	1045
编程接口信息.....	1046
商标.....	1046

IBM MQ Managed File Transfer 介绍

IBM MQ Managed File Transfer 将以受管和可审计方式在系统之间传输文件，而不用考虑文件大小或使用的操作系统。

您可以使用 IBM MQ Managed File Transfer 构建定制、可扩展且自动化的解决方案，使您能够管理和信任文件传输并保证文件传输的安全。IBM MQ Managed File Transfer 可消除昂贵的冗余、降低维护成本并最大限度地利用现有的 IT 投资。



该图显示了简单的 IBM MQ Managed File Transfer 拓扑。其中包含两个代理，分别与它们在 IBM MQ 网络中的代理队列管理器相连。通过 IBM MQ 网络，将文件从位于图中一侧的代理传输到位于图中另一侧的代理。在 IBM MQ 网络中还包含一个协调队列管理器和一个命令队列管理器。应用程序和工具连接到这些队列管理器，以在 IBM MQ 网络中配置，管理，操作和记录 IBM MQ Managed File Transfer 活动。

IBM MQ Managed File Transfer 可以作为四个不同的选项进行安装。这些选项为 IBM MQ Managed File Transfer 代理程序、IBM MQ Managed File Transfer 记录器、IBM MQ Managed File Transfer 服务或 IBM MQ Managed File Transfer 工具。有关更多信息，请参阅第 9 页的『IBM MQ Managed File Transfer 产品选项』。

可以使用 IBM MQ Managed File Transfer 来执行下列任务：

- 创建受管文件传输
 - 在 Linux® 或 Windows 平台上从 IBM MQ 资源管理器 创建新文件传输。
 - 通过所有受支持平台上的命令行创建新的文件传输。
 - 将文件传输功能集成到 Apache Ant 工具中。
 - 通过将消息放置在代理命令队列上，编写控制 IBM MQ Managed File Transfer 的应用程序。

- 安排稍后进行文件传输。还可以根据一系列文件系统事件（例如，正在创建新文件）触发调度的文件传输。
- 持续监视资源（例如，目录），并在该资源的内容满足某个预定义条件时启动任务。该任务可以是文件传输、Ant 脚本或 JCL 作业。
- 使用 IBM MQ Managed File Transfer Web 网关提供的 RESTful API 传输文件。
- 与 IBM MQ 队列进行文件传输。
- 在 FTP、FTPS 或 SFTP 服务器之间进行文件传输。
- 将文件传输至 Connect:Direct 节点和从这些节点传输文件。
- 传输文本和二进制文件。将在源和目标系统的代码页和行结束约定之间自动转换文本文件。
- 可以使用针对基于安全套接字层 (SSL) 连接的行业标准保证传输安全。
- 查看正在进行的传输并记录有关网络中所有传输的信息。
 - 在 Linux 或 Windows 平台上查看正在从 IBM MQ 资源管理器 进行的传输的状态。
 - 在 Linux 或 Windows 平台上使用 IBM MQ 资源管理器 检查已完成传输的状态。
 - 使用 IBM MQ Managed File Transfer 数据库记录器功能将日志消息保存到 Db2 或 Oracle 数据库。
 - 使用 IBM MQ Managed File Transfer Web 网关提供的 RESTful API 查看有关网络中所有传输的信息。

IBM MQ Managed File Transfer 基于 IBM MQ 构建，WebSphere MQ 在应用程序之间提供确定的仅一次性消息传递。您可以利用 IBM MQ 的各种功能。例如，您可以使用通道压缩来压缩通过 IBM MQ 通道在代理之间发送的数据，或使用 SSL 通道保证代理之间所发送数据的安全。文件可靠地传输，并且可以容忍通过其执行文件传输的基础结构的故障。如果经历网络中断，文件传输将在连接恢复时从断开的位置重新启动。

通过将文件传输与现有的 IBM MQ 网络进行整合，可以避免耗用维护两个独立的基础结构所需的资源。如果您还不是 IBM MQ 客户，请创建 IBM MQ 网络以支持 IBM MQ Managed File Transfer，从而为未来的 SOA 实施构建主干。如果您已是 IBM MQ 客户，那么 IBM MQ Managed File Transfer 可以利用现有的 IBM MQ 基础结构，包括 IBM MQ 因特网通路和 IBM Integration Bus。

IBM MQ Managed File Transfer 集成了一些其他 IBM 产品：

IBM MQ Advanced Message Security

使用 IBM MQ Advanced Message Security 为 IBM MQ Managed File Transfer 中的消息流量（特别是对队列上的数据）提供增强的安全性。有关更多信息，请参阅第 105 页的『[将 IBM MQ Advanced Message Security 与 IBM MQ Managed File Transfer 一起使用](#)』。

IBM Integration Bus

作为 IBM Integration Bus 流的一部分，通过 IBM MQ Managed File Transfer 传输的进程文件。有关更多信息，请参阅第 302 页的『[使用 IBM Integration Bus](#)』。

IBM Sterling Connect:Direct

使用 IBM MQ Managed File Transfer Connect:Direct 网桥与现有的 Connect:Direct 网络进行文件传输。有关更多信息，请参阅第 288 页的『[Connect:Direct 网桥](#)』。

IBM Tivoli 组合应用程序管理器

IBM Tivoli 组合应用程序管理器 提供可用于监视发布到协调队列管理器的信息的代理。

相关概念

第 9 页的『[IBM MQ Managed File Transfer 产品选项](#)』

根据您的操作系统和总体设置，IBM MQ Managed File Transfer 可以作为四个不同的选项进行安装。这些选项为 IBM MQ Managed File Transfer 代理程序、IBM MQ Managed File Transfer 记录器、IBM MQ Managed File Transfer 服务或 IBM MQ Managed File Transfer 工具。

第 14 页的『[IBM MQ Managed File Transfer 拓扑概述](#)』

相关参考

第 428 页的『[IBM MQ Managed File Transfer 如何工作？](#)』

IBM MQ Managed File Transfer 可以使用多种方式与 IBM MQ 进行交互。该主题描述了两个产品是如何进行交互的。

产品概述

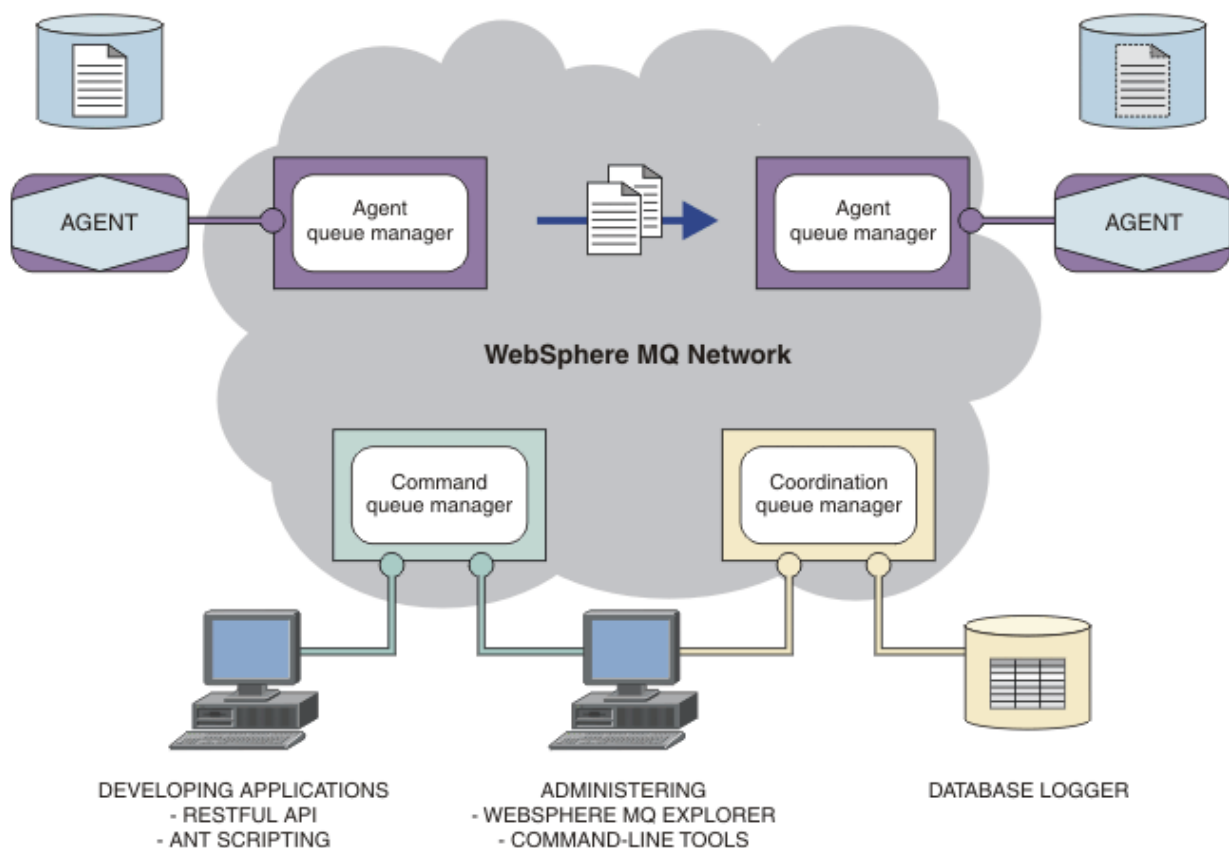
本部分提供了您用于开始使用 IBM MQ Managed File Transfer 的介绍信息。

- [第 5 页的『IBM MQ Managed File Transfer 介绍』](#)
- [第 9 页的『IBM MQ Managed File Transfer 产品选项』](#)
- [第 14 页的『IBM MQ Managed File Transfer 拓扑概述』](#)
- [第 15 页的『场景概述』](#)
- [第 15 页的『V8.0.0.0 中的新增功能和变更功能』](#)

IBM MQ Managed File Transfer 介绍

IBM MQ Managed File Transfer 将以受管和可审计方式在系统之间传输文件，而不用考虑文件大小或使用的操作系统。

您可以使用 IBM MQ Managed File Transfer 构建定制、可扩展且自动化的解决方案，使您能够管理和信任文件传输并保证文件传输的安全。IBM MQ Managed File Transfer 可消除昂贵的冗余、降低维护成本并最大限度地利用现有的 IT 投资。



该图显示了简单的 IBM MQ Managed File Transfer 拓扑。其中包含两个代理，分别与它们在 IBM MQ 网络中的代理队列管理器相连。通过 IBM MQ 网络，将文件从位于图中一侧的代理传输到位于图中另一侧的代理。在 IBM MQ 网络中还包含一个协调队列管理器和一个命令队列管理器。应用程序和工具连接到这些队列管理器，以在 IBM MQ 网络中配置，管理，操作和记录 IBM MQ Managed File Transfer 活动。

IBM MQ Managed File Transfer 可以作为四个不同的选项进行安装。这些选项为 IBM MQ Managed File Transfer 代理程序、IBM MQ Managed File Transfer 记录器、IBM MQ Managed File Transfer 服务或 IBM MQ Managed File Transfer 工具。有关更多信息，请参阅 [第 9 页的『IBM MQ Managed File Transfer 产品选项』](#)。

可以使用 IBM MQ Managed File Transfer 来执行下列任务：

- 创建受管文件传输
 - 在 Linux 或 Windows 平台上从 IBM MQ 资源管理器 创建新文件传输。
 - 通过所有受支持平台上的命令行创建新的文件传输。
 - 将文件传输功能集成到 Apache Ant 工具中。
 - 通过将消息放置在代理命令队列上，编写控制 IBM MQ Managed File Transfer 的应用程序。
 - 安排稍后进行文件传输。还可以根据一系列文件系统事件（例如，正在创建新文件）触发调度的文件传输。
 - 持续监视资源（例如，目录），并在该资源的内容满足某个预定义条件时启动任务。该任务可以是文件传输、Ant 脚本或 JCL 作业。
 - 使用 IBM MQ Managed File Transfer Web 网关提供的 RESTful API 传输文件。
 - 与 IBM MQ 队列进行文件传输。
 - 在 FTP、FTPS 或 SFTP 服务器之间进行文件传输。
 - 将文件传输至 Connect:Direct 节点和从这些节点传输文件。
 - 传输文本和二进制文件。将在源和目标系统的代码页和行结束约定之间自动转换文本文件。
 - 可以使用针对基于安全套接字层 (SSL) 连接的行业标准保证传输安全。
- 查看正在进行的传输并记录有关网络中所有传输的信息。
 - 在 Linux 或 Windows 平台上查看正在从 IBM MQ 资源管理器 进行的传输的状态。
 - 在 Linux 或 Windows 平台上使用 IBM MQ 资源管理器 检查已完成传输的状态。
 - 使用 IBM MQ Managed File Transfer 数据库记录器功能将日志消息保存到 Db2 或 Oracle 数据库。
 - 使用 IBM MQ Managed File Transfer Web 网关提供的 RESTful API 查看有关网络中所有传输的信息。

IBM MQ Managed File Transfer 基于 IBM MQ 构建，WebSphere MQ 在应用程序之间提供确定的仅一次性消息传递。您可以利用 IBM MQ 的各种功能。例如，您可以使用通道压缩来压缩通过 IBM MQ 通道在代理之间发送的数据，或使用 SSL 通道保证代理之间所发送数据的安全。文件可靠地传输，并且可以容忍通过其执行文件传输的基础结构的故障。如果经历网络中断，文件传输将在连接恢复时从断开的位置重新启动。

通过将文件传输与现有的 IBM MQ 网络进行整合，可以避免耗用维护两个独立的基础结构所需的资源。如果您还不是 IBM MQ 客户，请创建 IBM MQ 网络以支持 IBM MQ Managed File Transfer，从而为未来的 SOA 实施构建主干。如果您已是 IBM MQ 客户，那么 IBM MQ Managed File Transfer 可以利用现有的 IBM MQ 基础结构，包括 IBM MQ 因特网通路和 IBM Integration Bus。

IBM MQ Managed File Transfer 集成了一些其他 IBM 产品：

IBM MQ Advanced Message Security

使用 IBM MQ Advanced Message Security 为 IBM MQ Managed File Transfer 中的消息流量（特别是对队列上的数据）提供增强的安全性。有关更多信息，请参阅第 105 页的『[将 IBM MQ Advanced Message Security 与 IBM MQ Managed File Transfer 一起使用](#)』。

IBM Integration Bus

作为 IBM Integration Bus 流的一部分，通过 IBM MQ Managed File Transfer 传输的进程文件。有关更多信息，请参阅第 302 页的『[使用 IBM Integration Bus](#)』。

IBM Sterling Connect:Direct

使用 IBM MQ Managed File Transfer Connect:Direct 网桥与现有的 Connect:Direct 网络进行文件传输。有关更多信息，请参阅第 288 页的『[Connect:Direct 网桥](#)』。

IBM Tivoli 组合应用程序管理器

IBM Tivoli 组合应用程序管理器 提供可用于监视发布到协调队列管理器的信息的代理。

相关概念

第 9 页的『[IBM MQ Managed File Transfer 产品选项](#)』

根据您的操作系统和总体设置，IBM MQ Managed File Transfer 可以作为四个不同的选项进行安装。这些选项为 IBM MQ Managed File Transfer 代理程序、IBM MQ Managed File Transfer 记录器、IBM MQ Managed File Transfer 服务或 IBM MQ Managed File Transfer 工具。

[第 14 页的『IBM MQ Managed File Transfer 拓扑概述』](#)

相关参考

[第 428 页的『IBM MQ Managed File Transfer 如何工作？』](#)

IBM MQ Managed File Transfer 可以使用多种方式与 IBM MQ 进行交互。该主题描述了两个产品是如何进行交互的。

IBM MQ Managed File Transfer 产品选项

根据您的操作系统和总体设置，IBM MQ Managed File Transfer 可以作为四个不同的选项进行安装。这些选项为 IBM MQ Managed File Transfer 代理程序、IBM MQ Managed File Transfer 记录器、IBM MQ Managed File Transfer 服务或 IBM MQ Managed File Transfer 工具。

IBM MQ Managed File Transfer 代理程序

IBM MQ Managed File Transfer 代理程序安装选项用于安装一个文件传输代理。文件传输代理会连接到 IBM MQ 队列管理器，并将文件数据作为消息传输至其他文件传输代理。它们必须作为 IBM MQ Managed File Transfer 代理程序或 IBM MQ Managed File Transfer 服务安装选项的一部分进行安装。

IBM MQ Managed File Transfer 代理程序安装选项可以安装在没有 IBM MQ 服务器安装选项的系统上。仅当 IBM MQ Managed File Transfer 代理程序安装是安装在已安装 IBM MQ 服务器安装选项的系统上时，在 IBM MQ Managed File Transfer 代理程序安装过程中安装的文件传输代理的某些功能才可用。例如，执行协议网桥配置和操作的功能。

IBM MQ Managed File Transfer 记录器

IBM MQ Managed File Transfer 记录器安装选项用于安装一个文件传输记录器。文件传输记录器会连接到 IBM MQ 队列管理器（通常指定为协调队列管理器），并且将与文件传输审计相关的数据记录到数据库或文件中。

IBM MQ Managed File Transfer 记录器 安装选项必须和 IBM MQ 服务器安装选项安装在同一系统上。

IBM MQ Managed File Transfer 服务

IBM MQ Managed File Transfer 服务安装选项用于安装一个文件传输代理，此代理具有通过 IBM MQ Managed File Transfer 代理程序安装选项安装的文件传输代理所不包含的额外功能。这些额外功能有：

- 创建用于通过传统的 FTP、FTPS 或 SFTP 服务器发送和接收文件的协议网桥代理
- 部署 Web 网关功能部件，用于提供 RESTful 接口，以构建旨在传输文件的绑定 Web 应用程序

IBM MQ Managed File Transfer 服务 安装选项必须和 IBM MQ 服务器安装选项安装在同一系统上。

IBM MQ Managed File Transfer 工具

IBM MQ Managed File Transfer 工具安装选项安装用于和文件传输代理交互的命令行工具。这些工具使您可以通过命令行启动文件传输，安排文件传输和创建资源监视器。

在安装和使用 IBM MQ Managed File Transfer 工具安装选项的系统上，对于是否要安装文件传输代理并无要求。

在 UNIX 平台上，有一个额外的 IBM MQ Managed File Transfer 基本 安装组件。此组件包含对所有安装选项公用的文件。您必须首先安装 IBM MQ Managed File Transfer 基本组件，然后再安装任何代理程序、记录器、服务或工具组件。

有关 UNIX 平台上每个产品选项所需的 IBM MQ 组件的更多信息，请参阅以下主题：

- [第 10 页的『HP-UX 系统上每个 IBM MQ Managed File Transfer 产品选项的必需组件』](#)
- [第 11 页的『Linux 系统上每个 IBM MQ Managed File Transfer 产品选项所需的组件』](#)
- [第 12 页的『Solaris 系统上每个 IBM MQ Managed File Transfer 产品选项的必需组件』](#)
- [第 13 页的『AIX 系统上每个 IBM MQ Managed File Transfer 产品选项的必需组件』](#)

服务和代理程序选项提供的功能

IBM MQ Managed File Transfer 服务

- 建立到队列管理器的客户机或绑定方式连接。当文件传输代理和队列管理器位于同一个系统上时，我们建议您使用绑定方式连接。
- 将文件传输至其他 IBM MQ Managed File Transfer 代理和从这些代理传输文件。
- 与 SFTP、FTP 或 FTPS 协议服务器进行相互文件传输。
- 将文件传输至 Connect:Direct 节点和从这些节点传输文件。
- 通过 Web 网关从 HTTP 客户机传输文件。

某些功能仅在一部分受支持平台上可用。有关更多信息，请参阅“IBM MQ 系统需求”。

IBM MQ Managed File Transfer 代理程序

- 建立到队列管理器的客户机或绑定方式连接。当文件传输代理和队列管理器位于同一个系统上时，我们建议您使用绑定方式连接。
- 将文件传输至其他 IBM MQ Managed File Transfer 代理和从这些代理传输文件。
- 将文件传输至 Connect:Direct 节点和从这些节点传输文件。

在没有队列管理器的情况下使用 MQMFT 代理程序

在您将拥有没有队列管理器的主机，但您希望拥有 MQMFT 代理程序的情况下，您可能希望使用 IBM MQ Client Version 8.0 安装 SupportPac MQC8。

请注意，该 SupportPac 不包含 MQMFT 代码。

因此，要在仅安装了 IBM MQ V8.0 客户机的主机中安装 MFT 代理，您需要从下载了 IBM MQ 服务器文件集的主机复制以下文件集。

例如，在 AIX 上：

- mqm.ft.agent
- mqm.ft.base

在复制这些文件集之后，继续以 root 用户身份安装这些文件集。

相关概念

第 5 页的『IBM MQ Managed File Transfer 介绍』

IBM MQ Managed File Transfer 将以受管和可审计方式在系统之间传输文件，而不用考虑文件大小或使用的操作系统。

第 14 页的『IBM MQ Managed File Transfer 拓扑概述』

HP-UX 系统上每个 IBM MQ Managed File Transfer 产品选项的必需组件

根据您的操作系统和总体设置，IBM MQ Managed File Transfer 可以作为四个不同的选项进行安装。在 HP-UX 系统上，这些选项是 IBM MQ Managed File Transfer 代理、IBM MQ Managed File Transfer 记录器、IBM MQ Managed File Transfer 服务和 IBM MQ Managed File Transfer 工具，每个选项均需要特定组件。

IBM MQ Managed File Transfer 代理

MQSERIES.MQM-RUNTIME

MQSERIES.MQM-JAVA

MQSERIES.MQM-JAVAJRE

MQSERIES.MQM-FTBASE

MQSERIES.MQM-FTAGENT

IBM MQ Managed File Transfer 记录器

MQSERIES.MQM-RUNTIME
MQSERIES.MQM-SERVER
MQSERIES.MQM-JAVA
MQSERIES.MQM-JAVAJRE
MQSERIES.MQM-FTBASE
MQSERIES.MQM-FTLOGGER

IBM MQ Managed File Transfer 服务

MQSERIES.MQM-RUNTIME
MQSERIES.MQM-SERVER
MQSERIES.MQM-JAVA
MQSERIES.MQM-JAVAJRE
MQSERIES.MQM-FTBASE
MQSERIES.MQM-FTAGENT
MQSERIES.MQM-FTSERVICE

IBM MQ Managed File Transfer 工具

MQSERIES.MQM-RUNTIME
MQSERIES.MQM-JAVA
MQSERIES.MQM-JAVAJRE
MQSERIES.MQM-FTBASE
MQSERIES.MQM-FTTOOLS

Linux 系统上每个 IBM MQ Managed File Transfer 产品选项所需的组件

根据您的操作系统和总体设置，IBM MQ Managed File Transfer 可以作为四个不同的选项进行安装。在 Linux 系统上，这些选项是 IBM MQ Managed File Transfer 代理、IBM MQ Managed File Transfer 记录器、IBM MQ Managed File Transfer 服务和 IBM MQ Managed File Transfer 工具，并且各自需要特定组件。

IBM MQ Managed File Transfer 代理

MQSeriesRuntime
MQSeriesJava
MQSeriesJRE
MQSeriesFTBase
MQSeriesFTAgent

IBM MQ Managed File Transfer 记录器

MQSeriesRuntime
MQSeriesServer
MQSeriesJava
MQSeriesJRE

MQSeriesFTBase

MQSeriesFTLogger

IBM MQ Managed File Transfer 服务

MQSeriesRuntime

MQSeriesServer

MQSeriesJava

MQSeriesJRE

MQSeriesFTBase

MQSeriesFTAgent

MQSeriesFTService

IBM MQ Managed File Transfer 工具

MQSeriesRuntime

MQSeriesJava

MQSeriesJRE

MQSeriesFTBase

MQSeriesFTTools

Solaris 系统上每个 IBM MQ Managed File Transfer 产品选项的必需组件

根据您的操作系统和总体设置，IBM MQ Managed File Transfer 可以作为四个不同的选项进行安装。在 Solaris 系统上，这些选项是 IBM MQ Managed File Transfer 代理、IBM MQ Managed File Transfer 记录器、IBM MQ Managed File Transfer 服务和 IBM MQ Managed File Transfer 工具，每个选项均需要特定组件。

IBM MQ Managed File Transfer 代理

运行时

java

jre

ftbase

ftagent

IBM MQ Managed File Transfer 记录器

运行时

服务器

java

jre

ftbase

ftlogger

IBM MQ Managed File Transfer 服务

运行时

服务器

java

jre

ftbase

ftagent

ftservice

IBM MQ Managed File Transfer 工具

运行时

java

jre

ftbase

fttools

AIX 系统上每个 IBM MQ Managed File Transfer 产品选项的必需组件

根据您的操作系统和总体设置，IBM MQ Managed File Transfer 可以作为四个不同的选项进行安装。在 AIX 系统上，这些选项是 IBM MQ Managed File Transfer 代理、IBM MQ Managed File Transfer 记录器、IBM MQ Managed File Transfer 服务和 IBM MQ Managed File Transfer 工具，每个选项均需要特定组件。

IBM MQ Managed File Transfer 代理

mqm.base.runtime

mqm.java.rte

mqm.jre.rte

mqm.ft.base

mqm.ft.agent

IBM MQ Managed File Transfer 记录器

mqm.base.runtime

mqm.server.rte

mqm.java.rte

mqm.jre.rte

mqm.ft.base

mqm.ft.logger

IBM MQ Managed File Transfer 服务

mqm.base.runtime

mqm.server.rte

mqm.java.rte

mqm.jre.rte

mqm.ft.base

mqm.ft.agent

mqm.ft.service

IBM MQ Managed File Transfer 工具

mqm.base.runtime

mqm.java.rte

mqm.jre.rte

mqm.ft.base

mqm.ft.tools

IBM MQ Managed File Transfer 拓扑概述

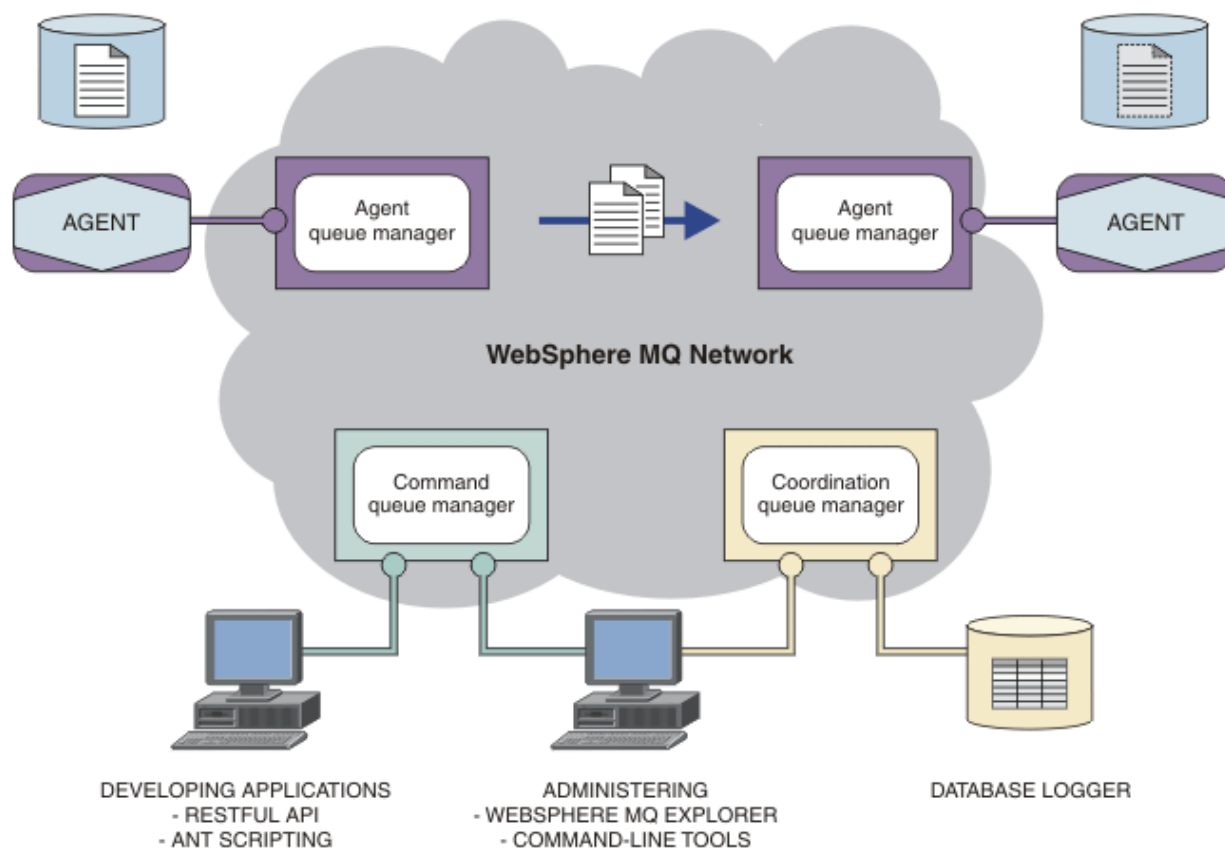
IBM MQ Managed File Transfer 代理发送并接收传输的文件。每个代理在其关联队列管理器上都有自己的队列集，并且代理将以绑定或客户机方式连接到其队列管理器。代理还可以使用协调队列管理器作为其队列管理器。

协调队列管理器广播审计和文件传输信息。协调队列管理器表示代理集合、传输状态和传输审计信息的单个端点。协调队列管理器并非执行传输所必需的。如果协调队列管理器临时不可用，传输将照常继续。审计和状态消息将会存储在代理队列管理器中，直至协调队列管理器可用，然后将照常处理。

代理注册协调队列管理器并将其详细信息发布到队列管理器。此代理信息供 IBM MQ Managed File Transfer 插件用于支持从 IBM MQ Explorer 启动传输。协调队列管理器上收集的代理信息还将供命令用来显示代理信息和代理状态。

传输状态和传输审计信息将在协调队列管理器上发布。传输状态和传输审计信息供 IBM MQ Managed File Transfer 插件用于从 IBM MQ Explorer 监视传输的进度。可以保留存储在协调队列管理器上的传输审计信息以提供可审计性。

命令队列管理器用于连接到 IBM MQ 网络，也是在您发出 IBM MQ Managed File Transfer 命令时要连接到的队列管理器。



相关概念

第 5 页的『[IBM MQ Managed File Transfer 介绍](#)』

IBM MQ Managed File Transfer 将以受管和可审计方式在系统之间传输文件，而不用考虑文件大小或使用的操作系统。

第 15 页的『[场景概述](#)』

此部分列出常见 IBM MQ Managed File Transfer 拓扑以及一个设置系统和传输测试消息的场景。

相关参考

第 428 页的『[IBM MQ Managed File Transfer 如何工作？](#)』

IBM MQ Managed File Transfer 可以使用多种方式与 IBM MQ 进行交互。该主题描述了两个产品是如何进行交互的。

V8.0.0.0 中的新增功能和变更功能

了解 IBM MQ Managed File Transfer V8.0 中的主要新增功能和变更功能。

有关新增功能的信息，请参阅 [IBM MQ V8.0 中的新增功能](#)。

有关变更功能的信息，请参阅 [IBM MQ V8.0 中的变更功能](#)。

场景概述

此部分列出常见 IBM MQ Managed File Transfer 拓扑以及一个设置系统和传输测试消息的场景。

- [常见拓扑](#)
- [配置基本服务器](#)

常见拓扑

本部分列出常见的 IBM MQ Managed File Transfer 拓扑。每张图中的双向箭头表示与队列管理器的连接。

请参阅第 18 页的『连接注意事项』，以获取有关队列管理器连接选项的更多信息。

包含一个队列管理器的基本拓扑

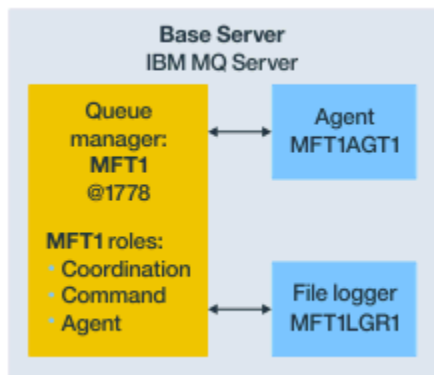


图 1: 包含一个队列管理器的基本拓扑

基本拓扑表示包含协调队列管理器的完整配置。配置名称与协调队列管理器名称相同。如果协调队列管理器名称为 MFT1，那么配置名称为 MFT1。

基本拓扑是您完成的第一个 IBM MQ Managed File Transfer 配置。在完成基本配置后，来自远程服务器的伙伴代理将添加到基本配置以交换文件。

基本拓扑不在基本拓扑服务器之外交换文件。但是，基本拓扑支持您将文件移动相同服务器中的不同位置，并且可用于开发目的。

包含一个伙伴代理的基本拓扑

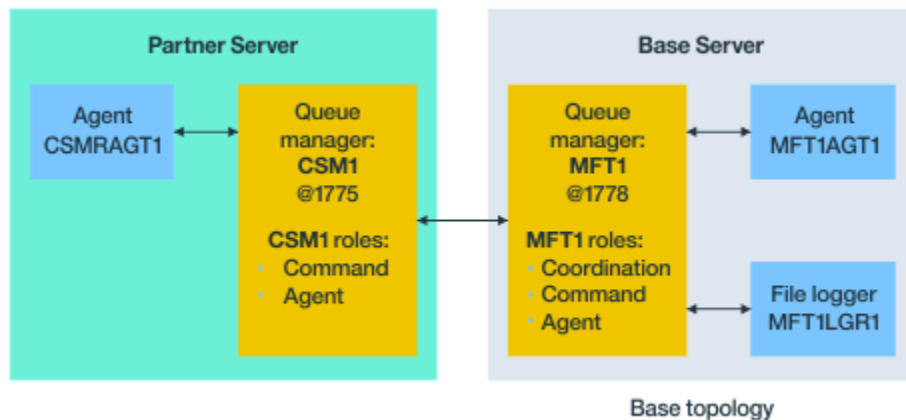


图 2: 包含一个伙伴代理的基本拓扑

此拓扑可在两个代理之间交换文件。可通过与添加的第一个代理类似的方式添加额外的伙伴代理。

您可以将单个队列管理器用于所有三个 IBM MQ Managed File Transfer 队列管理器角色，或者可以将专用队列管理器用于特定角色。

例如，您可以有一个专门用于协调队列管理器角色的队列管理器，并且命令和代理角色可共享另一个队列管理器。

远程代理队列管理器（位于不同于基本配置的服务器中）与基本配置协调队列管理器之间的连接必须配置为 IBM MQ 客户机或 MQI 通道。

与协调队列管理器的连接由 **fteSetupCoordination** 命令建立。如果未将协调队列管理器连接配置为 IBM MQ 客户机通道，那么在从伙伴代理服务器发出诸如 **fteListAgents** 之类的命令时，命令将失败。

具有单独的协调队列管理器和一个伙伴代理的基本拓扑

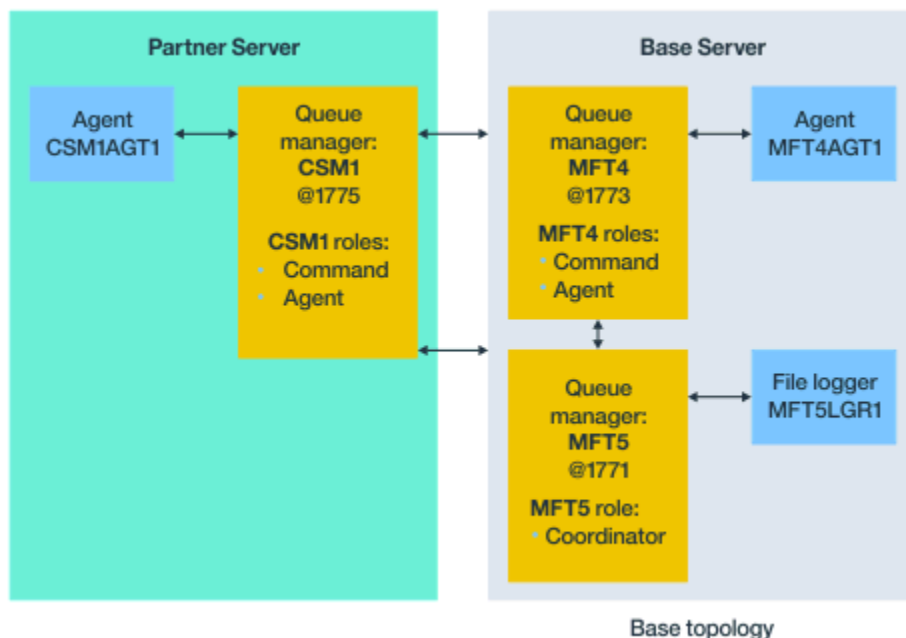


图 3: 具有单独的协调队列管理器和一个伙伴代理的基本拓扑

在图 3 的基本拓扑中，在基本服务器上，队列管理器 MFT4 由命令和代理角色共享，并且队列管理器 MFT5 专门用于协调队列管理器角色。

拓扑中的所有队列管理器（包括基本拓扑中的队列管理器 MFT4 和 MFT5）之间必须存在连接。

在伙伴服务器队列管理器上，队列管理器 CSM1 具有代理和命令队列管理器角色。

此拓扑可在两个代理之间交换文件。每个伙伴代理必须连接到一个队列管理器，如图中所示。可通过与添加的第一个伙伴代理类似的方式，添加额外的伙伴代理。

具有 IBM MQ Managed File Transfer 代理伙伴的基本拓扑

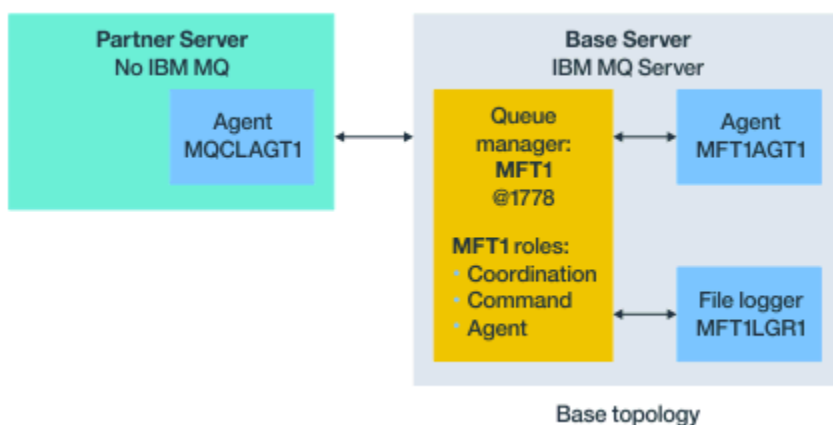


图 4: 具有 IBM MQ Managed File Transfer 代理伙伴的基本拓扑

此拓扑可在两个代理之间交换文件。

伙伴代理中的服务器（在图中描述为 MQCLAGT1）未安装 IBM MQ 服务器。

伙伴代理将使用与 IBM MQ 已安装服务器相同的命令进行配置，但存在一些例外情况：

- 此伙伴代理的配置必须使用与一个或多个基本队列管理器的 IBM MQ 客户机连接。

- 无需运行在伙伴代理服务器中通过配置命令创建的协调队列管理器角色 IBM MQ 定义。基本服务器中已存在协调队列管理器定义。

但是，您必须：

- 复制在伙伴服务器中创建代理时所生成的代理对象定义
- 将定义文件传输到基本配置服务器，以及
- 在标识为基本服务器中的代理队列管理器的队列管理器中创建定义。

在此情况下，MFT1 充当所有三个角色，并且您将在 MFT1 队列管理器中为代理 MQCLAGT1 创建对象。

作为将对象定义复制到基本服务器的替代方法，可以在代理队列管理器所在的基本服务器上针对代理 MQCLAGT1 运行 **fteDefine** 命令。使用 **fteDefine** 命令生成的定义以在代理队列管理器上创建必需的代理定义。

例如，在显示的图中，您会将文件 MQCLAGT1_create.mqsc 从伙伴服务器中的代理目录复制到基本配置服务器，并在 MFT1 队列管理器中创建必需的代理定义。

在伙伴代理服务器上完成的配置将创建 IBM MQ Managed File Transfer 配置目录和必需的属性文件。

连接注意事项

在先前的图中，各个代理和队列管理器之间的每条线都表示与队列管理器的连接。

此连接可以是：

- 本地连接
- 绑定或消息通道连接，或者
- IBM MQ 客户机或 MQI 连接。

配置中选择的连接类型取决于指定的参数。

- 在指定不含其他连接参数的队列管理器名称参数时，那么您指定的是绑定连接。

如果使用的队列管理器位于 IBM MQ Managed File Transfer 配置本地，那么在用于基本配置服务器中时，还表示本地连接。

- 如果指定队列管理器名称参数以及相应的主机、端口和通道名称参数，那么您指定的是 IBM MQ 客户机连接。

代理位于与代理队列管理器相同的主机上时，会产生本地连接的绑定类型规范更高效。

配置基本服务器

如何设置具有一个单独的配置队列管理器的基本服务器。

开始之前

以下示例假定：

- 您已查看第 18 页的『连接注意事项』部分并了解如何影响配置中队列管理器的连接类型。
- 您拥有有效的 IBM MQ 基础结构。请参阅[配置 IBM MQ 队列管理器](#)以获取有关设置队列管理器的信息。
- IBM MQ 安全任务已完成。

所有系统资源（例如，文件访问权）都配置了充足的安全性。

有关 IBM MQ Managed File Transfer 安全配置，请参阅[IBM MQ Managed File Transfer 的安全概述](#)和[有关 IBM MQ Managed File Transfer 操作的用户权限](#)。

- 在通过使用样本程序发送和接收消息或者使用样本 **amqscnxc** 测试 IBM MQ 客户机类型连接来配置 IBM MQ 之后，将测试所有 IBM MQ 连接。

通过在样本代码中定义通道连接，**amqscnxc** 样本连接到队列管理器，这类似于 IBM MQ Managed File Transfer 使用 MQI 或 IBM MQ 客户机类型连接时的连接方式。

- 指示信息假定您用于基本配置的服务器已安装一个 IBM MQ 版本。如果基本服务器中有多个 IBM MQ 安装，那么必须小心地将正确的文件路径用于想要使用的 IBM MQ 版本。
- 在这些指示信息中使用的队列管理器不需要连接认证。

虽然可以更简单地完成第一个配置而无需连接认证，但是如果企业需要立即使用连接认证，请参阅 [IBM MQ Managed File Transfer](#) 和 [IBM MQ 连接认证](#) 以获取有关如何配置 MQMFTCredentials.xml 凭证文件的指示信息。

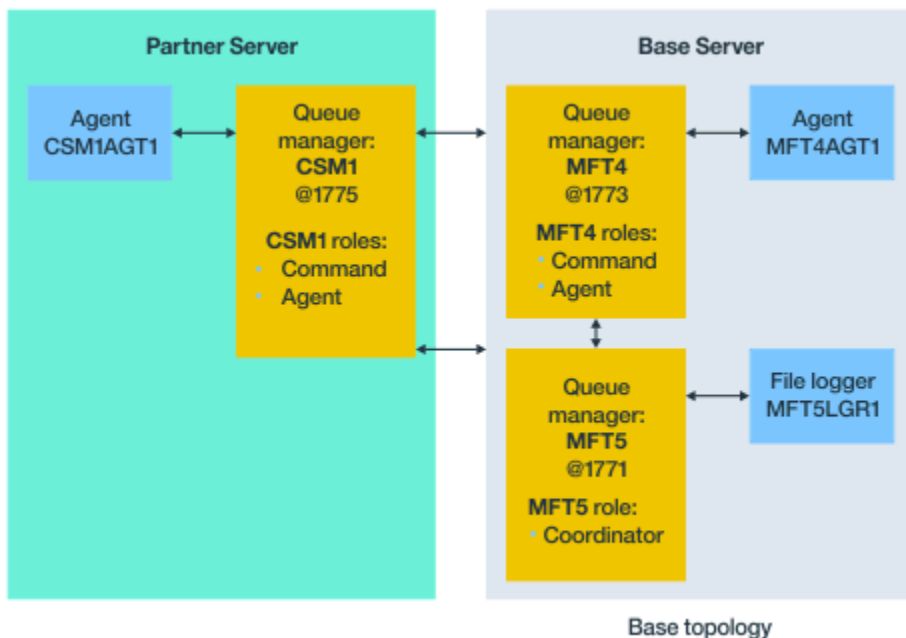


图 5: 具有单独的协调队列管理器和一个伙伴代理的基本拓扑

关于此任务

示例配置的队列管理器角色为：

- 基本服务器
 - 队列管理器 MFT5 是协调队列管理器
 - 队列管理器 MFT4 用作代理 MFT4AGT1 的代理队列管理器，而且还充当基本服务器上 MFT5 配置的命令队列管理器。
- 伙伴服务器
 - 队列管理器 CSM1 同时作为代理 CSM1AGT1 的代理队列管理器和伙伴服务器上 MFT5 配置的命令队列管理器。
 - 基本服务器上的队列管理器 MFT5 是协调队列管理器。

过程

1. [配置协调队列管理器](#)
2. [配置命令队列管理器](#)
3. [设置代理](#)
4. [设置记录器](#)
5. [配置伙伴服务器](#)

下一步做什么

设置具有 MQMFT 的 MQExplorer，以便可测试示例设置。

配置协调队列管理器

如何配置协调队列管理器来协调文件传输。

开始之前

确保在为此场景设置的队列管理器之间具有完整连接。

关于此任务

此任务设置协调队列管理器 MFT5，并且此部分中的指示信息假定您使用的是一个 IBM MQ 安装。

如果您有多个安装，那么在启动任何配置任务之前，必须使用 [setmqenv](#) 命令将 IBM MQ 路径设置为所需的 IBM MQ 版本。

过程

1. 以 IBM MQ 管理员身份登录。
2. 发出以下命令以识别协调队列管理器并设置配置目录结构：

```
fteSetupCoordination -coordinationQMGr MFT5
```

协调队列管理器目录

C:\<data>\mqft\config\MFT5

coordination.properties 文件

C:\<data>\mqft\config\MFT5\coordination.properties

此命令还会生成一个必须针对协调队列管理器运行的 MQSC 命令文件：

C:\<data>\mqft\config\MFT5\MFT5.mqsc。

3. 切换到 C:\<data>\mqft\config\MFT5 目录。
4. 通过运行以下命令，配置队列管理器以充当协调队列管理器。
您需要提供在步骤“[第 20 页的『2』](#)”中发出的命令所生成的 MQSC 命令文件：

```
runmqsc MFT5 < MFT5.mqsc > mft5.txt
```

5. 使用首选编辑器打开 mft5.txt 结果文件。确保已成功创建定义。

下一步做什么

设置命令队列管理器。

配置命令队列管理器

如何配置命令队列管理器。

开始之前

确保您已配置协调队列管理器。请参阅 [第 20 页的『配置协调队列管理器』](#)，以了解更多信息。

关于此任务

此任务标识命令队列管理器。

过程

发出以下命令：

```
fteSetupCommands -connectionQMGr MFT4
```

您将收到以下消息：BFGCL0245I: The file C:\<data>\mqft\config\MFT4\command.properties has been created successfully.

命令队列管理器无需额外的 IBM MQ 定义。运行 **fteSetupCommands** 后，将在 MFT5 配置目录中创建 command.properties 文件。

下一步做什么

设置代理。

设置代理

如何准备文件传输代理 MFT4AGT1，包括必须运行的 MQSC 脚本。

开始之前

您应该已经设置命令队列管理器。请参阅 [第 20 页的『配置命令队列管理器』](#)，以了解更多信息。

关于此任务

此任务准备 Windows 文件传输代理 MFT4AGT1。

过程

1. 发出以下命令：

```
fteCreateAgent -agentName MFT4AGT1 -agentQMgr MFT4
```

在使用 **fteCreateAgent** 命令创建代理后，会将 agents 目录以及代理 MFT4AGT1 的子目录添加到 MFT5 目录。

在 <data>\MFT5\agents\MFT4AGT1 目录中查找：

- agent.properties 文件
 - MFT4AGT1_create.mqsc 文件，其中包含代理所需的 IBM MQ 定义。
2. 切换到 <data>\MFT5\agents\MFT4AGT1 目录，并通过发出以下命令来创建必需的代理队列管理器定义：

```
runmqsc MFT4 < MFT4AGT1_create.mqsc > mft4.txt
```

3. 使用首选编辑器打开 mft4.txt 结果文件并确保已成功创建定义。
4. 通过输入以下命令启动代理：**fteStartAgent** MFT4AGT1。
5. 通过输入以下命令显示代理：**fteListAgents**。

您应该会看到类似于以下内容的输出：

```
5655-MFT, 5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
BFGPR0127W: No credentials file has been specified to connect to IBM MQ.
Therefore, the assumption is that IBM MQ authentication has been disabled.
Agent Name:      Queue Manager Name:      Status:
MFT4AGT1         MFT4                                READY
```

注：如果尚未在 IBM MQ Managed File Transfer 环境中启用连接认证，那么可忽略 BFGPR0127W 消息。

如果发出 **ftelistAgents** 命令并接收到以下消息：BFGCL0014W: 不存在与当前选择标准匹配的代理程序。请参阅[第 378 页的『如果 fteListAgents 命令未列出您的代理，该怎么办』](#)以获取进一步信息。

下一步做什么

设置记录器。

设置记录器

需要文件或数据库记录器以保留与配置的传输活动相关的历史记录和审计信息。在此示例中，您将创建文件记录器。

开始之前

您必须已设置：

- 配置队列管理器
- 命令队列管理器
- 代理程序

过程

1. 发出以下命令：

```
fteCreateLogger -loggerQMgr MFT5 -loggerType FILE  
-fileLoggerMode CIRCULAR -fileSize 5MB -fileCount 3 MFT5lgr1
```

在运行 **fteCreateLogger** 命令后，将创建具有 MFT5LGR1 子目录的 <data>\mqft\config\MFT5\loggers 目录。

MFT5LGR1 子目录保留 logger.properties 文件。另外，在该目录中还有一个名为 MFT5LGR1_create.mqsc 的文件，其中包含记录器所需的 IBM MQ 定义。

2. 切换到目录 <data>\mqft\config\MFT5\loggers\MFT5LGR1。
3. 运行关联的 MQSC 命令文件。

```
runmqsc MFT5 < MFT5_create.mqsc
```

以创建记录器所需的定义。

a) 查看对象定义的结果以确认已成功创建必需的对象。

4. 通过发出以下命令启动记录器：**fteStartLogger** MFT5LGR1。
5. 查看 <data>\mqft\logs\MFT5\loggers\MFT5LGR1\logs 中 output0.log 文件的内容。

在有关记录器的一些信息之后，最后一句应包含以下消息：BFGDB0023I: The logger has completed startup activities and is now running。

在记录器第一次启动时，偶尔可能不会将日志信息写入到 output0.log。如果 output0.log 文件为空，请通过输入 **fteStopLogger** MFT5LGR1 并按 **Enter** 键来重新启动记录器。

通过输入 **fteStartLogger** MFT5LGR1 并按 **Enter** 键来重新启动记录器。现在，文件 output0.log 将显示数据。

在第一次启动代理时，相同的行为将扩展到 output0.log 文件的代理版本。

通过使用 **fteStopAgent** 和 **fteStartAgent** 命令来停止和启动代理。然后，您将看到日志数据写入代理 output0.log 文件。

结果

您已配置基本服务器，其中包含此配置的协调队列管理器。

下一步做什么

现在，您针对包含远程代理的伙伴服务器执行类似工作。

配置伙伴服务器

在基本服务器具有单独的协调队列管理器时如何配置伙伴服务器

开始之前

确保已完全完成设置包含配置队列管理器的基本服务器的所有任务。

关于此任务

针对 IBM MQ 和安全配置以及 IBM MQ 路径的假定也适用于伙伴服务器。

首先设置 MFT5 配置目录并使用 **fteSetupCoordination** 命令来标识协调队列管理器。

过程

1. 通过发出以下命令来创建伙伴服务器配置目录：

```
fteSetupCoordination -coordinationQMGr MFT5  
-coordinationQMGrHost 177.16.20.15 -coordinationQMGrPort 1771  
-coordinationQMGrChannel MQMFT.MFT5.SVRCONN
```

注意：

- a. 在协调队列管理器位于与伙伴服务器不同的服务器上时，必须将与基本服务器协调队列管理器的连接定义为客户端连接。

如果未在伙伴服务器上将协调队列管理器连接定义为 IBM MQ 客户端连接，将导致连接到协调队列管理器的任何 IBM MQ Managed File Transfer 命令失败。

连接到协调队列管理器的命令的示例为 **fteListAgents**。

- b. 您无需创建 IBM MQ 定义，因为在配置基本服务器时已完成协调队列管理器所需的定义。

2. 通过发出以下命令来标识命令队列管理器：

```
fteSetupCommands -connectionQMGr CSM1
```

命令队列管理器不需要任何额外的 IBM MQ 定义。

3. 通过发出以下命令来标识伙伴代理队列管理器并创建伙伴代理队列管理器：

```
fteCreateAgent -agentName CSM1AGT1 -agentQMGr CSM1
```

4. 切换到 CSM1AGT1 目录。

5. 通过发出以下命令来创建代理所需的 IBM MQ 定义：

```
runmqsc CSM1 < CSM1AGT1_create.mqsc > csm1.txt
```

- a) 使用首选编辑器打开文件 csm1.txt 以确认已成功创建代理所需的所有定义。

6. 通过发出以下命令来启动代理：

```
fteStartAgent CSM1AGT1
```

7. 通过输入 **fteListAgents** 来显示代理

您应该会看到类似于以下内容的输出：

```
C:\>fteListAgents  
5655-MFT, 5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED  
BFGPR0127W: No credentials file has been specified to connect to IBM MQ. Therefo  
re, the assumption is that IBM MQ authentication has been disabled.  
Agent Name:      Queue Manager Name:      Status:  
CSM1AGT1         CSM1                     READY  
MFT4AGT1         MFT4                     READY
```

注：如果尚未在 IBM MQ Managed File Transfer 环境中启用连接认证，那么可忽略 BFGPR0127W 消息。

如果发出 **fteListAgents** 命令并接收到以下消息：BFGCL0014W: 不存在与当前选择标准匹配的代理程序。请参阅第 378 页的『如果 fteListAgents 命令未列出您的代理，该怎么办』以获取进一步信息。

如果其中一个代理的状态为 UNREACHABLE，请参阅第 380 页的『[当 fteListAgents 命令显示代理状态为 UNREACHABLE 时应如何处理](#)』以获取更多信息。

使用 MQMFT 设置 MQ Explorer

此任务帮助您将 IBM MQ Explorer 连接到 IBM MQ Managed File Transfer 配置。

过程

1. 启动 MQ Explorer。
2. 在左侧的 "Navigator" 面板中，向下滚动并展开文件夹: Managed File Transfer。
您会看到协调队列管理器的条目：MFT5
3. 右键单击 MFT5 并选择**连接**。
 - a) 选择显示的下拉菜单中的 Agents，并确保 MFT4AGT1 和 CSMAGT1 这两个代理均处于 Ready 状态。

下一步做什么

使用 [MQ Explorer](#) 测试示例设置。

使用 MQ Explorer 来测试文件传输

此任务提供了一个示例，说明如何将 IBM MQ Explorer 与 IBM MQ Managed File Transfer 配合使用，以在设置 MQ Explorer 后测试文件传输，如上一个主题中所述。

开始之前

确保您拥有一个有效的系统，代理处于 READY 状态并且 MQ Explorer 正常工作。请参阅第 24 页的『[使用 MQMFT 设置 MQ Explorer](#)』，以了解更多信息。

关于此任务

确定用于测试传输的文件以及要将其复制到的目录。对于此示例，假定使用 C:\temp\mft 目录中的文件 test-file.txt。

```
C:\temp\mft> dir *
<Date stamp> 61 test-file.txt
1 File(s) 61 bytes
```

过程

1. 在 Windows 中启动 MQ Explorer
2. 在左侧的 "Navigator" 面板中，展开文件夹: Managed File Transfer。
您会看到协调队列管理器的条目：MFT5
3. 右键单击 MFT5 并选择**连接**。
4. 连接后，右键单击 MFT5 并选择**新建传输**
 - a) 使用下拉菜单以针对源代理选择 MFT4AGT1，并针对目标代理选择 CSMAGT1。
 - b) 单击**下一步**。
 - c) 单击下一个窗口上的**添加**。
此时将显示一个宽对话框。左侧用于 Source，右侧用于 Destination。
5. 在 Source 面板上：
 - a) 选择**文本传输**，因为文件是文本。
 - b) 选择**浏览**以查找文件。
在本例中，文件为 C:\temp\mft\test-file.txt。



注意: 请勿单击**确定**，因为您需要完成 Destination 面板。

6. 在 Destination 面板上:
 - a) 输入要为目标上的文件指定的名称，例如，`test-file.txt`。
支持使用相对路径。完整路径的顶部是启动目标代理的用户标识的主目录。
 - b) 如果需要此选项，请选择 `Overwrite files if present`。
 - c) 单击**确定**。
选中的文件将在“**新建传输**”面板中显示。
7. 如果 MFT5 配置菜单已关闭，并且显示 +MFT5，请通过单击 + 号来展开该菜单。
8. 保留在选中的 MQMFT 配置上。
接下来，通过执行以下过程来检查传输的状态。
9. 单击协调队列管理器 MFT5 下面的**传输日志**。
10. 查看紧靠在 **传输日志** 顶部面板下方的 Managed File Transfer - Current Transfer progress panel 中的状态，并等待传输完成。
如果传输显示成功且背景为绿色，那么您已成功完成配置测试。
如果传输失败且背景为红色，那么发生了错误。
在大多数情况下，您可以使用顶部**传输日志**面板下方的滚动条来查看故障原因的摘要。
 - a) 如果无法确定传输失败的原因，请在顶部**传输日志**面板中双击该传输的条目。
 - b) 选择所显示弹出面板的左侧窗格中的 XML。
 - c) 滚动浏览信息以确定错误原因。
 - d) 执行所需的纠正并重新测试传输。

安装 IBM MQ Managed File Transfer

本主题概述了安装 IBM MQ Managed File Transfer 所必须执行的操作。

从 V7.5 或更高版本开始，IBM MQ Managed File Transfer 将作为 IBM MQ 的组件安装在 UNIX 平台和 Windows 上，不再作为单独的产品进行安装。

IBM MQ Managed File Transfer 在 IBM i 和 z/OS 上仍然是单独的产品。

要备份的配置数据

您应备份拓扑涉及的所有队列管理器。

另外，存在您需要与队列管理器数据同时备份的不位于队列管理器内的文本文件。以下是 MQ_DATA_PATH/mqft 文件，其中 MQ_DATA_PATH 位于：

Windows

C:\Program Files (x86)\IBM\WebSphere MQ\mqft

UNIX 平台

/var/mqm/mqft

产品选项

IBM MQ Managed File Transfer 可以作为四个不同的选项进行安装。这些选项为 IBM MQ Managed File Transfer 代理程序、IBM MQ Managed File Transfer 记录器、IBM MQ Managed File Transfer 服务或 IBM MQ Managed File Transfer 工具。

要决定安装哪些组件，请查看以下主题中的产品选项和拓扑信息：

- [第 9 页的『IBM MQ Managed File Transfer 产品选项』](#)
- [第 14 页的『IBM MQ Managed File Transfer 拓扑概述』](#)

如何安装

有关 IBM MQ 在 UNIX 平台和 Windows 上安装的概述，请参阅[安装和卸载](#)。

有关在 IBM i 上安装 IBM MQ Managed File Transfer 的信息，请参阅[在 IBM i 上安装](#)。

有关在 z/OS 上安装 IBM MQ Managed File Transfer 的信息，请参阅[在 z/OS 上安装](#)。

有关您的平台要安装哪些特定 受管文件传输 组件的信息，请参阅[选择安装内容](#)。

注: 您必须先更新数据库记录器实例，再更新 Managed File Transfer 网络的其他部分，以使这些实例能够正确处理它们收到的最新版本的传输日志消息。

相关概念

第 9 页的『[IBM MQ Managed File Transfer 产品选项](#)』

根据您的操作系统和总体设置，IBM MQ Managed File Transfer 可以作为四个不同的选项进行安装。这些选项为 IBM MQ Managed File Transfer 代理程序、IBM MQ Managed File Transfer 记录器、IBM MQ Managed File Transfer 服务或 IBM MQ Managed File Transfer 工具。

第 14 页的『[IBM MQ Managed File Transfer 拓扑概述](#)』

相关参考

第 430 页的『[已安装的命令集](#)』

下表显示了随每个组件一起安装的命令。

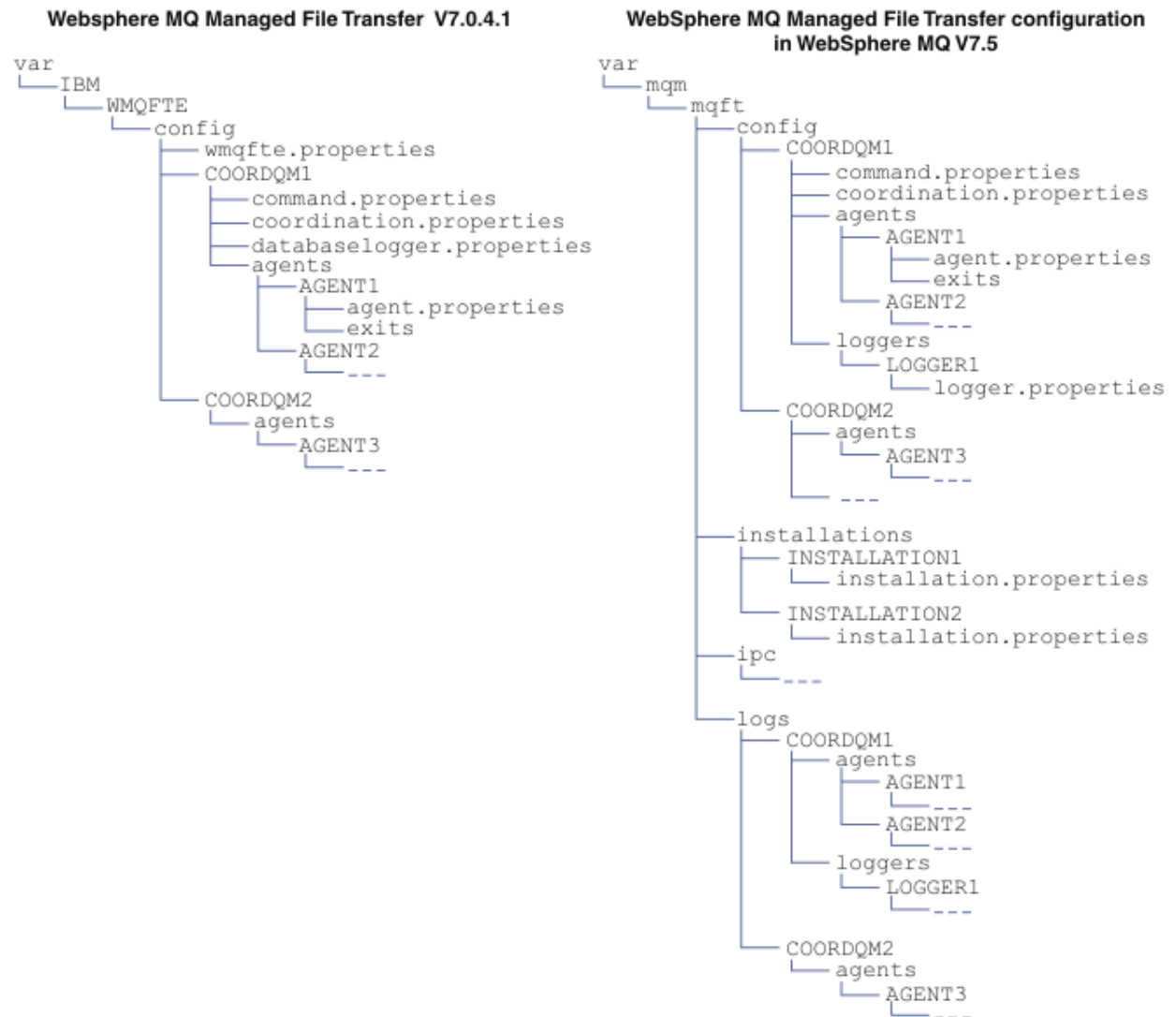
WebSphere MQ File Transfer Edition V7.0.4 或更低版本与 IBM WebSphere MQ V7.5 或更高版本之间的更改

如果计划从 WebSphere MQ File Transfer Edition V7.0.4 或更早版本移至 IBM WebSphere MQ V7.5 或更高版本，请查看以下信息，其中总结了各版本之间的变化。

配置更改

V7.5 或更高版本中刚安装之后的配置布局不同于 WebSphere MQ File Transfer Edition V7.0.4 或更早版本中刚安装之后的配置布局。

例如，该图首先显示了 WebSphere MQ File Transfer Edition V7.0.4.1 中刚安装之后的配置布局，然后显示了 WebSphere MQ V7.5 或更高版本中刚安装之后的配置布局。



WebSphere MQ File Transfer Edition V7.0.4 或更早版本文件名和缺省位置	V7.5 或更高版本的等效文件名和缺省位置
缺省配置目录位置 (wmqfte_configuration_directory): • UNIX 系统: /var/IBM/WMQFTE/config • Linux 系统: /var/ibm/WMQFTE/config • Windows: C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config	缺省配置目录位置和内容: 先前位于 WebSphere MQ File Transfer Edition 配置目录中的信息现在分割到四个独立的子目录中: config、installations、ipc 和 logs。 缺省产品根目录 (MQ_DATA_PATH) 如下所示: • UNIX 系统: /var/mqm • Linux 系统: /var/mqm • Windows: 配置目录的位置取决于主 IBM MQ 安装的位置。主安装的缺省位置如下: – 32 位: C:\Program Files (x86) \WebSphere MQ – 64 位: C:\Program Files\IBM\WebSphere MQ 配置子目录如下所示:

WebSphere MQ File Transfer Edition V7.0.4 或更早版本文件名和缺省位置	V7.5 或更高版本的等效文件名和缺省位置
	• <i>MQ_DATA_PATH/mqft/config</i> 目录包含对 受管文件传输 进程为只读的配置部分。例如, <i>agent.properties</i> 和 <i>command.properties</i>。 • <i>MQ_DATA_PATH/mqft/installations</i> 目录包含每个安装的配置信息。此目录的内容等同于 <i>wmqfte.properties</i> 文件的内容。 • <i>MQ_DATA_PATH/mqft/ipc</i> 目录包含内部用于在 受管文件传输 组件之间进行通信的 IPC 资源。仅适用于 UNIX 和 Linux 系统。 • <i>MQ_DATA_PATH/mqft/logs</i> 目录包含由 受管文件传输 进程写入的配置部分。例如, 跟踪信息和日志文件。
<i>wmqfte.properties</i> <i>wmqfte.properties</i> 文件可设置适用于整个 WebSphere MQ File Transfer Edition 安装的属性。 缺省位置为 <i>wmqfte_configuration_directory</i>	<i>installation.properties</i> <i>installation.properties</i> 文件是与 <i>wmqfte.properties</i> 等效的重命名并重新放置的文件。 在 UNIX 和 Linux 系统上, 缺省位置为 <i>MQ_DATA_PATH/mqft/installations/installation_name</i> 在 Windows 上, 缺省位置是 <i>MQ_DATA_PATH\mqft\installations\installation_name</i>
<i>databaselogger.properties</i> 该文件包含独立数据库记录器的属性信息。 缺省位置为 <i>wmqfte_configuration_directory/coordination_qmgr_name</i>	<i>logger.properties</i> 该文件现在包含独立文件记录器、独立数据库记录器和 JEE 数据库记录器的属性信息。 缺省位置是 <i>MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name</i>。

安全性更改

对于 IBM WebSphere MQ V7.5 或更高版本, 只有作为管理员 (mqm 组的成员) 的用户才能运行以下所列的 **fte** 命令:

- [第 454 页的『fteChangeDefaultConfigurationOptions \(更改缺省配置选项\)』](#)
- [第 460 页的『fteCreateAgent \(创建 IBM MQ Managed File Transfer 代理\)』](#)
- [第 463 页的『fteCreateBridgeAgent \(创建和配置 IBM MQ Managed File Transfer 协议网桥代理\)』](#)
- [第 469 页的『fteCreateCDAgent \(创建 Connect:Direct 网桥代理\)』](#)
- [第 474 页的『fteCreateLogger \(创建 IBM MQ Managed File Transfer 记录器\)』](#)
- [第 522 页的『fteDeleteAgent \(删除 IBM MQ Managed File Transfer 代理\)』](#)
- [第 525 页的『fteDeleteLogger \(删除 IBM MQ Managed File Transfer 记录器\)』](#)
- [第 540 页的『fteMigrate 代理 \(将 WebSphere MQ File Transfer Edition V7.0 代理迁移到 IBM WebSphere MQ V7.5 或更高版本\)』](#)
- [第 542 页的『fteMigrateConfigurationOptions \(将 WebSphere MQ File Transfer Edition V7.0 配置迁移至 IBM WebSphere MQ V7.5 或更高版本\)』](#)
- [第 544 页的『fteMigrateLogger \(将 WebSphere MQ File Transfer Edition V7.0 数据库记录器迁移至 IBM WebSphere MQ V7.5 或更高版本\)』](#)
- [第 546 页的『fteModifyAgent \(修改 IBM MQ Managed File Transfer 代理\)』](#)
- [第 548 页的『fteModifyLogger \(作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序\)』](#)

- [第 560 页的『fteSetupCommands（创建 command.properties 文件）』](#)
- [第 562 页的『fteSetupCoordination（设置协调详细信息）』](#)

对于 IBM WebSphere MQ V7.5 或更高版本，只有运行代理进程的用户才可以运行 [第 374 页的『fteSetAgentTraceLevel（设置 IBM MQ Managed File Transfer 代理跟踪级别）』](#) 命令。

在分布式平台上使用 IBM WebSphere MQ V7.5 或更高版本时，只有作为代理进程运行身份的用户才可以运行 **fteSetAgentTraceLevel** 命令。

V 8.0.0.6 对于 z/OS，**fteSetAgentTraceLevel** 命令可以由以下任一人员运行：

- 用于运行代理进程的同一用户标识。
- 代理属性 **adminGroup** 指定的组的成员。

V 8.0.0.6 对于 z/OS，**fteShowAgentDetails** 命令可以由以下任一人员运行：

- 用于运行代理进程的同一用户标识。
- 代理属性 **adminGroup** 指定的组的成员。

有关更多信息，请参阅 [第 591 页的『agent.properties 文件』](#) 中的 **adminGroup** 属性。

IBM MQ 版本 8 中的安全性更改

如果在 IBM WebSphere MQ V 7 上运行 IBM MQ Managed File Transfer 并迁移到 IBM MQ 版本 8，那么 MQMFTCredentials.xml 文件中的用户标识信息将传递到队列管理器，但不会对其执行操作。

这是因为仅在 IBM MQ 版本 8 中支持传递用户标识和密码信息。

commandPath 和代理沙箱

V 8.0.0.6

对于 IBM MQ 版本 8，如果代理配置了代理沙箱，并且设置了代理属性 **commandPath**，那么在代理启动时会自动将 **commandPath** 指定的目录添加到拒绝的路径。如果在未配置代理沙箱的代理上设置 **commandPath** 属性，那么在代理启动时会自动设置一个新沙箱并将 **commandPath** 指定的目录添加到拒绝的目录。

如果在未配置代理沙箱的代理上设置 **commandPath** 属性，那么在代理启动时会自动设置一个新沙箱并将 **commandPath** 指定的目录添加到拒绝的目录。

有关 **commandPath** 属性的更多信息，请参阅 [第 443 页的『commandPath 属性』](#) 和 [第 591 页的『agent.properties 文件』](#)。

commandPath 和用户沙箱

V 8.0.0.6

对于 IBM MQ 版本 8，如果代理配置了一个或多个用户沙箱，并且设置了代理属性 **commandPath**，那么在代理启动时会自动将 **commandPath** 指定的目录（及其所有子目录）作为 <exclude> 元素添加到每个用户沙箱的 <read> 和 <write> 元素。

有关 **commandPath** 属性的更多信息，请参阅 [第 443 页的『commandPath 属性』](#) 和 [第 591 页的『agent.properties 文件』](#)。

将 WebSphere MQ File Transfer Edition V7.0 代理迁移至 IBM WebSphere MQ V7.5 或更高版本

使用 **fteMigrateAgent** 命令将 WebSphere MQ File Transfer Edition V7.0 代理迁移至 IBM WebSphere MQ V7.5 或更高版本。如果已将代理配置为作为 Windows 服务运行，那么必须完成本任务中的可选步骤。

开始之前

识别作为 WebSphere MQ File Transfer Edition 的一部分安装的 WebSphere MQ File Transfer Edition 代理实例。

本主题中的信息不适用于 IBM i。如果您正在使用 IBM i，请在受管文件传输 V8 之前将队列管理器迁移到 IBM MQ V8。

关于此任务

要迁移至 MQ V7.5 或更高版本，请先停止代理，然后将队列管理器安装迁移至 MQ V7.5 或更高版本，并额外选择文件传输组件。迁移队列管理器后，可以使用 **fteMigrateAgent** 命令将代理程序的配置迁移到 MQ V7.5 或更高版本的安装。使用 MQ V7.5 或更高版本的队列管理器启动已迁移代理。

如果代理使用 MQ 绑定方式连接至其队列管理器并且此队列管理器为 V7.0.1.6 或更高版本，那么您还可以使用多个安装和下列步骤执行并行迁移。如果您不希望执行并行迁移，可使用上述步骤迁移队列管理器和代理。

如果代理使用 MQ 绑定方式连接至其队列管理器，并且此队列管理器为 V7.0.1.5 或更低版本，那么必须将此安装迁移至 7.0.1.6 以允许并行迁移，或者必须直接将此安装迁移至 IBM WebSphere MQ V7.5 或更高版本。

如果代理作为 MQ 客户机在网络上连接至其队列管理器，那么只需完成步骤 [第 30 页的『7』](#)，就可以将代理迁移至 IBM WebSphere MQ V7.5 或更高版本。

如果代理配置为 Windows 服务，那么必须使用 **-f** 参数运行该命令。有关更多信息，请参阅 [第 540 页的『fteMigrate 代理 \(将 WebSphere MQ File Transfer Edition V7.0 代理迁移到 IBM WebSphere MQ V7.5 或更高版本\)』](#)。

过程

1. 通过选择 IBM WebSphere MQ Server 和文件传输组件，随现有的队列管理器安装一起安装 MQ V7.5 或更高版本。
2. 停止 WMQFTE V7.0.x 代理。
3. 将队列管理器移至 IBM WebSphere MQ V7.5 或更高版本安装。
4. 使用 **fteMigrateAgent** 命令将代理程序配置从 WMQFTE V7.0.x 安装迁移到与 IBM WebSphere MQ V7.5 或更高版本集成的代理程序功能。
5. 可选：如果 WMQFTE V7.0.x 代理配置为 Windows 服务，请完成以下可选步骤：
 - a) 使用 **fteModifyAgent** 命令来重新配置代理程序，以使其不再是 Windows 服务。
 - b) 从 IBM WebSphere MQ V7.5 或更高版本的安装使用 **fteModifyAgent** 命令将 IBM WebSphere MQ V7.5 或更高版本的代理程序配置为 Windows 服务。
6. 启动 IBM WebSphere MQ V7.5 或更高版本的代理。
7. 可选：要迁移作为客户机跨网络连接至其队列管理器的代理：
 - a) 将 IBM WebSphere MQ V7.5 或更高版本安装到系统上
 - b) 使用 **fteMigrateAgent** 命令将代理程序配置从 WMQFTE v7.0.x 安装迁移到与 IBM WebSphere MQ V7.5 或更高版本集成的代理程序功能。
 - c) 停止 FTE V7.0.x 代理。
 - d) 如果将 WMQFTE v7.0.x 代理程序配置为 Windows 服务，请使用来自 FTE v7.0.x 安装的 **fteModifyAgent** 命令来重新配置代理程序，以使其不作为 Windows 服务运行
 - e) 如果 WMQFTE v7.0.x 代理程序配置为 Windows 服务，请从 IBM WebSphere MQ V7.5 或更高版本的安装使用 **fteModifyAgent** 命令将 IBM WebSphere MQ V7.5 或更高版本的代理程序配置为 Windows 服务。
 - f) 启动 IBM WebSphere MQ V7.5 或更高版本的代理。

将 WebSphere MQ File Transfer Edition V7.0 数据库记录器迁移至 IBM WebSphere MQ V7.5 或更高版本

使用 **fteMigrateLogger** 命令将独立 WebSphere MQ File Transfer Edition V7.0 数据库记录器迁移到 V7.5 或更高版本。如果您已将数据库记录器配置为 Windows 服务，那么必须完成额外的迁移步骤。

开始之前

识别独立数据库记录器的实例。您必须在更新受管文件传输网络的其他部分之前，先更新数据库记录器实例，以使这些实例能够正确处理它们收到的最新版本的传输日志消息。

关于此任务

要直接迁移至 V7.5 或更高版本，请先停止记录器，然后将此安装迁移至所需版本。完成此迁移后，请使用 **fteMigrateLogger** 命令将数据库记录器配置迁移到所需的 MQ 版本。

如果队列管理器为 IBM WebSphere MQ 7.0.1.6 或更高版本，还可以使用多个安装执行并行迁移，如以下步骤所述。如果您不希望执行并行迁移，可使用上述步骤迁移队列管理器和记录器。

如果数据库记录器连接至 IBM WebSphere MQ V7.0.1.5 或更低版本的队列管理器，那么您必须先将此安装迁移至 7.0.1.6 以允许并行迁移，或者必须直接将此安装迁移至 IBM WebSphere MQ V7.5 或更高版本，并额外选择文件传输组件。

如果数据库记录器配置为 Windows 服务，那么必须运行带有 **-f** 参数的 **fteMigrateLogger** 命令。有关更多信息，请参阅第 544 页的『[fteMigrateLogger（将 WebSphere MQ File Transfer Edition V7.0 数据库记录器迁移至 IBM WebSphere MQ V7.5 或更高版本）](#)』。

过程

1. 通过选择 MQ Server 和 受管文件传输 服务组件，随现有的 WebSphere MQ File Transfer Edition V7.0 安装一起安装 IBM WebSphere MQ V7.5 或更高版本。
2. 停止 WebSphere MQ File Transfer Edition V7.0 数据库记录器。
3. 将队列管理器移至 MQ V7.5 或更高版本安装。
4. 使用 **fteMigrateLogger** 命令将数据库记录器配置从 V7.0 安装迁移到 IBM WebSphere MQ V7.5 或更高版本的数据库记录器。
5. 通过在 `MQ_INSTALLATION_PATH/mqft/sql` 中运行 sql 脚本以创建启动数据库记录器所需的表。产品级别每次增加时都有一个脚本，请按顺序运行所有合适的脚本。

您必须按版本顺序运行 `upgrade sql` 脚本，以其当前的 WebSphere MQ File Transfer Edition 或 受管文件传输 级别开始。可用脚本（其中 *** 可以是 Db2、Oracle 或 z/OS）如下：

- `ftelog_tables_***_701-702.sql`
- `ftelog_tables_***_702-703.sql`
- `ftelog_tables_***_703-704.sql`
- `ftelog_tables_***_704-750.sql`
- `ftelog_tables_***_750-7502.sql`
- `ftelog_tables_***_7502-800.sql`

例如，如果使用的是 WebSphere MQ File Transfer Edition V7.0.3，并且要迁移至 IBM MQ V8.0.0.0，请运行 703-704、704-750、750-7502 和 7502-800。

注：在 z/OS 上，可以使用 `ftelog_tables_zos_704-800.sql` 直接从 704 迁移到 800

直到 `ftelog_tables_***_704-750.sql` 的脚本包含在 IBM MQ V8.0.0.0 中，并且位于 `MQ_INSTALLATION_PATH/mqft/sql` 目录中。IBM MQ V8.0.0.0 未随附 750 以上的升级；如果目录中未提供，您可以从 [APARIT01841](#) 中指定的链接下载这些升级。

6. 可选：如果将 WebSphere MQ File Transfer Edition V7.0 数据库记录器配置为 Windows 服务，请完成以下步骤：
 - a) 使用 `fteModifyDatabaseLogger` 命令重新配置数据库记录器，使其不再是 Windows 服务。

- b) 使用 **fteModifyLogger** 命令重新配置 IBM MQ Managed File Transfer V7.5 或更高版本的记录器，使其成为 Windows 服务。
7. 启动 IBM MQ Managed File Transfer V7.5 或更高版本的数据库记录器。

结果

数据库记录器现在已从 WebSphere MQ File Transfer Edition V7.0 迁移至 IBM WebSphere MQ V7.5 或更高版本。

在 Windows，UNIX 或 Linux 上的 Db2 上增加日志数据库的页面大小

如果您的数据库在 Windows，UNIX 或 Linux 系统上是 Db2，并且您创建的日志数据库的页面大小小于 8 KB，那么必须在迁移到 V7.0.3 或更高版本的表之前增大数据库的页面大小。

过程

1. 如果您尚未停止数据库记录器，请使用 **fteStopDatabaseLogger** 命令来停止该数据库记录器。
2. 使用 Db2 提供的工具来备份日志数据库。
3. 使用 Db2 **export** 命令将数据从日志数据库表传输到磁盘上的文件。

注：必须为包含大对象的表指定大对象文件。这些表是 CALL_RESULT 和 METADATA。

4. 删除日志数据库表。
5. 创建页面大小至少为 8KB 的表空间，并将页面大小至少为 8KB 的缓冲池与之关联。
为新表空间命名。例如，FTE8KSPACE。
6. 编辑 ftelog_tables_db2.sql 文件，以便这些命令在新表空间中创建表。
在 ftelog_tables_db2.sql 文件中，将出现的所有 IN “USERSPACE1” 文本更改为 IN “new_tablespace_name”。例如，将 IN “USERSPACE1” 更改为 IN “FTE8KSPACE”。
7. 针对您的数据库运行 ftelog_tables_db2.sql 文件中的 SQL 命令。
8. 使用 Db2 **load** 命令将导出的数据传输到新表。

注：

- 基于输入文件中找到的列名称来映射列名称。确保输入列名称和目标列名称在已更改其结构的这些表中相匹配。
 - 必须在除了 MONITOR 和 TRANSFER 之外的所有表的标识列上指定 IDENTITY OVERRIDE 行为。指定该行为将确保在装入操作期间不会重新生成行标识。
9. 运行 Db2 **set integrity** 命令，其完整性状态值为 **immediate** 和 **checked**，并按给定的顺序对下列各表执行操作：
 - CALL_ARGUMENT
 - MONITOR
 - MONITOR_ACTION
 - MONITOR_EXIT_RESULT
 - MONITOR_METADATA
 - SCHEDULE_ACTION
 - SCHEDULE
 - SCHEDULE_ITEM
 - TRANSFER
 - TRANSFER_CALLS
 - TRANSFER_EVENT
 - TRANSFER_ITEM
 - TRANSFER_STATS
 - TRIGGER_CONDITION

10. 在具有生成的标识列的表中，将标识生成器设置为从比现有最大标识值大 1 的值开始。

以下表具有生成的标识列：

- AUTH_EVENT
- CALL
- CALL_ARGUMENT
- CALL_RESULT
- FILE_SPACE_ENTRY
- METADATA
- MONITOR_ACTION
- MONITOR_EXIT_RESULT
- MONITOR_METADATA
- SCHEDULE
- SCHEDULE_ACTION
- SCHEDULE_ITEM
- SCHEDULE_SPEC
- TRANSFER_CALLS
- TRANSFER_CD_NODE
- TRANSFER_CORRELATOR
- TRANSFER_EVENT
- TRANSFER_EXIT
- TRANSFER_ITEM
- TRANSFER_ITEM_ATTRIBUTES
- TRANSFER_STATS
- TRIGGER_CONDITION

要将这些表的已生成标识设置为正确值，请为每个表执行以下步骤：

a) 确定现有数据中的最大标识值。

您可以通过运行此 SQL 语句来查找该值：

```
SELECT MAX(ID) FROM FTELOG.table_name
```

从该命令返回的值是指定的表中最大的现有标识。

b) 变更该表以设置标识生成器，使其从比上一步返回的值大 1 的新值开始。

您可以通过运行以下 SQL 语句来设置该值：

```
ALTER TABLE FTELOG.table_name ALTER COLUMN ID RESTART WITH value
```

相关任务

第 179 页的『从独立数据库记录器迁移至 Java EE 数据库记录器』

您可以从独立数据库记录器迁移到 Java EE 数据库记录器。必须停止独立数据库记录器，并安装 JEE 数据库记录器。为避免日志条目丢失或重复，您必须先停止正在向 SYSTEM.FTE 主题发布的消息，然后再停止独立数据库记录器，并在安装 Java EE 数据库记录器之后重新启动独立数据库记录器。备份您的数据库，然后再执行迁移。生成的新信息。

第 34 页的『将 Db2 on z/OS 上的数据库表迁移到 V8.0.0』

如果您的数据库是 Db2 on z/OS 系统，那么必须完成以下步骤，以在不同版本的 WebSphere MQ File Transfer Edition 之间进行迁移（即从 V7.0.3 迁移至 V7.0.4，以及从 WebSphere MQ File Transfer Edition V7.0.4 迁移至 IBM MQ Managed File Transfer V8.0.0）。Db2 表的结构与前发行版不同。例如，某些表中增加了新列，某些可变字符列可能更大，因此必须将前发行版中的表迁移至 V8.0 格式。

将 Db2 on z/OS 上的数据库表迁移到 V8.0.0

如果您的数据库是 Db2 on z/OS 系统，那么必须完成以下步骤，以在不同版本的 WebSphere MQ File Transfer Edition 之间进行迁移（即从 V7.0.3 迁移至 V7.0.4，以及从 WebSphere MQ File Transfer Edition V7.0.4 迁移至 IBM MQ Managed File Transfer V8.0.0）。Db2 表的结构与前发行版不同。例如，某些表中增加了新列，某些可变字符列可能更大，因此必须将前发行版中的表迁移至 V8.0 格式。

关于此任务

IBM MQ Managed File Transfer V8.0 支持传输非常大的文件，即超过可在一个整数（32 位）中存储的大小。提供了一个使用 BIGINT（64 位）数字的定义。您可以选择使用名为 `ftelog_tables_zos.sql` 的整数定义或位于 `prod/mqf/sql` 目录中名为 `ftelog_tables_zos_bigint.sql` 的 BIGINT 定义。

BIGINT 数据类型在 WebSphere MQ File Transfer Edition V7.0.4 Fix Pack 3 或者 IBM MQ Managed File Transfer V7.5 Fix Pack 1 或更高版本中可用。

要启用 BIGINT 数据类型，必须使用 Db2 V9.1 或更高版本。表示所传输文件大小和与每个传输关联的表标识的字段将使用 INTEGER 数据类型。如果要记录文件大小大于 2GB 的传输或者要在数据库中存储 2,147,483,648 次以上的独立传输，那么必须使用 BIGINT SQL 文件。

以下列表概述了您需要遵循的过程：

1. 您在产品 V7 中存在现有的表。这些表采用了一种模式，例如 FTELOG。
2. 使用其他模式名称 (例如 FTEV8) 创建 V8 表。这允许您将数据从 FTELOG.table 复制到 FTEV8.table。
3. 将数据复制到新表
4. 在新表中设置生成的标识值
5. 运行 **fteMigrateLogger** 命令以将属性文件移动到目录结构中的新位置。
6. 编辑记录器属性文件以指定新模式 (FTEV8)
7. 编辑现有的记录器 JCL，以使用 V8 IBM MQ Managed File Transfer 库。
8. 启动记录器。
9. 记录器工作后，您可以删除 FTELOG 表。

在以下描述中，IBM MQ Managed File Transfer 产品安装在 USS 中的 /HMF8800 目录中。

过程

1. 如果尚未停止数据库记录器，请使用 USS 或 **P loggerjob** 中的 **fteStopDatabaseLogger** 命令来停止数据库记录器。
2. 发出命令 **ls /HMF8800/mqft/sql** 以列出目录中的 SQL 文件。
如果要使用 BIGINT 数字副本，请将 `ftelog_tables_zos_bigint.sql` 复制到主目录，否则，请将 `ftelog_tables_zos.sql` 复制到主目录。
3. 编辑移动到主目录中的文件：
 - a) 将 `ftelog` 更改为新表的模式名称。
 - b) 确保每个索引都具有唯一名称。
为此，在编辑会话中：
 - i) 排除所有行。
 - ii) 查找 'CREATE UNIQUE INDEX ' ALL
 - iii) 更改 `_KEY _K8Y ALL NX`
4. 检查该文件以确保所有语句都在第 71 列中。
如果语句不在第 71 列中，请拆分第 71 列之前的行。
5. 您可能可以将该文件作为 SQL 的输入，也可能希望将其复制到 PDS。为此，编辑该 PDS 并使用 **copy** 命令，同时指定目录和文件名。
6. 与 Db2 管理员一起检查定义，因为您可能需要遵循一些站点标准。

7. 执行以下操作:

- a) 将位于 /HMF8800/mqft/sql/ftelog_tables 目录中的 _zos_704-800.sql 文件复制到您的主目录。
- b) 编辑此文件。将 FTESRC 更改为现有模式 (FTELOG), 将 FTEDEST 更改为新模式 (FTEV8)。
- c) 检查该文件以确保所有语句都在第 71 列中。
如果语句不在第 71 列中, 请拆分第 71 列之前的行。
- d) 如果将 **DB2 RUNSTATS** jcl 用于 IBM MQ Managed File Transfer 表, 请创建新的作业, 同时指定新的模式和表。

8. 某些表包含生成的标识, 用于为每个行强制实施唯一标识, 您需要设置这些标识。

以下表具有生成的标识列:

- AUTH_EVENT
- CALL
- CALL_ARGUMENT
- CALL_RESULT
- FILE_SPACE_ENTRY
- METADATA
- MONITOR_ACTION
- MONITOR_EXIT_RESULT
- MONITOR_METADATA
- SCHEDULE
- SCHEDULE_ACTION
- SCHEDULE_ITEM
- SCHEDULE_SPEC
- TRANSFER_CALLS
- TRANSFER_CD_NODE
- TRANSFER_CORRELATOR
- TRANSFER_EVENT
- TRANSFER_EXIT
- TRANSFER_ITEM
- TRANSFER_ITEM_ATTRIBUTES
- TRANSFER_STATS
- TRIGGER_CONDITION

要将这些表的已生成标识设置为正确值, 请为每个表执行以下步骤:

- a) 确定现有数据中的最大标识值。

您可以通过运行此 SQL 语句来查找该值:

```
SELECT MAX(ID) FROM schema_name.table_name
```

从该命令返回的值是指定的表中最大的现有标识。

- b) 变更该表以设置标识生成器, 使其从比上一步返回的值大 1 的新值开始。

您可以通过运行以下 SQL 语句来设置该值:

```
ALTER TABLE schema_name.table_name ALTER COLUMN ID RESTART WITH value
```

9. 编辑数据库属性文件以指定新的模式名称:

- a) 如果 IBM MQ Managed File Transfer 配置目录为 `/u/userid/fteconfig`，那么可以使用 USS 命令 **`find /u/userid/fteconfig -name databaselogger.properties`** 来查找记录器的属性文件。
 - b) 编辑此文件并将 `wmqfte.database.schema` 更改为新的模式值。
10. 尝试使用记录器之前，请发出以下命令，以将目录树结构转换为 V8.0.0 格式：
- a) **`fteMigrateConfigurationOptions`**
 - b) **`fteMigrateLogger`**
- 这会将 `databaselogger.properties` 复制到 `logger.properties`。
11. 编辑现有的 Logger JCL，以使用 V8.0.0 IBM MQ Managed File Transfer 库。
12. 启动记录器。
- 记录器开始工作后，您即可以删除 V7 FTELOG 表。

将 MFT 迁移到具有其他操作系统的新机器

成功将 MFT 配置迁移到新系统或平台所需的核心步骤。该任务主要关注 MFT 配置迁移，但也在适当情况下讨论队列管理器迁移。

开始之前

确保要迁移的任何代理已完成任何进行中或暂挂的传输，并且您已备份以下内容：

- 协调队列管理器
- 代理队列管理器
- 代理
- 资源监视器
- 传输模板
- 调度的传输

要点：一个系统上的 IBM MQ 安装名称不太可能与新系统上的安装名称匹配，除非旧系统和新系统只有一个安装，或者您在 IBM MQ 安装过程中指定安装名称。

关于此任务

以下迁移过程基于以下场景：QMA 既是拓扑的协调队列管理器，也是名为 **Agent1** 的代理的代理队列管理器。

Agent1 具有监视器，传输模板和调度的传输。QMA 还连接到名为 **QMB** 的队列管理器，该队列管理器在另一个系统上使用其发送方和接收方通道进行文件传输。

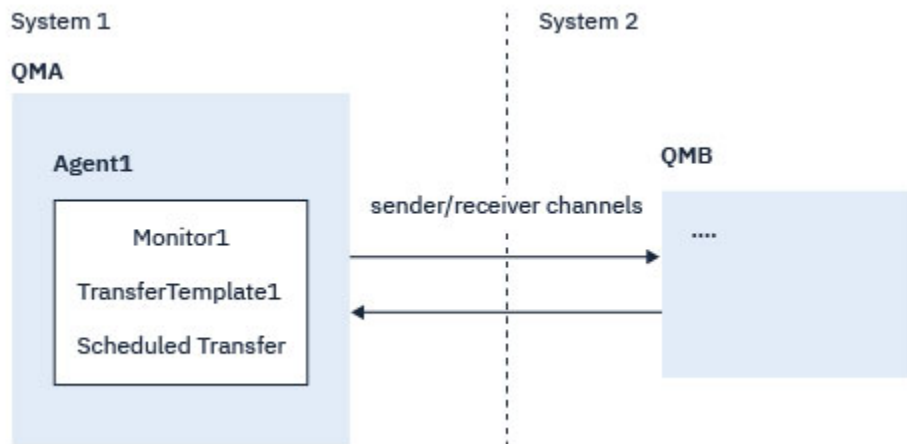


图 6: 在系统 1 上迁移 MFT 配置



注意: 以下过程仅说明如何备份和复原 MFT 配置。如果要 将 MFT 迁移到具有相同操作系统的新机器, 那么可以通过将所有数据文件从旧系统复制到新系统上的相应目录来备份和复原队列管理器数据和日志文件。

但是, 如果新机器具有不同的操作系统, 那么无法迁移数据文件, 因为它们是特定于平台创建的。

过程

1. 备份过程

- a) 使用 **dmpmqcfg** 命令保存队列管理器配置, 以便稍后根据其定义重新构建队列管理器配置。

例如:

```
dmpmqcfg -m QMA -a > /mq/backups/QMA.mqsc
```

- b) 备份存储在 IBM MQ data directory /MQ_DATA_PATH/mqft 下的代理程序的配置文件 **mqft** 目录通常有三个子目录, 即 **config**, **installation** 和 **logs**。这些文件分别包含代理程序安装数据, 配置和数据库记录器文件。

如果代理是协议网桥代理, 那么还需要备份代理配置目录中的

ProtocolBridgeCredentials.xml 文件。此文件定义协议网桥代理用于向其自身授权协议服务器的用户名和凭证信息。

- c) 使用带有 **-ox** 选项的 MFT **ftelistMonitors** 命令将资源监视器的配置导出到 XML 文件。

例如:

```
fteListMonitors -ma Agent1 -mn Monitor -ox Monitor1Definition.xml
```

- d) 使用带有 **-x** 和 **-o** 选项的 MFT **fteListTemplates** 命令将传输模板导出到 XML 文件。

例如, 以下命令将在当前目录中创建 **TransferTemplate1.xml** :

```
fteListTemplates -x -o . TransferTemplate1
```

- e) 手动备份调度的传输定义。

无法将定义导出到 XML 文件, 但您可以使用 MFT **fteListScheduledTransfers** 命令列出调度的传输, 并手动备份定义。

2. 重新创建过程

- a) 在新系统上安装 IBM MQ 和 MFT 后重新创建队列管理器 QMA。

- b) 通过运行 **runmqsc** 命令在步骤 第 37 页的『1.a』中保存的队列管理器配置中进行解析来复原 QMA 配置

例如:

```
runmqsc QMA< /mq/backups/QMA.mqsc
```

- c) 在系统 2 上重新创建连接到 QMB 的发送方和接收方通道。

- d) 在 QMB 队列管理器端, 更新连接详细信息, 例如连接到 QMA 的发送方通道的主机名和端口号。

- e) 通过将所有备份的代理程序配置文件复制到新系统来重新创建 Agent1, 然后启动代理程序。

- f) 使用带有 **-ix** 和 **-f** 选项的 MFT **fteCreateMonitor** 命令导入 Monitor1 的 XML 文件。

例如:

```
fteCreateMonitor -ix Monitor1Definition.xml -f
```

- g) 将消息体中包含 **TransferTemplate1.xml** 内容的消息发布到 **SYSTEM.FTE** 主题。

使用独立应用程序, 并指定主题字符串:

```
SYSTEM.FTE/Templates/<template_id>
```

其中 **<template_id>** 是可在 **TransferTemplate1.xml** 文件中找到的传输模板标识。

例如, 如果 xml 包含:

```
<?xml version="1.0" encoding="UTF-8"?><transferTemplateid="a7838085-0f2a-4980-b958-2dbbdfb22702" version="6.00">
```

，主题字符串应该为：

```
SYSTEM.FTE/Templates/a7838085-0f2a-4980-b958-2dbbdfb22702
```

h) 使用 MFT **[fteCreateTransfers](#)** 命令手动重新创建调度的传输。

在零售环境中使用 IBM MQ Managed File Transfer

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

如何安装

有关 IBM MQ 安装的概述，请参阅[安装和卸载](#)。

有关您的平台要安装哪些特定 MQMFT 组件的信息，请参阅[选择安装内容](#)。

有关在零售环境中准备安装 MQMFT 的信息，请参阅第 60 页的『[准备在 IBM 4690 系统上安装 IBM MQ Managed File Transfer](#)』。

相关参考

第 81 页的『[在 4690 OS 系统上运行时的限制](#)』

在零售环境中的 4690 OS 系统上运行 IBM MQ Managed File Transfer 时，存在一些限制和不受支持的功能。

零售环境中的场景

本部分提供了各种场景，介绍如何使用针对 IBM 4690 操作系统的 IBM MQ 支持。

子主题

- 第 38 页的『[1. 入门：使用商店中的 4690 OS 传输文件](#)』

本场景解释了如何在 4690 OS 平台上开始使用 IBM MQ Managed File Transfer。本场景有助于说明在将 IBM MQ Managed File Transfer 代理部署到 4690 OS 平台时的特殊注意事项。

- 第 49 页的『[2. 将文件从总部传输到商店中的 4690 OS 系统](#)』

创建一个双计算机拓扑，分别表示位于总部站点的一台计算机和位于零售商店的一台 4690 OS 商店控制器。了解如何创建从总部站点将文件传输到零售商店内部署的 4690 OS 系统所需的定义和权限。

- 第 54 页的『[3. 将文件从商店中的 4690 OS 系统传输到总部](#)』

包括创建自动将 4690 OS 系统上创建的文件传输到总部计算机所需的定义。了解如何使用 IBM MQ Managed File Transfer 的资源监视概念自动传输在特定目录内创建的任何文件。

相关概念

第 38 页的『[在零售环境中使用 IBM MQ Managed File Transfer](#)』

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

1. 入门：使用商店中的 4690 OS 传输文件

您可以使用本场景来帮助在 4690 OS 上开始使用 IBM MQ Managed File Transfer。

可控、可靠并具有审计跟踪的文件传输是在零售商店和零售总部之间交换数据的基本需求。本场景可让您基本了解如何使用 IBM MQ Managed File Transfer 在总部计算机和零售商店中部署的 4690 OS 商店控制器之间交换数据。后续主题中的场景演示了不同的交互模式。需要这些不同模式来演示 受管文件传输 组件如何解决零售商遇到的实际业务问题。

要完成本场景，您应具备 IBM MQ 和 IBM MQ Managed File Transfer 的基本知识。尤其应熟悉以下概念：

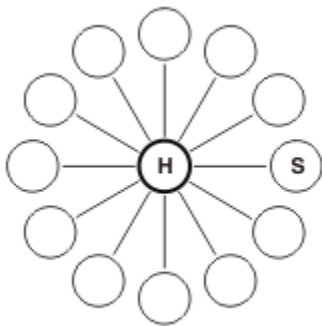
- 队列管理器的概念
- IBM MQ 的基本配置和管理
- IBM MQ 代理的概念
- IBM MQ Managed File Transfer 的基本配置和管理

有关 IBM MQ Managed File Transfer 功能的更多信息，请参阅 [第 5 页的『IBM MQ Managed File Transfer 介绍』](#)。

在本场景中，IBM MQ 可用于启动并跟踪从 4690 OS 回到同一系统的文件传输。本场景中的传输可帮助您理解后续主题中的场景，例如，将文件从总部计算机传输到商店中的 4690 OS，以及将文件从商店中的 4690 OS 传输到总部站点。

文件传输拓扑示例

本场景和后续场景都基于中心辐射型拓扑。该图显示包含一个中心站点 (H) 和多个周边站点 (S) 的概念性中心辐射型拓扑。



中心辐射型拓扑代表许多零售商的文件传输需求。在这个案例中，中心站点对应中央总部站点，周边站点对应零售商运营的商店。通常，软件配置、部署和管理都在总部站点内完成。

在本场景中，您可以完成以下任务：

- 规划解决方案。
- 在 4690 OS 上配置 IBM MQ 以进行文件传输。
- 为 4690 OS 上运行的代理创建配置。
- 将代理部署到 4690 OS。
- 传输文件以验证该场景。

相关概念

[第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』](#)

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先在非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

[第 40 页的『规划解决方案』](#)

要在 4690 OS 上使用 IBM MQ Managed File Transfer，必须具有包含至少 2 台计算机的配置。

[第 47 页的『将代理部署到 4690 OS 系统』](#)

实施此场景所描述的解决方案需要在 4690 OS 上部署 IBM MQ Managed File Transfer 代理。通过将该代理配置为 4690 OS 后台应用程序，可启动该代理。

第 48 页的『传输文件以验证该场景』

通过将文件从 4690 OS 系统 (COMPUTER2) 传输回主机 4690 OS 系统，来验证在此场景中构建的拓扑。本任务假设您使用的是 Windows 系统。对于 UNIX 系统，请替换相应的路径和命令。

相关任务

第 43 页的『为文件传输配置 IBM MQ』

通过发出命令构建用于基本文件传输方案的拓扑，来配置 IBM MQ 以进行文件传输。

第 45 页的『为 4690 OS 系统上的代理创建配置』

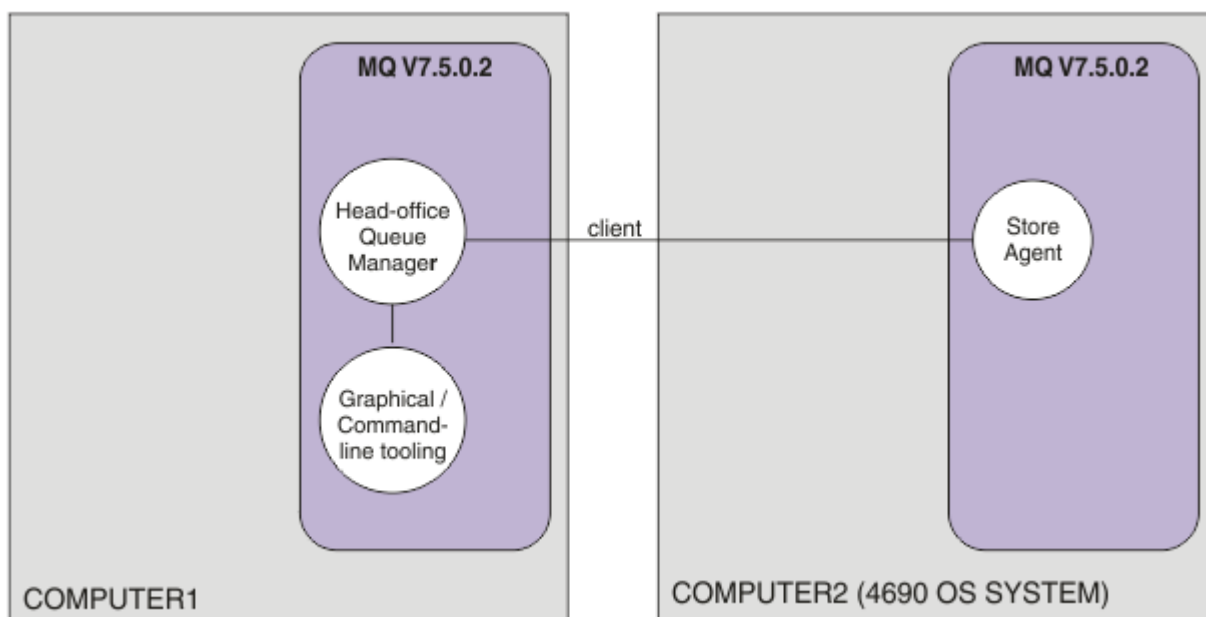
实施此场景所描述的解决方案需要创建配置捆绑软件。配置捆绑软件将 4690 OS IBM MQ Managed File Transfer 代理需要的所有配置打包在一起。

规划解决方案

要在 4690 OS 上使用 IBM MQ Managed File Transfer，必须具有包含至少 2 台计算机的配置。

最低配置包含 4690 OS 上的 IBM MQ Managed File Transfer 代理并需要 2 台计算机：

- 部署在总部站点的计算机。
- 部署在零售商店中的 4690 OS 商店控制器计算机。



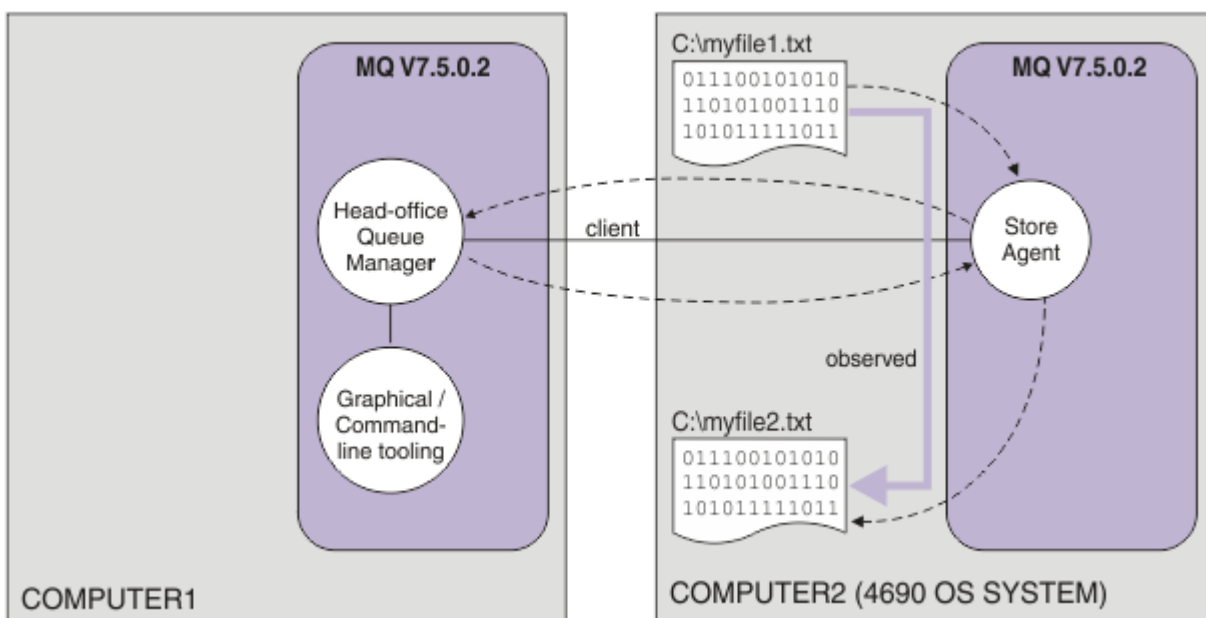
该图显示跨两台计算机的总部和商店拓扑。COMPUTER1 是总部计算机。COMPUTER2 是通常位于零售商店的 4690 OS 商店控制器。COMPUTER2 上的 IBM MQ Managed File Transfer 代理通过客户机连接与 COMPUTER1 上的队列管理器相连。

部署在总部站点的计算机用于完成以下活动：

1. 为 4690 OS 系统上运行的 IBM MQ Managed File Transfer 代理创建配置。
2. 运行用于为 IBM MQ Managed File Transfer 组件提供连接的 IBM MQ 队列管理器。
3. 运行用于配置和管理受管文件传输操作的图形或命令行工具。

为简单起见，本场景使用一台计算机来完成总部站点执行的所有活动。您必须考虑您的部署是否会因使用多台计算机完成这些活动而受益。

部署在零售商店中的 4690 OS 计算机运行 4690 OS IBM MQ Managed File Transfer 代理组件。在本场景中，将文件从零售商店传输到总部，然后再传输回零售商店。虽然不能代表实际用例，但传输文件可验证该场景是否正确配置，并且可充当后续场景的基础，这些后续场景在总部计算机和零售商店中的 4690 OS 计算机之间交换文件数据。



该图显示本场景演示的文件传输路线。在本示例中，假设 COMPUTER1 是 Windows 计算机，COMPUTER2 是 4690 OS 商店控制器。您可能想要将备用平台或体系结构用于 COMPUTER1；以获取受支持平台的完整列表，请参阅 <https://www.ibm.com/support/docview.wss?uid=swg27006467>

本场景假设您使用的是 Windows 系统。对于 UNIX 系统，请替换相应的路径和命令，并确保您具有所有相关目录的读写访问权。本场景还假设您基本了解队列管理器的用途。

了解安全模型

本场景以及后续主题中的场景将创建一个具有以下安全特征的文件传输拓扑：

- 在四个角色之间划分对 IBM MQ 的访问权：
 1. IBM MQ 管理员，负责配置 IBM MQ 并创建完成受管文件传输操作所需的配置。
 2. FTHOFFS，负责启动和停止总部部署的代理并与之交互。这在第 49 页的『2. 将文件从总部传输到商店中的 4690 OS 系统』场景中进行了描述。
 3. FTSTORES，对于位于总部的队列管理器，其代表启动和停止零售商店中部署的代理并与之交互的角色。
 4. FTUSERS，负责启动文件传输。

有关配置访问的更多信息，请参阅第 432 页的『特定于 IBM MQ Managed File Transfer 的资源权限』。

- 所有来自零售商店的基于代理的连接都映射到中心站点的单个用户 FTSTORES。
- 基于 IP 的弱认证用于对零售商店中部署的代理进行认证。

为清晰起见，本场景中的安全模型并不是完全安全的。在生产中部署相似拓扑之前，您必须清楚自己的拓扑需求和安全威胁。因此，您必须考虑是否处理以下潜在漏洞：

- 任何用户均可以模拟任何其他人。在文件传输资源的对象访问模型中考虑使用更细的粒度。
- 任何代理均可以模拟任何其他代理。考虑使用更强的认证，例如，在文件传输资源的对象访问模型中考虑使用 TLS/SSL 和更细粒度。
- 不考虑文件系统与 IBM MQ 安全之间的接口。考虑实施文件沙箱，并了解代理配置文件许可权的影响。
- 不描述代理和操作系统之间的接口。考虑实施文件沙箱。有关沙箱的更多信息，请参阅沙箱。

有关安全、IBM MQ 和文件传输的更多信息，请参阅第 48 页的『传输文件以验证该场景』中的“后续操作”。

先决条件和许可证

您需要以下几项才能完成此场景：

- 满足 IBM MQ 的硬件和操作系统先决条件的测试计算机，有关详细信息，请参阅 <https://www.ibm.com/support/docview.wss?uid=swg27006467>，其中不包含 IBM MQ 或 IBM MQ 数据的现有安装。
- 4690 OS 商店控制器，其满足 IBM MQ Managed File Transfer 的硬件和操作系统先决条件。
- IBM WebSphere MQ V7.5。
- IBM MQ FP7.5.0.2。您可以从 <https://www.ibm.com/support/docview.wss?uid=swg27038184> 下载该修订包。

准备总部计算机

确保测试计算机满足 IBM MQ V7.5 的安装需求。有关更多信息，请参阅[检查需求](#)。

在总部计算机上安装队列管理器

安装 IBM MQ V7.5 服务器及以下组件：

- 服务器
- IBM MQ Explorer
- IBM MQ Managed File Transfer 代理
- IBM MQ Managed File Transfer 命令行工具

有关要安装的组件的详细信息，请参阅[选择安装内容](#)。

确定管理 IBM MQ 的方式。您可以通过以下方式管理 IBM MQ：

- 使用 **setmqenv** 命令设置相应的环境。有关更多信息，请参阅 [setmqenv](#)。
- 调用标准的 IBM MQ 管理命令。

本场景假设您使用未安装先前 IBM MQ 或 IBM MQ Managed File Transfer 安装版本的干净计算机。如果情况并非如此，您必须确定是否支持共存，或者调整环境的安装机制和配置。有关详细信息，请参阅[多个安装](#)。

准备 4690 OS 系统

确保您的测试 4690 OS 商店控制器满足 IBM MQ Managed File Transfer V7.5.0.2 的安装需求，请参阅[检查需求](#)。

在 4690 OS 系统上安装代理

将 IBM MQ Managed File Transfer V7.5.0.2 安装到 4690 OS 商店控制器。有关更多信息，请参阅 [第 61 页的『在 4690 OS 上安装 IBM MQ Managed File Transfer』](#)。

创建组 and 用户

本场景中使用的安全模型假设您在总部计算机上创建了以下组 and 用户：

- 组
 - mqm
安装 IBM MQ 时创建。该组成员可管理 IBM MQ 及其资源。
 - FTHOFFS
必须创建该组。该组成员可启动和停止总部部署的代理并与之交互。有关更多信息，请参阅 [第 49 页的『2. 将文件从总部传输到商店中的 4690 OS 系统』](#)。
 - FTSTORES

必须创建该组。该组供位于总部的队列管理器使用，代表可启动和停止零售商店中部署的代理并与之交互的用户组。

– FTUSERS

必须创建该组。该组成员可启动文件传输。

• 用户

– mqmAdmin

IBM MQ 管理员。必须创建该用户：

- 在 Windows 上，此用户必须同时是 mqm 组的成员和 Windows 管理员，才能定义作为 Windows 服务运行的 WebSphere MQ File Transfer Edition 代理程序
- 在其他平台上，该用户只需是 mqm 组成员。

– ftuser

必须创建该用户，并使其成为 FTUSERS 组的成员。要避免可能对队列管理器进行的管理级别安全访问，请不要将该用户添加到 mqm 组，或使该用户成为 Windows 管理员。

– fthoff

必须创建该用户，并使其成为 FTHOFFS 组的成员。要避免可能对队列管理器进行的管理级别安全访问，请不要将该用户添加到 mqm 组，或使该用户成为 Windows 管理员。

在 Windows 上，该用户用于将 IBM MQ Managed File Transfer 代理进程作为 Windows 服务运行。该用户要求：

- 该帐户已设置密码
- 该帐户以服务权限登录，请参阅：[第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』](#)。

– ftstore

必须创建该用户，并使其成为 FTSTORES 组的成员。为防止该用户无意间具有队列管理器的管理级别安全访问权，请勿将该用户添加到 mqm 组，也不要使该用户成为 Windows 管理员。

相关概念

[第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』](#)

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

[第 38 页的『1. 入门：使用商店中的 4690 OS 传输文件』](#)

您可以使用本场景来帮助在 4690 OS 上开始使用 IBM MQ Managed File Transfer。

为文件传输配置 IBM MQ

通过发出命令构建用于基本文件传输方案的拓扑，来配置 IBM MQ 以进行文件传输。

关于此任务

在本任务中，您将完成以下活动：

- 在总部计算机上创建队列管理器。
- 设置代理（在 4690 OS 系统上运行）连接到队列管理器所需的 IBM MQ 对象定义。
- 创建基本 IBM MQ Managed File Transfer 拓扑所需的 IBM MQ 对象定义。
- 对此拓扑应用基本安全模型。

本任务假设您使用的是 Windows 系统。对于 UNIX 系统，请替换相应的路径和命令。

在 IBM MQ bin 目录 `MQ_INSTALL_ROOT\bin` 中，从用户 mqmAdmin 完成以下步骤。

过程

1. 创建名为 `hoffQM` 的队列管理器。

```
crtmqm hoffQM
```

2. 启动队列管理器。

```
strmqm hoffQM
```

3. 为 `hoffQM` 启动 MQSC 接口。

```
runmqsc hoffQM
```

4. 创建一个通道以用于处理来自 IBM MQ Managed File Transfer 代理（在 4690 OS Store Controller 系统上运行）的入局连接。

```
DEFINE CHANNEL(FTE.AGENT.SVRCONN) CHLTYPE(SVRCONN)
```

5. 创建通道认证记录，使来自 4690 OS 商店控制器系统的连接能进入队列管理器 `hoffQM`。通道认证记录可以限制哪些 IP 地址可以连接到队列管理器，并能将用户 `ftstore` 的标识分配给连接。必须为每个店内代理运行此命令。

```
SET CHLAUTH('FTE.AGENT.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('IP address of 4690 OS system')  
  USERSRC(MAP) MCAUSER('ftstore') DESCR('Rule to allow 4690 at store to connect')  
ACTION(ADD)
```

有关通道认证记录的更多信息，请参阅通道认证记录。此场景的目标是不锁定和保护拓扑，但是会演示基本文件传输。实施的安全模型支持此演示，但是您必须了解自己的安全性威胁，并在需要时采取适当的操作。有关可考虑使用的选项的讨论，请参阅第 48 页的『传输文件以验证该场景』中的“下一步做什么”。

6. 标识可用于与总部系统上运行的队列管理器进行网络通信的可用端口，例如 1414。定义侦听器 `LISTENER1` 以使用此可用端口。

```
DEFINE LISTENER(LISTENER1) TRPTYPE(TCP) CONTROL(QMGR) PORT(1414)
```

7. 启动侦听器 `LISTENER1`。

```
START LISTENER(LISTENER1)
```

8. 结束队列管理器 `hoffQM` 的 MQSC 接口。

```
end
```

9. 将队列管理器 `hoffQM` 配置为 IBM MQ Managed File Transfer 拓扑的协调队列管理器。

```
fteSetupCoordination -coordinationQMgr hoffQM
```

关于协调队列管理器角色的更多信息，请参阅第 14 页的『IBM MQ Managed File Transfer 拓扑概述』。

10. 使用 MQSC 接口定义使队列管理器 `hoffQM` 成为协调队列管理器所需的 IBM MQ 对象。

```
runmqsc hoffQM < ..\mqft\config\hoffQM\hoffQM.mqsc
```

11. 将队列管理器 `hoffQM` 配置为 IBM MQ Managed File Transfer 拓扑的命令队列管理器。

```
fteSetupCommands -connectionQMgr hoffQM
```

12. 在队列管理器 **hoffQM** 上创建代理 **SAGENT** 所需的对象定义。必须为每个店内代理运行此命令。

```
fteDefine -t agent SAGENT | runmqsc hoffQM
```

在 WebSphere MQ 7.5.0.2 修订包中引入了 **fteDefine** 命令。如果在 IBM MQ 安装中找不到此命令，请使用 **dspmqver** 命令检查该安装的服务级别。有关 **fteDefine** 命令的更多信息，请参阅第 519 页的『fteDefine（生成配置脚本）』。

13. 确保 FTSTORES 和 FTUSERS 组具有相应的访问权，以便能够访问完成文件传输操作所需的 IBM MQ 对象。您可以根据自己的安全性需求来定制该配置。

```
setmqaut -m hoffQM -t qmgr -g FTSTORES +connect +inq +setid +altusr  
setmqaut -m hoffQM -n SYSTEM.FTE -t queue -g FTSTORES +get +put  
setmqaut -m hoffQM -n SYSTEM.FTE -t topic -g FTSTORES +pub +sub  
setmqaut -m hoffQM -n SYSTEM.DEFAULT.MODEL.QUEUE -t queue -g FTSTORES +browse +dsp +get +put  
setmqaut -m hoffQM -t qmgr -g FTUSERS +connect  
setmqaut -m hoffQM -n SYSTEM.FTE -t topic -g FTUSERS +sub  
setmqaut -m hoffQM -n SYSTEM.DEFAULT.MODEL.QUEUE -t queue -g FTUSERS +browse +dsp +get +put
```

以下命令必须针对每个店内代理运行。

```
setmqaut -m hoffQM -n SYSTEM.FTE.COMMAND.SAGENT -t queue -g FTSTORES +browse +get +put  
+setid  
setmqaut -m hoffQM -n SYSTEM.FTE.DATA.SAGENT -t queue -g FTSTORES +get +put  
setmqaut -m hoffQM -n SYSTEM.FTE.EVENT.SAGENT -t queue -g FTSTORES +browse +get +put  
setmqaut -m hoffQM -n SYSTEM.FTE.REPLY.SAGENT -t queue -g FTSTORES +browse +get +put  
setmqaut -m hoffQM -n SYSTEM.FTE.STATE.SAGENT -t queue -g FTSTORES +browse +get +inq +put  
setmqaut -m hoffQM -n SYSTEM.FTE.COMMAND.SAGENT -t queue -g FTUSERS +put
```

有关 **setmqaut** 命令的更多信息，请参阅 [setmqaut](#)。有关向组授予权限的更多信息，请参阅第 433 页的『特定于 IBM MQ Managed File Transfer 的资源的组权限』。

14. 设置允许从 4690 OS 系统发出的文件发回至 4690 OS 系统所需的权限。只需要针对一个店内代理运行这些命令，便可以验证此安装。这些权限会在场景 2 结束时被移除。

```
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHTRN1.SAGENT -t queue -g FTUSERS +browse +put  
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHAGT1.SAGENT -t queue -g FTSTORES +browse +put
```

有关向特定 IBM MQ Managed File Transfer 操作授予权限的更多信息，请参阅第 438 页的『用户的 IBM MQ Managed File Transfer 操作权限』。

相关概念

第 38 页的『1. 入门：使用商店中的 4690 OS 传输文件』

您可以使用本场景来帮助在 4690 OS 上开始使用 IBM MQ Managed File Transfer。

第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先在非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

为 4690 OS 系统上的代理创建配置

实施此场景所描述的解决方案需要创建配置捆绑软件。配置捆绑软件将 4690 OS IBM MQ Managed File Transfer 代理需要的所有配置打包在一起。

关于此任务

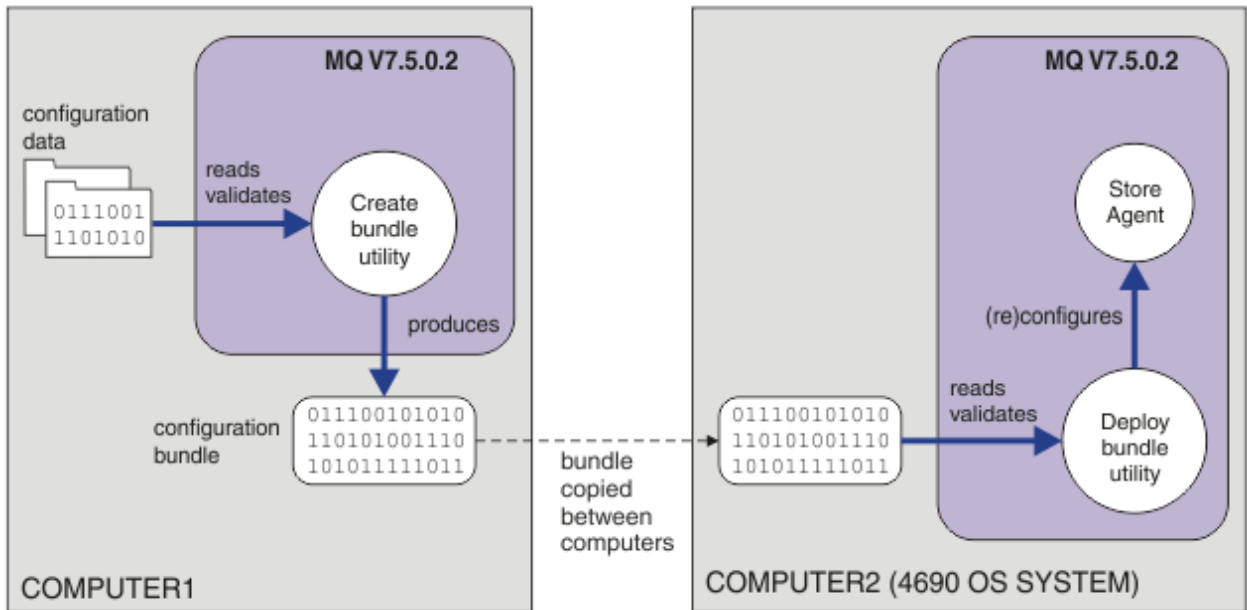
配置 4690 OS 受管文件传输代理与在 Windows 或 UNIX 上配置代理不同。

配置 4690 OS 代理是两个步骤便可以完成的任务：

1. 配置捆绑软件是使用 IBM MQ Managed File Transfer 随附的 **fteBundleConfiguration** 实用程序创建的。配置捆绑软件将配置 4690 OS 代理所需的所有信息都打包在一起。
2. 该配置捆绑软件将传输到 4690 OS 商店控制器，从此控制器中部署到 IBM MQ Managed File Transfer 安装。

选择此配置样式的原因如下：

- 将配置 4690 OS 代理网络所需要的 IBM MQ Managed File Transfer 知识集中在一起。例如，在零售场景中，这些技能可能集中在总部站点，而在个别零售商店中存在很少或者不存在 IT 技能。
- 提供了一种机制，在这种机制下，一个配置捆绑软件可以部署到多个 4690 OS 系统。该方法可以降低不同系统上所用配置之间不一致的几率。有关更多信息，请参阅第 48 页的『传输文件以验证该场景』。



该图显示了客户如何使用此场景创建和部署配置捆绑软件。要实现此场景，需完成以下步骤：

1. 在 COMPUTER1 上创建一系列配置数据。对于零售商，该操作通常在一个中心站点（如总部）发生。在此场景中，配置数据基于随 IBM MQ Managed File Transfer 提供的某个示例。
2. 在 COMPUTER1 上使用 **fteBundleConfiguration** 实用程序读取和验证配置数据。作为其输出，**fteBundleConfiguration** 实用程序会生成一个配置捆绑软件文件。
3. 配置捆绑软件文件将复制到 COMPUTER2 的文件系统中，这是一个 4690 OS 商店控制器系统。
4. **ftecfg** 命令用于在 4690 OS 系统上配置或重新配置 IBM MQ Managed File Transfer 安装。

本任务假设您使用的是 Windows 系统。对于 UNIX 系统，请替换相应的路径和命令，并确保您具有所有相关目录的读写访问权。本场景还假设您基本了解队列管理器的用途。

作为用户 mqmAdmin，请在 COMPUTER1 上执行下列操作：

过程

1. 创建操纵配置数据时要使用的临时目录。

```
mkdir %TEMP%\4690cfg
```

2. 使用 **fteBundleConfiguration** 命令解压样本配置捆绑软件。

```
fteBundleConfiguration -x MQ_INSTALL_PATH\mqft\samples\4690\basic.zip %TEMP%\4690cfg
```

3. 将目录更改为临时目录

```
cd %TEMP%\4690cfg
```

4. 编辑 `coordination.properties` 文件，使其包含以下属性：

```
coordinationQMgr=hoffQM
```

5. 对名称目录进行重命名以反映代理名称。

```
move name SAGENT
```

6. 编辑 `agent.properties` 文件（位于 `agents\names` 目录内），以使其包含以下六个属性：

```
agentName=SAGENT
agentQMgr=hoffQM
agentQMgrHost=host or ip address of COMPUTER1
agentQMgrPort=port number MQ is configured to listen on
agentQMgrChannel=FTE.AGENT.SVRCONN
authorityChecking=true
```

7. 使用 **fteBundleConfiguration** 命令创建配置捆绑软件。

```
fteBundleConfiguration s1cfg.zip %TEMP%\4690cfg
```

8. 通过使用通常用于向 4690 OS 传输文件的机制，将配置捆绑软件复制到 4690 OS 系统。确保文件作为 `C:\S1CFG.ZIP` 写入 `C:\` 驱动器的根目录。配置捆绑软件包含二进制数据。如果您要使用 FTP 协议传输此数据，请确保将配置捆绑软件作为二进制文件传输。
9. 可选：除去用于操纵配置数据的临时目录。

```
%TEMP% rmdir /s 4690cfg
```

相关概念

第 38 页的『[1. 入门：使用商店中的 4690 OS 传输文件](#)』

您可以使用本场景来帮助在 4690 OS 上开始使用 IBM MQ Managed File Transfer。

第 38 页的『[在零售环境中使用 IBM MQ Managed File Transfer](#)』

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先在非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

将代理部署到 4690 OS 系统

实施此场景所描述的解决方案需要在 4690 OS 上部署 IBM MQ Managed File Transfer 代理。通过将该代理配置为 4690 OS 后台应用程序，可启动该代理。

过程

在 COMPUTER2 的 IBM MQ Managed File Transfer bin 目录 `f:\adxetc\mft75\bin` 中，按照这些指示信息操作。

1. 使用 **ftecfig** 命令在 4690 OS 系统上创建或替换 IBM MQ Managed File Transfer 配置。

```
ftecfig C:\S1CFG.ZIP
```

有关 **ftecfig** 命令的更多信息，请参阅第 88 页的『[ftecfig（在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置）](#)』。关于配置过程的更多信息，请参阅第 66 页的『[在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置](#)』。

2. 创建 4690 OS 后台任务以启动该代理。有关启动代理的更多信息，请参阅第 78 页的『[在 4690 OS 系统上启动代理](#)』。

3. 重新 IPL 4690 OS 系统来启动该代理。

相关概念

第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先在非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

传输文件以验证该场景

通过将文件从 4690 OS 系统 (COMPUTER2) 传输回主机 4690 OS 系统，来验证在此场景中构建的拓扑。本任务假设您使用的是 Windows 系统。对于 UNIX 系统，请替换相应的路径和命令。

在 COMPUTER2（零售商店的 4690 OS 系统）上：

1. 创建源文件和目标目录：

- 创建要传输的样本文件，例如 C:\START\MYFILE.TXT
- 创建一个目录或确定一个现有目录作为文件传输的目标目录。例如，C:\END\

在 COMPUTER1 上（总部系统）：

1. 以 *ftuser* 用户身份输入 **fteCreateTransfer** 命令，启动从 C:\START\MYFILE.TXT 到 C:\END\MYFILE.TXT 的文件传输：

```
fteCreateTransfer -sa SAGENT -sm hoffQM -sd delete -da SAGENT -dm hoffQM -w -dd C:\END\  
C:\START\MYFILE.TXT
```

- -sa SAGENT 将源代理（即，文件从该代理中开始传输）定义为 SAGENT。
- -sm hoffQM 定义源代理 SAGENT 所连接的队列管理器。
- -sd delete 指定传输成功后删除源文件。
- -da SAGENT 将目标代理（即，文件将传输到的代理）定义为 SAGENT。
- -dm hoffQM 定义目标代理 SAGENT 所连接的队列管理器。
- -w 请求 **fteCreateTransfer** 命令等待确认传输成功。
- -dd C:\END\ 将目标目录定义为 C:\END\。
- C:\START\MYFILE.TXT 定义要传输的文件。

在 COMPUTER2（零售商店的 4690 OS 系统）上：

1. 检查样本文件是否在目录间成功移动：

- 检查样本文件是否不再存在于源目录中，例如 C:\START
- 检查样本文件是否存在于目标目录内，例如 C:\END\MYFILE.TXT

后续操作

- 扩展拓扑，以包含总部系统代理与此场景中创建的零售商店代理之间的传输。有关更多信息，请参阅第 49 页的『2. 将文件从总部传输到商店中的 4690 OS 系统』。
- 通过阅读第 99 页的『沙箱』、第 432 页的『特定于 IBM MQ Managed File Transfer 的资源权限』或第 438 页的『用户的 IBM MQ Managed File Transfer 操作权限』，扩展此场景中使用的安全模型。
- 进一步保护您的环境。您所需的访问模型可能不同于本场景中使用的访问模型。有关更多信息，请参阅“保护 WebSphere MQ File Transfer Edition V7”，网址为：https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html。
- 通过自动替换代理名称或代理配置值，创建可部署到多个 4690 OS 系统的单个配置捆绑软件。有关更多信息，请参阅第 67 页的『在 4690 OS 配置捆绑软件中定制代理名称』和第 69 页的『在 4690 OS 配置束中定制代理属性』。

相关概念

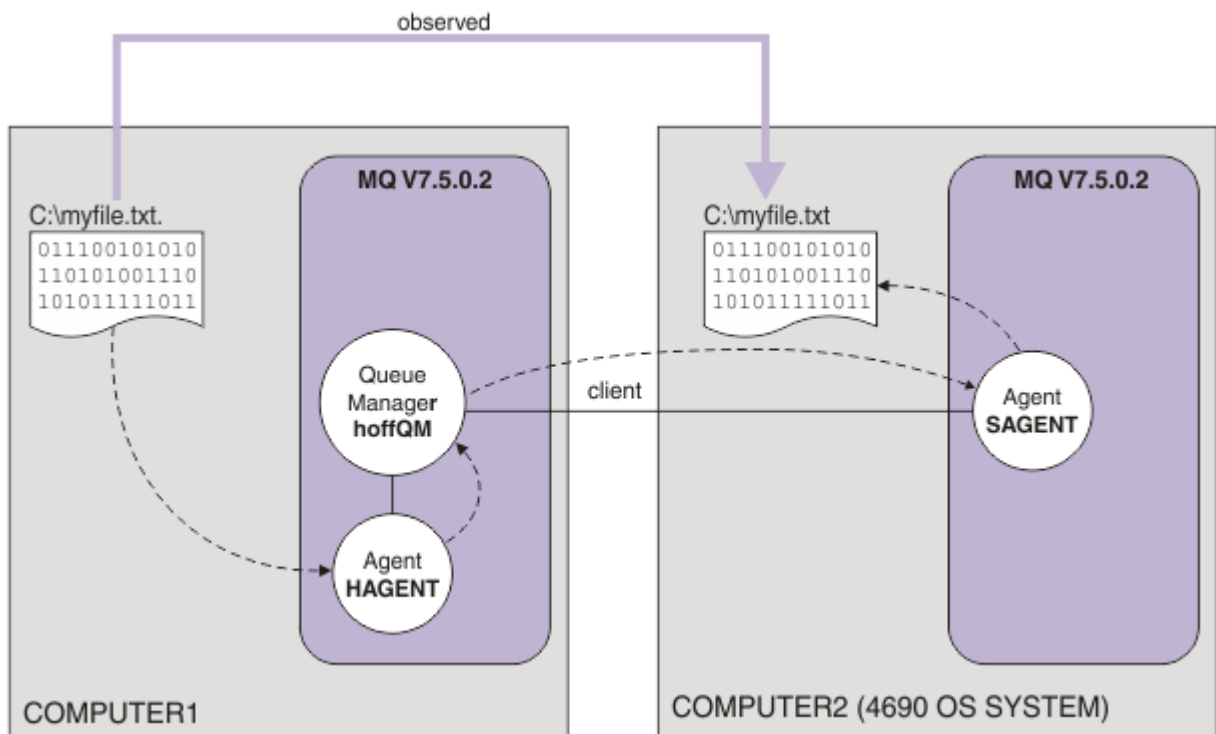
第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先在非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

2. 将文件从总部传输到商店中的 4690 OS 系统

本场景解释了如何使用 IBM MQ Managed File Transfer 将文件数据从总部站点发送到零售商店内的 4690 OS 商店控制器系统。

本场景基于先前的场景（第 38 页的『1. 入门：使用商店中的 4690 OS 传输文件』）而构建。请确保已完成该先前场景，然后才能开始本场景。



该图显示跨两台计算机的总部和商店拓扑。COMPUTER1 是总部计算机，在先前场景中创建了队列管理器 hoffQM。COMPUTER2 是 4690 OS 商店控制器，通常位于零售商店内。在该先前场景中，在 COMPUTER2 上创建了 IBM MQ Managed File Transfer 代理 SAGENT。本场景将在 COMPUTER1 上创建代理 HAGENT，并将文件从 COMPUTER1 传输到 COMPUTER2，如图所示。

要完成本场景，您必须具备 IBM MQ 和 IBM MQ Managed File Transfer 的基本知识。具体而言，必须熟悉以下概念：

- 队列管理器的概念
- IBM MQ 代理的概念
- IBM MQ Managed File Transfer 的基本配置和管理

有关 IBM MQ Managed File Transfer 功能的更多信息，请参阅第 5 页的『IBM MQ Managed File Transfer 介绍』。

相关概念

第 50 页的『在总部配置文件传输』

本场景扩展了第一个场景，进一步包含了从总部到商店中的 4690 OS 系统的文件传输。

第 52 页的『使用命令行传输文件』

您可以使用 IBM MQ Managed File Transfer 随附的命令行界面将文件从总部系统传输到零售商店中的 4690 OS 商店控制器系统。

第 52 页的『使用 IBM MQ 资源管理器 传输文件』

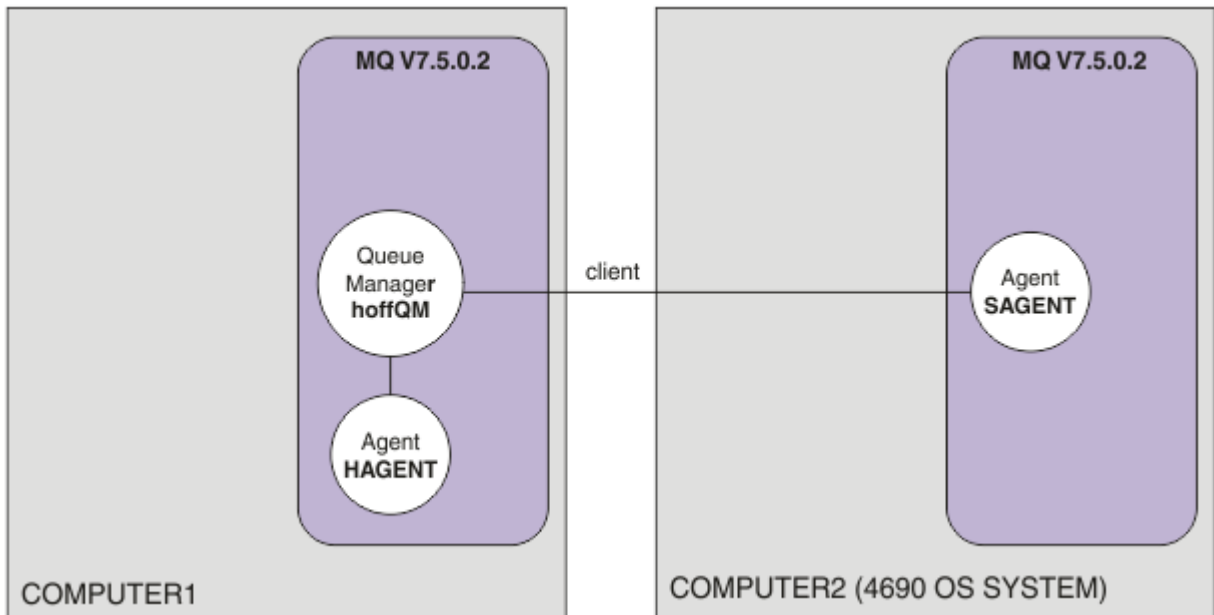
您可以使用 IBM MQ 资源管理器 将文件从总部系统传输到零售商店中的 4690 OS 商店控制器系统。

第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先在非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

在总部配置文件传输

本场景扩展了第一个场景，进一步包含了从总部到商店中的 4690 OS 系统的文件传输。



该图显示在本场景中创建的拓扑。在前一个场景中，在 COMPUTER1 上创建了队列管理器 hofQM，并在 COMPUTER2 上创建了代理 SAGENT。在完成本场景后，您可以完成以下任务：

- 在总部系统上定义 IBM MQ Managed File Transfer 代理
- 在总部系统上启动该代理
- 更新 FTUSERS 组成员具有的权限，以便仅允许从总部站点到零售商店站点的文件传输

本任务假设您使用的是 Windows 系统。对于 UNIX 系统，请替换相应的路径和命令。

先决条件

要完成此场景，您需要以下几项：

- COMPUTER1，部署在总部站点的计算机。
- COMPUTER2，部署在零售商店站点的 4690 OS 商店控制器。

这两台计算机都必须具有作为第 38 页的『1. 入门：使用商店中的 4690 OS 传输文件』的一部分创建的有效配置。

过程

以用户 mqmAdmin 身份，从 COMPUTER1 上的 IBM MQ bin 目录 (*MQ_INSTALL_ROOT\bin*) 完成以下步骤：

1. 创建名为 HAGENT 的 MQMFT 代理。

```
fteCreateAgent -agentName HAGENT -agentQMGr hoffQM -s -su fthoff -sp password for fthoff
```

该代理创建后将作为 Windows 服务启动。这意味着即使启动该代理的用户从系统注销，该代理仍会在 fthoff 用户帐户下继续运行。有关将代理作为 Windows 服务运行的更多信息，请参阅第 216 页的『将代理作为 Windows 服务启动』。如果使用 UNIX 系统作为 COMPUTER1（总部计算机），请参阅第 218 页的『在 UNIX 系统启动时启动代理』。

2. 编辑代理 HAGENT 的配置以启用对文件传输操作的用户权限的检查。

在 ..\mqft\config\hoffQM\agents\HAGENT\agent.properties 中的 agent.properties 文件中，添加以下条目：

```
authorityChecking=true
```

有关更多信息，请参阅第 438 页的『用户的 IBM MQ Managed File Transfer 操作权限』。

3. 使用 MQSC 接口来定义队列管理器 hoffQM 上的代理程序 HAGENT 所需的 IBM MQ 对象。

```
runmqsc hoffQM < ..\mqft\config\hoffQM\agents\HAGENT\HAGENT_create.mqsc
```

4. 确保 FTHOFFS， FTSTORES 和 FTUSER 组具有对属于代理程序 HAGENT 的 IBM MQ 对象的相应访问权。您可能需要根据自己的安全性需求来定制该配置。

```
setmqaut -m hoffQM -t qmgr -g FTHOFFS +connect +inq +setid +altusr
setmqaut -m hoffQM -n SYSTEM.FTE -t queue -g FTHOFFS +get +put
setmqaut -m hoffQM -n SYSTEM.FTE.COMMAND.HAGENT -t queue -g FTHOFFS +browse +get +put +setid
setmqaut -m hoffQM -n SYSTEM.FTE.DATA.HAGENT -t queue -g FTHOFFS +get +put
setmqaut -m hoffQM -n SYSTEM.FTE.EVENT.HAGENT -t queue -g FTHOFFS +browse +get +put
setmqaut -m hoffQM -n SYSTEM.FTE.REPLY.HAGENT -t queue -g FTHOFFS +browse +get +put
setmqaut -m hoffQM -n SYSTEM.FTE.STATE.HAGENT -t queue -g FTHOFFS +browse +get +inq +put
setmqaut -m hoffQM -n SYSTEM.FTE -t topic -g FTHOFFS +pub +sub
setmqaut -m hoffQM -n SYSTEM.DEFAULT.MODEL.QUEUE -t queue -g FTHOFFS +browse +dsp +get +put
setmqaut -m hoffQM -n SYSTEM.FTE.COMMAND.HAGENT -t queue -g FTSTORES +put
setmqaut -m hoffQM -n SYSTEM.FTE.DATA.HAGENT -t queue -g FTSTORES +put
setmqaut -m hoffQM -n SYSTEM.FTE.DATA.SAGENT -t queue -g FTHOFFS +put
setmqaut -m hoffQM -n SYSTEM.FTE.REPLY.HAGENT -t queue -g FTSTORES +put
setmqaut -m hoffQM -n SYSTEM.FTE.COMMAND.HAGENT -t queue -g FTUSERS +put
```

以下命令必须针对每个店内代理运行。

```
setmqaut -m hoffQM -n SYSTEM.FTE.COMMAND.SAGENT -t queue -g FTHOFFS +put
setmqaut -m hoffQM -n SYSTEM.FTE.DATA.SAGENT -t queue -g FTHOFFS +put
setmqaut -m hoffQM -n SYSTEM.FTE.REPLY.SAGENT -t queue -g FTHOFFS +put
```

5. 除去 FTUSERS 组成员的权限，使这些成员无法将文件从 4690 OS 系统传回其自身。

```
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHTRN1.SAGENT -t queue -g FTUSERS -browse -put
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHAGT1.SAGENT -t queue -g FTSTORES -browse -put
```

6. 向 FTUSERS 组成员授权，使其能够将文件从代理 HAGENT 传输到代理 SAGENT。

```
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHTRN1.HAGENT -t queue -g FTUSERS +browse
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHAGT1.HAGENT -t queue -g FTSTORES +browse
```

以下命令必须针对每个店内代理运行。

```
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHTRN1.SAGENT -t queue -g FTUSERS +put
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHAGT1.SAGENT -t queue -g FTHOFFS +put
```

7. 启动代理 HAGENT。

```
fteStartAgent HAGENT
```

相关概念

第 38 页的『[1. 入门：使用商店中的 4690 OS 传输文件](#)』

您可以使用本场景来帮助在 4690 OS 上开始使用 IBM MQ Managed File Transfer。

第 38 页的『[在零售环境中使用 IBM MQ Managed File Transfer](#)』

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先在非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

使用命令行传输文件

您可以使用 IBM MQ Managed File Transfer 随附的命令行界面将文件从总部系统传输到零售商店中的 4690 OS 商店控制器系统。

本任务假设您使用的是 Windows 系统。对于 UNIX 系统，请替换相应的路径和命令。

1. 在总部计算机上创建或确定可用于传输的适合文件。例如，C:\start\myfile.txt。
2. 在 4690 OS 商店控制器上创建或确定可将该文件复制到的适合目录。例如，C:\END\。
3. 在 COMPUTER1 上以用户 ftuser 身份输入以下 **fteCreateTransfer** 命令，以开始将文件从 C:\start\myfile.txt（位于 COMPUTER1 上）上传到 C:\END\MYFILE.TXT（位于 COMPUTER2 上）：

```
fteCreateTransfer -sa HAGENT -sm hoffQM -da SAGENT -dm hoffQM -w -dd C:\END\  
C:\start\myfile.txt
```

有关此命令的更多信息，请参阅第 497 页的『[fteCreateTransfer（创建新的文件传输）](#)』。

4. 确认该文件已复制到 4690 OS 商店控制器 (COMPUTER2)。

相关概念

第 38 页的『[1. 入门：使用商店中的 4690 OS 传输文件](#)』

您可以使用本场景来帮助在 4690 OS 上开始使用 IBM MQ Managed File Transfer。

第 38 页的『[在零售环境中使用 IBM MQ Managed File Transfer](#)』

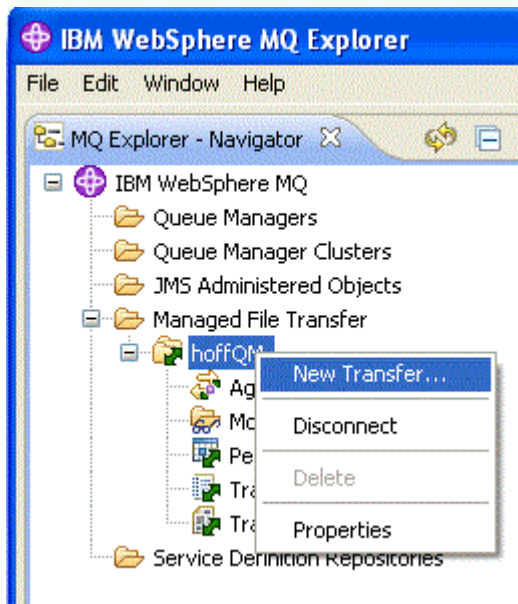
您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先在非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

使用 IBM MQ 资源管理器 传输文件

您可以使用 IBM MQ 资源管理器 将文件从总部系统传输到零售商店中的 4690 OS 商店控制器系统。

本任务假设您使用的是 Windows 系统。对于 UNIX 系统，请替换相应的路径和命令。

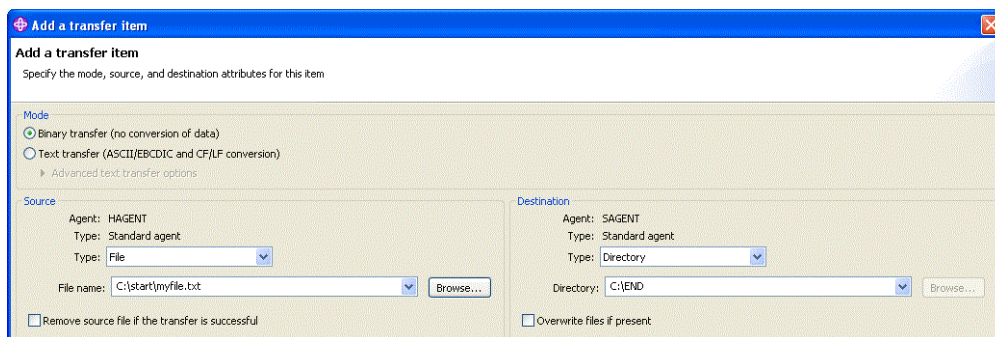
1. 确定或创建源文件和目标目录。
 - a. 在 COMPUTER1（位于总部的计算机）上确定或创建样本文件。例如：C:\start\myfile.txt
 - b. 在 COMPUTER2（4690 OS 商店控制器）上确定或创建目录。例如：C:\END\
2. 在 COMPUTER1 上以用户 ftuser 身份启动 IBM MQ 资源管理器。通过[开始菜单](#)（或等效操作）启动该程序，或运行 **strmqcfg** 命令。有关更多信息，请参阅[启动 IBM MQ Explorer](#)。
3. 在 IBM MQ 资源管理器 Navigator 中展开 **受管文件传输**，右键单击 hoffQM，然后选择 **新建传输** 以启动“新建传输向导”。



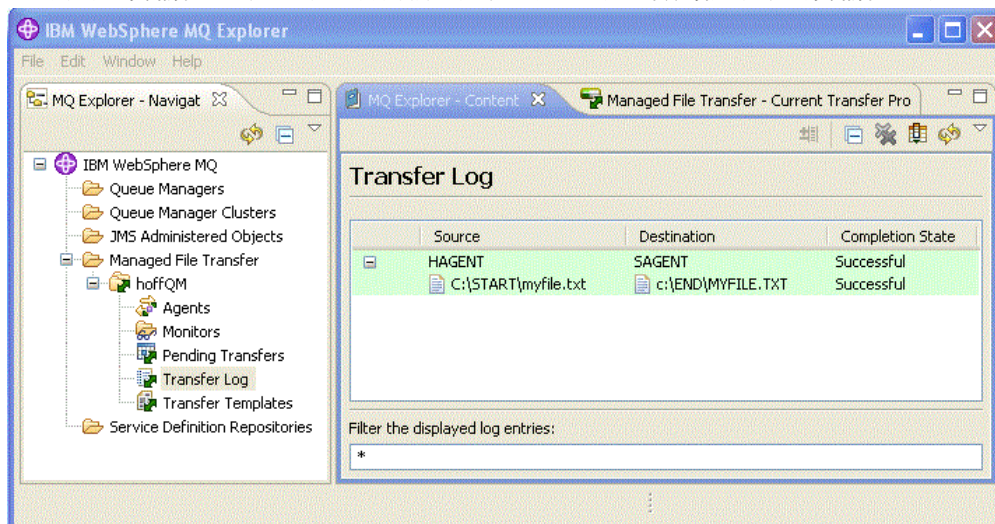
4. 选择 HAGENT 作为源代理，并选择 SAGENT 作为目标代理。单击下一步。

5. 单击添加。

6. 在“添加传输项目”面板上完成以下步骤：
- 对于源，在文件名字段中输入 C:\start\myfile.txt。
 - 对于目标，从类型列表中选择目录。
 - 对于目标，在目录字段中输入 C:\END\。单击确定。



7. 单击**完成**。这样就会开始传输。
8. 您可以在“**传输日志**”窗口中查看传输进度。可通过在“**导航器**”中单击**传输日志**来显示该视图。



9. 您还可以手动检查 COMPUTER2 的文件系统以确认新文件存在，例如：C:\END\MYFILE.TXT。

后续操作

您可以扩展自己的拓扑，以在零售商店中创建文件时自动将这些文件传输到总部系统。有关更多信息，请参阅第 54 页的『[3. 将文件从商店中的 4690 OS 系统传输到总部](#)』。

您可以进一步确保环境的安全。您所需的访问模型可能不同于本场景中使用的访问模型。有关这一方面的最佳实践的更多信息，请参阅 [Securing WebSphere MQ File Transfer Edition V7](#)。

相关概念

第 38 页的『[1. 入门：使用商店中的 4690 OS 传输文件](#)』

您可以使用本场景来帮助在 4690 OS 上开始使用 IBM MQ Managed File Transfer。

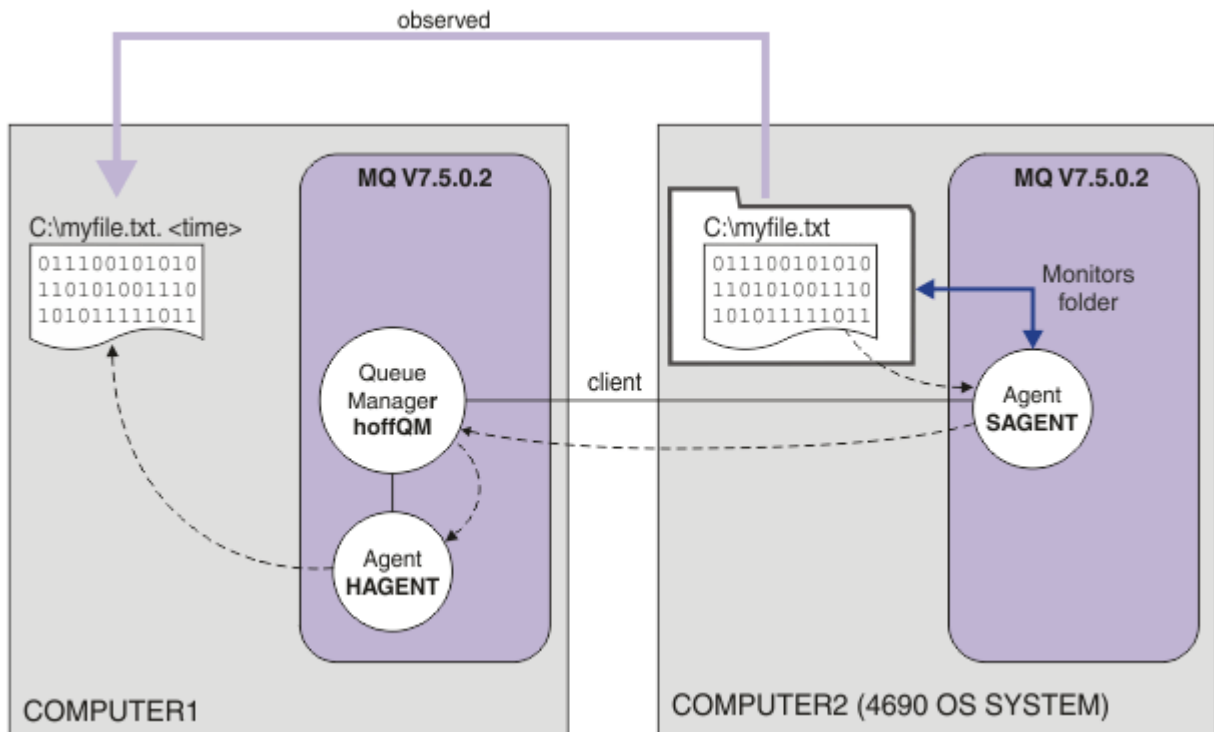
第 38 页的『[在零售环境中使用 IBM MQ Managed File Transfer](#)』

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先在非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

3. 将文件从商店中的 4690 OS 系统传输到总部

本场景解释了如何使用 IBM MQ Managed File Transfer 将文件数据从位于零售商店的 4690 操作系统商店控制器发送到总部站点。

本场景基于前两个场景而构建。有关更多信息，请参阅第 38 页的『[1. 入门：使用商店中的 4690 OS 传输文件](#)』和第 49 页的『[2. 将文件从总部传输到商店中的 4690 OS 系统](#)』。在开始本场景之前，必须已完成前两个场景。



该图显示作为本场景的一部分配置的文件传输操作。HAGENT 和 SAGENT 这两个代理以及队列管理器 hofQM 已在先前场景中进行了配置。

在本场景中，在 COMPUTER2 上配置代理 SAGENT 以监视 4690 OS 计算机文件系统上的某个目录。在文件到达此目录时，系统会将这些文件传输到 COMPUTER1 上的某个目录，并修改文件名以包含唯一的时间戳记。

要完成本场景，您需要具备 IBM MQ 和 IBM MQ Managed File Transfer 的基本知识。具体而言，应熟悉 IBM MQ 和 受管文件传输 的基本配置及管理、队列管理器的概念以及 受管文件传输 代理的概念。有关 受管文件传输 功能的更多信息，请参阅第 5 页的『IBM MQ Managed File Transfer 介绍』。

相关概念

第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

第 59 页的『通过将文件从 4690 OS 传输到总部来验证场景』

通过此场景，演示如何使用 IBM MQ Managed File Transfer 资源监视器将文件从零售商店内的 4690 OS 商店控制器传输到总部站点上的计算机。

相关任务

第 55 页的『在 4690 OS 上配置代理以监视目录』

一个常见文件传输需求是让 IBM MQ Managed File Transfer 监视目录并将找到的任何文件传输到另一个系统。通常，在文件到达目标系统时必须为其指定唯一名称，以避免出现重复文件名。例如，重复文件名可能会导致文件在得到处理之前被覆盖。

在 4690 OS 上配置代理以监视目录

一个常见文件传输需求是让 IBM MQ Managed File Transfer 监视目录并将找到的任何文件传输到另一个系统。通常，在文件到达目标系统时必须为其指定唯一名称，以避免出现重复文件名。例如，重复文件名可能会导致文件在得到处理之前被覆盖。

开始之前

这两台计算机都必须具有作为 [第 38 页的『1. 入门：使用商店中的 4690 OS 传输文件』](#) 的一部分创建的有效配置。

- COMPUTER1，部署在总部的计算机。
- COMPUTER2，部署在零售商店的 4690 OS 商店控制器。

关于此任务

在本场景中，您将完成以下任务：

- 更新 FTUSERS 组的权限，以便该组的成员能够针对代理 SAGENT 定义资源监视器，并将文件从 SAGENT 传输到 HAGENT。
- 创建所需的定义，以便代理 SAGENT 能够监视 4690 OS 商店控制器 COMPUTER2 的文件系统上的目录。到达此目录的文件将传输到总部的 COMPUTER1。

本任务假设您使用的是 Windows 系统。对于 UNIX 系统，请替换相应的路径和命令。

针对 SAGENT 运行的所有任务和命令必须针对每个商店中代理运行。

过程

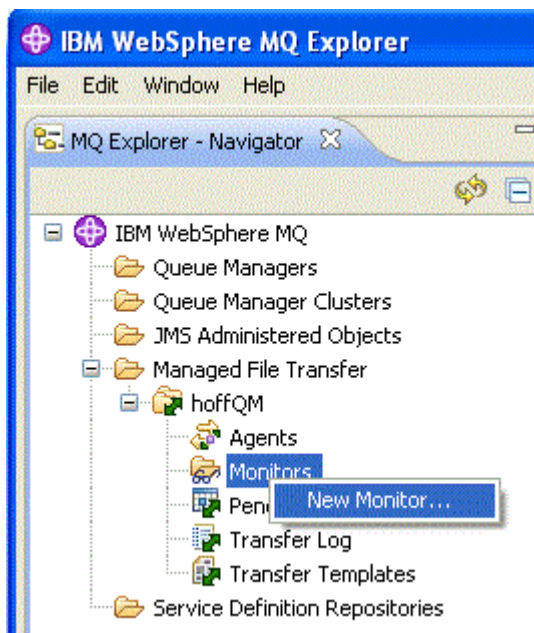
1. 确定或创建源和目标目录。
 - a) 在 4690 OS 商店控制器 COMPUTER2 上确定或创建目录。
例如，C:\MONITOR\
 - b) 在总部计算机 COMPUTER1 上确定或创建目录。
例如，C:\end\
2. 以用户 mqmAdmin 身份确保 FTUSERS 组成员具有在代理 SAGENT 上定义资源监视器的相应权限。

```
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHMON1.SAGENT -t queue -g FTUSERS +browse
```

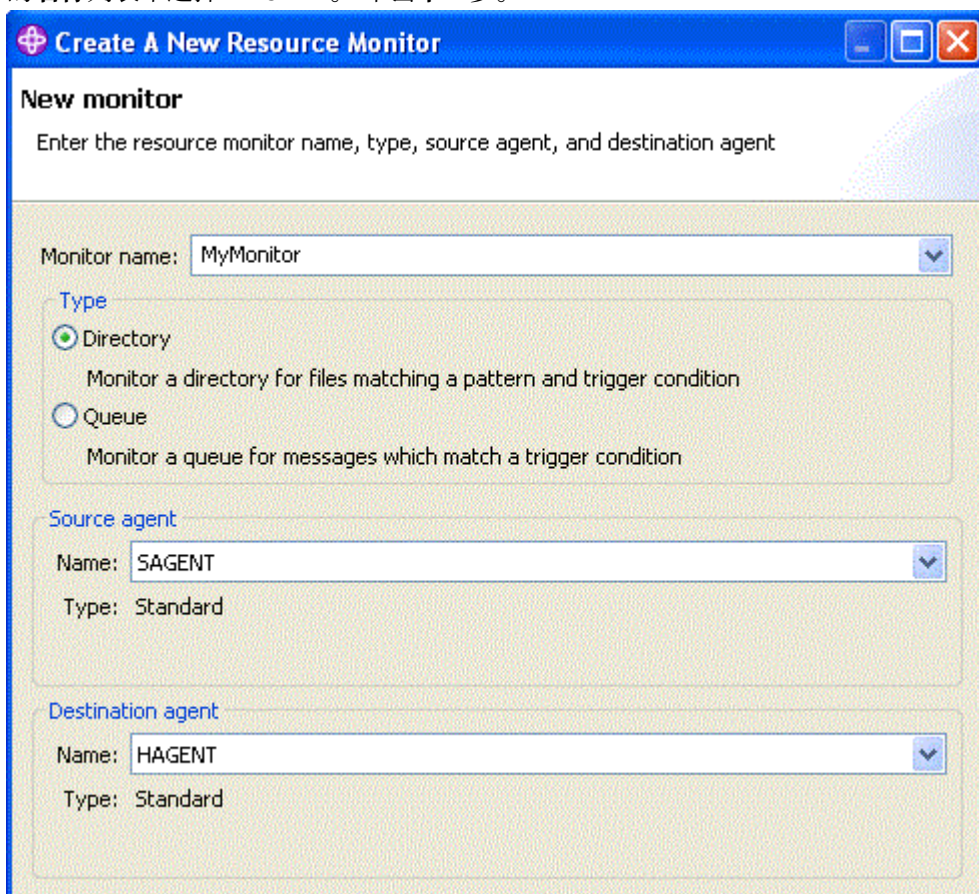
3. 以用户 mqmAdmin 身份确保组 FTHOFFS、FTSTORES 和 FTUSERS 具有相应权限以便可以将文件从代理 SAGENT 传输到代理 HAGENT。

```
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHTRN1.SAGENT -t queue -g FTUSERS +browse
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHTRN1.HAGENT -t queue -g FTUSERS +put
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHAGT1.SAGENT -t queue -g FTHOFFS +browse
setmqaut -m hoffQM -n SYSTEM.FTE.AUTHAGT1.HAGENT -t queue -g FTSTORES +put
```

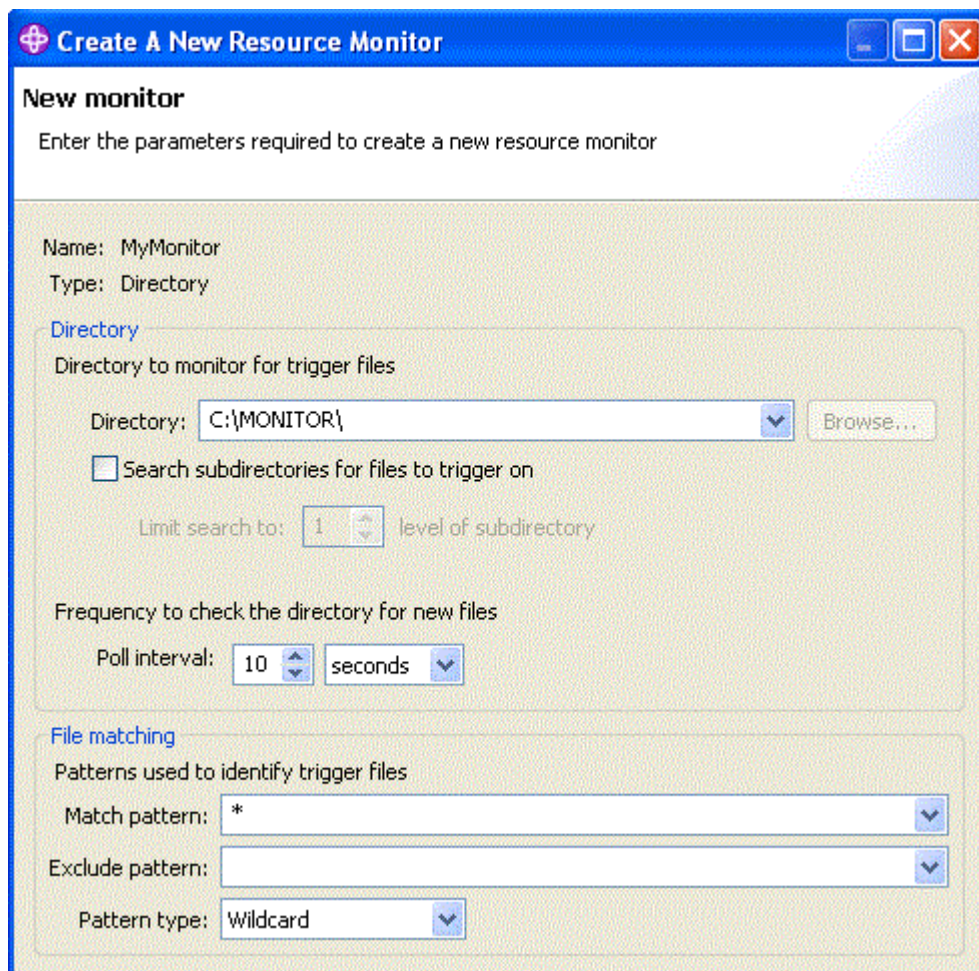
4. 以用户 ftuser 身份在 COMPUTER1 上启动 WebSphere MQ Explorer。通过**开始**菜单（或等效操作）启动该程序，或运行 **strmqcfcg** 命令。有关更多信息，请参阅[启动 WebSphere MQ Explorer](#)。
5. 在 IBM MQ 资源管理器 导航视图中单击 **受管文件传输**，右键单击 hoffQM 下的监视器，然后选择 **新建监视器** 以启动 **新建监视器向导**。



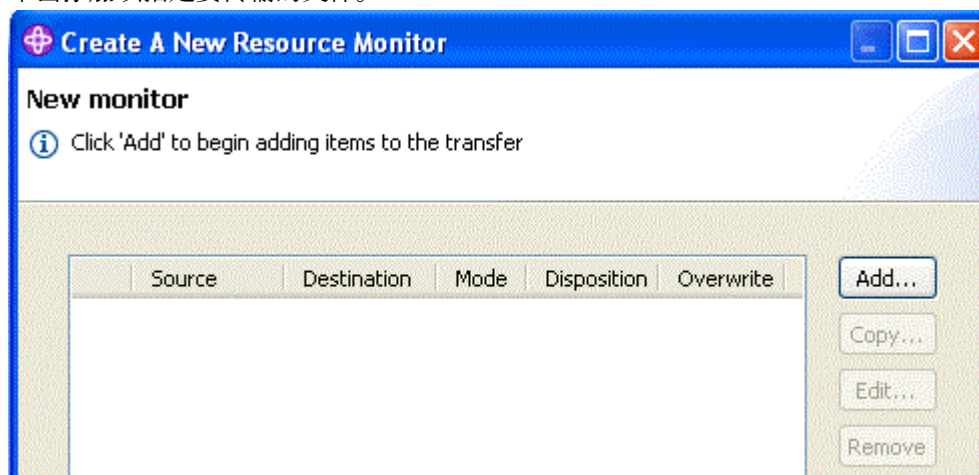
6. 在监视器名称字段中，输入监视器名称。在源代理窗格的名称列表中选择 SAGENT。在目标代理窗格的名称列表中选择 HAGENT。单击下一步。



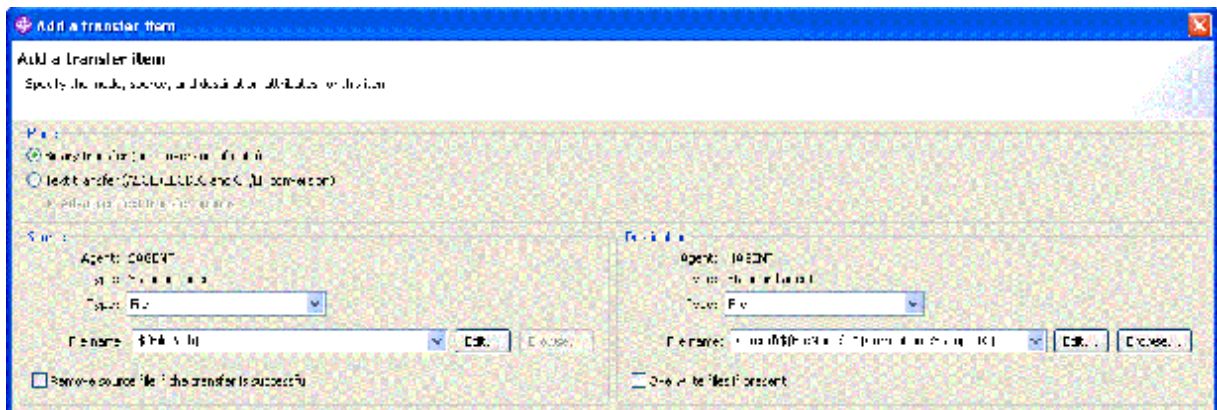
7. 在目录字段中输入 C:\MONITOR\。从轮询时间间隔列表中，选择 10 和秒。单击下一步。



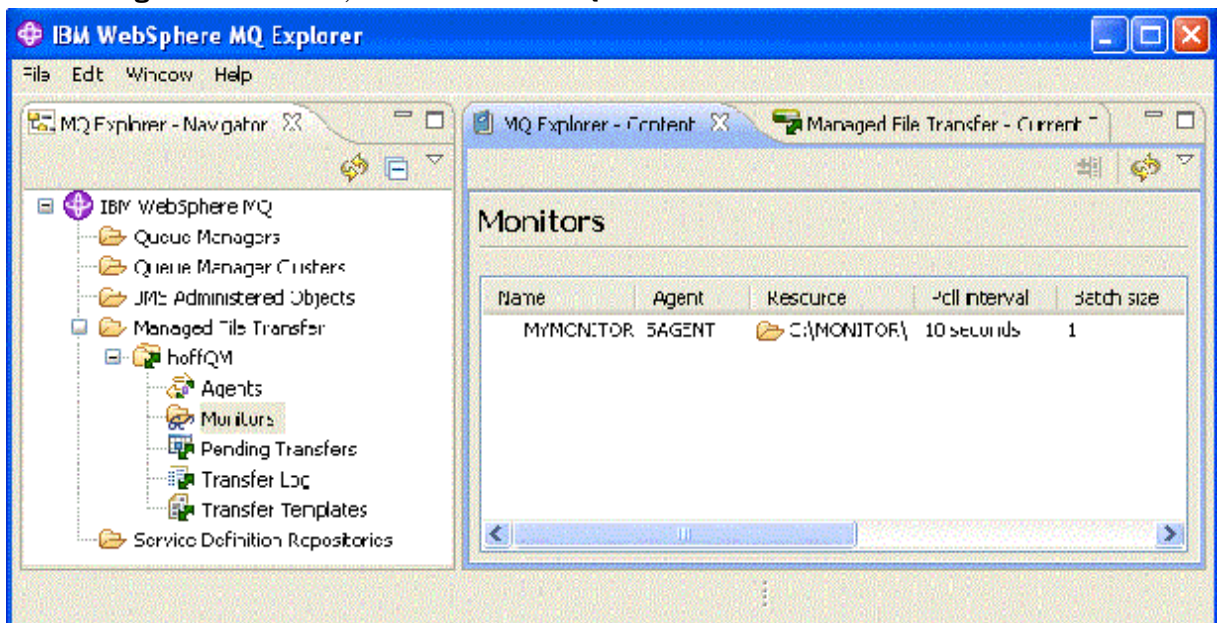
8. 在向导的下一页上单击**下一步**以接受“触发条件”的缺省值。
9. 单击**添加**以指定要传输的文件。



10. 在源窗格内的**文件名**字段中输入 `${FilePath}`。在目标窗格的**文件名**字段中输入 `C:\end\${FileName}.${CurrentTimeStampUTC}`。这些值包含传输开始时将被替换的变量，以便提供诸如资源监视器匹配的文件名之类的信息。有关变量替换的更多信息，请参阅：第 239 页的『使用变量替换定制 MFT 任务』。单击**确定**以完成对话框。



11. 单击**完成**以完成资源监视器的定义。
12. 要检查针对 IBM MQ Managed File Transfer 定义的资源监视器，请在 IBM MQ 资源管理器 导航视图中单击 **Managed File Transfer**，然后右键单击 hoffQM 下的“监视器”。



相关概念

第 38 页的『1. 入门：使用商店中的 4690 OS 传输文件』

您可以使用本场景来帮助在 4690 OS 上开始使用 IBM MQ Managed File Transfer。

第 49 页的『2. 将文件从总部传输到商店中的 4690 OS 系统』

本场景解释了如何使用 IBM MQ Managed File Transfer 将文件数据从总部站点发送到零售商店内的 4690 OS 商店控制器系统。

第 54 页的『3. 将文件从商店中的 4690 OS 系统传输到总部』

本场景解释了如何使用 IBM MQ Managed File Transfer 将文件数据从位于零售商店的 4690 操作系统商店控制器发送到总部站点。

通过将文件从 4690 OS 传输到总部来验证场景

通过此场景，演示如何使用 IBM MQ Managed File Transfer 资源监视器将文件从零售商店内的 4690 OS 商店控制器传输到总部站点上的计算机。

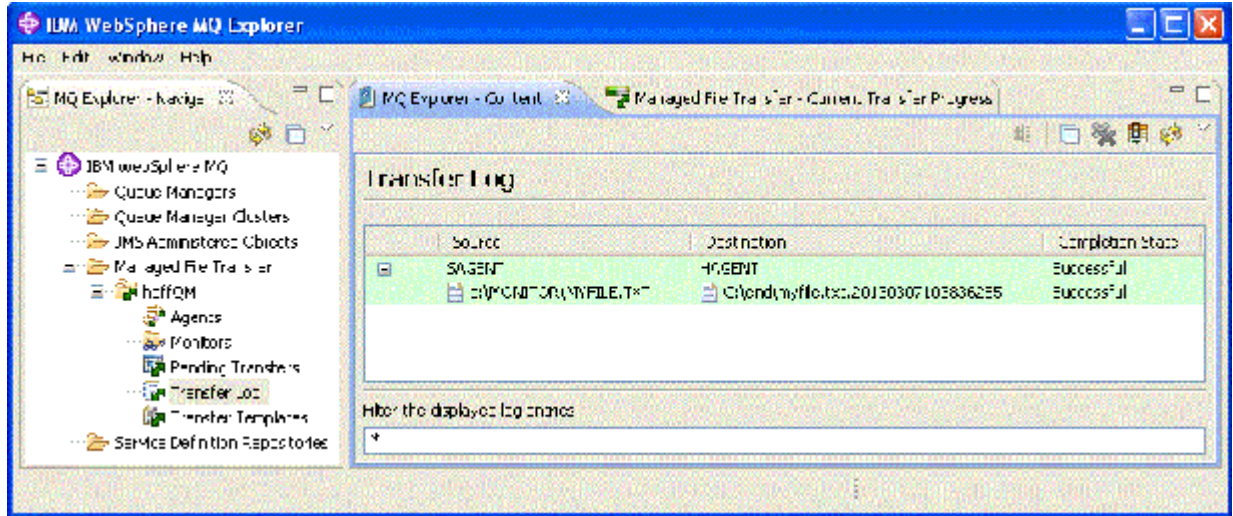
在之前的主题第 55 页的『在 4690 OS 上配置代理以监视目录』中，您已配置零售商店代理来监视目录。文件放置到该目录后，它们将被传输到总部计算机。这些步骤通过在受监视目录中创建一个文件，并检查此文件是否传输到总部计算机，验证拓扑是否正确工作。

1. 在 COMPUTER2 的受监视目录中创建一个文件，例如 C:\MONITOR\MYFILE.TXT。

```
echo A big hello to everyone at head-office > C:\MONITOR\MYFILE.TXT
```

4690 OS 系统 (SAGENT) 上的代理会自动将此文件传输给总部计算机。

2. 要查看有关文件传输（从在受监视目录中创建文件开始）操作的信息，请单击 IBM MQ 资源管理器 导航视图中的受管文件传输，展开 hoffQM，然后选择传输日志。



3. 您还可以手动检测 COMPUTER1 的文件系统，以确认新文件存在，例如：
C:\end\myfile.txt.20130307103836255

后续操作

- 要了解从命令行创建资源监视器定义，请参阅第 229 页的『资源监控』。
- 要了解 IBM MQ Managed File Transfer 如何将受管文件传输活动记录到数据库或文件系统，请参阅第 151 页的『配置 Managed File Transfer 记录器』。
- 要了解 IBM MQ Managed File Transfer 的更多常规功能，请参阅第 5 页的『IBM MQ Managed File Transfer 介绍』。

相关概念

第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。要完成此类型的安装，必须先在非 4690 机器上安装 IBM MQ Managed File Transfer。然后，您可以收集非 4690 机器上的安装和配置文件，并将它们传输到 IBM 4690 机器上。您可以使用这些文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

准备在 IBM 4690 系统上安装 IBM MQ Managed File Transfer

您可以在正在运行 IBM 4690 操作系统的设备上安装 IBM MQ Managed File Transfer。通过以下两个阶段完成此类型安装。第一阶段是在非 4690 机器上完成，包括收集配置文件（安装后需要）以及将配置文件和安装 .zip 文件传输至 IBM 4690 机器上。第二个阶段使用安装 .zip 文件在 IBM 4690 机器上安装 IBM MQ Managed File Transfer。

阶段 1。收集安装和配置文件

在未运行 IBM 4690 操作系统的机器上完成此阶段。收集安装和配置所需的文件并将它们传输至 IBM 4690 机器上。

完成以下步骤：

1. 创建一个包含您要在安装过程中使用的配置的 .zip 文件，例如 config.zip。有关创建此配置的更多信息，请参阅：第 66 页的『在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置』。

2. 确保具有安装 .zip 文件: MQMFT75.ZIP。

IBM MQ Managed File Transfer 的 MQMFT75.ZIP 安装文件在专门针对 4690 OS 的单独 DVD 上提供, 或者作为单独的 eImage(专门针对 4690 OS) 提供。您可从 [Passport Advantage](#) 和 [Passport Advantage Express Web 站点](#) 下载电子映像。

3. 通过使用 FTP 或其他方法将 MQMFT75.ZIP 文件和 config.zip 文件传输至 IBM 4690 系统 f: 盘的 root 目录中。

阶段 2。安装 IBM MQ Managed File Transfer

使用第 61 页的『在 4690 OS 上安装 IBM MQ Managed File Transfer』中的步骤在 IBM 4690 系统上完成此步骤。

相关概念

第 77 页的『在主/备份 4690 OS 控制器设置中配置 IBM MQ Managed File Transfer』

您可以在主备份 4690 OS 控制器设置中配置代理以提供故障容错。具有相同名称的代理可以配置为在零售环境中的多个控制器上运行。但是, 在任何一个时候都只能运行其中一个代理。该配置可以与配置为在不同环境中停止和启动的应用程序结合使用。

第 78 页的『在 4690 OS 控制器设置中配置多个 IBM MQ Managed File Transfer 代理』

您可以在单个商店控制器环境中或在商店内有多个商店控制器的环境中, 配置多个代理。

在 4690 OS 上安装 IBM MQ Managed File Transfer

使用 MQMFT75.ZIP 文件在 4690 OS 上安装 IBM MQ Managed File Transfer。

开始之前

- 确保您已在非 4690 系统上创建一个可在安装 MQMFT 后使用的 IBM MQ Managed File Transfer 配置。有关更多信息, 请参阅第 66 页的『在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置』
- IBM MQ Managed File Transfer 的 MQMFT75.ZIP 安装文件在专门针对 4690 OS 的单独 DVD 上提供, 或者作为单独的 eImage(专门针对 4690 OS) 提供。您可从 [Passport Advantage](#) 和 [Passport Advantage Express Web 站点](#) 下载电子映像。

关于此任务

过程

1. 将安装 .zip 文件 MQMFT75.ZIP 传输到 4690 OS 存储控制器上 F: 驱动器的根目录。
2. 登录并转至**命令模式** (系统主菜单上的选项 7)。
3. 检查是否已安装 IBM MQ Managed File Transfer。请求 F: 盘的目录列表, 以查看安装过程创建的目录结构是否存在。输入以下命令:

```
dir f:\adxetc\mft75
```

如果响应类似于以下输出, 那么表示未安装 MQMFT:

```
Volume in drive vf: has no label  
Directory of vf:adxetc/
```

```
0 Files      6753900 KB free
```

如果响应类似于以下输出, 那么表示已安装 MQMFT:


```
Volume in drive vf: has no label
Directory of vf:adxetc/mft75/
11-06-2012  11:00a  <DIR>      .
11-06-2012  11:00a  <DIR>      bin
11-06-2012  10:58a  <DIR>      ..
11-06-2012  11:00a  <DIR>      mqft
               4 Files        6715292 KB free
```

如果已安装产品，那么无需其他安装步骤。

4. 通过输入以下命令切换至 F: 盘的根目录：

```
f:
cd \
```

5. 通过运行以下命令抽取产品安装文件：

```
adxnszzl -xo f:\MQMFT75.ZIP
```

A successful extraction of the product files produces output similar to the following:

```
ADXNSZZL - Version 3.3.0 - Jan 06 2022 17:44:03
  tool code Copyright (c) 2003-2023 IBM - All Rights Reserved
  zlib code Copyright (c) 1995-2023 Jean-loup Gailly and Mark Adler

Archive: f:/MQMFT75.ZIP
Extracting: f:/adxetc/mft75/bin (0 bytes)...Done
Inflating : f:/adxetc/mft75/bin/fteRAS.bat (974 bytes)...Done
Inflating : f:/adxetc/mft75/bin/ftecfig.bat (993 bytes)...Done
Inflating : f:/adxetc/mft75/bin/ftediag.bat (988 bytes)...Done
Extracting: f:/adxetc/mft75/mqft (0 bytes)...Done
Extracting: f:/adxetc/mft75/mqft/lib (0 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/agenttype.properties (32 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/com.ibm.wmqfte.agent.jar (1682543 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/com.ibm.wmqfte.bootstrap.jar (33376 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/com.ibm.wmqfte.cmdline.jar (1556790 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/com.ibm.wmqfte.common.jar (5371185 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/com.ibm.wmqfte.daemon.jar (75261 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/com.ibm.wmqfte.embedded.agent.jar (190744 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/com.ibm.wmqfte.exitroutines.api.jar (241582 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/com.ibm.wmqfte.native.jni.jar (3466 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/commons-beanutils.jar (188671 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/commons-digester-1.8.jar (143602 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/commons-io-1.4.jar (109043 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/commons-lang-2.4.jar (261809 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/commons-logging-1.1.1.jar (60841 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/commons-net-2.0.jar (197316 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/logging.properties (802 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/template.pc (374 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/template.rsp (452 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/com.ibm.mq.headers.jar (269365 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/com.ibm.mq.jar (429548 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/com.ibm.mq.jmqi.jar (2754010 bytes)...Done
Inflating : f:/adxetc/mft75/mqft/lib/com.ibm.mqjms.jar (3053199 bytes)...Done
```

未成功抽取产品文件会生成最后一行与以下类似的输出：

```
ADXNSZZL - Exiting with rc=0x1F
```

如果抽取未成功，请使用以下步骤对问题进行故障诊断：

- a. 确保正确输入了用于抽取产品文件的命令。
 - b. 检查是否已将 MQMFT75.ZIP 文件正确复制到 F:\ 并且文件名完全是大写字母。
 - c. 将 MQMFT75.ZIP 文件再次复制到 4690 OS 系统。如果您使用 FTP 协议传输此文件，请确保其作为二进制文件进行传输。
 - d. 如果这些步骤没有解决问题，请与 IBM 支持中心联系。
6. 通过运行以下命令验证是否成功抽取产品文件：


```
dir f:\adxetc\mft75
```

如果响应与以下输出类似，那么安装了 MQMFT 代理：

```
Volume in drive vf: has no label
Directory of vf:adxetc\mft75/
11-06-2012  11:00a  <DIR>      .
11-06-2012  11:00a  <DIR>      bin
11-06-2012  10:58a  <DIR>      ..
11-06-2012  11:00a  <DIR>      mqft
               4 Files        6715292 KB free
```

如果响应与以下输出类似，那么未安装 MQMFT 代理。

```
Volume in drive vf: has no label
Directory of vf:adxetc
               0 Files        6753900 KB free
```

如果产品未显示已安装，那么再次小心尝试之前的每个步骤。如果重复这些步骤没有解决问题，请与 IBM 支持中心联系。

7. 阅读产品许可证并选择接受或拒绝该许可证。运行下列命令：

```
cd \adxetc\mft75\bin
ftelap
```

此命令会显示 MQMFT 的许可证并提示您接受或拒绝许可条款。有关 **ftelap** 命令的更多信息，请参阅第 87 页的『[ftelap（在 IBM MQ Managed File Transfer 安装期间接受许可协议）](#)』。

如果您选择不接受许可证，那么请完成在第 65 页的『[卸载 4690 系统中的 IBM MQ Managed File Transfer](#)』过程中除去 4690 OS 系统中 MQMFT 程序文件的步骤。

8. 检查产品是否正确安装。安装完成后，产品将安装在 f:\adxetc\mft75 目录中。

下一步做什么

完成安装后，您必须提供具有配置的 4690 OS。有关更多信息，请参阅第 88 页的『[ftecfg（在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置）](#)』。

在 4690 OS 上安装 IBM MQ Managed File Transfer 的修订包

使用 **FIXPACK.ZIP** 文件，对 4690 OS 上的 IBM MQ Managed File Transfer 安装应用修订包。

开始之前

- 确保正在应用修订包的 4690 OS 系统上已安装和配置 IBM MQ Managed File Transfer。

关于此任务

过程

1. 将修订包 .zip 文件 **FIXPACK.ZIP** 传输到 4690 OS 存储控制器上 f: 驱动器的根目录。
2. 停止任何正在运行的进程控制器和代理进程。通过访问后台应用程序菜单、停止所有进程控制器应用程序，然后停止所有代理应用程序，完成此任务。请在停止代理应用程序之前停止进程控制器应用程序，因为进程控制器应用程序会重新启动代理应用程序。可通过以 @f:/adxetc/mft75/ 开头并以 .pc 文件后缀结束的参数列表来识别进程控制器应用程序。可通过以 @f:/adxetc/mft75/ 开头并以 .rsp 文件后缀结束的参数列表来识别代理应用程序。
3. 创建当前安装的备份。如果遇到问题，此备份可用于回滚修订包应用程序。输入以下命令：

```
dir f:
```

```
cd \adxetc
adxnszzl -r -c MFTBACKUP.ZIP mft75\*
```

4. 通过解压 *FIXPACK.ZIP* 的内容来应用修订包。输入以下命令：

```
f:
cd \
adxnszzl -xo FIXPACK.ZIP
```

5. 对商店控制器执行 IPL 操作。

6. 通过检查代理的日志文件，验证代理是否已正确启动。输入以下命令：

```
f:
cd \adxetc\mft75\mqft\logs\coord_qm\agents\agent_name\logs\
type output0.log
```

7. 注：如果需要回滚修订包应用程序。请完成下列任务：

- 停止任何正在运行的进程控制器和代理进程。与步骤 2 一样。
- 解压缩在步骤 3 中创建的备份。输入以下命令：

```
f:
cd \adxetc
adxnszzl -xo MFTBACKUP.ZIP
```

- 对商店控制器执行 IPL 操作。

相关任务

第 61 页的『在 4690 OS 上安装 IBM MQ Managed File Transfer』
使用 MQMFT75.ZIP 文件在 4690 OS 上安装 IBM MQ Managed File Transfer。

第 66 页的『在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置』
要在 IBM 4690 系统上创建或替换 IBM MQ Managed File Transfer 配置，应先在非 4690 平台上创建配置。然后，将配置以压缩文件格式复制到 IBM 4690 系统上，并运行 **ftecfg** 命令以将配置部署到 IBM 4690 系统。

IBM MQ Managed File Transfer 上 IBM 4690 安装脚本的示例

如果您要在多个 4690 系统上安装 IBM MQ Managed File Transfer，那么可以编写安装脚本来帮助完成安装。您可以使用以下示例开始编写自己的安装脚本。

```
ECHO OFF

ECHO +-----+
ECHO +          IBM WebSphere MQ Managed File Transfer V7.5 Installation          +
ECHO +-----+
ECHO +-----+ > F:/install.log
ECHO +          IBM WebSphere MQ Managed File Transfer V7.5 Installation          + >> F:/install.log
ECHO +-----+ >> F:/install.log

REM Extract the contents of the product zip file into the F:/adxetc/mft75
REM directory.
ECHO + Installing MQMFT... +
ECHO + Installing MQMFT... + >> F:/install.log
adxnszzl -xo F:/MQMFT75.ZIP >> F:/install.log >>> F:/install.log
IF NOT ERRORLEVEL 0 GOTO EXTRACTFAIL

REM If you want to automatically accept the product license as part of
REM your installation, you can do so by using the supplied ftelap tool.
REM *****
REM * NOTE: Uncomment the following line only after you have read the *
REM *          product license and have fully accepted its terms. *
REM *****
REM COMMAND -C F:/adxetc/mft75/bin/ftelap -accept >> F:/install.log >>> F:/install.log
```

```

REM After the product files are extracted and the license is accepted, the
REM ftecfg command can be run to lay down a configuration and generate
REM all the files necessary to start an agent.
COMMAND -C F:/adxetc/mft75/bin/ftecfg F:/MFT75CFG.zip >> F:/install.log >>* F:/install.log
IF NOT ERRORLEVEL 0 GOTO CFGFAIL

REM Copy the install log file to the product directory
COPY F:/install.log F:/adxetc/mft75/install.log
DEL F:/install.log

ECHO +-----+
ECHO + Installation complete. +
ECHO + Product files can be found in the f:/adxetc/mft75 directory +
ECHO + Installation log written to file: f:/adxetc/mft75/install.log +
ECHO +-----+
ECHO +-----+ >> F:/adxetc/mft75/install.log
ECHO + Installation complete. + >> F:/adxetc/mft75/install.log
ECHO + Product files can be found in the f:/adxetc/mft75 directory + >> F:/adxetc/mft75/install.log
ECHO + Installation log written to file: f:/adxetc/mft75/install.log + >> F:/adxetc/mft75/install.log
ECHO +-----+ >> F:/adxetc/mft75/install.log

GOTO END

REM Log that the extract of the product files failed
:EXTRACTFAIL
ECHO + ERROR: Problem occurred extracting install files. See previous entries +
ECHO + in the install log for more details. +
ECHO +-----+
ECHO + ERROR: Problem occurred extracting install files. See previous entries + >> F:/fteinst.log
ECHO + in the install log for more details. + >> F:/fteinst.log
ECHO +-----+ >> F:/fteinst.log
GOTO FAIL

REM Log that the configuration command failed
:CFGFAIL
ECHO + ERROR: Problem occurred processing the supplied configuration zip +
ECHO + file. See previous entries in the install log for more details. +
ECHO +-----+
ECHO + ERROR: Problem occurred processing the supplied configuration zip + >> F:/install.log
ECHO + file. See previous entries in the install log for more details. + >> F:/install.log
ECHO +-----+ >> F:/install.log
GOTO FAIL

:FAIL
ECHO +-----+
ECHO + Installation failed, exiting. +
ECHO + Installation log written to file: f:/install.log +
ECHO +-----+
ECHO +-----+ >> F:/install.log
ECHO + Installation failed, exiting. + >> F:/install.log
ECHO + Installation log written to file: f:/install.log + >> F:/install.log
ECHO +-----+ >> F:/install.log
GOTO END

:END

```

相关任务

第 61 页的『在 4690 OS 上安装 IBM MQ Managed File Transfer』
使用 MQMFT75.ZIP 文件在 4690 OS 上安装 IBM MQ Managed File Transfer。

卸载 4690 系统中的 IBM MQ Managed File Transfer

要从 4690 存储控制器卸载 IBM MQ Managed File Transfer 代理，请完成以下步骤：

过程

1. 除去任何现有 IBM MQ Managed File Transfer 后台应用程序定义。浏览至定义后台应用程序屏幕，通过搜索以以下内容作为前缀的参数列表来标识这些定义：

```
@f:\adxetc\mft75\
```

2. 重新 IPL 存储控制器。
3. 运行 **uninstall** 命令以除去 IBM MQ Managed File Transfer 产品文件。将 **-a** 参数或 **-c** 参数与命令一起使用。

-a

卸载所有产品文件、配置文件和日志文件

-c

卸载所有产品文件，但保留配置文件和日志文件

例如，要卸载所有产品文件、配置文件和日志文件，请输入以下命令：

```
f:  
cd \adxetc\mft75  
uninstall -a
```

有关更多信息，请参阅第 94 页的『卸载（从 IBM 4690 系统卸载 IBM MQ Managed File Transfer）』。

4. 查看 `uninstalln.log` 文件（其中，*n* 是从零开始的数字）以确保在没有错误的情况下完成卸载。例如，`uninstall0.log`。该文件位于 `f:\adxetc\mft75`。
5. 除去卸载文件及其日志文件。例如：

```
del uninstall.bat  
del uninstall0.log
```

6. 可选：如果指定 `uninstall -a` 以除去所有产品、配置和日志文件，那么您还可以除去 `mft75` 目录。例如：

```
f:  
cd \adxetc  
rmdir mft75
```

在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置

要在 IBM 4690 系统上创建或替换 IBM MQ Managed File Transfer 配置，应先在非 4690 平台上创建配置。然后，将配置以压缩文件格式复制到 IBM 4690 系统上，并运行 **ftecfig** 命令以将配置部署到 IBM 4690 系统。

过程

要在 IBM 4690 系统上创建或替换 IBM MQ Managed File Transfer 配置，请完成以下步骤：

1. 通过运行 **fteSetupCoordination**、**fteSetupCommands** 和 **fteCreateAgent** 命令，在非 4690 平台上创建配置。
在 IBM 4690 系统上只能定义一个协调队列管理器。您可以配置多个代理，但代理名称最多只能有 23 个字符。4690 系统始终通过客户端连接至队列管理器，因此您必须通过指定客户端连接来创建 IBM MQ Managed File Transfer 组件。
2. 通过运行由 **fteSetupCoordination** 和 **fteCreateAgent** 命令生成的 MQSC 脚本完成配置设置。如果使用变量替换，请运行已使用 **fteDefine** 命令生成的代理 MQSC 脚本。
3. 通过使用 **fteBundleConfiguration** 命令创建一个在 `MQ_DATA_PATH/mqft/config` 目录下包含 `coordination.properties` 文件和 `agents` 目录的 `config.zip` 文件。请参阅第 85 页的『**fteBundleConfiguration**（创建 IBM MQ Managed File Transfer IBM 4690 代理配置 .zip 文件）』，以获取关于使用命令的详细信息。请参阅第 67 页的『IBM 4690 IBM MQ Managed File Transfer 配置压缩文件的结构』，以获取所需文件结构的详细信息。
4. 将 `config.zip` 文件复制到 IBM 4690 系统。

5. 在 IBM 4690 系统上，运行 **ftecfg** 命令并将路径作为参数传递至 config.zip 文件。

例如：

```
ftecfg f:\config.zip
```

如果存在现有配置，将先备份该配置，然后通过 **ftecfg** 命令删除该配置。有关更多信息，请参阅 [ftecfg](#)。

IBM 4690 IBM MQ Managed File Transfer 配置压缩文件的结构

您可以通过将包含配置详细信息的压缩文件作为参数传递至 **ftecfg** 命令，从而在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置。

压缩文件结构

压缩文件必须具有以下结构的 .zip 文件：

```
coordination.properties (properties file - the file must be populated with
                        the properties required to connect to the
                        coordination queue manager)

agents (directory)
    AgentName1 (directory - the name of the first agent, further agents can
                be defined if required)
    .
    .
    agent.properties (properties file - the file must be populated with
                      the properties required by the agent)
    .
    .
```

您可以通过压缩在非 4690 系统上创建的 IBM MQ Managed File Transfer 配置目录来创建文件。有关如何创建压缩文件的更多信息，请参阅第 85 页的『[fteBundleConfiguration（创建 IBM MQ Managed File Transfer IBM 4690 代理配置 .zip 文件）](#)』。

示例

```
coordination.properties
agents
  MyFirstAgent
    agent.properties
  MySecondAgent
    agent.properties
```

相关参考

[ftecfg](#)

ftecfg 命令在 IBM 4690 系统上配置一个或多个 IBM MQ Managed File Transfer 代理程序。

[fteBundleConfiguration](#)

使用 **fteBundleConfiguration** 命令可将指定目录中的 IBM MQ Managed File Transfer 配置树捆绑成 .zip 文件。

在 4690 OS 配置捆绑软件中定制代理名称

您可以将相同配置捆绑软件部署到多个不同的 4690 OS 商店控制器。要允许定制捆绑软件中的代理名称以与要将它们部署到的 4690 OS 商店控制器相匹配，可以使用基于商店编号和节点标识（有时称为商店控制器标识）的变量替换。

定制代理名称在以下示例中有用：

- 如果您要将相同配置部署到多个商店，那么可以将商店编号嵌入到商店的代理名称中，以创建唯一代理名称。
- 如果您在一个或多个商店中部署多个控制器功能 (MCF) 网络并且要查找特定商店控制器，那么可以将商店控制器的节点标识嵌入到代理名称中。

要将变量替换用于代理名称，必须使用 `@S` 和选择使用 `@N`，如下所示：

- 使用 `@S` 以表示对于商店控制器唯一的四位数商店编号。例如，1234。
- 使用 `@N` 以表示用于标识 MCF 网络中商店控制器的两位字符节点标识。例如，KD。

例如，代理名称 `AGENT@N@S` 展开为 `AGENTKD1234`。

当使用 `ftecfg` 命令将配置部署到安装时，会替换商店编号和节点标识。创建目录之前，`ftecfg` 命令创建的代理目录已完成名称替换。在磁盘上创建目录结构后，会更新 `agent.properties` 文件中的 `agentName` 属性，以与最终替换的代理名称相匹配。创建与下面类似的目录结构：

```
COORDQM (directory)
  coordination.properties
  MQMFTCcredentials.xml (optional)
  agents
    AGENTKD1234
    agent.properties
```

如何使用定制的代理名称

完成以下步骤：

1. 在系统上创建目录结构。
2. 更新代理名称以包含替换变量，并确保该目录结构中的代理名称使用大写。

如果代理名称不是大写，您将收到以下错误：

```
BFGCL0626W: Ignoring invalid path 'C:\Program Files (x86)\IBM\WebSphere
MQ\mqft\config\q1\agents\agent@N@S' within agents subdirectory of the source tree.
```

3. 确保您有相关文件，并在 `agent.properties` 文件中添加替换变量。
4. 运行 [第 519 页的『fteDefine（生成配置脚本）』](#) 命令，为代理生成 MQSC 脚本。
1. 修改 `agent.properties` 文件，对商店编号使用替换变量 `@S`，对节点标识使用 `@N`（如果必须包含商店编号和节点标识），以设置 `agentName` 属性。
2. 修改 `agent.properties` 文件来设置用于连接到代理队列管理器的属性。
3. 在配置捆绑软件目录结构中，重命名目录 `name` 以与在 `agent.properties` 文件中为 `agentName` 属性指定的代理名称值相匹配。
4. 修改 `coordination.properties` 文件来设置用于连接到协调队列管理器的属性。

样本

`custom1.zip` 文件包含一些样本文件，这些样本文件用于定制代理名称以包含商店编号和节点标识。有关如何根据您的系统定制这些样本文件的更多信息，请参阅 [第 76 页的『IBM 4690 系统的配置捆绑软件样本』](#)。

相关参考

[第 88 页的『ftecfg（在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置）』](#)

`ftecfg` 命令在 IBM 4690 系统上配置一个或多个 IBM MQ Managed File Transfer 代理程序。

[第 69 页的『在 4690 OS 配置束中定制代理属性』](#)

如果要将同一配置部署到多个商店，那么可以开发一个经过测试的标准配置束并将其部署到您的所有 4690 OS 商店控制器中，从而减少错误。然后，可以使用 `substitution.xml` 文件来定制所提供的束，以根据商店控制器属性来修改代理属性。

在 4690 OS 配置束中定制代理属性

如果要将同一配置部署到多个商店，那么可以开发一个经过测试的标准配置束并将其部署到您的所有 4690 OS 商店控制器中，从而减少错误。然后，可以使用 `substitution.xml` 文件来定制所提供的束， 以根据商店控制器属性来修改代理属性。

可以使用 `substitution.xml` 文件对以下条件表达式进行求值， 以提供 `true` 或 `false` 值：

- 商店控制器的商店编号等于某个值或在定义的范围
- 商店控制器的节点标识与某个正则表达式（不区分大小写）匹配
- 为商店控制器的网络接口卡 (NIC) 分配的 IP 地址等于某个值或在定义的范围
- 为商店控制器的 NIC 分配的 MAC 地址与某个值匹配

将按照这些条件在 `substitution.xml` 文件中的显示顺序对它们进行求值；第一个求值为 `true` 的条件将确定符号变量与值之间的映射。如果没有条件求值为 `true`，并且在 XML 中指定了缺省条件，那么将使用缺省值。否则，将输出一条部署时错误，并且不进行任何替换。

您可以使用以下布尔运算符来连接条件表达式：

- AND
- 或者
- 非

您可以嵌套运算符，系统会从嵌套的最深一层开始向外求值。最深一层的运算符的优先顺序最高。

在部署之前，`substitution.xml` 文件与配置束布局中的 `agent.properties` 文件位于同一目录中。例如：

```
COORDQM (directory)
  coordination.properties
  MQMFTCredentials.xml (optional)
  agents
    AGENT1
      agent.properties
      substitution.xml (optional)
      UserSandboxes.xml (optional)
```

使用 `ftecfg` 命令将配置束部署到商店控制器时，代理属性将替换为实际值。

如何替换代理属性

要替换代理属性，请完成以下步骤：

1. 将 `agent.properties` 文件内的条目设置为您针对要替换的属性选择的符号值。例如，您可以通过以下条目来替换代理队列管理器名称、代理队列管理器主机和代理队列管理器端口号的值：
 - `agentQMgr=${QM_NAME}`
 - `agentHost=${QM_HOST}`
 - `agentPort=${QM_PORT}`
2. 创建如下的 `substitution.xml` 文件，该文件用于定义替换这些符号值所必须满足的条件，并定义用来替换这些符号值的新值。

例如，根据 `substitution.xml` 文件，满足以下所有条件的商店控制器的 `agent.properties` 文件：

- 节点标识：KM
- 商店编号：1234
- IP 地址：192.168.10.1
- MAC 地址：08-00-27-00-94-2D

然后在部署时进行以下替换：

- agentQMgr=qmgr1
- agentHost=host1.example.org
- agentPort=1414

```
<?xml version="1.0" encoding="UTF-8"?>
  <tns:substitution xmlns:tns="http://wmqfte.ibm.com/Substitution"
                    xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
                    xsi:schemaLocation="http://wmqfte.ibm.com/Substitution
Substitution.xsd">
    <tns:condition>
      <tns:and>
        <tns:storeNumber low="0" high="4999"/>
        <tns:nodeId matches="KM"/>
        <tns:ipAddress address="192.168.10.1"/>
        <tns:macAddress address="08-00-27-00-94-2D"/>
      </tns:and>
      <tns:variable name="QM_NAME" value="qmgr1"/>
      <tns:variable name="QM_HOST" value="host1.example.org"/>
      <tns:variable name="QM_PORT" value="1414"/>
    </tns:condition>
    <tns:condition>
      <tns:or>
        <tns:storeNumber low="5000" high="9998"/>
        <tns:not><tns:nodeId matches="KM"/></tns:not>
        <tns:ipAddress address="192.168.56.101"/>
      </tns:or>
      <tns:variable name="QM_NAME" value="qmgr2"/>
      <tns:variable name="QM_HOST" value="host2.example.org"/>
      <tns:variable name="QM_PORT" value="1416"/>
    </tns:condition>
    <tns:default>
      <tns:variable name="QM_NAME" value="qmgr3"/>
      <tns:variable name="QM_HOST" value="host3.example.org"/>
      <tns:variable name="QM_PORT" value="1417"/>
    </tns:default>
  </tns:substitution>
```

样本

custom2.zip 文件包含根据商店控制器属性实施替换的样本文件。有关如何根据您的系统定制这些样本文件的更多信息，请参阅第 76 页的『IBM 4690 系统的配置捆绑软件样本』。

相关参考

第 70 页的『替换文件格式』

可以在配置束中包含一个 substitution.xml 文件，此文件根据要将配置束部署到的商店控制器的属性来定义如何定制 agent.properties 文件。

第 88 页的『ftcfg（在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置）』

ftcfg 命令在 IBM 4690 系统上配置一个或多个 IBM MQ Managed File Transfer 代理程序。

第 67 页的『在 4690 OS 配置捆绑软件中定制代理名称』

您可以将相同配置捆绑软件部署到多个不同的 4690 OS 商店控制器。要允许定制捆绑软件中的代理名称以与要将它们部署到的 4690 OS 商店控制器相匹配，可以使用基于商店编号和节点标识（有时称为商店控制器标识）的变量替换。

替换文件格式

可以在配置束中包含一个 substitution.xml 文件，此文件根据要将配置束部署到的商店控制器的属性来定义如何定制 agent.properties 文件。

substitution.xml 文件必须符合 Substitution.xsd 模式。Substitution.xsd 模式文档位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/4690` 目录中提供了一些样本文件。

模式

以下模式描述了哪些元素在 Substitution.xsd 模式中有效。

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
@start_non_restricted_prolog@
Version: %Z% %I% %W% %E% %U% [%H% %T%]

Licensed Materials - Property of IBM

5724-H72

Copyright IBM Corp. 2013, 2023. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
@end_non_restricted_prolog@
-->

<!-- This schema defines the format of a substitution.xml file. Files of this type
define a set of substitution variables and conditions on how they should be
applied to and agents agent.properties file at configuration deployment time.
-->

<!-- Example substitution.xml file:

<?xml version="1.0" encoding="UTF-8"?>
<tns:substitution xmlns:tns="http://wmqfte.ibm.com/Substitution"
                  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
                  xsi:schemaLocation="http://wmqfte.ibm.com/Substitution Substitution.xsd">
  <condition>
    <and>
      <storeNumber low="0" high="4999"/>
      <nodeId matches="KM"/>
      <ipAddress address="192.168.10.1"/>
      <macAddress address="08-00-27-00-94-2D"/>
    </and>
    <variable name="QM_NAME" value="qmgr1"/>
    <variable name="QM_HOST" value="host1.example.org"/>
    <variable name="QM_PORT" value="1414"/>
  </condition>
  <condition>
    <or>
      <storeNumber low="5000" high="9998"/>
      <not><nodeId matches="KM"/></not>
      <ipAddress address="192.168.10.1" mask="255.255.0.0"/>
    </or>
    <variable name="QM_NAME" value="qmgr2"/>
    <variable name="QM_HOST" value="host2.example.org"/>
    <variable name="QM_PORT" value="1416"/>
  </condition>
  <default>
    <variable name="QM_NAME" value="qmgr3"/>
    <variable name="QM_HOST" value="host3.example.org"/>
    <variable name="QM_PORT" value="1417"/>
  </default>
</tns:substitution>

-->

<schema targetNamespace="http://wmqfte.ibm.com/Substitution"
        elementFormDefault="qualified"
        xmlns="https://www.w3.org/2001/XMLSchema"
        xmlns:tns="http://wmqfte.ibm.com/Substitution">

  <element name="substitution" type="tns:mqmftSubstitutionType"/>

  <complexType name="mqmftSubstitutionType">
    <sequence>
      <element name="condition" type="tns:conditionType" minOccurs="0" maxOccurs="unbounded"/>
      <element name="default" type="tns:defaultType" minOccurs="0" maxOccurs="1"/>
    </sequence>
  </complexType>

<!--
Defines the condition element.
```

```

<condition>
  <and>
    ...
  </and>
  <variable name="QM_NAME" value="QM_test"/>
</condition>

or

<condition>
  <or>
    ...
  </or>
  <variable name="QM_HOST" value="qmttest.hursley.ibm.com"/>
  <variable name="QM_PORT" value="1414"/>
</condition>

or

<condition>
  <not>
    ...
  </not>
  <variable name="QM_HOST" value="qmttest.hursley.ibm.com"/>
</condition>

-->
<complexType name="conditionType">
  <sequence>
    <choice minOccurs="1" maxOccurs="1">
      <element name="and" type="tns:andType" minOccurs="0" maxOccurs="1"/>
      <element name="or" type="tns:orType" minOccurs="0" maxOccurs="1"/>
      <element name="not" type="tns:notType" minOccurs="0" maxOccurs="1"/>
      <element name="storeNumber" type="tns:storeNumberType" minOccurs="0" maxOccurs="1"/>
      <element name="nodeId" type="tns:nodeIdType" minOccurs="0" maxOccurs="1"/>
      <element name="ipAddress" type="tns:ipAddressType" minOccurs="0" maxOccurs="1"/>
      <element name="macAddress" type="tns:macAddressType" minOccurs="0" maxOccurs="1"/>
    </choice>
    <element name="variable" type="tns:variableType" minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!--
  Defines the and element.

  <and>
    <storeNumber low="0" high="4999"/>
    <nodeId matches="KM"/>
  </and>
-->
<complexType name="andType">
  <sequence>
    <choice minOccurs="2" maxOccurs="unbounded">
      <element name="storeNumber" type="tns:storeNumberType" minOccurs="0" maxOccurs="1"/>
      <element name="nodeId" type="tns:nodeIdType" minOccurs="0" maxOccurs="1"/>
      <element name="ipAddress" type="tns:ipAddressType" minOccurs="0" maxOccurs="1"/>
      <element name="macAddress" type="tns:macAddressType" minOccurs="0" maxOccurs="1"/>
      <element name="or" type="tns:orType" minOccurs="0" maxOccurs="unbounded"/>
      <element name="not" type="tns:notType" minOccurs="0" maxOccurs="unbounded"/>
    </choice>
  </sequence>
</complexType>

<!--
  Defines the or element.

  <or>
    <ipAddress address="192.168.10.1"/>
    <macAddress address="08-00-27-00-94-2D"/>
  </or>
-->
<complexType name="orType">
  <sequence>
    <choice minOccurs="2" maxOccurs="unbounded">
      <element name="storeNumber" type="tns:storeNumberType" minOccurs="0" maxOccurs="1"/>
      <element name="nodeId" type="tns:nodeIdType" minOccurs="0" maxOccurs="1"/>
      <element name="ipAddress" type="tns:ipAddressType" minOccurs="0" maxOccurs="1"/>
      <element name="macAddress" type="tns:macAddressType" minOccurs="0" maxOccurs="1"/>
      <element name="and" type="tns:andType" minOccurs="0" maxOccurs="unbounded"/>
      <element name="not" type="tns:notType" minOccurs="0" maxOccurs="unbounded"/>
    </choice>
  </sequence>

```

```

        </choice>
    </sequence>
</complexType>

<!--
    Defines the not element.

    <not><storeNumber low="0" high="4999"/></not>
-->
<complexType name="notType">
    <sequence>
        <choice minOccurs="1" maxOccurs="1">
            <element name="storeNumber" type="tns:storeNumberType" minOccurs="0" maxOccurs="1"/>
            <element name="nodeId" type="tns:nodeIdType" minOccurs="0" maxOccurs="1"/>
            <element name="ipAddress" type="tns:ipAddressType" minOccurs="0" maxOccurs="1"/>
            <element name="macAddress" type="tns:macAddressType" minOccurs="0" maxOccurs="1"/>
            <element name="and" type="tns:andType" minOccurs="0" maxOccurs="1"/>
            <element name="or" type="tns:orType" minOccurs="0" maxOccurs="1"/>
        </choice>
    </sequence>
</complexType>

<!--
    Defines the storeNumber element.

    <storeNumber low="0" high="3999"/>
-->
<complexType name="storeNumberType">
    <attribute name="low" type="nonNegativeInteger" use="required"/>
    <attribute name="high" type="nonNegativeInteger" use="required"/>
</complexType>

<!--
    Defines the nodeId element.

    <nodeId matches="KM"/>
-->
<complexType name="nodeIdType">
    <attribute name="matches" type="string" use="required"/>
</complexType>

<!--
    Defines the ipAddress element.

    <ipAddress address="192.168.0.1" mask="255.255.0.0"/>

    or

    <ipAddress address="192.168.0.1"/>
-->
<complexType name="ipAddressType">
    <attribute name="address" type="string" use="required"/>
    <attribute name="mask" type="string" use="optional"/>
</complexType>

<!--
    Defines the macAddress element.

    <macAddress address="08-00-27-00-94-2D"/>
-->
<complexType name="macAddressType">
    <attribute name="address" type="string" use="required"/>
</complexType>

<!--
    Defines the default element.

    <default>
        <variable name="QM_NAME" value="QM_test"/>
        <variable name="QM_HOST" value="qmttest.hursley.ibm.com"/>
        <variable name="QM_PORT" value="1414"/>
    </default>
-->
<complexType name="defaultType">
    <sequence>
        <element name="variable" type="tns:variableType" minOccurs="1" maxOccurs="unbounded"/>

```

```

</sequence>
</complexType>

<!--
  Defines the variable element.

  <variable name="QM_NAME" value="QM_test"/>
-->
<complexType name="variableType">
  <attribute name="name" type="string" use="required"/>
  <attribute name="value" type="string" use="required"/>
</complexType>

</schema>

```

Substitution.xsd 文件

以下列表描述了 Substitution.xsd 文件中使用的元素和属性。

<substitution>

这是一个组元素，其中包含一些元素来用于描述要从 substitution.xml 文件替换到 agent.properties 文件的信息以及需要满足的条件。

<condition>

依赖于商店控制器属性的条件表达式。 <condition> 元素可以包含 <and>、<or> 和 <not> 元素。

<default>

要用于被替换代理属性的缺省值（如果没有条件求值为 true）。

<variable>

要替换的代理属性的变量名称和变量值。

属性	描述
名	变量名称。例如，QM_NAME、QM_HOST 或 QM_PORT
值	您希望变量采用的值

<and>

用于连接条件表达式的 AND 布尔运算符。

<or>

用于连接条件表达式的 OR 布尔运算符。

<not>

用于排除条件表达式的 NOT 布尔运算符。

<storeNumber>

商店控制器的商店编号或商店控制器编号范围。例如，1234 或 0 - 4999。

属性	描述
low	商店编号范围内的最小编号
high	商店编号范围内的最大编号

<nodeId>

要进行匹配的节点标识（也称为商店控制器标识）。例如，KM。

属性	描述
匹配项	要进行匹配的节点标识（节点标识不区分大小写）

<ipAddress>

分配给商店控制器网络接口卡 (NIC) 的 IP 地址。

属性	描述
地址	要进行匹配的 IP 地址（例如 192.168.10.1）
掩码	匹配 IP 地址时要使用的子网掩码（例如 255.255.0.0）。该属性是可选的，但如果指定了该属性，将会导致商店控制器的 IP 地址与 IP 地址范围进行匹配。IP 地址范围由 address 和 mask 属性的组合来定义。

<macAddress>

分配给商店控制器网络接口卡 (NIC) 的 MAC 地址。例如，08-00-27-00-94-2D。

属性	描述
地址	要进行匹配的 MAC 地址

示例

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:substitution xmlns:tns="http://wmqfte.ibm.com/Substitution"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/Substitution Substitution.xsd">
  <tns:condition>
    <tns:and>
      <tns:storeNumber low="0" high="4999"/>
      <tns:nodeId matches="KM"/>
      <tns:ipAddress address="192.168.10.1"/>
      <tns:macAddress address="08-00-27-00-94-2D"/>
    </tns:and>
    <tns:variable name="QM_NAME" value="qmgr1"/>
    <tns:variable name="QM_HOST" value="host1.example.org"/>
    <tns:variable name="QM_PORT" value="1414"/>
  </tns:condition>
  <tns:condition>
    <tns:or>
      <tns:storeNumber low="5000" high="9998"/>
      <tns:not><tns:nodeId matches="KM"/></tns:not>
      <tns:ipAddress address="192.168.56.101"/>
    </tns:or>
    <tns:variable name="QM_NAME" value="qmgr2"/>
    <tns:variable name="QM_HOST" value="host2.example.org"/>
    <tns:variable name="QM_PORT" value="1416"/>
  </tns:condition>
  <tns:default>
    <tns:variable name="QM_NAME" value="qmgr3"/>
    <tns:variable name="QM_HOST" value="host3.example.org"/>
    <tns:variable name="QM_PORT" value="1417"/>
  </tns:default>
</tns:substitution>
```

相关参考

第 69 页的『在 4690 OS 配置束中定制代理属性』

如果要将同一配置部署到多个商店，那么可以开发一个经过测试的标准配置束并将其部署到您的所有 4690 OS 商店控制器中，从而减少错误。然后，可以使用 substitution.xml 文件来定制所提供的束，以根据商店控制器属性来修改代理属性。

第 76 页的『IBM 4690 系统的配置捆绑软件样本』

MQ_INSTALLATION_PATH/mqft/samples/4690 目录中提供了一些样本配置捆绑软件。您可以使用 **fteBundleConfiguration** 命令解包每个样本，然后针对您的特定配置来修改所抽取的文件。修改完成后，可以使用 **fteBundleConfiguration** 命令将这些文件打包成一个配置捆绑软件以便可以部署到 IBM 4690 系统。

IBM 4690 系统的配置捆绑软件样本

MQ_INSTALLATION_PATH/mqft/samples/4690 目录中提供了一些样本配置捆绑软件。您可以使用 **fteBundleConfiguration** 命令解包每个样本，然后针对您的特定配置来修改所抽取的文件。修改完成后，可以使用 **fteBundleConfiguration** 命令将这些文件打包成一个配置捆绑软件以便可以部署到 IBM 4690 系统。

basic.zip

可以使用 basic.zip 样本以利用最少的定制来配置代理。配置捆绑软件的结构如下：

```
agents/  
  name/  
    agent.properties  
  coordination.properties
```

要针对您的需求来定制该样本，请完成以下步骤：

1. 修改 agent.properties 文件来设置 agentName 属性以及设置用于连接到代理队列管理器的属性。
2. 在配置捆绑软件目录结构中，重命名目录 name 以与在 agent.properties 文件中为 agentName 属性指定的代理名称值相匹配。
3. 修改 coordination.properties 文件来设置用于连接到协调队列管理器的属性。

custom1.zip

您可以使用 custom1.zip 样本来配置代理名称以包含商店编号和/或节点标识。这样，该更新允许将配置部署到多个 4690 系统。配置捆绑软件的结构如下：

```
agents/  
  name/  
    agent.properties  
  coordination.properties
```

要针对您的需求来定制该样本，请完成以下步骤：

1. 修改 agent.properties 文件，对商店编号使用替换变量 @S，对节点标识使用 @N（如果必须包含商店编号和节点标识），以设置 agentName 属性。
2. 修改 agent.properties 文件来设置用于连接到代理队列管理器的属性。
3. 在配置捆绑软件目录结构中，重命名目录 name 以与在 agent.properties 文件中为 agentName 属性指定的代理名称值相匹配。
4. 修改 coordination.properties 文件来设置用于连接到协调队列管理器的属性。

custom2.zip

您可以使用 custom2.zip 样本以根据要将捆绑软件部署到的商店控制器的商店编号、节点标识、IP 地址和网络 MAC 地址来配置 agent.properties 文件的内容。该配置可通过在 substitution.xml 文件中定义一个或多个条件来完成。这些条件将设置随后可替换到 agent.properties 文件的替换变量。配置捆绑软件的结构如下：

```
agents/  
  name/  
    agent.properties  
    substitution.xml  
  coordination.properties
```

要针对您的需求来定制该样本，请完成以下步骤：

1. 修改 substitution.xml 文件来指定适用于配置的条件和变量。
2. 修改 agent.properties 文件来设置 agentName 属性。您可以对商店编号使用替换变量 @S，对节点标识使用 @N（与 custom1.zip 样本中所使用的方法相同）以对 agentName 进行定制。但是，不能使用 substitution.xml 文件中指定的替换变量对 agentName 进行定制。

3. 修改 `agent.properties` 文件来设置用于连接到代理队列管理器的属性，必要时请指定替换变量。
4. 在配置捆绑软件目录结构中，重命名目录 `name` 以与在 `agent.properties` 文件中为 `agentName` 属性指定的代理名称值相匹配。
5. 修改 `coordination.properties` 以设置用于连接到协调队列管理器的属性。

SSL.zip

您可以使用 `SSL.zip` 样本来配置代理以通过使用 SSL 连接到代理队列管理器。该样本指定要部署到单个 4690 系统的配置。然而，您也可以使用 `custom1` 和 `custom2` 样本中描述的使用 SSL 的方法来将配置部署到多个系统。配置捆绑软件的结构如下：

```
agents/  
  name/  
    agent.properties  
    coordination.properties  
    MQMFTCredentials.xml
```

要针对您的需求来定制该样本，请完成以下步骤：

1. 修改 `agent.properties` 文件来设置 `agentName` 属性以及用于连接到代理队列管理器的属性。
2. 修改 `agent.properties` 文件来设置用于配置 SSL 的属性。有关更多信息，请参阅样本 `agent.properties` 文件中的注释。
3. 在配置捆绑软件目录结构中，重命名目录 `name` 以与在 `agent.properties` 文件中为 `agentName` 属性指定的代理名称值相匹配。
4. 修改 `coordination.properties` 来设置用于连接到协调队列管理器的属性。
5. 修改 `MQMFTCredentials.xml` 文件来为 SSL 配置中指定的任何密钥库和信任库文件指定密码。

相关任务

[第 66 页的『在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置』](#)

要在 IBM 4690 系统上创建或替换 IBM MQ Managed File Transfer 配置，应先在非 4690 平台上创建配置。然后，将配置以压缩文件格式复制到 IBM 4690 系统上，并运行 **ftecfig** 命令以将配置部署到 IBM 4690 系统。

相关参考

[第 67 页的『在 4690 OS 配置捆绑软件中定制代理名称』](#)

您可以将相同配置捆绑软件部署到多个不同的 4690 OS 商店控制器。要允许定制捆绑软件中的代理名称以与要将它们部署到的 4690 OS 商店控制器相匹配，可以使用基于商店编号和节点标识（有时称为商店控制器标识）的变量替换。

[第 67 页的『IBM 4690 IBM MQ Managed File Transfer 配置压缩文件的结构』](#)

您可以通过将包含配置详细信息的压缩文件作为参数传递至 **ftecfig** 命令，从而在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置。

[第 69 页的『在 4690 OS 配置束中定制代理属性』](#)

如果要将同一配置部署到多个商店，那么可以开发一个经过测试的标准配置束并将其部署到您的所有 4690 OS 商店控制器中，从而减少错误。然后，可以使用 `substitution.xml` 文件来定制所提供的束，以根据商店控制器属性来修改代理属性。

[第 85 页的『fteBundleConfiguration（创建 IBM MQ Managed File Transfer IBM 4690 代理配置 .zip 文件）』](#)

使用 **fteBundleConfiguration** 命令可将指定目录中的 IBM MQ Managed File Transfer 配置树捆绑成 .zip 文件。

[第 88 页的『ftecfig（在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置）』](#)

ftecfig 命令在 IBM 4690 系统上配置一个或多个 IBM MQ Managed File Transfer 代理程序。

在主/备份 4690 OS 控制器设置中配置 IBM MQ Managed File Transfer

您可以在主备份 4690 OS 控制器设置中配置代理以提供故障容错。具有相同名称的代理可以配置为在零售环境中的多个控制器上运行。但是，在任何一个时候都只能运行其中一个代理。该配置可以与配置为在不同环境中停止和启动的应用程序结合使用。

您可以通过以下方式在 4690 上配置永久后台应用程序：

- 当商店控制器变成活动的主控制器时启动应用程序，当控制器不再是代理主控制器时停止应用程序。
- 当商店控制器变成活动的文件服务器时启用应用程序，当控制器不再是代理文件服务器时停止应用程序。

因此，在主和备用主商店控制器配置以及文件服务器和备用文件服务器商店控制器配置上，通过此方式将具有相同名称的代理配置成为后台应用程序，您可以对 4690 OS 系统上的 IBM MQ Managed File Transfer 提供容错设置。

有关如何配置代理的详细信息，请参阅第 78 页的『在 4690 OS 系统上启动代理』。

相关概念

第 78 页的『在 4690 OS 控制器设置中配置多个 IBM MQ Managed File Transfer 代理』

您可以在单个商店控制器环境中或在商店内有多个商店控制器的环境中，配置多个代理。

在 4690 OS 控制器设置中配置多个 IBM MQ Managed File Transfer 代理

您可以在单个商店控制器环境中或在商店内有多个商店控制器的环境中，配置多个代理。

您可以配置多个代理以在单个商店控制器上运行，前提是这些代理的名称对于该控制器是唯一的并且针对每个代理都提供了一个后台应用程序插槽。

对于商店中存在多个商店控制器的环境，为每个商店控制器配置以下代理很有用：

- 为商店的主和备用主商店控制器配置（以便仅在代理的商店控制器为活动主控制器时才运行该代理）的同名代理。该代理可用于传输在商店的两个控制器之间常见或共享的文件。您还可以对文件服务器和备用文件服务器商店控制器使用相同的配置。
- 其名称对商店控制器唯一且配置为每当商店控制器运行时才运行的代理。该代理可用于传输特定于商店控制器的文件。

在该示例中，商店 7777 具有 KD 和 KM 两个控制器。

为商店 7777 和控制器 KD 配置的代理为：

- 代理 1：AGENT7777 - 特定于商店的代理（主实例）
- 代理 2：AGENT7777KD - 特定于控制器（控制器 KD）的代理

为商店 7777 和控制器 KM 配置的代理为：

- 代理 1：AGENT7777 - 特定于商店的代理（备用实例）
- 代理 2：AGENT7777KM - 特定于控制器（控制器 KM）的代理

有关如何配置代理的详细信息，请参阅第 78 页的『在 4690 OS 系统上启动代理』。

相关概念

第 77 页的『在主/备份 4690 OS 控制器设置中配置 IBM MQ Managed File Transfer』

您可以在主备份 4690 OS 控制器设置中配置代理以提供故障容错。具有相同名称的代理可以配置为在零售环境中的多个控制器上运行。但是，在任何一个时候都只能运行其中一个代理。该配置可以与配置为在不同环境中停止和启动的应用程序结合使用。

在 4690 OS 系统上启动代理

要启动代理，必须先将其配置为后台应用程序。可以配置后台应用程序在系统变成代理主控制器或文件服务器时自动启动，在系统不再是代理主控制器或文件服务器时停止。

关于此任务

已通过运行 `f:\adxetc\mft75\bin\ftecfg.bat` 实用程序装入新配置，定义了可启动的代理。对于每个配置的代理，`f:\adxetc\mft75` 目录中都存在一个 `agent_name.rsp` 响应文件。所有这些响应文件都包含一条命令以在传递 ADXCHAIN.386 系统命令时启动特定代理。

过程

完成以下步骤以启动作为后台服务运行的代理：

1. 从“系统主菜单”面板中选择 **4（安装和更新帮助）**。
2. 从“安装和更新帮助”面板中选择 **1（更改配置数据）**。
3. 从“配置”面板中选择 **2（控制器配置）**。
4. 如果询问您是否要配置使用 IBM Multiple Controller Feature 的存储系统，请按 **Enter** 键以选择 **Yes**。
5. 如果显示“局域网配置”面板，请选择适用于您环境的选项，然后按 **Enter** 键。
6. 如果显示“SNA 配置”面板，请选择适用于您环境的选项，然后按 **Enter** 键。
7. 如果提示您输入存储控制器标识，请指定相应的控制器标识，然后按 **Enter** 键。
8. 选择要配置的存储控制器，然后按 **Enter** 键。
9. 从控制器配置项列表中，选择**后台配置**，然后按 **Enter** 键。
10. 从“后台应用程序”面板中选择 **1（定义后台应用程序）**。
11. 在“定义后台应用程序”面板上，指定以下条目：
 - 初始消息：MQMFT
 - 程序名：ADX_SPGM:ADXCHAIN.386
 - 参数列表：@f:\adxetc\mft75\agent_name.rsp

注：agent_name 限制最多 23 个字符，参数列表条目限制最多 45 个字符。响应文件的参数列表路径必须完全按照所示格式指定，即不能更改大写字符或正斜杠。不允许定义逻辑名以指定到响应文件的路径。
12. 按**下一页**以查看更多选项，并指定当系统变为代理主系统或不再是代理主系统时应用程序应启动还是停止。
13. 再次按**下一页**以查看更多选项，并指定当系统变为代理文件服务器或不再是代理文件服务器时应启动还是停止。
14. 按 **Enter** 键保存变更。
15. 按 **Esc** 键返回“配置”面板。
16. 从“配置”面板中选择 **4（激活配置）**。
17. 从“激活配置”面板中选择 **2（控制器配置）**。
这会激活控制器配置。
18. 重新 IPL 存储控制器。

代理状态消息

在 IBM 4690 系统上，您可以通过浏览至后台应用程序控制面板查看 IBM MQ Managed File Transfer 提供的后台应用程序的代理状态消息。可能有多个状态消息。

```
hh:mm Initializing
hh:mm Starting
hh:mm Not connected to MQ (last MQRC=nnnn)
hh:mm Recovering
hh:mm Waiting for work
hh:mm Transferring source:nnnn destination:nnnn
hh:mm Failed (RC=nnnn)
hh:mm Controlled stopping
hh:mm Immediate stopping
hh:mm Controlled stopped
hh:mm Immediate stopped
```

正在初始化

代理正在启动并正在初始化它所需的资源。

正在启动

代理已成功初始化，正准备开始处理传输。

Not connected to MQ

代理正在运行，但正在等待队列管理器变为可用。在成功重新连接之前，代理无法处理传输。有关 last MQRC 的更多信息，请参阅 [API 完成代码和原因码](#)。

Recovering

代理正在运行并重新建立与队列管理器的连接。代理正在恢复连接丢失时运行的任何传输。

Waiting for work

代理正在运行，但是没有传输。

Transferring source:nnnn destination:nnnn

代理正在运行，并且正在传输文件。代理为源和代理为目标的传输数由显示的值表示。

失败

代理失败，并且不可能恢复。有关 RC 的更多信息，请参阅本主题后部的返回码表。

Controlled stopping

代理正在停止。请求受控关闭，同时允许任何正在运行的传输完成。

Immediate stopping

代理正在停止。请求立即关闭。

Controlled stopped

代理已停止。它以受控方式关闭。

Immediate stopped

代理已停止。它以非受控方式关闭。

下表列出了返回码及其意义：

表 1: 返回码		
返回码	短名称	描述
0	成功	应用程序成功终止。
1	失败	一般应用程序故障返回码
2	退出	强制退出应用程序。例如，诊断系统请求终止应用程序。
70	异常结束	应用程序发生不可恢复的问题，被强行终止。
78	配置	由于启动配置数据存在问题，因此应用程序无法继续。

相关任务

第 78 页的『在 4690 OS 系统上启动代理』

要启动代理，必须先将其配置为后台应用程序。可以配置后台应用程序在系统变成代理主控制器或文件服务器时自动启动，在系统不再是代理主控制器或文件服务器时停止。

相关参考

第 80 页的『进程控制器状态消息』

在 IBM 4690 系统上，您可以通过浏览至后台应用程序控制面板来查看 IBM MQ Managed File Transfer 提供的后台应用程序的进程控制器状态消息。可能有多个状态消息。

进程控制器状态消息

在 IBM 4690 系统上，您可以通过浏览至后台应用程序控制面板来查看 IBM MQ Managed File Transfer 提供的后台应用程序的进程控制器状态消息。可能有多个状态消息。

```
hh:mm Starting
hh:mm Monitoring agent process
hh:mm nnnn agent restarts (last: hh:mm:ss)
```



```
hh:mm Failed (RC=nnnn)
hh:mm Stopped
```

正在启动

进程控制器已成功初始化，正准备开始对代理进行监控。

Monitoring agent process

进程控制器正在监控代理进程。

nnnn agent restarts (last: hh:mm:ss)

从进程控制器启动以来，进程控制器重新启动代理的总次数以及代理最后一次重新启动的时间。

失败

进程控制器失败。有关 RC 的更多信息，请参阅本主题后部的返回码表。

已停止

进程控制器已停止。它以受控方式关闭。

下表列出了返回码及其意义：

表 2: 返回码		
返回码	短名称	描述
0	成功	应用程序成功终止。
1	失败	一般应用程序故障返回码
2	退出	强制退出应用程序。例如，诊断系统请求终止应用程序。
70	异常结束	应用程序发生不可恢复的问题，被强行终止。
78	配置	由于启动配置数据存在问题，因此应用程序无法继续。

相关任务

第 78 页的『在 4690 OS 系统上启动代理』

要启动代理，必须先将其配置为后台应用程序。可以配置后台应用程序在系统变成代理主控制器或文件服务器时自动启动，在系统不再是代理主控制器或文件服务器时停止。

相关参考

第 79 页的『代理状态消息』

在 IBM 4690 系统上，您可以通过浏览至后台应用程序控制面板查看 IBM MQ Managed File Transfer 提供的后台应用程序的代理状态消息。可能有多个状态消息。

在 4690 OS 系统上运行时的限制

在零售环境中的 4690 OS 系统上运行 IBM MQ Managed File Transfer 时，存在一些限制和不受支持的功能。

以下限制适用于 4690 OS：

- 4690 OS 代理名称最多只能包含 23 个字符。
- 4690 OS 代理响应文件位于 `f:\adxetc\mft75` 目录中。
- 对于定义为后台应用程序的代理，PARAM 设置必须采用格式 `@f:\adxetc\mft75\agent_name.rsp`。

注：参数列表条目限制为最多 45 个字符。必须严格按显示的格式指定响应文件的参数列表路径。这意味着不能更改大写字符或正斜杠。不允许定义逻辑名以指定响应文件的路径。

- 在 4690 OS 系统上只能定义一个协调队列管理器。
- `transferRoot` 属性在 4690 OS 上没有缺省目录路径。如果您要使用相对路径与 4690 代理进行传输，那么必须设置 `transferRoot`。有关更多信息，请参阅[高级代理属性](#)。

4690 OS 上不支持以下功能：

- 不能直接在 4690 OS 平台上运行 `fteAnt` 脚本。
- 对于将 4690 OS 代理作为源代理的传输，无法从 `presrc` 或 `postsrc` 程序调用运行 Apache Ant 脚本。
- 对于将 4690 OS 代理作为目标代理的传输，无法从 `predest` 或 `postdest` 程序调用运行 Ant 脚本。
- 不能在 4690 OS 平台上配置 IBM MQ Managed File Transfer 记录器。此限制包括所有三个版本: 独立文件记录器，独立数据库记录器和 Java Platform Enterprise Edition (JEE) 记录器。
- 不能在 IBM 4690 平台上将代理配置为协议网桥或 `Connect:Direct` 网桥。
- 不能在 4690 OS 平台上配置 Web 网关。
- 4690 OS 平台上不支持 International Components for Unicode (ICU) for Java。这意味着对于文本文件传输，所使用的字符集转换器由 4690 OS 平台上的 Java 运行时环境版本 6.0 提供。

当配置 4690 OS 过程中需要路径值时，支持两种路径格式：

- 盘符:\其余路径。例如，`f:\adxetc\mft75`
- 逻辑名:其余路径。例如，`f_drive:adxetc\mft75`

对于 IBM MQ Managed File Transfer，如果 `remainder-of-path` 的开头包含反斜杠 (\)，将忽略反斜杠以提供可预测性更强的行为。

除了此限制外，4690 OS 上还不支持某些属性。下表列出了期望将路径作为其值的所有属性以及指示它们是否支持在 4690 OS 上使用。

4690 OS 上支持的路径属性。	4690 OS 上不支持的路径属性。
<code>agentSslKeyStore</code>	<code>cdTmpDir</code>
<code>agentSslKeyStoreCredentialsFile</code>	<code>cdNodeKeystoreCredentialsFile</code>
<code>agentSslTrustStore</code>	<code>cdNodeTruststoreCredentialsFile</code>
<code>agentSslTrustStoreCredentialsFile</code>	<code>exitNativeLibraryPath</code>
<code>commandPath</code>	<code>wmqfte.database.credentials.file</code>
<code>exitClassPath</code>	
<code>javaCoreTriggerFile</code>	
<code>sandboxRoot</code>	
<code>transferRoot</code>	
<code>connectionSslKeyStore</code>	
<code>connectionSslKeyStoreCredentialsFile</code>	
<code>connectionSslTrustStore</code>	
<code>connectionSslTrustStoreCredentialsFile</code>	
<code>coordinationSslKeyStore</code>	
<code>coordinationSslKeyStoreCredentialsFile</code>	
<code>coordinationSslTrustStore</code>	
<code>coordinationSslTrustStoreCredentialsFile</code>	

文件分发属性

当 IBM MQ Managed File Transfer 将文件传输至 IBM 4690 商店控制器上运行的目标代理时，它必须设置文件的文件分发属性。这些分发属性确定 IBM 4690 商店控制器是否复制其他商店控制器的文件内容以及如何在“多个控制器功能 (MCF)”网络中复制文件的内容。IBM MQ Managed File Transfer 文件分发属性结合 MCF 的文件类型和文件模式属性。

您可以通过以下方式设置目标处文件的分发属性：

- 使用 IBM MQ 资源管理器 中的“**添加传输项目**”面板。
- 使用 **fteCreateTransfer** 命令上的 **-dfa** 参数。

如果未为使用 4690 目标的传输指定 **-dfa** 参数，那么缺省目标属性为 LOCAL。如果文件已存在于目标处，那么将使用现有文件的当前分发属性。

有关更多信息，请参阅 [第 506 页的『指定目标的参数』](#)。

每种文件类型和文件模式组合都由以下符号和数字值表示。指定符号或数字值。

表 3: IBM MQ Managed File Transfer 中的文件分发属性		
符号值	数字值	描述
DIST(LOCAL)	DIST(1)	本地文件。 本地文件只存在于一个商店控制器中。当更新本地文件时，网络中其他商店控制器不受影响。
DIST(MIRRORED,UPDATE)	DIST(2)	映像文件，更新时分发 镜像文件存在于两个商店控制器中，包括主要版和镜像版。当主要版文件中的记录更改或删除时，镜像版中的相同记录也会更改或删除。
DIST(MIRRORED, CLOSE)	DIST(3)	映像文件，关闭时分发 镜像文件存在于两个商店控制器中，包括主要版和镜像版。当关闭主要版的文件时，也会更新镜像版的文件。
DIST(COMPOUND,UPDATE)	DIST(4)	复合文件，更新时分发 将复合文件分发给所有商店控制器，除了没有资格的下级商店控制器。当主要版文件中的记录更改或删除时，镜像版文件中的相同记录也会更改或删除。
DIST(COMPOUND,CLOSE)	DIST(5)	复合文件，关闭时分发 将复合文件分发给所有商店控制器，除了没有资格的下级商店控制器。当关闭主要版的文件时，也会更新镜像版的文件。

有关 MCF 的更多信息，请参阅 [4690 OS V 6 Release 3 User 's Guide](#) 中的第 22 章 "Using the Multiple Controller Feature"。

使用文件分发的目录需求

要按您期望那样分发文件，请确保网络中所有商店控制器上存在相同的目录结构。通常，在安装时按操作系统或用户在商店控制器上创建此目录结构。如果所有控制器上的目录结构不是复制的，那么对没有所需目录的控制器分发失败。忽略并且不记录分发失败。

例如，如果将具有分发属性 MIRRORED,UPDATE 的文件传输到 IBM 4690 并且目标代理创建目录，那么不会分发传输到那些新目录的文件，因为其他控制器上的目录不是由 IBM MQ Managed File Transfer 创建的。

支持分发的驱动器

分布式文件只能位于 root 目录下的目录中，一般只能在安装期间由操作系统创建的目录中找到。通常，分布式文件位于使用 8.3 命名约定的驱动器上，即 C: 和 D: 盘。F: 盘上不支持具有分发属性的文件。逻辑驱动器 M: 和 N: 例外，因为它们是使用 4690 虚拟文件系统创建的，创建这些驱动器目的是支持长文件和目录名称。

表 4: 支持分发的驱动器	
IBM 4690 盘符	是否支持文件分发?
C:	YES
D:	YES
F:	否
M:	YES 逻辑盘 M: 上的信息存储在使用 VFS 的 C: 盘上
N:	YES 逻辑盘 N: 上的信息存储在使用 VFS 的 D: 盘上

相关信息

[TRANSFER_ITEM_ATTRIBUTES 数据库记录器表](#)

[fte:filespec dstAttributes 属性](#)

在 IBM 4690 系统上的沙箱中工作

IBM MQ Managed File Transfer 使用沙箱来限制在传输过程中代理或用户可访问的文件系统区域。要限制访问，代理沙箱使用 sandboxRoot 属性，用户沙箱使用 UserSandboxes.xml 文件中的信息。

UserSandboxes.xml 文件中的 sandboxRoot 属性和信息都可以指定与传输请求中指定的路径相比较的路径信息。在 4690 上，解释这些文件中路径信息的方式服从以下规则：

- 假设路径为本机 4690 路径，例如：f:\adxetc
- F: 盘上的路径区分大小写。除了 F: 盘外，所有驱动器上的路径都不区分大小写。
- 您可以指定包含逻辑名的路径。比较期间展开逻辑名。因此，根据与 LN1 和 LN2 相关的路径信息，LN1:\DIR\FILE.TXT 可能与 LN2:\FILE.TXT 相同。

有关代理沙箱的更多信息，请参阅第 99 页的『使用代理沙箱』，有关用户沙箱的更多信息，请参阅第 100 页的『使用用户沙箱』。

零售环境中使用的 IBM MQ Managed File Transfer 命令的摘要

下面列出了零售环境中使用的所有 IBM MQ Managed File Transfer 命令及其详细描述的连接。

表 5: IBM MQ Managed File Transfer 命令及其用途	
命令名	用途
fteBundleConfiguration	将指定目录中的 IBM MQ Managed File Transfer 配置树捆绑成 .zip 文件。
ftelap	运行 License Acceptance Process (LAP) 工具。LAP 工具用于读取和接受与 IBM MQ Managed File Transfer 关联的许可证。
ftecfg	在 IBM 4690 系统上配置一个或多个 IBM MQ Managed File Transfer 代理。
ftediag	在 IBM 4690 系统上生成 IBM MQ Managed File Transfer 代理的诊断信息。

表 5: IBM MQ Managed File Transfer 命令及其用途 (继续)	
命令名	用途
<code>uninstall</code>	从 IBM 4690 商店控制器卸载 IBM MQ Managed File Transfer 代理。

将以语法图（称为轨道图）形式提供每个命令及其参数的语法。有关如何解释轨道图的信息，请参阅[如何阅读轨道图](#)。

fteBundleConfiguration（创建 IBM MQ Managed File Transfer IBM 4690 代理配置 .zip 文件）

使用 **fteBundleConfiguration** 命令可将指定目录中的 IBM MQ Managed File Transfer 配置树捆绑成 .zip 文件。

用途

fteBundleConfiguration 命令将 IBM 4690 安装所需的配置文件打包并解包。所创建的 .zip 文件可以提供给 **ftectfg** 命令以供 IBM 4690 安装使用。**fteBundleConfiguration** 命令验证目录中的文件以确保配置有效，如果输入内容的格式错误，将显示警告。将捆绑格式正确的任何代理配置，并生成一个可用的配置 .zip 文件。**ftectfg** 命令无法部署格式不正确的配置。

注: 仅支持 IBM MQ Managed File Transfer V7.5 和更高版本的配置。

SSL 属性文件的路径

SSL 属性 `agentSslKeyStore`、`agentSslKeyStoreCredentialsFile`、`agentSslTrustStore` 和 `agentSslTrustStoreCredentialsFile` 以及 `mqmftcredentials.xml` 文件包含配置捆绑软件中可包含的路径值。**fteBundleConfiguration** 命令根据路径是相对路径还是绝对路径来处理这些值。

本地（非 4690 平台）绝对路径

如果捆绑软件配置目录中是绝对路径并且该文件存在，那么该路径将转换为相对路径，并包含在配置捆绑软件中。在部署捆绑软件时，**ftectfg** 命令解包到绝对路径。

如果捆绑软件配置目录中不是绝对路径或者该文件不存在，那么将生成错误消息。

本地（非 4690 平台）相对路径

如果相对路径指向捆绑软件配置目录中存在的文件，那么该路径将包含在配置捆绑软件中。在部署捆绑软件时，**ftectfg** 命令解包到绝对路径。

如果相对路径所指向的文件不在捆绑软件配置目录中，那么将生成错误消息。

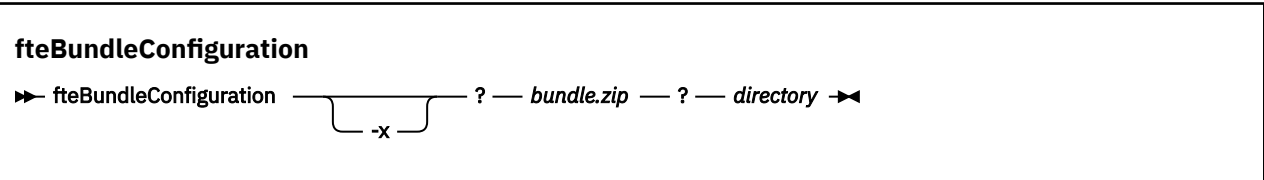
IBM 4690 absolute path in the format <drive_or_logical_name>:<路径的其余部分>

在 UNIX 上，假定绝对路径指向 IBM 4690 文件并包含在捆绑软件配置中。

在 Windows 上，如果绝对路径指向本地不存在的文件，那么假定该路径指向 IBM 4690 文件并包含在捆绑软件配置中。如果文件本地存在并位于捆绑软件配置目录下，那么该路径将转换为相对路径，并包含在配置捆绑软件中。

在 Windows 上，如果绝对路径指向本地存在但不在捆绑软件配置目录中的文件，那么假定该路径指向非 IBM 4690 文件。将生成错误消息。

语法



参数

bundle.zip

必需。要处理的配置捆绑软件的路径。如果不使用 **-x** 参数, *bundle.zip* 便是所创建的文件。

directory

必需。在其上运行命令的目录的路径。如果不使用 **-x** 参数, *directory* 便是要包含在压缩文件中的配置源。指定 **-x** 参数时, *directory* 是从压缩文件中抽取的配置的目标。

-x

可选。该参数用于将 *bundle.zip* 文件中的配置文件解包到指定的目录。不使用该参数时, 将从指定目录中创建 *bundle.zip* 文件。

示例

在该示例中, 使用 *QM_test* 目录中的配置成功创建了 *config.zip* 捆绑软件:

```
fteBundleConfiguration C:\config.zip C:\WebSphereMQ\7500\mqft\config\QM_test
```

```
BFGCL0620I: The bundle 'C:\config.zip' has been successfully created from the configuration in
directory C:\WebSphereMQ\7500\mqft\config\QM_test'.
```

在该示例中, *config.zip* 捆绑软件成功抽取到 *config_extract* 目录:

```
fteBundleConfiguration -x C:\config.zip C:\Temp\config_extract
```

```
BFGCL0621I: The contents of the bundle 'C:\config.zip' have been successfully extracted to
directory
'C:\Temp\config_extract'.
```

在该示例中, 由于代理属性文件的内容无效, 所以 *config.zip* 捆绑软件创建失败:

```
fteBundleConfiguration C:\config.zip C:\WebSphereMQ\7500\mqft\config\QM_test
```

```
BFGUB0039E: A required property of 'agentQMgr' is missing from the properties.
BFGCL0629E: The contents of the agent.properties file for agent 'FTEAGENT7777' are not valid.
Configuration for this agent will not be included in the bundle.
BFGCL0628E: No valid agent configurations found in directory
'C:\WebSphereMQ\7500\mqft\config\QM_test'. The new bundle cannot be created.
```

在该示例中, 由于协调属性文件的内容无效, 所以 *config.zip* 捆绑软件创建失败:

```
fteBundleConfiguration C:\config.zip C:\WebSphereMQ\7500\mqft\config\QM_test
```

```
BFGUB0022E: The property name "coordinationQMgrPort" has an invalid numeric value of "14B14"
BFGCL0631E: The contents of the coordination.properties file in directory
'C:\WebSphereMQ\7500\mqft\config\QM_test' are not valid. The new bundle cannot be created.
```

在该示例中, *config.zip* 捆绑软件创建成功, 从本地绝对路径更改为相对路径:

```
fteBundleConfiguration C:\config_ssl_paths.zip C:\WebSphereMQ\7500\mqft\config\QM_test
```

```
BFGCL0660I: The local absolute path
'C:\WebSphereMQ\7500\mqft\config\ssl_path_config\QM_gbthink\mqmftcredentials.xml'
supplied for property 'agentSslKeyStoreCredentialsFile' was converted to relative path
'mqmftcredentials.xml'
for inclusion in the bundle.
BFGCL0620I: The bundle 'C:\config_ssl_paths.zip' has been successfully created from the
```



```
configuration
in directory 'C:\WebSphereMQ\7500\mqft\config\QM_test'.
```

在该示例中，config.zip 捆绑软件创建成功，包含无法确定的路径：

```
fteBundleConfiguration C:\config_ssl_paths.zip C:\WebSphereMQ\7500\mqft\config\QM_test
```

```
BFGCL0662W: It could not be determined if the absolute path 'f:\keystore.jks' referred to a
local file
or a 4690 file so it will be left unchanged.
BFGCL0620I: The bundle 'C:\config_ssl_paths.zip' has been successfully created from the
configuration
in directory 'C:\WebSphereMQ\7500\mqft\config\QM_test'.
```

在该示例中，由于引用的本地绝对代理属性路径位于配置目录外部，所以 config.zip 捆绑软件创建失败：

```
fteBundleConfiguration C:\config_ssl_paths.zip C:\WebSphereMQ\7500\mqft\config\QM_test
```

```
BFGCL0659E: The local absolute path 'D:\\AGENTS\\mqmftcredentials.xml' supplied for property
'agentSslKeyStoreCredentialsFile' is outside of the configuration directory being bundled.
```

在该示例中，由于引用的本地绝对凭证路径位于配置目录外部，所以 config.zip 捆绑软件创建失败：

```
fteBundleConfiguration C:\config_ssl_paths.zip C:\WebSphereMQ\7500\mqft\config\QM_test
```

```
BFGCL0663E: The local absolute path 'C:\keystore.jks' supplied in credentials file
'C:\WebSphereMQ\7500\mqft\config\QM_test\agents\FTEAGENT7777\mqmftcredentials.xml' is outside
of
the configuration directory being bundled.
```

相关任务

第 66 页的『在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置』

要在 IBM 4690 系统上创建或替换 IBM MQ Managed File Transfer 配置，应先在非 4690 平台上创建配置。然后，将配置以压缩文件格式复制到 IBM 4690 系统上，并运行 **ftecfg** 命令以将配置部署到 IBM 4690 系统。

相关参考

第 67 页的『IBM 4690 IBM MQ Managed File Transfer 配置压缩文件的结构』

您可以通过将包含配置详细信息的压缩文件作为参数传递至 **ftecfg** 命令，从而在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置。

ftelap（在 IBM MQ Managed File Transfer 安装期间接受许可协议）

ftelap 命令可运行 License Acceptance Process (LAP) 工具。LAP 工具用于读取和接受与 IBM MQ Managed File Transfer 关联的许可证。

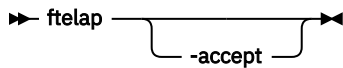
用途

在使用任何 IBM MQ Managed File Transfer 功能之前，请使用 **ftelap** 命令来查看和接受许可证。

解压安装文件后，首先必须接受产品许可证，然后才能通过转至 \adxetc\mft75\bin 目录并运行 **ftelap** 命令来继续安装。

语法

ftelap



参数

accept

可选。如果指定 **accept** 参数，将自动接受许可证。如果不指定 **accept** 参数，将显示许可证，然后您必须选择接受或拒绝。

相关任务

第 61 页的『在 4690 OS 上安装 IBM MQ Managed File Transfer』
使用 MQMFT75.ZIP 文件在 4690 OS 上安装 IBM MQ Managed File Transfer。

ftecfg (在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置)

ftecfg 命令在 IBM 4690 系统上配置一个或多个 IBM MQ Managed File Transfer 代理程序。

用途

使用 **ftecfg** 命令在 IBM 4690 系统上配置一个或多个 IBM MQ Managed File Transfer 代理。该命令采用压缩文件（使用 **fteBundleConfiguration** 命令创建）的路径作为参数。该压缩文件包含要安装的代理的配置详细信息，并生成直接从压缩文件填充的配置目录树。**ftecfg** 命令可验证压缩文件的结构。有关压缩文件结构的详细信息，请参阅第 67 页的『IBM 4690 IBM MQ Managed File Transfer 配置压缩文件的结构』。

如果存在配置目录，那么在创建新配置之前会通过 **ftecfg** 命令备份该目录。有关更多信息，请参阅第 89 页的『备份数』。

ftecfg 命令在 IBM MQ Managed File Transfer 安装目录 `f:\adxetc\mft75\mqft` 下生成 `config`、`installations` 和 `logs` 目录。

生成的 `installations` 和 `logs` 目录基于压缩文件中指定的协调队列管理器名称和代理名称。只定义了一个协调队列管理器。可以定义一个或多个代理。创建的安装的名称始终为 `installation1`。

如果您对代理名称使用变量替换（例如，代理名称 `AGENT@N@S` 扩展为 `AGENTKD1234`），那么 **ftecfg** 命令创建的代理目录将使用因替换而完全扩展的名称。在磁盘上创建目录结构后，会更新 `agent.properties` 文件中的 `agentName` 属性，以与最终替换的代理名称相匹配。有关更多信息，请参阅第 67 页的『在 4690 OS 配置捆绑软件中定制代理名称』。

生成的输出具有以下目录结构：

```
mqmft (directory)
  AgentName1.rsp (file - generated by ftecfg)
  AgentName1.pc (file - generated by ftecfg)
  .
  .
  .
  (further files for agents if defined)
  .
  .
  .
  mqft (directory)
    config (directory)
    coordination-qmgr (directory)
    agents (directory)
    AgentName1 (directory - from the compressed file)
```

```

        agent.properties (properties file - from the compressed file)
        .
        .
        .
        (further agents if defined)
        .
        .
        .
        command.properties (properties file - from the compressed file)
        coordination.properties (properties file - from the compressed file)
installations (directory)
    installation1 (directory)
        installation.properties (properties file)
logs

    agents (directory)
        AgentName1 (directory - name taken from the compressed file)
        .
        .
        .
        (further agents if defined)
        .
        .
        .

```

以下示例显示了生成的命令结构的样本：

```

mqmft
  MyFirstAgent.rsp
  MyFirstAgent.pc
  MySecondAgent.rsp
  MySecondAgent.pc
mqft
  config
    QM_gbthink
    agents
      MyFirstAgent
      agent.properties
      MySecondAgent
      agent.properties
      command.properties
      coordination.properties
  installations
    installation1
      installation.properties
  logs
    agents
      MyFirstAgent
      MySecondAgent

```

备份数

每次运行 **ftecfg** 命令时，在将新配置抽取到安装目录之前，会对现有配置和日志文件进行备份。该备份使您能够在发现新配置的问题时还原到先前的配置。

将配置（f:\adxetc\mft75\mqft\config 目录中）备份到采用以下命名格式的文件中：

```
f:\adxetc\mft75\backup-timestamp-config.zip
```

将日志文件（f:\adxetc\mft75\mqft\logs 目录中）备份到采用以下命名格式的文件中：

```
f:\adxetc\mft75\backup-timestamp-logs.zip
```

对于关联的备份配置和日志文件对，时间戳记值匹配，因此您可以识别先前的配置和日志文件。

创建新备份时，删除最旧的备份之前最多可以进行 10 次备份。此限制确保系统上备份使用的磁盘空间量受限。但是，建议您监控备份以确保磁盘未被填满。

语法

ftecfg

►► **ftecfg** — *compressed_file_path* ►►

参数

compressed_file_path

必需。指向包含代理配置详细信息的压缩 .zip 文件的路径。

示例

在该示例中，命令 **ftecfg** 成功运行。已达到最大备份次数，因此将在创建新备份文件之前删除最旧的备份文件。

```
ftecfg config.zip
```

```
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
BFGCL0643I: The maximum number of configuration backups has been reached. The oldest will be
deleted.
BFGCL0644I: The backup file 'f:/adxetc/mft75/backup-20121122102214379-config.zip' has been
deleted.
BFGCL0644I: The backup file 'f:/adxetc/mft75/backup-20121122102214379-logs.zip' has been deleted.
BFGCL0645I: A backup of the product configuration has been created in file
'f:/adxetc/mft75/backup-20121127104428148-config.zip'.
BFGCL0646I: A backup of the product logs has been created in file
'f:/adxetc/mft75/backup-20121127104428148-logs.zip'.
BFGCL0610I: Agent FTEAGENT7777 successfully configured. ADXCHAIN.386 response file:
f:\adxetc\mft75\FTEAGENT7777.rsp
```

在该示例中，**ftecfg** 命令失败，因为找不到 `agent.properties` 中的绝对路径。

```
ftecfg config.zip
```

```
BFGCL0645I: A backup of the product configuration has been created in file
'f:/adxetc/mft75/backup-20130129080041321-config.zip'.
BFGCL0646I: A backup of the product logs has been created in file
'f:/adxetc/mft75/backup-20130129080041321-logs.zip'.
BFGCL0668E: The absolute path 'F:\mqmftcredentials.xml' supplied for property
'agentSslKeyStoreCredentialsFile' does not refer to an existing local file.
BFGCL0635E: Failed to create the Managed File Transfer configuration.
```

在该示例中，**ftecfg** 命令失败，因为找不到凭证文件中的绝对路径。

```
ftecfg config.zip
```

```
BFGCL0645I: A backup of the product configuration has been created in file
'f:/adxetc/mft75/backup-20130129080733868-config.zip'.
BFGCL0646I: A backup of the product logs has been created in file
'f:/adxetc/mft75/backup-20130129080733868-logs.zip'.
BFGCL0669E: The absolute path 'D:\truststore.jks' supplied in credentials file
'mqmftcredentials.xml' does not refer to an existing local file.
BFGCL0635E: Failed to create the Managed File Transfer configuration.
```

相关参考

第 67 页的『[IBM 4690 IBM MQ Managed File Transfer 配置压缩文件的结构](#)』

您可以通过将包含配置详细信息的压缩文件作为参数传递至 **ftecfg** 命令，从而在 IBM 4690 系统上创建 IBM MQ Managed File Transfer 配置。

第 85 页的『[fteBundleConfiguration](#)（创建 IBM MQ Managed File Transfer IBM 4690 代理配置 .zip 文件）』

使用 **fteBundleConfiguration** 命令可将指定目录中的 IBM MQ Managed File Transfer 配置树捆绑成 .zip 文件。

ftediag（生成关于 IBM 4690 系统的 IBM MQ Managed File Transfer 代理诊断信息）

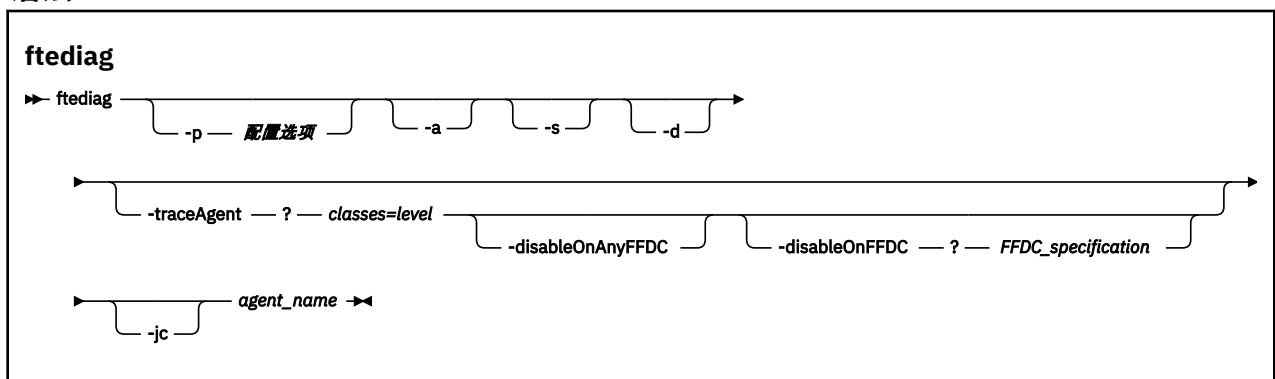
ftediag 命令可为 IBM 4690 系统上的 IBM MQ Managed File Transfer 代理生成诊断信息。

用途

使用 **ftediag** 命令可从 IBM 4690 系统上的 IBM MQ Managed File Transfer 代理程序启用，禁用和收集诊断跟踪信息。诊断信息包含代理的状态，并且还可以生成 Javacore 文件。即使当代理未与其队列管理器连接时，此命令也有效。诊断信息将输出到屏幕上。如果您指定 **-jc** 参数，那么文件系统中 Javacore 文件的位置也会显示在屏幕上。

仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。有关更多信息，请参阅第 591 页的『[agent.properties](#) 文件』。

语法



参数

-p configuration options

可选。指定由命令使用的配置选项集。按照约定，该选项是协调队列管理器的名称。如果未指定该参数，那么将使用缺省配置选项。

-a

可选。指定所有诊断选项，选中 **-s**、**-d** 和 **-jc**。

-s

可选。显示代理的状态。如果未指定任何其他选项，那么该选项为缺省选项。

-d

可选。指定显示 **agent_name** 的诊断信息。可以在代理正在运行时和在本地系统上使用此参数。

-traceAgent classes=level

可选。将代理跟踪设置到的级别和将该跟踪应用到的类。指定以下格式：

```
classes=level
```

例如：

```
com.ibm.wmqfte=all
```

指定您希望将该跟踪级别应用到的类规范的逗号分隔列表。如果未指定该参数，那么跟踪级别将应用于所有代理类。

如果 (*classes*) 以加号 (+) 开头，那么将向目前跟踪的所有现有跟踪类添加加号后面的跟踪类列表。

有效的跟踪级别选项如下所述，按照跟踪文件大小的升序和详细信息列出：

关

关闭代理跟踪，但继续将信息写入日志文件。

flow

捕获与代理中处理流相关联的跟踪点的数据。

moderate

捕获跟踪中的适量诊断信息。

详细

捕获跟踪中的大量诊断信息。

all

捕获跟踪中所有诊断信息。

-disableOnAnyFFDC

可选。如果指定了该参数，那么当代理生成“首次故障数据捕获”(FFDC) 文件时，跟踪将被禁用。此参数仅在也指定 **-traceAgent** 参数的情况下有效。

您只可以指定 **-disableOnAnyFFDC** 和 **-disableOnFFDC** 参数中的一个。

-disableOnFFDC *FFDC_specification*

可选。如果指定了该参数，那么当代理生成与 *FFDC_specification* 匹配的“首次故障数据捕获”(FFDC) 文件时，跟踪将被禁用。此参数仅在也指定 **-traceAgent** 参数的情况下有效。*FFDC_specification* 是以逗号分隔的以下一个或多个选项的列表：

class_name

FFDC 继承的类的名称。例如，com.ibm.wmqfte.classA。

class_name:probe_ID

FFDC 继承的类中的类的名称和位置的探测器标识。例如，com.ibm.wmqfte.classB:1。例如：

```
-disableonFFDC com.ibm.wmqfte.transfer,com.ibm.wmqfte:1
```

您只可以指定 **-disableOnAnyFFDC** 和 **-disableOnFFDC** 参数中的一个。

-jc

可选。请求代理来生成 Javacore 文件。IBM 服务团队可能会要求您使用该参数来运行此命令，以帮助诊断问题。如果运行带有 **-jc** 参数的命令，那么生成的 Javacore 文件的位置会显示在屏幕上。

agent_name

必需。要从中抽取诊断信息的 IBM MQ Managed File Transfer 代理的名称。

示例

在该示例中，只有 Javacore 参数与 **ftediag** 命令一起用于代理 FTEAGENT1997：

```
ftediag -jc FTEAGENT1997
```

```
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED  
BFGCL0549I: The javacore request was successfully sent to agent 'FTEAGENT1997'.
```


The created javacore file name is: /cdrive/f_drive/adxetc/java/core/javacore.20120927.073416.31718.0001.txt

在该示例中，设置类 `com.ibm.wmqfte` 的代理跟踪级别以捕获跟踪中的所有诊断信息，并且为代理 `FTEAGENT1997` 创建诊断和 Javacore 文件。

```
ftediag -d -jc -traceAgent com.ibm.wmqfte=all FTEAGENT1997
```

```
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
BFGCL0549I: The javacore request was successfully sent to agent 'FTEAGENT1997'.
The created javacore file name is: /cdrive/f_drive/adxetc/mqmft/mqft/logs/MUNGEE
/agents/FTEAGENT1997/javacore.20121101.123902.5728.0001.txt
Agent Information:
  Name: FTEAGENT1997
  Type: Standard
  Description:
  Operating System: 4690 OS
  Host Name: KD
  Time Zone: Greenwich Mean Time
  Product Version: 7.5.0.2
  Build Level: f000-personal-20121031-0905
  Trace Level: No trace specified
  Trace FFDC: No FFDC specified

Agent Controller Information:
  Status: STARTED_BY_AGENT
  Status Details: The agent has started the process controller.
  Agent Restarts within Interval: 0
  Total Agent Restart Count: 0

Agent Availability Information:
  Status: READY
  Status Details: The agent is running and is publishing its status at regular intervals. The last update was received within the expected time period. The agent is ready to process transfers, but none are currently in progress.

Queue Manager Information:
  Name: MUNGEE
  Transport: Client
  Host: 192.168.255.1
  Port: 1414
  Channel: SYSTEM.DEF.SVRCONN
  Last Status Reported: UNKNOWN
  Status Details: Information about the queue manager is not available because the agent has a client connection to the queue manager.

Maximum Number of Running Source Transfers: 25
Maximum Number of Queued Source Transfers: 1000
Source Transfer States:
  No current transfers

Maximum Number of Running Destination Transfers: 25
Destination Transfer States:
  No current transfers

Agent Diagnostic Information:
  Diagnostic Properties File name: f:\adxetc\mft75\mqft\logs\MUNGEE\agents\FTEAGENT1997\logs\diagnostics.20121101.123904.0909.1.properties

Command Handler Diagnostics:
  Last Command Queue Read Time: 2012-11-01T12:38:41.286Z
  Pending Command Queue Size: 0

Command Handler Worker Thread 0 Diagnostics:
  Status: Waiting

Command Handler Worker Thread 1 Diagnostics:
  Status: Waiting

Command Handler Worker Thread 2 Diagnostics:
  Status: Waiting
```

```

Command Handler Worker Thread 3 Diagnostics:
Status:                                     Waiting

Command Handler Worker Thread 4 Diagnostics:
Status:                                     Waiting

File Transfer Diagnostics:
Source Transfers:                           0
Destination Transfers:                       0

```

在该示例中，**-a** 参数与 **ftediag** 命令一起使用以选择代理 FTEAGENT1997 的状态信息、诊断文件和 Javacore 文件：

```
ftediag -a FTEAGENT1997
```

```

5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
BFGCL0549I: The javacore request was successfully sent to agent 'FTEAGENT1997'.
The created javacore file name is: /cdrive/f_drive/adxetc/java/core/javacore.201
20927.073454.31718.0002.txt
BFGCL0598I: The diagnostics request was successfully sent to agent 'FTEAGENT1997'.
The created diagnostics file name is: /cdrive/f_drive/adxetc/mqmft/mqft/logs/
MUNGEE/agents/FTEAGENT1997/logs/diagnostics.20120927.073454.0859.0.properties
Agent Information:
Name:                                     FTEAGENT1997
Type:                                     Standard
Description:
Operating System:                         4690 OS
Host Name:                               KD
Time Zone:                               Greenwich Mean Time
Product Version:                          7.5.0.2
Build Level:                             f000-personal-20120925-1131
Trace Level:                             No trace specified
Trace FFDC:                              No FFDC specified

Agent Controller Information:
Status:                                  STARTED
Status Details:                          The agent process controller has started
                                           the agent process.
Agent Restarts within Interval:           0
Total Agent Restart Count:                0

Agent Availability Information:
Status:                                  READY
Status Details:                          The agent is running and is publishing
                                           its status at regular intervals. The last
                                           update was received within the expected
                                           time period. The agent is ready to
                                           process transfers, but none are currently
                                           in progress.

Queue Manager Information:
Name:                                     MUNGEE
Transport:                               Client
Host:                                    192.168.255.1
Port:                                    1414
Channel:                                 SYSTEM.DEF.SVRCONN
Last Status Reported:                    UNKNOWN
Status Details:                          Information about the queue manager is
                                           not available because the agent has a
                                           client connection to the queue manager.

Maximum Number of Running Source Transfers: 25
Maximum Number of Queued Source Transfers: 1000
Source Transfer States:
No current transfers

Maximum Number of Running Destination Transfers: 25
Destination Transfer States:
No current transfers

```

卸载（从 IBM 4690 系统卸载 IBM MQ Managed File Transfer）

uninstall 命令从 IBM 4690 商店控制器卸载 IBM MQ Managed File Transfer。

用途

使用 **uninstall** 命令可卸载 IBM 4690 系统中的 IBM MQ Managed File Transfer 产品文件。您还可以选择使用此命令卸载配置和日志文件。

语法

uninstall

► **uninstall** — **-a** — **-c** ►

参数

-a

必需：您必须指定 **-a** 参数或 **-c** 参数。卸载系统中的所有 IBM MQ Managed File Transfer 产品文件、配置文件和日志文件。

-c

必需：您必须指定 **-c** 参数或 **-a** 参数。卸载所有 IBM MQ Managed File Transfer 产品文件，但不卸载配置文件或日志文件。

示例

在该示例中，卸载所有 IBM MQ Managed File Transfer 产品文件，但保留创建的配置文件和日志文件。

```
f:
cd \adxetc\mft75
uninstall -c
```

相关任务

第 65 页的『[卸载 4690 系统中的 IBM MQ Managed File Transfer](#)』

要从 4690 存储控制器卸载 IBM MQ Managed File Transfer 代理，请完成以下步骤：

对 IBM 4690 系统进行故障诊断

使用以下参考信息可帮助您诊断从 IBM 4690 系统返回的错误。

- [第 95 页的『对 IBM 4690 程序调用进行故障诊断』](#)

对 IBM 4690 程序调用进行故障诊断

可以在前台 (调用类型: EXECUTABLE) 或后台 (调用类型: OS4690BACKGROUND) 运行 IBM 4690 系统上的传输程序调用。这些调用类型支持运行本机 IBM 4690 应用程序和批处理命令脚本。如果您运行批处理脚本，那么将带有 -C 选项的 COMMAND.286 应用程序用作应用程序，将批处理脚本和任何指定的参数作为参数传递。

IBM 4690 应用程序故障

如果应用程序出于任何原因启动失败，那么命令调用结果会指示一个错误并给出原因。如果应用程序为批处理脚本，但脚本格式错误，那么即使失败，也会报告为成功运行。在以下示例中，批处理脚本错误地使用分号 (;) 作为命令分隔符：

```
echo "First Echo" > stdout1.txt ; echo "Second Echo" > stdout2.txt
```

如果此批处理脚本作为 IBM MQ Managed File Transfer 程序调用运行，那么结果报告为成功。但是，调用的以下错误文本将出现在传输日志中：

```
STDOUT: The STDOUT parameter was previously redirected.  
STDOUT can only be redirected to one place at a time.
```

这是因为正在运行批处理脚本的 COMMAND.286 应用程序针对此情况返回成功 (0) 退出码。出于这个原因，很有必要先验证要作为程序调用运行的这些批处理脚本，然后再使用。另外，如果批处理脚本在前台运行，那么它应该调用使用相应代码的 EXIT 命令，以便传输状态反映批处理脚本调用的结果。

IBM MQ Managed File Transfer 的安全性概述

安装后没有进行任何修改的情况下，IBM MQ Managed File Transfer 的安全级别适用于受保护环境中的测试或评估目的。但是在生产环境中，必须考虑适当控制可以启动文件传输操作的人员、可以读和写正在传输的文件的人员以及如何保护文件完整性。

相关概念

[第 99 页的『沙箱』](#)

您可以限制在传输过程中代理可访问的文件系统区域。将代理限制到的区域称为沙箱。您可以对代理或请求传输的用户应用限制。

[第 106 页的『保证 Web 网关的安全』](#)

可以使用一些方法来保护 Web 网关的安全。使用 Web 网关之前，必须先执行某些安全步骤。其他步骤可选，可增加 Web 网关和 IBM MQ Managed File Transfer 网络的安全性，但使用 Web 网关并非必须执行这些步骤。

相关任务

[第 103 页的『为 IBM MQ Managed File Transfer 配置 SSL 或 TLS 加密』](#)

将 SSL 或 TLS 与 IBM MQ 和 IBM MQ Managed File Transfer 一起使用可防止在代理与队列管理器之间建立未经授权的连接，并可对代理与队列管理器之间的消息通信进行加密。

[第 105 页的『将 IBM MQ Advanced Message Security 与 IBM MQ Managed File Transfer 一起使用』](#)

IBM MQ Advanced Message Security 提供了 IBM MQ Managed File Transfer 中的消息流量（尤其是队列上的不活动数据）的安全性。

相关参考

[第 433 页的『特定于 IBM MQ Managed File Transfer 的资源的组权限』](#)

无需针对可能涉及到的所有对象而为个别用户授予权限，而是请配置以下两个安全组以用于管理 IBM MQ Managed File Transfer 访问控制：FTEUSER 和 FTEAGENT。IBM MQ 管理员应负责创建并填充这些组。管理员可选择扩展或修改此处所述的建议配置。

[第 432 页的『特定于 IBM MQ Managed File Transfer 的资源的权限』](#)

针对任何文件传输请求，代理进程都需要其本地文件系统的某个访问权级别。此外，与代理进程关联的用户标识以及与执行文件传输操作的用户关联的用户标识都必须有权使用某些 IBM MQ 对象。

[第 443 页的『文件系统访问权限』](#)

针对任何文件传输请求，代理进程都需要其本地文件系统的某个访问权级别。

[第 97 页的『IBM MQ Managed File Transfer 和 IBM MQ 连接认证』](#)

IBM MQ Managed File Transfer V8.0 支持 IBM MQ V8.0 安全性功能，缺省模式为禁用。如果关联的队列管理器启用了安全性并要求提供凭证详细信息（用户标识和密码），那么必须启用此功能，才能成功连接到队列管理器。

[第 443 页的『commandPath 属性』](#)

使用 commandPath 属性以指定 IBM MQ Managed File Transfer 可从中运行命令的位置。设置该属性时务必谨慎，因为能够向代理发送命令的远程客户机系统可以有效调用指定的某个 commandPath 中的任何命令。

[第 442 页的『用于发布日志和状态消息的权限』](#)

代理可发出在协调队列管理器上发布的各种日志、进展和状态消息。这些消息的发布受 IBM MQ 安全模型控制，在某些情况下，您可能需要执行进一步配置才能启用发布。

IBM MQ Managed File Transfer 和 IBM MQ 连接认证

IBM MQ Managed File Transfer V8.0 支持 IBM MQ V8.0 安全性功能，缺省模式为禁用。如果关联的队列管理器启用了安全性并要求提供凭证详细信息（用户标识和密码），那么必须启用此功能，才能成功连接到队列管理器。

很多 IBM MQ Managed File Transfer 命令支持以下方法：

由命令行参数提供详细信息。

可以使用参数 **-mquserid** 和 **-mqpassword** 来指定凭证详细信息。如果没有提供 **-mqpassword**，将在未显示输入的地方要求用户输入密码。

由凭证文件 **MQMFTCredentials.xml** 提供详细信息。

可以在 MQMFTCredentials.xml 文件中将凭证详细信息预定义为明文或模糊文本。

MQMFTCredentials.xml 文件的位置由属性值来定义：

表 6:		
类别	属性文件	属性名
显示/列出命令	协调属性	coordinationQMGrAuthenticationCredentialsFile
修改/创建命令	命令属性	connectionQMGrAuthenticationCredentialsFile
代理/清除代理	代理属性	agentQMGrAuthenticationCredentialsFile
记录器	记录器属性	loggerQMGrAuthenticationCredentialsFile

QMGr 定义一对凭证，格式如下：

```
<tns:qmgr mquserid="MQ User ID" mqpassword="MQ Password" name="QMGr" user="user running command" />
```

用户属性为可选，如果未提供，那么将应用到所有用户。

优先顺序

确定凭证详细信息的过程为：

1. 命令行参数。
2. 关联的队列管理器和运行命令的用户的 MQMFTCredentials.xml 索引。
3. 关联的队列管理器的 MQMFTCredentials.xml 索引。
4. 未提供凭证详细信息的缺省向后兼容性方式，用于提供与 IBM MQ 前发行版的兼容性。

注意：

- **fteStartAgent** 和 **fteStartLogger** 命令不支持命令行参数 **-mquserid** 或 **-mqpassword**，并且只能使用 MQMFTCredentials.xml 文件来指定凭证详细信息。
- 在 z/OS 上，即使用户的密码中有小写字母，密码也必须为大写。例如，如果用户的密码是“password”，那么必须输入为“PASSWORD”。

相关任务

第 97 页的『配置 MFT 安全性』

相关参考

第 445 页的『MFT 命令和进程与队列管理器的连接关系』
每个 Managed File Transfer 拓扑都包含多个不同组件。

配置 MFT 安全性

关于此任务

MQMFTCredentials.xml 文件包含 XML 格式的凭证信息。第 873 页的『MQMFT 凭证文件格式』主题中定义了该 XML 文件的格式。

在 z/OS 平台上，可以使用分区数据集的成员来存储 MQMFTCredentials.xml 文件。

在 IBM MQ 资源管理器的 IBM MQ Managed File Transfer 插件中，用于连接到 IBM MQ Managed File Transfer 协同队列管理器的凭证详细信息取决于配置类型：

全局（本地磁盘上的配置）

将使用在协同和命令属性中指定的凭证文件。

本地（在 IBM MQ 资源管理器中定义）：

将使用 IBM MQ 资源管理器中的关联队列管理器的连接详细信息的属性。

V 8.0.0.7 启用 MQCSP 认证方式

从 IBM MQ 8.0.0 修订包 7 开始，您可以对与协调队列管理器或命令队列管理器相连的 MQ Explorer MFT 插件的连接认证启用 MQCSP 认证方式。也可以对与协调队列管理器或命令队列管理器相连的 Managed File Transfer 代理的连接认证启用 MQCSP 认证方式。

关于此任务

如果使用 MQ Explorer Managed File Transfer 插件或者具有使用 CLIENT 传输连接到队列管理器并指定密码的 Managed File Transfer 代理，那么当指定的密码长度大于 12 个字符时，代理将不会向队列管理器认证。这是因为代码不会使用 MQCSP 认证，而是使用兼容性方式进行认证，这会将密码长度限制为 12 个字符。

从 IBM MQ 8.0.0 修订包 7 开始，您可以禁用缺省兼容性方式并启用 MQCSP 认证方式。

过程

- 要对 MQ Explorer 中的协调队列管理器或命令队列管理器禁用兼容性方式并启用 MQCSP 认证，请完成以下步骤：
 - 选择要连接到的队列管理器。
 - 单击右键，并从弹出菜单中选择**连接详细信息 -> 属性**。
 - 单击**用户标识**选项卡。
 - 确保选中**启用用户标识**，并清除**用户标识兼容性方式**复选框。
- 要禁用兼容性方式并对 Managed File Transfer 代理程序启用 MQCSP 认证，请将参数 **useMQCSPAAuthentication** 添加到相关用户的 MQMFTCredentials.xml 文件中，并将其设置为 true。

必须将此参数设置为 true。如果未指定此参数，缺省情况下会将其设置为 false，并且将使用兼容性方式向队列管理器认证用户。

以下示例显示了如何设置 MQMFTCredentials.xml 文件中的 **useMQCSPAAuthentication** 参数：

```
<tns:qmgr name="CoordQueueMgr" user="ernest" mqUserId="ernest"
mqPassword="AveryL0ngPassw0rd2135" useMQCSPAAuthentication="true"/>
```

相关参考

第 97 页的『IBM MQ Managed File Transfer 和 IBM MQ 连接认证』

IBM MQ Managed File Transfer V8.0 支持 IBM MQ V8.0 安全性功能，缺省模式为禁用。如果关联的队列管理器启用了安全性并要求提供凭证详细信息（用户标识和密码），那么必须启用此功能，才能成功连接到队列管理器。

第 873 页的『MQMFT 凭证文件格式』

MQMFTCredentials.xml 文件包含敏感的用户标识和密码信息。MQMFTCredentials.xml 文件中的元素必须符合 MQMFTCredentials.xsd 模式。凭证文件的安全性由用户负责。

相关信息

[MQCSP 密码保护](#)

沙箱

您可以限制在传输过程中代理可访问的文件系统区域。将代理限制到的区域称为沙箱。您可以对代理或请求传输的用户应用限制。

当代理是协议网桥代理或 Connect:Direct 网桥代理时，不支持沙箱。您不能将代理程序沙箱用于需要传输至 WebSphere MQ 队列或从中向外传输的代理程序。

相关参考

[第 99 页的『使用代理沙箱』](#)

要向 IBM MQ Managed File Transfer 添加额外的安全性级别，可以限制代理可以访问的文件系统区域。

[第 100 页的『使用用户沙箱』](#)

您可以根据请求传输的 MQMD 用户名来限制可在其中传输文件的文件系统的区域。

使用代理沙箱

要向 IBM MQ Managed File Transfer 添加额外的安全性级别，可以限制代理可以访问的文件系统区域。

不能将代理沙箱用于需要与 IBM MQ 队列传输数据的代理。可使用沙箱限制对 IBM MQ 队列的访问，而不使用作为任何沙箱需求推荐解决方案的用户沙箱。有关用户沙箱的更多信息，请参阅[第 100 页的『使用用户沙箱』](#)

要启用代理沙箱，请向要限制的代理的 `agent.properties` 文件添加以下属性：

```
sandboxRoot=[!]restricted_directory_name<separator>...<separator>[!]restricted_directory_name
```

其中：

- `restricted_directory_name` 是要允许或拒绝的目录路径。
- `!` 是可选的，并指定拒绝 (排除) `restricted_directory_name` 的以下值。如果未指定 `!`，那么表示 `restricted_directory_name` 是一个允许 (包含) 路径。
- `<separator>` 是特定于平台的分隔符。

例如，如果要仅限制 AGENT1 对 `/tmp` 目录的访问权，但不允许访问子目录 `private`，可在属于 AGENT1 的 `agent.properties` 文件中按照如下方式设置属性：`sandboxRoot=/tmp:!/tmp/private`。

[高级代理属性](#)中描述了 `sandboxRoot` 属性。

在协议网桥代理或 Connect:Direct 网桥代理上不支持代理和用户沙箱。

在 UNIX，Linux 和 Windows 平台上的沙箱中工作

在 UNIX，Linux 和 Windows 平台上，沙箱限制 IBM MQ Managed File Transfer 代理程序可以读取和写入的目录。激活沙箱后，IBM MQ Managed File Transfer 代理可以读写指定为“allowed”的目录及指定目录包含的任何子目录，除非在 `sandboxRoot` 中将这子目录指定为“denied”。IBM MQ Managed File Transfer 沙箱不优先于操作系统安全性。已启动 IBM MQ Managed File Transfer 代理的用户必须具有相应的操作系统级别访问权，以便能够对目录进行读写。如果链接到的目录在指定的 `sandboxRoot` 目录（及子目录）之外，那么不遵循到目录的符号链接。

在 z/OS 上的沙箱中工作

在 z/OS 上，沙箱限制了 IBM MQ Managed File Transfer 代理可读写的数据集名称限定符。已启动 IBM MQ Managed File Transfer 代理的用户必须对任何涉及的数据集具有正确的操作系统权限。如果使用双引号将 `sandboxRoot` 数据集名称限定符括起来，那么该值遵循标准的 z/OS 约定，并视为标准值。如果省略双引号，那么 `sandboxRoot` 将以当前用户标识为前缀。例如，如果将 `sandboxRoot` 属性设置为 `sandboxRoot=//test`，那么代理程序可以访问以下数据集 (采用标准 z/OS 表示法) `//<username>.test.**` 在运行时，如果完全解析的数据集名称的初始级别与 `sandboxRoot` 不匹配，那么将拒绝传输请求。

在 IBM i 系统上的沙箱中工作

对于 IBM i 系统上集成文件系统上的文件，沙箱限制了 IBM MQ Managed File Transfer 代理可读写的目录。激活沙箱后，IBM MQ Managed File Transfer 代理可以读写指定为“allowed”的目录及指定目录包含的任何子目录，除非在 sandboxRoot 中将这子目录指定为“denied”。IBM MQ Managed File Transfer 沙箱不优先于操作系统安全性。已启动 IBM MQ Managed File Transfer 代理的用户必须具有相应的操作系统级别访问权，以便能够对目录进行读写。如果链接到的目录在指定的 sandboxRoot 目录（及子目录）之外，那么不遵循到目录的符号链接。

在 IBM 4690 上的沙箱中工作

有关如何在 IBM 4690 上解释 sandboxRoot 属性中指定的路径的信息，请参阅第 84 页的『在 IBM 4690 系统上的沙箱中工作』。

使用用户沙箱

您可以根据请求传输的 MQMD 用户名来限制可在其中传输文件的文件系统的区域。

当代理是协议网桥代理或 Connect:Direct 网桥代理时，不支持用户沙箱。

要启用用户沙箱，请为您想要限制的代理的 agent.properties 文件添加以下属性：

```
userSandboxes=true
```

如果此属性存在并且设置为 true，那么代理程序会使用 MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml 文件中的信息来确定请求传输的用户可以访问文件系统的哪些部分。

UserSandboxes.xml XML 由 <agent> 元素组成，该元素包含零个或多个 <sandbox> 元素。这些元素描述哪些规则适用于哪些用户。<sandbox> 元素的 user 属性是一种模式，用于匹配请求的 MQMD 用户。

代理会定期重新装入 UserSandboxes.xml 文件，对该文件的任何有效更改都会影响代理的行为。缺省重新装入时间间隔为 30 秒。此时间间隔可通过在 agent.properties 文件中指定代理程序属性 xmlConfigReloadInterval 进行更改。

如果指定 userPattern="regex" 属性或值，那么 user 属性将解释为 Java 正则表达式。有关更多信息，请参阅第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』。

如果不指定 userPattern="regex" 属性或值，那么 user 属性会被解释为具有以下通配符的模式：

- 星号 (*), 表示零个或多个字符
- 问号 (?), 表示仅一个字符

将按照 <sandbox> 元素在文件中的列示顺序来执行匹配项。将仅使用第一个匹配项，将忽略文件中的所有其余潜在匹配项。如果文件中指定的 <sandbox> 元素与传输请求消息的关联 MQMD 用户均不匹配，那么传输将无法访问文件系统。如果在 MQMD 用户名与 user 属性之间发现匹配项，那么该匹配项将标识 <sandbox> 元素中应用于传输的一组规则。这组规则用于确定在传输过程中可以对哪些文件或数据集执行读或写操作。

每组规则都可以指定 <read> 元素（标识可以读取的文件）和一个 <write> 元素（标识可以写入的文件）。如果在规则集中省略 <read> 或 <write> 元素，那么假定相应地不允许与规则集相关联的用户执行任何读或写操作。

注：在 UserSandboxes.xml 文件中，<read> 元素必须在 <write> 元素之前，<include> 元素必须在 <exclude> 元素之前。

每个 <read> 或 <write> 元素都包含一个或多个模式，用来确定文件是否在沙箱中以及是否可以传输。请使用 <include> 和 <exclude> 元素来指定这些模式。<include> 或 <exclude> 元素的 name 属性指定要匹配的模式。可选属性 type 指定名称值是文件还是队列模式。如果未指定 type 属性，那么代理程序会将该模式视为文件或目录路径模式。例如：

```
<tns:read>
  <tns:include name="/home/user/**"/>
  <tns:include name="USER.**" type="queue"/>
```

```
<tns:exclude name="/home/user/private/**"/>
</tns:read>
```

代理会使用 `<include>` 和 `<exclude>` `name` 模式来确定可以对哪些文件、数据集或队列执行读或写操作。如果规范文件路径、数据集或队列名称与至少一个包含的模式匹配，并且不与任何排除的模式匹配，那么允许执行操作。使用 `<include>` 和 `<exclude>` 元素的 `name` 属性指定的模式会使用适用于平台（在其上运行代理）的路径分隔符和约定。如果您指定相对文件路径，那么将相对于代理的 `transferRoot` 属性来解析该路径。

指定队列限制时，支持语法 `QUEUE@QUEUEMANAGER`，且规则如下：

- 如果条目中缺失 `@` 字符，那么该模式将被视为可在任何条目管理器上访问的队列名称。例如，如果模式为 `name`，那么该模式将同样被视为 `name@**`。
- 如果 `@` 字符是条目中的第一个字符，那么模式将视为队列管理器名称，可以访问队列管理器上所有队列。例如，如果模式为 `@name`，那么该模式将同样被视为 `**@name`。

如果您将以下通配符指定为 `<include>` 和 `<exclude>` 元素的 `name` 属性的一部分，那么这些通配符具有特殊含义：

单个星号与目录名称中的零个或零个以上字符匹配，或者与数据集名称或队列名称的限定符中的零个或零个以上字符匹配。

?

问号只与目录名称中的一个字符匹配，或者只与数据集名称或队列名称的限定符中的一个字符匹配。

两个星号字符与零个或零个以上目录名称匹配，或者与数据集名称或队列名称中的零个或零个以上限定符匹配。此外，以路径分隔符结尾的路径在路径末尾添加了隐式 `“***”`。因此，`/home/user/` 与 `/home/user/**` 相同。

例如：

- `/**/test/**` 匹配路径中含有 `test` 目录的任何文件
- `/test/file?` 匹配 `/test` 目录中以 `file` 字符串开头并后跟任何单个字符的任何文件
- `c:\test\*.txt` 匹配 `c:\test` 目录中扩展名为 `.txt` 的任何文件
- `c:\test\**\*.txt` 与 `'c:\test` 目录中的任何文件或其某个具有 `.txt` 扩展名的子目录相匹配
- `//'TEST.*.DATA'` 匹配第一位限定符为 `TEST` 且第二位和第三位限定符为 `DATA` 的任何数据集。
- `TEST.*.QUEUE@QM1` 匹配队列管理器 `QM1` 上第一位限定符为 `TEST` 且第二位和第三位限定符为 `QUEUE` 的任何队列。

符号链接

您必须在 `<include>` 和 `<exclude>` 元素中指定硬链接，以便完全解析在 `UserSandboxes.xml` 文件的文件路径中使用的任何符号链接。例如，如果您有一个符号链接，其中 `/var` 映射到 `/SYSTEM/var`，那么必须将此路径指定为 `<tns:include name="/SYSTEM/var"/>`，否则预期的传输将因用户沙箱安全性错误而失败。

IBM 4690 系统上的路径

有关如何在 IBM 4690 上解释 `UserSandboxes.xml` 文件中指定的路径的信息，请参阅第 84 页的『在 IBM 4690 系统上的沙箱中工作』。

示例

要允许具有 MQMD 用户名 `guest` 的用户从运行代理 `AGENT_JUPITER` 的系统上的 `/home/user/public` 目录或其任何子目录传输任何文件，请将以下 `<sandbox>` 元素添加到 `AGENT_JUPITER` 的配置目录中的文件 `UserSandboxes.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<tns:userSandboxes
  xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
  <tns:agent>
    <tns:sandbox user="guest">
      <tns:read>
        <tns:include name="/home/user/public/**"/>
      </tns:read>
    </tns:sandbox>
  </tns:agent>
</tns:userSandboxes>
```

示例

要允许 MQMD 用户名为 account 并且后跟单个数字的任何用户（例如，account4），请完成以下操作：

- 从 /home/account 目录或其任何子目录中传输文件（运行代理 AGENT_SATURN 的系统上的 /home/account/private 目录除外）
- 在运行代理 AGENT_SATURN 的系统上，将任何文件传输到 /home/account/output 目录或其任何子目录。
- 从本地队列管理器上的队列中读取消息，以前缀 ACCOUNT. 开头，除非它以 ACCOUNT.PRIVATE. 开头（即具有第二个级别的 PRIVATE）。
- 将数据传输到任何队列管理器上以前缀 ACCOUNT.OUTPUT. 开头的队列。

将以下 <sandbox> 元素添加到文件 UserSandboxes.xml 的 AGENT_SATURN 配置目录中，

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
  xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
  <tns:agent>
    <tns:sandbox user="account[0-9]" userPattern="regex">
      <tns:read>
        <tns:include name="/home/account/**"/>
        <tns:include name="ACCOUNT.**" type="queue"/>
        <tns:exclude name="ACCOUNT.PRIVATE.**" type="queue"/>
        <tns:exclude name="/home/account/private/**"/>
      </tns:read>
      <tns:write>
        <tns:include name="/home/account/output/**"/>
        <tns:include name="ACCOUNT.OUTPUT.**" type="queue"/>
      </tns:write>
    </tns:sandbox>
  </tns:agent>
</tns:userSandboxes>
```

► V8.0.0.6 针对通配符传输的额外检查

从 IBM MQ 8.0.0 修订包 6 开始，如果已使用用户或代理沙箱配置代理以便限制代理可将文件传输到以及从中传输文件的位置，那么可以指定对该代理的通配符传输执行额外检查。

additionalWildcardSandboxChecking 属性

要对通配符传输启用额外检查，请向您想要检查的代理的 agent.properties 文件中添加以下属性：

```
additionalWildcardSandboxChecking=true
```

如果将此属性设置为 true，并且代理发出传输请求以尝试读取已定义沙箱外部的用于进行通配符文件匹配的位置，那么传输将失败。如果一个传输请求中存在多个传输，并且其中一个请求因尝试读取沙箱外部的位而失败，那么整个传输也将失败。如果检查失败，将在错误消息中提供失败原因。

如果在代理的 agent.properties 文件中省略了 additionalWildcardSandboxChecking 属性或者将其设置为 false，那么将不会对该代理的通配符传输进行额外检查。

通配符检查的错误消息

从 V 8.0.0 修订包 6 开始，已更改了对已配置沙箱位置外部的的位置发出通配符传输请求时报告的消息。

当传输请求中的通配符文件路径位于受限沙箱外部时，将显示以下消息：

BFGSS0077E: 读取文件路径 <path> 的尝试遭到拒绝。
该文件路径在受限传输沙箱外部。

当多传输请求内的传输包含路径位于受限沙箱外部的通配符传输请求时，将显示以下消息：

BFGSS0078E: Attempt to read file path: <path> has been ignored as another transfer
受管传输中的项尝试在受限传输沙箱外部读取。

当文件位于受限沙箱外部时，将显示以下消息：

BFGSS0079E: 读取文件 <file path> 的尝试遭到拒绝。
因为该文件在受限传输沙箱外部。

多传输请求中显示以下消息，其中另一个通配符传输请求导致此请求被忽略：

BFGSS0080E: Attempt to read file: <file path> has been ignored as another transfer
受管传输中的项尝试在受限传输沙箱外部读取。

对于不包含通配符的单文件传输，传输涉及沙箱外部的文件时报告的消息与较早发行版中一样：

失败，BFGI00056E: 读取文件“<FILE>”的尝试遭到拒绝。
因为该文件在受限传输沙箱外部。

相关参考

[第 100 页的『使用用户沙箱』](#)

您可以根据请求传输的 MQMD 用户名来限制可在其中传输文件的文件系统的区域。

[第 99 页的『使用代理沙箱』](#)

要向 IBM MQ Managed File Transfer 添加额外的安全性级别，可以限制代理可以访问的文件系统区域。

[第 591 页的『agent.properties 文件』](#)

每个代理都有自己的属性文件 agent.properties，其中必须包含代理用于连接到其队列管理器的信息。
agent.properties 文件还可以包含用于更改代理程序行为的属性。

为 IBM MQ Managed File Transfer 配置 SSL 或 TLS 加密

将 SSL 或 TLS 与 IBM MQ 和 IBM MQ Managed File Transfer 一起使用可防止在代理与队列管理器之间建立未经授权的连接，并可对代理与队列管理器之间的消息通信进行加密。

开始之前

SSL 加密仅对队列管理器之间以及队列管理器与代理之间的通道上的消息进行加密。如果要在消息位于队列上时对其进行加密，那么必须使用 IBM MQ Advanced Message Security。有关更多信息，请参阅[第 105 页的『将 IBM MQ Advanced Message Security 与 IBM MQ Managed File Transfer 一起使用』](#)。

关于此任务

有关将 SSL 与 IBM MQ 一起使用的常规信息，请参阅[使用 SSL 或 TLS](#)。在 IBM MQ 术语中，IBM MQ Managed File Transfer 是标准 Java 客户机应用程序。

遵循以下步骤将 SSL 与 IBM MQ Managed File Transfer 一起使用：

过程

1. 创建信任密钥库文件和（可选）密钥库文件（这些文件可以是相同的文件）。如果不需要客户机认证（即，通道上 SSLCAUTH=OPTIONAL），那么不需要提供密钥库。您仅需要信任库来认证队列管理器的证书。

用于为信任库和密钥库创建证书的密钥算法必须是 RSA 才能使用 IBM MQ。

如果需要有关如何创建信任库文件和密钥库文件的指示信息，请参阅 IBM Developer 文章“[在 WebSphere MQ File Transfer Edition 中配置安全套接字层连接](#)”，或参阅 [Oracle keytool 文档](#) 中有关 keytool 的信息。
2. 设置 IBM MQ 队列管理器以使用 SSL。

有关设置队列管理器以使用 SSL（例如，通过 IBM MQ Explorer 进行设置）的信息，请参阅[在队列管理器上配置 SSL](#)。

3. 将信任密钥库文件和密钥库文件（如果有）保存到适当的位置。建议位置是 `config_directory/coordination_qmgr/agents/agent_name` 目录。
4. 在适当的 IBM MQ Managed File Transfer 属性文件中根据需要为每个启用 SSL 的队列管理器设置 SSL 属性。每个属性集引用单个队列管理器（代理、协调和命令），但一个队列管理器可执行其中的两个或更多角色。

需要 **CipherSpec** 或 **CipherSuite** 属性之一，否则客户机会在不使用 SSL 的情况下尝试连接。由于 IBM MQ 与 Java 之间的术语差异，提供了 **CipherSpec** 或 **CipherSuite** 属性。IBM MQ Managed File Transfer 接受任一属性并执行必要的转换，因此不需要设置这两个属性。如果指定了 **CipherSpec** 和 **CipherSuite** 属性，那么 **CipherSpec** 将优先。

PeerName 属性是可选的。您可以将该属性设置为要连接到的队列管理器的专有名称。IBM MQ Managed File Transfer 将拒绝到具有不匹配专有名称的错误 SSL 服务器的连接。

将 **SslTrustStore** 和 **SslKeyStore** 属性设置为指向信任密钥库文件和密钥库文件的文件名。如果要为已在运行的代理设置这些属性，请停止并重新启动代理以采用 SSL 方式重新连接。

属性文件包含明文密码，因此请考虑设置适当的文件系统许可权。

有关 SSL 属性的更多信息，请参阅第 634 页的『SSL 属性』。

5. 如果代理队列管理器使用 SSL，您不能在创建代理时提供必需的详细信息。请使用以下步骤创建代理：
 - a) 使用 **fteCreateAgent** 命令创建代理。您将收到有关无法将代理的存在情况发布到协调队列管理器的警告。
 - b) 编辑上一步创建的 `agent.properties` 文件，以添加 SSL 信息。成功启动代理后，重新尝试发布。
6. 如果 IBM MQ Explorer 的代理程序或实例正在运行，而 `agent.properties` 文件或 `coordination.properties` 文件中的 SSL 属性已更改，那么必须重新启动代理程序或 IBM MQ Explorer。

相关任务

第 105 页的『[将 IBM MQ Advanced Message Security 与 IBM MQ Managed File Transfer 一起使用](#)』
IBM MQ Advanced Message Security 提供了 IBM MQ Managed File Transfer 中的消息流量（尤其是队列上的不活动数据）的安全性。

相关参考

第 591 页的『[agent.properties 文件](#)』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

以客户机方式使用通道认证来连接至 WebSphere MQ V7.1 或更高版本队列管理器

WebSphere MQ V7.1 引入了通道认证记录以在通道级别更准确地控制访问权。此行为更改意味着在缺省情况下，新创建的 WebSphere MQ V7.1 或更高版本队列管理器将拒绝来自 受管文件传输 组件的客户机连接。

有关通道认证的更多信息，请参阅[通道认证记录](#)。

如果 IBM MQ Managed File Transfer 使用的 SVRCONN 的通道认证配置指定了没有特权的 MCAUSER 标识，那么您必须授予队列管理器、队列和主题的特定权限记录，才能使 受管文件传输 代理和命令可正确工作。使用 MQSC 命令 SET CHLAUTH 或 PCF 命令 Set Channel Authentication Record 来创建、修改或移除通道认证记录。对于要连接至 V7.1 或更高版本队列管理器的所有 受管文件传输 代理，您都可以设置要用于所有代理的 MCAUSER 标识，或者为每个代理设置单独的 MCAUSER 标识。

向每个 MCAUSER 标识都授予以下许可权：

- 队列管理器所需的权限记录：
 - 连接
 - setid
 - inq

- 队列所需的权限记录。

对于所有特定于代理的队列（即，以下列表中名称以 *agent_name* 结尾的队列），您必须为要使用客户机连接来连接至 WebSphere MQ V7.1 或更高版本队列管理器的每个代理创建这些队列权限记录。

- put, get, dsp (SYSTEM.DEFAULT.MODEL.QUEUE)
- put, get, setid, browse (SYSTEM.FTE.COMMAND.*agent_name*)
- put, get (SYSTEM.FTE.DATA.*agent_name*)
- put, get (SYSTEM.FTE.REPLY.*agent_name*)
- put, get, inq, browse (SYSTEM.FTE.STATE.*agent_name*)
- put, get, browse (SYSTEM.FTE.EVENT.*agent_name*)
- put, get (SYSTEM.FTE)

- 主题所需的权限记录：

- sub, pub (SYSTEM.FTE)

- 文件传输所需的权限记录。

如果源代理和目标代理具有单独的 MCAUSER 标识，那么请在源和目标的代理队列上都创建权限记录。

例如，如果源代理的 MCAUSER 标识为 **user1** 并且目标代理 MCAUSER 标识为 **user2**，那么请为代理用户设置以下权限：

代理用户	队列	所需权限
user1	SYSTEM.FTE.DATA. <i>destination_agent_name</i>	put
user1	SYSTEM.FTE.COMMAND. <i>destination_agent_name</i>	put
user2	SYSTEM.FTE.REPLY. <i>source_agent_name</i>	put
user2	SYSTEM.FTE.COMMAND. <i>source_agent_name</i>	put

将 IBM MQ Advanced Message Security 与 IBM MQ Managed File Transfer 一起使用

IBM MQ Advanced Message Security 提供了 IBM MQ Managed File Transfer 中的消息流量（尤其是队列上的不活动数据）的安全性。

关于此任务

在本主题中，IBM MQ Advanced Message Security 称为 WMQAMS，IBM MQ Managed File Transfer 称为 WMQFTE。有关 WMQAMS 的更多信息，请参阅 [IBM MQ Advanced Message Security](#)。

WMQAMS 提供了多种设施来拦截消息数据并向其应用安全操作。对于 WMQFTE，WMQAMS Java 拦截器用于在数据离开源代理之前对其进行加密，并在数据到达目标代理后对其进行解密。两个代理之间的消息转移是安全的。

WMQAMS 提供了一系列可应用于 IBM MQ 网络的安全策略。WebSphere MQ File Transfer Edition 7.0.3 或更高版本支持的配置是加密两个代理之间的文件数据；但是不支持对控制或状态消息的保护。

在添加 WMQAMS 以提供额外保护之前，请首先安装并配置 MQMFT，并确认您的安装正常运行。

过程

1. 在托管要保护的 MQMFT 代理程序的每个系统上安装 WMQAMS Java 拦截器。
遵循 WMQAMS 产品文档中的指示信息来安装 Java 拦截器组件。您必须还至少在一个系统上安装 WMQAMS 管理工具，并且针对每个队列管理器运行必需 MQSC 脚本，这也在 WMQAMS 产品文档中进行了描述。
2. 创建 WMQAMS 所使用的加密密钥库和策略。

此配置要求在所涉及的每个代理的数据队列上具备消息加密策略 (SYSTEM.FTE.DATA.agent_name)。请参阅 [IBM MQ Advanced Message Security](#)，以了解有关此步骤的详细信息。

3. 通过 WMQFTE 启用 WMQAMS

请为每个要使用 WMQAMS 的代理执行以下步骤：

- a) 停止代理。
- b) 向 agent.properties 文件中添加 **advancedSecurityPath** 属性。此属性的值是安装在该系统上的 WMQAMS Java 拦截器 JAR 文件 (com.ibm.mq.esse.jar) 的完整文件名。

请参阅第 591 页的『agent.properties 文件』以了解有关此文件和属性的更多信息。

注：注意，WMQAMS 文档中的指示信息（引用从 IBM MQ 目录装入的此 JAR 文件）不适用。WMQFTE 包含其自身的 IBM MQ 库，并且不需要或使用单独 IBM MQ 安装来进行客户机连接。

- c) 如果以 IBM MQ 绑定方式运行代理程序，请将 **mqs.intercept.bindingsJava** 属性设置为 1。

IBM MQ 绑定是一种连接方式，在代理不使用网络协议而直接连接到相同系统上的队列管理器的情况下使用。如果 agent.properties 文件包含 **agentQMGr** 属性但不包含 **agentQMGrHost** 属性，那么代理程序将使用 IBM MQ 绑定方式。

WMQAMS Java 拦截器仅在 **mqs.intercept.bindings** 属性设置为 1 的绑定方式连接上起作用。要设置 **mqs.intercept.bindings** 属性，请在启动代理程序之前运行以下命令：

- ```
export FTE_JVM_PROPERTIES="-Dmqs.intercept.bindings=1" # on Unix platforms
```
- ```
set FTE_JVM_PROPERTIES="-Dmqs.intercept.bindings=1" # on Windows platforms
```

- d) 启动代理。

下一步做什么

当 IBM MQ Advanced Message Security 用于保护代理数据队列时，传输的源和目标上的代理必须以相同队列保护策略进行配置。有关更多信息，请参阅[将 IBM MQ AMS 与 IBM MQ Managed File Transfer 一起使用](#)。

保证 Web 网关的安全

可以使用一些方法来保护 Web 网关的安全。使用 Web 网关之前，必须先执行某些安全步骤。其他步骤可选，可增加 Web 网关和 IBM MQ Managed File Transfer 网络的安全性，但使用 Web 网关并非必须执行这些步骤。

相关概念

第 106 页的『Web 网关的必需安全性』

使用 Web 网关之前，必须先完成某些安全配置步骤。这些步骤包括为 Web 网关配置用户角色、设置文件空间许可权，以及使用 WebSphere Application Server V7.0 时在应用程序服务器中设置正确的安全性级别。

第 108 页的『Web 网关的可选安全性』

使用 Web 网关之前，某些安全配置步骤并非必须执行。这些可选步骤可以加强 Web 网关和 IBM MQ Managed File Transfer 网络的安全性。可选步骤会过滤 Web 网关请求并在目标代理上启用沙箱。

Web 网关的必需安全性

使用 Web 网关之前，必须先完成某些安全配置步骤。这些步骤包括为 Web 网关配置用户角色、设置文件空间许可权，以及使用 WebSphere Application Server V7.0 时在应用程序服务器中设置正确的安全性级别。

IBM MQ Managed File Transfer 具有两个阶段的授权：用户角色和文件空间许可权。要上载文件或查询传输信息，必须已为用户分配适当的用户角色。要访问文件空间，必须为用户分配适当的用户角色，同时用户对他们尝试访问的文件空间必须具有适当的许可权级别。

应用程序服务器安全性

如果在 WebSphere Application Server V7.0 中部署 Web 网关，请使用全局安全性面板来启用正确的安全性级别。选择启用管理安全和启用应用程序安全。确保未选中使用 **Java 2** 安全性来限制应用程序对本地资源的访问。

Web 网关的用户角色

在 Web 网关用户使用 Web 网关之前，必须为他们分配一个或多个角色。将 Web 网关部署到应用程序服务器时，这些角色可映射到该应用程序服务器中存在的用户和组。

IBM MQ Managed File Transfer 定义了以下角色：

- wmqfte-agent-upload
- wmqfte-filespace-user
- wmqfte-filespace-create
- wmqfte-filespace-modify
- wmqfte-filespace-permissions
- wmqfte-filespace-delete
- wmqfte-audit
- wmqfte-admin

有关这些角色的更多信息，请参阅第 108 页的『Web 网关的用户角色』。

例如，如果应用程序服务器定义了组“Employees”、“Managers”和“Administrators”，那么可按如下所示将角色分配给这些组：

Employees

wmqfte-agent-upload
wmqfte-filespace-user

Managers

wmqfte-filespace-create
wmqfte-filespace-modify
wmqfte-filespace-permissions

管理员

wmqfte-admin

在该示例中，仅“Administrators”组中的用户能够删除文件空间。

文件空间许可权

如果 Web 网关用户是文件空间的所有者或者为他们指定了访问该文件空间的显式许可权，那么他们可以访问该文件空间。创建文件空间时，可以指定 授权 或 未授权 用户名或 Java 正则表达式的列表以匹配用户名。authorized 列表中的用户可以从文件空间下载文件，还可以向文件空间上载文件。unauthorized 列表中的用户不能访问文件空间，即使他们同时也位于 authorized 列表中或与 authorized 列表中的正则表达式匹配也是如此。有关更多信息，请参阅第 329 页的『示例：创建文件空间』。

相关概念

第 106 页的『保证 Web 网关的安全』

可以使用一些方法来保护 Web 网关的安全。使用 Web 网关之前，必须先执行某些安全步骤。其他步骤可选，可增加 Web 网关和 IBM MQ Managed File Transfer 网络的安全性，但使用 Web 网关并非必须执行这些步骤。

第 108 页的『Web 网关的可选安全性』

使用 Web 网关之前，某些安全配置步骤并非必须执行。这些可选步骤可以加强 Web 网关和 IBM MQ Managed File Transfer 网络的安全性。可选步骤会过滤 Web 网关请求并在目标代理上启用沙箱。

相关参考

第 108 页的『Web 网关的用户角色』

IBM MQ Managed File Transfer 已定义多个不同的角色，用于控制用户可执行的操作。

Web 网关的用户角色

IBM MQ Managed File Transfer 已定义多个不同的角色，用于控制用户可执行的操作。

请在部署 Web 网关之前或部署期间，在应用程序服务器上配置这些角色。有关如何配置 WebSphere Application Server Community Edition（包括如何设置安全角色）的信息，请参阅第 183 页的『准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关』。有关如何在 WebSphere Application Server V7.0 上部署 Web 网关（包括如何设置安全角色）的信息，请参阅第 198 页的『在 WebSphere 应用程序服务器 V7.0 中部署 Web 网关』。

下表列出了不同的角色及与每个角色关联的访问级别：

表 7: 角色及关联的许可权	
角色	描述
wmqfte-agent-upload	用户可以将文件上载至代理
wmqfte-filespace-user	用户可以列出自己文件空间的内容 用户可以从自己的文件空间下载 用户可以从自己的文件空间删除文件
wmqfte-filespace-create	用户可以创建文件空间（如果尚不存在使用该名称的文件空间）
wmqfte-filespace-modify	用户可以修改文件空间的属性
wmqfte-filespace-permissions	用户可以修改文件空间的许可权 ⁽¹⁾
wmqfte-filespace-delete	用户可以删除文件空间
wmqfte-audit	用户可以查看审计数据库中的信息。 注： 未与该角色关联的用户仅可以查看以下传输的审计信息： • 用户启动的上载 • 用户拥有的到文件空间的传输
wmqfte-admin	用户可以执行与所有角色关联的操作，以下项除外： • 用户不能接收用户从文件空间删除的文件的内容
(1) 可以对个别文件空间设置许可权。有关更多信息，请参阅主题 第 930 页的『Web 网关管理 API 引用』和第 331 页的『示例：修改文件空间配置』。	

Web 网关的可选安全性

使用 Web 网关之前，某些安全配置步骤并非必须执行。这些可选步骤可以加强 Web 网关和 IBM MQ Managed File Transfer 网络的安全性。可选步骤会过滤 Web 网关请求并在目标代理上启用沙箱。

过滤 Web 网关请求

作为 Web 网关管理员（具有 wmqfte-admin 角色），您可以使用应用程序服务器提供的 servlet 过滤功能来过滤向 Web 网关提交的 HTTP 请求。servlet 过滤允许在将请求传递到 Web 网关之前解析 HTTP 请求，并选择性地拒绝或修改。IBM MQ Managed File Transfer 包含 servlet 过滤器的样本实施，其中演示了该功能。

例如，出于安全考虑，您可能希望拒绝使用 `x-fte-postdest` 头指定某个命令在文件传输完成后执行的请求。或者，您可能希望修改请求中的某一个值，如队列管理器名称。

有关样本 `servlet` 过滤器的更多信息，请参阅第 109 页的『使用样本 `servlet` 过滤器过滤请求』。

目标代理上的沙箱

使用 Web 网关将文件上传到目标代理时，可以将文件上传到目标代理的系统上的绝对路径中。如果不希望允许来自 Web 网关的传输对目标代理的整个文件系统具有访问权，那么必须在 Web 网关文件上传的任何目标代理上配置代理沙箱或用户沙箱。

有关用户沙箱的更多信息，请参阅第 100 页的『使用用户沙箱』。有关代理沙箱的更多信息，请参阅第 99 页的『使用代理沙箱』。

防止跨站点请求伪造 (CSRF) 攻击

CSRF 攻击使用嵌入在恶意 Web 站点和 HTML 页面中的编码，在用户不知情的情况下向 Web 服务器提交请求。恶意用户利用此技术使可以在 Web 服务器上创建、修改或删除资源。例如，恶意用户可以创建和删除文件空间，或者修改文件空间的许可权。Web 网关提供了可对所有 HTTP POST 和 HTTP DELETE 请求执行检查的选项，以确保它们包含 CSRF 验证标记。令牌必须作为 HTTP 头或 HTML 表单属性并且必须包含当前 JSESSIONID 的值。这种双提交技术确保来自恶意源的代码无法创建完整的请求消息并且将被服务器拒绝。

缺省情况下，禁用 Web 网关的 CSRF 保护。要启用，请将 Web 网关 **CSRFProtection** 初始化参数的值设置为 `true`。

有关启用 CSRF 保护的更多信息，请参阅第 198 页的『在 WebSphere 应用程序服务器 V7.0 中部署 Web 网关』或第 183 页的『准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关』。

有关 HTTP 请求中 CSRF 标记的更多信息，请参阅第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』。

启用会话安全性

如果您正在使用 WebSphere 应用程序服务器，那么在应用程序服务器中启用此功能可确保特定 JSESSIONID 只能由对其授权的相同用户使用。这可以防止可能已拦截 JSESSIONID 的恶意用户再次使用它访问用户的帐户。

有关启用会话安全性的更多信息，请参阅[会话安全性支持](#)。

相关概念

第 106 页的『保证 Web 网关的安全』

可以使用一些方法来保护 Web 网关的安全。使用 Web 网关之前，必须先执行某些安全步骤。其他步骤可选，可增加 Web 网关和 IBM MQ Managed File Transfer 网络的安全性，但使用 Web 网关并非必须执行这些步骤。

第 106 页的『Web 网关的必需安全性』

使用 Web 网关之前，必须先完成某些安全配置步骤。这些步骤包括为 Web 网关配置用户角色、设置文件空间许可权，以及使用 WebSphere Application Server V7.0 时在应用程序服务器中设置正确的安全性级别。

相关任务

第 109 页的『使用样本 `servlet` 过滤器过滤请求』

将 HTTP 请求交付到 IBM MQ Managed File Transfer Web 网关之前，可以对其进行过滤以拒绝或进行修改。

使用样本 `servlet` 过滤器过滤请求

将 HTTP 请求交付到 IBM MQ Managed File Transfer Web 网关之前，可以对其进行过滤以拒绝或进行修改。

开始之前

您需要类路径上的 Java Platform, Enterprise Edition (Java EE) 库来编译样本 `servlet` 过滤器文件。

关于此任务

IBM MQ Managed File Transfer 随附的样本 servlet 过滤器显示有关如何过滤 HTTP 请求的示例。样本过滤器文件 `SampleServletFilter.java` 位于 IBM MQ Managed File Transfer 安装的 `samples/web/filter` 目录中。还将在本主题末尾重新生成该文件。

过程

1. 编译 `SampleServletFilter.java` 文件以创建 `SampleServletFilter.class` 和 `RequestWrapper.class` 文件。
2. 将编译的类文件放置在应用程序服务器类路径上。执行此操作的过程特定于您正在使用的应用程序服务器。例如，如果您正在使用 WebSphere Application Server V 7.0，请将类文件放入 JAR 文件中，然后将 JAR 文件复制到 `WAS_install_root/lib` 目录中。
3. 从 Web 网关 EAR 文件 `com.ibm.wmqfte.web.ear` 中抽取模块 `com.ibm.wmqfte.web.war`。EAR 文件位于 IBM MQ Managed File Transfer 服务 安装的 `MQ_INSTALLATION_PATH/mqft/web` 目录中。要抽取 `com.ibm.wmqfte.web.war` 文件，请运行以下命令：

```
jar -xf com.ibm.wmqfte.web.ear com.ibm.wmqfte.web.war
```

4. 运行以下命令，以从 `com.ibm.wmqfte.web.war` 文件中抽取 `web.xml` 文件：

```
jar -xf com.ibm.wmqfte.web.war WEB-INF/web.xml
```

5. 使用文本编辑器对 `web.xml` 文件中的以下行取消注释：

```
<filter>
  <filter-name>SampleServletFilter</filter-name>
  <filter-class>SampleServletFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>SampleServletFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

注: If you are writing your own servlet filter, change the `<filter-name>` and `<filter-class>` values in the `web.xml` file to match your servlet filter. 将 `url-pattern` 的值保留为 `/*`。

6. 运行以下命令，以使用已修改的 `WEB-INF/web.xml` 文件来更新 Web 网关应用程序：

```
jar -uf com.ibm.wmqfte.web.war WEB-INF/web.xml
```

7. 运行以下命令，以使用已更新的 WAR 文件来更新 EAR 文件：

```
jar -uf com.ibm.wmqfte.web.ear com.ibm.wmqfte.web.war
```

8. 将 Web 网关应用程序部署到应用程序服务器。有关部署应用程序的指示信息，请参阅第 197 页的『部署 IBM MQ Managed File Transfer Web 网关』。

示例

```
/*
 *
 * Version: %Z% %I% %W% %E% %U% [%H% %T%]
 *
 * Licensed Materials - Property of IBM
 *
 * 5724-H72
 *
 * Copyright IBM Corp. 2010, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
```



```

*
*/

import java.io.IOException;
import java.util.Enumeration;
import java.util.logging.Level;
import java.util.logging.LogRecord;
import java.util.logging.Logger;

import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;
import javax.servlet.http.HttpServletResponse;

/**
 * A sample servlet filter implementation that demonstrates how an application
 * server administrator can filter (reject or modify) HTTP requests before they
 * are passed to the Web Gateway. The filter is called when a request
 * is received by the application server for any servlet which has this
 * class configured as a filter.
 *
 * In this example implementation two parts of an HTTP request are checked before
 * the request is passed to the servlet:
 *
 * 1 - If the x-fte-postdest header has been set, the request is rejected by
 * returning an HTTP 400 Bad Request in a response to the HTTP client.
 *
 * This demonstrates how an administrator can use servlet filters to reject
 * WMQFTE HTTP requests that they don't want to reach the WMQFTE environment.
 * In this example, the filter rejects any HTTP request that specifies a
 * command to execute after the transfer has completed.
 *
 * 2 - If the destination agent that is specified in a file upload URI matches one
 * of the three aliases defined in this filter (ACCOUNTS, MARKETING and WAREHOUSE),
 * the destination alias is replaced with the actual destination agent and queue
 * manager values for that alias.
 *
 * This demonstrates how an administrator can use servlet filters to modify
 * any part of a request before it is passed through to the WMQFTE
 * environment. In this example, the destination agent is changed in the
 * request URI if it matches one of a number of known aliases.
 */
public class SampleServletFilter implements Filter {

    /*
     * (non-Javadoc)
     * @see javax.servlet.Filter#doFilter(javax.servlet.ServletRequest,
     * javax.servlet.ServletResponse, javax.servlet.FilterChain)
     */
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain
        chain) throws IOException, ServletException {

        Logger sampLogger = Logger.getLogger("SampleServletFilter");
        sampLogger.log(new LogRecord(Level.INFO, "WebSphere MQ File Transfer Edition Web
Gateway - SampleServletFilter invoked"));

        RequestWrapper modifiedRequest = null;

        if (request instanceof HttpServletRequest && response instanceof HttpServletResponse)
        {

            HttpServletRequest httpRequest = (HttpServletRequest) request;
            HttpServletResponse httpResponse = (HttpServletResponse) response;

            /*****
             * The first part of the filter - reject any requests that attempt
             * to run commands on the destination agent system
             *****/

            /*
             * Get any 'x-fte-postdest' headers which might have been set
             */
            Enumeration<?> postDestCalls = httpRequest.getHeaders("x-fte-postdest");

            if (postDestCalls != null && postDestCalls.hasMoreElements()) {

```

```

        * Because we want to filter out all requests that attempt to run commands
        * on the destination agent system, if we find any values at all for the
        * x-fte-postdest header then we reject the request instead of proceeding.
        */

        httpResponse.setContentType("text/html");
        httpResponse.sendError(HttpServletResponse.SC_BAD_REQUEST, "Request
rejected - an attempt to run commands was detected.");
    }

    /*****
    * The second part of the filter - map our own aliases for WMQFTE
    * agents to the correct agent and queue manager pair
    *****/
    String requestURI = httpRequest.getRequestURI();

    if (requestURI.indexOf("/agent/ACCOUNTS") >= 0) {
        modifiedRequest = new RequestWrapper(httpRequest);
        modifiedRequest.changeDestinationAgent("/agent/ACCOUNTS", "/agent/
ACTS.AGENT@ACTS.QM");
    } else if (requestURI.indexOf("/agent/MARKETING") >= 0) {
        modifiedRequest = new RequestWrapper(httpRequest);
        modifiedRequest.changeDestinationAgent("/agent/MARKETING", "/agent/
MKTG.AGENT@MKTG.QM");
    } else if (requestURI.indexOf("/agent/WAREHOUSE") >= 0) {
        modifiedRequest = new RequestWrapper(httpRequest);
        modifiedRequest.changeDestinationAgent("/agent/WAREHOUSE", "/agent/
WRHS.AGENT@WRHS.QM");
    } else {
        // Leave the original request URI in place
    }

    /*****
    * Finally call the next filter in the chain with the original
    * request (or a new wrapped request if one has been created) and
    * the original response.
    *****/
    if (modifiedRequest != null) {
        chain.doFilter(modifiedRequest, response);
    } else {
        chain.doFilter(request, response);
    }
} else {
    chain.doFilter(request, response);
}
}

/*
 * (non-Javadoc)
 * @see javax.servlet.Filter#destroy()
 */
public void destroy() {
    // Do nothing
}

/*
 * (non-Javadoc)
 * @see javax.servlet.Filter#init(javax.servlet.FilterConfig)
 */
public void init(FilterConfig config) throws ServletException {
    // Do nothing
}
}

/**
 * A class to wrap an <code>HttpServletRequest</code> so we can modify parts of the request
 */
class RequestWrapper extends HttpServletRequestWrapper {

    private String originalDestination, newDestinationAgent;

    /*
     * Constructor
     */
    public RequestWrapper(HttpServletRequest request) {
        super(request);
    }

    /*
     *
     * (non-Javadoc)

```

```

    * @see javax.servlet.http.HttpServletRequestWrapper#getRequestURI()
    */
    @Override
    public String getRequestURI() {
        String originalURI = super.getRequestURI();

        StringBuffer newURI = new StringBuffer();

        newURI.append(originalURI.substring(0, originalURI.indexOf(originalDestination)));
        newURI.append(newDestinationAgent);
        newURI.append(originalURI.substring(originalURI.indexOf(originalDestination)
+ originalDestination.length(),
                                originalURI.length()));

        return newURI.toString();
    }

    /**
     * Change the original destination agent/queue manager set in the request by the
     * HTTP client (or a previous filter) to a new destination agent/queue manager.
     *
     * @param originalDestination
     * @param newDestination
     */
    protected void changeDestinationAgent(String originalDestination, String newDestination) {
        this.originalDestination = originalDestination;
        this.newDestinationAgent = newDestination;
    }
}

```

配置 Connect:Direct 网桥代理和 Connect:Direct 节点之间的 SSL 或 TLS 连接

通过创建密钥库和信任库并在 Connect:Direct 网桥代理属性文件中设置属性来配置 Connect:Direct 网桥代理和 Connect:Direct 节点，以通过 SSL 协议彼此建立连接。

关于此任务

这些步骤中包含获取由认证中心签名的密钥的指示信息。如果您不使用认证中心，那么可以生成自签名证书。有关生成自签名证书的更多信息，请参阅在 [UNIX 和 Windows 系统上使用 SSL 或 TLS](#)。

这些步骤中包含有关为 Connect:Direct 网桥代理创建新的密钥库和信任库的指示信息。如果 Connect:Direct 网桥代理已具有用于安全连接到 WebSphere MQ 队列管理器的密钥库和信任库，那么在安全连接到 Connect:Direct 节点时您可以使用现有的密钥库和信任库。有关更多信息，请参阅 [第 103 页的『为 IBM MQ Managed File Transfer 配置 SSL 或 TLS 加密』](#)。

过程

对于 Connect:Direct 节点，请完成以下步骤：

1. 针对 Connect:Direct 节点生成密钥和签名证书。
您可以通过使用与 WebSphere MQ 一起提供的 IBM Key Management 工具来完成该操作。有关更多信息，请参阅[使用 SSL 或 TLS](#)。
2. 向认证中心发送请求以获取签名密钥。您将收到证书。
3. 创建文本文件；例如 /test/ssl/certs/CAcert，以包含认证机构公用密钥。
4. 在 Connect:Direct 节点上安装 Secure+ Option。
如果该节点已存在，那么可以通过再次运行安装程序、指定现有安装的位置以及选择只安装 Secure+ Option 来安装 Secure+ Option。
5. 创建新的文本文件；例如 /test/ssl/cd/keyCertFile/node_name.txt。
6. 将从认证中心收到的证书和位于 /test/ssl/cd/privateKeys/node_name.key 中的专用密钥复制到该文本文件中。

/test/ssl/cd/keyCertFile/node_name.txt 的内容必须是以下格式：

```

-----BEGIN CERTIFICATE-----
MIIICnzCCAgiAwIBAgIBGjANBgkqhkiG9w0BAQUFADBBeMQswCQYDVQQGEwJHQSJES
MBAGA1UECBMJSGFtchNoaXJ1MRAwDgYDVQQHEwdIdXJzbGV5MQwwCgYDVQQKEwNJ

```

```

Qk0xDjAMBgNVBAstBU1RSVBUMQswCQYDVQDEwJDQTAeFw0xMTAzMDExNjIwNDZa
Fw0yMTAyMjYxNjIwNDZaMFAxCzAJBgNVBAYTAkdCMRIwEAYDVQQIEw1lYw1wc2hp
cmUxODDAKBgNVBAoTA0lCTTEOMAwGA1UECzMFTVFGVEUxODZANBgNVBAMTBmJpbmJh
ZzCBnzANBgkqhkiG9w0BAQEFAA0BjQAwgYkCgYEAvgP1QIk1u9ypSKD1Xo0Do1yk
EyMFXB0UpZr1DVxjoSEC0vtWNCJ199e+Vc4UpNybdYBu+NkD1MNoF4QxeQcLAFj
WnhakQCiQ+JIAD5AurhnwChe0MV3kjA84GKH/10SVqt1984mu/1DyS819XcfSSN
c00MsK1KbneVSCIV2XECawEAAa7MHkwCQYDVR0TBAlwADA5Bg1ghkgBhvCAQ0E
HxYdT3B1b1NTTCBHZW51cmF0ZWQgQ2VydG1maWNhdGUwHQYDVRO0BBYEFNMIpSc
csBXUniW4A3UzZnCRsv3MB8GA1UdIwQYMBaAFDXY8rmj41Vz5+FVAoQb++cns+B4
MA0GCSqGSIb3DQEBBQUAA4GBAFc7k1Xa4pGKYgwhxKpE3ZF6FNwy4vBX5216/ja
8h/v18+iv010CL8t0ZOKSU95fyZLz0PKnCH7v+ItFSE3CIIEk9D1z2U6W091ICwn
17PL72TdfaL3kabwHYVf17IVcuL+VZsZ3HjLggP2qH09ZuJPseT9+AxFVMLiaAb
8eHw
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,64A02DA15B6B6EF9

57kqxLOJ/gRU0IQ6hVK2YN13B4E1jAi1gSme0I5ZpEIG8CHXISKB7/0cke2FTqsV
lvI99QyCxsDwoMNT5fj51v7aPmVeS60b0m+U1Gxe8B/Zel8JVj204K2Uh72rDCXE
5e6eFxDuM207sQDy20euBVELJtM2k0kL1R0doQQS1U3XQNgJw/t3ZIx5hPXWEQT
rjRQ064BEhb+PzzxPF8uwzZ9IzUK9BJ/UUnqC60dBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTK180GvBvWX04fHyvIX5aslwhBoAzXIS1AtNTxptPvoaP1zyIAeZ6OCVo/
Sfo+A2UhmteJeoJaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIETgu4A1+CV64RJED
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWdyJBvHU0jXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfdwP+bEjDzUaaarJTS7lIFeLLw7eJ8MNAKMGicDkycL0
EPBU9X5QnHKLK0fYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkt5e5+1YcX80VZ6
sHFPN1HluCny/r1UcBy9iviVeodX8Iom0chSy05DK18bwZNjYtUP+CtYHNFU5BaD
I+1uU0AeJ+wjQYKT1WaeIGZ3VxuNITJu18y5qDTXXfx7vxM50oWxa6U5+AYuGUMg
/itPZmUmNzHjTk7ghT6i1IQ0aBowXXKJB1Mmq/6BQXN2IhkD9ys2qrvM1hdi5nAf
egmdiG50loLnBRqWbfr+DykpAhK4SaDi2F52Uxovw3Lhiw8dQP71zQ==
-----END RSA PRIVATE KEY-----

```

7. 启动 Secure+ Admin Tool。

- 在 Linux 或 UNIX 系统上，运行命令 **spadmin.sh**。
- 在 Windows 系统上，单击开始 > 程序 > **Sterling Commerce Connect:Direct > CD Secure+ Admin Tool**

CD Secure+ Admin Tool 将启动。

8. 在 CD Secure+ Admin Tool 中，双击 **.Local** 行以编辑主 SSL 或 TLS 设置。

- 根据使用的协议，选择**启用 SSL 协议**或**启用 TLS 协议**。
- 选择**禁用覆盖**。
- 至少选择一个密码套件。
- 如果需要双向认证，请将 **启用客户机认证** 的值更改为 Yes。
- 在**可信根证书**字段中，输入认证中心的公用证书文件的路径 **/test/ssl/certs/CAcert**。
- 在**密钥证书文件**字段中，输入您创建的文件的路径，**/test/ssl/cd/keyCertFile/node_name.txt**。

9. 双击 **.Client** 行以编辑主 SSL 或 TLS 设置。

- 根据使用的协议，选择**启用 SSL 协议**或**启用 TLS 协议**。
- 选择**禁用覆盖**。

对于 Connect:Direct 网桥代理，请执行以下步骤：

10. 创建信任库。您可以通过创建虚密钥然后删除该虚密钥的方法来创建。

可以使用以下命令：

```
keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

```
keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

11. 将认证中心的公用证书导入该信任库。

可以使用以下命令：

```
keytool -import -trustcacerts -alias myCA
        -file /test/ssl/certs/CAcert
        -keystore /test/ssl/fte/stores/truststore.jks
```

12. 编辑 Connect:Direct 网桥代理属性文件。

在该文件的任何位置中包含以下行：

```
cdNodeProtocol=protocol
cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
cdNodeTruststorePassword=password
```

在此步骤的示例中，*protocol* 是您所使用的协议（即 SSL 或 TLS），*password* 是创建信任密钥库时指定的密码。

13. 如果希望双向认证，请为 Connect:Direct 网桥代理创建密钥和证书。

a) 创建密钥库和密钥。

可以使用以下命令：

```
keytool -genkey -keyalg RSA -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -validity 365
```

b) 生成签名请求。

可以使用以下命令：

```
keytool -certreq -v -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks -storepass password
        -file /test/ssl/fte/requests/agent_name.request
```

c) 将上述步骤中收到的证书导入密钥库。该证书必须为 x.509 格式。

可以使用以下命令：

```
keytool -import -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -file certificate_file_path
```

d) 编辑 Connect:Direct 网桥代理属性文件。

在该文件的任何位置中包含以下行：

```
cdNodeKeystore=/test/ssl/fte/stores/keystore.jks
cdNodeKeystorePassword=password
```

在此步骤的示例中，*password* 是创建密钥库时指定的密码。

相关任务

第 204 页的『配置 Connect:Direct 网桥』

配置 Connect:Direct 网桥，以在 IBM MQ Managed File Transfer 网络和 Connect:Direct 网络之间传输文件。Connect:Direct 网桥的组件包括 Connect:Direct 节点和专用于与该节点通信的 IBM MQ Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

配置 IBM MQ Managed File Transfer

安装完成以后，您可以配置 IBM MQ Managed File Transfer 的功能部件。

相关概念

第 141 页的『为首次使用配置 IBM MQ Managed File Transfer』

在首次使用 IBM MQ Managed File Transfer 代理和队列管理器之前，必须先对其执行一些配置任务。

第 116 页的『分布式平台上的配置选项』

IBM MQ Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。这些属性文件位于安装产品时所定义的配置目录中。

相关任务

第 181 页的『配置 Web 网关』

您必须配置 IBM MQ Managed File Transfer Web 网关以使用现有的 IBM MQ Managed File Transfer 环境。配置过程特定于您正在使用的应用程序服务器。在配置 Web 网关之前，请在与应用程序服务器相同的系统上创建 Web 代理。

第 204 页的『配置 Connect:Direct 网桥』

配置 Connect:Direct 网桥，以在 IBM MQ Managed File Transfer 网络和 Connect:Direct 网络之间传输文件。Connect:Direct 网桥的组件包括 Connect:Direct 节点和专用于与该节点通信的 IBM MQ Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

相关参考

第 447 页的『IBM MQ Managed File Transfer 命令汇总』

下表列出了所有 IBM MQ Managed File Transfer 命令以及指向其详细描述链接。

分布式平台上的配置选项

IBM MQ Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。这些属性文件位于安装产品时所定义的配置目录中。

您可以具有多组配置选项，每组配置选项包含一组目录和属性文件。这些属性文件中定义的值由所有 IBM MQ Managed File Transfer 命令用作缺省参数，除非在命令行中显式指定其他值。

要更改一组希望使用的缺省配置选项，可以使用 **fteChangeDefaultConfigurationOptions** 命令。要更改一组希望用于个别命令的配置选项，您可以将参数 **-p** 与任何 IBM MQ Managed File Transfer 命令一起使用。

配置选项集的名称是协调队列管理器的名称，建议不要进行更改。然而，可以更改配置选项集的名称，但必须更改 **config** 和 **logs** 目录的名称。在以下示例中，配置选项集的名称表示为 *coordination_qmgr_name*。

配置选项目录结构

在配置产品时，会在配置目录下生成如下结构的目录和属性文件。您也可以通过以下命令更改这些目录和属性文件：**fteSetupCoordination**、**fteSetupCommands**、**fteChangeDefaultConfiguration** 和 **fteCreateAgent**。

```
MQ_DATA_PATH/mqft/
  config/
    coordination_qmgr_name/
      coordination.properties
      command.properties
      agents/
        agent_name/
          agent.properties
          exits
      loggers/
        logger_name
        logger.properties
    installations/
      installation_name/
        installation.properties
```

coordination_qmgr_name 目录是配置选项目录。配置目录中可以有多组配置选项目录。*agent_name* 目录是代理目录。除了包含 **agent.properties** 文件外，该目录还包含 **exits** 目录，这是用户出口例程以及 **fteCreateBridgeAgent** 和 **fteCreateCDAgent** 命令生成的各种 XML 文件的缺省位置。一组配置选项的 **agents** 目录中可以有多组代理目录。

属性文件

installation.properties

installation.properties 文件指定一组缺省配置选项的名称。该条目将 IBM MQ Managed File Transfer 指向包含要使用的配置的一组结构化目录和属性文件。通常，配置选项集的名称是关联的协调

队列管理器的名称。有关 `installation.properties` 文件的更多信息，请参阅第 582 页的『`installation.properties` 文件』。

coordination.properties

`coordination.properties` 文件指定到协调队列管理器的连接详细信息。由于多个 IBM MQ Managed File Transfer 安装可能共享同一个协调队列管理器，因此您可以使用指向共享驱动器上的公共 `coordination.properties` 文件的符号链接。有关 `coordination.properties` 文件的更多信息，请参阅第 584 页的『`coordination.properties` 文件』。

command.properties

`command.properties` 文件指定发出命令时要连接的命令队列管理器以及 IBM MQ Managed File Transfer 联系该队列管理器所需的信息。有关 `command.properties` 文件的更多信息，请参阅第 588 页的『`command.properties` 文件』。

agent.properties

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。有关 `agent.properties` 文件的更多信息，请参阅第 591 页的『`agent.properties` 文件』。

logger.properties

`logger.properties` 文件指定记录器的配置属性。有关 `logger.properties` 文件的更多信息，请参阅第 163 页的『IBM MQ Managed File Transfer 的记录器配置属性』。

属性文件和代码页

由于 Java 的限制，所有 IBM MQ Managed File Transfer 属性文件的内容都必须保留美国英语。如果您在一个非美式英语系统中编辑属性文件，那么必须使用 Unicode 换码序列。

相关参考

第 582 页的『`installation.properties` 文件』

`installation.properties` 文件指定一组缺省配置选项的名称。该条目将 IBM MQ Managed File Transfer 指向包含要使用的配置的一组结构化目录和属性文件。通常，配置选项集的名称是关联的协调队列管理器的名称。

第 584 页的『`coordination.properties` 文件』

`coordination.properties` 文件指定到协调队列管理器的连接详细信息。由于多个 IBM MQ Managed File Transfer 安装可能共享同一个协调队列管理器，因此您可以使用指向共享驱动器上的公共 `coordination.properties` 文件的符号链接。

第 588 页的『`command.properties` 文件』

`command.properties` 文件指定发出命令时要连接的命令队列管理器以及 IBM MQ Managed File Transfer 联系该队列管理器所需的信息。

第 591 页的『`agent.properties` 文件』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

第 163 页的『IBM MQ Managed File Transfer 的记录器配置属性』

该记录器具有一组配置属性。在 `logger.properties` 文件中指定这些属性，该文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中。

第 634 页的『SSL 属性』

将 SSL 或 TLS 与 IBM MQ 和 IBM MQ Managed File Transfer 一起使用可防止在代理与队列管理器之间建立未经授权的连接，并可对代理与队列管理器之间的消息通信进行加密。

第 633 页的『Java 系统属性』

必须将许多 IBM MQ Managed File Transfer 命令和代理程序属性定义为 Java 系统属性，因为它们定义了无法使用命令或代理程序属性机制的早期功能的配置。

第 454 页的『`fteChangeDefaultConfigurationOptions`（更改缺省配置选项）』

使用 `fteChangeDefaultConfigurationOptions` 命令来更改您希望 IBM MQ Managed File Transfer 使用的缺省配置选项。配置选项的值定义 IBM MQ Managed File Transfer 使用的属性文件组。

第 560 页的『`fteSetupCommands`（创建 `command.properties` 文件）』

`fteSetupCommands` 命令可创建 `command.properties` 文件。在您发出命令时，该属性文件指定与 IBM MQ 网络连接的队列管理器的详细信息。

第 562 页的『fteSetupCoordination（设置协调详细信息）』

fteSetupCoordination 命令为 IBM MQ Managed File Transfer 创建属性文件和协调队列管理器目录。

第 460 页的『fteCreateAgent（创建 IBM MQ Managed File Transfer 代理）』

fteCreateAgent 命令创建代理及其关联的配置。

z/OS 上的配置选项

z/OS 上的 IBM MQ Managed File Transfer 配置选项与针对分布式平台的选项相同。

有关分布式平台上配置选项的更多信息，请参阅第 116 页的『分布式平台上的配置选项』。

在 z/OS 上，由环境变量 BFG_DATA 定义配置位置。如果 BFG_DATA 引用的 UNIX 系统服务目录下尚不存在配置，那么 MQMFT 命令 PDSE 库数据集的 BFGCUSTM JCL 脚本将生成创建配置时所需的作业。然后会在您运行这些生成的作业时创建配置。配置创建依赖于引用现有可访问目录的 BFG_DATA。

您还可以使用分布式平台和 z/OS 上提供的相同 **fte** 命令来创建和维护配置。要获取 **fte** 命令的列表，请参阅第 447 页的『IBM MQ Managed File Transfer 命令汇总』。

相关概念

第 116 页的『分布式平台上的配置选项』

IBM MQ Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。这些属性文件位于安装产品时所定义的配置目录中。

第 125 页的『创建 MFT 凭证文件』

您可以使用 MFT 凭证文件来存储用户标识和密码信息，用于连接到 IBM MQ 和 Db2，并具有每个代理程序的凭证文件。

第 128 页的『创建代理』

您需要复制 PDSE 以生成特定于代理的 PDSE，例如 *user.MFT.AGENT1*。从上一个代理或记录器配置（如果存在）中复制 PDSE。如果这是您的第一个配置，请复制 MFT 随附的 PDSE。

第 127 页的『定义协调队列管理器』

IBM MQ Managed File Transfer 要求创建的队列管理器可充当协调队列管理器。

相关任务

第 129 页的『更新现有的 IBM MQ Managed File Transfer 代理或记录器命令数据集』

您可以更新通过 MQMFT 命令模板数据集创建的 MQMFT 命令 PDSE 库数据集。

创建 IBM MQ Managed File Transfer 代理或记录器命令数据集

您可以通过 MQMFT 命令模板数据集，为特定协调的特定代理或记录器创建 PDSE 命令数据集。

关于此任务

完成以下步骤：

过程

1. 制作 MQMFT 命令模板 PDSE 库数据集 SBFGCMDS 的副本。

必须将 SBFGCMDS 复制到新库中，例如 <prefix>.<agent>.JCL_。您可以使用 SBFGCMDS(BFGCOPY) 成员的已更新版本，并进行以下替换操作：

- 将 *++supplied-library++* 替换为 SBFGCMDS PDSE 的标准名称。
- 将 *++service-library++* 替换为新 MQMFT 命令 PDSE 库数据集的标准名称。 *++service-library++* 是创建的代理或记录器服务的输出数据集。

2. 对于新的 MQMFT 命令 PDSE 库数据集，请编辑成员 BFGCUSTM，该成员是用于为代理或记录器定制命令的 JCL 脚本。以 *++variable name++* 格式指定每个变量，必须将变量替换为其必需值。要获取各种 JCL 变量的描述，请参阅第 130 页的『z/OS JCL 变量』。BFGSTDIN DD 语句可以定义三种类型的变量：Variables、Properties 和 Environment。该语句格式如下：

```
[Variables]
variable1=value1
```

```
variable2=value2
....
variableN=valueN
[Properties]
property1=property value1
property2=property value2
...
propertyN=property valueN
[Environment]
custom_variable1=value1
custom_variable2=value2
....
custom_variableN=valueN
```

Variables 定义每个命令需要的设置和环境变量集。

Properties 为 MQMFT 配置属性定义覆盖项。您可以根据需要添加代理和记录器属性，为您的环境定制代理或记录器。要获取所有属性的列表，请参阅第 139 页的『配置属性文件』。提供功能之后，将不需要再访问 MQMFT 配置属性文件，该文件需要作为 UNIX 系统服务文件进行维护。

Environment 定义了所需的其他所有定制环境变量。

3. 为新的 MQMFT 命令 PDSE 库数据集提交作业 BFGCUSTM。此作业将生成 JCL 命令集，作为 PDSE 的新成员，适合于代理或记录器。要获取完整的命令列表，请参阅第 133 页的『z/OS 代理和记录器命令 JCL 脚本』。

作业 BFGCUSTM 将更新包含 JCL（包含带 DISP=OLD 的 DD 语句）的库。您必须在提交后退出编辑器，才能允许执行作业。

检查输出作业日志，以查看 JCL 脚本是否已成功运行。如果有任何故障，请纠正并重新提交 BFGCUSTM 作业。

BFGCUSTM JCL 脚本还将根据需要来更新 UNIX 系统服务 MQMFT 配置属性文件，以使文件保持一致。如果 CoordinationQMgr 属性定义的配置不存在，那么将输出警告消息，并且您必须运行生成的 BFGCFGR 和 BFGCMCR 作业才能创建配置属性文件。必须针对代理运行 BFGAGCR，针对记录器编辑运行 BFGLGCRS。如果指定的配置已存在，那么将使用 BFTCUSTM JCL 脚本中定义的任何属性更新此配置。

相关概念

第 118 页的『z/OS 上的配置选项』

z/OS 上的 IBM MQ Managed File Transfer 配置选项与针对分布式平台的选项相同。

相关任务

第 129 页的『更新现有的 IBM MQ Managed File Transfer 代理或记录器命令数据集』

您可以更新通过 MQMFT 命令模板数据集创建的 MQMFT 命令 PDSE 库数据集。

针对 IBM MQ Managed File Transfer for z/OS 的配置任务

IBM MQ Managed File Transfer for z/OS 需要定制，使组件能够正确运行。

关于此任务

您需要：

1. 编辑 PDSE 成员以指定配置数据
2. 定义协调队列管理器
3. 定义命令队列管理器
4. 配置一个或多个代理
5. 可选：配置记录器任务以将数据存储在 Db2 中

在以下主题中详细描述了所需执行任务的顺序。

相关概念

第 120 页的『查看 IBM MQ Managed File Transfer 配置』

在开始之前，您需要查看系统的配置。

相关信息

[在 z/OS 上安装 IBM MQ Managed File Transfer](#)

查看 IBM MQ Managed File Transfer 配置

在开始之前，您需要查看系统的配置。

对于已定义的每种 MFT 配置，IBM MQ Managed File Transfer (MFT) 需要一个或多个队列管理器充当以下角色：

- 协调队列管理器，用于维护配置中各个代理状态的相关信息，这些信息发布在协调程序的相关主题中。
- 一个或多个命令或连接队列管理器，用作 MFT 命令的 IBM MQ 网络入口点。
- 一个或多个代理队列管理器，用于在 MFT 代理与 IBM MQ 网络之间提供通信。

以上每个角色都可以由单独的队列管理器执行；也可以组合角色，这样，在最简单的配置中，所有角色由同一个队列管理器执行。

如果要在现有 MFT 环境中添加 z/OS 队列管理器，您需要定义 z/OS 队列管理器与此配置中的其他队列管理器之间的连接。可以使用手动定义的传输队列或使用集群来实现此目的。

每个 MFT 代理都会与一个队列管理器通信。如果有多个代理与同一个队列管理器通信，那么代理队列管理器将为每个代理定义多个队列：

- SYSTEM.FTE.COMMAND.<agent name>
- SYSTEM.FTE.DATA.<agent name>
- SYSTEM.FTE.REPLY.<agent name>
- SYSTEM.FTE.STATE.<agent name>
- SYSTEM.FTE.EVENT.<agent name>
- SYSTEM.FTE.AUTHAGT1.<agent name>
- SYSTEM.FTE.AUTHTRN1.<agent name>
- SYSTEM.FTE.AUTHOPS1.<agent name>
- SYSTEM.FTE.AUTHSCH1.<agent name>
- SYSTEM.FTE.AUTHMON1.<agent name>
- SYSTEM.FTE.AUTHADM1.<agent name>

请注意，您可以定义要在其中使用诸如 SYSTEM.FTE.COMMAND.* 之类的概要文件的常规安全概要文件，也可以为每个代理定义特定的概要文件。

相关概念

[第 120 页的『在启动前』](#)

IBM MQ Managed File Transfer (MFT) 配置使用 UNIX 系统服务 (USS) 中的文件和 PDSE 数据集。

在启动前

IBM MQ Managed File Transfer (MFT) 配置使用 UNIX 系统服务 (USS) 中的文件和 PDSE 数据集。

通过 PDSE 使用 JCL 来执行大多数配置和操作，并且您需要熟悉 USS 环境中的工作。

您可以通过 ISPF 访问 OMVS，也可以在工作站上通过命令来使用 Telnet 类型会话（例如 Telnet Putty 或 SSH）。

如果通过 ISPF 使用 OMVS，那么可以使用标准 ISPF 编辑器以及浏览命令 **oedit** 和 **obrowse**。

您需要熟悉以下 USS 命令

表 8: 常用的 UNIX 系统服务命令	
命令	函数
ls -ltr path	列示有关路径中文件的信息。

表 8: 常用的 UNIX 系统服务命令 (继续)	
命令	函数
ls -ltrd directory	列示指定目录（而不是目录中的文件）的信息。
find path -name xxx	在路径目录中搜索名为 xxx 的文件。xxx 区分大小写，可以类似于 *zzz
chmod xxx path	更改文件访问许可权
df -k path	报告文件系统中剩余的可用空间。-k 报告可用空间 (KB)。
du -kt path	报告路径下目录的大小。以 KB 为单位报告大小。
oedit filename	编辑 OMVS 中的文件。
obrowse filename	浏览文件名。

查看下表中的各项，并用您企业的相应条目来填写该表。在编辑成员 `BFGCUSTM` 时需要这些值。

表 9: 成员 <code>BFGCUSTM</code> 所需的参数		
名称	示例数据	注释
BFG_JAVA_HOME	/java/java71_bit64_GA/J7.1_64/	
BFG_GROUP_NAME	MQM	
LIBRARY	SCEN.FTE.JCL	MFT PDSE 的名称。您需要每个代理或记录器任务的副本。
TMPDIR	/tmp	读和写临时文件的可访问 USS 路径。
SERVICE_TYPE	AGENT 或 LOGGER	
名称	AGENT1	
BFG_PROD	/var/ibm/wmqmft	
BFG_DATA		根据需要填写
BFG_JVM_PROPERTIES		根据需要填写
QMGR	MQPV	
MQ_PATH	/mqm/V8R0M0	
MQ_LANG	E	
Db2_HLQ	SYS2.Db2.V10	
FTE_CONFIG		在迁移中使用
CREDENTIAL_PATH		在迁移中使用
DB_PROPS_PATH		在迁移中使用
BFG_WTO	YES	获取系统日志中的 MFT 消息。
ADMIN_JOB1		作业卡。使用相同的 JCL 卡生成所有作业。
OUTPUT_CLASS	*	

表 9: 成员 BFGCUSTM 所需的参数 (继续)		
名称	示例数据	注释
JOB CARD1		这是长时间运行的任务、代理和记录器的作业卡。
PATH	bin:/usr/bin:/usr/sbin	
armELEMTYPE	如果正在使用 ARM, 那么使用 ARM 策略中指定的 ARM ELEMTYPE。例如, armELEMTYPE=SYSBFGAG (针对代理) 或 armELEMTYPE=SYSBFGLG (针对记录器)。如果未在使用 ARM, 那么将此参数设置为空白; 例如 armELEMTYPE=	
armELEMENT	如果正在使用 ARM, 那么对该代理或记录器使用 ARM 策略中指定的 ARM ELEMENT 值。如果未在使用 ARM, 那么将此参数设置为空白; 例如, armELEMENT=	
coordinationQMgr	MQPV	必需配置

此外, 还必须查看以下变量并根据需要提供相应的值:

- coordinationQMgrHost=
- coordinationQMgrPort=
- coordinationQMgrChannel=
- connectionQMgr=
- connectionQMgrHost=
- connectionQMgrPort=
- connectionQMgrChannel=

这些属性是 AGENT 或 LOGGER 的常用属性。

注: 对于客户机连接, Host、Port 和 Channel 是必需的, 但对于本地机器上的绑定连接, 这三项应留空。

相关概念

第 122 页的『要检查的项』

确保您具有足够的磁盘空间, 以及用于存储数据的目录, 并且存在必需文件。

第 126 页的『编辑成员 BFGCUSTM 并完成参数』

在运行作业之前, 必须编辑成员 BFGCUSTM, 然后输入您的企业所使用参数的值。

要检查的项

确保您具有足够的磁盘空间, 以及用于存储数据的目录, 并且存在必需文件。

检查您是否具有足够的磁盘空间

检查在将用于存储配置特定文件的文件系统上是否具有足够的可用磁盘空间。

如果已启用代理跟踪, 那么缺省情况下会使用 100 MB 的磁盘空间。

配置文件本身很小, 只有数 KB。

如果您计划使用两个代理和一个记录器, 那么至少需要 300 MB。您可以使用命令 **df -k path**, 其中 path 是安装特定文件的位置。这会提供可用空间和总空间 (KB)。

300 MB 为 307200 KB，因此，至少应预留 310000 KB。

创建并检查用于存储 IBM MQ Managed File Transfer 数据的目录

您需要一个用于存储 IBM MQ Managed File Transfer (MFT) 数据的目录。

检查您在文件系统 `df -k /var` 中是否具有足够的空间。该文件系统至少应具有 310000 KB 可用空间。

如果尚未创建此文件系统，请使用 `mkdir` 命令；例如 `mkdir /var/mft`。

使用命令 `ls -ltrd /var/mft` 来显示用户对此目录具有的许可权。

如果所有者或组不正确，请使用命令 `chown owner:group /var/mft`。

如果组的许可权不正确，请使用以下命令为所有者和组提供读、写和执行许可权。请注意，以下命令还将为所有用户提供读和执行许可权：`chmod 775 /var/mft`。

检查文件是否存在以及您是否具有这些文件的访问权

对您将在定制期间使用的的文件使用 `ls -ltr` 命令。例如：

```
ls -ltrd /java/java71_bit64_GA/J7.1_64/bin

drwxr-xr-x  4 SYSTASK TSouser   8192 Nov 15 2013 /java/java71_bit64_GA/J7.1_64/bin
```

其中，`drwxr-xr-x` 含义如下：

d 这是目录。

rwX 所有者 `SYSTASK` 具有目录的读、写和执行访问权。

r-X 组 `TSouser` 中的人员可以读和执行目录中的文件。

r-X 通用访问，即，任何人都可以读或执行目录中的文件。

检查下表中指定的文件：

表 10: 用户需要的特定文件访问权	
路径	用户执行配置时需要的访问权
BFG_JAVA_HOME	读和执行
/tmp	读和写
BFG_PROD	读
BFG_DATA	写入
MQ_PATH	读

相关概念

[第 120 页的『在启动前』](#)
IBM MQ Managed File Transfer (MFT) 配置使用 UNIX 系统服务 (USS) 中的文件和 PDSE 数据集。

[第 124 页的『常见配置』](#)
[不同的 IBM MQ Managed File Transfer 配置概述](#)

常见配置

不同的 IBM MQ Managed File Transfer 配置概述

IBM MQ Managed File Transfer 使用附加到队列管理器的代理来传输数据。

MFT 可以使用多个队列管理器：

- 一个或多个用于传输数据的队列管理器。
- 一个用于发出请求的命令队列管理器。例如，启动传输的请求将发送到此队列管理器，并且关联的命令将路由到 MFT 代理。
- 一个用于管理工作的协调队列管理器。

有三个公共 IBM MQ Managed File Transfer (MFT) 配置：

1. 具有一个或多个使用本地连接的代理的单个队列管理器。这可能用于将数据集的内容放入 IBM MQ 队列中。
2. 在使用客户机绑定的分布式机器上具有 MFT 客户机的单个队列管理器。
3. 通过通道连接的两个队列管理器，每台机器上一个或多个代理。这些代理可以是客户机或本地绑定。

请注意以下几点：

1. MFT 以 Java 编写，并使用一些 shell 脚本和 JCL 来配置和操作 MFT。
2. Db2 状态和活动可以记录并存储到 Db2 表中。
3. 配置 MFT 的人员必须熟悉 UNIX 系统服务 (USS)。例如：
 - 其文件具有 /u/userID/myfile.txt2 类似名称的目录结构
 - USS 命令，如：

cd (更改目录)

ls (列示)

chmod (更改文件许可权)

chown (更改可以访问文件或目录的文件所有权或组)

4. 在 USS 中需要以下产品才能够配置和运行 MFT：

- Java；例如 /java/java71_bit64_GA/J7.1_64/
- IBM MQ V800，例如 /mqm/V8R0M03。
- Db2 JDBC 库（如果您要将 Db2 用于状态和历史记录）；例如，/db2/db2v10/jdbc/lib

您需要协调队列管理器。但是，可以使用相同的队列管理器来运行代理、处理命令以及进行协调。如果正在使用多个队列管理器，那么必须挑选一个充当协调程序。

检查您的 IBM MQ 连接

如果您目前具有 MFT 协调程序队列管理器，那么在执行配置的队列管理器与协调和命令队列管理器之间需要连接。

MFT 凭证文件

MFT 凭证文件用于存储用户标识和密码信息。您可以为协调队列管理器、命令队列管理器、每个代理以及每个记录器分别提供一个 MFT 凭证文件。

凭证文件是可选项，但这更便于在定制环境前定义所需的文件，并且如果具备凭证文件，收到的警告消息就会更少。

警告消息将告知您 IBM MQ Managed File Transfer 认为队列管理器安全性已关闭，因此您未提供认证详细信息。

如果已启用队列管理器安全性，那么在没有 MQMFTCredentials.xml 文件的情况下可执行的操作仅限于允许您在命令行上指定 IBM MQ 用户标识的操作。

凭证文件可以位于 USS 中，但您可以通过使用数据集中的成员使其更安全。然后，可以使用安全管理器来保护数据集。

创建格式为 VB 且逻辑记录长度 (Lrecl) 为 200 的 PDSE。

在数据集内创建成员，记下该数据集和成员，然后向此成员添加以下代码：

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MQCredentials MQMFTCredentials.xsd">
</tns:mqmftCredentials>
```

<installdirectory>/mqft/samples/credentials/ 目录中提供了一个样本文件 MQMFTCredentials.xml。

请参阅 [MFT 凭证文件格式](#)，以获取更多信息。

如果编辑数据集的成员，那么可以使用 **copy** 编辑命令从 USS 文件进行复制。

您可以随时更新此成员。请注意，需要重新启动使用此进程的任何作业或服务，以便应用任何更改。

请参阅第 125 页的『创建 MFT 凭证文件』，以获取有关如何创建凭证文件的详细信息。

除了 MFT 文件，还有两个用于类似用途的凭证文件：

- 协议网桥凭证文件。请参阅[协议网桥凭证文件格式](#)，以获取更多信息。
- Connect:Direct 凭证文件。请参阅[Connect:Direct 凭证文件格式](#)，以获取更多信息。

相关概念

第 124 页的『常见配置』

不同的 IBM MQ Managed File Transfer 配置概述

第 126 页的『编辑成员 BFGCUSTM 并完成参数』

在运行作业之前，必须编辑成员 BFGCUSTM，然后输入您的企业所使用参数的值。

创建 MFT 凭证文件

您可以使用 MFT 凭证文件来存储用户标识和密码信息，用于连接到 IBM MQ 和 Db2，并具有每个代理程序的凭证文件。

如果为每个代理提供凭证文件，那么可以通过代理限制哪些用户可以访问凭证文件。

所需代码的示例：

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MFTCredentials.xsd">
<!--      name="MQPH" user="ADMIN" mqUserId="JOHNDOEH" -->

<tns:qmgr name="MQPH" user="ADMIN" mqUserId="JOHNDOEH" mqPassword="cXXXX" />
<!--      name="MQPI" user="ADMIN" mqUserId="JOHNDOE1" -->

<tns:qmgr name="MQPI" user="ADMIN" mqUserId="JOHNDOEI" mqPassword="yXXXX" />
<tns:qmgr name="MQPH"      mqUserId="NONEH" mqPassword="yXXXX" />
<tns:qmgr name="MQPI"      mqUserId="NONEI" mqPassword="yXXXX" />
</tns:mqmftCredentials>
```

当用户标识为 ADMIN 的作业需要连接到队列管理器 MQPH 时，将传递用户标识 JOHNDOEH 和用户密码 cXXXX。

如果通过任何其他用户标识来运行作业，并且连接 MQPH，那么该作业将传递用户标识 NONEH 和密码 yXXXX。

您可以使用安全产品（例如 RACF）来保护此文件，但运行 IBM MQ Managed File Transfer 命令的用户标识需要此文件的读访问权。

您可以在成员 BFGCROBS 中使用 JCL 来隐藏此文件中的信息。这会获取文件并对 IBM MQ 用户标识和密码进行加密。例如，成员 BFGCROBS 将获取以下行：

```
<tns:qmgr name="MQPI" user="JOHND0E2" mqUserId="JOHND0E1" mqPassword="yXXXX" />
```

同时创建

```
<tns:qmgr mqPasswordCipher="e977c61e9b9c363c" mqUserIdCipher="c394c5887867157c"
name="MQPI" user="JOHND0E2"/>
```

如果要保持该用户标识到 IBM MQ 用户标识映射，那么可以向此文件添加注释。例如

```
<!-- name="MQPI" user="ADMIN" mqUserId="JOHND0E1" -->
```

通过隐藏进程使这些注释保持不变。

请注意，内容会被隐藏，但未进行强加密。您应该限制哪些用户标识可以访问此文件。

相关概念

第 124 页的『MFT 凭证文件』

MFT 凭证文件用于存储用户标识和密码信息。您可以为协调队列管理器、命令队列管理器、每个代理以及每个记录器分别提供一个 MFT 凭证文件。

复制 SBFGCMDS 以创建 JCL 库

您需要为每个代理和记录器创建 JCL 库，该 JCL 包含用于创建和运行代理或记录器的配置和作业。

对于每个代理和记录器，通过编辑和运行 BFGCOPY 成员来创建 IBM 所提供 SBFGCMDS 库的副本。

该库用于定义代理或记录器的配置，并且在定制后，包含可用于创建所需 IBM MQ Managed File Transfer 配置以及代理或记录器的作业。

在此过程中创建成员 BFGCUSTM。

注: 如果您熟悉 USS 命令，那么可以使用在其他平台上使用的相同命令来配置 z/OS。

相关概念

第 124 页的『常见配置』

不同的 IBM MQ Managed File Transfer 配置概述

第 126 页的『编辑成员 BFGCUSTM 并完成参数』

在运行作业之前，必须编辑成员 BFGCUSTM，然后输入您的企业所使用参数的值。

编辑成员 BFGCUSTM 并完成参数

在运行作业之前，必须编辑成员 BFGCUSTM，然后输入您的企业所使用参数的值。

请参阅成员 BFGCUSTM 所需的参数，以获取需要特定值的参数的列表。

此外，还必须查看以下变量并根据需要提供相应的值：

- coordinationQMGrHost=
- coordinationQMGrPort=
- coordinationQMGrChannel=
- connectionQMGr=
- connectionQMGrHost=
- connectionQMGrPort=
- connectionQMGrChannel=

这些属性是 AGENT 或 LOGGER 的常用属性。

注: 对于客户机连接，Host、Port 和 Channel 是必需的，但对于本地机器上的绑定连接，这三项应留空。

如果这是 IBM MQ Managed File Transfer 环境中的第一个队列管理器，并且您希望对协调、命令和正在运行的代理使用相同的队列管理器，请将值设置为本地队列管理器名称。

```
coordinationQMGr=MQPV
connectionQMGr=MQPV
```

其中，MQPV 是您的本地队列管理器名称。

提交将更新 PDSE 的作业，并在指定路径下创建目录结构。

请注意，该作业要求独占使用，因此在运行该作业时，您必须停止使用 PSDE。

提示: 无论何时提交作业 BFGCUSTM，该作业都将替换所有 JCL 文件。您应该重命名更改的每个成员。

相关概念

第 120 页的『在启动前』

IBM MQ Managed File Transfer (MFT) 配置使用 UNIX 系统服务 (USS) 中的文件和 PDSE 数据集。

第 128 页的『创建代理』

您需要复制 PDSE 以生成特定于代理的 PDSE，例如 *user.MFT.AGENT1*。从上一个代理或记录器配置（如果存在）中复制 PDSE。如果这是您的第一个配置，请复制 MFT 随附的 PDSE。

定义协调队列管理器

IBM MQ Managed File Transfer 要求创建的队列管理器可充当协调队列管理器。

根据您的配置，该队列管理器将位于本地 MVS 系统或另一台机器上。在前一种情况下，指向它的连接是绑定连接，在后一种情况下，则为客户机连接。

成功运行配置步骤之后，PDSE 中将存在已配置的成员。

成员 BFGCFCR 定义协调队列管理器，并且此作业：

1. 在 IBM MQ Managed File Transfer (MFT) 目录中创建目录结构，并创建配置文件。
2. 运行 CSQUTIL 以定义 IBM MQ 资源。

如果协调队列管理器位于远程机器上，那么该作业步骤将失败。

成员 BCFCFCR 将在 USS 中创建文件并且会创建 MQ 定义。此作业将：

1. 创建 MFT 主题，
2. 创建 MFT 队列
3. 将 *NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)* 更改为
NAMES(SYSTEM.BROKER.DEFAULT.STREAM, SYSTEM.BROKER.ADMIN.STREAM, SYSTEM.FTE)
4. 执行 *ALTER QMGR PSMODE(ENABLED)*

在执行更改前发出 *DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)* 命令。如果您的 NAMLIST 不是缺省值，那么应更改名称列表以将 SYSTEM.FTE 添加到名称列表。

使用您自己的前缀重命名成员 BCFCFCR（例如 CCPCFCR），因为重新定制此文件会对其进行替换。

通过插入凭证文件的名称，编辑此已重命名的成员。例如：

```
%BFGCMD CMD=fteSetupCoordination +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
```

保存并提交此作业。请注意，如果需要重新提交此作业，必须添加 *-f* 选项。

运行此作业时，将列出创建的 IBM MQ 资源。您需要保护这些资源。

```
DEFINE TOPIC('SYSTEM.FTE') TOPICSTR('SYSTEM.FTE') REPLACE
ALTER TOPIC('SYSTEM.FTE') NPMGDLV(ALLAVAIL) PMSGDLV(ALLAVAIL)
DEFINE QLOCAL(SYSTEM.FTE) LIKE(SYSTEM.BROKER.DEFAULT.STREAM) REPLACE
ALTER QLOCAL(SYSTEM.FTE) DESCR('Stream for MFT Pub/Sub interface')
* Altering namelist: SYSTEM.QPUBSUB.QUEUE.NAMELIST
* Value prior to alteration:
```

```

DISPLAY NAMEDLIST(SYSTEM.QPUBSUB.QUEUE.NAMEDLIST)
ALTER NAMEDLIST(SYSTEM.QPUBSUB.QUEUE.NAMEDLIST) +
NAMES(SYSTEM.BROKER.DEFAULT.STREAM+
,SYSTEM.BROKER.ADMIN.STREAM,SYSTEM.FTE)
* Altering PSMODE. Value prior to alteration:
DISPLAY QMGR PSMODE
ALTER QMGR PSMODE(ENABLED)

```

相关概念

第 128 页的『定义命令队列管理器』

您需要命令队列管理器，但是，可以对协调和命令队列管理器使用相同的队列管理器。

定义命令队列管理器

您需要命令队列管理器，但是，可以对协调和命令队列管理器使用相同的队列管理器。

否则，您需要创建新的命令队列管理器。该命令队列管理器可以与协调队列管理器位于相同的机器上，但不是必须这样做。

使用您自己的前缀重命名成员 BFGCMCR（例如 CCPCMCR），因为重新定制此文件会对其进行替换。

通过插入凭证文件的名称，编辑此已重命名的成员。例如：

```

%BFGCMD CMD=fteSetupCommands +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +

```

保存并提交此作业。请注意，如果需要重新提交此作业，必须添加 *-f* 选项。

此队列管理器将用于诸如 **ftePingAgent** 之类的命令。

查看此成员，进行提交，然后查看输出。

请参阅第 128 页的『创建代理』，以获取有关如何创建代理的信息。

相关概念

第 127 页的『定义协调队列管理器』

IBM MQ Managed File Transfer 要求创建的队列管理器可充当协调队列管理器。

创建代理

您需要复制 PDSE 以生成特定于代理的 PDSE，例如 *user.MFT.AGENT1*。从上一个代理或记录器配置（如果存在）中复制 PDSE。如果这是您的第一个配置，请复制 MFT 随附的 PDSE。

查看成员 BFGCUSTM，如果需要使用其他凭证文件，请创建一个。

很多内容仍与第 126 页的『编辑成员 BFGCUSTM 并完成参数』中详细描述定制中的内容相同。

您需要：

- 更改 //SYSEXEC DD DSN=SCEN.FTE.JCL.AGENT1
- 更改 LIBRARY 以便与代理 PDSE 相匹配
- 更改 SERVICE_TYPE=AGENT
- 将 NAME 更改为代理（匹配 PDSE）JOB CARD 的名称
- 更改 BFG_JVM_PROPERTIES="-Xmx1024M"

提交此作业，请记住，作业需要数据集的独占访问权。

代理的作业全都具有 *BFGAG** 格式的名称。

重命名成员 *BFGAGCR*。此作业将更新 IBM MQ Managed File Transfer 目录中的文件，并使用 CSQUTIL 在本地队列管理器中创建特定于代理的队列。指定您的凭证文件的名称，例如 *-credentialsFile //'SCEN.FTE.JCL.VB(CREDOLD)*。如果未指定该名称，那么用于启动代理的作业不会使用凭证文件。

检查输出以确保进程已成功运行。

提示：将 *agent.properties* 文件的路径名从作业的输出复制到代理的 PDSE 中的成员。

例如，将 `/u/userid/fte/wmqmft/mqft/config/MQPA/agents/AGENT1/agent.properties` 复制到成员 AGENT 中。

如果需要显示属性文件并且添加行 `/u/userid/fte/wmqmft/mqft/logs/MQPA/agents/AGENT1/logs`，这将很有用。

这是跟踪文件的存储位置。

相关概念

第 127 页的『[定义协调队列管理器](#)』

IBM MQ Managed File Transfer 要求创建的队列管理器可充当协调队列管理器。

第 128 页的『[定义命令队列管理器](#)』

您需要命令队列管理器，但是，可以对协调和命令队列管理器使用相同的队列管理器。

第 129 页的『[使用代理](#)』

如何使用各种命令来确保代理正确工作。

使用代理

如何使用各种命令来确保代理正确工作。

启动代理

重命名成员 BFGAGST，查看该成员并提交作业。

如果有效，您将收到消息 BFGAG0059I：代理已成功启动。

显示活动的代理

重命名成员 BFGAGLI，查看该成员并提交使用协调队列管理器的作业。

您必须解决任何连接问题。

Ping 通代理以检查其是否在工作

重命名成员 BFGAGPI，查看该成员并提交使用命令队列管理器的作业。

您必须解决任何连接问题。

执行测试传输

请参阅第 134 页的『[执行验证传输](#)』，以获取进一步的信息。

停止代理

重命名成员 BFGAGSP，查看该成员并提交作业。

使用成员 BFGAGST 重新启动代理。

相关概念

第 128 页的『[创建代理](#)』

您需要复制 PDSE 以生成特定于代理的 PDSE，例如 `user.MFT.AGENT1`。从上一个代理或记录器配置（如果存在）中复制 PDSE。如果这是您的第一个配置，请复制 MFT 随附的 PDSE。

更新现有的 IBM MQ Managed File Transfer 代理或记录器命令数据集

您可以更新通过 MQMFT 命令模板数据集创建的 MQMFT 命令 PDSE 库数据集。

关于此任务

过程

1. 编辑 BFGCUSTM JCL 脚本成员，并更新 BFGSTDIN DD 语句中的变量和属性。

如果要除去先前定义的属性，请将其值设置为空，而不是除去条目。运行 BFGCUSTM JCL 脚本时，会将指定的属性将作为实际代理和记录器 UNIX 系统服务属性文件的更新加以应用；将属性设置为空值，该值指示将除去此属性。

2. 提交作业 BFGCUSTM。此作业将重新生成 JCL 命令集，适合于代理或记录器。要获取完整的命令列表，请参阅第 133 页的『z/OS 代理和记录器命令 JCL 脚本』。检查输出作业日志，以查看 JCL 脚本是否已成功运行。如果有任何故障，请纠正并重新提交 BFGCUSTM 作业。

结果

您可以修改生成的 JCL 脚本，并添加自己的逻辑。但是，在重新运行 BFGCUSTM 时应慎重，因为您可能会覆盖定制逻辑。

相关概念

第 118 页的『z/OS 上的配置选项』

z/OS 上的 IBM MQ Managed File Transfer 配置选项与针对分布式平台的选项相同。

相关任务

第 118 页的『创建 IBM MQ Managed File Transfer 代理或记录器命令数据集』

您可以通过 MQMFT 命令模板数据集，为特定协调的特定代理或记录器创建 PDSE 命令数据集。

z/OS JCL 变量

您可以在 BFGCUSTM 脚本中使用替换值、JCL 变量和配置属性。

下表列出了针对 MQMFT 命令 PDSE 库数据集中 BFGCUSTM JCL 脚本的替换值。在提交 BFGCUSTM 作业之前，必须将这些替换值替换为适当的值。

表 11: 针对 BFGCUSTM JCL 脚本的替换值	
替换变量	值
++library++	包含 MQMFT 命令 PDSE 库的数据集名称。
++bfg_java_home++	Java 安装的位置。
++bfg_prod++	MQMFT 产品安装 UNIX 系统服务根目录的位置。

下表描述了针对 MQMFT 命令 PDSE 库数据集中（在 [Variables] 部分中）BFGCUSTM JCL 脚本的 BFGSTDIN DD 语句的环境变量。在提交 BFGCUSTM 作业之前，必须将使用替换值（即用两个加号 ++ 括起来的值）指定的所有变量替换为适当的值。

表 12: 环境变量	
环境变量	值
LIBRARY	包含 MQMFT 命令 PDSE 库的数据集名称。
TMPDIR	临时文件的 UNIX 系统服务目录。
BFG_PROD	MQMFT 产品安装 UNIX 系统服务根目录的位置。
BFG_DATA	The location of the data directory for IBM MQ Managed File Transfer for z/OS, which is the path to <数据目录>.
BFG_JAVA_HOME	Java 安装的位置。
BFG_JVM_PROPERTIES	可选。设置 BFG_JVM_PROPERTIES 环境变量的值。这些属性将传递到 Java 虚拟机。

表 12: 环境变量 (继续)	
环境变量	值
BFG_GROUP_NAME	mqm 文件组通常与 MQMFT 配置数据文件和命令关联。因此，属于 mqm 组的所有用户都可以访问和更改 MQMFT 配置。要了解更多信息，请参阅第 431 页的『IBM MQ 中的 IBM MQ Managed File Transfer 的文件系统许可权』。 对于 z/OS 系统，文件组是 USS 文件系统实体，且不必定义 mqm 文件组。您可以使用 BFG_GROUP_NAME 环境变量来关联 MQMFT 配置数据文件的 z/OS USS 文件系统组。例如，在 USS shell 提示符中使用： <pre>export BFG_GROUP_NAME=FTEGB</pre> 该命令定义了 <i>FTEGB</i> 组，该组将与针对并发 USS 会话创建的任何后续配置文件关联。 您可以将 BFG_GROUP_NAME 设置为空值或将其移除。 注：当首次运行 BFGCUSTM 时，如果 MQMFT 配置要由多个用户标识使用，那么需要将 BFG_GROUP_NAME 设置为所有必需用户标识可访问的组。如果再次运行 BFGCUSTM，那么不得更改 BFG_GROUP_NAME（否则，必须同时更改目录中 BFG_DATA 引用的所有文件和目录的 USS 组文件权限，以反映新的 BFG_GROUP_NAME 设置）。 V 8.0.0.6 如果在将 BFG_GROUP_NAME 环境变量设置为非空白值的 z/OS 系统上运行 fteMigrateAgent 命令，那么该命令将检查用户是否是 BFG_GROUP_NAME 变量指定的组的成员。如果用户不在指定的组中，那么该命令可能会报告错误消息 BFGCL0502E：您无权执行所请求的操作。无法运行。有关用户为成功运行该命令而必须满足的条件的详细信息，请参阅第 540 页的『fteMigrate 代理 (将 WebSphere MQ File Transfer Edition V7.0 代理迁移到 IBM WebSphere MQ V7.5 或更高版本)』。
BFG_WTO	将 BFG_WTO 设置为 YES、ON 或 TRUE 时，将会启用 z/OS 记录。这将控制是否把写入代理事件日志的消息写入 z/OS 操作员日志工具，这在通过 JCL 运行代理时可以更轻松地访问自动化产品。路由代码是 Programmer Information (11)，描述符代码为 Informational (12)。
SERVICE_TYPE	指定 MQMFT 命令库是用于代理还是记录器。有效值为 AGENT 或 LOGGER。
名称	针对 SERVICE_TYPE 值的代理或记录器的名称。
QMGR	与针对 SERVICE_TYPE 值的代理或记录器关联的本地队列管理器的名称。
OUTPUT_CLASS	SYSOUT 数据集的输出类。缺省值为 *，这将请求与作业语句中 MSGCLASS 参数相同的输出类。

表 12: 环境变量 (继续)	
环境变量	值
MQ_PATH	在 BFGPROF 中用于创建 LIBPATH 环境变量。
MQ_HLQ	IBM MQ 数据集的高级限定符。
MQ_LANG	需要的语言。
DB2_HLQ	可选。Db2 数据集的高级限定符。
JOBCARD1	JCL 命令作业的标题行 1。
JOBCARD2	JCL 命令作业的标题行 2。
JOBCARD3	JCL 命令作业的标题行 3。
ADMIN_JOB1	admin 作业的标题行 1。
ADMIN_JOB2	admin 作业的标题行 2。
ADMIN_JOB3	admin 作业的标题行 3。
FTE_CONFIG	针对迁移的现有 WMQFTE 配置。如果无需迁移，请将其设置为空值。
CREDENTIAL_PATH	用于迁移的凭证文件的路径，例如 /u/user1/agent3。用于迁移 Managed File Transfer 8.0.0.0 的凭证文件必须位于到 WebSphere MQ File Transfer Edition 7.0.4.4 上的配置信息和配置文件的单独文件中。仅对于迁移命令 BFGAGMG 和 BFGLGMG JCL 脚本是必需的。如果无需迁移，请将其设置为空值。
DB_PROPS_PATH	为迁移指定数据库记录器属性文件。仅当属性文件未使用以下缺省名称和路径时，才需要此选项： config_directory/coordination_qmgr/ databaselogger.properties。如果无需迁移，请将其设置为空值。

注: IBM MQ jar 文件随 MQMFT 一起提供，在目录 <MQMFT product root>/java/lib 中始终使用，并且不可配置。

下表描述了针对 MQMFT 命令 PDSE 库数据集中 BFGCUSTM JCL 脚本的 BFGSTDIN DD 语句的必需 MQMFT 配置属性。在提交 BFGCUSTM 作业之前，必须将使用替换值（即用两个加号 ++ 括起来的值）指定的属性替换为适当的非空值。这些属性将为 MQMFT 配置属性定义覆盖项。您可以添加代理和记录器属性，为您的环境定制代理或记录器。要获取所有属性的列表，请参阅第 139 页的『配置属性文件』。

表 13: 针对 BFGSTDIN DD 语句的必需配置属性	
属性	值
coordinationQMGr	代理或记录器所关联配置的协调队列管理器的名称。
coordinationQMGrHost	可选。正在运行协调队列管理器的系统的主机名。如果将该属性的值留空，那么将采用绑定方式连接。
coordinationQMGrPort	可选。协调队列管理器正在侦听的端口号。只有当同时为 coordinationQMGrHost 属性指定了非空值时，才能使用该参数。
coordinationQMGrChannel	可选。用于连接到协调队列管理器的通道。只有当同时为 coordinationQMGrHost 属性指定了非空值时，才能使用该参数。
connectionQMGr	代理或记录器所关联配置的命令队列管理器的名称。

表 13: 针对 *BFGSTDIN DD* 语句的必需配置属性 (继续)

属性	值
connectionQMgrHost	可选。正在运行命令队列管理器的系统的主机名。如果将该属性的值留空，那么将采用绑定方式连接。
connectionQMgrPort	可选。命令队列管理器正在侦听的端口号。只有当同时为 connectionQMgrHost 属性指定了非空值时，才能使用该参数。
connectionQMgrChannel	可选。用于连接到命令队列管理器的通道。只有当同时为 connectionQMgrHost 属性指定了非空值时，才能使用该参数。

z/OS 代理和记录器命令 JCL 脚本

该表列举在 MQMFT 命令 PDSE 库数据集中可用的 JCL 命令集。

表 14:

成员	描述或 fte 命令行命令
BFGCOPY	用于创建此库副本的作业
BFGCUSTM	用于为代理或记录器定制此库的作业
BFGCFCR	fteSetupCoordination
BFGCMCR	fteSetupCommands
BFGAGCR	fteCreateAgent 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGLGCRS	fteCreateLogger 。仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGAGST	fteStartAgent 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGAGSTP	fteStartAgent 过程。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGAGPI	ftePingAgent 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGAGSP	fteStopAgent 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGLGST	fteStartLogger 。仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGLGSTP	fteStartLogger 过程。仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGLGSP	fteStopLogger 。仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGAGSH	fteShowAgentDetails 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGLGSH	fteShowLoggerDetails 。仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGCFDF	fteChangeDefaultConfigurationOptions

表 14: (继续)	
成员	描述或 fte 命令行命令
BFGAGCL	<code>fteCleanAgent</code> 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGAGDE	<code>fteDeleteAgent</code> 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGLGDE	<code>fteDeleteLogger</code> 。仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGPRSH	<code>fteDisplayVersion</code>
BFGAGLI	<code>fteListAgents</code> 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGMNLI	<code>fteListMonitors</code>
BFGSTLI	<code>fteListScheduledTransfers</code>
BFGTMLI	<code>fteListTemplates</code>
BFGAGMG	<code>fteMigrateAgent</code> 。仅当将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGLGMG	<code>fteMigrateLogger</code> 。仅当将 SERVICE_TYPE 变量设置为 LOGGER 时创建。
BFGCROBS	<code>fteObfuscate</code> 样本
BFGRAS	<code>fteRAS</code>
BFGAGTC	<code>fteSetAgentTraceLevel</code> 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGLGTC	第 557 页的『 <code>fteSetLoggerTraceLevel</code> 』。仅当将 SERVICE_TYPE 变量设置为 LOGGER 时创建。
BFGPRANS	<code>fteAnt</code> 样本
BFGTRCAS	<code>fteCancelTransfer</code> 样本
BFGMNCRS	<code>fteCreateMonitor</code> 样本
BFGTMCRS	<code>fteCreateTemplate</code> 样本
BFGTRCRS	<code>fteCreateTransfer</code> 样本
BFGMNDES	<code>fteDeleteMonitor</code> 样本
BFGSTDES	<code>fteDeleteScheduledTransfer</code> 样本
BFGTMDES	<code>fteDeleteTemplates</code> 样本

注: 对于创建 MQSC 或引用删除脚本的命令, JCL 将要求您运行一个脚本, 但该脚本已由作业运行。

执行验证传输

如何执行传输以检查产品是否在正确工作。

重命名并编辑成员 BFGTRCRS。

1. 在 %BFGCMD CMD=`fteCreateTransfer` -h 前添加 /*。
2. 除去成员中的其他注释。
3. 为 -sa 和 -da 指定当前代理名称

4. 保存 JCL

5. 提交 JCL

此 JCL 会连接到命令队列管理器。

配置记录任务

必须在与协调队列管理器相同的映像上运行记录任务。您可以记录到 Db2 中。

创建记录任务

复制 PDSE 以生成特定于记录器的 PDSE。例如，`user.MFT.LOGGER`。

如果需要使用其他凭证文件，请创建一个。有关更多信息，请参阅第 125 页的『创建 MFT 凭证文件』。

查看成员 `BFGCUSTM`。请注意，大多数内容仍与先前定制中的内容相同。

但是，您需要：

- 更改 `//SYSEXEC DD DSN=SCEN.FTE.JCL....`
- 更改 `LIBRARY` 以便与代理 PDSE 相匹配
- 将 `QMGR` 更改为协调队列管理器的名称
- 设置 `SERVICE_TYPE=LOGGER`
- 将 `NAME` 更改为记录器的名称（匹配 PDSE）
- 查看 `JOB CARD` 并更改作业名，使该名称与代理的作业名不同。
- 查看 `FTE_JVM_PROPERTIES="-Xmx1024M"`

如果正在使用 Db2 记录器，那么创建一个文件很有用，这样您就可以捕获 Db2 跟踪以帮助识别 Db2 问题。

在 JVM 属性中指定文件的名称，其中 JDBC 跟踪属性文件具有类似如下的内容：

```
db2.jcc.traceDirectory=/u/johndoe/fte
db2.jcc.traceFile=jccTrace1
db2.jcc.traceFileAppend=false
# turn on all traces
# db2.jcc.traceLevel=-1
# turn off all traces
db2.jcc.traceLevel=0
```

设置两个 JVM 属性

```
BFG_JVM_PROPERTIES=-Ddb2.jcc.propertiesFile=/u/.../sql.properties
-Ddb2.jcc.ssid=DBCA
```

其中 `/u/.../sql.properties` 是 Db2 跟踪属性文件的名称，`DBCA` 是 Db2 子系统的名称。

提交此作业，请注意，作业需要数据集的独占访问权。代理的作业全部具有诸如 `BFGLG*` 之类的名称。

记录到文件

有关记录到 Db2 的更多信息，请参阅第 136 页的『记录到 Db2 时创建记录任务』

重命名成员 `BFGLGCRS`。此作业将更新 IBM MQ Managed File Transfer (MFT) 目录中的文件，并使用 `CSQUTIL` 在本地队列管理器中创建特定于代理的队列。

原始文件具有命令 `%BFGCMD CMD=fteCreateLogger -h`，用于列示命令语法。

要通过在语句前放置 `/*` 来创建注释掉 `%BFGCMD CMD=fteCreateLogger -h` 的记录器任务，请确保第一列为空。

从第二列中移除注释，然后配置语句。例如：

```
%BFGCMD CMD=fteCreateLogger +
```

```
-p MQPH +
-loggerQMgr MQPH +
-loggerType FILE +
-fileLoggerMode circular +
-fileSize 5MB +
-fileCount 5 +
-p MQPH +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
LOGGER
```

检查输出以查看是否已成功处理。

提示: 将 `logger.properties` 文件的路径名从作业的输出复制到代理的 PDSE 中的成员。

例如，复制到成员 `APATH` 中

```
/u/<user ID>/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties
```

如果需要显示属性文件，那么此操作将很有用。

添加此文件的目录：

```
/u/<user ID>/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/
```

如果要记录到文件，那么日志文件将存储在此目录中，例如 `LOGGER0-20140522123654897.log`。

跟踪文件位于日志子目录中，例如

```
/u/<user ID>/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

您现在可以启动记录任务。

记录到 Db2 时创建记录任务

重命名成员 `BFGLGCRS`。

此作业将更新 MFT 目录中的文件，并使用 `CSQUTIL` 在本地队列管理器中创建特定于代理的队列。

您需要知道：

表 15: Db2 变量	
Db2 名称	示例
-dbName databaseName	您可以从 Db2 子系统的消息 DSNL004I 内的位置值中获取此信息。
-dbDriver filePath	例如， /db2/db2v10/jdbc/classes/db2jcc.jar
-dbLib filePath	例如， /db2/db2v10/jdbc/lib/libdb2jcct2zos_64.so

编辑文件。原始文件具有命令 `%BFGCMD CMD=fteCreateLogger -h`，用于列示命令语法。

从第二列中移除注释，然后配置语句。例如

```
%BFGCMD CMD=fteCreateLogger +
-p MQPH +
-loggerQMgr MQPH +
-loggerType DATABASE +
-dbType DB2 +
-dbName DSNDBCP +
-dbDriver /db2/db2v10/jdbc/classes/db2jcc.jar +
-dbLib /db2/db2v10/jdbc/lib/ +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
LOGGER
```

要通过在语句前放置 `/*` 来创建注释掉 `%BFGCMD CMD=fteCreateLogger -h` 的记录器任务，请确保第一列为空。

提交作业，并检查输出以查看是否已成功处理。

提示: 将 `logger.properties` 文件的路径名从作业的输出复制到代理的 PDSE 中的成员。

例如，复制到成员 `APATH` 中：

```
/u/<user ID>/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties into member USS
```

如果需要显示属性文件，那么此操作将很有用。

跟踪文件位于日志子目录中，例如：

```
/u/<user ID>/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

创建 Db2 表

您需要创建 Db2 表。定义位于 USS 文件 `mqft/sql/ftelog_tables_zos.sql` 中。

在您的 PDSE 中创建成员 `Db2`。编辑此成员，并在命令行上使用 `COPY` 命令。从 USS 定义文件进行复制。

由于特定于站点的需求可能会有很大的差异，因此该文件仅指定表的基本结构，以及它们所位于的表空间。

通过 SQL 脚本指定表空间，以确保使用缓冲池创建表空间，其中页面大小足以容纳可能的最大表行。请注意，未指定诸如 `LOB` 位置之类的属性。

您的数据库管理员可能需要修改该文件的副本，以定义这些与性能相关的属性。

该文件还将采用缺省模式名称 `FTELOG`、缺省表空间名称 `FTELOGTS` 和数据库名称 `FTELOGDB`。您可以遵循文件开头的注释中所描述的过程，根据需要更改这些名称，以便与现有的数据库和任何本地命名约定相匹配。

要点: 使用诸如 **SPUFI** 之类的在线工具来运行命令，因为文件中存在注释，而诸如 **DSNTINAD** 之类的批处理程序不接受注释。

启动记录器任务

重命名、查看和提交成员 `BFGLGST`。您应该会收到消息 `BFGDB0023I`：记录器已完成启动活动，现在正在运行。

记录器操作

要显示记录器状态，请重命名、查看并提交成员 `BFGLGSH`。

要停止记录器，请重命名、查看并提交成员 `BFGLGSP`。

IBM MQ Managed File Transfer for z/OS 的环境变量

如果直接通过 USS 环境或您自己的 JCL 脚本运行命令，那么在定制和配置之后，必须在运行 IBM MQ Managed File Transfer 提供的配置和管理脚本之前设置许多环境变量。必须在将调用脚本的每个环境中为每个用户设置这些变量。

为避免与其他产品冲突，您可以选择在主目录中创建 `.wmqfterc` 脚本。然后，每个 IBM MQ Managed File Transfer 脚本都将调用 `.wmqfterc` 脚本，并且您可以使用此脚本为 IBM MQ Managed File Transfer 提供定制环境设置。

同时还有一个可选的环境变量 `BFG_WTO`，通过 JCL 运行代理时，您可以设置该环境变量以将消息发送到操作程序日志。

表 16: 必需的 z/OS 环境变量	
环境变量	值
<code>BFG_JAVA_HOME</code>	Java 安装的位置。有关受支持的 Java 级别的更多信息，请参阅 WebSphere MQ 系统需求 。

表 16: 必需的 z/OS 环境变量 (继续)	
环境变量	值
BFG_DATA	IBM MQ Managed File Transfer for z/OS 的数据目录的位置。这是指向 <DATA_DIR> 的路径。
STEPLIB	必须包含以下 IBM MQ 数据集： • SCSQAUTH • SCSQANLE • SCSQLOAD 如果要在 z/OS 系统上运行数据库记录器组件，那么 STEPLIB 还必须按所示顺序包含以下 Db2 数据集： • SDSNEXIT • SDSNLOD2 • SDSNLOAD
LIBPATH	必须包含 IBM MQJava 库在 z/OS UNIX 系统服务空间中的位置 (对于 IBM MQ V 8，缺省值为 /mqm/V8R0M0/java/lib)。

以下是正确配置 IBM MQ Managed File Transfer 的环境变量的示例 .profile：

```
LIBPATH=/mqm/V8R0M0/java/lib:$LIBPATH
STEPLIB=MQM.V800.SCSQAUTH:MQM.V800.SCSQANLE:MQM.V800.SCSQLOAD
PATH=/u/fteuser/bin:/u/fteuser/J7.0/bin:/bin:/usr/bin:/u/fteuser/extras/bin:/bin:$PATH
BFG_JAVA_HOME=/u/fteuser/J7.0
BFG_DATA=/u/fteuser/<DATA_DIR>
export PATH LIBPATH STEPLIB BFG_JAVA_HOME BFG_DATA
```

此外，您还可以设置以下环境变量：

表 17: 可选 z/OS 环境变量	
环境变量	值
BFG_WTO	以下某个值将启用 BFG_WTO： • YES • 启用 • TRUE 以下某个值将禁用 BFG_WTO。这些值不区分大小写。 • NULL • 否 • 关闭 • FALSE 启用 z/OS 日志记录。缺省情况下，禁用该环境变量。 写入代理事件日志的消息也会写入 z/OS 操作程序日志设施，这在通过 JCL 运行代理时允许更轻松地了解自动化产品。路由代码是 Programmer Information (11)，描述符代码为 Informational (12)。

表 17: 可选 z/OS 环境变量 (继续)	
环境变量	值
BFG_GROUP_NAME	mqm 文件组通常与 MQMFT 配置数据文件和命令关联。因此，mqm 组中的所有成员用户都可以访问和更改 MQMFT 配置。有关更多信息，请参阅第 431 页的『IBM MQ 中的 IBM MQ Managed File Transfer 的文件系统许可权』。 对于 z/OS 系统，文件组是 USS 文件系统实体，并且不必定义 mqm 文件组。您可以使用 BFG_GROUP_NAME 环境变量，为 MQMFT 配置数据文件定义可替代的现有 z/OS USS 文件系统组。例如，在 USS shell 提示符： <pre>export BFG_GROUP_NAME=FTEGB</pre> 该命令定义了 FTEGB 组，该组将与针对并发 USS 会话创建的任何后续配置文件关联。 您可以将 BFG_GROUP_NAME 设置为空值或将其移除。 V 8.0.0.6 如果在将 BFG_GROUP_NAME 环境变量设置为非空白值的 z/OS 系统上运行 fteMigrateAgent 命令，那么该命令将检查用户是否是 BFG_GROUP_NAME 变量指定的组的成员。如果用户不在指定的组中，那么该命令可能会报告错误消息 BFGCL0502E：您无权执行所请求的操作。无法运行。有关用户为成功运行该命令而必须满足的条件的相关信息，请参阅第 540 页的『fteMigrate 代理 (将 WebSphere MQ File Transfer Edition V7.0 代理迁移到 IBM WebSphere MQ V7.5 或更高版本)』。

配置属性文件

IBM MQ Managed File Transfer 中所使用属性的摘要。

- 第 584 页的『[coordination.properties 文件](#)』
- 第 588 页的『[command.properties 文件](#)』
- 第 591 页的『[agent.properties 文件](#)』
- [记录器配置属性文件](#)

针对 z/OS 自动重新启动管理器 (ARM) 配置 MQMFT

IBM MQ Managed File Transfer 是启用了 ARM 的应用程序。

开始之前

有关启用 ARM 以及定义适用于您的系统的 ARM 策略的更多信息，请参阅[使用 z/OS 自动重新启动管理器 \(ARM\)](#)。

如果要使用 MFT DB 记录器功能自动重新启动并重新连接到 Db2 数据库，那么 ARM 是唯一受支持的重新启动管理器。

关于此任务

使用 ARM 时，可以通过设置代理/记录器属性 `armELEMTYPE` 和 `armELEMENT`，对代理和记录器进行配置以进行重新启动。属性 `armELEMTYPE` 定义了 ARM 元素的类型，而属性 `armELEMENT` 是 ARM 将要注册的元素的名称：

- 您可以将代理 `ELEMTYPE` 设置为 `SYSBFGAG`，将 `armELEMENT` 设置为与代理名称一致。
- 您可以将记录器 `ELEMTYPE` 设置为 `SYSBFLG`，将 `armELEMENT` 设置为与记录器名称一致。

注：由 ARM 配置为进行重新启动的代理和记录器只能从批处理作业或已启动的任务中成功运行。如果尝试直接从 USS 命令行启动代理或记录器，将会失败，并显示 ARM 错误原因码。

示例

以下重新启动策略的示例定义了依赖于队列管理器 `FT7C` 的代理 `BFGFT7CAG1`：

```
RESTART_ORDER
  LEVEL(3)
  ELEMENT_TYPE(SYSBFGAG,SYSBFLG)

RESTART_GROUP(GROUP7C)
  ELEMENT(SYSMQMGRFT7C)
  ELEMENT(BFGFT7CAG1)
  RESTART_ATTEMPTS(3,300)
```

在 IBM i 系统上安装 IBM MQ Managed File Transfer 后进行配置

要在安装后开始使用 IBM MQ Managed File Transfer，必须为协调队列管理器和代理完成某些配置。

关于此任务

安装后，必须运行 IBM MQ Managed File Transfer 为新的协调队列管理器和新的代理提供的配置脚本，然后才可使用协调队列管理器和代理来传输文件。随后，必须启动已创建的代理。

过程

1. 对于所有新的协调队列管理器：对协调队列管理器运行 `coordination_qmgr_name.mqsc` 文件中的 MQSC 命令。如果协调队列管理器与安装不在同一台计算机上，那么请将 MQSC 脚本文件复制到队列管理器所在计算机，然后运行该脚本。
 - a) 在 IBM i 命令行中，使用以下命令启动 qshell: `CALL QSHELL`
 - b) 切换到以下目录： `/QIBM/UserData/mqm/mqft/config/coordination_qmgr_name`
 - c) 发出以下命令将 `coordination_qmgr_name` 替换为您的队列管理器的名称：

```
/QSYS.LIB/QMQM.LIB/RUNMQSC.PGM coordination_qmgr_name < coordination_qmgr_name.mqsc
```

您可以改为手动配置协调队列管理器。有关更多信息，请参阅第 144 页的『配置协调队列管理器』。

2. 对于所有新代理程序：在 `<agent_name>_create.mqsc` 文件中针对代理队列管理器运行 MQSC 命令。如果代理队列管理器与代理不在同一台计算机上，请将 MQSC 脚本文件复制到队列管理器所在计算机，然后运行该脚本。
 - a) 在 IBM i 命令行中，使用以下命令启动 qshell: `CALL QSHELL`
 - b) 切换到以下目录： `/QIBM/UserData/mqm/mqft/config/agent_qmgr_name/agents`
 - c) 发出以下命令以将 `agent_qmgr_name` 替换为您的代理队列管理器的名称，将 `agent_name` 替换为您的代理的名称：

```
/QSYS.LIB/QMQM.LIB/RUNMQSC.PGM agent_qmgr_name < agent_name_create.mqsc
```

您可以改为手动配置代理队列管理器。有关更多信息，请参阅第 144 页的『配置代理队列管理器』。

3. 如果尚未在安装过程中启动 QMFT 子系统，请使用以下命令从 IBM i 命令行启动 QMFT 子系统: STRSBS SBSD(QMQMMFT/QMFT) 或 STRSBS QMQMMFT/QMFT
4. 使用 **fteStartAgent** 命令启动您的新代理。
 - a) 在 IBM i 命令中，使用以下命令启动 qshell: CALL QSHELL
 - b) 切换到以下目录: /QIBM/ProdData/mqm/bin
 - c) 发出以下命令以将 AGENT 替换为您的代理名称:

```
./fteStartAgent AGENT
```

下一步做什么

建议您设置沙箱以限制代理可访问的文件系统区域。在[使用沙箱](#)中描述了该功能。

相关概念

第 141 页的『[为首次使用配置 IBM MQ Managed File Transfer](#)』

在首次使用 IBM MQ Managed File Transfer 代理和队列管理器之前，必须先对其执行一些配置任务。

为首次使用配置 IBM MQ Managed File Transfer

在首次使用 IBM MQ Managed File Transfer 代理和队列管理器之前，必须先对其执行一些配置任务。

相关概念

第 142 页的『[连接到 IBM MQ](#)』

与 IBM MQ 队列管理器的所有网络通信（包括与 IBM MQ Managed File Transfer 相关的通信），都涉及 IBM MQ 通道。IBM MQ 通道代表网络链路的一端。通道可以归类为消息通道或 MQI 通道。

第 148 页的『[IBM MQ 多实例队列管理器](#)』

从 WebSphere MQ V7.0.1 开始，支持创建多实例队列管理器。多实例队列管理器能够在备用服务器上自动地重新启动。IBM MQ Managed File Transfer 支持连接到多个多实例代理队列管理器、一个多实例协调队列管理器和一个多实例命令队列管理器。

相关任务

第 142 页的『[配置 IBM MQ 队列管理器](#)』

如果您的 IBM MQ Managed File Transfer 网络包含多个 IBM MQ 队列管理器，那么这些 IBM MQ 队列管理器相互之间必须能够进行远程通信。

第 144 页的『[配置代理队列管理器](#)』

安装后，运行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中的 `agent_name_create.mqsc` 脚本以对代理队列管理器执行必要配置。但是，如果您希望手动执行此配置，那么请在代理队列管理器上完成以下步骤：

第 144 页的『[配置协调队列管理器](#)』

运行 **fteSetupCoordination** 命令之后，运行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目录中的 `coordination_qmgr_name.mqsc` 脚本以对协调队列管理器执行必要配置。但是，如果您希望手动执行此配置，请在协调队列管理器上完成以下步骤。

第 118 页的『[创建 IBM MQ Managed File Transfer 代理或记录器命令数据集](#)』

您可以通过 MQMFT 命令模板数据集，为特定协调的特定代理或记录器创建 PDSE 命令数据集。

第 129 页的『[更新现有的 IBM MQ Managed File Transfer 代理或记录器命令数据集](#)』

您可以更新通过 MQMFT 命令模板数据集创建的 MQMFT 命令 PDSE 库数据集。

相关参考

第 697 页的『[IBM MQ Managed File Transfer 的代理队列](#)』

由 **fteCreateAgent** 命令生成的 MQSC 命令脚本使用设置为以下值的参数创建代理队列。如果不使用提供用于创建队列的 MQSC 脚本，而是手动创建队列，那么请确保将以下参数设置为给定的值。

第 699 页的『[系统队列和系统主题](#)』

IBM MQ Managed File Transfer 包含大量系统队列和一个仅限内部使用的系统主题。

第 150 页的『[确保保留 IBM MQ Managed File Transfer 日志消息](#)』

IBM MQ Managed File Transfer 将文件传输进度和日志信息发送到协调队列管理器。协调队列管理器将此信息发布到 `SYSTEM.FTE` 主题的任何匹配预订。如果无任何预订，将不保留此信息。

连接到 IBM MQ

与 IBM MQ 队列管理器的所有网络通信（包括与 IBM MQ Managed File Transfer 相关的通信），都涉及 IBM MQ 通道。IBM MQ 通道代表网络链路的一端。通道可以归类为消息通道或 MQI 通道。

IBM MQ Managed File Transfer 和通道

IBM MQ Managed File Transfer 使用 MQI 通道以客户机方式将代理连接到其代理队列管理器，并将命令应用程序（例如 `fteCreateTransfer`）连接到其命令和协调队列管理器。在缺省配置中，这些连接由一个名为 `SYSTEM.DEF.SVRCONN` 的 `SVRCONN` 通道建立，`SYSTEM.DEF.SVRCONN` 缺省情况下存在于所有队列管理器上。由于这些缺省设置，因此您无需对 IBM MQ Managed File Transfer 基本安装更改任何 MQI 通道。

消息通道端点的类型有六种，但是本主题中只讨论发送方/接收方对。请参阅[分布式排队组件](#)以了解有关其他通道组合的信息。

所需的消息路径

IBM MQ 消息只能在消息通道上传输，因此您必须确保这些通道可用于 IBM MQ Managed File Transfer 所需的所有消息路径。这些路径不必是直接的；如果需要的话，可以通过中间队列管理器传递消息。本主题只讨论点到点的直接通信。请参阅[如何与远程队列管理器通信](#)，以了解有关这些选项的更多信息。

IBM MQ Managed File Transfer 使用的通信路径有以下几种：

代理到代理

在其关联的队列管理器之间传输文件需要双向通信的任何两个代理。由于该路径中传输批量数据，因此您可以根据需要，考虑使路径尽可能短、快速且经济。

代理到协调

参与传输的代理发出的日志消息必须能够传递到协调队列管理器。

命令到代理

命令应用程序或 IBM MQ Explorer（使用命令队列管理器）连接到的所有队列管理器都必须能够向使用这些命令应用程序控制的代理的队列管理器发送消息。要使命令显示反馈消息，应使用双向连接。

有关更多信息，请参阅[使用命令行验证服务器到服务器安装](#)。

相关概念

第 148 页的『IBM MQ 多实例队列管理器』

从 WebSphere MQ V7.0.1 开始，支持创建多实例队列管理器。多实例队列管理器能够在备用服务器上自动地重新启动。IBM MQ Managed File Transfer 支持连接到多个多实例代理队列管理器、一个多实例协调队列管理器和一个多实例命令队列管理器。

相关任务

第 142 页的『配置 IBM MQ 队列管理器』

如果您的 IBM MQ Managed File Transfer 网络包含多个 IBM MQ 队列管理器，那么这些 IBM MQ 队列管理器相互之间必须能够进行远程通信。

第 144 页的『配置协调队列管理器』

运行 `fteSetupCoordination` 命令之后，运行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目录中的 `coordination_qmgr_name.mqsc` 脚本以对协调队列管理器执行必要配置。但是，如果您希望手动执行此配置，请在协调队列管理器上完成以下步骤。

配置 IBM MQ 队列管理器

如果您的 IBM MQ Managed File Transfer 网络包含多个 IBM MQ 队列管理器，那么这些 IBM MQ 队列管理器相互之间必须能够进行远程通信。

关于此任务

要实现队列管理器之间的互相通信，您可以使用以下两种方法进行配置：

- 通过设置 IBM MQ 队列管理器集群。

要了解有关 IBM MQ 队列管理器集群及其配置方式的信息，请参阅[配置队列管理器集群](#)。

- 通过设置队列管理器之间的通道，如下所述：

设置队列管理器之间的通道

在您的队列管理器之间，设置以下消息通道：

- 从代理队列管理器到协调队列管理器
- 从命令队列管理器到代理队列管理器。
- 从代理队列管理器到命令队列管理器（从而能够通过命令显示反馈消息）。
- 从命令队列管理器到协调队列管理器
- 从代理队列管理器到 IBM MQ Managed File Transfer 网络中的其他任何代理队列管理器

如果需要有关如何设置这种通信的更多信息，请首先参阅以下信息：[使用 MQSC 管理远程 IBM MQ 对象](#)。

以下是建议的几个示例步骤：

过程

1. 在 IBM MQ 队列管理器上，创建一个与协调队列管理器同名的传输队列。

您可以使用以下 MQSC 命令：

```
DEFINE QLOCAL(coordination-qmgr-name) USAGE(XMITQ)
```

2. 在 IBM MQ 队列管理器上，创建到 IBM MQ Managed File Transfer 协调队列管理器的发送方通道。先前步骤中创建的传输队列的名称是此通道的必需参数。如果需要与 IBM MQ Managed File Transfer V7.5 或 WebSphere MQ File Transfer Edition 代理进行通信，请确保将发送方通道的 CONVERT 参数设置为 no。（IBM MQ Managed File Transfer 的更早版本始终以 UTF-8 格式发布消息，这意味着任何数据转换都有可能破坏消息。对于 IBM MQ Managed File Transfer V8.0 或更高版本上的代理，无需此设置，因为将会以空白格式发布消息。）

您可以使用以下 MQSC 命令：

```
DEFINE CHANNEL(channel-name) CHLTYPE(SDR) CONNAME('coordination-qmgr-host(coordination-qmgr-port)')  
XMITQ(coordination-qmgr-name) CONVERT(NO)
```

注：仅在需要时设置 CONVERT(NO)。

3. 在 IBM MQ Managed File Transfer 协调队列管理器上，创建到 IBM MQ 队列管理器的接收方通道。为该接收方通道指定与 IBM MQ 队列管理器上的发送方通道相同的名称。

您可以使用以下 MQSC 命令：

```
DEFINE CHANNEL(channel-name) CHLTYPE(RCVR)
```

下一步做什么

接下来，执行协调队列管理器的配置步骤：[配置协调队列管理器](#)。

相关概念

第 142 页的『[连接到 IBM MQ](#)』

与 IBM MQ 队列管理器的所有网络通信（包括与 IBM MQ Managed File Transfer 相关的通信），都涉及 IBM MQ 通道。IBM MQ 通道代表网络链路的一端。通道可以归类为消息通道或 MQI 通道。

第 148 页的『[IBM MQ 多实例队列管理器](#)』

从 WebSphere MQ V7.0.1 开始，支持创建多实例队列管理器。多实例队列管理器能够在备用服务器上自动地重新启动。IBM MQ Managed File Transfer 支持连接到多个多实例代理队列管理器、一个多实例协调队列管理器和一个多实例命令队列管理器。

相关任务

第 144 页的『配置协调队列管理器』

运行 **fteSetupCoordination** 命令之后，运行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目录中的 `coordination_qmgr_name.mqsc` 脚本以对协调队列管理器执行必要配置。但是，如果您希望手动执行此配置，请在协调队列管理器上完成以下步骤。

配置协调队列管理器

运行 **fteSetupCoordination** 命令之后，运行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目录中的 `coordination_qmgr_name.mqsc` 脚本以对协调队列管理器执行必要配置。但是，如果您希望手动执行此配置，请在协调队列管理器上完成以下步骤。

关于此任务

过程

1. 创建一个名为 SYSTEM.FTE 的本地队列。
2. 将 SYSTEM.FTE 队列添加到 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名称列表中。
3. 创建一个名为 SYSTEM.FTE 的主题，主题的字符串为 SYSTEM.FTE。
4. 确保 SYSTEM.FTE 主题的非持久性消息传递 (NPMGDLV) 属性和持久性消息传递 (PMGDLV) 属性设置为 ALLAVAIL。
5. 确保协调队列管理器的发布/预订方式 (PSMODE) 属性设置为 ENABLED。

下一步做什么

如果在已配置为协调队列管理器的队列管理器上运行 `strmqm -c` 命令，那么该命令会删除在 [步骤 2](#) (添加 SYSTEM.FTE 队列到 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名称列表)。这是因为 `strmqm -c` 会重新创建缺省 IBM MQ 对象并撤销 IBM MQ Managed File Transfer 更改。因此，如果已使用 `strmqm -c` 启动队列管理器，请完成以下步骤之一：

- 重新在队列管理器上运行 `coordination_qmgr_name.mqsc` 脚本。
- 重复[步骤 2](#)。

相关概念

第 142 页的『连接到 IBM MQ』

与 IBM MQ 队列管理器的所有网络通信（包括与 IBM MQ Managed File Transfer 相关的通信），都涉及 IBM MQ 通道。IBM MQ 通道代表网络链路的一端。通道可以归类为消息通道或 MQI 通道。

第 148 页的『IBM MQ 多实例队列管理器』

从 WebSphere MQ V7.0.1 开始，支持创建多实例队列管理器。多实例队列管理器能够在备用服务器上自动地重新启动。IBM MQ Managed File Transfer 支持连接到多个多实例代理队列管理器、一个多实例协调队列管理器和一个多实例命令队列管理器。

相关任务

第 142 页的『配置 IBM MQ 队列管理器』

如果您的 IBM MQ Managed File Transfer 网络包含多个 IBM MQ 队列管理器，那么这些 IBM MQ 队列管理器相互之间必须能够进行远程通信。

相关参考

第 562 页的『fteSetupCoordination（设置协调详细信息）』

fteSetupCoordination 命令为 IBM MQ Managed File Transfer 创建属性文件和协调队列管理器目录。

配置代理队列管理器

安装后，运行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中的 `agent_name_create.mqsc` 脚本以对代理队列管理器执行必要配置。但是，如果您希望手动执行此配置，那么请在代理队列管理器上完成以下步骤：

关于此任务

过程

1. 创建代理操作队列。

这些队列的名称如下：

- `SYSTEM.FTE.COMMAND.agent_name`
- `SYSTEM.FTE.DATA.agent_name`
- `SYSTEM.FTE.EVENT.agent_name`
- `SYSTEM.FTE.REPLY.agent_name`
- `SYSTEM.FTE.STATE.agent_name`

要了解有关队列参数的更多信息，请参阅 [第 697 页的『IBM MQ Managed File Transfer 的代理队列』](#)。

2. 创建代理权限队列。

这些队列的名称如下：

- `SYSTEM.FTE.AUTHADM1.agent_name`
- `SYSTEM.FTE.AUTHAGT1.agent_name`
- `SYSTEM.FTE.AUTHMON1.agent_name`
- `SYSTEM.FTE.AUTHOPS1.agent_name`
- `SYSTEM.FTE.AUTHSCH1.agent_name`
- `SYSTEM.FTE.AUTHTRN1.agent_name`

要了解有关队列参数的更多信息，请参阅 [第 697 页的『IBM MQ Managed File Transfer 的代理队列』](#)。

3. 如果代理是 Web 代理，请创建 Web 代理操作队列。

这些队列的名称如下：

- `SYSTEM.FTE.WEB.gateway_name`
- `SYSTEM.FTE.WEB.RESP.agent_name`
- 要了解有关队列参数的更多信息，请参阅 [第 697 页的『IBM MQ Managed File Transfer 的代理队列』](#)。

下一步做什么

有关创建和配置协议网桥代理的信息，请参阅 [第 463 页的『fteCreateBridgeAgent（创建和配置 IBM MQ Managed File Transfer 协议网桥代理）』](#) 和 [第 286 页的『配置 FTPS 服务器的协议网桥』](#)。

相关概念

[第 142 页的『连接到 IBM MQ』](#)

与 IBM MQ 队列管理器的所有网络通信（包括与 IBM MQ Managed File Transfer 相关的通信），都涉及 IBM MQ 通道。IBM MQ 通道代表网络链路的一端。通道可以归类为消息通道或 MQI 通道。

[第 148 页的『IBM MQ 多实例队列管理器』](#)

从 WebSphere MQ V7.0.1 开始，支持创建多实例队列管理器。多实例队列管理器能够在备用服务器上自动地重新启动。IBM MQ Managed File Transfer 支持连接到多个多实例代理队列管理器、一个多实例协调队列管理器和一个多实例命令队列管理器。

相关任务

[第 142 页的『配置 IBM MQ 队列管理器』](#)

如果您的 IBM MQ Managed File Transfer 网络包含多个 IBM MQ 队列管理器，那么这些 IBM MQ 队列管理器相互之间必须能够进行远程通信。

[第 144 页的『配置协调队列管理器』](#)

运行 `fteSetupCoordination` 命令之后，运行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目录中的 `coordination_qmgr_name.mqsc` 脚本以对协调队列管理器执行必要配置。但是，如果您希望手动执行此配置，请在协调队列管理器上完成以下步骤。

相关参考

第 697 页的『IBM MQ Managed File Transfer 的代理队列』

由 **fteCreateAgent** 命令生成的 MQSC 命令脚本使用设置为以下值的参数创建代理队列。如果不使用提供用于创建队列的 MQSC 脚本，而是手动创建队列，那么请确保将以下参数设置为给定的值。

第 562 页的『fteSetupCoordination（设置协调详细信息）』

fteSetupCoordination 命令为 IBM MQ Managed File Transfer 创建属性文件和协调队列管理器目录。

创建 IBM MQ 文件传输结构

您可以根据连接到同一台机器上的队列管理器的单一代理，配置 IBM MQ Managed File Transfer 结构。

关于此任务

MQMFT 配置存储在代理所在的机器上，IBM MQ DataPath 下的文件结构中。

以下样本配置适用于名为 SAMPLECOORD (已禁用安全性) 的 MQMFT V8 队列管理器以及名为 SAMPLEAGENT 的单个 MQMFT 代理程序：

```
+--- config
      +--- SAMPLECOORD
            +--- command.properties
            +--- coordination.properties
            +--- SAMPLECOORD.mqsc
            +--- agents
      +--- SAMPLEAGENT
            +--- agent.properties
            +--- SAMPLEAGENT_create.mqsc
            +--- SAMPLEAGENT_delete.mqsc

+---logs
      +--- SAMPLECOORD
            +--- agents
      +--- SAMPLEAGENT
            +--- logs
```

此示例假定队列管理器安全性已禁用。以下命令（在 **runmqsc** 中运行）将在重新启动队列管理器后禁用安全性：

```
runmqsc <queue manager>
alter qmgr CONNAUTH(NONE);
alter qmgr CHLAUTH(DISABLED);
end;
```

对于在 MQMFT V8 中启用了安全性的配置，**CONNAUTH** 需要所有与队列管理器相连的 MQMFT 命令来提供用户标识和密码凭证。您可以为每个命令额外应用参数 **-mquserid** 和 **-mqpassword**，或者定义 MQMFTCredentials.xml 文件。以下样本凭证文件定义了 fteuser 的用户标识，当连接到队列管理器 SAMPLECOORD 时将对其应用 MyPassword 密码：

```
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/MQMFTCredentials MQMFTCredentials.xsd">
  <tns:qmgr mqPassword="MyPassword" MyUserId="fteuser" name="SAMPLECOORD"/>
</tns:mqmftCredentials>
```

有关更多信息，请参阅第 97 页的『IBM MQ Managed File Transfer 和 IBM MQ 连接认证』。

注意：

- 要查找 MQMFT 配置目录，请使用 **fteDisplayVersion -v** 命令。
- 对于 z/OS 用户，MQMFTCredential.xml 文件可以作为分区数据集的成员来查找，其变量记录格式为 (RECFM=V)，或未定义记录格式 (RECFM=U)。
- 对于启用了安全性的配置，请将以下参数添加到以下步骤以将凭证与相关队列管理器相关联：
-credentialsFile <full credential file path>。

- MQMFTCredential.xml 中的明文密码可以使用以下命令进行加密：

```
fteObfuscate -credentialsFile <full file path to MQMFTCredentials.xml>
```

过程

1. 创建坐标。

坐标是单一队列管理器，用于接收来自其代理的所有传输日志和状态信息。运行以下命令：

```
fteSetupCoordination -coordinationQMGr <coordination_qmgr_name>
```

这创建了基本顶级配置，并创建了 IBM MQ 脚本文件来调用 `<coordination_qmgr_name>.mqsc`。然后，需要运行以下 IBM MQ 命令，将该配置装入到队列管理器中：

```
runmqsc <queue manager name> < <coordination_qmgr_name>.mqsc
```

注：要将 TCP 客户机连接到队列管理器，您可以使用：

```
fteSetupCoordination -coordinationQMGr <coordination_qmgr_name>  
-coordinationQMGrHost <coordination_qmgr_host> -coordinationQMGrPort  
<coordination_qmgr_port>  
-coordinationQMGrChannel <coordination_qmgr_channel>
```

对于已创建的 `<coordination_qmgr_name>.mqsc`，您将需要在协调队列管理器运行的同一台机器上运行 **runmqsc** 命令。

2. 创建命令。

命令是已经预先配置的单一队列管理器，以便 IBM MQ 基础结构可以将 MQMFT 请求路由至相关的代理。运行以下命令：

```
fteSetupCommands -connectionQMGr <Command QM Name> -p <Coordination QM Name>
```

这将在协调目录中创建 `command.properties` 文件。请注意，`-p` 是可选的，如果针对缺省坐标设置了命令，那么将不需要该选项。

注：要将 TCP 客户机连接到队列管理器，您可以使用：

```
fteSetupCommands -p <coordination_qmgr_name> -commandQMGr <connection_qmgr_name>  
-commandQMGrHost <connection_qmgr_host> -commandQMGrPort <connection_qmgr_port>  
-commandQMGrChannel <connection_qmgr_channel>
```

3. 创建代理。

代理是用于发送和接收文件的应用程序。运行以下命令：

```
fteCreateAgent -p <coordination_qmgr_name> -agentName <agent_name> -agentQMGr  
<agent_qmgr_name>
```

这将在坐标下创建代理配置，并创建 IBM MQ 脚本文件以调用代理配置目录中的 `<agent_name>.mqsc`。

运行以下 IBM MQ 命令，以将 IBM MQ 脚本文件装入到队列管理器中：

```
runmqsc <agent_qmgr_name> < <agent_name>_create.mqsc file
```

注：要将 TCP 客户机连接到队列管理器，您可以使用：

```
fteCreateAgent -p <coordination_qmgr_name> -agentName <agent_name> -agentQMGr  
<agent_qmgr_name>
```

```
-agentQMgrHost <agent_qmgr_host> -agentQMgrPort <agent_qmgr_port> -agentQMgrChannel  
<agent_qmgr_channel>
```

4. 启动代理。

运行以下命令：

```
fteStartAgent -p <coordination_qmgr_name> <agentName>
```

代理将在后台启动，并返回命令提示符。要检查代理是否正在运行，请运行以下命令：

```
fteListAgents -p <coordination_qmgr_name>
```

这将显示代理的状态。如果代理成功运行，那么该代理将报告为处于“就绪”状态。

结果

基本 MQMFT 基础结构准备就绪，您现在可以使用 **fteCreateTransfer** 命令请求传输。或者，如果 IBM MQ 资源管理器可用，使用 MQMFT 插件创建和监视传输。

通过重复步骤 3：创建代理，可向配置添加更多代理。如果使用了 TCP 客户机连接，那么这些代理可以位于不同机器上。对于不同机器，必须对每台机器重复 **fteSetupCoordination** 和 **fteSetupCommands** 命令，但是，不需要再运行 mqsc 脚本。

较复杂的配置的坐标和每个代理可能有独立的队列管理器。在这些情况中，需要将各种队列管理器连接起来。

相关参考

[第 562 页的『fteSetupCoordination（设置协调详细信息）』](#)

fteSetupCoordination 命令为 IBM MQ Managed File Transfer 创建属性文件和协调队列管理器目录。

[第 560 页的『fteSetupCommands（创建 command.properties 文件）』](#)

fteSetupCommands 命令可创建 command.properties 文件。在您发出命令时，该属性文件指定与 IBM MQ 网络连接的队列管理器的详细信息。

[第 460 页的『fteCreateAgent（创建 IBM MQ Managed File Transfer 代理）』](#)

fteCreateAgent 命令创建代理及其关联的配置。

[第 550 页的『fteObfuscate（加密敏感数据）』](#)

fteObfuscate 命令用于加密凭证文件中的敏感数据。这样，获取了对凭证文件的访问权的人员就无法读取该文件的内容。

[第 873 页的『MQMFT 凭证文件格式』](#)

MQMFTCredentials.xml 文件包含敏感的用户标识和密码信息。MQMFTCredentials.xml 文件中的元素必须符合 MQMFTCredentials.xsd 模式。凭证文件的安全性由用户负责。

[第 591 页的『agent.properties 文件』](#)

每个代理都有自己的属性文件 agent.properties，其中必须包含代理用于连接到其队列管理器的信息。agent.properties 文件还可以包含用于更改代理程序行为的属性。

[第 378 页的『如果 fteListAgents 命令未列出您的代理，该怎么办』](#)

如果 **fteListAgents** 命令未列出您的代理、您的代理未显示在 IBM MQ Explorer 中，或者您的文件传输未显示在 IBM MQ Explorer 的“传输日志”中，那么可以执行若干问题确定步骤来调查原因。

IBM MQ 多实例队列管理器

从 WebSphere MQ V7.0.1 开始，支持创建多实例队列管理器。多实例队列管理器能够在备用服务器上自动地重新启动。IBM MQ Managed File Transfer 支持连接到多个多实例代理队列管理器、一个多实例协调队列管理器和一个多实例命令队列管理器。

配置多实例队列管理器

要点：有关配置 IBM MQ 多实例队列管理器的信息，请参阅[多实例队列管理器](#)。在尝试配置多实例队列管理器以使用 IBM MQ Managed File Transfer 之前，请确保您已读取此信息。

将多实例队列管理器用作代理队列管理器

要使代理同时连接到多实例队列管理器的活动和备用实例，请将 `agentQMGrStandby` 属性添加到代理的 `agent.properties` 文件中。 `agentQMGrStandby` 属性为备用队列管理器实例定义用于客户机连接的主机名和端口号。必须以 MQ CONNAME 格式提供属性值，即 `host_name(port_number)`。

`agentQMGr` 属性指定多实例队列管理器的名称。 `agentQMGrHost` 属性为活动队列管理器实例指定主机名， `agentQMGrPort` 属性为活动队列管理器实例指定端口号。代理必须以客户机方式连接到多实例队列管理器的活动和备用实例。

请参阅 [第 591 页的『agent.properties 文件』](#)，以了解更多信息。

该示例显示了连接到名为 QM_JUPITER 的多实例队列管理器的 AGENT1 的 `agent.properties` 文件的内容。 QM_JUPITER 的活动实例位于系统 `host1` 上，并使用端口号 1414 进行客户机连接。 QM_JUPITER 的备用实例位于系统 `host2` 上，并使用端口号 1414 进行客户机连接。

```
agentName=AGENT1
agentDesc=
agentQMGr=QM_JUPITER
agentQMGrPort=1414
agentQMGrHost=host1
agentQMGrChannel=SYSTEM.DEF.SVRCONN
agentQMGrStandby=host2(1414)
```

将多实例队列管理器用作协调队列管理器

要启用与多实例协调队列管理器的活动实例和备用实例的连接，请将 `coordinationQMGrStandby` 属性添加到 IBM MQ Managed File Transfer 拓扑中的所有 `coordination.properties` 文件。

请参阅 [第 584 页的『coordination.properties 文件』](#)，以了解更多信息。

该示例显示了指定名为 QM_SATURN 的多实例协调队列管理器的连接详细信息的内容。 QM_SATURN 的活动实例位于系统 `coordination_host1` 上，并使用端口号 1420 进行客户机连接。 QM_SATURN 的备用实例位于系统 `coordination_host2` 上，并使用端口号 1420 进行客户机连接。

```
coordinationQMGr=QM_SATURN
coordinationQMGrHost=coordination_host1
coordinationQMGrPort=1420
coordinationQMGrChannel=SYSTEM.DEF.SVRCONN
coordinationQMGrStandby=coordination_host2(1420)
```

IBM MQ Managed File Transfer 独立的记录器必须始终以绑定方式连接到其队列管理器。在将独立的记录器与多实例协调队列管理器配合使用时，请以绑定方式将独立的记录器连接到另一队列管理器。第 168 页的 [『独立记录器的备用配置』](#) 中说明了您需要执行的步骤。必须使用多实例协调队列管理器的实例的主机名和端口号，定义独立的记录器队列管理器和协调队列管理器间的通道。有关如何完成此操作的信息，请参阅 [多实例队列管理器](#)。

IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件以客户机方式连接到协调队列管理器。如果多实例协调队列管理器的活动实例失败，那么协调队列管理器的备用实例将激活，并且该插件将重新进行连接。

IBM MQ Managed File Transfer 命令 **`fteList*`** 和 **`fteShowAgentDetails`** 直接连接到协调队列管理器。如果多实例协调的活动实例不可用，这些命令将尝试连接到协调队列管理器的备用实例。

将多实例队列管理器用作命令队列管理器

要同时连接到多实例命令队列管理器的活动实例和备用实例，请将 `connectionQMGrStandby` 属性添加到 IBM MQ Managed File Transfer 拓扑中的所有 `command.properties` 文件。

请参阅 [第 588 页的『command.properties 文件』](#)，以了解更多信息。

该示例显示了指定名为 QM_MARS 的多实例命令队列管理器的连接详细信息的 `command.properties` 文件的内容。QM_MARS 的活动实例位于系统 `command_host1` 上，并使用端口号 1424 进行客户机连接。QM_MARS 的备用实例位于系统 `command_host2` 上，并使用端口号 1424 进行客户机连接。

```
connectionQMgr=QM_SATURN
connectionQMgrHost=command_host1
connectionQMgrPort=1424
connectionQMgrChannel=SYSTEM.DEF.SVRCONN
connectionQMgrStandby=command_host2(1424)
```

相关概念

第 142 页的『[连接到 IBM MQ](#)』

与 IBM MQ 队列管理器的所有网络通信（包括与 IBM MQ Managed File Transfer 相关的通信），都涉及 IBM MQ 通道。IBM MQ 通道代表网络链路的一端。通道可以归类为消息通道或 MQI 通道。

相关任务

第 142 页的『[配置 IBM MQ 队列管理器](#)』

如果您的 IBM MQ Managed File Transfer 网络包含多个 IBM MQ 队列管理器，那么这些 IBM MQ 队列管理器相互之间必须能够进行远程通信。

第 144 页的『[配置协调队列管理器](#)』

运行 `fteSetupCoordination` 命令之后，运行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目录中的 `coordination_qmgr_name.mqsc` 脚本以对协调队列管理器执行必要配置。但是，如果您希望手动执行此配置，请在协调队列管理器上完成以下步骤。

确保保留 IBM MQ Managed File Transfer 日志消息

IBM MQ Managed File Transfer 将文件传输进度和日志信息发送到协调队列管理器。协调队列管理器将此信息发布到 SYSTEM.FTE 主题的任何匹配预订。如果无任何预订，将不保留此信息。

如果传输进度或日志信息对您的业务很重要，那么必须执行以下步骤之一以确保保留此信息：

- 使用 IBM MQ Managed File Transfer 数据库记录器将发布到 SYSTEM.FTE/Log 主题的消息复制到 Oracle 或 Db2 数据库。
- 定义 SYSTEM.FTE 主题预订（将发布存储在 IBM MQ 队列上）。在传输任何文件传输之前定义该预订，以确保所有进度和日志消息保留在队列上。
- 编写一个应用程序，使其使用消息队列接口 (MQI) 或 IBM MQ JMS 创建持久预订并处理交付到该预订的发布。该应用程序必须在传输任何文件之前处于运行状态，以确保应用程序接收所有进度和日志消息。

在接下来的部分中，更详细地描述了每种方法。

请勿依赖于 IBM MQ Explorer 插件来保留日志信息。

使用 IBM MQ Managed File Transfer 数据库记录器保留日志消息

数据库记录器是 IBM MQ Managed File Transfer 的可选组件，可使用它将日志信息复制到数据库中用于分析和审计目的。数据库记录器是您安装在托管协调队列管理器和数据库的系统上的独立 Java 应用程序。有关数据库记录器的更多信息，请参阅第 151 页的『[配置 Managed File Transfer 记录器](#)』。

使用 IBM MQ Explorer 插件保留进度和日志消息

首次启动 IBM MQ Explorer 插件的实例时，该实例将在协调队列管理器上创建持久预订。该持久预订用于收集“传输日志”和“当前传输进度”视图中显示的信息。持久预订名称的前缀为运行 IBM MQ Explorer 的关联实例的主机名。在管理员想要删除 IBM MQ Explorer 插件实例不再实际使用的持久预订时添加该前缀。

在协调队列管理器上使用持久预订可能会导致在 SYSTEM.MANAGED.DURABLE 队列上构建消息。如果您具有大容量 IBM MQ Managed File Transfer 网络，请不要频繁使用 IBM MQ Explorer 插件，否则此消息数据可能会填满本地文件系统。

要避免发生此情况，可以指定 IBM MQ Explorer 插件使用对协调队列管理器的非持久预订。在 IBM MQ Explorer 中执行以下步骤：

1. 选择窗口 > 首选项 > **IBM MQ Explorer** > 受管文件传输
2. 在传输日志订阅类型列表中，选择 NON_DURABLE。

将发布存储在 IBM MQ 队列上

要将日志或进度消息存储在 IBM MQ 队列上，请在将消息转发到该队列的协调队列管理器上配置预订。例如，要将所有日志消息转发到名为 LOG.QUEUE 的队列，请提交以下 MQSC 命令：

```
define sub(MY.SUB) TOPICSTR('Log/#') TOPICOBJ(SYSTEM.FTE) DEST(LOG.QUEUE) WSCHEMA(TOPIC)
```

将日志消息转发到 IBM MQ 队列后，这些消息将永久存储在该队列上，直至使用该队列的 IBM MQ 应用程序处理这些消息为止。

编写用于管理 SYSTEM.FTE 主题的持久预订的应用程序

您可以使用 IBM MQ 支持的某个应用程序编程接口，编写用于管理其自己的持久预订（针对 SYSTEM.FTE 主题）的应用程序。这些应用程序可以接收 IBM MQ 队列或日志消息，并根据您的业务需要相应地处理这些消息。

有关可用应用程序编程接口的更多信息，请参阅[开发应用程序](#)。

配置 Managed File Transfer 记录器

Managed File Transfer 传输文件时，它将有关其操作的信息发布到协调队列管理器上的主题。数据库记录器是 Managed File Transfer 的可选组件，您可以使用该组件将此信息复制到数据库以供分析和审计。

有三个版本的记录器：

- 独立文件记录器
- 独立数据库记录器
- Java Platform, Enterprise Edition (Java EE) 记录器

独立文件记录器仅在 V7.5 和更高版本中可用。

独立文件记录器

独立文件记录器是一个 Java 进程，它在托管协调队列管理器的系统上运行，或在托管与协调队列管理器连接的队列管理器的系统上运行。独立文件记录器使用 IBM MQ 绑定来连接到与其关联的队列管理器。此独立记录器使用 **fteCreateLogger** 命令来创建。

对于 V7.5 和更高版本，您可以将独立文件记录器作为 Windows 服务运行以确保文件记录器在您从 Windows 会话注销时继续运行，并且可以将其配置为在系统重新启动时自动启动。有关指示信息，请参阅第 153 页的『[安装和配置 IBM MQ Managed File Transfer 独立文件记录器](#)』。

在 z/OS 或 IBM i 上不支持独立文件记录器。

独立数据库记录器

独立数据库记录器是您安装在托管队列管理器和数据库的系统上的 Java 应用程序。独立数据库记录器通常安装在与协调队列管理器相同的系统上，但也可以安装在与连接到协调队列管理器的任何队列管理器相同的系统上。独立数据库记录器使用 IBM MQ 绑定来连接到与其关联的队列管理器，并使用 2 类或 4 类 JDBC 驱动程序连接到 Db2 或 Oracle 数据库。这些类型的连接是必需的，因为独立数据库记录器使用队列管理器的 XA 支持通过队列管理器和数据库协调全局事务，从而保护数据。

如果要使用 Windows 系统，您可以将独立记录器作为 Windows 服务运行以确保记录器在您从 Windows 会话注销时继续运行。有关指示信息，请参阅独立数据库记录器的第 160 页的『[安装 IBM MQ Managed File Transfer 独立数据库记录器](#)』。

Java EE 数据库记录器

Java EE 数据库记录器作为 EAR 文件提供，您可以将其安装到应用程序服务器中。如果您有现有 Java EE 应用程序服务器环境可用，那么这可能比使用独立数据库记录器更方便，因为可以将 Java EE 数据库记录器与其他企业应用程序一起管理。您还可以将 Java EE 数据库记录器安装在与托管 IBM MQ 服务器和数据库的系统不同的系统上。支持将 Java EE 数据库记录器用于 Db2 和 Oracle 数据库。安装在 WebSphere Application Server V7.0 上时，Java EE 数据库记录器还支持 Oracle Real Application Clusters。

有关如何配置记录器的指示信息，请参阅以下主题：

- [第 153 页的『安装和配置 IBM MQ Managed File Transfer 独立文件记录器』](#)
- [第 160 页的『安装 IBM MQ Managed File Transfer 独立数据库记录器』](#)
- [第 169 页的『安装 Java EE 数据库记录器』](#)

相关任务

[第 153 页的『安装和配置 IBM MQ Managed File Transfer 独立文件记录器』](#)

独立文件记录器是必须使用 IBM MQ 绑定连接到协调队列管理器的 Java 进程。要定义独立文件记录器，请使用 **fteCreateLogger** 命令并遵循本主题中的步骤。

[第 160 页的『安装 IBM MQ Managed File Transfer 独立数据库记录器』](#)

完成以下步骤以安装和配置独立数据库记录器。

[第 169 页的『安装 Java EE 数据库记录器』](#)

按照以下指示信息安装并配置 Java EE 数据库记录器以供 Managed File Transfer 使用。

[第 179 页的『从独立数据库记录器迁移至 Java EE 数据库记录器』](#)

您可以从独立数据库记录器迁移到 Java EE 数据库记录器。必须停止独立数据库记录器，并安装 JEE 数据库记录器。为避免日志条目丢失或重复，您必须先停止正在向 SYSTEM.FTE 主题发布的消息，然后再停止独立数据库记录器，并在安装 Java EE 数据库记录器之后重新启动独立数据库记录器。备份您的数据库，然后再执行迁移。生成的新信息。

[第 34 页的『将 Db2 on z/OS 上的数据库表迁移到 V8.0.0』](#)

如果您的数据库是 Db2 on z/OS 系统，那么必须完成以下步骤，以在不同版本的 WebSphere MQ File Transfer Edition 之间进行迁移（即从 V7.0.3 迁移至 V7.0.4，以及从 WebSphere MQ File Transfer Edition V7.0.4 迁移至 IBM MQ Managed File Transfer V8.0.0）。Db2 表的结构与前发行版不同。例如，某些表中增加了新列，某些可变字符列可能更大，因此必须将前发行版中的表迁移至 V8.0 格式。

[第 32 页的『在 Windows，UNIX 或 Linux 上的 Db2 上增加日志数据库的页面大小』](#)

如果您的数据库在 Windows，UNIX 或 Linux 系统上是 Db2，并且您创建的日志数据库的页面大小小于 8 KB，那么必须在迁移到 V7.0.3 或更高版本的表之前增大数据库的页面大小。

[第 161 页的『使用远程数据库』](#)

您可以使用 IBM MQ Managed File Transfer 记录器与远程系统上的数据库通信。

相关参考

[第 402 页的『记录器错误处理和拒绝』](#)

记录器确定两种类型的错误：每条消息的错误和通用错误。

[第 168 页的『独立记录器的备用配置』](#)

通常，不管独立记录器是文件还是数据库类型，都与协调队列管理器位于同一个系统上，并以 IBM MQ 绑定方式连接到协调队列管理器。但是，它也可以安装在与连接到协调队列管理器的任何队列管理器相同的系统上。独立记录器使用其自动创建的预订接收消息。这是在安装指示信息中描述的配置。

[第 163 页的『IBM MQ Managed File Transfer 的记录器配置属性』](#)

该记录器具有一组配置属性。在 `logger.properties` 文件中指定这些属性，该文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中。

[第 474 页的『fteCreateLogger（创建 IBM MQ Managed File Transfer 记录器）』](#)

使用 **fteCreateLogger** 命令可创建文件或数据库记录器。

[第 575 页的『fteStartLogger（启动记录器）』](#)

fteStartLogger 命令启动 IBM MQ Managed File Transfer 日志记录应用程序。

[第 548 页的『fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）』](#)

使用 **fteModifyLogger** 命令修改记录器，以便记录器能够作为 Windows 服务运行。只能在 Windows 平台上使用此命令，必须由作为 IBM MQ 管理员和 mqm 组成员的用户运行，并且必须首先使用 **fteStopLogger** 命令停止记录器。

第 579 页的『[fteStopLogger（停止记录器）](#)』

fteStopLogger 命令停止记录器。

第 525 页的『[fteDeleteLogger（删除 IBM MQ Managed File Transfer 记录器）](#)』

使用 **fteDeleteLogger** 命令可删除 IBM MQ Managed File Transfer 记录器及其配置。可以保留或删除与该记录器关联的现有日志文件。

第 731 页的『[记录器所使用的数据库表](#)』

在安装并配置记录器后，会创建以下数据库表：

第 441 页的『[记录器的权限](#)』

运行记录器的操作系统用户需要具有对记录器队列和 SYSTEM.FTE 主题的特定 IBM MQ 权限。

安装和配置 IBM MQ Managed File Transfer 独立文件记录器

独立文件记录器是必须使用 IBM MQ 绑定连接到协调队列管理器的 Java 进程。要定义独立文件记录器，请使用 **fteCreateLogger** 命令并遵循本主题中的步骤。

关于此任务

有关独立文件记录器的更多信息，请参阅第 151 页的『[配置 Managed File Transfer 记录器](#)』。本主题中的步骤将记录器配置为连接到协调队列管理器。有关备用记录器配置，请参阅第 168 页的『[独立记录器的备用配置](#)』。

在 z/OS 或 IBM i 上不支持独立文件记录器。

过程

1. 请确保您已安装 IBM MQ Managed File Transfer 记录器组件。有关更多信息，请参阅第 9 页的『[IBM MQ Managed File Transfer 产品选项](#)』。
2. 指定协调队列管理器并将参数 **-loggerType** 设置为 FILE，然后运行 **fteCreateLogger** 命令以创建独立文件记录器。有关更多信息，请参阅第 474 页的『[fteCreateLogger（创建 IBM MQ Managed File Transfer 记录器）](#)』。
3. 可选：如果要使用定制格式，那么可以修改由 **fteCreateLogger** 命令创建的 XML 文件。日志格式定义位于 `FileLoggerFormat.xml` 文件中。有关更多信息，请参阅第 154 页的『[独立文件记录器格式](#)』。
4. 针对协调队列管理器运行 **fteCreateLogger** 命令提供的 MQSC 命令，以创建记录器队列。
5. 确定要运行该记录器进程的用户，并为该用户配置许可权。有关更多信息，请参阅第 159 页的『[为独立文件记录器配置用户访问权](#)』。
6. 可选：您可以通过编辑在运行 **fteCreateLogger** 命令时创建的 `logger.properties` 文件来进一步配置独立文件记录器。此文件是由键/值对组成的 Java 属性文件。`logger.properties` 文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中。有关可用属性及其效果的更多信息，请参阅第 163 页的『[IBM MQ Managed File Transfer 的记录器配置属性](#)』。
7. 可选：如果使用 Windows 系统，那么可以将独立文件记录器作为 Windows 服务运行。运行带 **-s** 参数的 **fteModifyLogger** 命令。有关更多信息，请参阅第 548 页的『[fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）](#)』。
8. 使用 **fteStartLogger** 命令启动独立文件记录器。有关更多信息，请参阅第 575 页的『[fteStartLogger（启动记录器）](#)』。

如果执行了上一步并在 Windows 上使用了带有 **-s** 参数的 **fteModifyLogger** 命令，那么独立文件记录器将作为 Windows 服务启动。

9. 检查记录器输出。独立文件记录器生成两种类型的输出，即文件传输审计数据和记录器诊断数据。文件传输审计数据可在 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs` 中找到。记录器诊断数据可在 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name` 中找到。

10. 您可以通过使用 **fteStopLogger** 命令来停止该记录器。有关更多信息，请参阅 [第 579 页的『fteStopLogger（停止记录器）』](#)。

结果

相关任务

第 159 页的『为独立文件记录器配置用户访问权』

在测试环境中，您可以向常规用户帐户添加所需的任何新特权。在生产环境中，建议您创建具有执行作业所需的最低许可权的新用户。

相关参考

第 163 页的『IBM MQ Managed File Transfer 的记录器配置属性』

该记录器具有一组配置属性。在 `logger.properties` 文件中指定这些属性，该文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中。

第 575 页的『fteStartLogger（启动记录器）』

fteStartLogger 命令启动 IBM MQ Managed File Transfer 日志记录应用程序。

第 474 页的『fteCreateLogger（创建 IBM MQ Managed File Transfer 记录器）』

使用 **fteCreateLogger** 命令可创建文件或数据库记录器。

第 548 页的『fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）』

使用 **fteModifyLogger** 命令修改记录器，以便记录器能够作为 Windows 服务运行。只能在 Windows 平台上使用此命令，必须由作为 IBM MQ 管理员和 mqm 组成员的用户运行，并且必须首先使用

fteStopLogger 命令停止记录器。

第 579 页的『fteStopLogger（停止记录器）』

fteStopLogger 命令停止记录器。

第 154 页的『独立文件记录器格式』

文件记录器写入的消息信息的格式可在 `FileLoggerFormat.xml` 文件中定义。

第 441 页的『记录器的权限』

运行记录器的操作系统用户需要具有对记录器队列和 `SYSTEM.FTE` 主题的特定 IBM MQ 权限。

独立文件记录器格式

文件记录器写入的消息信息的格式可在 `FileLoggerFormat.xml` 文件中定义。

记录器的配置目录位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 中。创建新的文件记录器时，会创建该文件的一个版本，其中包含文件记录器使用的一组缺省定义。当设计您自己的日志格式定义时，可从该缺省文件入手。有关缺省日志格式定义的更多信息，请参阅第 639 页的『独立文件记录器缺省日志格式定义』。

定制的日志格式定义

日志格式定义由一组消息类型组成，每种消息类型都具有格式定义。消息类型的格式定义由一组以 XPATH 格式提供的插入和用于分隔每个插入的分隔符组成。对插入的排序决定了以何种顺序来将内容放置在为输出到日志文件而生成的行中。例如，这是 `callStarted` 消息类型的定义：

```
<callStarted>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/
        @time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/agent/
        @agent</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/agent/@QMGr</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/job/name</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/
        call/command/@type</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/
        call/command/@name</insert>
      <insert type="system" width="0" ignoreNull="true">callArguments</insert>
    </inserts>
    <separator></separator>
```

```
</format>
</callStarted>
```

该格式在日志文件中生成类似以下的行：

```
2011-11-25T10:53:04;414d5120514d5f67627468696e6b20206466cf4e20004f02;[CSTR];
AGENT1;AGENT_QM;Managed Call;executable;echo;call test;
```

在格式定义中提供的 insert 顺序是信息在日志文件行中出现的顺序。有关定义 FileLoggerFormat.xml 文件格式的 XML 模式的更多信息，请参阅第 644 页的『独立文件记录器格式 XSD』。

消息类型

FTE 代理将一系列不同的消息类型写入到 SYSTEM.FTE/Log 子主题。有关更多信息，请参阅第 646 页的『SYSTEM.FTE 主题』。日志文件定义可包含以下类型的消息的格式定义：

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
scheduleSubmitTransferSet
transferStarted
transferCancelled
transferComplete
transferDelete
transferProgress
```

消息格式会有所不同。大多数的消息类型都会在日志文件中为 SYSTEM.FTE/Log 子主题中耗用的每条日志消息写入单个行。这导致了简单的情况，即日志格式定义中提供的 XPATH 地址与消息根相关。以下是使用该方法来写入输出的消息类型：

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
transferStarted
transferCancelled
transferComplete
transferDelete
```

用于写入日志消息的另一种方法则使用多行来表示日志消息内传输集中的项。在此情况下，所提供的格式会应用于日志消息内传输集中的每个项。如果希望包含特定于传输集中每个项的信息，那么需要所提供的 XPATH 来将该项用作其 XPATH 根。以下是使用该方法来写入输出的消息类型：

```
scheduleSubmitTransferSet
transferProgress
```

将为传输集中的每个项都写入一行输出。要对于传输集中所有项固定的信息仍可以使用与日志消息的根相关的 XPATH 地址。在以下简化的 transferProgress 格式定义示例中，固定的是时间戳记和传输标识。

与作为其根的项相关的任何信息对于写入的每一行均会有变化。在此示例中，写入了每个项的源和目标文件信息。

```
<transferProgress>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/
        @time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="true">status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">source/file |
        source/queue</insert>
      <insert type="user" width="0" ignoreNull="false">source/file/@size |
        source/queue/@size</insert>
      <insert type="user" width="5" ignoreNull="true">source/@type</insert>
      <insert type="user" width="6" ignoreNull="true">source/@disposition</insert>
      <insert type="user" width="0" ignoreNull="false">destination/file |
        destination/queue</insert>
      <insert type="user" width="0" ignoreNull="false">destination/file/@size |
        destination/queue/@size</insert>
      <insert type="user" width="5" ignoreNull="true">destination/@type</insert>
      <insert type="user" width="9" ignoreNull="true">destination/@exist</insert>
      <insert type="user" width="0" ignoreNull="true">status/supplement</insert>
    </inserts>
    <separator></separator>
  </format>
</transferProgress>
```

这样会以下列格式生成具有一行或多行的日志文件条目：

```
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPRO];0
;/src/test1.file;3575;file;leave ;/dest/test1.file;3575;file;overwrite;;
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPRO];0
;/src/test2.file;3575;file;leave ;/dest/test2.file;3575;file;overwrite;;
```

插入格式

定义消息类型的格式时，有两种类型的插入可用：**user** 和 **system**。插入类型在插入元素的 **type** 属性中定义。这两种插入类型还可以使用插入元素的 **width** 和 **ignoreNull** 属性来定制其布局。例如：

```
<insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
```

在该示例中，插入操作会提取从位于 `/transaction/@ID` 的日志消息中找到的信息，并将该信息削减或填充为 48 个字符，然后将其写入日志。如果 `/transaction/@ID` 的内容为 `null`，那么插入操作会在将字符串 `null` 填充为 48 个字符后予以写入，原因是 `ignoreNull` 属性已设置为 `false`。如果 `ignoreNull` 设置为 `true`，那么会改为写入填充为 48 个字符的空字符串。设置 `width="0"` 表示不裁剪列宽，不表示裁剪宽度为 0。可以通过此方式使用 `ignoreNull` 属性，以在日志中发现非预期的空值时进行检测。当调试新的日志文件定义时，这很有用。

用户定义的插入

用户插入包含要在该插入中写入的信息的 XPATH 地址。该地址引用在 FTE 日志消息中找到的一段信息。有关日志消息格式的更多信息，请参阅：

- 第 662 页的『文件传输日志消息格式』
- 第 686 页的『已安排的传输日志消息格式』
- 第 691 页的『监视器日志消息格式』

系统定义的插入

系统定义的插入包含某个关键字，该关键字引用无法在日志消息中找到或者不便于使用 XPATH 语言进行定义的一段信息。

受支持的系统插入有：

- `type` - 以短格式写日志消息的类型。
- `callArguments` - 以空格分隔的格式写一组提供给受管调用的自变量。
- `transferMetaData` - Writes the set of metadata entries defined for a transfer in a comma separated <key>=<value> format.

下表列出了每种消息类型的系统定义插入的“type”值。

表 18: 受支持消息类型及其“type”系统插入的摘要。	
消息类型	“type”系统插入的值
<code>callCompleted</code>	[CCOM]
<code>callStarted</code>	[CSTR]
<code>monitorAction</code>	[MACT]
<code>monitorCreate</code>	[MCRT]
<code>monitorFired</code>	[MFIR]
<code>notAuthorized</code>	[AUTH]
<code>scheduleDelete</code>	[SDEL]
<code>scheduleExpire</code>	[SEXP]
<code>scheduleSkipped</code>	[SSKP]
<code>scheduleSubmitInfo</code>	[SSIN]
<code>scheduleSubmitTransfer</code>	[SSTR]
<code>scheduleSubmitTransferSet</code>	[SSTS]
<code>transferStarted</code>	[TSTR]
<code>transferCancelled</code>	[TCAN]
<code>transferComplete</code>	[TCOM]
<code>transferDelete</code>	[TDEL]
<code>transferProgress</code>	[TPRO]

相关概念

第 646 页的『[SYSTEM.FTE 主题](#)』

SYSTEM.FTE 主题是关于 IBM MQ Managed File Transfer 用于记录传输并存储有关代理、监视器、调度和模板的协调队列管理器的主题。

相关参考

第 639 页的『[独立文件记录器缺省日志格式定义](#)』

独立文件记录器的缺省日志文件格式定义

第 644 页的『[独立文件记录器格式 XSD](#)』

独立文件格式的模式。

第 662 页的『[文件传输日志消息格式](#)』

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 `IBM MQ Managed File Transfer` 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

第 686 页的『已安排的传输日志消息格式』

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 `SYSTEM.FTE/Log/agent_name/schedule ID` 主题中）。该消息符合 `ScheduleLog.xsd` XML 模式。

第 691 页的『监视器日志消息格式』

监视器日志消息发布至 `SYSTEM.FTE` 主题（含主题字符串 `Log/agent_name/Monitors/monitor_name/monitor_ID`）。

从独立文件记录器中排除消息类型

如果希望从文件记录器输出中排除特定消息类型，那么可以使用空消息类型元素。

关于此任务

示例

例如，以下格式定义阻止了文件记录器输出 `transferProgress` 消息。

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
  xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferProgress></transferProgress>
  </messageTypes>
</logFormatDefinition>
```

为独立文件记录器定义一组有限的定制格式

可以在日志格式定义内对定制消息类型子集进行定义以减少定制日志文件格式所需的配置量。

关于此任务

如果 `messageTypes` 元素未包含在 `FileLoggerFormat.xml` 文件中，那么该消息类型的格式使用缺省格式。您只需指定希望使之与缺省格式不同的格式。

示例

在该示例中，格式定义使用此仅输出已启动传输的用户的简化版本来替换 `transferStarted` 消息类型的缺省格式。所有其他消息类型都使用缺省格式，因为它们未包含在该日志格式定义中：

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
  xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferStarted>
      <format>
        <inserts>
          <insert type="user" width="19" ignoreNull="false">/transaction/action/
            @time</insert>
          <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6" ignoreNull="false">type</insert>
          <insert type="user" width="0" ignoreNull="true">/transaction/originator/
            userID</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </transferStarted>
  </messageTypes>
</logFormatDefinition>
```

相关参考

第 639 页的『独立文件记录器缺省日志格式定义』

独立文件记录器的缺省日志文件格式定义

第 644 页的『独立文件记录器格式 XSD』
独立文件格式的模式。

减少独立文件记录器中的重复消息

重复的日志消息可能会出现在独立文件记录器的日志中。通过使用 `logger.properties` 文件，您可以对独立文件记录器进行调整并减少重复消息的数量。

文件记录器日志中的重复消息

在发生故障的情况下，日志消息可能会写入独立文件记录器的日志中，而不会耗用专用于 WebSphere® MQ 的 `SYSTEM.FTE/Log#` 主题中的日志消息。如果发生此情况，那么当独立文件记录器重新启动时，它将再次检索同一消息，并将该消息再次写入日志文件。请计划在手动查看日志文件或者自动处理日志文件时应对出现这些重复消息的可能性。为了帮助检测重复消息，独立文件记录器在启动时会将会以下消息输出到日志文件：

```
BFGDB0054I: The file logger has successfully started
```

重复消息始终在独立文件记录器启动的时间左右出现，因为这是处理先前实例失败之前的最后一条消息的时间。通过了解新实例已在何时启动，您可以检测出是否应预期会出现重复消息，以及是否需要处理重复消息。

减少重复消息的数量

独立文件记录器会将其处理的日志消息分组到事务中以提高性能。此批次大小是您在发生故障的情况下可能看到的重复消息的最大数目。要减少重复消息的数量，您可以在 `logger.properties` 文件中对以下属性进行调整：

```
wmqfte.max.transaction.messages
```

例如，通过将此项设置为 1，复制的最大消息数将减少到 1。请注意，修改此值会影响独立文件记录器的性能，因此需要进行彻底测试以确保不会对系统造成负面影响。

`logger.properties` 文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中。有关可用属性及其影响的更多信息，请参阅第 163 页的『IBM MQ Managed File Transfer 的记录器配置属性』。

为独立文件记录器配置用户访问权

在测试环境中，您可以向常规用户帐户添加所需的任何新特权。在生产环境中，建议您创建具有执行作业所需的最低许可权的新用户。

关于此任务

您必须在同一个系统上安装独立文件记录器和 IBM MQ。请按照以下方式配置该用户的许可权：

过程

1. 确保用户具有对作为 IBM MQ Managed File Transfer 安装一部分所安装的文件进行读和（必要时）执行的许可权。
2. 确保该用户有权对 `logs` 目录（位于配置目录中）中的任何文件进行创建和写入。该目录用于事件日志，如有必要，还用于诊断跟踪和“首次故障数据捕获”(FFDC) 文件。
3. 确保该用户具有其自己的组，并且还不在于协调队列管理器具有广泛许可权的任何组中。该用户不应在 `mqm` 组中。在某些平台上，还会自动为 `staff` 组授予队列管理器访问权；独立文件记录器用户不应位于 `staff` 组中。您可以使用 IBM MQ Explorer 来查看队列管理器本身及其包含的对象的权限记录。右键单击该对象，然后选择**对象权限 > 管理权限记录**。在命令行上，可以使用命令 `dspsmqaut`（显示权限）或 `dmpmqaut`（转储权限）。
4. 使用 IBM MQ Explorer 中的“管理权限记录”窗口或 `setmqaut`（授予或撤销权限）命令，为用户自己的组添加权限（在 UNIX 上，IBM MQ 权限只与组关联，而与单个用户无关）。所需权限如下所示：

- 在队列管理器上连接和查询 (IBM MQ Java 库需要查询许可权才能操作)。
- 对 SYSTEM.FTE 主题的 SUBSCRIBE 许可权。
- 对 SYSTEM.FTE.LOG.RJCT.*logger_name* 队列的 PUT 许可权。
- 对 SYSTEM.FTE.LOG.CMD.*logger_name* 队列的 Get 许可权。

给定的 REJECT 和 COMMAND 队列名称是缺省名称。如果您在配置独立文件记录器队列时选择了不同的队列名称, 请改为向这些队列名称添加许可权。

安装 IBM MQ Managed File Transfer 独立数据库记录器

完成以下步骤以安装和配置独立数据库记录器。

关于此任务

有关独立数据库记录器的更多信息, 请参阅第 151 页的『配置 Managed File Transfer 记录器』。

注: 在任何同一时间, 不能针对数据库中的同一模式运行多个数据库记录器 (无论独立还是 JEE 类型)。如果尝试执行此操作, 那么在尝试向数据库写入传输日志数据时将导致冲突。

过程

1. 使用数据库的文档指导您安装数据库软件。

如果 JDBC 支持是您数据库的可选组件, 那么必须安装此组件。

2. 将参数 **-loggerType** 设置为 DATABASE, 然后运行 **fteCreateLogger** 命令以创建独立数据库记录器。要了解更多信息, 请参阅第 474 页的『fteCreateLogger (创建 IBM MQ Managed File Transfer 记录器)』。

缺省模式名称为 FTELOG。如果使用 FTELOG 之外的模式名称, 必须先编辑提供的 SQL 文件 (根据您的数据库, 为 `ftelog_tables_db2.sql` 或 `ftelog_tables_oracle.sql`) 以反映此模式名称, 然后继续下一步骤。有关更多信息, 请参阅第 163 页的『IBM MQ Managed File Transfer 的记录器配置属性』中的 `wmqfte.database.schema`。

3. 使用您数据库的工具创建所需的数据库表。

在分布式平台上, 文件 `ftelog_tables_db2.sql` 和 `ftelog_tables_oracle.sql` 包含可运行以创建表的 SQL 命令。

在 z/OS 上, 需要运行的文件取决于使用的 Db2 for z/OS 版本:

- 对于 Db2 for z/OS 9.0 以及更低版本, 运行文件 `ftelog_tables_zos.sql` 以创建表。此文件在创建表时, 将 INTEGER 数据类型用于表示所传输文件大小的字段以及表示与每个传输关联的表标识的字段。
- 对于 Db2 for z/OS 9.1 以及更高版本, 运行文件 `ftelog_tables_zos_bigint.sql` 以创建表。此文件在创建表时, 将 BIGINT 数据类型用于表示所传输文件大小的字段以及表示与每个传输关联的表标识的字段。

4. 针对您的记录器命令队列管理器运行 **fteCreateLogger** 命令所提供的 MQSC 命令以创建记录器队列。独立数据库记录器使用协调队列管理器上的两个队列。第一个队列为命令队列, 其中将放置用于控制独立数据库记录器的操作的消息。此命令队列的缺省名称是 `SYSTEM.FTE.LOG.CMD.logger_name`。第二个队列为拒绝队列。由于独立数据库记录器从不废弃日志消息, 因此如果记录器遇到无法处理的消息, 它会将该消息放置到拒绝队列上以进行检查, 并可能进行重新处理。建议您不要将队列管理器的死信队列用于此目的, 因为拒绝的消息没有 DLH 头, 并且拒绝的消息不应与由于其他原因放置到死信队列中的消息组合在一起。拒绝队列的缺省名称是 `SYSTEM.FTE.LOG.RJCT.logger_name`。这两个队列在 **fteCreateLogger** 命令生成的 MQSC 脚本文件中定义。

5. 选择用户并配置许可权

6. 可选: 您可以通过编辑步骤第 160 页的『2』中 **fteCreateLogger** 命令创建的 `logger.properties` 文件来进一步配置独立数据库记录器。此文件是由键/值对组成的 Java 属性文件。`logger.properties` 文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中。有关可用属性及其效果的更多信息, 请参阅第 163 页的『IBM MQ Managed File Transfer 的记录器配置属性』。

7. 可选：如果要使用 Windows 系统，您可以将独立数据库记录器作为 Windows 服务运行。运行带有 **-s** 参数的 **fteModifyLogger** 命令。有关更多信息，请参阅第 548 页的『[fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）](#)』。
8. 可选：如果使用的数据库是 Oracle，或者是远程连接到 Db2 数据库，那么将需要指定记录器将用于向数据库服务器认证的用户名和密码。此用户名和密码在符合 MQMFTCredentials.xsd 模式所定义的格式的凭证文件中指定。有关更多信息，请参阅第 873 页的『[MQMFT 凭证文件格式](#)』。创建凭证文件后，必须使用 **wmqfte.database.credentials.file** 属性在 **logger.properties** 文件中指定该凭证文件的位置。
9. 使用 **fteStartLogger** 命令启动独立数据库记录器。缺省情况下，独立数据库记录器在后台运行，并且独立数据库记录器会将输出放置到 **logs** 目录中的文件。如果要在前台运行独立数据库记录器，并将输出生成到控制台以及日志文件，请将 **-F** 参数添加到 **fteStartLogger** 命令。

如果执行了上一步并在 Windows 上使用了带有 **-s** 参数的 **fteModifyLogger** 命令，那么独立数据库记录器将作为 Windows 服务启动。

相关任务

第 162 页的『[为独立数据库记录器配置用户访问权](#)』

在测试环境中，您可以向常规用户帐户添加所需的任何新特权。在生产环境中，建议您创建具有执行作业所需的最低许可权的新用户。

相关参考

第 163 页的『[IBM MQ Managed File Transfer 的记录器配置属性](#)』

该记录器具有一组配置属性。在 **logger.properties** 文件中指定这些属性，该文件位于 **MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name** 目录中。

第 575 页的『[fteStartLogger（启动记录器）](#)』

fteStartLogger 命令启动 IBM MQ Managed File Transfer 日志记录应用程序。

第 548 页的『[fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）](#)』

使用 **fteModifyLogger** 命令修改记录器，以便记录器能够作为 Windows 服务运行。只能在 Windows 平台上使用此命令，必须由作为 IBM MQ 管理员和 mqm 组成员的用户运行，并且必须首先使用

fteStopLogger 命令停止记录器。

第 441 页的『[记录器的权限](#)』

运行记录器的操作系统用户需要具有对记录器队列和 SYSTEM.FTE 主题的特定 IBM MQ 权限。

使用远程数据库

您可以使用 IBM MQ Managed File Transfer 记录器与远程系统上的数据库通信。

关于此任务

如果数据库与 IBM MQ Managed File Transfer 安装在不同机器上，请完成以下步骤。这些步骤对于 Db2 和 Oracle 均适用，除非另有声明。

过程

1. 将数据库客户机安装在安装有 IBM MQ Managed File Transfer 的系统上。
2. 将远程数据库服务器添加到本地数据库客户机配置中。需要更新该配置，IBM MQ Managed File Transfer 和 WebSphere MQ 才能正确访问数据库。
3. 在 **logger.properties** 文件中指定新属性，以连接到使用凭证文件 **wmqfte.database.credentials.file** 的数据库。

注：较早版本的 IBM MQ Managed File Transfer 使用了属性 **wmqfte.oracle.user** 或 **wmqfte.database.user** 以及 **wmqfte.oracle.password** 或 **wmqfte.database.password**。现在不推荐使用这些属性。推荐使用 **wmqfte.database.credentials.file**。

4. 仅限 Oracle：为了能够远程连接数据库，可将协调队列管理器的 **qm.ini** 文件中的 XAResourceManager 节更改为以下内容（确保更改数据库名称、用户名和用户密码以匹配您自己的信息）：

Oracle_XA+Acc=P/ftelog/
qgw783jhT+SesTm=35+DB=FTEAUDIT1+**SqlNet=FTEAUDIT1**+threads=false, 更改将以粗体突出显示。

5. 仅限 **Oracle**: 在 `logger.properties` 文件中, 使用 **`wmqfte.oracle.host`** 和 **`wmqfte.oracle.port`** 属性指定主机和端口。主机和端口的缺省值允许您使用本地数据库客户机, 因此如果先前使用了本地数据库, 那么您可能没有设置这些值。

相关参考

第 163 页的『[IBM MQ Managed File Transfer 的记录器配置属性](#)』

该记录器具有一组配置属性。在 `logger.properties` 文件中指定这些属性, 该文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中。

为独立数据库记录器配置用户访问权

在测试环境中, 您可以向常规用户帐户添加所需的任何新特权。在生产环境中, 建议您创建具有执行作业所需的最低许可权的新用户。

关于此任务

运行独立数据库记录器所需的用户帐户的数量和类型取决于所用的系统数。您可以在单个系统上或跨两个系统安装独立数据库记录器、IBM MQ 和数据库。独立数据库记录器必须和 IBM MQ 位于同一个系统上。可以在以下拓扑中安装这些组件:

独立数据库记录器、IBM MQ 和数据库全都位于同一个系统上

您可以定义单个操作系统用户, 以用于所有这三个组件。这是适合独立数据库记录器的配置。独立数据库记录器使用绑定方式连接到 IBM MQ, 并使用本机连接来连接到数据库。

独立数据库记录器和 IBM MQ 位于一个系统上, 而数据库位于另一个系统上

您为该配置创建两个用户: 运行独立数据库记录器的系统上的一个操作系统用户, 以及对数据库服务器上的数据库具有远程访问权的一个操作系统用户。这是适合使用远程数据库的独立数据库记录器的配置。独立数据库记录器使用绑定方式连接到 IBM MQ, 并使用客户机连接来访问数据库。

例如, 这些指示信息的其余部分假定用户名为 `ftelog`, 但您也可以使用任何用户名。请按照以下方式配置该用户的许可权:

过程

1. 确保该用户有权读取并在必要时可以执行“IBM MQ Managed File Transfer 远程工具和文档”安装过程中所安装的文件。
2. 确保该用户有权创建和写入 `logs` 目录 (位于配置目录中) 中的任何文件。该目录用于事件日志, 如有必要, 还可用于诊断跟踪和 FFDC 文件。
3. 确保该用户具有自己的组, 并且未同时处于对协调队列管理器具有广泛许可权的任何组中。该用户不应在 `mqm` 组中。在某些平台上, 还会自动为 `staff` 组授予队列管理器访问权; 独立数据库记录器用户不应位于 `staff` 组中。您可以使用 WebSphere MQ Explorer 查看队列管理器本身以及其中对象的权限记录。右键单击该对象, 然后选择**对象权限 > 管理权限记录**。在命令行上, 可以使用命令 `dspmqaout` (显示权限) 或 `dmpmqaut` (转储权限)。
4. 使用 IBM MQ Explorer 中的“**管理权限记录**”窗口或 `setmqaut` (授予或撤销权限) 命令, 为用户自己的组添加权限 (在 UNIX 上, IBM MQ 权限只与组关联, 而与单个用户无关)。所需权限如下所示:
 - 在队列管理器上连接和查询 (IBM MQ Java 库需要查询许可权才能操作)。
 - 对 `SYSTEM.FTE` 主题的 `SUBSCRIBE` 许可权。
 - 对 `SYSTEM.FTE.LOG.RJCT.logger_name` 队列的 `PUT` 许可权。
 - 对 `SYSTEM.FTE.LOG.RJCT.logger_name` 队列的 `GET` 许可权。

给定的 `REJECT` 和 `COMMAND` 队列名称是缺省名称。如果您在配置独立数据库记录器队列时选择了不同的队列名称, 请改为向这些队列名称添加许可权。

5. 执行特定于所用数据库的用户配置。
 - 如果您的数据库是 Db2, 请执行以下步骤:

有几种机制可用于管理 Db2 的数据库用户。这些指示信息适用于基于操作系统用户的缺省方案。

- 确保 `ftelog` 用户未处于任何 Db2 管理组（例如，`db2iadm1`、`db2fadm1` 或 `dasadm1`）中
- 为用户提供连接到数据库的许可权，以及对[步骤 2：创建所需的数据库表](#)过程中所创建表的选择、插入和更新许可权
- 如果您的数据库是 Oracle，请执行以下步骤：
 - 确保 `ftelog` 用户不在任何 Oracle 管理组中（例如，Windows 上的 `ora_dba` 或 Unix 上的 `dba`）
 - 为用户提供连接到数据库的许可权，以及对[步骤 2：创建所需的数据库表](#)过程中所创建表的选择、插入和更新许可权

IBM MQ Managed File Transfer 的记录器配置属性

该记录器具有一组配置属性。在 `logger.properties` 文件中指定这些属性，该文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中。

对于 IBM WebSphere MQ V7.5 或更高版本，可以在某些表示文件或目录位置的 Managed File Transfer 属性中使用环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化，例如哪个用户在运行进程。有关更多信息，请参阅第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』。

注：在 Windows 上指定文件路径时，反斜杠 (\) 分隔符必须以双反斜杠 (\\)（即，转义的 \）的形式出现。或者，您也可以将单个正斜杠字符 (/) 用作分隔符。有关 Oracle 中 Java 属性文件中的字符转义的更多信息，请参阅 [属性类的 Javadoc](#)。

属性名	描述	缺省值
<code>wmqfte.logger.type</code>	正在使用的记录器类型：文件或数据库。将该值设置为 <code>FILE</code> 或 <code>DATABASE</code> 。	无缺省值
<code>wmqfte.max.transaction.messages</code>	落实事务之前在该事务中处理的消息的最大数目。在循环日志记录方式下，队列管理器可用于实时数据的空间容量是固定的。确保您将该属性设置为足够低的值，以便可用空间不会耗尽。	50
<code>wmqfte.max.transaction.time</code>	前后两次事务落实之间所经过的最大时间长度（毫秒）。	5000
<code>wmqfte.max.consecutive.reject</code>	可连续拒绝的最大消息数（即未遇到有效消息）。 如果超过该数量，那么记录器将认定问题不在于消息自身，而在于配置。例如，如果您使数据库中某个代理名称列的宽度小于所有代理名称，那么引用代理的所有消息都将被拒绝。	50
<code>wmqfte.reject.queue.name</code>	记录器将自己无法处理的消息放置到的队列的名称。如果您拥有数据库记录器，那么请参阅数据库记录器错误处理和拒绝以获取关于哪些消息可能会被置于此队列上的详细信息。	<code>SYSTEM.FTE.LOG.RJCT.logger_name</code>
<code>wmqfte.command.queue.name</code>	记录器从中读取控制其行为的命令消息的队列的名称。	<code>SYSTEM.FTE.LOG.CMD.logger_name</code>
<code>wmqfte.queue.manager</code>	记录器连接到的队列管理器（该队列管理器必须位于与记录器相同的机器上）。	无缺省值

属性名	描述	缺省值
wmqfte.message.source.type	以下某个值： automatic subscription 缺省值。记录器在 SYSTEM.FTE/Log/# 中定义的队列管理器上创建并使用其自己的持久的受管预订。这是适合大多数场景的值。 administrative subscription 如果自动预订不合适，那么您可以定义其他预订（例如，通过使用 IBM MQ Explorer、MQSC 或 PCF），并指示记录器使用该预订。例如，使用该值将日志空间分区，以便一个记录器处理代理 A 到 H，另一个记录器处理 I 到 P，第三个记录器处理 Q 到 Z。 queue 如果 IBM MQ 拓扑表示不方便创建记录器的预订，那么您可以改为使用队列。配置 IBM MQ，使该队列接收通常由协调队列管理器上的 SYSTEM.FTE/Log/# 预订接收的消息。	automatic subscription
wmqfte.message.source.name	如果消息源类型是 管理预订 或 队列 ，那么使用预订或队列的名称。如果源类型是 自动预订 ，那么忽略该属性。	无缺省值
wmqfte.database.credentials.file	包含用于连接至数据库的用户名和密码的文件。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。 有关更多信息，请参阅第 873 页的『MQMFT 凭证文件格式』。	此属性的缺省值为 %HOMEDRIVE% %HOMEPATH%\mqmftcredentials.xml" (在 Windows 上) 和 \$HOME/MQMFTCredentials.xml (在其他平台上)。
wmqfte.database.driver	数据库的 JDBC 驱动程序类的位置。这通常是 JAR 文件的路径和文件名。例如，Db2 on AIX 系统的 2 类驱动程序需要 /opt/IBM/db2/V9.5/java/db2jcc.jar 文件。在 Windows 系统上，将路径分隔符指定为正斜杠字符 (/)，例如 C:/Program Files/IBM/SQLLIB/java/db2jcc.jar。在 z/OS 上，指定 db2jcc.jar 文件的完整路径。例如，wmqfte.database.driver=/db2/db2v10/jdbc/classes/db2jcc.jar。 在 z/OS 系统上，您必须引用以下所有 JAR 文件： • db2jcc.jar • db2jcc_license_cisuz.jar • db2jcc_javax.jar 如果数据库驱动程序包含多个 JAR 文件（例如，Db2 V9.1 需要一个驱动程序 JAR 文件和一个许可证 JAR 文件），请将所有这些 JAR 文件包含在此属性中。使用针对您平台的类路径分隔符（即 Windows 系统上为分号字符 (;)，其他平台上为冒号字符 (:)）来分隔多个文件名。	无缺省值

属性名	描述	缺省值
wmqfte.database.exclude_duplicate_metadata	控制是否在元数据表中存储条目来包含可在数据库记录器模式内的其他表中找到的信息。将此值设置为 true 或 false。缺省情况下，不再存储这些元数据条目，因为这与现有数据重复，因此会浪费数据库存储容量。属性条目以及表（其中显示相同数据）如下所示： - com.ibm.wmqfte.SourceAgent TRANSFER_EVENT 或 CALL_REQUEST - com.ibm.wmqfte.DestinationAgent TRANSFER_EVENT - com.ibm.wmqfte.MqmdUser TRANSFER_EVENT 或 CALL_REQUEST - com.ibm.wmqfte.OriginatingUser TRANSFER_EVENT 或 CALL_REQUEST - com.ibm.wmqfte.OriginatingHost TRANSFER_EVENT 或 CALL_REQUEST - com.ibm.wmqfte.TransferId TRANSFER 或 CALL_REQUEST - com.ibm.wmqfte.JobName TRANSFER 或 CALL_REQUEST 将此属性的值设置为 false 会使这些元数据条目存储在元数据表中。	true
wmqfte.database.host	仅限 Db2: 对于 IBM WebSphere MQ V7.5 或更高版本，是指要使用第 4 类 JDBC 驱动程序连接到的数据库服务器的主机名。如果指定了此属性的值，那么还必须指定 <code>wmqfte.database.port</code> 的值。如果这两个属性都没有定义，那么数据库记录器将使用缺省类型 2 JDBC 驱动程序进行连接。 如果指定了此属性的值，那么必须具有此记录器的凭证文件（文件路径由 <code>wmqfte.database.credentials.file</code> 属性定义），并且必须能够访问该文件以定义用于连接至数据库的用户名和密码，即使数据库位于本地系统上也是如此。	无缺省值
wmqfte.database.name	包含 IBM MQ Managed File Transfer 日志表的数据库实例（或使用 Db2 for z/OS 时的子系统）的名称。	无缺省值
wmqfte.database.type	正在使用的数据库管理系统：Db2 或 Oracle。将该值设置为 db2 或 oracle 。	db2
wmqfte.database.port	仅限 Db2: 对于 IBM WebSphere MQ V7.5 或更高版本，是指要使用第 4 类 JDBC 驱动程序连接到的数据库服务器的端口号。如果指定了此属性的值，那么还必须指定 <code>wmqfte.database.host</code> 的值。如果这两个属性都没有定义，那么数据库记录器将使用缺省类型 2 JDBC 驱动程序进行连接。 如果指定了此属性的值，那么必须具有此记录器的凭证文件（文件路径由 <code>wmqfte.database.credentials.file</code> 属性定义），并且必须能够访问该文件以定义用于连接至数据库的用户名和密码，即使数据库位于本地系统上也是如此。	无缺省值
wmqfte.database.schema	包含 IBM MQ Managed File Transfer 日志记录表的数据库模式。在大多数情况下缺省值都适用，但您可能需要出于自身特定于站点的数据库考虑，指定其他值。	FTELOG

属性名	描述	缺省值
wmqfte.database.native.library.path	包含所选数据库驱动程序（如果有）所需的本机库的路径。例如，Db2 on AIX 系统的 2 类驱动程序需要 /opt/IBM/db2/V9.5/lib32/ 中的库。作为该属性的替代方案，您可以使用其他方法设置 java.library.path 系统属性。 在 Solaris 和 HP-UX 系统上，运行 fteStartLogger 命令之前，还必须设置并导出 LD_LIBRARY_PATH 环境变量以包含该路径。	无缺省值
wmqfte.file.logger.fileDirectory	文件记录器日志文件所在的目录。	mqft/logs/coordination_dir/ loggers/logger_name/logs
wmqfte.file.logger.fileSize	允许一个日志文件增大到的最大大小。该大小值是大于零的正整数，后跟以下某个单位：KB、MB、GB、m（分钟）、h（小时）、d（天）、w（周）。例如，wmqfte.file.logger.fileSize=5MB 指定最大文件大小为 5MB。 wmqfte.file.logger.fileSize=2d 指定最大文件大小为 2 天的数据。	10MB
wmqfte.file.logger.fileCount	要创建的日志文件的最大数目。数据量超过此数目的文件中可以存储的最大量时，将删除最旧的文件，以便文件数从不超过指定的值。	3
wmqfte.file.logger.mode	正在使用的记录器方式：circular 或 linear。将该值设置为 CIRCULAR 或 LINEAR。 循环-文件记录器将信息写入文件，直到该文件达到使用 wmqfte.file.logger.fileSize 属性定义的最大大小为止。达到最大大小时，文件记录器将启动新文件。在此方式下写入的最大文件数由使用 wmqfte.file.logger.fileCount 属性定义的值控制。达到该最大文件数时，文件记录器将删除首个文件并重新创建该文件以用作当前活动文件。如果 wmqfte.file.logger.fileSize 属性中定义的值是固定大小的字节单位（例如，KB、MB 或 GB），那么该方式下使用的磁盘空间的上限等于 fileSize 乘以 fileCount。如果 wmqfte.file.logger.fileSize 属性中定义的值是时间单位（例如，m，h，d 或 w），那么最大大小取决于这些时间段内系统中日志消息的吞吐量。在此方式下运行时使用的日志文件命名约定为：logger_namenumber-timestamp.log，其中： - logger_name 是在 fteCreateLogger 命令中为记录器指定的名称。 - number 是文件在集合内的编号。 - timestamp 是创建文件时的时间戳记。 例如，LOGGER1-20111216123430147.log LINEAR - 文件记录器将信息写入文件，直至该文件达到其最大大小（使用 wmqfte.file.logger.fileSize 属性定义）。达到最大大小时，文件记录器将启动新文件。先前已写入信息的文件将不会删除，从而允许将其保留为日志消息的历史记录。以 线性 方式运行时，不会删除文件，因此将忽略 wmqfte.file.logger.fileCount 属性，因为可以创建的文件数没有上限。因为在该方式下运行时没有上限，所以有必要跟踪日志文件所用的磁盘空间量以避免磁盘空间不足。在此方式下运行时使用的日志文件命名约定为：logger_name-timestamp.log，其中： - logger_name 是在 fteCreateLogger 命令中为记录器指定的名称。 - timestamp 是创建文件时的时间戳记。 例如，LOGGER-20111216123430147.log	无缺省值

属性名	描述	缺省值
wmqfte.max.retry.interval	记录器遇到持续性错误时，两次重试之间的最长时间（秒）。 某些错误情况（例如，丢失数据库连接）会使记录器无法继续运行。当出现这种情况时，记录器将回滚当前事务，等待一段时间，然后重试。记录器等待的时间最初很短，这样短暂错误可以快速得以解决。但是，记录器每次重试时，其等待的时间都会延长。这可防止在错误情况持续较长时间时（例如，数据库关闭进行维护时）执行过多的不必要工作。 使用此属性可设置等待时间长度的限制，以便在解决错误情况的合理时间内进行重试。	600
loggerQMgrRetryInterval	记录器的进程控制器对队列管理器的可用性进行检查的时间间隔（秒）。	30
maxRestartCount	在 maxRestartInterval 属性值所指定的时间间隔内可发生的最大重新启动次数。当超出该值时，记录器的进程控制器会停止重新启动记录器，而改为执行基于 maxRestartDelay 属性值的操作。	4
maxRestartInterval	一段时间间隔（秒），记录器的进程控制器度量此时间间隔内记录器的重新启动次数。如果该时间间隔内的重新启动次数超过 maxRestartCount 属性的值，那么记录器的进程控制器会停止重新启动记录器。记录器的进程控制器会改为执行基于 maxRestartDelay 属性值的操作。	120
maxRestartDelay	确定当记录器重新启动的频率超过 maxRestartCount 和 maxRestartInterval 属性值时，记录器的进程控制器的行为。如果指定小于或等于零的值，那么记录器的进程控制器会停止。如果指定大于零的值，那么这是重置记录器的进程控制器保存的重新启动历史记录信息并重新启动记录器之前等待的秒数。	-1
wmqfte.oracle.port	记录器用于连接到 Oracle 实例的端口。此端口也称为 TNS 侦听器。	1521
wmqfte.oracle.host	记录器用于连接到 Oracle 实例的主机。	localhost
armELEMTYPE	可选属性。如果 Automatic Restart Manager (ARM) 将记录器配置为重新启动，那么请将此属性设置为关联的 ARM 策略中指定的 ARM ELEMTYPE 参数值。对于记录器，请将 ELEMTYPE 设置为 SYSBFGLG。	未设定
armELEMENT	可选属性。如果 Automatic Restart Manager (ARM) 将记录器配置为重新启动，那么请将此属性设置为关联的 ARM 策略中指定的 ARM ELEMENT 参数值。您可以将 ELEMENT 值设置为与记录器名称相对应。	未设定
loggerQMgrAuthenticationCredentialsFile	包含用于连接到记录器的协调队列管理器的 MQ 连接凭证的文件的路径。	此属性的缺省值为 %HOMEDRIVE%\HOMEPATH%\mqmftcredentials.xml (在 Windows 上) 和 \$HOME/MQMFTCcredentials.xml (在其他平台上)。
跟踪	可选属性。要在记录器启动时启用跟踪的情况下运行记录器时的跟踪规范。跟踪规范是以逗号分隔的类、等于字符以及跟踪级别列表。 例如，com.ibm.wmqfte.databaseslogger、com.ibm.wmqfte.databaseslogger.operation=all。 您可以在冒号分隔的列表中指定多个跟踪规范。例如，com.ibm.wmqfte.databaseslogger=moderate:com.ibm.wmqfte.databaseslogger.operation=all	None
traceFiles	可选属性。保存的跟踪文件的总数。该值适用于记录器的进程控制器以及记录器自身。	5

属性名	描述	缺省值
traceSize	可选属性。将跟踪合并到下一个文件之前，每个跟踪文件的最大大小（MB）。该值适用于记录器的进程控制器以及记录器自身。	20

相关参考

第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』

从 IBM WebSphere MQ V7.5 起，可以在 受管文件传输 属性中使用表示文件或目录位置的环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化。例如，哪个用户在运行进程。

独立记录器的备用配置

通常，不管独立记录器是文件还是数据库类型，都与协调队列管理器位于同一个系统上，并以 IBM MQ 绑定方式连接到协调队列管理器。但是，它也可以安装在与连接到协调队列管理器的任何队列管理器相同的系统上。独立记录器使用其自动创建的预订接收消息。这是在安装指示信息中描述的配置。

但是，如果有特定于位置方面的考虑，那么可以将独立记录器配置为以其他两种由 `wmqfte.message.source.type` 属性控制的方法来接收消息。该属性在 [数据库记录器属性](#) 中进行描述。

管理预订

缺省情况下，独立记录器使用缺省持久预订选项和受管预订，针对 `SYSTEM.FTE/Log/#` 主题创建自己的预订（即，队列管理器控制用于在消息传递到应用程序前保存消息的支持队列）。如果预订或队列需要其他选项，那么您可以自行创建预订，设置所需的选项并将独立记录器配置为使用该预订。请记住，为独立记录器添加许可权以使用您创建的预订。

使用此配置的一个示例是通过使用两个通配符预订对日志空间进行分区，将名称以 `FINANCE` 开头的代理中的日志发送到一个数据库，并将名称以 `ACCOUNTING` 开头的代理中的日志发送到另一个数据库。这种配置需要两个独立记录器实例，每个实例都有各自针对所需预订的 `logger.properties` 文件，以及各自的命令队列和拒绝队列。

要仅从名称以 `ACCOUNTING` 开头的代理收集日志消息，可在协调队列管理器上创建主题字符串为 `SYSTEM.FTE/Log/ACCOUNTING*` 的预订对象。将 **通配符用法值** 设置为 **字符级通配符**。还必须将条目添加到记录器的 `logger.properties` 文件。例如，如果使用这些设置创建名为 `ACCOUNTING.LOGS` 的预订对象，请将以下条目添加到 `logger.properties` 文件：

```
wmqfte.message.source.type=administrative subscription
wmqfte.message.source.name=ACCOUNTING.LOGS
```

独立记录器仅处理以主题字符串 `SYSTEM.FTE/Log/` 开头的日志消息。您可以指定限制更多的主题字符串，但不能指定限制更少的字符串。如果您错误地指定了限制更少的字符串，那么与 `SYSTEM.FTE/Log/` 之外的主题字符串相关的所有发布都将转至拒绝队列，并且独立记录器会产生错误消息 `BFGDB0002E`。该错误消息表明独立记录器配置有问题。

队列

典型的拓扑是独立记录器与协调队列管理器在同一个系统上运行。如果这不可能，您可以通过将另一队列管理器上的队列作为预订目标而在协调队列管理器上创建一个预订（使用远程队列定义或使用预订的 `DESTQMGR` 属性）。然后，记录器可以在托管第二个队列管理器的系统上运行，并从队列中读取消息。为确保事务的完整性，独立记录器必须始终以绑定方式连接到其队列管理器。您必须在独立记录器所连接的第一队列管理器上定义拒绝队列和命令队列。队列管理器必须为 WebSphere MQ V7 或更高版本。

例如，要通过预订收集正放入队列 `USER.QUEUE` 的日志消息，可将以下条目添加到 `logger.properties` 文件：

```
wmqfte.message.source.type=queue
wmqfte.message.source.name=USER.QUEUE
```

安装 Java EE 数据库记录器

按照以下指示信息安装并配置 Java EE 数据库记录器以供 Managed File Transfer 使用。

关于此任务

有关 Java EE 数据库记录器的更多信息，请参阅主题第 151 页的『配置 Managed File Transfer 记录器』。

注: 不能同时运行 Java EE 数据库记录器和独立记录器，除非这些记录器使用单独的数据库实例。

过程

1. 安装 Java EE 数据库记录器之前，必须准备好环境。使用主题第 169 页的『准备安装 Java EE 数据库记录器』中的指示信息。
2. 在兼容 Java Platform, Enterprise Edition (Java EE) 的应用程序服务器中安装 Java EE 数据库记录器。有关指示信息，请参阅下列主题：
 - 第 171 页的『将 Java EE 数据库记录器与 WebSphere Application Server V7.0 一起安装』
 - 第 175 页的『将 Java EE 数据库记录器与 WebSphere 应用程序服务器 Community Edition 一起安装』

相关任务

第 169 页的『准备安装 Java EE 数据库记录器』

在安装 Java EE 数据库记录器以供 Managed File Transfer 使用之前，请按照以下指示信息来准备环境。

第 171 页的『将 Java EE 数据库记录器与 WebSphere Application Server V7.0 一起安装』

按照以下指示信息安装并配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器，以便将 Managed File Transfer 与 WebSphere Application Server V7 一起配合使用。

第 175 页的『将 Java EE 数据库记录器与 WebSphere 应用程序服务器 Community Edition 一起安装』

按照以下指示信息安装并配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 数据库记录器，以配合 WebSphere 应用程序服务器 Community Edition 一起使用。

第 178 页的『为 Java EE 数据库记录器配置用户访问权』

在配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 数据库记录器时，您需要用户帐户以访问 IBM MQ、数据库和操作系统。所需的操作系统用户数取决于用于托管这些组件的系统数。

第 179 页的『从独立数据库记录器迁移至 Java EE 数据库记录器』

您可以从独立数据库记录器迁移到 Java EE 数据库记录器。必须停止独立数据库记录器，并安装 JEE 数据库记录器。为避免日志条目丢失或重复，您必须先停止正在向 SYSTEM.FTE 主题发布的消息，然后再停止独立数据库记录器，并在安装 Java EE 数据库记录器之后重新启动独立数据库记录器。备份您的数据库，然后再执行迁移。生成的新信息。

相关参考

第 441 页的『记录器的权限』

运行记录器的操作系统用户需要具有对记录器队列和 SYSTEM.FTE 主题的特定 IBM MQ 权限。

准备安装 Java EE 数据库记录器

在安装 Java EE 数据库记录器以供 Managed File Transfer 使用之前，请按照以下指示信息来准备环境。

关于此任务

有关 Java EE 数据库记录器的更多信息，请参阅主题第 151 页的『配置 Managed File Transfer 记录器』。

过程

1. 使用数据库的文档指导您安装数据库软件。

如果 JDBC 支持是您数据库的可选组件，那么必须安装此组件。
2. 使用数据库提供的工具创建数据库。数据库的表空间和缓冲池页面大小至少应为 8K。

缺省模式名称为 FTELOG。如果使用不同于 FTELOG 的模式名称，那么必须编辑针对数据库提供的 SQL 文件 (ftelog_tables_db2.sql 或 ftelog_tables_oracle.sql) 以反映这一情况，然后继续下一步。

3. 使用您数据库的工具创建所需的数据库表。

在分布式平台上，文件 `ftelog_tables_db2.sql` 和 `ftelog_tables_oracle.sql` 包含可运行以创建表的 SQL 命令。

在 z/OS 上，需要运行的文件取决于使用的 Db2 for z/OS 版本：

- 对于 Db2 for z/OS 9.0 以及更低版本，运行文件 `ftelog_tables_zos.sql` 以创建表。此文件在创建表时，将 `INTEGER` 数据类型用于表示所传输文件大小的字段以及表示与每个传输关联的表标识的字段。
- 对于 Db2 for z/OS 9.1 以及更高版本，运行文件 `ftelog_tables_zos_bigint.sql` 以创建表。此文件在创建表时，将 `BIGINT` 数据类型用于表示所传输文件大小的字段以及表示与每个传输关联的表标识的字段。

4. 如果通过 FTELOG 更改了模式名称，那么必须更改 EAR 文件中的模式名称。有关更多信息，请参阅第 170 页的『在 Java EE 数据库记录器中更改模式名称』。

5. 在 IBM MQ 中创建拒绝队列。

由于记录器从不废弃日志消息，因此，如果记录器遇到其无法处理的消息，会将消息放入拒绝队列以进行检查，并有可能重新处理。请勿将队列管理器的死信队列用于此目的，这是因为被拒绝的消息没有 DLH 头，并且被拒绝的消息不得与由于其他原因而放入死信队列的消息混合在一起。

fteCreateLogger 命令用于创建拒绝队列。该拒绝队列的缺省名称是 `SYSTEM.FTE.LOG.RJCT.logger_name`。

6. 按照主题针对 JEE 记录器配置用户访问权中的指示信息操作。

下一步做什么

现在，您可以将 Java EE 数据库记录器安装在兼容 Java EE 的应用程序服务器中。基于使用的应用程序服务器，使用以下主题中的指示信息：

- 第 171 页的『将 Java EE 数据库记录器与 WebSphere Application Server V7.0 一起安装』
- 第 175 页的『将 Java EE 数据库记录器与 WebSphere 应用程序服务器 Community Edition 一起安装』

在 Java EE 数据库记录器中更改模式名称

Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 数据库记录器可以使用具有非缺省模式名称的数据库。您必须更改 JEE 数据库记录器 EAR 文件中的模式名称。

关于此任务

要更改 Java EE 数据库记录器所使用模式的名称，请完成以下步骤：

过程

1. 通过使用以下命令，从 EAR 文件中抽取 JPA JAR 文件：

```
jar -xvf ear_file lib/jpa_file
```

其中：

- `ear_file` 是 `com.ibm.wmqfte.databaseslogger.jee.oracle.ear` 或 `com.ibm.wmqfte.databaseslogger.jee.ear`，具体取决于您使用的是 Db2 还是 Oracle。
- `jpa_file` 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，具体取决于您使用的是 Db2 还是 Oracle。

2. 通过使用以下命令，从 JPA JAR 文件中抽取 persistence.xml 文件：

```
jar -xvf lib/jpa_file META-INF/persistence.xml
```

其中：

- `jpa_file` 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，具体取决于您使用的是 Db2 还是 Oracle。

3. 编辑 `persistence.xml` 文件，更改以下行：

```
<property name="openjpa.jdbc.Schema" value="schema_name" />
```

其中：

- `schema_name` 是您希望使用的模式名称。

4. 通过使用以下命令，以修改后的 `persistence.xml` 文件更新 JPA JAR：

```
jar -uvf lib/jpa_file META-INF/persistence.xml
```

其中：

- `jpa_file` 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，具体取决于您使用的是 Db2 还是 Oracle。

5. 通过使用以下命令，以修改后的 JPA JAR 文件更新 EAR 文件：

```
jar -uvf ear_file lib/jpa_file
```

其中：

- `ear_file` 是 `com.ibm.wmqfte.databaselogger.jee.oracle.ear` 或 `com.ibm.wmqfte.databaselogger.jee.ear`，具体取决于您使用的是 Db2 还是 Oracle。
- `jpa_file` 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，具体取决于您使用的是 Db2 还是 Oracle。

下一步做什么

使用修改后的 EAR 文件安装 Java EE 数据库记录器。

相关任务

第 171 页的『[将 Java EE 数据库记录器与 WebSphere Application Server V7.0 一起安装](#)』

按照以下指示信息安装并配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器，以便将 Managed File Transfer 与 WebSphere Application Server V7 一起配合使用。

第 175 页的『[将 Java EE 数据库记录器与 WebSphere 应用程序服务器 Community Edition 一起安装](#)』

按照以下指示信息安装并配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 数据库记录器，以配合 WebSphere 应用程序服务器 Community Edition 一起使用。

将 Java EE 数据库记录器与 WebSphere Application Server V7.0 一起安装

按照以下指示信息安装并配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器，以便将 Managed File Transfer 与 WebSphere Application Server V7 一起配合使用。

开始之前

安装 JEE 数据库记录器应用程序之前，请按照主题第 169 页的『[准备安装 Java EE 数据库记录器](#)』和第 196 页的『[在 WebSphere Application Server V7.0 中设置本机库路径](#)』中的指示信息进行操作。

关于此任务

有关 Java EE 数据库记录器的更多信息，请参阅第 151 页的『[配置 Managed File Transfer 记录器](#)』。

过程

1. 设置 XA JDBC 提供程序：

- a) 从 WebSphere Application Server V7.0 管理控制台导航选择资源 > **JDBC** > **JDBC 提供程序**。
- b) 单击**新建**，以使用控制台向导创建 JDBC 提供程序。
- c) 在向导的步骤 1 中，从**数据库类型**列表中选择正在使用的数据库，并从**提供程序类型**列表中选择关联的提供程序类型。从**实施类型**列表中选择 **XA 数据源**。单击**下一步**。

- d) 在向导的步骤 2 中，确保正确设置了必需数据库 jar 文件的目录位置。单击下一步。
- e) 单击摘要页面上的**完成**以创建 JDBC 提供程序。
- 2. 创建认证别名。为数据源创建一个别名，并为 IBM MQ 创建一个别名：
 - a) 从 WebSphere Application Server V7.0 管理控制台导航中选择**安全性 > 全局安全性**。
 - b) 在**认证**标题下，展开 **Java 认证和授权服务**。
 - c) 单击 **J2C 认证数据**。这样会打开认证别名页面。
 - d) 为数据源创建认证别名：
 - i) 单击**新建**。
 - ii) 输入**别名、用户标识、密码和描述**的详细信息。在**用户标识和密码**字段中输入的详细信息必须与在创建数据库用户时输入的详细信息相匹配。有关更多信息，请参阅[JEE 数据库记录器配置用户访问权](#)。
 - iii) 单击**确定**。
 - e) 为 IBM MQ 创建一个认证别名：
 - i) 单击**新建**。
 - ii) 输入**别名、用户标识、密码和描述**的详细信息。在**用户标识和密码**字段中输入的详细信息必须与 IBM MQ 安装的用户和密码设置相匹配。
 - iii) 单击**确定**。
- 3. 创建数据源：
 - a) 从 WebSphere Application Server V7.0 管理控制台导航选择 **资源 > JDBC > 数据源**。
 - b) 选择**作用域**下拉列表，并将作用域更改为适当值。例如，Node=yourNode，Server=yourServer。
 - c) 单击**新建**，以使用控制台向导创建数据源。
 - d) 在向导的步骤 1 中，在**数据源名称**字段中输入 wmqfte-database，并在 **JNDI 名称**字段中输入 jdbc/wmqfte-database。单击下一步。
 - e) 在向导的步骤 2 中，使用**选择现有的 JDBC 提供程序**下拉列表来选择先前步骤中创建的 JDBC 提供程序。单击下一步。
 - f) **Db2**：在向导的步骤 3 中，在**驱动程序类型**字段中输入 4。
 - g) **Db2**：在**数据库名称、服务器名称和端口号**字段中输入详细信息，然后单击下一步。

Oracle：在 **URL** 字段中输入连接 URL，并在**数据存储 帮助程序类**字段中选择正确的数据存储帮助程序。

Oracle RAC：当连接到 Oracle Real Application Cluster 时，连接 URL 必须包含连接所有可用数据库实例所需的主机信息。
 - h) 在向导的步骤 4 中，从用于 **XA 恢复的认证别名**列表中，选择您在步骤 2d 中定义的数据源认证别名的名称。从**组件管理的认证别名**和**容器管理的认证别名**列表中选择相同的名称。
 - i) 单击摘要页面上的**完成**以创建数据源。
- 4. 可选：验证数据源的配置：
 - a) 从 WebSphere Application Server V7.0 管理控制台导航选择 **资源 > JDBC > 数据源**。
 - b) 单击**测试连接**按钮。
- 5. 创建主题。
 - a) 从 WebSphere Application Server V7.0 管理控制台导航中，单击**资源 > JMS > 主题**。
 - b) 选择**作用域**下拉列表，并将作用域更改为适当值。例如，Node=yourNode，Server=yourServer。
 - c) 单击**新建**。
 - d) 单击 **IBM MQ 消息传递提供程序**。
 - e) 在主题属性页面的“**管理**”面板上，为**名称**和 **JNDI 名称**字段选择唯一值，稍后您将在配置中引用这些值。

f) 在“**IBM MQ 主题**”面板中的**主题名称**字段中，输入 `SYSTEM.FTE/Log/#`。

6. 创建激活规范：

- a) 从 WebSphere Application Server V7.0 管理控制台导航中，单击**资源 > JMS > 激活规范**。
 - b) 选择**作用域**下拉列表，并将作用域更改为适当值。例如，`Node=yourNode, Server=yourServer`。
 - c) 单击**新建**。
 - d) 单击 **IBM MQ 消息传递提供程序**。
 - e) 在向导的步骤 1 中，为**名称**和 **JNDI 名称**字段选择唯一值，稍后您将在配置中再次引用这些值。
 - f) 在步骤 1.1 中，在目标 **JNDI 名称**字段中输入您在步骤 5 中所设置主题的 JNDI 名称。
 - g) 从**目标类型**列表中选择**主题**。
 - h) 在向导的步骤 1.2 中，选择**持久预订**。在**预订名称**字段中输入 `SYSTEM.FTE.DATABASELOGGER.AUTO`。
 - i) 在向导的步骤 2 中，选择**在该向导中输入所有必需信息**。
 - j) 在步骤 2.1 中，在**队列管理器或队列共享组名**字段中输入队列管理器名称。
 - k) 在步骤 2.2 中，从**传输**列表中选择您所选的传输方法。如果选择**绑定**，那么无需任何其他信息。如果选择**客户机**或**先绑定，然后客户机**，那么输入**主机名、端口和服务器连接通道**的详细信息。
 - l) 可选：如果希望确认队列管理器是否存在，请单击**测试连接**。但是，您可能预期会收到 `NOT_AUTHORIZED`，直到您在步骤 6n 中引用了认证别名为止。
 - m) 单击**保存**。
 - n) 单击您创建的激活规范的名称。在“**配置**”选项卡的“**常规属性**”部分中，向下滚动至“**高级**”面板，并在**客户机标识**字段中输入用于标识 IBM MQ 连接的唯一名称。必须完成此步骤，否则 IBM MQ 将拒绝您的连接，并返回 `JMSCC0101` 错误代码。
 - o) 如果选择**客户机**作为传输方法，请向下滚动至“**安全性设置**”面板，并从**认证别名**列表中选择您在步骤 8 中定义的认证别名。
 - p) 单击**应用**。
 - q) 在配置选项卡的“**其他属性**”部分中，单击**高级属性**。在“**高级属性**”面板的“**连接使用程序**”部分中，将 1 输入到**最大服务器会话数**字段中。
- 注：**确保完成该步骤，然后再继续。如果不这样做，会导致记录器无法正确运行。
- r) 在配置选项卡的“**其他属性**”部分中，单击**高级属性**。将**如果消息传递失败，停止端点**的值设置为不低于 1。

如果 `_numberOfFailedAttemptsBeforeReject` 属性的值设置为大于 1（请参阅 9j 以了解更多信息），请将**如果消息传递失败，停止端点**设置为不低于 `_numberOfFailedAttemptsBeforeReject` 属性的值。这样可防止在收到无法处理的消息（例如，格式不正确的传输日志消息）时端点停止。有关更多信息，请参阅第 402 页的『[记录器错误处理和拒绝](#)』。

7. 创建队列连接工厂。

- a) 从 WebSphere Application Server V7.0 管理控制台导航中，单击**资源 > JMS > 队列连接工厂**。
- b) 选择**作用域**下拉列表，并将作用域更改为适当值。例如，`Node=yourNode, Server=yourServer`。
- c) 单击**新建**。
- d) 单击 **IBM MQ 消息传递提供程序**。
- e) 在向导的步骤 1 中，为**名称**和 **JNDI 名称**字段选择唯一值，稍后您将在配置中再次引用这些值。
- f) 在步骤 2 中，选择**在该向导中输入所有必需信息**。
- g) 在步骤 2.1 中，在**队列管理器或队列共享组名**字段中输入队列管理器名称。
- h) 在步骤 2.2 中，从**传输**列表中选择您所选的传输方法。如果选择**绑定**，那么无需任何其他信息。如果选择**客户机**或**先绑定，然后客户机**，那么输入**主机名、端口和服务器连接通道**的详细信息。
- i) 可选：如果希望确认队列管理器是否存在，请单击**测试连接**。但是，您可能预期会收到 `NOT_AUTHORIZED`，直到您在步骤 7h 中引用了认证别名为止。

- j) 如果选择了**客户机**或**先绑定**，然后**客户机**作为传输方法，请单击您刚创建的队列连接工厂的名称。向下滚动至配置选项卡的“安全性设置”面板，从用于**XA**恢复的认证别名和容器管理的认证别名列表中选择您在步骤 2e 中定义的认证别名。

8. 在 WebSphere Application Server 中创建拒绝队列：

- 从 WebSphere Application Server V7.0 管理控制台导航中，单击**资源 > JMS > 队列**。
- 选择**作用域**下拉列表，并将作用域更改为适当值。例如，Node=yourNode，Server=yourServer。
- 单击**新建**。
- 单击 **IBM MQ 消息传递提供程序**。
- 为**名称**和 **JNDI 名称**字段选择唯一值，稍后您将在配置中再次引用这些值。
- 在**队列名称**字段中输入 SYSTEM.FTE.LOG.RJCT.logger_name。确保在协调队列管理器上创建了该队列。
- 在**队列管理器名称**字段中输入队列管理器名称。
- 单击**确定**。

9. 安装 JEE 数据库记录器应用程序：

- 从 WebSphere Application Server V7.0 管理控制台中选择**应用程序 > 新建应用程序**。
- 选择**作用域**下拉列表，并将作用域更改为适当值。例如，Node=yourNode，Server=yourServer。
- 从选项列表中选择**新建企业应用程序**。
- 在**准备应用程序安装**页面上，从 IBM MQ Managed File Transfer 服务器安装的 MQ_INSTALLATION_PATH/mqft/web 目录中选择 com.ibm.wmqfte.databaselogger.jee.ear 文件或 com.ibm.wmqfte.databaselogger.jee.oracle.ear 文件，然后单击**下一步**。
- 在以下屏幕上，选择**详细**以显示所有安装选项和参数，然后单击**下一步**。
- 单击向导步骤 1-4 中的**下一步**，接受缺省值。
- 在向导的步骤 5 针对消息驱动的 **Bean** 绑定侦听器中，滚动至**侦听器绑定**部分。单击**激活规范**。输入以下字段的必需值：

目标资源 JNDI 名称

您在步骤 6d 中创建激活规范时指定的 JNDI 名称。

目标 JNDI 名称

您在步骤 5d 中创建主题时指定的 JNDI 名称。

单击**下一步**。

- 在向导的步骤 6“**将资源引用映射到资源**”中，在**目标资源 JNDI 名称**字段中输入详细信息。该名称是您在步骤 7c 中为拒绝队列连接工厂指定的 JNDI 名称。单击**下一步**。
- 在向导的步骤 7“**将资源环境条目引用映射到资源**”中，在**目标资源 JNDI 名称**字段中输入详细信息。该名称是您在步骤 8d 中创建的拒绝队列的 JNDI 名称。单击**下一步**。
- 在向导的步骤 8 中，**映射 EJB 模块的环境条目**，接受缺省值 1。单击**下一步**。

Oracle RAC: 连接到 Oracle Real Application Cluster 时，必须将

_numberOfFailedAttemptsBeforeReject 属性的值设置为**至少 2**。此属性确定发生故障后记录器尝试处理审计消息的次数。在数据库故障转移情况下，至少有可能发生一次失败。为了避免不必要地将消息移动到拒绝队列，增加该值可进行第二次尝试，这通常会成功，因为会连接到新的数据库实例。如果在测试时发现消息在数据库实例故障转移期间仍移动到拒绝队列，可进一步增加该值：实例之间的切换计时可能导致同一消息发生多次失败。但是，要知道增加该值会影响所有失败情况（例如，格式不正确的消息），而不仅仅是数据库故障转移，因此请慎重增加该值，以避免不必要的重试。

- 在向导的步骤 9 **模块的元数据**中，单击**下一步**。
- 在向导的步骤 10 **摘要**中，单击**完成**。

10. 您现在可以从 WebSphere Application Server V7.0 管理控制台中启动该应用程序：

- a) 从控制台导航中，选择应用程序 > 应用程序类型 > **WebSphere** 企业应用程序。
- b) 选中集合表中**记录器**企业应用程序的复选框，然后单击启动。

将 Java EE 数据库记录器与 WebSphere 应用程序服务器 Community Edition 一起安装

按照以下指示信息安装并配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 数据库记录器，以配合 WebSphere 应用程序服务器 Community Edition 一起使用。

开始之前

在安装 Java EE 数据库记录器应用程序之前，请按照第 169 页的『准备安装 Java EE 数据库记录器』主题中的指示信息执行操作。

关于此任务

有关 Java EE 数据库记录器的更多信息，请参阅主题第 151 页的『配置 Managed File Transfer 记录器』。

过程

1. 部署 IBM MQ 资源适配器 wmq.jmsra.rar。

注: 如果已在 WebSphere 应用程序服务器 Community Edition 环境中部署 IBM MQ Managed File Transfer Web 网关，那么表示您已拥有 IBM MQ 资源适配器。在此情况下，您需要卸载资源适配器的该实例，并使用包含 Web 网关和 Java EE 数据库记录器的组合资源的计划文件进行重新部署。

- 要使用协调队列管理器 QM_JUPITER 部署 Java EE 数据库记录器的 IBM MQ 资源适配器，请执行以下步骤。本示例适用于 WebSphere 应用程序服务器 Community Edition 实例与要连接到的 IBM MQ 队列管理器运行在同一系统上的情况。
 - a. 创建计划文件以定义与 MQMFT 协调队列管理器的连接。以下示例计划文件定义了与名为 QM_JUPITER 的队列管理器的连接，以及对该队列管理器上名为 SYSTEM.FTE.LOG.RJCT.LOGGER1 的队列的引用。

```
<?xml version="1.0" encoding="UTF-8"?>
<connector xmlns="http://geronimo.apache.org/xml/ns/j2ee/connector">
  <resourceadapter>
    <resourceadapter-instance>
      <resourceadapter-name>WMQ</resourceadapter-name>
      <workmanager>
        <gbean-link>DefaultWorkManager</gbean-link>
      </workmanager>
    </resourceadapter-instance>
    <outbound-resourceadapter>
      <connection-definition>
        <connectionfactory-interface>javax.jms.ConnectionFactory</connectionfactory-interface>
        <connectiondefinition-instance>
          <name>jms/WMQFTEJEEDBLoggerRejectQueueCF</name>
          <config-property-setting name="queueManager">QM_JUPITER</config-property-setting>
          <config-property-setting name="transportType">BINDINGS</config-property-setting>
          <connectionmanager>
            <xa-transaction>
              <transaction-caching/>
            </xa-transaction>
            <single-pool>
              <max-size>10</max-size>
              <min-size>1</min-size>
              <blocking-timeout-milliseconds>5000</blocking-timeout-milliseconds>
              <idle-timeout-minutes>2</idle-timeout-minutes>
              <match-all />
            </single-pool>
          </connectionmanager>
        </connectiondefinition-instance>
      </connection-definition>
    </outbound-resourceadapter>
  </resourceadapter>
  <adminobject>
    <adminobject-interface>javax.jms.Queue</adminobject-interface>
    <adminobject-class>com.ibm.mq.connector.outbound.MQQueueProxy</adminobject-class>
    <adminobject-instance>
      <message-destination-name>jms/WMQFTEJEEDBLoggerRejectQueue</message-destination-name>
      <config-property-setting name="baseQueueManagerName">QM_JUPITER</config-property-setting>
      <config-property-setting name="baseQueueName">SYSTEM.FTE.LOG.RJCT.LOGGER1</config-property-setting>
    </adminobject-instance>
  </adminobject>
</connector>
```

```
</adminobject>
</connector>
```

要在环境中使用此计划文件，请将 QM_JUPITER 更改为协调队列管理器的名称。

- b. 打开 WebSphere 应用程序服务器 CE 管理控制台。
 - c. 从“欢迎页面”上的公共控制台操作列表中，单击**部署新应用程序 > 新部署**。
 - d. 在归档字段中，输入 `mq_install_root/java/lib/jca/wmq.jmsra.rar`
 - e. 在计划字段中，输入您在步骤 1a 中创建的计划文件路径。
- 如果您的 WebSphere 应用程序服务器 Community Edition 实例与要连接到的 IBM MQ 队列管理器运行在不同系统上，请执行以下步骤以部署 IBM MQ 资源适配器。
 - a. 创建计划文件以定义与 WMQFTE 协调队列管理器的连接。以下示例计划文件定义了与队列管理器 QM_SATURN（与您的 WebSphere 应用程序服务器 Community Edition 安装位于不同系统上）的连接，以及对该队列管理器上名为 SYSTEM.FTE.LOG.RJCT.LOGGER1 的队列的引用。QM_SATURN 的主机名为 `saturn.example.com`。QM_SATURN 的端口为 1415。QM_SATURN 的通道为 SYSTEM.DEF.SVRCONN。

由于应用程序服务器和队列管理器位于不同的系统上，因此您必须使用客户机方式连接到队列管理器。以下计划文件将名为 `transportType` 的 `<config-property-setting>` 元素的值设置为 CLIENT。

```
<?xml version="1.0" encoding="UTF-8"?>
<connector xmlns="http://geronimo.apache.org/xml/ns/j2ee/connector">
  <resourceadapter>
    <resourceadapter-instance>
      <resourceadapter-name>WMQ</resourceadapter-name>
      <workmanager>
        <gbean-link>DefaultWorkManager</gbean-link>
      </workmanager>
    </resourceadapter-instance>
    <outbound-resourceadapter>
      <connection-definition>
        <connectionfactory-interface>javax.jms.ConnectionFactory</connectionfactory-interface>
        <connectiondefinition-instance>
          <name>jms/WMQFTEJEEDBLoggerRejectQueueCF</name>
          <config-property-setting name="queueManager">QM_SATURN</config-property-setting>
          <config-property-setting name="transportType">CLIENT</config-property-setting>
          <config-property-setting name="channel">SYSTEM.DEF.SVRCONN</config-property-setting>
          <config-property-setting name="hostName">saturn.example.com</config-property-setting>
          <config-property-setting name="port">1415</config-property-setting>
          <connectionmanager>
            <xa-transaction>
              <transaction-caching/>
            </xa-transaction>
            <single-pool>
              <max-size>10</max-size>
              <min-size>1</min-size>
              <blocking-timeout-milliseconds>5000</blocking-timeout-milliseconds>
              <idle-timeout-minutes>2</idle-timeout-minutes>
              <match-all />
            </single-pool>
          </connectionmanager>
        </connectiondefinition-instance>
      </connection-definition>
    </outbound-resourceadapter>
  </resourceadapter>
</adminobject>
<adminobject>
  <adminobject-interface>javax.jms.Queue</adminobject-interface>
  <adminobject-class>com.ibm.mq.connector.outbound.MQQueueProxy</adminobject-class>
  <adminobject-instance>
    <message-destination-name>jms/WMQFTEJEEDBLoggerRejectQueue</message-destination-name>
    <config-property-setting name="baseQueueManagerName">QM_SATURN</config-property-setting>
    <config-property-setting name="baseQueueName">SYSTEM.FTE.LOG.RJCT.LOGGER1</config-property-setting>
  </adminobject-instance>
</adminobject>
</connector>
```

要在环境中使用此计划文件，请将 QM_SATURN 更改为协调队列管理器的名称。将主机名、端口和通道的值更改为协调队列管理器的值。

- b. 将文件 `mq_install_root/java/lib/jca/wmq.jmsra.rar` 从安装 IBM MQ 的系统复制到安装 WebSphere 应用程序服务器 CE 的系统。
 - c. 打开 WebSphere 应用程序服务器 CE 管理控制台。
 - d. 从“欢迎页面”上的公共控制台操作列表中，单击**部署新应用程序 > 新部署**。
 - e. 在归档字段中，输入您获取的 `wmq.jmsra.rar` 文件副本的路径。
 - f. 在计划字段中，输入您创建的计划文件路径。
2. 您必须定义数据库连接器，这样 JEE 数据库记录器应用程序才能 WebSphere 应用程序服务器 Community Edition 环境访问所需数据库。

注: 如果已在 WebSphere 应用程序服务器 Community Edition 环境中部署 IBM MQ Managed File Transfer Web 网关，那么表示您已定义数据库连接器。在此情况下，您无需重复这些步骤。

从 WebSphere 应用程序服务器 Community Edition 管理控制台执行以下步骤：

- a) 根据所使用的 WebSphere 应用程序服务器 Community Edition 级别，从**控制台导航**中选择**服务 > 数据库池**，或选择**资源 > 数据源**。
 - b) 使用 Geronimo 数据库池向导创建数据库池。在**数据库池名称**字段中输入 `jdbc/wmqfte-database`。
 - c) 对于**数据库类型**，请选择 DB2 XA 或 Oracle Thin(视您的数据库而定)。
 - d) 单击**下一步**。
 - e) 在**驱动程序 jar** 字段中，选择适合您数据库的 jar。
 - f) 在**数据库名称**字段中，输入您为获取传输状态信息而要连接到的数据库的名称。
 - g) 在**用户名**字段中，输入要连接到数据库并对数据库进行认证的用户名。
 - h) 在**密码和确认密码**字段中，输入密码以在数据库中进行认证。
 - i) 在**端口号**字段中，输入您使用的端口号（如果不是缺省端口）。
 - j) 确保**驱动程序类型**的值为 4。
 - k) 从**事务类型**列表中选择 XA。
 - l) 单击**部署**。
3. 针对您的环境更新 IBM MQ Managed File Transfer JEE 数据库记录器应用程序 `openejb-jar.xml` 文件。使用 Java SDK jar 实用程序来完成以下步骤：
- a) 通过运行以下命令从提供的 EAR 文件中抽取 EJB jar 文件：

```
jar -xf ear_file_name com.ibm.wmqfte.databaseslogger.jee.ejb.jar
```

其中 `ear_file_name` 是 `com.ibm.wmqfte.databaseslogger.jee.ear` 或 `com.ibm.wmqfte.databaseslogger.jee.oracle.ear`，具体取决于您使用的是 Db2 还是 Oracle。该 EAR 文件位于 IBM MQ Managed File Transfer Server 安装的 `MQ_INSTALLATION_PATH/mqft/web` 目录中。

- b) 通过运行以下命令，从先前抽取的 EJB jar 文件 `com.ibm.wmqfte.databaseslogger.jee.ejb.jar` 中抽取 `META-INF/openejb-jar.xml` 文件：

```
jar -xf com.ibm.wmqfte.databaseslogger.jee.ejb.jar META-INF/openejb-jar.xml
```

- c) 使用文本编辑器编辑抽取的 `META-INF/openejb-jar.xml` 文件。将以下 `activation-config-property` 值更改为与您的环境相匹配：

queueManager

JEE 数据库记录器使用的 IBM MQ 队列管理器的名称。

hostName

用于连接到指定的 IBM MQ 队列管理器的主机名。如果您要以绑定方式连接到队列管理器，那么不需要该值。

transportType

以客户机方式还是绑定方式连接到指定的 IBM MQ 队列管理器。

port

如果将 **transportType** 指定为 bindings，那么不需要该值。用于连接到指定的 IBM MQ 队列管理器的端口。

通道

如果将 **transportType** 指定为 bindings，那么不需要该值。用于连接到指定的 IBM MQ 队列管理器的服务器通道。

- d) 通过运行以下命令，使用修改的 META-INF/openejb-jar.xml 文件更新 EJB jar 文件：

```
jar -uf com.ibm.wmqfte.databaseslogger.jee.ejb.jar META-INF/openejb-jar.xml
```

- e) 通过运行以下命令，使用更新的 EJB jar 文件更新提供的 EAR 文件：

```
jar -uf ear_file_name com.ibm.wmqfte.databaseslogger.jee.ejb.jar
```

其中 *ear_file_name* 是 com.ibm.wmqfte.databaseslogger.jee.ear 或 com.ibm.wmqfte.databaseslogger.jee.oracle.ear，具体取决于您的数据库。

4. 要将该 EAR 文件部署到应用程序服务器，请从 WebSphere 应用程序服务器 Community Edition 管理控制台完成以下步骤。
- 从**控制台导航菜单**中选择**应用程序 > 新部署**。
 - 在**归档**字段中，指定 EAR 文件 com.ibm.wmqfte.databaseslogger.jee.ear 或 com.ibm.wmqfte.databaseslogger.jee.oracle.ear，具体取决于您的数据库。
 - 将**计划**字段留空。
 - 确保选中了**安装后启动应用程序框**。
 - 单击**安装**。这样就会安装并启动 JEE 数据库记录器应用程序。

为 Java EE 数据库记录器配置用户访问权

在配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 数据库记录器时，您需要用户帐户以访问 IBM MQ、数据库和操作系统。所需的操作系统用户数取决于用于托管这些组件的系统数。

关于此任务

运行 Java EE 数据库记录器所需的用户帐户数和类型取决于所用的系统数。需要用户帐户才能访问以下三种环境：

- 本地操作系统
- IBM MQ
- 数据库

您可以在同一个系统上或不同系统上安装 JEE 数据库记录器、IBM MQ 和数据库。可以在以下示例拓扑中安装这些组件：

Java EE 数据库记录器、IBM MQ 和数据库全都位于同一个系统上

您可以定义单个操作系统用户，以用于所有这三个组件。该记录器使用绑定方式连接到 IBM MQ，并使用本机连接来连接到数据库。

Java EE 数据库记录器和 IBM MQ 位于一个系统上，而数据库位于另一个系统上

您为该配置创建两个用户：运行记录器的系统上的一个操作系统用户，以及对数据库服务器上的数据库具有远程访问权的一个操作系统用户。该记录器使用绑定方式连接到 IBM MQ，并使用客户机连接来访问数据库。

Java EE 数据库记录器、IBM MQ 和数据库分别位于不同的系统上

您为该配置创建三个用户：用于启动应用程序服务器的操作系统用户、用于访问要使用的队列和主题的 IBM MQ 用户，以及用于访问数据库表以及将内容插入数据库表的数据库服务器用户。该记录器使用客户机方式访问 IBM MQ，并使用客户机连接来访问数据库。

例如，这些指示信息的其余部分假定用户名为 **fte**log，但您也可以使用任何新的或现有的用户名。请按照以下方式配置该用户的许可权：

过程

1. 确保操作系统用户具有自己的组，并且不在对协调队列管理器具有广泛许可权的任何组中。该用户不应在 **mqm** 组中。在某些平台上，还会自动为 **staff** 组授予队列管理器访问权；记录器用户不应位于 **staff** 组中。您可以使用 MQ Explorer 来查看队列管理器本身及其中对象的权限记录。右键单击该对象，然后选择 **对象权限 > 管理权限记录**。在命令行上，可以使用命令 **dspmqaut**（显示权限）或 **dmpmqaut**（转储权限）。
2. 使用 MQ Explorer 或 **setmqaut (grant 或 revoke authority)** 命令中的 " **管理权限记录** " 窗口为 IBM MQ 用户自己的组添加权限（在 UNIX 上，IBM MQ 权限仅与组关联，而与个人用户无关）。所需权限如下所示：
 - 队列管理器上的 **CONNECT** 和 **INQUIRE**（IBM MQ Java 库需要 **INQUIRE** 许可权才能操作）。
 - 对 **SYSTEM.FTE** 主题的 **SUBSCRIBE** 许可权。
 - 对 **SYSTEM.FTE.LOG.RJCT.logger_name** 队列的 **PUT** 许可权。

给定的 **REJECT** 和 **COMMAND** 队列名称是缺省名称。如果您在配置记录器队列时选择了不同的队列名称，请改为向这些队列名称添加许可权。

3. 执行特定于您所用数据库的数据库用户配置。
 - 如果您的数据库是 **Db2**，请执行以下步骤：

注：有几种机制可用于管理 **Db2** 的数据库用户。这些指示信息适用于基于操作系统用户的缺省方案。

 - 确保 **fte**log 用户未处于任何 **Db2** 管理组（例如，**db2iadm1**、**db2fadm1** 或 **dasadm1**）中
 - 为用户提供连接到数据库的许可权，以及对 **步骤 2：创建所需的数据库表** 过程中所创建表的选择、插入和更新许可权
 - 如果您的数据库是 **Oracle**，请执行以下步骤：
 - 确保 **fte**log 用户不在任何 **Oracle** 管理组中（例如，**ora_dba** on Windows 或 **dba** on UNIX）
 - 为用户提供连接到数据库的许可权，以及对 **步骤 2：创建所需的数据库表** 过程中所创建表的选择、插入和更新许可权

从独立数据库记录器迁移至 Java EE 数据库记录器

您可以从独立数据库记录器迁移到 Java EE 数据库记录器。必须停止独立数据库记录器，并安装 JEE 数据库记录器。为避免日志条目丢失或重复，您必须先停止正在向 **SYSTEM.FTE** 主题发布的消息，然后再停止独立数据库记录器，并在安装 Java EE 数据库记录器之后重新启动独立数据库记录器。备份您的数据库，然后再执行迁移。生成的新信息。

关于此任务

过程

1. 在停止该数据库之前，请针对协调队列管理器运行以下 MQSC 命令：**ALTER QM PS.MODE (COMPAT)**
这会停止向 **SYSTEM.FTE/Log** 主题发布消息。等待该记录器处理完预订中的所有消息。缺省情况下，该预订名为 **SYSTEM.FTE.LOGGER.AUTO**。
2. 使用 **fteStopDatabaseLogger** 命令停止数据库记录器。
3. 使用数据库软件随附的工具来备份数据库。
4. 删除属于该独立数据库记录器的预订。
缺省情况下，该预订名为 **SYSTEM.FTE.LOGGER.AUTO**。
5. 如果您的数据库模式为较早版本，那么必须按顺序将该模式迁移到每个后续级别。例如，如果您的数据库模式为 **V7.0.1** 并且您要迁移到 **V7.0.4**，那么必须将该模式从 **V7.0.1** 迁移到 **V7.0.2**，然后从 **V7.0.2** 迁移到 **V7.0.3**，再从 **V7.0.3** 迁移到 **V7.0.4**。通过针对您必须迁移经过的每个模式版本执行以下某个操作，将数据库模式从 **VoId** 迁移到 **Vnew**（其中 **old** 和 **new** 是描述模式版本的变量）：

- 如果数据库是 Db2 on z/OS 并且您要在 V7.0.2 和 V7.0.3 模式之间或者 V7.0.3 和 V7.0.4 模式之间迁移, 那么您必须创建新的数据库模式, 然后将现有数据复制到其中。有关更多信息, 请参阅 [第 34 页的『将 Db2 on z/OS 上的数据库表迁移到 V8.0.0』](#)。
- 如果您的数据库不是 Db2, 或者您使用 8K 以上的页面大小创建了数据库, 那么可以通过完成以下步骤, 采用与其他版本相同的方法迁移模式。
- 如果您要在其他任何情况下在数据库表之间迁移, 请完成以下步骤:
 - a. 选择适合数据库平台且名称包含字符串 *old-new* 的文件。此文件位于“远程工具和文档”安装的 `MQ_INSTALLATION_PATH/mqft/sql` 目录中。
 - b. 如果您对初始模式进行了修改, 请复审迁移文件以确保该文件与修改的数据库兼容。
 - c. 针对您的数据库运行 SQL 文件。
- 6. 安装 Java EE 数据库记录器 EAR 文件。
- 7. 部署 Java EE 数据库记录器。有关更多信息, 请参阅 [第 169 页的『安装 Java EE 数据库记录器』](#)。
- 8. 针对协调队列管理器运行以下 MQSC 命令: `ALTER QMGR PSMODE(ENABLED)`
这会启用向 `SYSTEM.FTE/Log` 主题发布消息的操作。

结果

迁移 Java EE 数据库记录器

要将 WebSphere Application Server V 7 上的 Java EE 数据库记录器从 IBM WebSphere MQ File Transfer Edition V 7.0 迁移到 IBM WebSphere MQ 7.5 或更高版本, 请完成以下步骤:

过程

1. 打开 WebSphere Application Server 控制台。
2. 单击 **应用程序 > 应用程序类型 > 企业应用程序**。在应用程序列表中找到 IBM WebSphere MQ File Transfer Edition 数据库记录器应用程序。如果数据库记录器应用程序尚未停止, 那么请选择该应用程序并单击 **停止**。
3. 记录先前为 Java EE 数据库记录器设置的配置设置。稍后在步骤 [第 180 页的『7』](#) 中您将需要这些设置。
 - a) 如果您在安装数据库记录器 (请参阅步骤 9 以获取更多信息) 时最初从 EJB 模块的缺省设置做出了更改, 请单击 **企业应用程序 > WebSphere MQ File Transfer Edition 数据库记录器 > EJB 模块的环境条目**, 并记录窗格中的设置。
 - b) 单击 **企业应用程序 > WebSphere MQ File Transfer Edition 数据库记录器 > 消息驱动的 Bean 侦听器绑定**, 并记录所用激活规范、目标资源 JNDI 名称和目标 JNDI 名称。
 - c) 单击 **企业应用程序 > WebSphere MQ File Transfer Edition 数据库记录器 > 资源引用**, 并记录拒绝队列连接工厂详细信息。
 - d) 单击 **企业应用程序 > WebSphere MQ File Transfer Edition 数据库记录器 > 资源环境条目引用**, 并记录拒绝队列详细信息。
4. 通过单击 **应用程序 > 应用程序类型 > 企业应用程序** 来卸载 IBM WebSphere MQ File Transfer Edition 数据库记录器应用程序。选择该数据库记录器应用程序, 然后单击 **卸载**。
5. 可选: 如果正在使用多个安装迁移至 IBM WebSphere MQ 7.5 或更高版本, 并且本机库路径有所不同, 请通过单击 **资源 > JMS 提供程序 > WebSphere MQ 消息传递提供程序** 来更改此路径
例如, 如果本机库路径为: `C:\Program Files\IBM\WebSphere MQ\java\lib`, 请将路径更改为: `C:\Program Files\IBM\New MQ Installation Location\java\lib`
6. 可选: 如果使用多个安装来迁移至 IBM WebSphere MQ 7.5 或更高版本, 那么必须使用 `setmqm` 命令将队列管理器与新安装关联。
7. 使用第 171 页的 [『将 Java EE 数据库记录器与 WebSphere Application Server V7.0 一起安装』](#) 中的信息以及先前在步骤 [第 180 页的『3』](#) 中记录的信息来重新安装数据库记录器应用程序。
8. 通过单击 **应用程序 > 应用程序类型 > 企业应用程序** 来启动此新数据库记录器。选择此数据库记录器应用程序, 然后单击 **启动**。

9. 要验证迁移，请检查数据库以确保条目在被写入。

配置 Web 网关

您必须配置 IBM MQ Managed File Transfer Web 网关以使用现有的 IBM MQ Managed File Transfer 环境。配置过程特定于您正在使用的应用程序服务器。在配置 Web 网关之前，请在与应用程序服务器相同的系统上创建 Web 代理。

开始之前

在配置或使用 Web 网关之前，请参阅第 305 页的『Web 网关的方案』和第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』。这些主题解释了 Web 网关的用途和组件。

相关任务

第 182 页的『准备部署 Web 网关』

在部署 IBM MQ Managed File Transfer Web 网关之前，必须设置应用程序服务器环境和相关模块。本部分描述了 IBM MQ 和两个不同应用程序服务器的设置任务。

第 197 页的『部署 IBM MQ Managed File Transfer Web 网关』

必须将 IBM MQ Managed File Transfer Web 网关部署到与 Java Platform, Enterprise Edition 5 兼容的应用程序服务器。不同应用程序服务器的部署过程各不相同。本部分概述了两个应用程序服务器的部署过程。

第 181 页的『设置数据库以用于文件空间』

在可以使用文件空间之前，必须为 Web 网关设置数据库表以存储文件空间信息。您可以在现有的日志数据库中创建这些表，或创建新数据库以包含这些表。

第 200 页的『配置数据库记录器以用于 Web 网关』

以下示例显示当数据库记录器未正确配置时，请求传输状态的结果：

第 201 页的『验证 Web 网关安装』

遵循以下指示信息来检查是否已正确部署 IBM MQ Managed File Transfer Web 网关应用程序。

相关参考

第 516 页的『fteCreateWebAgent（创建 IBM MQ Managed File Transfer Web 代理）』

fteCreateWebAgent 命令创建代理及其关联的配置，以用于 Web 网关。IBM MQ Managed File Transfer 服务器随附该命令。

设置数据库以用于文件空间

在可以使用文件空间之前，必须为 Web 网关设置数据库表以存储文件空间信息。您可以在现有的日志数据库中创建这些表，或创建新数据库以包含这些表。

关于此任务

遵循这些指示信息来创建 Web 网关使用文件空间时所需的数据库表。

过程

1. 如果未安装数据库软件，请使用针对您数据库的文档来安装数据库软件。如果 JDBC 支持是数据库的可选组件，那么必须安装此组件。
2. 如果您没有数据库或要使用数据库记录器所使用的数据库以外的另一个数据库，请使用数据库工具创建数据库。

缺省模式名称为 FTEWEB。如果使用 FTEWEB 以外的模式名称，那么必须编辑提供的 SQL 文件 `webgateway_db2.sql`、`webgateway_oracle.sql` 或 `webgateway_zos.sql`，以在继续下一步之前反映这一点。如果要在与数据库记录器表相同的数据库中创建 Web 网关表，那么两个表集不能具有相同的模式名称。

3. 使用数据库工具创建必需的数据库表。

文件 `webgateway_db2.sql`、`webgateway_oracle.sql` 或 `webgateway_zos.sql` 包含可运行以创建表的 SQL 命令。这些文件位于 IBM MQ Managed File Transfer 服务安装的 `MQ_INSTALLATION_PATH/mqft/web/sql` 目录中。

文件开头的注释中描述了有关如何使用和定制 SQL 命令的信息。

注: 如果要从 WebSphere MQ File Transfer Edition V7.0.x 迁移到较高版本的 IBM MQ Managed File Transfer, 那么无需对 Web 网关的数据库模式进行任何更改。不存在要针对您的数据库运行的 SQL 迁移文件。

相关任务

第 200 页的『配置数据库记录器以用于 Web 网关』

以下示例显示当数据库记录器未正确配置时, 请求传输状态的结果:

相关参考

第 948 页的『Web 网关所使用的数据库表』

IBM MQ Managed File Transfer Web 网关使用以下数据库表配置用户文件空间并保证用户文件空间的安全。

更改 Web 网关中的模式名称

Web 网关可以使用具有非缺省模式名称的数据库。 您必须更改 Web 网关 EAR 文件中的模式名称。

关于此任务

缺省模式名称为 FTEWEB。 要更改 Web 网关所使用模式的名称, 请完成以下步骤:

过程

1. 使用以下命令解压缩 JAR 文件:

```
jar -xvf com.ibm.wmqfte.web.ear lib/com.ibm.wmqfte.web.jpa.fs.jar
```

可在 `<product_install_location>/mqft/web/com.ibm.wmqfte.web.ear` 中找到此 JAR 文件。

2. 通过使用以下命令, 从 JPA JAR 文件中抽取 persistence.xml 文件:

```
jar -xvf lib/com.ibm.wmqfte.web.jpa.fs.jar META-INF/persistence.xml
```

3. 编辑 META-INF/persistence.xml 文件, 更改以下行:

```
<property name="openjpa.jdbc.Schema" value="schema_name" />
```

其中:

- `schema_name` 是您所选的模式名称。缺省模式名称为 FTEWEB

4. 通过使用以下命令, 以修改后的 persistence.xml 文件更新 JPA JAR:

```
jar -uvf lib/com.ibm.wmqfte.web.jpa.fs.jar META-INF/persistence.xml
```

5. 通过使用以下命令, 以修改后的 JPA JAR 文件更新 EAR 文件:

```
jar -uvf com.ibm.wmqfte.web.ear lib/com.ibm.wmqfte.web.jpa.fs.jar
```

准备部署 Web 网关

在部署 IBM MQ Managed File Transfer Web 网关之前, 必须设置应用程序服务器环境和相关模块。 本部分描述了 IBM MQ 和两个不同应用程序服务器的设置任务。

开始之前

在配置或使用 Web 网关之前, 请参阅第 305 页的『Web 网关的方案』和 第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』。 这些主题解释了 Web 网关的用途和组件。

在部署 Web 网关应用程序之前，必须完成所需的安全步骤。有关更多信息，请参阅第 106 页的『Web 网关的必需安全性』。

要完成 Web 网关拓扑，还需要使用 Web 代理和数据库记录器。有关更多信息，请参阅第 516 页的『fteCreateWebAgent（创建 IBM MQ Managed File Transfer Web 代理）』和第 151 页的『配置 Managed File Transfer 记录器』。

相关任务

第 193 页的『准备使用 WebSphere Application Server V7.0 部署 Web 网关』

在将 IBM MQ Managed File Transfer Web 网关企业应用程序部署到 WebSphere Application Server V7.0 之前，请使用这些指示信息定义必需资源。必须针对您的环境定制示例部署计划。

第 183 页的『准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关』

在将 IBM MQ Managed File Transfer 服务 Web 网关企业应用程序部署到 WebSphere 应用程序服务器 Community Edition 之前，请使用这些指示信息设置您的环境。定制针对您环境的示例部署计划。

准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关

在将 IBM MQ Managed File Transfer 服务 Web 网关企业应用程序部署到 WebSphere 应用程序服务器 Community Edition 之前，请使用这些指示信息设置您的环境。定制针对您环境的示例部署计划。

开始之前

在配置或使用 Web 网关之前，请参阅第 305 页的『Web 网关的方案』和第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』。这些主题解释了 Web 网关的用途和组件。

要检查是否支持您正在使用的应用程序服务器版本与 Web 网关一起使用，请参阅 Web 页面 [IBM MQ System Requirements](#)。

注：运行应用程序服务器所使用的用户必须与 Web 代理运行所使用的用户相同或在同一个组中。

在启动应用程序服务器设置之前，请完成以下任务，准备您的 IBM MQ 环境以使用 Web 网关。

确定应用程序服务器用于连接 IBM MQ 的用户标识。在您的 IBM MQ 环境中，必须为该用户标识提供 **Set identity context** 许可权。例如，如果应用程序服务器作为 appuser1（是组 appgrp 的成员）运行，并且使用绑定方式连接到名为 qm1 的本地 IBM MQ 队列管理器，那么请运行以下命令：

```
setmqaut -m qm1 -g appgrp +setid -t qmgr
```

还必须在 Web 代理命令队列上为该用户标识提供 **Set identity context** 许可权。例如，如果应用程序服务器作为 appuser1（是组 appgrp 的成员）运行，Web 代理名为 WEBAGENT，并且应用程序服务器使用绑定方式连接到名为 qm2 的本地 IBM MQ 队列管理器，那么请运行以下命令：

```
setmqaut -m qm2 -g appgrp +setid -t queue -n SYSTEM.FTE.COMMAND.WEBAGENT
```

关于此任务

可以从以下 Web 页面获取 WebSphere 应用程序服务器 Community Edition：<https://www.ibm.com/software/webservers/appserv/community>

在部署 Web 网关应用程序之前，必须设置依赖的组件。这些组件包括 IBM MQ 资源适配器、IBM MQ Managed File Transfer 数据库记录器写入的数据库、数据库连接器以及安全域。还必须针对您的环境更新 web.xml 文件和部署计划。

Web 网关还要求 IBM MQ Managed File Transfer Web 代理与该应用程序安装在相同的系统上，并且以与应用程序服务器相同的用户或同一个组中的用户身份运行该代理。要获取有关如何创建和配置该代理的指示信息，请参阅第 516 页的『fteCreateWebAgent（创建 IBM MQ Managed File Transfer Web 代理）』。

过程

1. 部署 IBM MQ 资源适配器。

如果您的 WebSphere 应用程序服务器 Community Edition 实例与要连接到的 IBM MQ 队列管理器在同一系统上运行，请参阅第 184 页的『在与应用程序服务器相同的系统上部署 IBM MQ 资源适配器』。如果您的 WebSphere 应用程序服务器 Community Edition 实例与要连接到的 IBM MQ 队列管理器在不同的系统上运行，请参阅第 185 页的『在与应用程序服务器不同的系统上部署 IBM MQ 资源适配器』。

2. 定义数据库连接器以连接到日志数据库。
有关更多信息，请参阅第 186 页的『定义数据库连接器以连接到日志数据库』。
3. 定义数据库连接器以连接到文件空间数据库。
有关更多信息，请参阅第 187 页的『定义数据库连接器以连接到文件空间数据库』。
4. 定义安全域。
有关更多信息，请参阅第 187 页的『定义安全域』。
5. 更新 web.xml 文件。
有关更多信息，请参阅第 188 页的『更新 web.xml 文件』。
6. 更新 openejb-jar.xml 文件。
有关更多信息，请参阅第 190 页的『更新 openejb-jar.xml 以将 Web 网关配置为使用文件空间』。
7. 如果必须在非缺省环境中部署 Web 网关或正在使用自己的安全域，那么必须更新提供的部署计划或提供单独的部署计划。
有关更多信息，请参阅第 190 页的『更新部署计划』。
8. 可选：如果要在非缺省环境中部署 Web 网关管理控制台，请更新 com.ibm.wmqfte.web.admin.war 文件中提供的部署计划。
有关更多信息，请参阅第 192 页的『针对管理控制台更新部署计划』。

结果

您现在可以将 Web 网关 EAR 文件部署到应用程序服务器。执行主题第 197 页的『使用 WebSphere Application Server Community Edition 部署 Web 网关』中的步骤。

在与应用程序服务器相同的系统上部署 IBM MQ 资源适配器

关于此任务

如果 WebSphere 应用程序服务器 Community Edition 实例与要连接到的 IBM MQ 队列管理器运行在同一系统上，请执行以下步骤来部署 IBM MQ 资源适配器。

过程

1. 创建定义到源代理队列管理器的连接的计划文件。以下示例计划文件定义到名为 QM_JUPITER 的队列管理器的连接。

```
<?xml version="1.0" encoding="UTF-8"?>
<connector xmlns="http://geronimo.apache.org/xml/ns/j2ee/connector">
  <resourceadapter>
    <resourceadapter-instance>
      <resourceadapter-name>WMQ</resourceadapter-name>
      <workmanager>
        <gbean-link>DefaultWorkManager</gbean-link>
      </workmanager>
    </resourceadapter-instance>
    <outbound-resourceadapter>
      <connection-definition>
        <connectionfactory-interface>
          javax.jms.ConnectionFactory
        </connectionfactory-interface>
        <connectiondefinition-instance>
          <name>jms/WMQFTEWebAgentConnectionFactory</name>
          <config-property-setting name="queueManager">
            QM_JUPITER
          </config-property-setting>
          <config-property-setting name="transportType">
            BINDINGS
          </config-property-setting>
          <connectionmanager>
            <no-transaction />
            <no-pool/>
          </connectionmanager>
        </connectiondefinition-instance>
      </outbound-resourceadapter>
    </resourceadapter>
  </connector>
```

```

        </connectionmanager>
    </connectiondefinition-instance>
</connection-definition>
</outbound-resourceadapter>
</resourceadapter>
</connector>

```

要在您的环境中使用该计划文件，请将 QM_JUPITER 更改为源代理的队列管理器的名称。必须编辑的 XML 文件部分以**粗体**字型突出显示。

2. 打开 WebSphere 应用程序服务器 CE 管理控制台。
3. 从“欢迎页面”上的公共控制台操作列表中，单击**部署新应用程序 > 新部署**。
4. 在**归档**字段中，输入 `mq-install-root/java/lib/jca/wmq.jmsra.rar`
5. 在**计划**字段中，输入您在步骤 1 中创建的计划文件的路径。
6. 可选：如果接收到以下错误: HTTP Status 403 - The request body was too large to be cached during the authentication process, 那么必须增大最大发布大小。在 WebSphere 应用程序服务器 CE 管理控制台上，单击**服务器 > Web 服务器 > Tomcat Web 连接器 > 编辑**，然后将 **maxPostSize** 的值更改为 -1（无限制）。

下一步做什么

接下来，定义数据库连接器以连接到日志数据库。有关更多信息，请参阅 [第 186 页的『定义数据库连接器以连接到日志数据库』](#)。

在与应用程序服务器不同的系统上部署 IBM MQ 资源适配器

关于此任务

如果 WebSphere 应用程序服务器 Community Edition 实例与要连接到的 IBM MQ 队列管理器运行在不同系统上，请执行以下步骤来部署 IBM MQ 资源适配器

过程

1. 创建定义到源代理队列管理器的连接的计划文件。以下示例计划文件定义到队列管理器 QM_SATURN 的连接，该队列管理器位于与 WebSphere 应用程序服务器 Community Edition 安装不同的系统上。QM_SATURN 的主机名为 `saturn.example.com`。QM_SATURN 的端口为 1415。QM_SATURN 的通道为 `SYSTEM.DEF.SVRCONN`。

```

<?xml version="1.0" encoding="UTF-8"?>
<connector xmlns="http://geronimo.apache.org/xml/ns/j2ee/connector">
  <resourceadapter>
    <resourceadapter-instance>
      <resourceadapter-name>WMQ</resourceadapter-name>
      <workmanager>
        <gbean-link>DefaultWorkManager</gbean-link>
      </workmanager>
    </resourceadapter-instance>
    <outbound-resourceadapter>
      <connection-definition>
        <connectionfactory-interface>
          javax.jms.ConnectionFactory
        </connectionfactory-interface>
        <connectiondefinition-instance>
          <name>jms/WMQFTEWebAgentConnectionFactory</name>
          <config-property-setting name="channel">
            SYSTEM.DEF.SVRCONN
          </config-property-setting>
          <config-property-setting name="queueManager">
            QM_SATURN
          </config-property-setting>
          <config-property-setting name="hostName">
            saturn.example.com
          </config-property-setting>
          <config-property-setting name="port">
            1415
          </config-property-setting>
          <config-property-setting name="transportType">
            CLIENT
          </config-property-setting>
        </connectiondefinition-instance>
      </outbound-resourceadapter>
    </resourceadapter>
  </connector>

```

```

        <no-transaction />
        <no-pool/>
    </connectionmanager>
</connectiondefinition-instance>
</connection-definition>
</outbound-resourceadapter>
</resourceadapter>
</connector>

```

要在您的环境中使用该计划文件，请将 **QM_SATURN** 更改为源代理的队列管理器的名称。将主机名、端口和通道的值更改为源代理的队列管理器对应的值。必须编辑的 XML 文件部分以**粗体**字型突出显示。

2. 将文件 `mq-install-root/java/lib/jca/wmq.jmsra.rar` 从 IBM MQ 安装所在的系统复制到 WebSphere 应用程序服务器 Community Edition 安装所在的系统。
3. 打开 WebSphere 应用程序服务器 Community Edition 管理控制台。
4. 从“欢迎页面”上的**公共控制台操作**列表中，单击**部署新应用程序 > 新部署**。
5. 在**归档**字段中，输入您获取的 `wmq.jmsra.rar` 文件副本的路径。
6. 在**计划**字段中，输入您创建的计划文件路径。

下一步做什么

接下来，定义数据库连接器以连接到日志数据库。有关更多信息，请参阅 [第 186 页的『定义数据库连接器以连接到日志数据库』](#)。

定义数据库连接器以连接到日志数据库

开始之前

要获取传输状态信息，Web 网关应用程序需要 IBM MQ Managed File Transfer 数据库记录器写入的数据库的访问权。在向该数据库定义数据库连接器之前，必须先设置数据库和数据库记录器。有关如何设置数据库和如何使用数据库记录器应用程序的指示信息，请参阅主题 [第 151 页的『配置 Managed File Transfer 记录器』](#)。

关于此任务

要在 WebSphere 应用程序服务器 Community Edition 环境中访问该数据库，必须定义数据库连接器。要定义数据库连接器，请从 WebSphere 应用程序服务器 Community Edition 管理控制台执行以下步骤：

过程

1. 根据所使用的 WebSphere 应用程序服务器 Community Edition 级别，从**控制台导航**中选择**服务 > 数据库池**，或选择**资源 > 数据源**。
2. 使用 Geronimo 数据库池向导创建数据库池。在**数据库池名称**字段中输入 `jdbc/wmqfte-database`。
3. 对于 **数据库类型**，请选择 **DB2 XA** 或 **Oracle Thin**(针对您的数据库)。
4. 单击**下一步**。
5. 在**驱动程序 jar** 字段中，选择适合于您数据库的 jar 文件。
6. 在**数据库名称**字段中，输入您为获取传输状态信息而要连接到的数据库的名称。
7. 在**用户名**字段中，输入要连接到数据库并对数据库进行认证的用户名。
8. 在**密码**和**确认密码**字段中，输入密码以在数据库中进行认证。
9. 在**服务器名称**字段中，输入数据库驱动程序需要连接到的主机的主机名或 IP 地址。
10. 在**端口号**字段中，输入您使用的端口号（如果不是缺省端口）。
11. 确保**驱动程序类型**的值为 4。
12. 从**事务类型**列表中选择 **XA**。
13. 单击**部署**。

下一步做什么

接下来，定义数据库连接器以连接到文件空间数据库。有关更多信息，请参阅 [第 187 页的『定义数据库连接器以连接到文件空间数据库』](#)。

定义数据库连接器以连接到文件空间数据库

开始之前

在定义该数据库连接器之前，必须创建 Web 网关处理文件空间时所需的数据库和表。有关更多信息，请参阅 [第 181 页的『设置数据库以用于文件空间』](#)。

关于此任务

Web 网关应用程序需要数据库访问权，以存储有关您创建和使用的用户文件空间的信息。该数据库可以是 IBM MQ Managed File Transfer 数据库记录器所使用的同一个数据库，这在 [第 186 页的『定义数据库连接器以连接到日志数据库』](#) 中有所提及。即使对文件空间信息使用同一个数据库，也必须按照以下步骤所述创建第二个数据库连接器。要定义数据库连接器，请从 WebSphere 应用程序服务器 Community Edition 控制台执行以下步骤：

过程

1. 根据所使用的 WebSphere 应用程序服务器 Community Edition 级别，从**控制台导航**中选择**服务 > 数据库池**，或选择**资源 > 数据源**。
2. 使用 Geronimo 数据库池向导创建数据库池。在**数据库池名称**字段中输入 jdbc/wmqfte-filespace。
3. 对于 **数据库类型**，请选择 DB2 XA 或 Oracle Thin(针对您的数据库)。
4. 单击**下一步**。
5. 在**驱动程序 jar** 字段中，选择适合于您数据库的 jar 文件。
6. 在**数据库名称**字段中，输入要连接到的用于文件空间信息的数据库的名称。
7. 在**用户名**字段中，输入要连接到数据库并对数据库进行认证的用户名。
8. 在**密码**和**确认密码**字段中，输入密码以在数据库中进行认证。
9. 在**端口号**字段中，输入您使用的端口号（如果不是缺省端口）。
10. 确保**驱动程序类型**的值为 4。
11. 从**事务类型**列表中选择 XA。
12. 单击**部署**。

下一步做什么

接下来定义安全域。有关更多信息，请参阅 [第 187 页的『定义安全域』](#)。

定义安全域

关于此任务

缺省情况下，对于 Web 网关应用程序，需要名为 **WMQFTESecurityRealm** 的安全域。定义具有名为 *administrators*、*employees* 和 *managers* 的组的域。至少为每个组定义一个用户。要定义安全域，请从 WebSphere 应用程序服务器 Community Edition 管理控制台执行下列操作：

过程

1. 在**控制台导航**中选择**安全 > 安全域**。
2. 在显示的面板上，单击**添加新安全域**。
3. 在**安全域名称**字段中输入 WMQFTESecurityRealm。
4. 针对**域类型**：
 - 如果需要简单设置，请执行以下步骤：
 - a. 创建包含用户和密码信息的文件。每行的格式为 username=password。例如，

```
fteadmin=password1
fteuser=password2
```


- b. 创建包含组信息的文件。每行的格式为 `group=user,user`。例如，

```
administrators=fteadmin  
employees=fteadmin,fteuser  
managers=fteuser
```

- c. 针对域类型，请选择属性文件域，然后单击下一步。
d. 在以下字段中输入必需信息。

用户文件 URI

在步骤 4a 中创建的包含用户和密码信息的属性文件的位置。在所有平台上，都必须将路径分隔符指定为正斜杠 (/) 字符。该文件的路径相对于 WebSphere 应用程序服务器 Community Edition 安装目录。

组文件 URI

在步骤 4b 中创建的包含组信息的属性文件的位置。在所有平台上，都必须将路径分隔符指定为正斜杠 (/) 字符。该文件的路径相对于 WebSphere 应用程序服务器 Community Edition 安装目录。

摘要算法

对密码使用的消息摘要算法。示例值为 MD5 和 SHA1。针对简单设置或如果未使用任何摘要算法，请将该字段留空。

摘要编码

用于摘要算法的编码。示例值为 hex 和 base64。只有当指定了摘要算法时才会使用该值。如果未指定任何编码，请使用 hex。

- e. 单击下一步按钮。将显示高级配置面板。清除复选框。
f. 单击测试登录按钮。在测试登录面板上，为您在用户文件 URI 字段中定义的文件中指定的某个用户输入有效的用户名和密码。单击下一步按钮。
g. 在显示的面板上，单击部署域按钮。
- 如果需要更高级的设置，请参阅 [WebSphere 应用程序服务器 Community Edition 文档](#) 内的信息。

下一步做什么

接下来更新 web.xml 文件。有关更多信息，请参阅 [第 188 页的『更新 web.xml 文件』](#)。

更新 web.xml 文件

关于此任务

使用 Java SDK jar 实用程序为您的环境更新 Web 网关应用程序 web.xml 文件，以完成以下步骤：

过程

1. 运行以下命令，以从提供的 EAR 文件 com.ibm.wmqfte.web.ear 中抽取 Web 网关应用程序：

```
jar -xf com.ibm.wmqfte.web.ear com.ibm.wmqfte.web.war
```

EAR 文件位于 IBM MQ Managed File Transfer 服务安装的 `MQ_INSTALLATION_PATH/mqft/web` 目录中。

2. 运行以下命令，以从先前抽取的 Web 网关应用程序 com.ibm.wmqfte.web.war 中抽取 WEB-INF/web.xml 文件：

```
jar -xf com.ibm.wmqfte.web.war WEB-INF/web.xml
```

3. 使用文本编辑器编辑抽取的 WEB-INF/web.xml。更改以下参数：

agentName

必需。充当 Web 网关所启动传输的源的 Web 代理的名称。必须将该代理安装在与部署 Web 网关应用程序的应用程序服务器相同的系统上，并且以与应用程序服务器相同的用户或同一个组中的用户身

份运行该代理。有关如何创建该代理的信息，请参阅主题：[第 516 页的『fteCreateWebAgent（创建 IBM MQ Managed File Transfer Web 代理）』](#)。

代理名称不区分大小写，并且必须遵循 IBM MQ 对象命名约定。有关更多信息，请参阅[第 699 页的『IBM MQ Managed File Transfer 的对象命名约定』](#)。

coordinationQMgr

必需。Web 网关用于记录传输信息的协调队列管理器的名称。

fileSpaceRoot

可选。Web 网关创建和管理的文件空间的根目录路径。每个文件空间都位于该根目录下与此文件空间同名的子目录中。如果您将该参数的值留空，那么会将应用程序服务器主目录用作缺省文件空间根目录。如果在创建文件空间后更改该参数的值，那么这些文件空间的位置将仍保持不变。

webGatewayName

必需。部署的 Web 网关的名称。

Web 网关的名称不区分大小写，并且必须遵循 IBM MQ Managed File Transfer 对象命名约定。有关更多信息，请参阅[第 699 页的『IBM MQ Managed File Transfer 的对象命名约定』](#)。

tempFileUploadDir

可选。与 Web 网关所启动传输相关的临时文件存储器的目录路径。临时文件的上载目录用于将临时文件上载到 Web 网关时临时存储文件。完成上载到 Web 网关时，Web 代理会将文件从临时文件的上载目录传输到目标代理。如果未提供该参数的值，那么将使用应用程序服务器临时目录（java.io.tmpdir 的值）。

maxTempFileUploadSpace

可选。允许用户存储与 Web 网关启动的传输相关的临时文件的最大空间量 (MB)。用户将文件上载至代理时，会将这些文件临时存储在文件系统，直至将其传输。该参数可用于限制用户在任何一次上载可使用的空间量。如果未提供该参数的值，那么用户可用的临时文件存储空间将不受限制。

defaultMQMDUserID

必须将用户名映射到 MQMD 用户标识。如果未执行此操作，用户将无法使用 Web 网关执行文件传输。有两种方式可将用户映射到 MQMD 用户标识。必须执行下列两个操作或其中一个操作：

- 将该参数设置为缺省 IBM MQ 消息描述符 (MQMD) 用户标识，以在没有为用户定义特定的 MQMD 用户标识时将该缺省用户标识与请求用户关联。
- 使用 Web 网关用户管理 API 在用户和 MQMD 用户标识之间定义映射。

有关在用户和 MQMD 用户标识之间定义映射的更多信息，请参阅[第 942 页的『用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式』](#)、[第 930 页的『Web 网关管理 API 引用』](#)和[第 338 页的『示例：将 Web 用户标识映射到 MQMD 用户标识』](#)。

CSRFProtection

在 Web 网关中启用 CSRF（跨站点伪造请求）保护。当设置为 `true`（区分大小写）时，由 Web 网关处理的任何 POST 或 DELETE 请求必须指定 `x-fte-csrf-token` HTTP 头或“`csrf-token`”表单属性。头或属性的值必须与当前 `JSESSIONID` 的值相匹配。如果您将该参数的值留空或将其设置为任何其他值，那么 Web 网关将不会执行 CSRF 验证并且请求不需要包含 `csrf` 头或表单属性。

有关 Web 网关请求中包含正确 CSRF 标记的更多信息，请参阅[第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』](#)。

4. 运行以下命令，以使用已修改的 `WEB-INF/web.xml` 来更新 Web 网关应用程序：

```
jar -uf com.ibm.wmqfte.web.war WEB-INF/web.xml
```

5. 运行以下命令，以使用已更新的 Web 网关应用程序来更新提供的 ear 文件：

```
jar -uf com.ibm.wmqfte.web.ear com.ibm.wmqfte.web.war
```

下一步做什么

接下来更新 `openejb-jar.xml` 文件。有关更多信息，请参阅[第 190 页的『更新 openejb-jar.xml 以将 Web 网关配置为使用文件空间』](#)。

更新 *openejb-jar.xml* 以将 Web 网关配置为使用文件空间

关于此任务

如果要使用 Web 网关的文件空间功能，请针对您的环境更新 IBM MQ Managed File Transfer Web 网关应用程序 *openejb-jar.xml* 文件。使用 Java SDK jar 实用程序来完成以下步骤：

过程

1. 运行以下命令，以从提供的 EAR 文件 *com.ibm.wmqfte.web.ear* 中抽取 EJB jar 文件：

```
jar -xf com.ibm.wmqfte.web.ear com.ibm.wmqfte.web.mdb.jar
```

EAR 文件位于 IBM MQ Managed File Transfer 服务 安装的 *MQ_INSTALLATION_PATH/mqft/web* 目录中。

2. 运行以下命令，以从先前抽取的 EJB jar 文件 *com.ibm.wmqfte.web.mdb.jar* 中抽取 *META-INF/openejb-jar.xml* 文件：

```
jar -xf com.ibm.wmqfte.web.mdb.jar META-INF/openejb-jar.xml
```

3. 使用文本编辑器编辑抽取的 *META-INF/openejb-jar.xml* 文件。将以下 *activation-config-property* 值更改为与您的环境相匹配：

queueManager

Web 代理使用的 IBM MQ 队列管理器的名称。

hostName

用于连接到指定的 IBM MQ 队列管理器的主机名。

transportType

用于与指定的 IBM MQ 队列管理器进行通信的连接方法。该属性的值可以是 *CLIENT* 或 *BINDINGS*。

port

用于连接到指定的 IBM MQ 队列管理器的端口。只有当 *transportType* 设置为 *CLIENT* 时才需要该属性。

通道

用于连接到指定的 IBM MQ 队列管理器的服务器通道。只有当 *transportType* 设置为 *CLIENT* 时才需要该属性。

目的地

Web 网关使用的 IBM MQ Managed File Transfer Web 网关队列的名称。例如，如果 Web 网关名为 *JUPITER.GATEWAY*，那么请将该属性设置为 *SYSTEM.FTE.WEB.JUPITER.GATEWAY*。

4. 通过运行以下命令，使用修改的 *META-INF/openejb-jar.xml* 文件更新 EJB jar 文件：

```
jar -uf com.ibm.wmqfte.web.mdb.jar META-INF/openejb-jar.xml
```

5. 通过运行以下命令，使用更新的 EJB jar 文件更新提供的 EAR 文件：

```
jar -uf com.ibm.wmqfte.web.ear com.ibm.wmqfte.web.mdb.jar
```

下一步做什么

接下来更新部署计划。有关更多信息，请参阅 [第 190 页的『更新部署计划』](#)。

更新部署计划

关于此任务

如果必须对非缺省环境部署 Web 网关或正在使用自己的安全域，那么必须更新提供的部署计划或提供单独的部署计划。提供的部署计划位于 Web 网关应用程序文件 *com.ibm.wmqfte.web.war* 的文件 *WEB-*

INF/geronimo-web.xml 中。使用 Java SDK jar 实用程序更新为您的环境提供的部署计划，以完成以下步骤：

过程

1. 运行以下命令，以从提供的 EAR 文件 com.ibm.wmqfte.web.ear 中抽取 Web 网关应用程序：

```
jar -xf com.ibm.wmqfte.web.ear com.ibm.wmqfte.web.war
```

EAR 文件位于 IBM MQ Managed File Transfer 服务 安装的 `MQ_INSTALLATION_PATH/mqft/web` 目录中。

2. 运行以下命令，以从先前抽取的 Web 网关应用程序 com.ibm.wmqfte.web.war 中抽取 WEB-INF/geronimo-web.xml 文件：

```
jar -xf com.ibm.wmqfte.web.war WEB-INF/geronimo-web.xml
```

3. 使用文本编辑器编辑抽取的 WEB-INF/geronimo-web.xml。

以下示例部署计划显示了 WebSphere 应用程序服务器 Community Edition 的样本安全性配置：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Licensed Materials - Property of IBM Copyright IBM Corp. 2010, 2023. All Rights Reserved.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract
with IBM Corp. -->
<web:web-app xmlns:app="http://geronimo.apache.org/xml/ns/j2ee/application-2.0"
  xmlns:client="http://geronimo.apache.org/xml/ns/j2ee/application-client-2.0"
  xmlns:conn="http://geronimo.apache.org/xml/ns/j2ee/connector-1.2"
  xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.2"
  xmlns:ejb="http://openejb.apache.org/xml/ns/openejb-jar-2.2"
  xmlns:name="http://geronimo.apache.org/xml/ns/naming-1.2"
  xmlns:pers="http://java.sun.com/xml/ns/persistence"
  xmlns:pkgen="http://openejb.apache.org/xml/ns/pkgen-2.1"
  xmlns:sec="http://geronimo.apache.org/xml/ns/security-2.0"
  xmlns:web="http://geronimo.apache.org/xml/ns/j2ee/web-2.0.1">
  <dep:environment>
    <dep:moduleId>
      <dep:groupId>ibm</dep:groupId>
      <dep:artifactId>com.ibm.wmqfte.web.war</dep:artifactId>
      <dep:version>7.5</dep:version>
      <dep:type>car</dep:type>
    </dep:moduleId>
    <dep:dependencies>
      <dep:dependency>
        <dep:artifactId>wmq.jmsra.rar</dep:artifactId>
        <dep:type>rar</dep:type>
      </dep:dependency>
    </dep:dependencies>
  </dep:environment>
  <web:context-root>/wmq</web:context-root>
  <!-- Sample security configuration for WAS CE deployment -->
  <!-- With the following settings, WAS must be configured as follows: -->
  <!-- 1 - A security realm must be defined called 'WMQFTESecurityRealm' -->
  <!-- 2 - For each group add a <sec:principal> element into each <sec:role> -->
  <!-- for the roles required for that group. For example: -->
  <!-- <sec:principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal" name="[Group
Name]"/> -->
  <web:security-realm-name>WMQFTESecurityRealm</web:security-realm-name>
  <sec:security>
    <sec:role-mappings>
      <sec:role role-name="wmqfte-admin">
        <!-- Add groups here that are to have the highest administration roles -->

        <sec:principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal"
          name="administrators"/>
      </sec:role>
      <sec:role role-name="wmqfte-filespace-create">
        <!-- Add groups here that are to have the ability to create a file space -->

        <sec:principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal"
          name="managers"/>
        <sec:principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal"
          name="administrators"/>
      </sec:role>
      <sec:role role-name="wmqfte-filespace-modify">
        <!-- Add groups here that are to have the ability to modify properties of a file space -->

        <sec:principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal"
          name="managers"/>
        <sec:principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal"
          name="administrators"/>
      </sec:role>
      <sec:role role-name="wmqfte-filespace-permissions">
        <!-- Add groups here that are to have the ability to modify the user permissions of a file space -->
```

```

    <sec:principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal"
        name="managers"/>
    <sec:principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal"
        name="employees"/>
</sec:role>
<sec:role role-name="wmqfte-filespace-delete">
    <!-- Add groups here that are to have the ability to delete a file space -->
    <sec:principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal"
        name="administrators"/>
</sec:role>
<sec:role role-name="wmqfte-agent-upload">
    <!-- Add groups here that are to have the ability to upload a file to a file space -->
    <sec:principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal"
        name="employees"/>
</sec:role>
<sec:role role-name="wmqfte-filespace-user">
    <!-- Add groups here that are to have the ability to view information from a file space -->
    <sec:principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal"
        name="employees"/>
</sec:role>
<sec:role role-name="wmqfte-audit">
    <!-- Add groups here that are to have the ability to view information from the transfer logs --
>
    <sec:principal class="org.apache.geronimo.security.realm.providers.GeronimoGroupPrincipal"
        name="employees"/>
</sec:role>
</sec:role-mappings>
</sec:security>
</web:web-app>

```

将组添加到以**粗体**字型突出显示的 XML 文件部分，从而为组提供执行某些操作的许可权。有关 Web 网关角色的更多信息，请参阅主题 [第 108 页的『Web 网关的用户角色』](#)。

如果要使用自己的安全域更新部署计划 `web:security-realm-name` 元素以引用该域，请更新引用针对该域定义的组名的角色。

4. 可选：如果要对您的 Web 网关使用非缺省上下文根，可以编辑 `WEB-INF/geronimo-web.xml` 文件中的 `<web:context-root>` 元素。
5. 运行以下命令，以使用已修改的 `WEB-INF/geronimo-web.xml` 来更新 Web 网关应用程序：

```
jar -uf com.ibm.wmqfte.web.war WEB-INF/geronimo-web.xml
```

6. 运行以下命令，以使用已更新的 Web 网关应用程序来更新提供的 ear 文件：

```
jar -uf com.ibm.wmqfte.web.ear com.ibm.wmqfte.web.war
```

下一步做什么

接下来，如果要使用非缺省上下文根，请更新该部署计划。有关更多信息，请参阅 [第 192 页的『针对管理控制台更新部署计划』](#)。

针对管理控制台更新部署计划

关于此任务

您可以使用 Web 网关管理控制台，通过 Web 浏览器来管理文件空间和用户映射。有关更多信息，请参阅 [第 327 页的『Web 网关管理控制台』](#)。

如果要使用非缺省上下文根来部署 Web 网关管理控制台，那么必须更新提供的部署计划以包含非缺省上下文根。提供的部署计划位于管理控制台应用程序文件 `com.ibm.wmqfte.web.admin.war` 的文件 `WEB-INF/geronimo-web.xml` 中。使用 Java SDK `jar` 实用程序更新为您的环境提供的部署计划，以完成以下步骤：

过程

1. 运行以下命令，以从提供的 EAR 文件 `com.ibm.wmqfte.web.ear` 中抽取管理控制台应用程序：

```
jar -xf com.ibm.wmqfte.web.ear com.ibm.wmqfte.web.admin.war
```

EAR 文件位于 IBM MQ Managed File Transfer 服务 安装的 `MQ_INSTALLATION_PATH/mqft/web` 目录中。

2. 运行以下命令，以从先前抽取的管理控制台应用程序 `com.ibm.wmqfte.web.admin.war` 中抽取 `WEB-INF/geronimo-web.xml` 文件：

```
jar -xf com.ibm.wmqfte.web.admin.war WEB-INF/geronimo-web.xml
```

3. 使用文本编辑器编辑抽取的 `WEB-INF/geronimo-web.xml`。

以下示例部署计划显示了 WebSphere 应用程序服务器 Community Edition 的样本安全性配置：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Licensed Materials - Property of IBM Copyright IBM Corp. 2010, 2023. All Rights Reserved.
      US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract
      with IBM Corp. -->
<web:web-app xmlns:app="http://geronimo.apache.org/xml/ns/j2ee/application-2.0"
  xmlns:client="http://geronimo.apache.org/xml/ns/j2ee/application-client-2.0"
  xmlns:conn="http://geronimo.apache.org/xml/ns/j2ee/connector-1.2"
  xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.2"
  xmlns:ejb="http://openejb.apache.org/xml/ns/openejb-jar-2.2"
  xmlns:name="http://geronimo.apache.org/xml/ns/naming-1.2"
  xmlns:pers="http://java.sun.com/xml/ns/persistence"
  xmlns:pkgen="http://openejb.apache.org/xml/ns/pkgen-2.1"
  xmlns:sec="http://geronimo.apache.org/xml/ns/security-2.0"
  xmlns:web="http://geronimo.apache.org/xml/ns/j2ee/web-2.0.1">
  <dep:environment>
    <dep:moduleId>
      <dep:groupId>ibm</dep:groupId>
      <dep:artifactId>com.ibm.wmqfte.web.admin.war</dep:artifactId>
      <dep:version>7.0.3.0</dep:version>
      <dep:type>car</dep:type>
    </dep:moduleId>
  </dep:environment>
  <web:context-root>/wmqfteconsole</web:context-root>
  <web:security-realm-name>WMQFTESecurityRealm</web:security-realm-name>
</web:web-app>
```

编辑以**粗体**字型突出显示的 XML 文件中的文本，以更改管理控制台的上下文根。

4. 运行以下命令，以使用已修改的 `WEB-INF/geronimo-web.xml` 来更新 Web 网关应用程序：

```
jar -uf com.ibm.wmqfte.web.admin.war WEB-INF/geronimo-web.xml
```

5. 运行以下命令，以使用已更新的 Web 网关应用程序来更新提供的 ear 文件：

```
jar -uf com.ibm.wmqfte.web.ear com.ibm.wmqfte.web.admin.war
```

准备使用 *WebSphere Application Server V7.0* 部署 Web 网关

在将 IBM MQ Managed File Transfer Web 网关企业应用程序部署到 WebSphere Application Server V7.0 之前，请使用这些指示信息定义必需资源。必须针对您的环境定制示例部署计划。

开始之前

在配置或使用 Web 网关之前，请参阅第 305 页的『Web 网关的方案』和第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』。这些主题解释了 Web 网关的用途和组件。

要检查是否支持您正在使用的应用程序服务器版本与 Web 网关一起使用，请参阅 Web 页面 [IBM MQ System Requirements](#)。

注：运行应用程序服务器所使用的用户必须与 Web 代理运行所使用的用户相同或在同一个组中。

在启动应用程序服务器设置之前，请完成以下任务，准备您的 IBM MQ 环境以使用 Web 网关。

确定应用程序服务器用于连接 IBM MQ 的用户标识。在您的 IBM MQ 环境中，必须为该用户标识提供 **Set identity context** 许可权。例如，如果应用程序服务器作为 `appuser1`（是组 `appgrp` 的成员）运行，并且使用绑定方式连接到名为 `qm1` 的本地 IBM MQ 队列管理器，那么请运行以下命令：

```
setmqaut -m qm1 -g appgrp +setid -t qmgr
```

还必须在 Web 代理命令队列上为该用户标识提供 **Set identity context** 许可权。例如，如果应用程序服务器作为 appuser1（是组 appgrp 的成员）运行，Web 代理名为 WEBAGENT，并且应用程序服务器使用绑定方式连接到名为 qm2 的本地 IBM MQ 队列管理器，那么请运行以下命令：

```
setmqaut -m qm2 -g appgrp +setid -t queue -n SYSTEM.FTE.COMMAND.WEBAGENT
```

关于此任务

在部署 Web 网关应用程序之前，必须执行以下任务以设置应用程序服务器环境。要获取传输状态信息，Web 网关应用程序需要 IBM MQ Managed File Transfer 数据库记录器写入的数据库的访问权。有关如何设置数据库和如何使用数据库记录器应用程序的指示信息，请参阅第 151 页的『配置 Managed File Transfer 记录器』。要从 WebSphere Application Server 7.0 环境中访问此数据库，必须定义 Java 数据库连接 (JDBC) 提供程序和数据源。

Web 网关还要求 IBM MQ Managed File Transfer Web 代理与该应用程序安装在相同的系统上，并且以与应用程序服务器相同的用户或同一个组中的用户身份运行该代理。要获取有关如何创建和配置该代理的指示信息，请参阅第 516 页的『fteCreateWebAgent（创建 IBM MQ Managed File Transfer Web 代理）』。

注：在执行以下步骤期间，WebSphere Application Server V7.0 管理控制台会多次提示您保存配置。当出现提示时，请保存您的配置。

过程

1. 如果计划以绑定方式将 Web 网关或 Web 代理连接到队列管理器，那么必须设置本机库路径。
有关如何在 WebSphere Application Server V7.0 中设置本机库路径的信息，请参阅第 196 页的『在 WebSphere Application Server V7.0 中设置本机库路径』。
2. 在 WebSphere Application Server V7.0 中启用正确的安全性级别。
为此，请执行以下步骤：
 - a) 选择**安全性 > 全局安全性**。
 - b) 确保选中了**启用管理安全**。
 - c) 确保选中了**启用应用程序安全**。
 - d) 确保未选中使用 **Java 2 安全性**来限制应用程序对本地资源的访问。
 - e) 单击**应用**。
3. 定义 JNDI 队列连接工厂：
 - a) 从 WebSphere Application Server V7.0 管理控制台导航中选择**资源 > JMS > 队列连接工厂**。
 - b) 选择**作用域**下拉列表，并将作用域更改为适当值。例如，Node=yourNode，Server=yourServer。
 - c) 单击**新建**，以使用控制台向导创建队列连接工厂。
 - d) 选择 **WebSphere MQ 消息传递提供程序**，然后单击**确定**。
 - e) 在向导的步骤 1 中，在**名称**字段中输入 WMQFTEWebAgentConnectionFactory，并在 **JNDI 名称** 字段中输入 jms/WMQFTEWebAgentConnectionFactory。单击**下一步**。
 - f) 在向导的步骤 2 中，选择**将所有必填信息输入到此向导中**，然后单击**下一步**。
 - g) 在向导的步骤 2.1 中，在**队列管理器或队列共享组名**字段中，输入 Web 网关代理连接到的队列管理器的名称，然后单击**下一步**。
 - h) 在向导的步骤 2.2 中，输入 Web 网关代理连接到的队列管理器的连接详细信息，然后单击**下一步**。
 - i) 在向导的步骤 3 中，单击**测试连接**。单击**下一步**。
 - j) 在向导的步骤 4 中，复审摘要信息，然后单击**完成**。
 - k) 在**队列连接工厂**面板上，选择您创建的资源。
 - l) 在**高级**部分，确保选中**支持分布式两阶段落实协议**复选框。

注：确保完成该步骤，然后再继续。未执行本步骤可能会导致 Web 网关无法正常运行。

4. 定义 JNDI 队列：

- a) 从 WebSphere Application Server V7.0 管理控制台导航中选择**资源 > JMS > 队列**。
 - b) 选择**作用域**下拉列表，并将作用域更改为适当值。例如，Node=yourNode，Server=yourServer。
 - c) 单击**新建**，以使用控制台向导创建队列。
 - d) 选择 **WebSphere MQ 消息传递提供程序**，然后单击**确定**。
 - e) 在向导的步骤 1 中，在**名称**字段中输入 WMQFTEWebAgentRequestQueue。在 **JNDI 名称**字段中输入 jms/WMQFTEWebAgentRequestQueue。在**队列名称**字段中输入 SYSTEM.FTE.WEB.gateway_name。变量 gateway_name 是您选择提供给 Web 网关实例的名称。在**队列管理器或队列共享组名**字段中，输入 Web 网关代理连接到的队列管理器的名称，然后单击**确定**。
5. 定义激活规范：
- a) 从 WebSphere Application Server V7.0 管理控制台导航中选择**资源 > JMS > 激活规范**。
 - b) 选择**作用域**下拉列表，并将作用域更改为适当值。例如，Node=yourNode，Server=yourServer。
 - c) 单击**新建**，以使用控制台向导创建激活规范。
 - d) 选择 **WebSphere MQ 消息传递提供程序**，然后单击**确定**。
 - e) 在向导的步骤 1 中，在**名称**字段中输入 WMQFTEActivationSpec，并在 **JNDI 名称**字段中输入 jms/WMQFTEActivationSpec。单击**下一步**。
 - f) 在向导的步骤 1.1 中，在**目标 JNDI 名称**字段中输入 jms/WMQFTEWebAgentRequestQueue，从**目标类型**下拉列表选择 Queue，然后单击**下一步**。
 - g) 在向导的步骤 2 中，选择**将所有必填信息输入到此向导中**，然后单击**下一步**。
 - h) 在向导的步骤 2.1 中，在**队列管理器或队列共享组名**字段中，输入 Web 网关代理连接到的队列管理器的名称，然后单击**下一步**。
 - i) 在向导的步骤 2.2 中，输入 Web 网关代理连接到的队列管理器的连接详细信息，然后单击**下一步**。
 - j) 在向导的步骤 3 中，单击**测试连接**。单击**下一步**。
 - k) 在向导的步骤 4 中，复审摘要信息，然后单击**完成**。
 - l) 单击刚刚创建的激活规范的名称。在配置选项卡的“**其他属性**”部分中，单击**高级属性**。在“**高级属性**”面板的“**连接使用程序**”部分中，将 1 输入到**最大服务器会话数**字段中。
- 注：**确保完成该步骤，然后再继续。未执行本步骤可能会导致 Web 网关无法正常运行。
6. 定义 JDBC 提供程序。
- 如果您已部署 JEE 数据库记录器，那么已在您所选的作用域定义该数据源。
- a) 从 WebSphere Application Server V7.0 管理控制台导航选择**资源 > JDBC > JDBC 提供程序**。
 - b) 选择**作用域**下拉列表，并将作用域更改为适当值。例如，Node=yourNode，Server=yourServer。
 - c) 单击**新建**，以使用控制台向导创建 JDBC 提供程序。
 - d) 在向导的步骤 1 中，您提供的值取决于正在使用的数据库的类型。
 - 如果您正在使用 Db2，请从**数据库类型**列表中选择 **DB2**，从**提供程序类型**列表中选择 **DB2 通用 JDBC 驱动程序提供程序**，并从**实现类型**列表中选择 **XA 数据源**。单击**下一步**。
 - 如果您正在使用 Oracle，那么请从**数据库类型**列表中选择 **Oracle**，从**提供程序类型**列表中选择 **Oracle JDBC 驱动程序**，并从**实现类型**列表中选择 **XA 数据源**。单击**下一步**。
 - e) 在向导的步骤 2 中，确保正确设置了必需数据库 jar 文件的目录位置。单击**下一步**。
 - f) 单击摘要页面上的**完成**以创建 JDBC 提供程序。
7. 定义数据源，以使 Web 网关应用程序可以检索传输状态信息。
- 如果您已部署 JEE 数据库记录器，那么已在您所选的作用域定义该数据源。
- a) 从 WebSphere Application Server V7.0 管理控制台导航选择 **资源 > JDBC > 数据源**。
 - b) 选择**作用域**下拉列表，并将作用域更改为适当值。例如，Node=yourNode，Server=yourServer。

- c) 单击**新建**，以使用控制台向导创建数据源。
 - d) 在向导的步骤 1 中，在**数据源名称**字段中输入 `wmqfte-database`，并在 **JNDI 名称**字段中输入 `jdbc/wmqfte-database`。单击**下一步**。
 - e) 在向导的步骤 2 中，使用**选择现有的 JDBC 提供程序**下拉列表来选择先前步骤中创建的 JDBC 提供程序。单击**下一步**。
 - f) **Db2**：在向导的步骤 3 中，在**驱动程序类型**字段中输入 4。
 - g) **Db2**：在**数据库名称、服务器名称和端口号**字段中输入所需的详细信息，然后单击**下一步**。
Oracle：在 **URL** 字段中输入必需的连接 URL，并在**数据存储帮助程序类名**字段中选择正确的数据存储帮助程序。
 - h) 在向导的步骤 4 中，如果已对您的数据库配置认证，那么请在各个下拉框中提供必需的**组件受管认证别名**和**容器受管认证别名**，然后单击**下一步**。
 - i) 单击摘要页面上的**完成**以创建数据源。
8. 定义第二个数据源，以使 Web 网关应用程序可以存储有关您创建和使用的用户文件空间的信息：
- a) 创建处理文件空间时所必需的数据库和数据库表。
有关更多信息，请参阅第 181 页的『[设置数据库以用于文件空间](#)』。
 - b) 重复步骤 7a 到 7i，但是对于步骤 7d，请在**数据源名称**字段中输入 `wmqfte-filespace`，并在 **JNDI 名称**字段中输入 `jdbc/wmqfte-filespace`，然后单击**下一步**。
9. 可选：如果您已配置数据库，那么可以验证数据源的配置：
- a) 从 WebSphere Application Server V7.0 管理控制台导航选择 **资源 > JDBC > 数据源**。
 - b) 单击**测试连接**按钮。

结果

您现在可以将 Web 网关 EAR 文件部署到应用程序服务器。执行主题第 198 页的『[在 WebSphere 应用程序服务器 V7.0 中部署 Web 网关](#)』中的步骤。

在 WebSphere Application Server V7.0 中设置本机库路径

如果在 WebSphere Application Server V 7.0 上部署 Web 网关应用程序或 Java Platform, Enterprise Edition 数据库记录器应用程序，并且要在应用程序与 IBM MQ 之间使用绑定方式连接，那么必须使用系统上 IBM MQ 本机库的位置来配置 IBM MQ 消息传递提供程序。

关于此任务

如果不在应用程序服务器中设置本机库路径，那么可能会在 WebSphere Application Server V7.0 系统输出日志中接收到以下错误消息：

```
A connection could not be made to WebSphere MQ for the following reason:  
CC=2;RC=2495;AMQ8568: The native JNI library 'mqjbnd' was not found. [3=mqjbnd]
```

请使用 WebSphere Application Server V7.0 管理控制台完成以下步骤：

过程

1. 在导航窗格中，展开**资源 > JMS > JMS 提供程序**。
2. 为用于创建绑定方式连接的连接工厂或激活规范选择适当范围内的 IBM MQ 消息传递提供程序。
注：在使用时，**Server** 范围的本机路径信息优先于更高范围的本机路径信息，而 **Node** 范围的本机路径信息优先于 **Cell** 范围的本机路径信息。
3. 在“常规属性”下的**本机库路径**字段中，输入包含 IBM MQ 本机库的目录的全名。
例如，在 Linux 上，输入 `/opt/mqm/java/lib`。只输入一个目录名称。
4. 单击**确定**。
5. 重新启动应用程序服务器以刷新配置。
6. 必需：再次重新启动应用程序服务器以装入库。

相关任务

[第 193 页的『准备使用 WebSphere Application Server V7.0 部署 Web 网关』](#)

在将 IBM MQ Managed File Transfer Web 网关企业应用程序部署到 WebSphere Application Server V7.0 之前，请使用这些指示信息定义必需资源。必须针对您的环境定制示例部署计划。

[第 171 页的『将 Java EE 数据库记录器与 WebSphere Application Server V7.0 一起安装』](#)

按照以下指示信息安装并配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器，以便将 Managed File Transfer 与 WebSphere Application Server V7 一起配合使用。

部署 IBM MQ Managed File Transfer Web 网关

必须将 IBM MQ Managed File Transfer Web 网关部署到与 Java Platform, Enterprise Edition 5 兼容的应用程序服务器。不同应用程序服务器的部署过程各不相同。本部分概述了两个应用程序服务器的部署过程。

相关任务

[第 198 页的『在 WebSphere 应用程序服务器 V7.0 中部署 Web 网关』](#)

使用以下指示信息将 Web 网关企业应用程序部署到 WebSphere 应用程序服务器 V7.0 中。

[第 197 页的『使用 WebSphere Application Server Community Edition 部署 Web 网关』](#)

使用这些指示信息将 IBM MQ Managed File Transfer Web 网关企业应用程序部署到 WebSphere Application Server Community Edition。定制针对您环境的示例部署计划。

使用 WebSphere Application Server Community Edition 部署 Web 网关

使用这些指示信息将 IBM MQ Managed File Transfer Web 网关企业应用程序部署到 WebSphere Application Server Community Edition。定制针对您环境的示例部署计划。

开始之前

在配置或使用 Web 网关之前，请参阅 [第 305 页的『Web 网关的方案』](#) 和 [第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)。这些主题解释了 Web 网关的用途和组件。

在部署 Web 网关应用程序之前，必须执行主题 [第 183 页的『准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关』](#) 中所描述的任务。

关于此任务

要将 EAR 文件部署到应用程序服务器，请从 WebSphere Application Server Community Edition 管理控制台执行以下步骤。

过程

1. 在控制台导航中选择应用程序 > 新部署。
2. 在归档字段中，指定 EAR 文件：com.ibm.wmqfte.web.ear
3. 在计划字段中，指定您自己的部署计划文件，或将值留空以选择缺省部署计划 geronimo-web.xml。
4. 确保选择在安装后启动应用程序。
5. 单击安装。将安装并启动 Web 网关应用程序。

结果

现在，您可以开始使用 Web 网关，例如，部署使用 Web 网关的 Web 应用程序以提交文件传输并传输状态请求。要使用 Web 网关随附的样本应用程序，请遵循主题 [第 353 页的『样本 Web 页面』](#) 中的指示信息。

要检查 Web 网关的安装，请使用 Web 网关随附的安装验证应用程序。有关指示信息，请参阅 [第 201 页的『验证 Web 网关安装』](#)。

相关任务

[第 415 页的『在 WebSphere Application Server Community Edition 中启用跟踪』](#)

如果 Web 网关应用程序在 WebSphere Application Server Community Edition 中运行，请遵循以下指示信息来启用 Web 网关应用程序的跟踪。Web 网关应用程序在接收和处理请求时生成跟踪。

在 WebSphere 应用程序服务器 V7.0 中部署 Web 网关

使用以下指示信息将 Web 网关企业应用程序部署到 WebSphere 应用程序服务器 V7.0 中。

开始之前

在部署 Web 网关应用程序之前，必须遵循主题第 193 页的『准备使用 WebSphere Application Server V7.0 部署 Web 网关』中的指示信息来设置应用程序服务器环境。

关于此任务

在配置或使用 Web 网关之前，请参阅第 305 页的『Web 网关的方案』和第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』。这些主题解释了 Web 网关的用途和组件。

过程

1. 从 WebSphere Application Server V7.0 管理控制台中选择 **应用程序 > 新建应用程序**。
2. 从选项列表中选择 **新建企业应用程序**。
3. 在 **准备应用程序安装** 页面上，从 IBM MQ Managed File Transfer Server 安装的 `MQ_INSTALLATION_PATH/mqft/web` 目录中选择 `com.ibm.wmqfte.web.ear` 文件，然后单击 **下一步**。
4. 在以下屏幕上，选择 **详细** 以显示所有安装选项和参数，然后单击 **下一步**。
5. 在步骤 1 - 5 的每一步中单击 **下一步** 以接受缺省值。
6. 在步骤 6 (**初始化 servlet 的参数**) 中，提供以下参数的值：

agentName

充当 Web 网关所启动传输的源的 IBM MQ Managed File Transfer 代理的名称。该代理必须配置为 Web 代理，并且与部署 Web 网关应用程序的应用程序服务器安装在同一个系统上。您必须提供该参数的值。

部署过程不会创建 Web 代理，因此您必须自行创建该代理。有关如何创建 Web 代理的信息，请参阅第 516 页的『`fteCreateWebAgent`（创建 IBM MQ Managed File Transfer Web 代理）』。

coordinationQMGr

Web 网关用于记录传输信息的协调队列管理器的名称。您必须提供该参数的值。

fileSpaceRoot

Web 网关创建和管理的文件空间的根目录路径。每个文件空间都位于该根目录下与此文件空间同名的子目录中。如果在创建文件空间之后更改该参数的值，那么这些文件空间的位置依然不变。如果您将该参数的值留空，那么会将应用程序服务器主目录用作缺省文件空间根目录。

注：请将新的空目录用作您的文件空间根目录。

webGatewayName

部署的 Web 网关的名称。您必须提供该参数的值。

Web 网关的名称不区分大小写，但必须遵守 WebSphere MQ 对象的命名约定。有关更多信息，请参阅第 699 页的『IBM MQ Managed File Transfer 的对象命名约定』。

tempFileUploadDir

Web 网关所启动传输的相关临时文件存储器的目录路径。当文件上传到 Web 网关时，临时文件上传目录用于临时存储这些文件。上传到 Web 网关完成时，Web 代理将文件从临时文件上传目录传输到目标代理。如果未提供该参数的值，那么将使用应用程序服务器临时目录（`java.io.tmpdir` 的值）。

maxTempFileUploadSpace

允许用户存储与 Web 网关启动的传输相关的临时文件的最大空间量 (MB)。用户将文件上传至代理时，会将这些文件临时存储在文件系统，直至将其传输。该参数可用于限制用户在任何一次上传可使用的空间量。如果未提供该参数的值，那么用户可用的临时文件存储空间将不受限制。

defaultMQMDUserID

当没有为用户定义特定 MQMD 用户标识时，用于与请求的用户关联的缺省 WebSphere MQ 消息描述符 (MQMD) 用户标识。您可以使用 MQMFT Web 网关用户管理 API 来定义用户和 MQMD 用户标识之间的映射。如果未提供该参数的值，那么未定义 MQMD 用户标识的用户将无法执行文件上载。

有关定义用户和 MQMD 用户标识之间映射的更多信息，请参阅主题第 930 页的『Web 网关管理 API 引用』和第 338 页的『示例：将 Web 用户标识映射到 MQMD 用户标识』。

CSRFProtection

在 Web 网关中启用 CSRF（跨站点伪造请求）保护。当设置为 true（区分大小写）时，由 Web 网关处理的任何 POST 或 DELETE 请求必须指定 x-fte-csrf-token HTTP 头或“csrf-token”表单属性。头或属性的值必须与当前 JSESSIONID 的值相匹配。如果您将该参数的值留空或将其设置为任何其他值，那么 Web 网关将不会执行 CSRF 验证并且请求不需要包含 csrf 头或表单属性。

有关 Web 网关请求中包含正确 CSRF 标记的更多信息，请参阅第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』。

注：如果需要，您可以在部署之后更改这些值，而无需重新部署应用程序。要更改这些值，请转至应用程序 > 应用程序类型 > WebSphere 企业应用程序 > WebSphere MQ FTE Web 网关 > 初始化 servlet 的参数。

7. 单击下一步。
8. 在步骤 7（为消息驱动的 bean 绑定侦听器）的目标资源 JNDI 名称字段中，输入 `jms/WMQFTEActivationSpec`。单击下一步。
9. 在步骤 8 - 10 的每一步中单击下一步以接受缺省值。
10. 在步骤 11（将资源引用映射到资源）中，执行以下步骤：
 - a) 对于 `javax.jms.QueueConnectionFactory` 部分的每一项，在目标资源 JNDI 名称字段中输入 `jms/WMQFTEWebAgentConnectionFactory`。
 - b) 在 `javax.sql.DataSource` 部分，找到资源引用字段值为 `jdbc/wmqfte-filespace` 的条目。在目标资源 JNDI 名称字段中，输入 `jdbc/wmqfte-filespace`。
 - c) 在 `javax.sql.DataSource` 部分，找到资源引用字段值为 `jdbc/wmqfte-database` 的条目。在目标资源 JNDI 名称字段中，输入 `jdbc/wmqfte-database`。

单击下一步。

11. 在步骤 12 - 13 的每一步中单击下一步以接受缺省值。
12. 在步骤 14（将安全角色映射到用户或组）中，将必需的用户或组映射到企业应用程序中定义的角色。例如：
 - a) 从表中选择 `wmqfte-admin`、`wmqfte-filespace-create`、`wmqfte-filespace-modify` 和 `wmqfte-filespace-delete`。
 - b) 单击映射组。
 - c) 单击搜索。
 - d) 从列表中选择组 `administrators`，然后单击第一个方向按钮。
 - e) 单击确定。
 - f) 从表中选择 `wmqfte-filespace-create`、`wmqfte-filespace-modify` 和 `wmqfte-filespace-permissions`。
 - g) 单击映射组。
 - h) 单击搜索。
 - i) 从列表中选择组 `managers`，然后单击第一个方向按钮。
 - j) 单击确定。
 - k) 从表中选择 `wmqfte-filespace-permissions`、`wmqfte-agent-upload`、`wmqfte-filespace-user` 和 `wmqfte-audit`。
 - l) 单击映射组。
 - m) 单击搜索。
 - n) 从列表中选择组 `employees`，然后单击第一个方向按钮。

- o) 单击**确定**。
- 有关 Web 网关角色的更多信息，请参阅第 108 页的『Web 网关的用户角色』。单击**下一步**。
13. 可选：如果要使用 Web 网关的非缺省上下文根，那么可在步骤 13（映射 **Web** 模块的上下文根）中更改 Web 网关的上下文根。
 14. 可选：如果要使用 Web 网关管理控制台的非缺省上下文根，那么可在步骤 13（映射 **Web** 模块的上下文根）中更改管理控制台的上下文根。
 15. 单击摘要页面上的**完成**以安装企业应用程序。
 16. 您现在可以从 WebSphere Application Server V7.0 管理控制台中启动该应用程序：
 - a) 从控制台导航中，选择**应用程序 > 应用程序类型 > WebSphere 企业应用程序**。
 - b) 选中集合表中 **Web 网关** 企业应用程序的复选框，然后单击**启动**。

结果

现在，您可以开始使用 Web 网关，例如，部署使用 Web 网关的 Web 应用程序以提交文件传输并传输状态请求。要使用 Web 网关随附的样本应用程序，请遵循主题第 353 页的『样本 Web 页面』中的指示信息。

要检查 Web 网关的安装，请使用 Web 网关随附的安装验证应用程序。有关指示信息，请参阅第 201 页的『验证 Web 网关安装』。

相关任务

第 415 页的『在 WebSphere Application Server V7.0 中启用跟踪』

如果 Web 网关应用程序在 WebSphere Application Server V7.0 中运行，请遵循以下指示信息来启用 Web 网关应用程序的跟踪。Web 网关应用程序在接收和处理请求时生成跟踪。

配置数据库记录器以用于 Web 网关

以下示例显示当数据库记录器未正确配置时，请求传输状态的结果：

关于此任务

1. 该 HTTP 请求提交传输查询：

```
GET HTTP/1.1 /transfer/414d51204d554e474f2afed834435bc6edaf323520204cee
Host: example.com
User-Agent: mozilla
```

2. Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 500 Internal Server Error
Server: WAS/6.0
Content-length: 93
Content-type: text/plain

BFGWI0018E: The request could not be completed due to an internal
web application server error.
```

要配置数据库记录器以正确处理请求，请执行以下步骤：

过程

1. 安装 IBM MQ Managed File Transfer 数据库记录器。有关如何安装和配置数据库记录器的更多信息，请参阅第 151 页的『配置 Managed File Transfer 记录器』。
2. 如果已安装 IBM MQ Managed File Transfer 数据库记录器，请确保数据库表是最新的。使用以下目录中提供的 SQL 文件来更新数据库表：
 - 在分布式平台上：*MQ_INSTALLATION_PATH/mqft/sql*
 - 在 z/OS 上：*MQ_INSTALLATION_PATH/mqft/sql*

相关任务

第 169 页的『[安装 Java EE 数据库记录器](#)』

按照以下指示信息安装并配置 Java EE 数据库记录器以供 Managed File Transfer 使用。

第 160 页的『[安装 IBM MQ Managed File Transfer 独立数据库记录器](#)』

完成以下步骤以安装和配置独立数据库记录器。

验证 Web 网关安装

遵循以下指示信息来检查是否已正确部署 IBM MQ Managed File Transfer Web 网关应用程序。

开始之前

在验证 Web 网关配置之前，必须先遵循指示信息来部署 Web 网关应用程序。请参阅第 181 页的『[配置 Web 网关](#)』。

关于此任务

过程

1. 确保已使用具有 `wmqfte-admin` 安全角色的用户标识登录到应用程序服务器环境。有关更多信息，请参阅第 108 页的『[Web 网关的用户角色](#)』。
2. 在 Web 浏览器中，输入以下 URI：

```
http://host/wmqfte/ivt?logdbschema=FTELOG&webdbschema=FTEWEB
```

如果为 Web 网关应用程序定义的上下文根并非缺省值 `wmqfte`，请使用以下 URI：

```
http://host/context_root/ivt?logdbschema=FTELOG&webdbschema=FTEWEB
```

注：在配置 Web 网关期间，需要设置数据库表以存储有关文件空间和传输历史记录的信息。Web 网关安装验证应用程序假定您使用的是数据库模式名称的缺省值。如果定义的数据库模式名称并非缺省值（对于传输历史记录数据库，缺省值为 `FTELOG`；对于文件空间信息数据库，缺省值为 `FTEWEB`），那么必须更改 URI 中指定的模式名称。使用以下查询术语指定数据库模式名称：

logdbschema

传输历史记录数据库的模式名称

webdbschema

文件空间信息数据库的模式名称

例如，如果传输历史记录数据库和文件空间信息数据库的模式名称分别为 `MYLOG` 和 `MYWEB`，那么请使用以下 URI：

```
http://host/wmqfte/ivt?logdbschema=MYLOG&webdbschema=MYWEB
```

有关设置数据库的更多信息，请参阅第 181 页的『[设置数据库以用于文件空间](#)』和第 200 页的『[配置数据库记录器以用于 Web 网关](#)』。

结果

Web 浏览器显示的页面列出 Web 网关安装的配置信息，并显示某些基本 Web 网关功能的测试结果。有关更多信息，请参阅第 201 页的『[Web 网关安装验证应用程序](#)』。

Web 网关安装验证应用程序

IBM MQ Managed File Transfer 提供 Web 网关安装验证应用程序。使用此应用程序可查看 Web 网关安装的配置值并测试基本的 Web 网关功能。

有关如何访问安装验证应用程序的信息，请参阅第 201 页的『[验证 Web 网关安装](#)』。该应用程序显示两种类型的信息：Web 网关安装的配置值以及测试基本 Web 网关功能的结果。

配置值

在应用程序服务器中部署 Web 网关时，需要为多个初始化参数提供值。如果使用 WebSphere Application Server V7.0，那么需要在管理控制台中使用**初始化 servlet 的参数**步骤来提供这些值。如果使用 WebSphere Application Server Community Edition，那么可在 `web.xml` 文件中设置这些值。

该应用程序在标题 **Web 网关配置信息** 下面列出了以下 Web 网关设置的值：

servlet 信息

已部署的 Web 网关 servlet 的名称和版本。

Web 网关名称

部署的 Web 网关的名称。您需要为 **webGatewayName** 初始化参数提供该值。

上下文根

为 Web 网关应用程序定义的上下文根。在 WebSphere Application Server Community Edition 中，这是 `WEB-INF/geronimo-web.xml` 文件中 `<web:context-root>` 元素的值。对于 WebSphere Application Server V7.0，这是安装 Web 网关应用程序时在**映射 Web 模块**的上下文根步骤中设置的值。缺省值是 `wmqfte`。

文件空间根目录

Web 网关创建和管理的文件空间的根目录路径。您需要为 **fileSpaceRoot** 初始化参数提供该值。

临时文件上传根目录

Web 网关所启动传输的相关临时文件存储器的目录路径。您需要为 **tempFileUploadDir** 初始化参数提供该值。

临时文件上传目录的最大大小

允许用户存储与 Web 网关启动的传输相关的临时文件的最大空间量 (MB)。您需要为 **maxTempFileUploadSpace** 初始化参数提供该值。

MQMFT Web 代理名称

充当 Web 网关所启动传输的源的 IBM MQ Managed File Transfer 代理的名称。您需要为 **agentName** 初始化参数提供该值。这是运行 **fteCreateWebAgent** 命令时，使用 **-agentName** 参数为 Web 代理指定的名称。

协调队列管理器名称

Web 网关用于记录传输信息的协调队列管理器的名称。您需要为 **coordinationQMGr** 初始化参数提供该值。

缺省 MQMD 用户标识

当没有为用户定义特定 MQMD 用户标识时，用于与请求的用户关联的缺省 WebSphere MQ 消息描述符 (MQMD) 用户标识。您需要为 **defaultMQMDUserID** 初始化参数提供该值。

CSRF 保护

指示当前 Web 网关是否配置为执行 CSRF 标记验证以防止跨站点请求伪造攻击。您需要为 **CSRFProtection** 初始化参数提供该值。

应用程序服务器信息

托管 Web 网关应用程序的应用程序服务器的名称和版本。

Web 网关测试

安装验证应用程序在标题 **Web 网关测试结果** 下面显示多个测试的结果。如果测试失败，**信息**列中将显示 IBM MQ Managed File Transfer 错误代码和消息。有关错误消息的更多信息，请参阅[诊断消息](#)。列出了以下测试：

将文件上传到临时存储器

测试在**临时文件上传根目录**字段中指定的目录。应用程序会测试该目录是否存在、是否可读和可写，以及是否能够重新读取写入该目录的数据。

将文件上传到文件空间存储器

测试在**文件空间根目录**字段中指定的目录。应用程序会测试该目录是否存在、是否可读和可写，以及是否能够重新读取写入该目录的数据。

传输历史记录数据库访问

测试是否存在与传输历史记录数据库的连接。如果使用 WebSphere Application Server V7，那么应用程序会测试您在部署 Web 网关时配置的数据源。有关更多信息，请参阅第 193 页的『准备使用 WebSphere Application Server V7.0 部署 Web 网关』。如果使用 WebSphere Application Server Community Edition，那么应用程序会测试您在部署 Web 网关时配置的数据库池。有关更多信息，请参阅第 183 页的『准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关』。应用程序检查是否可以使用您在设置数据源或数据库池时提供的凭证来访问数据库。

应用程序还检查所需数据库表是否存在。有关更多信息，请参阅第 181 页的『设置数据库以用于文件空间』和第 200 页的『配置数据库记录器以用于 Web 网关』。

测试的最后部分检查是否正确定义了 Java 持久性 API (JPA) 对象。

文件空间信息数据库访问

测试是否存在与文件空间信息数据库的连接。如果使用 WebSphere Application Server V7，那么应用程序会测试您在部署 Web 网关时配置的数据源。有关更多信息，请参阅第 193 页的『准备使用 WebSphere Application Server V7.0 部署 Web 网关』。如果使用 WebSphere Application Server Community Edition，那么应用程序会测试您在部署 Web 网关时配置的数据库池。有关更多信息，请参阅第 183 页的『准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关』。应用程序检查是否可以使用您在设置数据源或数据库池时提供的凭证来访问数据库。

应用程序还检查所需数据库表是否存在。有关更多信息，请参阅第 181 页的『设置数据库以用于文件空间』和第 200 页的『配置数据库记录器以用于 Web 网关』。

测试的最后部分检查是否正确定义了 Java 持久性 API (JPA) 对象。

迁移 Web 网关

要将 WebSphere Application Server V7 上的 Web 网关从 IBM WebSphere MQ File Transfer Edition V7.0 迁移到 IBM WebSphere MQ 7.5 或更高版本，请完成以下步骤：

开始之前

在迁移之前，请记录多个现有设置。

过程

1. 打开 WebSphere Application Server 控制台。
2. 单击**应用程序 > 应用程序类型 > 企业应用程序**。在应用程序列表中找到 IBM WebSphere MQ File Transfer Edition Web 网关应用程序。如果 Web 网关应用程序尚未停止，那么请选择该应用程序并单击**停止**。
3. 记录先前为 Web 网关设置的配置设置。稍后在步骤 第 204 页的『7』中您将需要这些设置。
 - a) 单击**企业应用程序 > WebSphere MQ File Transfer Edition Web 网关 > servlet**的初始化参数，并记录以下 servlet 参数的设置：
 - webGatewayName
 - agentName
 - coordinationQMGr
 - fileSpaceRoot
 - tempFileUploadDir

- maxTempFileUploadSpace
 - defaultMQMDUserID
- b) 单击**企业应用程序 > WebSphere MQ File Transfer Edition Web 网关 > 安全角色到用户/组映射**，并记录窗格中的所有条目。
 - c) 单击**企业应用程序 > WebSphere MQ File Transfer Edition Web 网关 > Web 模块的上下文根**，并记录窗格中的所有条目。
 - d) 单击**企业应用程序 > WebSphere MQ File Transfer Edition Web 网关 > 消息驱动的 Bean 侦听器绑定 > 目标资源 JNDI 名称**，并记录激活规范。
例如，jms/WMQFTEActivationSpec
 - e) 单击**企业应用程序 > WebSphere MQ File Transfer Edition Web 网关 > 资源引用 - 队列连接工厂**，并记录所用的队列连接工厂和数据源。
例如，

```
jms/WMQFTEWebAgentConnectionFactory
DataSource: jdbc/wmqfte-filespace, jdbc/wmqfte-database values
```

4. 通过单击**应用程序 > 应用程序类型 > 企业应用程序**来卸载 IBM WebSphere MQ File Transfer Edition Web 网关应用程序。选择该 Web 网关应用程序，然后单击**卸载**。
5. 可选：如果正在使用多个安装迁移至 IBM WebSphere MQ 7.5 或更高版本，并且本机库路径有所不同，请通过单击**资源 > JMS 提供程序 > WebSphere MQ 消息传递提供程序**来更改此路径
例如，如果本机库路径为：C:\Program Files\IBM\WebSphere MQ\java\lib，请将路径更改为：
C:\Program Files\IBM\New MQ Installation Location\java\lib
6. 可选：如果使用多个安装来迁移至 IBM WebSphere MQ 7.5 或更高版本，那么必须使用 `setmqm` 命令将队列管理器与新安装关联。
7. 使用配置 Web 网关中的信息以及您先前在步骤 [第 203 页的『3』](#) 中记录的信息来重新安装 Web 网关应用程序。
8. 通过单击**应用程序 > 应用程序类型 > 企业应用程序**来启动此新应用程序。选择此 Web 网关应用程序，然后单击**启动**。
9. 运行 IVT 工具来验证此 Web 网关应用程序是否仍正常工作。有关更多信息，请参阅[验证 Web 网关安装](#)。

配置 Connect:Direct 网桥

配置 Connect:Direct 网桥，以在 IBM MQ Managed File Transfer 网络和 Connect:Direct 网络之间传输文件。Connect:Direct 网桥的组件包括 Connect:Direct 节点和专用于与该节点通信的 IBM MQ Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

关于此任务

完成以下步骤以配置 Connect:Direct 网桥：

过程

1. [第 205 页的『针对 Connect:Direct 网桥代理和节点选择操作系统』](#)。
2. [第 205 页的『选择和配置 Connect:Direct 节点』](#)。
3. [第 205 页的『创建和配置 Connect:Direct 网桥代理』](#)。
4. [第 206 页的『配置 ConnectDirectNodeProperties.xml 文件以包含有关远程 Connect:Direct 节点的信息』](#)。
5. [第 206 页的『配置 Connect:Direct 网桥代理和 Connect:Direct 节点之间的安全连接』](#)。

针对 Connect:Direct 网桥代理和节点选择操作系统

开始之前

组成 Connect:Direct 网桥的代理和节点必须在同一个系统上，或具有同一文件系统的访问权，例如，通过一个共享的 NFS 安装。该文件系统用于在涉及 Connect:Direct 网桥的文件传输期间在 **cdTmpDir** 参数定义的目录中临时存储文件。Connect:Direct 网桥代理和 Connect:Direct 网桥节点必须能够使用同一路径名称来对该目录进行寻址。例如，如果代理和节点在不同的 Windows 系统上，那么这些系统必须使用相同的盘符来安装共享文件系统。以下配置允许代理和节点使用同一路径名：

- 代理程序和节点位于同一系统上，该系统正在运行 Windows 或 Linux for System x
- 代理程序位于 Linux for System x 上，节点位于 UNIX 上
- 代理在一个 Windows 系统上，节点在另一个 Windows 系统上

以下配置不允许代理和节点使用同一路径名：

- 代理程序位于 Linux for System x 上，节点位于 Windows 上
- 代理在 Windows 上，节点在 UNIX 上

请在计划安装 Connect:Direct 网桥时考虑该限制。

要了解有关 Connect:Direct 网桥支持的操作系统版本的更多详细信息，请参阅 Web 页面 [WebSphere MQ 系统需求](#)。

过程

1. 选择在 System x 上运行 Windows 或 Linux 的系统以安装 Connect:Direct 网桥代理。
2. 选择 Connect:Direct for Windows 或 Connect:Direct for UNIX 支持的操作系统，以在其上安装 Connect:Direct 网桥节点。

选择和配置 Connect:Direct 节点

开始之前

在遵照以下指示信息之前，必须已安装 Connect:Direct 节点。

过程

1. 选择 Connect:Direct 节点，以供 IBM MQ Managed File Transfer 代理与其进行通信。
2. 检查所选 Connect:Direct 节点的网络映射。如果网络映射中包含在 Windows 操作系统上运行的远程节点的任何条目，那么必须确保这些条目指定这些节点在 Windows 上运行。
 - a) 如果针对 Connect:Direct 网桥选择的 Connect:Direct 节点在 Windows 上运行，那么请使用 Connect:Direct 请求程序来编辑该网络映射。确保将在 Windows 上运行的所有远程节点的操作系统字段都设置为 **Windows**。

创建和配置 Connect:Direct 网桥代理

关于此任务

Connect:Direct 网桥代理是专用于与 Connect:Direct 节点进行通信的 IBM MQ Managed File Transfer 代理。

过程

1. 使用 **fteCreateCDAgent** 命令创建 Connect:Direct 网桥代理。
 - a) 您必须为 **cdNode** 参数提供值。该参数指定代理用于作为 Connect:Direct 网桥的组成部分的 Connect:Direct 节点的名称。使用您在前一部分中选择的 Connect:Direct 节点的名称。
 - b) 为 **cdNodeHost** 和 **cdNodePort** 参数提供值，它们用于定义代理要与其通信的 Connect:Direct 节点。

如果不为 **cdNodeHost** 参数提供值，那么将使用本地系统的主机名或 IP 地址。如果不为 **cdNodePort** 参数提供值，那么将使用值 1363。

- c) 使用 [第 469 页的『fteCreateCDAgent（创建 Connect:Direct 网桥代理）』](#) 中的信息来确定是否需要为 **cdTmpDir** 参数指定值。
2. 将 IBM MQ Managed File Transfer 使用的用户凭证映射到 Connect:Direct 节点上的用户凭证。您可以通过使用以下某方法来映射凭证：
 - 创建 **ConnectDirectCredentials.xml** 文件以定义凭证映射信息。有关更多信息，请参阅 [第 207 页的『通过使用 ConnectDirectCredentials.xml 文件映射 Connect:Direct 的凭证』](#)。
 - 编写用户出口，以针对 Connect:Direct 网桥执行凭证映射。有关更多信息，请参阅 [第 209 页的『通过出口类映射 Connect:Direct 的凭证』](#)。

配置 **ConnectDirectNodeProperties.xml** 文件以包含有关远程 Connect:Direct 节点的信息

开始之前

在遵照以下指示信息之前，必须已创建 Connect:Direct 网桥代理。

过程

编辑 Connect:Direct 网桥代理配置目录中的模板 **ConnectDirectNodeProperties.xml**。对于要定义相关信息的每个 Connect:Direct 节点或节点组，请执行以下步骤：

- a) 在 **nodeProperties** 元素中，创建 **node** 元素。
- b) 将 **name** 属性添加到 **node** 元素。将该属性的值指定为模式，以与一个或多个远程 Connect:Direct 节点的名称匹配。
- c) 可选：将 **pattern** 属性添加到 **node** 元素，该元素指定 **name** 属性中的值的模式类型。有效值为 **regex** 和 **wildcard**。缺省选项为 **wildcard**。
- d) 将 **type** 属性添加到 **node** 元素，该元素指定运行 **name** 属性所指定的远程 Connect:Direct 节点的操作系统。

有效值包括：

- Windows - 节点在 Windows 上运行
- UNIX - 节点在 UNIX 或 Linux 上运行
- z/OS、zos、os/390 或 os390 - 节点在 z/OS 上运行

该属性的值不区分大小写。Connect:Direct 网桥不支持到其他操作系统上的远程节点的传输。

有关更多信息，请参阅 [第 621 页的『Connect:Direct 节点属性文件格式』](#)。

配置 Connect:Direct 网桥代理和 Connect:Direct 节点之间的安全连接

关于此任务

缺省情况下，Connect:Direct 网桥代理使用 TCP/IP 协议来连接到 Connect:Direct 节点。如果要在 Connect:Direct 网桥代理和 Connect:Direct 节点之间建立安全连接，那么您可以使用 SSL 协议或 TLS 协议。

过程

配置安全连接。有关如何执行此操作的示例，请参阅 [第 113 页的『配置 Connect:Direct 网桥代理和 Connect:Direct 节点之间的 SSL 或 TLS 连接』](#)。

相关概念

[第 423 页的『对 Connect:Direct 网桥进行故障诊断』](#)

使用以下参考信息和示例，以帮助您诊断 Connect:Direct 网桥返回的错误。

[第 288 页的『Connect:Direct 网桥』](#)

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关任务

第 113 页的『[配置 Connect:Direct 网桥代理和 Connect:Direct 节点之间的 SSL 或 TLS 连接](#)』

通过创建密钥库和信任库并在 Connect:Direct 网桥代理属性文件中设置属性来配置 Connect:Direct 网桥代理和 Connect:Direct 节点，以通过 SSL 协议彼此建立连接。

第 290 页的『[向 Connect:Direct 节点传输文件](#)』

您可以使用 Connect:Direct 网桥从 IBM MQ Managed File Transfer 代理向 Connect:Direct 节点传输文件。通过将 Connect:Direct 网桥代理指定为目标代理并以 `connect_direct_node_name:file_path` 格式指定目标文件，指定 Connect:Direct 节点作为传输的目标。

第 291 页的『[从 Connect:Direct 节点传输文件](#)』

您可以使用 Connect:Direct 网桥，将文件从 Connect:Direct 节点传输到 IBM MQ Managed File Transfer 代理。通过将 Connect:Direct 网桥代理指定为源代理并以 `connect_direct_node_name:file_path` 格式指定源规范，可以将 Connect:Direct 节点指定为传输源。

第 294 页的『[从 Connect:Direct 节点传输多个文件](#)』

您可以使用 Connect:Direct 网桥，将多个文件从 Connect:Direct 节点传输到 IBM MQ Managed File Transfer 代理。通过将 Connect:Direct 网桥代理指定为源代理并以 `connect_direct_node_name:file_path` 格式指定一个或多个源规范，可以将 Connect:Direct 节点指定为多文件传输的源。

映射 Connect:Direct 的凭证

通过使用 Connect:Direct 网桥代理的缺省凭证映射功能或编写自己的用户出口，将 IBM MQ Managed File Transfer 中的用户凭证映射到 Connect:Direct 节点上的用户凭证。IBM MQ Managed File Transfer 提供了执行用户凭证映射的样本用户出口。

相关任务

第 207 页的『[通过使用 ConnectDirectCredentials.xml 文件映射 Connect:Direct 的凭证](#)』

通过使用 Connect:Direct 网桥代理的缺省凭证映射功能，将 IBM MQ Managed File Transfer 中的用户凭证映射到 Connect:Direct 节点上的用户凭证。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以包含自己的凭证信息。

第 209 页的『[通过出口类映射 Connect:Direct 的凭证](#)』

如果不希望使用 Connect:Direct 网桥代理的缺省凭证映射功能，那么可以通过编写自己的用户出口，将 IBM MQ Managed File Transfer 中的用户凭证映射到 Connect:Direct 节点上的用户凭证。配置自己的凭证映射用户出口将会禁用缺省凭证映射功能。

相关参考

第 980 页的『[CDCredentialExit.java 接口](#)』

第 618 页的『[Connect:Direct 凭证文件格式](#)』

代理程序配置目录中的 `ConnectDirectCredentials.xml` 文件定义 Connect:Direct 代理程序用于向其自身授权 Connect:Direct 节点的用户名和凭证信息。

通过使用 *ConnectDirectCredentials.xml* 文件映射 *Connect:Direct* 的凭证

通过使用 Connect:Direct 网桥代理的缺省凭证映射功能，将 IBM MQ Managed File Transfer 中的用户凭证映射到 Connect:Direct 节点上的用户凭证。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以包含自己的凭证信息。

关于此任务

使用 `fteCreateCDAgent` 命令创建 Connect:Direct 网桥代理后，需要手动创建 `ConnectDirectCredentials.xml` 文件。必须编辑此文件以包含主机、用户和凭证信息，然后才能使用 Connect:Direct 网桥代理。有关更多信息，请参阅第 618 页的『[Connect:Direct 凭证文件格式](#)』。缺省情况下，此文件从当前用户主目录（例如 `/home/fteuser/ConnectDirectCredentials.xml`）中装入。如果要使用其他位置，那么必须通过 `ConnectDirectNodeProperties.xml` 文件中的 `<credentialsFile>` 元素指定该位置。

过程

1. 确保元素 `<tns:pnode name="Connect:Direct node host" pattern="wildcard">` 中的 `name` 属性包含 Connect:Direct 网桥代理连接到的 Connect:Direct 节点的名称值。该值必须与您为 **fteCreateCDAgent -cdNode** 参数指定的值相同。
`pattern` 属性的值可以是 `wildcard` 或 `regex`。如果未指定该属性，那么缺省值将为 `wildcard`。
2. 将用户标识和凭证信息作为 `<tns:pnode>` 的子元素插入该文件。
您可以将以下 `<tns:user>` 元素的一个或多个实例插入该文件：

```
<tns:user name="name"
          pattern="pattern"
          ignorecase="ignorecase"
          cdUserId="cdUserId"
          cdPassword="cdPassword"
          pnodeUserId="pnodeUserId"
          pnodePassword="pnodePassword">
</tns:user>
```

其中：

- `name` 是一种模式，用于匹配与 MQMFT 传输请求关联的 MQMD 用户标识。
- `pattern` 指定为 `name` 属性指定的模式是通配符表达式还是 Java 正则表达式。`pattern` 属性的值可以是 `wildcard` 或 `regex`。如果未指定该属性，那么缺省值将为 `wildcard`。
- `ignorecase` 指定 `name` 属性指定的模式是否区分大小写。如果未指定该属性，那么缺省值将为 `true`。
- `cdUserId` 是 Connect:Direct 网桥代理用于连接到 `<tns:pnode>` 元素的 `name` 属性指定的 Connect:Direct 节点的用户标识。如果可能，请确保 `cdUserId` 是 Connect:Direct 管理员用户标识。如果 `cdUserId` 不能为 Connect:Direct 管理员，那么请确保该用户标识在 Connect:Direct 网桥节点具有以下功能权限：
 - 对于 Windows 节点，请设置以下权限。为了便于阅读，该示例使用回车符进行了格式化：

```
View Processes in the TCQ      value: yes
Issue the copy receive, copy send, run job, and run task
Process statements
Issue the submit Process statement value: yes
Monitor, submit, change, and delete all Processes value: all
Access Process statistics      value: all
Use the trace tool or issue traceon and traceoff commands value: yes
Override Process options such as file attributes and remote node ID value: yes
```

- 对于 UNIX 节点，请在 `userfile.cfg` 文件中设置以下参数：

```
pstmt.copy          value: y
pstmt.upload         value: y
pstmt.download       value: y
pstmt.runjob         value: y
pstmt.runtask        value: y
cmd.submit           value: y
pstmt.submit         value: y
cmd.chgproc          value: y
cmd.delproc          value: y
cmd.flsproc          value: y
cmd.selproc          value: a
cmd.selstats         value: a
```

cmd.trace	value: y
snode.ovrd	value: y

- *cdPassword* 是与 *cdUserId* 属性指定的用户标识相关联的密码。
- 您可以选择性地指定 *pnodeUserId* 属性。此属性的值是由 `<tns:pnode>` 元素的 *name* 属性指定的 `Connect:Direct` 节点用于提交 `Connect:Direct` 进程的用户标识。如果未指定 *pnodeUserId* 属性，那么 `Connect:Direct` 节点将使用 *cdUserId* 属性指定的用户标识来提交 `Connect:Direct` 进程。
- 您可以选择性地指定 *pnodePassword* 属性。该属性的值是与 *pnodeUserId* 属性指定的用户标识相关联的密码。

如果不存在与 MQMD 用户标识匹配的用户元素，那么传输将失败。

3. 可选：您可以包含一个或多个 `<tns:snode>` 元素，作为 `<tns:user>` 元素的子元素。`<tns:snode>` 元素指定属于 `Connect:Direct` 网桥的 `Connect:Direct` 节点所使用的凭证。这些凭证包含 `Connect:Direct` 网桥节点用于连接到作为文件传输的源或目标的 `Connect:Direct` 节点的用户标识和密码。

将以下一个或多个元素插入该文件：

```
<tns:snode name="name"
  pattern="pattern"
  userId="userId"
  password="password" />
```

其中：

- *name* 是用于匹配作为文件传输的源或目标的 `Connect:Direct` 节点的名称的模式。
- *pattern* 指定为 *name* 属性指定的模式是通配符表达式还是 Java 正则表达式。*pattern* 属性的值可以是 *wildcard* 或 *regex*。如果未指定该属性，那么缺省值将为 *wildcard*。
- *userId* 是 `<tns:pnode>` 元素的 *name* 属性指定的 `Connect:Direct` 节点用于连接到与 `<tns:snode>` 的 *name* 属性指定的模式匹配的 `Connect:Direct` 节点的用户标识。
- *password* 是与 *userId* 属性指定的用户标识相关联的密码。

如果没有与文件传输的辅助节点匹配的 `<tns:snode>` 元素，那么这种情况不会导致传输失败。传输将启动并且未指定与 *snode* 一起使用的用户标识和密码。

结果

在搜索用户名或 `Connect:Direct` 节点名的模式匹配时，`Connect:Direct` 网桥代理将从文件开头搜索到文件末尾。找到的第一个匹配既是使用的匹配。

相关任务

第 204 页的『[配置 Connect:Direct 网桥](#)』

配置 `Connect:Direct` 网桥，以在 IBM MQ Managed File Transfer 网络和 `Connect:Direct` 网络之间传输文件。`Connect:Direct` 网桥的组件包括 `Connect:Direct` 节点和专用于与该节点通信的 IBM MQ Managed File Transfer 代理。该代理称为 `Connect:Direct` 网桥代理。

相关参考

第 618 页的『[Connect:Direct 凭证文件格式](#)』

代理程序配置目录中的 `ConnectDirectCredentials.xml` 文件定义 `Connect:Direct` 代理程序用于向其自身授权 `Connect:Direct` 节点的用户名和凭证信息。

第 469 页的『[fteCreateCDAgent \(创建 Connect:Direct 网桥代理\)](#)』

`fteCreateCDAgent` 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 `Connect:Direct` 网桥一起使用。

通过出口类映射 *Connect:Direct* 的凭证

如果不希望使用 `Connect:Direct` 网桥代理的缺省凭证映射功能，那么可以通过编写自己的用户出口，将 IBM MQ Managed File Transfer 中的用户凭证映射到 `Connect:Direct` 节点上的用户凭证。配置自己的凭证映射用户出口将会禁用缺省凭证映射功能。

关于此任务

您为映射 Connect:Direct 凭证创建的用户出口必须实现接口

`com.ibm.wmqfte.exitroutine.api.ConnectDirectCredentialExit`。有关更多信息，请参阅第 980 页的『CDCredentialExit.java 接口』。

配置 Connect:Direct 网桥代理和 Connect:Direct 节点之间的 SSL 或 TLS 连接

通过创建密钥库和信任库并在 Connect:Direct 网桥代理属性文件中设置属性来配置 Connect:Direct 网桥代理和 Connect:Direct 节点，以通过 SSL 协议彼此建立连接。

关于此任务

这些步骤中包含获取由认证中心签名的密钥的指示信息。如果您不使用认证中心，那么可以生成自签名证书。有关生成自签名证书的更多信息，请参阅在 UNIX 和 Windows 系统上使用 SSL 或 TLS。

这些步骤中包含有关为 Connect:Direct 网桥代理创建新的密钥库和信任库的指示信息。如果 Connect:Direct 网桥代理已具有用于安全连接到 WebSphere MQ 队列管理器的密钥库和信任库，那么在安全连接到 Connect:Direct 节点时您可以使用现有的密钥库和信任库。有关更多信息，请参阅第 103 页的『为 IBM MQ Managed File Transfer 配置 SSL 或 TLS 加密』。

过程

对于 Connect:Direct 节点，请完成以下步骤：

1. 针对 Connect:Direct 节点生成密钥和签名证书。

您可以通过使用与 WebSphere MQ 一起提供的 IBM Key Management 工具来完成该操作。有关更多信息，请参阅使用 SSL 或 TLS。

2. 向认证中心发送请求以获取签名密钥。您将收到证书。

3. 创建文本文件；例如 `/test/ssl/certs/CAcert`，以包含认证机构公用密钥。

4. 在 Connect:Direct 节点上安装 Secure+ Option。

如果该节点已存在，那么可以通过再次运行安装程序、指定现有安装的位置以及选择只安装 Secure+ Option 来安装 Secure+ Option。

5. 创建新的文本文件；例如 `/test/ssl/cd/keyCertFile/node_name.txt`。

6. 将从认证中心收到的证书和位于 `/test/ssl/cd/privateKeys/node_name.key` 中的专用密钥复制到该文本文件中。

`/test/ssl/cd/keyCertFile/node_name.txt` 的内容必须是以下格式：

```
-----BEGIN CERTIFICATE-----
MIIcCzCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJHJQjES
MBAGA1UECBMJSjSGfTcHNoaXJlMRAdBgYDVQHEwdIdXJzbGV5MQwwCgYDVQQKEwNJ
Qk0xZjAMBgNVBAStBU1RSVBUMQswCQYDVQDEwJDQTAeFw0xMTAzMDExNjIwNDZa
Fw0yMTAyMjYxNjIwNDZaMFAXCzAJBgNVBAYTAkdCMRIwEAYDVQQIEw1IYW1wc2hp
cmUxDDAKBgNVBAoTA01CTTEOMAwGA1UECzMFTVFGEUxZDzANBgNVBAMTBmJpbmJh
ZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwGykCgYEAvgP1QIk1U9ypSKD1Xo0Do1yk
EymFXBOUpZr1DVxjoSEC0vtWncJ199e+Vc4UpNybdyBu+Nkd1MNoF4X0xeQcLAFj
WnhakqCiQ+JIAD5Aurhn1wChe0MV3kjA84GKH/x0SVqt1984mu/1dyS819XcfSSn
c00MsK1KbneVSCIV2XECaWEAAa7MHkwCQYDVROTBAlwADAsBg1ghkgBhvhCAQ0E
HxYdT3B1b1NTTCBHZW51cmF0ZWQgQ2VydG1maWNhdGUwHQYDVRO0BBYEFNXXMIpSc
csBXUniW4A3UrzNCRsv3MB8GA1UdIwQYMBaAFDXY8rmj41Vz5+FVAoQb++cns+B4
MA0GCSqGSIb3DQEBBQUAA4GBAFc7k1Xa4pGKYgwchxKpE3ZF6FNwy4vBXS216/ja
8h/v18+iv010CL8t0ZOKSU95fyZLz0PKnCH7v+ItFSE3CIiEk9D1z2U6W091ICwn
17PL72TdfaL3kabwHYVf17IVcuL+VZsZ3HjLggP2qH09ZuJPspET9+AxFVMLiaAb
8eHw
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,64A02DA15B6B6EF9

57kqxL0J/gRU0IQ6hVK2YN13B4E1jAi1gSme0I5ZpEIG8CHXISKB7/0cke2FTqsV
1vI99QyCxsDw0Mnt5fj51v7aPmVeS60b0m+U1Gre8B/Zel8JVj204K2Uh72rDCXE
5e6eFxDuM207sQDy20euBVELJtM2k0kL1R0doQQS1U3XQNgJw/t3ZIx5hPXWEQT
rjRQ064BEhb+PzzPF8uwzZ9IiUK9BJ/UUnqC60dBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTK180GvBvWX04fHyvIX5as1whBoArXIS1AtNTptPvoaP1zyIAeZ60Cvo/
Sfo+A2UhmteJe0JaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIETgu4A1+CV64RJED
```

```
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWdyJBVhU0jXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfDwP+bEjDzUaaarJTS7lIFeLLw7eJ8MNAKMgicDkycL0
EPBU9X5QnHKLK0fYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+1YcX80VZ6
sHFPNlHluCNy/riUcBy9iviVeodX8Iom0chSy05DKl8bwZNjYtUP+CtYHNFU5BaD
I+1uU0AeJ+wjQYKT1WaeIGZ3VxuNITJu18y5qDTXXfX7vxM50oWxa6U5+AYuGUMg
/itPZmUmNzHjTk7ghT6i1IQ0aBowXXKJB1Mmq/6BQXN2IhkD9ys2qrvM1hdi5nAf
egmdiG50loLnBRqWbfR+DykpAhK4SaDi2F52Uxovw3Lhiw8dQP7lZQ==
-----END RSA PRIVATE KEY-----
```

7. 启动 Secure+ Admin Tool。

- 在 Linux 或 UNIX 系统上，运行命令 **spadmin.sh**。
- 在 Windows 系统上，单击开始 > 程序 > **Sterling Commerce Connect:Direct > CD Secure+ Admin Tool**

CD Secure+ Admin Tool 将启动。

8. 在 CD Secure+ Admin Tool 中，双击 **.Local** 行以编辑主 SSL 或 TLS 设置。

- 根据使用的协议，选择**启用 SSL 协议**或**启用 TLS 协议**。
- 选择**禁用覆盖**。
- 至少选择一个密码套件。
- 如果需要双向认证，请将 **启用客户机认证** 的值更改为 Yes。
- 在**可信根证书**字段中，输入认证中心的公用证书文件的路径 `/test/ssl/certs/CAcert`。
- 在**密钥证书文件**字段中，输入您创建的文件的名称，`/test/ssl/cd/keyCertFile/node_name.txt`。

9. 双击 **.Client** 行以编辑主 SSL 或 TLS 设置。

- 根据使用的协议，选择**启用 SSL 协议**或**启用 TLS 协议**。
- 选择**禁用覆盖**。

对于 Connect:Direct 网桥代理，请执行以下步骤：

10. 创建信任库。您可以通过创建虚密钥然后删除该虚密钥的方法来创建。

可以使用以下命令：

```
keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

```
keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

11. 将认证中心的公用证书导入该信任库。

可以使用以下命令：

```
keytool -import -trustcacerts -alias myCA
-file /test/ssl/certs/CAcert
-keystore /test/ssl/fte/stores/truststore.jks
```

12. 编辑 Connect:Direct 网桥代理属性文件。

在该文件的任何位置中包含以下行：

```
cdNodeProtocol=protocol
cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
cdNodeTruststorePassword=password
```

在此步骤的示例中，*protocol* 是您所使用的协议（即 SSL 或 TLS），*password* 是创建信任密钥库时指定的密码。

13. 如果希望双向认证，请为 Connect:Direct 网桥代理创建密钥和证书。

- 创建密钥库和密钥。

可以使用以下命令：

```
keytool -genkey -keyalg RSA -alias agent_name  
-keystore /test/ssl/fte/stores/keystore.jks  
-storepass password -validity 365
```

b) 生成签名请求。

可以使用以下命令：

```
keytool -certreq -v -alias agent_name  
-keystore /test/ssl/fte/stores/keystore.jks -storepass password  
-file /test/ssl/fte/requests/agent_name.request
```

c) 将上述步骤中收到的证书导入密钥库。该证书必须为 x.509 格式。

可以使用以下命令：

```
keytool -import -keystore /test/ssl/fte/stores/keystore.jks  
-storepass password -file certificate_file_path
```

d) 编辑 Connect:Direct 网桥代理属性文件。

在该文件的任何位置中包含以下行：

```
cdNodeKeystore=/test/ssl/fte/stores/keystore.jks  
cdNodeKeystorePassword=password
```

在此步骤的示例中，*password* 是创建密钥库时指定的密码。

相关任务

第 204 页的『配置 Connect:Direct 网桥』

配置 Connect:Direct 网桥，以在 IBM MQ Managed File Transfer 网络和 Connect:Direct 网络之间传输文件。Connect:Direct 网桥的组件包括 Connect:Direct 节点和专用于与该节点通信的 IBM MQ Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

使用 ConnectDirectProcessDefinition.xml 文件指定要启动的 Connect:Direct 进程

指定要在 IBM MQ Managed File Transfer 传输过程中启动的 Connect:Direct 进程。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以指定进程定义。

关于此任务

fteCreateCDAgent 命令在代理配置目录 *MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name* 中创建文件 *ConnectDirectProcessDefinitions.xml*。要从 Connect:Direct 网桥代理中调用用户定义的 Connect:Direct 进程，必须首先通过编辑该文件来设置进程定义。

对于要指定在传输过程中通过 Connect:Direct 网桥调用的每个进程，请执行以下步骤：

过程

1. 定义希望 Connect:Direct 网桥代理在传输过程中调用的 Connect:Direct 进程，并将进程模板保存在文件中。
2. 在文本编辑器中打开 *MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml* 文件。
3. 创建 `<processSet>` 元素。
4. 在 `<processSet>` 元素中，创建 `<condition>` 元素。
5. 在 `<condition>` 元素中，创建一个或多个元素，这些元素定义传输请求必须匹配的条件以调用您在步骤 1 中定义的 Connect:Direct 流程。这些元素可以是 `<match>` 元素或 `<defined>` 元素。
 - 使用 `<match>` 元素指定变量的值必须与模式匹配。创建具有以下属性的 `<match>` 元素：

- **variable** - 要比较其值的变量的名称。变量是一个内置符号。要了解更多信息，请参阅 [第 724 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』](#)。
- **value** - 要与指定变量的值比较的模式。
- (可选) **pattern** - **value** 属性的值使用的模式类型。该模式类型可以是 **wildcard** 或 **regex**。该属性为可选项，缺省值为 **wildcard**。
- 使用 **<defined>** 元素指定变量必须具有定义的值。创建具有以下属性的 **<defined>** 元素：
 - **variable** - 必须具有定义值的变量的名称。变量是一个内置符号。要了解更多信息，请参阅 [第 724 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』](#)。

在 **<condition>** 元素中指定的条件将通过逻辑 AND 组合。必须满足所有条件，Connect:Direct 网桥代理才能调用此 **<processSet>** 元素指定的进程。如果未指定 **<condition>** 元素，那么进程设置将匹配所有传输。

6. 在 **<processSet>** 元素中，创建 **<process>** 元素。

7. 在 **<process>** 元素中，创建 **<transfer>** 元素。

传输元素指定 Connect:Direct 网桥代理在传输过程中调用的 Connect:Direct 进程。创建具有以下属性的 **<transfer>** 元素：

- **process** - 您在步骤 1 中定义的 Connect:Direct 进程的位置。使用绝对路径或相对于 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` 目录指定此文件的位置。

结果

搜索条件匹配项时，Connect:Direct 网桥代理将从文件开头到文件末尾进行搜索。找到的第一个匹配既是使用的匹配。

相关任务

[第 204 页的『配置 Connect:Direct 网桥』](#)

配置 Connect:Direct 网桥，以在 IBM MQ Managed File Transfer 网络和 Connect:Direct 网络之间传输文件。Connect:Direct 网桥的组件包括 Connect:Direct 节点和专用于与该节点通信的 IBM MQ Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

相关参考

[第 623 页的『Connect:Direct 进程定义文件格式』](#)

Connect:Direct 网桥代理配置目录中的 `ConnectDirectProcessDefinitions.xml` 文件指定要在文件传输过程中启动的用户定义的 Connect:Direct 进程。

[第 469 页的『fteCreateCDAgent \(创建 Connect:Direct 网桥代理\)』](#)

`fteCreateCDAgent` 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

通过 MSCS 配置 IBM MQ Managed File Transfer 代理

如果平台是 MFT 支持的平台，并且正在运行某个版本的 Windows，那么支持 IBM MQ Managed File Transfer (MFT) 代理 MSCS 设置。

关于此任务

完成以下步骤以实现 MFT 代理故障转移：

过程

1. 在集群中的每台机器上本地安装 IBM MQ Managed File Transfer。
请参阅 [第 26 页的『如何安装』](#)，以获取适合您平台的安装指示信息的链接。
2. 在主机器上照常创建代理，确保在该代理的队列管理器上创建所有对象。
3. 将主机上的代理设置为 Windows 服务，在集群控制之下，如同针对现有服务一样。
请参阅 [第 216 页的『将代理作为 Windows 服务启动』](#)，以获得有关如何完成此操作的详细信息。
4. 在其他机器上创建相同的代理，但不启动代理，也不将代理设置为服务。

这将确保日志和属性等的文件结构存在于第二台机器上。

如果发生故障转移，代理服务可以传输到备份机器，并使用预期的文件结构继续工作。

管理 IBM MQ Managed File Transfer

使用 IBM MQ Managed File Transfer 命令来管理 IBM MQ Managed File Transfer。您也可以将 IBM MQ Explorer 用于某些管理任务。

通过将消息放入代理命令队列来启动传输

您还可以通过在源代理的命令队列上添加文件传输消息来启动文件传输。示例命令队列名称为 `SYSTEM.FTE.COMMAND.AGENT01`。您必须确保消息到达正确的源代理的命令队列；如果与 XML 中的源信息不匹配的代理收到消息，那么将拒绝消息。

The transfer request XML must conform to the `FileTransfer.xsd` schema and use the `<request>` element as the root element. 请参阅文件传输请求消息格式以获取有关传输队列消息的结构和内容信息。在代理命令队列上添加传输请求消息的方式特定于任务。例如，可以使用 IBM MQ Java API 以编程方式将消息放到队列中。

相关概念

[第 229 页的『资源监控』](#)

您可以监视 IBM MQ Managed File Transfer 资源；例如，队列或目录。当满足针对该资源的条件时，资源监视器将启动任务，如文件传输。您可以使用 `fteCreateMonitor` 命令或 IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

[第 250 页的『将数据从文件传输至消息』](#)

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

[第 263 页的『将数据从消息传输到文件』](#)

通过使用 IBM MQ Managed File Transfer 的消息到文件传输功能，您可以将 IBM MQ 队列上的一条或多条消息的数据传输到文件、数据集或用户文件空间。如果您拥有一个可创建或处理 IBM MQ 消息的应用程序，那么可以使用 IBM MQ Managed File Transfer 的消息到文件传输功能，将这些消息传输到 IBM MQ Managed File Transfer 网络中任何系统上的文件。

[第 151 页的『配置 Managed File Transfer 记录器』](#)

[第 275 页的『协议网桥』](#)

通过协议网桥，IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件，而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

[第 302 页的『使用 IBM Integration Bus』](#)

您可以通过 `FTEOutput` 和 `FTEInput` 节点使用来自 IBM Integration Bus 的 IBM MQ Managed File Transfer。

[第 302 页的『IBM MQ Managed File Transfer 的恢复和重新启动』](#)

如果由于任何原因（例如，由于电源故障或网络故障）您的代理或队列管理器不可用，那么 IBM MQ Managed File Transfer 将在这些场景下按如下方式恢复：

相关任务

[第 215 页的『启动 IBM MQ Managed File Transfer 代理』](#)

必须先启动 IBM MQ Managed File Transfer 代理，然后才能将其用于文件传输。

[第 221 页的『启动新的文件传输』](#)

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

[第 226 页的『从 IBM MQ Explorer 监视正在进行的文件传输』](#)

您可以使用 IBM MQ Explorer 中的 **受管文件传输 - 当前传输进度** 选项卡来监视正在进行的文件传输。可以从 IBM MQ Explorer 或命令行启动此文件传输。该选项卡还显示了启动调度传输时的调度传输的进度。

[第 228 页的『使用传输日志来查看文件传输的状态』](#)

您可以使用 IBM MQ Explorer 中的**传输日志**来查看文件传输的详细信息。这些可以通过命令行或 IBM MQ Explorer 启动的传输。您还可以定制**传输日志**中显示的内容。

[第 248 页的『使用传输模板』](#)

您可以使用文件传输模板来存储重复或复杂传输的公共文件传输设置。使用 **fteCreateTemplate** 命令从命令行创建传输模板，或者使用 IBM MQ Explorer 通过 "为受管文件传输创建新模板" 向导创建传输模板，或者通过选中 **将传输设置另存为模板** 复选框在创建文件传输时保存模板。**传输模板**窗口显示在您的 IBM MQ Managed File Transfer 网络中已创建的所有传输模板。

[第 273 页的『列出 IBM MQ Managed File Transfer 代理』](#)

可使用命令行或 IBM MQ Explorer 来列出已向特定队列管理器注册的代理。

[第 273 页的『停止 IBM MQ Managed File Transfer 代理』](#)

您可以通过命令行停止代理。在停止代理时，您将停顿代理并允许代理在停止前完成其当前文件传输。您还可以在命令行上指定 **-i** 参数来立即停止代理。在代理停止时，您无法使用该代理传输文件，直至重新启动代理。

相关参考

[第 703 页的『文件传输准则』](#)

根据传输源和目标的操作系统，以及使用二进制还是文本方式传输，提供有关期望行为的指南。

启动 IBM MQ Managed File Transfer 代理

必须先启动 IBM MQ Managed File Transfer 代理，然后才能将其用于文件传输。

关于此任务

您可以从命令行启动 IBM MQ Managed File Transfer 代理。在此情况下，当您从系统注销后，代理进程将停止。

在 Windows、UNIX 和 Linux 上，您可以配置代理以使其在您从系统注销后继续运行，并且可以继续接收文件传输。

在 z/OS 上，您可以配置代理以通过 JCL 将代理作为已启动任务启动，而无需交互式会话。

V8.0.0.3 请注意，从 IBM MQ 8.0.0 修订包 3 开始，如果在代理运行时遇到不可恢复错误，那么将生成第一个故障数据捕获 (FDC) 并且将停止代理。

过程

- 要从命令行启动代理，请使用 **fteStartAgent** 命令。
有关更多信息，请参阅 [第 573 页的『fteStartAgent（启动 IBM MQ Managed File Transfer 代理）』](#)。
- 要配置代理以使其在您从系统注销后继续运行：
 - 在 Windows 上，将代理配置为作为 Windows 服务运行。要了解更多信息，请参阅 [第 216 页的『将代理作为 Windows 服务启动』](#)。
 - 在 UNIX 和 Linux 上，配置代理以在重新引导期间使用脚本文件自动启动。有关更多信息，请参阅 [第 218 页的『在 UNIX 系统启动时启动代理』](#)。
- 在 z/OS 上，配置代理以通过 JCL 将代理作为已启动任务启动，而无需交互式会话。
有关更多信息，请参阅 [第 216 页的『在 z/OS 上启动代理程序』](#)。

相关任务

[第 273 页的『列出 IBM MQ Managed File Transfer 代理』](#)

可使用命令行或 IBM MQ Explorer 来列出已向特定队列管理器注册的代理。

[第 273 页的『停止 IBM MQ Managed File Transfer 代理』](#)

您可以通过命令行停止代理。在停止代理时，您将停顿代理并允许代理在停止前完成其当前文件传输。您还可以在命令行上指定 **-i** 参数来立即停止代理。在代理停止时，您无法使用该代理传输文件，直至重新启动代理。

相关参考

第 573 页的『[fteStartAgent（启动 IBM MQ Managed File Transfer 代理）](#)』
fteStartAgent 命令从命令行启动 IBM MQ Managed File Transfer 代理程序。

在 z/OS 上启动代理程序

在 z/OS 上，除了通过 UNIX 系统服务会话运行 **fteStartAgent** 命令，您还可以通过 JCL 将代理作为已启动任务启动，而无需交互式会话。

将使用已启动任务，由于它在特定的用户标识下运行，不受最终用户注销影响。

注：V8.0.0.6 已启动的任务通常由可能无登录权限的管理用户运行，因此，无法以运行代理的用户身份登录到 z/OS 系统。无法对该代理发出 **fteStartAgent**、**fteStopAgent** 和 **fteSetAgentTraceLevel** 命令以及指定了 **-d** 参数的 **fteShowAgentDetails** 命令。

从 IBM MQ 8.0.0 修订包 6 开始，代理属性 **adminGroup** 可用于 z/OS 上的 Managed File Transfer 代理程序。您可以定义安全管理器组（例如，*MFTADMIN*），然后将已启动任务用户标识和管理员 TSO 标识添加到该组。编辑代理属性文件并将 **adminGroup** 属性设置为该安全管理器组的名称。

```
adminGroup=MFTADMIN
```

然后，该组成员可以对作为已启动任务运行的代理发出 **fteStartAgent**、**fteStopAgent** 和 **fteSetAgentTraceLevel** 命令以及指定了 **-d** 参数的 **fteShowAgentDetails** 命令。

有关更多信息，请参阅第 591 页的『[agent.properties 文件](#)』中的 **adminGroup** 属性。

作为 Java 应用程序，代理程序是一个 UNIX 系统服务应用程序，您可以使用 BFGAGSTP 成员从为代理程序生成的 IBM MQ Managed File Transfer 命令 PDSE 库数据集从 JCL 运行该应用程序。有关如何创建 MQMFT 命令 PDSE 库数据集并针对所需的代理进行定制的更多信息，请参阅第 118 页的『[创建 IBM MQ Managed File Transfer 代理或记录器命令数据集](#)』。

相关参考

第 274 页的『[在 z/OS 上停止代理](#)』

如果要在 z/OS 上作为来自 JCL 的启动式任务运行 IBM MQ Managed File Transfer 代理程序，那么该代理程序将接受 z/OS 操作员命令 **MODIFY** 和 **STOP** 以及 **fteStopAgent** 命令。

将代理作为 Windows 服务启动

您可以将代理作为 Windows 服务启动，以便当您从 Windows 注销后，代理可以继续运行，并且可以接收文件传输。

关于此任务

在 Windows 上，从命令行启动代理时，代理进程将使用用于登录 Windows 的用户名运行。从系统注销后，代理进程将停止。为防止代理停止，可以配置代理以作为 Windows 服务运行。作为 Windows 服务运行还允许您配置代理，使其在 Windows 环境启动或重新启动时自动启动。

完成以下步骤以启动作为 Windows 服务运行的代理程序。必须在某个受支持的 Windows 版本上运行 IBM MQ Managed File Transfer，这样才能将代理作为 Windows 服务运行。要获取受支持环境的列表，请参阅 [WebSphere MQ 系统需求](#)。

确切的步骤取决于您已创建代理还是将要创建代理。以下步骤对这两个选项进行了描述。

过程

1. 如果要创建 MQMFT 代理程序，请使用 **fteCreateAgent**、**fteCreateWebAgent**、**fteCreateCDAgent** 或 **fteCreateBridgeAgent** 命令。指定 **-s** 参数以将代理程序作为 Windows 服务运行。在以下示例中，将创建具有代理队列管理器 QMGR1 的代理 AGENT1。该 Windows 服务将使用用户名 **fteuser**（具有关联的密码 **ftepassword**）运行。


```
fteCreateAgent -agentName AGENT1 -agentQMGR QMGR1 -s -su fteuser -sp ftepassword
```

您可以选择在 **-s** 参数后为服务指定名称。如果未指定名称，那么会将该服务命名为 `mqmftAgent<AGENT><QMGR>`，其中 *AGENT* 是您指定的代理名称，*QMGR* 是您的代理队列管理器名称。在该示例中，服务的缺省名称为 `mqmftAgentAGENT1QMGR1`。

注：使用 **-su** 参数指定的 Windows 用户帐户必须具有 **Log on as a service** 权限。有关如何进行此配置的信息，请参阅第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』。

有关更多信息，请参阅第 460 页的『[fteCreateAgent（创建 IBM MQ Managed File Transfer 代理）](#)』、第 516 页的『[fteCreateWebAgent（创建 IBM MQ Managed File Transfer Web 代理）](#)』、第 469 页的『[fteCreateCDAgent（创建 Connect:Direct 网桥代理）](#)』或第 463 页的『[fteCreateBridgeAgent（创建和配置 IBM MQ Managed File Transfer 协议网桥代理）](#)』。

2. 如果已遵循上述步骤创建代理，请运行通过 **fteCreateAgent**、**fteCreateWebAgent**、**fteCreateCDAgent** 或 **fteCreateBridgeAgent** 命令生成的 MQSC 命令。这些命令将创建代理所需的 WebSphere MQ 队列。

例如，对于名为 *AGENT1* 的代理，名为 *QMGR1* 的代理队列管理器和名为 *COORDQMGR1* 的协调队列管理器，将运行以下命令：

```
runmqsc QMGR1 <MQ_DATA_PATH>\mqft\config\COORDQMGR1\agents\AGENT1\AGENT1_create.mqsc
```

3. 如果未遵循上述步骤创建代理，而是希望配置现有代理以作为 Windows 服务运行，请先停止代理（如果它正在运行），然后修改其配置。

a) 以下示例使用名为 *AGENT1* 的代理。运行以下命令：

```
fteStopAgent AGENT1
```

b) 使用 **fteModifyAgent** 命令配置代理以作为 Windows 服务运行：

```
fteModifyAgent -agentName AGENT1 -s -su fteuser -sp ftepassword
```

有关更多信息，请参阅第 546 页的『[fteModifyAgent（修改 IBM MQ Managed File Transfer 代理）](#)』。

4. 使用 **fteStartAgent** 命令启动代理。另外，还可以使用 Windows 服务工具（在 Windows 桌面开始菜单上选择“控制面板”，然后选择“管理工具”即可找到）启动该服务。

```
fteStartAgent AGENT1
```

即使您注销 Windows，该服务仍会继续运行。为确保 Windows 在关闭之后重新启动时该服务也重新启动，缺省情况下，会将 Windows 服务工具中的启动类型字段设置为自动。如果不希望该服务在 Windows 重新启动时亦重新启动，请将此设置更改为手动。

5. 可选：要停止代理，请使用 **fteStopAgent** 命令，或者使用 Windows 服务工具。例如，从命令行运行以下命令：

```
fteStopAgent AGENT1
```

- 将 **fteStopAgent** 命令作为服务运行时，该命令会始终使用 **-i** 参数运行，而不管您是否已指定该参数。**-i** 参数会立即停止代理，而不完成任何正在进行的传输。Windows 服务的限制导致了此问题。

下一步做什么

如果存在有关启动 Windows 服务的问题，请参阅第 397 页的『[有关作为 Windows 服务运行代理或记录器的指南](#)』。本主题还将描述 Windows 服务日志文件的位置。

相关概念

[第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』](#)

您可以作为 Windows 服务来运行 IBM MQ Managed File Transfer 代理、独立数据库记录器和独立文件记录器。如果这些 Windows 服务出现任何问题，您可以使用服务日志文件以及本主题中的信息来诊断问题。

相关任务

第 273 页的『[列出 IBM MQ Managed File Transfer 代理](#)』

可使用命令行或 IBM MQ Explorer 来列出已向特定队列管理器注册的代理。

第 273 页的『[停止 IBM MQ Managed File Transfer 代理](#)』

您可以通过命令行停止代理。在停止代理时，您将停顿代理并允许代理在停止前完成其当前文件传输。您还可以在命令行上指定 **-i** 参数来立即停止代理。在代理停止时，您无法使用该代理传输文件，直至重新启动代理。

相关参考

第 460 页的『[fteCreateAgent（创建 IBM MQ Managed File Transfer 代理）](#)』

fteCreateAgent 命令创建代理及其关联的配置。

第 516 页的『[fteCreateWebAgent（创建 IBM MQ Managed File Transfer Web 代理）](#)』

fteCreateWebAgent 命令创建代理及其关联的配置，以用于 Web 网关。IBM MQ Managed File Transfer 服务器随附该命令。

第 469 页的『[fteCreateCDAgent（创建 Connect:Direct 网桥代理）](#)』

fteCreateCDAgent 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

第 463 页的『[fteCreateBridgeAgent（创建和配置 IBM MQ Managed File Transfer 协议网桥代理）](#)』

fteCreateBridgeAgent 命令创建协议网桥代理及其关联的配置。针对您要与之进行文件的发送和接收操作的每个文件服务器，创建一个协议网桥代理。

第 546 页的『[fteModifyAgent（修改 IBM MQ Managed File Transfer 代理）](#)』

fteModifyAgent 命令修改现有代理程序，以便可以将其作为 Windows 服务运行。此命令仅在 Windows 上可用，并且必须由作为 IBM MQ 管理员和 mqm 组成员的用户运行。

在 UNIX 系统启动时启动代理

可以将 IBM MQ Managed File Transfer 代理配置为在 UNIX 系统启动时启动。当您从 UNIX 系统注销后，您的代理将继续运行，并且可以接收文件传输。

当您使用某条 IBM MQ Managed File Transfer 命令（**fteCreateAgent**、**fteCreateCDAgent**、**fteCreateWebAgent** 或 **fteCreateBridgeAgent**）创建和配置了代理后，您可以使用一个脚本文件，将其配置为在 UNIX 机器重新引导期间自动启动，此脚本只需执行以下命令：

```
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name
```

其中，*mq_install_root* 是必需的 IBM MQ Managed File Transfer 7.5 安装的根目录，缺省值为 */opt/mqm*，*agent_name* 是要启动的 IBM MQ Managed File Transfer 代理程序的名称。该脚本文件的用法因特定 UNIX 操作系统而异。

Linux

对于 Linux 系统，有多种方法可在系统引导过程中启动应用程序。通常，我们建议您执行以下步骤：

1. 创建一个名为 */etc/rc.mqmft* 的文件，包含以下内容：

```
#!/bin/sh
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name"
```

其中，*mqmft_user* 是将作为代理进程运行身份的用户标识。该用户标识必须是 mqm 组的成员。

2. 使该文件可执行，例如：

```
chmod 755 /etc/rc.mqmft
```


3. 接下来，将以下行添加到 `/etc/inittab`:

```
mqmft:5:boot:/etc/rc.mqmft
```

在 Linux 上引导期间启动代理程序的其他方法包括将脚本行添加到 `/etc/rc.d/rc.local` 文件，或者在 Linux SuSe 上，将脚本行添加到 `/etc/init.d/boot.local` 文件。您应选择最适合自己的方法。以下是有关在受支持的特定 Linux 分发版上启动期间启动代理的其他方法的更多信息：

SLES 10 和 11

对于 SUSE Linux Enterprise Server (SLES) 10 和 11 系统，请按以下步骤执行操作：

1. 以系统 root 用户标识身份创建您自己的 `/etc/init.d/rc.rclocal` 文件。
2. 将以下行添加到 `rc.rclocal` 文件中：

```
#!/bin/sh
### BEGIN INIT INFO
# Provides: rc.rclocal
# Required-Start: $network $syslog
# Required-Stop: $network $syslog
# Default-Stop: 0 1 2 6
# Description: MQMFT agent startup
### END INIT INFO
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name"
```

3. 运行下列命令：

```
chmod 755 rc.rclocal
chkconfig --add rc.rclocal
```

Solaris

在 Solaris 上，请执行以下步骤：

1. 运行以下命令，然后记录返回的路径：

```
which sh
```

例如，路径可能为 `/usr/bin/sh`

2. 以系统 root 用户标识身份创建您自己的 `/etc/init.d/startmqmft` 文件。
3. 使用第 1 步返回的路径作为脚本中的第一行，编辑此文件，将以下脚本行添加到其中：

```
#!/usr/bin/sh
su mqmft_user mq_install_root/bin/fteStartAgent agent_name
```

4. 使该文件可执行，例如：

```
chmod 755 /etc/init.d/startmqmft
```

5. 创建该文件到 `rc3.d` 目录的符号链接：

```
ln -s /etc/init.d/startmqmft /etc/rc3.d/S98startmqmft
```

前缀 S 表示 Solaris 的缺省状态“已启动”。98 是序号。后缀是来自 `init.d` 的文件名

HP-UX

在 HP-UX 上，请执行以下步骤：

1. 创建一个名为 `/sbin/init.d/mqmft` 的文件，包含以下内容：

```
#!/bin/sh
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name
```

2. 创建一个名为 `/etc/rc.config.d/mqmft` 的文件，包含以下内容：

```
MQMFT=1
```

3. 创建该文件到 `rc3.d` 目录的符号链接并启动代理：

```
ln -s /sbin/init.d/mqmft /sbin/rc3.d/S84mqmft
```

相关任务

第 273 页的『[停止 IBM MQ Managed File Transfer 代理](#)』

您可以通过命令行停止代理。在停止代理时，您将停顿代理并允许代理在停止前完成其当前文件传输。您还可以在命令行上指定 **-i** 参数来立即停止代理。在代理停止时，您无法使用该代理传输文件，直至重新启动代理。

相关参考

第 460 页的『[fteCreateAgent（创建 IBM MQ Managed File Transfer 代理）](#)』

fteCreateAgent 命令创建代理及其关联的配置。

第 516 页的『[fteCreateWebAgent（创建 IBM MQ Managed File Transfer Web 代理）](#)』

fteCreateWebAgent 命令创建代理及其关联的配置，以用于 Web 网关。IBM MQ Managed File Transfer 服务器随附该命令。

第 469 页的『[fteCreateCDAgent（创建 Connect:Direct 网桥代理）](#)』

fteCreateCDAgent 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

第 463 页的『[fteCreateBridgeAgent（创建和配置 IBM MQ Managed File Transfer 协议网桥代理）](#)』

fteCreateBridgeAgent 命令创建协议网桥代理及其关联的配置。针对您要与之进行文件的发送和接收操作的每个文件服务器，创建一个协议网桥代理。

在 4690 OS 系统上启动代理

要启动代理，必须先将其配置为后台应用程序。可以配置后台应用程序在系统变成代理主控制器或文件服务器时自动启动，在系统不再是代理主控制器或文件服务器时停止。

关于此任务

已通过运行 `f:\adxetc\mft75\bin\ftecfg.bat` 实用程序装入新配置，定义了可启动的代理。对于每个配置的代理，`f:\adxetc\mft75` 目录中都存在一个 `agent_name.rsp` 响应文件。所有这些响应文件都包含一条命令以在传递 `ADXCHAIN.386` 系统命令时启动特定代理。

过程

完成以下步骤以启动作为后台服务运行的代理：

1. 从“系统主菜单”面板中选择 **4（安装和更新帮助）**。
2. 从“安装和更新帮助”面板中选择 **1（更改配置数据）**。
3. 从“配置”面板中选择 **2（控制器配置）**。
4. 如果询问您是否要配置使用 IBM Multiple Controller Feature 的存储系统，请按 **Enter** 键以选择 **Yes**。
5. 如果显示“局域网配置”面板，请选择适用于您环境的选项，然后按 **Enter** 键。
6. 如果显示“SNA 配置”面板，请选择适用于您环境的选项，然后按 **Enter** 键。
7. 如果提示您输入存储控制器标识，请指定相应的控制器标识，然后按 **Enter** 键。
8. 选择要配置的存储控制器，然后按 **Enter** 键。

9. 从控制器配置项列表中，选择**后台配置**，然后按 **Enter** 键。
10. 从“**后台应用程序**”面板中选择 **1**（定义后台应用程序）。
11. 在“**定义后台应用程序**”面板上，指定以下条目：
 - 初始消息：MQMFT
 - 程序名：ADX_SPGM:ADXCHAIN.386
 - 参数列表：@f:\adxetc\mft75\agent_name.rsp

注：agent_name 限制最多 23 个字符，参数列表条目限制最多 45 个字符。响应文件的参数列表路径必须完全按照所示格式指定，即不能更改大写字符或正斜杠。不允许定义逻辑名以指定到响应文件的路径。
12. 按**下一页**以查看更多选项，并指定当系统变为代理主系统或不再是代理主系统时应用程序应启动还是停止。
13. 再次按**下一页**以查看更多选项，并指定当系统变为代理文件服务器或不再是代理文件服务器时应启动还是停止。
14. 按 **Enter** 键保存变更。
15. 按 **Esc** 键返回“**配置**”面板。
16. 从“**配置**”面板中选择 **4**（**激活配置**）。
17. 从“**激活配置**”面板中选择 **2**（**控制器配置**）。
这会激活控制器配置。
18. 重新 IPL 存储控制器。

启动新的文件传输

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

关于此任务

要从命令行启动新的文件传输，请参阅 [fteCreateTransfer](#) 命令。

要使用 IBM MQ Explorer 中的**新建受管文件传输**向导来启动新的文件传输，请使用以下步骤：

过程

1. 在“**导航器**”视图中，单击**受管文件传输**。在“**内容**”视图中将显示**受管文件传输中心**。
2. 在“**导航器**”视图中将显示所有协调队列管理器。展开您希望用于传输的代理所注册到的协调队列管理器的名称。如果您当前连接到的协调队列管理器不是要用于传输的协调队列管理器，请在“**导航器**”视图中右键单击该协调队列管理器名称，然后单击**断开连接**。然后，右键单击想要使用的协调队列管理器的名称，并单击**连接**。
3. 通过以下任意一种方法，启动**创建新的受管文件传输**向导：
 - a) 在“**导航器**”视图中右键单击下列任意节点的名称：相关协调队列管理器、**传输模板**、**传输日志**或**暂挂传输**。然后单击**新建传输**来启动向导。
 - b) 单击**文件 > 新建 > 其他 > 受管文件传输向导 > 新建传输向导**
4. 遵循向导面板上的指示信息。对于每个面板，还提供有上下文相关帮助。要在 Windows 上访问上下文相关帮助，请按 F1。在 Linux 上，按 Ctrl+F1 或 Shift+F1。

相关概念

第 222 页的『使用传输定义文件』

您可以指定可用于创建文件传输的传输定义文件。传输定义文件是定义创建传输所需的部分或全部信息的 XML 文件。

相关任务

第 224 页的『创建调度的文件传输』

您可以从 IBM MQ Explorer 或从命令行调度新的文件传输。调度的传输可以将单个文件或多个文件包含在一个组中。您可以将调度的文件传输执行一次或者将该传输重复多次。

第 225 页的『触发文件传输』

您可以设置某些关于文件传输的触发条件，这些触发条件必须成立才能进行传输。如果触发条件不成立，那么文件传输将无法进行并且可以选择提交日志消息以记录传输未发生这一事实。然后将废弃文件传输请求。例如，您可以将文件传输设置为仅在以下情况下发生：源代理所在的系统上的指定文件超过了指定大小，或者源代理所在的系统上存在某一特定指定文件。可以通过 IBM MQ Explorer 或命令行来设置已触发的文件传输。

相关参考

第 497 页的『fteCreateTransfer（创建新的文件传输）』

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

第 842 页的『文件传输请求消息格式』

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

第 703 页的『文件传输准则』

根据传输源和目标的操作系统，以及使用二进制还是文本方式传输，提供有关期望行为的指南。

使用传输定义文件

您可以指定可用于创建文件传输的传输定义文件。传输定义文件是定义创建传输所需的部分或全部信息的 XML 文件。

希望在单个传输操作中指定多个源文件和多个目标文件时，传输定义文件非常有用。您可以使用传输定义文件来提交复杂文件传输。您可以复用和共享传输定义文件。

您可以使用两种格式的传输定义文件，虽然这些格式略有不同，但都符合 FileTransfer.xsd 模式。您可以在 IBM MQ Managed File Transfer 安装的 samples\schema 目录中找到此模式。

支持以下两种格式的传输定义文件：

- 传输的源和目标文件的定义。该定义使用 <transferSpecifications> 元素作为根。
- 整个传输的定义，包括源和目标文件以及源和目标代理。该定义使用 <request> 元素作为根。
 - 可使用 **-gt** 参数通过 **fteCreateTransfer** 命令生成此格式的文件。

以下示例显示仅指定传输的源和目标文件的传输定义文件格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<transferSpecifications xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <item checksumMethod="MD5" mode="text">
    <source recursive="false" disposition="leave">
      <file>textTransferTest.txt</file>
    </source>
    <destination type="directory" exist="overwrite">
      <file>c:\targetfiles</file>
    </destination>
  </item>
</transferSpecifications>
```

要提交该格式的传输定义文件，您必须在命令行上指定源和目标代理：

```
fteCreateTransfer -sa AGENT1 -sm agent1qm -da AGENT2 -dm agent2qm -td
c:\definitions\example1.xml
```

以下示例是指定传输所需的所有信息的传输定义文件格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="3.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
```

```

    <hostName>example.com.</hostName>
    <userID>fteuser</userID>
  </originator>
  <sourceAgent agent="AGENT1" QMgr="agent1qm"/>
  <destinationAgent agent="AGENT2" QMgr="agent2qm"/>
  <transferSet>
    <item mode="binary" checksumMethod="MD5">
      <source recursive="false" disposition="leave">
        <file>c:\sourcefiles\*.jpg</file>
      </source>
      <destination type="directory" exist="error">
        <file>/targetfiles/images</file>
      </destination>
    </item>
  </transferSet>
</managedTransfer>
</request>

```

您可以在 **fteCreateTransfer** 命令上使用 **-gt** 参数来生成该格式的文件。在提交使用该格式的传输定义文件时，无需在命令行上指定任何项：

```
fteCreateTransfer -td c:\definitions\example2.xml
```

除传输定义文件外，您还可以通过传入常规参数，来覆盖有关命令行的源和目标代理信息。例如：

```
fteCreateTransfer -da AGENT9 -dm agent9qm -td c:\definitions\example2.xml
```

该示例使用命令行选项来覆盖带有 **AGENT9** 的传输定义文件中定义的目标代理，以及作为 **agent9qm** 在传输定义文件中定义的目标队列管理器。

所描述的两种格式都可以包含一个或多个 **<item>** 元素。For further information about the **<item>** element, see [文件传输请求消息格式](#)，其中每个传输项都定义一个源和目标文件对以及其他属性，用于控制传输的行为。例如，您可以指定以下行为：

- 传输是否使用校验和
- 传输是文本还是二进制
- 是否在传输完成后删除源文件
- 如果目标文件存在，是否覆盖

使用传输定义文件的优点是您可以指定命令行中不可用的其他选项。例如，在执行消息至文件传输时，您可以使用传输定义文件来指定 **groupId** 属性。该属性指定从队列读取的消息的 WebSphere MQ 组标识。传输定义文件的另一个优点是您可以针对每个文件对指定不同的选项。例如，您可以基于文件，指定是否使用校验和，以及以文本还是二进制方式传输文件。如果使用命令行，那么相同的选项将应用于传输中的每个文件。

例如：

```

<item checksumMethod="none" mode="binary">
  <source disposition="leave">
    <file>c:\sourcefiles\source1.doc</file>
  </source>
  <destination type="file" exist="error">
    <file>c:\destinationfiles\destination1.doc</file>
  </destination>
</item>

<item checksumMethod="MD5" mode="text">
  <source disposition="delete">
    <file>c:\sourcefiles\source2.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file encoding="UTF8" EOL="CRLF">c:\destinationfiles\destination2.txt</file>
  </destination>
</item>

<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>c:\originfiles\source3.txt</file>
  </source>
  <destination type="file" exist="error">
    <file>c:\destinationfiles\destination3.txt</file>
  </destination>
</item>

```

```

</source>
<destination type="file" exist="overwrite">
  <file>c:\targetfiles\destination3.txt</file>
</destination>
</item>

```

您可以使用项目将文件从分布式系统传输到 z/OS 系统：

```

<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>textTransferTest.txt</file>
  </source>
  <destination type="dataset" exist="overwrite">
    <file encoding="IBM-1047">//TEXT.TRANS.TEST</file>
  </destination>
</item>

```

该示例以文本方式将文件 `textTransferTest.txt` 从源代理传输到目标代理上的数据集 `//TEXT.TRANS.TEST`。该传输将源数据从源代理的缺省编码（未指定任何源编码属性）转换为代码页：IBM-1047。

创建调度的文件传输

您可以从 IBM MQ Explorer 或从命令行调度新的文件传输。调度的传输可以将单个文件或多个文件包含在一个组中。您可以将调度的文件传输执行一次或者将该传输重复多次。

关于此任务

您可以将文件传输调度设置为执行一次或者按以下时间间隔执行：

- 每分钟
- 每小时
- 每天
- 每周
- 每月
- 每年

然后，您可以指定在以下点停止执行：

- 在定义的时间和日期
- 在执行定义的次数之后

或者，您也可以指定永续执行。

要使用命令行创建新的调度文件传输，请使用 `fteCreateTransfer` 命令的调度参数 (`-tb`，`-ss`，`-oi`，`-of`，`-oc` 和 `-es`)。

要使用 IBM MQ Explorer 中的**新建受管文件传输**向导来新建已调度的文件传输，请使用以下步骤：

过程

1. 在“导航器”视图中，单击**受管文件传输**。在“内容”视图将显示**受管文件传输中心**。
2. 在“导航器”视图将显示所有协调队列管理器。展开您希望用于传输的代理所注册到的协调队列管理器的名称。如果您当前连接到的协调队列管理器不是要用于传输的协调队列管理器，请在“导航器”视图中右键单击该协调队列管理器名称，然后单击**断开连接**。然后，右键单击想要使用的协调队列管理器的名称，并单击**连接**。
3. 使用以下任一方法来启动**新建受管文件传输**向导：
 - a) 在“导航器”视图中右键单击下列任意节点的名称：相关协调队列管理器、**传输模板**、**传输日志**或**暂挂传输**。然后单击**新建传输**来启动向导。
 - b) 单击**文件 > 新建 > 其他 > 受管文件传输向导 > 新建传输向导**

4. 遵循向导面板上的指示信息。确保您选中**启用调度传输**复选框并在**调度**选项卡上输入调度详细信息。如果不存在可能影响传输的问题，那么调度的文件传输将在调度开始时间后一分钟内开始。例如，您的网络或代理可能存在阻止调度传输开始的问题。为每个面板提供了上下文相关帮助。要在 Windows 上访问上下文相关帮助，请按 F1。在 Linux 上，按 Ctrl+F1 或 Shift+F1。

结果

有关调度的文件传输中所涉及消息的信息，请参阅[调度的传输的消息格式](#)。

从 IBM MQ Explorer 处理暂挂传输

您可以从 IBM MQ Explorer 查看暂挂的已调度文件传输。**暂挂的传输**窗口显示向当前连接的协调队列管理器注册的所有暂挂传输。

关于此任务

要查看尚未启动的所调度文件传输的状态，请使用以下步骤：

过程

1. 展开“导航器”视图中的**受管文件传输**。在“内容”视图中将显示**受管文件传输中心**。
2. 在“导航器”视图将显示所有协调队列管理器。展开您用于已调度传输的协调队列管理器的名称。如果要更改所连接的协调队列管理器，请在“导航器”视图中右键单击要使用的协调队列管理器的名称，然后单击**连接**。
3. 单击**暂挂的传输**。**暂挂的传输**窗口显示在“内容”视图中。
4. **暂挂的传输**窗口显示以下有关所调度文件传输的详细信息：
 - a) **名称** 所调度文件传输的编号。此编号是自动指定的。
 - b) **源** 源代理的名称。
 - c) **源文件** 要在主机系统上传输的文件名称。
 - d) **目标** 目标代理的名称。
 - e) **目标文件** 将文件传输至目标系统后的文件的名称。
 - f) **调度的开始时间（所选时区）** 调度的文件传输开始日期和时间（使用管理员选定的时区）。要更改显示的时区，请单击**窗口 > 首选项 > IBM MQ Explorer > 受管文件传输**，然后从**时区：**列表选择一个备选时区。单击**确定**。
 - g) **重复间隔** 如果您已选择重复执行“调度传输”，那么这是重复执行传输时依据的指定时间间隔（以数字表示）。
 - h) **重复类型** 如果选择重复调度传输，那么这是为文件传输指定的重复时间间隔的类型。该类型可以是以下某个值：**分钟、小时、天、星期、月或年**。
 - i) **重复到** 如果选择重复调度传输，那么这是要停止重复文件传输的时间的详细信息。例如，指定日期和时间或指定出现次数后。

结果

要刷新在**暂挂的传输**窗口中显示的内容，请单击“内容”视图工具栏上的“刷新”按钮 .

要取消暂挂的文件传输，请用鼠标右键单击特定的传输，然后单击**取消**。取消传输将完全废弃该文件传输请求。

触发文件传输

您可以设置某些关于文件传输的触发条件，这些触发条件必须成立才能进行传输。如果触发条件不成立，那么文件传输将无法进行并且可以选择提交日志消息以记录传输未发生这一事实。然后将废弃文件传输请求。例如，您可以将文件传输设置为仅在以下情况下发生：源代理所在的系统上的指定文件超过了指定大小，或

者源代理所在的系统上存在某一特定指定文件。可以通过 IBM MQ Explorer 或命令行来设置已触发的文件传输。

关于此任务

您可以持续监视资源以满足触发条件。要了解有关资源监视的更多信息，请参阅：[第 229 页的『资源监控』](#)。

您可以设置三种不同的触发条件。这些条件如下所示：

- 如果与源代理相同的系统上存在特定文件
- 如果与源代理相同的系统上不存在特定文件
- 如果在源代理所在的系统上，特定文件超过特定大小（大小可以表示为字节、KB、MB 或 GB）。这些计量单位使用 2^{10} 约定，例如，1KB 等于 1024 字节，1MB 等于 1024KB。

先前列表中的触发类型可以按以下两种方式来组合：

- 对于单个条件，您可以在源代理所在的系统上指定一个以上的文件。如果任何一个指定文件满足条件（布尔运算符 OR），那么将触发传输。
- 您可以指定多个条件。仅当满足所有条件时才会触发传输（布尔运算符 AND）。

您也可以将触发的传输与调度的传输相结合。请参阅创建调度的文件传输，以了解更多信息。在此情况下，将在调度应开始时评估触发条件，或者对于重复调度，那么在每次调度应开始时评估触发条件。

协议网桥代理上不支持触发的传输。

要使用命令行创建触发式文件传输，请在 `fteCreateTransfer` 命令上使用 `-tr` 参数。

要使用 IBM MQ Explorer 中的**新建受管文件传输**向导来创建已调度的文件传输，请使用以下步骤：

过程

1. 在“导航器”视图中，单击**受管文件传输**。在“内容”视图中将显示**受管文件传输中心**。
2. 在“导航器”视图中将显示所有协调队列管理器。展开您用于已调度传输的协调队列管理器的名称。如果要更改所连接的协调队列管理器，请在“导航器”视图中右键单击要使用的协调队列管理器的名称，然后单击**连接**。
3. 通过以下任意一种方法，启动**创建新的受管文件传输**向导：
 - a) 在“导航器”视图中右键单击下列任意节点的名称：相关协调队列管理器、**传输模板**、**传输日志**或**暂挂传输**。然后单击**新建传输**来打开向导。
 - b) 单击**文件 > 新建 > 其他 > 受管文件传输向导 > 新建传输向导**
4. 遵循向导面板上的指示信息。确保已选中**触发器**选项卡上的**启用触发传输**复选框，然后填写该选项卡的字段来设置触发。为每个面板提供了上下文相关帮助。要在 Windows 上访问上下文相关帮助，请按 F1。在 Linux 上，按 **Ctrl+F1** 或 **Shift+F1**。

从 IBM MQ Explorer 监视正在进行的文件传输

您可以使用 IBM MQ Explorer 中的**受管文件传输 - 当前传输进度**选项卡来监视正在进行的文件传输。可以从 IBM MQ Explorer 或命令行启动此文件传输。该选项卡还显示了启动调度传输时的调度传输的进度。

关于此任务

如果要使用 IBM MQ Explorer 来监视与远程系统上的协调队列管理器关联的传输，请遵循[第 227 页的『配置 IBM MQ Explorer 以监视远程协调队列管理器』](#)主题中的指示信息。

在停止并重新启动 IBM MQ Explorer 后，不会保留先前的文件传输信息。重新启动时，将从**当前的传输进度**选项卡中清除有关过去的传输的信息。您可以在 IBM MQ Explorer 处于打开状态时，随时使用**除去已完成的传输**  清除已完成的传输。

过程

在使用 IBM MQ Explorer 或命令行启动新的文件传输后，可以在**当前传输进度**选项卡中监视传输的进度。对于正在进行的每个传输，将显示以下信息：

- a) **源**。用于从源系统传输文件的代理的名称。
- b) **目标**。目标系统上用于接收文件的代理的名称。
- c) **当前文件**。当前正在传输的文件的名称。已传输的单个文件的一部分以 B、KiB 或 MiB 显示。GiB 或 TiB 以及文件的总大小位于括号中。所显示的度量单位取决于文件大小。
B 是指每秒字节数。KiB/s 是指每秒 KB 数，其中，1KB 等于 1024 字节。MiB/s 是指每秒 MB 数，其中，1MB 等于 1 048 576 字节。GiB/s 是指每秒 GB 数，其中，1GB 等于 1 073 741 824 字节。TiB/s 是指每秒 TB 数，其中，1TB 等于 1 099 511 627 776 字节。
- d) **文件号**。如果传输一个以上的文件，那么此编号表示已经传输到整个文件组中的第几个文件。
- e) **进度**。进度条显示当前的文件传输的完成情况（以百分比表示）。
- f) **速率**。文件的传输速率，以 KiB/s 也即每秒 KB 数表示（1KB 等于 1024 字节）。
- g) **开始时间（所选时区）**。文件传输开始的时间，以管理员选定的时区显示。要更改显示的时区，请单击**窗口 > 首选项 > IBM MQ Explorer > 受管文件传输**，然后从**时区**：列表中选择一个备选时区。单击**确定**。

如果在传输文件的过程中传输进入恢复状态，那么开始时间将会更新，以反映文件传输恢复的时间。

结果

该选项卡定期自动刷新其信息，但要强制刷新**当前传输进度**选项卡中显示的内容的视图，请单击“内容”视图工具栏上的**刷新** .

要从**当前传输进度**选项卡中删除文件传输，请单击“内容”视图工具栏上的**移除已完成的传输** 。单击该按钮只会从该选项卡移除文件传输详细信息，不会停止或取消当前或已调度的传输。

如果要在关闭后重新返回**当前传输进度**选项卡，您可以通过单击**窗口 > 显示视图 > 其他 > 其他 > 受管文件传输 - 当前传输进度**来显示该选项卡。单击**确定**。

相关任务

第 227 页的『[配置 IBM MQ Explorer 以监视远程协调队列管理器](#)』

使用 IBM MQ Explorer 来监视与远程系统上运行的协调队列管理器关联的文件传输。在 WebSphere MQ V7.5 或更高版本中，您需要能够运行 IBM MQ Explorer 的系统。需要安装 IBM MQ Explorer 组件以便能够连接至远程协调队列管理器。

第 228 页的『[使用传输日志来查看文件传输的状态](#)』

您可以使用 IBM MQ Explorer 中的**传输日志**来查看文件传输的详细信息。这些可以通过命令行或 IBM MQ Explorer 启动的传输。您还可以定制**传输日志**中显示的内容。

配置 IBM MQ Explorer 以监视远程协调队列管理器

使用 IBM MQ Explorer 来监视与远程系统上运行的协调队列管理器关联的文件传输。在 WebSphere MQ V7.5 或更高版本中，您需要能够运行 IBM MQ Explorer 的系统。需要安装 IBM MQ Explorer 组件以便能够连接至远程协调队列管理器。

关于此任务

假定：有权通过将队列管理器配置为允许远程连接，从而连接至远程协调队列管理器。

有关如何进行此配置的更多信息，请参阅第 104 页的『[以客户机方式使用通道认证来连接至 WebSphere MQ V7.1 或更高版本队列管理器](#)』和第 432 页的『[特定于 IBM MQ Managed File Transfer 的资源权限](#)』。

要监视未运行 Windows 或 Linux 的系统上的代理程序之间的队列管理器和文件传输，请使用以下步骤配置 IBM MQ Explorer 以连接到远程系统：

过程

1. 启动本地 IBM MQ Explorer。
2. 装入 IBM MQ Explorer 之后，右键单击**受管文件传输**文件夹并选择**新建配置**。
3. 继续完成向导，选择协调和命令队列管理器，然后定义该配置的名称。
4. 单击**完成**以完成定义。
5. 完成定义后，右键单击该定义并选择**连接**。

结果

现在启动 IBM MQ Explorer 并使用它监视与协调队列管理器关联的 IBM MQ Managed File Transfer 网络的传输活动。

相关任务

第 226 页的『从 IBM MQ Explorer 监视正在进行的文件传输』

您可以使用 IBM MQ Explorer 中的**受管文件传输 - 当前传输进度**选项卡来监视正在进行的文件传输。可以从 IBM MQ Explorer 或命令行启动此文件传输。该选项卡还显示了启动调度传输时的调度传输的进度。

第 228 页的『使用传输日志来查看文件传输的状态』

您可以使用 IBM MQ Explorer 中的**传输日志**来查看文件传输的详细信息。这些可以通过命令行或 IBM MQ Explorer 启动的传输。您还可以定制**传输日志**中显示的内容。

使用传输日志来查看文件传输的状态

您可以使用 IBM MQ Explorer 中的**传输日志**来查看文件传输的详细信息。这些可以通过命令行或 IBM MQ Explorer 启动的传输。您还可以定制**传输日志**中显示的内容。

过程

1. 在“导航器”视图中展开**受管文件传输**，然后展开要对其查看传输日志的协调队列管理器的名称。
2. 单击“导航器”视图中的**传输日志**。“**传输日志**”将显示在“内容”视图中。
3. “**传输日志**”窗口显示有关文件传输的以下详细信息：
 - a) **源**：源文件所在系统上的代理的名称。
 - b) **目标**：文件传输目标系统上的代理的名称。
 - c) **完成状态**：文件传输的状态。状态可以是以下值之一：“已启动”、“正在进行”、“成功”、“部分成功”、“已取消”或“失败”。
 - d) **所有者**：提交传输请求的主机上的用户标识。
 - e) **开始时间（选中的时区）**：IBM MQ Managed File Transfer 代理接受文件传输请求的时间和日期，以管理员选择的时区表示。要更改显示的时区，请单击**窗口 > 首选项 > IBM MQ Explorer > 受管文件传输**，然后从**时区**：列表中选择一个备选时区。单击**确定**。
 - f) **记录的状态（选中的时区）**：（缺省情况下，不会显示该列。您可以通过使用**配置传输日志列**  窗口来显示该列。）以管理员选择的时区记录完成状态的时间和日期。
 - g) **作业名** 用户通过使用 **fteCreateTransfer** 的 **-jn** 参数或在 Ant 脚本中指定的标识
 - h) **传输标识**：文件传输的唯一标识。
 - i) **Connect: Direct** 列举有关**进程号**、**进程名称**、**主节点**、**辅助节点**、**源类型**和**目标类型**的详细信息。

结果

注: IBM MQ V8.0.0.1 中针对 APAR IC99545 更改了传输日志的内部格式。因此，如果将 IBM MQ Explorer 升级至 V8.0.0.1 或更高版本，然后再复原到 V8.0.0.0，那么针对在 IBM MQ Explorer 为 V8.0.0.1 期间发生的传输，将不会显示审计 XML。在“**属性**”窗口中，针对这些传输的 XML 面板仅包含空文本框。

要查看有关已完成传输的更多详细信息，请展开您有兴趣使用加号 (+) 的传输。然后，您可以查看该传输中包含的所有源和目标文件名。但是，如果当前正在进行传输并且传输包含多个文件，那么只可以查看目前已传输的文件。

要刷新在**传输日志**中显示的内容，请单击“内容”视图工具栏上的**刷新按钮** 。在您停止并重新启动 IBM MQ Explorer 后，“传输日志”中的文件传输信息会被保留下来。如果要从日志中删除所有已完成的文件传输，那么请单击“内容”视图工具栏上的**移除已完成的传输** .

要从日志中删除单个已完成的文件传输，请右键单击该传输，然后单击**删除**。删除传输时，不会停止或取消正在进行的传输或已安排的传输；只会删除存储的历史数据。

要将传输的唯一标识复制到剪贴板，请右键单击该传输，然后单击**复制标识**。

可通过**属性**操作下的上下文菜单获取传输的元数据和完整的审计 XML。

相关任务

第 226 页的『从 IBM MQ Explorer 监视正在进行的文件传输』

您可以使用 IBM MQ Explorer 中的**受管文件传输 - 当前传输进度**选项卡来监视正在进行的文件传输。可以从 IBM MQ Explorer 或命令行启动此文件传输。该选项卡还显示了启动调度传输时的调度传输的进度。

第 229 页的『配置传输日志』

您可以对 IBM MQ Explorer 的**传输日志**中显示的信息及其显示方式进行配置。

配置传输日志

您可以对 IBM MQ Explorer 的**传输日志**中显示的信息及其显示方式进行配置。

关于此任务

要调整**传输日志**中列的顺序，请单击希望移动的列的标题，然后将该列拖到新的位置。新的列顺序仅保留到您下一次停止并重新启动 IBM MQ Explorer 时。

要过滤**传输日志**中的条目，请在**过滤显示的日志条目**字段中输入字符串。要复原日志的所有条目，请从该字段中删除您输入的字符串。可以在此字段中使用任何有效的 Java 正则表达式。有关更多信息，请参阅第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』。

要定制在“传输日志”中显示的列，请使用**配置传输日志列** 。通过以下步骤启动并使用**配置传输日志列**窗口。

过程

1. 确保**传输日志**窗口在“内容”视图中打开。单击“内容”视图工具栏上的**配置传输日志列** 。这样会打开**配置传输日志的列**窗口。
2. 要定制自己的**传输日志**视图，请选中或清除针对您要显示或隐藏的个别复选框。您可以单击**全选**，然后单击**确定**以选择所有复选框，或者选择**全部取消选择**，然后单击**确定**以清除所有复选框。

相关任务

第 226 页的『从 IBM MQ Explorer 监视正在进行的文件传输』

您可以使用 IBM MQ Explorer 中的**受管文件传输 - 当前传输进度**选项卡来监视正在进行的文件传输。可以从 IBM MQ Explorer 或命令行启动此文件传输。该选项卡还显示了启动调度传输时的调度传输的进度。

第 228 页的『使用传输日志来查看文件传输的状态』

您可以使用 IBM MQ Explorer 中的**传输日志**来查看文件传输的详细信息。这些可以通过命令行或 IBM MQ Explorer 启动的传输。您还可以定制**传输日志**中显示的内容。

资源监控

您可以监视 IBM MQ Managed File Transfer 资源；例如，队列或目录。当满足针对该资源的条件时，资源监视器将启动任务，如文件传输。您可以使用 **fteCreateMonitor** 命令或 IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件中的**监视器**视图来创建资源监视器。

常见场景是监视目录以了解是否存在触发器文件。外部应用程序可能正在处理多个文件，并将它们放置在已知源目录中。应用程序完成其处理后，通过将触发器文件放置在受监控位置，其指示文件已准备好可进行传输或执行操作。可以通过 IBM MQ Managed File Transfer 监视器检测触发器文件，并开始将这些文件从源目录传输到另一个 IBM MQ Managed File Transfer 代理。

下面为监视目录的两个示例：

- 监视触发器文件（例如，`trigger.file`），然后传输通配符（例如，`*.zip`）
- 监视 `*.zip`，然后传输 `${FilePath}`（例如，触发传输的文件）。有关变量替换的更多详细信息，请参阅第 239 页的『使用变量替换定制 MFT 任务』。

请勿创建用于先监视 `*.zip` 然后传输 `*.zip` 的监视器。该监视器将尝试为系统上的每个 `.zip` 文件启动 `*.zip` 传输。即，该监视器将对 `*.zip` 生成 * 个传输。

要查看创建用于监视目录的资源监视器的示例，请参阅第 236 页的『监视目录和使用变量替换』。

下面为监视队列的示例：

- 外部应用程序可能正在生成消息并将这些消息放置在具有相同组标识的已知队列。应用程序将这些消息放置在队列上后，表明该组是完整的。可以通过 IBM MQ Managed File Transfer 监视器检测完整的消息组，并开始将消息组从源队列传输到文件。

要查看创建用于监视队列的资源监视器的示例，请参阅第 238 页的『示例：配置资源监视器以监视队列』。

IBM MQ Managed File Transfer 资源监视使用以下术语：

监视器

一个进程，按预定义的固定时间间隔轮询资源（如目录或队列），以查看资源内容是否发生更改。如果发生更改，会将内容与该监视器的条件集进行比较。如果匹配，那么将启动该监视器的任务。

资源

监视器在每个轮询时间间隔内都会检查系统资源以将其与触发条件比较。队列、目录或嵌套目录结构可以是受监控资源。

条件

求值的表达式（通常针对受监控资源的内容）。如果表达式求值为 `true`，那么该条件将影响整体触发条件。

触发条件

整体条件，当所有条件都满足时才能满足该整体条件。满足触发条件后，任务可以继续。

任务

满足触发条件或条件集后启动的操作。受支持的任务包括文件传输和命令调用。

触发器文件

放置在受监控目录中的文件，用于指示任务（通常是传输）可以开始。例如，它可能指示所有要处理的文件已到达已知位置，可以进行传输或执行操作。触发器文件的名称可用于通过使用变量替换指定要传输的文件。有关更多信息，请参阅第 239 页的『使用变量替换定制 MFT 任务』。

触发器文件也称为就绪文件或执行文件。但是，在本文档中，始终将其称为触发器文件。

协议网桥代理、Connect:Direct 网桥代理或 Web 网关代理上不支持资源监视。

相关概念

第 231 页的『资源监视概念』

这里概述了 IBM MQ Managed File Transfer 资源监视功能的主要概念。

第 239 页的『使用变量替换定制 MFT 任务』

在满足活动资源监视器的触发条件时，会调用已定义的任务。除了每次使用相同的目标代理或相同的目标文件名调用传输或命令任务之外，您还可以在运行时修改任务定义。可通过将变量名插入到任务定义 XML 中来实现此目的。在监视器确定已满足触发条件并且任务定义包含变量名时，将用变量值替换变量名，然后调用任务。

相关任务

第 233 页的『配置监视器任务以启动命令和脚本』

资源监视器的作用不只限于将执行文件传输作为其相关的任务。您还可以配置监视器，以从监视代理调用其他命令，包括可执行程序、Ant 脚本或 JCL 作业。要调用命令，请编辑监视任务定义 XML，以将一个或多个命令元素包含在对应的命令调用参数（例如，自变量和属性）中。

第 238 页的『示例：配置资源监视器以监视队列』

您可以通过使用带 `-mq` 参数的 `fteCreateMonitor` 命令，将 IBM MQ 队列指定为要由资源监视器监视的资源。

第 243 页的『[监视队列和使用变量替换](#)』

您可以使用 **fteCreateMonitor** 命令监视队列，并将消息从受监视的队列传输到文件。从受监视队列读取的第一条消息中任何 IBM MQ 消息属性的值均可在任务 XML 定义中被替换，并用于定义传输行为。

相关参考

第 400 页的『[有关配置资源监视器以避免代理超负载的指南](#)。』

您可以配置 IBM MQ Managed File Transfer 资源监视器的属性和参数值以减少代理上的负载。减少代理上的负载可提高该代理的性能。有多个设置可供您使用，您可能需要通过反复试验来找到适合您系统配置的最佳设置。

第 478 页的『[fteCreateMonitor（新建资源监视器）](#)』

fteCreateMonitor 命令从命令行创建并启动新的资源监视器。您可以使用 WebSphere MQ Managed File Transfer 监视资源（例如，目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

第 534 页的『[fteListMonitors（列出 IBM MQ Managed File Transfer 资源监视器）](#)』

使用 **fteListMonitors** 命令可使用命令行列出 IBM MQ Managed File Transfer 网络中的所有现有资源监视器。

第 526 页的『[fteDeleteMonitor（删除 IBM MQ Managed File Transfer 资源监视器）](#)』

使用 **fteDeleteMonitor** 命令可通过命令行停止和删除现有 IBM MQ Managed File Transfer 资源监视器。针对资源监视代理发出此命令。

资源监视概念

这里概述了 IBM MQ Managed File Transfer 资源监视功能的主要概念。

监视器

资源监视器与 IBM MQ Managed File Transfer 代理关联，并且只有当启动该代理后它才会处于活动状态。当监视代理停止时，监视器也将停止。如果代理已在创建监视器时启动，那么监视器将立即启动。监视代理还必须是监视器所启动任务的源代理。

监视器名称在其代理内必须唯一。监视器名称最少必须具有一个字符，并且不能包含星号(*)、百分号(%)或问号(?)字符。将忽略所提供监视器名称的大小写，并且会将监视器名称转换为大写。如果您尝试使用已存在的名称来创建监视器，那么将忽略该请求，并且会将该尝试记录到监视器日志主题。

对可在代理上创建的监视器数量没有限制，并且所有监视器以相同的优先级运行。考虑重叠受监控资源的含义、冲突触发条件的含义以及轮询资源的频率。

监视器在每个轮询时间间隔后查看资源内容。会将资源内容与触发条件比较，如果满足这些条件，那么会调用与监视器关联的任务。

异步启动任务。如果存在一个条件匹配项，并且启动任务，那么监视器将继续轮询资源内容的进一步更改。例如，如果因名为 **reports.go** 的文件到达受监控目录而发生匹配，那么将立即启动任务。在下一个轮询时间间隔，即使该文件仍存在，也不再启动任务。但是，如果删除了该文件并将其重新放置到该目录中，或更新了该文件（如更改“上次修改日期”属性），那么下一次触发条件检查将导致再次调用任务。

资源

IBM MQ Managed File Transfer 中的监视器可以轮询目录或嵌套目录结构的内容。缺省情况下，将监视指定目录。要同时检查子目录，请在 **fteCreateTransfer** 命令中设置递归层次。

IBM MQ Managed File Transfer 中的监视器可以轮询 WebSphere MQ 队列的内容。您可以指定每个队列仅一个监视器。如果指定多个监视器轮询 WebSphere MQ 队列，将发生不可预测的行为。

不支持监视数据集。

触发条件

当资源包含与某个其他字符串或模式匹配的值时，表示满足该条件。条件可以是以下项之一：

- 针对文件名（模式）的匹配
- 针对文件名（模式）的不匹配

- 文件大小
- 如果对于大量轮询，文件大小保持相同，那么表示匹配

文件名匹配可以表达为：

- 精确的字符串匹配
- 简单的通配符匹配，如第 721 页的『使用通配符』中所述
- 正则表达式匹配

还可以使用通配符或 Java 正则表达式 (用于标识从不匹配的文件名) 从文件名匹配中排除文件名。

当检测到匹配文件时，将保留其上次修改时间戳记。如果后续轮询检查到文件已更改，那么再次满足触发条件，并且会启动任务。如果条件是检测何时文件不存在，那么当受监控目录中没有文件匹配文件名模式，那么会启动任务。如果随后将文件添加到与文件名模式不匹配的目录，那么只有当删除该文件后才会启动任务。

任务

IBM MQ Managed File Transfer 支持以下两种可配置为通过资源监视器启动的任务类型：

- 文件传输
- 命令

与任何其他文件传输相同的方式定义文件传输任务。生成监视器所需的任务 XML 的有用方法是运行带有 **-gt** 参数的 **fteCreateTransfer** 命令。该命令会将任务定义生成 XML 文档，包含传输规范。然后，将任务 XML 文档的名称作为 **fteCreateMonitor** 命令上的 **-mt** 参数值传递。当运行 **fteCreateMonitor** 时，其将读取任务 XML 文档。在运行 **fteCreateMonitor** 之后，监视器未使用对任务 XML 文件进行的任何更改。

命令任务可以运行 Ant 脚本、调用可执行程序或运行 JCL 作业。要获取更多信息，请参阅[配置监视器任务以调用命令和脚本](#)。

在使用文件传输任务时，您可以选择将多少个触发条件批处理到任务中。缺省值是对一个触发条件启动一个任务。您可以运行带有 **-bs** 选项的 **fteCreateMonitor** 命令，以选择一起批处理到一个任务中的触发条件数。

相关概念

[第 229 页的『资源监控』](#)

您可以监视 IBM MQ Managed File Transfer 资源；例如，队列或目录。当满足针对该资源的条件时，资源监视器将启动任务，如文件传输。您可以使用 **fteCreateMonitor** 命令或 IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

[第 239 页的『使用变量替换定制 MFT 任务』](#)

在满足活动资源监视器的触发条件时，会调用已定义的任务。除了每次使用相同的目标代理或相同的目标文件名调用传输或命令任务之外，您还可以在运行时修改任务定义。可通过将变量名插入到任务定义 XML 中来实现此目的。在监视器确定已满足触发条件并且任务定义包含变量名时，将用变量值替换变量名，然后调用任务。

相关任务

[第 233 页的『配置监视器任务以启动命令和脚本』](#)

资源监视器的作用不只限于将执行文件传输作为其相关的任务。您还可以配置监视器，以从监视代理调用其他命令，包括可执行程序、Ant 脚本或 JCL 作业。要调用命令，请编辑监视任务定义 XML，以将一个或多个命令元素包含在对应的命令调用参数（例如，自变量和属性）中。

[第 238 页的『示例：配置资源监视器以监视队列』](#)

您可以通过使用带 **-mq** 参数的 **fteCreateMonitor** 命令，将 IBM MQ 队列指定为要由资源监视器监视的资源。

[第 243 页的『监视队列和使用变量替换』](#)

您可以使用 **fteCreateMonitor** 命令监视队列，并将消息从受监视的队列传输到文件。从受监视队列读取的第一条消息中任何 IBM MQ 消息属性的值均可在任务 XML 定义中被替换，并用于定义传输行为。

相关参考

[第 478 页的『fteCreateMonitor（新建资源监视器）』](#)

fteCreateMonitor 命令从命令行创建并启动新的资源监视器。您可以使用 WebSphere MQ Managed File Transfer 监视资源（例如，目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

第 534 页的『[fteListMonitors](#)（列出 IBM MQ Managed File Transfer 资源监视器）』

使用 **fteListMonitors** 命令可使用命令行列出 IBM MQ Managed File Transfer 网络中的所有现有资源监视器。

第 526 页的『[fteDeleteMonitor](#)（删除 IBM MQ Managed File Transfer 资源监视器）』

使用 **fteDeleteMonitor** 命令可通过命令行停止和删除现有 IBM MQ Managed File Transfer 资源监视器。针对资源监视代理发出此命令。

配置监视器任务以启动命令和脚本

资源监视器的作用不只限于将执行文件传输作为其相关的任务。您还可以配置监视器，以从监视代理调用其他命令，包括可执行程序、Ant 脚本或 JCL 作业。要调用命令，请编辑监视任务定义 XML，以将一个或多个命令元素包含在对应的命令调用参数（例如，自变量和属性）中。

关于此任务

到您希望监视代理调用的可执行程序、Ant 脚本或 JCL 作业的文件路径必须包含在监视代理的 `commandPath` 中。要了解有关命令路径属性的更多信息，请参阅第 443 页的『[commandPath](#) 属性』。

您可以通过以下方式之一创建任务定义 XML 文档：

- 根据 `FileTransfer.xsd` 模式手工创建任务定义 XML 文档。有关更多信息，请参阅第 233 页的『[根据模式手工创建任务定义 XML](#)』。
- 编辑由 **fteCreateTransfer -gt** 参数生成的 XML 文档，以作为任务定义的基础。有关更多信息，请参阅第 235 页的『[通过修改生成的文档创建任务定义文档](#)』。

不论是传输任务还是命令任务，任务定义必须以 `<request>` 根元素开头。`<request>` 的子元素必须是 `<managedTransfer>` 或 `<managedCall>`。当有单一命令或脚本要运行时，您通常会选择 `<managedCall>`，如果希望任务包含一个文件传输和最多四个命令调用（可选），通常会选择 `<managedTransfer>`。

根据模式手工创建任务定义 XML

关于此任务

您可以根据 `FileTransfer.xsd` 模式手工创建任务定义 XML 文件。此模式可在 `MQ_INSTALLATION_PATH/mqft/samples/schema` 中找到。要了解有关此模式的更多信息，请参阅第 842 页的『[文件传输请求消息格式](#)』。

示例

以下示例显示了另存为 `cleanup.xml`，的示例任务定义 XML 文档，该 XML 利用 `<managedCall>` 元素调用名为 `RunCleanup.xml` 的 Ant 脚本。`RunCleanup.xml` Ant 脚本必须位于监视代理的 `commandPath` 中。

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedCall>
    <originator>
      <hostName>hostName</hostName>
      <userID>userID</userID>
      <mqmdUserID>mqmdUserID</mqmdUserID>
    </originator>
    <agent QMgr="QM1" agent="AGENT1"/>
    <reply QMGR="QM1">reply</reply>
    <transferSet priority="1">
      <metaDataSet>
        <metaData key="name1">value1</metaData>
      </metaDataSet>
      <call>
        <command name="RunCleanup.xml" type="antscript" retryCount="2"
          retryWait="30" successRC="0">
          <target>check_exists</target>
        </command>
      </call>
    </transferSet>
  </managedCall>
</request>
```

```

        <target>copy_to_archive</target>
        <target>rename_temps</target>
        <target>delete_files</target>
        <property name="trigger.filename" value="${FileName}"/>
        <property name="trigger.path" value="${FilePath}"/>
        </command>
    </call>
</transferSet>
<job>
    <name>JOBBCLEAN1</name>
</job>
</managedCall>
</request>

```

<agent> 元素指定在其 **commandPath** 上使用指定的 Ant 脚本配置的 IBM MQ Managed File Transfer 代理程序。

<call><command>... 结构定义了想要运行的可执行文件或脚本。该命令采用可选的 **type** 属性，此属性可以具有以下某个值：

antscript

在单独的 JVM 中运行 Ant 脚本。

executable

调用一个可执行程序。

jcl

调用一个 JCL 作业。

如果省略 **type** 属性，那么会使用缺省值 **executable**。

name 属性指定要运行的 Ant 脚本、可执行文件或 JCL 作业的名称，但不包含任何路径信息。代理会在其 **agent.properties** 文件中的 **commandPath** 属性指定的位置搜索脚本或程序。

retrycount 属性指定当程序未返回成功返回码时，尝试调用该程序的次数。为该属性指定的不能是负数。如果未指定 **retrycount** 属性，那么将使用缺省值 0。

retrywait 属性指定再次尝试调用程序前要等待的时间（秒）。为该属性指定的不能是负数。如果未指定 **retrywait** 属性，那么将使用缺省值 0。

successsrc 属性是一个用于确定何时成功运行程序调用的表达式。该表达式用于对命令的进程返回码进行求值。该值可以由一个或多个表达式组成，当中用竖线字符串 (|) 分隔以表示布尔运算 OR，或用 & 字符串分隔以表示布尔运算 AND。每个表达式可以是以下类型之一：

- 指示进程返回码和数字间的等同性测试的数字。
- 以大于号 (>) 为前缀的一个数字，表示数字和进程返回码之间的大于测试。
- 以小于号 (<) 为前缀的一个数字，表示数字和进程返回码之间的小于测试。
- 以惊叹号 (!) 为前缀的一个数字，表示数字和进程返回码之间的不等测试。例如：>2&<7&!5|0|14 解释为返回码为 0、3、4、6、14 时表示成功。所有其他返回码都解释为不成功。

如果未指定 **successsrc** 属性，那么将使用缺省值 0。这表示当且仅当该命令返回码为 0 时，才会“判定”该命令运行成功。

对于 Ant 脚本，您通常会指定 **<target>** 和 **<property>** 元素。**<target>** 元素值必须和 Ant 脚本中的目标名称匹配。

对于可执行程序，您可以指定 **<argument>** 元素。嵌套的自变量元素指定将在程序调用过程中传递给要调用的程序的自变量。程序自变量是根据自变量元素出现的顺序，通过自变量元素指定的值构建的。您可以指定零个、一个或多个自变量元素，作为程序调用的嵌套元素。

管理员可以使用包含 **<managedCall>** 元素的任务定义 XML 文档来定义并正常启动监视器。例如：

```

fteCreateMonitor -ma AGENT1 -mm QM1 -md /monitored -mn MONITOR01 -mt
/tasks/cleanuptask.xml -pi 30 -pu seconds -tr match,*.go

```

传输定义 XML 文档的路径必须位于您运行 **fteCreateMonitor** 命令的本地文件系统上（在本例中为 **/tasks/cleanuptask.xml**）。**cleanuptask.xml** 文档只用于创建资源监视器。**cleanuptask.xml** 文档引用的任何任务 (Ant 脚本或 JCL 作业) 必须位于监视代理程序的命令路径中。当监视器触发的条件满足

时，监视器中的实际值会替换任务定义 XML 中的所有参数。例如，`${FilePath}` 在要发送到代理的请求消息中替换为 `/monitored/cleanup.go`。请求消息会排入代理命令队列。命令处理器检测到该请求的目的是程序调用，并启动指定的程序。如果要调用 `antscript` 命令类型，会启动新的 JVM，Ant 任务在此 JVM 中运行。要了解有关使用变量替换的更多信息，请参阅[通过变量替换定置任务](#)。

相关概念

第 239 页的『使用变量替换定制 MFT 任务』

在满足活动资源监视器的触发条件时，会调用已定义的任务。除了每次使用相同的目标代理或相同的目标文件名调用传输或命令任务之外，您还可以在运行时修改任务定义。可通过将变量名插入到任务定义 XML 中来实现此目的。在监视器确定已满足触发条件并且任务定义包含变量名时，将用变量值替换变量名，然后调用任务。

相关参考

第 842 页的『文件传输请求消息格式』

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the `FileTransfer.xsd` schema and have the `<request>` element as the root element. `FileTransfer.xsd` 模式文档位于 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。`FileTransfer.xsd` 模式导入相同目录中的 `fteutils.xsd`。

第 443 页的『`commandPath` 属性』

使用 `commandPath` 属性以指定 IBM MQ Managed File Transfer 可从中运行命令的位置。设置该属性时务必谨慎，因为能够向代理发送命令的远程客户机系统可以有效调用指定的某个 `commandPath` 中的任何命令。

通过修改生成的文档创建任务定义文档

关于此任务

您可以通过修改由 `fteCreateTransfer` 的 `-gt` 选项生成的 XML 文档，创建监视任务的定义文档。在生成的文档中，`<request>` 后面会跟有 `<managedTransfer>` 元素。要将此任务定义转换为有效的 `<managedCall>` 结构，请完成以下步骤：

过程

1. 将开始标记和结束标记 `<managedTransfer>` 替换为标记 `<managedCall>`。
2. 移除所有的 `<schedule>` 元素和子节点。
3. 将开始标记和结束标记 `<sourceAgent>` 替换为 `<agent>`，以匹配监视代理配置详细信息。
4. 移除 `<destinationAgent>` 和 `<trigger>` 元素。
5. 移除 `<item>` 元素。
6. 在 `<transferSet>` 元素中插入新的 `<call>...</call>` 结构。该结构包含命令定义，如以下示例所示：

```
<call>
  <command name="RunCleanup.xml" type="antscript" retryCount="2"
  retryWait="30" successRC="0">
    <target>check_exists</target>
    <target>copy_to_archive</target>
    <target>rename_temps</target>
    <target>delete_files</target>
    <property name="trigger.filename" value="${FileName}"/>
    <property name="trigger.path" value="${FilePath}"/>
  </command>
</call>
```

示例

您也可以保留包含所有文件传输详细信息的 `<managedTransfer>` 元素，并且最多可以插入四个命令调用。在本例中，您选择将以下任何调用元素插入 `<metaDataSet>` 和 `<item>` 元素之间：

preSourceCall

在启动传输之前在源代理上调用程序。

postSourceCall

在传输完成之后在源代理上调用程序。

preDestinationCall

在启动传输之前在目标代理上调用程序。

postDestinationCall

在传输完成之后在目标代理上调用程序。

以上元素均采用 `<command>` 元素结构（详见之前的示例）。FileTransfer.xsd 模式定义由各种调用元素使用的类型。

以下示例显示任务定义文档中的 preSourceCall、postSourceCall、preDestinationCall 和 postDestinationCall 元素：

```
:
<transferSet priority="1">
  <metaDataSet>
    <metaData key="key1">value1</metaData>
  </metaDataSet>
  <preSourceCall>
    <command name="send.exe" retryCount="0" retryWait="0" successRC="0"
      type="executable">
      <argument>report1.pdf</argument>
      <argument>true</argument>
    </command>
  </preSourceCall>
  <postSourceCall>
    <command name="//DO_IT.JCL" retryCount="0" retryWait="0" successRC="0"
      type="jcl">
      <argument>argument</argument>
    </command>
  </postSourceCall>
  <preDestinationCall>
    <command name="ant_script.xml" retryCount="0" retryWait="0" successRC="0"
      type="antscript">
      <target>step1</target>
      <property name="name" value="value"/>
    </command>
  </preDestinationCall>
  <postDestinationCall>
    <command name="runit.cmd" retryCount="0" retryWait="0" successRC="0" />
  </postDestinationCall>
  <item checksumMethod="none" mode="binary">
:

```

您可以在一个传输中混合使用各种类型的命令。自变量、目标和属性元素都是可选的。

监视目录和使用变量替换

您可以使用 **fteCreateMonitor** 命令来监视目录。替换变量的值可以在任务 XML 定义中被替换并用于定义传输行为。

关于此任务

在该示例中，源代理称为 AGENT_HOP。AGENT_HOP 监视的目录称为 /test/monitored。代理每 5 分钟轮询一次目录。

在将 .zip 文件写入目录后，将该文件写入目录的应用程序会将触发器文件写入同一目录。触发器文件的名称与 .zip 文件名称相同，但文件扩展名不同。例如，在将文件 file1.zip 写入目录后，会将文件 file1.go 写入该目录。资源监视器监视与 *.go 匹配的文件的目录，然后使用变量替换来请求传输相关联的 .zip 文件。

过程

1. 创建任务 XML，以定义监视器被触发时应执行的任务。

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
```



```

        xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
<managedTransfer>
  <originator>
    <hostName>blue.example.com</hostName>
    <userID>USER1</userID>
  </originator>
  <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
  <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
  <transferSet>
    <item mode="binary" checksumMethod="none">
      <source>
        <file>/test/monitored/${fileName{token=1}{separator=.}}.zip</file>
      </source>
      <destination type="file" exist="overwrite">
        <file>/out/${fileName{token=1}{separator=.}}.zip</file>
      </destination>
    </item>
  </transferSet>
</managedTransfer>
</request>

```

变量（将替换为与触发器文件相关联的值）以**粗体**突出显示。该任务 XML 保存 to 文件 /home/USER1/task.xml

2. 创建资源监视器以监视目录 /test/monitored。

提交以下命令：

```

fteCreateMonitor -ma AGENT_HOP -mm QM_HOP -md /test/monitored
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr match,*.go -pi 5 -pu minutes

```

3. 用户或程序将文件 jump.zip 写入 /test/monitored 目录，然后将文件 jump.go 写入该目录。
4. 监视器由是否存在文件 jump.go 来触发。代理将在任务 XML 中替换有关触发器文件的信息。这导致任务 XML 转换为：

```

<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>blue.example.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
    <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <file>/test/monitored/jump.zip</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/out/jump.zip</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

结果

已执行任务 XML 所定义的传输。jump.zip 文件由 AGENT_HOP 从 /test/monitored 目录中读取，并传输至位于运行 AGENT_SKIP 的系统上的名为 /out/jump.zip 的文件。

相关概念

第 229 页的『资源监控』

您可以监视 IBM MQ Managed File Transfer 资源；例如，队列或目录。当满足针对该资源的条件时，资源监视器将启动任务，如文件传输。您可以使用 **fteCreateMonitor** 命令或 IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

第 239 页的『使用变量替换定制 MFT 任务』

在满足活动资源监视器的触发条件时，会调用已定义的任务。除了每次使用相同的目标代理或相同的目标文件名调用传输或命令任务之外，您还可以在运行时修改任务定义。可通过将变量名插入到任务定义 XML 中来实现此目的。在监视器确定已满足触发条件并且任务定义包含变量名时，将用变量值替换变量名，然后调用任务。

相关任务

第 233 页的『配置监视器任务以启动命令和脚本』

资源监视器的作用不只限于将执行文件传输作为其相关的任务。您还可以配置监视器，以从监视代理调用其他命令，包括可执行程序、Ant 脚本或 JCL 作业。要调用命令，请编辑监视任务定义 XML，以将一个或多个命令元素包含在对应的命令调用参数（例如，自变量和属性）中。

相关参考

第 478 页的『fteCreateMonitor（新建资源监视器）』

fteCreateMonitor 命令从命令行创建并启动新的资源监视器。您可以使用 WebSphere MQ Managed File Transfer 监视资源（例如，目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

示例：配置资源监视器以监视队列

您可以通过使用带 **-mq** 参数的 **fteCreateMonitor** 命令，将 IBM MQ 队列指定为要由资源监视器监视的资源。

关于此任务

在本示例中，要监视的资源是队列 *MONITORED_QUEUE*。该队列必须在监视代理的队列管理器 *QM_NEPTUNE* 上。监视队列的条件是存在一组完整的消息。条件满足时将执行的任务在文件 *task.xml* 中定义。

注：请勿创建多个资源监视器来监视单个队列。如果这样做，会发生不可预测的行为。

过程

输入以下命令：

```
fteCreateMonitor -ma AGENT_NEPTUNE -mn myMonitor -mm QM_NEPTUNE -mq MONITORED_QUEUE  
-mt task.xml -tr completeGroups -pi 5 -pu minutes
```

监视器每隔五分钟检查一次队列，以确定条件 *completeGroups* 是否为 *true*。如果队列上存在一个或多个完整的组，那么监视器会针对每个完整组运行 *task.xml* 文件中定义的任务一次。

相关概念

第 229 页的『资源监控』

您可以监视 IBM MQ Managed File Transfer 资源；例如，队列或目录。当满足针对该资源的条件时，资源监视器将启动任务，如文件传输。您可以使用 **fteCreateMonitor** 命令或 IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

第 239 页的『使用变量替换定制 MFT 任务』

在满足活动资源监视器的触发条件时，会调用已定义的任务。除了每次使用相同的目标代理或相同的目标文件名调用传输或命令任务之外，您还可以在运行时修改任务定义。可通过将变量名插入到任务定义 XML 中来实现此目的。在监视器确定已满足触发条件并且任务定义包含变量名时，将用变量值替换变量名，然后调用任务。

相关任务

第 233 页的『配置监视器任务以启动命令和脚本』

资源监视器的作用不只限于将执行文件传输作为其相关的任务。您还可以配置监视器，以从监视代理调用其他命令，包括可执行程序、Ant 脚本或 JCL 作业。要调用命令，请编辑监视任务定义 XML，以将一个或多个命令元素包含在对应的命令调用参数（例如，自变量和属性）中。

第 243 页的『监视队列和使用变量替换』

您可以使用 **fteCreateMonitor** 命令监视队列，并将消息从受监视的队列传输到文件。从受监视队列读取的第一条消息中任何 IBM MQ 消息属性的值均可在任务 XML 定义中被替换，并用于定义传输行为。

相关参考

第 478 页的『fteCreateMonitor（新建资源监视器）』

fteCreateMonitor 命令从命令行创建并启动新的资源监视器。您可以使用 WebSphere MQ Managed File Transfer 监视资源（例如，目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

使用变量替换定制 MFT 任务

在满足活动资源监视器的触发条件时，会调用已定义的任务。除了每次使用相同的目标代理或相同的目标文件名调用传输或命令任务之外，您还可以在运行时修改任务定义。可通过将变量名插入到任务定义 XML 中来实现此目的。在监视器确定已满足触发条件并且任务定义包含变量名时，将用变量值替换变量名，然后调用任务。

如果受监视的资源是队列

可在任务 XML 定义中替换从受监视队列中读取的第一条消息中任何 IBM MQ 消息属性的值。

用户定义的消息属性的前缀为 **usr.**，但在变量名中不包含此前缀。变量名称必须以美元符号 (\$) 字符开头，并括在花括号 {} 中。例如，**\${destFileName}** 将替换为要从源队列读取的第一条消息的 **usr.destFileName** 消息属性的值。有关更多信息，请参阅第 741 页的『从源队列上的消息中读取的 IBM MQ 消息属性』和第 243 页的『监视队列和使用变量替换』。

下表显示了缺省情况下提供的替换变量。例如，**\${AGENTNAME}** 将替换为资源监视器代理的名称。

表 19: 缺省情况下提供的替换变量	
变量	描述
AGENTNAME	资源监视器代理的名称。
queueName	正在监视的队列的名称。
ENCODING	队列上第一条消息或组中第一条消息的字符编码。
MESSAGEID	队列上第一条消息或组中第一条消息的 IBM MQ 消息标识。
GROUPID	组的 IBM MQ 组标识或消息标识（如果仅找到一条消息）。只有在监视整个组时才会设置此变量。
CurrentTimeStamp	基于触发监视器的本地时间的时间戳记。此时间戳记值对于代理是唯一的。
CurrentTimeStamp UTC	基于触发监视器的 UTC 时区中的时间的时间戳记。此时间戳记值对于代理是唯一的。

如果受监视资源是目录

下表显示了任务 XML 定义中可替换的变量名集。

表 20: 可替换的变量	
变量	描述
FilePath	触发器文件的完整路径名。
FileName	触发器的文件名部分。
LastModifiedTime	上次修改触发器文件的时间。该时间表示为用于运行代理的时区的本地时间，采用 ISO 8601 时间格式。
LastModifiedDate	上次修改触发器文件的日期。该日期表示为用于运行代理的时区的本地日期，采用 ISO 8601 日期格式。
LastModifiedTimeUTC	上次修改触发器文件的时间。该时间表示为已转换为 UTC 时区的本地时间，采用 ISO 8601 时间格式。

表 20: 可替换的变量 (继续)	
变量	描述
LastModifiedDateUTC	上次修改触发器文件的日期。该日期表示为已转换为 UTC 时区的本地日期，采用 ISO 8601 日期格式。
AgentName	资源监视器代理的名称。
CurrentTimeStamp	基于触发监视器的本地时间的时间戳记。此时间戳记值对于代理是唯一的。
CurrentTimeStampUTC	基于触发监视器的 UTC 时区中的时间的时间戳记。此时间戳记值对于代理是唯一的。

变量名称必须以美元符号 (\$) 字符开头，并括在花括号 {} 中。例如，\${FilePath} 将替换为匹配触发器文件的标准文件路径。

可以对变量名应用两个特殊关键字以提供进一步的优化。这些字段为：

标记

用于替换的标记索引（左侧从 1 开始，右侧从 -1 开始）

分隔符

用于标记变量值的单个字符。缺省值为正斜杠字符 (/)，但分隔符可以是能够出现在变量值中的任何有效字符。

如果在变量名中指定了分隔符关键字，那么会根据分隔符将变量值分割为标记。

分配给标记关键字的值将用作索引，以选择用于替换变量名的标记。标记索引相对于变量中的第一个字符，并且从 1 开始。如果未指定 token 关键字，那么将插入整个变量。

变量名不区分大小写。

采用不区分大小写的方式处理用于替换消息 XML 中代理名称的任何值。所有 Managed File Transfer 代理名称都是大写。如果用值 Paris 替换消息 XML 中的代理属性，那么会将该值解释为对代理 PARIS 的引用。

相关概念

第 240 页的『示例：变量替换』

使用 XML 和 MQ Explorer 对资源监视器定义进行变量替换的示例。

相关参考

第 401 页的『当变量替换导致多个文件传输到单个文件名时该怎么做』

对于 Managed File Transfer，如果要监视某个目录并将多个文件从源位置传输到目标位置，并且要使用 \${FileName} 变量替换，那么必须测试变量替换的结果。需要对结果进行测试的原因在于：使用变量替换可能导致调用意外的文件传输命令组合。

示例：变量替换

使用 XML 和 MQ Explorer 对资源监视器定义进行变量替换的示例。

用于演示变量替换工作方式的示例

假设匹配的触发器文件的文件路径为 c:\MONITOR\REPORTS\Paris\Report2009.doc，那么将按下表所示方式来替换变量。

表 21: 变量替换方式	
变量规范	变量替换后
\${FilePath}	c:\MONITOR\REPORTS\Paris\Report2009.doc
\${FilePath{token=1}{separator=.}}	c:\MONITOR\REPORTS\Paris\Report2009
\${FilePath{token=2}{separator=.}}	doc

表 21: 变量替换方式 (继续)	
变量规范	变量替换后
\${FilePath{token=3}}	REPORTS

也可以指定负标记索引，以选择相对于最后一个变量字符的标记，如下表所示。该表中的示例均使用相同的变量值 c:\MONITOR\REPORTS\Paris\Report2009.doc。

表 22: 使用负标记索引的示例	
变量规范	变量替换后
\${FilePath}	c:\MONITOR\REPORTS\Paris\Report2009.doc
\${FilePath{token=-2}{separator=.}}	c:\MONITOR\REPORTS\Paris\Report2009
\${FilePath{token=-2}{separator=\}}	巴黎
\${FilePath{token=-4}}	MONITOR

用于替换的变量只能用于正触发条件。只有 match 和 fileSize 触发条件会使得变量被替换。如果使用 noMatch 条件，并且任务定义中存在替换变量名称，那么将不调用任务，并且监视器将提供返回码 103 和错误消息 BFGDM0060E。

使用 XML 的示例

以下示例任务定义 XML 将监视器代理名称用作传输的源代理 (Paris)，将文件路径中的倒数第二个目录名用作传输的目标代理名称 (Report2009)，并将所传输的文件重命名为扩展名为 .rpt 的触发器文件名的根。

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="${AgentName}" QMgr="QM1" />
    <destinationAgent agent="${FilePath{token=-2}}" QMgr="QMD" />
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:/incoming/reports/summary/report.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>reports/${FileName{token=1}{separator=.}}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

这导致任务 XML 转换为：

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT1" QMgr="QM1" />
    <destinationAgent agent="Paris" QMgr="QMD" />
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:/incoming/reports/summary/report.doc</file>
        </source>
        <destination type="file" exist="overwrite">
```

```
<file>/reports/Report2009.rpt</file>
</destination>
</item>
</transferSet>
</managedTransfer>
</request>
```

<destinationAgent> 元素的 agent 属性中的变量 `${FilePath}{token=-2}` 将替换为值 Paris。将采用不区分大小写的方式处理该值，并且会将该值解释为对代理 PARIS 的引用。

使用 MQ Explorer 的示例

在通过 MQ Explorer 创建资源监视器并指定监视器属性和触发条件后，提供了用于将传输项添加到监视器的选项。以下示例演示如何在“添加传输项”面板中使用 `${FilePath}` 和 `${FileName}` 变量来定制由资源监视器匹配生成的传输。

示例 1

要在满足触发条件时仅仅将源文件传输到其他位置，可使用 `${FilePath}` 变量：

- 将源文件名设置为 `${FilePath}`。
- 从目标的类型下拉菜单中，选择目录。
- 将目标文件名设置为要将源文件传输到的位置，例如，C:\MFT\out\。

示例 2

要将源文件传输到其他位置并更改该文件的扩展名，可结合使用 `${FileName}` 变量和 `${FilePath}` 变量：

在以下示例中，假设源文件的文件路径为 C:\MONITOR\REPORTS\Paris\Report2009.doc：

- 将源文件名设置为 `${FilePath}`。
- 将目标文件名设置为要将源文件传输到的位置，后接 `${FileName}{token=1}{separator=.}`，再后接该文件的新扩展名。例如，C:\MFT\out\\${FileName}{token=1}{separator=.}.rpt，这等同于源文件名的 C:\MFT\out\Report2009.rpt。

示例 3

要使用源文件的文件路径的一部分来确定传输的目标，可结合使用 `${FilePath}` 变量与标记和分隔符规范。

在以下示例中，假设源文件的文件路径为 C:\MONITOR\REPORTS\Paris\Report2009.doc。

可以使用源文件路径的一部分来确定该文件的目标。如果使用文件路径示例

C:\MONITOR\REPORTS\Paris\Report2009.doc，并且要将文件传输到取决于源文件位置的文件夹（在此示例中为 Paris），那么可执行以下操作：

- 将源文件名设置为 `${FilePath}`。
- 将目标文件名设置为用于放置每个位置的文件夹的目标，然后附加文件路径的目标部分和文件名。例如，C:\MFT\out\\${FilePath}{token=-2}{separator=\}\\${FileName}，这等同于源文件名的 C:\MFT\out\Paris\Report2009.doc。

相关概念

[第 239 页的『使用变量替换定制 MFT 任务』](#)

在满足活动资源监视器的触发条件时，会调用已定义的任务。除了每次使用相同的目标代理或相同的目标文件名调用传输或命令任务之外，您还可以在运行时修改任务定义。可通过将变量名插入到任务定义 XML 中来实现此目的。在监视器确定已满足触发条件并且任务定义包含变量名时，将用变量值替换变量名，然后调用任务。

相关参考

[第 401 页的『当变量替换导致多个文件传输到单个文件名时该怎么做』](#)

对于 Managed File Transfer，如果要监视某个目录并将多个文件从源位置传输到目标位置，并且要使用 `${FileName}` 变量替换，那么必须测试变量替换的结果。需要对结果进行测试的原因在于：使用变量替换可能导致调用意外的文件传输命令组合。

监视队列和使用变量替换

您可以使用 **fteCreateMonitor** 命令监视队列，并将消息从受监视的队列传输到文件。从受监视队列读取的第一条消息中任何 IBM MQ 消息属性的值均可在任务 XML 定义中被替换，并用于定义传输行为。

关于此任务

在本示例中，源代理名为 AGENT_VENUS，与 QM_VENUS 连接。AGENT_VENUS 监视的队列名为 START_QUEUE，位于 QM_VENUS 上。代理每隔 30 分钟轮询一次队列。

将一组完整的消息写入到队列中时，监视器任务会将这组消息发送到某个目标代理中的文件，这些目标代理均连接到队列管理器 QM_MARS。这组消息传输到的文件的名称由该组中第一条消息的 IBM MQ 消息属性 `usr.fileName` 来定义。这组消息发送到的代理的名称由该组中第一条消息的 IBM MQ 消息属性 `usr.toAgent` 来定义。如果 `usr.toAgent` 头未设置，那么用于该目标代理的缺省值为 AGENT_MAGENTA。

当指定 `useGroups="true"` 时，如果未同时指定 `groupId="${GROUPID}"`，那么仅传输队列中的第一条消息。例如，如果使用变量替换来生成 `fileName`，那么 `a.txt` 的内容有可能不正确。这是因为 `fileName` 由监视器生成，但传输实际获取的一条消息并非应生成名为 `fileName` 的文件的消息。

过程

1. 创建任务 XML，以定义监视器被触发时应执行的任务。

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="${toAgent}" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/${fileName}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

替换为 IBM MQ 消息头值的变量以**粗体**突出显示。该任务 XML 保存到文件 `/home/USER1/task.xml`

2. 创建一个资源监视器以监视队列 START_QUEUE。

提交以下命令：

```
fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA
```

3. 用户或程序将一组消息写入队列 START_QUEUE 中。

该组中的第一条消息设置了以下 IBM MQ 消息属性：

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. 写入整组消息时，会触发监视器。代理将在任务 XML 中替换 IBM MQ 消息属性。
这导致任务 XML 转换为：

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/larmer.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

结果

这样会执行由任务 XML 定义的传输。由 AGENT_VENUS 从 START_QUEUE 读取的整组消息会写入到名为 /reports/larmer.rpt 的文件中，该文件位于运行 AGENT_VIOLET 的系统上。

下一步做什么

将每条消息传输到一个单独的文件

如果要监视某个队列，并将每条消息传输到一个单独的文件，可以使用与本主题中上述方法类似的方法。

1. 如前所述创建监视器，在 **fteCreateMonitor** 命令上指定 **-tr completeGroups** 参数。
2. 在任务 XML 中指定以下内容：

```
<queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
```

但是，在将消息放置到源队列上时，请不要将其放置在 IBM MQ 组中。向每条消息添加 IBM MQ 消息属性。例如，指定 **usr.filename** 属性，其中每条消息均有唯一的文件名值。这样实际上会使得 IBM MQ Managed File Transfer 代理将源队列上的每条消息视为单独的组。

相关概念

[第 229 页的『资源监控』](#)

您可以监视 IBM MQ Managed File Transfer 资源；例如，队列或目录。当满足针对该资源的条件时，资源监视器将启动任务，如文件传输。您可以使用 **fteCreateMonitor** 命令或 IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

[第 263 页的『将数据从消息传输到文件』](#)

通过使用 IBM MQ Managed File Transfer 的消息到文件传输功能，您可以将 IBM MQ 队列上的一条或多条消息的数据传输到文件、数据集或用户文件空间。如果您拥有一个可创建或处理 IBM MQ 消息的应用程序，那么可以使用 IBM MQ Managed File Transfer 的消息到文件传输功能，将这些消息传输到 IBM MQ Managed File Transfer 网络中任何系统上的文件。

[第 239 页的『使用变量替换定制 MFT 任务』](#)

在满足活动资源监视器的触发条件时，会调用已定义的任务。除了每次使用相同的目标代理或相同的目标文件名调用传输或命令任务之外，您还可以在运行时修改任务定义。可通过将变量名插入到任务定义 XML 中来实现此目的。在监视器确定已满足触发条件并且任务定义包含变量名时，将用变量值替换变量名，然后调用任务。

相关任务

[第 233 页的『配置监视器任务以启动命令和脚本』](#)

资源监视器的作用不只限于将执行文件传输作为其相关的任务。您还可以配置监视器，以从监视代理调用其他命令，包括可执行程序、Ant 脚本或 JCL 作业。要调用命令，请编辑监视任务定义 XML，以将一个或多个命令元素包含在对应的命令调用参数（例如，自变量和属性）中。

[第 238 页的『示例：配置资源监视器以监视队列』](#)

您可以通过使用带 **-mq** 参数的 **fteCreateMonitor** 命令，将 IBM MQ 队列指定为要由资源监视器监视的资源。

相关参考

[第 478 页的『fteCreateMonitor（新建资源监视器）』](#)

fteCreateMonitor 命令从命令行创建并启动新的资源监视器。您可以使用 WebSphere MQ Managed File Transfer 监视资源（例如，目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

[第 741 页的『从源队列上的消息中读取的 IBM MQ 消息属性』](#)

在从“消息到文件”传输过程中从源队列读取消息的代理将从该消息中读取 IBM MQ 消息属性。这些属性的值可用于确定传输行为。

[第 389 页的『如果队列资源监视器启动的传输所创建的目标文件包含错误数据，那么该执行哪些操作？』](#)

您可以创建资源监视器来监视队列，并将队列上的消息或一组消息传输到文件。可以通过使用消息上的 MQMD 消息述符描或组中的第一个消息来指定文件名。如果消息到文件的传输失败，并且消息或组留在队列上，那么下次触发监视器时，可能导致创建的文件中包含错误数据。

消息到文件传输的监视器重试行为

如果由资源监视器触发的消息到文件传输失败，并且触发该监视器的消息组留在队列中，那么将在后续轮询时间间隔中重新提交本个传输。重新提交传输的次数受监视代理的 **monitorGroupRetryLimit** 属性限制。

触发的消息到文件传输的次数由组中第一条消息的 MQMD 回退计数确定。

每次触发一次新的消息到文件传输时，都将为传输任务生成一个新的传输标识。

如果代理重新启动，监视器将再次触发一个传输，即使触发传输的次数已超出 **monitorGroupRetryLimit** 的值，也是如此。如果此传输会导致触发传输的次数超出 **monitorGroupRetryLimit** 的值，那么代理会将一个错误写入其事件日志。

单条消息被视为单个组，当消息保留在队列上，且触发传输的次数小于 **monitorGroupRetryLimit** 的值时，将在每次轮询时间间隔再次触发传输。

设置 monitorGroupRetryLimit 属性

monitorGroupRetryLimit 属性的值是消息组仍在于队列上时，监视器再次触发消息到文件传输的最大次数。该属性的缺省值为 10。该属性的值可设置为任何正整数值或 -1。如果对该属性指定值 -1，那么监视器会不限制次数地再次触发传输，直至触发条件不满足。

要在监视代理上设置 **monitorGroupRetryLimit** 属性，请执行以下步骤：

1. 使用 **fteStopAgent** 命令停止监视代理。
2. 编辑监视代理 `agent.properties` 文件以包含行
`monitorGroupRetryLimit=number_of_retries`。`agent.properties` 文件位于目录
`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/monitoring_agent_name`
中。
3. 使用 **fteStartAgent** 命令启动监视代理。

相关任务

[第 238 页的『示例：配置资源监视器以监视队列』](#)

您可以通过使用带 **-mq** 参数的 **fteCreateMonitor** 命令，将 IBM MQ 队列指定为要由资源监视器监视的资源。

相关参考

[第 591 页的『agent.properties 文件』](#)

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

使用触发文件的内容

您可以在资源监视器中使用触发器文件的内容，定义一组要在单个传输请求中传输的文件。每次检测到匹配的触发器文件，都会针对源文件路径和（可选）目标文件路径对其内容进行解析。然后，这些文件路径用于在您指定的任务传输 XML 文件（作为单个传输请求提交给代理）中定义文件项。资源监视器的定义将决定是否启用触发器内容。

您可以通过指定 **-tc**（触发器内容）参数，在创建监视器时启用文件内容触发。此 **-tc** 参数仅适用于文件触发器选项 `match` 和 `noSizeChange`。有关创建监视器的更多信息，请参阅第 478 页的『`fteCreateMonitor`（新建资源监视器）』。

每个触发器文件的格式是每个文本行上单个要传输的文件路径。行的缺省格式是单个源文件路径或逗号分隔的源和目标文件路径。将空格字符作为文件路径的一部分进行处理。

在对触发器文件进行解析后，将生成文件路径列表，并且会将此列表应用于您指定的传输任务 XML。对于所有监视器，传输任务 XML 的格式是 **fteCreateTransfer** 命令生成的完整传输任务 XML，定义了单个项目或文件。单个项目必须使用替换变量 `${contentSource}` 和（可选）`${contentDestination}`，作为源和目标文件路径的替换项。监视器将扩展传输任务 XML，以包含针对触发器文件中每行（文件路径）的文件项。

您不能通过 **-bs** 参数使用文件内容触发，因为 **-tc** 参数意味着每个触发器文件对应一个传输请求。

示例

以下示例定义要对以 `trig` 结束的文件触发的监视器，并读取此文件中的文件路径。

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
```

fteCreateTransfer 命令将为源文件路径为 `${sourceContent}` 的单个文件创建名为 `task.xml` 的文件。例如：

```
<item checksumMethod="MD5" mode="binary">
  <source disposition="leave" recursive="false">
    <file>${contentSource}</file>
  </source>
</item>
```

fteCreateMonitor 命令将扫描 `/home/trigdir` 目录中以 `trig` 结束的文件，并使用内容为该触发器文件中的所有路径创建基于 `task.xml` 的单个传输请求。触发器文件的格式必须是每行上一个文件路径（仅限源），无逗号分隔符。例如：

```
/home/file/first.txt
/home/file/second.txt
/home/different/third.txt
:
```

所有文件都传递到 `/file/destdir` 目录，且保留其文件名，但不保留文件路径，即 `/home/file/first.txt` 将传递到 `/file/destdir/first.txt`。

Alternatively, if you change the `-dd /file/destdir` parameter in the **fteCreateTransfer** command to `-df ${contentDestination}` and the format of the content of a trigger file to `<source file path>,<destination file path>`, you can define different destination paths for the same destination agent. 例如：

```
/home/file/first.txt,/home/other/sixth.txt
```

这样目标位置将变为 `/home/other/sixth.txt`。

可以标记解析替换变量。例如，您可以使用 `${contentDestination{token=-1}}` 从提供的路径中分离出文件名部分。因此，如果 **fteCreateTransfer** 目标定义为 `-df /file/destdir/${contentDestinstion{token=-1}}`，那么 `/home/file/first.txt` 的新目标为 `/file/destdir/sixth.txt`。

高级选项

您可以使用 **-tcr** *regex* 参数更改触发器文件内容的缺省行格式。提供与所需行格式匹配并提供一个或两个捕获组的正则表达式。第一个捕获组是源，而第二个可选捕获组是目标。例如：

- 用连字符分隔源和目标路径：

```
((?:[^\-])+)-((?:[^\-])+)
```

在本示例中，在三个位置定义了分隔符，连字符 (-) 的所有三个实例都可以更改为任意字符。确保对任何特殊字符转义。

- 用逗号分隔源路径和目标路径，具有尾部空格。将忽略数字符号 (#) 指示的注释。

```
((?:[^\,])+),((?:[^\,])+)*(?:#.*)+
```

文件路径不能包含数字符号 (#)。通常，条目如下所示：`/home/source/from.txt,/home/destination/to.txt # some comment`。

如果您使用 **-tcr** 参数，请确保正则表达式设计正确并经过测试，以便表达式可以检测错误并正确解析触发器文件。

您可以使用 **-tcc destSrc** 参数逆转捕获顺序。如果指定此参数，那么第一个捕获组是目标文件路径，而第二个组是源文件路径。

如何处理错误

空触发器文件

如果触发器文件为空，那么结果是无文件传输。即，监视器创建传输请求，但不指定任何文件项。

触发器文件存在错误

如果触发器文件中的任何条目无法按照预期格式解析，那么不会生成传输请求。将发布监视器错误日志，同时会将错误记录在事件日志中。会将触发器文件标记为已处理，并且监视器不会重试处理此文件，直至已更新此文件。

不匹配的传输任务 XML

传输任务 XML 必须与触发器文件匹配，即，如果传输任务 XML 具有 `${sourceContent}` 和 `${destinationContent}`，那么该监视器的所有触发器文件必须具有源和目标文件路径，反之亦然。在第一种情况下，如果触发器文件仅提供源文件路径，那么监视器将报告 `${destinationContent}` 替换失败。

示例

以下示例是触发器文件内容仅具有源文件路径的基本内容触发器：

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${sourceContent}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
```

-tcr 参数定义由空格字符分隔的任何字符序列的两个捕获组。**-tcc destSrc** 参数和选项指示要将捕获组先作为目标然后作为源来处理。

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -df ${destinationContent} ${sourceContent}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
-tcr "((?:[^\ ])+) ((?:[^\ ])+)" -tcc destSrc
```

相关概念

第 231 页的『资源监视概念』

这里概述了 IBM MQ Managed File Transfer 资源监视功能的主要概念。

第 239 页的『使用变量替换定制 MFT 任务』

在满足活动资源监视器的触发条件时，会调用已定义的任务。除了每次使用相同的目标代理或相同的目标文件名调用传输或命令任务之外，您还可以在运行时修改任务定义。可通过将变量名插入到任务定义 XML 中来实现此目的。在监视器确定已满足触发条件并且任务定义包含变量名时，将用变量值替换变量名，然后调用任务。

相关任务

第 243 页的『监视队列和使用变量替换』

您可以使用 **fteCreateMonitor** 命令监视队列，并将消息从受监视的队列传输到文件。从受监视队列读取的第一条消息中任何 IBM MQ 消息属性的值均可在任务 XML 定义中被替换，并用于定义传输行为。

相关参考

第 478 页的『fteCreateMonitor（新建资源监视器）』

fteCreateMonitor 命令从命令行创建并启动新的资源监视器。您可以使用 WebSphere MQ Managed File Transfer 监视资源（例如，目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

第 497 页的『fteCreateTransfer（创建新的文件传输）』

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

使用传输模板

您可以使用文件传输模板来存储重复或复杂传输的公共文件传输设置。使用 **fteCreateTemplate** 命令从命令行创建传输模板，或者使用 IBM MQ Explorer 通过“**为受管文件传输创建新模板**”向导创建传输模板，或者通过选中 **将传输设置另存为模板** 复选框在创建文件传输时保存模板。**传输模板**窗口显示在您的 IBM MQ Managed File Transfer 网络中已创建的所有传输模板。

关于此任务

要从命令行创建传输模板，请使用 **fteCreateTemplate** 命令。然后，在您想要提交从命令行中创建的传输模板时，单击 IBM MQ Explorer 中的**提交**。

要在 IBM MQ Explorer 中查看传输模板，请使用以下步骤：

过程

1. 展开“导航器”视图中的**受管文件传输**。在“内容”视图中将显示**受管文件传输中心**。
2. 您的所有协调队列管理器都将在“导航器”视图中列出。展开您用于已调度传输的协调队列管理器的名称。如果要更改所连接的协调队列管理器，请在“导航器”视图中右键单击要使用的协调队列管理器的名称，然后单击**连接**。
3. 单击**传输模板**。这将在“内容”视图中显示“**传输模板**”窗口。
4. “**传输模板**”窗口列出有关文件传输的以下详细信息：
 - a) **名称** 文件传输模板的名称。
 - b) **源** 用于从源系统传输文件的代理的名称。
 - c) **源文件** 要在主机系统上传输的文件名称。
展开传输模板信息以查看该字段。
 - d) **目标** 目标系统上用于接收文件的代理的名称。
 - e) **目标文件** 将文件传输至目标系统后的文件的名称。
展开传输模板信息以查看该字段。
 - f) **调度的开始时间（选中的时区）** 调度文件传输在管理员使用的时区开始的日期和时间。要更改显示的时区，请单击**窗口 > 首选项 > IBM MQ Explorer > 受管文件传输**，然后从**时区：**列表中选择一个备选时区。单击**确定**。

- g) **触发器事件** 触发文件传输启动的事件的类型。类型可以为以下值之一: `exists`、`does not exist` 或 `exceeds`。

结果

要刷新在**传输模板**窗口中显示的内容, 请单击“内容”视图工具栏上的“刷新”按钮 。

要提交传输模板并启动该模板中定义的传输, 请用鼠标右键单击该模板名, 然后单击**提交**。

要更改传输模板, 请右键单击模板名称, 然后单击**编辑**。原始模板中包含的所有文件将作为传输组的一部分列出, 即使它们未作为组的一部分包含在原始模板中。如果想要从模板中移除文件, 必须从组中选择文件规范, 然后单击**移除所选项**。如果想要将新文件规范添加到模板, 请使用模板面板中的字段, 然后单击**添加到组**按钮。进行编辑时, 系统将提示您对所编辑的模板指定新名称。

要从传输模板创建文件传输, 请右键单击模板名称, 然后单击**作为新传输进行编辑**。

要创建传输模板的副本, 请用鼠标右键单击模板名, 然后单击**复制**。将使用与原始模板相同的名称自动保存重复的传输模板, 其中追加“(copy)”。

要删除传输模板, 请用鼠标右键单击模板名, 然后单击**删除**。

相关任务

第 249 页的『使用 IBM MQ Explorer 创建文件传输模板』

您可以从 IBM MQ Explorer 或从命令行创建文件传输模板。然后, 您可以使用该模板来利用模板详细信息创建新的文件传输, 或者提交该模板以启动文件传输。

相关参考

第 484 页的『`fteCreateTemplate` (新建文件传输模板)』

`fteCreateTemplate` 命令创建文件传输模板, 您可以保留该模板以供将来使用。唯一的必需参数是 **`-tn`** (*template_name*) 参数。其他所有参数都是可选的, 但如果指定源文件, 还必须提供目标文件。类似地, 如果指定目标文件, 还必须指定源文件规范。

第 537 页的『`fteListTemplates` (列出 IBM MQ Managed File Transfer 模板)』

`fteListTemplates` 命令可用于列出协调队列管理器上的可用 IBM MQ Managed File Transfer 传输模板。

第 529 页的『`fteDeleteTemplates` (删除 IBM MQ Managed File Transfer 模板)』

使用 **`fteDeleteTemplates`** 命令可从协调队列管理器中删除现有的 IBM MQ Managed File Transfer 模板。

使用 IBM MQ Explorer 创建文件传输模板

您可以从 IBM MQ Explorer 或从命令行创建文件传输模板。然后, 您可以使用该模板来利用模板详细信息创建新的文件传输, 或者提交该模板以启动文件传输。

关于此任务

要从命令行创建文件传输模板, 请使用 `fteCreateTemplate` 命令。

要使用 IBM MQ Explorer 中的**为受管文件传输新建模板**向导来创建文件传输模板, 请使用以下步骤:

过程

1. 在“导航器”视图中, 单击**受管文件传输**。在“内容”视图将显示**受管文件传输中心**。
2. 在“导航器”视图将显示所有协调队列管理器。展开您用于已调度传输的协调队列管理器的名称。如果要更改所连接的协调队列管理器, 请在“导航器”视图中右键单击要使用的协调队列管理器的名称, 然后单击**连接**。
3. 通过右键单击**传输模板**, 然后单击**新建模板**, 从而启动**为受管文件传输新建模板**向导。
4. 遵循向导面板上的指示信息。为每个面板提供了上下文相关帮助。要在 Windows 上访问上下文相关帮助, 请按 F1。在 Linux 上, 按 Ctrl+F1 或 Shift+F1。

如果创建了包含所有必需传输详细信息的模板，请确保您选中**传输摘要**页面上的**将传输设置另存为模板**复选框（如果尚未选中此复选框）。此外，请在“名称”字段中输入该模板的名称。如果创建了尚未包含所有必需传输详细信息的模板，那么将自动为您选中**将传输设置另存为模板**复选框。

相关任务

第 248 页的『使用传输模板』

您可以使用文件传输模板来存储重复或复杂传输的公共文件传输设置。使用 **fteCreateTemplate** 命令从命令行创建传输模板，或者使用 IBM MQ Explorer 通过“**为受管文件传输创建新模板**”向导创建传输模板，或者通过选中 **将传输设置另存为模板** 复选框在创建文件传输时保存模板。**传输模板**窗口显示在您的 IBM MQ Managed File Transfer 网络中已创建的所有传输模板。

相关参考

第 484 页的『fteCreateTemplate（新建文件传输模板）』

fteCreateTemplate 命令创建文件传输模板，您可以保留该模板以供将来使用。唯一的必需参数是 **-tn** (*template_name*) 参数。其他所有参数都是可选的，但如果指定源文件，还必须提供目标文件。类似地，如果指定目标文件，还必须指定源文件规范。

第 537 页的『fteListTemplates（列出 IBM MQ Managed File Transfer 模板）』

fteListTemplates 命令可用于列出协调队列管理器上的可用 IBM MQ Managed File Transfer 传输模板。

第 529 页的『fteDeleteTemplates（删除 IBM MQ Managed File Transfer 模板）』

使用 **fteDeleteTemplates** 命令可从协调队列管理器中删除现有的 IBM MQ Managed File Transfer 模板。

将数据从文件传输至消息

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

要执行文件到消息和消息到文件的传输，传输的源和目标代理都必须为 IBM WebSphere MQ V7.5 或更高版本，或者 WebSphere MQ File Transfer Edition V7.0.3 或更高版本。有关“消息到文件”传输的信息，请参阅第 263 页的『将数据从消息传输到文件』。

文件到消息传输的目标代理不能为协议网桥代理或 Connect:Direct 网桥代理。

您可以将文件数据传输到 IBM MQ 消息数据。IBM MQ 消息可由应用程序读取和使用。支持以下类型的文件到消息传输：

- 从单个文件到单条消息。此消息未设置 IBM MQ 组标识。
- 从单个文件到多条消息，通过将文件分割为给定长度的消息。这些消息均具有相同的 IBM MQ 组标识。
- 通过在 Java 正则表达式定界符处分割文本文件，从单个文件到多条消息。这些消息均具有相同的 IBM MQ 组标识。
- 从单个文件到多条消息，通过以十六进制定界符来分割二进制文件。这些消息均具有相同的 IBM MQ 组标识。

如果要使用字节序列作为定界符来分割二进制文件，请使用 **fteCreateTransfer** 命令的 **-sqdb** 参数。有关更多信息，请参阅 [-sqdb 参数](#)。

缺省情况下，文件到消息传输创建的消息是持久性的。这些消息可设置为非持久性或者由目标队列定义持久性值。

如果您指定将文件分割为多条消息，那么从该文件创建的所有消息都具有相同的 IBM MQ 组标识。如果您未指定将文件分割为多条消息，那么只会从该文件创建一条消息，并且该消息未设置 IBM MQ 组标识。

如果要将文件传输到大型消息或者众多小型消息，那么可能需要更改某些 IBM MQ 或 IBM MQ Managed File Transfer 属性。有关信息，请参阅第 394 页的『用于设置与消息大小关联的 IBM MQ 属性和 IBM MQ Managed File Transfer 属性的指南』。

注：如果目标队列是集群队列或集群队列的别名，并且如果未将代理属性 `enableClusterQueueInputOutput` 设置为 `true`，那么在将文件传输到队列中时，您会收到一条错误消息。有关更多信息，请参阅第 390 页的『如果目标队列是集群队列或集群队列的别名，那么该执行什么操作？』

相关任务

第 251 页的『配置代理以执行文件到消息 (file-to-message) 传输』

缺省情况下，代理无法执行文件到消息或消息到文件传输。要启用该功能，必须将代理属性 `enableQueueInputOutput` 设置为 `true`。为了能够写入 IBM MQ 集群队列，还必须将代理属性 `enableClusterQueueInputOutput` 设置为 `true`。

第 253 页的『示例：将单个文件传输到单条消息』

您可通过将 **-dq** 参数与 **fteCreateTransfer** 命令一起使用来指定队列作为文件传输的目标。源文件必须小于目标队列上设置的最大消息长度。目标队列所在的队列管理器不必是目标代理连接到的队列管理器，但是这两个队列管理器必须能够通信。

第 254 页的『示例：按长度将单个文件分割为多条消息』

您可通过使用 **fteCreateTransfer** 命令的 **-qs** 参数将文件分割为多个 IBM MQ 消息。文件被分割成固定长度的段，每个段会写入单条消息。

第 256 页的『示例：使用正则表达式定界符分割文本文件，并在消息中包含该定界符』

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息，并在生成的消息中包含正则表达式匹配项。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 和 **-qi** 参数。

第 255 页的『示例：使用正则表达式定界符将文本文件分割为多条消息』

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 参数。

第 258 页的『示例：在文件到消息传输上设置 IBM MQ 消息属性』

您可以在 **fteCreateTransfer** 命令上使用 **-qmp** 参数来指定是否在传输写入目标队列的第一条消息上设置 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

第 260 页的『示例：在文件到消息传输上设置用户定义的属性』

用户定义的元数据将设置为由传输写入到目标队列的第一条消息上的 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

第 221 页的『启动新的文件传输』

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

相关参考

第 262 页的『文件到消息传输失败』

如果代理开始将文件数据写入到目标队列后文件到消息传输失败，那么代理会向队列写入一条消息以向使用这些消息的应用程序指明发生失败。

第 739 页的『写入到目标队列的消息上设置的 IBM MQ 消息属性』

当从文件传输到消息时，IBM MQ Managed File Transfer 可在写入到目标队列的第一条消息上设置 IBM MQ 消息属性。如果从“文件到消息”传输失败，那么会设置其他 IBM MQ 消息属性。

第 394 页的『用于设置与消息大小关联的 IBM MQ 属性和 IBM MQ Managed File Transfer 属性的指南』

在读取或写入各种大小的消息时，您可以更改 IBM MQ 属性和 IBM MQ Managed File Transfer 属性以影响 IBM MQ Managed File Transfer 的行为。

配置代理以执行文件到消息 (file-to-message) 传输

缺省情况下，代理无法执行文件到消息或消息到文件传输。要启用该功能，必须将代理属性 `enableQueueInputOutput` 设置为 `true`。为了能够写入 IBM MQ 集群队列，还必须将代理属性 `enableClusterQueueInputOutput` 设置为 `true`。

关于此任务

如果尝试执行文件到消息传输至 `enableQueueInputOutput` 属性未设置为 `true` 的目标代理，那么传输会失败。发布至协调队列管理器的传输日志消息包含以下消息：

```
BFGI00197E: An attempt to write to a queue was rejected by the destination agent. The agent must have enableQueueInputOutput=true set in the agent.properties file to support transferring to a queue.
```

要启用代理在队列上写入和读取，请执行以下步骤：

过程

1. 使用 **fteStopAgent** 命令停止目标代理。
2. 编辑 `agent.properties` 文件以包含行 `enableQueueInputOutput=true`。
`agent.properties` 文件位于目录 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name` 中。
3. 可选：编辑 `agent.properties` 文件以包含行 `enableClusterQueueInputOutput=true`。
`agent.properties` 文件位于目录 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name` 中。
4. 使用 **fteStartAgent** 命令启动目标代理。

相关概念

第 250 页的『将数据从文件传输至消息』

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

相关任务

第 253 页的『示例：将单个文件传输到单条消息』

您可通过将 **-dq** 参数与 **fteCreateTransfer** 命令一起使用来指定队列作为文件传输的目标。源文件必须小于目标队列上设置的最大消息长度。目标队列所在的队列管理器不必是目标代理连接到的队列管理器，但是这两个队列管理器必须能够通信。

第 254 页的『示例：按长度将单个文件分割为多条消息』

您可通过使用 **fteCreateTransfer** 命令的 **-qs** 参数将文件分割为多个 IBM MQ 消息。文件被分割成固定长度的段，每个段会写入单条消息。

第 256 页的『示例：使用正则表达式定界符分割文本文件，并在消息中包含该定界符』

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息，并在生成的消息中包含正则表达式匹配项。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 和 **-qi** 参数。

第 255 页的『示例：使用正则表达式定界符将文本文件分割为多条消息』

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 参数。

第 258 页的『示例：在文件到消息传输上设置 IBM MQ 消息属性』

您可以在 **fteCreateTransfer** 命令上使用 **-qmp** 参数来指定是否在传输写入目标队列的第一条消息上设置 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

第 260 页的『示例：在文件到消息传输上设置用户定义的属性』

用户定义的元数据将设置为由传输写入到目标队列的第一条消息上的 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

相关参考

第 576 页的『fteStopAgent（停止 IBM MQ Managed File Transfer 代理）』

使用 **fteStopAgent** 命令以受控方式停止 IBM MQ Managed File Transfer 代理程序，或者使用 **-i** 参数在必要时立即停止代理程序。

第 573 页的『fteStartAgent（启动 IBM MQ Managed File Transfer 代理）』

fteStartAgent 命令从命令行启动 IBM MQ Managed File Transfer 代理程序。

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

第 262 页的『文件到消息传输失败』

如果代理开始将文件数据写入到目标队列后文件到消息传输失败，那么代理会向队列写入一条消息以向使用这些消息的应用程序指明发生失败。

示例：将单个文件传输到单条消息

您可通过将 **-dq** 参数与 **fteCreateTransfer** 命令一起使用来指定队列作为文件传输的目标。源文件必须小于目标队列上设置的最大消息长度。目标队列所在的队列管理器不必是目标代理连接到的队列管理器，但是这两个队列管理器必须能够通信。

关于此任务

源文件名为 `/tmp/single_record.txt`，位于源代理 `AGENT_NEPTUNE` 所在的系统上。源代理 `AGENT_NEPTUNE` 使用队列管理器 `QM_NEPTUNE`。目标代理是 `AGENT_VENUS`，此代理连接至队列管理器 `QM_VENUS`。目标队列 `RECEIVING_QUEUE` 位于队列管理器 `QM_MERCURY` 上。`QM_MERCURY` 位于队列管理器 `QM_VENUS` 所在的 IBM MQ 网络中，并且可由该队列管理器访问。

过程

输入以下命令：

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -dm QM_VENUS  
-dq RECEIVING_QUEUE@QM_MERCURY /tmp/single_record.txt
```

如果目标队列所在的队列管理器不同于目标代理所使用的队列管理器，那么必须按以下格式指定 **-dq** 参数的值：`queue_name@queue_manager_name`。如果未在该值中指定 `@queue_manager_name`，那么目标代理会假定目标队列位于目标代理队列管理器上。`enableClusterQueueInputOutput` 代理属性已设置为 `true` 的情况例外。在此情况下，目标代理将使用标准 IBM MQ 解决过程来确定放置队列的位置。

源代理 `AGENT_NEPTUNE` 会从文件 `/tmp/single_record.txt` 读取数据，并将此数据传输至目标代理 `AGENT_VENUS`。目标代理 `AGENT_VENUS` 会将数据发送至队列 `RECEIVING_QUEUE@QM_MERCURY` 上的持久消息。此消息未设置 IBM MQ 组标识。

相关概念

[第 250 页的『将数据从文件传输至消息』](#)

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

相关任务

[第 251 页的『配置代理以执行文件到消息 \(file-to-message\) 传输』](#)

缺省情况下，代理无法执行文件到消息或消息到文件传输。要启用该功能，必须将代理属性 `enableQueueInputOutput` 设置为 `true`。为了能够写入 IBM MQ 集群队列，还必须将代理属性 `enableClusterQueueInputOutput` 设置为 `true`。

[第 254 页的『示例：按长度将单个文件分割为多条消息』](#)

您可通过使用 **fteCreateTransfer** 命令的 **-qs** 参数将文件分割为多个 IBM MQ 消息。文件被分割成固定长度的段，每个段会写入单条消息。

[第 256 页的『示例：使用正则表达式定界符分割文本文件，并在消息中包含该定界符』](#)

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息，并在生成的消息中包含正则表达式匹配项。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 和 **-qi** 参数。

[第 255 页的『示例：使用正则表达式定界符将文本文件分割为多条消息』](#)

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 参数。

[第 258 页的『示例：在文件到消息传输上设置 IBM MQ 消息属性』](#)

您可以在 **fteCreateTransfer** 命令上使用 **-qmp** 参数来指定是否在传输写入目标队列的第一条消息上设置 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

[第 260 页的『示例：在文件到消息传输上设置用户定义的属性』](#)

用户定义的元数据将设置为由传输写入到目标队列的第一条消息上的 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

[第 221 页的『启动新的文件传输』](#)

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

相关参考

[第 262 页的『文件到消息传输失败』](#)

如果代理开始将文件数据写入到目标队列后文件到消息传输失败，那么代理会向队列写入一条消息以向使用这些消息的应用程序指明发生失败。

示例：按长度将单个文件分割为多条消息

您可通过使用 **fteCreateTransfer** 命令的 **-qs** 参数将文件分割为多个 IBM MQ 消息。文件被分割成固定长度的段，每个段会写入单条消息。

关于此任务

源文件名为 `/tmp/source.file`，大小为 36KB。源文件位于源代理 AGENT_NEPTUNE 所在的系统上。源代理 AGENT_NEPTUNE 连接至队列管理器 QM_NEPTUNE。目标代理是 AGENT_MERCURY，此代理连接至队列管理器 QM_MERCURY。目标队列 RECEIVING_QUEUE 同样位于队列管理器 QM_MERCURY 上。传输会将源文件分割成大小为 1KB 的段，并将每个段写入 RECEIVING_QUEUE 上的消息。

过程

输入以下命令：

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -qs 1K /tmp/source.file
```

源代理 AGENT_NEPTUNE 会从文件 `/tmp/source.file` 读取数据，并将此数据传输至目标代理 AGENT_MERCURY。目标代理 AGENT_MERCURY 会将数据写入队列 RECEIVING_QUEUE@QM_MERCURY 上的三十六条 1KB 的持久消息。这些消息都具有相同的 IBM MQ 组标识；组中的最后一条消息设置了 IBM MQ LAST_MSG_IN_GROUP 标志。

相关概念

[第 250 页的『将数据从文件传输至消息』](#)

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

相关任务

[第 251 页的『配置代理以执行文件到消息 \(file-to-message\) 传输』](#)

缺省情况下，代理无法执行文件到消息或消息到文件传输。要启用该功能，必须将代理属性 `enableQueueInputOutput` 设置为 `true`。为了能够写入 IBM MQ 集群队列，还必须将代理属性 `enableClusterQueueInputOutput` 设置为 `true`。

[第 253 页的『示例：将单个文件传输到单条消息』](#)

您可通过将 **-dq** 参数与 **fteCreateTransfer** 命令一起使用来指定队列作为文件传输的目标。源文件必须小于目标队列上设置的最大消息长度。目标队列所在的队列管理器不必是目标代理连接到的队列管理器，但是这两个队列管理器必须能够通信。

[第 256 页的『示例：使用正则表达式定界符分割文本文件，并在消息中包含该定界符』](#)

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息，并在生成的消息中包含正则表达式匹配项。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 和 **-qi** 参数。

[第 255 页的『示例：使用正则表达式定界符将文本文件分割为多条消息』](#)

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 参数。

[第 258 页的『示例：在文件到消息传输上设置 IBM MQ 消息属性』](#)

您可以在 **fteCreateTransfer** 命令上使用 **-qmp** 参数来指定是否在传输写入目标队列的第一条消息上设置 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

[第 260 页的『示例：在文件到消息传输上设置用户定义的属性』](#)

用户定义的元数据将设置为由传输写入到目标队列的第一条消息上的 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

[第 221 页的『启动新的文件传输』](#)

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

相关参考

[第 262 页的『文件到消息传输失败』](#)

如果代理开始将文件数据写入到目标队列后文件到消息传输失败，那么代理会向队列写入一条消息以向使用这些消息的应用程序指明发生失败。

示例：使用正则表达式定界符将文本文件分割为多条消息

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 参数。

关于此任务

该文件会分割为可变长度的段，每个段会写入单条消息。在文件中的文本匹配给定正则表达式的每个点分割文本文件。源文件名为 `/tmp/names.text`，包含以下内容：

```
Jenny Jones,John Smith,Jane Brown
```

指定文件分割位置的正则表达式为逗号字符 (,)。

源文件位于连接至队列管理器 QM_NEPTUNE 的源代理 AGENT_NEPTUNE 所在的系统上。目标队列 *RECEIVING_QUEUE* 位于队列管理器 QM_MERCURY 上。QM_MERCURY 同样是目标代理 AGENT_MERCURY 使用的队列管理器。传输会将源文件分割为段，并将每个段写入 *RECEIVING_QUEUE* 上的消息。

过程

输入以下命令：

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -t text -dqdp postfix -dqdt "," /tmp/names.text
```

源代理 AGENT_NEPTUNE 会从文件 `/tmp/names.text` 读取数据，并将此数据传输至目标代理 AGENT_MERCURY。目标代理 AGENT_MERCURY 会将数据写入队列 *RECEIVING_QUEUE* 上的三个持久消息。这些消息均带有相同的 WebSphere MQ 组标识，并且组中的最后一条消息设置了 WebSphere MQ *LAST_MSG_IN_GROUP* 标志。

消息中的数据如下。

- 第一条消息：

```
Jenny Jones
```

- 第二条消息：

```
John Smith
```

- 第三条消息：

```
Jane Brown
```

相关概念

[第 250 页的『将数据从文件传输至消息』](#)

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

相关任务

第 251 页的『配置代理以执行文件到消息 (file-to-message) 传输』

缺省情况下，代理无法执行文件到消息或消息到文件传输。要启用该功能，必须将代理属性 `enableQueueInputOutput` 设置为 `true`。为了能够写入 IBM MQ 集群队列，还必须将代理属性 `enableClusterQueueInputOutput` 设置为 `true`。

第 253 页的『示例：将单个文件传输到单条消息』

您可通过将 **-dq** 参数与 **fteCreateTransfer** 命令一起使用来指定队列作为文件传输的目标。源文件必须小于目标队列上设置的最大消息长度。目标队列所在的队列管理器不必是目标代理连接到的队列管理器，但是这两个队列管理器必须能够通信。

第 254 页的『示例：按长度将单个文件分割为多条消息』

您可通过使用 **fteCreateTransfer** 命令的 **-qs** 参数将文件分割为多个 IBM MQ 消息。文件被分割成固定长度的段，每个段会写入单条消息。

第 256 页的『示例：使用正则表达式定界符分割文本文件，并在消息中包含该定界符』

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息，并在生成的消息中包含正则表达式匹配项。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 和 **-qi** 参数。

第 258 页的『示例：在文件到消息传输上设置 IBM MQ 消息属性』

您可以在 **fteCreateTransfer** 命令上使用 **-qmp** 参数来指定是否在传输写入目标队列的第一条消息上设置 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

第 260 页的『示例：在文件到消息传输上设置用户定义的属性』

用户定义的元数据将设置为由传输写入到目标队列的第一条消息上的 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

第 221 页的『启动新的文件传输』

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

相关参考

第 262 页的『文件到消息传输失败』

如果代理开始将文件数据写入到目标队列后文件到消息传输失败，那么代理会向队列写入一条消息以向使用这些消息的应用程序指明发生失败。

第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 Connect:Direct 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 `java.util.regex` API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

示例：使用正则表达式定界符分割文本文件，并在消息中包含该定界符

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息，并在生成的消息中包含正则表达式匹配项。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 和 **-qi** 参数。

关于此任务

将单个文本文件传输至队列上的多条消息。该文件会分割为可变长度的段，每个段会写入单条消息。在文件中的文本匹配给定正则表达式的每个点分割文本文件。源文件名为 `/tmp/customers.text`，包含以下内容：

```
Customer name: John Smith
Customer contact details: john@example.net
Customer number: 314

Customer name: Jane Brown
Customer contact details: jane@example.com
Customer number: 42
```

```
Customer name: James Jones
Customer contact details: jjones@example.net
Customer number: 26
```

指定文件分割位置的正则表达式为 `Customer\snumber:\s\d+`，该正则表达式匹配文本 “Customer number:” 后接任何数量的数字。在命令行指定的正则表达式必须以双引号括起，以避免命令 `shell` 对正则表达式进行求值。正则表达式将作为 Java 正则表达式进行求值。有关更多信息，请参阅第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』。

缺省情况下，正则表达式可匹配的字符数设置为五个。该示例中使用的正则表达式匹配长度超过五个字符的字符串。要启用长度超过五个字符的匹配，请编辑代理属性文件，以包含属性 **maxDelimiterMatchLength**。

缺省情况下，匹配正则表达式的文本不包含在消息中。要在消息中包含匹配正则表达式的文本（如该示例所示），请使用 **-qi** 参数。源文件位于连接至队列管理器 QM_NEPTUNE 的源代理 AGENT_NEPTUNE 所在的系统上。目标队列 RECEIVING_QUEUE 位于队列管理器 QM_MERCURY 上。QM_MERCURY 同样是目标代理 AGENT_MERCURY 使用的队列管理器。传输会将源文件分割为段，并将每个段写入 RECEIVING_QUEUE 上的消息。

过程

1. 使用以下命令停止目标代理：

```
fteStopAgent AGENT_MERCURY
```

2. 将以下行添加到 AGENT_MERCURY 的代理属性文件：

```
maxDelimiterMatchLength=25
```

注：增加 **maxDelimiterMatchLength** 的值可能会降低性能。

3. 使用以下命令启动目标代理：

```
fteStartAgent AGENT_MERCURY
```

4. 输入以下命令：

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY
-dq RECEIVING_QUEUE
text -dqdt "Customer\snumber:\s\d+" -qi -dqdp postfix /tmp/customers.text
```

源代理 AGENT_NEPTUNE 会从文件 `/tmp/customers.text` 读取数据，并将此数据传输至目标代理 AGENT_MERCURY。目标代理 AGENT_MERCURY 会将数据写入队列 RECEIVING_QUEUE 上的三个持久消息。这些消息都具有相同的 IBM MQ 组标识；组中的最后一条消息设置了 IBM MQ LAST_MSG_IN_GROUP 标志。

消息中的数据如下。

- 第一条消息：

```
Customer name: John Smith
Customer contact details: john@example.net
Customer number: 314
```

- 第二条消息：

```
Customer name: Jane Brown
Customer contact details: jane@example.com
Customer number: 42
```

- 第三条消息：

```
Customer name: James Jones
Customer contact details: jjones@example.net
Customer number: 26
```

相关概念

第 250 页的『将数据从文件传输至消息』

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

相关任务

第 251 页的『配置代理以执行文件到消息 (file-to-message) 传输』

缺省情况下，代理无法执行文件到消息或消息到文件传输。要启用该功能，必须将代理属性 `enableQueueInputOutput` 设置为 `true`。为了能够写入 IBM MQ 集群队列，还必须将代理属性 `enableClusterQueueInputOutput` 设置为 `true`。

第 253 页的『示例：将单个文件传输到单条消息』

您可通过将 `-dq` 参数与 `fteCreateTransfer` 命令一起使用来指定队列作为文件传输的目标。源文件必须小于目标队列上设置的最大消息长度。目标队列所在的队列管理器不必是目标代理连接到的队列管理器，但是这两个队列管理器必须能够通信。

第 254 页的『示例：按长度将单个文件分割为多条消息』

您可通过使用 `fteCreateTransfer` 命令的 `-qs` 参数将文件分割为多个 IBM MQ 消息。文件被分割成固定长度的段，每个段会写入单条消息。

第 255 页的『示例：使用正则表达式定界符将文本文件分割为多条消息』

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息。要执行该操作，请使用 `fteCreateTransfer` 命令的 `-dqdt` 参数。

第 258 页的『示例：在文件到消息传输上设置 IBM MQ 消息属性』

您可以在 `fteCreateTransfer` 命令上使用 `-qmp` 参数来指定是否在传输写入目标队列的第一条消息上设置 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

第 260 页的『示例：在文件到消息传输上设置用户定义的属性』

用户定义的元数据将设置为由传输写入到目标队列的第一条消息上的 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

第 221 页的『启动新的文件传输』

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

相关参考

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

第 262 页的『文件到消息传输失败』

如果代理开始将文件数据写入到目标队列后文件到消息传输失败，那么代理会向队列写入一条消息以向使用这些消息的应用程序指明发生失败。

第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 Connect:Direct 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 `java.util.regex` API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

示例：在文件到消息传输上设置 IBM MQ 消息属性

您可以在 `fteCreateTransfer` 命令上使用 `-qmp` 参数来指定是否在传输写入目标队列的第一条消息上设置 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

关于此任务

将参数 `-qmp true` 包含在 **fteCreateTransfer** 命令中。在此示例中，提交该命令的用户的 MQMD 用户标识为 `larmer`。

过程

输入以下命令：

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM -qmp true  
-t text /tmp/source_file.txt
```

由目标代理 `AGENT_SATURN` 写入到队列管理器 `MyQM` 上的队列 `MY_QUEUE` 的第一条消息的 IBM MQ 消息属性设置为以下值：

```
usr.WMQFTETransferId=414cbaedefa234889d999a8ed09782395ea213ebbc9377cd  
usr.WMQFTETransferMode=text  
usr.WMQFTESourceAgent=AGENT_JUPITER  
usr.WMQFTEDestinationAgent=AGENT_SATURN  
usr.WMQFTEFileName=source_file.txt  
usr.WMQFTEFileSize=1024  
usr.WMQFTEFileLastModified=1273740879040  
usr.WMQFTEFileIndex=0  
usr.WMQFTEMqmdUser=larmer
```

相关概念

[第 250 页的『将数据从文件传输至消息』](#)

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

相关任务

[第 251 页的『配置代理以执行文件到消息 \(file-to-message\) 传输』](#)

缺省情况下，代理无法执行文件到消息或消息到文件传输。要启用该功能，必须将代理属性 `enableQueueInputOutput` 设置为 `true`。为了能够写入 IBM MQ 集群队列，还必须将代理属性 `enableClusterQueueInputOutput` 设置为 `true`。

[第 253 页的『示例：将单个文件传输到单条消息』](#)

您可通过将 **-dq** 参数与 **fteCreateTransfer** 命令一起使用来指定队列作为文件传输的目标。源文件必须小于目标队列上设置的最大消息长度。目标队列所在的队列管理器不必是目标代理连接到的队列管理器，但是这两个队列管理器必须能够通信。

[第 254 页的『示例：按长度将单个文件分割为多条消息』](#)

您可通过使用 **fteCreateTransfer** 命令的 **-qs** 参数将文件分割为多个 IBM MQ 消息。文件被分割成固定长度的段，每个段会写入单条消息。

[第 256 页的『示例：使用正则表达式定界符分割文本文件，并在消息中包含该定界符』](#)

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息，并在生成的消息中包含正则表达式匹配项。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 和 **-qi** 参数。

[第 255 页的『示例：使用正则表达式定界符将文本文件分割为多条消息』](#)

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 参数。

[第 260 页的『示例：在文件到消息传输上设置用户定义的属性』](#)

用户定义的元数据将设置为由传输写入到目标队列的第一条消息上的 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

[第 221 页的『启动新的文件传输』](#)

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

相关参考

[第 262 页的『文件到消息传输失败』](#)

如果代理开始将文件数据写入到目标队列后文件到消息传输失败，那么代理会向队列写入一条消息以向使用这些消息的应用程序指明发生失败。

[第 739 页的『写入到目标队列的消息上设置的 IBM MQ 消息属性』](#)

当从文件传输到消息时，IBM MQ Managed File Transfer 可在写入到目标队列的第一条消息上设置 IBM MQ 消息属性。如果从“文件到消息”传输失败，那么会设置其他 IBM MQ 消息属性。

示例：在文件到消息传输上设置用户定义的属性

用户定义的元数据将设置为由传输写入到目标队列的第一条消息上的 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

关于此任务

在 **fteCreateTransfer** 命令中包含参数 `-qmp true` 和 `-md account=123456`，以在 RFH2 头中将 `usr.account` 属性设置为 123456。

过程

输入以下命令：

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM  
-qmp true -md account=123456 /tmp/source_file.txt
```

除了 IBM MQ 标准消息属性集以外，用户定义的属性也在由目标代理 AGENT_SATURN 写入到队列管理器 MyQM 上队列 MY_QUEUE 的第一条消息的消息头中设置。头设置为以下值：

```
usr.account=123456
```

前缀 `usr` 会添加至用户定义的元数据的名称开头。

相关概念

[第 250 页的『将数据从文件传输至消息』](#)

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

相关任务

[第 251 页的『配置代理以执行文件到消息 \(file-to-message\) 传输』](#)

缺省情况下，代理无法执行文件到消息或消息到文件传输。要启用该功能，必须将代理属性 `enableQueueInputOutput` 设置为 `true`。为了能够写入 IBM MQ 集群队列，还必须将代理属性 `enableClusterQueueInputOutput` 设置为 `true`。

[第 253 页的『示例：将单个文件传输到单条消息』](#)

您可通过将 **-dq** 参数与 **fteCreateTransfer** 命令一起使用来指定队列作为文件传输的目标。源文件必须小于目标队列上设置的最大消息长度。目标队列所在的队列管理器不必是目标代理连接到的队列管理器，但是这两个队列管理器必须能够通信。

[第 254 页的『示例：按长度将单个文件分割为多条消息』](#)

您可通过使用 **fteCreateTransfer** 命令的 **-qs** 参数将文件分割为多个 IBM MQ 消息。文件被分割成固定长度的段，每个段会写入单条消息。

[第 256 页的『示例：使用正则表达式定界符分割文本文件，并在消息中包含该定界符』](#)

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息，并在生成的消息中包含正则表达式匹配项。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 和 **-qi** 参数。

[第 255 页的『示例：使用正则表达式定界符将文本文件分割为多条消息』](#)

通过在给定 Java 正则表达式的每个匹配项处拆分文件，将单个文本文件传输到多条消息。要执行该操作，请使用 **fteCreateTransfer** 命令的 **-dqdt** 参数。

[第 258 页的『示例：在文件到消息传输上设置 IBM MQ 消息属性』](#)

您可以在 **fteCreateTransfer** 命令上使用 **-qmp** 参数来指定是否在传输写入目标队列的第一条消息上设置 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

第 221 页的『启动新的文件传输』

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

相关参考

第 739 页的『写入到目标队列的消息上设置的 IBM MQ 消息属性』

当从文件传输到消息时，IBM MQ Managed File Transfer 可在写入到目标队列的第一条消息上设置 IBM MQ 消息属性。如果从“文件到消息”传输失败，那么会设置其他 IBM MQ 消息属性。

示例：为文件到消息传输添加用户定义的消息属性

如果要为 IBM MQ Managed File Transfer 用于消息到文件受管传输，您可以针对生成的消息包含用户定义的消息属性。

关于此任务

您可以使用以下任何方法定义定制消息属性：

- 对传输请求指定 **-md** 参数。有关更多信息，请参阅第 260 页的『示例：在文件到消息传输上设置用户定义的属性』。
- 使用 Ant 任务；您可以使用 `fte:filecopy` 或 `fte:filemove`。以下示例是 `fte:filecopy` 任务：

```
<project xmlns:fte="antlib:com.ibm.wmqfte.ant.taskdefs" default="complete">
<!-- Initialise the properties used in this script.-->

<target name="init" description="initialise task properties">
  <property name="src.file" value="/home/user/file1.bin"/>
  <property name="dst.queue" value="TEST.QUEUE@qm2"/>
  <fte:uuid property="job.name" length="8"
prefix="copyjob#"/>
</target>
<target name="step1" depends="init" description="transfer file">

<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="copy.result">

<fte:metadata>
<fte:entry name="fileName" value="${FileName}"/>
</fte:metadata>

<fte:filespec srcfilespec="${src.file}" dstqueue="${dst.queue}"
dstmsgprops="true"/>

</fte:filecopy>

</target>
</project>
```

- 使用资源监视器和变量替换。以下示例显示某个传输任务 XML：

```
<?xml version="1.0" encoding="UTF-8"?>
<monitor:monitor
xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="5.00"
xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./Monitor.xsd">
  <name>METADATA</name>
  <pollInterval units="minutes">5</pollInterval>
  <batch maxSize="5"/>
  <agent>AGENT1</agent>
  <resources>
    <directory recursionLevel="0">e:\temp</directory>
  </resources>
  <triggerMatch>
    <conditions>
      <allof>
        <condition>
          <fileMatch>
```

```

        <pattern>*.txt</pattern>
      </fileMatch>
    </condition>
  </allOf>
</conditions>
</triggerMatch>
<tasks>
  <task>
    <name/>
    <transfer>
      <request version="5.00"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
        <managedTransfer>
          <originator>
            <hostName>mqjason.raleigh.ibm.com.</hostName>
            <userID>administrator</userID>
          </originator>
          <sourceAgent QMgr="AGENTQM" agent="AGENT1"/>
          <destinationAgent QMgr="AGENTQM" agent="AGENT2"/>
          <transferSet priority="0">
            <metaDataSet>
              <metaData key="FileName">${FileName}</metaData>
            </metaDataSet>
            <item checksumMethod="MD5" mode="text">
              <source disposition="delete" recursive="false">
                <file>${FilePath}</file>
              </source>
              <destination type="queue">
                <queue persistent="true"
setMqProps="true">TEST.QUEUE@AGENTQM</queue>
              </destination>
            </item>
          </transferSet>
        </job>
        <name>Metadata_example</name>
      </job>
    </managedTransfer>
  </request>
</transfer>
</task>
</tasks>
<originator>
  <hostName>mqjason.raleigh.ibm.com.</hostName>
  <userID>administrator</userID>
</originator>
</monitor:monitor>

```

相关任务

第 258 页的『示例：在文件到消息传输上设置 IBM MQ 消息属性』

您可以在 **fteCreateTransfer** 命令上使用 **-qmp** 参数来指定是否在传输写入目标队列的第一条消息上设置 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

相关参考

第 955 页的『fte:filecopy』

fte:filecopy 任务在 IBM MQ Managed File Transfer 代理程序之间复制文件。该文件不会从源代理删除。

第 958 页的『fte:filemove』

fte:filemove 任务在 IBM MQ Managed File Transfer 代理程序之间移动文件。当文件成功从源代理传输到目标代理时，会从源代理删除该文件。

文件到消息传输失败

如果代理开始将文件数据写入到目标队列后文件到消息传输失败，那么代理会向队列写入一条消息以向使用这些消息的应用程序指明发生失败。

如果发生失败，表示写入到目标队列的消息：

- 是空消息
- 具有与代理先前写入到目标队列的消息相同的 IBM MQ 组标识
- 设置 IBM MQ LAST_MSG_IN_GROUP 标志

- 包含额外的 IBM MQ 消息属性（前提是已启用这些消息属性）。有关更多信息，请参阅主题 [第 739 页的『失败属性』](#)。

示例

通过运行以下命令来请求传输：

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq RECEIVING_QUEUE  
-qmp true -qs 1K /tmp/source1.txt
```

文件 source1.txt 为 48KB。传输会将此文件分割为 1KB 的消息，并将这些消息写入到目标队列 RECEIVING_QUEUE。

传输进行过程中，代理将 16 条消息写入 RECEIVING_QUEUE 后，在源代理处发生失败。

代理将空消息写入 RECEIVING_QUEUE。除标准消息属性集之外，空消息设置了以下消息属性：

```
usr.WMQFTRetCode = 40  
usr.WMQFTESupplement = BFGTR0036I: The transfer failed to complete successfully.
```

相关概念

[第 250 页的『将数据从文件传输至消息』](#)

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

相关任务

[第 251 页的『配置代理以执行文件到消息 \(file-to-message\) 传输』](#)

缺省情况下，代理无法执行文件到消息或消息到文件传输。要启用该功能，必须将代理属性 enableQueueInputOutput 设置为 true。为了能够写入 IBM MQ 集群队列，还必须将代理属性 enableClusterQueueInputOutput 设置为 true。

[第 221 页的『启动新的文件传输』](#)

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

相关参考

[第 591 页的『agent.properties 文件』](#)

每个代理都有自己的属性文件 agent.properties，其中必须包含代理用于连接到其队列管理器的信息。agent.properties 文件还可以包含用于更改代理程序行为的属性。

[第 739 页的『写入到目标队列的消息上设置的 IBM MQ 消息属性』](#)

当从文件传输到消息时，IBM MQ Managed File Transfer 可在写入到目标队列的第一条消息上设置 IBM MQ 消息属性。如果从“文件到消息”传输失败，那么会设置其他 IBM MQ 消息属性。

将数据从消息传输到文件

通过使用 IBM MQ Managed File Transfer 的消息到文件传输功能，您可以将 IBM MQ 队列上的一条或多条消息的数据传输到文件、数据集或用户文件空间。如果您拥有一个可创建或处理 IBM MQ 消息的应用程序，那么可以使用 IBM MQ Managed File Transfer 的消息到文件传输功能，将这些消息传输到 IBM MQ Managed File Transfer 网络中任何系统上的文件。

要了解有关文件到消息传输的信息，请参阅 [第 250 页的『将数据从文件传输至消息』](#)。



注意：消息到文件传输的源代理不能是协议网桥代理或 Connect:Direct 网桥代理。

您可以将 IBM MQ 消息数据传输到文件。以下类型的消息到文件传输受支持：

- 从单条消息到单个文件
- 从多条消息到单个文件
- 从具有相同 IBM MQ 组标识的多条消息到单个文件。
- 从多条消息到单个文件，包括写入文件的每条消息中数据间的文本或二进制定界符。

如果要从大型消息或众多小型消息传输文件，那么可能需要更改某些 IBM MQ 或 IBM MQ Managed File Transfer 属性。要了解更多信息，请参阅第 394 页的『用于设置与消息大小关联的 IBM MQ 属性和 IBM MQ Managed File Transfer 属性的指南』。

V 8.0.0.8 从 V 8.0.0 修订包 8 开始，将更新 Managed File Transfer 以复原先前由 APAR IT18213 在 V 8.0.0 修订包 5 上除去的传输标识的比较检查以及传输请求 XML 有效内容中 groupId 属性的值。如果这两个标识等同，那么源代理将使用该标识作为消息到文件传输的输入队列上发出的第一次 MQGET 尝试的消息标识匹配选项（而不是组标识匹配选项）。

相关任务

第 264 页的『配置代理以执行消息到文件的传输』

缺省情况下，代理无法执行消息到文件或文件到消息的传输。要启用该功能，必须将代理属性 enableQueueInputOutput 设置为 true。

第 265 页的『示例：从队列传输到单个文件』

您可以通过使用带 **-sq** 参数的 **fteCreateTransfer** 命令，将 IBM MQ 队列指定为文件传输的源。

第 266 页的『示例：将一组消息从队列传输到单个文件』

您可以通过使用带 **-sq** 和 **-sqgi** 参数的 **fteCreateTransfer** 命令，将 IBM MQ 队列上的单个完整组指定为文件传输的源。

第 267 页的『示例：在每条消息的数据前插入文本定界符』

将数据以文本方式从源队列传输到文件时，您可以使用带 **-sq**、**-sqdt** 和 **-sqdp** 参数的 **fteCreateTransfer** 命令，指定将文本定界符插到各条消息的数据前。

第 268 页的『示例：在每条消息的数据后插入二进制定界符』

当以二进制方式从源队列传输到文件时，您可以使用带有 **fteCreateTransfer** 命令的 **-sq**，**-sqdb** 和 **-sqdp** 参数指定在来自各个消息的数据之后插入二进制定界符。

第 243 页的『监视队列和使用变量替换』

您可以使用 **fteCreateMonitor** 命令监视队列，并将消息从受监视的队列传输到文件。从受监视队列读取的第一条消息中任何 IBM MQ 消息属性的值均可在任务 XML 定义中被替换，并用于定义传输行为。

第 271 页的『示例：无法使用 IBM MQ 消息属性实现消息到文件的传输』

将 **usr.UserReturnCode** IBM MQ 消息属性设置为非零值会造成“消息到文件”传输失败。也可以通过设置 **usr.UserSupplement** IBM MQ 消息属性，指定有关失败原因的补充信息。

第 238 页的『示例：配置资源监视器以监视队列』

您可以通过使用带 **-mq** 参数的 **fteCreateMonitor** 命令，将 IBM MQ 队列指定为要由资源监视器监视的资源。

相关参考

第 741 页的『从源队列上的消息中读取的 IBM MQ 消息属性』

在从“消息到文件”传输过程中从源队列读取消息的代理将从该消息中读取 IBM MQ 消息属性。这些属性的值可用于确定传输行为。

第 394 页的『用于设置与消息大小关联的 IBM MQ 属性和 IBM MQ Managed File Transfer 属性的指南』

在读取或写入各种大小的消息时，您可以更改 IBM MQ 属性和 IBM MQ Managed File Transfer 属性以影响 IBM MQ Managed File Transfer 的行为。

配置代理以执行消息到文件的传输

缺省情况下，代理无法执行消息到文件或文件到消息的传输。要启用该功能，必须将代理属性 enableQueueInputOutput 设置为 true。

关于此任务

如果尝试从未将 enableQueueInputOutput 属性设置为 true 的源代理执行消息到文件的传输，那么传输失败。发布至协调队列管理器的传输日志消息包含以下消息：

```
BFGI00197E: An attempt to read from a queue was rejected by the source agent.  
The agent must have enableQueueInputOutput=true set in the agent.properties file  
to support transferring from a queue.
```

要启用代理在队列上写入和读取，请执行以下步骤：

过程

1. 使用 **fteStopAgent** 命令停止源代理。
2. 编辑 `agent.properties` 文件以包含行 `enableQueueInputOutput=true`。
`agent.properties` 文件位于目录 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/source_agent_name` 中。
3. 使用 **fteStartAgent** 命令启动源代理。

相关概念

[第 263 页的『将数据从消息传输到文件』](#)

通过使用 IBM MQ Managed File Transfer 的消息到文件传输功能，您可以将 IBM MQ 队列上的一条或多条消息的数据传输到文件、数据集或用户文件空间。如果您拥有一个可创建或处理 IBM MQ 消息的应用程序，那么可以使用 IBM MQ Managed File Transfer 的消息到文件传输功能，将这些消息传输到 IBM MQ Managed File Transfer 网络中任何系统上的文件。

相关任务

[第 265 页的『示例：从队列传输到单个文件』](#)

您可以通过使用带 **-sq** 参数的 **fteCreateTransfer** 命令，将 IBM MQ 队列指定为文件传输的源。

[第 266 页的『示例：将一组消息从队列传输到单个文件』](#)

您可以通过使用带 **-sq** 和 **-sqgi** 参数的 **fteCreateTransfer** 命令，将 IBM MQ 队列上的单个完整组指定为文件传输的源。

[第 267 页的『示例：在每条消息的数据前插入文本定界符』](#)

将数据以文本方式从源队列传输到文件时，您可以使用带 **-sq**、**-sqdt** 和 **-sqdp** 参数的 **fteCreateTransfer** 命令，指定将文本定界符插到各条消息的数据前。

[第 268 页的『示例：在每条消息的数据后插入二进制定界符』](#)

当以二进制方式从源队列传输到文件时，您可以使用带有 **fteCreateTransfer** 命令的 **-sq**、**-sqdb** 和 **-sqdp** 参数指定在来自各个消息的数据之后插入二进制定界符。

[第 243 页的『监视队列和使用变量替换』](#)

您可以使用 **fteCreateMonitor** 命令监视队列，并将消息从受监视的队列传输到文件。从受监视队列读取的第一条消息中任何 IBM MQ 消息属性的值均可在任务 XML 定义中被替换，并用于定义传输行为。

[第 271 页的『示例：无法使用 IBM MQ 消息属性实现消息到文件的传输』](#)

将 `usr.UserReturnCode` IBM MQ 消息属性设置为非零值会造成“消息到文件”传输失败。也可以通过设置 `usr.UserSupplement` IBM MQ 消息属性，指定有关失败原因的补充信息。

相关参考

[第 591 页的『agent.properties 文件』](#)

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

示例：从队列传输到单个文件

您可以通过使用带 **-sq** 参数的 **fteCreateTransfer** 命令，将 IBM MQ 队列指定为文件传输的源。

关于此任务

源数据包含在队列 `START_QUEUE` 上的三条消息中。该队列必须位于源代理的队列管理器 `QM_NEPTUNE` 中。

过程

输入以下命令：

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE
```



```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/output.txt  
-t text -sqdt "\n\u002D\u002D\u002D\u002D\n" -sqdp prefix -sq START_QUEUE
```

文本定界符通过源代理 AGENT_NEPTUNE 添加到 START_QUEUE 上四条消息中每条消息的数据开头位置。该数据写入到目标文件 /out/output.txt 中。

相关概念

[第 263 页的『将数据从消息传输到文件』](#)

通过使用 IBM MQ Managed File Transfer 的消息到文件传输功能，您可以将 IBM MQ 队列上的一条或多条消息的数据传输到文件、数据集或用户文件空间。如果您拥有一个可创建或处理 IBM MQ 消息的应用程序，那么可以使用 IBM MQ Managed File Transfer 的消息到文件传输功能，将这些消息传输到 IBM MQ Managed File Transfer 网络中任何系统上的文件。

相关任务

[第 264 页的『配置代理以执行消息到文件的传输』](#)

缺省情况下，代理无法执行消息到文件或文件到消息的传输。要启用该功能，必须将代理属性 enableQueueInputOutput 设置为 true。

[第 265 页的『示例：从队列传输到单个文件』](#)

您可以通过使用带 **-sq** 参数的 **fteCreateTransfer** 命令，将 IBM MQ 队列指定为文件传输的源。

[第 266 页的『示例：将一组消息从队列传输到单个文件』](#)

您可以通过使用带 **-sq** 和 **-sqgi** 参数的 **fteCreateTransfer** 命令，将 IBM MQ 队列上的单个完整组指定为文件传输的源。

[第 268 页的『示例：在每条消息的数据后插入二进制定界符』](#)

当以二进制方式从源队列传输到文件时，您可以使用带有 **fteCreateTransfer** 命令的 **-sq**、**-sqdb** 和 **-sqdp** 参数指定在来自各个消息的数据之后插入二进制定界符。

[第 243 页的『监视队列和使用变量替换』](#)

您可以使用 **fteCreateMonitor** 命令监视队列，并将消息从受监视的队列传输到文件。从受监视队列读取的第一条消息中任何 IBM MQ 消息属性的值均可在任务 XML 定义中被替换，并用于定义传输行为。

[第 271 页的『示例：无法使用 IBM MQ 消息属性实现消息到文件的传输』](#)

将 **usr.UserReturnCode** IBM MQ 消息属性设置为非零值会造成“消息到文件”传输失败。也可以通过设置 **usr.UserSupplement** IBM MQ 消息属性，指定有关失败原因的补充信息。

相关参考

[第 497 页的『fteCreateTransfer（创建新的文件传输）』](#)

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

示例：在每条消息的数据后插入二进制定界符

当以二进制方式从源队列传输到文件时，您可以使用带有 **fteCreateTransfer** 命令的 **-sq**、**-sqdb** 和 **-sqdp** 参数指定在来自各个消息的数据之后插入二进制定界符。

关于此任务

在该示例中，队列 START_QUEUE 上存在三条消息。该队列位于源代理的队列管理器 QM_NEPTUNE 中。插入到每条消息的数据后的二进制定界符必须表示为以逗号分隔的十六进制字节的列表，例如：
x34,xE7,xAE。

过程

输入以下命令：

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/binary.file  
-sqdp postfix -sqdb x34,xE7,xAE -sq START_QUEUE
```

二进制定界符通过源代理 AGENT_NEPTUNE 附加到 START_QUEUE 上三条消息中每条的数据之后。此数据将写入目标文件 /out/binary.file。

相关概念

[第 263 页的『将数据从消息传输到文件』](#)

通过使用 IBM MQ Managed File Transfer 的消息到文件传输功能，您可以将 IBM MQ 队列上的一条或多条消息的数据传输到文件、数据集或用户文件空间。如果您拥有一个可创建或处理 IBM MQ 消息的应用程序，那么可以使用 IBM MQ Managed File Transfer 的消息到文件传输功能，将这些消息传输到 IBM MQ Managed File Transfer 网络中任何系统上的文件。

相关任务

[第 264 页的『配置代理以执行消息到文件的传输』](#)

缺省情况下，代理无法执行消息到文件或文件到消息的传输。要启用该功能，必须将代理属性 enableQueueInputOutput 设置为 true。

[第 265 页的『示例：从队列传输到单个文件』](#)

您可以通过使用带 **-sq** 参数的 **fteCreateTransfer** 命令，将 IBM MQ 队列指定为文件传输的源。

[第 266 页的『示例：将一组消息从队列传输到单个文件』](#)

您可以通过使用带 **-sq** 和 **-sqgi** 参数的 **fteCreateTransfer** 命令，将 IBM MQ 队列上的单个完整组指定为文件传输的源。

[第 267 页的『示例：在每条消息的数据前插入文本定界符』](#)

将数据以文本方式从源队列传输到文件时，您可以使用带 **-sq**、**-sqdt** 和 **-sqdp** 参数的 **fteCreateTransfer** 命令，指定将文本定界符插到各条消息的数据前。

[第 243 页的『监视队列和使用变量替换』](#)

您可以使用 **fteCreateMonitor** 命令监视队列，并将消息从受监视的队列传输到文件。从受监视队列读取的第一条消息中任何 IBM MQ 消息属性的值均可在任务 XML 定义中被替换，并用于定义传输行为。

[第 271 页的『示例：无法使用 IBM MQ 消息属性实现消息到文件的传输』](#)

将 **usr.UserReturnCode** IBM MQ 消息属性设置为非零值会造成“消息到文件”传输失败。也可以通过设置 **usr.UserSupplement** IBM MQ 消息属性，指定有关失败原因的补充信息。

相关参考

[第 497 页的『fteCreateTransfer（创建新的文件传输）』](#)

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

监视队列和使用变量替换

您可以使用 **fteCreateMonitor** 命令监视队列，并将消息从受监视的队列传输到文件。从受监视队列读取的第一条消息中任何 IBM MQ 消息属性的值均可在任务 XML 定义中被替换，并用于定义传输行为。

关于此任务

在本示例中，源代理名为 AGENT_VENUS，与 QM_VENUS 连接。AGENT_VENUS 监视的队列名为 START_QUEUE，位于 QM_VENUS 上。代理每隔 30 分钟轮询一次队列。

将一组完整的消息写入到队列中时，监视器任务会将这组消息发送到某个目标代理中的文件，这些目标代理均连接到队列管理器 QM_MARS。这组消息传输到的文件的名称由该组中第一条消息的 IBM MQ 消息属性 **usr.fileName** 来定义。这组消息发送到的代理的名称由该组中第一条消息的 IBM MQ 消息属性 **usr.toAgent** 来定义。如果 **usr.toAgent** 头未设置，那么用于该目标代理的缺省值为 AGENT_MAGENTA。

当指定 **useGroups="true"** 时，如果未同时指定 **groupId="\${GROUPID}"**，那么仅传输队列中的第一条消息。例如，如果使用变量替换来生成 **fileName**，那么 **a.txt** 的内容有可能不正确。这是因为 **fileName** 由监视器生成，但传输实际获取的一条消息并非应生成名为 **fileName** 的文件的消息。

过程

1. 创建任务 XML，以定义监视器被触发时应执行的任务。

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="${toAgent}" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/${fileName}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

替换为 IBM MQ 消息头值的变量以**粗体**突出显示。该任务 XML 保存到文件 /home/USER1/task.xml

2. 创建一个资源监视器以监视队列 START_QUEUE。

提交以下命令：

```
fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA
```

3. 用户或程序将一组消息写入队列 START_QUEUE 中。

该组中的第一条消息设置了以下 IBM MQ 消息属性：

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. 写入整组消息时，会触发监视器。代理将在任务 XML 中替换 IBM MQ 消息属性。

这导致任务 XML 转换为：

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/larmer.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

结果

这样会执行由任务 XML 定义的传输。由 AGENT_VENUS 从 START_QUEUE 读取的整组消息会写入到名为 /reports/larmer.rpt 的文件中，该文件位于运行 AGENT_VIOLET 的系统上。

下一步做什么

将每条消息传输到一个单独的文件

如果要监视某个队列，并将每条消息传输到一个单独的文件，可以使用与本主题中上述方法类似的方法。

1. 如前所述创建监视器，在 **fteCreateMonitor** 命令上指定 **-tr completeGroups** 参数。
2. 在任务 XML 中指定以下内容：

```
<queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
```

但是，在将消息放置到源队列上时，请不要将其放置在 IBM MQ 组中。向每条消息添加 IBM MQ 消息属性。例如，指定 **usr.filename** 属性，其中每条消息均有唯一的文件名值。这样实际上会使得 IBM MQ Managed File Transfer 代理将源队列上的每条消息视为单独的组。

相关概念

[第 229 页的『资源监控』](#)

您可以监视 IBM MQ Managed File Transfer 资源；例如，队列或目录。当满足针对该资源的条件时，资源监视器将启动任务，如文件传输。您可以使用 **fteCreateMonitor** 命令或 IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

[第 263 页的『将数据从消息传输到文件』](#)

通过使用 IBM MQ Managed File Transfer 的消息到文件传输功能，您可以将 IBM MQ 队列上的一条或多条消息的数据传输到文件、数据集或用户文件空间。如果您拥有一个可创建或处理 IBM MQ 消息的应用程序，那么可以使用 IBM MQ Managed File Transfer 的消息到文件传输功能，将这些消息传输到 IBM MQ Managed File Transfer 网络中任何系统上的文件。

[第 239 页的『使用变量替换定制 MFT 任务』](#)

在满足活动资源监视器的触发条件时，会调用已定义的任务。除了每次使用相同的目标代理或相同的目标文件名调用传输或命令任务之外，您还可以在运行时修改任务定义。可通过将变量名插入到任务定义 XML 中来实现此目的。在监视器确定已满足触发条件并且任务定义包含变量名时，将用变量值替换变量名，然后调用任务。

相关任务

[第 233 页的『配置监视器任务以启动命令和脚本』](#)

资源监视器的作用不只限于将执行文件传输作为其相关的任务。您还可以配置监视器，以从监视代理调用其他命令，包括可执行程序、Ant 脚本或 JCL 作业。要调用命令，请编辑监视任务定义 XML，以将一个或多个命令元素包含在对应的命令调用参数（例如，自变量和属性）中。

[第 238 页的『示例：配置资源监视器以监视队列』](#)

您可以通过使用带 **-mq** 参数的 **fteCreateMonitor** 命令，将 IBM MQ 队列指定为要由资源监视器监视的资源。

相关参考

[第 478 页的『fteCreateMonitor（新建资源监视器）』](#)

fteCreateMonitor 命令从命令行创建并启动新的资源监视器。您可以使用 WebSphere MQ Managed File Transfer 监视资源（例如，目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

[第 741 页的『从源队列上的消息中读取的 IBM MQ 消息属性』](#)

在从“消息到文件”传输过程中从源队列读取消息的代理将从该消息中读取 IBM MQ 消息属性。这些属性的值可用于确定传输行为。

[第 389 页的『如果队列资源监视器启动的传输所创建的目标文件包含错误数据，那么该执行哪些操作？』](#)

您可以创建资源监视器来监视队列，并将队列上的消息或一组消息传输到文件。可以通过使用消息上的 MQMD 消息头符描述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。

示例：无法使用 IBM MQ 消息属性实现消息到文件的传输

将 **usr.UserReturnCode** IBM MQ 消息属性设置为非零值会造成“消息到文件”传输失败。也可以通过设置 **usr.UserSupplement** IBM MQ 消息属性，指定有关失败原因的补充信息。

在本示例中，队列 INPUT QUEUE 和文件 /home/user/output.file 之间正在执行一个传输。

向队列 INPUT QUEUE 写入消息的用户希望停止正在进行的传输，并删除已写入目标文件的所有数据。

1. 用户将消息写入设置了以下 IBM MQ 消息属性的队列 INPUT QUEUE 中:

此消息包含以下信息:

第 264 页的『配置代理以执行消息到文件的传输』

缺省情况下，代理无法执行消息到文件或文件到消息的传输。要启用该功能，必须将代理属性 `enableQueueInputOutput` 设置为 `true`。

[第 265 页的『示例：从队列传输到单个文件』](#)

您可以通过使用带 **-sq** 参数的 **fteCreateTransfer** 命令，将 IBM MQ 队列指定为文件传输的源。

[第 266 页的『示例：将一组消息从队列传输到单个文件』](#)

您可以通过使用带 **-sq** 和 **-sqgi** 参数的 **fteCreateTransfer** 命令，将 IBM MQ 队列上的单个完整组指定为文件传输的源。

[第 267 页的『示例：在每条消息的数据前插入文本定界符』](#)

将数据以文本方式从源队列传输到文件时，您可以使用带 **-sq**、**-sqdt** 和 **-sqdp** 参数的 **fteCreateTransfer** 命令，指定将文本定界符插到各条消息的数据前。

[第 268 页的『示例：在每条消息的数据后插入二进制定界符』](#)

当以二进制方式从源队列传输到文件时，您可以使用带有 **fteCreateTransfer** 命令的 **-sq**、**-sqdb** 和 **-sqdp** 参数指定在来自各个消息的数据之后插入二进制定界符。

[第 243 页的『监视队列和使用变量替换』](#)

您可以使用 **fteCreateMonitor** 命令监视队列，并将消息从受监视的队列传输到文件。从受监视队列读取的第一条消息中任何 IBM MQ 消息属性的值均可在任务 XML 定义中被替换，并用于定义传输行为。

相关参考

[第 741 页的『从源队列上的消息中读取的 IBM MQ 消息属性』](#)

在从“消息到文件”传输过程中从源队列读取消息的代理将从该消息中读取 IBM MQ 消息属性。这些属性的值可用于确定传输行为。

列出 IBM MQ Managed File Transfer 代理

可使用命令行或 IBM MQ Explorer 来列出已向特定队列管理器注册的代理。

关于此任务

要使用命令行列出代理，请参阅 [fteListAgents](#) 命令。

要使用 IBM MQ Explorer 列出代理，请在“导航器”视图中单击协调队列管理器名称下的代理。

如果代理程序未由 **fteListAgents** 命令列出或未显示在 IBM MQ Explorer 中，请使用以下主题中的诊断流程图来查找并解决问题：[如果代理程序未由 fteListAgents 命令列出](#)。

相关参考

[第 532 页的『fteListAgents（列出协调队列管理器的 IBM MQ Managed File Transfer 代理）』](#)

通过命令行，使用 **fteListAgents** 命令可列出向特定协调队列管理器注册的所有 IBM MQ Managed File Transfer 代理。

[第 700 页的『代理状态值』](#)

fteListAgents 和 **fteShowAgentDetails** 命令生成代理状态信息。该状态有多种可能的值。

[第 564 页的『fteShowAgentDetails（显示 IBM MQ Managed File Transfer 代理详细信息）』](#)

使用 **fteShowAgentDetails** 命令可显示特定 IBM MQ Managed File Transfer 代理的详细信息。这些是由其 IBM MQ Managed File Transfer 协调队列管理器存储的详细信息。

停止 IBM MQ Managed File Transfer 代理

您可以通过命令行停止代理。在停止代理时，您将停顿代理并允许代理在停止前完成其当前文件传输。您还可以在命令行上指定 **-i** 参数来立即停止代理。在代理停止时，您无法使用该代理传输文件，直至重新启动代理。

开始之前

如果想要检查与队列管理器关联的代理的名称，可以使用 IBM MQ Explorer 或命令行列出代理，请参阅 [fteListAgents](#) 命令。

关于此任务

要从命令行停止代理，请参阅 [fteStopAgent](#)。

如果已将代理程序配置为作为 Windows 服务运行，那么运行 **fteStopAgent** 命令也会停止 Windows 服务。或者，可以使用 Windows 服务工具停止服务来停止代理。有关更多信息，请参阅主题 [第 216 页的『将代理作为 Windows 服务启动』](#)。

相关参考

[第 576 页的『fteStopAgent（停止 IBM MQ Managed File Transfer 代理）』](#)

使用 **fteStopAgent** 命令以受控方式停止 IBM MQ Managed File Transfer 代理程序，或者使用 **-i** 参数在必要时立即停止代理程序。

[第 274 页的『在 z/OS 上停止代理』](#)

如果要在 z/OS 上作为来自 JCL 的启动式任务运行 IBM MQ Managed File Transfer 代理程序，那么该代理程序将接受 z/OS 操作员命令 **MODIFY** 和 **STOP** 以及 **fteStopAgent** 命令。

在 z/OS 上停止代理

如果要在 z/OS 上作为来自 JCL 的启动式任务运行 IBM MQ Managed File Transfer 代理程序，那么该代理程序将接受 z/OS 操作员命令 **MODIFY** 和 **STOP** 以及 **fteStopAgent** 命令。

将使用已启动任务，由于它在特定的用户标识下运行，不受最终用户注销影响。

注: **V8.0.0.6** 已启动的任务通常由可能无登录权限的管理用户运行，因此，无法以运行代理的用户身份登录到 z/OS 系统。无法对该代理发出 **fteStartAgent**、**fteStopAgent** 和 **fteSetAgentTraceLevel** 命令以及指定了 **-d** 参数的 **fteShowAgentDetails** 命令。

从 IBM MQ 8.0.0 修订包 6 开始，代理属性 **adminGroup** 可用于 z/OS 上的 Managed File Transfer 代理程序。您可以定义安全管理器组（例如，*MFTADMIN*），然后将已启动任务用户标识和管理员 TSO 标识添加到该组。编辑代理属性文件并将 **adminGroup** 属性设置为该安全管理器组的名称。

```
adminGroup=MFTADMIN
```

然后，该组成员可以对作为已启动任务运行的代理发出 **fteStartAgent**、**fteStopAgent** 和 **fteSetAgentTraceLevel** 命令以及指定了 **-d** 参数的 **fteShowAgentDetails** 命令。

有关更多信息，请参阅[第 591 页的『agent.properties 文件』](#)中的 **adminGroup** 属性。

使用 z/OS MODIFY 命令 (F) 以受控方式关闭代理

MODIFY 命令允许您以受控方式停止代理，可作为 **fteStopAgent** 命令的备用选择。代理将完成当前正在进行的所有传输，但代理不会启动任何新传输。

例如：

```
F job_name,APPL=STOP
```

其中，*job_name* 是代理进程正在其下运行的作业。

使用 z/OS STOP 命令 (P) 立即关闭代理

STOP 命令等同于使用具有 **-i** 参数的 **fteStopAgent** 命令执行立即停止。即使代理当前正在传输文件，也会立即停止代理。

例如：

```
P job_name
```

其中，*job_name* 是代理进程正在其下运行的作业。

协议网桥

通过协议网桥，IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件，而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

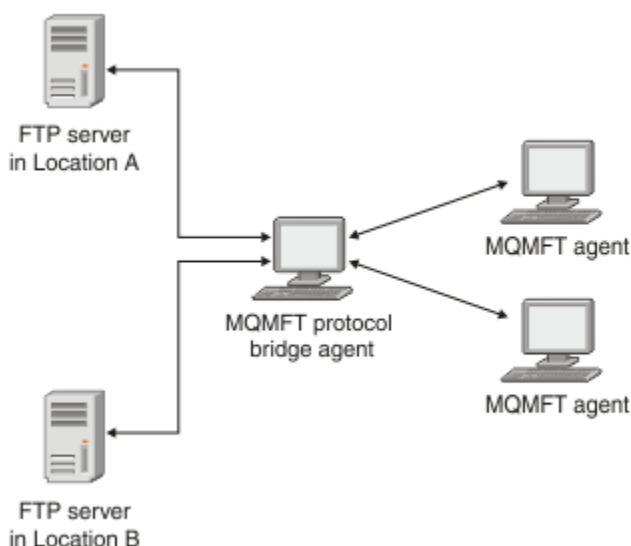
协议网桥作为 IBM MQ Managed File Transfer 的服务组件的一部分提供。在运行 MQMFT 的单个系统上可以有多个与不同文件服务器连接的专用代理程序。

您可以使用协议网桥代理同时将文件传输到多个端点。MQMFT 提供了一个名为 `ProtocolBridgeProperties.xml` 的文件，您可以编辑该文件以定义要将文件传输到的不同协议文件服务器。**fteCreateBridgeAgent** 命令为您将缺省协议文件服务器的详细信息添加到 `ProtocolBridgeProperties.xml` 中。该文件在第 611 页的『协议网桥属性文件格式』中进行了描述。

您可以使用协议网桥代理来执行以下操作：

- 将文件从 MQMFT 网络上载到使用 FTP、FTPS 或 SFTP 的远程服务器。
- 使用 FTP、FTPS 或 SFTP，将文件从远程服务器下载到 MQMFT 网络

注：协议网桥代理只能支持允许通过绝对文件路径访问文件的 FTP、FTPS 或 SFTP 服务器。如果在传输请求中指定了相对文件路径，那么该协议网桥代理会尝试根据用于登录协议服务器的主目录将相对路径转换为绝对文件路径。该协议网桥代理程序不支持这些允许仅根据当前目录访问文件的协议服务器。



图形显示了位于不同位置的两个 FTP 服务器。这两个 FTP 服务器用于通过 IBM MQ Managed File Transfer 代理来交换文件。协议网桥代理位于 FTP 服务器和其余的 MQMFT 网络之间，并且配置为与两个 FTP 服务器进行通信。

确保除了协议网桥代理外，在 MQMFT 网络中还有另一个代理。协议网桥代理只是到 FTP、FTPS 或 SFTP 服务器的网桥，不能将传输的文件写入本地磁盘。如果您要与 FTP、FTPS 或 SFTP 服务器进行文件传输，那么必须使用协议网桥代理作为文件传输的目标或源（代表 FTP、FTPS 或 SFTP 服务器）以及另一个标准代理作为相应的源或目标。

使用协议网桥传输文件时，该网桥必须有权读取包含要传输的文件的源或目标目录。例如，如果您要传输目录 `/home/fte/bridge`（只具有执行许可权 `(d--x--x--x)`）中的文件，那么您从该目录尝试的任何传输都会失败，并发出以下错误消息：

```
BFGBR0032E: Attempt to read filename from the protocol file server
has failed with server error 550. Failed to open file.
```

配置协议网桥代理

协议网桥代理类似一个标准的 MQMFT 代理。通过使用 **fteCreateBridgeAgent** 命令创建协议网桥代理。您可以使用 **ProtocolBridgeProperties.xml** 文件配置协议网桥代理（这在第 611 页的『协议网桥属性文件格式』中进行了描述）。如果使用更早版本，那么请使用高级代理属性中描述的特定协议网桥属性配置代理。对于所有版本，您还可以按照第 281 页的『映射文件服务器的凭证』中所述配置凭证映射。为特定协议文件服务器配置协议网桥代理程序后，您可以将该代理程序仅用于该用途。

协议网桥恢复

如果由于文件服务器不可用，协议网桥代理无法连接到文件服务器，那么所有文件传输请求都将排队，直至文件服务器变为可用。如果由于代理使用错误的凭证，协议网桥代理无法连接到文件服务器，那么传输失败并且传输日志消息会反映该错误。如果协议网桥代理由于任一原因而终止，那么所有请求的文件传输都将保留并在协议网桥重新启动时继续。

文件传输期间，文件通常在目标上作为临时文件写入，然后完成传输后重命名。然而，如果传输的目标是配置为限制写入的协议文件服务器（用户可以向协议文件服务器上载文件，但不能以任何方式更改这些上载文件；有效用户只能写入一次），那么传输的文件将直接写入到目标中。这意味着如果传输期间出现问题，部分写入的文件仍位于目标协议文件服务器，但 IBM MQ Managed File Transfer 不能删除或编辑这些文件。在这种情况下，传输失败。

相关任务

第 285 页的『示例：如何配置协议网桥代理以将专用密钥凭证用于 UNIX SFTP 服务器』

该示例演示了如何生成和配置 **ProtocolBridgeCredentials.xml** 文件。该示例是典型示例，细节会随着您的平台而不同，但原理是一致的。

第 276 页的『使用 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性』

定义要使用 **ProtocolBridgeProperties.xml** 文件（由 IBM MQ Managed File Transfer 在代理配置目录中提供）传输文件的一个或多个协议文件服务器的属性。

相关参考

第 463 页的『fteCreateBridgeAgent（创建和配置 IBM MQ Managed File Transfer 协议网桥代理）』

fteCreateBridgeAgent 命令创建协议网桥代理及其关联的配置。针对您要与之进行文件的发送和接收操作的每个文件服务器，创建一个协议网桥代理。

第 281 页的『映射文件服务器的凭证』

通过使用协议网桥代理的缺省凭证映射功能或编写您自己的用户出口，将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器的用户凭证。IBM MQ Managed File Transfer 提供了执行用户凭证映射的样本用户出口。

第 1007 页的『ProtocolBridgeCredentialExit.java 接口』

第 365 页的『协议网桥凭证用户出口样本』

第 728 页的『协议网桥提供的 FTPS 服务器支持』

协议网桥支持 FTPS 协议的子集，如 RFC-2228、RFC-4217 和名为 *Secure FTP over SSL* 的因特网草案所定义。

使用 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性

定义要使用 **ProtocolBridgeProperties.xml** 文件（由 IBM MQ Managed File Transfer 在代理配置目录中提供）传输文件的一个或多个协议文件服务器的属性。

关于此任务

fteCreateBridgeAgent 命令在代理程序配置目录 **MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name** 中创建 **ProtocolBridgeProperties.xml** 文件。如果运行命令时指定缺省值，那么该命令还会在此文件中为缺省协议文件服务器创建一个条目。

如果您要添加更多非缺省协议服务器，请编辑此文件以定义它们的属性。该示例添加一个额外的 FTP 服务器。

过程

1. 通过将以下行作为 `<tns:serverProperties>` 的子元素插入此文件中，定义协议文件服务器：

```
<tns:ftpServer name="myserver" host="myhost.hursley.ibm.com" port="1234"
  platform="windows"
    timeZone="Europe/London" locale="en-GB" fileEncoding="UTF-8"
    listFormat="unix" limitedWrite="false" >
<tns:limits maxListFileNames="10" maxListDirectoryLevels="500"/>
```

2. 然后更改属性的值：

- `name` 是协议文件服务器的名称
- `host` 是协议文件服务器的主机名或 IP 地址
- `port` 是协议文件服务器的端口号
- `platform` 是运行协议文件服务器的平台
- `timeZone` 是协议文件服务器运行的时区
- `locale` 是协议文件服务器上使用的语言
- `fileEncoding` 是协议文件服务器的字符编码
- `listFormat` 是从协议文件服务器返回的文件列表格式
- `limitedWrite` 确定是否在写入文件服务器时遵循缺省方式，这会创建临时文件，然后在完成传输后重命名此文件。对于配置为只写的文件服务器，会使用其最终名称直接创建此文件。该属性的值可以为 `true` 或 `false`。`limitedWrite` 属性和 `doNotUseTempOutputFile` 代理属性在协议网桥代理中共同使用。如果要使用临时文件，那么切勿设置 `doNotUseTempOutputFile` 的值，并且必须将 `limitedWrite` 的值设置为 `false`。其他设置组合意味着不会使用临时文件。
- `maxListFileNames` 是在协议文件服务器上的目录中扫描文件名时收集的最大名称数。
- `maxListDirectoryLevels` 是在协议文件服务器上的目录中扫描文件名时要递归的最大目录级别数。

有关这些属性的更多详细信息，包括它们是必需的还是可选的以及它们的缺省值，请参阅第 611 页的『[协议网桥属性文件格式](#)』。

相关参考

第 611 页的『[协议网桥属性文件格式](#)』

代理配置目录中的 `ProtocolBridgeProperties.xml` 文件定义协议文件服务器的属性。

第 724 页的『[IBM MQ Managed File Transfer 使用的正则表达式](#)』

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 `Connect:Direct` 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 `java.util.regex` API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

使用出口类 (ProtocolBridgePropertiesExit2) 查找协议文件服务器属性

如果您有大量协议文件服务器，可以实现

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` 接口来查找传输中引用的协议文件服务器属性。您可以实现此接口，而不是维护 `ProtocolBridgeProperties.xml` 文件。IBM MQ Managed File Transfer 提供了一个用于查找协议文件服务器属性的用户出口样本。

配置用于查找协议网桥属性的用户出口

关于此任务

用于查找协议网桥属性的任何用户出口都必须实现接口

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2`。有关更多信息，请参阅第 1009 页的『[ProtocolBridgePropertiesExit2.java 接口](#)』。

您可以通过类似方式将多个协议服务器属性出口与其他用户出口链接在一起。调用出口的顺序是使用 `protocolBridgePropertiesExitClasses` 属性在代理属性文件中指定这些出口的顺序。初始化方法

均单独返回，并且如果一个或多个方法返回一个 `false` 值，那么代理不会启动。代理事件日志中将报告此错误。

所有出口的 `getProtocolServerProperties` 方法只返回一个总体结果。如果方法作为结果代码返回属性对象，那么此值便是返回的结果，并且不会调用后续出口的 `getProtocolServerProperties` 方法。如果方法作为结果代码返回值 `null`，那么将调用下一个出口的 `getProtocolServerProperties` 方法。如果没有任何后续出口，那么会返回空结果。总体结果代码 `null` 会被协议网桥代理视为查询失败。

建议您使用 `ProtocolBridgePropertiesExit2.java` 接口，但有关 `ProtocolBridgePropertiesExit.java` 接口的信息，请参阅第 279 页的『使用出口类 (`ProtocolBridgePropertiesExit.java`) 查找协议文件服务器属性』。

要运行出口，请完成以下步骤：

过程

1. 编译协议服务器属性用户出口。
2. 创建包含已编译出口及其包结构的 Java 归档 (JAR) 文件。
3. 将包含出口类的 JAR 文件放置在协议网桥代理的 `exits` 目录中。此目录位于 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 目录中。
4. 编辑协议网桥代理的属性文件以包括属性 `protocolBridgePropertiesExitClasses`。对于此属性的值，请指定用于实现协议网桥服务器属性用户出口的类的逗号分隔列表。将按照此列表中指定出口类的顺序来调用这些出口类。有关更多信息，请参阅第 591 页的『agent.properties 文件』。
5. 您可以选择指定 `protocolBridgePropertiesConfiguration` 属性。您为此属性指定的值会作为字符串传递到 `protocolBridgePropertiesExitClasses` 指定的出口类的 `initialize()` 方法。有关更多信息，请参阅第 591 页的『agent.properties 文件』。

使用用户出口样本

关于此任务

`MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` 目录及第 367 页的『协议网桥属性用户出口样本』主题中提供了用于查找协议网桥属性的用户出口样本。

`SamplePropertiesExit2.java` 出口会读取包含用于协议服务器的属性的属性文件。属性文件中每个条目的格式如下：

```
serverName=type://host:port
```

属性文件的位置是从协议网桥代理属性 `protocolBridgePropertiesConfiguration` 中获取。

要运行用户出口样本，请完成以下步骤：

过程

1. 编译 `SamplePropertiesExit2.java` 文件。
2. 创建一个包含编译的出口及其程序包结构的 JAR 文件。
3. 将 JAR 文件放到 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/exits` 目录中。
4. 编辑 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` 文件以包含以下行：

```
protocolBridgePropertiesExitClasses=SamplePropertiesExit2
```

5. 在目录 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` 中创建协议网桥属性文件，例如 `protocol_bridge_properties.properties`。编辑此文件以包含以下格式的题目：


```
serverName=type://host:port
```

6. 编辑 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/agent.properties` 文件以包含以下行:

```
protocolBridgePropertiesConfiguration=MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/protocol_bridge_properties.properties
```

您必须使用 `protocol_bridge_properties.properties` 文件的绝对路径。

7. 通过使用 **`fteStartAgent`** 命令来启动协议网桥代理。

相关概念

第 275 页的『[协议网桥](#)』

通过协议网桥, IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件, 而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

相关参考

第 1009 页的『[ProtocolBridgePropertiesExit2.java 接口](#)』

第 367 页的『[协议网桥属性用户出口样本](#)』

第 591 页的『[agent.properties 文件](#)』

每个代理都有自己的属性文件 `agent.properties`, 其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

第 463 页的『[fteCreateBridgeAgent \(创建和配置 IBM MQ Managed File Transfer 协议网桥代理\)](#)』

`fteCreateBridgeAgent` 命令创建协议网桥代理及其关联的配置。针对您要与之进行文件的发送和接收操作的每个文件服务器, 创建一个协议网桥代理。

使用出口类 (*ProtocolBridgePropertiesExit.java*) 查找协议文件服务器属性

如果您有大量协议文件服务器, 可以实现

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` 接口来查找传输中引用的协议文件服务器属性。您可以实现此接口, 而不是维护 `ProtocolBridgeProperties.xml` 文件。建议您使用 `ProtocolBridgePropertiesExit2.java` 接口, 但 `ProtocolBridgePropertiesExit.java` 接口也受支持。如果在 WebSphere MQ File Transfer Edition 中存在 `ProtocolBridgePropertiesExit.java` 接口的现有实现, 那么可以在 IBM WebSphere MQ V7.5 或更高版本中使用该实现。`ProtocolBridgePropertiesExit2.java` 中新的 `getCredentialLocation` 方法使用 `ProtocolBridgeCredentials.xml` 文件的缺省位置, 即您的主目录。

配置用于查找协议网桥属性的用户出口

用于查找协议网桥属性的任何用户出口都必须实现接口

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit`。有关更多信息, 请参阅第 280 页的『[ProtocolBridgePropertiesExit.java 接口](#)』。

您可以通过类似方式将多个协议服务器属性出口与其他用户出口链接在一起。调用出口的顺序是使用 `protocolBridgePropertiesExitClasses` 属性在代理属性文件中指定这些出口的顺序。初始化方法均单独返回, 并且如果一个或多个方法返回一个 `false` 值, 那么代理不会启动。代理事件日志中将报告此错误。

所有出口的 `getProtocolServerProperties` 方法只返回一个总体结果。如果方法作为结果代码返回属性对象, 那么此值便是返回的结果, 并且不会调用后续出口的 `getProtocolServerProperties` 方法。如果方法作为结果代码返回值 `null`, 那么将调用下一个出口的 `getProtocolServerProperties` 方法。如果没有任何后续出口, 那么会返回空结果。总体结果代码 `null` 会被协议网桥代理视为查询失败。

要运行出口, 请完成以下步骤:

1. 编译协议服务器属性用户出口。
2. 创建包含已编译出口及其包结构的 Java 归档 (JAR) 文件。

3. 将包含出口类的 JAR 文件放置在协议网桥代理的 `exits` 目录中。此目录位于 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 目录中。
4. 编辑协议网桥代理的属性文件以包括属性 `protocolBridgePropertiesExitClasses`。对于此属性的值，请指定用于实现协议网桥服务器属性用户出口的类的逗号分隔列表。将按照此列表中指定出口类的顺序来调用这些出口类。有关更多信息，请参阅第 591 页的『[agent.properties 文件](#)』。
5. 您可以选择指定 `protocolBridgePropertiesConfiguration` 属性。您为此属性指定的值会作为字符串传递到 `protocolBridgePropertiesExitClasses` 指定的出口类的 `initialize()` 方法。有关更多信息，请参阅第 591 页的『[agent.properties 文件](#)』。

ProtocolBridgePropertiesExit.java 接口

```
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;
import java.util.Properties;

/**
 * An interface that is implemented by classes that are to be invoked as part of
 * user exit routine processing. This interface defines methods that will be
 * invoked by a protocol bridge agent to look up properties for protocol servers
 * that are referenced in transfers.
 * <p>
 * There will be one instance of each implementation class for each protocol
 * bridge agent. The methods can be called from different threads so the methods
 * must be synchronised.
 */
public interface ProtocolBridgePropertiesExit {

    /**
     * Invoked once when a protocol bridge agent is started. It is intended to
     * initialize any resources that are required by the exit.
     *
     * @param bridgeProperties
     *        The values of properties defined for the protocol bridge.
     *        These values can only be read, they cannot be updated by the
     *        implementation.
     * @return {@code true} if the initialization is successful and {@code
     *         false} if unsuccessful. If {@code false} is returned from an exit
     *         the protocol bridge agent will not start.
     */
    public boolean initialize(final Map<String, String> bridgeProperties);

    /**
     * Obtains a set of properties for the specified protocol server name.
     * <p>
     * The returned {@link Properties} must contain entries with key names
     * corresponding to the constants defined in
     * {@link ProtocolServerPropertyConstants} and in particular must include an
     * entry for all appropriate constants described as required.
     *
     * @param protocolServerName
     *        The name of the protocol server whose properties are to be
     *        returned. If a null or a blank value is specified, properties
     *        for the default protocol server are to be returned.
     * @return The {@link Properties} for the specified protocol server, or null
     *         if the server cannot be found.
     */
    public Properties getProtocolServerProperties(
        final String protocolServerName);

    /**
     * Invoked once when a protocol bridge agent is shut down. It is intended to
     * release any resources that were allocated by the exit.
     *
     * @param bridgeProperties
     *        The values of properties defined for the protocol bridge.
     *        These values can only be read, they cannot be updated by the
     *        implementation.
     */
    public void shutdown(final Map<String, String> bridgeProperties);
}
```

映射文件服务器的凭证

通过使用协议网桥代理的缺省凭证映射功能或编写您自己的用户出口，将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器的用户凭证。IBM MQ Managed File Transfer 提供了执行用户凭证映射的样本用户出口。

相关概念

[第 275 页的『协议网桥』](#)

通过协议网桥，IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件，而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

相关任务

[第 281 页的『使用 ProtocolBridgeCredentials.xml 文件映射文件服务器的凭证』](#)

通过使用协议网桥代理的缺省凭证映射功能，将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器的用户凭证。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以包含自己的凭证信息。

[第 283 页的『使用出口类映射文件服务器的凭证』](#)

如果您不想使用协议网桥代理的缺省凭证映射功能，那么您可以通过编写您自己的用户出口将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器上的用户凭证中。IBM MQ Managed File Transfer 提供了执行用户凭证映射的样本用户出口。如果您配置凭证映射用户出口，那么它们会取代缺省凭证映射功能。

[第 285 页的『示例：如何配置协议网桥代理以将专用密钥凭证用于 UNIX SFTP 服务器』](#)

该示例演示了如何生成和配置 ProtocolBridgeCredentials.xml 文件。该示例是典型示例，细节会随着您的平台而不同，但原理是一致的。

相关参考

[第 1007 页的『ProtocolBridgeCredentialExit.java 接口』](#)

[第 365 页的『协议网桥凭证用户出口样本』](#)

[第 591 页的『agent.properties 文件』](#)

每个代理都有自己的属性文件 agent.properties，其中必须包含代理用于连接到其队列管理器的信息。agent.properties 文件还可以包含用于更改代理程序行为的属性。

使用 ProtocolBridgeCredentials.xml 文件映射文件服务器的凭证

通过使用协议网桥代理的缺省凭证映射功能，将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器的用户凭证。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以包含自己的凭证信息。

关于此任务

ProtocolBridgeCredentials.xml 文件必须由用户手动创建。缺省情况下，此文件的位置为启动协议网桥代理的用户的主目录，但它可以存储在代理可访问的文件系统上的任何位置。To specify a different location, add the <credentialsFile> element to the ProtocolBridgeProperties.xml file. 例如，

```
<tns:credentialsFile path="/example/path/to/ProtocolBridgeCredentials.xml"/>
```

在可以使用协议网桥代理之前，设置凭证映射，方法是：编辑该文件以包含主机、用户和凭证信息。有关更多信息和样本，请参阅[第 607 页的『协议网桥凭证文件格式』](#)。

使用 IBM MQ Managed File Transfer V7.5 或更低版本在 z/OS 平台上创建

ProtocolBridgeCredentials.xml 文件时，必须先设置文件标记，然后才能编辑该文件。运行以下命令以将文件标记为具有 ASCII 内容：

```
chtag -t -c ISO8859-1 ProtocolBridgeCredentials.xml
```

注：在 z/OS 上，可以将协议网桥凭证文件存储在数据集中，其中 .xml 文件的名称可以由用户指定。

过程

1. 编辑行 `<tns:server name="server name">` 以将 name 属性的值更改为 ProtocolBridgeProperties.xml 文件中的服务器名称。

针对 V7.0.4 和更低版本创建的协议网桥代理没有 ProtocolBridgeProperties.xml 文件（或相关用户出口），因此，对于 V7.0.4.1 和更高版本，会自动为服务器名称指定服务器的主机名。

Therefore, if you use an updated ProtocolBridgeCredentials.xml file with `<server>` entries, a name corresponding to the server's host name will match.

您可以使用 pattern 属性来指定您已使用包含通配符或正则表达式的服务器名称。例如，

```
<tns:server name="serverA*" pattern="wildcard">
```

2. 将用户标识和凭证信息作为 `<tns:server>` 的子元素插入该文件。

您可以将以下某个或一些元素插入文件：

- 如果协议文件服务器为 FTP、FTPS 或 SFTP 服务器，那么您可以使用密码来对请求传输的用户进行认证。将以下行插入文件中：

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  serverPassword="Server Password">
</tns:user>
```

然后更改属性的值。

- name 是一个 Java 正则表达式，用于匹配与 MQMFT 传输请求关联的 MQMD 用户标识
- serverUserId 是作为登录用户标识传递给协议文件服务器的值。如果未指定 serverUserId 属性，那么将改为使用与 MQMFT 传输请求相关联的 MQMD 用户标识
- serverPassword 是与 serverUserId 相关联的密码。

name 属性可以包含 Java 正则表达式。凭证映射器尝试将 MQMFT 传输请求的 MQMD 用户标识与该正则表达式匹配。协议网桥代理尝试将 MQMD 用户标识与 `<tns:user>` 元素的 name 属性中的正则表达式匹配（按照这些元素在文件中的出现顺序）。发现匹配后，协议网桥代理不会再寻找其他匹配。如果发现匹配，那么会将相应的 serverUserId 和 serverPassword 值作为登录用户标识和密码传递到协议文件服务器中。MQMD 用户标识匹配区分大小写。

- 如果协议文件服务器是 SFTP 服务器，那么您可以使用公用和专用密钥来对请求传输的用户进行认证。将以下行插入文件中，并更改属性的值。 `<tns:user>` 元素包含一个或多个 `<tns:privateKey>` 元素。

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  hostKey="Host Key">
  <tns:privateKey associationName="association"
    keyPassword="Private key password">
    Private key file text
  </tns:privateKey>
</tns:user>
```

- name 是一个 Java 正则表达式，用于匹配与 MQMFT 传输请求关联的 MQMD 用户标识
- serverUserId 是作为登录用户标识传递给协议文件服务器的值。如果未指定 serverUserId 属性，那么将改为使用与 MQMFT 传输请求相关联的 MQMD 用户标识
- hostKey 是登录时服务器返回的预期密钥。
- key 是 serverUserId 的专用密钥
- keyPassword 是生成公用密钥的密钥的密码
- associationName 是用于标记以便进行跟踪和记录的值。

name 属性可以包含 Java 正则表达式。凭证映射器尝试将 MQMFT 传输请求的 MQMD 用户标识与该正则表达式匹配。协议网桥代理尝试将 MQMD 用户标识与 `<tns:user>` 元素的 name 属性中的正则表达式匹配（按照这些元素在文件中的出现顺序）。发现匹配后，协议网桥代理不会再寻找其他匹配。如果

找到匹配项，那么将使用相应的 `serverUserId` 和 `key` 值向协议文件服务器认证 MQMFT 用户。MQMD 用户标识匹配区分大小写。

有关使用协议网桥代理的专用密钥的更多信息，请参阅第 285 页的『[示例：如何配置协议网桥代理以将专用密钥凭证用于 UNIX SFTP 服务器](#)』。

注：

将传输请求写入命令队列时，如果源代理命令队列位于 z/OS 或 IBM i 系统上，MQMD 用户标识可能转换为大写。因此，根据传输请求中指定的源代理，相同发起方用户的 MQMD 用户标识以原来的大小写到达凭证出口或转换为大写。缺省凭证映射出口会针对提供的 MQMD 用户标识执行区分大小写的匹配，您可能需要在映射文件中考虑到这一点。

相关参考

第 607 页的『[协议网桥凭证文件格式](#)』

代理配置目录中的 `ProtocolBridgeCredentials.xml` 文件定义用户名和凭证信息，供协议网桥代理用于在协议服务器中为自身授权。

第 611 页的『[协议网桥属性文件格式](#)』

代理配置目录中的 `ProtocolBridgeProperties.xml` 文件定义协议文件服务器的属性。

第 724 页的『[IBM MQ Managed File Transfer 使用的正则表达式](#)』

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 Connect:Direct 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 `java.util.regex` API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

使用出口类映射文件服务器的凭证

如果您不想使用协议网桥代理的缺省凭证映射功能，那么您可以通过编写您自己的用户出口将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器上的用户凭证中。IBM MQ Managed File Transfer 提供了执行用户凭证映射的样本用户出口。如果您配置凭证映射用户出口，那么它们会取代缺省凭证映射功能。

配置协议网桥凭证用户出口

关于此任务

映射协议网桥凭证的用户出口必须实现以下某个接口：

- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`，支持协议网桥代理与一个缺省协议文件服务器相互传输文件
- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2`，支持您与多个端点相互传输文件。

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2` 接口包含与 `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` 相同的功能，并且还包含扩展功能。要了解更多信息，请参阅第 1007 页的『[ProtocolBridgeCredentialExit.java 接口](#)』和第 1008 页的『[ProtocolBridgeCredentialExit2.java 接口](#)』。

凭证出口可以与其他用户出口类似的方式链接在一起。按代理属性文件中使用 `protocolBridgeCredentialConfiguration` 属性指定出口的顺序调用出口。初始化方法均单独返回，并且如果一个或多个方法返回一个 `false` 值，那么代理不会启动。代理事件日志中将报告此错误。

针对所有出口的 `mapMQUserId` 方法，只会返回一个整体结果，如下所示：

- 如果方法返回 `USER_SUCCESSFULLY_MAPPED` 或 `USER_DENIED_ACCESS` 值作为结果代码，那么该值为返回的结果，不会调用后续出口的 `mapMQUserId` 方法。
- 如果方法返回 `NO_MAPPING_FOUND` 值作为结果代码，那么会调用下一个出口的 `mapMQUserId` 方法。
- 如果没有后续出口，那么返回 `NO_MAPPING_FOUND` 结果。
- `USER_DENIED_ACCESS` 或 `NO_MAPPING_FOUND` 的整体结果代码将视为网桥代理传输失败。

要运行出口，请完成以下步骤：

过程

1. 编译协议网桥凭证用户出口。
2. 创建包含已编译出口及其包结构的 Java 归档 (JAR) 文件。
3. 将包含出口类的 JAR 文件放置在网桥代理的 `exits` 目录中。此目录位于 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 目录中。
4. 编辑协议网桥代理的属性文件以包含属性 `protocolBridgeCredentialExitClasses`。对于该属性的值，请指定一个实现协议网桥凭证出口例程的以逗号分隔的类列表。将按照此列表中指定出口类的顺序来调用这些出口类。有关更多信息，请参阅第 591 页的『`agent.properties` 文件』。
5. 将协议网桥代理的属性文件编辑为包含：

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_n
ame\exits\SampleCredentialExit.jar
```

代理的 `agent.properties` 文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中。

如果更改 `agent.properties` 文件，那么必须重新启动代理以获取更改。

6. 您可以选择指定 `protocolBridgeCredentialConfiguration` 属性。针对该属性指定的值将作为字符串对象传入由 `protocolBridgeCredentialExitClasses` 指定的出口类的 `initialize()` 方法。有关更多信息，请参阅第 591 页的『`agent.properties` 文件』。
7. 使用 **`fteStartAgent`** 命令启动协议网桥代理。

使用用户出口样本

关于此任务

`MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` 目录和主题第 365 页的『协议网桥凭证用户出口样本』中提供了样本协议网桥凭证出口。此样本基于 `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` 接口。

`SampleCredentialExit.java` 出口读取将与传输请求相关的 MQMD 用户标识映射到服务器用户标识和服务器密码的属性文件。属性文件的位置取自协议网桥代理属性 `protocolBridgeCredentialConfiguration`。

要运行用户出口样本，请完成以下步骤：

过程

1. 编译 `SampleCredentialExit.java` 文件。
2. 创建一个包含编译的出口及其程序包结构的 JAR 文件。
3. 将该 JAR 文件放在 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/exits` 目录中。
4. 编辑 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` 文件以包含以下行：

```
protocolBridgeCredentialExitClasses=SampleCredentialExit
```

5. 将协议网桥代理的属性文件编辑为包含：

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_n
ame\exits\SampleCredentialExit.jar
```

代理的 `agent.properties` 文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中。

如果更改 `agent.properties` 文件，那么必须重新启动代理以获取更改。

- 在 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` 目录中创建凭证属性文件 (`credentials.properties`)，并对其进行编辑以包含以下格式的条目：

```
mqUserId=serverUserId,serverPassword
```

- 编辑 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` 文件以包含以下行：

```
protocolBridgeCredentialConfiguration=MQ_DATA_PATH/mqft/  
config/coordination_queue_manager/agents/bridge_agent_name/credentials.properties
```

您必须使用至 `credentials.properties` 文件的绝对路径。

- 通过使用 **`fteStartAgent`** 命令来启动协议网桥代理。

相关概念

第 275 页的『协议网桥』

通过协议网桥，IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件，而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

相关参考

第 1007 页的『ProtocolBridgeCredentialExit.java 接口』

第 1008 页的『ProtocolBridgeCredentialExit2.java 接口』

第 365 页的『协议网桥凭证用户出口样本』

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

第 463 页的『fteCreateBridgeAgent（创建和配置 IBM MQ Managed File Transfer 协议网桥代理）』

`fteCreateBridgeAgent` 命令创建协议网桥代理及其关联的配置。针对您要与之进行文件的发送和接收操作的每个文件服务器，创建一个协议网桥代理。

示例：如何配置协议网桥代理以将专用密钥凭证用于 UNIX SFTP 服务器

该示例演示了如何生成和配置 `ProtocolBridgeCredentials.xml` 文件。该示例是典型示例，细节会随着您的平台而不同，但原理是一致的。

关于此任务

过程

- 在 SFTP 客户机上，使用由协议网桥代理传递到 SFTP 服务器的登录标识进行登录，并运行 **`ssh-keygen`** 命令以创建一个公用/专用密钥序列。当要求提供口令时，请提供口令。**`ssh-keygen`** 命令生成以下两个文件：`id_rsa` 和 `id_rsa.pub`。如果您需要 DSA 格式，请在运行 **`ssh-keygen`** 命令时使用 **`-t dsa`**
- 将 `id_rsa.pub` 文件的内容复制到 SFTP 服务器上 SFTP 用户的 `~/.ssh/authorized_keys` 文件中。确保 SFTP 文件服务器进程具有该文件的读访问权。
- 运行以下命令以获取 SFTP 服务器的主机 ssh 指纹：`ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub`
- 在协议网桥代理系统上，编辑 `ProtocolBridgeCredentials.xml` 文件。使用您自己的值来代替下面示例中显示的斜体字值：

```
<tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"  
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials  
ProtocolBridgeCredentials.xsd ">
```

```

<tns:agent name="Agent_name">
  <tns:server name="SFTP_name">
    <tns:user name="mq_User_ID" serverUserId="SFTP_user_ID"
      hostKey="ssh_host_finger">
      <tns:privateKey associationName="name" keyPassword="pass_phrase">
        Complete contents of the id_rsa file including the entries
        -----BEGIN RSA PRIVATE KEY-----
        -----END RSA PRIVATE KEY-----
      </tns:privateKey>
    </tns:user>
  </tns:server>
</tns:agent>
</tns:credentials>

```

其中:

- *Agent_name* 是协议网桥代理的名称。
- *SFTP_host_name* 是 ProtocolBridgeProperties.xml 文件中所示的 SFTP 服务器的名称。
- *mq_User_ID* 是与传输请求关联的 MQMD 用户标识。
- *SFTP_user_ID* 是步骤 2 中使用的 SFTP 用户标识。它是传递到 SFTP 充当登录用户标识的值。
- *ssh_host_finger* 是第 3 步中收集的指纹。
- *name* 是您可以指定用于跟踪和记录目的的名称。
- *pass_phrase* 是您在第 1 步 ssh-keygen 中提供的口令。
- *id_rsa* 文件的完整内容 是从步骤 1 生成的 id_rsa 文件的完整内容。要防止连接错误，请确保同时包含以下两个条目：

```

-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----

```

您可以通过复制 <tns:privatekey> 元素添加其他密钥。

5. 启动协议网桥代理（如果未启动）。或者，协议网桥代理会定期轮询 ProtocolBridgeCredentials.xml 文件并获得更改。

相关参考

第 607 页的『协议网桥凭证文件格式』

代理配置目录中的 ProtocolBridgeCredentials.xml 文件定义用户名和凭证信息，供协议网桥代理用于在协议服务器中为自身授权。

第 463 页的『fteCreateBridgeAgent（创建和配置 IBM MQ Managed File Transfer 协议网桥代理）』

fteCreateBridgeAgent 命令创建协议网桥代理及其关联的配置。针对您要与之进行文件的发送和接收操作的每个文件服务器，创建一个协议网桥代理。

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 **agent.properties**，其中必须包含代理用于连接到其队列管理器的信息。**agent.properties** 文件还可以包含用于更改代理程序行为的属性。

第 281 页的『映射文件服务器的凭证』

通过使用协议网桥代理的缺省凭证映射功能或编写您自己的用户出口，将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器的用户凭证。IBM MQ Managed File Transfer 提供了执行用户凭证映射的样本用户出口。

配置 FTPS 服务器的协议网桥

以配置 FTP 服务器的相似方式配置 FTPS 服务器：创建服务器的网桥代理、定义服务器属性和映射用户凭证。

关于此任务

要配置 FTPS 服务器，请完成下列步骤：

过程

1. 使用 **fteCreateBridgeAgent** 命令为 FTPS 服务器创建协议网桥代理。适用于 FTP 的参数还适用于 FTPS，但还有三个特定于 FTPS 的必需参数：

- a) **-bt** 参数。指定 FTPS 作为该参数的值。
- b) 信任库文件的 **-bts** 参数。该命令假定只有服务器认证是必需的，并且您必须指定信任库文件的位置。

缺省情况下，FTPS 协议的显式格式是由 **fteCreateBridgeAgent** 命令配置的，但您可以通过更改协议网桥属性文件配置隐式格式。协议网桥始终以被动方式连接到 FTPS 服务器。

有关 **fteCreateBridgeAgent** 命令的更多信息，请参阅第 463 页的『[fteCreateBridgeAgent（创建和配置 IBM MQ Managed File Transfer 协议网桥代理）](#)』。

如果需要有关如何创建信任库文件的指示信息，请参阅 IBM Developer 文章“[在 WebSphere MQ File Transfer Edition 中配置安全套接字层连接](#)”，或参阅 [Oracle keytool 文档](#) 中有关 keytool 的信息。

2. Define the FTPS server properties within an <ftpsServer> element in the protocol bridge properties file: `ProtocolBridgeProperties.xml`. 有关更多信息，请参阅第 276 页的『[使用 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性](#)』。您也可以通过编辑协议网桥属性文件启用客户机认证。有关所有配置选项的详细信息，请参阅第 611 页的『[协议网桥属性文件格式](#)』。
3. 通过使用协议网桥代理的缺省凭证映射功能，或通过编写您自己的用户出口，将 IBM MQ Managed File Transfer 中的用户凭证映射到 FTPS 服务器上的用户凭证。有关更多信息，请参阅第 281 页的『[映射文件服务器的凭证](#)』。
4. 缺省情况下，将信任库文件配置为具有 JKS 格式；如果您要更改格式，请编辑协议网桥属性文件。

示例

协议网桥属性文件中 FTPS 服务器的示例条目如下所示：

```
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
    ProtocolBridgeProperties.xsd">
  <tns:defaultServer name="ftpsserver.mycompany.com" />

  <tns:ftpsServer name="ftpsserver.mycompany.com" host="ftpsserver.mycompany.com" port="990"
    platform="windows"
    timeZone="Europe/London" locale="en_US" fileEncoding="UTF8"
    listFormat="unix" limitedWrite="false"
    trustStore="c:\mydirec\truststore.jks" />

  <!-- Define servers here -->
</tns:serverProperties>
```

下一步做什么

有关受支持和不受支持的 FTPS 协议部分的信息，请参阅第 728 页的『[协议网桥提供的 FTPS 服务器支持](#)』。

相关概念

第 275 页的『[协议网桥](#)』

通过协议网桥，IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件，而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

相关任务

第 281 页的『[使用 ProtocolBridgeCredentials.xml 文件映射文件服务器的凭证](#)』

通过使用协议网桥代理的缺省凭证映射功能，将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器的用户凭证。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以包含自己的凭证信息。

第 276 页的『使用 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性』

定义要使用 ProtocolBridgeProperties.xml 文件（由 IBM MQ Managed File Transfer 在代理配置目录中提供）传输文件的一个或多个协议文件服务器的属性。

相关参考

第 463 页的『fteCreateBridgeAgent（创建和配置 IBM MQ Managed File Transfer 协议网桥代理）』

fteCreateBridgeAgent 命令创建协议网桥代理及其关联的配置。针对您要与之进行文件的发送和接收操作的每个文件服务器，创建一个协议网桥代理。

第 607 页的『协议网桥凭证文件格式』

代理配置目录中的 ProtocolBridgeCredentials.xml 文件定义用户名和凭证信息，供协议网桥代理用于在协议服务器中为自身授权。

第 611 页的『协议网桥属性文件格式』

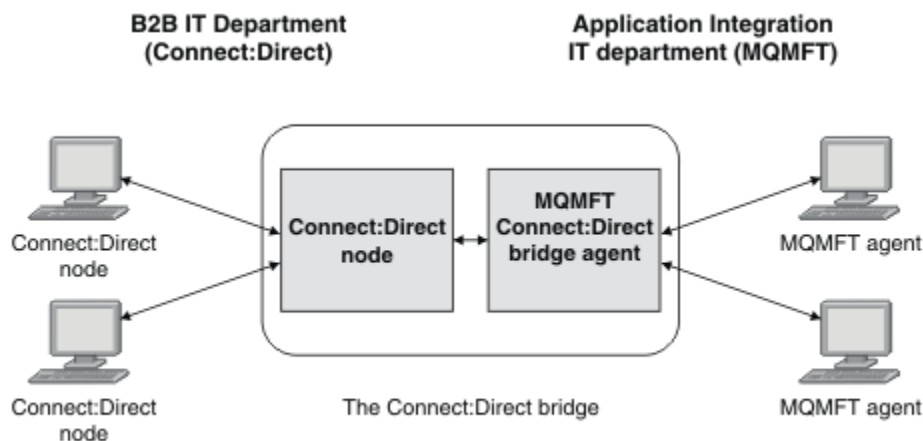
代理配置目录中的 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性。

第 728 页的『协议网桥提供的 FTPS 服务器支持』

协议网桥支持 FTPS 协议的子集，如 RFC-2228、RFC-4217 和名为 *Secure FTP over SSL* 的因特网草案所定义。

Connect:Direct 网桥

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。



该图显示了两个部门（B2B IT 部门和 Application Integration IT 部门）之间的 MQMFT Connect:Direct 网桥。B2B IT 部门使用 Connect:Direct 与公司的业务合作伙伴之间传输文件。应用程序集成 IT 部门使用 WebSphere MQ 作为其消息传递基础架构，因此选择了 IBM MQ Managed File Transfer 作为其文件传输解决方案。

通过使用 MQMFT Connect:Direct 网桥，这两个部门可以在 B2B IT 部门中的 Connect:Direct 网络 and 应用程序集成 IT 部门中的 MQMFT 网络之间传输文件。Connect:Direct 网桥是 IBM MQ Managed File Transfer 的一个组件，它包含一个 MQMFT 代理，用于和 Connect:Direct 节点通信。MQMFT 代理称为 Connect:Direct 网桥代理，专用于和 Connect:Direct 节点相互传输文件。

Connect:Direct 网桥作为 IBM MQ Managed File Transfer 的服务和代理程序组件的一部分提供，可用于以下任务：

1. 使用 受管文件传输 命令启动从 MQMFT 代理到 Connect:Direct 节点的一个或多个文件的传输。
2. 使用 受管文件传输 命令启动从 Connect:Direct 节点到 MQMFT 代理的一个或多个文件的传输。
3. 使用 受管文件传输 命令启动某个文件传输，该文件传输会启动用户定义的 Connect:Direct 进程。
4. 使用 Connect:Direct 进程，提交 MQMFT 文件传输请求。

Connect:Direct 网桥只可以与 Connect:Direct 节点之间传输文件。Connect:Direct 网桥只能在 Connect:Direct 进程提交的传输过程中将文件传输到其本地文件系统或从其本地文件系统传输文件。

您可以使用 Connect:Direct 网桥与位于 z/OS 系统中的 Connect:Direct 节点上的数据集之间进行传输。该传输与只涉及 IBM MQ Managed File Transfer 代理的数据集传输行为相比有所不同。有关更多信息，请参阅第 706 页的『与 Connect:Direct 节点之间传输数据集』。

支持的平台

Connect:Direct 网桥由一个 MQMFT Connect:Direct 网桥代理和一个 Connect:Direct 节点组成。对于 System x，代理程序在 Windows 和 Linux 上受支持。该节点在 IBM Sterling Connect:Direct for Windows 和 IBM Sterling Connect:Direct for UNIX 支持的平台上受支持。有关创建 Connect:Direct 网桥代理和配置 Connect:Direct 节点以与代理进行通信的指示信息，请参阅第 204 页的『配置 Connect:Direct 网桥』。

Connect:Direct 网桥可以将文件传输到 Connect:Direct 节点，也可以从其传输文件，这些节点作为 Connect:Direct for Windows、Connect:Direct for UNIX 或 Connect:Direct for z/OS 服务 安装的一部分运行。要了解有关受支持的 Connect:Direct 版本的详细信息，请参阅 Web 页面 [WebSphere MQ 系统需求](#)。

组成 Connect:Direct 网桥的代理和节点必须在同一个系统上，或具有同一文件系统的访问权，例如，通过一个共享的 NFS 安装。该文件系统用于在涉及 Connect:Direct 网桥的文件传输期间在 **cdTmpDir** 参数定义的目录中临时存储文件。Connect:Direct 网桥代理和 Connect:Direct 网桥节点必须能够使用同一路径名称来对该目录进行寻址。例如，如果代理和节点在不同的 Windows 系统上，那么这些系统必须使用相同的盘符来安装共享文件系统。以下配置允许代理和节点使用同一路径名：

- 代理程序和节点位于同一系统上，该系统正在运行 Windows 或 Linux for System x
- 代理程序位于 Linux for System x 上，节点位于 UNIX 上
- 代理在一个 Windows 系统上，节点在另一个 Windows 系统上

以下配置不允许代理和节点使用同一路径名：

- 代理程序位于 Linux for System x 上，节点位于 Windows 上
- 代理在 Windows 上，节点在 UNIX 上

请在计划安装 Connect:Direct 网桥时考虑该限制。

相关概念

第 296 页的『恢复并重新启动与 Connect:Direct 节点之间的传输』

在传输期间，IBM MQ Managed File Transfer 可能无法连接到您的 IBM Sterling Connect:Direct 节点；例如，当该节点不可用时。IBM MQ Managed File Transfer 会尝试恢复传输，或者传输失败并生成错误消息。

第 297 页的『通过文件传输请求提交用户定义的 Connect:Direct 进程』

对于通过 Connect:Direct 网桥代理以在文件传输过程中调用用户定义的 Connect:Direct 进程的传输，您可以提交一个传输请求。

第 300 页的『使用 Connect:Direct 进程来提交 IBM MQ Managed File Transfer 传输请求』

您可以通过 Connect:Direct 进程向 Connect:Direct 网桥代理提交传输请求。IBM MQ Managed File Transfer 提供了可以从 Connect:Direct 进程中的 **RUN TASK** 语句调用的命令。

第 423 页的『对 Connect:Direct 网桥进行故障诊断』

使用以下参考信息和示例，以帮助诊断 Connect:Direct 网桥返回的错误。

相关任务

第 204 页的『配置 Connect:Direct 网桥』

配置 Connect:Direct 网桥，以在 IBM MQ Managed File Transfer 网络和 Connect:Direct 网络之间传输文件。Connect:Direct 网桥的组件包括 Connect:Direct 节点和专用于与该节点通信的 IBM MQ Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

第 290 页的『向 Connect:Direct 节点传输文件』

您可以使用 Connect:Direct 网桥从 IBM MQ Managed File Transfer 代理向 Connect:Direct 节点传输文件。通过将 Connect:Direct 网桥代理指定为目标代理并以 `connect_direct_node_name:file_path` 格式指定目标文件，指定 Connect:Direct 节点作为传输的目标。

第 291 页的『从 Connect:Direct 节点传输文件』

您可以使用 Connect:Direct 网桥，将文件从 Connect:Direct 节点传输到 IBM MQ Managed File Transfer 代理。通过将 Connect:Direct 网桥代理指定为源代理并以 `connect_direct_node_name:file_path` 格式指定源规范，可以将 Connect:Direct 节点指定为传输源。

[第 292 页的『向 Connect:Direct 节点传输多个文件』](#)

您可以使用 Connect:Direct 网桥，将多个文件从 IBM MQ Managed File Transfer 代理传输到 Connect:Direct 节点。要使用 Connect:Direct 节点作为多文件传输的目标，请指定 Connect:Direct 网桥代理作为目标代理，并以 `connect_direct_node_name:directory_path` 格式指定目标目录。

[第 294 页的『从 Connect:Direct 节点传输多个文件』](#)

您可以使用 Connect:Direct 网桥，将多个文件从 Connect:Direct 节点传输到 IBM MQ Managed File Transfer 代理。通过将 Connect:Direct 网桥代理指定为源代理并以 `connect_direct_node_name:file_path` 格式指定一个或多个源规范，可以将 Connect:Direct 节点指定为多文件传输的源。

[第 294 页的『通过使用通配符向 Connect:Direct 传输多个文件』](#)

要将多个文件从 IBM MQ Managed File Transfer 代理传输到 Connect:Direct 节点，可使用 Connect:Direct 网桥。您可以在提供给 **fteCreateTransfer** 命令的源规范中使用通配符。与所有涉及通配符的 IBM MQ Managed File Transfer 传输一样，仅文件路径的最后一部分可包含通配符。例如，`/abc/def*` 是有效文件路径，而 `/abc*/def` 是无效文件路径。

相关参考

[第 469 页的『fteCreateCDAgent（创建 Connect:Direct 网桥代理）』](#)

fteCreateCDAgent 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

[第 728 页的『Connect:Direct 网桥代理的限制』](#)

Connect:Direct 网桥代理配置为用于与 Connect:Direct 节点之间传输文件。Connect:Direct 网桥代理无法执行一些功能。

向 Connect:Direct 节点传输文件

您可以使用 Connect:Direct 网桥从 IBM MQ Managed File Transfer 代理向 Connect:Direct 节点传输文件。通过将 Connect:Direct 网桥代理指定为目标代理并以 `connect_direct_node_name:file_path` 格式指定目标文件，指定 Connect:Direct 节点作为传输的目标。

开始之前

在传输文件之前，必须配置作为 IBM MQ Managed File Transfer 组成部分的 Connect:Direct 网桥。有关更多信息，请参阅 [第 204 页的『配置 Connect:Direct 网桥』](#)。

关于此任务

在该示例中，Connect:Direct 网桥代理名为 CD_BRIDGE。源代理名为 FTE_AGENT，可以是任意 WMQFTE 版本。目标 Connect:Direct 节点名为 CD_NODE1。要传输的文件位于 FTE_AGENT 所在的系统上的 `/home/helen/file.log` 文件路径中。该文件将传输到运行 CD_NODE1 的系统上的 `/files/data.log` 文件路径中。

过程

1. 将 **fteCreateTransfer** 命令与 **-df** (目标文件) 参数的值配合使用，格式为 `connect_direct_node_name:file_path`，并将 **-da** (目标代理) 参数的值指定为 Connect:Direct 网桥代理的名称。

注: `connect_direct_node_name` 指定的 Connect:Direct 节点是要将文件传输到的节点，而不是作为 Connect:Direct 网桥一部分运行的 Connect:Direct 节点。

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                 -df CD_NODE1:/files/data.log /home/helen/file.log
```

有关更多信息，请参阅 [第 497 页的『fteCreateTransfer（创建新的文件传输）』](#)。

2. 源代理 FTE_AGENT 将文件传输到 Connect:Direct 网桥代理 CD_BRIDGE。该文件临时存储在运行 Connect:Direct 网桥代理的系统上，位置由 `cdTmpDir` 代理属性定义。Connect:Direct 网桥代理将文件传输到 Connect:Direct 节点 CD_NODE1。

相关概念

[第 288 页的『Connect:Direct 网桥』](#)

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关任务

第 291 页的『从 Connect:Direct 节点传输文件』

您可以使用 Connect:Direct 网桥，将文件从 Connect:Direct 节点传输到 IBM MQ Managed File Transfer 代理。通过将 Connect:Direct 网桥代理指定为源代理并以 `connect_direct_node_name:file_path` 格式指定源规范，可以将 Connect:Direct 节点指定为传输源。

相关参考

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

从 Connect:Direct 节点传输文件

您可以使用 Connect:Direct 网桥，将文件从 Connect:Direct 节点传输到 IBM MQ Managed File Transfer 代理。通过将 Connect:Direct 网桥代理指定为源代理并以 `connect_direct_node_name:file_path` 格式指定源规范，可以将 Connect:Direct 节点指定为传输源。

开始之前

在传输文件之前，必须配置作为 IBM MQ Managed File Transfer 组成部分的 Connect:Direct 网桥。请参阅第 204 页的『配置 Connect:Direct 网桥』。

关于此任务

在该示例中，Connect:Direct 网桥代理名为 CD_BRIDGE。目标代理名为 FTE_AGENT，可以是任意 WMQFTE 版本。源 Connect:Direct 节点名为 CD_NODE1。要传输的文件位于 CD_NODE1 所在的系统上的 `/home/brian/in.file` 文件路径中。该文件将传输到运行 FTE_AGENT 的系统上的 `/files/out.file` 文件路径中。

过程

使用 `fteCreateTransfer` 命令，按如下格式指定源规范的值：

`connect_direct_node_name:file_path`，并且 `-sa` 参数的值指定为 Connect:Direct 网桥代理的名称。

注：`connect_direct_node_name` 指定的 Connect:Direct 节点是要从中传输文件的节点，而不是作为 Connect:Direct 网桥的一部分运行的 Connect:Direct 节点。例如：

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_AGENT
                  -df /files/out.file CD_NODE1:/home/brian/in.file
```

有关更多信息，请参阅第 497 页的『fteCreateTransfer（创建新的文件传输）』。

结果

Connect:Direct 网桥代理 CD_BRIDGE 请求来自 Connect:Direct 节点 CD_NODE1 的文件。Connect:Direct 节点向 Connect:Direct 网桥发送文件。从 Connect:Direct 节点传输文件时，Connect:Direct 网桥会将该文件临时存储在 `cdTmpDir` 代理属性定义的位置中。当从 Connect:Direct 节点到 Connect:Direct 网桥的文件传输完成时，Connect:Direct 网桥会将该文件发送给目标代理 FTE_AGENT 并从临时位置中删除该文件。

相关概念

第 288 页的『Connect:Direct 网桥』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关参考

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

向 z/OS 上的 Connect:Direct 节点传输数据集

您可以使用位于 Windows 或 Linux 系统上的 Connect:Direct 网桥将数据集从 z/OS 上的 IBM MQ Managed File Transfer 代理传输到 z/OS 上的 Connect:Direct 节点。

开始之前

在传输文件之前，必须配置作为 IBM MQ Managed File Transfer 组成部分的 Connect:Direct 网桥。请参阅第 204 页的『配置 Connect:Direct 网桥』。

关于此任务

在该示例中，参数 **-df** 用于指定传输的目标。当传输的源代理是任何版本的 IBM MQ Managed File Transfer 时，参数 **-df** 有效并可供使用。如果源代理是 V7.0.4 或更高版本，那么可以改为使用 **-ds** 参数。源代理名为 FTE_ZOS1，并且是 V7.0.3 代理。Connect:Direct 网桥代理名为 CD_BRIDGE，位于 Linux 系统上。目标 Connect:Direct 节点名为 CD_ZOS2。源代理和目标 Connect:Direct 节点都位于 z/OS 系统上。要传输的数据集位于 FTE_ZOS1 所在的系统上的 //FTEUSER.SOURCE.LIB 中。该数据集将传输到 CD_ZOS2 所在的系统上的 //CDUSER.DEST.LIB 数据集中。

过程

1. 将 fteCreateTransfer 命令与 **-df** 参数的值一起使用，格式为：
`connect_direct_node_name:data_set_name;attributes`，并将 **-da** (目标代理) 参数的值指定为 Connect:Direct 网桥代理的名称。

`connect_direct_node_name` 指定的 Connect:Direct 节点是要将数据集传输到的节点，而不是作为 Connect:Direct 网桥一部分运行的 Connect:Direct 节点。

`data_set_name` 指定的数据集名称必须是绝对名称而不是相对名称。Connect:Direct 不向数据集名称前添加用户名以作为前缀。

```
fteCreateTransfer -sa FTE_ZOS1 -sm QM_ZOS
                  -da CD_BRIDGE -dm QM_BRIDGE
                  -df CD_ZOS2:/'CDUSER.DEST.LIB;BLKSIZE(8000);LRECL(80)'
                  //'FTEUSER.SOURCE.LIB'
```

有关更多信息，请参阅第 497 页的『fteCreateTransfer (创建新的文件传输)』。

2. 源代理 FTE_ZOS1 将数据集中的数据传输给 Connect:Direct 网桥代理 CD_BRIDGE。这些数据临时作为平面文件存储在运行 Connect:Direct 网桥代理的系统上，位置由 cdTmpDir 代理属性定义。Connect:Direct 网桥代理将数据传输到 Connect:Direct 节点 CD_ZOS2。当传输完成时，将从运行 Connect:Direct 网桥代理的系统中删除该平面文件。

相关概念

第 288 页的『Connect:Direct 网桥』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关参考

第 706 页的『与 Connect:Direct 节点之间传输数据集』

可以使用 Connect:Direct 网桥在 IBM MQ Managed File Transfer 代理和 IBM Sterling Connect:Direct 节点之间传输数据集。您可以将数据集指定为传输源和/或传输目标。

第 708 页的『Connect:Direct 进程语句参数和 BPXWDYN 关键字之间的映射』

对于源或目标是 Connect:Direct 节点的数据集，当提交传输请求时，您提供的任何受支持的 BPXWDYN 关键字都将转换为 Connect:Direct 进程接受的格式。

向 Connect:Direct 节点传输多个文件

您可以使用 Connect:Direct 网桥，将多个文件从 IBM MQ Managed File Transfer 代理传输到 Connect:Direct 节点。要使用 Connect:Direct 节点作为多文件传输的目标，请指定 Connect:Direct 网桥代理作为目标代理，并以 `connect_direct_node_name:directory_path` 格式指定目标目录。

开始之前

在传输文件之前，必须配置作为 IBM MQ Managed File Transfer 组成部分的 Connect:Direct 网桥。请参阅第 204 页的『配置 Connect:Direct 网桥』。

关于此任务

在该示例中，源代理名为 FTE_AGENT。Connect:Direct 网桥代理名为 CD_BRIDGE。目标 Connect:Direct 节点名为 CD_NODE1。要传输的文件是 FTE_AGENT 所在系统上的 /home/jack/data.log、/logs/log1.txt 和 /results/latest。文件将传输到运行 CD_NODE1 的系统上的 /in/files 目录中。

过程

使用 fteCreateTransfer 命令，按如下格式指定 **-dd**（目标目录）参数的值：

`connect_direct_node_name:directory_path`。指定 **-da**（目标代理）参数的值作为 Connect:Direct 网桥代理的名称。

注：`connect_direct_node_name` 指定的 Connect:Direct 节点是要将文件传输到的节点，而不是作为 Connect:Direct 网桥一部分运行的 Connect:Direct 节点。

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/in/files /home/jack/data.log
                  /logs/log1.txt /results/latest
```

有关更多信息，请参阅第 497 页的『fteCreateTransfer（创建新的文件传输）』。

结果

源代理 FTE_AGENT 将第一个文件传输到 Connect:Direct 网桥代理 CD_BRIDGE。Connect:Direct 网桥代理将该文件临时存储在 cdTmpDir 属性定义的位置中。当文件完全从源代理传输到 Connect:Direct 网桥时，Connect:Direct 网桥代理会将该文件发送到 cdNode 代理属性定义的 Connect:Direct 节点。该节点会将该文件发送到目标 Connect:Direct 节点 CD_NODE1。当两个 Connect:Direct 节点之间的传输完成时，Connect:Direct 网桥代理会将该文件从临时位置中删除。将针对每个指定的源文件执行该过程。

相关概念

第 288 页的『Connect:Direct 网桥』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关任务

第 290 页的『向 Connect:Direct 节点传输文件』

您可以使用 Connect:Direct 网桥从 IBM MQ Managed File Transfer 代理向 Connect:Direct 节点传输文件。通过将 Connect:Direct 网桥代理指定为目标代理并以 `connect_direct_node_name:file_path` 格式指定目标文件，指定 Connect:Direct 节点作为传输的目标。

第 294 页的『通过使用通配符向 Connect:Direct 传输多个文件』

要将多个文件从 IBM MQ Managed File Transfer 代理传输到 Connect:Direct 节点，可使用 Connect:Direct 网桥。您可以在提供给 **fteCreateTransfer** 命令的源规范中使用通配符。与所有涉及通配符的 IBM MQ Managed File Transfer 传输一样，仅文件路径的最后一部分可包含通配符。例如，/abc/def* 是有效文件路径，而 /abc*/def 是无效文件路径。

第 291 页的『从 Connect:Direct 节点传输文件』

您可以使用 Connect:Direct 网桥，将文件从 Connect:Direct 节点传输到 IBM MQ Managed File Transfer 代理。通过将 Connect:Direct 网桥代理指定为源代理并以 `connect_direct_node_name:file_path` 格式指定源规范，可以将 Connect:Direct 节点指定为传输源。

第 294 页的『从 Connect:Direct 节点传输多个文件』

您可以使用 Connect:Direct 网桥，将多个文件从 Connect:Direct 节点传输到 IBM MQ Managed File Transfer 代理。通过将 Connect:Direct 网桥代理指定为源代理并以 `connect_direct_node_name:file_path` 格式指定一个或多个源规范，可以将 Connect:Direct 节点指定为多文件传输的源。

相关参考

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

从 Connect:Direct 节点传输多个文件

您可以使用 Connect:Direct 网桥，将多个文件从 Connect:Direct 节点传输到 IBM MQ Managed File Transfer 代理。通过将 Connect:Direct 网桥代理指定为源代理并以 `connect_direct_node_name:file_path` 格式指定一个或多个源规范，可以将 Connect:Direct 节点指定为多文件传输的源。

开始之前

在传输文件之前，必须配置作为 IBM MQ Managed File Transfer 组成部分的 Connect:Direct 网桥。请参阅第 204 页的『配置 Connect:Direct 网桥』。

关于此任务

在该示例中，Connect:Direct 网桥代理名为 `CD_BRIDGE`。目标代理名为 `FTE_Z`，并且运行在 `z/OS` 系统上。源 Connect:Direct 节点名为 `CD_NODE1`。要传输的文件位于 `CD_NODE1` 所在的系统上的文件路径 `/in/file1`、`/in/file2` 和 `/in/file3` 中。文件将传输到运行 `FTE_Z` 的系统上的分区数据集 `//OBJECT.LIB` 中。

过程

使用 `fteCreateTransfer` 命令，按如下格式指定源规范的值：

`connect_direct_node_name:file_path`，并且 `-sa` 参数的值指定为 Connect:Direct 网桥代理的名称。

注：`connect_direct_node_name` 指定的 Connect:Direct 节点是要从中传输文件的节点，而不是作为 Connect:Direct 网桥一部分运行的 Connect:Direct 节点。

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_Z
                  -dp '//OBJECT.LIB' CD_NODE1:/in/file1
                  CD_NODE1:/in/file2 CD_NODE1:/in/file3
```

有关更多信息，请参阅第 497 页的『`fteCreateTransfer`（创建新的文件传输）』。

结果

Connect:Direct 网桥代理 `CD_BRIDGE` 请求 Connect:Direct 节点 `CD_NODE1` 中的第一个文件。Connect:Direct 节点向 Connect:Direct 网桥发送文件。从 Connect:Direct 节点传输文件时，Connect:Direct 网桥会将该文件临时存储在 `cdTmpDir` 代理属性定义的位置中。当从 Connect:Direct 节点到 Connect:Direct 网桥的文件传输完成时，Connect:Direct 网桥会将该文件发送到目标代理 `FTE_Z`，之后从临时位置中删除该文件。将针对每个指定的源文件执行该过程。

相关概念

第 288 页的『Connect:Direct 网桥』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关参考

第 591 页的『`agent.properties` 文件』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

通过使用通配符向 Connect:Direct 传输多个文件

要将多个文件从 IBM MQ Managed File Transfer 代理传输到 Connect:Direct 节点，可使用 Connect:Direct 网桥。您可以在提供给 `fteCreateTransfer` 命令的源规范中使用通配符。与所有涉及通配符的 IBM MQ Managed File Transfer 传输一样，仅文件路径的最后一部分可包含通配符。例如，`/abc/def*` 是有效文件路径，而 `/abc*/def` 是无效文件路径。

开始之前

在传输文件之前，必须配置作为 IBM MQ Managed File Transfer 组成部分的 Connect:Direct 网桥。有关更多信息，请参阅 [第 204 页的『配置 Connect:Direct 网桥』](#)。

关于此任务

在该示例中，源代理名为 FTE_AGENT，Connect:Direct 网桥代理名为 CD_BRIDGE。目标 Connect:Direct 节点名为 CD_NODE1。要传输的文件位于 FTE_AGENT 所在的系统上的 /reports 目录中。仅传输名称以 report 开头、后跟两个字符和后缀 .log 的文件。例如，将传输文件 /reports/report01.log，但不会传输文件 /reports/report1.log。文件将传输到运行 CD_NODE1 的系统上的 /home/fred 目录中。

过程

1. 使用 fteCreateTransfer 命令，按如下格式指定 **-dd**（目标目录）参数的值：
`connect_direct_node_name:directory_path`。对于 **-da**（目标代理）参数，指定 Connect:Direct 网桥代理。

注：`connect_direct_node_name` 指定的 Connect:Direct 节点是要将文件传输到的节点，而不是作为 Connect:Direct 网桥一部分运行的 Connect:Direct 节点。

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE  
                  -dd CD_NODE1:/home/fred "/reports/report??.log"
```

有关更多信息，请参阅 [第 497 页的『fteCreateTransfer（创建新的文件传输）』](#)。

2. 源代理 FTE_AGENT 将第一个与 /reports/report??.log 模式匹配的文件传输到 Connect:Direct 网桥代理 CD_BRIDGE。Connect:Direct 网桥代理将该文件临时存储在 cdTmpDir 属性定义的位置中。当文件完全从源代理传输到 Connect:Direct 网桥时，Connect:Direct 网桥代理会将该文件发送到 cdNode 代理属性定义的 Connect:Direct 节点。该节点会将该文件发送到目标 Connect:Direct 节点 CD_NODE1。当两个 Connect:Direct 节点之间的传输完成时，Connect:Direct 网桥代理会将该文件从临时位置中删除。针对与通配符模式匹配的每个源文件重复此过程 /reports/report??.log。

注：根据源代理 FTE_AGENT 所在的系统的操作系统，与 /reports/report??.log 模式匹配的文件列表可能有所不同。

- 如果源代理位于运行 Windows 操作系统的系统上，那么模式匹配将不区分大小写。该模式与 /reports 目录中文件名格式为 report 后跟两个字符和后缀 .log 的所有文件都匹配，而与字母的大小写无关。例如，Report99.Log 为匹配项。
- 如果源代理程序位于具有 Linux 或 UNIX 操作系统的系统上，那么模式匹配区分大小写。该模式仅与 /reports 目录中文件名格式为 report 后跟两个字符和后缀 .log 的文件匹配。例如，reportAB.log 为匹配项，而 reportAB.LOG 和 Report99.Log 为非匹配项。

相关概念

[第 288 页的『Connect:Direct 网桥』](#)

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关任务

[第 290 页的『向 Connect:Direct 节点传输文件』](#)

您可以使用 Connect:Direct 网桥从 IBM MQ Managed File Transfer 代理向 Connect:Direct 节点传输文件。通过将 Connect:Direct 网桥代理指定为目标代理并以 `connect_direct_node_name:file_path` 格式指定目标文件，指定 Connect:Direct 节点作为传输的目标。

[第 292 页的『向 Connect:Direct 节点传输多个文件』](#)

您可以使用 Connect:Direct 网桥，将多个文件从 IBM MQ Managed File Transfer 代理传输到 Connect:Direct 节点。要使用 Connect:Direct 节点作为多文件传输的目标，请指定 Connect:Direct 网桥代理作为目标代理，并以 `connect_direct_node_name:directory_path` 格式指定目标目录。

[第 294 页的『从 Connect:Direct 节点传输多个文件』](#)

您可以使用 Connect:Direct 网桥，将多个文件从 Connect:Direct 节点传输到 IBM MQ Managed File Transfer 代理。通过将 Connect:Direct 网桥代理指定为源代理并以 `connect_direct_node_name:file_path` 格式指定一个或多个源规范，可以将 Connect:Direct 节点指定为多文件传输的源。

相关参考

第 591 页的『[agent.properties 文件](#)』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

第 721 页的『[使用通配符](#)』

为文件传输指定源文件名和源文件路径时，可以使用通配符。这使您能够同时选择多个文件。

恢复并重新启动与 Connect:Direct 节点之间的传输

在传输期间，IBM MQ Managed File Transfer 可能无法连接到您的 IBM Sterling Connect:Direct 节点；例如，当该节点不可用时。IBM MQ Managed File Transfer 会尝试恢复传输，或者传输失败并生成错误消息。

如果 Connect:Direct 节点不可用

如果 Connect:Direct 节点不可用；例如，由于网络或电源中断而导致不可用，那么 IBM MQ Managed File Transfer 将通过以下方式恢复文件传输：

- 如果 IBM MQ Managed File Transfer 先前未作为此传输请求的一部分成功连接到 Connect:Direct 节点，那么将在由 **cdMaxConnectionRetries** 和 **recoverableTransferRetryInterval properties** 的值确定的时间长度内重试传输。这些属性是在 Connect:Direct 网桥代理的 `agent.properties` 文件中指定的。传输失败，并且在尝试失败次数达到 **cdMaxConnectionRetries property** 的值后生成错误消息。缺省情况下，传输将每隔 60 秒无限期地尝试。
- 如果 IBM MQ Managed File Transfer 先前已作为此传输请求的一部分成功连接到 Connect:Direct 节点，那么将在 **cdMaxPartialWorkConnectionRetries** 和 **recoverableTransferRetryInterval** 属性的值所确定的时间长度内重试传输。在失败尝试次数达到 **cdMaxPartialWorkConnectionRetries** 属性的值后，传输失败并生成错误消息。缺省情况下，传输将每隔 60 秒无限期地尝试。
- 对于某些类型的 Connect:Direct 节点故障（例如，正在强制停止的节点），Connect:Direct 进程将在节点恢复时进入 **Held Due to Error (HE)** 状态。在节点恢复后，IBM MQ Managed File Transfer 将自动恢复与文件传输相关并且状态为 HE 的所有 Connect:Direct 进程。
- 如果该传输失败，那么将从托管 Connect:Direct 网桥的系统中删除与该传输相关的所有临时文件。这些临时文件的位置由 **cdTmpDir** 属性定义。
- 如果是从 IBM MQ Managed File Transfer 到 Connect:Direct 的传输，并且指定了删除源处置，那么如果传输失败，则不会删除源文件。

如果 Connect:Direct 节点用户凭证无效

如果由于 Connect:Direct 节点拒绝用户的凭证而使 IBM MQ Managed File Transfer 无法连接到该节点，那么该传输将失败并且生成错误消息。在这种情况下，请检查是否为 Connect:Direct 节点提供了正确的用户凭证。有关更多信息，请参阅第 207 页的『[映射 Connect:Direct 的凭证](#)』。

如果 Connect:Direct 网桥代理不可用

如果 Connect:Direct 网桥代理不可用，那么所有正在进行的文件传输将以与标准 IBM MQ Managed File Transfer 传输相同的方式恢复。有关更多信息，请参阅第 302 页的『[IBM MQ Managed File Transfer 的恢复和重新启动](#)』。

相关概念

第 288 页的『[Connect:Direct 网桥](#)』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

第 302 页的『[IBM MQ Managed File Transfer 的恢复和重新启动](#)』

如果由于任何原因（例如，由于电源故障或网络故障）您的代理或队列管理器不可用，那么 IBM MQ Managed File Transfer 将在这些场景下按如下方式恢复：

相关任务

第 204 页的『[配置 Connect:Direct 网桥](#)』

配置 Connect:Direct 网桥，以在 IBM MQ Managed File Transfer 网络和 Connect:Direct 网络之间传输文件。Connect:Direct 网桥的组件包括 Connect:Direct 节点和专用于与该节点通信的 IBM MQ Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

相关参考

第 591 页的『[agent.properties 文件](#)』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

通过文件传输请求提交用户定义的 Connect:Direct 进程

对于通过 Connect:Direct 网桥代理以在文件传输过程中调用用户定义的 Connect:Direct 进程的传输，您可以提交一个传输请求。

缺省情况下，当针对通过 Connect:Direct 网桥的传输提交文件传输请求时，Connect:Direct 网桥代理会生成用于与远程 Connect:Direct 节点之间传输文件的 Connect:Direct 进程。

但是，您可以配置 Connect:Direct 网桥代理，以改为使用 `ConnectDirectProcessDefinition.xml` 文件调用用户定义的 Connect:Direct 进程。

ConnectDirectProcessDefinition.xml 文件

fteCreateCDAgent 命令在代理配置目录 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` 中创建文件 `ConnectDirectProcessDefinitions.xml`。要从 Connect:Direct 网桥代理中调用用户定义的 Connect:Direct 进程，必须首先通过编辑该文件来设置进程定义。

该文件定义一个或多个进程集，其中包含传输过程中调用的一个或多个 Connect:Direct 进程的位置。每个进程集都包含大量的条件。如果传输满足进程集的所有条件，那么该进程集将用于指定由传输调用的 Connect:Direct 进程。有关更多信息，请参阅第 212 页的『[使用 ConnectDirectProcessDefinition.xml 文件指定要启动的 Connect:Direct 进程](#)』。

内置符号变量

您可以使用 IBM MQ Managed File Transfer 定义的内置符号变量，将值代入用户定义的 Connect:Direct 进程中。要遵循 Connect:Direct 命名约定，IBM MQ Managed File Transfer 使用的所有内部符号变量的格式均为 `%FTE`，后跟五个大写字母数字字符。

在创建进程以将文件从 Connect:Direct 节点传输到 Connect:Direct 网桥系统时，必须使用内置变量 `%FTETFILE` 作为 Connect:Direct 进程中 TO FILE 的值。当创建进程以向 Connect:Direct 节点传输来自 Connect:Direct 网桥系统的文件时，必须使用内置变量 `%FTEFFILE` 作为 Connect:Direct 进程中 FROM FILE 的值。这些变量包含 Connect:Direct 网桥代理用于与 IBM MQ Managed File Transfer 网络之间传输的临时文件路径。

有关内置符号变量的更多信息，请参阅 Connect:Direct 产品文档。

样本 Connect:Direct 进程

WebSphere MQ File Transfer Edition 提供样本 Connect:Direct 进程。这些样本位于以下目录中：
`MQ_INSTALLATION_PATH/mqft/samples/ConnectDirectProcessTemplates`。

相关任务

第 212 页的『[使用 ConnectDirectProcessDefinition.xml 文件指定要启动的 Connect:Direct 进程](#)』

指定要在 IBM MQ Managed File Transfer 传输过程中启动的 Connect:Direct 进程。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以指定进程定义。

第 299 页的『[在由 IBM MQ Managed File Transfer 调用的 Connect:Direct 进程中使用内置符号变量](#)』

您可以在 IBM MQ Managed File Transfer 传输中调用用户定义的 Connect:Direct 进程，并通过在进程定义中使用内置符号变量来将传输中的信息传递给 Connect:Direct 进程。

相关参考

第 623 页的『Connect:Direct 进程定义文件格式』

Connect:Direct 网桥代理配置目录中的 `ConnectDirectProcessDefinitions.xml` 文件指定要在文件传输过程中启动的用户定义的 Connect:Direct 进程。

第 724 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』

您可以通过使用特定于 IBM MQ Managed File Transfer 的内置符号变量来定义代入用户定义的 Connect:Direct 进程的值。

使用 `ConnectDirectProcessDefinition.xml` 文件指定要启动的 `Connect:Direct` 进程

指定要在 IBM MQ Managed File Transfer 传输过程中启动的 Connect:Direct 进程。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以指定进程定义。

关于此任务

fteCreateCDAgent 命令在代理配置目录 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` 中创建文件 `ConnectDirectProcessDefinitions.xml`。要从 Connect:Direct 网桥代理中调用用户定义的 Connect:Direct 进程，必须首先通过编辑该文件来设置进程定义。

对于要指定在传输过程中通过 Connect:Direct 网桥调用的每个进程，请执行以下步骤：

过程

1. 定义希望 Connect:Direct 网桥代理在传输过程中调用的 Connect:Direct 进程，并将进程模板保存在文件中。
2. 在文本编辑器中打开 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml` 文件。
3. 创建 `<processSet>` 元素。
4. 在 `<processSet>` 元素中，创建 `<condition>` 元素。
5. 在 `<condition>` 元素中，创建一个或多个元素，这些元素定义传输请求必须匹配的条件以调用您在步骤 1 中定义的 Connect:Direct 流程。这些元素可以是 `<match>` 元素或 `<defined>` 元素。
 - 使用 `<match>` 元素指定变量的值必须与模式匹配。创建具有以下属性的 `<match>` 元素：
 - `variable` - 要比较其值的变量的名称。变量是一个内置符号。要了解更多信息，请参阅第 724 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』。
 - `value` - 要与指定变量的值比较的模式。
 - （可选）`pattern` - `value` 属性的值使用的模式类型。该模式类型可以是 `wildcard` 或 `regex`。该属性为可选项，缺省值为 `wildcard`。
 - 使用 `<defined>` 元素指定变量必须具有定义的值。创建具有以下属性的 `<defined>` 元素：
 - `variable` - 必须具有定义值的变量的名称。变量是一个内置符号。要了解更多信息，请参阅第 724 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』。

在 `<condition>` 元素中指定的条件将通过逻辑 AND 组合。必须满足所有条件，Connect:Direct 网桥代理才能调用此 `<processSet>` 元素指定的进程。如果未指定 `<condition>` 元素，那么进程设置将匹配所有传输。

6. 在 `<processSet>` 元素中，创建 `<process>` 元素。
7. 在 `<process>` 元素中，创建 `<transfer>` 元素。

传输元素指定 Connect:Direct 网桥代理在传输过程中调用的 Connect:Direct 进程。创建具有以下属性的 `<transfer>` 元素：

- `process` - 您在步骤 1 中定义的 Connect:Direct 进程的位置。使用绝对路径或相对于 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` 目录指定此文件的位置。

结果

搜索条件匹配项时，Connect:Direct 网桥代理将从文件开头到文件末尾进行搜索。找到的第一个匹配既是使用的匹配。

相关任务

第 204 页的『配置 Connect:Direct 网桥』

配置 Connect:Direct 网桥，以在 IBM MQ Managed File Transfer 网络和 Connect:Direct 网络之间传输文件。Connect:Direct 网桥的组件包括 Connect:Direct 节点和专用于与该节点通信的 IBM MQ Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

相关参考

第 623 页的『Connect:Direct 进程定义文件格式』

Connect:Direct 网桥代理配置目录中的 ConnectDirectProcessDefinitions.xml 文件指定要在文件传输过程中启动的用户定义的 Connect:Direct 进程。

第 469 页的『fteCreateCDAgent（创建 Connect:Direct 网桥代理）』

fteCreateCDAgent 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

在由 IBM MQ Managed File Transfer 调用的 Connect:Direct 进程中使用内置符号变量

您可以在 IBM MQ Managed File Transfer 传输中调用用户定义的 Connect:Direct 进程，并通过在进程定义中使用内置符号变量来将传输中的信息传递给 Connect:Direct 进程。

关于此任务

该示例使用内置符号变量，将 IBM MQ Managed File Transfer 传输中的信息传递到用户定义的 Connect:Direct 进程中。有关 IBM MQ Managed File Transfer 使用的内置符号变量的更多信息，请参阅第 724 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』。

在该示例中，文件从 IBM MQ Managed File Transfer 代理传输到 Connect:Direct 网桥节点。该传输的第一部分由 IBM MQ Managed File Transfer 执行。第二部分由用户定义的 Connect:Direct 进程执行。

过程

1. 创建使用内置符号变量的 Connect:Direct 进程

```
%FTEPNAME PROCESS
  SNODE=%FTESNODE
  PNODEID=(%FTEPUSER,%FTEPPASS)
  SNODEID=(%FTESUSER,%FTESPASS)

COPY001 COPY
  FROM (
    FILE=%FTEFFILE
    DISP=%FTEFDISP
  )
  TO (
    FILE=%FTETFILE
    DISP=%FTETDISP
  )
PEND
```

2. 将此过程保存到位于以下位置的文本文件中: `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent/Example.cdp`
3. 编辑 ConnectDirectProcessDefinition.xml 文件以包含用于调用您在步骤 1 中创建的 Connect:Direct 流程的规则。

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:cdprocess xmlns:tns="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/
  ConnectDirectProcessDefinitions ConnectDirectProcessDefinitions.xsd">

  <tns:processSet>
    <tns:condition>
      <tns:match variable="%FTESNODE" value="TOBERMORY" pattern="wildcard" />
    </tns:condition>
```

```

<tns:process>
  <tns:transfer process="Example.cdp" />
</tns:process>
</tns:processSet>

</tns:cdprocess>

```

在此示例中，如果将传输请求提交到将 TOBERMORY 作为其源或目标 Connect:Direct 节点的 Connect:Direct 网桥代理，那么将调用 Example.cdp Connect:Direct 进程。

- 提交满足在步骤 3 的 ConnectDirectProcessDefinition.xml 文件中定义的条件的文件传输请求。例如，

```

fteCreateTransfer -sa ORINOCO -da CD_BRIDGE
                  -sm QM_WIMBLEDON -dm QM_COMMON
                  -de overwrite -df TOBERMORY:/home/bulgaria/destination.txt
                  -sd leave c:\bungo\source.txt

```

在该示例中，目标 Connect:Direct 节点为 TOBERMORY。该节点是传输中的辅助节点，并且 %FTESNODE 的值设置为 TOBERMORY。该命令满足 ConnectDirectProcessDefinition.xml 文件中设置的条件。

- IBM MQ Managed File Transfer 将源文件传输到与 Connect:Direct 网桥代理相同系统上的临时位置。

- Connect:Direct 网桥代理根据传输请求中的信息和配置信息来设置内置符号变量的值。

内置符号变量可设置为以下值：

- %FTEPNAME=*process_name* - 该值是由 Connect:Direct 网桥代理生成的 8 字符进程名称。
- %FTESNODE=TOBERMORY - 该值根据 **fteCreateTransfer** 命令的 **-df** 参数设置。
- %FTEPUSER=*primary_node_user* - 该信息取自 ConnectDirectCredentials.xml 文件。
- %FTEPPASS=*primary_node_user_password* - 该信息取自 ConnectDirectCredentials.xml 文件。
- %FTESUSER=*secondary_node_user* - 该信息取自 ConnectDirectCredentials.xml 文件。
- %FTESPASS=*secondary_node_user_password* - 该信息取自 ConnectDirectCredentials.xml 文件。
- %FTEFFILE=*temporary_location* - 该值是与 Connect:Direct 网桥代理在同一系统上的文件的临时位置。
- %FTEFDISP=leave - 该值根据 **fteCreateTransfer** 命令的 **-sd** 参数设置。
- %FTETFILE=/home/bulgaria/destination.txt - 该值根据 **fteCreateTransfer** 命令的 **-df** 参数设置。
- %FTETDISP=overwrite - 该值根据 **fteCreateTransfer** 命令的 **-de** 参数设置。

- Connect:Direct 进程在 Connect:Direct 网桥节点上启动。Connect:Direct 将文件从 Connect:Direct 网桥系统上的临时位置传输到运行 Connect:Direct 节点 TOBERMORY 的系统上的目标 /home/bulgaria/destination.txt。

相关概念

第 297 页的『通过文件传输请求提交用户定义的 Connect:Direct 进程』

对于通过 Connect:Direct 网桥代理以在文件传输过程中调用用户定义的 Connect:Direct 进程的传输，您可以提交一个传输请求。

相关参考

第 724 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』

您可以通过使用特定于 IBM MQ Managed File Transfer 的内置符号变量来定义代入用户定义的 Connect:Direct 进程的值。

使用 Connect:Direct 进程来提交 IBM MQ Managed File Transfer 传输请求

您可以通过 Connect:Direct 进程向 Connect:Direct 网桥代理提交传输请求。IBM MQ Managed File Transfer 提供了可以从 Connect:Direct 进程中的 **RUN TASK** 语句调用的命令。

IBM MQ Managed File Transfer 提供了以下与 Connect:Direct 进程一起使用的命令：

ftetag

在 **ftebxfer** 或 **ftecxfer** 命令之前的步骤中指定该命令，以针对传输创建所需的审计信息。该命令使用传输的源规范作为参数。有关源规范格式的信息，请参阅第 497 页的『[fteCreateTransfer（创建新的文件传输）](#)』。

ftebxfer

指定该命令，以在传输请求提交给的队列管理器与提交该命令的 Connect:Direct 节点在同一系统中时创建文件传输请求。该命令使用与 **fteCreateTransfer** 命令相同的参数。有关这些参数的信息，请参阅第 497 页的『[fteCreateTransfer（创建新的文件传输）](#)』。该命令还具有其他参数：

-qmgrname

必需。要将命令提交给的队列管理器的名称。

ftecxfer

指定该命令，以在传输请求提交给的队列管理器与提交该命令的 Connect:Direct 节点位于不同系统上时创建文件传输请求。该命令使用与 **fteCreateTransfer** 命令相同的参数。有关这些参数的信息，请参阅第 497 页的『[fteCreateTransfer（创建新的文件传输）](#)』。该命令还具有以下三个参数：

-qmgrname

必需。要将命令提交给的队列管理器的名称。

-connname

必需。要将命令提交给的队列管理器的主机和端口，以 WebSphere MQ CONNAME 格式指定。例如，host.example.com(1337)。

-channelname

可选。用于连接到将命令提交给的队列管理器的通道的名称。如果未指定该参数，那么将使用缺省值 SYSTEM.DEF.SVRCONN。

相关任务

第 301 页的『[通过使用 Connect:Direct 请求程序来创建和提交调用 IBM MQ Managed File Transfer 的 Connect:Direct 进程](#)』

Connect:Direct 请求程序是一种图形用户界面，可用于创建和提交调用 IBM MQ Managed File Transfer 的 Connect:Direct 进程。

相关参考

第 727 页的『[调用 ftecxfer 命令的 Connect:Direct 进程文件示例](#)』

调用 IBM MQ Managed File Transfer **ftetag** 命令和 **ftecxfer** 命令的 Connect:Direct 进程文件示例。

通过使用 *Connect:Direct* 请求程序来创建和提交调用 *IBM MQ Managed File Transfer* 的 *Connect:Direct* 进程

Connect:Direct 请求程序是一种图形用户界面，可用于创建和提交调用 IBM MQ Managed File Transfer 的 Connect:Direct 进程。

关于此任务

此任务描述如何创建调用 IBM MQ Managed File Transfer **ftecxfer** 命令或 **ftebxfer** 命令的 Connect:Direct 进程。如果将传输请求提交到的队列管理器与提交该命令的 Connect:Direct 节点位于不同的系统上，请使用 **ftecxfer** 命令。当传输请求提交给的队列管理器与提交该命令的 Connect:Direct 节点位于同一个系统上时，请使用 **ftebxfer** 命令。**ftecxfer** 命令用于建立到传输的源代理的代理队列管理器的客户机连接。在调用 **ftecxfer** 命令之前，必须调用 **ftetag** 命令并为其传递源规范信息。这样就能够采用与从 MQMFT 启动的传输相同的方式来记录和审计进程。

过程

1. 启动 Connect:Direct 请求程序。
2. 在面板的节点选项卡中，选择用作进程主节点的 Connect:Direct 节点。
3. 选择文件 > 新建 > 进程。此时将打开“进程属性”窗口。
4. 在名称：字段中，输入进程的名称。

5. 从 **Snode > 名称**: 列表中选择辅助节点。
6. 从 **Snode > 操作系统**: 列表中选择辅助节点的操作系统。
7. 可选: 在该窗口中完成您所需的任何进一步信息。
8. 单击**确定**。此时将关闭“**进程属性**”窗口。
9. 创建用于运行 MQMFT **ftetag** 命令的语句。
 - a) 在“**进程**”窗口中, 右键单击 **End** 语句。
 - b) 选择**插入 > 运行任务**。此时将打开“**运行任务语句**”窗口。
 - c) 在 **标签:** 字段中, 输入 Tag。
 - d) 在 **可选参数或命令** 字段中, 输入 `pgm(MQ_INSTALLATION_PATH/bin/ftetag) args(source_specification)`。有关 *source_specification* 格式的更多信息, 请参阅第 497 页的『[fteCreateTransfer \(创建新的文件传输\)](#)』。
 - e) 单击**确定**。此时将关闭“**运行任务语句**”窗口。
10. 创建运行 MQMFT **ftecxfier** 或 **ftebxfer** 命令的语句。
 - a) 在“**进程**”窗口中, 右键单击 **End** 语句。
 - b) 选择**插入 > 运行任务**。此时将打开“**运行任务语句**”窗口。
 - c) 在 **标签:** 字段中, 输入 Transfer。
 - d) 在 **可选参数或命令** 字段中, 根据您选择的命令, 输入 `pgm(MQ_INSTALLATION_PATH/bin/ftecxfier) args(parameters)` 或 `pgm(MQ_INSTALLATION_PATH/bin/ftebxfer) args(parameters)`。**ftecxfier** 和 **ftebxfer** 命令使用的参数与 **fteCreateTransfer** 命令使用的参数相同, 但另外有一些特定于 **ftecxfier** 和 **ftebxfer** 的额外参数。有关更多信息, 请参阅第 497 页的『[fteCreateTransfer \(创建新的文件传输\)](#)』和第 300 页的『[使用 Connect:Direct 进程来提交 IBM MQ Managed File Transfer 传输请求](#)』。
 - e) 单击**确定**。此时将关闭“**运行任务语句**”窗口。
11. 可选: 创建所需的所有其他语句。
12. 提交进程。
 - a) 在“**进程**”窗口中右键单击。
 - b) 选择**提交**。此时将打开“**Connect:Direct 连接**”窗口。
 - c) 输入用于运行进程的用户名和密码。
 - d) 单击**确定**。

相关概念

第 300 页的『[使用 Connect:Direct 进程来提交 IBM MQ Managed File Transfer 传输请求](#)』

您可以通过 Connect:Direct 进程向 Connect:Direct 网桥代理提交传输请求。IBM MQ Managed File Transfer 提供了可以从 Connect:Direct 进程中的 **RUN TASK** 语句调用的命令。

使用 IBM Integration Bus

您可以通过 FTEOutput 和 FTEInput 节点使用来自 IBM Integration Bus 的 IBM MQ Managed File Transfer。

- 通过 IBM MQ Managed File Transfer, 使用 FTEInput 节点在网络间传输文件, 然后将该文件作为 Integration Bus 流的一部分进行处理。
- 使用 FTEOutput 节点将已由 Integration Bus 流输出的文件传输到网络中的另一位置。

与 Broker 代理相互传输文件的代理可以处于 IBM MQ Managed File Transfer 的任何级别。

有关更多信息, 请参阅 [IBM Integration Bus 产品文档](#)。

IBM MQ Managed File Transfer 的恢复和重新启动

如果由于任何原因 (例如, 由于电源故障或网络故障) 您的代理或队列管理器不可用, 那么 IBM MQ Managed File Transfer 将在这些场景下按如下方式恢复:

- 通常，如果在文件传输过程中出现问题，IBM MQ Managed File Transfer 会在修复问题后恢复并重新启动该文件传输。
- 如果正在进行传输的文件在代理或队列管理器不可用时被删除或更改，那么传输将失败，并且您将在传输日志中收到一条消息，提供有关此失败的详细信息。
- 如果在文件传输期间代理进程失败，那么该传输将在您重新启动代理后继续。
- 如果代理失去了与其代理队列管理器的连接，那么代理将等待尝试重新连接到队列管理器。代理成功重新连接到其队列管理器后，当前传输将继续。
- 如果代理因任何原因而停止，那么与代理关联的任何资源监视器都将停止轮询。代理恢复后，将重新启动监视器，资源轮询将恢复。
- 对于源文件被删除的文件传输，如果在将所有数据从源代理发送到目标代理后进行恢复，那么会在删除源文件之前解锁源文件。解锁意味着可能在删除源文件之前对其进行了修改。因此，删除源文件可能是不安全的，并会显示以下警告：

```
BFGTR0075W: The source file has not been deleted because it is possible that the source file
was modified after the source file was transferred.
```

在此情况下，请确认没有修改源文件的内容，然后手动删除源文件。

您可以在 WebSphere MQ Explorer 中查看传输的状态。如果任何传输显示为 **Stalled**，那么您可能需要执行更正操作，因为停滞状态表示代理存在问题，或者在传输中涉及的两个代理之间存在问题。

开发应用程序

指定要运行的程序

您可在正在运行 IBM MQ Managed File Transfer 代理的系统上运行程序。作为文件传输请求的一部分，您可以指定在传输启动前或者完成后运行的程序。此外，您可以通过提交受管呼叫请求来启动不属于文件传输请求的程序。

您可在以下五种场景下指定要运行的程序：

- 作为传输请求的一部分，在传输启动前从源代理处
- 作为传输请求的一部分，在传输启动前从目标代理处
- 作为传输请求的一部分，在传输完成后从源代理处
- 作为传输请求的一部分，在传输完成后从目标代理处
- 不作为传输请求的一部分。您可向代理提交请求以运行程序。本场景有时候称为受管呼叫。

有多种方式来指定要运行的程序。这些选项如下：

使用 Apache Ant 任务

使用以下其中一个任务来启动程序：`fte:filecopy`、`fte:filemove` 或 `fte:call` Ant 任务。您可以使用 Ant 任务在五种场景（使用 `fte:presrc`、`fte:predst`、`fte:postdst`、`fte:postsrc` 和 `fte:command` 嵌套元素）下指定程序。有关更多信息，请参阅第 968 页的『程序调用嵌套元素』。

编辑文件传输请求消息

您可以编辑传输请求生成的 XML。您可以使用该方法在五种场景下通过将 **preSourceCall**、**postSourceCall**、**preDestinationCall**、**postDestinationCall** 和 **managedCall** 元素添加到 XML 文件中，以运行程序。然后，使用这一经过修改的 XML 文件作为新文件传输请求的传输定义，例如，使用 **fteCreateTransfer -td** 参数。有关更多信息，请参阅第 858 页的『调用请求消息示例』。

使用 **fteCreateTransfer** 命令

您可以使用 **fteCreateTransfer** 命令来指定要启动的程序。您可以使用该命令来指定在前四个场景中作为传输请求的一部分运行的程序，但是您无法启动受管呼叫。有关要使用的参数的信息，请参阅第

[497 页的『fteCreateTransfer（创建新的文件传输）』](#)。有关使用该命令的示例，请参阅[第 906 页的『使用 fteCreateTransfer 来启动程序的示例』](#)。

使用 Web 网关

如果您已配置 Web 网关，那么可以在传输完成后在目标代理处运行程序。您无法使用该方法在传输启动前提交受管呼叫请求或者在源代理或目标代理处运行程序。指定 `x-fte-postdest` 头或者在 HTTP 请求中使用 `postdest` 表单字段。有关更多信息，请参阅[第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』](#)。

相关参考

[第 443 页的『commandPath 属性』](#)

使用 `commandPath` 属性以指定 IBM MQ Managed File Transfer 可从中运行命令的位置。设置该属性时必须谨慎，因为能够向代理发送命令的远程客户机系统可以有效调用指定的某个 `commandPath` 中的任何命令。

IBM MQ Managed File Transfer Web 网关

Web 网关提供可用于与 IBM MQ Managed File Transfer 网络交互的 RESTful API。

本部分说明了 Web 网关的概念以及 Web 网关如何适合您的现有受管文件传输网络。有关更多信息，请参阅[第 305 页的『Web 网关的方案』](#)和[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)。有关可发送到 Web 网关的 HTTP 请求的示例，请参阅[第 312 页的『使用 IBM MQ Managed File Transfer 服务 Web 网关』](#)。

有关在应用程序服务器中配置 Web 网关和保护其安全的信息，请参阅[第 181 页的『配置 Web 网关』](#)和[第 106 页的『保证 Web 网关的安全』](#)。要检查 Web 网关的设置，请参阅[第 201 页的『验证 Web 网关安装』](#)。

有关 Web 网关 RESTful API 的参考信息，请参阅[第 908 页的『Web 网关 API 引用』](#)。

要解决与 Web 网关相关的问题，请参阅[第 412 页的『对 Web 网关进行故障诊断』](#)。

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 312 页的『使用 IBM MQ Managed File Transfer 服务 Web 网关』](#)

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

[第 327 页的『管理 IBM MQ Managed File Transfer 服务 Web 网关』](#)

您可以创建和删除文件空间，控制对个别文件空间具有访问权的用户。

[第 339 页的『文件空间』](#)

文件空间是与 Web 网关用户关联的文件存储器的保留区域。文件空间具有配额存储量。文件空间的访问权受限于有权对其执行读写的文件空间。

相关参考

[第 908 页的『Web 网关 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

[第 516 页的『fteCreateWebAgent（创建 IBM MQ Managed File Transfer Web 代理）』](#)

fteCreateWebAgent 命令创建代理及其关联的配置，以用于 Web 网关。IBM MQ Managed File Transfer 服务器随附该命令。

第 353 页的『样本 Web 页面』

IBM MQ Managed File Transfer Web 网关提供了样本 Web 页面。该样本使用 Web 网关 API 函数来上传文件、查看文件传输的状态、查看文件空间的内容以及从文件空间下载文件。

第 948 页的『Web 网关所使用的数据库表』

IBM MQ Managed File Transfer Web 网关使用以下数据库表配置用户文件空间并保证用户文件空间的安全。

Web 网关的方案

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

对于系统中的文件，如果您不想运行代理但可以使用 HTTP 客户机，那么 Web 网关将非常有用。例如，您可以使用 Web 网关执行以下任务：

- 将文件从 Web 页面发送到 IBM MQ Managed File Transfer 代理
- 监视来自 Web 页面的传输的状态
- 从不能运行 IBM MQ Managed File Transfer 基础结构但具有 HTTP 功能的便携式设备发送文件
- 从不支持 IBM MQ Managed File Transfer 代理的操作系统发送文件

使用 Web 网关上传文件

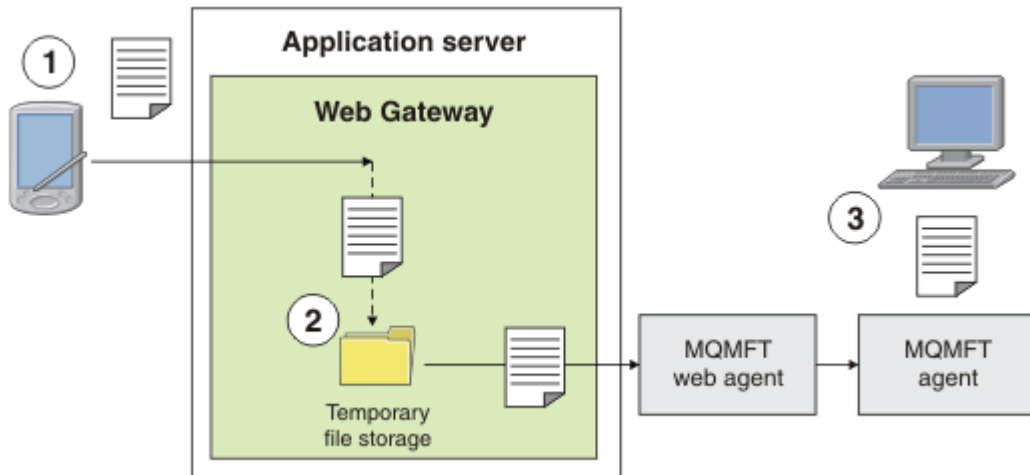


图 7: 使用 Web 网关将文件上传到 IBM MQ Managed File Transfer 网络

您可以使用 HTTP 客户机将文件上传到 Web 网关。托管 Web 网关应用程序的应用程序服务器会接收到 HTTP 请求而临时存储该文件，直到 Web 代理开始传输该文件。Web 代理将文件传输到原始传输请求中指定的目标代理。如图 1 所示，提交传输请求的 HTTP 客户机无需安装代理。目标系统必须已安装代理，并且托管 Web 网关应用程序的系统必须已安装 Web 代理。

从文件空间下载文件

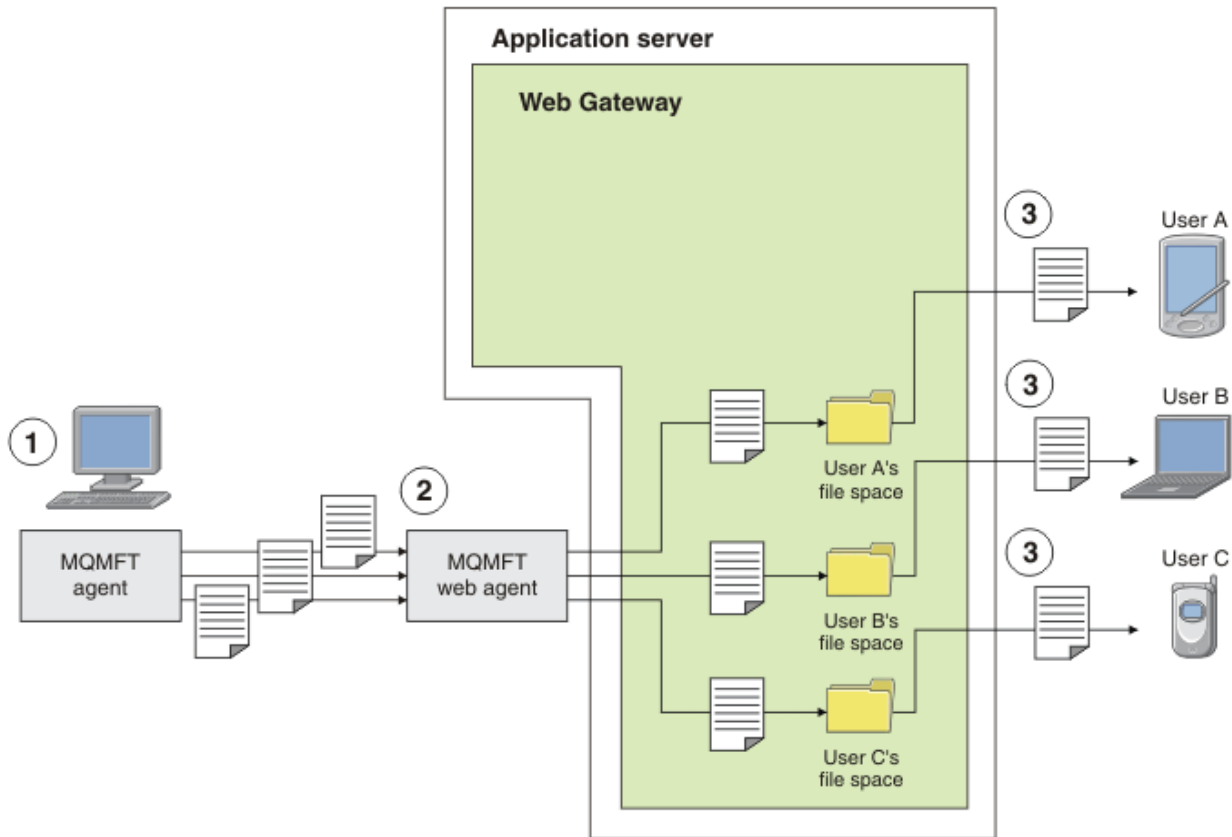


图 8: 使用 Web 网关从文件空间下载文件

您可以使用 Web 网关将文件提供给文件空间中的用户。文件空间是与 Web 网关用户关联的文件存储器的保留区域。可以使用代理将文件传输到 Web 网关。位于 Web 网关应用程序所在系统上的 Web 代理将文件传输到您在传输请求中指定的文件空间。拥有文件空间的用户可以在自己方便的时候下载文件，而且无需使用代理或其他 IBM MQ Managed File Transfer 基础结构即可下载文件。

如何使用 Web 网关

IBM MQ Managed File Transfer 提供了管理控制台。可以使用管理控制台创建文件空间，修改对文件空间具有访问权的用户集以及将用户映射到 IBM MQ 消息描述符 (MQMD) 用户标识。有关使用管理控制台的更多信息，请参阅第 327 页的『[管理 IBM MQ Managed File Transfer 服务 Web 网关](#)』。

如果您愿意，可以通过直接使用 Web 网关随附的应用程序编程接口 (API) 进行编程来构建定制应用程序。有关更多信息，请参阅第 908 页的『[Web 网关 API 引用](#)』和第 930 页的『[Web 网关管理 API 引用](#)』。该 API 主要支持三种构建应用程序的方法。这些字段为：

Web 应用程序

您可以编写一组 Web 页面或 Web 应用程序，他们使用 Web 网关 API 函数来执行其函数中与文件相关的部分。Web 网关随附了样本应用程序，其中演示了一种执行此操作的方法。有关更多信息，请参阅第 353 页的『[样本 Web 页面](#)』。

客户机应用程序

您可以使用 Perl、Ruby 或 Python 等语言编写在客户机系统上运行并使用 Web 网关 API 函数与 IBM MQ Managed File Transfer 通信的程序。几乎所有编程语言都提供了 HTTP 设施。该方法的优点是使您可以从不能部署 IBM MQ Managed File Transfer 代理的平台与 IBM MQ Managed File Transfer 交互。

系统集成

该方法使用与客户机应用程序选项相同的技术，但集成了数据中心内的不同系统。HTTP 提供支持在不同的工具和系统之间通信的通用标准。

相关概念

第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

第 312 页的『使用 IBM MQ Managed File Transfer 服务 Web 网关』

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

第 327 页的『管理 IBM MQ Managed File Transfer 服务 Web 网关』

您可以创建和删除文件空间，控制对个别文件空间具有访问权的用户。

第 339 页的『文件空间』

文件空间是与 Web 网关用户关联的文件存储器的保留区域。文件空间具有配额存储量。文件空间的访问权受限于有权对其执行读写的文件空间。

相关参考

第 908 页的『Web 网关 API 引用』

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

第 930 页的『Web 网关管理 API 引用』

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

使用 Web 网关可扩展现有 IBM MQ Managed File Transfer 网络，以支持使用 HTTP 协议的客户机。Web 网关提供从使用 HTTP 协议的客户机到已存在的 IBM MQ Managed File Transfer 网络的链接。使用 Web 网关的传输在整个传输过程中都会进行记录。有关 Web 网关用途的更多信息，请参阅第 305 页的『Web 网关的方案』。

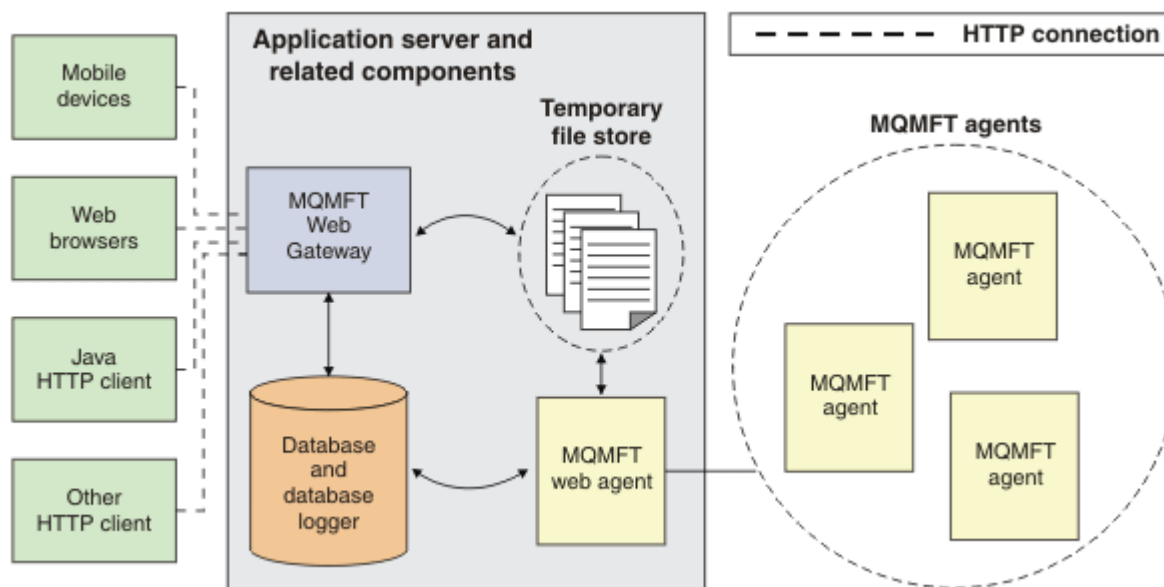


图 9: Web 网关体系结构的概述

Web 网关应用程序需要以下未随附于 IBM MQ Managed File Transfer 的组件：

符合 Java Platform, Enterprise Edition 5 的应用程序服务器

该应用程序服务器托管 Web 网关应用程序。来自客户机的 HTTP 请求会定向到该应用程序服务器，然后请求的内容会经过该应用程序服务器传递给应用程序。

Web 网关由以下几部分组成：

MQMFT Web 网关应用程序

Web 网关应用程序同时处理文件上传和传输状态请求。

上传文件时，Web 网关应用程序将文件数据写入到运行应用程序的系统上文件系统的临时存储器中。然后，Web 网关应用程序将文件传输请求提交给在同一系统上运行的 MQMFT 代理。有关该请求的更多信息，请参阅第 842 页的『文件传输请求消息格式』。

在收到状态信息请求后，Web 网关应用程序连接到 MQMFT 数据库记录器数据库（使用应用程序服务器提供的数据库访问功能），以检索所需信息。然后，应用程序会生成响应，并传递到客户机。

MQMFT Web 代理

Web 网关要求在应用程序所在的系统上安装 MQMFT 代理。可以使用 `fteCreateWebAgent` 命令创建此 Web 代理程序；请参阅第 516 页的『`fteCreateWebAgent`（创建 IBM MQ Managed File Transfer Web 代理）』。该代理接收前面部分所述的文件传输请求消息。请求消息是指临时存储器中的文件。该代理通过从文件系统存储读取文件，将文件传输到 MQMFT 网络中的现有代理。源处置行为设置为 `delete`，以便在传输成功完成后除去文件，请参阅 `fteCreateTransfer` 以获取更多信息。

您无需特别配置该代理，因为文件传输请求属于常规消息，并非特别针对 Web 网关。

MQMFT 数据库记录器和受支持的数据库

要提供有关使用 Web 或其他方法启动的传输的状态信息，Web 网关应用程序必须能够查询包含 MQMFT 活动审计信息的数据库。该数据库由产品随附的数据库记录器组件填充。每个应用程序服务器中包含的数据访问设施提供了数据库访问功能。该数据库无需与其他组件位于同一系统上。

Web 网关方案需要的组件

以下各图显示了文件传输请求中涉及的 IBM MQ Managed File Transfer 组件以及其他对象。必须在应用程序服务器中定义每个方案中使用的所有 Java Platform, Enterprise Edition (JEE) 资源，而无论您使用的是哪个方案。有关如何配置 JEE 资源的详细信息，请参阅第 181 页的『配置 Web 网关』。

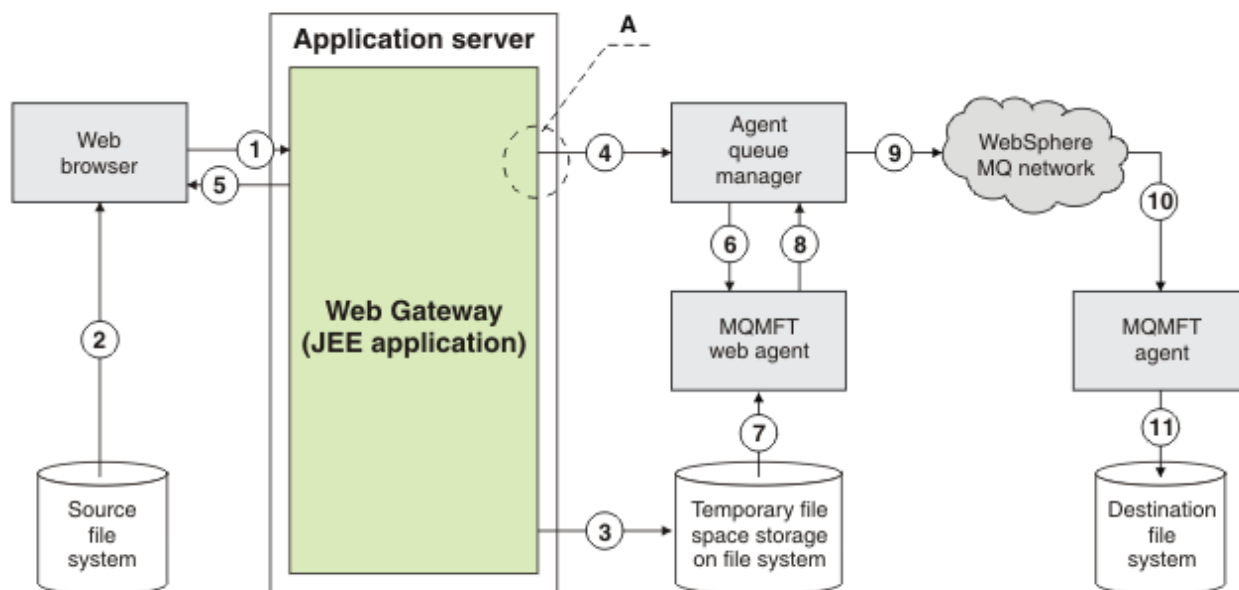


图 10: 通过 Web 网关上传文件涉及的 IBM MQ Managed File Transfer 组件

1. 用户的 Web 浏览器中运行的 JavaScript 应用程序使用 Web 网关提供的 RESTful API 定义的函数来上载文件。
2. 从与 Web 浏览器位于同一系统的文件存储器中读取文件数据，并使用 HTTP 协议将这些数据发送到托管 Web 网关应用程序的应用程序服务器。
3. Web Gateway Java Platform, Enterprise Edition (JEE) 应用程序接收作为 HTTP 请求主体的文件数据，并将其写入可从应用程序服务器和 Web 代理程序访问的文件存储器。如果 Web 网关应用程序和 Web 代理位于同一系统上，那么该存储器可以是系统的文件系统上的目录。
4. Web 网关应用程序将消息发送到 Web 代理所连接的代理队列管理器。该消息包含一些指示信息，用于标识要移动的文件和文件数据要发送到的 IBM MQ Managed File Transfer 代理。该信息来自步骤 1 中的 HTTP 请求。
5. Web 网关 JEE 应用程序将 HTTP 响应发送到 Web 浏览器。
6. Web 代理接收请求传输文件数据的消息。
7. Web 代理读取与步骤 1 中上载的文件相对应的文件数据。
8. Web 代理将文件数据作为消息序列传输到代理队列管理器。
9. 代理队列管理器通过 IBM MQ 网络传输与步骤 1 中上载的文件相对应的消息。这可能涉及在更多队列管理器间交换文件数据，直到数据到达目标系统上运行的代理所连接的队列管理器。
10. 目标系统上的代理接收包含文件数据的消息，并将数据重新转换为文件。
11. 文件数据写入到目标系统上的文件存储器中。

该方案中使用的 JEE 资源：

A - 名为 WMQFTEWebAgentConnectionFactory 的 JMS 队列连接工厂，其 JNDI 名称为 `jms/WMQFTEWebAgentConnectionFactory`

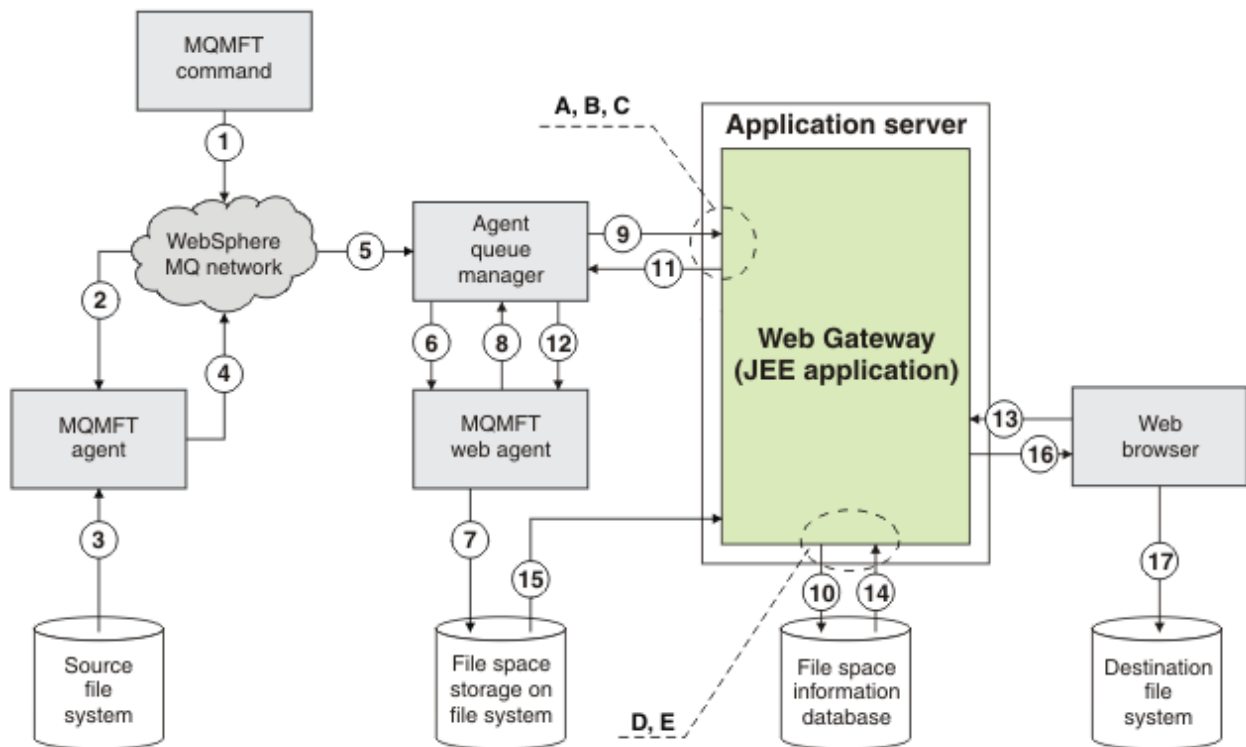


图 11: 在将文件上载到文件空间以及随后从文件空间下载文件的过程中涉及到的 IBM MQ Managed File Transfer 组件

1. 用户或进程将文件传输请求（以 IBM MQ 消息的形式）发送到 IBM MQ 网络。可以通过命令行或其他 MQMFT 界面发送该请求。消息会发送到源系统上的代理所连接的队列管理器。
2. 源系统上的代理接收指示该代理向 Web 代理传输文件的消息。

3. 代理从源文件系统读取文件，并将该文件转换为 IBM MQ 消息序列。
4. 代理将消息序列发送到 IBM MQ 网络中的队列管理器。
5. IBM MQ 网络将包含文件数据的消息路由到代理队列管理器。
6. Web 代理从代理队列管理器接收包含文件数据的消息。
7. Web 代理将文件数据作为文件写入到 Web 网关 JEE 应用程序可访问的文件系统的文件空间存储器中。
8. Web 代理将消息发送到代理队列管理器，以通知 Web 网关 JEE 应用程序有文件到达。
9. Web 网关 JEE 应用程序通过代理队列管理器接收从 Web 代理发送的通知消息。
10. Web 网关 JEE 应用程序更新包含文件空间中所存储文件的相关信息的数据库。
11. Web 网关 JEE 应用程序将指定发往 Web 代理的响应发送到代理队列管理器。
12. Web 代理接收响应消息，并完成文件传输操作。
13. 稍后，用户或进程向 Web 网关 JEE 应用程序发出 RESTful HTTP 请求，以从用户的文件空间检索文件。在该图中由 Web 浏览器发出该请求。任何 HTTP 客户机都可发出该请求。
14. Web 网关 JEE 应用程序接收 HTTP 请求，进行解码，并使用文件空间信息数据库来查找文件数据。
15. Web 网关 JEE 应用程序从文件空间存储器读取文件数据，该存储器位于可从 Web 网关 JEE 应用程序访问的文件系统中。
16. Web 网关 JEE 应用程序将文件数据发送回请求该数据的 Web 浏览器。
17. Web 浏览器将文件数据写入到目标系统上的文件系统中。

该方案中使用的 JEE 资源：

A - 名为 WMQFTEWebAgentRequestQueue 的 JMS 队列，其 JNDI 名称为 jms/WMQFTEWebAgentRequestQueue

B - 名为 WMQFTEWebAgentConnectionFactory 的 JMS 队列连接工厂，其 JNDI 名称为 jms/WMQFTEWebAgentConnectionFactory

C - 名为 WMQFTEActivationSpec 的激活规范，其 JNDI 名称为 jms/WMQFTEActivationSpec，该规范使用 Web 代理的队列管理器的连接详细信息进行配置

D - 名为 wmqfte-filespace 的数据源，其 JNDI 名称为 jdbc/wmqfte-filespace

E - 数据源 jdbc/wmqfte-filespace 引用的 JDBC 提供程序

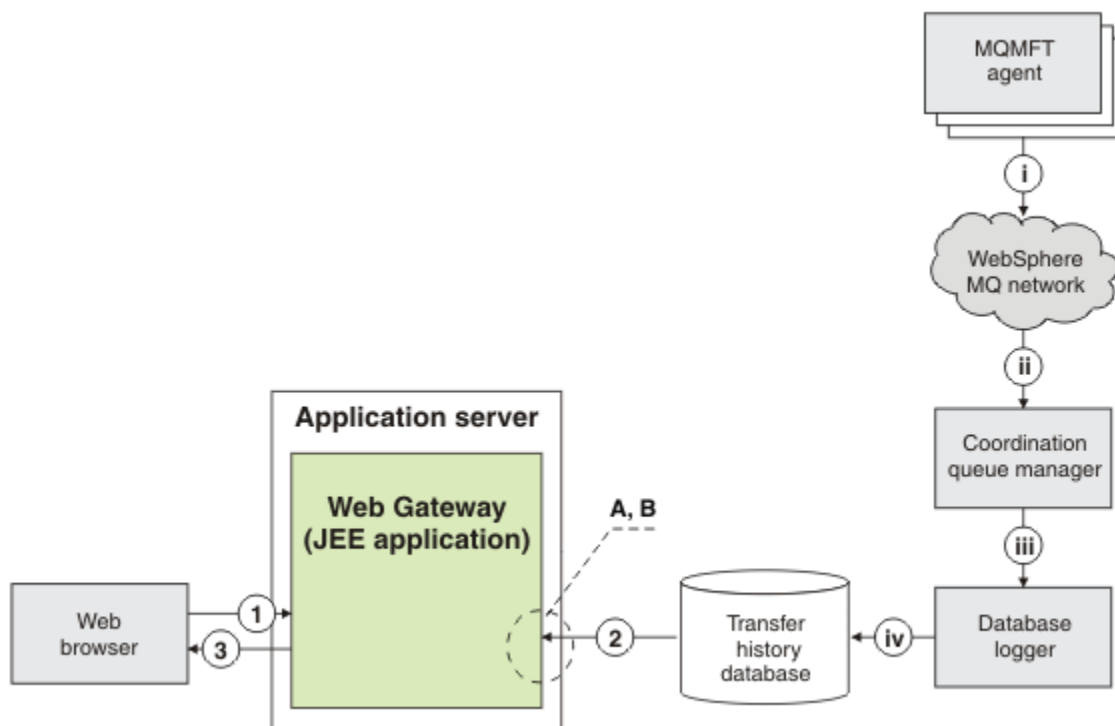


图 12: 通过 Web 网关请求文件传输的状态

1. 用户的 Web 浏览器中运行的 JavaScript 应用程序将 RESTful HTTP 请求发送到 Web 网关应用程序，请求有关传输的信息。
2. Web 网关应用程序查询一个数据库，该数据库中包含 IBM MQ Managed File Transfer 代理的网络中所进行的文件传输的相关信息。
3. Web 网关应用程序将查询结果返回到 JavaScript 应用程序。

先前步骤中发生的活动：

- i - IBM MQ Managed File Transfer 代理生成消息，其中包含有关所进行的文件传输的信息。
- ii - 队列管理器将这些消息路由到担当协调队列管理器角色的指定队列管理器。
- iii - 协调队列管理器连接到数据库记录器组件。数据库记录器接收与代理所执行的传输相关的每条消息的副本。
- iv - 数据库记录器在传输历史记录数据库中记录有关传输的信息，以便包括 Web 网关在内的其他应用程序可以查询。

该方案中使用的 JEE 资源：

- A - 名为 wmqfte-filespace 的数据源，其 JNDI 名称为 jdbc/wmqfte-database
- B - 数据源 wmqfte-database 引用的 JDBC 提供程序

相关概念

第 305 页的『Web 网关的方案』

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

第 312 页的『使用 IBM MQ Managed File Transfer 服务 Web 网关』

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

第 327 页的『管理 IBM MQ Managed File Transfer 服务 Web 网关』

您可以创建和删除文件空间，控制对个别文件空间具有访问权的用户。

第 339 页的『文件空间』

文件空间是与 Web 网关用户关联的文件存储器的保留区域。文件空间具有配额存储量。文件空间的访问权受限于有权对其执行读写的文件空间。

相关参考

[第 908 页的『Web 网关 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

使用 IBM MQ Managed File Transfer 服务 Web 网关

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

在配置或使用 Web 网关之前，请参阅 [第 305 页的『Web 网关的方案』](#)和 [第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)。这些主题解释了 Web 网关的用途和组件。

您可以使用 HTTP 头或 HTML 表单字段在请求中提供更多信息，从而定制 HTTP 请求。有关可用选项的更多信息，请参阅[第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』](#)。

以下主题描述了如何创建要提交到 Web 网关的 HTTP 请求。有关这些请求的格式和 Web 网关 API 的更多信息，请参阅 [第 908 页的『Web 网关 API 引用』](#)。

您无需管理权限也可使用这些示例。如果要管理 Web 网关（例如，为用户创建或删除文件空间），请参阅主题[第 327 页的『管理 IBM MQ Managed File Transfer 服务 Web 网关』](#)。

相关概念

[第 313 页的『示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

[第 339 页的『文件空间』](#)

文件空间是与 Web 网关用户关联的文件存储器的保留区域。文件空间具有配额存储量。文件空间的访问权受限于有权对其执行读写的文件空间。

相关任务

[第 312 页的『示例：将文件传输到文件空间』](#)

将单个文件传输到 IBM MQ Managed File Transfer 文件空间。通过在 **fteCreateTransfer** 命令中使用 **-du** 参数，可将文件空间指定为文件传输的目标。

[第 326 页的『示例：使用 HTML 表单发送文件』](#)

通过使用 IBM MQ Managed File Transfer Web 网关来提交请求，可以将单个文本文件发送到目标文件系统。

相关参考

[第 908 页的『Web 网关 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

示例：将文件传输到文件空间

将单个文件传输到 IBM MQ Managed File Transfer 文件空间。通过在 **fteCreateTransfer** 命令中使用 **-du** 参数，可将文件空间指定为文件传输的目标。

关于此任务

将文件传输到文件空间时，IBM MQ Managed File Transfer Web 网关将检查传输是否会导致超出文件空间配额。如果会超出配额，那么将生成错误，并且文件传输将失败。Web 网关管理员可以通过提交 HTTP 请求来增加文件空间配额的大小。有关示例请求，请参阅主题[第 331 页的『示例：修改文件空间配置』](#)。

文件空间配额检查在传输开始之前进行。如果使用多个代理将文件传输到同一个文件空间，或者当文件正在传输到文件空间时 Web 网关管理员降低了该文件空间的配额，那么尽管一个或多个传输导致超出文件空间配额，但它们仍可能会成功。

在该示例中，源文件名为 `/tmp/Accounts.csv`，该文件与源代理 `AGENT_1` 位于同一系统上。目标文件空间 `john` 属于用户 `john`，与代理 `FS_AGENT` 位于同一系统上。请求传输的用户对文件空间 `john` 具有写访问权。代理 `FS_AGENT` 使用队列管理器 `FS_QM`。

过程

输入以下命令：

```
fteCreateTransfer -sa AGENT_1 -da FS_AGENT -dm FS_QM -du john /tmp/Accounts.csv
```

文件 `/tmp/Accounts.csv` 传输到文件空间 `john`。用户 `john` 可在需要时从文件空间下载该文件。

相关概念

[第 339 页的『文件空间』](#)

文件空间是与 Web 网关用户关联的文件存储器的保留区域。文件空间具有配额存储量。文件空间的访问权受限于有权对其执行读写的文件空间。

相关任务

[第 221 页的『启动新的文件传输』](#)

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

示例 HTTP 流

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

您可以使用各种技术向 Web 网关提交请求或解释来自 Web 网关的响应。例如，您可以写 Web 应用程序。有关 Web 网关随附的示例 Web 应用程序的信息，请参阅[第 353 页的『样本 Web 页面』](#)。

如果要使用 Web 应用程序与 Web 网关通信，可以使用 HTML 格式或 Javascript XMLHttpRequest 函数。要上载文件，必须使用 HTML 格式，因为浏览器会出于安全原因阻止 Javascript 从本地系统访问文件。如果您愿意，可以通过 Javascript 控制和提交格式。要请求传输状态，XMLHttpRequest 最有可能适用，虽然也可用其他技术；例如，将内容装入不可视的 iFrame 元素。

也可以采用诸如 Ruby 或 Perl 之类语言编写客户机应用程序，以与 Web 网关 API 通信。

相关任务

[第 314 页的『示例：使用 HTTP 请求发送文件』](#)

通过 IBM MQ Managed File Transfer Web 网关提交请求，可将单个文件发送到目标代理文件系统。

[第 315 页的『示例：使用 HTTP 请求查看文件传输的状态』](#)

通过使用 IBM MQ Managed File Transfer Web 网关来提交请求，可以查看文件传输的状态。Web 网关将返回 XML 格式的信息，其中描述了指定传输的当前状态。要使用 Web 网关来查看文件传输的状态，您的 IBM MQ Managed File Transfer 网络中必须存在数据库记录器。

[第 316 页的『示例：使用 HTTP 请求查询多个文件传输』](#)

通过 IBM MQ Managed File Transfer Web 网关提交请求，可查询多个文件传输的状态。Web 网关会以 XML 或 JSON 格式返回信息，其中描述与查询匹配的传输的状态。

[第 322 页的『示例：列出文件空间中的所有文件』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可列出文件空间的内容。Web 网关会以 XML 或 JSON 格式返回响应，其中列出文件空间的内容。如果您是文件空间的所有者或具有安全角色 `wmqfte-admin`，那么您有权列出文件空间的内容。

[第 334 页的『示例：检查文件空间中文件的完整性』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可检查文件空间中文件的完整性。例如，如果要在数据丢失之后复原文件系统，那么可以检查文件空间中的文件是否位于文件系统上的正确位置。Web 网关会以 XML 或 JSON 格式返回响应，其中列出文件空间的内容以及用于指示每个文件的完整性检查结果的附加属性。

[第 323 页的『示例：列出文件空间中的特定文件子集』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交包含查询的 HTTP 请求，可查询文件空间的内容。Web 网关会以 XML 或 JSON 格式返回响应，其中仅描述文件空间中与查询匹配的文件。

[第 324 页的『示例：从文件空间检索文件』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可从文件空间检索文件。Web 网关提供使用 HTTP 协议下载文件的功能。

[第 325 页的『示例：从文件空间中删除文件』](#)

可以通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，从文件空间中删除文件。如果将头 `x-fte-include-file-in-response` 设置为 `true`，那么将通过 Web 网关在 HTTP 响应中返回文件内容。

示例：使用 HTTP 请求发送文件

通过 IBM MQ Managed File Transfer Web 网关提交请求，可将单个文件发送到目标代理文件系统。

关于此任务

可以使用 `multipart/form-data` Content-Type 将文件内容作为 POST 数据上载到任何标准 IBM MQ Managed File Transfer 代理程序。该请求应该提交到包含以下格式目标代理和文件目标的位置：`/fte/file/agent/agent_name@queue_manager/filepath`。您可以使用第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』中描述的定制 HTTP 头来修改文件传输请求参数。

使用 Web 网关提交文件传输请求时，系统会检查您在应用程序服务器环境中的用户标识是否映射到 WebSphere MQ 消息描述符 (MQMD) 用户标识。应用程序服务器用户标识 (Web 用户标识) 和 MQMD 用户标识之间的映射由 Web 网关管理员创建。有关更多详细信息，请参阅主题第 338 页的『示例：将 Web 用户标识映射到 MQMD 用户标识』。如果没有为您的 Web 用户标识定义 MQMD 用户标识，那么将使用 `defaultMQMDUserID` servlet 初始化参数的值。该参数在部署 Web 网关应用程序期间定义。

使用以下示例将文本文件传输到目标代理 ACCOUNTS 上目标文件路径 `destination-root-path/temp` 下的目标文件 `myfile.txt` 中。使用 MD5 校验和来检查所传输文件的完整性。该文件的内容为：

```
Account No, Balance
123456, 100.00
234567, 1022.00
345678, 2801.00
456789, 16.75
```

托管 IBM MQ Managed File Transfer Web 网关的服务器为 `example.com`。

过程

1. 使用以下格式创建 HTTP 请求：

```
POST HTTP/1.1 /fte/file/agent/ACCOUNTS@QM/temp
Host: example.com
Content-Type: multi-part/form-data; boundary=Aa6b74
x-fte-checksum: MD5

--Aa6b74
Content-Disposition: form-data; name="files"; filename="myfile.txt"
Content-Type: text/plain

Account No, Balance
123456, 100.00
234567, 1022.00
345678, 2801.00
456789, 16.75
--Aa6b74
```

2. 向 Web 网关提交该请求。Web 网关使用以下格式返回 HTTP 响应。

```
HTTP/1.1 200 OK
Server: WAS/6.0
```



```
Content-Length: 0
x-fte-id: 4d63c28ae6e72eb9c51cd812736acd4362ef5

<transfers>
  <submission id="4d63c28ae6e72eb9c51cd812736acd4362ef5">
  </submission>
</transfers>
```

x-fte-id 的值是传输标识。可以在 HTTP 请求中使用此传输标识来获取有关传输状态的信息。有关示例请求，请参阅主题第 315 页的『示例：使用 HTTP 请求查看文件传输的状态』。

相关参考

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

示例：使用 HTTP 请求查看文件传输的状态

通过使用 IBM MQ Managed File Transfer Web 网关来提交请求，可以查看文件传输的状态。Web 网关将返回 XML 格式的信息，其中描述了指定传输的当前状态。要使用 Web 网关来查看文件传输的状态，您的 IBM MQ Managed File Transfer 网络中必须存在数据库记录器。

关于此任务

成功的请求将返回 HTTP 状态码 200 以及用于描述传输当前状态的 XML 有效内容。可使用该 XML 来查看传输的详细信息，包括传输状态、传输标识、源和目标代理详细信息以及有关传输的源和目标文件的信息。

如果是由您启动上载或者您拥有文件要传输到的文件空间，那么您可以查看文件传输的状态。如果您的用户标识与 IBM MQ Managed File Transfer 安全角色 wmqfte-audit 或 wmqfte-admin 相关联，那么您可以查看 IBM MQ Managed File Transfer 网络中所有文件传输的状态。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 example.com，并且通过 Web 浏览器（将自身标识为 mozilla）提交了 HTTP 请求。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /transfer/414d51205245444841542e434f4f5244ed60b44b03310020
Host: example.com
User-Agent: mozilla
```

该 URL 的最后一部分是要查看的传输的有效十六进制 IBM MQ Managed File Transfer 传输标识（长度为 48 个字符）。

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 200 OK
Server: WAS/6.0
Content-Length: 1664
Content-type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<transfers>
  <transfer start-time="2010-04-01T13:10:04.209+01:00" status="Complete"
    id="414d51205245444841542e434f4f5244ed60b44b03310020">
    <source>
      <agent qmgr="REDHAT.SOURCE.QM" name="REDHAT.SOURCE.AGENT" />
      <metadata>
        <key value="REDHAT.SOURCE.AGENT" name="com.ibm.wmqfte.SourceAgent" />
        <key value="REDHAT.DEST.AGENT" name="com.ibm.wmqfte.DestinationAgent" />
        <key value="192.168.243.133" name="com.ibm.wmqfte.OriginatingHost" />
        <key value="fteuser" name="com.ibm.wmqfte.MqmdUser" />
        <key value="414d51205245444841542e434f4f5244ed60b44b03310020"
          name="com.ibm.wmqfte.TransferId" />
      </metadata>
    </source>
  </transfer>
</transfers>
```

```

        <key value="fteuser" name="com.ibm.wmqfte.OriginatingUser" />
      </metadata>
    </source>
    <destination>
      <agent qmgr="REDHAT.SOURCE.QM" name="REDHAT.SOURCE.AGENT" />
      <metadata>
        <key value="REDHAT.SOURCE.AGENT" name="com.ibm.wmqfte.SourceAgent" />
        <key value="REDHAT.DEST.AGENT" name="com.ibm.wmqfte.DestinationAgent" />
        <key value="fteuser" name="com.ibm.wmqfte.MqmdUser" />
        <key value="192.168.243.133" name="com.ibm.wmqfte.OriginatingHost" />
        <key value="fteuser" name="com.ibm.wmqfte.OriginatingUser" />
        <key value="414d51205245444841542e434f4f5244ed60b44b03310020"
          name="com.ibm.wmqfte.TransferId" />
      </metadata>
    </destination>
    <stats retry-count="0" file-warnings="0" file-failures="0"
      bytes-transferred="67" />
    <transfer-set>
      <file result-code="0" mode="text">
        <source-file name="/home/fteuser/accounts.txt">
          <attribute-values last-modified="2010-03-17T16:55:17.000Z"
            file-size="67" disposition="leave" checksum-method="none" />
        </source-file>
        <destination-file name="/tmp/accounts.txt">
          <attribute-values last-modified="2010-04-01T13:10:04.000+01:00"
            file-size="67" exists-action="error" checksum-method="none" />
        </destination-file>
      </file>
    </transfer-set>
  </transfer>
</transfers>

```

无效请求将返回 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。要确定错误原因，请参阅对 [Web 网关进行故障诊断](#)。

相关参考

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

示例：使用 HTTP 请求查询多个文件传输

通过 IBM MQ Managed File Transfer Web 网关提交请求，可查询多个文件传输的状态。Web 网关会以 XML 或 JSON 格式返回信息，其中描述与查询匹配的传输的状态。

关于此任务

您可以创建 URI 查询来请求与查询匹配的所有传输的传输信息。您可以按照传输关联的详细信息对传输进行查询，这些详细信息包括源代理、目标代理、源文件、目标文件、传输状态、元数据、传输开始时间、传输结束时间以及作业名。您可以对代理、状态、开始时间、结束时间或作业名返回的传输信息进行排序，并且可以指定要返回的结果数。成功的请求会返回 HTTP 状态码 200 以及描述与查询匹配的传输的状态的有效内容。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 example.com。查询请求符合以下条件的信息：

- 来自在 endbefore=2010-08-26T13:00:00 查询指定的时间 2010 年 8 月 26 日（星期四）下午 1 点 (UTC) 之前完成的传输
- 来自源代理或目标代理为 agent=AGENT_TITAN 查询指定的 AGENT_TITAN 的传输
- 根据 sortby=jobname 和 sort=ascending 查询中指定的值，按作业名的升序排序
- 根据 count=3 查询中指定的值，仅包含与全部查询匹配的前三个传输
- 它以 JSON 格式返回，由 Accept: application/json 头指定。

有关查询参数的更多信息，请参阅第 913 页的『查询参数』。有关用于对结果进行排序的参数的更多信息，请参阅第 916 页的『结果格式参数』。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 example.com，并且通过 Web 浏览器（将自身标识为 mozilla）提交了 HTTP 请求。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /transfer/?endbefore=2010-08-26T13:00:00&agent=AGENT_TITAN
&sortby=jobname&sort=ascending&count=3
Host: example.com
User-Agent: mozilla
Accept: application/json
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
{
  "transfers" : {
    "transfer" : {
      "end-time" : "2010-08-23T14:13:03.260Z",
      "status" : "Complete",
      "start-time" : "2010-08-23T14:12:39.076Z",
      "id" : "414d51205745422e46544520202020c1a1a34b03720120",
      "result" : {
        "code" : "0",
        "text" : "BFGRP0032I: The file transfer request has successfully completed."
      }
    },
    "destination" : {
      "metadata" : {
        "key" : [
          {
            "name" : "com.ibm.wmqfte.JobName",
            "value" : "ALPHA"
          },
          {
            "name" : "com.ibm.wmqfte.SourceAgent",
            "value" : "AGENT_TITAN"
          },
          {
            "name" : "com.ibm.wmqfte.DestinationAgent",
            "value" : "AGENT_MIMAS"
          },
          {
            "name" : "com.ibm.wmqfte.MqmdUser",
            "value" : "rich"
          },
          {
            "name" : "com.ibm.wmqfte.OriginatingHost",
            "value" : "iceman.example.com."
          },
          {
            "name" : "com.ibm.wmqfte.OriginatingUser",
            "value" : "rich"
          },
          {
            "name" : "com.ibm.wmqfte.TransferId",
            "value" : "414d51205745422e46544520202020c1a1a34b03720120"
          }
        ]
      }
    },
    "agent" : {
      "name" : "AGENT_MIMAS",
      "qmgi" : "QM_SATURN"
    }
  }
}
```

```

}
"stats" : {
  "bytes-transferred" : "259354303",
  "retry-count" : "0",
  "file-warnings" : "0",
  "file-failures" : "0"
}
"transfer-set" : {
  "file" : {
    "result-code" : "0",
    "mode" : "text",
    "source-file" : {
      "name" : "\\home\\rich\\file.zip",
      "attribute-values" : {
        "last-modified" : "2010-08-19T14:16:57.000Z",
        "checksum-value" : "98611a272a27d373f92d73a08cf0d4f4",
        "checksum-method" : "MD5",
        "file-size" : "259354303",
        "disposition" : "leave"
      }
    }
  }
  "destination-file" : {
    "name" : "\\tmp\\file.zip",
    "attribute-values" : {
      "exists-action" : "error",
      "last-modified" : "2010-08-23T14:13:02.000Z",
      "checksum-value" : "98611a272a27d373f92d73a08cf0d4f4",
      "checksum-method" : "MD5",
      "file-size" : "259354303"
    }
  }
}
"source" : {
  "metadata" : {
    "key" : [
      {
        "name" : "com.ibm.wmqfte.JobName",
        "value" : "ALPHA"
      },
      {
        "name" : "com.ibm.wmqfte.SourceAgent",
        "value" : "AGENT_TITAN"
      },
      {
        "name" : "com.ibm.wmqfte.DestinationAgent",
        "value" : "AGENT_MIMAS"
      },
      {
        "name" : "com.ibm.wmqfte.OriginatingHost",
        "value" : "iceman.example.com."
      },
      {
        "name" : "com.ibm.wmqfte.MqmdUser",
        "value" : "rich"
      },
      {
        "name" : "com.ibm.wmqfte.TransferId",
        "value" : "414d51205745422e46544520202020c1a1a34b03720120"
      },
      {
        "name" : "com.ibm.wmqfte.OriginatingUser",
        "value" : "rich"
      }
    ]
  }
}
"agent" : {

```

```

        "name" : "AGENT_TITAN",
        "qmgr" : "QM_SATURN"
    }
}
}
}
"transfer" : {
    "end-time" : "2010-08-25T15:20:03.260Z",
    "status" : "Complete",
    "start-time" : "2010-08-25T15:19:39.076Z",
    "id" : "414d51205745422e4654452020202020c1a1a34b03720120",
    "result" : {
        "code" : "0",
        "text" : "BFGRP0032I: The file transfer request has successfully completed."
    }
    "destination" : {
        "metadata" : {
            "key" : [
                {
                    "name" : "com.ibm.wmqfte.JobName",
                    "value" : "BRAVO"
                },
                {
                    "name" : "com.ibm.wmqfte.SourceAgent",
                    "value" : "AGENT_RHEA"
                },
                {
                    "name" : "com.ibm.wmqfte.DestinationAgent",
                    "value" : "AGENT_TITAN"
                },
                {
                    "name" : "com.ibm.wmqfte.MqmdUser",
                    "value" : "rich"
                },
                {
                    "name" : "com.ibm.wmqfte.OriginatingHost",
                    "value" : "iceman.example.com."
                },
                {
                    "name" : "com.ibm.wmqfte.OriginatingUser",
                    "value" : "rich"
                },
                {
                    "name" : "com.ibm.wmqfte.TransferId",
                    "value" : "414d51205745422e4654452020202020c1a1a34b03720120"
                }
            ]
        }
        "agent" : {
            "name" : "AGENT_TITAN",
            "qmgr" : "QM_SATURN"
        }
    }
    "stats" : {
        "bytes-transferred" : "259354303",
        "retry-count" : "0",
        "file-warnings" : "0",
        "file-failures" : "0"
    }
    "transfer-set" : {
        "file" : {
            "result-code" : "0",
            "mode" : "text",
            "source-file" : {
                "name" : "\\home\\rich\\file2.zip",
                "attribute-values" : {
                    "last-modified" : "2010-08-19T14:16:57.000Z",

```

```

        "checksum-value" : "98611a272a27d373f92d73a08cf0d4f4",
        "checksum-method" : "MD5",
        "file-size" : "259354303",
        "disposition" : "leave"
    }
}
    "destination-file" : {
        "name" : "\\tmp\\file2.zip",
        "attribute-values" : {
            "exists-action" : "error",
            "last-modified" : "2010-08-25T15:120:02.000Z",
            "checksum-value" : "98611a272a27d373f92d73a08cf0d4f4",
            "checksum-method" : "MD5",
            "file-size" : "259354303"
        }
    }
}
}
    "source" : {
        "metadata" : {
            "key" : [
                {
                    "name" : "com.ibm.wmqfte.JobName",
                    "value" : "BRAVO"
                },
                {
                    "name" : "com.ibm.wmqfte.SourceAgent",
                    "value" : "AGENT_RHEA"
                },
                {
                    "name" : "com.ibm.wmqfte.DestinationAgent",
                    "value" : "AGENT_TITAN"
                },
                {
                    "name" : "com.ibm.wmqfte.OriginatingHost",
                    "value" : "iceman.example.com."
                },
                {
                    "name" : "com.ibm.wmqfte.MqmdUser",
                    "value" : "rich"
                },
                {
                    "name" : "com.ibm.wmqfte.TransferId",
                    "value" : "414d51205745422e46544520202020c1a1a34b03720120"
                },
                {
                    "name" : "com.ibm.wmqfte.OriginatingUser",
                    "value" : "rich"
                }
            ]
        }
        "agent" : {
            "name" : "AGENT_RHEA",
            "qmgr" : "QM_SATURN"
        }
    }
}
    "transfer" : {
        "end-time" : "2010-08-21T14:13:03.260Z",
        "status" : "Complete",
        "start-time" : "2010-08-21T14:12:39.076Z",
        "id" : "414d51205745422e46544520202020c1a1a34b03720120",
        "result" : {
            "code" : "0",
            "text" : "BFGRP0032I: The file transfer request has successfully completed."
        }
        "destination" : {

```



```

"metadata" : {
  "key" : [
    {
      "name" : "com.ibm.wmqfte.JobName",
      "value" : "CHARLIE"
    },
    {
      "name" : "com.ibm.wmqfte.SourceAgent",
      "value" : "AGENT_TITAN"
    },
    {
      "name" : "com.ibm.wmqfte.DestinationAgent",
      "value" : "AGENT_DIONE"
    },
    {
      "name" : "com.ibm.wmqfte.MqmdUser",
      "value" : "rich"
    },
    {
      "name" : "com.ibm.wmqfte.OriginatingHost",
      "value" : "iceman.example.com."
    },
    {
      "name" : "com.ibm.wmqfte.OriginatingUser",
      "value" : "rich"
    },
    {
      "name" : "com.ibm.wmqfte.TransferId",
      "value" : "414d51205745422e4654452020202020c1a1a34b03720120"
    }
  ]
}

"agent" : {
  "name" : "AGENT_DIONE",
  "qmgr" : "QM_SATURN"
}

"stats" : {
  "bytes-transferred" : "259354303",
  "retry-count" : "0",
  "file-warnings" : "0",
  "file-failures" : "0"
}

"transfer-set" : {
  "file" : {
    "result-code" : "0",
    "mode" : "text",
    "source-file" : {
      "name" : "\\home\\rich\\file3.zip",
      "attribute-values" : {
        "last-modified" : "2010-08-19T14:16:57.000Z",
        "checksum-value" : "98611a272a27d373f92d73a08cf0d4f4",
        "checksum-method" : "MD5",
        "file-size" : "259354303",
        "disposition" : "leave"
      }
    }
  },
  "destination-file" : {
    "name" : "\\tmp\\file3.zip",
    "attribute-values" : {
      "exists-action" : "error",
      "last-modified" : "2010-08-21T14:13:02.000Z",
      "checksum-value" : "98611a272a27d373f92d73a08cf0d4f4",
      "checksum-method" : "MD5",
      "file-size" : "259354303"
    }
  }
}
}

```

```

    "source" : {
      "metadata" : {
        "key" : [
          {
            "name" : "com.ibm.wmqfte.JobName",
            "value" : "CHARLIE"
          },
          {
            "name" : "com.ibm.wmqfte.SourceAgent",
            "value" : "AGENT_TITAN"
          },
          {
            "name" : "com.ibm.wmqfte.DestinationAgent",
            "value" : "AGENT_DIONE"
          },
          {
            "name" : "com.ibm.wmqfte.OriginatingHost",
            "value" : "iceman.example.com."
          },
          {
            "name" : "com.ibm.wmqfte.MqmdUser",
            "value" : "rich"
          },
          {
            "name" : "com.ibm.wmqfte.TransferId",
            "value" : "414d51205745422e46544520202020c1a1a34b03720120"
          },
          {
            "name" : "com.ibm.wmqfte.OriginatingUser",
            "value" : "rich"
          }
        ]
      }
    },
    "agent" : {
      "name" : "AGENT_TITAN",
      "qmgr" : "QM_SATURN"
    }
  }
}

```

相关参考

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

示例：列出文件空间中的所有文件

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可列出文件空间的内容。Web 网关会以 XML 或 JSON 格式返回响应，其中列出文件空间的内容。如果您是文件空间的所有者或具有安全角色 wmqfte-admin，那么您有权列出文件空间的内容。

关于此任务

成功的请求会返回 HTTP 状态码 200 以及列出了文件空间中前 100 个文件的有效内容。该响应会以 XML（缺省值）或 JSON 格式返回，具体取决于在请求中指定的“Accept”头。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 example.com，并且通过 Web 浏览器（将自身标识为 mozilla）提交了 HTTP 请求。要列出的文件空

间的名称是“john”，其中包含两个文件。头“Accept: application/xml”指定 Web 网关应以 XML 格式返回结果。有关文件空间列表请求返回的格式的更多信息，请参阅第 925 页的『文件空间查询的响应格式』。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /fileSpace/john
Host: example.com
User-Agent: mozilla
Accept: application/xml
```

2. 向 Web 网关提交该请求。

结果

Web 网关将返回以下格式的 HTTP 响应：

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<fileSpaces xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="WebFileSpaceList.xsd">
  <fileSpace size="2" name="john">
    <file fileLink="/wmqfte/fileSpace/john/
      414d51205745422e46544520202020c1a1a34b03720120/filename"
      fsLocation="/var/ibm/WMQFTE/web/fte/transfer/
        414d51205745422e46544520202020c1a1a34b03720120/file-0"
      transferLink="/wmqfte/transfer/414d51205745422e46544520202020c1a1a34b03720120"
      transferID="414d51205745422e46544520202020c1a1a34b03720120">
      <attribute-values mode="text" created="2010-08-26T11:45:02.000Z" size="259354303"
        checksum-value="98611a272a27d373f92d73a08cf0d4f4" checksum-
method="MD5"/>
    </file>
    <file fileLink="/wmqfte/fileSpace/john/
      414d51205745422e46544520202020c1a1a34b06520120/filename"
      fsLocation="/var/ibm/WMQFTE/web/fte/transfer/
        414d51205745422e46544520202020c1a1a34b06520120/file-0"
      transferLink="/wmqfte/transfer/414d51205745422e46544520202020c1a1a34b06520120"
      transferID="414d51205745422e46544520202020c1a1a34b06520120">
      <attribute-values mode="text" created="2010-08-26T12:15:02.260Z" size="259554303"
        checksum-value="98611a272a27d37bf22d73a08cf0d4f4" checksum-
method="MD5"/>
    </file>
  </fileSpace>
</fileSpaces>
```

相关参考

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

第 925 页的『文件空间查询的响应格式』

从 IBM MQ Managed File Transfer 服务 Web 网关请求文件空间中的部分或全部文件的列表时，将根据您使用 Accept: 头指定的内容以 JSON 或 XML 格式返回响应。

示例：列出文件空间中的特定文件子集

通过向 IBM MQ Managed File Transfer Web 网关提交包含查询的 HTTP 请求，可查询文件空间的内容。Web 网关会以 XML 或 JSON 格式返回响应，其中仅描述文件空间中与查询匹配的文件。

关于此任务

您可以将查询附加到 HTTP 请求，以请求有关文件空间中与查询匹配的文件的文件的信息。您可以按照文件关联的详细信息来查询文件，这些详细信息包括始发用户、传输开始时间、传输结束时间以及将文件发送到文件空间的传输的标识。您还可以指定要返回的结果数。

成功的请求会返回 HTTP 状态码 200 以及描述与查询匹配的文件的有关内容。您可以请求以 XML 或 JSON 格式返回文件的详细信息。还可以编写 Web 应用程序来解析响应的内容，并以适当的格式呈现给 Web 用户。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 `example.com`。请求该信息的用户有权访问要查询的文件空间。查询请求以 JSON 格式返回的信息，由 `accept=json` 查询指定。该查询请求满足以下条件的文件的列表：

- 文件位于文件空间 `james` 中。
- 用户 `bob` (由 `originatoruser=bob` 查询指定) 已将这些文件发送到文件空间。
- 这些文件在 2010 年 8 月 26 日 13:00 (UTC) 之后发送到文件空间，由 `startafter=2010-08-26T13:00` 查询指定。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /fileSpace/james/?originatoruser=bob&startafter=2010-08-26T13:00&accept=json
Host: example.com
User-Agent: mozilla
```

2. 向 Web 网关提交该请求。Web 网关使用以下格式返回 HTTP 响应。在该示例中，只有一个文件与查询匹配。

```
{
  "fileSpaces" : {
    "fileSpace" : {
      "name" : "james",
      "size" : "1",
      "file" : {
        "transferLink" : "\wmqfte\transfer\414d51205745422e46544520202020c1a1a34b03720120",
        "fileLink" : "\wmqfte\fileSpace\james\414d51205745422e46544520202020c1a1a34b03720120\wibble",
        "name" : "\tmp\bobs_file.zip",
        "transferID" : "414d51205745422e46544520202020c1a1a34b03720120",
        "attribute-values" : {
          "checksum-value" : "98611a272a27d373f92d73a08cf0d4f4",
          "checksum-method" : "none",
          "time" : "2010-08-26T14:13:02.000Z",
          "file-size" : "259354303",
          "mode" : "text"
        }
      }
    }
  }
}
```

相关参考

第 925 页的『文件空间查询的响应格式』

从 IBM MQ Managed File Transfer 服务 Web 网关请求文件空间中的部分或全部文件的列表时，将根据您使用 `Accept` 头指定的内容以 JSON 或 XML 格式返回响应。

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 `/wmqfte`。

第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

示例：从文件空间检索文件

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可从文件空间检索文件。Web 网关提供使用 HTTP 协议下载文件的功能。

关于此任务

要从文件空间下载文件，您必须是文件空间的所有者或具有安全角色 `wmqfte-admin`。成功的请求会返回 HTTP 状态码 200 以及下载的文件。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 `example.com`。下载的文件是 `Accounts.csv`，将文件发送到文件空间的传输的标识是 `4142452b345f4d2e3c2a333d4ed3e4de43453bc2344a2020`。包含该文件的文件空间的名称是 `john`，且请求该信息的用户有权访问该文件空间。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /fileSPACE/john/4142452b345f4d2e3c2a333d4ed3e4de43453bc2344a2020/Accts.csv
Host: example.com
User-Agent: mozilla
```

2. 向 Web 网关提交该请求。Web 网关会在 HTTP 响应中返回文件。

在 HTTP 响应中设置了以下头：

- Content-Type: application/x-download
- Content-MD5: 98611a272a27d373f92d73a08cf0d4f4
- Content-Disposition: attachment; filename="Accts.csv"
- Content-Length: 8786

相关参考

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 `/wmqfte`。

第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

示例：从文件空间中删除文件

可以通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，从文件空间中删除文件。如果将头 `x-fte-include-file-in-response` 设置为 `true`，那么将通过 Web 网关在 HTTP 响应中返回文件内容。

关于此任务

成功的删除请求将返回 HTTP 状态码 200 和（如果已在请求中指定）被删除文件的内容。如果提交请求的用户不是文件空间的所有者，那么请求将失败。

注：安全角色 `wmqfte-admin` 可以从文件空间中删除文件，但不能接收被删除文件的内容。如果安全角色为 `wmqfte-admin` 的用户尝试删除文件并请求文件内容，那么请求将因资源错误而失败。有关更多信息，请参阅第 108 页的『Web 网关的用户角色』。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 `example.com`。文件空间的名称为 `jack`，它包含文件 `report.txt`，并且请求文件删除的用户是文件空间的所有者。传输标识 `414d5120514d5f67617265746862202067732c4c20c25a03` 是将文件放入文件空间的传输的十六进制标识，在您列出文件空间的内容时将返回该标识。有关文件空间查询响应格式的更多信息，请参阅第 925 页的『文件空间查询的响应格式』。

头 `x-fte-include-file-in-response:true` 指定在响应主体中返回 `report.txt` 的内容。如果未指定该头的内容，那么缺省值为 `false`，将删除该文件，但不会返回其内容。

过程

1. 创建以下格式的 HTTP 请求：

```
DELETE HTTP/1.1 /filesystem/jack/414d5120514d5f67617265746862202067732c4c20c25a03/report.txt
Host: example.com
User-Agent: mozilla
x-fte-include-file-in-response:true
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应:

```
HTTP/1.1 200 OK
Server: WAS/6.0
Content-Length: 1762
Content-MD5: 9608f0d8cdcb804d185ab3cb959dba6f
Content-type: text/plain; charset=Cp1252
Content-Disposition: attachment; filename="report.txt"

Account No, Balance
123456, 100.00
234567, 1022.00
345678, 2801.00
456789, 16.75
```

相关参考

第 108 页的『Web 网关的用户角色』

IBM MQ Managed File Transfer 已定义多个不同的角色，用于控制用户可执行的操作。

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

示例：使用 HTML 表单发送文件

通过使用 IBM MQ Managed File Transfer Web 网关来提交请求，可以将单个文本文件发送到目标文件系统。

关于此任务

此任务演示了如何使用 HTML 表单将文件传输请求提交到 Web 网关。使用 HTML 表单是提交 HTTP 请求的一种备用方法，如第 314 页的『示例：使用 HTTP 请求发送文件』中所述。

以下示例使用了几个可选的 HTML 表单字段。有关使用 HTML 表单字段的更多信息，请参阅第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』。

过程

1. 创建包含以下格式表单的 HTML 文件:

```
<form enctype="multipart/form-data"
  action="http://example.org/wmqfte/file/agent/AGENT1@QM1/webuploads"
  method="POST">
  <input type="HIDDEN" name="dest-exists-action" value="overwrite"/>
  <input type="HIDDEN" name="type" value="text"/>
  <input type="HIDDEN" name="jobname" value="TEST"/>
  <input type="HIDDEN" name="priority" value="1"/>
  <input type="HIDDEN" name="checksum" value="NONE"/>
  <input type="HIDDEN" name="metadata" value="fred=awesome,bob=cool"/>
  <input type="HIDDEN" name="metadata" value="lewis=fast,niall=slow"/>
  <input type="HIDDEN" name="postdest"
    value="[command=D:\postdest.cmd,type=executable,successsrc=0]"/>
  <input type="HIDDEN" name="postdest-args" value="[fred]"/>
  File: <input type="FILE" name="file"/>
  <input type="submit" name="Upload" value="Upload" />
</form>
```

此示例中使用的 dest-exists-action 表单字段是 V 7.5.0.2 的新增内容。dest-exists-action 将替换 action 表单字段，不推荐用于将来的发行版，但 7.5.0.2 仍支持此字段。

2. 在 Web 浏览器中打开该 HTML 文件。
3. 在文件字段中输入文件名，或单击[浏览](#)以浏览至该文件。
4. 单击[上传](#)以提交上传请求。Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 200 OK
Server: WAS/6.0
Content-Length: 0
x-fte-id: 4d63c28ae6e72eb9c51cd812736acd4362ef5

<transfers>
  <submission id="4d63c28ae6e72eb9c51cd812736acd4362ef5">
  </submission>
</transfers>
```

x-fte-id 的值是传输标识。可以在 HTTP 请求中使用此传输标识来获取有关传输状态的信息。有关示例请求，请参阅主题第 315 页的[『示例：使用 HTTP 请求查看文件传输的状态』](#)。

管理 IBM MQ Managed File Transfer 服务 Web 网关

您可以创建和删除文件空间，控制对个别文件空间具有访问权的用户。

可以采用以下方式管理 Web 网关：

- 使用 Web 网关管理控制台
- 使用 RESTful 管理 API，并手动构造 HTTP 请求

本部分中的示例演示了如何创建 HTTP 请求以管理 Web 网关工件。要了解有关这些请求的格式和 Web 网关管理 API 的更多信息，请参阅第 930 页的[『Web 网关管理 API 引用』](#)。

这些示例针对具有管理权限的用户。如果您正在查找对无管理权限的用户使用 Web 网关的示例（例如，在文件空间中上传文件或查询文件），请参阅主题第 312 页的[『使用 IBM MQ Managed File Transfer 服务 Web 网关』](#)。

在配置或使用 Web 网关之前，请参阅第 305 页的[『Web 网关的方案』](#)和第 307 页的[『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)。这些主题解释了 Web 网关的用途和组件。

相关概念

第 327 页的[『Web 网关管理控制台』](#)

IBM MQ Managed File Transfer 随附的 Web 网关管理控制台为您提供图形界面，用于管理文件空间和用户映射。如果具有安全角色 `wmqfte-admin`，那么可以使用管理控制台完成管理任务。

第 329 页的[『用于管理的示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

第 339 页的[『文件空间』](#)

文件空间是与 Web 网关用户关联的文件存储器的保留区域。文件空间具有配额存储量。文件空间的访问权受限于有权对其执行读写的文件空间。

相关参考

第 930 页的[『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

Web 网关管理控制台

IBM MQ Managed File Transfer 随附的 Web 网关管理控制台为您提供图形界面，用于管理文件空间和用户映射。如果具有安全角色 `wmqfte-admin`，那么可以使用管理控制台完成管理任务。

使用管理控制台

将 Web 网关部署到应用程序服务器之后，可以通过打开 Web 浏览器并输入 `http://host:port/wmqfteconsole` 来访问管理控制台。如果在部署 Web 网关时已更改上下文根缺省值 `wmqfteconsole`，那么必须使用这个值来代替 `wmqfteconsole`。

如果正在使用 WebSphere Application Server Community Edition, 那么可能会看到以下错误:
`ssl_error_no_cypher_overlap`。要纠正此问题, 请将 **TomcatWebSSLConnector** 的 **sslProtocol**
设置的值更改为 **SSL**, 然后重新启动连接器。

可以使用管理控制台执行的任务

您可以使用 Web 网关管理控制台管理两种类型的资源: 文件空间和用户映射。您可以使用管理控制台执行以下任务:

创建文件空间

您可以通过单击**文件空间**选项卡, 然后单击**添加**, 来创建文件空间。

编辑文件空间的属性

您可以通过单击**文件空间**选项卡, 然后单击**编辑**, 来编辑文件空间的属性。您可以编辑的属性有: 配额、已授权用户和未授权用户。

除去文件空间

您可以通过单击**文件空间**选项卡, 然后单击**除去**, 来除去文件空间。在删除文件空间之前, 确保未与文件空间进行任何传输。

检查所有文件空间的完整性

您可以通过单击**文件空间**选项卡, 然后单击**检查完整性**, 来检查与 Web 网关关联的所有文件空间的完整性。

将 Web 用户标识映射到 MQMD 用户标识

您可以通过单击**MQMD 用户标识**选项卡, 然后单击**添加**, 来将 Web 用户标识映射到 MQMD 用户标识。如果未在 Web 用户与 MQMD 用户标识之间指定映射, 那么将使用通过 **defaultMQMDUserID** 参数指定的值。

相关概念

第 327 页的『[管理 IBM MQ Managed File Transfer 服务 Web 网关](#)』
您可以创建和删除文件空间, 控制对个别文件空间具有访问权的用户。

第 307 页的『[Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑](#)』
使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理, 并使用 HTTP 客户机检索传输的状态。

相关任务

第 329 页的『[示例: 创建文件空间](#)』
必须先为用户创建文件空间, 才能将文件传输到用户文件空间。您可以使用 IBM MQ Managed File Transfer Web 网关创建文件空间。

第 336 页的『[示例: 删除文件空间](#)』
可以通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求, 删除现有的文件空间。如果正在对文件空间进行文件传输, 那么将不会删除文件空间。

第 331 页的『[示例: 修改文件空间配置](#)』
通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求, 可以修改现有文件空间。如果您的用户帐户具有关联的必要安全角色, 那么可以更改文件空间配额以及有权访问该文件空间的用户列表。

第 335 页的『[示例: 检查所有文件空间的完整性](#)』
通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求, 可以检查所有文件空间的完整性。例如, 如果要在数据丢失之后复原文件系统, 那么可以检查文件空间中的文件是否位于文件系统上的正确位置。Web 网关将返回 XML 或 JSON 格式的响应, 其中列出文件空间的名称以及一个用于指示文件空间条目是否与文件系统中的文件相匹配的属性。

第 338 页的『[示例: 将 Web 用户标识映射到 MQMD 用户标识](#)』
将文件上载提交到 IBM MQ Managed File Transfer Web 网关时, Web 网关会确定传输要使用的 IBM MQ 消息描述符 (MQMD) 用户标识。您可以使用 Web 网关来定义 Web 用户标识和 MQMD 用户标识之间的映射集。

用于管理的示例 HTTP 流

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

您可以使用各种技术向 Web 网关提交请求或解释来自 Web 网关的响应。例如，您可以写 Web 应用程序。有关 Web 网关随附的示例 Web 应用程序的信息，请参阅第 353 页的『样本 Web 页面』。

如果要使用 Web 应用程序与 Web 网关通信，可以使用 HTML 格式或 Javascript XMLHttpRequest 函数。要上载文件，必须使用 HTML 格式，因为浏览器会出于安全原因阻止 Javascript 从本地系统访问文件。如果您愿意，可以通过 Javascript 控制和提交格式。要请求传输状态，XMLHttpRequest 最有可能适用，虽然也可用其他技术；例如，将内容装入不可视的 iFrame 元素。

也可以采用诸如 Ruby 或 Perl 之类语言编写客户机应用程序，以与 Web 网关 API 通信。

相关任务

[第 329 页的『示例：创建文件空间』](#)

必须先为用户创建文件空间，才能将文件传输到用户文件空间。您可以使用 IBM MQ Managed File Transfer Web 网关创建文件空间。

[第 331 页的『示例：修改文件空间配置』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以修改现有文件空间。如果您的用户帐户具有关联的必要安全角色，那么可以更改文件空间配额以及有权访问该文件空间的用户列表。

[第 333 页的『示例：列出所有文件空间』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以列出所有文件空间。Web 网关将返回 XML 或 JSON 格式的响应，其中列出文件空间的名称、每个文件空间的配额以及有权和无权向每个文件空间写入内容的用户。

[第 335 页的『示例：检查所有文件空间的完整性』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以检查所有文件空间的完整性。例如，如果要在数据丢失之后复原文件系统，那么可以检查文件空间中的文件是否位于文件系统上的正确位置。Web 网关将返回 XML 或 JSON 格式的响应，其中列出文件空间的名称以及一个用于指示文件空间条目是否与文件系统中的文件相匹配的属性。

[第 336 页的『示例：删除文件空间』](#)

可以通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，删除现有的文件空间。如果正在对文件空间进行文件传输，那么将不会删除文件空间。

[第 338 页的『示例：将 Web 用户标识映射到 MQMD 用户标识』](#)

将文件上载提交到 IBM MQ Managed File Transfer Web 网关时，Web 网关会确定传输要使用的 IBM MQ 消息描述符 (MQMD) 用户标识。您可以使用 Web 网关来定义 Web 用户标识和 MQMD 用户标识之间的映射集。

示例：创建文件空间

必须先为用户创建文件空间，才能将文件传输到用户文件空间。您可以使用 IBM MQ Managed File Transfer Web 网关创建文件空间。

关于此任务

使用 Web 网关管理 API 请求创建用户文件空间。有关文件空间创建请求格式的更多信息，请参阅第 939 页的『文件空间创建或更改请求格式』。成功的请求将返回 HTTP 状态码 200。

必须具有与用户帐户关联的 `wmqfte-filespace-create` 角色或 `wmqfte-admin` 角色，才能创建文件空间。有关 Web 网关的安全角色的更多信息，请参阅第 108 页的『Web 网关的用户角色』和第 419 页的『尝试在不具有必需权限的情况下创建文件空间』。

如果具有安全角色 `wmqfte-admin`，那么还可以使用管理控制台创建文件空间。有关更多信息，请参阅第 327 页的『Web 网关管理控制台』。

以下步骤描述如何提交 POST 请求以创建文件空间。在本例中，托管 Web 网关的服务器是 `example.com`，并且提交 HTTP 请求使用的是自身标识为 `mozilla` 的 Web 浏览器。文件空间名称和拥有文件空间的用户名为 `andrew`，文件空间最多将占用文件系统上的 1,048,576 个字节。用户 `bill` 及用户名与正则表达式模式 `fte.*` 相匹配的任何用户都有权向文件空间发送文件。用户 `clive` 无权访问用户文件空间。您可以使用 Java 正则表达式来与 `authorized` 和 `unauthorized` XML 部分中的任一用户集或这两个

用户集进行模式匹配。有关更多信息，请参阅 [第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』](#)。

在以下示例中，authorized 部分中的 agent-user 条目之一将使用正则表达式 `fte.*`。该正则表达式与任何以 `fte` 开头的用户名都匹配。如果要授权除 `fteuser` 之外的以 `fte` 开头的所有用户名，可以在 unauthorized 部分中添加值为 `fteuser` 的额外 agent-user 条目。该元素优先于 `fte.*` 正则表达式，因为 unauthorized 条目和 authorized 条目求值相同时，前者将推翻后者。

在以下示例中，authorized 部分中的 agent-user 条目之一为用户名 `accounts1`。unauthorized 部分中的 agent-user 条目之一是正则表达式 `accounts*`，这将覆盖授予用户名 `accounts1` 的权限。针对该文件空间，未授权与正则表达式 `accounts*` 匹配的所有用户（包括 `accounts1`）。

过程

1. 创建以下格式的 HTTP 请求：

```
POST HTTP/1.1 /admin/filespace/andrew
Host: example.com
User-Agent: mozilla
Content-Type: application/xml
Content-Length: 266

<?xml version="1.0" encoding="UTF-8"?>
<filespaces>
  <filespace>
    <quota bytes="1048576"/>
    <writers>
      <authorized>
        <agent-user>bill</agent-user>
        <agent-user>accounts1</agent-user>
        <agent-user>fte.*</agent-user>
      </authorized>
      <unauthorized>
        <agent-user>fteuser</agent-user>
        <agent-user>accounts*</agent-user>
      </unauthorized>
    </writers>
  </filespace>
</filespaces>
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 200 OK
Server: WAS/7.0
Content-Length: 0
```

现在存在名为 `andrew` 的文件空间，可向其传输文件。用户 `andrew`、`bill` 及用户名以 `fte` 开头的任何用户（用户 `fteuser` 除外）都可以向文件空间传输文件。与正则表达式 `accounts*` 匹配的任何用户都不可以向文件空间传输文件。

有关如何向文件空间传输文件的信息，请参阅 [第 312 页的『示例：将文件传输到文件空间』](#)。

会将创建文件空间的请求记录到应用程序服务器事件日志中。有关更多信息，请参阅 [第 943 页的『文件空间管理日志记录格式』](#)。

无效请求将返回 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。要确定错误原因，请参阅 [第 412 页的『对 Web 网关进行故障诊断』](#)。

相关概念

[第 327 页的『Web 网关管理控制台』](#)

IBM MQ Managed File Transfer 随附的 Web 网关管理控制台为您提供图形界面，用于管理文件空间和用户映射。如果具有安全角色 `wmqfte-admin`，那么可以使用管理控制台完成管理任务。

[第 329 页的『用于管理的示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

[第 931 页的『用于管理 Web 网关的 HTTP 头』](#)

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

[第 939 页的『文件空间创建或更改请求格式』](#)

通过在 HTTP 请求中包含 XML 格式的内容，可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 `FileSpaceInfo.xsd`，此模式位于 WMQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

[第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』](#)

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 Connect:Direct 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 `java.util.regex` API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

示例：修改文件空间配置

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以修改现有文件空间。如果您的用户帐户具有关联的必要安全角色，那么可以更改文件空间配额以及有权访问该文件空间的用户列表。

关于此任务

使用 IBM MQ Managed File Transfer 角色 `wmqfte-admin` 和 `wmqfte-filespace-modify`，可以更改文件空间配额以及有权访问该文件空间的用户列表。有关保护 Web 网关的更多信息，请参阅[第 108 页的『Web 网关的用户角色』](#)。

如果您具有安全角色 `wmqfte-admin`，那么还可以使用管理控制台来修改文件空间。有关更多信息，请参阅[第 327 页的『Web 网关管理控制台』](#)。

如果在将文件传输到文件空间时更改了文件空间配额，那么即使这些文件传输会导致超出新配额值，这些文件传输也可能成功。但对于在更改配额后才开始的任何文件传输，它们只有在文件传输未导致超出新配额值时才会成功。

以下示例显示了如何更改文件空间配额、将用户添加到有权访问文件空间的用户列表，以及从无权访问文件空间的用户列表中移除用户。在此示例中，用于托管 Web 网关的服务器是 `example.com`。已创建的文件空间的名称为 `finlay`。文件空间的名称由 POST 请求所使用的 URI 的最后一个部分表示。

有关用于修改文件空间的 XML 请求的格式的更多信息，请参阅[第 939 页的『文件空间创建或更改请求格式』](#)。

过程

1. 如果要在现有用户列表中添加或移除用户，请在 `authorized` 和 `unauthorized` 元素上使用 `add` 操作或 `remove` 操作。例如，以下请求将两个用户添加到授权用户列表中并从未授权用户列表中移除一个用户：

```
POST HTTP/1.1 /admin/filespace/finlay
Host: example.com
User-Agent: mozilla
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<filespaces>
  <filespace>
    <quota bytes="1000000000"/>
    <writers>
      <authorized action="add">
        <agent-user>jonathan</agent-user>
        <agent-user>lauren</agent-user>
      </authorized>
      <unauthorized action="remove">
```

```

        <agent-user>marley</agent-user>
    </unauthorized>
</writers>
</filespace>
</filespaces>

```

如果要覆盖当前用户列表（而不是在现有列表中添加或删除用户），请在 `authorized` 和 `unauthorized` 元素上使用 `overwrite` 操作。例如，以下请求将覆盖当前的授权用户列表：

```

POST HTTP/1.1 /admin/filespace/finlay
Host: example.org
User-Agent: mozilla
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<filespaces>
  <filespace>
    <writers>
      <authorized action="overwrite">
        <agent-user>fte.*</agent-user>
        <agent-user>ella</agent-user>
        <agent-user>jonathan</agent-user>
        <agent-user>lauren</agent-user>
      </authorized>
    </writers>
  </filespace>
</filespaces>

```

您可以使用 Java 正则表达式来匹配多个用户名。例如，先前示例中的某个 `agent-user` 条目具有值 `fte.*`，该值将与名称以 `fte` 开头的任何用户匹配。

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```

HTTP/1.1 200 OK
Server: WAS/7.0
Content-Length: 0

```

用于修改文件空间的请求会记录到应用程序服务器的事件日志中。有关更多信息，请参阅 [第 943 页的『文件空间管理日志记录格式』](#)。

无效请求将返回 HTTP 错误代码和 WMQFTE 错误消息。要确定错误原因，请参阅 [第 412 页的『对 Web 网关进行故障诊断』](#)。

相关概念

[第 327 页的『Web 网关管理控制台』](#)

IBM MQ Managed File Transfer 随附的 Web 网关管理控制台为您提供图形界面，用于管理文件空间和用户映射。如果具有安全角色 `wmqfte-admin`，那么可以使用管理控制台完成管理任务。

[第 329 页的『用于管理的示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

[第 931 页的『用于管理 Web 网关的 HTTP 头』](#)

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

[第 939 页的『文件空间创建或更改请求格式』](#)

通过在 HTTP 请求中包含 XML 格式的内容，可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 `FileSpaceInfo.xsd`，此模式位于 WMQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

示例：列出所有文件空间

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以列出所有文件空间。Web 网关将返回 XML 或 JSON 格式的响应，其中列出文件空间的名称、每个文件空间的配额以及有权和无权向每个文件空间写入内容的用户。

关于此任务

成功的请求将返回 HTTP 状态码 200 以及用于描述最多 100 个文件空间的有效内容。

在此示例中，托管 Web 网关的服务器为 example.com。当前有三个属于用户 richard, suzanne 和 hamilton 的文件空间。当前未对文件空间 richard 执行任何文件传输。正在对文件空间 hamilton 执行一个文件传输，并对文件空间 suzanne 执行两个文件传输。请求信息的用户与安全角色 wmqfte-admin 相关联。头 Accept: application/xml 指定该查询将返回 XML 格式的结果。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /admin/filespace/  
Host: example.com  
User-Agent: mozilla  
Accept: application/xml
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
Content-Type: application/xml  
  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<filespaces xsi:noNamespaceSchemaLocation="FileSpaceInfo.xsd"  
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">  
  <filespace transfers="0" location="/mnt/gateway/richard" name="richard">  
    <quota bytes="1048576"/>  
    <writers>  
      <authorized>  
        <agent-user>charlene</agent-user>  
        <agent-user>alan</agent-user>  
      </authorized>  
    </writers>  
  </filespace>  
  <filespace transfers="2" location="/mnt/gateway/suzanne" name="suzanne">  
    <quota bytes="20489878"/>  
    <writers>  
      <authorized>  
        <agent-user>charlene</agent-user>  
        <agent-user>sammy</agent-user>  
      </authorized>  
      <unauthorized>  
        <agent-user>arnold</agent-user>  
        <agent-user>frank</agent-user>  
      </unauthorized>  
    </writers>  
  </filespace>  
  <filespace transfers="1" location="/mnt/gateway/hamilton" name="hamilton">  
    <quota bytes="666999"/>  
    <writers>  
      <authorized>  
        <agent-user>joseph</agent-user>  
      </authorized>  
      <unauthorized>  
        <agent-user>junior</agent-user>  
      </unauthorized>  
    </writers>  
  </filespace>  
</filespaces>
```

相关概念

第 329 页的『用于管理的示例 HTTP 流』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

第 930 页的『Web 网关管理 API 引用』

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

第 931 页的『用于管理 Web 网关的 HTTP 头』

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

第 937 页的『文件空间信息响应格式』

通过 IBM MQ Managed File Transfer 服务 Web 网关请求有关文件空间的定义和属性的信息时，将以 XML 格式或 JSON 格式返回响应。该 XML 响应符合模式 FileSpaceInfo.xsd，该模式位于 MQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

示例：检查文件空间中文件的完整性

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可检查文件空间中文件的完整性。例如，如果要在数据丢失之后复原文件系统，那么可以检查文件空间中的文件是否位于文件系统上的正确位置。Web 网关会以 XML 或 JSON 格式返回响应，其中列出文件空间的内容以及用于指示每个文件的完整性检查结果的附加属性。

关于此任务

成功的请求会返回 HTTP 状态码 200 以及列出了文件空间中前 100 个文件的有效内容。您可以请求以 XML 或 JSON 格式返回文件的详细信息。还可以编写 Web 应用程序来解析响应的内容，并以适当的格式呈现给 Web 用户。仅管理员有权使用完整性检查属性列出文件空间中的文件。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 example.com。要列示的文件空间的名称为 john，它包含两个文件。头 Accept: application/xml 指定查询以 XML 格式返回结果。头 x-fte-check-integrity 指定查询返回的结果中包含每个文件的额外完整性检查属性。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /fileSpace/john
Host: example.com
User-Agent: mozilla
Accept: application/xml
x-fte-check-integrity: true
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
<fileSpaces xsi:noNamespaceSchemaLocation="WebTransferStatus.xsd">
  <fileSpace size="2" name="john">
    <file fileLink="/wmqfte/fileSpace/john/
      414d51205745422e4654452020202020c1a1a34b03720120/ar5erh"
      transferLink="/wmqfte/transfer/
        414d51205745422e4654452020202020c1a1a34b03720120"
        transferID="414d51205745422e46544520202020c1a1a34b03720120"
        name="/tmp/file1.zip"
        fsLocation="/fileSpaces/john/
          414d51205745422e46544520202020c1a1a34b03720120/file-0">
      <attribute-values mode="text" time="2010-08-26T11:45:02.000Z"
file-size="259354303"
checksum-value="98611a272a27d373f92d73a08cf0d4f4"
checksum-method="none"
integrity-check-result="OK"/>
    </file>
    <file fileLink="/wmqfte/fileSpace/john/
      414d51205745422e4654452020202020c1a1a34b06520120/ar5erh"
      transferLink="/wmqfte/transfer/
        414d51205745422e46544520202020c1a1a34b06520120"
        transferID="414d51205745422e46544520202020c1a1a34b06520120"
        name="/tmp/file2.zip"
        fsLocation="/fileSpaces/john/
          414d51205745422e46544520202020c1a1a34b06520120/file-0">
      <attribute-values mode="text" time="2010-08-26T12:15:02.260Z"
file-size="259554303"
```

```
checksum-value="98611a272a27d37bf22d73a08cf0d4f4"  
checksum-method="none"  
integrity-check-result="MISSING-FILESYSTEM"/>  
</file>  
</fileSpace>  
</fileSpaces>
```

结果

该示例结果指示第一个文件已通过完整性检查。 `integrity-check-result` 属性值 `OK` 显示 Web 网关数据库中该文件存在且在文件系统中找到了匹配的文件。第二个文件的完整性检查失败。 `MISSING-FILESYSTEM` 的 `integrity-check-result` 属性值显示该文件存在于 Web 网关数据库中，但在 `fsLocation` 属性给定的位置中的文件系统上找不到该文件。在这种情况下，可能需要管理员从文件空间删除该文件，或从备份恢复文件空间目录。

有关 `integrity-check-result` 属性的可能值，请参阅 [第 937 页的『文件空间信息响应格式』](#)。

相关概念

[第 329 页的『用于管理的示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 911 页的『使用 Web 网关时的统一资源标识语法』](#)

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 `/wmqfte`。

[第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』](#)

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

[第 931 页的『用于管理 Web 网关的 HTTP 头』](#)

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

示例：检查所有文件空间的完整性

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以检查所有文件空间的完整性。例如，如果要在数据丢失之后复原文件系统，那么可以检查文件空间中的文件是否位于文件系统上的正确位置。Web 网关将返回 XML 或 JSON 格式的响应，其中列出文件空间的名称以及一个用于指示文件空间条目是否与文件系统中的文件相匹配的属性。

关于此任务

使用 Web 网关管理 API 来请求当前存在的所有文件空间的列表。成功的请求将返回 HTTP 状态码 200 以及用于描述最多 100 个文件空间的有效内容。在此示例中，托管 IBM MQ Managed File Transfer Web 网关的服务器为 `example.com`。当前有三个文件空间，分别属于用户 `richard`、`suzanne` 和 `hamilton`。请求该信息的用户与安全角色 `wmqfte-admin` 相关联。头 `Accept: application/xml` 指定该查询将返回 XML 格式的结果。头 `x-fte-check-integrity` 指定应检查每个文件空间以确保文件系统上存在匹配的目录。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /admin/filespace/  
Host: example.com  
User-Agent: mozilla  
Accept: application/xml  
x-fte-check-integrity: true
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
Content-Type: application/xml
```

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<filespaces xsi:noNamespaceSchemaLocation="FileSpaceInfo.xsd"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <filespace transfers="0" location="/mnt/gateway/richard" name="richard"
    integrity-check-result="OK">
    <quota bytes="1048576"/>
    <writers>
      <authorized>
        <agent-user>charlene</agent-user>
        <agent-user>alan</agent-user>
      </authorized>
    </writers>
  </filespace>
  <filespace transfers="2" location="/mnt/gateway/suzanne" name="suzanne"
    integrity-check-result="MISSING-FILESYSTEM">
    <quota bytes="20489878"/>
    <writers>
      <authorized>
        <agent-user>charlene</agent-user>
        <agent-user>sammy</agent-user>
      </authorized>
      <unauthorized>
        <agent-user>arnold</agent-user>
        <agent-user>frank</agent-user>
      </unauthorized>
    </writers>
  </filespace>
  <filespace transfers="1" location="/mnt/gateway/hamilton" name="hamilton"
    integrity-check-result="OK">
    <quota bytes="666999"/>
    <writers>
      <authorized>
        <agent-user>joseph</agent-user>
      </authorized>
      <unauthorized>
        <agent-user>junior</agent-user>
      </unauthorized>
    </writers>
  </filespace>
</filespaces>

```

结果

此示例结果指示结果集中的第一个和第三个文件空间已通过完整性检查。OK 的 `integrity-check-result` 属性值显示文件空间存在于 Web 网关数据库中，并且已在文件系统上找到匹配的目录。第二个文件空间未通过完整性检查。MISSING-FILESYSTEM 的 `integrity-check-result` 属性值显示文件空间存在于 Web 网关数据库中，但在文件系统上找不到 `location` 属性所指示的目录。在这种情况下，管理员可能需要删除该文件空间，或从备份复原该文件空间根目录。

如果您具有安全角色 `wmqfte-admin`，那么还可以使用管理控制台来检查所有文件空间的完整性。有关更多信息，请参阅第 327 页的『Web 网关管理控制台』。

要了解 `integrity-check-result` 属性的可能值，请参阅第 937 页的『文件空间信息响应格式』。

相关概念

第 327 页的『Web 网关管理控制台』

IBM MQ Managed File Transfer 随附的 Web 网关管理控制台为您提供图形界面，用于管理文件空间和用户映射。如果具有安全角色 `wmqfte-admin`，那么可以使用管理控制台完成管理任务。

相关任务

第 334 页的『示例：检查文件空间中文件的完整性』

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可检查文件空间中文件的完整性。例如，如果要在数据丢失之后复原文件系统，那么可以检查文件空间中的文件是否位于文件系统上的正确位置。Web 网关会以 XML 或 JSON 格式返回响应，其中列出文件空间的内容以及用于指示每个文件的完整性检查结果的附加属性。

示例：删除文件空间

可以通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，删除现有的文件空间。如果正在对文件空间进行文件传输，那么将不会删除文件空间。

关于此任务

要删除文件空间，必须具有与用户帐户关联的相应安全角色。与 IBM MQ Managed File Transfer 角色 `wmqfte-admin` 和 `wmqfte-filespace-delete` 关联的用户可以删除文件空间。有关保护 Web 网关的更多信息，请参阅第 108 页的『Web 网关的用户角色』。

如果具有安全角色 `wmqfte-admin`，那么还可以使用管理控制台删除文件空间。有关更多信息，请参阅第 327 页的『Web 网关管理控制台』。

成功删除文件空间

关于此任务

在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 `example.com`。文件空间属于用户 `richard`。当前未对文件空间 `richard` 执行任何文件传输。您可以通过列出文件空间，找出正在对 Web 网关环境中的文件空间进行的一些传输。有关更多信息，请参阅第 333 页的『示例：列出所有文件空间』。

过程

1. 要删除文件空间 `richard`，请使用以下格式创建 HTTP 请求：

```
DELETE HTTP/1.1 /admin/filespace/richard
Host: example.com
User-Agent: mozilla
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 200 OK
Server: WAS/7.0
```

将删除文件空间 `richard` 及其包含的任何文件。会将文件空间的删除记录到应用程序服务器事件日志中。有关更多信息，请参阅第 943 页的『文件空间管理日志记录格式』。

删除文件空间时可能出现的问题

关于此任务

在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 `example.com`。文件空间属于用户 `suzanne`。当前正在对文件空间 `suzanne` 进行两个传输。您可以通过列出文件空间，找出正在对 Web 网关环境中的文件空间进行的一些传输。有关更多信息，请参阅第 333 页的『示例：列出所有文件空间』。

过程

1. 要删除文件空间 `suzanne`，请使用以下格式创建 HTTP 请求：

```
DELETE HTTP/1.1 /admin/filespace/suzanne
Host: example.com
User-Agent: mozilla
```

2. 向 Web 网关提交该请求。该请求因正在对文件空间进行传输而失败。因此，您将收到来自 Web 网关的以下响应：

```
HTTP/1.1 409 Conflict
Server: WAS/7.0

BFGWI0060E: The file space 'suzanne' is currently in use, and cannot be deleted.
```

您必须等待到文件空间的传输完成，才能删除文件空间。

要确定您可能收到的任何其他错误的原因，请参阅第 412 页的『对 Web 网关进行故障诊断』。

相关概念

第 327 页的『Web 网关管理控制台』

IBM MQ Managed File Transfer 随附的 Web 网关管理控制台为您提供图形界面，用于管理文件空间和用户映射。如果具有安全角色 `wmqfte-admin`，那么可以使用管理控制台完成管理任务。

第 329 页的『用于管理的示例 HTTP 流』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 `/wmqfte`。

第 931 页的『用于管理 Web 网关的 HTTP 头』

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

示例：将 Web 用户标识映射到 MQMD 用户标识

将文件上载提交到 IBM MQ Managed File Transfer Web 网关时，Web 网关会确定传输要使用的 IBM MQ 消息描述符 (MQMD) 用户标识。您可以使用 Web 网关来定义 Web 用户标识和 MQMD 用户标识之间的映射集。

关于此任务

将 HTTP 请求提交到 Web 网关，在请求的正文中使用 XML 将 Web 用户标识映射到 MQMD 用户标识。有关 XML 格式的更多信息，请参阅第 942 页的『用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式』。成功的请求将返回 HTTP 状态码 200。

要创建映射集，您的用户帐户必须与 `wmqfte-admin` 角色关联。有关 Web 网关安全角色的更多信息，请参阅第 108 页的『Web 网关的用户角色』。

如果您具有安全角色 `wmqfte-admin`，那么还可以使用管理控制台将 Web 用户标识映射到 MQMD 用户标识。有关更多信息，请参阅第 327 页的『Web 网关管理控制台』。

以下步骤描述了如何提交创建映射集的 POST 请求。在本例中，托管 Web 网关的服务器是 `example.com`，并且提交 HTTP 请求使用的是自身标识为 `mozilla` 的 Web 浏览器。该请求包含两个用户的信息，这两个用户的 Web 用户标识分别为 `jim` 和 `rachel`。

过程

1. 创建以下格式的 HTTP 请求：

```
POST HTTP/1.1 /admin/user
Host: example.com
User-Agent: mozilla
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<users>
  <user>
    <userID>jim</userID>
    <mqmdUserID>mqjim</mqmdUserID>
  </user>
  <user>
    <userID>rachel</userID>
    <mqmdUserID>mqrachel</mqmdUserID>
  </user>
</users>
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 200 OK
```


无效请求将返回 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。要确定错误原因，请参阅第 412 页的『对 Web 网关进行故障诊断』。

结果

当用户 jim 或 rachel 中的某一位通过 Web 网关提交文件上载请求时，传输将使用相应的 MQMD 用户标识 mqjim 或 mqrachel。如果未定义 MQMD 用户标识的用户提交了文件上载请求，那么将使用 **defaultMQMDUserID** 参数的值。在这种情况下，如果 Web 网关部署期间未定义该参数，那么传输将失败。有关更多信息，请参阅第 198 页的『在 WebSphere 应用程序服务器 V7.0 中部署 Web 网关』和第 183 页的『准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关』。

相关概念

第 327 页的『Web 网关管理控制台』

IBM MQ Managed File Transfer 随附的 Web 网关管理控制台为您提供图形界面，用于管理文件空间和用户映射。如果具有安全角色 wmqfte-admin，那么可以使用管理控制台完成管理任务。

相关参考

第 942 页的『用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式』

通过向 IBM MQ Managed File Transfer Web 网关提交请求，可以创建 Web 用户标识与 IBM MQ 消息描述符 (MQMD) 用户标识之间的映射集。此 HTTP 请求必须包含以下 XML 格式的内容。

第 108 页的『Web 网关的用户角色』

IBM MQ Managed File Transfer 已定义多个不同的角色，用于控制用户可执行的操作。

第 932 页的『用于管理 Web 网关的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 IBM MQ URI 区分开。建议使用上下文根 /wmqfte。用于管理任务的 URI 通过术语 /admin 与现有 IBM MQ Managed File Transfer URI 区分开。

文件空间

文件空间是与 Web 网关用户关联的文件存储器的保留区域。文件空间具有配额存储量。文件空间的访问权受限于有权对其执行读写的文件空间。

您可以将文件从代理发送至用户的文件空间。文件存储在文件空间中，可以使用向 Web 网关 API 提交请求的 HTTP 客户机进行下载。文件空间可用于使文件对无权访问托管代理的系统的用户可用。会以与正常文件传输相同的方式记录到文件空间的传输和源自文件空间的下载。

无需文件空间即可使用 Web 网关将文件上载至 MQMFT 代理。如果要使文件对用户可用以使用 HTTP 客户机进行收集，那么需要创建文件空间。有关使用 Web 网关进行文件上载和下载的行为的更多信息，请参阅第 305 页的『Web 网关的方案』。

相关任务

第 329 页的『示例：创建文件空间』

必须先为用户创建文件空间，才能将文件传输到用户文件空间。您可以使用 IBM MQ Managed File Transfer Web 网关创建文件空间。

第 336 页的『示例：删除文件空间』

可以通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，删除现有的文件空间。如果正在对文件空间进行文件传输，那么将不会删除文件空间。

第 331 页的『示例：修改文件空间配置』

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以修改现有文件空间。如果您的用户帐户具有关联的必要安全角色，那么可以更改文件空间配额以及有权访问该文件空间的用户列表。

第 312 页的『示例：将文件传输到文件空间』

将单个文件传输到 IBM MQ Managed File Transfer 文件空间。通过在 **fteCreateTransfer** 命令中使用 **-du** 参数，可将文件空间指定为文件传输的目标。

第 322 页的『示例：列出文件空间中的所有文件』

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可列出文件空间的内容。Web 网关会以 XML 或 JSON 格式返回响应，其中列出文件空间的内容。如果您是文件空间的所有者或具有安全角色 `wmqfte-admin`，那么您有权列出文件空间的内容。

[第 334 页的『示例：检查文件空间中文件的完整性』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可检查文件空间中文件的完整性。例如，如果要在数据丢失之后复原文件系统，那么可以检查文件空间中的文件是否位于文件系统上的正确位置。Web 网关会以 XML 或 JSON 格式返回响应，其中列出文件空间的内容以及用于指示每个文件的完整性检查结果的附加属性。

[第 323 页的『示例：列出文件空间中的特定文件子集』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交包含查询的 HTTP 请求，可查询文件空间的内容。Web 网关会以 XML 或 JSON 格式返回响应，其中仅描述文件空间中与查询匹配的文件。

[第 324 页的『示例：从文件空间检索文件』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可从文件空间检索文件。Web 网关提供使用 HTTP 协议下载文件的功能。

[第 325 页的『示例：从文件空间中删除文件』](#)

可以通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，从文件空间中删除文件。如果将头 `x-fte-include-file-in-response` 设置为 `true`，那么将通过 Web 网关在 HTTP 响应中返回文件内容。

[第 333 页的『示例：列出所有文件空间』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以列出所有文件空间。Web 网关将返回 XML 或 JSON 格式的响应，其中列出文件空间的名称、每个文件空间的配额以及有权和无权向每个文件空间写入内容的用户。

[第 335 页的『示例：检查所有文件空间的完整性』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以检查所有文件空间的完整性。例如，如果要在数据丢失之后复原文件系统，那么可以检查文件空间中的文件是否位于文件系统上的正确位置。Web 网关将返回 XML 或 JSON 格式的响应，其中列出文件空间的名称以及一个用于指示文件空间条目是否与文件系统中的文件相匹配的属性。

[第 181 页的『设置数据库以用于文件空间』](#)

在可以使用文件空间之前，必须为 Web 网关设置数据库表以存储文件空间信息。您可以在现有的日志数据库中创建这些表，或创建新数据库以包含这些表。

相关参考

[第 420 页的『尝试创建已存在的文件空间』](#)

通过 IBM MQ Managed File Transfer Web 网关创建的文件空间必须具有唯一名称。如果尝试用已被使用的名称来创建文件空间，那么将被视为尝试修改该文件空间。如果您无权修改该文件空间，那么将收到 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。

示例：创建文件空间

必须先为用户创建文件空间，才能将文件传输到用户文件空间。您可以使用 IBM MQ Managed File Transfer Web 网关创建文件空间。

关于此任务

使用 Web 网关管理 API 请求创建用户文件空间。有关文件空间创建请求格式的更多信息，请参阅[第 939 页的『文件空间创建或更改请求格式』](#)。成功的请求将返回 HTTP 状态码 200。

必须具有与用户帐户关联的 `wmqfte-filespace-create` 角色或 `wmqfte-admin` 角色，才能创建文件空间。有关 Web 网关的安全角色的更多信息，请参阅[第 108 页的『Web 网关的用户角色』](#)和[第 419 页的『尝试在不具有必需权限的情况下创建文件空间』](#)。

如果具有安全角色 `wmqfte-admin`，那么还可以使用管理控制台创建文件空间。有关更多信息，请参阅[第 327 页的『Web 网关管理控制台』](#)。

以下步骤描述如何提交 POST 请求以创建文件空间。在本例中，托管 Web 网关的服务器是 `example.com`，并且提交 HTTP 请求使用的是自身标识为 `mozilla` 的 Web 浏览器。文件空间名称和拥有文件空间的用户名为 `andrew`，文件空间最多将占用文件系统上的 1,048,576 个字节。用户 `bill` 及用户名与正则表达式模式 `fte.*` 相匹配的任何用户都有权向文件空间发送文件。用户 `clive` 无权访问用户文件空

间。您可以使用 Java 正则表达式来与 `authorized` 和 `unauthorized` XML 部分中的任一用户集或这两个用户集进行模式匹配。有关更多信息，请参阅第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』。

在以下示例中，`authorized` 部分中的 `agent-user` 条目之一将使用正则表达式 `fte.*`。该正则表达式与任何以 `fte` 开头的用户名都匹配。如果要授权除 `fteuser` 之外的以 `fte` 开头的所有用户名，可以在 `unauthorized` 部分中添加值为 `fteuser` 的额外 `agent-user` 条目。该元素优先于 `fte.*` 正则表达式，因为 `unauthorized` 条目和 `authorized` 条目求值相同时，前者将推翻后者。

在以下示例中，`authorized` 部分中的 `agent-user` 条目之一为用户名 `accounts1`。`unauthorized` 部分中的 `agent-user` 条目之一是正则表达式 `accounts*`，这将覆盖授予用户名 `accounts1` 的权限。针对该文件空间，未授权与正则表达式 `accounts*` 匹配的所有用户（包括 `accounts1`）。

过程

1. 创建以下格式的 HTTP 请求：

```
POST HTTP/1.1 /admin/filespace/andrew
Host: example.com
User-Agent: mozilla
Content-Type: application/xml
Content-Length: 266

<?xml version="1.0" encoding="UTF-8"?>
<filespaces>
  <filespace>
    <quota bytes="1048576"/>
    <writers>
      <authorized>
        <agent-user>bill</agent-user>
        <agent-user>accounts1</agent-user>
        <agent-user>fte.*</agent-user>
      </authorized>
      <unauthorized>
        <agent-user>fteuser</agent-user>
        <agent-user>accounts*</agent-user>
      </unauthorized>
    </writers>
  </filespace>
</filespaces>
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 200 OK
Server: WAS/7.0
Content-Length: 0
```

现在存在名为 `andrew` 的文件空间，可向其传输文件。用户 `andrew`、`bill` 及用户名以 `fte` 开头的任何用户（用户 `fteuser` 除外）都可以向文件空间传输文件。与正则表达式 `accounts*` 匹配的任何用户都不可以向文件空间传输文件。

有关如何向文件空间传输文件的信息，请参阅第 312 页的『示例：将文件传输到文件空间』。

会将创建文件空间的请求记录到应用程序服务器事件日志中。有关更多信息，请参阅第 943 页的『文件空间管理日志记录格式』。

无效请求将返回 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。要确定错误原因，请参阅第 412 页的『对 Web 网关进行故障诊断』。

相关概念

第 327 页的『Web 网关管理控制台』

IBM MQ Managed File Transfer 随附的 Web 网关管理控制台为您提供图形界面，用于管理文件空间和用户映射。如果具有安全角色 `wmqfte-admin`，那么可以使用管理控制台完成管理任务。

第 329 页的『用于管理的示例 HTTP 流』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

[第 931 页的『用于管理 Web 网关的 HTTP 头』](#)

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

[第 939 页的『文件空间创建或更改请求格式』](#)

通过在 HTTP 请求中包含 XML 格式的内容，可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 `FileSpaceInfo.xsd`，此模式位于 WMQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

[第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』](#)

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 Connect:Direct 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 `java.util.regex` API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

示例：删除文件空间

可以通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，删除现有的文件空间。如果正在对文件空间进行文件传输，那么将不会删除文件空间。

关于此任务

要删除文件空间，必须具有与用户帐户关联的相应安全角色。与 IBM MQ Managed File Transfer 角色 `wmqfte-admin` 和 `wmqfte-filespace-delete` 关联的用户可以删除文件空间。有关保护 Web 网关的更多信息，请参阅[第 108 页的『Web 网关的用户角色』](#)。

如果具有安全角色 `wmqfte-admin`，那么还可以使用管理控制台删除文件空间。有关更多信息，请参阅[第 327 页的『Web 网关管理控制台』](#)。

成功删除文件空间

关于此任务

在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 `example.com`。文件空间属于用户 `richard`。当前未对文件空间 `richard` 执行任何文件传输。您可以通过列出文件空间，找出正在对 Web 网关环境中的文件空间进行的一些传输。有关更多信息，请参阅[第 333 页的『示例：列出所有文件空间』](#)。

过程

1. 要删除文件空间 `richard`，请使用以下格式创建 HTTP 请求：

```
DELETE HTTP/1.1 /admin/filespace/richard
Host: example.com
User-Agent: mozilla
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 200 OK
Server: WAS/7.0
```

将删除文件空间 `richard` 及其包含的任何文件。会将文件空间的删除记录到应用程序服务器事件日志中。有关更多信息，请参阅[第 943 页的『文件空间管理日志记录格式』](#)。

关于此任务

在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 `example.com`。文件空间属于用户 `suzanne`。当前正在对文件空间 `suzanne` 进行两个传输。您可以通过列出文件空间，找出正在对 Web 网关环境中的文件空间进行的一些传输。有关更多信息，请参阅第 333 页的『[示例：列出所有文件空间](#)』。

过程

1. 要删除文件空间 `suzanne`，请使用以下格式创建 HTTP 请求：

```
DELETE HTTP/1.1 /admin/filespace/suzanne
Host: example.com
User-Agent: mozilla
```

2. 向 Web 网关提交该请求。该请求因正在对文件空间进行传输而失败。因此，您将收到来自 Web 网关的以下响应：

```
HTTP/1.1 409 Conflict
Server: WAS/7.0

BFGWI0060E: The file space 'suzanne' is currently in use, and cannot be deleted.
```

您必须等待到文件空间的传输完成，才能删除文件空间。

要确定您可能收到的任何其他错误的原因，请参阅第 412 页的『[对 Web 网关进行故障诊断](#)』。

示例：修改文件空间配置

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以修改现有文件空间。如果您的用户帐户具有关联的必要安全角色，那么可以更改文件空间配额以及有权访问该文件空间的用户列表。

关于此任务

使用 IBM MQ Managed File Transfer 角色 `wmqfte-admin` 和 `wmqfte-filespace-modify`，可以更改文件空间配额以及有权访问该文件空间的用户列表。有关保护 Web 网关的更多信息，请参阅第 108 页的『[Web 网关的用户角色](#)』。

如果您具有安全角色 `wmqfte-admin`，那么还可以使用管理控制台来修改文件空间。有关更多信息，请参阅第 327 页的『[Web 网关管理控制台](#)』。

如果在将文件传输到文件空间时更改了文件空间配额，那么即使这些文件传输会导致超出新配额值，这些文件传输也可能成功。但对于在更改配额后才开始的任何文件传输，它们只有在文件传输未导致超出新配额值时才会成功。

以下示例显示了如何更改文件空间配额、将用户添加到有权访问文件空间的用户列表，以及从无权访问文件空间的用户列表中移除用户。在此示例中，用于托管 Web 网关的服务器是 `example.com`。已创建的文件空间的名称为 `finlay`。文件空间的名称由 POST 请求所使用的 URI 的最后一个部分表示。

有关用于修改文件空间的 XML 请求的格式的更多信息，请参阅第 939 页的『[文件空间创建或更改请求格式](#)』。

过程

1. 如果要在现有用户列表中添加或移除用户，请在 `authorized` 和 `unauthorized` 元素上使用 `add` 操作或 `remove` 操作。例如，以下请求将两个用户添加到授权用户列表中并从未授权用户列表中移除一个用户：

```
POST HTTP/1.1 /admin/filespace/finlay
Host: example.com
```



```
User-Agent: mozilla
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<filespaces>
  <filesystem>
    <quota bytes="100000000"/>
    <writers>
      <authorized action="add">
        <agent-user>jonathan</agent-user>
        <agent-user>lauren</agent-user>
      </authorized>
      <unauthorized action="remove">
        <agent-user>marley</agent-user>
      </unauthorized>
    </writers>
  </filesystem>
</filespaces>
```

如果要覆盖当前用户列表（而不是在现有列表中添加或删除用户），请在 **authorized** 和 **unauthorized** 元素上使用 **overwrite** 操作。例如，以下请求将覆盖当前的授权用户列表：

```
POST HTTP/1.1 /admin/filespace/finlay
Host: example.org
User-Agent: mozilla
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<filespaces>
  <filesystem>
    <writers>
      <authorized action="overwrite">
        <agent-user>fte.*</agent-user>
        <agent-user>ella</agent-user>
        <agent-user>jonathan</agent-user>
        <agent-user>lauren</agent-user>
      </authorized>
    </writers>
  </filesystem>
</filespaces>
```

您可以使用 Java 正则表达式来匹配多个用户名。例如，先前示例中的某个 **agent-user** 条目具有值 **fte.***，该值将与名称以 **fte** 开头的任何用户匹配。

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 200 OK
Server: WAS/7.0
Content-Length: 0
```

用于修改文件空间的请求会记录到应用程序服务器的事件日志中。有关更多信息，请参阅 [第 943 页的『文件空间管理日志记录格式』](#)。

无效请求将返回 HTTP 错误代码和 WMQFTE 错误消息。要确定错误原因，请参阅 [第 412 页的『对 Web 网关进行故障诊断』](#)。

相关概念

[第 327 页的『Web 网关管理控制台』](#)

IBM MQ Managed File Transfer 随附的 Web 网关管理控制台为您提供图形界面，用于管理文件空间 and 用户映射。如果具有安全角色 **wmqfte-admin**，那么可以使用管理控制台完成管理任务。

[第 329 页的『用于管理的示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

[第 931 页的『用于管理 Web 网关的 HTTP 头』](#)

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

第 939 页的『文件空间创建或更改请求格式』

通过在 HTTP 请求中包含 XML 格式的内容，可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 `FileSpaceInfo.xsd`，此模式位于 WMQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

示例：将文件传输到文件空间

将单个文件传输到 IBM MQ Managed File Transfer 文件空间。通过在 `fteCreateTransfer` 命令中使用 `-du` 参数，可将文件空间指定为文件传输的目标。

关于此任务

将文件传输到文件空间时，IBM MQ Managed File Transfer Web 网关将检查传输是否会导致超出文件空间配额。如果会超出配额，那么将生成错误，并且文件传输将失败。Web 网关管理员可以通过提交 HTTP 请求来增加文件空间配额的大小。有关示例请求，请参阅主题第 331 页的『示例：修改文件空间配置』。

文件空间配额检查在传输开始之前进行。如果使用多个代理将文件传输到同一个文件空间，或者当文件正在传输到文件空间时 Web 网关管理员降低了该文件空间的配额，那么尽管一个或多个传输导致超出文件空间配额，但它们仍可能会成功。

在该示例中，源文件名为 `/tmp/Accounts.csv`，该文件与源代理 `AGENT_1` 位于同一系统上。目标文件空间 `john` 属于用户 `john`，与代理 `FS_AGENT` 位于同一系统上。请求传输的用户对文件空间 `john` 具有写访问权。代理 `FS_AGENT` 使用队列管理器 `FS_QM`。

过程

输入以下命令：

```
fteCreateTransfer -sa AGENT_1 -da FS_AGENT -dm FS_QM -du john /tmp/Accounts.csv
```

文件 `/tmp/Accounts.csv` 传输到文件空间 `john`。用户 `john` 可在需要时从文件空间下载该文件。

相关概念

第 339 页的『文件空间』

文件空间是与 Web 网关用户关联的文件存储器的保留区域。文件空间具有配额存储量。文件空间的访问权受限于有权对其执行读写的文件空间。

相关任务

第 221 页的『启动新的文件传输』

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

示例：列出文件空间中的所有文件

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可列出文件空间的内容。Web 网关会以 XML 或 JSON 格式返回响应，其中列出文件空间的内容。如果您是文件空间的所有者或具有安全角色 `wmqfte-admin`，那么您有权列出文件空间的内容。

关于此任务

成功的请求会返回 HTTP 状态码 200 以及列出了文件空间中前 100 个文件的有效内容。该响应会以 XML（缺省值）或 JSON 格式返回，具体取决于在请求中指定的“Accept”头。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 `example.com`，并且通过 Web 浏览器（将自身标识为 `mozilla`）提交了 HTTP 请求。要列出的文件空间的名称是“`john`”，其中包含两个文件。头“`Accept: application/xml`”指定 Web 网关应以 XML 格式返回结果。有关文件空间列表请求返回的格式的更多信息，请参阅第 925 页的『文件空间查询的响应格式』。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /fileSpace/john
Host: example.com
User-Agent: mozilla
Accept: application/xml
```

2. 向 Web 网关提交该请求。

结果

Web 网关将返回以下格式的 HTTP 响应：

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<fileSpaces xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="WebFileSpaceList.xsd">
  <fileSpace size="2" name="john">
    <file fileLink="/wmqfte/fileSpace/john/
      414d51205745422e46544520202020c1a1a34b03720120/fileName"
      fsLocation="/var/ibm/WMQFTE/web/fte/transfer/
        414d51205745422e46544520202020c1a1a34b03720120/file-0"
      transferLink="/wmqfte/transfer/414d51205745422e46544520202020c1a1a34b03720120"
      transferID="414d51205745422e46544520202020c1a1a34b03720120">
      <attribute-values mode="text" created="2010-08-26T11:45:02.000Z" size="259354303"
        checksum-value="98611a272a27d373f92d73a08cf0d4f4" checksum-
method="MD5"/>
    </file>
    <file fileLink="/wmqfte/fileSpace/john/
      414d51205745422e46544520202020c1a1a34b06520120/fileName"
      fsLocation="/var/ibm/WMQFTE/web/fte/transfer/
        414d51205745422e46544520202020c1a1a34b06520120/file-0"
      transferLink="/wmqfte/transfer/414d51205745422e46544520202020c1a1a34b06520120"
      transferID="414d51205745422e46544520202020c1a1a34b06520120">
      <attribute-values mode="text" created="2010-08-26T12:15:02.260Z" size="259554303"
        checksum-value="98611a272a27d37bf22d73a08cf0d4f4" checksum-
method="MD5"/>
    </file>
  </fileSpace>
</fileSpaces>
```

相关参考

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

第 925 页的『文件空间查询的响应格式』

从 IBM MQ Managed File Transfer 服务 Web 网关请求文件空间中的部分或全部文件的列表时，将根据您使用 Accept: 头指定的内容以 JSON 或 XML 格式返回响应。

示例：检查文件空间中文件的完整性

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可检查文件空间中文件的完整性。例如，如果要在数据丢失之后复原文件系统，那么可以检查文件空间中的文件是否位于文件系统上的正确位置。Web 网关会以 XML 或 JSON 格式返回响应，其中列出文件空间的内容以及用于指示每个文件的完整性检查结果的附加属性。

关于此任务

成功的请求会返回 HTTP 状态码 200 以及列出了文件空间中前 100 个文件的有效内容。您可以请求以 XML 或 JSON 格式返回文件的详细信息。还可以编写 Web 应用程序来解析响应的内容，并以适当的格式呈现给 Web 用户。仅管理员有权使用完整性检查属性列出文件空间中的文件。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 example.com。要列示的文件空间的名称为 john，它包含两个文件。头 Accept:

application/xml 指定查询以 XML 格式返回结果。头 x-fte-check-integrity 指定查询返回的结果中包含每个文件的额外完整性检查属性。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /fileSpace/john
Host: example.com
User-Agent: mozilla
Accept: application/xml
x-fte-check-integrity: true
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
<fileSpaces xsi:noNamespaceSchemaLocation="WebTransferStatus.xsd">
  <fileSpace size="2" name="john">
    <file fileLink="/wmqfte/fileSpace/john/
      414d51205745422e46544520202020c1a1a34b03720120/ar5erh"
      transferLink="/wmqfte/transfer/
        414d51205745422e46544520202020c1a1a34b03720120"
      transferID="414d51205745422e46544520202020c1a1a34b03720120"
      name="/tmp/file1.zip"
      fsLocation="/fileSpaces/john/
        414d51205745422e46544520202020c1a1a34b03720120/file-0">
      <attribute-values mode="text" time="2010-08-26T11:45:02.000Z"
        file-size="259354303"
        checksum-value="98611a272a27d373f92d73a08cf0d4f4"
        checksum-method="none"
        integrity-check-result="OK"/>
    </file>
    <file fileLink="/wmqfte/fileSpace/john/
      414d51205745422e46544520202020c1a1a34b06520120/ar5erh"
      transferLink="/wmqfte/transfer/
        414d51205745422e46544520202020c1a1a34b06520120"
      transferID="414d51205745422e46544520202020c1a1a34b06520120"
      name="/tmp/file2.zip"
      fsLocation="/fileSpaces/john/
        414d51205745422e46544520202020c1a1a34b06520120/file-0">
      <attribute-values mode="text" time="2010-08-26T12:15:02.260Z"
        file-size="259554303"
        checksum-value="98611a272a27d37bf22d73a08cf0d4f4"
        checksum-method="none"
        integrity-check-result="MISSING-FILESYSTEM"/>
    </file>
  </fileSpace>
</fileSpaces>
```

结果

该示例结果指示第一个文件已通过完整性检查。integrity-check-result 属性值 OK 显示 Web 网关数据库中在该文件并且在文件系统中找到了匹配的文件。第二个文件的完整性检查失败。MISSING-FILESYSTEM 的 integrity-check-result 属性值显示该文件存在于 Web 网关数据库中，但在 fsLocation 属性给定的位置中的文件系统中找不到该文件。在这种情况下，可能需要管理员从文件空间删除该文件，或从备份恢复文件空间目录。

有关 integrity-check-result 属性的可能值，请参阅第 937 页的『文件空间信息响应格式』。

相关概念

第 329 页的『用于管理的示例 HTTP 流』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

第 931 页的『用于管理 Web 网关的 HTTP 头』

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

示例：列出文件空间中的特定文件子集

通过向 IBM MQ Managed File Transfer Web 网关提交包含查询的 HTTP 请求，可查询文件空间的内容。Web 网关会以 XML 或 JSON 格式返回响应，其中仅描述文件空间中与查询匹配的文件。

关于此任务

您可以将查询附加到 HTTP 请求，以请求有关文件空间中与查询匹配的文件的文件的信息。您可以按照文件关联的详细信息来查询文件，这些详细信息包括始发用户、传输开始时间、传输结束时间以及将文件发送到文件空间的传输的标识。您还可以指定要返回的结果数。

成功的请求会返回 HTTP 状态码 200 以及描述与查询匹配的文件的文件的有效内容。您可以请求以 XML 或 JSON 格式返回文件的详细信息。还可以编写 Web 应用程序来解析响应的内容，并以适当的格式呈现给 Web 用户。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 example.com。请求该信息的用户有权访问要查询的文件空间。查询请求以 JSON 格式返回的信息，由 accept=json 查询指定。该查询请求满足以下条件的文件的列表：

- 文件位于文件空间 james 中。
- 用户 bob(由 originatoruser=bob 查询指定) 已将这些文件发送到文件空间。
- 这些文件在 2010 年 8 月 26 日 13:00 (UTC) 之后发送到文件空间，由 startafter=2010-08-26T13:00 查询指定。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /fileSpace/james/?originatoruser=bob&startafter=2010-08-26T13:00&accept=json
Host: example.com
User-Agent: mozilla
```

2. 向 Web 网关提交该请求。Web 网关使用以下格式返回 HTTP 响应。在该示例中，只有一个文件与查询匹配。

```
{
  "fileSpaces" : {
    "fileSpace" : {
      "name" : "james",
      "size" : "1",
      "file" : {
        "transferLink" : "\\wmqfte\\transfer\\
          414d51205745422e46544520202020c1a1a34b03720120",
        "fileLink" : "\\wmqfte\\fileSpace\\james\\
          414d51205745422e4654452020202020c1a1a34b03720120\\wibble",
        "name" : "\\tmp\\bobs_file.zip",
        "transferID" : "414d51205745422e46544520202020c1a1a34b03720120",
        "attribute-values" : {
          "checksum-value" : "98611a272a27d373f92d73a08cf0d4f4",
          "checksum-method" : "none",
          "time" : "2010-08-26T14:13:02.000Z",
          "file-size" : "259354303",
          "mode" : "text"
        }
      }
    }
  }
}
```

相关参考

第 925 页的『文件空间查询的响应格式』

从 IBM MQ Managed File Transfer 服务 Web 网关请求文件空间中的部分或全部文件的列表时，将根据您使用 Accept: 头指定的内容以 JSON 或 XML 格式返回响应。

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

示例：从文件空间检索文件

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可从文件空间检索文件。Web 网关提供使用 HTTP 协议下载文件的功能。

关于此任务

要从文件空间下载文件，您必须是文件空间的所有者或具有安全角色 wmqfte-admin。成功的请求会返回 HTTP 状态码 200 以及下载的文件。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 example.com。下载的文件是 Accounts.csv，将文件发送到文件空间的传输的标识是 4142452b345f4d2e3c2a333d4ed3e4de43453bc2344a2020。包含该文件的文件空间的名称是 john，且请求该信息的用户有权访问该文件空间。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /filespace/john/4142452b345f4d2e3c2a333d4ed3e4de43453bc2344a2020/Accts.csv
Host: example.com
User-Agent: mozilla
```

2. 向 Web 网关提交该请求。Web 网关会在 HTTP 响应中返回文件。

在 HTTP 响应中设置了以下头：

- Content-Type: application/x-download
- Content-MD5: 98611a272a27d373f92d73a08cf0d4f4
- Content-Disposition: attachment; filename="Accts.csv"
- Content-Length: 8786

相关参考

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

示例：从文件空间中删除文件

可以通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，从文件空间中删除文件。如果将头 x-fte-include-file-in-response 设置为 true，那么将通过 Web 网关在 HTTP 响应中返回文件内容。

关于此任务

成功的删除请求将返回 HTTP 状态码 200 和（如果已在请求中指定）被删除文件的内容。如果提交请求的用户不是文件空间的所有者，那么请求将失败。

注: 安全角色 `wmqfte-admin` 可以从文件空间中删除文件，但不能接收被删除文件的内容。如果安全角色为 `wmqfte-admin` 的用户尝试删除文件并请求文件内容，那么请求将因资源错误而失败。有关更多信息，请参阅第 108 页的『Web 网关的用户角色』。

以下步骤描述了如何提交请求。在此示例中，用于托管 IBM MQ Managed File Transfer Web 网关的服务器是 `example.com`。文件空间的名称为 `jack`，它包含文件 `report.txt`，并且请求文件删除的用户是文件空间的所有者。传输标识 `414d5120514d5f67617265746862202067732c4c20c25a03` 是将文件放入文件空间的传输的十六进制标识，在您列出文件空间的内容时将返回该标识。有关文件空间查询响应格式的更多信息，请参阅第 925 页的『文件空间查询的响应格式』。

头 `x-fte-include-file-in-response:true` 指定在响应主体中返回 `report.txt` 的内容。如果未指定该头的内容，那么缺省值为 `false`，将删除该文件，但不会返回其内容。

过程

1. 创建以下格式的 HTTP 请求：

```
DELETE HTTP/1.1 /filepace/jack/414d5120514d5f67617265746862202067732c4c20c25a03/report.txt
Host: example.com
User-Agent: mozilla
x-fte-include-file-in-response:true
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 200 OK
Server: WAS/6.0
Content-Length: 1762
Content-MD5: 9608f0d8cdcb804d185ab3cb959dba6f
Content-type: text/plain; charset=Cp1252
Content-Disposition: attachment; filename="report.txt"

Account No, Balance
123456, 100.00
234567, 1022.00
345678, 2801.00
456789, 16.75
```

相关参考

第 108 页的『Web 网关的用户角色』

IBM MQ Managed File Transfer 已定义多个不同的角色，用于控制用户可执行的操作。

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 `/wmqfte`。

第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

示例：列出所有文件空间

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以列出所有文件空间。Web 网关将返回 XML 或 JSON 格式的响应，其中列出文件空间的名称、每个文件空间的配额以及有权和无权向每个文件空间写入内容的用户。

关于此任务

成功的请求将返回 HTTP 状态码 200 以及用于描述最多 100 个文件空间的有效内容。

在此示例中，托管 Web 网关的服务器为 `example.com`。当前有三个属于用户 `richard`，`suzanne` 和 `hamilton` 的文件空间。当前未对文件空间 `richard` 执行任何文件传输。正在对文件空间 `hamilton` 执行一个文件传输，并对文件空间 `suzanne` 执行两个文件传输。请求信息的用户与安全角色 `wmqfte-admin` 相关联。头 `Accept: application/xml` 指定该查询将返回 XML 格式的结果。

过程

1. 创建以下格式的 HTTP 请求：


```
GET HTTP/1.1 /admin/filespace/  
Host: example.com  
User-Agent: mozilla  
Accept: application/xml
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应:

```
HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
Content-Type: application/xml  
  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<filesystems xsi:noNamespaceSchemaLocation="FileSpaceInfo.xsd"  
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">  
  <filesystem transfers="0" location="/mnt/gateway/richard" name="richard">  
    <quota bytes="1048576"/>  
    <writers>  
      <authorized>  
        <agent-user>charlene</agent-user>  
        <agent-user>alan</agent-user>  
      </authorized>  
    </writers>  
  </filesystem>  
  <filesystem transfers="2" location="/mnt/gateway/suzanne" name="suzanne">  
    <quota bytes="20489878"/>  
    <writers>  
      <authorized>  
        <agent-user>charlene</agent-user>  
        <agent-user>sammy</agent-user>  
      </authorized>  
      <unauthorized>  
        <agent-user>arnold</agent-user>  
        <agent-user>frank</agent-user>  
      </unauthorized>  
    </writers>  
  </filesystem>  
  <filesystem transfers="1" location="/mnt/gateway/hamilton" name="hamilton">  
    <quota bytes="666999"/>  
    <writers>  
      <authorized>  
        <agent-user>joseph</agent-user>  
      </authorized>  
      <unauthorized>  
        <agent-user>junior</agent-user>  
      </unauthorized>  
    </writers>  
  </filesystem>  
</filesystems>
```

相关概念

第 329 页的『用于管理的示例 HTTP 流』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

第 930 页的『Web 网关管理 API 引用』

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

第 931 页的『用于管理 Web 网关的 HTTP 头』

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

第 937 页的『文件空间信息响应格式』

通过 IBM MQ Managed File Transfer 服务 Web 网关请求有关文件空间的定义和属性的信息时，将以 XML 格式或 JSON 格式返回响应。该 XML 响应符合模式 FileSpaceInfo.xsd，该模式位于 MQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

示例：检查所有文件空间的完整性

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以检查所有文件空间的完整性。例如，如果要在数据丢失之后复原文件系统，那么可以检查文件空间中的文件是否位于文件系统上的正确位

置。Web 网关将返回 XML 或 JSON 格式的响应，其中列出文件空间的名称以及一个用于指示文件空间条目是否与文件系统中的文件相匹配的属性。

关于此任务

使用 Web 网关管理 API 来请求当前存在的所有文件空间的列表。成功的请求将返回 HTTP 状态码 200 以及用于描述最多 100 个文件空间的有效内容。在此示例中，托管 IBM MQ Managed File Transfer Web 网关的服务器为 `example.com`。当前有三个文件空间，分别属于用户 `richard`、`suzanne` 和 `hamilton`。请求该信息的用户与安全角色 `wmqfte-admin` 相关联。头 `Accept: application/xml` 指定该查询将返回 XML 格式的结果。头 `x-fte-check-integrity` 指定应检查每个文件空间以确保文件系统中存在匹配的目录。

过程

1. 创建以下格式的 HTTP 请求：

```
GET HTTP/1.1 /admin/filespace/  
Host: example.com  
User-Agent: mozilla  
Accept: application/xml  
x-fte-check-integrity: true
```

2. 向 Web 网关提交该请求。Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
Content-Type: application/xml  
  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<filespaces xsi:noNamespaceSchemaLocation="FileSpaceInfo.xsd"  
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">  
  <filesystem transfers="0" location="/mnt/gateway/richard" name="richard"  
    integrity-check-result="OK">  
    <quota bytes="1048576"/>  
    <writers>  
      <authorized>  
        <agent-user>charlene</agent-user>  
        <agent-user>alan</agent-user>  
      </authorized>  
    </writers>  
  </filesystem>  
  <filesystem transfers="2" location="/mnt/gateway/suzanne" name="suzanne"  
    integrity-check-result="MISSING-FILESYSTEM">  
    <quota bytes="20489878"/>  
    <writers>  
      <authorized>  
        <agent-user>charlene</agent-user>  
        <agent-user>sammy</agent-user>  
      </authorized>  
      <unauthorized>  
        <agent-user>arnold</agent-user>  
        <agent-user>frank</agent-user>  
      </unauthorized>  
    </writers>  
  </filesystem>  
  <filesystem transfers="1" location="/mnt/gateway/hamilton" name="hamilton"  
    integrity-check-result="OK">  
    <quota bytes="666999"/>  
    <writers>  
      <authorized>  
        <agent-user>joseph</agent-user>  
      </authorized>  
      <unauthorized>  
        <agent-user>junior</agent-user>  
      </unauthorized>  
    </writers>  
  </filesystem>  
</filespaces>
```

结果

此示例结果指示结果集中的第一个和第三个文件空间已通过完整性检查。OK 的 `integrity-check-result` 属性值显示文件空间存在于 Web 网关数据库中，并且已在文件系统上找到匹配的目录。第二个文件空间未通过完整性检查。MISSING-FILESYSTEM 的 `integrity-check-result` 属性值显示文件空间存在于 Web 网关数据库中，但在文件系统上找不到 `location` 属性所指示的目录。在这种情况下，管理员可能需要删除该文件空间，或从备份复原该文件空间根目录。

如果您具有安全角色 `wmqfte-admin`，那么还可以使用管理控制台来检查所有文件空间的完整性。有关更多信息，请参阅第 327 页的『Web 网关管理控制台』。

要了解 `integrity-check-result` 属性的可能值，请参阅第 937 页的『文件空间信息响应格式』。

相关概念

第 327 页的『Web 网关管理控制台』

IBM MQ Managed File Transfer 随附的 Web 网关管理控制台为您提供图形界面，用于管理文件空间 and 用户映射。如果具有安全角色 `wmqfte-admin`，那么可以使用管理控制台完成管理任务。

相关任务

第 334 页的『示例：检查文件空间中文件的完整性』

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可检查文件空间中文件的完整性。例如，如果要在数据丢失之后复原文件系统，那么可以检查文件空间中的文件是否位于文件系统上的正确位置。Web 网关会以 XML 或 JSON 格式返回响应，其中列出文件空间的内容以及用于指示每个文件的完整性检查结果的附加属性。

样本 Web 页面

IBM MQ Managed File Transfer Web 网关提供了样本 Web 页面。该样本使用 Web 网关 API 函数来上载文件、查看文件传输的状态、查看文件空间的内容以及从文件空间下载文件。

样本应用程序文件名是 `com.ibm.wmqfte.web.samples.war`。您可以在 IBM MQ Managed File Transfer Server 安装的 `MQ_INSTALLATION_PATH/mqft/samples/web/servlet` 目录中找到此 WAR 文件。

设置样本之前，Web 网关应用程序必须已部署在应用程序服务器中并且正在运行。有关指示信息，请参阅第 181 页的『配置 Web 网关』。

安装样本

1. 将样本应用程序部署到应用程序服务器中。

如果将样本部署到 WebSphere Application Server V7.0 中：

- 为样本应用程序定义上下文根。例如，如果使用上下文根 `/wmqftesamples`，那么可通过 URI `/wmqftesamples` 访问样本 Web 页面。
- 必须为样本应用程序配置安全角色和用户。样本应用程序使用为 Web 网关定义的相同安全域。有关更多信息，请参阅第 198 页的『在 WebSphere 应用程序服务器 V7.0 中部署 Web 网关』。

如果将样本部署到 WebSphere Application Server Community Edition 中：

- 应用程序使用位于 Web 网关 EAR 文件的 `geronimo-web.xml` 部署计划中定义的上下文根。该上下文根是 `/wmqftesamples`。
- 必须为样本应用程序配置安全角色和用户。样本应用程序使用为 Web 网关定义的相同安全域。有关更多信息，请参阅第 187 页的『定义安全域』。

2. 打开 Web 浏览器，并基于部署样本时定义的上下文根输入样本的 URI。样本的 URI 是 `host:port/context_root`。

注：`port` 的值取决于您使用的应用程序服务器。例如，对于 WebSphere Application Server V7.0，应用程序使用的缺省端口是 9080。

3. 使用定义安全域时配置的用户名和密码登录到样本应用程序。
4. 如果为 Web 网关定义的上下文根并非缺省值 `wmqfte`，请使用样本应用程序的设置部分来指定 Web 网关上下文根。

5. 使用样本应用程序将文件上载到 Web 网关、查看文件空间中的文件、从文件空间下载和删除文件以及查看文件传输的状态。

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 327 页的『Web 网关管理控制台』](#)

IBM MQ Managed File Transfer 随附的 Web 网关管理控制台为您提供图形界面，用于管理文件空间和用户映射。如果具有安全角色 `wmqfte-admin`，那么可以使用管理控制台完成管理任务。

相关任务

[第 197 页的『部署 IBM MQ Managed File Transfer Web 网关』](#)

必须将 IBM MQ Managed File Transfer Web 网关 部署到与 Java Platform, Enterprise Edition 5 兼容的应用程序服务器。不同应用程序服务器的部署过程各不相同。本部分概述了两个应用程序服务器的部署过程。

将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

您可以使用 **fteAnt** 命令以在已配置的 IBM MQ Managed File Transfer 环境中运行 Ant 任务。您可以使用 Ant 脚本中的文件传输 Ant 任务，以通过解释后的脚本语言来协调复杂的文件传输操作。

fteAnt 命令不适用于 IBM 4690 环境。有关在 IBM 4690 环境中使用 IBM MQ Managed File Transfer 的更多信息，请参阅[第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』](#)

有关 Apache Ant 的更多信息，请参阅 Apache Ant 项目 Web 页面: <https://ant.apache.org/>

相关概念

[第 354 页的『入门：将 Ant 脚本用于 IBM MQ Managed File Transfer』](#)

将 Ant 脚本用于 IBM MQ Managed File Transfer 能够使您通过已解释的脚本语言协调复杂的文件传输操作。

相关参考

[第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』](#)

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

[第 450 页的『fteAnt \(在 IBM MQ Managed File Transfer 环境中运行 Ant 任务\)』](#)

fteAnt 命令会在具有可用的 IBM MQ Managed File Transfer Ant 任务的环境中运行 Ant 脚本。

[第 355 页的『Ant 任务样本』](#)

随 IBM MQ Managed File Transfer 安装一起提供了一些样本 Ant 脚本。这些样本位于目录 `MQ_INSTALLATION_PATH/mqft/samples/fteant` 中。每个样本脚本都包含 `init` 目标，请编辑 `init` 目标中设置的属性，以使用您的配置运行这些脚本。

入门：将 Ant 脚本用于 IBM MQ Managed File Transfer

将 Ant 脚本用于 IBM MQ Managed File Transfer 能够使您通过已解释的脚本语言协调复杂的文件传输操作。

Ant 脚本

Ant 脚本（或构建文件）是定义一个或多个目标的 XML 文档。这些目标中包含要运行的元素。IBM MQ Managed File Transfer 提供可用于将文件传输功能集成到 Apache Ant 中的任务。要了解有关 Ant 脚本的信息，请参阅 Apache Ant 项目 Web 页面: <https://ant.apache.org/>

产品安装的 `MQ_INSTALLATION_PATH/mqft/samples/fteant` 目录中提供了使用 IBM MQ Managed File Transfer 任务的 Ant 脚本示例

在协议网桥代理上，Ant 脚本运行在协议网桥代理系统上。这些 Ant 脚本不直接访问 FTP 或 SFTP 服务器上的文件。

名称空间

名称空间用于将文件传输 Ant 任务与其他可能享有同一名称的 Ant 任务区分开来。您可以在自己的 Ant 脚本的 project 标记中定义名称空间。

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns:fte="antlib:com.ibm.wmqfte.ant.taskdefs" default="do_ping">

  <target name="do_ping">
    <fte:ping cmdqm="qm@localhost@1414@SYSTEM.DEF.SVRCONN" agent="agent1@qm1"
      rcproperty="ping.rc" timeout="15"/>
  </target>

</project>
```

属性 xmlns:fte="antlib:com.ibm.wmqfte.ant.taskdefs" 指示 Ant 在 com.ibm.wmqfte.ant.taskdefs 库中查找前缀为 fte 的任务定义。

您不必将 fte 用作名称空间前缀；您可以使用任意值。名称空间前缀 fte 用于所有示例和 Ant 脚本样本。

运行 Ant 脚本

要运行包含文件传输 Ant 任务的 Ant 脚本，请使用 **fteAnt** 命令。例如：

```
fteAnt -file ant_script_location/ant_script_name
```

有关更多信息，请参阅 [第 450 页的『fteAnt \(在 IBM MQ Managed File Transfer 环境中运行 Ant 任务\)』](#)。

返回码

文件传输 Ant 任务返回与 IBM MQ Managed File Transfer 命令相同的返回码。有关更多信息，请参阅 [第 405 页的『IBM MQ Managed File Transfer 的返回码』](#)。

相关参考

[第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』](#)

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

[第 450 页的『fteAnt \(在 IBM MQ Managed File Transfer 环境中运行 Ant 任务\)』](#)

fteAnt 命令会在具有可用的 IBM MQ Managed File Transfer Ant 任务的环境中运行 Ant 脚本。

[第 355 页的『Ant 任务样本』](#)

随 IBM MQ Managed File Transfer 安装一起提供了一些样本 Ant 脚本。这些样本位于目录 `MQ_INSTALLATION_PATH/mqft/samples/fteant` 中。每个样本脚本都包含 `init` 目标，请编辑 `init` 目标中设置的属性，以使用您的配置运行这些脚本。

Ant 任务样本

随 IBM MQ Managed File Transfer 安装一起提供了一些样本 Ant 脚本。这些样本位于目录 `MQ_INSTALLATION_PATH/mqft/samples/fteant` 中。每个样本脚本都包含 `init` 目标，请编辑 `init` 目标中设置的属性，以使用您的配置运行这些脚本。

email

email 样本演示如何使用 Ant 任务传输文件以及在传输失败时如何将电子邮件发送到指定的电子邮件地址。脚本将检查源和目标代理是否处于活动状态以及是否能够使用 IBM MQ Managed File Transfer [ping](#) 任务来处理传输。如果两个代理都处于活动状态，那么脚本将使用 IBM MQ Managed File Transfer [filecopy](#) 任务在源和目标代理之间传输文件，而不删除原始文件。如果传输失败，那么脚本将使用标准的 Ant email 任务发送一封包含失败相关信息的电子邮件。

hub

中心 样本由两个脚本组成: `hubcopy.xml` 和 `hubprocess.xml`. `hubcopy.xml` 脚本显示如何使用 Ant 脚本编制来构建 "中心和辐射" 样式拓扑。在该样本中, 将两个文件从辐条机器上运行的代理传输到集线器机器上运行的代理。同时传输这两个文件, 当传输完成时, `hubprocess.xml` Ant 脚本在集线器机器上运行以处理这些文件。如果两个文件传输正确, 那么 Ant 脚本将合并这些文件的内容。如果文件未正确传输, 那么 Ant 脚本将通过删除传输的所有文件数据来进行清理。针对本示例, 要正确运行, 您必须将 `hubprocess.xml` 脚本放置在集线器代理的命令路径上。要获取有关设置代理的命令路径的更多信息, 请参阅 [commandPath](#)。

librarytransfer (仅限 IBM i 平台)

`librarytransfer` 样本演示如何使用 Ant 任务将一个 IBM i 系统上的 IBM i 库传输到第二个 IBM i 系统。

IBM i 上的 IBM MQ Managed File Transfer V7.0.2 不包含直接支持传输本机 IBM i 库对象。 `librarytransfer` 样本通过 IBM MQ Managed File Transfer 中可用的预定义 Ant 任务在 IBM i 上使用本机保存文件支持, 以在两个 IBM i 系统之间传输本机库对象。 The sample uses a `<presrc>` nested element in a IBM MQ Managed File Transfer filecopy task to invoke an executable script `librarysave.sh` that saves the requested library on the source agent system into a temporary save file. The save file is moved by the filecopy ant task to the destination agent system where a `<postdst>` nested element is used to invoke the executable script `libraryrestore.sh` to restore the library saved in the save file to the destination system.

在您运行该样本之前, 需要完成一些配置, 如 `librarytransfer.xml` 文件中所述。还必须在两个 IBM i 机器上具有运行良好的 IBM MQ Managed File Transfer 环境。该设置必须包含在第一个 IBM i 机器上运行的源代理和在第二个 IBM i 机器上运行的目标代理。两个代理必须能够彼此通信。

`librarytransfer` 样本包含以下三个文件:

- `librarytransfer.xml`
- `librarysave.sh` (`<presrc>` 可执行脚本)
- `libraryrestore.sh` (`<postdst>` 可执行脚本)

这些样本文件位于以下目录中: `/QIBM/ProdData/WMQFTE/V7/samples/fteant/ibmi/librarytransfer`

要运行该样本, 用户必须完成以下步骤:

1. 启动 Qshell 会话。在 IBM i 命令窗口输入: `STRQSH`
2. 将目录更改为 `bin` 目录, 如下所示:

```
cd /QIBM/ProdData/WMQFTE/V7/bin
```

3. 在完成所需配置后, 通过使用以下命令运行该样本:

```
fteant -f /QIBM/ProdData/WMQFTE/V7/samples/fteant/ibmi/librarytransfer/librarytransfer.xml
```

physicalfiletransfer (仅限 IBM i 平台)

`physicalfiletransfer` 样本演示如何使用 Ant 任务将源物理或数据库文件从一个 IBM i 系统上的库传输到另一个 IBM i 系统上的库。

IBM i 上的 IBM MQ Managed File Transfer V7.0.2 不包含直接支持传输 IBM i 上的本机源物理或数据库文件。 `physicalfiletransfer` 样本通过 IBM MQ Managed File Transfer 中可用的预定义 Ant 任务在 IBM i 上使用本机保存文件支持, 以在两个 IBM i 系统之间传输完整的源物理和数据库文件。 The sample uses a `<presrc>` nested element within a IBM MQ Managed File Transfer filecopy task to invoke an executable script `physicalfilesave.sh` to save the requested Source Physical or Database file from a library on the source agent system into a temporary save file. The save file is moved by the filecopy ant task to the destination agent system where a `<postdst>` nested element is used to invoke the executable script `physicalfilerestore.sh` then restores the file object inside the save file into a specified library on the destination system.

在您运行该样本之前，必须完成一些配置，如 `physicalfiletransfer.xml` 文件中所述。还必须在两个 IBM i 系统上具有运行良好的 IBM MQ Managed File Transfer 环境。该设置必须包含在第一个 IBM i 系统上运行的源代理和在第二个 IBM i 系统上运行的目标代理。两个代理必须能够彼此通信。

`physicalfiletransfer` 样本包含以下三个文件：

- `physicalfiletransfer.xml`
- `physicalfilesave.sh` (<presrc> 可执行脚本)
- `physicalfilerestore.sh` (<postdst> 可执行脚本)

这些样本文件位于以下目录中： `/QIBM/ProdData/WMQFTE/V7/samples/fteant/ibmi/physicalfiletransfer`

要运行该样本，用户必须完成以下步骤：

1. 启动 Qshell 会话。在 IBM i 命令窗口输入：STRQSH
2. 将目录更改为 `bin` 目录，如下所示：

```
cd /QIBM/ProdData/WMQFTE/V7/bin
```

3. 在完成所需配置后，通过使用以下命令运行该样本：

```
fteant -f /QIBM/ProdData/WMQFTE/V7/samples/fteant/ibmi/physicalfiletransfer/  
physicalfiletransfer.xml
```

timeout

`timeout` 样本演示如何使用 Ant 任务尝试文件传输，并在花费时间长于指定的超时值时取消传输。该脚本通过使用 IBM MQ Managed File Transfer `filecopy` 任务启动文件传输。将延迟该传输的结果。该脚本使用 IBM MQ Managed File Transfer 第 952 页的『[fte:awaitoutcome](#)』任务等待指定的秒数使传输完成。如果传输在指定的时间内没有完成，那么 IBM MQ Managed File Transfer 第 954 页的『[fte:cancel](#)』任务将用于取消文件传输。

vsamtransfer

`vsamtransfer` 样本演示如何通过 IBM MQ Managed File Transfer 使用 Ant 任务从 VSAM 数据集传输到另一个 VSAM 数据集。IBM MQ Managed File Transfer 当前不支持传输 VSAM 数据集。样本脚本通过使用 `presrc` 嵌套元素来调用可执行文件 `datasetcopy.sh`，将 VSAM 数据记录卸载到连续数据集。该脚本使用 IBM MQ Managed File Transfer 第 958 页的『[fte:filemove](#)』任务将连续数据集从源代理传输到目标代理。然后，该脚本使用 `postdst` 嵌套元素来调用 `loadvsam.jcl` 脚本。该 JCL 脚本会将传输的数据集记录装入目标 VSAM 数据集。该样本对目标调用使用 JCL 以演示该语言选项。通过使用第二个 `shell` 脚本也可以获得相同的结果。

该样本不需要源和目标数据集是 VSAM。如果源和目标数据集的类型相同，那么该样本对任何数据集都有效。

针对该样本，要正确运行，您必须将 `datasetcopy.sh` 脚本放置在源代理的命令路径上，将 `loadvsam.jcl` 脚本放置在目标代理的命令路径上。要获取有关设置代理的命令路径的更多信息，请参阅 [commandPath](#)。

zip

`zip` 样本由两个脚本组成：`zip.xml` 和 `zipfiles.xml`。该样本演示如何在执行文件传输移动操作之前使用 IBM MQ Managed File Transfer 第 958 页的『[fte:filemove](#)』任务中的 `presrc` 嵌套元素来运行 Ant 脚本。`zip.xml` 脚本中的 `presrc` 嵌套元素调用的 `zipfiles.xml` 脚本将压缩目录内容。`zip.xml` 脚本传输压缩的文件。该样本需要 `zipfiles.xml` Ant 脚本位于源代理的命令路径上。这是因为 `zipfiles.xml` Ant 脚本包含用于在源代理上压缩目录内容的目标。要获取有关设置代理的命令路径的更多信息，请参阅 [commandPath](#)。

相关概念

第 354 页的『入门：将 Ant 脚本用于 IBM MQ Managed File Transfer』
将 Ant 脚本用于 IBM MQ Managed File Transfer 能够使您通过已解释的脚本语言协调复杂的文件传输操作。

相关参考

第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』
IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

使用用户出口例程定制 IBM MQ Managed File Transfer

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

IBM MQ Managed File Transfer 在代码中提供了一些“点”，IBM MQ Managed File Transfer 可以通过这些点向您已编写的程序（用户出口例程）传递控件。这些“点”被称为用户出口点。程序完成其工作后，IBM MQ Managed File Transfer 便可恢复控制。您不一定要使用任何用户出口，但是，如果您希望扩展和定制 IBM MQ Managed File Transfer 系统的功能以满足您的特定需求，那么用户出口会很有用。

在文件传输处理期间，有两个“点”可用于调用源系统的用户出口，还有两个“点”可用于调用目标系统的用户出口。下表汇总了这些用户出口点中的每个用户出口点以及必须实现才能使用这些出口点的 Java 接口。

表 23: 源端和目标端出口点以及 Java 接口的摘要	
出口点	要实现的 Java 接口
源端出口点：	
整个文件传输开始前	SourceTransferStartExit.java
整个文件传输完成后	SourceTransferEndExit.java
目标端出口点：	
整个文件传输开始前	DestinationTransferStartExit.java
整个文件传输完成后	DestinationTransferEndExit.java

用户出口的调用顺序如下：

- 1. SourceTransferStartExit
- 2. DestinationTransferStartExit
- 3. DestinationTransferEndExit
- 4. SourceTransferEndExit

SourceTransferStartExit 出口和 DestinationTransferStartExit 出口产生的更改会作为对后续出口的输入而传播。例如，如果 SourceTransferStartExit 出口修改了传输元数据，这些更改会反映在对其他出口的输入传输元数据中。

构建用户出口

用于构建用户出口的接口包含在 `MQ_INSTALL_DIRECTORY/mqft/lib/com.ibm.wmqfte.exitroutines.api.jar` 中。构建出口时，您必须在类路径中包含该 .jar 文件。要运行出口，请作为 .jar 文件抽取出口，然后将该 .jar 文件置于后面部分中描述的目录中。

用户出口位置

您可以在以下两个可能位置中存储用户出口例程：

- `exits` 目录。每个代理目录下存在一个 `exits` 目录。例如：
`var\mqm\mqft\config\QM_JUPITER\agents\AGENT1\exits`
- 您可以设置 `exitClassPath` 属性来指定一个备用位置。如果在 `exits` 目录以及由 `exitClassPath` 设置的类路径中同时存在出口类，那么 `exits` 目录下的类优先，这意味着，如果在这两个位置中具有名称相同的类，那么 `exits` 目录中的类优先。

配置代理以使用用户出口

有四种代理属性可进行设置用以指定代理调用的用户出口。sourceTransferStartExitClasses、sourceTransferEndExitClasses、destinationTransferStartExitClasses 和 destinationTransferEndExitClasses。要了解有关如何使用这些属性的信息，请参阅第 977 页的『用户出口的代理属性』。

在协议网桥代理上运行用户出口

当源代理调用出口时，它会向出口传递要传输的源项目列表。对于一般代理，这是标准文件名列表。因为这些文件应为本地文件（或可通过安装进行访问），所以出口能够对其进行访问和加密。

但是，对于协议网桥代理，列表中的条目为以下格式：

```
"<file server identifier>:<fully-qualified file name of the file on the remote file server>"
```

对于列表中的每个条目，出口需要首先连接到文件服务器（使用 FTP、FTPS 或 SFTP 协议）、下载文件、在本地对其加密，然后将加密后的文件重新上传到文件服务器。

在 Connect:Direct 网桥代理上运行用户出口

不能在 Connect:Direct 网桥代理上运行用户出口。

相关概念

第 359 页的『IBM MQ Managed File Transfer 源和目标用户出口例程』

第 971 页的『用户出口例程元数据』

可向 IBM MQ Managed File Transfer 的用户出口例程提供三种不同类型的元数据：环境、传输和文件元数据。此元数据显示为 Java 键/值对的映射。

第 979 页的『用户出口例程的 Java 接口』

使用本节中的主题以获取有关用户出口例程的 Java 接口的参考信息。

相关参考

第 364 页的『为用户出口启用远程调试』

在开发用户出口时，您可能希望使用调试器来帮助找到代码中的问题。

第 364 页的『源传输最终用户出口样本』

第 365 页的『协议网桥凭证用户出口样本』

第 974 页的『资源监视器用户出口』

通过资源监视器用户出口，您可以配置在满足监视器的触发条件的情况下，要在关联任务启动之前运行的定制代码。

第 977 页的『用户出口的代理属性』

除 agent.properties 文件中的标准属性外，还存在多个特定于用户出口例程的高级属性。缺省情况下不包含这些属性，因此要使用它们中的任何一个，都必须手动编辑 agent.properties 文件。如果在代理运行期间对 agent.properties 文件进行了更改，请停止并重新启动该代理以提取这些更改。

IBM MQ Managed File Transfer 源和目标用户出口例程

目录分隔符

无论在 **fteCreateTransfer** 命令或 IBM MQ Explorer 中如何指定目录分隔符，源文件规范中的目录分隔符始终使用正斜杠 (/) 字符表示。编写出口时必须将此情况考虑在内。例如，如果要检查是否存在以下源文件：c:\a\b.txt，并且您已使用 **fteCreateTransfer** 命令或 IBM MQ Explorer 指定此源文件，请注意文件名实际上存储为：c:/a/b.txt 因此，如果搜索 c:\a\b.txt 的原始字符串，那么将找不到匹配项。

源端出口点

整个文件传输开始前

如果暂挂传输列表的下一步是传输请求，并且传输即将启动，那么由源代理调用该出口。

该出口点的示例用途是，使用外部命令分阶段将文件发送至该代理具有读/写访问权的目录，或者是对目标系统上的文件进行重命名。

将以下自变量传递到该出口：

- 源代理名称
- 目标代理名称
- 环境元数据
- 传输元数据
- 文件规范（包含文件元数据）

从该出口返回的数据如下：

- 已更新的传输元数据。可添加、修改和删除条目。
- 已更新的文件规范列表，其中包含源文件和目标文件名称对。可添加、修改和删除条目
- 指示符，指示是否继续传输
- 字符串，用于插入“传输日志”。

实施 [SourceTransferStartExit.java](#) 接口，以在该出口点调用用户出口代码。

整个文件传输完成后

整个文件传输完成后，会由源代理调用该出口。

下面是该出口点的使用示例：执行某些完成任务，例如发送电子邮件或 IBM MQ 消息以标记传输已完成。

将以下自变量传递到该出口：

- 传输出口结果
- 源代理名称
- 目标代理名称
- 环境元数据
- 传输元数据
- 文件结果

从该出口返回的数据如下：

- 已更新的字符串，以用于插入“传输日志”。

实现 [SourceTransferEndExit.java](#) 接口，以在该出口点调用用户出口代码。

目标端出口点

整个文件传输开始前

使用该出口点的示例为：验证位于目标的许可权。

将以下自变量传递到该出口：

- 源代理名称
- 目标代理名称
- 环境元数据
- 传输元数据
- 文件规范

从该出口返回的数据如下：

- 已更新的目标文件名集。可修改条目，但是不可添加或删除条目。
- 指示符，指示是否继续传输
- 字符串，用于插入“传输日志”。

实现 [DestinationTransferStartExit.java](#) 接口，以在该出口点调用用户出口代码。

整个文件传输完成后

使用该用户出口的示例为：启动使用已传输的文件的批处理过程，或者传输失败时发送电子邮件。

将以下自变量传递到该出口：

- 传输出口结果
- 源代理名称
- 目标代理名称
- 环境元数据
- 传输元数据
- 文件结果

从该出口返回的数据如下：

- 已更新的字符串，以用于插入“传输日志”。

实现 [DestinationTransferEndExit.java](#) 接口，以在该出口点调用用户出口代码。

相关概念

第 979 页的『用户出口例程的 Java 接口』

使用本节中的主题以获取有关用户出口例程的 Java 接口的参考信息。

相关参考

第 364 页的『为用户出口启用远程调试』

在开发用户出口时，您可能希望使用调试器来帮助找到代码中的问题。

第 364 页的『源传输最终用户出口样本』

第 974 页的『资源监视器用户出口』

通过资源监视器用户出口，您可以配置在满足监视器的触发条件的情况下，要在关联任务启动之前运行的定制代码。

使用 IBM MQ Managed File Transfer 传输 I/O 用户出口

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的底层文件系统 I/O 工作。

通常对于 MQMFT 传输，代理会选择某个内置的 I/O 提供程序，以便与用于传输的相应文件系统进行交互。内置 I/O 提供程序支持以下类型的文件系统：

- 常规 UNIX 类型和 Windows 类型文件系统
- z/OS 连续的或分区数据集（仅限 z/OS 上）
- IBM i 本机保存文件（仅限 IBM i 上）
- IBM MQ 队列
- 远程 FTP 和 SFTP 协议服务器（仅用于协议网桥代理）
- 远程 Connect:Direct 节点（仅用于 Connect:Direct 网桥连接代理）

对于不受支持的文件系统或者其中需要定制 I/O 行为的文件系统，那么您可以编写一个传输 I/O 用户出口。

传输 I/O 用户出口将现有基础结构用于用户出口。但是，这些传输 I/O 用户出口与其他用户出口不同，因为在整个传输中，将对每个文件多次访问这些传输 I/O 用户出口的功能。

请使用代理属性 `IOExitClasses`（在 `agent.properties` 文件中）来指定要装入的 I/O 出口类。用逗号来分隔每个出口类，例如：

```
IOExitClasses=testExits.TestExit1,testExits.testExit2
```

传输 I/O 用户出口的 Java 接口如下所示：

IOExit

主入口点，用以确定是否使用 I/O 出口。此实例负责生成 IOExitPath 实例。

您只需为代理属性 IOExitClasses 指定 IOExit I/O 出口接口。

IOExitPath

表示一个抽象接口；例如，某个数据容器或者表示一组数据容器的通配符。您无法创建实现此接口的类实例。该接口允许对路径进行检查并列出派生的路径。IOExitResourcePath 和 IOExitWildcardPath 接口扩展了 IOExitPath。

IOExitChannel

使数据可以在 IOExitPath 资源上读取和写入。

IOExitRecordChannel

为面向记录的 IOExitPath 资源扩展 IOExitChannel 接口，这使得能够在多个记录中的 IOExitPath 资源上读取和写入数据。

IOExitLock

表示对 IOExitPath 资源的锁定以进行共享或互斥存取。

IOExitRecordResourcePath

扩展 IOExitResourcePath 接口以表示面向记录的文件的数据容器；例如，z/OS 数据集。您可以使用该接口来查找数据，从而为读/写操作创建 IOExitRecordChannel 实例。

IOExitResourcePath

扩展 IOExitPath 接口以表示数据容器；例如，文件或目录。您可以使用此接口来查找数据。如果此接口表示目录，那么您可以使用 listPaths 方法来返回路径的列表。

IOExitWildcardPath

扩展 IOExitPath 接口以表示指示通配符的路径。您可以使用此接口来匹配多个 IOExitResourcePaths。

IOExitProperties

指定用于确定 IBM MQ Managed File Transfer 如何处理某些 I/O 方面的 IOExitPath 的属性。例如，是否使用中间文件，或是否在重新启动传输时从开始重新读取资源。

相关概念

第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 985 页的『IOExit.java 接口』

第 989 页的『IOExitChannel.java 接口』

第 990 页的『IOExitLock.java 接口』

第 991 页的『IOExitPath.java 接口』

第 994 页的『IOExitProperties.java 接口』

第 996 页的『IOExitRecordChannel.java 接口』

第 998 页的『IOExitRecordResourcePath.java 接口』

第 1000 页的『IOExitResourcePath.java 接口』

第 1005 页的『IOExitWildcardPath.java 接口』

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 agent.properties，其中必须包含代理用于连接到其队列管理器的信息。agent.properties 文件还可以包含用于更改代理程序行为的属性。

样本 IBM i 用户出口

IBM MQ Managed File Transfer 随安装文件一起提供了特定于 IBM i 的样本用户出口。这些样本位于目录 `MQMFT_install_dir/samples/ioexit-IBMi` 和 `MQMFT_install_dir/samples/userexit-IBMi` 中。

com.ibm.wmqfte.exit.io.ibm.i.qdls.FTEQDLSExit

`com.ibm.wmqfte.exit.io.ibm.i.qdls.FTEQDLSExit` 样本用户出口传输 IBM i 上 QDLS 文件系统上的文件。安装该出口后，对于以 /QDLS 开头的文件的任何传输，都会自动使用该出口。

要安装该出口，请完成以下步骤：

1. 将 `com.ibm.wmqfte.samples.ibm.i.ioexits.jar` 文件从 `WMQFTE_install_dir/samples/ioexit-IBMi` 目录复制到代理的 `exits` 目录。
2. 将 `com.ibm.wmqfte.exit.io.ibm.i.qdls.FTEQDLSExit` 添加到 `IOExitClasses` 属性。
3. 重新启动代理。

com.ibm.wmqfte.exit.user.ibm.i.FileMemberMonitorExit

`com.ibm.wmqfte.exit.user.ibm.i.FileMemberMonitorExit` 样本用户出口的行为方式类似于 MQMFT 文件监视器，它自动传输 IBM i 库中的物理文件成员。

要运行此出口，请指定“`library.qsys.monitor`”元数据字段的值（例如，使用 `-md` 参数）。该参数采用文件成员的 IFS 样式路径，并且可以包含文件和成员通配符。例如，`/QSYS.LIB/FOO.LIB/BAR.FILE/*.MBR`、`/QSYS.LIB/FOO.LIB/*.FILE/BAR.MBR`、`/QSYS.LIB/FOO.LIB/*.FILE/*.MBR`。

该样本出口还具有一个可选元数据字段“`naming.scheme.qsys.monitor`”，此字段可用于确定传输期间使用的命名方案。缺省情况下，该字段设置为“`unix`”，这导致目标文件命名为 `FOO.MBR`。您也可以指定值“`ibmi`”以使用 IBM i FTP FILE.MEMBER 方案，例如 `/QSYS.LIB/FOO.LIB/BAR.FILE/BAZ.MBR` 将作为 `BAR.BAZ` 传输。

要安装该出口，请完成以下步骤：

1. 将 `com.ibm.wmqfte.samples.ibm.i.userexits.jar` 文件从 `WMQFTE_install_dir/samples/userexit-IBMi` 目录复制到代理的 `exits` 目录。
2. 将 `com.ibm.wmqfte.exit.user.ibm.i.FileMemberMonitorExit` 添加到 `agent.properties` 文件中的 `sourceTransferStartExitClasses` 属性。
3. 重新启动代理。

com.ibm.wmqfte.exit.user.ibm.i.EmptyFileDeleteExit

当在传输期间删除源文件成员时，`com.ibm.wmqfte.exit.user.ibm.i.EmptyFileDeleteExit` 样本用户出口将删除空文件对象。因为 IBM i 文件对象可能保存许多成员，所以 MQMFT 将文件对象视为目录处理。所以，无法使用 MQMFT 对文件对象执行移动操作；只有在成员级别才支持移动操作。因此，在对成员执行移动操作时，当前的空文件会遗留下来。如果希望在传输请求期间删除这些空文件，请使用该样本出口。

当为“`empty.file.delete`”元数据字段指定“`true`”并传输 `FTEFileMember` 时，如果父文件为空，该样本出口将删除该文件。

要安装该出口，请完成以下步骤：

1. 将 `com.ibm.wmqfte.samples.ibm.i.userexits.jar` 文件从 `WMQFTE_install_dir/samples/userexit-IBMi` 目录复制到代理的 `exits` 目录。
2. 将 `com.ibm.wmqfte.exit.user.ibm.i.EmptyFileDeleteExit` 添加到 `agent.properties` 文件中的 `sourceTransferStartExitClasses` 属性。
3. 重新启动代理。

相关参考

第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的底层文件系统 I/O 工作。

第 977 页的『用户出口的代理属性』

除 `agent.properties` 文件中的标准属性外，还存在多个特定于用户出口例程的高级属性。缺省情况下不包含这些属性，因此要使用它们中的任何一个，都必须手动编辑 `agent.properties` 文件。如果在代理运行期间对 `agent.properties` 文件进行了更改，请停止并重新启动该代理以提取这些更改。

为用户出口启用远程调试

在开发用户出口时，您可能希望使用调试器来帮助找到代码中的问题。

由于出口在运行代理程序的 Java 虚拟机中运行，因此无法使用通常包含在集成开发环境中的直接调试支持。但是，您可以启用 JVM 的远程调试，然后连接一个合适的远程调试器。

要启用远程调试，请使用标准 JVM 参数 `-Xdebug` 和 `-Xrunjdwp`。这些属性会通过 `BFG_JVM_PROPERTIES` 环境变量传递到运行代理的 JVM。例如，在 UNIX 上，以下命令将启动代理，并使 JVM 在 TCP 端口 8765 上侦听调试器连接。

```
export BFG_JVM_PROPERTIES="-Xdebug -Xrunjdwp:transport=dt_socket,server=y,address=8765"
fteStartAgent -F TEST_AGENT
```

在调试器连接之前，代理不会启动。在 Windows 上使用 `set` 命令，而不是 `export` 命令。

您也可以在调试器和 JVM 之间使用其他通信方法。例如，JVM 可以打开与调试器的连接，反之则不可，或者您也可以使用共享内存来代替 TCP。请参阅 [Java 平台调试器体系结构](#) 文档以获取更多详细信息。

以远程调试方式启动代理时，必须使用 `-F`（前台）参数。

使用 Eclipse 调试器

以下步骤适用于 Eclipse 开发环境中的远程调试功能。您还可以使用与 JPDA 兼容的其他远程调试器。

1. 单击运行 > 打开调试对话框（或运行 > 调试配置或运行 > 调试对话框，具体取决于 Eclipse 的版本）。
2. 在配置类型列表中双击远程 **Java** 应用程序以创建一个调试配置。
3. 填写配置字段，然后保存调试配置。如果已通过调试方式启动了代理 JVM，那么现在可以连接到 JVM。

源传输最终用户出口样本

```
/*
 * A Sample Source Transfer End Exit that prints information about a transfer to standard
 * output.
 * If the agent is run in the background the output will be sent to the agent's event log file.
 * If
 * the agent is started in the foreground by specifying the -F parameter on the fteStartAgent
 * command the output will be sent to the console.
 *
 * To run the exit execute the following steps:
 *
 * Compile and build the exit into a jar file. You need the following in the class path:
 * {MQ_INSTALLATION_PATH}\mqft\lib\com.ibm.wmqfte.exitroutines.api.jar
 *
 * Put the jar in your agent's exits directory:
 * {MQ_DATA_PATH}\config\<coordQmgrName>\agents\<agentName>\exits\
 *
 * Update the agent's properties file:
 * {MQ_DATA_PATH}\config\<coordQmgrName>\agents\<agentName>\agent.properties
 * to include the following property:
 * sourceTransferEndExitClasses=[<packageName>.]SampleEndExit
 *
 * Restart agent to pick up the exit
 *
 * Send the agent a transfer request:
 * For example: fteCreateTransfer -sa myAgent -da YourAgent -df output.txt input.txt
 */

import java.util.List;
import java.util.Map;
import java.util.Iterator;

import com.ibm.wmqfte.exitroutine.api.SourceTransferEndExit;
```

```

import com.ibm.wmqfte.exitroutine.api.TransferExitResult;
import com.ibm.wmqfte.exitroutine.api.FileTransferResult;

public class SampleEndExit implements SourceTransferEndExit {

    public String onSourceTransferEnd(TransferExitResult transferExitResult,
        String sourceAgentName,
        String destinationAgentName,
        Map<String, String>environmentMetaData,
        Map<String, String>transferMetaData,
        List<FileTransferResult>fileResults) {

        System.out.println("Environment Meta Data: " + environmentMetaData);
        System.out.println("Transfer Meta Data: " + transferMetaData);

        System.out.println("Source agent: " +
            sourceAgentName);
        System.out.println("Destination agent: " +
            destinationAgentName);

        if (fileResults.isEmpty()) {
            System.out.println("No files in the list");
            return "No files";
        }
        else {

            System.out.println( "File list: ");

            final Iterator<FileTransferResult> iterator = fileResults.iterator();

            while (iterator.hasNext()){
                final FileTransferResult thisFileSpec = iterator.next();
                System.out.println("Source file spec: " +
                    thisFileSpec.getSourceFileSpecification() +
                    ", Destination file spec: " +
                    thisFileSpec.getDestinationFileSpecification());
            }
        }
        return "Done";
    }
}

```

协议网桥凭证用户出口样本

有关如何使用该样本用户出口的信息，请参阅第 283 页的『使用出口类映射文件服务器的凭证』

```

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Enumeration;
import java.util.HashMap;
import java.util.Map;
import java.util.Properties;
import java.util.StringTokenizer;

import com.ibm.wmqfte.exitroutine.api.CredentialExitResult;
import com.ibm.wmqfte.exitroutine.api.CredentialExitResultCode;
import com.ibm.wmqfte.exitroutine.api.CredentialPassword;
import com.ibm.wmqfte.exitroutine.api.CredentialUserId;
import com.ibm.wmqfte.exitroutine.api.Credentials;
import com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit;

/**
 * A sample protocol bridge credential exit
 *
 * This exit reads a properties file that maps mq user ids to server user ids
 * and server passwords. The format of each entry in the properties file is:
 *
 * mqUserId=serverUserId,serverPassword
 *
 * The location of the properties file is taken from the protocol bridge agent
 * property protocolBridgeCredentialConfiguration.
 *
 * To install the sample exit compile the class and export to a jar file.
 */

```

```

* Place the jar file in the exits subdirectory of the agent data directory
* of the protocol bridge agent on which the exit is to be installed.
* In the agent.properties file of the protocol bridge agent set the
* protocolBridgeCredentialExitClasses to SampleCredentialExit
* Create a properties file that contains the mqUserId to serverUserId and
* serverPassword mappings applicable to the agent. In the agent.properties
* file of the protocol bridge agent set the protocolBridgeCredentialConfiguration
* property to the absolute path name of this properties file.
* To activate the changes stop and restart the protocol bridge agent.
*
* For further information on protocol bridge credential exits refer to
* the WebSphere MQ Managed File Transfer documentation online at:
* https://www.ibm.com/docs/SSEP7X_7.0.4/welcome/WelcomePagev7r0.html
*/
public class SampleCredentialExit implements ProtocolBridgeCredentialExit {

    // The map that holds mq user id to serverUserId and serverPassword mappings
    final private Map<String,Credentials> credentialsMap = new HashMap<String, Credentials>();

    /* (non-Javadoc)
     * @see com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit#initialize(java.util.Map)
     */
    public synchronized boolean initialize(Map<String, String> bridgeProperties) {

        // Flag to indicate whether the exit has been successfully initialized or not
        boolean initialisationResult = true;

        // Get the path of the mq user id mapping properties file
        final String propertiesFilePath = bridgeProperties.get("protocolBridgeCredentialConfiguration");

        if (propertiesFilePath == null || propertiesFilePath.length() == 0) {
            // The properties file path has not been specified. Output an error and return false
            System.err.println("Error initializing SampleCredentialExit.");
            System.err.println("The location of the mqUserId mapping properties file has not been
specified in the
protocolBridgeCredentialConfiguration property");
            initialisationResult = false;
        }

        if (initialisationResult) {

            // The Properties object that holds mq user id to serverUserId and serverPassword
            // mappings from the properties file
            final Properties mappingProperties = new Properties();

            // Open and load the properties from the properties file
            final File propertiesFile = new File (propertiesFilePath);
            FileInputStream inputStream = null;
            try {
                // Create a file input stream to the file
                inputStream = new FileInputStream(propertiesFile);

                // Load the properties from the file
                mappingProperties.load(inputStream);
            }
            catch (FileNotFoundException ex) {
                System.err.println("Error initializing SampleCredentialExit.");
                System.err.println("Unable to find the mqUserId mapping properties file: " +
propertiesFilePath);
                initialisationResult = false;
            }
            catch (IOException ex) {
                System.err.println("Error initializing SampleCredentialExit.");
                System.err.println("Error loading the properties from the mqUserId mapping properties
file: " + propertiesFilePath);
                initialisationResult = false;
            }
            finally {
                // Close the inputStream
                if (inputStream != null) {
                    try {
                        inputStream.close();
                    }
                    catch (IOException ex) {
                        System.err.println("Error initializing SampleCredentialExit.");
                        System.err.println("Error closing the mqUserId mapping properties file: " +
propertiesFilePath);
                        initialisationResult = false;
                    }
                }
            }
        }
    }
}

```

```

        if (initialisationResult) {
            // Populate the map of mqUserId to server credentials from the properties
            final Enumeration<?> propertyNames = mappingProperties.propertyNames();
            while ( propertyNames.hasMoreElements()) {
                final Object name = propertyNames.nextElement();
                if (name instanceof String ) {
                    final String mqUserId = ((String)name).trim();
                    // Get the value and split into serverUserId and serverPassword
                    final String value = mappingProperties.getProperty(mqUserId);
                    final StringTokenizer valueTokenizer = new StringTokenizer(value, ",");
                    String serverUserId = "";
                    String serverPassword = "";
                    if (valueTokenizer.hasMoreTokens()) {
                        serverUserId = valueTokenizer.nextToken().trim();
                    }
                    if (valueTokenizer.hasMoreTokens()) {
                        serverPassword = valueTokenizer.nextToken().trim();
                    }
                    // Create a Credential object from the serverUserId and serverPassword
                    final Credentials credentials = new Credentials(new CredentialUserId(serverUserId), new
                    CredentialPassword(serverPassword));
                    // Insert the credentials into the map
                    credentialsMap.put(mqUserId, credentials);
                }
            }
        }

        return initialisationResult;
    }
    /* (non-Javadoc)
     * @see com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit#mapMQUserId(java.lang.String)
     */
    public synchronized CredentialExitResult mapMQUserId(String mqUserId) {
        CredentialExitResult result = null;
        // Attempt to get the server credentials for the given mq user id
        final Credentials credentials = credentialsMap.get(mqUserId.trim());
        if ( credentials == null) {
            // No entry has been found so return no mapping found with no credentials
            result = new CredentialExitResult(CredentialExitResultCode.NO_MAPPING_FOUND, null);
        }
        else {
            // Some credentials have been found so return success to the user along with the credentials
            result = new CredentialExitResult(CredentialExitResultCode.USER_SUCCESSFULLY_MAPPED,
            credentials);
        }
        return result;
    }
    /* (non-Javadoc)
     * @see com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit#shutdown(java.util.Map)
     */
    public void shutdown(Map<String, String> bridgeProperties) {
        // Nothing to do in this method because there are no resources that need to be released
    }
}

```

协议网桥属性用户出口样本

有关如何使用该样本用户出口的信息，请参阅第 277 页的『使用出口类 (ProtocolBridgePropertiesExit2) 查找协议文件服务器属性』

SamplePropertiesExit2.java

```

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Properties;

import com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2;
import com.ibm.wmqfte.exitroutine.api.ProtocolServerPropertyConstants;

```

```

/**
 * A sample protocol bridge properties exit. This exit reads a properties file
 * that contains properties for protocol servers.
 * <p>
 * The format of each entry in the properties file is:
 * {@literal <serverName>=<type>://<host>:<port>}}
 * Ensure there is a default entry such as
 * {@literal default=<type>://<host>:<port>}}
 * otherwise the agent will fail to start with a BFGBR0168 as it must have a
 * default server.
 * <p>
 * The location of the properties file is taken from the protocol bridge agent
 * property {@code protocolBridgePropertiesConfiguration}.
 * <p>
 * The methods {@code getCredentialLocation} returns the location of the associated
 * ProtocolBridgeCredentials.xml, this sample it is defined to be stored in a directory
 * defined by the environment variable CREDENTIALSHOME
 * <p>
 * To install the sample exit:
 * <ol>
 * <li>Compile the class and export to a jar file.
 * <li>Place the jar file in the {@code exits} subdirectory of the agent data directory
 * of the protocol bridge agent on which the exit is to be installed.
 * <li>In the {@code agent.properties} file of the protocol bridge agent
 * set the {@code protocolBridgePropertiesExitClasses} to
 * {@code SamplePropertiesExit2}.
 * <li>Create a properties file that contains the appropriate properties to specify the
 * required servers.
 * <li>In the {@code agent.properties} file of the protocol bridge agent
 * set the <code>protocolBridgePropertiesConfiguration</code> property to the
 * absolute path name of this properties file.
 * <li>To activate the changes stop and restart the protocol bridge agent.
 * </ol>
 * <p>
 * For further information on protocol bridge properties exits refer to the
 * WebSphere MQ Managed File Transfer documentation online at:
 * <p>
 * {@link https://www.ibm.com/docs/SSEP7X_7.0.4/welcome/WelcomePagev7r0.html}}
 */
public class SamplePropertiesExit2 implements ProtocolBridgePropertiesExit2 {

    /**
     * Helper class to encapsulate protocol server information.
     */
    private static class ServerInformation {
        private final String type;
        private final String host;
        private final int port;

        public ServerInformation(String url) {
            int index = url.indexOf("://");
            if (index == -1) throw new IllegalArgumentException("Invalid server URL: "+url);
            type = url.substring(0, index);

            int portIndex = url.indexOf(":", index+3);
            if (portIndex == -1) {
                host = url.substring(index+3);
                port = -1;
            } else {
                host = url.substring(index+3, portIndex);
                port = Integer.parseInt(url.substring(portIndex+1));
            }
        }

        public String getType() {
            return type;
        }

        public String getHost() {
            return host;
        }

        public int getPort() {
            return port;
        }
    }

    /** A {@code Map} that holds information for each configured protocol server */
    final private Map<String, ServerInformation> servers = new HashMap<String, ServerInformation>();

    /* (non-Javadoc)

```



```

    * @see
com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit#getProtocolServerProperties(java.lang.String)
    */
    public Properties getProtocolServerProperties(String protocolServerName) {
        // Attempt to get the protocol server information for the given protocol server name
        // If no name has been supplied then this implies the default.
        final ServerInformation info;
        if (protocolServerName == null || protocolServerName.length() == 0) {
            protocolServerName = "default";
        }
        info = servers.get(protocolServerName);

        // Build the return set of properties from the collected protocol server information, when
        // available.
        // The properties set here is the minimal set of properties to be a valid set.
        final Properties result;
        if (info != null) {
            result = new Properties();
            result.setProperty(ProtocolServerPropertyConstants.SERVER_NAME, protocolServerName);
            result.setProperty(ProtocolServerPropertyConstants.SERVER_TYPE, info.getType());
            result.setProperty(ProtocolServerPropertyConstants.SERVER_HOST_NAME, info.getHost());
            if (info.getPort() != -1)
                result.setProperty(ProtocolServerPropertyConstants.SERVER_PORT_VALUE, ""+info.getPort());
            result.setProperty(ProtocolServerPropertyConstants.SERVER_PLATFORM, "UNIX");
            if (info.getType().toUpperCase().startsWith("FTP")) { // FTP & FTPS
                result.setProperty(ProtocolServerPropertyConstants.SERVER_TIMEZONE, "Europe/London");
                result.setProperty(ProtocolServerPropertyConstants.SERVER_LOCALE, "en-GB");
            }
            result.setProperty(ProtocolServerPropertyConstants.SERVER_FILE_ENCODING, "UTF-8");
        } else {
            System.err.println("Error no default protocol file server entry has been supplied");
            result = null;
        }

        return result;
    }

    /* (non-Javadoc)
    * @see com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit#initialize(java.util.Map)
    */
    public boolean initialize(Map<String, String> bridgeProperties) {
        // Flag to indicate whether the exit has been successfully initialized or not
        boolean initialisationResult = true;

        // Get the path of the properties file
        final String propertiesFilePath = bridgeProperties.get("protocolBridgePropertiesConfiguration");
        if (propertiesFilePath == null || propertiesFilePath.length() == 0) {
            // The protocol server properties file path has not been specified. Output an error and
            return false;
            System.err.println("Error initializing SamplePropertiesExit.");
            System.err.println("The location of the protocol server properties file has not been
            specified in the
            protocolBridgePropertiesConfiguration property");
            initialisationResult = false;
        }

        if (initialisationResult) {
            // The Properties object that holds protocol server information
            final Properties mappingProperties = new Properties();

            // Open and load the properties from the properties file
            final File propertiesFile = new File (propertiesFilePath);
            FileInputStream inputStream = null;
            try {
                // Create a file input stream to the file
                inputStream = new FileInputStream(propertiesFile);

                // Load the properties from the file
                mappingProperties.load(inputStream);
            } catch (final FileNotFoundException ex) {
                System.err.println("Error initializing SamplePropertiesExit.");
                System.err.println("Unable to find the protocol server properties file: " +
                propertiesFilePath);
                initialisationResult = false;
            } catch (final IOException ex) {
                System.err.println("Error initializing SamplePropertiesExit.");
                System.err.println("Error loading the properties from the protocol server properties
                file: " + propertiesFilePath);
                initialisationResult = false;
            } finally {
                // Close the inputStream
                if (inputStream != null) {

```

```

        try {
            inputStream.close();
        } catch (final IOException ex) {
            System.err.println("Error initializing SamplePropertiesExit.");
            System.err.println("Error closing the protocol server properties file: " +
propertiesFilePath);
            initialisationResult = false;
        }
    }
}

if (initialisationResult) {
    // Populate the map of protocol servers from the properties
    for (Entry<Object, Object> entry : mappingProperties.entrySet()) {
        final String serverName = (String)entry.getKey();
        final ServerInformation info = new ServerInformation((String)entry.getValue());
        servers.put(serverName, info);
    }
}

return initialisationResult;
}

/* (non-Javadoc)
 * @see com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit#shutdown(java.util.Map)
 */
public void shutdown(Map<String, String> bridgeProperties) {
    // Nothing to do in this method because there are no resources that need to be released
}

/* (non-Javadoc)
 * @see com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2#getCredentialLocation()
 */
public String getCredentialLocation() {
    String envLocationPath;
    if (System.getProperty("os.name").toLowerCase().contains("win")) {
        // Windows style
        envLocationPath = "%CREDENTIALSHOME%\\ProtocolBridgeCredentials.xml";
    }
    else {
        // Unix style
        envLocationPath = "$CREDENTIALSHOME/ProtocolBridgeCredentials.xml";
    }
    return envLocationPath;
}
}

```

通过将消息放置在代理命令队列中来控制 IBM MQ Managed File Transfer

您可编写一个应用程序，通过将消息放置在代理命令队列中来控制 IBM MQ Managed File Transfer。

您可以将一条消息放置在代理的命令队列中，以请求代理执行以下某项操作：

- 创建一次文件传输
- 创建一次计划的文件传输
- 取消一次文件传输
- 取消一次计划的文件传输
- 调用一个命令
- 创建一个监视器
- 删除一个监视器
- 返回 ping 以表明代理处于活动状态

要请求代理执行这些操作之一，消息必须是采用以下某种模式编译的 XML 格式：

FileTransfer.xsd

这种格式的消息可用于创建文件传输或计划的文件传输，调用命令，或者取消文件传输或计划的文件传输。有关更多信息，请参阅第 842 页的『文件传输请求消息格式』。

Monitor.xsd

这种格式的消息可用于创建或删除资源监视器。有关更多信息，请参阅第 859 页的『监视器请求消息格式』。

PingAgent.xsd

这种格式的消息可用于 ping 代理，以检查其是否处于活动状态。有关更多信息，请参阅第 868 页的『Ping 代理请求消息格式』。

该代理返回对请求消息的应答。应答消息会放入在请求消息中定义的应答队列。应答消息采用通过以下模式定义的 XML 格式：

Reply.xsd

有关更多信息，请参阅第 870 页的『应答消息格式』。

IBM MQ Managed File Transfer 故障诊断

使用以下参考信息可帮助您诊断 IBM MQ Managed File Transfer 中的错误：

相关概念

第 371 页的『一般故障诊断』

使用以下参考信息可帮助您诊断 IBM MQ Managed File Transfer 中的错误：

第 412 页的『对 Web 网关进行故障诊断』

使用以下参考信息和示例可帮助您诊断 Web 网关返回的错误。

第 423 页的『对 Connect:Direct 网桥进行故障诊断』

使用以下参考信息和示例，以帮助诊断 Connect:Direct 网桥返回的错误。

第 95 页的『对 IBM 4690 系统进行故障诊断』

使用以下参考信息可帮助您诊断从 IBM 4690 系统返回的错误。

一般故障诊断

使用以下参考信息可帮助您诊断 IBM MQ Managed File Transfer 中的错误：

相关概念

第 392 页的『使用 IBM MQ Managed File Transfer 的提示与技巧』

以下建议可帮助您最有效地使用 IBM MQ Managed File Transfer：

第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』

您可以作为 Windows 服务来运行 IBM MQ Managed File Transfer 代理、独立数据库记录器和独立文件记录器。如果这些 Windows 服务出现任何问题，您可以使用服务日志文件以及本主题中的信息来诊断问题。

相关任务

第 402 页的『如果在 Oracle 数据库上更新数据库模式时收到错误』

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。

相关参考

第 372 页的『在 IBM MQ Managed File Transfer 上运行跟踪』

您可以使用以下方法跟踪 IBM MQ Managed File Transfer。

第 376 页的『常见问题』

IBM MQ Managed File Transfer 网络中可能出现的常见问题。

第 378 页的『如果 `fteListAgents` 命令未列出您的代理，该怎么办』

如果 `fteListAgents` 命令未列出您的代理、您的代理未显示在 IBM MQ Explorer 中，或者您的文件传输未显示在 IBM MQ Explorer 的“传输日志”中，那么可以执行若干问题确定步骤来调查原因。

第 380 页的『如果代理进程消失但未记录任何诊断信息，该怎么办？』

在 UNIX 平台上，如果代理进程已消失，但代理日志文件不包含任何解释，这可能由于启动此代理的方式所致。

第 387 页的『在您认为传输卡住时应该怎么办』

在高负荷的系统上或者在源和目标代理之间发生网络问题时，传输可能会偶尔显示为卡住，进入排队或恢复状态。导致此情况的因素有很多。

[第 387 页的『当协议网桥代理报告找不到某个文件时应如何处理』](#)

当协议网桥代理报告协议网桥所连接的 SFTP 或 FTP 服务器返回 File not found 错误消息时，此消息可能表示发生了许多不同的错误情况之一。

[第 389 页的『如果队列资源监视器启动的传输所创建的目标文件包含错误数据，那么该执行哪些操作？』](#)

您可以创建资源监视器来监视队列，并将队列上的消息或一组消息传输到文件。可以通过使用消息上的 MQMD 消息述符描或组中的第一个消息来指定文件名。如果消息到文件的传输失败，并且消息或组留在队列上，那么下次触发监视器时，可能导致创建的文件中包含错误数据。

[第 391 页的『如果消息在 SYSTEM.MANAGED.DURABLE 队列上聚集或者填充文件系统，那么该执行哪些操作？』](#)

如果 IBM MQ Explorer 插件在协调队列管理器上使用持久预订，那么消息可能会在 SYSTEM.MANAGED.DURABLE 队列上聚集。如果您具有大容量 IBM MQ Managed File Transfer 网络，请不要频繁使用 IBM MQ Explorer 插件，否则此消息数据可能会填满本地文件系统。

[第 391 页的『发布前检查消息』](#)

由于代理程序可以连接到 WebSphere MQ V 6 队列管理器，因此代理程序不会使用 WebSphere MQ V 7 中引入的直接发布方法。事实上，代理将普通消息发送到包含 MQRFH 头的协调队列管理器。MQRFH 头请求发布消息的有效内容。这些消息会被发送到协调队列管理器上的 SYSTEM.FTE 队列，通常会立即从该队列发布这些消息。如果错误条件阻止此发布，您可以在尝试发布前检查队列上的消息以帮助进行诊断。可通过完成以下步骤来执行此操作：

[第 393 页的『传输 IBM i 保存文件时可能发生的错误』](#)

如果您使用 IBM MQ Managed File Transfer 多次传输同一个 IBM i 保存文件，那么传输可能失败。

[第 394 页的『用于设置与消息大小关联的 IBM MQ 属性和 IBM MQ Managed File Transfer 属性的指南』](#)

在读取或写入各种大小的消息时，您可以更改 IBM MQ 属性和 IBM MQ Managed File Transfer 属性以影响 IBM MQ Managed File Transfer 的行为。

[第 402 页的『记录器错误处理和拒绝』](#)

记录器确定两种类型的错误：每条消息的错误和通用错误。

[第 403 页的『如果记录器已启动，但没有任何传输信息记录到数据库』](#)

IBM MQ 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。

[第 530 页的『fteDisplayVersion（显示 IBM MQ Managed File Transfer 的版本）』](#)

使用 **fteDisplayVersion** 命令来显示您已安装的 IBM MQ Managed File Transfer 的版本。

[第 405 页的『IBM MQ Managed File Transfer 的返回码』](#)

IBM MQ Managed File Transfer 命令、Ant 任务和日志消息提供返回码以指示是否成功完成了功能。

在 IBM MQ Managed File Transfer 上运行跟踪

您可以使用以下方法跟踪 IBM MQ Managed File Transfer。

- 通过使用 **fteSetAgentTraceLevel** 命令动态地更改代理跟踪的当前级别。
- 通过使用 **fteSetLoggerTraceLevel** 命令动态地更改记录器跟踪的当前级别。
- 通过使用 **-trace** 参数来跟踪任何 **fte** 命令。有关更多信息，请参阅[跟踪命令](#)。
- 通过在 `agent.properties` 文件中设置跟踪属性来配置代理以开始启用跟踪。有关更多信息，请参阅[高级代理属性](#)。

相关参考

[第 372 页的『跟踪 IBM MQ Managed File Transfer 命令』](#)

您可以跟踪任何 IBM MQ Managed File Transfer 命令以帮助通过命令行进行问题确定。

[第 374 页的『fteSetAgentTraceLevel（设置 IBM MQ Managed File Transfer 代理跟踪级别）』](#)

使用 **fteSetAgentTraceLevel** 命令可动态修改代理的当前跟踪级别。

[第 530 页的『fteDisplayVersion（显示 IBM MQ Managed File Transfer 的版本）』](#)

使用 **fteDisplayVersion** 命令来显示您已安装的 IBM MQ Managed File Transfer 的版本。

跟踪 IBM MQ Managed File Transfer 命令

您可以跟踪任何 IBM MQ Managed File Transfer 命令以帮助通过命令行进行问题确定。

用途

将 **-trace** 参数用于任何命令以启用指定级别的跟踪。生成的跟踪文件存放在您的当前工作目录中，除非包含 **-tracePath** 参数以标识其他目录。

因为运行跟踪可能会严重影响性能，并且可能会产生大量的跟踪数据，所以请谨慎运行跟踪并仅在必要时运行。通常情况下，请仅在您的 IBM 服务代表要求时启用跟踪。

您可以在 `agent.properties` 文件中设置更多跟踪属性，例如，跟踪文件大小和要保留的跟踪文件数量。[高级代理属性](#)中描述了这些属性。

语法

```
► fteCommandName — -trace — (classes=level) — -tracePath — (directory path) — ◄
```

参数

-trace (*classes=level*)

必需。要设置跟踪的级别以及要应用跟踪的类。指定以下格式：

```
classes=level
```

例如：

```
com.ibm.wmqfte=all
```

跟踪所有 IBM MQ Managed File Transfer 类。

指定想要应用跟踪级别的以冒号分隔的类规范列表。如果未指定该参数，那么跟踪级别将应用于所有代理类。

如果 (*classes*) 以加号 (+) 开头，那么将向当前正在跟踪的任何现有跟踪类添加跟随加号的跟踪类列表。

有效的跟踪级别选项如下所述，按照跟踪文件大小的升序和详细信息列出：

关

关闭代理跟踪，但继续将信息写入日志文件。这是缺省选项。

flow

捕获与代理中处理流相关联的跟踪点的数据。

moderate

捕获跟踪中的适量诊断信息。

详细

捕获跟踪中的大量诊断信息。

all

将代理跟踪设置为在所有代理类上运行。

-tracePath (*directory path*)

可选。指定想要写入跟踪的目录。例如，`c:\temp`。

如果未指定该参数，那么值为从中发出命令的目录。例如，在 z/OS 上：

```
/u/smith/fte/wmqmft/mqft/logs/MQPV/loggers/BFGLG1/logs/
```

仅当指定 **-trace** 参数时，该参数才有效。

示例

在此示例中，跟踪级别将设置为 all，意味着将针对 **fteStartAgent** 命令跟踪属于 AGENT.NAME 的所有类：

注：当代理程序启动时，跟踪将转至 <mft config>/logs<coordination qmgr>/agents/<agent>

```
fteStartAgent -trace com.ibm.wmqfte=all -tracePath /u/mft/trace AGENT.NAME
```

在此示例中，对于代理 AGENT.NAME 的 com.ibm.wmqfte.common 类，跟踪级别将设置为 moderate。将针对 **ftePingAgent** 命令捕获适量的跟踪：

```
ftePingAgent -trace com.ibm.wmqfte.common=moderate AGENT.NAME
```

在此示例中，对于代理 AGENT.NAME 的 com.ibm.wmqfte.common 类，跟踪级别将设置为 moderate，并且跟踪将写入 c:\\$user 目录。将针对 **ftePingAgent** 命令捕获适量的跟踪：

```
ftePingAgent -trace com.ibm.wmqfte.common=moderate -tracePath c:\$user AGENT.NAME
```

fteSetAgentTraceLevel（设置 IBM MQ Managed File Transfer 代理跟踪级别）

使用 **fteSetAgentTraceLevel** 命令可动态修改代理的当前跟踪级别。

用途

使用该命令可以开启和关闭代理跟踪，或更改所设置的代理跟踪级别。当使用 **fteSetAgentTraceLevel** 命令时，不必关闭和重新启动代理就可修改跟踪级别。生成的跟踪文件位于 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%/trace%PID%.txt` 中，其中 %PID% 是代理实例的进程标识。



注意：在分布式平台上使用 IBM WebSphere MQ 7.5 或更高版本时，只有作为代理进程运行身份的用户才可以运行 **fteSetAgentTraceLevel** 命令。

V8.0.0.6 对于 z/OS，**fteSetAgentTraceLevel** 命令可以由以下任一人员运行：

- 用于运行代理进程的同一用户标识。
- 代理属性 **adminGroup** 指定的组的成员。

有关更多信息，请参阅第 591 页的『agent.properties 文件』中的 **adminGroup** 属性。

在 IBM MQ Managed File Transfer V 7.5 和更高版本中，**fteSetAgentTraceLevel** 命令还会写入代理进程控制器的跟踪。生成的跟踪文件位于 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/pctrace%PID%/pctrace%PID%.txt` 中，其中 %PID% 是代理实例的进程标识。

您还可以使用该命令，让代理进程生成 Javacore。代理会在以下目录中生成一个 Javacore 文件：`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name`。

因为运行跟踪可能会严重影响性能，并且可能会产生大量的跟踪数据，所以请谨慎运行跟踪并仅在必要时运行。通常情况下，请仅在您的 IBM 服务代表要求时启用跟踪。



注意：

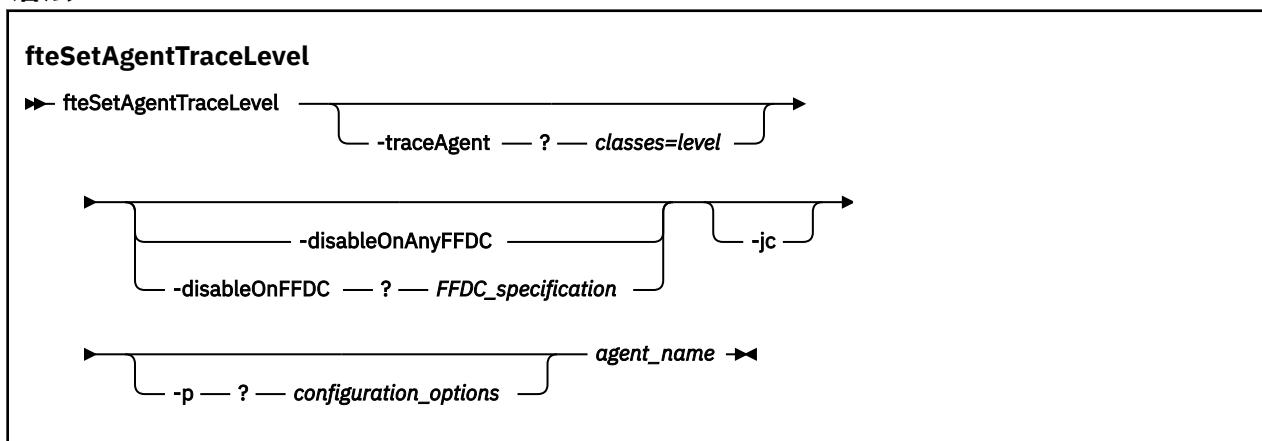
1. 必须在运行代理程序的系统上运行此命令。
2. 跟踪和日志记录不会在代理程序重新启动时持久存储。

如果代理程序终止并由 Process Controller 进程重新启动，那么在更新 agent.properties 文件以包含必需的跟踪和日志属性之前，动态跟踪和日志不会生效。

您可以在 agent.properties 文件中设置更多跟踪属性，例如，跟踪文件大小和要保留的跟踪文件数量。高级代理属性中描述了这些属性。

仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 -p 参数。请参阅第 591 页的『agent.properties 文件』，以了解更多信息。

语法



参数

-traceAgent classes=level

必需。将代理跟踪设置到的级别和将该跟踪应用到的类。指定以下格式：

```
classes=level
```

例如：

```
com.ibm.wmqfte=all
```

指定您希望将该跟踪级别应用到的类规范的逗号分隔列表。如果未指定该参数，那么跟踪级别将应用于所有代理类。

您可以使用 MQMFT 程序包名称来代替类，以仅跟踪特定的程序包。但是，由于此选项仅捕获代理的部分行为，所以通常不建议您使用程序包过滤。

如果 (classes) 以加号 (+) 开头，那么将向当前正在跟踪的任何现有跟踪类添加跟随加号的跟踪类列表。

有效的跟踪级别选项如下所述，按照跟踪文件大小的升序和详细信息列出：

关

关闭代理跟踪，但继续将信息写入日志文件。这是缺省选项。

flow

捕获与代理中处理流相关联的跟踪点的数据。

moderate

捕获跟踪中的适量诊断信息。

详细

捕获跟踪中的大量诊断信息。

all

将代理跟踪设置为在所有代理类上运行。

要开始全面跟踪代理，请运行以下命令：

```
fteSetAgentTraceLevel -traceAgent =all AGENT_NAME
```

要停止全面跟踪代理，请运行以下命令：

```
fteSetAgentTraceLevel -traceAgent =off AGENT_NAME
```

-disableOnAnyFFDC

可选。如果指定了该参数，那么当代理生成“首次故障数据捕获”(FFDC) 文件时，跟踪将被禁用。

您只能指定 **-disableOnAnyFFDC** 和 **-disableOnFFDC** 参数中的一个。

-disableOnFFDC *FFDC_specification*

可选。如果指定了该参数，那么当代理生成与 *FFDC_specification* 匹配的“首次故障数据捕获”(FFDC) 文件时，跟踪将被禁用。*FFDC_specification* 是以逗号分隔的值列表。值的格式可以是：

class_name

FFDC 继承的类的名称。例如，`com.ibm.wmqfte.classA`。

class_name:probe_ID

FFDC 继承的类中的类的名称和位置的探测器标识。例如，`com.ibm.wmqfte.classB:1`。

您只能指定 **-disableOnAnyFFDC** 和 **-disableOnFFDC** 参数中的一个。

-jc

可选。请求代理来生成 Javacore 文件。IBM 服务团队可能会要求您使用该参数来运行命令，以帮助诊断问题。此参数不能与除 **-p** 以外的任何其他参数一起使用。

-p *configuration_options*

可选。该参数确定用于设置代理跟踪级别的配置选项集。使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

agent_name

必需。您要为其设置跟踪级别的 IBM MQ Managed File Transfer 代理的名称。

-? or -h

可选。显示命令语法。

示例

在本示例中，针对 AGENT1 的所有类，跟踪级别都设置为 all：

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte=all AGENT1
```

在本示例中，针对 AGENT1 的 `com.ibm.wmqfte.agent.Agent` 和 `com.ibm.wmqfte.cmdhandler` 类，跟踪级别设置为 all：

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.agent.Agent,com.ibm.wmqfte.cmdhandler=moderate AGENT1
```

在本示例中，由于 **-traceLevel** 参数设置为 off，因此排除了对子类的跟踪。以 `com.ibm.outer` 开头的类（除以 `com.ibm.outer.inner` 开头的类外）都按详细级别进行跟踪：

```
fteSetAgentTraceLevel -traceAgent com.ibm.outer=verbose AGENT1  
fteSetAgentTraceLevel -traceAgent +com.ibm.outer.inner=off AGENT1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

常见问题

IBM MQ Managed File Transfer 网络中可能出现的常见问题。

- 如果文本传输失败，并具有以下错误：

```
BFGIO0060E: Text data conversion has failed
```

发生这种情况有两种原因：

1. 源文件中的一个或多个字符无法从源文件代码页转换到目标文件代码页。如果代码页具有不同字符集并且一些字符无法在这些字符集之间转换，那么可能会出现此问题。

如果您可以接受不对某些字符进行转换，那么可以在目标代理定义替换字符序列，以便传输不会失败。指定代理属性 **textReplacementCharacterSequence** 来定义替换字符序列。有关更多信息，请参阅第 592 页的表 50。

2. 源文件编码与源代理的缺省编码不匹配。在这种情况下，使用缺省设置执行文本传输会损坏字符数据。如果要传输的源文件与源代理的编码不同，请执行以下某个步骤：
 - a. 在传输定义文件中指定文件编码。有关更多信息，请参阅第 222 页的『使用传输定义文件』。
 - b. 将 **-sce** 参数和 **fteCreateTransfer** 命令结合使用来指定文件编码。有关更多信息，请参阅主题第 497 页的『fteCreateTransfer（创建新的文件传输）』。
 - c. 作为 Ant 移动或复制任务的一部分指定文件编码。有关更多信息，请参阅第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』。

要检查您是否为传输选择了正确的源文件编码，请执行以下步骤：

1. 将目标文件编码设置为 UTF-8。
 2. 以文本方式传输文件。
 3. 使用 UTF-8 文件查看器来查看文件的内容。如果文件中所有字符都正确显示，那么源文件编码正确。
- 如果您从 **fteCreateAgent** 命令中看到以下输出：

```
BFGMQ1007I: The coordination queue manager cannot be contacted or has refused a
connection attempt.
The WebSphere MQ reason code was 2058. The agent's presence will not be published.
```

它指示协调队列管理器无法访问，并提供 IBM MQ 原因码以说明原因。此参考消息可能指示协调队列管理器当前不可用或者您未正确定义配置。

- 如果您在使用用户出口例程，并且在调用用户出口时或者在已经调用出口之后发生故障（例如，产品故障或断电），那么可能多次调用用户出口。
- 如果您的系统上具有一个包含队列管理器的代理，并且其 IP 地址由 DHCP 分配（而不是静态 IP 地址），并且代理使用客户机 TCP/IP 连接来连接到该系统，那么您必须在设置了以下系统环境变量的情况下启动代理：
 - 在 Windows 上：

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

- 在 UNIX 上：

```
export BFG_JVM_PROPERTIES="-Dsun.net.inetaddr.ttl=<value>"
```

其中，<value> 是指清除 JVM 的高速缓存 DNS 值所经历的时间间隔（单位为秒）。如果队列管理器系统的 IP 地址由于任何原因而重新分配（例如，由于网络中断、IP 租约过期或系统重新引导），那么代理会报告其与队列管理器丢失连接。清除 JVM DNS 高速缓存后，代理可以成功地重新连接。如果未设置环境变量，那么在不重新启动 JVM 的情况下，代理无法在此场景下重新连接。此行为的原因是 JVM 在内部对主机名的 IP 地址进行了高速缓存，并且缺省情况下不会刷新这些 IP 地址。

- 如果运行 **fteStartAgent** 命令并且看到以下错误消息，那么您的环境可能具有与 IBM MQ Managed File Transfer 冲突的附加库路径：

```
BFGCL0001E: An internal error has occurred. The exception was: 'CC=2;RC=2495;AMQ8568:
The native JNI library 'mqjbnd' was not found. [3=mqjbnd]
```

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，那么当代理程序使用 32 位版本的 Java 运行时（当前对于大多数平台是如此），将发生此错误。

要解决此问题，请设置 IBM MQ Managed File Transfer 代理属性 `javaLibraryPath` 以引用库的正确位置。例如，对于 AIX 上的 `mqjbnd`，请设置为：`/usr/mqm/java/lib`。对于 Linux 上的 `mqjbnd`，请设置为：`/opt/mqm/java/lib`

- 如果您通过在代理属性文件中指定 `authorityChecking=true` 而启用了用户权限检查，并且所有权限检查都失败（即使用户对相关权限队列具有必需权限）：
 - 请确保运行的代理的用户在代理队列管理器上具有 `ALT_USER` 访问控制。
- 如果您通过在代理属性文件中指定 `authorityChecking=true` 而启用了用户权限检查，并且 IBM MQ 错误消息写入到代理 `output0.log` 文件中，请执行以下某项操作：
 - 请忽略这些消息，代理不会受影响。
 - 为运行代理的用户授予 `SYSTEM.FTE.AUTH*` 队列（属性此代理）上的 `GET` 权限。
- 如果您已编辑了代理属性文件，并且代理尚未获取这些文件：
 - 重新启动代理，以确保代理能读取新属性。

z/OS

- 如果您在 z/OS 上使用代理来传输到 PDS 或 PDSE 数据集并且发生异常终止，那么您的系统可能具有有限的磁盘空间。异常终止可能具有系统完成代码 B14 和返回码 0C，这表示没有任何剩余空间。
如果是传输到连续数据集，那么传输会失败并表明空间不足这一情况，但是代理会仍然运行。
- 如果您在 z/OS 上使用代理，并且 `WMQFTEP` 任务生成了一些 Java 核心转储之后变为无响应，那么请应用 OMVS 系统服务 APAR OA43472。
- 如果在 z/OS 上运行配置或管理脚本时看到以下输出：

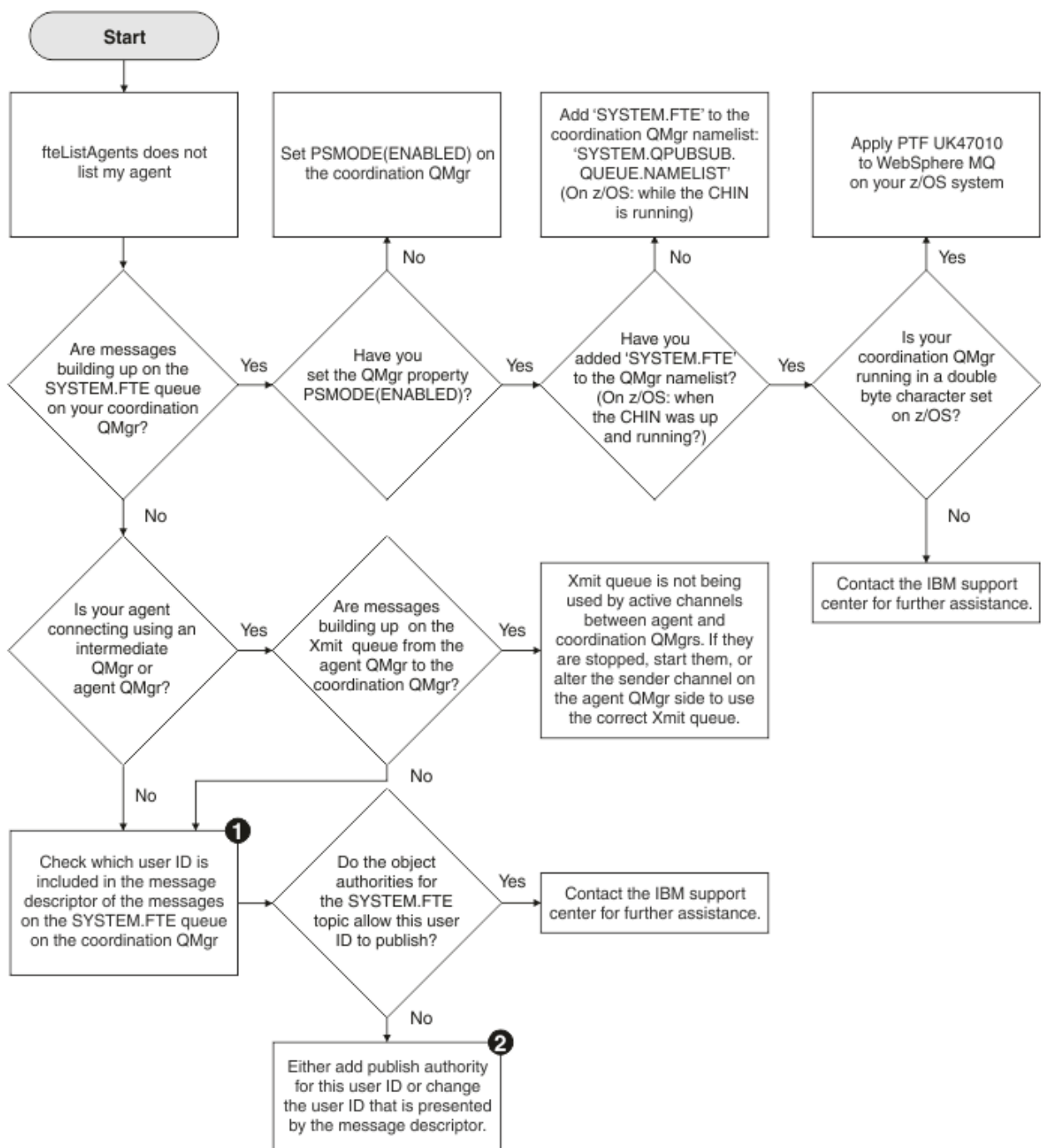
```
FSUM7332 syntax error: got (, expecting Newline
```

该输出指示在运行配置或管理脚本的环境中未设置环境变量 `_BPXK_AUTOCVT=ON`。有关该环境变量以及如何设置的更多信息，请参阅第 137 页的『IBM MQ Managed File Transfer for z/OS 的环境变量』。

如果 `fteListAgents` 命令未列出您的代理，该怎么办

如果 `fteListAgents` 命令未列出您的代理、您的代理未显示在 IBM MQ Explorer 中，或者您的文件传输未显示在 IBM MQ Explorer 的“传输日志”中，那么可以执行若干问题确定步骤来调查原因。

使用以下流程图可帮助您诊断问题并决定接下来采取的操作：



流程图要点：

1. 有关如何检查提供的用户标识的更多信息，请参阅第 391 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 队列所需权限的更多信息，请参阅第 442 页的『用于发布日志和状态消息的权限』。

如果代理进程消失但未记录任何诊断信息，该怎么办？

在 UNIX 平台上，如果代理进程已消失，但代理日志文件不包含任何解释，这可能由于启动此代理的方式所致。

您可通过如下方式查看代理诊断信息：

- 检查代理的日志文件是否声明已停止此代理。
- 检查代理锁文件 `agent.lck` 是否仍存在。

例如，如果从 shell 脚本启动代理，那么当该脚本完成时（包括代理进程），将移除与该脚本关联的所有子进程。要在调用代理的脚本的持续时间过后使此代理保持运行，请完成以下步骤：

1. 在 **fteStartAgent** 命令前面添加 **nohup** 命令，使 **fteStartAgent** 进程（及所有子进程）从脚本中分离。

将来，当脚本终止时，代理会继续运行。

当 **fteListAgents** 命令显示代理状态为 UNREACHABLE 时应如何处理

您的代理正在运行，并且可以成功响应 **ftePingAgent** 命令，文件正常传输，但是该代理被 **fteListAgents** 命令列为 UNREACHABLE。

发生此问题的原因

代理会定期将其状态发布到协调队列管理器。代理发布状态的频率由以下两个代理属性控制：

agentStatusPublishRateLimit

由于文件传输状态更改，代理重新发布其状态的最大速率（秒）。

agentStatusPublishRateMin

代理发布其状态的最小速率（秒）。该值必须大于或等于 **agentStatusPublishRateLimit** 属性的值。

使用缺省设置时，如果代理系统和协调队列管理器系统的时间差异超过 303 秒，那么这两个系统之间时钟不同步会导致此问题。如果代理状态消息在 **agentStatusPublishRateMin** 值与 **agentStatusJitterTolerance** 值之和（秒数）前发送，那么这些消息将被视为旧消息。带有旧状态消息的代理将由 **fteListAgents** 命令报告为 UNREACHABLE。

缺省情况下，**agentStatusJitterTolerance** 属性的值为 3000 毫秒，**agentStatusPublishRateMin** 属性的值为 300 秒。如果机器之间的时差加上有效发布率大于 **agentStatusPublishRateMin** 与 **agentStatusJitterTolerance** 之和，那么该时差会导致 UNREACHABLE 代理状态。

解决问题

可以通过以下任一方法解决此问题：

- 更正代理主机与托管协调队列管理器的机器间的时间设置差异，以使其同步。
- 增大 **agentStatusJitterTolerance** 属性的值以消除时间差异造成的影响。如果运行的是 **fteListAgents** 命令，那么 **agentStatusJitterTolerance** 的值由 MQMFTconfig 目录中的 `coordination.properties` 配置文件确定。因此，请设置 MQMFT 安装（运行 **fteListAgents** 命令的安装）的 `coordination.properties` 文件内的属性。

相关参考

第 591 页的『[agent.properties 文件](#)』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

当 **ftePingAgent** 超时并报告 BFGCL0214I 消息时应执行的操作

ftePingAgent 是随 IBM MQ Managed File Transfer 组件提供的一个有用的命令行实用程序，可用于检查代理当前是否正在运行。

命令的工作方式

在 Managed File Transfer 配置或拓扑中，可以有多个队列管理器，它们将担任一个或多个角色。请参阅 [MFT 拓扑概述](#) 以获取更多信息。

:

协调队列管理器

这是用于存储配置的所有代理状态、传输状态和传输审计信息的中央存储位置

命令队列管理器

用于将包含命令的消息路由到配置中的各个代理。

代理队列管理器

用于托管代理所需的所有内部系统队列。

IBM MQ 发送方和接收方通道用于：

- 将协调队列管理器连接到所有代理队列管理器。
- 将命令队列管理器连接到所有代理队列管理器。
- 将每个代理连接到配置中的所有其他代理。

运行 **ftePingAgent** 命令时，它将执行以下步骤。该命令：

- 连接到命令队列管理器。
- 在命令队列管理器上创建一个临时应答队列。

缺省情况下，该临时队列的名称以前缀 WMQFTE 开头。但是，您可以通过在配置的 [MFT command.properties](#) 文件中设置 **dynamicQueuePrefix** 属性来更改此属性。

- 通过命令队列管理器，将 Ping MFT 代理请求消息发送到代理队列管理器上的队列 `SYSTEM.FTE.COMMAND.agent_name`。该请求消息包含临时应答队列的名称。
- 等待 PingAgent 应答到达临时应答队列。

当代理启动时，它会连接到其代理队列管理器，然后创建一个内部 CommandHandler 线程。该线程旨在从 `SYSTEM.FTE.COMMAND.agent_name` 队列中选取消息，并相应地处理这些消息。

如果该线程接收到包含 PingAgent 请求的消息，那么该线程将构建 PingAgent 响应，并将此响应发送回命令队列管理器上的临时应答队列；此应答消息将通过代理队列管理器。

以下两个图显示了这一流程：

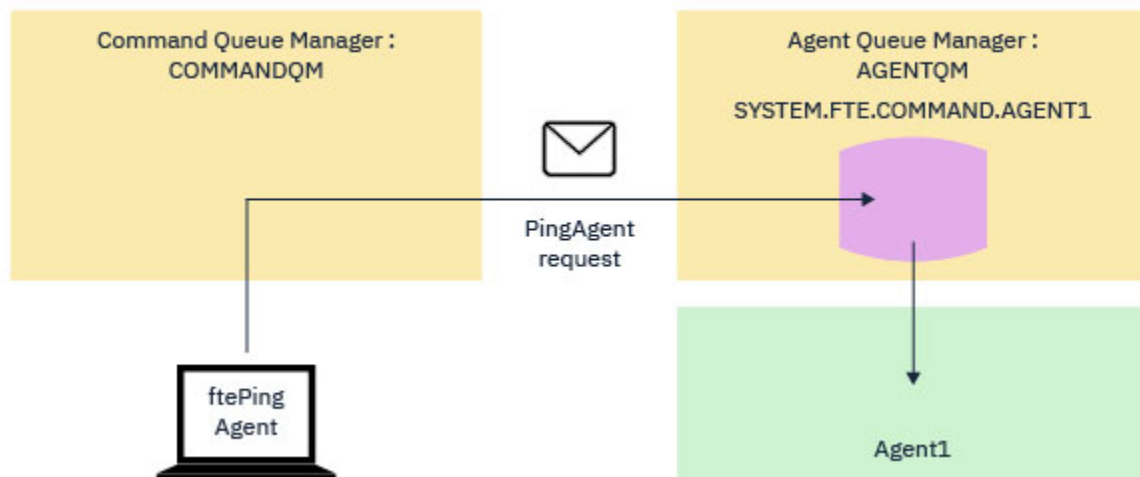


图 13: pingAgent 请求通过命令队列管理器进入代理队列管理器上的 `SYSTEM.FTE.COMMAND.agent_name` 队列

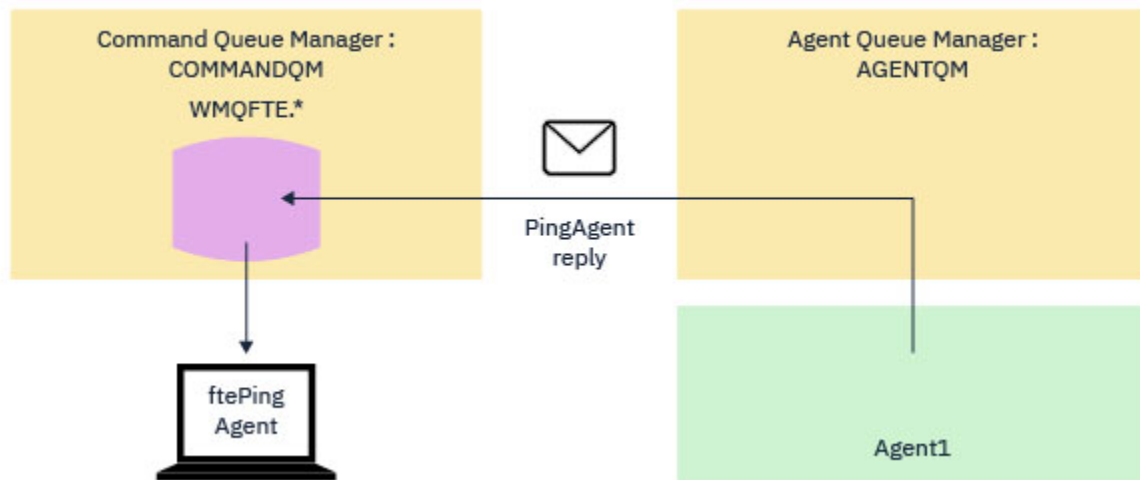


图 14: pingAgent 应答通过代理队列管理器返回到命令队列管理器。

当命令超时时应执行的操作

如果命令超时，并且您看到以下消息：

BFGCL0214I: *number* 秒后，代理 *agent_name* 未响应 ping。

这不一定表示该代理未在运行。

调查以下内容以了解发生超时的原因：

- 缺省情况下，**ftePingAgent** 命令等待 5 秒以获取响应消息。如果代理繁忙，那么内部 CommandHandler 线程可能耗用了 5 秒以上的时间来选取 PingAgent 消息并对其进行处理。

要查看情况是否如此，请尝试在更长的等待时间后重新运行该命令。通过指定 **-w** 参数来执行此操作。例如，要按 60 秒的等待时间间隔来运行 **ftePingAgent** 命令，请发出以下命令：

```
ftePingAgent -w 60 AGENT1
```

- 如果命令仍然超时，请检查命令队列管理器与代理队列管理器之间的发送方和接收方通道。如果这些通道发生故障，那么包含 PingAgent 请求的消息将停滞在命令队列管理器上的传输队列上。

如果这些通道已启动并正在运行，并且增加命令使用的等待时间间隔毫无帮助，那么需要通过配置来跟踪消息以了解：

- 消息是否到达 SYSTEM.FTE.COMMAND.*agent_name* 队列。
- 代理是否曾经从此队列中选取了消息。

如果您在执行此操作时需要帮助，请执行以下步骤：

- 在命令和代理队列管理器上启用队列管理器跟踪。
- 使用跟踪规范 `com.ibm.wmqfte=all` 在代理上启用跟踪。
- 运行 **ftePingAgent** 命令并指定额外的命令行选项：

```
-trace com.ibm.wmqfte=all
-tracePath directory_name
```

这将跟踪命令，并在 **-tracePath** 自变量指定的目录中生成跟踪文件。

当命令超时时，停止所有跟踪并将其提供给 IBM 支持人员进行分析。

当代理或记录器配置不安全时应如何处理

如果某个 IBM MQ Managed File Transfer 进程检测到某个配置文件包含敏感信息，是密钥库或信任密钥库文件，并且具有系统范围的读、写或删除许可权，那么如果是在启动时检测到该情况，该进程将无法启动。如

果该情况不是在启动时而是在运行时检测到的，那么 IBM MQ Managed File Transfer 会生成警告消息并忽略该配置文件的内容。这与协议网桥和 Connect:Direct 网桥功能相关，如果在代理运行期间某个配置发生更改，这些功能会重新装入该配置。

请完成以下检查以确定问题原因：

1. 从提供的错误消息中确定已报告为不安全的配置文件。
2. 确保该文件的访问许可权与所需的需求相匹配。有关更多信息，请参阅 [第 442 页的『包含敏感信息的配置文件的许可权』](#)。
3. 重新启动代理或记录器。或者，如果是协议网桥或 Connect:Direct 凭证文件，请等待下次重新装入。

示例

在此示例错误消息中，数据库记录器无法启动：

```
BFGDB0066E: The logger encountered a problem accessing its credentials file and will stop.  
Reported error: BFGNV0145E: The 'Everyone' group has access to the file 'C:\mqmftcredentials.xml'.
```

在此示例错误消息中，协议网桥代理无法启动：

```
BFGI00383E: The security permissions defined for credentials file 'C:\ProtocolBridgeCredentials.xml' do  
not meet the  
minimum requirements for a file of this type.  
Reported problem: BFGNV0145E: The 'Everyone' group has access to the file  
C:\ProtocolBridgeCredentials.xml'.
```

相关参考

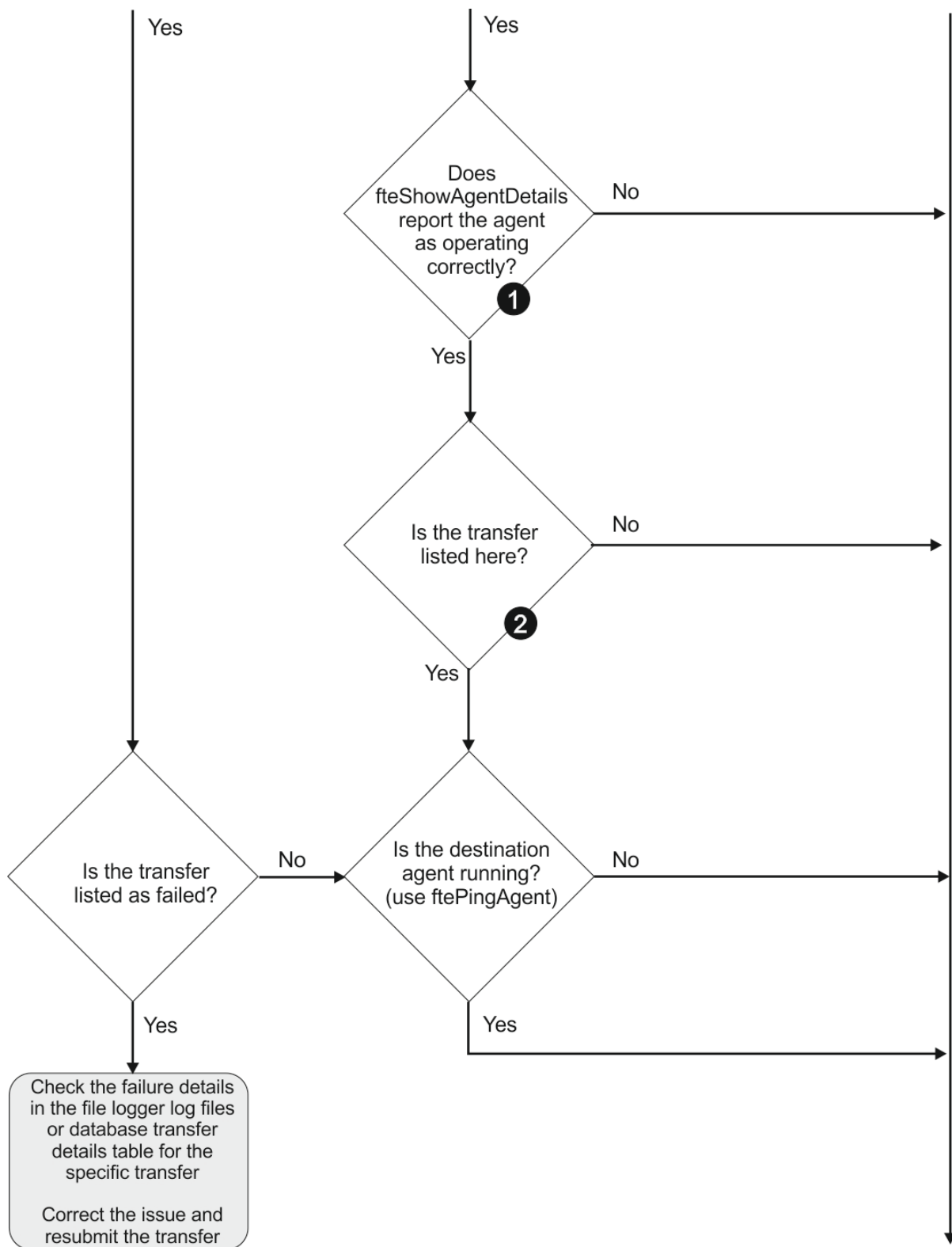
[第 442 页的『包含敏感信息的配置文件的许可权』](#)

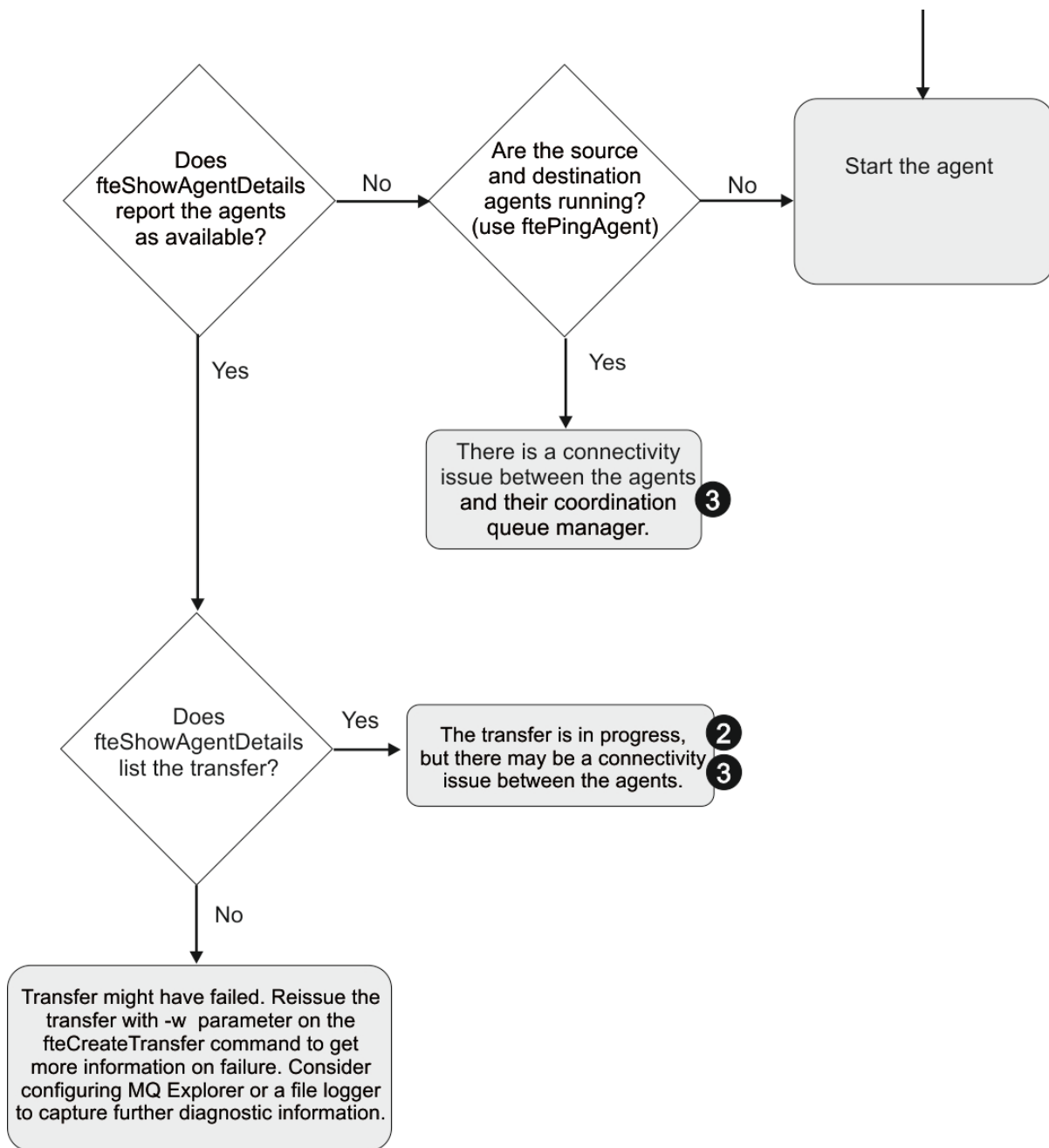
用于存储敏感配置信息的任何文件（指的是从 IBM MQ 配置树中引用的任何文件）不得具有系统范围的读、写或（适用时）删除许可权。这些限制也适用于信任密钥库和密钥库文件。

当传输没有完成时应如何处理

如果传输没有完成，您可以执行几个问题确定步骤来调查原因。

使用以下流程图可帮助您诊断问题并决定接下来采取的操作：





流程图要点：

1. 检查代理的 `output0.log` 中是否有错误。如果代理程序报告其已成功启动，但 WebSphere MQ Explorer 或 **fteShowAgentDetails** 都未报告代理程序正在运行，请检查代理程序队列管理器与协调队列管理器之间的连接。可能是因为队列管理器间通道不可用。
2. 如果源代理将传输标识列示为 `In progress` 传输，但目标代理未列示，那么源队列管理器与目标队列管理器之间可能存在连接问题。在 `command.properties` 文件中，从目标代理机器向使用目标代理队列管理器作为命令队列管理器的源代理使用 **ftePingAgent** 命令。您也可以按相反方向运行此命令，即从源到目标。
3. 如果源和目标代理都将传输标识列示为 `In progress`，那么这表明自传输启动以来，源和目标队列管理器之间存在连接问题。在 `command.properties` 文件中，从目标代理机器向使用目标代理队列管理器作为命令队列管理器的源代理使用 **ftePingAgent** 命令。您也可以按相反方向运行此命令，即从源到目标。
4. 如果已运行过此循环，请检查以下任一声明是否与您的情况相关：

- 源代理和目标代理都报告为 **Running**，但未列出任何传输。传输请求未到达代理命令队列，或者代理虽然报告为 **Running**，但不再监视命令队列。请检查源代理的 `output0.log` 中是否有错误。从发送传输的同一机器向源代理使用 **ftePingAgent** 命令，以验证命令队列管理器和代理队列管理器之间的连接，以及代理是否正在为命令队列提供服务。
- 源和目标代理都报告为 **Running**，并且传输列示为 **In progress, recovering**。在 `command.properties` 文件中，从目标代理机器向使用目标代理队列管理器作为命令队列管理器的源代理使用 **ftePingAgent** 命令。您也可以按相反方向运行此命令，即从源到目标。

在您认为传输卡住时应该怎么办

在高负荷的系统上或者在源和目标代理之间发生网络问题时，传输可能会偶尔显示为卡住，进入排队或恢复状态。导致此情况的因素有很多。

请完成以下检查以确定问题原因：

1. 使用 **ftePingAgent** 命令，或者在 WebSphere MQ Explorer **代理程序** 面板中右键单击代理程序名称并选择 **Ping**，以检查源代理程序和目标代理程序是否处于活动状态并响应新请求。查看代理日志以查看当前是否存在网络连接问题。
2. 检查目标代理是否以最高能力运行。可能存在大量源代理都在请求将文件传输到相同的目标代理。使用带有 **-v** (详细) 参数的 **fteShowAgentDetails** 命令，或者在 WebSphere 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** 以查看代理程序的当前传输活动。如果正在运行的目标传输的数量等于或接近目标传输的代理最大数量，那么这可能是某些源代理传输显示为卡住的原因。
3. 如果联系协议文件服务器时发生问题，那么进出协议网桥代理的传输将进入恢复状态。查看代理日志以查看当前是否存在连接问题。
4. 代理将按优先级顺序处理传输。因此，在装入系统中，低优先级传输可能仍保持排队状态一段时间，而将装入具有较高优先级传输的代理。最后，如果该传输已排了一段时间队，那么将启动低优先级传输，即使存在新的较高优先级的传输。

在已安排传输未运行或被延迟时应执行什么操作

如果您具有到期未运行或被延迟的已调度传输，那么原因可能是代理正在处理其命令队列中的命令。由于代理很繁忙，因此未检查已调度传输，从而未运行已调度传输。

要解决该问题，请使用以下步骤之一：

- 配置 `agent.properties` 文件中的 `maxSchedulerRunDelay` property 属性，以设置代理等待检查已调度传输的最大时间间隔（分钟）。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，请参阅第 591 页的『`agent.properties` 文件』。
- 或者，请使用资源监视器代替已调度传输。资源监视器的工作方式与已调度传输不同，不受繁忙代理的影响。例如，如果要获取目标系统上的最新文件，资源监视器将减少网络流量。这是因为只有当新版本可用时才会传输文件，而不是自动传输文件。但是，在协议网桥代理或 **Connect:Direct** 网桥代理上不支持资源监视。

有关更多信息，请参阅第 229 页的『资源监控』。

当协议网桥代理报告找不到某个文件时应如何处理

当协议网桥代理报告协议网桥所连接的 SFTP 或 FTP 服务器返回 **File not found** 错误消息时，此消息可能表示发生了许多不同的错误情况之一。

以下可能的场景可能导致 SFTP 或 FTP 服务器返回 **File not found** 错误。

- 文件不存在。检查托管 SFTP 或 FTP 服务器的系统上是否存在您尝试传输的文件。
- 文件路径不存在。检查托管 SFTP 或 FTP 服务器的系统上是否存在该文件路径。检查您是否已在传输请求中正确输入文件路径。如有必要，请纠正文件路径，然后重新提交传输请求。
- 文件被另一个应用程序锁定。检查文件是否被另一个应用程序锁定。等待直至文件不再被锁定，然后重新提交传输请求。

- 文件许可权不允许读取文件。检查文件是否具有正确的文件许可权。如有必要，请更改文件许可权，然后重新提交传输请求。
- SFTP 或 FTP 服务器使用虚拟的根路径。如果在传输请求中指定了相对文件路径，那么该协议网桥代理程序会尝试根据用于登录协议服务器的主目录将相对路径转换为绝对文件路径。IBM MQ Managed File Transfer 协议网桥代理程序只能支持允许通过绝对文件路径访问文件的 SFTP 或 FTP 服务器。该协议网桥代理程序不支持这些允许仅根据当前目录访问文件的协议服务器。

相关概念

第 275 页的『协议网桥』

通过协议网桥，IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件，而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

当目录资源监视器未触发文件时应如何处理

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。

未触发文件的可能原因

1. 目录资源监视器发现一个与触发器规范匹配的文件，但生成的传输请求无效并且代理无法处理请求。可能包含以下原因：
 - 目标代理无效
 - 缺少目标代理
 - 传输由程序调用取消

在所有这些示例中，即使传输失败，目录资源监视器仍将触发文件标记为已处理并忽略文件。

2. 文件超出了资源监视器触发器规范的范围。可能包含以下原因：
 - 触发器模式不正确
 - 正在监控不正确的目录
 - 文件许可权不足
 - 无法连接至远程文件系统

为什么文件会触发第二次传输

触发器文件可以出于以下原因而生成 IBM MQ 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。

如何调查未触发目录资源监视器的原因

在代理正在运行和通过触发器规范配置了目录资源监视器的情况下，使用以下命令：

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.monitor=all agent_name
```

示例

在该示例中，为 AGENT1 生成了跟踪文件，并显示具有触发器规范 *.packet 的监视器的单次轮询。文件 tomato.tin 与触发器规范不匹配。文件 rice.packet 与触发器规范匹配但自上次轮询未改变，因此不会触发传输。文件 biscuit.packet 触发了传输，因为文件正常结束，且自上次轮询以来为新建或已更改。

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.monitor=all AGENT1
```

```
08:36:53.908.00 0004 ... dftStartPoll data [@MON0001SP0001:HOTEL]
08:36:54.178.00 0004 ... dftItem data [@MON0001FL0002:Ignored:/home/mondir/shop/tomato.tin::Pattern mis-match]
08:36:54.335.02 0004 ... dftItem data [@MON0001FL0003:Ignored:/home/mondir/shop/rice.packet:11 secs::Unchanged since
last trigger]
08:36:54.487.00 0004 ... dftItem data [@MON0001FL0004:Triggerred:/home/mondir/shop/biscuit.packet:::]
08:36:54.488.00 0004 ... dftGeneral data [@MON0001GN0005:task items matches = 1]
08:36:54.553.00 0004 ... dftTask data [@MON0001TK0006:01:[AGENTNAME=PETER,FILEPATH=/home/mondir/shop/
biscuit.packet, ... ]]
08:36:55.151.02 0004 ... dftTransferRequest data [@MON0001TK0007: ...]
08:36:55.632.00 0004 ... dftEndPoll data [@MON0001EP0008:HOTEL]
```

每行都包含如下格式的标识：

@MONmmmmAAssss

- mmmm 值为分配给此跟踪的监视器的数字。
- AA 是所用操作的代码。
- ssss 是以 0001 开始的语句号。

AA 代码可以采取以下某个值：

- SP - 开始轮询
- FL - 文件
- GN - 一般
- EX - 异常
- TK - 任务
- FD - FFDC
- EP - 结束轮询

如果跟踪文件中存在相当大数量的 Ignored 文件，那么可以通过使用以下命令消除这些条目：

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.monitor=moderate AGENT1
```

相关参考

第 374 页的『[fteSetAgentTraceLevel（设置 IBM MQ Managed File Transfer 代理跟踪级别）](#)』
使用 **fteSetAgentTraceLevel** 命令可动态修改代理的当前跟踪级别。

如果队列资源监视器启动的传输所创建的目标文件包含错误数据，那么该执行哪些操作？

您可以创建资源监视器来监视队列，并将队列上的消息或一组消息传输到文件。可以通过使用消息上的 MQMD 消息描述符或组中的第一个消息来指定文件名。如果消息到文件的传输失败，并且消息或组留在队列上，那么下次触发监视器时，可能导致创建的文件中包含错误数据。

发生此问题的原因

1. 消息到文件的传输失败，并且消息或组留在队列上。
2. 新的消息或组到达队列。
3. 新的消息或组触发资源监视器。
4. 资源监视器创建一个新传输，此传输使用新消息或组上的 MQMD 消息描述符以及队列上的第一个消息或组中的数据。

5. 创建的文件中包含错误数据。

避免此问题

要避免遇到此问题，必须使用 **fteCreateTransfer** 命令手动创建传输定义文件，并编辑文件的 `<queue>` 元素，以包含属性 `groupId="{GROUPLD}"`。然后使用 **fteCreateMonitor** 命令来提交传输定义文件。

示例

在此示例中：源代理名为 AGENT_MON，同时也在监视代理；目标代理名为 AGENT_DEST；目标文件名为 `/out/files/{WMQFTEFileName}`。此示例要求消息已设置了 MQMD 消息描述符 WMQFTEFileName。正在监视的队列为 LIVE_QUEUE。

1. 通过运行以下命令来创建传输定义文件：

```
fteCreateTransfer -sa AGENT_MON -da AGENT_DEST -df "/out/files/{WMQFTEFileName}"  
-de error -gt /tmp/TransferDefinition1.xml -sqgi -sq LIVE_QUEUE
```

这将生成传输定义文件 `/tmp/TransferDefinition1.xml`。

2. 编辑 `<queue>` 元素，以包含属性 `groupId="{GROUPLD}"`。将行

```
<queue useGroups="true">LIVE_QUEUE</queue>
```

到

```
<queue useGroups="true" groupId="{GROUPLD}">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。

如果目标队列是集群队列或集群队列的别名，那么该执行什么操作？

使用 IBM MQ Managed File Transfer 将文件传输到队列中时，如果使用的目标队列是集群队列或集群队列的别名，那么您将收到原因码 2085 或 2082。从 V7.5.0.4 和更高版本开始，将属性 `enableClusterQueueInputOutput` 设置为 `true` 可解决此问题。

发生此问题的原因

当 `-dq` 上无显式队列管理器名称时，会将目标代理的队列管理器名称附加到 `-dq` 参数的队列名称。由于在连接到没有该本地集群队列的集群 MQ `queueManager` 时，无法对 MQOPEN 调用指定 `queueManager` 对象，因此会出现原因码 2085 或 2082。

避免此问题

1. 在队列管理器上创建集群队列。
2. 设置指向集群队列的远程队列定义。

示例

本示例使用远程队列定义。

配置：

- 源代理：SAGENT

- 源代理队列管理器: *SQM*
- 目标代理: *DAGENT*
- 目标代理队列管理器: *DQM*
- 传输的目标队列是队列管理器 *SQM* 上的 *CQ6*

要针对 *SQM* 中的集群队列 *CQ6* 定义 *DQM* 上的远程队列定义 *Q6_SQM* (假设已在 *SQM* 中定义集群队列 *CQ6*)，请在 *DQM* 队列管理器上发出 *MQSC* 命令：

```
define qremote(Q6_SQM) rname(CQ6) rqmname(SQM) xmitq(SQM)
```

注: *rname* 指向集群队列。

您现在可以传输至队列。例如：

```
fteCreateTransfer -sa SAGENT -sm SQM -da DAGENT -dm DQM -dq Q6_SQM /tmp/single_record.txt
```

如果消息在 **SYSTEM.MANAGED.DURABLE** 队列上聚集或者填充文件系统，那么该执行哪些操作？

如果 IBM MQ Explorer 插件在协调队列管理器上使用持久预订，那么消息可能会在 **SYSTEM.MANAGED.DURABLE** 队列上聚集。如果您具有大容量 IBM MQ Managed File Transfer 网络，请不要频繁使用 IBM MQ Explorer 插件，否则此消息数据可能会填满本地文件系统。

要除去 **SYSTEM.MANAGED.DURABLE** 队列上消息的聚集，可执行以下其中一种操作：

- 启动使用持久预订的 IBM MQ Explorer。IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件使用队列中的消息。
- 手动从队列删除消息。

要避免发生此情况，可以指定 IBM MQ Explorer 插件使用对协调队列管理器的非持久预订。在 IBM MQ Explorer 中执行以下步骤：

1. 选择窗口 > 首选项 > **IBM MQ Explorer** > 受管文件传输
2. 在传输日志订阅类型列表中，选择 **NON_DURABLE**。

发布前检查消息

由于代理程序可以连接到 WebSphere MQ V 6 队列管理器，因此代理程序不会使用 WebSphere MQ V 7 中引入的直接发布方法。事实上，代理将普通消息发送到包含 **MQRFH** 头的协调队列管理器。**MQRFH** 头请求发布消息的有效内容。这些消息会被发送到协调队列管理器上的 **SYSTEM.FTE** 队列，通常会立即从该队列发布这些消息。如果错误条件阻止此发布，您可以在尝试发布前检查队列上的消息以帮助进行诊断。可通过完成以下步骤来执行此操作：

1. 在协调队列管理器中禁用发布/预订引擎。

您可以使用 IBM MQ Explorer 或使用 *MQSC* 命令来完成该步骤。请注意，如果协调队列管理器同样用于其他目的，那么这将暂时停止队列管理器上的所有发布/预订活动，包括与 IBM MQ 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 IBM MQ 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。

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

IBM MQ Explorer:

- 在“导航器”视图中，展开协调队列管理器并单击**队列**。在“内容”视图中，右键单击 SYSTEM.FTE 队列并选择**浏览消息**。将打开**消息浏览器**窗口并显示将发布的消息。
- 用户标识**列显示消息描述符中所含的用户标识。发布失败的常见原因是该用户标识对于 SYSTEM.FTE 主题没有发布权限。
- 您可以通过右键单击消息并选择**属性**，查找有关每条消息（包括将发布的 XML）的更多信息。

没有用于检查消息内容的 MQSC 命令。如果您没有 IBM MQ Explorer，那么必须使用可用于浏览队列并显示所发现消息的所有方面的其他程序。您可以使用 **amqsbcg** 样本程序（如果已安装），如以下主题中所述：浏览队列。UserIdentifier 行显示用户标识。或者，您可以使用 **dmpmqmsg**；可以在如下行中找到消息的用户标识：

```
A RTM MQ24
A USR JOHND0E
A ACC 1A0FD4D8F2F4C3C8C9D5F1F9C6F7C1C3F3F00019F7AC3000000000000000000
```

该示例中的第二行是该消息的消息描述符用户标识。

4. 重新启用协调队列管理器的发布/预订引擎。

您可以使用 IBM MQ Explorer 或使用 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)
```

使用 IBM MQ Managed File Transfer 的提示与技巧

以下建议可帮助您最有效地使用 IBM MQ Managed File Transfer:

- 如果您更改 agent.properties 文件，请停止并重新启动代理以应用更改。
- 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 IBM MQ 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 UNIX 类型的平台：请运行命令：**ulimit -m 1048576**（或者约 1GB）。该最大常驻集合大小足以允许最多 25 个并行传输（25 个并行传输是一个代理的缺省最大传输数）。
 - 对于所有平台：按如下所示设置 **BFG_JVM_PROPERTIES** 环境变量：**BFG_JVM_PROPERTIES="-Xmx1024M"**

如果要允许并行传输数大于缺省最大值 25，请针对 **ulimit** 和 **BFG_JVM_PROPERTIES** 使用大于建议值的大小。

注：对于 Connect:Direct 网桥代理，最大并行传输数量的缺省值为 5。

- 使用 IBM MQ Managed File Transfer 以文本方式在不同平台间传输文件时，源平台的缺省文件编码可能不受目标平台的支持。这会导致传输失败，并出现以下错误：

```
BFGI00058E: The transfer source encoding xxx is illegal or for an unsupported character set.
```

您可通过使用环境变量将源编码设置为目标平台支持的编码来解决此问题。按如下所示在源系统上设置 **BFG_JVM_PROPERTIES** 系统环境变量：**BFG_JVM_PROPERTIES="-Dfile.encoding=xxx"**，其中 **xxx** 是目标平台支持的编码。例如，如果您要以文本方式将文件从 Sun Solaris 平台传输至不同的平台，并且源语言环境设置为“ja”，请按如下所示设置 **BFG_JVM_PROPERTIES**：**BFG_JVM_PROPERTIES="-Dfile.encoding=EUC-JP"**。如果源语言环境设置为“ja_JP.PCK”，请按如下所示设置 **BFG_JVM_PROPERTIES**：**BFG_JVM_PROPERTIES="-Dfile.encoding=Shift_JIS"**。

您还可在启动新传输时使用 **-sce** 参数来解决个别传输的该错误。有关更多信息，请参阅主题 [第 497 页的『fteCreateTransfer（创建新的文件传输）』](#)。

相关参考

[第 633 页的『Java 系统属性』](#)

必须将许多 IBM MQ Managed File Transfer 命令和代理程序属性定义为 Java 系统属性，因为它们定义了无法使用命令或代理程序属性机制的早期功能的配置。

传输 IBM i 保存文件时可能发生的错误

如果您使用 IBM MQ Managed File Transfer 多次传输同一个 IBM i 保存文件，那么传输可能失败。

IBM MQ Managed File Transfer 可能产生以下错误中的一个或全部：

- ```
BFGII0003E: Unable to open file "/qsys.lib/library.lib/SAVF.FILE" for reading
```
- ```
BFGII0082E: A file open for read failed due to a Java IOException with message text "Sharing violation occurred"
```

如果向 MQMFT 代理发出多个请求来传输同一个 IBM i 保存文件，那么可能会发生这些错误。如果希望同时多次传输同一个保存文件，那么必须使用多个源代理。请对每个并发传输使用不同源代理。

要使单个源代理来多次传输同一个保存文件，您必须等到前一个传输请求完成，然后再提交每个新传输请求。

相关任务

[第 140 页的『在 IBM i 系统上安装 IBM MQ Managed File Transfer 后进行配置』](#)

要在安装后开始使用 IBM MQ Managed File Transfer，必须为协调队列管理器代理完成某些配置。

相关参考

[第 716 页的『在 IBM i 系统上传输文件』](#)

如果您使用文本方式的 IBM MQ Managed File Transfer 在 IBM i 系统上传输文件，并且希望转换文件中的数据，请研究本主题中的信息。

第 719 页的『传输位于 IBM i 系统上 QSYS.LIB 文件系统上的保存文件』

IBM MQ Managed File Transfer 支持在两个 IBM i 系统之间传输 QSYS.LIB 文件系统上的保存文件。在请求对保存文件进行文件传输时，请考虑以下信息。

用于设置与消息大小关联的 IBM MQ 属性和 IBM MQ Managed File Transfer 属性的指南

在读取或写入各种大小的消息时，您可以更改 IBM MQ 属性和 IBM MQ Managed File Transfer 属性以影响 IBM MQ Managed File Transfer 的行为。

如果从源队列读取或写入目标队列的消息大小超过 1048576 字节 (1 MB)，那么必须将 IBM MQ Managed File Transfer 代理属性 **maxInputOutputMessageLength** 的值增大到大于或等于要读取或写入的最大消息大小的值。

如果源队列上的消息大于 1048576 字节，您必须在源代理上设置 **maxInputOutputMessageLength** 属性。如果目标队列上的消息大于 1048576 字节，您必须在目标代理上设置 **maxInputOutputMessageLength** 属性。有关 **maxInputOutputMessageLength** 属性的更多信息，请参阅 高级代理程序属性。

- 如果代理写入或读取的队列对于代理队列管理器来说是本地队列，那么您可能必须更改 IBM MQ 队列管理器、队列和通道 **MAXMSGL** 属性。

确保源或目标队列的最大消息大小的值大于或等于 **maxInputOutputMessageLength** 代理属性的值。

确保以下每个 IBM MQ 属性的值（字节）：

- 代理队列管理器的最大消息大小
- The maximum message size of the SYSTEM.FTE.STATE.<代理程序名称> queue
- 客户机通道的最大消息大小（如果您的代理以客户机方式连接到队列管理器）

大于或等于以下计算的结果：

对于“文件到消息”传输（支持最高 100 MB 的文件大小）：
maxInputOutputMessageLength 的值

对于消息到文件的传输：

$3 * (\text{maxInputOutputMessageLength}) + 1048576$ 的值

（该计算源于如下事实：三个检查点可以存储在状态消息中，并且每个检查点可能都必须缓冲，直至数据消息量的最大大小。）

- 如果代理要写入的队列是远程队列，那么您可能必须更改 IBM MQ 队列管理器，队列和通道 **MAXMSGL** 属性。

确保以下每个 IBM MQ 属性的值大于或等于 **maxInputOutputMessageLength** 代理属性的值：

- 代理队列管理器上远程队列管理器传输队列的最大消息大小
- 从代理队列管理器到远程队列管理器的通道的最大消息大小
- 远程队列管理器上目标队列的最大消息大小
- 远程队列管理器的最大消息大小

确保以下每个 IBM MQ 属性的值（字节）：

- 代理队列管理器的最大消息大小
- The maximum message size of the SYSTEM.FTE.STATE.<代理程序名称> queue
- 客户机通道的最大消息大小（如果您的代理以客户机方式连接到队列管理器）

大于或等于以下计算的结果：

对于“文件到消息”传输（支持最高 100 MB 的文件大小）：
maxInputOutputMessageLength 的值

对于消息到文件的传输：

$3 * (\text{maxInputOutputMessageLength}) + 1048576$ 的值

(该计算源于如下事实：三个检查点可以存储在状态消息中，并且每个检查点可能都必须缓冲，直至数据消息量的最大大小。)

如果超出其中某个属性的值，那么代理将停止，代理事件日志中会出现以下错误：

```
BFGUT0002E: An internal error has occurred. Product failure data was captured in file
"FFDC.FTE.20100928170828514.8172766022149157013.log".
BFGSS0025E: An internal error has occurred. The exception is: cc=2 rc=2010 op=put - MQPUT to
SYSTEM.FTE.STATE.<agent_name>
BFGAG0061E: The agent ended abnormally
```

以下 IBM MQ 原因码必须包含在代理事件日志中的此条消息内：

- **rc=2010** 该原因码映射到 **MQRC_DATA_LENGTH_ERROR**，表明已超出客户机通道最大消息大小的值。要解决此问题，请确保代理队列管理器的客户机通道最大消息大小大于或等于以下计算结果：

$3 * (\text{maxInputOutputMessageLength}) + 1048576$

- **rc=2030** This reason code maps to **MQRC_MSG_TOO_BIG_FOR_Q** and indicates that the value of the maximum message size of the **SYSTEM.FTE.STATE.<代理程序名称>** queue was exceeded. To resolve this problem ensure that the maximum message size of the **SYSTEM.FTE.STATE.<代理程序名称>** queue is greater than or equal to the result of the following calculation:

$3 * (\text{maxInputOutputMessageLength}) + 1048576$

- **rc=2031** 该原因码映射到 **MQRC_MSG_TOO_BIG_FOR_Q_MGR**，表明已超出代理队列管理器的最大消息大小的值。要解决此问题，请确保代理队列管理器的最大消息大小大于或等于以下计算结果：

$3 * (\text{maxInputOutputMessageLength}) + 1048576$

如果要传输许多小的消息

如果代理从队列读取或写入队列的消息的平均大小低于 1310 字节，且代理读取或写入了超过 10000 条消息，那么您必须增加队列管理器上未落实消息属性的最大数量，或减少检查点时间间隔中的数据量。

当代理在队列中读写消息时，对应的 **GET** 或 **PUT** 将一起分组到事务。事务中 **GET** 或 **PUT** 的数量由处理检查点时间间隔中所有数据所需的数量确定。检查点时间间隔中的大概数据量使用以下计算从代理属性中确定：

Checkpoint interval data size (in bytes) = $\text{agentCheckpointInterval} * \text{agentFrameSize} * \text{agentWindowSize} * \text{agentChunkSize}$.

缺省检查点数据大小为 $1 * 5 * 10 * 262144$ 字节 = 13107200 字节 (12.5MB)。队列管理器支持的事务中未落实消息的最大数目由 **MaxUncommittedMsgs** 队列管理器属性控制。此属性的缺省值为 10000 条消息。如果平均消息大小小于约 1310 字节，且要写入的消息数超过 10000 条，那么将超出未落实消息的缺省最大数。

如果超出 **MaxUncommittedMsgs** 限制，那么代理将停止，代理事件日志中出现以下错误：

```
BFGSS0024E: The agent has received a reason code of '2024' from the message queue interface (MQI).
The agent cannot continue processing and will now end.
BFGAG0139I: The agent has suspended its current transfers and is now stopping.
```

原因码 2024 映射到：**MQRC_SYNCPOINT_LIMIT_REACHED**。

要解决此问题，请执行以下某项操作

- 增加读写队列的代理连接到的队列管理器的 **MaxUncommittedMsgs** 队列管理器属性的值。请参阅 **MaxUncommittedMsgs (MQLONG)**。
- 减少检查点时间间隔中的数据量。要执行此操作，请减少以下一个或多个代理属性的值：

- agentCheckpointInterval
- agentFrameSize
- agentWindowSize
- agentChunkSize

要了解有关这些代理属性的信息，请参阅[高级代理属性](#)。

如果将消息持久写入队列

如果要传输到队列并将消息持久写入队列，那么可能需要增加队列管理器日志文件空间的大小，以便能够记录检查点时间间隔中的所有数据。

如果超出队列管理器日志文件空间，那么代理将停止，代理事件日志中出现以下错误：

```
BFGSS0024E: The agent has received a reason code of '2102' from the message queue interface (MQI).
The agent cannot continue processing and will now end.
BFGAG0062E: The agent has received MQI reason code '2102'. The agent cannot continue processing and
will now end.
BFGAG0061E: The agent ended abnormally
```

原因码“2102”映射到：MQRC_RESOURCE_PROBLEM。

要解决此问题，请增加目标代理队列管理器日志文件空间的大小。

相关概念

第 263 页的『将数据从消息传输到文件』

通过使用 IBM MQ Managed File Transfer 的消息到文件传输功能，您可以将 IBM MQ 队列上的一条或多条消息的数据传输到文件、数据集或用户文件空间。如果您拥有一个可创建或处理 IBM MQ 消息的应用程序，那么可以使用 IBM MQ Managed File Transfer 的消息到文件传输功能，将这些消息传输到 IBM MQ Managed File Transfer 网络中任何系统上的文件。

第 250 页的『将数据从文件传输至消息』

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

相关参考

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

使用用户帐户控制 (UAC) 和虚拟存储器

在 Windows Server 2008 R2 和其他类似操作系统中提供了用户帐户控制。这是一个安全基础结构，其功能之一是将中央 Program Files 目录中存储的用户数据转移到用户位置，这称为虚拟存储器。

如果仅使用 IBM MQ Managed File Transfer 工具来管理数据结构，那么 IBM MQ Managed File Transfer 不受 UAC 和虚拟存储器的影响。但是，如果某个非 IBM MQ 管理员使用标准操作系统工具更改或重建了目录结构，那么新结构可能会转移到虚拟存储器。这可能会导致发生以下一种或多种情况：

- 用户（包括 IBM MQ 管理员）无法再在预期位置中看到文件。
- 代理可能无法启动，并报告消息 BFGCL0315，但不提供任何支持性原因码。
- 在代理报告的位置找不到任何日志文件。
- 代理在通过 **-F** 参数启动时可能无法启动，并报告如下消息：

```
The current directory is invalid
```

要更正上述所有情况：

- 以 IBM MQ 管理员身份使用 **fteDeleteAgent** 和 **fteCreateAgent** 命令来重建代理结构。

- 以操作系统管理员身份从受影响用户的虚拟存储器中移除 IBM MQ 条目。例如，在 Windows 上，虚拟存储器的位置如下：`%USERPROFILE%\AppData\Local\VirtualStore\`

相关参考

第 522 页的『[fteDeleteAgent](#)（删除 IBM MQ Managed File Transfer 代理）』

fteDeleteAgent 命令删除 IBM MQ Managed File Transfer 代理及其配置。如果该代理是协议网桥代理，那么用户凭证文件将留在文件系统上。

第 460 页的『[fteCreateAgent](#)（创建 IBM MQ Managed File Transfer 代理）』

fteCreateAgent 命令创建代理及其关联的配置。

有关作为 Windows 服务运行代理或记录器的指南

您可以作为 Windows 服务来运行 IBM MQ Managed File Transfer 代理、独立数据库记录器和独立文件记录器。如果这些 Windows 服务出现任何问题，您可以使用服务日志文件以及本主题中的信息来诊断问题。

要了解有关配置代理、独立记录器或独立文件记录器以作为 Windows 服务运行的更多信息，请参阅第 216 页的『[将代理作为 Windows 服务启动](#)』和第 548 页的『[fteModifyLogger](#)（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）』。

日志文件的位置

如果使用 **fteCreateAgent**、**fteCreateWebAgent**、**fteCreateCDAgent**、**fteCreateBridgeAgent**、**fteModifyAgent**、**fteCreateLogger** 或 **fteModifyLogger** 命令以作为 Windows 服务来运行代理或记录器，那么您可以使用 **-sl** 参数来选择日志记录的级别。此参数的可能值为 `error`、`info`、`warn` 和 `debug`。缺省值为 `info`。

Windows 服务的日志文件的文件名为 `servicedate.log`，其中 `date` 是启动服务的日期。代理的文件会写入目录 `MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\agents\agent_name` 中。此目录就是 IBM MQ Managed File Transfer 代理跟踪文件写入的同一目录。记录器的文件会写入目录 `MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\loggers\logger_name` 中。

如果在作为 Windows 服务启动代理程序或独立记录器时迁到问题，请尝试使用 **-sl** 参数将日志记录级别设置为 `debug`。其他信息会写入 `servicedate.log` 文件。

注: 当日志记录级别设置为 `debug` 时，用于运行 Windows 服务的用户帐户和密码将以纯文本形式显示在日志文件中。

日志文件的数目

使用 **fteCreateAgent**、**fteCreateWebAgent**、**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。作为命令的一部分指定以下文本以更改日志文件的数目：`-sj -Dcom.ibm.wmqfte.daemon.windows.windowsServiceLogFiles=number`，其中 `number` 是您需要的日志文件的数目（表达为正整数）。如果不指定日志文件的数目，那么缺省值为 5。

“作为服务登录”权限

用于运行服务的 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。

要防止此错误，请向用于运行服务的 Windows 帐户提供 **Log on as a service** 权限。例如，在 Windows 7 上，请完成以下步骤：

1. 从开始菜单，单击管理工具 > 本地安全策略。
2. 在安全性设置窗格中，展开本地策略，然后单击用户权限分配。
3. 在策略和安全性设置窗格中，双击作为服务登录。
4. 单击添加用户或组，然后将您要运行该服务的用户添加到具有 **Log on as a service** 权限的用户列表中。您已在运行 **fteCreateAgent**、**fteCreateWebAgent**、**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** 来运行，此用户名的关联密码为 **ftepassword**。运行 **fteCreateAgent** 命令或其他一些接受 **-s** 参数的命令时，您将以纯文本形式为 Windows 帐户指定密码。如果偏好于不显示密码，请执行以下步骤：

1. 运行 **fteCreateAgent**、**fteCreateWebAgent**、**fteCreateCDAgent**、**fteCreateBridgeAgent**、**fteModifyAgent**、**fteCreateLogger** 或 **fteModifyLogger** 命令时，不要指定 **-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 的“服务”工具设置密码，然后服务才能成功启动。

2. 打开 Windows 的“服务”窗口
3. 在服务列表中，右键单击代理或独立记录器服务，然后选择属性。代理程序服务显示名称为 WebSphere MQ 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 WebSphere MQ Managed File Transfer logger for property set coordination_qmgr_name，其中 coordination_qmgr_name 是您为要用作其属性集的独立记录器指定的协调队列管理器。有关属性集合的更多信息，请参阅第 575 页的『fteStartLogger（启动记录器）』和第 548 页的『fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）』。
4. 在属性窗口中，选择登录选项卡。
5. 在密码和确认密码字段中，输入运行服务的用户帐户的密码。输入密码时，密码字符是隐藏的。
6. 单击确定。

已知问题

使用 **JAVA_HOME** 系统环境变量时出现问题（仅适用于 **IBM MQ Managed File Transfer V7.5.0.1** 或更低版本）。

不得设置 **JAVA_HOME** 系统环境变量，否则，代理或记录器 Windows 服务不可能启动。代理程序或记录器 Windows 服务必须与 Websphere MQ Java 运行时一起运行。

相关任务

第 216 页的『将代理作为 Windows 服务启动』

您可以将代理作为 Windows 服务启动，以便当您从 Windows 注销后，代理可以继续运行，并且可以接收文件传输。

相关参考

第 460 页的『fteCreateAgent（创建 IBM MQ Managed File Transfer 代理）』

fteCreateAgent 命令创建代理及其关联的配置。

第 546 页的『[fteModifyAgent（修改 IBM MQ Managed File Transfer 代理）](#)』

fteModifyAgent 命令修改现有代理程序，以便可以将其作为 Windows 服务运行。此命令仅在 Windows 上可用，并且必须由作为 IBM MQ 管理员和 mqm 组成员的用户运行。

第 516 页的『[fteCreateWebAgent（创建 IBM MQ Managed File Transfer Web 代理）](#)』

fteCreateWebAgent 命令创建代理及其关联的配置，以用于 Web 网关。IBM MQ Managed File Transfer 服务器随附该命令。

第 469 页的『[fteCreateCDAgent（创建 Connect:Direct 网桥代理）](#)』

fteCreateCDAgent 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

第 463 页的『[fteCreateBridgeAgent（创建和配置 IBM MQ Managed File Transfer 协议网桥代理）](#)』

fteCreateBridgeAgent 命令创建协议网桥代理及其关联的配置。针对您要与之进行文件的发送和接收操作的每个文件服务器，创建一个协议网桥代理。

第 474 页的『[fteCreateLogger（创建 IBM MQ Managed File Transfer 记录器）](#)』

使用 **fteCreateLogger** 命令可创建文件或数据库记录器。

第 548 页的『[fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）](#)』

使用 **fteModifyLogger** 命令修改记录器，以便记录器能够作为 Windows 服务运行。只能在 Windows 平台上使用此命令，必须由作为 IBM MQ 管理员和 mqm 组成员的用户运行，并且必须首先使用

fteStopLogger 命令停止记录器。

第 575 页的『[fteStartLogger（启动记录器）](#)』

fteStartLogger 命令启动 IBM MQ Managed File Transfer 日志记录应用程序。

V 8.0.0.9 用于更新代理或记录器 JVM 选项的指南

如果使用 **fteModifyAgent** 或 **fteModifyLogger** 命令的 **-serviceJVMOptions** 参数，以通过更新、添加或移除 Java 系统属性来修改代理或记录器的现有 Windows 服务定义，那么会首先删除现有 Windows 服务，然后在其所在位置创建新服务，并使用新 Windows 服务的属性来更新代理或记录器属性文件。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 V 8.0.0 修订包 9 起，在 APAR IT22423 下添加了其他检查，以便对使用 **fteModifyAgent** 或 **fteModifyLogger** 命令的 **-serviceJVMOptions** 参数指定的代理或记录器 JVM 选项所做的任何更新都会进行验证，确保已正确指定这些选项。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，并且删除现有 Windows 服务成功，但是在 **fteModifyAgent** 或 **fteModifyLogger** 命令创建新的 Windows 服务时发生故障，那么该命令会尝试从代理程序或记录器属性文件中除去用于定义替换 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

第 546 页的『[fteModifyAgent（修改 IBM MQ Managed File Transfer 代理）](#)』

fteModifyAgent 命令修改现有代理程序，以便可以将其作为 Windows 服务运行。此命令仅在 Windows 上可用，并且必须由作为 IBM MQ 管理员和 mqm 组成员的用户运行。

第 548 页的『[fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）](#)』

使用 **fteModifyLogger** 命令修改记录器，以便记录器能够作为 Windows 服务运行。只能在 Windows 平台上使用此命令，必须由作为 IBM MQ 管理员和 mqm 组成员的用户运行，并且必须首先使用

fteStopLogger 命令停止记录器。

有关配置资源监视器以避免代理超负载的指南。

您可以配置 IBM MQ 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。

资源监视概述

资源监视器轮询目录或队列时，代理将完成以下阶段：

- 查找所有与触发模式匹配的文件（例如，目录中的所有 *.txt 文件）。或查找队列上所有完整的消息组。
- 确定哪些文件是新文件，哪些文件已发生更改，或者确定队列上的哪些组是新组。
- 对与前两个阶段中的条件匹配的文件或组启动传输。
- 添加到已传输文件和组的列表中，这样不会再次传输它们直至其发生更改。

对于目录监视器，源目录中的文件越多，触发模式越广泛，代理必须进行解析并与已传输文件列表进行比较的文件列表也就越长。

对于队列监视器，队列上的组越多，针对已传输组列表，代理必须比较的组列表就越大。

考虑以下关键设置：

- 使用代理属性 **monitorMaxResourcesInPoll** 设置代理每次轮询时包含的最大文件或组数。使用该参数限制轮询时间间隔内的传输数。这还表示，在为该数量的文件或组启动传输之前，代理需要执行较少的解析。目录监视器或队列监视器下次轮询时，代理将包含下一组文件或组。代理程序属性 **monitorMaxResourcesInPoll** 在 IBM MQ Managed File Transfer V 7.0.4.1 和更高版本中可用，对于较低版本的 IBM MQ Managed File Transfer，它可用作 APAR IC78011 的临时修订。
- 创建目录监视器时，请确保所配置的传输定义的源处置为 **delete**。设置该处置意味着，文件传输完成后，会将其从受监视的目录中移除并且代理不再将其保留在内部列表中。
- 创建目录监视器时，请在 **fteCreateMonitor** 命令中使用 **-rl** 参数来限制代理必须递归的目录级别数。使用该参数表示不必扫描较低级别的目录。

创建资源监视器时的更多注意事项

资源监视器轮询过程将耗用代理资源。增大监视器的轮询时间间隔将减少代理上的负载。但是，轮询时间间隔的设置必须均衡，以防止每个轮询时间间隔内生成过多的传输。为资源监视器设置轮询时间间隔时，请考虑以下内容：

- 将文件放入目录或将组放入队列后，需要以多快的速度启动传输。
- 将文件放入目录或将组放入队列的速率。
- 最大代理传输速率。代理必须能够处理监视器生成的所有传输。

使用 **fteCreateMonitor** 命令来创建资源监视器时，通过指定 **-pi**（轮询时间间隔）和 **-pu**（轮询时间间隔单位）参数指定轮询时间间隔。您可能需要试验以确定适合您配置的最佳设置。

可提高运行资源监视器的高负载代理的稳定性的选项是降低代理属性 **maxSourceTransfers** 的值。通过此选项，代理将在资源监视器和传输文件之间分配其处理时间。代理属性 **maxSourceTransfers** 的值越高，传输文件所耗用的处理时间就越多，资源监视器可用的处理时间就越少。如果降低代理属性 **maxSourceTransfers** 的值，那么代理将并行执行较少的传输，但应具有足够的处理时间来轮询其资源监视器。如果降低该代理属性的值，那么应考虑增加代理属性 **maxQueuedTransfers** 的值，因为排队传输数可能会增加。

如果在优化监视器后发现一些传输进入恢复状态，请考虑增加代理超时值。与目标代理协商启动传输时，代理上的重负载可能意味着传输超时。该超时将导致传输进入恢复状态并延迟完成传输。代理属性 **maxTransferNegotiationTime** 指定源代理等待目标代理响应的的时间。如果超出该时间，那么传输将进入恢复状态。该属性的缺省值为 30000 毫秒（30 秒）。增加该属性的值（例如，增加至 300000 毫秒（5 分钟））可以使传输在不超时的情况下继续并避免进入恢复状态。

相关概念

第 229 页的『资源监控』

您可以监视 IBM MQ Managed File Transfer 资源；例如，队列或目录。当满足针对该资源的条件时，资源监视器将启动任务，如文件传输。您可以使用 **fteCreateMonitor** 命令或 IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

第 222 页的『使用传输定义文件』

您可以指定可用于创建文件传输的传输定义文件。传输定义文件是定义创建传输所需的部分或全部信息的 XML 文件。

相关参考

第 478 页的『fteCreateMonitor（新建资源监视器）』

fteCreateMonitor 命令从命令行创建并启动新的资源监视器。您可以使用 WebSphere MQ Managed File Transfer 监视资源（例如，目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

当变量替换导致多个文件传输到单个文件名时该怎么做

对于 Managed File Transfer，如果要监视某个目录并将多个文件从源位置传输到目标位置，并且要使用 `${FileName}` 变量替换，那么必须测试变量替换的结果。需要对结果进行测试的原因在于：使用变量替换可能导致调用意外的文件传输命令组合。

要确定是否发生此问题，请查看那些看似传输了多个文件但实际只有一个文件到达目标的案例。您可能在文件传输日志中看到一些错误，这些错误表明有多个文件尝试传输到相同目标文件名但传输到相同文件名失败。

发生此问题的原因

当 MFT 目录监视器处理多个文件时，将针对监视器在受监视目录中找到的每个文件都运行任务 XML。如果仅在 XML 任务文件的目标中指定 `${FileName}`，而在源中未指定，那么将针对每个文件多次调用传输（针对每个文件名组合各调用一次）。

例如：

```
<source disposition="delete" recursive="false">
  <file>e:\temp</file>
</source>
<destination exist="overwrite" type="file">
  <file>s:\outdir\${FileName}</file>
</destination>
```

避免此问题

如果要在源和目标中使用 `${FileName}` 变量替换，并且期望相同文件名的变体到达目标，请务必在任务 XML 定义的源和目标中同时指定 `${FileName}`。

以下示例将提取 `e:\temp\<filename>` 中的一个文件，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>
```

相关概念

第 239 页的『使用变量替换定制 MFT 任务』

在满足活动资源监视器的触发条件时，会调用已定义的任务。除了每次使用相同的目标代理或相同的目标文件名调用传输或命令任务之外，您还可以在运行时修改任务定义。可通过将变量名插入到任务定义 XML 中来实现此目的。在监视器确定已满足触发条件并且任务定义包含变量名时，将用变量值替换变量名，然后调用任务。

第 240 页的『示例：变量替换』

使用 XML 和 MQ Explorer 对资源监视器定义进行变量替换的示例。

如果在 Oracle 数据库上更新数据库模式时收到错误

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

要修复该问题，必须先编辑 `ftelog_tables_oracle_702_703.sql` 的内容，然后再运行该文件。

过程

1. 查明 IBM MQ Managed File Transfer 数据库记录器表所使用的序列和触发器位于哪个模式中。
 - 在 Db2 上，可以使用“控制中心”来查看表和模式。
 - 在 Oracle 上，可以使用 Enterprise Manager 来查看表和模式。
2. 在文本编辑器中打开 `ftelog_tables_oracle_702_703.sql` 文件。
3. 在每次出现的文本 `SELECT FTELOG.sequence_name.nextval` 中，将文本 `FTELOG` 替换为现有序列所在模式的名称。
4. 在每次出现文本 `CREATE OR REPLACE TRIGGER FTELOG.trigger_name` 之前，插入文本 `DROP TRIGGER schema_name.trigger_name`，其中 `schema_name` 是现有触发器所在的模式的名称。
5. 使用编辑过的 `ftelog_tables_oracle_702_703.sql` 文件来更新数据库表。

记录器错误处理和拒绝

记录器确定两种类型的错误：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 IBM MQ Managed File Transfer 日志消息

通用错误是每条消息错误之外的所有错误。这些错误的原因可能是配置问题或程序错误。

遇到每条消息的错误时，记录器会将该消息放到拒绝队列上以拒绝该消息。不会向输出日志写入任何内容，因此需要定期检查或持续监视拒绝队列以检测拒绝的消息。

如果连续拒绝了过多消息，而没有任何消息成功写入数据库，那么这种情况将视为通用错误。例如，考虑这样一个站点，它始终使用 10 字符代码作为作业名，但无意间将作业名列重新配置为两个字符宽。尽管过宽的数据通常为每条消息错误，但在此情况下，配置问题是通用的，因此将检测为通用错误。您可以使用 **`wmqfte.max.consecutive.reject`** 属性来调整造成通用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。如果记录器是通过 **`-F`** 参数以前台方式启动的，那么会将确定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。

拒绝队列

产生每条消息错误的消息将移动到拒绝队列中。在每条拒绝的消息上，将设置一个消息属性以指示拒绝该消息的原因。该属性的全名是 **`usr.WMQFTE_ReasonForRejection`**，尽管是 `usr`。在某些上下文（包括 JMS 和 IBM MQ Explorer）中省略。

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。

有时，您可能希望重新处理拒绝队列中的消息。在本主题先前描述的示例中，对于数据库中的两字符作业名，在数据库列宽度增加后可以成功处理这些消息。作为另一示例，如果“传输完成”消息由于缺少关联的“传输启动”而被拒绝，可能会在以后接收到“传输启动”消息。这样，重新处理“传输完成”将会成功。

要重新处理消息，请将这些消息从拒绝队列移动到输入队列。在正常安装中（其中记录器创建了自己的受管预订），输入队列由队列管理器定义并具有类似 SYSTEM.MANAGED.DURABLE.49998CFF20006204 的名称。您可以通过查看预订 SYSTEM.FTE.DATABASELogger.AUTO 的属性中的**目标名称**或使用以下 MQSC 命令来识别输入队列：

```
DISPLAY SUB(SYSTEM.FTE.DATABASELogger.AUTO) DEST
```

在队列之间移动消息的一种方法是使用 [MA01 SupportPac](#)，例如：

```
q -IFTE.REJECT -oSYSTEM.MANAGED.DURABLE.49998CFF20006204
```

拒绝队列可能包含由于各种原因而被拒绝的消息，其中仅有一部分得以解决。在此情况下，您仍可以重新处理所有消息；现在可接受的消息将被使用，而不能接受的消息则再次移动到拒绝队列。

传输日志中的 **Malformed** 日志消息不是由记录器记录的。这些消息不会视为重要消息，因此这些消息将发送到拒绝队列。有关传输日志消息的更多信息，请参阅第 662 页的『[文件传输日志消息格式](#)』。

如果记录器已启动，但没有任何传输信息记录到数据库

IBM MQ Managed File Transfer 记录器使用的数据库表要求数据库的页面大小为 8KB 或更大。如果数据库的页面大小不足，那么无法正确创建数据库表，并且您将看到 SQLSTATE=42704 错误。

如果您正在使用 Java Platform, Enterprise Edition 数据库记录器，那么可能会在 WebSphere Application Server 系统输出日志中看到以下消息；如果您正在使用独立数据库记录器，那么可能会在 output0.log 文件中看到以下错误：

```
DB2 SQL Error: SQLCODE=-204, SQLSTATE=42704
SQLERRMC=FTELOG.TRANSFER_EVENT, DRIVER=3.40.152
```

SQLSTATE 值为 42704 指示记录器预期存在的某个表（在本例中为 FTELOG.TRANSFER_EVENT）并不存在。

要修复此问题，请执行下列步骤：

1. 检查表是否存在且是否完整。有关记录器所使用的表及其列的信息，请参阅第 731 页的『[记录器所使用的数据库表](#)』。
2. 如果表不存在或者不完整，请检查数据库的页面大小。
3. 如果数据库大小小于 8KB，请增加数据库页面的大小。
 - 如果数据库位于测试系统上或者数据库中没有数据，您可以删除表并重新创建一个页面大小大于 8KB 的数据库。
 - 有关如何增加页面大小的信息，请参阅第 32 页的『[在 Windows, UNIX 或 Linux 上的 Db2 上增加日志数据库的页面大小](#)』或第 34 页的『[将 Db2 on z/OS 上的数据库表迁移到 V8.0.0](#)』。

未能从 AMS 中的密钥库配置文件读取密钥库属性时应该执行的操作

密钥库配置文件位置 (如果在缺省位置中不存在) 必须由 MQS_KEYSTORE_CONF 变量指定，以便 Java AMS 以客户机方式运行。如果未指定此位置，那么 IBM MQ Managed File Transfer 代理日志将显示以下错误消息：“Failed to read keystore properties from the keystore configuration file”。

密钥库配置文件的缺省位置为 <home_directory>/mqsc/keystore.conf。如果密钥库配置文件的位置不是缺省位置，请完成以下步骤：

1. 以客户机方式启动 FTE 代理。
2. 将 AMS 安全性应用于 SYSTEM.FTE.DATA.<agent name> 队列。如果密钥库配置文件不在此位置中，那么所有传输都将失败，并且不进行任何确认。
3. 针对 **fteStartAgent** 命令将系统变量 `FTE_JVM_PROPERTIES` 设置为 `FTE_JVM_PROPERTIES=-DMQS_KEYSTORE_CONF=<path to keystore_config file>`。
4. 针对 **fteStartAgent** 命令将系统变量 `MQS_KEYSTORE_CONF` 设置为 `MQS_KEYSTORE_CONF=<path to keystore_config file>`。凡是代理，不管其运行方式如何，必须设置此变量才能确保可运行。

注: 如果 Java AMS 以绑定方式运行，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。

BFGSS0023E 错误以及如何避免这些错误

如果从安装中卸载修订包以返回至产品的先前版本，并在发生卸载时受管传输涉及了与安装相关联的代理，那么该代理无法启动，并将报告 BFGSS0023E 错误。要避免此错误，您可以完成一些步骤以防止在重新启动代理时显示 BFGSS0023E 消息。

对于当前涉及代理的正在进行中的每个受管传输，在代理的 SYSTEM.FTE.STATE.*agent_name* 队列上有一条消息。此消息在受管传输上存储检查点信息，并且在受管传输进入恢复时使用。在受管传输完成后，将除去 SYSTEM.FTE.STATE.*agent_name* 队列上的相应消息。

每条状态消息都包含一些内部头信息，指示在受管传输运行时代理所使用的 IBM MQ 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 V 8.0.0 修订包 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 V 8.0.0 修订包 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 V 8.0.0 修订包 5 代理在停止并降级到 V 8.0.0 修订包 4 级别时正在运行一些进行中的传输，那么代理在下次启动时，会检查其 SYSTEM.FTE.STATE.*agent_name* 队列上的消息，并发现这些消息是在它使用 V 8.0.0 修订包 5 时写入的。由于代理现在使用的是 V 8.0.0 修订包 4，因此它会报告先前段落中描述的 BFGSS0023E 错误并关闭自身。

作为一般规则，如果您要除去 IBM MQ Managed File Transfer 组件的修订包，那么完成以下步骤应该会防止在重新启动代理时显示 BFGSS0023E 消息：

1. 确保其所有代理已完成其受管传输。
2. 停止这些代理。
3. 除去修订包。
4. 重新启动这些代理。

相关任务

[第 215 页的『启动 IBM MQ Managed File Transfer 代理』](#)

必须先启动 IBM MQ Managed File Transfer 代理，然后才能将其用于文件传输。

相关参考

[第 697 页的『IBM MQ Managed File Transfer 的代理队列』](#)

由 **fteCreateAgent** 命令生成的 MQSC 命令脚本使用设置为以下值的参数创建代理队列。如果不使用提供用于创建队列的 MQSC 脚本，而是手动创建队列，那么请确保将以下参数设置为给定的值。

相关信息

[BFGSS0001 - BFGSS9999](#)

当受管传输失败并返回 BFGIO0341E 错误时应执行的操作

如果受管传输正在将文件传输到由外部进程监视的位置，那么该受管传输可能会失败并返回以下错误：

BFGIO0341E: 将临时文件 *destination_filename.part* 重命名为 *destination_filename* 失败，因为该临时文件不存在。这是由于受管传输的目标代理在写入目标文件时使用临时文件的方式造成的。

目标代理使用临时文件的方式

缺省情况下，当进行受管文件传输时，目标代理将执行以下步骤：

- 创建名为 `destination_filename.part` 的临时文件。
- 锁定该临时文件。
- 从源代理接收文件数据时，会将这些数据写入到该临时文件。
- 在接收并写入所有文件数据后，解锁该临时文件。
- 将临时文件从 `destination_filename.part` 重命名为 `destination_filename`。

如果受管传输进入恢复状态，那么目标代理可能会创建名为 `destination_filename.partnumber` 的临时文件。然后，目标代理会将文件数据写入此文件，而不是名为 `destination_filename.part` 的文件。

如果已存在临时文件名 `destination_filename.partnumber`，那么目标代理会尝试创建名为 `destination_filename.part(number + 1)` 的新临时文件。如果已存在该文件，那么目标代理会尝试创建名为 `destination_filename.part(number + 2)` 的临时文件，以此类推，直到能够成功创建该文件。在代理尝试创建临时文件 `destination_filename.part1000` 但失败的情况下，它将直接写入到目标文件，而不使用临时文件。

当受管传输完成时，目标代理会删除名为 `destination_filename.partnumber` 的所有临时文件，因为会假定这些文件是由代理在受管传输期间创建的。

注：如果代理属性 **doNotUseTempOutputFile** 设置为值 `true`，那么目标代理不使用临时文件。而是直接写入到目标文件。有关 **doNotUseTempOutputFile** 属性的更多信息，请参阅 [MFT agent.properties](#) 文件。

发生此问题的原因

如果目标代理尝试重命名临时文件，但发现该文件不再存在，那么会生成 BFGIO0341E 错误。导致此问题的典型场景如下：

- 已在目标文件系统中设置了一个登台目录。
- 外部进程配置为监视该登台目录，并将其找到的任何文件移到新位置。
- 目标代理会在登台目录中创建并锁定临时文件 `destination_filename.part`。
- 目标代理会将文件数据写入到该临时文件。
- 将所有文件数据写入到该临时文件后，目标代理会解锁该文件。
- 外部进程找到该临时文件，并将其移到新位置。
- 目标代理尝试重命名该临时文件，发现该文件已不存在。因此，传输项将标记为 "失败"，并返回 BFGIO0341E 错误。

避免此问题

可以通过两种方法来防止发生 BFGIO0341E 错误：

- 目标代理写入的临时文件始终以 `.part` 或 `.partnumber` 后缀结尾。如果可以将外部进程配置为忽略这些文件（而不是移动它们），那么当目标代理执行重命名操作时，这些文件仍将存在于目标目录中。
- 也可以配置目标代理，使其不使用临时文件，而是直接写入到目标文件。只有在所有文件数据都已写入到目标文件时，目标文件才会解锁，此时外部进程可以选取目标文件。

要将目标代理配置为直接写入到目标文件，请设置代理属性 **doNotUseTempOutputFile=true**。有关此属性的更多信息，请参阅 [MFT agent.properties](#) 文件。

IBM MQ Managed File Transfer 的返回码

IBM MQ Managed File Transfer 命令、Ant 任务和日志消息提供返回码以指示是否成功完成了功能。

下表列出了产品返回码及其含义：

表 24: 返回码		
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容答复命令状态。缺省情况下，该超时对于受管调用和传输命令而言是无限的。例如，当您使用 fteCreateTransfer 命令指定 -w 参数时。缺省情况下，该超时对于其他命令是 5 秒。
3	确认超时	代理在指定的超时内未确认接收到命令。缺省情况下，该超时是 5 秒。
4	错误的代理	将命令发送到了错误的代理。在命令 XML 中指定的代理不是正在读消息所在命令队列的代理。
20	传输部分成功	传输已完成，但只有部分成功，传输了一些文件。
21	传输已停止	传输已被一个用户出口停止。
22	取消传输超时	代理已收到取消传输的请求，但该取消未能在 30 秒内完成。未取消该传输。
26	未找到取消标识	代理已收到取消传输的请求，但未找到该传输。这可能是因为代理处理该取消请求之前，该传输已完成。这也可能是因为，您向 fteCancelTransfer 命令提供的传输标识不正确。已忽略该取消请求。
27	正在取消中	代理已收到取消传输的请求，但该传输已处于被取消的过程中。已忽略新的取消传输请求。
40	失败	传输失败，未传输任何指定的文件。
41	已取消	已取消该传输。
42	触发失败	传输未进行，因为传输是有条件的并且未满足所需条件。
43	格式不正确的 XML	XML 消息的格式不正确。
44	已超出源代理容量	源代理没有足够的容量来执行该传输。
45	已超出目标代理容量	目标代理没有足够的容量来执行该传输。
46	已超出源代理最大文件数	要传输的文件数已超出源代理的限制。

表 24: 返回码 (继续)

返回码	短名称	描述
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 WebSphere MQ 问题, 源代理不能将消息发送到目标代理。例如, 如果源代理队列管理器未正确配置以与目标代理队列管理器进行通信, 将发生该问题。
50	试用版违例	试用版代理尝试与非试用版代理进行通信。
51	未准许源传输	maxSourceTransfers 代理属性已设置为 0。不允许此代理作为任何传输的源。
52	未准许目标传输	maxDestinationTransfers 代理属性已设置为 0。不允许将此代理程序作为任何传输的目标。
53	未经授权	用户无权执行该操作。请参阅随附消息, 以获取进一步的详细信息。
54	权限级别不匹配	源代理和目标代理的 authorityChecking 代理属性值不匹配。
55 个	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。
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。
60	文件空间查找异常	Web 网关代理文件空间查找失败。
61	未找到文件空间异常	Web 网关代理未在数据库中找到任何数据。
62	未授权文件空间异常	许可权数据库未授权 Web 网关代理文件空间用户完成传输。
63	文件空间删除操作异常	Web 网关代理文件空间正在被 Web 网关删除。
64	两个队列都不被允许	传输的源和目标是队列。

表 24: 返回码 (继续)

返回码	短名称	描述
65	一般数据队列错误	访问 IBM MQ Managed File Transfer 代理数据队列时发生错误。
66	数据队列 put 权限错误	访问 IBM MQ Managed File Transfer 代理数据队列时发生错误。未启用 WebSphere MQ Advanced Message Security。
67	数据队列 put AMS 错误	访问 IBM MQ Managed File Transfer 代理数据队列时发生权限错误。启用 WebSphere MQ Advanced Message Security。
68	不支持传输	尝试创建 Web 代理不支持的传输。Web 代理仅支持它充当目标代理并且目标是文件空间的传输。
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。
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	尝试创建或删除协议网桥代理或 Web 代理上的资源监视器。该行为不受支持。

表 24: 返回码 (继续)		
返回码	短名称	描述
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在此监视器资源队列。
118	监视器资源表达式无效	对 XPath 表达式求值时发生错误。对 XPath 表达式求值，以访问消息头中用户定义的属性。消息位于资源监视器监视的队列上。
119	监视器任务源代理队列管理器缺失	监视器任务定义中缺少源代理名称或源代理队列管理器名称。
120	监视器队列未启用	监视器资源队列未启用。
121	访问监视队列时发生意外错误	访问监视资源队列时发生了意外错误。
122	未对上下文标识启用监视器命令队列	未对设置上下文标识启用监视器代理命令队列。

下表列出了产品中间回复代码及其含义：

表 25: 中间回复代码		
回复代码	短名称	描述
-2	ACK	请求已收到，但正在暂挂完成。
-3	PROGRESS	该请求用于多个文件，但其中某些仍暂挂完成。

注：

仅在生成请求的进程提供了回复队列的情况下，才存在回复代码。这些都是中间回复，IBM MQ Managed File Transfer 命令仅返回最终回复代码。

相关参考

第 409 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。

第 410 页的『[HTTP 响应代码](#)』

向 IBM MQ Managed File Transfer Web 网关发出的请求的 HTTP 响应中会返回状态码。

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 "C:\destinationfiles\dest1.doc">
```

```
already exists.</supplement>
</status>
```

下表描述了 `resultCode` 的可能值：

表 26: 传输中的文件结果代码	
结果代码值	描述
0	成功。文件传输成功。
1	失败。文件传输失败。请参阅 <code><supplement></code> 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 <code><supplement></code> 元素，以获取该警告的更多详细信息。

HTTP 响应代码

向 IBM MQ Managed File Transfer Web 网关发出的请求的 HTTP 响应中会返回状态码。

Web 网关返回的响应头中包含 HTTP 响应代码。以下示例中的 HTTP 头包含 HTTP 响应代码 200 OK：

```
HTTP/1.1 200 OK
Server: WAS/6.0
Content-length: 0
```

下表描述了 HTTP 响应代码的可能值，以及可返回的关联 IBM MQ Managed File Transfer 错误代码的示例。有关 IBM MQ Managed File Transfer 错误代码的更多信息，请参阅[诊断消息](#)。

表 27: HTTP 响应代码		
HTTP 响应代码	IBM MQ Managed File Transfer 错误代码示例	示例描述
200 OK	None	已正确处理有效请求，且已选择性地向用户提供响应。
202 Accepted	None	已正确处理有效请求，但 IBM MQ Managed File Transfer 不保证请求的操作已完成。 例如，文件上载传输请求已处理并提交到 IBM MQ Managed File Transfer 代理，但尚未进行传输。
400 Bad Request	BFGWI0001	由于 URI 缺少资源类型，因此该 URI 无效。
403 Forbidden	BFGWI0056	没有为该用户定义 IBM MQ 消息描述符 (MQMD) 用户标识。
404 Not Found	BFGWI0015	找不到请求的资源。
405 Method Not Allowed	BFGWI0016	请求的资源不支持请求中所使用的 HTTP 谓词。 例如，对仅支持 POST 或 DELETE 的资源使用了 GET 操作。
410 Resource Gone	BFGWI0031	所请求的资源不再可用。例如，请求的文件已从文件空间中删除。

表 27: HTTP 响应代码 (继续)		
HTTP 响应代码	IBM MQ Managed File Transfer 错误代码示例	示例描述
413 Request Entity Too Large	BFGWI0026	请求包含的某个文件过大，服务器无法处理。
415 Unsupported Media Type	BFGWI0017	接收的请求具有 Content-type HTTP 头指定的介质类型，但该类型不受支持。
500 Internal Server Error	BFGWI0018	处理请求时迁到内部错误。已生成 FFDC 或 ABEND 文件。
502 Bad Gateway	BFGWI0019	由于 IBM MQ Managed File Transfer 外部发生错误，无法完成请求。例如，IBM MQ 队列管理器不可用。
503 Service Unavailable	BFGWI0020	该目标暂时不可用。例如，IBM MQ 队列已满。
504 Gateway Timeout	BFGWI0021	由于超出 IBM MQ Managed File Transfer 或 HTTP 客户机设定的时间限制，尝试完成请求超时。

相关概念

第 412 页的『[对 Web 网关进行故障诊断](#)』

使用以下参考信息和示例可帮助您诊断 Web 网关返回的错误。

第 305 页的『[Web 网关的方案](#)』

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

第 307 页的『[Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑](#)』

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

第 312 页的『[使用 IBM MQ Managed File Transfer 服务 Web 网关](#)』

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

第 313 页的『[示例 HTTP 流](#)』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

相关参考

第 908 页的『[Web 网关 API 引用](#)』

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

第 909 页的『[使用 Web 网关时的 HTTP 头和 HTML 表单字段](#)』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

第 911 页的『[使用 Web 网关时的统一资源标识语法](#)』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 918 页的『[使用 Web 网关的内容类型](#)』

向 IBM MQ Managed File Transfer Web 网关提交的文件传输请求必须对应于特定的介质类型。来自 Web 网关的响应具有介质类型 application/xml 或 application/json。

第 919 页的『[响应格式：XML 和 JSON](#)』

IBM MQ Managed File Transfer Web 网关采用以下两种格式之一返回查询的响应：XML 或 JavaScript Object Notation (JSON)。

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

对 Web 网关进行故障诊断

使用以下参考信息和示例可帮助您诊断 Web 网关返回的错误。

相关概念

[第 201 页的『Web 网关安装验证应用程序』](#)

IBM MQ Managed File Transfer 提供 Web 网关安装验证应用程序。使用此应用程序可查看 Web 网关安装的配置值并测试基本的 Web 网关功能。

相关任务

[第 201 页的『验证 Web 网关安装』](#)

遵循以下指示信息来检查是否已正确部署 IBM MQ Managed File Transfer Web 网关应用程序。

相关参考

[第 414 页的『启用 Web 网关跟踪』](#)

在托管 Web 网关的应用程序服务器上启用跟踪，以诊断 Web 网关的问题。

[第 416 页的『常见问题』](#)

以下引用和任务信息包含 Web 网关返回的错误示例，以及有关如何避免产生错误的提示。

验证 Web 网关安装

遵循以下指示信息来检查是否已正确部署 IBM MQ Managed File Transfer Web 网关应用程序。

开始之前

在验证 Web 网关配置之前，必须先遵循指示信息来部署 Web 网关应用程序。请参阅[第 181 页的『配置 Web 网关』](#)。

关于此任务

过程

1. 确保已使用具有 `wmqfte-admin` 安全角色的用户标识登录到应用程序服务器环境。有关更多信息，请参阅[第 108 页的『Web 网关的用户角色』](#)。
2. 在 Web 浏览器中，输入以下 URI：

```
http://host/wmqfte/ivt?logdbschema=FTELOG&webdbschema=FTEWEB
```

如果为 Web 网关应用程序定义的上下文根并非缺省值 `wmqfte`，请使用以下 URI：

```
http://host/context_root/ivt?logdbschema=FTELOG&webdbschema=FTEWEB
```

注：在配置 Web 网关期间，需要设置数据库表以存储有关文件空间和传输历史记录的信息。Web 网关安装验证应用程序假定您使用的是数据库模式名称的缺省值。如果定义的数据库模式名称并非缺省值（对于传输历史记录数据库，缺省值为 `FTELOG`；对于文件空间信息数据库，缺省值为 `FTEWEB`），那么必须更改 URI 中指定的模式名称。使用以下查询术语指定数据库模式名称：

logdbschema

传输历史记录数据库的模式名称

webdbschema

文件空间信息数据库的模式名称

例如，如果传输历史记录数据库和文件空间信息数据库的模式名称分别为 MYLOG 和 MYWEB，那么请使用以下 URI：

```
http://host/wmqfte/ivt?logdbschema=MYLOG&webdbschema=MYWEB
```

有关设置数据库的更多信息，请参阅第 181 页的『设置数据库以用于文件空间』和第 200 页的『配置数据库记录器以用于 Web 网关』。

结果

Web 浏览器显示的页面列出 Web 网关安装的配置信息，并显示某些基本 Web 网关功能的测试结果。有关更多信息，请参阅第 201 页的『Web 网关安装验证应用程序』。

Web 网关安装验证应用程序

IBM MQ Managed File Transfer 提供 Web 网关安装验证应用程序。使用此应用程序可查看 Web 网关安装的配置值并测试基本的 Web 网关功能。

有关如何访问安装验证应用程序的信息，请参阅第 201 页的『验证 Web 网关安装』。该应用程序显示两种类型的信息：Web 网关安装的配置值以及测试基本 Web 网关功能的结果。

配置值

在应用程序服务器中部署 Web 网关时，需要为多个初始化参数提供值。如果使用 WebSphere Application Server V7.0，那么需要在管理控制台中使用**初始化 servlet 的参数**步骤来提供这些值。如果使用 WebSphere Application Server Community Edition，那么可在 web.xml 文件中设置这些值。

该应用程序在标题 **Web 网关配置信息** 下面列出了以下 Web 网关设置的值：

servlet 信息

已部署的 Web 网关 servlet 的名称和版本。

Web 网关名称

部署的 Web 网关的名称。您需要为 **webGatewayName** 初始化参数提供该值。

上下文根

为 Web 网关应用程序定义的上下文根。在 WebSphere Application Server Community Edition 中，这是 WEB-INF/geronimo-web.xml 文件中 <web:context-root> 元素的值。对于 WebSphere Application Server V7.0，这是安装 Web 网关应用程序时在映射 **Web** 模块的上下文根步骤中设置的值。缺省值是 **wmqfte**。

文件空间根目录

Web 网关创建和管理的文件空间的根目录路径。您需要为 **fileSpaceRoot** 初始化参数提供该值。

临时文件上传根目录

Web 网关所启动传输的相关临时文件存储器的目录路径。您需要为 **tempFileUploadDir** 初始化参数提供该值。

临时文件上传目录的最大大小

允许用户存储与 Web 网关启动的传输相关的临时文件的最大空间量 (MB)。您需要为 **maxTempFileUploadSpace** 初始化参数提供该值。

MQMFT Web 代理名称

充当 Web 网关所启动传输的源的 IBM MQ Managed File Transfer 代理的名称。您需要为 **agentName** 初始化参数提供该值。这是运行 **fteCreateWebAgent** 命令时，使用 **-agentName** 参数为 Web 代理指定的名称。

协调队列管理器名称

Web 网关用于记录传输信息的协调队列管理器的名称。您需要为 **coordinationQMGr** 初始化参数提供该值。

缺省 MQMD 用户标识

当没有为用户定义特定 MQMD 用户标识时，用于与请求的用户关联的缺省 WebSphere MQ 消息描述符 (MQMD) 用户标识。您需要为 **defaultMQMDUserID** 初始化参数提供该值。

CSRF 保护

指示当前 Web 网关是否配置为执行 CSRF 标记验证以防止跨站点请求伪造攻击。您需要为 **CSRFProtection** 初始化参数提供该值。

应用程序服务器信息

托管 Web 网关应用程序的应用程序服务器的名称和版本。

Web 网关测试

安装验证应用程序在标题 **Web 网关测试结果** 下面显示多个测试的结果。如果测试失败，**信息**列中将显示 IBM MQ Managed File Transfer 错误代码和消息。有关错误消息的更多信息，请参阅[诊断消息](#)。列出了以下测试：

将文件上载到临时存储器

测试在**临时文件上载根目录**字段中指定的目录。应用程序会测试该目录是否存在、是否可读和可写，以及是否能够重新读取写入该目录的数据。

将文件上载到文件空间存储器

测试在**文件空间根目录**字段中指定的目录。应用程序会测试该目录是否存在、是否可读和可写，以及是否能够重新读取写入该目录的数据。

传输历史记录数据库访问

测试是否存在与传输历史记录数据库的连接。如果使用 WebSphere Application Server V7，那么应用程序会测试您在部署 Web 网关时配置的数据源。有关更多信息，请参阅第 193 页的『[准备使用 WebSphere Application Server V7.0 部署 Web 网关](#)』。如果使用 WebSphere Application Server Community Edition，那么应用程序会测试您在部署 Web 网关时配置的数据库池。有关更多信息，请参阅第 183 页的『[准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关](#)』。应用程序检查是否可以使用您在设置数据源或数据库池时提供的凭证来访问数据库。

应用程序还检查所需数据库表是否存在。有关更多信息，请参阅第 181 页的『[设置数据库以用于文件空间](#)』和第 200 页的『[配置数据库记录器以用于 Web 网关](#)』。

测试的最后部分检查是否正确定义了 Java 持久性 API (JPA) 对象。

文件空间信息数据库访问

测试是否存在与文件空间信息数据库的连接。如果使用 WebSphere Application Server V7，那么应用程序会测试您在部署 Web 网关时配置的数据源。有关更多信息，请参阅第 193 页的『[准备使用 WebSphere Application Server V7.0 部署 Web 网关](#)』。如果使用 WebSphere Application Server Community Edition，那么应用程序会测试您在部署 Web 网关时配置的数据库池。有关更多信息，请参阅第 183 页的『[准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关](#)』。应用程序检查是否可以使用您在设置数据源或数据库池时提供的凭证来访问数据库。

应用程序还检查所需数据库表是否存在。有关更多信息，请参阅第 181 页的『[设置数据库以用于文件空间](#)』和第 200 页的『[配置数据库记录器以用于 Web 网关](#)』。

测试的最后部分检查是否正确定义了 Java 持久性 API (JPA) 对象。

启用 Web 网关跟踪

在托管 Web 网关的应用程序服务器上启用跟踪，以诊断 Web 网关的问题。

相关任务

第 415 页的『[在 WebSphere Application Server Community Edition 中启用跟踪](#)』

如果 Web 网关应用程序在 WebSphere Application Server Community Edition 中运行，请遵循以下指示信息来启用 Web 网关应用程序的跟踪。Web 网关应用程序在接收和处理请求时生成跟踪。

第 415 页的『在 WebSphere Application Server V7.0 中启用跟踪』

如果 Web 网关应用程序在 WebSphere Application Server V7.0 中运行，请遵循以下指示信息来启用 Web 网关应用程序的跟踪。Web 网关应用程序在接收和处理请求时生成跟踪。

在 WebSphere Application Server Community Edition 中启用跟踪

如果 Web 网关应用程序在 WebSphere Application Server Community Edition 中运行，请遵循以下指示信息来启用 Web 网关应用程序的跟踪。Web 网关应用程序在接收和处理请求时生成跟踪。

关于此任务

跟踪文件会写入到应用程序服务器标准输出 (STDOUT) 文件中。要在 WebSphere Application Server Community Edition 中启用跟踪，请执行以下步骤：

过程

1. 在文本编辑器中打开应用程序服务器 Java 运行时环境的 logging.properties 文件。
可以在 <WASCE_JRE>/jre/lib 目录中找到 logging.properties 文件，其中 WASCE_JRE 是 WebSphere Application Server Community Edition 所使用的 Java 运行时环境的位置。
2. 将以下行添加到 logging.properties 文件中：

```
com.ibm.wmqfte.level=FINEST
com.ibm.wmqfte.handlers=com.ibm.wmqfte.ras.container.EventLogFileHandler,com.ibm.wmqfte.ras.container.TraceLogFileHandler
java.util.logging.ConsoleHandler.level=FINEST
```

3. 保存 logging.properties 文件。
4. 重新启动 WebSphere Application Server Community Edition。

相关任务

第 183 页的『准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关』

在将 IBM MQ Managed File Transfer 服务 Web 网关企业应用程序部署到 WebSphere 应用程序服务器 Community Edition 之前，请使用这些指示信息设置您的环境。定制针对您环境的示例部署计划。

在 WebSphere Application Server V7.0 中启用跟踪

如果 Web 网关应用程序在 WebSphere Application Server V7.0 中运行，请遵循以下指示信息来启用 Web 网关应用程序的跟踪。Web 网关应用程序在接收和处理请求时生成跟踪。

关于此任务

您无需重新启动应用程序服务器，即可启用跟踪。跟踪文件会写入到应用程序服务器日志目录中。要在 WebSphere Application Server V7.0 中启用跟踪，请执行以下步骤：

过程

1. 从 WebSphere Application Server V7.0 管理控制台中选择**故障诊断 -> 日志和跟踪**。
2. 在**记录和跟踪**面板上，单击部署了 Web 网关应用程序的应用程序服务器的名称。此时将打开新面板。
3. 单击**更改日志详细信息级别**以查看应用程序服务器的当前记录级别。
4. 选择**运行时**选项卡，以在应用程序服务器当前运行的实例上启用跟踪。
 - a) 将跟踪级别 `com.ibm.wmqfte=all` 添加到现有配置中。
如果已配置现有跟踪级别，请使用冒号来分隔跟踪级别。例如，如果服务器已配置跟踪级别 `*=info`，请设置 `*=info:com.ibm.wmqfte=all` 来添加 Web 网关跟踪。
 - b) 单击**确定**以保存更改。
5. 可选：如果希望在应用程序服务器重新启动后启用跟踪，请选择**配置**选项卡。
 - a) 将跟踪级别 `com.ibm.wmqfte=all` 添加到现有配置中。
如果已配置现有跟踪级别，请使用冒号来分隔跟踪级别。例如，如果服务器已配置跟踪级别 `*=info`，请设置 `*=info:com.ibm.wmqfte=all` 来添加 Web 网关跟踪。
 - b) 单击**确定**以保存更改。

相关任务

第 193 页的『准备使用 WebSphere Application Server V7.0 部署 Web 网关』

在将 IBM MQ Managed File Transfer Web 网关企业应用程序部署到 WebSphere Application Server V7.0 之前，请使用这些指示信息定义必需资源。必须针对您的环境定制示例部署计划。

常见问题

以下引用和任务信息包含 Web 网关返回的错误示例，以及有关如何避免产生错误的提示。

相关任务

第 200 页的『配置数据库记录器以用于 Web 网关』

以下示例显示当数据库记录器未正确配置时，请求传输状态的结果：

第 421 页的『由于编码问题，请求失败』

如果 WebSphere 应用程序服务器 V7.0 在缺省编码不是 UTF-8 或缺省编码无法映射到 UTF-8（例如，cp1252）的机器上运行，那么 Web 网关无法完成请求。

第 196 页的『在 WebSphere Application Server V7.0 中设置本机库路径』

如果在 WebSphere Application Server V 7.0 上部署 Web 网关应用程序或 Java Platform, Enterprise Edition 数据库记录器应用程序，并且要在应用程序与 IBM MQ 之间使用绑定方式连接，那么必须使用系统上 IBM MQ 本机库的位置来配置 IBM MQ 消息传递提供程序。

相关参考

第 416 页的『统一资源标识是否区分大小写』

通过 Web 网关的请求的 URI 有些部分区分大小写，有些部分不区分大小写。

第 418 页的『无效的查看传输状态请求』

通过 Web 网关提交查看文件传输状态的请求时，可能会接收到 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。以下示例显示请求无效传输标识的状态将返回的结果。

第 418 页的『上载文件出现的问题』

通过 Web 网关提交上载文件的请求时，可能会接收到 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。以下示例显示请求上载文件时可能导致发生错误的一些原因。

第 419 页的『尝试在不具有必需权限的情况下创建文件空间』

要通过 IBM MQ Managed File Transfer Web 网关来创建文件空间，您的用户标识必须与相应的 MQMFT 安全角色相关联。如果您尝试在不具有适当权限的情况下创建文件空间，那么将收到 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。以下示例显示了某个不具有适当权限的用户尝试创建文件空间。

第 420 页的『尝试创建已存在的文件空间』

通过 IBM MQ Managed File Transfer Web 网关创建的文件空间必须具有唯一名称。如果尝试用已被使用的名称来创建文件空间，那么将被视为尝试修改该文件空间。如果您无权修改该文件空间，那么将收到 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。

第 421 页的『Web 代理无法启动』

如果您收到来自 **fteStartAgent** 命令的错误，并且您正在尝试启动 Web 代理，请检查 `SYSTEM.FTE.WEB.gateway_name` 队列是否存在。

第 421 页的『将文件发送到文件空间时超时』

将文件从源代理程序发送到目标文件空间时，可能会看到返回码 58 和以下消息：BFGFS0008E: Failed to look up a file space '*file_space_name*' for user '*user_name*' due to a timeout. 仅当 Web 网关部署在 WebSphere Application Server Version 7.0 上时，才会发生此问题。

统一资源标识是否区分大小写

通过 Web 网关的请求的 URI 有些部分区分大小写，有些部分不区分大小写。

有关更多信息，请参阅第 911 页的『使用 Web 网关时的统一资源标识语法』。以下示例显示在 URI 中使用大写字母查找 *transfer* 资源地址的结果。

1. 该 HTTP 请求提交获取传输信息的请求：

```
GET HTTP/1.1 /TRANSFER/414d51204d554e474f4e474f4d55474d512474f4e4ca74f2
Host: example.com
User-Agent: mozilla
```


2. Web 网关将返回以下格式的 HTTP 响应:

```
HTTP/1.1 404 Not Found
Content-Type    text/html; charset=ISO-8859-1
Content-Language en-US
Content-Length  97
Connection     Close
Date           Wed, 28 Apr 2010 15:34:28 GMT
Server         WebSphere Application Server/7.0
Error 404: SRVE0190E:
File not found: /TRANSFER/414d51204d554e474f4e474f4d55474d512474f4e4ca74f2
```

错误消息从应用程序服务器返回。具体的错误消息措辞取决于部署 Web 网关的应用程序服务器。

要使请求生效, 请在请求的 URI 中以小写字母指定资源名称, 如以下示例所示:

```
1. GET HTTP/1.1 /transfer/414d51204d554e474f4e474f4d55474d512474f4e4ca74f2
Host: example.com
User-Agent: mozilla
```

如果您接收的 HTTP 响应的状态码不是 200, 请参阅 [HTTP 响应代码](#) 主题以获取更多信息。

配置数据库记录器以用于 Web 网关

以下示例显示当数据库记录器未正确配置时, 请求传输状态的结果:

关于此任务

1. 该 HTTP 请求提交传输查询:

```
GET HTTP/1.1 /transfer/414d51204d554e474f2afed834435bc6edaf323520204cee
Host: example.com
User-Agent: mozilla
```

2. Web 网关将返回以下格式的 HTTP 响应:

```
HTTP/1.1 500 Internal Server Error
Server: WAS/6.0
Content-length: 93
Content-type: text/plain

BFGWI0018E: The request could not be completed due to an internal
web application server error.
```

要配置数据库记录器以正确处理请求, 请执行以下步骤:

过程

1. 安装 IBM MQ Managed File Transfer 数据库记录器。有关如何安装和配置数据库记录器的更多信息, 请参阅第 151 页的『[配置 Managed File Transfer 记录器](#)』。
2. 如果已安装 IBM MQ Managed File Transfer 数据库记录器, 请确保数据库表是最新的。使用以下目录中提供的 SQL 文件来更新数据库表:
 - 在分布式平台上: `MQ_INSTALLATION_PATH/mqft/sql`
 - 在 z/OS 上: `MQ_INSTALLATION_PATH/mqft/sql`

相关任务

第 169 页的『[安装 Java EE 数据库记录器](#)』

按照以下指示信息安装并配置 Java EE 数据库记录器以供 Managed File Transfer 使用。

第 160 页的『[安装 IBM MQ Managed File Transfer 独立数据库记录器](#)』

完成以下步骤以安装和配置独立数据库记录器。

无效的查看传输状态请求

通过 Web 网关提交查看文件传输状态的请求时，可能会接收到 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。以下示例显示请求无效传输标识的状态将返回的结果。

1. 该 HTTP 请求提交已截断的传输标识：

```
GET HTTP/1.1 /transfer/414d51204d554e474f2
Host: example.com
User-Agent: mozilla
```

2. Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 400 Bad Request
Server: WAS/6.0
Content-length: 64
Content-type: text/plain

BFGWI0022E: The supplied transfer ID did not have a length of 48 characters.
This is not a valid transfer ID.
```

如果您接收的 HTTP 响应的状态码不是 200，请参阅 [HTTP 响应代码](#) 主题以获取更多信息。

上载文件出现的问题

通过 Web 网关提交上载文件的请求时，可能会接收到 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。以下示例显示请求上载文件时可能导致发生错误的一些原因。

未能指定 MQMD 用户标识

如果使用 Web 网关请求上载文件，但没有定义 WebSphere MQ 消息描述符 (MQMD) 用户标识，那么传输将失败并返回 HTTP 响应代码 403。有关 Web 网关返回的 HTTP 响应代码的更多信息，请参阅主题 [第 410 页的『HTTP 响应代码』](#)。如果已为托管 Web 网关的应用程序服务器启用跟踪，那么以下信息会写入跟踪文件中：

```
BFGWI0056E: User fte-user is not permitted to access the system due to an MQMD
user identifier not being available.
```

在本示例中，*fte-user* 是提交文件上载请求的用户。有关在应用程序服务器中配置跟踪的指示信息，请参阅 [第 414 页的『启用 Web 网关跟踪』](#)。

要通过 Web 网关成功提交文件传输请求，必须定义传输要使用的 MQMD 用户标识。可以为每个用户定义特定的 MQMD 用户标识，也可以定义缺省的 MQMD 用户标识。

要定义 Web 用户标识和 MQMD 用户标识之间的映射集，请使用 Web 网关管理 API。有关更多详细信息，请参阅主题 [第 338 页的『示例：将 Web 用户标识映射到 MQMD 用户标识』](#) 和 [第 942 页的『用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式』](#)。如果未定义 MQMD 用户标识的用户提交了文件上载请求，那么将使用 **defaultMQMDUserID** 参数的值。有关设置该参数的指示信息，请参阅主题 [第 183 页的『准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关』](#) 和 [第 198 页的『在 WebSphere 应用程序服务器 V7.0 中部署 Web 网关』](#)。

未能指定目标代理

1. 该 HTTP 请求在未指定目标代理的情况下提交上载文件的请求：

```
POST HTTP/1.1 /file/agent/
Host: example.com
User-Agent: mozilla
Content-Type: multi-part/form-data; boundary=Aa6b74
x-fte-checksum: MD5

--Aa6b74
Content-Disposition: form-data; name="files"; filename="myfile.txt"
Content-Type: text/plain
```

```
Account No, Balance
123456, 100.00
234567, 1022.00
345678, 2801.00
456789, 16.75
--Aa6b74
```

2. Web 网关将返回以下格式的 HTTP 响应:

```
HTTP/1.1 400 Bad Request
Server: WAS/6.0
Content-length: 62
Content-type: text/plain

BFGWI0002E: URI is incomplete: missing destination agent name.
```

要使请求生效, 请在请求的 URI 中指定目标代理名称, 如以下示例所示:

```
1. POST HTTP/1.1 /file/agent/ACCOUNTS
Host: example.com
User-Agent: mozilla
Content-Type: multi-part/form-data; boundary=Aa6b74
x-fte-checksum: MD5

--Aa6b74
Content-Disposition: form-data; name="files"; filename="myfile.txt"
Content-Type: text/plain

Account No, Balance
123456, 100.00
234567, 1022.00
345678, 2801.00
456789, 16.75
--Aa6b74
```

如果您接收的 HTTP 响应的状态码不是 200, 请参阅 [HTTP 响应代码](#) 主题以获取更多信息。

尝试在不具有必需权限的情况下创建文件空间

要通过 IBM MQ Managed File Transfer Web 网关来创建文件空间, 您的用户标识必须与相应的 MQMFT 安全角色相关联。如果您尝试在不具有适当权限的情况下创建文件空间, 那么将收到 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。以下示例显示了某个不具有适当权限的用户尝试创建文件空间。

1. 此 HTTP 请求将按照所需格式来创建文件空间。提交该请求的用户 jill 是组 employees 的成员。在用于托管 Web 网关的应用程序服务器环境中定义了 employees 组。组 employees 未与 wmqfte-filespace-create 角色或 wmqfte-admin 角色相关联。用户 jill 尝试创建名为 kevin 的文件空间, 以便用户 jill 和 lakshmi 可向其中传输文件。

```
POST HTTP/1.1 /admin/filespace/kevin
Host: example.com
User-Agent: mozilla
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<filespaces>
  <filespace>
    <quota bytes="1048576"/>
    <writers>
      <authorized action="add">
        <agent-user>jill</agent-user>
        <agent-user>lakshmi</agent-user>
      </authorized>
      <unauthorized action="add">
        <agent-user>mary</agent-user>
      </unauthorized>
    </writers>
  </filespace>
</filespaces>
```

2. Web 网关将返回以下格式的 HTTP 响应:

```
HTTP/1.1 401 Unauthorized
Server: Apache-Coyote/1.1
Content-Type: text/plain; charset=ISO-8859-1
```

```
BFGWIO014E: User not authorized to perform the request.
```

要使该请求生效，必须将用户 `jill` 添加到与 MQMFT 角色 `wmqfte-admin` 或 `wmqfte-filespace-create` 相关联的应用程序服务器组中。Web 网关随附的示例部署计划显示了 WebSphere Application Server Community Edition 的样本安全配置。该计划将 `wmqfte-admin` 角色与 `administrators` 组相关联，并将 `wmqfte-filespace-create` 角色与 `managers` 和 `administrators` 组相关联。用户 `jill` 不属于以上任何组，因此其无法创建文件空间。

有关在应用程序服务器中配置安全许可权的更多信息，请参阅主题第 183 页的『准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关』和第 198 页的『在 WebSphere 应用程序服务器 V7.0 中部署 Web 网关』。

有关 Web 网关管理 API 所返回的错误代码的更多信息，请参阅 [HTTP 响应代码](#) 主题。

相关概念

第 106 页的『保证 Web 网关的安全』

可以使用一些方法来保护 Web 网关的安全。使用 Web 网关之前，必须先执行某些安全步骤。其他步骤可选，可增加 Web 网关和 IBM MQ Managed File Transfer 网络的安全性，但使用 Web 网关并非必须执行这些步骤。

相关参考

第 108 页的『Web 网关的用户角色』

IBM MQ Managed File Transfer 已定义多个不同的角色，用于控制用户可执行的操作。

尝试创建已存在的文件空间

通过 IBM MQ Managed File Transfer Web 网关创建的文件空间必须具有唯一名称。如果尝试用已被使用的名称来创建文件空间，那么将被视为尝试修改该文件空间。如果您无权修改该文件空间，那么将收到 HTTP 错误代码和 IBM MQ Managed File Transfer 错误消息。

1. 此 HTTP 请求将提交一个请求来创建名为 `murray` 的文件空间。在此示例中，文件空间 `murray` 已存在，并且提交该请求的用户无权修改该文件空间。

```
POST HTTP/1.1 /admin/filespace/murray
Host: example.com
User-Agent: mozilla
Content-Type: application/xml
Content-Length: 266

<?xml version="1.0" encoding="UTF-8"?>
<filespaces>
  <filespace>
    <quota bytes="1048576"/>
    <writers>
      <authorized>
        <agent-user>neerav</agent-user>
        <agent-user>SYS.ADMIN.*</agent-user>
      </authorized>
      <unauthorized>
        <agent-user>olivia</agent-user>
      </unauthorized>
    </writers>
  </filespace>
</filespaces>
```

2. Web 网关将返回以下格式的 HTTP 响应：

```
HTTP/1.1 400 Bad Request
Server: Apache-Coyote/1.1
Content-Type: text/plain; charset=ISO-8859-1
```

```
BFGWIO014E: User not authorized to perform the request.
```

要使该请求生效，请指定未被使用的文件空间名称。有关列出 IBM MQ Managed File Transfer 环境中文件空间的信息，请参阅主题第 333 页的『示例：列出所有文件空间』和 第 930 页的『Web 网关管理 API 引用』。

有关 Web 网关管理 API 所返回的错误代码的更多信息，请参阅 [HTTP 响应代码](#) 主题。

Web 代理无法启动

如果您收到来自 **fteStartAgent** 命令的错误，并且您正在尝试启动 Web 代理，请检查 `SYSTEM.FTE.WEB.gateway_name` 队列是否存在。

示例错误

在您运行 **fteCreateWebAgent** 命令时，将创建多个 WebSphere MQ 队列。通过 Web 代理运行 **fteStartAgent** 命令时，只有当这些队列都存在时才可以启动该代理。如果这些队列之一缺失，那么代理将无法启动，并且会将一条消息写入代理日志：

```
The agent received MQI reason code 2085 when opening queue 'SYSTEM.FTE.WEB.WG1_GTWY' on local queue manager 'QM1'.
The agent cannot continue and will end.
00000001 AgentRuntime E BFGAG0061E: The agent ended abnormally
```

如果看到该错误，请检查 `SYSTEM.FTE.WEB.RESP.agent_name` 和 `SYSTEM.FTE.WEB.gateway_name` 队列是否存在。`SYSTEM.FTE.WEB.gateway_name` 队列在与该 Web 网关关联的所有 Web 代理之间共享，因此，在您运行 **fteDeleteAgent** 命令时，如果另一个 Web 代理仍在运行，将不会删除该队列。用户必须手动删除该队列，因此另一个 Web 网关用户可能已删除该队列，而未认识到已创建另一个 Web 代理。

将文件发送到文件空间时超时

将文件从源代理程序发送到目标文件空间时，可能会看到返回码 58 和以下消息：BFGFS0008E: Failed to look up a file space '*file_space_name*' for user '*user_name*' due to a timeout. 仅当 Web 网关部署在 WebSphere Application Server Version 7.0 上时，才会发生此问题。

应用程序服务器中未选中支持分布式两阶段落实协议时可能会导致该问题。要启用该行为，请执行以下步骤：

1. 从 WebSphere Application Server V7.0 管理控制台导航中选择资源 > **JMS** > 队列连接工厂。
2. 在队列连接工厂面板上，选择名为 `jms/WMQFTEWebAgentConnectionFactory` 的资源。
3. 在高级部分，确保选中支持分布式两阶段落实协议复选框。

由于编码问题，请求失败

如果 WebSphere 应用程序服务器 V7.0 在缺省编码不是 UTF-8 或缺省编码无法映射到 UTF-8（例如，cp1252）的机器上运行，那么 Web 网关无法完成请求。

关于此任务

请求失败，并返回以下错误：

```
BFGWI0018E:(WEBGATEWAY) The request could not be completed due to an internal web application server error. Caused by: Invalid byte 2 of 4-byte UTF-8 sequence.
```

要解决此问题，请通过完成以下步骤在 JVM 上设置 Java file.encoding 系统属性：

过程

1. 打开 WebSphere 应用程序服务器 管理控制台并浏览至：应用程序服务器 > **Web 网关所在服务器的名称** > 进程定义 > **Java 虚拟机**。
2. 将以下自变量添加到通用 **JVM 自变量**：

-Dfile.encoding=UTF8

3. 关闭并重新启动 WebSphere 应用程序服务器 以刷新配置。

HTTP 响应代码

向 IBM MQ Managed File Transfer Web 网关发出的请求的 HTTP 响应中会返回状态码。

Web 网关返回的响应头中包含 HTTP 响应代码。以下示例中的 HTTP 头包含 HTTP 响应代码 200 OK:

```
HTTP/1.1 200 OK
Server: WAS/6.0
Content-length: 0
```

下表描述了 HTTP 响应代码的可能值，以及可返回的关联 IBM MQ Managed File Transfer 错误代码的示例。有关 IBM MQ Managed File Transfer 错误代码的更多信息，请参阅[诊断消息](#)。

表 28: HTTP 响应代码		
HTTP 响应代码	IBM MQ Managed File Transfer 错误代码示例	示例描述
200 OK	None	已正确处理有效请求，且已选择性地向用户提供响应。
202 Accepted	None	已正确处理有效请求，但 IBM MQ Managed File Transfer 不保证请求的操作已完成。 例如，文件上载传输请求已处理并提交到 IBM MQ Managed File Transfer 代理，但尚未进行传输。
400 Bad Request	BFGWI0001	由于 URI 缺少资源类型，因此该 URI 无效。
403 Forbidden	BFGWI0056	没有为该用户定义 IBM MQ 消息描述符 (MQMD) 用户标识。
404 Not Found	BFGWI0015	找不到请求的资源。
405 Method Not Allowed	BFGWI0016	请求的资源不支持请求中所使用的 HTTP 谓词。 例如，对仅支持 POST 或 DELETE 的资源使用了 GET 操作。
410 Resource Gone	BFGWI0031	所请求的资源不再可用。例如，请求的文件已从文件空间中删除。
413 Request Entity Too Large	BFGWI0026	请求包含的某个文件过大，服务器无法处理。
415 Unsupported Media Type	BFGWI0017	接收的请求具有 Content-type HTTP 头指定的介质类型，但该类型不受支持。
500 Internal Server Error	BFGWI0018	处理请求时迁到内部错误。已生成 FFDC 或 ABEND 文件。
502 Bad Gateway	BFGWI0019	由于 IBM MQ Managed File Transfer 外部发生错误，无法完成请求。例如，IBM MQ 队列管理器不可用。

表 28: HTTP 响应代码 (继续)		
HTTP 响应代码	IBM MQ Managed File Transfer 错误代码示例	示例描述
503 Service Unavailable	BFGWI0020	该目标暂时不可用。例如，IBM MQ 队列已满。
504 Gateway Timeout	BFGWI0021	由于超出 IBM MQ Managed File Transfer 或 HTTP 客户机设定的时间限制，尝试完成请求超时。

相关概念

第 412 页的『[对 Web 网关进行故障诊断](#)』

使用以下参考信息和示例可帮助您诊断 Web 网关返回的错误。

第 305 页的『[Web 网关的方案](#)』

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

第 307 页的『[Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑](#)』

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

第 312 页的『[使用 IBM MQ Managed File Transfer 服务 Web 网关](#)』

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

第 313 页的『[示例 HTTP 流](#)』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

相关参考

第 908 页的『[Web 网关 API 引用](#)』

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

第 909 页的『[使用 Web 网关时的 HTTP 头和 HTML 表单字段](#)』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

第 911 页的『[使用 Web 网关时的统一资源标识语法](#)』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 918 页的『[使用 Web 网关的内容类型](#)』

向 IBM MQ Managed File Transfer Web 网关提交的文件传输请求必须对应于特定的介质类型。来自 Web 网关的响应具有介质类型 application/xml 或 application/json。

第 919 页的『[响应格式：XML 和 JSON](#)』

IBM MQ Managed File Transfer Web 网关采用以下两种格式之一返回查询的响应：XML 或 JavaScript Object Notation (JSON)。

第 930 页的『[Web 网关管理 API 引用](#)』

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

对 Connect:Direct 网桥进行故障诊断

使用以下参考信息和示例，以帮助诊断 Connect:Direct 网桥返回的错误。

- 第 424 页的『[跟踪 Connect:Direct 网桥](#)』
- 第 424 页的『[Connect:Direct 网桥的日志信息](#)』
- 第 425 页的『[解决 Connect:Direct 节点的许可权问题](#)』

- [第 425 页的『如果与 Connect:Direct 节点之间的文本传输未正确转换数据，该如何操作』](#)
- [第 426 页的『如果通过 Connect:Direct 网桥到 PDS 或 PDS 成员的传输失败，该如何操作』](#)
- [第 426 页的『使用双正斜杠指定 Connect:Direct 文件路径』](#)
- [第 426 页的『增加 Connect:Direct 网桥的并行传输数』](#)
- [第 427 页的『调试文件传输调用的 Connect:Direct 进程』](#)

跟踪 Connect:Direct 网桥

您可以从作为 Connect:Direct 网桥组成部分的 Connect:Direct 节点来捕获跟踪，以帮助确定问题。

关于此任务

要启用跟踪，请完成以下步骤：

过程

1. 停止 Connect:Direct 网桥代理。
2. 编辑 Connect:Direct 网桥代理属性文件，以包含以下行：

```
cdTrace=true
```

3. 启动 Connect:Direct 网桥代理。

结果

跟踪信息将写入 Connect:Direct 网桥代理配置目录中的 output0.log 文件内。

相关参考

[第 591 页的『agent.properties 文件』](#)

每个代理都有自己的属性文件 agent.properties，其中必须包含代理用于连接到其队列管理器的信息。agent.properties 文件还可以包含用于更改代理程序行为的属性。

Connect:Direct 网桥的日志信息

您可以使用 Connect:Direct 网桥代理，在 MQMFT 代理和 Connect:Direct 节点之间传输文件。有关这些传输中涉及的 Connect:Direct 节点和进程的日志信息都显示在 IBM MQ Explorer 插件中，并存储在您的日志数据库内。

Connect:Direct 网桥代理必须是 WebSphere MQ File Transfer Edition V7.0.4 或更高版本。该传输中涉及的其他代理可以是任意版本的 IBM MQ Managed File Transfer。但是，要记录有关 Connect:Direct 节点和进程的信息，传输中涉及的所有 MQMFT 代理都必须是 V7.0.4 或更高版本。要在 IBM MQ Explorer 插件中显示该信息，该插件必须为 V7.0.4 或更高版本。要在日志数据库中存储该信息，数据库记录器和数据库模式必须是 V7.0.4 或更高版本。

有关文件传输中涉及的 Connect:Direct 节点和 Connect:Direct 进程的日志信息，包含在发布到协调队列管理器上的 SYSTEM.FTE 主题的日志消息中。有关更多信息，请参阅 [第 662 页的『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 网桥节点名称的值是 MQMFTConnect: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

第 683 页的『Connect:Direct 网桥传输消息示例』

当目标代理或源代理为 Connect:Direct 网桥代理时，destinationAgent 或 sourceAgent 元素会包含更多的属性。已开始日志消息只包含一部分有关 Connect:Direct 传输的信息。进行中和已完成日志消息包含有关 Connect:Direct 传输的完整信息。

解决 Connect:Direct 节点的许可权问题

如果 IBM MQ Managed File Transfer 和 Connect:Direct 之间的传输失败，并且出现关于没有足够许可权的错误，请使用本主题中的信息。

对于涉及 Connect:Direct 网桥的传输，连接到 Connect:Direct 节点的用户标识由与传输请求相关联的 WebSphere 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，请参阅第 207 页的『映射 Connect:Direct 的凭证』。

您可能看到传输失败，并且出现以下某错误：

- BFGCD0001E: This task was rejected by the Connect:Direct API with the following error message: Connect:Direct Node detected error.
LCCA000I The user has no functional authority to issue the selp command
- BFGCD0026I: Connect:Direct messages: The submit of the process succeeded. Process number 1092 (name F35079AE, SNODE MYNODE) executing. User fteuser does not have permission to override SNODEID. User fteuser does not have permission to override SNODEID. User fteuser does not have permission to override SNODEID.

如果看到以上任一错误，请确定与传输请求中使用的 MQMD 用户标识相关联的 Connect:Direct 用户标识。该 Connect:Direct 用户标识必须具有执行 Connect:Direct 网桥要求的 Connect:Direct 操作的权限。有关所需的功能权限列表，以及有关如何授予这些权限的指导，请参阅第 207 页的『通过使用 ConnectDirectCredentials.xml 文件映射 Connect:Direct 的凭证』。

如果与 Connect:Direct 节点之间的文本传输未正确转换数据，该如何操作

当以文本方式在 MQMFT 代理和 Connect:Direct 节点之间传输文件时，会执行代码页转换和行尾字符转换。该传输使用 Connect:Direct 网桥节点的网络映射中的操作系统信息来确定远程节点的行尾字符。如果网络映射中的信息不正确，那么行尾字符转换可能未正确执行。

确保 Connect:Direct 网桥节点和用作传输目标的任意 Connect:Direct 节点的网络映射包含正确的平台描述。

- 如果 Connect:Direct 网桥节点在 Windows 系统上，那么请确保针对网络映射中的每个远程节点，从**操作系统**列表中选择正确的值。
 - 如果远程节点在 Windows 系统上，那们请选择 Windows。
 - 如果远程节点位于 UNIX 或 Linux 系统上，请选择 UNIX。
 - 如果远程节点在 z/OS 系统上，那们请选择 OS/390。

Connect:Direct 网桥不支持到其他操作系统上的远程节点的传输。

- 对于要进行文件传输的每个远程节点，确保在 Connect:Direct 网桥代理配置目录中的 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，请参阅第 206 页的『配置 ConnectDirectNodeProperties.xml 文件以包含有关远程 Connect:Direct 节点的信息』和 第 621 页的『Connect:Direct 节点属性文件格式』。

相关参考

第 715 页的『在 Connect:Direct 和 IBM MQ Managed File Transfer 之间传输文本文件』

文本传输涉及从一个代码页到另一个代码页的文件代码页转换。文本文件传输还涉及系统之间的 CRLF（回车换行）字符转换。本主题总结了 MQMFT 代理和 Connect:Direct 节点之间的传输中的文本文件传输行为。

如果通过 Connect:Direct 网桥到 PDS 或 PDS 成员的传输失败，该如何操作

如果传输的目标是 z/OS 上的 Connect:Direct 节点，并且是 PDS 或 PDS 成员，那么如果未使用覆盖值指定 **-de** 参数，那么传输将失败。

关于此任务

如果使用 **fteCreateTransfer** 或 **fteCreateTemplate** 命令提交了传输，请执行以下步骤：

过程

1. 更改提交的命令，以包含 **-de overwrite**。
2. 再次提交命令。

使用 IBM MQ Explorer 插件

关于此任务

如果使用 IBM MQ Explorer 插件提交了传输，请执行以下步骤：

过程

1. 在“创建新的受管文件传输”向导中指定源和目标信息。
2. 选择覆盖目标文件系统中具有相同名称的文件。
3. 再次提交命令。

使用双正斜杠指定 Connect:Direct 文件路径

在文件传输过程中，如果使用以双正斜杠 (//) 开始的文件路径指定位于 Connect:Direct 节点上的文件，那么会将该文件视为数据集。

Connect:Direct 节点上的源和目标都以格式 *cd_node_name:file_path* 指定。如果 *file_path* 以双正斜杠 (//) 开始，那么该源或目标将被视为数据集。即使 Connect:Direct 节点未在 z/OS 上，也是如此。如果意外将文件路径指定为以双正斜杠 (//) 开始并且文件并非数据集，那么会导致传输失败。

如果不希望指定的文件被视为数据集，那么请勿指定以双正斜杠 (//) 开始的 *file_path*。

相关概念

第 423 页的『对 Connect:Direct 网桥进行故障诊断』

使用以下参考信息和示例，以帮助您诊断 Connect:Direct 网桥返回的错误。

相关参考

第 706 页的『与 Connect:Direct 节点之间传输数据集』

可以使用 Connect:Direct 网桥在 IBM MQ Managed File Transfer 代理和 IBM Sterling Connect:Direct 节点之间传输数据集。您可以将数据集指定为传输源和/或传输目标。

增加 Connect:Direct 网桥的并行传输数

要增加 Connect:Direct 网桥代理可以处理的并行传输数，必须更改三个代理属性。您还必须增加 Connect:Direct 节点接受的最大连接数。

Connect:Direct 网桥代理可处理的最大并行传输数取决于某些代理属性的值。**maxSourceTransfers** 和 **maxDestinationTransfers** 代理属性针对 Connect:Direct 网桥代理的传输数缺省值为 5。该缺省值低于针对其他类型的代理的传输数缺省值 (25)。配置代理为 **maxSourceTransfers** 和 **maxDestinationTransfers** 的缺省值的 Connect:Direct 网桥，在任何时候一次最多可处理 10 个传输：代理作为源的 5 个代理和代理作为目标的 5 个传输。

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。Connect:Direct 节点在 UNIX 上接受的最大连接数由 **api.max.connects** Connect:Direct 参数控制。对于 Windows 上的 Connect:Direct 节点，等效参数为 **max.api.connects**。

如果 Connect:Direct 网桥执行大量文件传输的速率不够，那么可以增加 Connect:Direct 网桥代理处理的并行传输数。针对 Connect:Direct 网桥代理更改以下代理属性：

maxSourceTransfers

将该属性设置为大于 5 且小于等于 25 的值。如果选择大于 25 的值，那么除非增加可用于代理使用的 JVM 的内存量，否则该代理将耗尽内存。

maxDestinationTransfers

将该属性设置为大于 5 且小于等于 25 的值。如果选择大于 25 的值，那么除非增加可用于代理使用的 JVM 的内存量，否则该代理将耗尽内存。

ioThreadPoolSize

ioThreadPoolSize 的缺省值为 10。该属性限制 Connect:Direct 网桥代理充当源代理的传输的 Connect:Direct 节点 API 连接数。这些传输从 Connect:Direct 到 IBM MQ Managed File Transfer。使用以下指导信息来设置该属性的值：

- 如果 **maxSourceTransfers** 的值小于 **maxDestinationTransfers** 的值，请将 **ioThreadPoolSize** 设置为 **maxSourceTransfers** 值的 2 倍或 10（两者中选用较大值）
- 如果 **maxSourceTransfers** 的值大于 **maxDestinationTransfers** 的值，请将 **ioThreadPoolSize** 设置为 **maxSourceTransfers** 和 **maxDestinationTransfers** 的总和

除这些代理属性外，还必须更改作为 Connect:Direct 网桥的一部分的 Connect:Direct 节点的最大并行 API 连接数。控制该数值的 Connect:Direct 参数为：如果节点在 UNIX 上，那么为 **api.max.connects**；如果节点在 Windows 上，那么为 **max.api.connects**。对相应的参数执行以下更改：

api.max.connects（如果 Connect:Direct 网桥中的节点在 UNIX 上）

将此参数设置为大于 **maxSourceTransfers** 和 **maxDestinationTransfers** 之和的值。

api.max.connects 参数的缺省值为 16。有关如何设置该参数的更多信息，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。有关如何设置该参数的更多信息，请参阅 Connect:Direct 文档。

相关任务

[第 204 页的『配置 Connect:Direct 网桥』](#)

配置 Connect:Direct 网桥，以在 IBM MQ Managed File Transfer 网络和 Connect:Direct 网络之间传输文件。Connect:Direct 网桥的组件包括 Connect:Direct 节点和专用于与该节点通信的 IBM MQ Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

相关参考

[第 591 页的『agent.properties 文件』](#)

每个代理都有自己的属性文件 **agent.properties**，其中必须包含代理用于连接到其队列管理器的信息。**agent.properties** 文件还可以包含用于更改代理程序行为的属性。

调试文件传输调用的 Connect:Direct 进程

您可以配置 Connect:Direct 网桥代理，以将有关由文件传输调用的 Connect:Direct 进程的日志信息写入 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** 文件以包含属性 **logCDProcess**。

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

有关 Connect:Direct 进程的信息将记录到 Connect:Direct 网桥代理的 output0.log 文件中。记录的信息包括:

- MQMFT 传输标识
- Connect:Direct 进程名称
- Connect:Direct 进程号
- 生成的进程定义
- 如果 Connect:Direct 进程是用户定义的, 那么还包含进程模板的文件名

相关概念

[第 423 页的『对 Connect:Direct 网桥进行故障诊断』](#)

使用以下参考信息和示例, 以帮助您诊断 Connect:Direct 网桥返回的错误。

相关参考

[第 591 页的『agent.properties 文件』](#)

每个代理都有自己的属性文件 agent.properties, 其中必须包含代理用于连接到其队列管理器的信息。agent.properties 文件还可以包含用于更改代理程序行为的属性。

参考

产品概述

IBM MQ Managed File Transfer 如何工作?

IBM MQ Managed File Transfer 可以使用多种方式与 IBM MQ 进行交互。该主题描述了两个产品是如何进行交互的。

- IBM MQ Managed File Transfer 通过将每个文件分割为一条或多条消息并通过 IBM MQ 网络传输这些消息, 从而在代理进程之间传输文件。
- 代理进程通过使用非持久消息来移动文件数据, 以在最大程度上降低对 IBM MQ 日志的影响。通过彼此通信, 代理进程可控制好传输包含文件数据的消息流。这可防止包含文件数据的消息在 IBM MQ 传输队列上聚集, 并确保当有任何非持久消息未交付时会重新发送文件数据。
- IBM MQ Managed File Transfer 代理会使用多个 IBM MQ 队列。有关更多信息, 请参阅[系统队列和系统主题](#)。
- 虽然三个队列中的部分队列严格限制为内部使用, 但是代理可接受形式为特殊格式的命令消息的请求, 这些消息发送至特定队列以供该代理从中读取。命令行命令和 IBM MQ Explorer 插件均可将 IBM MQ 消息发送至代理以指示代理执行所需的操作。您可以使用此方式来编写用于与代理进行交互的 IBM MQ 应用程序。有关更多信息, 请参阅[第 370 页的『通过将消息放置在代理命令队列中来控制 IBM MQ Managed File Transfer』](#)。
- IBM MQ Managed File Transfer 代理会将有关其状态以及传输进度和结果的信息发送至已被指定为协调队列管理器的 IBM MQ 队列管理器。此信息由协调队列管理器发布, 可供希望监视传输进度或保留发生的传输记录的应用程序来预订。命令行命令和 IBM MQ Explorer 插件均可使用已发布的信息。您可编写使用此信息的 IBM MQ 应用程序。有关信息发布到的主题的更多信息, 请参阅[第 646 页的『SYSTEM.FTE 主题』](#)。
- IBM MQ Managed File Transfer 的关键组件将会利用 IBM MQ 队列管理器的功能来存储和转发消息。这意味着如果您遇到停机, 那么基础结构中未受影响的部分仍可继续传输文件。这可扩展至协调队列管理器,

在此情况下，存储转发和持久预订的组合允许协调队列管理器承受变为不可用，而不会丢失有关发生的文件传输的关键信息。

如何读轨道（语法）图

每个轨道图均以一个双右箭头开始，以一对左右箭头结束。以一个单右箭头开始的行是连续行。按照箭头的方向，以从左到右、从上到下的方式阅读轨道图。

轨道图中使用的其他约定包括：

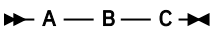
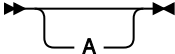
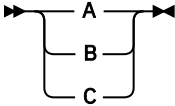
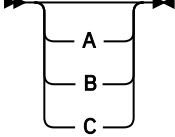
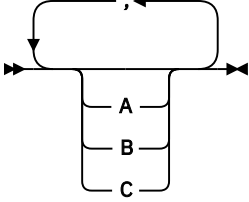
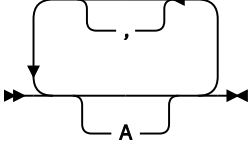
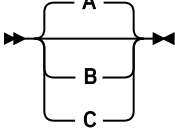
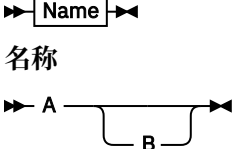
表 29: 如何阅读轨道图	
约定	含义
	必须指定值 A， B 和 C。必需值显示在铁路图的主线上。
	您可以指定值 A。可选值显示在铁路图的主线下方。
	值 A、B 和 C 是备选项，必须指定其中一项。
	值 A、B 和 C 是备选项，可以指定其中一项。
	您可以指定一个或多个值 A， B 和 C。多个或重复值的任何必需分隔符 (在此示例中，逗号 (，)) 显示在箭头上。
	您可以多次指定值 A。本示例中的分隔符是可选的。
	值 A、B 和 C 是备选项，可以指定其中一项。如果不指定显示的任何值，将使用缺省值 A（主线之上显示的值）。
 名称	轨道图片段 Name 是与主轨道图分开显示的。

表 29: 如何阅读轨道图 (继续)	
约定	含义
标点符号和大写值	请完全按照显示的内容进行指定。
小写值 (例如, <i>name</i>)	请提供您自己的文本代替 <i>name</i> 变量。

安装

IBM MQ Managed File Transfer 硬件和软件先决条件

在您安装 IBM MQ Managed File Transfer 之前, 请检查您的系统是否满足产品的硬件和软件需求。对于所有平台, 在您的 IBM MQ Managed File Transfer 网络中, 必须有一个 WebSphere MQ V7.0 或更高版本的队列管理器可供用作协调队列管理器。

请参阅 [WebSphere MQ 系统需求](#) 以了解必备硬件和软件。

已安装的命令集

下表显示了随每个组件一起安装的命令。

表 30: 每个命令集中可用的 IBM MQ Managed File Transfer 命令				
命令	代理程序 命令集	服务 命令集	Tools 命令集	记录器命令集
fteAnt			X	
fteBundleConfiguration			X (仅限分布式平台)	
fteCancelTransfer			X	
fteChangeDefaultConfigurationOptions	X	X	X	X
fteCleanAgent	X	X		
fteCreateAgent	X	X		
fteCreateBridgeAgent		X		
fteCreateCDAgent	X (仅限分布式平台)	X (仅限分布式平台)		
fteCreateLogger				X
fteCreateMonitor			X	
fteCreateTemplate			X	
fteCreateTransfer			X	
fteCreateWebAgent		X (仅限分布式平台)		
fteDefine			X (仅限分布式平台)	
fteDelete			X (仅限分布式平台)	
fteDeleteAgent	X	X		
fteDeleteLogger				X
fteDeleteMonitor			X	
fteDeleteScheduledTransfer			X	
fteDeleteTemplates			X	
fteDisplayVersion	X	X		X
fteListAgents	X	X	X	X
fteListMonitors			X	

表 30: 每个命令集中可用的 IBM MQ Managed File Transfer 命令 (继续)				
命令	代理程序 命令集	服务 命令集	Tools 命令集	记录器命令集
fteListScheduledTransfers			X	
fteListTemplates			X	
fteMigrateAgent	X	X		
fteMigrateConfigurationOptions	X	X	X	X
fteMigrateLogger				X
fteModifyAgent	X (仅 Windows)	X (仅 Windows)		
fteModifyLogger				X (仅 Windows)
fteObfuscate	X	X		X
ftePingAgent			X	
fteSetAgentTraceLevel	X	X		
fteSetLoggerTraceLevel				X
fteSetupCommands	X	X	X	X
fteSetupCoordination	X	X	X	X
fteShowAgentDetails	X	X	X	X
fteShowLoggerDetails				X
fteStartAgent	X	X		
fteStartLogger				X
fteStopAgent	X	X		
fteStopLogger				X

安全性

IBM MQ 中的 IBM MQ Managed File Transfer 的文件系统许可权

当您安装并配置 IBM MQ 的 IBM MQ Managed File Transfer 组件时，将使用以下许可权来创建 configuration、installations 和 logs 目录。

UNIX 和 Linux

表 31: UNIX 和 Linux 上的目录许可权摘要	
目录	许可权
/var/mqm/mqft/config	- 可由 mqm 组写入 - 全局可读 mqm 组中的用户具有对这些目录和文件的写访问权
/var/mqm/mqft/installations	- 可由 mqm 组写入 - 全局可读
/var/mqm/mqft/logs	全局可读写

Windows

表 32: Windows 上目录的许可权摘要	
目录	许可权
<code>MQ_DATA_PATH\mqft\config</code>	以下用户具有完全的读写访问权： • 管理员 • 系统帐户 • mqm 组 其他用户具有读访问权
<code>MQ_DATA_PATH\mqft\installations</code>	以下用户具有完全的读写访问权： • 管理员 • 系统帐户 • mqm 组 其他用户具有读访问权
<code>MQ_DATA_PATH\mqft\logs</code>	以下用户具有完全的读写访问权： • 管理员 • 系统帐户 • mqm 组 其他用户具有读写访问权

z/OS

表 33: z/OS 上目录的许可权摘要	
目录	许可权
<code>DATA_PATH/mqft/config</code>	• 可由 mqm 组或者环境变量 BFG_GROUP_NAME 中确定的组名写入 • 全局可读 mqm 组中的用户或者环境变量 BFG_GROUP_NAME 中的值具有对这些目录和文件的写访问权
<code>DATA_PATH/mqft/installations</code>	• 可由 mqm 组或者环境变量 BFG_GROUP_NAME 中确定的组名写入 • 全局可读
<code>DATA_PATH/mqft/logs</code>	全局可读写

注: `DATA_PATH` 从环境变量 `BFG_DATA` 派生。

特定于 IBM MQ Managed File Transfer 的资源权限

针对任何文件传输请求，代理进程都需要其本地文件系统的某个访问权级别。此外，与代理进程关联的用户标识以及与执行文件传输操作的用户关联的用户标识都必须有权使用某些 IBM MQ 对象。

由可能具有操作角色的用户发出命令，这些用户通常会启动文件传输。另外，他们可能具有管理角色，还可以控制何时创建、启动、删除或清除代理（即何时从所有代理系统队列中除去消息）。用户发出命令时，会

将包含命令请求的消息放置在代理的 SYSTEM.FTE.COMMAND 队列中。代理进程会从 SYSTEM.FTE.COMMAND 队列中检索包含命令请求的消息。代理进程还会使用以下四个其他系统队列：

- SYSTEM.FTE.DATA.agent_name
- SYSTEM.FTE.EVENT.agent_name
- SYSTEM.FTE.REPLY.agent_name
- SYSTEM.FTE.STATE.agent_name

如果代理是 Web 代理，那么将具有两个额外的队列。这些队列具有以下名称：

- SYSTEM.FTE.WEB.RESP.agent_name
- SYSTEM.FTE.WEB.网关名称

由于发出命令的用户使用上述队列的方式不同于代理进程，所以会向与每个队列关联的用户标识或用户组分配不同的 IBM MQ 权限。请参阅第 433 页的『特定于 IBM MQ Managed File Transfer 的资源组权限』，以了解更多信息。

代理具有额外的队列，可用来授予用户执行某些操作的权限。请参阅第 438 页的『用户的 IBM MQ Managed File Transfer 操作权限』，以获取有关如何使用权限队列的信息。代理不会在这些队列上放置消息，也不会从这些队列中获取消息。但是，对于用于运行代理进程的用户标识以及有权执行某些操作的用户关联的用户标识，必须确保队列为这些用户标识分配正确的 IBM MQ 权限。权限队列如下：

- SYSTEM.FTE.AUTHADM1.agent_name
- SYSTEM.FTE.AUTHAGT1.agent_name
- SYSTEM.FTE.AUTHMON1.agent_name
- SYSTEM.FTE.AUTHOPS1.agent_name
- SYSTEM.FTE.AUTHSCH1.agent_name
- SYSTEM.FTE.AUTHTRN1.agent_name

如果要从低于 V7.0.2 的 WebSphere MQ File Transfer Edition 版本迁移至 WebSphere MQ V7.5 或更高版本，并要保留现有代理配置，那么将需要手动创建权限队列。使用以下 MQSC 命令创建队列：

```
DEFINE QLOCAL(authority_queue_name) DEFPRTY(0) DEFSOPT(SHARED) GET(ENABLED) MAXDEPTH(0) +  
MAXMSGL(0) MSGDLVSQ(PRIORITY) PUT(ENABLED) RETINTVL(999999999) SHARE NOTRIGGER +  
USAGE(NORMAL) REPLACE
```

代理进程还会使用 SYSTEM.FTE 队列将消息发布到协调队列管理器上的 SYSTEM.FTE 主题。代理进程可能需要权限来读、写、更新和删除文件，这取决于代理进程使用源代理的角色还是目标代理的角色。

您可以使用 IBM MQ Explorer 创建和修改 IBM MQ 对象的权限记录。右键单击对象，然后单击**对象权限 > 管理权限记录**。还可以使用 **setmqaut** 命令创建权限记录，在 **setmqaut**（授予或撤销权限）命令中对该命令进行了描述。

相关参考

第 433 页的『特定于 IBM MQ Managed File Transfer 的资源组权限』

无需针对可能涉及到的所有对象而为个别用户授予权限，而是请配置以下两个安全组以用于管理 IBM MQ Managed File Transfer 访问控制：FTEUSER 和 FTEAGENT。IBM MQ 管理员应负责创建并填充这些组。管理员可选择扩展或修改此处所述的建议配置。

第 438 页的『用户的 IBM MQ Managed File Transfer 操作权限』

除使用组来管理对资源的访问权之外，您还可以启用其他安全级别来限制用户可采取的代理操作。将代理权限队列上的权限授予用户以给予用户执行特定代理操作的许可权。

第 441 页的『记录器的权限』

运行记录器的操作系统用户需要具有对记录器队列和 SYSTEM.FTE 主题的特定 IBM MQ 权限。

特定于 IBM MQ Managed File Transfer 的资源组权限

无需针对可能涉及到的所有对象而为个别用户授予权限，而是请配置以下两个安全组以用于管理 IBM MQ Managed File Transfer 访问控制：FTEUSER 和 FTEAGENT。IBM MQ 管理员应负责创建并填充这些组。管理员可选择扩展或修改此处所述的建议配置。

连接至队列管理器的权限

由操作用户、管理用户和 IBM MQ Explorer 运行的命令需要能够连接到命令队列管理器和协调队列管理器。用于创建、更改或删除代理而运行的代理进程和命令需要能够连接至代理队列管理器。

- 授予 FTEUSER 组针对命令队列管理器和协调队列管理器的连接权限。例如：

对于 **UNIX, Linux 和 Windows 系统:**

```
setmqaut -m command_queue_manager -t qmgr -g FTEUSER +connect
setmqaut -m coordination_queue_manager -t qmgr -g FTEUSER +connect
```

对于 **IBM i:**

```
GRTMQMAUT OBJ('command_queue_manager') OBJTYPE(*MQM) USER(FTEUSER) AUT(*CONNECT)
GRTMQMAUT OBJ('coordination_queue_manager') OBJTYPE(*MQM) USER(FTEUSER) AUT(*CONNECT)
```

对于 **z/OS:**

```
RDEFINE MQCONN command_queue_manager.BATCH UACC(NONE)
PERMIT command_queue_manager.BATCH CLASS(MQCONN) ID(FTEUSER) ACCESS(READ)
RDEFINE MQCONN coordination_queue_manager.BATCH UACC(NONE)
PERMIT coordination_queue_manager.BATCH CLASS(MQCONN) ID(FTEUSER) ACCESS(READ)
```

- 授予 FTEAGENT 组对于代理队列管理器的 connect 和 inquire 权限。例如：

对于 **UNIX, Linux 和 Windows 系统:**

```
setmqaut -m agent_queue_manager -t qmgr -g FTEAGENT +connect +inq +setid
```

对于 **IBM i:**

```
GRTMQMAUT OBJ('agent_queue_manager') OBJTYPE(*MQM) USER(FTEAGENT) AUT(*CONNECT)
```

对于 **z/OS:**

```
RDEFINE MQCONN agent_queue_manager.BATCH UACC(NONE)
PERMIT agent_queue_manager.BATCH CLASS(MQCONN) ID(FTEAGENT) ACCESS(READ)
```

有关哪个命令直接连接到哪个队列管理器的信息，请参阅 [第 445 页的『MFT 命令和进程与队列管理器的连接关系』](#)

在属于代理的 COMMAND 队列上放置消息的权限

代理命令队列必须可供有权请求代理执行某一操作的任何用户使用。要满足此需求，

- 请仅授予 FTEUSER 组对 SYSTEM.FTE.COMMAND.agent_name 队列的 put 访问权。例如：

对于 **UNIX, Linux 和 Windows 系统:**

```
setmqaut -m QM1 -n SYSTEM.FTE.COMMAND.agent_name -t queue -g FTEUSER +put
```

对于 **IBM i:**

```
GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEUSER) AUT(*PUT)
MQMNAME('QM1')
```

对于 **z/OS:**

```
RDEFINE MQQUEUE QM1.SYSTEM.FTE.COMMAND.agent_name UACC(NONE)
PERMIT QM1.SYSTEM.FTE.COMMAND.agent_name CLASS(MQQUEUE) ID(FTEUSER) ACCESS(UPDATE)
```

- 请授予 FTEAGENT 组对 SYSTEM.FTE.COMMAND.agent_name 队列的 put、get 和 setid 访问权。例如：

对于 **UNIX, Linux 和 Windows 系统:**

```
setmqaut -m QM1 -n SYSTEM.FTE.COMMAND.agent_name -t queue -g FTEAGENT +browse +put +get
+setid
```


对于 **IBM i**:

```
GRTRMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*PUT)
MQMNAME('QM1')
GRTRMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*GET)
MQMNAME('QM1')
GRTRMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*SETID)
MQMNAME('QM1')
```

对于 **z/OS**:

```
RDEFINE MQQUEUE QM1.SYSTEM.FTE.COMMAND.agent_name UACC(NONE)
PERMIT QM1.SYSTEM.FTE.COMMAND.agent_name CLASS(MQQUEUE) ID(FTEAGENT) ACCESS(UPDATE)
RDEFINE MQADMIN QM1.CONTEXT.SYSTEM.FTE.COMMAND.agent_name UACC(NONE)
PERMIT QM1.CONTEXT.SYSTEM.FTE.COMMAND.agent_name CLASS(MQADMIN) ID(FTEAGENT)
ACCESS(UPDATE)
```

代理需要访问权以将消息放入其他代理的命令队列中。如果存在连接到远程队列管理器的代理，那么可能需要授予额外的权限以允许通道将消息放入此队列中。

在属于代理的 **DATA**、**STATE**、**EVENT** 和 **REPLY** 队列上放置消息的权限

只有 IBM MQ Managed File Transfer 代理需要使用这些系统队列，因此仅为 FTEAGENT 组授予 put、get 和 inquire 访问权限。这些系统队列的名称如下所示：

- DATA - SYSTEM.FTE.DATA.agent_name
- STATE - SYSTEM.FTE.STATE.agent_name
- EVENT - SYSTEM.FTE.EVENT.agent_name
- REPLY - SYSTEM.FTE.REPLY.agent_name

例如，对 SYSTEM.FTE.DATA.agent_name 队列使用如下命令：

对于 **UNIX**，**Linux** 和 **Windows** 系统：

```
setmqaut -m QM1 -n SYSTEM.FTE.DATA.agent_name -t queue -g FTEAGENT +put +get +inq
```

对于 **IBM i**:

```
GRTRMQMAUT OBJ('SYSTEM.FTE.DATA.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*PUT)
MQMNAME('QM1')
GRTRMQMAUT OBJ('SYSTEM.FTE.DATA.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*GET)
MQMNAME('QM1')
```

对于 **z/OS**:

```
RDEFINE MQQUEUE QM1.SYSTEM.FTE.DATA.agent_name UACC(NONE)
PERMIT QM1.SYSTEM.FTE.DATA.agent_name CLASS(MQQUEUE) ID(FTEAGENT) ACCESS(UPDATE)
```

代理需要访问权以将消息放入其他代理的数据和回复队列中。如果存在连接到远程队列管理器的代理，那么可能需要授予额外的权限以允许通道将消息放入这些队列中。

代理过程运行时使用的权限

代理过程运行时使用的权限会影响代理可在文件系统上读取和写入的文件，以及代理可访问的队列和主题。如何配置权限取决于系统。将代理过程运行时使用的用户标识添加到 FTEAGENT 组。有关将用户标识添加到组的更多信息，请参阅[设置安全性并浏览至针对您操作系统的信息](#)。

运行命令和 **IBM MQ Explorer** 时使用的权限

管理命令 (例如，**fteStartAgent** 命令) 和 WebSphere MQ Explorer 的 IBM MQ Managed File Transfer 插件需要能够将消息放入 SYSTEM.FTE.COMMAND.agent_name 队列并从该队列检索已发布的信息。将有权运行命令或 IBM MQ Explorer 的用户标识添加到 FTEUSER 组。发起方用户标识记录在传输日志中。有关将用户标识添加到组的更多信息，请参阅[设置安全性并浏览至针对您操作系统的信息](#)。

在 SYSTEM.FTE 队列和 SYSTEM.FTE 主题上放置消息的权限

仅代理进程才需要在 SYSTEM.FTE 队列和 SYSTEM.FTE 主题上放置消息这一功能。授予 FTEAGENT 组对于 SYSTEM.FTE 队列的 put、get 和 inquire 权限，并授予 FTEAGENT 组对 SYSTEM.FTE 主题的 publish 和 subscribe 权限。例如：

对于 **UNIX, Linux 和 Windows 系统**:

```
setmqaut -m QM1 -n SYSTEM.FTE -t queue -g FTEAGENT +put +get +inq
setmqaut -m QM1 -n SYSTEM.FTE -t topic -g FTEAGENT +pub +sub +resume
```

对于 **IBM i**:

```
GRTRMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*Q) USER(FTEAGENT) AUT(*PUT) MQMNAME('QM1')
GRTRMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*Q) USER(FTEAGENT) AUT(*GET) MQMNAME('QM1')
GRTRMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEAGENT) AUT(*PUB) MQMNAME('QM1')
GRTRMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEAGENT) AUT(*SUB) MQMNAME('QM1')
```

对于 **z/OS**:

```
RDEFINE MQQUEUE QM1.SYSTEM.FTE UACC(NONE)
PERMIT QM1.SYSTEM.FTE CLASS(MQQUEUE) ID(FTEAGENT) ACCESS(UPDATE)
RDEFINE MXTOPIC QM1.PUBLISH.SYSTEM.FTE UACC(NONE)
PERMIT QM1.PUBLISH.SYSTEM.FTE CLASS(MXTOPIC) ID(FTEAGENT) ACCESS(UPDATE)
```

如果存在连接到远程队列管理器的代理，那么可能需要授予额外的权限以允许通道将消息放入 SYSTEM.FTE 队列中。

要将消息发布到 SYSTEM.FTE 主题，SYSTEM.FTE 主题的权限记录必须允许按照消息的消息描述符结构 (MQMD) 中包含的用户标识进行发布。在[发布日志和状态消息的权限](#)中对此进行了描述。

要允许用户在 z/OS 上发布到 SYSTEM.FTE 主题，必须授予通道启动程序用户标识有关发布到 SYSTEM.FTE 主题的访问权限。如果 RESLEVEL 安全概要文件导致要检查两个用户标识的通道启动程序连接，那么您还需要授予针对包含在消息的消息描述符结构 (MQMD) 中的用户标识的访问权限。有关更多信息，请参阅[RESLEVEL 安全概要文件](#)。

在 SYSTEM.FTE 主题上接收发布的权限

传输日志消息、进度消息和状态消息旨在供常规使用，因此请为 FTEUSER 组授予预订 SYSTEM.FTE 主题的权限。例如：

对于 **UNIX, Linux 和 Windows 系统**:

```
setmqaut -m QM1 -n SYSTEM.FTE -t topic -g FTEUSER +sub
```

对于 **IBM i**:

```
GRTRMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEUSER) AUT(*SUB) MQMNAME('QM1')
```

对于 **z/OS**:

```
RDEFINE MXTOPIC QM1.SUBSCRIBE.SYSTEM.FTE UACC(NONE)
PERMIT QM1.SUBSCRIBE.SYSTEM.FTE CLASS(MXTOPIC) ID(FTEUSER) ACCESS(ALTER)
```

使用传输队列连接至远程队列管理器的权限

在多个队列管理器的拓扑中，代理请求传输队列上的 put 权限（用于连接至远程队列管理器）。

为文件传输创建临时回复队列的权限

文件传输请求将等待传输完成，并且依赖于正在创建和填充的临时回复队列。授予 FTEUSER 组对临时模型队列定义的 DISPLAY、PUT、GET 和 BROWSE 权限。例如：

对于 UNIX, Linux 和 Windows 系统:

```
setmqaut -m QM1 -n SYSTEM.DEFAULT.MODEL.QUEUE -t queue -g FTEUSER +dsp +put +get +browse
```

对于 IBM i:

```
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(FTEUSER) AUT(*ADMSP)
MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(FTEUSER) AUT(*PUT)
MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(FTEUSER) AUT(*GET)
MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(FTEUSER) AUT(*BROWSE)
MQMNAME('QM1')
```

对于 z/OS:

```
RDEFINE MQQUEUE QM1.SYSTEM.DEFAULT.MODEL.QUEUE UACC(NONE)
PERMIT QM1.SYSTEM.DEFAULT.MODEL.QUEUE CLASS(MQQUEUE) ID(FTEUSER) ACCESS(UPDATE)
```

缺省情况下, 该队列是 SYSTEM.DEFAULT.MODEL.QUEUE, 但是您可以通过在 command.properties 文件中设置“modelQueueName”和“dynamicQueuePrefix”属性的值来配置该名称。

在 z/OS 上, 您还必须授予权限来访问 FTEUSER 的临时队列。例如:

```
RDEFINE MQQUEUE QM1.WMQFTE.** UACC(NONE)
PERMIT QM1.WMQFTE.** CLASS(MQQUEUE) ID(FTEUSER) ACCESS(UPDATE)
```

缺省情况下, z/OS 上的每个临时队列的名称以 WMQFTE 开头。

下表总结了所描述的安全方案中 FTEUSER 和 FTEAGENT 的访问控制配置:

表 34: FTEUSER 和 FTEAGENT 的访问控制配置摘要			
Object	对象类型	FTEUSER	FTEAGENT
代理队列管理器	队列管理器		CONNECT、INQ 和 SETID。还需要 ALT_USER 来启用用户权限检查。
协调队列管理器	队列管理器		
命令队列管理器	队列管理器	连接	连接
SYSTEM.FTE	本地队列		GET 和 PUT
SYSTEM.FTE.COMMAND.agent_name	本地队列	PUT	BROWSE、GET、PUT 和 SETID
SYSTEM.FTE.DATA.agent_name	本地队列		GET 和 PUT
SYSTEM.FTE.EVENT.agent_name	本地队列		BROWSE、GET 和 PUT
SYSTEM.FTE.REPLY.agent_name	本地队列		GET 和 PUT
SYSTEM.FTE.STATE.agent_name	本地队列		BROWSE、GET、INQ 和 PUT
SYSTEM.FTE.WEB.gateway_name	本地队列		PUT
SYSTEM.FTE.WEB.RESP.agent_name	本地队列		GET
SYSTEM.FTE	本地主题	SUBSCRIBE	PUBLISH 和 SUBSCRIBE

表 34: FTEUSER 和 FTEAGENT 的访问控制配置摘要 (继续)			
Object	对象类型	FTEUSER	FTEAGENT
SYSTEM.DEFAULT.MODEL.QUEUE (或 IBM MQ Managed File Transfer 中定义的用于创建临时回复队列的模型队列。)	模型队列	BROWSE、 DISPLAY、GET 和 PUT	BROWSE、 DISPLAY、GET 和 PUT
与远程队列管理器通信的传输队列	本地队列		PUT

通过 MQ Explorer 管理传输的权限

除了向此页面上已提及的情境中的用户授予 MFT 权限外，还需要向通过 MQ Explorer 管理和执行所有 MFT 操作的 MFT 代理程序用户授予进一步的权限。要发出命令（如创建、取消、调度文件传输、创建、删除资源监视器和创建传输模板），MQ Explorer 用户必须具有如下权限：

- 协调队列管理器：连接、查询和显示
- 命令队列管理器：连接、查询和显示
- SYSTEM.FTE 主题：发布和预订
- SYSTEM.MQEXPLORER.REPLY.MODEL：显示、查询、获取和浏览
- SYSTEM.ADMIN.COMMAND.QUEUE：查询、放置和显示
- SYSTEM.DEFAULT.MODEL.QUEUE：获取、放置、查询、显示和浏览

有关哪个命令直接连接到哪个队列管理器的信息，请参阅第 445 页的『MFT 命令和进程与队列管理器的连接关系』。

相关参考

第 438 页的『用户的 IBM MQ Managed File Transfer 操作权限』

除使用组来管理对资源的访问权之外，您还可以启用其他安全级别来限制用户可采取的代理操作。将代理权限队列上的权限授予用户以给予用户执行特定代理操作的许可权。

第 441 页的『记录器的权限』

运行记录器的操作系统用户需要具有对记录器队列和 SYSTEM.FTE 主题的特定 IBM MQ 权限。

用户的 IBM MQ Managed File Transfer 操作权限

除使用组来管理对资源的访问权之外，您还可以启用其他安全级别来限制用户可采取的代理操作。将代理权限队列上的权限授予用户以给予用户执行特定代理操作的许可权。

启用用户权限管理

要开启用户对代理操作的权限检查，请完成以下步骤：

1. 在 agent.properties 文件中，将 authorityChecking 的值设置为 true。
2. 确保运行代理的用户具有对代理队列管理器的 IBM MQ 备选用户 (ALT_USER) 权限。

请注意，在 z/OS 上，必须向运行代理的用户授予 ALT_USER 权限，使这些用户标识可以请求执行代理操作的许可权。

传输中所涉及的代理都必须启用相同的安全级别，即，在代理的属性文件中，authorityChecking 必须设置为相同的值。authorityChecking 属性值不同的代理之间的传输将失败。

代理权限队列

代理具有的权限队列可用于管理哪些用户有权限执行某些代理操作。代理无法将消息置入这些队列中，也无法从其中获取消息。代理权限队列如下：

- SYSTEM.FTE.AUTHADM1.agent_name
- SYSTEM.FTE.AUTHAGT1.agent_name
- SYSTEM.FTE.AUTHMON1.agent_name

- SYSTEM.FTE.AUTHOPS1.agent_name
- SYSTEM.FTE.AUTHSCH1.agent_name
- SYSTEM.FTE.AUTHTRN1.agent_name

当通过设置代理属性 **authorityChecking=true** 来启用用户权限管理时，用户对代理权限队列具有的权限可指定该用户有权执行的操作。

要点: **V8.0.0.8** 从 IBM MQ 8.0.0 修订包 8 开始，**inquire** 在所有代理权限队列上都是必需的许可权。

下表汇总了用户或组需要的 IBM MQ 访问权限 以及代理权限队列上的 **inquire permission** 以执行特定操作。

表 35: 用户或组在代理权限队列上执行特定操作所需的 IBM MQ 访问权限级别。			
用户操作	IBM MQ Managed File Transfer 访问权限	权限队列	IBM MQ 访问权限 (分布式平台)
在 fteStopAgent 命令上使用 -m 选项来关闭代理。	管理	SYSTEM.FTE.AUTHADM1.agent_name	BROWSE
启动来自该代理的文件传输	传输源	SYSTEM.FTE.AUTHTRN1.source_agent_name	BROWSE
取消由相同用户启动的来自该代理的文件传输			
启动到该代理的文件传输	传输目标	SYSTEM.FTE.AUTHTRN1.destination_agent_name	PUT
取消由相同用户启动的至该代理的文件传输			
创建资源监视器	监视器	SYSTEM.FTE.AUTHMON1.monitoring_agent_name	BROWSE
删除由相同用户创建的资源监视器			
删除由任何用户创建的资源监视器	监视操作	SYSTEM.FTE.AUTHOPS1.agent_name	SET
创建调度的传输	调度	SYSTEM.FTE.AUTHSCH1.source_agent_name	BROWSE
删除由同一用户创建的调度传输			
删除由任何用户或组创建的调度传输	调度操作	SYSTEM.FTE.AUTHOPS1.agent_name	PUT
取消由任何用户或组创建的传输	传输操作	SYSTEM.FTE.AUTHOPS1.source_agent_name SYSTEM.FTE.AUTHOPS1.destination_agent_name	BROWSE

表 36: 用户或组在代理权限队列上执行特定操作所需的 IBM MQ 访问权限级别。				
用户操作	IBM MQ Managed File Transfer 访问权限	权限队列	IBM MQ 访问权限 (分布式平台)	RACF 访问级别 (仅 z/OS)
在 fteStopAgent 命令上使用 -m 选项来关闭代理。	管理	SYSTEM.FTE.AUTHADM1.agent_name	BROWSE	读取
启动来自该代理的文件传输	传输源	SYSTEM.FTE.AUTHTRN1.source_agent_name	BROWSE	读取
取消由相同用户启动的来自该代理的文件传输				

表 36: 用户或组在代理权限队列上执行特定操作所需的 IBM MQ 访问权限级别。(继续)				
用户操作	IBM MQ Managed File Transfer 访问权限	权限队列	IBM MQ 访问权限 (分布式平台)	RACF 访问级别 (仅 z/OS)
启动到该代理的文件传输	传输目标	SYSTEM.FTE.AUTHTRN1.destination_agent_name	PUT	更新
取消由相同用户启动的至该代理的文件传输				
创建资源监视器	监视器	SYSTEM.FTE.AUTHMON1.monitoring_agent_name	BROWSE	读取
删除由相同用户创建的资源监视器				
删除由任何用户创建的资源监视器	监视操作	SYSTEM.FTE.AUTHOPS1.agent_name	SET	更改
创建调度的传输	调度	SYSTEM.FTE.AUTHSCH1.source_agent_name	BROWSE	读取
删除由同一用户创建的调度传输				
删除由任何用户或组创建的调度传输	调度操作	SYSTEM.FTE.AUTHOPS1.agent_name	PUT	更新
取消由任何用户或组创建的传输	传输操作	SYSTEM.FTE.AUTHOPS1.source_agent_name SYSTEM.FTE.AUTHOPS1.destination_agent_name	BROWSE	读取

注: 要授予用户或组许可权以设置启动传输的资源监视器或调度的传输，用户需要 **监视器** 或 **调度** 权限以及 **传输源** 和 **传输目标** 权限。

用户可以启动一个代理并希望它与另一个代理交互。两个代理的交互方式取决于用户在另一个代理权限队列上具有的访问权限级别。

表 37: 启动代理的用户为在代理之间传输文件而在另一个代理权限队列上需要具有的 IBM MQ 访问权限级别。				
代理操作	IBM MQ Managed File Transfer 访问权限	权限队列	IBM MQ 访问权限 (分布式平台)	RACF 访问级别 (仅 z/OS)
接收的传输来自于 <source_agent>	代理源	SYSTEM.FTE.AUTHAGT1.source_agent_name	BROWSE	读取
发送传输至 <目标代理程序>	代理目标	SYSTEM.FTE.AUTHAGT1.destination_agent_name	PUT	更新

配置用户权限管理

要授权用户在代理上执行操作，请在相关的权限队列上授予该用户适当权限。要向用户授予权限，请完成以下步骤：

1. 在代理队列管理器所在系统上创建用户，该用户与您要授予其执行代理操作权限的用户同名。该用户无需处于活动状态。
2. 在相关的权限队列上授予用户适当的权限。如果您正在使用 Linux，UNIX 或 Windows，那么可以使用 `setmqaut` 命令。
3. 刷新队列管理器的安全配置。您可以使用 `REFRESH SECURITY MQSC` 命令。

示例

`setmqaut` 命令不可在 z/OS 或 IBM i 系统上使用。对于 z/OS，请使用 RACF 代替。请参阅 [Setting up security on z/OS](#)，以了解更多信息。

对于 IBM i，请参阅 [IBM MQ 对象的访问权限](#)，其中描述了如何完成 IBM MQ 对象的授权。在 IBM i 上提供了三个相关 CL 命令：**Grant MQ Object Authority (GRTMQMAUT)**，**Revoke MQ Object Authority (RVKMQMAUT)**和 **Refresh MQ Authority (RFRMQMAUT)**。

某用户（组 requestor_group 的成员）要在 AGENT1 上设置资源监视器，以将文件从用户 user1（组 user1_group 的成员）下运行的 AGENT1 传输到用户 user2（组 user2_group 的成员）下运行的 AGENT2。AGENT1 连接到 QM1；AGENT2 连接到 QM2。两个代理均已启用权限检查。为此，请执行以下步骤：

1. requestor 必须具有 AGENT1 上的**监视**权限。通过在正运行 QM1 的系统上运行以下命令来设置此权限：

```
setmqaut -m QM1 -t queue -n SYSTEM.FTE.AUTHMON1.AGENT1 -g requestor_group +browse
```

2. requestor 必须具有 AGENT1 上的**传输源**权限。通过在正运行 QM1 的系统上运行以下命令来设置此权限：

```
setmqaut -m QM1 -t queue -n SYSTEM.FTE.AUTHTRN1.AGENT1 -g requestor_group +browse
```

3. requestor 必须具有 AGENT2 上的**传输目标**权限。通过在正运行 QM2 的系统上运行以下命令来设置此权限：

```
setmqaut -m QM2 -t queue -n SYSTEM.FTE.AUTHTRN1.AGENT2 -g requestor_group +put
```

4. user2 必须具有 AGENT1 上的**代理源**权限。通过在正运行 QM1 的系统上运行以下命令来设置此权限：

```
setmqaut -m QM1 -t queue -n SYSTEM.FTE.AUTHAGT1.AGENT1 -g user2_group +browse
```

5. user1 必须具有 AGENT2 上的**代理目标**权限。通过在正运行 QM2 的系统上运行以下命令来设置此权限：

```
setmqaut -m QM2 -t queue -n SYSTEM.FTE.AUTHAGT1.AGENT2 -g user1_group +put
```

日志记录

如果已启用用户权限检查，那么失败的权限检查会导致将未经授权的日志消息发布到协调队列管理器。请参阅第 871 页的『[针对安全性的消息格式](#)』，以获取更多信息。

有关用户权限的消息可写入代理事件日志。您可通过在代理属性文件中设置 logAuthorityChecks 属性来配置写入代理事件日志的信息量。缺省情况下，权限检查记录的级别为 None。您还可将 logAuthorityChecks 的值设置为 Failures，以指定仅报告失败的授权检查，或者将该值设置为 All 以指定报告失败和成功的授权检查。

请参阅第 591 页的『[agent.properties 文件](#)』，以获取更多信息。

相关参考

第 433 页的『[特定于 IBM MQ Managed File Transfer 的资源的组权限](#)』

无需针对可能涉及到的所有对象而为个别用户授予权限，而是请配置以下两个安全组以用于管理 IBM MQ Managed File Transfer 访问控制：FTEUSER 和 FTEAGENT。IBM MQ 管理员应负责创建并填充这些组。管理员可选择扩展或修改此处所述的建议配置。

第 441 页的『[记录器的权限](#)』

运行记录器的操作系统用户需要具有对记录器队列和 SYSTEM.FTE 主题的特定 IBM MQ 权限。

第 576 页的『[fteStopAgent（停止 IBM MQ Managed File Transfer 代理）](#)』

使用 **fteStopAgent** 命令以受控方式停止 IBM MQ Managed File Transfer 代理程序，或者使用 **-i** 参数在必要时立即停止代理程序。

记录器的权限

运行记录器的操作系统用户需要具有对记录器队列和 SYSTEM.FTE 主题的特定 IBM MQ 权限。

运行记录器的操作系统用户需要具有以下 IBM MQ 权限：

- 协调队列管理器上的 CONNECT 和 INQUIRE。
- SYSTEM.FTE 主题上的 SUBSCRIBE 许可权。
- 对 SYSTEM.FTE.LOG.RJCT.*logger_name* 队列的 PUT 许可权。
- 对 SYSTEM.FTE.LOG.CMD.*logger_name* 队列的 GET 许可权。

相关参考

第 433 页的『特定于 IBM MQ Managed File Transfer 的资源的组权限』

无需针对可能涉及到的所有对象而为个别用户授予权限，而是请配置以下两个安全组以用于管理 IBM MQ Managed File Transfer 访问控制：FTEUSER 和 FTEAGENT。IBM MQ 管理员应负责创建并填充这些组。管理员可选择扩展或修改此处所述的建议配置。

第 438 页的『用户的 IBM MQ Managed File Transfer 操作权限』

除使用组来管理对资源的访问权之外，您还可以启用其他安全级别来限制用户可采取的代理操作。将代理权限队列上的权限授予用户以给予用户执行特定代理操作的许可权。

包含敏感信息的配置文件的许可权

用于存储敏感配置信息的任何文件（指的是从 IBM MQ 配置树中引用的任何文件）不得具有系统范围的读、写或（适用时）删除许可权。这些限制也适用于信任密钥库和密钥库文件。

如果某个 IBM MQ Managed File Transfer 进程检测到某个配置文件包含敏感信息，是密钥库或信任密钥库文件，并且具有系统范围的读、写或删除许可权，那么该进程将执行以下某个操作：

- 如果是在启动时检测到该情况，那么无法启动。
- 如果是在运行时检测到该情况，那么生成警告消息并忽略该配置文件的内容。这与协议网桥和 Connect:Direct 网桥相关，如果在进程运行期间某个配置发生变更，这些网桥会重新装入该配置。

在具有 UNIX 类型文件系统的系统上

确定某个文件是否具有不可接受的系统范围许可权的条件为：

- 已向其他类授予对该文件的读许可权
- 已向其他类授予对该文件的写许可权
- 已向其他类授予对包含该文件的目录的写许可权

在 Windows 系统上

确定某个文件是否具有不可接受的系统范围许可权的条件为：

- Everyone、Guests、Users 组中的任何组具有以下任何许可权：
 - 对该文件的“读数据”许可权
 - 对该文件的“附加数据”许可权
 - 对该文件的“写数据”许可权
- Everyone、Guests 或 Users 组中的任何组具有对包含该文件的文件夹的“创建文件”许可权，并且还具有以下任何许可权：
 - 对包含该文件的文件夹的“删除子文件夹和文件”许可权
 - 对该文件的删除许可权

用于发布日志和状态消息的权限

代理可发出在协调队列管理器上发布的各种日志、进展和状态消息。这些消息的发布受 IBM MQ 安全模型控制，在某些情况下，您可能需要执行进一步配置才能启用发布。

有关 IBM MQ 安全性的更多信息，请参阅以[安全性](#)开头的部分。

IBM MQ Managed File Transfer 代理使要发布的消息流入协调队列管理器上的 SYSTEM.FTE 队列。每条消息都在其消息描述符 (MQMD) 中具有一个用户标识。消息使用又名 SYSTEM.FTE 的主题对象进行发布。要发布给定的消息，SYSTEM.FTE 主题的权限记录必须允许消息的 MQMD 中所包含的用户标识进行发布。

在 z/OS 上，通道启动程序用户标识需要具有发布到 SYSTEM.FTE 主题的访问权限。如果 RESLEVEL 安全概要文件导致选中两个用户标识来进行通道启动程序连接，那么消息 MQMD 中的用户标识也需要具有发布到此主题的访问权限。

最初包含在消息中的用户标识取决于代理连接到其自身队列管理器的方式。来自与绑定连接的代理的消息包含用于运行代理的用户标识。来自与客户机连接的代理的消息均包含内部 IBM MQ 用户标识。

您可以更改消息中的用户标识。对于同时与客户机和绑定连接的代理，可以使用属性 publicationMDUser（在 agent.properties 文件中）来指定用户标识，该标识用于来自该代理的所有日志和状态消息。必须由代理自身的队列管理器授予其使用该备用用户标识的许可权；通过将 setid 权限授予用于运行代理的用户标识来授予该许可权。

您还可以在代理用于连接其队列管理器的通道上使用 MCAUSER 属性，更改来自与客户机连接代理的所有消息中包含的用户标识。

您可以使用通道出口更改消息中的用户标识，例如，在使消息进入协调队列管理器的接收方通道上。

根据 IBM MQ 拓扑和策略，IBM MQ 管理员可以通过多种方式使用本主题中的信息来确保发布了状态和日志消息。以下列举两个示例：

- 确定网络中的代理使用的所有用户标识。明确地为其中每个标识授予权限记录。
- 创建一个或多个常见用户名，以发布日志和状态消息。在协调队列管理器上创建这些用户名的权限记录。将每个代理的 publicationMDUser 属性设置为常见用户名。在每个代理队列管理器上，将 setid 权限授予给用于运行代理的用户标识，使其接受 publicationMDUser 属性。

文件系统访问权限

针对任何文件传输请求，代理进程都需要其本地文件系统的某个访问权级别。

- 要从源文件传输，源代理运行所使用的用户标识必须具有对源文件的读访问权。此外，您可能需要根据源处置属性，向源代理提供删除或写权限。
- 要传输至文件或目录，目标代理运行所使用的用户标识必须具有对指定路径的写权限。此外，您可能需要根据目标存在属性，向目标代理提供更新权限。
- 除您授予代理进程的文件访问权限外，您还可以使用沙箱来指定并强制执行受限的文件路径区域。有关更多信息，请参阅第 99 页的『沙箱』。
- 如果您要接收或发送的文件不在代理可访问的位置（例如，VSAM 数据集或所在位置受沙箱功能限制），那么可以使用 IBM MQ Managed File Transfer 用户出口，将文件移至代理可访问的位置，或从代理可访问位置移出文件。有关更多信息，请参阅第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』。

commandPath 属性

使用 commandPath 属性以指定 IBM MQ Managed File Transfer 可从中运行命令的位置。设置该属性时务必谨慎，因为能够向代理发送命令的远程客户机系统可以有效调用指定的某个 commandPath 中的任何命令。

您可以指定在某个系统上运行命令，在该系统上，代理通过 IBM MQ Managed File Transfer 的受管传输和受管调用功能运行。有关更多信息，请参阅[程序调用](#)。但是，命令必须位于 commandPath 代理属性引用的路径上。

如果指定的命令不标准，IBM MQ Managed File Transfer 会尝试在命令路径中查找匹配的命令。如果命令路径中存在多条匹配的命令，那么会使用第一条匹配的命令。

缺省情况下，commandPath 属性为空，因此代理无法调用任何命令。

指定 commandPath 代理属性，如下所示：

```
commandPath=command_directory_name  
separator...command_directory_name
```

或者，仅适用于 z/OS，请指定：

```
commandPath=command_directory_name_or_data_set_name_prefix  
separator...command_directory_name_or_data_set_name_prefix
```

其中：

- `command_directory_name` 表示可以运行命令的目录路径。
- `command_directory_name_or_data_set_name_prefix` 表示可以运行命令的 z/OS UNIX 系统服务的目录路径，或表示以 "/" 开头的数据集名称的前缀。您可以选择使用标准或非标准的数据集名称前缀（即 `//HLQ...` 或 `//HLQ()` 的形式）。以 `//HLQ()...` 或 `//HLQ()...` 形式指定分区数据集。使用数据集只能指定 JCL 脚本命令。
- `separator` 是特定于平台的分隔符。

例如，在 UNIX 系统上，如果要运行位于 `/home/user/cmds1` 和 `/home/user/cmds2` 目录中的命令，请按如下所示设置 `commandPath` 代理属性：

```
commandPath=/home/user/cmds1:/home/user/cmds2
```

例如，在 Windows 系统上，如果要运行位于目录 `C:\File Transfer\commands` 和 `C:\File Transfer\agent commands` 中的命令，请按如下所示设置 `commandPath` 代理属性：

```
commandPath=C:\\File Transfer\\commands;C:\\File Transfer\\agent commands
```

在 Windows 系统上，必须对分隔符反斜杠 (\) 进行转义，并输入为双反斜杠 (\\)。反斜杠字符 (\) 也可以替换为正斜杠 (/)。

例如，在 IBM 4690 系统上，分隔符是分号 (;)。要运行位于目录 `f:/fteuser/cmds` 和 `mqftcmds:/public` 中的命令，其中 `mqftcmds` 是对包含这些命令的目录定义的 IBM 4690 逻辑名，请按如下所示设置 `commandPath` 代理属性：

```
commandPath=f:/fteuser/cmds;mqftcmds:/public
```

例如，在 z/OS 系统中，如果您希望运行的命令：

- 位于 `/home/user/cmds1` 和 `/home/user/cmds2` 目录下
- 位于以 `//'USER.CMD1'`，`//CMD2`，开头的数据集中
- 是名为 `//'USER.CMD5'` 的标准 PSD 的成员

请按照以下方式设置 `commandPath` 代理属性：

```
commandPath=/home/user/cmds1:/home/user/cmds2://'USER.CMD1'://CMD2://'USER.CMD5()'
```

要点: **V8.0.0.6** 设置该属性时要千万小心，因为能够向代理发送命令的远程客户机系统可以调用指定的某个 `commandPath` 中的任何命令。为此，缺省情况下，当您指定 `commandPath` 属性时，会配置沙箱，以便针对传输自动拒绝访问所有 `commandPath` 目录（及其子目录）：

- 如果代理配置为使用代理沙箱，那么在代理启动时会自动将 `commandPath` 目录添加到拒绝的目录列表。
- 如果代理配置了一个或多个用户沙箱，那么在代理启动时会将 `commandPath` 目录作为 `<exclude>` 元素添加到每个用户沙箱的 `<read>` 和 `<write>` 元素中。
- 如果未将代理配置为使用代理沙箱或用户沙箱，那么在代理启动时会创建一个新的代理沙箱，其中将 `commandPath` 目录指定为拒绝的目录。

V8.0.0.6 您可以覆盖此行为以兼容以下发行版：

- IBM WebSphere MQ 文件传输版。
- IBM WebSphere MQ V7.5.0.1 受管文件传输 组件（或更低版本）。

- 未设置安装属性 enableFunctionalFixPack=7502 的安装上的 IBM WebSphere MQ V7.5.0.2 受管文件传输组件（或更高版本）。

您可以通过向 `agent.properties` 文件添加以下属性来覆盖此行为：

```
addCommandPathToSandbox=false
```

V 8.0.0.6 如果 `addCommandPathToSandbox` 属性存在且设置为 `false`，那么会发生以下行为：

- 如果代理配置为使用代理沙箱并且沙箱未指定任何允许的目录，那么在代理启动时会自动将 `commandPath` 目录添加到拒绝的目录列表。
- 如果将代理配置为使用代理沙箱并且沙箱已指定一个或多个允许的目录，那么在代理启动时不会将 `commandPath` 目录添加到拒绝的目录列表。
- 如果代理配置了一个或多个用户沙箱，那么这些用户沙箱不会发生更改，并且不会将 `commandPath` 目录作为 `<exclude>` 元素添加到每个用户沙箱的 `<read>` 和 `<write>` 元素中。
- 如果未将代理配置为使用代理沙箱或用户沙箱，那么在代理启动时会创建一个新的代理沙箱，其中将 `commandPath` 目录指定为拒绝的目录。

`commandPath` 属性在第 591 页的『`agent.properties` 文件』中进行了描述。

MFT 命令和进程与队列管理器的连接关系

每个 Managed File Transfer 拓扑都包含多个不同组件。

这些组件是：

- 一个或多个代理，包括其关联的代理队列管理器
- 一个协调队列管理器
- 一个命令队列管理器
- 多个用于管理拓扑和提交受管传输的命令
- 可选记录器，用于收集有关由拓扑中的代理执行的受管传输的信息
- MQ Explorer Managed File Transfer 插件，可用于执行某些管理任务和查看有关受管传输的信息。

代理、记录器、命令和 MQ Explorer Managed File Transfer 插件在运行时会连接到一个或多个队列管理器。

下表汇总了代理、记录器、命令和 MQ Explorer Managed File Transfer 插件在运行时连接到的队列管理器。

如果针对表中某条命令或某个进程没有出现 X 字符，那么表明此命令运行时不与任何队列管理器或进程连接。

表 38: Managed File Transfer 命令与队列管理器的连接关系的汇总				
命令名	代理队列管理器	命令队列管理器	协调队列管理器	记录器队列管理器
fteAnt				
fteCancelTransfer		X		
fteChangeDefaultConfigurationOptions				
fteCleanAgent	X			
fteCreateAgent	X			
fteCreateBridgeAgent	X			
fteCreateCDAgent	X			
fteCreateLogger				
fteCreateMonitor		X		
fteCreateTemplate			X	
fteCreateTransfer		X		
fteDefine				

表 38: Managed File Transfer 命令与队列管理器的连接关系的汇总 (继续)				
命令名	代理队列管理器	命令队列管理器	协调队列管理器	记录器队列管理器
fteDelete				
fteDeleteAgent	X		X	
fteDeleteLogger				
fteDeleteMonitor		X		
fteDeleteScheduledTransfer		X		
fteDeleteTemplates			X	
fteDisplayVersion				
fteListAgents			X	
fteListMonitors			X	
fteListScheduledTransfers			X	
fteListTemplates			X	
fteMigrateAgent				
fteMigrateConfigurationOptions				
fteMigrateLogger				
fteModifyAgent				
fteModifyLogger				
fteObfuscate				
ftePingAgent		X		
fteRAS				
fteSetAgentTraceLevel				
fteSetLoggerTraceLevel				
fteSetupCommands				
fteSetupCoordination				
fteShowAgentDetails			X	
fteShowLoggerDetails				
fteStartAgent				
fteStartLogger				
fteStopAgent		X		
fteStopLogger		X		

表 39: Managed File Transfer 进程与队列管理器的连接关系的汇总				
进程	代理队列管理器	命令队列管理器	协调队列管理器	记录器队列管理器
Managed File Transfer 代理程序	X			
Managed File Transfer 插件 MQ Explorer		X	X	
Managed File Transfer 记录器			X	X

可在关联的属性文件中指定包含连接到每种类型的队列管理器（即，代理、命令和协调队列管理器）所需的凭证信息的文件。例如，协调队列管理器有一个 `coordination.properties` 文件。在该文件中，您可以将 **coordinationQMgrAuthenticationCredentialsFile** 属性设置为指向凭证文件。

连接到协调队列管理器的命令将使用该文件中指定的凭证信息。如果在队列管理器上已启用安全性，但未正确设置该属性，那么 MFT 命令将无法成功完成。有关更多信息，请参阅 [MFT 和 IBM MQ 连接认证](#)。

IBM MQ Managed File Transfer 命令汇总

下表列出了所有 IBM MQ Managed File Transfer 命令以及指向其详细描述链接。

表 40: IBM MQ Managed File Transfer 命令及其用途	
命令名	用途
迁移命令:	
fteMigrateAgent	将代理及其配置从 IBM MQ Managed File Transfer V7.0 迁移至 WebSphere MQ V7.5 或更高版本
fteMigrateConfigurationOptions	将 IBM MQ Managed File Transfer V7.0 配置迁移到 WebSphere MQ V7.5 或更高版本
fteMigrateLogger	将独立数据库记录器的配置从 IBM MQ Managed File Transfer V7.0.1 或更高版本迁移至 WebSphere MQ V7.5 或更高版本。
配置命令:	
fteChangeDefaultConfigurationOptions	更改希望 IBM MQ Managed File Transfer 使用的缺省配置选项
fteCreateAgent	创建 IBM MQ Managed File Transfer 代理
fteCreateWebAgent	创建 IBM MQ Managed File Transfer Web 代理
fteCreateBridgeAgent	创建 IBM MQ Managed File Transfer 协议网桥代理
fteCreateCDAgent	创建 IBM MQ Managed File Transfer Connect:Direct 网桥代理
fteCreateLogger	创建 IBM MQ Managed File Transfer 记录器
fteDefine	生成定义指定对象所需的配置脚本。
fteDelete	生成移除指定对象所需的配置脚本。
fteDeleteAgent	删除特定的 IBM MQ Managed File Transfer 代理
fteDeleteLogger	删除 IBM MQ Managed File Transfer 记录器
fteModifyAgent	仅适用于 Windows。修改代理、Web 代理、Connect:Direct 网桥代理或协议网桥代理，使之作为 Windows 服务运行。
fteModifyLogger	仅适用于 Windows。修改记录器，使之作为 Windows 服务运行。
fteSetupCommands	指定当您发出命令时连接到 WebSphere MQ 网络的队列管理器的详细信息
fteSetupCoordination	配置 IBM MQ Managed File Transfer 协调队列管理器
管理命令:	
fteAnt	在可以使用文件传输 Ant 任务的环境中运行 Ant 脚本。
fteCancelTransfer	取消一次文件传输
fteCleanAgent	清除由代理使用的队列
fteCreateMonitor	创建并启动新的资源监视器
fteCreateTemplate	创建传输模板供以后使用
fteCreateTransfer	创建并启动新的文件传输
fteDeleteMonitor	停止并移除现有的资源监视器
fteDeleteScheduledTransfer	删除先前调度的特定文件传输
fteDeleteTemplates	删除现有的文件传输模板
fteListAgents	列出所有针对特定协调队列管理器注册的代理程序
fteListMonitors	列出所有针对特定协调队列管理器注册的资源管理器
fteListScheduledTransfers	列出先前使用命令行或 WebSphere MQ Explorer 创建的所有 IBM MQ Managed File Transfer 传输。
fteListTemplates	列出协调队列管理器的所有文件传输模板

表 40: IBM MQ Managed File Transfer 命令及其用途 (继续)	
命令名	用途
ftePingAgent	对代理程序执行 Ping 操作以确定代理程序是否活动以及是否可以处理传输。
fteShowAgentDetails	显示特定代理的详细信息
fteShowLoggerDetails	显示特定记录器的详细信息
fteStartAgent	在将特定代理程序用于文件传输之前先启动该代理程序
fteStartLogger	启动记录器
fteStopAgent	停止特定代理
fteStopLogger	停止记录器
安全命令:	
fteObfuscate	加密凭证文件中的敏感数据。
用于故障诊断的命令:	
fteDisplayVersion	显示产品版本
fteSetAgentTraceLevel	设置要运行的代理跟踪的级别
fteSetLoggerTraceLevel	设置要运行的记录器跟踪的级别
fteRAS	运行 RAS 收集工具

请参阅第 430 页的『已安装的命令集』，以获取关于哪些命令与哪种 IBM MQ Managed File Transfer 产品一起安装的表。

将以语法图（称为轨道图）形式提供每个命令及其参数的语法。有关如何解释轨道图的信息，请参阅[如何阅读轨道图](#)。

相关概念

第 448 页的『使用 IBM MQ Managed File Transfer 命令的权限』

第 699 页的『IBM MQ Managed File Transfer 的对象命名约定』

相关参考

第 445 页的『MFT 命令和进程与队列管理器的连接关系』

每个 Managed File Transfer 拓扑都包含多个不同组件。

第 452 页的『fteBatch、fteCommon 和 ftePlatform 脚本』

fteBatch， fteCommon 和 ftePlatform 是 IBM MQ Managed File Transfer 在 MQ_INSTALLATION_PATH/bin 目录中作为辅助脚本提供的脚本。并非每一个平台上都有这些脚本。

使用 IBM MQ Managed File Transfer 命令的权限

如果您要发出 IBM MQ Managed File Transfer 命令，您的用户标识必须是 mqm 组的成员，除非已将 IBM MQ 配置为允许不属于 mqm 组的用户发出这些命令。

有关在 z/OS 上定义 mqm 组的替代组的更多信息，请参阅[向 IBM MQ for z/OS 发出命令](#)

有关授权的更多信息，请参阅[IBM MQ 的管理权限](#)。如果您正在使用 IBM i，请从以下主题开始：[IBM MQ 权限](#)。

您可以使用 IBM MQ Explorer 发出一部分 IBM MQ Managed File Transfer 命令。

从 Windows 和 UNIX 系统发出命令

发出命令时请注意以下特定于环境的信息：

IBM MQ Managed File Transfer（针对 Windows）

所有命令均可从命令行发出。命令名不区分大小写：您可以大写、小写或大小写组合来输入命令。但是，控制命令（例如队列名称）和参数（例如，队列管理器名称的 **-m**）的自变量区分大小写。

在语法描述中，连字符（-）用作标志指示符。

IBM MQ Managed File Transfer for UNIX 系统

所有 IBM MQ Managed File Transfer 命令均可从 shell 发出。所有命令都区分大小写。

从 z/OS 系统发出命令

IBM MQ Managed File Transfer 命令安装在安装产品时选择的位置的 bin 子目录中。可通过以下任一选项来运行命令：

- 通过指定命令路径或在用户命令路径中包含 bin 子目录，直接从 USS 环境运行命令。
- 针对特定的代理或记录器，从根据 PDSE 命令模板库进行配置的命令的 PDSE 数据集中运行命令。要了解更多信息，请参阅第 118 页的『创建 IBM MQ Managed File Transfer 代理或记录器命令数据集』。

从 IBM i 平台发出命令

在 IBM i 上发出命令时请注意以下特定于环境的信息：

- 您可以使用 Qshell 解释器启动 IBM MQ Managed File Transfer 命令。要启动 Qshell 解释器，请从 IBM i 系统命令行发出 **STRQSH** 命令。
- 在 Qshell 环境中运行命令时，命令名不区分大小写：您可以大写、小写或大小写组合来输入命令。但是，控制命令（例如队列名称）和参数（例如，队列管理器名称的 **-m**）的自变量区分大小写。

从 IBM 4690 系统发出命令

这些参考主题中详细描述 IBM MQ Managed File Transfer 命令不适用于 IBM 4690 环境。有关在 IBM 4690 环境中使用 IBM MQ Managed File Transfer 的更多信息，请参阅第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』。

已安装的命令集

下表显示了随每个组件一起安装的命令。

表 41: 每个命令集中可用的 IBM MQ Managed File Transfer 命令				
命令	代理程序 命令集	服务 命令集	Tools 命令集	记录器命令集
fteAnt			X	
fteBundleConfiguration			X（仅限分布式平台）	
fteCancelTransfer			X	
fteChangeDefaultConfigurationOptions	X	X	X	X
fteCleanAgent	X	X		
fteCreateAgent	X	X		
fteCreateBridgeAgent		X		
fteCreateCDAgent	X（仅限分布式平台）	X（仅限分布式平台）		
fteCreateLogger				X
fteCreateMonitor			X	
fteCreateTemplate			X	
fteCreateTransfer			X	
fteCreateWebAgent		X（仅限分布式平台）		
fteDefine			X（仅限分布式平台）	
fteDelete			X（仅限分布式平台）	
fteDeleteAgent	X	X		
fteDeleteLogger				X

表 41: 每个命令集中可用的 IBM MQ Managed File Transfer 命令 (继续)				
命令	代理程序 命令集	服务 命令集	Tools 命令集	记录器命令集
fteDeleteMonitor			X	
fteDeleteScheduledTransfer			X	
fteDeleteTemplates			X	
fteDisplayVersion	X	X		X
fteListAgents	X	X	X	X
fteListMonitors			X	
fteListScheduledTransfers			X	
fteListTemplates			X	
fteMigrateAgent	X	X		
fteMigrateConfigurationOptions	X	X	X	X
fteMigrateLogger				X
fteModifyAgent	X (仅 Windows)	X (仅 Windows)		
fteModifyLogger				X (仅 Windows)
fteObfuscate	X	X		X
ftePingAgent			X	
fteSetAgentTraceLevel	X	X		
fteSetLoggerTraceLevel				X
fteSetupCommands	X	X	X	X
fteSetupCoordination	X	X	X	X
fteShowAgentDetails	X	X	X	X
fteShowLoggerDetails				X
fteStartAgent	X	X		
fteStartLogger				X
fteStopAgent	X	X		
fteStopLogger				X

fteAnt (在 IBM MQ Managed File Transfer 环境中运行 Ant 任务)

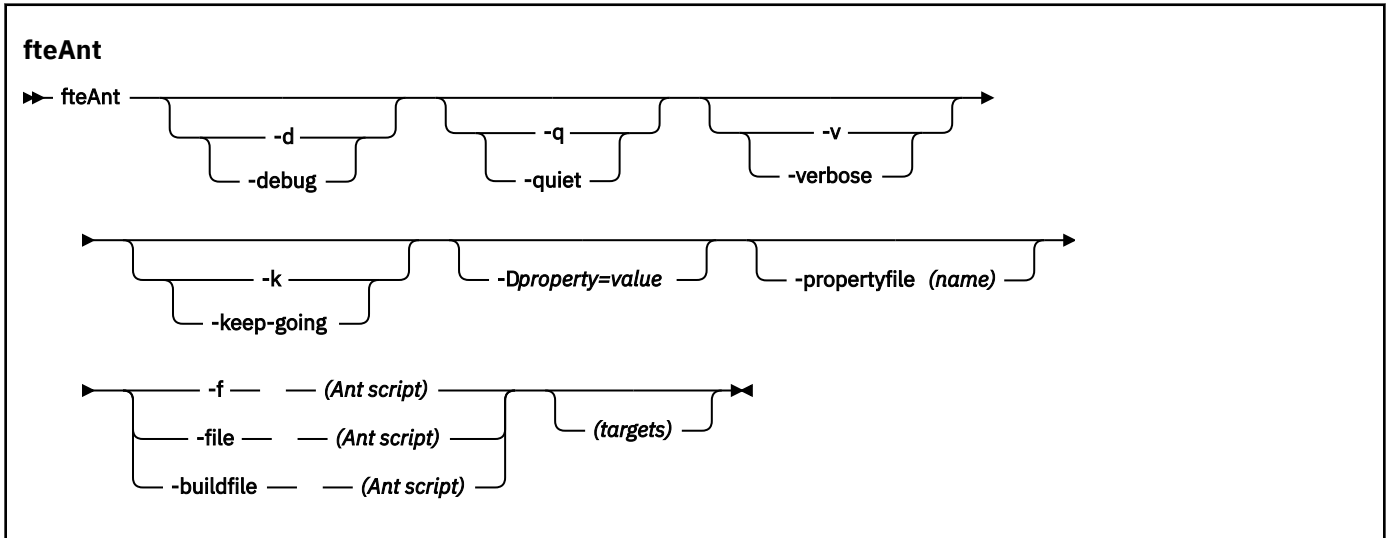
fteAnt 命令会在具有可用的 IBM MQ Managed File Transfer Ant 任务的环境中运行 Ant 脚本。

用途

使用 **fteAnt** 命令在具有 IBM MQ Managed File Transfer 的环境中运行 Ant 脚本。不同于标准 **ant** 命令，**fteAnt** 要求您定义脚本文件。

不能直接在 IBM 4690 系统上运行 **fteAnt** 命令。但可通过 Ant 脚本引用 IBM 4690 系统。有关在 IBM 4690 环境中使用 IBM MQ Managed File Transfer 的更多信息，请参阅第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』

语法



参数

-debug 或 -d

可选。生成调试输出。

-quiet 或 -q

可选。生成最小输出。

-verbose 或 -v

可选。生成详细输出。

-keep-going 或 -k

可选。执行不依赖于故障目标的所有目标。

-D property=value

可选。将 *value* 用于给定的 *property*。使用 **-D** 设置的属性的优先顺序高于属性文件中设置的属性。

使用 **com.ibm.wmqfte.propertyset** 属性来指定用于 Ant 任务的配置选项集。使用非缺省协调队列管理器的名称作为该属性的值。随后，Ant 任务会使用与该非缺省协调队列管理器关联的配置选项集。如果您未指定该属性，那么会使用基于缺省协调队列管理器的缺省配置选项集。如果您针对 Ant 任务指定 **cmdqm** 属性，那么该属性的优先顺序高于针对 **fteAnt** 命令指定的配置选项集。无论您使用缺省配置选项集还是使用 **com.ibm.wmqfte.propertyset** 属性指定设置，此行为均适用。

-propertyfile (name)

可选。从带有 **-D** 属性（高优先顺序）的文件装入所有属性。

-f (Ant 脚本)、-file (Ant 脚本) 或 -buildfile (Ant 脚本)

必需。指定要运行的 Ant 脚本的名称。

targets

可选。要从 Ant 脚本运行的一个或多个目标的名称。如果您不针对该参数指定值，那么会运行脚本的缺省目标。

-version

可选。显示 IBM MQ Managed File Transfer 命令和 Ant 版本。

-? or -h

可选。显示命令语法。

示例

在此示例中，会运行 Ant 脚本 `fte_script.xml` 中的目标 **copy**，该命令会将调试输出写入标准输出。

```
fteAnt -d -f fte_script.xml copy
```

返回码

0

命令成功完成。

1

命令结束但未成功。

其他状态返回码也可从 Ant 脚本指定，例如，通过使用 Ant fail 任务。

请参阅 [失败](#) 以获取更多信息。

相关概念

第 354 页的『入门：将 Ant 脚本用于 IBM MQ Managed File Transfer』

将 Ant 脚本用于 IBM MQ Managed File Transfer 能够使您通过已解释的脚本语言协调复杂的文件传输操作。

相关参考

第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

第 355 页的『Ant 任务样本』

随 IBM MQ Managed File Transfer 安装一起提供了一些样本 Ant 脚本。这些样本位于目录 `MQ_INSTALLATION_PATH/mqft/samples/fteant` 中。每个样本脚本都包含 `init` 目标，请编辑 `init` 目标中设置的属性，以使用您的配置运行这些脚本。

fteBatch、fteCommon 和 ftePlatform 脚本

`fteBatch`，`fteCommon` 和 `ftePlatform` 是 IBM MQ Managed File Transfer 在 `MQ_INSTALLATION_PATH/bin` 目录中作为辅助脚本提供的脚本。并非每一个平台上都有这些脚本。

fteBatch 脚本（仅 z/OS）

`fteBatch` 是用于从 JZOS Batch Launcher 运行 IBM MQ Managed File Transfer 的帮助脚本。`fteBatch` 仅安装在 z/OS 上。通常，IBM MQ Managed File Transfer 是使用提供的命令 `shell` 脚本启动的，这些脚本在启动适合于该函数的 Java 类之前执行一些环境配置。使用 JZOS Batch Launcher 启动 IBM MQ Managed File Transfer 时，将直接从 Launcher 启动 Java 类。可以在启动程序设置过程中调用 `fteBatch`，以将所需类名放入环境变量中，并执行正常命令 `shell` 脚本在启动 Java 之前执行的设置工作。这可在您的作业与 WebSphere MQ File Transfer Edition 使用的内部类名之间提供一定级别的隔离。

由于您可通过命令的新 PDSE 数据集来运行 IBM MQ Managed File Transfer，因此将不再使用 IBM MQ Managed File Transfer V8.0 的 `fteBatch` 命令。有关更多信息，请参阅第 118 页的『创建 IBM MQ Managed File Transfer 代理或记录器命令数据集』。

fteCommon

`fteCommon` 是由其他 IBM MQ Managed File Transfer 命令脚本启动的辅助脚本，用于在启动 Java 之前执行公共设置处理。

ftePlatform

`ftePlatform` 是由 `fteCommon` 脚本启动的帮助脚本，用于执行特定于平台的设置过程。

fteCancelTransfer (取消 IBM MQ Managed File Transfer 传输)

使用 **fteCancelTransfer** 命令可取消 IBM MQ Managed File Transfer 传输。您可以针对传输的源或目标代理发出该命令。

用途

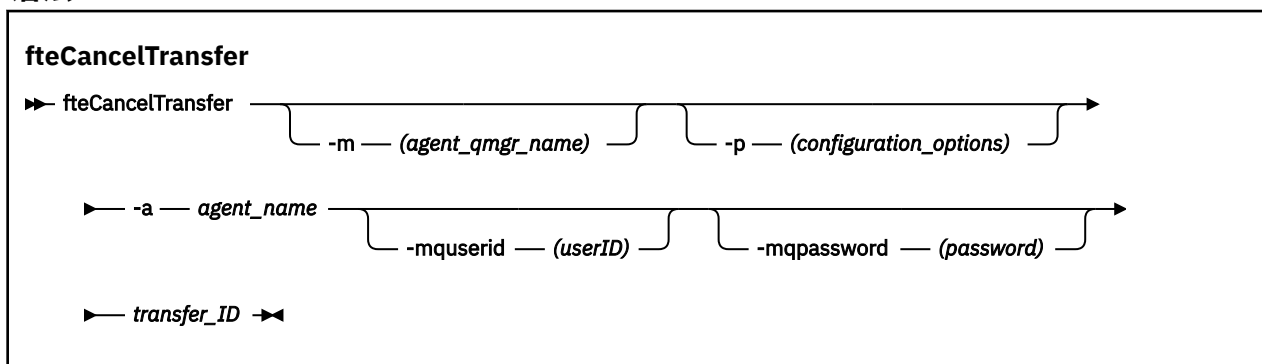
如果在传输正在进行时发出 **fteCancelTransfer** 命令，那么已在该传输过程中传输的任何文件都将保留在目标系统上而不会被删除。在该传输过程中部分传输的任何文件都将从目标系统中删除。传输为“已取消”的传输日志的目标端。

如果到 Connect:Direct 节点的传输被取消，那么在该传输过程中部分传输的任何文件都将保留在目标系统上而不会被删除。

您可以从可连接到 IBM MQ 网络并路由到代理队列管理器的任何系统中运行 **fteCancelTransfer** 命令。具体而言，要运行此命令，您必须已在该系统上安装了 IBM MQ Managed File Transfer，并将 IBM MQ Managed File Transfer 配置为与 IBM MQ 网络进行通信。如果无任何可用的连接详细信息，那么会改为将代理队列管理器详细信息用于连接，前提是这些详细信息可用。

仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。请参阅[配置选项](#)，以获取更多信息。

语法



参数

-m (agent_qmgr_name)

可选。代理队列管理器的名称。该代理必须是要取消的传输的源或目标代理。如果未指定该参数，那么会将取消请求发送至您使用的配置选项集标识的队列管理器。

-p (配置选项)

可选。该参数确定用于取消传输的配置选项集。按照约定，请使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-a (agent_name)

必需。要取消的传输的源或目标代理的名称。

-mquserid (userID)

可选。指定要向命令队列管理器进行认证的用户标识。

-mqpassword (password)

可选。指定要向命令队列管理器进行认证的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

transfer_ID

必需。要取消的传输的标识。在发出 **fteCreateTransfer** 命令之后，该传输标识（也称为请求标识）将显示在命令行中。传输标识也包含在文件传输日志消息中或显示在 "IBM MQ 资源管理器传输日志" 面板中。

-? or -h

可选。显示命令语法。

示例

在该示例中，AGENT1 是要取消的传输的源代理。

```
fteCancelTransfer -a AGENT1 414d5120514d5f4c4d343336303920201159c54820027102
```

返回码

0

命令已成功完成，或代理无法识别指定的传输标识。如果代理无法识别该传输标识，那么最有可能的原因是该传输已完成或已取消。

1

命令结束但未成功。

相关参考

第 497 页的『fteCreateTransfer（创建新的文件传输）』

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

fteChangeDefaultConfigurationOptions（更改缺省配置选项）

使用 **fteChangeDefaultConfigurationOptions** 命令来更改您希望 IBM MQ Managed File Transfer 使用的缺省配置选项。配置选项的值定义 IBM MQ Managed File Transfer 使用的属性文件组。

要点: 在分布式系统上，只有 IBM MQ 管理员（和 mqm 组的成员）身份的用户才可以运行此命令。如果尝试以非 IBM MQ 管理员用户身份运行此命令，那么您将收到错误消息 BFGCL0502E: You are not authorized to perform the requested operation. 并且不会运行该命令。

在 z/OS 系统上，用户必须满足以下至少一个条件才能运行迁移命令：

- 是 mqm 组的成员（如果在系统上定义了 mqm 组）。
- **V8.0.0.6** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V8.0.0.6** 在运行命令时，在 BFG_GROUP_NAME 环境变量中未设置任何值。

用途

首次使用 **fteSetup** 协调 命令将队列管理器配置为协调队列管理器时，将建立缺省 Managed File Transfer 配置选项。在 MFT 产品安装期间，如果 mqft 目录尚不存在，那么将在 <MQ_DATA_PATH> 下创建该目录。此外，如果配置，安装和日志目录尚不存在，那么将在 mqft 目录下创建这些目录。

通过使用 **fteChangeDefaultConfigurationOptions** 命令，您可以更改在 `installation.properties` 文件中定义的缺省协调队列管理器。如果更改该协调队列管理器，那么 IBM MQ Managed File Transfer 将使用由一组结构化的目录及其所包含的属性文件所指定的配置选项，缺省情况下您使用这些选项作为 `configuration_options` 的输入。该目录名称与该配置下代理使用的协调队列管理器名称相同。

请参阅 第 116 页的『分布式平台上的配置选项』，以获取有关 `installation.properties` 文件的更多信息。

语法

fteChangeDefaultConfigurationOptions

► **fteChangeDefaultConfigurationOptions** — *configuration_options* ◄

参数

configuration_options

必需。该参数指定要更改为的缺省配置选项。使用非缺省协调队列管理器的名称作为该参数的输入。

-? or -h

可选。显示命令语法。

示例

在该示例中，缺省配置选项被更改为 QM_COORD2:

```
fteChangeDefaultConfigurationOptions QM_COORD2
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

[第 116 页的『分布式平台上的配置选项』](#)

IBM MQ Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。这些属性文件位于安装产品时所定义的配置目录中。

fteCleanAgent（清除 IBM MQ Managed File Transfer 代理）

使用 **fteCleanAgent** 命令通过从代理使用的持久和非持久队列中删除消息来清除 IBM MQ Managed File Transfer 代理使用的队列。如果您在启动代理时存在问题，并且该问题可能是由于代理使用的队列上保留的信息所造成的，那么可以使用 **fteCleanAgent** 命令。

用途

使用 **fteCleanAgent** 命令删除代理使用的持久性和非持久性队列中的消息。具体来说，该命令可以执行以下操作：

- 在与该代理之间的传输停止之前，移除任何正在进行的传输。在代理重新启动后，不会恢复这些传输
- 移除已提交给该代理但尚未执行的所有命令。
- 删除该代理上存储的所有资源监视器
- 删除该代理上存储的所有调度的传输
- 删除该代理上存储的所有无效消息

如果该代理是 Connect:Direct 网桥代理，那么 **-ms**、**-ss** 和 **-ims** 参数无效。对于 Connect:Direct 网桥代理，该命令还执行以下操作：

- 从 Connect:Direct 网桥代理在传输文件期间用于临时存储文件的目录中删除所有文件。该目录的位置由 **cdTmpDir** 参数定义
- 显示有关与所有正在进行的传输相关联的 Connect:Direct 进程的信息

在 IBM MQ 8.0.0 修订包 7 之前，如果仅运行带有 **agent_name** 参数的 **fteCleanAgent** 命令，那么在缺省情况下，该命令将像提供了 **-all** 参数一样运行，这将导致清除代理上的所有已调度传输，资源监视器和已调度传输定义。

V 8.0.0.7 从 IBM MQ 8.0.0 修订包 7 开始，缺省情况下必须通过将相应参数传递到 **fteCleanAgent** 命令并提供代理程序名称来指定要清除的 Managed File Transfer 状态。这表示在缺省情况下，**fteCleanAgent** 不会清除所指定代理的所有进行中的传输和暂挂传输、资源监视器定义以及已调度的传输定义。可以通过将 `command.properties` 文件中的 `failCleanAgentWithNoArguments` 属性设置为相应的值来启用或禁用此行为：

- 缺省情况下，`failCleanAgentWithNoArguments` 的值为 `true`，这表示如果仅指定 **agent_name** 参数，那么 **fteCleanAgent** 命令将运行失败。
- 如果将 `failCleanAgentWithNoArguments` 设置为 `false` 并且仅指定 **agent_name** 参数，那么 **fteCleanAgent** 的行为方式将与指定 **-all** 参数时的行为方式相同。

您必须在已停止的代理上运行 **fteCleanAgent** 命令。如果在目前正在运行的代理上尝试运行 **fteCleanAgent** 命令，那么将收到一条错误信息。该命令不会启动该代理。**fteCleanAgent** 命令会清除发出该命令的系统上的代理。您无法清除远程系统上的代理。要运行 **fteCleanAgent** 命令，您必须具有对代理程序锁定文件的写访问权，该文件位于

`MQ_DATA_PATH\mqft\logs\coordination_QMgr_name\agents\agent_name\agent.lock`

FTEAGENT 组必须对以下队列具有 GET 和 BROWSE 权限才能成功运行 **fteCleanAgent**：

- `SYSTEM.FTE.COMMAND.agent_name`
- `SYSTEM.FTE.EVENT.agent_name`
- `SYSTEM.FTE.STATE.agent_name`

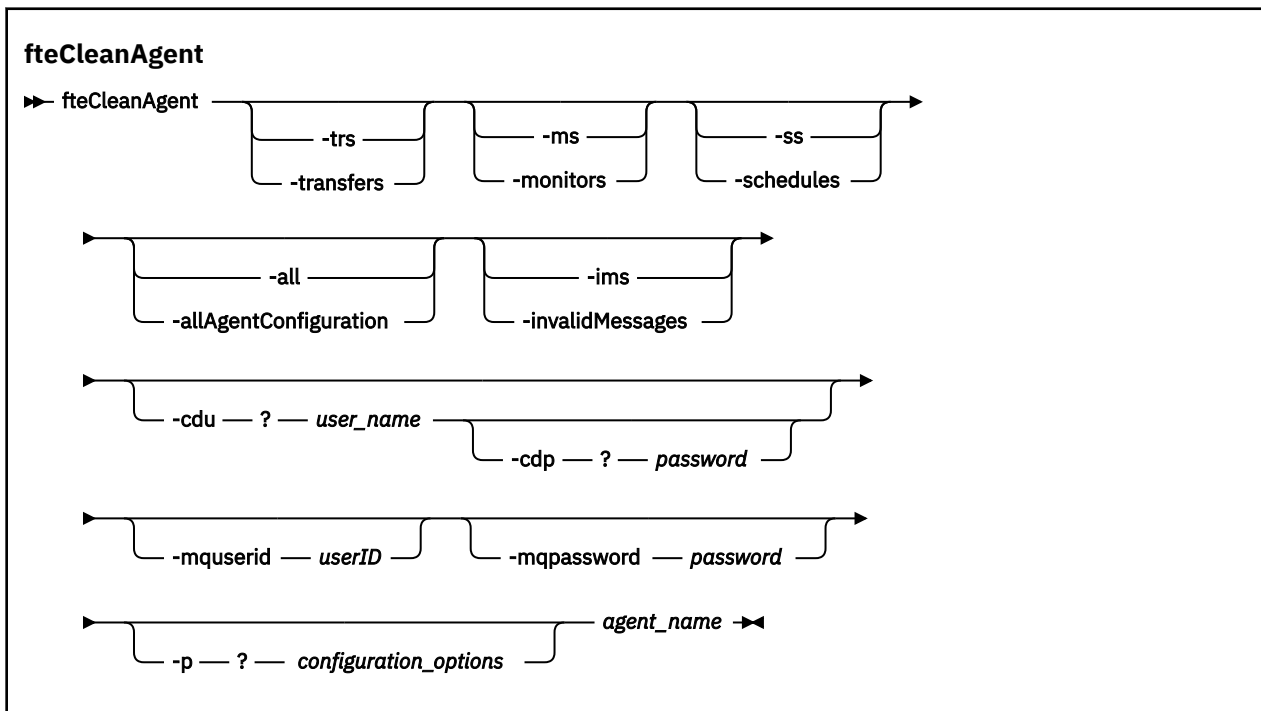
请参阅第 433 页的『特定于 IBM MQ Managed File Transfer 的资源的组权限』，以获取有关 FTEAGENT 组和限制组权限的更多信息。

如果要在以绑定方式连接到队列管理器的代理上运行 **fteCleanAgent** 命令，并且该代理刚停止运行，那么 **fteCleanAgent** 命令可能会报告消息传递问题：MQRC 2042。该 MQRC 出现的原因是队列管理器中仍存在代理的队列句柄。在短暂延迟后，队列管理器将移除该句柄，之后您可以重新发出 **fteCleanAgent**。

仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。请参阅配置选项，以获取更多信息。

注：清除 Connect:Direct 网桥代理时，用于运行 **fteCleanAgent** 命令的用户标识必须具有对 Connect:Direct 网桥代理临时目录的读写访问权。

语法



参数

您可以使用 **fteCleanAgent** 命令来删除特定工件。例如，您可以指定 **-trs** 命令来删除暂挂传输，而不更改任何资源监视器和调度的传输。

-trs 或 **-transfers**

可选。指定将从代理中删除正在进行的传输和暂挂的传输。不能将该参数与 **-all** 或 **-ims** 参数一起指定。

-ms 或 **-monitors**

可选。指定将从代理中删除所有资源监视器定义。不能将该参数与 **-all** 或 **-ims** 参数一起指定。

-ss 或 **-schedules**

可选。指定将从代理中删除所有调度的传输定义。不能将该参数与 **-all** 或 **-ims** 参数一起指定。

-all 或 **-allAgentConfiguration**

可选。指定将从代理中删除所有传输、资源监视器定义和调度的传输定义。不能将该参数与 **-trs**、**-ss**、**-ms** 或 **-ims** 参数一起指定。



注意: 仅当没有其他选项可用时，才应使用 **all** 参数。删除传输，资源监视器定义和调度的传输定义的操作可能会对您的企业产生重大影响。

-ims 或 **-invalidMessages**

可选。指定将从代理中删除所有无效的消息。不能将该参数与 **-trs**、**-ss**、**-ms** 或 **-all** 参数一起指定。

-cdu user_name

可选。仅当要清除的代理为 Connect:Direct 网桥代理时才有效。如果指定了该参数，那么该命令将使用提供的用户名来建立到 Connect:Direct 网桥节点的连接，并检索有关现有 Connect:Direct 进程的更多信息。如果未指定该参数，那么将清除代理但不会显示有关 Connect:Direct 进程的信息。

-cdp password

可选。仅当要清除的代理程序是 Connect:Direct 网桥代理并且您已指定 **-cdu** 参数时才有效。如果指定 **-cdp** 参数，那么该命令将使用提供的密码与 Connect:Direct 网桥节点建立连接，并检索有关现有

Connect:Direct 进程的其他信息。如果未指定该参数，但指定了 **-cdu** 参数，那么将以交互方式要求您提供密码。

-mquserid (*userID*)

可选。指定通过代理队列管理器进行认证时所使用的用户标识。

-mqpassword (*password*)

可选。指定通过代理队列管理器进行认证时所使用的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

-p (配置选项)

可选。该参数确定用于清除代理的配置选项集。按照约定，请使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

agent_name

必需。要清除的 IBM MQ Managed File Transfer 代理的名称。

-? or -h

可选。显示命令语法。

示例

在该基本示例中，将清除 AGENT2 使用的所有队列：

```
C:\Documents and Settings\Administrator>fteCleanAgent -all AGENT2
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED

All messages will be deleted from all queues

State Queue Entries:

Transfer Identifier:          414d5120716d31202020202020202020202020202786de4d20485b03
Source Agent Name:           AGENT2
Destination Agent Name:      AGENT3

Transfer Identifier:          414d5120716d31202020202020202020202020202786de4d20487203
Source Agent Name:           AGENT2
Destination Agent Name:      AGENT3

Command Queue New Transfer Entries:

Scheduler Queue Schedule Entries:

Directory Monitor Configuration for "MONITOR1" has been cleared from the Agent.

Schedule Identifier:         1
Source Agent Name:           AGENT2
Destination Agent Name:      AGENT3

BFGCL0149I: The agent 'AGENT2' has been cleaned.
```

在该示例中，将清除 AGENT2 使用的无效消息队列：

```
C:\Documents and Settings\Administrator>fteCleanAgent -ims AGENT2
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED

Invalid messages will be deleted from all queues

State Queue Entries:

Warning - Invalid message found on the queue

Command Queue New Transfer Entries:

Warning - Invalid message found on the queue
```


在该示例中，将清除 Connect:Direct 网桥代理 AGENT_CD_BRIDGE 使用的传输队列：

BFGCL0149I: The agent 'AGENT_CD_BRIDGE' has been cleaned.

fteCommon

fteCommon 是由其他 IBM MQ Managed File Transfer 命令脚本启动的辅助脚本，用于在启动 Java 之前执行公共设置处理。

ftePlatform

ftePlatform 是由 fteCommon 脚本启动的帮助脚本，用于执行特定于平台的设置过程。

fteCreateAgent（创建 IBM MQ Managed File Transfer 代理）

fteCreateAgent 命令创建代理及其关联的配置。

您可以控制对代理的访问。请参阅第 438 页的『[用户的 IBM MQ Managed File Transfer 操作权限](#)』以获取进一步信息。您必须使用 **-ac** 参数，并提供许可权来访问一些队列。

要点: 在分布式系统上，只有 IBM MQ 管理员（和 mqm 组的成员）身份的用户才可以运行此命令。如果尝试以非 IBM MQ 管理员用户身份运行此命令，那么您将收到错误消息 BFGCL0502E: You are not authorized to perform the requested operation. 并且不会运行该命令。

在 z/OS 系统上，用户必须满足以下至少一个条件才能运行迁移命令：

- 是 mqm 组的成员（如果在系统上定义了 mqm 组）。
- **V8.0.0.6** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V8.0.0.6** 在运行命令时，在 BFG_GROUP_NAME 环境变量中未设置任何值。

用途

使用 **fteCreateAgent** 命令可创建代理。该命令向您提供 MQSC 命令，您必须针对代理队列管理器运行这些命令以创建以下代理队列：

- SYSTEM.FTE.AUTHADM1.agent_name
- SYSTEM.FTE.AUTHAGT1.agent_name
- SYSTEM.FTE.AUTHMON1.agent_name
- SYSTEM.FTE.AUTHOPS1.agent_name
- SYSTEM.FTE.AUTHSCH1.agent_name
- SYSTEM.FTE.AUTHTRN1.agent_name
- SYSTEM.FTE.COMMAND.agent_name
- SYSTEM.FTE.DATA.agent_name
- SYSTEM.FTE.EVENT.agent_name
- SYSTEM.FTE.REPLY.agent_name
- SYSTEM.FTE.STATE.agent_name

这些队列是内部系统队列，除非您要删除该代理，否则不得修改、删除或读取这些队列中的消息。以下位置的文件中也提供了要运行的 MQSC 命令：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_create.mqsc`。

如果您以后希望删除该代理，该命令还向您提供了 MQSC 命令，必须运行这些命令以清除并随后删除该代理使用的队列。MQSC 命令位于以下位置的文件中：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_delete.mqsc`。

IBM MQ Managed File Transfer 提供帮助您配置代理的高级代理属性。在 [agent.properties](#) 文件中描述了这些属性。

您可能需要创建 MQMFTCredentials.xml 凭证文件才能使用您的代理。此文件的样本位于 `MQ_INSTALLATION_PATH\mqft\samples\credentials/` 中。有关更多信息和示例，请参阅第 873 页的『[MQMFT 凭证文件格式](#)』。

语法

fteCreateAgent

► fteCreateAgent — -agentName — *agent_name* — -agentQMGr — *agent_qmgr_name* →

└─ -agentQMGrHost — *agent_qmgr_host* ─┘

└─ -agentQMGrPort — *agent_qmgr_port* ─┘

└─ -agentQMGrChannel — *agent_qmgr_channel* ─┘

└─ -agentDesc — *agent_description* ─┘ └─ -ac ─┘
└─ -authorityChecking ─┘

└─ -s ─┘ └─ *service_name* ─┘ └─ -su ─┘ └─ *user_name* ─┘ └─ -sp ─┘ └─ *password* ─┘ └─ -sj ─┘ └─ *options* ─┘ └─ -sl ─┘ └─ *options* ─┘
└─ -n ─┘

└─ -mquserid ─┘ └─ *userID* ─┘ └─ -mqpassword ─┘ └─ *password* ─┘

└─ -credentialsFile ─┘ └─ *filePath* ─┘ └─ -p ─┘ └─ *配置选项* ─┘ └─ -f ─┘

参数

-agentName (*agent_name*)

必需。要创建的代理的名称。代理名称对其协调队列管理器必须唯一。

有关命名代理的更多信息，请参阅[对象命名约定](#)。

-agentQMGr (*agent_qmgr_name*)

必需。代理队列管理器的名称。

-agentQMGrHost (*agent_qmgr_host*)

可选。代理队列管理器的主机名或 IP 地址。

-agentQMGrPort (*agent_qmgr_port*)

可选。用于客户机与代理队列管理器的连接的端口号。

-agentQMGrChannel (*agent_qmgr_channel*)

可选。用于连接到代理队列管理器的通道名称。

-agentDesc (*agent_description*)

可选。在 IBM MQ Explorer 中显示的代理的描述。

-ac 或 -authorityChecking

可选。此参数允许进行权限检查。如果指定该参数，那么代理将检查提交请求的用户是否有权执行请求的操作。有关更多信息，请参阅第 438 页的『[用户的 IBM MQ Managed File Transfer 操作权限](#)』。

-s (*service_name*)

可选（仅限 Windows）。指示代理将作为 Windows 服务运行，必须使用 Windows 管理员用户标识运行该命令。如果未指定服务名称，那么服务将命名为 mqmftAgent<AGENT><QMGR>，其中 <AGENT> 是代理名称，<QMGR> 是代理队列管理器名称。

服务的显示名称（显示在窗口 **服务** 窗口的 **名称** 列中）始终为 **IBM MQ Managed File Transfer 代理程序 <AGENT>@<QMGR>**。

-su (user_name)

可选（仅限 Windows）。如果该代理程序将作为 Windows 服务运行，那么该参数指定作为该服务运行身份的帐户的名称。要使用 Windows 域用户帐户来运行代理，请以 DomainName\UserName 格式指定该值。要使用本地内置域中的帐户来运行该服务，请以 UserName 格式指定该值。

使用 **-su** 参数指定的 Windows 用户帐户必须具有 **Log on as a service** 权限。有关如何授予该权限的信息，请参阅第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』。

在指定 **-s** 时为必需项。等同于 **-serviceUser**。

-sp (password)

可选（仅限 Windows）。由 **-su** 或 **-serviceUser** 参数设置的用户帐户的密码。

该参数仅在指定 **-s** 时才有效。等同于 **-servicePassword**。如果在指定 **-s** 参数时不指定该参数，将生成警告消息。该消息提醒您必须使用 Windows“服务”工具来设置密码，然后该服务才会成功启动。

-sj (options)

可选（仅限 Windows）。当代理作为 Windows 服务启动时，以 **-D** 或 **-X** 格式来定义要传递到 JVM 的选项列表。这些选项使用井号 (#) 或分号 (;) 字符进行分隔。如果您必须嵌入任何 # 或分号 (;) 字符，请将其置于单引号内。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceJVMOptions**。

-sl (options)

可选（仅限 Windows）。设置 Windows 服务日志级别。有效选项是：error、info、warn、debug。缺省值为 info。如果 Windows 服务出现问题，那么该选项可能很有用。如果将其设置为 debug，那么会在服务日志文件中提供更详细的信息。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceLogLevel**。

-n

可选（仅限 Windows）。指示代理将作为正常进程运行。这与 **-s** 选项互斥。如果未指定 **-s** 参数和 **-n** 参数中的任何一个，那么会将代理程序配置为正常的 Windows 进程。

等同于 **-normal**。

-p (配置选项)

可选。该参数确定用于创建代理程序的配置选项集。按照约定，请使用非缺省协调队列管理器的名称作为该参数的输入。然后，**fteCreateAgent** 命令使用与此非缺省协调队列管理器相关联的属性文件集。

仅在您希望使用不同于缺省值的配置选项时，才指定可选的 **-p** 参数。如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-mquserid (userID)

可选。指定通过协调队列管理器进行认证时所使用的用户标识。

-mqpassword (password)

可选。指定通过协调队列管理器进行认证时所使用的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

-credentialsFile (filePath)

可选。现有或新凭证文件的完整文件路径，可以向该路径添加 IBM MQ 认证详细信息。

此命令支持向指定的 IBM MQ Managed File Transfer 凭证文件添加一组 IBM MQ 认证详细信息。启用 IBM MQ 连接认证之后，请使用此命令。如果要更新现有的详细信息，则必须使用 **-f** 强制参数。

-credentialPath (credentials_path)

该命令定义要将凭证信息迁移到的位置。此参数可以是现有凭证文件的目录路径，也可以是新凭证文件的目录路径。在 z/OS 平台上，凭证文件可以是预先存在的扩展分区数据集 (PDSE)。PDSE 可以包含现有成员，或凭证文件的新成员。必须更新 PDSE 的现有成员，以包含凭证文件。PDSE 的格式必须为分块变量。

-f

可选。强制命令覆盖不匹配的现有参数。指定此参数不会强制替换现有 Windows 服务代理。

-? or -h

可选。显示命令语法。

示例

在此示例中，AGENT3 是使用代理队列管理器 QM_NEPTUNE 创建的，并且使用缺省协调队列管理器：

```
fteCreateAgent -agentName AGENT3 -agentQMGr QM_NEPTUNE  
-agentQMGrHost myhost.ibm.com -agentQMGrPort 1415 -agentQMGrChannel CHANNEL1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

[第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』](#)

您可以作为 Windows 服务来运行 IBM MQ Managed File Transfer 代理、独立数据库记录器和独立文件记录器。如果这些 Windows 服务出现任何问题，您可以使用服务日志文件以及本主题中的信息来诊断问题。

相关任务

[第 216 页的『将代理作为 Windows 服务启动』](#)

您可以将代理作为 Windows 服务启动，以便当您从 Windows 注销后，代理可以继续运行，并且可以接收文件传输。

相关参考

[第 573 页的『fteStartAgent（启动 IBM MQ Managed File Transfer 代理）』](#)

fteStartAgent 命令从命令行启动 IBM MQ Managed File Transfer 代理程序。

[第 516 页的『fteCreateWebAgent（创建 IBM MQ Managed File Transfer Web 代理）』](#)

fteCreateWebAgent 命令创建代理及其关联的配置，以用于 Web 网关。IBM MQ Managed File Transfer 服务器随附该命令。

[第 81 页的『在 4690 OS 系统上运行时的限制』](#)

在零售环境中的 4690 OS 系统上运行 IBM MQ Managed File Transfer 时，存在一些限制和不受支持的功能。

[第 522 页的『fteDeleteAgent（删除 IBM MQ Managed File Transfer 代理）』](#)

fteDeleteAgent 命令删除 IBM MQ Managed File Transfer 代理及其配置。如果该代理是协议网桥代理，那么用户凭证文件将留在文件系统中。

fteCreateBridgeAgent（创建和配置 IBM MQ Managed File Transfer 协议网桥代理）

fteCreateBridgeAgent 命令创建协议网桥代理及其关联的配置。针对您要与之进行文件的发送和接收操作的每个文件服务器，创建一个协议网桥代理。

要点: 在分布式系统上，只有 IBM MQ 管理员（和 mqm 组的成员）身份的用户才可以运行此命令。如果尝试以非 IBM MQ 管理员用户身份运行此命令，那么您将收到错误消息 BFGCL0502E: You are not authorized to perform the requested operation. 并且不会运行该命令。

在 z/OS 系统上，用户必须满足以下至少一个条件才能运行迁移命令：

- 是 mqm 组的成员（如果在系统上定义了 mqm 组）。
-  是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
-  在运行命令时，在 BFG_GROUP_NAME 环境变量中未设置任何值。

用途

使用 **fteCreateBridgeAgent** 命令可创建协议网桥代理。有关如何使用协议网桥的概述，请参阅第 275 页的『协议网桥』。该 **fteCreateBridgeAgent** 命令向您提供 MQSC 命令，您必须针对代理队列管理器运行这些命令以创建以下代理队列：

- `SYSTEM.FTE.AUTHADM1.agent_name`
- `SYSTEM.FTE.AUTHAGT1.agent_name`
- `SYSTEM.FTE.AUTHMON1.agent_name`
- `SYSTEM.FTE.AUTHOPS1.agent_name`
- `SYSTEM.FTE.AUTHSCH1.agent_name`
- `SYSTEM.FTE.AUTHTRN1.agent_name`
- `SYSTEM.FTE.COMMAND.agent_name`
- `SYSTEM.FTE.DATA.agent_name`
- `SYSTEM.FTE.EVENT.agent_name`
- `SYSTEM.FTE.REPLY.agent_name`
- `SYSTEM.FTE.STATE.agent_name`

这些队列是内部系统队列，除非您要删除该代理，否则不得修改、删除或读取这些队列中的消息。以下位置的文件中也提供了要运行的 MQSC 命令：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_create.mqsc`

如果您以后希望删除该代理，该命令还向您提供了 MQSC 命令，必须运行这些命令以清除并随后删除该代理使用的队列。MQSC 命令位于以下位置的文件中：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_delete.mqsc`

fteCreateBridgeAgent 命令在以下目录中创建 `ProtocolBridgeProperties.xml` XML 文件：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name`。用户必须手动创建 `ProtocolBridgeCredentials.xml` 文件。`ProtocolBridgeCredentials.xml` 文件允许您定义用户名和凭证信息以供协议网桥代理用于请求协议服务器对自己进行授权，而

`ProtocolBridgeProperties.xml` 文件允许您定义多个协议文件服务器，以便您可以传输到多个端点。

`MQ_INSTALLATION_PATH\mqft\samples\credentials\` 目录中存在一个样本

`ProtocolBridgeCredentials.xml`。有关更多信息，请参阅第 607 页的『协议网桥凭证文件格式』和第 611 页的『协议网桥属性文件格式』。如果运行 **fteCreateBridgeAgent** 命令并指定缺省协议文件服务器，那么该缺省服务器包含在 `ProtocolBridgeProperties.xml` 文件中且其主机名用作服务器名称。如果不指定缺省服务器，那么在 `ProtocolBridgeProperties.xml` 文件中没有条目；您必须手动添加至少一个服务器，然后才能进行传输。

IBM MQ Managed File Transfer 提供帮助您配置协议网桥代理的高级代理属性。与协议网桥相关的属性以 `protocol` 开头。在 `agent.properties` 文件中描述了这些属性。如果您在协议网桥中发现意外的行为，请复审这些 `protocol` 属性并确保您为系统正确设置了这些属性。

如果您看到来自 **fteCreateBridgeAgent** 命令的以下输出：

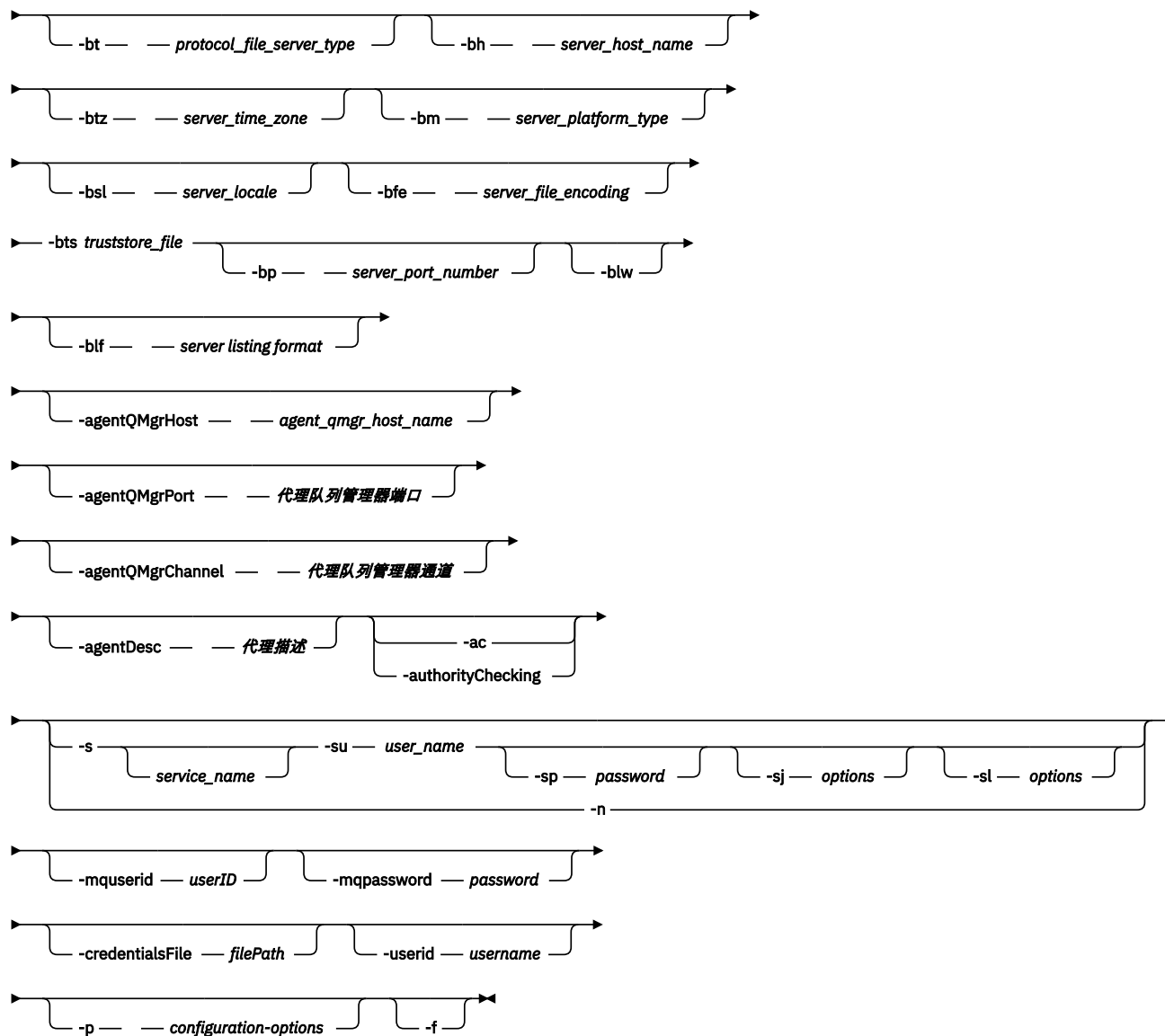
```
BFGMQ1007I: The coordination queue manager cannot be contacted or has refused a connection attempt.  
The WebSphere MQ reason code was 2058. The agent's presence will not be published.
```

它指示协调队列管理器无法访问，并提供 WebSphere MQ 原因码以说明原因。此参考消息可能指示协调队列管理器当前不可用或者您未正确定义配置。

语法

fteCreateBridgeAgent

► fteCreateBridgeAgent — -agentName 代理名称 -agentQMGr 代理队列管理器名称 →



参数

-agentName (agent_name)

必需。要创建的代理的名称。代理名称在其管理域中必须是唯一的。

有关命名代理的更多信息，请参阅[对象命名约定](#)。

-agentQMGr (agent_qmgr_name)

必需。代理队列管理器的名称。

-bt (protocol_file_server_type)

可选。指定您想要定义缺省协议文件服务器。指定以下某个选项：

FTP

标准 FTP 服务器

SFTP

SSH FTP 服务器

FTPS

由 SSL 或 TLS 提供保护的 FTP 服务器

如果未指定该参数，那么不会定义缺省协议服务器。

-bh (server_host_name)

只有当您同时使用 **-bt** 参数指定缺省协议文件服务器时，该参数才为必需参数。协议文件服务器的 IP 主机名或 IP 地址。

-btz (server_time_zone)

只有当您同时指定 **-bt** 参数时，该参数才为必需参数（仅限 FTP 和 FTPS 服务器）。协议文件服务器的时区。请按以下格式指定时区：区域/位置。例如：欧洲/伦敦。

您可以使用 **-htz** 参数来列出 **-btz** 的可能值。例如：fteCreateBridgeAgent -htz

-bm (server_platform)

只有当您同时使用 **-bt** 参数指定缺省协议文件服务器时，该参数才为必需参数。协议文件服务器的平台类型。指定以下某个选项：

UNIX

类属 UNIX 平台

Windows

类属 Windows 平台

-bsl (server_locale)

只有当您同时指定 **-bt** 参数时，该参数才为必需参数（仅限 FTP 和 FTPS 服务器）。协议文件服务器的语言环境。请按以下格式指定语言环境：*xx_XX*。例如：en_GB。

- *xx* 是 ISO 语言代码。有关有效值的列表，请参阅 [Codes for the Representation of Names of Languages](#)
- *XX* 是 ISO 国家或地区代码。有关有效值的列表，请参阅 [Country names and code elements](#)

-bfe (server_file_encoding)

只有当您同时使用 **-bt** 参数指定缺省协议文件服务器时，该参数才为必需参数。存储在协议文件服务器上的文件的字符编码格式。例如：UTF-8。

您可以使用 **-hcs** 参数来列出 **-bfe** 的可能值。例如：fteCreateBridgeAgent -hcs

-bts (truststore_file)

当您指定 **-bt** 参数时，该参数为必需参数（仅限 FTPS 服务器）。指定用于验证 FTPS 服务器所提供证书的信任密钥库的路径。

仅当您在 **-bt** 参数上还指定了 FTPS 选项时，才能指定 **-bts** 参数。

-bp (server_port)

可选。协议文件服务器连接到的 IP 端口。仅当协议文件服务器不使用该协议的缺省端口时，才指定该参数。如果未指定此参数，IBM MQ Managed File Transfer 将使用针对文件服务器的这种协议类型的缺省端口。

-blw

可选。将协议文件服务器定义为具有受限的写入能力。缺省情况下，协议网桥代理期望协议文件服务器允许删除文件、重命名文件以及打开文件进行附加写入。指定此参数以指示协议文件服务器不允许这些文件操作。相反，文件服务器只允许从文件读取和写入文件。如果指定此参数，那么任何传输在中断后可能都无法恢复，并且可能导致当前的文件传输失败。

-blf (server_listing_format)

可选，仅限 FTP 和 FTPS 服务器。定义从缺省协议文件服务器返回的已列出文件信息的服务器列表格式。选项如下：

UNIX

类属 UNIX 平台

Windows

类属 Windows 平台

要确定将要选择的格式，请使用 FTP 客户机程序，列出目录并选择最适合的格式。例如，UNIX 显示以下类型的列表：

```
-rwxr-xr-x 2 userid groupId 4096 2009-07-23 09:36 filename
```

Windows 显示以下类型的列表：

```
437,909 filename
```

缺省值为 UNIX，这是大多数服务器使用的格式。

-agentQMGrHost (agent_qmgr_host)

可选。代理队列管理器的主机名或 IP 地址。

-agentQMGrPort (agent_qmgr_port)

可选。用于客户机与代理队列管理器的连接的端口号。

-agentQMGrChannel (agent_qmgr_channel)

可选。用于连接到代理队列管理器的通道名称。

-agentDesc (agent_description)

可选。代理的描述，显示在 WebSphere MQ Explorer 中。

-ac 或 -authorityChecking

可选。此参数允许进行权限检查。如果指定该参数，那么代理将检查提交请求的用户是否有权执行请求的操作。有关更多信息，请参阅第 438 页的『[用户的 IBM MQ Managed File Transfer 操作权限](#)』。

-s (service_name)

可选（仅限 Windows）。指示代理程序将作为 Windows 服务运行。如果未指定服务名称，那么服务将命名为 mqmftAgent<AGENT><QMGR>，其中 <AGENT> 是代理名称，<QMGR> 是代理队列管理器名称。

服务的显示名称（显示在窗口 **服务** 窗口的 **名称** 列中）始终为 **WebSphere MQ Managed File Transfer 代理程序 <AGENT>@<QMGR>**。

-su (user_name)

可选（仅限 Windows）。如果该代理程序将作为 Windows 服务运行，那么该参数指定作为该服务运行身份的帐户的名称。要使用 Windows 域用户帐户来运行代理，请以 DomainName\UserName 格式指定该值。要使用本地内置域中的帐户来运行该服务，请以 UserName 格式指定该值。

使用 **-su** 参数指定的 Windows 用户帐户必须具有 **Log on as a service** 权限。有关如何授予该权限的信息，请参阅第 397 页的『[有关作为 Windows 服务运行代理或记录器的指南](#)』。

在指定 **-s** 时为必需项。等同于 **-serviceUser**。

-sp (password)

可选（仅限 Windows）。由 **-su** 或 **-serviceUser** 参数设置的用户帐户的密码。

该参数仅在指定 **-s** 时才有效。等同于 **-servicePassword**。如果在指定 **-s** 参数时不指定该参数，将生成警告消息。该消息提醒您必须使用 Windows“服务”工具来设置密码，然后该服务才会成功启动。

-sj (options)

可选（仅限 Windows）。当代理作为 Windows 服务启动时，以 -D 或 -X 格式来定义要传递到 JVM 的选项列表。这些选项使用井号 (#) 或分号 (;) 字符进行分隔。如果您必须嵌入任何 # 或分号 (;) 字符，请将其置于单引号内。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceJVMOptions**。

-sl (options)

可选（仅限 Windows）。设置 Windows 服务日志级别。有效选项是：error、info、warn、debug。缺省值为 info。如果 Windows 服务出现问题，那么该选项可能很有用。如果将其设置为 debug，那么会在服务日志文件中提供更详细的信息。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceLogLevel**。

-n

可选（仅限 Windows）。指示代理将作为正常进程运行。这与 **-s** 选项互斥。如果未指定 **-s** 参数和 **-n** 参数中的任何一个，那么会将代理程序配置为正常的 Windows 进程。

等同于 **-normal**。

-p (configuration-options)

可选。该参数确定用于创建代理程序的配置选项集。按照约定，请使用非缺省协调队列管理器的名称作为该参数的输入。然后，**fteCreateBridgeAgent** 命令使用与此非缺省协调队列管理器关联的属性文件集。

仅在您希望使用不同于缺省值的配置选项时，才指定可选的 **-p** 参数。如果未指定 **-p**，那么将使用 **installation.properties** 文件中定义的配置选项。请参阅第 116 页的『[分布式平台上的配置选项](#)』，以了解更多信息。

-f

可选。强制该命令覆盖现有配置。

-htz

可选。显示受支持时区（您可以将其用作 **-btz** 参数的输入）的列表。

-hcs

可选。显示受支持字符集（您可以将其用作 **-bfe** 参数的输入）的列表。

运行 **fteCreateBridgeAgent -hcs** 命令以列出 JVM 的已知代码页。通过外部源无法使用此信息，因为已知代码页在 JVM 之间有所变化。

-mquserid (userID)

可选。指定要向命令队列管理器进行认证的用户标识。

-mqpassword (password)

可选。指定要向命令队列管理器进行认证的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

-credentialsFile (filePath)

可选。现有或新凭证文件的完整文件路径，可以向该路径添加 IBM MQ 认证详细信息。

此命令支持向指定的 IBM MQ Managed File Transfer 凭证文件添加一组 IBM MQ 认证详细信息。启用 IBM MQ 连接认证之后，请使用此命令。如果要更新现有的详细信息，则必须使用 **-f** 强制参数。

-userid (username)

可选。用于关联凭证详细信息的用户标识。如果未指定用户标识，将向所有用户应用凭证详细信息。此外，还必须指定 **-credentialsFile** 参数。

-? or -h

可选。显示命令语法。

不推荐使用的参数

不推荐使用以下参数，并且在 IBM MQ V7.5 或者 IBM MQ Managed File Transfer V7.0.2 或更高版本上不支持这些参数。

-brd (reconnect_delay)

不推荐。可选。指定重新建立与协议文件服务器之间所丢失连接的相邻尝试之间的延迟时间段（以秒为单位）。缺省值为 10 秒。

-brr (reconnect_retries)

不推荐。可选。指定尝试重新建立与缺省协议文件服务器之间丢失的连接时的最大重试次数。达到此最大次数之后，当前文件传输将归类为失败。缺省值为 2。

示例

在此示例中，新协议网桥代理程序 ACCOUNTS1 是使用代理程序队列管理器 QM_ACCOUNTS 创建的，并且使用缺省协调队列管理器。ACCOUNTS1 连接到 FTP 服务器 accountshost.ibm.com。此 FTP 服务器使用时区“欧洲/柏林”、语言环境 de_DE 和文件编码 UTF-8 在 Windows 上运行。重新连接重试次数是 4：

```
fteCreateBridgeAgent -agentName ACCOUNTS1 -agentQMGr QM_ACCOUNTS -bt FTP  
-bh accountshost.ibm.com -bm WINDOWS -btz Europe/Berlin -bsl de_DE -bfe UTF8  
-agentQMGrHost myhost.ibm.com -agentQMGrPort 1415 -agentQMGrChannel CHANNEL1
```

在此示例中，新协议网桥代理 ACCOUNTS2 是使用代理队列管理器 QM_ACCOUNTS 创建的，并且使用缺省协调管理器。ACCOUNTS2 是在没有缺省协议文件服务器的情况下创建的。

```
fteCreateBridgeAgent -agentName ACCOUNTS2 -agentQMGr QM_ACCOUNTS
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

第 275 页的『协议网桥』

通过协议网桥，IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件，而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

fteCreateCDAgent (创建 Connect:Direct 网桥代理)

fteCreateCDAgent 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

要点: 在分布式系统上，只有 IBM MQ 管理员（和 mqm 组的成员）身份的用户才可以运行此命令。如果尝试以非 IBM MQ 管理员用户身份运行此命令，那么您将收到错误消息 BFGCL0502E: You are not authorized to perform the requested operation. 并且不会运行该命令。

在 z/OS 系统上，用户必须满足以下至少一个条件才能运行迁移命令：

- 是 mqm 组的成员（如果在系统上定义了 mqm 组）。
-  是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
-  在运行命令时，在 BFG_GROUP_NAME 环境变量中未设置任何值。

用途

使用 **fteCreateCDAgent** 命令可创建 Connect:Direct 网桥代理。该类型的代理专用于与 Connect:Direct 节点之间传输文件。有关更多信息，请参阅第 288 页的『Connect:Direct 网桥』。要了解有关 Connect:Direct 网桥支持的操作系统版本的更多详细信息，请参阅 Web 页面 [WebSphere MQ 系统需求](#)。

该命令向您提供 MQSC 命令，您必须针对代理队列管理器运行这些命令以创建以下代理队列：

- SYSTEM.FTE.AUTHADM1.agent_name
- SYSTEM.FTE.AUTHAGT1.agent_name

- SYSTEM.FTE.AUTHMON1.agent_name
- SYSTEM.FTE.AUTHOPS1.agent_name
- SYSTEM.FTE.AUTHSCH1.agent_name
- SYSTEM.FTE.AUTHTRN1.agent_name
- SYSTEM.FTE.COMMAND.agent_name
- SYSTEM.FTE.DATA.agent_name
- SYSTEM.FTE.EVENT.agent_name
- SYSTEM.FTE.REPLY.agent_name
- SYSTEM.FTE.STATE.agent_name

这些队列是内部系统队列，除非您要删除该代理，否则不得修改、删除或读取这些队列中的消息。以下位置的文件中也提供了要运行的 MQSC 命令：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_create.mqsc`。

如果稍后要删除代理，该命令还为您提供 MQSC 命令，必须运行这些 MQSC 命令才能清除然后删除属于代理的队列。MQSC 命令位于以下位置的文件中：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_delete.mqsc`。

IBM MQ Managed File Transfer 提供帮助您配置代理的高级代理属性。第 591 页的『[agent.properties 文件](#)』中描述了这些属性。

fteCreateCDAgent 命令在代理属性目录中创建两个 XML 文件。

`ConnectDirectNodeProperties.xml` (用于定义有关传输中远程节点的信息) 和

`ConnectDirectProcessDefinitions.xml` (用于指定由传输启动哪些用户定义的 Connect:Direct 进程)。

要定义 Connect:Direct 网桥代理用于连接到 Connect:Direct 节点的用户名和密码，必须手动创建 `ConnectDirectCredentials.xml` 文件。样本 XML 文件位于 `MQ_INSTALLATION_PATH/mqft/samples/credentials/` 中。有关更多信息和示例，请参阅第 618 页的『[Connect:Direct 凭证文件格式](#)』。

fteCreateCDAgent

► fteCreateCDAgent — -agentName — — 代理名称 — -agentQMGr — — 代理队列管理器名称 — ►

► -cdNode — — cd_node_name — — — -agentQMGrHost — — 代理队列管理器主机 — ►

► -agentQMGrPort — — 代理队列管理器端口 — ►

► -agentQMGrChannel — — 代理队列管理器通道 — ►

► -agentDesc — — 代理描述 — — -ac — — -authorityChecking — ►

► -p — — configuration_options — — -f — ►

► -cdNodeHost — — cd_node_host — — -cdNodePort — — cd_node_port — ►

► -cdTmpDir — — cd_tmp_dir — ►

► -s — — 服务名称 — -su — — 用户名 — -sp — — 密码 — -sj — — 选项 — -sl — — 选项 — -n — ►

► -mquserid — — userID — — -mqpassword — — password — ►

► -credentialsFile — — filePath — — -userid — — username — ►

参数

-agentName (*agent_name*)

必需。要创建的代理的名称。代理名称对其协调队列管理器必须唯一。

有关命名代理的更多信息，请参阅[对象命名约定](#)。

-agentQMGr (*agent_qmgr_name*)

必需。代理队列管理器的名称。

-cdNode *cd_node_name*

必需。用于从该代理向目标 Connect:Direct 节点传输消息的 Connect:Direct 节点的名称。该参数的值用于日志记录，而不是指定节点要连接到的 Connect:Direct 网桥代理。-cdNodeHost 和 -cdNodePort 的值指定作为 Connect:Direct 网桥的一部分的 Connect:Direct 节点。

-agentQMGrHost (*agent_qmgr_host*)

可选。代理队列管理器的主机名或 IP 地址。

-agentQMGrPort (*agent_qmgr_port*)

可选。用于客户机与代理队列管理器的连接的端口号。

-agentQMGrChannel (*agent_qmgr_channel*)

可选。用于连接到代理队列管理器的通道名称。

-agentDesc (*agent_description*)

可选。在 IBM MQ Explorer 中显示的代理的描述。

-ac 或 -authorityChecking

可选。此参数允许进行权限检查。如果指定该参数，那么代理将检查提交请求的用户是否有权执行请求的操作。有关更多信息，请参阅第 438 页的『[用户的 IBM MQ Managed File Transfer 操作权限](#)』。

-p (配置选项)

可选。该参数确定用于创建代理程序的配置选项集。按照约定，请使用非缺省协调队列管理器的名称作为该参数的输入。之后，**fteCreateCDAgent** 命令将使用与该非缺省协调队列管理器相关联的属性文件集。

仅在您希望使用不同于缺省值的配置选项时，才指定可选的 **-p** 参数。如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-f

可选。强制命令覆盖不匹配的现有参数。指定此参数不会强制替换现有 Windows 服务代理。

-cdNodeHost cd_node_host_name

可选。由 **-cdNode** 参数指定的 Connect:Direct 节点所在系统的主机名或 IP 地址。如果不指定 **-cdNodeHost** 参数，那么将使用本地系统的缺省主机名或 IP 地址。

在大多数情况下，Connect:Direct 节点与 Connect:Direct 网桥代理位于同一系统上。在这些情况下，该属性的缺省值（本地系统的 IP 地址）正确。如果系统具有多个 IP 地址，或您的 Connect:Direct 节点不在 Connect:Direct 网桥代理所在的系统上，并且它们的系统共享一个文件系统，那么使用该属性可指定正确的 Connect:Direct 节点主机名。

-cdNodePort cd_node_port_name

可选。客户机应用程序用于与 **-cdNode** 参数指定的节点进行通信的 Connect:Direct 节点的端口号。在 Connect:Direct 产品文档中，该端口称为 API 端口。如果不指定 **-cdNodePort** 参数，那么将假定缺省端口号为 1363。

-cdTmpDir cd_tmp_directory

可选。在将文件传输到目标 Connect:Direct 节点之前，该代理用于临时存储这些文件的目录。该参数指定临时存储文件的目录的完整路径。例如，如果将 **cdTmpDir** 设置为 /tmp，那么文件将临时存放在 /tmp 目录中。如果未指定 **-cdTmpDir** 参数，那么文件将临时存储在名为 **cdbridge-agent_name** 的目录中。该缺省目录是在 **java.io.tmpdir** 属性的值定义的位置中创建的。

Connect:Direct 网桥代理和 Connect:Direct 网桥节点必须能够使用相同路径名访问该参数指定的目录。请在计划安装 Connect:Direct 网桥时考虑这一点。如果可能，请在作为 Connect:Direct 网桥的组成部分的 Connect:Direct 节点所在的系统上创建该代理。如果代理和节点不在同一个系统上，那么该目录必须在共享文件系统上，并且可从两个系统使用相同路径名进行访问。有关受支持配置的更多信息，请参阅第 288 页的『[Connect:Direct 网桥](#)』。

注: 如果运行 **fteCleanAgent** 命令，那么将删除该目录中的所有文件。

-s (service_name)

可选（仅限 Windows）。指示代理将作为 Windows 服务运行，必须使用 Windows 管理员用户标识运行该命令。如果未指定服务名称，那么服务将命名为 **mqmftAgent<AGENT><QMGR>**，其中 **<AGENT>** 是代理名称，**<QMGR>** 是代理队列管理器名称。

服务的显示名称（显示在窗口 **服务** 窗口的 **名称** 列中）始终为 **IBM MQ Managed File Transfer 代理程序 <AGENT>@<QMGR>**。

-su (user_name)

可选（仅限 Windows）。如果该代理程序将作为 Windows 服务运行，那么该参数指定作为该服务运行身份的帐户的名称。要使用 Windows 域用户帐户来运行代理，请以 **DomainName\UserName** 格式指定该值。要使用本地内置域中的帐户来运行该服务，请以 **UserName** 格式指定该值。

使用 **-su** 参数指定的 Windows 用户帐户必须具有 **Log on as a service** 权限。有关如何授予该权限的信息，请参阅第 397 页的『[有关作为 Windows 服务运行代理或记录器的指南](#)』。

在指定 **-s** 时为必需项。等同于 **-serviceUser**。

-sp (password)

可选（仅限 Windows）。由 **-su** 或 **-serviceUser** 参数设置的用户帐户的密码。

该参数仅在指定 **-s** 时才有效。等同于 **-servicePassword**。如果在指定 **-s** 参数时不指定该参数，将生成警告消息。该消息提醒您必须使用 Windows“服务”工具来设置密码，然后该服务才会成功启动。

-sj (options)

可选（仅限 Windows）。当代理作为 Windows 服务启动时，以 **-D** 或 **-X** 格式来定义要传递到 JVM 的选项列表。这些选项使用井号 (#) 或分号 (;) 字符进行分隔。如果您必须嵌入任何 # 或分号 (;) 字符，请将其置于单引号内。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceJVMOptions**。

-sl (options)

可选（仅限 Windows）。设置 Windows 服务日志级别。有效选项是：error、info、warn、debug。缺省值为 info。如果 Windows 服务出现问题，那么该选项可能很有用。如果将其设置为 debug，那么会在服务日志文件中提供更详细的信息。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceLogLevel**。

-n

可选（仅限 Windows）。指示代理将作为正常进程运行。这与 **-s** 选项互斥。如果未指定 **-s** 参数和 **-n** 参数中的任何一个，那么会将代理程序配置为正常的 Windows 进程。

等同于 **-normal**。

-mquserid (userID)

可选。指定要向命令队列管理器进行认证的用户标识。

-mqpassword (password)

可选。指定要向命令队列管理器进行认证的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

-credentialsFile (filePath)

可选。现有或新凭证文件的完整文件路径，可以向该路径添加 IBM MQ 认证详细信息。

此命令支持向指定的 IBM MQ Managed File Transfer 凭证文件添加一组 IBM MQ 认证详细信息。启用 IBM MQ 连接认证之后，请使用此命令。如果要更新现有的详细信息，则必须使用 **-f** 强制参数。

-userid (username)

可选。用于关联凭证详细信息的用户标识。如果未指定用户标识，将向所有用户应用凭证详细信息。此外，还必须指定 **-credentialsFile** 参数。

示例

在该示例中，将通过代理队列管理器 QM_NEPTUNE 创建新的 Connect:Direct 网桥代理 CD_BRIDGE。该代理使用 Connect:Direct 节点 BRIDGE_NODE 来向其他 Connect:Direct 节点传输文件。BRIDGE_NODE 节点与代理位于同一个系统上，并将缺省端口用于客户机连接。传输到 Connect:Direct 或从传输的文件将临时存储在目录 /tmp/cd-bridge 中。

```
fteCreateCDAgent -agentName CD_BRIDGE -agentQMgr QM_NEPTUNE
                  -cdNode BRIDGE_NODE -cdTmpDir /tmp/cd-bridge
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

第 288 页的『Connect:Direct 网桥』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关任务

第 204 页的『配置 Connect:Direct 网桥』

配置 Connect:Direct 网桥，以在 IBM MQ Managed File Transfer 网络和 Connect:Direct 网络之间传输文件。Connect:Direct 网桥的组件包括 Connect:Direct 节点和专用于与该节点通信的 IBM MQ Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

第 290 页的『向 Connect:Direct 节点传输文件』

您可以使用 Connect:Direct 网桥从 IBM MQ Managed File Transfer 代理向 Connect:Direct 节点传输文件。通过将 Connect:Direct 网桥代理指定为目标代理并以 `connect_direct_node_name:file_path` 格式指定目标文件，指定 Connect:Direct 节点作为传输的目标。

第 291 页的『从 Connect:Direct 节点传输文件』

您可以使用 Connect:Direct 网桥，将文件从 Connect:Direct 节点传输到 IBM MQ Managed File Transfer 代理。通过将 Connect:Direct 网桥代理指定为源代理并以 `connect_direct_node_name:file_path` 格式指定源规范，可以将 Connect:Direct 节点指定为传输源。

fteCreateLogger（创建 IBM MQ Managed File Transfer 记录器）

使用 **fteCreateLogger** 命令可创建文件或数据库记录器。

要点: 在分布式系统上，只有 IBM MQ 管理员（和 mqm 组的成员）身份的用户才可以运行此命令。如果尝试以非 IBM MQ 管理员用户身份运行此命令，那么您将收到错误消息 BFGCL0502E: You are not authorized to perform the requested operation. 并且不会运行该命令。

在 z/OS 系统上，用户必须满足以下至少一个条件才能运行迁移命令：

- 是 mqm 组的成员（如果在系统上定义了 mqm 组）。
- **V8.0.0.6** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V8.0.0.6** 在运行命令时，在 BFG_GROUP_NAME 环境变量中未设置任何值。

用途

fteCreateLogger 命令向您提供 MQSC 命令，您必须针对记录器命令队列管理器运行这些命令以创建以下记录器队列：

- SYSTEM.FTE.LOG.CMD.*logger_name*
- SYSTEM.FTE.LOG.RJCT.*logger_name*

这些队列是内部系统队列，除非您要删除该记录器，否则不得修改、删除或读取这些队列中的消息。以下位置的文件中也提供了要运行的 MQSC 命令：

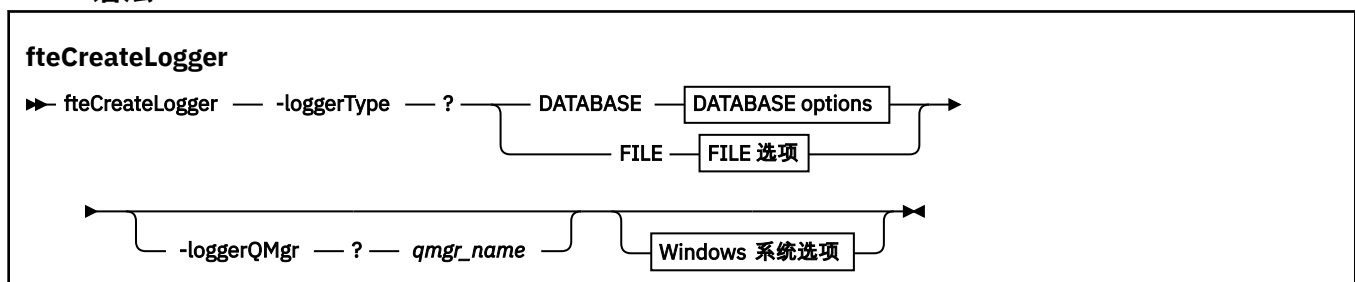
`MQ_DATA_PATH\mqft\config\coordination_qmgr\loggers\logger_name\logger_name_create.mqsc`

如果您以后想要删除记录器，请使用 **fteDeleteLogger** 命令。

IBM MQ Managed File Transfer 提供可以帮助您配置记录器的高级记录器属性。请参阅第 163 页的『IBM MQ Managed File Transfer 的记录器配置属性』。

如果要创建的记录器是数据库记录器，并且它不是连接到本地 Db2 数据库，那么您将需要手动创建 MQMFTCredentials.xml 文件。新的 `wmqfte.database.credentials.file` 属性用于指定凭证文件（如果需要）的位置。此凭证文件的样本位于 `MQ_INSTALLATION_PATH/mqft/samples/credentials/` 中。

语法



```

➤ -dbType — ? — database_type — -dbName — ? — database_name — -dbDriver — ? — ➤
    driver ————— ➤
    |
    | -dbLib — ? — path
    |
    |

```

-fileLoggerMode — ? — CIRCULAR
LINEAR — -fileSize — ? — size — -fileCount — ? →

▶ *number* ◀

▶ -s — ? — *service_name* — -su — ? — *user_name* — — -sp — ? — *password* —
 — -sj — ? — *options* — — -sl — ? — *options* —

```

graph LR
    subgraph Line1 [ ]
        direction LR
        A["-credentialsFile"] --- B["filePath"]
        C["-userid"] --- D["username"]
    end
    subgraph Line2 [ ]
        direction LR
        E["-p"] --- F["配置_选项"]
        G["-f"] --- H["logger_name"]
    end

```

-fileLoggerMode (mode)

-loggerType 是 FILE 时为必需项。指定用于存储受管文件传输信息的文件系统的类型。选项包括 LINEAR 和 CIRCULAR。

选项 LINEAR 表示文件记录器将信息写入文件，直至该文件达到由 -fileSize 定义的最大大小。达到最大大小时，文件记录器将启动新文件。先前已写入信息的文件将不会删除，从而允许将其保留为日志消息的历史记录。在该方式下运行时不会删除文件，从而将忽略 -fileCount，这是因为可创建的文件数量没有上限。因为在该方式下运行时没有上限，所以有必要跟踪日志文件所用的磁盘空间量以避免磁盘空间不足。

选项 CIRCULAR 表示文件记录器将信息写入文件，直至该文件达到由 -fileSize 定义的最大大小。达到最大大小时，文件记录器将启动新文件。在该方式下写入了信息的文件的最大数量由使用 -fileCount 定义的值控制。当达到文件的此最大数量时，文件记录器将删除首个文件并将其重新创建以用作当前活动文件。如果 -fileSize 中定义的值是固定大小的字节单位，那么以此方式使用的磁盘空间的上限将等于 fileSize 乘以 fileCount。如果 -fileSize 中定义的值是时间单位，那么最大大小将取决于这些时间段内系统中的日志消息吞吐量。

有关更多信息，请参阅第 163 页的『IBM MQ Managed File Transfer 的记录器配置属性』。

-fileSize (size)

-loggerType 是 FILE 时为必需项。允许一个日志文件增大到的最大大小。该值是大于零的正整数，后跟以下某个单位：KB、MB、GB、m（分钟）、h（小时）、d（天）、w（周）。例如：-fileSize 5MB（指定最大大小为 5MB），-fileSize 2d（指定最多两天的数据）。

-fileCount (number)

-loggerType 是 FILE 且 -fileLoggerMode 是 CIRCULAR 为必需项。要创建的日志文件的最大数目。数据量超过此数量的文件中可以存储的最大量时，将删除最旧的文件，以便日志文件数从不超过此参数中指定的值。

-s (service_name)

可选（仅限 Windows 系统）。指示记录器将作为 Windows 服务运行。如果未指定服务名称，那么服务将命名为 mqmftLogger<LOGGER><QMGR>，其中 <LOGGER> 是记录器名称，<QMGR> 是记录器队列管理器名称。

服务的显示名称（显示在窗口 服务 窗口的 名称 列中）始终为 WebSphere MQ Managed File Transfer 记录器 <LOGGER>@<QMGR>。

-su (user_name)

可选（仅限 Windows）。如果该记录器将作为 Windows 服务运行，那么该参数指定作为该服务运行身份的帐户的名称。要使用 Windows 域用户帐户来运行记录器，请以 DomainName\UserName 格式指定该值。要使用本地内置域中的帐户来运行该服务，请以 UserName 格式指定该值。

使用 -su 参数指定的 Windows 用户帐户必须具有 Log on as a service 权限。有关如何授予此权限的信息，请参阅第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』。

在指定 -s 时为必需项。等同于 -serviceUser。

-sp (password)

可选（仅限 Windows）。由 -su 或 -serviceUser 参数设置的用户帐户的密码。

该参数仅在指定 -s 时才有效。等同于 -servicePassword。如果在指定 -s 参数时不指定该参数，将生成警告消息。该消息提醒您必须使用 Windows“服务”工具来设置密码，然后该服务才会成功启动。

-sj (options)

可选（仅限 Windows）。记录器作为 Windows 服务启动时，以 -D 或 -X 格式定义要传递到 JVM 的选项的列表。这些选项使用井号 (#) 或分号 (;) 字符进行分隔。如果您必须嵌入任何 (#) 或分号 (;) 字符，请将其置于单引号内。

该参数仅在指定 -s 时才有效。等同于 -serviceJVMOptions。

-sl (options)

可选（仅限 Windows）。设置 Windows 服务日志级别。有效选项是：error、info、warn、debug。缺省值为 info。如果 Windows 服务出现问题，那么该选项可能很有用。如果将其设置为 debug，那么会在服务日志文件中提供更详细的信息。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceLogLevel**。

-p (configuration options)

可选。指定用于创建记录器的配置选项集。按照约定，此值是协调队列管理器的名称。如果未指定该参数，那么会使用缺省的配置选项集。

-f

可选。强制该命令覆盖现有配置。

(logger_name)

必需。要创建的记录器的名称。这将合并到 IBM MQ Managed File Transfer 队列名称中，因此必须仅包含字母、数字以及句点 (.) 和下划线字符 (_)。它还限制为最大长度为 28 个字符。

-credentialsFile (filePath)

可选。现有或新凭证文件的完整文件路径，可以向该路径添加 IBM MQ 认证详细信息。

此命令支持向指定的 IBM MQ Managed File Transfer 凭证文件添加一组 IBM MQ 认证详细信息。启用 IBM MQ 连接认证之后，请使用此命令。如果要更新现有的详细信息，则必须使用 **-f** 强制参数。

-userid (username)

可选。用于关联凭证详细信息的用户标识。如果未指定用户标识，将向所有用户应用凭证详细信息。此外，还必须指定 **-credentialsFile** 参数。

-? or -h

可选。显示命令语法。

示例

在本例中，将创建名为 filelogger1 的循环文件记录器。该文件记录器将创建最多 10 个文件，每个文件的大小为 10MB，总计使用最多 100MB 的磁盘空间：

```
fteCreateLogger -loggerType FILE -fileLoggerMode CIRCULAR -fileSize 10MB -fileCount 10
filelogger1
```

在此示例中，将创建称为 dblogger1 的数据库记录器。该数据库记录器连接到称为 FTEDB 的 Db2 数据库：

```
fteCreateLogger -loggerType DATABASE -dbName FTEDB -dbType DB2
-dbDriver "C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc4.jar" dblogger1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

[第 151 页的『配置 Managed File Transfer 记录器』](#)

相关参考

[第 548 页的『fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）』](#)

使用 **fteModifyLogger** 命令修改记录器，以便记录器能够作为 Windows 服务运行。只能在 Windows 平台上使用此命令，必须由作为 IBM MQ 管理员和 mqm 组成员的用户运行，并且必须首先使用

fteStopLogger 命令停止记录器。

[第 575 页的『fteStartLogger（启动记录器）』](#)

fteStartLogger 命令启动 IBM MQ Managed File Transfer 日志记录应用程序。

第 579 页的『[fteStopLogger](#) (停止记录器)』

fteStopLogger 命令停止记录器。

第 525 页的『[fteDeleteLogger](#) (删除 IBM MQ Managed File Transfer 记录器)』

使用 **fteDeleteLogger** 命令可删除 IBM MQ Managed File Transfer 记录器及其配置。可以保留或删除与该记录器关联的现有日志文件。

第 402 页的『[记录器错误处理和拒绝](#)』

记录器确定两种类型的错误：每条消息的错误和通用错误。

第 163 页的『[IBM MQ Managed File Transfer 的记录器配置属性](#)』

该记录器具有一组配置属性。在 `logger.properties` 文件中指定这些属性，该文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中。

fteCreateMonitor (新建资源监视器)

fteCreateMonitor 命令从命令行创建并启动新的资源监视器。您可以使用 WebSphere MQ Managed File Transfer 监视资源（例如，目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

用途

使用 **fteCreateMonitor** 命令可创建新的资源监视器并随后使用 IBM MQ Managed File Transfer 代理将其启动。例如，您可以按以下方式使用资源监视器：外部应用程序将一个或多个文件放在已知目录中，在处理完成时，外部应用程序将触发器文件放在受监视的目录中。然后，将检测到该触发器文件，并启动已定义的文件传输，从而将已知目录中的文件复制到目标代理。

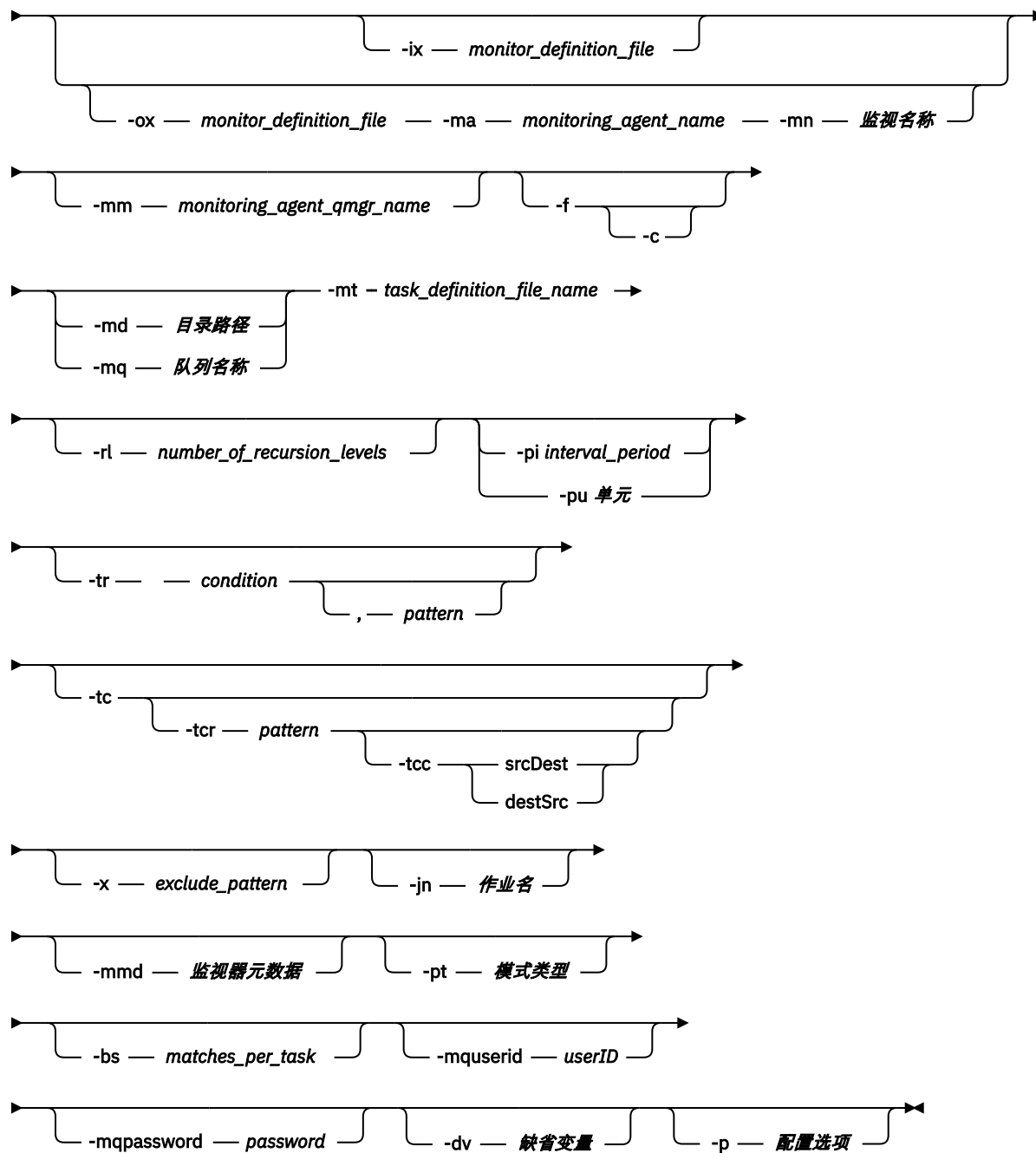
您可以使用 **-ox** 和 **-ix** 参数，分别将资源监视器配置导出到 XML 文件，以及从 XML 文件导入资源监视器配置。如果使用 **fteCreateMonitor** 命令导入此文件，将使用与 **fteCreateMonitor** 命令中提供的资源监视器相同的参数来创建新的资源监视器以导出到 XML 文件。您还可以使用 **fteListMonitors** 命令将资源监视器配置导出到 XML 文件。此外，您可以使用 **-f** 和 **-c** 参数动态覆盖监视器配置。

fteCreateMonitor 命令在协议网桥代理上不受支持。

语法

fteCreateMonitor

►► fteCreateMonitor ►►



参数

-ix (xml_filename)

可选。从 XML 文件导入资源监视器配置。

-ox (xml_filename)

可选。此参数必须与 -ma 和 -mn 参数一起指定。将资源监视器配置导出到 XML 文件。

-mn (monitor_name)

必需。您分配给此监视器的名称。监视器名称对于监视代理必须是唯一的。但是，您可以删除一个监视器，然后创建同名的监视器。

资源监视器名称的最大长度为 256 个字符。资源监视器名称不区分大小写。以小写或大小写混合形式输入的资源监视器名称将转换为大写。资源监视器名称不得包含星号 (*)、百分号 (%) 或问号 (?) 字符。

-ma (monitoring_agent_name)

必需。要执行资源监视的代理的名称。此监视代理必须是您要触发的监视器任务的源代理。

-mm (monitoring_agent_qmgr_name)

监视代理连接到的队列管理器的名称。由于监视代理和源代理必须相同，因此该队列管理器也是源代理队列管理器。

注: **fteCreateMonitor** 命令将连接到 IBM MQ Managed File Transfer 拓扑的命令队列管理器。如果命令队列管理器也是监视代理的代理队列管理器，那么该参数是可选参数。否则，该参数是必需参数。

-f

可选。使用该参数可覆盖资源监视器配置。例如，如果您选定的资源监视器名称在资源监视代理上已经存在，并且您希望将其更新，而不是将其删除并重新创建同名的监视器。使用此参数将使代理重新启动监视进程。

-c

可选。此参数清除更新的资源监视器的历史记录，这使资源监视器重新检查触发条件。只能将此参数与 **-f** 参数配合使用。

-md (directory_path)

可选。您要监视的目录路径的绝对名称。除非您要使用 **-ix** 或 **-ox** 参数，否则必须指定 **-md** 或 **-mq** 参数之一。

-mq (queue_name)

可选。要监视的队列的名称。此队列必须位于监视代理程序队列管理器上。除非您要使用 **-ix** 或 **-ox** 参数，否则必须指定 **-md** 或 **-mq** 参数之一。

-mt (task_definition_file_name)

可选。XML 文档的名称，该文档包含您要在满足触发条件时执行的任务定义。传输定义 XML 文档的路径必须位于从中运行 **fteCreateMonitor** 命令的本地文件系统中。除非您正在使用 **-ix** 或 **-ox** 参数，否则此参数将是必需参数。

您可以使用 **fteCreateTransfer** 命令上的 **-gt** 参数来生成包含文件传输请求的模板 XML 文档。监视器将传输模板用作其任务定义。

在 z/OS 上，您必须将任务定义文档存储在 z/OS UNIX 系统服务上的 UNIX 文件中。您无法将任务定义文档存储在 z/OS 顺序文件或 PDS 成员中。

在 IBM i 上，您必须将任务定义文档存储在集成文件系统中。

-rl (number_of_recursion_levels)

可选。根监视目录的监视递归级别，即，要向下深入的子目录级别数。例如，在类似以下示例的目录结构中，将 C:\wmqfte\monitor 设置为根监视目录：

```
C:\wmqfte\monitor
C:\wmqfte\monitor\reports
C:\wmqfte\monitor\reports\2009
C:\wmqfte\monitor\reports\2009\April
```

如果指定 **-rl 2**，那么 IBM MQ Managed File Transfer 仅向下搜索

C:\wmqfte\monitor\reports\2009 目录及其同代目录。将忽略

C:\wmqfte\monitor\reports\2009\April 目录。缺省情况下，递归设置为无。

-pi (interval_period)

可选。一个目录的每次监视之间的时间间隔。轮询时间间隔必须是正整数值。**-pi** 的缺省值是 1。

-pu (units)

可选。监视器轮询时间间隔的时间单位。如果指定 **-pu** 参数，还必须指定 **-pi** 参数。**-pu** 的缺省值为 **minutes**。指定以下某个选项：

秒

分钟

小时

days

-tr

可选。指定为执行定义的任务所必须满足的触发条件。如果不满足该条件，那么根据源代理，监视器任务（例如，文件传输）将不会启动。一个触发条件包含两个可选部分：条件和模式（以逗号分隔）。指定以下某种格式：

- `condition,pattern`

其中条件是以下某个值：

match

对于满足的每个触发器，将执行所定义的任务。**match** 是缺省值。

例如，如果匹配项为 `*.go`，并且存在文件 `LONDON.go` 和 `MANCHESTER.go`，那么将对 `LONDON.go` 执行任务，并对 `MANCHESTER.go` 执行另一个任务。

如果存在来自先前轮询的同一触发器文件（即，该文件尚未修改），那么此文件具有未满足的触发条件。即，匹配触发器文件必须是新的，并且自上次轮询以来必须已修改，才能执行定义的任务。

noMatch

受监视的目录中没有文件与该模式匹配。即，如果受监视的目录中的任何文件都不存在，即表示满足该条件。如果创建监视器时没有文件匹配触发条件，那么该监视器将立即启动，但是只有在找到文件匹配项并随即将其移除之后才会再次启动。

noSizeChange=n

目录中至少一个文件与该模式匹配，并且其文件大小在 n 个轮询时间间隔内没有更改。 n 的值是正整数。

fileSize>=size

目录中至少一个文件与该模式匹配，并且其文件大小至少大于或等于大小。值大小是一个整数与可选大小单位（B、KB、MB 或 GB）的组合。例如，`fileSize">"=10KB`。如果未指定大小单位，使用的缺省大小是字节。在所有操作系统上，您在命令行上指定 **filesize** 选项时必须将大于号 (>) 括在双引号内，如该示例中所示。

该模式是通配符或 Java 正则表达式格式的文件模式匹配序列。模式的缺省值是 `*`（即匹配任何文件），缺省格式是通配符格式。使用 **-pt** 来指定模式的格式。

例如，如果受监视的目录中存在后缀为 `.go` 的文件，就满足了以下触发条件。

```
-tr match,*.go
```

如果受监视的目录中不存在后缀为 `.stop` 的文件，就满足了以下触发条件。

```
-tr noMatch,*.stop
```

仅当还指定了 **-md** 参数时，才能指定 `condition,pattern`。

- `condition`

其中条件是以下某个值：

queueNotEmpty

受监视的队列不为空。即，如果在受监视的队列上存在任何 WebSphere MQ 消息，就满足该条件。将针对队列上的所有消息运行单个任务。

completeGroups

受监视的队列上存在完整组。即，如果受监视的队列上有任何 WebSphere MQ 消息组是完整的，就满足该条件。将针对队列上的每个完整组分别运行一个任务。

如果不在组内的单条消息放在队列上，那么会将其视为完整组，并且将针对此单条消息运行一个任务。

仅当还指定了 **-mq** 参数时，才能指定 *condition*。

对于您创建的每个监视器，只能指定一次 **-tr** 参数。

-tc

可选。指示触发的文件包含一个或多个用于生成传输请求的文件路径。触发器文件内容的缺省格式是每行上一个文件条目。Specify the file paths either as <source file path> or <source file path>,<destination file path>. This parameter is available only for directory monitor triggers **匹配** and **noSize 更改**。

-tcr (pattern)

可选。指定用于解析触发器文件的替换正则表达式。如果指定 **-tcr** 参数，还必须指定 **-tc** 参数。

设计模式以使用一个或两个捕获组完全解析每行条目。第一个组定义源文件路径，而第二个可选组则定义目标文件路径。这是缺省行为，可以使用 **-tcc** 参数进行更改。

有关更多信息和示例，请参阅 [第 246 页的『使用触发文件的内容』](#)。

-tcc

可选。定义正则表达式捕获组顺序。

srcDest

缺省值，第一个组是源文件路径，而第二个组则是目标文件路径。

destSrc

与 **srcDest** 顺序相反。第一个组是目标文件路径，而第二个组是源文件路径。确保 **destSrc** 的正则表达式具有两个捕获组。

如果指定 **-tcc** 参数，还必须指定 **-tcr** 参数。

-x (exclude_pattern)

可选。指定从触发器模式匹配中排除的文件。触发器模式由 **-tr** 参数指定。

该模式是通配符或 Java 正则表达式格式的文件模式匹配序列。缺省格式是通配符格式。使用 **-pt** 参数来指定模式的格式。

-jn (job name)

可选。指定作业名引用，即针对请求的用户定义的标识。

-mmd (monitor metadata)

可选。指定传递到监视器的出口点的用户定义的元数据。此参数可以采用以逗号分隔的一个或多个名称对。Each name pair consists of a <name>=<value>. You can use the **-mmd** parameter more than once in a command.

-pt (pattern_type)

可选。**-tr** 和 **-x** 参数所使用的模式类型。有效值包括：

wildcard

模式将作为通配符模式进行求值。星号 (*) 与零个或更多字符匹配，而问号 (?) 只与一个字符匹配。这是缺省值。

regex

这些模式将作为 Java 正则表达式进行求值。有关更多信息，请参阅 [第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』](#)。

-bs (matches_per_task)

可选。要包含在单个任务中的触发器匹配项的最大数量。例如，如果为 *matches_per_task* 指定了值 5，而在单个轮询时间间隔内出现九个触发器匹配项，那么将执行两个任务。第一个任务对应于触发器 1-5（含 1 和 5），第二个任务对应于触发器 6-9。*matches_per_task* 的缺省值为 1。

仅当提供给 **-mt** 参数的任务定义 XML 是 *managedTransfer* 时，才支持 **-bs** 参数。**-bs** 参数不支持 *managedCall*。

-mquserid (userID)

可选。指定要向命令队列管理器进行认证的用户标识。

-mqpassword (password)

可选。指定要向命令队列管理器进行认证的密码。您还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

-dv (default_variables)

可选。缺省变量的逗号分隔列表，监视队列时可以将这些变量用于变量替换。值的格式为键/值对。例如：

```
-dv size=medium,color=blue
```

有关变量替换的更多信息，请参阅第 239 页的『使用变量替换定制 MFT 任务』。仅当还指定了 **-mq** 参数时，才能指定 **-dv** 参数。

-? or -h

可选。显示命令语法。

-p (configuration_options)

可选。该参数确定用于取消传输的配置选项集。按照约定，请使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

示例

在此示例中，将使用监视代理程序 MYAGENT 创建称为 MYMONITOR 的新资源监视器。如果触发条件是目录 C:\wmqfte\monitors 中存在大于 5 MB 的文件，那么将启动文件 C:\templates\transfer_reports.xml 中定义的文件传输。MYAGENT 也是 C:\templates\transfer_reports.xml 中定义的文件传输的源代理：

```
fteCreateMonitor -ma MYAGENT -md C:\wmqfte\monitors -mn MYMONITOR -mt C:\templates\transfer_reports.xml -tr fileSize">"=5MB,*go
```

在此示例中，将创建名为 MONITOR1 的资源监视器（使用代理程序 AGENT1）以传输大于 5 MB 的文件，并将其导出到 XML 文件 monitor.xml。

```
fteCreateMonitor -ox monitor.xml -ma AGENT1 -mn MONITOR1 -mt task.xml -tr "fileSize>=5MB,*zip"
```

然后，将导入该 XML 文件并将其更改为排除大于 10MB 的任何文件。

```
fteCreateMonitor -ix monitor.xml -x "fileSize>=10MB,*zip" -f
```

在本例中，将创建使用代理 MYAGENT 且名为 MYMONITOR 的新资源监视器。

```
fteCreateMonitor -ma MYAGENT -md c:\wmqfte -mn MYMONITOR -mt c:\templates\transfer_reports.xml -tr "fileSize>=5MB,*go"
```

但是，触发器最初未正确设置为监视 `c:\wmqfte` 而不是 `c:\wmqfte\monitors`。将立即重新发出 **fteCreateMonitor** 请求，并更正监视器目录以及用于更新监视器的 **-f** (覆盖) 和 **-c** (清除历史记录) 参数。

```
fteCreateMonitor -ma MYAGENT -md c:\wmqfte\monitors -mn MYMONITOR -mt c:\templates\transfer_reports.xml -tr "fileSize>=5MB,*.go" -f -c
```

返回码

返回码	描述
0	命令成功完成。
1	命令结束但未成功。

相关概念

第 229 页的『资源监控』

您可以监视 IBM MQ Managed File Transfer 资源；例如，队列或目录。当满足针对该资源的条件时，资源监视器将启动任务，如文件传输。您可以使用 **fteCreateMonitor** 命令或 IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

第 239 页的『使用变量替换定制 MFT 任务』

在满足活动资源监视器的触发条件时，会调用已定义的任务。除了每次使用相同的目标代理或相同的目标文件名调用传输或命令任务之外，您还可以在运行时修改任务定义。可通过将变量名插入到任务定义 XML 中来实现此目的。在监视器确定已满足触发条件并且任务定义包含变量名时，将用变量值替换变量名，然后调用任务。

相关任务

第 233 页的『配置监视器任务以启动命令和脚本』

资源监视器的作用不只限于将执行文件传输作为其相关的任务。您还可以配置监视器，以从监视代理调用其他命令，包括可执行程序、Ant 脚本或 JCL 作业。要调用命令，请编辑监视任务定义 XML，以将一个或多个命令元素包含在对应的命令调用参数（例如，自变量和属性）中。

相关参考

第 534 页的『fteListMonitors（列出 IBM MQ Managed File Transfer 资源监视器）』

使用 **fteListMonitors** 命令可使用命令行列出 IBM MQ Managed File Transfer 网络中的所有现有资源监视器。

第 526 页的『fteDeleteMonitor（删除 IBM MQ Managed File Transfer 资源监视器）』

使用 **fteDeleteMonitor** 命令可通过命令行停止和删除现有 IBM MQ Managed File Transfer 资源监视器。针对资源监视代理发出此命令。

fteCreateTemplate（新建文件传输模板）

fteCreateTemplate 命令创建文件传输模板，您可以保留该模板以供将来使用。唯一的必需参数是 **-tn** (*template_name*) 参数。其他所有参数都是可选的，但如果指定源文件，还必须提供目标文件。类似地，如果指定目标文件，还必须指定源文件规范。

用途

使用 **fteCreateTemplate** 命令可创建文件传输模板来存储您的传输详细信息，以供您将来某天使用之需。使用传输模板来存储重复或复杂传输的公共文件传输设置。创建了传输模板之后，使用 WebSphere MQ Explorer 来提交该模板。您不能从命令行提交传输模板。

使用 **fteCreateTemplate** 命令创建的传输模板与使用 **fteCreateTransfer** 命令上的 **-gt** 参数创建的 XML 消息不同。您不能交换使用两种不同类型的模板。

您可以从可连接到 WebSphere MQ 网络的任何系统中运行 **fteCreateTemplate** 命令，然后路由到协调队列管理器。具体而言，要运行此命令，您必须已在该系统上安装了 IBM MQ Managed File Transfer，并将 IBM MQ Managed File Transfer 组件配置为与 WebSphere MQ 网络通信。

此命令使用 `command.properties` 文件来连接到 IBM MQ Managed File Transfer 拓扑的命令队列管理器。如果 `command.properties` 文件包含 **connectionQMGrHost** 属性，那么该命令将使用 CLIENT 传输连接到命令队列管理器。否则，该命令将使用 BINDINGS 传输连接到命令队列管理器。如果 `command.properties` 文件不存在，那么该命令将失败并生成以下错误：

```
BFGCL0491E: Missing or corrupt command.properties file. Use the fteSetupCommands
command to correct this condition. Additional information might be contained in this
exception BFGUB0009E: The following required property file is missing:
"MQ_DATA_PATH\mqft\coordination\coordination_qmgr_name\command.properties"
```

有关更多信息，请参阅 [command.properties](#) 文件。。

您可以为一个文件传输指定多个源文件，但是只能指定一个目标代理；不支持将一个文件传输到多个目标代理。但是，您可以将多个源文件传输到单个目标代理上的多个目标文件。

有关如何传输文件的指导信息，请参阅第 703 页的『文件传输准则』。

特殊字符

使用包含特殊字符的参数时，请谨慎操作，以免命令 `shell` 错误地解释这些字符。例如，命令 `shell` 可能会解释包含单引号的标准数据集名称和包含星号字符的源规范，而不是在传输请求中传递。要避免命令 `shell` 解释字符，请将整个参数括在双引号中（如第 496 页的『示例』中的最后两个示例所示），或者使用命令 `shell` 的转义序列将特殊字符转义。

相对路径

fteCreateTemplate 命令支持使用相对文件路径。在分布式系统和 z/OS UNIX 系统服务上，缺省情况下将路径视为相对于运行代理时所采用的用户的主目录。要更改与路径名称相对的目录，请在 `agent.properties` 文件中设置 `transferRoot` 属性。此文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` 目录中。将以下行添加到该文件：

```
transferRoot=directory_name
```

您必须对 Windows 路径进行转义或者使用 UNIX 格式书写。例如，将 `C:\TransferRoot` 指定为 `C:\TransferRoot` 或 `C:/TransferRoot`。

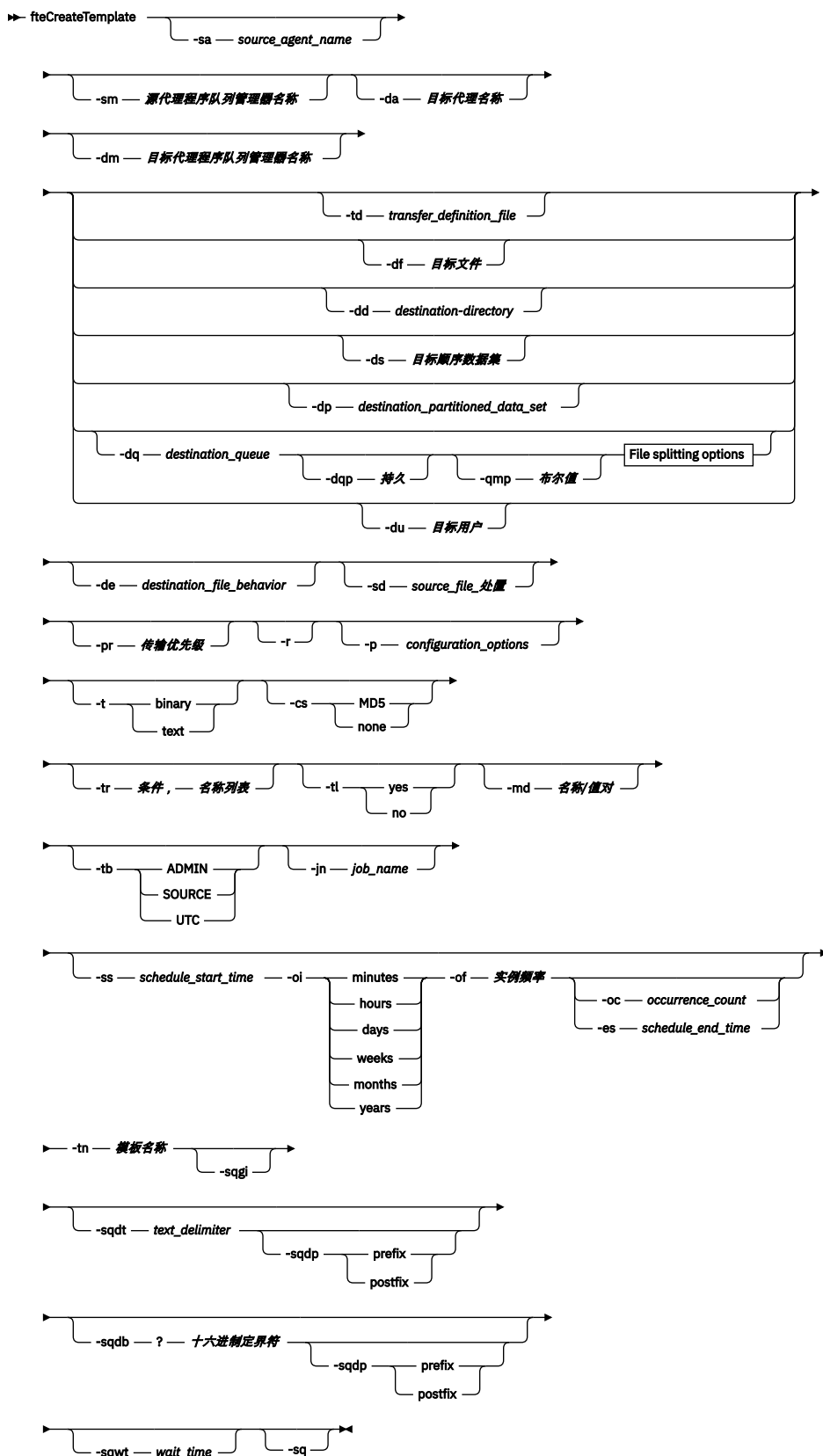
在 z/OS 上，缺省情况下，作为代理当前运行所用身份的用户名将作为高级限定符前缀添加到尚未标准化的数据集规范。例如：`//ABC.DEF`。要更改作为前缀添加到数据集名称的值，请在 `agent.properties` 文件中设置 `transferRootHLQ` 属性。此文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` 目录中。将以下行添加到该文件：

```
transferRootHLQ=prepend_value
```

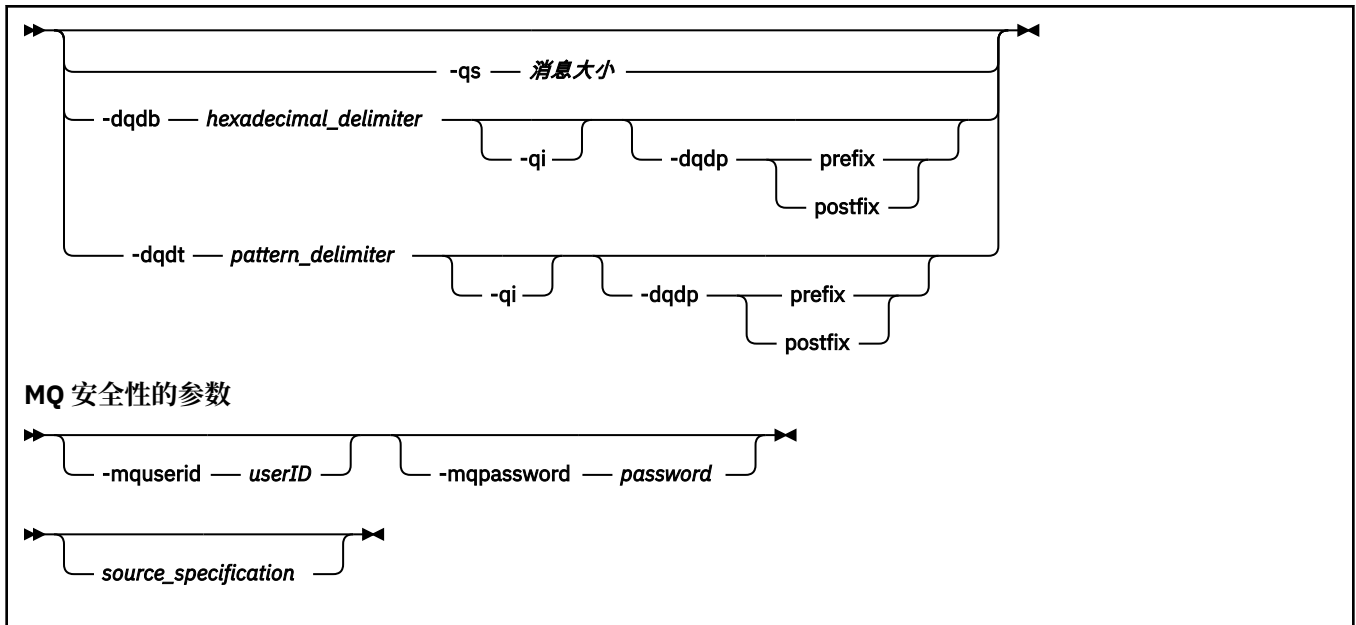
然而，对于涉及 z/OS 系统上的 Connect:Direct 节点的传输，数据集规范将解释为标准名称。不会向数据集名称添加任何高级限定符。

语法

fteCreateTemplate



File splitting options



参数

-sa source_agent_name

可选。从中传输源文件的代理程序的名称。如果创建模板时没有指定此代理程序名称，那么在使用模板时必须指定源代理程序名称。

-sm source_agent_qmgr_name

可选。源代理连接到的队列管理器的名称。

如果未指定 **-sm** 参数，那么使用的队列管理器将由使用的配置选项集确定（基于源代理名称）。如果使用这些选项无法确定队列管理器名称，那么传输模板创建失败。例如，如果找不到源代理程序的 `agent.properties` 文件，那么模板创建失败。

-da destination_agent_name

可选。文件传输到的代理程序的名称。如果创建模板时没有指定目标代理程序名称，那么在使用模板时必须指定目标代理程序名称。

-dm destination_agent_qmgr_name

可选。目标代理连接到的队列管理器的名称。

如果未指定 **-dm** 参数，那么使用的队列管理器将由使用的配置选项集确定（基于目标代理名称）。如果使用这些选项无法确定队列管理器名称，那么传输模板创建失败。例如，如果找不到目标代理程序的 `agent.properties` 文件，那么模板创建失败。

-td 传输定义文件

可选。XML 文档的名称，该文档定义传输的一个或多个源和目标文件规范。

必须指定 **-td**、**-df**、**-dd**、**-ds**、**-dq**、**-du** 和 **-dp** 参数之一。如果指定 **-td** 参数，那么不能指定源文件，或者指定 **-df**、**-dd**、**-ds**、**-dp**、**-dq**、**-du**、**-sd**、**-r**、**-de**、**-t** 或 **-cs** 参数。

fteCreateTemplate 命令会相对于当前目录查找传输定义文件。如果无法使用相对路径注释来指定传输定义文件的位置，请使用传输定义文件的标准路径和文件名代替。

在 z/OS 上，您必须将传输定义文件存储在 z/OS UNIX 系统服务上的 UNIX 文件中。您无法在 z/OS 顺序文件或 PDS 成员中存储传输定义文件。

在 IBM i 上，您必须将传输定义文件存储在集成文件系统中。

有关更多信息，请参阅[使用传输定义文件](#)。

-df 目标文件

可选。目标文件的名称。指定目标代理程序运行所在的系统上有效的文件名。

如果目标代理是 Connect:Direct 网桥代理，那么将以 `connect_direct_node_name:file_path` 格式指定目标文件。Connect:Direct 网桥代理仅接受以此格式指定的文件路径。如果目标代理是 Connect:Direct 网桥代理，并且目标是 PDS 成员，那么还必须使用覆盖值指定 **-de** 参数。

必须指定 **-td**、**-df**、**-dd**、**-ds**、**-dq**、**-du** 和 **-dp** 参数之一。如果指定 **-df** 参数，那么不能指定 **-td**、**-dd**、**-dp**、**-dq**、**-du** 或 **-ds** 参数，因为这些参数是互斥的。

-dd 目标目录

可选。要将文件传输到的目录的名称。指定目标代理程序运行所在的系统上有效的目录名称。

如果目标代理是 Connect:Direct 网桥代理，那么将以 `connect_direct_node_name:directory_path` 格式指定目标目录。如果目标代理是 Connect:Direct 网桥代理，而目标是 PDS，那么还必须使用覆盖值指定 **-de** 参数。

必须指定 **-td**、**-df**、**-dd**、**-ds**、**-dq**、**-du** 和 **-dp** 参数之一。如果指定 **-dd** 参数，那么不能指定 **-td**、**-df**、**-dp**、**-dq**、**-du** 或 **-ds** 参数，因为这些参数是互斥的。

-ds 目标顺序数据集

仅适用于 z/OS。可选。要将文件传输到的顺序数据集或 PDS 成员的名称。指定连续数据集名称或分区数据集成员。

必须指定 **-td**、**-df**、**-dd**、**-ds**、**-dq**、**-du** 和 **-dp** 参数之一。如果指定 **-ds** 参数，那么不能指定 **-td**、**-dd**、**-df**、**-dq**、**-du** 或 **-dp** 参数，因为这些参数是互斥的。

数据集名称的语法如下所示：

```
//data_set_name{;attribute;...;attribute}
```

或

```
//pds_data_set_name(member_name){;attribute;...;attribute}
```

即，数据集名称说明符，以 // 为前缀，并（可选）后跟以分号分隔的若干属性。

如果数据集位于 Connect:Direct 节点上，那么必须为数据集名称前缀节点名称。例如：

```
CD_NODE1:/'OBJECT.LIB';RECFM(F,B);BLKSIZE(800);LRECL(80)
```

如果目标代理是 Connect:Direct 网桥代理，并且目标是 PDS 成员，那么还必须使用覆盖值指定 **-de** 参数。有关数据集传输目标或源 Connect:Direct 节点的更多信息，请参阅第 706 页的『与 Connect:Direct 节点之间传输数据集』。

对于仅涉及 IBM MQ Managed File Transfer 代理的传输，如果使用单引号字符将数据集名称部分引起来，那么其指定标准数据集名称。如果未使用单引号字符将数据集名称引起来，那么系统针对目标代理程序添加缺省高级限定符（如果未设置 transferRootHLQ，那么这是 transferRootHLQ 代理程序属性的值或运行代理程序的用户标识）。

注：然而，对于涉及 z/OS 系统上的 Connect:Direct 节点的传输，数据集规范将解释为标准名称。不会向数据集名称添加任何高级限定符。即使是使用单引号字符将数据集名称引起来时也是如此。

数据集属性用于创建数据集或者确保现有数据集兼容。数据集属性的规范采用适用于 BPXWDYN 的格式（请参阅 [Requesting dynamic allocation](#) 以获取更多信息）。代理要创建目标数据集时，将自动指定以下 BPXWDYN 属性：DSN(`data_set_name`) NEW CATALOG MSG(`numeric_file_descriptor`)，其中 `numeric_file_descriptor` 是 IBM MQ Managed File Transfer 生成的文件描述符。对于数据集之间的传输，将为新目标数据集选择来自源的属性：RECFM、LRECL 和 BLKSIZE。请注意，IBM MQ Managed File Transfer 没有设置新目标数据集的 SPACE 设置，而将使用系统缺省值。因此，建议您在将要创建新数据集时指定 SPACE 属性。您可以在 `agent.properties` 文件中使用 **bpxwdynAllocAdditionalProperties** 属性来设置适用于所有传输的 BPXWDYN 选项。有关更多信息，请参阅第 591 页的『`agent.properties` 文件』。

使用 **fteCreateTemplate** 命令、**fteCreateTransfer** 命令或 `agent.properties` 文件中的 **bpxwdynAllocAdditionalOptions** 属性时，不能指定某些 BPXWDYN 选项。有关这些属性的列表，请参阅第 712 页的『不能与 IBM MQ Managed File Transfer 一起使用的 BPXWDYN 属性』。

将文件或数据集传输到磁带时，磁带上的任何现有数据集都将被替换。新数据集的属性将根据传输定义中传递的属性进行设置。如果未指定任何属性，那么新数据集的属性将设置为与源数据集相同，或者，如果源数据集是文件，那么这些属性将设置为缺省值。将忽略现有磁带数据集的属性。

当目标代理为协议网桥代理时，不支持 **-ds** 参数。

-dp 目标分区数据集

仅适用于 z/OS。可选。要将文件传输到的目标 PDS 的名称。指定分区数据集名称。如果创建 PDS 作为传输的结果，那么缺省情况下将创建该 PDS 作为 PDSE。您可以通过指定 `DSNTYPE=PDS` 来覆盖缺省值。

必须指定 **-td**、**-df**、**-dd**、**-ds**、**-dq**、**-du** 和 **-dp** 参数之一。如果指定 **-dp** 参数，那么不能指定 **-td**、**-dd**、**-df**、**-dq**、**-du** 或 **-ds** 参数，因为这些参数是互斥的。

PDS 数据集名称的语法如下所示：

```
//pds_data_set_name{;attribute;...;attribute}
```

数据集名称的语法与 **-ds** (*destination_sequential_data_set*) 参数描述的语法相同。用于指定位于 Connect:Direct 节点上的数据集的所有语法详细信息也适用于 **-dp** 参数。如果目标代理是 Connect:Direct 网桥代理，那么还必须指定值为 `overwrite` 的 **-de** 参数。

当目标代理为协议网桥代理时，不支持 **-dp** 参数。

-du 目标用户

可选。要将文件传输到其目标文件空间的用户的名称。有关文件空间的更多信息，请参阅第 339 页的『文件空间』。

必须指定 **-td**、**-df**、**-dd**、**-ds**、**-dp**、**-du** 和 **-dq** 参数之一。如果指定 **-du** 参数，那么不能指定 **-td**、**-dd**、**-df**、**-dp**、**-dq** 或 **-ds** 参数，因为这些参数是互斥的。

当目标代理为协议网桥代理或 Connect:Direct 网桥代理时，不支持 **-du** 参数。

-dq 目标队列

可选。要将文件传输到的目标队列的名称。您可以选择使用 `QUEUE@QUEUEMANAGER` 格式将队列管理器名称包含在此规范中。如果未指定队列管理器名称，并且未将 `enableClusterQueueInputOutput` 代理属性设置为 `true`，那么将使用目标代理队列管理器名称。如果已将 `enableClusterQueueInputOutput` 代理属性设置为 `true`，那么目标代理将使用标准 WebSphere MQ 解决过程来确定放置队列的位置。必须指定队列管理器上已存在的有效队列名称。

必须指定 **-td**、**-df**、**-dd**、**-ds**、**-dp**、**-du** 和 **-dq** 参数之一。如果指定 **-dq** 参数，那么不能指定 **-td**、**-dd**、**-df**、**-dp**、**-du** 或 **-ds** 参数，因为这些参数是互斥的。

当目标代理为协议网桥代理或 Connect:Direct 网桥代理时，或者在源规范为队列时，不支持 **-dq** 参数。

-dqp persistent

可选。指定写入目标队列的消息是否持久。有效选项如下：

true

将持久消息写入目标队列。这是缺省值。

false

将非持久性消息写入目标队列。

qdef

从目标队列的 `DefPersistence` 属性获取持久性值。

仅当同时指定了 **-dq** 参数时，才能指定 **-dqp** 参数。

-qmp 布尔值

可选。指定传输写入目标队列的第一条消息是否已设置 WebSphere MQ 消息属性。有效选项如下：

true

在传输创建的第一条消息上设置消息属性。

false

不在传输创建的第一条消息上设置消息属性。这是缺省值。

仅当同时指定了 **-dq** 参数时，才能指定 **-qmp** 参数。有关更多信息，请参阅第 739 页的『写入到目标队列的消息上设置的 IBM MQ 消息属性』。

-qs 消息大小

可选。指定是否将文件分割为多条固定长度的消息。所有消息都具有相同的 WebSphere MQ 组标识；组中的最后一条消息设置了 WebSphere MQ LAST_MSG_IN_GROUP 标志。消息的大小由 *message_size* 的值指定。The format of 消息大小 is <长度><单位>, where 长度 is a positive integer value and 单位 is one of the following values:

B

字节。允许的最小值是目标消息的代码页的最大“每个字符的字节数”值的两倍。

K

等于 1024 字节。

M

这等于 1048576 字节。

如果为 **-t** 参数指定值 *text* 且该文件采用双字节字符集或多字节字符集，那么沿着最接近指定消息大小的字符边界将该文件分割为消息。

仅当同时指定了 **-dq** 参数时，才能指定 **-qs** 参数。您只能指定 **-qs**、**-dqdb** 和 **-dqdt** 参数之一。

-dqdb 十六进制定界符

可选。指定将二进制文件分割为多条消息时使用十六进制定界符。所有消息都具有相同的 WebSphere MQ 组标识；组中的最后一条消息设置了 WebSphere MQ LAST_MSG_IN_GROUP 标志。指定十六进制字节作为定界符的格式为 *xNN*，其中，*N* 是 0-9 或 a-f 范围内的一个字符。您可以通过指定十六进制字节的逗号分隔列表（例如：x3e,x20,x20,xbf），将一系列十六进制字节指定为定界符。

仅当同时指定了 **-dq** 参数且传输采用二进制方式时，才能指定 **-dqdb** 参数。您只能指定 **-qs**、**-dqdb** 和 **-dqdt** 参数之一。

-dqdt pattern

可选。指定将文本文件分割为多条消息时要使用的正则表达式。所有消息都具有相同的 WebSphere MQ 组标识；组中的最后一条消息设置了 WebSphere MQ LAST_MSG_IN_GROUP 标志。用于将正则表达式指定为定界符的格式是括在括号 (*regular_expression*) 中的正则表达式。此参数的值求值为 aJava 正则表达式。有关更多信息，请参阅第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』。

缺省情况下，目标代理将正则表达式可匹配的字符串的长度限制为 5 个字符。您可使用 **maxDelimiterMatchLength** 代理属性来更改此行为。有关更多信息，请参阅第 591 页的『高级代理属性』。

仅当同时指定了 **-dq** 参数且为 **-t** 参数指定了值 *text* 时，才能指定 **-dqdt** 参数。您只能指定 **-qs**、**-dqdb** 和 **-dqdt** 参数之一。

-dqdp

可选。指定在分割文件时目标文本和二进制定界符的预期位置。仅当同时指定了 **-dqdt** 和 **-dqdb** 参数之一时，才能指定 **-dqdp** 参数。

指定以下某个选项：

前缀

定界符原应位于每行的开头。

postfix

预期定界符在每行的末尾。这是缺省选项。

-qi

可选。指定用于将文件分割为多条消息的定界符是否要包括在消息中。如果指定了 **-qi**，那么在包含定界符之前的文件数据的消息中，将在末尾包括定界符。缺省情况下，消息中不包含定界符。

仅当同时指定了 **-dqdt** 和 **-dqdb** 参数之一时，才能指定 **-qi** 参数。

-de 目标文件行为

可选。指定在目标系统上已存在目标文件时执行的操作。有效选项如下：

Error

报告错误并且不传输文件。这是缺省值。

覆盖

覆盖现有目标文件。

如果指定 **-de** 参数，那么不能指定 **-td** 参数，因为这些参数是互斥的。

-sd 源文件处置

可选。指定在将源文件成功传输到其目标时要对源文件执行的操作。有效选项如下：

leave

源文件将保持不变。这是缺省值。

delete

在成功传输源文件后，将从源系统删除源文件。

在 z/OS 上，如果源是磁带数据集并且指定了 **delete** 选项，那么将重新安装磁带以删除数据集。该行为是由于系统环境的行为。

如果源是队列，并且指定 **leave** 选项，那么命令返回一个错误且不请求传输。

如果源代理是 Connect:Direct 网桥代理，并且指定了 **delete** 选项，那么行为不同于常用的源处置行为。将出现以下某种情况：

- 如果 Connect:Direct 使用 IBM MQ Managed File Transfer 生成的进程从源移动文件或数据集，那么指定 **delete** 选项将导致传输失败。要指定删除源文件，您必须提交用户定义的 Connect:Direct 进程。有关更多信息，请参阅第 297 页的『通过文件传输请求提交用户定义的 Connect:Direct 进程』。
- 如果 Connect:Direct 使用用户定义的进程从源移动文件或数据集，那么该参数将通过 **%FTEFDISP** 内置符号变量传递给此进程。用户定义的进程确定是否删除源。传输返回的结果取决于用户定义的进程返回的结果。

如果指定 **-sd** 参数，那么不能指定 **-td** 参数，因为这些参数是互斥的。然而，您可以在传输定义文件中指定源处置行为。

-pr 传输优先级

可选。指定传输的优先级。优先级为 0-9 之间的一个值，其中 0 为最低优先级。缺省优先级为 0，并且缺省情况下，传输使用源代理的优先级。

该值与 WebSphere MQ 使用的消息优先级值相匹配，请参阅 [Getting messages from a queue: priority](#) 以获取更多信息。文件传输数据的消息通信缺省为优先级别 0，从而允许 WebSphere MQ 消息通信获得优先。

-p configuration_options

可选。此参数确定用于创建传输模板的配置选项集。使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-r

可选。当 *source_specification* 包含通配符时，那么将以递归方式传输子目录中的文件。如果以通配符作为源规范提供给 IBM MQ Managed File Transfer，那么仅当您指定了 **-r** 参数时才会传输与通配符匹配

的任何目录。在 *source_specification* 与子目录匹配时，始终传输该目录及其子目录中的所有文件（包含隐藏的文件）。

有关 IBM MQ Managed File Transfer 如何处理通配符的更多信息，请参阅[使用通配符](#)

如果指定 **-r** 参数，那么不能指定 **-td** 参数，因为这些参数是互斥的。然而，您可以在传输定义文件中指定递归行为。

-t

可选。指定文件传输的类型：二进制方式或文本方式。

二进制

文件中的数据将在不经任何转换的情况下进行传输。这是缺省值。

文本

将转换文件的代码页和行尾字符。执行的确切转换取决于源代理程序和目标代理程序的操作系统。

例如，从 Windows 传输到 z/OS 的文件将其代码页从 ASCII 转换为 EBCDIC。在文件从 ASCII 转换为 EBCDIC 时，行尾字符将从 ASCII 回车符 (CR) 和换行 (LF) 字符对转换为 EBCDIC 换行 (NL) 字符。

有关如何传输 z/OS 数据集的更多信息，请参阅[在 z/OS 与分布式系统之间传输文件和数据集以及在数据集之间传输](#)。

如果指定 **-t** 参数，那么不能指定 **-td** 参数，因为这些参数是互斥的。然而，您可以在传输定义文件中指定传输方式行为。

-cs

可选。指定是否针对文件传输数据运行校验和算法，以检查所传输文件的完整性。指定以下某个选项：

MD5

计算数据的 MD5 校验和。针对源和目标文件生成的校验和将写入传输日志以供验证之用。缺省情况下，IBM MQ Managed File Transfer 计算所有文件传输的 MD5 校验和。

none

不会为文件传输数据计算 MD5 校验和。传输日志记录校验和设置为 none，并且校验和值为空。例如：

```
<checksum method="none"></checksum>
```

如果使用 none 选项，那么可能会提高文件传输性能，这取决于环境。然而，选择该选项意味着不会验证源或目标文件。

如果指定 **-cs** 参数，那么不能指定 **-td** 参数，因为这些参数是互斥的。但是，您可以在传输定义文件中指定校验和行为。

-tr

可选。指定为执行此文件传输所必须满足的条件。如果不满足该条件，那么根据源代理，将废弃该文件传输，不会执行任何传输。指定以下格式：

```
condition,namelist
```

其中条件是以下某个值：

file=exist

名称列表中至少有一个文件存在。即，如果存在名称列表中的任何文件，那么就满足条件。

file!=exist

名称列表中至少有一个文件不存在。即，如果名称列表中有任何文件不存在，那么满足条件。

filesize>=size

至少存在名称列表中的一个文件，并且最小大小为 *size* 指定的大小。值大小是整数，带有可选大小单位（KB、MB 或 GB）。例如，**filesize">"=10KB**。如果未指定大小单位，大小将假定以字节

为单位。在所有操作系统上，您在命令行上指定 **filesize** 选项时必须将大于号 (>) 括在双引号内，如该示例中所示。

其中名称列表是位于源系统上的文件名的逗号分隔列表。根据操作系统，如果您要在名称列表中使用包含空格的路径名或文件名，可能必须将路径名和文件名括在双引号中。

您可以多次使用 **-tr** 参数来指定多个触发条件。然而，在此情况下，必须满足每个单独的触发条件才能进行文件传输。

注：要持续监视资源至满足触发条件，建议您使用[资源监视](#)。您可以使用 **fteCreateMonitor** 命令来创建资源监视器。

在以下示例中，文件 **file1.doc** 将从 AGENT1 传输到 AGENT2，条件是文件 **A.txt** 和/或文件 **B.txt** 位于 AGENT1 上，而且文件 **A.txt** 和/或文件 **B.txt** 不小于 1GB：

```
fteCreateTemplate -tn JUPITER_AGENT_TRIGGER_TEST_TEMPLATE -sa AGENT1 -sm QM_JUPITER -da AGENT2 -dm
QM_NEPTUNE
-tr file=exist,C:\export\A.txt,C:\export\B.txt
-tr filesize">"=1GB,C:\export\A.txt,C:\export\B.txt
-df C:\import\file1.doc C:\export\file1.doc
```

您可以将触发参数与调度参数相结合。如果同时指定这两种类型的参数，那么触发条件将应用于调度参数创建的文件传输。

-tl

可选。指定是否记录触发器失败情况。指定以下某个选项：

yes

为失败的触发传输创建日志条目。这是缺省行为，即使未指定 **-tl** 参数。

否

不为失败的触发传输创建日志条目。

-md

可选。指定传递到代理出口点的用户定义的元数据。**-md** 参数可以采用以逗号分隔的一个或多个名称/值对。Each name pair consists of <name>=<value>. You can use the **-md** parameter more than once in a command.

-tb

可选。指定想要用于已安排的文件传输的时基。即，想要使用系统时间还是全球标准时间 (UTC)。该参数只能与 **-ss** 参数一起使用。指定以下某个选项：

admin

用于调度的传输的开始时间和结束时间基于管理员使用的系统的时间和日期。这是缺省值。

源

用于调度的传输的开始时间和结束时间基于源代理所在系统的日期和时间。

UTC

用于调度的传输的开始时间和结束时间基于全球标准时间 (UTC)。

-jn *job_name*

可选。传输开始时添加到日志消息的用户定义的作业名标识。

-ss *schedule_start_time*

可选。指定您希望调度的传输开始执行的时间和日期。请使用以下某种格式来指定时间和日期。使用 24 小时制指定时间：

```
yyyy-MM-ddThh:mm
```

```
hh:mm
```

如果不存在可能影响传输的问题，那么调度的文件传输将在调度开始时间后一分钟内开始。例如，您的网络或代理可能存在阻止调度传输开始的问题。

-oi

可选。指定调度的传输出现的时间间隔。该参数只能与 **-ss** 参数一起使用。指定以下某个选项：

分钟

小时

days

weeks

months

years

-of occurrence_frequency

可选。指定调度的传输出现的频率。例如，每 **5** 周或每 **2** 个月。该参数只能与 **-oi** 和 **-ss** 参数一起指定。如果未指定该参数，那么将使用缺省值 **1**。

-oc occurrence_count

可选。指定您希望此调度的传输发生的次数。达到发生计数之后，将删除调度的传输。

该参数只能与 **-oi** 和 **-ss** 参数一起指定。

如果指定 **-oc** 参数，那么不能指定 **-es** 参数，因为这些参数是互斥的。

您可以同时省略 **-oc** 和 **-es** 参数以创建无限重复的传输。

-es schedule_end_time

可选。重复的调度传输结束的日期和时间。

该参数只能与 **-oi** 和 **-ss** 参数一起指定。

如果指定 **-es** 参数，那么不能指定 **-oc** 参数，因为这些参数是互斥的。

您可以同时省略 **-es** 和 **-oc** 参数以创建无限重复的传输。

使用以下某种格式来指定结束时间和日期。使用 **24** 小时制指定时间：

```
yyyy-MM-ddThh:mm
```

```
hh:mm
```

-tn template_name

必需。要创建的模板的名称。请使用描述性字符串，这样您可以在将来某天为传输选择正确的模板。对此字符串的长度没有特定限制，但是请注意，在某些用户界面上可能无法正确显示过长的名称。

请勿创建同名的多个模板。

-sqgi

可选。指定按 WebSphere MQ 组标识分组的消息。第一个完整组将写入目标文件。如果不指定此参数，那么源队列上的所有消息将写入目标文件。

仅当同时指定了 **-sq** 参数时，才能指定 **-sqgi** 参数。

-sqdt text_delimiter

可选。指定一个文本序列，以在将多条消息追加到文本文件时作为定界符插入。可以在定界符中包含字符串字面值的 Java 转义序列。例如，**-sqdt \u007d\n**。

仅当同时指定了 **-sq** 参数且为 **-t** 参数指定了值 **text** 时，才能指定 **-sqdt** 参数。

-sqdb hexadecimal_delimiter

可选。指定一个或多个字节的值，以在将多条消息追加到二进制文件时作为定界符插入。每个值都必须指定为两个十六进制数字，范围为 **00-FF**，前缀为 **x**。多个字节必须以逗号分隔。例如，**-sqdb x08,xA4**。

仅当同时指定了 **-sq** 参数时，才能指定 **-sqdb** 参数。如果还指定了 **-t** 参数的 `text` 值，那么无法指定 **-sqdb** 参数。

-sqdp

可选。指定源文本和二进制定界符的插入位置。仅当同时指定了 **-sqdt** 和 **-sqdb** 参数之一时，才能指定 **-sqdp** 参数。

指定以下某个选项：

前缀

在每条消息的开头插入定界符

postfix

在每条消息尾部插入定界符。这是缺省选项。

-sqwt wait_time

可选。指定等待以下某个条件得以满足的时间（秒）：

- 有新消息放在队列上
- 如果指定了 **-sqgi** 参数，有完整组放在队列上

如果在 *wait_time* 指定的时间内这两个条件均未满足，那么源代理将停止从队列读取并完成传输。如果未指定 **-sqwt** 参数，那么在以下条件下源代理立即停止读取源队列：源队列为空、指定了 **-sqgi** 参数或者队列中无完整组。

仅当同时指定了 **-sq** 参数时，才能指定 **-sqwt** 参数。

-sq

可选。指定传输的源是队列。

-mquserid (userID)

可选。指定通过协调队列管理器进行认证时所使用的用户标识。

-mqpassword (password)

可选。指定通过协调队列管理器进行认证时所使用的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

source_specification

如果指定了 **-df**、**-dd**、**-dp**、**-dp** 或 **-ds** 参数之一，那么为必需。如果指定 **-td** 参数，请勿指定 *source_specification*。

- 如果尚未指定 **-sq** 参数，那么 *source_specification* 是用于确定文件传输的源的一个或多个文件规范。文件规范以空格定界。文件规范可采用五种格式之一，并且可包含通配符。有关 WMQFTE 中的通配符的更多信息，请参阅第 721 页的『使用通配符』。您可以在文件规范中使用两个星号字符 (**) 来转义属于文件规范的星号。

要传输其文件名中包含空格的文件，请将包含空格的文件名括在双引号内。例如，要将文件 `a b.txt` 传输到文件 `c d.txt`，请在 **fteCreateTemplate** 命令中指定以下文本：

```
-df "c d.txt" "a b.txt"
```

每个文件规范必须采用以下某种格式：

文件名称

文件的名称，使用源代理程序运行所在系统的相应表示法来表示。将文件名指定为源文件规范时，将复制该文件的内容。

目录

目录的名称，使用源代理程序运行所在系统的相应表示法来表示。当目录指定为源文件规范时，将复制目录的内容。更准确来说，将复制目录及其所有子目录中的所有文件，包括隐藏的文件。

例如，要仅将 DIR1 的内容复制到 DIR2，请指定 `DIR1/* DIR2`

连续数据集

(仅限 z/OS)。顺序数据集或分区数据集成员的名称。通过在数据集名称之前加上两个正斜杠字符 (//) 来表示数据集。

分区数据集

(仅限 z/OS)。分区数据集的名称。通过在数据集名称之前加上两个正斜杠字符 (//) 来表示数据集名称。

Connect:Direct 节点上的文件名或目录

(仅限 Connect:Direct 网桥代理程序)。Connect:Direct 节点的名称、冒号字符 (:), 以及托管 Connect:Direct 节点的系统上的文件或目录路径。例如, `connect_direct_node_name:file_path`。

如果源代理是 Connect:Direct 网桥代理, 那么将仅接受此格式的源规范。

注: 如果源代理是 Connect:Direct 网桥代理, 那么文件路径中不支持通配符。

- 如果指定了 **-sq** 参数, 那么 `source_specificatio` 是源代理队列管理器上本地队列的名称。您只能指定一个源队列。采用以下格式指定源队列:

```
QUEUE_NAME
```

源队列规范中未包含队列管理器名称, 因为队列管理器必须与源代理队列管理器相同。

-? or -h

可选。显示命令语法。

示例

在此示例中, 将创建名为 `payroll accounts monthly report template` 的传输模板。提交时, 此模板将指定的目录中扩展名为 `.xls` 的任何文件从代理程序 `PAYROLL1` 传输到代理程序 `ACCOUNTS`:

```
fteCreateTemplate -tn "payroll accounts monthly report template" -sa PAYROLL -sm QM_PAYROLL1 -da ACCOUNTS -dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

在此示例中, 将创建名为 `jupiter_neptune_sched_template` 的传输模板。提交时, 模板将文件 `originalfile.txt` 从 `QM_JUPITER` 所在的系统传输到 `QM_NEPTUNE` 所在的系统。文件传输调度为 `09:00` (基于源代理程序所在系统的系统时间) 执行, 并且每两个小时执行四次:

```
fteCreateTemplate -tn jupiter_neptune_sched_template -sa AGENT1 -sm QM_JUPITER -da AGENT2 -dm QM_NEPTUNE -tb source -ss 09:00 -oi hours -of 2 -oc 4 -df C:\import\transferredfile.txt C:\export\originalfile.txt
```

在此示例中, 将创建名为 `jupiter neptune trigger template` 的传输模板。提交模板时, 文件 `originalfile.txt` 将从 `AGENT1` 传输到 `AGENT2`, 条件是文件 `A.txt` 在 `AGENT1` 上存在:

```
fteCreateTemplate -tn "jupiter neptune trigger template" -sa AGENT1 -sm QM_JUPITER -da AGENT2 -dm QM_NEPTUNE -tr file=exist,C:\export\A.txt -df C:\import\transferredfile.txt C:\export\originalfile.txt
```

在此示例中, 创建了称为 `ascii_ebcdic_template` 的模板。提交模板时, 文件 `originalfile.txt` 将从 `AGENT1` 所在系统传输到 `AGENT2` 所在系统上的数据集 `//'USERID.TRANS.FILE.TXT'`。已选择文本方式来将数据从 ASCII 转换到 EBCDIC。

```
fteCreateTemplate -tn ascii_ebcdic_template -t text -sa AGENT1 -da AGENT2 -ds '//TRANS.FILE.TXT;RECFM(V,B);BLKSIZE(6144);LRECL(1028);SPACE(5,1)' C:\export\originalfile.txt
```

在此示例中，创建了称为 `ebcdic_ascii_template` 的模板。提交模板时，AGENT1 所在系统上的标准数据集的成员将传输到 AGENT2 所在系统上的文件。已选择文本方式来将文件从 EBCDIC 转换到 ASCII。

```
fteCreateTemplate -tn ebcdic_ascii_template -t text -sa AGENT1 -da AGENT2 -df /tmp/IEEUJV.txt  
"//'SYS1.SAMPLIB(IEEUJV)'"
```

返回码

返回码	描述
0	命令成功完成。
1	命令结束但未成功。

相关任务

[第 248 页的『使用传输模板』](#)

您可以使用文件传输模板来存储重复或复杂传输的公共文件传输设置。使用 **fteCreateTemplate** 命令从命令行创建传输模板，或者使用 IBM MQ Explorer 通过 "为受管文件传输创建新模板" 向导创建传输模板，或者通过选中 **将传输设置另存为模板** 复选框在创建文件传输时保存模板。传输模板窗口显示在您的 IBM MQ Managed File Transfer 网络中已创建的所有传输模板。

[第 249 页的『使用 IBM MQ Explorer 创建文件传输模板』](#)

您可以从 IBM MQ Explorer 或从命令行创建文件传输模板。然后，您可以使用该模板来利用模板详细信息创建新的文件传输，或者提交该模板以启动文件传输。

相关参考

[第 497 页的『fteCreateTransfer（创建新的文件传输）』](#)

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

[第 537 页的『fteListTemplates（列出 IBM MQ Managed File Transfer 模板）』](#)

fteListTemplates 命令可用于列出协调队列管理器上的可用 IBM MQ Managed File Transfer 传输模板。

[第 529 页的『fteDeleteTemplates（删除 IBM MQ Managed File Transfer 模板）』](#)

使用 **fteDeleteTemplates** 命令可从协调队列管理器中删除现有的 IBM MQ Managed File Transfer 模板。

fteCreateTransfer（创建新的文件传输）

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

用途

使用 **fteCreateTransfer** 命令可创建新文件传输，然后从 IBM MQ Managed File Transfer 代理程序启动新文件传输。要了解如何传输文件（包括文本文件、数据集和生成数据组 (GDG)），请参阅 [第 703 页的『文件传输准则』](#)。

您可以从可连接到 WebSphere MQ 网络的任何系统运行 **fteCreateTransfer** 命令，然后路由到源代理队列管理器。具体而言，为使命令运行，您必须在该系统上安装 IBM MQ Managed File Transfer 组件（服务或代理程序），并将此系统上的 IBM MQ Managed File Transfer 组件配置为与 WebSphere MQ 网络通信。

此命令使用名为 `command.properties` 的属性文件来连接到 WebSphere MQ 网络。如果 `command.properties` 文件不包含属性信息，将对本地系统上的缺省队列管理器进行绑定方式连接。如果 `command.properties` 文件不存在，那么将生成错误。有关更多信息，请参阅 [第 588 页的『command.properties 文件』](#)。

您可以针对一个文件传输指定多个源文件，但它们必须源自一个源代理，并终止于一个目标代理。不支持在单个传输中将一个源文件传输到相同代理或多个不同代理上的多个目标文件。Ant 脚本编制可用于将相同的源文件发送到一个或多个代理上的多个目标。有关更多信息，请参阅 [第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』](#)。

特殊字符

使用包含特殊字符的参数时，请谨慎操作，以免命令 `shell` 错误地解释这些字符。例如，命令 `shell` 可能会解释包含单引号的标准数据集名称和包含星号字符的源规范，而不是在传输请求中传递。为避免命令 `shell` 对字符进行解释，请使用双引号括起整个参数，或使用命令 `shell` 的转义序列对特殊字符进行转义。

相对路径

fteCreateTransfer 命令支持使用相对文件路径。在分布式系统和 z/OS UNIX 系统服务上，缺省情况下将路径视为相对于运行代理时所采用的用户的主目录。要更改与路径名称相对的目录，请在 `agent.properties` 文件中设置 `transferRoot` 属性。此文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` 目录中。将以下行添加到该文件：

```
transferRoot=directory_name
```

例如，将 `C:\TransferRoot` 指定为 `C:\\TransferRoot` 或 `C:/TransferRoot`。

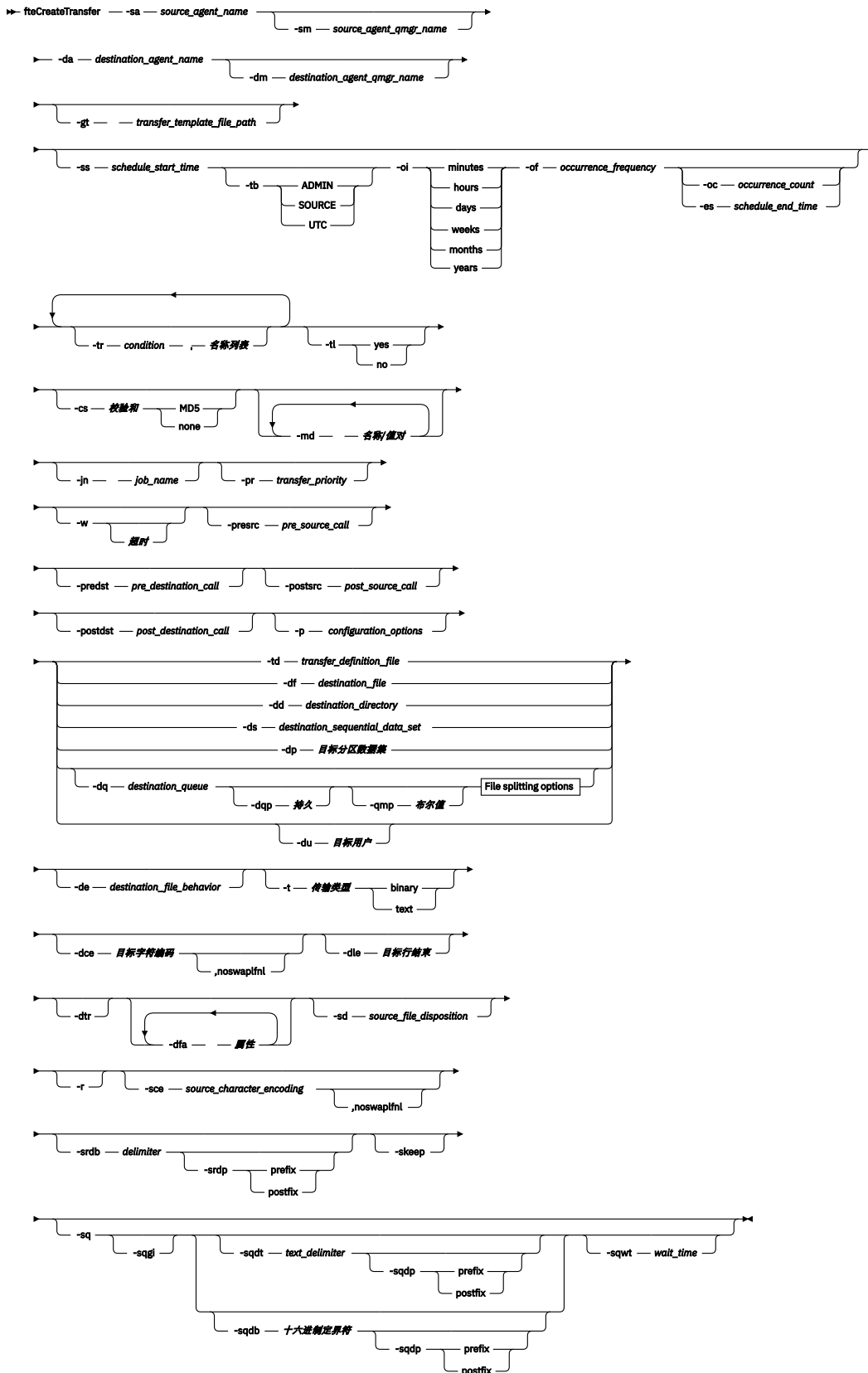
在 z/OS 上，缺省情况下，运行代理所使用的用户名将作为高级限定符前缀添加到尚未标准化的数据集规范。例如：`//ABC.DEF`。要更改作为前缀添加到数据集名称的值，请在 `agent.properties` 文件中设置 `transferRootHLQ` 属性。此文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` 目录中。将以下行添加到该文件：

```
transferRootHLQ=prepend_value
```

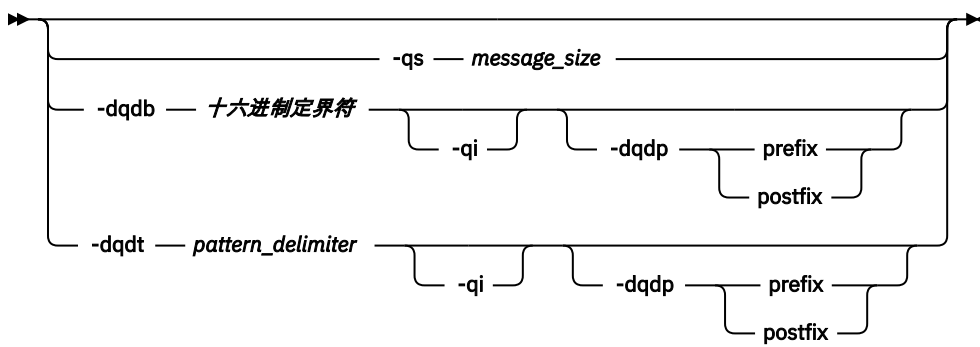
然而，对于涉及 z/OS 系统上的 Connect:Direct 节点的传输，数据集规范将解释为标准名称。不会向数据集名称添加任何高级限定符。

语法

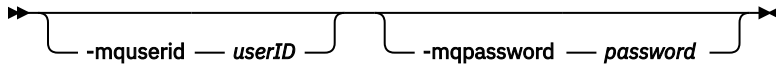
fteCreateTransfer



File splitting options



MQ 安全性的参数



➡ **source_specification** ➡

代理规范的参数

-sa *source_agent_name*

必需。从中传输源文件的代理程序的名称。

如果指定协议网桥代理作为源代理，那么您之后无法指定数据集作为源文件规范。

如果指定 **-td** 参数，并且传输定义文件包含要用于传输的源代理，请不要指定 **-sa** 参数。

-sm *source_agent_qmgr_name*

可选。源代理连接到的队列管理器的名称。

如果未指定 **-sm** 参数，那么所使用的队列管理器由正在使用的配置选项集确定，该配置选项基于源代理程序名称。如果无法找到源代理程序的 **agent.properties** 文件，那么文件传输将失败。

-da 目标代理名称

必需。要将文件传输到的代理程序的名称。

如果指定 **-td** 参数，并且传输定义文件包含要用于传输的目标代理，请不要指定 **-da** 参数。

-dm *destination_agent_qmgr_name*

可选。目标代理连接到的队列管理器的名称。

如果未指定 **-dm** 参数，那么所使用的队列管理器由正在使用的配置选项集确定，该配置选项基于目标代理名称。如果找不到目标代理的 **agent.properties** 文件，那么文件传输将失败。

生成传输模板的参数

-gt *transfer_template_file_path*

可选。生成传输模板 XML 消息并将该消息写入文件。如果指定该参数，那么不会将任何传输请求发送给 IBM MQ Managed File Transfer。而是将传输请求消息的内容写入指定的 XML 文档。然后，您可以使用此 XML 文档来定义用于资源监视的任务。请参阅 **fteCreateMonitor** 命令以获取有关如何创建资源监视器的信息。如果未指定该参数，那么缺省行为将发生并且将执行实际传输请求。

必须提供 XML 输出文件的完整路径和名称作为该参数的输入，例如，

C:\templates\transfer_reports.xml

在 z/OS 上，必须将传输模板文档存储在 z/OS UNIX 系统服务上的 UNIX 文件中。您无法在 z/OS 顺序文件或 PDS 成员中存储传输模板文档。

在 IBM i 上，必须在集成文件系统中存储传输模板文档。

使用 **-gt** 参数创建的传输模板 XML 消息不同于使用 **fteCreateTemplate** 命令创建的传输，这意味着您无法交互使用两种不同类型的模板。

注: 如果要通过运行带有 **-gt** 参数的 **fteCreateTransfer** 命令来生成传输模板 XML 文档，然后使用 **-td** 参数提供该传输模板 XML 文档作为 **fteCreateTransfer** 命令的输入，那么必须确保生成了指定了与 **-td** 选项互斥的那些参数的传输模板 XML 文档。

与 **-td** 选项互斥的参数包括:

- **-dd** *destination_directory*
- 源路径
- **-df** 目标文件
- **-cs** 校验和
- **-de** *destination_file_behavior*
- **-dq** *destination_queue*
- **-t** 传输类型
- **-sd** *source_file_处置*

例如，无法在 **fteCreateTransfer** 命令上同时指定 **-td** 和 **-t** 参数 (指示传输是二进制传输还是文本传输)。这意味着，如果要将传输模板 XML 文档传递到命令并指定该传输应该是文本传输，那么应通过指定 **-gt** 和 **-t** 文本参数来创建 XML 文档。

调度传输的参数

-ss *schedule_start_time*

可选。指定您希望调度的传输开始执行的时间和日期。请使用以下某种格式来指定时间和日期。使用 24 小时时钟指定时间:

```
yyyy-MM-ddThh:mm  
hh:mm
```

如果不存在可能影响传输的问题，那么调度的文件传输将在调度开始时间后一分钟内开始。例如，您的网络或代理可能存在阻止调度传输开始的问题。

-tb

可选。指定想要用于已安排的文件传输的时基。即，想要使用系统时间还是全球标准时间 (UTC)。该参数只能与 **-ss** 参数一起使用。指定以下某个选项:

admin

用于已安排的传输的开始和结束时间基于本地管理员使用的系统的时间和日期。这是缺省值。

源

用于调度的传输的开始时间和结束时间基于源代理所在系统的日期和时间。

UTC

用于调度的传输的开始时间和结束时间基于全球标准时间 (UTC)。

-oi

可选。指定调度的传输出现的时间间隔。该参数只能与 **-ss** 参数一起使用。指定以下某个选项:

分钟

小时

days

weeks

months

years

-of 实例频率

可选。指定调度的传输出现的频率。例如，每 **5** 周或每 **2** 个月。必须仅将此参数与 **-oi** 和 **-ss** 参数一起指定。如果未指定该参数，那么将使用缺省值 **1**。

-oc 实例计数

可选。指定您希望此调度的传输发生的次数。达到发生计数之后，将删除调度的传输。

仅将此参数与 **-oi** 和 **-ss** 参数一起指定。

如果指定 **-oc** 参数，那么无法指定 **-es** 参数，因为这些参数是互斥的。

您可以省略 **-oc** 和 **-es** 参数以创建无限重复的传输。

-es schedule_end_time

可选。重复的调度传输结束的日期和时间。

该参数只能与 **-oi** 和 **-ss** 参数一起指定。

如果指定 **-es** 参数，那么无法指定 **-oc** 参数，因为这些参数是互斥的。

您可以省略 **-es** 和 **-oc** 参数以创建无限重复的传输。

使用以下某种格式来指定结束时间和日期。使用 **24** 小时时钟指定时间：

```
yyyy-MM-ddThh:mm  
hh:mm
```

触发传输的参数

-tr

可选。指定为执行此文件传输所必须满足的条件。如果不满足该条件，那么根据源代理，将废弃该文件传输，不会执行任何传输。指定以下格式：

```
condition,namelist
```

其中条件是以下某个值：

file=exist

名称列表中至少有一个文件存在。即，如果存在名称列表中的任何文件，那么就满足条件。

file!=exist

名称列表中至少有一个文件不存在。即，如果名称列表中有任何文件不存在，那么满足条件。

filesize>=size

至少存在名称列表中的一个文件，并且最小大小为 *size* 指定的大小。*size* 是一个整数，可选大小单位为 KB、MB 或 GB。例如，`filesize">"=10KB`。如果未指定大小单位，大小将假定以字节为单位。在所有操作系统上，您在命令行上指定 **filesize** 选项时必须将大于号 (>) 括在双引号内，如该示例中所示。

并且，其中 *namelist* 是源代理所在系统上的以逗号分隔的文件名列表。根据操作系统，如果您要在名称列表中使用包含空格的路径名或文件名，可能必须将路径名和文件名括在双引号中。

可以多次使用 **-tr** 参数指定多个触发条件。然而，在此情况下，必须满足每个单独的触发条件才能进行文件传输。

注：要持续监视资源以了解触发条件是否满足，强烈建议您使用[资源监视](#)。您可以使用 [fteCreateMonitor](#) 命令来创建资源监视器。

在以下示例中，将文件 `file1.doc` 从 `AGENT1` 传输到 `AGENT2`，条件是文件 `A.txt` 和/或文件 `B.txt` 都存在于 `AGENT1` 和上，而文件 `A.txt` 和/或文件 `B.txt` 都等于或大于 **1 GB**：

```
fteCreateTransfer -sa AGENT1 -sm QM_JUPITER -da AGENT2 -dm QM_NEPTUNE  
-tr file=exist,C:\export\A.txt,C:\export\B.txt
```

```
-tr filesize">"=1GB,C:\export\A.txt,C:\export\B.txt  
-df C:\import\file1.doc C:\export\file1.doc
```

您可以将触发参数与调度参数相结合。如果同时指定这两种类型的参数，那么触发条件将应用于调度参数创建的文件传输。

协议网桥代理程序不支持 **-tr** 参数。

-tl

可选。指定是否将触发器失败写入传输日志。指定以下某个选项：

yes

针对失败的已触发传输创建传输日志条目。这是缺省行为，即使未指定 **-tl** 参数也是如此。

否

不针对失败的已触发传输创建任何传输日志条目。

指定传输选项的参数

-jn 作业名

可选。启动传输时添加到传输日志消息的用户定义的作业名称标识。

-md

可选。指定将传递给代理程序运行的出口点的用户定义的元数据。 **-md** 参数可以采用以逗号分隔的一个或多个名称/值对。每个名称对由 *name=value* 组成。您可以在一个命令中多次使用 **-md** 参数。

当代理属性 `enableUserMetadataOptions` 设置为 `true` 值时，某些用户定义的元数据键会为传输提供更多选项。有关当前支持的用户定义的元数据键的更多信息，请参阅第 606 页的『[支持的用户定义的元数据键](#)』。当 `enableUserMetadataOptions` 属性设置为 `true` 时，用户定义的使用不支持以 `com.ibm.wmqfte.` 开头的密钥名称。

-cs 校验和

可选。指定是否针对文件传输数据运行校验和算法，以检查所传输文件的完整性。指定以下某个选项：

MD5

计算数据的 MD5 校验和。针对源和目标文件生成的校验和将写入传输日志以供验证之用。缺省情况下，IBM MQ Managed File Transfer 计算所有文件传输的 MD5 校验和。

none

不会为文件传输数据计算 MD5 校验和。传输日志记录校验和设置为 `none`，并且校验和值为空。例如：

```
<checksum method="none"></checksum>
```

如果使用 `none` 选项，那么可能会提高文件传输性能，这取决于环境。然而，选择该选项意味着不会验证源或目标文件。

如果指定 **-cs** 参数，那么无法指定 **-td** 参数，因为这些参数是互斥的。但是，您可以在传输定义文件中指定校验和行为。

-pr 传输优先级

可选。指定传输的优先级。优先级为 0-9 之间的一个值，其中 0 为最低优先级。缺省优先级是源代理的优先级。

该值与 WebSphere MQ 的消息优先级值相匹配，请参阅 [Getting messages from a queue: priority](#) 以获取更多信息。文件传输数据的消息通信缺省为优先级 0，从而允许 WebSphere MQ 消息通信获得优先。

-qmp 布尔值

可选。指定传输写入目标队列的第一条消息是否已设置 WebSphere MQ 消息属性。有效选项如下：

true

在传输创建的第一条消息上设置消息属性。

false

不会在传输创建的第一条消息上设置消息属性。这是缺省值。

只有在指定了 **-dq** 参数的情况下才可以指定 **-qmp** 参数。有关更多信息，请参阅第 739 页的『写入到目标队列的消息上设置的 IBM MQ 消息属性』。

-qs 消息大小

可选。指定是否将文件分割为多条固定长度的消息。所有消息都具有相同的 WebSphere MQ 组标识；组中的最后一条消息设置了 WebSphere MQ LAST_MSG_IN_GROUP 标志。消息的大小由 *message_size* 的值指定。The format of 消息大小 is <长度><单位>, where 长度 is a positive integer value and 单位 is one of the following values:

B

字节。允许的最小值为目标消息的代码页的最大“每个字符的字节数”值的两倍。

K

等于 1024 字节。

M

这等于 1048576 字节。

如果文件以文本方式传输，并且采用双字节字符集或者多字节字符集，那么该文件会拆分为字符边界最接近指定消息大小的消息。

只有在指定了 **-dq** 参数的情况下才可以指定 **-qs** 参数。只能指定 **-qs**，**-dqdb** 和 **-dqdt** 参数之一。

-qi

可选。使用此选项包括用于将文件拆分为消息中的多条消息的定界符。定界符包含在消息头或尾部，取决于 **-dqdp** 参数（指定前缀或后缀）。缺省情况下，消息中不包含定界符。

只有在指定了 **-dqdt** 和 **-dqdb** 参数之一时，才可以指定 **-qi** 参数。

-p 配置选项

可选。该参数确定用于创建文件传输的一组配置选项。使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果您未指定该参数，那么会使用基于缺省协调队列管理器的配置选项集。

-w 超时

可选。指定 **-w** 参数将导致 **fteCreateTransfer** 命令在返回前等待代理的响应。如果不指定此参数，那么 **fteCreateTransfer** 命令最多等待 5 秒钟，以接收源代理针对传输所发送的关于已接收到传输请求的确认。如果在等待的 5 秒时间内未收到确认，**fteCreateTransfer** 命令会返回以下警告消息：

```
BFGCL0253W: No acknowledgment to command from agent within timeout.
```

timeout 参数为可选。如果指定 *timeout*，那么 **fteCreateTransfer** 命令等待最多 *timeout* 秒，以便代理响应。如果在达到时间限制之前代理程序未响应，那么该命令将生成警告并以返回码 2 结束。如果未指定 *timeout* 值，或者指定 *timeout* 值 -1，那么该命令将等待代理程序响应。

调用程序的参数

有关如何从 IBM MQ Managed File Transfer 启动程序的更多信息，请参阅第 303 页的『指定要运行的程序』。有关使用此部分中描述的参数来指定要调用的程序的示例，请参阅第 906 页的『使用 **fteCreateTransfer** 来启动程序的示例』。

-presrc 源前调用

可选。指定在传输启动前要在源代理程序调用的程序。请将以下格式用于 *pre_source_call*：

```
[type:]commandspec[, [retrycount][, [retrywait][, successsrc]]]
```

在此语法中，变量为：

类型

可选。有效值为 **executable**、**antscript**、**jcl** 和 **os4690background**。缺省值为 **executable**。

仅当针对 z/OS 环境中的代理程序时，**jcl** 值才适用。

针对 IBM 4690 环境中的代理程序时，**antscript** 值不适用。

commandspec

必需。命令规范。请使用以下格式之一：

- 类型 **executable**: `command[(arg1,arg2,...)]`
- 类型 **antscript**: `command[(name1=var1|target1,name2=var2|target2,...)]`
- 类型 **jcl**: `command`
- 类型 **os4690background**: `command[(arg1,arg2,...)]`

其中：

command

必需。要调用的程序的名称。

仅当针对 z/OS 环境中的代理程序时，**jcl** 值才适用。

针对 IBM 4690 环境中的代理程序时，**antscript** 值不适用。

方括号 ([]) 中的参数是可选的，并且语法取决于命令类型。命令或参数中出现的圆括号 (()) 和反斜杠字符 (\) 必须使用反斜杠字符 (\) 进行转义。

retrycount

可选。在程序未返回成功返回码时重试调用程序的次数。缺省值为 0。

retrywait

可选。重新尝试调用程序之前，要等待的秒数。缺省值为 0（重试之间无等待）。

successsrc

可选。用于确定程序调用何时成功运行的表达式。该表达式可由一个或多个表达式组成。组合这些表达式，其中用竖线字符 (|) 表示布尔值 OR，或者用和符号 (&) 表示布尔值 AND。每个表达式都采用以下格式：

```
[>|<|!]value
```

其中：

>

可选。*value* 的大于测试。

<

可选。*value* 的小于测试。

!

可选。*value* 的不等于测试。

value

必需。有效整数。

priority

可选（仅限 **os4690background**）。分配给 IBM 4690 系统上后台任务的优先级别。缺省值为 5，有效值范围为 1 - 9。

message

可选（仅限 os4690background）。针对执行的命令显示在 IBM 4690 系统后台控制屏幕上的状态消息。

-predst pre_destination_call

可选。指定在传输启动前要在目标代理程序调用的程序。 *pre_destination_call* 具有与 *pre_source_call* 相同的格式。

-postsrc post_source_call

可选。指定在传输完成后要在源代理调用的程序。 *post_source_call* 具有与 *pre_source_call* 相同的格式。

-postdst post_destination_call

可选。指定在传输完成后要在目标代理调用的程序。 *post_destination_call* 具有与 *pre_source_call* 相同的格式。

指定目标的参数

需要 **-td**、**-df**、**-dd**、**-ds**、**-dq**、**-du** 和 **-dp** 参数之一。您无法在一个传输请求中指定其中多个参数；这些参数互斥。

-td transfer_definition_file

可选。XML 文档的名称，该文档定义传输的一个或多个源和目标文件规范。或者，包含受管传输请求（可能已由 **-gt** 参数生成）的 XML 文档的名称。如果指定 **-td** 参数并在命令行上指定任何其他参数，那么这些其他参数将覆盖传输定义文件中的相应值。

fteCreateTransfer 命令查找与当前目录相关的传输定义文件。如果无法使用相对路径注释来指定传输定义文件的位置，请使用传输定义文件的标准路径和文件名代替。

在 z/OS 上，您必须将传输定义文件存储在 z/OS UNIX 系统服务上的 UNIX 文件中。您无法在 z/OS 顺序文件或 PDS 成员中存储传输定义文件。

在 IBM i 上，您必须将传输定义文件存储在集成文件系统中。

有关更多信息，请参阅[使用传输定义文件](#)。

-df 目标文件

可选。目标文件的名称。

如果目标代理是 Connect:Direct 网桥代理，那么将以 *connect_direct_node_name:file_path* 格式指定目标文件。Connect:Direct 网桥代理仅接受以此格式指定的文件路径。如果目标代理是 Connect:Direct 网桥代理，并且目标是 PDS 成员，那么还必须使用覆盖值指定 **-de** 参数。

注意以下信息：

- 如果目标代理是协议网桥代理，并且您想要为文件指定端点，那么请使用以下格式：

```
protocol_server:file_path
```

其中，*protocol_server* 是协议服务器（可选）的名称，*file_path* 是到协议服务器系统上的文件的路径。如果未指定协议服务器，那么将使用缺省协议服务器。

- 如果想要调用针对目标代理定义的任何 IBM MQ Managed File Transfer 传输 I/O 用户出口，那么您可以在传输中使用 **-df** 参数。
- 当目标代理在 z/OS 上时，如果指定的文件以 // 开头，那么假定它是分区 z/OS 数据集。

-dd 目标目录

可选。要将文件传输到的目录的名称。在运行目标代理程序的系统上指定一个有效目录名称。

如果目标代理是 Connect:Direct 网桥代理，那么将以 `connect_direct_node_name:directory_path` 格式指定目标目录。如果目标代理是 Connect:Direct 网桥代理，而目标是 PDS，那么还必须使用覆盖值指定 **-de** 参数。

注意以下信息：

- 如果目标代理是协议网桥代理，并且您想要在特定端点指定一个目录，请使用以下格式：

```
protocol_server:directory_path
```

其中，`protocol_serve` 是协议服务器（可选）的名称，而 `directory_path` 是到协议服务器系统上的目录的路径。如果未指定协议服务器，那么将使用缺省协议服务器。

- 如果想要调用针对目标代理定义的任何 IBM MQ Managed File Transfer 传输 I/O 用户出口，那么您可以在传输中使用 **-dd** 参数。
- 当目标代理在 z/OS 上时，如果指定的文件以 `//` 开头，那么假定它是 z/OS 分区数据集。

-ds destination_sequential_data_set

仅适用于 z/OS。可选。要将文件传输到的顺序数据集或 PDS 成员的名称。指定连续数据集名称或分区数据集成员。有关传输数据集的信息，请参阅第 703 页的『文件传输准则』。

数据集名称的语法如下所示：

```
//data_set_name{;attribute(value);...;attribute(value)}
```

或

```
//pds_data_set_name(member_name){;attribute(value);...;attribute(value)}
```

即，数据集名称说明符，以 `//` 为前缀，并（可选）后跟以分号分隔的若干属性。

例如：

```
//'TEST.FILE.NAME';DSNTYPE(PDS);RECFM(F,B);BLKSIZE(800);LRECL(80);CYL;SPACE(2,2)
```

如果数据集位于 Connect:Direct 节点上，那么必须为数据集名称前缀节点名称。例如：

```
CD_NODE1://'OBJECT.LIB';RECFM(F,B);BLKSIZE(800);LRECL(80)
```

如果目标代理是 Connect:Direct 网桥代理，并且目标是 PDS 成员，那么还必须使用覆盖值指定 **-de** 参数。有关数据集传输目标或源 Connect:Direct 节点的更多信息，请参阅第 706 页的『与 Connect:Direct 节点之间传输数据集』。

对于仅涉及 IBM MQ Managed File Transfer 代理的传输，如果使用单引号字符将数据集名称部分引起来，那么其指定标准数据集名称。如果未使用单引号字符将数据集名称引起来，那么系统针对目标代理程序添加缺省高级限定符（如果未设置 `transferRootHLQ`，那么这是 `transferRootHLQ` 代理程序属性的值或运行代理程序的用户标识）。

注：然而，对于涉及 z/OS 系统上的 Connect:Direct 节点的传输，数据集规范将解释为标准名称。不会向数据集名称添加任何高级限定符。即使是使用单引号字符将数据集名称引起来时也是如此。

将文件或数据集传输到磁带时，磁带上的任何现有数据集都将被替换。新数据集的属性将根据传输定义中传递的属性进行设置。如果未指定任何属性，那么新数据集的属性将设置为与源数据集相同，或者，如果源数据集是文件，那么这些属性将设置为缺省值。将忽略现有磁带数据集的属性。

数据集属性用于创建数据集或者确保现有数据集兼容。数据集属性的规范采用适用于 BPXWDYN 的格式（请参阅 [Requesting dynamic allocation](#) 以获取更多信息）。在代理将创建目标数据集时，将自动指定以下 BPXWDYN 属性：`DSN(data_set_name) NEW CATALOG MSG(numeric_file_descriptor)`。`numeric_file_descriptor` 的值由 IBM MQ Managed File Transfer 生成。对于数据集之间的传输，将为新目标数据集选择来自源的属性：`RECFM`、`LRECL` 和 `BLKSIZE`。新目标数据集的 `SPACE` 设置并非由 IBM MQ Managed File Transfer 设置，而是使用系统缺省值。因此，建议您在将要创建新数据集时指定 `SPACE` 属性。您可以在 `agent.properties` 文件中使用 **bpxwdynAllocAdditionalProperties**

属性来设置适用于所有传输的 BPXWDYN 选项。有关更多信息，请参阅 [第 591 页的『agent.properties 文件』](#)。

使用 **fteCreateTemplate** 命令，**fteCreateTransfer** 命令或 agent.properties 文件中的 **bpxwdynAllocAdditionalProperties** 属性时，不得指定某些 BPXWDYN 选项。有关这些属性的列表，请参阅 [第 712 页的『不能与 IBM MQ Managed File Transfer 一起使用的 BPXWDYN 属性』](#)。

当目标代理为协议网桥代理时，不支持 **-ds** 参数。

如果要调用针对代理定义的任何 IBM MQ Managed File Transfer 传输 I/O 用户出口，请不要在传输中指定 **-ds** 参数。使用 **-ds** 参数可防止针对目标调用传输 I/O 用户出口，并表示改为使用标准 IBM MQ Managed File Transfer I/O。

-dp destination_partitioned_data_set

仅适用于 z/OS。可选。要将文件传输到的目标 PDS 的名称。指定分区数据集名称。如果创建 PDS 作为传输的结果，那么缺省情况下将创建该 PDS 作为 PDSE。您可以通过指定 DSNTYPE=PDS 来覆盖缺省值。

PDS 数据集名称的语法如下所示：

```
//pds_data_set_name{;attribute;...;attribute}
```

数据集名称的语法与 **-ds (destination_sequential_data_set)** 参数描述的语法相同。用于指定位于 Connect:Direct 节点上的数据集的所有语法详细信息也适用于 **-dp** 参数。如果目标代理是 Connect:Direct 网桥代理，那么还必须指定值为 overwrite 的 **-de** 参数。

当目标代理为协议网桥代理时，不支持 **-dp** 参数。

如果要调用针对代理定义的任何 IBM MQ Managed File Transfer 传输 I/O 用户出口，请不要在传输中指定 **-dp** 参数。使用 **-dp** 参数可防止针对目标调用传输 I/O 用户出口，并表示改为使用标准 IBM MQ Managed File Transfer I/O。

-du 目标用户

可选。要将文件传输到其目标文件空间的用户的名称。有关文件空间的更多信息，请参阅 [第 339 页的『文件空间』](#)。

当目标代理是协议网桥代理或 Connect:Direct 网桥代理时，不支持 **-du** 参数。

如果要调用针对代理定义的任何 IBM MQ Managed File Transfer 传输 I/O 用户出口，请不要在传输中指定 **-du** 参数。使用 **-du** 参数将阻止针对目标调用传输 I/O 用户出口，并且意味着将使用标准 IBM MQ Managed File Transfer I/O 代替。

-dq 目标队列

可选。要将文件传输到的目标队列的名称。您可以选择使用格式 QUEUE@QUEUEMANAGER 在此规范中包含一个队列管理器名称。如果未指定队列管理器名称，那么将使用目标代理队列管理器名称。必须指定队列管理器上已存在的有效队列名称。

当目标代理是协议网桥代理或 Connect:Direct 网桥代理时，或者当源规范是队列时，不支持 **-dq** 参数。

如果要调用针对代理定义的任何 IBM MQ Managed File Transfer 传输 I/O 用户出口，请不要在传输中指定 **-dq** 参数。使用 **-dq** 参数可防止针对目标调用传输 I/O 用户出口，并表示改为使用标准 IBM MQ Managed File Transfer I/O。

-dqp persistent

可选。指定写入目标队列的消息是否持久。有效选项如下：

true

将持久消息写入目标队列。这是缺省值。

false

将非持久性消息写入目标队列。

qdef

持久性值将从目标队列的 DefPersistence 属性获取。

只有在指定了 **-dq** 参数的情况下才可以指定 **-dqdp** 参数。

-dqdb 十六进制定界符

可选。指定将二进制文件分割为多条消息时使用十六进制定界符。所有消息都具有相同的 WebSphere MQ 组标识；组中的最后一条消息设置了 WebSphere MQ LAST_MSG_IN_GROUP 标志。将十六进制字节指定为定界符的格式为 xNN，其中 N 是 0-9 或 a-f 范围内的字符。您可以通过指定十六进制字节的逗号分隔列表（例如：x3e,x20,x20,xbf），将一系列十六进制字节指定为定界符。

只有在指定了 **-dq** 参数并且传输处于二进制方式时，才可以指定 **-dqdb** 参数。您只能指定 **-qs**、**-dqdb** 和 **-dqdt** 参数之一。

-dqdt pattern

可选。指定将文本文件分割为多条消息时使用 Java 正则表达式。所有消息都具有相同的 WebSphere MQ 组标识；组中的最后一条消息设置了 WebSphere MQ LAST_MSG_IN_GROUP 标志。将正则表达式指定为定界符的格式是括在括号 (*regular_expression*) 中或括在双引号 "*regular_expression*" 中的正则表达式。有关更多信息，请参阅第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』。

缺省情况下，目标代理将正则表达式可匹配的字符串的长度限制为 5 个字符。您可以通过编辑 **maxDelimiterMatchLength** 代理程序属性来更改此行为。有关更多信息，请参阅第 591 页的『高级代理属性』。

仅当还为 **-t** 参数指定了 **-dq** 参数和值 **text** 时，才能指定 **-dqdt** 参数。只能指定 **-qs**、**-dqdb** 和 **-dqdt** 参数之一。

-dqdp position

可选。指定在分割文件时目标文本和二进制定界符的预期位置。只有在指定了 **-dqdt** 和 **-dqdb** 参数之一时，才可以指定 **-dqdp** 参数。

指定以下某个选项：

前缀

定界符原应位于每行的开头。

postfix

预期定界符在每行的末尾。这是缺省选项。

-de destination_file_behavior

可选。指定在目标系统上已存在目标文件时执行的操作。有效选项如下：

Error

报告错误并且不传输文件。这是缺省值。

覆盖

覆盖现有目标文件。

如果指定 **-de** 参数，那么无法指定 **-td** 参数，因为这些参数是互斥的。然而，您可以在传输定义文件中指定目标文件存在行为。

-t 传输类型

可选。指定文件传输的类型：二进制方式或文本方式。

二进制

文件中的数据将在不经任何转换的情况下进行传输。这是缺省值。

文本

将转换文件的代码页和行尾字符。您可以使用 **-sce**、**-dce** 或 **-dle** 参数指定要用于转换的代码页和行结尾。如果未指定 **-sce**、**-dce** 或 **-dle** 参数，那么执行的确切转换取决于源代理程序和目标代理程序的操作系统。

例如，从 Windows 传输到 z/OS 的文件将其代码页从 ASCII 转换为 EBCDIC。在文件从 ASCII 转换为 EBCDIC 时，行尾字符将从 ASCII 回车符 (CR) 和换行 (LF) 字符对转换为 EBCDIC 换行 (NL) 字符。

有关如何传输 z/OS 数据集的更多信息，请参阅[在 z/OS 和分布式系统之间传输文件和数据集](#)。以及在[数据集之间传输](#)。

如果指定 **-t** 参数，那么无法指定 **-td** 参数，因为这些参数是互斥的。然而，您可以在传输定义文件中指定传输方式行为。

-dce 目标字符编码

可选。指定用于在目标写入文件的字符编码。该选项仅适用于文本文件，而且还必须指定 **-t text**。适用于转换的代码页取决于目标代理程序的平台。有关可用代码页的列表，请参阅主题第 745 页的『可用代码页』。

noswaplfnl

缺省情况下，WebSphere MQ 受管文件传输将 **swaplfnl** 用于受支持的 EBCDIC 字符集。使用 **swaplfnl** 可更改与 EBCDIC LF 0x25 字符之间的字符集映射的行为。但是，该操作有时候会产生不需要的映射。使用 **noswaplfnl** 覆盖此行为。

-dle destination_line_ending

可选。指定在目标编写文件时使用的行尾字符。此选项仅适用于文本文件，因此您还必须指定 **-t text** 参数。有效选项为：

LF

换行。这是 UNIX 平台和 z/OS UNIX 系统服务文件的缺省设置。在将 IBM MQ Managed File Transfer 随附的标准 EBCDIC 代码页用于 EBCDIC 文件时，行尾字符将映射到 NL 字符 (0x15)，而不会映射到 LF 字符 (0x25)。

CRLF

后跟换行的回车符。这是 Microsoft Windows 的缺省值。

如果传输的目标是 z/OS 数据集，那么将忽略该选项。

-dtr

可选。指定将截断长于 LRECL 数据集属性的目标记录。如果未指定该参数，那么将合并记录。该参数仅对目标是数据集的文本方式传输有效。

-dfa attributes

可选。指定与传输中目标文件相关联的以分号分隔的文件属性列表。可以使用或不使用值来指定 **-dfa** 参数。例如，没有值：

```
-dfa ATTRIBUTE1;ATTRIBUTE2
```

例如，具有值：

```
-dfa ATTRIBUTE1(VALUE);ATTRIBUTE2(VALUE)
```

例如，一个属性有值，而另一个没有值：

```
-dfa ATTRIBUTE1;ATTRIBUTE2(VALUE)
```

可以在命令中多次使用 **-dfa** 参数。

有关文件属性的更多信息，请参阅第 82 页的『文件分发属性』。

安全性的参数

-mquserid (userID)

可选。指定要向命令队列管理器进行认证的用户标识。

-mqpassword (password)

可选。指定要向命令队列管理器进行认证的密码。您还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

指定源的参数

-sd *source_file* 处置

可选。指定当源文件成功传输到其目标时，在文件到文件或文件到消息传输中对源文件执行的操作。有效选项如下：

leave

源文件将保持不变。这是缺省值。

delete

源文件传输成功后将从源系统中删除。

注：对于消息到文件传输，源队列上的消息一旦成功传输，将始终被删除。这意味着如果针对消息到文件传输将 **-sd** 参数设置为 **leave**，那么将忽略该值。

在 z/OS 上，如果源是磁带数据集并且指定了 **delete** 选项，那么将重新安装磁带以删除数据集。该行为是由于系统环境的行为。

如果源是队列，并且指定 **leave** 选项，那么命令返回一个错误且不请求传输。

如果源代理是 Connect:Direct 网桥代理，并且指定了 **delete** 选项，那么行为不同于常用的源处置行为。将出现以下某种情况：

- 如果 Connect:Direct 使用 IBM MQ Managed File Transfer 生成的进程从源移动文件或数据集，那么指定 **delete** 选项将导致传输失败。要指定删除源文件，您必须提交用户定义的 Connect:Direct 进程。有关更多信息，请参阅第 297 页的『通过文件传输请求提交用户定义的 Connect:Direct 进程』。
- 如果 Connect:Direct 使用用户定义的进程从源移动文件或数据集，那么该参数将通过 **%FTEFDISP** 内置符号变量传递给此进程。用户定义的进程确定是否删除源。传输返回的结果取决于用户定义的进程返回的结果。

如果指定 **-sd** 参数，那么无法指定 **-td** 参数，因为这些参数是互斥的。然而，您可以在传输定义文件中指定源处置行为。

-r

可选。当 *source_specification* 包含通配符时，那么将以递归方式传输子目录中的文件。当向 IBM MQ Managed File Transfer 提供通配符作为 *source_specification* 时，仅当指定了 **-r** 参数时，才会传输与通配符匹配的任何子目录。在 *source_specification* 与子目录匹配时，始终传输该目录及其子目录中的所有文件（包含隐藏的文件）。

有关 IBM MQ Managed File Transfer 如何处理通配符的更多信息，请参阅[使用通配符](#)

如果指定 **-r** 参数，那么无法指定 **-td** 参数，因为这些参数是互斥的。然而，您可以在传输定义文件中指定递归行为。

-sce *source_character_encoding*

可选。指定用于在执行字符转换时读取源文件的字符编码。该选项仅适用于文本文件，而且还必须指定 **-t text**。适用于转换的代码页取决于目标代理程序的平台，因为转换在目标系统上执行。有关可用代码页的列表，请参阅主题第 745 页的『可用代码页』。

noswaplfnl

缺省情况下，WebSphere MQ 受管文件传输将 **swaplfnl** 用于受支持的 EBCDIC 字符集。使用 **swaplfnl** 可更改与 EBCDIC LF 0x25 字符之间的字符集映射的行为。但是，该操作有时候会产生不需要的映射。使用 **noswaplfnl** 覆盖此行为。

-skeep

可选。指定在文本方式传输过程中保留从固定长度格式的面向记录的文件读取的源记录上的尾部空格（例如，z/OS 数据集）。如果未指定该参数，将从源记录移除尾部空格。

-srdb 定界符

可选。对于面向记录的源文件（例如，z/OS 数据集），指定一个或多个字节值以在将记录追加到二进制文件时作为定界符插入。必须将每个值指定为 00-FF 范围内的两个十六进制数字，前缀为 x。使用逗号分隔多个字节。例如：

```
-srdb x0A
```

或

```
-srdb x0D,x0A
```

必须以二进制方式配置传输。

-srdp position

可选。指定要插入源记录定界符的位置。只有在指定了 **-srdb** 参数的情况下才可以指定 **-srdp** 参数。

指定以下某个选项：

前缀

在每条记录的开头插入的定界符。

postfix

在每条记录的尾部插入的定界符。这是缺省选项。

-sq

可选。指定传输的源是队列。

如果要调用针对代理定义的任何 IBM MQ Managed File Transfer 传输 I/O 用户出口，请不要在传输中指定 **-sq** 参数。使用 **-sq** 参数可防止针对源调用传输 I/O 用户出口，并表示改为使用标准 IBM MQ Managed File Transfer I/O。

-sqgi

可选。指定按 WebSphere MQ 组标识分组的消息。第一个完整组将写入目标文件。如果不指定此参数，那么源队列上的所有消息将写入目标文件。

只有在指定了 **-sq** 参数的情况下才可以指定 **-sqgi** 参数。

-sqdt 文本定界符

可选。指定一个文本序列，以在将多条消息追加到文本文件时作为定界符插入。可以在定界符中包含字符串字面值的 Java 转义序列。例如，**-sqdt \u007d\n**。

使用传输的源编码将文本定界符编码为二进制格式。将以二进制格式读取每条消息。编码的定界符将以二进制格式附加到消息（如 **-sqdp** 参数所指定），并且结果将以二进制格式传输到目标代理。如果源代理代码页包含移入和移出状态，那么代理假定每条消息在消息结束时进入移出状态。在目标代理上，以与文件相同的方式将二进制数据转换为文件文本传输。

仅当还为 **-t** 参数指定了 **-sq** 参数和值 **text** 时，才能指定 **-sqdt** 参数。

-sqdb 十六进制定界符

可选。指定一个或多个字节的值，以在将多条消息追加到二进制文件时作为定界符插入。每个值都必须指定为两个十六进制数字，范围为 00-FF，前缀为 x。多个字节必须以逗号分隔。例如，**-sqdb x08,xA4**。

只有在指定了 **-sq** 参数的情况下才可以指定 **-sqdb** 参数。仅当同时为 **-t** 参数指定值 **text** 时，才能指定 **-sqdb** 参数。

-sqdp position

可选。指定源文本和二进制定界符的插入位置。仅当还指定了其中一个 **-sqdt** 和 **-sqdb** 参数时，才能指定 **-sqdp** 参数。

指定以下某个选项：

前缀

在每条消息的开头插入定界符

postfix

在每条消息尾部插入定界符。这是缺省选项。

-sqwt wait_time

可选。指定等待以下某个条件得以满足的时间（秒）：

- 对于要在队列上出现的新消息
- 如果指定了 **-sqgi** 参数，那么针对要在队列上显示的完整组

如果在等待时间指定的时间内，这两个条件均未满足，那么源代理将停止从队列读取并完成传输。如果未指定 **-sqwt** 参数，那么如果源队列为空，或者在指定了 **-sqgi** 参数的情况下，如果队列上没有完整组，那么源代理将立即停止从源队列读取。

有关使用 **-sqwt** 参数的信息，请参阅第 744 页的『用于对“消息到文件”传输指定等待时间的指南』。

只有在指定了 **-sq** 参数的情况下才可以指定 **-sqwt** 参数。

source_specification

确定文件传输源的一个或多个文件规范。

如果指定 **-df**，**-dd**，**-dp**，**-dq**，**-du**，或 **-ds** 参数之一，那么此参数是必需的。如果指定 **-td** 参数，请勿指定 *source_specification*。

- 如果未指定 **-sq** 参数，那么 *source_specification* 是一个或多个文件规范，用于确定文件传输的源。文件规范可采用五种格式之一，并且可包含通配符。有关通配符的更多信息，请参阅第 721 页的『使用通配符』。您可以在文件规范中使用两个星号字符 (**) 来转义属于文件规范的星号。

您可以指定由空格字符分隔的多个源文件规范。但是，如果为 **-df** 或 **-ds** 参数指定了多个源规范，并且还指定了 **-de overwrite**，那么目标将仅包含您最后指定的源文件的数据。如果未指定 **-de overwrite**，那么传输只能部分成功。如果目标文件预先不存在，那么将包含首先指定的源文件的数据。

要传输文件名中包含空格的文件，例如，a b.txt 到文件 c d.txt，请用双引号将包含空格的文件名引起来。在 **fteCreateTransfer** 命令中指定以下文本：

```
-df "c d.txt" "a b.txt"
```

每个文件规范都必须为以下类别之一：

文件名称

针对运行源代理程序的系统以适当的表示法表示的文件名称。将文件名指定为源文件规范时，将复制该文件的内容。

目录

针对运行源代理程序的系统以适当的表示法表示的目录的名称。当目录指定为源文件规范时，将复制目录的内容。更准确来说，将复制目录及其所有子目录中的所有文件，包括隐藏的文件。

例如，要仅将 DIR1 的内容复制到 DIR2，请指定 **fteCreateTransfer ... -dd DIR2 DIR1/***

连续数据集

（仅限 z/OS）。顺序数据集或分区数据集成员的名称。通过在数据集名称之前加上两个正斜杠字符 (//) 来表示数据集。

如果指定协议网桥代理作为源代理，那么您之后无法指定数据集作为源文件规范。

分区数据集

（仅限 z/OS）。分区数据集的名称。通过在数据集名称之前加上两个正斜杠字符 (//) 来表示数据集名称。

如果指定协议网桥代理作为源代理，那么您之后无法指定数据集作为源文件规范。

Connect:Direct 节点上的文件名或目录

(仅限 Connect:Direct 网桥代理程序)。Connect:Direct 节点的名称、冒号字符 (:), 以及托管 Connect:Direct 节点的系统上的文件或目录路径。例如, *connect_direct_node_name:file_path*。

如果源代理是 Connect:Direct 网桥代理, 那么将仅接受此格式的源规范。

注: 如果源代理是 Connect:Direct 网桥代理, 那么文件路径中不支持通配符。

协议文件服务器上的文件名或目录

协议文件服务器的名称、冒号 (:), 以及协议服务器系统上的文件或目录路径。例如, *protocol_server:file_path*。

如果未指定协议服务器, 那么将使用缺省协议服务器。

- 如果指定 **-sq** 参数, 那么 *source_specification* 是源代理队列管理器上的本地队列的名称。您只能指定一个源队列。采用以下格式指定源队列:

```
QUEUE_NAME
```

源队列规范中未包含队列管理器名称, 因为队列管理器必须与源代理队列管理器相同。

- 如果源代理程序位于 z/OS 上, 那么假定以 // 开头的源文件是 z/OS 分区数据集。

其他参数

-? or -h

可选。显示命令语法。

示例

在此基本示例中, 文件 *originalfile.txt* 从 AGENT1 传输到同一系统上的 AGENT2, 并重命名为 *transferredfile.txt*

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -df C:\import\transferredfile.txt C:\export\originalfile.txt
```

在此示例中, 文件 *originalfile.txt* 和 *originalfile2.txt* 从 AGENT1 传输到同一系统上的 AGENT2, 并传输到目录 *C:\import*

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -dd C:\import C:\export\originalfile.txt  
C:\export\originalfile2.txt
```

在此示例中, 文件 *originalfile.txt* 从 AGENT1 的系统传输到 AGENT2 的系统。文件传输安排在源程序系统的系统时间 09:00 启动, 并且将每 2 小时发生 4 次:

```
fteCreateTransfer -sa AGENT1 -sm QM_JUPITER -da AGENT2 -dm QM_NEPTUNE  
-tb source -ss 09:00 -oi hours -of 2 -oc 4  
-df C:\import\transferredfile.txt C:\export\originalfile.txt
```

在此示例中, 文件 *originalfile.txt* 从 AGENT1 传输到 AGENT2, 条件是文件 *A.txt* 存在于 AGENT1:

```
fteCreateTransfer -sa AGENT1 -sm QM_JUPITER -da AGENT2 -dm QM_NEPTUNE  
-tr file=exist,C:\export\A.txt -df C:\import\transferredfile.txt C:\export\originalfile.txt
```

在此示例中, 文件 *originalfile.txt* 从 AGENT1 的系统传输到 AGENT2 的系统上的数据集 // "USERID.TRANS.FILE.TXT"。已选择文本方式来将数据从 ASCII 转换到 EBCDIC。

```
fteCreateTransfer -t text -sa AGENT1 -da AGENT2  
-ds "//TRANS.FILE.TXT;RECFM(V,B);BLKSIZE(6144);LRECL(1028);  
SPACE(5,1)" C:\export\originalfile.txt
```

在此示例中，AGENT1 的系统上标准数据集的一个成员将传输到 AGENT2 系统上的一个文件。已选择文本模式以将文件从 EBCDIC 转换为 AGENT2 的系统的缺省代码页。

```
fteCreateTransfer -t text -sa AGENT1 -da AGENT2 -df /tmp/IEEUJV.txt "'SYS1.SAMPLIB(IEEUJV)'"
```

在此示例中，使用目标代理 BRIDGE1 将代理 AGENT1 上名为 file.bin 的文件传输到协议文件服务器 accountshost.ibm.com 上名为 file.bin 的目标文件。

```
fteCreateTransfer -sa AGENT1 -da BRIDGE1 -df accountshost.ibm.com:/tmp/file.bin /tmp/file.bin
```

在此示例中，使用通配符时不带引号。AGENT1 当前工作目录中以 .txt 结尾的所有文件都将传输到 AGENT2 上的目录 C:\import。文件名保持不变。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -dd C:\import *.txt
```

在此示例中，使用通配符时带有双引号。AGENT1 的传输根目录中所有以 .txt 结尾的文件都将传输到 AGENT2 上的目录 C:\import。文件名保持不变。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -dd C:\import "*.txt"
```

在该示例中，属性是一次性指定的，指定时使用分号来连接值。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -dfa NAME1(VALUE1);NAME2(VALUE2) -dd c:\adx_test xyz.tx
```

在该示例中，在相同命令上多次指定了一个属性以允许传递多个属性。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -dfa NAME1(VALUE1) -dfa NAME2(VALUE2) -dd c:\adx_test xyz.tx
```

返回码

返回码	描述
0	命令成功完成。
1	命令结束但未成功。
2	命令由于超时而结束。命令向代理程序发送了一条消息，但代理程序未在指定的时间内响应。
20	命令已完成，部分成功且某些文件已传输。
21	在确定传输结果之前， fteCreateTransfer 命令所连接到的队列管理器已停止。
40	失败。未传输任何指定的文件。
41	已取消该传输。
42	传输未进行，因为传输是有条件的并且未满足所需条件。
43	传输请求消息格式不正确。
44	源代理没有足够的容量来执行该传输。
45	目标代理没有足够的容量来执行该传输。
46	正在传输的文件数超出源代理的限制。
47	已传输的文件数超出目标代理的限制。

相关概念

[第 222 页的『使用传输定义文件』](#)

您可以指定可用于创建文件传输的传输定义文件。传输定义文件是定义创建传输所需的部分或全部信息的 XML 文件。

相关任务

[第 221 页的『启动新的文件传输』](#)

您可以从 IBM MQ Explorer 或从命令行启动新的文件传输，并且可以选择传输组中的一个文件或多个文件。

[第 224 页的『创建调度的文件传输』](#)

您可以从 IBM MQ Explorer 或从命令行调度新的文件传输。调度的传输可以将单个文件或多个文件包含在一个组中。您可以将调度的文件传输执行一次或者将该传输重复多次。

[第 225 页的『触发文件传输』](#)

您可以设置某些关于文件传输的触发条件，这些触发条件必须成立才能进行传输。如果触发条件不成立，那么文件传输将无法进行并且可以选择提交日志消息以记录传输未发生这一事实。然后将废弃文件传输请求。例如，您可以将文件传输设置为仅在以下情况下发生：源代理所在的系统上的指定文件超过了指定大小，或者源代理所在的系统上存在某一特定指定文件。可以通过 IBM MQ Explorer 或命令行来设置已触发的文件传输。

fteCreateWebAgent (创建 IBM MQ Managed File Transfer Web 代理)

fteCreateWebAgent 命令创建代理及其关联的配置，以用于 Web 网关。IBM MQ Managed File Transfer 服务器随附该命令。

用途

使用 **fteCreateWebAgent** 命令以创建 Web 代理。该命令为您提供 MQSC 命令，必须在代理使用的队列管理器上运行这些 MQSC 命令才能创建以下代理队列：

- `SYSTEM.FTE.AUTHADM1.agent_name`
- `SYSTEM.FTE.AUTHAGT1.agent_name`
- `SYSTEM.FTE.AUTHMON1.agent_name`
- `SYSTEM.FTE.AUTHOPS1.agent_name`
- `SYSTEM.FTE.AUTHSCH1.agent_name`
- `SYSTEM.FTE.AUTHTRN1.agent_name`
- `SYSTEM.FTE.COMMAND.agent_name`
- `SYSTEM.FTE.DATA.agent_name`
- `SYSTEM.FTE.EVENT.agent_name`
- `SYSTEM.FTE.REPLY.agent_name`
- `SYSTEM.FTE.STATE.agent_name`

由于代理用于 Web 网关，因此除了先前列表外，还将创建两个队列：

- `SYSTEM.FTE.WEB.RESP.agent_name`
- `SYSTEM.FTE.WEB.gateway_name`

这些队列是内部系统队列，除非您要删除该代理，否则不得修改、删除或读取这些队列中的消息。以下位置的文件中也提供了要运行的 MQSC 命令：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_create.mqsc`。

如果稍后要删除代理，该命令还为您提供 MQSC 命令，必须运行这些 MQSC 命令才能清除然后删除属于代理的队列。MQSC 命令位于以下位置的文件中：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_delete.mqsc`。

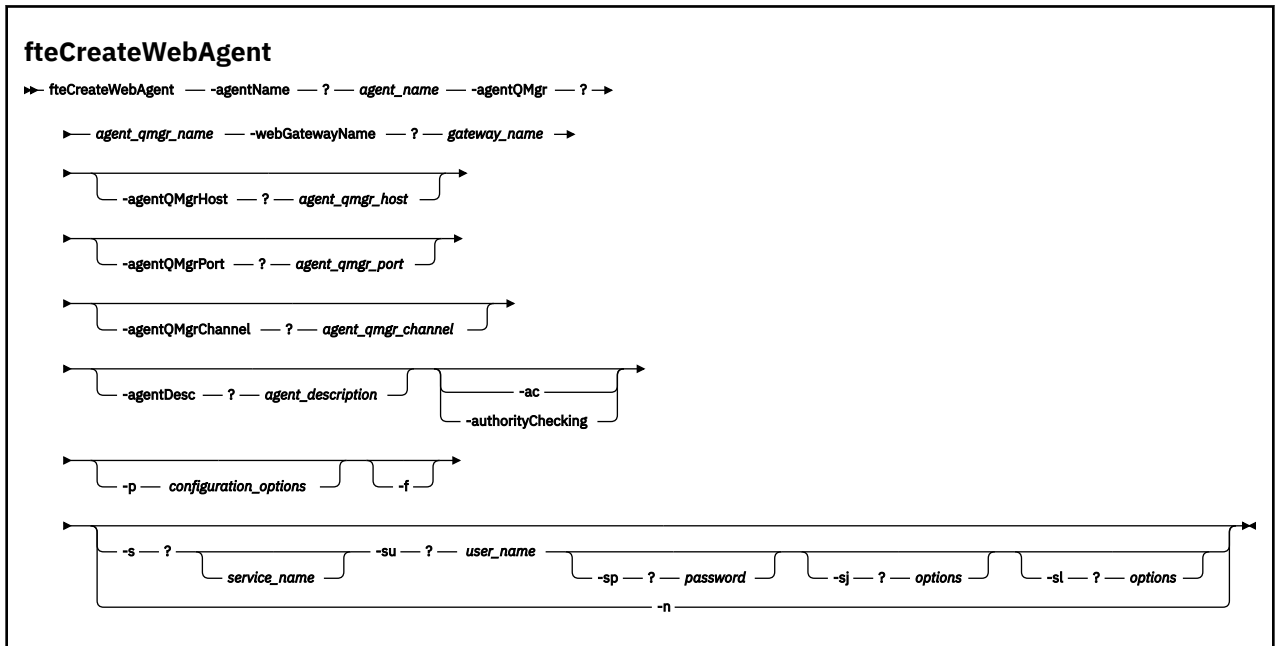
IBM MQ Managed File Transfer 提供帮助您配置代理的高级代理属性。[第 591 页的『agent.properties 文件』](#)中描述了这些属性。

注：运行 Web 代理所使用的用户必须与运行应用程序服务器所使用的用户相同或在同一个组中。

Web 代理的限制

- Web 代理只能是通过 Web 网关启动的传输的源代理。如果尝试通过另一种方法，将 Web 代理用作源代理来执行传输，那么传输将失败，返回码为 68 (TRANSFER_NOT_SUPPORTED)。
- 在将目标指定为文件空间时，Web 代理只能是传输的目标代理。如果尝试将 Web 代理用作目标代理（但作为不同目标类型）来执行传输，那么该传输将失败，并返回以下错误消息：BFGCH0103: 传输请求指定 Web 网关代理“web_agent_name”作为目标代理。Web 网关代理只能是到文件空间的传输的目标。
- Web 代理不能监视资源。如果尝试为 Web 代理创建资源监视器，那么该命令将失败，返回码为 113 (MONITOR_NOT_SUPPORTED)。

语法



参数

-agentName agent_name

必需。要创建的代理的名称。代理名称对其协调队列管理器必须唯一。

有关命名代理的更多信息，请参阅[对象命名约定](#)。

-agentQMGr agent_qmgr_name

必需。代理队列管理器的名称。

-webGatewayName gateway_name

必需。代理是其一个组件的 Web 网关的名称。

有关命名 Web 网关的更多信息，请参阅[对象命名约定](#)。

-agentQMGrHost agent_qmgr_host

可选。代理队列管理器的主机名或 IP 地址。如果未指定该参数，那么将采用绑定方式连接。

-agentQMGrPort agent_qmgr_port

可选。用于客户机与代理队列管理器的连接的端口号。只有当同时指定了 **agentQMGrHost** 参数时，才能使用该参数。如果未指定 **agentQMGrPort** 参数，那么将使用缺省端口 1414。

-agentQMGrChannel agent_qmgr_channel

可选。只有当同时指定了 **agentQMGrHost** 参数时，才能使用该参数。如果未指定 **agentQMGrChannel** 参数，那么将使用缺省通道 SYSTEM.DEF.SVRCONN。

-agentDesc agent_description

可选。在 IBM MQ Explorer 中显示的代理的描述。

-ac 或 -authorityChecking

可选。此参数允许进行权限检查。如果指定该参数，那么代理将检查提交请求的用户是否有权执行请求的操作。

-p configuration_options

可选。用于创建代理的配置选项集的名称。按照约定，这是协调队列管理器的名称。如果未指定该参数，那么会使用缺省的配置选项集。

-f

可选。强制该命令覆盖现有配置。

-s service_name

可选（仅限 Windows）。指示代理程序将作为 Windows 服务运行。如果未指定服务名称，那么服务将命名为 mqmftAgent<AGENT><QMGR>，其中 <AGENT> 是代理名称，<QMGR> 是代理队列管理器名称。

服务的显示名称（显示在 窗口 服务 窗口的 名称 列中）始终为 **IBM MQ Managed File Transfer 代理程序 <AGENT>@<QMGR>**。

-su user_name

可选（仅限 Windows）。当代理程序作为 Windows 服务运行时，该参数指定运行服务所使用的帐户名。要使用 Windows 域用户帐户来运行代理，请以 DomainName\UserName 格式指定该值。要使用本地内置域中的帐户来运行该服务，请以 UserName 格式指定该值。

使用 **-su** 参数指定的 Windows 用户帐户必须具有 **Log on as a service** 权限。有关如何授予该权限的信息，请参阅第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』。

在指定 **-s** 时为必需项。等同于 **-serviceUser**。

-sp password

可选（仅限 Windows）。由 **-su** 或 **-serviceUser** 参数设置的用户帐户的密码。

该参数仅在指定 **-s** 时才有效。等同于 **-servicePassword**。如果在指定 **-s** 参数时不指定该参数，将生成警告消息。该消息警告您必须先使用 Windows 服务工具设置密码，这样服务才能成功启动。

-sj options

可选（仅限 Windows）。当代理作为 Windows 服务启动时，以 -D 或 -X 格式来定义将传递至 JVM 的选项列表。这些选项使用井号 (#) 或分号 (;) 字符进行分隔。如果需要嵌入任何 # 或 ; 字符，请使用单引号将其引起来。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceJVMOptions**。

-sl options

可选（仅限 Windows）。设置 Windows 服务日志级别。有效选项包括：error、info、warn 和 debug。缺省值为 info。如果 Windows 服务出现问题，那么该选项可能很有用。如果将其设置为 debug，那么会在服务日志文件中提供更详细的信息。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceLogLevel**。

-n

可选（仅限 Windows）。指示代理将作为正常进程运行。这与 **-s** 选项互斥。如果既未指定 **-s** 选项，也未指定 **-n** 选项，那么会将代理程序配置为正常的 Windows 进程。

等同于 **-normal**。

-? or -h

可选。显示命令语法。

示例

在本示例中，使用代理队列管理器 QM_NEPTUNE 和 Web 网关 GATEWAY_ONE 来创建代理 WEBAGENT1。该代理使用缺省的协调队列管理器：

```
fteCreateWebAgent -agentName WEBAGENT1 -webGatewayName GATEWAY_ONE -agentQMGr QM_NEPTUNE  
-agentQMGrHost myhost.ibm.com -agentQMGrPort 1415 -agentQMGrChannel CHANNEL1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

[第 304 页的『IBM MQ Managed File Transfer Web 网关』](#)

Web 网关提供可用于与 IBM MQ Managed File Transfer 网络交互的 RESTful API。

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』](#)

您可以作为 Windows 服务来运行 IBM MQ Managed File Transfer 代理、独立数据库记录器和独立文件记录器。如果这些 Windows 服务出现任何问题，您可以使用服务日志文件以及本主题中的信息来诊断问题。

相关任务

[第 182 页的『准备部署 Web 网关』](#)

在部署 IBM MQ Managed File Transfer Web 网关之前，必须设置应用程序服务器环境和相关模块。本部分描述了 IBM MQ 和两个不同应用程序服务器的设置任务。

[第 197 页的『部署 IBM MQ Managed File Transfer Web 网关』](#)

必须将 IBM MQ Managed File Transfer Web 网关部署到与 Java Platform, Enterprise Edition 5 兼容的应用程序服务器。不同应用程序服务器的部署过程各不相同。本部分概述了两个应用程序服务器的部署过程。

[第 216 页的『将代理作为 Windows 服务启动』](#)

您可以将代理作为 Windows 服务启动，以便当您从 Windows 注销后，代理可以继续运行，并且可以接收文件传输。

相关参考

[第 421 页的『Web 代理无法启动』](#)

如果您收到来自 **fteStartAgent** 命令的错误，并且您正在尝试启动 Web 代理，请检查 SYSTEM.FTE.WEB.gateway_name 队列是否存在。

fteDefine（生成配置脚本）

使用 **fteDefine** 命令生成定义指定的代理队列管理器对象所需的配置脚本。

用途

当需要在对于包含配置数据的系统而言为远程的系统上运行一些配置步骤时，您会期望使用 **fteDefine** 命令。例如，在队列管理器上为代理配置要通过客户机连接访问的队列。

语法

fteDefine

►► **fteDefine** — **-t type** — **-d outputDir** — **name** — ►►

参数

-t type

必需。要定义的对象类型。类型选项为 **agent**。

-d outputDir

可选。脚本写入到的目录路径。如果未提供，会将脚本写入到标准输出流。

名称

必需。要定义的一个或多个对象名称。要为多个对象指定名称，请使用空格分隔。例如，*name1 name2...*

-? or -h

可选。显示命令语法。

示例

在此示例中，**fteDefine** 命令与 **-t agent** 参数和单个代理程序名称一起指定。输出将写入文件。

```
fteDefine -t agent EXAMPLE.AGENT >EXAMPLE.AGENT_create.mqsc
```

此命令生成的输出是要对代理队列管理器运行的且用于创建必需的代理队列的 MQSC 命令脚本：

```
$ fteDefine -t agent EXAMPLE.AGENT
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
DEFINE QLOCAL(SYSTEM.FTE.COMMAND.EXAMPLE.AGENT) +
  DEFPRTY(0) +
  DEFSOPT(SHARED) +
  GET(ENABLED) +
  MAXDEPTH(5000) +
  MAXMSGL(4194304) +
  MSGDLVSQ(PRIORITY) +
  PUT(ENABLED) +
  RETINTVL(999999999) +
  SHARE +
  NOTRIGGER +
  USAGE(NORMAL) +
  REPLACE
DEFINE QLOCAL(SYSTEM.FTE.DATA.EXAMPLE.AGENT) +
  DEFPRTY(0) +
  DEFSOPT(SHARED) +
  GET(ENABLED) +
  MAXDEPTH(5000) +
  MAXMSGL(4194304) +
  MSGDLVSQ(PRIORITY) +
  PUT(ENABLED) +
  RETINTVL(999999999) +
  SHARE +
  NOTRIGGER +
  USAGE(NORMAL) +
  REPLACE
...
etc.
```

在本示例中，指定 **fteDefine** 命令时会指定 **-d outputDir** 参数和多个代理名称。

```
fteDefine -t agent -d /tmp EXAMPLE.AGENT.1 EXAMPLE.AGENT.2 EXAMPLE.AGENT.3
```

此命令生成的输出是 MQSC 命令脚本位置的绝对文件路径：

```
$ fteDefine -t agent -d /tmp EXAMPLE.AGENT.1 EXAMPLE.AGENT.2 EXAMPLE.AGENT.3
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
BFGCM0239I: A file has been created containing the MQSC definitions to define the agent
EXAMPLE.AGENT.1.
The file can be found here: '/tmp/EXAMPLE.AGENT.1_create.mqsc'.
BFGCM0239I: A file has been created containing the MQSC definitions to define the agent
EXAMPLE.AGENT.2.
The file can be found here: '/tmp/EXAMPLE.AGENT.2_create.mqsc'.
BFGCM0239I: A file has been created containing the MQSC definitions to define the agent
EXAMPLE.AGENT.3.
The file can be found here: '/tmp/EXAMPLE.AGENT.3_create.mqsc'.
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关参考

第 521 页的『[fteDelete](#)（生成用于除去配置脚本）』

使用 **fteDelete** 命令可生成除去指定的代理队列管理器对象所需的配置脚本。

fteDelete（生成用于除去配置脚本）

使用 **fteDelete** 命令可生成除去指定的代理队列管理器对象所需的配置脚本。

用途

当需要在对于包含配置数据的系统而言为远程的系统上运行一些配置步骤时，您会期望使用 **fteDelete** 命令。例如，移除本地队列管理器上远程客户机代理的队列。

语法

fteDelete

```
➔ fteDelete -t type -d outputDir name ➔
```

参数

-t type

必需。要删除的对象的类型。类型选项为 **agent**。

-d outputDir

可选。脚本写入到的目录路径。如果未提供，会将脚本写入到标准输出流。

名称

必需。要删除的一个或多个对象名称。要为多个对象指定名称，请使用空格分隔。例如，*name1 name2...*

-? or -h

可选。显示命令语法。

示例

在此示例中，**fteDelete** 命令与 **-t agent** 参数和单个代理程序名称一起指定。输出将写入文件。

```
fteDelete -t agent EXAMPLE.AGENT >EXAMPLE.AGENT_delete.mqsc
```

此命令生成的输出是要对代理队列管理器运行的且用于删除代理队列的 MQSC 命令脚本：

```
$ fteDelete -t agent EXAMPLE.AGENT
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
CLEAR QLOCAL(SYSTEM.FTE.COMMAND.EXAMPLE.AGENT)
DELETE QLOCAL(SYSTEM.FTE.COMMAND.EXAMPLE.AGENT)
CLEAR QLOCAL(SYSTEM.FTE.DATA.EXAMPLE.AGENT)
DELETE QLOCAL(SYSTEM.FTE.DATA.EXAMPLE.AGENT)
CLEAR QLOCAL(SYSTEM.FTE.REPLY.EXAMPLE.AGENT)
DELETE QLOCAL(SYSTEM.FTE.REPLY.EXAMPLE.AGENT)
CLEAR QLOCAL(SYSTEM.FTE.STATE.EXAMPLE.AGENT)
DELETE QLOCAL(SYSTEM.FTE.STATE.EXAMPLE.AGENT)
CLEAR QLOCAL(SYSTEM.FTE.EVENT.EXAMPLE.AGENT)
DELETE QLOCAL(SYSTEM.FTE.EVENT.EXAMPLE.AGENT)
...
etc.
```

在此示例中，**fteDelete** 命令与 **-d outputDir** 参数和多个代理程序名称一起指定。

```
fteDelete -t agent -d /tmp EXAMPLE.AGENT.1 EXAMPLE.AGENT.2 EXAMPLE.AGENT.3
```

此命令生成的输出是 MQSC 命令脚本位置的绝对文件路径：

```
$ fteDelete -t agent -d /tmp EXAMPLE.AGENT.1 EXAMPLE.AGENT.2 EXAMPLE.AGENT.3
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
BFGCM0241I: A file has been created containing the MQSC definitions to delete the agent
EXAMPLE.AGENT.1.
The file can be found here: '/tmp/EXAMPLE.AGENT.1_delete.mqsc'.
BFGCM0241I: A file has been created containing the MQSC definitions to delete the agent
EXAMPLE.AGENT.2.
The file can be found here: '/tmp/EXAMPLE.AGENT.2_delete.mqsc'.
BFGCM0241I: A file has been created containing the MQSC definitions to delete the agent
EXAMPLE.AGENT.3.
The file can be found here: '/tmp/EXAMPLE.AGENT.3_delete.mqsc'.
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关参考

第 519 页的『**fteDefine**（生成配置脚本）』

使用 **fteDefine** 命令生成定义指定的代理队列管理器对象所需的配置脚本。

fteDeleteAgent（删除 IBM MQ Managed File Transfer 代理）

fteDeleteAgent 命令删除 IBM MQ Managed File Transfer 代理及其配置。如果该代理是协议网桥代理，那么用户凭证文件将留在文件系统中。

用途

在运行 **fteDeleteAgent** 命令之前，使用 [fteStopAgent](#) 命令停止代理。

如果已将代理程序配置为作为 Windows 服务运行，那么运行 **fteDeleteAgent** 命令将删除服务定义。只有 IBM MQ 管理员（和 mqm 组的成员）身份的用户才可以运行此命令。如果尝试以非 IBM MQ 管理员用户身份运行此命令，那么您将收到一条错误消息，并且该命令将不会运行。

fteDeleteAgent 命令向您提供 MQSC 命令，您必须针对代理的队列管理器运行这些命令以清除和删除代理的系统队列。这些队列如下所示：

- SYSTEM.FTE.AUTHADM1.agent_name
- SYSTEM.FTE.AUTHAGT1.agent_name

- SYSTEM.FTE.AUTHMON1.agent_name
- SYSTEM.FTE.AUTHOPS1.agent_name
- SYSTEM.FTE.AUTHSCH1.agent_name
- SYSTEM.FTE.AUTHTRN1.agent_name
- SYSTEM.FTE.COMMAND.agent_name
- SYSTEM.FTE.DATA.agent_name
- SYSTEM.FTE.EVENT.agent_name
- SYSTEM.FTE.REPLY.agent_name
- SYSTEM.FTE.STATE.agent_name

如果您的代理是 Web 代理，那么还必须删除另外两个队列。 **fteDeleteAgent** 命令将清除并删除以下队列：

- SYSTEM.FTE.WEB.RESP.agent_name

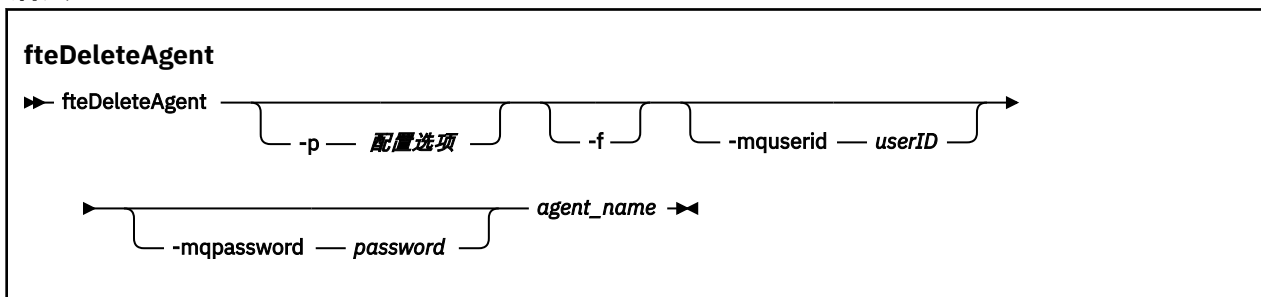
The **fteDeleteAgent** command does not delete the SYSTEM.FTE.WEB.<gateway_name> queue, because this queue is shared between multiple web agents. 为 Web 代理运行 **fteDeleteAgent** 命令之后，您必须手动删除 SYSTEM.FTE.WEB.网关名称 队列。

注：仅在已删除与此 Web 网关相关联的所有 Web 代理的情况下，才删除 SYSTEM.FTE.WEB.网关名称 队列。

fteCreateAgent 命令还在以下位置的文件中提供这些命令：

MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/agent_name_delete.mqsc

语法



参数

-p (配置选项)

可选。如果有多个协调队列管理器，请使用此参数来显式指定希望删除哪个代理配置。使用非缺省协调队列管理器的名称作为该参数的输入。然后，该命令将使用与该非缺省协调队列管理器关联的配置选项。

仅在您希望使用不同于缺省值的配置选项时，才指定可选的 **-p** 参数。如果未指定 **-p**，那么将使用 `installation.properties` 文件中定义的配置选项。请参阅第 116 页的『[分布式平台上的配置选项](#)』，以了解更多信息。

-f

可选。强制该命令从协调队列管理器注销该代理，即使找不到该代理的配置文件也是如此。由于在此情况下，该代理的队列管理器的相关信息不可用，因此该命令将直接连接到协调队列管理器，而不是像通常那样使用代理队列管理器。

-mquserid (userID)

可选。指定通过代理队列管理器进行认证时所使用的用户标识，除非存在 **-f** 强制参数。如果存在 **-f** 参数，将会指定通过协调队列管理器进行认证时所需的用户标识。

-mqpassword (password)

可选。指定通过代理队列管理器进行认证时所使用的密码，除非存在 **-f** 强制参数。如果存在 **-f** 参数，将会指定通过协调队列管理器进行认证时所需的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

agent_name

必需。要删除的代理的名称。

-? or -h

可选。显示命令语法。

示例

在此示例中，将在协调队列管理器 QM_COORD1 上删除 AGENT3 及其配置：

```
fteDeleteAgent -p QM_COORD1 AGENT3
```

此示例命令输出以下 MQSC 命令，用于删除该代理的三个队列：

```
CLEAR QLOCAL(SYSTEM.FTE.COMMAND.AGENT3)
DELETE QLOCAL(SYSTEM.FTE.COMMAND.AGENT3)
CLEAR QLOCAL(SYSTEM.FTE.DATA.AGENT3)
DELETE QLOCAL(SYSTEM.FTE.DATA.AGENT3)
CLEAR QLOCAL(SYSTEM.FTE.REPLY.AGENT3)
DELETE QLOCAL(SYSTEM.FTE.REPLY.AGENT3)
CLEAR QLOCAL(SYSTEM.FTE.STATE.AGENT3)
DELETE QLOCAL(SYSTEM.FTE.STATE.AGENT3)
CLEAR QLOCAL(SYSTEM.FTE.EVENT.AGENT3)
DELETE QLOCAL(SYSTEM.FTE.EVENT.AGENT3)
CLEAR QLOCAL(SYSTEM.FTE.AUTHADM1.AGENT3)
DELETE QLOCAL(SYSTEM.FTE.AUTHADM1.AGENT3)
CLEAR QLOCAL(SYSTEM.FTE.AUTHAGT1.AGENT3)
DELETE QLOCAL(SYSTEM.FTE.AUTHAGT1.AGENT3)
CLEAR QLOCAL(SYSTEM.FTE.AUTHTRN1.AGENT3)
DELETE QLOCAL(SYSTEM.FTE.AUTHTRN1.AGENT3)
CLEAR QLOCAL(SYSTEM.FTE.AUTHOPS1.AGENT3)
DELETE QLOCAL(SYSTEM.FTE.AUTHOPS1.AGENT3)
CLEAR QLOCAL(SYSTEM.FTE.AUTHSCH1.AGENT3)
DELETE QLOCAL(SYSTEM.FTE.AUTHSCH1.AGENT3)
CLEAR QLOCAL(SYSTEM.FTE.AUTHMON1.AGENT3)
DELETE QLOCAL(SYSTEM.FTE.AUTHMON1.AGENT3)
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关参考

[第 576 页的『fteStopAgent（停止 IBM MQ Managed File Transfer 代理）』](#)

使用 **fteStopAgent** 命令以受控方式停止 IBM MQ Managed File Transfer 代理程序，或者使用 **-i** 参数在必要时立即停止代理程序。

[第 455 页的『fteCleanAgent（清除 IBM MQ Managed File Transfer 代理）』](#)

使用 **fteCleanAgent** 命令通过从代理使用的持久和非持久队列中删除消息来清除 IBM MQ Managed File Transfer 代理使用的队列。如果您在启动代理时存在问题，并且该问题可能是由于代理使用的队列上保留的信息所造成的，那么可以使用 **fteCleanAgent** 命令。

[第 460 页的『fteCreateAgent（创建 IBM MQ Managed File Transfer 代理）』](#)

fteCreateAgent 命令创建代理及其关联的配置。

[第 573 页的『fteStartAgent（启动 IBM MQ Managed File Transfer 代理）』](#)

fteStartAgent 命令从命令行启动 IBM MQ Managed File Transfer 代理程序。

fteDeleteLogger (删除 IBM MQ Managed File Transfer 记录器)

使用 **fteDeleteLogger** 命令可删除 IBM MQ Managed File Transfer 记录器及其配置。可以保留或删除与该记录器关联的现有日志文件。

要点: 在分布式系统上，只有 IBM MQ 管理员（和 mqm 组的成员）身份的用户才可以运行此命令。如果尝试以非 IBM MQ 管理员用户身份运行此命令，那么您将收到错误消息 BFGCL0502E: You are not authorized to perform the requested operation. 并且不会运行该命令。

在 z/OS 系统上，用户必须满足以下至少一个条件才能运行迁移命令：

- 是 mqm 组的成员（如果在系统上定义了 mqm 组）。
- **V8.0.0.6** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V8.0.0.6** 在运行命令时，在 BFG_GROUP_NAME 环境变量中未设置任何值。

用途

在运行 **fteDeleteLogger** 命令之前，使用 **fteStopLogger** 命令停止该记录器。

如果已将记录器配置为作为 Windows 服务运行，那么运行 **fteDeleteLogger** 命令将删除服务定义。

记录器配置目录包含用于删除记录器的队列和预订的 MQSC 脚本。这些队列如下所示：

- SYSTEM.FTE.LOG.CMD.*logger_name*
- SYSTEM.FTE.LOG.RJCT.*logger_name*

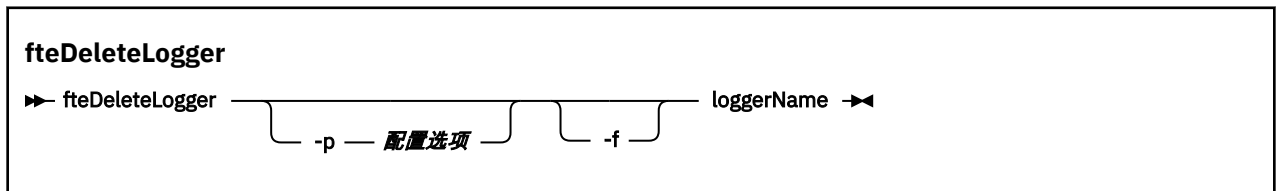
预订名称如下所示：

- SYSTEM.FTE.AUTO.SUB.*logger_name*

MQSC 脚本可在

MQ_DATA_PATH\mqft\config\coordination_qmgr\loggers*logger_name**logger_name_delete*.mqsc 中找到

语法



参数

-p configuration_options

可选。确定用于启动独立数据库记录器的一组配置选项。使用一组配置选项的名称作为 **-p** 参数的值。按照约定，该值是协调队列管理器的名称。如果未指定该参数，那么会使用缺省的配置选项集。

-f

可选。强制除去此记录器创建的任何日志文件。如果省略此参数，那么将保留由记录器创建的任何日志文件，当不再需要这些日志文件时，必须手工将其除去。

logger_name

必需。要删除的记录器的名称。

-? or -h

可选。显示命令语法。

示例

在此示例中，将删除称为 logger1 的记录器。已指定 **-f** 参数，该参数将除去记录器的日志文件以及记录器的配置文件。

```
fteDeleteLogger -f logger1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关参考

第 474 页的『[fteCreateLogger（创建 IBM MQ Managed File Transfer 记录器）](#)』
使用 **fteCreateLogger** 命令可创建文件或数据库记录器。

第 575 页的『[fteStartLogger（启动记录器）](#)』

fteStartLogger 命令启动 IBM MQ Managed File Transfer 日志记录应用程序。

第 579 页的『[fteStopLogger（停止记录器）](#)』

fteStopLogger 命令停止记录器。

第 548 页的『[fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）](#)』

使用 **fteModifyLogger** 命令修改记录器，以便记录器能够作为 Windows 服务运行。只能在 Windows 平台上使用此命令，必须由作为 IBM MQ 管理员和 mqm 组成员的用户运行，并且必须首先使用

fteStopLogger 命令停止记录器。

fteDeleteMonitor（删除 IBM MQ Managed File Transfer 资源监视器）

使用 **fteDeleteMonitor** 命令可通过命令行停止和删除现有 IBM MQ Managed File Transfer 资源监视器。针对资源监视代理发出此命令。

用途

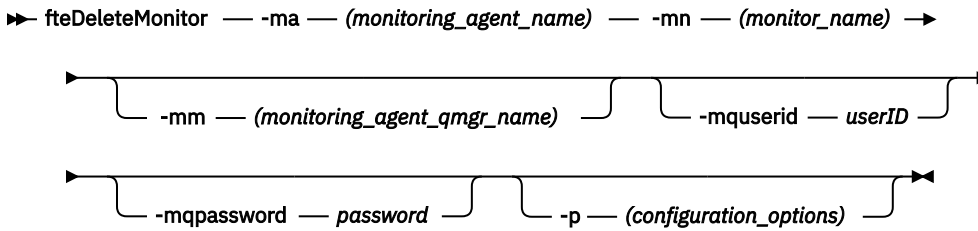
使用 **fteDeleteMonitor** 命令可停止监视资源并从监视代理移除监视器的定义。运行此命令时，将不再进行资源轮询，也不再启动其他任务。

您可以从可连接到 WebSphere MQ 网络的任何系统中运行 **fteDeleteMonitor** 命令，然后路由到代理的队列管理器。具体而言，要运行此命令，您必须已在该系统上安装了 IBM MQ Managed File Transfer 组件（服务或代理程序），并且将该系统的 IBM MQ Managed File Transfer 配置为与 WebSphere MQ 网络通信。如果无任何可用的连接详细信息，那么会改为将代理队列管理器详细信息用于连接，前提是这些详细信息可用。

仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。请参阅[配置选项](#)，以获取更多信息。

语法

fteDeleteMonitor



参数

-ma (monitoring_agent_name)

必需。执行资源监视的代理的名称。此监视代理还必须是您要触发的文件传输的源代理。

-mn (monitor_name)

必需。您分配给此资源监视器的名称。您可以删除一个资源监视器，然后创建同名的新监视器。

-mm (monitoring_agent_qmgr_name)

可选。监视代理的队列管理器的名称。由于监视器触发的传输的监视代理和源代理必须相同，因此该队列管理器也是源代理的队列管理器。

-mquserid (userID)

可选。指定要向命令队列管理器进行认证的用户标识。

-mqpassword (password)

可选。指定要向命令队列管理器进行认证的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

-p (配置选项)

可选。该参数确定用于取消传输的配置选项集。按照约定，请使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-? or -h

可选。显示命令语法。

示例

在此示例中，将删除带有监视（和文件传输源代理）AGENT1 的资源监视器 MONITOR1：

```
fteDeleteMonitor -ma AGENT1 -mm QM_JUPITER -mn MONITOR1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

第 229 页的『资源监控』

您可以监视 IBM MQ Managed File Transfer 资源；例如，队列或目录。当满足针对该资源的条件时，资源监视器将启动任务，如文件传输。您可以使用 **fteCreateMonitor** 命令或 IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

相关参考

第 478 页的『[fteCreateMonitor](#)（新建资源监视器）』

fteCreateMonitor 命令从命令行创建并启动新的资源监视器。您可以使用 WebSphere MQ Managed File Transfer 监视资源（例如，目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

第 534 页的『[fteListMonitors](#)（列出 IBM MQ Managed File Transfer 资源监视器）』

使用 **fteListMonitors** 命令可使用命令行列出 IBM MQ Managed File Transfer 网络中的所有现有资源监视器。

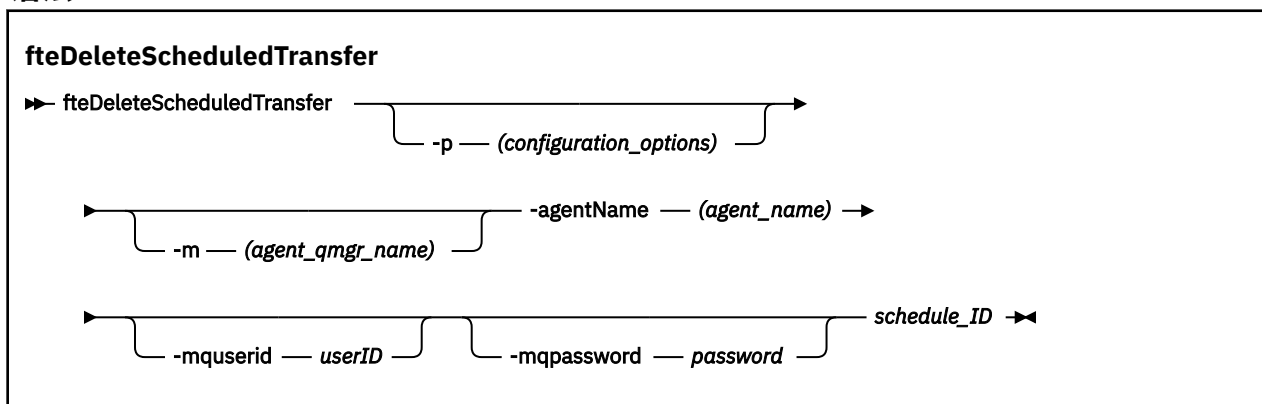
fteDeleteScheduledTransfer（删除调度的文件传输）

用途

使用 **fteDeleteScheduledTransfer** 命令可删除先前使用命令行或 WebSphere MQ Explorer 创建的 IBM MQ Managed File Transfer 调度传输。

仅当您要使用不同于缺省值的配置选项时，才可为该命令指定可选的 **-p** 参数。如果未指定 **-p**，那么将使用在 `installation.properties` 中定义的配置选项。请参阅第 116 页的『[分布式平台上的配置选项](#)』，以了解更多信息。

语法



参数

-p（配置选项）

可选。如果有多个协调队列管理器，请使用此参数来显式指定希望删除哪个调度的传输。使用非缺省协调队列管理器的名称作为该参数的输入。然后，该命令将使用与该非缺省协调队列管理器关联的配置选项。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项。

-m (*agent_qmgr_name*)

可选。源代理连接到的队列管理器的名称。如果未指定该参数，那么将根据使用中的配置选项来确定代理的队列管理器。

-agentName (*agent_name*)

必需。希望从中删除调度传输的源代理的名称。

-mquserid (*userID*)

可选。指定要向命令队列管理器进行认证的用户标识。

-mqpassword (*password*)

可选。指定要向命令队列管理器进行认证的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

schedule_ID

必需。希望删除的调度传输的标识。

可通过针对源代理的名称运行 `fteListScheduledTransfers` 命令来查找调度标识。

-? or -h
可选。显示命令语法。

示例

在本示例中，将删除源代理 AGENT2 上标识为 27 的调度传输：

```
fteDeleteScheduledTransfer -agentName AGENT2 27
```

返回码

- 0**
命令成功完成。
- 1**
命令结束但未成功。

相关任务

第 224 页的『[创建调度的文件传输](#)』
您可以从 IBM MQ Explorer 或从命令行调度新的文件传输。调度的传输可以将单个文件或多个文件包含在一个组中。您可以将调度的文件传输执行一次或者将该传输重复多次。

相关参考

第 536 页的『[fteListScheduledTransfers](#)（列出调度的文件传输）』
使用 **fteListScheduledTransfers** 命令可列出先前使用命令行或 WebSphere MQ Explorer 创建的所有 IBM MQ Managed File Transfer 传输。

fteDeleteTemplates（删除 IBM MQ Managed File Transfer 模板）

使用 **fteDeleteTemplates** 命令可从协调队列管理器中删除现有的 IBM MQ Managed File Transfer 模板。

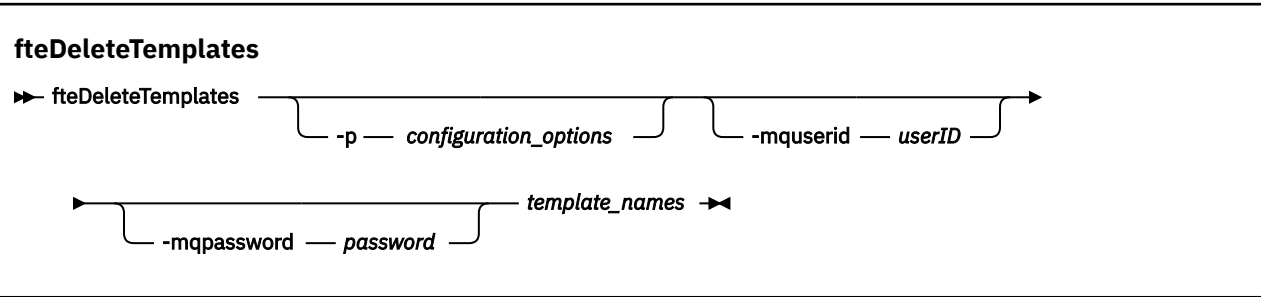
用途

fteDeleteTemplates 命令会从协调队列管理器中移除一个或多个文件传输模板。运行此命令时，会将一个请求传递到 IBM MQ 系统以从协调队列管理器中移除模板，这样，该模板将不再可用于 IBM MQ Explorer 或命令行。在该命令完成后但在 IBM MQ 系统处理该请求之前，有一小段时间可能可以继续访问您所删除的模板。

您可以从连接到 IBM MQ 网络并路由到协调队列管理器的任何系统中运行 **fteDeleteTemplates** 命令。具体而言，要运行此命令，您必须已在该系统上安装了 IBM MQ Managed File Transfer，并将该系统的 IBM MQ Managed File Transfer 配置为与 IBM MQ 网络进行通信。如果无任何可用的连接详细信息，那么会改为将代理队列管理器详细信息用于连接，前提是这些详细信息可用。

仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。请参阅[配置选项](#)，以获取更多信息。

语法



参数

-p (配置选项)

可选。该参数确定用于删除该模板的配置选项集。按照约定，请使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-mquserid (userID)

可选。指定通过协调队列管理器进行认证时所使用的用户标识。

-mqpassword (password)

可选。指定通过协调队列管理器进行认证时所使用的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

(template_names)

必需。指定希望删除的一个或多个模板的名称。根据 **fteListTemplates** 命令显示的内容来指定名称。

-? or -h

可选。显示命令语法。

示例

该本示例中，将删除模板 STANDBY：

```
fteDeleteTemplates STANDBY
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关任务

[第 248 页的『使用传输模板』](#)

您可以使用文件传输模板来存储重复或复杂传输的公共文件传输设置。使用 **fteCreateTemplate** 命令从命令行创建传输模板，或者使用 IBM MQ Explorer 通过 "为受管文件传输创建新模板" 向导创建传输模板，或者通过选中 **将传输设置另存为模板** 复选框在创建文件传输时保存模板。传输模板窗口显示在您的 IBM MQ Managed File Transfer 网络中已创建的所有传输模板。

[第 249 页的『使用 IBM MQ Explorer 创建文件传输模板』](#)

您可以从 IBM MQ Explorer 或从命令行创建文件传输模板。然后，您可以使用该模板来利用模板详细信息创建新的文件传输，或者提交该模板以启动文件传输。

相关参考

[第 484 页的『fteCreateTemplate（新建文件传输模板）』](#)

fteCreateTemplate 命令创建文件传输模板，您可以保留该模板以供将来使用。唯一的必需参数是 **-tn** (template_name) 参数。其他所有参数都是可选的，但如果指定源文件，还必须提供目标文件。类似地，如果指定目标文件，还必须指定源文件规范。

[第 537 页的『fteListTemplates（列出 IBM MQ Managed File Transfer 模板）』](#)

fteListTemplates 命令可用于列出协调队列管理器上的可用 IBM MQ Managed File Transfer 传输模板。

fteDisplayVersion（显示 IBM MQ Managed File Transfer 的版本）

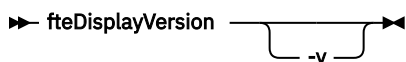
使用 **fteDisplayVersion** 命令来显示您已安装的 IBM MQ Managed File Transfer 的版本。

用途

IBM 服务代表可能会请求您运行 **fteDisplayVersion** 命令以帮助确定问题。

语法

fteDisplayVersion



参数

-v

可选。显示有关产品版本的详细信息。

在指定 **-v** 参数时显示的确切详细信息可能会因产品发行版而异。建议您不要依赖于 **fteDisplayVersion -v** 命令的输出中提供具体信息。

-? or -h

可选。显示命令语法。

示例

在本示例中，指定 **fteDisplayVersion** 命令时不带任何参数。

```
fteDisplayVersion
```

此命令的输出为产品版本级别。例如，V7.5，如下所示：

```
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Name:      WebSphere MQ Managed File Transfer
Version:    7.5
```

在本示例中，指定 **fteDisplayVersion** 命令时会指定 **-v** 参数。

```
fteDisplayVersion -v
```

此命令的输出为有关产品版本的以下更多详细信息：

```
C:\Program Files\IBM\WebSphere MQ\bin>fteDisplayVersion.cmd -v
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Name:      WebSphere MQ Managed File Transfer
Version:    7.5
Level:      f000-20120518-1027
Platform:   Windows 7 (6.1 build 7601 Service Pack 1)
Architecture: x86
JVM:        JRE 1.6.0 IBM J9 2.4 Windows 7 x86-32 jvmwi3260sr10fp1-20120202_101568 (JIT enabled, AOT
enabled)
           JVM - 20120202_101568
           JIT  - r9_20111107_21307ifix1
           GC   - 20120202_AA
Product:     C:\Program Files\IBM\WebSphere MQ
Configuration: C:\Program Files\IBM\WebSphere MQ\mqft

WebSphere MQ Components:

Name:        Java Message Service Client
Version:     7.5.0.0
Level:       p000-L120520

Name:        WebSphere MQ classes for Java Message Service
Version:     7.5.0.0
Level:       p000-L120520

Name:        IBM WebSphere MQ JMS Provider
Version:     7.5.0.0
Level:       p000-L120520

Name:        Common Services for Java Platform, Standard Edition
```

返回码

0

命令成功完成。

1

命令结束但未成功。

fteListAgents (列出协调队列管理器的 IBM MQ Managed File Transfer 代理)

通过命令行，使用 **fteListAgents** 命令可列出向特定协调队列管理器注册的所有 IBM MQ Managed File Transfer 代理。

用途

您可以从连接到协调队列管理器的任何系统运行 **fteListAgents** 命令。每个代理的下列详细信息将指向标准输出设备 (STDOUT):

- 代理程序名称
- 代理队列管理器
- 如果代理是协议网桥代理，那么代理名称将附加 (FTP bridge) 或 (SFTP bridge)
- 如果代理是 Web 代理，那么代理名称后会追加 (Web Gateway)
- 如果代理是 Connect:Direct 网桥代理，那么代理名称将追加 (Connect:Direct bridge)
- 代理状态

该命令使用 `coordination.properties` 文件连接到协调队列管理器。有关更多信息，请参阅第 584 页的『`coordination.properties` 文件』。

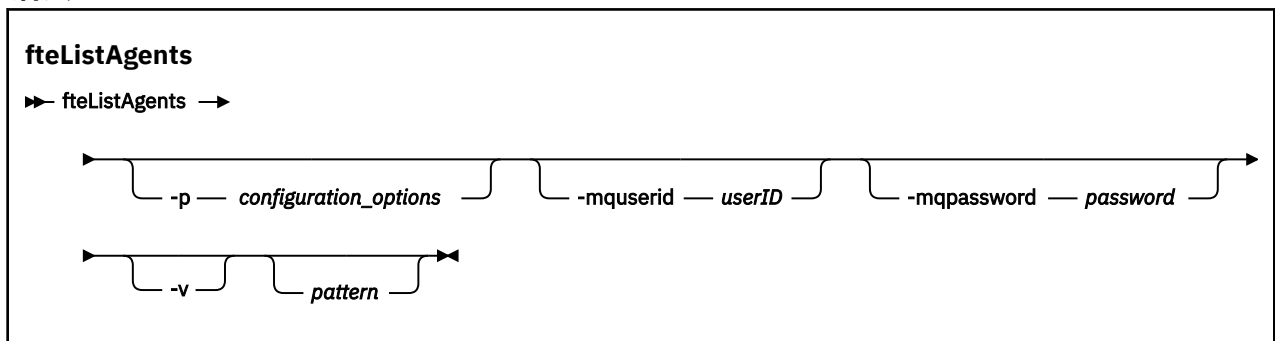
仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。有关更多信息，请参阅第 116 页的『分布式平台上的配置选项』。

如果 **fteListAgents** 命令未列出代理程序，请使用以下主题中的诊断流程图来查找并解决问题: [如果 fteListAgents 命令未列出代理程序](#)。

代理状态信息

该命令生成的代理状态信息是通过代理发布到 SYSTEM.FTE 主题的状态消息生成的。主题第 647 页的『代理状态消息格式』中描述了这些消息。**fteListAgents** 命令产生的状态信息给出了发布最后一条状态消息时的代理状态。这些状态消息的频率取决于 `agentStatusPublishRateLimit` 属性的值。有关该属性的更多详细信息，请参阅主题第 591 页的『`agent.properties` 文件』。

语法



参数

- p (配置选项)**
可选。该参数确定一组用于发出列出代理请求的配置选项。使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。
如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。
- mquserid (userID)**
可选。指定通过协调队列管理器进行认证时所使用的用户标识。
- mqpassword (password)**
可选。指定通过协调队列管理器进行认证时所使用的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。
- v**
可选。指定详细方式。详细方式生成每个代理的其他输出，包括格式为 Source/Destination 的当前传输数量，其中 Source 是当前源传输数量，Destination 是当前目标传输数量。
当前传输信息通过代理状态发布获得，这在以下主题中进行了描述：第 647 页的『代理状态消息格式』。因此，该传输信息仅精确到 [agentStatusPublishRateLimit](#) 代理属性值（缺省为 30 秒）设置。

pattern

- 可选。用于过滤 IBM MQ Managed File Transfer 代理列表的模式。该模式与代理名称匹配。星号 (*) 字符解释为与任何值（包括零个字符）匹配的通配符。
在 UNIX 和 Linux 系统上，必须使用引号 (") 或双引号 (") 对星号 (*) 和数字符号 (#) 之类的特殊字符进行转义 如果要将它们作为字面值处理。如果未对这些字符进行转义，那么将根据它们在特定 UNIX 或 Linux 系统上的含义对它们进行解释。
如果未指定该参数，那么会列出所有向协调队列管理器注册的代理。

-? or -h

- 可选。显示命令语法。

示例

在该示例中，列出在队列管理器（在配置选项中详述）上注册并且名称以 B 开头的所有代理：

```
fteListAgents "B*"
```

在该示例中，以详细方式列出向协调队列管理器 QM_EUROPE（非缺省协调队列管理器）注册的代理：

```
fteListAgents -p QM_EUROPE -v
```

此命令的输出如下所示：

Agent Name:	Queue Manager Name:	Transfers: (Source/Destination)	Status:
BERLIN	QM_BERLIN	7/0	RUNNING
LONDON	QM_LONDON	0/0	RUNNING
MADRID	QM_MADRID	0/1	UNREACHABLE

有关可能的代理状态值及其含义的列表，请参阅主题第 700 页的『代理状态值』。

在该示例中，以详细方式列出向协调队列管理器注册并且名称以 BRIDGE 开头的所有代理：

```
fteListAgents -v "BRIDGE*"
```

此命令的输出如下所示：

```
C:\Program Files\IBM\WMQFTE\bin>fteListAgents -v
```

5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED

Agent Name:	Queue Manager Name:	Transfers:	Status:
		(Source/Destination)	
BRIDGE_FTP (FTP bridge)	QM_JUPITER	0/0	STOPPED
BRIDGE_CD1 (Connect:Direct bridge)	QM_JUPITER	0/0	STOPPED

返回码

- 0
命令成功完成。
- 1
命令结束但未成功。

相关任务

第 273 页的『列出 IBM MQ Managed File Transfer 代理』
可使用命令行或 IBM MQ Explorer 来列出已向特定队列管理器注册的代理。

相关参考

第 700 页的『代理状态值』
fteListAgents 和 **fteShowAgentDetails** 命令生成代理状态信息。 该状态有多种可能的值。
第 564 页的『fteShowAgentDetails（显示 IBM MQ Managed File Transfer 代理详细信息）』
使用 **fteShowAgentDetails** 命令可显示特定 IBM MQ Managed File Transfer 代理的详细信息。 这些是由其 IBM MQ Managed File Transfer 协调队列管理器存储的详细信息。
第 380 页的『当 fteListAgents 命令显示代理状态为 UNREACHABLE 时应如何处理』
您的代理正在运行，并且可以成功响应 **ftePingAgent** 命令，文件正常传输，但是该代理被 **fteListAgents** 命令列为 UNREACHABLE。

fteListMonitors（列出 IBM MQ Managed File Transfer 资源监视器）

使用 **fteListMonitors** 命令可使用命令行列出 IBM MQ Managed File Transfer 网络中的所有现有资源监视器。

用途

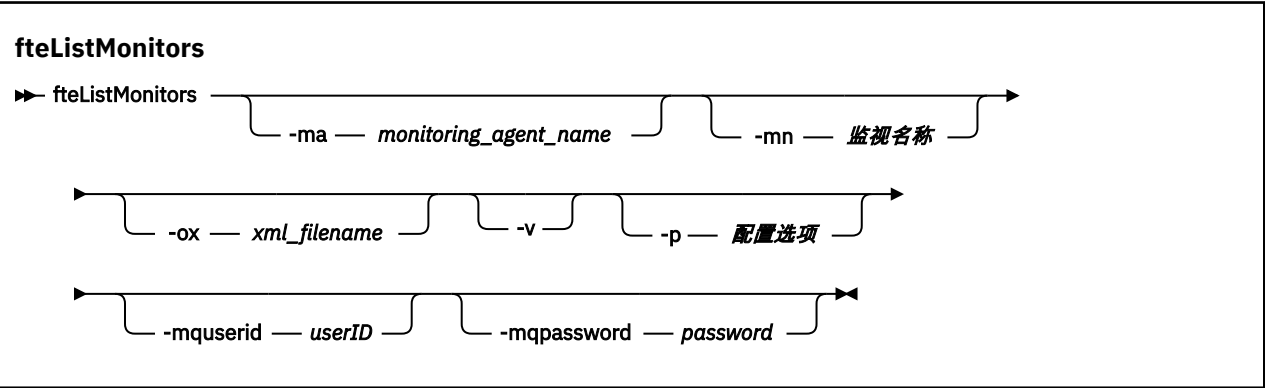
fteListMonitors 命令用于列出现有资源监视器。 您可以通过指定代理名称和资源监视器名称来过滤命令输出。

该命令使用 `coordination.properties` 文件连接到协调队列管理器。 有关更多信息，请参阅第 584 页的『`coordination.properties` 文件』。

您可以使用 **-ox** 参数将资源监视器导出到 XML 文件。 请参阅第 478 页的『fteCreateMonitor（新建资源监视器）』，了解有关如何使用该 XML 文件的信息。

仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。 请参阅配置选项，以获取更多信息。

语法



参数

-ma (*monitoring_agent_name*)

可选。使用您作为输入提供的模式，按代理名称过滤资源监视器。星号 (*) 字符解释为与零个或多个字符匹配的通配符。如果未指定 **-ma** 参数，那么缺省情况下会列出与缺省协调队列管理器的所有代理关联的所有资源监视器。

-mn (*monitor_name*)

可选。使用您作为输入提供的模式，按监视器名称过滤资源监视器。星号 (*) 字符解释为与零个或多个字符匹配的通配符。如果未指定 **-mn** 参数，那么缺省情况下会列出与缺省协调队列管理器的所有代理关联的所有资源监视器。

-ox (*xml_filename*)

可选。必须将该参数与 **-ma** 和 **-mn** 参数一起指定。将资源监视器导出到 XML 文件，然后可通过 **fteCreateMonitor** 命令使用该文件。

-v

可选。生成包含监视器状态相关附加信息的详细输出，其中包括监视器是已启动还是已停止、监视的目录资源路径及触发条件。

-p (配置选项)

可选。该参数确定用于取消传输的配置选项集。按照约定，请使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-mquserid (*userID*)

可选。指定通过协调队列管理器进行认证时所使用的用户标识。

-mqpassword (*password*)

可选。指定通过协调队列管理器进行认证时所使用的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

-? or -h

可选。显示命令语法。

示例

在该示例中，列出了与监视代理（以及与监视器关联的文件传输的源代理）AGENT1 关联的所有资源监视器：

```
fteListMonitors -ma AGENT1
```

在该示例中，AGENT1 上的资源监视器 MONITOR1 导出到 XML 文件 filename1.xml：

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -ox filename1.xml
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

第 229 页的『资源监控』

您可以监视 IBM MQ Managed File Transfer 资源；例如，队列或目录。当满足针对该资源的条件时，资源监视器将启动任务，如文件传输。您可以使用 **fteCreateMonitor** 命令或 IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

相关参考

第 478 页的『[fteCreateMonitor](#)（新建资源监视器）』

fteCreateMonitor 命令从命令行创建并启动新的资源监视器。您可以使用 WebSphere MQ Managed File Transfer 监视资源（例如，目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

第 526 页的『[fteDeleteMonitor](#)（删除 IBM MQ Managed File Transfer 资源监视器）』

使用 **fteDeleteMonitor** 命令可通过命令行停止和删除现有 IBM MQ Managed File Transfer 资源监视器。针对资源监视代理发出此命令。

fteListScheduledTransfers（列出调度的文件传输）

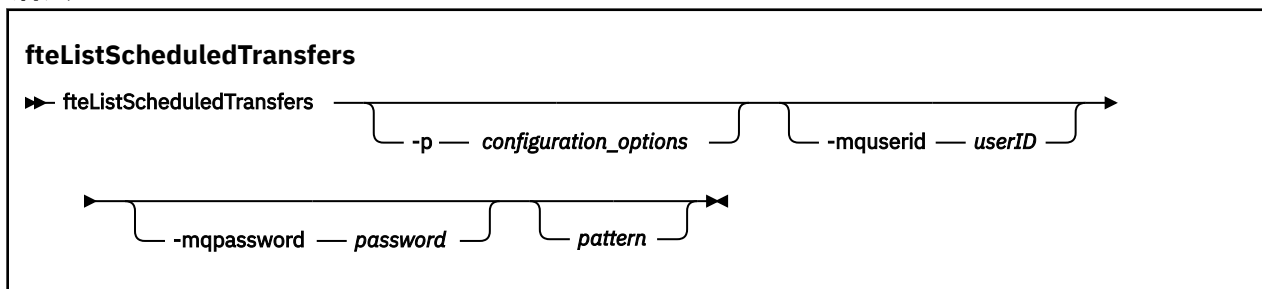
使用 **fteListScheduledTransfers** 命令可列出先前使用命令行或 WebSphere MQ Explorer 创建的所有 IBM MQ Managed File Transfer 传输。

用途

可以根据源代理名称或协调队列管理器，列出所有调度的传输。

仅当您要使用不同于缺省值的配置选项时，才可为该命令指定可选的 **-p** 参数。如果未指定 **-p**，那么将使用 `installation.properties` 中定义的配置选项。请参阅第 116 页的『[分布式平台上的配置选项](#)』，以了解更多信息。

语法



参数

-p（配置选项）

可选。如果有多个协调队列管理器，使用该参数可显式指定要为其列出调度传输的代理。使用非缺省协调队列管理器的名称作为该参数的输入。然后，该命令将使用与该非缺省协调队列管理器关联的配置选项。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项。

-mquserid (*userID*)

可选。指定通过协调队列管理器进行认证时所使用的用户标识。

-mqpassword (*password*)

可选。指定通过协调队列管理器进行认证时所使用的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

pattern

可选。用于过滤 IBM MQ Managed File Transfer 调度传输列表的模式。该模式与源代理名称匹配。星号 (*) 字符解释为与零个或更多个字符匹配的通配符。

如果未指定该参数，那么缺省情况下会列出所有向协调队列管理器注册的调度传输。

-? or -h

可选。显示命令语法。

示例

在该示例中，列出源代理与模式 *2 匹配的所有调度传输：

```
fteListScheduledTransfers "*2"
```

该示例命令产生以下输出。调度开始时间和下一个传输时间以全球标准时间 (UTC) 显示：

```
Schedule Identifier:      1
Source Agent Name:       AGENT2
Source File Name:        C:/export/Test/workspace/A.exe
Conversion Type:         binary
Destination File Name:    C:/import/Test/workspace/B001.zzx
Destination Agent Name:   AGENT1
Schedule Start Time:     2008-10-23T16:08+0100
Next Transfer:           2008-10-23T16:08+0100
Schedule Time Base:      source
Repeat Interval:         minutes
Repeat Frequency:        1
Repeat Count:            30
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关任务

第 224 页的『创建调度的文件传输』

您可以从 IBM MQ Explorer 或从命令行调度新的文件传输。调度的传输可以将单个文件或多个文件包含在一个组中。您可以将调度的文件传输执行一次或者将该传输重复多次。

相关参考

第 528 页的『fteDeleteScheduledTransfer（删除调度的文件传输）』

fteListTemplates（列出 IBM MQ Managed File Transfer 模板）

fteListTemplates 命令可用于列出协调队列管理器上的可用 IBM MQ Managed File Transfer 传输模板。

用途

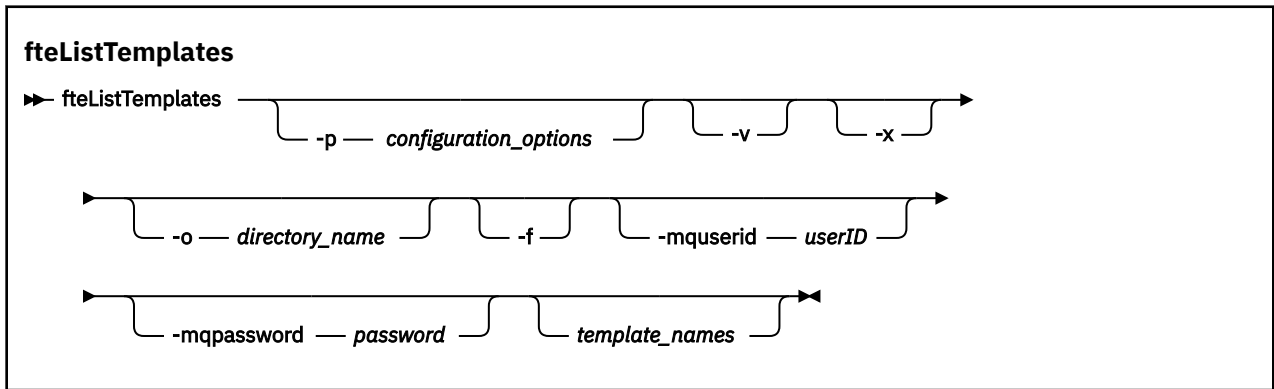
该命令列出所有模板名称或过滤的模板名称选择。列表的输出格式可以是以下任何一种：

- 仅模板名称（缺省行为）
- 模板名称和模板摘要（详细方式）
- 描述模板的完整 XML 消息（**-x** 和 **-o** 参数）

该命令使用 `coordination.properties` 文件连接到协调队列管理器。有关更多信息，请参阅第 584 页的『`coordination.properties` 文件』。

仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。请参阅[配置选项](#)，以获取更多信息。

语法



参数

-p

可选。该参数确定用于删除该模板的配置选项集。按照约定，请使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-v

可选。指定详细方式并提供每个匹配模板的简短摘要。如果您还指定了 **-x** 参数，那么会忽略该参数。

-v 参数包含每个模板的摘要。例如：

```
Template Name: STANDBY
Source Agent Name: AGENT1
Source QMgr: QM_JUPITER
Destination Agent Name: AGENT2
Destination QMgr: QM_NEPTUNE
Transfer Priority: 0
Transfer file specification
File Item Details
  Mode: binary
  Checksum: MD5
  Source File:
    C:\payroll_reports\*.xls
  Recursive: false
  Disposition: leave
  Destination File:
    C:\payroll_backup\*.xls
  Type: file
  Exist: error
```

如果未指定 **-v** 参数，那么缺省输出方式为列出匹配的模板名称。

-x

可选。提供每个匹配模板的 XML 格式消息。将忽略该参数，除非您还指定 **-o** 参数。

-o (directory_name)

可选。将 XML 格式消息发送到指定目录中的文件。会为每个模板创建一个文件，每个文件的名称与模板名称相同，后缀为 .xml。将忽略该参数，除非您还指定 **-x** 参数。

-f

可选。强制覆盖任何现有的输出文件。将忽略该参数，除非您还指定 **-o** 参数。如果未指定 **-f**，但指定了现有输出文件的名称，那么缺省行为是报告错误并继续。

-mquserid (userID)

可选。指定通过协调队列管理器进行认证时所使用的用户标识。

-mqpassword (password)

可选。指定通过协调队列管理器进行认证时所使用的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

(*template_names*)

可选。要列出的一个或多个模板名称的列表。模板名称可以包含星号，作为与零个或更多字符匹配的通配符。根据操作系统，您可能需要对包含通配符的任何模板名称加双引号 (") 或单引号 (')，以避免 shell 扩展。Shell 扩展会导致意外行为。

如果没有为 *template_names* 指定任何值，那么缺省行为是列出所有模板。

-? or -h

可选。显示命令语法。

示例

在该示例中，列出名称以 ST 开头的所有模板：

```
fteListTemplates "ST*"
```

该示例以 XML 格式消息创建模板 STANDBY，该消息将发送到当前目录中的文件 STANDBY.xml：

```
fteListTemplates -x -o . STANDBY
```

该命令在 STANDBY.xml 中创建以下输出：

```
<?xml version="1.0" encoding="UTF-8" ?>
- <transferTemplate id="1864c1dd-ba02-4b34-bda9-dc6862448418" version="3.00">
  <name>STANDBY</name>
  <sourceAgentName>AGENT1</sourceAgentName>
  <sourceAgentQMgr>QM_JUPITER</sourceAgentQMgr>
  <sourceAgentQMgrHost>null</sourceAgentQMgrHost>
  <sourceAgentQMgrPort>-1</sourceAgentQMgrPort>
  <sourceAgentQMgrChannel>null</sourceAgentQMgrChannel>
  <destinationAgentName>AGENT2</destinationAgentName>
  <destinationAgentQMgr>QM_NEPTUNE</destinationAgentQMgr>
- <fileSpecs>
  - <item checksumMethod="MD5" mode="binary">
    - <source disposition="leave" recursive="false">
      <file>C:\payroll_reports\*.xls</file>
    </source>
    - <destination exist="error" type="file">
      <file>C:\payroll_backup\*.xls</file>
    </destination>
  </item>
</fileSpecs>
<priority>0</priority>
</transferTemplate>
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关任务

[第 248 页的『使用传输模板』](#)

您可以使用文件传输模板来存储重复或复杂传输的公共文件传输设置。使用 **fteCreateTemplate** 命令从命令行创建传输模板，或者使用 IBM MQ Explorer 通过 "为受管文件传输创建新模板" 向导创建传输模板，或者通过选中 **将传输设置另存为模板** 复选框在创建文件传输时保存模板。传输模板窗口显示在您的 IBM MQ Managed File Transfer 网络中已创建的所有传输模板。

[第 249 页的『使用 IBM MQ Explorer 创建文件传输模板』](#)

您可以从 IBM MQ Explorer 或从命令行创建文件传输模板。然后，您可以使用该模板来利用模板详细信息创建新的文件传输，或者提交该模板以启动文件传输。

相关参考

[第 484 页的『fteCreateTemplate（新建文件传输模板）』](#)

fteCreateTemplate 命令创建文件传输模板，您可以保留该模板以供将来使用。唯一的必需参数是 **-tn** (*template_name*) 参数。其他所有参数都是可选的，但如果指定源文件，还必须提供目标文件。类似地，如果指定目标文件，还必须指定源文件规范。

第 529 页的『[fteDeleteTemplates](#)（删除 IBM MQ Managed File Transfer 模板）』

使用 **fteDeleteTemplates** 命令可从协调队列管理器中删除现有的 IBM MQ Managed File Transfer 模板。

fteMigrate 代理 (将 WebSphere MQ File Transfer Edition V7.0 代理迁移到 IBM WebSphere MQ V7.5 或更高版本)

如果要将现有代理及其关联配置从 WebSphere MQ File Transfer Edition 的任何版本迁移至 IBM WebSphere MQ V7.5 或更高版本，请使用 **fteMigrateAgent** 命令进行迁移。该命令可用于迁移标准代理、Connect:Direct 代理、协议网桥代理或 Web 代理。该命令还可用于在单个请求中迁移多个代理。

注：如果要从 7.0 或更高版本迁移，并且希望继续使用 FTE_CONFIG 环境变量，您可以在不更改 FTE_CONFIG 值的情况下执行此操作。您可以执行标准迁移，但不能设置 BFG_DATA，并且必须按照 7.0 中的用法设置 FTE_CONFIG。

要点：在分布式系统上，只有 IBM MQ 管理员（和 mqm 组的成员）身份的用户才可以运行此命令。如果尝试以非 IBM MQ 管理员用户身份运行此命令，那么您将收到错误消息 BFGCL0502E: You are not authorized to perform the requested operation. 并且不会运行该命令。

在 z/OS 系统上，用户必须满足以下至少一个条件才能运行迁移命令：

- 是 mqm 组的成员（如果在系统上定义了 mqm 组）。
- **V8.0.0.6** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V8.0.0.6** 在运行命令时，在 BFG_GROUP_NAME 环境变量中未设置任何值。

如果代理程序配置为作为 Windows 服务运行，请使用 **fteModifyAgent** 命令来重新配置代理程序，以使其不再是 Windows 服务。迁移完成之后，再次使用 **fteModifyAgent** 命令将新代理配置为 Windows 服务。或者，如果包含 **-f** 参数，那么此命令会完成运行，但会生成一条警告。

在运行 **fteMigrateAgent** 命令之前，必须使用 **fteStopAgent** 命令来停止要迁移的代理。

如果运行带有 **-f** 参数的此命令，那么仅会刷新关于该代理的信息。如果缺少必需的文件，那么该命令将失败。

具体而言，将迁移以下与代理关联的属性文件、XML 文件和目录：

表 42: fteMigrateAgent 命令迁移的代理文件	
fteMigrateAgent 命令为每个代理迁移的文件的名称	信息
wmqfte.properties	wmqfte.properties 文件在 IBM WebSphere MQ V7.5 或更高版本中重命名为 installation.properties。
command.properties	
coordination.properties	
coordination_queue_manager.mqsc	
agent_name_create.mqsc	
agent_name_delete.mqsc	
exits 目录	该命令会复制 exits 目录中的所有文件。
仅适用于标准代理和 Web 代理：	
UserSandboxes.xml	
仅适用于 Connect:Direct 网桥代理程序：	

表 42: `fteMigrateAgent` 命令迁移的代理文件 (继续)

fteMigrateAgent 命令为每个代理迁移的文件的名称	信息
ConnectDirectCredentials.xml	
ConnectDirectNodeProperties.xml	
ConnectDirectProcessDefinitions.xml	
仅适用于协议网桥代理：	
ProtocolBridgeCredentials.xml	
ProtocolBridgeProperties.xml	该文件仅存在于 WebSphere MQ File Transfer Edition V7.0.4.1 或更高版本上。

fteMigrateAgent 命令用于迁移安装，协调和命令队列管理器的文件，并将其复制到 IBM WebSphere MQ V7.5 或更高版本 (如果这些文件在 V7.5 或更高版本上尚不存在)。如果这些文件已存在，那么它们不会作为该命令的一部分被复制。

语法

fteMigrateAgent

```
►► fteMigrateAgent — -agentName — — 代理名称 — -config — — configuration_directory →
```

► -credentialPath — — *credentials_path* — *-f* ►

-p — configuration options

参数

-agentName *agent name*

必需。要迁移到 IBM WebSphere MQ V7.5 或更高版本的代理的名称。

-config *configuration_directory*

必需。要从中迁移代理程序的安装版的配置目录路径。例如，C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config

-credentialPath *credentials_path*

必需。定义要将凭证信息迁移到的位置。此参数可以是现有凭证文件存在于的目录路径，也可以是接收新凭证文件的新位置。对于 z/OS 平台，这可以是预先存在的扩展分区数据集 (PDSE)，具有要更新的现有成员，或者无现有成员，以针对这些凭证包含一个新成员。

注: 如果使用 PDSE, 它必须是分块变量。

-f

可选。即使一般情况下迁移的某些配置文件与现有配置冲突，也强制代理进行迁移。例如，如果 WebSphere MQ File Transfer Edition 上的属性文件与 IBM WebSphere MQ V7.5 或更高版本上的属性文件不匹配，那么指定 **-f** 参数意味着将忽略此不匹配。

-p *configuration_options*

可选。该参数确定用于查找要迁移的配置的配置选项集。将一个配置选项集的名称用作 **-p** 参数的值。按照约定，这是协调队列管理器的名称。如果未指定该参数，那么会使用缺省的配置选项集。有关更多信息，请参阅第 116 页的『分布式平台上的配置选项』。

-? or -h

可选。显示命令语法。

示例

在此示例中，AGENT3 及其在 /var/ibm/WMQFTE/config 中的配置将迁移到 IBM WebSphere MQ V7.5 或更高版本：

```
fteMigrateAgent -agentName AGENT3 -config /var/ibm/WMQFTE/config -credentialPath /home/user1/AGENT3
```

在此示例中，C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config 中的所有代理程序及其配置都将迁移到 IBM WebSphere MQ V7.5 或更高版本。Windows 文件路径包含在双引号 (") 中。指定了 **-f** 参数以强制迁移并忽略任何属性文件不匹配：

```
fteMigrateAgent -agentName "*" -config "C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config" -credentialPath "C:\Documents and Settings\user1\AGENT3" -f
```

返回码

0

命令成功完成。

1

命令结束但未成功。

有关返回码的更多信息，请参阅第 405 页的『IBM MQ Managed File Transfer 的返回码』。

相关任务

第 29 页的『将 WebSphere MQ File Transfer Edition V7.0 代理迁移至 IBM WebSphere MQ V7.5 或更高版本』

使用 **fteMigrateAgent** 命令将 WebSphere MQ File Transfer Edition V7.0 代理迁移至 IBM WebSphere MQ V7.5 或更高版本。如果已将代理配置为作为 Windows 服务运行，那么必须完成本任务中的可选步骤。

相关参考

第 26 页的『WebSphere MQ File Transfer Edition V7.0.4 或更低版本与 IBM WebSphere MQ V7.5 或更高版本之间的更改』

如果计划从 WebSphere MQ File Transfer Edition V7.0.4 或更早版本移至 IBM WebSphere MQ V7.5 或更高版本，请查看以下信息，其中总结了各版本之间的变化。

第 542 页的『fteMigrateConfigurationOptions（将 WebSphere MQ File Transfer Edition V7.0 配置迁移至 IBM WebSphere MQ V7.5 或更高版本）』

fteMigrateConfigurationOptions 命令从 WebSphere MQ File Transfer Edition V7.0 迁移一组配置选项，并将其复制到 IBM WebSphere MQ V7.5 或更高版本，前提是这些文件在 V7.5 或更高版本上尚不存在。如果这些文件已存在，那么会输出一条消息，而且命令无法继续。

fteMigrateConfigurationOptions（将 WebSphere MQ File Transfer Edition V7.0 配置迁移至 IBM WebSphere MQ V7.5 或更高版本）

fteMigrateConfigurationOptions 命令从 WebSphere MQ File Transfer Edition V7.0 迁移一组配置选项，并将其复制到 IBM WebSphere MQ V7.5 或更高版本，前提是这些文件在 V7.5 或更高版本上尚不存在。如果这些文件已存在，那么会输出一条消息，而且命令无法继续。

注：如果要从 7.0 或更高版本迁移，并且希望继续使用 FTE_CONFIG 环境变量，您可以在不更改 FTE_CONFIG 值的情况下执行此操作。您可以执行标准迁移，但不能设置 BFG_DATA，并且必须按照 7.0 中的用法设置 FTE_CONFIG。

要点：在分布式系统上，只有 IBM MQ 管理员（和 mqm 组的成员）身份的用户才可以运行此命令。如果尝试以非 IBM MQ 管理员用户身份运行此命令，那么您将收到错误消息 BFGCL0502E: You are not authorized to perform the requested operation. 并且不会运行该命令。

在 z/OS 系统上，用户必须满足以下至少一个条件才能运行迁移命令：

- 是 mqm 组的成员（如果在系统上定义了 mqm 组）。
- **V8.0.0.6** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V8.0.0.6** 在运行命令时，在 BFG_GROUP_NAME 环境变量中未设置任何值。

语法

fteMigrateConfigurationOptions

```
► fteMigrateConfigurationOptions — -config — — configuration_directory — -credentialPath — — ►  
  
► — credentials_path — -configurationOptionsName — — configuration_options_name ►
```

参数

-config (*configuration_directory*)

必需。要从中进行迁移的安装的配置目录路径。例如，C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config

-credentialPath (*credentials_path*)

必需。定义要将凭证信息迁移到的位置。此参数可以是现有凭证文件存在于的目录路径，也可以是接收新凭证文件的新位置。对于 z/OS 平台，这可以是预先存在的扩展分区数据集 (PDSE)，具有要更新的现有成员，或者无现有成员，以针对这些凭证包含一个新成员。

注: 如果使用 PDSE，那么它必须是可变块。

-configurationOptionsName (*configuration_options_name*)

必需。要迁移的配置选项集的名称。您可以通过使用星号字符 (*) 来表示零个或零个以上字符，以迁移多个配置选项集。您可以将星号与字符串结合使用。例如，要迁移名称以 IBM 开头的所有配置选项集，请按如下所示使用此参数: -configurationOptionsName IBM*。

示例

在此示例中，将迁移目录 C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config 中的所有配置。目录路径用双引号括起：

```
fteMigrateConfigurationOptions -config "C:\Documents and Settings\All Users\Application  
Data\IBM\WMQFTE\config"  
-credentialPath "C:\Documents and Settings\user1\configurationoptions" -configurationOptionsName *
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关参考

第 26 页的『[WebSphere MQ File Transfer Edition V7.0.4 或更低版本与 IBM WebSphere MQ V7.5 或更高版本之间的更改](#)』

如果计划从 WebSphere MQ File Transfer Edition V7.0.4 或更早版本移至 IBM WebSphere MQ V7.5 或更高版本，请查看以下信息，其中总结了各版本之间的变化。

第 540 页的『[fteMigrate 代理 \(将 WebSphere MQ File Transfer Edition V7.0 代理迁移到 IBM WebSphere MQ V7.5 或更高版本\)](#)』

如果要将现有代理及其关联配置从 WebSphere MQ File Transfer Edition 的任何版本迁移至 IBM WebSphere MQ V7.5 或更高版本，请使用 **fteMigrateAgent** 命令进行迁移。该命令可用于迁移标准代理、Connect:Direct 代理、协议网桥代理或 Web 代理。该命令还可用于在单个请求中迁移多个代理。

第 544 页的『[fteMigrateLogger \(将 WebSphere MQ File Transfer Edition V7.0 数据库记录器迁移至 IBM WebSphere MQ V7.5 或更高版本\)](#)』

如果要将现有独立数据库记录器的配置从 WebSphere MQ File Transfer Edition V7.0.1 或更高版本迁移到 IBM WebSphere MQ V7.5 或更高版本，请使用 **fteMigrateLogger** 命令。

fteMigrateLogger（将 WebSphere MQ File Transfer Edition V7.0 数据库记录器迁移至 IBM WebSphere MQ V7.5 或更高版本）

如果要将现有独立数据库记录器的配置从 WebSphere MQ File Transfer Edition V7.0.1 或更高版本迁移到 IBM WebSphere MQ V7.5 或更高版本，请使用 **fteMigrateLogger** 命令。

您无法使用该命令来迁移 JEE 数据库记录器：请改为使用将 WebSphere Application Server V7 JEE 数据库记录器从 WMQFTE V7.0 迁移至 WMQ V7.5 或更高版本中的信息。

注: 如果要从 7.0 或更高版本迁移，并且希望继续使用 FTE_CONFIG 环境变量，您可以在不更改 FTE_CONFIG 值的情况下执行此操作。您可以执行标准迁移，但不能设置 BFG_DATA，并且必须按照 7.0 中的用法设置 FTE_CONFIG。

要点: 在分布式系统上，只有 IBM MQ 管理员（和 mqm 组的成员）身份的用户才可以运行此命令。如果尝试以非 IBM MQ 管理员用户身份运行此命令，那么您将收到错误消息 BFGCL0502E: You are not authorized to perform the requested operation. 并且不会运行该命令。

在 z/OS 系统上，用户必须满足以下至少一个条件才能运行迁移命令：

- 是 mqm 组的成员（如果在系统上定义了 mqm 组）。
- **V8.0.0.6** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V8.0.0.6** 在运行命令时，在 BFG_GROUP_NAME 环境变量中未设置任何值。

如果已将独立数据库记录器配置为作为 Windows 服务运行，那么无法使用 **fteMigrateLogger** 命令来迁移该记录器的配置。如果在配置为作为 Windows 服务运行的记录器上运行 **fteMigrateLogger** 命令，那么该命令将产生错误并且不会继续。或者，如果包含 **-f 参数**，那么此命令会完成运行，但会生成一条警告。

在运行 **fteMigrateLogger** 命令之前，请在 WebSphere MQ File Transfer Edition V7.0 上使用 **fteStopDatabaseLogger** 命令来停止要迁移其配置的数据库记录器。

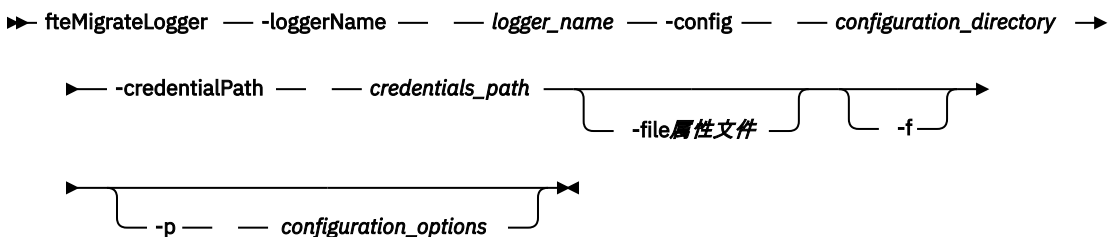
如果运行带有 **-f 参数** 的此命令，那么仅会刷新关于此记录器的信息。如果缺少必需的文件，那么该命令将失败。具体而言，将迁移以下与记录器配置关联的属性文件和 .mqsc 文件：

表 43: fteMigrateLogger 命令迁移的文件	
fteMigrateLogger 命令迁移的文件的名称	信息
wmqfte.properties	wmqfte.properties 文件基于 IBM WebSphere MQ V7.5, 或更高版本中的 installation.properties
command.properties	
coordination.properties	
coordination_queue_manager.mqsc	
使用 -file 参数指定的 databaselogger.properties 或其他属性文件	databaselogger.properties 用于在 IBM WebSphere MQ V7.5 或更高版本中创建 logger.properties 文件。

fteMigrateLogger 命令会迁移安装，协调和命令队列管理器的文件，并将其复制到 IBM WebSphere MQ V7.5 或更高版本，前提是这些文件在 V7.5 或更高版本上尚不存在。如果这些文件已存在，那么它们不会作为该命令的一部分被复制。

语法

fteMigrateLogger



参数

-loggerName *logger name*

必需。您希望为 IBM WebSphere MQ V7.5 或更高版本中的已迁移记录器配置指定的名称。有关记录器名称（针对 V7.5 的新名称）的更多信息，请参阅 `logger_name` 参数。

-config *configuration directory*

必需。要从中迁移记录器配置的安装的配置目录路径。

-credentialPath *credentials path*

必需。定义要将凭证信息迁移到的位置。此参数可以是现有凭证文件存在于的目录路径，也可以是接收新凭证文件的新位置。对于 z/OS 平台，这可以是预先存在的扩展分区数据集 (PDSE)，具有要更新的现有成员，或者无现有成员，以针对这些凭证包含一个新成员。

注: 如果使用 PDSE, 那么它必须是可变块。

-file *properties_file*

可选。指定要迁移的数据库记录器属性文件。 仅当属性文件不使用以下缺省名称和路径时，该参数才必需：`configuration_directory/coordination_qmgr_name/databaselogger.properties`

-f

可选。即使一般情况下迁移的某些配置文件与现有配置冲突，也强制进行迁移。例如，如果 WebSphere MQ File Transfer Edition 上的数据库记录器属性文件与 IBM WebSphere MQ V7.5 或更高版本上的属性文件不匹配，那么指定 **-f** 参数意味着将忽略此不匹配。

-p *configuration_options*

可选。该参数确定用于查找要迁移的记录器配置的配置选项集。将一个配置选项集的名称用作 **-p** 参数的值。按照约定，这是协调队列管理器的名称。如果未指定该参数，那么会使用缺省的配置选项集。要了解更多信息，请参阅第 116 页的『分布式平台上的配置选项』。

-? or -h

可选。显示命令语法。

示例

在此示例中，位于 `/var/ibm/WMQFTE/config` 中的独立数据库记录器的配置将迁移到 IBM WebSphere MQ V7.5，并命名为 `FTELOGGER1`：

```
fteMigrateLogger -loggerName FTELOGGER1 -config /var/ibm/WMQFTE/config
-credentialPath /home/user1/FTELOGGER1
```

返回码

O

命令成功完成。

命令结束但未成功。

运行 fteMigrateLogger 命令之后

相关参考

如果计划从 WebSphere MQ File Transfer Edition V7.0.4 或更早版本移至 IBM WebSphere MQ V7.5 或更高版本，请查看以下信息，其中总结了各版本之间的变化。

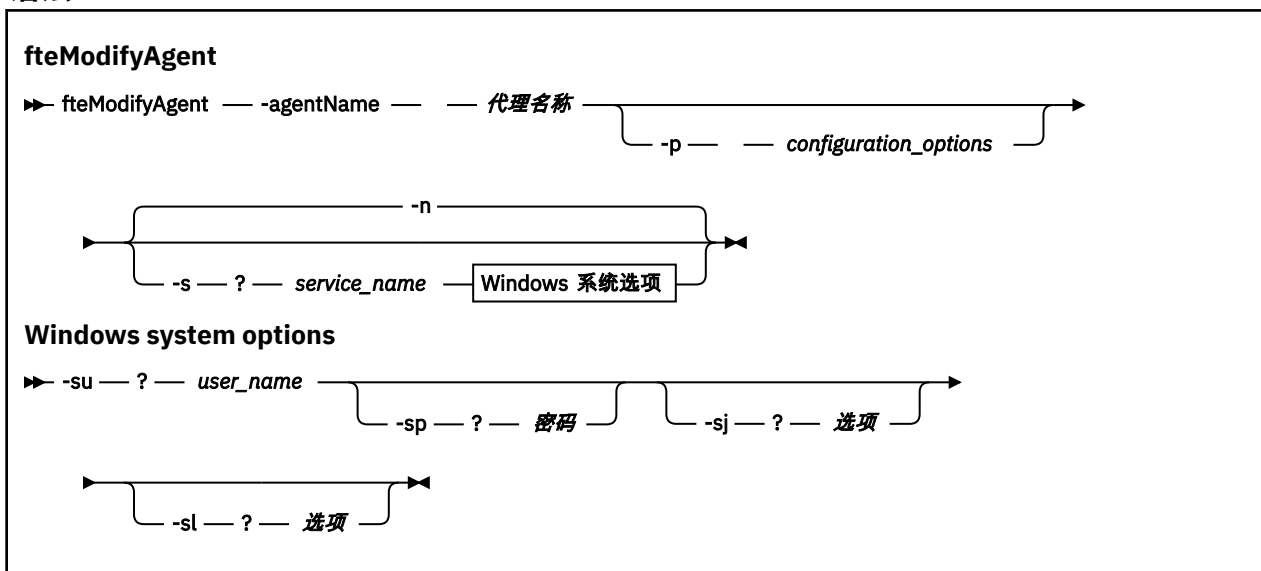
如果要将现有代理及其关联配置从 WebSphere MQ File Transfer Edition 的任何版本迁移至 IBM WebSphere MQ V7.5 或更高版本，请使用 **fteMigrateAgent** 命令进行迁移。该命令可用于迁移标准代理、Connect:Direct 代理、协议网桥代理或 Web 代理。该命令还可用于在单个请求中迁移多个代理。

fteMigrateConfigurationOptions 命令从 WebSphere MQ File Transfer Edition V7.0 迁移一组配置选项，并将其复制到 IBM WebSphere MQ V7.5 或更高版本，前提是这些文件在 V7.5 或更高版本上尚不存在。如果这些文件已存在，那么会输出一条消息，而且命令无法继续。

fteModifyAgent (修改 IBM MQ Managed File Transfer 代理)

fteModifyAgent 命令修改现有代理程序，以便可以将其作为 Windows 服务运行。此命令仅在 Windows 上可用，并且必须由作为 IBM MO 管理员和 mgm 组成员的用户运行。

语法



参数

-agentName *agent_name*
必需。要修改的代理的名称。

-p configuration_options

可选。该参数可确定用于修改代理的配置选项集。按照约定，请使用非缺省协调队列管理器的名称作为该参数的输入。然后，**fteModifyAgent** 命令使用与此非缺省协调队列管理器相关联的属性文件集。

仅在您希望使用不同于缺省值的配置选项时，才指定可选的 **-p** 参数。如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-s service_name

可选。指示代理程序将作为 Windows 服务运行。如果未指定 服务名称，那么服务将命名为 **mqmftAgent<AGENT><QMGR>**，其中 **<AGENT>** 是代理名称，**<QMGR>** 是代理队列管理器名称。

服务的显示名称 (显示在 窗口 服务 窗口的 名称 列中) 始终为 **WebSphere MQ Managed File Transfer 代理程序 <AGENT>@<QMGR>**。

-su user_name

可选。当代理程序作为 Windows 服务运行时，该参数指定运行服务所使用的帐户名。要使用 Windows 域用户帐户来运行代理，请以 **DomainName\UserName** 格式指定该值。要使用本地内置域中的帐户来运行该服务，请以 **UserName** 格式指定该值。

使用 **-su** 参数指定的 Windows 用户帐户必须具有 **Log on as a service** 权限。有关如何授予该权限的信息，请参阅第 397 页的『[有关作为 Windows 服务运行代理或记录器的指南](#)』。

当指定 **-s** 时，该参数为必需参数。等同于 **-serviceUser**。

-sp password

可选。由 **-su** 或 **-serviceUser** 参数设置的用户帐户的密码。

该参数仅在指定 **-s** 时才有效。等同于 **-servicePassword**。如果在指定 **-s** 参数时不指定该参数，将生成警告消息。该消息警告您必须先使用 Windows 服务工具设置密码，这样服务才能成功启动。

-sj options

可选。当代理作为 Windows 服务启动时，此参数以 **-D** 或 **-X** 格式来定义将传递到 Java 虚拟机 (JVM) 的选项列表。系统会使用数字符号 (#) 或分号 (;) 字符分隔这些选项。如果需要嵌入任何 # 或 ; 字符，请使用单引号将其引起来。

 有关 **fteModifyAgent** 命令以何种方式验证 **-serviceJVMOptions** 参数所指定 JVM 选项的更新的更多信息，请参阅[用于更新代理或记录器 JVM 选项的指南](#)。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceJVMOptions**。

-sl options

可选。设置 Windows 服务日志级别。有效选项包括：error、info、warn 和 debug。缺省值为 info。如果 Windows 服务出现问题，那么该选项可能很有用。如果将其设置为 debug，那么会在服务日志文件中提供更详细的信息。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceLogLevel**。

-n

可选。指示代理将作为正常进程运行。这与 **-s** 选项互斥。如果既未指定 **-s** 选项，也未指定 **-n** 选项，那么会将代理程序配置为正常的 Windows 进程。

等同于 **-normal**。

-? or -h

可选。显示命令语法。

示例

在该例中，AGENT1 被修改以作为 Windows 服务运行：

```
fteModifyAgent -agentName AGENT1 -s -su fteuser -sp ftepassword
```

```
fteModifyAgent -agentName AGENT1
```

返回码

命令成功完成。

命令结束但未成功。

第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』

相关任务

您可以将代理作为 Windows 服务启动，以便当您从 Windows 注销后，代理可以继续运行，并且可以接收文件传输。

第 460 页的『fteCreateAgent (创建 IBM MQ Managed File Transfer 代理)』

第 548 页的『fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）』

fteStopLogger 命令停止记录器。

使用 **fteModifyLogger** 命令修改记录器，以便记录器能够作为 Windows 服务运行。只能在 Windows 平台上使用此命令，必须由作为 IBM MQ 管理员和 mam 组成员的用户运行，并且必须首先使用

fteStopLogger 命令停止记录器。

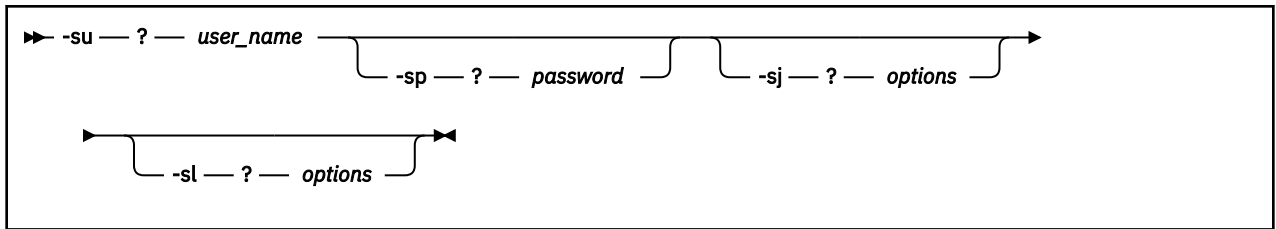
独立记录器 (无论是针对文件还是针对数据库) 在 **服务** 应用程序的 **名称** 列中显示为 “IBM MQ Managed File Transfer 属性集 逻辑名称@*logger qm* 的记录器”。 *logger qm* 的值是记录器的命令队列管理器的名称。

fteModifyLogger

► fteModifyLogger — -loggerName — — (logger_name) — { — -p — 配置_选项 — } —

{ — -n — } — { — -s — ? — service_name — Windows systems options — } ►

Windows system options



参数

-loggerName (logger_name)

必需。要修改的 IBM MQ Managed File Transfer 记录器的名称。

-p configuration_options

可选。该参数可确定用于修改记录器的配置选项集。按照约定，请使用非缺省协调队列管理器的名称作为该参数的输入。然后，**fteModifyLogger** 命令使用与此非缺省协调队列管理器相关联的属性文件集。

仅在您希望使用不同于缺省值的配置选项时，才指定可选的 **-p** 参数。如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-s service_name

可选。指示记录器将作为 Windows 服务运行。如果未指定 服务名称，那么服务将命名为 `mqmftLogger<LOGGER><QMGR>`，其中 `<LOGGER>` 是记录器名称，`<QMGR>` 是记录器队列管理器名称。

服务的显示名称 (显示在 窗口 **服务** 窗口的 **名称** 列中) 始终为 **WebSphere MQ Managed File Transfer 记录器 <LOGGER>@<QMGR>**。

-su 或 -serviceUser user_name

当指定 **-s** 时必需。指定运行 Windows 服务所使用的帐户名。要使用 Windows 域用户帐户运行代理，请以 `DomainName\UserName` 形式指定该值。要使用本地内置域中的帐户来运行该服务，请以 `UserName` 格式指定该值。

您使用 **-su** 参数指定的 Windows 用户帐户必须具有作为服务登录的许可权。要了解如何授予此许可权的信息，请参阅第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』。

-sp 或 -servicePassword password

可选。仅当指定 **-s** 时才有效。由 **-su** 或 **-serviceUser** 参数设置的用户帐户的密码。

如果在指定 **-s** 参数时未指定此参数，那么将警告您必须先使用 Windows 服务工具设置密码，然后才能成功启动服务。

-sj 或 -serviceJVMOptions options

可选。仅当指定 **-s** 时才有效。当记录器作为 Windows 服务启动时，以 **-D** 或 **-X** 格式来定义将传递至 JVM 的选项列表。系统会使用数字符号 (#) 或分号 (;) 字符分隔这些选项。如果需要嵌入任何 # 或 ; 字符，请使用单引号 (') 将其括起。

V 8.0.0.9 有关 **fteModifyLogger** 命令处理通过 **-serviceJVMOptions** 参数指定的 JVM 选项更新验证的方式的更多信息，请参阅 [有关更新代理程序或记录器 JVM 选项的指南](#)。

-sl 或 -serviceLogLevel options

可选。仅当指定 **-s** 时才有效。设置 Windows 服务日志级别。有效选项是: `error`、`info`、`warn`、`debug`。缺省选项为 `info`。如果 Windows 服务出现问题，那么该选项可能很有用。如果将其设置为 `debug`，那么会在服务日志文件中提供更详细的信息。

-n 或 -normal

可选。指示记录器将作为正常进程运行。这与 **-s** 选项互斥。如果既未指定 **-s** 选项，也未指定 **-n** 选项，那么会将记录器配置为正常的 Windows 进程。

-? or -h

可选。显示命令语法。

示例

必须先使用 **fteStopLogger** 命令来停止记录器，然后才能运行 **fteModifyLogger** 命令。

在该例中，先前已创建了名为 **logger1** 的记录器。该命令显示如何更改记录器以作为 Windows 服务运行：

```
fteModifyLogger -loggerName logger1 -s -su fteuser -sp ftepassword
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』

您可以作为 Windows 服务来运行 IBM MQ Managed File Transfer 代理、独立数据库记录器和独立文件记录器。如果这些 Windows 服务出现任何问题，您可以使用服务日志文件以及本主题中的信息来诊断问题。

相关任务

第 216 页的『将代理作为 Windows 服务启动』

您可以将代理作为 Windows 服务启动，以便当您从 Windows 注销后，代理可以继续运行，并且可以接收文件传输。

相关参考

第 575 页的『fteStartLogger（启动记录器）』

fteStartLogger 命令启动 IBM MQ Managed File Transfer 日志记录应用程序。

第 579 页的『fteStopLogger（停止记录器）』

fteStopLogger 命令停止记录器。

fte0bfuscate（加密敏感数据）

fte0bfuscate 命令用于加密凭证文件中的敏感数据。这样，获取了对凭证文件的访问权的人员就无法读取该文件的内容。

用途

凭证文件中的用户名和密码属性可以进行模糊处理。这些属性将变换为带有 Cipher 后缀的新相关属性。例如：

```
<!--
MQMFTCredentials properties before
-->
<tns:logger name="logger1" user="user1" password="passw0rd" />
<tns:file path="$HOME/trust.jks" password="passw0rd" />

<!--
MQMFTCredentials properties after
-->
<tns:logger name="logger1" userCipher="e71vKCg2pf" passwordCipher="se71vKCg" />
<tns:file path="$HOME/trust.jks" passwordCipher="e71vKCg2pf" />

<!--
ProtocolBridgeCredentials Properties before
-->
<tns:user name="Fred" serverUserId="fred" serverPassword="passw0rd" />

<!--
ProtocolBridgeCredentials properties after
-->
<tns:user name="Fred" serverUserIdCipher="e51vVCg2pf" serverPasswordCipher="se51vBCg" />

<!--
ConnectDirectCredentials properties before
-->
```

```

<tns:user name="fteuser" ignorecase="true" pattern="wildcard"
    cdUserId="cdUser" cdPassword="cdPassword" pnodeUserId="pnodeUser"
    pnodePassword="pnodePassword">
  <tns:snode name="snode1" pattern="wildcard" userId="snodeUser" password="snodePassword"/>
</tns:user>

<!--
  ConnectDirectCredentials properties after
-->
<tns:user name="fteuser" ignorecase="true" pattern="wildcard"
    cdUserIdCipher="e71vKCg2pf" cdPasswordCipher="se71vKCg"
    pnodeUserIdCipher="2f1vgCg6df" pnodePasswordCipher="e71vKCg2pf">
  <tns:snode name="snode1" pattern="wildcard" userIdCipher="e51vVCg2pf" passwordCipher="se51vBCg"/>
</tns:user>

```

语法

fteObfuscate

➤ **fteObfuscate** — **-credentialsFile** — *credentials_file_name* ➤

参数

-credentialsFile

必需。内容将进行模糊处理的凭证文件的名称。

-? or -h

可选。显示命令语法。

示例

在此示例中，MQMFTCredentials.xml 的内容已进行模糊处理。

```
fteObfuscate -credentialsFile /home/fteuser/MQMFTCredentials.xml
```

返回码

0

命令成功完成。

1

命令结束但未成功。

ftePingAgent (检查 IBM MQ Managed File Transfer 代理是否处于活动状态)

ftePingAgent 命令对 IBM MQ Managed File Transfer 代理程序执行 ping 操作，以确定该代理程序是否可访问，如果可访问，那么它是否能够响应简单查询。

用途

使用 **ftePingAgent** 命令来检查 Managed File Transfer 代理程序是否可访问，如果可访问，那么检查它是否能够按照 are you there? 的行来响应简单查询。此命令的输出示例如下：

```

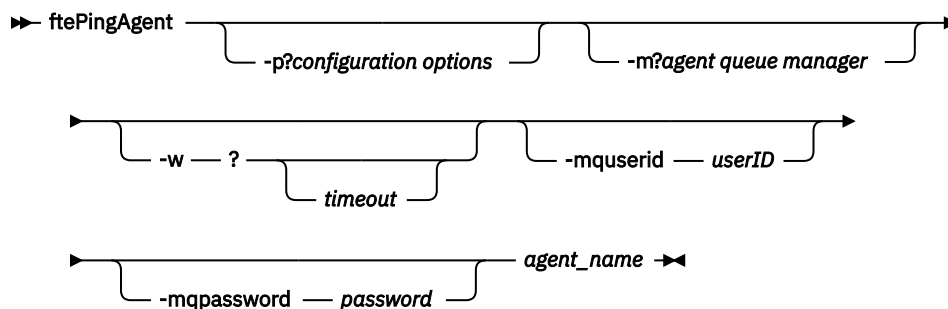
C:\> ftePingAgent AGENT86
5724-H72 Copyright IBM Corp. 2008 , 2023. ALL RIGHTS RESERVED
BFGPR0127W: No credentials file has been specified to connect to IBM MQ. Therefore, the
assumption is that IBM MQ authentication has been disabled.
BFGCL0212I: Issuing ping request to agent AGENT86
BFGCL0213I: agent AGENT86 responded to ping in 0.094 seconds.

```

仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。请参阅[配置选项](#)，以获取更多信息。

语法

ftePingAgent



参数

-p (*configuration options*)

可选。该参数确定用于发出 ping 代理请求的配置选项集。使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。请参阅[配置选项](#)，以获取更多信息。

-m (*queue manager*)

可选。您要 ping 的代理连接到的队列管理器的名称。如果您未指定 -m 参数，那么根据使用的配置选项集确定使用的队列管理器。

-w (*timeout*)

可选。指定命令应最多等待 *timeout* 秒供代理进行响应。如果您未指定超时，或指定超时值为 -1，那么命令会无限期等待，直至代理响应。如果您未指定该选项，那么缺省情况下最多等待五秒供代理响应。

如果指定了 *timeout*，那么 **ftePingAgent** 命令消息将在两倍的 *timeout* 值后超时，而不是转至指定的死信队列。如果命令已设置为无限期等待，那么命令消息将不会超时。

-mquserid (*user ID*)

可选。指定要向命令队列管理器进行认证的用户标识。

-mqpassword (*password*)

可选。指定要向命令队列管理器进行认证的密码。此外，还必须指定 -mquserid 参数。如果指定了 -mquserid，但未指定 -mqpassword，将会提示您提供相关联的密码。将不会显示密码。

(*agent name*)

必需。您要 ping 的 IBM MQ Managed File Transfer 代理的名称。

-? or -h

可选。显示命令语法。

示例

在该示例中，命令 ping 与 QM_MERCURY 连接的代理 AGENT1。命令在返回之前最多等待 40 秒供 AGENT1 响应。

```
ftePingAgent -m QM_MERCURY -w 40 AGENT1
```

返回码

0

命令成功完成。代理处于活动状态并能够处理传输。

1

命令结束但未成功。命令无法将消息发送给代理。

2

命令由于超时而结束。命令向代理发送消息，但代理在给定时间内未响应。

相关参考

第 532 页的『[fteListAgents](#)（列出协调队列管理器的 IBM MQ Managed File Transfer 代理）』

通过命令行，使用 **fteListAgents** 命令可列出向特定协调队列管理器注册的所有 IBM MQ Managed File Transfer 代理。

第 564 页的『[fteShowAgentDetails](#)（显示 IBM MQ Managed File Transfer 代理详细信息）』

使用 **fteShowAgentDetails** 命令可显示特定 IBM MQ Managed File Transfer 代理的详细信息。这些是由其 IBM MQ Managed File Transfer 协调队列管理器存储的详细信息。

第 387 页的『[在您认为传输卡住时应该怎么办](#)』

在高负荷的系统上或者在源和目标代理之间发生网络问题时，传输可能会偶尔显示为卡住，进入排队或恢复状态。导致此情况的因素有很多。

fteBatch、fteCommon 和 ftePlatform 脚本

fteBatch，fteCommon 和 ftePlatform 是 IBM MQ Managed File Transfer 在 `MQ_INSTALLATION_PATH/bin` 目录中作为辅助脚本提供的脚本。并非每一个平台上都有这些脚本。

fteBatch 脚本（仅 z/OS）

fteBatch 是用于从 JZOS Batch Launcher 运行 IBM MQ Managed File Transfer 的帮助脚本。fteBatch 仅安装在 z/OS 上。通常，IBM MQ Managed File Transfer 是使用提供的命令 shell 脚本启动的，这些脚本在启动适合于该函数的 Java 类之前执行一些环境配置。使用 JZOS Batch Launcher 启动 IBM MQ Managed File Transfer 时，将直接从 Launcher 启动 Java 类。可以在启动程序设置过程中调用 fteBatch，以将所需类名放入环境变量中，并执行正常命令 shell 脚本在启动 Java 之前执行的设置工作。这可在您的作业与 WebSphere MQ File Transfer Edition 使用的内部类名之间提供一定级别的隔离。

由于您可通过命令的新 PDSE 数据集来运行 IBM MQ Managed File Transfer，因此将不再使用 IBM MQ Managed File Transfer V8.0 的 fteBatch 命令。有关更多信息，请参阅第 118 页的『[创建 IBM MQ Managed File Transfer 代理或记录器命令数据集](#)』。

fteCommon

fteCommon 是由其他 IBM MQ Managed File Transfer 命令脚本启动的辅助脚本，用于在启动 Java 之前执行公共设置处理。

ftePlatform

ftePlatform 是由 fteCommon 脚本启动的帮助脚本，用于执行特定于平台的设置过程。

fteRAS（收集 MFT 故障诊断信息）

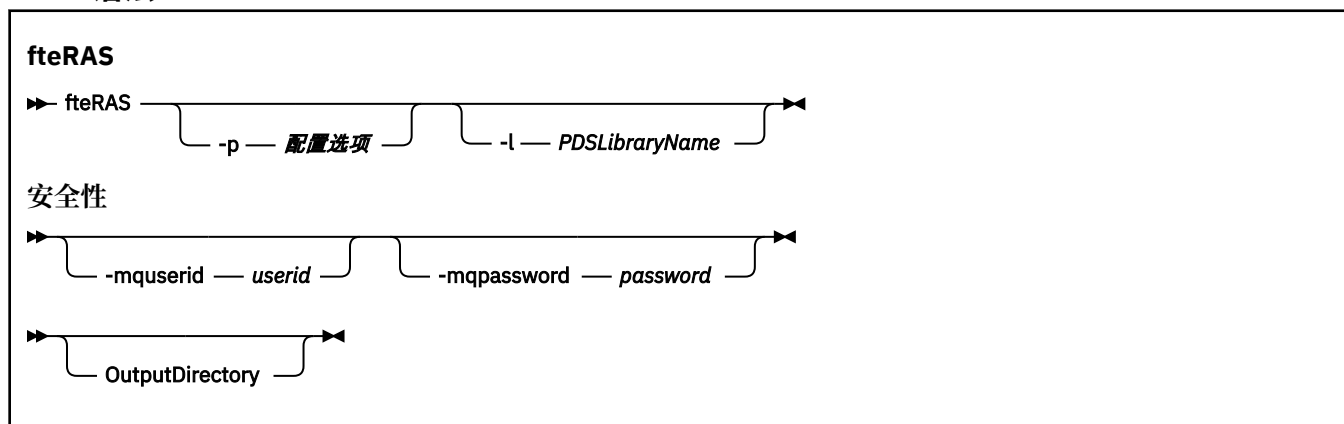
fteRAS 命令收集 Managed File Transfer 的故障诊断信息 (MustGather 数据)。**fteRAS** 收集的信息特定于运行程序的系统上的 Managed File Transfer 安装。

用途

如果需要收集故障诊断信息以用于在 Managed File Transfer 代理程序，数据库记录器或其他命令报告问题或无法正常工作时帮助查找解决方案，请使用 **fteRAS** 命令来运行可靠性、可用性和可维护性信息 (RAS) 收集工具。

运行 **fteRAS** 命令时，用于放置所生成的归档 (.zip) 文件的输出目录可以是缺省位置或您选择的目录。

语法



参数

-p configuration_options

可选。确定用于收集故障诊断信息 (例如, 代理程序列表) 的配置选项集。将配置选项集的名称用作 **-p** 参数的值。按照约定, 此名称是协调队列管理器的名称。如果未指定该参数, 那么会使用缺省的配置选项集。

-l

可选。仅适用于 z/OS。指定 PDS 库的名称, 此库包含用于对特定代理或记录器调用 MQMFT 命令的 JCL 脚本。从命令 PDS 库的 BFGRAS JCL 脚本中运行此命令时始终会设置此选项, 以便在输出目录中捕获 PDS 库的所有成员。

-mquserid user id

可选。指定要向命令队列管理器进行认证的用户标识。

-mqpassword password

可选。指定要向命令队列管理器进行认证的密码。此外, 还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**, 但未同时指定 **-mqpassword**, 那么系统会提示您提供相关的密码。在屏幕上不会显示此密码。

OutputDirectory

可选。要在收集 RAS 数据时使用的目录, 以及在成功收集数据后存储输出文件 (例如 fteRAS.zip) 的位置。如果此目录不存在, 那么将创建此目录。缺省位置为 mqft 日志目录。

-你说什么 或 -h

可选。显示命令语法。

示例

在 UNIX 和 Linux 上, 要将输出文件 fteRAS.zip 存储在 /var/mqm/errors 目录中, 请运行 **fteRAS**, 如以下示例中所示:

```
fteRAS /var/mqm/errors
```

以下消息确认已成功完成该命令:

```
BFGCL0604I: fteRAS command completed successfully. Output is stored in /var/mqm/errors/fteRAS.zip
```

在 Windows 上, 要将输出文件 fteRAS.zip 存储在新安装的 IBM MQ 的缺省错误目录中, 请运行 **fteRAS**, 如以下示例中所示:

```
fteRAS "C:\ProgramData\IBM\MQ\errors"
```

以下消息确认已成功完成该命令:

```
BFGCL0604I: fteRAS command completed successfully. Output is stored in  
C:\ProgramData\IBM\MQ\errors\fteRAS.zip
```


注: 对于 IBM MQ 8.0 或更高版本, 如果这不是该产品版本的新安装, 那么您系统上的错误目录位置可能有所不同。有关更多信息, 请参阅 [Windows 上的程序和数据目录位置](#)。

在 IBM i 上, 要将输出文件复制到 /QIBM/UserData/mqm/errors, 请从 Qshell 运行 **fteRAS** 命令, 如下示例中所示:

```
/QIBM/ProdData/mqm/bin/fteRAS /QIBM/UserData/mqm/errors
```

以下消息确认已成功完成该命令:

```
BFGCL0604I: fteRAS command completed successfully. Output is stored in /QIBM/UserData/mqm/errors/fteRAS.zip
```

相关参考

第 371 页的『[IBM MQ Managed File Transfer 故障诊断](#)』

使用以下参考信息可帮助您诊断 IBM MQ Managed File Transfer 中的错误:

fteSetAgentTraceLevel (设置 IBM MQ Managed File Transfer 代理跟踪级别)

使用 **fteSetAgentTraceLevel** 命令可动态修改代理的当前跟踪级别。

用途

使用该命令可以开启和关闭代理跟踪, 或更改所设置的代理跟踪级别。当使用 **fteSetAgentTraceLevel** 命令时, 不必关闭和重新启动代理就可修改跟踪级别。生成的跟踪文件位于 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%/trace%PID%.txt` 中, 其中 `%PID%` 是代理实例的进程标识。



注意: 在分布式平台上使用 IBM WebSphere MQ 7.5 或更高版本时, 只有作为代理进程运行身份的用户才可以运行 **fteSetAgentTraceLevel** 命令。

V8.0.0.6 对于 z/OS, **fteSetAgentTraceLevel** 命令可以由以下任一人员运行:

- 用于运行代理进程的同一用户标识。
- 代理属性 **adminGroup** 指定的组的成员。

有关更多信息, 请参阅第 591 页的『[agent.properties 文件](#)』中的 **adminGroup** 属性。

在 IBM MQ Managed File Transfer V 7.5 和更高版本中, **fteSetAgentTraceLevel** 命令还会写入代理进程控制器的跟踪。生成的跟踪文件位于 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/pctrace%PID%/pctrace%PID%.txt` 中, 其中 `%PID%` 是代理实例的进程标识。

您还可以使用该命令, 让代理进程生成 Javacore。代理会在以下目录中生成一个 Javacore 文件:
`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name`。

因为运行跟踪可能会严重影响性能, 并且可能会产生大量的跟踪数据, 所以请谨慎运行跟踪并仅在必要时运行。通常情况下, 请仅在您的 IBM 服务代表要求时启用跟踪。



注意:

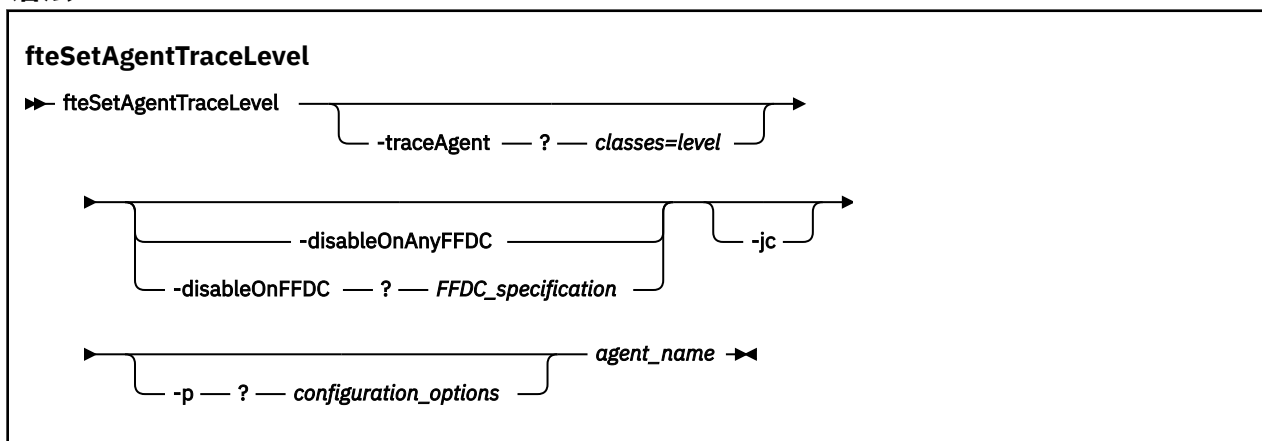
1. 必须在运行代理程序的系统上运行此命令。
2. 跟踪和日志记录不会在代理程序重新启动时持久存储。

如果代理程序终止并由 Process Controller 进程重新启动, 那么在更新 **agent.properties** 文件以包含必需的跟踪和日志属性之前, 动态跟踪和日志不会生效。

您可以在 **agent.properties** 文件中设置更多跟踪属性, 例如, 跟踪文件大小和要保留的跟踪文件数量。[高级代理属性](#)中描述了这些属性。

仅当希望使用不同于缺省集的配置选项集时, 才对该命令指定可选的 **-p** 参数。请参阅第 591 页的『[agent.properties 文件](#)』, 以了解更多信息。

语法



参数

-traceAgent classes=level

必需。将代理跟踪设置到的级别和将该跟踪应用到的类。指定以下格式：

```
classes=level
```

例如：

```
com.ibm.wmqfte=all
```

指定您希望将该跟踪级别应用到的类规范的逗号分隔列表。如果未指定该参数，那么跟踪级别将应用于所有代理类。

您可以使用 MQMFT 程序包名称来代替类，以仅跟踪特定的程序包。但是，由于此选项仅捕获代理的部分行为，所以通常不建议您使用程序包过滤。

如果 (classes) 以加号 (+) 开头，那么将向当前正在跟踪的任何现有跟踪类添加跟随加号的跟踪类列表。

有效的跟踪级别选项如下所述，按照跟踪文件大小的升序和详细信息列出：

关

关闭代理跟踪，但继续将信息写入日志文件。这是缺省选项。

flow

捕获与代理中处理流相关联的跟踪点的数据。

moderate

捕获跟踪中的适量诊断信息。

详细

捕获跟踪中的大量诊断信息。

all

将代理跟踪设置为在所有代理类上运行。

要开始全面跟踪代理，请运行以下命令：

```
fteSetAgentTraceLevel -traceAgent =all AGENT_NAME
```

要停止全面跟踪代理，请运行以下命令：

```
fteSetAgentTraceLevel -traceAgent =off AGENT_NAME
```

-disableOnAnyFFDC

可选。如果指定了该参数，那么当代理生成“首次故障数据捕获”(FFDC) 文件时，跟踪将被禁用。

您只能指定 **-disableOnAnyFFDC** 和 **-disableOnFFDC** 参数中的一个。

-disableOnFFDC *FFDC_specification*

可选。如果指定了该参数，那么当代理生成与 *FFDC_specification* 匹配的“首次故障数据捕获”(FFDC) 文件时，跟踪将被禁用。*FFDC_specification* 是以逗号分隔的值列表。值的格式可以是：

class_name

FFDC 继承的类的名称。例如，`com.ibm.wmqfte.classA`。

class_name:probe_ID

FFDC 继承的类中的类的名称和位置的探测器标识。例如，`com.ibm.wmqfte.classB:1`。

您只能指定 **-disableOnAnyFFDC** 和 **-disableOnFFDC** 参数中的一个。

-jc

可选。请求代理来生成 Javacore 文件。IBM 服务团队可能会要求您使用该参数来运行命令，以帮助诊断问题。此参数不能与除 **-p** 以外的任何其他参数一起使用。

-p *configuration_options*

可选。该参数确定用于设置代理跟踪级别的配置选项集。使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

agent_name

必需。您要为其设置跟踪级别的 IBM MQ Managed File Transfer 代理的名称。

-? or -h

可选。显示命令语法。

示例

在本示例中，针对 AGENT1 的所有类，跟踪级别都设置为 all：

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte=all AGENT1
```

在本示例中，针对 AGENT1 的 `com.ibm.wmqfte.agent.Agent` 和 `com.ibm.wmqfte.cmdhandler` 类，跟踪级别设置为 all：

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.agent.Agent,com.ibm.wmqfte.cmdhandler=moderate AGENT1
```

在本示例中，由于 **-traceLevel** 参数设置为 off，因此排除了对子类的跟踪。以 `com.ibm.outer` 开头的类（除以 `com.ibm.outer.inner` 开头的类外）都按详细级别进行跟踪：

```
fteSetAgentTraceLevel -traceAgent com.ibm.outer=verbose AGENT1  
fteSetAgentTraceLevel -traceAgent +com.ibm.outer.inner=off AGENT1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

fteSetLoggerTraceLevel

使用 **fteSetLoggerTraceLevel** 命令可动态修改 IBM MQ Managed File Transfer 记录器的当前跟踪级别。

用途

使用该命令来将记录器跟踪打开和关闭，或者更改已设置的记录器跟踪级别。当使用 **fteSetLoggerTraceLevel** 命令时，不必关闭并重新启动记录器即可修改跟踪级别。生成的跟踪文件位于 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%/trace%PID%.txt` 中，其中 `%PID%` 是记录器实例的进程标识。

在 IBM MQ Managed File Transfer V 7.5 和更高版本中，**fteSetLoggerTraceLevel** 命令还会写入记录器进程控制器的跟踪。生成的跟踪文件位于 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/pctrace%PID%/pctrace%PID%.txt` 中，其中 `%PID%` 是记录器实例的进程标识。

该命令还可用于促使记录器进程生成 Javacore。记录器会在以下目录中生成一个 Javacore 文件：
`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name`。

因为运行跟踪可能会严重影响性能，并且可能会产生大量的跟踪数据，所以请谨慎运行跟踪并仅在必要时运行。通常情况下，请仅在您的 IBM 服务代表要求时启用跟踪。



注意：

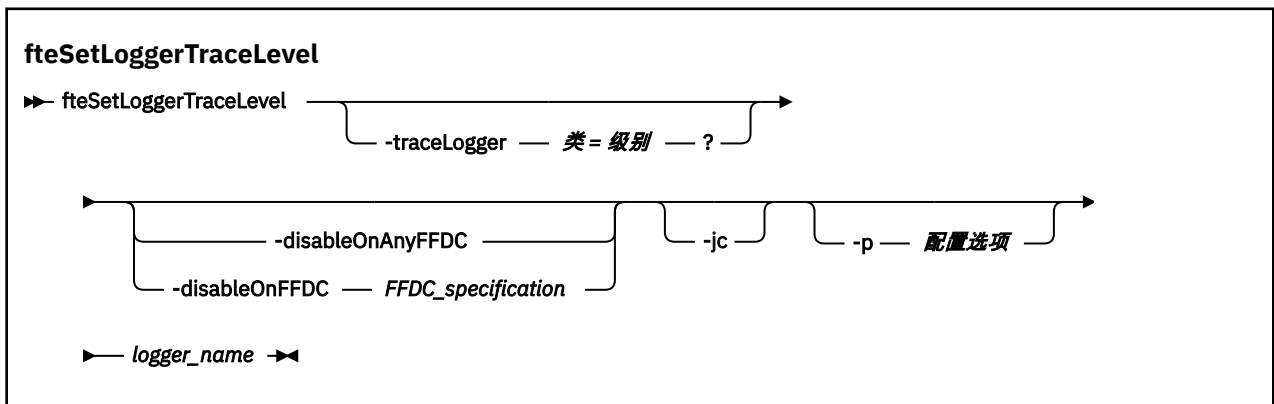
1. 必须在运行代理程序的系统上运行此命令。
2. 跟踪和日志记录不会在代理程序重新启动时持久存储。

如果代理程序终止并由 Process Controller 进程重新启动，那么在更新 `agent.properties` 文件以包含必需的跟踪和日志属性之前，动态跟踪和日志不会生效。

您可以在 `logger.properties` 文件中设置更多跟踪属性，例如跟踪文件大小和要保留的跟踪文件数量。这些属性在[记录器属性](#)中进行了描述。

仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 `-p` 参数。请参阅 [第 163 页的『IBM MQ Managed File Transfer 的记录器配置属性』](#)，以获取更多信息。

语法



参数

-traceLogger classes=level

必需。将记录器跟踪设置到的级别和将该跟踪应用于的类。指定以下格式：

```
classes=level
```

例如：

```
com.ibm.wmqfte=all
```

指定您希望将该跟踪级别应用到的类规范的逗号分隔列表。如果不指定该参数，那么该跟踪级别将应用于所有记录器类。

如果 (*classes*) 以加号 (+) 开头, 那么将向当前正在跟踪的任何现有跟踪类添加跟随加号的跟踪类列表。

有效的跟踪级别选项如下所述, 按照跟踪文件大小的升序和详细信息列出:

关

关闭记录器跟踪, 但继续将信息写入日志文件。这是缺省选项。

flow

捕获与记录器中处理流关联的跟踪点的数据。

moderate

捕获跟踪中的适量诊断信息。

详细

捕获跟踪中的大量诊断信息。

all

将记录器跟踪设置为对所有记录器类运行。

-disableOnAnyFFDC

可选。如果指定了该参数, 那么当记录器生成“首次故障数据捕获”(FFDC) 文件时, 会在记录器上禁用跟踪。

您只可以指定 **-disableOnAnyFFDC** 和 **-disableOnFFDC** 参数中的一个。

-disableOnFFDC *FFDC_specification*

可选。如果指定了该参数, 那么当记录器生成与 *FFDC_specification* 匹配的“首次故障数据捕获”(FFDC) 文件时, 会在记录器上禁用跟踪。 *FFDC_specification* 是以逗号分隔的值列表。值可以是以下某种格式:

class_name

FFDC 继承的类的名称。例如, `com.ibm.wmqfte.classA`。

class_name:probe_ID

FFDC 继承的类中的类的名称和位置的探测器标识。例如, `com.ibm.wmqfte.classB:1`。

您只可以指定 **-disableOnAnyFFDC** 和 **-disableOnFFDC** 参数中的一个。

-jc

可选。请求记录器生成 Javacore 文件。IBM 服务团队可能会要求您使用该参数来运行此命令, 以帮助诊断问题。不能将 **-jc** 参数与任何其他参数一起使用。

-p 配置选项

可选。该参数确定用于设置记录器跟踪级别的配置选项集。使用非缺省协调队列管理器的名称作为该参数的输入。然后, 命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数, 那么将使用基于缺省协调队列管理器的配置选项集。

logger_name

必需。您要为其设置跟踪级别的 IBM MQ Managed File Transfer 记录器的名称。

-? or -h

可选。显示命令语法。

示例

在本示例中, 对于 `LOGGER1` 的所有类, 跟踪级别都设置为 `all`:

```
fteSetLoggerTraceLevel -traceLogger com.ibm.wmqfte=all LOGGER1
```

在本示例中, 对于 `LOGGER1` 的 `com.ibm.wmqfte.logger.logger` 和 `com.ibm.wmqfte.cmdhandler` 类, 跟踪级别都设置为 `all`:

```
fteSetLoggerTraceLevel -traceLogger com.ibm.wmqfte.logger.logger,com.ibm.wmqfte.cmdhandler=moderate  
LOGGER1
```

在本示例中，由于 **-traceLevel** 参数设置为 **off**，因此排除了对子类的跟踪。以 **com.ibm.outer** 开头的类（除以 **com.ibm.outer.inner** 开头的类外）都按详细级别进行跟踪：

```
fteSetLoggerTraceLevel -traceLogger com.ibm.outer=verbose LOGGER1
fteSetLoggerTraceLevel -traceLogger +com.ibm.outer.inner=off LOGGER1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

fteSetupCommands (创建 command.properties 文件)

fteSetupCommands 命令可创建 `command.properties` 文件。在您发出命令时，该属性文件指定与 IBM MO 网络连接的队列管理器的详细信息。

要点: 在分布式系统上，只有 IBM MQ 管理员（和 mqm 组的成员）身份的用户才可以运行此命令。如果尝试以非 IBM MQ 管理员用户身份运行此命令，那么您将收到错误消息 BFGCL0502E: You are not authorized to perform the requested operation. 并且不会运行该命令。

在 z/OS 系统上，用户必须满足以下至少一个条件才能运行迁移命令：

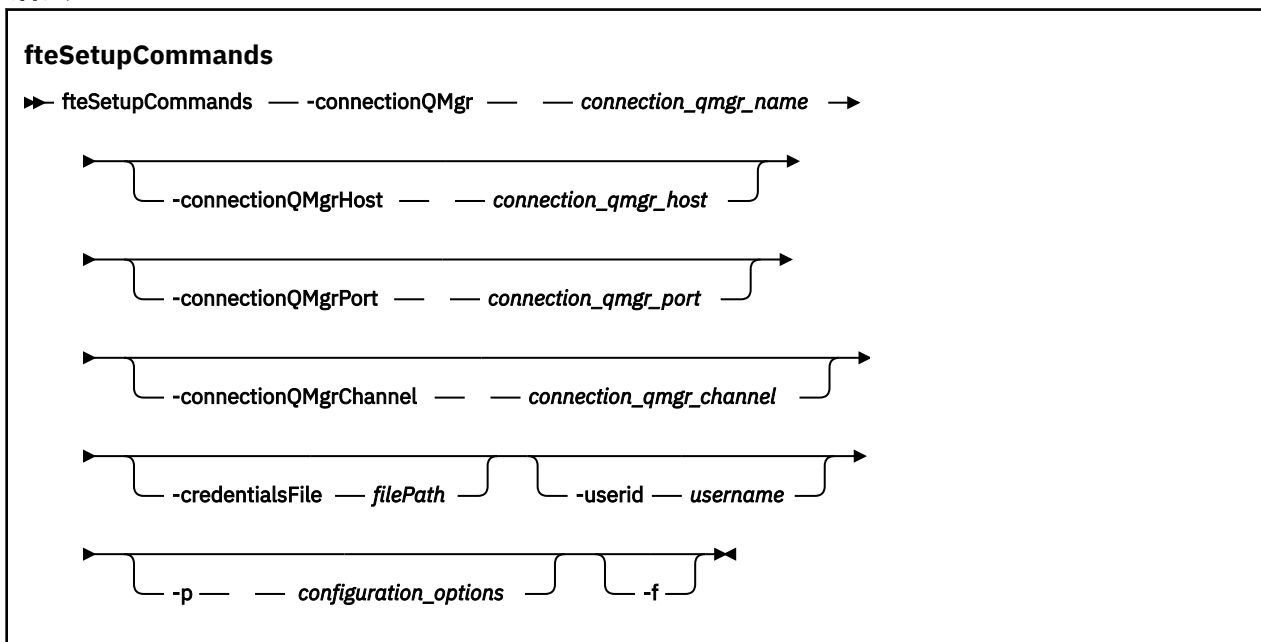
- 是 mqm 组的成员（如果在系统上定义了 mqm 组）。
- **V8.0.0.6** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V8.0.0.6** 在运行命令时，在 BFG_GROUP_NAME 环境变量中未设置任何值。

用途

使用 **fteSetupCommands** 命令可在协调队列管理器配置目录中创建 `command.properties` 文件。此命令使用 `MQ_DATA_PATH` 环境变量和 `installation.properties` 文件确定在何处查找 `command.properties` 文件。在发出 **fteSetupCommands** 命令之前，请确保您已创建并配置协调队列管理器。

要了解有关属性文件的更多信息，请参阅第 588 页的『command.properties 文件』。

语法



参数

-connectionQMgr (连接队列管理器的名称)

必需。用于连接到 IBM MQ 网络以发出命令的队列管理器的名称。

-connectionQMgrHost (连接队列管理器的主机)

可选。连接队列管理器的主机名或 IP 地址。

如果未指定 **-connectionQMgrHost** 参数，那么将采用绑定方式连接。因此，如果要使用客户机方式连接，该参数是必需的。

如果为 **-connectionQMgrHost** 参数指定了值，但没有为 **-connectionQMgrPort** 和 **-connectionQMgrChannel** 属性指定值，那么缺省情况下将使用端口号 1414 和通道 SYSTEM.DEF.SVRCONN。

-connectionQMgrPort (连接队列管理器的端口号)

可选。用于以客户机方式连接到连接队列管理器的端口号。如果指定 **-connectionQMgrPort** 参数，那么还必须指定 **-connectionQMgrHost** 参数。

-connectionQMgrChannel (连接队列管理器的通道)

可选。用于连接到连接队列管理器的通道名称。如果指定 **-connectionQMgrChannel** 参数，那么还必须指定 **-connectionQMgrHost** 参数。

-p (配置选项)

可选。该参数确定用于设置命令队列管理器的配置选项集。使用非缺省协调队列管理器的名称作为该参数的输入。然后，**fteSetupCommands** 命令会使用与该非缺省协调队列管理器关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-credentialsFile (*filePath*)

可选。现有或新凭证文件的完整文件路径，可以向该路径添加 IBM MQ 认证详细信息。

此命令支持向指定的 IBM MQ Managed File Transfer 凭证文件添加一组 IBM MQ 认证详细信息。启用 IBM MQ 连接认证之后，请使用此命令。如果要更新现有的详细信息，则必须使用 **-f** 强制参数。

-userid (*username*)

可选。用于关联凭证详细信息的用户标识。如果未指定用户标识，将向所有用户应用凭证详细信息。此外，还必须指定 **-credentialsFile** 参数。

-f

可选。强制使用该命令中指定的详细信息覆盖现有的 `command.properties` 文件。

-? or -h

可选。显示命令语法。

示例

```
fteSetupCommands -connectionQMgr QM_NEPTUNE -connectionQMgrHost 9.146.157.241  
-connectionQMgrPort 1414 -connectionQMgrChannel SYSTEM.DEF.SVRCONN
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关参考

第 588 页的『`command.properties` 文件』

command.properties 文件指定发出命令时要连接的命令队列管理器以及 IBM MQ Managed File Transfer 联系该队列管理器所需的信息。

第 562 页的『fteSetupCoordination（设置协调详细信息）』

fteSetupCoordination 命令为 IBM MQ Managed File Transfer 创建属性文件和协调队列管理器目录。

fteSetupCoordination（设置协调详细信息）

fteSetupCoordination 命令为 IBM MQ Managed File Transfer 创建属性文件和协调队列管理器目录。

要点: 在分布式系统上，只有 IBM MQ 管理员（和 mqm 组的成员）身份的用户才可以运行此命令。如果尝试以非 IBM MQ 管理员用户身份运行此命令，那么您将收到错误消息 BFGCL0502E: You are not authorized to perform the requested operation. 并且不会运行该命令。

在 z/OS 系统上，用户必须满足以下至少一个条件才能运行迁移命令：

- 是 mqm 组的成员（如果在系统上定义了 mqm 组）。
- **V8.0.0.6** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V8.0.0.6** 在运行命令时，在 BFG_GROUP_NAME 环境变量中未设置任何值。

用途

使用 **fteSetupCoordination** 命令可创建以下 IBM MQ Managed File Transfer 对象：

- 协调队列管理器目录
- 数据目录 mqft (如果不存在)
- installation.properties 文件
- coordination.properties 文件

该命令还为您提供以下 MQSC 命令，必须对协调队列管理器运行这些命令才能配置 IBM MQ Managed File Transfer。MQSC 命令创建主题、主题字符串、SYSTEM.FTE 队列和缺省的数据库记录器队列。这些命令还会更新名称列表，并将协调队列管理器的 PSMODE 属性设置为 ENABLED。

如果协调队列管理器位于 z/OS 上，那么在运行这些 MQSC 命令之前，必须确保已存在以下必需对象：

- SYSTEM.BROKER.DEFAULT.STREAM 队列
- SYSTEM.QPUBSUB.QUEUE.NAMELIST 名称列表
- SYSTEM.BROKER.DEFAULT.STREAM 和 SYSTEM.BROKER.ADMIN.STREAM 流

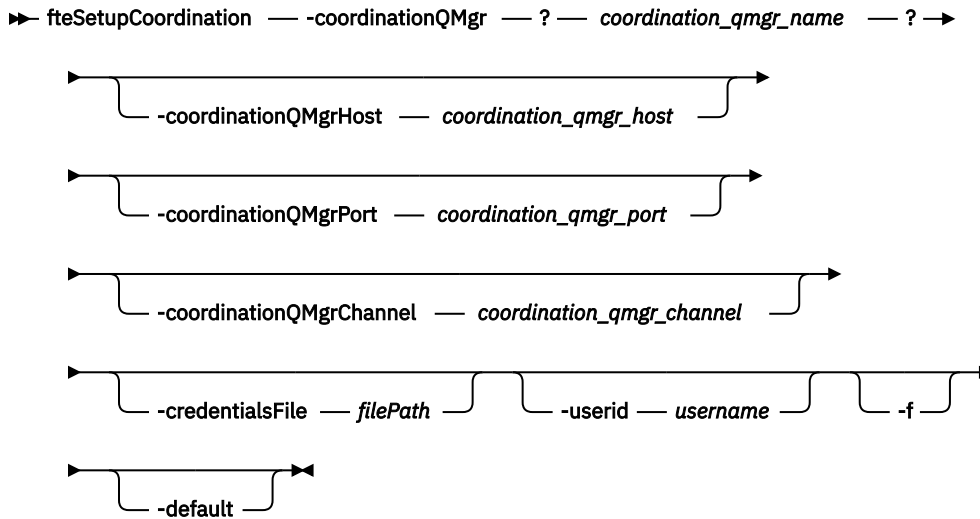
```
DEFINE TOPIC('SYSTEM.FTE') TOPICSTR('SYSTEM.FTE') REPLACE
ALTER TOPIC('SYSTEM.FTE') NPMGDLV(ALLAVAIL) PMGDLV(ALLAVAIL)
DEFINE QLOCAL(SYSTEM.FTE) LIKE(SYSTEM.BROKER.DEFAULT.STREAM) REPLACE
ALTER QLOCAL(SYSTEM.FTE) DESCR('Stream for WMQFTE Pub/Sub interface')
* Altering namelist: SYSTEM.QPUBSUB.QUEUE.NAMELIST
* Value prior to alteration:
DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)
ALTER NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST) +
  NAMES(SYSTEM.BROKER.DEFAULT.STREAM+
    ,SYSTEM.BROKER.ADMIN.STREAM,SYSTEM.FTE)
* Altering PSMODE. Value prior to alteration:
DISPLAY QMGR PSMODE
ALTER QMGR PSMODE(ENABLED)
```

有关这些属性的更多信息，请参阅[配置选项](#)。

如果要使用 z/OS，那么可以使用根据 IBM MQ Managed File Transfer 命令模板 PDSE 库数据集生成的脚本，从 JCL 发出 **fteSetupCoordination** 命令和其他命令。要了解更多信息，请参阅 [第 118 页的『创建 IBM MQ Managed File Transfer 代理或记录器命令数据集』](#)。

语法

fteSetupCoordination



参数

-coordinationQMgr (协调队列管理器名称)

必需。协调队列管理器的名称。该队列管理器必须是 WebSphere MQ V7.0 或更高版本的队列管理器。

-coordinationQMgrHost (协调队列管理器主机)

可选。协调队列管理器的主机名或 IP 地址。

如果未指定 **-coordinationQMgrHost** 参数，那么将采用绑定方式连接。

如果为 **-coordinationQMgrHost** 参数指定了值，但没有为 **-coordinationQMgrPort** 和 **-coordinationQMgrChannel** 参数指定值，那么缺省情况下将使用端口号 1414 和通道 SYSTEM.DEF.SVRCONN。

-coordinationQMgrPort (coordination_qmgr_port)

可选。用于以客户机方式连接协调队列管理器的端口号。如果指定 **-coordinationQMgrPort** 参数，那么还必须指定 **-coordinationQMgrHost** 参数。

-coordinationQMgrChannel (协调队列管理器通道)

可选。用于连接到协调队列管理器的通道名称。如果指定 **-coordinationQMgrChannel** 参数，那么还必须指定 **-coordinationQMgrHost** 参数。

-credentialsFile (filePath)

可选。现有或新凭证文件的完整文件路径，可以向该路径添加 IBM MQ 认证详细信息。

此命令支持向指定的 IBM MQ Managed File Transfer 凭证文件添加一组 IBM MQ 认证详细信息。启用 IBM MQ 连接认证之后，请使用此命令。如果要更新现有的详细信息，则必须使用 **-f** 强制参数。

-userid (username)

可选。用于关联凭证详细信息用户标识。如果未指定用户标识，将向所有用户应用凭证详细信息。此外，还必须指定 **-credentialsFile** 参数。

-f

可选。强制使用该命令中指定的详细信息覆盖现有的协调队列管理器配置。

-default

可选。将缺省配置选项更新为与该命令中指定的协调队列管理器关联的选项。

-? or -h

可选。显示命令语法。

示例

在本示例中，为以客户机方式连接的名为 QM_SATURN 的协调队列管理器设置必需对象：

```
fteSetupCoordination -coordinationQMGr QM_SATURN  
-coordinationQMGrHost myhost.ibm.com -coordinationQMGrPort 1415  
-coordinationQMGrChannel SYSTEM.DEF.SVRCONN
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

第 116 页的『分布式平台上的配置选项』

IBM MQ Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。这些属性文件位于安装产品时所定义的配置目录中。

第 124 页的『MFT 凭证文件』

MFT 凭证文件用于存储用户标识和密码信息。您可以为协调队列管理器、命令队列管理器、每个代理以及每个记录器分别提供一个 MFT 凭证文件。

相关任务

第 144 页的『配置协调队列管理器』

运行 **fteSetupCoordination** 命令之后，运行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目录中的 `coordination_qmgr_name.mqsc` 脚本以对协调队列管理器执行必要配置。但是，如果您希望手动执行此配置，请在协调队列管理器上完成以下步骤。

相关参考

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

fteShowAgentDetails（显示 IBM MQ Managed File Transfer 代理详细信息）

使用 **fteShowAgentDetails** 命令可显示特定 IBM MQ Managed File Transfer 代理的详细信息。这些是由其 IBM MQ Managed File Transfer 协调队列管理器存储的详细信息。

用途

您可以从任何可连接到协调队列管理器的系统运行 **fteShowAgentDetails** 命令。该命令使用 `coordination.properties` 文件连接到协调队列管理器。

该命令生成的代理状态信息是通过代理发布到 `SYSTEM.FTE` 主题的状态消息生成的。第 647 页的『代理状态消息格式』中描述了这些消息。由 **fteShowAgentDetails** 命令生成的状态信息提供发布最后一条状态消息时的代理状态。这些状态消息的频率取决于 `agentStatusPublishRateLimit` 属性的值。有关此属性的更多详细信息，请参阅第 591 页的『agent.properties 文件』。

对于 IBM WebSphere MQ 7.5.0 修订包 1 或更高版本，如果您要查看关于本地代理的诊断信息，请在该命令中指定可选 **-d** 参数。此信息包含当前传输、调度的传输、监视器和代理队列深度。您可以使用此信息来确定本地代理的运行状况和状态。

V 8.0.0.6 对于 z/OS IBM MQ 8.0.0 修订包 6，仅当运行 **fteShowAgentDetails** 命令时，才能指定 **-d** 参数：

- 用于运行代理进程的同一用户标识。
- 代理属性 **adminGroup** 指定的组的成员。

有关更多信息，请参阅第 591 页的『agent.properties 文件』中的 **adminGroup** 属性。

对于 IBM WebSphere MQ 7.5 或更高版本，如果在代理所在的系统上运行该命令，那么将会提供代理进程控制器和队列管理器的状态。您可以使用此信息来帮助确定问题。另外，当您在代理所处的同一个系统上运行该命令时，会为代理意外终止情况提供更多详细的代理状态信息。

要获取可能的代理状态值及其含义的列表，请参阅第 700 页的『代理状态值』。

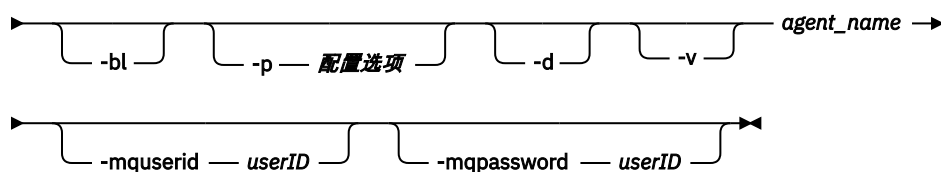
要获取代理进程控制器的可能状态值及其含义的列表，请参阅第 701 页的『代理进程控制器状态值』。

要获取代理跟踪值和 FFDC 规范及其含义的列表，请参阅第 374 页的『fteSetAgentTraceLevel（设置 IBM MQ Managed File Transfer 代理跟踪级别）』。

语法

fteShowAgentDetails

►► fteShowAgentDetails ►►



参数

-bl

可选。额外输出代理的产品构建级别。

-p（配置选项）

可选。该参数确定用于发出请求以显示代理详细信息的配置选项集。使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-d

可选。指定针对 *agent_name* 收集诊断信息。

诊断信息将输出到控制台，并写入到目录

MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\agents\agent_name\logs 内名为 diagnostics.<yyyyMMdd>.<HHmmss>.<ssss>.<number>.properties 的文件中。将创建最多五个包含代理诊断信息的历史文件。如果在运行指定有 **-d** 参数的 **fteShowAgentDetails** 命令时已为代理创建五个历史文件，那么将删除最早的历史文件，并将其替换为包含代理最新诊断信息的新文件。

您只能在代理正在运行时和在本地系统上使用此参数。

-v

可选。指定详细方式，该方式将为代理生成更多输出。其中包括主机名、产品版本、产品构建级别、跟踪级别和首次失败数据捕获 (FFDC) 规范以及每次当前源和目标传输的传输状态的列表。

从代理状态发布中获取当前传输信息，这在第 647 页的『代理状态消息格式』中进行了描述。因此，该传输信息仅在 *agentStatusPublishRateLimit* 属性的值内是准确的。有关此属性的更多详细信息，请参阅第 591 页的『agent.properties 文件』。

agent_name

必需。要显示的 IBM MQ Managed File Transfer 代理的名称。

-mquserid (userID)

可选。指定通过协调队列管理器进行认证时所使用的用户标识。

-mpassword (password)

可选。指定通过协调队列管理器进行认证时所使用的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mpassword**，将会提示您提供相关联的密码。将不会显示密码。

-? or -h

可选。显示命令语法。

示例

在以下示例中，将运行绑定代理，并在本地向该代理发出 **fteShowAgentDetails** 命令：

```
fteShowAgentDetails -v AGENT1
```

```
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Agent Information:
  Name:                AGENT1
  Type:                Standard
  Description:
  Operating System:    Windows Server 2003
  Time Zone:           Greenwich Mean Time
  Product Version:     7.5
  Build Level:         f000-20120312-0957
  Trace Level:         com.ibm.wmqfte.Agent=all
                      com.ibm.wmqfte.common=all
  Trace FFDC:          com.ibm.wmqfte.common:Any
                      com.ibm.wmqfte.Agent:1

Agent Controller Information:
  Controller type:     MQMFT Process Controller
  Status:              STARTED
  Status Details:      The agent process controller has
                      started the agent process.

  Agent Restarts within Interval: 0
  Total Agent Restart Count:      0

Agent Availability Information:
  Status:              READY
  Status Details:      The agent is running and is publishing
                      its status at regular intervals. The
                      last update was received within the
                      expected time period. The agent is
                      ready to process transfers, but none
                      are currently in progress.

Queue Manager Information:
  Name:                QM1
  Transport:           Bindings
  Last Status Reported: AVAILABLE (Last Error MQRC: 0)
  Status Details:      The queue manager is available.

Maximum Number of Running Source Transfers: 25
Maximum Number of Queued Source Transfers: 1000
Source Transfer States:
  No current transfers

Maximum Number of Running Destination Transfers: 25
Destination Transfer States:
  TransferId                State
  414d51204d49414f5720202020202020822c5b4a648c0b20  progress
  414d51204d49414f5720202020202020822c5b4a346c0b20  progress
```

在以下示例中，QMGR1 是用作配置选项输入的非缺省协调队列管理器，诊断信息是通过 **-d** 参数请求的。在具有本地代理程序的 Websphere MQ V7.5.0.1 系统上发出 **fteShowAgentDetails** 命令：

```
fteShowAgentDetails -p QMGR1 -d AGENT1
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Agent Information:
  Name:                AGENT1
  Type:                Standard
  Description:
  Operating System:    Linux
  Time Zone:           Greenwich Mean Time

Agent Controller Information:
  Controller type:     MQMFT Process Controller
```



```

Status: STARTED
Status Details: The agent process controller has started
the agent process.
Agent Restarts within Interval: 0
Total Agent Restart Count: 0

Agent Availability Information:
Status: ACTIVE
Status Details: The agent is running and is publishing
its status at regular intervals. The last
update was received within the expected
time period. The agent is currently
processing one or more transfers.

Queue Manager Information:
Name: QMGR1
Transport: Client
Host: host1.hursley.ibm.com
Port: 2021
Channel: SYSTEM.DEF.SVRCONN
Last Status Reported: UNKNOWN
Status Details: Information about the queue manager is
not available because the agent has a
client connection to the queue manager.

Agent Diagnostic Information:

Command Handler Diagnostics:
Last Command Queue Read Time: 2012-07-30T15:23:10.705Z
Pending Command Queue Size: 0
Last Internal Command Type: Resync Request (from sender) -
414d5120514d43414e4445202020202079e20f5064230010
Last Internal Command Time: 2012-07-30T14:17:10.506Z
Last External Command Type: New Monitor Request
Last External Command Time: 2012-07-30T14:10:57.751Z
Diagnostic Properties File name: C:\Program Files (x86)\IBM\WebSphere
MQ\mqft\logs\MUNGEE\agents\MUNGEE\logs\di
agnostics.20121031.083420.0477.1.properti
es

Command Handler Worker Thread 0 Diagnostics:
Status: Waiting

Command Handler Worker Thread 1 Diagnostics:
Status: Waiting

Command Handler Worker Thread 2 Diagnostics:
Status: Waiting

Command Handler Worker Thread 3 Diagnostics:
Status: Waiting

Command Handler Worker Thread 4 Diagnostics:
Status: Waiting

File Transfer Diagnostics:
Source Transfers: 1
Destination Transfers: 2

File Transfer 0 Diagnostics:
Transfer Id: 414d5120514d43414e4445202020202079e20f5064230010
Role: SOURCE
State: ReSynchronisingTransfer
Status: INACTIVE
Start Time: Not started
Retry Count: 0
CheckPoint Index: 0
CheckPoint Position: 0

File Transfer 1 Diagnostics:
Transfer Id: 414d5120514d43414e44452020202020c8fbd54f144f0d20
Role: DESTINATION
State: RunningTransfer
CheckPoint Index: 0
CheckPoint Position: 0
Write Index: 0
Write Position: 0

File Transfer 2 Diagnostics:
Transfer Id: 414d5120514d43414e4445202020202079e20f5086020010
Role: DESTINATION
State: RunningTransfer

```

```

CheckPoint Index:          9
CheckPoint Position:       0
Write Index:               3
Write Position:            140923

Monitor 0 Diagnostics:
Name:                      MONITOR1
Status:                    STARTED
Resource Type:             directory
Resource:                  /tmp/monitor
Poll Interval:             1 minutes
Batch Size:                2
Condition:                 Match
Pattern:                   * (wildcard)
Executing:                 false
Last Execute Start Time:   2012-04-04T16:19:01.852Z
Last Execute End Time:     2012-04-04T16:19:01.852Z
Last Execute Match Count:  0

Schedule 1 Diagnostics:
Id:                        1
Next Trigger Time:         2012-07-17T16:00+0100
Occurrences So Far:        14
Repeat Interval:           hours
Repeat Frequency:          5
Source Agent:              AGCANDE
Destination Agent:         AGCANDE
Source File:                /tmp/source/a.txt, ...
Destination File:          /tmp/dest/a.txt, ...

```

在以下示例中，已停止绑定代理，并从该代理远程发出 **fteShowAgentDetails** 命令：

```

fteShowAgentDetails AGENT2
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Agent Information:
  Name:                     AGENT2
  Type:                     Standard
  Description:
  Operating System:         Linux
  Time Zone:                Greenwich Mean Time

Agent Controller Information:
  Controller type:          MQMFT Process Controller
  Status:                   UNKNOWN
  Status Details:           Information about the agent controller
                             is not available, either because the
                             agent is not running or the agent is
                             running on a different system.

  Agent Restarts within Interval: 0
  Total Agent Restart Count:    0

Agent Availability Information:
  Status:                   STOPPED
  Status Details:           The agent has been stopped. It was shut
                             down in a controlled manner.

Queue Manager Information:
  Name:                     QM2
  Transport:                Bindings
  Last Status Reported:     UNKNOWN
  Status Details:           Information about the queue manager is
                             not available, either because the agent
                             is not running or the agent is running
                             on a different system.

```

在以下示例中，绑定代理等待代理队列管理器停止后重新启动。代理在 Total Agent Restart Count: 1 之前已重新启动过一次，可能是由于先前的代理队列管理器重新启动：

注：针对队列管理器信息的 Last Status Reported 的 Last Error MQRC；即使队列管理器变为可用，此信息也将保留。

```

fteShowAgentDetails AGENT1
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Agent Information:
  Name:                     AGENT1
  Type:                     Standard
  Description:
  Operating System:         Windows Server 2003

```

```

Time Zone: Greenwich Mean Time

Agent Controller Information:
Controller type: MQMFT Process Controller
Status: WAITING
Status Details: The agent process controller is waiting
for the queue manager to become
available before starting the agent.

Agent Restarts within Interval: 0
Total Agent Restart Count: 1

Agent Availability Information:
Status: STOPPED
Status Details: The agent has been stopped. It was shut
down in a controlled manner.

Queue Manager Information:
Name: QM1
Transport: Bindings
Last Status Reported: UNAVAILABLE (Last Error MQRC: 2059)
Status Details: The queue manager is unavailable. It
might be that the queue manager has not
been started or an incorrect queue
manager name has been configured. Look
up the MQ reason code reported against
the status to understand the problem.

```

在以下示例中，客户机方式代理刚刚意外终止，代理进程控制器尝试通过在 `maxRestartDelay` 代理属性值指定的延迟后重新启动此代理，以恢复状况。缺省 `maxRestartDelay` 代理属性值为 -1，这将导致代理进程控制器终止；因此，在此示例中，`maxRestartDelay` 属性值必须设置为大于 0 的值。Current Agent Restart Count: 4 表示已在 `maxRestartInterval` 代理属性时间段内重新启动 4。如果 `maxRestartCount` 代理属性为 4，那么在 `maxRestartInterval` 以内重新启动 4 次后，代理进程控制器将在重新启动代理之前等待 `maxRestartDelay` 秒，也就是本例中的情况。Total Agent restart Count: 8 建议先前已发生此情况。该示例并不典型，您只能期望看到，当代理耗尽内存或定制用户出口导致某种运行时错误时，会发生代理意外终止。代理意外终止原因的完整详细信息位于代理的 `output0.log` 文件中：

```

fteShowAgentDetails AGENT3
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Agent Information:
Name: AGENT3
Type: Standard
Description:
Operating System: Windows Server 2003
Time Zone: Greenwich Mean Time

Agent Controller Information:
Controller type: MQMFT Process Controller
Status: RECOVERING
Status Details: The agent process unexpectedly stopped
and the process controller will attempt
to restart it.

Current Agent Restart Count: 4
Total Agent Restart Count: 8

Agent Availability Information:
Status: ENDED UNEXPECTEDLY
Status Details: The agent has ended unexpectedly due to
an unrecoverable problem. The agent
will be automatically restarted.

Queue Manager Information:
Name: QM3
Transport: Client
Host: host3.hursley.ibm.com
Port: 3031
Channel: SYSTEM.DEF.SVRCONN

```

在以下示例中，显示了 Connect:Direct 网桥代理的结果：

```

fteShowAgentDetails AG_CD1
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Agent Information:
Name: AG_CD1
Type: Connect:Direct bridge
Description:

```

```

Connect:Direct Node Name:      CDNODE
Connect:Direct Node Host:     localhost:1363
Operating System:              Windows Server 2003
Time Zone:                     Greenwich Mean Time

Agent Controller Information:
Controller type:                MQMFT Process Controller
Status:                        UNKNOWN
Status Details:                 Information about the agent controller
                                is not available, either because the
                                agent is not running or the agent is
                                running on a different system.

Agent Restarts within Interval: 0
Total Agent Restart Count:     0

Agent Availability Information:
Status:                        STOPPED
Status Details:                 The agent has been stopped. It was shut
                                down in a controlled manner.

Queue Manager Information:
Name:                          QM_JUPITER
Transport:                     Bindings
Last Status Reported:          UNKNOWN
Status Details:                 Information about the queue manager is
                                not available, either because the agent
                                is not running or the agent is running
                                on a different system.

```

在以下示例中，将向自动重新启动管理器 (ARM) 注册 z/OS 上运行的代理：

```

fteShowAgentDetails AGENTZ
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Agent Information:
Name:                          AGENTZ
Type:                          Standard
Description:
Operating System:              z/OS
Time Zone:                     Greenwich Mean Time

Agent Controller Information:
Controller Type:                z/OS Automatic Restart Manager (ARM)
Agent registered with ARM:      Yes (ELEMTYPE: SYSBFGAG, ELEMENT: AGENTZ)
Agent Restarted:                No

Agent Availability Information:
Status:                        READY
Status Details:                 The agent is running and is publishing
                                its status at regular intervals. The last
                                update was received within the expected
                                time period. The agent is ready to
                                process transfers, but none are currently
                                in progress.

Queue Manager Information:
Name:                          ZQM
Transport:                     Bindings
Last Status Reported:          AVAILABLE
Status Details:                 The queue manager is available.

```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关参考

第 532 页的『[fteListAgents](#)（列出协调队列管理器的 IBM MQ Managed File Transfer 代理）』

通过命令行，使用 **fteListAgents** 命令可列出向特定协调队列管理器注册的所有 IBM MQ Managed File Transfer 代理。

第 700 页的『[代理状态值](#)』

fteListAgents 和 **fteShowAgentDetails** 命令生成代理状态信息。该状态有多种可能的值。

第 701 页的『代理进程控制器状态值』

fteShowAgentDetails 命令会生成代理进程控制器状态信息。该状态有多种可能的值。

fteShowLoggerDetails（显示 IBM MQ Managed File Transfer 记录器详细信息）

使用 **fteShowLoggerDetails** 命令显示特定 IBM MQ Managed File Transfer 记录器的详细信息。

用途

您必须在记录器所在的相同系统上运行 **fteShowLoggerDetails** 命令。它显示记录器进程控制器和记录器队列管理器的状态，您可以使用该信息帮助确定问题。**fteShowLoggerDetails** 命令列出特定 IBM MQ Managed File Transfer 记录器的以下详细信息：

- 记录器控制器状态。
- 特定时间间隔内的记录器重新启动次数
- 记录器重新启动总计数
- 记录器可用性状态
- 记录器队列管理器名称
- 记录器队列管理器传输类型
- 记录器队列管理器上次报告的状态（仅适用于绑定传输方式）

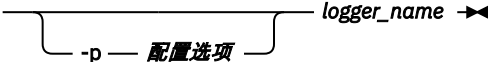
仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。请参阅第 116 页的『分布式平台上的配置选项』，以了解更多信息。

有关可能的记录器状态值及其含义的列表，请参阅第 701 页的『记录器状态值』。

有关记录器进程控制器的可能状态值及其含义的列表，请参阅第 702 页的『记录器进程控制器状态值』。

语法

fteShowLoggerDetails

➔ **fteShowLoggerDetails**  **logger_name** ➔

参数

-p configuration_options

可选。此参数确定用于发出请求以显示记录器详细信息的配置选项集。使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

logger_name

必需。要显示的 IBM MQ Managed File Transfer 记录器名称。

-? or -h

可选。显示命令语法。

示例

在本示例中，有一个已启动的记录器，将在本地向该记录器发出 **fteShowLoggerDetails** 命令：

```
fteShowLoggerDetails LOGGER1
```

```
Status: STARTED
Status Details: The logger process controller has
started the logger process.

Logger Restarts within Interval: 0
Total Logger Restart Count: 0

Queue Manager Information:
Name: QM_gbthink
Transport: Bindings
Last Status Reported: AVAILABLE
Status Details: The queue manager is available.
```

在本示例中，有一个记录器由于队列管理器不可用而正在等待，将在本地向该记录器发出 **fteShowLoggerDetails** 命令：

```
fteShowLoggerDetails LOGGER2

5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Logger Controller Information:
Status: WAITING
Status Details: The logger process controller is
waiting for the queue manager to
become available before starting the
logger.

Logger Restarts within Interval: 0
Total Logger Restart Count: 0

Logger Availability Information:
Status: STOPPED
Status Details: The logger has been stopped. It was
shut down in a controlled manner.

Queue Manager Information:
Name: QM_gbthink
Transport: Bindings
Last Status Reported: UNAVAILABLE (Last Error MQRC: 2059)
Status Details: The queue manager is unavailable. It
might be that the queue manager has
not been started or an incorrect
queue manager name has been
configured. Look up the MQ reason code
reported against the status to
understand the problem.
```

在本示例中，在 z/OS 上有一个正在运行的记录器（未向 ARM 注册）：

```
fteShowLoggerDetails loggerv8

5655-MFT, 5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Logger Controller Information:
Controller Type: z/OS Automatic Restart Manager (ARM)
Registered with ARM: No
Restarted: n/a

Queue Manager Information:
Name: FT8E
Transport: Bindings
Last Status Reported: AVAILABLE
Status Details: The queue manager is available.
```

在本示例中，在 z/OS 上有一个未运行或在其他系统上运行的记录器：

```
fteShowLoggerDetails loggerv8

5655-MFT, 5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Logger Controller Information:
Controller Type: UNKNOWN
```



```

Queue Manager Information:
  Name: FT8E
  Transport: Bindings
  Last Status Reported: UNKNOWN
  Status Details: Information about the queue manager is
                  not available, either because the
                  logger is not running, or the logger
                  is running on a different system.

```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关参考

第 701 页的『记录器状态值』

fteShowLoggerDetails 命令生成记录器状态信息。该状态有多种可能的值。

第 702 页的『记录器进程控制器状态值』

fteShowLoggerDetails 命令生成记录器进程控制器状态信息。该状态有多种可能的值。

fteStartAgent (启动 IBM MQ Managed File Transfer 代理)

fteStartAgent 命令从命令行启动 IBM MQ Managed File Transfer 代理程序。

用途

使用 **fteStartAgent** 命令可启动 IBM MQ Managed File Transfer 代理。必须先启动代理，才能使用它执行文件传输。**fteStartAgent** 命令在您发出命令的系统上启动代理：不能在远程系统上启动代理。

对于 IBM WebSphere MQ V7.5 或更高版本，代理进程控制器将管理代理的启动。但是，在某些情况下，例如当代理运行失败的频率比较高时，代理进程控制器可能会等待一段时间，然后再重新尝试启动代理。作为 IBM WebSphere MQ 管理员，可以使用 **fteStartAgent** 命令来覆盖此等待时间值，并开始代理启动过程。如果代理进程控制器在等待队列管理器变为可用，那么该命令还将启动尝试重新连接到队列管理器的代理进程控制器。

如果已通过使用 **fteCreateAgent** 或 **fteModifyAgent** 命令将代理配置为作为 Windows 服务运行，那么运行 **fteStartAgent** 命令将启动 Windows 服务。

如果代理未启动或已启动，那么该命令将返回错误。代理根据 `agent.properties` 文件中定义的值与其队列管理器通信。

仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。请参阅第 591 页的『`agent.properties` 文件』，以了解更多信息。

fteStartAgent 命令不适用于 IBM 4690 环境。有关在 IBM 4690 环境中使用 IBM MQ Managed File Transfer 的更多信息，请参阅第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』

语法

fteStartAgent

```

➔ fteStartAgent -F -p 配置选项 agent_name ➔

```

参数

-F

可选。该参数将代理守护程序作为前台进程运行。缺省值是针对要在后台运行的代理守护程序。

如果您正在 Windows 上运行，并且已将代理程序配置为使用 **fteCreateAgent** 或 **fteModifyAgent** 命令作为 Windows 服务运行，那么 **-F** 参数将覆盖此配置。

-p configuration_options

可选。该参数确定用于发出请求以启动代理的配置选项集。使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

agent_name

必需。要启动的 IBM MQ Managed File Transfer 代理的名称。

-? or -h

可选。显示命令语法。

示例

在本示例中，AGENT2 将启动并在前台运行。

```
fteStartAgent -F AGENT2
```

在以下示例中 (对于 UNIX 和 Linux 系统)，使用非缺省协调队列管理器 QM_SATURN 启动 AGENT2：

```
./fteStartAgent -p QM_SATURN AGENT2
```

您还可以按照如下方式指定 **fteStartAgent** 的路径来运行此命令：

```
<path>/fteStartAgent agentname
```

返回码

0

命令成功完成。

1

命令结束但未成功。

响应

在一些情况下，您可能会在运行 **fteStartAgent** 命令后看到错误消息：

- 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 IBM MQ Managed File Transfer 冲突的附加库路径：

```
BFGCL0001E: An internal error has occurred. The exception was: 'CC=2;RC=2495;AMQ8568:  
The native JNI library 'mqjbnd' was not found. [3=mqjbnd]
```

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，那么当代理程序使用 32 位版本的 Java 运行时 (当前对于大多数平台是如此)，将发生此错误。

要解决此问题，请设置 IBM MQ Managed File Transfer 代理属性 javaLibraryPath 以引用库的正确位置。例如，对于 AIX 上的 mqjbnd，请设置为: /usr/mqm/java/lib。对于 Linux 上的 mqjbnd，请设置为: /opt/mqm/java/lib

相关任务

第 216 页的『将代理作为 Windows 服务启动』

您可以将代理作为 Windows 服务启动，以便当您从 Windows 注销后，代理可以继续运行，并且可以接收文件传输。

第 273 页的『列出 IBM MQ Managed File Transfer 代理』

可使用命令行或 IBM MQ Explorer 来列出已向特定队列管理器注册的代理。

第 273 页的『[停止 IBM MQ Managed File Transfer 代理](#)』

您可以通过命令行停止代理。在停止代理时，您将停顿代理并允许代理在停止前完成其当前文件传输。您还可以在命令行上指定 **-i** 参数来立即停止代理。在代理停止时，您无法使用该代理传输文件，直至重新启动代理。

相关参考

第 216 页的『[在 z/OS 上启动代理程序](#)』

在 z/OS 上，除了通过 UNIX 系统服务会话运行 **fteStartAgent** 命令，您还可以通过 JCL 将代理作为已启动任务启动，而无需交互式会话。

fteStartLogger（启动记录器）

fteStartLogger 命令启动 IBM MQ Managed File Transfer 日志记录应用程序。

用途

使用 **fteStartLogger** 命令可启动记录器。记录器可以是一个文件，或者是与协调队列管理器在相同的系统上运行的数据库应用程序。有关更多信息，请参阅第 151 页的『[配置 Managed File Transfer 记录器](#)』主题。对于 IBM WebSphere MQ V7.5 或更高版本，记录器进程控制器将管理记录器的启动。但是，在某些情况下，例如当记录器运行失败的频率比较高时，记录器进程控制器可能会等待一段时间，然后再重新尝试启动记录器。作为 IBM WebSphere MQ 管理员，您可以使用 **fteStartLogger** 命令来覆盖此等待时间值，并开始启动记录器。如果记录器进程控制器在等待队列管理器变为可用，那么该命令还将启动尝试重新连接到队列管理器的记录器进程控制器。

如果您已通过使用 **fteModifyLogger** 命令将记录器配置为作为 Windows 服务运行，那么运行 **fteStartLogger** 命令将启动 Windows 服务。

如果记录器无法启动或已启动，那么该命令将返回错误。记录器根据 `logger.properties` 文件中定义的值与其队列管理器通信。

仅当您使用一组不同于缺省设置的配置选项时，才可为该命令指定 **-p** 参数。有关记录器属性的更多信息，请参阅第 163 页的『[IBM MQ Managed File Transfer 的记录器配置属性](#)』

语法

fteStartLogger

► fteStartLogger  **logger_name**  ? 

参数

logger_name

必需。想要启动的 IBM MQ Managed File Transfer 记录器的名称。

-p configuration_options

可选。此参数确定用于发出请求以启动记录器的配置选项集。使用非缺省协调队列管理器的名称作为该参数的输入。然后，**fteStartLogger** 使用与此非缺省协调队列管理器关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-F

可选。将记录器作为前台进程运行（而非作为缺省后台进程）。如果已使用 **fteModifyLogger** 命令将记录器配置为作为 Windows 服务运行，那么 **-F** 参数将覆盖此配置。

-? or -h

可选。显示命令语法。

示例

在此示例中，预先创建了名为 `logger1` 的记录器。该命令显示如何作为前台进程启动记录器：

```
fteStartLogger -F logger1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

[第 151 页的『配置 Managed File Transfer 记录器』](#)

相关参考

[第 548 页的『fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）』](#)

使用 **fteModifyLogger** 命令修改记录器，以便记录器能够作为 Windows 服务运行。只能在 Windows 平台上使用此命令，必须由作为 IBM MQ 管理员和 `mqm` 组成员的用户运行，并且必须首先使用

fteStopLogger 命令停止记录器。

[第 579 页的『fteStopLogger（停止记录器）』](#)

fteStopLogger 命令停止记录器。

[第 402 页的『记录器错误处理和拒绝』](#)

记录器确定两种类型的错误：每条消息的错误和通用错误。

fteStopAgent（停止 IBM MQ Managed File Transfer 代理）

使用 **fteStopAgent** 命令以受控方式停止 IBM MQ Managed File Transfer 代理程序，或者使用 **-i** 参数在必要时立即停止代理程序。

用途

在使用 **fteStopAgent** 命令停止代理时，您可以允许代理在停止之前完成当前文件传输，或者立即停止代理，即使代理当前正在传输文件。在代理停止后，您无法使用代理来传输文件，直至重新启动代理。

如果您希望停止的代理已连接到 IBM MQ 网络，那么可以从可连接到 IBM MQ 网络并路由至代理队列管理器的任何系统中运行 **fteStopAgent** 命令。具体而言，要运行此命令，您必须已在该系统上安装了 IBM MQ Managed File Transfer 组件（服务或代理程序），并将该组件配置为与 IBM MQ 网络进行通信。如果没有可用的连接详细信息，那么将建立与本地系统上缺省队列管理器的绑定方式连接。如果 `command.properties` 不存在，那么将生成错误。

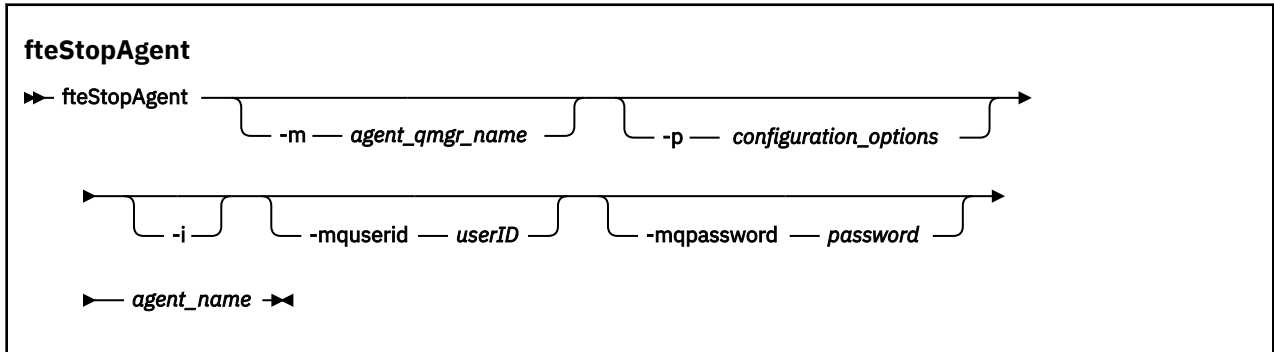
如果您希望停止的代理未连接到 IBM MQ 网络（例如，如果 IBM MQ 网络当前不可用），那么只能从运行该代理的系统中运行 **fteStopAgent** 命令。要停止未连接到 IBM MQ 网络的代理程序，必须从运行代理程序的用户运行 **fteStopAgent** 命令。或者，如果代理在 Windows 系统上运行，那么您可以作为管理员来运行该命令。

仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。请参阅 [第 591 页的『agent.properties 文件』](#)，以了解更多信息。

如果代理程序作为 Windows 服务运行，那么运行 **fteStopAgent** 命令将停止 Windows 服务。要了解更多信息，请参阅 [第 216 页的『将代理作为 Windows 服务启动』](#)。

fteStopAgent 命令不适用于 IBM 4690 环境。有关在 IBM 4690 环境中使用 IBM MQ Managed File Transfer 的更多信息，请参阅 [第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』](#)

语法



参数

-m (*agent_qmgr_name*)

可选。想要停止的代理连接到的队列管理器的名称。

如果代理位于远程系统上，或位于本地系统但您不是启动它的用户，那么必须使用 **-m** 参数并具有相应的权限。有关权限的更多信息，请参阅第 433 页的『特定于 IBM MQ Managed File Transfer 的资源的组权限』。

-p (配置选项)

可选。该参数确定用于发出请求以停止代理的配置选项集。使用非缺省协调队列管理器的名称作为该参数的输入。然后，命令使用与此非缺省协调队列管理器相关联的属性文件集。

如果未指定该参数，那么将使用基于缺省协调队列管理器的配置选项集。

-i

可选。立即停止代理。代理不会完成当前正在执行的任何传输。

如果未指定 **-i** 参数，那么代理完成当前正在执行的任何传输，但不会启动任何新传输。

-mquserid (*userID*)

可选。指定要向命令队列管理器进行认证的用户标识。

-mqpassword (*password*)

可选。指定要向命令队列管理器进行认证的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

agent_name

必需。想要停止的 IBM MQ Managed File Transfer 代理的名称。

-? or -h

可选。显示命令语法。

示例

在此示例中，将停止队列管理器 QM_JUPITER 上的代理 AGENT2。将使用 **-m** 参数，因为 AGENT2 连接到的这一队列管理器不同于配置选项集指定的队列管理器。

```
fteStopAgent -m QM_JUPITER AGENT2
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关任务

第 273 页的『停止 IBM MQ Managed File Transfer 代理』

您可以通过命令行停止代理。在停止代理时，您将停顿代理并允许代理在停止前完成其当前文件传输。您还可以在命令行上指定 **-i** 参数来立即停止代理。在代理停止时，您无法使用该代理传输文件，直至重新启动代理。

相关参考

第 573 页的『fteStartAgent（启动 IBM MQ Managed File Transfer 代理）』

fteStartAgent 命令从命令行启动 IBM MQ Managed File Transfer 代理程序。

第 274 页的『在 z/OS 上停止代理』

如果要在 z/OS 上作为来自 JCL 的启动式任务运行 IBM MQ Managed File Transfer 代理程序，那么该代理程序将接受 z/OS 操作员命令 **MODIFY** 和 **STOP** 以及 **fteStopAgent** 命令。

fteStopDatabaseLogger（停止独立数据库记录器）

fteStopDatabaseLogger 命令停止独立数据库记录器。

用途

fteStopDatabaseLogger 命令在 IBM MQ Managed File Transfer V7.0.1 以及更高版本上受支持。

使用 **fteStopDatabaseLogger** 命令以停止独立数据库记录器。独立数据库记录器是一个独立 Java 应用程序，与协调队列管理器和数据库在同一系统上运行。

有关停止独立数据库记录器的其他注意事项

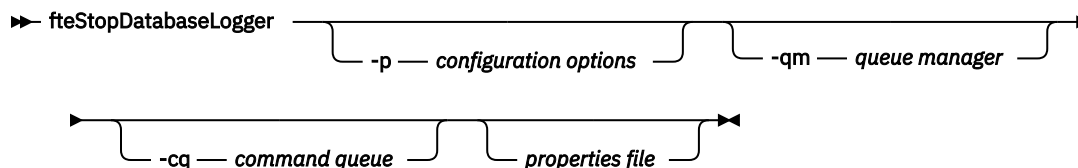
fteStopDatabaseLogger 命令将消息发送给独立数据库记录器使用的命令队列。如果运行 **fteStopDatabaseLogger**，而独立数据库记录器未在运行，那么命令消息仍将添加到队列。在独立数据库记录器下一次启动时，记录器立即接收此命令消息，然后关闭。如果向一个未在运行的独立数据库记录器发出多个停止命令，那么必须重复启动记录器直至消耗完所有停止命令。或者，您可以清除命令队列以移除所有暂挂命令。

如果您的独立数据库记录器正在作为 Windows 服务运行，那么运行 **fteStopDatabaseLogger** 命令将停止 Windows 服务。

某些错误条件，通常伴随消息 BFGDB0038E，将阻止独立数据库记录器读取命令。要停止处于该状态的独立数据库记录器，请使用您的操作系统设施以结束进程（例如，UNIX **kill** 命令或 Windows 任务管理器）。独立数据库记录器使用的 XA 事务协议确保在结束进程时不会丢失任何消息。

语法

fteStopDatabaseLogger



参数

-p (configuration options)

可选。确定用于停止独立数据库记录器的配置选项集。将配置选项集的名称用作 **-p** 参数的值。按照约定，该值是协调队列管理器的名称。如果未指定该参数，那么会使用缺省的配置选项集。

-qm (queue manager)

可选。缺省情况下，独立数据库记录器使用的命令队列将假定为位于 **-p** 参数（或其缺省值）指定的协调队列管理器上。如果需要将独立数据库记录器命令发送到任意位置上的命令队列，那么 **-qm** 参数可用。

于指定备用目标。在所有情况下，请注意，命令连接到 **-p** 参数指示的命令队列管理器，不管消息的最终目标为何。

-cq (command queue)

可选。指定要向其发送停止消息的命令队列。在大多数情况下，独立数据库记录器使用缺省队列名称，并且该参数非必需。

properties file

可选。缺省情况下，独立数据库记录器的属性文件假定为位于协调队列管理器的目录中。您可以选择提供自己的属性文件标准路径，其中包含独立数据库记录器运行所需的属性。如果针对 **fteStartDatabaseLogger** 命令指定了属性文件，那么针对此命令指定相同的属性文件。

-? or -h

可选。显示命令语法。

示例

在此示例中，将停止独立数据库记录器以及队列管理器 PLUTO 上的命令队列 FTE.LOGGER2.COMMAND。

```
fteStopDatabaseLogger -qm PLUTO -cq FTE.LOGGER2.COMMAND
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

[第 151 页的『配置 Managed File Transfer 记录器』](#)

fteStopLogger (停止记录器)

fteStopLogger 命令停止记录器。

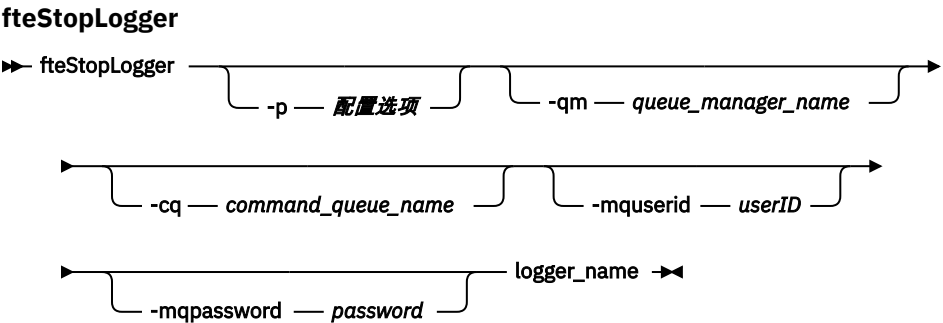
用途

使用 **fteStopLogger** 命令以停止记录器。记录器可以是将受管文件传输活动的历史记录记录到文件中的文件记录器，也可以是将历史记录记录到数据库中的数据库记录器。

有关停止独立记录器进程的其他注意事项

如果记录器作为 Windows 服务运行，那么运行 **fteStopLogger** 命令将停止 Windows 服务。

语法



参数

-p (配置选项)

可选。确定用于停止记录器的配置选项集。将配置选项集的名称用作 **-p** 参数的值。按照约定，该值是协调队列管理器的名称。如果未指定该参数，那么会使用缺省的配置选项集。

-qm (*queue_manager_name*)

可选。缺省情况下，记录器的命令队列假定位于 **-p** 参数（或其缺省值）指定的协调队列管理器上。如果要将记录器命令发送到位于其他位置的命令队列，请使用 **-qm** 参数指定备用目标。在所有情况下，该命令都会连接到 **-p** 参数指示的命令队列管理器，不管消息的最终目标是什么。

-cq (*command_queue_name*)

可选。指定要向其发送停止消息的命令队列。在大多数情况下，记录器使用缺省队列名称，表示该参数非必需参数。

-mquserid (*userID*)

可选。指定要向命令队列管理器进行认证的用户标识。

-mqpassword (*password*)

可选。指定要向命令队列管理器进行认证的密码。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，将会提示您提供相关联的密码。将不会显示密码。

logger_name

必需。想要停止的 IBM MQ Managed File Transfer 记录器的名称。

-? or -h

可选。显示命令语法。

示例

在此示例中，预先创建了名为 `logger1` 的记录器，而且其当前正在运行。该命令显示如何停止记录器：

```
fteStopLogger logger1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

[第 151 页的『配置 Managed File Transfer 记录器』](#)

相关参考

[第 548 页的『fteModifyLogger（作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序）』](#)

使用 **fteModifyLogger** 命令修改记录器，以便记录器能够作为 Windows 服务运行。只能在 Windows 平台上使用此命令，必须由作为 IBM MQ 管理员和 mqm 组成员的用户运行，并且必须首先使用 **fteStopLogger** 命令停止记录器。

[第 575 页的『fteStartLogger（启动记录器）』](#)

fteStartLogger 命令启动 IBM MQ Managed File Transfer 日志记录应用程序。

配置

在 IBM MQ Managed File Transfer 属性中使用环境变量

从 IBM WebSphere MQ V7.5 起，可以在受管文件传输属性中使用表示文件或目录位置的环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化。例如，哪个用户在运行进程。

以下属性可接受文件或目录位置，因此可以包含环境变量：

- agentSslKeyStore
- agentSslKeyStoreCredentialsFile
- agentSslTrustStore
- agentSslTrustStoreCredentialsFile
- cdNodeKeystoreCredentialsFile
- cdNodeTruststoreCredentialsFile
- cdTmpDir
- cdNodeKeystore
- cdNodeTruststore
- commandPath
- connectionSslKeyStore
- connectionSslKeyStoreCredentialsFile
- connectionSslTrustStore
- connectionSslTrustStoreCredentialsFile
- coordinationSslKeyStore
- coordinationSslKeyStoreCredentialsFile
- coordinationSslTrustStore
- coordinationSslTrustStoreCredentialsFile
- exitClassPath
- exitNativeLibraryPath
- javaCoreTriggerFile
- sandboxRoot
- transferRoot
- wmqfte.database.credentials.file

示例

本示例表示在 Windows 系统上，用户 `fteuser` 使用环境变量 `USERPROFILE`：

```
wmqfte.database.credentials.file=%USERPROFILE%\logger\mqmftcredentials.xml
```

解析为以下文件路径：

```
C:\Users\fteuser\logger\mqmftcredentials.xml
```

本示例表示在 UNIX 系统上，用户 `fteuser` 使用环境变量 `HOME`：

```
transferRoot=$HOME/fte/
```

解析为以下文件路径：

```
/home/fteuser/fte/
```

相关参考

第 584 页的『[coordination.properties 文件](#)』

`coordination.properties` 文件指定到协调队列管理器的连接详细信息。由于多个 IBM MQ Managed File Transfer 安装可能共享同一个协调队列管理器，因此您可以使用指向共享驱动器上的公共 `coordination.properties` 文件的符号链接。

[第 588 页的『command.properties 文件』](#)

`command.properties` 文件指定发出命令时要连接的命令队列管理器以及 IBM MQ Managed File Transfer 联系该队列管理器所需的信息。

[第 591 页的『agent.properties 文件』](#)

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

[第 634 页的『SSL 属性』](#)

将 SSL 或 TLS 与 IBM MQ 和 IBM MQ Managed File Transfer 一起使用可防止在代理与队列管理器之间建立未经授权的连接，并可对代理与队列管理器之间的消息通信进行加密。

[第 163 页的『IBM MQ Managed File Transfer 的记录器配置属性』](#)

该记录器具有一组配置属性。在 `logger.properties` 文件中指定这些属性，该文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中。

[第 977 页的『用户出口的代理属性』](#)

除 `agent.properties` 文件中的标准属性外，还存在多个特定于用户出口例程的高级属性。缺省情况下不包含这些属性，因此要使用它们中的任何一个，都必须手动编辑 `agent.properties` 文件。如果在代理运行期间对 `agent.properties` 文件进行了更改，请停止并重新启动该代理以提取这些更改。

[第 611 页的『协议网桥属性文件格式』](#)

代理配置目录中的 `ProtocolBridgeProperties.xml` 文件定义协议文件服务器的属性。

[第 621 页的『Connect:Direct 节点属性文件格式』](#)

Connect:Direct 网桥代理配置目录中的 `ConnectDirectNodeProperties.xml` 文件指定有关文件传输中涉及的远程 Connect:Direct 节点的信息。

[第 623 页的『Connect:Direct 进程定义文件格式』](#)

Connect:Direct 网桥代理配置目录中的 `ConnectDirectProcessDefinitions.xml` 文件指定要在文件传输过程中启动的用户定义的 Connect:Direct 进程。

installation.properties 文件

`installation.properties` 文件指定一组缺省配置选项的名称。该条目将 IBM MQ Managed File Transfer 指向包含要使用的配置的一组结构化目录和属性文件。通常，配置选项集的名称是关联的协调队列管理器的名称。

该文件由安装程序创建，可通过使用 **`fteChangeDefaultConfigurationOptions`** 命令进行更改。

`installation.properties` 文件位于您的 `MQ_DATA_PATH` 目录中。例如，在 Windows 上，缺省文件位置为 `MQ_DATA_PATH\mqft\installations\installation_name`，而在 UNIX 和 Linux 系统上，缺省文件位置为 `/var/mqm/mqft/installations/installation_name`。

`installation.properties` 文件包含以下值：

表 44: 基本属性

属性名	描述	缺省值
commandMessagePriority	为 fteStopAgent、fteCancelTransfer 和 ftePingAgent 命令的内部消息和命令消息设置优先级。 例如，如果您提交大量传输请求以快速连续传输小文件，那么新传输请求会在源代理的命令队列中排队。外部和内部消息具有缺省 IBM MQ 消息优先级，因此内部消息会被新传输请求阻止。这可能导致超出传输协商时间以及恢复传输。 您还可以使用 commandMessagePriority 属性设置内部确认和期望确认消息的优先级。 要将内部 IBM MQ Managed File Transfer 消息的优先级设置为高于新的传输请求，请将该属性设置为 1（最低值）到 9（最高值）之间的值。 V 8.0.0.3 从 IBM MQ 8.0.0 修订包 3 开始，缺省值将更改为 8。这意味着，如果代理命令队列上的 IBM MQ 属性 DEFPRTY (缺省优先级) 小于或等于 7，那么将在新的传输请求之前对内部协商消息进行优先级划分。如果将 DEFPRTY 属性的值设置为 8 或 9 以保持 commandMessagePriority 属性的有效性，必须更改 DEFPRTY 或 commandMessagePriority 属性。	在 IBM MQ 8.0.0 修订包 3 之前，缺省值为 MQPRI_PRIORITY_AS_Q_DEF 常量，具有值 -1。 V 8.0.0.3 从 IBM MQ 8.0.0 修订包 3 开始，缺省值为 8。
defaultProperties	缺省配置选项集的名称。该值是位于配置目录中的目录名称，它包含指定配置信息的目录和属性文件。	没有缺省值
enableFunctionalFixPack	启用修订包功能级别。缺省情况下，不会启用修订包中的任何新功能。将该属性设置为版本标识，以启用该版本提供的新功能。 可以指定带有或不带句点字符 (.) 的版本标识。例如，要使用 IBM MQ 8.0.0 修订包 2 提供的函数，请将此属性设置为 8002 或 8.0.0.2。	没有缺省值

表 44: 基本属性 (继续)		
属性名	描述	缺省值
V 8.0.0.7 V 8.0.0.7 messagePublicationFormat	允许您指定 MFT 代理对其状态 XML 消息使用的消息发布格式。可以将该属性设置为以下值： messagePublicationFormat=mixed 该值是 版本 8.0 的缺省值。不采用任何格式发布消息，使用主题字符串 /LOG 发布到 SYSTEM.FTE 主题的消息除外，其格式为 String。 messagePublicationFormat=MQFMT_NONE MQMD.FORMAT 设置为 MQFMT_NONE。 messagePublicationFormat=MQFMT_STRING MQMD.FORMAT 设置为 MQFMT_STRING。 在 版本 8.0 之前，MFT 代理采用字符串格式 (MQFMT_STRING) 将 XML 状态消息发布至 SYSTEM.FTE 主题。如果可以，必须更新先前使用 版本 7.5 的应用程序，以采用 版本 8.0 格式处理消息。如果无法更改应用程序，请将 messagePublicationFormat 属性设置为 string 以还原至 版本 7.5 行为。	messagePublicationFormat=mixed

以下文本是 installation.properties 文件的内容示例。

```
defaultProperties=ERIS
```

ERIS 是一个目录名称，与 installation.properties 文件位于同一个目录中。目录 ERIS 包含描述一组配置选项的目录和属性文件。

相关概念

第 116 页的『分布式平台上的配置选项』

IBM MQ Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。这些属性文件位于安装产品时所定义的配置目录中。

相关参考

第 454 页的『fteChangeDefaultConfigurationOptions（更改缺省配置选项）』

使用 **fteChangeDefaultConfigurationOptions** 命令来更改您希望 IBM MQ Managed File Transfer 使用的缺省配置选项。配置选项的值定义 IBM MQ Managed File Transfer 使用的属性文件组。

coordination.properties 文件

coordination.properties 文件指定到协调队列管理器的连接详细信息。由于多个 IBM MQ Managed File Transfer 安装可能共享同一个协调队列管理器，因此您可以使用指向共享驱动器上的公共 coordination.properties 文件的符号链接。

`coordination.properties` 文件是由安装程序或 **fteSetupCoordination** 命令创建的。您可以使用具有 **-f** 标志的 **fteSetupCoordination** 命令更改此文件中的基本协调队列管理器属性。要更改或添加高级协调队列管理器属性，您必须在文本编辑器中编辑此文件。

`coordination.properties` 文件位于您的 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目录中。

`coordination.properties` 文件包含以下值：

表 45: 协调队列管理器属性		
属性名	描述	缺省值
<code>coordinationQMGr</code>	协调队列管理器的名称。	没有缺省值
<code>coordinationQMGrHost</code>	协调队列管理器的主机名或 IP 地址。	没有缺省值
<code>coordinationQMGrPort</code>	用于以客户机方式连接协调队列管理器的端口号。	1414
<code>coordinationQMGrChannel</code>	用于与协调队列管理器连接的 SVRCONN 通道名称。	SYSTEM.DEF.SVRCONN

如果没有为 `coordinationQMGrHost` 属性指定值，那么缺省情况下使用绑定方式。

如果您为 `coordinationQMGrHost` 属性指定值，但未为 `coordinationQMGrPort` 和 `coordinationQMGrChannel` 属性指定值，那么缺省情况下使用端口号 1414 和通道 SYSTEM.DEF.SVRCONN。

表 46: 高级协调队列管理器属性		
属性名	描述	缺省值
代理属性：		
<code>agentStatusJitterTolerance</code>	消息视为过期之前，代理状态消息发布延迟的最大时间量。该值度量单位为毫秒。 状态消息的期限取决于在协调队列管理器上发布的时间。但是，代理会在协调队列管理器接收消息之前的一段时间发出消息，以腾出在 IBM MQ 网络中传输所需的时间。如果该传输所用时间始终相同，那么每隔 60 秒创建的消息，每隔 60 秒就会发布，而与实际传输时间无关。但是，如果消息之间的传输时间不同，那么可能每隔 60 秒创建消息，但每隔诸如 61、59、58 和 62 秒等时间发布消息。60 的最大偏差为抖动（在该示例中为 2 秒）。该属性确定消息视为过期之前，由于抖动造成的最大延迟。	3000
代码页属性：		

表 46: 高级协调队列管理器属性 (继续)		
属性名	描述	缺省值
coordinationCcsid	命令连接协作队列管理器所使用的代码页。代理还使用该代码页执行至协调队列管理器的任何发布。如果为 coordinationCcsid 指定值, 那么还必须为 coordinationCcsidName 指定值。	1208
coordinationCcsidName	coordinationCcsid 的 Java 表示。如果为 coordinationCcsidName 指定值, 那么还必须为 coordinationCcsid 指定值。	UTF8
连接属性:		
javaLibraryPath	以绑定方式连接到队列管理器时, IBM MQ Managed File Transfer 必须有权访问 IBM MQ Java 绑定库。缺省情况下, IBM MQ Managed File Transfer 在 IBM MQ 定义的缺省位置中查找绑定库。如果绑定库位于不同位置, 那么请使用该属性以指定绑定库的位置。	MQ_INSTALLATION_PATH/ java/lib
多实例队列管理器属性:		
coordinationQMgrStandby	针对通过 coordinationQMgr 属性定义的多实例协调队列管理器的备用实例, 用于客户机连接的主机名和端口号, 采用 IBM MQ CONNAME 格式。例如, <i>host_name(port_number)</i>	没有缺省值
队列属性:		
dynamicQueuePrefix	该属性定义 IBM MQ 前缀, 以用于生成临时队列名称。 dynamicQueuePrefix 属性的格式遵循 IBM MQ MQOD 结构的 DynamicQName 字段的格式。有关更多信息, 请参阅创建动态队列。 如果您要将特定的 IBM MQ 前缀用于由需要代理响应的命令生成的临时应答队列, 那么还可以在 <code>command.properties</code> 文件中定义此属性。	WMQFTE.*

表 46: 高级协调队列管理器属性 (继续)		
属性名	描述	缺省值
modelQueueName	该属性定义 IBM MQ 模型队列，以用于生成临时队列。 如果您要将特定的 IBM MQ 模型队列用于由需要代理响应的命令生成的临时应答队列，那么还可以在 <code>command.properties</code> 文件中定义此属性。有关更多信息，请参阅 第 588 页的『<code>command.properties</code> 文件』。	SYSTEM.DEFAULT.MODEL.QUEUE
安全属性:		
userIdForClientConnect	经客户机连接流至 IBM MQ 的用户标识。如果指定了 <code>java</code> ，那么 JVM 报告的用户名会作为 IBM MQ 连接请求的一部分流动。该属性的值可以为 <code>None</code> 或 <code>java</code> 。	None
coordinationQMGrAuthenticationCredentialsFile	包含用于连接到协调队列管理器的 MQ 连接凭证的文件的途径。	此属性的缺省值为 <code>%HOMEDRIVE%%HOMEPATH%\mqmftcredentials.xml</code> (在 Windows 上) 和 <code>\$HOME/MQMFTCredentials.xml</code> (在其他平台上)。
预订属性:		
coordinationSubscriptionTopic	使用此属性指定 <code>SYSTEM.FTE</code> 以外要预订的主题，以便获取有关受管文件传输网络状态的出版物。所有工具仍会发布至 <code>SYSTEM.FTE</code> 主题，但您可以更改自己的 IBM MQ 拓扑，以根据内容将这些出版物发布至不同的主题。然后，您可以使用此功能强制工具预订其他主题中的一个主题。 对于 IBM MQ V 7.5 和更高版本的修订包，您需要 APAR IC96850 的临时修订，IBM MQ Explorer 插件和 <code>fteListMonitors</code> 命令才能识别该属性。	SYSTEM.FTE

以下文本是 `coordination.properties` 文件的内容示例。

```
coordinationQMGr=ERIS
coordinationQMGrHost=kuiper.example.com
coordinationQMGrPort=2005
coordinationQMGrChannel=SYSTEM.DEF.SVRCONN
```

ERIS 是位于系统 `kuiper.example.com` 上的 WebSphere MQ 队列管理器的名称。队列管理器 ERIS 是 IBM MQ Managed File Transfer 向其发送日志信息的队列管理器。

相关概念

第 116 页的『分布式平台上的配置选项』
IBM MQ Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。这些属性文件位于安装产品时所定义的配置目录中。

相关参考

第 562 页的『fteSetupCoordination（设置协调详细信息）』
fteSetupCoordination 命令为 IBM MQ Managed File Transfer 创建属性文件和协调队列管理器目录。

command.properties 文件

command.properties 文件指定发出命令时要连接的命令队列管理器以及 IBM MQ Managed File Transfer 联系该队列管理器所需的信息。

command.properties 文件是由安装程序或 **fteSetupCommands** 命令创建的。您可以使用具有 **-f** 标志的 **fteSetupCommands** 命令更改此文件中的基本命令队列管理器属性。要更改或添加高级命令队列管理器属性，您必须在文本编辑器中编辑此文件。

某些 IBM MQ Managed File Transfer 命令连接到代理队列管理器或协调队列管理器，而不是命令队列管理器。有关哪些命令与哪个队列管理器连接的信息，请参阅第 445 页的『MFT 命令和进程与队列管理器的连接关系』。

command.properties 文件位于您的 MQ_DATA_PATH/mqft/config/coordination_qmgr_name 目录中。

command.properties 文件包含以下值：

表 47: 基本命令队列管理器属性		
属性名	描述	缺省值
connectionQMgr	用于连接到 IBM MQ 网络的队列管理器的名称。	没有缺省值
connectionQMgrHost	连接队列管理器的主机名或 IP 地址。	没有缺省值
connectionQMgrPort	用于以客户机方式连接到连接队列管理器的端口号。	1414
connectionQMgrChannel	用于与连接队列管理器连接的 SVRCONN 通道名称。	SYSTEM.DEF.SVRCONN

如果没有为 connectionQMgrHost 属性指定值，那么缺省情况下使用绑定方式。

如果您为 connectionQMgrHost 属性指定值，但没有为 connectionQMgrPort 和 connectionQMgrChannel 属性指定值，那么缺省情况下使用端口号 1414 和通道 SYSTEM.DEF.SVRCONN。

表 48: 高级命令队列管理器属性		
属性名	描述	缺省值
代码页属性：		
connectionCcsid	命令连接命令队列管理器所使用的代码页。如果为 connectionCcsid 指定值，那么还必须为 connectionCcsidName 指定值。	1208
connectionCcsidName	connectionCcsid 的 Java 表示。如果为 connectionCcsidName 指定值，那么还必须为 connectionCcsid 指定值。	UTF8
多实例队列管理器属性：		

表 48: 高级命令队列管理器属性 (继续)		
属性名	描述	缺省值
connectionQMGrStandby	由 connectionQMGr 属性定义的多实例命令队列管理器的备用实例在建立客户机连接时所使用的本机名和端口号, 采用 IBM MQ CONNAME 格式。 例如, <i>host_name(port_number)</i>	没有缺省值
安全属性:		
userIdForClientConnect	经客户机连接流至 IBM MQ 的用户标识。如果指定了 <i>java</i> , 那么 JVM 报告的用户名会作为 IBM MQ 连接请求的一部分流动。该属性的值可以为 None 或 <i>java</i> 。	None
connectionQMGrAuthenticationCredentials 文件	包含用于连接到命令队列管理器的 MQ 连接凭证的文件的途径。	此属性的缺省值为 %HOMEDRIVE%\%HOMEPATH%\mqmftcredentials.xml (在 Windows 上) 和 \$HOME/MQMFTCredentials.xml (在其他平台上)。
队列属性:		
dynamicQueuePrefix	对于需要代理响应的命令, 该属性定义 IBM MQ 前缀以用于生成临时应答队列名称。 dynamicQueuePrefix 属性的格式遵循 IBM MQ MQOD 结构的 DynamicQName 字段的格式。有关更多信息, 请参阅 创建动态队列 。 如果要将特定 IBM MQ 前缀用于 WMQFTE 生成的临时队列, 那么还可以在 <code>coordination.properties</code> 文件中定义此属性。	WMQFTE.*
modelQueueName	对于需要代理响应的命令, 该属性定义 IBM MQ 模型队列以用于生成临时应答队列。 如果您要将特定 IBM MQ 模型队列用于由 WMQFTE 生成的临时队列, 那么您还可以在 <code>coordination.properties</code> 文件中定义该属性。有关更多信息, 请参阅 第 584 页的『coordination.properties 文件』 。	SYSTEM.DEFAULT.MODEL.QUEUE
连接属性:		

表 48: 高级命令队列管理器属性 (继续)		
属性名	描述	缺省值
javaLibraryPath	以绑定方式连接到队列管理器时，IBM MQ Managed File Transfer 必须有权访问 IBM MQ Java 绑定库。缺省情况下，IBM MQ Managed File Transfer 在 IBM MQ 定义的缺省位置中查找绑定库。如果绑定库位于不同位置，那么请使用该属性以指定绑定库的位置。	/opt/mqm/java/lib
legacyXMLMessageMQMDFormat	IBM MQ Managed File Transfer 命令 XML 消息现在将发送至具有空白 MQMD 格式字段的队列。先前版本的产品将 MQMD 格式字段设置为 MQSTR（文本消息字符串）。如果将此属性设置为 true，那么就能够将 IBM MQ Managed File Transfer 命令 XML 消息发送至 MQMD 格式字段为 MQSTR 的队列。 如果将 MQMD 格式字段设置为 MQSTR，并且如果在启用了数据转换的 MQ 网络中存在通道，那么 IBM MQ Managed File Transfer 命令 XML 消息可能会损坏。	false
 failCleanAgentWithNoArguments	缺省情况下，此属性的值为 true，这表示如果仅指定代理名称参数，那么 fteCleanAgent 命令将运行失败。将此属性设置为 false 表示，如果仅设置代理名称参数，那么 fteCleanAgent 命令的行为与指定 -all 参数的情况下的行为相同。	true

以下文本是 `command.properties` 文件的内容示例。

```
connectionQMgr=PLUTO
connectionQMgrHost=kuiper.example.com
connectionQMgrPort=1930
connectionQMgrChannel=SYSTEM.DEF.SVRCONN
```

PLUTO 是位于系统 `kuiper.example.com` 上的 IBM MQ 队列管理器的名称。队列管理器 PLUTO 是 IBM MQ Managed File Transfer 命令连接的队列管理器。

相关概念

[第 116 页的『分布式平台上的配置选项』](#)

IBM MQ Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。这些属性文件位于安装产品时所定义的配置目录中。

相关参考

[第 633 页的『Java 系统属性』](#)

必须将许多 IBM MQ Managed File Transfer 命令和代理程序属性定义为 Java 系统属性，因为它们定义了无法使用命令或代理程序属性机制的早期功能的配置。

[第 634 页的『SSL 属性』](#)

将 SSL 或 TLS 与 IBM MQ 和 IBM MQ Managed File Transfer 一起使用可防止在代理与队列管理器之间建立未经授权的连接，并可对代理与队列管理器之间的消息通信进行加密。

[第 560 页的『fteSetupCommands（创建 `command.properties` 文件）』](#)

fteSetupCommands 命令可创建 `command.properties` 文件。在您发出命令时，该属性文件指定与 IBM MQ 网络连接的队列管理器的详细信息。

agent.properties 文件

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

`agent.properties` 文件是由安装程序或 **fteCreateAgent**、**fteCreateWebAgent**、**fteCreateBridgeAgent** 或 **fteCreateCDAgent** 命令创建的。您可以将任意这些命令与 **-f** 标志一起使用，以更改基本代理队列管理器属性和那些与创建的代理类型关联的高级代理属性。要更改或添加高级代理属性，您必须在文本编辑器中编辑此文件。

代理程序的 `agent.properties` 文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中。

如果更改 `agent.properties` 文件，那么必须重新启动代理以获取更改。

对于 IBM WebSphere MQ V7.5 或更高版本，您可以在表示文件或目录位置的某些 IBM MQ Managed File Transfer 属性中使用环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化，例如哪个用户在运行进程。有关更多信息，请参阅第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』。

基本代理程序属性

每个 `agent.properties` 文件都包含以下基本属性。

表 49: 基本代理程序属性		
属性名	描述	缺省值
agentName	代理的名称。代理的名称必须符合 WebSphere MQ 对象命名约定。有关更多信息，请参阅第 699 页的『IBM MQ Managed File Transfer 的对象命名约定』。	没有缺省值
agentDesc	代理的描述 - 如果您选择创建描述。	没有缺省值
agentQMgr	代理队列管理器名称。	没有缺省值
agentQMgrHost	代理队列管理器的主机名或 IP 地址。	没有缺省值
agentQMgrPort	与代理队列管理器建立的客户机连接所使用的端口号。	1414
agentQMgrChannel	用于与代理队列管理器连接的 SVRCONN 通道名称。	SYSTEM.DEF.SVRCONN
agentType	代理类型： - 标准非网桥代理 (STANDARD) - 协议网桥代理 (BRIDGE) - Connect:Direct 网桥代理 (CD_BRIDGE) - Web 网关代理 (WEB_GATEWAY) - IBM Integration Bus 使用的嵌入式代理 (EMBEDDED) - Sterling File Gateway 嵌入式代理 (SFG)	标准

如果没有为 `agentQMgrHost` 属性指定值，那么缺省情况下使用绑定方式。

如果为 `agentQMgrHost` 属性指定值，但没有为 `agentQMgrPort` 和 `agentQMgrChannel` 属性指定值，那么缺省情况下使用端口号 1414 和通道 `SYSTEM.DEF.SVRCONN`。

高级代理属性

IBM MQ Managed File Transfer 还提供帮助您配置代理的更多高级代理属性。如果您要使用以下任意属性，请手动编辑 `agent.properties` 文件以添加所需的高级属性。在 Windows 上指定文件路径时，请确保将

分隔符反斜杠 (\) 输入为双反斜杠 (\\)，即转义反斜杠 (\)。或者，可以使用单个正斜杠 (/) 字符作为分隔符。有关 Java 属性文件中的字符转义的更多信息，请参阅 Oracle 文档 [Javadoc for the Properties class](#)。

- [代理大小属性](#)
- [代码页属性](#)
- [命令属性](#)
- [连接属性](#)
- [Connect:Direct 网桥属性](#)
- [文件到消息和消息到文件代理属性](#)
- [一般代理属性](#)
- [输入/输出属性](#)
- [多通道支持属性](#)
- [多实例属性](#)
- [进程控制器属性](#)
- [协议网桥属性](#)
- [队列属性](#)
- [资源监视属性](#)
- [根目录属性](#)
- [调度程序属性](#)
- [安全属性](#)
- [超时属性](#)
- [跟踪和记录属性](#)
- [传输限制属性](#)
- [用户出口例程属性](#)
- [WebSphere MQ 客户机压缩属性](#)
- [特定于 z/OS 的属性](#)
- [其他属性](#)

表 50: 高级代理属性		
属性名	描述	缺省值
代理大小属性:		
agentCheckpointInterval	为了恢复目的而采用检查点的间隔（以完整数据帧为单位）。这是高级属性，对于大多数 IBM MQ Managed File Transfer 配置而言，无需修改其值。 如果存在导致传输进入恢复状态的问题，那么传输只可以恢复到检查点边界。因此，该值越大（具有较大的 agentChunkSize、agentWindowSize 和 agentFrameSize 值），代理恢复传输所需的时间就越长。对于传输很少进入恢复状态的可靠 IBM MQ Managed File Transfer 网络，增加该值可能很有益，这样可提高整体性能。	1
agentChunkSize	针对文件数据传输的每个传输块的大小。因此，指示在源和目标代理之间传输的最大 WebSphere MQ 消息大小。这是高级属性，对于大多数 IBM MQ Managed File Transfer 配置而言，无需修改其值。 该值是通过源代理和目标代理协商而定，使用两个值中较大一个。如果要更改此属性的值，请在源代理和目标代理处更改该值。 agentChunkSize 是一个整数值。例如：agentChunkSize=10240 将块大小设置为 10 KB。	262144 字节（等于 256KB）

表 50: 高级代理属性 (继续)		
属性名	描述	缺省值
agentFrameSize	传输帧的窗口数。这是高级属性，对于大多数 IBM MQ Managed File Transfer 配置而言，无需修改其值。 对于具有较长等待时间的网络，增加此值可以提高整体性能，因为这会使代理同时具有更多活动消息块。 该属性的值乘以 agentWindowSize 再乘以 agentChunkSize，指示每个传输的代理内存使用量上限。例如，针对每个传输，262144 字节块 x 10 x 5 = 12.5MB。 注：如果单个传输中传输的文件大小小于 12.5MB，那么增加该属性值对传输性能无影响。	5
agentWindowSize	每个窗口的块数。这是高级属性，对于大多数 IBM MQ Managed File Transfer 配置而言，无需修改其值。 对于具有较长等待时间的网络，增加此值可以提高整体性能。因为这会使代理同时具有更多活动消息块，并会降低将确认消息发送回源代理的频率。 该属性的值乘以 agentFrameSize 再乘以 agentChunkSize，指示每个传输的代理内存使用量上限，并指示目标代理命令队列上的 WebSphere MQ 消息数据量上限。例如，针对每个传输，262144 字节块 x 10 x 5 = 上限 12.5MB。 注：如果单个传输中传输的文件大小小于 12.5MB，那么增加该属性值对传输性能无影响。	10
代码页属性：		
agentCcsid	代理连接其代理队列管理器所使用的代码页。如果为 agentCcsid 指定值，那么还必须为 agentCcsidName 指定值。有关如何查看 JVM 的已知代码页的信息，请参阅 fteCreateBridgeAgent 命令中的 -hsc 参数。	1208
agentCcsidName	agentCcsid 的 Java 表示。如果为 agentCcsidName 指定值，那么还必须为 agentCcsid 指定值。	UTF8
命令属性：		
maxCommandHandlerThreads	控制用于传输命令消息初始解析和处理的线程数。当线程处于活动状态时，需要一个到队列管理器的连接，但在空闲时，线程会释放连接。	5
maxCommandOutput	用于命令输出存储的最大字节数。该属性适用于为受管调用指定的命令以及为受管传输指定的 preSource、postSource、preDestination 和 postDestination 命令。这会限制写入 SYSTEM.FTE 主题上传输日志的命令输出的长度。	10240
maxCommandRetries	代理允许重试命令的最大次数。该属性适用于为受管调用指定的命令以及为受管传输指定的 preSource、postSource、preDestination 和 postDestination 命令。	9
maxCommandWait	代理允许重试之间的最长等待时间（秒）。该属性适用于为受管调用指定的命令以及为受管传输指定的 preSource、postSource、preDestination 和 postDestination 命令。	60
immediateShutdownTimeout	对于代理的立即关闭，您可以使用该属性来指定在强制关闭之前，代理等待完成传输的最长时间（秒）。 注：请勿将该属性的值更改为小于缺省值（10 秒）。代理的立即关闭需要足够的时间来终止任何外部进程。如果该属性的值太低，那么进程可能仍保持运行。 如果为该属性指定值 0，那么代理会等待所有未完成的传输停止。如果为该属性指定的值无效，那么将使用缺省值。	10
连接属性：		
javaLibraryPath	以绑定方式连接到队列管理器时，IBM MQ Managed File Transfer 必须有权访问 WebSphere MQ Java 绑定库。缺省情况下，IBM MQ Managed File Transfer 会在 WebSphere MQ 定义的缺省位置中查找绑定库。如果绑定库位于其他位置，那么请使用该属性指定绑定库的位置。	None
Connect:Direct 网桥属性：		
cdNode	当您使用 Connect:Direct 网桥时所需的属性。 用于将消息从 Connect:Direct 网桥代理传输到目标 Connect:Direct 节点的 Connect:Direct 节点的名称。该节点属于 Connect:Direct 网桥，不是作为传输源或目标的远程节点。有关更多信息，请参阅第 288 页的『Connect:Direct 网桥』。	没有缺省值

表 50: 高级代理属性 (继续)		
属性名	描述	缺省值
cdNodeHost	用于将文件从 Connect:Direct 网桥代理传输到目标节点（Connect:Direct 网桥节点）的 Connect:Direct 节点的主机名或 IP 地址。 在大多数情况下，Connect:Direct 网桥节点与 Connect:Direct 网桥代理位于同一个系统上。在这些情况下，该属性的缺省值（本地系统的 IP 地址）正确。如果系统具有多个 IP 地址，或您的 Connect:Direct 网桥节点不在 Connect:Direct 网桥代理所在的系统上，并且它们的系统共享同一个文件系统，请使用该属性指定 Connect:Direct 网桥节点的正确主机名。 如果您未设置 cdNode 属性，那么会忽略该属性。	本地系统的主机名或 IP 地址
cdNodePort	客户机应用程序用于与其进行通信的 Connect:Direct 网桥节点的端口号。在 Connect:Direct 产品文档中，该端口称为 API 端口。 如果您未设置 cdNode 属性，那么会忽略该属性。	1363
cdTmpDir	将文件传输到目标 Connect:Direct 节点之前在运行 Connect:Direct 网桥代理的系统上临时存储文件的位置。 该属性指定了临时存储文件的目录的完整路径。例如，如果 cdTmpDir 设置为 /tmp，那么会将文件临时放置在 /tmp 目录中。 Connect:Direct 网桥代理和 Connect:Direct 网桥节点必须能够使用相同路径名访问该参数指定的目录。请在计划安装 Connect:Direct 网桥时考虑这一点。如果可能，请在作为 Connect:Direct 网桥的组成部分的 Connect:Direct 节点所在的系统上创建该代理。如果代理和节点不在同一个系统上，那么该目录必须在共享文件系统上，并且可从两个系统使用相同路径名进行访问。有关受支持配置的更多信息，请参阅第 288 页的『Connect:Direct 网桥』。 如果您未设置 cdNode 属性，那么会忽略该属性。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。	<code>value_of_java.io.tmpdir /cdbridge-agentName</code> 在 Windows 上: <code>value_of_java.io.tmpdir \\cdbridge-agentName</code>
cdTrace	代理是否跟踪在 Connect:Direct 网桥代理与其 Connect:Direct 节点之间发送的数据。该属性的值可以为 true 或 false。	false
cdMaxConnectionRetries	传输失败之前，针对尚未进行成功连接的文件传输的最大 Connect:Direct 连接尝试次数。	-1（无数次尝试）
cdMaxPartialWorkConnectionRetries	传输失败之前，针对先前连接尝试已成功并且传输工作已完成的文件传输的最大 Connect:Direct 连接尝试次数。	-1（无数次尝试）
cdMaxWaitForProcessEndStats	Connect:Direct 进程终止之后，断定文件传输已失败之前，等待进程完成信息在 Connect:Direct 节点统计信息内变为可用的最长时间（毫秒）。通常，信息会立即可用，但在某些失败情况下，不会发布信息。在这些情况下，等待由该属性指定的时间段后文件传输失败。	60000
cdAppName	Connect:Direct 网桥代理用于连接到属于网桥的 Connect:Direct 节点的应用程序名称。	IBM MQ Managed File Transfer <i>current version</i> ，其中 <i>current version</i> 为产品的版本号。
cdNodeLocalPortRange	用于 Connect:Direct 网桥代理和属于网桥的 Connect:Direct 节点之间套接字连接的本地端口范围。该值的格式为以逗号分隔的值列表或范围列表。 缺省情况下，操作系统选择本地端口号。	None
cdNodeProtocol	Connect:Direct 网桥代理用于连接到属于网桥的 Connect:Direct 节点的协议。有效值包括： • TCPIP • SSL • TLS	TCPIP
cdNodeKeystore	Connect:Direct 网桥代理和属于该网桥的 Connect:Direct 节点之间安全通信所使用的密钥库的路径。 如果未将 cdNodeProtocol 属性设置为 SSL 或 TLS，那么会忽略该属性。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。	None
cdNodeKeystoreType	cdNodeKeystore 属性指定的密钥库的文件格式。以下值有效：jks 和 pkcs12。 如果未将 cdNodeProtocol 属性设置为 SSL 或 TLS，那么会忽略该属性。	jks
cdNodeKeystoreCredentialsFile	包含 cdNodeKeystore 凭证的文件的完整路径。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。	此属性的缺省值为 %HOMEDRIVE%\%HOMEPATH%\mqmftcredentials.xml (在 Windows 上) 和 \$HOME/MQMFTCredentials.xml (在其他平台上)。

表 50: 高级代理属性 (继续)		
属性名	描述	缺省值
cdNodeTruststore	Connect:Direct 网桥代理和属于该网桥的 Connect:Direct 节点之间安全通信所使用的信任库的路径。 如果未将 cdNodeProtocol 属性设置为 SSL 或 TLS, 那么会忽略该属性。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。	None
cdNodeTruststoreType	cdNodeTruststore 属性指定的信任库的文件格式。以下值有效: jks 和 pkcs12。 如果未将 cdNodeProtocol 属性设置为 SSL 或 TLS, 那么会忽略该属性。	jks
cdNodeTruststoreCredentialsFile	包含 cdNodeTruststore 凭证的文件的名称。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。	此属性的缺省值为 %HOMEDRIVE%\%HOMEPATH%\mqmftcredentials.xml (在 Windows 上) 和 \$HOME/MQMFTCredentials.xml (在其他平台上)。
logCDProcess	记录在 output0.log 文件的代理事件日志中的 Connect:Direct 处理记录的级别。该属性的值可以为 None 或 Failures 或 All。	None
文件到消息和消息到文件代理属性:		
deleteTmpFileAfterRenameFailure	将该属性的值设置为 false 可确保在重命名操作失败的情况下不会从目标中删除临时文件。在此情况下, 已传输的数据会作为临时 (.part) 文件保留在目标中。稍后您可以手动重命名此文件。缺省情况下, 该属性的值为 true。该属性同时应用于“消息到文件”传输和“文件到文件”传输。 该属性不适用于 IBM i。	true
enableQueueInputOutput	缺省情况下, 在传输过程中, 代理不能读取源队列中的数据或将数据写入目标队列。将该值设置为 true, 可使代理执行文件到消息和“消息到文件”传输。该属性的值可以为 true 或 false。	false
enableSystemQueueInputOutput	指定代理是否可以读取或写入 WebSphere MQ 系统队列。系统队列以限定词 SYSTEM 为前缀。 注: 系统队列由 WebSphere MQ、IBM MQ Managed File Transfer 和其他应用程序用于传输重要信息。更改该属性, 可使代理访问这些队列。如果启用该属性, 那么请使用用户沙箱来限制代理可以访问的队列。	false
enableClusterQueueInputOutput	指定代理是否可以读取或写入 WebSphere MQ 集群队列。 注: 除 enableQueueInputOutput 属性外, 还必须指定 enableClusterQueueInputOutput 代理属性。	false
maxDelimiterMatchLength	Java 正则表达式可以匹配的最大字符数, 该表达式用于将文本文件作为文件到消息传输的一部分拆分为多条消息。	5
maxInputOutputMessageLength	由代理从源队列读取或写入目标队列的消息的最大长度 (字节)。传输中源代理的 maxInputOutputMessageLength 属性确定可以从源队列消息读取的字节数。传输中目标代理的 maxInputOutputMessageLength 属性确定可写入到目标队列消息的字节数。如果消息的长度超出该属性的值, 那么传输失败并产生错误。该属性不会影响 IBM MQ Managed File Transfer 内部队列。有关更改该属性的信息, 请参阅第 394 页的『用于设置与消息大小关联的 IBM MQ 属性和 IBM MQ Managed File Transfer 属性的指南』。	1048576
monitorGroupRetryLimit	队列中仍存在消息组时, 监视器再次触发“消息到文件”传输的最大次数。触发“消息到文件”传输的次数由组中第一条消息的 MQMD 回退计数确定。 如果重新启动代理, 那么即使触发传输的次数已超出 monitorGroupRetryLimit 的值, 监视器仍会再次触发传输。如果该行为导致触发传输的次数超出 monitorGroupRetryLimit 的值, 那么代理会向其事件日志中写入一个错误。 如果对该属性指定值 -1, 那么监视器会不限制次数地再次触发传输, 直至不满足触发条件。	10
一般代理属性:		
agentStatusPublishRateLimit	由于文件传输状态更改, 代理重新发布其状态的最大速率 (秒)。 如果该属性值设置太小, 那么 WebSphere MQ 网络的性能可能会受负面影响。	30
agentStatusPublishRateMin	代理发布其状态的最小速率 (秒)。该值必须大于或等于 agentStatusPublishRateLimit 属性的值。	300

表 50: 高级代理属性 (继续)		
属性名	描述	缺省值
enableMemoryAllocationChecking	确定 MQMFT 代理是否执行以下检查: 检查在启动传输之前是否有足够的内存可用于运行传输。源和目标代理上都会执行该检查。如果可用内存不足, 传输将被置于恢复状态, 以防止代理因内存不足错误而失败。 在计算传输所需的内存时, 将使用传输所需的最大内存。所以, 该值可能大于传输使用的实际内存。因此, 如果 enableMemoryAllocationChecking 属性设置为 true, 可运行的并行传输的数量可能会下降。建议仅在遇到因内存不足错误而导致 MQMFT 失败的问题时才将该属性设置为 true。可能使用大量内存的传输包括消息大小较大的“文件到消息”传输和“消息到文件”传输。	false
enableDetailedReplyMessages	将此属性设置为 true 使受管传输请求回复包含有关传输文件的详细信息。详细信息和格式与进度消息中发布到传输日志的相同, 即 <transferSet> 元素。有关更多信息, 请参阅第 662 页的『文件传输日志消息格式』。 仅当受管传输请求指定详细回复信息为必需时, 才会包含详细回复信息。To specify this requirement, set the detailed attribute of the <reply> element of the managedTransfer XML request message sent to the source agent. 有关更多信息, 请参阅第 842 页的『文件传输请求消息格式』。 可以为每个传输请求生成多条回复消息。该数目等于传输的传输日志进度消息数目加 1 (其中, 第一条回复消息为简单 ACK 回复)。除 ACK 回复消息外, 所有消息中都包含详细信息, 但总体传输结果仅包含在最后一详细回复消息中。	true
enableUserMetadataOptions	确定是否可以在新传输请求中使用已知的用户定义元数据键以提供更多传输选项。这些已知键始终以前缀 com.ibm.wmqfte. 开头。因此, 将 enableUserMetadataOptions 属性设置为 true 时, 不支持将使用此前缀的键用于用户定义的用途。将 enableUserMetadataOptions 属性设置为 true 时, 当前支持如下键: - com.ibm.wmqfte.insertRecordLineSeparator - com.ibm.wmqfte.newRecordOnLineSeparator - com.ibm.wmqfte.convertLineSeparators 有关这些键含义的信息, 请参阅第 497 页的『fteCreateTransfer (创建新的文件传输)』。 该属性的值可以为 true 或 false。	false
  failTransferOnFirstFailure	允许将代理配置为受管传输中的一个传输项失败, 受管传输就失败。 要启用此功能, 必须对源代理和目标代理应用 APAR IT03450, 并且必须在源代理的 agent.properties 文件中将 failTransferOnFirstFailure 属性设置为 true。在目标代理上将该属性设置为 true 是可选的。 当 failTransferOnFirstFailure 属性设置为 true 时, 代理将正常开始处理受管传输请求。但是, 一个传输项失败, 就会将受管传输标记为失败, 不会处理更多传输项。将采用以下方式处理在受管传输失败前成功处理的传输项: - 采用这些传输项的源处置。例如, 如果已将传输项的源处置设置为 delete, 那么将删除源文件。 - 编写的目标文件仍在目标文件系统中, 不会将其删除。 如果未将 failTransferOnFirstFailure 属性设置为 true、受管文件传输包含多个文件并且其中一个文件传输失败 (例如, 由于目标文件已存在并且覆盖属性设置为 error), 那么源代理将继续并尝试传输请求中的任何剩余文件。	false
itemsPerProgressMessage	代理发布其下一条进度日志消息之前传输的文件数。使用该属性控制在传输期间将进度日志消息发布到协调队列管理器的速率。 该属性可以设置的最大值为 1000。 注: 进度消息包含有关发布上条进度消息之后传输的每个文件的信息。增加此值将增加进度消息的大小, 有可能会影响性能。	50
maxInlineFileSize	对于单个文件到文件或“文件到消息”传输, 可以自动包含在初始传输请求消息中的最大文件大小 (字节)。 您可以使用该属性提高传输速度, 但如果将文件大小设置为过大的值, 这可能会降低性能。该属性的建议初始大小是 100KB, 但建议彻底测试不同值, 直到找到最适合系统的文件大小。	0
输入/输出属性:		
doNotUseTempOutputFile	缺省情况下, 代理将写入目标上的临时文件中, 并在文件传输完成后, 将该临时文件重命名为所需的文件名。将该值设置为 true 会使代理对最终的目标文件进行直接写入。 在 IBM 4690 上, 请勿将该属性设置为 true: 在 4690 OS 上不支持直接写入到目标上的最终文件。 在 z/OS 系统上, 不会将该行为应用于连续数据集, 但会应用于 PDS 数据集成员。 该属性的用于传输的值由目标代理定义。	false

表 50: 高级代理属性 (继续)		
属性名	描述	缺省值
enableMandatoryLocking	访问正常文件时, IBM MQ Managed File Transfer 会获取用于读取的共享锁定和用于写入的互斥锁定。在 UNIX 类型的平台上, 跨进程实现文件锁定。但是, 在 Windows 上, 只是建议文件锁定。将该属性设置为 true 时, IBM MQ Managed File Transfer 会强制文件锁定。在 Windows 上, 这意味着如果另一个应用程序打开一个文件, 那么不会触发监视该文件, 直至文件关闭。涉及该文件的 IBM MQ Managed File Transfer 传输失败。对于 UNIX 类平台, 设置该属性不会有任何影响。 该属性的值可以为 true 或 false。	false
ioIdleThreadTimeout	线程关闭之前, 文件系统输入/输出线程保持空闲的时间 (毫秒)。	10000
ioQueueDepth	排队的最大输入/输出请求数。	10
ioThreadPoolSize	可用的最大文件系统输入/输出线程数。通常, 每个传输都使用它自己的文件系统输入/输出线程, 但是如果并行传输超过此限制, 那么会在传输之间共享文件系统输入/输出线程。 如果您认为您可能会经常具有比 ioThreadPoolSize 值更多的并行传输, 那么您通过增加该值会看到改进, 以便每个传输都有它自己的文件系统输入/输出线程。	10
textReplacementCharacterSequence	对于文本方式传输, 如果不能将任何数据字节从源代码页转换到目标代码页, 那么缺省行为为文件传输失败。 通过插入指定的字符值, 将该属性设置为允许传输成功完成。该属性值是单字符。通常, 问号 (?) 用于表示任何不可映射的字符。例如, 使用以下格式: textReplacementCharacterSequence=?, 其中问号 (?) 是替换字符。不能使用空格字符作为替换字符。	None
多通道支持:		
agentMultipleChannelsEnabled	设置该属性为 true, 使 IBM MQ Managed File Transfer 代理通过多个 IBM MQ 通道发送传输数据消息。在某些情况下, 设置该属性可提高性能。但是, 只有当存在显著的性能优势时, 才应启用多通道支持。在多个通道之间只发送置于 SYSTEM.FTE.DATA.destinationAgentName 队列的消息。对所有其他消息的行为保持不变。 将该属性设置为 true 时, 您还必须在以下某个主题中完成 IBM MQ 配置步骤才能启用多通道支持: - 第 604 页的『在集群中配置多个 IBM MQ 通道』 - 第 605 页的『在非集群配置中配置多个 IBM MQ 通道』 另外, 您还必须完成 IBM MQ Managed File Transfer 代理所需的标准 IBM MQ 配置步骤, 这在第 141 页的『为首次使用配置 IBM MQ Managed File Transfer』中进行了详细描述。 该属性的值可以为 true 或 false。	false
agentMessageBatchSize	配置多通道后, 源代理以循环法为每个通道之间的传输发送数据消息。该属性控制每次沿每个通道发送的消息数。	5
多实例队列管理器属性:		
agentQMGrStandby	针对通过 agentQMGr 定义的多实例代理队列管理器的备用实例, 用于客户机连接的主机名和端口号, 采用 WebSphere MQ CONNAME 格式。例如, host_name(port_number)	没有缺省值
进程控制器属性:		
agentQMGrRetryInterval	代理的进程控制器对队列管理器的可用性进行检查的时间间隔 (秒)。	30
maxRestartCount	在 maxRestartInterval 属性值所指定的时间间隔内可发生的最大重新启动次数。当超出该值时, 代理的进程控制器会停止重新启动代理, 而改为执行基于 maxRestartDelay 属性值的操作。	4
maxRestartInterval	代理的进程控制器度量代理重新启动的时间间隔 (秒)。如果该时间间隔内的重新启动次数超过 maxRestartCount 属性的值, 那么代理的进程控制器会停止重新启动代理。代理的进程控制器会改为执行基于 maxRestartDelay 属性值的操作。	120
maxRestartDelay	确定代理的进程控制器在代理重新启动的速率超过 maxRestartCount 和 maxRestartInterval 属性值时的行为。如果指定小于或等于零的值, 那么代理的进程控制器会停止。如果指定大于零的值, 那么这即是重置代理的进程控制器持有的重新启动历史记录信息并重新启动代理之前等待的秒数。	-1
processControllerPollingInterval	IBM 4690 代理控制器每次尝试检查其关联代理所提供状态的时间间隔 (秒)。该时间间隔从收到最后一条来自该代理的成功状态消息后开始。	30

表 50: 高级代理属性 (继续)		
属性名	描述	缺省值
processControllerPollingTimeout	IBM 4690 代理控制器在假定代理不再响应之前等待来自该代理的预期状态更新的时间（秒）。在 processControllerPollingInterval 属性定义的时间耗尽后，代理控制器将开始检查状态更新。如果在该属性定义的时间段内未收到状态更新，那么会假定代理无响应。	10
useProcessController	缺省情况下，IBM 4690 代理会尝试启动进程控制器以监视其执行。建议将该值设置为 true，但如果内存使用开销过高，可以将其设置为 false。代理将不会尝试启动进程控制器，但这可能会降低代理可靠性。	true
协议网桥属性：		
protocolBridgeCredentialConfiguration	该属性的值作为字符串传入由 protocolBridgeCredentialExitClasses 指定的出口类的 initialize() 方法。	null
protocolBridgeCredentialExitClasses	指定实现了协议网桥凭证用户出口例程的类的逗号分隔列表。要了解更多信息，请参阅第 283 页的『使用出口类映射文件服务器的凭证』。	没有缺省值。
protocolBridgeDataTimeout	协议网桥代理等待建立通往 FTP 服务器的数据连接或通过已经建立的连接接收来自 FTP 服务器的数据的超时时间（以毫秒为单位）。如果将此属性值设置为 0，协议网桥代理将会无限期等待。如果到达超时时间，则协议网桥代理将关闭通往 FTP 服务器的任何现有数据连接，并在恢复当前传输之前尝试建立新数据连接。如果建立新数据连接的尝试失败，那么当前传输也会失败。	0
protocolBridgeLogoutBeforeDisconnect	指定协议网桥代理在关闭 FTP 会话并断开连接之前是否会使用户从文件服务器注销。如果将此属性设置为 true，协议网桥代理将向文件服务器发出 FTP QUIT 命令。	false
protocolBridgePropertiesConfiguration	作为某个网桥属性传递到由 protocolBridgeServerPropertiesExitClasses 属性指定的出口类的 initialize() 方法。	没有缺省值
protocolBridgePropertiesExitClasses	指定实现了协议网桥服务器属性用户出口例程的类的逗号分隔列表。 有关更多信息，请参阅第 277 页的『使用出口类 (ProtocolBridgePropertiesExit2) 查找协议文件服务器属性』。	没有缺省值
队列属性：		
publicationMDUser	与协议队列管理器发布的已发送消息关联的 MQMD 用户标识。如果您未设置该属性，那么根据用于设置 MQMD 用户标识的 WebSphere MQ 规则设置 MQMD 用户标识。	没有缺省值
资源监视属性：		
monitorFilepathPlatformSeparator	指定是否在 \$FILEPATH 变量中使用特定于平台的路径分隔符。true 值表示使用特定于平台的路径分隔符。false 值表示在所有平台上都使用 UNIX 样式的正斜杠 (/) 路径分隔符。	true
monitorMaxResourcesInPoll	指定在每个轮询时间间隔内触发的最大资源监视数。例如，如果将监视模式指定为 *.txt，将轮询时间间隔指定为 10 秒并将 monitorMaxResourcesInPoll 属性设置为 10，那么 monitorMaxResourcesInPoll 属性会限制代理每个轮询时间间隔最多触发 10 个匹配项。在以后的轮询时间间隔内会触发 10 个限制以上的匹配资源。 另外，您可以将 monitorMaxResourcesInPoll 属性与 fteCreateMonitor 命令的匹配 -bs 参数（例如）结合使用，以限制每个轮询时间间隔只触发一个传输。 小于或等于零的值表示在轮询时间间隔内触发的监视资源数不受限制。	-1
monitorReportTriggerFail	指定是否将在监视器中检测到的环境和配置中的失败情况作为日志消息报告给 SYSTEM.FTE 主题。true 值记录消息。false 值不记录消息。	true
monitorReportTriggerNotSatisfied	指定不满足条件的触发器是否向包含详细信息的 SYSTEM.FTE 主题发送日志消息。true 值记录消息。false 值不记录消息。	false
monitorReportTriggerSatisfied	指定满足条件的触发器是否向包含详细信息的 SYSTEM.FTE 主题发送日志消息。true 值记录消息。false 值不记录消息。	false
monitorSilenceOnTriggerFailure	不再报告失败之前，资源监视器触发器的连续失败数。	5
monitorStopOnInternalFailure	资源监视器将其状态更改为停止之前，资源监视器的连续内部 FFDC 条件数。	10
根目录属性：		

表 50: 高级代理属性 (继续)

属性名	描述	缺省值
commandPath	指定可调用命令的路径集，可以通过使用一下方法之一： - Agent Ant call、filecopy 或 filemove 任务 - 在传递给代理的 XML 消息中，通过使用某个受支持的 IBM MQ Managed File Transfer 代理命令 XML 模式（例如，managedCall 或 managedTransfer）。 有关 commandPath 属性值的有效语法的信息，请参阅 第 443 页的『commandPath 属性』。 要点:  设置该属性时务必谨慎，因为能够向代理发送命令的远程客户机系统可以有效调用指定的某个 commandPath 中的任何命令。为此，缺省情况下，在指定 commandPath 时： - 代理在启动时会配置任何现有代理沙箱，从而将所有 commandPath 目录自动添加到拒绝访问传输的目录列表。 - 代理在启动时会更新任何现有用户沙箱，从而将所有 commandPath 目录（及其子目录）作为 <exclude> 元素添加到 <read> 和 <write> 元素中。 - 如果未将代理配置为使用代理沙箱或用户沙箱，那么在代理启动时会创建一个新的代理沙箱，其中将 commandPath 目录指定为拒绝的目录。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。 您可以将 addCommandPathToSandbox 属性设置为 false 以覆盖此缺省行为，从而兼容以下发行版： - IBM WebSphere MQ 文件传输版。 - IBM WebSphere MQ V7.5.0.1 受管文件传输 组件（或更低版本）。 - 未设置安装属性 enableFunctionalFixPack=7502 的安装上的 IBM WebSphere MQ V7.5.0.2 受管文件传输 组件（或更高版本）。 要点: 请注意，此覆盖实际上支持客户机将任何命令传输到代理系统并调用该命令，因此在使用时务必谨慎。	None - 不能调用任何命令
  addCommandPathToSandbox	指定是否应将 commandPath 属性指定的目录（及其所有子目录）添加到： - 适用于现有代理沙箱的拒绝的目录。 - 对于已定义的任何用户沙箱，<read> 和 <write> 元素的 <exclude> 元素。 - 新的代理沙箱，如果没有为代理配置一个代理沙箱或者一个或多个用户沙箱。 这可兼容以下发行版： - IBM WebSphere MQ 文件传输版。 - IBM WebSphere MQ V7.5.0.1 受管文件传输 组件（或更低版本）。 - 未设置安装属性 enableFunctionalFixPack=7502 的安装上的 IBM WebSphere MQ V7.5.0.2 受管文件传输 组件（或更高版本）。 有关更多信息，请参阅 第 443 页的『commandPath 属性』。	True
  additionalWildcardSandboxChecking	指定是否对使用用户或代理沙箱配置的代理的通配符传输进行额外检查，以便限制代理可以将文件传输到以及从中传输文件的位置。 将该属性设置为 true 时，会启用额外检查。如果传输请求尝试读取进行通配符文件匹配的已定义沙箱外部的位，那么传输将失败。如果一个传输请求中存在多个传输，并且其中一个请求因尝试读取沙箱外部的位而失败，那么整个传输也将失败。如果检查失败，将在错误消息中提供失败原因（请参阅 第 102 页的『针对通配符传输的额外检查』）。 如果忽略此属性或者将其设置为 false，那么将不会对通配符传输进行额外检查。	None
sandboxRoot	指定使用沙箱时包含和排除的根路径集。请参阅使用沙箱以获取有关该功能的信息。 使用特定于平台的路径分隔符分隔路径。路径以惊叹号 (!) 字符为前缀以表示从沙箱排除路径。当您要包含的根路径下排除子目录时，该功能很有用。 sandboxRoot 属性在协议网桥代理上不受支持。 您不能同时指定 sandboxRoot 属性和 userSandboxes 属性。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。	None - 没有沙箱

表 50: 高级代理属性 (继续)		
属性名	描述	缺省值
transferRoot	为代理指定的相对路径的缺省根目录。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。 请参阅第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』, 以了解更多信息。 在 IBM 4690 上, 没有针对该属性的缺省目录。因此, 如果要使用相对路径, 必须将该属性的值设置为目录路径。如果未设置该属性, 那么与指定源或目标相对路径的 4690 代理之间的文件传输将失败, 并返回一条错误消息。	启动代理进程的用户的根目录。 没有针对 IBM 4690 的缺省目录。
transferRootHLQ	为代理指定的非标准数据集的缺省 HLQ (用户标识)	启动代理进程的用户的用户名。
userSandboxes	根据请求传输的用户的 MQMD 用户名, 限制可以传输文件的文件系统区域。有关更多信息, 请参阅第 100 页的『使用用户沙箱』。 userSandboxes 属性在协议网桥代理上不受支持。 您不能同时指定 sandboxRoot 属性和 userSandboxes 属性。	false
调度程序属性:		
maxSchedulerRunDelay	代理等待检查已调度传输的最大时间间隔 (分钟)。指定一个正整数以启用该属性。有关您为什么要使用该属性的更多信息, 请参阅第 387 页的『在已安排传输未运行或被延迟时应执行什么操作』。 因为代理可能在调度的传输将要运行时从其命令队列读取命令, 因此在启动调度的传输之前, 可能存在额外的延迟。在此情况下, 调度程序将在该命令完成后立即运行。	-1
安全属性:		
agentQMGrAuthenticationCredentialsFile	包含 MQ 连接凭证的文件的路径。	此属性的缺省值为 %HOMEDRIVE%\%HOMEPATH%\mqmftcredentials.xml (在 Windows 上) 和 \$HOME/MQMFTCredentials.xml (在其他平台上)。
authorityChecking	指定是否启用第 438 页的『用户的 IBM MQ Managed File Transfer 操作权限』中描述的安全功能。  从 IBM MQ 8.0.0 修订包 8 开始, inquire 在所有代理权限队列上都是必需的许可权。	false
logAuthorityChecks	output0.log 文件的代理事件日志中记录的权限检查记录的级别。该属性的值可以为 None 或 Failures 或 All。	None
userIdForClientConnect	经客户机连接流至 WebSphere MQ 的用户标识。如果指定了 java, 那么 JVM 报告的用户名会作为 WebSphere MQ 连接请求的一部分流动。该属性的值可以为 None 或 java。	None
超时属性:		
maxTransferNegotiationTime	传输等待目标代理完成协商的最长时间 (毫秒)。如果在这期间未完成协商, 那么传输会进入再同步状态, 并允许其他传输 (如果可用) 运行。 在源或目标代理负载过重的情况下, 可能是针对代理的缺省值太低以致无法足够快速地响应协商请求。当源代理定义了资源监视器或其资源监视器正在监视包含大量文件的目录时, 最有可能发生此情况。但是, 在向代理提交大量传输请求时, 也可能发生此情况。在此类情况下, 可能需要将该属性的值增加至 200,000 或更大值。	30 000
recoverableTransferRetryInterval	检测可恢复传输错误和尝试恢复传输之间等待的时间 (毫秒)。	60 000
senderTransferRetryInterval	由于目标已运行最大数量的传输而重试拒绝的传输之前等待的时间 (毫秒)。最小值为 1000。	30 000
transferAckTimeout	发出重试之前, 传输等待来自另一端的应答或数据的超时时间 (毫秒)。这是高级属性, 对于大多数 IBM MQ Managed File Transfer 配置而言, 无需修改其值。 只要收到完整的数据窗口, 就向发送代理发送来自接收代理的应答。对于带宽受到限制或不可靠的网络以及大的 agentWindowSize 和 agentChunkSize 设置, 缺省值可能不够长。这可能会导致代理之间不必要的数据重新传输。因此, 增加该值可能是有益的, 可能会降低由于网络速度缓慢而使传输进入恢复方式的可能性。	60 000
transferAckTimeoutRetries	代理放弃并将传输移至恢复状态之前, 在没有响应的情况下, 传输的最大应答重试次数。	5

表 50: 高级代理属性 (继续)		
属性名	描述	缺省值
xmlConfigReloadInterval	运行时期间代理重新载入 XML 配置文件之间的时间间隔 (秒)。为防止代理在运行时期间重新载入 XML 配置文件, 请将该属性设置为 -1。该属性将影响以下 XML 配置文件: - ConnectDirectCredentials.xml - ConnectDirectNodeProperties.xml - ConnectDirectProcessDefinitions.xml - ProtocolBridgeCredentials.xml - ProtocolBridgeProperties.xml - UserSandboxes.xml	30
跟踪和记录属性:		
javaCoreTriggerFile	代理监视的文件位置的完整路径。如果文件存在于指定位置, 那么代理程序启动将触发 Javacore。启动代理程序后, 如果在此位置更新文件, 那么代理程序将再次触发 Javacore 文件。 单独的线程会每 30 秒轮询该文件以检查是否已创建或更新文件。如果自上次轮询之后已创建或更新文件, 那么代理将在以下某个目录中生成 Javacore 文件: - UNIX: <code>MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name</code> - Linux: <code>MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name</code> - Windows: <code>MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\agents\agent_name</code> - IBM 4690: <code>f:\adxetc\java\core</code> 指定该属性后, 代理会在启动时输出以下消息: <pre>BFGAG0092I The <insert_0> file will be used to request JVM diagnostic information.</pre> 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。	None
跟踪	要在代理启动时启用跟踪的情况下运行代理时的跟踪规范。跟踪规范是以逗号分隔的类、等于字符以及跟踪级别列表。例如, <code>com.ibm.wmqfte.agent.Agent,com.ibm.wmqfte.commandhandler=all</code>。您可以在冒号分隔的列表中指定多个跟踪规范。例如, <code>com.ibm.wmqfte.agent.Agent=all:com.ibm.wmqfte.commandhandler=moderate</code>。	None
outputLogFiles	保存的 output.log 文件的总数。该值适用于代理的进程控制器以及代理自身。	5
outputLogSize	将输出合并到下一个文件之前, 每个 output.log 文件的最大大小 (MB)。该值适用于代理的进程控制器以及代理自身。	1
outputLogEncoding	写入 output.log 文件时, 代理使用的字符编码。	正在运行代理的平台的缺省字符编码。
traceFiles	保存的跟踪文件的总数。该值也适用于代理的进程控制器和代理自身。	5
traceSize	将跟踪合并到下一个文件之前, 每个跟踪文件的最大大小 (MB)。该值适用于代理的进程控制器以及代理自身。	20
traceMaxBytes	对在跟踪文件中输出的消息数据数量的限制。	4096 字节
logTransferRecovery	如果将该属性的值设置为 true, 每当传输进入恢复状态时, 都会向 output0.log 文件中代理的事件日志报告诊断事件。	在 IBM MQ 8.0.0 修订包 3 之前, 缺省值为 false。  从 IBM MQ 8.0.0 修订包 3 开始, 缺省值为 true。
logCapture	捕获提交给该代理的传输请求消息和代理发布到协调队列管理器的日志消息。在调试传输问题时, 捕获的这些消息可能会很有帮助。捕获的消息存储在代理日志目录中名为 <code>capture?.log</code> 的文件。? 是数字值。包含数字 0 的文件保存最新捕获的消息。	false
logCaptureFileSize	定义捕获文件的最大大小 (兆字节)。	10
logCaptureFiles	定义在废弃最旧的文件之前保留的最大捕获文件数。	10
logCaptureFilter	代理程序用于匹配消息主题名称的 Java 正则表达式。仅捕获那些与正则表达式匹配的消息。	.* (match all)

表 50: 高级代理属性 (继续)		
属性名	描述	缺省值
传输限制属性:		
maxDestinationTransfers	目标代理在任意时间点处理的最大并行传输数。向代理提交的每个传输请求都会针对该总数计数，而与满足该请求的传输文件数无关。这意味着传输一个文件的传输请求与传输 10 个文件的传输请求的计数方法相同。 如果目标代理达到 maxDestinationTransfers 属性指定的限制，那么此代理会将传输排队。 如果代理属性值 maxSourceTransfers + maxDestinationTransfers + maxQueuedTransfers 的总和 超出状态存储队列 (SYSTEM.FTE.STATE.agent name) 的 MAXDEPTH 设置值，那么不启动代理。	25 (对于 Connect:Direct 网桥代理之外的所有代理) 5 (对于 Connect:Direct 网桥代理)
maxFilesForTransfer	单个受管传输允许的最大传输项数。如果受管传输包含比 maxFilesForTransfer 的值更多的项，那么受管传输将失败并且不会处理任何传输项。 设置该属性可防止您由于错误的传输请求而意外传输太多文件，例如，如果用户无意间在 UNIX 系统上指定根目录 / 的传输。	5000
maxSourceTransfers	源代理在任何时间点处理的最大并行传输数。向代理提交的每个传输请求都会针对该总数计数，而与满足该请求的传输文件数无关。这意味着传输一个文件的传输请求与传输 10 个文件的传输请求的计数方法相同。 如果目标代理达到 maxSourceTransfers 属性指定的限制，那么源代理会将传输排队。 如果代理属性值 maxSourceTransfers + maxDestinationTransfers + maxQueuedTransfers 的总和 超出状态存储队列 (SYSTEM.FTE.STATE.agent name) 的 MAXDEPTH 设置值，那么不启动代理。	25 (对于 Connect:Direct 网桥代理之外的所有代理) 5 (对于 Connect:Direct 网桥代理)
maxQueuedTransfers	在代理拒绝新的传输请求之前，源代理可排队的暂挂传输的最大数目。您可以设置该属性，以便不管满足还是超出 maxDestinationTransfers 和 maxSourceTransfers 的限制，都接受您此时提交的任何新传输请求，将其排队，并在稍后执行。 排队的传输请求的处理顺序取决于其优先级及排队的时间长短。首先，会选择旧优先级和高优先级暂挂的传输。队列中具有低优先级但排队时间较长的传输优先于较新且具有较高优先级的传输。 如果代理属性值 maxSourceTransfers + maxDestinationTransfers + maxQueuedTransfers 的总和 超出状态存储队列 (SYSTEM.FTE.STATE.agent name) 的 MAXDEPTH 设置值，那么不启动代理。	1000
用户出口例程属性:		
agentForceConsistentPathDelimiters	将提供给传输出口的源文件和目标文件信息中的路径定界符强制为 UNIX 样式: 正斜杠 (/)。有效选项为 true 和 false。	false
destinationTransferEndExitClasses	指定一个实现目标传输最终用户出口例程的以逗号分隔的类列表。	没有缺省值
destinationTransferStartExitClasses	指定一个实现目标传输开始用户出口例程的以逗号分隔的类列表。	没有缺省值
exitClassPath	指定一个特定于平台、字符分隔的目录列表，充当用户出口例程的类路径。在此类路径中的任何条目之前搜索代理程序出口目录。	代理的 exits 目录
exitNativeLibraryPath	指定一个特定于平台、字符分隔的目录列表，充当用户出口例程的本机库路径。	代理的 exits 目录
ioMaxRecordLength	面向记录的文件支持的最大记录长度（字节）。IBM MQ Managed File Transfer 支持写入具有任意记录长度的面向记录文件。但是，较大的记录长度可能会导致内存不足错误，因此为了避免这些错误，缺省情况下最大记录长度限制为 64 K。从面向记录的文件中读取时，整个记录必须适合单个传输块，因此记录长度还受到传输块大小的限制。该属性仅用于 I/O 用户出口的面向记录文件。	64KB
monitorExitClasses	指定实现了监视器出口例程的类的逗号分隔列表。要了解更多信息，请参阅第 974 页的『资源监视器用户出口』。	没有缺省值
protocolBridgeCredentialExitClasses	指定实现了协议网桥凭证用户出口例程的类的逗号分隔列表。要了解更多信息，请参阅第 283 页的『使用出口类映射文件服务器的凭证』。	没有缺省值。
sourceTransferEndExitClasses	指定一个实现源传输结束出口例程的以逗号分隔的类列表。	没有缺省值
sourceTransferStartExitClasses	指定一个实现源传输开始出口例程的以逗号分隔的类列表。	没有缺省值
IOExitClasses	指定实现了 I/O 用户出口例程的类的逗号分隔列表。仅列出实现了 IOExit 接口的类，即不列出实现了其他 I/O 用户出口接口的类，例如，IOExitResourcePath 和 IOExitChannel。有关更多信息，请参阅第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』。	没有缺省值。

表 50: 高级代理属性 (继续)		
属性名	描述	缺省值
webGatewayName	必需。部署的 Web 网关的名称。 Web 网关的名称不区分大小写，但必须遵守 WebSphere MQ 对象的命名约定。有关更多信息，请参阅第 699 页的『IBM MQ Managed File Transfer 的对象命名约定』。	没有缺省值。
WebSphere MQ 客户机压缩：		
agentDataCompression	仅针对客户机连接支持此属性。 以逗号分隔的与远程 IBM MQ 服务器协商的文件数据传输的压缩类型列表。您可以在以下主题中查找有关这些压缩类型的信息：消息数据压缩列表。 将检查这些值的有效性，然后按外观顺序将其作为属性传递到代理程序客户机通道。然后，IBM MQ 客户机处理该客户机通道和远程服务器通道之间的协商，以寻找两个通道上压缩属性之间匹配的最低共同特性。如果未发现匹配项，那么始终选择 MQCOMPRESS_NONE。	MQCOMPRESS_NONE
agentHeaderCompression	仅针对客户机连接支持此属性。 以逗号分隔的与远程 IBM MQ 服务器协商的头数据传输的压缩类型列表。接受的值为 MQCOMPRESS_NONE 或 MQCOMPRESS_SYSTEM。您可以在以下主题中查找有关这些压缩类型的信息：消息头压缩列表。 将检查这些值的有效性，然后按外观顺序将其作为属性传递到代理程序客户机通道。然后，IBM MQ 客户机处理该客户机通道和远程服务器通道之间的协商，以寻找两个通道上压缩属性之间匹配的最低共同特性。如果未发现匹配项，那么始终选择 MQCOMPRESS_NONE。	MQCOMPRESS_NONE
特定于 z/OS：		
  adminGroup	安全管理器组。该组成员可以： - 使用 fteStartAgent 命令启动代理。 - 使用 fteStopAgent 命令停止代理。 - 使用 fteSetAgentTraceLevel 命令启用或禁用代理的跟踪。 - 通过运行指定有 -d 参数的 fteShowAgentDetails 命令来显示本地代理的详细信息。 定义安全管理器组（例如， MFTADMIN ），然后将已启动任务用户标识和管理员 TSO 标识添加到该组。编辑代理属性文件并将 adminGroup 属性设置为该安全管理器组的名称。 <pre>adminGroup=MFTADMIN</pre>	None
bpxwdynAllocAdditionalOptions	IBM MQ Managed File Transfer 使用 BPXWDYN 文本接口创建和打开 z/OS 数据集。当 BPXWDYN 用于数据集分配时，缺省情况下，IBM MQ Managed File Transfer 可能确保安装数据集设备（不是基于磁盘的数据集所需，而是磁带数据集所需）。由于某些环境可能不支持选项，因此请使用该属性更改此行为。另外，传输到数据集时，还可以在命令行指定 BPXWDYN 的选项；这些选项是对该属性指定的那些选项的补充。 使用 agent.properties 文件中的 bpxwdynAllocAdditionalOptions 属性时，不得指定某些 BPXWDYN 选项。有关这些属性的列表，请参阅第 712 页的『不能与 IBM MQ Managed File Transfer 一起使用的 BPXWDYN 属性』。	缺省值如下： 对于 z/OS V1R8 和更高版本，为 MOUNT
armELEMENTYPE	可选属性。如果 Automatic Restart Manager (ARM) 将代理配置为重新启动，那么请将此属性设置为关联的 ARM 策略中指定的 ARM ELEMENTYPE 参数值。对于代理，请将 ELEMENTYPE 设置为 SYSBFGAG。	未设定
armELEMENT	可选属性。如果 Automatic Restart Manager (ARM) 将代理配置为重新启动，那么请将此属性设置为关联的 ARM 策略中指定的 ARM ELEMENT 参数值。您可以将 ELEMENT 值设置为与代理名称相对应。	未设定
其他属性：		
legacyXMLMessageMQMDFormat	通过代理生成的 IBM MQ Managed File Transfer XML 消息（例如，日志和传输进度消息）现在将发送至具有空白 MQMD 格式字段的队列。先前版本的产品将 MQMD 格式字段设置为 MQSTR（文本消息字符串）。如果将此属性设置为 true，那么就能够将通过代理生成的 IBM MQ Managed File Transfer XML 消息发送至 MQMD 格式字段为 MQSTR 的队列。 注：将使用与对应的命令请求匹配的消息格式发送针对命令的代理应答消息。 如果将 MQMD 格式字段设置为 MQSTR，并且如果在启用了数据转换的 MQ 网络中存在通道，那么 IBM MQ Managed File Transfer 命令 XML 消息可能会损坏。	false

相关概念

第 116 页的『[分布式平台上的配置选项](#)』

IBM MQ Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。这些属性文件位于安装产品时所定义的配置目录中。

相关参考

第 633 页的『[Java 系统属性](#)』

必须将许多 IBM MQ Managed File Transfer 命令和代理程序属性定义为 Java 系统属性，因为它们定义了无法使用命令或代理程序属性机制的早期功能的配置。

第 634 页的『[SSL 属性](#)』

将 SSL 或 TLS 与 IBM MQ 和 IBM MQ Managed File Transfer 一起使用可防止在代理与队列管理器之间建立未经授权的连接，并可对代理与队列管理器之间的消息通信进行加密。

第 460 页的『[fteCreateAgent（创建 IBM MQ Managed File Transfer 代理）](#)』

fteCreateAgent 命令创建代理及其关联的配置。

第 463 页的『[fteCreateBridgeAgent（创建和配置 IBM MQ Managed File Transfer 协议网桥代理）](#)』

fteCreateBridgeAgent 命令创建协议网桥代理及其关联的配置。针对您要与之进行文件的发送和接收操作的每个文件服务器，创建一个协议网桥代理。

第 469 页的『[fteCreateCDAgent（创建 Connect:Direct 网桥代理）](#)』

fteCreateCDAgent 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

第 516 页的『[fteCreateWebAgent（创建 IBM MQ Managed File Transfer Web 代理）](#)』

fteCreateWebAgent 命令创建代理及其关联的配置，以用于 Web 网关。IBM MQ Managed File Transfer 服务器随附该命令。

第 580 页的『[在 IBM MQ Managed File Transfer 属性中使用环境变量](#)』

从 IBM WebSphere MQ V7.5 起，可以在 受管文件传输 属性中使用表示文件或目录位置的环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化。例如，哪个用户在运行进程。

在集群中配置多个 IBM MQ 通道

如果您希望在集群配置中使用 IBM MQ 多通道支持，请首先将 `agentMultipleChannelsEnabled` 属性设置为 `true`，然后完成本主题中的步骤。

关于此任务

在集群中，多通道支持只能由目标代理的队列管理器上的 IBM MQ 定义启用。

除了完成在第 141 页的『[为首次使用配置 IBM MQ Managed File Transfer](#)』中列出的 IBM MQ Managed File Transfer 代理所需的 IBM MQ 标准配置以外，您还必须完成本主题中的步骤。

以下配置示例使用 **runmqsc** 命令。

过程

1. 为每个要使用的通道定义一个集群接收方通道。例如，如果您要使用两个通道：

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
```

其中：

- `DESTQMGRNAME` 是目标代理的队列管理器的名称。
- `MFTCLUSTER` 是 IBM MQ 集群的名称。

建议您对通道使用 `MFTCLUSTER.DESTMGRNAME_n` 命名约定，但不作强制要求。

2. 为每个对应的通道定义一个队列管理器别名。例如：

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
```

您必须对队列管理器的别名使用 `SYSTEM.FTE.DESTQMGRNAME_n` 命名约定，因为发送方代理会按此格式搜索队列管理器别名。用于 `n` 的编号必须从 1 开始，而且必须是连续的。您必须进行集群范围的定义，以使这些定义可在源代理的队列管理器上使用。

为了使源代理和目标代理能够正确确定队列管理器别名的数量，**请勿**为队列管理器定义缺省的 `XMITQ`。

相关概念

第 141 页的『[为首次使用配置 IBM MQ Managed File Transfer](#)』

在首次使用 IBM MQ Managed File Transfer 代理和队列管理器之前，必须先对其执行一些配置任务。

相关参考

第 591 页的『[agent.properties 文件](#)』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

在非集群配置中配置多个 IBM MQ 通道

如果您希望在非集群配置中使用 IBM MQ 多通道支持，请首先将 `agentMultipleChannelsEnabled` 属性设置为 `true`，然后完成本主题中的步骤。

关于此任务

在非集群配置中，多通道支持由源代理和目标代理的队列管理器上的 IBM MQ 定义启用。

除了完成在第 141 页的『[为首次使用配置 IBM MQ Managed File Transfer](#)』中列出的 IBM MQ Managed File Transfer 代理所需的 IBM MQ 标准配置以外，您还必须完成本主题中的步骤。

以下步骤假定源队列管理器 and 目标队列管理器之间使用发送方/接收方通道进行通信。

以下配置示例使用 `runmqsc` 命令。

过程

1. 在目标代理的队列管理器中，为每个要使用的通道定义一个接收方通道。例如，如果您要使用两个通道：

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(RCVR) TRPTYPE(TCP)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(RCVR) TRPTYPE(TCP)
```

其中：`DESTQMGRNAME` 是目标代理的队列管理器的名称。

建议您对通道使用 `TO.DESTMGRNAME_n` 命名约定，但不作强制要求。接收方通道的名称必须和源代理的队列管理器上对应的发送方通道的名称匹配。

2. 在源代理的队列管理器中，为每个要使用的通道定义一个传输队列。例如，如果您要使用两个通道：

```
DEFINE QLOCAL(DESTQMGRNAME_1) USAGE(XMITQ)
DEFINE QLOCAL(DESTQMGRNAME_2) USAGE(XMITQ)
```

建议您对传输队列使用 `DESTMGRNAME_n` 命名约定，但不作强制要求。您定义的传输队列会在后续步骤中的发送方通道定义和队列管理器别名定义中引用。

3. 在源代理的队列管理器中，为每个要使用的通道定义一个发送方通道。例如，如果您要使用两个通道：

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
XMITQ(DESTQMGRNAME_1)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
XMITQ(DESTQMGRNAME_2)
```

建议您对通道使用 `TO.DESTMGRNAME_n` 命名约定，但不作强制要求。发送方通道的名称必须和目标代理的队列管理器上对应的接收方通道的名称匹配。

4. 在源代理的队列管理器中，为每个对应的通道定义一个队列管理器别名。例如：

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_1)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_2)
```

您必须对队列管理器的别名使用 `SYSTEM.FTE.DESTQMGRNAME_n` 命名约定，因为发送方代理会按此格式搜索队列管理器。用于 `n` 的编号必须从 1 开始，而且必须是连续的。

为了使代理能够正确确定队列管理器别名的数量，**请勿**为队列管理器定义缺省的 `XMITQ`。

相关概念

第 141 页的『[为首次使用配置 IBM MQ Managed File Transfer](#)』
在首次使用 IBM MQ Managed File Transfer 代理和队列管理器之前，必须先对其执行一些配置任务。

相关参考

第 591 页的『[agent.properties 文件](#)』
每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。
`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

支持的用户定义的元数据键

如果代理属性 `enableUserMetadataOptions` 设置为值 `true`，那么在指定给新的传输请求时以下用户定义的元数据键受支持。

表 51: 元数据键		
键名称	描述	缺省值
com.ibm.wmqfte.insertRecordLineSeparator	用于文本传输。如果该键设置为 <code>true</code> ，那么指定在读取面向记录的文件（如 z/OS 数据集）时，在记录之间插入行分隔符。 如果该键设置为 <code>false</code> ，那么指定在读取面向记录的文件时，不在记录之间插入行分隔符。	true
com.ibm.wmqfte.newRecordOnLineSeparator	用于文本传输。如果该键设置为 <code>true</code> ，那么指定在写入面向记录的文件（如 z/OS 数据集）时，行分隔符指示新记录并且不作为数据的一部分写入。 如果该键设置为 <code>false</code> ，那么指定在写入面向记录的文件时，会将行分隔符与任何其他字符一样对待（即，没有记录分隔）。	true
com.ibm.wmqfte.convertLineSeparators	用于文本传输。指定是否将行分隔符序列 CRLF 和 LF 转换为目标所需的行分隔符序列。此转换目前只在以下情况下生效： - 当将用户定义的元数据键 <code>com.ibm.wmqfte.newRecordOnLineSeparator</code> 设置为 <code>false</code> 并且传输目标是面向记录的文件时。 - 当将用户定义的元数据键 <code>com.ibm.wmqfte.com.ibm.wmqfte.insertRecordLineSeparator</code> 设置为 <code>false</code> 并且传输源是面向记录的文件时。	true

相关信息

第 592 页的『[表 50](#)』
[fteCreateTransfer -md 参数](#)

附加代理配置文件

除了 `agent.properties` 文件外，代理程序的配置目录中还可以有多个 XML 配置文件。

配置文件

以下 XML 配置文件可用于指定代理使用的其他信息：

ProtocolBridgeCredentials.xml

如果代理为协议网桥代理，那么您可以使用该文件来指定要使用的凭证以登录到该代理所连接到的 FTP 或 SFTP 服务器。

ProtocolBridgeProperties.xml

如果代理为协议网桥代理，那么您可以使用该文件来定义该代理所连接到的非缺省协议文件服务器的属性。**fteCreateBridgeAgent** 命令在此文件中为您创建缺省协议文件服务器。

ConnectDirectCredentials.xml

如果代理为 Connect:Direct 网桥代理，那么您可以使用该文件来指定要使用的凭证以连接传输中涉及的 Connect:Direct 节点。

ConnectDirectNodeProperties.xml

如果代理为 Connect:Direct 网桥代理，那么可以使用该文件来指定传输中涉及的 Connect:Direct 节点的相关操作系统信息。

ConnectDirectProcessDefinition.xml

如果代理为 Connect:Direct 网桥代理，那么可以使用该文件来指定要在文件传输过程中调用的用户定义的 Connect:Direct 进程。

UserSandboxes.xml

您可以使用该文件来指定文件系统的哪些区域可由代理进行读取或写入。

更新配置文件

与 agent.properties 文件不同，您可以更新 XML 配置文件并让代理程序获取更改，而不必重新启动代理程序。

在提交传输时，如果距离代理上次检查 XML 配置文件的时间超过 10 秒，那么代理将检查 XML 配置文件的上次修改时间。如果 XML 配置文件在上次代理读取该文件之后已被修改，那么该代理将再次读取该文件。如果在与 XML 模式比较时该文件的内容有效，那么代理将更新其信息。如果该文件的内容无效，那么代理程序将使用该文件的先前版本中的信息并将消息写入 output0.log 文件。

相关参考

[第 607 页的『协议网桥凭证文件格式』](#)

代理配置目录中的 ProtocolBridgeCredentials.xml 文件定义用户名和凭证信息，供协议网桥代理用于在协议服务器中为自身授权。

[第 611 页的『协议网桥属性文件格式』](#)

代理配置目录中的 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性。

[第 618 页的『Connect:Direct 凭证文件格式』](#)

代理程序配置目录中的 ConnectDirectCredentials.xml 文件定义 Connect:Direct 代理程序用于向其自身授权 Connect:Direct 节点的用户名和凭证信息。

[第 621 页的『Connect:Direct 节点属性文件格式』](#)

Connect:Direct 网桥代理配置目录中的 ConnectDirectNodeProperties.xml 文件指定有关文件传输中涉及的远程 Connect:Direct 节点的信息。

[第 623 页的『Connect:Direct 进程定义文件格式』](#)

Connect:Direct 网桥代理配置目录中的 ConnectDirectProcessDefinitions.xml 文件指定要在文件传输过程中启动的用户定义的 Connect:Direct 进程。

[第 100 页的『使用用户沙箱』](#)

您可以根据请求传输的 MQMD 用户名来限制可在其中传输文件的文件系统的区域。

协议网桥凭证文件格式

代理配置目录中的 ProtocolBridgeCredentials.xml 文件定义用户名和凭证信息，供协议网桥代理用于在协议服务器中为自身授权。

ProtocolBridgeCredentials.xml 文件必须符合 ProtocolBridgeCredentials.xsd 模式。ProtocolBridgeCredentials.xsd 模式文档位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。用户负责手动创建 ProtocolBridgeCredentials.xml 文件，该文件不再由 **fteCreateBridgeAgent** 命令创建。MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples` 目录中提供了样本文件。

V7.5 引入了新的 <agent> 元素，其中包含指定代理的 <server> 或 <serverHost> 元素。

ProtocolBridgeCredentials.xml 文件由代理定期重新装入，并且对该文件所做的任何有效更改都将影响代理的行为。缺省重新装入时间间隔为 30 秒。此时间间隔可通过在 agent.properties 文件中指定代理程序属性 xmlConfigReloadInterval 进行更改。

模式 - V7.5 或更高版本

以下模式描述了 V8 的 ProtocolBridgeCredentials.xml 文件中哪些元素有效。

```
<schema targetNamespace="http://wmqfte.ibm.com/ProtocolBridgeCredentials" elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema" xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials">

  <!--
    <?xml version="1.0" encoding="UTF-8"?>
    <tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"
      xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials
        ProtocolBridgeCredentials.xsd">
      <tns:agent name="agent1">
        <tns:serverHost name="myserver">
          <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
          <tns:user name="jane" serverUserId="jane" hostKey="1F:2e:f3">
            <tns:privateKey associationName="test" keyPassword="pwd2">
              ... private key ...
            </tns:privateKey>
          </tns:user>
        </tns:serverHost>
      </tns:agent>

      <tns:agent name="agent2">
        <tns:server name="server*" pattern="wildcard">
          <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
          <tns:user name="jane" serverUserId="jane" hostKey="1F:2e:f3">
            <tns:privateKey associationName="test" keyPassword="pwd2">
              ... private key ...
            </tns:privateKey>
          </tns:user>
        </tns:server>
      </tns:agent>

      <tns:agent name="agent3">
        <tns:serverHost name="ftpsServer"
          keyStorePassword="keypass"
          trustStorePassword="trustpass">
          <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
        </tns:serverHost>
      </tns:agent>
    </tns:credentials>
  -->

  <element name="credentials" type="tns:credentialsType"/>

  <complexType name="credentialsType">
    <sequence>
      <element name="agent" type="tns:agentType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <complexType name="agentType">
    <choice minOccurs="0" maxOccurs="1">
      <element name="serverHost" type="tns:serverHostType" minOccurs="0" maxOccurs="unbounded"/>
      <element name="server" type="tns:serverType" minOccurs="0" maxOccurs="unbounded"/>
    </choice>
    <attribute name="name" type="string" use="required"/>
  </complexType>

  <complexType name="serverHostType">
    <sequence>
      <element ref="tns:user" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="name" type="string" use="required"/>
    <attribute name="keyStorePassword" type="string" use="optional"/>
    <attribute name="keyStorePasswordCipher" type="string" use="optional"/>
    <attribute name="trustStorePassword" type="string" use="optional"/>
    <attribute name="trustStorePasswordCipher" type="string" use="optional"/>
  </complexType>
```



```

<complexType name="serverType">
  <sequence>
    <element ref="tns:user" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="name" type="string" use="required"/>
  <attribute name="pattern" type="tns:patternType" use="optional" />
  <attribute name="keyStorePassword" type="string" use="optional"/>
  <attribute name="keyStorePasswordCipher" type="string" use="optional"/>
  <attribute name="trustStorePassword" type="string" use="optional"/>
  <attribute name="trustStorePasswordCipher" type="string" use="optional"/>
</complexType>

<element name="user" type="tns:userType"/>

<complexType name="userType">
  <sequence>
    <element ref="tns:privateKey" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="name" type="string" use="required"/>
  <attribute name="serverUserId" type="string" use="optional"/>
  <attribute name="serverUserIdCipher" type="string" use="optional"/>
  <attribute name="serverPassword" type="string" use="optional"/>
  <attribute name="serverPasswordCipher" type="string" use="optional"/>
  <attribute name="hostKey" use="optional">
    <simpleType>
      <restriction base="string">
        <pattern
          value="([a-zA-F0-9]){2}(:([a-zA-F0-9]){2})*" />
        </pattern>
      </restriction>
    </simpleType>
  </attribute>
</complexType>

<element name="privateKey" type="tns:privateKeyType"/>

<complexType name="privateKeyType">
  <simpleContent>
    <extension base="string">
      <attribute name="keyPassword" type="string" use="optional"/>
      <attribute name="keyPasswordCipher" type="string" use="optional"/>
      <attribute name="associationName" type="string" use="required"/>
    </extension>
  </simpleContent>
</complexType>

<!--
  Determines the type of pattern matching to use.
-->
<simpleType name="patternType">
  <restriction base="string">
    <enumeration value="regex" />
    <enumeration value="wildcard" />
  </restriction>
</simpleType>
</schema>

```

了解 ProtocolBridgeCredentials.xml 文件

下表中描述了在 ProtocolBridgeCredentials.xml 文件中使用的元素和属性。

<credentials>

组元素，其中包含描述由协议网桥代理用于连接协议服务器的凭证的元素。

<agent>

包含指定代理程序的 <server> 或 <serverHost> 定义的元素。

<server>

协议网桥连接到的协议服务器。

V7.0.4 或更早版本不支持 <server> 元素。

属性	描述
名	协议服务器的名称。

属性	描述
模式 (pattern)	如果您已使用通配符或正则表达式来指定协议服务器名称的模式，那么请使用 wildcard 或 regex 。
trustStorePassword 或 trustStorePasswordCipher	在 <server> 元素引用 FTPS 服务器时为必需。 用于访问信任库的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
keyStorePassword 或 keyStorePasswordCipher	可选。用于访问密钥库的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。

<serverHost>

协议网桥连接的协议服务器的主机名称。

The ProtocolBridgeCredentials.xml file can either contain <serverHost> elements or <server> elements but you cannot use a mixture of the two different types. 当您使用 <serverHost> 时，名称与协议服务器的主机名匹配。When you use <server>, the name is matched against the protocol server's name (as defined in the ProtocolBridgeProperties.xml file).

属性	描述
名	协议服务器的主机名或 IP 地址。
trustStorePassword 或 trustStorePasswordCipher	在 <serverHost> 元素引用 FTPS 服务器时为必需。用于访问信任库的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
keyStorePassword 或 keyStorePasswordCipher	可选。用于访问密钥库的密码。此属性是可选的，除非您设置了 keyStore 属性，在这种情况下为必需。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。

<user>

从 IBM MQ Managed File Transfer 用户名到协议服务器用户名的用户映射。

属性	描述
名	在 IBM MQ Managed File Transfer 上使用的用户名。
serverUserId 或 serverUserIdCipher	在协议服务器上使用的用户名。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
serverPassword 或 serverPasswordCipher	在协议服务器上使用的用户名的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
hostKey	服务器主机 SSH 指纹。

<privateKey>

用户的专用密钥。

属性	描述
keyPassword 或 keyStorePasswordCipher	专用密钥的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
associationName	用于跟踪和记录的名称。

相关概念

第 275 页的『协议网桥』

通过协议网桥，IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件，而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协

议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

相关任务

第 281 页的『使用 ProtocolBridgeCredentials.xml 文件映射文件服务器的凭证』

通过使用协议网桥代理的缺省凭证映射功能，将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器的用户凭证。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以包含自己的凭证信息。

第 285 页的『示例：如何配置协议网桥代理以将专用密钥凭证用于 UNIX SFTP 服务器』

该示例演示了如何生成和配置 ProtocolBridgeCredentials.xml 文件。该示例是典型示例，细节会随着您的平台而不同，但原理是一致的。

第 276 页的『使用 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性』

定义要使用 ProtocolBridgeProperties.xml 文件（由 IBM MQ Managed File Transfer 在代理配置目录中提供）传输文件的一个或多个协议文件服务器的属性。

相关参考

第 550 页的『fteObfuscate（加密敏感数据）』

fteObfuscate 命令用于加密凭证文件中的敏感数据。这样，获取了对凭证文件的访问权的人员就无法读取该文件的内容。

协议网桥属性文件格式

代理配置目录中的 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性。

ProtocolBridgeProperties.xml 文件必须符合 ProtocolBridgeProperties.xsd 模式。

ProtocolBridgeProperties.xsd 模式文档位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。模板文件 ProtocolBridgeProperties.xml 通过

fteCreateBridgeAgent 命令创建在代理配置目录中。

ProtocolBridgeProperties.xml 文件由代理定期重新装入，并且对该文件所做的任何有效更改都将影响代理的行为。缺省重新装入时间间隔为 30 秒。此时间间隔可通过在 `agent.properties` 文件中指定代理程序属性 `xmlConfigReloadInterval` 进行更改。

模式

以下模式描述 ProtocolBridgeProperties.xml 文件。

注: IBM MQ V7.5 或者 IBM MQ Managed File Transfer V7.0.2 或更高版本上不支持 `maxReconnectRetry` 和 `reconnectWaitPeriod` 属性。

```
<schema targetNamespace="http://wmqfte.ibm.com/ProtocolBridgeProperties" elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema" xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties">

  <!--
    Example: ProtocolBridgeProperties.xml

    <?xml version="1.0" encoding="UTF-8"?>
    <tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
      xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
        ProtocolBridgeProperties.xsd">
      <tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
      <tns:defaultServer name="myserver" />
      <tns:ftpServer name="myserver" host="myhost.hursley.ibm.com" port="1234" platform="windows"
        timeZone="Europe/London" locale="en-GB" fileEncoding="UTF-8"
        listFormat="unix" limitedWrite="false" />
      <tns:sftpServer name="server1" host="myhost.hursley.ibm.com" platform="windows"
        fileEncoding="UTF-8" limitedWrite="false">
        <limits maxListFileNames="10" />
      </tns:sftpServer>
    </tns:serverProperties>
  -->

  <!-- Root element for the document -->
  <element name="serverProperties" type="tns:serverPropertiesType"/></element>

  <!--
    A container for all protocol bridge server properties
  -->
```

```

<complexType name="serverPropertiesType">
  <sequence>
    <element name="credentialsFile" type="tns:credentialsFileName" minOccurs="0" maxOccurs="1" />
    <element name="defaultServer" type="tns:serverName" minOccurs="0" maxOccurs="1" />
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="ftpServer" type="tns:ftpServerType" />
      <element name="sftpServer" type="tns:sftpServerType" />
      <element name="ftpsServer" type="tns:ftpsServerType" />
      <element name="ftpsfgServer" type="tns:ftpsfgServerType" />
      <element name="ftpssfgServer" type="tns:ftpssfgServerType" />
    </choice>
  </sequence>
</complexType>

<!--
  A container for a server name
-->
<complexType name="serverName">
  <attribute name="name" type="tns:serverNameType" use="required" />
</complexType>

<!--
  A container for a credentials file name
-->
<complexType name="credentialsFileName">
  <attribute name="path" type="string" use="required" />
</complexType>

<!--
  A container for all the information about an FTP server
-->
<complexType name="ftpServerType">
  <sequence>
    <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
  </sequence>
  <attributeGroup ref="tns:ftpServerAttributes"/>
  <attribute name="passiveMode" type="boolean" use="optional" />
</complexType>

<!--
  A container for all the information about an SFG FTP server
-->
<complexType name="ftpsfgServerType">
  <sequence>
    <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
  </sequence>
  <attributeGroup ref="tns:ftpServerAttributes"/>
</complexType>

<!--
  A container for all the information about an SFTP server
-->
<complexType name="sftpServerType">
  <sequence>
    <element name="limits" type="tns:sftpLimitsType" minOccurs="0" maxOccurs="1" />
  </sequence>
  <attributeGroup ref="tns:sftpServerAttributes"/>
</complexType>

<!--
  A container for all the information about a FTPS server
-->
<complexType name="ftpsServerType">
  <sequence>
    <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
  </sequence>
  <attributeGroup ref="tns:ftpsServerAttributes"/>
</complexType>

<!--
  A container for all the information about a SFG FTPS server
-->
<complexType name="ftpssfgServerType">
  <sequence>
    <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
  </sequence>
  <attributeGroup ref="tns:ftpsServerAttributes"/>
</complexType>

<!--
  Attributes common to all server types
-->

```

```

-->
<attributeGroup name="generalServerAttributes">
  <attribute name="name" type="tns:serverNameType" use="required" />
  <attribute name="host" type="string" use="required" />
  <attribute name="port" type="nonNegativeInteger" use="optional" />
  <attribute name="platform" type="tns:platformType" use="required" />
  <attribute name="fileEncoding" type="string" use="required" />
  <attribute name="limitedWrite" type="boolean" use="optional" />
  <attribute name="controlEncoding" type="string" use="optional" />
</attributeGroup>

<!--
  Attributes common to ftp and ftps server types
-->
<attributeGroup name="ftpServerAttributes">
  <attributeGroup ref="tns:generalServerAttributes"/>
  <attribute name="timeZone" type="string" use="required" />
  <attribute name="locale" type="tns:localeType" use="required" />
  <attribute name="listFormat" type="tns:listFormatType" use="optional" />
  <attribute name="listFileRecentDateFormat" type="tns:dateFormatType" use="optional" />
  <attribute name="listFileOldDateFormat" type="tns:dateFormatType" use="optional" />
  <attribute name="monthShortNames" type="tns:monthShortNamesType" use="optional" />
</attributeGroup>

<!--
  Attributes common to ftps server types
-->
<attributeGroup name="ftpsServerAttributes">
  <attributeGroup ref="tns:ftpServerAttributes"/>
  <attribute name="ftpsType" type="tns:ftpsTypeType" use="optional" />
  <attribute name="trustStore" type="string" use="required" />
  <attribute name="trustStoreType" type="string" use="optional" />
  <attribute name="keyStore" type="string" use="optional" />
  <attribute name="keyStoreType" type="string" use="optional" />
  <attribute name="ccc" type="boolean" use="optional" />
  <attribute name="protFirst" type="boolean" use="optional" />
  <attribute name="auth" type="string" use="optional" />
  <attribute name="connectTimeout" type="nonNegativeInteger" use="optional" />
</attributeGroup>

<!--
  A container for limit-type attributes for a server. Limit parameters
  are optional, and if not specified a system default will be used.
-->
<complexType name="generalLimitsType">
  <attributeGroup ref="tns:generalLimitAttributes"/>
</complexType>

<complexType name="sftpLimitsType">
  <attributeGroup ref="tns:generalLimitAttributes"/>
  <attribute name="connectionTimeout" type="nonNegativeInteger" use="optional" />
</complexType>

<!--
  Attributes for limits common to all server types
-->
<attributeGroup name="generalLimitAttributes">
  <attribute name="maxListFileNames" type="positiveInteger" use="optional" />
  <attribute name="maxListDirectoryLevels" type="nonNegativeInteger" use="optional" />
  <attribute name="maxReconnectRetry" type="nonNegativeInteger" use="optional" />
  <attribute name="reconnectWaitPeriod" type="nonNegativeInteger" use="optional" />
  <attribute name="maxSessions" type="positiveInteger" use="optional" />
  <attribute name="socketTimeout" type="nonNegativeInteger" use="optional" />
</attributeGroup>

<!--
  The type for matching valid server names. Server names must be at least 2 characters in length
and
  are limited to alphanumeric characters and the following characters: ".", "_", "/" and "%".
-->
<simpleType name="serverNameType">
  <restriction base="string">
    <pattern value="[0-9a-zA-Z\._%]{2,}" />
  </restriction>
</simpleType>

<!--
  The types of platform supported.
-->
<simpleType name="platformType">
  <restriction base="string">

```

```

    </restriction>
</simpleType>

<!--
    The type for matching a locale specification.
-->
<simpleType name="localeType">
    <restriction base="string">
        <pattern value="(\.)([_])(\.)"/>
    </restriction>
</simpleType>

<!--
    The types of list format supported (for FTP servers).
-->
<simpleType name="listFormatType">
    <restriction base="string">
    </restriction>
</simpleType>

<!--
    Date format for FTP client directory listing on an FTP server. This is
    the format to be passed to methods setDefaultDateFormatStr and
    setRecentDateFormatStr for Java class:
    org.apache.commons.net.ftp.FTPClientConfig
-->
<simpleType name="dateFormatType">
    <restriction base="string">
    </restriction>
</simpleType>

<!--
    A list of language-defined short month names can be specified. These are
    used for translating the directory listing received from the FTP server.
    The format is a string of three character month names separated by "|"
-->
<simpleType name="monthShortNamesType">
    <restriction base="string">
        <pattern value="(\.\.\\|){11}(\.\.)/>
    </restriction>
</simpleType>

<!--
    The enumerations of the allowed FTPS types: "implicit" & "explicit"
    If not specified the default is "explicit"
-->
<simpleType name="ftpsTypeType">
    <restriction base="string">
        <enumeration value="explicit"/>
        <enumeration value="implicit"/>
    </restriction>
</simpleType>

<!--
    Attribute Group for SFTP Servers
-->
<attributeGroup name="sftpServerAttributes">
    <attributeGroup ref="tns:generalServerAttributes"/>
    <attribute name="cipherList" type="string" use="optional" />
</attributeGroup>
</schema>

```

了解 ProtocolBridgeProperties.xml 文件

以下列表中描述了 ProtocolBridgeProperties.xml 文件中使用的元素和属性:

<serverProperties>

XML 文档的根元素

<credentialsFile>

包含凭证的文件的路径。对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。有关更多信息, 请参阅第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』。

<defaultServer>

充当文件传输的缺省服务器的协议文件服务器

<ftpServer>

FTP 文件服务器

<sftpServer>

SFTP 文件服务器

<ftpsServer>

FTPS 文件服务器

适用于所有类型的协议文件服务器的一般服务器属性:

属性	描述
名	必需。协议文件服务器的名称。协议服务器名称长度必须至少为两个字符，不区分大小写，且局限于字母数字字符和以下字符： • 句点 (.) • 下划线 (_) • 正斜杠 (/) • 百分号 (%)
主机	必需。要向其发送文件或接收其文件的协议文件服务器的主机名或 IP 地址。
port	可选。要向其发送文件或接收其文件的协议文件服务器的端口号。
platform	必需。要向其发送文件或接收其文件的协议文件服务器的平台。指定 UNIX 或 WINDOWS。根据您在 FTP、FTPS 或 SFTP 服务器上输入路径的方法，设置该属性。例如，如果您正在 Windows 上运行 FTP 服务器，但当您登录该服务器时，必须输入 UNIX 样式的路径（即，采用正斜杠），将该值设置为 UNIX，而不是 WINDOWS。Windows 上运行的服务器通常有 UNIX 样式文件系统。
fileEncoding	必需。定义文件服务器使用的字符编码。当您以文本方式传输文件时，使用该属性，以便在平台之间移动文件时更改正确的编码序列。例如，UTF-8。
limitedWrite	可选。写入文件服务器时的缺省方式是创建临时文件，然后在完成传输后重命名该文件。对于配置为只写的文件服务器，会使用其最终名称直接创建此文件。该属性的值可以为 true 或 false。缺省值为 false。
controlEncoding	可选。正发送给协议文件服务器的控制消息的控制编码值。该属性影响使用的文件名称的编码，并且必须与协议文件服务器的控制编码兼容。缺省值为 UTF-8。

仅适用于 FTP 和 FTPS 服务器的常规属性:

属性	描述
timeZone	必需。要向其发送文件或接收其文件的协议文件服务器的时区。例如：America/New_York 或 Asia/Tokyo。
locale	必需。要向其发送文件或接收其文件的协议文件服务器上使用的语言。例如：en_US 或 ja_JP
listFormat	可选。定义从协议文件服务器返回的文件列出信息格式的列表格式。使用 Windows 或 UNIX。缺省值为 UNIX。
listFileRecentDateFormat	可选。FTP 服务器上列出的 FTP 客户机目录的最近日期格式（少于一年）。该属性和 listFileOldDateFormat 属性允许您重新定义由协议文件服务器返回的预期日期格式。缺省值是由协议文件服务器定义的。
listFileOldDateFormat	可选。FTP 服务器上列出的 FTP 客户机目录的旧日期格式（一年以上）。该属性和 listFileRecentDateFormat 属性允许您重新定义由协议文件服务器返回的预期日期格式。缺省值是由协议文件服务器定义的。

属性	描述
monthShortNames	可选。用于对从协议文件服务器返回的日期信息进行解码的月名称的替换列表。该属性包含 12 个以逗号分隔的名称的列表，以覆盖缺省语言环境月值。缺省值是由协议文件服务器定义的。

仅适用于 **FTP** 服务器的常规属性：

属性	描述
passiveMode	可选。控制以被动方式还是主动方式连接到 FTP 服务器。 如果您将该属性的值设置为 false ，那么连接采用主动方式。如果您将该值设置为 true ，那么连接采用被动方式。缺省值为 false 。

仅适用于 **FTPS** 服务器的常规属性：

属性	描述
ftpType	可选。指定使用 FTPS 协议的显式还是隐式格式。缺省值为 explicit 。
trustStore	必需。用于确定 FTPS 服务器提供的证书是否可信的信任库的位置。
trustStoreType	可选。信任库文件的格式。缺省值为 JKS 。
keyStore	可选。用于在 FTPS 服务器质询时提供证书信息的密钥库的位置。缺省情况下，协议网桥不能与配置为需要客户机认证的 FTPS 服务器连接。
keyStoreType	可选。密钥库文件的格式。缺省值为 JKS 。
ccc	可选。选择在认证完成时是否使用清除（未加密）命令通道。缺省值为 false ，这表示在整个 FTPS 会话期间命令通道保持加密。该属性仅在 ftpType 设置为 explicit 时才适用。
protFirst	可选。指定在 PBSZ/PROT 命令之前还是之后向 FTPS 服务器发出 USER/PASS 命令。缺省值为 false ，这表示先发送 USER/PASS 命令，然后发送 PBSZ/PROT 命令。该属性仅在 ftpType 设置为 explicit 时才适用。
auth	可选。指定在 AUTH 命令中指定的协议。将首先尝试指定的协议，然后缺省行为是尝试 TLS、SSL、TLS-C 或 TLS-P，直至 FTPS 服务器不会以 504 应答代码进行拒绝。该属性仅在 ftpType 设置为 explicit 时才适用。

<limits>

所有类型服务器的常见属性和特定于某类型服务器的属性的容器元素：

适用于所有类型的协议文件服务器的一般限制属性：

属性	描述
maxListFileNames	可选。在协议文件服务器上的目录中扫描文件名时收集的最大名称数。缺省值为 999999999。
maxListDirectoryLevels	可选。在协议服务器上递归扫描文件名的最大目录级别数。缺省值是 1000。
maxReconnectRetry (现在不推荐使用该属性。)	不推荐使用 。该属性在 IBM MQ V7.5 或者 IBM MQ Managed File Transfer V7.0.2 或更高版本上不受支持。 可选。协议网桥代理停止尝试之前，协议服务器尝试重新连接的最大次数。缺省值为 2。
reconnectWaitPeriod	不推荐使用 。该属性在 IBM MQ V7.5 或者 IBM MQ Managed File Transfer V7.0.2 或更高版本上不受支持。

属性	描述
(现在不推荐使用该属性。)	可选。尝试重新连接之前等待的时间段（秒）。缺省值为 10 秒。
maxSessions	可选。协议服务器的最大会话数。该数字必须大于或等于协议网桥代理源和目标传输的最大数之和。缺省值是代理属性 maxSource 传输, maxDestination 传输和 maxCommandHandlerThreads 以及 1 的值总和。如果这三个属性正在使用其缺省值 25,25 和 5，那么 maxSessions 缺省值为 56。
socketTimeout	可选。套接字超时（秒）。文件流期间使用该属性的值。缺省值为 30 秒。

仅限于适用于 **SFTP** 服务器的属性：

属性	描述
connectionTimeout	可选。等待协议文件服务器对连接请求响应的时间（秒）。超时表示协议文件服务器不可用。缺省值是 30 秒。
cipherList	可选。指定用于在协议网桥代理和 SFTP 服务器之间通信的密码的逗号分隔列表。密码调用顺序与此列表中指定密码的顺序相同。在可以使用密码之前，必须在服务器和客户机上提供该密码。 协议网桥代理支持的密码如下所示： • blowfish-cbc • 3des-cbc • aes128-cbc • aes192-cbc • aes256-cbc • aes128-ctr • aes192-ctr • aes256-ctr • 3des-ctr • arcfour • arcfour128 • arcfour256 缺省情况下，协议网桥代理所使用的密码列表为 aes128-cbc,aes192-cbc,aes256-cbc。

相关概念

[第 275 页的『协议网桥』](#)

通过协议网桥，IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件，而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

相关任务

[第 276 页的『使用 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性』](#)

定义要使用 ProtocolBridgeProperties.xml 文件（由 IBM MQ Managed File Transfer 在代理配置目录中提供）传输文件的一个或多个协议文件服务器的属性。

[第 281 页的『使用 ProtocolBridgeCredentials.xml 文件映射文件服务器的凭证』](#)

通过使用协议网桥代理的缺省凭证映射功能，将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器的用户凭证。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以包含自己的凭证信息。

第 285 页的『示例：如何配置协议网桥代理以将专用密钥凭证用于 UNIX SFTP 服务器』

该示例演示了如何生成和配置 `ProtocolBridgeCredentials.xml` 文件。该示例是典型示例，细节会随着您的平台而不同，但原理是一致的。

相关参考

第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』

从 IBM WebSphere MQ V7.5 起，可以在受管文件传输属性中使用表示文件或目录位置的环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化。例如，哪个用户在运行进程。

Connect:Direct 凭证文件格式

代理程序配置目录中的 `ConnectDirectCredentials.xml` 文件定义 Connect:Direct 代理程序用于向其自身授权 Connect:Direct 节点的用户名和凭证信息。

`ConnectDirectCredentials.xml` 文件必须符合 `ConnectDirectCredentials.xsd` 模式。`ConnectDirectCredentials.xsd` 模式文档位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。`ConnectDirectCredentials.xml` 样本文件位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/credentials` 目录中。

文件 `ConnectDirectCredentials.xml` 由代理定期重载，对该文件的任何有效更改都将影响该代理的行为。缺省重新装入时间间隔为 30 秒。可以通过在 `agent.properties` 文件中指定代理属性 `xmlConfigReloadInterval` 来更改此时间间隔。

模式

以下模式描述在 `ConnectDirectCredentials.xml` 文件中有效的元素。

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
  This schema defines the format of the XML file that is located in the agent properties
  directory of a Connect:Direct bridge agent. The XML file ConnectDirectCredentials.xml
  is used by the default credential validation of the Connect:Direct bridge.
  For more information, see the WebSphere MQ InfoCenter
-->

<schema targetNamespace="http://wmqfte.ibm.com/ConnectDirectCredentials"
  elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://wmqfte.ibm.com/ConnectDirectCredentials"

  <!--
    <?xml version="1.0" encoding="UTF-8"?>

    <tns:credentials xmlns:tns="http://wmqfte.ibm.com/ConnectDirectCredentials"
      xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://wmqfte.ibm.com/ConnectDirectCredentials
        ConnectDirectCredentials.xsd">
      <tns:agent name="CDAGENT01">
        <tns:pnode name="cdnode*" pattern="wildcard">
          <tns:user name="MUSR_.*"
            ignorecase="true"
            pattern="regex"
            cdUserId="bob"
            cdPassword="passw0rd"
            pnodeUserId="bill"
            pnodePassword="alacazam">
          <tns:snode name="cdnode2" pattern="wildcard" userId="sue" password="foo"/>
          </tns:user>
        </tns:pnode>
      </tns:agent>
    </tns:credentials>

    -->

    <element name="credentials" type="tns:credentialsType"/>

    <complexType name="credentialsType">
```

```

    <sequence>
      <element name="agent" type="tns:agentType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <complexType name="agentType">
    <sequence>
      <element name="pnode" type="tns:pnodeType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="name" type="string" use="required"/>
  </complexType>

  <complexType name="pnodeType">
    <sequence>
      <element name="user" type="tns:userType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="name" type="string" use="required"/>
    <attribute name="pattern" type="tns:patternType" use="optional"/>
  </complexType>

  <complexType name="userType">
    <sequence>
      <element name="snode" type="tns:snodeType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="name" type="string" use="required"/>
    <attribute name="ignorecase" type="boolean" use="optional"/>
    <attribute name="pattern" type="tns:patternType" use="optional"/>
    <attribute name="cdUserId" type="string" use="optional"/>
    <attribute name="cdUserIdCipher" type="string" use="optional"/>
    <attribute name="cdPassword" type="string" use="optional"/>
    <attribute name="cdPasswordCipher" type="string" use="optional"/>
    <attribute name="pnodeUserId" type="string" use="optional"/>
    <attribute name="pnodeUserIdCipher" type="string" use="optional"/>
    <attribute name="pnodePassword" type="string" use="optional"/>
    <attribute name="pnodePasswordCipher" type="string" use="optional"/>
  </complexType>

  <complexType name="snodeType">
    <attribute name="name" type="string" use="required"/>
    <attribute name="pattern" type="tns:patternType" use="optional"/>
    <attribute name="userId" type="string" use="optional"/>
    <attribute name="userIdCipher" type="string" use="optional"/>
    <attribute name="password" type="string" use="optional"/>
    <attribute name="passwordCipher" type="string" use="optional"/>
  </complexType>

  <simpleType name="patternType">
    <restriction base="string">
      <enumeration value="regex"/>
      <enumeration value="wildcard"/>
    </restriction>
  </simpleType>
</schema>

```

了解 ConnectDirectCredentials.xml 文件

下表中描述了在 ConnectDirectCredentials.xml 文件中使用的元素和属性。

<credentials>

包含描述 Connect:Direct 网桥代理用于连接到 Connect:Direct 节点的凭证的元素的组元素。

<agent>

包含指定代理的 <pnode> 定义元素的组元素。

<pnode>

Connect:Direct 传输中的主节点 (PNODE)。该节点可启动到辅助节点 (SNODE) 的连接。

属性	描述
名	Connect:Direct 节点的名称。该属性的值可以是匹配多个节点名称的模式。
模式 (pattern)	指定用于 name 属性的值的模式类型。pattern 属性的有效值为 - wildcard - 使用通配符 - regex-使用 Java 正则表达式

<user>

提交传输请求的 WebSphere MQ 用户。

属性	描述
名	在 IBM MQ Managed File Transfer 上使用的用户名。该属性的值可以是匹配多个用户名的模式。
ignorecase	指定该名称是否区分大小写。ignorecase 属性的有效值包括 • true - 名称不区分大小写 • false - 名称区分大小写
模式 (pattern)	指定用于 name 属性的值的模式类型。pattern 属性的有效值为 • wildcard - 使用通配符 • regex-使用 Java 正则表达式
cdUserId 或 cdUserIdCipher	Connect:Direct 网桥用于连接到其关联的 Connect:Direct 节点的用户名。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
cdPassword 或 cdPasswordCipher	与 cdUserId 属性指定的用户名相关联的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
pnodeUserId 或 pnodeUserIdCipher	Connect:Direct 主节点所使用的用户名。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
pnodePassword 或 pnodePasswordCipher	与 pnodeUserId 属性指定的用户名相关联的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。

<snode>

在 Connect:Direct 文件传输期间执行辅助节点 (SNODE) 角色的 Connect:Direct 节点。

属性	描述
名	Connect:Direct 节点的名称。该属性的值可以是匹配多个节点名称的模式。
模式 (pattern)	指定用于 name 属性的值的模式类型。pattern 属性的有效值为 • wildcard - 使用通配符 • regex-使用 Java 正则表达式
userId 或 userIdCipher	在文件传输期间用于连接到该节点的用户名。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
password 或 passwordCipher	与 userId 属性指定的用户名相关联的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。

示例

在此示例中，Connect:Direct 网桥代理连接到 Connect:Direct 节点 pnode1。当用户名以前缀 fteuser 开头并后跟单个字符 (例如 fteuser2) 的 WebSphere MQ 用户请求涉及 Connect:Direct 网桥的传输时，Connect:Direct 网桥代理将使用用户名 cduser 和密码 passw0rd 来连接到 Connect:Direct 节点 pnode1。当 Connect:Direct 节点 pnode1 执行其部分传输时，它将使用用户名 pnodeuser 和密码 passw0rd1。

如果 Connect:Direct 传输中的辅助节点具有以前缀 FISH 开头的名称，那么节点 pnode1 将使用用户名 fishuser 和密码 passw0rd2 来连接到辅助节点。如果 Connect:Direct 传输中的辅助节点具有以前缀 CHIPS 开头的名称，那么节点 pnode1 将使用用户名 chipsuser 和密码 passw0rd3 来连接到辅助节点。

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:credentials xmlns:tns="http://wmqfte.ibm.com/ConnectDirectCredentials"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ConnectDirectCredentials
```



```

ConnectDirectCredentials.xsd">
  <tns:agent name="CDAGENT01">
    <tns:pnode name="pnode1" pattern="wildcard">
      <tns:user name="fteuser?" pattern="wildcard" ignorecase="true"
        cdUserId="cduser" cdPassword="passw0rd"
        pnodeUserId="pnodeuser" pnodePassword="passw0rd1">
      <tns:snode name="FISH*" pattern="wildcard"
        userId="fishuser" password="passw0rd2"/>
      <tns:snode name="CHIPS*" pattern="wildcard"
        userId="chipsuser" password="passw0rd3"/>
    </tns:user>
  </tns:pnode>
</tns:agent>
</tns:credentials>

```

相关概念

第 288 页的『Connect:Direct 网桥』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关参考

第 469 页的『fteCreateCDAgent（创建 Connect:Direct 网桥代理）』

fteCreateCDAgent 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 Connect:Direct 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 `java.util.regex` API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

Connect:Direct 节点属性文件格式

Connect:Direct 网桥代理配置目录中的 `ConnectDirectNodeProperties.xml` 文件指定有关文件传输中涉及的远程 Connect:Direct 节点的信息。

`ConnectDirectNodeProperties.xml` 文件必须符合 `ConnectDirectNodeProperties.xsd` 模式。`ConnectDirectNodeProperties.xsd` 模式文档位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。模板 `ConnectDirectNodeProperties.xml` 文件由 **fteCreateCDAgent** 命令在代理配置目录中创建。

文件 `ConnectDirectNodeProperties.xml` 由代理定期重载，对该文件的任何有效更改都将影响该代理的行为。缺省重新装入时间间隔为 30 秒。该时间间隔可以通过指定 `agent.properties` 文件中的代理属性 `xmlConfigReloadInterval` 进行更改。

模式

以下模式描述在 `ConnectDirectNodeProperties.xml` 文件中有效的元素。

```

<?xml version="1.0" encoding="UTF-8"?>

<schema targetNamespace="http://wmqfte.ibm.com/ConnectDirectNodeProperties"
  elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://wmqfte.ibm.com/ConnectDirectNodeProperties">

  <element name="nodeProperties" type="tns:nodePropertiesType"></element>

  <complexType name="nodePropertiesType">
    <sequence>
      <element name="credentialsFile" type="tns:credentialsFileName" minOccurs="0" maxOccurs="1" />
      <element name="node" type="tns:nodeType" minOccurs="0" maxOccurs="unbounded"></element>
    </sequence>
  </complexType>

  <complexType name="nodeType">
    <attribute name="name" type="string" use="required" />
    <attribute name="pattern" type="tns:patternType" use="optional" />
    <attribute name="type" type="string" use="required" />
  </complexType>

```

```

<simpleType name="patternType">
  <restriction base="string">
    <enumeration value="regex" />
    <enumeration value="wildcard" />
  </restriction>
</simpleType>

</schema>

```

了解 ConnectDirectNodeProperties.xml 文件

下表中描述了 ConnectDirectNodeProperties.xml 文件中使用的元素和属性。

nodeProperties

XML 文档的根元素。

credentialsFile

存储敏感信息的凭证文件的路径。对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。有关更多信息，请参阅第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』。

node

指定一个或多个 Connect:Direct 节点。

属性	描述
名	一种模式，用于标识使用节点元素指定的定义的 Connect:Direct 节点的名称。模式匹配不区分大小写。
模式 (pattern)	指定用于 name 属性值的模式类型。 pattern 属性的有效值为: - wildcard - 使用通配符 - regex-使用 Java 正则表达式 要了解有关 MQMFT 使用的正则表达式类型的信息，请参阅 第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』。
类型	指定与 name 属性给定的模式匹配的一个或多个 Connect:Direct 节点的操作系统类型。 type 属性的有效值为: - Windows - 节点在 Windows 上运行 - UNIX -节点在 UNIX 或 Linux 上运行 - z/OS、zos、os/390 或 os390 - 节点在 z/OS 上运行 该属性的值不区分大小写。

示例

在该示例中，文件将指定以下关联：

- 所有名称以“cdnodew”开头的 Connect:Direct 节点都在 Windows 平台上运行。
- 所有名称以“cdnodeu”开头的 Connect:Direct 节点都在 UNIX 平台上运行。
- 所有名称以“cdnodez”开头的 Connect:Direct 节点都在 z/OS 上运行。
- 所有其他 Connect:Direct 节点都在 UNIX 平台上运行。

Connect:Direct 网桥代理搜索从文件开头到结尾的匹配项，并使用找到的第一个匹配项。Connect:Direct 凭证文件已指定为 ConnectDirectCredentials.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<tns:nodeProperties xmlns:tns="http://wmqfte.ibm.com/ConnectDirectNodeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ConnectDirectNodeProperties
    ConnectDirectNodeProperties.xsd">

  <tns:credentialsFile path="ConnectDirectCredentials.xml" />
  <tns:node name="cdnodew*" pattern="wildcard" type="windows" />
  <tns:node name="cdnodeu.*" pattern="regex" type="unix" />

```

```
<tns:node name="cdnodez*" pattern="wildcard" type="zos" />
<tns:node name="*" pattern="wildcard" type="unix" />

</tns:nodeProperties>
```

相关概念

第 288 页的『[Connect:Direct 网桥](#)』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关参考

第 469 页的『[fteCreateCDAgent（创建 Connect:Direct 网桥代理）](#)』

`fteCreateCDAgent` 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

第 724 页的『[IBM MQ Managed File Transfer 使用的正则表达式](#)』

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 Connect:Direct 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 `java.util.regex` API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

第 580 页的『[在 IBM MQ Managed File Transfer 属性中使用环境变量](#)』

从 IBM WebSphere MQ V7.5 起，可以在受管文件传输属性中使用表示文件或目录位置的环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化。例如，哪个用户在运行进程。

Connect:Direct 进程定义文件格式

Connect:Direct 网桥代理配置目录中的 `ConnectDirectProcessDefinitions.xml` 文件指定要在文件传输过程中启动的用户定义的 Connect:Direct 进程。

`ConnectDirectProcessDefinitions.xml` 文件必须符合

`ConnectDirectProcessDefinitions.xsd` 模式。`ConnectDirectProcessDefinitions.xsd` 模式文档位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。模板 `ConnectDirectProcessDefinitions.xml` 文件由 `fteCreateCDAgent` 命令在代理配置目录中创建。

文件 `ConnectDirectProcessDefinitions.xml` 由代理定期重载，对该文件的任何有效更改都将影响该代理的行为。缺省重新装入时间间隔为 30 秒。该时间间隔可以通过指定 `agent.properties` 文件中的代理属性 `xmlConfigReloadInterval` 进行更改。

模式

以下模式描述在 `ConnectDirectProcessDefinitions.xml` 文件中有效的元素。

```
<schema targetNamespace="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions"
  elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions">

  <element name="cdprocess" type="tns:cdprocessType"></element>

  <complexType name="cdprocessType">
    <sequence>
      <element name="processSet" type="tns:processSetType"
        minOccurs="0" maxOccurs="unbounded"></element>
    </sequence>
  </complexType>

  <complexType name="processSetType">
    <sequence>
      <element name="condition" type="tns:conditionType"
        minOccurs="0" maxOccurs="1" />
      <element name="process" type="tns:processType"
        minOccurs="1" maxOccurs="1" />
    </sequence>
  </complexType>

  <complexType name="conditionType">
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="match" type="tns:matchType" />
    </choice>
  </complexType>
```

```

        <element name="defined" type="tns:definedType" />
    </choice>
</complexType>

<complexType name="matchType">
    <attribute name="variable" type="string" use="required" />
    <attribute name="value" type="string" use="required" />
    <attribute name="pattern" type="tns:patternType" use="optional" />
</complexType>

<complexType name="definedType">
    <attribute name="variable" type="string" use="required" />
</complexType>

<complexType name="processType">
    <sequence>
        <element name="preTransfer" type="tns:transferType"
            minOccurs="0" maxOccurs="1" />
        <element name="transfer" type="tns:transferType"
            minOccurs="0" maxOccurs="1" />
        <element name="postTransferSuccess" type="tns:transferType"
            minOccurs="0" maxOccurs="1" />
        <element name="postTransferFailure" type="tns:transferType"
            minOccurs="0" maxOccurs="1" />
    </sequence>
</complexType>

<complexType name="transferType">
    <attribute name="process" type="string" use="required" />
</complexType>

<simpleType name="patternType">
    <restriction base="string">
        <enumeration value="regex" />
        <enumeration value="wildcard" />
    </restriction>
</simpleType>

</schema>
```

了解 ConnectDirectProcessDefinitions.xml 文件

下表中描述了 ConnectDirectProcessDefinitions.xml 文件中使用的元素和属性。

cdProcess

XML 文档的根元素。

processSet

包含所有有关用户定义的进程集的信息的组元素。

condition

组元素，包含测试传输所基于的条件，用于确定是否使用 processSet 元素中包含的进程集。

match

一种条件，测试变量的值是否与给定值匹配。

属性	描述
variable	指定变量。此变量的值将与 value 属性的值进行比较。变量是一个内置符号。有关更多信息，请参阅第 724 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』。
值	指定要与 variable 属性指定的变量值匹配的模式。
模式 (pattern)	指定用于 value 属性值的模式类型。pattern 属性的有效值为 - wildcard - 使用通配符 - regex -使用 Java 正则表达式 该属性为可选项，缺省值为 wildcard。

defined

一种条件，用于测试是否已定义变量。

属性	描述
variable	指定变量。如果该变量存在，那么将满足匹配条件。变量是一个内置符号。要了解更多信息，请参阅第 724 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』。

process

组元素，其中包含有关在找到匹配项时要调用的 Connect:Direct 进程的位置的信息。

transfer

要在传输请求期间调用的 Connect:Direct 进程。

属性	描述
进程	可选。指定包含要在传输请求期间调用的 Connect:Direct 进程的文件的名称。文件路径相对于 Connect:Direct 网桥代理配置目录。该属性是可选的，缺省值是使用由 MQMFT 生成的进程。对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。有关更多信息，请参阅第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』。

示例

在此示例中，有三个 processSet 元素。

第一个 processSet 元素指定如果传输请求具有值与模式 Client* 相匹配的 %FTESNODE 变量和值为 Admin 的 %FTESUSER 变量，那么 Connect:Direct 网桥代理会在传输过程中提交位于 agent_configuration_directory/AdminClient.cdp 中的 Connect:Direct 进程。

第二个 processSet 元素指定如果传输请求具有值与模式 Client* 匹配的 %FTESNODE 变量，那么 Connect:Direct 网桥代理会在传输过程中提交位于 agent_configuration_directory/Client.cdp 中的 Connect:Direct 进程。Connect:Direct 网桥代理按 processSet 元素的定义顺序读取这些元素，如果找到匹配项，那么将使用第一个匹配项，并且不会查找其他匹配项。对于与第一个和第二个 processSet 的条件都匹配的传输请求，Connect:Direct 网桥代理仅调用第一个 processSet 指定的进程。

第三个 processSet 元素没有任何条件，并且与所有传输都匹配。如果传输请求与第一个或第二个 processSet 的条件不匹配，那么 Connect:Direct 网桥代理将提交由第三个条件指定的 Connect:Direct 进程。在传输过程中，该进程位于 agent_configuration_directory/Default.cdp 中。

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:cdprocess xmlns:tns="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions
ConnectDirectProcessDefinitions.xsd">

  <tns:processSet>
    <tns:condition>
      <tns:match variable="%FTESNODE" value="Client*" pattern="wildcard" />
      <tns:match variable="%FTESUSER" value="Admin" pattern="wildcard" />
    </tns:condition>
    <tns:process>
      <tns:transfer process="AdminClient.cdp" />
    </tns:process>
  </tns:processSet>

  <tns:processSet>
    <tns:condition>
      <tns:match variable="%FTESNODE" value="Client*" pattern="wildcard" />
    </tns:condition>
    <tns:process>
      <tns:transfer process="Client.cdp" />
    </tns:process>
  </tns:processSet>

  <tns:processSet>
    <tns:process>
      <tns:transfer process="Default.cdp" />
    </tns:process>
  </tns:processSet>

</tns:cdprocess>
```


相关概念

第 288 页的『[Connect:Direct 网桥](#)』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关任务

第 212 页的『[使用 ConnectDirectProcessDefinition.xml 文件指定要启动的 Connect:Direct 进程](#)』

指定要在 IBM MQ Managed File Transfer 传输过程中启动的 Connect:Direct 进程。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以指定进程定义。

相关参考

第 469 页的『[fteCreateCDAgent（创建 Connect:Direct 网桥代理）](#)』

`fteCreateCDAgent` 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

第 724 页的『[IBM MQ Managed File Transfer 使用的正则表达式](#)』

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 Connect:Direct 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 `java.util.regex` API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

第 580 页的『[在 IBM MQ Managed File Transfer 属性中使用环境变量](#)』

从 IBM WebSphere MQ V7.5 起，可以在受管文件传输属性中使用表示文件或目录位置的环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化。例如，哪个用户在运行进程。

使用用户沙箱

您可以根据请求传输的 MQMD 用户名来限制可在其中传输文件的文件系统的区域。

当代理是协议网桥代理或 Connect:Direct 网桥代理时，不支持用户沙箱。

要启用用户沙箱，请您想要限制的代理的 `agent.properties` 文件添加以下属性：

```
userSandboxes=true
```

如果此属性存在并且设置为 `true`，那么代理程序会使用 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml` 文件中的信息来确定请求传输的用户可以访问文件系统的哪些部分。

`UserSandboxes.xml` XML 由 `<agent>` 元素组成，该元素包含零个或多个 `<sandbox>` 元素。这些元素描述哪些规则适用于哪些用户。`<sandbox>` 元素的 `user` 属性是一种模式，用于匹配请求的 MQMD 用户。

代理会定期重新装入 `UserSandboxes.xml` 文件，对该文件的任何有效更改都会影响代理的行为。缺省重新装入时间间隔为 30 秒。此时间间隔可通过在 `agent.properties` 文件中指定代理程序属性 `xmlConfigReloadInterval` 进行更改。

如果指定 `userPattern="regex"` 属性或值，那么 `user` 属性将解释为 Java 正则表达式。有关更多信息，请参阅第 724 页的『[IBM MQ Managed File Transfer 使用的正则表达式](#)』。

如果不指定 `userPattern="regex"` 属性或值，那么 `user` 属性会被解释为具有以下通配符的模式：

- 星号 (*)，表示零个或多个字符
- 问号 (?)，表示仅一个字符

将按照 `<sandbox>` 元素在文件中的列示顺序来执行匹配项。将仅使用第一个匹配项，将忽略文件中的所有其余潜在匹配项。如果文件中指定的 `<sandbox>` 元素与传输请求消息的关联 MQMD 用户均不匹配，那么传输将无法访问文件系统。如果在 MQMD 用户名与 `user` 属性之间发现匹配项，那么该匹配项将标识 `<sandbox>` 元素中应用于传输的一组规则。这组规则用于确定在传输过程中可以对哪些文件或数据集执行读或写操作。

每组规则都可以指定 `<read>` 元素（标识可以读取的文件）和一个 `<write>` 元素（标识可以写入的文件）。如果在规则集中省略 `<read>` 或 `<write>` 元素，那么假定相应地不允许与规则集相关联的用户执行任何读或写操作。

注: 在 UserSandboxes.xml 文件中, <read> 元素必须在 <write> 元素之前, <include> 元素必须在 <exclude> 元素之前。

每个 <read> 或 <write> 元素都包含一个或多个模式, 用来确定文件是否在沙箱中以及是否可以传输。请使用 <include> 和 <exclude> 元素来指定这些模式。<include> 或 <exclude> 元素的 name 属性指定要匹配的模式。可选属性 type 指定名称值是文件还是队列模式。如果未指定 type 属性, 那么代理程序会将该模式视为文件或目录路径模式。例如:

```
<tns:read>
  <tns:include name="/home/user/**"/>
  <tns:include name="USER.**" type="queue"/>
  <tns:exclude name="/home/user/private/**"/>
</tns:read>
```

代理会使用 <include> 和 <exclude> name 模式来确定可以对哪些文件、数据集或队列执行读或写操作。如果规范文件路径、数据集或队列名称与至少一个包含的模式匹配, 并且不与任何排除的模式匹配, 那么允许执行操作。使用 <include> 和 <exclude> 元素的 name 属性指定的模式会使用适用于平台 (在其上运行代理) 的路径分隔符和约定。如果您指定相对文件路径, 那么将相对于代理的 transferRoot 属性来解析该路径。

指定队列限制时, 支持语法 QUEUE@QUEUEMANAGER, 且规则如下:

- 如果条目中缺失 @ 字符, 那么该模式将被视为可在任何条目管理器上访问的队列名称。例如, 如果模式为 name, 那么该模式将同样被视为 name@**。
- 如果 @ 字符是条目中的第一个字符, 那么模式将视为队列管理器名称, 可以访问队列管理器上所有队列。例如, 如果模式为 @name, 那么该模式将同样被视为 **@name。

如果您将以下通配符指定为 <include> 和 <exclude> 元素的 name 属性的一部分, 那么这些通配符具有特殊含义:

单个星号与目录名称中的零个或零个以上字符匹配, 或者与数据集名称或队列名称的限定符中的零个或零个以上字符匹配。

?

问号只与目录名称中的一个字符匹配, 或者只与数据集名称或队列名称的限定符中的一个字符匹配。

两个星号字符与零个或零个以上目录名称匹配, 或者与数据集名称或队列名称中的零个或零个以上限定符匹配。此外, 以路径分隔符结尾的路径在路径末尾添加了隐式 "***"。因此, /home/user/ 与 /home/user/** 相同。

例如:

- /**/test/** 匹配路径中含有 test 目录的任何文件
- /test/file? 匹配 /test 目录中以 file 字符串开头并后跟任何单个字符的任何文件
- c:\test*.txt 匹配 c:\test 目录中扩展名为 .txt 的任何文件
- c:\test***.txt 与 'c:\test 目录中的任何文件或其某个具有 .txt 扩展名的子目录相匹配
- //'TEST.*.DATA' 匹配第一位限定符为 TEST 且第二位和第三位限定符为 DATA 的任何数据集。
- TEST.*.QUEUE@QM1 匹配队列管理器 QM1 上第一位限定符为 TEST 且第二位和第三位限定符为 QUEUE 的任何队列。

符号链接

您必须在 <include> 和 <exclude> 元素中指定硬链接, 以便完全解析在 UserSandboxes.xml 文件的文件路径中使用的任何符号链接。例如, 如果您有一个符号链接, 其中 /var 映射到 /SYSTEM/var, 那么必须将此路径指定为 <tns:include name="/SYSTEM/var"/>, 否则预期的传输将因用户沙箱安全性错误而失败。

IBM 4690 系统上的路径

有关如何在 IBM 4690 上解释 UserSandboxes.xml 文件中指定的路径的信息，请参阅第 84 页的『在 IBM 4690 系统上的沙箱中工作』。

示例

要允许具有 MQMD 用户名 guest 的用户从运行代理 AGENT_JUPITER 的系统上的 /home/user/public 目录或其任何子目录传输任何文件，请将以下 <sandbox> 元素添加到 AGENT_JUPITER 的配置目录中的文件 UserSandboxes.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
  xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
  <tns:agent>
    <tns:sandbox user="guest">
      <tns:read>
        <tns:include name="/home/user/public/**"/>
      </tns:read>
    </tns:sandbox>
  </tns:agent>
</tns:userSandboxes>
```

示例

要允许 MQMD 用户名为 account 并且后跟单个数字的任何用户（例如，account4），请完成以下操作：

- 从 /home/account 目录或其任何子目录中传输文件（运行代理 AGENT_SATURN 的系统上的 /home/account/private 目录除外）
- 在运行代理 AGENT_SATURN 的系统上，将任何文件传输到 /home/account/output 目录或其任何子目录。
- 从本地队列管理器上的队列中读取消息，以前缀 ACCOUNT. 开头，除非它以 ACCOUNT.PRIVATE. 开头（即具有第二个级别的 PRIVATE）。
- 将数据传输到任何队列管理器上以前缀 ACCOUNT.OUTPUT. 开头的队列。

将以下 <sandbox> 元素添加到文件 UserSandboxes.xml 的 AGENT_SATURN 配置目录中，

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
  xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
  <tns:agent>
    <tns:sandbox user="account[0-9]" userPattern="regex">
      <tns:read>
        <tns:include name="/home/account/**"/>
        <tns:include name="ACCOUNT.**" type="queue"/>
        <tns:exclude name="ACCOUNT.PRIVATE.**" type="queue"/>
        <tns:exclude name="/home/account/private/**"/>
      </tns:read>
      <tns:write>
        <tns:include name="/home/account/output/**"/>
        <tns:include name="ACCOUNT.OUTPUT.**" type="queue"/>
      </tns:write>
    </tns:sandbox>
  </tns:agent>
</tns:userSandboxes>
```

IBM MQ Managed File Transfer 的记录器配置属性

该记录器具有一组配置属性。在 logger.properties 文件中指定这些属性，该文件位于 MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name 目录中。

对于 IBM WebSphere MQ V7.5 或更高版本，可以在某些表示文件或目录位置的 Managed File Transfer 属性中使用环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化，例如哪个用户

在运行进程。有关更多信息，请参阅第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』。

注: 在 Windows 上指定文件路径时，反斜杠 (\) 分隔符必须以双反斜杠 (\\) (即，转义的 \) 的形式出现。或者，您也可以将单个正斜杠 (/) 用作分隔符。有关 Oracle 中 Java 属性文件中的字符转义的更多信息，请参阅属性类的 贾瓦多克。

属性名	描述	缺省值
wmqfte.logger.type	正在使用的记录器类型：文件或数据库。将该值设置为 FILE 或 DATABASE。	无缺省值
wmqfte.max.transaction.messages	落实事务之前在该事务中处理的消息的最大数目。在循环日志记录方式下，队列管理器可用于实时数据的空间容量是固定的。确保您将该属性设置为足够低的值，以便可用空间不会耗尽。	50
wmqfte.max.transaction.time	前后两次事务落实之间所经过的最大时间长度（毫秒）。	5000
wmqfte.max.consecutive.reject	可连续拒绝的最大消息数（即未遇到有效消息）。 如果超过该数量，那么记录器将认定问题不在于消息自身，而在于配置。例如，如果您使数据库中某个代理名称列的宽度小于所有代理名称，那么引用代理的所有消息都将被拒绝。	50
wmqfte.reject.queue.name	记录器将自己无法处理的消息放置到的队列的名称。如果您拥有数据库记录器，那么请参阅数据库记录器错误处理和拒绝以获取关于哪些消息可能会被置于此队列上的详细信息。	SYSTEM.FTE.LOG.RJCT.logger_name
wmqfte.command.queue.name	记录器从中读取控制其行为的命令消息的队列的名称。	SYSTEM.FTE.LOG.CMD.logger_name
wmqfte.queue.manager	记录器连接到的队列管理器（该队列管理器必须位于与记录器相同的机器上）。	无缺省值
wmqfte.message.source.type	以下某个值： automatic subscription 缺省值。记录器在 SYSTEM.FTE/Log/# 中定义的队列管理器上创建并使用其自己的持久的受管预订。这是适合大多数场景的值。 administrative subscription 如果自动预订不合适，那么您可以定义其他预订（例如，通过使用 IBM MQ Explorer、MQSC 或 PCF），并指示记录器使用该预订。例如，使用该值将日志空间分区，以便一个记录器处理代理 A 到 H，另一个记录器处理 I 到 P，第三个记录器处理 Q 到 Z。 queue 如果 IBM MQ 拓扑表示不方便创建记录器的预订，那么您可以改为使用队列。配置 IBM MQ，使该队列接收通常由协调队列管理器上的 SYSTEM.FTE/Log/# 预订接收的消息。	automatic subscription
wmqfte.message.source.name	如果消息源类型是 管理预订 或 队列 ，那么使用预订或队列的名称。如果源类型是 自动预订 ，那么忽略该属性。	无缺省值
wmqfte.database.credentials.file	包含用于连接至数据库的用户名和密码的文件。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。 有关更多信息，请参阅第 873 页的『MQMFT 凭证文件格式』。	此属性的缺省值为 %HOMEDRIVE% %HOMEPATH%\mqmftcredentials.xml" (在 Windows 上) 和 \$HOME/ MQMFTCredentials.xml (在其他平台 上)。

属性名	描述	缺省值
wmqfte.database.driver	数据库的 JDBC 驱动程序类的位置。这通常是 JAR 文件的路径和文件名。例如, Db2 on AIX 系统的 2 类驱动程序需要 /opt/IBM/db2/V9.5/java/db2jcc.jar 文件。在 Windows 系统上, 将路径分隔符指定为正斜杠字符 (/), 例如 C:/Program Files/IBM/SQLLIB/java/db2jcc.jar。在 z/OS 上, 指定 db2jcc.jar 文件的完整路径。例如, <code>wmqfte.database.driver=/db2/db2v10/jdbc/classes/db2jcc.jar</code>。 在 z/OS 系统上, 您必须引用以下所有 JAR 文件: • db2jcc.jar • db2jcc_license_cisuz.jar • db2jcc_javax.jar 如果数据库驱动程序包含多个 JAR 文件 (例如, Db2 V9.1 需要一个驱动程序 JAR 文件和一个许可证 JAR 文件), 请将所有这些 JAR 文件包含在此属性中。使用针对您平台的类路径分隔符 (即 Windows 系统上为分号字符 (;), 其他平台上为冒号字符 (:)) 来分隔多个文件名。	无缺省值
wmqfte.database.exclude_duplicate_metadata	控制是否在元数据表中存储条目来包含可在数据库记录器模式内的其他表中找到的信息。将此值设置为 <code>true</code> 或 <code>false</code>。缺省情况下, 不再存储这些元数据条目, 因为这与现有数据重复, 因此会浪费数据库存储容量。属性条目以及表 (其中显示相同数据) 如下所示: • com.ibm.wmqfte.SourceAgent TRANSFER_EVENT 或 CALL_REQUEST • com.ibm.wmqfte.DestinationAgent TRANSFER_EVENT • com.ibm.wmqfte.MqmdUser TRANSFER_EVENT 或 CALL_REQUEST • com.ibm.wmqfte.OriginatingUser TRANSFER_EVENT 或 CALL_REQUEST • com.ibm.wmqfte.OriginatingHost TRANSFER_EVENT 或 CALL_REQUEST • com.ibm.wmqfte.TransferId TRANSFER 或 CALL_REQUEST • com.ibm.wmqfte.JobName TRANSFER 或 CALL_REQUEST 将此属性的值设置为 <code>false</code> 会使这些元数据条目存储在元数据表中。	true
wmqfte.database.host	仅限 Db2: 对于 IBM WebSphere MQ V7.5 或更高版本, 是指要使用第 4 类 JDBC 驱动程序连接到的数据库服务器的主机名。如果指定了此属性的值, 那么还必须指定 <code>wmqfte.database.port</code> 的值。如果这两个属性都没有定义, 那么数据库记录器将使用缺省类型 2 JDBC 驱动程序进行连接。 如果指定了此属性的值, 那么必须具有此记录器的凭证文件 (文件路径由 <code>wmqfte.database.credentials.file</code> 属性定义), 并且必须能够访问该文件以定义用于连接至数据库的用户名和密码, 即使数据库位于本地系统上也是如此。	无缺省值
wmqfte.database.name	包含 IBM MQ Managed File Transfer 日志表的数据库实例 (或使用 Db2 for z/OS 时的子系统) 的名称。	无缺省值
wmqfte.database.type	正在使用的数据库管理系统: Db2 或 Oracle。将该值设置为 <code>db2</code> 或 <code>oracle</code> 。	db2

属性名	描述	缺省值
wmqfte.database.port	仅限 Db2: 对于 IBM WebSphere MQ V7.5 或更高版本, 是指要使用第 4 类 JDBC 驱动程序连接到的数据库服务器的端口号。如果指定了此属性的值, 那么还必须指定 <code>wmqfte.database.host</code> 的值。如果这两个属性都没有定义, 那么数据库记录器将使用缺省类型 2 JDBC 驱动程序进行连接。 如果指定了此属性的值, 那么必须具有此记录器的凭证文件 (文件路径由 <code>wmqfte.database.credentials.file</code> 属性定义), 并且必须能够访问该文件以定义用于连接至数据库的用户名和密码, 即使数据库位于本地系统上也是如此。	无缺省值
wmqfte.database.schema	包含 IBM MQ Managed File Transfer 日志记录表的数据库模式。在大多数情况下缺省值都适用, 但您可能需要出于自身特定于站点的数据库考虑, 指定其他值。	FTELOG
wmqfte.database.native.library.path	包含所选数据库驱动程序 (如果有) 所需的本机库的路径。例如, Db2 on AIX 系统的 2 类驱动程序需要 <code>/opt/IBM/db2/V9.5/lib32/</code> 中的库。作为该属性的替代方案, 您可以使用其他方法设置 <code>java.library.path</code> 系统属性。 在 Solaris 和 HP-UX 系统上, 运行 <code>fteStartLogger</code> 命令之前, 还必须设置并导出 <code>LD_LIBRARY_PATH</code> 环境变量以包含该路径。	无缺省值
wmqfte.file.logger.fileDirectory	文件记录器日志文件所在的目录。	<code>mqft/logs/coordination_dir/loggers/logger_name/logs</code>
wmqfte.file.logger.fileSize	允许一个日志文件增大到的最大大小。该大小值是大于零的正整数, 后跟以下某个单位: KB、MB、GB、m (分钟)、h (小时)、d (天)、w (周)。例如, <code>wmqfte.file.logger.fileSize=5MB</code> 指定最大文件大小为 5MB。 <code>wmqfte.file.logger.fileSize=2d</code> 指定最大文件大小为 2 天的数据。	10MB
wmqfte.file.logger.fileCount	要创建的日志文件的最大数目。数据量超过此数目的文件中可以存储的最大量时, 将删除最旧的文件, 以便文件数从不超过指定的值。	3

属性名	描述	缺省值
wmqfte.file.logger.mode	正在使用的记录器方式：circular 或 linear。将该值设置为 CIRCULAR 或 LINEAR。 循环-文件记录器将信息写入文件，直到该文件达到使用 wmqfte.file.logger.fileSize 属性定义的最大大小为止。达到最大大小时，文件记录器将启动新文件。在此方式下写入的最大文件数由使用 wmqfte.file.logger.fileCount 属性定义的值控制。达到该最大文件数时，文件记录器将删除首个文件并重新创建该文件以用作当前活动文件。如果 wmqfte.file.logger.fileSize 属性中定义的值是固定大小的字节单位（例如，KB、MB 或 GB），那么该方式下使用的磁盘空间的上限等于 fileSize 乘以 fileCount。如果 wmqfte.file.logger.fileSize 属性中定义的值是时间单位（例如，m、h、d 或 w），那么最大大小取决于这些时间段内系统中日志消息的吞吐量。在此方式下运行时使用的日志文件命名约定为：logger_namenumber-timestamp.log，其中： - logger_name 是在 fteCreateLogger 命令中为记录器指定的名称。 - number 是文件在集合内的编号。 - timestamp 是创建文件时的时间戳记。 例如，LOGGER1-20111216123430147.log LINEAR - 文件记录器将信息写入文件，直至该文件达到其最大大小（使用 wmqfte.file.logger.fileSize 属性定义）。达到最大大小时，文件记录器将启动新文件。先前已写入信息的文件将不会删除，从而允许将其保留为日志消息的历史记录。以 线性 方式运行时，不会删除文件，因此将忽略 wmqfte.file.logger.fileCount 属性，因为可以创建的文件数没有上限。因为在该方式下运行时没有上限，所以有必要跟踪日志文件所用的磁盘空间量以避免磁盘空间不足。在此方式下运行时使用的日志文件命名约定为：logger_name-timestamp.log，其中： - logger_name 是在 fteCreateLogger 命令中为记录器指定的名称。 - timestamp 是创建文件时的时间戳记。 例如，LOGGER-20111216123430147.log	无缺省值
wmqfte.max.retry.interval	记录器遇到持续性错误时，两次重试之间的最长时间（秒）。 某些错误情况（例如，丢失数据库连接）会使记录器无法继续运行。当出现这种情况时，记录器将回滚当前事务，等待一段时间，然后重试。记录器等待的时间最初很短，这样短暂错误可以快速得以解决。但是，记录器每次重试时，其等待的时间都会延长。这可防止在错误情况持续较长时间时（例如，数据库关闭进行维护时）执行过多的不必要工作。 使用此属性可设置等待时间长度的限制，以便在解决错误情况的合理时间内进行重试。	600
loggerQMgrRetryInterval	记录器的进程控制器对队列管理器的可用性进行检查的时间间隔（秒）。	30
maxRestartCount	在 maxRestartInterval 属性值所指定的时间间隔内可发生的最大重新启动次数。当超出该值时，记录器的进程控制器会停止重新启动记录器，而改为执行基于 maxRestartDelay 属性值的操作。	4

属性名	描述	缺省值
maxRestartInterval	一段时间间隔（秒），记录器的进程控制器度量此时间间隔内记录器的重新启动次数。如果该时间间隔内的重新启动次数超过 maxRestartCount 属性的值，那么记录器的进程控制器会停止重新启动记录器。记录器的进程控制器会改为执行基于 maxRestartDelay 属性值的操作。	120
maxRestartDelay	确定当记录器重新启动的频率超过 maxRestartCount 和 maxRestartInterval 属性值时，记录器的进程控制器的行为。如果指定小于或等于零的值，那么记录器的进程控制器会停止。如果指定大于零的值，那么这是重置记录器的进程控制器保存的重新启动历史记录信息并重新启动记录器之前等待的秒数。	-1
wmqfte.oracle.port	记录器用于连接到 Oracle 实例的端口。此端口也称为 TNS 侦听器。	1521
wmqfte.oracle.host	记录器用于连接到 Oracle 实例的主机。	localhost
armELEMTYPE	可选属性。如果 Automatic Restart Manager (ARM) 将记录器配置为重新启动，那么请将此属性设置为关联的 ARM 策略中指定的 ARM ELEMTYPE 参数值。对于记录器，请将 ELEMTYPE 设置为 SYSBFGLG。	未设定
armELEMENT	可选属性。如果 Automatic Restart Manager (ARM) 将记录器配置为重新启动，那么请将此属性设置为关联的 ARM 策略中指定的 ARM ELEMENT 参数值。您可以将 ELEMENT 值设置为与记录器名称相对应。	未设定
loggerQMGrAuthenticationCredentialsFile	包含用于连接到记录器的协调队列管理器的 MQ 连接凭证的文件的路径。	此属性的缺省值为 %HOMEDRIVE%\%HOMEPATH%\mqmftcredentials.xml (在 Windows 上) 和 \$HOME/MQMFTCredentials.xml (在其他平台上)。
跟踪	可选属性。要在记录器启动时启用跟踪的情况下运行记录器时的跟踪规范。跟踪规范是以逗号分隔的类、等于字符以及跟踪级别列表。 例如，com.ibm.wmqfte.databaseslogger、com.ibm.wmqfte.databaseslogger.operation=all。 您可以在冒号分隔的列表中指定多个跟踪规范。例如， com.ibm.wmqfte.databaseslogger=moderate: com.ibm.wmqfte.databaseslogger.operation=all	None
traceFiles	可选属性。保存的跟踪文件的总数。该值适用于记录器的进程控制器以及记录器自身。	5
traceSize	可选属性。将跟踪合并到下一个文件之前，每个跟踪文件的最大大小（MB）。该值适用于记录器的进程控制器以及记录器自身。	20

相关参考

第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』

从 IBM WebSphere MQ V7.5 起，可以在受管文件传输属性中使用表示文件或目录位置的环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化。例如，哪个用户在运行进程。

Java 系统属性

必须将许多 IBM MQ Managed File Transfer 命令和代理程序属性定义为 Java 系统属性，因为它们定义了无法使用命令或代理程序属性机制的早期功能的配置。

通过定义环境变量 `BFG_JVM_PROPERTIES`，为要运行 IBM MQ Managed File Transfer 命令的 JVM 定义系统属性和其他 JVM 选项。例如，要在 UNIX 类平台上设置 `com.ibm.wmqfte.maxConsoleLineLength` 属性，请将该变量定义如下：

```
export BFG_JVM_PROPERTIES="-Dcom.ibm.wmqfte.maxConsoleLineLength=132"
```

如果要将代理程序作为 Windows 服务运行，那么可以通过在 `fteModifyAgent` 命令上指定 `-sj` 参数来修改代理程序的 Java 系统属性。

表 52: Java 系统属性		
属性名	描述	值
<code>com.ibm.wmqfte.maxConsoleLineLength</code>	可以写入控制台的最大行长度。超过该长度的行会自动换行。该值用字节表示（不是字符）。	IBM i 的缺省长度为 132 字节。对于所有其他平台，长度没有限制。
<code>com.ibm.wmqfte.daemon.windows.windowsServiceLogFilesm</code>	（仅限 Windows。）指定要保存的最大 Windows 服务日志文件数。当这些应用程序作为 Windows 服务运行时，会在代理和数据库记录器日志目录中创建 Windows 服务日志文件。Windows 服务日志文件以 <code>service</code> 为前缀命名，包含关于服务启动和停止的消息。	5

相关概念

第 116 页的『分布式平台上的配置选项』

IBM MQ Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。这些属性文件位于安装产品时所定义的配置目录中。

第 392 页的『使用 IBM MQ Managed File Transfer 的提示与技巧』

以下建议可帮助您最有效地使用 IBM MQ Managed File Transfer：

SSL 属性

将 SSL 或 TLS 与 IBM MQ 和 IBM MQ Managed File Transfer 一起使用可防止在代理与队列管理器之间建立未经授权的连接，并可对代理与队列管理器之间的消息通信进行加密。

要了解有关将 SSL 与 IBM MQ Managed File Transfer 一起使用的信息，请参阅第 103 页的『为 IBM MQ Managed File Transfer 配置 SSL 或 TLS 加密』。

对于 IBM WebSphere MQ V7.5 或更高版本，可以在某些表示文件或目录位置的 Managed File Transfer 属性中使用环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化，例如哪个用户在运行进程。有关更多信息，请参阅第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』。

表 53: <i>agent.properties</i> 文件的 SSL 属性		
属性名	描述	缺省值
<code>agentSslCipherSpec</code>	指定在代理和代理队列管理器之间交换数据时，使用的协议、散列算法和加密算法，以及加密密钥中使用的位数。 <code>agentSslCipherSpec</code> 的值为 CipherSpec 名称。该 CipherSpec 名称与代理队列管理器通道中使用的 CipherSpec 名称相同。有效 CipherSpec 名称的列表包含在 IBM MQ Java 类中的 SSL/TLS CipherSpecs 和 CipherSuites 和针对 JMS 的 IBM MQ 类中的 SSL/TLS CipherSpecs 和 CipherSuites 中。 <code>agentSslCipherSpec</code> 类似于 <code>agentSslCipherSuite</code>。如果指定了 <code>agentSslCipherSuite</code> 和 <code>agentSslCipherSpec</code>，那么使用 <code>agentSslCipherSpec</code> 的值。	None

表 53: agent.properties 文件的 SSL 属性 (继续)

属性名	描述	缺省值
agentSslCipherSuite	指定代理程序和代理程序队列管理器交换数据方式的 SSL 方面。 agentSslCipherSuite 的值为 CipherSuite 名称。CipherSuite 名称将映射到代理队列管理器通道中使用的 CipherSpec 名称。有关更多信息, 请参阅 CipherSuite 和 CipherSpec 名称映射。 agentSslCipherSuite 类似于 agentSslCipherSpec。如果指定了 agentSslCipherSuite 和 agentSslCipherSpec, 那么使用 agentSslCipherSpec 的值。	None
agentSslPeerName	指定必须与代理队列管理器提供的名称相匹配的专有名称框架。专有名称用于检查连接时队列管理器提供的标识证书。	None
agentSslTrustStore	指定代理程序信任的证书的位置。agentSslTrustStore 的值为文件路径。如果它是 Windows 文件路径, 那么反斜杠字符 (\) 必须转义为 (\\)。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。	None
agentSslKeyStore	指定代理程序专用密钥的位置。agentSslKeyStore 的值为文件路径。如果是 Windows 文件路径, 那么必须对反斜杠字符 (\) 进行转义 (\\)。仅当代理队列管理器需要客户机认证时, 才需要此属性。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。	None
agentSslFipsRequired	指定要在代理级别启用 FIPS 支持。该属性的值可以为 true 或 false。有关更多信息, 请参阅 第 730 页的『FIPS 支持』 。	false
agentSslKeyStoreType	您要使用的 SSL 密钥库的类型。支持 JKS 和 PKCS#12 密钥库。该属性的值可以为 jks 或 pkcs12。	jks
agentSslKeyStoreCredentialsFile	包含 agentSslKeyStore 凭证的文件的文件的路径。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。	此属性的缺省值为 %USERPROFILE%\MQMFTCredentials.xml (在 Windows 上), f:/adxetc/mft75/mqft/config/mqmftcredentials.xml (在 IBM 4690 上) 和 \$HOME/MQMFTCredentials.xml (在其他平台上)。
agentSslTrustStoreType	您要使用的 SSL 密钥库的类型。支持 JKS 和 PKCS#12 密钥库。该属性的值可以为 jks 或 pkcs12。	jks
agentSslTrustStoreCredentialsFile	包含 agentSslTrustStore 的凭证的文件的文件的路径。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。	此属性的缺省值为 %USERPROFILE%\MQMFTCredentials.xml (在 Windows 上), f:/adxetc/mft75/mqft/config/mqmftcredentials.xml (在 IBM 4690 上) 和 \$HOME/MQMFTCredentials.xml (在其他平台上)。

表 54: *coordination.properties* 文件的 SSL 属性

属性名	描述	缺省值
coordinationSslCipherSpec	指定在命令和协调队列管理器之间交换数据时，使用的协议、散列算法和加密算法，以及加密密钥中使用的位数。 coordinationSslCipherSpec 的值为 CipherSpec 名称。该 CipherSpec 名称与协调队列管理器通道中使用的 CipherSpec 名称相同。有效 CipherSpec 名称的列表包含在 SSL/TLS CipherSpecs 和 CipherSuites (在 IBM MQ classes for Java 中) 以及 SSL/TLS CipherSpecs 和 CipherSuites (在 IBM MQ classes for JMS 中) 中。 coordinationSslCipherSpec 类似于 coordinationSslCipherSuite。如果指定了 coordinationSslCipherSuite 和 coordinationSslCipherSpec，那么使用 coordinationSslCipherSpec 的值。	None
coordinationSslCipherSuite	指定命令和协调队列管理器交换数据方式的 SSL 方面。 coordinationSslCipherSuite 的值为 CipherSuite 名称。CipherSuite 名称将映射到代理队列管理器通道中使用的 CipherSpec 名称。有关更多信息，请参阅 CipherSuite 和 CipherSpec 名称映射。 coordinationSslCipherSuite 类似于 coordinationSslCipherSpec。如果指定了 coordinationSslCipherSuite 和 coordinationSslCipherSpec，那么使用 coordinationSslCipherSpec 的值。	None
coordinationSslPeerName	指定必须与协调队列管理器提供的名称相匹配的专有名称框架。专有名称用于检查连接时协调队列管理器提供的标识证书。	None
coordinationSslTrustStore	指定命令信任的证书的位置。coordinationSslTrustStore 的值为文件路径。如果它是 Windows 文件路径，那么反斜杠字符 (\) 必须转义为 (\\)。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。	None
coordinationSslTrustStoreType	您要使用的 SSL 密钥库的类型。支持 JKS 和 PKCS#12 密钥库。该属性的值可以为 jks 或 pkcs12。	jks
coordinationSslTrustStoreCredentialsFile	包含 coordinationSslTrustStore 凭证的文件的文件的路径。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。	此属性的缺省值为 %USERPROFILE%\MQMFTCredentials.xml (在 Windows 上)，f:/adxetc/mft75/mqft/config/mqmftcredentials.xml (在 IBM 4690 上) 和 \$HOME/MQMFTCredentials.xml (在其他平台上)。
coordinationSslKeyStore	指定命令专用密钥的位置。coordinationSslKeyStore 的值为文件路径。如果是 Windows 文件路径，那么必须对反斜杠字符 (\) 进行转义 (\\)。仅当协调队列管理器需要客户机认证时，才需要此属性。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。	None
coordinationSslKeyStoreType	您要使用的 SSL 密钥库的类型。支持 JKS 和 PKCS#12 密钥库。该属性的值可以为 jks 或 pkcs12。	jks

表 54: <i>coordination.properties</i> 文件的 SSL 属性 (继续)		
属性名	描述	缺省值
coordinationSslKeyStoreCredentialsFile	包含 coordinationSslKeyStore 凭证的文件的路径。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。	此属性的缺省值为 %USERPROFILE%\MQMFTCredentials.xml (在 Windows 上), f:/adxetc/mft75/mqft/config/mqmftcredentials.xml (在 IBM 4690 上) 和 \$HOME/MQMFTCredentials.xml (在其他平台上)。
coordinationSslFipsRequired	指定要在协调队列管理器级别启用 FIPS 支持。该属性的值可以为 true 或 false。有关更多信息, 请参阅第 730 页的『FIPS 支持』。	false

表 55: <i>command.properties</i> 文件的 SSL 属性		
属性名	描述	缺省值
connectionSslCipherSpec	指定在命令和命令队列管理器之间交换数据时, 使用的协议、散列算法和加密算法, 以及加密密钥中使用的位数。 connectionSslCipherSpec 的值为 CipherSpec 名称。该 CipherSpec 名称与命令队列管理器通道中使用的 CipherSpec 名称相同。有效 CipherSpec 名称的列表包含在 SSL/TLS CipherSpecs 和 CipherSuites (在 IBM MQ classes for Java 中) 以及 SSL/TLS CipherSpecs 和 CipherSuites (在 IBM MQ classes for JMS 中) 中。 connectionSslCipherSpec 类似于 connectionSslCipherSuite。如果指定了 connectionSslCipherSuite 和 connectionSslCipherSpec, 那么使用 connectionSslCipherSpec 的值。	None
connectionSslCipherSuite	指定命令和命令队列管理器交换数据方式的 SSL 方面。 connectionSslCipherSuite 的值为 CipherSuite 名称。CipherSuite 名称将映射到代理队列管理器通道中使用的 CipherSpec 名称。有关更多信息, 请参阅 CipherSuite 和 CipherSpec 名称映射。 connectionSslCipherSuite 类似于 connectionSslCipherSpec。如果指定了 connectionSslCipherSuite 和 connectionSslCipherSpec, 那么使用 connectionSslCipherSpec 的值。	None
connectionSslPeerName	指定必须与命令队列管理器提供的名称相匹配的专有名称框架。专有名称用于检查连接时命令队列管理器提供的标识证书。	None
connectionSslTrustStore	指定命令信任的证书的位置。connectionSslTrustStore 的值为文件路径。如果它是 Windows 文件路径, 那么反斜杠字符 (\) 必须转义为 (\\)。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。	None
connectionSslTrustStoreType	您要使用的 SSL 信任库的类型。支持 JKS 和 PKCS#12 密钥库。该属性的值可以为 jks 或 pkcs12。	jks
connectionSslTrustStoreCredentialsFile	包含 connectionSslTrustStore 凭证的文件的路径。 对于 IBM WebSphere MQ V7.5 或更高版本, 此属性的值可以包含环境变量。	此属性的缺省值为 %USERPROFILE%\MQMFTCredential s.xml (在 Windows 上), f:/adxetc/mft75/mqft/config/mqmftcredentials.xml (在 IBM 4690 上) 和 \$HOME/MQMFTCredentials.xml (在其他平台上)。

表 55: <i>command.properties</i> 文件的 SSL 属性 (继续)		
属性名	描述	缺省值
connectionSslKeyStore	指定命令专用密钥的位置。connectionSslKeyStore 的值为文件路径。如果是 Windows 文件路径，那么必须对反斜杠字符 (\) 进行转义 (\\)。仅当命令队列管理器需要客户机认证时，才需要此属性。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。	None
connectionSslKeyStoreType	您要使用的 SSL 密钥库的类型。支持 JKS 和 PKCS#12 密钥库。该属性的值可以为 jks 或 pkcs12。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。	jks
connectionSslKeyStoreCredentialsFile	包含 connectionSslKeyStore 凭证的文件的完整路径。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。	此属性的缺省值为 %USERPROFILE%\MQMFTCredentials.xml (在 Windows 上)，f:/adxetc/mft75/mqft/config/mqmftcredentials.xml (在 IBM 4690 上) 和 \$HOME/MQMFTCredentials.xml (在其他平台上)。
connectionSslFipsRequired	指定要在命令队列管理器级别启用 FIPS 支持。该属性的值可以为 true 或 false。有关更多信息，请参阅第 730 页的『FIPS 支持』。	false

相关概念

第 116 页的『分布式平台上的配置选项』

IBM MQ Managed File Transfer 提供了一组执行操作所必需的属性文件，其中包含有关设置的重要信息。这些属性文件位于安装产品时所定义的配置目录中。

相关参考

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 agent.properties，其中必须包含代理用于连接到其队列管理器的信息。agent.properties 文件还可以包含用于更改代理程序行为的属性。

第 588 页的『command.properties 文件』

command.properties 文件指定发出命令时要连接的命令队列管理器以及 IBM MQ Managed File Transfer 联系该队列管理器所需的信息。

第 584 页的『coordination.properties 文件』

coordination.properties 文件指定到协调队列管理器的连接详细信息。由于多个 IBM MQ Managed File Transfer 安装可能共享同一个协调队列管理器，因此您可以使用指向共享驱动器上的公共 coordination.properties 文件的符号链接。

第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』

从 IBM WebSphere MQ V7.5 起，可以在受管文件传输属性中使用表示文件或目录位置的环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化。例如，哪个用户在运行进程。

SHA-2 CipherSpecs 和 CipherSuites

IBM MQ Managed File Transfer 支持 SHA-2 CipherSpecs 和 CipherSuites。

要在 IBM MQ V8 中对代理程序与 IBM MQ 队列管理器之间的连接启用 SHA-2 CipherSpecs 和 CipherSuites，必须使用 IBM JRE 6.0 SR13 FP2，7.0 SR4 FP2 或更高版本。

要在 IBM MQ Managed File Transfer V8 中启用 SHA-2 CipherSpecs 和 CipherSuites，以使用 FTPS 方式的协议网桥连接到 FTPS 服务器，必须使用 IBM JRE 6.0 SR13 FP2，7.0 SR4 FP2 或更高版本。

有关可用于代理和 IBM MQ 队列管理器之间连接的 CipherSpec 和 CipherSuite 的更多信息，请参阅 [SSL CipherSpec 和 CipherSuite](#)。

有关配置 CipherSpecs 和 CipherSuites 以使用网桥协议代理和 FTPS 服务器的更多信息，请参阅第 728 页的『[协议网桥提供的 FTPS 服务器支持](#)』和第 611 页的『[协议网桥属性文件格式](#)』。

不支持与 4690 OS 的 SHA-2 连接。

IBM i JVM 不支持在 MQ 8.0 中指定 CipherSpecs 中详细描述的较新密码。因此，针对 IBM i 平台的 SHA-2 支持仅包含在 MQ 7.5 中指定 CipherSpecs 中详细介绍的密码。

如果要符合 SP 800-131A 标准，必须满足以下要求：

- 必须使用正确配置的 FTPS；不支持 SFTP。
- 远程服务器必须仅发送符合 SP 800-131A 标准的密码套件。

相关参考

第 634 页的『[SSL 属性](#)』

将 SSL 或 TLS 与 IBM MQ 和 IBM MQ Managed File Transfer 一起使用可防止在代理与队列管理器之间建立未经授权的连接，并可对代理与队列管理器之间的消息通信进行加密。

文件记录器配置文件

除了 logger.properties 文件，独立文件记录器的配置目录中还有 XML 配置文件。此配置文件称为 FileLoggerFormat.xml，它定义文件记录器用于将消息写入日志文件的格式。此文件的内容必须符合 FileLoggerFormat.xsd 文件中定义的 XML 模式。

相关参考

第 163 页的『[IBM MQ Managed File Transfer 的记录器配置属性](#)』

该记录器具有一组配置属性。在 logger.properties 文件中指定这些属性，该文件位于 MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name 目录中。

第 639 页的『[独立文件记录器缺省日志格式定义](#)』

独立文件记录器的缺省日志文件格式定义

第 644 页的『[独立文件记录器格式 XSD](#)』

独立文件格式的模式。

第 154 页的『[独立文件记录器格式](#)』

文件记录器写入的消息信息的格式可在 FileLoggerFormat.xml 文件中定义。

独立文件记录器缺省日志格式定义

独立文件记录器的缺省日志文件格式定义

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="1.00" xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <callCompleted>
      <format>
        <inserts>
          <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
          <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6" ignoreNull="false">type</insert>
          <insert type="user" width="3" ignoreNull="false">/transaction/status/@resultCode</insert>
          <insert type="user" width="0" ignoreNull="false">/transaction/agent/@agent</insert>
          <insert type="user" width="0" ignoreNull="false">/transaction/agent/@QMGr</insert>
          <insert type="user" width="0" ignoreNull="false">/transaction/job/name</insert>
          <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/call/command/
@type</insert>
          <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/call/command/
@name</insert>
          <insert type="system" width="0" ignoreNull="true">callArguments</insert>
          <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/call/callResult/
@outcome</insert>
          <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/call/callResult/
result/error</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </callCompleted>
```

```

<callStarted>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/agent/@agent</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/agent/@QMgr</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/job/name</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/call/command/
@type</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/call/command/
@name</insert>
      <insert type="system" width="0" ignoreNull="true">callArguments</insert>
    </inserts>
    <separator>;</separator>
  </format>
</callStarted>
<monitorAction>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/monitorLog/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/monitorLog/@referenceId</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="false">/monitorLog/status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/@monitorName</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@agent</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@QMgr</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/action</insert>
    </inserts>
    <separator>;</separator>
  </format>
</monitorAction>
<monitorCreate>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/monitorLog/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/monitorLog/@referenceId</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/@monitorName</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@agent</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@QMgr</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/action</insert>
    </inserts>
    <separator>;</separator>
  </format>
</monitorCreate>
<monitorFired>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/monitorLog/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/monitorLog/@referenceId</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="false">/monitorLog/status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/@monitorName</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@agent</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@QMgr</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/action</insert>
      <insert type="user" width="48" ignoreNull="false">/monitorLog/references/taskRequest</insert>
    </inserts>
    <separator>;</separator>
  </format>
</monitorFired>
<notAuthorized>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/notAuthorized/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/notAuthorized/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="false">/notAuthorized/status/@resultCode</insert>
      <insert type="user" width="12" ignoreNull="false">/notAuthorized/action</insert>
      <insert type="user" width="12" ignoreNull="false">/notAuthorized/authority</insert>
      <insert type="user" width="0" ignoreNull="false">/notAuthorized/originator/userID</insert>
      <insert type="user" width="0" ignoreNull="false">/notAuthorized/status/supplement</insert>
    </inserts>
    <separator>;</separator>
  </format>
</notAuthorized>
<scheduleDelete>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>

```

```

        <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
        <insert type="system" width="6" ignoreNull="false">type</insert>
        <insert type="user" width="3" ignoreNull="false">/schedulelog/status/@resultCode</insert>
        <insert type="user" width="0" ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
        <insert type="user" width="12" ignoreNull="false">/schedulelog/action</insert>
        <insert type="user" width="0" ignoreNull="false">/schedulelog/originator/userID</insert>
        <insert type="user" width="0" ignoreNull="true">/schedulelog/status/supplement</insert>
    </inserts>
    <separator>;</separator>
</format>
</scheduleDelete>
<scheduleExpire>
    <format>
        <inserts>
            <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
            <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
            <insert type="system" width="6" ignoreNull="false">type</insert>
            <insert type="user" width="3" ignoreNull="false">/schedulelog/status/@resultCode</insert>
            <insert type="user" width="0" ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
            <insert type="user" width="12" ignoreNull="false">/schedulelog/action</insert>
            <insert type="user" width="0" ignoreNull="false">/schedulelog/originator/userID</insert>
            <insert type="user" width="0" ignoreNull="true">/schedulelog/status/supplement</insert>
        </inserts>
        <separator>;</separator>
    </format>
</scheduleExpire>
<scheduleSkipped>
    <format>
        <inserts>
            <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
            <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
            <insert type="system" width="6" ignoreNull="false">type</insert>
            <insert type="user" width="3" ignoreNull="false">/schedulelog/status/@resultCode</insert>
            <insert type="user" width="0" ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
            <insert type="user" width="12" ignoreNull="false">/schedulelog/action</insert>
            <insert type="user" width="0" ignoreNull="false">/schedulelog/originator/userID</insert>
            <insert type="user" width="0" ignoreNull="true">/schedulelog/status/supplement</insert>
        </inserts>
        <separator>;</separator>
    </format>
</scheduleSkipped>
<scheduleSubmitInfo>
    <format>
        <inserts>
            <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
            <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
            <insert type="system" width="6" ignoreNull="false">type</insert>
            <insert type="user" width="3" ignoreNull="false">/schedulelog/status/@resultCode</insert>
            <insert type="user" width="0" ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
            <insert type="user" width="12" ignoreNull="false">/schedulelog/action</insert>
            <insert type="user" width="0" ignoreNull="false">/schedulelog/originator/userID</insert>
            <insert type="user" width="0" ignoreNull="true">/schedulelog/schedule/submit</insert>
            <insert type="user" width="0" ignoreNull="true">/schedulelog/schedule/submit/@timezone</
insert>
            <insert type="user" width="3" ignoreNull="true">/schedulelog/schedule/repeat/frequency</
insert>
            <insert type="user" width="12" ignoreNull="true">/schedulelog/schedule/repeat/frequency/
@interval</insert>
            <insert type="user" width="3" ignoreNull="true">/schedulelog/schedule/repeat/expireCount</
insert>
            <insert type="user" width="0" ignoreNull="true">/schedulelog/status/supplement</insert>
        </inserts>
        <separator>;</separator>
    </format>
</scheduleSubmitInfo>
<scheduleSubmitTransfer>
    <format>
        <inserts>
            <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
            <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
            <insert type="system" width="10" ignoreNull="false">type</insert>
            <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
/transaction/sourceWebUser/@webGatewayAgentName |
/transaction/sourceWebGateway/@webGatewayAgentName</insert>
            <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@QMGr |
/transaction/sourceWebUser/@webGatewayAgentQMGr |
/transaction/sourceWebGateway/@webGatewayAgentQMGr</insert>
            <insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@agent |
/transaction/destinationWebUser/@webGatewayAgentName |
/transaction/destinationWebGateway/@webGatewayAgentName</insert>
            <insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@QMGr |
/transaction/destinationWebUser/@webGatewayAgentQMGr |

```

```

    /transaction/destinationWebGateway/@webGatewayAgentQMGr</insert>
  </inserts>
  <separator>;</separator>
</format>
</scheduleSubmitTransfer>
<scheduleSubmitTransferSet>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
      <insert type="system" width="10" ignoreNull="false">type</insert>
      <insert type="user" width="0" ignoreNull="false">source/file | source/queue</insert>
      <insert type="user" width="5" ignoreNull="true">source/@type</insert>
      <insert type="user" width="6" ignoreNull="true">source/@disposition</insert>
      <insert type="user" width="0" ignoreNull="false">destination/file | destination/queue</
insert>
      <insert type="user" width="5" ignoreNull="true">destination/@type</insert>
      <insert type="user" width="9" ignoreNull="true">destination/@exist</insert>
    </inserts>
    <separator>;</separator>
  </format>
</scheduleSubmitTransferSet>
<transferStarted>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="true">/transaction/status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
/transaction/sourceWebUser/@webGatewayAgentName |
/transaction/sourceWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@QMGr |
/transaction/sourceWebUser/@webGatewayAgentQMGr |
/transaction/sourceWebGateway/@webGatewayAgentQMGr</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@agentType |
/transaction/sourceWebUser/@webGatewayAgentType |
/transaction/sourceWebGateway/@webGatewayAgentType</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@agent |
/transaction/destinationWebUser/@webGatewayAgentName |
/transaction/destinationWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/destinationAgent/@QMGr |
/transaction/destinationWebUser/@webGatewayAgentQMGr |
/transaction/destinationWebGateway/@webGatewayAgentQMGr</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/originator/userID</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/job/name</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/scheduleLog/@ID</insert>
    </inserts>
    <separator>;</separator>
  </format>
</transferStarted>
<transferCancelled>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="true">/transaction/status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
/transaction/sourceWebUser/@webGatewayAgentName |
/transaction/sourceWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@QMGr |
/transaction/sourceWebUser/@webGatewayAgentQMGr |
/transaction/sourceWebGateway/@webGatewayAgentQMGr</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@agentType |
/transaction/sourceWebUser/@webGatewayAgentType |
/transaction/sourceWebGateway/@webGatewayAgentType</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@agent |
/transaction/destinationWebUser/@webGatewayAgentName |
/transaction/destinationWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/destinationAgent/@QMGr |
/transaction/destinationWebUser/@webGatewayAgentQMGr |
/transaction/destinationWebGateway/@webGatewayAgentQMGr</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/destinationAgent/@agentType |
/transaction/destinationWebUser/@webGatewayAgentType |
/transaction/destinationWebGateway/@webGatewayAgentType</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/originator/userID</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/job/name</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/status/supplement</insert>
    </inserts>
    <separator>;</separator>
  </format>

```

```

</transferCancelled>
<transferComplete>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="true">/transaction/status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
        /transaction/sourceWebUser/@webGatewayAgentName |
        /transaction/sourceWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@QMGr |
        /transaction/sourceWebUser/@webGatewayAgentQMGr |
        /transaction/sourceWebGateway/@webGatewayAgentQMGr</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@agentType |
        /transaction/sourceWebUser/@webGatewayAgentType |
        /transaction/sourceWebGateway/@webGatewayAgentType</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@agent |
        /transaction/destinationWebUser/@webGatewayAgentName |
        /transaction/destinationWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/destinationAgent/@QMGr |
        /transaction/destinationWebUser/@webGatewayAgentQMGr |
        /transaction/destinationWebGateway/@webGatewayAgentQMGr</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/destinationAgent/@agentType |
        /transaction/destinationWebUser/@webGatewayAgentType |
        /transaction/destinationWebGateway/@webGatewayAgentType</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/originator/userID</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/job/name</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/status/supplement</insert>
    </inserts>
    <separator>;</separator>
  </format>
</transferComplete>
<transferDelete>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="true">/transaction/status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
        /transaction/sourceWebUser/@webGatewayAgentName |
        /transaction/sourceWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@QMGr |
        /transaction/sourceWebUser/@webGatewayAgentQMGr |
        /transaction/sourceWebGateway/@webGatewayAgentQMGr</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@agentType |
        /transaction/sourceWebUser/@webGatewayAgentType |
        /transaction/sourceWebGateway/@webGatewayAgentType</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@agent |
        /transaction/destinationWebUser/@webGatewayAgentName |
        /transaction/destinationWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/destinationAgent/@QMGr |
        /transaction/destinationWebUser/@webGatewayAgentQMGr |
        /transaction/destinationWebGateway/@webGatewayAgentQMGr</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/destinationAgent/@agentType |
        /transaction/destinationWebUser/@webGatewayAgentType |
        /transaction/destinationWebGateway/@webGatewayAgentType</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/originator/userID</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/job/name</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/status/supplement</insert>
    </inserts>
    <separator>;</separator>
  </format>
</transferDelete>
<transferProgress>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="true">status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">source/file | source/queue</insert>
      <insert type="user" width="0" ignoreNull="false">source/file/@size | source/queue/@size</
insert>
      <insert type="user" width="5" ignoreNull="true">source/@type</insert>
      <insert type="user" width="6" ignoreNull="true">source/@disposition</insert>
      <insert type="user" width="0" ignoreNull="true">source/file/@alias | source/queue/@alias</
insert>
      <insert type="user" width="0" ignoreNull="true">source/file/@filesystem | source/queue/
@filesystem</insert>
      <insert type="user" width="0" ignoreNull="true">source/@correlationBoolean1</insert>

```

```

        <insert type="user" width="0" ignoreNull="true">source/@correlationNum1</insert>
        <insert type="user" width="0" ignoreNull="true">source/@correlationString1</insert>
        <insert type="user" width="0" ignoreNull="false">destination/file | destination/queue</
insert>
        <insert type="user" width="0" ignoreNull="false">destination/file/@size | destination/queue/
@size</insert>
        <insert type="user" width="5" ignoreNull="true">destination/@type</insert>
        <insert type="user" width="9" ignoreNull="true">destination/@exist</insert>
        <insert type="user" width="0" ignoreNull="true">destination/file/@alias | destination/queue/
@alias</insert>
        <insert type="user" width="0" ignoreNull="true">destination/file/@filespace | destination/
queue/@filespace</insert>
        <insert type="user" width="0" ignoreNull="true">destination/file/@truncateRecords</insert>
        <insert type="user" width="0" ignoreNull="true">destination/@correlationBoolean1</insert>
        <insert type="user" width="0" ignoreNull="true">destination/@correlationNum1</insert>
        <insert type="user" width="0" ignoreNull="true">destination/@correlationString1</insert>
        <insert type="user" width="0" ignoreNull="true">status/supplement</insert>
    </inserts>
    <separator>;</separator>
</format>
</transferProgress>
</messageTypes>
</logFormatDefinition>

```

相关参考

第 154 页的『独立文件记录器格式』

文件记录器写入的消息信息的格式可在 `FileLoggerFormat.xml` 文件中定义。

第 644 页的『独立文件记录器格式 XSD』

独立文件格式的模式。

独立文件记录器格式 XSD

独立文件格式的模式。

模式

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
@start_non_restricted_prolog@
Version: %Z% %I% %W% %E% %U% [%H% %T%]

Licensed Materials - Property of IBM

5724-H72

Copyright IBM Corp. 2011, 2023. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
@end_non_restricted_prolog@
-->

<!--
This schema defines the format of the FileLoggerFormat XML file that contains the definition
of the format to use when logging FTE log messages to a file. When an XML file that conforms
to this schema is processed by a file logger it can contain definitions for one or more
message type(s) that define how log messages of those types are output to the file log.
-->

<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>

  <!--
    Defines the logFileDefinition and version number
  >
    <logFileDefinition version="1.00" ...
      <messageTypes>
        ...
      </messageTypes>
    </logFileDefinition>
  -->
  <xsd:element name="logFileDefinition">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="messageTypes" type="messageTypesType" maxOccurs="1" minOccurs="1"/>

```



```

        </xsd:sequence>
        <xsd:attribute name="version"          type="versionType"          use="required"/>
    </xsd:complexType>
</xsd:element>

<!--
    Defines the set of accepted message types. The definition of individual message types
    is optional. If a particular types element is present but empty then no line will be
    output for messages of that type. If a particular types element is not present then
    the default format will be used to format messages of that type.
-->
<xsd:complexType name="messageTypesType">
    <xsd:sequence>
        <xsd:element name="callCompleted"          type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="callStarted"            type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="monitorAction"          type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="monitorCreate"          type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="monitorFired"           type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="notAuthorized"          type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="scheduleDelete"          type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="scheduleExpire"          type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="scheduleSkipped"         type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="scheduleSubmitInfo"      type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="scheduleSubmitTransfer" type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="scheduleSubmitTransferSet" type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="transferStarted"         type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="transferCancelled"       type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="transferComplete"        type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="transferDelete"         type="messageType"    maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="transferProgress"        type="messageType"    maxOccurs="1"
minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<!--
    Defines the content of a message type definition e.g.
    <callStarted>
    <format>
    ...
    </format>
    <callStarted>
-->
<xsd:complexType name="messageType">
    <xsd:sequence>
        <xsd:element name="format"          type="messageFormatType"    maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<!--
    Defines the content of a message format definition e.g.
    <format>
    <inserts>
    ...
    </inserts>
    <separator>;</separator>
    </format>
-->
<xsd:complexType name="messageFormatType">
    <xsd:sequence>
        <xsd:element name="inserts"          type="insertsType"    maxOccurs="1" minOccurs="1"/>
        <xsd:element name="separator"        type="scheduleType"    maxOccurs="1" minOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

```

```

<!--
  Defines the content of the inserts element e.g.

  <inserts>
    <insert ...>
    <insert ...>
    ...
  </inserts>
-->
<xsd:complexType name="insertsType">
  <xsd:sequence>
    <xsd:element name="insert" type="insertType" maxOccurs="unbounded" minOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<!--
  Defines the content of an insert definition e.g.

  <insert type="user" width="0" ignoreNull="true">/transaction/@ID</insert>
-->
<xsd:complexType name="insertType">
  <xsd:attribute name="type" type="insertTypeType" use="required"/>
  <xsd:attribute name="width" type="xsd:nonNegativeInteger" use="required"/>
  <xsd:attribute name="ignoreNull" type="xsd:boolean" use="required"/>
</xsd:complexType>

<!--
  Defines the accepted choices for the insert type attribute.
-->
<xsd:simpleType name="insertTypeType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="user"/>
    <xsd:enumeration value="system"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

相关参考

第 154 页的『独立文件记录器格式』

文件记录器写入的消息信息的格式可在 `FileLoggerFormat.xml` 文件中定义。

第 639 页的『独立文件记录器缺省日志格式定义』

独立文件记录器的缺省日志文件格式定义

SYSTEM.FTE 主题

SYSTEM.FTE 主题是关于 IBM MQ Managed File Transfer 用于记录传输并存储有关代理、监视器、调度和模板的协调队列管理器的主题。

主题结构

```

SYSTEM.FTE
  /Agents
    /agent_name
  /monitors
    /agent_name
  /Scheduler
    /agent_name
  /Templates
    /template_ID
  /Transfers
    /agent_name
    /transfer_ID
  /Log
    /agent_name
    /Monitors
    /schedule_ID
    /transfer_ID

```

SYSTEM.FTE/Agents/agent_name

本主题包含描述您的 IBM MQ Managed File Transfer 网络中的代理及其属性的保留出版物。将定期更新有关本主题的消息以及代理状态。有关更多信息，请参阅第 647 页的『代理状态消息格式』。

SYSTEM.FTE/monitors/agent_name

本主题包含描述与代理 *agent_name* 相关联的资源监视器的保留出版物。保留出版物的 XML 符合模式 *MonitorList.xsd*。有关更多信息，请参阅第 650 页的『监视器列表消息格式』。

SYSTEM.FTE/Scheduler/agent_name

本主题包含描述与代理 *agent_name* 相关联的所有活动调度的保留出版物。保留出版物的 XML 符合模式 *ScheduleList.xsd*。有关更多信息，请参阅第 654 页的『安排列表消息格式』。

SYSTEM.FTE/Templates

本主题包含描述 IBM MQ Managed File Transfer 拓扑中定义的所有模板的保留出版物。

- 与每个模板相关的出版物将发布到名为 *SYSTEM.FTE/Templates/template_ID* 的子主题。

有关此保留出版物的内容的示例，请参阅第 658 页的『示例模板 XML 消息』。

SYSTEM.FTE/Transfers/agent_name

本主题包含描述源自代理 *agent_name* 的传输状态的出版物。与每个传输相关的出版物将发布到名为 *SYSTEM.FTE/Transfers/agent_name/transfer_ID* 的子主题。这些出版物供 WebSphere MQ Explorer 插件用来提供有关单个传输的进度信息。出版物的 XML 符合模式 *TransferStatus.xsd*。有关更多信息，请参阅第 658 页的『文件传输状态消息格式』。

SYSTEM.FTE/Log/agent_name

本主题包含记录有关源自代理 *agent_name* 的传输、监视器和调度信息的出版物。这些出版物可由数据库记录器记录，以提供 IBM MQ Managed File Transfer 网络中所发生事件的审计记录。

- 与每个传输相关的出版物将发布到名为 *SYSTEM.FTE/Log/agent_name/transfer_ID* 的子主题，并且出版物的 XML 符合模式 *TransferLog.xsd*。有关更多信息，请参阅第 662 页的『文件传输日志消息格式』。
- 与每个调度的传输相关的出版物将发布到名为 *SYSTEM.FTE/Log/agent_name/schedule_ID* 的子主题，并且出版物的 XML 符合模式 *ScheduleLog.xsd*。有关更多信息，请参阅第 686 页的『已安排的传输日志消息格式』。
- 与每次监视相关的出版物将发布到名为 *SYSTEM.FTE/Log/agent_name/Monitors/monitor_name/monitor_ID* 的子主题，并且出版物的 XML 符合模式 *MonitorLog.xsd*。有关更多信息，请参阅第 691 页的『监视器日志消息格式』。

代理状态消息格式

在创建或启动代理时，该代理会向其协调队列管理器上的 *SYSTEM.FTE* 主题（在 *SYSTEM.FTE/Agents/agent_name* 主题上）发布其详细信息。

包含以下信息：

- 代理程序名称
- 运行代理的平台
- 代理描述（如果提供）
- 代理队列管理器
- 代理运行的时区
- 代理版本
- 代理传输限制
- 代理的所有当前传输的状态。[代理传输状态](#)中列出了这些状态
- 代理类型

如果代理是协议网桥代理，那么还将包含以下信息：

- 协议网桥代理类型
- 协议网桥服务器的主机名或 IP 地址

如果代理是 Web 代理，那么还将包含以下信息：

- Web 代理连接到的 Web 网关的名称

只要代理传输状态发生更改，就会重新发布代理状态，但缺省情况下不超过 30 秒。您可以使用 `agentStatusPublishRateLimit` 代理属性来更改该缺省设置，[高级代理属性](#)中描述了该属性。

以下示例的输出显示用于代理状态中的每个数据元素的键：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="agentOsName">Windows 7</entry>
  <entry key="agentDescription"/>
  <entry key="queueManager">QM1</entry>
  <entry key="agentTimeZone">Europe/London</entry>
  <entry key="agentVersion">1.00</entry>
  <entry key="agentName">FTEAGENT</entry>
  <entry key="maxDestinationTransfers">25</entry>
  <entry key="maxSourceTransfers">25</entry>
  <entry key="maxQueuedTransfers">100</entry>
  <entry
key="DestinationTransferStates">414d51204d554e474f202020202020d857374a60a72622=RunningTransfer
414d51204d554e474f202020202020d857374a69a72622=RunningTransfer
414d51204d554e474f202020202020d857374a75a72622=RunningTransfer
  </entry>
  <entry
key="SourceTransferStates">414d51204d554e474f202020202020d857374a93a72622=NegotiatingTransfer
414d51204d554e474f202020202020d857374a78a72622=RunningTransfer
414d51204d554e474f202020202020d857374aaba72622=NewSenderTransfer
414d51204d554e474f202020202020d857374a63a72622=RunningTransfer
  </entry>
</properties>
```

以下示例的输出显示用于协议网桥代理的代理状态中的每个数据元素的键：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="agentOsName">Windows 7</entry>
  <entry key="agentDescription"/>
  <entry key="queueManager">QM1</entry>
  <entry key="agentTimeZone">Europe/London</entry>
  <entry key="agentVersion">1.00</entry>
  <entry key="agentName">BRIDGE</entry>
  <entry key="protocolBridgeType">ftp</entry>
  <entry key="protocolBridgeServerHost">ftpserver.example.org</entry>
  <entry key="maxDestinationTransfers">25</entry>
  <entry key="maxSourceTransfers">25</entry>
  <entry key="maxQueuedTransfers">100</entry>
  <entry key="DestinationTransferStates">414d51204d554e474f202020202020d857374a60a72622=RunningTransfer
  </entry>
  <entry key="SourceTransferStates">414d51204d554e474f202020202020d857374a93a72622=NegotiatingTransfer
  </entry>
</properties>
```

相关参考

[第 649 页的『代理传输状态』](#)

启动的代理将其详细信息发布在其协调队列管理器上的 `SYSTEM.FTE` 主题中。这些详细信息中包括涉及该代理的每个当前传输的状态。包含的状态如下：

[第 842 页的『文件传输请求消息格式』](#)

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the `FileTransfer.xsd` schema and have the `<request>` element as the root element. `FileTransfer.xsd` 模式文档位于 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。`FileTransfer.xsd` 模式导入相同目录中的 `fteutils.xsd`。

[第 658 页的『文件传输状态消息格式』](#)

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 `SYSTEM.FTE/Transfers/agent_name/transfer ID` 主题中），它符合 `TransferStatus.xsd` XML 模式。`TransferStatus.xsd` 文件位于 WMQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

[第 662 页的『文件传输日志消息格式』](#)

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 IBM MQ Managed File Transfer 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

第 686 页的『已安排的传输日志消息格式』

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 `SYSTEM.FTE/Log/agent name/schedule ID` 主题中）。该消息符合 `ScheduleLog.xsd` XML 模式。

第 859 页的『监视器请求消息格式』

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 `fteCreateMonitor` 命令或使用 WebSphere MQ Explorer 接口。

第 871 页的『针对安全性的消息格式』

本主题描述发布到协调队列管理器且与安全性相关的消息。

代理传输状态

启动的代理将其详细信息发布在其协调队列管理器上的 `SYSTEM.FTE` 主题中。这些详细信息中包括涉及该代理的每个当前传输的状态。包含的状态如下：

传输状态	说明
<code>NewSenderTransfer</code>	新的传输，来自未针对其启动协商的源代理。
<code>NewReceiverTransfer</code>	已在协商的过程中在目标代理处创建了新的传输，但该传输尚未运行。
<code>NegotiatingTransfer</code>	运行传输前，源代理会与目标代理协商。
<code>RunningTransfer</code>	来自源代理或目标代理且处于正常运行状态的传输
<code>RecoveringTransfer</code>	当源代理或目标代理启动恢复过程时，处于运行状态的所有传输都将转变为传输状态。当向同级代理发送再同步消息时，传输将由该状态进入 <code>ReSynchronisingTransfer</code> 状态。 例如，如果目标代理针对正在运行的传输启动恢复过程，那么在向其源代理发送再同步消息时，该传输将进入 <code>ReSynchronisingTransfer</code> 状态。
<code>ReSynchronisingTransfer</code>	传输源代理或目标代理发现了问题并向其各自的目标代理或源代理发送再同步消息。
<code>CompletedTransfer</code>	目标代理已完成传输并向源代理发送了完成消息。目标代理正在等待来自源代理的确认消息。
<code>CompleteReceivedTransfer</code>	源代理收到来自目标代理的完成消息，并向目标代理发送了回复消息以确认该完成消息。
<code>CancelledNewTransfer</code>	源代理收到有关取消新传输的消息。
<code>CancelledInProgressTransfer</code>	源代理收到有关取消正在进行的传输的消息。
<code>ResumingTransfer</code>	源代理收到再同步响应消息并且正在安排重新启动该传输。
<code>RestartingTransfer</code>	源代理或目标代理收到再同步请求消息并在等待各自的目标代理或源代理重新启动。
<code>WaitingForDestinationCapacity</code>	源代理从目标代理处收到 <code>DESTINATION_CAPACITY_EXCEEDED</code> 错误。传输现处于等待状态，以在某时间段之后重试。
<code>FailedTransferEnding</code>	传输已失败，但未发布完成日志消息，并且未将该传输从状态存储中移除。例如，如果代理进程在收到来自目标代理的失败响应之后但在后续处理完成之前停止，那么将会出现该状态。

相关参考

第 700 页的『代理状态值』

`fteListAgents` 和 **`fteShowAgentDetails`** 命令生成代理状态信息。该状态有多种可能的值。

第 647 页的『代理状态消息格式』

在创建或启动代理时，该代理会向其协调队列管理器上的 SYSTEM.FTE 主题（在 SYSTEM.FTE/Agents/*agent name* 主题上）发布其详细信息。

第 842 页的『文件传输请求消息格式』

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

第 658 页的『文件传输状态消息格式』

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 SYSTEM.FTE/Transfers/*agent_name/transfer ID* 主题中），它符合 TransferStatus.xsd XML 模式。TransferStatus.xsd 文件位于 WMQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

第 662 页的『文件传输日志消息格式』

文件传输日志消息发布在主题字符串为 Log/*agent_name/transfer ID* 的 SYSTEM.FTE 主题内。这些消息符合模式 TransferLog.xsd，该模式位于 IBM MQ Managed File Transfer 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

第 686 页的『已安排的传输日志消息格式』

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/*agent name/schedule ID* 主题中）。该消息符合 ScheduleLog.xsd XML 模式。

第 859 页的『监视器请求消息格式』

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 fteCreateMonitor 命令或使用 WebSphere MQ Explorer 接口。

第 871 页的『针对安全性的消息格式』

本主题描述发布到协调队列管理器且与安全性相关的消息。

监视器列表消息格式

作为保留出版物发布至主题字符串 SYSTEM.FTE/monitors/*agent_name/monitor_name* 的 XML 消息符合 MonitorList.xsd 模式。每条 XML 消息均列出属于此代理的活动监视器。此信息由 **fteListMonitors** 命令和 WebSphere MQ Explorer 插件用于向用户显示监视器列表。MonitorList.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。MonitorList.xsd 模式导入位于同一目录中的 Monitor.xsd。

模式

以下模式描述了监视器列表 XML 消息中哪些元素有效。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema"
  targetNamespace="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
  xmlns="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition">

  <xsd:include schemaLocation="Monitor.xsd"/>

  <xsd:element name="monitorList">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="status" type="monitorStatusType" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="configuration" type="monitorConfigurationType" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="pollInterval" type="pollIntervalType" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="batch" type="batchType" minOccurs="1" maxOccurs="1"/>
        <xsd:any minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="agent" type="xsd:string" use="required"/>
      <xsd:attribute name="monitor" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="monitorStatusType">
    <xsd:sequence>
      <xsd:any minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="state" type="xsd:token"/>
    <xsd:anyAttribute/>
  </xsd:complexType>

  <xsd:complexType name="monitorConfigurationType">
    <xsd:sequence>
      <xsd:element name="description" type="xsd:string" minOccurs="1" maxOccurs="1" />
      <xsd:element name="resources" type="monitorResourcesType" minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```



```
<xsd:element name="triggerMatch" type="triggerMatchType" minOccurs="0" maxOccurs="1" />
<xsd:element name="tasks" type="monitorListTasksType" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:anyAttribute/>
</xsd:complexType>

<xsd:complexType name="monitorListTasksType">
  <xsd:sequence>
    <xsd:element name="task" type="monitorListTaskType" minOccurs="1" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="monitorListTaskType">
  <xsd:sequence>
    <xsd:element name="name" type="monitorTaskNameType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1" />
    <xsd:element name="taskXML" type="xsd:string" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>
```

了解监视器列表消息

以下列表中描述了监视器列表消息中使用的元素和属性：

<monitorList>

包含元素的组元素描述了为代理定义的监视器。

属性	描述
代理程序	必需。 在其上定义资源监视器的代理的名称。
monitor	必需。 监视器的名称。 对于此代理是唯一的。
version	必需。 监视器列表消息格式的版本。

<status>

监视器的状态。

属性	描述
状态	监视器的状态。

<configuration>

组元素， 包含描述监视器配置的元素。

<description>

监视器的描述。（当前未使用。）

<resources>

正在监视的资源。

<directory>

要监视的目录。

属性	描述
recursionLevel	要监视的从顶级向下的目录级别数。
标识	资源的标识。

<queue>

要监视的队列。

属性	描述
标识	资源的标识。

<triggerMatch>

包含 <conditions> 元素的元素。

<conditions>
包含资源监视器监视的条件的元素。该元素只能包含以下元素中的一个：<allOf>、<anyOf> 或 <condition>。

<allOf>
包含资源监视器监视的条件的元素。该元素可包含一个或多个 <condition> 元素。要触发资源监视器，必须满足该元素内的所有条件。

<anyOf>
包含资源监视器监视的条件的元素。该元素可包含一个或多个 <condition> 元素。要触发资源监视器，必须仅满足该元素内的一个条件。

<condition>
包含资源监视器监视的一项条件的元素。This element can contain only one of the following elements: <fileMatch>, <fileNoMatch>, <fileSize>, <queueNotEmpty>, <completeGroups>, or <fileSizeSame>. It can also contain a <name> element and a <resource> element.

如果正在监视的资源是一个目录，那么必须在条件中指定以下三个元素中的一个：

- fileMatch
- fileNoMatch
- fileSize

如果监视中的资源是队列，那么必须在条件中指定以下两个元素中的一个：

- queueNotEmpty
- completeGroups

<fileMatch>
文件名匹配条件的组元素。

<pattern>
指定文件名匹配模式。资源上的文件必须与模式相匹配才能满足条件。缺省模式为 *（任何文件都将匹配）。

<fileNoMatch>
逆文件名匹配条件的组元素。

<pattern>
指定逆文件名匹配模式。如果受监视资源上的所有文件均不匹配，那么满足条件。缺省模式为 *（缺少任何文件都将匹配）。

<fileSize>
文件大小比较的组元素。

<compare>
指定文件大小比较。该值必须是非负整数。

属性	描述
操作员	要使用的比较运算符。仅支持> = '。
单元	指定文件大小单位，可以是以下其中一项： • B - 字节 • KB - 千字节 • MB - 兆字节 • GB - 千兆字节 单位值不区分大小写，因此 mb' 与 MB' 作用相同。

<pattern>
要匹配的文件名模式。缺省值为 *（任何文件都将匹配）。

<queueNotEmpty>

只能在资源是队列时指定此属性。指定要触发监视器，队列上必须有消息。

<completeGroups>

只能在资源是队列时指定此属性。指定要触发监视器，队列上必须有完整的一组消息。针对队列上每个完整的组执行一项传输任务。

<name>

条件的名称。

<resource>

标识要与条件相比较的资源定义。

属性	描述
标识	资源的唯一标识。

<tasks>

组元素，包含的元素可指定在满足监视器触发器条件时要调用的任务。

<task>

组元素，定义在满足触发器条件时监视器将调用的个别任务。当前只能指定一项任务。

<name>

任务的名称。接受任何字母数字字符。

<description>

任务的描述。允许任何文本值。

<taskXML>

描述监视器将要执行的任务的 XML 消息。该元素的内容采用转义 XML 格式。

<pollInterval>

每次针对触发器条件检查资源的时间间隔。

属性	描述
单元	指定轮询时间间隔的时间单位。有效值包括： • 秒 • 分钟 • 小时 • days • weeks • months • years

<batch>

要在一个批次中包含的触发器匹配项的最大数目。

属性	描述
maxSize	要在一个批次中包含的触发器匹配项的最大数量

以下 XML 显示了在 AGENT_JUPITER 上创建名为 MONITORTWO 的监视器时，发布至主题字符串 SYSTEM.FTE/monitors/*agent_name*/MONITORTWO 的保留出版物的示例。<taskXML> 元素内的转义 XML 描述了当满足监视器条件时提交的任务。

```
<?xml version="1.0" encoding="UTF-8"?>
<lst:monitorList xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:lst="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
  xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition MonitorList.xsd"
  version="4.00"
  agent="AGENT_JUPITER"
  monitor="MONITORTWO">
  <status state="started"/>
</lst:monitorList>
```

```

<configuration>
  <description/>
  <resources>
    <directory recursionLevel="0" id="">/srv/nfs/incoming</directory>
  </resources>
  <triggerMatch>
    <conditions>
      <condition>
        <name/>
        <resource id=""/>
        <fileMatch>
          <pattern>*.completed</pattern>
        </fileMatch>
      </condition>
    </conditions>
  </triggerMatch>
  <tasks>
    <task>
      <name/>
      <description/>
      <taskXML>&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;request
xmlns: xsi="https://www.w3.org/2001/XMLSchema-instance" version="4.00"
xsi: noNamespaceSchemaLocation="FileTransfer.xsd"&gt;&lt;managedTransfer&gt;
&lt;originator&gt;&lt;hostName&gt;example.com.&lt;/hostName&gt;
&lt;userID&gt;mqm&lt;/userID&gt;&lt;/originator&gt;
&lt;sourceAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/&gt;
&lt;destinationAgent QMgr="QM_JUPITER" agent="AGENT_SATURN"/&gt;
&lt;transferSet&gt;&lt;item checksumMethod="MD5" mode="binary"&gt;
&lt;source disposition="leave" recursive="false"&gt;&lt;file
&gt;/srv/nfs/incoming/*.txt&lt;/file&gt;&lt;/source&gt;
&lt;destination exist="error" type="directory"&gt;
&lt;file&gt;/srv/backup&lt;/file&gt;&lt;/destination&gt;
&lt;/item&gt;&lt;/transferSet&gt;&lt;/managedTransfer&gt;
&lt;/request&gt;
&lt;/taskXML>
    </task>
  </tasks>
</configuration>
<pollInterval units="minutes">1</pollInterval>
<batch maxSize="1"/>
</lst:monitorList>

```

安排列表消息格式

发布到主题字符串为 `SYSTEM.FTE/Scheduler/agent_name` 的保留发布上的 XML 消息符合 `ScheduleList.xsd` 模式。XML 消息将列出所有属于该代理的活动安排。

fteListScheduledTransfers 命令和 WebSphere MQ Explorer 插件会使用该信息以向用户显示安排列表。`ScheduleList.xsd` 模式文档位于 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。`ScheduleList.xsd` 模式将导入位于相同目录中的 `FileTransfer.xsd`。

模式

以下模式描述了监视器列表 XML 消息中哪些元素有效。

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="FileTransfer.xsd"/>

  <xsd:element name="schedules">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="managedTransfer" type="scheduledManagedTransferType" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="size" type="xsd:nonNegativeInteger" use="required"/>
      <xsd:attribute name="agent" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="scheduledManagedTransferType">
    <xsd:sequence>
      <xsd:element name="originator" type="origRequestType" maxOccurs="1" minOccurs="1"/>
      <xsd:element name="schedule" type="scheduleListType" maxOccurs="1" minOccurs="0"/>
      <xsd:element name="sourceAgent" type="agentType" maxOccurs="1" minOccurs="1"/>
      <xsd:element name="destinationAgent" type="agentClientType" maxOccurs="1" minOccurs="1"/>
      <xsd:element name="trigger" type="triggerType" maxOccurs="1" minOccurs="0"/>
      <xsd:element name="reply" type="replyType" maxOccurs="1" minOccurs="0"/>
      <xsd:element name="transferSet" type="transferSetType" maxOccurs="1" minOccurs="1"/>
      <xsd:element name="job" type="jobType" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="idType" use="required"/>
  </xsd:complexType>

  <xsd:complexType name="scheduleListType">
    <xsd:sequence>
      <xsd:element name="submit" type="submitType" maxOccurs="1" minOccurs="1"/>
      <xsd:element name="repeat" type="repeatType" maxOccurs="1" minOccurs="0"/>
      <xsd:element name="next" type="noZoneTimeType" maxOccurs="1" minOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>

</xsd:schema>

```

了解安排列表消息

以下列表描述了安排列表消息中使用的元素和属性：

<schedules>

包含有关单个代理上定义的所有安排的信息的组元素。

属性	描述
代理程序	必需。在其上定义安排的源代理的名称。
size	必需。在该代理上定义的安排的数量。
version	必需。安排列表消息格式的版本。

<managedTransfer>

包含有关单个安排的信息的组元素。

属性	描述
标识	必需。安排请求消息的十六进制字符串标识。

<originator>

安排请求的发起方。

<hostName>

提交安排请求的主机名。

<userID>

提交安排请求的用户的用户标识。

<mqmdUserID>

提交安排请求的用户的 MQMD 用户标识。

<webBrowser>

如果通过 Web 网关提交了安排请求，那么是指提交请求的 Web 浏览器。

<webUserID>

如果通过 Web 网关提交安排请求，那么是指提交安排请求的用户的 Web 用户标识。

<schedule>

包含描述何时发生已安排传输的元素的元素。

<submit>

指定已调度传输应启动的日期和时间。

属性	描述
timebase	指定要使用的时区。该属性的值可以是以下值之一： • source - 使用源代理的时区 • admin - 使用发出命令的管理员的时区 • UTC - 使用全球标准时间
时区	依据 timebase 值的时区描述

<repeat>

包含有关已调度传输重复频率、已调度传输重复次数以及已调度传输何时停止重复的详细信息的组元素。

属性	描述
区间 (interval)	时间间隔单位，必须是以下值之一： • 分钟

属性	描述
	• 小时 • days • weeks • months • years

<frequency>

重复传输之前必须等待的时间段。

属性	描述
区间 (interval)	时间间隔单位，必须是以下值之一： • 分钟 • 小时 • days • weeks • months • years

<expireTime>

指定重复的已调度传输停止的日期和时间的可选元素。该元素与 <expireCount> 元素互斥。

<expireCount>

指定在停止前发生已调度文件传输的次数的可选元素。该元素与 <expireTime> 元素互斥。

<next>

指定下一个已安排传输应启动的日期和时间。

<sourceAgent>

指定源文件所在系统上的代理的名称。

属性	描述
代理程序	指定代理的名称。
QMgr	代理队列管理器的名称。

<destinationAgent>

指定您要向其传输文件的系统上代理的名称。

属性	描述
代理程序	指定代理的名称。
QMgr	代理队列管理器的名称。

<trigger>

指定在文件传输发生时必须满足的条件的可选元素。

属性	描述
日志	指示是否记录触发失败的标志。 以下是有效值： • yes - 为失败的已触发传输创建日志条目 • no - 不为失败的已触发传输来创建日志条目

<reply>

指定为同步文件传输生成的临时应答队列的名称（在命令行上使用 **-w** 参数指定）。队列的名称由 `command.properties` 配置文件中的键 **dynamicQueuePrefix** 或缺省值 `WMQFTE.*` 定义。（如果未指定）。

属性	描述
QMGR	在其上生成临时动态队列以接收应答的命令队列管理器的名称。

<transferSet>

指定要与已安排传输一起执行的一组文件传输。在传输期间，<transferSet> 是包含 <item> 元素的组元素。

属性	描述
priority	传输的优先级。优先级为 0-9 之间的一个值，其中 0 为最低优先级。缺省优先级为 0，并且缺省情况下，传输使用源代理的优先级。

<job>

Optional group element containing job information for the entire transfer specification. <job> is a user-defined job name identifier that is added to the log message when the transfer has started. This <job> element is the same as the <job> element that appears in the transfer log message, which is described in the following topic: [第 662 页的『文件传输日志消息格式』](#)。

示例

```
<?xml version="1.0" encoding="UTF-8"?>
<schedules xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  size="2"
  version="4.00"
  agent="AGENT_JUPITER"
  xsi:noNamespaceSchemaLocation="ScheduleList.xsd">
  <managedTransfer id="1">
    <originator>
      <hostName>example.com.</hostName>
      <userID>mqm</userID>
    </originator>
    <schedule>
      <submit timebase="admin" timezone="Europe/London">2010-01-01T21:00+0000</
submit>
      <next>2010-01-01T21:00+0000</next>
    </schedule>
    <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
    <destinationAgent agent="AGENT_SATURN" QMgr="QM_JUPITER"/>
    <reply QMGR="QM_JUPITER">WMQFTE.4D400F8B20004E06</reply>
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>etc/passwd</file>
        </source>
        <destination type="directory" exist="overwrite">
          <file>/tmp</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
  <managedTransfer id="2">
    <originator>
      <hostName>example.com.</hostName>
      <userID>mqm</userID>
    </originator>
    <schedule>
      <submit timebase="admin" timezone="Europe/London">2010-12-31T09:00+0000</
submit>
      <next>2010-12-31T09:00+0000</next>
    </schedule>
    <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
    <destinationAgent agent="AGENT_NEPTUNE" QMgr="QM_JUPITER"/>
    <reply QMGR="QM_JUPITER">WMQFTE.4D400F8B20004E09</reply>
    <transferSet>
```

```

        <item mode="binary" checksumMethod="MD5">
            <source recursive="false" disposition="leave">
                <file>/etc/passwd</file>
            </source>
            <destination type="directory" exist="overwrite">
                <file>/tmp</file>
            </destination>
        </item>
    </transferSet>
</managedTransfer>
</schedules>

```

示例模板 XML 消息

在创建模板时，将向主题字符串为 `Templates/template_ID` 的 `SYSTEM.FTE` 主题发布一条消息。该示例 XML 描述在您的 IBM MQ Managed File Transfer 网络中定义的单个模板。

```

<?xml version="1.0" encoding="UTF-8"?>
<transferTemplate version="4.00" id="baf9df73-45c2-4bb0-a085-292232ab66bc">
    <name>BASIC_TEMPLATE</name>
    <sourceAgentName>AGENT_JUPITER</sourceAgentName>
    <sourceAgentQMgr>QM_JUPITER</sourceAgentQMgr>
    <destinationAgentName>AGENT_SATURN</destinationAgentName>
    <destinationAgentQMgr>QM_JUPITER</destinationAgentQMgr>
    <fileSpecs>
        <item mode="binary" checksumMethod="MD5">
            <source recursive="false" disposition="leave">
                <file>/etc/passwd</file>
            </source>
            <destination type="directory" exist="overwrite">
                <file>/tmp</file>
            </destination>
        </item>
    </fileSpecs>
    <priority>0</priority>
</transferTemplate>

```

相关任务

第 249 页的『使用 IBM MQ Explorer 创建文件传输模板』

您可以从 IBM MQ Explorer 或从命令行创建文件传输模板。然后，您可以使用该模板来利用模板详细信息创建新的文件传输，或者提交该模板以启动文件传输。

相关参考

第 484 页的『fteCreateTemplate（新建文件传输模板）』

fteCreateTemplate 命令创建文件传输模板，您可以保留该模板以供将来使用。唯一的必需参数是 **-tn** (*template_name*) 参数。其他所有参数都是可选的，但如果指定源文件，还必须提供目标文件。类似地，如果指定目标文件，还必须指定源文件规范。

文件传输状态消息格式

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 `SYSTEM.FTE/Transfers/agent_name/transfer ID` 主题中），它符合 `TransferStatus.xsd` XML 模式。`TransferStatus.xsd` 文件位于 WMQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

模式

以下模式描述了在传输状态 XML 消息中哪些元素有效。

```

<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation="fteutils.xsd"/>
    <xsd:element name="transaction">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="sourceAgent" type="agentType"
                    maxOccurs="1" minOccurs="1"/>
                <xsd:element name="destinationAgent" type="agentType"
                    maxOccurs="1" minOccurs="1"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>

```

```

        <xsd:element name="transferSet" type="transferSetType"
            maxOccurs="1" minOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="version" type="versionType" use="required"/>
        <xsd:attribute name="ID" type="IDType" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:complexType name="transferSetType">
    <xsd:sequence>
        <xsd:element name="stats" type="statsType"
            maxOccurs="1" minOccurs="1" />
        <xsd:element name="current" type="currentType"
            maxOccurs="1" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="time" type="xsd:dateTime" use="required" />
</xsd:complexType>

<xsd:complexType name="currentType">
    <xsd:sequence>
        <xsd:element name="source" type="fileSourceType"
            maxOccurs="1" minOccurs="1" />
        <xsd:element name="destination" type="fileDestinationType"
            maxOccurs="1" minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="transferred" type="xsd:nonNegativeInteger"
        use="required" />
    <xsd:attribute name="size" type="xsd:nonNegativeInteger" use="required" />
</xsd:complexType>

    <xsd:complexType name="statsType">
        <xsd:attribute name="bytes" type="xsd:nonNegativeInteger"
            use="required" />
        <xsd:attribute name="seconds" type="xsd:decimal"
            use="required" />
        <xsd:attribute name="currentItem" type="xsd:nonNegativeInteger"
            use="required" />
        <xsd:attribute name="totalItems" type="xsd:nonNegativeInteger" use="required" />
    </xsd:complexType>
</xsd:schema>

```

了解传输状态消息

下表描述了传输状态消息中使用的元素和属性：

<transaction>

包含用于文件传输的所有元素的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 提供的该元素的版本，指定该值。
标识	文件传输的唯一标识。

<sourceAgent>

指定源文件所在系统上的代理的名称。

属性	描述
代理程序	代理的名称。
QMgr	代理队列管理器的名称。

<destinationAgent>

指定您要向其传输文件的系统上代理的名称。

属性	描述
代理程序	代理的名称。
QMgr	代理队列管理器的名称。

<transferSet>

指定正一起执行的一组文件传输。传输中的所有文件都必须开始于相同的源代理，结束于相同的目标代理。

属性	描述
time	指定日期和时间（采用日期时间格式）。

<stats>

必需。定义关于在给定的秒数内传输的度量值，包括迄今为止复制的字节数。同时还在 <transferSet> 中提供当前项号与总项数。

属性	描述
字节	迄今为止复制的字节数。
秒	传输这些字节所花费的秒数。
currentItem	正传输的当前项的索引。
totalItems	正传输的总项数。

<current>

可选元素。包含指定目前正在进行的文件传输的元素的组元素。<current> 元素指示迄今为止已为当前项传输的数据字节数以及预期的字节总数。

<source>

包含指定源文件名的元素的组元素。

<file>

指定要传输的文件的源路径。该路径是为传输指定的路径。该路径可能与作为传输日志一部分输出的路径不同，后者是路径的绝对形式。

<destination>

包含指定目标文件名或规范的元素的组元素。

<file>

指定要传输的文件的目標路径。该路径是为传输指定的路径。该路径可能与作为传输日志一部分输出的路径不同，后者是路径的绝对形式。

属性	描述
别名	指定目标文件的别名。该别名是源文件的名称，不包括针对传输指定的任何目录路径。
文件空间	指定将目标文件写入的文件空间的名称。

<queue>

与 <destination> 元素一起使用时，指定要传输到的队列的名称。该名称格式为 QUEUE 或 QUEUE@QUEUE_MANAGER。

相关参考

第 661 页的『传输进度消息示例』

传输期间，会将消息发布到主题字符串为 Transfers/agent_name/transfer_ID 的 SYSTEM.FTE 主题。XML 示例会显示单个文件传输和多个文件传输的进度消息。

第 647 页的『代理状态消息格式』

在创建或启动代理时，该代理会向其协调队列管理器上的 SYSTEM.FTE 主题（在 SYSTEM.FTE/Agents/agent_name 主题上）发布其详细信息。

第 842 页的『文件传输请求消息格式』

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于

MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

第 662 页的『文件传输日志消息格式』

文件传输日志消息发布在主题字符串为 Log/agent_name/transfer_ID 的 SYSTEM.FTE 主题内。这些消息符合模式 TransferLog.xsd，该模式位于 IBM MQ Managed File Transfer 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

第 686 页的『已安排的传输日志消息格式』

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/agent_name/schedule ID 主题中）。该消息符合 ScheduleLog.xsd XML 模式。

第 859 页的『监视器请求消息格式』

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 fteCreateMonitor 命令或使用 WebSphere MQ Explorer 接口。

第 871 页的『针对安全性的消息格式』

本主题描述发布到协调队列管理器且与安全性相关的消息。

传输进度消息示例

传输期间，会将消息发布到主题字符串为 Transfers/agent_name/transfer_ID 的 SYSTEM.FTE 主题。XML 示例会显示单个文件传输和多个文件传输的进度消息。

单个文件传输

以下示例显示了正在进行的单个文件传输的详细信息。

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d223d0020"
  xsi:noNamespaceSchemaLocation="TransferStatus.xsd">
  <sourceAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
  <destinationAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
  <transferSet time="2011-01-26T13:03:26.542Z">
    <stats bytes="1198" seconds="0.018" currentItem="1" totalItems="1"/>
    <current transferred="1151" size="1151">
      <source>
        <file>/etc/passwd</file>
      </source>
      <destination>
        <file>/tmp/passwd</file>
      </destination>
    </current>
  </transferSet>
</transaction>
```

多个文件传输

如果传输集中存在多个文件，那么传输状态消息会表明正在处理哪个文件，以及到目前为止，已传输多少字节。

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d035c0020"
  xsi:noNamespaceSchemaLocation="TransferStatus.xsd">
  <sourceAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
  <destinationAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
  <transferSet time="2011-01-26T13:12:58.636Z">
    <stats bytes="440" seconds="0.082" currentItem="10" totalItems="10"/>
    <current transferred="0" size="0">
      <source>
        <file>/srv/nfs/incoming/file10.txt</file>
      </source>
      <destination>
        <file>/srv/nfs/outgoing/file10.txt</file>
      </destination>
    </current>
  </transferSet>
</transaction>
```

```
</transferSet>
</transaction>
```

文件传输日志消息格式

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 `IBM MQ Managed File Transfer` 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

如果要监视文件传输或收集有关它们的数据，请设置一个针对您感兴趣的传输而定制的通配符主题预订。例如：

```
Log/#
```

或者，

```
Log/FTEAGENT/#
```

该预订可以是持久或非持久的。持久预订在预订应用程序与队列管理器之间的连接关闭后继续存在。非持久预订只有在预订应用程序与队列管理器之间的连接处于打开状态时才存在。

模式

以下模式描述在传输日志 XML 消息中有效的元素。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>
  <xsd:element name="transaction">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="action" type="actionType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceAgent" type="agentExitStatusType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceWebGateway" type="webGatewayType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceWebUser" type="webUserType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="destinationAgent" type="agentExitStatusType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="destinationWebGateway" type="webGatewayType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="destinationWebUser" type="webUserType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="agent" type="agentExitStatusType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="originator" type="origRequestType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="status" type="statusType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="trigger" type="triggerType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="transferSet" type="transferSetType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="job" type="jobType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="scheduleLog" type="scheduleLogType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="statistics" type="statisticsType" maxOccurs="1" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="ID" type="IDType" use="required"/>
      <xsd:attribute name="relatedID" type="IDType" use="optional"/>
      <xsd:attribute name="agentRole" type="agentRoleType" use="optional"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="agentExitStatusType">
```



```

        <xsd:complexContent>
            <xsd:extension base="agentType">
                <xsd:sequence>
                    <xsd:element name="startExits" type="exitGroupType" minOccurs="0"
maxOccurs="1"/>
                    <xsd:element name="endExits" type="exitGroupType" minOccurs="0"
maxOccurs="1"/>
                    <xsd:element name="systemInfo" type="systemInfoType" minOccurs="0"
maxOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="transferSetType">
        <xsd:sequence>
            <xsd:element name="metaDataSet" type="metaDataSetType"
maxOccurs="1" minOccurs="0"/>
            <xsd:element name="call" type="callGroupType"
maxOccurs="1" minOccurs="0"/>
            <xsd:element name="preSourceCall" type="callGroupType"
maxOccurs="1" minOccurs="0"/>
            <xsd:element name="postSourceCall" type="callGroupType"
maxOccurs="1" minOccurs="0"/>
            <xsd:element name="preDestinationCall" type="callGroupType"
maxOccurs="1" minOccurs="0"/>
            <xsd:element name="postDestinationCall" type="callGroupType"
maxOccurs="1" minOccurs="0"/>
            <xsd:element name="item" type="itemType"
maxOccurs="unbounded" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="index" type="xsd:nonNegativeInteger" use="optional" />
        <xsd:attribute name="size" type="xsd:nonNegativeInteger" use="optional" />
        <xsd:attribute name="startTime" type="xsd:dateTime" use="required" />
        <xsd:attribute name="total" type="xsd:nonNegativeInteger" use="required" />
        <xsd:attribute name="bytesSent" type="xsd:nonNegativeInteger" use="required" />
    </xsd:complexType>

    <xsd:complexType name="itemType">
        <xsd:sequence>
            <xsd:element name="source" type="fileSourceChecksumType"
maxOccurs="1" minOccurs="1"/>
            <xsd:element name="destination" type="fileDestinationChecksumType"
maxOccurs="1" minOccurs="1"/>
            <xsd:element name="status" type="statusType"
maxOccurs="1" minOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="mode" type="modeType" use="required" />
    </xsd:complexType>

    <xsd:complexType name="fileSourceChecksumType">
        <xsd:complexContent>
            <xsd:extension base="fileSourceType">
                <xsd:sequence>
                    <xsd:element name="checksum" type="checksumType" minOccurs="0"
maxOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="fileDestinationChecksumType">
        <xsd:complexContent>
            <xsd:extension base="fileDestinationType">
                <xsd:sequence>
                    <xsd:element name="checksum" type="checksumType"
minOccurs="0" maxOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="actionType">
        <xsd:simpleContent>
            <xsd:extension base="actionEnumType">
                <xsd:attribute name="time" type="xsd:dateTime" use="required" />
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>

    <xsd:simpleType name="actionEnumType">
        <xsd:restriction base="xsd:token">

```

```

        <xsd:enumeration value="cancelled"/>
        <xsd:enumeration value="started"/>
        <xsd:enumeration value="progress"/>
        <xsd:enumeration value="completed"/>
        <xsd:enumeration value="malformed"/>
        <xsd:enumeration value="notAuthorized"/>
        <xsd:enumeration value="deleted"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="systemInfoType">
    <xsd:attribute name="architecture" type="xsd:string" use="required"/>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="version" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:element name="malformed">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="action" type="actionType"
                maxOccurs="1" minOccurs="1"/>
            <xsd:element name="agent" type="agentExitStatusType"
                maxOccurs="1" minOccurs="0"/>
            <xsd:element name="status" type="statusType"
                maxOccurs="1" minOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="version" type="versionType" use="required"/>
        <xsd:attribute name="ID" type="IDType" use="required"/>
        <xsd:attribute name="agentRole" type="agentRoleType" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="notAuthorized">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="action" type="actionType"
                maxOccurs="1" minOccurs="1"/>
            <xsd:element name="originator" type="origRequestType"
                maxOccurs="1" minOccurs="1"/>
            <xsd:element name="authority" type="xsd:string"
                minOccurs="1" maxOccurs="1"/>
            <xsd:element name="status" type="statusType"
                maxOccurs="1" minOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="version" type="versionType" use="required"/>
        <xsd:attribute name="ID" type="IDType" use="required"/>
        <xsd:attribute name="agentRole" type="agentRoleType" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:complexType name="statisticsType">
    <xsd:sequence>
        <xsd:element name="actualStartTime" type="xsd:dateTime"
            maxOccurs="1" minOccurs="0"/>
        <xsd:element name="retryCount" type="xsd:nonNegativeInteger"
            maxOccurs="1" minOccurs="1"/>
        <xsd:element name="numFileFailures" type="xsd:nonNegativeInteger"
            maxOccurs="1" minOccurs="1"/>
        <xsd:element name="numFileWarnings" type="xsd:nonNegativeInteger"
            maxOccurs="1" minOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="webGatewayType">
    <xsd:attribute name="webGatewayName" type="xsd:string" use="optional" />
    <xsd:attribute name="webGatewayAgentName" type="xsd:string" use="optional" />
    <xsd:attribute name="webGatewayAgentQMgr" type="xsd:string" use="optional" />
</xsd:complexType>

<xsd:complexType name="webUserType">
    <xsd:attribute name="webGatewayName" type="xsd:string" use="required" />
    <xsd:attribute name="webGatewayAgentName" type="xsd:string" use="optional" />
    <xsd:attribute name="webGatewayAgentQMgr" type="xsd:string" use="optional" />
</xsd:complexType>
</xsd:schema>

```

了解传输日志消息

<transaction>

用于指定一组要一起执行的传输的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 的详细说明指定该元素的版本。
标识	指定唯一事务标识。该标识最多可包含 48 个字母数字字符。
relatedID	可选。如果事务是从文件空间中删除或下载文件，那么 relatedID 指定将该文件上载到该文件空间中的传输的事务标识。
agentRole	可选。指定关注的代理是否在源或目标系统上
xmlns:xsi	名称空间声明。指示此模式中使用的元素和数据类型派生自 "https://www.w3.org/2001/XMLSchema-instance" 名称空间。
xsi:noNamespaceSchemaLocation	指定 XML 模式文档的名称和位置，以在没有名称空间声明的情况下验证该消息。为此属性指定的值必须引用 IBM MQ Managed File Transfer TransferLog.xsd 文档。

<action>

描述在 time 属性记录的时间，文件传输的状态。状态可以是以下某个值：

- 已启动
- 正在进行
- 已完成
- 已取消
- malformed（指示无法解释文件传输请求消息的内容）
- notAuthorized
- deleted

属性	描述
time	捕获传输状态的时间，以 UTC 格式表示。

<sourceAgent>

指定源文件所在系统上的代理的名称。只能指定 <sourceAgent>、<sourceWebUser> 或 <sourceWebGateway>。

<startExits>

包含一个或多个用户出口元素的组元素。该元素只能出现一次。

<endExits>

包含一个或多个用户出口元素的组元素。该元素只能出现一次。

<systemInfo>

描述系统体系结构、名称和版本。该元素只能出现一次。

属性	描述
代理程序	源系统上的代理的名称。
QMgr	源系统上的队列管理器的名称。
agentType	代理的类型。有效值包括： • STANDARD - 一般代理

属性	描述
	• BRIDGE - 协议网桥代理 • CD_BRIDGE - Connect:Direct 网桥代理 • EMBEDDED - 嵌入式代理 • WEB_GATEWAY - Web 代理 • SFG - Sterling File Gateway 嵌入式代理
bridgeURL	可选。如果代理是协议网桥代理，那么是指托管协议服务器的系统的主机名称。
pnode	可选。如果代理是 Connect:Direct 网桥代理，那么是指参与传输的 Connect:Direct 主节点的名称。
snode	可选。如果代理为 Connect:Direct 网桥代理，那么是指参与传输的 Connect:Direct 辅助节点的名称。
bridgeNode	可选。如果代理为 Connect:Direct 网桥代理，那么是指作为 Connect:Direct 网桥的组成部分的 Connect:Direct 节点的名称。这与主节点或辅助节点是同一个节点。

<sourceWebUser>

指定将源文件上载到 Web 网关的 Web 用户的名称。只能指定 <sourceAgent>、<sourceWebUser> 或 <sourceWebGateway>。

属性	描述
webGatewayName	接收来自 Web 用户的文件的 Web 网关的名称。
webGatewayAgentName	Web 网关用来向目标发送文件的 Web 代理的名称。
webGatewayAgentQMgr	Web 代理的队列管理器的名称。

<sourceWebGateway>

指定源文件下载自的 Web 网关的名称。只能指定 <sourceAgent>、<sourceWebUser> 或 <sourceWebGateway>。

属性	描述
webGatewayName	接收来自 Web 用户的文件的 Web 网关的名称。
webGatewayAgentName	Web 网关用来向目标发送文件的 Web 代理的名称。
webGatewayAgentQMgr	Web 代理的队列管理器的名称。

<destinationAgent>

指定文件要传输到的系统上的代理的名称。只能指定 <destinationAgent>、<destinationWebGateway> 或 <destinationWebUser>。

属性	描述
代理程序	目标系统上的代理的名称。
QMgr	目标系统上的队列管理器的名称。
agentType	代理的类型。有效值包括： • STANDARD - 一般代理 • BRIDGE - 协议网桥代理 • CD_BRIDGE - Connect:Direct 网桥代理 • EMBEDDED - 嵌入式代理

属性	描述
	• WEB_GATEWAY - Web 代理 • SFG - Sterling File Gateway 嵌入式代理
bridgeURL	可选。如果代理是协议网桥代理，那么是指托管协议服务器的系统的主机名称。
pnode	可选。如果代理是 Connect:Direct 网桥代理，那么是指参与传输的 Connect:Direct 主节点的名称。
snode	可选。如果代理为 Connect:Direct 网桥代理，那么是指参与传输的 Connect:Direct 辅助节点的名称。
bridgeNode	可选。如果代理为 Connect:Direct 网桥代理，那么是指作为 Connect:Direct 网桥的组成部分的 Connect:Direct 节点的名称。这与主节点或辅助节点是同一个节点。

<startExits>

包含一个或多个用户出口元素的组元素。该元素只能出现一次。

<endExits>

包含一个或多个用户出口元素的组元素。该元素只能出现一次。

<systemInfo>

描述系统体系结构、名称和版本。该元素只能出现一次。

<destinationWebUser>

指定从 Web 网关下载文件的 Web 用户的名称。只能指定 <destinationAgent>、<destinationWebGateway> 或 <destinationWebUser>。

属性	描述
webGatewayName	接收来自 Web 用户的文件的 Web 网关的名称。

<destinationWebGateway>

指定从 Web 网关下载文件的 Web 用户的名称。只能指定 <destinationAgent>、<destinationWebGateway> 或 <destinationWebUser>。

属性	描述
webGatewayName	接收来自 Web 用户的文件的 Web 网关的名称。
webGatewayAgentName	Web 网关使用的 Web 代理的名称。
webGatewayAgentQMGr	Web 代理的队列管理器的名称。

<originator>

包含指定请求发起方的元素的组元素。

<hostName>

源文件所在系统的主机名。

<userID>

发起文件传输的用户标识。

<mqmdUserID>

在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<webUserID>

可选。向提交传输请求的 Web 浏览器提供的用户标识。

<webBrowser>

可选。提交传输请求的 Web 浏览器。

<status>

结果代码和补充消息。

<trigger>

包含在原始传输请求中定义的触发器元素的组元素。 这些元素可以是以下中的任何一个或所有：

<fileExist>

基于是否存在文件的触发条件

<fileSize>

基于文件是否满足或超出了指定大小的触发条件。

<transferSet>

指定一组要一起执行的文件传输。 在传输期间，<transferSet> 是包含 <item> 元素的组元素。

属性	描述
startTime	记录一组传输开始的时间，以 UTC 格式表示。
total	指定该传输集中的总项数。
索引 (index)	可选属性。 指定第一个项在传输集进度中的位置。index 属性自零开始递增。例如，如果将 index 设置为 1，那么进度消息则为两个项中的第二个。
size	可选属性。 指定进度报告中的项数。
priority	可选属性。 传输的优先级。 优先级为 0-9 之间的一个值，其中 0 为最低优先级。 缺省优先级为 0，并且缺省情况下，传输使用源代理的优先级。

<metaDataSet>

包含一个或多个以下属性的组元素：

<metaData>

属性	描述
key	元数据键值对中的键部分。 <metaData> 元素内容包含键值对的值部分。 例如，<metaData key="testkey1">testvalue1</metaData>

<job>

Group element that contains an element specifying job details. <job> is a user-defined job name identifier that is added to the log message when the transfer has started. This <job> element is the same as the <job> element that is included in the transfer request message, which is described in the following topic: [第 842 页的『文件传输请求消息格式』](#) .

<name>

名称可以是任意字符串。

<scheduleLog>

包含指定源和目标文件名和位置的元素的组元素。

属性	描述
标识	如果传输为安排的传输，那么其与调度标识匹配。

<item>

包含指定源和目标文件名和位置的元素的组元素。

<source>

包含 <file> 元素或 <queue> 元素以及源系统上文件的 <checksum> 元素的组元素。

属性	描述
recursive	指定当 <source> 元素为目录或包含通配符时，以递归方式传输子目录中的文件。

属性	描述
disposition	指定在 <source> 已成功传输到其目标时要在 <source> 元素上执行的操作。有效选项如下： • leave - 源文件保持不变。 • delete - 在成功传输源文件后将源文件从源系统中删除。
correlationBoolean	布尔关联值。如果源为 Connect:Direct 网桥，那么该属性指定 Connect:Direct 进程是否为用户定义的。
correlationString1	字符串关联值。如果源为 Connect:Direct 网桥，那么该属性指定在传输的目标出现的 Connect:Direct 进程的名称。
correlationNum1	数字相关值。如果源为 Connect:Direct 网桥，那么该属性指定在传输的目标出现的 Connect:Direct 进程的标识号。

<queue>

在与 <source> 元素一起使用时，指定从中读取所传输消息的队列名称，该队列位于源代理队列管理器上。

属性	描述
messageCount	从队列中读取的消息数目。
groupId	从队列读取的消息的 WebSphere MQ 组标识。

<destination>

包含 <file> 元素或 <queue> 元素以及目标的 <checksum> 元素的组元素。

只有 <file> 和 <queue> 中的一个作为目标的子元素。

属性	描述
类型	目标的类型。有效选项如下： • file - 将文件指定为目标 • directory - 将目录指定为目标 • dataset - 指定 z/OS 数据集作为目标 • pds - 指定 z/OS 分区数据集作为目标 • queue - 指定 WebSphere MQ 队列作为目标 仅当 <destination> 元素的子元素为 <queue> 时，值 队列 才有效。 仅当 <destination> 元素具有子元素 <file> 时，其他值才有效。
exist	指定在目标系统上已存在目标文件时执行的操作。有效选项如下： • error - 报告一个错误，并且未传输文件。 • overwrite - 覆盖现有的目标文件。 如果 <destination> 元素具有子元素 <queue> ，那么此属性不能存在。
correlationBoolean	布尔关联值。如果目标为 Connect:Direct 网桥，那么该属性指定 Connect:Direct 进程是否为用户定义的。
correlationString1	字符串关联值。如果目标为 Connect:Direct 网桥，那么该属性指定在传输的目标出现的 Connect:Direct 进程的名称。
correlationNum1	数字相关值。如果目标为 Connect:Direct 网桥，那么该属性指定在传输的目标出现的 Connect:Direct 进程的标识号。

<file>
指定传输的文件的绝对路径（在源和目标上的路径）。标准路径与操作系统的格式一致，例如 C:/from/here.txt。文件 URI 未使用。

<queue>
在与 <destination> 元素一起使用时，指定传输到的队列名称，该队列位于与目标代理队列管理器连接的任何队列管理器上。

属性	描述
messageCount	写入队列的消息数目。
messageLength	写入队列的消息的长度。
groupId	如果传输请求指定文件被划分为多条消息，那么该属性的值将为写入队列的消息的 WebSphere MQ 组标识。
messageId	如果传输请求未指定文件被划分为多条消息，那么该属性的值将为写入队列的消息的 WebSphere MQ 消息标识。

<checksum>
可选元素。

指定生成消息摘要以创建数字签名的散列算法的类型。目前，IBM MQ Managed File Transfer 仅支持消息摘要算法 5 (MD5)。校验和提供一种方法来确认传输文件的完整性。

<malformed>
表示格式错误的消息的组元素。

属性	描述
version	
标识	
agentRole	源代理或目标代理

<statistics>
表示所传输的统计信息（如果可用）的组元素。

<actualStartTime>
代理开始运行传输的实际时间。通常，该时间与记录的传输启动时间相同（或非常接近）。然而，当代理忙碌时，提交的传输可能需要排队，直到代理有能力运行传输。

<retryCount>
传输进入恢复状态并由代理程序重试的次数。传输可能进入恢复状态，原因是由于 WebSphere MQ 网络错误或由于其在某时间段内未收到数据或确认消息，导致源或目标代理中断通信。该时间段由以下代理程序属性确定：transferAckTimeout 和 transferAckTimeoutRetries。

<numFileFailures>
传输失败的 transferSet 中的文件数。

<numFileWarnings>
传输期间出生警告但却成功传输的 transferSet 中的文件数。

示例
为以下每种类型的传输提供符合该模式的 XML 消息示例：

- [单个文件的传输](#)
- [包含多个文件的传输](#)
- [失败的文件传输](#)
- [定义有触发器的传输](#)
- [按照安排启动的传输](#)

- 调用用户出口的传输
- 通过 Web 网关请求的传输
- 通过 Connect:Direct 网桥节点的传输

相关参考

第 671 页的『单个传输日志消息示例』

进行传输时，会将消息发布到主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题。以下 XML 示例显示了已启动、正在进行和已完成的单个文件传输的日志消息。

第 673 页的『多文件传输日志消息示例』

进行包含多个文件的传输时发布至 SYSTEM.FTE 主题（含主题字符串 `Log/agent_name/transfer_ID`）的消息示例。

第 675 页的『失败的传输日志消息示例』

进行传输时，会将消息发布到主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题。XML 示例显示了传输开始时、传输过程中以及传输完成后失败的文件传输的日志消息。

第 677 页的『触发的传输消息格式』

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。以下 XML 示例显示了在启动一个包含触发条件的文件传输时创建的日志消息。

第 678 页的『用户出口消息格式』

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。该 XML 示例显示了在进行的文件传输包含对用户出口的调用时所创建的日志消息。

第 681 页的『基于 Web 的传输的消息格式中添加的内容』

来自通过 IBM MQ Managed File Transfer Web 网关 请求的传输的 Started 和 Completed 日志消息包含额外的元数据。该元数据包含有关 HTTP 请求以及有关托管 Web 网关的应用程序服务器的信息。

第 683 页的『Connect:Direct 网桥传输消息示例』

当目标代理或源代理为 Connect:Direct 网桥代理时，`destinationAgent` 或 `sourceAgent` 元素会包含更多的属性。已开始日志消息只包含一部分有关 Connect:Direct 传输的信息。进行中和已完成日志消息包含有关 Connect:Direct 传输的完整信息。

单个传输日志消息示例

进行传输时，会将消息发布到主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题。以下 XML 示例显示了已启动、正在进行和已完成的单个文件传输的日志消息。

单个文件传输 - 已启动

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d223d0020"
  agentRole="sourceAgent"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-01-26T13:03:26.484Z">started</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </sourceAgent>
  <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
  <originator>
    <hostName>dhcp-9-20-240-199.hursley.ibm.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <transferSet startTime="2011-01-26T13:03:26.484Z" total="1" bytesSent="0">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">dhcp-9-20-240-199.hursley.ibm.com.</
metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d223d0020</
metaData>
      <metaData key="com.ibm.wmqfte.ScheduleId">3</metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
</transaction>
```

```

    </metaDataSet>
  </transferSet>
  <scheduleLog ID="3"/>
</transaction>

```

单个文件传输成功 - 正在进行

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d223d0020"
  agentRole="sourceAgent"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-01-26T13:03:26.615Z">progress</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </sourceAgent>
  <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </destinationAgent>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <transferSet index="0" size="1" startTime="2011-01-26T13:03:26.484Z" total="1"
  bytesSent="1198">
    <item mode="binary">
      <source disposition="leave" type="file">
        <file size="1151" last-modified="2009-11-02T10:37:01.000Z">/etc/passwd</file>
        <checksum method="MD5">2287181c07199f879de28296371cb24c</checksum>
      </source>
      <destination type="file">
        <file size="1151" last-modified="2011-01-26T13:03:26.000Z">/tmp/passwd</file>
        <checksum method="MD5">2287181c07199f879de28296371cb24c</checksum>
      </destination>
      <status resultCode="0"/>
    </item>
  </transferSet>
</transaction>

```

单个文件传输成功 - 已完成

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d223d0020"
  agentRole="sourceAgent"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-01-26T13:03:26.622Z">completed</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </sourceAgent>
  <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </destinationAgent>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <status resultCode="0">
    <supplement>BFGRP0032I: The file transfer request has successfully completed.</supplement>
  </status>
  <transferSet startTime="2011-01-26T13:03:26.484Z" total="1" bytesSent="1198">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">example.com.</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d223d0020</
    metaDataSet>
    <metaData key="com.ibm.wmqfte.ScheduleId">3</metaData>
  </transferSet>
</transaction>

```

```

        <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
</transferSet>
<statistics>
    <actualStartTime>2011-01-26T13:03:26.541Z</actualStartTime>
    <retryCount>0</retryCount>
    <numFileFailures>0</numFileFailures>
    <numFileWarnings>0</numFileWarnings>
</statistics>
</transaction>

```

相关参考

第 677 页的『触发的传输消息格式』

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。以下 XML 示例显示了在启动一个包含触发条件的文件传输时创建的日志消息。

第 678 页的『用户出口消息格式』

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。该 XML 示例显示了在进行的文件传输包含对用户出口的调用时所创建的日志消息。

第 681 页的『基于 Web 的传输的消息格式中添加的内容』

来自通过 IBM MQ Managed File Transfer Web 网关请求的传输的 Started 和 Completed 日志消息包含额外的元数据。该元数据包含有关 HTTP 请求以及有关托管 Web 网关的应用程序服务器的信息。

第 662 页的『文件传输日志消息格式』

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 IBM MQ Managed File Transfer 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

多文件传输日志消息示例

进行包含多个文件的传输时发布至 SYSTEM.FTE 主题（含主题字符串 `Log/agent_name/transfer_ID`）的消息示例。

多个文件传输 - 已启动

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    version="4.00"
    ID="414d51205553322e42494e44494e47538b0f404d035c0020"
    agentRole="sourceAgent"
    xsi:noNamespaceSchemaLocation="TransferLog.xsd"
    xmlns="">
    <action time="2011-01-26T13:12:58.534Z">started</action>
    <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
        <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
    </sourceAgent>
    <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
    <originator>
        <hostName>example.com</hostName>
        <userID>mqm</userID>
        <mqmdUserID>mqm</mqmdUserID>
    </originator>
    <transferSet startTime="2011-01-26T13:12:58.534Z" total="6" bytesSent="0">
        <metaDataSet>
            <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
            <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
            <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
            <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
            <metaData key="com.ibm.wmqfte.OriginatingHost">example.com</metaData>
            <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d035c0020</
metaData>
            <metaData key="com.ibm.wmqfte.Priority">0</metaData>
        </metaDataSet>
    </transferSet>
</transaction>

```

多个文件传输 - 正在进行

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    version="4.00"

```

```

ID="414d51205553322e42494e44494e47538b0f404d035c0020"
agentRole="sourceAgent"
xsi:noNamespaceSchemaLocation="TransferLog.xsd"
xmlns=""
<action time="2011-01-26T13:12:58.753Z">progress</action>
<sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
  <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
</sourceAgent>
<destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
  <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
</destinationAgent>
<originator>
  <hostName>example.com.</hostName>
  <userID>mqm</userID>
  <mqmdUserID>mqm</mqmdUserID>
</originator>
<transferSet index="0" size="6" startTime="2011-01-26T13:12:58.534Z" total="6" bytesSent="440">
  <item mode="binary">
    <source disposition="leave" type="file">
      <file size="0" last-modified="2011-01-26T13:10:19.000Z">/srv/nfs/incoming/file01.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </source>
    <destination type="file">
      <file size="0" last-modified="2011-01-26T13:12:58.000Z">/srv/nfs/outgoing/file01.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </destination>
    <status resultCode="0"/>
  </item>
  <item mode="binary">
    <source disposition="leave" type="file">
      <file size="0" last-modified="2011-01-26T13:10:19.000Z">/srv/nfs/incoming/file02.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </source>
    <destination type="file">
      <file size="0" last-modified="2011-01-26T13:12:58.000Z">/srv/nfs/outgoing/file02.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </destination>
    <status resultCode="0"/>
  </item>
  <item mode="binary">
    <source disposition="leave" type="file">
      <file size="0" last-modified="2011-01-26T13:10:19.000Z">/srv/nfs/incoming/file03.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </source>
    <destination type="file">
      <file size="0" last-modified="2011-01-26T13:12:58.000Z">/srv/nfs/outgoing/file03.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </destination>
    <status resultCode="0"/>
  </item>
  <item mode="binary">
    <source disposition="leave" type="file">
      <file size="0" last-modified="2011-01-26T13:10:19.000Z">/srv/nfs/incoming/file04.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </source>
    <destination type="file">
      <file size="0" last-modified="2011-01-26T13:12:58.000Z">/srv/nfs/outgoing/file04.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </destination>
    <status resultCode="0"/>
  </item>
  <item mode="binary">
    <source disposition="leave" type="file">
      <file size="0" last-modified="2011-01-26T13:10:19.000Z">/srv/nfs/incoming/file05.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </source>
    <destination type="file">
      <file size="0" last-modified="2011-01-26T13:12:58.000Z">/srv/nfs/outgoing/file05.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </destination>
    <status resultCode="0"/>
  </item>

```



```

        <item mode="binary">
            <source disposition="leave" type="file">
                <file size="0" last-modified="2011-01-26T13:10:19.000Z">/srv/nfs/incoming/file06.txt</
file>
                <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
            </source>
            <destination type="file">
                <file size="0" last-modified="2011-01-26T13:12:58.000Z">/srv/nfs/outgoing/file06.txt</
file>
                <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
            </destination>
            <status resultCode="0"/>
        </item>
    </transferSet>
</transaction>

```

多个文件传输 - 已完成

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    version="4.00"
    ID="414d51205553322e42494e44494e47538b0f404d035c0020"
    agentRole="sourceAgent"
    xsi:noNamespaceSchemaLocation="TransferLog.xsd"
    xmlns="">
    <action time="2011-01-26T13:12:58.766Z">completed</action>
    <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
        <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
    </sourceAgent>
    <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
        <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
    </destinationAgent>
    <originator>
        <hostName>example.com.</hostName>
        <userID>mqm</userID>
        <mqmdUserID>mqm</mqmdUserID>
    </originator>
    <status resultCode="0">
        <supplement>BFGRP0032I: The file transfer request has successfully completed.</supplement>
    </status>
    <transferSet startTime="2011-01-26T13:12:58.534Z" total="6" bytesSent="440">
        <metaDataSet>
            <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
            <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
            <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
            <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
            <metaData key="com.ibm.wmqfte.OriginatingHost">example.com.</metaData>
            <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d035c0020</
metaData>
            <metaData key="com.ibm.wmqfte.Priority">0</metaData>
        </metaDataSet>
    </transferSet>
    <statistics>
        <actualStartTime>2011-01-26T13:12:58.634Z</actualStartTime>
        <retryCount>0</retryCount>
        <numFileFailures>0</numFileFailures>
        <numFileWarnings>0</numFileWarnings>
    </statistics>
</transaction>

```

失败的传输日志消息示例

进行传输时，会将消息发布到主题字符串为 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题。XML 示例显示了传输开始时、传输过程中以及传输完成后失败的文件传输的日志消息。

文件传输失败 - 已启动

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    version="4.00"
    ID="414d51205553322e42494e44494e47538b0f404d03620020"
    agentRole="sourceAgent"
    xsi:noNamespaceSchemaLocation="TransferLog.xsd"
    xmlns="">
    <action time="2011-01-26T13:19:15.767Z">started</action>
    <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">

```

```

    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </sourceAgent>
  <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <transferSet startTime="2011-01-26T13:19:15.767Z" total="1" bytesSent="0">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">example.com.</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d03620020</
metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
</transaction>

```

文件传输失败 - 进行中

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d03620020"
  agentRole="sourceAgent"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-01-26T13:19:15.944Z">progress</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </sourceAgent>
  <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </destinationAgent>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <transferSet index="0" size="1" startTime="2011-01-26T13:19:15.767Z" total="1" bytesSent="0">
    <item mode="binary">
      <source disposition="leave" type="file">
        <file size="0" last-modified="2011-01-26T13:10:19.000Z">/srv/nfs/incoming/file01.txt</
file>
        <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
      </source>
      <destination type="file">
        <file>/srv/nfs/outgoing/file01.txt</file>
      </destination>
      <status resultCode="1">
        <supplement>BFGIO0006E: File "/srv/nfs/outgoing/file01.txt" already exists.</
supplement>
      </status>
    </item>
  </transferSet>
</transaction>

```

文件传输失败 - 已完成

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d03620020"
  agentRole="sourceAgent"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-01-26T13:19:15.948Z">completed</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </sourceAgent>
  <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </destinationAgent>

```

```

<originator>
  <hostName>example.com.</hostName>
  <userID>mqm</userID>
  <mqmdUserID>mqm</mqmdUserID>
</originator>
<status resultCode="40">
  <supplement>BFGRP0034I: The file transfer request has
    completed with no files being transferred.
  </supplement>
</status>
<transferSet startTime="2011-01-26T13:19:15.767Z" total="1" bytesSent="0">
  <metaDataSet>
    <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
    <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
    <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
    <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
    <metaData key="com.ibm.wmqfte.OriginatingHost">example.com.</metaData>
    <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d03620020</
metaData>
    <metaData key="com.ibm.wmqfte.Priority">0</metaData>
  </metaDataSet>
</transferSet>
<statistics>
  <actualStartTime>2011-01-26T13:19:15.878Z</actualStartTime>
  <retryCount>0</retryCount>
  <numFileFailures>1</numFileFailures>
  <numFileWarnings>0</numFileWarnings>
</statistics>
</transaction>

```

触发的传输消息格式

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。以下 XML 示例显示了在启动一个包含触发条件的文件传输时创建的日志消息。

触发器单个文件传输成功 - 已启动

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d3120202020202020207e970d492000a102" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T22:05:18.703Z">started</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1" />
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
    <mqmdUserID>USER1 </mqmdUserID>
  </originator>
  <trigger log="yes">
    <fileExist comparison="=" value="exist">c:\trigger.txt</fileExist>
  </trigger>
  <transferSet startTime="2008-11-02T22:05:18.703Z" total="1"></transferSet>
</transaction>

```

相关参考

第 671 页的『单个传输日志消息示例』

进行传输时，会将消息发布到主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题。以下 XML 示例显示了已启动、正在进行和已完成的单个文件传输的日志消息。

第 678 页的『用户出口消息格式』

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。该 XML 示例显示了在进行的文件传输包含对用户出口的调用时所创建的日志消息。

第 681 页的『基于 Web 的传输的消息格式中添加的内容』

来自通过 IBM MQ Managed File Transfer Web 网关请求的传输的 Started 和 Completed 日志消息包含额外的元数据。该元数据包含有关 HTTP 请求以及有关托管 Web 网关的应用程序服务器的信息。

第 662 页的『文件传输日志消息格式』

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 `IBM MQ Managed File Transfer` 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

已安排的传输日志消息示例

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题发布消息。以下 XML 示例显示根据某个安排进行文件传输时创建的日志消息。

安排传输事务消息

当根据安排条目到期而启动文件传输时，文件传输将遵循惯常的顺序在 `SYSTEM.FTE/Log/agent_name` 主题上发布事务消息：

- 操作已启动 (`TransferLog.xsd`)
- 操作正在进行 (`TransferLog.xsd`)
- 操作已完成 (`TransferLog.xsd`)

只有操作状态为“已开始”的日志事务消息才会在 `<scheduleLog>` 元素的 `ID` 属性中包含已调度传输的标识。这允许在整个传输的生命周期内将安排标识与传输标识绑定在一起。

已开始：

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d3120202020202020202020202020248e294920004016" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-23T21:55:03.111Z">started</action>
  .
  .
  <scheduleLog ID="6" />
</transaction>
```

进行中：

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d3120202020202020202020202020248e294920004016" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-23T21:55:03.377Z">progress</action>
  .
  .
</transaction>
```

已完成：

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d3120202020202020202020202020248e294920004016" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-23T21:55:03.424Z">completed</action>
  .
  .
</transaction>
```

用户出口消息格式

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题发布消息。该 XML 示例显示了在进行的文件传输包含对用户出口的调用时所创建的日志消息。

继续退出单个文件传输 - 已启动

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d31202020202020202020207e970d492000d502" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T22:36:13.046Z">started</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1" />
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
    <mqmdUserID>USER1 </mqmdUserID>
  </originator>
  <transferSet startTime="2008-11-02T22:36:13.046Z" total="1">
    <metaDataSet>
      <metaData key="testkey1">testvalue1</metaData>
      <metaData key="testkey2">testvalue2</metaData>
    </metaDataSet>
  </transferSet>
</transaction>
```

继续退出单个文件传输 - 已完成

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d31202020202020202020207e970d492000d502"
  agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T22:36:13.546Z">completed</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <startExits>
      <exit name="class testExits.SourceExit1">
        <status resultCode="proceed">
          <supplement>Source Start, modified metadata</supplement>
        </status>
      </exit>
    </startExits>
    <endExits>
      <exit name="class testExits.SourceExit1">
        <status>
          <supplement>Source End</supplement>
        </status>
      </exit>
    </endExits>
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1">
    <startExits>
      <exit name="class testExits.DestinationExitProceed">
        <status resultCode="proceed">
          <supplement>Destination start, with proceed</supplement>
        </status>
      </exit>
    </startExits>
    <endExits>
      <exit name="class testExits.DestinationExitProceed">
        <status>
          <supplement>destination end</supplement>
        </status>
      </exit>
    </endExits>
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </destinationAgent>
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
    <mqmdUserID>USER1 </mqmdUserID>
  </originator>
```

```

<transferSet startTime="2008-11-02T22:36:13.046Z" total="1">
  <metaDataSet>
    <metaData key="newkey2">newvalue2</metaData>
    <metaData key="newkey1">newvalue1</metaData>
    <metaData key="newkey4">newvalue4</metaData>
    <metaData key="newkey3">newvalue3</metaData>
    <metaData key="newkey5">newvalue5</metaData>
    <metaData key="testkey1">testvalue1</metaData>
    <metaData key="testkey2">testvalue2</metaData>
  </metaDataSet>
</transferSet>
</transaction>

<!--
  In this example the source transfer start exit has modified the
  metadata as follows:

  Added keys and values for:
  newkey1, newvalue1
  newkey2, newvalue2
  newkey3, newvalue3
  newkey4, newvalue4
  newkey5, newvalue5

  Replaced values for:
  key1 to modifiedValue1

  Deleted keys and values for:
  key2
-->

```

取消退出单个文件传输 - 已取消

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d312020202020202020207e970d492000c702" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T22:25:59.328Z">cancelled</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <startExits>
      <exit name="class testExits.SourceExit1">
        <status resultCode="proceed">
          <supplement>Source Start, modified metadata</supplement>
        </status>
      </exit>
    </startExits>
    <endExits>
      <exit name="class testExits.SourceExit1">
        <status>
          <supplement>Source End</supplement>
        </status>
      </exit>
    </endExits>
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1">
    <startExits>
      <exit name="class testExits.DestinationExit1">
        <status resultCode="cancelTransfer">
          <supplement>Destination start, with cancel</supplement>
        </status>
      </exit>
    </startExits>
    <endExits>
      <exit name="class testExits.DestinationExit1">
        <status>
          <supplement>destination end</supplement>
        </status>
      </exit>
    </endExits>
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </destinationAgent>
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
  </originator>

```



```

        <mqmdUserID>USER1          </mqmdUserID>
      </originator>
      <transferSet startTime="2008-11-02T22:25:59.078Z" total="1" />
    </transaction>

```

相关参考

第 671 页的『单个传输日志消息示例』

进行传输时，会将消息发布到主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题。以下 XML 示例显示了已启动、正在进行和已完成的单个文件传输的日志消息。

第 677 页的『触发的传输消息格式』

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。以下 XML 示例显示了在启动一个包含触发条件的文件传输时创建的日志消息。

第 681 页的『基于 Web 的传输的消息格式中添加的内容』

来自通过 IBM MQ Managed File Transfer Web 网关 请求的传输的 Started 和 Completed 日志消息包含额外的元数据。该元数据包含有关 HTTP 请求以及有关托管 Web 网关的应用程序服务器的信息。

第 662 页的『文件传输日志消息格式』

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 IBM MQ Managed File Transfer 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

基于 Web 的传输的消息格式中添加的内容

来自通过 IBM MQ Managed File Transfer Web 网关 请求的传输的 Started 和 Completed 日志消息包含额外的元数据。该元数据包含有关 HTTP 请求以及有关托管 Web 网关的应用程序服务器的信息。

Web 元数据的定义

com.ibm.wmqfte.web.request.authtype

向 Web 网关提交请求的用户使用的授权方法。

com.ibm.wmqfte.web.request.locale

向 Web 网关提交请求的用户的语言环境。

com.ibm.wmqfte.web.appsrv.type

托管 Web 网关的应用程序服务器的类型。

com.ibm.wmqfte.web.appsrv.host

系统的主机名或 IP 地址，该系统运行托管 Web 网关的应用程序服务器。

com.ibm.wmqfte.web.appsrv.port

托管 Web 网关的应用程序服务器所侦听的端口号。

以下示例中突出显示了通过 Web 网关所请求传输的日志消息中包括的元数据。

单个文件传输 - 成功

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d312020202020202020207e970d4920008202" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T21:20:37.578Z">started</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1" />
  <originator>
    <hostName>requestor.example.com</hostName>
    <userID>USER1 </userID>
    <mqmdUserID>USER1</mqmdUserID>
  </originator>
  <transferSet startTime="2008-11-02T21:20:37.593Z" total="1">

```

单个文件传输成功 - 已完成

注：注：此处描述的 XML 消息格式与 Web 网关所返回响应的消息格式不同。以下主题中描述了 Web 网关返回的 XML 格式：第 919 页的『响应格式：XML 和 JSON』。

第 671 页的『单个传输日志消息示例』

第 677 页的『触发的传输消息格式』

第 678 页的『用户出口消息格式』

第 662 页的『文件传输日志消息格式』

682 Managed File Transfer

Connect:Direct 网桥传输消息示例

当目标代理或源代理为 Connect:Direct 网桥代理时，destinationAgent 或 sourceAgent 元素会包含更多的属性。已开始日志消息只包含一部分有关 Connect:Direct 传输的信息。进行中和已完成日志消息包含有关 Connect:Direct 传输的完整信息。

源代理为 Connect:Direct 网桥代理

已开始:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  ID="414d5120514d5f696b6b796f20202020a704654d20092507"
  agentRole="sourceAgent"
  version="4.00"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-03-07T13:05:01.838Z">started</action>
  <sourceAgent QMgr="QM_KUIPER" agent="VARUNA" agentType="CD_BRIDGE" bridgeNode="CDNODE_VARUNA">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="QM_KUIPER" agent="IXION"/>
  <originator>
    <hostName>kuiper.example.com.</hostName>
    <userID>sol</userID>
    <mqmdUserID>sol</mqmdUserID>
  </originator>
  <transferSet bytesSent="0" startTime="2011-03-07T13:05:01.838Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">VARUNA</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">IXION</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">kuiper.example.com.</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d5120514d5f696b6b796f20202020a704654d20092507</
metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
</transaction>
```

进行中:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  ID="414d5120514d5f696b6b796f20202020a704654d20092507"
  agentRole="sourceAgent"
  version="4.00"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-03-07T13:05:03.448Z">progress</action>
  <sourceAgent QMgr="QM_KUIPER" agent="VARUNA" agentType="CD_BRIDGE"
    bridgeNode="CDNODE_VARUNA" pnode="CDNODE_VARUNA" snode="CDNODE_ERIS">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="QM_KUIPER" agent="IXION" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </destinationAgent>
  <originator>
    <hostName>kuiper.example.com.</hostName>
    <userID>sol</userID>
    <mqmdUserID>sol</mqmdUserID>
  </originator>
  <transferSet bytesSent="48" index="0" size="1" startTime="2011-03-07T13:05:01.838Z" total="1">
    <item mode="binary">
      <source disposition="leave" processName="f2007567" processNumber="68" type="file">
        <file last-modified="2011-03-07T13:05:02.573Z" size="4">CDNODE_ERIS:D:/AGENTS/
CDNODE_ERIS/test.txt</file>
        <checksum method="MD5">098f6bcd4621d373cade4e832627b4f6</checksum>
      </source>
      <destination type="file">
        <file last-modified="2011-03-07T13:05:03.338Z" size="4">D:\AGENTS\IXION\test.txt</file>
        <checksum method="MD5">098f6bcd4621d373cade4e832627b4f6</checksum>
      </destination>
      <status resultCode="0"/>
    </item>
  </transferSet>
</transaction>
```

已完成:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  ID="414d5120514d5f696b6b796f20202020a704654d20092507"
  agentRole="sourceAgent"
  version="4.00" xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-03-07T13:05:03.495Z">completed</action>
  <sourceAgent QMgr="QM_KUIPER" agent="VARUNA" agentType="CD_BRIDGE"
    bridgeNode="CDNODE_VARUNA" pnode="CDNODE_VARUNA" snode="CDNODE_ERIS">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="QM_KUIPER" agent="IXION" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </destinationAgent>
  <originator>
    <hostName>kuiper.example.com.</hostName>
    <userID>sol</userID>
    <mqmdUserID>sol</mqmdUserID>
  </originator>
  <status resultCode="0">
    <supplement>BFGRP0032I: The file transfer request has successfully completed.</supplement>
  </status>
  <transferSet bytesSent="48" startTime="2011-03-07T13:05:01.838Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">VARUNA</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">IXION</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">kuiper.example.com.</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d5120514d5f696b6b796f20202020a704654d20092507</
metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
  <statistics>
    <actualStartTime>2011-03-07T13:05:02.041Z</actualStartTime>
    <retryCount>0</retryCount>
    <numFileFailures>0</numFileFailures>
    <numFileWarnings>0</numFileWarnings>
  </statistics>
</transaction>
```

目标代理为 Connect:Direct 网桥代理

已开始:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  ID="414d5120514d5f696b6b796f20202020a704654d2008e102"
  agentRole="sourceAgent"
  version="4.00"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-03-07T10:29:44.854Z">started</action>
  <sourceAgent QMgr="QM_asteroid" agent="PALLAS" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="QM_asteroid" agent="VESTA"/>
  <originator>
    <hostName>belt.example.com.</hostName>
    <userID>sol</userID>
    <mqmdUserID>sol</mqmdUserID>
  </originator>
  <transferSet bytesSent="0" startTime="2011-03-07T10:29:44.854Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">PALLAS</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">VESTA</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">belt.example.com.</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d5120514d5f696b6b796f20202020a704654d2008e102</
metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
</transaction>
```

进行中:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  ID="414d5120514d5f696b6b796f20202020a704654d2008e102"
  agentRole="sourceAgent"
  version="4.00"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-03-07T10:29:46.682Z">progress</action>
  <sourceAgent QMgr="QM_asteroid" agent="PALLAS" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="QM_asteroid" agent="VESTA" agentType="CD_BRIDGE"
    bridgeNode="CDNODE_VESTA" pNode="CDNODE_VESTA" sNode="CDNODE_HYGIEA">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </destinationAgent>
  <originator>
    <hostName>belt.example.com</hostName>
    <userID>sol</userID>
    <mqmdUserID>sol</mqmdUserID>
  </originator>
  <transferSet bytesSent="48" index="0" size="1" startTime="2011-03-07T10:29:44.854Z" total="1">
    <item mode="binary">
      <source disposition="leave" type="file">
        <file last-modified="2011-03-04T14:53:28.323Z" size="4">D:\AGENTS\PALLAS\test.txt</
file>
        <checksum method="MD5">098f6bcd4621d373cade4e832627b4f6</checksum>
      </source>
      <destination processName="f2006965" processNumber="59" type="file">
        <file size="4">CDNODE_VESTA:D:/AGENTS/CDNODE_VESTA/test.txt</file>
        <checksum method="MD5">098f6bcd4621d373cade4e832627b4f6</checksum>
      </destination>
      <status resultCode="0"/>
    </item>
  </transferSet>
</transaction>
```

已完成:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  ID="414d5120514d5f696b6b796f20202020a704654d2008e102"
  agentRole="sourceAgent"
  version="4.00"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-03-07T10:29:46.698Z">completed</action>
  <sourceAgent QMgr="QM_asteroid" agent="PALLAS" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="QM_asteroid" agent="VESTA" agentType="CD_BRIDGE"
    bridgeNode="CDNODE_VESTA" pNode="CDNODE_VESTA" sNode="CDNODE_HYGIEA">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </destinationAgent>
  <originator>
    <hostName>belt.example.com</hostName>
    <userID>sol</userID>
    <mqmdUserID>sol</mqmdUserID>
  </originator>
  <status resultCode="0">
    <supplement>BFGRP0032I: The file transfer request has successfully completed.</supplement>
  </status>
  <transferSet bytesSent="48" startTime="2011-03-07T10:29:44.854Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">PALLAS</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">VESTA</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">belt.example.com.</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d5120514d5f696b6b796f20202020a704654d2008e102</
metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
  <statistics>
    <actualStartTime>2011-03-07T10:29:45.010Z</actualStartTime>
    <retryCount>0</retryCount>
    <numFileFailures>0</numFileFailures>
    <numFileWarnings>0</numFileWarnings>
```

```
</statistics>
</transaction>
```

已安排的传输日志消息格式

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/agent name/schedule ID 主题中）。该消息符合 ScheduleLog.xsd XML 模式。

模式

以下模式描述哪些元素在调度日志 XML 消息中有效。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>

  <xsd:element name="schedulelog">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="originator" type="hostUserIDType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="action" type="actionType"
          maxOccurs="1" minOccurs="1"/>
        <xsd:element name="schedule" type="scheduleType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceAgent" type="agentType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="destinationAgent" type="agentClientType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="status" type="statusType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="transferSet" type="transferSetType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="job" type="jobType"
          maxOccurs="1" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="ID" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="actionType">
    <xsd:simpleContent>
      <xsd:extension base="actionEnumType">
        <xsd:attribute name="time" type="xsd:dateTime" use="required" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>

  <xsd:simpleType name="actionEnumType">
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="submit"/>
      <xsd:enumeration value="delete"/>
      <xsd:enumeration value="expire"/>
      <xsd:enumeration value="skipped"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:complexType name="transferSetType">
    <xsd:sequence>
      <xsd:element name="item" type="itemType"
        maxOccurs="unbounded" minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="size" type="xsd:int" use="required" />
    <xsd:attribute name="priority" type="priorityType" use="optional" />
  </xsd:complexType>

  <xsd:complexType name="itemType">
    <xsd:sequence>
      <xsd:element name="source" type="fileSourceType"
        maxOccurs="1" minOccurs="1" />
      <xsd:element name="destination" type="fileDestinationType"
        maxOccurs="1" minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="mode" type="modeType" use="required" />
    <xsd:attribute name="checksumMethod" type="checksumMethod" use="required" />
  </xsd:complexType>
```


</xsd:schema>

了解调度日志消息

下面描述了调度日志消息中使用的元素和属性：

<schedulelog>

用于描述单个提交的已调度文件传输的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 的详细说明指定该元素的版本。
标识	已提交的调度文件传输的唯一标识。

<originator>

包含指定请求发起方的元素的组元素。

<hostName>

源文件所在系统的主机名。

<userID>

发起文件传输的用户标识。

<mqmdUserID>

在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<action>

指定要对与 <schedulelog> 元素的 ID 属性匹配的已调度传输执行的操作。该元素必须是以下某个值：

- submit - 新建已调度传输
- delete - 取消调度传输
- expire - 即将处理调度传输条目
- skipped - 由于代理处于脱机状态，因此无法启动调度的传输。当代理变得可用时，将记录该消息以指明跳过了该传输。

属性	描述
time	指定发布日志条目的日期和时间（采用日期时间格式）。

<sourceAgent>

指定源文件所在系统上的代理的名称。

属性	描述
代理程序	指定代理的名称。
QMgr	代理队列管理器的名称。

<destinationAgent>

指定您要向其传输文件的系统上代理的名称。

属性	描述
代理程序	指定代理的名称。
QMgr	代理队列管理器的名称。

<status>

结果代码和补充消息。

<transferSet>

指定一组要一起执行的文件传输。在传输期间，<transferSet> 是包含 <item> 元素的组元素。

属性	描述
size	指定传输项数。
priority	传输的优先级。优先级为 0-9 之间的一个值，其中 0 为最低优先级。缺省优先级为 0，并且缺省情况下，传输使用源代理的优先级。

<item>

包含指定源和目标文件名和位置的元素的组元素。

属性	描述
mode	将传输方式指定为二进制或文本。
checksumMethod	指定生成消息摘要以创建数字签名的散列算法的类型。允许的值包括 MD5 或 none

<source>

包含源系统上文件的 <file> 和 <checksum> 元素的组元素。

属性	描述
recursive	指定当 <source> 元素为目录或包含通配符时，以递归方式传输子目录中的文件。
disposition	指定在 <source> 已成功传输到其目标时要在 <source> 元素上执行的操作。有效选项如下： • leave - 源文件保持不变。 • delete - 在成功传输源文件后将源文件从源系统中删除。

<destination>

包含目标系统上文件的 <file> 和 <checksum> 元素的组元素。

属性	描述
类型	位于目标的文件或目录的类型。有效选项如下： • file - 将文件指定为目标 • directory - 将目录指定为目标 • dataset - 指定 z/OS 数据集作为目标 • PDS - 将 z/OS 分区数据集指定为目标
exist	指定在目标系统上已存在目标文件时执行的操作。有效选项如下： • error - 报告一个错误，并且未传输文件。 • overwrite - 覆盖现有的目标文件。

<file>

指定要传输的文件的名称。使用与操作系统格式一致的标准路径，例如 C:/from/here.txt。请勿使用文件 URI。

属性	描述
编码	文本文件传输的编码。
EOL	指定行结束标记符。允许的值包括： • LF - 仅限换行字符 • CRLF - 回车符和换行符序列

<job>

包含用于指定作业详细信息的元素的组元素。 <job> is a user-defined job name identifier that is added to the log message when the transfer has started. This <job> element is the same as the <job> element that is included in the transfer request message, which is described in the following topic: [第 842 页的『文件传输请求消息格式』](#)。

<name>

名称可以是任意字符串。

示例

为以下每个已调度传输操作提供了符合该模式的 XML 消息示例：

- [创建已调度传输](#)
- [取消已调度传输](#)
- [使已调度传输到期](#)

根据调度启动的传输与标准传输采用相同的记录方式。 要获取根据调度启动的传输的日志消息的示例，请参阅 [第 678 页的『已安排的传输日志消息示例』](#)。

相关参考

[第 647 页的『代理状态消息格式』](#)

在创建或启动代理时，该代理会向其协调队列管理器上的 SYSTEM.FTE 主题（在 SYSTEM.FTE/Agents/*agent name* 主题上）发布其详细信息。

[第 842 页的『文件传输请求消息格式』](#)

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。 The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 *MQ_INSTALLATION_PATH/mqft/samples/schema* 目录中。 FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

[第 658 页的『文件传输状态消息格式』](#)

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。 每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 SYSTEM.FTE/Transfers/*agent_name/transfer ID* 主题中），它符合 TransferStatus.xsd XML 模式。 TransferStatus.xsd 文件位于 WMQMFT 安装的 *MQ_INSTALLATION_PATH/mqft/samples/schema* 目录中。

[第 662 页的『文件传输日志消息格式』](#)

文件传输日志消息发布在主题字符串为 *Log/agent_name/transfer_ID* 的 SYSTEM.FTE 主题内。 这些消息符合模式 TransferLog.xsd，该模式位于 IBM MQ Managed File Transfer 安装的 *MQ_INSTALLATION_PATH/mqft/samples/schema* 目录中。

[第 859 页的『监视器请求消息格式』](#)

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 fteCreateMonitor 命令或使用 WebSphere MQ Explorer 接口。

[第 871 页的『针对安全性的消息格式』](#)

本主题描述发布到协调队列管理器且与安全性相关的消息。

安排日志示例

执行已安排的传输操作时，发布到主题字符串为 *Log/agent_name/schedule_ID* 的 SYSTEM.FTE 主题的消息示例。

已安排的传输日志消息

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/*agent name/schedule ID* 主题中）。 该消息符合 ScheduleLog.xsd XML 模式。 例如：

```
<?xml version="1.0" encoding="UTF-8"?>
<schedulelog version="1.00" ID="5"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
```

```

xsi:noNamespaceSchemaLocation="ScheduleLog.xsd">
<originator>
  <hostName>reportserver.com</hostName>
  <userID>USER1</userID>
</originator>
<action time="2008-11-23T21:32:01Z">submit</action>
<schedule>
  <submit timebase="admin" timezone="Europe/London">2008-11-23T22:00</submit>
</schedule>
<sourceAgent agent="FTEAGENT" QMgr="QM1" />
<destinationAgent agent="FTEAGENT" QMgr="QM1" />
<status resultCode="0" />
<transferSet size="1" priority="0">
  <item mode="binary" checksumMethod="MD5">
    <source recursive="false" disposition="leave">
      <file>c:\sourcefiles\source1.doc</file>
    </source>
    <destination type="file" exist="overwrite">
      <file>c:\destinationfiles\dest1.doc</file>
    </destination>
  </item>
</transferSet>
</schedulelog>

```

该消息是一个包含以下信息的日志：

- 发起请求的人员
- 提交请求的时间
- 已安排传输的启动时间
- 源和目标代理详细信息
- 传输规范

<schedulelog> 元素的 ID 属性是该已调度传输的唯一标识（在源代理中）。该标识用于将安排条目与实际的文件传输关联。

提交的 <action> 元素值确认已收到请求。

已安排的传输取消日志消息

代理收到取消暂挂的已安排文件传输的请求时，会将以下消息发布到 SYSTEM.FTE/Log/agent_name 主题：

```

<?xml version="1.0" encoding="UTF-8"?>
<schedulelog version="1.00" ID="5"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ScheduleLog.xsd">
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
  </originator>
  <action time="2008-11-23T21:56:27Z">delete</action>
  <status resultCode="0" />
</schedulelog>

```

ID 属性值对应于安排消息中暂挂的传输请求的标识。

已安排的传输到期日志消息

当前时间与调度列表中最早的暂挂文件传输的时间匹配（由 <next> 元素的值指示）时，将发布调度日志消息以指示已调度传输条目已到期：

```

<?xml version="1.0" encoding="UTF-8"?>
<schedulelog xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00" ID="3"
  xsi:noNamespaceSchemaLocation="ScheduleLog.xsd">
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>

```

```
<action time="2011-01-26T13:03:26Z">expire</action>
<sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
<status resultCode="0"/>
</schedulelog>
```

“expire”的 <action> 元素值确认调度条目现在已从调度列表中移除，并且即将进行处理。将发布代理的安排消息，不再存在到期条目。

相关参考

第 686 页的『已安排的传输日志消息格式』

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/agent name/schedule ID 主题中）。该消息符合 ScheduleLog.xsd XML 模式。

第 678 页的『已安排的传输日志消息示例』

在进行传输的过程中，会向具有主题字符串 Log/agent_name/transfer_ID 的 SYSTEM.FTE 主题发布消息。以下 XML 示例显示根据某个安排进行文件传输时创建的日志消息。

监视器日志消息格式

监视器日志消息发布至 SYSTEM.FTE 主题（含主题字符串 Log/agent_name/Monitors/monitor_name/monitor_ID）。

如果要收集数据或查看监视器操作，请将预订设置为适合您感兴趣的监视器的通配符主题。例如：

```
Log/#
```

或者，

```
Log/agent_name/#
```

该预订可以是持久或非持久的。持久预订在预订应用程序与队列管理器之间的连接关闭后继续存在。非持久预订只有在预订应用程序与队列管理器之间的连接处于打开状态时才存在。

MonitorLog.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。MonitorLog.xsd 模式导入位于同一目录中的 fteutils.xsd。

模式

以下模式描述了监视器日志 XML 消息中哪些元素有效。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>
  <xsd:element name="monitorLog">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="originator" type="hostUserIDType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="references" type="referencesType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="action" type="monitorActionType" maxOccurs="1"
minOccurs="1"/>
        <xsd:element name="monitorAgent" type="agentType" maxOccurs="1"
minOccurs="1"/>
        <xsd:element name="status" type="statusType" maxOccurs="1"
minOccurs="1"/>
        <xsd:element name="monitorMetaData" type="monitorMetaDataType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="monitorExits" type="exitGroupType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="jobDetails" type="jobType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="taskXMLRequest" type="taskXMLRequestType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="monitorXMLRequest" type="monitorXMLRequestType"
maxOccurs="1" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="monitorName" type="xsd:string" use="required"/>
    </complexType>
  </xsd:element>
</xsd:schema>
```

```

        <xsd:attribute name="referenceId" type="xsd:string" use="optional"/>
    </xsd:complexType>
</xsd:element>

<xsd:complexType name="monitorActionType">
    <xsd:simpleContent>
        <xsd:extension base="monitorActionEnumType">
            <xsd:attribute name="time" type="xsd:dateTime" use="required" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="monitorActionEnumType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="create"/>
        <xsd:enumeration value="delete"/>
        <xsd:enumeration value="start"/>
        <xsd:enumeration value="stop"/>
        <xsd:enumeration value="triggerSatisfied"/>
        <xsd:enumeration value="triggerNotSatisfied"/>
        <xsd:enumeration value="triggerFail"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="monitorMetaDataType">
    <xsd:sequence>
        <xsd:element name="originalMetaData" type="metaDataSetType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="updatedMetaData" type="metaDataSetType" maxOccurs="unbounded"
minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="taskXMLRequestType">
    <xsd:sequence>
        <xsd:element name="originalRequest" type="xsd:string" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="updatedRequest" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="taskId" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="referencesType">
    <xsd:sequence>
        <xsd:element name="createRequest" type="xsd:string" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="taskRequest" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="monitorXMLRequestType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="type" type="xmlContentEnumType" use="required" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="xmlContentEnumType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="escapedXML"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

了解监视器日志消息

以下列表中描述了监视器日志消息中使用的元素和属性：

<monitorLog>

组元素，包含描述监视器已执行的操作的元素。

属性	描述
version	必需。监视器列表消息格式的版本。
monitorName	必需。监视器的名称。对于定义监视器所在的代理是唯一的。
referenceId	监视器操作的标识。

<originator>

包含指定请求发起方的元素的组元素。

<hostName>

源文件所在系统的主机名。

<userID>

发起文件传输的用户标识。

<mqmdUserID>

可选。在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<references>

与该监视器操作相关联的其他消息的标识的引用。

<createRequest>

用于创建监视器的 XML 请求消息的消息标识。

<taskRequest>

由于此操作，监视器提交的 XML 请求消息的消息标识。

<action>

发生的与该日志消息相关联的操作。元素内的值可以是以下某个值：create、delete、start、stop、triggerSatisfied、triggerNotSatisfied 或 triggerFail。

<monitorAgent>

正在监视资源的代理。

属性	描述
代理程序	必需。代理的名称。
QMgr	可选。代理所连接的队列管理器的名称。
bridgeURL	可选。如果代理是协议网桥代理，那么为协议服务器的 URL。

<status>

正在记录的资源监视器操作的状态。

属性	描述
resultCode	必需。操作的整数结果代码。

<supplement>

有关正在记录的资源监视器操作的状态的其他信息。

<monitorMetaData>

包含 <originalMetaData> 和 <updatedMetaData> 元素的组元素。

<originalMetaData>

包含一个或多个描述操作发生前的监视器元数据的 <metadata> 元素。

<updatedMetaData>

包含一个或多个描述操作发生后的监视器元数据的 <metadata> 元素。

<metadata>

定义元数据键值对。键是元素的属性；值是元素的内容。

属性	描述
key	元数据的键。

<monitorExits>

包含一个或多个 <exit> 元素的组元素。

<exits>

描述资源监视器运行的出口的元素。

属性	描述
名	必需。资源监视器出口的名称。

<status>

正在记录的资源监视器出口的状态。

属性	描述
resultCode	必需。出口的整数结果代码。

<supplement>

有关正在记录的资源监视器出口的状态的其他信息。

<jobDetails>

包含一个 <name> 元素的元素。

<name>

作业的名称。

<taskXMLRequest>

包含 <originalRequest> 和 <updatedRequest> 元素的组元素。

属性	描述
taskId	任务请求消息的标识。

<originalRequest>

包含监视器执行的任务的转义 XML 请求消息的元素。

<updatedRequest>

包含监视器执行的任务的更新转义 XML 请求消息的元素。

<monitorXMLRequest>

监视器 XML 请求。

属性	描述
类型	必需。<monitorXMLRequest> 元素内的监视器 XML 请求数据的格式。唯一有效的值是 escapedXML。

示例

针对以下每项监视器操作提供了符合该模式的 XML 消息的示例：

- [已创建监视器](#)
- [监视器轮询资源时满足监视器的条件](#)
- [监视器轮询资源时不满足监视器的条件](#)
- [已删除监视器](#)

相关参考

第 694 页的『监视器日志示例』

监视操作发生时发布至 SYSTEM.FTE 主题（含主题字符串 Log/agent_name/monitor_ID）的消息示例。

监视器日志示例

监视操作发生时发布至 SYSTEM.FTE 主题（含主题字符串 Log/agent_name/monitor_ID）的消息示例。

监视器创建的日志消息

```
<?xml version="1.0" encoding="UTF-8"?>
<monitorLog xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  monitorName="MONITORTWO"
```

```

        referenceId="414d51205553322e42494e44494e47538b0f404d04410020"
        xsi:noNamespaceSchemaLocation="MonitorLog.xsd">
<originator>
  <hostName>example.com.</hostName>
  <userID>mqm</userID>
  <mqmdUserID>mqm</mqmdUserID>
</originator>
<references>
  <createRequest>414d51205553322e42494e44494e47538b0f404d04410020</createRequest>
</references>
<action time="2011-01-26T12:41:24Z">start</action>
<monitorAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
<status resultCode="0"/>
</monitorLog>

```

符合监视器条件的日志消息

```

<?xml version="1.0" encoding="UTF-8"?>
<monitorLog xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  monitorName="MONITORONE"
  referenceId="414d51205553322e42494e44494e47538b0f404d09430020"
  xsi:noNamespaceSchemaLocation="MonitorLog.xsd">
<originator>
  <hostName>example.com.</hostName>
  <userID>mqm</userID>
  <mqmdUserID>mqm</mqmdUserID>
</originator>
<references>
  <createRequest>414d51205553322e42494e44494e47538b0f404d09430020</createRequest>
</references>
<action time="2011-01-26T12:56:46Z">triggerSatisfied</action>
<monitorAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
<status resultCode="0"/>
<monitorMetaData>
  <originalMetaData>
    <metaData key="AGENTNAME">AGENT_JUPITER</metaData>
    <metaData key="LASTMODIFIEDDATEUTC">2011-01-26</metaData>
    <metaData key="CURRENTTIMESTAMPUTC">20110126125646793</metaData>
    <metaData key="CURRENTTIMESTAMP">20110126125646793</metaData>
    <metaData key="LASTMODIFIEDDATE">2011-01-26</metaData>
    <metaData key="FILENAME">new.completed</metaData>
    <metaData key="LASTMODIFIEDTIMEUTC">12.56</metaData>
    <metaData key="LASTMODIFIEDTIME">12.56</metaData>
    <metaData key="FILESIZE">0</metaData>
    <metaData key="FILEPATH">/srv/nfs/incoming/new.completed</metaData>
  </originalMetaData>
  <updatedMetaData>
    <metaData key="AGENTNAME">AGENT_JUPITER</metaData>
    <metaData key="LASTMODIFIEDDATEUTC">2011-01-26</metaData>
    <metaData key="CURRENTTIMESTAMPUTC">20110126125646793</metaData>
    <metaData key="CURRENTTIMESTAMP">20110126125646793</metaData>
    <metaData key="LASTMODIFIEDDATE">2011-01-26</metaData>
    <metaData key="FILENAME">new.completed</metaData>
    <metaData key="LASTMODIFIEDTIMEUTC">12.56</metaData>
    <metaData key="LASTMODIFIEDTIME">12.56</metaData>
    <metaData key="FILESIZE">0</metaData>
    <metaData key="FILEPATH">/srv/nfs/incoming/new.completed</metaData>
  </updatedMetaData>
</monitorMetaData>
<taskXMLRequest taskId="null">
  <originalRequest>&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;request
    xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="4.00"
    xsi:noNamespaceSchemaLocation="FileTransfer.xsd"&gt;&lt;managedTransfer&gt;
      &lt;originator&gt;&lt;hostName&gt;example.com.&lt;/hostName&gt;
      &lt;userID&gt;mqm&lt;/userID&gt;&lt;/originator&gt;
      &lt;sourceAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/&gt;
      &lt;destinationAgent QMgr="QM_JUPITER" agent="AGENT_SATURN"/&gt;
      &lt;transferSet&gt;&lt;item checksumMethod="MD5" mode="binary"&gt;
        &lt;file&gt;/srv/nfs/incoming/*.txt&lt;/file&gt;&lt;/source&gt;
        &lt;destination exist="error" type="directory"&gt;
          &lt;file&gt;/srv/backup&lt;/file&gt;&lt;/destination&gt;
        &lt;/item&gt;&lt;/transferSet&gt;&lt;/managedTransfer&gt;&lt;/request&gt;
      &lt;/originalRequest>
    &lt;updatedRequest>&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;request
      xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="4.00"
      xsi:noNamespaceSchemaLocation="FileTransfer.xsd"&gt;&lt;managedTransfer&gt;
        &lt;originator&gt;&lt;hostName&gt;example.com.&lt;/hostName&gt;

```

```

        <!--sourceAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/>
        <!--destinationAgent QMgr="QM_JUPITER" agent="AGENT_SATURN"/>
        <!--transferSet--><!--item checksumMethod="MD5" mode="binary">
        <!--source disposition="leave" recursive="false">
        <!--file-->/srv/nfs/incoming/*.txt<!--/file-->
        <!--/source--><!--destination exist="error" type="directory">
        <!--file-->/srv/backup<!--/file--><!--/destination-->
        <!--item--><!--/transferSet--><!--/managedTransfer--><!--/request-->
    </updatedRequest>
</taskXMLRequest>
</monitorLog>

```

不符合监视器条件的日志消息

```

<?xml version="1.0" encoding="UTF-8"?>
<monitorLog xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    version="4.00"
    monitorName="MONITORONE"
    referenceId="414d51205553322e42494e44494e47538b0f404d09430020"
    xsi:noNamespaceSchemaLocation="MonitorLog.xsd">
    <originator>
        <hostname>example.com.</hostname>
        <userID>mqm</userID>
        <mqmdUserID>mqm</mqmdUserID>
    </originator>
    <references>
        <createRequest>414d51205553322e42494e44494e47538b0f404d09430020</createRequest>
    </references>
    <action time="2011-01-26T12:58:46Z">triggerNotSatisfied</action>
    <monitorAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
    <status resultCode="0"/>
</monitorLog>

```

监视器删除的日志消息

```

<?xml version="1.0" encoding="UTF-8"?>
<lst:monitorList xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    xmlns:lst="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
    version="4.00"
    agent="AGENT_JUPITER"
    monitor="MONITORONE"
    xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition
MonitorList.xsd">
    <status state="deleted"/>
    <configuration>
        <description/>
        <resources>
            <directory recursionLevel="0" id="">/srv/nfs/incoming</directory>
        </resources>
        <triggerMatch>
            <conditions>
                <condition>
                    <name/>
                    <resource id=""/>
                    <fileMatch>
                        <pattern>*.completed</pattern>
                    </fileMatch>
                </condition>
            </conditions>
        </triggerMatch>
        <tasks>
            <task>
                <name/>
                <description/>
                <taskXML><?xml version="1.0" encoding="UTF-8"?><!--request
                    xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="4.00"
                    xsi:noNamespaceSchemaLocation="FileTransfer.xsd"><!--managedTransfer-->
                    <!--originator--><!--hostname-->example.ibm.com.<!--/hostname-->
                    <!--userID-->mqm<!--/userID--><!--originator-->
                    <!--destinationAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/>
                    <!--destinationAgent QMgr="QM_JUPITER" agent="AGENT_SATURN"/>
                    <!--transferSet--><!--item checksumMethod="MD5" mode="binary">
                    <!--source disposition="leave" recursive="false">
                    <!--file-->/srv/nfs/incoming/*.txt<!--/file--><!--/source-->
                    <!--destination exist="error" type="directory">
                    <!--file-->/srv/backup<!--/file--><!--/destination-->

```

```

        <!--/item--><!--/transferSet--><!--/managedTransfer--><!--/request-->
      </taskXML>
    </task>
  </tasks>
</configuration>
<pollInterval units="minutes">1</pollInterval>
<batch maxSize="1"/>
</lst:monitorList>

```

IBM MQ Managed File Transfer 的代理队列

由 **fteCreateAgent** 命令生成的 MQSC 命令脚本使用设置为以下值的参数创建代理队列。如果不使用提供用于创建队列的 MQSC 脚本，而是手动创建队列，那么请确保将以下参数设置为给定的值。

代理操作队列

代理的操作队列具有以下名称：

- SYSTEM.FTE.COMMAND.*agent_name*
- SYSTEM.FTE.DATA.*agent_name*
- SYSTEM.FTE.EVENT.*agent_name*
- SYSTEM.FTE.REPLY.*agent_name*
- SYSTEM.FTE.STATE.*agent_name*

表 56: 代理操作队列参数

参数	值（如果适用）
DEFPRTY	0
DEFSOPT	SHARED
GET	ENABLED
MAXDEPTH	5000
MAXMSGL	4194304
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
USAGE	正常
替换	

代理权限队列

代理的权限队列具有以下名称：

- SYSTEM.FTE.AUTHADM1.*agent_name*
- SYSTEM.FTE.AUTHAGT1. 代理名称
- SYSTEM.FTE.AUTHMON1.*agent_name*
- SYSTEM.FTE.AUTHOPS1.*agent_name*
- SYSTEM.FTE.AUTHSCH1.*agent_name*
- SYSTEM.FTE.AUTHTRN1.*agent_name*

表 57: 代理权限队列参数	
参数	值 (如果适用)
DEFPRTY	0
DEFSOPT	SHARED
GET	ENABLED
MAXDEPTH	0
MAXMSGL	0
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
USAGE	正常
替换	

Web 代理操作队列

如果代理为 Web 代理，那么将具有两个额外的队列。这些队列具有以下名称：

- SYSTEM.FTE.WEB.gateway_name
- SYSTEM.FTE.WEB.RESP.agent_name

表 58: Web 代理操作队列参数	
参数	值 (如果适用)
DEFPRTY	0
DEFSOPT	SHARED
GET	ENABLED
MAXDEPTH	5000
MAXMSGL	4194304
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
USAGE	正常
替换	

相关参考

第 460 页的『fteCreateAgent (创建 IBM MQ Managed File Transfer 代理)』
fteCreateAgent 命令创建代理及其关联的配置。

系统队列和系统主题

IBM MQ Managed File Transfer 包含大量系统队列和一个仅限内部使用的系统主题。

名称以 SYSTEM.FTE 是 Managed File Transfer (MFT) 的内部系统队列。请勿删除这些队列，因为这样做会阻止 IBM MQ MFT 正常工作。

如果代理正在参与消息到文件或文件到消息传输，那么可能需要修改 `SYSTEM.FTE.STATE.agent_name` 队列的定义以允许执行这些受管传输。有关此信息的更多信息，请参阅 [有关设置与消息大小相关联的 MQ 属性和 MFT 属性的指南](#)。



注意: 不应更改其他系统队列的定义。

此外，请勿修改或删除 SYSTEM.FTE 主题，因为这也仅供内部使用。

临时队列

IBM MQ Managed File Transfer 针对各种用途创建临时队列。每个队列的名称以 WMQFTE 开头。缺省情况下。（句点是缺省前缀的一部分。）如果想要更改此前缀，您可以在 `command.properties` 文件和/或 `coordination.properties` 文件中使用 **dynamicQueuePrefix** 属性。`command.properties` 文件中的属性用于设置为响应需要代理响应的命令而创建的临时队列的前缀。`coordination.properties` 文件中的属性用于设置为其他目的创建的临时队列的前缀；例如，`WMQFTE.FTE.TIMECHCK.QUEUE`，其中 `WMQFTE` 是 **dynamicQueuePrefix** 属性定义的值。

IBM MQ Managed File Transfer 的对象命名约定

请针对 IBM MQ Managed File Transfer 对象使用以下命名约定：

- 代理名称最长可以有 28 个字符，并且不区分大小写。以小写形式或大小写混合形式输入的代理名称将转换为大写形式。代理名称必须符合标准的 WebSphere MQ 对象命名约定。以下详细描述了这些约定：[命名 WebSphere MQ 对象的规则](#)。
- 除 WebSphere MQ 对象命名约定外，代理名称中不能使用正斜杠 (/) 字符。
- 除 WebSphere MQ 对象命名约定外，代理名称中不能使用百分号 (%) 字符。
- 属性文件中的属性名称区分大小写。
- 队列管理器名称区分大小写。
- 对于某些平台，文件名区分大小写。
- 资源监视器名称不区分大小写。以小写或大小写混合方式输入的资源监视器名称将转换为大写。资源监视器名称不得包含星号 (*)，百分号 (%) 或问号 (?) 字符。
- 协议文件服务器名称必须最少为 2 个字符，但没有最大长度限制，并且不区分大小写。协议服务器名称必须符合标准的 WebSphere MQ 对象命名约定。以下详细描述了这些约定：[命名 WebSphere MQ 对象的规则](#)。

Web 网关名称

- Web 网关名称最长可以有 28 个字符，并且不区分大小写。以小写形式或大小写混合形式输入的 Web 网关名称将转换为大写形式。Web 网关名称必须符合标准的 WebSphere MQ 对象命名约定。以下详细描述了这些约定：[命名 WebSphere MQ 对象的规则](#)。除 WebSphere MQ 对象命名约定外，Web 网关名称中不能使用正斜杠 (/) 字符和百分号 (%) 字符。
- 如果部署同一 Web 网关的多个实例，请为每个实例使用同一名称。
- 如果部署一个以上的独立 Web 网关，请为每个网关使用不同的名称。不要创建一个以上的名称相同的 Web 网关。
- 为作为 Web 网关的组件的 Web 代理赋予与 Web 网关名称相似的名称。例如，如果 Web 网关名为 `WG1_GTWY`，请将 Web 代理命名为 `WG1_AGNT_QM1`。

IBM i 集成文件系统 (IFS) 中的文件

IFS 中的文件名不能包含以下任何字符：

- 反斜杠 (\)
- 正斜杠 (/)
- 冒号 (:)
- 星号 (*)
- 问号 (?)
- 引号 (")
- 小于号 (<)
- 大于号 (>)
- 竖线 (|)

如果尝试将名称中包含这些字符的文件传输到 IBM i IFS，那么传输这些文件将失败。

数据集名称

数据集具有命名限制，这影响名称最大长度以及可用于数据集名称的可用字符。PDS 数据集成员名最多可有 8 个字符且不能包含点 (.) 字符。当传输到数据集时，必须明确指定名称，这意味着这些命名限制不会产生问题。但当从文件传输到 PDS 成员时，文件路径可能不会映射到 PDS 成员名。当传输到 PDS 数据集时，每个源文件都将成为 PDS 成员，并且会通过源名称生成每个成员名。

PDS 成员名是 z/OS 非限定名，由以下正则表达式定义：

```
[a-zA-Z$#@][a-zA-Z0-9$#@]{0-7}
```

以下方案用于将源数据集或源文件名转换为有效的 PDS 成员名。将以所列顺序应用注意事项：

1. 仅使用名称中最后一个正斜杠 (/)、最后一个反斜杠 (\) 或最后一个冒号 (:) 字符后的字符。即，仅使用文件路径的名称部分。
2. 对于源文件（非数据集或 PDS 成员），将忽略最后一个点号 (.)（含）后面的字符。
3. 对于任何长度超过八个字符的名称，仅使用最后八个字符。
4. 使用 @ 符号字符替换点号字符。
5. 使用 @ 符号字符替换无效字符。
6. 如果转换未生成任何字符，那么 PDS 成员名为 @。

管理

代理状态值

fteListAgents 和 **fteShowAgentDetails** 命令生成代理状态信息。该状态有多种可能的值。

活动的

代理正在运行并且正在发送或接收文件。代理正在定期发布其状态。最近一次更新是在预期时间段内收到的。

准备就绪

代理正在运行，但是未发送或接收文件。代理正在定期发布其状态。最近一次更新是在预期时间段内收到的。

正在启动

代理正在启动，但还未准备好执行传输。

UNREACHABLE

在预期的时间间隔内未接收到代理状态更新。代理可能由于错误已停止运行，或者已突然关闭，或者正在运行但遇到通信问题。

已停止

代理已停止。它以受控方式关闭。

ENDED UNEXPECTEDLY

代理已意外终止。将自动重新启动代理程序，除非在 `maxRestartInterval` 时间段内重新启动次数超过 `maxRestartCount` 次，并且 `maxRestartDelay` 值小于或等于 0。有关这些属性的更多信息，请参阅第 591 页的『[agent.properties 文件](#)』。

NO_INFORMATION

代理版本可能为 WebSphere MQ File Transfer Edition V7.0.2 或更早版本。代理未以该命令可以处理的格式发布更新。

未知

无法确定代理的状态。它可能发布了该工具无法识别的状态。如果在您的网络上混合使用了产品版本，那么升级该工具的安装版本可修复此问题。

PROBLEM

代理命令处理程序可能不能工作。代理正在发布状态消息，但这些状态消息已过时。

相关参考

第 532 页的『[fteListAgents](#)（列出协调队列管理器的 IBM MQ Managed File Transfer 代理）』

通过命令行，使用 **`fteListAgents`** 命令可列出向特定协调队列管理器注册的所有 IBM MQ Managed File Transfer 代理。

第 564 页的『[fteShowAgentDetails](#)（显示 IBM MQ Managed File Transfer 代理详细信息）』

使用 **`fteShowAgentDetails`** 命令可显示特定 IBM MQ Managed File Transfer 代理的详细信息。这些是由其 IBM MQ Managed File Transfer 协调队列管理器存储的详细信息。

第 387 页的『[在您认为传输卡住时应该怎么办](#)』

在高负荷的系统上或者在源和目标代理之间发生网络问题时，传输可能会偶尔显示为卡住，进入排队或恢复状态。导致此情况的因素有很多。

代理进程控制器状态值

`fteShowAgentDetails` 命令会生成代理进程控制器状态信息。该状态有多种可能的值。

WAITING

代理进程控制器在启动代理之前，正在等待队列管理器变为可用。

已启动

代理进程控制器已启动代理进程。

已停止

代理进程控制器已停止，因为有停止代理的请求或者在重新启动时间间隔内已有过多次的代理进程重新启动。

RECOVERING

代理进程已意外停止，而进程控制器将尝试将该进程重新启动。

ISTOPPING

代理进程已接收到立即关闭的请求。当代理进程已停止时，进程控制器将停止。

CSTOPPING

代理进程已接收到以受控方式关闭的请求。当代理进程已停止时，进程控制器将停止。

未知

无法确定代理进程控制器状态。这可能是由于代理进程控制器未在运行，或者是因为该控制器正在与 `fteShowAgentDetails` 命令的运行位置不同的系统上运行。

相关参考

第 564 页的『[fteShowAgentDetails](#)（显示 IBM MQ Managed File Transfer 代理详细信息）』

使用 **`fteShowAgentDetails`** 命令可显示特定 IBM MQ Managed File Transfer 代理的详细信息。这些是由其 IBM MQ Managed File Transfer 协调队列管理器存储的详细信息。

记录器状态值

`fteShowLoggerDetails` 命令生成记录器状态信息。该状态有多种可能的值。

活动的

记录器正在运行并且正在发送或接收文件。记录器正在定期发布其状态。最近一次更新是在预期时间段内收到的。

准备就绪

记录器正在运行，但是未在发送或接收文件。记录器正在定期发布其状态。最近一次更新是在预期时间段内收到的。

正在启动

记录器正在启动，但还未准备好执行传输。

UNREACHABLE

在预期的时间间隔内未接收到记录器状态更新。记录器可能已由于发生错误而停止运行，或者已突然关闭，或者正在运行但遇到通信问题。

已停止

已停止记录器。它以受控方式关闭。

ENDED UNEXPECTEDLY

记录器已意外结束。将自动重新启动记录器，除非在 `maxRestartInterval` 时间段内重新启动次数超过 `maxRestartCount` 次，并且 `maxRestartDelay` 值小于或等于 0。有关这些属性的更多信息，请参阅第 163 页的『IBM MQ Managed File Transfer 的记录器配置属性』。

对于 **fteShowLoggerDetails** 命令，此状态的详细信息将包含状态码，该状态码是记录器进程的出口码。请参阅“进程出口码”以获取已知出口码的列表。

NO_INFORMATION

记录器版本可能为 WebSphere MQ File Transfer Edition V7.0.2 或更低版本。记录器未以此命令可以处理的格式发布更新。

未知

无法确定记录器的状态。它可能发布了该工具无法识别的状态。如果在您的网络上混合使用了产品版本，那么升级该工具的安装版本可修复此问题。

PROBLEM

记录器命令处理程序可能无法运作。记录器正在发布状态消息，但这些状态消息已过时。

相关参考

第 571 页的『fteShowLoggerDetails（显示 IBM MQ Managed File Transfer 记录器详细信息）』
使用 **fteShowLoggerDetails** 命令显示特定 IBM MQ Managed File Transfer 记录器的详细信息。

记录器进程控制器状态值

fteShowLoggerDetails 命令生成记录器进程控制器状态信息。该状态有多种可能的值。

WAITING

记录器进程控制器正在等待队列管理器变为可用，以便随后启动记录器。

已启动

记录器进程控制器已启动记录器进程。

已停止

记录器进程控制器已停止，因为发出了停止该记录器的请求，或者在重新启动时间间隔内记录器进程重新启动的次数过多。

RECOVERING

记录器进程已意外停止，而进程控制器将尝试重新启动该进程。

ISTOPPING

记录器进程已接收到立即关闭的请求。在记录器进程已停止时，进程控制器将停止。

CSTOPPING

记录器进程已接收到以受控方式关闭的请求。在记录器进程已停止时，进程控制器将停止。

未知

无法确定记录器进程控制器状态。这可能是由于记录器进程控制器未在运行，或者是因为该控制器正在与 **fteShowLoggerDetails** 命令的运行位置不同的系统上运行。

相关参考

第 571 页的『[fteShowLoggerDetails](#)（显示 IBM MQ Managed File Transfer 记录器详细信息）』
使用 **fteShowLoggerDetails** 命令显示特定 IBM MQ Managed File Transfer 记录器的详细信息。

文件传输准则

根据传输源和目标的操作系统，以及使用二进制还是文本方式传输，提供有关期望行为的指南。

请阅读相关主题以获取更多信息。

相关参考

第 703 页的『[在 z/OS 与分布式系统之间传输文件和数据集](#)』

您可以使用 IBM MQ Managed File Transfer 在 z/OS 与分布式文件系统之间传输文件和受支持的数据集类型。仔细查看以下行为，这取决于要与其进行传输的系统的类型。

第 705 页的『[在数据集之间传输](#)』

您可以使用 IBM MQ Managed File Transfer 在 z/OS 数据集之间传输。仔细复审以下行为以确保正确传输数据集。

第 706 页的『[与 Connect:Direct 节点之间传输数据集](#)』

可以使用 Connect:Direct 网桥在 IBM MQ Managed File Transfer 代理和 IBM Sterling Connect:Direct 节点之间传输数据集。您可以将数据集指定为传输源和/或传输目标。

第 708 页的『[Connect:Direct 进程语句参数和 BPXWDYN 关键字之间的映射](#)』

对于源或目标是 Connect:Direct 节点的数据集，当提交传输请求时，您提供的任何受支持的 BPXWDYN 关键字都将转换为 Connect:Direct 进程接受的格式。

第 712 页的『[不能与 IBM MQ Managed File Transfer 一起使用的 BPXWDYN 属性](#)』

使用 **fteCreateTemplate** 命令、**fteCreateTransfer** 命令或 `agent.properties` 文件中的 **bpxwdynAllocAdditionalOptions** 属性时，不能指定某些 BPXWDYN 选项。

第 713 页的『[传输文本文件](#)』

文本文件传输包括将文件代码页从一个代码页转换为另一个。文本文件传输还包括在系统之间转换 CRLF（回车-换行符）字符。本主题概述 IBM MQ Managed File Transfer 的文本文件传输行为。

第 715 页的『[在 Connect:Direct 和 IBM MQ Managed File Transfer 之间传输文本文件](#)』

文本传输涉及从一个代码页到另一个代码页的文件代码页转换。文本文件传输还涉及系统之间的 CRLF（回车换行）字符转换。本主题总结了 MQMFT 代理和 Connect:Direct 节点之间的传输中的文本文件传输行为。

第 716 页的『[与协议网桥代理传输文件](#)』

您可以使用协议网桥代理将文件传输至您的 IBM MQ Managed File Transfer 网络外部的 FTP 或 SFTP 文件服务器，或从这些服务器传输文件。

第 716 页的『[在 IBM i 系统上传输文件](#)』

如果您使用文本方式的 IBM MQ Managed File Transfer 在 IBM i 系统上传输文件，并且希望转换文件中的数据，请研究本主题中的信息。

第 719 页的『[传输位于 IBM i 系统上 QSYS.LIB 文件系统上的保存文件](#)』

IBM MQ Managed File Transfer 支持在两个 IBM i 系统之间传输 QSYS.LIB 文件系统上的保存文件。在请求对保存文件进行文件传输时，请考虑以下信息。

第 721 页的『[传输世代数据组 \(GDG\)](#)』

IBM MQ Managed File Transfer 在 z/OS 上支持针对源和目标数据集的生成数据组 (GDG)。支持绝对和相对 GDG 名称。写入到新的生成时，基本 GDG 必须存在。

第 721 页的『[使用通配符](#)』

为文件传输指定源文件名和源文件路径时，可以使用通配符。这使您能够同时选择多个文件。

在 z/OS 与分布式系统之间传输文件和数据集

您可以使用 IBM MQ Managed File Transfer 在 z/OS 与分布式文件系统之间传输文件和受支持的数据集类型。仔细查看以下行为，这取决于要与其进行传输的系统的类型。

IBM MQ Managed File Transfer 在 z/OS 上支持针对源和目标数据集的生成数据组 (GDG)。支持绝对和相对 GDG 名称。写入到新的生成时，基本 GDG 必须存在。

将文件或数据集传输到磁带时，磁带上的任何现有数据集都将被替换。新数据集的属性将根据传输定义中传递的属性进行设置。如果未指定任何属性，那么新数据集的属性将设置为与源数据集的那些属性相同，或者，如果源数据集是文件，那么这些属性将设置为缺省值。将忽略现有磁带数据集的属性。

从文件传输到数据集 - 二进制传输

目标数据集的格式决定目标记录长度。确保数据集存在于目标系统上，或指定具有正确属性的目标数据集，以便正确创建数据集。如果未指定属性，那么系统将指定以下缺省值：未定义其记录格式的物理连续数据集，以及设备（根据 DEVTYPE 宏返回的内容）的最大块大小 (BLKSIZE)。例如，对于 DASD，大小为 6144，而对于磁带，大小为 32760。如果要以二进制方式将分布式系统上的文件传输到 z/OS 数据集，那么请注意以下行为：

物理连续 (PS) 目标数据集：

- 按次序读取分布式系统上的源文件以填充每条记录或每个块。
- 对于可变格式数据集，会将每条记录填充到容量。

分区数据集 (PDS) 目标数据集：

- 会将每个源文件复制到具有相同或等效名称的 PDS 成员。如果文件名长于成员名的最大允许长度，那么会将文件名转换为有效的成员名。有关成员名的更多信息，请参阅[对象命名约定](#)。如果源文件是一个目录，那么该目录中的每个文件都将成为 PDS 的成员。
- 如果存在 PDS 成员并已指定覆盖传输的现有目标文件，那么将覆盖成员。如果未指定覆盖，那么传输将失败。
- 按次序读取分布式系统上的源文件以填充成员的每条记录或每个块。
- 对于可变格式 PDS 成员，会将每条记录填充到容量。

从文件传输到数据集 - 文本传输

目标数据集的格式决定目标记录长度。确保数据集存在于目标系统上，或指定具有正确属性的目标数据集，以便正确创建数据集。如果要将分布式系统上的文件作为文本传输到 z/OS 数据集，那么请注意以下行为：

物理连续 (PS) 目标数据集：

- 每行文本将成为一条记录（或块，针对未定义记录格式 (RECFM=U) 数据集）。数据集记录中不显示行结束字符（仅针对非 ASA 数据集）。
- 在目标数据集中使用 ASA 格式控制字符时，会将行结束字符有效地转换为等效的 ASA 格式控制代码。
- 当行长于记录时，会在记录界限分割行并流向下一条记录。

PDS 目标数据集：

- 会将每个源文件复制到具有相同或等效名称的 PDS 成员。如果文件名长于成员名的最大允许长度，那么会将文件名转换为有效的成员名。有关成员名的更多信息，请参阅[对象命名约定](#)。如果源文件是一个目录，那么该目录中的每个文件都将成为 PDS 的成员。
- 如果存在 PDS 成员并已指定覆盖传输的现有目标文件，那么将覆盖成员。如果未指定覆盖，那么传输将失败。
- 每行文本将成为一条记录（或块，针对未定义记录格式 (RECFM=U) 数据集）。成员记录中不显示行结束字符（仅针对非 ASA 数据集）。
- 在目标数据集中使用 ASA 格式控制字符时，会将行结束字符有效地转换为等效的 ASA 格式控制代码。
- 当行长于记录时，会在记录界限分割行并流向下一条记录。

从数据集传输到文件 - 二进制和文本传输

如果要以二进制或文本形式从数据集传输到文件，那么请注意以下行为：

- 会将每条记录的内容以二进制形式传输到文件；不会传输记录、块格式信息或 ASA 格式控制字符。
- 仅针对文本传输，每条数据集记录将成为将文本转换为目标代理代码页的行。即，针对 Windows 目标系统，将追加回车-换行符 (CRLF)；针对 UNIX 目标系统，将追加回车符 (CR)。

- **非 VSAM 和 PS 源数据集。** 会将源数据集的记录传输到目标文件并且会将这些记录合并在一起。如果已存在目标文件，那么将覆盖该文件，这取决于您为文件传输指定的目标文件行为选项。
- **PDS 源数据集。** 会将每个指定的成员或（如果未指定任何成员）所有成员抽取到目标。如果目标指定了一个目录，那么会将成员抽取到单独的文件。否则，会将每个指定的成员写入目标文件，因此实际上仅传输一个成员。如果针对某个成员已存在目标文件，那么将覆盖该文件，这取决于您为文件传输指定的目标文件行为选项。

相关参考

第 703 页的『文件传输准则』

根据传输源和目标的操作系统，以及使用二进制还是文本方式传输，提供有关期望行为的指南。

在数据集之间传输

您可以使用 IBM MQ Managed File Transfer 在 z/OS 数据集之间传输。仔细复审以下行为以确保正确传输数据集。

IBM MQ Managed File Transfer 不支持磁盘或磁带上的未编目数据集。现有数据集必须进行编目，并且新数据集也将进行编目。

请考虑以下情况：

如果在 z/OS 系统之间复制或移动数据集，并且目标不存在。

缺省情况下，将使用与源相同的特征创建目标数据集。您可以指定目标数据集的属性以覆盖缺省特征。如果执行此操作，将执行兼容性检查以确保可以进行传输。

如果在 z/OS 系统之间复制或移动数据集，并且目标已存在。

- 如果您为目标数据集指定属性以覆盖缺省特征，将执行兼容性检查以确保能够以所需方式访问目标数据集。但是，您不能覆盖以下属性：
 - 基本数据集组织和类型
 - 逻辑记录长度 (LRECL)
 - 块大小 (BLKSIZE)

如果要将数据集传输到磁带。

将数据集传输到磁带时，将替换磁带上的任何现有数据集。新数据集的属性将根据传输定义中传递的属性进行设置。如果未指定任何属性，那么属性将设置为与源数据集的属性相同，或者，如果源是文件，那么设置为缺省值。将忽略现有磁带数据集的属性。

数据集兼容性

复审以下行为和限制以检查数据集兼容性：

记录格式和长度差异：

- 可变格式记录在记录数据中使用 4 字节记录长度字段。因此，对于从固定记录到可变记录数据集的传输，可变记录长度必须大于或等于固定记录长度加 4。对于从变量格式记录数据集到固定格式记录数据集的传输，固定格式记录数据集记录长度必须大于或等于变量记录长度减去 4。

块大小差异：

- 对于固定以及可变格式记录数据，块大小差异会造成源和目标数据集布局不同。
- 对于未定义格式的记录，在目标块大小大于或等于源数据集块大小的前提下，您可以传输数据集。
- 对于未定义格式的数据集，如果源块大小大于目标块大小，那么您不能进行传输。

分区数据集 (PDS) 和分区数据集扩展 (PDSE) 数据集

以下行为和限制对 PDS 和 PDSE 同等适用：

- 如果您将 PDS 或 PDSE 成员传输到目标 PDS 或 PDSE，那么将创建目标 PDS 或 PDSE 的成员。如果目标 PDS 或 PDSE 成员已经存在，将覆盖该成员。如果您将 PDS 或 PDSE 成员传输到非 PDS 或非 PDSE 目标数据集，那么将创建目标数据集以包含成员数据。如果目标数据集已经存在，将覆盖该数据集。

- 如果尝试将 PDS 或 PDSE 传输到非 PDS 或非 PDSE 目标，那么这会将 PDS 或 PDSE 的所有成员写入非 PDSE 目标。每个后续成员传输将覆盖非 PDSE 目标的先前内容或者会失败，具体取决于传输选项。
- 将 PDS 或 PDSE 传输到目标 PDS 或 PDSE 时，将在目标处创建整个 PDS 或 PDSE 的副本。如果目标 PDS 或 PDSE 已经存在，将添加源中的成员。如果 PDS 或 PDSE 成员已经存在于目标处，将覆盖该成员。
- 将非 PDS 或非 PDSE 传输到目标 PDS 或 PDSE，会将非 PDS 或非 PDSE 的内容添加为该 PDS 或 PDSE 的新成员。如果 PDS 成员已经存在，将覆盖该成员。如果未指定新成员的名称，将从源数据集或 DD 名称生成一个名称。
- 在磁盘空间受限的系统上传输到 PDS 和 PDSE 数据集时，存在一个已知限制。有关更多详细信息，请参阅主题对 [WebSphere MQ File Transfer Edition](#) 进行故障诊断。
- **注：**将 PDS 或 PDSE 传输到目标 PDS 或 PDSE 时，不会保留成员信息和统计信息。例如，如果您传输一个存储为 PDS 的装入库，那么目标 PDS 不可用作装入库。

二进制和文本传输

数据集的二进制传输将定义为其二进制格式的记录数据，即使用缺省记录格式（类型为记录）从数据集读取的数据。将以逐个记录的方式读取和写入数据。系统服务执行必要的记录和块转换（其中数据集具有不同的记录和块设置）以及必要的 ASA 和机器控制代码转换。如果为 ASA 格式控制字符定义了一个数据集而另一数据集不适合，那么将使用 C/C++ 系统库函数行为来转换到正常控制代码。

生成数据组 (GDG)

IBM MQ Managed File Transfer 在 z/OS 上支持针对源和目标数据集的生成数据组 (GDG)。支持绝对和相对 GDG 名称。写入新的生成时，基本 GDG 必须已经存在。

相关参考

[第 703 页的『文件传输准则』](#)

根据传输源和目标的操作系统，以及使用二进制还是文本方式传输，提供有关期望行为的指南。

[第 721 页的『传输世代数据组 \(GDG\)』](#)

IBM MQ Managed File Transfer 在 z/OS 上支持针对源和目标数据集的生成数据组 (GDG)。支持绝对和相对 GDG 名称。写入到新的生成时，基本 GDG 必须存在。

[第 706 页的『与 Connect:Direct 节点之间传输数据集』](#)

可以使用 Connect:Direct 网桥在 IBM MQ Managed File Transfer 代理和 IBM Sterling Connect:Direct 节点之间传输数据集。您可以将数据集指定为传输源和/或传输目标。

与 Connect:Direct 节点之间传输数据集

可以使用 Connect:Direct 网桥在 IBM MQ Managed File Transfer 代理和 IBM Sterling Connect:Direct 节点之间传输数据集。您可以将数据集指定为传输源和/或传输目标。

指定数据集名称

要在传输请求中指定 Connect:Direct 节点上的数据集，请使用 IBM MQ Managed File Transfer 代理间数据集传输所使用的语法，但需执行以下两个更改：

- 必须以 Connect:Direct 节点名和冒号 (:) 作为数据集名称的前缀。语法如下：

```
cdNode:data_set_name{;attrib1;...;attribN}
```

例如，要在 Connect:Direct 节点 CD_NODE1 所在的系统上指定名为 OBJECT.LIB 的分区数据集，请使用以下语法：

```
CD_NODE1:/'OBJECT.LIB';RECFM(F,B);BLKSIZE(800);LRECL(80)
```

在该示例中，三个可选属性由文本 RECFM(F,B);BLKSIZE(800);LRECL(80) 指定。

- 指定的数据集名称将解释为标准数据集名称，而与是否由单引号括起来无关。系统不会添加任何前缀。如果要指定前缀，如用于运行代理的用户标识，那么必须将其指定为数据集名称的一部分。这不同于仅涉及

IBM MQ Managed File Transfer 代理的数据集传输行为，对于这种数据集传输，如果指定的数据集名称未使用单引号字符括起来，那么系统将为目标代理添加缺省高级限定符的前缀。

除上述两处更改外，可使用在 IBM MQ Managed File Transfer 代理之间的数据集传输使用的语法来指定数据集名称和所有可选属性，其规则如下：

- 必须在数据集名称前添加两个正斜杠 (//) 字符作为前缀。
- 如果要指定数据集属性，请在数据集名称之后提供，并使用分号分隔。必须按照 *key(value)* 格式提供属性，该格式适合于 BPXWDYN。

有关在传输请求中指定数据集的更多信息，请参阅第 497 页的『[fteCreateTransfer（创建新的文件传输）](#)』和第 484 页的『[fteCreateTemplate（新建文件传输模板）](#)』。

用于传输请求中的参数

对于涉及 Connect:Direct 节点上的数据集的大部分传输请求，您都可以按照仅涉及 IBM MQ Managed File Transfer 代理的数据集传输方式来指定源和目标数据集。将 **source_specification**、**-ds** 和 **-dp** 参数用于 **fteCreateTransfer** 或 **fteCreateTemplate** 命令。该语法在以下情况下受支持：

- 传输中涉及的所有代理均为 V7.0.4 或更高版本
- 源代理为 Connect:Direct 网桥代理，因此为 V7.0.4 或更高版本，而目标代理为 V7.0.3 或更早版本

如果目标代理为 Connect:Direct 网桥代理，而源代理为 V7.0.3 或更早版本，那么必须对传输请求执行以下更改：

- 要指定连续数据集或分区数据集 (PDS) 成员作为传输的目标，请使用 **-df** 参数。
- 要指定 PDS 作为传输的目标，请使用 **-dd** 参数。

您也可用使用该语法来替代源代理为 V7.0.4 或更高版本的传输的惯用 **-ds** 和 **-dp** 参数。例如，如果您希望所有情况中使用一致语法而某些情况涉及 V7.0.3 或更早版本的源代理，请使用 **-df** 和 **-dd** 参数。

注：如果传输的目标是 PDS，而目标代理是 Connect:Direct 网桥代理，那么必须使用值 **overwrite** 指定 **-de** 参数。

指定数据集属性

某些数据集属性由 IBM MQ Managed File Transfer 设置，并作为参数传递到 Connect:Direct **COPY** 进程。您也可以通过指定适当的 BPXWDYN 关键字，在传输请求中提供某些属性。Connect:Direct 网桥将具有等效 Connect:Direct 属性的关键字转换为 Connect:Direct 要求的格式。例如，在数据集规范 `CD_NODE1: // 'OBJECT.LIB'; RECFM(F,B); BLKSIZE(800); LRECL(80)` 中，属性 `RECFM(F,B); BLKSIZE(800); LRECL(80)` 将转换为 `DCB=(RECFM=FB, BLKSIZE=800, LRECL=80)`。

关于这两种类型的参数之间的映射详细信息（包括支持与 Connect:Direct 传输一起使用的 BPXWDYN 关键字的详细信息），请参阅第 708 页的『[Connect:Direct 进程语句参数和 BPXWDYN 关键字之间的映射](#)』。并非所有 BPXWDYN 关键字都具有等效的 Connect:Direct 进程参数，同时并非所有 Connect:Direct 进程参数都具有等效 BPXWDYN 关键字。

其他考虑事项

- 如果传输目标是位于 Connect:Direct 节点上的分区数据集，那么必须在传输之前创建该分区数据集，因为 Connect:Direct 节点不会为您创建。

相关概念

第 288 页的『[Connect:Direct 网桥](#)』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关任务

将数据集传输到 z/OS 上的 Connect:Direct 节点

您可以使用位于 Windows 或 Linux 系统上的 Connect:Direct 网桥将数据集从 z/OS 上的 IBM MQ Managed File Transfer 代理传输到 z/OS 上的 Connect:Direct 节点。

相关参考

第 705 页的『在数据集之间传输』

您可以使用 IBM MQ Managed File Transfer 在 z/OS 数据集之间传输。仔细复审以下行为以确保正确传输数据集。

第 497 页的『fteCreateTransfer（创建新的文件传输）』

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

第 484 页的『fteCreateTemplate（新建文件传输模板）』

fteCreateTemplate 命令创建文件传输模板，您可以保留该模板以供将来使用。唯一的必需参数是 **-tn** (*template_name*) 参数。其他所有参数都是可选的，但如果指定源文件，还必须提供目标文件。类似地，如果指定目标文件，还必须指定源文件规范。

第 426 页的『使用双正斜杠指定 Connect:Direct 文件路径』

在文件传输过程中，如果使用以双正斜杠 (//) 开始的文件路径指定位于 Connect:Direct 节点上的文件，那么会将该文件视为数据集。

Connect:Direct 进程语句参数和 BPXWDYN 关键字之间的映射

对于源或目标是 Connect:Direct 节点的数据集，当提交传输请求时，您提供的任何受支持的 BPXWDYN 关键字都将转换为 Connect:Direct 进程接受的格式。

有关 IBM Sterling Connect:Direct 流程语句的更多信息，请参阅 [Connect:Direct Process Language Reference Guide](#)。

表 59: Connect:Direct COPY 语句的参数以及 IBM MQ Managed File Transfer 使用的等效 BPXWDYN 键	
Connect:Direct COPY 语句的参数	BPXWDYN 关键字
DSN	DSN（针对到数据集或来自数据集的传输有效）。指定该关键字将覆盖 IBM MQ Managed File Transfer 指定的参数值，该值基于传输请求中提供的源或目标文件规范。
文件	针对数据集不存在任何映射。
PNODE	不存在任何映射。传输的主节点由 IBM MQ Managed File Transfer 标识。如果尝试为该参数提供值，那么将生成一个错误。
SNODE	不存在任何映射。传输的辅助节点由 IBM MQ Managed File Transfer 标识。如果尝试为该参数提供值，那么将生成一个错误。
DCB	请参阅 映射 DCB 的子参数
DISP	请参阅 映射 COPY From 语句的 DISP 的子参数 和 映射 COPY To 语句的 DISP 的子参数
RESGDG	不存在任何映射
LABEL	请参阅 映射 LABEL 的子参数
MSVGP	不存在任何映射
UNIT	UNIT
VOL	请参阅 映射 VOL 的子参数
ALIAS	不存在任何映射
EXCLUDE	不存在任何映射
PDS.DIR	不存在任何映射。IBM MQ Managed File Transfer 将此进程参数的值设置为 N，因此不会发送目录中与用户相关的信息。

表 59: Connect:Direct COPY 语句的参数以及 IBM MQ Managed File Transfer 使用的等效 BPXWDYN 键 (继续)	
Connect:Direct COPY 语句的参数	BPXWDYN 关键字
REPLACE NOREPLACE	无等效 BPXWDYN。目标系统上已存在目标数据集时的行为由传输请求中 -de (destination_file_behavior) 参数的值定义。有关目标数据集已存在时的 IBM MQ Managed File Transfer 缺省行为的更多信息，请参阅第 705 页的『在数据集之间传输』。
SELECT	无等效 BPXWDYN。选择用于复制的数据集成员由传输请求中的源文件规范定义。
BUFND	不存在任何映射
IOEXIT	不存在任何映射
DATAEXIT	不存在任何映射
SYSOPTS	请参阅映射 SYSOPTS 的子参数
类型	不存在任何映射
AVGREC	不存在任何映射
DATACLAS	DATACLAS
DSNTYPE	DSNTYPE。为该关键字指定 PDS 值将覆盖由 IBM MQ Managed File Transfer 分配的参数值（该值为 LIBRARY）。不存在针对任何其他值的映射 - EXTPREF、EXTREQ、BASIC 或 LARGE。指定任何这些不受支持的值将会生成错误。针对连续数据集指定 PDS 或 LIBRARY 会生成错误。
KEYLEN	不存在任何映射
KEYOFF	不存在任何映射
LIKE	LIKE
LRECL	不存在任何映射
MGMTCLAS	MGMTCLAS
RECORG	不存在任何映射
SECMODEL	不存在任何映射
STORCLAS	STORCLAS
SPACE	请参阅映射 SPACE 的子参数
SYSOUT	不存在任何映射
CKPT	不存在任何映射
压缩	不存在任何映射
SECURE	不存在任何映射

表 60: Connect:Direct COPY 语句的 DCB 参数的子参数以及 IBM MQ Managed File Transfer 使用的等效 BPXWDYN 键	
DCB 参数的子参数	BPXWDYN 关键字
model-file-name	不存在任何映射

表 60: Connect:Direct **COPY** 语句的 **DCB** 参数的子参数以及 IBM MQ Managed File Transfer 使用的等效 BPXWDYN 键 (继续)

DCB 参数的子参数	BPXWDYN 关键字
BLKSIZE	BLKSIZE
NCP	BUFNO
DEN	不存在任何映射
DSORG	DSORG
KEYLEN	不存在任何映射
LIMCT	不存在任何映射
LRECL	LRECL
OPTCD	不存在任何映射
RECFM	RECFM
RKP	不存在任何映射
TRTCH	TRTCH

表 61: Connect:Direct **COPY From** 语句的 **DISP** 参数的子参数以及 IBM MQ Managed File Transfer 使用的等效 BPXWDYN 键

COPY From 语句的 DISP 参数的子参数	BPXWDYN 关键字	详细信息
[OLD SHR]	[OLD SHR]	指定传输前数据集的状态。IBM MQ Managed File Transfer 将此子参数设置为 SHR 。
[KEEP DELETE]	[KEEP DELETE] 或 PATHDISP	指定传输成功完成后数据集的状态。IBM MQ Managed File Transfer 设置的值取决于 -sd 参数定义的源文件处置。
[KEEP DELETE]	[KEEP DELETE] 或 PATHDISP	指定在传输异常完成后数据集的状态。IBM MQ Managed File Transfer 将此子参数设置为 KEEP 。

表 62: Connect:Direct **COPY To** 语句的 **DISP** 参数的子参数以及 IBM MQ Managed File Transfer 使用的等效 BPXWDYN 键

COPY To 语句的 DISP 参数的子参数	BPXWDYN 关键字	详细信息
[NEW OLD MOD RPL SHR]	[NEW OLD MOD SHR]	指定传输前数据集的状态。IBM MQ Managed File Transfer 设置的值取决于传输请求中 -de (destination_file_behavior) 参数的值。如果目标数据集尚不存在，那么该子参数的值将为 NEW 。如果该数据集已存在，那么该子参数的值将为 RPL 。IBM MQ Managed File Transfer 不支持传输请求中提供的密钥 RPL 。

表 62: Connect:Direct **COPY To** 语句的 **DISP** 参数的子参数以及 IBM MQ Managed File Transfer 使用的等效 BPXWDYN 键 (继续)

COPY To 语句的 DISP 参数的子参数	BPXWDYN 关键字	详细信息
[KEEP CATLG]	[KEEP CATLOG] 或 PATHDISP	指定传输成功完成后数据集的状态。IBM MQ Managed File Transfer 将该子参数设置为 CATLOG 。
[KEEP CATLG DELETE]	[KEEP DELETE] 或 PATHDISP	指定在传输异常完成后数据集的状态。IBM MQ Managed File Transfer 将该子参数设置为 DELETE 。

表 63: Connect:Direct **COPY** 语句的 **LABEL** 参数的子参数以及 IBM MQ Managed File Transfer 使用的等效 BPXWDYN 键

COPY 语句的 LABEL 参数的子参数	BPXWDYN 关键字	详细信息
file-sequence-number	序列	
[SL AL BLP LTM NL]	LABEL(类型)	类型的可能值包含 NL、SL、NSL、SUL、BLP、LTM、AL 和 AUL。Connect:Direct 接受这些值的子集。如果指定 Connect:Direct 不支持的值，那么 Connect:Direct 会生成错误消息。
[PASSWORD NOPWREAD]	不存在任何映射	
[IN OUT]	不存在任何映射	
[RETPD EXPDT]	RETPD	EXPDT 不受支持。

表 64: Connect:Direct **COPY** 语句的 **VOL** 参数的子参数以及 IBM MQ Managed File Transfer 使用的等效 BPXWDYN 键

COPY 语句的 VOL 参数的子参数	BPXWDYN 关键字
专用	不存在任何映射
RETAIN	不存在任何映射
volume-sequence-no	不存在任何映射
volume-count	MAXVOL
SER	VOL
REF	不存在任何映射

表 65: Connect:Direct **COPY** 语句的 **SYSOPTS** 参数的子参数以及 IBM MQ Managed File Transfer 使用的等效 BPXWDYN 键

COPY 语句的 SYSOPTS 参数的子参数	BPXWDYN 关键字
DBCS	不存在任何映射
代码页	值取决于 IBM MQ Managed File Transfer 传输选项。有关更多信息，请参阅第 713 页的『传输文本文件』。
数据类型	不存在任何映射。IBM MQ Managed File Transfer 将此值设置为 TEXT，用于对数据集进行文本传输或从数据集进行文本传输，否则设置为 BINARY。

表 65: *Connect:Direct COPY* 语句的 **SYSOPTS** 参数的子参数以及 *IBM MQ Managed File Transfer* 使用的等效 *BPXWDYN* 键 (继续)

COPY 语句的 SYSOPTS 参数的子参数	BPXWDYN 关键字
XLATE	当 DATATYPE 的值为 TEXT 时，不存在任何映射. IBM MQ Managed File Transfer 将此值设置为 NO。
STRIP.BLANKS	当 DATATYPE 的值为 TEXT 时，不存在任何映射. IBM MQ Managed File Transfer 将此值设置为 YES。
PERMISS	不存在任何映射
PRECOMP	不存在任何映射
UNIQUE	不存在任何映射
SYSOUT	不存在任何映射

表 66: *Connect:Direct COPY* 语句的 **SPACE** 参数的子参数以及 *IBM MQ Managed File Transfer* 使用的等效 *BPXWDYN* 键

COPY 语句的 SPACE 参数的子参数	BPXWDYN 关键字
CYL	CYL
TRK	TRACKS
blk	BLOCKS
av-rec-len	不存在任何映射
prim, [sec], [dir]	SPACE(prim[,sec]), DIR
RLSE	释放
CONTIG	不存在任何映射
ROUND	不存在任何映射

相关概念

第 288 页的『[Connect:Direct 网桥](#)』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关任务

将数据集传输到 z/OS 上的 [Connect:Direct 节点](#)

您可以使用位于 Windows 或 Linux 系统上的 Connect:Direct 网桥将数据集从 z/OS 上的 IBM MQ Managed File Transfer 代理传输到 z/OS 上的 Connect:Direct 节点。

相关参考

与 Connect:Direct 节点传输数据集

可以使用 Connect:Direct 网桥在 IBM MQ Managed File Transfer 代理和 IBM Sterling Connect:Direct 节点之间传输数据集。您可以将数据集指定为传输源和/或传输目标。

不能与 *IBM MQ Managed File Transfer* 一起使用的 *BPXWDYN* 属性

使用 **fteCreateTemplate** 命令、**fteCreateTransfer** 命令或 **agent.properties** 文件中的 **bpxwdynAllocAdditionalOptions** 属性时，不能指定某些 BPXWDYN 选项。

有些 BPXWDYN 选项不能与 IBM MQ Managed File Transfer 一起指定，因为这些选项由代理使用或者它们不受支持。如果使用这些选项，它们可能会导致不可预测的行为；下表中列举了这些选项。

BPXWDYN 选项	描述
DA DSN	指定要分配的数据集名称。

BPXWDYN 选项	描述
FI DD	指定要分配的 ddname。
FILEDATA	针对顺序存取方法服务，指定将数据作为文本还是二进制进行处理。
OLD SHR MOD NEW SYSOUT	指定数据集状态。
REUSE	指定已命名数据集在执行函数前被释放。
HOLD	指定输出数据集将设为持有，直到用户或操作员将其释放。
KEEP DELETE CATALOG UNCATALOG	指定数据集被释放后的处置。
RECORGLS)	创建 VSAM 线性数据集
MSG	定向分配消息 注: 该选项可以使用，但由于 IBM MQ Managed File Transfer 使用该选项将错误信息定向到传输日志，因此使用该选项可能导致不可预测的行为。

相关参考

第 497 页的『fteCreateTransfer（创建新的文件传输）』

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

第 484 页的『fteCreateTemplate（新建文件传输模板）』

fteCreateTemplate 命令创建文件传输模板，您可以保留该模板以供将来使用。唯一的必需参数是 **-tn** (*template_name*) 参数。其他所有参数都是可选的，但如果指定源文件，还必须提供目标文件。类似地，如果指定目标文件，还必须指定源文件规范。

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 **agent.properties**，其中必须包含代理用于连接到其队列管理器的信息。**agent.properties** 文件还可以包含用于更改代理程序行为的属性。

传输文本文件

文本文件传输包括将文件代码页从一个代码页转换为另一个。文本文件传输还包括在系统之间转换 CRLF（回车-换行符）字符。本主题概述 IBM MQ Managed File Transfer 的文本文件传输行为。

除非另行指定，否则将从文件源系统的缺省代码页转换为其目标系统的缺省代码页。或者，文本文件传输执行换行转换，这意味着目标文件的换行字符是其目标平台的本地换行符。您可以指定用于读取源文件和写入目标文件的代码页，来覆盖系统上使用的缺省代码页。您还可以指定要用于目标文件的行尾字符序列。有关更多信息，请参阅主题 第 497 页的『fteCreateTransfer（创建新的文件传输）』和第 222 页的『使用传输定义文件』。

文本文件传输在代码页之间执行简单的代码点替换。文本文件传输不执行复杂传输或数据转换，例如，双向数据或文本形成的可视和逻辑形式之间的转换。

表 67: 针对所有平台的文本文件传输行为		
区域	缺省行为	是否可以更改此行为?
源文件编码	源平台编码	Yes 在指定源文件编码并且源是数据集时，编码必须是 EBCDIC 代码页，否则传输将失败。同样，如果目标是数据集，那么目标编码必须为 EBCDIC 代码页。
源文件行尾字符序列	将单个 (LF) 或 (CRLF) 序列转换为目标行尾字符序列	否

表 67: 针对所有平台的文本文件传输行为 (继续)		
区域	缺省行为	是否可以更改此行为?
目标文件编码	目标平台编码	Yes 在指定源文件编码并且源是数据集时，编码必须是 EBCDIC 代码页，否则传输将失败。同样，如果目标是数据集，那么目标编码必须为 EBCDIC 代码页。
目标文件行尾字符序列	目标平台 EOL	Yes
针对源或目标中无法映射或格式不正确字符的文本替换字符序列	空，意味着如果存在无法映射字符或格式不正确的字符，传输将失败。您可以使用 <code>textReplacementCharacterSequence</code> 属性来指定替换文本，如 第 591 页的『agent.properties 文件』 中所述。	Yes

z/OS 数据集

在以文本方式访问数据集记录时，每条记录表示单独一行。记录中不存在换行字符，但对于 ASA 格式数据集，ASA 格式控制代码字符是表示换行（或其他控制字符）的集合。在将带有终止换行符的文本行写入记录时，将根据需要自动移除换行字符或者设置适当的 ASA 控制代码。在读取记录时，会自动向返回数据追加换行字符。对于 ASA 格式数据集，根据需要，该字符可以是针对记录的 ASA 控制代码的多个换行或一个换页。

此外，对于固定格式的数据集，在读取记录时，会在非空格字符的记录中的最后一个字符之后追加换行，从而使固定格式的数据集适合存储文本。

表 68: 特定于 z/OS 的其他文本文件传输行为		
区域	缺省行为	是否可以更改此行为?
最大行长度	目标数据集 LRECL 或 BLKSIZE 设置（根据需要）	否
合并过长行	合并。根据需要行拆分为多条记录。块。	否

在运行 IBM MQ Managed File Transfer 代理时，环境变量 `_EDC_ZERO_RECLLEN` 始终设置为“Y”。对于可变和固定块数据集，该设置使 IBM MQ Managed File Transfer 文本传输行为与 FTP 相同。然而，对于未定义的格式数据集，IBM MQ Managed File Transfer 将单个空格行转换为一个空行，并保留空行。FTP 将空行转换为一个空格行，并保留单个空格行。表 3 描述 IBM MQ Managed File Transfer 行为以及 FTP 行为的差别。

数据集的格式还确定如何将每个文本行写入一条记录。对于非 ASA 格式数据集，不会将换行和回车字符写入记录。对于 ASA 格式数据集，每条记录中的第一个字节是表示行尾、换页和其他代码的 ASA 控制代码（根据需要）。因为 ASA 控制代码位于每条记录开头，如果源文本文件不是以换行字符序列开头，那么将插入空格(' ') ASA 控制字符序列（这等于换行）。这意味着，如果将 ASA 数据集传输到文件，那么将在文件开头显示一个空行。

表 69: IBM MQ Managed File Transfer 的数据集行为				
数据集格式	文件中的原始文本行	数据集记录	读取数据集记录	FTP 读取行为
已纠正的块	空行	填充空格的记录	空行	与 MQMFT 相同

表 69: IBM MQ Managed File Transfer 的数据集行为 (继续)

数据集格式	文件中的原始文本行	数据集记录	读取数据集记录	FTP 读取行为
已纠正的块	单个空格	填充空格的记录	空行	与 MQMFT 相同
变量块	空行	空记录	空行	与 MQMFT 相同
变量块	单个空格	单个空格记录	单个空格	与 MQMFT 相同
未定义	空行	单个空格记录	空行	单个空格
未定义	单个空格	单个空格记录	空行	单个空格

相关参考

第 703 页的『文件传输准则』

根据传输源和目标的操作系统，以及使用二进制还是文本方式传输，提供有关期望行为的指南。

第 715 页的『在 Connect:Direct 和 IBM MQ Managed File Transfer 之间传输文本文件』

文本传输涉及从一个代码页到另一个代码页的文件代码页转换。文本文件传输还涉及系统之间的 CRLF（回车换行）字符转换。本主题总结了 MQMFT 代理和 Connect:Direct 节点之间的传输中的文本文件传输行为。

第 745 页的『可用代码页』

本参考主题列出了 IBM MQ Managed File Transfer 在各种平台支持的可用于文本文件转换的所有字符编码格式。

在 Connect:Direct 和 IBM MQ Managed File Transfer 之间传输文本文件

文本传输涉及从一个代码页到另一个代码页的文件代码页转换。文本文件传输还涉及系统之间的 CRLF（回车换行）字符转换。本主题总结了 MQMFT 代理和 Connect:Direct 节点之间的传输中的文本文件传输行为。

有关 IBM MQ Managed File Transfer 中文本传输行为的信息，请参阅第 713 页的『传输文本文件』。

- 确保 Connect:Direct 网桥节点和用作传输目标的任意 Connect:Direct 节点的网络映射包含正确的平台描述。
 - 如果 Connect:Direct 网桥节点在 Windows 系统上，那么请确保针对网络映射中的每个远程节点，从**操作系统**列表中选择正确的值。
 - 如果远程节点在 Windows 系统上，那么请选择 Windows。
 - 如果远程节点位于 UNIX 或 Linux 系统上，请选择 UNIX。
 - 如果远程节点在 z/OS 系统上，那么请选择 OS/390。

Connect:Direct 网桥不支持到其他操作系统上的远程节点的传输。

- 对于要进行文件传输的每个远程节点，确保在 Connect:Direct 网桥代理配置目录中的 `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，请参阅第 206 页的『配置 ConnectDirectNodeProperties.xml 文件以包含有关远程 Connect:Direct 节点的信息』和第 621 页的『Connect:Direct 节点属性文件格式』。

Connect:Direct 使用网络映射信息来确定要使用的行尾。

- 如果传输目标为 MQMFT 代理，那么该 MQMFT 代理将执行行尾转换。
- 如果传输目标为 Connect:Direct 节点，那么 Connect:Direct 网桥代理将执行行尾转换。

相关参考

第 713 页的『传输文本文件』

文本文件传输包括将文件代码页从一个代码页转换为另一个。文本文件传输还包括在系统之间转换 CRLF（回车-换行符）字符。本主题概述 IBM MQ Managed File Transfer 的文本文件传输行为。

第 425 页的『如果与 Connect:Direct 节点之间的文本传输未正确转换数据，该如何操作』

当以文本方式在 MQMFT 代理和 Connect:Direct 节点之间传输文件时，会执行代码页转换和行尾字符转换。该传输使用 Connect:Direct 网桥节点的网络映射中的操作系统信息来确定远程节点的行尾字符。如果网络映射中的信息不正确，那么行尾字符转换可能未正确执行。

与协议网桥代理传输文件

您可以使用协议网桥代理将文件传输至您的 IBM MQ Managed File Transfer 网络外部的 FTP 或 SFTP 文件服务器，或从这些服务器传输文件。

使用协议网桥传输文件时，该网桥必须有权读取包含要传输的文件的源或目标目录。例如，如果要传输来自只具有执行许可权 (d--x-x-x) 的 /home/fte/bridge 目录中的文件，那么针对该目录的所有传输尝试都将失败，并报告以下错误消息：

```
BFGBR0032E: Attempt to read filename from the protocol file server has failed with server error 550
Failed to open file.
```

文件传输期间，文件通常在目标上作为临时文件写入，然后完成传输后重命名。然而，如果传输的目标是配置为限制写入的协议文件服务器（用户可以向协议文件服务器上载文件，但不能以任何方式更改这些上载文件；有效用户只能写入一次），那么传输的文件将直接写入到目标中。这意味着，如果在传输过程中出现问题，部分写入的文件将保留在目标协议文件服务器上，并且 IBM MQ Managed File Transfer 将无法删除或编辑这些文件。在这种情况下，传输将失败。

确保除了协议网桥代理外，在 IBM MQ Managed File Transfer 网络中还有另一个代理。协议网桥代理是只到 FTP 或 SFTP 服务器的网桥，并且不会向本地磁盘写入传输的文件。如果要与 FTP 或 SFTP 服务器传输文件，那么必须使用协议网桥代理作为文件传输的目标或源（代表 FTP 或 SFTP 服务器），而另一个标准代理作为相应的源或目标。

相关概念

第 275 页的『协议网桥』

通过协议网桥，IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件，而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

在 IBM i 系统上传输文件

如果您使用文本方式的 IBM MQ Managed File Transfer 在 IBM i 系统上传输文件，并且希望转换文件中的数据，请研究本主题中的信息。

IBM i 系统上的每个文件均标记有编码字符集标识 (CCSID) 值，以标识文件的数据编码。例如，包含 EBCDIC 数据的文件可能具有 CCSID 值 037，而包含 ASCII 数据的文件可能具有 CCSID 值 819。

对于文本方式传输，在源文件和目标文件之间存在文件编码差异时，IBM MQ Managed File Transfer 会转换数据。但是，IBM MQ Managed File Transfer 当前会忽略与 IBM i 系统上的文件关联的 CCSID 标记。它会使用运行源代理和目标代理的 JVM 的 JVM 文件编码属性。此属性的缺省值基于语言环境（但您可以使用以下部分中描述的 SystemDefault.properties 文件在 IBM i 系统上覆盖此缺省值：第 716 页的『更改 SystemDefault.properties 文件中的 file.encoding 记录』）。通过该缺省实施，对于以文本方式传输文件的代理，其使用不同文件编码来处理文本文件的能力受到限制。例如，如果不通过适当（即，EBCDIC 或 ASCII）文件编码覆盖来停止并重新启动代理，那么便不能使用相同代理来同时传输包含 EBCDIC 文本的文件和包含 ASCII 文本的文件。在 IBM i V6R1 系统上，可以使用 WRKJVMJOB 选项 7 来显示当前 Java 系统属性，以检查正在运行代理程序作业的 JVM 的文件编码值。（WRKJVMJOB 命令在 IBM i V5R4 系统上不存在。）

如果您计划使用 IBM MQ Managed File Transfer 来传输带有不同文件编码的文本文件，请考虑创建多个代理和多个启动这些代理的用户，以便每个唯一编码都具有能够且准备好传输此类型的数据的代理。

例如，如果您要将包含 EBCDIC 文本的文件（CCSID 值为 037）从 IBM i 系统（源）传输至另一个 IBM i V6R1 系统（目标），并且希望位于目标的文件内容转换为 ASCII 文本（CCSID 值为 819），请完成以下步骤：

1. 选择 JVM 文件编码为 Cp037 的源代理。
2. 选择 JVM 文件编码为 ISO8859_1 的目标代理。
3. 选择文本方式传输以及其他所需的规范。

更改 SystemDefault.properties 文件中的 file.encoding 记录

要启用针对特定编码运行代理的 JVM，请完成以下步骤：

1. 确定由哪些用户启动在 IBM i 系统上运行的代理。该代理可维护 IBM MQ Managed File Transfer 文件传输请求。

根据需要，在该用户的主目录中创建 `SystemDefault.properties` 文件。例如，如果您启动代理，那么请使用 Qshell 来运行以下命令：

```
touch -C 819 /home/your_userID/SystemDefault.properties
```

2. 使用 Qshell，运行 `/qibm/proddata/mqm/bin/fteStopAgent` 命令以根据需要停止代理程序。
3. 更新步骤 1 中所描述的 `SystemDefault.properties` 文件，以确保该文件包含如下所示的记录：

```
file.encoding=java_encoding
```

其中，`java encoding` 对应于文件中包含的数据的类型，并匹配来自下表的 `file.encoding` 值：
[File.encoding 值和 System i5 CCSID。](#)

4. 在步骤 1 中标识的用户必须完成以下步骤：

- a. 仅限在 IBM i V5R4 上：将 `QIBM_PASE_DESCRIPTOR_STDIO` 环境变量（*JOB 作用域）添加到“B”（如果使用 EBCDIC 文件编码）或者“T”（如果使用 ASCII 编码）。例如：

```
ADDENVVAR ENVVAR('QIBM_PASE_DESCRIPTOR_STDIO') VALUE('B') REPLACE(*YES)
```

- b. 如果 Qshell 处于活动状态，请按 **F3 = 退出** 以终止 Qshell。
- c. 启动 Qshell 并根据需要运行 `/qibm/proddata/mqm/bin/fteStartAgent` 命令以重新启动代理程序。

当运行代理的 JVM 的文件编码已更改时，会使用该编码来编写代理日志。如果您要读取代理日志的内容，必须使用针对该编码启用的查看器。

对数据转换使用传输定义

在传输文件时转换数据的替代方法是创建指定文件编码或使用 `fteCreateTransfer` 命令的 `-sce` 和 `-dce` 参数的传输定义。如果在目标是 IBM i 系统的情况下使用这些参数，那么可能导致文件中出现错误的 CCSID 标记。为此，使用位于 IBM i 系统上的文件控制数据转换的建议方法是如先前部分中所述使用 `SystemDefault.properties`。

协议网桥限制

在 IBM i 上，您无法使用协议网桥代理在 SFTP 服务器之间传输 EBCDIC 文件。

相关参考

第 703 页的『文件传输准则』

根据传输源和目标的操作系统，以及使用二进制还是文本方式传输，提供有关期望行为的指南。

第 719 页的『传输位于 IBM i 系统上 QSYS.LIB 文件系统上的保存文件』

IBM MQ Managed File Transfer 支持在两个 IBM i 系统之间传输 QSYS.LIB 文件系统上的保存文件。在请求对保存文件进行文件传输时，请考虑以下信息。

相关信息

[在 IBM i 上安装 WebSphere MQ 服务器](#)

传输驻留在 IBM i 系统上的 QSYS.LIB 文件系统上的物理文件成员

IBM MQ Managed File Transfer 支持在两个 IBM i 系统间传输 QSYS.LIB 文件系统上的物理文件成员。请求对物理文件成员进行文件传输时，请考虑以下信息：

IBM i 上的物理文件成员位于物理文件中，而物理文件位于 IBM i 上的库中。库可以是操作系统随附的标准库之一（例如 QSYS 或 QGPL），也可以是您创建的库。

可以通过两种不同的方式在 IBM i 上标识 QSYS.LIB 文件系统中的物理文件。在 IBM i 命令行上运行 CL 命令时，请使用以下命名语法：

```
FILE(library name/file name) MBR(member name)
```

例如，名为 MYMBR 的物理文件成员位于名为 SOMELIB 的库中的名为 MYFILE 的文件中，那么该物理文件成员的标识为 FILE(SOMELIB/MYFILE) MBR(MYMBR)。您还可以通过指定遵循集成文件系统 (IFS) 命名约定的类似于 UNIX 的路径名，来标识相同的物理文件成员。通过使用 IFS 命名约定，SOMELIB 中的 MYFILE 中的 MYMBR 具有以下路径名：

```
/QSYS.LIB/SOMELIB.LIB/MYFILE.FILE/MYMBR.MBR
```

有关更多信息，请参阅 [QSYS.LIB 文件系统中的路径名](#)。

IBM i 上的 IBM MQ Managed File Transfer 识别 IFS 命名约定，但不支持 CL 命令使用的语法。以下示例解释 MQMFT 的有效和无效路径名。以下是物理文件成员的有效路径名的示例：

```
/QSYS.LIB/SOMELIB.LIB/MYFILE.FILE/MYMBR.MBR
```

此示例假设 MYFILE 是 SOMELIB 库中的物理文件，并包含名为 MYMBR 的成员。

以下是用于物理文件成员传输的有效路径名称的示例：

- /QSYS.LIB/SOMELIB.LIB/MYFILE.FILE（假设 .FILE 为 SAVF，不是物理文件。如果 MYFILE 是物理文件，则传输失败，并显示无效文件类型错误）
- /QSYS.LIB/MYLIB.LIB/（需要物理文件和成员名）
- /QSYS.LIB/SOMELIB.LIB/MYFILE.FILE/MYMBR（成员名必须包含 .MBR 扩展名）
- /QSYS.LIB/SOMELIB.LIB/MYFILE/MYMBR.MBR（物理文件名称扩展必须为 .FILE）

在单个传输请求中传输物理文件中的多个物理文件成员

IBM i 上的 IBM MQ Managed File Transfer 支持以单个传输请求来传输单个物理文件中的多个物理文件成员。您可以指定适当的包含通配符的路径名，如下示例所示：

- ABCLIB 包含带有多个成员的物理文件 MYFILE。要在单个请求中传输所有这些成员，请指定以下路径名：/QSYS.LIB/ABCLIB.LIB/MYFILE.FILE/*.MBR
- XYZLIB 包含物理文件 MYFILE，该文件的成员名仅有一个字符的差别：TEST1.MBR、TEST2.MBR、TEST3.MBR 等。要在单个请求中传输所有这些成员，请指定以下路径名：/QSYS.LIB/XYZLIB.LIB/MYFILE.FILE/TEST?.MBR。

传输多个物理文件成员时不支持以下类型的传输请求，这些请求将导致错误：

- /QSYS.LIB/MYLIB.LIB/*.*
- /QSYS.LIB/MYLIB.LIB/*
- /QSYS.LIB/MYLIB.LIB/*.FILE/MYMBR.MBR
- /QSYS.LIB/MYLIB.LIB/MYFILE*.FILE/*.MBR（文件名不支持通配符，仅成员名支持）
- /QSYS.LIB/MYLIB.LIB/*.FILE/*.MBR
- /QSYS.LIB/MYLIB.LIB/MYFILE.FILE（假设 .FILE 为 SAVF，不是物理文件，如果 MYFILE 是物理文件，传输将会失败，并显示无效文件类型错误）

在非 IBM i 系统中传输物理文件成员

MQMFT 支持在非 IBM i 系统 (例如 Windows，Linux 和 UNIX) 之间传输物理文件成员。所有传输都必须以文本方式进行。以下示例说明了使用非 IBM i 系统时所支持的一些 **fteCreateTransfer** 请求：

- 此命令将 IBM i 上的物理文件成员 FILE (FROMIBMI/FILE1) MBR (FILE1) 传输到 Linux 上的文本文件 /home/qfte/fromibmi/linux.mbr.txt:

```
fteCreateTransfer -da linux -dm QM1 -sa ibmi -sm QM1 -t text -df /home/qfte/fromibmi/
linux.mbr.txt /qsys.lib/fromibmi.lib/file1.file/file1.mbr
```

- 此命令将 IBM i 上的物理文件成员 FILE (FROMIBMI/FILE1) MBR (FILE1) 传输到 Windows 上的文本文件 C:\FTE\fromibmi\windows.mbr.txt:

```
fteCreateTransfer -da windows -dm QM1 -sa ibmi -sm QM1 -t text -df
C:\FTE\fromibmi\windows.mbr.txt /qsys.lib/fromibmi.lib/file1.file/file1.mbr
```

- 此命令将 Windows 上的文本文件 C:\FTE\toibmi\file.txt 传输到 IBM i 上的物理文件成员 FILE (TOIBMI/EXISTS) MBR (WINDOWS):

```
fteCreateTransfer -da ibmi -dm QM1 -sa windows -sm QM1 -t text -df /qsys.lib/toibmi.lib/
exists.file/windows.mbr C:\FTE\toibmi\file.txt
```

以下命令是非 IBM i 系统上的无效物理文件成员传输的示例:

- 此命令之所以失败, 是因为 Windows 上的源文件拥有 .txt 文件扩展名, 但目标目录指定了 .file 扩展名。当传输使用目标目录参数来指定目标物理文件时, 源文件扩展名必须为 .mbr 文件, 例如 C:\FTE\toibmi\file.mbr

```
fteCreateTransfer -da ibmi -dm QM1 -sa windows -sm QM1 -t text -dd /qsys.lib/toibmi.lib/
windows.file C:\FTE\toibmi\file.txt
```

- 缺省传输模式为二进制, 且在传输物理文件成员时必须指定文本模式。

```
fteCreateTransfer -da windows -dm QM1 -sa ibmi -sm QM1 -df C:\FTE\fromibmi\file.bin /qsys.lib/
fromibmi.lib/file1.file/file1.mbr
```

MQMFT 支持传输 QSYS.LIB 文件系统中的物理文件成员, 但不支持传输 QSYS.LIB 文件系统中的源物理文件成员。支持使用所提供的样本用户出口来传输 QDLS 文件系统中的文件。您可以使用 MQMFT 中提供的用户出口样本来执行以下任务:

- 在 QDLS 文件系统中传输文件。
- 采用与 MQMFT 文件监视器相同的方式, 自动从 IBM i 库中传输物理文件成员。
- 在传输期间删除源文件成员时, 请删除空文件对象。

有关更多信息, 请参阅 [第 363 页的『样本 IBM i 用户出口』](#)。

相关参考

[第 703 页的『文件传输准则』](#)

根据传输源和目标的操作系统, 以及使用二进制还是文本方式传输, 提供有关期望行为的指南。

[第 716 页的『在 IBM i 系统上传输文件』](#)

如果您使用文本方式的 IBM MQ Managed File Transfer 在 IBM i 系统上传输文件, 并且希望转换文件中的数据, 请研究本主题中的信息。

传输位于 IBM i 系统上 QSYS.LIB 文件系统中的保存文件

IBM MQ Managed File Transfer 支持在两个 IBM i 系统之间传输 QSYS.LIB 文件系统中的保存文件。在请求对保存文件进行文件传输时, 请考虑以下信息。

IBM i 上的保存文件位于 IBM i 上的库中。库可以是操作系统随附的标准库之一，如 QSYS 或 QGPL，也可以是用户创建的库。以两种不同的方式在 IBM i 上标识 QSYS.LIB 文件系统中的保存文件。在 IBM i 命令行上使用 CL 命令时，使用的命名语法如下：

```
FILE(library name/file name)
```

例如，位于 SOMELIB 库中的名为 MYSAVF 的保存文件将标识为 FILE(SOMELIB/MYSAVF)。

您还可以通过指定遵循集成文件系统 (IFS) 命名约定的类似于 UNIX 的路径名来标识相同的保存文件。请参阅 [QSYS.LIB 文件系统中的路径名](#)，以获取更多信息。通过使用 IFS 命名约定，SOMELIB 中的 MYSAVF 具有以下路径名：

```
/QSYS.LIB/SOMELIB.LIB/MYSAVF.FILE
```

IBM i 上的 IBM MQ Managed File Transfer 识别 IFS 命名约定，但不支持 CL 命令使用的语法。以下示例解释 IBM MQ Managed File Transfer 的有效和无效路径名。

针对保存文件传输的一些有效路径名示例如下：

- /QSYS.LIB/SOMELIB.LIB/MYSAVF.FILE（假设 MYSAVF 保存文件位于库 SOMELIB 中）
- /QSYS.LIB/MYSAVF.FILE（假设 MYSAVF 位于库 QSYS 中）

针对保存文件传输的一些无效路径名示例如下：

- SOMELIB.LIB/MYSAVF.FILE（路径名必须以 /QSYS.LIB 开始）
- /QSYS.LIB/MYLIB.LIB（路径名必须以保存文件名而非库名结束）
- /QSYS.LIB/MYLIB.LIB/（保存文件名是必需的）
- /QSYS.LIB/SOMELIB.LIB/MYSAVF（保存文件名必须具有 .FILE 扩展名）
- /QSYS.LIB/SOMELIB.LIB/MYSAVF.SAVF（保存文件名的扩展名必须为 .FILE）

在单个传输请求中从一个库中传输多个保存文件

IBM i 上的 IBM MQ Managed File Transfer 支持以单个传输请求形式传输单个库中的多个保存文件。您可以指定适当的包含通配符的路径名，如以下示例所示：

- ABCLIB 包含许多保存文件。要在单个请求中传输所有这些文件，请指定以下路径名：

```
/QSYS.LIB/ABCLIB.LIB/*.FILE
```

- XYZLIB 包含多个名称与单个字符不同的保存文件，即：TEST1.FILE，TEST2.FILE，TEST3.FILE，依此类推。要在单个请求中传输所有这些文件，请指定以下路径名：

```
/QSYS.LIB/XYZLIB.LIB/TEST?.FILE
```

传输多个保存文件时不支持以下类型的传输请求，这些请求将导致错误：

- ```
/QSYS.LIB/MYLIB.LIB/*.*
```
- ```
/QSYS.LIB/MYLIB.LIB/*
```

IBM MQ Managed File Transfer 支持传输 QSYS.LIB 文件系统中的保存文件，但不支持传输 QSYS.LIB 文件系统中的其他类型的文件。但是，IBM MQ Managed File Transfer 提供使用保存文件支持和使用预定义 fteAnt 任务的样本，以演示如何在两个 IBM i 系统之间传输完整的库、源物理文件或数据库文件。有关如何定制和使用这些样本的详细信息，请参阅第 354 页的『入门：将 Ant 脚本用于 IBM MQ Managed File Transfer』。

相关参考

第 703 页的『文件传输准则』

根据传输源和目标的操作系统，以及使用二进制还是文本方式传输，提供有关期望行为的指南。

[第 716 页的『在 IBM i 系统上传输文件』](#)

如果您使用文本方式的 IBM MQ Managed File Transfer 在 IBM i 系统上传输文件，并且希望转换文件中的数据，请研究本主题中的信息。

传输世代数据组 (GDG)

IBM MQ Managed File Transfer 在 z/OS 上支持针对源和目标数据集的生成数据组 (GDG)。支持绝对和相对 GDG 名称。写入到新的生成时，基本 GDG 必须存在。

注: 使用 BASEGDG(+n) 在批处理环境中创建 GDG 条目时，稍后无法使用相同的代号（正数）在同一个作业中引用该条目。在作业步骤之间维护相同的 GDG 条目编号是 JCL 提供的一项功能，而使用动态分配更新 GDG 的实用程序功能则未提供此功能。因此，使用 BASEGDG(+1) 创建新代的作业可以发现：成功完成传输后会立即更新 GDG，并且需要将同一个数据集引用为 BASEGDG(0)。

GDG 示例

以下是使用 GDG 的 **fteCreateTransfer** 命令示例。在示例中，名称 BASEGDG 表示现有的基本 GDG 名称。名称 DSET 表示要创建的连续数据集。名称 /u/user/file.dat 表示源数据文件的名称。

该命令会将 file.dat 复制到 BASEGDG 中的新生成中。新生成的绝对名称在传输日志中报告：

```
fteCreateTransfer -sa A1 -da A2 -ds "//BASEGDG(+1)" /u/user/file.dat
```

该命令会将 file.dat 复制到带有 BASEGDG 中指定的绝对名称的生成中：

```
fteCreateTransfer -sa A1 -da A2 -ds "//BASEGDG.G00009V00" /u/user/file.dat
```

该命令会将 BASEGDG 中最新的生成复制到 DSET。该生成的绝对名称在传输日志中报告：

```
fteCreateTransfer -sa A1 -da A2 -ds "//DSET" "//BASEGDG(0)"
```

该命令会将 BASEGDG 中次最新的生成复制到 DSET。该生成的绝对名称在传输日志中报告：

```
fteCreateTransfer -sa A1 -da A2 -ds "//DSET" "//BASEGDG(-1)"
```

相关参考

[第 703 页的『文件传输准则』](#)

根据传输源和目标的操作系统，以及使用二进制还是文本方式传输，提供有关期望行为的指南。

[第 497 页的『fteCreateTransfer（创建新的文件传输）』](#)

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

[第 705 页的『在数据集之间传输』](#)

您可以使用 IBM MQ Managed File Transfer 在 z/OS 数据集之间传输。仔细复审以下行为以确保正确传输数据集。

使用通配符

为文件传输指定源文件名和源文件路径时，可以使用通配符。这使您能够同时选择多个文件。

分布式平台

您可以在分布式平台上使用以下通配符：

?

使用问号 (?) 正好表示一个字符。匹配的文件名中需要存在指定的所有其他字符。

例如，ab?d.jpg 与文件 abcd.jpg、abed.jpg 和 abfd.jpg 相匹配。

★

使用星号字符 (*) 表示零个或零个以上字符。

例如, *.txt 与文件 abc.txt 和 x.txt 相匹配。

模式 *txt 与文件 abc.txt、x.txt 和 newtxt 相匹配, 因为句点 (.) 在文件名中是必需字符。

必须将星号字符 (*) 用双引号引起来。如果不这样做, 命令 shell 将解释该字符, 可能会导致命令失败。

在 UNIX 和 Linux 上, 使用星号字符 (*) 将不包含伪隐藏文件, 例如 .bashrc。

如果对于文件和路径名, 操作系统是区分大小写的 (例如 Windows), 那么模式匹配也将区分大小写。只可以使用通配符指定文件名: 不能在目录名中使用通配符。

协议网桥代理

如果要使用协议网桥代理传输来自 FTP、FTPS 或 SFTP 文件服务器的文件, 那么通配符匹配是区分大小写的, 这与实际运行文件服务器的平台无关。

Connect:Direct 网桥

当传输源是从 Connect:Direct 节点请求文件的 Connect:Direct 网桥代理时, 将不支持通配符。

IBM i

您可以在 IBM i 平台上使用以下通配符:

?

使用问号 (?) 正好表示一个字符。匹配的文件名中需要存在指定的所有其他字符。

例如, ab?d.jpg 与文件 abcd.jpg、abed.jpg 和 abfd.jpg 相匹配。

★

使用星号字符 (*) 表示零个或零个以上字符。

例如, *.txt 与文件 abc.txt 和 x.txt 相匹配。

模式 *txt 与文件 abc.txt、x.txt 和 newtxt 相匹配, 因为句点 (.) 在模式中是必需字符。

有关将通配符与保存文件传输一起使用的其他注意事项, 请参阅[传输位于 IBM i 系统上 QSYS.LIB 文件系统中的保存文件](#)。

z/OS

针对 z/OS 系统, IBM MQ Managed File Transfer 的通配符规则一般遵循标准的 ISPF 通配符约定。连续数据集和分区数据集的特定规则如下:

顺序数据集

引用连续数据集时, 可以按如下所述使用包含星号 (*) 和百分号 (%) 的数据集名称限定符:

★

使用单个星号 (*) 表示至少一个限定符。限定符中的单个星号表示零个或零个以上字符。

★★

使用双星号 (**) 表示零个或零个以上限定符。不能在限定符中使用双星号。

%

使用单个百分号 (%) 表示一个字母数字或本地语言字符。

%%

使用一到八个百分号表示零个或零个以上字符。

分区数据库

引用分区数据集时, 只能为成员名指定通配符。可以按如下所述使用包含星号 (*), 下划线 (_) 和问号 (?) 的数据集名称限定符:

★

使用星号 (*) 字符表示零个或零个以上字符。

—

使用下划线 (_) 字符仅表示一个字符。

?

使用问号 (?) 字符正好表示一个字符。问号是下划线字符的备用选择，作为 ISPF 约定的附加项提供。

目录

缺省情况下，如果使用与子目录匹配的通配符模式创建文件传输，那么不会传输子目录。您可以在 **fteCreateTransfer** 命令上指定 **-r** 参数，以包含与通配符模式匹配的子目录。传输子目录时，将传输子目录的全部内容和结构：包括其所有文件、子目录和隐藏文件。

例如，如果具有名为 **abc** 的目录，那么指定源文件路径 **/opt/abc** 和 **/opt/abc/*** 之间的行为存在差异。在 **/opt/abc** 的情况下，由于传输目录，因此将在目标路径创建名为 **abc** 的目录，并将传输所有文件内容。在 **/opt/abc/*** 的情况下，会将 **abc** 的内容传输到目标路径。

隐藏文件

通配符与隐藏文件不匹配，但在 UNIX 类型的平台上，通配符模式以点字符 (.) 开头时除外。例如：**/opt/.*** 传输 **opt** 目录中的所有隐藏文件。

在 Windows 上，如果要传输隐藏文件，请准确指定文件名或传输包含隐藏文件的目录。

符号链接

符号链接是包含指向另一个文件或目录的指针的文件类型，在 Windows 上称为快捷方式。您可以使用通配符来匹配符号链接文件。但是，从符号链接形式的源创建目标文件时，目标文件将成为硬链接（即常规文件）。无法将符号链接成功地传输到目录，因为这可能会创建递归路径。

传输文件名中具有通配符的文件

文件名自身包含通配符时，可以传输文件。如果准确指定文件名，那么只会传输该文件，而不会传输与通配符匹配的文件集。

例如，如果具有名为 **/opt/abc*.txt** 的文件并为 **/opt/abc*.txt** 创建文件传输，那么只会传输文件 **/opt/abc*.txt**。但如果为 **/opt/ab*.txt** 创建文件传输，那么将传输所有与模式 **/opt/ab*.txt** 匹配的文件，包括文件 **/opt/abc*.txt**。

传输包含通配符的目录路径

用双引号 (") 或单引号 (') 引起任何包含通配符的目录路径，以避免 shell 扩展。如果操作系统在将字符传递到 IBM MQ Managed File Transfer 命令之前扩展通配符，将发生 shell 扩展，并且可能会导致意外行为。

例如，如果在 UNIX 上运行带有 **-gt** 参数的以下 **fteCreateTransfer** 命令，其中 **\${...}** 是来自资源监视器的变量替换：

```
fteCreateTransfer -p QM_VENUS -sa AGT.QM_JUPITER -sm QM_JUPITER -da AGT.QM_NEPTUNE -dm QM_NEPTUNE -r -sd delete
-t binary -de overwrite -jn MONTASK -gt /home/fteadmin/bin/TransferTask.xml -df "${FilePath}" "${FilePath}"
```

shell 解析 **\${FilePath}**，并且不会将其传递至命令。变通方法为使用双引号将 **\${FilePath}** 括起来，即 **"\${FilePath}"**。

即使通配符匹配零个文件，也将传输报告为成功

如果您尝试传输不存在的文件，那么 IBM MQ Managed File Transfer 将此尝试视为失败的传输。如果显式指定文件名（例如，**/a/missing/filename.txt**），并且 MQMFT 找不到该文件，那么会在日志中报告以下错误消息：

```
BFGI00001E: File "/a/missing/filename.txt" does not exist
```

作为这个过程的一部分，源代理（可能找不到该文件）通知目标代理已取消此文件传输（因为源代理无法找到要读取的源文件）。如果您计划在传输后触发一个出口，那么目标代理会为该文件名触发包含 CANCEL_FILE 的 FileExitResultCode 的 DestinationTransferEndExit。

但是，如果您尝试传输通配符（例如，/a/missing/*.txt）但源代理找不到任何与该通配符匹配的文件，那么 MQMFT 将此报告为成功传输。这是因为从技术上而言，源代理被要求传输 0 个文件。会在日志中报告以下错误消息：

```
The transfer request has successfully completed, although no files were transferred.
```

在该示例中，由于传输中未涉及目标代理，因此未调用其出口。

相关参考

第 703 页的『文件传输准则』

根据传输源和目标的操作系统，以及使用二进制还是文本方式传输，提供有关期望行为的指南。

第 497 页的『fteCreateTransfer（创建新的文件传输）』

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

IBM MQ Managed File Transfer 使用的正则表达式

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 Connect:Direct 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 `java.util.regex` API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

有关 Java 正则表达式的更多信息，请参阅 [Java 教程 正则表达式](#)。

示例

要匹配所有模式，请使用以下正则表达式：

```
.*
```

要匹配所有以字符串 `fte` 开头的模式，请使用以下正则表达式：

```
fte.*
```

要匹配所有以字符串 `accounts` 开头、后跟单个数字并且以 `.txt` 结束的模式，请使用以下正则表达式：

```
accounts[0-9]\.txt
```

与用户定义的 Connect:Direct 进程一起使用的替换变量

您可以通过使用特定于 IBM MQ Managed File Transfer 的内置符号变量来定义代入用户定义的 Connect:Direct 进程的值。

要遵循 Connect:Direct 命名约定，IBM MQ Managed File Transfer 使用的所有内部符号变量的格式均为 %FTE，后跟五个大写字母数字字符。有关内置符号变量的更多信息，请参阅 [Connect:Direct 产品文档](#)。

在创建进程以将文件从 Connect:Direct 节点传输到 Connect:Direct 网桥系统时，必须使用内置变量 %FTETFILE 作为 Connect:Direct 进程中 TO FILE 的值。当创建进程以向 Connect:Direct 节点传输来自 Connect:Direct 网桥系统的文件时，必须使用内置变量 %FTEFFILE 作为 Connect:Direct 进程中 FROM FILE 的值。这些变量包含 Connect:Direct 网桥代理用于与 IBM MQ Managed File Transfer 网络之间传输的临时文件路径。

表 70: IBM MQ Managed File Transfer 和 Connect:Direct 使用的内置符号变量	
变量名称	描述
%FTESAGNT	IBM MQ Managed File Transfer 源代理的名称。仅针对从 IBM MQ Managed File Transfer 代理到 Connect:Direct 节点的传输设置该变量。
%FTEDAGNT	IBM MQ Managed File Transfer 目标代理的名称。仅针对从 Connect:Direct 节点到 IBM MQ Managed File Transfer 代理的传输设置该变量。
%FTEPNODE	Connect:Direct 主节点名称。该值始终是作为 Connect:Direct 网桥的一部分的 Connect:Direct 节点的名称。
%FTEPPLAT	运行 Connect:Direct 主节点的平台。该变量的可能值包括 UNIX 和 WINDOWS。该信息由 Connect:Direct 网桥代理提供。
%FTEPUSER	Connect:Direct 进程中使用的 Connect:Direct 主节点用户标识。该信息取自 ConnectDirectCredentials.xml 文件。
%FTEPPASS	要与 %FTEPUSER 变量定义的用户名一起使用的密码。该信息取自 ConnectDirectCredentials.xml 文件。
%FTESNODE	Connect:Direct 辅助节点名。该值始终是传入或传出文件的 Connect:Direct 节点的名称。
%FTESPLAT	运行 Connect:Direct 辅助节点的平台。该变量的可能值为 UNIX、WINDOWS 和 ZOS。该信息取自 ConnectDirectNodeProperties.xml 文件。
%FTESUSER	用于 Connect:Direct 进程中的 Connect:Direct 辅助节点用户标识。该信息取自 ConnectDirectCredentials.xml 文件。
%FTESPASS	要与 %FTESUSER 变量定义的用户名一起使用的密码。该信息取自 ConnectDirectCredentials.xml 文件。
%FTEFFILE	源文件名。仅针对在每个文件作用域提交的 Connect:Direct 进程设置该变量。 当从 IBM MQ Managed File Transfer 代理向 Connect:Direct 节点传输文件时，该值是与 Connect:Direct 网桥在同一系统上的文件的标准位置。 当从 Connect:Direct 节点向 IBM MQ Managed File Transfer 代理传输文件时，该值是在 IBM MQ Managed File Transfer 传输请求中指定为源文件的文件的名称。
%FTEFDISP	进程完成时对源文件的处置。该变量的值与平台有关，等效于 MQMFT 传输请求的值。仅针对在每个文件作用域提交的 Connect:Direct 进程设置该变量。 当从 IBM MQ Managed File Transfer 代理向 Connect:Direct 节点传输文件时，删除或不删除源文件的操作由 IBM MQ Managed File Transfer 网桥代理执行。 当从 Connect:Direct 节点向 IBM MQ Managed File Transfer 代理传输文件时，删除或不删除源文件的操作必须由 Connect:Direct 进程执行。
%FTEFCP	用于源文件的代码页。仅针对在每个文件作用域提交的 Connect:Direct 进程设置该变量。 当从 IBM MQ Managed File Transfer 代理向 Connect:Direct 节点传输文件时，该值为 UTF-8，或者如果传输为二进制传输，那么将不设置该值。 当从 Connect:Direct 节点向 IBM MQ Managed File Transfer 代理传输文件时，该值由 Connect:Direct 指定，或者如果传输为二进制传输，那么将不设置该值。
%FTEFSYSO	传输源的 Connect:Direct SYSOPTS。如果远程 Connect:Direct 节点位于 Linux、UNIX 或 Windows 上，那么此值包含有关传输源的代码页和数据类型的信息。如果远程节点在 z/OS 上，那么该值还包含其他信息。
%FTEFNODE	标识源文件驻留的 Connect:Direct 节点。该变量的值将设置为 PNODE 或 SNODE。

表 70: IBM MQ Managed File Transfer 和 Connect:Direct 使用的内置符号变量 (继续)	
变量名称	描述
%FTETFILE	目标文件名。 仅针对在每个文件作用域提交的 Connect:Direct 进程设置该变量。 当从 IBM MQ Managed File Transfer 代理向 Connect:Direct 节点传输文件时，该值是在 IBM MQ Managed File Transfer 传输请求中指定为目标文件的文件的名称。 当从 Connect:Direct 节点向 IBM MQ Managed File Transfer 代理传输文件时，该值是将文件写入与 Connect:Direct 网桥在同一系统上的位置的标准名称。
%FTETDISP	目标文件的处置。 该变量的值与平台有关，等效于 MQMFT 传输请求的值。 仅针对在每个文件作用域提交的 Connect:Direct 进程设置该变量。 当从 IBM MQ Managed File Transfer 代理向 Connect:Direct 节点传输文件时，创建文件或替换现有文件的操作必须由 Connect:Direct 进程执行。 当从 Connect:Direct 节点向 IBM MQ Managed File Transfer 代理传输文件时，创建文件或替换现有文件的操作由 IBM MQ Managed File Transfer 网桥代理执行。
%FTETCP	用于目标文件的代码页。 仅针对在每个文件作用域提交的 Connect:Direct 进程设置该变量。 当从 IBM MQ Managed File Transfer 代理向 Connect:Direct 节点传输文件时，该值由 Connect:Direct 指定，或者如果传输为二进制传输，那么将不设置该值。 当从 Connect:Direct 节点向 IBM MQ Managed File Transfer 代理传输文件时，该值为 UTF-8，或者如果传输为二进制传输，那么将不设置该值。
%FTETSYSO	传输目标的 Connect:Direct SYSOPTS。 如果远程 Connect:Direct 节点位于 Linux，UNIX 或 Windows 上，那么此值包含有关传输目标的代码页和数据类型的信息。 如果远程节点在 z/OS 上，那么该值还包含其他信息。
%FTETNODE	标识目标文件将驻留的 Connect:Direct 节点。 该变量的值将设置为 PNODE 或 SNODE。
%FTEDTYPE	传输的数据类型或方式。 该变量的可能值为 text 或 binary。 仅针对在每个文件作用域提交的 Connect:Direct 进程设置该变量。
%FTETRID	来自 IBM MQ Managed File Transfer 输出的 48 字符十六进制传输标识。
%FTEJOB	来自 IBM MQ Managed File Transfer 传输请求的作业名。 该变量的值将截断为 256 个字符，并可用于进程记帐数据中。
%FTEPNAME	由 IBM MQ Managed File Transfer 网桥代理生成的 Connect:Direct 进程名称。 该变量的值为 8 个字母数字字符。 该值始终以字母字符开头。
%FTEMETA(key)	来自 IBM MQ Managed File Transfer 传输请求的元数据。 key 的值为元数据的关键字。 key 的值不区分大小写。 关键字 ABC 与关键字 abc 视为同一关键字。 如果将 ABC 和 abc 同时定义为元数据关键字，那么定义的第二个元数据的值将覆盖定义的第一个元数据的值。

下表中包含有关传输中远程 Connect:Direct 节点在 z/OS 平台上时使用的其他内置符号变量的信息。

表 71:	
变量名称	描述
%FTEFDCB	传输源的 DCB 参数值。
%FTEFSPACE	传输源的 SPACE 参数值。
%FTEFLBEL	传输源的 LABEL 参数值。

表 71: (继续)	
变量名称	描述
%FTEFUNIT	传输源的 UNIT 参数值。
%FTEFVOL	传输源的 VOL 参数值。
%FTEFDACL	传输源的 DATACLAS 参数值。
%FTETDCB	传输目标的 DCB 参数值。
%FTETSPCE	传输目标的 SPACE 参数值。
%FTETLBEL	传输目标的 LABEL 参数值。
%FTETUNIT	传输目标的 UNIT 参数值。
%FTETVOL	传输目标的 VOL 参数值。
%FTETDACL	传输目标的 DATACLAS 参数值。
%FTETDSTY	传输目标的 DSNTYPE 参数值。
%FTETLIKE	传输目标的 LIKE 参数值。
%FTETMGCL	传输目标的 MGMTCLAS 参数值。
%FTETSTCL	传输目标的 STORCLAS 参数值。

调用 **ftecxfer** 命令的 **Connect:Direct** 进程文件示例

调用 IBM MQ Managed File Transfer **ftetag** 命令和 **ftecxfer** 命令的 Connect:Direct 进程文件示例。

在该示例中，执行了以下操作：

1. Connect:Direct COPY 语句将文件从运行辅助节点的系统上的 C:\test\from\sent.txt 传输到运行主节点的系统上的 C:\test\tmp\midpoint.txt。
2. Connect:Direct 进程调用 **ftetag** 命令，以在 MQMFT 中创建审计信息。
3. Connect:Direct 进程调用 **ftecxfer** 命令。
4. **ftecxfer** 命令将文件从运行主节点和代理 CD_BRIDGE 的系统上的 C:\test\tmp\midpoint.txt 传输到代理 LINUX_AGENT 所在的系统上的 /test/to/arrived.txt。

```
/*BEGIN_REQUESTER_COMMENTS
  $PNODE$="cd_win01" $PNODE_OS$="Windows"
  $SNODE$="CD_WIN01" $SNODE_OS$="Windows"
  $OPTIONS$="WDO$"
END_REQUESTER_COMMENTS*/

TESTPRO PROCESS
  SNODE=CD_WIN01

  COPY
    FROM (
      FILE=C:\test\from\sent.txt
      SNODE
    )
    TO (
      FILE=C:\test\tmp\midpoint.txt
      PNODE
      DISP=RPL
    )
  COMPRESS Extended

  RUN TASK PNODE
    SYSOPTS="pgm(C:\wmqfte\bin\ftetag) args(C:\test\tmp\midpoint.txt)"

  RUN TASK PNODE
    SYSOPTS="pgm(C:\wmqfte\bin\ftecxfer) args(-qmgrname QM_CDBA -connname fish.example.com(1441)
    -channelname SYSTEM.DEF.SVRCONN
    -sa CD_BRIDGE -da LINUX_AGENT -sm QM_CDBA -dm QM_LINUX -de overwrite -df /test/to/arrived.txt
    C:\test\tmp\midpoint.txt"
```

相关概念

第 300 页的『[使用 Connect:Direct 进程来提交 IBM MQ Managed File Transfer 传输请求](#)』

您可以通过 Connect:Direct 进程向 Connect:Direct 网桥代理提交传输请求。IBM MQ Managed File Transfer 提供了可以从 Connect:Direct 进程中的 **RUN TASK** 语句调用的命令。

相关任务

第 301 页的『[通过使用 Connect:Direct 请求程序来创建和提交调用 IBM MQ Managed File Transfer 的 Connect:Direct 进程](#)』

Connect:Direct 请求程序是一种图形用户界面，可用于创建和提交调用 IBM MQ Managed File Transfer 的 Connect:Direct 进程。

Connect:Direct 网桥代理的限制

Connect:Direct 网桥代理配置为用于与 Connect:Direct 节点之间传输文件。Connect:Direct 网桥代理无法执行一些功能。

- Connect:Direct 网桥代理无法从队列中读取消息或向队列写入消息。在文件到消息传输中不能充当目标代理，或在消息到文件传输中不能充当源代理。
- 无法在 Connect:Direct 网桥代理上定义资源监视器。
- Connect:Direct 网桥代理不能同时充当传输的源和目标。无法通过 Connect:Direct 网桥实现从 Connect:Direct 节点到 Connect:Direct 节点的传输。
- Connect:Direct 网桥代理不支持在传输之前或之后调用的用户出口。Connect:Direct 网桥代理支持凭证映射出口。有关更多信息，请参阅第 209 页的『[通过出口类映射 Connect:Direct 的凭证](#)』。
- 对于将 Connect:Direct 网桥代理作为源代理的传输，无法定义 presrc 或 postsrc 程序调用。有关更多信息，请参阅第 968 页的『[程序调用嵌套元素](#)』。
- 对于将 Connect:Direct 网桥代理作为目标代理的传输，无法定义 predst 或 postdst 程序调用。有关更多信息，请参阅第 968 页的『[程序调用嵌套元素](#)』。
- 如果源代理为 Connect:Direct 网桥代理，那么无法在源规范中指定通配符。
- 在从 Connect:Direct 节点传输文件或数据集时，如果将源处置 (**-sd**) 指定为 delete，那么其行为将有别于通常的源处置行为。将出现以下某种情况：
 - 如果 Connect:Direct 使用 IBM MQ Managed File Transfer 生成的进程从源移动文件或数据集，那么指定 delete 选项将导致传输失败。要指定删除源文件，您必须提交用户定义的 Connect:Direct 进程。有关更多信息，请参阅第 297 页的『[通过文件传输请求提交用户定义的 Connect:Direct 进程](#)』。
 - 如果 Connect:Direct 使用用户定义的进程从源移动文件或数据集，那么该参数将通过 **%FTEFDISP** 内置符号变量传递给此进程。用户定义的进程确定是否删除源。传输返回的结果取决于用户定义的进程返回的结果。

相关概念

第 288 页的『[Connect:Direct 网桥](#)』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

协议网桥提供的 FTPS 服务器支持

协议网桥支持 FTPS 协议的子集，如 RFC-2228、RFC-4217 和名为 *Secure FTP over SSL* 的因特网草案所定义。

有关协议网桥代理与 FTPS 服务器之间的连接的有效密码套件值的列表，请参阅 IBM SDK 和 Runtime Environment Java Technology Edition V 7 产品文档中的 [密码套件](#)。

支持 FTPS 协议的以下功能：

- 隐式和显式的操作方式。
- 服务器证书验证。
- （可选）使用客户机证书检查进行相互认证。

- (可选) 初始认证完成并且选择了数据通道的保护级别后使用清除控制通道。
- 支持 SHA-2 密码套件和 FIPS 140-2 合规性。需要以下 Java 版本: IBM JRE 6.0 SR13 FP2, 7.0 SR4 FP2 或更高版本。

不支持 FTPS 协议功能和运行时环境的以下功能:

- 使用 **ADAT** 命令进行额外的安全数据交换。
- 将 FTPS 仅用于通道加密, 即, 服务器证书未经验证的情况。
- 使用 **PROT** 命令选择 **清除**, **安全** 或 **保密** 级别的保护。
- 使用 **MIC**, **CONF** 和 **ENC** 命令对每个命令进行加密。
- 如果服务器不支持显式 FTPS, 那么回退至 FTP 协议。使用协议网桥提供的 FTP 支持来处理此类服务器。
- 使用 **FEAT** 命令来确定 FTPS 服务器的可用功能。
- 使用针对 DN 字段的模式匹配来进行证书验证。
- 证书撤销检查。
- 通过发出受信认证中心来对证书进行验证。
- 显式选择可用于建立会话的 SSL 协商阶段的密码套件。
- 使用特定于 z/OS 或 IBM i 的扩展, 这些扩展将密码术与操作系统相集成。尤其是, 使用 z/OS 密钥环或非分层文件系统来存储密钥和信任信息, 例如数据集。如果这些功能由 JVM 进行透明化管理并且无需显式应用程序代码, 那么会使用密码硬件和卸载引擎。

相关概念

第 275 页的『[协议网桥](#)』

通过协议网桥, IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件, 而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

协议网桥支持 SFTP 服务器

协议网桥支持由 IETF 因特网草稿授权 SSH 文件传输协议版本 6 草稿 13 定义的 SFTP 协议。

使用 SFTP 协议连接到文件服务器时, 协议网桥代理支持以下密码:

- blowfish-cbc
- 3des-cbc
- aes128-cbc
- aes192-cbc
- aes256-cbc
- aes128-ctr
- aes192-ctr
- aes256-ctr
- 3des-ctr
- arcfour
- arcfour128
- arcfour256

缺省情况下, 协议网桥代理使用的密码列表为 aes128-cbc,aes192-cbc,aes256-cbc。有关如何配置协议网桥代理以使用指定不同密码的信息, 请参阅第 611 页的『[协议网桥属性文件格式](#)』。

认证方法

如果为 ProtocolBridgeCredentials.xml 文件中的单个用户提供了带有专用密钥和服务器密码的 IBM MQ Managed File Transfer (MFT) 协议网桥代理代码，那么缺省情况下，MFT 协议网桥代理会将 JSch 库配置为在建立连接时使用两种认证方法 (如果 SFTP 文件服务器需要)。

如果在 ProtocolBridgeCredentials.xml 文件中为单个用户配置了专用密钥和服务器密码，但 SFTP 文件服务器仅需要其中一种认证方法，那么 MFT 协议网桥代理会将 JSch 库配置为使用公用/专用密钥认证而不是基于密码的认证。

如果 SFTP 文件服务器拒绝尝试使用公用/专用密钥认证，那么使用 JSch 库的 MFT 协议网桥代理将尝试基于用户名和密码的认证。

如果其中任何一个认证成功，那么将建立与 SFTP 文件服务器的连接。

要为与 MFT 协议网桥代理相关联的 ProtocolBridgeCredentials.xml 文件配置专用密钥和密码认证，需要指定：

- 元素中的 **serverPassword** 属性 (具有关联值)，用于从 MFT 用户名映射到协议服务器用户名，以及
- 由父元素定义的 MFT 用户的元素。

例如，语法可以如下所示：

```
-----BEGIN RSA PRIVATE KEY-----  
...  
-----END RSA PRIVATE KEY-----
```

键盘交互方法

MFT 协议网桥代理使用 JSch，第三方库来连接到 SFTP 文件服务器。您可以配置 JSch 库，以便在 ProtocolBridgeCredentials.xml 文件中未指定专用密钥时，它可以尝试使用 键盘-交互式 方法向 SFTP 文件服务器进行认证。

请注意，仅当 SFTP 文件服务器使用字符串 password: (大写，小写或混合大小写) 提示输入密码时，使用 键盘-交互式 方法进行认证才有效。在使用 键盘-交互式 认证方法并且 SFTP 文件服务器使用不同于 password: 的字符串进行响应的情况下，连接尝试失败。

当 SFTP 文件服务器响应使用此字符串的初始连接尝试时，协议网桥代理将使用 JSch 库发送在 ProtocolBridgeCredentials.xml 文件中 user 元素的 **serverPassword** 属性中配置的密码。

相关信息

[协议网桥](#)

FIPS 支持

IBM MQ Managed File Transfer 支持在从代理、命令和 IBM MQ Explorer 到队列管理器的客户机连接中使用符合 FIPS 的密码术模块。到队列管理器的所有 SSL 连接都仅使用 TLS 协议。提供针对 JKS 和 PKCS#12 密钥库类型的支持。

指定您是否希望启用针对代理、协调队列管理器或命令队列管理器的 FIPS 支持，如下所示：

- 如果您要针对特定代理启用 FIPS，请在此代理的 agent.properties 文件内设置适当的 agentSsl 属性。有关更多信息，请参阅第 634 页的『SSL 属性』。
- 如果您要针对特定协调队列管理器启用 FIPS，请在此协调队列管理器的 coordination.properties 文件内设置适当的 coordinationSsl 属性。有关更多信息，请参阅第 634 页的『SSL 属性』。
- 如果您要针对特定命令队列管理器启用 FIPS，请在此命令队列管理器的 command.properties 文件内设置适当的 connectionSsl 属性。有关更多信息，请参阅第 634 页的『SSL 属性』。

在 IBM MQ Managed File Transfer for IBM i 上不支持 FIPS。

在协议网桥或 Connect:Direct 网桥连接上不支持 FIPS。

有关 IBM MQ 和 FIPS 以及所需的配置步骤的更多信息，请参阅[联合信息处理标准 \(FIPS\)](#)。

如果您要使用 FIPS，那么 CipherSuite 必须符合 FIPS，否则连接会失败。有关 IBM MQ 支持的 CipherSpecs 的更多信息，请参阅 [IBM MQ Java 类中的 SSL/TLS CipherSpecs 和 CipherSuites](#) 和 [针对 JMS 的 IBM MQ 类中的 SSL/TLS CipherSpecs 和 CipherSuites](#)。

记录器所使用的数据库表

在安装并配置记录器后，会创建以下数据库表：

AUTH_EVENT

与权限检查相关的事件，通常是由于特权不足而拒绝请求。

- **ID**：行标识。
- **ACTION**：所发生的操作的类型。
- **COMMAND_ID**：请求事件的原始消息的 WebSphere MQ 消息标识。在传输请求的情况下，这还将是传输标识。
- **TIME**：发生事件的时间。
- **ORIGINATOR_MQ_USER**：WebSphere MQ 消息中包含的用户标识，权限检查是针对该用户标识执行的。
- **AUTHORITY**：请求的操作所需的权限。
- **ORIGINAL_XML_REQUEST**：命令消息的有效内容，表明拒绝了什么操作。
- **RESULTCODE**：标识结果的数字代码。
- **RESULT_TEXT**：用于说明权限事件的结果的消息。

CALL

操作系统命令、z/OS JCL 作业或 Ant 脚本的远程运行由 IBM MQ Managed File Transfer 管理。调用可以嵌入在传输中，或者由 call_request 行引用。

一个调用（即该表中的一行）可以属于常规传输（在此情况下，TRANSFER_CALLS 用于将其链接到 TRANSFERS 中的相关条目），也可以是自身独立的受管调用（仅通过 Ant 或直接插入消息来提供）。在后一种情况下，会使用 CALL_REQUEST 表而不是 TRANSFERS 表；无需 TRANSFER_CALLS 的等效项，因为每个调用请求只能有一个调用。

- **ID**：行标识。
- **COMMAND**：运行的命令。该字段不包括传递到命令的任何自变量，也不包括命令所在路径。
- **TYPE**：命令的类型，如 Ant 或 JCL。
- **RETRIES**：请求的重试次数。
- **RETRY_WAIT**：最初请求的在两次重试之间的等待时间间隔（秒）。
- **SUCCESS_RC**：表明命令成功完成的返回码。如果接收到任何其他代码，那么将报告运行失败。
- **EXECUTED_COMMAND**：所运行命令的全名，包括路径。
- **CAPPED_RETRIES**：可用的重试次数；如果代理的重试限制低于所请求的重试次数，那么该数字可能小于请求的重试次数。
- **CAPPED_RETRY_WAIT**：两次重试之间所用的时间间隔；如果代理的已配置限制低于所请求的重试等待时间，那么该数字可能小于请求的时间间隔。
- **OUTCOME**：该调用整体上是否成功。如果尝试了多次，会在 CALL_RESULT 表中单独记录每次尝试的结果。
- **PRIORITY**：当该调用的类型为 os4690background 时，指定给后台应用程序的应用程序优先级。
- **MESSAGE**：当该调用的类型为 os4690background 时，后台应用程序的初始状态消息。如果类型不是 os4690background，那么将包含 NULL。

CALL_ARGUMENT

为调用的命令提供的自变量或参数。

- **ID**: 行标识。
- **CALL_ID**: 与自变量关联的调用。
- **KEY**: 键或名称（在自变量为键值对的情况下）。
- **TYPE**: 自变量的类型：一些是操作系统命令的位置参数，另一些是用于 Ant 的指定属性。
- **VALUE**: 自变量的值。

CALL_REQUEST

不属于文件传输的命令调用的工具。您可以使用 Ant 以及使用直接 XML 注入来提交 ManagedCall 消息。

- **ID**: 受管调用请求的十六进制标识。
- **CALL_ID**: CALL 表中描述该调用的行的数据库标识。
- **ACTION_TIME**: 发生操作的时间。
- **AGENT**: 运行命令所在的代理。
- **AGENT_QM**: 运行命令所在的代理所用的队列管理器。
- **ARCHITECTURE**: 运行代理所在的系统的机器体系结构。
- **OS_NAME**: 运行代理所在的操作系统的名称。
- **OS_VERSION**: 操作系统的版本。
- **ORIGINATOR_HOST**: 从中提交调用请求的主机名。
- **ORIGINATOR_USER**: 提交调用请求的用户名称，如请求 XML 中所报告。
- **ORIGINATOR_MQ_USER**: 提交调用请求的用户名称，如请求的 WebSphere MQ 消息描述符中所包含。
- **JOB_NAME**: 用户指定的作业名。
- **RESULTCODE**: 调用的整体结果代码。
- **RESULTTEXT**: 调用的整体结果消息。

CALL_RESULT

调用一条命令的详细结果。如果启用了重试，那么一个调用可能有多个结果。

- **ID**: 行标识。
- **CALL_ID**: CALL 表中该结果所适用的行的数据库标识。
- **SEQUENCE**: 该结果所适用的尝试（在尝试了多次的情况下）。
- **OUTCOME**: 命令的结果（例如，成功或失败）。
- **RETURN_CODE**: 命令返回码。
- **TIME**: 完成命令的时间。
- **STDOUT**: 来自命令的标准输出流（如果已启动）。
- **STDERR**: 来自命令的标准错误流（如果已启动）。
- **ERROR**: 如果无法启动命令，IBM MQ Managed File Transfer 会生成一条错误消息来说明该问题。

FILE_SPACE_ENTRY

每行都表示已向指定文件空间发送的文件。

- **ID**: 文件空间条目的标识。
- **FILE_SPACE_NAME**: 文件空间的名称。这是文件空间所属用户的名称。
- **TRANSFER_ITEM_ID**: 与该行相关的传输项的标识。
- **ALIAS**: 该文件空间条目的别名。通常，该别名是传输的源文件的名称。
- **DELETED**: 从文件空间删除文件的时间。如果尚未删除文件，那么该值为 null。

METADATA

与传输关联的元数据。

- **ID**: 行标识。
- **TRANSFER_EVENT_ID**: 该元数据所关联的 `transfer_event` 行（如果该元数据与传输相关）。如果元数据与独立受管调用关联，那么该字段为 `null`。
- **STANDALONE_CALL_ID**: 如果元数据与独立受管调用关联，那么为相关受管调用请求的标识。
- **KEY**: 元数据项的名称。
- **VALUE**: 元数据项的值。

MONITOR

根据外部条件触发 IBM MQ Managed File Transfer 操作的资源监视器。

- **AGENT**: 运行监视器所在的代理。
- **ID**: 监视器的十六进制标识。
- **NAME**: 监视器的名称。
- **QMGR**: 运行监视器所在的代理的队列管理器。

MONITOR_ACTION

每行都表示针对监视器所发生的操作（例如，创建和触发）

- **ID**: 行标识。
- **ACTION**: 所发生的操作的类型。
- **JOB_NAME**: 所提交的作业的名称（如果适用）。
- **MONITOR**: 发生该操作所在的监视器。如果由于针对不存在的监视器请求操作而导致该操作失败，那么这可能为 `null`。
- **ORIGINAL_XML_REQUEST**: 如果该操作是 `create` 或 `triggerSatisfied` 操作，那么为触发监视器时启动的 XML 请求。
- **ORIGINATOR_MQ_USER**: WebSphere MQ 消息中包含的发起操作的用户标识。
- **ORIGINATOR_USER**: 提交请求以执行操作的用户名。
- **ORIGINATOR_HOST**: 用户从中提交请求以执行操作的机器。
- **TIME**: 发生操作的时间。
- **UPDATED_XML_REQUEST**: 如果操作是 `triggerSatisfied`，那么为启动的 XML 请求。该请求可能会由于变量替换而不同于最初发出的 XML 请求。

MONITOR_EXIT_RESULT

运行资源监视器出口的结果。

- **ID**: 行标识。
- **ACTION_ID**: 与结果关联的监视器操作。
- **EXIT_NAME**: 产生该结果的出口的名称。
- **RESULTCODE**: 来自出口的数字结果代码。
- **RESULTTEXT**: 来自出口的文本输出（如果有提供）。

MONITOR_METADATA

与资源监视器关联的元数据项。

- **ID**: 行标识。
- **ACTION_ID**: 与元数据关联的 `monitor_action`。

- **KEY:** 元数据项的名称。
- **PHASE:** 该元数据项是表示最初提交的数据还是变量替换之后的更新版本。
- **VALUE:** 元数据项的值。

SCHEDULE

向代理注册的传输调度。

- **AGENT:** 具有该调度的代理的名称。
- **CREATION_DATE:** 创建该调度的时间点。
- **ID:** 该调度的唯一数据库（而不是代理）标识。
- **ID_ON_AGENT:** 代理针对数据库标识所使用的标识。该标识在代理之间不是唯一的，甚至可能在同一个代理中也不是唯一的（如果重置了代理的持久性状态）。
- **LATEST_ACTION:** 修改该调度的状态的最近操作。

SCHEDULE_ACTION

在发生修改调度状态的事件时，会记录一个操作。

- **ACTION_TYPE:** 发生的操作。
- **ID:** 行标识
- **ORIGINATOR_HOST:** 从中提交更改请求的机器。
- **ORIGINATOR_USER:** 提交更改请求所用的用户名。
- **SCHEDULE_ID:** 该操作所适用的调度。
- **SPEC_AFTERWARDS:** 用于表示该调度在发生操作之后的状态的 schedule_spec。
- **STATUS_CODE:** 用于描述操作结果的数字返回码。
- **STATUS_TEXT:** 操作结果的文本描述。如果操作成功，通常为 null。
- **TIME:** 发生操作的时间点

SCHEDULE_SPEC

个别调度传输的详细信息。

- **ID:** 行标识。
- **DESTINATION_AGENT:** 文件传输到的代理。
- **DESTINATION_QM:** 目标代理所用的队列管理器。
- **REPEAT_COUNT:** 如果调度重复并受出现次数而不是结束时间约束，那么为重复的次数。
- **REPEAT_FREQUENCY:** 调度的传输之间存在的 repeat_interval 数。
- **REPEAT_INTERVAL:** 如果传输重复，那么为发生重复的时间间隔（例如，分钟数或周数）。
- **SOURCE_AGENT:** 从中传输文件的代理。
- **SOURCE_QM:** 源代理所用的队列管理器。
- **START_TIME:** 调度中的第一个传输将发生的时间。
- **START_TIMEBASE:** 针对与传输关联的时间的时基。例如，是根据代理的时区还是管理员的时区来运行。
- **START_TIMEZONE:** 对应于时基且将用于运行调度的时区。

SCHEDULE_ITEM

每个文件（或传输时要匹配的模式）都由 schedule_item 来表示。

- **ID:** 行标识。

- **CHECKSUM_METHOD**: 文件的校验和的计算方式
- **DESTINATION_EXISTS_ACTION**: 目标代理在目标处已存在该文件时所采取的操作。
- **DESTINATION_FILENAME**: 文件传输到的文件或目录。
- **DESTINATION_QUEUE**: 文件到消息传输的目标队列名称。
- **DESTINATION_TYPE**: destination_filename 列是指文件、目录还是数据集。
- **DESTINATION_TYPE**: destination_filename 列是指文件还是目录。
- **FILE_MODE**: 传输文件所用的方式 (例如, *text* 或 *binary*) 。
- **RECURSIVE**: 在代理根据调度创建传输时, 代理是 (Y) 否 (N) 对源目录进行递归操作。
- **SCHEDULE_SPEC_ID**: 与该项关联的 schedule_spec。
- **SOURCE_DISPOSITION**: 传输完成之后要对源文件执行的操作。
- **SOURCE_FILENAME**: 源文件、目录名称或模式。
- **SOURCE_QUEUE**: 消息到文件传输的源队列名称。

TRANSFER

一个或多个文件的单个传输。

- **TRANSFER_ID**: 传输的十六进制标识。
- **JOB_NAME**: 传输的用户指定作业名。
- **SCHEDULE_ID**: 如果该传输是调度的结果, 那么为相关调度的数据库行标识。
- **START_ID**: 表示传输开始的 transfer_event 的行标识。
- **COMPLETE_ID**: 表示传输结束的 transfer_event 的行标识。
- **RESULTCODE**: 传输的整体结果代码。 以下主题中列出了该列的可能值: [第 405 页的『IBM MQ Managed File Transfer 的返回码』](#)。 这些代码适用于整个传输; 请参阅 [TRANSFER_ITEM.RESULTCODE](#) 以了解各个项的状态。
- **RESULTTEXT**: 传输的整体结果文本 (如果存在) 。
- **STATUS**: 传输的状态。 该列的可能值为已启动、成功、部分成功、失败和已取消。
- **RELATED_TRANSFER_ID**: 与该传输相关的先前传输的十六进制标识。 例如, 如果该传输是使用 Web 网关的文件下载, 那么该字段是指上载该文件的传输。

TRANSFER_CALLS

链接对传输的可运行命令调用

- **ID**: 行标识。
- **POST_DESTINATION_CALL**: 传输完成之后在目标处进行的调用。
- **POST_SOURCE_CALL**: 传输完成之后在源代理处进行的调用。
- **PRE_DESTINATION_CALL**: 传输开始之前在目标代理处进行的调用。
- **PRE_SOURCE_CALL**: 传输开始之前在源代理处进行的调用。
- **TRANSFER_ID**: 与该行中的调用关联的传输。

TRANSFER_CD_NODE

传输中使用的 Connect:Direct 节点的相关信息。

- **PNODE**: 传输中的主节点。
- **SNODE**: 传输中的辅助节点。
- **BRIDGE_IS_PNODE**: 用于表明哪个节点是属于 Connect:Direct 网桥的节点的字符。 如果该值为 Y, 那么主节点为网桥节点。 如果该值为 N, 那么辅助节点为网桥节点。
- **ID**: 该行的标识。

TRANSFER_CORRELATOR

每行包含一个关联字符串以及与传输项关联的数字。

- **CORRELATION_BOOLEAN**: 布尔关联值。以单个字符 Y 表示 true, 并以 N 表示 false。
- **CORRELATION_STRING**: 字符串关联值。
- **CORRELATION_NUMBER**: 数字关联值。
- **ID**: 该行的标识。

TRANSFER_EVENT

与传输相关的事件（开始或结束）。

- **ID**: 行标识。
- **ACTION_TIME**: 发生传输操作的时间。
- **SOURCE_AGENT**: 从中传输文件的代理的名称。
- **SOURCE_AGENT_TYPE**: 从中传输文件的代理的类型。可能为以下值: 1 = STANDARD、2 = BRIDGE、3 = WEB_GATEWAY、4 = EMBEDDED、5 = CD_BRIDGE 和 6 = SFG。
- **SOURCE_QM**: 源代理所用的队列管理器。
- **SOURCE_ARCHITECTURE**: 托管源代理的系统的机器体系结构。
- **SOURCE_OS_NAME**: 源代理机器的操作系统。
- **SOURCE_OS_VERSION**: 源代理机器的操作系统版本。
- **SOURCE_BRIDGE_URL**: 如果源代理为协议网桥代理, 那么为该代理与之形成网桥的数据源的 URL。
- **SOURCE_WEB_GATEWAY**: 从中传输文件的 Web 网关的名称。
- **SOURCE_CD_NODE_ID**: 作为传输源的 Connect:Direct 节点。
- **DESTINATION_AGENT**: 文件传输到的代理的名称。
- **DESTINATION_AGENT_TYPE**: 文件传输到的代理的类型。可能为以下值: 1 = STANDARD、2 = BRIDGE、3 = WEB_GATEWAY、4 = EMBEDDED、5 = CD_BRIDGE 和 6 = SFG。
- **DESTINATION_QM**: 目标代理所用的队列管理器。
- **DESTINATION_BRIDGE_URL**: 如果目标代理为网桥代理, 那么为该代理与之形成网桥的数据源的 URL。
- **DESTINATION_WEB_GATEWAY**: 文件传输到的 Web 网关的名称。
- **DESTINATION_CD_NODE_ID**: 作为传输目标的 Connect:Direct 节点。
- **ORIGINATOR_HOST**: 从中提交传输请求的主机名。
- **ORIGINATOR_USER**: 提交传输请求的用户名称, 由 **fteCreateTransfer** 命令所报告。
- **ORIGINATOR_MQ_USER**: 提交传输请求的用户名称, 如请求的 WebSphere MQ 消息描述符中所包含。
- **ORIGINATOR_WEB_USER**: 提交请求的 Web 网关用户的名称, 在应用程序服务器环境中配置。
- **TRANSFERSET_TIME**: 创建传输集的时间。
- **TRANSFERSET_SIZE**: 传输的项数。
- **TRIGGER_LOG**: 对于涉及触发器的传输定义, 是否记录未导致传输的触发器求值。

TRANSFER_EXIT

每行都表示在文件传输的过程中执行的传输出口。

- **ID**: 行标识。
- **EXIT_NAME**: 出口的名称。
- **TRANSFER_ID**: 该出口所适用的已完成或已取消传输的标识。
- **TYPE**: 出口的类型。这可以是以下某个值: *SourceStart*、*SourceEnd*、*DestinationStart* 或 *DestinationEnd*。

- **STATUS:** 出口返回的值。这可以是 *cancel* 或 *proceed*。
- **SUPPLEMENT:** 用于说明出口状态的可选消息。

TRANSFER_ITEM

每行都表示在传输的过程中发送的文件。

- **DESTINATION_CHECKSUM_METHOD:** 用于计算目标文件的校验和的算法。如果由于传输未成功完成而未计算校验和，那么这可能为 null。
- **DESTINATION_CHECKSUM_VALUE:** 目标文件的校验和值。如果禁用了校验和计算，那么该值可能为 null。
- **DESTINATION_ENCODING:** 如果目标文件以文本形式传输，那么为目标文件上使用的字符编码。
- **DESTINATION_EXISTS_ACTION:** 在目标处存在该文件时要执行的操作。
- **DESTINATION_FILE_SIZE:** 在目标处要使用的文件名或数据集名称的大小。
- **DESTINATION_FILENAME:** 在目标处使用的文件名或数据集名称。
- **DESTINATION_LINEEND:** 如果目标文件以文本形式传输，那么为目标文件中使用的行尾格式。
- **DESTINATION_MESSAGE_QUEUE_NAME:** 在文件到消息的传输期间，从源文件中生成的消息的目标队列。
- **DESTINATION_MESSAGE_GROUP_ID:** 如果生成了多条消息，那么为文件到消息的传输期间，用于从源文件中生成的消息的组标识。
- **DESTINATION_MESSAGE_MESSAGE_ID:** 如果仅生成了一条消息，那么为文件到消息的传输期间，从源文件中所生成消息的消息标识。
- **DESTINATION_MESSAGE_COUNT:** 在文件到消息的传输期间，将源文件分割为的消息数。
- **DESTINATION_MESSAGE_LENGTH:** 在文件到消息的传输期间，从源文件中生成的消息的长度（字节）。仅当您指定输出消息的长度（例如，通过使用 **fteCreateTransfer** 命令的 **-qs** 选项）时，才会设置该值。如果指定 **-qs 20K** 并且源文件的大小为 50 KB，那么生成的三条消息的大小分别为 20 KB，20 KB 和 10 KB。在此情况下，**DESTINATION_MESSAGE_LENGTH** 的值设置为 20480。
- **DESTINATION_CORRELATOR_ID:** 针对目标的相关因子信息的标识。
- **FILE_MODE:** 文件传输方式，例如 *text* 或 *binary*。
- **ID:** 行标识
- **RESULTCODE:** 用于表明该项的传输结果的数字代码。以下主题中列出了该列的可能值：[第 409 页的『针对传输中文件的返回码』](#)。这些代码适用于传输中的个别项；请参阅 [TRANSFER.RESULTCODE](#) 以了解整体的传输结果。
- **RESULT_TEXT:** 传输结果的文本说明。如果传输成功，通常为 null。
- **SOURCE_CHECKSUM_METHOD:** 用于计算源文件的校验和的算法。
- **SOURCE_CHECKSUM_VALUE:** 源文件的校验和值。如果禁用了校验和计算，那么该值可能为 null。
- **SOURCE_DISPOSITION:** 传输完成时要对源文件执行的操作。
- **SOURCE_ENCODING:** 如果源文件以文本形式传输，那么为源文件上使用的字符编码。
- **SOURCE_FILE_SIZE:** 在源处要使用的文件名或数据集名称的大小。
- **SOURCE_FILENAME:** 源文件名称或数据集名称。
- **SOURCE_LINEEND:** 如果源文件以文本形式传输，那么为源文件中使用的行尾格式。
- **SOURCE_MESSAGE_QUEUE_NAME:** 针对消息到文件的传输，目标文件中包含的消息的源队列。
- **SOURCE_MESSAGE_GROUP_ID:** 针对消息到文件的传输，目标文件中包含的消息的组标识。
- **SOURCE_MESSAGE_COUNT:** 针对消息到文件的传输，目标文件中包含的消息数。
- **SOURCE_CORRELATOR_ID:** 针对源的相关因子信息的标识。
- **TRANSFER_ID:** 该项所属的传输。
- **TRUNCATE_RECORDS:** 指示将截断还是合并超过长度的数据集记录。

TRANSFER_ITEM_ATTRIBUTES

每行表示与 TRANSFER_ITEM 表中某行关联的属性名称/值对。

- **ID**: 行标识。
- **TRANSFER_ITEM ID**: 与该属性名称/值对关联的 TRANSFER_ITEM 行。
- **ATTRIBUTE_NAME**: 该属性的名称。例如, DIST
- **ATTRIBUTE_VALUE**: 该属性的值 (如果有)。例如, MIRRORED,CLOSE 或 3

有关 IBM 4690 上 IBM MQ Managed File Transfer 的分发属性的更多信息, 请参阅第 82 页的『文件分发属性』。

TRANSFER_STATS

在传输结束时生成的统计信息集。

- **ID**: 行标识。
- **TRANSFER_ID**: 统计信息所引用的传输。
- **START_TIME**: 开始传输的时间。在繁忙或间歇性连接的系统中, 该时间可能晚于“已启动”消息中报告的时间, 因为后者表示初始处理的开始时间点, 而不是成功数据传输的开始时间点。
- **RETRY_COUNT**: 由于装入或可用性问题而必须重试传输的次数。
- **FILE_FAILURES**: 未能传输的文件数。
- **FILE_WARNINGS**: 传输时针对其报告了警告的文件数。

TRIGGER_CONDITION

基本 IBM MQ Managed File Transfer 条件传输中的一个条件。例如, “文件 example.file 存在”。

- **ID**: 行标识。
- **TRANSFER_EVENT_ID**: 与触发器相关的传输事件。
- **CONDITION_TYPE**: 触发器中使用的检查类型。例如, 文件是否存在, 或者文件的大小。
- **COMPARISON**: 要进行的特定比较。例如, “大于或等于”。
- **VALUE**: 要进行比较的值。
- **FILENAME**: 要检查的文件名。

相关概念

第 151 页的『配置 Managed File Transfer 记录器』

相关参考

第 575 页的『fteStartLogger (启动记录器)』

fteStartLogger 命令启动 IBM MQ Managed File Transfer 日志记录应用程序。

第 548 页的『fteModifyLogger (作为 Windows 服务运行 IBM MQ Managed File Transfer 日志记录应用程序)』

使用 **fteModifyLogger** 命令修改记录器, 以便记录器能够作为 Windows 服务运行。只能在 Windows 平台上使用此命令, 必须由作为 IBM MQ 管理员和 mqm 组成员的用户运行, 并且必须首先使用

fteStopLogger 命令停止记录器。

第 579 页的『fteStopLogger (停止记录器)』

fteStopLogger 命令停止记录器。

记录器的权限

运行记录器的操作系统用户需要具有对记录器队列和 SYSTEM.FTE 主题的特定 IBM MQ 权限。

运行记录器的操作系统用户需要具有以下 IBM MQ 权限:

- 协调队列管理器上的 CONNECT 和 INQUIRE。
- SYSTEM.FTE 主题上的 SUBSCRIBE 许可权。

- 对 SYSTEM.FTE.LOG.RJCT.*logger_name* 队列的 PUT 许可权。
- 对 SYSTEM.FTE.LOG.CMD.*logger_name* 队列的 GET 许可权。

相关参考

第 433 页的『特定于 IBM MQ Managed File Transfer 的资源的组权限』

无需针对可能涉及到的所有对象而为个别用户授予权限，而是请配置以下两个安全组以用于管理 IBM MQ Managed File Transfer 访问控制：FTEUSER 和 FTEAGENT。IBM MQ 管理员应负责创建并填充这些组。管理员可选择扩展或修改此处所述的建议配置。

第 438 页的『用户的 IBM MQ Managed File Transfer 操作权限』

除使用组来管理对资源的访问权之外，您还可以启用其他安全级别来限制用户可采取的代理操作。将代理权限队列上的权限授予用户以给予用户执行特定代理操作的许可权。

写入到目标队列的消息上设置的 IBM MQ 消息属性

当从文件传输到消息时，IBM MQ Managed File Transfer 可在写入到目标队列的第一条消息上设置 IBM MQ 消息属性。如果从“文件到消息”传输失败，那么会设置其他 IBM MQ 消息属性。

通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。请参阅[消息属性](#)。

本主题描述了 **fteCreateTransfer** 和 **fteCreateTemplate** 命令中用于指示应该将消息属性添加到写入目标队列的第一条消息的参数。您还可以指定应使用 **fte:filespec** 参数的 *dstmsgprop* 值将消息属性添加到写入目标队列的第一条消息。

标准属性

您可以使用 **fteCreateTransfer** 命令或 **fteCreateTemplate** 命令上的 **-qmp** 参数来指定是否在传输写入目标队列的第一条消息上设置 IBM MQ 消息属性。要获取如何使用该参数的示例，请参阅主题第 258 页的『示例：在文件到消息传输上设置 IBM MQ 消息属性』

IBM MQ 消息属性包含传输元数据。消息属性名使用前缀 **usr.WMQFTE**。**usr.** 前缀使这些消息属性可用于 JMS 应用程序。

usr.WMQFTETransferId

唯一的十六进制传输标识。

usr.WMQFTETransferMode

文件传输类型：二进制方式或文本方式。

usr.WMQFTESourceAgent

源代理的名称。

usr.WMQFTEDestinationAgent

目标代理的名称。

usr.WMQFTEFileName

源文件的名称。

usr.WMQFTEFileSize

源文件大小（以字节为单位）。

usr.WMQFTEFileLastModified

上一次修改源文件的时间。该值使用毫秒单位，从 1970 年 1 月 1 日 00:00:00 UTC 起度量。

usr.WMQFTEFileIndex

正在传输的文件列表中当前文件的索引。列表中第一个文件的索引为 0。

usr.WMQFTEMqmdUser

提交传输请求的用户的 MQMD 用户标识。

失败属性

如果目标代理在将至少一条消息写入到目标队列后，从“文件到消息”传输失败，那么 IBM MQ Managed File Transfer 会将空消息写入到目标队列。如果将 **-qmp** 参数设置为 true，此空消息将设置两个 IBM MQ 消息属性。有关从“文件到消息”传输失败的示例，请参阅第 262 页的『文件到消息传输失败』。

从文件到消息的传输完全失败后，IBM MQ Managed File Transfer 会将空消息写入到目标队列。如果将 **-qmp** 参数设置为 true，并且消息数据的长度大于 `maxInputOutputMessageLength` 值，那么将在命令行上显示以下错误消息。

```
Name WMQFTEResultCode
Value 40
Name WMQFTESupplement
Value BFGTR0072E: The transfer failed to complete due to the exception BFGI00205E:The message
data length 1290843 being written
to the output queue "M2F@q2" is greater than the maximum allowed 1048576.
```

IBM MQ 消息属性包含有关失败的信息。如同标准消息属性，消息属性名前缀为 **usr.WMQFTE**，并且可供 JMS 应用程序使用。

usr.WMQFTEReturnCode

传输的返回码。有关此返回码的可能值的列表，请参阅主题 [第 405 页的『IBM MQ Managed File Transfer 的返回码』](#)。

usr.WMQFTESupplement

详细描述传输失败的原因的补充消息。

用户定义的属性

使用带 **-md** 参数的 **fteCreateTransfer** 命令指定的元数据可设置为 IBM MQ 消息属性。如果将 **-qmp** 参数设置为 true，那么用户指定的所有元数据都将添加至第一条消息的消息头。

元数据名称会加以前缀 **usr.**。例如，如果元数据为 `department=accounts`，IBM MQ 消息头将设置为 `usr.department=accounts`。

不能使用元数据指定以 **usr.WMQFTE** 或 **usr.com.ibm.wmqfte** 开始的头。如果指定名称以 **WMQFTE** 或 **com.ibm.wmqfte** 开始的元数据，那么此元数据将不在消息属性中使用且将被忽略。

相关概念

[第 250 页的『将数据从文件传输至消息』](#)

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

相关任务

[第 258 页的『示例：在文件到消息传输上设置 IBM MQ 消息属性』](#)

您可以在 **fteCreateTransfer** 命令上使用 **-qmp** 参数来指定是否在传输写入目标队列的第一条消息上设置 IBM MQ 消息属性。通过使用 IBM MQ 消息属性，应用程序可以选择要处理的消息，或在无需访问 IBM MQ 消息描述符 (MQMD) 或 MQRFH2 头的情况下检索有关消息的信息。

相关参考

[第 741 页的『从源队列上的消息中读取的 IBM MQ 消息属性』](#)

在从“消息到文件”传输过程中从源队列读取消息的代理将从该消息中读取 IBM MQ 消息属性。这些属性的值可用于确定传输行为。

[第 405 页的『IBM MQ Managed File Transfer 的返回码』](#)

IBM MQ Managed File Transfer 命令、Ant 任务和日志消息提供返回码以指示是否成功完成了功能。

[第 262 页的『文件到消息传输失败』](#)

如果代理开始将文件数据写入到目标队列后文件到消息传输失败，那么代理会向队列写入一条消息以向使用这些消息的应用程序指明发生失败。

[第 497 页的『fteCreateTransfer（创建新的文件传输）』](#)

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

[第 962 页的『fte:filespec』](#)

fte:filespec 参数用作为其他任务中的嵌套元素。使用 **fte:filespec** 来描述一个或多个源文件，目录或数据集与目标之间的映射。通常在表述要移动或复制的一组文件、数据集或目录时使用该元素。

从源队列上的消息中读取的 IBM MQ 消息属性

在从“消息到文件”传输过程中从源队列读取消息的代理将从该消息中读取 IBM MQ 消息属性。这些属性的值可用来确定传输行为。

用于取消从消息到文件的传输的头

在组中的最后一条消息上设置以下 IBM MQ 消息属性，以取消该组从“消息到文件”传输：

usr.UserReturnCode

必需。传输的返回码。将该头设置为非零值，以指示将取消此传输。

usr.UserSupplement

可选。描述取消传输的原因的文本。

如果从消息到文件的传输的源代理从已将 **usr.UserReturnCode** 消息属性设置为非零值的源队列读取消息，那么它将停止从队列读取消息并在传输日志 XML 中报告此传输已失败。传输日志 XML 包含返回码以及消息头中设置的补充文本。如果目标代理已将数据写入临时文件，那么将从目标中删除该文件。

变量替换所使用的头

从受监视队列中读取的第一条消息中任何 IBM MQ 消息属性的值均可在任务 XML 定义中被替换。用户定义的消息属性前缀为 **usr.**，但不在变量名中包含该前缀。变量名必须以美元符号 (\$) 字符开头，并括在花括号 ({}) 中。例如，`${destFileName}` 将替换为要从源队列中读取的第一条消息的 **usr.destFileName** 消息属性的值。

例如，将消息置于受监视队列中的用户或程序可在组中的第一条消息上设置 IBM MQ 消息属性，以指定要用作文件传输目标的代理以及要将数据传输到的文件的名称。

有关更多信息，请参阅 [第 243 页的『监视队列和使用变量替换』](#)。

相关概念

[第 263 页的『将数据从消息传输到文件』](#)

通过使用 IBM MQ Managed File Transfer 的消息到文件传输功能，您可以将 IBM MQ 队列上的一条或多条消息的数据传输到文件、数据集或用户文件空间。如果您拥有一个可创建或处理 IBM MQ 消息的应用程序，那么可以使用 IBM MQ Managed File Transfer 的消息到文件传输功能，将这些消息传输到 IBM MQ Managed File Transfer 网络中任何系统上的文件。

相关任务

[第 264 页的『配置代理以执行消息到文件的传输』](#)

缺省情况下，代理无法执行消息到文件或文件到消息的传输。要启用该功能，必须将代理属性 **enableQueueInputOutput** 设置为 **true**。

[第 271 页的『示例：无法使用 IBM MQ 消息属性实现消息到文件的传输』](#)

将 **usr.UserReturnCode** IBM MQ 消息属性设置为非零值会造成“消息到文件”传输失败。也可以通过设置 **usr.UserSupplement** IBM MQ 消息属性，指定有关失败原因的补充信息。

相关参考

[第 739 页的『写入到目标队列的消息上设置的 IBM MQ 消息属性』](#)

当从文件传输到消息时，IBM MQ Managed File Transfer 可在写入到目标队列的第一条消息上设置 IBM MQ 消息属性。如果从“文件到消息”传输失败，那么会设置其他 IBM MQ 消息属性。

[第 497 页的『fteCreateTransfer（创建新的文件传输）』](#)

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

用于设置与消息大小关联的 IBM MQ 属性和 IBM MQ Managed File Transfer 属性的指南

在读取或写入各种大小的消息时，您可以更改 IBM MQ 属性和 IBM MQ Managed File Transfer 属性以影响 IBM MQ Managed File Transfer 的行为。

如果从源队列读取或写入目标队列的消息大小超过 1048576 字节 (1 MB)，那么必须将 IBM MQ Managed File Transfer 代理属性 **maxInputOutputMessageLength** 的值增大到大于或等于要读取或写入的最大消息大小的值。

如果源队列上的消息大于 1048576 字节，您必须在源代理上设置 **maxInputOutputMessageLength** 属性。如果目标队列上的消息大于 1048576 字节，您必须在目标代理上设置 **maxInputOutputMessageLength** 属性。有关 **maxInputOutputMessageLength** 属性的更多信息，请参阅 [高级代理程序属性](#)。

- 如果代理写入或读取的队列对于代理队列管理器来说是本地队列，那么您可能必须更改 IBM MQ 队列管理器、队列和通道 **MAXMSGL** 属性。

确保源或目标队列的最大消息大小的值大于或等于 **maxInputOutputMessageLength** 代理属性的值。

确保以下每个 IBM MQ 属性的值（字节）：

- 代理队列管理器的最大消息大小
- The maximum message size of the SYSTEM.FTE.STATE.<代理程序名称> queue
- 客户机通道的最大消息大小（如果您的代理以客户机方式连接到队列管理器）

大于或等于以下计算的结果：

对于“文件到消息”传输（支持最高 100 MB 的文件大小）：
maxInputOutputMessageLength 的值

对于消息到文件的传输：

$3 * (\text{maxInputOutputMessageLength}) + 1048576$ 的值

（该计算源于如下事实：三个检查点可以存储在状态消息中，并且每个检查点可能都必须缓冲，直至数据消息量的最大大小。）

- 如果代理要写入的队列是远程队列，那么您可能必须更改 IBM MQ 队列管理器，队列和通道 **MAXMSGL** 属性。

确保以下每个 IBM MQ 属性的值大于或等于 **maxInputOutputMessageLength** 代理属性的值：

- 代理队列管理器上远程队列管理器传输队列的最大消息大小
- 从代理队列管理器到远程队列管理器的通道的最大消息大小
- 远程队列管理器上目标队列的最大消息大小
- 远程队列管理器的最大消息大小

确保以下每个 IBM MQ 属性的值（字节）：

- 代理队列管理器的最大消息大小
- The maximum message size of the SYSTEM.FTE.STATE.<代理程序名称> queue
- 客户机通道的最大消息大小（如果您的代理以客户机方式连接到队列管理器）

大于或等于以下计算的结果：

对于“文件到消息”传输（支持最高 100 MB 的文件大小）：
maxInputOutputMessageLength 的值

对于消息到文件的传输：

$3 * (\text{maxInputOutputMessageLength}) + 1048576$ 的值

（该计算源于如下事实：三个检查点可以存储在状态消息中，并且每个检查点可能都必须缓冲，直至数据消息量的最大大小。）

如果超出其中某个属性的值，那么代理将停止，代理事件日志中会出现以下错误：

```
BFGUT0002E: An internal error has occurred. Product failure data was captured in file
"FFDC.FTE.20100928170828514.8172766022149157013.log".
BFGSS00025E: An internal error has occurred. The exception is: cc=2 rc=2010 op=put - MQPUT to
SYSTEM.FTE.STATE.<agent_name>
BFGAG00061E: The agent ended abnormally
```

以下 IBM MQ 原因码必须包含在代理事件日志中的此条消息内：

- **rc=2010** 该原因码映射到 **MQRC_DATA_LENGTH_ERROR**，表明已超出客户机通道最大消息大小的值。要解决此问题，请确保代理队列管理器的客户机通道最大消息大小大于或等于以下计算结果：

$$3 * (\text{maxInputOutputMessageLength}) + 1048576$$

- **rc=2030** This reason code maps to **MQRC_MSG_TOO_BIG_FOR_Q** and indicates that the value of the maximum message size of the **SYSTEM.FTE.STATE.<代理程序名称> queue** was exceeded. To resolve this problem ensure that the maximum message size of the **SYSTEM.FTE.STATE.<代理程序名称> queue** is greater than or equal to the result of the following calculation:

$$3 * (\text{maxInputOutputMessageLength}) + 1048576$$

- **rc=2031** 该原因码映射到 **MQRC_MSG_TOO_BIG_FOR_Q_MGR**，表明已超出代理队列管理器的最大消息大小的值。要解决此问题，请确保代理队列管理器的最大消息大小大于或等于以下计算结果：

$$3 * (\text{maxInputOutputMessageLength}) + 1048576$$

如果要传输许多小的消息

如果代理从队列读取或写入队列的消息的平均大小低于 1310 字节，且代理读取或写入了超过 10000 条消息，那么您必须增加队列管理器上未落实消息属性的最大数量，或减少检查点时间间隔中的数据量。

当代理在队列中读写消息时，对应的 **GET** 或 **PUT** 将一起分组到事务。事务中 **GET** 或 **PUT** 的数量由处理检查点时间间隔中所有数据所需的数量确定。检查点时间间隔中的大概数据量使用以下计算从代理属性中确定：

$$\text{Checkpoint interval data size (in bytes)} = \text{agentCheckpointInterval} * \text{agentFrameSize} * \text{agentWindowSize} * \text{agentChunkSize}.$$

缺省检查点数据大小为 $1 * 5 * 10 * 262144$ 字节 = 13107200 字节 (12.5MB)。队列管理器支持的事务中未落实消息的最大数目由 **MaxUncommittedMsgs** 队列管理器属性控制。此属性的缺省值为 10000 条消息。如果平均消息大小小于约 1310 字节，且要写入的消息数超过 10000 条，那么将超出未落实消息的缺省最大数。

如果超出 **MaxUncommittedMsgs** 限制，那么代理将停止，代理事件日志中出现以下错误：

```
BFGSS0024E: The agent has received a reason code of '2024' from the message queue interface (MQI).
The agent cannot continue processing and will now end.
BFGAG0139I: The agent has suspended its current transfers and is now stopping.
```

原因码 2024 映射到：**MQRC_SYNCPOINT_LIMIT_REACHED**。

要解决此问题，请执行以下某项操作

- 增加读写队列的代理连接到的队列管理器的 **MaxUncommittedMsgs** 队列管理器属性的值。请参阅 [MaxUncommittedMsgs \(MQLONG\)](#)。
- 减少检查点时间间隔中的数据量。要执行此操作，请减少以下一个或多个代理属性的值：
 - **agentCheckpointInterval**
 - **agentFrameSize**
 - **agentWindowSize**
 - **agentChunkSize**

要了解有关这些代理属性的信息，请参阅[高级代理属性](#)。

如果将消息持久写入队列

如果要传输到队列并将消息持久写入队列，那么可能需要增加队列管理器日志文件空间的大小，以便能够记录检查点时间间隔中的所有数据。

如果超出队列管理器日志文件空间，那么代理将停止，代理事件日志中出现以下错误：

```
BFGSS0024E: The agent has received a reason code of '2102' from the message queue interface (MQI).  
The agent cannot continue processing and will now end.  
BFGAG0062E: The agent has received MQI reason code '2102'. The agent cannot continue processing and  
will now end.  
BFGAG0061E: The agent ended abnormally
```

原因码“2102”映射到：MQRC_RESOURCE_PROBLEM。

要解决此问题，请增加目标代理队列管理器日志文件空间的大小。

相关概念

第 263 页的『将数据从消息传输到文件』

通过使用 IBM MQ Managed File Transfer 的消息到文件传输功能，您可以将 IBM MQ 队列上的一条或多条消息的数据传输到文件、数据集或用户文件空间。如果您拥有一个可创建或处理 IBM MQ 消息的应用程序，那么可以使用 IBM MQ Managed File Transfer 的消息到文件传输功能，将这些消息传输到 IBM MQ Managed File Transfer 网络中任何系统上的文件。

第 250 页的『将数据从文件传输至消息』

您可以使用 IBM MQ Managed File Transfer 的文件到消息传输功能，将文件中的数据传输到 IBM MQ 队列上的单条或多条消息。

相关参考

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

用于对“消息到文件”传输指定等待时间的指南

指定消息到文件传输时，您可选择使用 `-sqwt` 参数对传输指定等待时间。`-sqwt` 的值是源代理等待源队列上出现消息（如果源队列为空或成为空队列）或源队列上出现完整组（如果指定了 `-sqgi` 属性）的时间长度。

本主题描述 `fteCreateTransfer` 命令中用于指定等待时间的参数。您还可以使用 `fte:filespec` 参数的 `srcqueuetimeout` 值来指定等待时间。

如果 `-sqwt` 参数的值大于或等于目标代理等待源代理完成传输的时间长度，那么传输不会完成。目标代理等待传输完成的时间长度通过以下公式计算得出：

```
transferAckTimeout * transferAckTimeoutRetries
```

属性 `transferAckTimeout` 和 `transferAckTimeoutRetries` 在目标代理 `agent.properties` 文件中设置。要了解有关这些代理属性的更多信息，请参阅第 591 页的『agent.properties 文件』。

为防止传输无法完成，您必须执行以下某个步骤：

- 减少 `-sqwt` 参数的值，使其小于目标代理 `transferAckTimeout` 属性的值。

注: `transferAckTimeout` 属性的缺省值为 60,000 毫秒。`-sqwt` 参数的值以秒为单位指定，将该值设置为 59 或更小。

- 增加目标代理 `transferAckTimeout` 属性的值，使其大于 `-sqwt` 参数的值。

注: `transferAckTimeout` 属性的值以毫秒为单位指定。`-sqwt` 参数的值以秒为单位指定。

相关参考

第 497 页的『fteCreateTransfer（创建新的文件传输）』

`fteCreateTransfer` 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

第 591 页的『agent.properties 文件』

每个代理都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。`agent.properties` 文件还可以包含用于更改代理程序行为的属性。

第 962 页的『fte:filespec』

`fte:filespec` 参数用作为其他任务中的嵌套元素。使用 `fte:filespec` 来描述一个或多个源文件，目录或数据集与目标之间的映射。通常在表述要移动或复制的一组文件、数据集或目录时使用该元素。

可用代码页

本参考主题列出了 IBM MQ Managed File Transfer 在各种平台支持的可用于文本文件转换的所有字符编码格式。

常见编码

以下字符编码格式可以在所有受支持的平台上使用。如果资源文件使用此表中的某种格式进行编码，而您希望以此表中的另一种格式写目标文件，那么可以直接进行此项操作，而无需考虑平台的问题。您也可以使用规范名称或任何别名来指定编码格式。

规范名称	别名
windows-1256	ibm-1256, Cp1256
windows-1255	ibm-1255, Cp1255
windows-1254	Cp1254, ibm-1254
windows-1253	Cp1253, ibm-1253
windows-1252	ibm-1252, Cp1252
windows-1251	ibm-1251, Cp1251
windows-1250	Cp1250, ibm-1250
UTF-8	UTF_8, UTF8
UTF-16LE	X-UTF-16LE, UTF16LE, UTF_16LE, UnicodeLittleUnmarked
UTF-16BE	UTF16BE, UnicodeBigUnmarked, ISO-10646-UCS-2, UTF_16BE, X-UTF-16BE
US-ASCII	Cp367, iso-ir-6, ANSI_X3.4-1968, ANSI_X3.4-1986, default, ASCII, us, iso-646.irv:1983, csASCII, 646, ascii7, ISO646-US, ibm-367, ISO-646.irv:1991, direct
TIS-620	tis620, tis620.2533
IBM-1122	Cp1122, ibm1122
IBM-1006	Cp1006, ibm1006
IBM-037	ibm-37
GB18030	windows-54936, gb18030-2000, ibm-1392
EUC-TW	x-euc-tw, euctw, cns11643, euc_tw
EUC-KR	ibm-euckr, euc_kr, ksc_5601, ks_c_5601-1987, ksc5601_1987, euckr, ksc5601-1987, ibm-970, Cp970, 5601
EUC-JP	x-euc-jp, euc_jp, eucjp, x-eucjp, euc_jp_linux, euc-jp-linux
EUC-CN	x-euc-cn, ibm-euccn, euc_cn, euccn
Big5	big5-0, big5, Big5-HKSCS
IBM-1025	Cp1025, ibm1025
IBM-1026	ibm1026, Cp1026
IBM-1046	Cp1046, ibm1046
IBM-1097	Cp1097, ibm1097
IBM-1098	Cp1098, ibm1098
IBM-1112	ibm1112, Cp1112
IBM-1383	Cp1383, ibm1383
IBM-273	Cp273, ibm273
IBM-277	Cp277, ibm277
IBM-278	Cp278, ibm278
IBM-280	ibm280, Cp280

规范名称	别名
IBM-284	ibm284, Cp284
IBM-285	Cp285, ibm285
IBM-297	ibm297, Cp297
IBM-420	Cp420, ibm420
IBM-860	Cp860, ibm860
IBM-861	ibm861, Cp861
IBM-862	Cp862, ibm862
IBM-863	Cp863, ibm863
IBM-864	Cp864, ibm864
IBM-865	ibm865, Cp865
windows-1257	Cp1257, ibm-1257
windows-1258	Cp1258, ibm-1129, ibm-1258
windows-31j	ms_kanji, cswindows31j, MS932, windows-932
windows-874	MS874
windows-936	MS936, x-mswin-936, 936
windows-949	MS949, Cp1361, ibm-1361, ibm1361, ms1361, ksc5601-1992, x-windows-949
windows-950	MS950, x-windows-950
IBM-857	ibm857, Cp857, csibm857
IBM-856	Cp856, ibm856
IBM-855	Cp855, ibm855
IBM-852	cspcp852, ibm852, Cp852
IBM-850	Cp850, ibm850, cspc850multilingual
IBM-838	Cp838, ibm838
IBM-834	Cp834, ibm834
IBM-775	ibm775, Cp775
IBM-737	Cp737, ibm737
IBM-500	Cp500, ibm500
IBM-437	ibm437, Cp437, cspc8codepage437
IBM-424	ibm424, Cp424
IBM-1123	Cp1123, ibm1123
IBM-1124	Cp1124, ibm1124
IBM-1381	Cp1381, ibm1381
IBM-866	Cp866, ibm866
IBM-868	Cp868, ibm868
IBM-869	ibm869, Cp869
IBM-870	Cp870, ibm870
IBM-871	ibm871, Cp871
IBM-874	ibm874, Cp874
IBM-875	Cp875, ibm875
IBM-921	Cp921, ibm921

规范名称	别名
IBM-922	Cp922, ibm922
IBM-933	Cp933, ibm933
IBM-935	Cp935, ibm935
IBM-937	Cp937, ibm937
IBM-942	Cp942, ibm942
IBM-943	Cp943, ibm943
IBM-948	ibm948, Cp948
IBM-949	ibm949, Cp949
IBM-950	ibm950, Cp950
ISCII91	iscii
ISO-2022-CN	iso2022-cn-cns, iso2022cn-cns, iso-2022-cn-cns, iso2022cn, iso2022-cn
ISO-2022-CN-GB	iso2022-cn-gb, iso2022cn-gb
ISO-2022-JP	iso2022jp, jis, iso2022-jp, iso-2022-jp2, csiso2022jp2, csjisencoding, jis-encoding
ISO-2022-KR	csiso2022kr, iso2022-kr, iso2022kr
ISO-8859-1	iso8859_1, iso8859-1, ibm819, l1, csisolatin1, Cp819, iso-ir-100, iso-8859-1:1987, ibm-819, latin1, 8859-1
ISO-8859-13	iso8859-13, 8859-13, iso8859_13
ISO-8859-15	csisolatin9, iso8859-15, ibm923, latin9, ibm-923, l9, iso8859_15, iso8859_15_fdis, Cp923, latin0
ISO-8859-2	Cp912, ibm912, iso8859-2, iso-8859-2:1987, l2, iso8859_2, csisolatin2, latin2, ibm-912, 8859-2, iso-ir-101
ISO-8859-3	iso8859-3, Cp913, l3, iso8859_3, iso-ir-109, iso-8859-3:1988, latin3, ibm-913, 8859-3, csisolatin3
ISO-8859-4	Cp914, latin4, iso8859_4, l4, iso-8859-4:1988, ibm-914, iso8859-4, 8859-4, csisolatin4, iso-ir-110
ISO-8859-5	csisolatincyrillic, iso-ir-144, cyrillic, iso8859_5, iso-8859-5:1988, ibm-915, 8859-5, Cp915, ibm915, iso8859-5
ISO-8859-6	csisolatinarabic, Cp1089, iso-8859-6:1987, ecma-114, iso-ir-127, asmo-708, iso8859_6, 8859-6, ibm1089, arabic, iso8859-6, ibm-1089
ISO-8859-7	ecma-118, ibm813, csisolatingreek, elot-928, iso-ir-126, Cp813, 8859-7, iso-8859-7:1987, iso8859_7, greek, greek8, ibm-813, iso8859-7
ISO-8859-8	iso-ir-138, iso-8859-8:1988, csisolatinhebrew, hebrew, iso8859-8, 8859-8, ibm-916, iso8859_8, Cp916, ibm916
ISO-8859-9	ibm-920, ibm920, latin5, 8859-9, Cp920, l5, iso8859-9, iso8859_9, csisolatin5, iso-ir-148
JIS0212	
KOI8-R	koi8, ibm-878, cskoi8r, koi8_r
MacArabic	
MacCentralEurope	ibm-1282
MacCroatian	ibm-1284
MacCyrillic	ibm-1283
MacGreek	ibm-1280
MacIceland	ibm-1286
MacRoman	ibm-1275
MacRomania	ibm-1285

规范名称	别名
MacSymbol	Adobe-Symbol-Encoding, ibm-1038
MacTurkish	ibm-1281

源平台缺省编码

如果您没有为源文件或目标文件指定编码，那么将使用此平台的缺省编码。转换由目标代理执行，目标代理的平台必须支持源编码和目标编码才能执行转换。由于目标缺省编码始终在目标代理上受支持，所以不指定该编码也无妨。但是，使用缺省源编码可能并不安全，因为目标代理可能不支持源的缺省编码。

如果您正在使用缺省源编码，那么应当使用本主题中的表格，以确保您的编码组合受支持。

平台	缺省编码
Solaris	ISO-8859-1
System x 上的 SUSE Linux Enterprise Server	UTF-8
IBM i	ISO-8859-1
HP-UX (Itanium)	ISO-8859-1
Linux for IBM Z 系统	UTF-8
AIX	ISO-8859-1
Microsoft Windows	windows-1252
System x 上的 Red Hat Enterprise Linux	UTF-8
z/OS	IBM-1047
Linux on POWER Systems - Big Endian	UTF-8
HP (PA-RISC)	ISO-8859-1

特定于平台的编码

注：以下两张表包含相同的信息。该信息按两种不同的方式进行组织，以帮助您找到正确的信息，具体取决于您要按平台还是按编码进行查找。

编码（按平台）

规范名称以粗体表示，后跟加括号的别名。

仅支持常见编码表中已出现的编码的平台未在此处列出。

平台	受支持的编码（常见代码表中未收录）
Solaris	x-IBM33722 (ibm33722, 33722, ibm-33722_vascii_vpua, ibm-5050, ibm-33722, cp33722) x-IBM930 (cp930, ibm930, ibm-930, 930) x-IBM939 (ibm-939, ibm939, cp939, 939) x-IBM964 (964, cp964, ibm-964, ibm964) x-ISO-2022-CN-CNS (ISO-2022-CN-CNS, ISO2022CN_CNS) x-iso-8859-11 (iso-8859-11, iso8859_11) x-JISAutoDetect (JISAutoDetect) x-MS932_0213 () x-MS950-HKSCS (MS950_HKSCS) x-PCK (pck) x-SJIS_0213 () X-UTF-32BE-BOM (UTF_32BE_BOM, UTF-32BE-BOM) x-MacUkraine (macukraine) x-MacThai (macthai) x-MacHebrew (machebrew) x-MacDingbat (macdingbat) x-KSC5601 (ksc5601) x-JIS0208 (jis_c6226-1983, jis_x0208-1983, csiso87jisx0208, x0208, iso-ir-87, jis0208) x-IBM949C (ibm949c, cp949c, 949c, ibm-949c) x-IBM943C (cp943c, 943c, ibm-943c, ibm943c) JIS_X0201 (jis_x0201, x0201, cshalfwidthkatakana, jis0201) x-windows-iso2022jp (windows-iso2022jp) x-windows-50221 (ms50221, cp50221) x-windows-50220 (cp50220, ms50220) X-UTF-32LE-BOM (UTF_32LE_BOM, UTF-32LE-BOM) x-eucJP-Open (EUC_JP_Solaris, eucJP-open) x-Big5-Solaris (Big5_Solaris) ISO-2022-JP-2 (csISO2022JP2, iso2022jp2) IBM918 (cp918, ebcdic-cp-ar2, ibm-918, 918) IBM1047 (cp1047, 1047, ibm-1047) IBM01149 (cp1149, cp01149, ccsid01149, 1149) IBM01148 (cp1148, ccsid01148, 1148, cp01148) IBM01147 (ccsid01147, cp1147, 1147, cp01147) IBM01146 (ccsid01146, cp01146, cp1146, 1146) IBM01145 (cp1145, cp01145, ccsid01145, 1145) IBM01144 (cp01144, cp1144, ccsid01144, 1144) IBM01143 (cp01143, 1143, ccsid01143, cp1143) IBM01142 (cp01142, cp1142, 1142, ccsid01142) IBM01141 (cp1141, ccsid01141, cp01141, 1141) IBM01140 (ccsid01140, cp01140, 1140, cp1140) IBM00858 (cp858, ccsid00858, 858, cp00858) X-UnicodeLittle (UnicodeLittle) X-UnicodeBig (UnicodeBig) COMPOUND_TEXT (x-compound-text, x11-compound-text) IBM-942C (Cp942C, ibm942C) KOI8-U (koi8_u, ibm-1167) UTF-32 (UCS-4, UTF32, ISO-10646-UCS-4) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE)

平台	受支持的编码（常见代码表中未收录）
System x 上的 SUSE Linux Enterprise Server	windows-1256S (Cp1256s, ibm-1256s) UTF-8J (UTF8J) UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) UTF-32 (UCS-4, UTF32, ISO-10646-UCS-4) PTCP154 (PT154, IBM-1169, Cyrillic-Asian, csPTCP154) KOI8-RU (ibm-1168, koi8_ru) ISO-8859-16 (8859-16, iso8859_16, iso8859-16) ISO-8859-14 (ISO_8859-14:1998, 8859-14, latin8, iso-ir-199, iso8859-14, l8, isoceltic, iso8859_14) IBM01141 (cp1141, ccsid01141, cp01141, 1141) IBM01142 (cp01142, cp1142, 1142, ccsid01142) IBM01143 (cp01143, 1143, ccsid01143, cp1143) IBM01144 (cp01144, cp1144, ccsid01144, 1144) IBM01145 (cp1145, cp01145, ccsid01145, 1145) IBM01146 (ccsid01146, cp01146, cp1146, 1146) IBM01147 (ccsid01147, cp1147, 1147, cp01147) IBM01148 (cp1148, ccsid01148, 1148, cp01148) IBM01149 (cp1149, cp01149, ccsid01149, 1149) IBM1047 (cp1047, 1047, ibm-1047) IBM918 (cp918, ebcdic-cp-ar2, ibm-918, 918) ISO-2022-JP-2 (csISO2022JP2, iso2022jp2) x-Big5-Solaris (Big5_Solaris) x-eucJP-Open (EUC_JP_Solaris, eucJP-open) x-IBM33722 (ibm33722, 33722, ibm-33722_vascii_vpua, ibm-5050, ibm-33722, cp33722) x-IBM930 (cp930, ibm930, ibm-930, 930) x-IBM939 (ibm-939, ibm939, cp939, 939) x-IBM964 (964, cp964, ibm-964, ibm964) x-ISO-2022-CN-CNS (ISO-2022-CN-CNS, ISO2022CN_CNS) x-iso-8859-11 (iso-8859-11, iso8859_11) x-JISAutoDetect (JISAutoDetect) x-MS932_0213 () x-MS950-HKSCS (MS950_HKSCS) x-PCK (pck) x-IBM1363C (ibm1363c, cp1363c, ibm-1363c) x-IBM420S (420s, ibm-420s, csibm420s, ibm420s, cp420s) x-IBM864S (csibm864s, ibm864s, cp864s, 864s, ibm-864s) x-IBM943C (cp943c, 943c, ibm-943c, ibm943c) x-IBM949C (ibm949c, cp949c, 949c, ibm-949c) x-IBM954C (cp954c, 954c, ibm-954c, ibm954c) x-ISO-8859-6S (8859_6s, iso8859-6s, iso8859_6s, iso-8859-6s) x-JIS0208 (jis_c6226-1983, jis_x0208-1983, csiso87jisx0208, x0208, iso-ir-87, jis0208) x-KSC5601 (ksc5601) x-MacDingbat (macdingbat) x-MacHebrew (machebrew) x-MacThai (mac thai) x-MacUkraine (macukraine) x-IBM1046S (ibm-1046s, 1046s, cp1046s, ibm1046s) x-IBM-udcJP (IBM-udcJP) JIS_X0201 (jis_x0201, x0201, cshalfwidthkatakana, jis0201) IBM-939A (Cp939A, ibm939A) IBM-930A (ibm930A, Cp930A) IBM-33722A (Cp33722A, ibm33722A) x-windows-iso2022jp (windows-iso2022jp) x-windows-50221 (ms50221, cp50221) x-windows-50220 (cp50220, ms50220) X-UTF-32LE-BOM (UTF_32LE_BOM, UTF-32LE-BOM) X-UTF-32BE-BOM (UTF_32BE_BOM, UTF-32BE-BOM) x-SJIS_0213 () IBM01140 (ccsid01140, cp01140, 1140, cp1140) IBM00858 (cp858, ccsid00858, 858, cp00858)

平台	受支持的编码（常见代码表中未收录）
System x 上的 SUSE Linux Enterprise Server	IBM-837 (ibm837, Cp837) IBM-836 (ibm836, Cp836) IBM-835 (ibm835, Cp835) IBM-833 (ibm833, Cp833) IBM-808 (Cp808, ibm808) IBM-720 (Cp720, ibm720) IBM-33722C (ibm-eucjp, Cp33722c) IBM-301 (Cp301, ibm301) IBM-300 (Cp300, ibm300) IBM-290 (ibm290, Cp290) IBM-1399 (ibm1399, Cp1399) IBM-1390 (Cp1390, ibm1390) IBM-1388 (Cp1388, ibm1388) IBM-1385 (Cp1385, ibm1385) IBM-1382 (ibm1382, Cp1382) IBM-1088 (Cp1088, ibm1088) IBM-1043 (Cp1043, ibm1043) IBM-1041 (Cp1041, ibm1041) IBM-1027 (Cp1027, ibm1027) CESU-8 (CESU8) COMPOUND_TEXT (x-compound-text, x11-compound-text) GB2312 (gb2312-1980, gb2312-80) GBK (GBK) hp-roman8 (roman8, ibm-1051, r8, Cp1051) IBM-1114 (Cp1114, ibm1114) IBM-1115 (Cp1115, ibm1115) IBM-1351 (Cp1351, ibm1351) IBM-1362 (Cp1362, ibm1362) IBM-1363 (ibm1363, Cp1363) IBM-1364 (Cp1364, ibm1364) IBM-1370 (Cp1370, ibm1370) IBM-1371 (Cp1371, ibm1371) IBM-1380 (Cp1380, ibm1380) IBM-867 (Cp867, ibm867) IBM-897 (Cp897, ibm897) IBM-924 (Cp924, ibm924) IBM-927 (ibm927, Cp927) IBM-932 (ibm932, Cp932) IBM-947 (Cp947, ibm947) IBM-951 (Cp951, ibm951) IBM-954 (ibm954, Cp954) IBM-971 (Cp971, ibm971) ISO-8859-10 (latin6, 8859-10, ISO_8859-10:1992, iso8859_10, iso-ir-157, ibm-919, iso8859-10, l6, csisolatin6)

平台	受支持的编码（常见代码表中未收录）
IBM i	windows-1256S (Cp1256s, ibm-1256s) UTF-8J (UTF8J) IBM-1146 (Cp1146, ibm1146) IBM-1145 (Cp1145, ibm1145) IBM-1144 (ibm1144, Cp1144) IBM-1143 (Cp1143, ibm1143) IBM-1142 (Cp1142, ibm1142) IBM-1141 (Cp1141, ibm1141) IBM-1140 (ibm1140, Cp1140) IBM-1115 (Cp1115, ibm1115) IBM-1114 (Cp1114, ibm1114) hp-roman8 (roman8, ibm-1051, r8, Cp1051) GBK (GBK) GB2312 (gb2312-1980, gb2312-80) COMPOUND_TEXT (x-compound-text, x11-compound-text) CESU-8 (CESU8) IBM-1027 (Cp1027, ibm1027) IBM-1041 (Cp1041, ibm1041) IBM-1043 (Cp1043, ibm1043) IBM-1046S (ibm1046S, Cp1046S) IBM-1047 (Cp1047, ibm1047) IBM-1088 (Cp1088, ibm1088) IBM-1382 (ibm1382, Cp1382) IBM-1385 (Cp1385, ibm1385) IBM-1386 (ibm1386, Cp1386) IBM-1388 (Cp1388, ibm1388) IBM-836 (ibm836, Cp836) IBM-837 (ibm837, Cp837) IBM-858 (Cp858, ibm858) IBM-859 (Cp859, ibm859) IBM-864S (ibm864S, Cp864S) X-UnicodeBig (UnicodeBig) X-UnicodeLittle (UnicodeLittle) IBM-1047_LF (Cp1047_LF, ibm1047_LF) IBM-1141_LF (Cp1141_LF, ibm1141_LF) IBM-33722A (Cp33722A, ibm33722A) IBM-924_LF (Cp924_LF, ibm924_LF) IBM-930A (ibm930A, Cp930A) IBM-939A (Cp939A, ibm939A) IBM-835 (ibm835, Cp835) IBM-833 (ibm833, Cp833) IBM-808 (Cp808, ibm808) IBM-720 (Cp720, ibm720) IBM-420S (Cp420S, ibm420S) IBM-33722C (ibm-eucjp, Cp33722c) IBM-33722 (5050, Cp5050) IBM-301 (Cp301, ibm301) IBM-300 (Cp300, ibm300) IBM-290 (ibm290, Cp290) IBM-1399 (ibm1399, Cp1399) IBM-1390 (Cp1390, ibm1390) IBM-1147 (Cp1147, ibm1147) IBM-1148 (ibm1148, Cp1148) IBM-1149 (Cp1149, ibm1149) IBM-1351 (Cp1351, ibm1351) IBM-1362 (Cp1362, ibm1362) IBM-1363 (ibm1363, Cp1363) IBM-1363C (ibm1363C, Cp1363C) IBM-1364 (Cp1364, ibm1364) IBM-1370 (Cp1370, ibm1370) IBM-1371 (Cp1371, ibm1371) IBM-1380 (Cp1380, ibm1380) IBM-867 (Cp867, ibm867) IBM-897 (Cp897, ibm897)

平台	受支持的编码（常见代码表中未收录）
IBM i	IBM-918 (ibm918, Cp918) IBM-924 (Cp924, ibm924) IBM-927 (ibm927, Cp927) IBM-930 (Cp5026, 5026) IBM-932 (ibm932, Cp932) IBM-939 (Cp5035, 5035) IBM-942C (Cp942C, ibm942C) IBM-943C (ibm943C, Cp943C) IBM-947 (Cp947, ibm947) IBM-949C (Cp949C, ibm949C) IBM-951 (Cp951, ibm951) IBM-954 (ibm954, Cp954) IBM-954C (Cp954c) IBM-964 (ibm-euctw, Cp964) IBM-971 (Cp971, ibm971) ISO-8859-10 (latin6, 8859-10, ISO_8859-10:1992, iso8859_10, iso-ir-157, ibm-919, iso8859-10, l6, csisolatin6) ISO-8859-14 (ISO_8859-14:1998, 8859-14, latin8, iso-ir-199, iso8859-14, l8, isoceltic, iso8859_14) ISO-8859-16 (8859-16, iso8859_16, iso8859-16) ISO-8859-6S (iso8859-6S, iso8859_6S) JIS0201 () JIS0208 () Johab (x-johab) KOI8-RU (ibm-1168, koi8_ru) KOI8-U (koi8_u, ibm-1167) KSC5601 () MacDingbat () MacHebrew () MacThai () MacUkraine () PTCP154 (PT154, IBM-1169, Cyrillic-Asian, csPTCP154) Shift_JIS () UTF-16 (UTF16, Unicode, UTF_16, UCS-2) UTF-32 (UCS-4, UTF32, ISO-10646-UCS-4) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE)

平台	受支持的编码（常见代码表中未收录）
HP-UX (Itanium)	UTF-16 (UTF16, Unicode, UTF_16, UCS-2) MacUkraine () MacThai () MacHebrew () MacDingbat () JISO208 () JISO201 () IBM-949C (Cp949C, ibm949C) IBM-943C (ibm943C, Cp943C) IBM-942C (Cp942C, ibm942C) IBM00858 (cp858, ccsid00858, 858, cp00858) IBM01140 (ccsid01140, cp01140, 1140, cp1140) x-eucJP-Open (EUC_JP_Solaris, eucJP-open) x-IBM33722 (ibm33722, 33722, ibm-33722_vascii_vpua, ibm-5050, ibm-33722, cp33722) x-IBM930 (cp930, ibm930, ibm-930, 930) x-IBM939 (ibm-939, ibm939, cp939, 939) x-IBM964 (964, cp964, ibm-964, ibm964) x-ISO-2022-CN-CNS (ISO-2022-CN-CNS, ISO2022CN_CNS) x-iso-8859-11 (iso-8859-11, iso8859_11) x-JISAutoDetect (JISAutoDetect) x-MS950-HKSCS (MS950_HKSCS) x-PCK (pck) x-windows-50220 (cp50220, ms50220) x-windows-50221 (ms50221, cp50221) x-windows-iso2022jp (windows-iso2022jp) x-Big5-Solaris (Big5_Solaris) IBM918 (cp918, ebcdic-cp-ar2, ibm-918, 918) IBM1047 (cp1047, 1047, ibm-1047) IBM01149 (cp1149, cp01149, ccsid01149, 1149) IBM01148 (cp1148, ccsid01148, 1148, cp01148) IBM01147 (ccsid01147, cp1147, 1147, cp01147) IBM01146 (ccsid01146, cp01146, cp1146, 1146) IBM01145 (cp1145, cp01145, ccsid01145, 1145) IBM01144 (cp01144, cp1144, ccsid01144, 1144) IBM01143 (cp01143, 1143, ccsid01143, cp1143) IBM01142 (cp01142, cp1142, 1142, ccsid01142) IBM01141 (cp1141, ccsid01141, cp01141, 1141)

平台	受支持的编码（常见代码表中未收录）
Linux for IBM Z 系统	windows-1256S (Cp1256s, ibm-1256s) UTF-8J (UTF8J) UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) UTF-32 (UCS-4, UTF32, ISO-10646-UCS-4) PTCP154 (PT154, IBM-1169, Cyrillic-Asian, csPTCP154) KOI8-RU (ibm-1168, koi8_ru) ISO-8859-16 (8859-16, iso8859_16, iso8859-16) ISO-8859-14 (ISO_8859-14:1998, 8859-14, latin8, iso-ir-199, iso8859-14, l8, isoceltic, iso8859_14) IBM01141 (cp1141, ccsid01141, cp01141, 1141) IBM01142 (cp01142, cp1142, 1142, ccsid01142) IBM01143 (cp01143, 1143, ccsid01143, cp1143) IBM01144 (cp01144, cp1144, ccsid01144, 1144) IBM01145 (cp1145, cp01145, ccsid01145, 1145) IBM01146 (ccsid01146, cp01146, cp1146, 1146) IBM01147 (ccsid01147, cp1147, 1147, cp01147) IBM01148 (cp1148, ccsid01148, 1148, cp01148) IBM01149 (cp1149, cp01149, ccsid01149, 1149) IBM1047 (cp1047, 1047, ibm-1047) IBM918 (cp918, ebcdic-cp-ar2, ibm-918, 918) ISO-2022-JP-2 (csISO2022JP2, iso2022jp2) x-Big5-Solaris (Big5_Solaris) x-eucJP-Open (EUC_JP_Solaris, eucJP-open) x-IBM33722 (ibm33722, 33722, ibm-33722_vascii_vpua, ibm-5050, ibm-33722, cp33722) x-IBM930 (cp930, ibm930, ibm-930, 930) x-IBM939 (ibm-939, ibm939, cp939, 939) x-IBM964 (964, cp964, ibm-964, ibm964) x-ISO-2022-CN-CNS (ISO-2022-CN-CNS, ISO2022CN_CNS) x-iso-8859-11 (iso-8859-11, iso8859_11) x-JISAutoDetect (JISAutoDetect) x-MS932_0213 () x-MS950-HKSCS (MS950_HKSCS) x-PCK (pck) x-IBM1363C (ibm1363c, cp1363c, ibm-1363c) x-IBM420S (420s, ibm-420s, csibm420s, ibm420s, cp420s) x-IBM864S (csibm864s, ibm864s, cp864s, 864s, ibm-864s) x-IBM943C (cp943c, 943c, ibm-943c, ibm943c) x-IBM949C (ibm949c, cp949c, 949c, ibm-949c) x-IBM954C (cp954c, 954c, ibm-954c, ibm954c) x-ISO-8859-6S (8859_6s, iso8859-6s, iso8859_6s, iso-8859-6s) x-JIS0208 (jis_c6226-1983, jis_x0208-1983, csiso87jisx0208, x0208, iso-ir-87, jis0208) x-KSC5601 (ksc5601) x-MacDingbat (macdingbat) x-MacHebrew (machebrew) x-MacThai (macthai) x-MacUkraine (macukraine) x-IBM1046S (ibm-1046s, 1046s, cp1046s, ibm1046s) x-IBM-udcJP (IBM-udcJP) JIS_X0201 (jis_x0201, x0201, cshalfwidthkatakana, jis0201) IBM-939A (Cp939A, ibm939A) IBM-930A (ibm930A, Cp930A) IBM-33722A (Cp33722A, ibm33722A) x-windows-iso2022jp (windows-iso2022jp) x-windows-50221 (ms50221, cp50221) x-windows-50220 (cp50220, ms50220) X-UTF-32LE-BOM (UTF_32LE_BOM, UTF-32LE-BOM) X-UTF-32BE-BOM (UTF_32BE_BOM, UTF-32BE-BOM) x-SJIS_0213 () IBM01140 (ccsid01140, cp01140, 1140, cp1140) IBM00858 (cp858,

平台	受支持的编码（常见代码表中未收录）
Linux for IBM Z 系统	IBM-837 (ibm837, Cp837) IBM-836 (ibm836, Cp836) IBM-835 (ibm835, Cp835) IBM-833 (ibm833, Cp833) IBM-808 (Cp808, ibm808) IBM-720 (Cp720, ibm720) IBM-33722C (ibm-eucjp, Cp33722c) IBM-301 (Cp301, ibm301) IBM-300 (Cp300, ibm300) IBM-290 (ibm290, Cp290) IBM-1399 (ibm1399, Cp1399) IBM-1390 (Cp1390, ibm1390) IBM-1388 (Cp1388, ibm1388) IBM-1385 (Cp1385, ibm1385) IBM-1382 (ibm1382, Cp1382) IBM-1088 (Cp1088, ibm1088) IBM-1043 (Cp1043, ibm1043) IBM-1041 (Cp1041, ibm1041) IBM-1027 (Cp1027, ibm1027) CESU-8 (CESU8) COMPOUND_TEXT (x-compound-text, x11-compound-text) GB2312 (gb2312-1980, gb2312-80) GBK (GBK) hp-roman8 (roman8, ibm-1051, r8, Cp1051) IBM-1114 (Cp1114, ibm1114) IBM-1115 (Cp1115, ibm1115) IBM-1351 (Cp1351, ibm1351) IBM-1362 (Cp1362, ibm1362) IBM-1363 (ibm1363, Cp1363) IBM-1364 (Cp1364, ibm1364) IBM-1370 (Cp1370, ibm1370) IBM-1371 (Cp1371, ibm1371) IBM-1380 (Cp1380, ibm1380) IBM-867 (Cp867, ibm867) IBM-897 (Cp897, ibm897) IBM-924 (Cp924, ibm924) IBM-927 (ibm927, Cp927) IBM-932 (ibm932, Cp932) IBM-947 (Cp947, ibm947) IBM-951 (Cp951, ibm951) IBM-954 (ibm954, Cp954) IBM-971 (Cp971, ibm971) ISO-8859-10 (latin6, 8859-10, ISO_8859-10:1992, iso8859_10, iso-ir-157, ibm-919, iso8859-10, l6, csisolatin6)

平台	受支持的编码（常见代码表中未收录）
AIX	windows-1256S (Cp1256s, ibm-1256s) UTF-8J (UTF8J) UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) UTF-32 (UCS-4, UTF32, ISO-10646-UCS-4) UTF-16 (UTF16, Unicode, UTF_16, UCS-2) Shift_JIS () PTCP154 (PT154, IBM-1169, Cyrillic-Asian, csPTCP154) MacUkraine () MacThai () MacHebrew () MacDingbat () KSC5601 () KOI8-U (koi8_u, ibm-1167) KOI8-RU (ibm-1168, koi8_ru) Johab (x-johab) JISO208 () JISO201 () ISO-8859-6S (iso8859-6S, iso8859_6S) ISO-8859-16 (8859-16, iso8859_16, iso8859-16) ISO-8859-14 (ISO_8859-14:1998, 8859-14, latin8, iso-ir-199, iso8859-14, l8, isoceltic, iso8859_14) ISO-8859-10 (latin6, 8859-10, ISO_8859-10:1992, iso8859_10, iso-ir-157, ibm-919, iso8859-10, l6, csisolatin6) IBM-971 (Cp971, ibm971) IBM-964 (ibm-euctw, Cp964) IBM-954C (Cp954c) IBM-954 (ibm954, Cp954) IBM-951 (Cp951, ibm951) IBM-949C (Cp949C, ibm949C) IBM-947 (Cp947, ibm947) IBM-943C (ibm943C, Cp943C) IBM-942C (Cp942C, ibm942C) IBM-939 (Cp5035, 5035) IBM-932 (ibm932, Cp932) IBM-930 (Cp5026, 5026) IBM-927 (ibm927, Cp927) IBM-924 (Cp924, ibm924) IBM-918 (ibm918, Cp918) IBM-897 (Cp897, ibm897) IBM-867 (Cp867, ibm867) IBM-1380 (Cp1380, ibm1380) IBM-1371 (Cp1371, ibm1371) IBM-1370 (Cp1370, ibm1370) IBM-1364 (Cp1364, ibm1364) IBM-1363C (ibm1363C, Cp1363C) IBM-1047 (Cp1047, ibm1047) IBM-1088 (Cp1088, ibm1088) IBM-1382 (ibm1382, Cp1382) IBM-1385 (Cp1385, ibm1385) IBM-1386 (ibm1386, Cp1386) IBM-1388 (Cp1388, ibm1388) IBM-1390 (Cp1390, ibm1390) IBM-1399 (ibm1399, Cp1399) IBM-290 (ibm290, Cp290) IBM-300 (Cp300, ibm300) IBM-301 (Cp301, ibm301) IBM-33722 (5050, Cp5050) X-UnicodeLittle (UnicodeLittle) X-UnicodeBig (UnicodeBig) IBM-864S (ibm864S, Cp864S) IBM-859 (Cp859, ibm859) IBM-858 (Cp858, ibm858)

平台	受支持的编码（常见代码表中未收录）
AIX	IBM-837 (ibm837, Cp837) IBM-836 (ibm836, Cp836) IBM-835 (ibm835, Cp835) IBM-833 (ibm833, Cp833) IBM-808 (Cp808, ibm808) IBM-720 (Cp720, ibm720) IBM-420S (Cp420S, ibm420S) IBM-33722C (ibm-eucjp, Cp33722c) IBM-1046S (ibm1046S, Cp1046S) IBM-1043 (Cp1043, ibm1043) IBM-1041 (Cp1041, ibm1041) IBM-1027 (Cp1027, ibm1027) CESU-8 (CESU8) COMPOUND_TEXT (x-compound-text, x11-compound-text) GB2312 (gb2312-1980, gb2312-80) GBK (GBK) hp-roman8 (roman8, ibm-1051, r8, Cp1051) IBM-1114 (Cp1114, ibm1114) IBM-1115 (Cp1115, ibm1115) IBM-1140 (ibm1140, Cp1140) IBM-1141 (Cp1141, ibm1141) IBM-1142 (Cp1142, ibm1142) IBM-1143 (Cp1143, ibm1143) IBM-1144 (ibm1144, Cp1144) IBM-1145 (Cp1145, ibm1145) IBM-1146 (Cp1146, ibm1146) IBM-1147 (Cp1147, ibm1147) IBM-1148 (ibm1148, Cp1148) IBM-1149 (Cp1149, ibm1149) IBM-1351 (Cp1351, ibm1351) IBM-1362 (Cp1362, ibm1362) IBM-1363 (ibm1363, Cp1363)

平台	受支持的编码（常见代码表中未收录）
Microsoft Windows	windows-1256S (Cp1256s, ibm-1256s) UTF-8J (UTF8J) UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) PTCP154 (PT154, IBM-1169, Cyrillic-Asian, csPTCP154) KOI8-RU (ibm-1168, koi8_ru) ISO-8859-16 (8859-16, iso8859_16, iso8859-16) ISO-8859-14 (ISO_8859-14:1998, 8859-14, latin8, iso-ir-199, iso8859-14, l8, isoceltic, iso8859_14) IBM01141 (cp1141, ccsid01141, cp01141, 1141) IBM01142 (cp01142, cp1142, 1142, ccsid01142) IBM01143 (cp01143, 1143, ccsid01143, cp1143) IBM01144 (cp01144, cp1144, ccsid01144, 1144) IBM01145 (cp1145, cp01145, ccsid01145, 1145) IBM01146 (ccsid01146, cp01146, cp1146, 1146) IBM01147 (ccsid01147, cp1147, 1147, cp01147) IBM01148 (cp1148, ccsid01148, 1148, cp01148) IBM01149 (cp1149, cp01149, ccsid01149, 1149) IBM1047 (cp1047, 1047, ibm-1047) ISO-2022-JP-2 (csISO2022JP2, iso2022jp2) x-Big5-Solaris (Big5_Solaris) x-eucJP-Open (EUC_JP_Solaris, eucJP-open) x-IBM33722 (ibm33722, 33722, ibm-33722_vascii_vpua, ibm-5050, ibm-33722, cp33722) x-IBM930 (cp930, ibm930, ibm-930, 930) x-IBM939 (ibm-939, ibm939, cp939, 939) x-IBM964 (964, cp964, ibm-964, ibm964) x-ISO-2022-CN-CNS (ISO-2022-CN-CNS, ISO2022CN_CNS) x-iso-8859-11 (iso-8859-11, iso8859_11) x-JISAutoDetect (JISAutoDetect) x-MS932_0213 () x-MS950-HKSCS (MS950_HKSCS) x-PCK (pck) x-IBM1363C (ibm1363c, cp1363c, ibm-1363c) x-IBM420S (420s, ibm-420s, csibm420s, ibm420s, cp420s) x-IBM864S (csibm864s, ibm864s, cp864s, 864s, ibm-864s) x-IBM943C (cp943c, 943c, ibm-943c, ibm943c) x-IBM949C (ibm949c, cp949c, 949c, ibm-949c) x-IBM954C (cp954c, 954c, ibm-954c, ibm954c) x-ISO-8859-6S (8859_6s, iso8859-6s, iso8859_6s, iso-8859-6s) x-JIS0208 (jis_c6226-1983, jis_x0208-1983, csiso87jisx0208, x0208, iso-ir-87, jis0208) x-KSC5601 (ksc5601) x-MacDingbat (macdingbat) x-MacHebrew (machebrew) x-MacThai (macthai) x-MacUkraine (macukraine) x-IBM1046S (ibm-1046s, 1046s, cp1046s, ibm1046s) x-IBM-udcJP (IBM-udcJP) JIS_X0201 (jis_x0201, x0201, cshalfwidthkatakana, jis0201) IBM-939A (Cp939A, ibm939A) IBM-930A (ibm930A, Cp930A) IBM-33722A (Cp33722A, ibm33722A) x-windows-iso2022jp (windows-iso2022jp) x-windows-50221 (ms50221, cp50221) x-windows-50220 (cp50220, ms50220) X-UTF-32LE-BOM (UTF_32LE_BOM, UTF-32LE-BOM) X-UTF-32BE-BOM (UTF_32BE_BOM, UTF-32BE-BOM) x-SJIS_0213 () IBM01140 (ccsid01140, cp01140, 1140, cp1140) IBM00858 (cp858, ccsid00858, 858, cp00858) X-UnicodeLittle (UnicodeLittle)

平台	受支持的编码（常见代码表中未收录）
Microsoft Windows	IBM-836 (ibm836, Cp836) IBM-835 (ibm835, Cp835) IBM-833 (ibm833, Cp833) IBM-808 (Cp808, ibm808) IBM-720 (Cp720, ibm720) IBM-33722C (ibm-eucjp, Cp33722c) IBM-301 (Cp301, ibm301) IBM-300 (Cp300, ibm300) IBM-290 (ibm290, Cp290) IBM-1399 (ibm1399, Cp1399) IBM-1390 (Cp1390, ibm1390) IBM-1388 (Cp1388, ibm1388) IBM-1385 (Cp1385, ibm1385) IBM-1382 (ibm1382, Cp1382) IBM-1088 (Cp1088, ibm1088) IBM-1043 (Cp1043, ibm1043) IBM-1041 (Cp1041, ibm1041) IBM-1027 (Cp1027, ibm1027) CESU-8 (CESU8) COMPOUND_TEXT (x-compound-text, x11-compound-text) GB2312 (gb2312-1980, gb2312-80) GBK (GBK) hp-roman8 (roman8, ibm-1051, r8, Cp1051) IBM-1115 (Cp1115, ibm1115) IBM-1351 (Cp1351, ibm1351) IBM-1362 (Cp1362, ibm1362) IBM-1363 (ibm1363, Cp1363) IBM-1364 (Cp1364, ibm1364) IBM-1370 (Cp1370, ibm1370) IBM-1371 (Cp1371, ibm1371) IBM-1380 (Cp1380, ibm1380) IBM-867 (Cp867, ibm867) IBM-897 (Cp897, ibm897) IBM-924 (Cp924, ibm924) IBM-927 (ibm927, Cp927) IBM-932 (ibm932, Cp932) IBM-947 (Cp947, ibm947) IBM-951 (Cp951, ibm951) IBM-954 (ibm954, Cp954) IBM-971 (Cp971, ibm971) ISO-8859-10 (latin6, 8859-10, ISO_8859-10:1992, iso8859_10, iso-ir-157, ibm-919, iso8859-10, l6, csisolatin6)

平台	受支持的编码（常见代码表中未收录）
System x 上的 Red Hat Enterprise Linux	windows-1256S (Cp1256s, ibm-1256s) UTF-8J (UTF8J) UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) UTF-32 (UCS-4, UTF32, ISO-10646-UCS-4) PTCP154 (PT154, IBM-1169, Cyrillic-Asian, csPTCP154) KOI8-RU (ibm-1168, koi8_ru) ISO-8859-16 (8859-16, iso8859_16, iso8859-16) ISO-8859-14 (ISO_8859-14:1998, 8859-14, latin8, iso-ir-199, iso8859-14, l8, isoceltic, iso8859_14) IBM01141 (cp1141, ccsid01141, cp01141, 1141) IBM01142 (cp01142, cp1142, 1142, ccsid01142) IBM01143 (cp01143, 1143, ccsid01143, cp1143) IBM01144 (cp01144, cp1144, ccsid01144, 1144) IBM01145 (cp1145, cp01145, ccsid01145, 1145) IBM01146 (ccsid01146, cp01146, cp1146, 1146) IBM01147 (ccsid01147, cp1147, 1147, cp01147) IBM01148 (cp1148, ccsid01148, 1148, cp01148) IBM01149 (cp1149, cp01149, ccsid01149, 1149) IBM1047 (cp1047, 1047, ibm-1047) IBM918 (cp918, ebcdic-cp-ar2, ibm-918, 918) ISO-2022-JP-2 (csISO2022JP2, iso2022jp2) x-Big5-Solaris (Big5_Solaris) x-eucJP-Open (EUC_JP_Solaris, eucJP-open) x-IBM33722 (ibm33722, 33722, ibm-33722_vascii_vpua, ibm-5050, ibm-33722, cp33722) x-IBM930 (cp930, ibm930, ibm-930, 930) x-IBM939 (ibm-939, ibm939, cp939, 939) x-IBM964 (964, cp964, ibm-964, ibm964) x-ISO-2022-CN-CNS (ISO-2022-CN-CNS, ISO2022CN_CNS) x-iso-8859-11 (iso-8859-11, iso8859_11) x-JISAutoDetect (JISAutoDetect) x-MS932_0213 () x-MS950-HKSCS (MS950_HKSCS) x-PCK (pck) x-IBM1363C (ibm1363c, cp1363c, ibm-1363c) x-IBM420S (420s, ibm-420s, csibm420s, ibm420s, cp420s) x-IBM864S (csibm864s, ibm864s, cp864s, 864s, ibm-864s) x-IBM943C (cp943c, 943c, ibm-943c, ibm943c) x-IBM949C (ibm949c, cp949c, 949c, ibm-949c) x-IBM954C (cp954c, 954c, ibm-954c, ibm954c) x-ISO-8859-6S (8859_6s, iso8859-6s, iso8859_6s, iso-8859-6s) x-JIS0208 (jis_c6226-1983, jis_x0208-1983, csiso87jisx0208, x0208, iso-ir-87, jis0208) x-KSC5601 (ksc5601) x-MacDingbat (macdingbat) x-MacHebrew (machebrew) x-MacThai (macthai) x-MacUkraine (macukraine) x-IBM1046S (ibm-1046s, 1046s, cp1046s, ibm1046s) x-IBM-udcJP (IBM-udcJP) JIS_X0201 (jis_x0201, x0201, cshalfwidthkatakana, jis0201) IBM-939A (Cp939A, ibm939A) IBM-930A (ibm930A, Cp930A) IBM-33722A (Cp33722A, ibm33722A) x-windows-iso2022jp (windows-iso2022jp) x-windows-50221 (ms50221, cp50221) x-windows-50220 (cp50220, ms50220) X-UTF-32LE-BOM (UTF_32LE_BOM, UTF-32LE-BOM) X-UTF-32BE-BOM (UTF_32BE_BOM, UTF-32BE-BOM) x-SJIS_0213 () IBM01140 (ccsid01140, cp01140, 1140, cp1140) IBM00858 (cp858,

平台	受支持的编码（常见代码表中未收录）
System x 上的 Red Hat Enterprise Linux	IBM-837 (ibm837, Cp837) IBM-836 (ibm836, Cp836) IBM-835 (ibm835, Cp835) IBM-833 (ibm833, Cp833) IBM-808 (Cp808, ibm808) IBM-720 (Cp720, ibm720) IBM-33722C (ibm-eucjp, Cp33722c) IBM-301 (Cp301, ibm301) IBM-300 (Cp300, ibm300) IBM-290 (ibm290, Cp290) IBM-1399 (ibm1399, Cp1399) IBM-1390 (Cp1390, ibm1390) IBM-1388 (Cp1388, ibm1388) IBM-1385 (Cp1385, ibm1385) IBM-1382 (ibm1382, Cp1382) IBM-1088 (Cp1088, ibm1088) IBM-1043 (Cp1043, ibm1043) IBM-1041 (Cp1041, ibm1041) IBM-1027 (Cp1027, ibm1027) CESU-8 (CESU8) COMPOUND_TEXT (x-compound-text, x11-compound-text) GB2312 (gb2312-1980, gb2312-80) GBK (GBK) hp-roman8 (roman8, ibm-1051, r8, Cp1051) IBM-1114 (Cp1114, ibm1114) IBM-1115 (Cp1115, ibm1115) IBM-1351 (Cp1351, ibm1351) IBM-1362 (Cp1362, ibm1362) IBM-1363 (ibm1363, Cp1363) IBM-1364 (Cp1364, ibm1364) IBM-1370 (Cp1370, ibm1370) IBM-1371 (Cp1371, ibm1371) IBM-1380 (Cp1380, ibm1380) IBM-867 (Cp867, ibm867) IBM-897 (Cp897, ibm897) IBM-924 (Cp924, ibm924) IBM-927 (ibm927, Cp927) IBM-932 (ibm932, Cp932) IBM-947 (Cp947, ibm947) IBM-951 (Cp951, ibm951) IBM-954 (ibm954, Cp954) IBM-971 (Cp971, ibm971) ISO-8859-10 (latin6, 8859-10, ISO_8859-10:1992, iso8859_10, iso-ir-157, ibm-919, iso8859-10, l6, csisolatin6)

平台	受支持的编码（常见代码表中未收录）
z/OS	windows-1256S (Cp1256s, ibm-1256s) UTF-8J (UTF8J) UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) UTF-32 (UCS-4, UTF32, ISO-10646-UCS-4) UTF-16 (UTF16, Unicode, UTF_16, UCS-2) Shift_JIS () PTCP154 (PT154, IBM-1169, Cyrillic-Asian, csPTCP154) MacUkraine () MacThai () MacHebrew () MacDingbat () KSC5601 () KOI8-U (koi8_u, ibm-1167) KOI8-RU (ibm-1168, koi8_ru) Johab (x-johab) JISO208 () JISO201 () ISO-8859-6S (iso8859-6S, iso8859_6S) ISO-8859-16 (8859-16, iso8859_16, iso8859-16) ISO-8859-14 (ISO_8859-14:1998, 8859-14, latin8, iso-ir-199, iso8859-14, l8, isoceltic, iso8859_14) ISO-8859-10 (latin6, 8859-10, ISO_8859-10:1992, iso8859_10, iso-ir-157, ibm-919, iso8859-10, l6, csisolatin6) IBM-971 (Cp971, ibm971) IBM-964 (ibm-euctw, Cp964) IBM-954C (Cp954c) IBM-954 (ibm954, Cp954) IBM-951 (Cp951, ibm951) IBM-949C (Cp949C, ibm949C) IBM-947 (Cp947, ibm947) IBM-943C (ibm943C, Cp943C) IBM-942C (Cp942C, ibm942C) IBM-939 (Cp5035, 5035) IBM-932 (ibm932, Cp932) IBM-930 (Cp5026, 5026) IBM-927 (ibm927, Cp927) IBM-924 (Cp924, ibm924) IBM-918 (ibm918, Cp918) IBM-897 (Cp897, ibm897) IBM-867 (Cp867, ibm867) IBM-1380 (Cp1380, ibm1380) IBM-1371 (Cp1371, ibm1371) IBM-1370 (Cp1370, ibm1370) IBM-1364 (Cp1364, ibm1364) IBM-1363C (ibm1363C, Cp1363C) IBM-1363 (ibm1363, Cp1363) IBM-1088 (Cp1088, ibm1088) IBM-1382 (ibm1382, Cp1382) IBM-1385 (Cp1385, ibm1385) IBM-1386 (ibm1386, Cp1386) IBM-1388 (Cp1388, ibm1388) IBM-1390 (Cp1390, ibm1390) IBM-1399 (ibm1399, Cp1399) IBM-290 (ibm290, Cp290) IBM-300 (Cp300, ibm300) IBM-301 (Cp301, ibm301) IBM-33722 (5050, Cp5050) IBM-33722C (ibm-eucjp, Cp33722c) IBM-930A (ibm930A, Cp930A) X-UnicodeLittle (UnicodeLittle) X-UnicodeBig (UnicodeBig) IBM-864S (ibm864S, Cp864S) IBM-859 (Cp859, ibm859) IBM-858 (Cp858, ibm858)

平台	受支持的编码（常见代码表中未收录）
z/OS	IBM-837 (ibm837, Cp837) IBM-836 (ibm836, Cp836) IBM-835 (ibm835, Cp835) IBM-833 (ibm833, Cp833) IBM-808 (Cp808, ibm808) IBM-720 (Cp720, ibm720) IBM-420S (Cp420S, ibm420S) IBM-1047 (Cp1047, ibm1047) IBM-1046S (ibm1046S, Cp1046S) IBM-1043 (Cp1043, ibm1043) IBM-1041 (Cp1041, ibm1041) IBM-1027 (Cp1027, ibm1027) CESU-8 (CESU8) COMPOUND_TEXT (x-compound-text, x11-compound-text) GB2312 (gb2312-1980, gb2312-80) GBK (GBK) hp-roman8 (roman8, ibm-1051, r8, Cp1051) IBM-1114 (Cp1114, ibm1114) IBM-1115 (Cp1115, ibm1115) IBM-1140 (ibm1140, Cp1140) IBM-1141 (Cp1141, ibm1141) IBM-1142 (Cp1142, ibm1142) IBM-1143 (Cp1143, ibm1143) IBM-1144 (ibm1144, Cp1144) IBM-1145 (Cp1145, ibm1145) IBM-1146 (Cp1146, ibm1146) IBM-1147 (Cp1147, ibm1147) IBM-1148 (ibm1148, Cp1148) IBM-1149 (Cp1149, ibm1149) IBM-1351 (Cp1351, ibm1351) IBM-1362 (Cp1362, ibm1362)

平台	受支持的编码（常见代码表中未收录）
Linux on POWER Systems - Big Endian	windows-1256S (Cp1256s, ibm-1256s) UTF-8J (UTF8J) UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) UTF-32 (UCS-4, UTF32, ISO-10646-UCS-4) UTF-16 (UTF16, Unicode, UTF_16, UCS-2) Shift_JIS () PTCP154 (PT154, IBM-1169, Cyrillic-Asian, csPTCP154) MacUkraine () MacThai () MacHebrew () MacDingbat () KSC5601 () KOI8-U (koi8_u, ibm-1167) KOI8-RU (ibm-1168, koi8_ru) Johab (x-johab) JISO208 () JISO201 () ISO-8859-6S (iso8859-6S, iso8859_6S) ISO-8859-16 (8859-16, iso8859_16, iso8859-16) ISO-8859-14 (ISO_8859-14:1998, 8859-14, latin8, iso-ir-199, iso8859-14, l8, isoceltic, iso8859_14) ISO-8859-10 (latin6, 8859-10, ISO_8859-10:1992, iso8859_10, iso-ir-157, ibm-919, iso8859-10, l6, csisolatin6) IBM-971 (Cp971, ibm971) IBM-964 (ibm-euctw, Cp964) IBM-954C (Cp954c) IBM-954 (ibm954, Cp954) IBM-951 (Cp951, ibm951) IBM-949C (Cp949C, ibm949C) IBM-947 (Cp947, ibm947) IBM-943C (ibm943C, Cp943C) IBM-942C (Cp942C, ibm942C) IBM-939 (Cp5035, 5035) IBM-932 (ibm932, Cp932) IBM-930 (Cp5026, 5026) IBM-927 (ibm927, Cp927) IBM-924 (Cp924, ibm924) IBM-918 (ibm918, Cp918) IBM-897 (Cp897, ibm897) IBM-867 (Cp867, ibm867) IBM-1380 (Cp1380, ibm1380) IBM-1371 (Cp1371, ibm1371) IBM-1370 (Cp1370, ibm1370) IBM-1364 (Cp1364, ibm1364) IBM-1363C (ibm1363C, Cp1363C) IBM-1047 (Cp1047, ibm1047) IBM-1088 (Cp1088, ibm1088) IBM-1382 (ibm1382, Cp1382) IBM-1385 (Cp1385, ibm1385) IBM-1386 (ibm1386, Cp1386) IBM-1388 (Cp1388, ibm1388) IBM-1390 (Cp1390, ibm1390) IBM-1399 (ibm1399, Cp1399) IBM-290 (ibm290, Cp290) IBM-300 (Cp300, ibm300) IBM-301 (Cp301, ibm301) IBM-33722 (5050, Cp5050) X-UnicodeLittle (UnicodeLittle) X-UnicodeBig (UnicodeBig) IBM-864S (ibm864S, Cp864S) IBM-859 (Cp859, ibm859) IBM-858 (Cp858, ibm858)

平台	受支持的编码（常见代码表中未收录）
Linux on POWER Systems - Big Endian	IBM-837 (ibm837, Cp837) IBM-836 (ibm836, Cp836) IBM-835 (ibm835, Cp835) IBM-833 (ibm833, Cp833) IBM-808 (Cp808, ibm808) IBM-720 (Cp720, ibm720) IBM-420S (Cp420S, ibm420S) IBM-33722C (ibm-eucjp, Cp33722c) IBM-1046S (ibm1046S, Cp1046S) IBM-1043 (Cp1043, ibm1043) IBM-1041 (Cp1041, ibm1041) IBM-1027 (Cp1027, ibm1027) CESU-8 (CESU8) COMPOUND_TEXT (x-compound-text, x11-compound-text) GB2312 (gb2312-1980, gb2312-80) GBK (GBK) hp-roman8 (roman8, ibm-1051, r8, Cp1051) IBM-1114 (Cp1114, ibm1114) IBM-1115 (Cp1115, ibm1115) IBM-1140 (ibm1140, Cp1140) IBM-1141 (Cp1141, ibm1141) IBM-1142 (Cp1142, ibm1142) IBM-1143 (Cp1143, ibm1143) IBM-1144 (ibm1144, Cp1144) IBM-1145 (Cp1145, ibm1145) IBM-1146 (Cp1146, ibm1146) IBM-1147 (Cp1147, ibm1147) IBM-1148 (ibm1148, Cp1148) IBM-1149 (Cp1149, ibm1149) IBM-1351 (Cp1351, ibm1351) IBM-1362 (Cp1362, ibm1362) IBM-1363 (ibm1363, Cp1363)

平台	受支持的编码（常见代码表中未收录）
HP (PA-RISC)	UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) IBM01147 (ccsid01147, cp1147, 1147, cp01147) IBM01148 (cp1148, ccsid01148, 1148, cp01148) IBM01149 (cp1149, cp01149, ccsid01149, 1149) IBM1047 (cp1047, 1047, ibm-1047) IBM918 (cp918, ebcdic-cp-ar2, ibm-918, 918) ISO-2022-JP-2 (csISO2022JP2, iso2022jp2) Roman9 (Roman9) x-Big5-Solaris (Big5_Solaris) x-eucJP-Open (EUC_JP_Solaris, eucJP-open) x-IBM33722 (ibm33722, 33722, ibm-33722_vascii_vpua, ibm-5050, ibm-33722, cp33722) x-IBM930 (cp930, ibm930, ibm-930, 930) x-IBM939 (ibm-939, ibm939, cp939, 939) x-windows-iso2022jp (windows-iso2022jp) x-windows-50221 (ms50221, cp50221) x-windows-50220 (cp50220, ms50220) X-UTF-32LE-BOM (UTF_32LE_BOM, UTF-32LE-BOM) X-UTF-32BE-BOM (UTF_32BE_BOM, UTF-32BE-BOM) x-SJIS_0213 () x-PCK (pck) x-MS950-HKSCS (MS950_HKSCS) x-MS932_0213 () x-JISAutoDetect (JISAutoDetect) x-iso-8859-11 (iso-8859-11, iso8859_11) x-ISO-2022-CN-CNS (ISO-2022-CN-CNS, ISO2022CN_CNS) x-IBM964 (964, cp964, ibm-964, ibm964) IBM01146 (ccsid01146, cp01146, cp1146, 1146) IBM01145 (cp1145, cp01145, ccsid01145, 1145) IBM01144 (cp01144, cp1144, ccsid01144, 1144) IBM01143 (cp01143, 1143, ccsid01143, cp1143) IBM01142 (cp01142, cp1142, 1142, ccsid01142) IBM01141 (cp1141, ccsid01141, cp01141, 1141) IBM01140 (ccsid01140, cp01140, 1140, cp1140) IBM00858 (cp858, ccsid00858, 858, cp00858) X-UnicodeLittle (UnicodeLittle) X-UnicodeBig (UnicodeBig) COMPOUND_TEXT (x-compound-text, x11-compound-text) hp-roman8 (roman8, ibm-1051, r8, Cp1051) IBM-1364 (Cp1364, ibm1364) IBM-942C (Cp942C, ibm942C) IBM-943C (ibm943C, Cp943C) IBM-949C (Cp949C, ibm949C) JIS0201 () JIS0208 () KOI8-U (koi8_u, ibm-1167) MacDingbat () MacHebrew () MacThai () MacUkraine () UTF-32 (UCS-4, UTF32, ISO-10646-UCS-4)

平台（按编码）

编码	别名	对此编码提供支持的平台
x-MacUkraine	macukraine	Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-MacThai	mac thai	Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-MacHebrew	machebrew	Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-MacDingbat	macdingbat	Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-KSC5601	ksc5601	Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-JIS0208	jis_c6226-1983, jis_x0208-1983, csiso87jisx0208, x0208, iso-ir-87, jis0208	Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-ISO-8859-6S	8859_6s, iso8859-6s, iso8859_6s, iso-8859-6s	SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-IBM954C	cp954c, 954c, ibm-954c, ibm954c	SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-IBM949C	ibm949c, cp949c, 949c, ibm-949c	Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux

编码	别名	对此编码提供支持的平台
x-IBM943C	cp943c, 943c, ibm-943c, ibm943c	Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-IBM864S	csibm864s, ibm864s, cp864s, 864s, ibm-864s	SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-IBM420S	420s, ibm-420s, csibm420s, ibm420s, cp420s	SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-IBM1363C	ibm1363c, cp1363c, ibm-1363c	SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-IBM1046S	ibm-1046s, 1046s, cp1046s, ibm1046s	SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
x-IBM-udcJP	IBM-udcJP	SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
JIS_X0201	jis_x0201, x0201, cshalfwidthkatakana, jis0201	Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
IBM-939A	Cp939A, ibm939A	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
IBM-930A	ibm930A, Cp930A	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS
IBM-924_LF	Cp924_LF, ibm924_LF	IBM i

编码	别名	对此编码提供支持的平台
IBM-33722A	Cp33722A, ibm33722A	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux
IBM-1141_LF	Cp1141_LF, ibm1141_LF	IBM i
IBM-1047_LF	Cp1047_LF, ibm1047_LF	IBM i
x-windows-iso2022jp	windows-iso2022jp	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
x-windows-50221	ms50221, cp50221	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
x-windows-50220	cp50220, ms50220	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
X-UTF-32LE-BOM	UTF_32LE_BOM, UTF-32LE-BOM	Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
X-UTF-32BE-BOM	UTF_32BE_BOM, UTF-32BE-BOM	Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)

编码	别名	对此编码提供支持的平台
x-SJIS_0213		Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
x-PCK	pck	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
x-MS950-HKSCS	MS950_HKSCS	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
x-MS932_0213		Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
x-JISAutoDetect	JISAutoDetect	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
x-iso-8859-11	iso-8859-11, iso8859_11	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)

编码	别名	对此编码提供支持的平台
x-ISO-2022-CN-CNS	ISO-2022-CN-CNS, ISO2022CN_CNS	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统 Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
x-IBM964	964, cp964, ibm-964, ibm964	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
x-IBM939	ibm-939, ibm939, cp939, 939	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
x-IBM930	cp930, ibm930, ibm-930, 930	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
x-IBM33722	ibm33722, 33722, ibm-33722_vascii_vpua, ibm-5050, ibm-33722, cp33722	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
x-eucJP-Open	EUC_JP_Solaris, eucJP-open	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)

编码	别名	对此编码提供支持的平台
x-Big5-Solaris	Big5_Solaris	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
Roman9	Roman9	HP (PA-RISC)
ISO-2022-JP-2	csISO2022JP2, iso2022jp2	Solaris, SUSE Linux Enterprise Server on System x Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
IBM918	cp918, ebcdic-cp-ar2, ibm-918, 918	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
IBM1047	cp1047, 1047, ibm-1047	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
IBM01149	cp1149, cp01149, ccsid01149, 1149	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
IBM01148	cp1148, ccsid01148, 1148, cp01148	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)

编码	别名	对此编码提供支持的平台
IBM01147	ccsid01147, cp1147, 1147, cp01147	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
IBM01146	ccsid01146, cp01146, cp1146, 1146	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
IBM01145	cp1145, cp01145, ccsid01145, 1145	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
IBM01144	cp01144, cp1144, ccsid01144, 1144	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
IBM01143	cp01143, 1143, ccsid01143, cp1143	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
IBM01142	cp01142, cp1142, 1142, ccsid01142	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)

编码	别名	对此编码提供支持的平台
IBM01141	cp1141, ccsid01141, cp01141, 1141	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
IBM01140	ccsid01140, cp01140, 1140, cp1140	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
IBM00858	cp858, ccsid00858, 858, cp00858	Solaris, SUSE Linux Enterprise Server on System x HP-UX (Itanium), Linux for IBM Z 系统, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , HP (PA-RISC)
X-UnicodeLittle	UnicodeLittle	Solaris, SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems-大尾数法。 HP (PA-RISC)
X-UnicodeBig	UnicodeBig	Solaris, SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems-大尾数法。 HP (PA-RISC)

编码	别名	对此编码提供支持的平台
IBM-864S	ibm864S, Cp864S	IBM i, AIX, z/OS. Linux on POWER Systems - Big Endian
IBM-859	Cp859, ibm859	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-858	Cp858, ibm858	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-837	ibm837, Cp837	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-836	ibm836, Cp836	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-835	ibm835, Cp835	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian

编码	别名	对此编码提供支持的平台
IBM-833	ibm833, Cp833	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-808	Cp808, ibm808	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-720	Cp720, ibm720	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-420S	Cp420S, ibm420S	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-33722C	ibm-eucjp, Cp33722c	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-33722	5050, Cp5050	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian

编码	别名	对此编码提供支持的平台
IBM-301	Cp301, ibm301	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-300	Cp300, ibm300	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-290	ibm290, Cp290	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1399	ibm1399, Cp1399	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1390	Cp1390, ibm1390	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian

编码	别名	对此编码提供支持的平台
IBM-1388	Cp1388, ibm1388	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1386	ibm1386, Cp1386	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-1385	Cp1385, ibm1385	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1382	ibm1382, Cp1382	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1088	Cp1088, ibm1088	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1047	Cp1047, ibm1047	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-1046S	ibm1046S, Cp1046S	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian

编码	别名	对此编码提供支持的平台
IBM-1043	Cp1043, ibm1043	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1041	Cp1041, ibm1041	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1027	Cp1027, ibm1027	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
CESU-8	CESU8	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
COMPOUND_TEXT	x-compound-text, x11-compound-text	Solaris, SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)

编码	别名	对此编码提供支持的平台
GB2312	gb2312-1980, gb2312-80	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
GBK	GBK	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
hp-roman8	roman8, ibm-1051, r8, Cp1051	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
IBM-1114	Cp1114, ibm1114	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1115	Cp1115, ibm1115	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian

编码	别名	对此编码提供支持的平台
IBM-1140	ibm1140, Cp1140	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-1141	Cp1141, ibm1141	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-1142	Cp1142, ibm1142	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-1143	Cp1143, ibm1143	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-1144	ibm1144, Cp1144	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-1145	Cp1145, ibm1145	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-1146	Cp1146, ibm1146	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-1147	Cp1147, ibm1147	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-1148	ibm1148, Cp1148	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-1149	Cp1149, ibm1149	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian

编码	别名	对此编码提供支持的平台
IBM-1351	Cp1351, ibm1351	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1362	Cp1362, ibm1362	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1363	ibm1363, Cp1363	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1363C	ibm1363C, Cp1363C	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-1364	Cp1364, ibm1364	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)

编码	别名	对此编码提供支持的平台
IBM-1370	Cp1370, ibm1370	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1371	Cp1371, ibm1371	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-1380	Cp1380, ibm1380	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-867	Cp867, ibm867	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-897	Cp897, ibm897	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-918	ibm918, Cp918	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian

编码	别名	对此编码提供支持的平台
IBM-924	Cp924, ibm924	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-927	ibm927, Cp927	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-930	Cp5026, 5026	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-932	ibm932, Cp932	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-939	Cp5035, 5035	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-942C	Cp942C, ibm942C	Solaris, IBM i, HP-UX (Itanium), AIX, z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)

编码	别名	对此编码提供支持的平台
IBM-943C	ibm943C, Cp943C	IBM i, HP-UX (Itanium), AIX, z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
IBM-947	Cp947, ibm947	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-949C	Cp949C, ibm949C	IBM i, HP-UX (Itanium), AIX, z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
IBM-951	Cp951, ibm951	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-954	ibm954, Cp954	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
IBM-954C	Cp954c	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
IBM-964	ibm-euctw, Cp964	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian

编码	别名	对此编码提供支持的平台
IBM-971	Cp971, ibm971	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
ISO-8859-10	latin6, 8859-10, ISO_8859-10:1992, iso8859_10, iso-ir-157, ibm-919, iso8859-10, l6, csisolatin6	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
ISO-8859-14	ISO_8859-14:1998, 8859-14, latin8, iso- ir-199, iso8859-14, l8, isoceltic, iso8859_14	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
ISO-8859-16	8859-16, iso8859_16, iso8859-16	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
ISO-8859-6S	iso8859-6S, iso8859_6S	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
JIS0201		IBM i, HP-UX (Itanium), AIX, z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)

编码	别名	对此编码提供支持的平台
JIS0208		IBM i, HP-UX (Itanium), AIX, z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
Johab	x-johab	IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
KOI8-RU	ibm-1168, koi8_ru	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
KOI8-U	koi8_u, ibm-1167	Solaris, IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
KSC5601		IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
MacDingbat		IBM i, HP-UX (Itanium), AIX, z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
MacHebrew		IBM i, HP-UX (Itanium), AIX, z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)

编码	别名	对此编码提供支持的平台
MacThai		IBM i, HP-UX (Itanium), AIX, z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
MacUkraine		IBM i, HP-UX (Itanium), AIX, z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
PTCP154	PT154, IBM-1169, Cyrillic-Asian, csPTCP154	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
Shift_JIS		IBM i, AIX, z/OS, Linux on POWER Systems - Big Endian
UTF-16	UTF16, Unicode, UTF_16, UCS-2	IBM i, HP-UX (Itanium), AIX, z/OS, Linux on POWER Systems - Big Endian
UTF-32	UCS-4, UTF32, ISO-10646-UCS-4	Solaris, SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)

编码	别名	对此编码提供支持的平台
UTF-32BE	UTF_32BE, X-UTF-32BE, UTF32BE	Solaris, SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
UTF-32LE	UTF_32LE, X-UTF-32LE, UTF32LE	Solaris, SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
UTF-8J	UTF8J	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian
windows-1256S	Cp1256s, ibm-1256s	SUSE Linux Enterprise Server on System x IBM i, Linux for IBM Z 系统, AIX, Microsoft Windows, System x 上的 Red Hat Enterprise Linux , z/OS, Linux on POWER Systems - Big Endian

相关概念

第 222 页的『使用传输定义文件』

您可以指定可用于创建文件传输的传输定义文件。传输定义文件是定义创建传输所需的部分或全部信息的 XML 文件。

相关参考

第 713 页的『传输文本文件』

文本文件传输包括将文件代码页从一个代码页转换为另一个。文本文件传输还包括在系统之间转换 CRLF（回车-换行符）字符。本主题概述 IBM MQ Managed File Transfer 的文本文件传输行为。

第 497 页的『fteCreateTransfer（创建新的文件传输）』

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

IBM MQ Managed File Transfer 的消息格式

IBM MQ Managed File Transfer 出于多种目的使用 XML 格式的消息：与 Web 网关交互；向代理发出命令；记录有关监视器、计划安排和传输的信息；以及定义用于配置的信息。用于这些目的的 XML 格式的逻辑结构由 XML 模式描述。

每个版本的 IBM MQ Managed File Transfer 使用 XML 模式来验证以 XML 编写的消息。代理抽取 XML 模式版本，并确定该模式是否受支持。

安装 IBM MQ Managed File Transfer 后，可以在以下目录中找到 IBM MQ Managed File Transfer 消息模式文件：*MQ_INSTALLATION_PATH/mqft/samples/schema*。包含以下模式：

Web 网关使用的 XML 消息的模式

- Filespace.xsd
- FileSpaceInfo.xsd
- UserInfo.xsd
- WebFileSpaceList.xsd
- WebTransferStatus.xsd

要了解有关 Web 网关所用模式的更多信息，请参阅第 937 页的『[管理响应和请求格式](#)』和第 919 页的『[响应格式：XML 和 JSON](#)』。

可放置在代理命令队列上的 XML 消息的模式

- FileTransfer.xsd
- Internal.xsd
- Monitor.xsd
- PingAgent.xsd

要了解有关将 XML 消息放置在代理命令队列的更多信息，请参阅第 370 页的『[通过将消息放置在代理命令队列中来控制 IBM MQ Managed File Transfer](#)』。

发布到 **SYSTEM.FTE** 主题的 XML 消息的模式

- MonitorList.xsd
- MonitorLog.xsd
- ScheduleList.xsd
- ScheduleLog.xsd
- TransferLog.xsd
- TransferStatus.xsd

要了解有关发布到 SYSTEM.FTE 主题的 XML 消息以及 SYSTEM.FTE 主题的结构的信息，请参阅第 646 页的『[SYSTEM.FTE 主题](#)』。

IBM MQ Managed File Transfer 所用的其他模式

fteutils.xsd。此模式包含公共元素定义，并且由某些其他模式包含。

- Notification.xsd
- ProtocolBridgeCredentials.xsd
- ProtocolBridgeProperties.xsd
- ConnectDirectCredentials.xsd
- ConnectDirectNodeProperties.xsd
- ConnectDirectProcessDefinitions.xsd
- Reply.xsd
- UserSandboxes.xsd

相关参考

[第 647 页的『代理状态消息格式』](#)

在创建或启动代理时，该代理会向其协调队列管理器上的 SYSTEM.FTE 主题（在 SYSTEM.FTE/Agents/*agent name* 主题上）发布其详细信息。

[第 842 页的『文件传输请求消息格式』](#)

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

[第 658 页的『文件传输状态消息格式』](#)

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 SYSTEM.FTE/Transfers/*agent_name/transfer ID* 主题中），它符合 TransferStatus.xsd XML 模式。TransferStatus.xsd 文件位于 WMQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

[第 662 页的『文件传输日志消息格式』](#)

文件传输日志消息发布在主题字符串为 Log/*agent_name/transfer ID* 的 SYSTEM.FTE 主题内。这些消息符合模式 TransferLog.xsd，该模式位于 IBM MQ Managed File Transfer 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

[第 686 页的『已安排的传输日志消息格式』](#)

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/*agent name/schedule ID* 主题中）。该消息符合 ScheduleLog.xsd XML 模式。

[第 859 页的『监视器请求消息格式』](#)

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 fteCreateMonitor 命令或使用 WebSphere MQ Explorer 接口。

[第 871 页的『针对安全性的消息格式』](#)

本主题描述发布到协调队列管理器且与安全性相关的消息。

[第 607 页的『协议网桥凭证文件格式』](#)

代理配置目录中的 ProtocolBridgeCredentials.xml 文件定义用户名和凭证信息，供协议网桥代理用于在协议服务器中为自身授权。

[第 611 页的『协议网桥属性文件格式』](#)

代理配置目录中的 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性。

[第 618 页的『Connect:Direct 凭证文件格式』](#)

代理程序配置目录中的 ConnectDirectCredentials.xml 文件定义 Connect:Direct 代理程序用于向其自身授权 Connect:Direct 节点的用户名和凭证信息。

[第 621 页的『Connect:Direct 节点属性文件格式』](#)

Connect:Direct 网桥代理配置目录中的 ConnectDirectNodeProperties.xml 文件指定有关文件传输中涉及的远程 Connect:Direct 节点的信息。

[第 623 页的『Connect:Direct 进程定义文件格式』](#)

Connect:Direct 网桥代理配置目录中的 ConnectDirectProcessDefinitions.xml 文件指定要在文件传输过程中启动的用户定义的 Connect:Direct 进程。

[第 868 页的『Ping 代理请求消息格式』](#)

您可以通过发出 **ftePingAgent** 命令或将 XML 消息放置在代理命令队列中 ping 代理。ping 代理请求 XML 必须符合 PingAgent.xsd 模式。安装 IBM MQ Managed File Transfer 后，您可以在以下目录中找到 PingAgent.xsd 模式文件：MQ_INSTALLATION_PATH/mqft/samples/schema。PingAgent.xsd 模式导入相同目录中的 fteutils.xsd。

[第 870 页的『应答消息格式』](#)

代理在其代理命令队列中收到 XML 消息时，如果需要响应，那么代理会向原始消息中定义的应答队列发送一条 XML 应答消息。应答 XML 符合 Reply.xsd 模式。Reply.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。Reply.xsd 模式将导入位于相同目录中的 fteutils.xsd。

代理状态消息格式

在创建或启动代理时，该代理会向其协调队列管理器上的 SYSTEM.FTE 主题（在 SYSTEM.FTE/Agents/*agent name* 主题上）发布其详细信息。

包含以下信息：

- 代理程序名称
- 运行代理的平台
- 代理描述（如果提供）
- 代理队列管理器
- 代理运行的时区
- 代理版本
- 代理传输限制
- 代理的所有当前传输的状态。 [代理传输状态](#)中列出了这些状态
- 代理类型

如果代理是协议网桥代理，那么还将包含以下信息：

- 协议网桥代理类型
- 协议网桥服务器的主机名或 IP 地址

如果代理是 Web 代理，那么还将包含以下信息：

- Web 代理连接到的 Web 网关的名称

只要代理传输状态发生更改，就会重新发布代理状态，但缺省情况下不超过 30 秒。您可以使用 `agentStatusPublishRateLimit` 代理属性来更改该缺省设置，[高级代理属性](#)中描述了该属性。

以下示例的输出显示用于代理状态中的每个数据元素的键：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="agentOsName">Windows 7</entry>
  <entry key="agentDescription"/>
  <entry key="queueManager">QM1</entry>
  <entry key="agentTimeZone">Europe/London</entry>
  <entry key="agentVersion">1.00</entry>
  <entry key="agentName">FTEAGENT</entry>
  <entry key="maxDestinationTransfers">25</entry>
  <entry key="maxSourceTransfers">25</entry>
  <entry key="maxQueuedTransfers">100</entry>
  <entry
key="DestinationTransferStates">414d51204d554e474f202020202020d857374a60a72622=RunningTransfer
414d51204d554e474f202020202020d857374a69a72622=RunningTransfer
414d51204d554e474f202020202020d857374a75a72622=RunningTransfer
  </entry>
  <entry
key="SourceTransferStates">414d51204d554e474f202020202020d857374a93a72622=NegotiatingTransfer
414d51204d554e474f202020202020d857374a78a72622=RunningTransfer
414d51204d554e474f202020202020d857374aaba72622=NewSenderTransfer
414d51204d554e474f202020202020d857374a63a72622=RunningTransfer
  </entry>
</properties>
```

以下示例的输出显示用于协议网桥代理的代理状态中的每个数据元素的键：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="agentOsName">Windows 7</entry>
  <entry key="agentDescription"/>
  <entry key="queueManager">QM1</entry>
  <entry key="agentTimeZone">Europe/London</entry>
  <entry key="agentVersion">1.00</entry>
  <entry key="agentName">BRIDGE</entry>
  <entry key="protocolBridgeType">ftp</entry>
```

```

<entry key="protocolBridgeServerHost">ftpsrvr.example.org</entry>
<entry key="maxDestinationTransfers">25</entry>
<entry key="maxSourceTransfers">25</entry>
<entry key="maxQueuedTransfers">100</entry>
<entry key="DestinationTransferStates">414d51204d554e474f202020202020d857374a60a72622=RunningTransfer
</entry>
<entry key="SourceTransferStates">414d51204d554e474f202020202020d857374a93a72622=NegotiatingTransfer
</entry>
</properties>

```

相关参考

第 649 页的『代理传输状态』

启动的代理将其详细信息发布在其协调队列管理器上的 SYSTEM.FTE 主题中。这些详细信息中包括涉及该代理的每个当前传输的状态。包含的状态如下：

第 842 页的『文件传输请求消息格式』

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

第 658 页的『文件传输状态消息格式』

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 SYSTEM.FTE/Transfers/agent_name /transfer ID 主题中），它符合 TransferStatus.xsd XML 模式。TransferStatus.xsd 文件位于 WMQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

第 662 页的『文件传输日志消息格式』

文件传输日志消息发布在主题字符串为 Log/agent_name /transfer_ID 的 SYSTEM.FTE 主题内。这些消息符合模式 TransferLog.xsd，该模式位于 IBM MQ Managed File Transfer 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

第 686 页的『已安排的传输日志消息格式』

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/agent name /schedule ID 主题中）。该消息符合 ScheduleLog.xsd XML 模式。

第 859 页的『监视器请求消息格式』

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 fteCreateMonitor 命令或使用 WebSphere MQ Explorer 接口。

第 871 页的『针对安全性的消息格式』

本主题描述发布到协调队列管理器且与安全性相关的消息。

代理传输状态

启动的代理将其详细信息发布在其协调队列管理器上的 SYSTEM.FTE 主题中。这些详细信息中包括涉及该代理的每个当前传输的状态。包含的状态如下：

传输状态	说明
NewSenderTransfer	新的传输，来自未针对其启动协商的源代理。
NewReceiverTransfer	已在协商的过程中在目标代理处创建了新的传输，但该传输尚未运行。
NegotiatingTransfer	运行传输前，源代理会与目标代理协商。
RunningTransfer	来自源代理或目标代理且处于正常运行状态的传输
RecoveringTransfer	当源代理或目标代理启动恢复过程时，处于运行状态的所有传输都将转变为传输状态。当向同级代理发送再同步消息时，传输将由该状态进入 ReSynchronisingTransfer 状态。 例如，如果目标代理针对正在运行的传输启动恢复过程，那么在向其源代理发送再同步消息时，该传输将进入 ReSynchronisingTransfer 状态。

传输状态	说明
ReSynchronisingTransfer	传输源代理或目标代理发现了问题并向其各自的目标代理或源代理发送再同步消息。
CompletedTransfer	目标代理已完成传输并向源代理发送了完成消息。目标代理正在等待来自源代理的确认消息。
CompleteReceivedTransfer	源代理收到来自目标代理的完成消息，并向目标代理发送了回复消息以确认该完成消息。
CancelledNewTransfer	源代理收到有关取消新传输的消息。
CancelledInProgressTransfer	源代理收到有关取消正在进行的传输的消息。
ResumingTransfer	源代理收到再同步响应消息并且正在安排重新启动该传输。
RestartingTransfer	源代理或目标代理收到再同步请求消息并在等待各自的目标代理或源代理重新启动。
WaitingForDestinationCapacity	源代理从目标代理处收到 DESTINATION_CAPACITY_EXCEEDED 错误。传输现处于等待状态，以在某时间段之后重试。
FailedTransferEnding	传输已失败，但未发布完成日志消息，并且未将该传输从状态存储中移除。例如，如果代理进程在收到来自目标代理的失败响应之后但在后续处理完成之前停止，那么将会出现该状态。

相关参考

第 700 页的『代理状态值』

fteListAgents 和 **fteShowAgentDetails** 命令生成代理状态信息。该状态有多种可能的值。

第 647 页的『代理状态消息格式』

在创建或启动代理时，该代理会向其协调队列管理器上的 SYSTEM.FTE 主题（在 SYSTEM.FTE/Agents/*agent name* 主题上）发布其详细信息。

第 842 页的『文件传输请求消息格式』

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 *MQ_INSTALLATION_PATH/mqft/samples/schema* 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

第 658 页的『文件传输状态消息格式』

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 SYSTEM.FTE/Transfers/*agent_name*/*transfer ID* 主题中），它符合 TransferStatus.xsd XML 模式。TransferStatus.xsd 文件位于 WMQMFT 安装的 *MQ_INSTALLATION_PATH/mqft/samples/schema* 目录中。

第 662 页的『文件传输日志消息格式』

文件传输日志消息发布在主题字符串为 *Log/agent_name/transfer_ID* 的 SYSTEM.FTE 主题内。这些消息符合模式 TransferLog.xsd，该模式位于 IBM MQ Managed File Transfer 安装的 *MQ_INSTALLATION_PATH/mqft/samples/schema* 目录中。

第 686 页的『已安排的传输日志消息格式』

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/*agent name/schedule ID* 主题中）。该消息符合 ScheduleLog.xsd XML 模式。

第 859 页的『监视器请求消息格式』

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 fteCreateMonitor 命令或使用 WebSphere MQ Explorer 接口。

第 871 页的『针对安全性的消息格式』

本主题描述发布到协调队列管理器且与安全性相关的消息。

监视器列表消息格式

作为保留出版物发布至主题字符串 SYSTEM.FTE/monitors/agent_name/monitor_name 的 XML 消息符合 MonitorList.xsd 模式。 每条 XML 消息均列出属于此代理的活动监视器。 此信息由 **fteListMonitors** 命令和 WebSphere MQ Explorer 插件用于向用户显示监视器列表。 MonitorList.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。 MonitorList.xsd 模式导入位于同一目录中的 Monitor.xsd。

模式

以下模式描述了监视器列表 XML 消息中哪些元素有效。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema"
  targetNamespace="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
  xmlns="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition">

  <xsd:include schemaLocation="Monitor.xsd"/>

  <xsd:element name="monitorList">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="status" type="monitorStatusType" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="configuration" type="monitorConfigurationType" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="pollInterval" type="pollIntervalType" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="batch" type="batchType" minOccurs="1" maxOccurs="1"/>
        <xsd:any minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="agent" type="xsd:string" use="required"/>
      <xsd:attribute name="monitor" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="monitorStatusType">
    <xsd:sequence>
      <xsd:any minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="state" type="xsd:token"/>
    <xsd:anyAttribute/>
  </xsd:complexType>

  <xsd:complexType name="monitorConfigurationType">
    <xsd:sequence>
      <xsd:element name="description" type="xsd:string" minOccurs="1" maxOccurs="1" />
      <xsd:element name="resources" type="monitorResourcesType" minOccurs="0" maxOccurs="1" />
      <xsd:element name="triggerMatch" type="triggerMatchType" minOccurs="0" maxOccurs="1" />
      <xsd:element name="tasks" type="monitorListTasksType" minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
    <xsd:anyAttribute/>
  </xsd:complexType>

  <xsd:complexType name="monitorListTasksType">
    <xsd:sequence>
      <xsd:element name="task" type="monitorListTaskType" minOccurs="1" maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="monitorListTaskType">
    <xsd:sequence>
      <xsd:element name="name" type="monitorTaskNameType" minOccurs="0" maxOccurs="1" />
      <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1" />
      <xsd:element name="taskXML" type="xsd:string" minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>

</xsd:schema>
```

了解监视器列表消息

以下列表中描述了监视器列表消息中使用的元素和属性：

<monitorList>

包含元素的组元素描述了为代理定义的监视器。

属性	描述
代理程序	必需。 在其上定义资源监视器的代理的名称。
monitor	必需。 监视器的名称。 对于此代理是唯一的。
version	必需。 监视器列表消息格式的版本。

<status>

监视器的状态。

属性	描述
状态	监视器的状态。

<configuration>

组元素，包含描述监视器配置的元素。

<description>

监视器的描述。（当前未使用。）

<resources>

正在监视的资源。

<directory>

要监视的目录。

属性	描述
recursionLevel	要监视的从顶级向下的目录级别数。
标识	资源的标识。

<queue>

要监视的队列。

属性	描述
标识	资源的标识。

<triggerMatch>

包含 <conditions> 元素的元素。

<conditions>

包含资源监视器监视的条件的元素。该元素只能包含以下元素中的一个：<allOf>、<anyOf> 或 <condition>。

<allOf>

包含资源监视器监视的条件的元素。该元素可包含一个或多个 <condition> 元素。要触发资源监视器，必须满足该元素内的所有条件。

<anyOf>

包含资源监视器监视的条件的元素。该元素可包含一个或多个 <condition> 元素。要触发资源监视器，必须仅满足该元素内的一个条件。

<condition>

包含资源监视器监视的一项条件的元素。This element can contain only one of the following elements: <fileMatch>, <fileNoMatch>, <fileSize>, <queueNotEmpty>, <completeGroups>, or <fileSizeSame>. It can also contain a <name> element and a <resource> element.

如果正在监视的资源是一个目录，那么必须在条件中指定以下三个元素中的一个：

- fileMatch
- fileNoMatch
- fileSize

如果监视中的资源是队列，那么必须在条件中指定以下两个元素中的一个：

- queueNotEmpty
- completeGroups

<fileMatch>

文件名匹配条件的组元素。

<pattern>
指定文件名匹配模式。资源上的文件必须与模式相匹配才能满足条件。缺省模式为 *（任何文件都将匹配）。

<fileNoMatch>
逆文件名匹配条件的组元素。

<pattern>
指定逆文件名匹配模式。如果受监视资源上的所有文件均不匹配，那么满足条件。缺省模式为 *（缺少任何文件都将匹配）。

<fileSize>
文件大小比较的组元素。

<compare>
指定文件大小比较。该值必须是非负整数。

属性	描述
操作员	要使用的比较运算符。仅支持> = '。
单元	指定文件大小单位，可以是以下其中一项： • B - 字节 • KB - 千字节 • MB - 兆字节 • GB - 千兆字节 单位值不区分大小写，因此 mb' 与 MB' 作用相同。

<pattern>
要匹配的文件名模式。缺省值为 *（任何文件都将匹配）。

<queueNotEmpty>
只能在资源是队列时指定此属性。指定要触发监视器，队列上必须有消息。

<completeGroups>
只能在资源是队列时指定此属性。指定要触发监视器，队列上必须有完整的一组消息。针对队列上每个完整的组执行一项传输任务。

<name>
条件的名称。

<resource>
标识要与条件相比较的资源定义。

属性	描述
标识	资源的唯一标识。

<tasks>
组元素，包含的元素可指定在满足监视器触发器条件时要调用的任务。

<task>
组元素，定义在满足触发器条件时监视器将调用的个别任务。当前只能指定一项任务。

<name>
任务的名称。接受任何字母数字字符。

<description>
任务的描述。允许任何文本值。

<taskXML>
描述监视器将要执行的任务的 XML 消息。该元素的内容采用转义 XML 格式。

<pollInterval>
每次针对触发器条件检查资源的时间间隔。

属性	描述
单元	指定轮询时间间隔的时间单位。有效值包括： • 秒 • 分钟 • 小时 • days • weeks • months • years

<batch>

要在一个批次中包含的触发器匹配项的最大数目。

属性	描述
maxSize	要在一个批次中包含的触发器匹配项的最大数量

以下 XML 显示了在 AGENT_JUPITER 上创建名为 MONITORTWO 的监视器时，发布至主题字符串 SYSTEM.FTE/monitors/*agent_name*/MONITORTWO 的保留出版物的示例。<taskXML> 元素内的转义 XML 描述了当满足监视器条件时提交的任务。

```
<?xml version="1.0" encoding="UTF-8"?>
<lst:monitorList xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:lst="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
  xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition MonitorList.xsd"
  version="4.00"
  agent="AGENT_JUPITER"
  monitor="MONITORTWO">
  <status state="started"/>
  <configuration>
    <description/>
    <resources>
      <directory recursionLevel="0" id="">/srv/nfs/incoming</directory>
    </resources>
    <triggerMatch>
      <conditions>
        <condition>
          <name/>
          <resource id="">
            <fileMatch>
              <pattern>*.completed</pattern>
            </fileMatch>
          </condition>
        </conditions>
      </triggerMatch>
      <tasks>
        <task>
          <name/>
          <description/>
          <taskXML>&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;request
            xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="4.00"
            xsi:noNamespaceSchemaLocation="FileTransfer.xsd"&gt;&lt;managedTransfer&gt;
              &lt;originator&gt;&lt;hostname&gt;example.com.&lt;/hostname&gt;
              &lt;userID&gt;mqm&lt;/userID&gt;&lt;/originator&gt;
              &lt;sourceAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/&gt;
              &lt;destinationAgent QMgr="QM_JUPITER" agent="AGENT_SATURN"/&gt;
              &lt;transferSet&gt;&lt;item checksumMethod="MD5" mode="binary"&gt;
                &lt;source disposition="leave" recursive="false"&gt;&lt;file
                  &gt;/srv/nfs/incoming/*.txt&lt;/file&gt;&lt;/source&gt;
                &lt;destination exist="error" type="directory"&gt;
                  &lt;file&gt;/srv/backup&lt;/file&gt;&lt;/destination&gt;
                &lt;/item&gt;&lt;/transferSet&gt;&lt;/managedTransfer&gt;
              &lt;/request&gt;
            </taskXML>
          </task>
        </tasks>
      </configuration>
      <pollInterval units="minutes">1</pollInterval>
      <batch maxSize="1"/>
    </lst:monitorList>
```

安排列表消息格式

发布到主题字符串为 SYSTEM.FTE/Scheduler/*agent_name* 的保留发布上的 XML 消息符合 ScheduleList.xsd 模式。XML 消息将列出所有属于该代理的活动安排。

fteListScheduledTransfers 命令和 WebSphere MQ Explorer 插件会使用该信息以向用户显示安排列表。ScheduleList.xsd 模式文档位于 *MQ_INSTALLATION_PATH*/mqft/samples/schema 目录中。ScheduleList.xsd 模式将导入位于相同目录中的 FileTransfer.xsd。

模式

以下模式描述了监视器列表 XML 消息中哪些元素有效。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="FileTransfer.xsd"/>

  <xsd:element name="schedules">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="managedTransfer" type="scheduledManagedTransferType" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="size" type="xsd:nonNegativeInteger" use="required"/>
      <xsd:attribute name="agent" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="scheduledManagedTransferType">
    <xsd:sequence>
      <xsd:element name="originator" type="origRequestType" maxOccurs="1" minOccurs="1"/>
      <xsd:element name="schedule" type="scheduleListType" maxOccurs="1" minOccurs="0"/>
      <xsd:element name="sourceAgent" type="agentType" maxOccurs="1" minOccurs="1"/>
      <xsd:element name="destinationAgent" type="agentClientType" maxOccurs="1" minOccurs="1"/>
      <xsd:element name="trigger" type="triggerType" maxOccurs="1" minOccurs="0"/>
      <xsd:element name="reply" type="replyType" maxOccurs="1" minOccurs="0"/>
      <xsd:element name="transferSet" type="transferSetType" maxOccurs="1" minOccurs="1"/>
      <xsd:element name="job" type="jobType" maxOccurs="1" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="id" type="idType" use="required"/>
  </xsd:complexType>

  <xsd:complexType name="scheduleListType">
    <xsd:sequence>
      <xsd:element name="submit" type="submitType" maxOccurs="1" minOccurs="1"/>
      <xsd:element name="repeat" type="repeatType" maxOccurs="1" minOccurs="0"/>
      <xsd:element name="next" type="noZoneTimeType" maxOccurs="1" minOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>

</xsd:schema>
```

了解安排列表消息

以下列表描述了安排列表消息中使用的元素和属性：

<schedules>

包含有关单个代理上定义的所有安排的信息的组元素。

属性	描述
代理程序	必需。 在其上定义安排的源代理的名称。
size	必需。 在该代理上定义的安排的数量。
version	必需。 安排列表消息格式的版本。

<managedTransfer>

包含有关单个安排的信息的组元素。

属性	描述
标识	必需。 安排请求消息的十六进制字符串标识。

<originator>

安排请求的发起方。

<hostName>

提交安排请求的主机名。

<userID>

提交安排请求的用户的用户标识。

<mqmdUserID>

提交安排请求的用户的 MQMD 用户标识。

<webBrowser>

如果通过 Web 网关提交了安排请求，那么是指提交请求的 Web 浏览器。

<webUserID>

如果通过 Web 网关提交安排请求，那么是指提交安排请求的用户的 Web 用户标识。

<schedule>

包含描述何时发生已安排传输的元素的元素。

<submit>

指定已调度传输应启动的日期和时间。

属性	描述
timebase	指定要使用的时区。该属性的值可以是以下值之一： • source - 使用源代理的时区 • admin - 使用发出命令的管理员的时区 • UTC - 使用全球标准时间
时区	依据 timebase 值的时区描述

<repeat>

包含有关已调度传输重复频率、已调度传输重复次数以及已调度传输何时停止重复的详细信息的组元素。

属性	描述
区间 (interval)	时间间隔单位，必须是以下值之一： • 分钟 • 小时 • days • weeks • months • years

<frequency>

重复传输之前必须等待的时间段。

属性	描述
区间 (interval)	时间间隔单位，必须是以下值之一： • 分钟 • 小时 • days • weeks • months • years

<expireTime>

指定重复的已调度传输停止的日期和时间的可选元素。该元素与 <expireCount> 元素互斥。

<expireCount>

指定在停止前发生已调度文件传输的次数的可选元素。该元素与 <expireTime> 元素互斥。

<next>

指定下一个已安排传输应启动的日期和时间。

<sourceAgent>

指定源文件所在系统上的代理的名称。

属性	描述
代理程序	指定代理的名称。
QMgr	代理队列管理器的名称。

<destinationAgent>

指定您要向其传输文件的系统上代理的名称。

属性	描述
代理程序	指定代理的名称。
QMgr	代理队列管理器的名称。

<trigger>

指定在文件传输发生时必须满足的条件的可选元素。

属性	描述
日志	指示是否记录触发失败的标志。 以下是有效值： • yes - 为失败的已触发传输创建日志条目 • no - 不为失败的已触发传输来创建日志条目

<reply>

指定为同步文件传输生成的临时应答队列的名称（在命令行上使用 **-w** 参数指定）。队列的名称由 `command.properties` 配置文件中的键 **dynamicQueuePrefix** 或缺省值 `WMQFTE.*` 定义。（如果未指定）。

属性	描述
QMGR	在其上生成临时动态队列以接收应答的命令队列管理器的名称。

<transferSet>

指定要与已安排传输一起执行的一组文件传输。在传输期间，<transferSet> 是包含 <item> 元素的组元素。

属性	描述
priority	传输的优先级。优先级为 0-9 之间的一个值，其中 0 为最低优先级。缺省优先级为 0，并且缺省情况下，传输使用源代理的优先级。

<job>

Optional group element containing job information for the entire transfer specification. <job> is a user-defined job name identifier that is added to the log message when the transfer has started. This <job> element is the same as the <job> element that appears in the transfer log message, which is described in the following topic: [第 662 页的『文件传输日志消息格式』](#)。

示例

```
<?xml version="1.0" encoding="UTF-8"?>
<schedules xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  size="2"
  version="4.00"
  agent="AGENT_JUPITER"
  xsi:namespaceSchemaLocation="ScheduleList.xsd">
  <managedTransfer id="1">
    <originator>
      <hostName>example.com.</hostName>
      <userID>mqm</userID>
    </originator>
```

```

        <schedule>
          <submit timebase="admin" timezone="Europe/London">2010-01-01T21:00+0000</
submit>
            <next>2010-01-01T21:00+0000</next>
        </schedule>
        <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
        <destinationAgent agent="AGENT_SATURN" QMgr="QM_JUPITER"/>
        <reply QMgr="QM_JUPITER">WMQFTE.4D400F8B20004E06</reply>
        <transferSet>
          <item mode="binary" checksumMethod="MD5">
            <source recursive="false" disposition="leave">
              <file>/etc/passwd</file>
            </source>
            <destination type="directory" exist="overwrite">
              <file>/tmp</file>
            </destination>
          </item>
        </transferSet>
      </managedTransfer>
      <managedTransfer id="2">
        <originator>
          <hostName>example.com.</hostName>
          <userID>mqm</userID>
        </originator>
        <schedule>
          <submit timebase="admin" timezone="Europe/London">2010-12-31T09:00+0000</
submit>
            <next>2010-12-31T09:00+0000</next>
        </schedule>
        <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
        <destinationAgent agent="AGENT_NEPTUNE" QMgr="QM_JUPITER"/>
        <reply QMgr="QM_JUPITER">WMQFTE.4D400F8B20004E09</reply>
        <transferSet>
          <item mode="binary" checksumMethod="MD5">
            <source recursive="false" disposition="leave">
              <file>/etc/passwd</file>
            </source>
            <destination type="directory" exist="overwrite">
              <file>/tmp</file>
            </destination>
          </item>
        </transferSet>
      </managedTransfer>
    </schedules>

```

示例模板 XML 消息

在创建模板时，将向主题字符串为 `Templates/template_ID` 的 `SYSTEM.FTE` 主题发布一条消息。该示例 XML 描述在您的 IBM MQ Managed File Transfer 网络中定义的单个模板。

```

<?xml version="1.0" encoding="UTF-8"?>
<transferTemplate version="4.00" id="baf9df73-45c2-4bb0-a085-292232ab66bc">
  <name>BASIC_TEMPLATE</name>
  <sourceAgentName>AGENT_JUPITER</sourceAgentName>
  <sourceAgentQMgr>QM_JUPITER</sourceAgentQMgr>
  <destinationAgentName>AGENT_SATURN</destinationAgentName>
  <destinationAgentQMgr>QM_JUPITER</destinationAgentQMgr>
  <fileSpecs>
    <item mode="binary" checksumMethod="MD5">
      <source recursive="false" disposition="leave">
        <file>/etc/passwd</file>
      </source>
      <destination type="directory" exist="overwrite">
        <file>/tmp</file>
      </destination>
    </item>
  </fileSpecs>
  <priority>0</priority>
</transferTemplate>

```

相关任务

第 249 页的『使用 IBM MQ Explorer 创建文件传输模板』

您可以从 IBM MQ Explorer 或从命令行创建文件传输模板。然后，您可以使用该模板来利用模板详细信息创建新的文件传输，或者提交该模板以启动文件传输。

相关参考

第 484 页的『fteCreateTemplate (新建文件传输模板)』

fteCreateTemplate 命令创建文件传输模板，您可以保留该模板以供将来使用。唯一的必需参数是 **-tn** (*template_name*) 参数。其他所有参数都是可选的，但如果指定源文件，还必须提供目标文件。类似地，如果指定目标文件，还必须指定源文件规范。

文件传输状态消息格式

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 `SYSTEM.FTE/Transfers/agent_name/transfer ID` 主题中），它符合 `TransferStatus.xsd` XML 模式。`TransferStatus.xsd` 文件位于 WMQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

模式

以下模式描述了在传输状态 XML 消息中哪些元素有效。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>
  <xsd:element name="transaction">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="sourceAgent" type="agentType"
          maxOccurs="1" minOccurs="1"/>
        <xsd:element name="destinationAgent" type="agentType"
          maxOccurs="1" minOccurs="1"/>
        <xsd:element name="transferSet" type="transferSetType"
          maxOccurs="1" minOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="ID" type="IDType" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="transferSetType">
    <xsd:sequence>
      <xsd:element name="stats" type="statsType"
        maxOccurs="1" minOccurs="1" />
      <xsd:element name="current" type="currentType"
        maxOccurs="1" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="time" type="xsd:dateTime" use="required" />
  </xsd:complexType>
  <xsd:complexType name="currentType">
    <xsd:sequence>
      <xsd:element name="source" type="fileSourceType"
        maxOccurs="1" minOccurs="1" />
      <xsd:element name="destination" type="fileDestinationType"
        maxOccurs="1" minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="transferred" type="xsd:nonNegativeInteger"
      use="required" />
    <xsd:attribute name="size" type="xsd:nonNegativeInteger" use="required" />
  </xsd:complexType>
  <xsd:complexType name="statsType">
    <xsd:attribute name="bytes" type="xsd:nonNegativeInteger"
      use="required" />
    <xsd:attribute name="seconds" type="xsd:decimal"
      use="required" />
    <xsd:attribute name="currentItem" type="xsd:nonNegativeInteger"
      use="required" />
    <xsd:attribute name="totalItems" type="xsd:nonNegativeInteger" use="required" />
  </xsd:complexType>
</xsd:schema>
```

了解传输状态消息

下表描述了传输状态消息中使用的元素和属性：

<transaction>

包含用于文件传输的所有元素的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 提供的该元素的版本，指定该值。
标识	文件传输的唯一标识。

<sourceAgent>

指定源文件所在系统上的代理的名称。

属性	描述
代理程序	代理的名称。
QMGr	代理队列管理器的名称。

<destinationAgent>

指定您要向其传输文件的系统上代理的名称。

属性	描述
代理程序	代理的名称。
QMGr	代理队列管理器的名称。

<transferset>

指定正一起执行的一组文件传输。传输中的所有文件都必须开始于相同的源代理，结束于相同的目标代理。

属性	描述
time	指定日期和时间（采用日期时间格式）。

<stats>

必需。定义关于在给定秒数内传输的度量值，包括迄今为止复制的字节数。同时还在 <transferSet> 中提供当前项号与总项数。

属性	描述
字节	迄今为止复制的字节数。
秒	传输这些字节所花费的秒数。
currentItem	正传输的当前项的索引。
totalItems	正传输的总项数。

<current>

可选元素。包含指定目前正在进行的文件传输的元素的组元素。<current> 元素指示迄今为止已为当前项传输的数据字节数以及预期的字节总数。

<source>

包含指定源文件名的元素的组元素。

<file>

指定要传输的文件的源路径。该路径是为传输指定的路径。该路径可能与作为传输日志一部分输出的路径不同，后者是路径的绝对形式。

<destination>

包含指定目标文件名或规范的元素的组元素。

<file>

指定要传输的文件的目标路径。该路径是为传输指定的路径。该路径可能与作为传输日志一部分输出的路径不同，后者是路径的绝对形式。

属性	描述
别名	指定目标文件的别名。该别名是源文件的名称，不包括针对传输指定的任何目录路径。
文件空间	指定将目标文件写入的文件空间的名称。

<queue>

与 <destination> 元素一起使用时，指定要传输到的队列的名称。该名称格式为 QUEUE 或 QUEUE@QUEUE_MANAGER。

相关参考

- 第 661 页的『传输进度消息示例』
传输期间，会将消息发布到主题字符串为 Transfers/agent_name/transfer_ID 的 SYSTEM.FTE 主题。XML 示例会显示单个文件传输和多个文件传输的进度消息。
- 第 647 页的『代理状态消息格式』
在创建或启动代理时，该代理会向其协调队列管理器上的 SYSTEM.FTE 主题（在 SYSTEM.FTE/Agents/agent_name 主题上）发布其详细信息。
- 第 842 页的『文件传输请求消息格式』
文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。
- 第 662 页的『文件传输日志消息格式』
文件传输日志消息发布在主题字符串为 Log/agent_name/transfer_ID 的 SYSTEM.FTE 主题内。这些消息符合模式 TransferLog.xsd，该模式位于 IBM MQ Managed File Transfer 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。
- 第 686 页的『已安排的传输日志消息格式』
代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/agent_name/schedule ID 主题中）。该消息符合 ScheduleLog.xsd XML 模式。
- 第 859 页的『监视器请求消息格式』
当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 fteCreateMonitor 命令或使用 WebSphere MQ Explorer 接口。
- 第 871 页的『针对安全性的消息格式』
本主题描述发布到协调队列管理器且与安全性相关的消息。

传输进度消息示例

传输期间，会将消息发布到主题字符串为 Transfers/agent_name/transfer_ID 的 SYSTEM.FTE 主题。XML 示例会显示单个文件传输和多个文件传输的进度消息。

单个文件传输

以下示例显示了正在进行的单个文件传输的详细信息。

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d223d0020"
  xsi:noNamespaceSchemaLocation="TransferStatus.xsd">
  <sourceAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
  <destinationAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
  <transferSet time="2011-01-26T13:03:26.542Z">
    <stats bytes="1198" seconds="0.018" currentItem="1" totalItems="1"/>
    <current transferred="1151" size="1151">
      <source>
```



```

        <file>/etc/passwd</file>
      </source>
      <destination>
        <file>/tmp/passwd</file>
      </destination>
    </current>
  </transferSet>
</transaction>

```

多个文件传输

如果传输集中存在多个文件，那么传输状态消息会表明正在处理哪个文件，以及到目前为止，已传输多少字节。

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d035c0020"
  xsi:noNamespaceSchemaLocation="TransferStatus.xsd">
  <sourceAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
  <destinationAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
  <transferSet time="2011-01-26T13:12:58.636Z">
    <stats bytes="440" seconds="0.082" currentItem="10" totalItems="10"/>
    <current transferred="0" size="0">
      <source>
        <file>/srv/nfs/incoming/file10.txt</file>
      </source>
      <destination>
        <file>/srv/nfs/outgoing/file10.txt</file>
      </destination>
    </current>
  </transferSet>
</transaction>

```

文件传输日志消息格式

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 IBM MQ Managed File Transfer 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

如果要监视文件传输或收集有关它们的数据，请设置一个针对您感兴趣的传输而定制的通配符主题预订。例如：

```
Log/#
```

或者，

```
Log/FTEAGENT/#
```

该预订可以是持久或非持久的。持久预订在预订应用程序与队列管理器之间的连接关闭后继续存在。非持久预订只有在预订应用程序与队列管理器之间的连接处于打开状态时才存在。

模式

以下模式描述在传输日志 XML 消息中有效的元素。

```

<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>
  <xsd:element name="transaction">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="action" type="actionType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceAgent" type="agentExitStatusType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceWebGateway" type="webGatewayType" maxOccurs="1" minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

```

        maxOccurs="1"                minOccurs="0" />
<xsd:element name="sourceWebUser"    type="webUserType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="destinationAgent" type="agentExitStatusType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="destinationWebGateway" type="webGatewayType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="destinationWebUser" type="webUserType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="agent"            type="agentExitStatusType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="originator"        type="origRequestType"
maxOccurs="1"                minOccurs="1" />
<xsd:element name="status"            type="statusType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="trigger"          type="triggerType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="transferSet"       type="transferSetType"
maxOccurs="1"                minOccurs="1" />
<xsd:element name="job"              type="jobType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="scheduleLog"       type="scheduleLogType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="statistics"        type="statisticsType"
maxOccurs="1"                minOccurs="0" />
</xsd:sequence>
<xsd:attribute name="version"         type="versionType"         use="required" />
<xsd:attribute name="ID"              type="IDType"              use="required" />
<xsd:attribute name="relatedID"       type="IDType"              use="optional" />
<xsd:attribute name="agentRole"       type="agentRoleType"       use="optional" />
</xsd:complexType>
</xsd:element>

<xsd:complexType name="agentExitStatusType">
  <xsd:complexContent>
    <xsd:extension base="agentType">
      <xsd:sequence>
        <xsd:element name="startExits" type="exitGroupType" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="endExits"  type="exitGroupType" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="systemInfo" type="systemInfoType" minOccurs="0"
maxOccurs="1" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="transferSetType">
  <xsd:sequence>
    <xsd:element name="metaDataSet"    type="metaDataSetType"
maxOccurs="1"                minOccurs="0" />
    <xsd:element name="call"          type="callGroupType"
maxOccurs="1"                minOccurs="0" />
    <xsd:element name="preSourceCall"  type="callGroupType"
maxOccurs="1"                minOccurs="0" />
    <xsd:element name="postSourceCall" type="callGroupType"
maxOccurs="1"                minOccurs="0" />
    <xsd:element name="preDestinationCall" type="callGroupType"
maxOccurs="1"                minOccurs="0" />
    <xsd:element name="postDestinationCall" type="callGroupType"
maxOccurs="1"                minOccurs="0" />
    <xsd:element name="item"          type="itemType"
maxOccurs="unbounded"        minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="index"         type="xsd:nonNegativeInteger" use="optional" />
  <xsd:attribute name="size"          type="xsd:nonNegativeInteger" use="optional" />
  <xsd:attribute name="startTime"     type="xsd:dateTime"         use="required" />
  <xsd:attribute name="total"         type="xsd:nonNegativeInteger" use="required" />
  <xsd:attribute name="bytesSent"     type="xsd:nonNegativeInteger" use="required" />
</xsd:complexType>

<xsd:complexType name="itemType">
  <xsd:sequence>
    <xsd:element name="source"         type="fileSourceChecksumType"
maxOccurs="1"                minOccurs="1" />
    <xsd:element name="destination"    type="fileDestinationChecksumType"
maxOccurs="1"                minOccurs="1" />
    <xsd:element name="status"         type="statusType"
maxOccurs="1"                minOccurs="1" />
  </xsd:sequence>
  <xsd:attribute name="mode"          type="modeType"             use="required" />

```

```

</xsd:complexType>

<xsd:complexType name="fileSourceChecksumType">
  <xsd:complexContent>
    <xsd:extension base="fileSourceType">
      <xsd:sequence>
        <xsd:element name="checksum" type="checksumType" minOccurs="0"
maxOccurs="1" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="fileDestinationChecksumType">
  <xsd:complexContent>
    <xsd:extension base="fileDestinationType">
      <xsd:sequence>
        <xsd:element name="checksum" type="checksumType"
minOccurs="0" maxOccurs="1" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="actionType">
  <xsd:simpleContent>
    <xsd:extension base="actionEnumType">
      <xsd:attribute name="time" type="xsd:dateTime" use="required" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="actionEnumType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="cancelled" />
    <xsd:enumeration value="started" />
    <xsd:enumeration value="progress" />
    <xsd:enumeration value="completed" />
    <xsd:enumeration value="malformed" />
    <xsd:enumeration value="notAuthorized" />
    <xsd:enumeration value="deleted" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="systemInfoType">
  <xsd:attribute name="architecture" type="xsd:string" use="required" />
  <xsd:attribute name="name" type="xsd:string" use="required" />
  <xsd:attribute name="version" type="xsd:string" use="required" />
</xsd:complexType>

<xsd:element name="malformed">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="action" type="actionType"
maxOccurs="1" minOccurs="1" />
      <xsd:element name="agent" type="agentExitStatusType"
maxOccurs="1" minOccurs="0" />
      <xsd:element name="status" type="statusType"
maxOccurs="1" minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="version" type="versionType" use="required" />
    <xsd:attribute name="ID" type="IDType" use="required" />
    <xsd:attribute name="agentRole" type="agentRoleType" use="required" />
  </xsd:complexType>
</xsd:element>

<xsd:element name="notAuthorized">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="action" type="actionType"
maxOccurs="1" minOccurs="1" />
      <xsd:element name="originator" type="origRequestType"
maxOccurs="1" minOccurs="1" />
      <xsd:element name="authority" type="xsd:string"
minOccurs="1" maxOccurs="1" />
      <xsd:element name="status" type="statusType"
maxOccurs="1" minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="version" type="versionType" use="required" />
    <xsd:attribute name="ID" type="IDType" use="required" />
    <xsd:attribute name="agentRole" type="agentRoleType" use="required" />
  </xsd:complexType>

```

```

</xsd:element>

<xsd:complexType name="statisticsType">
  <xsd:sequence>
    <xsd:element name="actualStartTime" type="xsd:dateTime"
      maxOccurs="1" minOccurs="0"/>
    <xsd:element name="retryCount" type="xsd:nonNegativeInteger"
      maxOccurs="1" minOccurs="1"/>
    <xsd:element name="numFileFailures" type="xsd:nonNegativeInteger"
      maxOccurs="1" minOccurs="1"/>
    <xsd:element name="numFileWarnings" type="xsd:nonNegativeInteger"
      maxOccurs="1" minOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="webGatewayType">
  <xsd:attribute name="webGatewayName" type="xsd:string" use="optional" />
  <xsd:attribute name="webGatewayAgentName" type="xsd:string" use="optional" />
  <xsd:attribute name="webGatewayAgentQMgr" type="xsd:string" use="optional" />
</xsd:complexType>

<xsd:complexType name="webUserType">
  <xsd:attribute name="webGatewayName" type="xsd:string" use="required" />
  <xsd:attribute name="webGatewayAgentName" type="xsd:string" use="optional" />
  <xsd:attribute name="webGatewayAgentQMgr" type="xsd:string" use="optional" />
</xsd:complexType>

</xsd:schema>

```

了解传输日志消息

<transaction>

用于指定一组要一起执行的传输的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 的详细说明指定该元素的版本。
标识	指定唯一事务标识。该标识最多可包含 48 个字母数字字符。
relatedID	可选。如果事务是从文件空间中删除或下载文件，那么 relatedID 指定将该文件上载到该文件空间中的传输的事务标识。
agentRole	可选。指定关注的代理是否在源或目标系统上
xmlns:xsi	名称空间声明。指示此模式中使用的元素和数据类型派生自 "https://www.w3.org/2001/XMLSchema-instance" 名称空间。
xsi:noNamespaceSchemaLocation	指定 XML 模式文档的名称和位置，以在没有名称空间声明的情况下验证该消息。为此属性指定的值必须引用 IBM MQ Managed File Transfer TransferLog.xsd 文档。

<action>

描述在 time 属性记录的时间，文件传输的状态。状态可以是以下某个值：

- 已启动
- 正在进行
- 已完成
- 已取消
- malformed（指示无法解释文件传输请求消息的内容）
- notAuthorized
- deleted

属性	描述
time	捕获传输状态的时间，以 UTC 格式表示。

<sourceAgent>

指定源文件所在系统上的代理的名称。只能指定 <sourceAgent>、<sourceWebUser> 或 <sourceWebGateway>。

<startExits>

包含一个或多个用户出口元素的组元素。该元素只能出现一次。

<endExits>

包含一个或多个用户出口元素的组元素。该元素只能出现一次。

<systemInfo>

描述系统体系结构、名称和版本。该元素只能出现一次。

属性	描述
代理程序	源系统上的代理的名称。
QMgr	源系统上的队列管理器的名称。
agentType	代理的类型。有效值包括： • STANDARD - 一般代理 • BRIDGE - 协议网桥代理 • CD_BRIDGE - Connect:Direct 网桥代理 • EMBEDDED - 嵌入式代理 • WEB_GATEWAY - Web 代理 • SFG - Sterling File Gateway 嵌入式代理
bridgeURL	可选。如果代理是协议网桥代理，那么是指托管协议服务器的系统的主机名称。
pnode	可选。如果代理是 Connect:Direct 网桥代理，那么是指参与传输的 Connect:Direct 主节点的名称。
snode	可选。如果代理为 Connect:Direct 网桥代理，那么是指参与传输的 Connect:Direct 辅助节点的名称。
bridgeNode	可选。如果代理为 Connect:Direct 网桥代理，那么是指作为 Connect:Direct 网桥的组成部分的 Connect:Direct 节点的名称。这与主节点或辅助节点是同一个节点。

<sourceWebUser>

指定将源文件上载到 Web 网关的 Web 用户的名称。只能指定 <sourceAgent>、<sourceWebUser> 或 <sourceWebGateway>。

属性	描述
webGatewayName	接收来自 Web 用户的文件的 Web 网关的名称。
webGatewayAgentName	Web 网关用来向目标发送文件的 Web 代理的名称。
webGatewayAgentQMgr	Web 代理的队列管理器的名称。

<sourceWebGateway>

指定源文件下载自的 Web 网关的名称。只能指定 <sourceAgent>、<sourceWebUser> 或 <sourceWebGateway>。

属性	描述
webGatewayName	接收来自 Web 用户的文件的 Web 网关的名称。

属性	描述
webGatewayAgentName	Web 网关用来向目标发送文件的 Web 代理的名称。
webGatewayAgentQMgr	Web 代理的队列管理器的名称。

<destinationAgent>

指定文件要传输到的系统上的代理的名称。只能指定 <destinationAgent>、<destinationWebGateway> 或 <destinationWebUser>。

属性	描述
代理程序	目标系统上的代理的名称。
QMgr	目标系统上的队列管理器的名称。
agentType	代理的类型。有效值包括： • STANDARD - 一般代理 • BRIDGE - 协议网桥代理 • CD_BRIDGE - Connect:Direct 网桥代理 • EMBEDDED - 嵌入式代理 • WEB_GATEWAY - Web 代理 • SFG - Sterling File Gateway 嵌入式代理
bridgeURL	可选。如果代理是协议网桥代理，那么是指托管协议服务器的系统的主机名称。
pnode	可选。如果代理是 Connect:Direct 网桥代理，那么是指参与传输的 Connect:Direct 主节点的名称。
snode	可选。如果代理为 Connect:Direct 网桥代理，那么是指参与传输的 Connect:Direct 辅助节点的名称。
bridgeNode	可选。如果代理为 Connect:Direct 网桥代理，那么是指作为 Connect:Direct 网桥的组成部分的 Connect:Direct 节点的名称。这与主节点或辅助节点是同一个节点。

<startExits>

包含一个或多个用户出口元素的组元素。该元素只能出现一次。

<endExits>

包含一个或多个用户出口元素的组元素。该元素只能出现一次。

<systemInfo>

描述系统体系结构、名称和版本。该元素只能出现一次。

<destinationWebUser>

指定从 Web 网关下载文件的 Web 用户的名称。只能指定 <destinationAgent>、<destinationWebGateway> 或 <destinationWebUser>。

属性	描述
webGatewayName	接收来自 Web 用户的文件的 Web 网关的名称。

<destinationWebGateway>

指定从 Web 网关下载文件的 Web 用户的名称。只能指定 <destinationAgent>、<destinationWebGateway> 或 <destinationWebUser>。

属性	描述
webGatewayName	接收来自 Web 用户的文件的 Web 网关的名称。
webGatewayAgentName	Web 网关使用的 Web 代理的名称。

属性	描述
webGatewayAgentQMgr	Web 代理的队列管理器的名称。

<originator>

包含指定请求发起方的元素的组元素。

<hostName>

源文件所在系统的主机名。

<userID>

发起文件传输的用户标识。

<mqmdUserID>

在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<webUserID>

可选。向提交传输请求的 Web 浏览器提供的用户标识。

<webBrowser>

可选。提交传输请求的 Web 浏览器。

<status>

结果代码和补充消息。

<trigger>

包含在原始传输请求中定义的触发器元素的组元素。这些元素可以是以下中的任何一个或所有：

<fileExist>

基于是否存在文件的触发条件

<fileSize>

基于文件是否满足或超出了指定大小的触发条件。

<transferSet>

指定一组要一起执行的文件传输。在传输期间，<transferSet> 是包含 <item> 元素的组元素。

属性	描述
startTime	记录一组传输开始的时间，以 UTC 格式表示。
total	指定该传输集中的总项数。
索引 (index)	可选属性。指定第一个项在传输集进度中的位置。index 属性自零开始递增。例如，如果将 index 设置为 1，那么进度消息则为两个项中的第二个。
size	可选属性。指定进度报告中的项数。
priority	可选属性。传输的优先级。优先级为 0-9 之间的一个值，其中 0 为最低优先级。缺省优先级为 0，并且缺省情况下，传输使用源代理的优先级。

<metaDataSet>

包含一个或多个以下属性的组元素：

<metaData>

属性	描述
key	元数据键值对中的键部分。<metaData> 元素内容包含键值对的值部分。例如，<metaData key="testkey1">testvalue1</metaData>

<job>

Group element that contains an element specifying job details. <job> is a user-defined job name identifier that is added to the log message when the transfer has started. This <job> element is the same as the <job> element that is included in the transfer request message, which is described in the following topic: [第 842 页的『文件传输请求消息格式』](#)。

<name>

名称可以是任意字符串。

<scheduleLog>

包含指定源和目标文件名和位置的元素的组元素。

属性	描述
标识	如果传输为安排的传输，那么其与调度标识匹配。

<item>

包含指定源和目标文件名和位置的元素的组元素。

<source>

包含 <file> 元素或 <queue> 元素以及源系统上文件的 <checksum> 元素的组元素。

属性	描述
recursive	指定当 <source> 元素为目录或包含通配符时，以递归方式传输子目录中的文件。
disposition	指定在 <source> 已成功传输到其目标时要在 <source> 元素上执行的操作。有效选项如下： • leave - 源文件保持不变。 • delete - 在成功传输源文件后将源文件从源系统中删除。
correlationBoolean	布尔关联值。如果源为 Connect:Direct 网桥，那么该属性指定 Connect:Direct 进程是否为用户定义的。
correlationString1	字符串关联值。如果源为 Connect:Direct 网桥，那么该属性指定在传输的目标出现的 Connect:Direct 进程的名称。
correlationNum1	数字相关值。如果源为 Connect:Direct 网桥，那么该属性指定在传输的目标出现的 Connect:Direct 进程的标识号。

<queue>

在与 <source> 元素一起使用时，指定从中读取所传输消息的队列名称，该队列位于源代理队列管理器上。

属性	描述
messageCount	从队列中读取的消息数目。
groupId	从队列读取的消息的 WebSphere MQ 组标识。

<destination>

包含 <file> 元素或 <queue> 元素以及目标的 <checksum> 元素的组元素。

只有 <file> 和 <queue> 中的一个作为目标的子元素。

属性	描述
类型	目标的类型。有效选项如下： • file - 将文件指定为目标 • directory - 将目录指定为目标 • dataset - 指定 z/OS 数据集作为目标 • pds - 指定 z/OS 分区数据集作为目标 • queue - 指定 WebSphere MQ 队列作为目标 仅当 <destination> 元素的子元素为 <queue> 时，值 队列 才有效。

属性	描述
	仅当 <destination> 元素具有子元素 <file> 时，其他值才有效。
exist	指定在目标系统上已存在目标文件时执行的操作。有效选项如下： • error - 报告一个错误，并且未传输文件。 • overwrite - 覆盖现有的目标文件。 如果 <destination> 元素具有子元素 <queue> ，那么此属性不能存在。
correlationBoolean	布尔关联值。如果目标为 Connect:Direct 网桥，那么该属性指定 Connect:Direct 进程是否为用户定义的。
correlationString1	字符串关联值。如果目标为 Connect:Direct 网桥，那么该属性指定在传输的目标出现的 Connect:Direct 进程的名称。
correlationNum1	数字相关值。如果目标为 Connect:Direct 网桥，那么该属性指定在传输的目标出现的 Connect:Direct 进程的标识号。

<file>

指定传输的文件的绝对路径（在源和目标上的路径）。标准路径与操作系统的格式一致，例如 C:/from/here.txt。文件 URI 未使用。

<queue>

在与 **<destination>** 元素一起使用时，指定传输到的队列名称，该队列位于与目标代理队列管理器连接的任何队列管理器上。

属性	描述
messageCount	写入队列的消息数目。
messageLength	写入队列的消息的长度。
groupId	如果传输请求指定文件被划分为多条消息，那么该属性的值将为写入队列的消息的 WebSphere MQ 组标识。
messageId	如果传输请求未指定文件被划分为多条消息，那么该属性的值将为写入队列的消息的 WebSphere MQ 消息标识。

<checksum>

可选元素。

指定生成消息摘要以创建数字签名的散列算法的类型。目前，IBM MQ Managed File Transfer 仅支持消息摘要算法 5 (MD5)。校验和提供一种方法来确认传输文件的完整性。

<malformed>

表示格式错误的消息的组元素。

属性	描述
version	
标识	
agentRole	源代理或目标代理

<statistics>

表示所传输的统计信息（如果可用）的组元素。

<actualStartTime>

代理开始运行传输的实际时间。通常，该时间与记录的传输启动时间相同（或非常接近）。然而，当代理忙碌时，提交的传输可能需要排队，直到代理有能力运行传输。

<retryCount>

传输进入恢复状态并由代理程序重试的次数。传输可能进入恢复状态，原因是由于 WebSphere MQ 网络错误或由于其在某时间段内未收到数据或确认消息，导致源或目标代理中断通信。该时间段由以下代理程序属性确定：transferAckTimeout 和 transferAckTimeoutRetries。

<numFileFailures>

传输失败的 transferSet 中的文件数。

<numFileWarnings>

传输期间出生警告但却成功传输的 transferSet 中的文件数。

示例

为以下每种类型的传输提供符合该模式的 XML 消息示例：

- [单个文件的传输](#)
- [包含多个文件的传输](#)
- [失败的文件传输](#)
- [定义有触发器的传输](#)
- [按照安排启动的传输](#)
- [调用用户出口的传输](#)
- [通过 Web 网关请求的传输](#)
- [通过 Connect:Direct 网桥节点的传输](#)

相关参考

[第 671 页的『单个传输日志消息示例』](#)

进行传输时，会将消息发布到主题字符串为 Log/agent_name/transfer_ID 的 SYSTEM.FTE 主题。以下 XML 示例显示了已启动、正在进行和已完成的单个文件传输的日志消息。

[第 673 页的『多文件传输日志消息示例』](#)

进行包含多个文件的传输时发布至 SYSTEM.FTE 主题（含主题字符串 Log/agent_name/transfer_ID）的消息示例。

[第 675 页的『失败的传输日志消息示例』](#)

进行传输时，会将消息发布到主题字符串为 Log/agent_name/transfer_ID 的 SYSTEM.FTE 主题。XML 示例显示了传输开始时、传输过程中以及传输完成后失败的文件传输的日志消息。

[第 677 页的『触发的传输消息格式』](#)

在进行传输的过程中，会向具有主题字符串 Log/agent_name/transfer_ID 的 SYSTEM.FTE 主题发布消息。以下 XML 示例显示了在启动一个包含触发条件的文件传输时创建的日志消息。

[第 678 页的『用户出口消息格式』](#)

在进行传输的过程中，会向具有主题字符串 Log/agent_name/transfer_ID 的 SYSTEM.FTE 主题发布消息。该 XML 示例显示了在进行的文件传输包含对用户出口的调用时所创建的日志消息。

[第 681 页的『基于 Web 的传输的消息格式中添加的内容』](#)

来自通过 IBM MQ Managed File Transfer Web 网关 请求的传输的 Started 和 Completed 日志消息包含额外的元数据。该元数据包含有关 HTTP 请求以及有关托管 Web 网关的应用程序服务器的信息。

[第 683 页的『Connect:Direct 网桥传输消息示例』](#)

当目标代理或源代理为 Connect:Direct 网桥代理时，destinationAgent 或 sourceAgent 元素会包含更多的属性。已开始日志消息只包含一部分有关 Connect:Direct 传输的信息。进行中和已完成日志消息包含有关 Connect:Direct 传输的完整信息。

单个传输日志消息示例

进行传输时，会将消息发布到主题字符串为 Log/agent_name/transfer_ID 的 SYSTEM.FTE 主题。以下 XML 示例显示了已启动、正在进行和已完成的单个文件传输的日志消息。

单个文件传输 - 已启动

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d223d0020"
  agentRole="sourceAgent"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-01-26T13:03:26.484Z">started</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </sourceAgent>
  <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
  <originator>
    <hostName>dhcp-9-20-240-199.hursley.ibm.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <transferSet startTime="2011-01-26T13:03:26.484Z" total="1" bytesSent="0">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">dhcp-9-20-240-199.hursley.ibm.com.</
metaData>
    <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d223d0020</
metaData>
    <metaData key="com.ibm.wmqfte.ScheduleId">3</metaData>
    <metaData key="com.ibm.wmqfte.Priority">0</metaData>
  </metaDataSet>
</transferSet>
<scheduleLog ID="3"/>
</transaction>
```

单个文件传输成功 - 正在进行

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d223d0020"
  agentRole="sourceAgent"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-01-26T13:03:26.615Z">progress</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </sourceAgent>
  <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </destinationAgent>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <transferSet index="0" size="1" startTime="2011-01-26T13:03:26.484Z" total="1"
bytesSent="1198">
    <item mode="binary">
      <source disposition="leave" type="file">
        <file size="1151" last-modified="2009-11-02T10:37:01.000Z">/etc/passwd</file>
        <checksum method="MD5">2287181c07199f879de28296371cb24c</checksum>
      </source>
      <destination type="file">
        <file size="1151" last-modified="2011-01-26T13:03:26.000Z">/tmp/passwd</file>
        <checksum method="MD5">2287181c07199f879de28296371cb24c</checksum>
      </destination>
      <status resultCode="0"/>
    </item>
  </transferSet>
</transaction>
```

单个文件传输成功 - 已完成

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d223d0020"
  agentRole="sourceAgent"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-01-26T13:03:26.622Z">completed</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </sourceAgent>
  <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </destinationAgent>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <status resultCode="0">
    <supplement>BFGRP0032I: The file transfer request has successfully completed.</supplement>
  </status>
  <transferSet startTime="2011-01-26T13:03:26.484Z" total="1" bytesSent="1198">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">example.com.</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d223d0020</
metaData>
      <metaData key="com.ibm.wmqfte.ScheduleId">3</metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
  <statistics>
    <actualStartTime>2011-01-26T13:03:26.541Z</actualStartTime>
    <retryCount>0</retryCount>
    <numFileFailures>0</numFileFailures>
    <numFileWarnings>0</numFileWarnings>
  </statistics>
</transaction>
```

相关参考

第 677 页的『触发的传输消息格式』

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。以下 XML 示例显示了在启动一个包含触发条件的文件传输时创建的日志消息。

第 678 页的『用户出口消息格式』

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。该 XML 示例显示了在进行的文件传输包含对用户出口的调用时所创建的日志消息。

第 681 页的『基于 Web 的传输的消息格式中添加的内容』

来自通过 IBM MQ Managed File Transfer Web 网关 请求的传输的 Started 和 Completed 日志消息包含额外的元数据。该元数据包含有关 HTTP 请求以及有关托管 Web 网关的应用程序服务器的信息。

第 662 页的『文件传输日志消息格式』

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 IBM MQ Managed File Transfer 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

多文件传输日志消息示例

进行包含多个文件的传输时发布至 SYSTEM.FTE 主题（含主题字符串 `Log/agent_name/transfer_ID`）的消息示例。

多个文件传输 - 已启动

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
```

```

ID="414d51205553322e42494e44494e47538b0f404d035c0020"
agentRole="sourceAgent"
xsi:noNamespaceSchemaLocation="TransferLog.xsd"
xmlns=""
<action time="2011-01-26T13:12:58.534Z">started</action>
<sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
  <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
</sourceAgent>
<destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
<originator>
  <hostName>example.com</hostName>
  <userID>mqm</userID>
  <mqmdUserID>mqm</mqmdUserID>
</originator>
<transferSet startTime="2011-01-26T13:12:58.534Z" total="6" bytesSent="0">
  <metaDataSet>
    <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
    <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
    <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
    <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
    <metaData key="com.ibm.wmqfte.OriginatingHost">example.com</metaData>
    <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d035c0020</
metaData>
    <metaData key="com.ibm.wmqfte.Priority">0</metaData>
  </metaDataSet>
</transferSet>
</transaction>

```

多个文件传输 - 正在进行

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d035c0020"
  agentRole="sourceAgent"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns=""
  <action time="2011-01-26T13:12:58.753Z">progress</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </sourceAgent>
  <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </destinationAgent>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <transferSet index="0" size="6" startTime="2011-01-26T13:12:58.534Z" total="6" bytesSent="440">
    <item mode="binary">
      <source disposition="leave" type="file">
file>
        <file size="0" last-modified="2011-01-26T13:10:19.000Z"/>srvcnfs/incoming/file01.txt</
        <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
      </source>
      <destination type="file">
file>
        <file size="0" last-modified="2011-01-26T13:12:58.000Z"/>srvcnfs/outgoing/file01.txt</
        <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
      </destination>
      <status resultCode="0"/>
    </item>
    <item mode="binary">
      <source disposition="leave" type="file">
file>
        <file size="0" last-modified="2011-01-26T13:10:19.000Z"/>srvcnfs/incoming/file02.txt</
        <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
      </source>
      <destination type="file">
file>
        <file size="0" last-modified="2011-01-26T13:12:58.000Z"/>srvcnfs/outgoing/file02.txt</
        <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
      </destination>
      <status resultCode="0"/>
    </item>
    <item mode="binary">
      <source disposition="leave" type="file">
        <file size="0" last-modified="2011-01-26T13:10:19.000Z"/>srvcnfs/incoming/file03.txt</

```



```

file>
    <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
  </source>
  <destination type="file">
    <file size="0" last-modified="2011-01-26T13:12:58.000Z">/srv/nfs/outgoing/file03.txt</
file>
    <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
  </destination>
  <status resultCode="0"/>
</item>
<item mode="binary">
  <source disposition="leave" type="file">
    <file size="0" last-modified="2011-01-26T13:10:19.000Z">/srv/nfs/incoming/file04.txt</
file>
    <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
  </source>
  <destination type="file">
    <file size="0" last-modified="2011-01-26T13:12:58.000Z">/srv/nfs/outgoing/file04.txt</
file>
    <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
  </destination>
  <status resultCode="0"/>
</item>
<item mode="binary">
  <source disposition="leave" type="file">
    <file size="0" last-modified="2011-01-26T13:10:19.000Z">/srv/nfs/incoming/file05.txt</
file>
    <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
  </source>
  <destination type="file">
    <file size="0" last-modified="2011-01-26T13:12:58.000Z">/srv/nfs/outgoing/file05.txt</
file>
    <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
  </destination>
  <status resultCode="0"/>
</item>
<item mode="binary">
  <source disposition="leave" type="file">
    <file size="0" last-modified="2011-01-26T13:10:19.000Z">/srv/nfs/incoming/file06.txt</
file>
    <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
  </source>
  <destination type="file">
    <file size="0" last-modified="2011-01-26T13:12:58.000Z">/srv/nfs/outgoing/file06.txt</
file>
    <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
  </destination>
  <status resultCode="0"/>
</item>
</transferSet>
</transaction>

```

多个文件传输 - 已完成

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d035c0020"
  agentRole="sourceAgent"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-01-26T13:12:58.766Z">completed</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </sourceAgent>
  <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </destinationAgent>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <status resultCode="0">
    <supplement>BFGRP0032I: The file transfer request has successfully completed.</supplement>
  </status>
  <transferSet startTime="2011-01-26T13:12:58.534Z" total="6" bytesSent="440">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
    </metaDataSet>
  </transferSet>
</transaction>

```

```

        <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
        <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
        <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
        <metaData key="com.ibm.wmqfte.OriginatingHost">example.com.</metaData>
        <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d035c0020</
metaData>
        <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
</transferSet>
<statistics>
    <actualStartTime>2011-01-26T13:12:58.634Z</actualStartTime>
    <retryCount>0</retryCount>
    <numFileFailures>0</numFileFailures>
    <numFileWarnings>0</numFileWarnings>
</statistics>
</transaction>

```

失败的传输日志消息示例

进行传输时，会将消息发布到主题字符串为 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题。XML 示例显示了传输开始时、传输过程中以及传输完成后失败的文件传输的日志消息。

文件传输失败 - 已启动

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    version="4.00"
    ID="414d51205553322e42494e44494e47538b0f404d03620020"
    agentRole="sourceAgent"
    xsi:noNamespaceSchemaLocation="TransferLog.xsd"
    xmlns="">
    <action time="2011-01-26T13:19:15.767Z">started</action>
    <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
        <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
    </sourceAgent>
    <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
    <originator>
        <hostName>example.com.</hostName>
        <userID>mqm</userID>
        <mqmdUserID>mqm</mqmdUserID>
    </originator>
    <transferSet startTime="2011-01-26T13:19:15.767Z" total="1" bytesSent="0">
        <metaDataSet>
            <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
            <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
            <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
            <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
            <metaData key="com.ibm.wmqfte.OriginatingHost">example.com.</metaData>
            <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d03620020</
metaData>
            <metaData key="com.ibm.wmqfte.Priority">0</metaData>
        </metaDataSet>
    </transferSet>
</transaction>

```

文件传输失败 - 进行中

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    version="4.00"
    ID="414d51205553322e42494e44494e47538b0f404d03620020"
    agentRole="sourceAgent"
    xsi:noNamespaceSchemaLocation="TransferLog.xsd"
    xmlns="">
    <action time="2011-01-26T13:19:15.944Z">progress</action>
    <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
        <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
    </sourceAgent>
    <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
        <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
    </destinationAgent>
    <originator>
        <hostName>example.com.</hostName>
        <userID>mqm</userID>
        <mqmdUserID>mqm</mqmdUserID>
    </originator>

```

```

    <transferSet index="0" size="1" startTime="2011-01-26T13:19:15.767Z" total="1" bytesSent="0">
      <item mode="binary">
        <source disposition="leave" type="file">
          <file size="0" last-modified="2011-01-26T13:10:19.000Z">/srv/nfs/incoming/file01.txt</
file>
          <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
        </source>
        <destination type="file">
          <file>/srv/nfs/outgoing/file01.txt</file>
        </destination>
        <status resultCode="1">
          <supplement>BFGIO0006E: File "/srv/nfs/outgoing/file01.txt" already exists.</
supplement>
        </status>
      </item>
    </transferSet>
  </transaction>

```

文件传输失败 - 已完成

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d03620020"
  agentRole="sourceAgent"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-01-26T13:19:15.948Z">completed</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </sourceAgent>
  <destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
    <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
  </destinationAgent>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <status resultCode="40">
    <supplement>BFGRP0034I: The file transfer request has
      completed with no files being transferred.
    </supplement>
  </status>
  <transferSet startTime="2011-01-26T13:19:15.767Z" total="1" bytesSent="0">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">example.com.</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d03620020</
metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
  <statistics>
    <actualStartTime>2011-01-26T13:19:15.878Z</actualStartTime>
    <retryCount>0</retryCount>
    <numFileFailures>1</numFileFailures>
    <numFileWarnings>0</numFileWarnings>
  </statistics>
</transaction>

```

触发的传输消息格式

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题发布消息。以下 XML 示例显示了在启动一个包含触发条件的文件传输时创建的日志消息。

触发器单个文件传输成功 - 已启动

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d312020202020202020207e970d492000a102" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"

```

相关参考

进行传输时，会将消息发布到主题字符串为 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题。以下 XML 示例显示了已启动、正在进行和已完成的单个文件传输的日志消息。

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题发布消息。该 XML 示例显示了在进行的文件传输包含对用户出口的调用时所创建的日志消息。

来自通过 IBM MQ Managed File Transfer Web 网关 请求的传输的 Started 和 Completed 日志消息包含额外的元数据。该元数据包含有关 HTTP 请求以及有关托管 Web 网关的应用程序服务器的信息。

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 `IBM MQ Managed File Transfer` 安装的 `MQ INSTALLATION PATH/mqft/samples/schema` 目录中。

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。以下 XML 示例显示根据某个安排进行文件传输时创建的日志消息。

当根据安排条目到期而启动文件传输时，文件传输将遵循惯常的顺序在 `SYSTEM.FTE/Log/agent_name` 主题上发布事务消息：

- 只有操作状态为“已开始”的日志事务消息才会在 `<scheduleLog>` 元素的 ID 属性中包含已调度传输的标识。这允许在整个传输的生命周期内将安排标识与传输标识绑定在一起。

IBM MQ Managed File Transfer 介绍 823

进行中:

```
<?xml version="1.0" encoding="UTF-8"?>  
<transaction version="1.00"  
  ID="414d5120514d31202020202020202020248e294920004016" agentRole="sourceAgent"  
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"  
  xmlns="">  
    <action time="2008-11-23T21:55:03.377Z">progress</action>  
    .  
    .  
    .  
</transaction>
```

已完成：

[illegible]

用户出口消息格式

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。该 XML 示例显示了在进行的文件传输包含对用户出口的调用时所创建的日志消息。

继续退出单个文件传输 - 已启动

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d3120202020202020207e970d492000d502" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T22:36:13.046Z">started</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1" />
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
    <mqmdUserID>USER1          </mqmdUserID>
  </originator>
  <transferSet startTime="2008-11-02T22:36:13.046Z" total="1">
    <metaDataSet>
      <metaData key="testkey1">testvalue1</metaData>
      <metaData key="testkey2">testvalue2</metaData>
    </metaDataSet>
  </transferSet>
</transaction>
```

继续退出单个文件传输 - 已完成

```
<?xml version="1.0" encoding="UTF-8"?>  
<transaction version="1.00"  
  ID="414d5120514d31202020202020202020e970d492000d502"  
  agentRole="sourceAgent"  
    xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"  
    xsi:noNamespaceSchemaLocation="TransferLog.xsd"  
    xmlns=""  
    <action time="2008-11-02T22:36:13.546Z">completed</action>  
    <sourceAgent agent="FTEAGENT" QMgr="QM1">  
      <startExits>  
        <exit name="class testExits.SourceExit1">  
          <status resultCode="proceed">
```

```

        <supplement>Source Start, modified metadata</supplement>
      </status>
    </exit>
  </startExits>
</endExits>
  <exit name="class testExits.SourceExit1">
    <status>
      <supplement>Source End</supplement>
    </status>
  </exit>
</endExits>
<systemInfo architecture="x86" name="Windows 7"
  version="6.1 build 7601 Service Pack 1" />
</sourceAgent>
<destinationAgent agent="FTEAGENT" QMgr="QM1">
  <startExits>
    <exit name="class testExits.DestinationExitProceed">
      <status resultCode="proceed">
        <supplement>Destination start, with proceed</supplement>
      </status>
    </exit>
  </startExits>
</endExits>
  <exit name="class testExits.DestinationExitProceed">
    <status>
      <supplement>destination end</supplement>
    </status>
  </exit>
</endExits>
<systemInfo architecture="x86" name="Windows 7"
  version="6.1 build 7601 Service Pack 1" />
</destinationAgent>
<originator>
  <hostName>reportserver.com</hostName>
  <userID>USER1</userID>
  <mqmdUserID>USER1 </mqmdUserID>
</originator>
<transferSet startTime="2008-11-02T22:36:13.046Z" total="1">
  <metaDataSet>
    <metaData key="newkey2">newvalue2</metaData>
    <metaData key="newkey1">newvalue1</metaData>
    <metaData key="newkey4">newvalue4</metaData>
    <metaData key="newkey3">newvalue3</metaData>
    <metaData key="newkey5">newvalue5</metaData>
    <metaData key="testkey1">testvalue1</metaData>
    <metaData key="testkey2">testvalue2</metaData>
  </metaDataSet>
</transferSet>
</transaction>

<!--
  In this example the source transfer start exit has modified the
  metadata as follows:

  Added keys and values for:
  newkey1, newvalue1
  newkey2, newvalue2
  newkey3, newvalue3
  newkey4, newvalue4
  newkey5, newvalue5

  Replaced values for:
  key1 to modifiedValue1

  Deleted keys and values for:
  key2
-->

```

取消退出单个文件传输 - 已取消

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d312020202020202020207e970d492000c702" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T22:25:59.328Z">cancelled</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <startExits>

```

```

        <exit name="class testExits.SourceExit1">
            <status resultCode="proceed">
                <supplement>Source Start, modified metadata</supplement>
            </status>
        </exit>
    </startExits>
</endExits>
    <exit name="class testExits.SourceExit1">
        <status>
            <supplement>Source End</supplement>
        </status>
    </exit>
</endExits>
<systemInfo architecture="x86" name="Windows 7"
    version="6.1 build 7601 Service Pack 1" />
</sourceAgent>
<destinationAgent agent="FTEAGENT" QMgr="QM1">
    <startExits>
        <exit name="class testExits.DestinationExit1">
            <status resultCode="cancelTransfer">
                <supplement>Destination start, with cancel</supplement>
            </status>
        </exit>
    </startExits>
</endExits>
    <exit name="class testExits.DestinationExit1">
        <status>
            <supplement>destination end</supplement>
        </status>
    </exit>
</endExits>
<systemInfo architecture="x86" name="Windows 7"
    version="6.1 build 7601 Service Pack 1" />
</destinationAgent>
<originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
    <mqmdUserID>USER1 </mqmdUserID>
</originator>
<transferSet startTime="2008-11-02T22:25:59.078Z" total="1" />
</transaction>

```

相关参考

第 671 页的『单个传输日志消息示例』

进行传输时，会将消息发布到主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题。以下 XML 示例显示了已启动、正在进行和已完成的单个文件传输的日志消息。

第 677 页的『触发的传输消息格式』

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。以下 XML 示例显示了在启动一个包含触发条件的文件传输时创建的日志消息。

第 681 页的『基于 Web 的传输的消息格式中添加的内容』

来自通过 IBM MQ Managed File Transfer Web 网关请求的传输的 Started 和 Completed 日志消息包含额外的元数据。该元数据包含有关 HTTP 请求以及有关托管 Web 网关的应用程序服务器的信息。

第 662 页的『文件传输日志消息格式』

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 IBM MQ Managed File Transfer 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

基于 Web 的传输的消息格式中添加的内容

来自通过 IBM MQ Managed File Transfer Web 网关请求的传输的 Started 和 Completed 日志消息包含额外的元数据。该元数据包含有关 HTTP 请求以及有关托管 Web 网关的应用程序服务器的信息。

Web 元数据的定义

com.ibm.wmqfte.web.request.authtype

向 Web 网关提交请求的用户使用的授权方法。

com.ibm.wmqfte.web.request.locale

向 Web 网关提交请求的用户的语言环境。

托管 Web 网关的应用程序服务器的类型。

系统的主机名或 IP 地址, 该系统运行托管 Web 网关的应用程序服务器。

托管 Web 网关的应用程序服务器所侦听的端口号。

以下示例中突出显示了通过 Web 网关所请求传输的日志消息中包括的元数据。

单个文件传输 - 成功

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d3120202020202020207e970d4920008202" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T21:20:37.578Z">started</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1" />
  <originator>
    <hostName>requestor.example.com</hostName>
    <userID>USER1 </userID>
    <mqmdUserID>USER1</mqmdUserID>
  </originator>
  <transferSet startTime="2008-11-02T21:20:37.593Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.web.request.authtype">BASIC</metaData>
      <metaData key="com.ibm.wmqfte.web.request.locale">en_GB</metaData>
      <metaData key="com.ibm.wmqfte.web.appsrv.type">
        Apache Geronimo (Embedded Tomcat/6.0.20-20090724)
      </metaData>
      <metaData key="com.ibm.wmqfte.web.appsrv.port">8080</metaData>
      <metaData key="com.ibm.wmqfte.web.appsrv.host">gateway.example.com</metaData>
    </metaDataSet>
  </transferSet>
</transaction>
```

单个文件传输成功 - 已完成

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d3120202020202020207e970d4920008202" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T21:20:38.234Z">completed</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </destinationAgent>
  <originator>
    <hostName>requestor.example.com</hostName>
    <userID>USER1</userID>
    <mqmdUserID>USER1 </mqmdUserID>
  </originator>
  <transferSet startTime="2008-11-02T21:20:37.593Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.web.request.authtype">BASIC</metaData>
      <metaData key="com.ibm.wmqfte.web.request.locale">en_GB</metaData>
      <metaData key="com.ibm.wmqfte.web.appsrv.type">
        Apache Geronimo (Embedded Tomcat/6.0.20-20090724)
      </metaData>
    </metaDataSet>
  </transferSet>
</transaction>
```

```

        <metaData key="com.ibm.wmqfte.web.appsrv.port">8080</metaData>
        <metaData key="com.ibm.wmqfte.web.appsrv.host">gateway.example.com</metaData>
    </metaDataSet>
</transferSet>
</transaction>

```

注：注：此处描述的 XML 消息格式与 Web 网关所返回响应的消息格式不同。以下主题中描述了 Web 网关返回的 XML 格式：第 919 页的『响应格式：XML 和 JSON』。

相关参考

第 671 页的『单个传输日志消息示例』

进行传输时，会将消息发布到主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题。以下 XML 示例显示了已启动、正在进行和已完成的单个文件传输的日志消息。

第 677 页的『触发的传输消息格式』

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。以下 XML 示例显示了在启动一个包含触发条件的文件传输时创建的日志消息。

第 678 页的『用户出口消息格式』

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题发布消息。该 XML 示例显示了在进行的文件传输包含对用户出口的调用时所创建的日志消息。

第 662 页的『文件传输日志消息格式』

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 SYSTEM.FTE 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 IBM MQ Managed File Transfer 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

Connect:Direct 网桥传输消息示例

当目标代理或源代理为 Connect:Direct 网桥代理时，`destinationAgent` 或 `sourceAgent` 元素会包含更多的属性。已开始日志消息只包含一部分有关 Connect:Direct 传输的信息。进行中和已完成日志消息包含有关 Connect:Direct 传输的完整信息。

源代理为 Connect:Direct 网桥代理

已开始：

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    ID="414d5120514d5f696b6b796f20202020a704654d20092507"
    agentRole="sourceAgent"
    version="4.00"
    xsi:noNamespaceSchemaLocation="TransferLog.xsd"
    xmlns="">
    <action time="2011-03-07T13:05:01.838Z">started</action>
    <sourceAgent QMgr="QM_KUIPER" agent="VARUNA" agentType="CD_BRIDGE" bridgeNode="CDNODE_VARUNA">
        <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
    </sourceAgent>
    <destinationAgent QMgr="QM_KUIPER" agent="IXION"/>
    <originator>
        <hostname>kuiper.example.com.</hostname>
        <userID>sol</userID>
        <mqmdUserID>sol</mqmdUserID>
    </originator>
    <transferSet bytesSent="0" startTime="2011-03-07T13:05:01.838Z" total="1">
        <metaDataSet>
            <metaData key="com.ibm.wmqfte.SourceAgent">VARUNA</metaData>
            <metaData key="com.ibm.wmqfte.DestinationAgent">IXION</metaData>
            <metaData key="com.ibm.wmqfte.MqmdUser">sol</metaData>
            <metaData key="com.ibm.wmqfte.OriginatingUser">sol</metaData>
            <metaData key="com.ibm.wmqfte.OriginatingHost">kuiper.example.com.</metaData>
            <metaData key="com.ibm.wmqfte.TransferId">414d5120514d5f696b6b796f20202020a704654d20092507</
metaData>
            <metaData key="com.ibm.wmqfte.Priority">0</metaData>
        </metaDataSet>
    </transferSet>
</transaction>

```

进行中：

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    ID="414d5120514d5f696b6b796f20202020a704654d20092507"

```

```

        agentRole="sourceAgent"
        version="4.00"
        xsi:noNamespaceSchemaLocation="TransferLog.xsd"
        xmlns=""
    <action time="2011-03-07T13:05:03.448Z">progress</action>
    <sourceAgent QMgr="QM_KUIPER" agent="VARUNA" agentType="CD_BRIDGE"
        bridgeNode="CDNODE_VARUNA" pnode="CDNODE_VARUNA" snode="CDNODE_ERIS">
        <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
    </sourceAgent>
    <destinationAgent QMgr="QM_KUIPER" agent="IXION" agentType="STANDARD">
        <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
    </destinationAgent>
    <originator>
        <hostName>kuiper.example.com.</hostName>
        <userID>sol</userID>
        <mqmdUserID>sol</mqmdUserID>
    </originator>
    <transferSet bytesSent="48" index="0" size="1" startTime="2011-03-07T13:05:01.838Z" total="1">
        <item mode="binary">
            <source disposition="leave" processName="f2007567" processNumber="68" type="file">
                <file last-modified="2011-03-07T13:05:02.573Z" size="4">CDNODE_ERIS:D:/AGENTS/
CDNODE_ERIS/test.txt</file>
                <checksum method="MD5">098f6bcd4621d373cade4e832627b4f6</checksum>
            </source>
            <destination type="file">
                <file last-modified="2011-03-07T13:05:03.338Z" size="4">D:\AGENTS\IXION\test.txt</file>
                <checksum method="MD5">098f6bcd4621d373cade4e832627b4f6</checksum>
            </destination>
            <status resultCode="0"/>
        </item>
    </transferSet>
</transaction>

```

已完成:

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    ID="414d5120514d5f696b6b796f20202020a704654d20092507"
    agentRole="sourceAgent"
    version="4.00" xsi:noNamespaceSchemaLocation="TransferLog.xsd"
    xmlns=""
    <action time="2011-03-07T13:05:03.495Z">completed</action>
    <sourceAgent QMgr="QM_KUIPER" agent="VARUNA" agentType="CD_BRIDGE"
        bridgeNode="CDNODE_VARUNA" pnode="CDNODE_VARUNA" snode="CDNODE_ERIS">
        <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
    </sourceAgent>
    <destinationAgent QMgr="QM_KUIPER" agent="IXION" agentType="STANDARD">
        <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
    </destinationAgent>
    <originator>
        <hostName>kuiper.example.com.</hostName>
        <userID>sol</userID>
        <mqmdUserID>sol</mqmdUserID>
    </originator>
    <status resultCode="0">
        <supplement>BFGRP0032I: The file transfer request has successfully completed.</supplement>
    </status>
    <transferSet bytesSent="48" startTime="2011-03-07T13:05:01.838Z" total="1">
        <metaDataSet>
            <metaData key="com.ibm.wmqfte.SourceAgent">VARUNA</metaData>
            <metaData key="com.ibm.wmqfte.DestinationAgent">IXION</metaData>
            <metaData key="com.ibm.wmqfte.MqmdUser">sol</metaData>
            <metaData key="com.ibm.wmqfte.OriginatingUser">sol</metaData>
            <metaData key="com.ibm.wmqfte.OriginatingHost">kuiper.example.com.</metaData>
            <metaData key="com.ibm.wmqfte.TransferId">414d5120514d5f696b6b796f20202020a704654d20092507</
metaData>
            <metaData key="com.ibm.wmqfte.Priority">0</metaData>
        </metaDataSet>
    </transferSet>
    <statistics>
        <actualStartTime>2011-03-07T13:05:02.041Z</actualStartTime>
        <retryCount>0</retryCount>
        <numFileFailures>0</numFileFailures>
        <numFileWarnings>0</numFileWarnings>
    </statistics>
</transaction>

```

目标代理为 Connect:Direct 网桥代理

已开始:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  ID="414d5120514d5f696b6b796f20202020a704654d2008e102"
  agentRole="sourceAgent"
  version="4.00"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-03-07T10:29:44.854Z">started</action>
  <sourceAgent QMgr="QM_ASTEROID" agent="PALLAS" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="QM_ASTEROID" agent="VESTA"/>
  <originator>
    <hostName>belt.example.com.</hostName>
    <userID>sol</userID>
    <mqmdUserID>sol</mqmdUserID>
  </originator>
  <transferSet bytesSent="0" startTime="2011-03-07T10:29:44.854Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">PALLAS</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">VESTA</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">belt.example.com.</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d5120514d5f696b6b796f20202020a704654d2008e102</
metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
</transaction>
```

进行中:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  ID="414d5120514d5f696b6b796f20202020a704654d2008e102"
  agentRole="sourceAgent"
  version="4.00"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-03-07T10:29:46.682Z">progress</action>
  <sourceAgent QMgr="QM_ASTEROID" agent="PALLAS" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="QM_ASTEROID" agent="VESTA" agentType="CD_BRIDGE"
    bridgeNode="CDNODE_VESTA" pnode="CDNODE_VESTA" snode="CDNODE_HYGIEA">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </destinationAgent>
  <originator>
    <hostName>belt.example.com.</hostName>
    <userID>sol</userID>
    <mqmdUserID>sol</mqmdUserID>
  </originator>
  <transferSet bytesSent="48" index="0" size="1" startTime="2011-03-07T10:29:44.854Z" total="1">
    <item mode="binary">
      <source disposition="leave" type="file">
        <file last-modified="2011-03-04T14:53:28.323Z" size="4">D:\AGENTS\PALLAS\test.txt</
file>
        <checksum method="MD5">098f6bcd4621d373cade4e832627b4f6</checksum>
      </source>
      <destination processName="f2006965" processNumber="59" type="file">
        <file size="4">CDNODE_VESTA:D:/AGENTS/CDNODE_VESTA/test.txt</file>
        <checksum method="MD5">098f6bcd4621d373cade4e832627b4f6</checksum>
      </destination>
      <status resultCode="0"/>
    </item>
  </transferSet>
</transaction>
```

已完成:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  ID="414d5120514d5f696b6b796f20202020a704654d2008e102"
  agentRole="sourceAgent">
```

```

        version="4.00"
        xsi:noNamespaceSchemaLocation="TransferLog.xsd"
        xmlns="">
<action time="2011-03-07T10:29:46.698Z">completed</action>
<sourceAgent QMgr="QM_asteroid" agent="PALLAS" agentType="STANDARD">
  <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
</sourceAgent>
<destinationAgent QMgr="QM_asteroid" agent="VESTA" agentType="CD_BRIDGE"
  bridgeNode="CDNODE_VESTA" pNode="CDNODE_VESTA" sNode="CDNODE_HYGIEA">
  <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
</destinationAgent>
<originator>
  <hostName>belt.example.com</hostName>
  <userID>sol</userID>
  <mqmdUserID>sol</mqmdUserID>
</originator>
<status resultCode="0">
  <supplement>BFGRP0032I: The file transfer request has successfully completed.</supplement>
</status>
<transferSet bytesSent="48" startTime="2011-03-07T10:29:44.854Z" total="1">
  <metaDataSet>
    <metaData key="com.ibm.wmqfte.SourceAgent">PALLAS</metaData>
    <metaData key="com.ibm.wmqfte.DestinationAgent">VESTA</metaData>
    <metaData key="com.ibm.wmqfte.MqmdUser">sol</metaData>
    <metaData key="com.ibm.wmqfte.OriginatingUser">sol</metaData>
    <metaData key="com.ibm.wmqfte.OriginatingHost">belt.example.com.</metaData>
    <metaData key="com.ibm.wmqfte.TransferId">414d5120514d5f696b6b796f20202020a704654d2008e102</
metaData>
    <metaData key="com.ibm.wmqfte.Priority">0</metaData>
  </metaDataSet>
</transferSet>
<statistics>
  <actualStartTime>2011-03-07T10:29:45.010Z</actualStartTime>
  <retryCount>0</retryCount>
  <numFileFailures>0</numFileFailures>
  <numFileWarnings>0</numFileWarnings>
</statistics>
</transaction>

```

已安排的传输日志消息格式

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/agent name/schedule ID 主题中）。该消息符合 ScheduleLog.xsd XML 模式。

模式

以下模式描述哪些元素在调度日志 XML 消息中有效。

```

<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>

  <xsd:element name="schedulelog">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="originator" type="hostUserIDType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="action" type="actionType"
          maxOccurs="1" minOccurs="1"/>
        <xsd:element name="schedule" type="scheduleType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceAgent" type="agentType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="destinationAgent" type="agentClientType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="status" type="statusType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="transferSet" type="transferSetType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="job" type="jobType"
          maxOccurs="1" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="ID" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="actionType">

```

```

        <xsd:simpleContent>
          <xsd:extension base="actionEnumType">
            <xsd:attribute name="time" type="xsd:dateTime" use="required" />
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>

      <xsd:simpleType name="actionEnumType">
        <xsd:restriction base="xsd:token">
          <xsd:enumeration value="submit"/>
          <xsd:enumeration value="delete"/>
          <xsd:enumeration value="expire"/>
          <xsd:enumeration value="skipped"/>
        </xsd:restriction>
      </xsd:simpleType>

      <xsd:complexType name="transferSetType">
        <xsd:sequence>
          <xsd:element name="item" type="itemType"
            maxOccurs="unbounded" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="size" type="xsd:int" use="required" />
        <xsd:attribute name="priority" type="priorityType" use="optional" />
      </xsd:complexType>

      <xsd:complexType name="itemType">
        <xsd:sequence>
          <xsd:element name="source" type="fileSourceType"
            maxOccurs="1" minOccurs="1" />
          <xsd:element name="destination" type="fileDestinationType"
            maxOccurs="1" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="mode" type="modeType" use="required" />
        <xsd:attribute name="checksumMethod" type="checksumMethod" use="required" />
      </xsd:complexType>
    </xsd:schema>

```

了解调度日志消息

下面描述了调度日志消息中使用的元素和属性：

<schedulelog>

用于描述单个提交的已调度文件传输的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 的详细说明指定该元素的版本。
标识	已提交的调度文件传输的唯一标识。

<originator>

包含指定请求发起方的元素的组元素。

<hostName>

源文件所在系统的主机名。

<userID>

发起文件传输的用户标识。

<mqmdUserID>

在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<action>

指定要对与 <schedulelog> 元素的 ID 属性匹配的已调度传输执行的操作。该元素必须是以下某个值：

- submit - 新建已调度传输
- delete - 取消调度传输
- expire - 即将处理调度传输条目
- skipped - 由于代理处于脱机状态，因此无法启动调度的传输。当代理变得可用时，将记录该消息以指明跳过了该传输。

属性	描述
time	指定发布日志条目的日期和时间（采用日期时间格式）。

<sourceAgent>

指定源文件所在系统上的代理的名称。

属性	描述
代理程序	指定代理的名称。
QMgr	代理队列管理器的名称。

<destinationAgent>

指定您要向其传输文件的系统上代理的名称。

属性	描述
代理程序	指定代理的名称。
QMgr	代理队列管理器的名称。

<status>

结果代码和补充消息。

<transferSet>

指定一组要一起执行的文件传输。在传输期间，<transferSet> 是包含 <item> 元素的组元素。

属性	描述
size	指定传输项数。
priority	传输的优先级。优先级为 0-9 之间的一个值，其中 0 为最低优先级。缺省优先级为 0，并且缺省情况下，传输使用源代理的优先级。

<item>

包含指定源和目标文件名和位置的元素的组元素。

属性	描述
mode	将传输方式指定为二进制或文本。
checksumMethod	指定生成消息摘要以创建数字签名的散列算法的类型。允许的值包括 MD5 或 none

<source>

包含源系统上文件的 <file> 和 <checksum> 元素的组元素。

属性	描述
recursive	指定当 <source> 元素为目录或包含通配符时，以递归方式传输子目录中的文件。
disposition	指定在 <source> 已成功传输到其目标时要在 <source> 元素上执行的操作。有效选项如下： • leave - 源文件保持不变。 • delete - 在成功传输源文件后将源文件从源系统中删除。

<destination>

包含目标系统上文件的 <file> 和 <checksum> 元素的组元素。

属性	描述
类型	位于目标的文件或目录的类型。有效选项如下： • file - 将文件指定为目标 • directory - 将目录指定为目标 • dataset - 指定 z/OS 数据集作为目标 • PDS - 将 z/OS 分区数据集指定为目标
exist	指定在目标系统上已存在目标文件时执行的操作。有效选项如下： • error - 报告一个错误，并且未传输文件。 • overwrite - 覆盖现有的目标文件。

<file>

指定要传输的文件名称。使用与操作系统格式一致的标准路径，例如 C:/from/here.txt。请勿使用文件 URI。

属性	描述
编码	文本文件传输的编码。
EOL	指定行结束标记符。允许的值包括： • LF - 仅限换行字符 • CRLF - 回车符和换行符序列

<job>

包含用于指定作业详细信息的元素的组元素。<job> is a user-defined job name identifier that is added to the log message when the transfer has started. This <job> element is the same as the <job> element that is included in the transfer request message, which is described in the following topic: [第 842 页的『文件传输请求消息格式』](#)。

<name>

名称可以是任意字符串。

示例

为以下每个已调度传输操作提供了符合该模式的 XML 消息示例：

- [创建已调度传输](#)
- [取消已调度传输](#)
- [使已调度传输到期](#)

根据调度启动的传输与标准传输采用相同的记录方式。要获取根据调度启动的传输的日志消息的示例，请参阅[第 678 页的『已安排的传输日志消息示例』](#)。

相关参考

[第 647 页的『代理状态消息格式』](#)

在创建或启动代理时，该代理会向其协调队列管理器上的 SYSTEM.FTE 主题（在 SYSTEM.FTE/Agents/agent name 主题上）发布其详细信息。

[第 842 页的『文件传输请求消息格式』](#)

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

[第 658 页的『文件传输状态消息格式』](#)

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 SYSTEM.FTE/Transfers/*agent_name*/*transfer ID* 主题中），它符合 TransferStatus.xsd XML 模式。TransferStatus.xsd 文件位于 WMQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

第 662 页的『文件传输日志消息格式』

文件传输日志消息发布在主题字符串为 Log/*agent_name*/*transfer ID* 的 SYSTEM.FTE 主题内。这些消息符合模式 TransferLog.xsd，该模式位于 IBM MQ Managed File Transfer 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

第 859 页的『监视器请求消息格式』

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 fteCreateMonitor 命令或使用 WebSphere MQ Explorer 接口。

第 871 页的『针对安全性的消息格式』

本主题描述发布到协调队列管理器且与安全性相关的消息。

安排日志示例

执行已安排的传输操作时，发布到主题字符串为 Log/*agent_name*/*schedule ID* 的 SYSTEM.FTE 主题的消息示例。

已安排的传输日志消息

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/*agent name*/*schedule ID* 主题中）。该消息符合 ScheduleLog.xsd XML 模式。例如：

```
<?xml version="1.0" encoding="UTF-8"?>
<schedulelog version="1.00" ID="5"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ScheduleLog.xsd">
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
  </originator>
  <action time="2008-11-23T21:32:01Z">submit</action>
  <schedule>
    <submit timebase="admin" timezone="Europe/London">2008-11-23T22:00</submit>
  </schedule>
  <sourceAgent agent="FTEAGENT" QMgr="QM1" />
  <destinationAgent agent="FTEAGENT" QMgr="QM1" />
  <status resultCode="0" />
  <transferSet size="1" priority="0">
    <item mode="binary" checksumMethod="MD5">
      <source recursive="false" disposition="leave">
        <file>c:\sourcefiles\source1.doc</file>
      </source>
      <destination type="file" exist="overwrite">
        <file>c:\destinationfiles\dest1.doc</file>
      </destination>
    </item>
  </transferSet>
</schedulelog>
```

该消息是一个包含以下信息的日志：

- 发起请求的人员
- 提交请求的时间
- 已安排传输的启动时间
- 源和目标代理详细信息
- 传输规范

<schedulelog> 元素的 ID 属性是该已调度传输的唯一标识（在源代理中）。该标识用于将安排条目与实际的文件传输关联。

提交的 <action> 元素值确认已收到请求。

已安排的传输取消日志消息

代理收到取消暂挂的已安排文件传输的请求时，会将以下消息发布到 `SYSTEM.FTE/Log/agent_name` 主题：

```
<?xml version="1.0" encoding="UTF-8"?>
<schedulelog version="1.00" ID="5"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ScheduleLog.xsd">
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
  </originator>
  <action time="2008-11-23T21:56:27Z">delete</action>
  <status resultCode="0" />
</schedulelog>
```

ID 属性值对应于安排消息中暂挂的传输请求的标识。

已安排的传输到期日志消息

当前时间与调度列表中最早的暂挂文件传输的时间匹配（由 `<next>` 元素的值指示）时，将发布调度日志消息以指示已调度传输条目已到期：

```
<?xml version="1.0" encoding="UTF-8"?>
<schedulelog xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00" ID="3"
  xsi:noNamespaceSchemaLocation="ScheduleLog.xsd">
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <action time="2011-01-26T13:03:26Z">expire</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
  <status resultCode="0"/>
</schedulelog>
```

“expire”的 `<action>` 元素值确认调度条目现在已从调度列表中移除，并且即将进行处理。将发布代理的安排消息，不再存在到期条目。

相关参考

第 686 页的『已安排的传输日志消息格式』

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 `SYSTEM.FTE/Log/agent name/schedule ID` 主题中）。该消息符合 `ScheduleLog.xsd` XML 模式。

第 678 页的『已安排的传输日志消息示例』

在进行传输的过程中，会向具有主题字符串 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题发布消息。以下 XML 示例显示根据某个安排进行文件传输时创建的日志消息。

监视器日志消息格式

监视器日志消息发布至 `SYSTEM.FTE` 主题（含主题字符串 `Log/agent_name/Monitors/monitor_name/monitor_ID`）。

如果要收集数据或查看监视器操作，请将预订设置为适合您感兴趣的监视器的通配符主题。例如：

```
Log/#
```

或者，

```
Log/agent_name/#
```

该预订可以是持久或非持久的。持久预订在预订应用程序与队列管理器之间的连接关闭后继续存在。非持久预订只有在预订应用程序与队列管理器之间的连接处于打开状态时才存在。

MonitorLog.xsd 模式文档位于 *MQ_INSTALLATION_PATH/mqft/samples/schema* 目录中。
MonitorLog.xsd 模式导入位于同一目录中的 fteutils.xsd。

模式

以下模式描述了监视器日志 XML 消息中哪些元素有效。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>
  <xsd:element name="monitorLog">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="originator" type="hostUserIDType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="references" type="referencesType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="action" type="monitorActionType" maxOccurs="1"
minOccurs="1"/>
        <xsd:element name="monitorAgent" type="agentType" maxOccurs="1"
minOccurs="1"/>
        <xsd:element name="status" type="statusType" maxOccurs="1"
minOccurs="1"/>
        <xsd:element name="monitorMetaData" type="monitorMetaDataType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="monitorExits" type="exitGroupType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="jobDetails" type="jobType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="taskXMLRequest" type="taskXMLRequestType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="monitorXMLRequest" type="monitorXMLRequestType"
maxOccurs="1" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="monitorName" type="xsd:string" use="required"/>
      <xsd:attribute name="referenceId" type="xsd:string" use="optional"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="monitorActionType">
    <xsd:simpleContent>
      <xsd:extension base="monitorActionEnumType">
        <xsd:attribute name="time" type="xsd:dateTime" use="required" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:simpleType name="monitorActionEnumType">
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="create"/>
      <xsd:enumeration value="delete"/>
      <xsd:enumeration value="start"/>
      <xsd:enumeration value="stop"/>
      <xsd:enumeration value="triggerSatisfied"/>
      <xsd:enumeration value="triggerNotSatisfied"/>
      <xsd:enumeration value="triggerFail"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="monitorMetaDataType">
    <xsd:sequence>
      <xsd:element name="originalMetaData" type="metaDataSetType" maxOccurs="1" minOccurs="0"/>
      <xsd:element name="updatedMetaData" type="metaDataSetType" maxOccurs="unbounded"
minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="taskXMLRequestType">
    <xsd:sequence>
      <xsd:element name="originalRequest" type="xsd:string" maxOccurs="1" minOccurs="1"/>
      <xsd:element name="updatedRequest" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="taskId" type="xsd:string" use="required"/>
  </xsd:complexType>
  <xsd:complexType name="referencesType">
    <xsd:sequence>
```

```

        <xsd:element name="createRequest" type="xsd:string" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="taskRequest" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="monitorXMLRequestType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="type" type="xmlContentEnumType" use="required" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="xmlContentEnumType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="escapedXML" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

了解监视器日志消息

以下列表中描述了监视器日志消息中使用的元素和属性：

<monitorLog>

组元素，包含描述监视器已执行的操作的元素。

属性	描述
version	必需。监视器列表消息格式的版本。
monitorName	必需。监视器的名称。对于定义监视器所在的代理是唯一的。
referenceId	监视器操作的标识。

<originator>

包含指定请求发起方的元素的组元素。

<hostName>

源文件所在系统的主机名。

<userID>

发起文件传输的用户标识。

<mqmdUserID>

可选。在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<references>

与该监视器操作相关联的其他消息的标识的引用。

<createRequest>

用于创建监视器的 XML 请求消息的消息标识。

<taskRequest>

由于此操作，监视器提交的 XML 请求消息的消息标识。

<action>

发生的与该日志消息相关联的操作。元素内的值可以是以下某个值：create、delete、start、stop、triggerSatisfied、triggerNotSatisfied 或 triggerFail。

<monitorAgent>

正在监视资源的代理。

属性	描述
代理程序	必需。代理的名称。
QMgr	可选。代理所连接的队列管理器的名称。
bridgeURL	可选。如果代理是协议网桥代理，那么为协议服务器的 URL。

<status>

正在记录的资源监视器操作的状态。

属性	描述
resultCode	必需。 操作的整数结果代码。

<supplement>

有关正在记录的资源监视器操作的状态的其他信息。

<monitorMetaData>

包含 <originalMetaData> 和 <updatedMetaData> 元素的组元素。

<originalMetaData>

包含一个或多个描述操作发生前的监视器元数据的 <metadata> 元素。

<updatedMetaData>

包含一个或多个描述操作发生后的监视器元数据的 <metadata> 元素。

<metadata>

定义元数据键值对。 键是元素的属性；值是元素的内容。

属性	描述
key	元数据的键。

<monitorExits>

包含一个或多个 <exit> 元素的组元素。

<exits>

描述资源监视器运行的出口的元素。

属性	描述
名	必需。 资源监视器出口的名称。

<status>

正在记录的资源监视器出口的状态。

属性	描述
resultCode	必需。 出口的整数结果代码。

<supplement>

有关正在记录的资源监视器出口的状态的其他信息。

<jobDetails>

包含一个 <name> 元素的元素。

<name>

作业的名称。

<taskXMLRequest>

包含 <originalRequest> 和 <updatedRequest> 元素的组元素。

属性	描述
taskId	任务请求消息的标识。

<originalRequest>

包含监视器执行的任务的转义 XML 请求消息的元素。

<updatedRequest>

包含监视器执行的任务的更新转义 XML 请求消息的元素。

<monitorXMLRequest>

监视器 XML 请求。

属性	描述
类型	必需。 <monitorXMLRequest> 元素内的监视器 XML 请求数据的格式。唯一有效的值是 escapedXML。

示例

针对以下每项监视器操作提供了符合该模式的 XML 消息的示例：

- 已创建监视器
- [监视器轮询资源时满足监视器的条件](#)
- [监视器轮询资源时不满足监视器的条件](#)
- 已删除监视器

相关参考

第 694 页的『监视器日志示例』

监视操作发生时发布至 SYSTEM.FTE 主题（含主题字符串 `Log/agent_name/monitor_ID`）的消息示例。

监视器日志示例

监视操作发生时发布至 SYSTEM.FTE 主题（含主题字符串 `Log/agent_name/monitor_ID`）的消息示例。

监视器创建的日志消息

```
<?xml version="1.0" encoding="UTF-8"?>
<monitorLog xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  monitorName="MONITORTWO"
  referenceId="414d51205553322e42494e44494e47538b0f404d04410020"
  xsi:noNamespaceSchemaLocation="MonitorLog.xsd">
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <references>
    <createRequest>414d51205553322e42494e44494e47538b0f404d04410020</createRequest>
  </references>
  <action time="2011-01-26T12:41:24Z">start</action>
  <monitorAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
  <status resultCode="0"/>
</monitorLog>
```

符合监视器条件的日志消息

```
<?xml version="1.0" encoding="UTF-8"?>
<monitorLog xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  monitorName="MONITORONE"
  referenceId="414d51205553322e42494e44494e47538b0f404d09430020"
  xsi:noNamespaceSchemaLocation="MonitorLog.xsd">
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <references>
    <createRequest>414d51205553322e42494e44494e47538b0f404d09430020</createRequest>
  </references>
  <action time="2011-01-26T12:56:46Z">triggerSatisfied</action>
  <monitorAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
  <status resultCode="0"/>
  <monitorMetaData>
    <originalMetaData>
      <metaData key="AGENTNAME">AGENT_JUPITER</metaData>
      <metaData key="LASTMODIFIEDDATEUTC">2011-01-26</metaData>
      <metaData key="CURRENTTIMESTAMPUTC">20110126125646793</metaData>
    </originalMetaData>
  </monitorMetaData>
</monitorLog>
```



```

        <metaData key="CURRENTTIMESTAMP">20110126125646793</metaData>
        <metaData key="LASTMODIFIEDDATE">2011-01-26</metaData>
        <metaData key="FILENAME">new.completed</metaData>
        <metaData key="LASTMODIFIEDTIMEUTC">12.56</metaData>
        <metaData key="LASTMODIFIEDTIME">12.56</metaData>
        <metaData key="FILESIZE">0</metaData>
        <metaData key="FILEPATH">/srv/nfs/incoming/new.completed</metaData>
    </originalMetaData>
    <updatedMetaData>
        <metaData key="AGENTNAME">AGENT_JUPITER</metaData>
        <metaData key="LASTMODIFIEDDATEUTC">2011-01-26</metaData>
        <metaData key="CURRENTTIMESTAMPUTC">20110126125646793</metaData>
        <metaData key="CURRENTTIMESTAMP">20110126125646793</metaData>
        <metaData key="LASTMODIFIEDDATE">2011-01-26</metaData>
        <metaData key="FILENAME">new.completed</metaData>
        <metaData key="LASTMODIFIEDTIMEUTC">12.56</metaData>
        <metaData key="LASTMODIFIEDTIME">12.56</metaData>
        <metaData key="FILESIZE">0</metaData>
        <metaData key="FILEPATH">/srv/nfs/incoming/new.completed</metaData>
    </updatedMetaData>
</monitorMetaData>
<taskXMLRequest taskId="null">
    <originalRequest>&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;&lt;request
        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="4.00"
        xsi:noNamespaceSchemaLocation="FileTransfer.xsd"&gt;&lt;&lt;managedTransfer&gt;
            &lt;&lt;originator&gt;&lt;&lt;hostName&gt;example.com.&lt;/hostName&gt;
            &lt;&lt;userID&gt;mqm&lt;/userID&gt;&lt;/originator&gt;
            &lt;&lt;sourceAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/&gt;
            &lt;&lt;destinationAgent QMgr="QM_JUPITER" agent="AGENT_SATURN"/&gt;
            &lt;&lt;transferSet&gt;&lt;&lt;item checksumMethod="MD5" mode="binary"&gt;
            &lt;&lt;source disposition="leave" recursive="false"&gt;
            &lt;&lt;file&gt;/srv/nfs/incoming/*.txt&lt;/file&gt;&lt;/source&gt;
            &lt;&lt;destination exist="error" type="directory"&gt;
            &lt;&lt;file&gt;/srv/backup&lt;/file&gt;&lt;/destination&gt;
            &lt;&lt;/item&gt;&lt;/transferSet&gt;&lt;/managedTransfer&gt;&lt;/request&gt;
        &lt;/originalRequest>
    <updatedRequest>&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;&lt;request
        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="4.00"
        xsi:noNamespaceSchemaLocation="FileTransfer.xsd"&gt;&lt;&lt;managedTransfer&gt;
            &lt;&lt;originator&gt;&lt;&lt;hostName&gt;example.com.&lt;/hostName&gt;
            &lt;&lt;userID&gt;mqm&lt;/userID&gt;&lt;/originator&gt;
            &lt;&lt;sourceAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/&gt;
            &lt;&lt;destinationAgent QMgr="QM_JUPITER" agent="AGENT_SATURN"/&gt;
            &lt;&lt;transferSet&gt;&lt;&lt;item checksumMethod="MD5" mode="binary"&gt;
            &lt;&lt;source disposition="leave" recursive="false"&gt;
            &lt;&lt;file&gt;/srv/nfs/incoming/*.txt&lt;/file&gt;
            &lt;&lt;/source&gt;&lt;&lt;destination exist="error" type="directory"&gt;
            &lt;&lt;file&gt;/srv/backup&lt;/file&gt;&lt;/destination&gt;
            &lt;&lt;/item&gt;&lt;/transferSet&gt;&lt;/managedTransfer&gt;&lt;/request&gt;
        &lt;/updatedRequest>
    </taskXMLRequest>
</monitorLog>

```

不符合监视器条件的日志消息

```

<?xml version="1.0" encoding="UTF-8"?>
<monitorLog xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    version="4.00"
    monitorName="MONITORONE"
    referenceId="414d5120553322e42494e44494e47538b0f404d09430020"
    xsi:noNamespaceSchemaLocation="MonitorLog.xsd">
    <originator>
        <hostName>example.com.</hostName>
        <userID>mqm</userID>
        <mqmdUserID>mqm</mqmdUserID>
    </originator>
    <references>
        <createRequest>414d5120553322e42494e44494e47538b0f404d09430020</createRequest>
    </references>
    <action time="2011-01-26T12:58:46Z">triggerNotSatisfied</action>
    <monitorAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
    <status resultCode="0"/>
</monitorLog>

```

监视器删除的日志消息

```
<?xml version="1.0" encoding="UTF-8"?>
<lst:monitorList xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:lst="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
  version="4.00"
  agent="AGENT_JUPITER"
  monitor="MONITORONE"
  xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition
Monitorlist.xsd">
  <status state="deleted"/>
  <configuration>
    <description/>
    <resources>
      <directory recursionLevel="0" id="">/srv/nfs/incoming</directory>
    </resources>
    <triggerMatch>
      <conditions>
        <condition>
          <name/>
          <resource id=""/>
          <fileMatch>
            <pattern>*.completed</pattern>
          </fileMatch>
        </condition>
      </conditions>
    </triggerMatch>
    <tasks>
      <task>
        <name/>
        <description/>
        <taskXML><?xml version="1.0" encoding="UTF-8"?><request
          xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="4.00"
          xsi:noNamespaceSchemaLocation="FileTransfer.xsd"><managedTransfer>
            <originator><hostName>example.ibm.com.</hostName>
            <userID>mqm</userID></originator>
            <sourceAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/>
            <destinationAgent QMgr="QM_JUPITER" agent="AGENT_SATURN"/>
            <transferSet><item checksumMethod="MD5" mode="binary">
              <source disposition="leave" recursive="false">
                <file>/srv/nfs/incoming/*.txt</file></source>
                <destination exist="error" type="directory">
                  <file>/srv/backup</file></destination>
                </item></transferSet></managedTransfer></request>
            </taskXML>
          </task>
        </tasks>
      </configuration>
      <pollInterval units="minutes">1</pollInterval>
      <batch maxSize="1"/>
    </lst:monitorList>
```

文件传输请求消息格式

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

文件传输消息可以包含以下三个根元素之一：

- <request> - 用于新文件传输请求、受管调用请求或删除暂挂的调度传输
- <cancel> - 用于取消正在进行的文件传输
- <transferSpecifications> - for specifying multiple transfer file groups, used by the **fteCreateTransfer** command

For information about specifying multiple transfer groups by using the <transferSpecifications> element, see [使用传输定义文件](#).

模式

以下模式描述在传输请求 XML 消息中有效的元素。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
<xsd:include schemaLocation="fteutils.xsd"/>

<!--
  Defines the request of a managed transfer and version number
  <request version="1.00" ...
    <managedTransfer>
      ...
    </managedTransfer>
  </request>
-->
<xsd:element name="request">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element name="managedTransfer" type="managedTransferType"/>
      <xsd:element name="deleteScheduledTransfer" type="deleteScheduledTransferType" />
      <xsd:element name="managedCall" type="managedCallType"/>
    </xsd:choice>
    <xsd:attribute name="version" type="versionType" use="required" />
  </xsd:complexType>
</xsd:element>

<!--
  Defines the cancel request of a managed transfer and version number
  <cancel version="1.00"
    xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
    <originator>
      <hostName>myMachine</hostName>
      <userID>myUserId</userID>
    </originator>      - Delete a scheduled transfer.

    <transfer>
      Transfer ID to Cancel
    </transfer>
  </cancel>
-->
<xsd:element name="cancel">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="originator" type="hostUserIDType" maxOccurs="1" minOccurs="1" />
      <xsd:choice>
        <xsd:element name="transfer" type="IDType" maxOccurs="1" minOccurs="1" />
        <xsd:element name="call" type="IDType" maxOccurs="1" minOccurs="1" />
      </xsd:choice>
      <xsd:element name="reply" type="replyType" maxOccurs="1" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="version" type="versionType" use="required" />
  </xsd:complexType>
</xsd:element>

<!--
  Defines the transfer definition element structure.
  <transferSpecifications>
    <item ...
    <item ...
  </transferSpecifications>
-->
<xsd:element name="transferSpecifications">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="item" type="itemType" minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--
  Define a managed transfer of an instigator and request
  <managedTransfer>

    <originator>
      ...
    </originator>

    <schedule>
```

```

        <submit timebase="source"|"UTC">2008-12-07T16:07</submit>
        <repeat>
            <frequency interval="hours">2</frequency>
            <expireTime>2008-12-0816:07</expireTime>
        </repeat>
    </schedule>

    <sourceAgent agent="here" QMgr="near" />
    <destinationAgent agent="there" QMgr="far" />

    <trigger>
        ...
    </trigger>

    <transferSet>
        ...
    </transferSet>
</managedTransfer>
-->

<xsd:complexType name="managedTransferType">
    <xsd:sequence>
        <xsd:element name="originator" type="origTransferRequestType" maxOccurs="1"
minOccurs="1"/>
        <xsd:element name="schedule" type="scheduleType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceAgent" type="agentType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="destinationAgent" type="agentClientType" maxOccurs="1" minOccurs="1" />
        <xsd:element name="trigger" type="triggerType" maxOccurs="1" minOccurs="0" />
        <xsd:element name="reply" type="replyType" maxOccurs="1" minOccurs="0" />
        <xsd:element name="transferSet" type="transferSetType" maxOccurs="1" minOccurs="1" />
        <xsd:element name="job" type="jobType" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<!--
    This is a modified form of origRequestType which is used on a managed transfer request.
    The hostName and userID are mandatory attributes in this case.
-->
<xsd:complexType name="origTransferRequestType">
    <xsd:sequence>
        <xsd:element name="hostName" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="userID" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="mqmdUserID" type="xsd:string" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="webBrowser" type="xsd:string" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="webUserID" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

<!--
    Defines the transferset of source and destination agent and one or more files
    <transferset priority="1">
        <metaDataSet>
            <metaData key="keyname">keyvalue</metaData>
            <metaData key="keyname">keyvalue</metaData>
        </metaDataSet>

        <item>
            ...
        </item>
    </transferset>
-->
<xsd:complexType name="transferSetType">
    <xsd:sequence>
        <xsd:element name="metaDataSet" type="metaDataSetType" maxOccurs="1" minOccurs="0" />
        <xsd:element name="preSourceCall" type="commandActionType" maxOccurs="1"
minOccurs="0" />
        <xsd:element name="postSourceCall" type="commandActionType" maxOccurs="1"
minOccurs="0" />
        <xsd:element name="preDestinationCall" type="commandActionType" maxOccurs="1"
minOccurs="0" />
        <xsd:element name="postDestinationCall" type="commandActionType" maxOccurs="1"
minOccurs="0" />
        <xsd:element name="item" type="itemType" maxOccurs="unbounded" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="priority" type="priorityType" use="optional" />
</xsd:complexType>

<!--
    Define a file pair with source and destination
    <item mode=[binary|text]>
        <source recursive="false" disposition="leave">

```

```

        <file>filename</file>
    </source>

    <destination type="file" exist="error">
        <file>filename</file>
    </destination>

</item>
-->
<xsd:complexType name="itemType">
    <xsd:sequence>
        <xsd:element name="source" type="fileSourceType" maxOccurs="1" minOccurs="1" />
        <xsd:element name="destination" type="fileDestinationType" maxOccurs="1" minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="mode" type="modeType" use="required" />
    <xsd:attribute name="checksumMethod" type="checkSumMethod" use="required" />
</xsd:complexType>

<!--
    Defines the request to delete scheduled file transfer.
    <deleteScheduledTransfer>
        <originator>
            <delete>
                <hostName>myMachine</hostName>
                <userID>myUserID</userID>
            </delete>
        </originator>
        <ID>56</ID>
    </deleteScheduledTransfer>
-->
<xsd:complexType name="deleteScheduledTransferType">
    <xsd:sequence>
        <xsd:element name="originator" type="origDeleteType" maxOccurs="1" minOccurs="1" />
        <xsd:element name="ID" type="idType" maxOccurs="1" minOccurs="1" />
        <xsd:element name="reply" type="replyType" maxOccurs="1" minOccurs="0" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="managedCallType">
    <xsd:sequence>
        <xsd:element name="originator" type="origRequestType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="agent" type="agentType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="reply" type="replyType" maxOccurs="1" minOccurs="0" />
        <xsd:element name="transferSet" type="callTransferSetType" maxOccurs="1" minOccurs="1" />
        <xsd:element name="job" type="jobType" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="callTransferSetType">
    <xsd:sequence>
        <xsd:element name="metaDataSet" type="metaDataSetType" maxOccurs="1" minOccurs="0" />
        <xsd:element name="call" type="commandActionType" maxOccurs="1" minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="priority" type="priorityType" use="optional" />
</xsd:complexType>
</xsd:schema>

```

了解传输请求消息

以下列表描述了在传输请求消息中使用的元素和属性：

元素描述

<request>

包含指定文件传输请求所需的所有元素的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 提供的该元素的版本，指定该值。

<managedTransfer>

包含单个文件传输或一组文件传输所需的所有元素的组元素。

<deleteScheduledTransfer>
包含发起方和标识信息以供取消调度传输的组元素。

<managedCall>
包含程序或可执行文件的单个受管调用所需的所有元素的组元素。

<ID>
指定要从暂挂的调度传输列表删除的传输请求的唯一标识。

<originator>
包含指定请求发起方的元素的组元素。

<hostName>
源文件所在系统的主机名。

<userID>
发起文件传输的用户标识。

<mqmdUserID>
可选。在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<schedule>
描述文件传输的调度时间、重复行为以及下一次出现的到期时间的组元素。

<submit>
指定已调度传输应启动的日期和时间。

属性	描述
timebase	指定要使用的时区。此属性可以具有下列其中一个值： • source - 使用源代理的时区 • admin - 使用发出命令的管理员的时区 • UTC - 使用全球标准时间
时区	依据 timebase 值的时区描述

<repeat>
包含有关已调度传输重复频率、已调度传输重复次数以及已调度传输何时停止重复的详细信息的一组元素。

<frequency>
重复传输之前必须等待的时间段。

属性	描述
区间 (interval)	时间间隔单位，必须是以下值之一： • 分钟 • 小时 • days • weeks • months • years

<expireTime>
指定重复的已调度传输停止的日期和时间的可选元素。该元素与 <expireCount> 元素互斥。

<expireCount>

指定在停止前发生已调度文件传输的次数的可选元素。该元素与 <expireTime> 元素互斥。

<sourceAgent>

指定源文件所在系统上的代理的名称。

属性	描述
代理程序	指定代理的名称。
QMGr	代理队列管理器的名称。

<destinationAgent>

指定您要向其传输文件的系统上代理的名称。

属性	描述
代理程序	指定代理的名称。
QMGr	代理队列管理器的名称。
hostName	代理队列管理器的主机名或 IP 地址。
portNumber	用于到目标代理队列管理器的客户机连接的端口号。
通道	用于连接到目标代理队列管理器的通道名称。

<trigger>

指定在文件传输发生时必须满足的条件的可选元素。

属性	描述
日志	指示是否记录触发失败的标志。有效值如下： • yes - 为失败的已触发传输创建日志条目 • no - 不为失败的已触发传输来创建日志条目

<fileExist>

指定位于与源代理相同系统上的文件名的以逗号分隔的列表。如果此名称列表中的文件满足触发器条件，那么将发生传输。该元素与 <fileSize> 元素互斥。

属性	描述
comparison	指示如何针对名称列表对源文件名求值。有效值如下： • = 名称列表中必须至少有一个文件名匹配 • != 名称列表中至少有一个文件不存在
值	指示比较类型： • exist: 文件必须存在

<fileSize>

指定位于与源代理相同系统上的文件名的以逗号分隔的列表。如果此名称列表中的文件满足触发器条件，那么将发生传输。该元素与 <fileExist> 元素互斥。

属性	描述
comparison	指示如何针对名称列表对源文件名求值。有效值如下： • >= 名称列表中存在一个文件名，并且至少为值属性中指定的最小大小

属性	描述
值	指定为整数值的文件大小，带有指定为以下之一的单位： • B - 字节 • KB - 千字节 • MB - 兆字节 • GB - 千兆字节 (单位值不区分大小写)

<reply>

指定为同步文件传输生成的临时应答队列的名称（在命令行上使用 **-w** 参数指定）。队列的名称由 `command.properties` 配置文件中的键 **dynamicQueuePrefix** 或缺省值 `WMQFTE.*` 定义。（如果未指定）。

属性	描述
详细信息	回复消息中是否需要详细传输结果信息。可以针对每个传输生成多条回复消息。有效值如下： • true - 需要详细回复信息。信息的格式与进度消息中发布到传输日志的相同，即 <code><transferSet></code> 元素。有关更多信息，请参阅第 662 页的『文件传输日志消息格式』。仅当传输源代理的 <code>enableDetailedReplyMessages</code> 属性设置为 <code>true</code> 时，才会显示详细回复信息。 • false - 不需要详细回复信息。 缺省值是 <code>false</code> 。
QMGR	在其上生成临时动态队列以接收应答的命令队列管理器的名称。
persistent	写入应答队列的消息是否为持久性消息。有效值如下： • true - 消息持久 • false - 消息不持久 • qdef - 消息的持久性由应答队列的属性定义 缺省值是 <code>false</code> 。

<transferSet>

指定想要一起执行的一组文件传输或想要一起执行的一组受管调用。在传输期间，`<transferSet>` 是包含 `<item>` 元素的组元素。

属性	描述
priority	传输的优先级。优先级为 0-9 之间的一个值，其中 0 为最低优先级。缺省优先级为 0，并且缺省情况下，传输使用源代理的优先级。

<metaDataSet>

包含一个或多个元数据项的可选组元素。

<metaData>

指定将传递给代理调用的出口点的用户定义的元数据。元素包含元数据值作为字符串。

属性	描述
key	作为字符串的元数据名称

<call>

包含指定要调用的程序或可执行文件的 `<command>` 元素的组元素。

<command>

指定要调用的程序或可执行文件。命令必须位于代理命令路径上。有关更多信息，请参阅第 592 页的表 50。该元素可以包含可选的 <argument> 元素。

属性	描述
名	命令的名称。
successRC	该命令返回的成功返回码。缺省值为 0。
retryCount	命令失败时将重试的次数。
retryWait	命令重试之间等待的时间（秒）。
类型	要调用的程序的类型。有效值为 antscript、jcl 或 executable。

<argument>

指定要传递给命令的参数。

<item>

包含指定源和目标文件名和位置的元素的组元素。

属性	描述
mode	指定传输方式为二进制还是文本。
checksumMethod	指定生成消息摘要以创建数字签名的散列算法的类型。有效值为 MD5 或 none。

<source>

指定源系统上的文件以及在传输完成后是否移除这些文件的组元素

属性	描述
recursive	指定当 <source> 元素为目录或包含通配符时，以递归方式传输子目录中的文件。
disposition	指定在 <source> 已成功传输到其目标时要在 <source> 元素上执行的操作。有效值如下： • leave - 源文件保持不变。 • delete - 在成功传输源文件后将源文件从源系统中删除。

<file>

指定传输源。对于分布式平台和 IBMi，传输源可以是文件或目录名称。对于 z/OS 平台，传输源可以是文件、目录、数据集或 PDS 名称。使用与操作系统格式一致的标准路径，例如 C:/from/here.txt。请勿使用文件 URI。

属性	描述
别名	指定源文件的别名。该别名是源文件的名称，不包括针对传输指定的任何目录路径。
EOL	指定文本传输的行尾标记符。有效值包括： • LF - 仅限换行字符 • CRLF - 回车符和换行符序列
编码	文本文件传输的源文件编码。
delimiter	指定在面向记录的源文件中的记录之间包含的定界符，例如，z/OS 数据集。将定界符值指定为 00-FF 范围内的两个十六进制数字，前缀为 x。例如，x12 或 x03,x7F。

属性	描述
delimiterType	指定目标文件中包含的位于单条消息数据之后的定界符类型。有效值如下： binary - 十六进制定界符 仅当启用了 V7.0.4.1 功能时，该属性才可用。
delimiterPosition	指定在将面向记录的源文件记录写入常规文件时插入定界符的位置。有效值如下： - prefix - 定界符将插入目标文件中，在每条面向源记录的文件记录中的数据之前。 - postfix - 定界符将插入目标文件中，在每条面向源记录的文件记录的数据之后。
includeDelimiterInFile	指定是否在面向记录的源文件中的记录之间包含定界符。
keepTrailingSpaces	指定在文本方式传输过程中是否保留从固定长度格式的数据集读取的源记录上的尾部空格。缺省值为除去尾部空格。有效值如下： - true - 在从固定长度格式的数据集读取的源记录上保留尾部空格 - false - 除去从固定长度格式的数据集读取的源记录上的尾部空格

<queue>

在用于 <source> 元素时，指定作为传输源的队列的名称，这必须位于源代理队列管理器上。使用格式 *QUEUE*。请勿包含队列管理器名称，队列必须位于源代理队列管理器上。如果已经在 <destination> 元素中使用 <queue> 元素，那么将无法在 <source> 元素中使用该元素。

属性	描述
useGroups	指定是否仅传输源队列中第一个完整的消息组。有效值如下： - true - 仅传输第一个完整的消息组 - false - 传输源队列上的所有消息
groupId	指定要从源队列读取的消息组。仅当 useGroups 属性的值为 true 时，该属性才有效。
delimiterType	指定目标文件中包含的位于单条消息数据之后的定界符类型。有效值如下： - text-文本或 Java 文字定界符 - binary - 十六进制定界符
delimiter	指定在目标文件的单条消息数据之间包含的定界符。
delimiterPosition	指定目标文件包含的定界符是位于单个消息数据之前还是之后。有效值如下： - prefix - 定界符包含在数据之前 - postfix - 定界符包含在数据之后
编码	指定源队列编码。
waitTime	指定源代理等待以下一个条件的时间（秒）： - 在源队列上出现一条消息（如果队列为空或已变为空） - 在源队列上出现一个完整组（如果 useGroups 属性已设置为 true）

属性	描述
	有关设置 waitTime 值的信息，请参阅第 744 页的『用于对“消息到文件”传输指定等待时间的指南』。

<destination>

指定文件位于目标代理上时的目标和行为的组元素。

只能指定 <file> 和 <queue> 之一作为目标的子元素。

属性	描述
类型	目标的类型。有效值如下： • file - 将文件指定为目标 • directory - 将目录指定为目标 • dataset - 指定 z/OS 数据集作为目标 • pds - 指定 z/OS 分区数据集作为目标 • queue - 指定 WebSphere MQ 队列作为目标 • filespace- 指定文件空间作为目标 仅当 <destination> 元素的子元素为 <queue>时，值 队列 才有效。 仅当 <destination> 元素的子元素为 <filespace>时，值 文件空间 才有效。 仅当 <destination> 元素具有子元素 <file>时，其他值才有效。
exist	指定在目标系统上已存在目标文件时执行的操作。有效值如下： • error - 报告一个错误，并且未传输文件。 • overwrite - 覆盖现有的目标文件。 如果 <destination> 元素具有子元素 <queue> 或 <filespace>，那么此属性无效。

<file>

指定先前描述的 <destination> 元素的其他设置。使用与操作系统格式一致的标准路径，例如 C:/from/here.txt。使用与操作系统格式一致的标准路径，例如 C:/from/here.txt。请勿使用文件 URI。

属性	描述
别名	指定目标文件的别名。该别名是源文件的名称，不包括针对传输指定的任何目录路径。
编码	文本文件传输的目标文件编码。
EOL	指定文本传输的行尾标记符。有效值包括： • LF - 仅限换行字符 • CRLF - 回车符和换行符序列
truncateRecords	可选。指定将截断长于 LRECL 数据集属性的目标记录。 • True - 截断记录 • False - 合并记录 缺省设置为 false。

<queue>

在用于 <destination> 元素时，指定作为传输目标的队列的名称，该队列可以位于连接到目标代理队列管理器的任何队列管理器上。使用格式 *QUEUE@QM*，其中 *QUEUE* 是要放置消息的队列的名称，*QM* 是队列所在的队列管理器。如果已经在 <source> 元素中使用 <queue> 元素，那么将无法在 <destination> 元素中使用该元素。

属性	描述
delimiter	将文件拆分为多条消息的定界符。
delimiterType	指定定界符的类型。有效值如下： • text-Java 正则表达式 • binary - 十六进制字节序列 • size - 字节、千字节或兆字节数。例如，1 B、1 K 或 1 M。
delimiterPosition	指定期望定界符位于要在单条消息中包含的数据之前还是之后。有效选项如下： • prefix - 期望定界符在数据之前 • postfix - 期望定界符在数据之后
includeDelimiterInMessage	指定是否要在消息尾部包含用于将文件拆分为多条消息的定界符的布尔值。
编码	指定目标队列编码。
persistent	指定消息是否持久。有效值如下： • true - 消息持久 • false - 消息不持久 • qdef - 消息的持久性值由目标队列上的设置定义
setMqProps	指定是否在文件中的第一条消息上设置 WebSphere MQ 消息属性，以及在发生错误时是否将任何消息写入队列的布尔值。
unrecognisedCodePage	指定在目标队列管理器无法识别数据代码页时，文本方式传输将失败还是执行转换。有效值如下： • fail - 传输报告失败 • binary - 将数据转换为目标代码页，并且描述数据格式的 WebSphere MQ 消息头将设置为 MQFMT_NONE。 缺省行为是 fail。

<filespace>

指定要传输到的文件空间名称的组元素。

<name>

在和 <filespace> 元素一起使用时，该元素的值指定文件空间的名称。

<attributes>

Optional group element that contains one or more <attribute> elements to specify distribution attribute information if you are transferring files to a IBM 4690 store controller.

<attribute>

指定文件分发属性的可选元素。指定符号或数字值。

表 72: IBM MQ Managed File Transfer 中文件分发属性的有效值		
符号值	数字值	描述
DIST(LOCAL)	DIST(1)	本地文件
DIST(MIRRORED,UPDATE)	DIST(2)	映像文件，更新时分发
DIST(MIRRORED,CLOSE)	DIST(3)	映像文件，关闭时分发
DIST(COMPOUND,UPDATE)	DIST(4)	复合文件，更新时分发
DIST(COMPOUND,CLOSE)	DIST(5)	复合文件，关闭时分发

有关 IBM 4690 上 IBM MQ Managed File Transfer 的分发属性的更多信息，请参阅第 82 页的『文件分发属性』。

<preSourceCall>

指定在传输开始前要在传输源处调用的命令的组元素。

<postSourceCall>

指定在传输完成后要在传输源调用的命令的组元素。

<preDestinationCall>

指定在传输开始前要在传输目标处调用的命令的组元素。

<postDestinationCall>

指定在传输完成后要在传输目标处调用的命令的组元素。

<command>

在和 <preSourceCall>、<postSourceCall>、<preDestinationCall> 或 <postDestinationCall> 元素一起使用时，该元素指定要调用的命令。命令必须位于代理命令路径上。有关更多信息，请参阅第 592 页的表 50。

属性	描述
名	要运行的命令的名称。
successRC	在命令成功运行时期望的返回码。

<argument>

在和 <command> 元素一起使用时，该元素指定要传递到命令的参数。您可以在 <command> 元素中使用任意数量的 <argument> 元素。

<job>

Optional group element containing job information for the entire transfer specification. <job> is a user-defined job name identifier that is added to the log message when the transfer has started. This <job> element is the same as the <job> element that appears in the transfer log message, which is described in the following topic: 第 662 页的『文件传输日志消息格式』。

<name>

在和 <job> 元素一起使用时，该元素的值可指定作业名称。

<transferSpecifications>

包含多个传输组的 <item> 元素的组元素。请参阅[使用传输定义文件](#)以获取有关如何使用该元素的更多详细信息。

<cancel>

包含取消正在进行的文件传输所需的所有元素的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 提供的该元素的版本，指定该值。

- <transfer>**
在和 <cancel> 元素一起使用时，该元素的值指定要取消的传输请求标识。
- <job>**
包含作业信息的组元素。
- <jobName>**
指定逻辑作业标识。

文件传输取消消息格式

文件传输请求返回一个标识特定代理的传输的 48 字符标识。该标识用于取消传输。

了解传输取消消息
描述传输取消消息中使用的元素和属性：

- <cancel>**
包含取消正在进行的文件传输所需的所有元素的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 提供的该元素的版本，指定该值。

- <originator>**
包含指定请求发起方的元素的组元素。

- <hostName>**
源文件所在系统的主机名。

- <userID>**
发起文件传输的用户标识。

- <mqmdUserID>**
可选。在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

- <transfer>**
在和 <cancel> 元素一起使用时，该元素的值指定要取消的传输请求标识。

- <job>**
可选。包含作业信息的组元素。

- <jobName>**
指定逻辑作业标识。

示例

针对以下每个请求提供了符合该模式的 XML 消息的示例：

- [创建文件传输](#)
- [创建异步文件传输请求](#)
- [取消文件传输](#)
- [创建调度传输](#)
- [删除调度传输](#)
- [创建受管调用](#)
- [创建包含受管调用的文件传输](#)

相关参考

[第 855 页的『传输请求示例』](#)

您可以在代理命令队列上添加以请求代理创建或取消传输的消息的示例。

[第 857 页的『已安排的传输消息示例』](#)

可以置于代理命令队列以请求代理创建或删除调度的消息示例。

[第 858 页的『调用请求消息示例』](#)

可以放置在代理命令队列中以请求代理创建受管调用或创建调用程序的传输的消息示例。

[第 647 页的『代理状态消息格式』](#)

在创建或启动代理时，该代理会向其协调队列管理器上的 SYSTEM.FTE 主题（在 SYSTEM.FTE/Agents/*agent name* 主题上）发布其详细信息。

[第 658 页的『文件传输状态消息格式』](#)

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 SYSTEM.FTE/Transfers/*agent_name/transfer ID* 主题中），它符合 TransferStatus.xsd XML 模式。TransferStatus.xsd 文件位于 WMQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

[第 662 页的『文件传输日志消息格式』](#)

文件传输日志消息发布在主题字符串为 Log/*agent_name/transfer_ID* 的 SYSTEM.FTE 主题内。这些消息符合模式 TransferLog.xsd，该模式位于 IBM MQ Managed File Transfer 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

[第 686 页的『已安排的传输日志消息格式』](#)

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/*agent name/schedule ID* 主题中）。该消息符合 ScheduleLog.xsd XML 模式。

[第 859 页的『监视器请求消息格式』](#)

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 fteCreateMonitor 命令或使用 WebSphere MQ Explorer 接口。

[第 871 页的『针对安全性的消息格式』](#)

本主题描述发布到协调队列管理器且与安全性相关的消息。

传输请求示例

您可以在代理命令队列上添加以请求代理创建或取消传输的消息的示例。

创建传输请求

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
version="4.00"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/>
    <destinationAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/>
    <transferSet>
      <item checksumMethod="MD5" mode="binary">
        <source disposition="leave" recursive="false">
          <file>etc/passwd</file>
        </source>
        <destination exist="overwrite" type="directory">
          <file>tmp</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

创建传输请求 - 传输至 IBM 4690

在该示例 XML 中，当将文件 xyz.txt 传输至 IBM 4690 存储控制器的目录 c:\adx_test 中时，设置为关闭时进行镜像。

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="5.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName><userID>bob</userID>
    </originator>
    <sourceAgent agent="AGENT_A" QMgr="qm_a"/>
    <destinationAgent agent="AGENT_B" QMgr="qm_b"/>
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>xyz.txt</file>
        </source>
        <destination type="directory" exist="error">
          <file>c:\adx_test</file>
          <attributes>
            <attribute>DIST(MIRRORED,CLOSE)</attribute>
          </attributes>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

有关 IBM 4690 上 IBM MQ Managed File Transfer 的分发类型属性的更多信息，请参阅第 82 页的『文件分发属性』。

创建传输请求 - 同步

在用户请求分块同步请求时，也就是，他们等待传输完成并接收状态消息，命令队列上放置的消息将包含一个应答元素，指定要向其发送应答消息的队列。以下示例显示在 FTEAGENT 使用的命令队列上放置的消息：

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="FTEAGENT"
      QMgr="QM1"/>
    <destinationAgent agent="AGENT2"
      QMgr="QM2"/>
    <reply QMGR="QM1">WMQFTE.492D0D5502770020</reply>
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:\sourcefiles\source1.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>c:\destinationfiles\dest1.doc</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

将使用命令队列管理器的名称填充 <reply> 元素，其中，已创建临时动态队列来接收有关成功（或相反）完成传输的应答。临时动态队列名称由两部分组成：

- 由 command.properties 配置文件中的键 **dynamicQueuePrefix** 定义的前缀（它是 WMQFTE。缺省情况下）
- IBM MQ 生成的队列的标识

取消传输请求

```
<?xml version="1.0" encoding="UTF-8"?>
<cancel xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
  </originator>
  <transfer>414D51205553322E42494E44494E47538B0F404D032C0020</transfer>
  <reply QMGR="QM_JUPITER">WMQFTE.4D400F8B20002007</reply>
</cancel>
```

相关参考

第 842 页的『文件传输请求消息格式』

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

已安排的传输消息示例

可以置于代理命令队列以请求代理创建或删除调度的消息示例。

创建已安排的传输

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName>
      <userID>mqm</userID>
    </originator>
    <schedule>
      <submit timebase="admin" timezone="Europe/London">2010-01-01T21:00</submit>
    </schedule>
    <sourceAgent QMgr="US2.BINDINGS" agent="US2.BINDINGS.FILE"/>
    <destinationAgent QMgr="US2.BINDINGS" agent="US2.BINDINGS.FILE"/>
    <transferSet>
      <item checksumMethod="MD5" mode="binary">
        <source disposition="leave" recursive="false">
          <file>/etc/passwd</file>
        </source>
        <destination exist="overwrite" type="directory">
          <file>/tmp</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

删除已安排的传输

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <deleteScheduledTransfer>
    <originator>
      <delete>
        <hostName>example.com.</hostName>
        <userID>mqm</userID>
      </delete>
    </originator>
    <ID>1</ID>
    <reply QMGR="US2.BINDINGS">WMQFTE.4D400F8B20003902</reply>
  </deleteScheduledTransfer>
</request>
```

```
</deleteScheduledTransfer>
</request>
```

相关参考

第 842 页的『文件传输请求消息格式』

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

调用请求消息示例

可以放置在代理命令队列中以请求代理创建受管调用或创建调用程序的传输的消息示例。

受管调用请求示例

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="1.00"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedCall>
    <originator>
      <hostName>example.com.</hostName>
      <userID>mqm</userID>
    </originator>
    <agent agent="DNWE" QMgr="QM1"/>
    <transferSet>
      <call>
        <command name="echo" successRC="0">
          <argument>call</argument>
          <argument>test</argument>
        </command>
      </call>
    </transferSet>
  </managedCall>
  <job>
    <name>managedCallCalls.xml</name>
  </job>
</request>
```

具有调用的受管传输请求示例

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="1.00"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent agent="DNWE" QMgr="QM1"/>
    <destinationAgent agent="DNWE" QMgr="QM1"/>
    <transferSet>
      <preSourceCall>
        <command name="echo" successRC="0">
          <argument>preSourceCall</argument>
          <argument>test</argument>
        </command>
      </preSourceCall>
      <postSourceCall>
        <command name="echo" successRC="0">
          <argument>postSourceCall</argument>
          <argument>test</argument>
        </command>
      </postSourceCall>
      <preDestinationCall>
        <command name="echo" successRC="0">
          <argument>preDestinationCall</argument>
          <argument>test</argument>
        </command>
      </preDestinationCall>
    </transferSet>
  </managedTransfer>
</request>
```

```

        </preDestinationCall>
        <postDestinationCall>
            <command name="echo" successRC="0">
                <argument>postDestinationCall</argument>
                <argument>test</argument>
            </command>
        </postDestinationCall>
    </transferSet>
</job>
    <name>managedTransferCalls.xml</name>
</job>
</managedTransfer>
</request>

```

相关概念

第 303 页的『指定要运行的程序』

您可在正在运行 IBM MQ Managed File Transfer 代理的系统上运行程序。作为文件传输请求的一部分，您可以指定在传输启动前或者完成后运行的程序。此外，您可以通过提交受管呼叫请求来启动不属于文件传输请求的程序。

相关参考

第 842 页的『文件传输请求消息格式』

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

监视器请求消息格式

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 fteCreateMonitor 命令或使用 WebSphere MQ Explorer 接口。

监视器 XML 必须符合 Monitor.xsd 模式（即，将 <monitor> 元素用作根元素）。

监视器消息可以具有下列其中一个根元素：

- <monitor> - 用于创建和启动新的资源监视器
- <deleteMonitor> - 用于停止和删除现有监视器

fteListMonitors 命令没有命令消息，因为该命令直接从 SYSTEM.FTE 主题检索匹配的监视定义。

模式

以下模式描述了监视器请求 XML 消息中哪些元素有效。

```

<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema"
    targetNamespace="https://www.ibm.com/xmlns/wmqfte/7.0.1/
    MonitorDefinition"
    xmlns="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition">

    <xsd:include schemaLocation="FileTransfer.xsd" />

    <xsd:element name="monitor">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="name" type="monitorNameType"
                    minOccurs="1" maxOccurs="1" />
                <xsd:element name="description" type="xsd:string"
                    minOccurs="0" maxOccurs="1" />
                <xsd:element name="pollInterval" type="pollIntervalType"
                    minOccurs="1" maxOccurs="1"
                    default="10" />
                <xsd:element name="batch" type="batchType"
                    minOccurs="0" maxOccurs="1" />
                <xsd:element name="agent" type="agentNameType"
                    minOccurs="1" maxOccurs="1" />
                <xsd:element name="resources" type="monitorResourcesType"
                    minOccurs="0" maxOccurs="1" />
                <xsd:element name="triggerMatch" type="triggerMatchType"

```

```

        maxOccurs="1"          minOccurs="1" />
        <xsd:element name="reply" type="replyType"
        maxOccurs="1"          minOccurs="0" />
        <xsd:element name="tasks" type="monitorTasksType"
        maxOccurs="1"          minOccurs="1" />
        <xsd:element name="originator" type="origRequestType"
        maxOccurs="1"          minOccurs="1"/>
        <xsd:element name="job" type="jobType"
        maxOccurs="1"          minOccurs="0"/>
        <xsd:element name="defaultVariables" type="defaultVariablesType"
        maxOccurs="1"          minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="version" type="versionType" use="required" />
</xsd:complexType>
</xsd:element>

<xsd:element name="deleteMonitor">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="name" type="monitorNameType"
            minOccurs="1"          maxOccurs="1" />
            <xsd:element name="originator" type="origRequestType"
            maxOccurs="1"          minOccurs="1"/>
            <xsd:element name="reply" type="replyType"
            maxOccurs="1"          minOccurs="0" />
        </xsd:sequence>
        <xsd:attribute name="version" type="versionType" use="required" />
    </xsd:complexType>
</xsd:element>

<xsd:complexType name="transferRequestType">
    <xsd:choice>
        <xsd:element name="managedTransfer" type="managedTransferType" />
        <xsd:element name="managedCall" type="managedCallType" />
    </xsd:choice>
    <xsd:attribute name="version" type="versionType" />
</xsd:complexType>

<xsd:complexType name="monitorResourcesType">
    <xsd:choice>
        <xsd:sequence>
            <xsd:element name="directory" type="monitoredDirectoryType"
            minOccurs="1"          maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:element name="queue" type="monitoredQueueType" />
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="monitoredDirectoryType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="recursionLevel"
            type="xsd:nonNegativeInteger" />
            <xsd:attribute name="id" type="resourceIdAttrType" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="monitoredQueueType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="id" type="resourceIdAttrType" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="triggerMatchType">
    <xsd:sequence>
        <xsd:element name="conditions" type="conditionsType"
        minOccurs="1"          maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="conditionsType">
    <xsd:choice minOccurs="1">
        <xsd:element name="allOf" type="listPredicateType"
        minOccurs="1"          maxOccurs="1" />
        <xsd:element name="anyOf" type="listPredicateType"
        minOccurs="1"          maxOccurs="1" />
        <xsd:element name="condition" type="conditionType"
        minOccurs="1"          maxOccurs="1" />
    </xsd:choice>

```

```

</xsd:complexType>

<xsd:complexType name="listPredicateType">
  <xsd:choice>
    <xsd:element name="condition" type="conditionType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="conditionType">
  <xsd:sequence>
    <xsd:element name="name" type="conditionNameType"
      minOccurs="0" maxOccurs="1" />
    <xsd:element name="resource" type="resourceIdType"
      minOccurs="0" maxOccurs="1" />
    <xsd:choice minOccurs="1">
      <xsd:element name="fileMatch"
type="fileMatchConditionType"
      minOccurs="1" maxOccurs="1" />
      <xsd:element name="fileNoMatch"
type="fileNoMatchConditionType"
      minOccurs="1"
maxOccurs="1" />
      <xsd:element name="fileSize"
type="fileSizeConditionType"
      minOccurs="1" maxOccurs="1" />
      <xsd:element name="queueNotEmpty"
type="queueNotEmptyConditionType"
      minOccurs="1" maxOccurs="1" />
      <xsd:element name="completeGroups"
type="completeGroupsConditionType"
      minOccurs="1" maxOccurs="1" />
      <xsd:element name="fileSizeSame"
type="fileSizeSameType"
      minOccurs="1" maxOccurs="1"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="fileMatchConditionType">
  <xsd:sequence>
    <xsd:element name="pattern" type="conditionPatternType"
      minOccurs="0" default="*.*" />
    <xsd:element name="exclude" type="conditionPatternType"
      minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="fileNoMatchConditionType">
  <xsd:sequence>
    <xsd:element name="pattern" type="conditionPatternType"
      minOccurs="0" default="*.*" />
    <xsd:element name="exclude" type="conditionPatternType"
      minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="fileSizeConditionType">
  <xsd:sequence>
    <xsd:element name="compare" type="sizeCompareType"
      minOccurs="1" default="0" />
    <xsd:element name="pattern" type="conditionPatternType"
      minOccurs="0" default="*.*" />
    <xsd:element name="exclude" type="conditionPatternType"
      minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="sizeCompareType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:int">
      <xsd:attribute name="operator" type="sizeOperatorType"
use="required" />
      <xsd:attribute name="units" type="fileSizeUnitsType"
use="required" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="sizeOperatorType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value=">=" />
  </xsd:restriction>

```



```

</xsd:simpleType>

<xsd:simpleType name="fileSizeUnitsType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[bB]|[kK][bB]|[mM][bB]|[gG][bB]" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="conditionPatternType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="type" type="patternTypeAttributeType"
        use="optional" default="wildcard"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="patternTypeAttributeType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="regex" />
    <xsd:enumeration value="wildcard" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="conditionNameType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string" />
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="queueNotEmptyConditionType"/>

<xsd:complexType name="completeGroupsConditionType"/>

<xsd:complexType name="fileSizeSameType">
  <xsd:sequence>
    <xsd:element name="pattern" type="conditionPatternType"
      minOccurs="1" maxOccurs="1"/>
    <xsd:element name="exclude" type="conditionPatternType"
      minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="polls" type="positiveIntegerType" use="required" />
</xsd:complexType>

<xsd:complexType name="pollIntervalType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:int">
      <xsd:attribute name="units" type="timeUnitsType"
        use="optional" default="minutes" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="batchType">
  <xsd:attribute name="maxSize" type="positiveIntegerType" use="required"/>
</xsd:complexType>

<xsd:simpleType name="timeUnitsType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="seconds" />
    <xsd:enumeration value="minutes" />
    <xsd:enumeration value="hours" />
    <xsd:enumeration value="days" />
    <xsd:enumeration value="weeks" />
    <xsd:enumeration value="months" />
    <xsd:enumeration value="years" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="monitorTasksType">
  <xsd:sequence>
    <xsd:element name="task" type="monitorTaskType"
      minOccurs="1" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="monitorTaskType">
  <xsd:sequence>
    <xsd:element name="name" type="monitorTaskNameType"
      minOccurs="1" maxOccurs="1" />
    <xsd:element name="description" type="xsd:string"
      minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

```

```

        <xsd:element name="transfer"          type="transferTaskType"
                    minOccurs="0"            maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="transferTaskType">
    <xsd:sequence>
        <xsd:element name="request"          type="transferRequestType"
                    minOccurs="1"            maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="resourceIdType">
    <xsd:attribute name="id" type="xsd:string" use="optional" />
</xsd:complexType>

<xsd:simpleType name="resourceIdAttrType">
    <xsd:restriction base="xsd:string"></xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="monitorNameType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="^[^%\\*]+"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="agentNameType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="^[._0-9A-Z]*"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="monitorTaskNameType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value=".*"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="defaultVariablesType">
    <xsd:sequence>
        <xsd:element name="variable"          type="variableType"
                    maxOccurs="unbounded" minOccurs="1" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="variableType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="key" type="xsd:string" use="required" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

</xsd:schema>

```

了解创建监视器消息

以下描述了创建监视器消息中使用的元素和属性：

元素描述

<monitor>

包含取消正在进行的文件传输所需的所有元素的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 提供的该元素的版本，指定该值。

<name>

监视器的名称，在监视器代理内是唯一的。

<description>

监视器的描述（当前未使用）。

<pollInterval>

每次针对触发器条件检查资源的时间间隔。

属性	描述
单元	指定轮询时间间隔的时间单位。有效值包括： • 秒 • 分钟 • 小时 • days • weeks • months • years

<agent>

与监视器相关联的代理的名称。

<resources>

组元素，包含用于指定要监视的资源元素。

<directory>

标准路径，用于指定要监视的监视器代理机器上的目录。

属性	描述
recursionLevel	除指定目录外，要监视的子目录数。
标识	资源的唯一标识。

<queue>

队列名称，用于指定要在监视代理的队列管理器上监视的队列。

<triggerMatch>

组元素，包含的元素用于指定要与受监视资源相比较的触发器条件。

<conditions>

组元素，包含的元素用于指定要与受监视资源相比较的条件的类型。

<allOf>

谓词，用于指定必须满足包含的所有条件。

<anyOf>

谓词，用于指定必须满足包含的任意条件。

<condition>

定义将促成整个监视器触发器条件的比较条件。

<name>

条件的名称。

<resource>

标识要与条件相比较的资源定义。

属性	描述
标识	资源的唯一标识。

如果正在监视的资源是一个目录，那么必须在条件中指定以下三个元素中的一个：

- fileMatch
- fileNoMatch
- fileSize

如果监视中的资源是队列，那么必须在条件中指定以下两个元素中的一个：

- queueNotEmpty
- completeGroups

<fileMatch>

文件名匹配条件的组元素。

<pattern>

指定文件名匹配模式。资源上的文件必须与模式相匹配才能满足条件。缺省模式为 *（任何文件都将匹配）。

<fileNoMatch>

逆文件名匹配条件的组元素。

<pattern>

指定逆文件名匹配模式。如果受监视资源上的所有文件均不匹配，那么满足条件。缺省模式为 *（缺少任何文件都将匹配）。

<fileSize>

文件大小比较的组元素。

<compare>

指定文件大小比较。该值必须是非负整数。

属性	描述
操作员	要使用的比较运算符。仅支持 > = '。
单元	指定文件大小单位，可以是以下其中一项： • B - 字节 • KB - 千字节 • MB - 兆字节 • GB - 千兆字节 单位值不区分大小写，因此 mb' 与 MB' 作用相同。

<pattern>

要匹配的文件名模式。缺省值为 *（任何文件都将匹配）。

<queueNotEmpty>

只能在资源是队列时指定此属性。指定要触发监视器，队列上必须有消息。

<completeGroups>

只能在资源是队列时指定此属性。指定要触发监视器，队列上必须有完整的一组消息。针对队列上每个完整的组执行一项传输任务。

<reply>

可选元素，用于为异步请求指定应答队列。

属性	描述
QMGR	队列管理器名称。

<tasks>

组元素，包含的元素可指定在满足监视器触发器条件时要调用的任务。

<task>

组元素，定义在满足触发器条件时监视器将调用的个别任务。当前只能指定一项任务。

<name>

任务的名称。接受任何字母数字字符。

<description>

任务的描述。允许任何文本值。

<transfer>

用于定义传输任务的组元素。

<request>

用于定义任务类型的组元素。这必须包含从 `FileTransfer.xsd` 模式定义继承的以下其中一个元素：

- [managedTransfer](#)
- [managedCall](#)

属性	描述
version	请求的版本（由 IBM MQ Managed File Transfer 提供）。这采用 n.mm 格式，其中 n 是主要发行版本，mm 是次要版本。例如，1.00。

<originator>

包含指定请求发起方的元素的组元素。

<hostName>

源文件所在系统的主机名。

<userID>

发起文件传输的用户标识。

<mqmdUserID>

可选。在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<job>

包含作业信息的组元素。

<jobName>

指定逻辑作业标识。

<defaultVariables>

包含一个或多个 `variable` 元素的组元素。监视队列时，将在变量替换中使用这些变量。有关变量替换的更多信息，请参阅第 239 页的『使用变量替换定制 MFT 任务』。

<variable>

元素，包含与 `key` 属性提供的键相关联的值。

属性	描述
key	缺省变量的名称。

了解删除监视器消息

以下描述了删除监视器消息中使用的元素和属性：

元素描述

<deleteMonitor>

组元素，包含停止和删除监视器所需的所有元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 提供的该元素的版本，指定该值。

<name>

要删除的监视器的名称。

<originator>

包含指定请求发起方的元素的组元素。

<hostName>

源文件所在系统的主机名。

<userID>

发起文件传输的用户标识。

<mqmdUserID>

可选。在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<reply>

指定为请求生成的临时应答队列的名称。队列名称由 `command.properties` 配置文件中的 `dynamicQueuePrefix` 键来定义。如果未指定，那么队列名称的缺省值为 `WMQFTE`。

属性	描述
QMGR	在其上生成临时动态队列以接收应答的命令队列管理器的名称。

示例

针对以下每项监视器请求提供了符合该模式的 XML 消息的示例：

- [创建监视器](#)
- [删除监视器](#)

相关参考

[第 867 页的『监视器请求消息示例』](#)

可置于代理命令队列上的消息示例（用于请求代理来创建或删除监视器）。

[第 647 页的『代理状态消息格式』](#)

在创建或启动代理时，该代理会向其协调队列管理器上的 `SYSTEM.FTE` 主题（在 `SYSTEM.FTE/Agents/agent name` 主题上）发布其详细信息。

[第 842 页的『文件传输请求消息格式』](#)

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the `FileTransfer.xsd` schema and have the `<request>` element as the root element. `FileTransfer.xsd` 模式文档位于 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。`FileTransfer.xsd` 模式导入相同目录中的 `fteutils.xsd`。

[第 658 页的『文件传输状态消息格式』](#)

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 `SYSTEM.FTE/Transfers/agent_name/transfer ID` 主题中），它符合 `TransferStatus.xsd` XML 模式。`TransferStatus.xsd` 文件位于 `WMQMFT` 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

[第 662 页的『文件传输日志消息格式』](#)

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 `IBM MQ Managed File Transfer` 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

[第 686 页的『已安排的传输日志消息格式』](#)

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 `SYSTEM.FTE/Log/agent name/schedule ID` 主题中）。该消息符合 `ScheduleLog.xsd` XML 模式。

[第 871 页的『针对安全性的消息格式』](#)

本主题描述发布到协调队列管理器且与安全性相关的消息。

监视器请求消息示例

可置于代理命令队列上的消息示例（用于请求代理来创建或删除监视器）。

创建监视器请求

```
<?xml version="1.0" encoding="UTF-8"?>
<monitor:monitor xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
  version="4.00"
  xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./
Monitor.xsd">
  <name>EXAMPLEMONITOR</name>
  <pollInterval>1</pollInterval>
  <agent>US2.BINDINGS.FILE</agent>
```

```

<resources>
  <directory recursionLevel="0">/srv/nfs/incoming</directory>
</resources>
<triggerMatch>
  <conditions>
    <allOf>
      <condition>
        <fileMatch>
          <pattern>*.completed</pattern>
        </fileMatch>
      </condition>
    </allOf>
  </conditions>
</triggerMatch>
<reply QMGR="US2.BINDINGS">WMQFTE.4D400F8B20003702</reply>
<tasks>
  <task>
    <name/>
    <transfer>
      <request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        version="4.00"
        xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
        <managedTransfer>
          <originator>
            <hostName>example.com.</hostName>
            <userID>mqm</userID>
          </originator>
          <sourceAgent QMgr="US2.BINDINGS" agent="US2.BINDINGS.FILE"/>
          <destinationAgent QMgr="US2.BINDINGS" agent="US2.BINDINGS.FILE"/>
          <transferSet>
            <item checksumMethod="MD5" mode="binary">
              <source disposition="leave" recursive="false">
                <file>/srv/nfs/incoming/*.txt</file>
              </source>
              <destination exist="error" type="directory">
                <file>/srv/backup</file>
              </destination>
            </item>
          </transferSet>
        </managedTransfer>
      </request>
    </transfer>
  </task>
</tasks>
<originator>
  <hostName>example.com.</hostName>
  <userID>mqm</userID>
</originator>
</monitor:monitor>

```

删除监视器请求

```

<?xml version="1.0" encoding="UTF-8"?>
<monitor:deleteMonitor xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
  version="4.00"
  xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./
Monitor.xsd">
  <name>EXAMPLEMONITOR</name>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
  </originator>
  <reply QMGR="QM_JUPITER">WMQFTE.4D400F8B20003705</reply>
</monitor:deleteMonitor>

```

相关参考

第 859 页的『监视器请求消息格式』

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 `fteCreateMonitor` 命令或使用 WebSphere MQ Explorer 接口。

Ping 代理请求消息格式

您可以通过发出 `ftePingAgent` 命令或将 XML 消息放置在代理命令队列中 ping 代理。ping 代理请求 XML 必须符合 `PingAgent.xsd` 模式。安装 IBM MQ Managed File Transfer 后，您可以在以下目录中找到

PingAgent.xsd 模式文件: MQ_INSTALLATION_PATH/mqft/samples/schema。PingAgent.xsd 模式导入相同目录中的 fteutils.xsd。

当代理在其命令队列上收到 ping 代理请求消息时，如果代理处于活动状态，那么它会向将 ping 代理请求消息放置在命令队列上的命令或应用程序返回一条 XML 响应消息。代理的响应消息采用 Reply.xsd 定义的格式。有关该格式的更多信息，请参阅第 870 页的『应答消息格式』。

模式

以下模式描述了在 ping 代理请求 XML 消息中有效的元素。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema"
  xmlns="https://www.ibm.com/xmlns/wmqfte/7.0.1/PingAgent"
  targetNamespace="https://www.ibm.com/xmlns/wmqfte/7.0.1/PingAgent">

  <xsd:include schemaLocation="fteutils.xsd"/>

  <xsd:element name="pingAgent">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="originator" type="origRequestType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="agent" type="agentType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="reply" type="replyType" maxOccurs="1" minOccurs="0" />
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required" />
    </xsd:complexType>
  </xsd:element>

</xsd:schema>
```

了解 ping 代理请求消息。

下表中描述了在 ping 代理请求消息中使用的元素和属性：

- <pingAgent>
包含指定 ping 代理请求所需的所有元素的组元素。
- <originator>
包含指定 ping 请求的发起方所需的所有元素的组元素。
- <hostName>
发起请求的主机的名称。
- <userID>
请求的发起方的用户名。
- <mqmdUserID>
请求的发起方的 MQMD 用户名。
- <agent>
ping 的代理。

属性	描述
代理程序	必需。代理的名称。
QMgr	可选。代理连接到的队列管理器。

- <reply>
要向其发送应答消息的代理的队列名称。

属性	描述
QMGR	必需。应答队列所在的队列管理器的名称。

示例

该示例显示了发送给代理 AGENT_JUPITER 的 ping 代理消息。如果 AGENT_JUPITER 处于活动状态，并且能够处理代理请求，那么它会向 QM_JUPITER 上的队列 WMQFTE.4D400F8B20003708 发送一条响应消息。

```
<?xml version="1.0" encoding="UTF-8"?>
<ping:pingAgent xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:ping="https://www.ibm.com/xmlns/wmqfte/7.0.1/PingAgent"
  version="4.00">
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
  </originator>
  <agent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
  <reply QMgr="QM_JUPITER">WMQFTE.4D400F8B20003708</reply>
</ping:pingAgent>
```

应答消息格式

代理在其代理命令队列中收到 XML 消息时，如果需要响应，那么代理会向原始消息中定义的应答队列发送一条 XML 应答消息。应答 XML 符合 Reply.xsd 模式。Reply.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。Reply.xsd 模式将导入位于相同目录中的 fteutils.xsd。

模式

以下模式描述哪些元素在应答 XML 消息中有效。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
<xsd:include schemaLocation="TransferLog.xsd"/>
  <xsd:element name="reply">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="transferSet" type="transferSetType" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="status" type="statusType" minOccurs="1" maxOccurs="1" />
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="ID" type="IDType" use="required"/>
      <xsd:attribute name="detailedReplyMessagesDisabled" type="xsd:boolean"
use="optional"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

了解应答消息

以下列表中描述了应答消息中使用的元素和属性：

<reply>

包含用于指定应答信息的元素的元素。

属性	描述
标识	应答的标识。
version	应答消息格式的版本。
detailedReplyMessagesDisabled	关于代理禁用了详细回复（enableDetailedReplyMessages 代理属性设置为 false）的通知。

<transferSet>

指定请求传输的文件的传输结果信息。有关更多信息，请参阅第 662 页的『文件传输日志消息格式』。

- 请求的状态
- 请求规范

了解未授权日志消息

下面描述了未授权消息中使用的元素和属性：

<notAuthorized>

描述单个失败用户授权检查的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 的详细说明指定该元素的版本。
标识	未授权的请求的唯一标识。

<originator>

包含指定请求发起方的元素的组元素。

<authority>

指定用户执行请求的操作时所必需的 IBM MQ Managed File Transfer 访问权限级别。

<mqmdUserID>

在消息描述符 (MQMD) 中提供的 IBM MQ 用户标识

<action>

指定与 <notAuthorized> 元素的 ID 属性匹配的请求的权限状态。

属性	描述
time	指定发布日志条目的日期和时间（采用日期时间格式）。

<status>

结果代码和补充消息。

相关参考

[第 647 页的『代理状态消息格式』](#)

在创建或启动代理时，该代理会向其协调队列管理器上的 SYSTEM.FTE 主题（在 SYSTEM.FTE/Agents/*agent name* 主题上）发布其详细信息。

[第 842 页的『文件传输请求消息格式』](#)

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 *MQ_INSTALLATION_PATH/mqft/samples/schema* 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

[第 658 页的『文件传输状态消息格式』](#)

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 SYSTEM.FTE/Transfers/*agent_name*/*transfer ID* 主题中），它符合 TransferStatus.xsd XML 模式。TransferStatus.xsd 文件位于 WMQMFT 安装的 *MQ_INSTALLATION_PATH/mqft/samples/schema* 目录中。

[第 662 页的『文件传输日志消息格式』](#)

文件传输日志消息发布在主题字符串为 *Log/agent_name/transfer_ID* 的 SYSTEM.FTE 主题内。这些消息符合模式 TransferLog.xsd，该模式位于 IBM MQ Managed File Transfer 安装的 *MQ_INSTALLATION_PATH/mqft/samples/schema* 目录中。

[第 686 页的『已安排的传输日志消息格式』](#)

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/*agent name/schedule ID* 主题中）。该消息符合 ScheduleLog.xsd XML 模式。

[第 859 页的『监视器请求消息格式』](#)

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 `fteCreateMonitor` 命令或使用 WebSphere MQ Explorer 接口。

MQMFT 凭证文件格式

MQMFTCredentials.xml 文件包含敏感的用户标识和密码信息。MQMFTCredentials.xml 文件中的元素必须符合 MQMFTCredentials.xsd 模式。凭证文件的安全性由用户负责。

MQMFTCredentials.xml 文件是 Managed File Transfer 版本 7.5 的新增文件。

从版本 8.0 开始，此文件也可以是 z/OS 上的 PDSE 成员。

V 8.0.0.7 从 V 8.0.0 修订包 7 开始，您可以通过将新参数 **useMQCSPAAuthentication** 添加到相关用户的 MFT 凭证文件 MQMFTCredentials.xml 并将其设置为 `true`，对 Managed File Transfer 代理程序禁用缺省兼容性方式并启用 MQCSP 认证。有关更多信息，请参阅启用 MQCSP 认证方式。

MQMFTCredentials.xml 文件必须符合 MQMFTCredentials.xsd 模式。MQMFTCredentials.xml 模式文档位于 IBM MQ Managed File Transfer 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

模式

以下模式描述了在 MQMFTCredentials.xml 文件中哪些元素有效。

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  @start_non_restricted_prolog@
  Version: %Z% %I% %W% %E% %U% [%H% %T%]

  Licensed Materials - Property of IBM

  5724-H72

  Copyright IBM Corp. 2012, 2023. All Rights Reserved.

  US Government Users Restricted Rights - Use, duplication or
  disclosure restricted by GSA ADP Schedule Contract with
  IBM Corp.
  @end_non_restricted_prolog@
-->

<!--
  This schema defines the format of an MQMFTCredentials file. Files of this type
  store credential information for agent and logger processes. They can contain
  user names and passwords either in clear text or which have been obfuscated
  using the fteObfuscate command.
-->

<!-- Example mqmftCredentials.xml file:
<?xml version="1.0" encoding="UTF-8"?>
  <tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/
MQMFTCredentials"
    xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://wmqfte.ibm.com/MQMFTCredentials MQMFTCredentials.xsd">

    <tns:logger name="LOG1" user="user1" password="passw0rd"/>
    <tns:logger name="ORACLE" userCipher="kj2h3dfkgf" passwordCipher="1a3n67eaer"/>
    <tns:file path="home/emma/trust.jks" password="passw0rd"/>
    <tns:file path="/var/tmp/keystore.jks" passwordCipher="e71vKCg2pf"/>

    <tns:qmgr name="QM_COORD" user="tim" mqUserId="user1" mqPassword="passw0rd"/>
    <tns:qmgr name="QM_COORD" user="tom" mqUserId="user1" mqPasswordCipher="e71vKCg2pf"/>

V 8.0.0.7 <tns:qmgr name="QM_COORD" user="ernest" mqUserId="ernest"
mqPassword="AveryL0ngPassw0rd2135" useMQCSPAAuthentication="true"/>
  </tns:mqmftCredentials>
-->

<schema targetNamespace="http://wmqfte.ibm.com/MQMFTCredentials"
  elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials">

  <element name="mqmftCredentials" type="tns:mqmftCredentialsType"/>
-->
```

```

<complexType name="mqmftCredentialsType">
  <sequence>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="logger" type="tns:loggerType"/>
      <element name="file" type="tns:fileType"/>
      <element name="qmgr" type="tns:mqUserPassType"/>
    </choice>
  </sequence>
</complexType>

<complexType name="loggerType">
  <attribute name="name" type="string" use="required"/>
  <attribute name="user" type="string" use="optional"/>
  <attribute name="userCipher" type="string" use="optional"/>
  <attribute name="password" type="string" use="optional"/>
  <attribute name="passwordCipher" type="string" use="optional"/>
</complexType>

<complexType name="fileType">
  <attribute name="path" type="string" use="required"/>
  <attribute name="password" type="string" use="optional"/>
  <attribute name="passwordCipher" type="string" use="optional"/>
</complexType>

<!-- Example XML:

<tns:qmgr name="QM_COORD" user="tim" mqUserId="user1" mqPassword="passw0rd"/>
<tns:qmgr name="QM_COORD" user="tom" mqUserIdCipher="xh5U7812x"
mqPasswordCipher="e71vKCg2pf"/>
<tns:qmgr name="QM_COORD" mqUserId="defaultUser" mqPassword="passw0rd"/>
V 8.0.0.7 <tns:qmgr name="QM_COORD" user="ernest" mqUserId="ernest"
mqPassword="AveryL0ngPassw0rd2135" useMQCSPAAuthentication="true"/>
-->

<complexType name="mqUserPassType">
  <attribute name="name" type="string" use="required"/>
  <attribute name="user" type="string" use="optional"/>
  <attribute name="mqUserId" type="string" use="optional"/>
  <attribute name="mqUserIdCipher" type="string" use="optional"/>
  <attribute name="mqPassword" type="string" use="optional"/>
  <attribute name="mqPasswordCipher" type="string" use="optional"/>
V 8.0.0.7 <attribute name="useMQCSPAAuthentication" type="boolean" use="optional"/>
</complexType>

</schema>

```

了解 MQMFTCredentials.xml 文件

以下列表中描述了 MQMFTCredentials.xml 文件中使用的元素和属性。

<mqmftCredentials>

XML 文档的根元素。

<file>

传输中的文件。

属性	描述
path	所访问的密钥库文件或信任密钥库文件的路径。
密码	用于访问文件的密码。

<logger>

负责记录活动的记录器。

属性	描述
名	记录器的名称。
用户	记录器将用于连接到其数据库的用户名。
密码	记录器将用于连接到其数据库的密码。

<qmgr>
MQ 队列管理器连接。

属性	描述
名	关联的 MQ 队列管理器的名称。
用户	可选：请求连接的用户名称。
mqUserId or mqUserIdCipher	提供给 MQ 队列管理器的明文用户标识 (mqUserId) 或模糊文本用户标识 (mqUserIdCipher)。
mqPassword or mqPasswordCipher	提供给 MQ 队列管理器的明文密码 (mqPassword) 或模糊文本密码 (mqPasswordCipher)。

注: MQMFTCredentials.xml 文件可能包含敏感信息，因此，在创建该文件时，请确保复查文件权限。当使用沙箱时，将其设置为排除。有关沙箱的更多信息，请参阅第 99 页的『使用代理沙箱』。

相关参考

第 550 页的『fteObfuscate（加密敏感数据）』
fteObfuscate 命令用于加密凭证文件中的敏感数据。这样，获取了对凭证文件的访问权的人员就无法读取该文件的内容。

协议网桥凭证文件格式

代理配置目录中的 ProtocolBridgeCredentials.xml 文件定义用户名和凭证信息，供协议网桥代理用于在协议服务器中为自身授权。

ProtocolBridgeCredentials.xml 文件必须符合 ProtocolBridgeCredentials.xsd 模式。ProtocolBridgeCredentials.xsd 模式文档位于 MQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。用户负责手动创建 ProtocolBridgeCredentials.xml 文件，该文件不再由 **fteCreateBridgeAgent** 命令创建。MQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples 目录中提供了样本文件。

V7.5 引入了新的 <agent> 元素，其中包含指定代理的 <server> 或 <serverHost> 元素。

ProtocolBridgeCredentials.xml 文件由代理定期重新装入，并且对该文件所做的任何有效更改都将影响代理的行为。缺省重新装入时间间隔为 30 秒。此时间间隔可通过在 agent.properties 文件中指定代理程序属性 xmlConfigReloadInterval 进行更改。

模式 - V7.5 或更高版本

以下模式描述了 V8 的 ProtocolBridgeCredentials.xml 文件中哪些元素有效。

```
<schema targetNamespace="http://wmqfte.ibm.com/ProtocolBridgeCredentials" elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema" xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials">
  <!--
    <?xml version="1.0" encoding="UTF-8"?>
    <tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"
      xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials
        ProtocolBridgeCredentials.xsd">
      <tns:agent name="agent1">
        <tns:serverHost name="myserver">
          <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
          <tns:user name="jane" serverUserId="june" hostKey="1F:2e:f3">
            <tns:privateKey associationName="test" keyPassword="pwd2">
              .... private key ...
            </tns:privateKey>
          </tns:user>
        </tns:serverHost>
      </tns:agent>

      <tns:agent name="agent2">
        <tns:server name="server*" pattern="wildcard">
          <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
          <tns:user name="jane" serverUserId="june" hostKey="1F:2e:f3">
            <tns:privateKey associationName="test" keyPassword="pwd2">
```



```

        .... private key ...
        </tns:privateKey>
    </tns:user>
</tns:server>
</tns:agent>

    <tns:agent name="agent3">
        <tns:serverHost name="ftpsServer"
            keyStorePassword="keypass"
            trustStorePassword="trustpass">
            <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
        </tns:serverHost>
    </tns:agent>

</tns:credentials>
-->

<element name="credentials" type="tns:credentialsType"/>

<complexType name="credentialsType">
    <sequence>
        <element name="agent" type="tns:agentType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>

<complexType name="agentType">
    <choice minOccurs="0" maxOccurs="1">
        <element name="serverHost" type="tns:serverHostType" minOccurs="0" maxOccurs="unbounded"/>
        <element name="server" type="tns:serverType" minOccurs="0" maxOccurs="unbounded"/>
    </choice>
    <attribute name="name" type="string" use="required"/>
</complexType>

<complexType name="serverHostType">
    <sequence>
        <element ref="tns:user" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="name" type="string" use="required"/>
    <attribute name="keyStorePassword" type="string" use="optional"/>
    <attribute name="keyStorePasswordCipher" type="string" use="optional"/>
    <attribute name="trustStorePassword" type="string" use="optional"/>
    <attribute name="trustStorePasswordCipher" type="string" use="optional"/>
</complexType>

<complexType name="serverType">
    <sequence>
        <element ref="tns:user" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="name" type="string" use="required"/>
    <attribute name="pattern" type="tns:patternType" use="optional"/>
    <attribute name="keyStorePassword" type="string" use="optional"/>
    <attribute name="keyStorePasswordCipher" type="string" use="optional"/>
    <attribute name="trustStorePassword" type="string" use="optional"/>
    <attribute name="trustStorePasswordCipher" type="string" use="optional"/>
</complexType>

<element name="user" type="tns:userType"/>

<complexType name="userType">
    <sequence>
        <element ref="tns:privateKey" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="name" type="string" use="required"/>
    <attribute name="serverUserId" type="string" use="optional"/>
    <attribute name="serverUserIdCipher" type="string" use="optional"/>
    <attribute name="serverPassword" type="string" use="optional"/>
    <attribute name="serverPasswordCipher" type="string" use="optional"/>
    <attribute name="hostKey" type="string" use="optional">
        <simpleType>
            <restriction base="string">
                <pattern>
                    value="([a-zA-F0-9]){2}(:([a-zA-F0-9]){2})*">
                </pattern>
            </restriction>
        </simpleType>
    </attribute>
</complexType>

<element name="privateKey" type="tns:privateKeyType"/>

<complexType name="privateKeyType">
    <simpleContent>

```

```

        <extension base="string">
            <attribute name="keyPassword" type="string" use="optional"/>
            <attribute name="keyPasswordCipher" type="string" use="optional"/>
            <attribute name="associationName" type="string" use="required"/>
        </extension>
    </simpleContent>
</complexType>

<!--
    Determines the type of pattern matching to use.
-->
<simpleType name="patternType">
    <restriction base="string">
        <enumeration value="regex" />
        <enumeration value="wildcard" />
    </restriction>
</simpleType>
</schema>

```

了解 ProtocolBridgeCredentials.xml 文件

下表中描述了在 ProtocolBridgeCredentials.xml 文件中使用的元素和属性。

<credentials>

组元素，其中包含描述由协议网桥代理用于连接协议服务器的凭证的元素。

<agent>

包含指定代理程序的 <server> 或 <serverHost> 定义的元素。

<server>

协议网桥连接到的协议服务器。

V7.0.4 或更早版本不支持 <server> 元素。

属性	描述
名	协议服务器的名称。
模式 (pattern)	如果您已使用通配符或正则表达式来指定协议服务器名称的模式，那么请使用 wildcard 或 regex。
trustStorePassword 或 trustStorePasswordCipher	在 <server> 元素引用 FTPS 服务器时为必需。 用于访问信任库的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
keyStorePassword 或 keyStorePasswordCipher	可选。用于访问密钥库的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。

<serverHost>

协议网桥连接的协议服务器的主机名称。

The ProtocolBridgeCredentials.xml file can either contain <serverHost> elements or <server> elements but you cannot use a mixture of the two different types. 当您使用 <serverHost> 时，名称与协议服务器的主机名匹配。When you use <server>, the name is matched against the protocol server's name (as defined in the ProtocolBridgeProperties.xml file).

属性	描述
名	协议服务器的主机名或 IP 地址。
trustStorePassword 或 trustStorePasswordCipher	在 <serverHost> 元素引用 FTPS 服务器时为必需。用于访问信任库的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。

属性	描述
keyStorePassword 或 keyStorePasswordCipher	可选。用于访问密钥库的密码。此属性是可选的，除非您设置了 keyStore 属性，在这种情况下为必需。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。

<user>

从 IBM MQ Managed File Transfer 用户名到协议服务器用户名的用户映射。

属性	描述
名	在 IBM MQ Managed File Transfer 上使用的用户名。
serverUserId 或 serverUserIdCipher	在协议服务器上使用的用户名。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
serverPassword 或 serverPasswordCipher	在协议服务器上使用的用户名的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
hostKey	服务器主机 SSH 指纹。

<privateKey>

用户的专用密钥。

属性	描述
keyPassword 或 keyStorePasswordCipher	专用密钥的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
associationName	用于跟踪和记录的名称。

相关概念

[第 275 页的『协议网桥』](#)

通过协议网桥，IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件，而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

相关任务

[第 281 页的『使用 ProtocolBridgeCredentials.xml 文件映射文件服务器的凭证』](#)

通过使用协议网桥代理的缺省凭证映射功能，将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器的用户凭证。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以包含自己的凭证信息。

[第 285 页的『示例：如何配置协议网桥代理以将专用密钥凭证用于 UNIX SFTP 服务器』](#)

该示例演示了如何生成和配置 ProtocolBridgeCredentials.xml 文件。该示例是典型示例，细节会随着您的平台而不同，但原理是一致的。

[第 276 页的『使用 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性』](#)

定义要使用 ProtocolBridgeProperties.xml 文件（由 IBM MQ Managed File Transfer 在代理配置目录中提供）传输文件的一个或多个协议文件服务器的属性。

相关参考

[第 550 页的『fteObfuscate（加密敏感数据）』](#)

fteObfuscate 命令用于加密凭证文件中的敏感数据。这样，获取了对凭证文件的访问权的人员就无法读取该文件的内容。

协议网桥属性文件格式

代理配置目录中的 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性。

ProtocolBridgeProperties.xml 文件必须符合 ProtocolBridgeProperties.xsd 模式。

ProtocolBridgeProperties.xsd 模式文档位于 MQMFT 安装的 MQ_INSTALLATION_PATH/mqft/

samples/schema 目录中。模板文件 ProtocolBridgeProperties.xml 通过 **fteCreateBridgeAgent** 命令创建在代理配置目录中。

ProtocolBridgeProperties.xml 文件由代理定期重新装入，并且对该文件所做的任何有效更改都将影响代理的行为。缺省重新装入时间间隔为 30 秒。此时间间隔可通过在 agent.properties 文件中指定代理程序属性 xmlConfigReloadInterval 进行更改。

模式

以下模式描述 ProtocolBridgeProperties.xml 文件。

注: IBM MQ V7.5 或者 IBM MQ Managed File Transfer V7.0.2 或更高版本上不支持 maxReconnectRetry 和 reconnectWaitPeriod 属性。

```
<schema targetNamespace="http://wmqfte.ibm.com/ProtocolBridgeProperties" elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema" xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties">

  <!--
    Example: ProtocolBridgeProperties.xml

    <?xml version="1.0" encoding="UTF-8"?>
    <tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
      xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
        ProtocolBridgeProperties.xsd">
      <tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
      <tns:defaultServer name="myserver" />
      <tns:ftpServer name="myserver" host="myhost.hursley.ibm.com" port="1234" platform="windows"
        timeZone="Europe/London" locale="en-GB" fileEncoding="UTF-8"
        listFormat="unix" limitedWrite="false" />
      <tns:sftpServer name="server1" host="myhost.hursley.ibm.com" platform="windows"
        fileEncoding="UTF-8" limitedWrite="false">
        <limits maxListFileNames="10" />
      </tns:sftpServer>
    </tns:serverProperties>
  -->

  <!-- Root element for the document -->
  <element name="serverProperties" type="tns:serverPropertiesType"></element>

  <!--
    A container for all protocol bridge server properties
  -->
  <complexType name="serverPropertiesType">
    <sequence>
      <element name="credentialsFile" type="tns:credentialsFileName" minOccurs="0" maxOccurs="1" />
      <element name="defaultServer" type="tns:serverName" minOccurs="0" maxOccurs="1" />
      <choice minOccurs="0" maxOccurs="unbounded">
        <element name="ftpServer" type="tns:ftpServerType" />
        <element name="sftpServer" type="tns:sftpServerType" />
        <element name="ftpsServer" type="tns:ftpsServerType" />
        <element name="ftpsfgServer" type="tns:ftpsfgServerType" />
        <element name="ftpssfgServer" type="tns:ftpssfgServerType" />
      </choice>
    </sequence>
  </complexType>

  <!--
    A container for a server name
  -->
  <complexType name="serverName">
    <attribute name="name" type="tns:serverNameType" use="required" />
  </complexType>

  <!--
    A container for a credentials file name
  -->
  <complexType name="credentialsFileName">
    <attribute name="path" type="string" use="required" />
  </complexType>

  <!--
    A container for all the information about an FTP server
  -->
  <complexType name="ftpServerType">
    <sequence>
      <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
    </sequence>
  </complexType>
</schema>
```

```

        </sequence>
        <attributeGroup ref="tns:ftpServerAttributes"/>
        <attribute name="passiveMode" type="boolean" use="optional" />
    </complexType>

    <!--
        A container for all the information about an SFG FTP server
    -->
    <complexType name="ftpsfgServerType">
        <sequence>
            <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
        </sequence>
        <attributeGroup ref="tns:ftpServerAttributes"/>
    </complexType>

    <!--
        A container for all the information about an SFTP server
    -->
    <complexType name="sftpServerType">
        <sequence>
            <element name="limits" type="tns:sftpLimitsType" minOccurs="0" maxOccurs="1" />
        </sequence>
        <attributeGroup ref="tns:sftpServerAttributes"/>
    </complexType>

    <!--
        A container for all the information about a FTPS server
    -->
    <complexType name="ftpsServerType">
        <sequence>
            <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
        </sequence>
        <attributeGroup ref="tns:ftpsServerAttributes"/>
    </complexType>

    <!--
        A container for all the information about a SFG FTPS server
    -->
    <complexType name="ftpssfgServerType">
        <sequence>
            <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
        </sequence>
        <attributeGroup ref="tns:ftpsServerAttributes"/>
    </complexType>

    <!--
        Attributes common to all server types
    -->
    <attributeGroup name="generalServerAttributes">
        <attribute name="name" type="tns:serverNameType" use="required" />
        <attribute name="host" type="string" use="required" />
        <attribute name="port" type="nonNegativeInteger" use="optional" />
        <attribute name="platform" type="tns:platformType" use="required" />
        <attribute name="fileEncoding" type="string" use="required" />
        <attribute name="limitedWrite" type="boolean" use="optional" />
        <attribute name="controlEncoding" type="string" use="optional" />
    </attributeGroup>

    <!--
        Attributes common to ftp and ftps server types
    -->
    <attributeGroup name="ftpServerAttributes">
        <attributeGroup ref="tns:generalServerAttributes"/>
        <attribute name="timeZone" type="string" use="required" />
        <attribute name="locale" type="tns:localeType" use="required" />
        <attribute name="listFormat" type="tns:listFormatType" use="optional" />
        <attribute name="listFileRecentDateFormat" type="tns:dateFormatType" use="optional" />
        <attribute name="listFileOldDateFormat" type="tns:dateFormatType" use="optional" />
        <attribute name="monthShortNames" type="tns:monthShortNamesType" use="optional" />
    </attributeGroup>

    <!--
        Attributes common to ftps server types
    -->
    <attributeGroup name="ftpsServerAttributes">
        <attributeGroup ref="tns:ftpServerAttributes"/>
        <attribute name="ftpsType" type="tns:ftpsTypeType" use="optional" />
        <attribute name="trustStore" type="string" use="required" />
        <attribute name="trustStoreType" type="string" use="optional" />
        <attribute name="keyStore" type="string" use="optional" />
        <attribute name="keyStoreType" type="string" use="optional" />
    </attributeGroup>

```

```

    <attribute name="ccc" type="boolean" use="optional" />
    <attribute name="protFirst" type="boolean" use="optional" />
    <attribute name="auth" type="string" use="optional" />
    <attribute name="connectTimeout" type="nonNegativeInteger" use="optional"/>
</attributeGroup>

```

```

<!--
  A container for limit-type attributes for a server. Limit parameters
  are optional, and if not specified a system default will be used.
-->

```

```

<complexType name="generalLimitsType">
  <attributeGroup ref="tns:generalLimitAttributes"/>
</complexType>

```

```

<complexType name="sftplimitsType">
  <attributeGroup ref="tns:generalLimitAttributes"/>
  <attribute name="connectionTimeout" type="nonNegativeInteger" use="optional" />
</complexType>

```

```

<!--
  Attributes for limits common to all server types
-->

```

```

<attributeGroup name="generalLimitAttributes">
  <attribute name="maxListFileNames" type="positiveInteger" use="optional" />
  <attribute name="maxListDirectoryLevels" type="nonNegativeInteger" use="optional" />
  <attribute name="maxReconnectRetry" type="nonNegativeInteger" use="optional" />
  <attribute name="reconnectWaitPeriod" type="nonNegativeInteger" use="optional" />
  <attribute name="maxSessions" type="positiveInteger" use="optional" />
  <attribute name="socketTimeout" type="nonNegativeInteger" use="optional" />
</attributeGroup>

```

```

<!--
  The type for matching valid server names. Server names must be at least 2 characters in length
and
  are limited to alphanumeric characters and the following characters: ".", "_", "/" and "%".
-->

```

```

<simpleType name="serverNameType">
  <restriction base="string">
    <pattern value="[0-9a-zA-Z\._/%]{2,}" />
  </restriction>
</simpleType>

```

```

<!--
  The types of platform supported.
-->

```

```

<simpleType name="platformType">
  <restriction base="string">
  </restriction>
</simpleType>

```

```

<!--
  The type for matching a locale specification.
-->

```

```

<simpleType name="localeType">
  <restriction base="string">
    <pattern value="(.)[-_](.)/>
  </restriction>
</simpleType>

```

```

<!--
  The types of list format supported (for FTP servers).
-->

```

```

<simpleType name="listFormatType">
  <restriction base="string">
  </restriction>
</simpleType>

```

```

<!--
  Date format for FTP client directory listing on an FTP server. This is
  the format to be passed to methods setDefaultDateFormatStr and
  setRecentDateFormatStr for Java class:
  org.apache.commons.net.ftp.FTPClientConfig
-->

```

```

<simpleType name="dateFormatType">
  <restriction base="string">
  </restriction>
</simpleType>

```

```

<!--
  A list of language-defined short month names can be specified. These are
  used for translating the directory listing received from the FTP server.

```

```

    The format is a string of three character month names separated by "|"
-->
<simpleType name="monthShortNamesType">
  <restriction base="string">
    <pattern value="(...\|){11}(...)" />
  </restriction>
</simpleType>

<!--
  The enumerations of the allowed FTPS types: "implicit" & "explicit"
  If not specified the default is "explicit"
-->
<simpleType name="ftpsTypeType">
  <restriction base="string">
    <enumeration value="explicit" />
    <enumeration value="implicit" />
  </restriction>
</simpleType>

<!--
  Attribute Group for SFTP Servers
-->
<attributeGroup name="sftpServerAttributes">
  <attributeGroup ref="tns:generalServerAttributes" />
  <attribute name="cipherList" type="string" use="optional" />
</attributeGroup>
</schema>

```

了解 ProtocolBridgeProperties.xml 文件

以下列表中描述了 ProtocolBridgeProperties.xml 文件中使用的元素和属性：

<serverProperties>

XML 文档的根元素

<credentialsFile>

包含凭证的文件的路径。对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。有关更多信息，请参阅第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』。

<defaultServer>

充当文件传输的缺省服务器的协议文件服务器

<ftpServer>

FTP 文件服务器

<sftpServer>

SFTP 文件服务器

<ftpsServer>

FTPS 文件服务器

适用于所有类型的协议文件服务器的一般服务器属性：

属性	描述
名	必需。协议文件服务器的名称。协议服务器名称长度必须至少为两个字符，不区分大小写，且局限于字母数字字符和以下字符： • 句点 (.) • 下划线 (_) • 正斜杠 (/) • 百分号 (%)
主机	必需。要向其发送文件或接收其文件的协议文件服务器的主机名或 IP 地址。
port	可选。要向其发送文件或接收其文件的协议文件服务器的端口号。
platform	必需。要向其发送文件或接收其文件的协议文件服务器的平台。指定 UNIX 或 WINDOWS。根据您在 FTP、FTPS 或 SFTP 服务器上输入路径的方法，设置该属性。

属性	描述
	例如，如果您正在 Windows 上运行 FTP 服务器，但当您登录该服务器时，必须输入 UNIX 样式的路径（即，采用正斜杠），将该值设置为 UNIX，而不是 WINDOWS。Windows 上运行的服务器通常有 UNIX 样式文件系统。
fileEncoding	必需。定义文件服务器使用的字符编码。当您以文本方式传输文件时，使用该属性，以便在平台之间移动文件时更改正确的编码序列。例如，UTF-8。
limitedWrite	可选。写入文件服务器时的缺省方式是创建临时文件，然后在完成传输后重命名该文件。对于配置为只写的文件服务器，会使用其最终名称直接创建此文件。该属性的值可以为 true 或 false。缺省值为 false。
controlEncoding	可选。正发送给协议文件服务器的控制消息的控制编码值。该属性影响使用的文件名称的编码，并且必须与协议文件服务器的控制编码兼容。缺省值为 UTF-8。

仅适用于 FTP 和 FTPS 服务器的常规属性：

属性	描述
timeZone	必需。要向其发送文件或接收其文件的协议文件服务器的时区。例如：America/New_York 或 Asia/Tokyo。
locale	必需。要向其发送文件或接收其文件的协议文件服务器上使用的语言。例如：en_US 或 ja_JP
listFormat	可选。定义从协议文件服务器返回的文件列出信息格式的列表格式。使用 Windows 或 UNIX。缺省值为 UNIX。
listFileRecentDateFormat	可选。FTP 服务器上列出的 FTP 客户机目录的最近日期格式（少于一年）。该属性和 listFileOldDateFormat 属性允许您重新定义由协议文件服务器返回的预期日期格式。缺省值是由协议文件服务器定义的。
listFileOldDateFormat	可选。FTP 服务器上列出的 FTP 客户机目录的旧日期格式（一年以上）。该属性和 listFileRecentDateFormat 属性允许您重新定义由协议文件服务器返回的预期日期格式。缺省值是由协议文件服务器定义的。
monthShortNames	可选。用于对从协议文件服务器返回的日期信息进行解码的月名称的替换列表。该属性包含 12 个以逗号分隔的名称的列表，以覆盖缺省语言环境月值。缺省值是由协议文件服务器定义的。

仅适用于 FTP 服务器的常规属性：

属性	描述
passiveMode	可选。控制以被动方式还是主动方式连接到 FTP 服务器。 如果您将该属性的值设置为 false，那么连接采用主动方式。如果您将该值设置为 true，那么连接采用被动方式。缺省值为 false。

仅适用于 FTPS 服务器的常规属性：

属性	描述
ftpsType	可选。指定使用 FTPS 协议的显式还是隐式格式。缺省值为 explicit。
trustStore	必需。用于确定 FTPS 服务器提供的证书是否可信的信任库的位置。
trustStoreType	可选。信任库文件的格式。缺省值为 JKS。
keyStore	可选。用于在 FTPS 服务器质询时提供证书信息的密钥库的位置。缺省情况下，协议网桥不能与配置为需要客户机认证的 FTPS 服务器连接。
keyStoreType	可选。密钥库文件的格式。缺省值为 JKS。

属性	描述
ccc	可选。选择在认证完成时是否使用清除（未加密）命令通道。缺省值为 <code>false</code> ，这表示在整个 FTPS 会话期间命令通道保持加密。该属性仅在 <code>ftpsType</code> 设置为 <code>explicit</code> 时才适用。
protFirst	可选。指定在 PBSZ/PROT 命令之前还是之后向 FTPS 服务器发出 USER/PASS 命令。缺省值为 <code>false</code> ，这表示先发送 USER/PASS 命令，然后发送 PBSZ/PROT 命令。该属性仅在 <code>ftpsType</code> 设置为 <code>explicit</code> 时才适用。
auth	可选。指定在 AUTH 命令中指定的协议。将首先尝试指定的协议，然后缺省行为是尝试 TLS、SSL、TLS-C 或 TLS-P，直至 FTPS 服务器不会以 504 应答代码进行拒绝。该属性仅在 <code>ftpsType</code> 设置为 <code>explicit</code> 时才适用。

<limits>

所有类型服务器的常见属性和特定于某类型服务器的属性的容器元素：

适用于所有类型的协议文件服务器的一般限制属性：

属性	描述
maxListFileNames	可选。在协议文件服务器上的目录中扫描文件名时收集的最大名称数。缺省值为 999999999。
maxListDirectoryLevels	可选。在协议服务器上递归扫描文件名的最大目录级别数。缺省值是 1000。
maxReconnectRetry (现在不推荐使用该属性。)	不推荐使用。 该属性在 IBM MQ V7.5 或者 IBM MQ Managed File Transfer V7.0.2 或更高版本上不受支持。 可选。协议网桥代理停止尝试之前，协议服务器尝试重新连接的最大次数。缺省值为 2。
reconnectWaitPeriod (现在不推荐使用该属性。)	不推荐使用。 该属性在 IBM MQ V7.5 或者 IBM MQ Managed File Transfer V7.0.2 或更高版本上不受支持。 可选。尝试重新连接之前等待的时间段（秒）。缺省值为 10 秒。
maxSessions	可选。协议服务器的最大会话数。该数字必须大于或等于协议网桥代理源和目标传输的最大数之和。缺省值是代理属性 <code>maxSource</code> 传输， <code>maxDestination</code> 传输和 <code>maxCommandHandlerThreads</code> 以及 1 的值总和。如果这三个属性正在使用其缺省值 25、25 和 5，那么 <code>maxSessions</code> 缺省值为 56。
socketTimeout	可选。套接字超时（秒）。文件流期间使用该属性的值。缺省值为 30 秒。

仅限于适用于 **SFTP** 服务器的属性：

属性	描述
connectionTimeout	可选。等待协议文件服务器对连接请求响应的的时间（秒）。超时表示协议文件服务器不可用。缺省值是 30 秒。
cipherList	可选。指定用于在协议网桥代理和 SFTP 服务器之间通信的密码的逗号分隔列表。密码调用顺序与此列表中指定密码的顺序相同。在可以使用密码之前，必须在服务器和客户机上提供该密码。 协议网桥代理支持的密码如下所示： • blowfish-cbc

属性	描述
	• 3des-cbc • aes128-cbc • aes192-cbc • aes256-cbc • aes128-ctr • aes192-ctr • aes256-ctr • 3des-ctr • arcfour • arcfour128 • arcfour256 缺省情况下，协议网桥代理所使用的密码列表为 aes128-cbc , aes192-cbc , aes256-cbc。

相关概念

第 275 页的『协议网桥』

通过协议网桥，IBM MQ Managed File Transfer (MQMFT) 网络可以访问您的 MQMFT 网络之外的文件服务器上存储的文件，而无论是在本地域中还是在远程位置。此文件服务器可以使用 FTP、FTPS 或 SFTP 网络协议。每个文件服务器至少需要一个专用代理。专用代理称为协议网桥代理。网桥代理可以与多个文件服务器交互。

相关任务

第 276 页的『使用 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性』

定义要使用 ProtocolBridgeProperties.xml 文件（由 IBM MQ Managed File Transfer 在代理配置目录中提供）传输文件的一个或多个协议文件服务器的属性。

第 281 页的『使用 ProtocolBridgeCredentials.xml 文件映射文件服务器的凭证』

通过使用协议网桥代理的缺省凭证映射功能，将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器的用户凭证。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以包含自己的凭证信息。

第 285 页的『示例：如何配置协议网桥代理以将专用密钥凭证用于 UNIX SFTP 服务器』

该示例演示了如何生成和配置 ProtocolBridgeCredentials.xml 文件。该示例是典型示例，细节会随着您的平台而不同，但原理是一致的。

相关参考

第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』

从 IBM WebSphere MQ V7.5 起，可以在 受管文件传输 属性中使用表示文件或目录位置的环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化。例如，哪个用户在运行进程。

Connect:Direct 凭证文件格式

代理程序配置目录中的 ConnectDirectCredentials.xml 文件定义 Connect:Direct 代理程序用于向其自身授权 Connect:Direct 节点的用户名和凭证信息。

ConnectDirectCredentials.xml 文件必须符合 ConnectDirectCredentials.xsd 模式。ConnectDirectCredentials.xsd 模式文档位于 MQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。ConnectDirectCredentials.xml 样本文件位于 MQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/credentials 目录中。

文件 ConnectDirectCredentials.xml 由代理定期重载，对该文件的任何有效更改都将影响该代理的行为。缺省重新装入时间间隔为 30 秒。可以通过在 agent.properties 文件中指定代理属性

xmlConfigReloadInterval 来更改此时间间隔。

模式

以下模式描述在 ConnectDirectCredentials.xml 文件中有效的元素。

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
  This schema defines the format of the XML file that is located in the agent properties
  directory of a Connect:Direct bridge agent. The XML file ConnectDirectCredentials.xml
  is used by the default credential validation of the Connect:Direct bridge.
  For more information, see the WebSphere MQ InfoCenter
-->

<schema targetNamespace="http://wmqfte.ibm.com/ConnectDirectCredentials"
  elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://wmqfte.ibm.com/ConnectDirectCredentials"

  <!--
    <?xml version="1.0" encoding="UTF-8"?>

    <tns:credentials xmlns:tns="http://wmqfte.ibm.com/ConnectDirectCredentials"
      xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://wmqfte.ibm.com/ConnectDirectCredentials
        ConnectDirectCredentials.xsd">
      <tns:agent name="CDAGENT01">
        <tns:pnode name="cdnode*" pattern="wildcard">
          <tns:user name="MUSR_.*"
            ignorecase="true"
            pattern="regex"
            cdUserId="bob"
            cdPassword="passw0rd"
            pnodeUserId="bill"
            pnodePassword="alacazam">
          <tns:snode name="cdnode2" pattern="wildcard" userId="sue" password="foo"/>
          </tns:user>
        </tns:pnode>
      </tns:agent>
    </tns:credentials>

    -->

    <element name="credentials" type="tns:credentialsType"/>

    <complexType name="credentialsType">
      <sequence>
        <element name="agent" type="tns:agentType" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </complexType>

    <complexType name="agentType">
      <sequence>
        <element name="pnode" type="tns:pnodeType" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="name" type="string" use="required"/>
    </complexType>

    <complexType name="pnodeType">
      <sequence>
        <element name="user" type="tns:userType" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="name" type="string" use="required"/>
      <attribute name="pattern" type="tns:patternType" use="optional"/>
    </complexType>

    <complexType name="userType">
      <sequence>
        <element name="snode" type="tns:snodeType" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="name" type="string" use="required"/>
      <attribute name="ignorecase" type="boolean" use="optional"/>
      <attribute name="pattern" type="tns:patternType" use="optional"/>
      <attribute name="cdUserId" type="string" use="optional"/>
      <attribute name="cdUserIdCipher" type="string" use="optional"/>
      <attribute name="cdPassword" type="string" use="optional"/>
      <attribute name="cdPasswordCipher" type="string" use="optional"/>
      <attribute name="pnodeUserId" type="string" use="optional"/>
      <attribute name="pnodeUserIdCipher" type="string" use="optional"/>
  -->
-->
```

```

    <attribute name="pnodePassword" type="string" use="optional"/>
    <attribute name="pnodePasswordCipher" type="string" use="optional"/>
  </complexType>

  <complexType name="snodeType">
    <attribute name="name" type="string" use="required"/>
    <attribute name="pattern" type="tns:patternType" use="optional"/>
    <attribute name="userId" type="string" use="optional"/>
    <attribute name="userIdCipher" type="string" use="optional"/>
    <attribute name="password" type="string" use="optional"/>
    <attribute name="passwordCipher" type="string" use="optional"/>
  </complexType>

  <simpleType name="patternType">
    <restriction base="string">
      <enumeration value="regex"/>
      <enumeration value="wildcard"/>
    </restriction>
  </simpleType>
</schema>

```

了解 ConnectDirectCredentials.xml 文件

下表中描述了在 ConnectDirectCredentials.xml 文件中使用的元素和属性。

<credentials>

包含描述 Connect:Direct 网桥代理用于连接到 Connect:Direct 节点的凭证的元素的组元素。

<agent>

包含指定代理的 <pnode> 定义元素的组元素。

<pnode>

Connect:Direct 传输中的主节点 (PNODE)。该节点可启动到辅助节点 (SNODE) 的连接。

属性	描述
名	Connect:Direct 节点的名称。该属性的值可以是匹配多个节点名称的模式。
模式 (pattern)	指定用于 name 属性的值的模式类型。pattern 属性的有效值为 - wildcard - 使用通配符 - regex-使用 Java 正则表达式

<user>

提交传输请求的 WebSphere MQ 用户。

属性	描述
名	在 IBM MQ Managed File Transfer 上使用的用户名。该属性的值可以是匹配多个用户名的模式。
ignorecase	指定该名称是否区分大小写。ignorecase 属性的有效值包括 - true - 名称不区分大小写 - false - 名称区分大小写
模式 (pattern)	指定用于 name 属性的值的模式类型。pattern 属性的有效值为 - wildcard - 使用通配符 - regex-使用 Java 正则表达式
cdUserId 或 cdUserIdCipher	Connect:Direct 网桥用于连接到其关联的 Connect:Direct 节点的用户名。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
cdPassword 或 cdPasswordCipher	与 cdUserId 属性指定的用户名相关联的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。

属性	描述
pnodeUserId 或 pnodeUserIdCipher	Connect:Direct 主节点所使用的用户名。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
pnodePassword 或 pnodePasswordCipher	与 pnodeUserId 属性指定的用户名相关联的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。

<snode>

在 Connect:Direct 文件传输期间执行辅助节点 (SNODE) 角色的 Connect:Direct 节点。

属性	描述
名	Connect:Direct 节点的名称。该属性的值可以是匹配多个节点名称的模式。
模式 (pattern)	指定用于 name 属性的值的模式类型。pattern 属性的有效值为 - wildcard - 使用通配符 - regex-使用 Java 正则表达式
userId 或 userIdCipher	在文件传输期间用于连接到该节点的用户名。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。
password 或 passwordCipher	与 userId 属性指定的用户名相关联的密码。如果已使用 fteObfuscate 命令，那么必须使用该属性的加密版本。

示例

在此示例中，Connect:Direct 网桥代理连接到 Connect:Direct 节点 pnode1。当用户名以前缀 fteuser 开头并后跟单个字符 (例如 fteuser2) 的 WebSphere MQ 用户请求涉及 Connect:Direct 网桥的传输时，Connect:Direct 网桥代理将使用用户名 cduser 和密码 passw0rd 来连接到 Connect:Direct 节点 pnode1。当 Connect:Direct 节点 pnode1 执行其部分传输时，它将使用用户名 pnodeuser 和密码 passw0rd1。

如果 Connect:Direct 传输中的辅助节点具有以前缀 FISH 开头的名称，那么节点 pnode1 将使用用户名 fishuser 和密码 passw0rd2 来连接到辅助节点。如果 Connect:Direct 传输中的辅助节点具有以前缀 CHIPS 开头的名称，那么节点 pnode1 将使用用户名 chipsuser 和密码 passw0rd3 来连接到辅助节点。

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:credentials xmlns:tns="http://wmqfte.ibm.com/ConnectDirectCredentials"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ConnectDirectCredentials
ConnectDirectCredentials.xsd">
  <tns:agent name="CDAgent01">
    <tns:pnode name="pnode1" pattern="wildcard">
      <tns:user name="fteuser?" pattern="wildcard" ignorecase="true"
        cdUserId="cduser" cdPassword="passw0rd"
        pnodeUserId="pnodeuser" pnodePassword="passw0rd1">
      <tns:snode name="FISH*" pattern="wildcard"
        userId="fishuser" password="passw0rd2"/>
      <tns:snode name="CHIPS*" pattern="wildcard"
        userId="chipsuser" password="passw0rd3"/>
      </tns:user>
    </tns:pnode>
  </tns:agent>
</tns:credentials>
```

相关概念

第 288 页的『Connect:Direct 网桥』

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关参考

第 469 页的『fteCreateCDAgent (创建 Connect:Direct 网桥代理)』

fteCreateCDAgent 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 Connect:Direct 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 `java.util.regex` API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

Connect:Direct 进程定义文件格式

Connect:Direct 网桥代理配置目录中的 `ConnectDirectProcessDefinitions.xml` 文件指定要在文件传输过程中启动的用户定义的 Connect:Direct 进程。

`ConnectDirectProcessDefinitions.xml` 文件必须符合

`ConnectDirectProcessDefinitions.xsd` 模式。`ConnectDirectProcessDefinitions.xsd` 模式文档位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。模板

`ConnectDirectProcessDefinitions.xml` 文件由 `fteCreateCDAgent` 命令在代理配置目录中创建。

文件 `ConnectDirectProcessDefinitions.xml` 由代理定期重载，对该文件的任何有效更改都将影响该代理的行为。缺省重新装入时间间隔为 30 秒。该时间间隔可以通过指定 `agent.properties` 文件中的代理属性 `xmlConfigReloadInterval` 进行更改。

模式

以下模式描述在 `ConnectDirectProcessDefinitions.xml` 文件中有效的元素。

```
<schema targetNamespace="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions"
  elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions">

  <element name="cdprocess" type="tns:cdprocessType"></element>

  <complexType name="cdprocessType">
    <sequence>
      <element name="processSet" type="tns:processSetType"
        minOccurs="0" maxOccurs="unbounded"></element>
    </sequence>
  </complexType>

  <complexType name="processSetType">
    <sequence>
      <element name="condition" type="tns:conditionType"
        minOccurs="0" maxOccurs="1" />
      <element name="process" type="tns:processType"
        minOccurs="1" maxOccurs="1" />
    </sequence>
  </complexType>

  <complexType name="conditionType">
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="match" type="tns:matchType" />
      <element name="defined" type="tns:definedType" />
    </choice>
  </complexType>

  <complexType name="matchType">
    <attribute name="variable" type="string" use="required" />
    <attribute name="value" type="string" use="required" />
    <attribute name="pattern" type="tns:patternType" use="optional" />
  </complexType>

  <complexType name="definedType">
    <attribute name="variable" type="string" use="required" />
  </complexType>

  <complexType name="processType">
    <sequence>
      <element name="preTransfer" type="tns:transferType"
        minOccurs="0" maxOccurs="1" />
      <element name="transfer" type="tns:transferType"
        minOccurs="0" maxOccurs="1" />
      <element name="postTransferSuccess" type="tns:transferType"
        minOccurs="0" maxOccurs="1" />
      <element name="postTransferFailure" type="tns:transferType"
        minOccurs="0" maxOccurs="1" />
    </sequence>
  </complexType>
```



```

minOccurs="0"                maxOccurs="1" />
</sequence>
</complexType>

<complexType name="transferType">
  <attribute name="process" type="string" use="required" />
</complexType>

<simpleType name="patternType">
  <restriction base="string">
    <enumeration value="regex" />
    <enumeration value="wildcard" />
  </restriction>
</simpleType>

</schema>
```

了解 **ConnectDirectProcessDefinitions.xml** 文件

下表中描述了 ConnectDirectProcessDefinitions.xml 文件中使用的元素和属性。

cdProcess

XML 文档的根元素。

processSet

包含所有有关用户定义的进程集的信息的组元素。

condition

组元素，包含测试传输所基于的条件，用于确定是否使用 processSet 元素中包含的进程集。

match

一种条件，测试变量的值是否与给定值匹配。

属性	描述
variable	指定变量。此变量的值将与 value 属性的值进行比较。变量是一个内置符号。有关更多信息，请参阅第 724 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』。
值	指定要与 variable 属性指定的变量值匹配的模式。
模式 (pattern)	指定用于 value 属性值的模式类型。pattern 属性的有效值为 - wildcard - 使用通配符 - regex -使用 Java 正则表达式 该属性为可选项，缺省值为 wildcard。

defined

一种条件，用于测试是否已定义变量。

属性	描述
variable	指定变量。如果该变量存在，那么将满足匹配条件。变量是一个内置符号。要了解更多信息，请参阅 第 724 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』。

process

组元素，其中包含有关在找到匹配项时要调用的 Connect:Direct 进程的位置的信息。

transfer

要在传输请求期间调用的 Connect:Direct 进程。

属性	描述
进程	可选。指定包含要在传输请求期间调用的 Connect:Direct 进程的文件的名称。文件路径相对于 Connect:Direct 网桥代理配置目录。该属性是可选的，缺省值是使用由 MQMFT 生成的进程。对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。有关更多信息，请参阅第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』。

示例

在此示例中，有三个 processSet 元素。

第一个 processSet 元素指定如果传输请求具有值与模式 Client* 相匹配的 %FTESNODE 变量和值为 Admin 的 %FTESUSER 变量，那么 Connect:Direct 网桥代理会在传输过程中提交位于 agent_configuration_directory/AdminClient.cdp 中的 Connect:Direct 进程。

第二个 processSet 元素指定如果传输请求具有值与模式 Client* 匹配的 %FTESNODE 变量，那么 Connect:Direct 网桥代理会在传输过程中提交位于 agent_configuration_directory/Client.cdp 中的 Connect:Direct 进程。Connect:Direct 网桥代理按 processSet 元素的定义顺序读取这些元素，如果找到匹配项，那么将使用第一个匹配项，并且不会查找其他匹配项。对于与第一个和第二个 processSet 的条件都匹配的传输请求，Connect:Direct 网桥代理仅调用第一个 processSet 指定的进程。

第三个 processSet 元素没有任何条件，并且与所有传输都匹配。如果传输请求与第一个或第二个 processSet 的条件不匹配，那么 Connect:Direct 网桥代理将提交由第三个条件指定的 Connect:Direct 进程。在传输过程中，该进程位于 agent_configuration_directory/Default.cdp 中。

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:cdprocess xmlns:tns="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions
ConnectDirectProcessDefinitions.xsd">

  <tns:processSet>
    <tns:condition>
      <tns:match variable="%FTESNODE" value="Client*" pattern="wildcard" />
      <tns:match variable="%FTESUSER" value="Admin" pattern="wildcard" />
    </tns:condition>
    <tns:process>
      <tns:transfer process="AdminClient.cdp" />
    </tns:process>
  </tns:processSet>

  <tns:processSet>
    <tns:condition>
      <tns:match variable="%FTESNODE" value="Client*" pattern="wildcard" />
    </tns:condition>
    <tns:process>
      <tns:transfer process="Client.cdp" />
    </tns:process>
  </tns:processSet>

  <tns:processSet>
    <tns:process>
      <tns:transfer process="Default.cdp" />
    </tns:process>
  </tns:processSet>

</tns:cdprocess>
```

相关概念

[第 288 页的『Connect:Direct 网桥』](#)

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关任务

[第 212 页的『使用 ConnectDirectProcessDefinition.xml 文件指定要启动的 Connect:Direct 进程』](#)

指定要在 IBM MQ Managed File Transfer 传输过程中启动的 Connect:Direct 进程。IBM MQ Managed File Transfer 提供一个 XML 文件，您可以编辑该文件以指定进程定义。

相关参考

[第 469 页的『fteCreateCDAgent \(创建 Connect:Direct 网桥代理\)』](#)

fteCreateCDAgent 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

[第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』](#)

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 Connect:Direct 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条

消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 `java.util.regex` API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』

从 IBM WebSphere MQ V7.5 起，可以在 受管文件传输 属性中使用表示文件或目录位置的环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化。例如，哪个用户在运行进程。

Connect:Direct 节点属性文件格式

Connect:Direct 网桥代理配置目录中的 `ConnectDirectNodeProperties.xml` 文件指定有关文件传输中涉及的远程 Connect:Direct 节点的信息。

`ConnectDirectNodeProperties.xml` 文件必须符合 `ConnectDirectNodeProperties.xsd` 模式。`ConnectDirectNodeProperties.xsd` 模式文档位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。模板 `ConnectDirectNodeProperties.xml` 文件由 **fteCreateCDAgent** 命令在代理配置目录中创建。

文件 `ConnectDirectNodeProperties.xml` 由代理定期重载，对该文件的任何有效更改都将影响该代理的行为。缺省重新装入时间间隔为 30 秒。该时间间隔可以通过指定 `agent.properties` 文件中的代理属性 `xmlConfigReloadInterval` 进行更改。

模式

以下模式描述在 `ConnectDirectNodeProperties.xml` 文件中有效的元素。

```
<?xml version="1.0" encoding="UTF-8"?>

<schema targetNamespace="http://wmqfte.ibm.com/ConnectDirectNodeProperties"
  elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://wmqfte.ibm.com/ConnectDirectNodeProperties">

  <element name="nodeProperties" type="tns:nodePropertiesType"></element>

  <complexType name="nodePropertiesType">
    <sequence>
      <element name="credentialsFile" type="tns:credentialsFileName" minOccurs="0" maxOccurs="1" />
      <element name="node" type="tns:nodeType" minOccurs="0" maxOccurs="unbounded"></element>
    </sequence>
  </complexType>

  <complexType name="nodeType">
    <attribute name="name" type="string" use="required" />
    <attribute name="pattern" type="tns:patternType" use="optional" />
    <attribute name="type" type="string" use="required" />
  </complexType>

  <simpleType name="patternType">
    <restriction base="string">
      <enumeration value="regex" />
      <enumeration value="wildcard" />
    </restriction>
  </simpleType>

</schema>
```

了解 ConnectDirectNodeProperties.xml 文件

下表中描述了 `ConnectDirectNodeProperties.xml` 文件中使用的元素和属性。

nodeProperties

XML 文档的根元素。

credentialsFile

存储敏感信息的凭证文件的路径。对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。有关更多信息，请参阅第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』。

node

指定一个或多个 Connect:Direct 节点。

属性	描述
名	一种模式，用于标识使用节点元素指定的定义的 Connect:Direct 节点的名称。模式匹配不区分大小写。
模式 (pattern)	指定用于 name 属性值的模式类型。pattern 属性的有效值为： - wildcard - 使用通配符 - regex-使用 Java 正则表达式 要了解有关 MQMFT 使用的正则表达式类型的信息，请参阅 第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』 。
类型	指定与 name 属性给定的模式匹配的一个或多个 Connect:Direct 节点的操作系统类型。type 属性的有效值为： - Windows - 节点在 Windows 上运行 - UNIX - 节点在 UNIX 或 Linux 上运行 - z/OS、zos、os/390 或 os390 - 节点在 z/OS 上运行 该属性的值不区分大小写。

示例

在该示例中，文件将指定以下关联：

- 所有名称以“cdnodew”开头的 Connect:Direct 节点都在 Windows 平台上运行。
- 所有名称以“cdnodeu”开头的 Connect:Direct 节点都在 UNIX 平台上运行。
- 所有名称以“cdnodez”开头的 Connect:Direct 节点都在 z/OS 上运行。
- 所有其他 Connect:Direct 节点都在 UNIX 平台上运行。

Connect:Direct 网桥代理搜索从文件开头到结尾的匹配项，并使用找到的第一个匹配项。Connect:Direct 凭证文件已指定为 ConnectDirectCredentials.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:nodeProperties xmlns:tns="http://wmqfte.ibm.com/ConnectDirectNodeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ConnectDirectNodeProperties
    ConnectDirectNodeProperties.xsd">

  <tns:credentialsFile path="ConnectDirectCredentials.xml" />
  <tns:node name="cdnodew*" pattern="wildcard" type="windows" />
  <tns:node name="cdnodeu.*" pattern="regex" type="unix" />
  <tns:node name="cdnodez*" pattern="wildcard" type="zos" />
  <tns:node name="*" pattern="wildcard" type="unix" />

</tns:nodeProperties>
```

相关概念

[第 288 页的『Connect:Direct 网桥』](#)

您可以与现有 IBM Sterling Connect:Direct 网络相互传输文件。使用作为 IBM MQ Managed File Transfer 组件的 Connect:Direct 网桥在 MQMFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关参考

[第 469 页的『fteCreateCDAgent（创建 Connect:Direct 网桥代理）』](#)

fteCreateCDAgent 命令用于创建 IBM MQ Managed File Transfer 代理及其相关联的配置，以与 Connect:Direct 网桥一起使用。

[第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』](#)

IBM MQ Managed File Transfer 在一些场景中会使用正则表达式。例如，正则表达式用于与 Connect:Direct 安全凭证的用户标识相匹配，或通过每次对正则表达式进行匹配时创建一条新消息来将一个文件分割为多条消息。IBM MQ Managed File Transfer 使用的正则表达式语法是 java.util.regex API 支持的语法。该正则表达式语法类似于但不同于 Perl 语言使用的正则表达式语法。

[第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』](#)

从 IBM WebSphere MQ V7.5 起，可以在 受管文件传输 属性中使用表示文件或目录位置的环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化。例如，哪个用户在运行进程。

***fteutils.xsd* 模式文件**

此模式定义许多其他 IBM MQ Managed File Transfer 模式所使用的元素和类型。

模式

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
@start_non_restricted_prolog@
Version: %Z% %I% %W% %E% %U% [%H% %T%]

Licensed Materials - Property of IBM

5724-H72

Copyright IBM Corp. 2008, 2023. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
@end_non_restricted_prolog@
-->

<!--
This schema defines elements and types used by many of the other MQMFT schemas.
For more information about MQMFT XML message formats, see
https://www.ibm.com/docs/SSEP7X\_7.0.4/com.ibm.wmqfte.doc/message\_formats.htm
-->
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <!--
    Defines the version type 1.00 - 99.00
  -->
  <transaction version= 1.00
  <xsd:simpleType name="versionType">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[0-9]+\.[0-9][0-9]"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!--
    Defines the transaction reference
  -->
  <transaction version= 1.00 ID="414d5120514d3120202020202020205ecf0a4920011802"
  <xsd:simpleType name="IDType">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[0-9a-fA-F]{48}"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!--
    This is an alias for hostUserIDType.
    Here to allow addition of attributes on originator elements
  -->
  <xsd:complexType name="origRequestType">
    <xsd:complexContent>
      <xsd:extension base="hostUserIDType">
        <xsd:sequence>
          <xsd:element name="webBrowser" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
          <xsd:element name="webUserID" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!--
    Defines a Delete originator as a machine and user pair
  -->
  <xsd:complexType name="origDeleteType">
    <xsd:sequence>
      <xsd:element name="delete" type="hostUserIDType" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <!--
    Defines a machine, user, MQMD userID triple
  -->
  <hostName>myMachine</hostName>
```

```

        <userID>myUserId</userID>
        <mqmdUserID>MQMDUSERID</mqmdUserID>
    -->
    <xsd:complexType name="hostUserIDType">
        <xsd:sequence>
            <xsd:element name="hostName" type="xsd:string" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="userID" type="xsd:string" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="mqmdUserID" type="xsd:string" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <!--
        Define the destinationAgent with agent and queue manager name
        <destinationAgent agent="there" QMgr="far" agentType="BRIDGE" bridgeURL="ftp://
server.example.net:21" bridgeNode="DEST_NODE"/>
        optional agentType attribute expected to be one of STANDARD, BRIDGE, WEB_GATEWAY,
        EMBEDDED, CD_BRIDGE
    -->
    <xsd:complexType name="agentType">
        <xsd:attribute name="agent" type="xsd:string" use="required"/>
        <xsd:attribute name="agentType" type="xsd:string" use="optional"/>
        <xsd:attribute name="QMgr" type="xsd:string" use="optional"/>
        <xsd:attribute name="bridgeURL" type="xsd:string" use="optional"/>
        <xsd:attribute name="bridgeNode" type="xsd:string" use="optional"/>
        <xsd:attribute name="pnode" type="xsd:string" use="optional"/>
        <xsd:attribute name="snode" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <!--
        Defines the status type; attr/resultCode and 0 or many supplements
        There may also be additional command specific data, either: transfer, ping or call data
        <status resultCode="8011">
            <supplement>Azionamento del USB</supplement>
            <supplement>morto come norweign azzurro</supplement>
        </status>
    -->
    <xsd:complexType name="statusType">
        <xsd:sequence>
            <xsd:element name="supplement" type="xsd:string" maxOccurs="unbounded"
minOccurs="0"/>
            <xsd:choice>
                <xsd:element name="fileSpace" type="fileSpaceReplyType" minOccurs="0"
maxOccurs="1"/>
            </xsd:choice>
        </xsd:sequence>
        <xsd:attribute name="resultCode" type="resultCodeType" use="required"/>
    </xsd:complexType>
    <!--
        Defines the fileSpace type for use with communication between a web agent
        and a web gateway
        <fileSpace name="" location="">Quota bytes=""</fileSpace>
    -->
    <xsd:complexType name="fileSpaceReplyType">
        <xsd:attribute name="name" use="required" type="xsd:string"/>
        <xsd:attribute name="location" use="required" type="xsd:string"/>
        <xsd:attribute name="quota" use="required" type="xsd:long"/>
    </xsd:complexType>
    <!--
        Defines the destinationAgent with agent and queue manager name, plus connection
        details.
        <destinationAgent agent="there" QMgr="far"/>
    -->
    <xsd:complexType name="agentClientType">
        <xsd:attribute name="agent" type="xsd:string" use="required"/>
        <xsd:attribute name="QMgr" type="xsd:string" use="optional"/>
        <xsd:attribute name="hostName" type="xsd:string" use="optional"/>
        <xsd:attribute name="portNumber" type="xsd:nonNegativeInteger" use="optional"/>
        <xsd:attribute name="channel" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <!--
        Defines the fileURI type as string
        <file encoding="UTF8" EOL="CR">C:/from/here.txt</file>
    -->
    <xsd:complexType name="fileType">
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
                <xsd:attribute name="fileSpace" type="fileSpaceNameType" use="optional"/>
                <xsd:attribute name="alias" type="xsd:string" use="optional"/>
                <xsd:attribute name="encoding" type="encodingType" use="optional"/>
                <xsd:attribute name="EOL" type="EOLType" use="optional"/>
                <xsd:attribute name="size" type="xsd:long" use="optional"/>
                <xsd:attribute name="last-modified" type="xsd:dateTime" use="optional"/>
                <xsd:attribute name="delimiter" type="xsd:string" use="optional"/>
                <xsd:attribute name="delimiterType" type="xsd:string" use="optional"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>

```

```

        <xsd:attribute name="delimiterPosition" type="delimiterPositionType"
use="optional"/>
        <xsd:attribute name="includeDelimiterInFile" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="keepTrailingSpaces" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="truncateRecords" type="xsd:boolean" use="optional"/>
    </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!--
    Defines the filespace type as string
    <filespace>
        <name>tarquin</name>
    </filespace>
-->
<xsd:complexType name="filespaceType">
    <xsd:sequence>
        <xsd:element name="name" type="filespaceNameType"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    Defines a name element
    <name>bob</name>
-->
<xsd:simpleType name="filespaceNameType">
    <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<!--
    Defines the accepted choices for the persistent attribute.
-->
<xsd:simpleType name="persistenceType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="true"/>
        <xsd:enumeration value="false"/>
        <xsd:enumeration value="qdef"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the queueURI type as string with all supported attributes.
    <queue>QUEUE@QM</queue>
-->
<xsd:complexType name="queueType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="persistent" type="persistenceType" use="optional"/>
            <xsd:attribute name="eofMarker" type="xsd:boolean" use="optional"/>
            <xsd:attribute name="setMqProps" type="xsd:boolean" use="optional"/>
            <xsd:attribute name="split" type="xsd:boolean" use="optional"/>
            <xsd:attribute name="useGroups" type="xsd:boolean" use="optional"/>
            <xsd:attribute name="delimiter" type="xsd:string" use="optional"/>
            <xsd:attribute name="delimiterType" type="xsd:string" use="optional"/>
            <xsd:attribute name="delimiterPosition" type="delimiterPositionType"
use="optional"/>
            <xsd:attribute name="includeDelimiterInMessage" type="xsd:boolean"
use="optional"/>
            <xsd:attribute name="groupId" type="groupIdType" use="optional"/>
            <xsd:attribute name="messageId" type="messageIdType" use="optional"/>
            <xsd:attribute name="messageCount" type="xsd:nonNegativeInteger"
use="optional"/>
            <xsd:attribute name="messageLength" type="xsd:nonNegativeInteger"
use="optional"/>
            <xsd:attribute name="waitTime" type="xsd:nonNegativeInteger" use="optional"/>
            <xsd:attribute name="encoding" type="encodingType" use="optional"/>
            <xsd:attribute name="EOL" type="EOLType" use="optional"/>
            <xsd:attribute name="unrecognisedCodePage" type="unrecognisedCodePageType"
use="optional"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<!--
    Defines the accepted values for the delimiterPosition attribute.
-->
<xsd:simpleType name="delimiterPositionType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="postfix"/>
        <xsd:enumeration value="prefix"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the groupId type
    <queue groupId="414d5120514d312020202020202020205ecf0a4920011802">
    Also allow a substitution variable of the form ${variable}
-->

```



```

<xsd:simpleType name="groupIdType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9a-fA-F]{48}|\${.*\}"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the messageId type
  <queue messageId="414d5120514d3120202020202020205ecf0a4920011802">
    Also allow a substitution variable of the form ${variable}
  -->
<xsd:simpleType name="messageIdType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9a-fA-F]{48}|\${.*\}"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- Defines the accepted values for the unrecognisedCodePage attribute. -->
<xsd:simpleType name="unrecognisedCodePageType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="fail"/>
    <xsd:enumeration value="binary"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines a single source file reference
  <source type="file" recursive="false" disposition="leave">
    <file>filename</file>
  </source>
  -->
<xsd:complexType name="fileSourceType">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="file" type="fileType"/>
      <xsd:element name="queue" type="queueType"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="type" type="SourceType" use="optional"/>
  <xsd:attribute name="recursive" type="xsd:boolean" use="optional"/>
  <xsd:attribute name="disposition" type="sourceDispositionType" use="optional"/>
  <xsd:attribute name="correlationString1" type="xsd:string" use="optional"/>
  <xsd:attribute name="correlationNum1" type="xsd:nonNegativeInteger" use="optional"/>
  <xsd:attribute name="correlationBoolean1" type="xsd:boolean" use="optional"/>
</xsd:complexType>
<!--
  Defines the enumeration values for source type
  type="file|queue"
  -->
<xsd:simpleType name="SourceType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="file"/>
    <xsd:enumeration value="directory"/>
    <xsd:enumeration value="queue"/>
    <xsd:enumeration value="dataset"/>
    <xsd:enumeration value="pds"/>
    <xsd:enumeration value="filespace"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the enumeration values for source disposition
  disposition="leave|delete"
  -->
<xsd:simpleType name="sourceDispositionType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="leave"/>
    <xsd:enumeration value="delete"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines a single destination file reference
  <destination type="file" exist="overwrite">
    <file>filename</file>
  </destination>
  -->
<xsd:complexType name="fileDestinationType">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="file" type="fileType"/>
      <xsd:element name="filespace" type="filespaceType"/>
      <xsd:element name="queue" type="queueType"/>
    </xsd:choice>
    <xsd:element name="attributes" type="attributeType" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="type" type="DestinationType" use="optional"/>

```

```

        <xsd:attribute name="exist" type="existType" use="optional"/>
        <xsd:attribute name="correlationString1" type="xsd:string" use="optional"/>
        <xsd:attribute name="correlationNum1" type="xsd:nonNegativeInteger" use="optional"/>
        <xsd:attribute name="correlationBoolean1" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <!--
        Defines the enumeration values for destination file type
        type="file|directory|queue|dataset|pds|filesystem"
        'dataset' and 'pds' only apply to z/OS environments.
    -->
    <xsd:simpleType name="DestinationType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="file"/>
            <xsd:enumeration value="directory"/>
            <xsd:enumeration value="queue"/>
            <xsd:enumeration value="dataset"/>
            <xsd:enumeration value="pds"/>
            <xsd:enumeration value="filesystem"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--
        Defines the enumerations values for file exists on destination behavior
        exist="error|overwrite"
    -->
    <xsd:simpleType name="existType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="error"/>
            <xsd:enumeration value="overwrite"/>
        </xsd:restriction>
    </xsd:simpleType>

    <!--
        Defines one or more file attributes
        <destination encoding=? CFLF=?>
            <file>filename</file>
            <attributes>
                <attribute>DIST(MIRRORED,UPDATE)</attribute>
            </attributes>
        </destination>
    -->
    <xsd:complexType name="attributeType">
        <xsd:sequence>
            <xsd:element name="attribute" type="xsd:string" maxOccurs="unbounded"
minOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>

    <!--
        Defines a single file reference
        <source encoding=? CFLF=?>
            <file>filename</file>
            <checksum method="MD5">3445678</checksum>
        </source>
        .. or ..
        <destination encoding=? CFLF=?>
            <file>filename</file>
            <checksum method="MD5">3445678</checksum>
        </destination>
    -->
    <xsd:complexType name="fileChecksumType">
        <xsd:sequence>
            <xsd:element name="file" type="fileType"/>
            <xsd:element name="checksum" type="checksumType" maxOccurs="1" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <!--
        Defines the checksum type and method
        <checksum method="MD5|none">3445678</checksum>
    -->
    <xsd:complexType name="checksumType">
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
                <xsd:attribute name="method" type="checksumMethod" use="required"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <!--
        Defines the enumeration values for checksumMethod
        <checksum method="MD5|none">3445678</checksum>
        Note: uppercase is used since MD5 is acronym and normally written uppercase.
    -->

```

```

<xsd:simpleType name="checksumMethod">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="none"/>
    <xsd:enumeration value="MD5"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the enumeration values for agentRole
  agentRole="sourceAgent|destinationAgent"
-->
<xsd:simpleType name="agentRoleType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="sourceAgent"/>
    <xsd:enumeration value="destinationAgent"/>
    <xsd:enumeration value="callAgent"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the enumeration values for mode.
  text, binary or a substitution variable
  <item mode="binary|text|${variableName}">
-->
<xsd:simpleType name="modeType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="binary|text|${.*\}"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the enumeration values for EOL
  <file EOL="LF|CRLF">
-->
<xsd:simpleType name="EOLType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="LF"/>
    <xsd:enumeration value="CRLF"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the encoding type as a string
-->
<xsd:simpleType name="encodingType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<!--
  <schedule>
    <submit timebase="source|"admin">2008-12-07T16:07"</submit>
    <repeat>
      <frequency interval="hours">2</frequency>
      <expireTime>2008-12-0816:07</expireTime>
    </repeat>
  </schedule>
-->
<xsd:complexType name="scheduleType">
  <xsd:sequence>
    <xsd:element name="submit" type="submitType" maxOccurs="1" minOccurs="1"/>
    <xsd:element name="repeat" type="repeatType" maxOccurs="1" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<!--
  <submit timebase="source|admin|UTC">2008-12-07T16:07"</submit>
-->
<xsd:complexType name="submitType">
  <xsd:simpleContent>
    <xsd:extension base="noZoneTimeType">
      <xsd:attribute name="timebase" type="timebaseType" use="required"/>
      <xsd:attribute name="timezone" type="xsd:string" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<!--
  <repeat>
    <frequency interval="hours">2</frequency>
    ..optionally..
    <expireTime>2008-12-0816:07</expireTime>
    ..or..
    <expireCount>2</expireCount>
  </repeat>
-->
<xsd:complexType name="repeatType">
  <xsd:sequence>
    <xsd:element name="frequency" type="freqType" maxOccurs="1" minOccurs="1"/>
    <xsd:choice minOccurs="0">

```

```

        <xsd:element name="expireTime" type="noZoneTimeType"/>
        <xsd:element name="expireCount" type="positiveIntegerType"/>
    </xsd:choice>
</xsd:sequence>
</xsd:complexType>
<!--
    <frequency interval="hours">2</frequency>
-->
<xsd:complexType name="freqType">
    <xsd:simpleContent>
        <xsd:extension base="positiveIntegerType">
            <xsd:attribute name="interval" type="intervalType" use="required"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<!--
    Defines positive integer type
    i.e., 1+
-->
<xsd:simpleType name="positiveIntegerType">
    <xsd:restriction base="xsd:integer">
        <xsd:minInclusive value="1"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the interval enumeration values of
    "minutes", "hours", "days", "weeks", "months" or "years"
-->
<xsd:simpleType name="intervalType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="minutes"/>
        <xsd:enumeration value="hours"/>
        <xsd:enumeration value="days"/>
        <xsd:enumeration value="weeks"/>
        <xsd:enumeration value="months"/>
        <xsd:enumeration value="years"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the interval of either "source", "admin" or "UTC"
    source = use timezone of the source Agent.
    admin = use timezone of the administrator executing the command script.
    UTC = Timezone is UTC.
-->
<xsd:simpleType name="timebaseType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="source"/>
        <xsd:enumeration value="admin"/>
        <xsd:enumeration value="UTC"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines a date and time without a time zone (2008-12-08T16:07)
-->
<xsd:simpleType name="noZoneTimeType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="[\n\r\t]*d{4}-(0[1-9]|1[0-2])\-(0[1-9]|[1-2][0-9]|
3[0-1])T([0-1][0-9]|2[0-3]):[0-5][0-9](\[+[-]\d{4}|Z)?[\n\r\t]*"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the ID element, e.g. 56
-->
<xsd:simpleType name="idType">
    <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<!--
    Defines the resultCode type -2 - 9999
    <status resultCode="8011">
-->
<xsd:simpleType name="resultCodeType">
    <xsd:restriction base="xsd:int">
        <xsd:minInclusive value="-2"/>
        <xsd:maxInclusive value="9999"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Define the metaDataSet type comprising one or more key value pairs
    <metaDataSet>
        <metaData key="name">value</metaData>
        <metaData key="name">value</metaData>
    </metaDataSet>

```

```

-->
<xsd:complexType name="metaDataSetType">
  <xsd:sequence>
    <xsd:element name="metaData" type="metaDataType" maxOccurs="unbounded"
minOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<!--
  Define the metaData type which is made up of a key and a value
  <metaData key="name">value</metaData>
-->
<xsd:complexType name="metaDataType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="key" type="xsd:string" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<!--
  Defines containing element for triggers
  <trigger log="yes">
    <fileExist comparison="=" value="Exist">file1</fileExist>
    <fileSize comparison=">=" value="1GB">file1</fileSize>
  </trigger>
-->
<xsd:complexType name="triggerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="fileExist" type="fileExistTriggerType" maxOccurs="unbounded"
minOccurs="1"/>
    <xsd:element name="fileSize" type="fileSizeTriggerType" maxOccurs="unbounded"
minOccurs="1"/>
  </xsd:choice>
  <xsd:attribute name="log" type="logEnabledType" use="required"/>
</xsd:complexType>
<!--
  Defines the file exists trigger type
  <fileExist comparison="=" value="Exist">file1</trigger>
-->
<xsd:complexType name="fileExistTriggerType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="comparison" type="comparisonFileExistTriggerType"
use="required"/>
      <xsd:attribute name="value" type="valueFileExistTriggerType" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<!--
  Defines file size trigger type
  <fileSize comparison="=" value="1GB">file1,file2,file3</trigger>
-->
<xsd:complexType name="fileSizeTriggerType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="comparison" type="comparisonFileSizeTriggerType"
use="required"/>
      <xsd:attribute name="value" type="valueFileSizeTriggerType" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<!--
  Defines the enumeration values for file exists trigger conditions
  valueFileExistTriggerType="exist|noexist"
-->
<xsd:simpleType name="valueFileExistTriggerType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="exist"/>
    <xsd:enumeration value="noexist"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the enumeration values for file exists trigger comparison operator
  comparisonFileExistTriggerType="="|"!="
-->
<xsd:simpleType name="comparisonFileExistTriggerType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="="/>
    <xsd:enumeration value="!=">
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the enumeration values for file size trigger comparison operator

```

```

        comparisonFileSizeTriggerType=">="
-->
<xsd:simpleType name="comparisonFileSizeTriggerType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="&gt;"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the file size value pattern
  <fileSize comparison=">=" value="10|10B|10KB|10MB|10GB">file1</fileSize>
-->
<xsd:simpleType name="valueFileSizeTriggerType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0123456789]+([bB]|[kK][bB]|[mM][bB]|[gG][bB])"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the enumeration values for trigger logging enabled flag
  <trigger log="yes|no">
-->
<xsd:simpleType name="logEnabledType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="yes"/>
    <xsd:enumeration value="no"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the reply type
  <reply QMGR="QMGR name" persistent="true">Queue Name</reply>
-->
<xsd:complexType name="replyType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="QMGR" type="xsd:string" use="required"/>
      <xsd:attribute name="persistent" type="persistenceType" use="optional"/>
      <xsd:attribute name="detailed" type="detailedType"
use="optional" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

  <!--
    Defines the accepted choices for the detailed attribute.
  -->
  <xsd:simpleType name="detailedType">
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="true"/>
      <xsd:enumeration value="false"/>
    </xsd:restriction>
  </xsd:simpleType>

  <!--
    Defines the priority type
    <transferset priority="1">
  -->
  <xsd:simpleType name="priorityType">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[0123456789]"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!--
    Define the job information element
    <job>
      <name>JOBNAME</name>
    </job>
  -->
  <xsd:complexType name="jobType">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
  <!--
    Defines an action
    <action>
      <runCommand name="myCommand.sh" />
    </action>
  -->
  <xsd:complexType name="commandActionType">
    <xsd:choice>
      <xsd:element name="command" type="commandType" maxOccurs="1" minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>

```

```

<!--
  Defines a command
  <command name="runme" successRC="0" maxReplyLength="1024">
    <argument>firstArg</argument>
    <argument>secondArg</argument>
  </command>
-->
<xsd:complexType name="commandType">
  <xsd:sequence>
    <xsd:element name="argument" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>
    <xsd:element name="target" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>
    <xsd:element name="property" type="propertyType" maxOccurs="unbounded"
minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
  <xsd:attribute name="successRC" type="xsd:string" use="optional"/>
  <xsd:attribute name="retryCount" type="nonNegativeIntegerType" use="optional"/>
  <xsd:attribute name="retryWait" type="nonNegativeIntegerType" use="optional"/>
  <xsd:attribute name="type" type="callTypeType" use="optional"/>
  <xsd:attribute name="priority" type="commandPriorityType" use="optional"/>
  <xsd:attribute name="message" type="xsd:string" use="optional"/>
</xsd:complexType>
<!--
  Defines the enumeration values for the type of a command
  type="executable|antscript|jcl|os4690background"
-->
<xsd:simpleType name="callTypeType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="executable"/>
    <xsd:enumeration value="antscript"/>
    <xsd:enumeration value="jcl"/>
    <xsd:enumeration value="os4690background"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the priority type for a command
  priority="5"
-->
<xsd:simpleType name="commandPriorityType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[123456789]"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the property type that is used as a child of commandType
  <property name="xxx" value="yyy"/>
-->
<xsd:complexType name="propertyType">
  <xsd:attribute name="name" type="xsd:string" use="required"/>
  <xsd:attribute name="value" type="xsd:string" use="required"/>
</xsd:complexType>
<!-- Defines a non-negative integer type -->
<xsd:simpleType name="nonNegativeIntegerType">
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="0"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the transfer command specific reply information, to be included as part the
  general reply
  <transferReply>
    <preSourceData>
      <runCommandReply resultCode="0">
        <stdout>
          <line>the quick brown fox jumped over the lazy dog</line>
        </stdout>
        <stderr></stderr>
      </runCommandReply>
    </preSourceData>
  </transferReply>
-->
<xsd:complexType name="transferReplyType">
  <xsd:sequence>
    <xsd:element name="preSourceData" type="actionReplyType" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="postSourceData" type="actionReplyType" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="preDestinationData" type="actionReplyType" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="postDestinationData" type="actionReplyType" minOccurs="0"
maxOccurs="1"/>
  </xsd:sequence>

```



```

    </xsd:sequence>
</xsd:complexType>
<!--
    Define the action reply type information
    <actionReply>
        <runCommandReply resultCode="1">
            <stdout></stdout>
            <stderr>
                <line>permission denied</line>
            </stderr>
        </runCommandReply>
    </actionReply>
-->
<xsd:complexType name="actionReplyType">
    <xsd:choice>
        <xsd:element name="runCommandReply" type="commandReplyType" maxOccurs="1"
minOccurs="0"/>
    </xsd:choice>
</xsd:complexType>
<!--
    Defines command specific reply information, to be included as part the general reply
    <commandReply resultCode="0">
        <stdout>
            <line>first line  of output text</line>
            <line>second line of output text</line>
        </stdout>
        <stderr>
            <line>line of error text</line>
        </stderr>
    </commandReply>
-->
<xsd:complexType name="commandReplyType">
    <xsd:sequence>
        <xsd:element name="stdout" type="textLinesType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="stderr" type="textLinesType" maxOccurs="1" minOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="resultCode" type="xsd:int" use="required"/>
</xsd:complexType>
<!-- Defines type for lines of text -->
<xsd:complexType name="textLinesType">
    <xsd:sequence>
        <xsd:element name="line" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    Defines the ping agent command specific reply information, to be included as part the
general reply
    <pingAgentReply resultCode="0">
        <agentVersion>Build level: f000-20090408-1200</agentVersion>
    </pingAgentReply>
-->
<xsd:complexType name="pingAgentReplyType">
    <xsd:sequence>
        <xsd:element name="agentVersion" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    Defines sequence of exit elements
    <exit ...
    <exit ...
-->
<xsd:complexType name="exitGroupType">
    <xsd:sequence>
        <xsd:element name="exit" type="exitType" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    Defines the outcome of calling a command
    <command ...
    <callResult ...
-->
<xsd:complexType name="callGroupType">
    <xsd:sequence>
        <xsd:element name="command" type="commandType" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="callResult" type="callResultType" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    Defines either the successful call of a command, or a failed attempt to call a command
    <callResultType outcome="success|failure|error" retries="X">
        <result ... />
    </callResultType>

```

```

-->
<xsd:complexType name="callResultType">
  <xsd:sequence>
    <xsd:element name="result" type="resultType" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="returnCode" type="xsd:integer" use="optional"/>
  <xsd:attribute name="retries" type="xsd:integer" use="optional"/>
  <xsd:attribute name="outcome" type="outcomeType" use="required"/>
</xsd:complexType>
<!--
  Defines the information recorded for the successful call of a command
  <result...>
    <stdout...
    <stderr...
    <error...
  </result...>
-->
<xsd:complexType name="resultType">
  <xsd:sequence>
    <xsd:element name="stdout" type="outputType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="stderr" type="outputType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="error" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="returnCode" type="xsd:integer" use="optional"/>
  <xsd:attribute name="outcome" type="outcomeType" use="required"/>
  <xsd:attribute name="time" type="xsd:dateTime" use="required"/>
</xsd:complexType>
<!-- Enumeration of call outcomes - success, failure or error -->
<xsd:simpleType name="outcomeType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="success"/>
    <xsd:enumeration value="failure"/>
    <xsd:enumeration value="error"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the information recorded for each line of standard output / standard error
  generated by calling a program
  <line>line 1</line>
  <line>line 2</line>
  etc.
-->
<xsd:complexType name="outputType">
  <xsd:sequence>
    <xsd:element name="line" type="xsd:string" maxOccurs="unbounded" minOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<!--
  Defines the information recorded for an unsuccessful program call.
-->
<xsd:complexType name="callFailedType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string"/>
  </xsd:simpleContent>
</xsd:complexType>
<!--
  Defines the exit type; records the transfer exit class name and a status message
  <exit name="class com.example.exit.StartExit">
    <status ...
  </exit>
-->
<xsd:complexType name="exitType">
  <xsd:sequence>
    <xsd:element name="status" type="exitStatusType" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
</xsd:complexType>
<!--
  Defines exit status to record whether exit voted to proceed or cancel transfer.
  <status resultCode="proceed">
    <supplement>go ahead</supplement>
  </status>
-->
<xsd:complexType name="exitStatusType">
  <xsd:sequence>
    <xsd:element name="supplement" type="xsd:string" maxOccurs="unbounded"
minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="resultCode" type="exitResultEnumType" use="optional"/>
</xsd:complexType>
<!--
  Defines the enumeration for transfer exit result values.

```

```

        <status resultCode="proceed">
-->
<xsd:simpleType name="exitResultEnumType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="proceed"/>
    <xsd:enumeration value="cancelTransfer"/>
    <xsd:enumeration value="cancelTask"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

相关概念

第 791 页的『IBM MQ Managed File Transfer 的消息格式』

IBM MQ Managed File Transfer 出于多种目的使用 XML 格式的消息：与 Web 网关交互；向代理发出命令；记录有关监视器、计划安排和传输的信息；以及定义用于配置的信息。用于这些目的的 XML 格式的逻辑结构由 XML 模式描述。

开发应用程序

在传输之前或之后运行程序

使用 *fteCreateTransfer* 来启动程序的示例

您可以使用 **fteCreateTransfer** 命令来指定在传输之前或之后要运行的程序。

除使用 **fteCreateTransfer** 之外，还可通过其他方法在传输之前或之后调用程序。有关更多信息，请参阅第 303 页的『指定要运行的程序』。

所有这些示例均使用以下语法来指定程序：

```
[type:]commandspec[, [retrycount][, [retrywait][, successsrc]]]
```

有关该语法的更多信息，请参阅第 497 页的『fteCreateTransfer（创建新的文件传输）』。

运行可执行程序

以下示例指定一个称为 mycommand 的可执行程序并将两个自变量（a 和 b）传递给该程序。

```
mycommand(a,b)
```

要在传输启动之前从源代理 AGENT1 运行该程序，请使用以下命令：

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -presrc mycommand(a,b)
destinationSpecification sourceSpecification
```

运行和重试可执行程序

以下示例指定一个称为 simple 的可执行程序，该程序不接受任何自变量。针对 **retrycount** 指定值 1，针对 **retrywait** 指定值 5。这些值意味着如果该程序不返回成功返回码，那么在等待五秒钟后，将重试该程序。针对 **successsrc** 不指定任何值，因此唯一的成功返回码是缺省值 0。

```
executable:simple,1,5
```

要在传输完成之后从源代理 AGENT1 运行该程序，请使用以下命令：

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -postsr executable:simple,1,5
destinationSpecification sourceSpecification
```

在后台运行 IBM 4690 可执行程序

以下示例指定了一个称为 `mycmd.bat` 的可执行程序。针对 `priority` 指定值 9，针对 `message` 指定值 `message123`。这些值表示将为此程序提供最高优先级，`message123` 将显示在命令的 IBM 4690 系统后台控制屏幕上。针对 `successrc` 不指定任何值，因此唯一的成功返回码是缺省值 0。

```
os4690background:mycmd.bat(arg1,arg2),,,9,message123
```

要在传输启动之前从源代理 AGENT1 运行该程序，请使用以下命令：

```
fteCreateTransfer -sa AGENT1 -presrc os4690background:mycmd.bat(arg1,arg2),,,9,message123
```

运行 Ant 脚本并指定成功返回码

以下示例指定一个称为 `myscript` 的 Ant 脚本，并将两个属性传递给该脚本。使用 **fteAnt** 命令运行该脚本。`successrc` 的值指定为 `>2&<7&!5|0|14`，这样会指定返回码 0、3、4、6，并且 14 表明成功。

```
antscript:myscript(prop1=fred,prop2=bob),,,>2&<7&!5|0|14
```

要在传输启动之前从目标代理 AGENT2 运行该程序，请使用以下命令：

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -predst  
"antscript:myscript(prop1=fred,prop2=bob),,,>2&<7&!5|0|14"destinationSpecification sourceSpecification
```

运行 Ant 脚本并指定要呼叫的目标

以下示例指定称为 `script2` 的 Ant 脚本和要呼叫的两个目标：`target1` 和 `target2`。属性 `prop1` 也会传入，值为 `recmf(F,B)`。该值中的逗号 (,) 和括号会使用反斜杠字符 (\) 进行转义。

```
antscript:script2(target1,target2,prop1=recmf\F\B\),,,>2&<7&!5|0|14
```

要在传输完成之后从目标代理 AGENT2 运行该程序，请使用以下命令：

```
fteCreateTransfer -sa AGENT1 -da AGENT2  
-postdst "antscript:script2(target1,target2,prop1=recmf\F\B\),,,>2&<7&!5|0|14"  
destinationSpecification sourceSpecification
```

运行 JCL 脚本

以下示例指定了一个称为 `zosbatch` 的 JCL 脚本。针对 `retrycount` 指定值 3、针对 `retrywait` 指定值 30，并针对 `successrc` 指定值 0。这些值意味着如果该脚本不返回成功返回码 0，那么在每次尝试之间等待三十秒后，将重试该脚本三次。

```
jcl:zosbatch,3,30,0
```

要在传输完成之后从源代理 AGENT1 运行该程序，请使用以下命令：

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -postsrc jcl:zosbatch,3,30,0  
destinationSpecification sourceSpecification
```

相关概念

[第 303 页的『指定要运行的程序』](#)

您可在正在运行 IBM MQ Managed File Transfer 代理的系统上运行程序。作为文件传输请求的一部分，您可以指定在传输启动前或者完成后运行的程序。此外，您可以通过提交受管呼叫请求来启动不属于文件传输请求的程序。

相关参考

[第 497 页的『fteCreateTransfer（创建新的文件传输）』](#)

fteCreateTransfer 命令从命令行创建并启动新的文件传输。此命令可以立即启动文件传输，调度在未来日期和时间进行文件传输，将调度的传输重复一次或多次，以及基于某些条件触发文件传输。

使用 Web 网关

Web 网关 API 引用

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

该引用主题描述用于用户操作的 API。针对诸如创建和删除文件空间之类的管理操作，请参阅 [第 930 页的『Web 网关管理 API 引用』](#)。

资源类型

该规范支持以下 IBM MQ Managed File Transfer 对象类型：

文件

传输至 IBM MQ Managed File Transfer 代理的文件，或从该代理传输的文件。

文件空间

包含已发送至与该文件空间关联的用户或组的文件的逻辑区域。

传输

IBM MQ Managed File Transfer 传输的实例。

HTTP 动词

该规范支持下表中的 HTTP 动词。

HTTP 动词	IBM MQ Managed File Transfer 操作
POST	将文件上载至目标代理
GET	- 检索先前传输的状态 - 检索文件空间中的文件列表 - 从文件空间中下载文件
DELETE	从文件空间中删除和（可选）下载文件

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 312 页的『使用 IBM MQ Managed File Transfer 服务 Web 网关』](#)

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

[第 313 页的『示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』](#)

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 918 页的『使用 Web 网关的内容类型』

向 IBM MQ Managed File Transfer Web 网关提交的文件传输请求必须对应于特定的介质类型。来自 Web 网关的响应具有介质类型 application/xml 或 application/json。

第 919 页的『响应格式：XML 和 JSON』

IBM MQ Managed File Transfer Web 网关采用以下两种格式之一返回查询的响应：XML 或 JavaScript Object Notation (JSON)。

第 410 页的『HTTP 响应代码』

向 IBM MQ Managed File Transfer Web 网关发出的请求的 HTTP 响应中会返回状态码。

第 930 页的『Web 网关管理 API 引用』

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

使用 Web 网关时的 HTTP 头和 HTML 表单字段

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

HTTP 约定是在定制头前添加 x-，后跟特定于产品的标识。IBM MQ Managed File Transfer 使用产品标识 fte-。

有关下表中列出的每个标题或表单字段的可能值，请参阅描述等效 IBM MQ Managed File Transfer 命令和参数的主题。例如，x-fte-action 头的可能值集就是在使用 -de 参数时 fteCreateTransfer 命令的可能值集。

头名称	表单字段名称	支持的对象类型和动词	命令行命令和参数	描述	头的示例用法	表单字段的示例用法
x-fte-action	action	文件 (POST)	fteCreateTransfer 和 -de	指定当目标文件存在时要执行的操作。有效选项是： • overwrite • error - 这是缺省值。	x-fte-action:overwrite	<input type="HIDDEN" name="action" value="overwrite"/>
x-fte-priority	priority	文件 (POST)	fteCreateTransfer 和 -pr	指定传输的优先级。	x-fte-priority:5	<input type="HIDDEN" name="priority" value="5"/>
x-fte-type	type	文件 (POST)	fteCreateTransfer 和 -t	指定以二进制还是文本方式执行传输。有效选项是： • text • binary - 这是缺省值。	x-fte-type:binary	<input type="HIDDEN" name="type" value="binary"/>
x-fte-checksum	checksum	文件 (POST)	fteCreateTransfer 和 -cs	指定用于检查传输是否已正确发送数据的校验和算法。有效选项是： • 无 • MD5 - 这是缺省值。	x-fte-checksum:MD5	<input type="HIDDEN" name="checksum" value="MD5"/>
x-fte-metadata	metadata	文件 (POST)	fteCreateTransfer 和 -md	指定要与传输关联的元数据。在一次请求中可多次指定元数据头或表单字段。	x-fte-metadata:a=b,c=d,e=f	<input type="HIDDEN" name="metadata" value="a=b,c=d,e=f"/>
x-fte-jobname	jobname	文件 (POST)	fteCreateTransfer 和 -jn	指定要与传输关联的作业名。	x-fte-jobname:BatchOrder_1	<input type="HIDDEN" name="jobname" value="BatchOrder_1"/>

头名称	表单字段名称	支持的对象类型和动词	命令行命令和参数	描述	头的示例用法	表单字段的示例用法
x-fte-postdest	postdest	文件 (POST)	fteAnt, 其中所调用的 Ant 脚本包含嵌套的 postdst 元素	指定当文件传输完成时要在目标代理上执行的命令。支持可用于程序调用的所有属性。有关这些属性的更多详细信息, 请参阅主题第 968 页的『程序调用嵌套元素』。	x-fte-postdest: [command=virus_scan.sh, type=executable, successrc=0]	<input type="HIDDEN" name="postdest" value="[command=virus_scan.sh, type=executable, successrc=0]" />
x-fte-postdest-args	postdest-args	文件 (POST)	fteAnt, 其中所调用的 Ant 脚本在 postdst 元素上指定 arg 元素	指定如果使用 x-fte-postdest 头指定了命令, 那么会将一个或多个参数传递给该命令。仅当 x-fte-postdest 头中指定的 type 属性为 executable 时才有效。支持可用于程序调用的所有属性。有关这些属性的更多详细信息, 请参阅主题第 968 页的『程序调用嵌套元素』。	x-fte-postdest-args:[argument1, argument2]	<input type="HIDDEN" name="postdest-args" value="[argument1, argument2]" />
x-fte-postdest-properties	postdest-properties	文件 (POST)	fteAnt, 其中所调用的 Ant 脚本在 postdst 元素上指定 property 元素	指定如果使用 x-fte-postdest 头指定了 Ant 脚本, 那么会将一个或多个属性传递给该脚本。仅当 x-fte-postdest 头中指定的 type 属性为 antscript 时才有效。支持可用于程序调用的所有属性。有关这些属性的更多详细信息, 请参阅主题第 968 页的『程序调用嵌套元素』。	x-fte-postdest-properties : [scanspeed=fast, resultoutput=scan.log]	<input type="HIDDEN" name="postdest-properties" value="[scanspeed=fast, resultoutput=scan.log]" />
x-fte-postdest-targets	postdest-targets	文件 (POST)	fteAnt, 其中所调用的 Ant 脚本在 postdst 元素上指定 target 元素	指定如果使用 x-fte-postdest 头指定了 Ant 脚本, 那么将从该脚本运行一个或多个目标。仅当 x-fte-postdest 头中指定的 type 属性为 antscript 时才有效。有关更多详细信息, 请参阅主题第 968 页的『程序调用嵌套元素』。	x-fte-postdest-targets:[scanfile]	<input type="HIDDEN" name="postdest-targets" value="[scanfile]" />
x-fte-include-file-in-response	None	文件 (DELETE)	None	指定删除的文件是否包含在 HTTP 响应中。缺省值为 false。	x-fte-include-file-in-response:true 指定删除的文件是否包含在 HTTP 响应中。缺省值为 false。	None
x-fte-check-integrity	None	文件空间 (GET)	None	在要查看文件空间内容的请求中, 指定应对文件空间中的文件执行完整性检查。在要列出所有文件空间的请求中, 指定应对文件空间根目录执行完整性检查。缺省值为 false。	x-fte-check-integrity:true	None
x-fte-csrf-token	csrf-token	文件 (POST, DELETE) 文件空间 (POST, DELETE)	None	指定当前 JSESSIONID, Web 网关应在请求创建或删除文件以及创建、删除或修改文件空间时, 将其与自己的 JSESSIONID 进行比较。如果 Web 网关上启用了 CSRF 保护, 那么这可以针对跨站点请求伪造 (CSRF) 攻击提供保护。	x-fte-csrf-token: E324D3C64E6B4662 0843C88A25B4C32D	<input type="HIDDEN" name="csrf-token" value="E324D3C64E6B46620843C88A25B4C32D" />

相关概念

第 305 页的『Web 网关的方案』

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理, 并使用 HTTP 客户机检索传输的状态。

第 312 页的『使用 IBM MQ Managed File Transfer 服务 Web 网关』

通过创建提交到 Web 网关的 HTTP 请求，可上传文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

第 313 页的『示例 HTTP 流』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

相关参考

第 908 页的『Web 网关 API 引用』

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

第 911 页的『使用 Web 网关时的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 918 页的『使用 Web 网关的内容类型』

向 IBM MQ Managed File Transfer Web 网关提交的文件传输请求必须对应于特定的介质类型。来自 Web 网关的响应具有介质类型 application/xml 或 application/json。

第 919 页的『响应格式：XML 和 JSON』

IBM MQ Managed File Transfer Web 网关采用以下两种格式之一返回查询的响应：XML 或 JavaScript Object Notation (JSON)。

第 410 页的『HTTP 响应代码』

向 IBM MQ Managed File Transfer Web 网关发出的请求的 HTTP 响应中会返回状态码。

第 930 页的『Web 网关管理 API 引用』

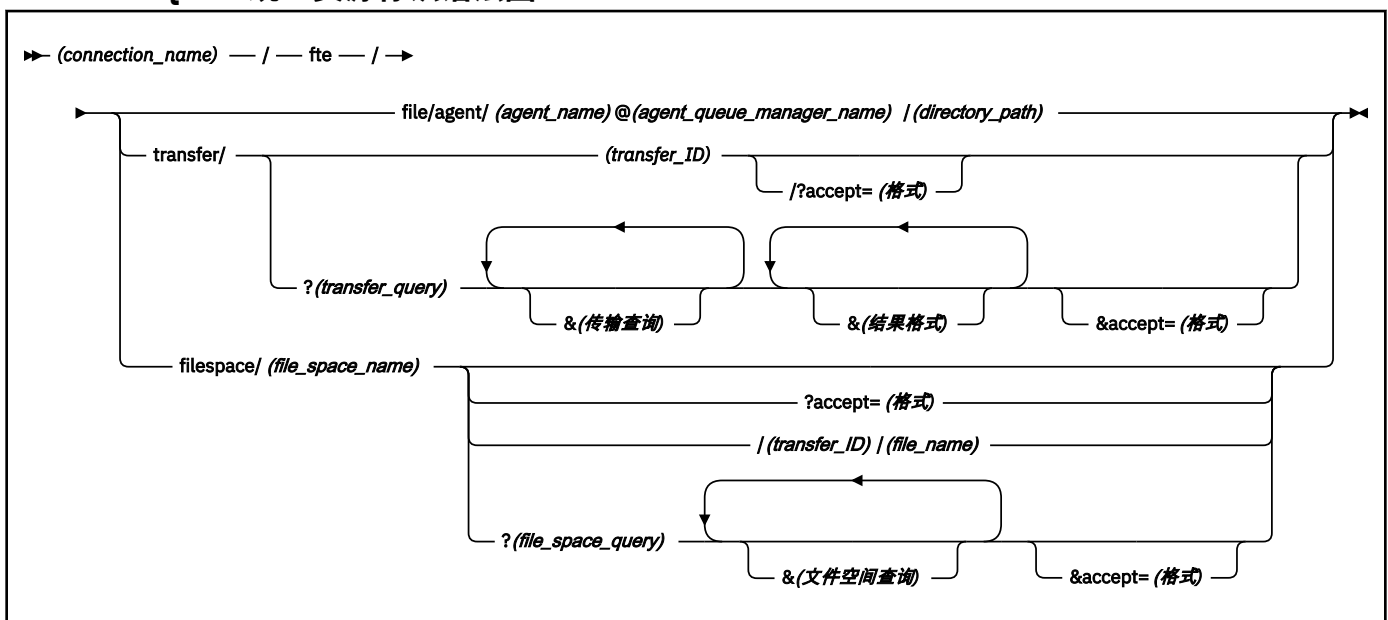
IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

使用 Web 网关时的统一资源标识语法

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

IBM MQ Managed File Transfer 资源之间会按各自的类型进行区分。资源的地址通过其资源类型和标识标记确定。

WMQFTE 统一资源标识语法图



参数

(connection_name)

必需。托管 IBM MQ Managed File Transfer Web 网关的服务器的服务器的主机名和（可选）端口。不区分大小写。

fte

必需。用于指示 URI 指向 IBM MQ Managed File Transfer Web 网关的前缀。区分大小写。

file

可选。指示您要查找文件资源的地址。区分大小写。

agent

可选。指示目标的类型是代理。区分大小写。

(agent_name)

可选。文件要发送到的代理的名称。不区分大小写，代理名称会转换为大写。

(agent_queue_manager_name)

必需。被代理用来接收文件的队列管理器的名称。区分大小写。

(directory_path)

可选。您要查找其地址的目标代理文件系统上的目录路径。目录路径必须仅包含非保留字符或转义字符。区分大小写。

如果 URI 的 *directory_path* 部分以正斜杠 (/) 字符（除用作路径分隔符的正斜杠之外）开头，那么 *directory_path* 将解析为绝对路径。如果要将文件上载到绝对路径，那么必须将正斜杠编码为字符串 %2F，这样才不会除去该正斜杠。如果希望 Web 网关上载无法写入到目标代理的文件系统上的绝对路径中，那么必须在目标代理上配置用户或代理沙箱。

如果 *directory_path* 不以额外的正斜杠字符开头，那么将相对于目标代理的传输根目录来解析目录路径。

transfer

可选。指示您要查找传输资源的地址。区分大小写。

(transfer_ID)

可选。传输标识是用于标识传输的唯一十六进制字符串（共 48 个字符）。不区分大小写。

accept=(format)

可选。指定 Web 网关返回的响应的格式。 *format* 的值是以下某个值：

- **JSON** - 指定响应采用 JavaScript Object Notation 格式。
- **XML** - 指定响应采用 XML 格式。这是缺省值。

不区分大小写。您还可以在请求中使用 **Accept:** 头来设置响应的格式。使用 URI 设置的格式优先于使用 **Accept:** 头设置的格式。

(transfer_query)

可选。从 IBM MQ Managed File Transfer Web 网关请求与查询匹配的所有传输的信息。可以指定多个查询，并以“与”字符 (&) 分隔，但每种查询类型只能包含一个查询。

查询可以是以下某种类型：

- **srcagent=(agent_name)**
- **destagent=(agent_name)**
- **agent=(agent_name)**
- **status=(status_value)**
- **metadata=(metadata_info)**

- endafter=(date)
- endbefore=(date)
- startafter=(date)
- startbefore=(date)
- srcfile=(file_path)
- destfile=(file_path)
- jobname=(job_name)
- returncode=(return_code)

有关这些查询的更多信息，请参阅第 913 页的『查询参数』。

fileSpace

可选。指示您要查找文件空间资源的地址。区分大小写。

(file_space_name)

可选。您要查找其地址的文件空间的名称。这是与文件空间关联的用户的名称。区分大小写。

(file_name)

可选。要下载的文件名称。如果文件名中包含空格字符，那么该字符在 URI 中必须以字符串 %20 表示。区分大小写。

(file_space_query)

可选。从 IBM MQ Managed File Transfer Web 网关请求与查询匹配的文件空间中所有文件的信息。可以指定多个查询，并以“与”字符 (&) 分隔，但每种查询类型只能包含一个查询。

查询可以是以下某种类型：

- endafter=(date)
- endbefore=(date)
- startafter=(date)
- startbefore=(date)

有关这些查询的更多信息，请参阅第 913 页的『查询参数』。

(result_format)

- sortby=(sort_by_values)
- sort=(sort_values)
- start=(start_value)
- count=(count_value)

有关这些结果格式的更多信息，请参阅第 916 页的『结果格式参数』。

查询参数

srcagent=(agent_name)

请求源代理为 *agent_name* 的传输的信息。*agent_name* 的值不区分大小写，代理名称将转换为大写。

如果使用 **srcagent** 查询，那么不能使用 **agent** 查询。

destagent=(agent_name)

请求目标代理为 *agent_name* 的传输的信息。*agent_name* 的值不区分大小写，代理名称将转换为大写。

如果使用 **destagent** 查询，那么不能使用 **agent** 查询。

agent=(agent_name)

请求源代理和/或目标代理为 *agent_name* 的传输的信息。 *agent_name* 的值不区分大小写，代理名称将转换为大写。

如果使用 **agent** 查询，那么不能使用 **srcagent** 或 **destagent** 查询。

status=(status_value)

请求其传输状态为 *status_value* 的传输的信息。 *status_value* 的值区分大小写，是用方括号 ([]) 括起的逗号分隔列表。逗号分隔列表包含以下一个或多个值：

- **submitted**
- **started**
- **success**
- **partial success**
- **cancelled**
- **failure**

metadata=(metadata_info)

请求其元数据中包含 *metadata_info* 的传输的信息。

metadata_info 的值采用以下某种格式：

name

元数据名称/值对的名称部分。如果传输的元数据具有该名称及任何值，那么该传输与查询匹配。

name=value

元数据名称/值对。如果传输的元数据具有该名称及该值，那么该传输与查询匹配。

endafter=(date)

请求在 *date* 值指定的日期之后完成的传输的信息。 *date* 的值采用以下某种格式：

yyyy-MM-ddTHH:mm:ss

日期和时间。例如，2010-08-26T12:25:40。

yyyy-MM-ddTHH:mm

日期和时间，不包括秒数。例如，2010-08-26T12:25，该值会估算为 2010-08-26T12:25:00。

yyyy-MM-ddTHH

日期和时间，不包括秒数和分钟数。例如，2010-08-26T12，该值会估算为 2010-08-26T12:00:00。

yyyy-MM-dd

日期。例如，2010-08-26，该值会估算为 2010-08-26T00:00:00。

yyyy-MM

日期，不包括天数。例如，2010-08，该值会估算为 2010-07-31T23:59:59。

yyyy

年份。例如，2010，该值会估算为 2009-12-31T23:59:59。

日期和时间以全球标准时间 (UTC) 表示。

您可以通过以下方法指定不同时区的日期：在日期后面添加四位数字并且前面附上加号 (+) 或减号 (-)，以指示 UTC 和您使用的时区之间的差异。例如，要指定时区为旧金山的太平洋夏令时时间 2010 年 8 月 26 日晚上 7 点（比 UTC 晚 7 小时），请使用以下值：2010-08-26T19:00-0700。

endbefore=(date)

请求在 *date* 值指定的日期之前完成的传输的信息。 *date* 的值采用以下某种格式：

yyyy-MM-ddTHH:mm:ss

日期和时间。例如，2010-08-26T12:25:40。

yyyy-MM-ddTHH:mm

日期和时间，不包括秒数。例如，2010-08-26T12:25，该值会估算为 2010-08-26T12:25:00。

yyyy-MM-ddTHH

日期和时间，不包括秒数和分钟数。例如，2010-08-26T12，该值会估算为 2010-08-26T12:00:00。

yyyy-MM-dd

日期。例如，2010-08-26，该值会估算为 2010-08-26T00:00:00。

yyyy-MM

日期，不包括天数。例如，2010-08，该值会估算为 2010-07-31T23:59:59。

yyyy

年份。例如，2010，该值会估算为 2009-12-31T23:59:59。

日期和时间以全球标准时间 (UTC) 表示。

您可以通过以下方法指定不同时区的日期：在日期后面添加四位数字并且前面附上加号 (+) 或减号 (-)，以指示 UTC 和您使用的时区之间的差异。例如，要指定时区为旧金山的太平洋夏令时时间 2010 年 8 月 26 日晚上 7 点（比 UTC 晚 7 小时），请使用以下值：2010-08-26T19:00-0700。

startafter=(date)

请求在 *date* 值指定的日期之后开始的传输的信息。 *date* 的值采用以下某种格式：

yyyy-MM-ddTHH:mm:ss

日期和时间。例如，2010-08-26T12:25:40。

yyyy-MM-ddTHH:mm

日期和时间，不包括秒数。例如，2010-08-26T12:25，该值会估算为 2010-08-26T12:25:00。

yyyy-MM-ddTHH

日期和时间，不包括秒数和分钟数。例如，2010-08-26T12，该值会估算为 2010-08-26T12:00:00。

yyyy-MM-dd

日期。例如，2010-08-26，该值会估算为 2010-08-26T00:00:00。

yyyy-MM

日期，不包括天数。例如，2010-08，该值会估算为 2010-07-31T23:59:59。

yyyy

年份。例如，2010，该值会估算为 2009-12-31T23:59:59。

日期和时间以全球标准时间 (UTC) 表示。

您可以通过以下方法指定不同时区的日期：在日期后面添加四位数字并且前面附上加号 (+) 或减号 (-)，以指示 UTC 和您使用的时区之间的差异。例如，要指定时区为旧金山的太平洋夏令时时间 2010 年 8 月 26 日晚上 7 点（比 UTC 晚 7 小时），请使用以下值：2010-08-26T19:00-0700。

startbefore=(date)

请求在 *date* 值指定的日期之前开始的传输的信息。 *date* 的值采用以下某种格式：

yyyy-MM-ddTHH:mm:ss

日期和时间。例如，2010-08-26T12:25:40。

yyyy-MM-ddTHH:mm

日期和时间，不包括秒数。例如，2010-08-26T12:25，该值会估算为 2010-08-26T12:25:00。

yyyy-MM-ddTHH

日期和时间，不包括秒数和分钟数。例如，2010-08-26T12，该值会估算为 2010-08-26T12:00:00。

yyyy-MM-dd

日期。例如，2010-08-26，该值会估算为 2010-08-26T00:00:00。

yyyy-MM

日期，不包括天数。例如，2010-08，该值会估算为 2010-07-31T23:59:59。

yyyy

年份。例如，2010，该值会估算为 2009-12-31T23:59:59。

日期和时间以全球标准时间 (UTC) 表示。

您可以通过以下方法指定其他时区的日期：在列出的任何格式的日期末尾添加 Z。Z 的值是一个四位数，用于指示 UTC 和您使用的时区之间的差异。例如，要指定时区为旧金山的太平洋夏令时间 2010 年 8 月 26 日晚上 7 点（比 UTC 晚 7 小时），请使用以下值：2010-08-26T19:00-0700。

srcfile=(file_path)

请求完整源文件路径为 *file_path* 的传输的信息。区分大小写。

如果文件路径中包含空格字符，那么该字符在查询中必须以字符串 %20 表示。

destfile=(file_path)

请求完整目标文件路径为 *file_path* 的传输的信息。区分大小写。

如果文件路径中包含空格字符，那么该字符在查询中必须以字符串 %20 表示。

jobname=(job_name)

请求其作业名为 *job_name* 的传输的信息。作业名区分大小写。

returncode=(return_code)

请求其返回码为 *return_code* 的传输的信息。传输的返回码是一个正整数。有关可能的返回码的列表，请参阅 [第 405 页的『IBM MQ Managed File Transfer 的返回码』](#)。

transferid=(transfer_ID)

可选。传输标识是用于标识将文件传输到文件空间的传输的唯一十六进制字符串（共 48 个字符）。不区分大小写。

结果格式参数**sortby=(sort_by_values)**

指定对结果进行排序所依据的值。对于传输查询，*sort_by_value* 的值是以下某个值：

- srcagent
- destagent
- status
- startdate
- enddate
- jobname

缺省情况下，结果按 **startdate** 排序。

sort=(sort_value)

指定返回的结果按 **sortby** 查询的指定值以升序还是降序进行排序。*sort_value* 的值是以下某个值：

- ascending
- descending

仅当已指定 **sortby** 查询时，才能指定 **sort** 查询。

start=(start_value)

指定要返回的第一个结果的索引。*start_value* 的值是 0 或正整数。Web 网关找到的第一个结果的索引为 0。

count=(count_value)

指定要返回的结果数。count_value 的值是小于 100 的正整数。一次只能返回 100 个结果。

示例

例如，要使用 POST 请求将文件资源传输到名为 ACCOUNTS 且使用代理队列管理器 DEPT1 的目标代理，请使用以下 URI：

`http://example.org/wmqfte/file/agent/ACCOUNTS@DEPT1/`

在本示例中：

- `http://example.org` 是主机系统。
- `/wmqfte` 指示该 URI 是一个 IBM MQ Managed File Transfer URI。
- `/file` 指示要查找地址的资源是文件资源。
- `/agent/ACCOUNTS@DEPT1/` 是标识标记。该标识标记是目标类型（本例中为 agent）、目标代理名称（本例中为 ACCOUNTS）和前缀为 @ 符号的目标代理队列管理器名称（本例中为 @DEPT1）的组合。

例如，查找传输资源的地址：

`http://example.org/wmqfte/transfer/
414d5120514d5f4c4d343336303920201159c54820027102`

在本示例中：

- `http://example.org` 是主机系统。
- `/wmqfte` 指示该 URI 是一个 IBM MQ Managed File Transfer URI。
- `/transfer` 指示要查找地址的资源是传输资源。
- `/414d5120514d5f4c4d343336303920201159c54820027102` 是标识标记，在本例中为十六进制的传输标识。

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 312 页的『使用 IBM MQ Managed File Transfer 服务 Web 网关』](#)

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

[第 313 页的『示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 908 页的『Web 网关 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

[第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』](#)

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

[第 918 页的『使用 Web 网关的内容类型』](#)

向 IBM MQ Managed File Transfer Web 网关提交的文件传输请求必须对应于特定的介质类型。来自 Web 网关的响应具有介质类型 `application/xml` 或 `application/json`。

[第 919 页的『响应格式：XML 和 JSON』](#)

IBM MQ Managed File Transfer Web 网关采用以下两种格式之一返回查询的响应：XML 或 JavaScript Object Notation (JSON)。

[第 410 页的『HTTP 响应代码』](#)

向 IBM MQ Managed File Transfer Web 网关发出的请求的 HTTP 响应中会返回状态码。

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

使用 Web 网关的内容类型

向 IBM MQ Managed File Transfer Web 网关提交的文件传输请求必须对应于特定的介质类型。来自 Web 网关的响应具有介质类型 application/xml 或 application/json。

请求

使用 HTTP 传输至 IBM MQ Managed File Transfer 的内容必须采用下表中的格式之一。

表 73: 接受不同介质类型的 MQMFT 资源和 HTTP 动词		
介质类型	有效的 MQMFT 资源	允许的动词
multipart/form-data	文件（多文件传输或通过元数据传输）	POST、GET 和 DELETE
application/xml	传输和文件空间	POST、GET 和 DELETE

当作为多部分请求的一部分 POST 文件时，可以在每个多部分范围内使用任何介质类型。文件的介质类型决定以二进制还是文本方式进行文件传输，除非使用 x-fte-type 头来覆盖方式。

表 74: 缺省情况下对不同介质类型使用的传输方式	
介质类型	使用的传输方式
text/*	文本
application/xml	二进制
任何其他介质类型	二进制

响应体

Web 网关可能会返回介质类型为 application/xml 或 application/json 的响应，用于响应文件上载请求（FILE 资源的 POST）和传输状态请求（TRANSFER 资源的 GET）。有关 JSON 和 XML 响应格式的更多信息，请参阅[第 919 页的『响应格式：XML 和 JSON』](#)。Web 网关可能会返回具有任何介质类型的响应，用于响应文件下载请求（FILESPACE 资源的 GET）。

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 312 页的『使用 IBM MQ Managed File Transfer 服务 Web 网关』](#)

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

[第 313 页的『示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 908 页的『Web 网关 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

[第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』](#)

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

[第 911 页的『使用 Web 网关时的统一资源标识语法』](#)

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

[第 919 页的『响应格式：XML 和 JSON』](#)

IBM MQ Managed File Transfer Web 网关采用以下两种格式之一返回查询的响应：XML 或 JavaScript Object Notation (JSON)。

[第 410 页的『HTTP 响应代码』](#)

向 IBM MQ Managed File Transfer Web 网关发出的请求的 HTTP 响应中会返回状态码。

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

响应格式：XML 和 JSON

IBM MQ Managed File Transfer Web 网关采用以下两种格式之一返回查询的响应：XML 或 JavaScript Object Notation (JSON)。

通过在请求中包含 **Accept: *return-type*** 头或在 URI 中包含查询 **accept=*return-type***，可指定从 Web 网关返回的响应的格式。还可以使用 Web 应用程序来解析 XML 或 JSON 响应的内容，并以适当的格式呈现给 Web 用户。

缺省格式是 XML。如果同时使用 **Accept:** 头和 URI 中的查询 **accept=** 指定格式，那么 Web 网关将以 URI 中的查询指定的格式返回响应。

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 312 页的『使用 IBM MQ Managed File Transfer 服务 Web 网关』](#)

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

[第 313 页的『示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 920 页的『传输查询响应格式』](#)

当您从 IBM MQ Managed File Transfer Web 网关请求一个或多个传输的状态时会以 JSON 或 XML 格式返回响应。

[第 925 页的『文件空间查询的响应格式』](#)

从 IBM MQ Managed File Transfer 服务 Web 网关请求文件空间中的部分或全部文件的列表时，将根据您使用 **Accept:** 头指定的内容以 JSON 或 XML 格式返回响应。

[第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』](#)

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

[第 911 页的『使用 Web 网关时的统一资源标识语法』](#)

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

[第 918 页的『使用 Web 网关的内容类型』](#)

向 IBM MQ Managed File Transfer Web 网关提交的文件传输请求必须对应于特定的介质类型。来自 Web 网关的响应具有介质类型 application/xml 或 application/json。

[第 410 页的『HTTP 响应代码』](#)

向 IBM MQ Managed File Transfer Web 网关发出的请求的 HTTP 响应中会返回状态码。

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

传输查询响应格式

当您从 IBM MQ Managed File Transfer Web 网关请求一个或多个传输的状态时会以 JSON 或 XML 格式返回响应。

XML

以下示例显示简单传输查询的 XML 响应格式。

```
<transfers xsi:noNamespaceSchemaLocation="WebTransferStatus.xsd">
  <transfer end-time="2010-08-26T12:00:00.260Z"
    start-time="2010-08-26T11:55:00.076Z"
    status="Success"
    id="414d51205745422e46544520202020c1a1a34b03720120">
    <source>
      <agent qmgr="QM_JUPITER" name="AGENT_CALLISTO"/>
      <metadata>
        <key value="FIRST_JOB" name="com.ibm.wmqfte.JobName"/>
        <key value="AGENT_CALLISTO" name="com.ibm.wmqfte.SourceAgent"/>
        <key value="AGENT_EUROPA" name="com.ibm.wmqfte.DestinationAgent"/>
        <key value="serenity.example.com."
          name="com.ibm.wmqfte.OriginatingHost"/>
        <key value="user1" name="com.ibm.wmqfte.MqmdUser"/>
        <key value="414d51205745422e46544520202020c1a1a34b03720120"
          name="com.ibm.wmqfte.TransferId"/>
        <key value="user1" name="com.ibm.wmqfte.OriginatingUser"/>
      </metadata>
    </source>
    <destination>
      <agent qmgr="QM_JUPITER" name="AGENT_EUROPA"/>
      <metadata>
        <key value="FIRST_JOB" name="com.ibm.wmqfte.JobName"/>
        <key value="AGENT_CALLISTO" name="com.ibm.wmqfte.SourceAgent"/>
        <key value="AGENT_EUROPA" name="com.ibm.wmqfte.DestinationAgent"/>
        <key value="user1" name="com.ibm.wmqfte.MqmdUser"/>
        <key value="serenity.example.com."
          name="com.ibm.wmqfte.OriginatingHost"/>
        <key value="user1" name="com.ibm.wmqfte.OriginatingUser"/>
        <key value="414d51205745422e46544520202020c1a1a34b03720120"
          name="com.ibm.wmqfte.TransferId"/>
      </metadata>
    </destination>
    <stats retry-count="0" file-warnings="0" file-failures="0"
      bytes-transferred="259354303"/>
    <result text="BFGRP0032I: The file transfer request has successfully completed."
      code="0"/>
    <transfer-set>
      <file result-code="0" mode="text">
        <source-file name="/home/user1/output.zip">
          <attribute-values last-modified="2010-08-19T14:16:57.000Z"
            file-size="259354303" disposition="leave"
            checksum-value="98611a272a27d373f92d73a08cf0d4f4"
            checksum-method="MD5"/>
        </source-file>
        <destination-file name="/tmp/output.zip">
          <attribute-values last-modified="2010-08-26T12:00:00.000Z"
            file-size="259354303" exists-action="error"
            checksum-value="98611a272a27d373f92d73a08cf0d4f4"
            checksum-method="MD5"/>
        </destination-file>
      </file>
    </transfer-set>
  </transfer>
</transfers>
```

```
</transfer>
</transfers>
```

JSON

以下示例显示简单传输查询的 JSON 响应格式。

```
{
  "transfers" : {
    "transfer" : {
      "end-time" : "2010-08-26T12:00:00.260Z",
      "status" : "Success",
      "start-time" : "2010-08-26T11:55:00.076Z",
      "id" : "414d51205745422e46544520202020c1a1a34b03720120",
      "result" : {
        "code" : "0",
        "text" : "BFGRP0032I: The file transfer request has successfully completed."
      }
    },
    "destination" : {
      "metadata" : {
        "key" : [
          {
            "name" : "com.ibm.wmqfte.JobName",
            "value" : "FIRST_JOB"
          },
          {
            "name" : "com.ibm.wmqfte.SourceAgent",
            "value" : "AGENT_CALLISTO"
          },
          {
            "name" : "com.ibm.wmqfte.DestinationAgent",
            "value" : "AGENT_EUROPA"
          },
          {
            "name" : "com.ibm.wmqfte.MqmdUser",
            "value" : "user1"
          },
          {
            "name" : "com.ibm.wmqfte.OriginatingHost",
            "value" : "serenity.example.com."
          },
          {
            "name" : "com.ibm.wmqfte.OriginatingUser",
            "value" : "user1"
          },
          {
            "name" : "com.ibm.wmqfte.TransferId",
            "value" : "414d51205745422e46544520202020c1a1a34b03720120"
          }
        ]
      },
      "agent" : {
        "name" : "AGENT_EUROPA",
        "qmgr" : "QM_JUPITER"
      }
    },
    "stats" : {
      "bytes-transferred" : "259354303",
      "retry-count" : "0",
      "file-warnings" : "0",
      "file-failures" : "0"
    },
    "transfer-set" : {
      "file" : {
        "result-code" : "0",
        "mode" : "text",
        "source-file" : {
          "name" : "\\home\\user1\\output.zip",

```

```

        "attribute-values" : {
            "last-modified" : "2010-08-19T14:16:57.000Z",
            "checksum-value" : "98611a272a27d373f92d73a08cf0d4f4",
            "checksum-method" : "MD5",
            "file-size" : "259354303",
            "disposition" : "leave"
        }
    }
    "destination-file" : {
        "name" : "\\tmp\\output.zip",
        "attribute-values" : {
            "exists-action" : "error",
            "last-modified" : "2010-08-26T12:00:00.000Z",
            "checksum-value" : "98611a272a27d373f92d73a08cf0d4f4",
            "checksum-method" : "MD5",
            "file-size" : "259354303"
        }
    }
}
}
"source" : {
    "metadata" : {
        "key" : [
            {
                "name" : "com.ibm.wmqfte.JobName",
                "value" : "FIRST_JOB"
            },
            {
                "name" : "com.ibm.wmqfte.SourceAgent",
                "value" : "AGENT_CALLISTO"
            },
            {
                "name" : "com.ibm.wmqfte.DestinationAgent",
                "value" : "AGENT_EUROPA"
            },
            {
                "name" : "com.ibm.wmqfte.OriginatingHost",
                "value" : "serenity,example.com."
            },
            {
                "name" : "com.ibm.wmqfte.MqmdUser",
                "value" : "user1"
            },
            {
                "name" : "com.ibm.wmqfte.TransferId",
                "value" : "414d51205745422e46544520202020c1a1a34b03720120"
            },
            {
                "name" : "com.ibm.wmqfte.OriginatingUser",
                "value" : "user1"
            }
        ]
    }
    "agent" : {
        "name" : "AGENT_CALLISTO",
        "qmgr" : "QM_JUPITER"
    }
}
}
}
}

```

了解传输查询的响应

XML 响应格式中元素和属性的名称与 JSON 响应格式中对象的名称相同。以下列表中描述了这些元素、属性和对象：

transfers

包含与查询匹配的所有传输信息的组。

transfer

包含单个传输信息的组。

属性或对象	描述
end-time	传输完成的时间，以全球标准时间表示。
start-time	传输开始的时间，以全球标准时间表示。
状态	传输状态。
标识	传输的唯一十六进制标识。

源

包含有关传输的源信息的组。

目的地

包含有关传输的目标信息的组。

代理程序

指定代理的名称及其队列管理器。该元素或对象用于元素或对象 **source** 中时会指定源代理信息，而用于元素或对象 **destination** 中时会指定目标代理信息。

属性或对象	描述
名	代理的名称。
qmgr	代理连接到的队列管理器。

元数据

在名称/值对中包含传输信息的组。

key

指定名称/值对。

属性或对象	描述
名	元数据片段的标识。
值	元数据片段的值。

统计信息

指定有关整个传输的信息。

属性或对象	描述
retry-count	传输进入恢复状态并由代理重试的次数。
file-warnings	传输集内，在传输时生成警告但仍成功传输的文件数。
file-failures	传输集内传输失败的文件数。
bytes-transferred	在该传输中传输的字节数。

结果

指定传输的返回码和补充信息。

属性或对象	描述
code	传输的返回码。有关更多信息，请参阅第 405 页的『 IBM MQ Managed File Transfer 的返回码 』。
文本	传输的补充信息。

transfer-set

包含有关已传输文件的信息的组。

文件

包含有关传输内一个文件的信息的组。

属性或对象	描述
result-code	单个文件的传输的返回码。有关更多信息，请参阅 第 409 页的『针对传输中文件的返回码』 。
mode	传输方式。有效值包括： • 文本 • 二进制

source-file

指定源文件的名称。

属性或对象	描述
名	源系统中文件的名称。

destination-file

指定目标文件的名称。

属性或对象	描述
名	目标系统中文件的名称。

attribute-values

指定有关正在传输的文件的其他信息。在元素或对象 **source-file** 中使用时，此元素或对象指定有关源系统上文件的信息；在元素或对象 **destination-file** 中使用时，此元素或对象指定有关目标系统上文件的信息。

属性或对象	描述
file-size	文件大小。
exists-action	指定当目标文件已存在时要执行的操作。有效值包括： • Error • 覆盖 仅当属性值元素或对象用于目标文件元素或对象中时，该属性才有效。
disposition	指定传输完成后要对源文件执行的操作。有效值包括： • delete • leave 仅当属性值元素或对象用于源文件元素或对象中时，该属性才有效。
checksum-method	用于生成该文件校验和值的方法。
checksum-value	文件的校验和值。
last-modified	最后修改文件的时间，以全球标准时间表示。

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

第 307 页的『[Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑](#)』

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

第 312 页的『[使用 IBM MQ Managed File Transfer 服务 Web 网关](#)』

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

第 313 页的『[示例 HTTP 流](#)』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

相关任务

第 315 页的『[示例：使用 HTTP 请求查看文件传输的状态](#)』

通过使用 IBM MQ Managed File Transfer Web 网关来提交请求，可以查看文件传输的状态。Web 网关将返回 XML 格式的信息，其中描述了指定传输的当前状态。要使用 Web 网关来查看文件传输的状态，您的 IBM MQ Managed File Transfer 网络中必须存在数据库记录器。

第 316 页的『[示例：使用 HTTP 请求查询多个文件传输](#)』

通过 IBM MQ Managed File Transfer Web 网关提交请求，可查询多个文件传输的状态。Web 网关会以 XML 或 JSON 格式返回信息，其中描述与查询匹配的传输的状态。

相关参考

第 919 页的『[响应格式：XML 和 JSON](#)』

IBM MQ Managed File Transfer Web 网关采用以下两种格式之一返回查询的响应：XML 或 JavaScript Object Notation (JSON)。

第 925 页的『[文件空间查询的响应格式](#)』

从 IBM MQ Managed File Transfer 服务 Web 网关请求文件空间中的部分或全部文件的列表时，将根据您使用 Accept: 头指定的内容以 JSON 或 XML 格式返回响应。

第 937 页的『[文件空间信息响应格式](#)』

通过 IBM MQ Managed File Transfer 服务 Web 网关请求有关文件空间的定义和属性的信息时，将以 XML 格式或 JSON 格式返回响应。该 XML 响应符合模式 FileSpaceInfo.xsd，该模式位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

第 908 页的『[Web 网关 API 引用](#)』

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

第 930 页的『[Web 网关管理 API 引用](#)』

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

文件空间查询的响应格式

从 IBM MQ Managed File Transfer 服务 Web 网关请求文件空间中的部分或全部文件的列表时，将根据您使用 Accept: 头指定的内容以 JSON 或 XML 格式返回响应。

XML

以下示例显示简单文件空间查询的 XML 响应格式。

```
<fileSpaces xsi:noNamespaceSchemaLocation="WebFileSpaceList.xsd">
  <fileSpace size="1" name="james">
    <file fileLink="/wmqfte/filespace/james/414d51205745422e46544520202020c1a1a34b03720120/file.zip"
      transferLink="/wmqfte/transfer/414d51205745422e46544520202020c1a1a34b03720120"
      transferID="414d51205745422e46544520202020c1a1a34b03720120"
      name="/tmp/ae55bc7">
      <attribute-values mode="text" time="2010-08-26T19:00:02.000Z"
        file-size="259354303"
        checksum-value="98611a272a27d373f92d73a08cf0d4f4"
        checksum-method="none"/>
    </file>
  </fileSpace>
</fileSpaces>
```

该 XML 响应符合模式 WebFileSpaceList.xsd，该模式位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

JSON

以下示例显示简单文件空间查询的 JSON 响应格式。

```
{
  "fileSpaces" : {
    "fileSpace" : {
      "name" : "james",
      "size" : "1",
      "file" : {
        "transferLink" : "\\wmqfte\\transfer\\414d51205745422e46544520202020c1a1a34b03720120",
        "fileLink" : "\\wmqfte\\fileSpace\\1234\\414d51205745422e46544520202020c1a1a34b03720120\\file.zip",
        "name" : "\\tmp\\ae55bc7",
        "transferID" : "414d51205745422e46544520202020c1a1a34b03720120",
        "attribute-values" : {
          "checksum-value" : "98611a272a27d373f92d73a08cf0d4f4",
          "checksum-method" : "none",
          "time" : "2010-08-26T19:00:02.000Z",
          "file-size" : "259354303",
          "mode" : "text"
        }
      }
    }
  }
}
```

了解文件空间查询的响应

XML 响应格式中元素和属性的名称与 JSON 响应格式中对象的名称相同。以下列表中描述了这些元素、属性和对象：

fileSpaces

包含文件空间信息的组。

文件空间

包含单个文件空间信息的组。

属性或对象	描述
size	查询所返回的文件空间中的文件数。
名	文件空间的名称。

文件

包含文件信息的组。

属性或对象	描述
fileLink	用于从文件空间下载文件的 URI 的一部分。 用于下载文件的完整 URI 是 <i>host-name/fileLink</i>
transferLink	URI 的一部分，该 URI 用于查看将文件放入文件空间的传输的信息。 用于查看传输信息的完整 URI 是 <i>host-name/transferLink</i>
transferID	将文件放入文件空间的传输的唯一十六进制标识。
名	托管文件空间的系统上文件的路径。

attribute-values

指定有关正在传输的文件的其他信息。

属性或对象	描述
file-size	文件大小。
mode	传输方式。有效值包括： • 文本 • 二进制
checksum-method	用于生成该文件校验和值的方法。
checksum-value	文件的校验和值。
time	文件传输到文件空间的时间，以全球标准时间表示。
integrity-check-result	文件的完整性检查结果。有效值包括： • 确定 • MISSING_FILESYSTEM • MISSING_DATABASEENTRY

相关概念

第 305 页的『[Web 网关的方案](#)』

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

第 307 页的『[Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑](#)』

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

第 312 页的『[使用 IBM MQ Managed File Transfer 服务 Web 网关](#)』

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

第 313 页的『[示例 HTTP 流](#)』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

相关任务

第 322 页的『[示例：列出文件空间中的所有文件](#)』

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可列出文件空间的内容。Web 网关会以 XML 或 JSON 格式返回响应，其中列出文件空间的内容。如果您是文件空间的所有者或具有安全角色 wmqfte-admin，那么您有权列出文件空间的内容。

第 323 页的『[示例：列出文件空间中的特定文件子集](#)』

通过向 IBM MQ Managed File Transfer Web 网关提交包含查询的 HTTP 请求，可查询文件空间的内容。Web 网关会以 XML 或 JSON 格式返回响应，其中仅描述文件空间中与查询匹配的文件。

相关参考

第 919 页的『[响应格式：XML 和 JSON](#)』

IBM MQ Managed File Transfer Web 网关采用以下两种格式之一返回查询的响应：XML 或 JavaScript Object Notation (JSON)。

第 920 页的『[传输查询响应格式](#)』

当您从 IBM MQ Managed File Transfer Web 网关请求一个或多个传输的状态时会以 JSON 或 XML 格式返回响应。

第 937 页的『[文件空间信息响应格式](#)』

通过 IBM MQ Managed File Transfer 服务 Web 网关请求有关文件空间的定义和属性的信息时，将以 XML 格式或 JSON 格式返回响应。该 XML 响应符合模式 FileSpaceInfo.xsd，该模式位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

第 908 页的『[Web 网关 API 引用](#)』

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

第 930 页的『Web 网关管理 API 引用』

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

HTTP 响应代码

向 IBM MQ Managed File Transfer Web 网关发出的请求的 HTTP 响应中会返回状态码。

Web 网关返回的响应头中包含 HTTP 响应代码。以下示例中的 HTTP 头包含 HTTP 响应代码 200 OK:

```
HTTP/1.1 200 OK
Server: WAS/6.0
Content-length: 0
```

下表描述了 HTTP 响应代码的可能值，以及可返回的关联 IBM MQ Managed File Transfer 错误代码的示例。有关 IBM MQ Managed File Transfer 错误代码的更多信息，请参阅[诊断消息](#)。

表 75: HTTP 响应代码		
HTTP 响应代码	IBM MQ Managed File Transfer 错误代码示例	示例描述
200 OK	None	已正确处理有效请求，且已选择性地向用户提供响应。
202 Accepted	None	已正确处理有效请求，但 IBM MQ Managed File Transfer 不保证请求的操作已完成。 例如，文件上传传输请求已处理并提交到 IBM MQ Managed File Transfer 代理，但尚未进行传输。
400 Bad Request	BFGWI0001	由于 URI 缺少资源类型，因此该 URI 无效。
403 Forbidden	BFGWI0056	没有为该用户定义 IBM MQ 消息描述符 (MQMD) 用户标识。
404 Not Found	BFGWI0015	找不到请求的资源。
405 Method Not Allowed	BFGWI0016	请求的资源不支持请求中所使用的 HTTP 谓词。 例如，对仅支持 POST 或 DELETE 的资源使用了 GET 操作。
410 Resource Gone	BFGWI0031	所请求的资源不再可用。例如，请求的文件已从文件空间中删除。
413 Request Entity Too Large	BFGWI0026	请求包含的某个文件过大，服务器无法处理。
415 Unsupported Media Type	BFGWI0017	接收的请求具有 Content-type HTTP 头指定的介质类型，但该类型不受支持。
500 Internal Server Error	BFGWI0018	处理请求时迁到内部错误。已生成 FFDC 或 ABEND 文件。

表 75: HTTP 响应代码 (继续)		
HTTP 响应代码	IBM MQ Managed File Transfer 错误代码示例	示例描述
502 Bad Gateway	BFGWI0019	由于 IBM MQ Managed File Transfer 外部发生错误，无法完成请求。例如，IBM MQ 队列管理器不可用。
503 Service Unavailable	BFGWI0020	该目标暂时不可用。例如，IBM MQ 队列已满。
504 Gateway Timeout	BFGWI0021	由于超出 IBM MQ Managed File Transfer 或 HTTP 客户机设定的时间限制，尝试完成请求超时。

相关概念

第 412 页的『[对 Web 网关进行故障诊断](#)』

使用以下参考信息和示例可帮助您诊断 Web 网关返回的错误。

第 305 页的『[Web 网关的方案](#)』

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

第 307 页的『[Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑](#)』

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

第 312 页的『[使用 IBM MQ Managed File Transfer 服务 Web 网关](#)』

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

第 313 页的『[示例 HTTP 流](#)』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

相关参考

第 908 页的『[Web 网关 API 引用](#)』

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

第 909 页的『[使用 Web 网关时的 HTTP 头和 HTML 表单字段](#)』

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

第 911 页的『[使用 Web 网关时的统一资源标识语法](#)』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 /wmqfte。

第 918 页的『[使用 Web 网关的内容类型](#)』

向 IBM MQ Managed File Transfer Web 网关提交的文件传输请求必须对应于特定的介质类型。来自 Web 网关的响应具有介质类型 application/xml 或 application/json。

第 919 页的『[响应格式: XML 和 JSON](#)』

IBM MQ Managed File Transfer Web 网关采用以下两种格式之一返回查询的响应：XML 或 JavaScript Object Notation (JSON)。

第 930 页的『[Web 网关管理 API 引用](#)』

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

Web 网关管理 API 引用

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

该引用信息描述用于管理 Web 网关对象（如文件空间）的 API。有关用于非管理任务的 API 的信息，请参阅 [第 908 页的『Web 网关 API 引用』](#)。

资源类型

该规范支持以下 IBM MQ Managed File Transfer 对象类型：

文件空间

包含已发送至与该文件空间关联的用户或组的文件的逻辑区域。

用户

Web 用户标识与 WebSphere MQ 消息描述符 (MQMD) 用户标识之间的映射集。这些映射控制用于文件传输请求的 MQMD 用户标识。

HTTP 动词

该规范支持下表中的 HTTP 动词。

HTTP 动词	IBM MQ Managed File Transfer 操作
POST	- 创建文件空间的实例。 - 修改现有文件空间的配置。 - 在 Web 用户标识与 MQMD 用户标识之间创建映射集。 - 在 Web 用户标识与 MQMD 用户标识之间修改映射集，或向映射集添加内容。
GET	- 查看文件空间的当前配置。该配置包含文件空间名称、最大文件空间大小以及有权对文件空间执行写操作的人员列表。 - 列出当前存在的所有文件空间。 - 查看 Web 用户标识与 MQMD 用户标识之间的当前映射集。
DELETE	- 删除文件空间的实例。 - 删除 Web 用户标识与 MQMD 用户标识之间的映射集或映射子集。

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 327 页的『管理 IBM MQ Managed File Transfer 服务 Web 网关』](#)

您可以创建和删除文件空间，控制对个别文件空间具有访问权的用户。

[第 329 页的『用于管理的示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 931 页的『用于管理 Web 网关的 HTTP 头』](#)

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

[第 932 页的『用于管理 Web 网关的统一资源标识语法』](#)

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 IBM MQ URI 区分开。建议使用上下文根 /wmqfte。用于管理任务的 URI 通过术语 /admin 与现有 IBM MQ Managed File Transfer URI 区分开。

[第 934 页的『用于管理 Web 网关的内容类型』](#)

您提交到 IBM MQ Managed File Transfer Web 网关管理 API 的 HTTP 请求必须具有介质类型 application/xml。来自 Web 网关的响应也具有介质类型 application/xml。

[第 935 页的『来自 Web 网关管理 API 的 HTTP 响应代码』](#)

向 IBM MQ Managed File Transfer Web 网关管理 API 发出的请求的 HTTP 响应中会返回状态码。

[第 939 页的『文件空间创建或更改请求格式』](#)

通过在 HTTP 请求中包含 XML 格式的内容，可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 FileSpaceInfo.xsd，此模式位于 WMQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

[第 942 页的『用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交请求，可以创建 Web 用户标识与 IBM MQ 消息描述符 (MQMD) 用户标识之间的映射集。此 HTTP 请求必须包含以下 XML 格式的内容。

[第 943 页的『文件空间管理日志记录格式』](#)

创建、更改或删除文件空间时，对文件空间的更改将记录在托管 Web 网关的应用程序服务器的事件日志中。这使得管理员能够查看对文件空间所做的更改。

[第 908 页的『Web 网关 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

用于管理 Web 网关的 HTTP 头

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

HTTP 约定是在定制头前添加 x-，后跟特定于产品的标识。IBM MQ Managed File Transfer 使用产品标识 fte-。有关 Web 网关 API 支持的头的详细信息，请参阅[第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』](#)。没有为管理用途定义任何其他头。

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 327 页的『管理 IBM MQ Managed File Transfer 服务 Web 网关』](#)

您可以创建和删除文件空间，控制对个别文件空间具有访问权的用户。

[第 329 页的『用于管理的示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

[第 932 页的『用于管理 Web 网关的统一资源标识语法』](#)

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 IBM MQ URI 区分开。建议使用上下文根 /wmqfte。用于管理任务的 URI 通过术语 /admin 与现有 IBM MQ Managed File Transfer URI 区分开。

第 934 页的『用于管理 Web 网关的内容类型』

您提交到 IBM MQ Managed File Transfer Web 网关管理 API 的 HTTP 请求必须具有介质类型 application/xml。来自 Web 网关的响应也具有介质类型 application/xml。

第 935 页的『来自 Web 网关管理 API 的 HTTP 响应代码』

向 IBM MQ Managed File Transfer Web 网关管理 API 发出的请求的 HTTP 响应中会返回状态码。

第 939 页的『文件空间创建或更改请求格式』

通过在 HTTP 请求中包含 XML 格式的内容，可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 FileSpaceInfo.xsd，此模式位于 WMQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

第 942 页的『用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式』

通过向 IBM MQ Managed File Transfer Web 网关提交请求，可以创建 Web 用户标识与 IBM MQ 消息描述符 (MQMD) 用户标识之间的映射集。此 HTTP 请求必须包含以下 XML 格式的内容。

第 943 页的『文件空间管理日志记录格式』

创建、更改或删除文件空间时，对文件空间的更改将记录在托管 Web 网关的应用程序服务器的事件日志中。这使得管理员能够查看对文件空间所做的更改。

第 908 页的『Web 网关 API 引用』

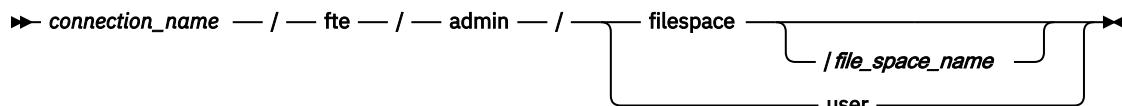
IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

用于管理 Web 网关的统一资源标识语法

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 IBM MQ URI 区分开。建议使用上下文根 /wmqfte。用于管理任务的 URI 通过术语 /admin 与现有 IBM MQ Managed File Transfer URI 区分开。

IBM MQ Managed File Transfer 资源之间会按各自的类型进行区分。资源的地址通过其资源类型和标识标记确定。

IBM MQ Managed File Transfer 管理统一资源标识语法图



参数

connection_name

必需。托管 Web 网关的服务器的主机名和（可选）端口。不区分大小写。

fte

必需。指示该 URI 指向 Web 网关。区分大小写。

admin

必需。指示您正在使用 Web 网关的管理功能。区分大小写。

file_space

指示您要查找文件空间资源的地址。有关文件空间的更多信息，请参阅第 339 页的『文件空间』。区分大小写。

参数 **file_space** 或 **user** 其中之一是必需的。

file_space_name

您要查找其地址的文件空间的名称。这是与文件空间关联的用户的名称。*file_space_name* 值的长度必须不超过 255 个字符。区分大小写。

仅当指定 **filespace** 时才适用。如果使用 HTTP 动词 GET，那么该参数可选；如果使用 POST 或 DELETE，那么该参数是必需的。如果使用 HTTP 动词 GET，并且未提供 *file_space_name* 的值，那么 Web 网关将返回所有文件空间的列表。

user

指示您要查找 Web 用户标识和 MQMD 用户标识之间映射集的地址。有关该映射集格式的更多信息，请参阅第 942 页的『[用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式](#)』。区分大小写。

参数 **filespace** 或 **user** 其中之一是必需的。

示例

例如，要查找用户 sarah 拥有的文件空间资源的地址，请使用以下 URI：

<http://example.org/wmqfte/admin/filespace/sarah/>

在本示例中：

- <http://example.org> 是主机系统。
- /wmqfte 指示 URI 是 IBM MQ Managed File Transfer URI。
- /admin 指示您要访问 Web 网关的管理功能。
- /filespace 指示要查找地址的资源是文件空间资源。
- /sarah/ 是标识标记。该标记是文件空间的名称，同时也是拥有该文件空间的用户的名称。

例如，要查找用户标识和 MQMD 标识之间映射的地址，请使用以下 URI：

<http://example.org/wmqfte/admin/user>

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 327 页的『管理 IBM MQ Managed File Transfer 服务 Web 网关』](#)

您可以创建和删除文件空间，控制对个别文件空间具有访问权的用户。

[第 329 页的『用于管理的示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

[第 931 页的『用于管理 Web 网关的 HTTP 头』](#)

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

[第 934 页的『用于管理 Web 网关的内容类型』](#)

您提交到 IBM MQ Managed File Transfer Web 网关管理 API 的 HTTP 请求必须具有介质类型 application/xml。来自 Web 网关的响应也具有介质类型 application/xml。

[第 935 页的『来自 Web 网关管理 API 的 HTTP 响应代码』](#)

向 IBM MQ Managed File Transfer Web 网关管理 API 发出的请求的 HTTP 响应中会返回状态码。

[第 939 页的『文件空间创建或更改请求格式』](#)

通过在 HTTP 请求中包含 XML 格式的内容，可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 FileSpaceInfo.xsd，此模式位于 WMQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

[第 942 页的『用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交请求，可以创建 Web 用户标识与 IBM MQ 消息描述符 (MQMD) 用户标识之间的映射集。此 HTTP 请求必须包含以下 XML 格式的内容。

[第 943 页的『文件空间管理日志记录格式』](#)

创建、更改或删除文件空间时，对文件空间的更改将记录在托管 Web 网关的应用程序服务器的事件日志中。这使得管理员能够查看对文件空间所做的更改。

[第 908 页的『Web 网关 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

用于管理 Web 网关的内容类型

您提交到 IBM MQ Managed File Transfer Web 网关管理 API 的 HTTP 请求必须具有介质类型 application/xml。来自 Web 网关的响应也具有介质类型 application/xml。

请求

作为对管理 API 的请求的一部分，使用 HTTP 传输至 IBM MQ Managed File Transfer 的内容必须采用下表中的格式之一。

介质类型	有效的 IBM MQ Managed File Transfer 资源	允许的动词
application/xml	文件空间	• POST（创建文件空间的实例） • GET（查看文件空间的当前配置） • DELETE（删除文件空间的实例）

响应体

如果 HTTP 请求成功，那么 Web 网关将返回介质类型为 application/xml 的响应。有关针对该响应的 XML 模式的详细信息，请参阅[第 937 页的『文件空间信息响应格式』](#)。

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 327 页的『管理 IBM MQ Managed File Transfer 服务 Web 网关』](#)

您可以创建和删除文件空间，控制对个别文件空间具有访问权的用户。

[第 329 页的『用于管理的示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

[第 931 页的『用于管理 Web 网关的 HTTP 头』](#)

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

第 932 页的『用于管理 Web 网关的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 IBM MQ URI 区分开。建议使用上下文根 /wmqfte。用于管理任务的 URI 通过术语 /admin 与现有 IBM MQ Managed File Transfer URI 区分开。

第 935 页的『来自 Web 网关管理 API 的 HTTP 响应代码』

向 IBM MQ Managed File Transfer Web 网关管理 API 发出的请求的 HTTP 响应中会返回状态码。

第 939 页的『文件空间创建或更改请求格式』

通过在 HTTP 请求中包含 XML 格式的内容，可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 FileSpaceInfo.xsd，此模式位于 WMQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

第 942 页的『用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式』

通过向 IBM MQ Managed File Transfer Web 网关提交请求，可以创建 Web 用户标识与 IBM MQ 消息描述符 (MQMD) 用户标识之间的映射集。此 HTTP 请求必须包含以下 XML 格式的内容。

第 943 页的『文件空间管理日志记录格式』

创建、更改或删除文件空间时，对文件空间的更改将记录在托管 Web 网关的应用程序服务器的事件日志中。这使得管理员能够查看对文件空间所做的更改。

第 908 页的『Web 网关 API 引用』

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

来自 Web 网关管理 API 的 HTTP 响应代码

向 IBM MQ Managed File Transfer Web 网关管理 API 发出的请求的 HTTP 响应中会返回状态码。

Web 网关返回的响应头中包含 HTTP 响应代码。以下示例中的 HTTP 头包含 HTTP 响应代码 200 OK:

```
HTTP/1.1 200 OK
Server: WAS/6.0
Content-length: 0
```

下表描述了 HTTP 响应代码的可能值，以及管理 API 可返回的一些其他 IBM MQ Managed File Transfer 错误代码。

表 76: HTTP 响应代码		
HTTP 响应代码	IBM MQ Managed File Transfer 错误代码示例	示例描述
200 OK	None	已正确处理有效请求，且已选择性地向用户提供响应。
202 Accepted	None	已正确处理有效请求，但 IBM MQ Managed File Transfer 不保证请求的操作已完成。
400 Bad Request	BFGWI0501	由于 URI 缺少管理资源类型，因此该 URI 无效。
404 Not Found	BFGWI0515	找不到请求的资源。例如，文件空间不存在。
405 Method Not Allowed	BFGWI0516	请求的资源不支持请求中所使用的 HTTP 谓词。 例如，/admin/filespace/ <i>file_space_name</i> / 上使用了 HEAD 方法，且有效方法仅包括 POST、GET 或 DELETE。

表 76: HTTP 响应代码 (继续)

HTTP 响应代码	IBM MQ Managed File Transfer 错误代码示例	示例描述
415 Unsupported Media Type	BFGWI0517	接收的管理请求具有 Content-type HTTP 头指定的介质类型, 但该类型不受支持。有关管理 API 支持的介质类型的更多信息, 请参阅主题第 934 页的『用于管理 Web 网关的内容类型』。
500 Internal Server Error	BFGWI0018	处理请求时迁到内部错误。已生成 FFDC 或 ABEND 文件。
502 Bad Gateway	BFGWI0019	由于 IBM MQ Managed File Transfer 外部发生错误, 无法完成请求。例如, IBM MQ 队列管理器不可用。
503 Service Unavailable	BFGWI0020	该目标暂时不可用。例如, IBM MQ 队列已满。
504 Gateway Timeout	BFGWI0021	由于超出 IBM MQ Managed File Transfer 或 HTTP 客户机设定的时间限制, 尝试完成请求超时。

有关 Web 网关可返回的其他 IBM MQ Managed File Transfer 错误响应代码的信息, 请参阅主题[诊断消息](#)。

相关概念

第 305 页的『Web 网关的方案』

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理, 并使用 HTTP 客户机检索传输的状态。

第 327 页的『管理 IBM MQ Managed File Transfer 服务 Web 网关』

您可以创建和删除文件空间, 控制对个别文件空间具有访问权的用户。

第 329 页的『用于管理的示例 HTTP 流』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

第 930 页的『Web 网关管理 API 引用』

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

第 931 页的『用于管理 Web 网关的 HTTP 头』

使用 HTTP 头, 可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关, 未定义任何用于管理 API 的头。

第 932 页的『用于管理 Web 网关的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 IBM MQ URI 区分开。建议使用上下文根 /wmqfte。用于管理任务的 URI 通过术语 /admin 与现有 IBM MQ Managed File Transfer URI 区分开。

第 934 页的『用于管理 Web 网关的内容类型』

您提交到 IBM MQ Managed File Transfer Web 网关管理 API 的 HTTP 请求必须具有介质类型 application/xml。来自 Web 网关的响应也具有介质类型 application/xml。

第 939 页的『文件空间创建或更改请求格式』

通过在 HTTP 请求中包含 XML 格式的内容，可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 `FileSpaceInfo.xsd`，此模式位于 WMQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

[第 942 页的『用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交请求，可以创建 Web 用户标识与 IBM MQ 消息描述符 (MQMD) 用户标识之间的映射集。此 HTTP 请求必须包含以下 XML 格式的内容。

[第 943 页的『文件空间管理日志记录格式』](#)

创建、更改或删除文件空间时，对文件空间的更改将记录在托管 Web 网关的应用程序服务器的事件日志中。这使得管理员能够查看对文件空间所做的更改。

[第 908 页的『Web 网关 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

管理响应和请求格式

IBM MQ Managed File Transfer Web 网关采用以下两种格式之一返回查询的响应：XML 或 JavaScript Object Notation (JSON)。您可以提交请求，仅以 XML 格式创建、修改和删除文件空间或者将用户名映射到 Web 网关的 MQMD 用户标识。

通过在请求中包含 `Accept: return-type` 头或在 URI 中包含查询 `accept=return-type`，可指定从 Web 网关返回的响应的格式。还可以使用 Web 应用程序来解析 XML 或 JSON 响应的内容，并以适当的格式呈现给 Web 用户。

缺省格式是 XML。如果同时使用 `Accept:` 头和 URI 中的查询 `accept=` 指定格式，那么 Web 网关将以 URI 中的查询指定的格式返回响应。

相关参考

[第 939 页的『文件空间创建或更改请求格式』](#)

通过在 HTTP 请求中包含 XML 格式的内容，可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 `FileSpaceInfo.xsd`，此模式位于 WMQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

[第 942 页的『用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交请求，可以创建 Web 用户标识与 IBM MQ 消息描述符 (MQMD) 用户标识之间的映射集。此 HTTP 请求必须包含以下 XML 格式的内容。

[第 937 页的『文件空间信息响应格式』](#)

通过 IBM MQ Managed File Transfer 服务 Web 网关请求有关文件空间的定义和属性的信息时，将以 XML 格式或 JSON 格式返回响应。该 XML 响应符合模式 `FileSpaceInfo.xsd`，该模式位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

[第 909 页的『使用 Web 网关时的 HTTP 头和 HTML 表单字段』](#)

通过使用 HTTP 头或 HTML 表单字段，可定制请求以创建或检索资源。每个参数都映射到 IBM MQ Managed File Transfer 的属性或函数。

[第 911 页的『使用 Web 网关时的统一资源标识语法』](#)

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 WebSphere MQ URI 区分开。建议使用上下文根 `/wmqfte`。

文件空间信息响应格式

通过 IBM MQ Managed File Transfer 服务 Web 网关请求有关文件空间的定义和属性的信息时，将以 XML 格式或 JSON 格式返回响应。该 XML 响应符合模式 `FileSpaceInfo.xsd`，该模式位于 MQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

XML

以下示例显示了简单文件空间信息 XML 响应的格式。

```
<filespaces xsi:noNamespaceSchemaLocation="FileSpaceInfo.xsd">
  <filespace transfers="1" location="/tmp/filespace/daniel" name="daniel">
    <quota bytes="1048576"/>
    <writers>
```

```
<authorized>
  <agent-user>daniel</agent-user>
  <agent-user>SYS.ADMIN.*</agent-user>
</authorized>
<unauthorized>
  <agent-user>dave</agent-user>
</unauthorized>
</writers>
</filespace>
</filespaces>
```

JSON

以下示例显示了简单文件空间信息 JSON 响应的格式。

```
{
  "filespaces":{
    "filespace":{
      "transfers":"1",
      "location":"/tmp/filespace/daniel",
      "name":"daniel",
      "writers":{
        "authorized":{
          "agent-user":"daniel",
          "agent-user":"SYS.ADMIN.*"
        },
        "unauthorized":{
          "agent-user":"dave"
        }
      }
    },
    "quota":{
      "bytes":"1048576"
    }
  }
}
```

理解文件空间信息响应

以下列表描述了文件空间信息响应的元素和属性：

filespaces

包含一个或多个 <filespace> 元素的组。

文件空间

包含文件空间信息的组。

属性	描述
transfers	正在对文件空间进行的传输数。
位置	文件空间的文件系统的位置。
名	文件空间的名称。
integrity-check-result	对文件空间进行完整性检查的结果。 有效值包括： • 确定 • MISSING_FILESYSTEM • MISSING_DATABASEENTRY

quota

用于描述文件空间可使用的文件系统空间量的元素。

属性	描述
字节	文件空间可使用的最大字节数。

writers

包含有权访问或无权访问文件空间的用户的相关信息的一组。

authorized

包含有权访问文件空间的用户列表的一组。

未获授权

包含无权访问文件空间的用户列表的一组。如果在授权和未授权列表中都显示用户名或用户通配符匹配，那么用户无权访问文件空间。

agent-user

包含已授权或未授权的用户名的元素。此用户名可以包含通配符以匹配多个用户。

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 312 页的『使用 IBM MQ Managed File Transfer 服务 Web 网关』](#)

通过创建提交到 Web 网关的 HTTP 请求，可上载文件、查询文件空间中的文件、查看文件传输的状态以及从文件空间删除文件。

[第 313 页的『示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本请求和来自 Web 网关的相应 HTTP 响应。

相关任务

[第 333 页的『示例：列出所有文件空间』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交 HTTP 请求，可以列出所有文件空间。Web 网关将返回 XML 或 JSON 格式的响应，其中列出文件空间的名称、每个文件空间的配额以及有权和无权向每个文件空间写入内容的用户。

相关参考

[第 919 页的『响应格式：XML 和 JSON』](#)

IBM MQ Managed File Transfer Web 网关采用以下两种格式之一返回查询的响应：XML 或 JavaScript Object Notation (JSON)。

[第 920 页的『传输查询响应格式』](#)

当您从 IBM MQ Managed File Transfer Web 网关请求一个或多个传输的状态时会以 JSON 或 XML 格式返回响应。

[第 925 页的『文件空间查询的响应格式』](#)

从 IBM MQ Managed File Transfer 服务 Web 网关请求文件空间中的部分或全部文件的列表时，将根据您使用 Accept: 头指定的内容以 JSON 或 XML 格式返回响应。

[第 908 页的『Web 网关 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

文件空间创建或更改请求格式

通过在 HTTP 请求中包含 XML 格式的内容，可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 FileSpaceInfo.xsd，此模式位于 WMQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

以下示例显示了用于创建文件空间的 XML 请求的格式。

```
<filespaces>
```

```
<filesystem>
  <quota bytes="1048576"/>
  <writers>
    <authorized>
      <agent-user>SYS.ADMIN.*</agent-user>
    </authorized>
    <unauthorized>
      <agent-user>dave</agent-user>
    </unauthorized>
  </writers>
</filesystem>
</filesystems>
```

以下示例显示了用于修改现有文件空间配置的 XML 请求的格式。必须使用 `action=add`、`action=remove` 和 `action=overwrite` 属性来更改已授权和未授权的编写者列表。

```
<filesystems>
  <filesystem>
    <quota bytes="2097152"/>
    <writers>
      <authorized action="add">
        <agent-user>emily</agent-user>
      </authorized>
      <unauthorized action="remove">
        <agent-user>dave</agent-user>
      </unauthorized>
    </writers>
  </filesystem>
</filesystems>
```

了解文件空间创建或修改请求

以下列表描述了该请求的元素和属性：

filesystems

包含一个 <filesystem> 元素的元素。

文件空间

包含文件空间信息的组元素。

quota

用于描述文件空间可使用的文件系统空间量的元素。 如果用户提交的文件传输请求会导致文件空间超出其配额，那么该传输将失败，并且将生成一条错误。

属性	描述
字节	文件空间可使用的最大字节数。

writers

包含有权访问或无权访问文件空间的用户的相关信息的一组。

authorized

包含有权访问文件空间的用户列表的组。

属性	描述
操作	要对子元素中指定的代理用户名执行的操作。有效选项是： • add - 向授权用户列表添加新的代理用户名 • remove - 从授权用户列表中移除代理用户名 • overwrite - 用提供的列表替换所有现有的授权用户列表。

未获授权

包含无权访问文件空间的用户列表的组。 如果授权用户列表和未授权用户列表中都包含某个用户，那么该用户无权访问文件空间。

属性	描述
操作	要对子元素中指定的代理用户名执行的操作。有效选项是： • add - 向未授权用户列表添加新的代理用户名 • remove - 从未授权用户列表中移除代理用户名 • overwrite - 用提供的列表替换所有现有的未授权用户列表。

agent-user

包含已授权或未授权的用户名的元素。此用户名可以包含通配符以匹配多个用户。

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

[第 327 页的『管理 IBM MQ Managed File Transfer 服务 Web 网关』](#)

您可以创建和删除文件空间，控制对个别文件空间具有访问权的用户。

[第 329 页的『用于管理的示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关任务

[第 329 页的『示例：创建文件空间』](#)

必须先为用户创建文件空间，才能将文件传输到用户文件空间。您可以使用 IBM MQ Managed File Transfer Web 网关创建文件空间。

相关参考

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

[第 931 页的『用于管理 Web 网关的 HTTP 头』](#)

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

[第 932 页的『用于管理 Web 网关的统一资源标识语法』](#)

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 IBM MQ URI 区分开。建议使用上下文根 /wmqfte。用于管理任务的 URI 通过术语 /admin 与现有 IBM MQ Managed File Transfer URI 区分开。

[第 934 页的『用于管理 Web 网关的内容类型』](#)

您提交到 IBM MQ Managed File Transfer Web 网关管理 API 的 HTTP 请求必须具有介质类型 application/xml。来自 Web 网关的响应也具有介质类型 application/xml。

[第 935 页的『来自 Web 网关管理 API 的 HTTP 响应代码』](#)

向 IBM MQ Managed File Transfer Web 网关管理 API 发出的请求的 HTTP 响应中会返回状态码。

[第 942 页的『用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交请求，可以创建 Web 用户标识与 IBM MQ 消息描述符 (MQMD) 用户标识之间的映射集。此 HTTP 请求必须包含以下 XML 格式的内容。

[第 943 页的『文件空间管理日志记录格式』](#)

创建、更改或删除文件空间时，对文件空间的更改将记录在托管 Web 网关的应用程序服务器的事件日志中。这使得管理员能够查看对文件空间所做的更改。

[第 908 页的『Web 网关 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式

通过向 IBM MQ Managed File Transfer Web 网关提交请求，可以创建 Web 用户标识与 IBM MQ 消息描述符 (MQMD) 用户标识之间的映射集。此 HTTP 请求必须包含以下 XML 格式的内容。

以下示例显示了用于创建映射集的 XML 请求的格式。要修改现有的映射集，请使用相同的格式。

```
<users>
  <user>
    <userID>mike</userID>
    <mqmdUserID>mqmike</mqmdUserID>
  </user>
  <user>
    <userID>lisa</userID>
    <mqmdUserID>qmlisa</mqmdUserID>
  </user>
</users>
```

如果尝试使用未映射到 MQMD 用户标识的 Web 用户标识来启动文件上载，那么将使用 defaultMQMDUserID 初始化参数的值。在将 Web 网关应用程序部署到应用程序服务器环境时，会设置该参数的值。有关更多信息，请参阅第 198 页的『在 WebSphere 应用程序服务器 V7.0 中部署 Web 网关』和第 183 页的『准备使用 WebSphere 应用程序服务器 Community Edition 部署 Web 网关』。

了解用于创建或更改用户标识映射的请求

以下列表描述了该请求的元素和属性：

users

包含 <user> 元素的组元素。

用户

包含 Web 网关用户信息的元素。

userID

包含用户的 Web 用户标识的元素。这是在用于托管 Web 网关的应用程序服务器环境中定义的用户标识。

mqmdUserID

包含 MQMD 用户标识（消息描述符中提供的 IBM MQ 用户标识）的名称的元素，在用户启动的文件上载传输中将使用该用户标识。

mqmdUserID 属性的最大长度为 12 个字符。

相关概念

第 305 页的『Web 网关的方案』

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理，并使用 HTTP 客户机检索传输的状态。

第 327 页的『管理 IBM MQ Managed File Transfer 服务 Web 网关』

您可以创建和删除文件空间，控制对个别文件空间具有访问权的用户。

第 329 页的『用于管理的示例 HTTP 流』

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关任务

第 338 页的『示例：将 Web 用户标识映射到 MQMD 用户标识』

将文件上载提交到 IBM MQ Managed File Transfer Web 网关时，Web 网关会确定传输要使用的 IBM MQ 消息描述符 (MQMD) 用户标识。您可以使用 Web 网关来定义 Web 用户标识和 MQMD 用户标识之间的映射集。

相关参考

第 930 页的『Web 网关管理 API 引用』

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

第 931 页的『用于管理 Web 网关的 HTTP 头』

使用 HTTP 头，可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关，未定义任何用于管理 API 的头。

第 932 页的『用于管理 Web 网关的统一资源标识语法』

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 IBM MQ URI 区分开。建议使用上下文根 /wmqfte。用于管理任务的 URI 通过术语 /admin 与现有 IBM MQ Managed File Transfer URI 区分开。

第 934 页的『用于管理 Web 网关的内容类型』

您提交到 IBM MQ Managed File Transfer Web 网关管理 API 的 HTTP 请求必须具有介质类型 application/xml。来自 Web 网关的响应也具有介质类型 application/xml。

第 935 页的『来自 Web 网关管理 API 的 HTTP 响应代码』

向 IBM MQ Managed File Transfer Web 网关管理 API 发出的请求的 HTTP 响应中会返回状态码。

第 939 页的『文件空间创建或更改请求格式』

通过在 HTTP 请求中包含 XML 格式的内容，可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 FileSpaceInfo.xsd，此模式位于 WMQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

第 943 页的『文件空间管理日志记录格式』

创建、更改或删除文件空间时，对文件空间的更改将记录在托管 Web 网关的应用程序服务器的事件日志中。这使得管理员能够查看对文件空间所做的更改。

第 908 页的『Web 网关 API 引用』

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API)，以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

文件空间管理日志记录格式

创建、更改或删除文件空间时，对文件空间的更改将记录在托管 Web 网关的应用程序服务器的事件日志中。这使得管理员能够查看对文件空间所做的更改。

日志格式

```
FTELOG: operation - status. Requested by user_ID at host_name.  
Information: information
```

操作

请求对文件空间执行的操作。操作值为：

- create file space
- modify file space
- delete file space

状态

请求的操作是否成功。状态值为：

- successful
- failed, 在此情况下，还会提供失败原因

用户标识

请求文件空间操作的用户的用户名。

主机名

用户从其中发出请求的系统的主机名。

参考

有关请求的信息。例如：

```
File space: fred, quota: 123456 bytes, added authorized writers: [tom dick harry],
added unauthorized writers: [tarzan jane], removed unauthorized writers: [bob]
```

会将这些日志消息写入应用程序服务器的事件日志。可以在以下目录中找到这些文件：

- 针对 WebSphere Application Server V7.0, *WAS7_install_location/profiles/profile_name/logs/server_name*
- 针对 WebSphere Application Server Community Edition, *WASCE_install_location/var/log*

相关概念

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理, 并使用 HTTP 客户机检索传输的状态。

[第 327 页的『管理 IBM MQ Managed File Transfer 服务 Web 网关』](#)

您可以创建和删除文件空间, 控制对个别文件空间具有访问权的用户。

[第 329 页的『用于管理的示例 HTTP 流』](#)

您可以构造 HTTP 请求并将其提交给 IBM MQ Managed File Transfer Web 网关。这些示例显示了样本管理请求和来自 Web 网关的相应 HTTP 响应。

相关参考

[第 930 页的『Web 网关管理 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义用于管理文件传输工件的 RESTful HTTP 应用程序编程接口 (API)。

[第 931 页的『用于管理 Web 网关的 HTTP 头』](#)

使用 HTTP 头, 可定制创建或检索资源的请求。对于 IBM MQ Managed File Transfer Web 网关, 未定义任何用于管理 API 的头。

[第 932 页的『用于管理 Web 网关的统一资源标识语法』](#)

部署时指定的上下文根会将 IBM MQ Managed File Transfer 统一资源标识 (URI) 与其他 IBM MQ URI 区分开。建议使用上下文根 /wmqfte。用于管理任务的 URI 通过术语 /admin 与现有 IBM MQ Managed File Transfer URI 区分开。

[第 934 页的『用于管理 Web 网关的内容类型』](#)

您提交到 IBM MQ Managed File Transfer Web 网关管理 API 的 HTTP 请求必须具有介质类型 application/xml。来自 Web 网关的响应也具有介质类型 application/xml。

[第 935 页的『来自 Web 网关管理 API 的 HTTP 响应代码』](#)

向 IBM MQ Managed File Transfer Web 网关管理 API 发出的请求的 HTTP 响应中会返回状态码。

[第 939 页的『文件空间创建或更改请求格式』](#)

通过在 HTTP 请求中包含 XML 格式的内容, 可以请求通过 IBM MQ Managed File Transfer Web 网关来创建或更改文件空间。该 XML 格式符合模式 FileSpaceInfo.xsd, 此模式位于 WMQMFT 安装的 *MQ_INSTALLATION_PATH/mqft/samples/schema* 目录中。

[第 942 页的『用于将 Web 用户标识映射到 MQMD 用户标识的 XML 格式』](#)

通过向 IBM MQ Managed File Transfer Web 网关提交请求, 可以创建 Web 用户标识与 IBM MQ 消息描述符 (MQMD) 用户标识之间的映射集。此 HTTP 请求必须包含以下 XML 格式的内容。

[第 908 页的『Web 网关 API 引用』](#)

IBM MQ Managed File Transfer Web 网关定义 RESTful HTTP 应用程序编程接口 (API), 以使用 HTTP 请求和响应创建传输、从文件空间中下载文件和查看已提交传输的状态。

fteCreateWebAgent (创建 ***IBM MQ Managed File Transfer Web*** 代理)

fteCreateWebAgent 命令创建代理及其关联的配置, 以用于 Web 网关。IBM MQ Managed File Transfer 服务器随附该命令。

用途

使用 **fteCreateWebAgent** 命令以创建 Web 代理。该命令为您提供 MQSC 命令，必须在代理使用的队列管理器上运行这些 MQSC 命令才能创建以下代理队列：

- SYSTEM.FTE.AUTHADM1.agent_name
- SYSTEM.FTE.AUTHAGT1.agent_name
- SYSTEM.FTE.AUTHMON1.agent_name
- SYSTEM.FTE.AUTHOPS1.agent_name
- SYSTEM.FTE.AUTHSCH1.agent_name
- SYSTEM.FTE.AUTHTRN1.agent_name
- SYSTEM.FTE.COMMAND.agent_name
- SYSTEM.FTE.DATA.agent_name
- SYSTEM.FTE.EVENT.agent_name
- SYSTEM.FTE.REPLY.agent_name
- SYSTEM.FTE.STATE.agent_name

由于代理用于 Web 网关，因此除了先前列表外，还将创建两个队列：

- SYSTEM.FTE.WEB.RESP.agent_name
- SYSTEM.FTE.WEB.gateway_name

这些队列是内部系统队列，除非您要删除该代理，否则不得修改、删除或读取这些队列中的消息。以下位置的文件中也提供了要运行的 MQSC 命令：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_create.mqsc`。

如果稍后要删除代理，该命令还为您提供 MQSC 命令，必须运行这些 MQSC 命令才能清除然后删除属于代理的队列。MQSC 命令位于以下位置的文件中：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_delete.mqsc`。

IBM MQ Managed File Transfer 提供帮助您配置代理的高级代理属性。[第 591 页的『agent.properties 文件』](#)中描述了这些属性。

注：运行 Web 代理所使用的用户必须与运行应用程序服务器所使用的用户相同或在同一个组中。

Web 代理的限制

- Web 代理只能是通过 Web 网关启动的传输的源代理。如果尝试通过另一种方法，将 Web 代理用作源代理来执行传输，那么传输将失败，返回码为 68 (TRANSFER_NOT_SUPPORTED)。
- 在将目标指定为文件空间时，Web 代理只能是传输的目标代理。如果尝试将 Web 代理用作目标代理（但作为不同目标类型）来执行传输，那么该传输将失败，并返回以下错误消息：BFGCH0103: 传输请求指定 Web 网关代理“web_agent_name”作为目标代理。Web 网关代理只能是到文件空间的传输的目标。
- Web 代理不能监视资源。如果尝试为 Web 代理创建资源监视器，那么该命令将失败，返回码为 113 (MONITOR_NOT_SUPPORTED)。

语法

fteCreateWebAgent

► fteCreateWebAgent — -agentName — ? — *agent_name* — -agentQMGr — ? — ►

► — *agent_qmgr_name* — -webGatewayName — ? — *gateway_name* — ►

► — -agentQMGrHost — ? — *agent_qmgr_host* — ►

► — -agentQMGrPort — ? — *agent_qmgr_port* — ►

► — -agentQMGrChannel — ? — *agent_qmgr_channel* — ►

► — -agentDesc — ? — *agent_description* — ► — -ac — ► — -authorityChecking — ►

► — -p — *configuration_options* — ► — -f — ►

► — -s — ? — *service_name* — ► — -su — ? — *user_name* — ► — -sp — ? — *password* — ► — -sj — ? — *options* — ► — -sl — ? — *options* — ► — -n — ►

参数

-agentName *agent_name*

必需。要创建的代理的名称。代理名称对其协调队列管理器必须唯一。

有关命名代理的更多信息，请参阅[对象命名约定](#)。

-agentQMGr *agent_qmgr_name*

必需。代理队列管理器的名称。

-webGatewayName *gateway_name*

必需。代理是其一个组件的 Web 网关的名称。

有关命名 Web 网关的更多信息，请参阅[对象命名约定](#)。

-agentQMGrHost *agent_qmgr_host*

可选。代理队列管理器的主机名或 IP 地址。如果未指定该参数，那么将采用绑定方式连接。

-agentQMGrPort *agent_qmgr_port*

可选。用于客户机与代理队列管理器的连接的端口号。只有当同时指定了 **agentQMGrHost** 参数时，才能使用该参数。如果未指定 **agentQMGrPort** 参数，那么将使用缺省端口 1414。

-agentQMGrChannel *agent_qmgr_channel*

可选。只有当同时指定了 **agentQMGrHost** 参数时，才能使用该参数。如果未指定 **agentQMGrChannel** 参数，那么将使用缺省通道 SYSTEM.DEF.SVRCONN。

-agentDesc *agent_description*

可选。在 IBM MQ Explorer 中显示的代理的描述。

-ac 或 -authorityChecking

可选。此参数允许进行权限检查。如果指定该参数，那么代理将检查提交请求的用户是否有权执行请求的操作。

-p *configuration_options*

可选。用于创建代理的配置选项集的名称。按照约定，这是协调队列管理器的名称。如果未指定该参数，那么会使用缺省的配置选项集。

-f

可选。强制该命令覆盖现有配置。

-s service_name

可选（仅限 Windows）。指示代理程序将作为 Windows 服务运行。如果未指定服务名称，那么服务将命名为 mqmftAgent<AGENT><QMGR>，其中 <AGENT> 是代理名称，<QMGR> 是代理队列管理器名称。

服务的显示名称（显示在 窗口 **服务** 窗口的 **名称** 列中）始终为 **IBM MQ Managed File Transfer 代理程序 <AGENT>@<QMGR>**。

-su user_name

可选（仅限 Windows）。当代理程序作为 Windows 服务运行时，该参数指定运行服务所使用的帐户名。要使用 Windows 域用户帐户来运行代理，请以 DomainName\UserName 格式指定该值。要使用本地内置域中的帐户来运行该服务，请以 UserName 格式指定该值。

使用 **-su** 参数指定的 Windows 用户帐户必须具有 **Log on as a service** 权限。有关如何授予该权限的信息，请参阅第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』。

在指定 **-s** 时为必需项。等同于 **-serviceUser**。

-sp password

可选（仅限 Windows）。由 **-su** 或 **-serviceUser** 参数设置的用户帐户的密码。

该参数仅在指定 **-s** 时才有效。等同于 **-servicePassword**。如果在指定 **-s** 参数时不指定该参数，将生成警告消息。该消息警告您必须先使用 Windows 服务工具设置密码，这样服务才能成功启动。

-sj options

可选（仅限 Windows）。当代理作为 Windows 服务启动时，以 -D 或 -X 格式来定义将传递至 JVM 的选项列表。这些选项使用井号 (#) 或分号 (;) 字符进行分隔。如果需要嵌入任何 # 或 ; 字符，请使用单引号将其引起来。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceJVMOptions**。

-sl options

可选（仅限 Windows）。设置 Windows 服务日志级别。有效选项包括：error、info、warn 和 debug。缺省值为 info。如果 Windows 服务出现问题，那么该选项可能很有用。如果将其设置为 debug，那么会在服务日志文件中提供更详细的信息。

该参数仅在指定 **-s** 时才有效。等同于 **-serviceLogLevel**。

-n

可选（仅限 Windows）。指示代理将作为正常进程运行。这与 **-s** 选项互斥。如果既未指定 **-s** 选项，也未指定 **-n** 选项，那么会将代理程序配置为正常的 Windows 进程。

等同于 **-normal**。

-? or -h

可选。显示命令语法。

示例

在本示例中，使用代理队列管理器 QM_NEPTUNE 和 Web 网关 GATEWAY_ONE 来创建代理 WEBAGENT1。该代理使用缺省的协调队列管理器：

```
fteCreateWebAgent -agentName WEBAGENT1 -webGatewayName GATEWAY_ONE -agentQMgr QM_NEPTUNE  
-agentQMgrHost myhost.ibm.com -agentQMgrPort 1415 -agentQMgrChannel CHANNEL1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

[第 304 页的『IBM MQ Managed File Transfer Web 网关』](#)

Web 网关提供可用于与 IBM MQ Managed File Transfer 网络交互的 RESTful API。

[第 305 页的『Web 网关的方案』](#)

使用 IBM MQ Managed File Transfer Web 网关可通过 HTTP 客户机将文件传输到 IBM MQ Managed File Transfer 代理并检索传输的状态。

[第 307 页的『Web 网关如何适合您的 IBM MQ Managed File Transfer 拓扑』](#)

使用 IBM MQ Managed File Transfer 服务 Web 网关将文件传输到 IBM MQ Managed File Transfer (MQMFT) 代理, 并使用 HTTP 客户机检索传输的状态。

[第 397 页的『有关作为 Windows 服务运行代理或记录器的指南』](#)

您可以作为 Windows 服务来运行 IBM MQ Managed File Transfer 代理、独立数据库记录器和独立文件记录器。如果这些 Windows 服务出现任何问题, 您可以使用服务日志文件以及本主题中的信息来诊断问题。

相关任务

[第 182 页的『准备部署 Web 网关』](#)

在部署 IBM MQ Managed File Transfer Web 网关之前, 必须设置应用程序服务器环境和相关模块。本部分描述了 IBM MQ 和两个不同应用程序服务器的设置任务。

[第 197 页的『部署 IBM MQ Managed File Transfer Web 网关』](#)

必须将 IBM MQ Managed File Transfer Web 网关部署到与 Java Platform, Enterprise Edition 5 兼容的应用程序服务器。不同应用程序服务器的部署过程各不相同。本部分概述了两个应用程序服务器的部署过程。

[第 216 页的『将代理作为 Windows 服务启动』](#)

您可以将代理作为 Windows 服务启动, 以便当您从 Windows 注销后, 代理可以继续运行, 并且可以接收文件传输。

相关参考

[第 421 页的『Web 代理无法启动』](#)

如果您收到来自 **fteStartAgent** 命令的错误, 并且您正在尝试启动 Web 代理, 请检查 `SYSTEM.FTE.WEB.gateway_name` 队列是否存在。

Web 网关所使用的数据库表

IBM MQ Managed File Transfer Web 网关使用以下数据库表配置用户文件空间并保证用户文件空间的安全。

以下数据库表由 Web 网关使用: 请勿删除或修改这些表或其中包含的任何数据。

- FILE_SPACE
- FILE_SPACE_ENTRY
- PERMISSIONS
- USER_MQMD_MAPPING
- WEBGATEWAY_CONFIG

Web 网关还会使用数据库记录器表中的审计信息为用户提供传输信息。有关更多信息, 请参阅 [第 731 页的『记录器所使用的数据库表』](#)。

Web 网关使用的数据库表可能位于与数据库记录器所使用的表相同的数据库中, 只要两个表集具有不同的模式名称。

相关任务

[第 181 页的『设置数据库以用于文件空间』](#)

在可以使用文件空间之前, 必须为 Web 网关设置数据库表以存储文件空间信息。您可以在现有的日志数据库中创建这些表, 或创建新数据库以包含这些表。

将 Apache Ant 与 WebSphere MQ 受管文件传输一起使用

fteAnt (在 *IBM MQ Managed File Transfer* 环境中运行 *Ant* 任务)

fteAnt 命令会在具有可用的 IBM MQ Managed File Transfer Ant 任务的环境中运行 Ant 脚本。

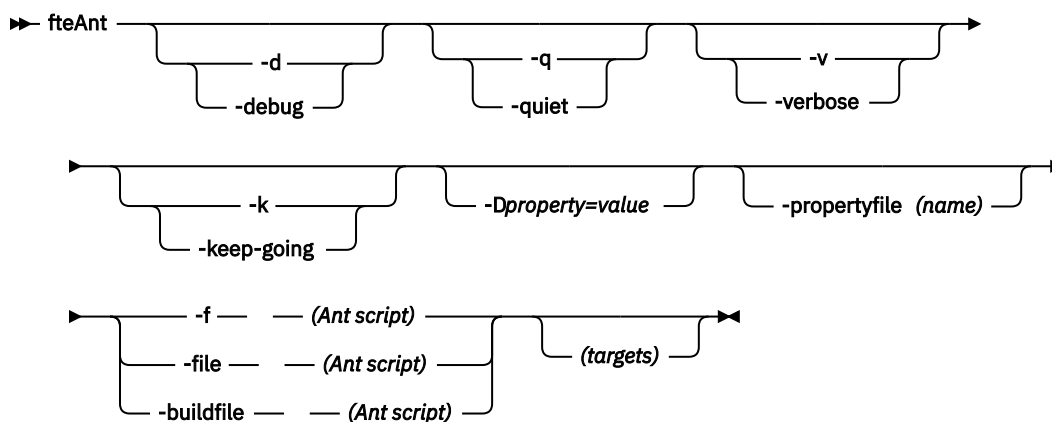
用途

使用 **fteAnt** 命令在具有 IBM MQ Managed File Transfer 的环境中运行 Ant 脚本。不同于标准 **ant** 命令，**fteAnt** 要求您定义脚本文件。

不能直接在 IBM 4690 系统上运行 **fteAnt** 命令。但可通过 Ant 脚本引用 IBM 4690 系统。有关在 IBM 4690 环境中使用 IBM MQ Managed File Transfer 的更多信息，请参阅第 38 页的『在零售环境中使用 IBM MQ Managed File Transfer』

语法

fteAnt



参数

-debug 或 **-d**

可选。生成调试输出。

-quiet 或 **-q**

可选。生成最小输出。

-verbose 或 **-v**

可选。生成详细输出。

-keep-going 或 **-k**

可选。执行不依赖于故障目标的所有目标。

-D property=value

可选。将 *value* 用于给定的 *property*。使用 **-D** 设置的属性的优先顺序高于属性文件中设置的属性。

使用 **com.ibm.wmqfte.propertyset** 属性来指定用于 Ant 任务的配置选项集。使用非缺省协调队列管理器的名称作为该属性的值。随后，Ant 任务会使用与该非缺省协调队列管理器关联的配置选项集。如果您未指定该属性，那么会使用基于缺省协调队列管理器的缺省配置选项集。如果您针对 Ant 任务指定 **cmdqm** 属性，那么该属性的优先顺序高于针对 **fteAnt** 命令指定的配置选项集。无论您使用缺省配置选项集还是使用 **com.ibm.wmqfte.propertyset** 属性指定设置，此行为均适用。

-propertyfile (name)

可选。从带有 **-D** 属性（高优先顺序）的文件装入所有属性。

-f (Ant 脚本)、-file (Ant 脚本) 或 -buildfile (Ant 脚本)

必需。指定要运行的 Ant 脚本的名称。

targets

可选。要从 Ant 脚本运行的一个或多个目标的名称。如果您不针对该参数指定值，那么会运行脚本的缺省目标。

-version

可选。显示 IBM MQ Managed File Transfer 命令和 Ant 版本。

-? or -h

可选。显示命令语法。

示例

在此示例中，会运行 Ant 脚本 `fte_script.xml` 中的目标 **copy**，该命令会将调试输出写入标准输出。

```
fteAnt -d -f fte_script.xml copy
```

返回码

0

命令成功完成。

1

命令结束但未成功。

其他状态返回码也可从 Ant 脚本指定，例如，通过使用 Ant fail 任务。

请参阅 [失败](#) 以获取更多信息。

相关概念

第 354 页的『入门：将 Ant 脚本用于 IBM MQ Managed File Transfer』

将 Ant 脚本用于 IBM MQ Managed File Transfer 能够使您通过已解释的脚本语言协调复杂的文件传输操作。

相关参考

第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

第 355 页的『Ant 任务样本』

随 IBM MQ Managed File Transfer 安装一起提供了一些样本 Ant 脚本。这些样本位于目录 `MQ_INSTALLATION_PATH/mqft/samples/fteant` 中。每个样本脚本都包含 `init` 目标，请编辑 `init` 目标中设置的属性，以使用您的配置运行这些脚本。

IBM MQ Managed File Transfer 提供的 Ant 任务

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

任务

- 第 952 页的『[fte:awaitoutcome](#)』
- [fte:call](#)
- [fte:cancel](#)
- [fte:filecopy](#)
- [fte:filemove](#)
- [fte:ignoreoutcome](#)

- [fte:ping](#)
- [fte:uuid](#)

嵌套参数

以下嵌套参数描述嵌套元素集，这些元素在提供的多个 Ant 任务间共用：

- [fte:filespec](#)
- [fte:metadata](#)
- [用于程序调用的参数](#)

相关概念

第 354 页的『入门：将 Ant 脚本用于 IBM MQ Managed File Transfer』

将 Ant 脚本用于 IBM MQ Managed File Transfer 能够使您通过已解释的脚本语言协调复杂的文件传输操作。

相关参考

第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

第 450 页的『fteAnt (在 IBM MQ Managed File Transfer 环境中运行 Ant 任务)』

fteAnt 命令会在具有可用的 IBM MQ Managed File Transfer Ant 任务的环境中运行 Ant 脚本。

第 355 页的『Ant 任务样本』

随 IBM MQ Managed File Transfer 安装一起提供了一些样本 Ant 脚本。这些样本位于目录 `MQ_INSTALLATION_PATH/mqft/samples/fteant` 中。每个样本脚本都包含 `init` 目标，请编辑 `init` 目标中设置的属性，以使用您的配置运行这些脚本。

第 952 页的『fte:awaitoutcome』

等待 **fte:filecopy**、**fte:filemove** 或 **fte:call** 操作完成。

第 952 页的『fte:call』

您可以使用 **fte:call** 任务来远程调用脚本和程序。

第 954 页的『fte:cancel』

取消 IBM MQ Managed File Transfer 受管传输或受管调用。受管传输可能是使用 **fte:filecopy** 或 **fte:filemove** 任务创建的。受管调用可能是使用 **fte:call** 任务创建的。

第 955 页的『fte:filecopy』

fte:filecopy 任务在 IBM MQ Managed File Transfer 代理程序之间复制文件。该文件不会从源代理删除。

第 958 页的『fte:filemove』

fte:filemove 任务在 IBM MQ Managed File Transfer 代理程序之间移动文件。当文件成功从源代理传输到目标代理时，会从源代理删除该文件。

第 960 页的『fte:ignoreoutcome』

忽略 **fte:filecopy**、**fte:filemove** 或 **fte:call** 命令的结果。指定 **fte:filecopy**、**fte:filemove** 或 **fte:call** 任务以获取 `defer` 的结果时，Ant 任务会分配资源以跟踪此结果。如果您对结果不再感兴趣，可以使用 **fte:ignoreoutcome** 任务来清除这些资源。

第 961 页的『fte:ping』

Ping 代理以产生响应，以此确定代理是否能够处理传输。

第 961 页的『fte:uuid』

生成伪随机唯一标识，并将其分配到给定属性。例如，您可以使用该标识为其他文件传输操作生成作业名。

第 962 页的『fte:filespec』

fte:filespec 参数用作为其他任务中的嵌套元素。使用 **fte:filespec** 来描述一个或多个源文件，目录或数据集与目标之间的映射。通常在表述要移动或复制的一组文件、数据集或目录时使用该元素。

第 967 页的『fte:metadata』

元数据用于通过文件传输操作传送更多用户定义的信息。

第 968 页的『程序调用嵌套元素』

可使用五个嵌套元素中的一个来启动程序：`fte:presrc`、`fte:predst`、`fte:postdst`、`fte:postsrc` 和 `fte:command`。这些嵌套元素指示代理作为其处理过程的一部分来调用外部程序。要能够启动程序，必须确保命令位于运行该命令的代理的 `agent.properties` 文件中 `commandPath` 属性所指定的位置中。

fte:awaitoutcome

等待 **fte:filecopy**、**fte:filemove** 或 **fte:call** 操作完成。

属性

id

必需。标识要等待其结果的传输。通常，这是由 `fte:filecopy`、`fte:filemove` 或 `fte:call` 任务的 `idProperty` 属性 (attribute) 设置的特性 (property)。

rcproperty

必需。指定用于存储 **fte:awaitoutcome** 任务的返回码的属性。

timeout

可选。等待操作完成的最大时间（秒）。最小超时为一秒。如果未指定超时值，**fte:awaitoutcome** 任务将一直等待直至确定操作的结果。

示例

在本示例中，文件复制将启动，其标识将存储在 `copy.id` 属性中。在复制期间，其他处理可同时进行。**fte:awaitoutcome** 语句用于等待，直到复制操作完成。**fte:awaitoutcome** 语句标识等待使用存储在 `copy.id` 属性中的标识的操作。**fte:awaitoutcome** 将用于指示复制操作结果的返回码存储在称为 `copy.result` 的属性内。

```
<-- issue a file copy request -->
<fte:filecopy
  src="AGENT1@QM1"
  dst="AGENT2@QM2"
  idproperty="copy.id"
  outcome="defer">

  <fte:filespec
    srcfilespec="/home/fteuser1/file.bin"
    dstdir="/home/fteuser2"/>

  </fte:filecopy>

  <fte:awaitoutcome id="${copy.id}" rcProperty="copy.rc"/>

  <echo>Copy id=${copy.id} rc=${copy.rc}</echo>
```

相关参考

第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:call

您可以使用 **fte:call** 任务来远程调用脚本和程序。

该任务允许您向代理发送 **fte:call** 请求。代理通过运行脚本或程序并返回结果来处理该请求。要调用的命令必须可供代理访问。确保 `agent.properties` 文件中的 `commandPath` 属性值包括要调用的命令的位置。由命令嵌套元素指定的任何路径信息必须相对于 `commandPath` 属性指定的位置。缺省情况下，`commandPath` 为空，以使代理无法调用任何命令。有关该属性的更多信息，请参阅[使用 commandPath](#)。

有关 `agent.properties` 文件的更多信息，请参阅第 591 页的『[agent.properties 文件](#)』。

属性

agent

必需。指定要将 **fte:call** 请求提交到的代理。按以下格式指定代理信息: *agentname@qmgrname*, 其中 *agentname* 是代理的名称, *qmgrname* 是该代理直接连接到的队列管理器的名称。

cmdqm

可选。请求提交到的命令队列管理器。按以下格式指定此信息: *qmgrname@host@port@channel*, 其中:

- *qmgrname* 是队列管理器的名称
- *host* 是正在运行队列管理器的系统的可选主机名
- *port* 是队列管理器正在侦听的可选端口号
- *channel* 是要使用的可选 SVRCONN 通道

如果省略命令队列管理器的 *host*、*port* 或 *channel* 信息, 那么会使用 `command.properties` 文件中指定的连接信息。有关更多信息, 请参阅第 588 页的『`command.properties` 文件』。

您可以使用 **com.ibm.wmqfte.propertySet** 属性指定要使用的 `command.properties` 文件。有关更多信息, 请参阅 `com.ibm.wmqfte.propertySet`。

如果您不使用 `cmdqm` 属性, 那么该任务会缺省使用

`com.ibm.wmqfte.ant.commandQueueManager` 属性 (如果已设置该属性)。如果未设置 `com.ibm.wmqfte.ant.commandQueueManager` 属性, 会尝试连接至 `command.properties` 文件中定义的缺省队列管理器。 `com.ibm.wmqfte.ant.commandQueueManager` 属性的格式与 `cmdqm` 属性相同, 即 *qmgrname@host@port@channel*。

idproperty

除非您已指定 *outcome* 为 **defer**, 否则可选。指定传输标识要分配到的属性的名称。在提交传输请求时会生成传输标识, 您可以使用传输标识来跟踪传输进展、诊断传输问题以及取消传输。

如果您还指定了 *ignore* 的 *outcome* 属性, 那么不能指定该属性。但是, 如果您已将 *outcome* 属性指定为 **defer**, 那么还必须指定 *idproperty*。

jobname

可选。向 **fte:call** 请求指定作业名。您可以使用作业名来创建传输的逻辑组。使用第 961 页的『`fte:uuid`』任务来生成伪唯一的作业名。如果您不使用 *jobname* 属性, 那么该任务缺省使用 `com.ibm.wmqfte.ant.jobName` 属性值 (如果已设置该属性)。如果未设置该属性, 那么将不存在与 **fte:call** 请求关联的作业名。

origuser

可选。指定要与 **fte:call** 请求相关联的始发用户标识。如果您不使用 *origuser* 属性, 那么该任务会缺省使用用于运行 Ant 脚本的用户标识。

outcome

可选。确定任务在将控制权返回给 Ant 脚本之前是否等待 **fte:call** 操作完成。指定以下某个选项:

await

任务在返回之前等待 **fte:call** 操作完成。如果将 *outcome* 指定为 **await**, 那么 *idproperty* 属性为可选。

defer

任务在提交 **fte:call** 请求之后立即返回, 并且假定调用操作的结果在稍后使用 `awaitoutcome` 或 `ignoreoutcome` 任务处理。如果将 *outcome* 指定为 **defer**, 那么 *idproperty* 属性是必需的。

忽略

如果 **fte:call** 操作的结果不重要, 那么可以指定值 **ignore**。随后, 任务会在 **fte:call** 请求提交之后立即返回, 而不会分配任何资源来跟踪命令的结果。如果将 *outcome* 指定为 **ignore**, 那么不能指定 *idproperty* 属性。

如果您不指定 *outcome* 属性, 那么该任务会缺省使用值 **await**。

rcproperty

可选。指定要将 **fte:call** 请求的结果代码指定给的属性的名称。结果代码反映 **fte:call** 请求的总体结果。

如果您已将 outcome 属性指定为 ignore 或 defer, 那么无法指定该属性。然而, 如果将 outcome 指定为 await, 那么必须指定 rcproperty。

指定为嵌套元素的参数

fte:command

指定将由代理调用的命令。您只能将单个 **fte:command** 元素与给定的 **fte:call** 操作关联。要调用的命令必须位于代理的 agent.properties 文件中的 commandPath 属性指定的路径上。

fte:metadata

您可以指定要与调用操作关联的元数据。该元数据记录在由调用操作生成的日志消息中。您只能将一个元数据块与一个给定传输元素相关联; 但是, 该块可包含多份元数据。

示例

该示例显示如何在队列管理器 QM1 上运行的 AGENT1 处调用命令。要调用的命令为脚本 command.sh, 该脚本是通过 xyz 的单个参数调用的。命令 command.sh 位于代理的 agent.properties 文件中的 commandPath 属性指定的路径中。

```
<fte:call cmdqm="QM0@localhost@1414@SYSTEM.DEF.SVRCONN"
  agent="AGENT1@QM1"
  rcproperty="call.rc"
  origuser="bob"
  jobname="{job.id}">

  <fte:command command="command.sh" successrc="1" retrycount="5" retrywait="30">
    <fte:arg value="xyz"/>
  </fte:command>

  <fte:metadata>
    <fte:entry name="org.foo.accountName" value="BDG3R"/>
  </fte:metadata>

</fte:call>
```

相关参考

第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』

IBM MQ Managed File Transfer 提供了一些任务, 您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:cancel

取消 IBM MQ Managed File Transfer 受管传输或受管调用。受管传输可能是使用 **fte:filecopy** 或 **fte:filemove** 任务创建的。受管调用可能是使用 **fte:call** 任务创建的。

属性

agent

必需。指定要将 **fte:cancel** 请求提交给的代理。值的格式如下: *agentname@qmgrname*, 其中 *agentname* 是代理的名称, *qmgrname* 是该代理直接连接到的队列管理器的名称。

cmdqm

可选。请求提交到的命令队列管理器。按以下格式指定此信息: *qmgrname@host@port@channel*, 其中:

- *qmgrname* 是队列管理器的名称
- *host* 是正在运行队列管理器的系统的可选主机名
- *port* 是队列管理器正在侦听的可选端口号

- *channel* 是要使用的可选 SVRCONN 通道

如果省略命令队列管理器的 *host*、*port* 或 *channel* 信息，那么会使用 `command.properties` 文件中指定的连接信息。有关更多信息，请参阅第 588 页的『`command.properties` 文件』。

您可以使用 **com.ibm.wmqfte.propertySet** 属性指定要使用的 `command.properties` 文件。有关更多信息，请参阅 `com.ibm.wmqfte.propertySet`。

如果您不使用 `cmdqm` 属性，那么该任务会缺省使用 `com.ibm.wmqfte.ant.commandQueueManager` 属性（如果已设置该属性）。如果未设置 `com.ibm.wmqfte.ant.commandQueueManager` 属性，会尝试连接至 `command.properties` 文件中定义的缺省队列管理器。`com.ibm.wmqfte.ant.commandQueueManager` 属性的格式与 `cmdqm` 属性相同，即 `qmgrname@host@port@channel`。

id

必需。指定要取消的传输的传输标识。传输标识是在通过 `fte:filecopy` 和 `fte:filemove` 任务提交传输请求时生成的。

origuser

可选。指定要与 **cancel** 请求相关联的始发用户标识。如果未使用 `origuser` 属性，那么该任务将缺省为使用用于运行 Ant 脚本的用户标识。

示例

将 **fte:cancel** 请求发送给命令队列管理器 `qm0` 的示例。**fte:cancel** 请求针对队列管理器 `qm1` 上的 `agent1` 且由 `transfer.id` 变量填充的传输标识。使用 “bob” 用户标识运行请求。

```
<fte:cancel cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  agent="agent1@qm1"
  id="${transfer.id}"
  origuser="bob"/>
```

相关参考

第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:filecopy

fte:filecopy 任务在 IBM MQ Managed File Transfer 代理程序之间复制文件。该文件不会从源代理删除。

属性

cmdqm

可选。请求提交到的命令队列管理器。按以下格式指定此信息：`qmgrname@host@port@channel`，其中：

- *qmgrname* 是队列管理器的名称
- *host* 是正在运行队列管理器的系统的可选主机名
- *port* 是队列管理器正在侦听的可选端口号
- *channel* 是要使用的可选 SVRCONN 通道

如果省略命令队列管理器的 *host*、*port* 或 *channel* 信息，那么会使用 `command.properties` 文件中指定的连接信息。有关更多信息，请参阅第 588 页的『`command.properties` 文件』。

您可以使用 **com.ibm.wmqfte.propertySet** 属性指定要使用的 `command.properties` 文件。有关更多信息，请参阅 `com.ibm.wmqfte.propertySet`。

如果您不使用 `cmdqm` 属性，那么该任务会缺省使用 `com.ibm.wmqfte.ant.commandQueueManager` 属性（如果已设置该属性）。如果未设置 `com.ibm.wmqfte.ant.commandQueueManager` 属性，会尝试连接至 `command.properties` 文件

中定义的缺省队列管理器。 `com.ibm.wmqfte.ant.commandQueueManager` 属性的格式与 `cmdqm` 属性相同，即 `qmgrname@host@port@channel`。

dst

必需。指定复制操作的目标代理。按以下格式指定此信息：`agentname@qmgrname`，其中 `agentname` 是目标代理的名称，`qmgrname` 是该代理直接连接到的队列管理器的名称。

idproperty

除非您已指定 `outcome` 为 `defer`，否则可选。指定传输标识要分配到的属性的名称。在提交传输请求时会生成传输标识，您可以使用传输标识来跟踪传输进展、诊断传输问题以及取消传输。

如果您还指定了 `ignore` 的 `outcome` 属性，那么不能指定该属性。但是，如果您已将 `outcome` 属性指定为 `defer`，那么还必须指定 `idproperty`。

jobname

可选。将作业名分配到复制请求。您可以使用作业名来创建传输的逻辑组。使用第 961 页的『`fte:uuid`』任务来生成伪唯一的作业名。如果您不使用 `jobname` 属性，那么该任务缺省使用 `com.ibm.wmqfte.ant.jobName` 属性值（如果已设置该属性）。如果您不设置该属性，没有作业名会与复制请求相关联。

origuser

可选。指定与复制请求关联的始发用户标识。如果您不使用 `origuser` 属性，那么该任务会缺省使用用于运行 Ant 脚本的用户标识。

outcome

可选。确定该任务是否等待复制操作完成后再将控制权返回至 Ant 脚本。指定以下某个选项：

await

该任务等待复制操作完成后再返回。如果将 `outcome` 指定为 `await`，那么 `idproperty` 属性为可选。

defer

一旦提交复制请求，该任务会立即返回，并假定复制操作的结果稍候使用 `awaitoutcome` 或第 960 页的『`fte:ignoreoutcome`』任务进行处理。如果将 `outcome` 指定为 `defer`，那么 `idproperty` 属性是必需的。

忽略

如果复制操作的结果不重要，那么可以将值指定为 `ignore`。随后，一旦提交复制请求，该任务会立即返回，而不会分配任何资源用于跟踪传输的结果。如果将 `outcome` 指定为 `ignore`，那么不能指定 `idproperty` 属性。

如果您不指定 `outcome` 属性，那么该任务会缺省使用值 `await`。

priority

可选。指定与复制请求关联的优先级。一般，优先级较高的传输请求的优先顺序高于优先级较低的请求。优先级值必须在 0 - 9（含 0 和 9）的范围内。优先级值 0 是最低的优先级，优先级值 9 是最高的优先级。如果您不指定 `priority` 属性，那么传输会缺省使用优先级值 0。

rcproperty

可选。指定复制请求的结果代码要分配到的属性的名称。结果代码反映了复制请求的总体结果。

如果您已将 `outcome` 属性指定为 `ignore` 或 `defer`，那么无法指定该属性。但是如果您将 `outcome` 指定为 `await`，那么必须指定 `rcproperty`。

src

必需。指定复制操作的源代理。按以下格式指定此信息：`agentname@qmgrname`，其中 `agentname` 是源代理的名称，`qmgrname` 是该代理直接连接到的队列管理器的名称。

指定为嵌套元素的参数

fte:filespec

必需。您必须至少指定一个文件规范以标识要复制的文件。如果需要，您可以指定多个文件规范。请参阅 `fte:filespec` 主题，以获取更多信息。

fte:metadata

您可以指定要与复制操作关联的元数据。此元数据附带于传输，并且在传输生成的日志消息中记录。您只能将一个元数据块与一个给定传输元素相关联；但是，该块可包含多份元数据。请参阅 [fte:metadata](#) 主题以获取更多信息。

fte:presrc

指定在传输启动前，在源代理处发生程序调用。您只能将单个 **fte:presrc** 元素与一个给定传输相关联。请参阅[程序调用主题](#)，以获取更多信息。

fte:predst

指定在传输启动前，在目标代理处发生程序调用。您只能将单个 **fte:predst** 元素与一个给定传输相关联。请参阅[程序调用主题](#)，以获取更多信息。

fte:postsrc

指定在传输完成后，在源代理处发生程序调用。您只能将单个 **fte:postsrc** 元素与一个给定传输相关联。请参阅[程序调用主题](#)，以获取更多信息。

fte:postdst

指定在传输完成后，在目标代理处发生程序调用。您只能将单个 **fte:postdst** 元素与一个给定传输相关联。请参阅[程序调用主题](#)，以获取更多信息。

如果 **fte:presrc**、**fte:predst**、**fte:postsrc**、**fte:postdst** 和出口不返回成功状态，那么按如下顺序指定规则：

1. 运行源起点出口。如果源起点出口失败，那么传输失败，并且不再继续运行。
2. 运行前期源调用（如果存在）。如果前期源调用失败，那么传输失败，并且不再继续运行。
3. 运行目标起点出口。如果目标起点出口失败，那么传输失败，并且不再运行。
4. 运行前期目标调用（如果存在）。如果前期目标调用失败，那么传输失败，并且不再继续运行。
5. 执行文件传输。
6. 运行目标端出口。针对这些出口没有失败状态。
7. 如果传输成功（只要某些文件传输成功，该传输就视为成功），请运行目标后调用（如果存在）。如果目标后调用失败，那么传输将失败。
8. 运行源端出口。针对这些出口没有失败状态。
9. 如果传输成功，请运行源后调用（如果存在）。如果源后调用失败，那么传输将失败。

示例

该示例展示了 **agent1** 与 **agent2** 之间的基本文件传输。将使用客户机传输方式连接将用于启动文件传输的命令发送到名为 **qm0**，的队列管理器。文件传输操作的结果分配给称为 **copy.result** 的属性。

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="copy.result">

  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>

</fte:filecopy>
```

该示例展示了相同的文件传输，但是在传输完成后，添加了元数据，并且在源代理处启动程序。

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="copy.result">

  <fte:metadata>
    <fte:entry name="org.example.departId" value="ACCOUNTS"/>
    <fte:entry name="org.example.batchGroup" value="A1"/>
  </fte:metadata>

  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>

  <fte:postsrc command="/home/fteuser2/scripts/post.sh" successsrc="1" >
    <fte:arg value="/home/fteuser2/file.bin"/>
  </fte:postsrc>
</fte:filecopy>
```



```
</fte:postsrc>  
</fte:filecopy>
```

相关参考

第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:filemove

fte:filemove 任务在 IBM MQ Managed File Transfer 代理程序之间移动文件。当文件成功从源代理传输到目标代理时，会从源代理删除该文件。

属性

cmdqm

可选。请求提交到的命令队列管理器。按以下格式指定此信息：*qmgrname@host@port@channel*，其中：

- *qmgrname* 是队列管理器的名称
- *host* 是正在运行队列管理器的系统的可选主机名
- *port* 是队列管理器正在侦听的可选端口号
- *channel* 是要使用的可选 SVRCONN 通道

如果省略命令队列管理器的 *host*、*port* 或 *channel* 信息，那么会使用 *command.properties* 文件中指定的连接信息。有关更多信息，请参阅第 588 页的『*command.properties* 文件』。

您可以使用 **com.ibm.wmqfte.propertySet** 属性指定要使用的 *command.properties* 文件。有关更多信息，请参阅 *com.ibm.wmqfte.propertySet*。

如果您不使用 *cmdqm* 属性，那么该任务会缺省使用

com.ibm.wmqfte.ant.commandQueueManager 属性（如果已设置该属性）。如果未设置 *com.ibm.wmqfte.ant.commandQueueManager* 属性，会尝试连接至 *command.properties* 文件中定义的缺省队列管理器。*com.ibm.wmqfte.ant.commandQueueManager* 属性的格式与 *cmdqm* 属性相同，即 *qmgrname@host@port@channel*。

dst

必需。指定复制操作的目标代理。按以下格式指定此信息：*agentname@qmgrname*，其中 *agentname* 是目标代理的名称，*qmgrname* 是该代理直接连接到的队列管理器的名称。

idproperty

除非您已指定 *outcome* 为 *defer*，否则可选。指定传输标识要分配到的属性的名称。在提交传输请求时会生成传输标识，您可以使用传输标识来跟踪传输进展、诊断传输问题以及取消传输。

如果您还指定了 *ignore* 的 *outcome* 属性，那么不能指定该属性。但是，如果您已将 *outcome* 属性指定为 *defer*，那么还必须指定 *idproperty*。

jobname

可选。将作业名分配到移动请求。您可以使用作业名来创建传输的逻辑组。使用 *fte:uuid* 任务来生成伪唯一的作业名。如果您不使用 *jobname* 属性，那么该任务缺省使用 *com.ibm.wmqfte.ant.jobName* 属性值（如果已设置该属性）。如果您不设置该属性，没有作业名会与移动请求相关联。

origuser

可选。指定与移动请求关联的始发用户标识。如果您不使用 *origuser* 属性，那么该任务会缺省使用用于运行 Ant 脚本的用户标识。

outcome

可选。确定该任务是否等待移动操作完成后再将控制权返回至 Ant 脚本。指定以下某个选项：

await

该任务等待移动操作完成后再返回。如果将 `outcome` 指定为 `await`，那么 `idproperty` 属性为可选。

defer

一旦提交移动请求，该任务会立即返回，并假定移动操作的结果稍候以第 952 页的『`fte:awaitoutcome`』或第 960 页的『`fte:ignoreoutcome`』任务进行处理。如果将 `outcome` 指定为 `defer`，那么 `idproperty` 属性是必需的。

忽略

如果移动操作的结果不重要，那么可以将值指定为 `ignore`。随后，一旦提交移动请求，该任务会立即返回，而不会分配任何资源用于跟踪传输的结果。如果将 `outcome` 指定为 `ignore`，那么不能指定 `idproperty` 属性。

如果您不指定 `outcome` 属性，那么该任务会缺省使用值 `await`。

priority

可选。指定与移动请求关联的优先级。一般，优先级较高的传输请求的优先顺序高于优先级较低的请求。优先级值必须在 0 - 9（含 0 和 9）的范围内。优先级值 0 是最低的优先级，优先级值 9 是最高的优先级。如果您不指定 `priority` 属性，那么传输会缺省使用优先级值 0。

rcproperty

可选。指定移动请求的结果代码要分配到的属性的名称。结果代码反映了移动请求的总体结果。

如果您已将 `outcome` 属性指定为 `ignore` 或 `defer`，那么无法指定该属性。然而，如果将 `outcome` 指定为 `await`，那么必须指定 `rcproperty`。

src

必需。指定移动操作的源代理。按以下格式指定此信息：`agentname@qmgrname`，其中 `agentname` 是源代理的名称，`qmgrname` 是该代理直接连接到的队列管理器的名称。

指定为嵌套元素的参数**fte:filespec**

必需。您必须至少指定一个文件规范以指定要移动的文件。如果需要，您可以指定多个文件规范。请参阅 [fte:filespec](#) 主题，以获取更多信息。

fte:metadata

可选。您可以指定要与文件移动操作关联的元数据。此元数据附带于传输，并且在传输生成的日志消息中记录。您只能将一个元数据块与一个给定传输元素相关联；但是，该块可包含多份元数据。请参阅 [fte:metadata](#) 主题以获取更多信息。

fte:presrc

可选。指定在传输启动前，在源代理处发生程序调用。您只能将单个 `fte:presrc` 元素与一个给定传输相关联。请参阅[程序调用](#)主题，以获取更多信息。

fte:predst

可选。指定在传输启动前，在目标代理处发生程序调用。您只能将单个 `fte:predst` 元素与一个给定传输相关联。请参阅[程序调用](#)主题，以获取更多信息。

fte:postsrc

可选。指定在传输完成后，在源代理处发生程序调用。您只能将单个 `fte:postsrc` 元素与一个给定传输相关联。请参阅[程序调用](#)主题，以获取更多信息。

fte:postdst

可选。指定在传输完成后，在目标代理处发生程序调用。您只能将单个 `fte:postdst` 元素与一个给定传输相关联。请参阅[程序调用](#)主题，以获取更多信息。

如果 `fte:presrc`、`fte:predst`、`fte:postsrc`、`fte:postdst` 和出口不返回成功状态，那么按如下顺序指定规则：

1. 运行源起点出口。如果源起点出口失败，那么传输失败，并且不再继续运行。
2. 运行前期源调用（如果存在）。如果前期源调用失败，那么传输失败，并且不再继续运行。
3. 运行目标起点出口。如果目标起点出口失败，那么传输失败，并且不再运行。
4. 运行前期目标调用（如果存在）。如果前期目标调用失败，那么传输失败，并且不再继续运行。

5. 执行文件传输。
6. 运行目标端出口。针对这些出口没有失败状态。
7. 如果传输成功（只要某些文件传输成功，该传输就视为成功），请运行目标后调用（如果存在）。如果目标后调用失败，那么传输将失败。
8. 运行源端出口。针对这些出口没有失败状态。
9. 如果传输成功，请运行源后调用（如果存在）。如果源后调用失败，那么传输将失败。

示例

该示例展示了 agent1 与 agent2 之间的基本文件移动。使用客户机传输方式连接将用于启动文件移动的命令发送到名为 qm0，的队列管理器。文件传输操作的结果分配给称为 move.result 的属性。

```
<fte:filemove cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="move.result">

  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filemove>
```

相关参考

第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:ignoreoutcome

忽略 **fte:filecopy**、**fte:filemove** 或 **fte:call** 命令的结果。指定 **fte:filecopy**、**fte:filemove** 或 **fte:call** 任务以获取 defer 的结果时，Ant 任务会分配资源以跟踪此结果。如果您对结果不再感兴趣，可以使用 **fte:ignoreoutcome** 任务来清除这些资源。

属性

id

必需。标识不再感兴趣的结果。通常，您使用使用第 955 页的『fte:filecopy』，第 958 页的『fte:filemove』或第 952 页的『fte:call』任务的 idproperty 属性设置的属性来指定此标识。

示例

该示例显示了您如何使用 fte:ignoreoutcome 任务来释放分配给跟踪先前的第 955 页的『fte:filecopy』任务结果的资源。

```
<!-- issue a file copy request -->
<fte:filecopy cmdqm="qm1@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent1@qm1"
  idproperty="copy.id"
  outcome="defer"/>

<!-- do some other things -->

<!-- decide that the result of the copy is not interesting -->
<fte:ignoreoutcome id="${copy.id}"/>
```

相关参考

第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:ping

Ping 代理以产生响应，以此确定代理是否能够处理传输。

属性

agent

必需。指定将 **fte:ping** 请求提交到的代理。值的格式如下: *agentname@qmgrname*，其中 *agentname* 是代理的名称，*qmgrname* 是该代理直接连接到的队列管理器的名称。

cmdqm

可选。请求提交到的命令队列管理器。按以下格式指定此信息: *qmgrname@host@port@channel*，其中:

- *qmgrname* 是队列管理器的名称
- *host* 是正在运行队列管理器的系统的可选主机名
- *port* 是队列管理器正在侦听的可选端口号
- *channel* 是要使用的可选 SVRCONN 通道

如果省略命令队列管理器的 *host*、*port* 或 *channel* 信息，那么会使用 `command.properties` 文件中指定的连接信息。有关更多信息，请参阅第 588 页的『`command.properties` 文件』。

您可以使用 **com.ibm.wmqfte.propertySet** 属性指定要使用的 `command.properties` 文件。有关更多信息，请参阅 `com.ibm.wmqfte.propertySet`。

如果您不使用 `cmdqm` 属性，那么该任务会缺省使用 `com.ibm.wmqfte.ant.commandQueueManager` 属性（如果已设置该属性）。如果未设置 `com.ibm.wmqfte.ant.commandQueueManager` 属性，会尝试连接至 `command.properties` 文件中定义的缺省队列管理器。`com.ibm.wmqfte.ant.commandQueueManager` 属性的格式与 `cmdqm` 属性相同，即 *qmgrname@host@port@channel*。

rcproperty

必需。命名属性以存储 **ping** 操作的返回码。

timeout

可选。任务等待代理响应的最长时间（秒）。最小超时为零秒，但也可以将超时指定为 -1 秒，这样命令会永远等待代理响应。如果没有为 `timeout` 指定任何值，那么缺省值为最多等待 5 秒供代理响应。

示例

该示例向由 `qm1` 托管的 `agent1` 发送一个 **fte:ping** 请求。**fte:ping** 请求等待 15 秒供代理进行响应。**fte:ping** 请求的结果存储在名为 `ping.rc` 的属性中。

```
<fte:ping agent="agent1@qm1" rcproperty="ping.rc" timeout="15"/>
```

返回码

0

命令成功完成。

2

命令超时。

相关参考

第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:uuid

生成伪随机唯一标识，并将其分配到给定属性。例如，您可以使用该标识为其他文件传输操作生成作业名。

属性

length

必需。要生成的 UUID 的数字长度。该长度值不包括 `prefix` 参数指定的任何前缀的长度。

property

必需。要将生成的 UUID 分配到的属性的名称。

prefix

可选。要添加到生成的 UUID 的前缀。不会将前缀计算为 `length` 参数指定的 UUID 长度的一部分。

示例

该示例定义了一个以字母 ABC 开头，后跟 16 个伪随机十六进制字符的 UUID。UUID 会分配到名为 `uuid.property` 的属性。

```
<fte:uuid length="16" property="uuid.property" prefix="ABC"/>
```

相关参考

第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:filespec

fte:filespec 参数用作为其他任务中的嵌套元素。使用 **fte:filespec** 来描述一个或多个源文件，目录或数据集与目标之间的映射。通常在表述要移动或复制的一组文件、数据集或目录时使用该元素。

嵌套者：

- [fte:filecopy](#) 任务
- [fte:filemove](#) 任务

源规范属性

您必须指定 `srcfilespec` 或 `srcqueue` 之一。

srcfilespec

指定文件操作的源。该属性的值可包含通配符。

srcqueue

指定传输的源为队列。传输会从存储在该属性指定的队列上的消息中移出数据。如果 **fte:filespec** 任务嵌套在 **fte:filecopy** 任务中，那么您不能指定该属性。

当源代理为协议网桥代理时，不支持 `srcqueue` 属性。

目标规范属性

您必须指定 `dstdir`、`dstds`、`dstfilespace`、`dstfile`、`dstqueue` 或 `dstpds` 之一。

dstdir

指定目录作为文件操作的目标。

dstds

指定数据集作为文件操作的目标。

仅当目标代理程序在 z/OS 平台上运行时，才支持该属性。

dstfile

指定文件作为文件操作的目标。

dstfilespace

指定文件空间作为文件操作的目标。

dstpds

指定分区数据集作为文件操作的目标。

仅当目标代理程序在 z/OS 平台上运行时，才支持该属性。

dstqueue

指定队列作为文件到消息操作的目标。您可以选择使用 QUEUE@QUEUEMANAGER 格式将队列管理器名称包含在此规范中。如果未指定队列管理器名称，并且未将 enableClusterQueueInputOutput 代理属性设置为 true，那么将使用目标代理队列管理器。如果将 enableClusterQueueInputOutput 属性设置为 true，那么目标代理将使用标准 WebSphere MQ 过程来确定放置队列的位置。必须指定队列管理器上已存在的有效队列名称。

如果您指定 dstqueue 属性，那么不能指定 srcqueue 属性，原因是这些属性互斥。

当目标代理为协议网桥代理时，不支持 dstqueue 属性。

源选项属性

srcencoding

可选。用于传输文件的字符集编码。

仅当 转换 属性设置为值 text. 时，才能指定此属性

如果您不指定 srcencoding 属性，那么源系统的字符集会用于文本传输。

srceol

可选。正在传输的文件所使用的行末定界符。有效值如下：

- CRLF - 使用回车字符后接换行字符作为行末定界符。此规则通常用于 Windows 系统。
- LF - 使用换行字符作为行末定界符。此规则通常用于 UNIX 系统。

仅当 conversion 属性值设置为 text 时，才可以指定该属性。如果您不指定 srceol 属性，那么文本传输会基于源代理的操作系统自动确定正确的值。

srckeeptrailingspaces

可选。确定在从固定长度格式的数据集读取的源记录上是否保留尾部空格作为文本方式传输的一部分。有效值如下：

- true - 保留尾部空格。
- false - 删除尾部空格。

如果您不指定 srckeeptrailingspaces 属性，那么会指定缺省值 false。

仅当还指定了 srcfilespec 属性并且将 转换 属性设置为值 text. 时，才能指定此属性

srcmsgdelimbytes

可选。指定一个或多个字节的值，以在将多条消息追加到二进制文件时作为定界符插入。每个值必须指定为两个十六进制数字（00 到 FF 之间）并以 x 为前缀。多个字节必须以逗号分隔。例如，srcmsgdelimbytes="x08,xA4"。仅当同时指定了 srcqueue 属性时，才可以指定 srcmsgdelimbytes 属性。如果已将 conversion 属性值指定为 text，那么不能指定 srcmsgdelimbytes 属性。

srcmsgdelimtext

可选。指定一个文本序列，以在将多条消息追加到文本文件时作为定界符插入。可以在定界符中包含字符串字面值的 Java 转义序列。例如，srcmsgdelimtext="\u007d\n"。文本定界符由源代理插入在每条消息之后。文本定界符使用传输的源编码来编码为二进制格式。每条消息均以二进制格式读取，编

码的定界符以二进制格式追加到消息，从而以二进制格式传输至目标代理。如果源代理代码页包含移入和移出状态，那么代理假定每条消息在消息结束时进入移出状态。在目标代理上，以与文件相同的方式将二进制数据转换为文件文本传输。仅当同时指定了 `srcqueue` 属性，并且将 `conversion` 属性的值指定为 `text` 时，才可以指定 `srcmsgdelimtext` 属性。

srcmsgdelimposition

可选。指定文本或二进制定界符插入的位置。有效值如下：

- `prefix` - 定界符插入目标文件中来自每条消息的数据之前的位置。
- `postfix` - 定界符插入目标文件中来自每条消息的数据之后的位置。

仅当指定了 `srcmsgdelimbytes` 或 `srcmsgdelimtext` 属性时，才可以指定 `srcmsgdelimposition` 属性。

srcmsggroups

可选。指定按 WebSphere MQ 组标识分组的消息。第一个完整组将写入目标文件。如果未指定该属性，那么源队列上的所有消息都会写入目标文件。仅当同时指定了 `srcqueue` 属性时，才可以指定 `srcmsggroups` 属性。

srcqueuetimeout

可选。指定等待以下某个条件得以满足的时间（秒）：

- 等待新消息写入队列。
- 如果已指定 `srcmsggroups` 属性，那么等待完整的组写入队列。

如果在 `srcqueuetimeout` 的值指定的时间内没有任何条件得到满足，那么源代理会停止从队列读取，并完成传输。如果未指定 `srcqueuetimeout` 属性，那么当源队列为空时，或者当队列上没有完整的组并指定了 `srcmsggroups` 属性时，源代理会立即停止从源队列读取。仅当同时指定了 `srcqueue` 属性时，才可以指定 `srcqueuetimeout` 属性。

有关设置 `srcqueuetimeout` 的值的 [信息](#)，请参阅第 744 页的『用于对“消息到文件”传输指定等待时间的指南』。

srcrecdelimbytes

可选。指定一个或多个字节的值，以在将多条记录从面向记录的源文件追加到二进制文件时作为定界符插入。必须将每个值指定为 00-FF 范围内的两个十六进制数字，前缀为 `x`。多个字节必须以逗号分隔。例如：

```
srcrecdelimbytes="x08,xA4"
```

仅当传输源文件面向记录（例如 z/OS 数据集），并且目标文件为正常的、非面向记录的文件时，才可以指定 `srcrecdelimbytes` 属性。如果已针对 `conversion` 属性指定值 `text`，那么不能指定 `srcrecdelimbytes` 属性。

srcrecdelimpos

可选。指定二进制定界符插入的位置。有效值如下：

- `prefix` - 定界符插入目标文件中来自每条面向记录的源文件记录的数据之前的位置。
- `postfix` - 定界符插入目标文件中来自每条面向记录的源文件记录的数据之后。

仅当同时指定了 `srcrecdelimbytes` 属性时，才可以指定 `srcrecdelimpos` 属性。

目标选项属性

dstAttributes

可选。指定与传输中的目标文件关联的以分号分隔的文件属性列表。您可以指定具有值或没有值的属性。

例如，没有值的属性：

```
dstAttributes="ATTRIBUTE1;ATTRIBUTE2"
```

例如，具有值的属性：

```
dstAttributes="ATTRIBUTE1(VALUE);ATTRIBUTE2(VALUE)"
```

例如，一个具有值的属性和一个没有值的属性：

```
dstAttributes="ATTRIBUTE1;ATTRIBUTE2(VALUE)"
```

有关 IBM 4690 上 IBM MQ Managed File Transfer 的文件属性的信息，请参阅第 82 页的『文件分发属性』。

dstencoding

可选。用于传输的文件的字符集编码。

仅当转换属性设置为值 `text` 时，才能指定此属性

如果未指定 `dstencoding` 属性，那么目标系统的字符集会用于文本传输。

dsteol

可选。用于传输的文件的行未定界符。有效值如下：

- CRLF - 使用回车字符后接换行字符作为行未定界符。此规则通常用于 Windows 系统。
- LF - 使用换行字符作为行未定界符。此规则通常用于 UNIX 系统。

仅当转换属性设置为值 `text` 时，才能指定此属性

如果您不指定 `dsteol` 属性，那么文本传输会基于目标代理程序的操作系统自动确定正确的值。

dstmsgdelimbytes

可选。指定将二进制文件分割为多条消息时使用十六进制定界符。所有消息都具有相同的 WebSphere MQ 组标识；组中的最后一条消息设置了 WebSphere MQ `LAST_MSG_IN_GROUP` 标志。将十六进制字节指定为定界符的格式为 `xNN`，其中 `N` 是范围在 0 - 9 或 `a - f` 内的字符。您可以通过指定十六进制字节的逗号分隔列表（例如：`x3e,x20,x20,xbf`），将一系列十六进制字节指定为定界符。

仅当同时指定了 `dstqueue` 属性并且以二进制方式进行传输时，才可以指定 `dstmsgdelimbytes` 属性。您只能指定 `dstmsgsize`、`dstmsgdelimbytes` 或 `dstmsgdelimpattern` 属性之一。

dstmsgdelimpattern

可选。指定将文本文件分割为多条消息时要使用的 Java 正则表达式。所有消息都具有相同的 WebSphere MQ 组标识；组中的最后一条消息设置了 WebSphere MQ `LAST_MSG_IN_GROUP` 标志。指定正则表达式作为定界符的格式为以括号括起的正则表达式 (*regular_expression*)，或者以双引号括起的正则表达式 "*regular_expression*"。有关更多信息，请参阅第 724 页的『IBM MQ Managed File Transfer 使用的正则表达式』。

缺省情况下，目标代理将正则表达式可匹配的字符串的长度限制为 5 个字符。您可使用 `maxDelimiterMatchLength` 代理属性来更改此行为。有关更多信息，请参阅第 591 页的『高级代理属性』。

仅当同时指定了 `dstqueue` 属性并且以文本方式进行传输时，才可以指定 `dstmsgdelimpattern` 属性。您只能指定 `dstmsgsize`、`dstmsgdelimbytes` 或 `dstmsgdelimpattern` 属性之一。

dstmsgdelimposition

可选。指定期望文本或二进制定界符所在的位置。有效值如下：

- `prefix` - 期望定界符位于每行开头。
- `postfix` - 期望定界符位于每行末尾。

仅当同时指定了 `dstmsgdelimpattern` 属性时，才可以指定 `dstmsgdelimposition` 属性。

dstmsgincludedelim

可选。指定用于将文件分割为多条消息的定界符是否要包括在消息中。如果已指定 `dstmsgincludedelim` 属性，那么定界符包含在包含定界符前的文件数据的消息末尾。缺省情况下，消息

中不包含定界符。仅当同时指定了 `dstmsgdelimpattern` 或 `dstmsgdelimbytes` 属性时，才可以指定 `dstmsgincludedelim` 属性。

dstmsgpersist

可选。指定写入目标队列的消息是否持久。有效值如下：

- `true` - 将持久消息写入到目标队列。这是缺省值。
- `false` - 将非持久消息写入到目标队列。
- `qdef` - 持久性值取自目标队列的 `DefPersistence` 属性。

仅当同时指定了 `dstqueue` 属性时，才可以指定该属性。

dstmsgprops

可选。指定传输写入目标队列的第一条消息是否已设置 WebSphere MQ 消息属性。可能的值为：

- `true` - 在传输创建的第一条消息上设置消息属性。
- `false` - 不在传输创建的第一条消息上设置消息属性。这是缺省值。

要了解更多信息，请参阅第 739 页的『写入到目标队列的消息上设置的 IBM MQ 消息属性』。

仅当同时指定了 `dstqueue` 属性时，才可以指定该属性。

dstmsgsize

可选。指定是否将文件分割为多条固定长度的消息。所有消息均具有相同的 WebSphere MQ 组标识；并且组中的最后一条消息设置了 WebSphere MQ `LAST_MSG_IN_GROUP` 标志。消息的大小由 `dstmsgsize` 的值指定。The format of `Dstmsgsize` is `<长度><单位>`, where 长度 is a positive integer value and 单位 is one of the following values:

- B - 字节。允许的最小值是目标消息的代码页的最大“每个字符的字节数”值的两倍。
- K - 千字节。等于 1024 字节。
- M - 兆字节。等于 1024 千字节。

如果文件以文本方式传输，并且采用双字节字符集或者多字节字符集，那么该文件会拆分为字符边界最接近指定消息大小的消息。

仅当同时指定了 `dstqueue` 属性时，才能指定 `dstmsgsize` 属性。您只能指定 `dstmsgsize`、`dstmsgdelimbytes` 或 `dstmsgdelimpattern` 属性之一。

dstunsupportedcodepage

可选。指定 `dstqueue` 属性指定的目标队列管理器不支持将文件数据作为文本传输至队列时所使用的代码页时采取的操作。该属性的有效值如下：

- `binary` - 继续传输，但是不将代码页转换应用到传输的数据。指定该值等于不将转换属性设置为 `text`。
- `fail` - 不继续传输操作。文件记录为传输失败。这是缺省值。

仅当同时指定了 `dstqueue` 属性，并且 `conversion` 属性的值为 `text` 时，才可以指定 `dstunsupportedcodepage` 属性。

dsttruncaterecords

可选。指定将截断长于 LRECL 数据集属性的目标记录。如果设置为 `true`，将截断记录。如果设置为 `false`，将记录文本换行。缺省设置为 `false`。该参数仅对目标是数据集的文本方式传输有效。

其他属性

checksum

可选。确定用于对传输文件进行校验和的算法。

- MD5 - 使用 MD5 散列算法。
- NONE - 不使用校验和算法。

如果您不指定 `checksum` 属性，那么会使用缺省值 MD5。

conversion

可选。指定应用于正在传输的文件的转换类型。可能的值为：

- `binary` - 不应用转换。
- `text` - 在源系统和目标系统之间应用代码页转换。同时应用行定界符的转换。`srcencoding`、`dstencoding`、`srceol` 和 `dsteol` 属性会影响所应用的转换。

如果您不指定 `conversion` 属性，那么会指定缺省值 `binary`。

overwrite

可选。确定操作是否可覆盖现有目标文件或数据集。指定值 `true` 时，会覆盖任何现有的目标文件或数据集。指定值 `false` 时，如果目标位置存在重复的文件或数据集，会导致操作失败。如果未指定 `overwrite` 属性，那么会指定缺省值 `false`。

recurse

可选。确定文件传输是否递归至子目录中。指定值为 `true` 时，传输递归至子目录中。指定值为 `false` 时，传输不会递归至子目录中。如果未指定 `recurse` 属性，那么会指定缺省值 `false`。

示例

此示例指定具有源文件 `file1.bin` 和目标文件 `file2.bin` 的 `fte:filespec`

```
<fte:filespec srcfilespec="/home/fteuser/file1.bin" dstfile="/home/fteuser/file2.bin"/>
```

相关参考

[第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』](#)

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

[第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』](#)

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:metadata

元数据用于通过文件传输操作传送更多用户定义的信息。

有关 IBM MQ Managed File Transfer 如何使用元数据的更多信息，请参阅[第 971 页的『用户出口例程元数据』](#)。

嵌套者：

- [fte:filecopy](#) 任务
- [fte:filemove](#) 任务
- [fte:call](#) 任务

指定为嵌套元素的参数

fte:entry

必须在 `fte:metadata` 嵌套元素中至少指定一个条目。您可以选择指定多个入口。入口将一个键名称与一个值关联。键在 `fte:metadata` 块中必须唯一

入口属性

name

必需。属于该入口的键的名称。该名称在 `fte:metadata` 元素内嵌套的所有 `entry` 参数间必须唯一。

value

必需。要分配给该入口的值。

示例

此示例显示包含两个条目的 `fte:metadata` 定义。

```
<fte:metadata>
  <fte:entry name="org.foo.partColor" value="red"/>
  <fte:entry name="org.foo.partSize" value="medium"/>
</fte:metadata>
```

相关参考

第 354 页的『[将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用](#)』

IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。

第 950 页的『[IBM MQ Managed File Transfer 提供的 Ant 任务](#)』

IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

程序调用嵌套元素

可使用五个嵌套元素中的一个来启动程序：`fte:presrc`、`fte:predst`、`fte:postdst`、`fte:postsrc` 和 `fte:command`。这些嵌套元素指示代理作为其处理过程的一部分来调用外部程序。要能够启动程序，必须确保命令位于运行该命令的代理的 `agent.properties` 文件中 `commandPath` 属性所指定的位置中。

即使每个程序调用元素的名称不同，它们也可共享同一组属性和同一组嵌套元素。可通过 **`fte:filecopy`**、**`fte:filemove`** 和 **`fte:command`** Ant 任务启动程序。如果已配置 Web 网关以允许将文件上载至代理，请通过指定 `x-fte-postdest` 头或在 HTTP 请求中使用 `postdest` 表单字段来配置 `fte:postdst` 程序调用。

不能从 Connect:Direct 网桥代理调用程序。

可调用程序的 Ant 任务：

- `fte:filecopy` 任务使用 `fte:predst`、`fte:postdst`、`fte:presrc` 和 `fte:postsrc` 嵌套元素来嵌套程序调用参数。
- `fte:filemove` 任务使用 `fte:predst`、`fte:postdst`、`fte:presrc` 和 `fte:postsrc` 嵌套元素来嵌套程序调用参数。
- `fte:call` 任务使用 `fte:command` 嵌套元素来嵌套程序调用参数。

属性

command

必需。指定要调用的程序。要使代理能够运行命令，该命令必须在代理的 `agent.properties` 文件中 `commandPath` 属性所指定的位置。要了解更多信息，请参阅第 443 页的『[commandPath 属性](#)』。`command` 属性中指定的所有路径信息视为相对于 `commandPath` 属性指定的位置。如果 `type` 为 `executable`，将期望可执行程序，否则将期望适合于调用类型的脚本。

retrycount

可选。在程序未返回成功返回码时重试调用程序的次数。由 `command` 属性指定的程序最多可调用此数目的次数。为该属性指定的值必须为非负数。如果未指定 `retrycount` 属性，那么将使用缺省值 0。

retrywait

可选。重新尝试调用程序之前，要等待的秒数。如果 `command` 属性指定的程序未返回成功返回码，并且 `retrycount` 属性指定非零值，那么该参数可确定重试之间等待的时间。为该属性指定的值必须为非负数。如果未指定 `retrywait` 属性，那么将使用缺省值 0。

successrc

可选。该属性的值用于确定程序调用成功运行的时间。该表达式用于对命令的进程返回码进行求值。The value can be composed of one or more expressions combined with a vertical bar character (|) to signify Boolean 或, or an ampersand (&) character to signify Boolean 与. 每个表达式可以是以下类型之一：

- 指示进程返回码和数字间的等同性测试的数字。
- 以 “>” 字符为前缀的数字，用于指示数字与进程返回码之间的大于检验。
- 以 “<” 字符为前缀的数字，用于指示数字与进程返回码之间的 “小于” 测试。
- 以 “!” 为前缀的数字 字符为前缀的数字，指示该数字和进程返回码间的 “不等于” 测试。

例如: >2&<7&!5|0|14 解释为以下返回码成功: 0, 3, 4, 6 和 14。所有其他返回码都解释为不成功。如果未指定 `successrc` 属性, 那么将使用缺省值 0。这表示当且仅当该命令返回码为 0 时, 才会 “判定” 该命令运行成功。

priority

可选 (仅限 `os4690background`)。分配给 IBM 4690 系统上后台任务的优先级别。缺省值为 5, 有效值范围为 1 - 9。

message

可选 (仅限 `os4690background`)。针对执行的命令显示在 IBM 4690 系统后台控制屏幕上的状态消息。

type

可选。该属性的值指定正在调用的程序的类型。指定以下某个选项:

executable

该任务调用可执行程序。可使用 `arg` 嵌套元素指定其他自变量。可在 `commandPath` 上访问该程序, 并在适用情况下具有执行许可权集。只要指定 `shell` 程序 (例如, `shell` 脚本文件的第一行是: `#!/bin/sh`), 就可以调用 UNIX 脚本。写入 `stderr` 或 `stdout` 的命令输出将发送到调用的 WebSphere MQ File Transfer Edition 日志。然而, 数据输出量受代理配置的限制。缺省值为 10KB 数据, 但可使用代理属性 `maxCommandOutput` 覆盖此缺省值。

antscript

该任务使用 **fteAnt** 命令运行指定的 Ant 脚本。可使用 `property` 嵌套元素指定属性。可使用 `target` 嵌套元素指定 Ant 目标。应在 `commandPath` 上访问 Ant 脚本。写入 `stderr` 或 `stdout` 的 Ant 输出会被发送到 IBM MQ Managed File Transfer 日志进行调用。然而, 数据输出量受代理配置的限制。缺省值为 10KB 数据, 但可使用代理属性 `maxCommandOutput` 覆盖此缺省值。

jcl

`jcl` 值仅在 z/OS 上受支持, 并可运行指定的 z/OS JCL 脚本。JCL 作为作业提交且需要有作业卡。成功提交作业时, 写入 IBM MQ Managed File Transfer 日志的 JCL 命令输出包含以下文本: `JOB job_name(job_id)`, 其中:

- `job_name` 是 JCL 中的作业卡所标识的作业的名称。
- `job_id` 是 z/OS 系统生成的作业标识。

如果不能成功提交作业, 那么 JCL 脚本命令将失败并将指出失败原因 (例如, 不存在任何作业卡) 的消息写入日志中。要了解是否成功运行或完成该作业, 请使用系统服务, 如 SDSF。由于 IBM MQ Managed File Transfer 只提交作业, 因此不会提供信息; 系统会确定何时运行该作业以及如何显示作业输出。由于将 JCL 脚本提交为批处理作业, 因此不建议为 `presrc` 或 `predst` 嵌套元素指定 `jcl`, 因为您只知道已成功提交该作业, 但不知道该作业是否在传输开始前成功运行直至完成。`jcl` 类型的嵌套元素均无效。

以下示例显示了 JCL 作业:

```
//MYJOB JOB
//*
//MYJOB EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=H
//SYSUT1 DD DSN=FRED.DEMO.TXT,DISP=SHR
//SYSUT2 DD DSN=BOB.DEMO.TXT,DISP=(NEW,CATLG),
// RECFM=VB,LRECL=133,BLKSIZE=2048,
// SPACE=(TRK,(30,5),RLSE)
//SYSIN DD DUMMY
```

os4690background

该任务调用 OS4690BACKGROUND 程序。可以在 IBM® 4690 系统上后台运行传输程序调用。这些调用类型支持运行本机 IBM 4690 应用程序和批处理命令脚本。如果您运行批处理脚本，那么将带有 -C 选项的 COMMAND.286 应用程序作为应用程序使用，将该批处理脚本和任何指定自变量作为参数传递。

指定为嵌套元素的参数

fte:arg

只有在 type 属性值为 executable 时才有效。使用嵌套 fte:arg 元素，为作为程序调用的一部分而调用的程序指定自变量。程序自变量以遇到 fte:arg 元素的顺序从 fte:arg 元素指定的值构建。您可以选择指定零个或更多的 fte:arg 元素作为程序调用的嵌套元素。

fte:property

只有在 type 属性值为 antscript 时才有效。使用嵌套 fte:property 元素的 name 和 value 属性，以名称值对传递到 Ant 脚本。您可以选择指定零个或更多的 fte:property 元素作为程序调用的嵌套元素。

fte:target

只有在 type 属性值为 antscript 时才有效。指定要调用的 Ant 脚本中的目标。您可以选择指定零个或更多的 fte:target 元素作为程序调用的嵌套元素。

Arg 属性

value

必需。要传递到正在调用的程序的自变量的值。

属性

name

必需。传递到 Ant 脚本的属性的名称。

value

必需。与正在传递到 Ant 脚本的属性名相关联的值。

示例

该示例显示了指定为 fte:filecopy 任务的一部分的 fte:postsrc 程序调用。程序调用适用于名为 post.sh 的程序，并且提供了 /home/fteuser2/file.bin. 的单个自变量

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="copy.result">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>

  <fte:postsrc command="post.sh" successsrc="1" >
    <fte:arg value="/home/fteuser2/file.bin"/>
  </fte:postsrc>
</fte:filecopy>
```

该示例显示了指定为 fte:call 任务的一部分的 fte:command 程序调用。程序调用针对名为 command.sh 的可执行文件进行，该文件未传递任何命令行自变量。如果 command.sh 未返回成功返回码 1，那么会在 30 秒后重试该命令。

```
<fte:call cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  agent="agent1@qm1"
  rcproperty="call.rc"
  origuser="bob"
  jobname="{job.id}">
  <fte:command command="command.sh" successsrc="1" retrycount="5" retrywait="30"/>
</fte:call>
```

该示例显示了指定为 `fte:call` 任务的一部分的 `fte:command` 程序调用。程序调用针对名为 `script.xml` 的 Ant 脚本中的复制和压缩目标进行，该脚本可传递两个属性。

```
<fte:call cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  agent="agent1@qm1"
  rcproperty="call.rc"
  origuser="bob"
  jobname="{${job.id}}">
  <fte:command command="script.xml" type="antscript">
    <property name="src" value="AGENT5@QM5"/>
    <property name="dst" value="AGENT3@QM3"/>
    <target name="copy"/>
    <target name="compress"/>
  </fte:command>
</fte:call>
```

相关概念

第 303 页的『指定要运行的程序』
您可在正在运行 IBM MQ Managed File Transfer 代理的系统上运行程序。作为文件传输请求的一部分，您可以指定在传输启动前或者完成后运行的程序。此外，您可以通过提交受管呼叫请求来启动不属于文件传输请求的程序。

相关参考

第 354 页的『将 Apache Ant 与 IBM MQ Managed File Transfer 结合使用』
IBM MQ Managed File Transfer 提供了一些任务，您可以使用这些任务将文件传输功能集成到 Apache Ant 工具中。
第 950 页的『IBM MQ Managed File Transfer 提供的 Ant 任务』
IBM MQ Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

使用用户出口进行定制

用户出口例程元数据

可向 IBM MQ Managed File Transfer 的用户出口例程提供三种不同类型的元数据：环境、传输和文件元数据。此元数据显示为 Java 键/值对的映射。

环境元数据

环境元数据传递到所有用户出口例程，并描述从中调用用户出口例程的代理运行时环境。该元数据是只读数据，不能由任何用户出口例程更新。

表 77: 环境元数据	
键	描述
AGENT_CONFIGURATION_DIRECTORY_KEY	包含代理配置信息的目录的名称。
AGENT_PRODUCT_DIRECTORY_KEY	已在其中安装了代理代码的目录的名称。
AGENT_VERSION_KEY	调用出口例程的代理运行时的版本号。

表 1 中提供的键名和值名是 EnvironmentMetaDataConstants 接口中定义的常量。

传输元数据

传输元数据传递到所有用户出口例程。元数据包括系统提供的值和用户提供的值。如果更改任何系统提供的值，那么将忽略这些更改。最初由用户为源传输启动用户出口提供的值是基于在定义传输时提供的值。源代理可作为处理源传输启动用户出口的一部分来更改用户提供的值。在启动整个文件传输前将调用此用户出口。这些更改将在后续调用与此传输相关的其他出口例程时使用。传输元数据应用于整个传输。

虽然所有用户出口可从传输元数据读取值，但只有源传输启动用户出口才能更改传输元数据
不能使用传输元数据在不同的文件传输间传播信息。

表 2 中详细描述了系统提供的传输元数据：

表 78: 传输元数据	
键	描述
DESTINATION_AGENT_KEY	作为传输目标的代理的名称。
JOB_NAME_KEY	与传输请求相关联的作业名
MQMD_USER_KEY	用于提交传输请求的消息的 MQMD 用户字段
ORIGINATING_HOST_KEY	传输请求中指定为始发主机名的主机名
ORIGINATING_USER_KEY	传输请求中指定为始发用户标识的用户名
SOURCE_AGENT_KEY	作为传输的源的代理名称
TRANSFER_ID_KEY	传输标识

表 2 中提供的键名和值名是 TransferMetaDataConstants 接口中定义的常量。

文件元数据

文件元数据作为文件规范的一部分而传递到源传输启动出口。源和目标文件存在独立的文件元数据。

不能使用文件元数据在不同的文件传输间传送信息。

表 79: 文件元数据		
键	允许的值	描述
CONVERT_LINE_SEPARATORS		用于文本传输的键值，指示是否将源数据中的 CRLF（回车-换行符）或 LF（换行符）行分隔符序列转换为目标中的行分隔符序列。
DELIMITER_KEY		用于定义定界符的键值，将面向记录的数据传输至常规文件时，该定界符用于分隔记录数据。 也用于消息到文件和文件到消息传输。
DELIMITER_POSITION_KEY	DELIMITER_POSITION_PREFIX_VALUE DELIMITER_POSITION_POSTFIX_VALUE	与 DELIMITER_KEY 一起使用，定义定界符的位置；前缀或后缀。
DELIMITER_TYPE_KEY	DELIMITER_TYPE_BINARY_VALUE DELIMITER_TYPE_TEXT_VALUE DELIMITER_TYPE_SIZE_VALUE	与 DELIMITER_KEY 一起使用，定义定界符的类型。
DESTINATION_EXIST_KEY	DESTINATION_EXIST_KEY_ERROR_VALUE DESTINATION_EXIST_KEY_OVERWRITE_VALUE	确定目标文件存在时的文件传输行为。
FILE_ALIAS_KEY		用于为正在传输的文件定义别名的键值。
FILE_CHECKSUM_METHOD_KEY	FILE_CHECKSUM_METHOD_NONE_VALUE FILE_CHECKSUM_METHOD_MD5_VALUE	确定传输文件时要使用的校验和方法。
FILE_CONVERSION_KEY	FILE_CONVERSION_TEXT_VALUE FILE_CONVERSION_BINARY_VALUE	确定应用于文件内容的转换的类型。
FILE_ENCODING_KEY		确定用于文本文件的编码。
FILE_END_OF_LINE_KEY	FILE_END_OF_LINE_LF_VALUE FILE_END_OF_LINE_CRLF_VALUE	确定表示行尾的字符序列：<LF> 或 <CR><LF>。
FILE_SPACE_ALIAS		确定文件空间中文件的别名。 注： 仅当 FILE_TYPE_KEY 为 FILE_TYPE_FILE_SPACE_VALUE 时才能使用此元数据

表 79: 文件元数据 (继续)		
键	允许的值	描述
FILE_SPACE_NAME		确定文件空间的名称。 注: 仅当 FILE_TYPE_KEY 为 FILE_TYPE_FILE_SPACE_VALUE 时才能使用此元数据
FILE_TYPE_KEY	FILE_TYPE_FILE_VALUE FILE_TYPE_DIRECTORY_VALUE FILE_TYPE_DATASET_VALUE FILE_TYPE_PDS_VALUE FILE_TYPE_QUEUE_VALUE FILE_TYPE_FILE_SPACE_VALUE	确定目标文件、队列或文件空间规范。
GROUP_ID_KEY		用于消息到文件传输的键值，确定要从源队列读取的消息组。仅当 USE_GROUPS_KEY 的值为 USE_GROUPS_TRUE_VALUE 时，该属性才有效。
INCLUDE_DELIMITER_IN_MESSAGE_KEY	INCLUDE_DELIMITER_IN_MESSAGE_TRUE_VALUE INCLUDE_DELIMITER_IN_MESSAGE_FALSE_VALUE	用于文件到消息传输的键值，确定是否要在消息尾部包含用于将文件拆分为多条消息的定界符。仅当 DELIMITER_TYPE_KEY 的值为 DELIMITER_TYPE_BINARY_VALUE DELIMITER_TYPE_TEXT_VALUE 时，该属性才有效。
INSERT_RECORD_LINE_SEPARATOR_KEY		用于来自面向记录文件的文本传输的键值，指定是否在每条记录后将行分隔符插入数据。
KEEP_TRAILING_SPACES_KEY	KEEP_TRAILING_SPACES_TRUE_VALUE KEEP_TRAILING_SPACES_FALSE_VALUE	用于确定是否从固定长度格式数据集读取的记录中除去结尾空格的键值。
NEW_RECORD_ON_LINE_SEPARATOR_KEY		用于至面向记录文件的文本传输的键值，指定将数据中的行分隔符与记录数据包含在一起，还是生成新记录（且不写入）。
PERSISTENT_KEY	PERSISTENT_TRUE_VALUE PERSISTENT_FALSE_VALUE PERSISTENT_QDEF_VALUE	用于文件到消息传输的键值，确定消息是否为持久消息。
SET_MQ_PROPS_KEY	SET_MQ_PROPS_TRUE_VALUE SET_MQ_PROPS_FALSE_VALUE	用于文件到消息传输的键值，确定在文件中的第一条消息上和发生错误时写入队列的任何消息上是否设置 IBM MQ 消息属性。
UNRECOGNISED_CODE_PAGE_KEY	UNRECOGNISED_CODE_PAGE_FAIL_VALUE UNRECOGNISED_CODE_PAGE_BINARY_VALUE	用于文件到消息传输的键值，确定在目标队列管理器无法识别数据代码页时，文本方式传输将失败还是执行转换。
USE_GROUPS_KEY	USE_GROUPS_TRUE_VALUE USE_GROUPS_FALSE_VALUE	用于消息到文件传输的键值，确定是否仅传输源队列中的一个完整消息组。
WAIT_TIME_KEY		用于消息到文件传输的键值，确定源代理等待满足以下某个条件的时间（秒）： - 在源队列上出现一条消息（前提是队列为空或已变为空，并且 USE_GROUPS_KEY 的值为 FALSE）。 - 在源队列上出现一个完整组（前提是 USE_GROUPS_KEY 的值为 TRUE）。

表 3 中提供的键名和值名是 FileMetaDataConstants 接口中定义的常量。

相关概念

[第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』](#)

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

第 979 页的『用户出口例程的 Java 接口』

使用本节中的主题以获取有关用户出口例程的 Java 接口的参考信息。

相关参考

第 974 页的『资源监视器用户出口』

通过资源监视器用户出口，您可以配置在满足监视器的触发条件的情况下，要在关联任务启动之前运行的定制代码。

第 977 页的『用户出口的代理属性』

除 `agent.properties` 文件中的标准属性外，还存在多个特定于用户出口例程的高级属性。缺省情况下不包含这些属性，因此要使用它们中的任何一个，都必须手动编辑 `agent.properties` 文件。如果在代理运行期间对 `agent.properties` 文件进行了更改，请停止并重新启动该代理以提取这些更改。

资源监视器用户出口

通过资源监视器用户出口，您可以配置在满足监视器的触发条件的情况下，要在关联任务启动之前运行的定制代码。

建议不要直接通过用户出口代码调用新传输。在某些情况下，由于用户出口无法适应代理重新启动，这将导致传输文件多次。

资源监视器用户出口将现有基础结构用于用户出口。调用监视器用户出口的时间为：在触发监视器后，但在监视器的任务运行相应的任务前。这允许用户出口修改要运行的任务并确定任务是否应继续。您可以通过更新监视器元数据来修改监视器任务，此元数据随后用于在创建原始监视器时创建的任务文档中进行变量替换。或者，监视器出口可替换或更新作为参数传递的任务定义 XML 字符串。监视器出口可针对任务返回结果代码“`proceed`”或“`cancel`”。如果返回“`cancel`”，那么将不会启动任务，并且直到所监视的资源符合触发条件时，才会重新启动监视器。如果资源尚未更改，那么触发器将不会启动。如同其他用户出口，您可以将监视器出口连接到一起。如果某一出口返回结果代码“`cancel`”，那么将取消整个结果且不启动任务。

- 环境元数据图（与其他用户出口相同）
- 监视器元数据（包括不变的系统元数据和变化的用户元数据）图。不变的系统元数据如下：
 - `FILENAME` - 满足触发条件的文件的名称
 - `FILEPATH` - 满足触发条件的文件的路径
 - `FILESIZE`（字节 - 此元数据可能不会显示）- 满足触发条件的文件的大小
 - `LASTMODIFIEDDATE`（本地）- 上一次更改满足触发条件的文件的日期。该日期表示为代理运行所在时区的本地日期，并采用 ISO 8601 日期格式。
 - `LASTMODIFIEDTIME`（本地）- 上一次更改满足触发条件的文件的时间（本地格式）。该时间表示为代理运行在的时区的本地时间，并采用 ISO 8601 时间格式。
 - `LASTMODIFIEDDATEUTC` - 上一次更改满足触发条件的文件的日期（通用格式）。该日期表示为已转换为 UTC 时区的本地日期，采用 ISO 8601 日期格式。
 - `LASTMODIFIEDTIMEUTC` - 上一次更改满足触发条件的文件的时间（通用格式）。该时间表示为转换为 UTC 时区的本地时间，并采用 ISO 8601 时间格式。
 - `AGENTNAME` - 监视器代理名称
- 表示作为监视器触发器的结果运行的任务的 XML 字符串。

监视器出口返回以下数据：

- 指定是否进一步执行的指示符（`proceed` 或 `cancel`）
- 插入到满足触发条件的日志消息中的字符串

由于运行监视器出口代码，最初作为参数传递的监视器元数据和任务定义 XML 字符串也可能已经更新。

代理属性 `monitorExitClasses` 的值（`agent.properties` 文件中）指定要装入的监视器出口类（由逗号分隔每个出口类）。例如：

```
monitorExitClasses=testExits.TestExit1,testExits.testExit2
```

监视器用户出口的接口为:

```
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately prior to starting a task as the result of a monitor trigger
 */
public interface MonitorExit {

    /**
     * Invoked immediately prior to starting a task as the result of a monitor
     * trigger.
     *
     * @param environmentMetaData
     *      meta data about the environment in which the implementation
     *      of this method is running. This information can only be read,
     *      it cannot be updated by the implementation. The constant
     *      defined in <code>EnvironmentMetaDataConstants</code> class can
     *      be used to access the data held by this map.
     *
     * @param monitorMetaData
     *      meta data to associate with the monitor. The meta data passed
     *      to this method can be altered, and the changes will be
     *      reflected in subsequent exit routine invocations. This map
     *      also contains keys with IBM reserved names. These entries are
     *      defined in the <code>MonitorMetaDataConstants</code> class and
     *      have special semantics. The the values of the IBM reserved names
     *      cannot be modified by the exit
     *
     * @param taskDetails
     *      An XML String representing the task to be executed as a result of
     *      the monitor triggering. This XML string may be modified by the
     *      exit
     *
     * @return
     *      a monitor exit result object which is used to determine if the
     *      task should proceed, or be cancelled.
     */
    MonitorExitResult onMonitor(Map<String, String> environmentMetaData,
                               Map<String, String> monitorMetaData,
                               Reference<String> taskDetails);
}
```

监视器元数据中 IBM 保留值的常量如下:

```
package com.ibm.wmqfte.exitroutine.api;

/**
 * Constants for IBM reserved values placed into the monitor meta data
 * maps used by the monitor exit routines.
 */
public interface MonitorMetaDataConstants {

    /**
     * The value associated with this key is the name of the trigger
     * file associated with the monitor. Any modification performed
     * to this property by user exit routines will be ignored.
     */
    final String FILE_NAME_KEY = "FILENAME";

    /**
     * The value associated with this key is the path to the trigger
     * file associated with the monitor. Any modification performed
     * to this property by user exit routines will be ignored.
     */
    final String FILE_PATH_KEY = "FILEPATH";
}
```

```

/**
 * The value associated with this key is the size of the trigger
 * file associated with the monitor. This will not be present in
 * the cases where the size cannot be determined. Any modification
 * performed to this property by user exit routines will be ignored.
 */
final String FILE_SIZE_KEY = "FILESIZE";

/**
 * The value associated with this key is the local date on which
 * the trigger file associated with the monitor was last modified.
 * Any modification performed to this property by user exit routines
 * will be ignored.
 */
final String LAST_MODIFIED_DATE_KEY = "LASTMODIFIEDDATE";

/**
 * The value associated with this key is the local time at which
 * the trigger file associated with the monitor was last modified.
 * Any modification performed to this property by user exit routines
 * will be ignored.
 */
final String LAST_MODIFIED_TIME_KEY = "LASTMODIFIEDTIME";

/**
 * The value associated with this key is the UTC date on which
 * the trigger file associated with the monitor was last modified.
 * Any modification performed to this property by user exit routines
 * will be ignored.
 */
final String LAST_MODIFIED_DATE_KEY_UTC = "LASTMODIFIEDDATEUTC";

/**
 * The value associated with this key is the UTC time at which
 * the trigger file associated with the monitor was last modified.
 * Any modification performed to this property by user exit routines
 * will be ignored.
 */
final String LAST_MODIFIED_TIME_KEY_UTC = "LASTMODIFIEDTIMEUTC";

/**
 * The value associated with this key is the name of the agent on which
 * the monitor is running. Any modification performed to this property by
 * user exit routines will be ignored.
 */
final String MONITOR_AGENT_KEY = "AGENTNAME";
}

```

监视器用户出口示例

该示例类实现了 MonitorExit 接口。该示例将定制替换变量添加到名为 *REDIRECTEDAGENT* 的监视器元数据中，如果一天中的小时为奇数且偶数小时的值为 PARIS，那么此元数据将填充 LONDON 值。监视器出口结果代码设置为始终返回 *proceed*。

```

package com.ibm.wmqfte.monitor;

import java.util.Calendar;
import java.util.Map;

import com.ibm.wmqfte.exitroutine.api.MonitorExit;
import com.ibm.wmqfte.exitroutine.api.MonitorExitResult;
import com.ibm.wmqfte.exitroutine.api.Reference;

/**
 * Example resource monitor user exit that changes the monitor mutable
 * metadata value between 'LONDON' and 'PARIS' depending on the hour of the day.
 */
public class TestMonitorExit implements MonitorExit {

    // custom variable that will substitute destination agent
    final static String REDIRECTED_AGENT = "REDIRECTEDAGENT";

    public MonitorExitResult onMonitor(
        Map<String, String> environmentMetaData,
        Map<String, String> monitorMetaData,
        Reference<String> taskDetails) {

```

```

        // always succeed
        final MonitorExitResult result = MonitorExitResult.PROCEED_RESULT;

        final int hour = Calendar.getInstance().get(Calendar.HOUR_OF_DAY);

        if (hour%2 == 1) {
            monitorMetaData.put(REDIRECTED_AGENT, "LONDON");
        } else {
            monitorMetaData.put(REDIRECTED_AGENT, "PARIS");
        }

        return result;
    }
}

```

使用 *REDIRECTEDAGENT* 替换变量的监视器相应任务可能与以下内容类似：

```

<?xml version="1.0" encoding="UTF-8"?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT1"
      QMgr="QM1"/>
    <destinationAgent agent="{REDIRECTEDAGENT}"
      QMgr="QM2"/>
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="delete">
          <file>c:\sourcefiles\reports.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>c:\destinationfiles\reports.doc</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

此传输开始之前，<destinationAgent> 元素的 agent 属性的值将替换为 LONDON 或 PARIS。

必须以大写形式指定监视器出口类和任务定义 XML 中的替换变量。

相关概念

[第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』](#)

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

[第 971 页的『用户出口例程元数据』](#)

可向 IBM MQ Managed File Transfer 的用户出口例程提供三种不同类型的元数据：环境、传输和文件元数据。此元数据显示为 Java 键/值对的映射。

[第 979 页的『用户出口例程的 Java 接口』](#)

使用本节中的主题以获取有关用户出口例程的 Java 接口的参考信息。

相关参考

[第 977 页的『用户出口的代理属性』](#)

除 agent.properties 文件中的标准属性外，还存在多个特定于用户出口例程的高级属性。缺省情况下不包含这些属性，因此要使用它们中的任何一个，都必须手动编辑 agent.properties 文件。如果在代理运行期间对 agent.properties 文件进行了更改，请停止并重新启动该代理以提取这些更改。

用户出口的代理属性

除 agent.properties 文件中的标准属性外，还存在多个特定于用户出口例程的高级属性。缺省情况下不包含这些属性，因此要使用它们中的任何一个，都必须手动编辑 agent.properties 文件。如果在代理运行期间对 agent.properties 文件进行了更改，请停止并重新启动该代理以提取这些更改。

对于 IBM WebSphere MQ V7.5 或更高版本，可以在某些表示文件或目录位置的 Managed File Transfer 属性中使用环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化，例如哪个用户

在运行进程。有关更多信息，请参阅 [第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』](#)。

用户出口例程属性

将按照下表中列出的顺序调用用户出口例程。有关 `agent.properties` 文件的更多信息，请参阅 [高级代理程序属性: 用户出口例程](#)。

表 80: 用户出口的代理属性	
属性名	描述
<code>sourceTransferEndExitClasses</code>	指定一个实现源传输结束出口例程的以逗号分隔的类列表。
<code>sourceTransferStartExitClasses</code>	指定一个实现源传输开始出口例程的以逗号分隔的类列表。
<code>destinationTransferStartExitClasses</code>	指定一个实现目标传输开始用户出口例程的以逗号分隔的类列表。
<code>destinationTransferEndExitClasses</code>	指定一个实现目标传输最终用户出口例程的以逗号分隔的类列表。
<code>exitClassPath</code>	指定一个特定于平台、字符分隔的目录列表，充当用户出口例程的类路径。 在此类路径中的任何条目之前搜索代理程序出口目录。 如果在 Windows 上使用此属性，请使用正斜杠字符 (/) 作为路径定界符，而不是反斜杠字符 (\)。例如： <pre>exitClassPath=C:/mycomp/mqft/exits/encryptFileExit.jar; C:/mycomp/mqft/exits/fileFilter.jar.</pre> 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。 对于 IBM 4690 系统，此属性可以包含逻辑名，但目录必须位于 F: 驱动器上。
<code>exitNativeLibraryPath</code>	指定一个特定于平台、字符分隔的目录列表，充当用户出口例程的本机库路径。 对于 IBM WebSphere MQ V7.5 或更高版本，此属性的值可以包含环境变量。
<code>monitorExitClasses</code>	指定实现了监视器出口例程的类的逗号分隔列表。要了解更多信息，请参阅 第 974 页的『资源监视器用户出口』 。
<code>protocolBridgeCredentialExitClasses</code>	指定实现了协议网桥凭证用户出口例程的类的逗号分隔列表。要了解更多信息，请参阅 第 283 页的『使用出口类映射文件服务器的凭证』 。
<code>protocolBridgePropertiesExitClasses</code>	指定实现了协议网桥服务器属性用户出口例程的类的逗号分隔列表。 有关更多信息，请参阅 第 277 页的『使用出口类 (ProtocolBridgePropertiesExit2) 查找协议文件服务器属性』。
<code>IOExitClasses</code>	指定实现了 I/O 用户出口例程的类的逗号分隔列表。仅列出实现了 <code>IOExit</code> 接口的类，即不列出实现了其他 I/O 用户出口接口的类，例如， <code>IOExitResourcePath</code> 和 <code>IOExitChannel</code> 。有关更多信息，请参阅 第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』 。

出口调用的顺序

源和目标出口按照以下顺序调用：

1. `SourceTransferStartExit`
2. `DestinationTransferStartExit`
3. `DestinationTransferEndExit`
4. `SourceTransferEndExit`

更改源和目标出口

如果指定多个出口，那么将首先调用列表中的第一个出口，然后是第二个，依次类推。第一个出口所做的任何更改将作为随后调用的出口的输入，依次类推。例如，如果存在两个源传输启动出口，那么第一个出口对传输元数据所做的任何更改将作为第二个出口的输入。每个出口都会返回其自己的结果。如果给定类型的所有出口都返回 `PROCEED` 作为传输结果代码，那么总体结果将为 `PROCEED`。如果一个或多个出口返回

CANCEL_TRANSFER, 那么总体结果将为 CANCEL_TRANSFER。由出口返回的所有结果代码和字符串都将在传输日志中输出。

如果源传输启动出口的总体结果为 PROCEED, 那么传输将使用由出口所做的任何更改继续。如果总体结果为 CANCEL_TRANSFER, 那么将调用源传输结束出口, 然后取消该传输。传输日志中的完成状态将为“已取消”。

如果目标传输启动出口的总体结果为 PROCEED, 那么该传输将使用由出口所做的任何更改继续。如果总体结果为 CANCEL_TRANSFER, 那么将调用目标传输结束出口, 然后调用源传输结束出口。最后, 将取消该传输。传输日志中的完成状态将为“已取消”。

如果源或目标出口需要向链中或执行顺序中的后续出口传递信息, 那么必须通过更新传输元数据来完成。传输元数据的用途特定于出口实现。例如, 如果出口将返回结果设置为 CANCEL_TRANSFER, 并需要与已取消传输的后续出口进行通信, 那么必须通过以其他出口能够理解的方式设置传输元数据值来完成。

示例

```
sourceTransferStartExitClasses=com.ibm.wmqfte.test.MFTTestSourceTransferStartExit
sourceTransferEndExitClasses=com.ibm.wmqfte.test.MFTTestSourceTransferEndExit
destinationTransferStartExitClasses=com.ibm.wmqfte.test.MFTTestDestinationTransferStartExit
destinationTransferEndExitClasses=com.ibm.wmqfte.test.MFTTestDestinationTransferEndExit
exitClassPath=C:/mycomp/mqft/exits/encryptFileExit.jar;C:/mycomp/mqft/exits/fileFilter.jar
```

相关概念

[第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』](#)

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

[第 971 页的『用户出口例程元数据』](#)

可向 IBM MQ Managed File Transfer 的用户出口例程提供三种不同类型的元数据: 环境、传输和文件元数据。此元数据显示为 Java 键/值对的映射。

[第 979 页的『用户出口例程的 Java 接口』](#)

使用本节中的主题以获取有关用户出口例程的 Java 接口的参考信息。

相关参考

[第 974 页的『资源监视器用户出口』](#)

通过资源监视器用户出口, 您可以配置在满足监视器的触发条件的情况下, 要在关联任务启动之前运行的定制代码。

[第 580 页的『在 IBM MQ Managed File Transfer 属性中使用环境变量』](#)

从 IBM WebSphere MQ V7.5 起, 可以在受管文件传输属性中使用表示文件或目录位置的环境变量。这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化。例如, 哪个用户在运行进程。

[第 591 页的『agent.properties 文件』](#)

每个代理都有自己的属性文件 agent.properties, 其中必须包含代理用于连接到其队列管理器的信息。agent.properties 文件还可以包含用于更改代理程序行为的属性。

用户出口例程的 Java 接口

使用本节中的主题以获取有关用户出口例程的 Java 接口的参考信息。

相关概念

[第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』](#)

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

[第 983 页的『DestinationTransferStartExit.java 接口』](#)

[第 982 页的『DestinationTransferEndExit.java 接口』](#)

[第 985 页的『IOExit.java 接口』](#)

[第 989 页的『IOExitChannel.java 接口』](#)

[第 990 页的『IOExitLock.java 接口』](#)

[第 991 页的『IOExitPath.java 接口』](#)

[第 994 页的『IOExitProperties.java 接口』](#)

[第 996 页的『IOExitRecordChannel.java 接口』](#)

[第 998 页的『IOExitRecordResourcePath.java 接口』](#)

[第 1000 页的『IOExitResourcePath.java 接口』](#)

[第 1005 页的『IOExitWildcardPath.java 接口』](#)

[第 1006 页的『MonitorExit.java 接口』](#)

[第 1007 页的『ProtocolBridgeCredentialExit.java 接口』](#)

[第 1008 页的『ProtocolBridgeCredentialExit2.java 接口』](#)

[第 1009 页的『ProtocolBridgePropertiesExit2.java 接口』](#)

[第 1013 页的『SourceTransferStartExit.java 接口』](#)

[第 1012 页的『SourceTransferEndExit.java 接口』](#)

CDCredentialExit.java 接口

CDCredentialExit.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;

/**
 * An interface that is implemented by classes that are invoked as part of
 * user exit routine processing. This interface defines methods that are
 * invoked by a Connect:Direct bridge agent to map the WebSphere MQ user ID of the transfer to
 * credentials
 * that are used to access the Connect:Direct node.
 * There will be one instance of each implementation class per Connect:Direct bridge agent. The methods
 * can be called from different threads so the methods must be synchronized.
 */
public interface CDCredentialExit {

    /**
     * Invoked once when a Connect:Direct bridge agent is started. It is intended to initialize
     * any resources that are required by the exit
     *
     * @param bridgeProperties
     * The values of properties defined for the Connect:Direct bridge.
     * These values can only be read, they cannot be updated by
     * the implementation.
     *
     * @return
     * true if the initialisation is successful and false if unsuccessful
     * If false is returned from an exit the Connect:Direct bridge agent does not
     * start.
     */
    public boolean initialize(final Map<String, String> bridgeProperties);

    /**
     * Invoked once per transfer to map the WebSphere MQ user ID in the transfer message to the
     * credentials to be used to access the Connect:Direct node.
     *
     * @param mqUserId
     * The WebSphere MQ user ID from which to map to the credentials to be used
     * to access the Connect:Direct node
     *
     * @param snode
     * The name of the Connect:Direct SNODE specified as the cdNode in the
     * file path. This is used to map the correct user ID and password for the
     * SNODE.
     */
}
```



```

* @return      A credential exit result object that contains the result of the map and
*              the credentials to use to access the Connect:Direct node
*/

public CDCredentialExitResult mapMQUserId(final String mqUserId, final String snode);

/**
 * Invoked once when a Connect:Direct bridge agent is shutdown. This method releases
 * any resources that were allocated by the exit
 *
 * @param bridgeProperties
 *        The values of properties defined for the Connect:Direct bridge.
 *        These values can only be read, they cannot be updated by
 *        the implementation.
 *
 * @return
 */
public void shutdown(final Map<String, String> bridgeProperties);    }

```

CredentialExitResult.java 接口

CredentialExitResult.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */

package com.ibm.wmqfte.exitroutine.api;

/**
 * The result of invoking a Credential mapMQUserId exit method. It is composed of a result
 * code, which determines whether the mapping of the user id was successful, and an optional
 * Credentials object if the mapping is successful.
 */
public class CredentialExitResult {

    private final CredentialExitResultCode resultCode;
    private final Credentials credentials;

    /**
     * Constructor. Creates a credential exit result object with a specified result
     * code and optionally credentials.
     *
     * @param resultCode
     *        The result code to associate with the exit result being created.
     *
     * @param credentials
     *        The credentials to associate with the exit result being created.
     *        A value of <code>null</code> can be specified to indicate no
     *        credentials. If the resultCode is USER_SUCCESSFULLY_MAPPED the
     *        credentials must be set to a non-null value,
     */
    public CredentialExitResult(CredentialExitResultCode resultCode, Credentials credentials) {
        this.resultCode = resultCode;
        this.credentials = credentials;
    }

    /**
     * Returns the result code associated with this credential exit result
     *
     * @return    the result code associated with this exit result.
     */
    public CredentialExitResultCode getResultCode() {
        return resultCode;
    }
}

```

```

    * Returns the credentials associated with this credential exit result
    *
    * @return    the explanation associated with this credential exit result.
    */
    public Credentials getCredentials() {
        return credentials;
    }
}

```

相关参考

[第 1013 页的『SourceTransferStartExit.java 接口』](#)

[第 983 页的『DestinationTransferStartExit.java 接口』](#)

[第 982 页的『DestinationTransferEndExit.java 接口』](#)

[第 1006 页的『MonitorExit.java 接口』](#)

[第 1007 页的『ProtocolBridgeCredentialExit.java 接口』](#)

相关信息

[使用用户出口定制 MFT](#)

DestinationTransferEndExit.java 接口

DestinationTransferEndExit.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * Copyright IBM Corp. 2008, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitpoint.api;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately after completing a transfer on the agent acting as the
 * destination of the transfer.
 */
public interface DestinationTransferEndExit {

    /**
     * Invoked immediately after the completion of a transfer on the agent acting as
     * the destination of the transfer.
     *
     * @param transferExitResult
     *        a result object reflecting whether or not the transfer completed
     *        successfully.
     *
     * @param sourceAgentName
     *        the name of the agent acting as the source of the transfer.
     *
     * @param destinationAgentName
     *        the name of the agent acting as the destination of the
     *        transfer. This is the name of the agent that the
     *        implementation of this method will be invoked from.
     *
     * @param environmentMetaData
     *        meta data about the environment in which the implementation
     *        of this method is running. This information can only be read,
     *        it cannot be updated by the implementation. The constants
     *        defined in <code>EnvironmentMetaDataConstants</code> class can
     *        be used to access the data held by this map.
     *
     * @param transferMetaData
     *        meta data to associate with the transfer. The information can
     *        only be read, it cannot be updated by the implementation. This
     */
}

```

```

*          map may also contain keys with IBM reserved names. These
*          entries are defined in the <code>TransferMetaDataConstants</code>
*          class and have special semantics.
*
* @param fileResults
*          a list of file transfer result objects that describe the source
*          file name, destination file name and result of each file transfer
*          operation attempted.
*
* @return   an optional description to enter into the log message describing
*          transfer completion. A value of <code>null</code> can be used
*          when no description is required.
*/
String onDestinationTransferEnd(TransferExitResult transferExitResult,
                                String sourceAgentName,
                                String destinationAgentName,
                                Map<String, String>environmentMetaData,
                                Map<String, String>transferMetaData,
                                List<FileTransferResult>fileResults);
}

```

相关概念

[第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』](#)

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

[第 1013 页的『SourceTransferStartExit.java 接口』](#)

[第 1012 页的『SourceTransferEndExit.java 接口』](#)

[第 983 页的『DestinationTransferStartExit.java 接口』](#)

[第 1006 页的『MonitorExit.java 接口』](#)

[第 1007 页的『ProtocolBridgeCredentialExit.java 接口』](#)

DestinationTransferStartExit.java 接口

DestinationTransferStartExit.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitpoint.api;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately prior to starting a transfer on the agent acting as the
 * destination of the transfer.
 */
public interface DestinationTransferStartExit {

    /**
     * Invoked immediately prior to starting a transfer on the agent acting as
     * the destination of the transfer.
     *
     * @param sourceAgentName
     *          the name of the agent acting as the source of the transfer.
     *
     * @param destinationAgentName
     *          the name of the agent acting as the destination of the
     *          transfer. This is the name of the agent that the
     *          implementation of this method will be invoked from.
     *
     * @param environmentMetaData
     */
}

```

```

*      meta data about the environment in which the implementation
*      of this method is running. This information can only be read,
*      it cannot be updated by the implementation. The constants
*      defined in <code>EnvironmentMetaDataConstants</code> class can
*      be used to access the data held by this map.
*
* @param transferMetaData
*      meta data to associate with the transfer. The information can
*      only be read, it cannot be updated by the implementation. This
*      map may also contain keys with IBM reserved names. These
*      entries are defined in the <code>TransferMetaDataConstants</code>
*      class and have special semantics.
*
* @param fileSpecs
*      a list of file specifications that govern the file data to
*      transfer. The implementation of this method can modify the
*      entries in this list and the changes will be reflected in the
*      files transferred. However, new entries may not be added and
*      existing entries may not be removed.
*
* @return    a transfer exit result object which is used to determine if the
*            transfer should proceed, or be cancelled.
*/
TransferExitResult onDestinationTransferStart(String sourceAgentName,
                                             String destinationAgentName,
                                             Map<String, String> environmentMetaData,
                                             Map<String, String> transferMetaData,
                                             List<Reference<String>> fileSpecs);

```

相关概念

[第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』](#)

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

[第 1013 页的『SourceTransferStartExit.java 接口』](#)

[第 1012 页的『SourceTransferEndExit.java 接口』](#)

[第 982 页的『DestinationTransferEndExit.java 接口』](#)

[第 1006 页的『MonitorExit.java 接口』](#)

[第 1007 页的『ProtocolBridgeCredentialExit.java 接口』](#)

FileTransferResult.java 接口

FileTransferResult.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */

package com.ibm.wmqfte.exitroutine.api;

/**
 * Result information about a file transfer.
 */
public interface FileTransferResult {

    /** An enumeration for the <code>getCorrelatorType()</code> method. */
    public enum CorrelationInformationType {
        /** No correlation information is available for this result */
        NONE,
        /**
         * The correlation information relates to work done in
         * IBM Sterling File Gateway.
         */
    }
}

```

```

        */
        SFG
    }

    /**
     * Returns the source file specification, from which the file was transferred.
     *
     * @return the source file specification, from which the file was
     *         transferred.
     */
    String getSourceFileSpecification();

    /**
     * Returns the destination file specification, to which the file was transferred.
     *
     * @return the destination file specification, to which the file was
     *         transferred. A value of null may be returned
     *         if the transfer did not complete successfully.
     */
    String getDestinationFileSpecification();

    /**
     * Returns the result of the file transfer operation.
     *
     * @return the result of the file transfer operation.
     */
    FileExitResult getExitResult();

    /**
     * @return an enumerated value that identifies the product to which this correlating
     *         information relates.
     */
    CorrelationInformationType getCorrelatorType();

    /**
     * @return the first string component of the correlating identifier that relates
     *         this transfer result to work done in another product. A value of null
     *         may be returned either because the other product does not utilize a
     *         string based correlation information or because there is no correlation
     *         information.
     */
    String getString1Correlator();

    /**
     * @return the first long component of the correlating identifier that relates
     *         this transfer result to work done in another product. A value of zero
     *         is returned when there is no correlation information or the other
     *         product does not utilize long based correlation information or because
     *         the value really is zero!
     */
    long getLong1Correlator();
}

```

相关参考

[第 1013 页的『SourceTransferStartExit.java 接口』](#)

[第 983 页的『DestinationTransferStartExit.java 接口』](#)

[第 982 页的『DestinationTransferEndExit.java 接口』](#)

[第 1006 页的『MonitorExit.java 接口』](#)

[第 1007 页的『ProtocolBridgeCredentialExit.java 接口』](#)

相关信息

[使用用户出口定制 MFT](#)

[IOExit.java 接口](#)

IOExit.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 */

```

```

* 5724-H72
*
*   Copyright IBM Corp. 2011, 2023. All Rights Reserved.
*
*   US Government Users Restricted Rights - Use, duplication or
*   disclosure restricted by GSA ADP Schedule Contract with
*   IBM Corp.
*/
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;
import java.util.Map;

import com.ibm.wmqfte.exitroutine.api.IOExitRecordResourcePath.RecordFormat;

/**
 * An interface that is implemented by classes that you want to be invoked as
 * part of user exit routine processing. This interface defines methods that
 * will be invoked during transfers to perform the underlying file system I/O
 * work for WMQFTE transfers.
 * <p>
 * The {@link #initialize(Map)} method will be called once when the exit is
 * first installed. The WMQFTE agent properties are passed to this method, thus
 * enabling the exit to understand its environment.
 * <p>
 * The {@link #isSupported(String)} method will be invoked during WMQFTE
 * transfers to determine whether the user exit should be used. If the
 * {@link #isSupported(String)} method returns a value of {@code true}, the
 * {@link #newPath(String)} method will be invoked for the paths specified for
 * the transfer request. The returned {@link IOExitPath} instance from a
 * {@link #newPath(String)} method invocation will then be used by the WMQFTE
 * transfer to obtain information about the resource and to transfer data to or
 * from the resource.
 * <p>
 * To obtain transfer context for an I/O exit, a {@link SourceTransferStartExit}
 * or {@link DestinationTransferStartExit} as appropriate, should be installed
 * to enable information to be seen by this exit. The
 * {@link SourceTransferStartExit} or {@link DestinationTransferStartExit} are
 * passed the transfer's environment, metadata, and a list of file
 * specifications for the transfer. The paths for the file specifications are
 * the paths passed to the I/O exit's {@link #newPath(String)} method.
 * <p>
 * Note also that the {@link #isSupported(String)} and {@link #newPath(String)}
 * methods might be called at other times by a WMQFTE agent and not just during
 * transfers. For example, at transfer setup time the I/O system is queried to
 * resolve the full resource paths for transfer.
 */
public interface IOExit {

    /**
     * Invoked once when the I/O exit is first required for use. It is intended
     * to initialize any resources that are required by the exit.
     *
     * @param agentProperties
     *     The values of properties defined for the WMQFTE agent. These
     *     values can only be read, they cannot be updated by the
     *     implementation.
     * @return {@code true} if the initialization is successful and {@code
     *     false} if unsuccessful. If {@code false} is returned from an
     *     exit, the exit will not be used.
     */
    boolean initialize(final Map<String, String> agentProperties);

    /**
     * Indicates whether this I/O user exit supports the specified path.
     * <p>
     * This method is used by WMQFTE to determine whether the I/O user exit
     * should be used within a transfer. If no I/O user exit returns true for
     * this method, the default WMQFTE file I/O function will be used.
     *
     * @param path
     *     The path to the required I/O resource.
     * @return {@code true} if the specified path is supported by the I/O exit,
     *     {@code false} otherwise
     */
    boolean isSupported(String path);

    /**
     * Obtains a new {@link IOExitPath} instance for the specified I/O resource
     * path.
     * <p>
     * This method will be invoked by WMQFTE only if the

```

```

* {@link #isSupported(String)} method has been called for the path and
* returned {@code true}.
*
* @param path
*     The path to the required I/O resource.
* @return A {@link IOExitPath} instance for the specified path.
* @throws IOException
*     If the path cannot be created for any reason.
*/
IOExitPath newPath(String path) throws IOException;

/**
* Obtains a new {@link IOExitPath} instance for the specified I/O resource
* path and passes record format and length information required by the
* WMQFTE transfer.
* <p>
* Typically this method will be called for the following cases:
* <ul>
* <li>A path where a call to {@link #newPath(String)} has previously
* returned a {@link IOExitRecordResourcePath} instance and WMQFTE is
* re-establishing a new {@link IOExitPath} instance for the path, from an
* internally-serialized state. The passed recordFormat and recordLength
* will be the same as those for the original
* {@link IOExitRecordResourcePath} instance.</li>
* <li>A transfer destination path where the source of the transfer is
* record oriented. The passed recordFormat and recordLength will be the
* same as those for the source.</li>
* </ul>
* The implementation can act on the record format and length information as
* deemed appropriate. For example, for a destination agent if the
* destination does not already exist and the source of the transfer is
* record oriented, the passed recordFormat and recordLength information
* could be used to create an appropriate record-oriented destination path.
* If the destination path already exists, the passed recordFormat and
* recordLength information could be used to perform a compatibility check
* and throw an {@link IOException} if the path is not compatible. A
* compatibility check could ensure that a record oriented path's record
* format is the same as the passed record format or that the record length
* is greater or equal to the passed record length.
* <p>
* This method will be invoked by WMQFTE only if the
* {@link #isSupported(String)} method has been called for the path and
* returned {@code true}.
*
* @param path
*     The path to the required I/O resource.
* @param recordFormat
*     The advised record format.
* @param recordLength
*     The advised record length.
* @return A {@link IOExitPath} instance for the specified path.
* @throws IOException
*     If the path cannot be created for any reason. For example,
*     the passed record format or length is incompatible with the
*     path's actual record format or length.
*/
IOExitPath newPath(String path, RecordFormat recordFormat, int recordLength)
    throws IOException;

```

相关概念

第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的底层文件系统 I/O 工作。

IOExit2.java 接口

IOExit2.java

```

/*
* Licensed Materials - Property of IBM
*

```



```

* "Restricted Materials of IBM"
*
* 5724-H72
*
* Copyright IBM Corp. 2012, 2023. All Rights Reserved.
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*/
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;
import java.util.List;

import com.ibm.wmqfte.exitroutine.api.IOExitRecordResourcePath.RecordFormat;

public interface IOExit2 extends IOExit {

    /**
     * An extension to the {@link IOExit#newPath(String)} which
     * allows path attributes to be specified, for use when
     * creating or updating a path.
     *
     * @param path as per {@link IOExit#newPath(String)}
     *
     * @param attributes a list of path attributes which the
     *                    exit can choose to apply to file system
     *                    objects that are created, updated
     *                    or read using this path.
     *
     * @return as per {@link IOExit#newPath(String)}
     *
     * @throws IOException as per {@link IOExit#newPath(String)}.
     *                    Can also be thrown if the
     *                    <code>attributes</code> parameter
     *                    contains a
     *                    <code>IOExitPathAttribute</code> which
     *                    the exit implementation does not
     *                    understand.
     */
    IOExitPath newPath(String path,
                       List<IOExitPathAttribute> attributes)
        throws IOException;

    /**
     * An extension to the
     * {@link IOExit#newPath(String, RecordFormat, int)} which
     * allows path attributes to be specified, for use when
     * creating or updating a path.
     *
     * @param path as per {@link IOExit#newPath(String)}
     *
     * @param attributes a list of path attributes which the exit
     *                    can choose to apply to file system
     *                    objects that are created, updated
     *                    or read using this path.
     *
     * @param recordFormat as per {@link IOExit#newPath(String)}
     *
     * @param recordLength as per {@link IOExit#newPath(String)}
     *
     * @return as per {@link IOExit#newPath(String)}
     *
     * @throws IOException as per {@link IOExit#newPath(String)}.
     *                    Can also be thrown if the
     *                    <code>attributes</code> parameter
     *                    contains a
     *                    <code>IOExitPathAttribute</code>
     *                    which the exit implementation
     *                    does not understand.
     */
    IOExitPath newPath(String path,
                       List<IOExitPathAttribute> attributes,
                       RecordFormat recordFormat,
                       int recordLength)
        throws IOException;
}

```

相关概念

第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的底层文件系统 I/O 工作。

IOExitChannel.java 接口

IOExitChannel.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;
import java.nio.ByteBuffer;

/**
 * Represents a channel that enables data to be read from or written to an
 * {@link IOExitResourcePath} resource.
 */
public interface IOExitChannel {

    /**
     * Obtains the data size for the associated {@link IOExitResourcePath} in
     * bytes.
     *
     * @return The data size in bytes.
     * @throws IOException If a problem occurs while attempting obtain the size.
     */
    long size() throws IOException;

    /**
     * Closes the channel, flushing any buffered write data to the resource and
     * releasing any locks.
     *
     * @throws RecoverableIOException If a recoverable problem occurs while closing the resource.
     * This means that WMQFTE can attempt to recover the transfer.
     * @throws IOException If some other I/O problem occurs. For example, the channel might
     already be closed.
     */
    void close() throws RecoverableIOException, IOException;

    /**
     * Reads data from this channel into the given buffer, starting at this
     * channel's current position, and updates the current position by the
     * amount of data read.
     *
     * <p>
     * Data is copied into the buffer starting at its current position and up to
     * its limit. On return, the buffer's position is updated to reflect the
     * number of bytes read.
     *
     * @param buffer The buffer that the data is to be copied into.
     * @return The number of bytes read, which might be zero, or -1 if the end of
     data has been reached.
     * @throws RecoverableIOException If a recoverable problem occurs while reading the data. For a
     WMQFTE transfer this means that it will attempt to recover.
     */
}
```

```

    * @throws IOException
    *         If some other I/O problem occurs. For a WMQFTE transfer this
    *         means that it will be failed.
    */
    int read(ByteBuffer buffer) throws RecoverableIOException, IOException;

    /**
     * Writes data to this channel from the given buffer, starting at this
     * channel's current position, and updates the current position by the
     * amount of data written. The channel's resource is grown to accommodate
     * the data, if necessary.
     * <p>
     * Data is copied from the buffer starting at its current position and up to
     * its limit. On return, the buffer's position is updated to reflect the
     * number of bytes written.
     *
     * @param buffer
     *        The buffer containing the data to be written.
     * @return The number of bytes written, which might be zero.
     * @throws RecoverableIOException
     *        If a recoverable problem occurs while writing the data. For a
     *        WMQFTE transfer this means that it will attempt to recover.
     * @throws IOException
     *        If some other I/O problem occurs. For a WMQFTE transfer this
     *        means that it will be failed.
     */
    int write(ByteBuffer buffer) throws RecoverableIOException, IOException;

    /**
     * Forces any updates to this channel's resource to be written to its
     * storage device.
     * <p>
     * This method is required to force changes to both the resource's content
     * and any associated metadata to be written to storage.
     *
     * @throws RecoverableIOException
     *        If a recoverable problem occurs while performing the force.
     *        For a WMQFTE transfer this means that it will attempt to
     *        recover.
     * @throws IOException
     *        If some other I/O problem occurs. For a WMQFTE transfer this
     *        means that it will be failed.
     */
    void force() throws RecoverableIOException, IOException;

    /**
     * Attempts to lock the entire resource associated with the channel for
     * shared or exclusive access.
     * <p>
     * The intention is for this method not to block if the lock is currently
     * unavailable.
     *
     * @param shared
     *        {@code true} if a shared lock is required, {@code false} if an
     *        exclusive lock is required.
     * @return A {@link IOExitLock} instance representing the newly acquired
     *        lock or null if the lock cannot be obtained.
     * @throws IOException
     *        If a problem occurs while attempting to acquire the lock.
     */
    IOExitLock tryLock(boolean shared) throws IOException;
}

```

相关概念

[第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』](#)

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

[第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』](#)

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的的底层文件系统 I/O 工作。

IOExitLock.java 接口

IOExitLock.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;

/**
 * Represents a lock on a resource for either shared or exclusive access.
 * {@link IOExitLock} instances are returned from
 * {@link IOExitChannel#tryLock(boolean)} calls and WMQFTE will request the
 * release of the lock at the appropriate time during a transfer. Additionally, when
 * a {@link IOExitChannel#close()} method is called it will be the
 * responsibility of the channel to release any associated locks.
 */
public interface IOExitLock {

    /**
     * Releases the lock.
     * <p>
     * After this method has been successfully called the lock is to be deemed as invalid.
     *
     * @throws IOException
     *         If the channel associated with the lock is not open or
     *         another problem occurs while attempting to release the lock.
     */
    void release() throws IOException;

    /**
     * Indicates whether this lock is valid.
     * <p>
     * A lock is considered valid until its @ {@link #release()} method is
     * called or the associated {@link IOExitChannel} is closed.
     *
     * @return {@code true} if this lock is valid, {@code false} otherwise.
     */
    boolean isValid();

    /**
     * @return {@code true} if this lock is for shared access, {@code false} if
     *         this lock is for exclusive access.
     */
    boolean isShared();
}
```

相关概念

[第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』](#)

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

[第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』](#)

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的的底层文件系统 I/O 工作。

IOExitPath.java 接口

IOExitPath.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
```

```

*
* 5724-H72
*
* Copyright IBM Corp. 2011, 2023. All Rights Reserved.
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*/
package com.ibm.wmqfte.exitroutine.api;

/**
 * Represents an abstract path that can be inspected and queried by WMQFTE for
 * transfer purposes.
 * <p>
 * There are two types of path supported:
 * <ul>
 * <li>{@link IOExitResourcePath} - Represents a path that denotes a data
 * resource. For example, a file, directory, or group of database records.</li>
 * <li>{@link IOExitWildcardPath} - Represents a wildcard path that can be
 * expanded to multiple {@link IOExitResourcePath} instances.</li>
 * </ul>
 */
public abstract interface IOExitPath {

    /**
     * Obtains the abstract path as a {@link String}.
     *
     * @return The abstract path as a {@link String}.
     */
    String getPath();

    /**
     * Obtains the name portion of this abstract path as a {@link String}.
     * <p>
     * For example, a UNIX-style file system implementation evaluates the
     * path {@code /home/fteuser/file1.txt} as having a name of {@code
     * file1.txt}.
     *
     * @return the name portion of this abstract path as a {@link String}.
     */
    String getName();

    /**
     * Obtains the parent path for this abstract path as a {@link String}.
     * <p>
     * For example, a UNIX-style file system implementation evaluates the
     * path {@code /home/fteuser/file1.txt} as having a parent path of {@code
     * /home/fteuser}.
     *
     * @return The parent portion of the path as a {@link String}.
     */
    String getParent();

    /**
     * Obtains the abstract paths that match this abstract path.
     * <p>
     * If this abstract path denotes a directory resource, a list of paths
     * for all resources within the directory are returned.
     * <p>
     * If this abstract path denotes a wildcard, a list of all paths
     * matching the wildcard are returned.
     * <p>
     * Otherwise null is returned, because this abstract path probably denotes a
     * single file resource.
     *
     * @return An array of {@link IOExitResourcePath}s that
     *         match this path, or null if this method is not applicable.
     */
    IOExitResourcePath[] listPaths();
}

```

相关概念

第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的的底层文件系统 I/O 工作。

IOExitPathAttribute.java 接口

IOExitPathAttribute.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

/**
 * Represents an attribute associated with an IOExit path.
 * The exit can choose to apply attributes to file system objects
 *
 */
public class IOExitPathAttribute {

    private final String name;
    private final String value;

    /**
     * Constructor for an attribute with a name but no value
     * @param name
     */
    public IOExitPathAttribute(final String name) {
        this.name = name;
        this.value = null;
    }

    /**
     * Constructor for an attribute with a name and value
     * @param name The name of the attribute
     * @param value The value of the attribute
     */
    public IOExitPathAttribute(final String name,
                               final String value) {
        this.name = name;
        this.value = value;
    }

    public boolean hasValue() {
        return value != null;
    }

    public String getName() {
        return name;
    }

    public String getValue() {
        return value;
    }
}
```

相关概念

第 358 页的『[使用用户出口例程定制 IBM MQ Managed File Transfer](#)』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 361 页的『[使用 IBM MQ Managed File Transfer 传输 I/O 用户出口](#)』

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的的底层文件系统 I/O 工作。

IOExitProperties.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

/**
 * Properties that determine how WMQFTE treats an {@link IOExitPath} for certain
 * aspects of I/O. For example, whether to use intermediate files.
 */
public class IOExitProperties {

    private boolean rereadSourceOnRestart = true;
    private boolean rechecksumSourceOnRestart = true;
    private boolean rechecksumDestinationOnRestart = true;
    private boolean useIntermediateFileAtDestination = true;
    private boolean requiresSingleThreadedChannelIO = false;

    /**
     * Determines whether the I/O exit implementation expects the resource to be
     * re-read from the start if a transfer is restarted.
     *
     * @return {@code true} if, on restart, the I/O exit expects the source
     *         resource to be opened at the beginning and re-read from the
     *         beginning (the {@link IOExitPath#openForRead(long)} method is
     *         always invoked with 0L as an argument). {@code false} if, on
     *         restart, the I/O exit expects the source to be opened at the
     *         offset that the source agent intends to start reading from (the
     *         {@link IOExitPath#openForRead(long)} method can be invoked with a
     *         non-zero value as its argument).
     */
    public boolean getRereadSourceOnRestart() {
        return rereadSourceOnRestart;
    }

    /**
     * Sets the value to determine whether the I/O exit implementation expects
     * the resource to be re-read from the beginning if a transfer is restarted.
     * <p>
     * The default is {@code true}. The I/O exit should call this method when
     * required to change this value.
     *
     * @param rereadSourceOnRestart
     *        {@code true} if, on restart, the I/O exit expects the source
     *        resource to be opened at the beginning and re-read from the
     *        beginning (the {@link IOExitPath#openForRead(long)} method
     *        is always invoked with 0L as an argument). {@code false}
     *        if, on restart, the I/O exit expects the source to be opened
     *        at the offset that the source agent intends to start reading
     *        from (the {@link IOExitPath#openForRead(long)} method can be
     *        invoked with a non-zero value as its argument).
     */
    public void setRereadSourceOnRestart(boolean rereadSourceOnRestart) {
        this.rereadSourceOnRestart = rereadSourceOnRestart;
    }

    /**
     * Determines whether the I/O exit implementation requires the source
     * resource to be re-checksummed if the transfer is restarted.
     * Re-checksumming takes place only if the
     * {@link #getRereadSourceOnRestart()} method returns {@code true}.
     *
     * @return {@code true} if, on restart, the I/O exit expects the already-
     *         transferred portion of the source to be re-checksummed for
     *         inconsistencies. Use this option in environments
     *         where the source could be changed during a restart. {@code

```



```

*         false} if, on restart, the I/O exit does not require the
*         already-transferred portion of the source to be re-checksummed.
*/
public boolean getRechecksumSourceOnRestart() {
    return rechecksumSourceOnRestart;
}

/**
 * Sets the value to determine whether the I/O exit implementation requires
 * the source resource to be re-checksummed if the transfer is restarted.
 * Re-checksumming takes place only if the
 * {@link #getRereadSourceOnRestart()} method returns {@code true}.
 * <p>
 * The default is {@code true}. The I/O exit should call this method when
 * required to change this value.
 *
 * @param rechecksumSourceOnRestart
 *         {@code true} if, on restart, the I/O exit expects the already
 *         transferred portion of the source to be re-checksummed
 *         for inconsistencies. Use this option in environments
 *         where the source could be changed during a restart.
 *         {@code false} if, on restart, the I/O exit does not
 *         require the already-transferred portion of the source to be
 *         re-checksummed.
 */
public void setRechecksumSourceOnRestart(boolean rechecksumSourceOnRestart) {
    this.rechecksumSourceOnRestart = rechecksumSourceOnRestart;
}

/**
 * Determines whether the I/O exit implementation requires the destination
 * resource to be re-checksummed if the transfer is restarted.
 *
 * @return {@code true} if, on restart, the I/O exit expects the already
 *         transferred portion of the destination to be re-checksummed to
 *         check for inconsistencies. This option should be used in
 *         environments where the destination could have been changed while
 *         a restart is occurring. {@code false} if, on restart, the I/O exit
 *         does not require the already transferred portion of the
 *         destination to be re-checksummed.
 */
public boolean getRechecksumDestinationOnRestart() {
    return rechecksumDestinationOnRestart;
}

/**
 * Sets the value to determine whether the I/O exit implementation requires
 * the destination resource to be re-checksummed if the transfer is
 * restarted.
 * <p>
 * The default is {@code true}. The I/O exit should call this method when
 * required to change this value.
 *
 * @param rechecksumDestinationOnRestart
 *         {@code true} if, on restart, the I/O exit expects the already-
 *         transferred portion of the destination to be re-checksummed
 *         for inconsistencies. Use this option in environments
 *         where the destination could have been changed during a
 *         restart. {@code false} if, on restart, the I/O exit does not
 *         require the already-transferred portion of the destination
 *         to be re-checksummed.
 */
public void setRechecksumDestinationOnRestart(
    boolean rechecksumDestinationOnRestart) {
    this.rechecksumDestinationOnRestart = rechecksumDestinationOnRestart;
}

/**
 * Determines whether the I/O exit implementation requires the use of an
 * intermediate file when writing the data at the destination. The
 * intermediate file mechanism is typically used to prevent an incomplete
 * destination resource from being processed.
 *
 * @return {@code true} if data should be written to an intermediate file at
 *         the destination and then renamed (to the requested destination
 *         path name as specified in the transfer request) after the transfer is
 *         complete. {@code false} if data should be written directly to the
 *         requested destination path name without the use of an
 *         intermediate file.
 */
public boolean getUseIntermediateFileAtDestination() {
    return useIntermediateFileAtDestination;
}

```

```

}

/**
 * Sets the value to determine whether the I/O exit implementation requires
 * the use of an intermediate file when writing the data at the destination.
 * The intermediate file mechanism is typically used to prevent an
 * incomplete destination resource from being processed.
 *
 * <p>
 * The default is {@code true}. The I/O exit should call this method when
 * required to change this value.
 *
 * @param useIntermediateFileAtDestination
 *        {@code true} if data should be written to an intermediate file
 *        at the destination and then renamed (to the requested
 *        destination path name as specified in the transfer request) after
 *        the transfer is complete. {@code false} if data should be written
 *        directly to the requested destination path name without the
 *        use of an intermediate file
 */
public void setUseIntermediateFileAtDestination(
    boolean useIntermediateFileAtDestination) {
    this.useIntermediateFileAtDestination = useIntermediateFileAtDestination;
}

/**
 * Determines whether the I/O exit implementation requires
 * {@link IOExitChannel} instances to be accessed by a single thread only.
 *
 * @return {@code true} if {@link IOExitChannel} instances are to be
 *         accessed by a single thread only.
 */
public boolean requiresSingleThreadedChannelIO() {
    return requiresSingleThreadedChannelIO;
}

/**
 * Sets the value to determine whether the I/O exit implementation requires
 * channel operations for a particular instance to be accessed by a
 * single thread only.
 *
 * <p>
 * For certain I/O implementations it is necessary that resource path
 * operations such as open, read, write, and close are invoked only from a
 * single execution {@link Thread}. When set {@code true}, WMQFTE ensures
 * that the following are invoked on a single thread:
 *
 * <ul>
 * <li>{@link IOExitResourcePath#openForRead(long)} method and all methods of
 * the returned {@link IOExitChannel} instance.</li>
 * <li>{@link IOExitResourcePath#openForWrite(boolean)} method and all
 * methods of the returned {@link IOExitChannel} instance.</li>
 * </ul>
 *
 * <p>
 * This has a slight performance impact, hence enable single-threaded channel
 * I/O only when absolutely necessary.
 *
 * <p>
 * The default is {@code false}. The I/O exit should call this method when
 * required to change this value.
 *
 * @param requiresSingleThreadedChannelIO
 *        {@code true} if {@link IOExitChannel} instances are to be
 *        accessed by a single thread only.
 */
public void setRequiresSingleThreadedChannelIO(boolean requiresSingleThreadedChannelIO) {
    this.requiresSingleThreadedChannelIO = requiresSingleThreadedChannelIO;
}
}

```

相关概念

第 358 页的『[使用用户出口例程定制 IBM MQ Managed File Transfer](#)』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 361 页的『[使用 IBM MQ Managed File Transfer 传输 I/O 用户出口](#)』

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的底层文件系统 I/O 工作。

IOExitRecordChannel.java 接口

IOExitRecordChannel.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;
import java.nio.ByteBuffer;

/**
 * Represents a channel that enables records of data to be read from or written
 * to an {@link IOExitRecordResourcePath} resource.
 * <p>
 * This is an extension of the {@link IOExitChannel} interface such that the
 * {@link #read(java.nio.ByteBuffer)} and {@link #write(java.nio.ByteBuffer)}
 * methods are expected to deal in whole records of data only. That is, the
 * {@link java.nio.ByteBuffer} returned from the read method and passed to the
 * write method is assumed to contain one or more complete records.
 */
public interface IOExitRecordChannel extends IOExitChannel {

    /**
     * Reads records from this channel into the given buffer, starting at this
     * channel's current position, and updates the current position by the
     * amount of data read.
     * <p>
     * Record data is copied into the buffer starting at its current position
     * and up to its limit. On return, the buffer's position is updated to
     * reflect the number of bytes read.
     * <p>
     * Only whole records are copied into the buffer.
     * <p>
     * For a fixed-record-format resource, this might be multiple records. The
     * amount of data in the return buffer does not necessarily need to be a
     * multiple of the record length, but the last record is still to be treated
     * as a complete record and padded as required by the caller.
     * <p>
     * For a variable-format resource, this is a single whole record of a size
     * corresponding to the amount of return data or multiple whole records with
     * all except the last being treated as records of maximum size.
     *
     * @param buffer
     *         The buffer that the record data is to be copied into.
     * @return The number of bytes read, which might be zero, or -1 if the end of
     *         data has been reached.
     * @throws RecoverableIOException
     *         If a recoverable problem occurs while reading the data. For a
     *         WMQFTE transfer this means that it will attempt to recover.
     * @throws IOException
     *         If some other I/O problem occurs, for example, if the passed
     *         buffer is insufficient to contain at least one complete
     *         record). For a WMQFTE transfer this means that it will be
     *         failed.
     */
    int read(ByteBuffer buffer) throws RecoverableIOException, IOException;

    /**
     * Writes records to this channel from the given buffer, starting at this
     * channel's current position, and updates the current position by the
     * amount of data written. The channel's resource is grown to accommodate
     * the data, if necessary.
     * <p>
     * Record data is copied from the buffer starting at its current position
     * and up to its limit. On return, the buffer's position is updated to
     * reflect the number of bytes written.
     * <p>
     * The buffer is expected to contain only whole records.
     * <p>
     * For a fixed-record-format resource, this might be multiple records and if

```

```

* there is insufficient data in the buffer for a complete record, the
* record is to be padded as required to complete the record.
* <p>
* For a variable-record format resource the buffer is normally expected to
* contain a single record of length corresponding to the amount of data
* within the buffer. However, if the amount of data within the buffer
* exceeds the maximum record length, the implementation can either:
* <ol>
* <li>throw an {@link IOException} indicating that it cannot handle the
* situation.</li>
* <li>Consume a record's worth of data from the buffer, leaving the remaining
* data within the buffer.</li>
* <li>Consume all the buffer data and just write what it can to the current
* record. This effectively truncates the data.</li>
* <li>Consume all the buffer data and write to multiple records.</li>
* </ol>
*
* @param buffer
*         The buffer containing the data to be written.
* @return The number of bytes written, which might be zero.
* @throws RecoverableIOException
*         If a recoverable problem occurs while writing the data. For a
*         WMQFTE transfer this means that it will attempt to recover.
* @throws IOException
*         If some other I/O problem occurs. For a WMQFTE transfer this
*         means that it will be failed.
*/
int write(ByteBuffer buffer) throws RecoverableIOException, IOException;
}

```

相关概念

第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的底层文件系统 I/O 工作。

IOExitRecordResourcePath.java 接口

IOExitRecordResourcePath.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;

/**
 * Represents a path that denotes a record-oriented data resource (for example,
 * a z/OS data set). It allows the data to be located, the record format to be
 * understood, and {@link IOExitRecordChannel} instances to be created for read
 * or write operations.
 */
public interface IOExitRecordResourcePath extends IOExitResourcePath {

    /**
     * Record formats for record-oriented resources.
     */
    public enum RecordFormat {
        FIXED, VARIABLE
    }
}

```

```

/**
 * Obtains the record length for records that are maintained by the resource
 * denoted by this abstract path.
 * <p>
 * For a resource with fixed-length records, the data for each record read
 * and written is assumed to be this length.
 * <p>
 * For a resource with variable-length records, this is the maximum length
 * for a record's data.
 * <p>
 * This method should return a value greater than zero, otherwise it can
 * result in the failure of a WMQFTE transfer that involves this abstract
 * path.
 *
 * @return The record length, in bytes, for records maintained by the
 *         resource.
 */
int getRecordLength();

/**
 * Obtains record format, as a {@link RecordFormat} instance, for records
 * that are maintained by the resource denoted by this abstract path.
 *
 * @return A {@link RecordFormat} instance for the record format for records
 *         that are maintained by the resource denoted by this abstract
 *         path.
 */
RecordFormat getRecordFormat();

/**
 * Opens a {@link IOExitRecordChannel} instance for reading data from the
 * resource denoted by this abstract path. The current data byte position
 * for the resource is expected to be the passed position value, such that
 * when {@link IOExitRecordChannel#read(java.nio.ByteBuffer)} is called,
 * data starting from that position is read.
 * <p>
 * Note that the data byte read position will be on a record boundary.
 *
 * @param position
 *         The required data byte read position.
 * @return A new {@link IOExitRecordChannel} instance allowing data to be
 *         read from the resource denoted by this abstract path.
 * @throws RecoverableIOException
 *         If a recoverable problem occurs while attempting to open the
 *         resource for reading. This means that WMQFTE can attempt to
 *         recover the transfer.
 * @throws IOException
 *         If some other I/O problem occurs.
 */
IOExitRecordChannel openForRead(long position)
    throws RecoverableIOException, IOException;

/**
 * Opens a {@link IOExitRecordChannel} instance for writing data to the
 * resource denoted by this abstract path. Writing of data, using the
 * {@link IOExitRecordChannel#write(java.nio.ByteBuffer)} method, starts at
 * either the beginning of the resource or end of the current data for the
 * resource, depending on the specified append parameter.
 *
 * @param append
 *         When {@code true} indicates that data written to the resource
 *         should be appended to the end of the current data. When
 *         {@code false} indicates that writing of data is to start at
 *         the beginning of the resource; any existing data is lost.
 * @return A new {@link IOExitRecordChannel} instance allowing data to be
 *         written to the resource denoted by this abstract path.
 * @throws RecoverableIOException
 *         If a recoverable problem occurs while attempting to open the
 *         resource for writing. This means that WMQFTE can attempt to
 *         recover the transfer.
 * @throws IOException
 *         If some other I/O problem occurs.
 */
IOExitRecordChannel openForWrite(boolean append)
    throws RecoverableIOException, IOException;
}

```

相关概念

[第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』](#)

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

[第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』](#)

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的底层文件系统 I/O 工作。

IOExitRecordResourcePath2.java 接口

IOExitRecordResourcePath2.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

public interface IOExitRecordResourcePath2
extends IOExitResourcePath2, IOExitRecordResourcePath {
}
```

相关概念

[第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』](#)

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

[第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』](#)

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的底层文件系统 I/O 工作。

IOExitResourcePath.java 接口

IOExitResourcePath.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;

/**
 * Represents a path that denotes a data resource (for example, a file,
 * directory, or group of database records). It allows the data to be located
 * and {@link IOExitChannel} instances to be created for read or write
 * operations.
 * <p>
 * There are two types of data resources as follows:
 * <ul>
 * <li>Directory - a container for other data resources. The
```

```

* {@link #isDirectory()} method returns {@code true} for these.</li>
* <li>File - a data container. This allows data to be read from or written to
* it. The {@link #isFile()} method returns {@code true} for these.</li>
* </ul>
*/
public interface IOExitResourcePath extends IOExitPath {

    /**
     * Creates a new {@link IOExitResourcePath} instance for a child path of the
     * resource denoted by this abstract path.
     * <p>
     * For example, with a UNIX-style path, {@code
     * IOExitResourcePath("/home/fteuser/test").newPath("subtest")} could be
     * equivalent to: {@code IOExitResourcePath("/home/fteuser/test/subtest")}
     * </p>
     * @param child
     *     The child path name.
     * @return A new {@link IOExitResourcePath} instance that represents a child
     *     of this path.
     */
    IOExitResourcePath newPath(final String child);

    /**
     * Creates the directory path for the resource denoted by this abstract
     * path, including any necessary but nonexistent parent directories. If the
     * directory path already exists, this method has no effect.
     * <p>
     * If this operation fails, it might have succeeded in creating some of the
     * necessary parent directories.
     * </p>
     * @throws IOException
     *     If the directory path cannot be fully created, when it does
     *     not already exist.
     */
    void makePath() throws IOException;

    /**
     * Obtains the canonical path of the abstract path as a {@link String}.
     * <p>
     * A canonical path is defined as being absolute and unique. For example,
     * the path can be represented as UNIX-style relative path: {@code
     * test/file.txt} but the absolute and unique canonical path representation
     * is: {@code /home/fteuser/test/file.txt}
     * </p>
     * @return The canonical path as a {@link String}.
     * @throws IOException
     *     If the canonical path cannot be determined for any reason.
     */
    String getCanonicalPath() throws IOException;

    /**
     * Tests if this abstract path is an absolute path.
     * <p>
     * For example, a UNIX-style path, {@code /home/fteuser/test} is an absolute
     * path, whereas {@code fteuser/test} is not.
     * </p>
     * @return {@code true} if this abstract path is an absolute path, {@code
     *     false} otherwise.
     */
    boolean isAbsolute();

    /**
     * Tests if the resource denoted by this abstract path exists.
     * @return {@code true} if the resource denoted by this abstract path
     *     exists, {@code false} otherwise.
     * @throws IOException
     *     If the existence of the resource cannot be determined for any
     *     reason.
     */
    boolean exists() throws IOException;

    /**
     * Tests whether the calling application can read the resource denoted by
     * this abstract path.
     * @return {@code true} if the resource for this path exists and can be
     *     read, {@code false} otherwise.
     * @throws IOException
     *     If a problem occurs while attempting to determine if the
     *     resource can be read.
     */
}

```



```

boolean canRead() throws IOException;

/**
 * Tests whether the calling application can modify the resource denoted by
 * this abstract path.
 *
 * @return {@code true} if the resource for this path exists and can be
 *         modified, {@code false} otherwise.
 * @throws IOException
 *         If a problem occurs while attempting to determine if the
 *         resource can be modified.
 */
boolean canWrite() throws IOException;

/**
 * Tests whether the specified user is permitted to read the resource
 * denoted by this abstract path.
 *
 * <p>
 * When WMQFTE invokes this method, the user identifier is the MQMD user
 * identifier for the requesting transfer.
 *
 * @param userId
 *         User identifier to test for access.
 * @return {@code true} if the resource for this abstract path exists and is
 *         permitted to be read by the specified user, {@code false}
 *         otherwise.
 * @throws IOException
 *         If a problem occurs while attempting to determine if the user
 *         is permitted to read the resource.
 */
boolean readPermitted(String userId) throws IOException;

/**
 * Tests whether the specified user is permitted to modify the resource
 * denoted by this abstract path.
 *
 * <p>
 * When WMQFTE invokes this method, the user identifier is the MQMD user
 * identifier for the requesting transfer.
 *
 * @param userId
 *         User identifier to test for access.
 * @return {@code true} if the resource for this abstract path exists and is
 *         permitted to be modified by the specified user, {@code false}
 *         otherwise.
 * @throws IOException
 *         If a problem occurs while attempting to determine if the user
 *         is permitted to modify the resource.
 */
boolean writePermitted(String userId) throws IOException;

/**
 * Tests if the resource denoted by this abstract path is a directory-type
 * resource.
 *
 * @return {@code true} if the resource denoted by this abstract path is a
 *         directory type resource, {@code false} otherwise.
 */
boolean isDirectory();

/**
 * Creates the resource denoted by this abstract path, if it does not
 * already exist.
 *
 * @return {@code true} if the resource does not exist and was successfully
 *         created, {@code false} if the resource already existed.
 * @throws RecoverableIOException
 *         If a recoverable problem occurs while attempting to create
 *         the resource. This means that WMQFTE can attempt to recover
 *         the transfer.
 * @throws IOException
 *         If some other I/O problem occurs.
 */
boolean createNewPath() throws RecoverableIOException, IOException;

/**
 * Tests if the resource denoted by this abstract path is a file-type
 * resource.
 *
 * @return {@code true} if the resource denoted by this abstract path is a
 *         file type resource, {@code false} otherwise.
 */
boolean isFile();

```

```

/**
 * Obtains the last modified time for the resource denoted by this abstract
 * path.
 * <p>
 * This time is measured in milliseconds since the epoch (00:00:00 GMT,
 * January 1, 1970).
 *
 * @return The last modified time for the resource denoted by this abstract
 *         path, or a value of 0L if the resource does not exist or a
 *         problem occurs.
 */
long lastModified();

/**
 * Deletes the resource denoted by this abstract path.
 * <p>
 * If the resource is a directory, it must be empty for the delete to work.
 *
 * @throws IOException
 *         If the delete of the resource fails for any reason.
 */
void delete() throws IOException;

/**
 * Renames the resource denoted by this abstract path to the specified
 * destination abstract path.
 * <p>
 * The rename should still be successful if the resource for the specified
 * destination abstract path already exists and it is possible to replace
 * it.
 *
 * @param destination
 *        The new abstract path for the resource denoted by this
 *        abstract path.
 * @throws IOException
 *        If the rename of the resource fails for any reason.
 */
void renameTo(IOExitResourcePath destination) throws IOException;

/**
 * Creates a new path to use for writing to a temporary resource that did
 * not previously exist.
 * <p>
 * The implementation can choose the abstract path name for the temporary
 * resource. However, for clarity and problem diagnosis, the abstract path
 * name for the temporary resource should be based on this abstract path
 * name with the specified suffix appended and additional characters to make
 * the path unique (for example, sequence numbers), as required.
 * <p>
 * When WMQFTE transfers data to a destination it normally attempts to first
 * write to a temporary resource then on transfer completion renames the
 * temporary resource to the required destination. This method is called by
 * WMQFTE to create a new temporary resource path. The returned path should
 * be new and the resource should not previously exist.
 *
 * @param suffix
 *        Recommended suffix to use for the generated temporary path.
 *
 * @return A new {@link IOExitResourcePath} instance for the temporary
 *         resource path, that did not previously exist.
 * @throws RecoverableIOException
 *         If a recoverable problem occurs whilst attempting to create
 *         the temporary resource. This means that WMQFTE can attempt to
 *         recover the transfer.
 * @throws IOException
 *         If some other I/O problem occurs.
 */
IOExitResourcePath createTempPath(String suffix)
    throws RecoverableIOException, IOException;

/**
 * Opens a {@link IOExitChannel} instance for reading data from the resource
 * denoted by this abstract path. The current data byte position for the
 * resource is expected to be the passed position value, such that when
 * {@link IOExitChannel#read(java.nio.ByteBuffer)} is called, data starting
 * from that position is read.
 *
 * @param position
 *        The required data byte read position.
 * @return A new {@link IOExitChannel} instance allowing data to be read
 *         from the resource denoted by this abstract path.
 */

```

```

    * @throws RecoverableIOException
    *         If a recoverable problem occurs while attempting to open the
    *         resource for reading. This means that WMQFTE can attempt to
    *         recover the transfer.
    * @throws IOException
    *         If some other I/O problem occurs.
    */
    IOExitChannel openForRead(long position) throws RecoverableIOException,
        IOException;

    /**
     * Opens a {@link IOExitChannel} instance for writing data to the resource
     * denoted by this abstract path. Writing of data, using the
     * {@link IOExitChannel#write(java.nio.ByteBuffer)} method, starts at either
     * the beginning of the resource or end of the current data for the
     * resource, depending on the specified append parameter.
     *
     * @param append
     *         When {@code true} indicates that data written to the resource
     *         should be appended to the end of the current data. When
     *         {@code false} indicates that writing of data is to start at
     *         the beginning of the resource; any existing data is lost.
     * @return A new {@link IOExitChannel} instance allowing data to be written
     *         to the resource denoted by this abstract path.
     * @throws RecoverableIOException
     *         If a recoverable problem occurs whilst attempting to open the
     *         resource for writing. This means that WMQFTE can attempt to
     *         recover the transfer.
     * @throws IOException
     *         If some other I/O problem occurs.
     */
    IOExitChannel openForWrite(boolean append) throws RecoverableIOException,
        IOException;

    /**
     * Tests if the resource denoted by this abstract path is in use by another
     * application. Typically, this is because another application has a lock on
     * the resource either for shared or exclusive access.
     *
     * @return {@code true} if resource denoted by this abstract path is in use
     *         by another application, {@code false} otherwise.
     */
    boolean inUse();

    /**
     * Obtains a {@link IOExitProperties} instance for properties associated
     * with the resource denoted by this abstract path.
     * <p>
     * WMQFTE will read these properties to govern how a transfer behaves when
     * interacting with the resource.
     *
     * @return A {@link IOExitProperties} instance for properties associated
     *         with the resource denoted by this abstract path.
     */
    IOExitProperties getProperties();
}

```

相关概念

第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的底层文件系统 I/O 工作。

IOExitResourcePath2.java 接口

IOExitResourcePath2.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"

```

```

*
* 5724-H72
*
*   Copyright IBM Corp. 2011, 2023. All Rights Reserved.
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*/
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;
import java.util.List;

public interface IOExitResourcePath2 extends IOExitResourcePath {

    /**
     * @return a list of path attributes which the exit wishes
     *         to associate with data read from the path.
     */
    List<IOExitPathAttribute> getAttributes();

    /**
     * An extension to the
     * {@link IOExitResourcePath#newPath(String)} which allows
     * path attributes to be specified, for use when creating
     * or updating a path.
     *
     * @param child as per
     *             {@link IOExitResourcePath#newPath(String)}
     *
     * @param attributes a list of path attributes which the
     *                   exit can choose to apply to file system
     *                   objects that are created, updated
     *                   or read using this path.
     *
     * @return as per {@link IOExitResourcePath#newPath(String)}
     *
     * @throws IOException as per
     *             {@link IOExitResourcePath#newPath(String)}.
     *             Can also be thrown if the
     *             <code>attributes</code> parameter
     *             contains a
     *             <code>IOExitPathAttribute</code> which
     *             the exit implementation does not
     *             understand.
     */
    IOExitResourcePath newPath(final String child,
                               List<IOExitPathAttribute> attributes)
        throws IOException;
}

```

相关概念

第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的的底层文件系统 I/O 工作。

IOExitWildcardPath.java 接口

IOExitWildcardPath.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 *   Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or

```

```

* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*/
package com.ibm.wmqfte.exitroutine.api;

/**
 * Represents a path that denotes a wildcard. This can be used to match multiple
 * resource paths.
 */
public interface IOExitWildcardPath extends IOExitPath {

```

相关概念

第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 361 页的『使用 IBM MQ Managed File Transfer 传输 I/O 用户出口』

您可以使用 IBM MQ Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 IBM MQ Managed File Transfer 传输的底层文件系统 I/O 工作。

MonitorExit.java 接口

MonitorExit.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2009, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately prior to starting a task as the result of a monitor trigger
 */
public interface MonitorExit {

    /**
     * Invoked immediately prior to starting a task as the result of a monitor
     * trigger.
     *
     * @param environmentMetaData
     *      meta data about the environment in which the implementation
     *      of this method is running. This information can only be read,
     *      it cannot be updated by the implementation. The constant
     *      defined in <code>EnvironmentMetaDataConstants</code> class can
     *      be used to access the data held by this map.
     *
     * @param monitorMetaData
     *      meta data to associate with the monitor. The meta data passed
     *      to this method can be altered, and the changes will be
     *      reflected in subsequent exit routine invocations. This map
     *      also contains keys with IBM reserved names. These entries are
     *      defined in the <code>MonitorMetaDataConstants</code> class and
     *      have special semantics. The the values of the IBM reserved names
     *      cannot be modified by the exit
     *
     * @param taskDetails
     *      An XML String representing the task to be executed as a result of
     *      the monitor triggering. This XML string may be modified by the
     *      exit
     *
     * @return
     *      a monitor exit result object which is used to determine if the
     *      task should proceed, or be cancelled.
     */

```

```

        */
        MonitorExitResult onMonitor(Map<String, String> environmentMetaData,
                                    Map<String, String> monitorMetaData,
                                    Reference<String> taskDetails);
    }

```

相关概念

第 229 页的『资源监控』

您可以监视 IBM MQ Managed File Transfer 资源；例如，队列或目录。当满足针对该资源的条件时，资源监视器将启动任务，如文件传输。您可以使用 **fteCreateMonitor** 命令或 IBM MQ Explorer 的 IBM MQ Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 1013 页的『SourceTransferStartExit.java 接口』

第 1012 页的『SourceTransferEndExit.java 接口』

第 983 页的『DestinationTransferStartExit.java 接口』

第 982 页的『DestinationTransferEndExit.java 接口』

第 1007 页的『ProtocolBridgeCredentialExit.java 接口』

ProtocolBridgeCredentialExit.java 接口

ProtocolBridgeCredentialExit.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * Copyright IBM Corp. 2008, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;

/**
 * An interface that is implemented by classes that are to be invoked as part of
 * user exit routine processing. This interface defines methods that will
 * be invoked by a protocol bridge agent to map the MQ user id of the transfer to credentials
 * that are to be used to access the protocol server.
 * There will be one instance of each implementation class per protocol bridge agent. The methods
 * can be called from different threads so the methods must be synchronized.
 */
public interface ProtocolBridgeCredentialExit {

    /**
     * Invoked once when a protocol bridge agent is started. It is intended to initialize
     * any resources that are required by the exit
     *
     * @param bridgeProperties
     *            The values of properties defined for the protocol bridge.
     *            These values can only be read, they cannot be updated by
     *            the implementation.
     *
     * @return
     *            true if the initialization is successful and false if unsuccessful
     *            If false is returned from an exit the protocol bridge agent will not
     *            start
     */
    public boolean initialize(final Map<String> bridgeProperties);

}

```

```

    * Invoked once for each transfer to map the MQ user id in the transfer message to the
    * credentials to be used to access the protocol server
    *
    * @param mqUserId The MQ user id from which to map to the credentials to be used
    *                access the protocol server
    * @return        A credential exit result object that contains the result of the map and
    *                the credentials to use to access the protocol server
    */

public CredentialExitResult mapMQUserId(final String mqUserId);

/**
 * Invoked once when a protocol bridge agent is shutdown. It is intended to release
 * any resources that were allocated by the exit
 *
 * @param bridgeProperties
 *        The values of properties defined for the protocol bridge.
 *        These values can only be read, they cannot be updated by
 *        the implementation.
 *
 * @return
 */
public void shutdown(final Map<String> bridgeProperties);
}

```

相关概念

[第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』](#)

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关任务

[第 283 页的『使用出口类映射文件服务器的凭证』](#)

如果您不想使用协议网桥代理的缺省凭证映射功能，那么您可以通过编写您自己的用户出口将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器上的用户凭证中。IBM MQ Managed File Transfer 提供了执行用户凭证映射的样本用户出口。如果您配置凭证映射用户出口，那么它们会取代缺省凭证映射功能。

ProtocolBridgeCredentialExit2.java 接口

ProtocolBridgeCredentialExit2.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

/**
 * An interface that is implemented by classes that are invoked as part of user
 * exit routine processing. This interface defines methods that are invoked by a
 * protocol bridge agent to map the MQ user ID of the transfer to credentials
 * used to access a specified protocol bridge server. There will be one instance
 * of each implementation class for each protocol bridge agent. The methods can
 * be called from different threads so the methods must be synchronized.
 */
public interface ProtocolBridgeCredentialExit2 extends
    ProtocolBridgeCredentialExit {

    /**
     * Invoked once for each transfer to map the MQ user ID in the transfer
     * message to the credentials used to access a specified protocol server.
     *
     * @param endPoint
     *        Information that describes the protocol server to be accessed.
     */
}

```



```

    * @param mqUserId
    *         The MQ user ID from which to map the credentials used to
    *         access the protocol server.
    * @return A {@link CredentialExitResult} instance that contains the result
    *         of the map and the credentials to use to access the protocol
    *         server.
    */
    public CredentialExitResult mapMQUserId(
        final ProtocolServerEndPoint endPoint, final String mqUserId);
}

```

相关概念

第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关任务

第 283 页的『使用出口类映射文件服务器的凭证』

如果您不想使用协议网桥代理的缺省凭证映射功能，那么您可以通过编写您自己的用户出口将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器上的用户凭证中。IBM MQ Managed File Transfer 提供了执行用户凭证映射的样本用户出口。如果您配置凭证映射用户出口，那么它们会取代缺省凭证映射功能。

ProtocolBridgePropertiesExit2.java 接口

ProtocolBridgePropertiesExit2.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;
import java.util.Properties;

/**
 * An interface that is implemented by classes that are to be invoked as part of
 * user exit routine processing. This interface defines methods that will be
 * invoked by a protocol bridge agent to look up properties for protocol servers
 * that are referenced in transfers.
 * <p>
 * There will be one instance of each implementation class for each protocol
 * bridge agent. The methods can be called from different threads so the methods
 * must be synchronised.
 */
public interface ProtocolBridgePropertiesExit2 {

    /**
     * Invoked once when a protocol bridge agent is started. It is intended to
     * initialize any resources that are required by the exit.
     *
     * @param bridgeProperties
     *         The values of properties defined for the protocol bridge.
     *         These values can only be read, they cannot be updated by the
     *         implementation.
     * @return {@code true} if the initialization is successful and {@code
     *         false} if unsuccessful. If {@code false} is returned from an exit
     *         the protocol bridge agent will not start.
     */
    public boolean initialize(final Map<String, String> bridgeProperties);

    /**
     * Invoked when the Protocol Bridge needs to access the protocol bridge credentials XML file.
     *
     * @return a {@link String} object giving the location of the ProtocolBridgeCredentials.xml
     */
}

```

```

*/
public String getCredentialLocation ();

/**
 * Obtains a set of properties for the specified protocol server name.
 * <p>
 * The returned {@link Properties} must contain entries with key names
 * corresponding to the constants defined in
 * {@link ProtocolServerPropertyConstants} and in particular must include an
 * entry for all appropriate constants described as required.
 *
 * @param protocolServerName
 *      The name of the protocol server whose properties are to be
 *      returned. If a null or a blank value is specified, properties
 *      for the default protocol server are to be returned.
 * @return The {@link Properties} for the specified protocol server, or null
 *      if the server cannot be found.
 */
public Properties getProtocolServerProperties(
    final String protocolServerName);

/**
 * Invoked once when a protocol bridge agent is shut down. It is intended to
 * release any resources that were allocated by the exit.
 *
 * @param bridgeProperties
 *      The values of properties defined for the protocol bridge.
 *      These values can only be read, they cannot be updated by the
 *      implementation.
 */
public void shutdown(final Map<String, String> bridgeProperties);
}

```

相关概念

[第 279 页的『使用出口类 \(ProtocolBridgePropertiesExit.java\) 查找协议文件服务器属性』](#)

如果您有大量协议文件服务器，可以实现

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` 接口来查找传输中引用的协议文件服务器属性。您可以实现此接口，而不是维护 `ProtocolBridgeProperties.xml` 文件。建议您使用 `ProtocolBridgePropertiesExit2.java` 接口，但 `ProtocolBridgePropertiesExit.java` 接口也受支持。如果在 WebSphere MQ File Transfer Edition 中存在 `ProtocolBridgePropertiesExit.java` 接口的现有实现，那么可以在 IBM WebSphere MQ V7.5 或更高版本中使用该实现。`ProtocolBridgePropertiesExit2.java` 中新的 `getCredentialLocation` 方法使用 `ProtocolBridgeCredentials.xml` 文件的缺省位置，即您的主目录。

[第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』](#)

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关任务

[第 283 页的『使用出口类映射文件服务器的凭证』](#)

如果您不想使用协议网桥代理的缺省凭证映射功能，那么您可以通过编写您自己的用户出口将 IBM MQ Managed File Transfer 中的用户凭证映射到文件服务器上的用户凭证中。IBM MQ Managed File Transfer 提供了执行用户凭证映射的样本用户出口。如果您配置凭证映射用户出口，那么它们会取代缺省凭证映射功能。

SourceFileExitFileSpecification.java 类

SourceFileExitFileSpecification.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2012, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

```

```

import java.util.Map;

/**
 * A specification of the file names to use for a file transfer, as evaluated by the
 * agent acting as the source of the transfer.
 */
public final class SourceFileExitFileSpecification {

    private final String sourceFileSpecification;
    private final String destinationFileSpecification;
    private final Map<String, String> sourceFileMetaData;
    private final Map<String, String> destinationFileMetaData;

    /**
     * Constructor. Creates a source file exit file specification.
     *
     * @param sourceFileSpecification
     *        the source file specification to associate with the source file
     *        exit file specification.
     *
     * @param destinationFileSpecification
     *        the destination file specification to associate with the
     *        source file exit file specification.
     *
     * @param sourceFileMetaData
     *        the source file meta data.
     *
     * @param destinationFileMetaData
     *        the destination file meta data .
     */
    public SourceFileExitFileSpecification(final String sourceFileSpecification,
                                           final String destinationFileSpecification,
                                           final Map<String, String> sourceFileMetaData,
                                           final Map<String, String> destinationFileMetaData) {
        this.sourceFileSpecification = sourceFileSpecification;
        this.destinationFileSpecification = destinationFileSpecification;
        this.sourceFileMetaData = sourceFileMetaData;
        this.destinationFileMetaData = destinationFileMetaData;
    }

    /**
     * Returns the destination file specification.
     *
     * @return the destination file specification. This represents the location,
     *        on the agent acting as the destination for the transfer, where the
     *        file should be written. Exit routines installed into the agent
     *        acting as the destination for the transfer may override this value.
     */
    public String getDestination() {
        return destinationFileSpecification;
    }

    /**
     * Returns the source file specification.
     *
     * @return the source file specification. This represents the location where
     *        the file data will be read from.
     */
    public String getSource() {
        return sourceFileSpecification;
    }

    /**
     * Returns the file meta data that relates to the source file specification.
     *
     * @return the file meta data that relates to the source file specification.
     */
    public Map<String, String> getSourceFileMetaData() {
        return sourceFileMetaData;
    }

    /**
     * Returns the file meta data that relates to the destination file specification.
     *
     * @return the file meta data that relates to the destination file specification.
     */
    public Map<String, String> getDestinationFileMetaData() {
        return destinationFileMetaData;
    }
}

```

相关概念

第 971 页的『用户出口例程元数据』

可向 IBM MQ Managed File Transfer 的用户出口例程提供三种不同类型的元数据：环境、传输和文件元数据。此元数据显示为 Java 键/值对的映射。

SourceTransferEndExit.java 接口

SourceTransferEndExit.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitpoint.api;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately after completing a transfer on the agent acting as the
 * source of the transfer.
 */
public interface SourceTransferEndExit {

    /**
     * Invoked immediately after the completion of a transfer on the agent acting as
     * the source of the transfer.
     *
     * @param transferExitResult
     *        a result object reflecting whether or not the transfer completed
     *        successfully.
     *
     * @param sourceAgentName
     *        the name of the agent acting as the source of the transfer.
     *        This is the name of the agent that the implementation of this
     *        method will be invoked from.
     *
     * @param destinationAgentName
     *        the name of the agent acting as the destination of the
     *        transfer.
     *
     * @param environmentMetaData
     *        meta data about the environment in which the implementation
     *        of this method is running. This information can only be read,
     *        it cannot be updated by the implementation. The constants
     *        defined in <code>EnvironmentMetaDataConstants</code> class can
     *        be used to access the data held by this map.
     *
     * @param transferMetaData
     *        meta data to associate with the transfer. The information can
     *        only be read, it cannot be updated by the implementation. This
     *        map may also contain keys with IBM reserved names. These
     *        entries are defined in the <code>TransferMetaDataConstants</code>
     *        class and have special semantics.
     *
     * @param fileResults
     *        a list of file transfer result objects that describe the source
     *        file name, destination file name and result of each file transfer
     *        operation attempted.
     *
     * @return
     *        an optional description to enter into the log message describing
     *        transfer completion. A value of <code>null</code> can be used
     *        when no description is required.
     */
    String onSourceTransferEnd(TransferExitResult transferExitResult,
                              String sourceAgentName,
                              String destinationAgentName,
                              Map<String, String>environmentMetaData,
                              Map<String, String>transferMetaData,
```

```
List<FileTransferResult>fileResults);
```

```
}
```

相关概念

第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 1013 页的『SourceTransferStartExit.java 接口』

第 983 页的『DestinationTransferStartExit.java 接口』

第 982 页的『DestinationTransferEndExit.java 接口』

第 1006 页的『MonitorExit.java 接口』

第 1007 页的『ProtocolBridgeCredentialExit.java 接口』

SourceTransferStartExit.java 接口

SourceTransferStartExit.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitpoint.api;

import java.util.List;
import java.util.Map;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately prior to starting a transfer on the agent acting as the
 * source of the transfer.
 */
public interface SourceTransferStartExit {

    /**
     * Invoked immediately prior to starting a transfer on the agent acting as
     * the source of the transfer.
     *
     * @param sourceAgentName
     *        the name of the agent acting as the source of the transfer.
     *        This is the name of the agent that the implementation of this
     *        method will be invoked from.
     *
     * @param destinationAgentName
     *        the name of the agent acting as the destination of the
     *        transfer.
     *
     * @param environmentMetaData
     *        meta data about the environment in which the implementation
     *        of this method is running. This information can only be read,
     *        it cannot be updated by the implementation. The constants
     *        defined in <code>EnvironmentMetaDataConstants</code> class can
     *        be used to access the data held by this map.
     *
     * @param transferMetaData
     *        meta data to associate with the transfer. The meta data passed
     *        to this method can be altered, and the changes to will be
     *        reflected in subsequent exit routine invocations. This map may
     *        also contain keys with IBM reserved names. These entries are
     *        defined in the <code>TransferMetaDataConstants</code> class and
     */
}
```

```

        *          have special semantics.
        *
        * @param fileSpecs
        *          a list of file specifications that govern the file data to
        *          transfer. The implementation of this method can add entries,
        *          remove entries, or modify entries in this list and the changes
        *          will be reflected in the files transferred.
        *
        * @return    a transfer exit result object which is used to determine if the
        *          transfer should proceed, or be cancelled.
        */
    TransferExitResult onSourceTransferStart(String sourceAgentName,
                                           String destinationAgentName,
                                           Map<String, String> environmentMetaData,
                                           Map<String, String> transferMetaData,
                                           List<SourceFileExitFileSpecification> fileSpecs);
}

```

相关概念

第 358 页的『使用用户出口例程定制 IBM MQ Managed File Transfer』

您可以通过使用称为用户出口例程的个人程序来定制 IBM MQ Managed File Transfer 的功能。

相关参考

第 1010 页的『SourceFileExitFileSpecification.java 类』

第 1012 页的『SourceTransferEndExit.java 接口』

第 983 页的『DestinationTransferStartExit.java 接口』

第 982 页的『DestinationTransferEndExit.java 接口』

第 1006 页的『MonitorExit.java 接口』

第 1007 页的『ProtocolBridgeCredentialExit.java 接口』

TransferExitResult.java 接口

TransferExitResult.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */

package com.ibm.wmqfte.exitroutine.api;

/**
 * The result of invoking a transfer exit routine. It is composed of a result
 * code, which determines if the transfer should proceed, and an optional explanatory
 * message. The explanation, if present, is entered into the log message.
 */
public class TransferExitResult {

    private final TransferExitResultCode resultCode;
    private final String explanation;

    /**
     * For convenience, a static "proceed" result with no associated explanation
     * message.
     */
    public static final TransferExitResult PROCEED_RESULT =
        new TransferExitResult(TransferExitResultCode.PROCEED, null);

    /**
     * Constructor. Creates a transfer exit result object with a specified result
     * code and explanation.
     */
}

```

```

    * @param resultCode
    *         The result code to associate with the exit result being created.
    *
    * @param explanation
    *         The explanation to associate with the exit result being created.
    *         A value of <code>null</code> can be specified to indicate no
    *         explanation.
    */
    public TransferExitResult(TransferExitResultCode resultCode, String explanation) {
        this.resultCode = resultCode;
        this.explanation = explanation;
    }

    /**
     * Returns the explanation associated with this transfer exit result.
     *
     * @return    the explanation associated with this exit result.
     */
    public String getExplanation() {
        return explanation;
    }

    /**
     * Returns the result code associated with this transfer exit result.
     *
     * @return    the result code associated with this exit result.
     */
    public TransferExitResultCode getResultCode() {
        return resultCode;
    }
}

```

相关参考

[第 1013 页的『SourceTransferStartExit.java 接口』](#)

[第 983 页的『DestinationTransferStartExit.java 接口』](#)

[第 982 页的『DestinationTransferEndExit.java 接口』](#)

[第 1006 页的『MonitorExit.java 接口』](#)

[第 1007 页的『ProtocolBridgeCredentialExit.java 接口』](#)

相关信息

[使用用户出口定制 MFT](#)

您可放置在代理命令队列中消息的消息格式

以下 XML 模式定义了可放置到代理命令队列中消息的格式，这些消息用于请求代理执行某项操作。XML 消息可通过使用命令行命令或某个应用程序放置到代理命令队列中。

相关参考

[第 842 页的『文件传输请求消息格式』](#)

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the `FileTransfer.xsd` schema and have the `<request>` element as the root element. `FileTransfer.xsd` 模式文档位于 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。 `FileTransfer.xsd` 模式导入相同目录中的 `fteutils.xsd`。

[第 859 页的『监视器请求消息格式』](#)

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 `fteCreateMonitor` 命令或使用 WebSphere MQ Explorer 接口。

[第 868 页的『Ping 代理请求消息格式』](#)

您可以通过发出 **`ftePingAgent`** 命令或将 XML 消息放置在代理命令队列中 ping 代理。ping 代理请求 XML 必须符合 `PingAgent.xsd` 模式。安装 IBM MQ Managed File Transfer 后，您可以在以下目录中找到 `PingAgent.xsd` 模式文件：`MQ_INSTALLATION_PATH/mqft/samples/schema`。 `PingAgent.xsd` 模式导入相同目录中的 `fteutils.xsd`。

[第 870 页的『应答消息格式』](#)

代理在其代理命令队列中收到 XML 消息时，如果需要响应，那么代理会向原始消息中定义的应答队列发送一条 XML 应答消息。应答 XML 符合 Reply.xsd 模式。Reply.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。Reply.xsd 模式将导入位于相同目录中的 fteutils.xsd。

文件传输请求消息格式

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

文件传输消息可以包含以下三个根元素之一：

- <request> - 用于新文件传输请求、受管调用请求或删除暂挂的调度传输
- <cancel> - 用于取消正在进行的文件传输
- <transferSpecifications> - for specifying multiple transfer file groups, used by the **fteCreateTransfer** command

For information about specifying multiple transfer groups by using the <transferSpecifications> element, see [使用传输定义文件](#).

模式

以下模式描述在传输请求 XML 消息中有效的元素。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
<xsd:include schemaLocation="fteutils.xsd"/>

<!--
  Defines the request of a managed transfer and version number
  <request version="1.00" ...
    <managedTransfer>
      ...
    </managedTransfer>
  </request>
-->
<xsd:element name="request">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element name="managedTransfer" type="managedTransferType"/>
      <xsd:element name="deleteScheduledTransfer" type="deleteScheduledTransferType" />
      <xsd:element name="managedCall" type="managedCallType"/>
    </xsd:choice>
    <xsd:attribute name="version" type="versionType" use="required" />
  </xsd:complexType>
</xsd:element>

<!--
  Defines the cancel request of a managed transfer and version number
  <cancel version="1.00"
    xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
    <originator>
      <hostName>myMachine</hostName>
      <userID>myUserId</userID>
    </originator>      - Delete a scheduled transfer.

    <transfer>
      Transfer ID to Cancel
    </transfer>
  </cancel>
-->
<xsd:element name="cancel">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="originator" type="hostUserIDType" maxOccurs="1" minOccurs="1" />
      <xsd:choice>
        <xsd:element name="transfer" type="IDType" maxOccurs="1" minOccurs="1" />
        <xsd:element name="call" type="IDType" maxOccurs="1" minOccurs="1" />
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```

        <xsd:element name="reply" type="replyType" maxOccurs="1" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="version" type="versionType" use="required" />
</xsd:complexType>
</xsd:element>

<!--
    Defines the transfer definition element structure.
    <transferSpecifications>
        <item ...
        <item ...
    </transferSpecifications>
-->
<xsd:element name="transferSpecifications">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="item" type="itemType" minOccurs="1" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<!--
    Define a managed transfer of an instigator and request
    <managedTransfer>

        <originator>
            ...
        </originator>

        <schedule>
            <submit timebase="source"|"UTC">2008-12-07T16:07</submit>
            <repeat>
                <frequency interval="hours">2</frequency>
                <expireTime>2008-12-08T16:07</expireTime>
            </repeat>
        </schedule>

        <sourceAgent agent="here" QMgr="near" />
        <destinationAgent agent="there" QMgr="far" />

        <trigger>
            ...
        </trigger>

        <transferSet>
            ...
        </transferSet>
    </managedTransfer>
-->

    <xsd:complexType name="managedTransferType">
        <xsd:sequence>
            <xsd:element name="originator" type="origTransferRequestType" maxOccurs="1"
minOccurs="1"/>
            <xsd:element name="schedule" type="scheduleType" maxOccurs="1" minOccurs="0"/>
            <xsd:element name="sourceAgent" type="agentType" maxOccurs="1" minOccurs="1"/>
            <xsd:element name="destinationAgent" type="agentClientType" maxOccurs="1" minOccurs="1" />
            <xsd:element name="trigger" type="triggerType" maxOccurs="1" minOccurs="0" />
            <xsd:element name="reply" type="replyType" maxOccurs="1" minOccurs="0" />
            <xsd:element name="transferSet" type="transferSetType" maxOccurs="1" minOccurs="1" />
            <xsd:element name="job" type="jobType" maxOccurs="1" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>

<!--
    This is a modified form of origRequestType which is used on a managed transfer request.
    The hostName and userID are mandatory attributes in this case.
-->
<xsd:complexType name="origTransferRequestType">
    <xsd:sequence>
        <xsd:element name="hostName" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="userID" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="mqmdUserID" type="xsd:string" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="webBrowser" type="xsd:string" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="webUserID" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

<!--
    Defines the transferset of source and destination agent and one or more files
    <transferset priority="1">
        <metaDataSet>

```

```

        <metaData key="keyname">keyvalue</metaData>
        <metaData key="keyname">keyvalue</metaData>
    </metaDataSet>

    <item>
        ...
    </item>
</transferSet>
-->
<xsd:complexType name="transferSetType">
    <xsd:sequence>
        <xsd:element name="metaDataSet" type="metaDataSetType" maxOccurs="1" minOccurs="0" />
        <xsd:element name="preSourceCall" type="commandActionType" maxOccurs="1"
minOccurs="0" />
        <xsd:element name="postSourceCall" type="commandActionType" maxOccurs="1"
minOccurs="0" />
        <xsd:element name="preDestinationCall" type="commandActionType" maxOccurs="1"
minOccurs="0" />
        <xsd:element name="postDestinationCall" type="commandActionType" maxOccurs="1"
minOccurs="0" />
        <xsd:element name="item" type="itemType" maxOccurs="unbounded" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="priority" type="priorityType" use="optional" />
</xsd:complexType>

<!--
    Define a file pair with source and destination
    <item mode=[binary|text]>
        <source recursive="false" disposition="leave">
            <file>filename</file>
        </source>

        <destination type="file" exist="error">
            <file>filename</file>
        </destination>

    </item>
-->
<xsd:complexType name="itemType">
    <xsd:sequence>
        <xsd:element name="source" type="fileSourceType" maxOccurs="1" minOccurs="1" />
        <xsd:element name="destination" type="fileDestinationType" maxOccurs="1" minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="mode" type="modeType" use="required" />
    <xsd:attribute name="checksumMethod" type="checkSumMethod" use="required" />
</xsd:complexType>

<!--
    Defines the request to delete scheduled file transfer.
    <deleteScheduledTransfer>
        <originator>
            <delete>
                <hostName>myMachine</hostName>
                <userID>myUserId</userID>
            </delete>
        </originator>
        <ID>56</ID>
    </deleteScheduledTransfer>
-->
<xsd:complexType name="deleteScheduledTransferType">
    <xsd:sequence>
        <xsd:element name="originator" type="origDeleteType" maxOccurs="1" minOccurs="1" />
        <xsd:element name="ID" type="idType" maxOccurs="1" minOccurs="1" />
        <xsd:element name="reply" type="replyType" maxOccurs="1" minOccurs="0" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="managedCallType">
    <xsd:sequence>
        <xsd:element name="originator" type="origRequestType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="agent" type="agentType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="reply" type="replyType" maxOccurs="1" minOccurs="0" />
        <xsd:element name="transferSet" type="callTransferSetType" maxOccurs="1" minOccurs="1" />
        <xsd:element name="job" type="jobType" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="callTransferSetType">
    <xsd:sequence>
        <xsd:element name="metaDataSet" type="metaDataSetType" maxOccurs="1" minOccurs="0" />

```

```

        <xsd:element name="call" type="commandActionType" maxOccurs="1" minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="priority" type="priorityType" use="optional" />
</xsd:complexType>
</xsd:schema>

```

了解传输请求消息

以下列表描述了在传输请求消息中使用的元素和属性：

元素描述

<request>

包含指定文件传输请求所需的所有元素的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 提供的该元素的版本，指定该值。

<managedTransfer>

包含单个文件传输或一组文件传输所需的所有元素的组元素。

<deleteScheduledTransfer>

包含发起方和标识信息以供取消调度传输的组元素。

<managedCall>

包含程序或可执行文件的单个受管调用所需的所有元素的组元素。

<ID>

指定要从暂挂的调度传输列表删除的传输请求的唯一标识。

<originator>

包含指定请求发起方的元素的组元素。

<hostName>

源文件所在系统的主机名。

<userID>

发起文件传输的用户标识。

<mqmdUserID>

可选。在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<schedule>

描述文件传输的调度时间、重复行为以及下一次出现的到期时间的组元素。

<submit>

指定已调度传输应启动的日期和时间。

属性	描述
timebase	指定要使用的时区。此属性可以具有下列其中一个值： - source - 使用源代理的时区 - admin - 使用发出命令的管理员的时区 - UTC - 使用全球标准时间
时区	依据 timebase 值的时区描述

<repeat>

包含有关已调度传输重复频率、已调度传输重复次数以及已调度传输何时停止重复的详细信息的一组元素。

<frequency>

重复传输之前必须等待的时间段。

属性	描述
区间 (interval)	时间间隔单位，必须是以下值之一： • 分钟 • 小时 • days • weeks • months • years

<expireTime>

指定重复的已调度传输停止的日期和时间的可选元素。该元素与 <expireCount> 元素互斥。

<expireCount>

指定在停止前发生已调度文件传输的次数的可选元素。该元素与 <expireTime> 元素互斥。

<sourceAgent>

指定源文件所在系统上的代理的名称。

属性	描述
代理程序	指定代理的名称。
QMgr	代理队列管理器的名称。

<destinationAgent>

指定您要向其传输文件的系统上代理的名称。

属性	描述
代理程序	指定代理的名称。
QMgr	代理队列管理器的名称。
hostName	代理队列管理器的主机名或 IP 地址。
portNumber	用于到目标代理队列管理器的客户机连接的端口号。
通道	用于连接到目标代理队列管理器的通道名称。

<trigger>

指定在文件传输发生时必须满足的条件的可选元素。

属性	描述
日志	指示是否记录触发失败的标志。有效值如下： • yes - 为失败的已触发传输创建日志条目 • no - 不为失败的已触发传输来创建日志条目

<fileExist>

指定位于与源代理相同系统上的文件名的以逗号分隔的列表。如果此名称列表中的文件满足触发器条件，那么将发生传输。该元素与 <fileSize> 元素互斥。

属性	描述
comparison	指示如何针对名称列表对源文件名求值。有效值如下： • = 名称列表中必须至少有一个文件名匹配 • != 名称列表中至少有一个文件不存在
值	指示比较类型： • exist: 文件必须存在

<fileSize>

指定位于与源代理相同系统上的文件名的以逗号分隔的列表。如果此名称列表中的文件满足触发器条件，那么将发生传输。该元素与 <fileExist> 元素互斥。

属性	描述
comparison	指示如何针对名称列表对源文件名求值。有效值如下： • >= 名称列表中存在一个文件名，并且至少为值属性中指定的最小大小
值	指定为整数值的文件大小，带有指定为以下之一的单位： • B - 字节 • KB - 千字节 • MB - 兆字节 • GB - 千兆字节 (单位值不区分大小写)

<reply>

指定为同步文件传输生成的临时应答队列的名称（在命令行上使用 **-w** 参数指定）。队列的名称由 `command.properties` 配置文件中的键 **dynamicQueuePrefix** 或缺省值 `WMQFTE.*` 定义。（如果未指定）。

属性	描述
详细信息	回复消息中是否需要详细传输结果信息。可以针对每个传输生成多条回复消息。有效值如下： • true - 需要详细回复信息。信息的格式与进度消息中发布到传输日志的相同，即 <transferSet> 元素。有关更多信息，请参阅第 662 页的『文件传输日志消息格式』。仅当传输源代理的 <code>enableDetailedReplyMessages</code> 属性设置为 <code>true</code> 时，才会显示详细回复信息。 • false - 不需要详细回复信息。 缺省值是 <code>false</code> 。
QMGR	在其上生成临时动态队列以接收应答的命令队列管理器的名称。
persistent	写入应答队列的消息是否为持久性消息。有效值如下： • true - 消息持久 • false - 消息不持久 • qdef - 消息的持久性由应答队列的属性定义 缺省值是 <code>false</code> 。

<transferSet>

指定想要一起执行的一组文件传输或想要一起执行的一组受管调用。在传输期间，<transferSet> 是包含 <item> 元素的组元素。

属性	描述
priority	传输的优先级。优先级为 0-9 之间的一个值，其中 0 为最低优先级。缺省优先级为 0，并且缺省情况下，传输使用源代理的优先级。

<metaDataSet>

包含一个或多个元数据项的可选组元素。

<metaData>

指定将传递给代理调用的出口点的用户定义的元数据。元素包含元数据值作为字符串。

属性	描述
key	作为字符串的元数据名称

<call>

包含指定要调用的程序或可执行文件的 <command> 元素的组元素。

<command>

指定要调用的程序或可执行文件。命令必须位于代理命令路径上。有关更多信息，请参阅第 592 页的表 50。该元素可以包含可选的 <argument> 元素。

属性	描述
名	命令的名称。
successRC	该命令返回的成功返回码。缺省值为 0。
retryCount	命令失败时将重试的次数。
retryWait	命令重试之间等待的时间（秒）。
类型	要调用的程序的类型。有效值为 antscript、jcl 或 executable。

<argument>

指定要传递给命令的参数。

<item>

包含指定源和目标文件名和位置的元素的组元素。

属性	描述
mode	指定传输方式为二进制还是文本。
checksumMethod	指定生成消息摘要以创建数字签名的散列算法的类型。有效值为 MD5 或 none。

<source>

指定源系统上的文件以及在传输完成后是否移除这些文件的组元素

属性	描述
recursive	指定当 <source> 元素为目录或包含通配符时，以递归方式传输子目录中的文件。
disposition	指定在 <source> 已成功传输到其目标时要在 <source> 元素上执行的操作。有效值如下： • leave - 源文件保持不变。

属性	描述
	• delete - 在成功传输源文件后将源文件从源系统中删除。

<file>

指定传输源。对于分布式平台和 IBMi，传输源可以是文件或目录名称。对于 z/OS 平台，传输源可以是文件、目录、数据集或 PDS 名称。使用与操作系统格式一致的标准路径，例如 C:/from/here.txt。请勿使用文件 URI。

属性	描述
别名	指定源文件的别名。该别名是源文件的名称，不包括针对传输指定的任何目录路径。
EOL	指定文本传输的行尾标记符。有效值包括： • LF - 仅限换行字符 • CRLF - 回车符和换行符序列
编码	文本文件传输的源文件编码。
delimiter	指定在面向记录的源文件中的记录之间包含的定界符，例如，z/OS 数据集。将定界符值指定为 00-FF 范围内的两个十六进制数字，前缀为 x。例如，x12 或 x03,x7F。
delimiterType	指定目标文件中包含的位于单条消息数据之后的定界符类型。有效值如下： • binary - 十六进制定界符 仅当启用了 V7.0.4.1 功能时，该属性才可用。
delimiterPosition	指定在将面向记录的源文件记录写入常规文件时插入定界符的位置。有效值如下： • prefix - 定界符将插入目标文件中，在每条面向源记录的文件记录中的数据之前。 • postfix - 定界符将插入目标文件中，在每条面向源记录的文件记录的数据之后。
includeDelimiterInFile	指定是否在面向记录的源文件中的记录之间包含定界符。
keepTrailingSpaces	指定在文本方式传输过程中是否保留从固定长度格式的数据集读取的源记录上的尾部空格。缺省值为除去尾部空格。有效值如下： • true - 在从固定长度格式的数据集读取的源记录上保留尾部空格 • false - 除去从固定长度格式的数据集读取的源记录上的尾部空格

<queue>

在用于 <source> 元素时，指定作为传输源的队列的名称，这必须位于源代理队列管理器上。使用格式 *QUEUE*。请勿包含队列管理器名称，队列必须位于源代理队列管理器上。如果已经在 <destination> 元素中使用 <queue> 元素，那么将无法在 <source> 元素中使用该元素。

属性	描述
useGroups	指定是否仅传输源队列中第一个完整的消息组。有效值如下： • true - 仅传输第一个完整的消息组 • false - 传输源队列上的所有消息

属性	描述
groupId	指定要从源队列读取的消息组。 仅当 useGroups 属性的值为 true 时，该属性才有效。
delimiterType	指定目标文件中包含的位于单条消息数据之后的定界符类型。 有效值如下： • text-文本或 Java 文字定界符 • binary - 十六进制定界符
delimiter	指定在目标文件的单条消息数据之间包含的定界符。
delimiterPosition	指定目标文件包含的定界符是位于单个消息数据之前还是之后。 有效值如下： • prefix - 定界符包含在数据之前 • postfix - 定界符包含在数据之后
编码	指定源队列编码。
waitTime	指定源代理等待以下一个条件的时间（秒）： • 在源队列上出现一条消息（如果队列为空或已变为空） • 在源队列上出现一个完整组（如果 useGroups 属性已设置为 true） 有关设置 waitTime 值的信息，请参阅第 744 页的『用于对“消息到文件”传输指定等待时间的指南』。

<destination>

指定文件位于目标代理上时的目标和行为的组元素。

只能指定 <file> 和 <queue> 之一作为目标的子元素。

属性	描述
类型	目标的类型。 有效值如下： • file - 将文件指定为目标 • directory - 将目录指定为目标 • dataset - 指定 z/OS 数据集作为目标 • pds - 指定 z/OS 分区数据集作为目标 • queue - 指定 WebSphere MQ 队列作为目标 • filespace- 指定文件空间作为目标 仅当 <destination> 元素的子元素为 <queue>时，值 队列 才有效。 仅当 <destination> 元素的子元素为 <filespace>时，值 文件空间 才有效。 仅当 <destination> 元素具有子元素 <file>时，其他值才有效。
exist	指定在目标系统上已存在目标文件时执行的操作。 有效值如下： • error - 报告一个错误，并且未传输文件。 • overwrite - 覆盖现有的目标文件。 如果 <destination> 元素具有子元素 <queue> 或 <filespace>，那么此属性无效。

<file>

指定先前描述的 <destination> 元素的其他设置。使用与操作系统格式一致的标准路径，例如 C:/from/here.txt。使用与操作系统格式一致的标准路径，例如 C:/from/here.txt。请勿使用文件 URI。

属性	描述
别名	指定目标文件的别名。该别名是源文件的名称，不包括针对传输指定的任何目录路径。
编码	文本文件传输的目标文件编码。
EOL	指定文本传输的行尾标记符。有效值包括： • LF - 仅限换行字符 • CRLF - 回车符和换行符序列
truncateRecords	可选。指定将截断长于 LRECL 数据集属性的目标记录。 • True - 截断记录 • False - 合并记录 缺省设置为 false。

<queue>

在用于 <destination> 元素时，指定作为传输目标的队列的名称，该队列可以位于连接到目标代理队列管理器的任何队列管理器上。使用格式 *QUEUE@QM*，其中 *QUEUE* 是要放置消息的队列的名称，*QM* 是队列所在的队列管理器。如果已经在 <source> 元素中使用 <queue> 元素，那么将无法在 <destination> 元素中使用该元素。

属性	描述
delimiter	将文件拆分为多条消息的定界符。
delimiterType	指定定界符的类型。有效值如下： • text-Java 正则表达式 • binary - 十六进制字节序列 • size - 字节、千字节或兆字节数。例如，1 B、1 K 或 1 M。
delimiterPosition	指定期望定界符位于要在单条消息中包含的数据之前还是之后。有效选项如下： • prefix - 期望定界符在数据之前 • postfix - 期望定界符在数据之后
includeDelimiterInMessage	指定是否要在消息尾部包含用于将文件拆分为多条消息的定界符的布尔值。
编码	指定目标队列编码。
persistent	指定消息是否持久。有效值如下： • true - 消息持久 • false - 消息不持久 • qdef - 消息的持久性值由目标队列上的设置定义
setMqProps	指定是否在文件中的第一条消息上设置 WebSphere MQ 消息属性，以及在发生错误时是否将任何消息写入队列的布尔值。

属性	描述
unrecognisedCodePage	指定在目标队列管理器无法识别数据代码页时，文本方式传输将失败还是执行转换。有效值如下： - fail - 传输报告失败 - binary - 将数据转换为目标代码页，并且描述数据格式的 WebSphere MQ 消息头将设置为 MQFMT_NONE。 缺省行为是 fail。

<filespace>

指定要传输到的文件空间名称的组元素。

<name>

在和 <filespace> 元素一起使用时，该元素的值指定文件空间的名称。

<attributes>

Optional group element that contains one or more <attribute> elements to specify distribution attribute information if you are transferring files to a IBM 4690 store controller.

<attribute>

指定文件分发属性的可选元素。指定符号或数字值。

表 81: IBM MQ Managed File Transfer 中文件分发属性的有效值		
符号值	数字值	描述
DIST(LOCAL)	DIST(1)	本地文件
DIST(MIRRORED,UPDATE)	DIST(2)	映像文件，更新时分发
DIST(MIRRORED,CLOSE)	DIST(3)	映像文件，关闭时分发
DIST(COMPOUND,UPDATE)	DIST(4)	复合文件，更新时分发
DIST(COMPOUND,CLOSE)	DIST(5)	复合文件，关闭时分发

有关 IBM 4690 上 IBM MQ Managed File Transfer 的分发属性的更多信息，请参阅第 82 页的『文件分发属性』。

<preSourceCall>

指定在传输开始前要在传输源处调用的命令的组元素。

<postSourceCall>

指定在传输完成后要在传输源调用的命令的组元素。

<preDestinationCall>

指定在传输开始前要在传输目标处调用的命令的组元素。

<postDestinationCall>

指定在传输完成后要在传输目标处调用的命令的组元素。

<command>

在和 <preSourceCall>、<postSourceCall>、<preDestinationCall> 或 <postDestinationCall> 元素一起使用时，该元素指定要调用的命令。命令必须位于代理命令路径上。有关更多信息，请参阅第 592 页的表 50。

属性	描述
名	要运行的命令的名称。
successRC	在命令成功运行时期望的返回码。

<argument>
在和 <command> 元素一起使用时，该元素指定要传递到命令的参数。您可以在 <command> 元素中使用任意数量的 <argument> 元素。

<job>
Optional group element containing job information for the entire transfer specification. <job> is a user-defined job name identifier that is added to the log message when the transfer has started. This <job> element is the same as the <job> element that appears in the transfer log message, which is described in the following topic: [第 662 页的『文件传输日志消息格式』](#)。

<name>
在和 <job> 元素一起使用时，该元素的值可指定作业名称。

<transferSpecifications>
包含多个传输组的 <item> 元素的组元素。请参阅[使用传输定义文件](#)以获取有关如何使用该元素的更多详细信息。

<cancel>
包含取消正在进行的文件传输所需的所有元素的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 提供的该元素的版本，指定该值。

<transfer>
在和 <cancel> 元素一起使用时，该元素的值指定要取消的传输请求标识。

<job>
包含作业信息的组元素。

<jobName>
指定逻辑作业标识。

文件传输取消消息格式

文件传输请求返回一个标识特定代理的传输的 48 字符标识。该标识用于取消传输。

了解传输取消消息
描述传输取消消息中使用的元素和属性：

<cancel>
包含取消正在进行的文件传输所需的所有元素的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 提供的该元素的版本，指定该值。

<originator>
包含指定请求发起方的元素的组元素。

<hostName>
源文件所在系统的主机名。

<userID>
发起文件传输的用户标识。

<mqmdUserID>
可选。在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<transfer>

在和 <cancel> 元素一起使用时，该元素的值指定要取消的传输请求标识。

<job>

可选。包含作业信息的组元素。

<jobName>

指定逻辑作业标识。

示例

针对以下每个请求提供了符合该模式的 XML 消息的示例：

- [创建文件传输](#)
- [创建异步文件传输请求](#)
- [取消文件传输](#)
- [创建调度传输](#)
- [删除调度传输](#)
- [创建受管调用](#)
- [创建包含受管调用的文件传输](#)

相关参考

[第 855 页的『传输请求示例』](#)

您可以在代理命令队列上添加以请求代理创建或取消传输的消息的示例。

[第 857 页的『已安排的传输消息示例』](#)

可以置于代理命令队列以请求代理创建或删除调度的消息示例。

[第 858 页的『调用请求消息示例』](#)

可以放置在代理命令队列中以请求代理创建受管调用或创建调用程序的传输的消息示例。

[第 647 页的『代理状态消息格式』](#)

在创建或启动代理时，该代理会向其协调队列管理器上的 SYSTEM.FTE 主题（在 SYSTEM.FTE/Agents/*agent name* 主题上）发布其详细信息。

[第 658 页的『文件传输状态消息格式』](#)

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 SYSTEM.FTE/Transfers/*agent_name* /*transfer ID* 主题中），它符合 TransferStatus.xsd XML 模式。TransferStatus.xsd 文件位于 WMQMFT 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

[第 662 页的『文件传输日志消息格式』](#)

文件传输日志消息发布在主题字符串为 Log/*agent_name*/*transfer_ID* 的 SYSTEM.FTE 主题内。这些消息符合模式 TransferLog.xsd，该模式位于 IBM MQ Managed File Transfer 安装的 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。

[第 686 页的『已安排的传输日志消息格式』](#)

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 SYSTEM.FTE/Log/*agent name*/*schedule ID* 主题中）。该消息符合 ScheduleLog.xsd XML 模式。

[第 859 页的『监视器请求消息格式』](#)

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 fteCreateMonitor 命令或使用 WebSphere MQ Explorer 接口。

[第 871 页的『针对安全性的消息格式』](#)

本主题描述发布到协调队列管理器且与安全性相关的消息。

传输请求示例

您可以在代理命令队列上添加以请求代理创建或取消传输的消息的示例。

创建传输请求

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
version="4.00"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/>
    <destinationAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/>
    <transferSet>
      <item checksumMethod="MD5" mode="binary">
        <source disposition="leave" recursive="false">
          <file>/etc/passwd</file>
        </source>
        <destination exist="overwrite" type="directory">
          <file>/tmp</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

创建传输请求 - 传输至 IBM 4690

在该示例 XML 中，当将文件 xyz.txt 传输至 IBM 4690 存储控制器的目录 c:\adx_test 中时，设置为关闭时进行镜像。

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="5.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName><userID>bob</userID>
    </originator>
    <sourceAgent agent="AGENT_A" QMgr="qm_a"/>
    <destinationAgent agent="AGENT_B" QMgr="qm_b"/>
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>xyz.txt</file>
        </source>
        <destination type="directory" exist="error">
          <file>c:\adx_test</file>
          <attributes>
            <attribute>DIST(MIRRORED,CLOSE)</attribute>
          </attributes>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

有关 IBM 4690 上 IBM MQ Managed File Transfer 的分发类型属性的更多信息，请参阅第 82 页的『文件分发属性』。

创建传输请求 - 同步

在用户请求分块同步请求时，也就是，他们等待传输完成并接收状态消息，命令队列上放置的消息将包含一个应答元素，指定要向其发送应答消息的队列。以下示例显示在 FTEAGENT 使用的命令队列上放置的消息：

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
```



```

    <userID>USER1</userID>
  </originator>
  <sourceAgent agent="FTEAGENT"
    QMgr="QM1"/>
  <destinationAgent agent="AGENT2"
    QMgr="QM2"/>
  <reply QMGR="QM1">WMQFTE.492D0D5502770020</reply>
  <transferSet>
    <item mode="binary" checksumMethod="MD5">
      <source recursive="false" disposition="leave">
        <file>c:\sourcefiles\source1.doc</file>
      </source>
      <destination type="file" exist="overwrite">
        <file>c:\destinationfiles\dest1.doc</file>
      </destination>
    </item>
  </transferSet>
</managedTransfer>
</request>

```

将使用命令队列管理器的名称填充 <reply> 元素，其中，已创建临时动态队列来接收有关成功（或相反）完成传输的应答。临时动态队列名称由两部分组成：

- 由 `command.properties` 配置文件中的键 **dynamicQueuePrefix** 定义的前缀（它是 WMQFTE。缺省情况下）
- IBM MQ 生成的队列的标识

取消传输请求

```

<?xml version="1.0" encoding="UTF-8"?>
<cancel xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
  </originator>
  <transfer>414D51205553322E42494E44494E47538B0F404D032C0020</transfer>
  <reply QMGR="QM_JUPITER">WMQFTE.4D400F8B20002007</reply>
</cancel>

```

相关参考

第 842 页的『文件传输请求消息格式』

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the `FileTransfer.xsd` schema and have the <request> element as the root element. `FileTransfer.xsd` 模式文档位于 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。`FileTransfer.xsd` 模式导入相同目录中的 `fteutils.xsd`。

已安排的传输消息示例

可以置于代理命令队列以请求代理创建或删除调度的消息示例。

创建已安排的传输

```

<?xml version="1.0" encoding="UTF-8"?>
<request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName>
      <userID>mqm</userID>
    </originator>
    <schedule>
      <submit timebase="admin" timezone="Europe/London">2010-01-01T21:00</submit>
    </schedule>
    <sourceAgent QMgr="US2.BINDINGS" agent="US2.BINDINGS.FILE"/>
    <destinationAgent QMgr="US2.BINDINGS" agent="US2.BINDINGS.FILE"/>
    <transferSet>
      <item checksumMethod="MD5" mode="binary">

```

```

        <source disposition="leave" recursive="false">
            <file>/etc/passwd</file>
        </source>
        <destination exist="overwrite" type="directory">
            <file>/tmp</file>
        </destination>
    </item>
</transferSet>
</managedTransfer>
</request>

```

删除已安排的传输

```

<?xml version="1.0" encoding="UTF-8"?>
<request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    version="4.00"
    xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
    <deleteScheduledTransfer>
        <originator>
            <delete>
                <hostName>example.com.</hostName>
                <userID>mqm</userID>
            </delete>
        </originator>
        <ID>1</ID>
        <reply QMGR="US2.BINDINGS">WMQFTE.4D400F8B20003902</reply>
    </deleteScheduledTransfer>
</request>

```

相关参考

第 842 页的『文件传输请求消息格式』

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

调用请求消息示例

可以放置在代理命令队列中以请求代理创建受管调用或创建调用程序的传输的消息示例。

受管调用请求示例

```

<?xml version="1.0" encoding="UTF-8"?>
<request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    version="1.00"
    xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
    <managedCall>
        <originator>
            <hostName>example.com.</hostName>
            <userID>mqm</userID>
        </originator>
        <agent agent="DNWE" QMgr="QM1" />
        <transferSet>
            <call>
                <command name="echo" successRC="0">
                    <argument>call</argument>
                    <argument>test</argument>
                </command>
            </call>
        </transferSet>
    </managedCall>
    <job>
        <name>managedCallCalls.xml</name>
    </job>
</request>

```

具有调用的受管传输请求示例

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="1.00"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent agent="DNWE" QMgr="QM1"/>
    <destinationAgent agent="DNWE" QMgr="QM1"/>
    <transferSet>
      <preSourceCall>
        <command name="echo" successRC="0">
          <argument>preSourceCall</argument>
          <argument>test</argument>
        </command>
      </preSourceCall>
      <postSourceCall>
        <command name="echo" successRC="0">
          <argument>postSourceCall</argument>
          <argument>test</argument>
        </command>
      </postSourceCall>
      <preDestinationCall>
        <command name="echo" successRC="0">
          <argument>preDestinationCall</argument>
          <argument>test</argument>
        </command>
      </preDestinationCall>
      <postDestinationCall>
        <command name="echo" successRC="0">
          <argument>postDestinationCall</argument>
          <argument>test</argument>
        </command>
      </postDestinationCall>
    </transferSet>
  </managedTransfer>
</request>
```

相关概念

第 303 页的『指定要运行的程序』

您可在正在运行 IBM MQ Managed File Transfer 代理的系统上运行程序。作为文件传输请求的一部分，您可以指定在传输启动前或者完成后运行的程序。此外，您可以通过提交受管呼叫请求来启动不属于文件传输请求的程序。

相关参考

第 842 页的『文件传输请求消息格式』

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 模式文档位于 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。FileTransfer.xsd 模式导入相同目录中的 fteutils.xsd。

监视器请求消息格式

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 `fteCreateMonitor` 命令或使用 WebSphere MQ Explorer 接口。

监视器 XML 必须符合 Monitor.xsd 模式（即，将 <monitor> 元素用作根元素）。

监视器消息可以具有下列其中一个根元素：

- <monitor> - 用于创建和启动新的资源监视器
- <deleteMonitor> - 用于停止和删除现有监视器

`fteListMonitors` 命令没有命令消息，因为该命令直接从 SYSTEM.FTE 主题检索匹配的监视定义。

模式

以下模式描述了监视器请求 XML 消息中哪些元素有效。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema"
            targetNamespace="https://www.ibm.com/xmlns/wmqfte/7.0.1/
MonitorDefinition"
            xmlns="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition">

<xsd:include schemaLocation="FileTransfer.xsd" />

  <xsd:element name="monitor">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="name" type="monitorNameType"
          minOccurs="1" maxOccurs="1" />
        <xsd:element name="description" type="xsd:string"
          minOccurs="0" maxOccurs="1" />
        <xsd:element name="pollInterval" type="pollIntervalType"
          minOccurs="1" maxOccurs="1"
          default="10" />
        <xsd:element name="batch" type="batchType"
          minOccurs="0" maxOccurs="1" />
        <xsd:element name="agent" type="agentNameType"
          minOccurs="1" maxOccurs="1" />
        <xsd:element name="resources" type="monitorResourcesType"
          minOccurs="0" maxOccurs="1" />
        <xsd:element name="triggerMatch" type="triggerMatchType"
          minOccurs="1" maxOccurs="1" />
        <xsd:element name="reply" type="replyType"
          minOccurs="0" maxOccurs="1" />
        <xsd:element name="tasks" type="monitorTasksType"
          minOccurs="1" maxOccurs="1" />
        <xsd:element name="originator" type="origRequestType"
          minOccurs="1" maxOccurs="1" />
        <xsd:element name="job" type="jobType"
          minOccurs="0" maxOccurs="1" />
        <xsd:element name="defaultVariables" type="defaultVariablesType"
          minOccurs="0" maxOccurs="1" />
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="deleteMonitor">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="name" type="monitorNameType"
          minOccurs="1" maxOccurs="1" />
        <xsd:element name="originator" type="origRequestType"
          minOccurs="1" maxOccurs="1" />
        <xsd:element name="reply" type="replyType"
          minOccurs="0" maxOccurs="1" />
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required" />
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="transferRequestType">
    <xsd:choice>
      <xsd:element name="managedTransfer" type="managedTransferType" />
      <xsd:element name="managedCall" type="managedCallType" />
    </xsd:choice>
    <xsd:attribute name="version" type="versionType" />
  </xsd:complexType>

  <xsd:complexType name="monitorResourcesType">
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="directory" type="monitoredDirectoryType"
          minOccurs="1" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:element name="queue" type="monitoredQueueType" />
    </xsd:choice>
  </xsd:complexType>

  <xsd:complexType name="monitoredDirectoryType">
    <xsd:simpleContent>
```

```

        <xsd:extension base="xsd:string">
            <xsd:attribute name="recursionLevel"
type="xsd:nonNegativeInteger" />
            <xsd:attribute name="id" type="resourceIdAttrType" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="monitoredQueueType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="id" type="resourceIdAttrType" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="triggerMatchType">
    <xsd:sequence>
        <xsd:element name="conditions" type="conditionsType"
minOccurs="1" maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="conditionsType">
    <xsd:choice minOccurs="1">
        <xsd:element name="allOf" type="listPredicateType"
minOccurs="1" maxOccurs="1" />
        <xsd:element name="anyOf" type="listPredicateType"
minOccurs="1" maxOccurs="1" />
        <xsd:element name="condition" type="conditionType"
minOccurs="1" maxOccurs="1" />
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="listPredicateType">
    <xsd:choice>
        <xsd:element name="condition" type="conditionType"
minOccurs="1" maxOccurs="unbounded" />
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="conditionType">
    <xsd:sequence>
        <xsd:element name="name" type="conditionNameType"
minOccurs="0" maxOccurs="1" />
        <xsd:element name="resource" type="resourceIdType"
minOccurs="0" maxOccurs="1" />
        <xsd:choice minOccurs="1">
            <xsd:element name="fileMatch"
type="fileMatchConditionType"
minOccurs="1" maxOccurs="1" />
            <xsd:element name="fileNoMatch"
type="fileNoMatchConditionType"
minOccurs="1"
maxOccurs="1" />
            <xsd:element name="fileSize"
type="fileSizeConditionType"
minOccurs="1" maxOccurs="1" />
            <xsd:element name="queueNotEmpty"
type="queueNotEmptyConditionType"
minOccurs="1" maxOccurs="1" />
            <xsd:element name="completeGroups"
type="completeGroupsConditionType"
minOccurs="1" maxOccurs="1" />
            <xsd:element name="fileSizeSame" type="fileSizeSameType"
minOccurs="1" maxOccurs="1" />
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="fileMatchConditionType">
    <xsd:sequence>
        <xsd:element name="pattern" type="conditionPatternType"
minOccurs="0" default="*.*" />
        <xsd:element name="exclude" type="conditionPatternType"
minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="fileNoMatchConditionType">
    <xsd:sequence>
        <xsd:element name="pattern" type="conditionPatternType"

```

```

        minOccurs="0" default="*.*" />
        <xsd:element name="exclude" type="conditionPatternType"
        minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="fileSizeConditionType">
    <xsd:sequence>
        <xsd:element name="compare" type="sizeCompareType"
        minOccurs="1" default="0" />
        <xsd:element name="pattern" type="conditionPatternType"
        minOccurs="0" default="*.*" />
        <xsd:element name="exclude" type="conditionPatternType"
        minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="sizeCompareType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:int">
            <xsd:attribute name="operator" type="sizeOperatorType"
            use="required" />
            <xsd:attribute name="units" type="fileSizeUnitsType"
            use="required" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="sizeOperatorType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value=">=" />
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="fileSizeUnitsType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="[bB]|[kK][bB]|[mM][bB]|[gG][bB]" />
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="conditionPatternType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="type" type="patternTypeAttributeType"
            use="optional" default="wildcard" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="patternTypeAttributeType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="regex" />
        <xsd:enumeration value="wildcard" />
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="conditionNameType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string" />
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="queueNotEmptyConditionType"/>

<xsd:complexType name="completeGroupsConditionType"/>

<xsd:complexType name="fileSizeSameType">
    <xsd:sequence>
        <xsd:element name="pattern" type="conditionPatternType"
        minOccurs="1" maxOccurs="1" />
        <xsd:element name="exclude" type="conditionPatternType"
        minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="polls" type="positiveIntegerType" use="required" />
</xsd:complexType>

<xsd:complexType name="pollIntervalType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:int">
            <xsd:attribute name="units" type="timeUnitsType"
            use="optional" default="minutes" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

```

```

    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="batchType">
  <xsd:attribute name="maxSize" type="positiveIntegerType" use="required"/>
</xsd:complexType>

<xsd:simpleType name="timeUnitsType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="seconds" />
    <xsd:enumeration value="minutes" />
    <xsd:enumeration value="hours" />
    <xsd:enumeration value="days" />
    <xsd:enumeration value="weeks" />
    <xsd:enumeration value="months" />
    <xsd:enumeration value="years" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="monitorTasksType">
  <xsd:sequence>
    <xsd:element name="task" type="monitorTaskType"
      minOccurs="1" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="monitorTaskType">
  <xsd:sequence>
    <xsd:element name="name" type="monitorTaskNameType"
      minOccurs="1" maxOccurs="1" />
    <xsd:element name="description" type="xsd:string"
      minOccurs="0" maxOccurs="1" />
    <xsd:element name="transfer" type="transferTaskType"
      minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="transferTaskType">
  <xsd:sequence>
    <xsd:element name="request" type="transferRequestType"
      minOccurs="1" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="resourceIdType">
  <xsd:attribute name="id" type="xsd:string" use="optional" />
</xsd:complexType>

<xsd:simpleType name="resourceIdAttrType">
  <xsd:restriction base="xsd:string"></xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="monitorNameType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="^[^\*]*" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="agentNameType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="^[._0-9A-Z]*" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="monitorTaskNameType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value=".*" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="defaultVariablesType">
  <xsd:sequence>
    <xsd:element name="variable" type="variableType"
      maxOccurs="unbounded" minOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="variableType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="key" type="xsd:string" use="required" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```



```

    </xsd:simpleContent>
  </xsd:complexType>
</xsd:schema>

```

了解创建监视器消息

以下描述了创建监视器消息中使用的元素和属性：

元素描述

<monitor>

包含取消正在进行的文件传输所需的所有元素的组元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 提供的该元素的版本，指定该值。

<name>

监视器的名称，在监视器代理内是唯一的。

<description>

监视器的描述（当前未使用）。

<pollInterval>

每次针对触发器条件检查资源的时间间隔。

属性	描述
单元	指定轮询时间间隔的时间单位。有效值包括： • 秒 • 分钟 • 小时 • days • weeks • months • years

<agent>

与监视器相关联的代理的名称。

<resources>

组元素，包含用于指定要监视的资源元素。

<directory>

标准路径，用于指定要监视的监视器代理机器上的目录。

属性	描述
recursionLevel	除指定目录外，要监视的子目录数。
标识	资源的唯一标识。

<queue>

队列名称，用于指定要在监视代理的队列管理器上监视的队列。

<triggerMatch>

组元素，包含的元素用于指定要与受监视资源相比较的触发器条件。

<conditions>

组元素，包含的元素用于指定要与受监视资源相比较的条件的类型。

<allOf>
谓词，用于指定必须满足包含的所有条件。

<anyOf>
谓词，用于指定必须满足包含的任意条件。

<condition>
定义将促成整个监视器触发器条件的比较条件。

<name>
条件的名称。

<resource>
标识要与条件相比较的资源定义。

属性	描述
标识	资源的唯一标识。

如果正在监视的资源是一个目录，那么必须在条件中指定以下三个元素中的一个：

- fileMatch
- fileNoMatch
- fileSize

如果监视中的资源是队列，那么必须在条件中指定以下两个元素中的一个：

- queueNotEmpty
- completeGroups

<fileMatch>
文件名匹配条件的组元素。

<pattern>
指定文件名匹配模式。资源上的文件必须与模式相匹配才能满足条件。缺省模式为 *（任何文件都将匹配）。

<fileNoMatch>
逆文件名匹配条件的组元素。

<pattern>
指定逆文件名匹配模式。如果受监视资源上的所有文件均不匹配，那么满足条件。缺省模式为 *（缺少任何文件都将匹配）。

<fileSize>
文件大小比较的组元素。

<compare>
指定文件大小比较。该值必须是非负整数。

属性	描述
操作员	要使用的比较运算符。仅支持> = '。
单元	指定文件大小单位，可以是以下其中一项： • B - 字节 • KB - 千字节 • MB - 兆字节 • GB - 千兆字节 单位值不区分大小写，因此 mb' 与 MB' 作用相同。

<pattern>
要匹配的文件名模式。缺省值为 *（任何文件都将匹配）。

<queueNotEmpty>

只能在资源是队列时指定此属性。指定要触发监视器，队列上必须有消息。

<completeGroups>

只能在资源是队列时指定此属性。指定要触发监视器，队列上必须有完整的一组消息。针对队列上每个完整的组执行一项传输任务。

<reply>

可选元素，用于为异步请求指定应答队列。

属性	描述
QMGR	队列管理器名称。

<tasks>

组元素，包含的元素可指定在满足监视器触发器条件时要调用的任务。

<task>

组元素，定义在满足触发器条件时监视器将调用的个别任务。当前只能指定一项任务。

<name>

任务的名称。接受任何字母数字字符。

<description>

任务的描述。允许任何文本值。

<transfer>

用于定义传输任务的组元素。

<request>

用于定义任务类型的组元素。这必须包含从 `FileTransfer.xsd` 模式定义继承的以下其中一个元素：

- [managedTransfer](#)
- [managedCall](#)

属性	描述
version	请求的版本（由 IBM MQ Managed File Transfer 提供）。这采用 n.mm 格式，其中 n 是主要发行版本，mm 是次要版本。例如，1.00。

<originator>

包含指定请求发起方的元素的组元素。

<hostName>

源文件所在系统的主机名。

<userID>

发起文件传输的用户标识。

<mqmdUserID>

可选。在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<job>

包含作业信息的组元素。

<jobName>

指定逻辑作业标识。

<defaultVariables>

包含一个或多个 `variable` 元素的组元素。监视队列时，将在变量替换中使用这些变量。有关变量替换的更多信息，请参阅第 239 页的『[使用变量替换定制 MFT 任务](#)』。

<variable>

元素，包含与 `key` 属性提供的键相关联的值。

属性	描述
key	缺省变量的名称。

了解删除监视器消息

以下描述了删除监视器消息中使用的元素和属性：

元素描述

<deleteMonitor>

组元素，包含停止和删除监视器所需的所有元素。

属性	描述
version	按照 IBM MQ Managed File Transfer 提供的该元素的版本，指定该值。

<name>

要删除的监视器的名称。

<originator>

包含指定请求发起方的元素的组元素。

<hostName>

源文件所在系统的主机名。

<userID>

发起文件传输的用户标识。

<mqmdUserID>

可选。在消息描述符 (MQMD) 中提供的 WebSphere MQ 用户标识。

<reply>

指定为请求生成的临时应答队列的名称。队列名称由 `command.properties` 配置文件中的 `dynamicQueuePrefix` 键来定义。如果未指定，那么队列名称的缺省值为 WMQFTE。

属性	描述
QMGR	在其上生成临时动态队列以接收应答的命令队列管理器的名称。

示例

针对以下每项监视器请求提供了符合该模式的 XML 消息的示例：

- [创建监视器](#)
- [删除监视器](#)

相关参考

[第 867 页的『监视器请求消息示例』](#)

可置于代理命令队列上的消息示例（用于请求代理来创建或删除监视器）。

[第 647 页的『代理状态消息格式』](#)

在创建或启动代理时，该代理会向其协调队列管理器上的 `SYSTEM.FTE` 主题（在 `SYSTEM.FTE/Agents/agent name` 主题上）发布其详细信息。

[第 842 页的『文件传输请求消息格式』](#)

文件传输由抵达代理命令队列的 XML 消息启动，通常是用户发出文件传输命令或使用 WebSphere MQ Explorer 插件的结果。The transfer request XML must conform to the `FileTransfer.xsd` schema and have the `<request>` element as the root element. `FileTransfer.xsd` 模式文档位于 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。`FileTransfer.xsd` 模式导入相同目录中的 `fteutils.xsd`。

[第 658 页的『文件传输状态消息格式』](#)

将消息发布到协调队列管理器中以指示传输集中每个文件的传输状态。每次由代理处理文件传输的请求时，都会将事务消息发布到协调队列管理器中（在其 `SYSTEM.FTE/Transfers/agent_name/transfer ID` 主题中），它符合 `TransferStatus.xsd` XML 模式。`TransferStatus.xsd` 文件位于 WMQMFT 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

[第 662 页的『文件传输日志消息格式』](#)

文件传输日志消息发布在主题字符串为 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题内。这些消息符合模式 `TransferLog.xsd`，该模式位于 `IBM MQ Managed File Transfer` 安装的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目录中。

第 686 页的『已安排的传输日志消息格式』

代理每次处理已调度文件传输的请求时，都会向协调队列管理器发布一条调度日志消息（在其 `SYSTEM.FTE/Log/agent name/schedule ID` 主题中）。该消息符合 `ScheduleLog.xsd` XML 模式。

第 871 页的『针对安全性的消息格式』

本主题描述发布到协调队列管理器且与安全性相关的消息。

监视器请求消息示例

可置于代理命令队列上的消息示例（用于请求代理来创建或删除监视器）。

创建监视器请求

```
<?xml version="1.0" encoding="UTF-8"?>
<monitor:monitor xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
  version="4.00"
  xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./
Monitor.xsd">
  <name>EXAMPLEMONITOR</name>
  <pollInterval>1</pollInterval>
  <agent>US2.BINDINGS.FILE</agent>
  <resources>
    <directory recursionLevel="0">/srv/nfs/incoming</directory>
  </resources>
  <triggerMatch>
    <conditions>
      <allOf>
        <condition>
          <fileMatch>
            <pattern>*.completed</pattern>
          </fileMatch>
        </condition>
      </allOf>
    </conditions>
  </triggerMatch>
  <reply QMGR="US2.BINDINGS">WMQFTE.4D400F8B20003702</reply>
  <tasks>
    <task>
      <name/>
      <transfer>
        <request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
          version="4.00"
          xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
          <managedTransfer>
            <originator>
              <hostName>example.com.</hostName>
              <userID>mqm</userID>
            </originator>
            <sourceAgent QMgr="US2.BINDINGS" agent="US2.BINDINGS.FILE"/>
            <destinationAgent QMgr="US2.BINDINGS" agent="US2.BINDINGS.FILE"/>
            <transferSet>
              <item checksumMethod="MD5" mode="binary">
                <source disposition="leave" recursive="false">
                  <file>/srv/nfs/incoming/*.txt</file>
                </source>
                <destination exist="error" type="directory">
                  <file>/srv/backup</file>
                </destination>
              </item>
            </transferSet>
          </managedTransfer>
        </request>
      </transfer>
    </task>
  </tasks>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
  </originator>
</monitor:monitor>
```

删除监视器请求

```
<?xml version="1.0" encoding="UTF-8"?>
<monitor:deleteMonitor xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
    version="4.00"
    xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./
Monitor.xsd">
  <name>EXAMPLEMONITOR</name>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
  </originator>
  <reply QMGR="QM_JUPITER">WMQFTE.4D400F8B20003705</reply>
</monitor:deleteMonitor>
```

相关参考

第 859 页的『监视器请求消息格式』

当适合的 XML 消息到达代理的命令队列时，将创建资源监视器，通常是由于发出 `fteCreateMonitor` 命令或使用 WebSphere MQ Explorer 接口。

Ping 代理请求消息格式

您可以通过发出 **`ftePingAgent`** 命令或将 XML 消息放置在代理命令队列中 ping 代理。ping 代理请求 XML 必须符合 `PingAgent.xsd` 模式。安装 IBM MQ Managed File Transfer 后，您可以在以下目录中找到 `PingAgent.xsd` 模式文件：`MQ_INSTALLATION_PATH/mqft/samples/schema`。`PingAgent.xsd` 模式导入相同目录中的 `fteutils.xsd`。

当代理在其命令队列上收到 ping 代理请求消息时，如果代理处于活动状态，那么它会向将 ping 代理请求消息放置在命令队列上的命令或应用程序返回一条 XML 响应消息。代理的响应消息采用 `Reply.xsd` 定义的格式。有关该格式的更多信息，请参阅第 870 页的『应答消息格式』。

模式

以下模式描述了在 ping 代理请求 XML 消息中有效的元素。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema"
    xmlns="https://www.ibm.com/xmlns/wmqfte/7.0.1/PingAgent"
    targetNamespace="https://www.ibm.com/xmlns/wmqfte/7.0.1/PingAgent">

  <xsd:include schemaLocation="fteutils.xsd"/>

  <xsd:element name="pingAgent">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="originator" type="origRequestType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="agent" type="agentType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="reply" type="replyType" maxOccurs="1" minOccurs="0" />
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required" />
    </xsd:complexType>
  </xsd:element>

</xsd:schema>
```

了解 ping 代理请求消息。

下表中描述了在 ping 代理请求消息中使用的元素和属性：

<pingAgent>

包含指定 ping 代理请求所需的所有元素的组元素。

<originator>

包含指定 ping 请求的发起方所需的所有元素的组元素。

<hostName>

发起请求的主机的名称。

<userID>

请求的发起方的用户名。

<mqmdUserID>

请求的发起方的 MQMD 用户名。

<agent>

ping 的代理。

属性	描述
代理程序	必需。代理的名称。
QMgr	可选。代理连接到的队列管理器。

<reply>

要向其发送应答消息的代理的队列名称。

属性	描述
QMGR	必需。应答队列所在的队列管理器的名称。

示例

该示例显示了发送给代理 AGENT_JUPITER 的 ping 代理消息。如果 AGENT_JUPITER 处于活动状态，并且能够处理代理请求，那么它会向 QM_JUPITER 上的队列 WMQFTE.4D400F8B20003708 发送一条响应消息。

```
<?xml version="1.0" encoding="UTF-8"?>
<ping:pingAgent xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:ping="https://www.ibm.com/xmlns/wmqfte/7.0.1/PingAgent"
  version="4.00">
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
  </originator>
  <agent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
  <reply QMGR="QM_JUPITER">WMQFTE.4D400F8B20003708</reply>
</ping:pingAgent>
```

应答消息格式

代理在其代理命令队列中收到 XML 消息时，如果需要响应，那么代理会向原始消息中定义的应答队列发送一条 XML 应答消息。应答 XML 符合 Reply.xsd 模式。Reply.xsd 模式文档位于 MQ_INSTALLATION_PATH/mqft/samples/schema 目录中。Reply.xsd 模式将导入位于相同目录中的 fteutils.xsd。

模式

以下模式描述哪些元素在应答 XML 消息中有效。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="TransferLog.xsd"/>
  <xsd:element name="reply">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="transferSet" type="transferSetType" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="status" type="statusType" minOccurs="1" maxOccurs="1" />
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="ID" type="IDType" use="required"/>
      <xsd:attribute name="detailedReplyMessagesDisabled" type="xsd:boolean"
use="optional"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```


了解应答消息

以下列表中描述了应答消息中使用的元素和属性：

<reply>
包含用于指定应答信息的元素的元素。

属性	描述
标识	应答的标识。
version	应答消息格式的版本。
detailedReplyMessagesDisabled	关于代理禁用了详细回复（enableDetailedReplyMessages 代理属性设置为 false）的通知。

<transferSet>
指定请求传输的文件的传输结果信息。 有关更多信息，请参阅第 662 页的『文件传输日志消息格式』。

<status>
请求代理执行的操作的状态。

属性	描述
resultCode	从代理所执行操作返回的结果代码。

<supplement>
关于请求代理执行的操作的更多响应信息。

示例

以下部分是应答消息示例：

```
<reply version="1.00"          xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
                                xsi:noNamespaceSchemaLocation="Reply.xsd"
                                ID="0102020300000000000000000000000000000000000000000000000000000000">
  <status resultCode="65">
    <supplement>Additional reply information</supplement>
  </status>
</reply>
```

IBM MQ Managed File Transfer 诊断消息

诊断消息在此处按数字顺序提供，并根据它们源自的 Managed File Transfer 部分进行分组。
有关这些消息的详细信息，请参阅 IBM Documentation: https://www.ibm.com/docs/SSFKSJ_8.0.0/com.ibm.wmqfte.doc/messages_main.html

声明

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或默示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务的操作，由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以以书面形式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

知识产权许可
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 063-8506 Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区： International Business Machines Corporation “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗示的）保证，包括但不限于暗示的有关非侵权，适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗示的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本出版物中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：(i) 允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及 (ii) 允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation
软件互操作性协调员，部门 49XA
北纬 3605 号公路
罗切斯特，明尼苏达州 55901
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中 获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含日常商业运作所使用的数据和报表的示例。为了尽可能全面地说明这些数据和报表，这些示例包括个人、公司、品牌和产品的名称。所有这些名称都是虚构的，如与实际商业企业所使用的名称和地址有任何雷同，纯属巧合。

版权许可：

本信息包含源语言形式的样本应用程序，用以阐明在不同操作平台上的编程技术。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或默示这些程序的可靠性、可维护性或功能。

如果您正在查看本信息的软拷贝，图片和彩色图例可能无法显示。

编程接口信息

编程接口信息 (如果提供) 旨在帮助您创建用于此程序的应用软件。

本书包含有关允许客户编写程序以获取 WebSphere MQ 服务的预期编程接口的信息。

但是，该信息还可能包含诊断、修改和调优信息。提供诊断、修改和调优信息是为了帮助您调试您的应用程序软件。

要点: 请勿将此诊断，修改和调整信息用作编程接口，因为它可能会发生更改。

商标

IBM IBM 徽标 ibm.com 是 IBM Corporation 在全球许多管辖区域的商标。当前的 IBM 商标列表可从 Web 上的“Copyright and trademark information”www.ibm.com/legal/copytrade.shtml 获取。其他产品和服务名称可能是 IBM 或其他公司的商标。

Microsoft 和 Windows 是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

UNIX 是 Open Group 在美国和其他国家或地区的注册商标。

Linux 是 Linus Torvalds 在美国和/或其他国家或地区的商标。

此产品包含由 Eclipse 项目 (<http://www.eclipse.org/>) 开发的软件。

Java 和所有基于 Java 的商标和徽标是 Oracle 和/或其附属公司的商标或注册商标。



部件号:

(1P) P/N: