

9.0

Managed File Transfer

IBM

注

在使用本资料及其支持的产品之前，请阅读第 815 页的『声明』中的信息。

本版本适用于 IBM® MQ V 9 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.

内容

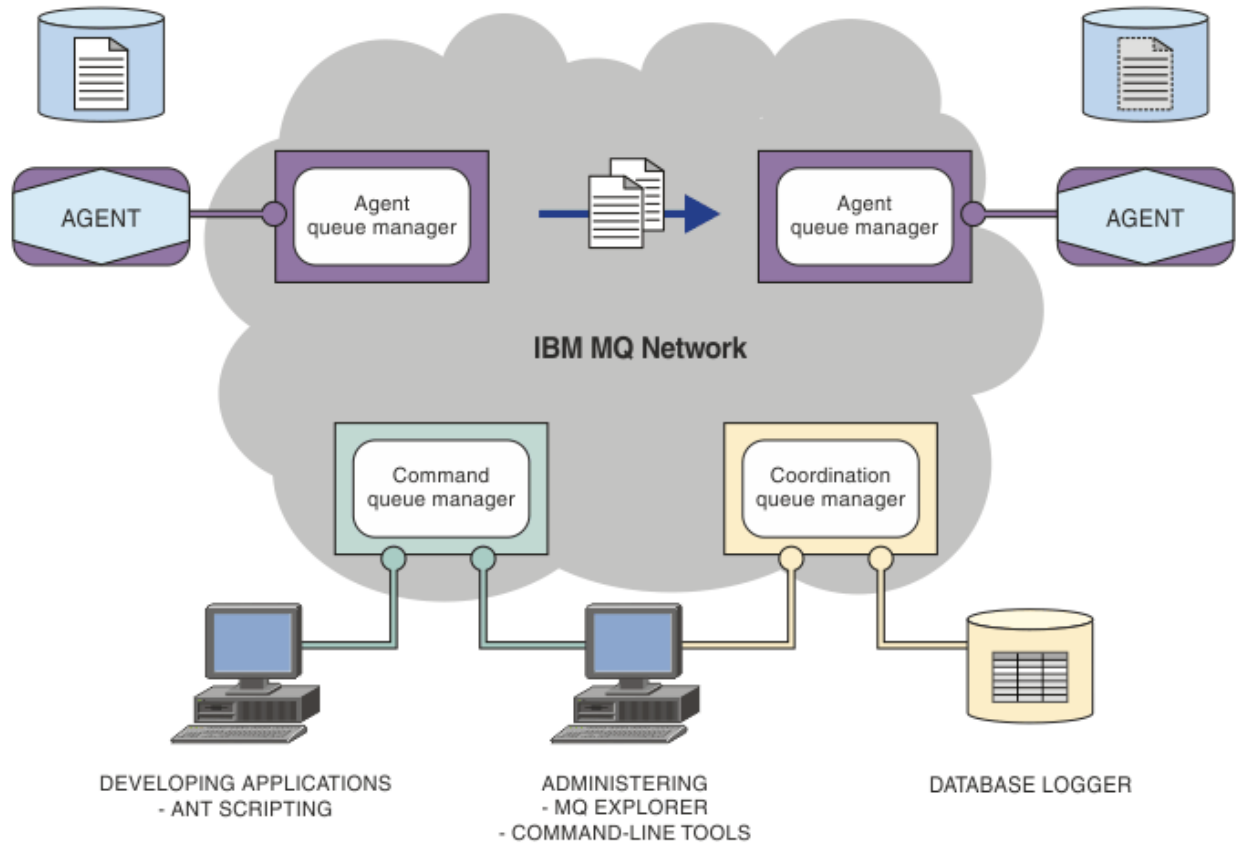
Managed File Transfer.....	5
MFT 概述.....	6
MFT 简介.....	7
MFT 如何使用 IBM MQ?	8
MFT 拓扑概述.....	9
MFT REST API 概述.....	10
IBM MQ 9.0 中的 MFT 中新增和更改了哪些内容?.....	10
MFT 的迁移注意事项.....	12
MFT 场景概述.....	25
MFT 常见拓扑.....	25
配置基本服务器.....	28
保护 MFT.....	35
MFT 和 IBM MQ 连接认证.....	36
MFT 沙箱.....	39
为 MFT 配置 SSL 或 TLS 加密.....	44
使用通道认证以客户机方式连接到队列管理器.....	45
在 Connect:Direct 网桥代理与 Connect:Direct 节点之间配置 SSL 或 TLS.....	46
配置 MFT.....	48
多平台上的 MFT 配置选项.....	49
z/OS 上的 MFT 配置选项.....	50
配置 Redistributable Managed File Transfer Agent.....	50
创建 MFT 代理或记录器命令数据集.....	53
配置 Managed File Transfer for z/OS.....	54
在 IBM i 上配置 MFT.....	79
为首次使用配置 MFT.....	80
配置 MFT 记录器.....	89
配置 Connect:Direct 网桥.....	118
通过 MSCS 配置 MFT 代理.....	126
管理 MFT.....	128
启动 MFT 代理.....	128
启动新的文件传输.....	132
创建调度的文件传输.....	135
处理暂挂文件传输.....	136
触发文件传输.....	136
监视正在进行的文件传输.....	137
查看传输日志中文件传输的状态.....	139
监视 MFT 资源.....	141
使用文件传输模板.....	162
将数据从文件传输到消息.....	163
将数据从消息传输到文件.....	171
列出 MFT 代理.....	176
停止 MFT 代理.....	176
协议网桥.....	177
Connect:Direct 网桥.....	191
从 IBM Integration Bus 使用 MFT.....	204
MFT 恢复和重新启动.....	204
恢复中文件传输的超时选项.....	205
为 MFT 开发应用程序.....	207
指定要使用 MFT 运行的程序.....	207
将 Apache Ant 与 MFT 结合使用.....	209
使用用户出口定制 MFT.....	213
通过将消息放置在代理命令队列中来控制 MFT.....	225

对 MFT 进行故障诊断.....	226
MFT 一般故障诊断.....	226
对 Connect:Direct 网桥进行故障诊断.....	286
MFT 引用.....	291
MFT 安全性参考.....	291
MFT 命令和进程与队列管理器的连接关系.....	304
MFT 命令.....	310
MFT 配置参考.....	448
MFT 管理参考.....	521
MFT 开发应用程序参考.....	723
MFT 诊断消息.....	813
声明.....	815
编程接口信息.....	816
商标.....	816

Managed File Transfer

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 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。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 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 Managed File Transfer 活动。

根据您的操作系统和总体设置，Managed File Transfer 可以作为四个不同的选项进行安装。这些选项为 Managed File Transfer Agent、Managed File Transfer Logger、Managed File Transfer Service 或 Managed File Transfer Tools。有关更多信息，请参阅 [Managed File Transfer 产品选项](#)。

您可以使用 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 Explorer 创建新的文件传输。
 - 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 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。
- 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。
 - 从 Linux 或 Windows 平台上的 IBM MQ Explorer 查看正在进行的传输的状态。
 - 在 Linux 或 Windows 平台上使用 IBM MQ Explorer 检查已完成传输的状态。
 - 使用 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。

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 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 Managed File Transfer 可以利用现有 IBM MQ 基础结构，包括 IBM MQ Internet Pass-Thru 和 IBM Integration Bus。

Managed File Transfer 与许多其他 IBM 产品集成：

IBM Integration Bus

作为 IBM Integration Bus 流的一部分，通过 Managed File Transfer 传输的进程文件。有关更多信息，请参阅第 204 页的『[从 IBM Integration Bus 使用 MFT](#)』。

IBM Sterling Connect:Direct

使用 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 191 页的『[Connect:Direct 网桥](#)』。

IBM Tivoli Composite Application Manager

IBM Tivoli Composite Application Manager 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『[MFT 拓扑概述](#)』

有关 Managed File Transfer 代理如何连接 IBM MQ 网络中的协调队列管理器的概述。

第 8 页的『[MFT 如何使用 IBM MQ?](#)』

Managed File Transfer 可以使用多种方式与 IBM MQ 进行交互。

相关信息

[Managed File Transfer 产品选项](#)

MFT 概述

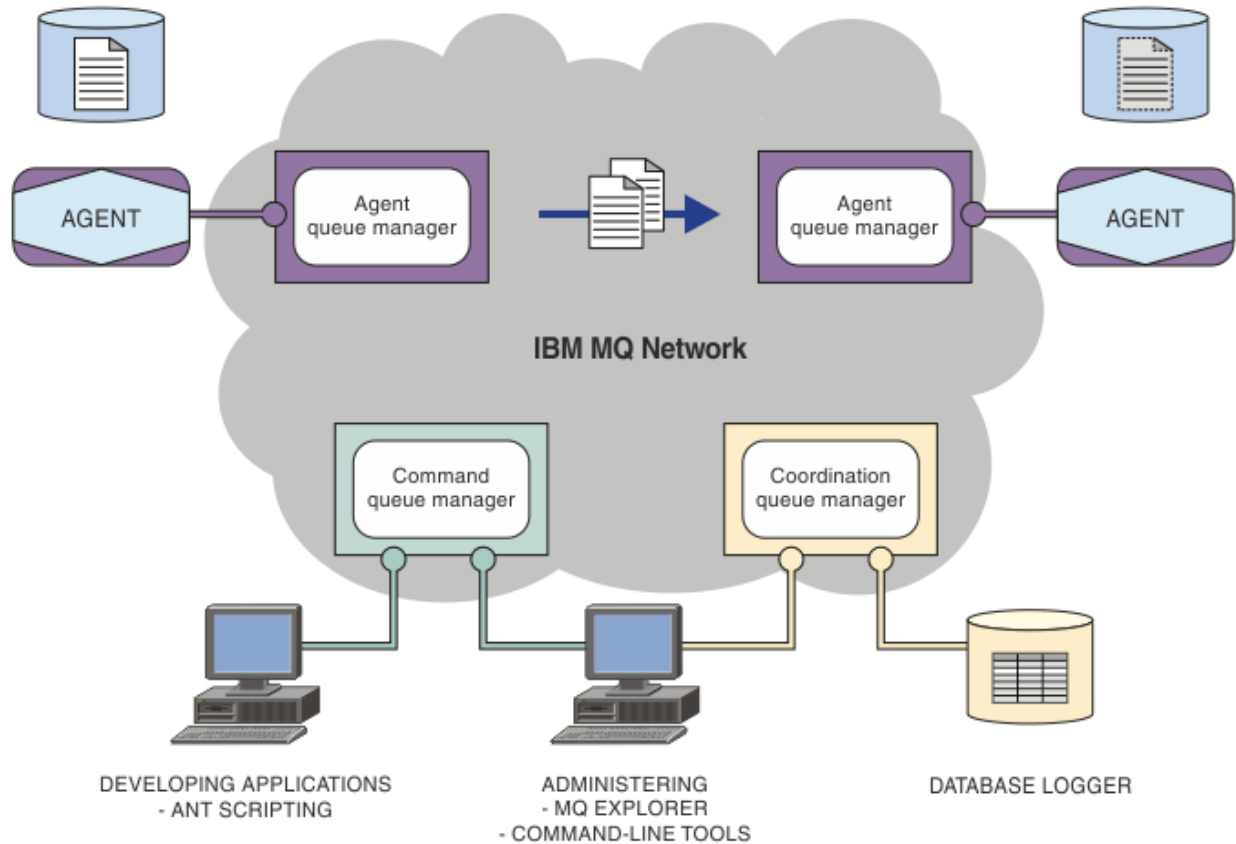
本部分提供了您用于开始使用 Managed File Transfer 的介绍信息。

- [第 5 页的『Managed File Transfer』](#)
- [Managed File Transfer 产品选项](#)
- [第 9 页的『MFT 拓扑概述』](#)
- [第 10 页的『IBM MQ 9.0 中的 MFT 中新增和更改了哪些内容?』](#)

Managed File Transfer

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 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。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 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 Managed File Transfer 活动。

根据您的操作系统和总体设置，Managed File Transfer 可以作为四个不同的选项进行安装。这些选项为 Managed File Transfer Agent、Managed File Transfer Logger、Managed File Transfer Service 或 Managed File Transfer Tools。有关更多信息，请参阅 [Managed File Transfer 产品选项](#)。

您可以使用 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 Explorer 创建新的文件传输。
 - 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 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。

- 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。
 - 从 Linux 或 Windows 平台上的 IBM MQ Explorer 查看正在进行的传输的状态。
 - 在 Linux 或 Windows 平台上使用 IBM MQ Explorer 检查已完成传输的状态。
 - 使用 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。

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 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 Managed File Transfer 可以利用现有 IBM MQ 基础结构，包括 IBM MQ Internet Pass-Thru 和 IBM Integration Bus。

Managed File Transfer 与许多其他 IBM 产品集成：

IBM Integration Bus

作为 IBM Integration Bus 流的一部分，通过 Managed File Transfer 传输的进程文件。有关更多信息，请参阅 [第 204 页的『从 IBM Integration Bus 使用 MFT』](#)。

IBM Sterling Connect:Direct

使用 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 191 页的『Connect:Direct 网桥』](#)。

IBM Tivoli Composite Application Manager

IBM Tivoli Composite Application Manager 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『MFT 拓扑概述』](#)

有关 Managed File Transfer 代理如何连接 IBM MQ 网络中的协调队列管理器的概述。

[第 8 页的『MFT 如何使用 IBM MQ?』](#)

Managed File Transfer 可以使用多种方式进行交互。

相关信息

[Managed File Transfer 产品选项](#)

MFT 如何使用 IBM MQ?

Managed File Transfer 可以使用多种方式进行交互。

- 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。
- Managed File Transfer 代理会使用多个 IBM MQ 队列。有关更多信息，请参阅 [第 519 页的『MFT 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 225 页的『[通过将消息放置在代理命令队列中来控制 MFT](#)』。

- 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指定为协调队列管理器的 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 517 页的『[SYSTEM.FTE 主题](#)』。
- 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。

MFT 拓扑概述

有关 Managed File Transfer 代理如何连接 IBM MQ 网络中的协调队列管理器的概述。

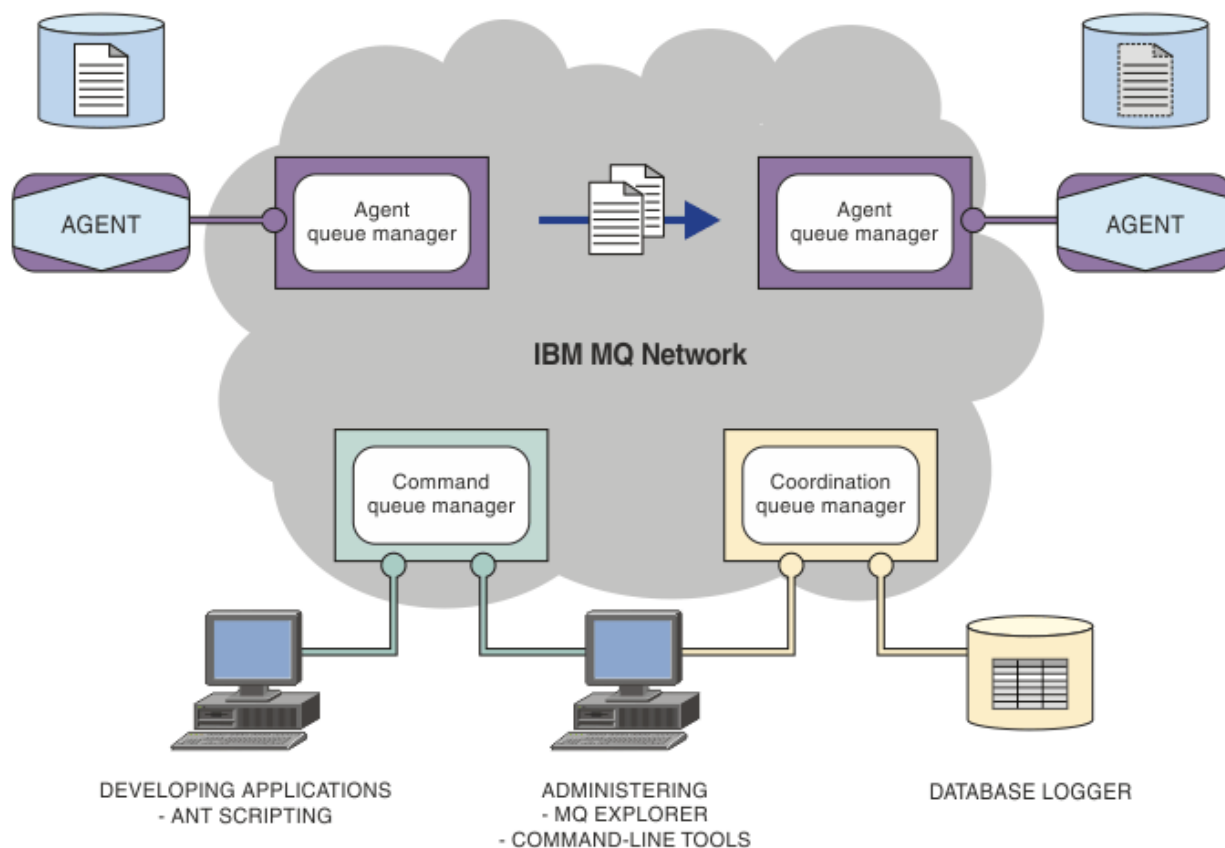
Managed File Transfer 代理发送并接收传输的文件。每个代理在其关联队列管理器上都有自己的队列集，并且代理将以绑定或客户机方式连接到其队列管理器。代理还可以使用协调队列管理器作为其队列管理器。

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 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 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 Managed File Transfer 命令时要连接到的队列管理器。



相关概念

第 5 页的『[Managed File Transfer](#)』

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 8 页的『[MFT 如何使用 IBM MQ?](#)』

Managed File Transfer 可以使用多种方式与 IBM MQ 进行交互。

第 25 页的『[MFT 场景概述](#)』

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

V 9.0.5 MFT REST API 概述

针对 Managed File Transfer 的 REST API 增强功能概述

从 IBM MQ 9.0.5 开始，REST API 添加了对某些 Managed File Transfer 命令（包括列出传输）的支持以及有关文件传输代理的详细信息。

请参阅 [REST API 的更多增强功能](#)，以获取更多信息。

IBM MQ 9.0 中的 MFT 中新增和更改了哪些内容?

指向有关 IBM MQ 9.0 中的主要新增功能和已更改的 Managed File Transfer 功能的信息的链接。

IBM MQ 9.0.0

> V 9.0.0

- > V 9.0.0 [Managed File Transfer 协议网桥代理日志记录增强功能](#)
- > V 9.0.0 [移除 Managed File Transfer Web 网关支持](#)

IBM MQ 9.0.x Continuous Delivery 发行版

CD

- Windows V 9.0.1 Linux [Managed File Transfer 已停止传输的可配置恢复超时](#)
- Windows V 9.0.1 Linux [Managed File Transfer Agent 可重新分发包](#)
- Windows V 9.0.1 Linux [对远程 Managed File Transfer 代理程序的 **DISPLAY CHSTATUS** 输出值的更改](#)
- Windows V 9.0.1 Linux [更改了传到 Managed File Transfer 中的文件传输的消息](#)
- V 9.0.1 [新增了 MFT 代理属性 `additionalWildcardSandboxChecking`](#)
- Windows V 9.0.2 Linux [改进了 IBM MQ Explorer MFT 插件创建的预订的命名方式](#)
- Windows V 9.0.2 Linux [当上次报告的状态为 UNKNOWN 时，改进了有关 MFT 代理程序的信息](#)
- V 9.0.2 z/OS MQ Adv. VUE [用于 z/OS 上的 MFT 代理程序的新代理程序属性 `adminGroup`](#)
- Windows z/OS Linux V 9.0.3 [Managed File Transfer 资源监视器事件日志记录](#)
- z/OS 产品标识 (PID) 记录中的更改
- z/OS V 9.0.3 MQ Adv. VUE [代理与 IBM MQ Advanced for z/OS, Value Unit Edition 的远程连接](#)
- Windows z/OS Linux [更改 Managed File Transfer **fteCleanAgent** 命令的行为](#)
- Windows Linux [在 Managed File Transfer 安装属性中更正了 `messagePublicationFormat=mixed` 的行为](#)
- Windows z/OS Linux V 9.0.3 [对 Managed File Transfer 启用 MQCSP 认证](#)
- Windows Linux AIX V 9.0.4 [能够以客户机方式运行 Managed File Transfer Logger](#)
- V 9.0.5 Multi [第 10 页的『MFT REST API 概述』](#)

IBM MQ 9.0.0.x Long Term Support

LTS

- V 9.0.0.1 [**fteMigrateAgent** 命令更改](#)
- V 9.0.0.1 [新增了 MFT 代理属性 `addCommandPathToSandbox`](#)
- V 9.0.0.1 [新增了 MFT 代理属性 `additionalWildcardSandboxChecking`](#)
- z/OS V 9.0.0.1 [新增了用于 z/OS 上的 MFT 代理的代理属性 `adminGroup`](#)
- V 9.0.0.2 [更改 Managed File Transfer **fteCleanAgent** 命令的行为](#)
- V 9.0.0.2 [在 Managed File Transfer 中启用 MQCSP 认证](#)
- V 9.0.0.2 [Managed File Transfer `installation.properties` 文件的新 `messagePublicationFormat` 属性](#)
- V 9.0.0.2 [为“消息到文件”传输复原的传输标识和 `groupId` 属性值的比较检查](#)
- V 9.0.5 [能够在 `fteSetAgentTraceLevel` 和 `fteSetLoggerTracelevel` 命令中使用冒号分隔的列表](#)

相关信息

V 9.0.0 IBM MQ 9.0.0 中新增及更改的内容

CD IBM MQ 9.0.x Continuous Delivery 中的新增功能和已更改的功能

LTS IBM MQ 9.0.x 长期支持中的更改内容

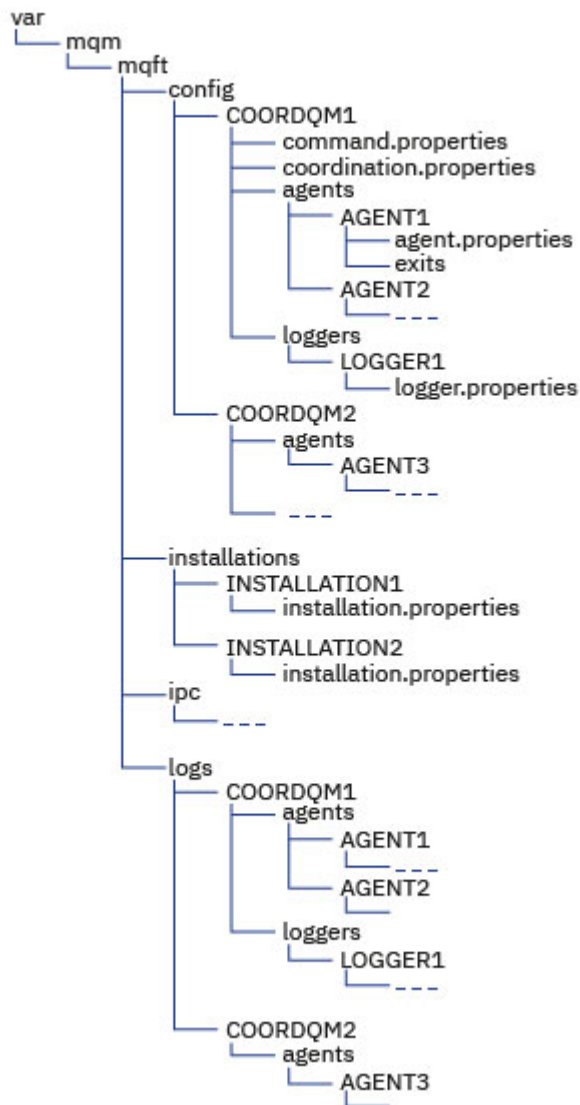
MFT 的迁移注意事项

如果计划迁移 Managed File Transfer，请查看以下信息。

在 MFT 中安装后直接进行配置布局

该图显示安装后的配置布局 (如 IBM MQ 中所示)。

IBM MQ Managed File Transfer configuration



缺省配置目录:

信息拆分为四个独立的子目录: 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) \IBM\WebSphere MQ
 - 64 位: C:\Program Files\IBM\MQ

配置子目录如下所示:

- `MQ_DATA_PATH/mqft/config` 目录包含对 Managed File Transfer 进程为只读的配置部分。例如, `agent.properties` 和 `command.properties`。
- `MQ_DATA_PATH/mqft/installations` 目录包含每个安装的配置信息。此目录的内容等同于 `wmqfte.properties` 文件的内容。
- `MQ_DATA_PATH/mqft/ipc` 目录包含内部用于在 Managed File Transfer 组件之间进行通信的 IPC 资源。仅适用于 UNIX 和 Linux 系统。
- `MQ_DATA_PATH/mqft/logs` 目录包含由 Managed File Transfer 进程写入的配置部分。例如, 跟踪信息和日志文件。

installation.properties 文件

Linux **UNIX** 在 UNIX and Linux 系统上, 缺省位置为 `MQ_DATA_PATH/mqft/installations/installation_name`

Windows 在 Windows 上, 缺省位置为 `MQ_DATA_PATH\mqft\installations\installation_name`

logger.properties 文件

此文件合并了独立文件记录器, 独立数据库记录器和 Java EE 数据库记录器的属性信息。
缺省位置是 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name`。

安全性更改

对于 IBM WebSphere MQ 7.5 或更高版本, 只有作为管理员 (mqm 组的成员) 的用户才能运行下列 **fte** 命令:

- [第 318 页的『fteChangeDefaultConfigurationOptions』](#)
- [第 323 页的『fteCreateAgent \(创建 MFT 代理\)』](#)
- [第 327 页的『fteCreateBridgeAgent \(创建并配置 MFT 协议网桥代理\)』](#)
- [第 335 页的『fteCreateCDAgent \(创建 Connect:Direct 网桥代理\)』](#)
- [第 340 页的『fteCreate 记录器 \(创建 MFT 文件或数据库记录器\)』](#)
- [第 386 页的『fteDeleteAgent』](#)
- [第 388 页的『fteDeleteLogger』](#)
- [第 403 页的『fteMigrate 代理: 将 FTE 7.0 代理迁移到 MQ 7.5 或更高版本』](#)
- [第 406 页的『fteMigrateConfigurationOptions: 将 FTE 7.0 配置迁移到 MQ 7.5 或更高版本』](#)
- [第 407 页的『fteMigrate 记录器: 将 FTE 7.0 数据库记录器迁移到 MQ 7.5 或更高版本』](#)
- [第 410 页的『fteModify 代理 \(将 MFT 代理作为 Windows 服务运行\)』](#)
- [第 411 页的『fteModify 记录器 \(将 MFT 记录器作为 Windows 服务运行\)』](#)
- [第 428 页的『fteSetupCommands: 创建 MFT command.properties 文件』](#)
- [第 430 页的『fteSetupCoordination』](#)

Multi 在 多平台上使用 IBM WebSphere MQ 7.5 或更高版本时, 只有运行代理进程的用户才能运行 **fteSetAgentTraceLevel** 命令。

V 9.0.2 **z/OS** 对于 z/OS, 针对 Continuous Delivery, 从 IBM MQ 9.0.2 开始, **fteSetAgentTraceLevel** 命令可由以下任一人员运行:

- 用于运行代理进程的统一用户标识。
- 代理属性 **adminGroup** 指定的组的成员。

V 9.0.0.1 这也适用于来自 IBM MQ 9.0.0 Fix Pack 1 的 Long Term Support 发行版。

V 9.0.2 **z/OS** 对于 z/OS，针对 Continuous Delivery，从 IBM MQ 9.0.2 开始，如果运行 **fteShowAgentDetails** 命令以显示有关本地代理的信息，那么该命令可由以下任一人员运行：

- 用于运行代理进程的统一用户标识。
- 代理属性 **adminGroup** 指定的组的成员。

V 9.0.0.1 这也适用于来自 IBM MQ 9.0.0 Fix Pack 1 的 Long Term Support 发行版。

有关更多信息，请参阅第 460 页的『MFT agent.properties 文件』中的 **adminGroup** 属性。

IBM MQ 8.0 中的安全性更改

如果您正在 IBM WebSphere MQ 7.0 上运行 Managed File Transfer 并迁移到 IBM MQ 8.0，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 8.0 中支持传递用户标识和密码信息。

commandPath 和代理沙箱

对于 IBM MQ 8.0 和更高版本，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 303 页的『**commandPath MFT 属性**』和第 460 页的『MFT agent.properties 文件』。

commandPath 和用户沙箱

对于 IBM MQ 8.0 和更高版本，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 303 页的『**commandPath MFT 属性**』和第 460 页的『MFT agent.properties 文件』。

将作为 Windows 服务运行的 MFT 代理程序从 IBM WebSphere MQ 7.5 迁移到 IBM MQ 9.0

在 IBM WebSphere MQ 7.5、IBM MQ 8.0 和 IBM MQ 9.0 之间，缺省 IBM MQ 安装路径在 Windows 平台上已发生更改。

如果队列管理器要从 IBM WebSphere MQ 7.5 迁移到 IBM MQ 8.0 或 IBM MQ 9.0，那么在与队列管理器相同系统上运行的所有应用程序都必须重新配置，以从新安装位置装入 IBM MQ 库。这包括作为 Windows 服务运行的任何 IBM MQ Managed File Transfer 代理程序。

执行以下过程：

1. 停止与 IBM WebSphere MQ 7.5 队列管理器关联的代理。请参阅[停止 MFT 代理](#)，以获取更多信息。例如，发出以下命令：

```
fteStopAgent <agent_name>
```

2. 修改代理以移除代理的服务定义。请参阅[将 MFT 代理作为 Windows 服务运行](#)，以获取更多信息。

例如，发出以下命令：

```
fteModifyAgent -agentName <agent_name>
```

3. 接下来，使用 **setmqm** 命令将代理队列管理器从 IBM WebSphere MQ 7.5 迁移到更高版本。有关 **setmqm** 命令的更多信息，请参阅 [将队列管理器与安装相关联](#)。
4. 最后，通过使用 **fteModifyAgent** 命令，修改代理程序以重新配置代理程序以再次作为 Windows 服务运行。

例如：

```
fteModifyAgent -agentName AGENT1 -s -su fteuser -sp ftepassword
```

将 IBM WebSphere MQ File Transfer Edition 7.0 代理迁移至 IBM WebSphere MQ 7.5 或更高版本

使用 **fteMigrateAgent** 命令将 IBM WebSphere MQ File Transfer Edition 7.0 代理迁移至 IBM WebSphere MQ 7.5 或更高版本。如果代理配置为作为 Windows 服务运行，那么必须完成本任务中的可选步骤。

开始之前

V 9.0.0 从 IBM MQ 9.0 开始，Managed File Transfer 不支持 Web 代理。如果尝试使用 **fteMigrateAgent** 命令将 Web 代理从较低版本迁移到 IBM MQ 9.0，那么会显示错误消息来说明不支持迁移 Web 代理。

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

IBM i 本主题中的信息不适用于 IBM i。如果您正在使用 IBM i，请在 IBM MQ 8.0 中使用 Managed File Transfer 之前将队列管理器迁移到 IBM MQ 8.0。

关于此任务

要迁移至 IBM WebSphere MQ 7.5 或更高版本，请首先停止代理，然后将队列管理器安装迁移至 IBM WebSphere MQ 7.5 或更高版本，此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 IBM WebSphere MQ 7.5 或更高版本的安装。使用 IBM WebSphere MQ 7.5 或更高版本的队列管理器启动已迁移代理。

如果代理是使用 IBM WebSphere MQ 绑定方式连接到其队列管理器，并且该队列管理器为 IBM WebSphere MQ 7.0.1 Fix Pack 6 或更高版本，那么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。

如果代理是使用 IBM WebSphere MQ 绑定方式连接到其队列管理器，并且该队列管理器为 IBM WebSphere MQ 7.0.1 Fix Pack 5 或更低版本，那么必须将此安装迁移至 IBM WebSphere MQ 7.0.1 Fix Pack 6 以允许并行迁移，或者必须将此安装直接迁移至 IBM WebSphere MQ 7.5 或更高版本。

如果代理作为 IBM WebSphere MQ 客户机通过网络连接到其队列管理器，那么只能通过完成步骤 [第 16 页的『7』](#) 将代理迁移到 IBM WebSphere MQ 7.5 或更高版本。

如果代理配置为 Windows 服务，那么必须运行带有 **-f** 参数的命令。有关更多信息，请参阅 [第 403 页的『fteMigrate 代理: 将 FTE 7.0 代理迁移到 MQ 7.5 或更高版本』](#)。

过程

1. 通过选择 MQ Server 和文件传输组件，随现有队列管理器安装一起安装 IBM WebSphere MQ 7.5 或更高版本。
2. 停止 IBM WebSphere MQ File Transfer Edition 7.0.x 代理程序。
3. 将队列管理器移至 IBM WebSphere MQ 7.5 或更高版本的安装。

4. 使用 **fteMigrateAgent** 命令将代理程序配置从 IBM WebSphere MQ File Transfer Edition 7.0.x 安装迁移到与 IBM WebSphere MQ 7.5 或更高版本集成的代理程序功能。
5. 可选: 如果将 IBM WebSphere MQ File Transfer Edition 7.0.x 代理程序配置为 Windows 服务, 请完成以下可选步骤:
 - a) 使用 **fteModifyAgent** 命令重新配置代理, 使其不再是 Windows 服务。
 - b) 从 IBM WebSphere MQ 7.5 或更高版本的安装使用 **fteModifyAgent** 命令, 将 IBM WebSphere MQ 7.5 或更高版本的代理程序配置为 Windows 服务。
6. 启动 IBM WebSphere MQ 7.5 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 7.5 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 7.0.x 安装迁移到与 IBM WebSphere MQ 7.5 或更高版本集成的代理程序功能。
 - c) 停止 IBM WebSphere MQ File Transfer Edition 7.0.x 代理程序。
 - d) 如果 IBM WebSphere MQ File Transfer Edition 7.0.x 代理程序配置为 Windows 服务, 请使用 IBM WebSphere MQ File Transfer Edition 7.0.x 安装中的 **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 IBM WebSphere MQ File Transfer Edition 7.0.x 代理程序配置为 Windows 服务, 请使用 IBM WebSphere MQ 7.5 或更高版本的安装中的 **fteModifyAgent** 命令将 IBM WebSphere MQ 7.5 或更高版本的代理程序配置为 Windows 服务。
 - f) 启动 IBM WebSphere MQ 7.5 或更高版本的代理。

将 FTE 7.0 数据库记录器迁移到 MQ 7.5 或更高版本

使用 **fteMigrateLogger** 命令将独立 IBM WebSphere MQ File Transfer Edition 7.0 数据库记录器迁移到 IBM WebSphere MQ 7.5 或更高版本。如果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。

关于此任务

要直接迁移到 IBM WebSphere MQ 7.5 或更高版本, 请首先停止记录器, 然后将此安装迁移到所需版本。当此迁移完成时, 使用 **fteMigrateLogger** 命令将数据库记录器配置迁移到 IBM MQ 的所需版本。

如果队列管理器为 IBM WebSphere MQ File Transfer Edition 7.0.1 Fix Pack 6 或更高版本, 那么还可以使用多个安装执行并行迁移, 如以下步骤中所详述。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到 IBM WebSphere MQ File Transfer Edition 7.0.1 Fix Pack 5 或更低版本的队列管理器, 那么必须首先将此安装迁移到 IBM WebSphere MQ File Transfer Edition 7.0.1 Fix Pack 6 以允许并行迁移, 或者必须将此安装直接迁移到 IBM WebSphere MQ 7.5 或更高版本, 此外还要选择文件传输组件。

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 407 页的『[fteMigrate 记录器: 将 FTE 7.0 数据库记录器迁移到 MQ 7.5 或更高版本](#)』。

过程


1. 与现有 IBM WebSphere MQ File Transfer Edition 7.0 安装一起安装 IBM WebSphere MQ 7.5 或更高版本, 选择 MQ Server 和 Managed File Transfer Service 组件。
2. 停止 IBM WebSphere MQ File Transfer Edition 7.0 数据库记录器。
3. 将队列管理器移至 IBM WebSphere MQ 7.5 或更高版本的安装。

4. 使用 **fteMigrateLogger** 命令将数据库记录器配置从 IBM WebSphere MQ File Transfer Edition 7.0 安装迁移到 IBM WebSphere MQ 7.5 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 IBM WebSphere MQ File Transfer Edition 或 Managed File Transfer 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，如果您正在使用 IBM WebSphere MQ File Transfer Edition 7.0.3 并正在迁移到 IBM MQ 8.0.0.0，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 8.0.0.0 中，并且位于 `MQ_INSTALLATION_PATH/mqft/sql` 目录中。IBM MQ 8.0.0.0 未随附超过 750 的升级；如果这些升级不在目录中，那么可以从 [APAR IT01841](#) 中指定的链接下载这些升级。

6. 可选：如果 IBM WebSphere MQ File Transfer Edition 7.0 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) 在 IBM WebSphere MQ 7.5 或更高版本的记录器中重新配置 Managed File Transfer，以便它是使用 **fteModifyLogger** 命令的 Windows 服务。
7. 在 IBM WebSphere MQ 7.5 或更高版本的数据库记录器中启动 Managed File Transfer。

结果

数据库记录器现已从 IBM WebSphere MQ File Transfer Edition 7.0 迁移到 IBM WebSphere MQ 7.5 或更高版本。

在 Windows、UNIX 或 Linux 上增大 Db2 上的日志数据库的页面大小

如果您的数据库是 Windows、UNIX 或 Linux 系统上的 Db2，并且您创建的日志数据库的页面大小小于 8 KB，那么在迁移到 IBM WebSphere MQ File Transfer Edition 7.0.3 或更高版本的表之前，必须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. 按照给定顺序针对下表运行 Db2 **set integrity** 命令, 其中完整性状态值为 **immediate** 和 **checked**:
- 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 117 页的『从独立数据库记录器迁移至 MFT 的 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 Java EE 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 JEE 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。

第 19 页的『将 Db2 on z/OS 上的数据库表迁移至 IBM MQ 8.0』

如果您的数据库是 z/OS 系统上的 Db2，那么必须完成以下步骤以从 IBM WebSphere MQ File Transfer Edition 7.0.3 迁移到 IBM WebSphere MQ File Transfer Edition 7.0.4，并从 IBM WebSphere MQ File Transfer Edition 7.0.4 迁移到 IBM MQ 8.0 中的 Managed File Transfer。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 Db2 格式。

将 Db2 on z/OS 上的数据库表迁移至 IBM MQ 8.0

如果您的数据库是 z/OS 系统上的 Db2，那么必须完成以下步骤以从 IBM WebSphere MQ File Transfer Edition 7.0.3 迁移到 IBM WebSphere MQ File Transfer Edition 7.0.4，并从 IBM WebSphere MQ File Transfer Edition 7.0.4 迁移到 IBM MQ 8.0 中的 Managed File Transfer。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 Db2 格式。

关于此任务

IBM MQ 8.0 中的 Managed File Transfer 支持传输非常大的文件，其中文件的大小大于可以存储在整数 (32 位) 数字中的大小。提供了一个使用 BIGINT (64 位) 数字的定义。您可以选择使用整数定义 (称为 `ftelog_tables_zos.sql`) 或 BIGINT 定义 (称为 `ftelog_tables_zos_bigint.sql`)，这些定义位于 `prod/mqf/sql` 目录中。

BIGINT 数据类型在 IBM WebSphere MQ File Transfer Edition 7.0.4 Fix Pack 3 中提供，在 IBM WebSphere MQ 7.5.0 Fix Pack 1 或更高版本中提供 Managed File Transfer。

要启用 BIGINT 数据类型，您必须使用 Db2 9.1 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. 您具有 IBM WebSphere MQ 7.0 中的现有表。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 IBM MQ 8.0 表。这允许您将数据从 FTELOG.*table* 复制到 FTE8.*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. 编辑现有记录器 JCL，以使用 IBM MQ 8.0 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，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 **FTE_SRC** 更改为现有模式 (FTELOG)，将 **FTE_DEST** 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 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 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. 在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 8.0.0 格式：

a) **fteMigrateConfigurationOptions**

b) **fteMigrateLogger**

这会将 `databaselogger.properties` 复制到 `logger.properties`。

11. 编辑现有记录器 JCL 以使用 8.0.0 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 Managed File Transfer for z/OS 代理程序迁移到更高版本

遵循以下指示信息将 IBM MQ Managed File Transfer for z/OS 代理程序迁移到更高版本。本主题使用 IBM MQ 8.0 作为初始版本，使用 IBM MQ 9.0 作为更高版本。

要点：这些指示信息假定以下几点：

- Managed File Transfer for z/OS (MFT) 拓扑的协调队列管理器和命令队列管理器在迁移期间不会更改。
- 未迁移代理队列管理器。在迁移之前和之后，代理队列管理器保持处于相同的 IBM MQ for z/OS 级别。
- 正在迁移的代理程序使用 JCL 进行管理。

步骤 1-初始设置

1. 将 IBM MQ Managed File Transfer for z/OS 9.0 安装到与 IBM MQ Managed File Transfer for z/OS 8.0 相同的 LPAR 上。

- 在 IBM MQ 9.0 中找到针对 IBM MQ MFT 的 SBFGCMDSDS PDSE。
有关 SBFGCMDSDS PDSE 的更多信息，请参阅 SBFGCMDSDS 库。
- 在此数据集中，编辑 BFGCOPY 成员以指定要用于管理代理程序的 PDSE 的名称。
- 提交 BFGCOPY 成员。这会将现有 SBFGCMDSDS PDSE 复制到具有步骤 第 22 页的『3』中指定的名称的新 PDSE 中。

步骤 2-修改新的 PDSE

- 在使用 BFGCOPY 成员创建的新 PDSE 中，编辑 BFGCUSTM 并更新以下变量：

变量	值
++library++	将此变量替换为新 PDSE 的名称。
armELEMENT	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。
armELEMENTYPE	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。
BFG_DATA	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。
BFG_GROUP_NAME	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。
BFG_JAVA_HOME	这需要设置为用于运行代理程序的 Java 运行时环境的 UNIX 系统服务 (USS) 路径。将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。
BFG_JVM_PROPERTIES	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。
BFG_PROD	此变量应该设置为 IBM MQ 9.0 安装中 Managed File Transfer for z/OS 的 USS 路径。
BFG_WTO	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。
CLEAN_AGENT_PROPS	将此项设置为空白。
connectionQMGr	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。
coordinationQMGr	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。
CREDENTIALS_PATH	将此项设置为空白。
DB_PROPS_PATH	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。
DLQ_HLQ	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。
FTE_CONFIG	将此项设置为空白。
MQ_HLQ	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。
MQ_PATH	此变量需要设置为与代理队列管理器关联的 IBM MQ 安装的 USS 路径。
名称	将此项设置为要迁移的代理程序的名称。
QM	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。这将确保在代理程序运行 IBM MQ 9.0 时，代理程序使用它在运行 IBM MQ 8.0 时所使用的同一代理程序队列管理器。
SERVICE_TYPE	代理程序
TMPDIR	将此值设置为与 IBM MQ 8.0 代理程序使用的值相同。

- 保存修改后的 BFGCUSTM 成员。
- 提交 BFGCUSTM。该成员在 PDSE 中创建可用于在代理程序使用 IBM MQ 9.0 时对其进行管理的新成员。



注意: 该成员需要对 PDSE 的独占访问权，因此您必须在 BFGCUSTM 运行时停止使用该成员。

步骤 3-执行迁移

1. 找到用于管理代理程序的 IBM MQ 8.0 PDSE。
2. 在此 PDSE 中，提交 BFGAGSP 成员以停止代理程序。代理在完成其所有受管传输后停止。
3. 接下来，找到刚刚为代理程序创建的新 IBM MQ 9.0 PDSE。
4. 在此 PDSE 中，提交 BFGAGST 成员。这将使用 IBM MQ Managed File Transfer for z/OS 9.0 启动代理程序

相关概念

使用 Managed File Transfer for z/OS

您可以在 IBM MQ for z/OS 系统上应用本主题中的指示信息作为在企业中使用 Managed File Transfer 的替代方法。

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。

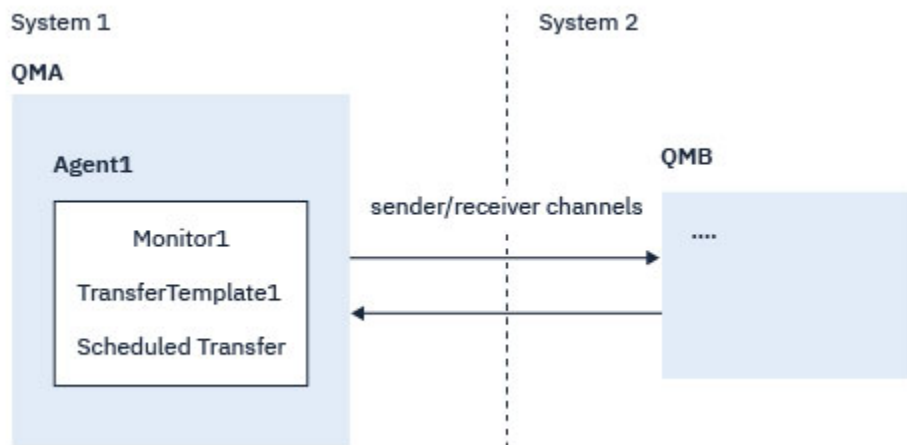


图 1: 在系统 1 上迁移 MFT 配置



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 **dmpmqcfig** 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：

```
dmpmqcfig -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 24 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。

MFT 场景概述

此部分列出常见 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](#)

MFT 常见拓扑

本部分列出常见的 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 28 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

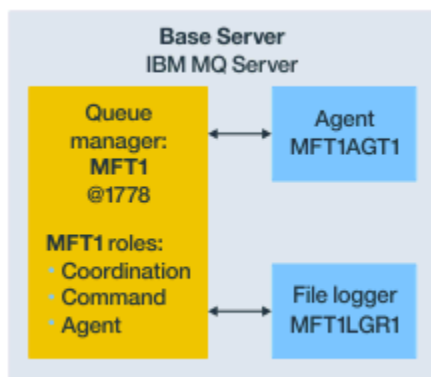


图 2: 包含一个队列管理器的基本拓扑

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 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到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

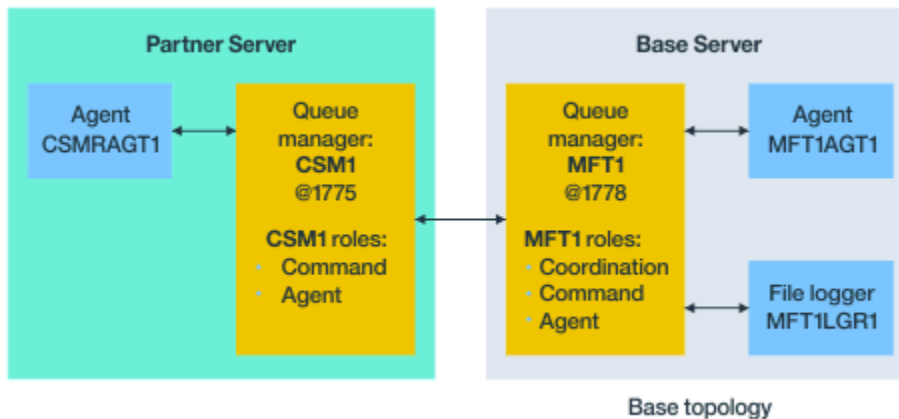


图 3: 包含一个伙伴代理的基本拓扑

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 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

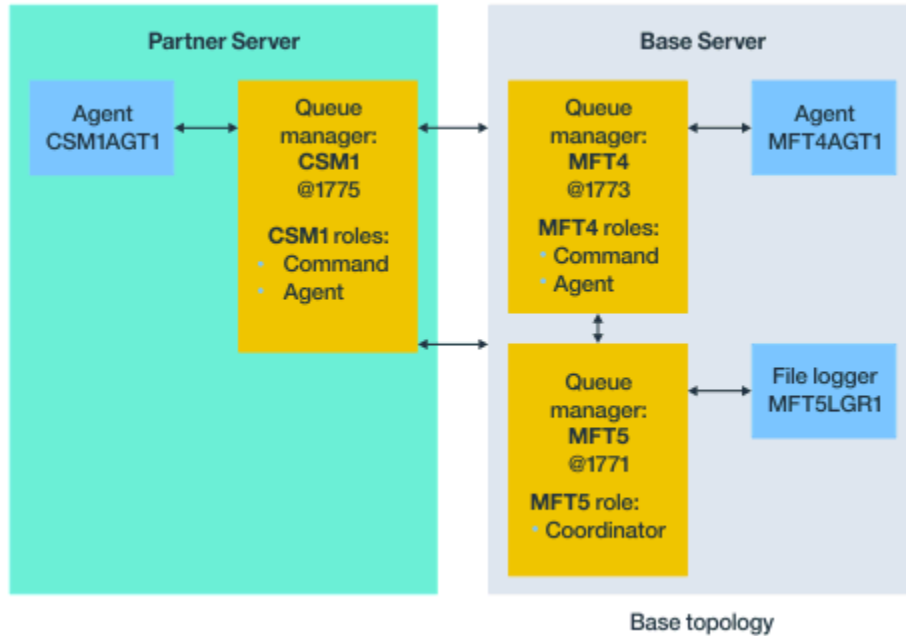


图 4: 具有单独的协调队列管理器和一个伙伴代理的基本拓扑

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 Managed File Transfer Agent 伙伴的基本拓扑

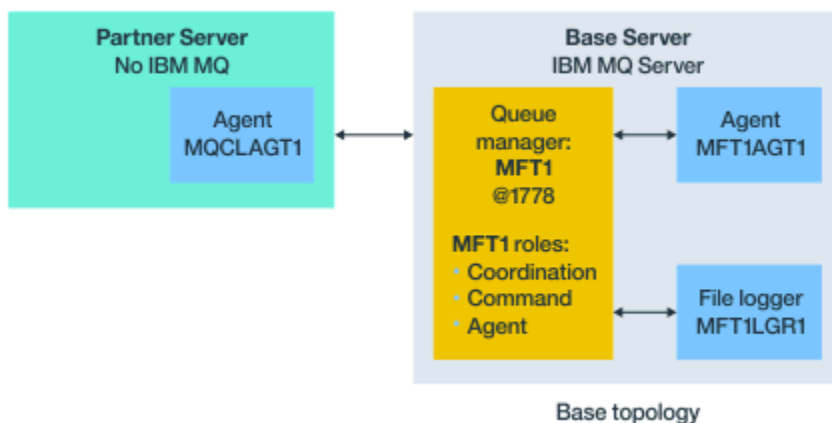


图 5: 具有 Managed File Transfer Agent 伙伴的基本拓扑

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 Managed File Transfer 配置目录和必需的属性文件。

V 9.0.1 在合作伙伴服务器上，从 IBM MQ 9.0.1 开始，您可以安装来自 Fix Central 的 Managed File Transfer 可再分发组件客户机。

注：MQMFT 可再分发组件客户机不同于 IBM MQ 可再分发组件客户机之处在于，它已打包且无需使用 `genmqpkg` 实用程序。有关更多信息，请参阅[可再分发组件客户机](#)。

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 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 28 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 Managed File Transfer 安全配置，请参阅 [Managed File Transfer 的安全概述](#)和 [有关 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 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 [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。

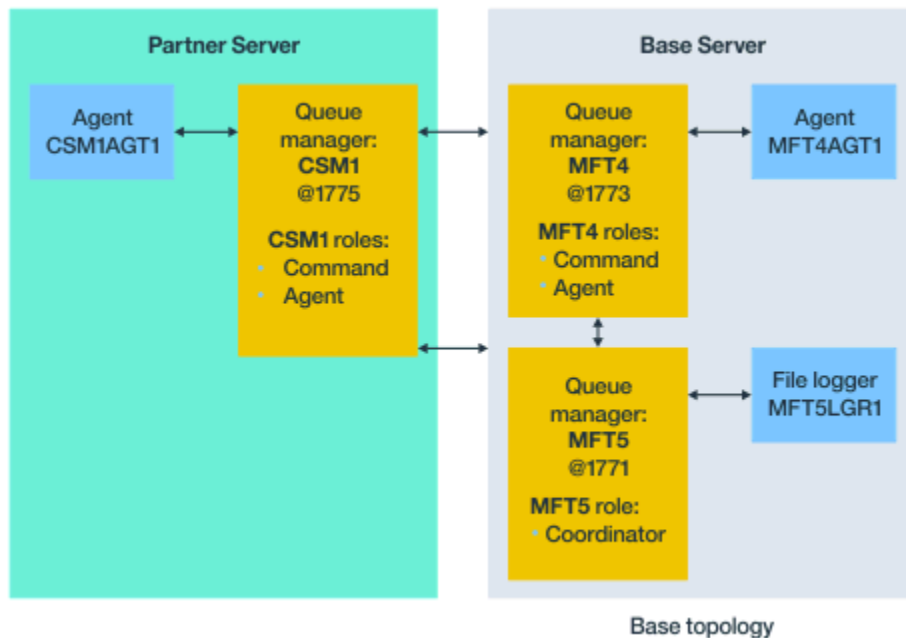


图 6: 具有单独的协调队列管理器和一个伙伴代理的基本拓扑

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 Managed File Transfer 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 30 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 30 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.

命令队列管理器不需要外部 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 30 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 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 251 页的『当 fteListAgents 命令未列出您的 MFT 代理时该怎么做』](#) 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 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外部 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 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 251 页的『[当 ftelistAgents 命令未列出您的 MFT 代理时该怎么做](#)』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 253 页的『[如果代理被显示为处于 UNKNOWN 状态, 该怎么办](#)』以获取更多信息。

使用 MFT 设置 IBM MQ Explorer

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

过程

1. 启动 IBM 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 [IBM MQ Explorer 测试示例设置](#)。

使用 IBM 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 Managed File Transfer 配合使用, 以在设置 IBM 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 IBM MQ Explorer 正常工作。请参阅第 34 页的『[使用 MFT 设置 IBM 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 IBM 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 MFT 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。

保护 MFT

安装后没有进行任何修改的情况下，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 39 页的『MFT 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。

相关任务

[第 44 页的『为 MFT 配置 SSL 或 TLS 加密』](#)

您可以将 SSL 或 TLS 与 IBM MQ Managed File Transfer 配合使用，以保护代理及其代理队列管理器之间的通信，命令及其连接到的队列管理器之间的通信，以及拓扑中的各种队列管理器到队列管理器连接之间的通信。

相关参考

[第 293 页的『限制特定于 MFT 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 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 292 页的『管理特定于 MFT 资源的权限』](#)

针对任何文件传输请求，Managed File Transfer 代理进程都需要其本地文件系统的某种访问权级别。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 302 页的『供 MFT 用于访问文件系统的权限』](#)

针对任何文件传输请求，Managed File Transfer 代理进程都需要其本地文件系统的某种访问权级别。

[第 36 页的『MFT 和 IBM MQ 连接认证』](#)

连接认证允许将队列管理器配置为使用提供的用户标识和密码对应用程序进行认证。如果关联的队列管理器已启用安全性，并且需要凭证详细信息(用户标识和密码)，那么必须先启用连接认证功能，才能成功连接到队列管理器。可以在兼容性方式或 MQCSP 认证方式下运行连接认证。

[第 303 页的『commandPath MFT 属性』](#)

使用 commandPath 属性以指定 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 302 页的『用于发布 MFT 代理日志和状态消息的权限』](#)

Managed File Transfer 代理可发出在协调队列管理器上发布的各种日志、进度和状态消息。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。

MFT 和 IBM MQ 连接认证

连接认证允许将队列管理器配置为使用提供的用户标识和密码对应用程序进行认证。如果关联的队列管理器已启用安全性，并且需要凭证详细信息(用户标识和密码)，那么必须先启用连接认证功能，才能成功连接到队列管理器。可以在兼容性方式或 MQCSP 认证方式下运行连接认证。

提供凭证详细信息的方法

许多 Managed File Transfer 命令支持以下提供凭证详细信息的方法：

由命令行参数提供详细信息。

可以使用 **-mquserid** 和 **-mqpassword** 参数来指定凭证详细信息。如果未提供 **-mqpassword**，那么将要求用户提供未显示输入的密码。

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。

有关在 IBM MQ for Multiplatforms 上设置 MQMFTCredentials.xml 文件的信息，请参阅 [第 37 页的『在多个平台上配置 MQMFTCredentials.xml』](#)。

有关在 IBM MQ for z/OS 上设置 MQMFTCredentials.xml 文件的信息，请参阅 [在 z/OS 上配置 MQMFTCredentials.xml](#)。

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. 缺省向后兼容性方式，其中未提供凭证详细信息以允许与 IBM MQ 或 IBM WebSphere MQ 的先前发行版兼容

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。

• z/OS

在 z/OS 上，即使用户的密码包含小写字母，该密码也必须为大写形式。例如，如果用户的密码是“password”，那么必须输入为“PASSWORD”。

相关参考

MFT 命令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。

[MFT 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。

在多平台上配置 MQMFTCredentials.xml

如果 Managed File Transfer (MFT) 配置为启用安全性，那么连接认证需要与队列管理器连接的所有 MFT 命令来提供用户标识和密码凭证。同样，在连接到数据库时，可能需要 MFT 记录器来指定用户标识和密码。此凭证信息可以存储在 MFT 凭证文件中。

关于此任务

MQMFTCredentials.xml 文件中的元素必须符合 MQMFTCredentials.xsd 模式。有关 MQMFTCredentials.xml 格式的信息，请参阅 [MFT 凭证文件格式](#)。

您可以在 MQ_INSTALLATION_PATH/mqft/samples/credentials 目录中找到样本凭证文件。

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。或者，您可以有一个文件供拓扑中的所有内容使用。

MFT 凭证文件的缺省位置如下所示：

Linux **UNIX** **UNIX and Linux**
\$HOME

Windows **Windows**
%USERPROFILE% 或 %HOMEDRIVE%%HOMEPATH%

如果凭证文件存储在其他位置，那么可以使用以下属性来指定命令应在何处查找该文件：

表 1: : 用于定义各种命令的 MQMFTCredentials.xml 文件的位置的属性。		
命令类型	属性文件	属性名
用于连接到协调队列管理器的命令	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
连接到命令队列管理器的命令	connection.properties	connectionQMgrAuthenticationCredentialsFile
连接到代理进程的命令	agent.properties	agentQMgrAuthenticationCredentialsFile
连接到记录器进程的命令	logger.properties	loggerQMgrAuthenticationCredentialsFile

表 2: : 用于定义代理程序和记录器进程的 MQMFTCredentials.xml 文件的位置的属性。		
命令类型	属性文件	属性名
MFT 代理程序	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT loggers	logger.properties	loggerQMgrAuthenticationCredentialsFile

有关哪些命令和进程连接到哪个队列管理器的详细信息，请参阅 [哪些 MFT 命令和进程连接到哪个队列管理器](#)。

由于凭证文件包含用户标识和密码信息，因此需要特殊许可权以防止对其进行未经授权的访问：

Linux **UNIX** **UNIX and Linux**

```
chown <agent owner userid>
chmod 600
```

Windows **Windows**

确保未启用继承，然后除去运行将使用凭证文件的代理程序或记录器的用户标识以外的所有用户标识。

用于连接到 IBM MQ Explorer Managed File Transfer 插件中的 MFT 协调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)

全局配置将使用在协调和命令属性中指定的凭证文件。

局部 (在 IBM MQ Explorer 中定义) :

局部配置将使用 IBM MQ Explorer 中的关联队列管理器的连接详细信息的属性。

相关任务

第 38 页的『启用 MQCSP 认证』

从 IBM MQ 9.0.3 for Continuous Delivery 和 IBM MQ 9.0.0 Fix Pack 2 for Long Term Support 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 IBM 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。

相关参考

MFT 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

[fteObfuscate](#): 对敏感数据进行加密

V 9.0.0.2 V 9.0.3 启用 MQCSP 认证

从 IBM MQ 9.0.3 for Continuous Delivery 和 IBM MQ 9.0.0 Fix Pack 2 for Long Term Support 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 IBM 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 IBM 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 9.0.3 和 IBM MQ 9.0.0 Fix Pack 2 开始, 可以禁用缺省兼容性方式并启用 MQCSP 认证方式。

过程

- 要对 IBM 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:
 - a) 选择要连接到的队列管理器。
 - b) 单击右键, 并从弹出菜单中选择**连接详细信息 -> 属性**。
 - c) 单击**用户标识**选项卡。
 - d) 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 **useMQCSPAuthentication** 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 **useMQCSPAuthentication** 参数:

```
<tns:qmgr name="CoordQueueMgr" user="ernest" mqUserId="ernest"
mqPassword="AveryL0ngPassw0rd2135" useMQCSPAuthentication="true"/>
```

相关参考

第 36 页的『MFT 和 IBM MQ 连接认证』

连接认证允许将队列管理器配置为使用提供的用户标识和密码对应用程序进行认证。如果关联的队列管理器已启用安全性，并且需要凭证详细信息 (用户标识和密码)，那么必须先启用连接认证功能，才能成功连接到队列管理器。可以在兼容性方式或 MQCSP 认证方式下运行连接认证。

第 691 页的『MFT 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](#)

MFT 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。不能将代理沙箱用于需要与 IBM MQ 队列传输数据的代理。

相关参考

第 39 页的『使用 MFT 代理沙箱』

要向 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 40 页的『使用 MFT 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 MFT 代理沙箱

要向 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 40 页的『使用 MFT 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_nameseparator...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 Managed File Transfer Agent 可对哪些目录进行读写。激活沙箱后，Managed File Transfer Agent 可以读写指定为允许的目录以及指定目录包含的任何子目录，除非在 `sandboxRoot` 中将这些子目录指定为拒绝。Managed File Transfer 沙箱不优先于操作系统安全性。启动 Managed File Transfer Agent 的用户必须对任何目录具有相应的操作系统级别访问权，才能对

该目录进行读写。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 系统上的沙箱中工作

z/OS 在 z/OS 上，沙箱限制 Managed File Transfer Agent 可读写的数据集名称限定符。启动 Managed File Transfer Agent 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 数据集名称限定符值引起来，那么该值遵循标准的 z/OS 约定并被视为标准值。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 属性设置如下：sandboxRoot=//test，那么代理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，那么会拒绝传输请求。

在 IBM i 系统上的沙箱中工作

IBM i 对于 IBM i 系统上集成文件系统中的文件，沙箱限制了 Managed File Transfer Agent 可读写的目录。激活沙箱后，Managed File Transfer Agent 可以读写指定为允许的目录以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拒绝。Managed File Transfer 沙箱不优先于操作系统安全性。启动 Managed File Transfer Agent 的用户必须对任何目录具有相应的操作系统级别访问权，才能对该目录进行读写。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。

相关参考

第 43 页的『针对通配符传输的额外检查』

从 IBM MQ 9.0.1 (在 Continuous Delivery 中) 和 IBM MQ 9.0.0 Fix Pack 1 (在 Long Term Support 中) 开始，如果已使用用户或代理沙箱配置代理以限制代理可以将文件传输到的位置，那么可以指定要对该代理的通配符传输进行其他检查。

第 39 页的『使用 MFT 代理沙箱』

要向 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 460 页的『MFT agent.properties 文件』

每个 Managed File Transfer Agent 都有自己的属性文件 agent.properties，该文件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 MFT 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 属性会解释为 Java 正则表达式。有关更多信息，请参阅第 549 页的『MFT 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，那么代理会把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或数据集名称或队列名称的限定符中的零个或多个字符匹配。

?

问号与目录名称或数据集名称或队列名称的限定符中的一个字符完全匹配。

两个星号字符与零个或多个目录名匹配，或者与数据集名称或队列名称中的零个或多个限定符匹配。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。
- *@QM1 与队列管理器 QM1 上具有单个限定符的任何队列匹配。

- 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。
- **@QM1 与队列管理器 QM1 上的任何队列匹配。

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。

示例

此示例显示如何通过将以下 `<sandbox>` 元素添加到 AGENT_JUPITER 的配置目录中的文件 `UserSandboxes.xml`，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：

```
<?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

此示例说明了 MQMD 用户名为 `account` 且后跟单个数字的任何用户（例如，`account4`）如何完成以下操作：

- 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。

要允许具有 MQMD 用户名 `account` 的用户完成这些操作，请将以下 `<sandbox>` 元素添加到 AGENT_SATURN 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="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>
```

相关参考

第 43 页的『[针对通配符传输的额外检查](#)』

从 IBM MQ 9.0.1 (在 Continuous Delivery 中) 和 IBM MQ 9.0.0 Fix Pack 1 (在 Long Term Support 中) 开始, 如果已使用用户或代理沙箱配置代理以限制代理可以将文件传输到的位置, 那么可以指定要对该代理的通配符传输进行其他检查。

第 40 页的『[使用 MFT 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 460 页的『[MFT agent.properties 文件](#)』

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`, 该文件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。

V 9.0.1 V 9.0.0.1 针对通配符传输的额外检查

从 IBM MQ 9.0.1 (在 Continuous Delivery 中) 和 IBM MQ 9.0.0 Fix Pack 1 (在 Long Term Support 中) 开始, 如果已使用用户或代理沙箱配置代理以限制代理可以将文件传输到的位置, 那么可以指定要对该代理的通配符传输进行其他检查。

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 Continuous Delivery 中的 IBM MQ 9.0.1 和 Long Term Support 中的 IBM MQ 9.0.0 Fix Pack 1 开始, 对配置的沙箱位置外部的位发出通配符传输请求时报告的消息已更改。

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: 尝试读取文件路径 path 已作为另一个传输被忽略
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: 尝试读取文件 文件路径 已作为另一个传输被忽略
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 40 页的『[使用 MFT 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 39 页的『使用 MFT 代理沙箱』

要向 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 460 页的『MFT agent.properties 文件』

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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 MFT 配置 SSL 或 TLS 加密

您可以将 SSL 或 TLS 与 IBM MQ Managed File Transfer 配合使用，以保护代理及其代理队列管理器之间的通信，命令及其连接到的队列管理器之间的通信，以及拓扑中的各种队列管理器到队列管理器连接之间的通信。

开始之前

您可以使用 SSL 或 TLS 加密对流经 IBM MQ Managed File Transfer 拓扑的消息进行加密。其中包括：

- 在代理与其代理队列管理器之间传递的消息。
- 命令的消息以及它们要连接到的队列管理器。
- 在拓扑中的代理队列管理器，命令队列管理器和协调队列管理器之间流动的内部消息。

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，Managed File Transfer 是标准 Java 客户机应用程序。

遵循以下步骤将 SSL 与 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。

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 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 和 Java 之间的术语差异，提供了 **CipherSpec** 或 **CipherSuite** 属性。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。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 504 页的『MFT 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. 如果在运行 IBM MQ Explorer 代理或实例期间更改了 `agent.properties` 文件或 `coordination.properties` 文件中的 SSL 属性，那么必须重新启动代理或 IBM MQ Explorer。

相关参考

第 460 页的『MFT `agent.properties` 文件』

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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。

使用通道认证以客户机方式连接到队列管理器

IBM WebSphere MQ 7.1 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 IBM WebSphere MQ 7.1 或更高版本队列管理器将拒绝来自 Managed File Transfer 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 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，Managed File Transfer Agent 和命令才能正确工作。使用 MQSC 命令 SET CHLAUTH 或 PCF 命令 Set Channel Authentication Record 来创建、修改或删除通道认证记录。对于要连接到 IBM WebSphere MQ 7.1 或更高版本队列管理器的所有 Managed File Transfer 代理程序，可以设置要用于所有代理程序的 MCAUSER 标识，也可以为每个代理程序设置单独的 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到 IBM WebSphere MQ 7.1 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

代理用户	队列	所需权限
user2	SYSTEM.FTE.COMMAND.source_agent_name	put

在 Connect:Direct 网桥代理与 Connect:Direct 节点之间配置 SSL 或 TLS

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 IBM 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，可以使用现有的密钥库和信任库。有关更多信息，请参阅第 44 页的『为 MFT 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。
您可以使用 IBM MQ 随附的 IBM 密钥管理工具来执行此操作。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-----
MIIcCzCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBBeMQswCQYDVQGEwJHqjES
MBAGA1UECBMJSjGfTcHNoaXJlMRAwDgYDVQQHEwdIdXJzbGV5M0wwCgYDVQQKEwNj
Qk0xOjQjAMBgNVBAsTBUI1RSVBUQswCQYDVQDEwJDQTAeFw0xMTAzMDExNjIwNDZa
Fw0yMTAyMjYxNjIwNDZaMFAXCzAJBgNVBAYTAkdCMRlWIAEYDVQQIEwLIYw1w2hp
cmUxDDAKBgNVBAoTA0lCTTEOMAwGA1UECxMFTVFGVEUxZDZANBgNVBAMTBmJpbmJh
ZzCBnzANBgkqhkiG9w0BAQEFAA0BjQAwgYkCgYEAvgP1QIk1U9ypSKD1Xo0Do1yk
EymFXBOUpZrDvXjoSEC0vtWncJ199e+Vc4UpNybdyBu+Nkd1MnofX4QxeQcLAFj
WnhakqCiQ+JIAD5AurhnwChe0MV3kjA84GKH/z0SVqt1984mu/ldyS819XcfsSn
c00MsK1KbneVSCIV2XECaWEAAaN7MHkwcQYDVR0TBAlwADAsBglghkgBhvhCAQ0E
HxYdTB3B1b1NTTCBHZW51cmF0ZlZlZlVydGlmawNhdGUwHQYDVR0BBYEFNXXMIpSc
csBXUniW4A3UzrZnCRsv3MB8GA1UdIwQYMBaAFDXY8rmj41Vz5+FVAoQb++cns+B4
MA0GCSqGSIb3DQEBBQUAA4GBAfc7k1Xa4pGKYgwhxKpE3ZF6FNwy4vBXS216/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
1vI99QyCxsDwoMnt5fj51v7aPmVes60b0m+U1Gre8B/Zel8JVj204K2Uh72rDCXE
5e6eFxDUM207sQdy20euBVELJtM2k0kL1R0doQs1U3XQNgJw/t3ZIx5hPXWEQT
rjRQ064BEhb+PzzxPF8uwzZ9IzUK9BJ/UUnqC60dBR87IEA4pnJD1Jvb2ML7EN9Z
5Y+50hTKI80GvBvWx04fHyvIX5as1whBoArXIS1AtNTtPv0aP1zyIAeZ60Cvo/
Sfo+A2UhmTEJeOJaZG2XZ3H495fAw/EHmjehzIACwuk09nSIETgu4A1+CV64RJED
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWdyJBVhU0jXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfdwP+bEjDzUaaarJTS7lIFeLlw7eJ8MNAKMGicDkycL0

```

```
EPBU9X5QnHKLK0fYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+1YcX80VZ6
sHFPN1H1uCNy/riUcBy9iviVeodX8Iom0chSy05DK18bwZNjYtUP+CtYHNFU5BaD
I+1uU0AeJ+wjQYKT1WaeIGZ3VxuNITJul8y5qDTXXfX7vxM50oWXa6U5+AYuGUMg
/itPZmUmNzHjTk7ghT6i1IQ0aBowXXKJB1Mmq/6BQXN2IhkD9ys2qrvM1hdi5nAf
egmdiG50l0LnBRqWbfR+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。

- a) 根据使用的协议，选择启用 **SSL 协议**或启用 **TLS 协议**。
- b) 选择**禁用覆盖**。
- c) 至少选择一个密码套件。
- d) 如果需要双向认证，请将 **启用客户机认证** 的值更改为 Yes。
- e) 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`。
- f) 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。

- a) 根据使用的协议，选择启用 **SSL 协议**或启用 **TLS 协议**。
- b) 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 118 页的『配置 Connect:Direct 网桥』](#)

配置 Connect:Direct 网桥，以在 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 Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

配置 MFT

安装完成以后，您可以配置 Managed File Transfer 的功能部件。

相关概念

[第 213 页的『使用用户出口定制 MFT』](#)

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

[第 204 页的『MFT 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 Managed File Transfer 将在这些场景下按如下方式恢复：

[第 227 页的『使用 MFT 的提示与技巧』](#)

以下建议可帮助您最有效地使用 Managed File Transfer。

[第 128 页的『管理 MFT』](#)

使用 Managed File Transfer 命令来管理 Managed File Transfer。您也可以将 IBM MQ Explorer 用于某些管理任务。

相关任务

[第 141 页的『监视 MFT 资源』](#)

您可以监视 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 Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

[第 37 页的『在多平台上配置 MQMFTCredentials.xml』](#)

如果 Managed File Transfer (MFT) 配置为启用安全性，那么连接认证需要与队列管理器连接的所有 MFT 命令来提供用户标识和密码凭证。同样，在连接到数据库时，可能需要 MFT 记录器来指定用户标识和密码。此凭证信息可以存储在 MFT 凭证文件中。

相关参考

[第 310 页的『MFT 命令』](#)

下表列出了所有 Managed File Transfer 命令以及指向其详细描述的连接。

[第 35 页的『保护 MFT』](#)

安装后没有进行任何修改的情况下，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。

[第 460 页的『MFT agent.properties 文件』](#)

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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 209 页的『将 Apache Ant 与 MFT 结合使用』](#)

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 226 页的『对 MFT 进行故障诊断』](#)

使用以下参考信息可帮助您诊断 Managed File Transfer 中的错误：

多平台上的 MFT 配置选项

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 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 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个配置选项目录。`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` 目录中可以有多个代理目录。

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 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。要了解有关 `installation.properties` 文件的更多信息，请参阅 [第 450 页的『MFT 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 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。要了解有关 `coordination.properties` 文件的更多信息，请参阅第 453 页的『MFT `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 Managed File Transfer 联系该队列管理器所需的信息。要了解有关 `command.properties` 文件的更多信息，请参阅第 457 页的『MFT `command.properties` 文件』。

agent.properties

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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。要了解有关 `agent.properties` 文件的更多信息，请参阅第 460 页的『MFT `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 98 页的『MFT 记录器配置属性』。

属性文件和代码页

由于 Java 的限制，所有 Managed File Transfer 属性文件的内容都必须保持美国英语形式。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。

z/OS 上的 MFT 配置选项

z/OS 上的 Managed File Transfer 配置选项与针对分布式平台的选项相同。

有关多平台上的配置选项的更多信息，请参阅第 49 页的『多平台上的 MFT 配置选项』。

在 z/OS 上，配置位置由环境变量 `BFG_DATA` 定义。如果在 `BFG_DATA` 引用的 UNIX 系统服务目录下尚不存在配置，那么 MFT 命令 PDSE 库数据集的 `BFGCUSTOM 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`。

您还可以使用在多平台和 z/OS 上都可用的相同 `fte` 命令来创建和维护配置。要获取 `fte` 命令的列表，请参阅第 310 页的『MFT 命令』。

V 9.0.1 配置 Redistributable Managed File Transfer Agent

配置可选择提供的 Redistributable Managed File Transfer Agent 以连接到现有的 IBM MQ 基础结构，并支持用户在没有安装 IBM MQ 的情况下传输文件 以获得 Managed File Transfer 功能。

开始之前

Redistributable Managed File Transfer Agent 提供 Managed File Transfer 的功能，但存在以下例外情况：

- 不支持与协调、命令和代理队列管理器的绑定模式连接，您必须使用客户机模式连接。发出命令时，必须提供在使用作为 IBM MQ 的一部分安装的 Managed File Transfer 时可选的参数：队列管理器主机，端口，名称和通道名称。
- 不包括以下命令：
 - `fteCreateCDAgent.cmd`
 - `fteCreateLogger.cmd`
 - `fteDeleteLogger.cmd`
 - `fteMigrateLogger.cmd`
 - `fteSetLoggerTraceLevel.cmd`
 - `fteShowLoggerDetails.cmd`
 - `fteStartLogger.cmd`

- fteStopLogger.cmd

有关可用命令的完整列表，请参阅[已安装的 MFT 命令集](#)。

- 不支持 Managed File Transfer Connect:Direct。
- 不包括 IBM MQ Explorer。

Windows 您必须在系统上安装以下 Microsoft 库才能使用 Redistributable Managed File Transfer Agent:

- Microsoft Visual C++ Redistributable 2008
- Microsoft Visual C++ Redistributable 2012

这些库可从 Microsoft 中获取。请参阅 [The latest supported Visual C++ downloads](#)。

关于此任务

您希望允许用户在本地环境与现有 IBM MQ 基础结构之间传输文件，而无需安装 IBM MQ。遵循以下步骤来下载并解压缩 Redistributable Managed File Transfer Agent:

过程

1. 从 [IBM Fix Central](#) 下载 Redistributable Managed File Transfer Agent 软件包。

a) 为您的操作系统选择软件包:

- **Windows** 9.0.1.0-IBM-MQFA-Redist-Win64
- **Linux** 9.0.1.0-IBM-MQFA-Redist-LinuxX64
- **Linux** 9.0.1.0-IBM-MQFA-Redist-LinuxS390X
- **Linux** 9.0.1.0-IBM-MQFA-Redist-LinuxPPC64LE

b) 确定您要在其中将软件包解压缩的目录，例如:

- **Windows** C:\MFTZ
- **Linux** /home/MFTZ

2. 将所下载软件包的内容解压缩:

- **Windows** 使用 Windows Explorer 工具来解压缩。
- **Linux** 按如下所示解压缩:

```
gunzip 9.0.1.0-IBM-MQFA-Redist-LinuxX64.tar.gz
```

然后,

```
tar xvf 9.0.1.0-IBM-MQFA-Redist-LinuxX64.tar
```

创建以下目录:

- **Windows** **Linux** bin: 包含所有必需的 MFT 命令
- **Windows** bin64: 包含 Windows 64 位操作系统支持所需的库
- **Windows** **Linux** java: 包含 IBM JRE 和 IBM MQ 库
- **Windows** **Linux** licenses: 包含许可证文件
- **Windows** **Linux** mqft: 包含 Ant 支持和核心 MFT 功能支持所需的 ant 和 lib 目录

-   `swtag`: 包含 `swidtag` 文件, 许可证管理器需要此文件来识别计算机上的安装

下一步做什么

您准备好配置 MFT 代理。有关后续步骤, 请参阅以下主题: [第 52 页的『创建初始配置』](#)。

相关参考

第 268 页的『[配置 MFT 代理可再分发组件时可能出现的错误](#)』
配置 Redistributable Managed File Transfer Agent 时的错误消息

相关信息

视频: [如何安装和配置 Redistributable Managed File Transfer Agent \(YouTube\)](#)

创建初始配置

您可以配置 Managed File Transfer Agent 以连接到现有 IBM MQ 配置。

开始之前

 确保您下载并解压缩 Redistributable Managed File Transfer Agent 软件包的内容。有关更多信息, 请参阅 [第 50 页的『配置 Redistributable Managed File Transfer Agent』](#)。

关于此任务

创建环境, 从而可以使用命令来设置与 IBM MQ 服务器上运行的队列管理器的连接。

过程

1. 为 Redistributable Managed File Transfer Agent 创建环境。

运行 `fteCreateEnvironment` 命令时, 将创建包含 MFT 代理程序的配置信息的 MFT 数据目录。确保您在 `bin` 目录中。此目录是在您将下载的 MFT 代理可再分发组件解压缩时创建的。运行以下命令:

- 

```
fteCreateEnvironment.cmd -d datapath location
```

- 

```
./fteCreateEnvironment -d datapath location
```

此命令采用一个可选参数:

-d

在其中创建、存储和维护 MFT 配置的数据路径位置。如果用户在未指定数据位置的情况下运行 `fteCreateEnvironment`, 那么将在解压缩 Redistributable Managed File Transfer Agent 的位置中创建目录 `mftdata`。

注: 如果可再发行代理将作为 Windows 服务运行, 那么 `BFG_DATA` 环境变量需要在系统环境中进行设置, 以便服务正常工作。

您还可以设置环境变量 `BFG_DATA` 以及数据路径位置:

```
BFG_DATA=Datapath location
```

在创建、启动和停止代理或者其他任何命令之前, 您必须确保 `BFG_DATA` 变量设置为正确的数据路径位置。

2. 设置 IBM MQ 连接。
 - a) 使用 `fteSetupCoordination` 命令来设置协调队列管理器。

fteSetupCoordination 命令创建协调队列管理器所需的设置以及进一步配置所需的目录。Redistributable Managed File Transfer Agent 以客户机方式工作，因此您必须向此命令提供额外参数来避免错误，因为不支持绑定方式。

```
fteSetupCoordination -coordinationQMgr PRMFTDEM02
                    -coordinationQMgrHost 9.121.59.233 -coordinationQMgrPort 3002
                    -coordinationQMgrChannel SYSTEM.DEF.SVRCONN
```

有关使用 **fteSetupCoordination** 命令的更多详细信息和步骤，请参阅第 430 页的『[fteSetupCoordination](#)』。有关如何配置协调队列管理器的信息，请参阅第 82 页的『[为 MFT 配置协调队列管理器](#)』。

b) 创建并设置命令队列管理器：

```
fteSetupCommands -p PRMFTDEM02 -connectionQMgrHost 9.121.59.233
                 -connectionQMgrPort 3002 -connectionQMgrChannel SYSTEM.DEF.SVRCONN
                 -connectionQMgr PRMFTDEM02 -f
```

有关使用 **fteSetupCommands** 命令的更多详细信息和步骤，请参阅第 428 页的『[fteSetupCommands: 创建 MFT command.properties 文件](#)』。

3. 为端点创建 MFT 代理定义。

```
fteCreateAgent -p PRMFTDEM02 -agentQMgrHost 9.121.59.233
              -agentQMgrPort 3002 -agentQMgrChannel SYSTEM.DEF.SVRCONN
              -agentName AGENT.TRI.BANK -agentQMgr PRMFTDEM02 -f
```

有关使用 **fteCreateAgent** 命令来配置代理和代理队列管理器的更多信息，请参阅第 323 页的『[fteCreateAgent \(创建 MFT 代理\)](#)』。

在每个代理的步骤 [第 52 页的『2』](#) 和 [第 53 页的『3』](#) 中，创建有关代理队列管理器的队列和主题定义。

4. 启动代理，然后随时可以开始传输文件。

```
fteStartAgent -p PRMFTDEM02 AGENT.TRI.BANK
```

您可以通过运行以下命令来验证代理的状态：

```
fteListAgents
```

有关使用 **fteListAgents** 命令的更多详细信息，请参阅第 395 页的『[fteListAgents: 列出协调队列管理器的 MFT 代理](#)』。

相关概念

第 48 页的『[配置 MFT](#)』

安装完成以后，您可以配置 Managed File Transfer 的功能部件。

第 49 页的『[多平台上的 MFT 配置选项](#)』

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 364 页的『[fteCreateTransfer: 启动新的文件传输](#)』

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。

z/OS 创建 MFT 代理或记录器命令数据集

您可以通过 Managed File Transfer 命令模板数据集，为特定协调的特定 Managed File Transfer Agent 或 Managed File Transfer Logger 创建 PDSE 命令数据集。

关于此任务

完成以下步骤：

过程

1. 制作 MFT 命令模板 PDSE 库数据集 SBFGCMD5 的副本。

z/OS 必须将 SBFGCMD5 复制到新库中，例如 *prefix.agent.JCL_*。您可以使用 SBFGCMD5(BFGCOPY) 成员的已更新版本（进行以下替换操作）：

- 将 *++supplied-library++* 替换为 SBFGCMD5 PDSE 的标准名称。
- **z/OS** 将 *++service-library++* 替换为新 MFT 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 MFT 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 65 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 MFT 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 76 页的『配置属性文件』。提供此功能之后，将不必访问作为 UNIX 系统服务文件进行维护的 MFT 配置属性文件。

Environment 定义了所需的其他所有定制环境变量。

3. 为新的 MFT 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 69 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 脚本还会根据需要来更新 UNIX 系统服务 MFT 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 BFLGCRS。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 50 页的『z/OS 上的 MFT 配置选项』

z/OS 上的 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 65 页的『在 z/OS 上更新现有 MFT 代理或记录器命令数据集』

您可以更新从 Managed File Transfer 命令模板数据集创建的 Managed File Transfer 命令 PDSE 库数据集。

z/OS 配置 Managed File Transfer for z/OS

Managed File Transfer for z/OS 需要定制以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 55 页的『[查看 MFT 配置](#)』

在开始之前，您需要查看系统的配置。

相关信息

[安装 Managed File Transfer for z/OS](#)

查看 MFT 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，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 56 页的『[在启动前](#)』

Managed File Transfer (MFT) 配置使用 UNIX 系统服务 (USS) 中的文件和 PDSE 数据集。

z/OS 在启动前

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

命令	函数
ls -ltr path	列示有关路径中文件的信息。
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。

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	

表 4: 成员 BFGCUSTM 所需的参数 (继续)		
名称	示例数据	注释
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	*	
JOBCARD1		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 57 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 61 页的『编辑成员 BFGCUSTM』

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 Managed File Transfer 数据的目录

您需要一个用于存储 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` 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
```

提供

```
drwxr-xr-x 4 SYSTASK TSUSER 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 `TSUSER` 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：

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 56 页的『在启动前』

Managed File Transfer (MFT) 配置使用 UNIX 系统服务 (USS) 中的文件和 PDSE 数据集。

第 59 页的『适用于 z/OS 的常见 MFT 配置』

不同的 Managed File Transfer 配置概述

z/OS 适用于 z/OS 的常见 MFT 配置

不同的 Managed File Transfer 配置概述

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 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

如果您具有现有 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。

z/OS 在 z/OS 上配置 MQMFTCredentials.xml

如果 Managed File Transfer (MFT) 配置为启用安全性，那么连接认证需要所有 MFT 代理程序以及连接到队列管理器的命令来提供用户标识和密码凭证。

同样，在连接到数据库时，可能需要 MFT 记录器来指定用户标识和密码。

此凭证信息可以存储在 MFT 凭证文件中。请注意，凭证文件是可选的，但是，在定制环境之前，更容易定义所需的一个或多个文件。

除此之外，如果您有凭证文件，那么会收到更少的警告消息。警告消息将告知您 MFT 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。

您可以在 MQ_INSTALLATION_PATH/mqft/samples/credentials 目录中找到样本凭证文件。

以下是 MQMFTCredentials.xml 文件的示例：

```
<?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">
  <tns:qmgr name="MQPH" user="ADMIN" mqUserId="JOHNDOEH" mqPassword="cXXXX" />
  <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。

MQMFTCredentials.xml 文件的缺省位置是用户在 z/OS Unix 系统服务 (USS) 上的主目录。还可以将文件存储在 USS 上的其他位置或分区数据集中的成员中。

如果凭证文件存储在其他位置，那么可以使用以下属性来指定命令应在何处查找该文件：

表 6：用于定义各种命令的 MQMFTCredentials.xml 文件的位置的属性。		
命令类型	属性文件	属性名
用于连接到协调队列管理器的命令	coordination.properties	coordinationQMGrAuthenticationCredentialsFile
连接到命令队列管理器的命令	connection.properties	connectionQMGrAuthenticationCredentialsFile
连接到代理进程的命令	agent.properties	agentQMGrAuthenticationCredentialsFile
连接到记录器进程的命令	logger.properties	loggerQMGrAuthenticationCredentialsFile

表 7：用于定义代理程序和记录器进程的 MQMFTCredentials.xml 文件的位置的属性。		
命令类型	属性文件	属性名
MFT 代理程序	agent.properties	agentQMGrAuthenticationCredentialsFile
MFT loggers	logger.properties	loggerQMGrAuthenticationCredentialsFile

有关哪些命令和进程连接到哪个队列管理器的详细信息，请参阅 [哪些 MFT 命令和进程连接到哪个队列管理器](#)。

要在分区数据集中创建凭证文件，请执行以下步骤：

- 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/MFTCredentials MQMFTCredentials.xsd">
  <!--credentials information goes here-->
</tns:mqmftCredentials>
```

您可以使用安全性产品 (例如, RACF) 来保护凭证文件, 但是运行 Managed File Transfer 命令并管理代理程序和记录器进程的用户标识需要对此文件的读访问权。

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。

相关任务

在 Multiplatforms 版上配置 MQMFTCredentials.xml

如果 Managed File Transfer (MFT) 配置为启用安全性, 那么连接认证需要与队列管理器连接的所有 MFT 命令来提供用户标识和密码凭证。同样, 在连接到数据库时, 可能需要 MFT 记录器来指定用户标识和密码。此凭证信息可以存储在 MFT 凭证文件中。

z/OS 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 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 59 页的『适用于 z/OS 的常见 MFT 配置』

不同的 Managed File Transfer 配置概述

第 61 页的『编辑成员 BFGCUSTM』

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。

z/OS 编辑成员 BFGCUSTM

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 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 56 页的『在启动前』

Managed File Transfer (MFT) 配置使用 UNIX 系统服务 (USS) 中的文件和 PDSE 数据集。

第 64 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

Managed File Transfer 要求创建的队列管理器可充当协调队列管理器。

根据您的选择的配置, 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 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 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)
```

相关任务

第 63 页的『定义命令队列管理器』

可以使用同一个队列管理器作为协调队列管理器和命令队列管理器，或者创建新的命令队列管理器。

定义命令队列管理器

可以使用同一个队列管理器作为协调队列管理器和命令队列管理器，或者创建新的命令队列管理器。

关于此任务

您必须具有命令队列管理器，但可以使用同一个队列管理器作为协调队列管理器和命令队列管理器。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。

过程

1. 使用您自己的前缀重命名成员 BFGCMCR，例如 CCPCMCR。
您必须重命名 BFGCMCR，因为重新定制此文件时会替换此文件。
2. 通过插入凭证文件的名称，编辑重命名的成员。
例如：

```
%BFGCMD CMD=fteSetupCommands +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
```

3. 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。
4. 查看此成员，进行提交，然后查看输出。

下一步做什么

请参阅第 64 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 62 页的『定义协调队列管理器』

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 37 页的『在多平台上配置 MQMFTCredentials.xml』

如果 Managed File Transfer (MFT) 配置为启用安全性，那么连接认证需要与队列管理器连接的所有 MFT 命令来提供用户标识和密码凭证。同样，在连接到数据库时，可能需要 MFT 记录器来指定用户标识和密码。此凭证信息可以存储在 MFT 凭证文件中。

相关参考

第 691 页的『MFT 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。

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 61 页的『[编辑成员 BFGCUSTM](#)』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 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 62 页的『[定义协调队列管理器](#)』

Managed File Transfer 要求创建的队列管理器可充当协调队列管理器。

第 64 页的『[使用代理](#)』

如何使用各种命令来确保代理正确工作。

相关任务

第 63 页的『[定义命令队列管理器](#)』

可以使用同一个队列管理器作为协调队列管理器和命令队列管理器，或者创建新的命令队列管理器。

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 70 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 64 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。

在 z/OS 上更新现有 MFT 代理或记录器命令数据集

您可以更新从 Managed File Transfer 命令模板数据集创建的 Managed File Transfer 命令 PDSE 库数据集。

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，该值指示将移除此属性。

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 69 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 50 页的『[z/OS 上的 MFT 配置选项](#)』

z/OS 上的 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 53 页的『[创建 MFT 代理或记录器命令数据集](#)』

您可以通过 Managed File Transfer 命令模板数据集，为特定协调的特定 Managed File Transfer Agent 或 Managed File Transfer Logger 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 MFT 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。

替换变量	值
++library++	包含 MFT 命令 PDSE 库的数据集名称。

表 8: 针对 <i>BFGCUSTM JCL</i> 脚本的替换值 (继续)	
替换变量	值
++bfg_java_home++	Java 安装的位置。
++bfg_prod++	MFT 产品安装 UNIX 系统服务根目录的位置。

下表描述了针对 MFT 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 9: 环境变量	
环境变量	值
LIBRARY	包含 MFT 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	MFT 产品安装 UNIX 系统服务根目录的位置。
BFG_DATA	Managed File Transfer for z/OS 的数据目录的位置，即指向 <i>DATA_DIR</i> 的路径。
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。这些属性将传递至 Java 虚拟机。

表 9: 环境变量 (继续)

环境变量	值
BFG_GROUP_NAME	<p>mqm 文件组通常与 MFT 配置数据文件和命令关联。因此，属于 mqm 组成员的所有用户都可以访问和更改 MFT 配置。要了解更多信息，请参阅第 291 页的『IBM MQ 中 MFT 的文件系统许可权』。</p> <p>对于 z/OS 系统，文件组是 USS 文件系统实体，并且未必会定义 mqm 文件组。可以使用 BFG_GROUP_NAME 环境变量来关联 MFT 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：</p> <pre>export BFG_GROUP_NAME=FTEGB</pre> <p>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。</p> <p>您可以将 BFG_GROUP_NAME 设置为空值或将其移除。</p> <p>注：当首次运行 BFGCUSTM 时，如果 MFT 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）。</p> <p>V9.0.0.1 如果在 BFG_GROUP_NAME 环境变量设置为非空值的 z/OS 系统上运行 fteMigrateAgent 命令，那么该命令会检查以确定用户是否为 BFG_GROUP_NAME 变量所指定的组的成员。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 403 页的『fteMigrate 代理: 将 FTE 7.0 代理迁移到 MQ 7.5 或更高版本』。</p>
BFG_WTO	<p>将 BFG_WTO 设置为 YES、ON 或 TRUE 时，将会启用 z/OS 日志记录。这可控制是否将写入到代理事件日志的消息也写入到 z/OS 操作员日志工具，从而在通过 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)。</p>
SERVICE_TYPE	<p>指定 MFT 命令库是用于代理还是记录器。有效值为 AGENT 或 LOGGER。</p>
名称	<p>针对 SERVICE_TYPE 值的代理或记录器的名称。</p>
QMGR	<p>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。</p>
OUTPUT_CLASS	<p>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。</p>
MQ_PATH	<p>在 BFGPROF 中用于创建 LIBPATH 环境变量。</p>

表 9: 环境变量 (继续)	
环境变量	值
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 Managed File Transfer 迁移的凭证文件必须位于到 IBM 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。该凭证文件仅对迁移命令 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。仅当属性文件不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 MFT 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 MFT 命令 PDSE 库数据集中 BFGCUSTM JCL 脚本的 BFGSTDIN DD 语句的必需 MFT 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 MFT 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 76 页的『配置属性文件』。

表 10: 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 10: 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

MFT 命令 PDSE 库数据集中可用的 JCL 命令集。

表 11: 可以在 MFT 命令 PDSE 库数据集中找到 JCL 命令

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。

表 11: 可以在 MFT 命令 PDSE 库数据集中找到 JCL 命令 (继续)

成员	描述或 fte 命令行命令
BFGCFDF	fteChangeDefaultConfigurationOptions
BFGAGCL	fteCleanAgent 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGAGDE	fteDeleteAgent 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGLGDE	fteDeleteLogger 。仅在将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGPRSH	fteDisplayVersion
BFGAGLI	fteListAgents 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGMNL	fteListMonitors
BFGSTLI	fteListScheduledTransfers
BFGTMLI	fteListTemplates
BFGAGMG	fteMigrateAgent 。仅当将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGLGMG	fteMigrateLogger 。仅当将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGCROBS	fteObfuscate 样本
BFGRAS	fteRAS
BFGAGTC	fteSetAgentTraceLevel 。仅在将 SERVICE_TYPE 变量设置为 AGENT 时才创建。
BFGLGTC	第 425 页的『 fteSetLoggerTraceLevel 』。仅当将 SERVICE_TYPE 变量设置为 LOGGER 时才创建。
BFGPRANS	fteAnt 样本
BFGTRCAS	fteCancelTransfer 样本
BFGMNCRS	fteCreateMonitor 样本
BFGTMCRS	fteCreateTemplate 样本
BFGTRCRS	fteCreateTransfer 样本
BFGMNDES	fteDeleteMonitor 样本
BFGSTDES	fteDeleteScheduledTransfer 样本
BFGTMDES	fteDeleteTemplates 样本

> z/OS

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。

> z/OS 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 59 页的『在 z/OS 上配置 MQMFTCredentials.xml』](#)。

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。
- 复审 `BFG_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=DBC
```

其中 `/u/.../sql.properties` 是 Db2 跟踪属性文件的名称，`DBC` 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 `BFG*LG*` 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 72 页的『记录到 Db2 时创建记录任务』](#)

重命名成员 `BFG*LG*CRS`。此作业将更新 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 +
-loggerMgr 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：

Db2 名称	示例
<code>-dbName databaseName</code>	您可以从 Db2 子系统的消息 <code>DSNL004I</code> 内的位置值中获取此信息。
<code>-dbDriver filePath</code>	例如， <code>/db2/db2v10/jdbc/classes/db2jcc.jar</code>
<code>-dbLib filePath</code>	例如， <code>/db2/db2v10/jdbc/lib/libdb2jcc2zos_64.so</code>

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 +
-loggerMgr MQPH +
-loggerType DATABASE +
-dbType DB2 +
-databaseName 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/fteelog_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`。

z/OS 上的 MFT 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 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 Managed File Transfer 脚本会调用 `.wmqfterc` 脚本，您可以使用该脚本为 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 13: 必需 z/OS 环境变量	
环境变量	值
BFG_JAVA_HOME	Java 安装的位置。有关受支持 Java 级别的更多信息，请参阅 IBM MQ 系统需求 。
BFG_DATA	Managed File Transfer for z/OS 的数据目录的位置。这是指向 <i>DATA_DIR</i> 的路径。
STEPLIB	<p>必须包含以下 IBM MQ 数据集：</p> <ul style="list-style-type: none"> • SCSQAUTH • SCSQANLE • SCSQLOAD <p>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：</p> <ul style="list-style-type: none"> • SDSNEXIT • SDSNLOD2 • SDSNLOAD
LIBPATH	必须在 z/OS UNIX 系统服务空间中包含 IBM MQJava 库的位置 (对于 IBM MQ 8.0，缺省值为 <i>/mqm/V8R0M0/java/lib</i>)。

以下是为 Managed File Transfer 正确配置环境变量的示例 `.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 14: 可选 z/OS 环境变量

环境变量	值
BFG_WTO	<p>以下某个值将启用 BFG_WTO:</p> <ul style="list-style-type: none"> • YES • 启用 • TRUE <p>以下某个值将禁用 BFG_WTO。这些值不区分大小写。</p> <ul style="list-style-type: none"> • NULL • 否 • 关闭 • FALSE <p>启用 z/OS 日志记录。缺省情况下，禁用该环境变量。</p> <p>写入到代理事件日志的消息也会写入到 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访问自动化产品。路由代码是 Programmer Information (11)，描述符代码为 Informational (12)。</p>
BFG_GROUP_NAME	<p>mqm 文件组通常与 Managed File Transfer 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 Managed File Transfer 配置。要了解更多信息，请参阅第 291 页的『IBM MQ 中 MFT 的文件系统许可权』。</p> <p>对于 z/OS 系统，文件组是 USS 文件系统实体，并且未必会定义 mqm 文件组。您可以使用 BFG_GROUP_NAME 环境变量，为 Managed File Transfer 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：</p> <pre style="background-color: #f0f0f0; padding: 5px;">export BFG_GROUP_NAME=FTEGB</pre> <p>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。</p> <p>您可以将 BFG_GROUP_NAME 设置为空值或将其移除。</p> <p>V9.0.0.1 如果在 BFG_GROUP_NAME 环境变量设置为非空值的 z/OS 系统上运行 fteMigrateAgent 命令，那么该命令会检查以确定用户是否为 BFG_GROUP_NAME 变量所指定的组的成员。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 403 页的『fteMigrate 代理: 将 FTE 7.0 代理迁移到 MQ 7.5 或更高版本』。</p>

z/OS 配置属性文件

Managed File Transfer 中所使用属性的摘要。

- [第 453 页的『MFT coordination.properties 文件』](#)
- [第 457 页的『MFT command.properties 文件』](#)
- [第 460 页的『MFT agent.properties 文件』](#)
- [记录器配置属性文件](#)

z/OS 为 z/OS 自动重新启动管理器 (ARM) 配置 MFT

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 `SYSBFGLG`，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,SYSBFGLG)

RESTART_GROUP(GROUP7C)
  ELEMENT(SYSMQMGRFT7C)
  ELEMENT(BFGFT7CAG1)
  RESTART_ATTEMPTS(3,300)
```

将 Managed File Transfer for z/OS 与 JZOS Java 启动程序配合使用

您可以在 IBM MQ for z/OS 系统上应用本主题中的指示信息作为在企业中使用 Managed File Transfer 的替代方法。

概述

Managed File Transfer for z/OS (MFT) 使用标准 z/OS 安装过程。运行 MFT 命令的替代方法是使用 JCL 和 JZOS Java Launcher。

请参阅 [JZOS Batch Launcher and Toolkit](#)，以获取更多详细信息。

如果您的 JCL 无法正确地处理，请参阅 [第 251 页的『MFT 中有关 JZOS 的常见问题』](#)。

IBM MQ 8.0 及更高版本的示例 JCL



注意: 对于 IBM WebSphere MQ File Transfer Edition 7.0, 参数以 FTE_ 开头, 而不是以 BFG_ 开头。

```
//JOHNDOEA JOB 1,MSGCLASS=H
// JCLLIB ORDER=(SCEN.MFT.JCL) (1)
// INCLUDE MEMBER=BFGJCL8 (2)
// DD * (2A)
. ${BFG_PROD}/bin/fteBatch createAgent (3)
export IBM_JAVA_OPTIONS="${BFG_JAVA_OPTIONS} ${BFG_LANG}" (4)
export JZOS_MAIN_ARGS="${BFG_MAIN_ARGS}" (4)
//MAINARGS DD *
-agentName MYAGENT (5)
-f
-agentQMgr MQPD
-p MQPD
/*
```

其中:

- (1) 是所包含的 JCL 语句的位置
- (2) 包括从 1) 中的位置指定的 JCL 成员
- (2A) 对 //STDENV 进行扩展 - 请见下文
- (3) 这是要执行的命令, 不带前导 fte 前缀
- (4) 这些行是必需内容, 它们用于设置 JZOS 的信息
- (5) 用于该命令的参数
- BFGJCL8 成员 (您可以选择您自己的名称) 调用 JZOS。该成员包含运行 MFT 所需的 STEPLIB 和其他 JCL。

您需要包括的其他 JCL

您应该包括 IBM MQ for z/OS 库的 JCL, 如果您使用的是 Db2 记录器, 则要包括 Db2 库的 JCL。

例如:

```
//WMQFTE EXEC PGM=JVMLDM86,REGION=0M PARM='+T' (1)
//STEPLIB DD DSN=SYS1.SIEALNKE,DISP=SHR (2)
//* MQ libraries
// DD DSN=MQM.V800.SCSQAUTH,DISP=SHR MQ Bindings
// DD DSN=MQM.V800.SCSQANLE,DISP=SHR MQ Bindings
// DD DSN=MQM.V800.SCSQLOAD,DISP=SHR MQ Bindings

//* DB2 libraries
// DD DISP=SHR,DSN=SYS2.DB2.V10.SDSNEXIT.DBCP
// DD DISP=SHR,DSN=SYS2.DB2.V10.SDSNLOAD
// DD DISP=SHR,DSN=SYS2.DB2.V10.SDSNLOD2
//SYSOUT DD SYSOUT=H
//SYSPRINT DD SYSOUT=H
//STDOUT DD SYSOUT=H
//STDERR DD SYSOUT=H

//STDENV DD DSN=SCEN.MFT.JCL(BFGZENV8),DISP=SHR (3)
```

其中:

- (1) 是 JZOS 程序的名称。查看 SYS1.SIEALNKE 以了解您系统上的版本。添加 ,PARM='+T' 以提供其他诊断。
- (2) 这是带有 JZOS 程序的数据集。
- (3) 这是 shell 脚本的成员名。用于定义 MFT 所需的参数。请参阅第 78 页的『用于定义 MFT 的 shell 脚本』。

它可以是任何数据集和成员。该名称需要位于文件末尾, 因为 JCL 作业对此进行了扩展。请参阅第 77 页的『IBM MQ 8.0 及更高版本的示例 JCL』中的 2A。

用于定义 MFT 的 shell 脚本

在第 77 页的『您需要包括的其他 JCL』示例中，使用了成员 BFGZENV8。这是基于 JZOS 概要文件。

您可以为 MFT V8 和 IBM WebSphere MQ File Transfer Edition 7.0 使用相同的配置文件，只需要进行稍许更改。请记住，在 MFT V8 之前，参数以 FTE 开头。请参阅第 78 页的『示例文件』。

您需要知道：

- 安装了 Java 的位置
- IBM MQ for z/OS Java 库的位置
- MFT 文件的位置
- 用户标识需要位于特定组内才能被视为 IBM MQ for z/OS 管理员。您需要此组的名称
- 如果消息不以英语表示，您需要知道要指定哪种语言。

示例文件

```
# This is a shell script that configures
# any environment variables for the Java JVM.
# Variables must be exported to be seen by the launcher.
# Use PARM='+T' and set -x to debug environment script problems
set -x
# . /etc/profile
#
# Java configuration (including MQ Java interface)
#
export _BPXK_AUTOCVT="ON"
export JAVA_HOME="/java/java71_bit64_sr3_fp30/J7.1_64/"
export PATH="/bin:${JAVA_HOME}/bin/classic/"
LIBPATH="/lib:/usr/lib:${JAVA_HOME}/bin"
LIBPATH="$LIBPATH:${JAVA_HOME}/bin/classic"
LIBPATH=$LIBPATH:"/mqm/V8R0M0/java/lib/"
export LIBPATH

export BFG_JAVA_HOME="${JAVA_HOME}"
export BFG_WTO="YES"
export BFG_GROUP_NAME=MQADM
export BFG_PROD="/HMF8800/"
export BFG_CONFIG="/u/johndoe/fteconfig"
# export BFG_LANG=" -Duser.language=de "
export BFG_LANG=" "
```

其中：

export _BPXK_AUTOCVT="ON"

对于 Unicode 转换是必须项

export JAVA_HOME="/java/java71_bit64/J7.1_64/"

是 Java 目录的位置。指定 Java 的路径名称。该目录包含 bin 和其他目录。

export PATH="/bin:\${JAVA_HOME}/bin/classic/"

设置 Java 可执行语句的路径语句

LIBPATH="/lib:/usr/lib:\${JAVA_HOME}/bin"

设置 Java 可执行语句的库路径

LIBPATH="\$LIBPATH:\${JAVA_HOME}/bin/classic"

在 LIBPATH 语句中添加更多 Java 库。

LIBPATH=\$LIBPATH:"/mqm/V8R0M0/java/lib/"

在库路径中添加 IBM MQ for z/OS 库。指定 USS 中的 IBM MQ for z/OS 库的名称。

export LIBPATH

使 LIBPATH 可用于 JZOS

export BFG_JAVA_HOME="\${JAVA_HOME}"

将 BFG_JAVA_HOME 设置为以上指定的 JAVA_HOME 的值

export BFG_WTO="YES"

将 BFG_WTO 设置为 YES 会使用 WTO 将消息显示在作业日志中

export BFG_GROUP_NAME=MQADM

属于指定组成员的用户标识，将视为 IBM MQ for z/OS 管理员

export BFG_PROD="/HMF8800/"

是 MFT 代码所在位置的路径

export BFG_DATA="/u/johndoe/fteconfig"

是存储 MFT 配置信息的位置

export BFG_LANG="-Duser.language=de "

将语言定义为德语的已注释掉的语句

export BFG_LANG=" "

指定语言为缺省值英语。

`/lib/messages/BFGNVMessages_*.properties` 中 MFT 产品的内容列出了可用的语言。缺省值为将值保留为空白，表示将使用英语。

对于 V7，请指定：

```
export FTE_JAVA_HOME="${JAVA_HOME}"
export FTE_WTO="YES"
export FTE_GROUP_NAME=SCENU
export FTE_PROD="/HMF7100/"
export FTE_CONFIG="/u/johndoe/fteconfig"
export BFG_LANG=" "
```

请注意，`/u/johndoe/fteconfig` 与 `BFG_DATA` 的路径不同

相关信息

[Managed File Transfer for z/OS](#)

[规划 Managed File Transfer](#)

IBM i 在 IBM i 上配置 MFT

要在安装后开始使用 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 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) 从 IBM i 命令行使用以下命令启动 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 82 页的『为 MFT 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) 从 IBM i 命令行使用以下命令启动 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 83 页的『配置 MFT 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) 从 IBM i 命令行使用以下命令启动 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。 [第 39 页的『使用 MFT 代理沙箱』](#) 中对此功能进行了描述。

相关概念

[第 80 页的『为首次使用配置 MFT』](#)

在首次使用 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 MFT

在首次使用 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。

连接到 IBM MQ

与 IBM MQ 队列管理器的所有网络通信 (包括与 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。

Managed File Transfer 和通道

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`) 连接到其命令队列管理器和协调队列管理器。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 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 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。

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，请参阅针对您的企业所用平台的“验证 IBM MQ 安装”。

相关概念

第 86 页的『配置多实例队列管理器以使用 MFT』

从 IBM WebSphere MQ 7.0.1 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。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 81 页的『配置 MFT 网络队列管理器』

如果您的 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 82 页的『为 MFT 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 MFT 网络队列管理器

如果您的 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 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 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 WebSphere MQ 7.5 或 Managed File Transfer 代理程序中的 Managed File Transfer 进行通信，请确保发送方通道的 CONVERT 参数设置为 no。（IBM WebSphere MQ File Transfer Edition 的早期版本始终以 UTF-8 格式发布消息，这意味着任何数据转换都会损坏消息。对于 IBM MQ 8.0 或更高版本的 Managed File Transfer 上的代理程序，这不是必需的，因为将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 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](#)。

为 MFT 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 (NPMSGDLV) 属性和持久性消息传递 (PMSGDLV) 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` 将重新创建缺省的 IBM MQ 对象并撤销 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 80 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 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 86 页的『配置多实例队列管理器以使用 MFT』](#)

从 IBM WebSphere MQ 7.0.1 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。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 81 页的『[配置 MFT 网络队列管理器](#)』

如果您的 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 430 页的『[fteSetupCoordination](#)』

fteSetupCoordination 命令为 Managed File Transfer 创建属性文件和协调队列管理器目录。

配置 MFT 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，如果要手动执行此配置，请在代理队列管理器上完成以下步骤。

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 518 页的『MFT 代理队列设置』](#)。

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 518 页的『MFT 代理队列设置』](#)。

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 327 页的『fteCreateBridgeAgent（创建并配置 MFT 协议网桥代理）』](#)和 [第 190 页的『配置 FTPS 服务器的协议网桥』](#)。

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 Managed File Transfer 结构。

关于此任务

MFT 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 MFT IBM MQ 8.0 队列管理器以及名为 SAMPLEAGENT 的单个 MFT 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 IBM MQ 8.0 或更高版本的 MFT 中启用了安全性的配置, **CONNAUTH** 需要所有与队列管理器连接的 MFT 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 36 页的『[MFT 和 IBM MQ 连接认证](#)』。

注意:

- 要查找 MFT 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进行查找, 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. 创建协调队列管理器。

协调队列管理器是一个队列管理器, 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. 创建命令队列管理器。

命令队列管理器是已预先配置的一个队列管理器，以便 IBM MQ 基础结构可以将 MFT 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，如果针对缺省协调设置了命令，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
```

这将在协调下创建代理配置，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 MFT 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 Explorer 可用，使用 MFT 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 430 页的『fteSetupCoordination』](#)

fteSetupCoordination 命令为 Managed File Transfer 创建属性文件和协调队列管理器目录。

[第 428 页的『fteSetupCommands: 创建 MFT command.properties 文件』](#)

fteSetupCommands 命令创建 Managed File Transfer 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 323 页的『fteCreateAgent \(创建 MFT 代理\)』](#)

fteCreateAgent 命令创建 Managed File Transfer Agent 及其关联的配置。

[第 413 页的『fteObfuscate: 对敏感数据进行加密』](#)

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 691 页的『MFT 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 460 页的『MFT agent.properties 文件』](#)

每个 Managed File Transfer Agent 都有自己的属性文件 agent.properties，该文件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 251 页的『当 fteListAgents 命令未列出您的 MFT 代理时该怎么做』](#)

如果代理未通过 **fteListAgents** 命令列出，或者未显示在 IBM MQ Explorer 中，或者文件传输未显示在 IBM MQ Explorer 的传输日志中，那么您可以执行许多问题确定步骤来调查原因。

配置多实例队列管理器以使用 MFT

从 IBM WebSphere MQ 7.0.1 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。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 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 460 页的『MFT 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 Managed File Transfer 拓扑中的所有 coordination.properties 文件。

请参阅第 453 页的『MFT 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 coordination.properties 文件的内容。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)
```

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 105 页的『MFT 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 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。

Managed File Transfer 命令 **ftelist*** 和 **fteShowAgentDetails** 可直连到协调队列管理器。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

要启用与多实例命令队列管理器的活动实例和备用实例的连接，请将 connectionQMgrStandby 属性添加到 Managed File Transfer 拓扑中的所有 command.properties 文件。

请参阅第 457 页的『MFT 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_MARS
connectionQMgrHost=command_host1
connectionQMgrPort=1424
connectionQMgrChannel=SYSTEM.DEF.SVRCONN
connectionQMgrStandby=command_host2(1424)
```

相关概念

第 80 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 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 81 页的『配置 MFT 网络队列管理器』

如果您的 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 82 页的『为 MFT 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。

保留 MFT 日志消息

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 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，用于将发布存储在 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 Managed File Transfer 数据库记录器保留日志消息

数据库记录器是 Managed File Transfer 的可选组件，可使用它将日志信息复制到数据库中以便于分析和审计目的。数据库记录器是您在托管协调队列管理器和数据库的系统上安装的独立 Java 应用程序。有关数据库记录器的更多信息，请参阅第 89 页的『配置 MFT 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。

持久预订的名称添加了前缀以显示该预订是通过 IBM MQ Explorer MFT 插件、主机名和用户名创建的，例如，MQExplorer_MFT_Plugin_HOST_TJWatson。

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 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 > 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)WSHEMA(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

您可以将管理自己的持久预订的应用程序写入 SYSTEM.FTE 主题，使用 IBM MQ 支持的其中一个应用程序编程接口。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 MFT 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

要点: IBM i 平台上不支持 Managed File Transfer 记录器。

独立文件记录器

独立文件记录器是一种 Java 进程，它在托管协调队列管理器的系统上运行，或在托管连接到协调队列管理器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。

您可以将独立文件记录器作为 Windows 服务运行，以确保在从 Windows 会话注销时文件记录器继续运行，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 90 页的『[安装 MFT 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在托管队列管理器和数据库的系统上安装的 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。

如果使用的是 Windows 系统，那么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 96 页的『[安装 MFT 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 7.0 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 90 页的『安装 MFT 独立文件记录器』](#)
- [第 96 页的『安装 MFT 独立数据库记录器』](#)
- [第 105 页的『为 MFT 安装 Java EE 数据库记录器』](#)

安装 MFT 独立文件记录器

独立文件记录器是一种 Java 进程，它必须使用 IBM MQ 绑定来连接到协调队列管理器。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 89 页的『配置 MFT 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 105 页的『MFT 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 Managed File Transfer Logger 组件。有关更多信息，请参阅 [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。要了解更多信息，请参阅第 340 页的『fteCreate 记录器 (创建 MFT 文件或数据库记录器)』。
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 90 页的『MFT 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 95 页的『为 MFT 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 98 页的『MFT 记录器配置属性』。
7. **Windows**
可选：如果使用的是 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 411 页的『fteModify 记录器 (将 MFT 记录器作为 Windows 服务运行)』。
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 444 页的『fteStartLogger』。

如果在 Windows 上执行上一步并使用带有 `-s` 参数的 **fteModifyLogger** 命令，那么独立文件记录器作为 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 447 页的『fteStopLogger』。

结果

MFT 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。有关缺省日志格式定义的更多信息，请参阅第 510 页的『MFT 独立文件记录器缺省日志格式』。

如果要指定您自己的定制日志格式，请编辑 `FileLoggerFormat.xml` 文件。

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;414d5120514d5f67627468696e6b202064666cf4e20004f02;[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 514 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 517 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 632 页的『文件传输日志消息格式』](#)
- [第 653 页的『已调度的文件传输日志消息格式』](#)
- [第 658 页的『MFT 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` - 以逗号分隔的 `key=value` 格式编写为传输定义的元数据条目集。

下表列出了每种消息类型的系统定义插入的“type”值。

消息类型	“type”系统插入的值
callCompleted	[CCOM]
callStarted	[CSTR]
monitorAction	[MACT]
monitorCreate	[MCRT]
monitorFired	[MFIR]
notAuthorized	[AUTH]
scheduleDelete	[SDEL]
scheduleExpire	[SEXP]
scheduleSkipped	[SSKP]
scheduleSubmitInfo	[SSIN]
scheduleSubmitTransfer	[SSTR]
scheduleSubmitTransferSet	[SSTS]
transferStarted	[TSTR]
transferCancelled	[TCAN]
transferComplete	[TCOM]
transferDelete	[TDEL]

表 15: 受支持消息类型及其“type”系统插入的摘要。(继续)

消息类型	“type”系统插入的值
transferProgress	[TPRO]

从 MFT 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。

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 MFT 独立文件记录器定义定制格式

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 510 页的『MFT 独立文件记录器缺省日志格式』
Managed File Transfer 独立文件记录器的缺省日志文件格式定义

第 514 页的『独立文件记录器格式 XSD』
独立文件格式的模式。

减少 MFT 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，可能会将日志消息写入独立文件记录器的日志中，而不会使用 SYSTEM.FTE/Log# 主题正在落实到 IBM MQ。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 98 页的『MFT 记录器配置属性』。

为 MFT 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 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 `dspmqaut`（显示权限）或 `dmpmqaut`（转储权限）。
4. 使用 IBM MQ Explorer 或 `setmqaut` (`grant` 或 `revoke authority`) 命令中的“管理权限记录”窗口来添加用户自己的组的权限（在 UNIX 上，IBM MQ 权限仅与组关联，而不与个人用户关联）。所需权限如下所示：
 - 对队列管理器的 Connect 和 Inquire 许可权（IBM MQ Java 库需要 Inquire 许可权才能运行）。
 - 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 MFT 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

要点: IBM i 平台上不支持 Managed File Transfer 记录器。

有关独立数据库记录器的更多信息，请参阅第 89 页的『配置 MFT 记录器』。

注: 在任何一个时间都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 Java EE 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 340 页的『fteCreate 记录器 (创建 MFT 文件或数据库记录器)』。

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 98 页的『MFT 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。

Multi 在多平台上，`ftelog_tables_db2.sql` 和 `ftelog_tables_oracle.sql` 文件包含可运行来创建表的 SQL 命令。

z/OS 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 96 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 98 页的『MFT 记录器配置属性』。

7. **Windows**

可选：如果使用的是 Windows 系统，那么可以将独立数据库记录器作为 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 411 页的『fteModify 记录器 (将 MFT 记录器作为 Windows 服务运行)』。

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。要了解更多信息，请参阅第 691 页的『MFT 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。

如果在 Windows 上执行上一步并使用带有 `-s` 参数的 `fteModifyLogger` 命令，那么独立数据库记录器作为 Windows 服务启动。

将 MFT 与远程数据库配合使用

您可以使用 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 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 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。Managed File Transfer 和 IBM MQ 需要此配置更新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：Managed File Transfer 的较低版本使用属性 `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 98 页的『MFT 记录器配置属性』

Managed File Transfer 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 MFT 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 `fteelog`，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“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 IBM 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）。
4. 使用 IBM MQ Explorer 或 `setmqaut` (`grant` 或 `revoke authority`) 命令中的“管理权限记录”窗口来添加用户自己的组的权限（在 UNIX 上，IBM MQ 权限仅与组关联，而不与个人用户关联）。所需权限如下所示：
 - 对队列管理器的 `Connect` 和 `Inquire` 许可权（IBM MQ Java 库需要 `Inquire` 许可权才能运行）。
 - 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 `fteelog` 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 `fteelog` 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](#)

MFT 记录器配置属性

Managed File Transfer 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 7.5 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 448 页的『在 MFT 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 [Properties 类的 Javadoc](#)。

绑定方式连接属性




表 16: 绑定方式的连接属性		
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。此参数是必需的, 并且是与队列管理器的绑定方式连接所需的所有参数。(有关用于连接到远程队列管理器的属性, 请参阅第 104 页的表 17。)	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, 以便队列接收通常由 SYSTEM.FTE/Log/#。	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 7.5 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 691 页的『MFT 凭证文件格式』。	 有关创建认证凭证文件的详细信息, 请参阅 创建 MFT 凭证文件  请参阅 配置 MQMFTCredentials.xml , 以获取有关此文件的位置和许可权的信息。  有关创建认证凭证文件的更多详细信息, 请参阅 MFT 和 IBM MQ 连接认证






表 16: 绑定方式的连接属性 (继续)		
属性名	描述	缺省值
wmqfte.database.driver	<p>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, AIX 系统上 Db2 的类型 2 驱动程序需要文件 /opt/IBM/db2/V9.5/java/db2jcc.jar。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。</p> <p> 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。</p> <p> 在 z/OS 系统上, 必须引用以下所有 JAR 文件:</p> <ul style="list-style-type: none"> • db2jcc.jar • db2jcc_license_cisuz.jar • db2jcc_javax.jar <p>如果数据库驱动程序由多个 JAR 文件组成 (例如, Db2 9.1 需要驱动程序 JAR 文件和许可证 JAR 文件), 请在此属性中包含所有这些 JAR 文件。通过针对平台使用类路径分隔符 (即, 在 Windows 系统上为分号字符 (;), 在其他平台上为冒号字符 (:)), 分隔多个文件名。</p>	无缺省值
wmqfte.database.exclude_duplicate_metadata	<p>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:</p> <ul style="list-style-type: none"> • 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 <p>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。</p>	true
wmqfte.database.host	<p>仅限 Db2:</p> <p>对于 IBM WebSphere MQ 7.5 或更高版本, 是指要使用 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, 那么也必须指定 wmqfte.database.port 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。</p> <p>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 wmqfte.database.credentials.file 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。</p>	无缺省值
wmqfte.database.name	包含 Managed File Transfer 日志表的数据库实例 (或使用 Db2 对 z/OS 时的子系统) 的名称。	无缺省值

表 16: 绑定方式的连接属性 (继续)		
属性名	描述	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	<p>仅限 Db2:</p> <p>对于 IBM WebSphere MQ 7.5 或更高版本, 是指要使用 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, 那么也必须指定 <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。</p> <p>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。</p>	无缺省值
wmqfte.database.schema	包含 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	<p>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, AIX 系统上 Db2 的类型 2 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。</p> <p>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。</p>	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	<p>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。</p> <p><code>wmqfte.file.logger.fileSize=2d</code> 指定最大文件大小为 2 天的数据。</p>	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

表 16: 绑定方式的连接属性 (继续)

属性名	描述	缺省值
wmqfte.file.logger.mode	<p>正在使用的记录器方式: circular 或 linear。将该值设置为 CIRCULAR 或 LINEAR。</p> <p>CIRCULAR - 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。在该方式下写入了信息的文件的最大数量由使用 wmqfte.file.logger.fileCount 属性定义的值来控制。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), 那么最大大小将取决于这些时间段内系统中的日志消息吞吐量。在该方式下运行时使用的日志文件命名约定为: <i>logger_namenumbers-timestamp.log</i>, 其中:</p> <ul style="list-style-type: none"> • <i>logger_name</i> 是在 fteCreateLogger 命令中为记录器指定的名称。 • <i>number</i> 是文件在集合内的编号。 • <i>timestamp</i> 是创建文件时的时间戳记。 <p>例如, LOGGER1-20111216123430147.log</p> <p>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。在 linear 方式下运行时不会删除文件, 从而将忽略 wmqfte.file.logger.fileCount 属性, 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。在该方式下运行时使用的日志文件命名约定为: <i>logger_name-timestamp.log</i>, 其中:</p> <ul style="list-style-type: none"> • <i>logger_name</i> 是在 fteCreateLogger 命令中为记录器指定的名称。 • <i>timestamp</i> 是创建文件时的时间戳记。 <p>例如, LOGGER-20111216123430147.log</p>	无缺省值
wmqfte.max.retry.interval	<p>记录器遇到持续性错误时, 两次重试之间的最长时间 (秒)。</p> <p>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。</p> <p>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。</p>	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

表 16: 绑定方式的连接属性 (继续)		
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。	 有关创建认证凭证文件的详细信息, 请参阅 创建 MFT 凭证文件  请参阅 配置 MQMFTCredentials.xml , 以获取有关此文件的位置和许可权的信息。  有关创建认证凭证文件的更多详细信息, 请参阅 MFT 和 IBM MQ 连接认证
跟踪	<p>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。</p> <p>例如, com.ibm.wmqfte.databaselogger、com.ibm.wmqfte.databaselogger.operation=all。</p> <p>您可以在冒号分隔的列表中指定多个跟踪规范。例如,</p> <pre>com.ibm.wmqfte.databaselogger=moderate: com.ibm.wmqfte.databaselogger.operation=all</pre>	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

客户机方式连接属性

V 9.0.4

下表显示了支持到记录器队列管理器的客户机方式连接时需要的其他记录器属性。

表 17: 客户机方式的连接属性		
属性名	描述	缺省值
wmqfte.queue.manager.host	记录器队列管理器的主机名或 IP 地址。	无缺省值
wmqfte.queue.manager.port	记录器队列管理器正在侦听的端口。	1414
wmqfte.queue.manager.channel	记录器队列管理器上服务器连接通道的名称。	SYSTEM.DEF.SVRCONN
wmqfte.Ssl.CipherSuite	指定记录器和记录器队列管理器交换数据方式的 TLS 方面。 wmqfte.Ssl.CipherSuite 的值为 CipherSuite 名称。CipherSuite 名称将映射到记录器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。	无缺省值
wmqfte.Ssl.PeerName	指定必须与记录器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。	无缺省值
wmqfte.Ssl.TrustStore	指定记录器信任的证书的位置。 wmqfte.Ssl.TrustStore 的值为文件路径。如果文件路径是 Windows 文件路径, 那么反斜杠字符 (\) 必须使用另一个反斜杠字符进行转义 (\\)。 请注意, 此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.TrustStoreCredentialsFile	包含 wmqfte.Ssl.TrustStore 凭证的文件的 路径。 请注意, 此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.TrustStoreType	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
wmqfte.Ssl.KeyStore	指定记录器的专用密钥的位置。 wmqfte.Ssl.KeyStore 的值是文件路径。如果文件路径是 Windows 文件路径, 那么反斜杠字符 (\) 必须使用另一个反斜杠字符进行转义 (\\)。 请注意, 此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.KeyStore.CredentialsFile	包含 wmqfte.Ssl.KeyStore 凭证的文件的 路径。 请注意, 此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.KeyStoreType	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
wmqfte.Ssl.FipsRequired	指定要在记录器级别启用 FIPS 支持。该属性的值可以为 true 或 false。有关更多信息, 请参阅第 555 页的『MFT 中的 FIPS 支持』。	false

相关参考

[第 448 页的『在 MFT 属性中使用环境变量』](#)

在 IBM WebSphere MQ 7.5 中, 可以在 Managed File Transfer 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 504 页的『MFT 的 SSL 属性』](#)

将 SSL 或 TLS 与 IBM MQ 和 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。

MFT 独立记录器的备用配置

通常，不管 Managed File Transfer 独立记录器是文件还是数据库类型，它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另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 IBM WebSphere MQ 7.5 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
```

为 MFT 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 数据库记录器以配合 Managed File Transfer 一起使用。

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 89 页的『[配置 MFT 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 106 页的『准备为 MFT 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 109 页的『使用 WebSphere Application Server 7.0 安装 MFT 的 Java EE 数据库记录器』
 - 第 112 页的『将 MFT 的 Java EE 数据库记录器与 WebSphere Application Server Community Edition 一起安装』

相关任务

第 106 页的『准备为 MFT 安装 Java EE 数据库记录器』

在安装 Java EE 数据库记录器之前，请按照以下指示信息来准备 Managed File Transfer 环境。

第 109 页的『使用 WebSphere Application Server 7.0 安装 MFT 的 Java EE 数据库记录器』

遵循以下指示信息，为 Managed File Transfer with WebSphere Application Server 7.0 安装和配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器。

第 112 页的『将 MFT 的 Java EE 数据库记录器与 WebSphere Application Server 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 Application Server Community Edition 一起使用。

第 115 页的『为 MFT 的 Java EE 数据库记录器配置用户访问权』

为 Managed File Transfer 配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器时，需要用户帐户才能访问 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 117 页的『从独立数据库记录器迁移至 MFT 的 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 Java EE 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 JEE 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。

相关参考

第 301 页的『MFT 记录器的权限』

运行记录器的操作系统用户需要对记录器队列和 SYSTEM.FTE 主题具有特定 IBM MQ 权限。

准备为 MFT 安装 Java EE 数据库记录器

在安装 Java EE 数据库记录器之前，请按照以下指示信息来准备 Managed File Transfer 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 89 页的『配置 MFT 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。

 在多平台上，ftelog_tables_db2.sql 和 ftelog_tables_oracle.sql 文件包含可运行来创建表的 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 107 页的『在 MFT 的 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 109 页的『使用 WebSphere Application Server 7.0 安装 MFT 的 Java EE 数据库记录器』
- 第 112 页的『将 MFT 的 Java EE 数据库记录器与 WebSphere Application Server Community Edition 一起安装』

在 MFT 的 Java EE 数据库记录器中更改模式名称

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 Java EE 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.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。

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 109 页的『使用 WebSphere Application Server 7.0 安装 MFT 的 Java EE 数据库记录器』](#)

遵循以下指示信息, 为 Managed File Transfer with WebSphere Application Server 7.0 安装和配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器。

[第 112 页的『将 MFT 的 Java EE 数据库记录器与 WebSphere Application Server 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 Application Server Community Edition 一起使用。

在 WebSphere Application Server 7.0 中设置本机库路径

如果在 WebSphere Application Server 7.0 上部署 Java Platform, Enterprise Edition (Java EE) 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 本机库的位置配置 IBM MQ 消息传递提供程序。

关于此任务

如果未在应用程序服务器中设置本机库路径, 那么您可能会在 WebSphere Application Server 7.0 系统输出日志中收到以下错误消息:

```
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]
```

使用 WebSphere Application Server 7.0 管理控制台以完成以下步骤:

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。

使用 WebSphere Application Server 7.0 安装 MFT 的 Java EE 数据库记录器

遵循以下指示信息，为 Managed File Transfer with WebSphere Application Server 7.0 安装和配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器。

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 106 页的『[准备为 MFT 安装 Java EE 数据库记录器](#)』和第 108 页的『[在 WebSphere Application Server 7.0 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 89 页的『[配置 MFT 记录器](#)』。

过程

1. 设置 XA JDBC 提供程序：
 - a) 从 WebSphere Application Server 7.0 管理控制台导航中选择 **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 7.0 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，请参阅为 [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 7.0 管理控制台导航中选择 **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 帮助程序类名**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 7.0 管理控制台导航中选择 **资源 > JDBC > 数据源**。
 - b) 单击**测试连接**按钮。
- 5. 创建主题。
 - a) 在 WebSphere Application Server 7.0 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) 在 WebSphere Application Server 7.0 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: 单击**测试连接**以确认队列管理器存在。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, 向下滚动到“**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。您必须完成该步骤, 否则 IBM MQ 将拒绝该连接并返回 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**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 275 页的『MFT 记录器错误处理和拒绝』。
7. 创建队列连接工厂。
- 在 WebSphere Application Server 7.0 管理控制台导航中，单击**资源 > JMS > 队列连接工厂**。
 - 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**。
 - 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。
 - 在步骤 2 中，选择在该向导中输入**所有必需信息**。
 - 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。
 - 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。
 - 可选：单击**测试连接**以确认队列管理器存在。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。
 - 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：
- 在 WebSphere Application Server 7.0 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 7.0 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 Managed File Transfer Service 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。

- g) 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**。

- h) 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**。
- i) 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**。
- j) 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。

- k) 在向导的步骤 9 **模块的元数据**中，单击**下一步**。

- l) 在向导的步骤 10 **摘要**中，单击**完成**。

10. 您现在可以从 WebSphere Application Server 7.0 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 MFT 的 Java EE 数据库记录器与 WebSphere Application Server 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 Application Server 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 106 页的『[准备为 MFT 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 89 页的『[配置 MFT 记录器](#)』。

过程

1. 部署 IBM MQ 资源适配器 `wmq.jmsra.rar`。

- 要使用协调队列管理器 `QM_JUPITER` 部署 JEE 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 Application Server 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。
 - 创建计划文件以定义与 MFT 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/WMQFTEJEEEDBLoggerRejectQueueCF</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/WMQFTEJEEEDBLoggerRejectQueue</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>

```

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 Application Server 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 Application Server 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 Application Server 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/WMQFTEJEEEDBLoggerRejectQueueCF</name>
      <config-property-setting name="queueManager">QM_SATURN</config-property-setting>
      <config-property-setting name="transportType">CLIENT</config-property-setting>
    </connectiondefinition-instance>
  </connection-definition>
</connector>

```

```

<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-interface>javax.jms.Queue</adminobject-interface>
  <adminobject-class>com.ibm.mq.connector.outbound.MQQueueProxy</adminobject-class>
  <adminobject-instance>
    <message-destination-name>jms/WMQFTEJEEEDBLoggerRejectQueue</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 Application Server CE 的系统。
 - c. 打开 WebSphere Application Server 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 Application Server Community Edition 环境访问所需数据库。
- 从 WebSphere Application Server Community Edition 管理控制台执行以下步骤：
- a) 根据所使用的 WebSphere Application Server 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**，根据您的数据库，选择 DB2 XA 或 Oracle Thin。
 - 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 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.databaselogger.jee.ejb.jar
```

其中 *ear_file_name* 是 `com.ibm.wmqfte.databaselogger.jee.ear` 或 `com.ibm.wmqfte.databaselogger.jee.oracle.ear`，具体取决于您使用的是 Db2 还是 Oracle。EAR 文件位于 IBM WebSphere MQ File Transfer Edition 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.databaselogger.jee.ejb.jar` 中抽取 META-INF/openejb-jar.xml 文件：

```
jar -xf com.ibm.wmqfte.databaselogger.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.databaselogger.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.databaselogger.jee.ejb.jar
```

其中 *ear_file_name* 是 `com.ibm.wmqfte.databaselogger.jee.ear` 或 `com.ibm.wmqfte.databaselogger.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 Application Server Community Edition 管理控制台完成以下步骤。

- a) 从**控制台导航菜单**中选择**应用程序 > 新部署**。
- b) 在**归档**字段中，指定 EAR 文件 `com.ibm.wmqfte.databaselogger.jee.ear` 或 `com.ibm.wmqfte.databaselogger.jee.oracle.ear`，具体取决于您的数据库。
- c) 将**计划**字段留空。
- d) 确保选中了**安装后启动应用程序框**。
- e) 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 MFT 的 Java EE 数据库记录器配置用户访问权

为 Managed File Transfer 配置 Java Platform, Enterprise Edition (Java EE) 数据库记录器时，需要用户帐户才能访问 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在不同系统上安装 Java EE 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 `ftelog`，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 IBM 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 `dspmqaout`（显示权限）或 `dmpmqaout`（转储权限）。
2. 使用 IBM 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：
 - 对队列管理器的 `CONNECT` 和 `INQUIRE` 许可权（IBM MQ Java 库需要 `INQUIRE` 许可权才能运行）。
 - 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 `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，`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

迁移 Java EE 数据库记录器

要将 WebSphere Application Server 7.0 上的 Java EE 数据库记录器从 IBM WebSphere MQ File Transfer Edition 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 JEE 数据库记录器设置的配置设置。稍后在步骤 [第 117 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 [通过 WebSphere Application Server 7.0 安装 IBM MQ Managed File Transfer Java EE 数据库记录器](#) 中的信息以及先前在步骤 [第 117 页的『3』](#) 中记录的信息，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。


从独立数据库记录器迁移至 MFT 的 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 Java EE 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 JEE 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。

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 PSMODE(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 `fteStopLogger` 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，如果数据库模式位于 7.0.1，并且您要迁移到 7.0.4，那么必须将模式从 7.0.1 迁移到 7.0.2，然后从 7.0.2 迁移到 7.0.3，然后从 7.0.3 迁移到 7.0.4。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 *old* 迁移到 *new*（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 7.0.2 和 7.0.3 模式之间或者在 7.0.3 和 7.0.4 模式之间进行迁移，那么必须创建新的数据库模式并将现有数据复制到其中。有关更多信息，请参阅第 19 页的『将 Db2 on z/OS 上的数据库表迁移至 IBM MQ 8.0』。
 - 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 105 页的『为 MFT 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

配置 Connect:Direct 网桥

配置 Connect:Direct 网桥，可在 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 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 118 页的『针对 Connect:Direct 网桥代理和节点选择操作系统』。
2. 第 119 页的『选择和配置 Connect:Direct 节点』。
3. 第 119 页的『创建和配置 Connect:Direct 网桥代理』。
4. 第 120 页的『配置 ConnectDirectNodeProperties.xml 文件以包含有关远程 Connect:Direct 节点的信息』。
5. 第 120 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。此文件系统用于在涉及 Connect:Direct 网桥的文件传输期间将文件临时存储在由 `cdTmpDir` 参数定义的目录中。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：

- 代理和节点在同一系统上，该系统运行 Windows 或 Linux for System x
- 代理在 Linux for System x 上，而该节点在 UNIX 上

- 代理在一个 Windows 系统上，而节点在另一个 Windows 系统上

以下配置不允许代理和节点使用同一路径名：

- 代理在 Linux for System x 上，而该节点在 Windows 上
- 代理在 Windows 上，而该节点在 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 [IBM MQ 系统需求](#)。

过程

1. 选择运行 Windows 或 Linux on System x 的系统，以用于安装 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，以用于安装 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 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 节点正在 Windows 上运行，那么使用 Connect:Direct 请求程序来编辑网络映射。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 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 335 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 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 120 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 123 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：

- 在 nodeProperties 元素中，创建 node 元素。
- 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。
- 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。
- 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 节点的操作系统的。

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 490 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 46 页的『[在 Connect:Direct 网桥代理与 Connect:Direct 节点之间配置 SSL 或 TLS](#)』。

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 Managed File Transfer 中的用户凭证映射到 Connect:Direct 节点上的用户凭证。Managed File Transfer 提供了执行用户凭证映射的样本用户出口。

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 Managed File Transfer 中的用户凭证映射到 Connect:Direct 节点上的用户凭证。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 487 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。要使用其他位置，请使用 `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` 是一个模式，用于匹配与 MFT 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，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 value: yes
Issue the submit Process statement value: all
Monitor, submit, change, and delete all Processes value: all
Access Process statistics value: yes
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，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。该属性的值是 `<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>` 元素指定由作为 `Connect:Direct` 网桥组成部分的 `Connect:Direct` 节点使用的凭证。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 118 页的『配置 `Connect:Direct` 网桥』](#)

配置 `Connect:Direct` 网桥，以在 `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 `Managed File Transfer` 代理。该代理称为 `Connect:Direct` 网桥代理。

相关参考

[第 487 页的『`Connect:Direct` 凭证文件格式』](#)

`Managed File Transfer Agent` 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 335 页的『`fteCreateCDAgent`（创建 `Connect:Direct` 网桥代理）』](#)

`fteCreateCDAgent` 命令用于创建 `Managed File Transfer Agent` 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 *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

针对映射 *Connect:Direct* 凭证而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 755 页的『[CDCredentialExit.java 接口](#)』。

在 *Connect:Direct* 网桥代理与 *Connect:Direct* 节点之间配置 SSL 或 TLS

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 IBM 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，可以使用现有的密钥库和信任库。有关更多信息，请参阅第 44 页的『[为 MFT 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。
您可以使用 IBM MQ 随附的 IBM 密钥管理工具来执行此操作。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-----
MIIcCzCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEWJHqjES
MBAGA1UECBMJSjGfTcHNoaXJlMRAwDgYDVQQHEwdIdXJzbGV5MjV5MjV5MjV5MjV5
Qk0xMjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5
Fw0yMTAyMjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5
cmUxMjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5MjV5
ZzCBnzANBgkqhkiG9w0BAQEFAA0BjQAwgYkCgYEAvgP1QIk1U9ypSKD1Xo0Do1yk
EyMFXB0UpZrDvXj0SEC0vtWncJ199e+Vc4UpNybdyBu+Nkd1MNoF4QxeQcLAFj
WnhakqCiQ+JIAD5AurhnrwChe0MV3kjA84GKH/r0SVqt1984mu/1DyS819XcfSSn
c00MsK1KbneVSCIV2XECaWAAaA7MHkwCQYDVR0TBAlwADAsBglghkgBhvhCAQ0E
HxYdTB3B1b1NTTCBHZW51cmF0ZWQgQ2VydG1maWNhdGUwHQYDVR00BBYEFNXXMIpSc
csBXUniW4A3UzZnCRsv3MB8GA1UdIwQYMBaAFDXy8rmj41Vz5+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
lvI99QyCxsDwoMnt5fj51v7aPmVeS60b0m+UlGre8B/Ze18JVj204K2Uh72rDCXE
5e6eFxDuM207sQDy20euBVELJtM2k0kL1R0doQQS1U3XQNgJw/t3ZIX5hPXWEQT
rjRQ064BEhb+PzzxPF8uwzZ9IrkUK9BJ/UUnqC6OdBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTKI80GvBvWX04fHyvIX5aslwhBoArXIS1AtNtrptPvoaP1zyIAeZ60Cvo/
Sfo+A2UhmTEJe0JaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIETgu4A1+CV64RJED
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWdyJBVhU0jXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfdwP+bEjDzUaaarJTS7lIFeLlW7eJ8MNAkMGicDkycL0
EPBU9X5QnHKLK0fYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+1YcX80VZ6
sHFPN1HluCny/riUcBy9iviVeodX8Iom0chSy05DK18bwZNjYtUP+CtYHNFU5BaD
I+1uU0AeJ+wjQYKT1WaeIGZ3VxuNITJul8y5qDTXXfX7vxM50oWxa6U5+AYuGUMg
/itPZmUmNrhjT7ghT6i1IQ0aBowXXKJB1Mmq/6BQXN2IhkD9ys2qrvM1hdi5nAf
egmdiG50LoLnBRqWbfr+DykpAhK4SaDi2F52Uxovw3Lhwi8dQP7lZQ==
-----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。

- a) 根据使用的协议，选择启用 **SSL 协议**或启用 **TLS 协议**。
- b) 选择**禁用覆盖**。
- c) 至少选择一个密码套件。
- d) 如果需要双向认证，请将 **启用客户机认证** 的值更改为 Yes。
- e) 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`。
- f) 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。

- a) 根据使用的协议，选择启用 **SSL 协议**或启用 **TLS 协议**。
- b) 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 118 页的『[配置 Connect:Direct 网桥](#)』

配置 Connect:Direct 网桥，以在 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 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 Managed File Transfer 传输过程中启动的 Connect:Direct 进程。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 `ConnectDirectProcessDefinitions.xml` 文件。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文件 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml`。
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 550 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 550 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 网桥代理才能够调用由该 <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。此文件的位置是使用绝对路径或相对于 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` 目录指定的。

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 118 页的『配置 Connect:Direct 网桥』](#)

配置 Connect:Direct 网桥, 以在 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 Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

相关参考

[第 492 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` 文件指定用户定义的 Connect:Direct 进程, 以在文件传输过程中启动。

[第 335 页的『fteCreateCDAgent \(创建 Connect:Direct 网桥代理\)』](#)

`fteCreateCDAgent` 命令用于创建 Managed File Transfer Agent 及其相关联的配置, 以与 Connect:Direct 网桥一起使用。

通过 MSCS 配置 MFT 代理

Managed File Transfer (MFT) 代理 Microsoft Cluster Service (MSCS) 设置受支持, 前提是平台受 MFT 支持, 并运行 Windows 的某个版本。

关于此任务

此任务描述了为实现 MFT 代理程序的故障转移而可以遵循的两种方案:

- 方案 1: 将代理程序配置为 MSCS 资源。
- 方案 2: 将代理队列管理器和代理配置为 MSCS 资源。

过程

方案 1: 将代理程序配置为 MSCS 资源

- 要将代理程序配置为 MSCS 资源, 请完成以下步骤:
 - a) 在集群中的每台机器上本地安装 Managed File Transfer。
请参阅[安装 Managed File Transfer](#)。
 - b) 在集群中的主机器上创建代理程序。
应该将代理配置为使用 CLIENT 传输连接到代理队列管理器。确保在队列管理器上为此代理创建所有对象。有关如何执行此操作的信息, 请参阅[设置代理程序](#)。
 - c) 通过将 Windows 服务工具中代理程序服务的 **启动类型** 字段设置为 **手动**, 修改代理程序以作为 Windows 服务运行, 并对其进行配置, 以使其不会在 Windows 重新启动时自动启动。
有关更多信息, 请参阅[将 MFT 代理作为 Windows 服务启动](#)。
 - d) 在辅助机器上重复方案 1 的步骤 [第 127 页的『2』](#) 和步骤 [第 127 页的『3』](#)。
这将确保集群中的另一台机器上存在日志, 属性等的文件结构。请注意, 无需像步骤 [第 127 页的『2』](#) 中那样创建队列管理器对象。
 - e) 在主机器上, 将代理程序添加为 MSCS 控制下的 "通用服务"。
要执行此操作:
 - a. 右键单击集群, 然后选择 **角色-> 添加资源-> "通用服务"**。
 - b. 从 Windows 服务列表中, 选择代理程序服务, 然后单击 **下一步** 以完成配置向导。
代理程序服务现在添加为 MSCS 资源。如果发生故障转移, 那么将在另一台机器上启动代理程序服务。

方案 2: 将代理队列管理器和代理配置为 MSCS 资源

- 要将代理队列管理器和代理配置为 MSCS 资源, 请完成以下步骤:
 - a) 配置代理队列管理器以作为 MSCS 资源运行。
有关如何执行此操作的信息, 请参阅[将队列管理器置于 MSCS 控制下](#)。
 - b) 在集群中的主机器上创建代理程序。
应该将代理配置为使用 BINDINGS 传输连接到代理队列管理器。确保在队列管理器上为此代理创建所有对象。有关如何执行此操作的信息, 请参阅[设置代理程序](#)。
 - c) 通过将 Windows 服务工具中代理程序服务的 **启动类型** 字段设置为 **手动**, 修改代理程序以作为 Windows 服务运行, 并对其进行配置, 以使其不会在 Windows 重新启动时自动启动。
有关更多信息, 请参阅[将 MFT 代理作为 Windows 服务启动](#)。
 - d) 确保代理队列管理器 (受 MSCS 控制) 正在辅助机器上运行。
在此机器上创建的代理将使用 BINDINGS 传输连接到队列管理器, 因此在创建代理时需要可用。
 - e) 在辅助机器上重复方案 2 的步骤 [第 127 页的『2』](#) 和步骤 [第 127 页的『3』](#)。
这将确保集群中的另一台机器上存在日志, 属性等的文件结构。请注意, 无需像步骤 [第 127 页的『2』](#) 中那样创建队列管理器对象。
 - f) 将代理程序添加为 MSCS 控制下的 "通用服务"。
要执行此操作:
 - a. 右键单击集群, 然后选择 **角色-> 添加资源-> "通用服务"**。
 - b. 从 Windows 服务列表中, 选择代理程序服务, 然后单击 **下一步** 以完成配置向导。
 - g) 修改代理程序服务的资源属性以在依赖关系列表中添加队列管理器资源。
这将确保在启动代理程序之前启动队列管理器资源。
 - h) 使队列管理器资源脱机, 然后使代理程序资源联机。验证队列管理器资源和代理程序是否都已启动。
如果发生故障转移, 那么将在辅助机器上启动代理服务和代理队列管理器。

管理 MFT

使用 Managed File Transfer 命令来管理 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。

启动 MFT 代理

必须先启动 Managed File Transfer 代理，才能将其用于文件传输。

关于此任务

您可以从命令行启动 Managed File Transfer Agent。在此情况下，当您从系统注销后，代理进程将停止。

ULW 在 UNIX, Linux, and Windows 上，您可以将代理配置为在您从系统注销后继续运行并可以继续接收文件传输。

z/OS 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。请注意，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 442 页的『fteStartAgent』](#)。
- **ULW**
要配置代理以使其在您从系统注销后继续运行：
 - **Windows** 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 128 页的『将 MFT 代理作为 Windows 服务启动』](#)。
 - **Linux** **UNIX** 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 130 页的『在 UNIX 和 Linux 系统启动时启动 MFT 代理』](#)。
- **z/OS**
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 131 页的『在 z/OS 上启动 MFT 代理』](#)。

Windows 将 MFT 代理作为 Windows 服务启动

您可以将代理作为 Windows 服务启动，以便在注销 Windows 时，代理继续运行并可接收文件传输。

关于此任务

在 Windows 上，当从命令行启动代理时，代理进程使用您用于登录到 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，可以将代理配置为作为 Windows 服务运行。通过作为 Windows 服务运行，还可以将代理配置为在 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。您必须是在某个受支持的 Windows 版本上运行 Managed File Transfer，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 [IBM 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 Managed File Transfer 代理程序，请使用 **fteCreateAgent**、**fteCreateCDAgent** 或 **fteCreateBridgeAgent** 命令。指定 **-s** 参数以将代理作为 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。Windows 服务使用具有关联密码 ftepassword 的用户名 fteuser 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 `mqmftAgentAGENTQMGR`，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 271 页的『用于将 MFT 代理或记录器作为 Windows 服务运行的指南』。

有关更多信息，请参阅第 323 页的『fteCreateAgent (创建 MFT 代理)』，第 335 页的『fteCreateCDAgent (创建 Connect:Direct 网桥代理)』或第 327 页的『fteCreateBridgeAgent (创建并配置 MFT 协议网桥代理)』。

2. 如果遵循上一步创建代理程序，请运行由 **fteCreateAgent**、**fteCreateCDAgent** 或 **fteCreateBridgeAgent** 命令生成的 MQSC 命令。这些命令将创建代理所需的 IBM 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. 如果未按照先前步骤创建代理，而是要将现有代理配置为作为 Windows 服务运行，请首先停止代理（如果其正在运行），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** 命令将代理程序配置为作为 Windows 服务运行：

```
fteModifyAgent -agentName AGENT1 -s -su fteuser -sp ftepassword
```

有关更多信息，请参阅第 410 页的『fteModify 代理 (将 MFT 代理作为 Windows 服务运行)』。

4. 使用 **fteStartAgent** 命令启动代理。或者，可以使用 Windows 服务工具（从 Windows 桌面开始菜单选择控制面板，然后选择“管理工具”即可获取）启动该服务。

```
fteStartAgent AGENT1
```

即使注销 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 重新启动时也重新启动，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。这是由于 Windows 服务的限制所导致。

下一步做什么

如果在启动 Windows 服务时发生问题，请参阅第 271 页的『[用于将 MFT 代理或记录器作为 Windows 服务运行的指南](#)』。本主题还描述 Windows 服务日志文件的位置。

Linux

UNIX

在 UNIX 和 Linux 系统启动时启动 MFT 代理

Managed File Transfer Agent 可以配置为在 UNIX 上系统启动时启动。当注销 UNIX 系统时，代理会继续运行并可接收文件传输。

使用下列其中一个 Managed File Transfer 命令 (**fteCreateAgent**, **fteCreateCDAgent**, 或 **fteCreateBridgeAgent**) 创建和配置代理程序后，可以将其配置为在 UNIX 机器上使用仅执行以下操作的脚本文件在重新引导期间自动启动：

```
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name
```

其中 *mq_install_root* 是必需 Managed File Transfer 安装的根目录，缺省值为: /opt/mqm, *agent_name* 是要启动的 Managed File Transfer Agent 的名称。此脚本文件的用法根据特定 UNIX 操作系统而异。

Solaris

Solaris

在 Solaris 上，请按照以下步骤执行操作：

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

HP-UX

在 HP-UX 上，请按照以下步骤执行操作：

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
```

Linux

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，请考虑以下步骤：

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环境的方法。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
```

z/OS 在 z/OS 上启动 MFT 代理

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，不受用户注销影响。

注: **V 9.0.2** **V 9.0.0.1** 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 9.0.2 和 IBM MQ 9.0.0 Fix Pack 1 开始，代理程序属性 **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 460 页的『MFT agent.properties 文件』中的 **adminGroup** 属性。

作为 Java 应用程序，代理是您可以使用 BFGAGSTP 成员从某个针对代理生成的 Managed File Transfer 命令 PDSE 库数据集，通过 JCL 来运行的 UNIX 系统服务应用程序。有关如何创建 MFT 命令 PDSE 库数据集并针对所需代理定制该数据集的更多信息，请参阅第 53 页的『创建 MFT 代理或记录器命令数据集』。

仅启用代理与 IBM MQ Advanced for z/OS, Value Unit Edition 的远程连接

V 9.0.3 **MQ Adv. VUE**

在 IBM MQ Advanced for z/OS VUE 的产品标识 (PID) 下运行的 z/OS 上的 MFT 代理程序可以使用客户机连接来连接到 z/OS 上的队列管理器。

当代理程序启动时，它会将 BFGPR0137I 消息写入其事件日志 (output0.log)，显示其运行所使用的 PID。此消息的一个示例是：

```
BFGPR0137I: 已针对产品 "MQ z/OS MFT" , 产品标识 "5655-MF9" 启动产品使用情况数据记录。
```

有关 IBM MQ 产品，其关联 PID 值和导出分类的详细信息，请参阅 [IBM MQ 产品标识和导出信息](#)。

在任何其他 PID 下运行的 z/OS 上的 MFT 代理只能使用绑定连接来连接到本地队列管理器。

z/OS 上的 MFT 代理只能连接到也在 z/OS 上运行的队列管理器，而与 MFT PID 无关。

如果 IBM MQ Advanced for z/OS VUE 代理程序尝试连接到未在 z/OS 上运行的队列管理器，那么将发出以下 **BFGMQ1044E** 消息，并结束代理程序启动：

```
BFGMQ1044E: z/OS 上的代理客户机连接必须与 z/OS 上的队列管理器连接
```

相关参考

第 176 页的『在 z/OS 上停止 MFT 代理』

如果要从 JCL 将 Managed File Transfer Agent on z/OS 作为启动式任务运行，那么代理程序将接受 z/OS 操作程序命令 **MODIFY** 和 **STOP** 以及 **fteStopAgent** 命令。

第 460 页的『MFT agent.properties 文件』

每个 Managed File Transfer Agent 都有自己的属性文件 **agent.properties**，该文件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。

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还是多个文件。

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 "创建新的受管文件传输" 向导来启动新的文件传输，请执行以下步骤：

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. 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 键。在 Linux 上，按 Ctrl+F1 或 Shift+F1。

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 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 **IBM 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="overwrite">
    <file>c:\targetfiles\destination3.txt</file>
  </destination>
</item>

```

```
<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 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。

V9.0.0.9

如果调度的传输每天同时运行，请使用代理属性文件中的 `adjustScheduleTimeForDaylightSaving` 属性来调整时钟更改时调度发生的时间。请参阅 `MFT agent.properties` 文件以获取更多信息。

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 "创建新的受管文件传输" 向导来创建新的调度文件传输，请执行以下步骤：

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 键。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](#)。

处理暂挂文件传输

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的名称。
 - 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 > Managed File Transfer**，然后从**时区：**列表中选择备用时区。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 141 页的『监视 MFT 资源』](#)。

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 "创建新的受管文件传输" 向导来创建调度文件传输，请执行以下步骤：

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 键。在 Linux 上，按 **Ctrl+F1** 或 **Shift+F1**。

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 138 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。您可以使用 **除去已完成的传输**  在打开 IBM MQ Explorer 时的任何时刻清除已完成的传输。


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 > Managed File Transfer**，然后从**时区：**列表中选择备用时区。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**。

下一步做什么

此外，可以开发应用程序以用于监视定制文件传输。可通过创建对相应 Managed File Transfer 管理主题的预订（以编程方式或以管理方式）来实现这一点，随后监视应用程序可接收有关该主题的 Managed File Transfer 文件传输活动发布。有关预订主题和发布消息格式的更多信息，请参阅第 632 页的『[文件传输进度消息示例](#)』。

相关任务

第 138 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 IBM WebSphere MQ 7.5 或更高版本中，您需要一个能够运行 IBM MQ Explorer 的系统。需要安装 IBM MQ Explorer 组件才能连接到远程协调队列管理器。

第 139 页的『[查看传输日志中文件传输的状态](#)』

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。这些可以是通过命令行或 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 IBM WebSphere MQ 7.5 或更高版本中，您需要一个能够运行 IBM MQ Explorer 的系统。需要安装 IBM MQ Explorer 组件才能连接到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 45 页的『使用通道认证以客户机方式连接到队列管理器』和第 292 页的『管理特定于 MFT 资源的权限』。

要在不运行 Windows 或 Linux 的系统上监视代理之间的队列管理器和文件传输，请使用以下步骤将 IBM MQ Explorer 配置为连接到远程系统：

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 Managed File Transfer 网络的传输活动。

相关任务

第 137 页的『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 139 页的『查看传输日志中文件传输的状态』

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。

查看传输日志中文件传输的状态

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）**：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 > Managed File Transfer**，然后从**时区**：列表中选择备用时区。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**：用户使用 **fteCreateTransfer** 的 **-jn** 参数指定的标识，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 8.0.0 Fix Pack 1 中针对 APAR IC99545 更改了传输日志的内部格式。因此, 如果 IBM MQ Explorer 升级到 8.0.0.1 或更高版本, 然后复原到 8.0.0.0, 那么对于 IBM MQ Explorer 处于 8.0.0.1 时发生的传输, 不会显示任何审计 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, 请通过单击加号 (+) 展开您感兴趣的传输。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**操作下的弹出菜单获取传输的元数据和完整的审计 XML。

相关概念

第 205 页的『[恢复中文件传输的超时选项](#)』

您可以设置时间量 (以秒为单位), 在此期间, 源代理会一直尝试恢复停止的文件传输。如果代理达到重试时间间隔的超时值时传输不成功, 则传输失败。

相关任务

第 137 页的『[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 140 页的『[配置传输日志](#)』

您可以在 IBM MQ Explorer 的 **传输日志** 中配置所显示的信息以及信息的显示方式。

配置传输日志

您可以在 IBM MQ Explorer 的 **传输日志** 中配置所显示的信息以及信息的显示方式。


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。该字段中, 您可以使用任何有效的 Java 正则表达式。有关更多信息, 请参阅 [第 549 页的『MFT 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 137 页的『[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 139 页的『查看传输日志中文件传输的状态』

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。这些可以是通过命令行或 IBM MQ Explorer 启动的传输。您还可以定制**传输日志**中显示的内容。

监视 MFT 资源

您可以监视 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 Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

关于此任务

Managed File Transfer 资源监视使用以下术语：

monitor

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 (resource)

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 (condition)

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 149 页的『使用变量替换定制 MFT 资源监视器任务』。

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。

协议网桥代理或 Connect:Direct 网桥代理上不支持资源监视。

MFT 资源监视概念

这里概述了 Managed File Transfer 资源监视功能的主要概念。

监视器

资源监视器与 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。监视器名称的长度必须至少为一个字符，并且不得包含星号 (*)，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。

重叠的资源监视器可能导致：

- 源位置/项的可能争用。
- 针对相同源项的可能重复传输请求。
- 由于源项的冲突，导致传输发生意外错误或失败。

如果多个监视器扫描同一位置并可以在同一项上触发，那么可能导致两个不同的监视器提交同一项的受管传输请求的问题。

资源监视器在每个轮询时间间隔之后查看资源的内容。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

Managed File Transfer 中的资源监视器可以轮询以下两种类型的资源的内容：

目录或嵌套目录结构

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。Managed File Transfer 资源监视器可以检测到触发器文件，并且会启动将这些文件从源目录传输到另一个 Managed File Transfer Agent 的操作。

缺省情况下，将监视指定目录。要检查子目录，请在 `fteCreateTransfer` 命令中设置递归级别。

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信息，请参阅第 149 页的『使用变量替换定制 MFT 资源监视器任务』。

注：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。

有关创建资源监视器以监视目录的示例，请参阅第 147 页的『监视目录和使用变量替换』。

IBM MQ 队列

监视队列的一个示例是，外部应用程序可能正在生成消息并将它们放在具有相同组标识的已知队列上。应用程序将这些消息放置在队列上后，表明该组是完整的。Managed File Transfer 资源监视器可以检测到完整的消息组，并启动将消息组从源队列传输到文件的操作。要获取有关创建用于监视队列的资源监视器的示例，请参阅第 149 页的『示例：配置 MFT 资源』。

注：您可以指定每个队列仅一个监视器。如果指定多个监视器来轮询 IBM 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 545 页的『对 MFT 使用通配符』中所述
- 正则表达式匹配

也可以通过使用通配符或标识从未匹配的文件名的 Java 正则表达式，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

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为 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 **-mt** 参数的值传递。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。

备份和复原资源监视器

您可以备份已定义的资源监视器，以便将来复用。有多种选项可供您使用，如下所示：

- 使用带有 **-ox** 参数的 **fteCreateMonitor** 命令将资源监视器配置导出到 XML 文件，并使用 **-ix** 参数通过从 XML 文件导入资源监视器配置来复原资源监视器。
- 将 **fteListMonitors** 命令与 **-ox** 配合使用，以将单个资源监视器的定义导出到 XML 文件。
- **V 9.0.5** 从 IBM MQ 9.0.5 开始，将 **fteListMonitors** 命令与 **-od** 配合使用，以将多个资源监视器定义导出到指定的目录。每个资源监视器定义均保存到一个独立的 XML 文件。您还可以使用 **-od** 选项将单个资源监视器定义导出到指定的目录。

有关更多信息，请参阅第 160 页的『[备份和复原 MFT 资源监视器](#)』。

资源监视器日志记录

V 9.0.3

从 IBM MQ 9.0.3 开始，Managed File Transfer 包含资源监视器日志记录。有关更多信息，请参阅第 158 页的『[记录 MFT 资源监视器](#)』。

相关概念

第 149 页的『[使用变量替换定制 MFT 资源监视器任务](#)』

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 144 页的『[配置 MFT 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 149 页的『[示例：配置 MFT 资源](#)』

通过将 **-mq** 参数与 **fteCreateMonitor** 命令配合使用，可以将 IBM MQ 队列指定为要由资源监视器监视的资源。

第 154 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 345 页的『[fteCreateMonitor: 创建 MFT 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 Managed File Transfer 监视资源（例如，某个目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

第 397 页的『[fteListMonitors: 列出 MFT 资源监视器](#)』

通过命令行，使用 **fteListMonitors** 命令可列出 Managed File Transfer 网络中的所有现有资源监视器。

第 389 页的『[fteDeleteMonitor: 删除 MFT 资源监视器](#)』

使用 **fteDeleteMonitor** 命令可通过命令行停止和删除现有 Managed File Transfer 资源监视器。针对资源监视代理发出此命令。

配置 MFT 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，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 303 页的『[commandPath MFT 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 144 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 146 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 663 页的『[文件传输请求消息格式](#)』。

示例

以下示例显示了另存为 `cleanuptask.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>
    </transferSet>
  </managedCall>
</request>
```



```

        <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>
</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 Managed File Transfer Agent。

<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。

`successrc` 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 `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。

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`

文档引用的所有任务（Ant 脚本或 JCL 作业）都必须位于监视代理的命令路径中。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 149 页的『[使用变量替换定制 MFT 资源监视器任务](#)』

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 663 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（通常是由于用户发出文件传输命令或使用 IBM MQ Explorer）启动。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 303 页的『[commandPath MFT 属性](#)』

使用 `commandPath` 属性以指定 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 保存到文件 /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 149 页的『使用变量替换定制 MFT 资源监视器任务』

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 144 页的『配置 MFT 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 345 页的『fteCreateMonitor: 创建 MFT 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 Managed File Transfer 监视资源（例如，某个目录的内容），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 MFT 资源

通过将 **-mq** 参数与 **fteCreateMonitor** 命令配合使用，可以将 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。

使用变量替换定制 MFT 资源监视器任务

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。



注意：变量名不区分大小写。

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。如果使用了 **noMatch** 条件，并且任务定义中存在替换变量名称，那么不会调用该任务，并且监视器会产生返回码 **110** 和错误消息 **BFGDM0060E**。

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。变量名前面必须加美元符号 (\$) 字符并用花括号 {} 括起来。

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 566 页的『MFT 从源队列上的消息读取的 IBM MQ 消息属性』和第 154 页的『监视队列和使用变量替换』。

如果变量未定义为消息属性，那么监视器将报告 BFGDM0060E 错误并返回返回码 110 (Monitor 任务变量替换失败)。除此之外，代理程序还会将以下错误消息写入其事件日志 (outputN.log):

```
BFGDM0113W: Trigger failure for <monitor name> for reason BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

如果为监视器启用了中度或详细资源监视器日志记录，那么监视器会将以下消息写入代理程序的资源监视器事件日志 (resmoneventN.log):

```
BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

有关资源监视器日志记录的更多信息，请参阅第 158 页的『记录 MFT 资源监视器』。

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。

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。

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。
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。

如果受监视资源是触发器文件

下表显示了当资源监视器使用触发器文件的内容来确定需要传输的文件时可以替换的变量名称集。

变量	描述
contentSource	源文件的完整路径名。
contentDestination	目标文件的完整路径名。

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 UNIX 平台上的正斜杠字符 (/) 或 Windows 平台上的反斜杠字符 (\), 但分隔符可以是变量值中可能出现的任何有效字符。

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。

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 Agent 名称都采用大写形式。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 151 页的『[示例: 资源监视器定义的变量替换](#)』

使用 XML 和 IBM MQ Explorer 对资源监视器定义进行变量替换的示例。

相关参考

第 274 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: 资源监视器定义的变量替换

使用 XML 和 IBM 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

假定匹配触发器文件的文件路径为 `c:\MONITOR\REPORTS\Paris\Report2009.doc` (在 Windows 上) 和 `/MONITOR/REPORTS/Paris/Report2009.doc` (在 UNIX 平台上), 那么将替换变量, 如下表中所示。

变量规范	变量替换后
<code>\${FilePath}</code>	Windows : <code>c:\MONITOR\REPORTS\Paris\Report2009.doc</code> UNIX : <code>/MONITOR/REPORTS/Paris/Report2009.doc</code>

变量规范	变量替换后
<code>\${FilePath{token=1}{separator=.}}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009 UNIX :/MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=2}{separator=.}}</code>	Windows :doc UNIX :doc
<code>\${FilePath{token=3}}</code>	Windows : 报告 UNIX : 巴黎

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。表中的示例使用相同的变量值 `c:\MONITOR\REPORTS\Paris\Report2009.doc` (在 Windows 上) 和 `/MONITOR/REPORTS/Paris/Report2009.doc` (在 UNIX 平台上)。

变量规范	变量替换后
<code>\${FilePath}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009.doc UNIX :/MONITOR/REPORTS/Paris/Report2009.doc
<code>\${FilePath{token=-2}{separator=.}}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009 UNIX :/MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=-2}{separator=\}}</code>	Windows : 巴黎 UNIX : 巴黎
<code>\${FilePath{token=-4}}</code>	Windows : 监控 UNIX : 监控

用于替换的变量仅可用于以下正触发条件和 `noSizeChange` 选项，这是正触发条件规则的例外情况：

- `match`
- `fileSize`
- `noSize` 更改

如果使用了 `noMatch` 条件，并且任务定义中存在替换变量名称，那么不会调用该任务，并且监视器会产生返回码 110 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 IBM MQ Explorer 的示例

在通过 IBM 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 149 页的『使用变量替换定制 MFT 资源监视器任务』

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 274 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，将消息放入源队列时，请勿将其放入 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 Managed File Transfer Agent 将源队列上的每条消息视为单独的组。

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 149 页的『示例：配置 MFT 资源』](#)

通过将 `-mq` 参数与 `fteCreateMonitor` 命令配合使用，可以将 IBM MQ 队列指定为要由资源监视器监视的资源。

相关参考

[第 460 页的『MFT agent.properties 文件』](#)

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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。

使用触发器文件

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 345 页的『fteCreateMonitor: 创建 MFT 资源监视器』](#)。

使用触发器内容文件时，每行的缺省格式为：

- 单个源文件路径，或者

- 源文件路径和目标文件路径，以逗号分隔

其中空格字符作为文件路径的一部分进行处理。可以通过在 **fteCreateMonitor** 命令上指定 **-tcr** 和 **-tcc** 参数来更改缺省行格式。有关更多信息，请参阅第 157 页的『高级选项』。

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`。

或者，如果将 **fteCreateTransfer** 命令中的 **-dd /file/destdir** 参数更改为 **-df \$**`{contentDestination}`，并且将触发器文件的内容格式更改为 `source file path,destination file path`，那么可以为同一个目标代理定义不同的目标路径。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/**`{contentDestination{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
```

V 9.0.3 记录 MFT 资源监视器

从 IBM MQ 9.0.3 开始，Managed File Transfer 提供一种改进方法以用于获取有关资源监视器的诊断信息。

关于此任务

从 IBM MQ 9.0.3 开始，您可以对资源监视器使用日志记录。您可以使用 **fteSetAgentLogLevel** 命令或 `agent.properties` 文件来控制资源监视器日志记录。

请注意，现有的跟踪点仍用于捕获信息。

资源监视器日志将写入名为 `resmoneventN.log` 的文件，其中 *N* 代表数字；例如，`resmonevent0.log`。



注意: 代理的所有资源监视器将写入同一个日志文件中。

有关 `resmoneventN.log` 文件的输出示例, 请参阅第 263 页的『当 MFT 目录资源监视器未触发文件时该怎么做』。

下表列出了资源监视器写入日志文件的事件的类型。第三列描述捕获每个事件所需的日志级别, 其中最低级别为 INFO, 最高级别为 VERBOSE。

请注意, 设置较高的日志级别也将写入较低级别的事件。例如, 将日志级别设置为 MODERATE 也将写入 INFO 级别的事件, 但不会写入 VERBOSE 级别的事件。

号码	事件	日志级别	描述
1	监视器创建	信息	已创建资源监视器。
2	监视器删除	信息	已删除资源监视器。
3	监视器停止	信息	已停止资源监视器。
4	监视器启动	信息	已启动资源监视器。
5	监视器启动轮询	信息	资源监视器已启动新的轮询周期。
6	监视器结束轮询	信息	已结束资源监视器轮询周期。
7	模式匹配	VERBOSE	已发现触发器监视器目录中的文件或所指定模式匹配的队列中的消息。
8	模式不匹配	VERBOSE	已发现触发器监视器目录中不匹配的文件或与所指定模式不匹配的队列中的消息。
9	传输提交	信息	资源监视器已启动传输。
10	目录太深	VERBOSE	资源监视器所监视的目录中要轮询的子目录数比资源监视器配置中指定的数量要多。
11	文件已锁定	MODERATE	资源监视器所监视的触发器文件被另一个进程锁定。
12	文件大小较小	MODERATE	触发器文件小于资源监视器配置中指定的大小。
13	文件大小不稳定	MODERATE	触发器文件的更改频率高于资源监视器配置的预期。
14	轮询次数过多	MODERATE	资源监视器轮询不稳定的触发器文件次数过多。
15	已匹配项	信息	在资源监视器轮询的目录中找到的触发器文件总数。
16	传输中的项数	信息	传输请求中的总项数。
17	FDC 或异常	MODERATE	资源监视器已生成异常。
18	传输请求	信息	资源监视器已提交传输请求。
19 位	监视器启动失败	MODERATE	资源监视器启动失败。

过程

- 要使用 `fteSetAgentLogLevel` 来打开和关闭资源监视器日志记录, 请参阅 `fteSetAgentLogLevel` 以获取 `logMonitor` 参数的描述以及有关如何使用各种选项的示例。
- 要使用 `agent.properties` 文件来控制资源监视器日志记录, 请参阅第 460 页的『MFT `agent.properties` 文件』以获取允许您执行以下日志记录活动的附加属性的描述:

- 开启或关闭日志记录
- 限制每个日志文件的大小
- 限制资源监视器可以生成的日志数

示例

以下样本消息为队列管理器 MFTDEMO 上的代理 HA2 设置 verbose 级别日志记录：

```
<?xml version="1.0"?>
<log:log version="6.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:log="https://www.ibm.com/log">
  <log:originator>
    <log:request>
      <log:hostname>192.168.7.1</log:hostname>
      <log:userID>johndoe</log:userID>
    </log:request>
  </log:originator>
  <log:endpoint agent="HA2" QMgr="MFTDEMO"/>
  <log:logMonitor>MON1="verbose"</log:logMonitor>
</log:log>
```

相关参考

[fteSetAgentLogLevel 命令](#)

使用 **fteSetAgentLogLevel** 命令对 Managed File Transfer 协议网桥代理与 FTP/SFTP/FTPS 文件服务器之间的文件传输启用或禁用诊断信息日志记录。

[第 460 页的『MFT agent.properties 文件』](#)

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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。

备份和复原 MFT 资源监视器

对于您希望将来可用的资源监视器，可以通过将其定义导出到 XML 文件来备份这些监视器，以后可通过导入此 XML 文件来从备份创建新的资源监视器。

关于此任务

您可能需要备份先前已定义的资源监视器以便将来可以复用其定义，例如，在其他基础结构中重新创建资源监视器时或在由于队列管理器问题而需要重新创建资源监视器时进行复用。

您可以通过使用 **fteCreateMonitor** 命令或者带有 **-ox** 参数的 **fteListMonitors** 命令来备份单个资源管理器定义。在上述两种情况下，可通过将资源管理器定义导出到 XML 文件来对其进行备份。随后，您可以使用 **fteCreateMonitor** 命令的 **-ix** 参数，通过从 XML 文件导入定义来创建新的资源管理器。

如果使用 **-ox** 参数，每次只能备份一个资源监视器定义。

V 9.0.5 从 IBM MQ 9.0.5 开始，已将 **-od** 参数添加到 **fteListMonitors** 命令中。通过指定此参数，即可通过将多个资源监视器定义批量导出到指定的目录来同时备份这些资源监视器。每个资源监视器定义都保存到一个单独的 XML 文件中，其名称格式为 `agent name.monitor name.xml`。

V 9.0.5 如果您需要备份大量资源监视器，那么 **-od** 参数特别有用，因为您只需运行一次 **fteListMonitors -od** 命令即可，而无需针对每个资源定义单独运行 **fteListMonitors -ox** 命令或者使用独立脚本来针对每个资源监视器运行 **fteListMonitors -ox** 命令。

过程

- 要通过将一个资源监视器定义导出到 XML 文件来对其进行备份，请使用以下任一命令：
 - 带有 **-ox** 参数的 **fteCreateMonitor** 命令。
 - 带有 **-ox** 参数的 **fteListMonitors** 命令。

使用 **-ox** 参数时，还必须指定 **-ma** 和 **-mn** 参数，如以下示例中所示：

```
ftelistMonitors -ma AGENT1 -mn MONITOR1 -ox filename1.xml
```

V 9.0.5

要通过将多个资源监视器定义导出到指定目录中的 XML 文件来备份这些定义，请使用带有 **-od** 参数的 **ftelistMonitors** 命令，如以下示例中所示：

```
ftelistMonitors -od /usr/mft/resmonbackup
```

批量备份资源监视器时，必须指定有效的目标目录。如果未指定目标路径，那么会导致出现一条错误消息，如以下示例中所示：

BFGCL0762E: Output directory not specified. 指定有效路径以重新运行该命令。

-od 参数不能与 **-ox** 参数组合使用，否则会显示以下错误消息：

BFGCL0761E: It is not valid to specify both the '-od' and '-ox' parameters together.

您可以定义要包含在备份中的一组特定的资源监视器。例如，通过使用 **-ma** 参数指定某个代理的名称，可以备份此代理的所有资源监视器，如以下示例中所示：

```
ftelistMonitors -ma AGENT1 -od /usr/mft/resmonbackup
```

在定义用于匹配代理名称和/或监视器名称的模式时，您还可以通过包含星号字符 (*) 来使用通配符匹配。以下示例中，针对名称与指定模式匹配的代理，将备份其中名称与指定模式匹配的所有资源监视器：

```
ftelistMonitors -ma AGENT* -mn MON* -od /usr/mft/resmonbackup
```

运行该命令时，将显示以下进度报告消息：

```
A total of number matching resource monitor definitions found.  
index of number resource monitor definitions saved to file system.
```

如果使用 **verbose** 选项，那么仍会显示正在运行的总数，而不显示

```
index of number resource monitor definitions saved to file system
```

，该命令显示正在保存的监视器定义的名称，例如：

```
BFGCL0762I: Definition of monitor 'FILEMON' of agent 'XFERAGENT' saved as  
FILEMON.XFERAGENT.XML to file system.
```

V 9.0.5

要通过将特定代理程序导出到指定目录中的 XML 文件来备份该代理程序的一个资源监视器，请使用带有 **-od** 参数的 **ftelistMonitors** 命令：

```
ftelistMonitors -ma AGENT1 -mn MONITOR1 -od /usr/mft/resmonbackup
```

使用 **-od** 参数备份单个资源监视器与使用 **-ox** 参数相似，但输出文件名采用 *agent name.monitor name.xml* 格式。

- 要从备份复原资源监视器定义，请使用带有 **-ix** 参数的 **fteCreateMonitor** 命令，如以下示例中所示：

```
fteCreateMonitor -ix file name
```

有关如何使用 **-od** 参数的更多示例，请参阅第 397 页的『[ftelistMonitors: 列出 MFT 资源监视器](#)』。

相关参考

[第 345 页的『fteCreateMonitor: 创建 MFT 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 Managed File Transfer 监视资源（例如，某个目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

[第 397 页的『ftelistMonitors: 列出 MFT 资源监视器』](#)

通过命令行，使用 **ftelistMonitors** 命令可列出 Managed File Transfer 网络中的所有现有资源监视器。

使用文件传输模板

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。可以从命令行中使用 **fteCreateTemplate** 命令创建传输模板，也可以在 IBM MQ Explorer 中使用**新建受管文件传输模板**向导来创建传输模板，或者在创建文件传输时选中**将传输设置保存为模板**复选框来保存该模板。传输模板窗口显示在您的 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，如果要提交在命令行上创建的传输模板，请单击 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的名称。
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 > Managed File Transfer**，然后从**时区：**列表中选择备用时区。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 163 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 351 页的『[fteCreateTemplate: 创建新的文件传输模板](#)』

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 401 页的『[fteListTemplates](#)』

fteListTemplates 命令可用于列出协调队列管理器上的可用 Managed File Transfer 传输模板。

第 392 页的『[fteDeleteTemplates](#)』

使用 **fteDeleteTemplates** 命令可从协调队列管理器中删除现有的 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 "为受管文件传输创建新模板" 向导来创建文件传输模板，请执行以下步骤：

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 键。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 162 页的『[使用文件传输模板](#)』

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。可以从命令行中使用 **fteCreateTemplate** 命令创建传输模板，也可以在 IBM MQ Explorer 中使用**新建受管文件传输模板**向导来创建传输模板，或者在创建文件传输时选中**将传输设置保存为模板**复选框来保存该模板。**传输模板**窗口显示在您的 Managed File Transfer 网络中已创建的所有传输模板。

相关参考

第 351 页的『[fteCreateTemplate: 创建新的文件传输模板](#)』

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 401 页的『[fteListTemplates](#)』

fteListTemplates 命令可用于列出协调队列管理器上的可用 Managed File Transfer 传输模板。

第 392 页的『[fteDeleteTemplates](#)』

使用 **fteDeleteTemplates** 命令可从协调队列管理器中删除现有的 Managed File Transfer 模板。

将数据从文件传输到消息

您可以使用 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”传输，传输的源和目标代理都必须为 IBM WebSphere MQ 7.5 或更高版本，或者 IBM WebSphere MQ File Transfer Edition 7.0.3 或更高版本。有关“消息到文件”传输的信息，请参阅第 171 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。
- 从单个文件到多条消息，通过以 Java 正则表达式定义符来分割文本文件。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 Managed File Transfer 属性。有关信息，请参阅第 268 页的『用于设置与消息大小关联的 MQ 属性和 MFT 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 265 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？』

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。

示例：将单个文件传输到单条消息

您可以通过将 **-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。

示例：按长度将单个文件分割为多条消息

您可以使用 **fteCreateTransfer** 命令的 **-qs** 参数将文件拆分为多条 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。

示例：使用正则表达式定界符将文本文件分割为多条消息

通过在每次匹配给定的 Java 正则表达式时分割文件来将单个文本文件传输至多条消息。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：

```
Jenny Jones
```

- 第二条消息：

```
John Smith
```

- 第三条消息：

```
Jane Brown
```

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

通过在每次匹配给定的 Java 正则表达式时分割文件来将单个文本文件传输至多条消息，并在产生的消息中包含正则表达式匹配。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。该正则表达式作为 Java 正则表达式进行求值。有关更多信息，请参阅 [第 549 页的『MFT 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
```

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
```

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。

示例：为文件到消息传输添加用户定义的消息属性

如果要将 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 168 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:namespaceSchemaLocation="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 168 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 730 页的『[fte:filecopy Ant 任务](#)』

fte:filecopy 任务会在 Managed File Transfer 代理之间复制文件。该文件不会从源代理删除。

第 733 页的『[fte:filemove Ant 任务](#)』

fte:filemove 任务会在 Managed File Transfer 代理之间移动文件。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 565 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.WMQFTEResultCode = 40
usr.WMQFTESupplement = BFGTR0036I: The transfer failed to complete successfully.
```

将数据从消息传输到文件

通过使用 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（在 z/OS 上）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 Managed File Transfer 的消息到文件传输功能，将这些消息传输到 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 163 页的『将数据从文件传输到消息』。



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 Managed File Transfer 属性。要了解更多信息，请参阅第 268 页的『用于设置与消息大小关联的 MQ 属性和 MFT 属性的指南』。

V 9.0.1 从 IBM MQ 9.0.1 开始，在消息到文件传输中，源代理在 IBM MQ 9.0.1 之前的版本中浏览来自源队列的消息（与破坏性 GET 不同）。浏览完所有消息（如果使用了消息分组，则成组浏览）且数据写入目标文件后，消息将从源队列中移除。这使得传输失败或取消时消息能够保留在源队列中。由于此更改，必须随 GET 权限一起提供 BROWSE 权限才能运行消息到文件传输。

V 9.0.0.2 从 IBM MQ 9.0.0 Fix Pack 2 开始，Managed File Transfer 已更新为复原 APAR IT18213 先前除去的传输请求 XML 有效内容中传输标识和 groupId 属性值的比较检查。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）。

配置代理以执行“消息到文件”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。

示例：从队列传输到单个文件

通过将 `-sq` 参数与 `fteCreateTransfer` 命令配合使用，可以将 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  
-da AGENT_VENUS -df /out/three_to_one.txt  
-sq START_QUEUE
```

队列 `START_QUEUE` 上消息中的数据会写入运行 `AGENT_VENUS` 的系统上的文件 `/out/three_to_one.txt` 中。

示例：将一组消息从队列传输到单个文件

通过将 `-sq` 和 `-sqgi` 参数与 `fteCreateTransfer` 命令配合使用，可以将 IBM MQ 队列上的单个完整组指定为文件传输的源。

关于此任务

在该示例中，队列 `START_QUEUE` 上有十条消息。该队列必须位于源代理的队列管理器 `QM_NEPTUNE` 中。前三条消息属于 IBM MQ 组标识为 `41424b3ef3a22020202020202020202020202020202020201111` 的组；该组不是一个完整组。接下来的五条消息属于 IBM MQ 组标识为 `41424b3ef3a2202020202020202020202020202020202020222` 的组；该组是一个完整组。其余的两条消息属于 IBM MQ 组标识为 `41424b3ef3a2203333` 的组；该组是一个完整组。

过程

输入以下命令：

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS  
-df /out/group.txt -sqgi -sq START_QUEUE
```

属于队列 `START_QUEUE` 上第一个完整组 (具有 IBM MQ 组标识 `41424b3ef3a2202020202020202020202020202020202020222` 的组) 的消息中的数据将写入正在运行 `AGENT_VENUS` 的系统上的文件 `/out/group.txt`。

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。

关于此任务

在该示例中，队列 `START_QUEUE` 上存在四条消息。该队列位于源代理的队列管理器 `QM_NEPTUNE` 中。要插入到每条消息的数据前的文本定界符可表示为 Java 字符串，例如：`\n\u002D\u002D\u002D\n`。

过程

输入以下命令：

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/output.txt
-t text -sqdt "\n\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。

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`。

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，将消息放入源队列时，请勿将其放入 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 Managed File Transfer Agent 将源队列上的每条消息视为单独的组。

示例：无法使用 IBM MQ 消息属性实现“消息到文件”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。

关于此任务

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。

某用户正在创建消息，并将消息放置到队列 INPUT_QUEUE 上。源代理正在使用来自队列 INPUT_QUEUE 中的消息，并将传输数据发送到目标代理。目标代理正在将该数据写入文件 /home/user/output.file 中。

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。

过程

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：

```
usr.UserReturnCode=1  
usr.UserSupplement="Cancelling transfer - sent wrong data."
```

2. 源代理读取 IBM MQ 消息属性，并停止处理来自此队列的消息。目标代理删除已写入到目标目录的所有文件数据。
3. 源代理向协调队列管理器发送传输日志消息，报告传输失败。

此消息包含以下信息：

```
<?xml version="1.0" encoding="UTF-8"?>  
<transaction version="1.00"  
  ID="414d5120514d31202020202020202020202020207e970d4920008702" agentRole="sourceAgent"  
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"  
  xmlns="">  
  <action time="2008-11-02T21:28:09.593Z">progress</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>reportserver.com</hostName>  
    <userID>USER1</userID>  
    <mqmdUserID>USER1 </mqmdUserID>
```

```

</originator>
<transferSet index="0" size="1"
  startTime="2008-11-02T21:28:09.281Z"
  total="1">
  <item mode="binary">
    <source>
      <queue>INPUT_QUEUE@QM1</queue>
    </source>
    <destination exist="error">
      <file>/home/user/output.file</file>
    </destination>
    <status resultCode="1">
      <supplement>Cancelling transfer - sent wrong data.</supplement>
    </status>
  </item>
</transferSet>
</transaction>

```

列出 MFT 代理

您可以使用命令行或 IBM MQ Explorer 列出向特定队列管理器注册的 Managed File Transfer 代理程序。

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](#)。

停止 MFT 代理

您可以从命令行停止 Managed File Transfer 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** 命令还会停止 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 128 页的『将 MFT 代理作为 Windows 服务启动』](#)。

在 z/OS 上停止 MFT 代理

如果要从 JCL 将 Managed File Transfer Agent on z/OS 作为启动式任务运行，那么代理程序将接受 z/OS 操作程序命令 **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，不受用户注销影响。

注:   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 9.0.2 和 IBM MQ 9.0.0 Fix Pack 1 开始，代理程序属性 **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 460 页的『MFT 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。

相关参考

第 131 页的『在 z/OS 上启动 MFT 代理』

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。

第 460 页的『MFT agent.properties 文件』

每个 Managed File Transfer Agent 都有自己的属性文件 **agent.properties**，该文件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。

协议网桥

通过协议网桥，Managed File Transfer (MFT) 网络可以访问您的 MFT 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。

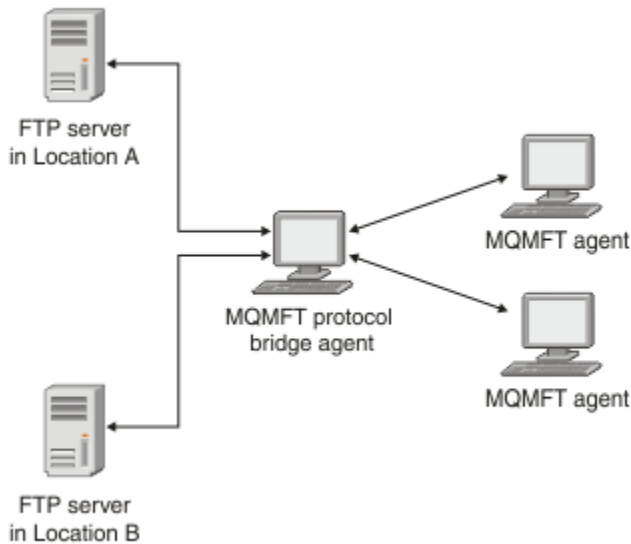
协议网桥作为 Managed File Transfer 的服务组件的一部分提供。在运行 MFT 的单个系统上可以有多个与不同文件服务器连接的专用代理程序。

您可以使用协议网桥代理同时将文件传输到多个端点。MFT 提供一个名为 **ProtocolBridgeProperties.xml** 的文件，您可以编辑该文件以定义要将文件传输至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 481 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 MFT 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 MFT 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 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 MFT 网络之间, 并且配置为与两个 FTP 服务器进行通信。

确保了协议网桥代理外, 在 MFT 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 MFT 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 481 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 185 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 Managed File Transfer 不能删除或编辑这些文件。在这种情况下, 传输失败。

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 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。

消息 BFGCL0392I 将提供 ProtocolBridgeProperties.xml 文件的位置。

```
<?xml version="1.0" encoding="IBM-1047"?>
<!--
This ProtocolBridgeProperties.xml file determines the protocol servers that will be accessed by
the
MQMFT protocol bridge agent.

Each protocol server is defined using either a <tns:ftpServer>, <tns:ftpsServer>, or
<tns:sftpServer>
element - depending on the protocol used to communicate with the server. When the protocol
bridge agent participates in a managed file transfer it will determine which server to use
based on
the prefix (if any) present on the file path. For example a file path of 'server1:/home/user/
file.txt' would
be interpreted as a request to transfer /home/user/file.txt using 'server1'. The server name
is compared
to the 'name' attribute of each <tns:ftpServer>, <tns:ftpsServer> or <tns:sftpServer> element
in this
XML document and the first match is used to determine which protocol server the protocol bridge
agent will connect to. If no match is found then the managed file transfer operation will fail.

If a file path is not prefixed with a server name, for example '/home/user/file.txt' then this
XML
document can specify a default server to use for the managed file transfer. To specify a
default server use the <tns:defaultServer> element as the first element inside the
<tns:serverProperties>
element. The default server will be used whenever the protocol bridge agent participates in
a managed file transfer for file names which do not specify a prefix.

An optional <tns:limits> element can be specified within each server definition. This element
contains
attributes that govern the amount of resources used by each defined server.

An optional <tns:credentialsFile> element can be specified within each serverProperties
definition. This
element contains a path to a file containing credentials to be used when connecting to defined
servers.

An example ProtocolBridgeProperties.xml file is as follows:

<?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="myFTPserver"/>

  <tns:ftpServer name="myFTPserver" host="windows.hursley.ibm.com" port="1234"
platform="windows"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
  listFormat="unix" limitedWrite="false">

    <tns:limits maxListFileNames="100" maxListDirectoryLevels="999999999"
      maxReconnectRetry="2" reconnectWaitPeriod="10"
      maxSessions="60" socketTimeout="30"/>

  </tns:ftpServer>

  <tns:ftpsServer name="myFTPSserver" host="unix.hursley.ibm.com" platform="unix"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF8"
  listFormat="unix" limitedWrite="false" ftpsType="explicit
```

```

        trustStore="C:\FTE\keystores\myFTPSServer\FTPSServer.jks"
trustStorePassword="password">

        <tns:limits maxReconnectRetry="10" connectionTimeout="10"/>

</tns:ftpServer>

<tns:sftpServer name="mySFTPserver" host="windows.hursley.ibm.com" platform="windows"
timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
limitedWrite="false">

        <tns:limits connectionTimeout="60"/>

</tns:sftpServer>
</tns:serverProperties>

```

This example shows the outermost <tns:serverProperties> element which must exist for the document to be valid, an optional <tns:defaultServer> element, as well as definitions for an FTP, FTPS and SFTP server.

The attributes of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements determine the characteristics of the connection established to the server. These attributes correspond to the command line parameters for the 'fteCreateBridgeAgent' command.

The following attributes are valid for all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: name, host, port, platform, fileEncoding, limitedWrite and controlEncoding.

The following attributes are valid for the <tns:ftpServer> and <tns:ftpsServer> elements: timeZone, locale, listFormat, listFileRecentDateFormat, listFileOldDateFormat, and monthShortNames.

The following attributes are valid for the <tns:ftpServer> element only: passiveMode

The following attributes are valid for the <tns:ftpsServer> element only: ftpsType, trustStore, trustStorePassword, trustStoreType, keyStore, keyStorePassword, keyStoreType, ccc, protFirst, auth, and connectTimeout.

The following attributes are valid for the <tns:limits> element within all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: maxListFileNames, maxListDirectoryLevels, maxReconnectRetry, reconnectWaitPeriod, maxSessions and socketTimeout

```

-->
<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">

    <!-- By default the location of the credentials file is in the home directory of the user
that started the -->
    <!-- protocol bridge agent. If you wish to specify a different location use the
credentialsFile element to -->
    <!-- do this. For
example: -->
    <!-- <tns:credentialsFile path="/test/
ProtocolBridgeCredentials.xml"/> -->

    <tns:defaultServer name="WINMVSCA.HURSLEY.IBM.COM"/>
    <tns:ftpServer name="WINMVSCA.HURSLEY.IBM.COM" host="WINMVSCA.HURSLEY.IBM.COM"
platform="UNIX"
        timeZone="Europe/London" locale="en-GB" fileEncoding="US-ASCII"
        listFormat="unix" limitedWrite="false" />

    <!-- Define servers here -->
</tns:serverProperties>

```

该命令可以生成以下消息：BFGCL0532I:

为使该代理正常运行，必须手工创建一个额外的凭证文件。缺省情况下，此文件称为 ProtocolBridgeCredentials.xml，位于主目录中启动代理程序的用户的目录。例如，如果此用户启动了代理程序位置将为：\$HOME/ProtocolBridgeCredentials.xml

如果使用凭证文件:

1. 请参阅以下文本以获取有关如何创建凭证文件的进一步信息。
2. 凭证文件必须位于具有受限许可权的目录中。例如，其他用户不能具有读访问权。
3. 在已启动代理的用户标识的 \$HOME 环境变量中为凭证文件指定目录的位置，或者编辑 ProtocolBridgeProperties.xml 文件并指定位置：

```
<tns:credentialsFile path="/test/ProtocolBridgeCredentials.xml"/>
```

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。

注：协议网桥代理不支持文件锁定。这是因为 Managed File Transfer 不支持文件服务器上的文件锁定机制。

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。其他任何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 481 页的『协议网桥属性文件格式』。

相关参考

第 481 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 549 页的『MFT 使用的正则表达式』

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。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。

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。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 780 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 183 页的『ProtocolBridgePropertiesExit: 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 460 页的『MFT 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 460 页的『MFT agent.properties 文件』。

使用用户出口样本

关于此任务

`MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` 目录和第 222 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 文件放在 `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` 文件以包含此行：

```
protocolBridgePropertiesExitClasses=SamplePropertiesExit2
```

5. 在目录 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` 中创建一个协议网桥属性文件，例如 `protocol_bridge_properties.properties`。编辑此文件以包含以下格式的行：

```
serverName=type://host:port
```

6. 编辑 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/agent.properties` 文件以包含此行：

```
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 177 页的『协议网桥』](#)

通过协议网桥，Managed File Transfer (MFT) 网络可以访问您的 MFT 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 780 页的『ProtocolBridgePropertiesExit2.java 接口』](#)

[第 222 页的『协议网桥属性用户出口样本』](#)

[第 460 页的『MFT agent.properties 文件』](#)

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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 327 页的『fteCreateBridgeAgent（创建并配置 MFT 协议网桥代理）』](#)

fteCreateBridgeAgent 命令将创建 Managed File Transfer 协议网桥代理及其关联配置。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。

ProtocolBridgePropertiesExit: 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 IBM 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 7.5 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 184 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 460 页的『[MFT 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 460 页的『[MFT 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 Managed File Transfer 中的用户凭证映射到文件服务器的用户凭证。Managed File Transfer 提供了执行用户凭证映射的样本用户出口。

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 Managed File Transfer 中的用户凭证映射到文件服务器的用户凭证。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 478 页的『协议网桥凭证文件格式』。

 使用 IBM WebSphere MQ 7.5 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 IS08859-1 ProtocolBridgeCredentials.xml
```

注： 在 z/OS 上，您可以将协议网桥凭证文件存储在数据集中，可以由用户指定 .xml 文件的名称。

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 IBM WebSphere MQ File Transfer Edition 7.0.4 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 IBM WebSphere MQ File Transfer Edition 7.0.4 Fix Pack 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。将以下行插入到文件中：

```
<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 Java 正则表达式，用于匹配与 MFT 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，那么会改为使用与 MFT 传输请求关联的 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 MFT 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 Java 正则表达式，用于匹配与 MFT 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，那么会改为使用与 MFT 传输请求关联的 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 MFT 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，那么会使用对应的 serverUserId 和 key 值，向协议文件服务器认证 MFT 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 188 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: z/OS

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。

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 Managed File Transfer 中的用户凭证映射到文件服务器上的用户凭证中。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。

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 778 页的『`ProtocolBridgeCredentialExit.java` 接口』和第 779 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的 `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 `mqMQUserId` 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 文件放在网桥代理的 `exits` 目录中。该目录位于 `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 460 页的『`MFT 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_name\exits\SampleCredentialExit.jar
```

代理的 `agent.properties` 文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/bridge_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。为此属性指定的值作为字符串对象传入到由 `protocolBridgeCredentialExitClasses` 指定的出口类的 `initialize()` 方法。有关更多信息，请参阅第 460 页的『`MFT 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 220 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。该样本基于 `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. 创建包含已编译出口及其包结构的 JAR 文件。
3. 将 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` 文件以包含此行：

```
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。

6. 在目录 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 中创建凭证属性文件 (`credentials.properties`) 并编辑该文件以包含以下格式的条目：

```
mqUserId=serverUserId,serverPassword
```

7. 编辑 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` 文件以包含此行：

```
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。

8. 通过使用 `fteStartAgent` 命令来启动协议网桥代理。

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

1. 生成要用于向 SFTP 服务器进行认证的公用密钥和专用密钥。

例如，在 Linux 主机系统上，可以使用作为 "openssh" 包的一部分提供的工具 **ssh-keygen** 来创建公用/专用密钥对。

缺省情况下，在没有参数的情况下，**ssh-keygen** 命令会提示输入两个密钥文件的位置和口令，缺省为名称：

```
id_rsa      <-- Private key
id_rsa.pub  <-- Public key
```



注意: 如果要从最新版本的 OpenSSH(例如 RHEL 8 随附的命令) 使用 **ssh-keygen** 命令，那么所使用的密钥格式与协议网桥代理不兼容，并且对 SFTP 服务器的传输尝试将失败并返回以下消息：

```
BFGBR0216E: Authentication to protocol server 'sftp.host.address' failed
because of invalid private key.
```

要使用这些较新版本的 OpenSSH 创建兼容的专用密钥，请使用 **ssh-keygen** 命令的以下自变量指定密钥格式：

```
ssh-keygen -m PEM
```

然后，`id_rsa` 专用密钥的内容具有以下第一行和最后一行：

```
-----BEGIN RSA PRIVATE KEY-----
...
-----END RSA PRIVATE KEY-----
```

与协议网桥代理兼容。

2. 将 `id_rsa.pub` 文件的整个内容复制到 SFTP 服务器上 SFTP 用户的 `~/.ssh/authorized_keys` 文件中。

确保为此文件和 `~/.ssh` 目录设置相应的文件许可权，以便 SFTP 服务器允许密钥认证。这些许可权通常为：

```
~/.ssh          Mode 700
~/.ssh/authorized_keys  Mode 600
```

3. Managed File Transfer 需要使用 MD5 算法生成主机 ssh 指纹。运行下列其中一个命令以获取 SFTP 服务器的主机 ssh 指纹。

- 对于 Red Hat® Enterprise Linux V 6.x 和更低版本以及 Linux Ubuntu 14.04，请运行以下命令：

```
ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
```

- 从 Red Hat Enterprise Linux V 7.x，Linux Ubuntu 16.04 和 SuSE Linux 12.4 开始，缺省情况下，`ssh-keygen` 命令使用 SHA56 算法生成 ssh 指纹。要使用 MD5 算法生成 ssh 指纹，请运行以下命令：

```
ssh-keygen -l -E MD5 -f /etc/ssh/ssh_host_rsa_key.pub
```

该命令的输出将类似于以下示例：

```
2048 MD5:64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27 no comment (RSA)
```

仅抽取输出的十六进制部分以用作 `ProtocolBridgeCredentials.xml` 文件中的 `hostKey` (请参阅步骤第 189 页的『4』)。因此，在此示例中，您将抽取 `64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27`。

4. 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。

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 327 页的『[fteCreateBridgeAgent \(创建并配置 MFT 协议网桥代理\)](#)』。

如果您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 IBM 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 179 页的『[使用 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性](#)』。您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 481 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 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 185 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"/>

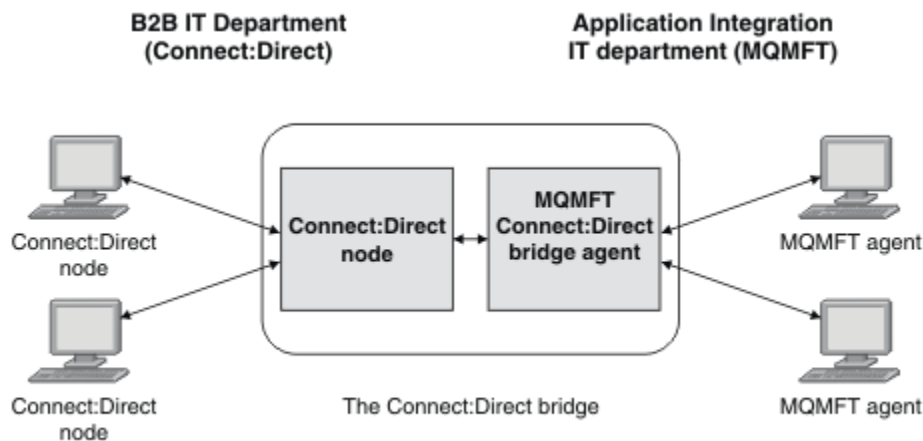
  <!-- Define servers here -->
</tns:serverProperties>
```

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 554 页的『[协议网桥提供的 FTPS 服务器支持](#)』。

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 Managed File Transfer 组件的 Connect:Direct 网桥在 MFT 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 MFT 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 IBM MQ 作为其消息传递基础结构, 因此选择了 Managed File Transfer 作为其文件传输解决方案。

通过使用 MFT Connect:Direct 网桥, 这两个部门可以在 B2B IT 部门中的 Connect:Direct 网络和应用程序集成 IT 部门中的 MFT 网络之间传输文件。Connect:Direct 网桥是 Managed File Transfer 的一个组件, 它包含一个 MFT 代理, 用于和 Connect:Direct 节点通信。MFT 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 网桥可用作 Managed File Transfer 的服务和代理组件的一部分, 可以用于以下任务:

1. 使用 Managed File Transfer 命令启动从 MFT 代理到 Connect:Direct 节点的一个或多个文件的传输。
2. 使用 Managed File Transfer 命令启动从 Connect:Direct 节点到 MFT 代理的一个或多个文件的传输。
3. 使用 Managed File Transfer 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 MFT 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 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 530 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 MFT Connect:Direct 网桥代理和一个 Connect:Direct 节点组成。此代理在 Windows 和 Linux for System x 上受支持。在 IBM Sterling Connect:Direct for Windows 和 IBM Sterling Connect:Direct for UNIX 支持的平台上支持此节点。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 118 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 [IBM 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。此文件系统用于在涉及 Connect:Direct 网桥的文件传输期间将文件临时存储在由 `cdTmpDir` 参数定义的目录中。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:

- 代理和节点在同一系统上, 该系统运行 Windows 或 Linux for System x
- 代理在 Linux for System x 上, 而该节点在 UNIX 上
- 代理在一个 Windows 系统上, 而节点在另一个 Windows 系统上

以下配置不允许代理和节点使用同一路径名:

- 代理在 Linux for System x 上, 而该节点在 Windows 上
- 代理在 Windows 上, 而该节点在 UNIX 上

请在计划安装 Connect:Direct 网桥时考虑该限制。

向 Connect:Direct 节点传输文件

您可以使用 Connect:Direct 网桥从 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, 将 Connect:Direct 节点指定为传输的目标。

开始之前

在传输文件之前，必须配置作为 Managed File Transfer 组件的 Connect:Direct 网桥。有关更多信息，请参阅 [第 118 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. 使用 fteCreateTransfer 命令，按如下格式指定 **-df**（目标文件）参数的值：
`connect_direct_node_name:file_path`，并且 **-da**（目标代理）参数的值指定为 Connect:Direct 网桥代理的名称。

注：由 `connect_direct_node_name` 指定的 Connect:Direct 节点是希望将文件传输给的节点，而不是作为 Connect:Direct 网桥的一部分运行的 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 364 页的『fteCreateTransfer: 启动新的文件传输』](#)。

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 191 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 Managed File Transfer 组件的 Connect:Direct 网桥在 MFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关任务

[第 193 页的『从 Connect:Direct 节点传输文件』](#)

您可以使用 Connect:Direct 网桥，将一个文件从 Connect:Direct 节点传输到 Managed File Transfer Agent。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 460 页的『MFT agent.properties 文件』](#)

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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，将一个文件从 Connect:Direct 节点传输到 Managed File Transfer Agent。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 Managed File Transfer 组件的 Connect:Direct 网桥。请参阅 [第 118 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 Managed File Transfer 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：由 `connect_direct_node_name` 指定的 Connect:Direct 节点是希望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 364 页的『[fteCreateTransfer: 启动新的文件传输](#)』。

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。在从 Connect:Direct 节点传输文件期间，Connect:Direct 网桥将文件临时存储在 `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 191 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 Managed File Transfer 组件的 Connect:Direct 网桥在 MFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关参考

第 460 页的『[MFT agent.properties 文件](#)』

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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。

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，从 z/OS 上的 Managed File Transfer 代理将数据集传输到 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 Managed File Transfer 组件的 Connect:Direct 网桥。请参阅第 118 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 Managed File Transfer 时，参数 `-df` 有效并可供使用。如果源代理是 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本，那么可以改为使用 `-ds` 参数。源代理称为 FTE_ZOS1，并且是 IBM WebSphere MQ File Transfer Edition 7.0.3 代理程序。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. 使用 `fteCreateTransfer` 命令，按如下格式指定 `-df` 参数的值：

`connect_direct_node_name:data_set_name;attributes`，并且 `-da`（目标代理）参数的值指定为 Connect:Direct 网桥代理的名称。

由 `connect_direct_node_name` 指定的 Connect:Direct 节点是希望将数据集传输给的节点，而不是作为 Connect:Direct 网桥的一部分运行的 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 364 页的『fteCreateTransfer: 启动新的文件传输』](#)。

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 191 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 Managed File Transfer 组件的 Connect:Direct 网桥在 MFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关参考

[第 530 页的『与 Connect:Direct 节点之间传输数据集』](#)

您可以使用 Connect:Direct 网桥在 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 531 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 Managed File Transfer Agent 传输到 Connect:Direct 节点。要将 Connect:Direct 节点作为多文件传输的目标，可将 Connect:Direct 网桥代理指定为目标代理，并指定 `connect_direct_node_name:directory_path` 格式的目标目录。

开始之前

在传输文件之前，必须配置作为 Managed File Transfer 组成部分的 Connect:Direct 网桥。请参阅 [第 118 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`。将 `-da`（目标代理）参数的值指定为 Connect:Direct 网桥代理的名称。

注：由 `connect_direct_node_name` 指定的 Connect:Direct 节点是希望将文件传输给的节点，而不是作为 Connect:Direct 网桥的一部分运行的 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 364 页的『fteCreateTransfer: 启动新的文件传输』](#)。

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 191 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 Managed File Transfer 组件的 Connect:Direct 网桥在 MFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关任务

[第 192 页的『向 Connect:Direct 节点传输文件』](#)

您可以使用 Connect:Direct 网桥从 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，将 Connect:Direct 节点指定为传输的目标。

[第 197 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 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 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 193 页的『从 Connect:Direct 节点传输文件』](#)

您可以使用 Connect:Direct 网桥，将一个文件从 Connect:Direct 节点传输到 Managed File Transfer Agent。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 196 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 Managed File Transfer Agent。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 460 页的『MFT agent.properties 文件』](#)

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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 Managed File Transfer Agent。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 Managed File Transfer 组件的 Connect:Direct 网桥。请参阅[第 118 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: 由 `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。

```
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 364 页的『[fteCreateTransfer: 启动新的文件传输](#)』。

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。在从 Connect:Direct 节点传输文件期间, Connect:Direct 网桥将文件临时存储在 `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 191 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 Managed File Transfer 组件的 Connect:Direct 网桥在 MFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关参考

第 460 页的『[MFT agent.properties 文件](#)』

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`, 该文件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 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 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 Managed File Transfer 组件的 Connect:Direct 网桥。有关更多信息, 请参阅第 118 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, 请指定 Connect:Direct 网桥代理。

注: 由 `connect_direct_node_name` 指定的 Connect:Direct 节点是希望将文件传输给的节点, 而不是作为 Connect:Direct 网桥的一部分运行的 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 364 页的『[fteCreateTransfer: 启动新的文件传输](#)』。

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 `/reports/report??.log` 匹配的每个源文件重复该过程。

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。
- 如果源代理位于运行 Linux 或 UNIX 操作系统的系统上，那么模式匹配将区分大小写。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。

恢复并重新启动与 Connect:Direct 节点之间的传输

在传输期间，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。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 Managed File Transfer 将通过以下方式恢复文件传输：

- 如果 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。
- 如果先前在此传输请求期间 Managed File Transfer 成功连接到了 Connect:Direct 节点，那么在由 **cdMaxPartialWorkConnectionRetries** 和 **recoverableTransferRetryInterval** 属性值确定的时间长度之后将再次尝试该传输。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。
- 对于特定类型的 Connect:Direct 节点故障，如强行停止的节点，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，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 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 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 120 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 Managed File Transfer 传输相同的方式恢复。有关更多信息，请参阅第 204 页的『[MFT 恢复和重新启动](#)』。

相关概念

第 191 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 Managed File Transfer 组件的 Connect:Direct 网桥在 MFT 和 IBM Sterling Connect:Direct 之间传输文件。

第 204 页的『[MFT 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 Managed File Transfer 将在这些场景下按如下方式恢复：

相关任务

第 118 页的『[配置 Connect:Direct 网桥](#)』

配置 Connect:Direct 网桥，以在 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 Managed File Transfer 代理。该代理称为 Connect:Direct 网桥代理。

相关参考

第 460 页的『[MFT agent.properties 文件](#)』

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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。

然而，您可以通过使用 `ConnectDirectProcessDefinition.xml` 文件来配置 Connect:Direct 网桥代理，以调用用户定义的 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 `ConnectDirectProcessDefinitions.xml` 文件。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 125 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 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，Managed File Transfer 使用的所有内置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 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

Managed File Transfer 提供了样本 Connect:Direct 进程。这些样本位于以下目录中：
`MQ_INSTALLATION_PATH/mqft/samples/ConnectDirectProcessTemplates`。

使用 `ConnectDirectProcessDefinition.xml` 文件指定要启动的 `Connect:Direct` 进程

指定要在 Managed File Transfer 传输过程中启动的 Connect:Direct 进程。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 `ConnectDirectProcessDefinitions.xml` 文件。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文件 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml`。
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 550 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 550 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` 网桥代理才能够调用由该 `<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。此文件的位置是使用绝对路径或相对于 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` 目录指定的。

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 118 页的『配置 `Connect:Direct` 网桥』](#)

配置 `Connect:Direct` 网桥，以在 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 Managed File Transfer 代理。该代理称为 `Connect:Direct` 网桥代理。

相关参考

[第 492 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` 文件指定用户定义的 `Connect:Direct` 进程，以在文件传输过程中启动。

第 335 页的『fteCreateCDAgent (创建 Connect:Direct 网桥代理)』

fteCreateCDAgent 命令用于创建 Managed File Transfer Agent 及其相关联的配置，以与 Connect:Direct 网桥一起使用。

在由 *Managed File Transfer* 调用的 *Connect:Direct* 进程中使用内置符号变量

您可以在 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 Managed File Transfer 传输中的信息传递到用户定义的 Connect:Direct 进程中。有关 Managed File Transfer 使用的内置符号变量的更多信息，请参阅第 550 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 Managed File Transfer Agent 传输到 Connect:Direct 网桥节点。该传输的第一部分由 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. 将此进程保存到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，以包含调用在步骤 1 中创建的 Connect:Direct 进程的规则。

```
<?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>
```

在该示例中，如果将传输请求提交给将 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。

4. 提交满足在步骤 3 的 `ConnectDirectProcessDefinition.xml` 文件中定义的条件文件传输请求。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。

5. 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。

6. 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。

7. 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 199 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 550 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 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 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。Managed File Transfer 提供了可以从 Connect:Direct 进程中的 **RUN TASK** 语句调用的命令。

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 364 页的『fteCreateTransfer: 启动新的文件传输』。

ftebxfer

指定该命令，以在传输请求提交给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 364 页的『fteCreateTransfer: 启动新的文件传输』。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 364 页的『[fteCreateTransfer: 启动新的文件传输](#)』。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。要向其提交命令的队列管理器的主机和端口，以 IBM 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 203 页的『[通过使用 Connect:Direct 请求程序来创建和提交调用 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 Managed File Transfer 的 Connect:Direct 进程。

相关参考

第 552 页的『[示例：用于调用 MFT 命令的 Connect:Direct 进程文件](#)』

调用 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 *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 Managed File Transfer 的 Connect:Direct 进程。

关于此任务

此任务描述如何创建调用 Managed File Transfer **ftecxfer** 命令或 **ftebxfer** 命令的 Connect:Direct 进程。当传输请求提交给的队列管理器与提交该命令的 Connect:Direct 节点位于不同系统上时，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 Managed File Transfer 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. 创建一条语句以运行 Managed File Transfer **ftetag** 命令。
 - 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 364 页的『[fteCreateTransfer: 启动新的文件传输](#)』。

- e) 单击**确定**。此时将关闭“运行任务语句”窗口。
10. 创建一条语句以运行 Managed File Transfer **ftecxf** 或 **ftebxf** 命令。
 - 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/ftecxf) args(parameters)` 或 `pgm(MQ_INSTALLATION_PATH/bin/ftebxf) args(parameters)`。**ftecxf** 和 **ftebxf** 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 **ftecxf** 和 **ftebxf** 的额外参数。有关更多信息，请参阅第 364 页的『[fteCreateTransfer: 启动新的文件传输](#)』和第 202 页的『[使用 Connect:Direct 进程来提交 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 202 页的『[使用 Connect:Direct 进程来提交 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。Managed File Transfer 提供了可以从 Connect:Direct 进程中的 **RUN TASK** 语句调用的命令。

从 IBM Integration Bus 使用 MFT

您可以通过 FTEOutput 和 FTEInput 节点使用来自 IBM Integration Bus 的 Managed File Transfer。

- 通过 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 Managed File Transfer 的任何级别。

有关更多信息，请参阅 [IBM Integration Bus 产品文档](#)。

MFT 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 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，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 IBM 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。

相关概念

第 205 页的『恢复中文件传输的超时选项』

您可以设置时间量（以秒为单位），在此期间，源代理会一直尝试恢复停止的文件传输。如果代理达到重试时间间隔的超时值时传输不成功，则传输失败。

V 9.0.1 恢复中文件传输的超时选项

您可以设置时间量（以秒为单位），在此期间，源代理会一直尝试恢复停止的文件传输。如果代理达到重试时间间隔的超时值时传输不成功，则传输失败。

在 IBM MQ 9.0.1 之前，Managed File Transfer 源代理的缺省行为是一直尝试恢复停滞的传输，直至传输成功。因为新参数为可选参数，您不一定要设置此参数，传输可遵循缺省行为。

从 IBM MQ 9.0.1 开始，您可以指定以下选项之一：

-1

代理继续尝试恢复停止的传输，直至传输成功。使用此选项相当于未设置属性时代理的缺省行为。

0

一旦进入恢复，代理将停止文件传输。

>0

在由指定正整数值设置的时间量（以秒为单位）内，代理继续尝试恢复停止的传输。值 21600 表示代理会在进入恢复后六小时内一直尝试恢复传输。此参数的最大值为 999999999。

为一个源代理的所有传输设置恢复重试超时

要设置适用于源代理的所有传输的恢复超时，请将此参数和值对添加到 `agent.properties` 文件中。

在此示例中，将传输恢复超时值设置为 -1 表示代理会继续尝试恢复停止的传输，直至传输成功完成。

```
transferRecoveryTimeout=-1
```

将传输恢复超时值设置为 0 表示代理会在进入恢复后立即将传输标记为失败：

```
transferRecoveryTimeout=0
```

将传输恢复超时值设置为 21600 表示将传输标记为失败之前的六小时，代理会一直重试停止的传输：

```
transferRecoveryTimeout=21600
```

设置或覆盖个别传输的恢复重试超时

在执行以下操作时，您可以为个别传输设置恢复重试超时参数：

- 在 Managed File Transfer 命令行界面或 IBM MQ Explorer 中使用 `fteCreateTransfer` 命令创建传输
- 在 Managed File Transfer 命令行界面或 IBM MQ Explorer 中使用第 351 页的『`fteCreateTemplate`: 创建新的文件传输模板』命令创建传输模板
- 使用 `fteCreateMonitor` Managed File Transfer 命令行接口或 IBM MQ Explorer 创建监视器
- 使用 `fte:filecopy` 或 `fte:filemove` Ant 任务复制或移动文件

将传输的 **-rt** 值设置为 **-1** 相当于缺省行为，即恢复将一直继续，直至传输成功或被用户手动取消，例如

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt -1 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

有关在创建新传输时使用传输恢复参数的更多信息，请参阅 [fteCreateTransfer](#) 命令。

将 **-rt** 参数值设置为 **0** 表示如果使用此模板启动的传输停止，传输将立即失败且不会尝试恢复，例如

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 0 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

有关在创建新传输模板时使用传输恢复超时参数的更多信息，请参阅 [fteCreateTemplate](#) 命令。

您可以使用 IBM MQ Explorer 为传输设置恢复超时参数和值。有关使用 IBM MQ Explorer 配置传输的更多信息，请参阅第 132 页的『启动新的文件传输』和第 163 页的『使用 IBM MQ Explorer 创建文件传输模板』。

您还可以将恢复超时设置为包含 **transferRecoveryTimeout** 选项和值，其中带有用于通过 Ant 任务移动或复制文件的 **fte:filecopy** 或 **fte:filemove** 元素，例如

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="copy.result" transferRecoveryTimeout="0">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filecopy>
```

文件复制任务启动且传输进入恢复后，传输将立即停止，不尝试任何恢复。使用 **fte:filecopy** 或 **fte:filemove** 设置 **transferRecoveryTimeout** 选项，以覆盖 `agent.properties` 文件中设置的值。如果 **transferRecoveryTimeout** 值没有设置 **fte:filecopy** 或 **fte:filemove**，将使用来自 `agent.properties` 文件的 **transferRecoveryTimeout** 参数的值。有关更多信息，请参阅第 730 页的『fte:filecopy Ant 任务』和第 733 页的『fte:filemove Ant 任务』。

处理恢复超时优先顺序

通过创建传输，模板或监视器命令的命令行界面参数（包括在 IBM MQ Explorer 向导中设置选项）或在 **fte:filespec** 嵌套元素中指定的传输恢复超时值优先于在源代理的 `agent.properties` 文件中为 **transferRecoveryTimeout** 参数指定的值。例如，对于命令

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

（不以 **-rt** 参数和值对开头），源代理 AGENT1 将检查 `agent.properties` 文件的 **transferRecoveryTimeout** 值来确定恢复超时行为。

如果在 `agent.properties` 文件中，**transferRecoveryTimeout** 未设置或设置为 **-1**，那么代理遵循缺省行为，并在传输成功之前一直尝试恢复传输。

通过 Managed File Transfer 命令行界面（例如，使用 **fteCreateTransfer** 命令）指定恢复超时选项 **-rt** 时，此值优先于 `agent.properties` 文件中的值，并用作传输的设置。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

处理恢复超時計数器

恢复超時計数器在传输进入恢复状态时启动。将向带有主题字符串 `Log/agent_name/transfer_ID` 的 `SYSTEM.FTE` 主题发布传输日志消息，用以指示传输状态已更改为正在恢复以及状态更改时的源代理时间。如果传输在设置的重试时间间隔内恢复，且没有达到恢复超时（计数器<=恢复超时），那么计数器将重置为 0，并准备在传输进入恢复时重新启动。

如果计数器达到为恢复超时设置的最大值（计数器==恢复超时），那么恢复传输将停止，源代理将此传输报告为失败。这种因为传输达到恢复超时而导致的传输失败由新消息代码 `RECOVERY TIMEOUT (69)` 来指

示。将向带有主题字符串 Log/agent_name/transfer_ID 的 SYSTEM.FTE 主题发布另一个传输日志消息，以指示传输失败，其中包括新消息、新返回码和源代理的事件日志。恢复期间发生以下任意事件时，源代理的事件日志将更新消息。

- 当恢复超时参数设置为大于 -1 的值时，Managed File Transfer 会进入恢复。代理的事件日志将发生更新，用以指示 **TransferId** 的恢复计时器开始以及源代理启动恢复超时处理之前要等待的时间量。
- 恢复 Managed File Transfer 复原时，源代理的事件日志将以新消息更新，用以指示曾处于恢复的 **TransferId** 已复原。
- 当恢复 Managed File Transfer 超时时，源代理的事件日志将发生更新，以指示 **TransferId** 在恢复过程中由于恢复超时而失败。

这些日志消息使用户（订户和记录器）能够识别因为传输恢复超时而失败的传输。

恢复超时计数器始终位于源代理中。但是，如果目标代理无法及时接收来自源代理的信息，目标代理会向源代理发送将此传输放入恢复的请求。对于设置了恢复超时选项的传输，源代理会在收到目标代理的请求时启动恢复超时计数器。

对于不使用恢复超时选项的传输，失败的传输和部分完成的传输，仍需要手动处理。

对于向多个文件发送了单个传输请求，且某些文件已成功完成，但一个文件只是部分完成的传输集合，此传输仍会标记为失败，因为传输没有按照预期完成。在传输部分完成的文件时，源代理可能就已经超时。

确保目标代理和文件服务器已经准备好接受文件传输。

您必须再次为整个集合发出传输请求，但为了避免因初次传输尝试时目标上保留了某些文件而导致的问题，发出新请求时可以指定覆盖现有文件选项。此选项可以确保文件再次写入目标之前，之前传输中的不完整文件集将作为新传输的一部分被清除。

跟踪和消息

包含跟踪点的目的在于诊断。将记录恢复超时值、重试时间间隔开始时间、恢复时间段和计数器重置开始时间以及传输是否超时和失败。在发生问题或出现异常行为的情况下，您可以收集源代理输出日志和跟踪文件，并在 IBM 支持人员请求时提供这些资料以帮助进行故障诊断。

当传输进入恢复 (BFGTR0081I)、因为恢复超时而终止 (BFGSS0081E) 以及从恢复中复原 (BFGTR0082I) 时，会有消息通知用户。

BFGTR0001 - BFGTR9999

BFGTR0081I

BFGTR0082I

BFGSS0001 - BFGSS9999

BFGSS0081E

为 MFT 开发应用程序

指定要使用 MFT 运行的程序

您可在正在运行 Managed File Transfer Agent 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。

用户出口和程序调用按以下顺序进行调用：

```
- SourceTransferStartExit(onSourceTransferStart).
- PRE_SOURCE Command.
- DestinationTransferStartExits(onDestinationTransferStart).
- PRE_DESTINATION Command.
- The Transfer request is performed.
- DestinationTransferEndExits(onDestinationTransferEnd).
- POST_DESTINATION Command.
- SourceTransferEndExits(onSourceTransferEnd).
- POST_SOURCE Command.
```

注意:

1. 仅当传输成功或部分成功完成时，才会运行 **DestinationTransferEndExits**。
2. 仅当传输成功或部分成功完成时，才会运行 **postDestinationCall**。
3. 针对成功、部分成功或失败的传输运行 **SourceTransferEndExits**。
4. 仅在以下情况下调用 **postSourceCall**:
 - 未取消该传输。
 - 存在成功或部分成功的结果。
 - 任何目标后传输程序已成功运行。

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 744 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 678 页的『MFT 代理调用请求消息示例』。

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 364 页的『fteCreateTransfer: 启动新的文件传输』。有关使用该命令的示例, 请参阅第 723 页的『使用 fteCreateTransfer 来启动程序的示例』。

相关参考

第 303 页的『commandPath MFT 属性』

使用 **commandPath** 属性以指定 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。

受管调用

Managed File Transfer (MFT) 代理程序通常用来传输文件或消息。它们称为受管传输。代理程序也可用于运行命令、脚本或 JCL, 而无需传输文件或消息。此功能称为受管调用。

受管调用请求可以通过多种方式提交给代理程序:

- 使用 **fte: call** Ant 任务。
- 使用用于运行命令或脚本的任务 XML 配置资源监视器。请参阅配置监视器任务以启动命令和脚本来获取更多信息。
- 将 XML 消息直接放入代理程序的命令队列中。有关受管调用 XML 模式的更多详细信息, 请参阅文件传输请求消息格式。

对于受管调用，必须在代理程序属性 **commandPath** 中指定包含要运行的命令或脚本的目录。

受管调用无法运行位于代理程序的 **commandPath** 中未指定的目录内的命令或脚本。这是为了确保该代理程序不会运行任何恶意代码。

此外，您还可以对代理程序启用权限检查，以确保仅允许授权用户提交受管调用请求。有关更多信息，请参阅限制针对 MFT 代理程序操作的用户权限。

作为受管调用的一部分调用的命令、脚本或 JCL 将作为外部进程运行，该进程由代理程序监视。当进程退出时，受管调用将完成，并且来自进程的返回码可供调用 **fte: Ant call** Ant 任务的代理或脚本使用。

如果受管调用是由 **fte:call** Ant 任务启动的，那么 Ant 脚本可以检查返回码的值以确定受管调用是否成功。

对于所有其他类型的受管调用，您可以指定应该使用哪些返回码值来指示受管调用已成功完成。外部进程完成时，代理程序会将来自进程的返回码与这些返回码进行比较。

注: 由于受管调用作为外部进程运行，因此一旦启动，就无法取消这些调用。

受管调用和源传输插槽

代理程序包含多个源传输插槽（由代理程序属性 **maxSourceTransfers** 指定）如高级代理程序属性：[传输限制](#)中所述。

每当运行受管调用或受管传输时，它们会占用源传输插槽。该插槽在受管调用或受管传输完成时释放。

如果在代理程序收到新的受管调用或受管转移请求时所有源传输插槽都在使用中，则该请求将由代理程序排队，直到有可用插槽。

如果受管调用启动受管传输（例如，如果受管调用运行 Ant 脚本并且 Ant 脚本使用 [fte: filecopy](#) 或 [fte: filemove](#) 任务来传输文件），那么需要两个源传输插槽：

- 一个用于受管传输
- 一个用于受管调用

在这种情况下，请务必注意，如果受管传输需要很长时间才能完成或进入恢复，那么将占用两个源传输插槽，直到受管传输完成或取消

因此，您应该考虑受管调用的设计，以确保它在很长一段时间内不会占用源传输插槽。

将 Apache Ant 与 MFT 结合使用

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** 命令在已配置的 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。

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 210 页的『入门：将 Ant 脚本用于 MFT』](#)

将 Ant 脚本用于 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 726 页的『MFT 提供的 Ant 任务』](#)

Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

[第 315 页的『fteAnt: 在 MFT 中运行 Ant 任务』](#)

fteAnt 命令会在具有可用的 Managed File Transfer Ant 任务的环境中运行 Ant 脚本。

[第 211 页的『MFT 的样本 Ant 任务』](#)

随 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 MFT

将 Ant 脚本用于 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。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 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 315 页的『[fteAnt: 在 MFT 中运行 Ant 任务](#)』。

返回码

文件传输 Ant 任务返回与 Managed File Transfer 命令相同的返回码。有关更多信息，请参阅第 282 页的『[MFT 的返回码](#)』。

相关参考

第 726 页的『[MFT 提供的 Ant 任务](#)』

Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

第 315 页的『[fteAnt: 在 MFT 中运行 Ant 任务](#)』

`fteAnt` 命令会在具有可用的 Managed File Transfer Ant 任务的环境中运行 Ant 脚本。

第 211 页的『[MFT 的样本 Ant 任务](#)』

随 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。

MFT 的样本 Ant 任务

随 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 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 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)

IBM i

IBM i 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

IBM i 上的 IBM WebSphere MQ File Transfer Edition 7.0.2 不包含对传输本机 IBM i 库对象的直接支持。librarytransfer 样本通过 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 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.

IBM i

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 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。

IBM i

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`

IBM i

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)

IBM i 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 IBM i 上的 IBM WebSphere MQ File Transfer Edition 7.0.2 并不直接支持传输 IBM i 上的本机源物理或数据库文件。physicalfiletransfer 样本通过 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 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.

IBM i 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 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。

IBM i 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

IBM i 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 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 Managed File Transfer 第 727 页的『fte:awaitoutcome Ant 任务』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 Managed File Transfer 第 729 页的『fte:cancel Ant 任务』任务将用于取消文件传输。

vsamtransfer



z/OS vsamtransfer 样本演示如何通过 Managed File Transfer 使用 Ant 任务从 VSAM 数据集传输到另一个 VSAM 数据集。Managed File Transfer 当前不支持传输 VSAM 数据集。该样本脚本通过使用 presrc 嵌套元素调用可执行文件 datasetcopy.sh，将 VSAM 数据记录卸载到连续数据集。该脚本使用 Managed File Transfer 第 733 页的『fte:filemove Ant 任务』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。

z/OS 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。

z/OS 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 Managed File Transfer 第 733 页的『fte:filemove Ant 任务』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](#)。

使用用户出口定制 MFT

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

Managed File Transfer 在代码中提供了一些“点”，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，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 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。下表概述了以下各个用户出口点以及要使用出口点而必须实现的 Java 接口。

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。

用户出口和程序调用按以下顺序进行调用：

- SourceTransferStartExit(onSourceTransferStart).
- PRE_SOURCE Command.
- DestinationTransferStartExits(onDestinationTransferStart).
- PRE_DESTINATION Command.
- The Transfer request is performed.
- DestinationTransferEndExits(onDestinationTransferEnd).
- POST_DESTINATION Command.
- SourceTransferEndExits(onSourceTransferEnd).
- POST_SOURCE Command.

注意:

1. 仅当传输成功或部分成功完成时，才会运行 **DestinationTransferEndExits**。
2. 仅当传输成功或部分成功完成时，才会运行 **postDestinationCall**。
3. 针对成功、部分成功或失败的传输运行 **SourceTransferEndExits**。
4. 仅在以下情况下调用 **postSourceCall**:
 - 未取消该传输。
 - 存在成功或部分成功的结果。
 - 任何目标后传输程序已成功运行。

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 753 页的『[用户出口的 MFT 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、在本地对其进行加密，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。

MFT 源和目标用户出口

目录分隔符

无论您如何在 **fteCreateTransfer** 命令或 IBM MQ Explorer 中指定目录分隔符，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 指定了此源文件，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 755 页的『MFT 用户出口的 Java 接口』](#)

使用本部分中的主题来获取有关用户出口例程的 Java 接口的参考信息。

相关参考

[第 218 页的『为 MFT 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 219 页的『MFT 源传输用户出口样本』](#)



[第 749 页的『MFT 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 MFT 传输 I/O 用户出口

您可以使用 Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 Managed File Transfer 传输的底层文件系统 I/O 工作。

通常对于 MFT 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（仅限 Connect:Direct 网桥代理）

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。

z/OS 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 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。

IBM i 上的样本 MFT 用户出口

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.qdls.FTEQDLSExit

com.ibm.wmqfte.exit.io.ibm.qdls.FTEQDLSExit 样本用户出口传输 IBM i 上 QDLS 文件系统中的文件。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.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.qdls.FTEQDLSExit 添加到 IOExitClasses 属性。
3. 重新启动代理。

com.ibm.wmqfte.exit.user.ibm.FileMemberMonitorExit

com.ibm.wmqfte.exit.user.ibm.FileMemberMonitorExit 样本用户出口的行为方式类似于 MFT 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.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.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.EmptyFileDeleteExit

当在传输期间删除源文件成员时，com.ibm.wmqfte.exit.user.ibm.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 MFT 将文件对象视为目录处理。所以，无法使用 MFT 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.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.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 216 页的『使用 MFT 传输 I/O 用户出口』

您可以使用 Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 Managed File Transfer 传输的底层文件系统 I/O 工作。

第 753 页的『用户出口的 MFT 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 MFT 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的 Java 虚拟机中运行，因此您无法直接使用调试支持（通常包含在集成开发环境中）。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，而非使用 **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](#) 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。

MFT 源传输用户出口样本

```
/*
 * 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 186 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 181 页的『[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 MFT

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 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 663 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 679 页的『MFT 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 688 页的『Ping MFT 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：

对 MFT 进行故障诊断

使用以下参考信息可帮助您诊断 Managed File Transfer 中的错误：

MFT 一般故障诊断

使用以下参考信息可帮助您诊断 Managed File Transfer 中的错误：

相关概念

[第 227 页的『使用 MFT 的提示与技巧』](#)

以下建议可帮助您最有效地使用 Managed File Transfer。

[第 271 页的『用于将 MFT 代理或记录器作为 Windows 服务运行的指南』](#)

您可以将 Managed File Transfer 代理、独立数据库记录器和独立文件记录器作为 Windows 服务运行。如果这些 Windows 服务出现任何问题，您可以使用服务日志文件以及本主题中的信息来诊断问题。

相关任务

[第 228 页的『在 MFT 上运行跟踪』](#)

您可以使用本部分中描述的方法来跟踪 Managed File Transfer。

[第 275 页的『在 Oracle 数据库上更新 MFT 数据库模式期间收到错误时该怎么做』](#)

使用 `ftelog_tables_oracle_702_703.sql` 文件将数据库模式更新到最新级别时，可能会收到以下错误消息：ERROR at line 1: ORA-02289: sequence does not exist。出现该错误的原因是表所使用的序列和触发器与这些表未处于同一模式中。

相关参考

[第 249 页的『MFT 常见问题』](#)

Managed File Transfer 网络中可能出现的常见问题。

[第 251 页的『当 `ftelistagents` 命令未列出您的 MFT 代理时该怎么做』](#)

如果代理未通过 `ftelistagents` 命令列出，或者未显示在 IBM MQ Explorer 中，或者文件传输未显示在 IBM MQ Explorer 的 **传输日志** 中，那么您可以执行许多问题确定步骤来调查原因。

[第 253 页的『在 MFT 代理进程消失但未记录任何诊断信息时该怎么做』](#)

在 UNIX 平台上，如果代理进程已消失，但代理日志文件不包含任何解释，这可能由于启动此代理的方式所致。

[第 262 页的『在您认为文件传输卡住时应该怎么办』](#)

在高负荷的系统上或者在源和目标代理之间发生网络问题时，传输可能会偶尔显示为卡住，进入排队或恢复状态。导致此情况的因素有很多。

[第 262 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 264 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 266 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 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 266 页的『发布前检查消息』](#)

由于代理可连接到 IBM WebSphere MQ 6.0 队列管理器，因此代理不使用 IBM WebSphere MQ 7.0 中引入的直接发布方法。事实上，代理将普通消息发送到包含 MQRFH 头的协调队列管理器。MQRFH 头请求发布消息的有效内容。这些消息会被发送到协调队列管理器上的 SYSTEM.FTE 队列，通常会立即从该队列发布这些消息。如果错误条件阻止此发布，您可以在尝试发布前检查队列上的消息以帮助进行诊断。可通过完成以下步骤来执行此操作：

第 267 页的『传输 IBM i 保存文件时可能发生的错误』

如果您使用 Managed File Transfer 多次传输同一个 IBM i 保存文件，那么传输可能失败。

第 268 页的『用于设置与消息大小关联的 MQ 属性和 MFT 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 Managed File Transfer 属性以影响 Managed File Transfer 的行为。

第 275 页的『MFT 记录器错误处理和拒绝』

Managed File Transfer 记录器可识别两种类型的错误：每条消息的错误和通用错误。

第 276 页的『在 MFT 记录器已启动但未将任何传输信息记录到数据库时该怎么做』

Managed File Transfer 记录器使用的数据库表要求数据库的页面大小为 8KB 或更大。如果数据库的页面大小不足，那么无法正确创建数据库表，并且您将看到 SQLSTATE=42704 错误。

第 394 页的『fteDisplayVersion』

使用 **fteDisplayVersion** 命令来显示您已安装的 Managed File Transfer 的版本。

第 280 页的『BFGSS0023E 错误以及如何避免这些错误』

如果从安装中卸载修订包以返回至产品的先前版本，并在发生卸载时受管传输涉及了与安装相关联的代理，那么该代理无法启动，并将报告 BFGSS0023E 错误。要避免此错误，您可以完成一些步骤以防止在重新启动代理时显示 BFGSS0023E 消息。

第 282 页的『MFT 的返回码』

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。

使用 MFT 的提示与技巧

以下建议可帮助您最有效地使用 Managed File Transfer。

- 如果您更改 `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 Managed File Transfer，尤其是如果您正在运行多个并行传输的情况下。如果代理存在错误，指示其内存不足，那么请根据需要检查并更新以下参数：

– **UNIX** 对于 UNIX 平台：请运行命令：`ulimit -m 1048576`（或者大约 1 GB）。该最大常驻集合大小足以允许最多 25 个并行传输（25 个并行传输是一个代理的缺省最大传输数）。

– 对于所有平台：按如下所示设置 **BFG_JVM_PROPERTIES** 环境变量：`BFG_JVM_PROPERTIES="-Xmx1024M"`

如果要允许并行传输数大于缺省最大值 25，请针对 **ulimit** 和 **BFG_JVM_PROPERTIES** 使用大于建议值的大小。

注：对于 Connect:Direct 网桥代理，最大并行传输数量的缺省值为 5。

- 使用 Managed File Transfer 以文本方式在不同平台间传输文件时，源平台的缺省文件编码可能不受目标平台的支持。这会导致传输失败，并出现以下错误：

```
BFGI00058E: The transfer source encoding xxx is illegal or for an unsupported character set.
```

您可通过使用环境变量将源编码设置为目标平台支持的编码来解决此问题。按如下所示在源系统上设置 **BFG_JVM_PROPERTIES** 系统环境变量：`BFG_JVM_PROPERTIES="-Dfile.encoding=xxx"`，其中 `xxx` 是目标平台支持的编码。例如，如果要以文本方式将文件从 Solaris 平台传输到其他平台，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 [第 364 页的『fteCreateTransfer: 启动新的文件传输』](#)。

相关参考

第 504 页的『MFT 的 Java 系统属性』

必须将一些 Managed File Transfer 命令和代理属性定义为 Java 系统属性，因为它们为无法使用命令或代理属性机制的早期功能定义配置。

在 MFT 上运行跟踪

您可以使用本部分中描述的方法来跟踪 Managed File Transfer。

关于此任务

请参阅第 231 页的『在多平台上跟踪 Managed File Transfer 资源』和第 236 页的『跟踪 Managed File Transfer for z/OS 资源』，以了解有关跟踪各种 Managed File Transfer 资源的详细信息。

fteSetAgentTraceLevel

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:

Multi 在多平台上使用 IBM WebSphere MQ 7.5 或更高版本时，只有运行代理进程的用户才能运行 **fteSetAgentTraceLevel** 命令。

V 9.0.2 **z/OS** 对于 z/OS，针对 Continuous Delivery，从 IBM MQ 9.0.2 开始，**fteSetAgentTraceLevel** 命令可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 9.0.0.1 这也适用于来自 IBM MQ 9.0.0 Fix Pack 1 的 Long Term Support 发行版。

有关更多信息，请参阅第 460 页的『MFT agent.properties 文件』中的 **adminGroup** 属性。

在 IBM WebSphere MQ 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，仅在您的 IBM 服务代表要求时才启用跟踪。



注意:

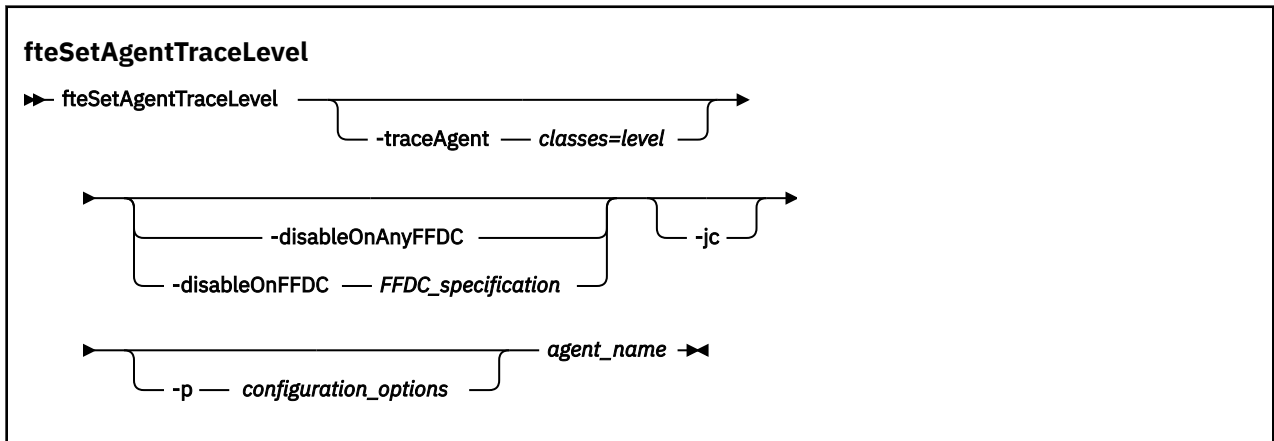
1. 必须在运行代理程序的系统上运行此命令。
2. 跟踪和日志记录不会在代理程序重新启动时持久存储。

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 460 页的『MFT 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。

V 9.0.5 从 IBM MQ 9.0.5 开始，您可以指定冒号分隔的类规范列表。此选项使您能够在不同的级别设置不同类的跟踪。例如：

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.agent=all:com.ibm.wmqfte.cmdhandler=moderate  
AGENT1
```

您仍可以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。请使用以下格式：

```
classes=level
```

例如：

```
com.ibm.wmqfte=all
```

您可以使用 Managed File Transfer 程序包名称来代替 `classes`，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 Managed File Transfer Agent 的名称。

-你说什么 或 -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` 开头的所有类（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。

Multi 在多平台上跟踪 *Managed File Transfer* 资源

利用 *Managed File Transfer* 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。您可以跟踪各种不同的资源。

关于此任务

请参阅

- 第 231 页的『[在多平台上跟踪 *Managed File Transfer* 代理](#)』，以了解有关如何跟踪代理的信息。
- 第 233 页的『[在多平台上跟踪 *Managed File Transfer* 命令](#)』，以了解有关如何跟踪命令的信息。

Multi 在多平台上跟踪 *Managed File Transfer* 代理

利用 *Managed File Transfer* 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。可使用多个命令和属性来控制此功能的行为。

关于此任务

如果要求您提供跟踪输出来调查代理问题，请根据能否在短时间内停止代理，使用以下选项之一。

如果您不确定要使用的选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果可以在短时间内停止代理，请从启动开始收集代理的跟踪信息。
有关更多信息，请参阅第 231 页的『[从启动开始收集 *Managed File Transfer* 代理跟踪信息](#)』。
- 如果无法停止代理，请使用 **fteSetAgentTraceLevel** 命令来动态收集跟踪信息。
有关更多信息，请参阅第 232 页的『[自动收集 *Managed File Transfer* 代理跟踪信息](#)』。

Multi 从启动开始收集 *Managed File Transfer* 代理跟踪信息

如果可以在短时间内停止代理，那么应从启动开始收集 *Managed File Transfer* 代理跟踪信息。

开始之前

您需要在 `agent.properties` 文件中为需要跟踪的代理设置各种属性。

关于此任务

要从启动开始收集跟踪信息，请完成以下步骤。

过程

1. 找到需要跟踪的代理的 `agent.properties` 文件。
可以在 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中找到 `agent.properties` 文件。
2. 编辑文件并为以下属性添加条目：
 - **trace**=跟踪规范

trace 属性将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

- **traceFiles**=要使用的跟踪文件数
- **traceSize**=每个跟踪文件的大小 (MB)

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要使用跟踪规范 `com.ibm.wmqfte=all` 收集 1GB 的打包跟踪，请将以下行添加到 `agent.properties` 文件中：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

这会导致代理将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些代理属性的更多信息，请参阅 [MFT agent.properties](#) 文件。

3. 使用 **fteStopAgent** 命令来停止需要跟踪的代理。
4. 通过运行 **fteStartAgent** 命令来启动代理。
5. 重现该问题。
6. 停止代理。
7. 编辑代理的 `agent.properties` 文件，并除去在步骤 [第 231 页的『2』](#) 中添加的 **trace**、**traceFiles** 和 **traceSize** 属性的条目。

这样可确保下次重新启动代理时不会启用跟踪。

结果

生成的跟踪文件将写入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目录，其中 `%PID%` 是代理的进程标识。

Multi

自动收集 *Managed File Transfer* 代理跟踪信息

使用 **fteSetAgentTraceLevel** 命令，可以从正在运行的代理中收集跟踪信息。如果 IBM 支持人员需要查看无法停止的代理的跟踪信息，这会非常有用。

关于此任务

要使用 **fteSetAgentTraceLevel** 命令从代理程序收集跟踪，请完成以下步骤。

过程

1. 通过运行以下命令来对代理开启跟踪：

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

`-traceAgent` 参数将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

2. 重现该问题。
3. 通过运行以下命令来关闭代理程序的跟踪：

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

如果代理程序繁忙，那么跟踪文件可能会快速合并并覆盖调查问题所需的信息。如果是这样，请安排一些时间来停止代理程序，然后按照以下步骤中的详细信息继续操作。如果无法在短时间内停止代理程序，请与 IBM 支持代表联系并讨论要使用的替代跟踪规范，以减少正在生成的跟踪数据量。

4. 找到需要跟踪的代理的 `agent.properties` 文件。

可以在 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中找到 `agent.properties` 文件。

5. 编辑文件并为以下属性添加条目：

```
traceFiles=number_of_trace_files_to_use
traceSize=size_of_each_trace_file_in_MB
```

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。

traceFiles 属性的缺省值为 5，**traceSize** 属性的缺省值为 20MB。这意味着，如果您动态开启跟踪并且尚未设置属性，那么代理会将跟踪信息写入到 5 个合并跟踪文件，每个文件的最大大小为 20MB。

您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要收集 1GB 的合并跟踪，请将以下行添加到 `agent.properties` 文件中：

```
traceFiles=5
traceSize=200
```

这会导致代理将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些代理属性的更多信息，请参阅 `MFT agent.properties` 文件。

6. 通过运行 **fteStopAgent** 命令来停止代理程序。

7. 通过运行 **fteStartAgent** 命令来启动代理。

8. 通过运行以下命令，对代理程序启用跟踪：

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

除非您的 IBM 支持代表另有指定，否则请将 **-traceAgent** 属性设置为值 `com.ibm.wmqfte=all`。

9. 重现该问题。

10. 通过运行以下命令在代理程序上关闭跟踪：

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

结果

生成的跟踪文件将写入到 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目录中，其中 `%PID%` 是代理的进程标识。

Multi 在多平台上跟踪 *Managed File Transfer* 命令

利用 *Managed File Transfer* 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。您可以使用此功能来跟踪命令。

关于此任务



注意：跟踪命令将仅收集命令所执行的处理的相关信息。它不会跟踪在处理该命令时代理可能执行的任何活动。

过程

1. 打开命令提示符，浏览至 `MQ_INSTALLATION_PATH\bin` 目录。

2. 运行以下命令：

Linux UNIX

```
./command_name -trace classes=level -tracePath directory_path command_arguments
```

Windows

```
command_name -trace classes=level -tracePath directory_path command_arguments
```

其中：

- `command_name` 是要跟踪的命令的名称。
- `classes=level` 是要使用的跟踪级别，以及要启用跟踪的类别。除非 IBM 支持代表另有指定，否则请将其设置为 `com.ibm.wmqfte=all`。
- `directory_path` 是跟踪文件将写入的目录。
- `command_arguments` 是需要传递给命令的自变量，例如，用于 **ftePingAgent** 命令的代理的名称。

结果

生成的跟踪文件将写入到 **-tracePath** 参数指定的目录。

跟踪文件被命名为 `trace%PID%.txt.number`，其中：

- `%PID%` 是命令的进程标识。
- `number` 是跟踪文件的序号。通常，命令生成的跟踪信息包含在序号为 0 的单个跟踪文件中。

但是，命令可能会生成大量跟踪信息。在这种情况下，会将跟踪信息写入到多个文件中。当前跟踪文件的序号为 0，下一个最旧的跟踪文件的序号为 1，依此类推。

命令的跟踪输出最多写入到 5 个合并跟踪文件中。每个跟踪文件的最大大小为 20MB。

注：如果运行命令的用户无权写入到 **-tracePath** 参数指定的目录，那么跟踪输出将写入到标准错误。

示例

在此示例中，将跟踪 **fteListAgents** 命令，并将跟踪信息写入到 `C:\trace` 目录中：

```
fteListAgents -trace com.ibm.wmqfte=all -tracePath C:\trace
```

在此示例中，将跟踪 **fteCreateTransfer** 命令，并将跟踪信息写入到 `/tmp` 目录中：

```
fteCreateTransfer -trace com.ibm.wmqfte=all -tracePath /tmp -t text -sa AGENT1  
-da AGENT2 -df /import/transferredfile.txt /export/originalfile.txt
```

写入到 `/tmp` 的跟踪文件仅包含 **fteCreateTransfer** 命令所执行的处理的相关信息，例如，该命令如何构建要发送到代理的传输请求消息，以及它等待多久后代理会发送回表示已收到该请求的确认。该跟踪文件不包含有关传输本身的任何信息。

Multi 在多平台上跟踪 *Managed File Transfer* 独立记录器

利用 *Managed File Transfer* 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。可使用多个命令和属性来控制此功能的行为。

关于此任务

如果要求您提供跟踪输出来调查记录器问题，请根据能否在短时间内停止记录器，使用以下选项之一。

如果您不确定要使用的选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果可以在短时间内停止记录器，请从启动开始收集记录器的跟踪信息。
请参阅第 234 页的『从启动开始收集 *Managed File Transfer* 独立记录器跟踪信息』。
- 如果无法停止记录器，请使用 **fteSetLoggerTraceLevel** 命令动态收集跟踪信息。
请参阅第 235 页的『动态收集 *Managed File Transfer* 独立记录器跟踪信息』。

Multi 从启动开始收集 *Managed File Transfer* 独立记录器跟踪信息

如果可以在短时间内停止记录器，那么应从启动开始收集 *Managed File Transfer* 记录器跟踪信息。

开始之前

您必须在需要跟踪的记录器的 `logger.properties` 文件中设置各种属性。

关于此任务

要从启动开始收集跟踪信息，请完成以下步骤。

过程

1. 找到需要跟踪的记录器的 `logger.properties` 文件。
可以在 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中找到 `logger.properties` 文件。
2. 编辑文件并为以下属性添加条目：

- **trace**=跟踪规范
trace 属性将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。
- **traceFiles**=要使用的跟踪文件数
- **traceSize**=每个跟踪文件的大小 (MB)
traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要使用跟踪规范 `com.ibm.wmqfte=all` 来收集 1GB 的合并跟踪，请将以下行添加到 `logger.properties` 文件中：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

这会导致记录器将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些记录器属性的更多信息，请参阅 [MFT `logger.properties` 文件](#)。

3. 使用 **`fteStopLogger`** 命令停止需要跟踪的记录器。
4. 通过运行 **`fteStartLogger`** 命令来启动记录器。
5. 重现该问题。
6. 停止记录器。
7. 编辑记录器的 `logger.properties` 文件，并除去您在步骤 [第 235 页的『2』](#) 中添加的 **`trace`**、**`traceFiles`** 和 **`traceSize`** 属性的条目。
这样可确保下次重新启动记录器时不会启用跟踪。

结果

生成的跟踪文件将写入到 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目录中，其中 `%PID%` 是记录器的进程标识。

Multi

动态收集 *Managed File Transfer* 独立记录器跟踪信息

使用 **`fteSetLoggerTraceLevel`** 命令，可以从正在运行的记录器收集跟踪信息。如果 IBM 支持人员需要查看无法停止的记录器的跟踪信息，这会非常有用。

关于此任务

要使用 **`fteSetLoggerTraceLevel`** 命令从记录器收集跟踪，请完成以下步骤。

过程

1. 通过运行以下命令来对记录器开启跟踪：

```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

-traceLogger 参数将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

2. 重现该问题。
3. 使用以下命令来关闭跟踪：

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

4. 如果记录器繁忙，那么跟踪文件可能会迅速合并和覆盖调查此问题所需的信息。

如果可以在短时间内停止记录器，请完成以下步骤以减少收集的跟踪数据量。否则，请联系 IBM 支持人员并讨论替代跟踪规范，以减少收集的跟踪数据量。

- a) 安排一些时间来停止记录器。
- b) 找到需要跟踪的记录器的 `logger.properties` 文件。

可以在 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中找到 `logger.properties` 文件。

- c) 编辑文件并为以下属性添加条目：

```
traceFiles=number_of_trace_files_to_use
```

```
traceSize=size_of_each_trace_file_in_MB
```

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。

traceFiles 属性的缺省值为 5，**traceSize** 属性的缺省值为 20MB。这意味着，如果您动态开启跟踪并且尚未设置属性，那么代理会将跟踪信息写入到 5 个合并跟踪文件，每个文件的最大大小为 20MB。

您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要收集 1GB 的合并跟踪，请将以下行添加到 `logger.properties` 文件中：

```
traceFiles=5  
traceSize=200
```

这会导致记录器将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些记录器属性的更多信息，请参阅 [MFT logger.properties](#) 文件。

- d) 通过运行 **fteStopLogger** 命令来停止记录器。
- e) 通过运行 **fteStartLogger** 命令来启动记录器。
- f) 通过运行以下命令，打开记录器的跟踪。除非您的 IBM 支持代表另有指定，否则请将 **-traceLogger** 属性设置为值 `com.ibm.wmqfte=all`。

```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

- g) 重现该问题。
- h) 通过运行以下命令，关闭记录器的跟踪：

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

结果

生成的跟踪文件将写入到 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目录中，其中 `%PID%` 是记录器的进程标识。

跟踪 Managed File Transfer for z/OS 资源

利用 Managed File Transfer for z/OS 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。您可以跟踪各种不同的资源。

关于此任务

请参阅

- 第 237 页的『跟踪 Managed File Transfer for z/OS 代理』，以了解有关如何跟踪代理的信息。
- 第 242 页的『跟踪 Managed File Transfer for z/OS 命令』，以了解有关如何跟踪命令的信息。

z/OS 跟踪 Managed File Transfer for z/OS 代理

利用 Managed File Transfer for z/OS 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。可使用多个命令和属性来控制此功能的行为。

关于此任务

如果要求您提供跟踪输出来调查代理问题，请使用以下选项之一。

如果您不确定要使用的选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果可以在短时间内停止代理，请从启动开始收集代理的跟踪信息。
有关更多信息，请参阅第 237 页的『从启动开始收集 Managed File Transfer for z/OS 代理跟踪信息』。
- 如果无法停止代理，请使用 **fteSetAgentTraceLevel** 命令来动态收集跟踪信息。
有关更多信息，请参阅第 239 页的『动态收集 Managed File Transfer for z/OS 代理跟踪信息』。

z/OS 从启动开始收集 Managed File Transfer for z/OS 代理跟踪信息

如果可以在短时间内停止代理，那么应从启动开始收集 IBM MQ Managed File Transfer 代理跟踪信息。

关于此任务

收集跟踪信息的方法取决于是使用 Unix 系统服务 (USS) 还是 JCL 来管理代理。

如果您不确定要使用以下哪个选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果您正在使用 z/OS UNIX，请参阅第 237 页的『从启动开始使用 USS 收集代理跟踪信息』。
- 如果您正在使用 JCL，请参阅第 238 页的『从启动开始使用 JCL 收集代理跟踪信息』。

z/OS 从启动开始使用 USS 收集代理跟踪信息

要从启动开始收集使用 UNIX 系统服务 (USS) 管理的 Managed File Transfer for z/OS 代理的跟踪信息，您必须先在该代理的 `agent.properties` 文件中设置各种必需属性，然后再启动该代理。

关于此任务

要从启动开始收集跟踪信息，请完成以下步骤。

过程

1. 找到需要跟踪的代理的 `agent.properties` 文件。
可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中找到 `agent.properties` 文件。
2. 编辑文件并为以下属性添加条目：
 - **trace**=跟踪规范

trace 属性将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

- **traceFiles**=要使用的跟踪文件数
- **traceSize**=每个跟踪文件的大小 (MB)

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要使用跟踪规范 `com.ibm.wmqfte=all` 来收集 1GB 的合并跟踪，请将以下行添加到 `agent.properties` 文件中：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```


这会导致代理将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些代理属性的更多信息，请参阅 [MFT agent.properties](#) 文件。

3. 使用 **fteStopAgent** 命令停止需要跟踪的代理程序。
4. 通过运行 **fteStartAgent** 命令来启动代理。
5. 重现该问题。
6. 停止代理。
7. 编辑代理程序的 `agent.properties` 文件，并除去您在步骤 [第 237 页的『2』](#) 中添加的 **trace**，**traceFiles** 和 **traceSize** 属性的条目。
这样可确保下次重新启动代理时不会启用跟踪。

结果

生成的跟踪文件将写入到 `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目录中，其中 `%PID%` 是代理的进程标识。

 从启动开始使用 JCL 收集代理跟踪信息

要从启动开始收集使用 JCL 管理的 Managed File Transfer for z/OS 代理的跟踪信息，就需要在该代理启动之前在 `agent.properties` 文件中为其设置所需的各种属性。

关于此任务

要从启动开始收集跟踪信息，请完成以下步骤。

过程

1. 找到需要跟踪的代理的 `agent.properties` 文件。
可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中找到 `agent.properties` 文件。
2. 编辑文件并为以下属性添加条目：
 - **trace**=跟踪规范
trace 属性将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。
 - **traceFiles**=要使用的跟踪文件数
 - **traceSize**=每个跟踪文件的大小 (MB)
traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要使用跟踪规范 `com.ibm.wmqfte=all` 收集 1GB 的打包跟踪，请将以下行添加到 `agent.properties` 文件中：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

这会导致代理将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些代理属性的更多信息，请参阅 [MFT agent.properties](#) 文件。

3. 找到包含需要跟踪的代理的 JCL 的数据集。
4. 提交数据集中的 BFGAGSP 成员以停止代理。
5. 通过提交数据集中的 BFGAGST 成员来重新启动代理。
6. 重现该问题。
7. 提交数据集中的 BFGAGSP 成员以再次停止代理。
8. 编辑代理的 `agent.properties` 文件，并除去在步骤 [第 238 页](#) 的『2』中添加的 **trace**、**traceFiles** 和 **traceSize** 属性的条目。

这样可确保下次重新启动代理时不会启用跟踪。

结果

生成的跟踪文件将写入 `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目录，其中 `%PID%` 是代理的进程标识。

z/OS 动态收集 *Managed File Transfer for z/OS* 代理跟踪信息

如果无法在短时间内停止代理，那么应动态收集 *Managed File Transfer for z/OS* 代理跟踪信息。

关于此任务

收集跟踪信息的方法取决于是使用 Unix 系统服务 (USS) 还是 JCL 来管理代理。

如果您不确定要使用以下哪个选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果您使用：
 - Unix 系统服务 (USS)，请参阅 [第 239 页](#) 的『使用 USS 动态收集代理跟踪信息』。
 - JCL，请参阅 [第 240 页](#) 的『使用 JCL 动态收集代理跟踪信息』。

z/OS 使用 USS 动态收集代理跟踪信息

使用 `fteSetAgentTraceLevel` 命令，可以从正在运行的代理中收集跟踪信息。如果 IBM 支持人员需要查看无法停止的代理的跟踪信息，这会非常有用。

关于此任务

要使用 `fteSetAgentTraceLevel` 命令从 *Managed File Transfer for z/OS* 代理程序收集跟踪，请完成以下步骤。

过程

1. 通过运行以下命令，打开代理程序的跟踪：

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

`-traceAgent` 参数将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

2. 重现该问题。
3. 通过运行以下命令，关闭代理程序的跟踪：

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

4. 如果代理程序繁忙，那么跟踪文件可能会快速合并并覆盖调查问题所需的信息。如果可以在短时间内停止代理程序，请完成以下步骤以减少收集的跟踪数据量。否则，请联系 IBM 支持人员并讨论替代跟踪规范，以减少收集的跟踪数据量。

- a) 安排一些时间来停止代理。
- b) 找到需要跟踪的代理的 `agent.properties` 文件。

可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中找到 `agent.properties` 文件。

- c) 编辑文件并为以下属性添加条目：

```
traceFiles=number_of_trace_files_to_use
traceSize=size_of_each_trace_file_in_MB
```

`traceFiles` 和 `traceSize` 属性用于控制要收集的跟踪数据量。

`traceFiles` 属性的缺省值为 5，`traceSize` 属性的缺省值为 20MB。这意味着，如果您动态开启跟踪并且尚未设置属性，那么代理会将跟踪信息写入到 5 个合并跟踪文件，每个文件的最大大小为 20MB。

您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要收集 1GB 的合并跟踪，请将以下行添加到 `agent.properties` 文件中：

```
traceFiles=5
traceSize=200
```

这会导致代理将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些代理属性的更多信息，请参阅 [MFT agent.properties](#) 文件。

- d) 通过运行 `fteStopAgent` 命令来停止代理程序。
- e) 通过运行 `fteStartAgent` 命令来启动代理。
- f) 通过运行以下命令，打开代理程序的跟踪：

```
fteSetAgentTraceLevel -traceAgent trace_specification agent_name
```

- g) 重现该问题。
- h) 通过运行以下命令，关闭代理程序的跟踪：

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

结果

生成的跟踪文件将写入到 `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目录中，其中 `%PID%` 是代理的进程标识。

使用 JCL 动态收集代理跟踪信息

对于需要跟踪的代理，您可以使用包含 JCL 的数据集中的 BFGAGTC 成员，从正在运行的 Managed File Transfer for z/OS 代理中收集跟踪信息。如果 IBM 支持人员需要查看无法停止的代理的跟踪信息，这会非常有用。

关于此任务

要使用 BFGAGTC 成员从代理中收集跟踪信息，请完成以下步骤。

过程

1. 找到包含需要跟踪的代理的 JCL 的数据集。
2. 编辑数据集中的 BFGAGTC 成员，然后找到包含以下文本的行：

```
-traceAgent
```

其下方的文本包含要跟踪的内部类和包的列表。缺省情况下，将此列表设置为：

```
com.ibm.wmqfte=all
```

除非您的 IBM 支持代表另有指定，否则请按原样保留此值。

3. 提交 BFGAGTC 成员。
4. 重现该问题。
5. 再次编辑 BFGAGTC 成员，然后将 **-traceAgent** 参数设置为 `=off`，如下所示：

```
-traceAgent =off +
```

6. 重新提交 BFGAGTC 成员以关闭跟踪。
7. 如果代理繁忙，那么跟踪文件可能会迅速合并和覆盖调查此问题所需的信息。
在此情况下，可以使用以下两个选项：

a) 第一个选项是：

i) 安排一些时间来停止代理。

ii) 找到需要跟踪的代理的 `agent.properties` 文件。可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中找到 `agent.properties` 文件。

iii) 编辑文件并为以下属性添加条目：

- **traceFiles**=要使用的跟踪文件数
- **traceSize**=每个跟踪文件的大小 (MB)

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。

traceFiles 属性的缺省值为 5，**traceSize** 属性的缺省值为 20MB。这意味着，如果您动态开启跟踪并且尚未设置属性，那么代理会将跟踪信息写入到 5 个合并跟踪文件，每个文件的最大大小为 20MB。

您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要收集 1GB 的合并跟踪，请将以下行添加到 `agent.properties` 文件中：

```
traceFiles=5  
traceSize=200
```

这会导致代理将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些代理属性的更多信息，请参阅 [MFT agent.properties](#) 文件。

- iv) 找到包含需要跟踪的代理的 JCL 的数据集。
- v) 提交数据集中的 BFGAGSP 成员以停止代理。
- vi) 通过提交数据集中的 BFGAGST 成员来重新启动代理。
- vii) 编辑数据集中的 BFGAGTC 成员，然后找到包含以下文本的行：

```
-traceAgent
```

其下方的文本包含要跟踪的内部类和包的列表。缺省情况下，将此列表设置为：

```
com.ibm.wmqfte=all
```

除非您的 IBM 支持代表另有指定，否则请按原样保留此值。

- viii) 当需要启用跟踪时，请提交 BFGAGTC 成员。
- ix) 重现该问题。
- x) 再次编辑 BFGAGTC 成员，然后将 **-traceAgent** 参数设置为 *=off*，如下所示：

```
-traceAgent =off +
```

- xi) 重新提交 BFGAGTC 成员以关闭跟踪。
- b) 第二个选项是联系您的 IBM 支持代表（如果无法在短时间内停止代理）。然后，您可以讨论要使用的备用跟踪规范，以减少要生成的跟踪数据量。

结果

生成的跟踪文件将写入到 *BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%* 目录中，其中 %PID% 是代理的进程标识。

跟踪 Managed File Transfer for z/OS 命令

利用 Managed File Transfer for z/OS 中的跟踪功能，可以帮助 IBM 支持人员诊断问题。您可以使用此功能来跟踪命令。

关于此任务



注意：跟踪命令将仅收集命令所执行的处理的相关信息。它不会跟踪在处理该命令时代理可能执行的任何活动。

收集跟踪信息的方法取决于使用 Unix 系统服务 (USS) 还是 JCL 来运行命令。

如果您不确定要使用的选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果您正在使用 z/OS UNIX，请参阅 [第 237 页的『从启动开始使用 USS 收集代理跟踪信息』](#)。
- 如果您正在使用 JCL，请参阅 [第 238 页的『从启动开始使用 JCL 收集代理跟踪信息』](#)。

使用 USS 收集命令的跟踪信息

要使用 Unix 系统服务 (USS) 收集 Managed File Transfer for z/OS 命令的跟踪信息，请执行以下过程。

过程

1. 打开命令提示符，浏览至 *BFG_PROD/bin* 目录。
2. 运行以下命令：

```
./command_name -trace classes=level -tracePath directory_path command_arguments
```

其中，

- *command_name* 是要跟踪的命令的名称。
- *classes=level* 是要使用的跟踪级别，以及要启用跟踪的类别。除非 IBM 支持代表另有指定，否则请将其设置为 *com.ibm.wmqfte=all*。
- *directory_path* 是跟踪文件将写入的目录。
- *command_arguments* 是需要传递给命令的自变量，例如，用于 **ftePingAgent** 命令的代理的名称。

结果

生成的跟踪文件将写入到 **-tracePath** 参数指定的目录。

跟踪文件被命名为 *trace%PID%.txt.number*，其中：

- `%PID%` 是命令的进程标识。
 - `number` 是跟踪文件的序号。通常，命令生成的跟踪信息包含在序号为 0 的单个跟踪文件中。
- 但是，命令可能会生成大量跟踪信息。在这种情况下，会将跟踪信息写入到多个文件中。当前跟踪文件的序号为 0，下一个最旧的跟踪文件的序号为 1，依此类推。

命令的跟踪输出最多写入到 5 个合并跟踪文件中。每个跟踪文件的最大大小为 20MB。

注: 如果运行命令的用户无权写入到 `-tracePath` 参数指定的目录，那么跟踪输出将写入到标准错误。

示例

在此示例中，将跟踪 `fteListAgents` 命令，并将跟踪信息写入到 `/u/fteuser` 目录中：

```
./fteListAgents -trace com.ibm.wmqfte=all -tracePath /u/fteuser
```

在此示例中，将跟踪 `fteCreateTransfer` 命令，并将跟踪信息写入到 `/tmp` 目录中：

```
./fteCreateTransfer -trace com.ibm.wmqfte=all -tracePath /tmp -t text -sa AGENT1
-da AGENT2 -df /tmp/IEEUJV.txt "'SYS1.SAMPLIB(IEEUJV)'"
```

写入到 `/tmp` 的跟踪文件仅包含 `fteCreateTransfer` 命令所执行的处理的相关信息，例如，该命令如何构建要发送到代理的传输请求消息，以及它等待多久后代理会发送回表示已收到该请求的确认。该跟踪文件不包含有关传输本身的任何信息。

z/OS

使用 JCL 收集命令的跟踪信息

要收集使用 JCL 提交的 Managed File Transfer for z/OS 命令的跟踪信息，需要完成以下步骤。

过程

1. 找到包含需要跟踪的命令的 JCL 的数据集。
2. 在该数据集中，找到该命令的成员。
3. 编辑该成员，并找到包含需要跟踪的命令名称的行。修改此行，使其包含命令名之后和 + 号之前的文本：

```
-trace classes=level -tracePath directory_path
```

其中：

- `classes=level` 是要使用的跟踪级别，以及要启用跟踪的类别。除非 IBM 支持代表另有指定，否则请将其设置为 `com.ibm.wmqfte=all`。
- `directory_path` 是要将跟踪文件写入到的 USS 目录。

4. 提交该成员。
5. 重现该问题后，再次编辑此成员，并去掉文本：

```
-trace classes=level -tracePath directory_path
```

这是您在步骤 [第 243 页的『3』](#) 中添加的内容。

结果

生成的跟踪文件将写入到 `-tracePath` 参数指定的目录。

跟踪文件被命名为 `trace%PID%.txt.number`，其中：

- `%PID%` 是命令的进程标识。
- `number` 是跟踪文件的序号。通常，命令生成的跟踪信息包含在序号为 0 的单个跟踪文件中。

但是，命令可能会生成大量跟踪信息。在这种情况下，会将跟踪信息写入到多个文件中。当前跟踪文件的序号为 0，下一个最旧的跟踪文件的序号为 1，依此类推。

命令的跟踪输出最多写入到 5 个合并跟踪文件中。每个跟踪文件的最大大小为 20MB。

注: 如果运行命令的用户无权写入到 **-tracePath** 参数指定的目录, 那么跟踪输出将写入到标准错误。

示例

在此示例中, 已修改成员 BFGMNL1 以跟踪 **fteListMonitors** 命令:

```
/******  
/* <copyright  
/* notice="lm-source"  
/* pids="5655-MF9"  
/* years="2013,2016"  
/* crc="3927276320" >  
/* Licensed Materials - Property of IBM  
/*  
/* 5655-MF9  
/*  
/* (C) Copyright IBM Corp. 2013, 2023. All Rights Reserved.  
/* </copyright>  
/******  
/* fteListMonitors  
/******  
//BFGCMD EXEC PGM=IKJEFT01,REGION=0M  
//SYSEXEC DD DSN=++LIBRARY++,DISP=SHR  
//SYSTSPRT DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//STDERR DD SYSOUT=*  
//SYSTSIN DD *  
%BFGCMD CMD=fteListMonitors -trace com.ibm.wmqfte=all -tracePath /u/trace +  
-v -p QM1  
/*  
//
```

提交该成员后, **fteListMonitors** 命令会将跟踪信息写入到 USS 目录 /u/trace。

z/OS 跟踪 Managed File Transfer for z/OS 独立数据库记录器

利用 Managed File Transfer for z/OS 中的跟踪功能, 可以帮助 IBM 支持人员诊断问题。可使用多个命令和属性来控制此功能的行为。

关于此任务

如果要求您提供跟踪输出来调查独立数据库记录器问题, 请使用以下选项之一。

如果您不确定要使用的选项, 请联系 IBM 支持代表, 他们将提出最佳方式建议, 帮助您为所发现的问题收集跟踪信息。

过程

- 如果可以在短时间内停止记录器, 请从启动开始收集记录器的跟踪信息。
有关更多信息, 请参阅第 244 页的『从启动开始收集 Managed File Transfer for z/OS 独立数据库记录器跟踪信息』。
- 如果无法停止记录器, 请使用 **fteSetLoggerTraceLevel** 命令动态收集跟踪信息。
有关更多信息, 请参阅第 246 页的『动态收集 Managed File Transfer for z/OS 独立数据库记录器跟踪信息』。

z/OS 从启动开始收集 Managed File Transfer for z/OS 独立数据库记录器跟踪信息

如果可以在短时间内停止记录器, 那么应从启动开始收集 IBM MQ Managed File Transfer 记录器跟踪信息。

关于此任务

收集跟踪信息的方法取决于使用 UNIX 系统服务 (USS) 还是 JCL 来管理记录器。

如果您不确定要使用以下哪个选项, 请联系 IBM 支持代表, 他们将提出最佳方式建议, 帮助您为所发现的问题收集跟踪信息。

过程

- 如果您使用：
 - UNIX 系统服务 (USS)，请参阅第 245 页的『使用 USS 从启动开始收集独立数据库记录器跟踪信息』。
 - JCL，请参阅第 245 页的『使用 JCL 从启动开始收集独立数据库记录器跟踪信息』。

z/OS 使用 USS 从启动开始收集独立数据库记录器跟踪信息
要从启动开始收集使用 UNIX 系统服务 (USS) 管理的 Managed File Transfer for z/OS 记录器的跟踪信息，需要在此记录器的 `logger.properties` 文件中设置各种属性，然后再启动记录器。

关于此任务

要从启动开始收集跟踪信息，请完成以下步骤。

过程

1. 找到需要跟踪的记录器的 `logger.properties` 文件。
可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中找到 `logger.properties` 文件。
2. 编辑文件并为以下属性添加条目：
 - **trace**=跟踪规范
trace 属性将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。
 - **traceFiles**=要使用的跟踪文件数
 - **traceSize**=每个跟踪文件的大小 (MB)
traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要使用跟踪规范 `com.ibm.wmqfte=all` 来收集 1GB 的合并跟踪，请将以下行添加到 `logger.properties` 文件中：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

这会导致记录器将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些记录器属性的更多信息，请参阅 `MFT logger.properties` 文件。

3. 使用 **fteStopLogger** 命令停止需要跟踪的记录器。
4. 通过运行 **fteStartLogger** 命令来启动记录器。
5. 重现该问题。
6. 停止记录器。
7. 编辑记录器的 `logger.properties` 文件，并除去您在步骤 第 245 页的『2』中添加的 **trace**，**traceFiles** 和 **traceSize** 属性的条目。
这样可确保下次重新启动记录器时不会启用跟踪。

结果

生成的跟踪文件将写入到 `BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目录，其中 `%PID%` 是记录器的进程标识。

z/OS 使用 JCL 从启动开始收集独立数据库记录器跟踪信息
要从启动开始收集使用 JCL 管理的 Managed File Transfer for z/OS 记录器的跟踪信息，您需要先在此记录器的 `logger.properties` 文件中设置各种属性，然后再启动该记录器。

关于此任务

要从启动开始收集跟踪信息，请完成以下步骤。

过程

1. 找到需要跟踪的记录器的 `logger.properties` 文件。

可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中找到 `logger.properties` 文件。

2. 编辑文件并为以下属性添加条目：

- **trace**=跟踪规范

trace 属性将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定，否则请将此属性设置为值 `com.ibm.wmqfte=all`。

- **traceFiles**=要使用的跟踪文件数

- **traceSize**=每个跟踪文件的大小 (MB)

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要使用跟踪规范 `com.ibm.wmqfte=all` 来收集 1GB 的合并跟踪，请将以下行添加到 `logger.properties` 文件中：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

这会导致记录器将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些记录器属性的更多信息，请参阅 `MFT logger.properties` 文件。

3. 找到包含需要跟踪的记录器的 JCL 的数据集。
4. 提交数据集中的 BFGLGSP 成员以停止记录器。
5. 通过提交数据集中的 BFGLGST 成员来重新启动记录器。
6. 重现该问题。
7. 提交数据集中的 BFGLGSP 成员以再次停止记录器。
8. 编辑记录器的 `logger.properties` 文件，并除去您在步骤 [第 246 页的『2』](#) 中添加的 **trace**，**traceFiles** 和 **traceSize** 属性的条目。
这样可确保下次重新启动记录器时不会启用跟踪。

结果

生成的跟踪文件将写入到 `BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目录，其中 `%PID%` 是记录器的进程标识。

 动态收集 *Managed File Transfer for z/OS* 独立数据库记录器跟踪信息
如果无法在短时间内停止记录器，那么应动态收集 *Managed File Transfer for z/OS* 记录器跟踪信息。

关于此任务

收集跟踪信息的方法取决于是使用 UNIX 系统服务 (USS) 还是 JCL 来管理记录器。

如果您不确定要使用以下哪个选项，请联系 IBM 支持代表，他们将提出最佳方式建议，帮助您为所发现的问题收集跟踪信息。

过程

- 如果您使用：
 - Unix 系统服务 (USS)，请参阅 [第 247 页的『使用 USS 动态收集独立数据库记录器跟踪信息』](#)。

- JCL, 请参阅第 248 页的『使用 JCL 动态收集独立数据库记录器跟踪信息』。

z/OS 使用 USS 动态收集独立数据库记录器跟踪信息

在 (USS) 下, 可以使用 **fteSetLoggerTraceLevel** 命令从正在运行的记录器收集跟踪。如果 IBM 支持人员需要查看无法停止的记录器的跟踪信息, 这会非常有用。

关于此任务

要使用 **fteSetLoggerTraceLevel** 命令从 Managed File Transfer for z/OS 记录器收集跟踪, 请完成以下步骤。

过程

1. 通过运行以下命令, 打开记录器的跟踪:

```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

-traceLogger 参数将确定要跟踪的内部类和包。除非您的 IBM 支持代表另有指定, 否则请将此属性设置为值 `com.ibm.wmqfte=all`。

2. 重现该问题。
3. 通过运行以下命令, 关闭记录器的跟踪:

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

4. 如果记录器繁忙, 那么跟踪文件可能会快速合并并覆盖调查该问题所需的信息。

如果可以在短时间内停止记录器, 请完成以下步骤以减少收集的跟踪数据量。否则, 请联系 IBM 支持人员并讨论替代跟踪规范, 以减少收集的跟踪数据量。

- a) 安排一些时间来停止记录器。
- b) 找到需要跟踪的记录器的 `logger.properties` 文件。

可以在 `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` 目录中找到 `logger.properties` 文件。

- c) 编辑文件并为以下属性添加条目:

```
traceFiles=number_of_trace_files_to_use  
traceSize=size_of_each_trace_file_in_MB
```

traceFiles 和 **traceSize** 属性用于控制要收集的跟踪数据量。

traceFiles 属性的缺省值为 5, **traceSize** 属性的缺省值为 20MB。这意味着, 如果您动态开启跟踪并且尚未设置属性, 那么代理会将跟踪信息写入到 5 个合并跟踪文件, 每个文件的最大大小为 20MB。

您应将这些属性设置为较大的值, 以尽可能多地收集跟踪数据。

例如, 要收集 1GB 的合并跟踪, 请将以下行添加到 `logger.properties` 文件中:

```
traceFiles=5  
traceSize=200
```

这会导致记录器将跟踪数据写入到最多 5 个文件中, 每个文件的大小都为 200MB。

有关这些记录器属性的更多信息, 请参阅 [MFT logger.properties](#) 文件。

- d) 通过运行 **fteStopLogger** 命令来停止记录器。
- e) 通过运行 **fteStartLogger** 命令来启动记录器。
- f) 通过运行以下命令, 打开记录器的跟踪:

```
fteSetLoggerTraceLevel -traceLogger trace_specification logger_name
```

- g) 重现该问题。
- h) 通过运行以下命令，关闭记录器的跟踪：

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

结果

跟踪文件将写入 *BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%* 目录，其中 %PID% 是记录器的进程标识。

z/OS 使用 JCL 动态收集独立数据库记录器跟踪信息
对于需要跟踪的记录器，您可以使用包含 JCL 的数据集中的 BFGLGTC 成员，从正在运行的 Managed File Transfer for z/OS 记录器收集跟踪信息。如果 IBM 支持人员需要查看无法停止的记录器的跟踪信息，这会非常有用。

关于此任务

要使用 BFGLGTC 成员从记录器收集跟踪信息，请完成以下步骤。

过程

1. 找到包含需要跟踪的记录器的 JCL 的数据集。
2. 编辑数据集中的 BFGLGTC 成员，然后找到包含以下文本的行：

```
-traceLogger
```

其下方的文本包含要跟踪的内部类和包的列表。缺省情况下，将此列表设置为：

```
com.ibm.wmqfte=all
```

除非您的 IBM 支持代表另有指定，否则请按原样保留此值。

3. 提交 BFGLGTC 成员。
4. 重现该问题。
5. 再次编辑 BFGLGTC 成员，并将 **-traceLogger** 参数设置为 *=off*，如下所示：

```
-traceLogger =off +
```

6. 重新提交 BFGLGTC 成员以关闭跟踪。
7. 如果记录器繁忙，那么跟踪文件可能会迅速合并和覆盖调查此问题所需的信息。
在此情况下，可以使用以下两个选项：

- a) 第一个选项是：
 - i) 安排一些时间来停止记录器。
 - ii) 找到需要跟踪的记录器的 *logger.properties* 文件。可以在 *BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name* 目录中找到 *logger.properties* 文件。
 - iii) 编辑文件并为以下属性添加条目：
 - **traceFiles**=要使用的跟踪文件数
 - **traceSize**=每个跟踪文件的大小 (MB)**traceFiles** 和 **traceSize** 属性用于控制要收集的跟踪数据量。

traceFiles 属性的缺省值为 5，**traceSize** 属性的缺省值为 20MB。这意味着，如果您动态开启跟踪并且尚未设置属性，那么记录器会将跟踪信息写入到 5 个合并跟踪文件，每个文件的最大大小为 20MB。

您应将这些属性设置为较大的值，以尽可能多地收集跟踪数据。

例如，要收集 1GB 的合并跟踪，请将以下行添加到 `logger.properties` 文件中：

```
traceFiles=5
traceSize=200
```

这会导致记录器将跟踪数据写入到最多 5 个文件中，每个文件的大小都为 200MB。

有关这些记录器属性的更多信息，请参阅 [MFT `logger.properties` 文件](#)。

- iv) 找到包含需要跟踪的记录器的 JCL 的数据集。
- v) 提交数据集中的 BFGLGSP 成员以停止记录器。
- vi) 通过提交数据集中的 BFGLGST 成员来重新启动记录器。
- vii) 编辑数据集中的 BFGLGTC 成员，然后找到包含以下文本的行：

```
-traceLogger
```

其下方的文本包含要跟踪的内部类和包的列表。缺省情况下，将此列表设置为：

```
com.ibm.wmqfte=all
```

除非您的 IBM 支持代表另有指定，否则请按原样保留此值。

- viii) 当需要启用跟踪时，请提交 BFGLGTC 成员。
- ix) 重现该问题。
- x) 再次编辑 BFGLGTC 成员，并将 `-traceLogger` 参数设置为 `=off`，如下所示：

```
-traceLogger =off +
```

- xi) 重新提交 BFGLGTC 成员以关闭跟踪。
- b) 第二个选项是联系您的 IBM 支持代表（如果无法在短时间内停止记录器）。然后，您可以讨论要使用的备用跟踪规范，以减少要生成的跟踪数据量。

结果

生成的跟踪文件将写入到 `BFG_DATA/mqft/logs/coordination_qmgr_name/logger_name/logs/trace%PID%` 目录，其中 `%PID%` 是记录器的进程标识。

MFT 常见问题

Managed File Transfer 网络中可能出现的常见问题。

- 如果文本传输失败，并具有以下错误：

```
BFGI00060E: Text data conversion has failed
```

发生这种情况有两种原因：

1. 源文件中的一个或多个字符无法从源文件代码页转换到目标文件代码页。如果代码页具有不同字符集并且一些字符无法在这些字符集之间转换，那么可能会出现此问题。
如果您可以接受不对某些字符进行转换，那么可以在目标代理定义替换字符序列，以便传输不会失败。指定代理属性 `textReplacementCharacterSequence` 来定义替换字符序列。有关更多信息，请参阅 [第 461 页的表 45](#)。
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 133 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 364 页的『fteCreateTransfer: 启动新的文件传输』](#)。
 - 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 209 页的『将 Apache Ant 与 MFT 结合使用』](#)。

要检查您是否为传输选择了正确的源文件编码，请执行以下步骤：

1. 将目标文件编码设置为 UTF-8。
 2. 以文本方式传输文件。
 3. 使用 UTF-8 文件查看器来查看文件的内容。如果文件中所有字符都正确显示，那么源文件编码正确。
- 如果您从 **fteCreateAgent** 命令中看到以下输出：

```
BFGMQ1007I: 无法联系协调队列管理器或已拒绝协调队列管理器
connection attempt.
IBM MQ 原因码是 2058。 将不会发布有关该代理是否存在的信息。
```

它指示协调队列管理器无法访问，并提供 IBM MQ 原因码以说明原因。此参考消息可能指示协调队列管理器当前不可用或者您未正确定义配置。

- 如果您在使用用户出口例程，并且在调用用户出口时或者在已经调用出口之后发生故障（例如，产品故障或断电），那么可能多次调用用户出口。
- 如果您的系统上具有一个包含队列管理器的代理，并且其 IP 地址由 DHCP 分配（而不是静态 IP 地址），并且代理使用客户机 TCP/IP 连接来连接到该系统，那么您必须在设置了以下系统环境变量的情况下启动代理：

– **Windows** 在 Windows 上：

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

– **UNIX** 在 UNIX 上：

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

其中，*value* 是指清除 JVM 的高速缓存 DNS 值所经历的时间间隔（单位为秒）。如果队列管理器系统的 IP 地址由于任何原因而重新分配（例如，由于网络中断、IP 租约过期或系统重新引导），那么代理会报告其与队列管理器丢失连接。清除 JVM DNS 高速缓存后，代理可以成功地重新连接。如果未设置环境变量，那么在不重新启动 JVM 的情况下，代理无法在此场景下重新连接。此行为的原因是 JVM 在内部对主机名的 IP 地址进行了高速缓存，并且缺省情况下不会刷新这些 IP 地址。

- 如果运行 **fteStartAgent** 命令并且看到以下错误消息，那么您的环境可能具有与 Managed File Transfer 冲突的附加库路径：

```
BFGCL0001E: An internal error has occurred. 异常为："CC=2;RC=2495;AMQ8568："
找不到本地 JNI 库 "mqjbnd"。[3=mqjbnd]
```

如果 LD_LIBRARY_PATH 或 LIBPATH 环境变量设置为先引用 64 位版本的库，然后引用 32 位版本的库，而同时正在使用 32 位版本的 Java 运行代理（对于大部分平台，都是这种情况），那么会发生此错误。

要解决此问题，请设置 Managed File Transfer 代理属性 `javaLibraryPath` 以引用库的正确位置。例如，对于 AIX 上的 `mqjbnd`，请设置为：`/usr/mqm/java/lib`。对于 Linux 上的 `mqjbnd`，请设置为：`/opt/mqm/java/lib`

- 如果您通过在代理属性文件中指定 `authorityChecking=true` 而启用了用户权限检查，并且所有权限检查都失败（即用户对相关权限队列具有必需权限）：
 - 请确保运行的代理的用户在代理队列管理器上具有 ALT_USER 访问控制。
- 如果已通过在代理属性文件中指定 `authorityChecking=true` 来启用用户权限检查，并且 IBM MQ 错误消息已写入代理 `output0.log` 文件，请执行下列其中一项操作：
 - 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 73 页的『z/OS 上的 MFT 环境变量』。

MFT 中有关 JZOS 的常见问题

下面是在遇到 JZOS 问题时的一些建议。

- 如果 JZOS 无法成功处理，请执行下列操作：
 - 在 JCL 中添加“，PARM='+T'”。例如：

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

- 在环境文件中添加 `set -x`。

- 如果结果为：

```
JVMJZBL1038E 子 shell 进程退出，退出代码为 1
```

```
JVMJZBL1042E JZOS 批处理启动程序失败，返回码=102
```

这表示您的环境文件和 Managed File Transfer 命令有问题。这可能是因为指定的路径无效。

- 从您的环境文件中找到 **BFG_PROD** 的值。

1. 转至 OMVS 并使用 `ls -ltr` 命令。

例如，如果 **BFG_PROD** 为 `/HMF8800/`，请输入以下命令：

```
ls -ltr HMF8800/bin/fteBatch
```

2. 检查此文件是否存在，以及批处理作业是否有权读取此文件。
3. 解决所有问题。

- 如果 JCL 仍然无法正确处理：

1. 请在 USS 中创建一个文件（例如，`myenv`），然后使用编辑器将信息从环境文件复制到此 `myenv` 文件。
2. 保存该文件。
3. 在命令行中，使用命令 `chmod +x myenv` 以允许运行此文件。
4. 发出 `. myenv` 命令。注意，这是（点空格文件名）。

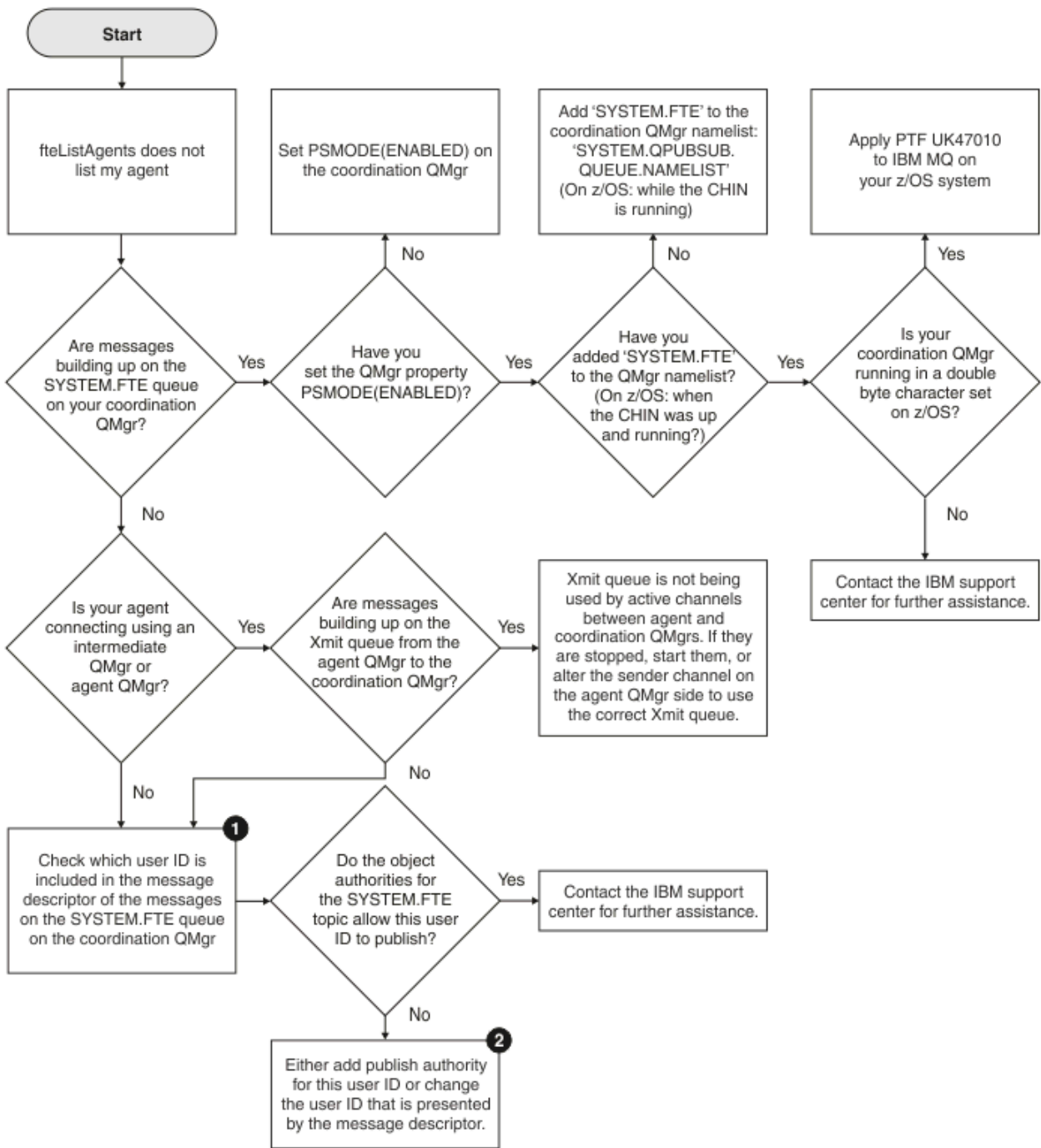
运行此命令将报告 `myenv` 文件中的任何错误。

5. 更正 `myenv` 和环境文件中的任何错误。

当 `fteListAgents` 命令未列出您的 MFT 代理时该怎么做

如果代理未通过 `fteListAgents` 命令列出，或者未显示在 IBM MQ Explorer 中，或者文件传输未显示在 IBM MQ Explorer 的 **传输日志** 中，那么您可以执行许多问题确定步骤来调查原因。

使用以下流程图可帮助您诊断问题并决定接下来采取的操作：



流程图要点：

1. 有关如何检查提供的用户标识的更多信息，请参阅第 266 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 302 页的『用于发布 MFT 代理日志和状态消息的权限』。

在 MFT 代理进程消失但未记录任何诊断信息时该怎么做

在 UNIX 平台上，如果代理进程已消失，但代理日志文件不包含任何解释，这可能由于启动此代理的方式所致。

您可通过如下方式查看代理诊断信息：

- 检查代理的日志文件是否声明已停止此代理。
- 检查代理锁文件 `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。

如果代理被显示为处于 UNKNOWN 状态，该怎么办

您的代理正在运行并成功响应 `ftePingAgent` 命令，且正在正常传输项目。但是，`fteListAgents` 和 `fteShowAgentDetails` 命令以及 IBM MQ Explorer Managed File Transfer 插件报告代理处于 UNKNOWN 状态。

发生此问题的原因

每个代理都会定期将其状态发布到协调队列管理器上的 SYSTEM.FTE 主题。代理发布其状态的频率由以下代理属性控制：

`agentStatusPublishRateLimit`

由于文件传输状态的变化，代理重新发布其状态的最大速率（以秒为单位）。此属性的缺省值为 30 秒。

`agentStatusPublishRateMin`

代理发布其状态的最小速率（以秒为单位）。此值必须大于或等于 `agentStatusPublishRateLimit` 属性的值。`agentStatusPublishRateMin` 属性的缺省值为 300 秒（或 5 分钟）。

`fteListAgents` 和 `fteShowAgentDetails` 命令以及 IBM MQ Explorer Managed File Transfer (MFT) 插件通过这些发布来确定代理的状态。为此，这些命令和插件会执行以下步骤：

1. 连接到协调队列管理器。
2. 预订 SYSTEM.FTE 主题。
3. 接收代理状态发布。
4. 在协调队列管理器上创建临时队列。
5. 将消息放入临时队列，并保存放置时间，以获取协调队列管理器系统上的当前时间。
6. 关闭临时队列。
7. 使用发布中包含的信息以及当前时间来确定代理的状态。
8. 与协调队列管理器断开连接。

如果代理发布状态的时间与当前时间的差值大于以下值，就会认为代理的状态消息是过时的：代理属性 `agentStatusPublishRateMin`（包含在状态消息中）的值加上高级协调队列管理器属性 `agentStatusJitterTolerance` 的值。

缺省情况下，`agentStatusJitterTolerance` 属性的值为 3000 毫秒（3 秒）。

如果将 `agentStatusPublishRateMin` 和 `agentStatusJitterTolerance` 属性设置为其缺省值，那么当代理发布状态的时间与当前时间的差值大于 303 秒（或 5 分 3 秒）时，会将该代理的状态视为已过时。

`fteListAgents` 和 `fteShowAgentDetails` 命令以及 IBM MQ Explorer MFT 插件会将具有过时状态消息的所有代理都报告为处于 UNKNOWN 状态。

由于下列一个原因，代理的状态发布可能已过时：

1. 运行代理队列管理器的系统与协调队列管理器所在系统之间的系统时间存在显著差异。
2. 代理队列管理器与协调队列管理器之间的通道已停止使用（这会阻止新的状态消息到达协调队列管理器）。
3. 授权问题阻止代理将其状态发布到协调队列管理器上的 SYSTEM.FTE 主题。
4. 代理发生故障。

对问题进行故障诊断

需要执行一些步骤来确定代理的状态为何会被报告为 UNKNOWN：

1. 通过登录代理系统，检查代理是否正在运行。如果代理已停止运行，请调查它不再运行的原因。一旦它再次运行，即检查现在是否正确报告了其状态。
2. 检查协调队列管理器是否正在运行。如果未运行，请将其重新启动，然后使用 **fteListAgents** 或 **fteShowAgentDetails** 命令或者 IBM MQ Explorer MFT 插件来查看现在是否正确报告了代理状态。
3. 如果代理和协调队列管理器正在运行，请检查其错误日志以查看是否存在阻止代理发布其状态消息的任何授权问题。如果日志显示出现授权问题，请确保运行代理进程的用户具有正确的权限，能够将消息发布到协调队列管理器上的 SYSTEM.FTE 主题。

如果队列管理器的错误日志未报告任何授权问题，请确认状态消息未卡在 IBM MQ 网络中。验证用于将消息从代理队列管理器路由至协调队列管理器的所有发送方和接收方通道是否正在运行。

如果这些通道正在运行，请检查与通道相关联的传输队列，以确保状态消息未卡在这些通道上。此外，您还应该检查队列管理器的任何死信队列，确保没有出于某种原因而将状态消息放置在此处。

4. 如果通道正在运行，并且状态消息正在通过 IBM MQ 网络流动，那么接下来要检查的是队列管理器的排队发布/预订引擎是否正在提取消息。

fteSetupCoordination 命令 (用于定义协调队列管理器) 为您提供了一些 MQSC 命令，这些命令必须在协调队列管理器上运行，才能配置排队的发布/预订引擎以接收发布。这些命令执行以下步骤：

- 创建 SYSTEM.FTE 主题及其关联的主题字符串。
- 定义名为 SYSTEM.FTE 。
- 通过将队列管理器上的 **PSMODE** 属性设置为 **ENABLED**，启用已排队的发布/预订引擎。
- 修改 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名称列表，由排队的发布/预订引擎使用，以便它包含新 SYSTEM.FTE 队列。

有关这方面的更多信息，包括需要运行的 MQSC 命令，请参阅第 430 页的『[fteSetupCoordination](#)』。

如果 SYSTEM.FTE 队列，然后您应该检查 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名称列表，并包含该队列的条目。如果缺少该条目，那么排队的发布/预订引擎将不会检测来自代理程序的任何入局状态消息，并且不会处理这些消息。

您还应确保队列管理器上的 **PSMODE** 属性设置为 **ENABLED**，这将打开排队的发布/预订引擎。

5. 如果通道正在运行，并且状态消息正在流经 IBM MQ 网络并且正在由队列管理器的已排队发布/预订引擎从 SYSTEM.FTE 队列中获取，请收集以下跟踪：
 - 代理的 IBM MQ MFT 跟踪信息，涵盖的时间段等于代理属性 **agentStatusPublishRateMin** 值的三倍。这会确保在跟踪所涵盖的时间范围内，代理至少发布三条包含其状态的消息。应该使用跟踪规范动态收集跟踪信息：

```
com.ibm.wmqfte.statestore.impl.FTEAgentStatusPublisher,  
com.ibm.wmqfte.utils.AgentStatusDetails,  
com.ibm.wmqfte.wmqiface.AgentPublicationUtils,  
com.ibm.wmqfte.wmqiface.RFHMessageFactory=all
```

注：使用这些字符串输出减少的跟踪量。

有关如何为 IBM MQ for Multiplatforms 上运行的代理启用跟踪的信息，请参阅第 232 页的『[自动收集 Managed File Transfer 代理跟踪信息](#)』。

有关如何为 IBM MQ for z/OS 上运行的代理启用跟踪的信息，请参阅第 239 页的『[动态收集 Managed File Transfer for z/OS 代理跟踪信息](#)』。

- 队列管理器的并发跟踪，用于将状态消息从代理队列管理器路由到协调队列管理器。
- **fteListAgents** 命令的跟踪，涵盖代理被显示为处于 UNKNOWN 状态的时间。应该使用跟踪规范来收集跟踪信息：

```
com.ibm.wmqfte=all
```

有关如何为 IBM MQ for Multiplatforms 上运行的命令启用跟踪的信息，请参阅第 233 页的『[在多平台上跟踪 Managed File Transfer 命令](#)』。

有关如何为 IBM MQ for z/OS 上运行的命令启用跟踪的信息，请参阅第 242 页的『[跟踪 Managed File Transfer for z/OS 命令](#)』。

收集跟踪后，应将其提供给 IBM 支持人员进行分析。

相关信息

[fteListAgents](#)

[fteShowAgentDetails](#)

[MFT 代理状态值](#)

[MFT agent.properties 文件](#)

[MFT coordination.properties 文件](#)

当 ftePingAgent 超时并报告 BFGCL0214I 消息时应执行的操作

ftePingAgent 是随 IBM MQ Managed File Transfer 提供实用命令行实用程序，它使您能够检查代理程序是否可访问以及是否能够对请求作出响应。

命令的工作方式

您可以使用 **ftePingAgent** 命令检查代理程序是否可访问，以及它是否能够处理请求。运行该命令时，它将执行以下步骤：

- 连接到用于 Managed File Transfer (MFT) 拓扑的命令队列管理器。
- 在命令队列管理器上创建一个临时应答队列。

缺省情况下，该临时队列的名称以前缀 WMQFTE 开头。但是，您可以通过在 [MFT 命令 .properties 文件](#) 中为安装设置 **dynamicQueuePrefix** 属性更改此值。

- 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。
- 等待应答消息到达临时应答队列。

代理程序中的其中一个线程是 CommandHandler。此线程从代理的 SYSTEM.FTE.COMMAND.agent_name 队列获取消息并对其进行处理。

如果此线程接收到包含 Ping MFT 代理请求的消息，那么它会构建 MFT 代理应答消息，并将其发送到命令队列管理器上的临时队列。此消息将通过代理的队列管理器。

一旦消息到达临时队列，**ftePingAgent** 命令就会选取该消息。然后，该命令会在退出之前将类似如下所示的消息写入控制台：

```
BFGCL0213I: agent <agent_name> responded to ping in 0.088 seconds.
```

以下两个图显示了这一流程：

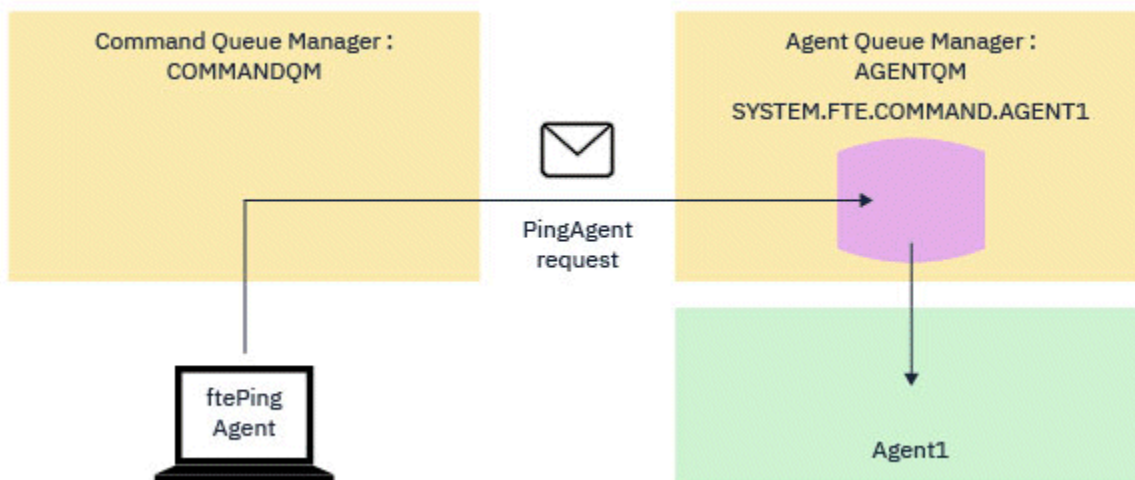


图 7: pingAgent 请求通过命令队列管理器进入代理队列管理器上的 `SYSTEM.FTE.COMMAND.agent_name` 队列

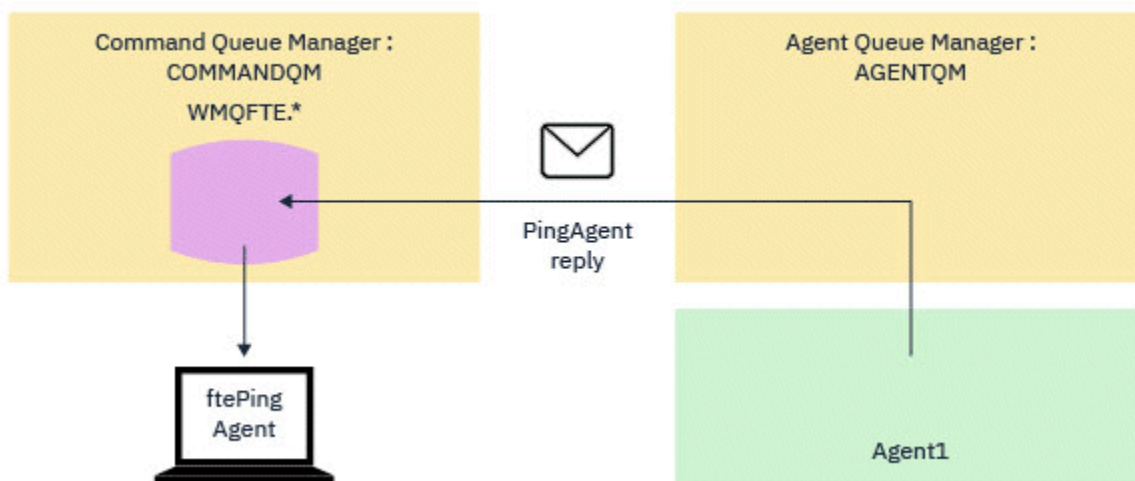


图 8: pingAgent 应答通过代理队列管理器返回到命令队列管理器。

当命令超时时应执行的操作

缺省情况下，**ftePingAgent** 命令会等待 5 秒，以便 MFT 代理程序应答消息到达临时队列。如果应答消息未在 5 秒内到达，那么该命令会将 BFGCL0214I 消息写入控制台。以下消息是一个示例：

BFGCL0214I: 代理 AGENT1 在 5 秒后未响应 ping 操作。

您可以使用以下步骤来调查应答消息未到达的原因：

- 首先要执行的操作是检查代理程序是否正在运行。如果不是，那么它无法对命令发送的 Ping MFT 代理请求作出响应。
- 如果代理程序正在运行并且忙于处理请求，那么可能需要超过 5 秒才能获取 Ping MFT 代理程序请求并将回复发送回来。

要查看是否应该再次重新运行 **ftePingAgent** 命令，请使用 **-w** 参数指定更长的等待时间间隔。例如，要指定 60 秒的等待时间间隔，请发出以下命令：

```
ftePingAgent -w 60 AGENT1
```


- 如果该命令仍超时，请检查命令队列管理器与代理队列管理器之间通过 IBM MQ 网络的路径。如果路径中的一个或多个通道发生故障，那么 Ping MFT 代理程序请求消息和/或 MFT 代理程序应答消息将停留在某个位置的传输队列上。在此情况下，您应重新启动通道并重新运行 **ftePingAgent** 命令。

如果该命令在您执行上述步骤后仍报告 BFGCL0214I 消息，那么需要跟踪 Ping MFT 代理程序请求和 MFT 代理程序回复消息，因为它们流经 IBM MQ 网络以查看：

- Ping MFT 代理消息是否曾经到达 SYSTEM.FTE.COMMAND.agent_name 队列。
- 如果代理从队列中获取消息，并将 MFT 代理应答消息发送回。

要执行此操作，您应该执行以下步骤：

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

执行此操作的方式取决于运行代理程序的平台。对于正在运行的代理程序：

- IBM MQ for Multiplatforms，请参阅第 232 页的『自动收集 Managed File Transfer 代理跟踪信息』。
- IBM MQ for z/OS，请参阅第 239 页的『动态收集 Managed File Transfer for z/OS 代理跟踪信息』。
- 接下来，使用跟踪规范 com.ibm.wmqfte=all 在启用跟踪的情况下运行 **ftePingAgent** 命令。有关跟踪命令的信息：
 - IBM MQ for Multiplatforms，请参阅第 233 页的『在多平台上跟踪 Managed File Transfer 命令』。
 - IBM MQ for z/OS，请参阅第 242 页的『跟踪 Managed File Transfer for z/OS 命令』。

当该命令超时时，停止代理程序跟踪和队列管理器跟踪。然后，应该将代理程序和队列管理器跟踪以及命令中的跟踪提供给 IBM 支持人员进行分析。

当 MFT 代理或记录器配置不安全时该怎么做

如果某个 Managed File Transfer 进程检测到某个配置文件包含敏感信息，是密钥库或信任密钥库文件，并且具有系统范围的读、写或删除许可权，那么如果是在启动时检测到该情况，该进程将无法启动。如果该情况不是在启动时而是在运行时检测到的，那么 Managed File Transfer 会生成警告消息并忽略该配置文件的内容。这与协议网桥和 Connect:Direct 网桥功能相关，如果在代理运行期间某个配置发生更改，这些功能会重新装入该配置。

请完成以下检查以确定问题原因：

1. 从提供的错误消息中确定已报告为不安全的配置文件。
2. 确保该文件的访问许可权与所需的需求相匹配。有关更多信息，请参阅第 301 页的『用于访问敏感配置信息的 MFT 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

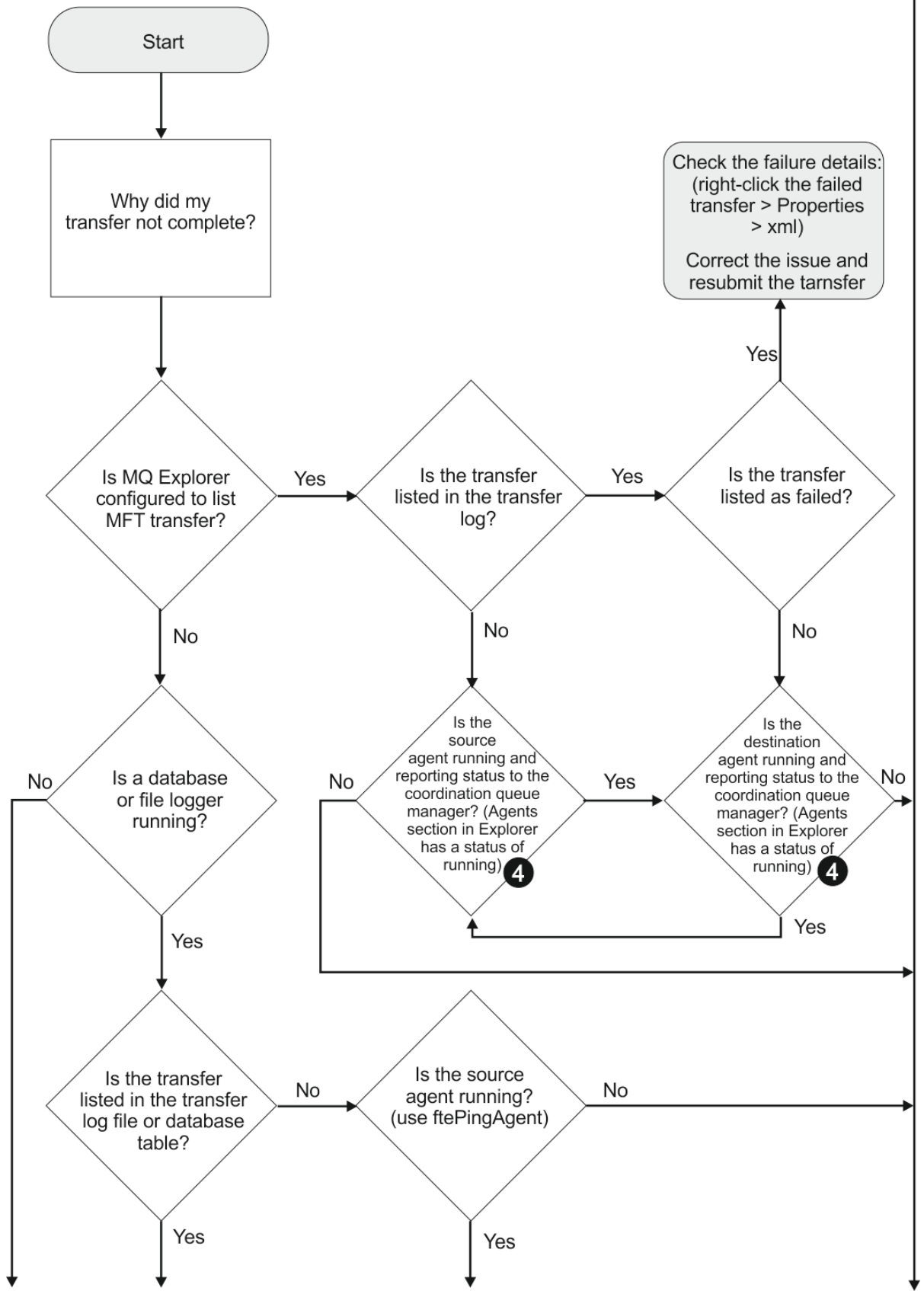
第 301 页的『用于访问敏感配置信息的 MFT 许可权』

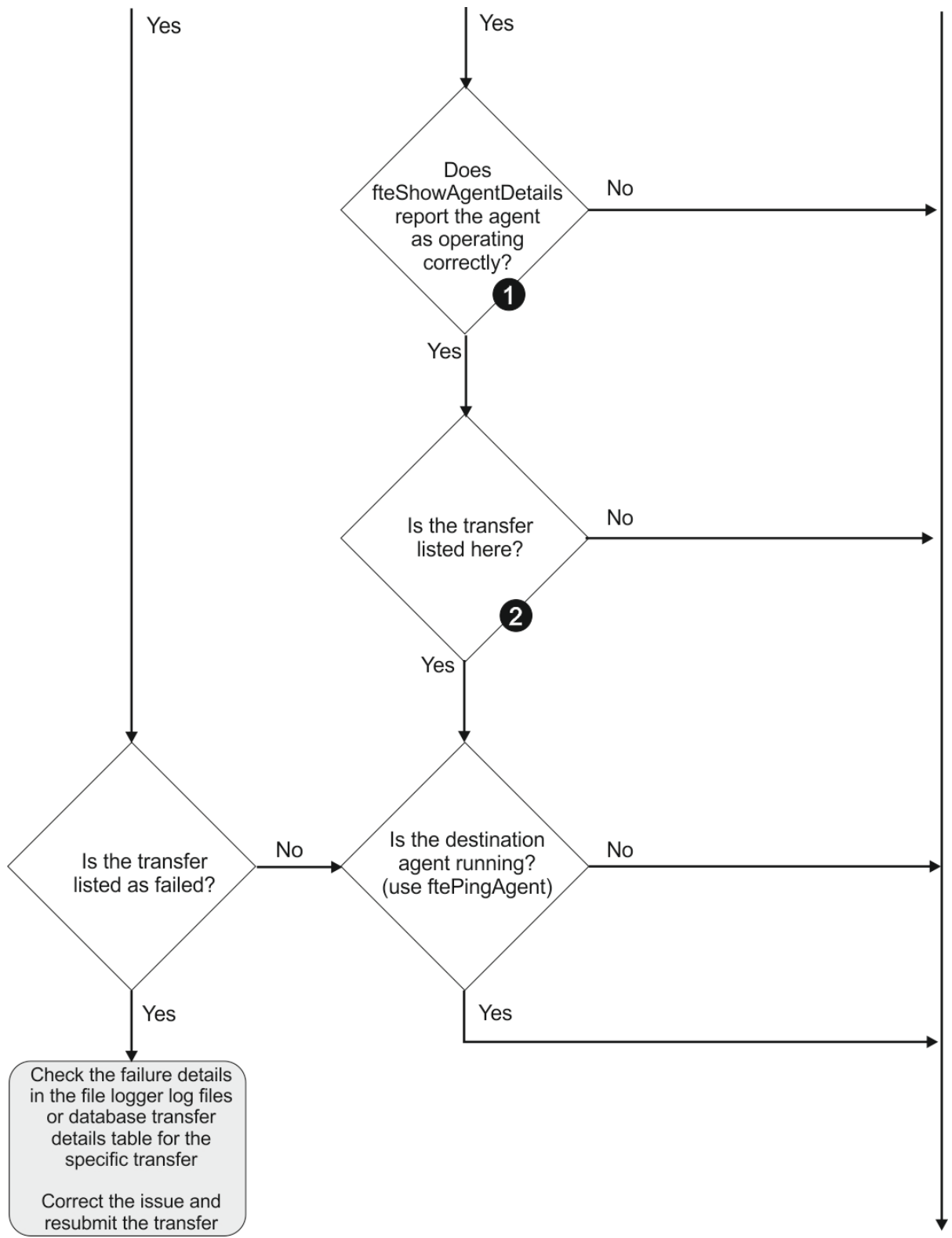
用于存储敏感配置信息的任何文件（指的是从 IBM MQ 配置树中引用的任何文件）不得具有系统范围的读、写或（适用时）删除许可权。这些限制也适用于信任库和密钥库文件。

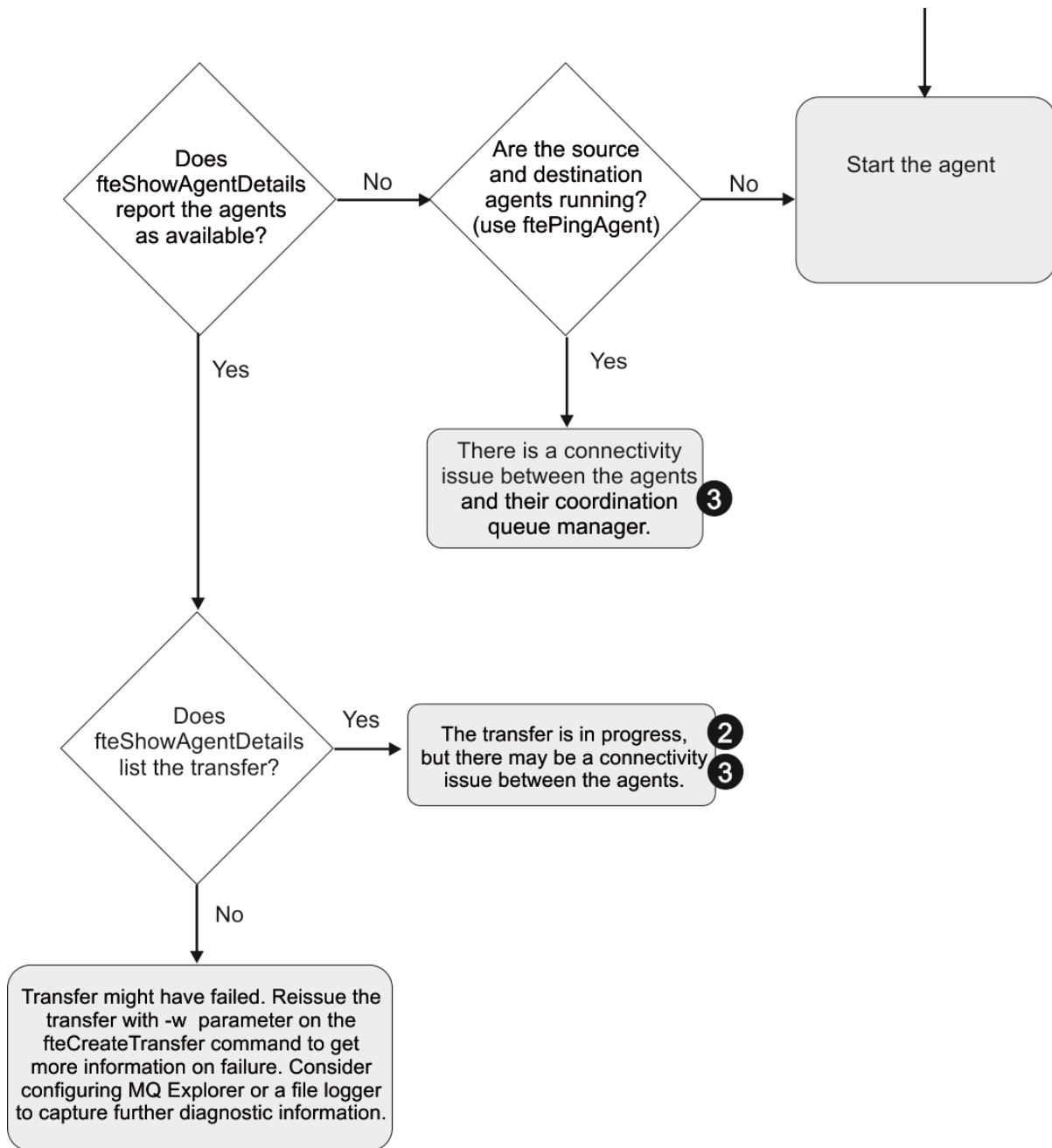
当传输没有完成时应如何处理

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。如果代理报告自己已成功启动，但 IBM MQ Explorer 或 **fteShowAgentDetails** 都没有报告该代理正在运行，请检查代理队列管理器与协调队列管理器之间的连接。可能是因为队列管理器间通道不可用。
2. 如果源代理将传输标识列示为 `In progress` 传输，但目标代理未列示，那么源队列管理器与目标队列管理器之间可能存在连接问题。在 `command.properties` 文件中，从目标代理机器向使用目标代理队列管理器作为命令队列管理器的源代理使用 **ftePingAgent** 命令。您也可以按相反方向运行此命令，即从源到目标。
3. 如果源和目标代理都将传输标识列示为 `In progress`，那么这表明自传输启动以来，源和目标队列管理器之间存在连接问题。在 `command.properties` 文件中，从目标代理机器向使用目标代理队列管理器作为命令队列管理器的源代理使用 **ftePingAgent** 命令。您也可以按相反方向运行此命令，即从源到目标。
4. 如果已运行过此循环，请检查以下任一声明是否与您的情况相关：

- 源代理和目标代理都报告为 **Running**，但未列出任何传输。传输请求未到达代理命令队列，或者代理虽然报告为 **Running**，但不再监视命令队列。请检查源代理的 `output0.log` 中是否有错误。从发送传输的同一机器向源代理使用 **ftePingAgent** 命令，以验证命令队列管理器和代理队列管理器之间的连接，以及代理是否正在为命令队列提供服务。
- 源和目标代理都报告为 **Running**，并且传输列示为 **In progress, recovering**。在 `command.properties` 文件中，从目标代理机器向使用目标代理队列管理器作为命令队列管理器的源代理使用 **ftePingAgent** 命令。您也可以按相反方向运行此命令，即从源到目标。

在您认为文件传输卡住时应该怎么办

在高负荷的系统上或者在源和目标代理之间发生网络问题时，传输可能会偶尔显示为卡住，进入排队或恢复状态。导致此情况的因素有很多。

请完成以下检查以确定问题原因：

1. 使用 **ftePingAgent** 命令或在 IBM 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。可以使用带有 **-v** (verbose) 参数的 **fteShowAgentDetails** 命令，或者在 IBM MQ Explorer 代理面板中右键单击代理名称并选择**属性**，以查看代理的当前传输活动。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。

当已调度的文件传输未运行或被延迟时该怎么做

如果您具有到期未运行或被延迟的已调度传输，那么原因可能是代理正在处理其命令队列中的命令。由于代理很繁忙，因此未检查已调度传输，从而未运行已调度传输。

要解决该问题，请使用以下步骤之一：

- 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 460 页的『MFT agent.properties 文件』。
- 或者，请使用资源监视器代替已调度传输。资源监视器的工作方式与已调度传输不同，不受繁忙代理的影响。例如，如果要获取目标系统上的最新文件，资源监视器将减少网络流量。这是因为只有当新版本可用时才会传输文件，而不是自动传输文件。但是，在协议网桥代理或 **Connect:Direct** 网桥代理上不支持资源监视。

有关更多信息，请参阅第 141 页的『监视 MFT 资源』。

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。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 177 页的『协议网桥』

通过协议网桥，Managed File Transfer (MFT) 网络可以访问您的 MFT 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 MFT 目录资源监视器未触发文件时该怎么做

目录资源监视器对与触发器规范匹配的文件目录进行轮询。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生成 Managed File Transfer 传输请求：

- 检测到之前不存在的触发器文件。
- 更新了触发器文件，导致最后修改日期发生变化。

第二次触发的潜在场景包括：

- 移除和替换了文件。
- 文件被一个应用程序锁定，然后被另一个应用程序解锁。
- 监控文件系统发生故障。例如，如果出现网络连接故障，则可能导致文件显示为被移除和替换。
- 文件目录被另一个应用程序更新，导致最后修改日期发生变化。

用于对代理的所有资源监视器设置参考级别输出的命令

V 9.0.3

在此示例中，将监视所有资源监视器，因为您尚未指定特定资源监视器的名称。代理的名称为 AGENT1。

```
fteSetAgentLogLevel -logMonitor=info AGENT1
```

请参阅 [fteSetAgentLogLevel](#)，以获取 **logMonitor** 参数的详细信息以及有关如何使用各种选项的示例。

```

=====
[21/04/2017 11:08:49:367 IST] BFGUT0036I: Resource monitor event log level has changed to "info" for all resource monitors of
this agent.
=====

=====
Date      Time          Thread ID   Monitor Name      Event
Description
=====
[21/04/2017 11:08:51:842 IST] 00000023   QMON             Monitor Started   Resource Monitor Started
[21/04/2017 11:08:51:844 IST] 00000025   QMON             Start Poll        New poll cycle started
[21/04/2017 11:08:51:924 IST] 00000023   MON1            Monitor Started   Resource Monitor Started
[21/04/2017 11:08:51:925 IST] 00000026   MON1            Start Poll        New poll cycle started
[21/04/2017 11:08:52:029 IST] 00000026   MON1            End Poll          Poll cycle completed in 105
milli seconds. Trigger items matched [ 0 ]
[21/04/2017 11:08:52:055 IST] 00000025   QMON             End Poll          Poll cycle completed in 212
milli seconds. Trigger items matched [ 0 ]
[21/04/2017 11:09:51:840 IST] 00000025   QMON             Start Poll        New poll cycle started
[21/04/2017 11:09:51:875 IST] 00000025   QMON             End Poll          Poll cycle completed in 34
milli seconds. Trigger items matched [ 0 ]
[21/04/2017 11:09:51:924 IST] 00000026   MON1            Start Poll        New poll cycle started
[21/04/2017 11:09:51:969 IST] 00000026   MON1            End Poll          Poll cycle completed in 45
milli seconds. Trigger items matched [ 0 ]
[21/04/2017 11:10:51:840 IST] 00000025   QMON             Start Poll        New poll cycle started
[21/04/2017 11:10:51:924 IST] 00000026   MON1            Start Poll        New poll cycle started
[21/04/2017 11:10:51:962 IST] 00000025   QMON             End Poll          Poll cycle completed in 121
milli seconds. Trigger items matched [ 0 ]
[21/04/2017 11:10:51:963 IST] 00000026   MON1            End Poll          Poll cycle completed in 39
milli seconds. Trigger items matched [ 0 ]
[21/04/2017 11:10:55:063 IST] 00000041   MON1            Monitor Stopped   Resource Monitor Stopped
[21/04/2017 11:10:55:079 IST] 00000041   QMON            Monitor Stopped   Resource Monitor Stopped
=====

```

相关参考

第 418 页的『[fteSetAgentLogLevel: 设置 MFT 协议网桥代理日志级别](#)』

使用 **fteSetAgentLogLevel** 命令对 Managed File Transfer 协议网桥代理与 FTP/SFTP/FTPS 文件服务器之间的文件传输启用或禁用诊断信息日志记录。

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。

发生此问题的原因

1. 消息到文件的传输失败，并且消息或组留在队列上。
2. 新的消息或组到达队列。
3. 新的消息或组触发资源监视器。
4. 资源监视器创建一个新传输，此传输使用新消息或组上的 MQMD 消息描述符以及队列上的第一个消息或组中的数据。
5. 创建的文件中包含错误数据。

避免此问题

要避免遇到此问题，必须使用 **fteCreateTransfer** 命令手动创建传输定义文件，并编辑文件的 <queue> 元素，以包含属性 `groupId="${GROUPID}"`。然后使用 **fteCreateMonitor** 命令来提交传输定义文件。

示例

在此示例中：源代理名为 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="\${GROUPID}"。将行

```
<queue useGroups="true">LIVE_QUEUE</queue>
```

到

```
<queue useGroups="true" groupId="${GROUPID}">LIVE_QUEUE</queue>
```

该属性是必需的, 以便传输从队列中 (而非队列上的第一个组或消息中) 读取触发传输的组或消息。

3. 通过运行以下命令来创建监视器:

```
fteCreateMonitor -ma AGENT_MON -mq LIVE_QUEUE -mn QueueMon1 -mt /tmp/TransferDefinition1.xml  
-tr completeGroups -dv WMQFTEFileName=UNKNOWN
```

监视器会每隔 60 秒对队列进行一次轮询, 以查看是否有新的组或消息到达队列。

如果目标队列是集群队列或集群队列的别名, 那么该执行什么操作?

使用 Managed File Transfer 将文件传输到队列中时, 如果使用的目标队列是集群队列或集群队列的别名, 那么您将收到原因码 2085 或 2082。从 IBM WebSphere MQ 7.5.0 Fix Pack 4 开始, 将属性 enableClusterQueueInputOutput 设置为 true 可解决此问题。

发生此问题的原因

当 **-dq** 上无显式队列管理器名称时, 会将目标代理的队列管理器名称附加到 **-dq** 参数的队列名称。由于在连接到没有该本地集群队列的集群 MQ queueManager 时, 无法在 MQOPEN 调用中指定 queueManager 对象, 由此产生原因码 2085 或 2082。

避免此问题

1. 在队列管理器上创建集群队列。
2. 设置指向集群队列的远程队列定义。

示例

本示例使用远程队列定义。

配置:

- 源代理: *SAGENT*
- 源代理队列管理器: *SQM*
- 目标代理: *DAGENT*
- 目标代理队列管理器: *DQM*
- 传输的目标队列是队列管理器 *SQM* 上的 *CQ6*

要针对 *SQM* 中的集群队列 *CQ6* 定义 *DQM* 上的远程队列定义 *Q6_SQM* (假设已在 *SQM* 中定义集群队列 *CQ6*), 请在 *DQM* 队列管理器上发出 MQSC 命令:

```
define qremote(Q6_SQM) rname(CQ6) rqmname(SQM) xmitq(SQM)
```

注: rname 指向集群队列。

您现在可以传输至队列。例如:

```
fteCreateTransfer -sa SAGENT -sm SQM -da DAGENT -dm DQM -dq Q6_SQM /tmp/single_record.txt
```

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 Managed File Transfer 网络，请不要频繁使用 IBM MQ Explorer 插件，否则此消息数据可能会填满本地文件系统。

要除去 **SYSTEM.MANAGED.DURABLE** 队列上消息的聚集，可执行以下其中一种操作：

- 启动使用持久预订的 IBM MQ Explorer。IBM MQ Explorer 的 Managed File Transfer 插件使用队列中的消息。
- 手动从队列删除消息。

V 9.0.2

您可以通过以下某种方法避免在持久队列上积压消息：

- 指定 IBM MQ Explorer 插件使用对协调队列管理器的非持久预订。在 IBM MQ Explorer 中执行以下步骤：
 1. 选择窗口 > 首选项 > **IBM MQ Explorer > Managed File Transfer**
 2. 在**传输日志订阅类型**列表中，选择 **NON_DURABLE**。
- 清除 IBM MQ Explorer MFT 插件所创建的来自协调队列管理器的持久预订。

持久预订的名称添加了前缀以显示该预订是通过 IBM MQ Explorer MFT 插件、主机名和用户名创建的，例如，MQExplorer_MFT_Plugin_HOST_TJWatson。

相关参考

第 88 页的『保留 MFT 日志消息』

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。

发布前检查消息

由于代理可连接到 IBM WebSphere MQ 6.0 队列管理器，因此代理不使用 IBM WebSphere MQ 7.0 中引入的直接发布方法。事实上，代理将普通消息发送到包含 **MQRFH** 头的协调队列管理器。**MQRFH** 头请求发布消息的有效内容。这些消息会被发送到协调队列管理器上的 **SYSTEM.FTE** 队列，通常会立即从该队列发布这些消息。如果错误条件阻止此发布，您可以在尝试发布前检查队列上的消息以帮助进行诊断。可通过完成以下步骤来执行此操作：

1. 在协调队列管理器中禁用发布/预订引擎。

您可以使用 IBM MQ Explorer 或 MQSC 命令来完成该步骤。请注意，如果协调队列管理器同样用于其他目的，那么这将暂时停止队列管理器上的所有发布/预订活动，包括与 Managed File Transfer 无关的活动。

IBM MQ Explorer:

- a. 在“导航器”视图中，右键单击协调队列管理器，然后选择**属性**。
- b. 从**属性**窗格中选择**发布/预订**。
- c. 从**发布/预订方式**列表中选择**兼容性**。

MQSC:

```
ALTER QMGR PSMODE(COMPAT)
```

2. 发送另一消息。

执行存在发布问题的 Managed File Transfer 操作。例如，对于代理注册，任何时候启动代理时都将发送消息（您无需重复删除和创建代理以生成注册消息）。由于已禁用发布/预订引擎，因此将不会进行发布。

3. 浏览协调队列管理器上的 SYSTEM.FTE 队列。

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

IBM MQ Explorer:

- a. 在“导航器”视图中，展开协调队列管理器并单击**队列**。在“内容”视图中，右键单击 SYSTEM.FTE 队列并选择**浏览消息**。将打开**消息浏览器**窗口并显示将发布的消息。
- b. **用户标识**列显示消息描述符中所含的用户标识。发布失败的常见原因是该用户标识对于 SYSTEM.FTE 主题没有发布权限。
- c. 您可以通过右键单击消息并选择**属性**，查找有关每条消息（包括将发布的 XML）的更多信息。

没有用于检查消息内容的 MQSC 命令。如果您没有 IBM MQ Explorer，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 **amqsbcbg** 样本程序 (如果已安装)，如以下主题中所述: **浏览队列**。 **UserIdentifier** 行显示用户标识。或者，您可以使用 **dmpmqmsg**; 可以在如下行中找到消息的用户标识:

```
A RTM MQ24
A USR JOHNDOE
A ACC 1A0FD4D8F2F4C3C8C9D5F1F9C6F7C1C3F3F00019F7AC3000000000000000000
```

该示例中的第二行是该消息的消息描述符用户标识。

4. 重新启用协调队列管理器的发布/预订引擎。

您可以使用 IBM MQ Explorer 或 MQSC 命令来完成该步骤。重新启用协调队列管理器中的发布/预订引擎后，会立即处理 SYSTEM.FTE 队列上的所有消息。

IBM MQ Explorer:

- 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**已启用**。

MQSC:

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

IBM i 传输 IBM i 保存文件时可能发生的错误

如果您使用 Managed File Transfer 多次传输同一个 IBM i 保存文件，那么传输可能失败。

Managed File Transfer 可能产生以下错误中的一个或全部:

- BFGII0003E: Unable to open file "/qsys.lib/library.lib/SAVF.FILE" for reading
- BFGII0082E: A file open for read failed due to a Java IOException with message text "Sharing violation occurred"

如果向 MFT 代理发出多个请求来传输同一个 IBM i 保存文件，那么可能会发生这些错误。如果希望同时多次传输同一个保存文件，那么必须使用多个源代理。请对每个并发传输使用不同源代理。

要使单个源代理来多次传输同一个保存文件，您必须等到前一个传输请求完成，然后再提交每个新传输请求。

配置 Redistributable Managed File Transfer Agent 时的错误消息

无法装入 Windows 的本机库

Windows 您必须在系统上安装以下 Microsoft 库才能使用 Redistributable Managed File Transfer Agent:

- Microsoft Visual C++ Redistributable 2008
- Microsoft Visual C++ Redistributable 2012

这些库可从 Microsoft 中获取。请参阅 [The latest supported Visual C++ downloads](#)。

如果您在未安装这些库的情况下尝试运行 MFT 命令，将报告以下错误:

- BFGUB0070E: Internal error: Native library for platform Windows 7 (architecture amd64) could not be loaded because mqmft (Not found in java.library.path).
- BFGCL0043I: Specify the '-h' command line parameter to see more usage information.

请检查是否安装了 Microsoft 库。如果未安装这些库，请先安装库然后重新运行此命令。

不支持使用绑定模式

Redistributable Managed File Transfer Agent 只能在客户机模式下连接到 IBM MQ。如果您尝试在绑定模式下运行命令，将报告以下错误:

- BFGCL0408E: Unable to obtain IBM MQ installation information for queue manager '*queue manager name*'. Reason Cannot run program "../bin/dspmq": error=2, No such file or directory

发出命令时，您必须提供队列管理器主机、端口、名称和通道名称。

用于设置与消息大小关联的 MQ 属性和 MFT 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 Managed File Transfer 属性以影响 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 Managed File Transfer Agent 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
- **SYSTEM.FTE.STATE.agent_name** 队列的最大消息大小
- 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
- SYSTEM.FTE.STATE.agent_name 队列的最大消息大小
- 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）：

$\text{maxInputOutputMessageLength}$ 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** 该原因码映射到 **MQRC_MSG_TOO_BIG_FOR_Q**，表明已超出 SYSTEM.FTE.STATE.agent_name 队列的最大消息大小值。要解决此问题，请确保 SYSTEM.FTE.STATE.agent_name 队列的最大消息大小大于或等于以下计算结果：

$3 * (\text{maxInputOutputMessageLength}) + 1048576$

- **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) = agentCheckpointInterval * agentFrameSize *
                                           agentWindowSize * 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。

用于将 UAC 和虚拟存储器与 MFT 配合使用的指南

在 Windows Server 2008 R2 和其他类似操作系统中提供了用户帐户控制 (UAC)。这是一个安全基础结构，其功能之一是在中央 Program Files 目录中存储的用户数据转移到用户位置，这称为虚拟存储器。

如果仅使用 Managed File Transfer 工具来管理数据结构，那么 Managed File Transfer 不受 UAC 和虚拟存储器的影响。但是，如果某个非 IBM MQ 管理员使用标准操作系统工具更改或重建了目录结构，那么新结构可能会转移到虚拟存储器。这可能会导致发生以下一种或多种情况：

- 用户（包括 IBM MQ 管理员）无法再在预期位置中看到文件。

- 代理可能无法启动，并报告消息 BFGCL0315，但不提供任何支持性原因码。
- 在代理报告的位置找不到任何日志文件。
- 代理在通过 **-F** 参数启动时可能无法启动，并报告如下消息：

```
The current directory is invalid
```

要更正上述所有情况：

- 以 IBM MQ 管理员身份使用 **fteDeleteAgent** 和 **fteCreateAgent** 命令来重建代理结构。
- 以操作系统管理员身份从受影响用户的虚拟存储器中移除 IBM MQ 条目。例如，在 Windows 上，虚拟存储器的位置如下：`%USERPROFILE%\AppData\Local\VirtualStore\`

相关参考

[第 386 页的『fteDeleteAgent』](#)

fteDeleteAgent 命令将删除 Managed File Transfer Agent 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。

[第 323 页的『fteCreateAgent（创建 MFT 代理）』](#)

fteCreateAgent 命令创建 Managed File Transfer Agent 及其关联的配置。

Windows 用于将 MFT 代理或记录器作为 Windows 服务运行的指南

您可以将 Managed File Transfer 代理、独立数据库记录器和独立文件记录器作为 Windows 服务运行。如果这些 Windows 服务出现任何问题，您可以使用服务日志文件以及本主题中的信息来诊断问题。

要了解有关配置代理、独立记录器或独立文件记录器以作为 Windows 服务运行的更多信息，请参阅[第 128 页的『将 MFT 代理作为 Windows 服务启动』](#)和[第 411 页的『fteModify 记录器\(将 MFT 记录器作为 Windows 服务运行\)』](#)。

注：如果可再发行代理将作为 Windows 服务运行，那么 **BFG_DATA** 环境变量需要在系统环境中进行设置，以便服务正常工作。

日志文件的位置

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

如果在启动代理或作为 Windows 服务的独立记录器时遇到问题，请尝试使用 **-sl** 参数将日志记录级别设置为 `debug`。其他信息会写入 `servicedate.log` 文件。

注：如果记录级别设置为 `debug`，那么运行 Windows 服务所使用的用户帐户和密码会以纯文本形式显示在日志文件中。

日志文件的数目

使用 **fteCreateAgent**、**fteCreateCDAgent**、**fteCreateBridgeAgent**、**fteModifyAgent**、**fteCreateLogger** 或 **fteModifyLogger** 命令将代理程序或独立记录器作为 Windows 服务运行时，可以使用 **-sj** 参数来选择日志文件数。在命令中指定以下文本以更改日志文件数：`-sj -Dcom.ibm.wmqfte.daemon.windows.windowsServiceLogFiles=number`，其中 `number` 是您要表示为正整数的日志文件数。如果不指定日志文件的数目，那么缺省值为 5。

“作为服务登录”权限

您用于运行该服务的 Windows 帐户必须具有 **Log on as a service** 权限。如果您尝试使用 **fteStartAgent**、**fteStartLogger** 命令或 Windows 的 **Sc.exe** 命令启动服务，但是您使用的用户帐户不具有此权限，那么这将打开“服务”窗口。假如您希望启动的服务是用于运行代理，那么此窗口包含以下消息：

```
Unable to start Windows service mqmftAgentAGENT@QMGR.  
System error 1069: The service did not start due to a logon failure.
```

在此消息中，*AGENT* 为代理的名称，*QMGR* 为代理队列管理器名称。如果您尝试作为服务来运行独立记录器，那么将生成一个类似消息，此消息引用的是记录器，而非代理。

要防止发生此错误，请为用于运行服务的 Windows 帐户授予 **Log on as a service** 权限。例如，在 Windows 7 上，完成以下步骤：

1. 从开始菜单，单击管理工具 > 本地安全策略。
2. 在安全性设置窗格中，展开本地策略，然后单击用户权限分配。
3. 在策略和安全性设置窗格中，双击作为服务登录。
4. 单击添加用户或组，然后将您要运行该服务的用户添加到具有 **Log on as a service** 权限的用户列表中。您在运行 **fteCreateAgent**、**fteCreateCDAgent**、**fteCreateBridgeAgent**、**fteModifyAgent**、**fteCreateLogger** 或 **fteModifyLogger** 命令时提供了此用户名。

注：错误 System error 1069: The service did not start due to a logon failure. 也可能由不正确的密码引起。

隐藏您的 Windows 帐户密码

将代理或独立记录器配置为作为 Windows 服务运行时，您将指定要使用的用户名和密码。在以下示例中，将创建代理 AGENT1（此代理具有代理队列管理器 QMGR1）并且配置为作为 Windows 服务运行：

```
fteCreateAgent -agentName AGENT1 -agentQMGr QMGR1 -s -su fteuser -sp ftepassword
```

在此示例中，Windows 服务使用用户名 **fteuser** 来运行，此用户名的关联密码为 **ftepassword**。运行 **fteCreateAgent** 命令或接受 **-s** 参数的其他某个命令时，请以明文形式指定 Windows 帐户的密码。如果偏好于不显示密码，请执行以下步骤：

1. 运行命令 (**fteCreateAgent**、**fteCreateCDAgent**、**fteCreateBridgeAgent**、**fteModifyAgent**、**fteCreateLogger** 或 **fteModifyLogger**) 而不指定 **-sp** 参数。例如：

```
fteCreateAgent -agentName AGENT1 -agentQMGr QMGR1 -s -su fteuser
```

注：此命令会产生一条消息，警告您必须使用 Windows“服务”工具设置密码，然后服务才能成功启动。

2. 打开 Windows“服务”窗口。
3. 在服务列表中，右键单击代理或独立记录器服务，然后选择属性。代理程序服务显示名称为 **Managed File Transfer Agent AGENT @ QMGR**，其中 *AGENT* 是代理程序名称，*QMGR* 是代理程序队列管理器名称。记录器服务显示名称为 **Managed File Transfer Logger for property set coordination_qmgr_name**，其中 *coordination_qmgr_name* 是您为要用作其属性集的独立记录器指定的协调队列管理器。有关属性集合的更多信息，请参阅第 444 页的『[fteStartLogger](#)』和第 411 页的『[fteModify 记录器 \(将 MFT 记录器作为 Windows 服务运行\)](#)』。
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 WebSphere MQ 7.5.0 Fix Pack 1** 或更低版本中的 **Managed File Transfer**）。

不得设置 **JAVA_HOME** 系统环境变量，否则代理或记录器 Windows 服务不大可能启动。代理或记录器 Windows 服务必须与 IBM MQ Java 运行时一起运行。

Windows 用于更新代理或记录器 JVM 选项的指南

如果使用 **fteModifyAgent** 或 **fteModifyLogger** 命令的 **-sj** 参数通过更新，添加或移除 Java 系统属性修改代理或记录器的现有 Windows 服务定义，那么首先删除现有 Windows 服务，然后在其位置创建新服务，然后使用新 Windows 服务的属性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 IBM MQ 9.0.0 Fix Pack 4 开始，在 APAR IT22423 下添加了其他检查，以便对使用 **fteModifyAgent** 或 **fteModifyLogger** 命令的 **-sj** 参数对代理或记录器的 JVM 选项进行的任何更新进行验证，以确保正确指定了这些选项。如果发现属性无效或者无法验证，那么 **fteModifyAgent** 或 **fteModifyLogger** 命令将失败并显示相应的错误消息。

如果 JVM 属性有效并且成功删除现有 Windows 服务，但在 **fteModifyAgent** 或 **fteModifyLogger** 命令创建新的 Windows 服务时出现故障，那么此命令会尝试从代理或记录器属性文件中移除用于定义替换 Windows 服务的属性。在此情况下，会返回一些错误消息，用于说明无法修改代理或记录器，已删除旧的 Windows 服务，但无法创建新的 Windows 服务，因此代理或记录器将无法作为 Windows 服务运行。然后，您必须手动验证 Windows 服务定义的状态与代理或记录器属性文件中定义的 Windows 服务属性是否一致，并执行相应的操作来纠正任何不一致情况。

相关信息

fteModifyAgent: 将 MFT 代理作为 Windows 服务运行

fteModifyLogger: 将 MFT 记录器作为 Windows 服务运行

用于配置 MFT 资源监控器以避免重载代理的指南

您可以配置 Managed File Transfer 资源监视器的属性和参数值以减少代理上的负载。减少代理上的负载可提高该代理的性能。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 WebSphere MQ File Transfer Edition 7.0.4 和更高版本中可用，对于较低版本的 IBM WebSphere MQ File Transfer Edition，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了过多的传输。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 133 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 141 页的『监视 MFT 资源』

您可以监视 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 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 345 页的『fteCreateMonitor: 创建 MFT 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 Managed File Transfer 监视资源（例如，某个目录的内容），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 149 页的『[使用变量替换定制 MFT 资源监视器任务](#)』

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 151 页的『[示例: 资源监视器定义的变量替换](#)』

使用 XML 和 IBM MQ Explorer 对资源监视器定义进行变量替换的示例。

在 Oracle 数据库上更新 MFT 数据库模式期间收到错误时该怎么做

使用 `ftelog_tables_oracle_702_703.sql` 文件将数据库模式更新到最新级别时，可能会收到以下错误消息：`ERROR at line 1: ORA-02289: sequence does not exist`。出现该错误的原因是表所使用的序列和触发器与这些表未处于同一模式中。

关于此任务

要修复该问题，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 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。

MFT 记录器错误处理和拒绝

Managed File Transfer 记录器可识别两种类型的错误：每条消息的错误和通用错误。

每条消息错误可能是由一条或几条个别消息的相关问题导致的。确定为每条消息错误的一些情境示例如下所示：

- 某条消息中缺少结果代码（必需的数据项）
- 某个传输中指定了长度为 3000 个字符且对于关联数据库列过大的作业名
- 接收到有关传输的进度消息，但并没有记录表明传输已启动（可能是因为错误路由或延迟了“传输启动”消息）
- 接收到一条消息，但不是 Managed File Transfer 日志消息

通用错误是每条消息错误之外的所有错误。这些错误的原因可能是配置问题或程序错误。

遇到每条消息的错误时，记录器会将该消息放到拒绝队列上以拒绝该消息。不会向输出日志写入任何内容，因此需要定期检查或持续监视拒绝队列以检测拒绝的消息。

如果连续拒绝了过多消息，而没有任何消息成功写入数据库，那么这种情况将视为通用错误。例如，考虑这样一个站点，它始终使用 10 字符代码作为作业名，但无意间将作业名列重新配置为两个字符宽。尽管过宽的数据通常为每条消息错误，但在此情况下，配置问题是通用的，因此将检测为通用错误。您可以使用 **wmqfte.max.consecutive.reject** 属性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`。在某些上下文中省略（包括 JMS 和 IBM MQ Explorer）。

如果要使用 IBM MQ Explorer，那么可以通过右键单击拒绝队列并单击**浏览消息**来查看该队列的内容。要查看拒绝消息的原因，请双击该消息以打开其属性对话框，然后选择“**已命名属性**”页面。您将看到一个名为 **WMQFTE_ReasonForRejection** 的属性。或者，您也可以编写或配置监视工具以自动获取此信息。

有时，您可能希望重新处理拒绝队列中的消息。在本主题先前描述的示例中，对于数据库中的两字符作业名列，在数据库列宽度增加后可以成功处理这些消息。作为另一示例，如果“传输完成”消息由于缺少关联的“传输启动”而被拒绝，可能会在以后接收到“传输启动”消息。这样，重新处理“传输完成”将会成功。

要重新处理消息，请将这些消息从拒绝队列移动到输入队列。在正常安装中（其中记录器创建了自己的受管预订），输入队列由队列管理器定义并具有类似 `SYSTEM.MANAGED.DURABLE.49998CFF20006204` 的名称。您可以通过查看预订 `SYSTEM.FTE.DATABASELogger.AUTO` 的属性中的**目标名称**或使用以下 MQSC 命令来识别输入队列：

```
DISPLAY SUB(SYSTEM.FTE.DATABASELogger.AUTO) DEST
```

在队列之间移动消息的一种方法是使用 [MA01 SupportPac](#)，例如：

```
q -IFTE.REJECT -oSYSTEM.MANAGED.DURABLE.49998CFF20006204
```

拒绝队列可能包含由于各种原因而被拒绝的消息，其中仅有一部分得以解决。在此情况下，您仍可以重新处理所有消息；现在可接受的消息将被使用，而不能接受的消息则再次移动到拒绝队列。

传输日志中的 `Malformed` 日志消息不是由记录器记录的。这些消息不会视为重要消息，因此这些消息将发送到拒绝队列。有关传输日志消息的更多信息，请参阅第 632 页的『[文件传输日志消息格式](#)』。

在 MFT 记录器已启动但未将任何传输信息记录到数据库时该怎么做


Managed File Transfer 记录器使用的数据库表要求数据库的页面大小为 8KB 或更大。如果数据库的页面大小不足，那么无法正确创建数据库表，并且您将看到 `SQLSTATE=42704` 错误。

如果使用的是 Java Platform, Enterprise Edition 数据库记录器, 那么您可能会在 WebSphere Application Server 系统输出日志中看到以下消息; 或者, 如果使用的是独立数据库记录器, 那么您可能会在 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 556 页的『MFT 数据库记录器表』。
2. 如果表不存在或者不完整, 请检查数据库的页面大小。
3. 如果数据库大小小于 8KB, 请增加数据库页面的大小。
 - 如果数据库位于测试系统上或者数据库中没有数据, 您可以删除表并重新创建一个页面大小大于 8KB 的数据库。
 - 有关如何增加页面大小的信息, 请参阅第 17 页的『在 Windows、UNIX 或 Linux 上增大 Db2 上的日志数据库的页面大小』  或第 19 页的『将 Db2 on z/OS 上的数据库表迁移至 IBM MQ 8.0』。

在 MFT 无法从 AMS 中的密钥库配置文件读取密钥库属性时该怎么做

如果缺省位置中未提供密钥库配置文件, 那么必须由 `MQS_KEYSTORE_CONF` 变量指定该文件的位置, Java AMS 才能够以客户机方式运行。如果未指定此位置, 那么 Managed File Transfer Agent 日志将显示以下错误消息: “Failed to read keystore properties from the keystore configuration file”。

密钥库配置文件的缺省位置为 `home_directory/.mqsc/keystore.conf`。如果密钥库配置文件的位置不是缺省位置, 请完成以下步骤:

1. 以客户机方式启动 FTE 代理。
2. 将 AMS 安全性应用于 `SYSTEM.FTE.DATA.<agent name>` 队列。如果密钥库配置文件不在此位置中, 那么所有传输都将失败, 并且不进行任何确认。
3. 针对 `fteStartAgent` 命令, 将系统变量 `BFG_JVM_PROPERTIES` 设置为 `BFG_JVM_PROPERTIES=-DMQS_KEYSTORE_CONF=path to keystore_config file`。
4. 针对 `fteStartAgent` 命令, 将系统变量 `MQS_KEYSTORE_CONF` 设置为 `MQS_KEYSTORE_CONF=path to keystore_config file`。凡是代理, 不管其运行方式如何, 必须设置此变量才能确保可运行。

注: 如果 Java AMS 以绑定方式运行, 并且密钥库配置文件不在缺省位置中, 那么队列管理器的错误日志中将显示错误 AMQ9062。

当由于本机内存耗尽导致 MFT 代理异常终止且显示 `java.lang.OutOfMemoryError` 时该怎么做

在处理大量受管传输请求时, 例如, 文件至文件传输、消息至文件传输或文件至消息传输, 代理异常结束 (异常终止), 报告 `java.lang.OutOfMemoryError`, 并且此时未完全利用全部 RAM 内存。此异常是本机内存耗尽所导致的。

诊断问题

发生此问题时, 受影响的代理异常终止并生成两个文件, 其中提供有关根本原因的详细信息:

- ABEND 文件。此文件的名称符合命名约定 `ABEND.FTE.date_timestamp.identifier.log`。

 在多平台上, 此文件将写入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/ffdc` 目录。

z/OS 在 z/OS 上，此文件将写入 USS 位置 \$BFG_CONFIG/mqft/logs/
coordination_qmgr_name/agents/agent_name/logs/ffdc

- Javacore 文件。此文件的名称具有以下格式：
javacore.datestamp.timestamp.pid.identifier.txt

Multi 在多平台上，此文件将写入 MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/
agents/agent_name 目录。

z/OS 在 z/OS 上，该文件将写入 USS 位置 \$BFG_CONFIG/mqft/logs/
coordination_qmgr_name/agents/agent_name 目录。

ABEND 和 Javacore 对包含类似于以下所示示例的信息：

示例：第 1 对

Abend 文件

```
Filename:
C:\ProgramData\IBM\MQ\mqft\logs\COORDQM\agents\AGENT1\logs\ffdc\ABEND.FTE.20200109113518046.1764802189777
906538.log
Level: p900-005-180821
Time: 09/01/2020 11:35:18 GMT
Thread: 96 (TransferSender[414d51204d44424b5250303720202045fbd6532ebfaa02])
Class: com.ibm.wmqfte.thread.FTETHread
Instance: 55b455b4
Method: uncaughtException
Probe: ABEND_001
Cause: java.lang.OutOfMemoryError: native memory exhausted

java.lang.OutOfMemoryError: native memory exhausted
at com.ibm.mq.jmqi.local.internal.base.Native.MQPUT(Native Method)
at com.ibm.mq.jmqi.local.LocalMQ.MQPUT(LocalMQ.java)
at com.ibm.wmqfte.wmqiface.WMQQueueImpl.put(WMQQueueImpl.java)
at com.ibm.wmqfte.wmqiface.WMQQueueImpl.put(WMQQueueImpl.java)
at com.ibm.wmqfte.transfer.impl.TransferSenderRunnable.doTransfer(TransferSenderRunnable.java)
at com.ibm.wmqfte.transfer.impl.TransferSenderRunnable.run(TransferSenderRunnable.java)
at java.lang.Thread.run(Thread.java)
at com.ibm.wmqfte.thread.FTETHread.run(FTETHread.java)
```

Javacore 文件

```
NULL -----
0SECTION TITLE subcomponent dump routine
NULL =====
1TISIGINFO Dump Event "systhrow" (00040000) Detail "java/lang/OutOfMemoryError" "native memory
exhausted" received
1TIDATETIME Date: 2020/01/09 at 11:35:18
1TIFILENAME Javacore filename:
C:\ProgramData\IBM\MQ\mqft\logs\COORDQM\agents\AGENT1\javacore.20200109.113518.14148.0002.txt
```

示例：第 2 对

ABEND 文件

```
Filename:
C:\ProgramData\IBM\MQ\mqft\logs\COORDQM\agents\AGENT1\logs\ffdc\ABEND.FTE.20200109143700286.3177895731698
464509.log
Level: p900-005-180821
Time: 09/01/2020 14:37:00:286 GMT
Thread: 918 (AgentStatusPublisher)
Class: com.ibm.wmqfte.thread.FTETHread
Instance: bc10bc1
Method: uncaughtException
Probe: ABEND_001
Cause: java.lang.OutOfMemoryError: Failed to create a thread: retVal -1073741830, errno 12

java.lang.OutOfMemoryError: Failed to create a thread: retVal -1073741830, errno 12
at java.lang.Thread.startImpl(Native Method)
at java.lang.Thread.start(Thread.java)
```

Javacore 文件

```
NULL -----
0SECTION TITLE subcomponent dump routine
NULL =====
1TISIGINFO Dump Event "systhrow" (00040000) Detail "java/lang/OutOfMemoryError" "Failed to create a
thread: retVal -1073741830, errno 12" received
1TIDATETIME Date: 2020/01/09 at 14:37:00
1TIFILENAME Javacore filename: C
C:\ProgramData\IBM\MQ\mqft\logs\C00RDQM\agents\AGENT1\javacore.20200109.143700.2652.0003.txt
```

发生此问题的原因

由于运行代理的系统上的本机堆内存耗尽而发生此问题。

有关 Java 堆内存与本机堆内存之间的区别的更多信息，请参阅 [第 618 页的『MFT 代理程序如何使用 Java 堆和本机堆内存』](#)。

避免问题

您可以执行一些操作，以帮助减少由于本机内存耗尽而导致 MFT 代理程序由于 `java.lang.OutOfMemoryError` 而停止的可能性：

1. 减小运行 MFT 代理的 JVM 的 Java 堆的大小。

分配的 Java 堆大小越大，可用于本机堆的内存越少。减少代理使用的 Java 堆的大小可释放更多内存用于本机堆。

缺省情况下，代理的 Java 堆设置为 512 MB。如果已将此项更改为更大的值，请考虑减小此项，并使用类似生产的工作负载进行测试。

- 要在将代理作为正常进程运行时降低或更改 Java 堆，请执行以下操作：

设置 `BFG_JVM_PROPERTIES` 环境变量以将选项目录传递到 JVM。例如，在 Windows 上，要将最大堆大小设置为 1024 MB，请在使用 `fteStartAgent` 命令之前运行以下命令：

```
set BFG_JVM_PROPERTIES="-Xmx1024M"
```

有关如何使用 `BFG_JVM_PROPERTIES` 环境变量设置 Java 系统属性的更多信息，请参阅 [MFT 的 Java 系统属性](#)。

- 要在将代理作为 Windows 服务运行时降低或更改 Java 堆，请执行以下操作：

要将选项传递到将代理程序作为 Windows 服务运行的 JVM，请使用 `fteModifyAgent` 命令中指定的 `-sj` 参数修改代理程序。

以下示例使用 `fteModifyAgent` 命令和 `-sj` 参数，针对运行 Windows 服务配置的代理的 JVM 设置 Java 堆的最大大小：

```
fteModifyAgent.cmd -agentName AGENT1 -s -su user1 -sp passw0rd -sj -Xmx1024M
```

您可以通过在代理重新启动后复查代理的 `output0.log` 文件，检查是否已成功设置此项。在 `Start Display Current Environment` 部分中，将报告值 1024 MB，如下所示：

```
The maximum amount of memory that the Java virtual machine will attempt to use is: '1024'MB
```

2. 限制本机内存使用

通常，如果代理使用 `BINDINGS` 传输连接到其代理队列管理器，那么会看到本机堆耗尽所导致的 `java.lang.OutOfMemoryErrors`。在已将代理配置为使用 `BINDINGS` 传输时，只要需要与队列管理器通信，代理就会调用本机方法。

这意味着，由于队列管理器连接的增加和消息通信的增加，本机内存使用量随代理工作负载的增加而增加。在此情况下，减少工作负载可能很有用。要执行此操作，请将以下代理程序属性设置为小于缺省值 25 的值：

- **maxSourceTransfers**
- **maxDestinationTransfers**

这会减少可发生的并行传输的数量，因此降低代理的最大并行工作负载。

3. 配置代理以在连接到其代理队列管理器时使用 CLIENT 传输。可以通过设置以下代理属性来执行此操作：

- **agentQMgrHost**
- **agentQMgrPort**
- **agentQMgrChannel**

您可以在 [MFT agent.properties](#) 文件主题中查找有关这些属性的信息。

这确保代理与队列管理器之间的所有通信均通过 TCP/IP 进行，而不是本机代码，因此减少了代理所使用的本机内存量。

要点: 执行此操作也会降低性能。使用与本地主机的 TCP/IP 连接，而非本机代码的情况下，当代理查询与队列管理器的交互时，此配置效率不高。

BFGSS0023E 错误以及如何避免这些错误

如果从安装中卸载修订包以返回至产品的先前版本，并在发生卸载时受管传输涉及了与安装相关联的代理，那么该代理无法启动，并将报告 BFGSS0023E 错误。要避免此错误，您可以完成一些步骤以防止在重新启动代理时显示 BFGSS0023E 消息。

对于当前涉及代理的正在进行中的每个受管传输，在代理的 `SYSTEM.FTE.STATE.agent_name` 队列上有一条消息。此消息在受管传输上存储检查点信息，并且在受管传输进入恢复时使用。在受管传输完成后，将除去 `SYSTEM.FTE.STATE.agent_name` 队列上的相应消息。

每条状态消息都包含一些内部头信息，指示在受管传输运行时代理所使用的 Managed File Transfer 组件版本。版本信息会显示具体的修订包级别，因此，举例来说，如果 IBM MQ 8.0.0 Fix Pack 5 代理在运行受管传输，那么该受管传输的状态消息将包含对 IBM MQ 8.0.0 Fix Pack 5 的引用。

如果从安装中卸载修订包，并且与该安装相关联的代理关联了正在进行的传输，那么该代理无法启动，并将报告以下错误：

```
BFGSS0023E: The agent is configured to use IBM MQ queues that contain data created using a later version of the product. The agent cannot run in this configuration and will end.
```

例如，如果 IBM MQ 8.0.0 Fix Pack 5 代理在停止并降级到 IBM MQ 8.0.0 Fix Pack 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 IBM MQ 8.0.0 Fix Pack 5 时写入的。由于代理现在使用的是 IBM MQ 8.0.0 Fix Pack 4，因此它会报告先前段落中描述的 BFGSS0023E 错误并关闭自身。

作为一般规则，如果您要除去任一 Managed File Transfer 组件的修订包，那么完成以下步骤应该会防止在重新启动代理时显示 BFGSS0023E 消息：

1. 确保其所有代理已完成其受管传输。
2. 停止这些代理。
3. 除去修订包。
4. 重新启动这些代理。

相关任务

[第 128 页的『启动 MFT 代理』](#)

必须先启动 Managed File Transfer 代理，才能将其用于文件传输。

相关参考

[第 518 页的『MFT 代理队列设置』](#)

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](#)

在 UNIX 上将队列管理器还原为先前版本
在 Windows 上将队列管理器还原为先前版本

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，该传输项被标记为失败，并返回 BFGIO0341E 错误。

避免此问题

可以通过两种方法来防止发生 BFGIO0341E 错误：

- 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**。有关该属性的更多信息，请参阅 MFT agent.properties 文件。

MFT 的返回码

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答复命令状态。缺省情况下，该超时对于受管调用和传输命令而言是无限的。例如，当您使用 fteCreateTransfer 命令指定 -w 参数时。缺省情况下，该超时对于其他命令是 5 秒。
3	确认超时	代理在指定的超时内未确认接收到命令。缺省情况下，该超时是 5 秒。
4	错误的代理	将命令发送到了错误的代理。在命令 XML 中指定的代理不是正在读消息所在命令队列的代理。
20	传输部分成功	传输已完成，但只有部分成功，传输了一些文件。
21	传输已停止	传输已被一个用户出口停止。
22	取消传输超时	代理已收到取消传输的请求，但该取消未能在 30 秒内完成。未取消该传输。
26	未找到取消标识	代理已收到取消传输的请求，但未找到该传输。这可能是由于代理处理该取消请求之前，该传输已完成。这也可能是因为，您向 fteCancelTransfer 命令提供的传输标识不正确。已忽略该取消请求。
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。

表 24: 返回码 (继续)

返回码	短名称	描述
44	已超出源代理容量	源代理没有足够的容量来执行该传输。
45	已超出目标代理容量	目标代理没有足够的容量来执行该传输。
46	已超出源代理最大文件数	要传输的文件数已超出源代理的限制。
47	已超出目标代理最大文件数	传输的文件数已超出目标代理的限制。
48	无效的日志消息属性	日志消息的格式不正确。这是内部错误。如果您收到该返回码, 请与 IBM 支持中心联系以获取进一步的帮助。
49	目标不可达	由于出现 IBM MQ 问题, 源代理不能将消息发送到目标代理。例如, 如果源代理队列管理器未正确配置以与目标代理队列管理器进行通信, 将发生该问题。
50	试用版违例	试用版代理尝试与非试用版代理进行通信。
51	未准许源传输	<code>maxSourceTransfers</code> 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	<code>maxDestinationTransfers</code> 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 <code>authorityChecking</code> 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。
64	两个队列都不被允许	传输的源和目标是队列。
65	一般数据队列错误	访问 Managed File Transfer Agent 数据队列时发生错误。

表 24: 返回码 (继续)

返回码	短名称	描述
66	数据队列 put 权限错误	访问 Managed File Transfer Agent 数据队列时发生错误。未启用 Advanced Message Security。
67	数据队列 put AMS 错误	访问 Managed File Transfer Agent 数据队列时发生权限错误。已启用 Advanced Message Security。
V 9.0.1 V 9.0.1 69	传输恢复发生超时	在指定的 transferRecoveryTimeout 值后, 恢复传输的操作发生超时。
70	代理已异常结束	应用程序发生不可恢复的问题并且被强制终止。
75	队列管理器不可用	应用程序无法继续, 因为应用程序的队列管理器不可用。
78	启动配置存在问题	由于启动配置数据存在问题, 因此应用程序无法继续。
85	数据库服务器存在问题	由于数据库存在问题 (通常仅由记录器返回), 因此应用程序无法继续。
100	监视器替换无效	监视器任务 XML 脚本内的变量替换的格式不正确。
101	监视器资源错误	监视器资源定义数无效。
102	监视器触发器错误	监视器触发器定义数无效。
103	监视器任务错误	监视器任务定义数无效。
104	监视器缺失	请求的监视器不存在。
105	监视器已存在	请求的监视器已存在。
106	监视器用户出口错误	监视器用户出口已在资源监视器轮询期间生成错误。
107	监视器用户出口已取消	监视器用户出口已请求取消事务。
108	监视器任务失败	由于在处理任务时出错, 监视器任务未能完成。
109	监视器资源失败	无法将监视器资源定义应用到指定资源。
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。

表 24: 返回码 (继续)

返回码	短名称	描述
112	监视器任务源队列管理器无效	监视器传输任务的源代理队列管理器与资源监视器的代理队列管理器不匹配。
113	不支持监视器	已尝试在协议网桥代理上创建或删除资源监视器。该行为不受支持。
114	监视器资源被拒绝	监视器资源扫描的目录拒绝访问。
115	监视器资源队列正在使用中	监视器资源队列已打开，但不能使用共享访问权进行输入。
116	未知监视器资源队列	监视器的关联队列管理器上不存在此监视器资源队列。
118	监视器资源表达式无效	对 XPath 表达式求值时发生错误。对 XPath 表达式求值，以访问消息头中用户定义的属性。消息位于资源监视器监视的队列上。
119	监视器任务源代理队列管理器缺失	监视器任务定义中缺少源代理名称或源代理队列管理器名称。
120	监视器队列未启用	监视器资源队列未启用。
121	访问监视队列时发生意外错误	访问监视资源队列时发生了意外错误。
122	未对上下文标识启用监视器命令队列	未对设置上下文标识启用监视器代理命令队列。

下表列出了产品中间回复代码及其含义：

表 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，Managed File Transfer 命令仅返回最终回复代码。

相关参考

第 285 页的『针对传输中文件的返回码』

传输中的各个文件都具有自己的结果代码，这些代码的含义不同于命令的总体返回码。

针对传输中文件的返回码

传输中的各个文件都具有自己的结果代码，这些代码的含义不同于命令的总体返回码。

在将 <action> 元素设置为“progress”值的传输日志进度消息中，报告的每个文件都有一个带有 resultCode 的 <status> 元素。例如：



```
<action time="2009-11-23T21:28:09.593Z">progress</action>
...
  <status resultCode="1">
    <supplement>BFGI00006E: File &quot;C:\destinationfiles\dest1.doc&quot;
      already exists.</supplement>
  </status>
```

下表描述了 resultCode 的可能值：

结果代码值	描述
0	成功。文件传输成功。
1	失败。文件传输失败。请参阅 <supplement> 元素，以获取该错误的更多详细信息。
2	警告。文件已传输，但报告了一条警告消息。例如，虽然源处置设置为删除，但无法删除源文件。请参阅 <supplement> 元素，以获取该警告的更多详细信息。

对 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 286 页的『跟踪 Connect:Direct 网桥』](#)
- [第 287 页的『Connect:Direct 网桥的日志信息』](#)
- [第 287 页的『解决 Connect:Direct 节点的许可权问题』](#)
- [第 288 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 288 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 289 页的『使用双正斜杠指定 Connect:Direct 文件路径』](#)
- [第 289 页的『增加 Connect:Direct 网桥的并行传输数』](#)
- [第 290 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 460 页的『MFT agent.properties 文件』

每个 Managed File Transfer Agent 都有自己的属性文件 agent.properties，该文件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 MFT 代理和 Connect:Direct 节点之间传输文件。有关这些传输中涉及的 Connect:Direct 节点和进程的日志信息都显示在 IBM MQ Explorer 插件中，并存储在您的日志数据库内。

Connect:Direct 网桥代理必须是 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本。该传输中涉及的其他代理可以是任意版本的 Managed File Transfer。但是，要记录有关 Connect:Direct 节点和进程的信息，传输中涉及的所有 MFT 代理都必须是 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本。要在 IBM MQ Explorer 插件中显示该信息，该插件必须为 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本。要在日志数据库中存储该信息，数据库记录器和数据库模式必须是 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本。

有关文件传输中涉及的 Connect:Direct 节点和 Connect:Direct 进程的日志信息，包含在发布到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 632 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 网桥节点名的值是 MFT Connect:Direct 网桥代理所识别的网桥节点的名称。主节点名称和辅助节点名称是用于引用 Connect:Direct 网桥节点的网络映射中的节点的名称。

相关参考

第 650 页的『Connect:Direct 网桥传输日志消息示例』

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 Managed File Transfer 和 Connect:Direct 之间的传输失败，并且出现关于没有足够许可权的错误，请使用本主题中的信息。

对于涉及 Connect:Direct 网桥的传输，连接到 Connect:Direct 节点的用户标识由与传输请求相关联的 IBM MQ 消息描述符 (MQMD) 用户标识来确定。您可以将特定的 MQMD 用户标识映射到特定的 Connect:Direct 用户标识。有关更多信息，请参阅第 120 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 120 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 MFT 代理和 Connect:Direct 节点之间传输文件时，会执行代码页转换和行尾字符转换。该传输使用 Connect:Direct 网桥节点的网络映射中的操作系统信息来确定远程节点的行尾字符。如果网络映射中的信息不正确，那么行尾字符转换可能未正确执行。

确保 Connect:Direct 网桥节点和用作传输目标的任意 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。
 - 如果远程节点在 UNIX 或 Linux 系统上，请选择 UNIX。
 -  如果远程节点在 z/OS 系统上，那么请选择 OS/390。

Connect:Direct 网桥不支持到其他操作系统上的远程节点的传输。

- 确保对于作为文件传输目标或源的每个远程节点，您在 Connect:Direct 网桥代理配置目录下的 `ConnectDirectNodeProperties.xml` 文件中指定了远程 Connect:Direct 节点的操作系统类型。有关更多信息，请参阅第 120 页的『[配置 ConnectDirectNodeProperties.xml 文件以包含有关远程 Connect:Direct 节点的信息](#)』和 第 490 页的『[Connect:Direct 节点属性文件格式](#)』。

相关参考

第 538 页的『[在 Connect:Direct 和 MFT 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 Managed File Transfer Agent 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上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 286 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。

相关参考

第 530 页的『与 Connect:Direct 节点之间传输数据集』

您可以使用 Connect:Direct 网桥在 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)。Connect:Direct 网桥 (其中使用缺省值 **maxSourceTransfers** 和 **maxDestinationTransfers** 配置了代理) 在任何时候最多可以处理 10 个传输: 5 个传输 (其中代理是源)，5 个传输 (其中代理是目标)。

这些缺省值可确保 Connect:Direct 网桥代理不超出到 Connect:Direct 节点的最大 API 连接数。使用缺省配置的 Connect:Direct 网桥代理最多使用 10 个到 Connect:Direct 节点的 API 连接。UNIX 上的 Connect:Direct 节点接受的最大连接数由 **api.max.connects** Connect:Direct 参数控制。对于 Windows 上的 Connect:Direct 节点，等效参数为 **max.api.connects**。

如果 Connect:Direct 网桥执行大量文件传输的速率不够，那么可以增加 Connect:Direct 网桥代理处理的并行传输数。针对 Connect:Direct 网桥代理更改以下代理属性：

maxSourceTransfers

将该属性设置为大于 5 且小于等于 25 的值。如果选择大于 25 的值，那么除非增加可用于代理使用的 JVM 的内存量，否则该代理将耗尽内存。

maxDestinationTransfers

将该属性设置为大于 5 且小于等于 25 的值。如果选择大于 25 的值，那么除非增加可用于代理使用的 JVM 的内存量，否则该代理将耗尽内存。

ioThreadPoolSize

ioThreadPoolSize 的缺省值为 10。该属性限制 Connect:Direct 网桥代理充当源代理的传输的 Connect:Direct 节点 API 连接数。这些传输从 Connect:Direct 到 Managed File Transfer。使用以下指导信息来设置该属性的值：

- 如果 **maxSourceTransfers** 的值小于 **maxDestinationTransfers** 的值，请将 **ioThreadPoolSize** 设置为 **maxSourceTransfers** 值的 2 倍或 10 (两者中选用较大值)
- 如果 **maxSourceTransfers** 的值大于 **maxDestinationTransfers** 的值，请将 **ioThreadPoolSize** 设置为 **maxSourceTransfers** 和 **maxDestinationTransfers** 的总和

除这些代理属性外，还必须更改作为 Connect:Direct 网桥的一部分的 Connect:Direct 节点的最大并行 API 连接数。控制此数字的 Connect:Direct 参数为 **api.max.connects** (如果节点在 UNIX 上) 或 **max.api.connects** (如果节点在 Windows 上)。对相应的参数执行以下更改：

api.max.connects (如果 **Connect:Direct** 网桥中的节点在 **UNIX** 上)

将此参数设置为大于 **maxSourceTransfers** 和 **maxDestinationTransfers** 之和的值。

api.max.connects 参数的缺省值为 16。有关如何设置该参数的更多信息，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 118 页的『[配置 **Connect:Direct** 网桥](#)』

配置 **Connect:Direct** 网桥，以在 **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 **Managed File Transfer** 代理。该代理称为 **Connect:Direct** 网桥代理。

相关参考

第 460 页的『[MFT agent.properties 文件](#)』

每个 **Managed File Transfer Agent** 都有自己的属性文件 **agent.properties**，该文件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到 **Connect:Direct** 网桥代理配置目录中的 **output0.log** 文件中。

关于此任务

要配置 **Connect:Direct** 进程的日志记录，请完成以下步骤：

过程

1. 停止 **Connect:Direct** 网桥代理。
2. 编辑 **MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name** 目录中的 **agent.properties** 文件以包含 **logCDProcess** 属性。
logCDProcess 属性可具有以下某个值：
 - **None** - 未记录任何信息。这是缺省值。
 - **Failures** - 记录了有关 **Connect:Direct** 进程失败的信息。
 - **All** - 记录了有关所有 **Connect:Direct** 进程的信息。
3. 启动 **Connect:Direct** 网桥代理。

结果

有关 **Connect:Direct** 进程的信息记录在 **Connect:Direct** 网桥代理的 **output0.log** 文件中。记录的信息包括：

- **MFT** 传输标识
- **Connect:Direct** 进程名称
- **Connect:Direct** 进程号
- 生成的进程定义
- 如果 **Connect:Direct** 进程是用户定义的，那么还包含进程模板的文件名

相关概念

第 286 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 460 页的『MFT agent.properties 文件』

每个 Managed File Transfer Agent 都有自己的属性文件 agent.properties，该文件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。

MFT 引用

MFT 安全性参考

IBM MQ 中 MFT 的文件系统许可权

安装和配置 IBM MQ 的 Managed File Transfer 组件时，将使用以下许可权创建 configuration, installations 和 logs 目录。

UNIX 和 Linux



表 27: UNIX 和 Linux 上目录的许可权摘要

目录	许可权
/var/mqm/mqft/config	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 可由 mqm 组写入• 全局可读
/var/mqm/mqft/logs	全局可读写

Windows



表 28: Windows 上目录的许可权摘要

目录	许可权
MQ_DATA_PATH\mqft\config	以下用户具有完全的读写访问权： <ul style="list-style-type: none">• 管理员• 系统帐户• mqm 组 其他用户具有读访问权
MQ_DATA_PATH\mqft\installations	以下用户具有完全的读写访问权： <ul style="list-style-type: none">• 管理员• 系统帐户• mqm 组 其他用户具有读访问权

表 28: Windows 上目录的许可权摘要 (继续)

目录	许可权
MQ_DATA_PATH\mqft\logs	以下用户具有完全的读写访问权: <ul style="list-style-type: none"> • 管理员 • 系统帐户 • mqm 组 其他用户具有读写访问权

z/OS



表 29: z/OS 上目录的许可权摘要

目录	许可权
DATA_PATH/mqft/config	<ul style="list-style-type: none"> • 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
DATA_PATH/mqft/installations	<ul style="list-style-type: none"> • 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
DATA_PATH/mqft/logs	全局可读写

注: DATA_PATH 从环境变量 BFG_DATA 派生。

管理特定于 MFT 资源的权限

针对任何文件传输请求，Managed File Transfer 代理进程都需要其本地文件系统的某种访问权级别。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

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 293 页的『限制特定于 MFT 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 298 页的『限制 MFT 代理操作的用户权限』，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与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 7.0.2 的 Managed File Transfer 版本迁移到 IBM WebSphere MQ 7.5 或更高版本，并保留现有代理程序配置，那么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(99999999) 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 为 IBM MQ 对象创建和修改权限记录。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 293 页的『限制特定于 MFT 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 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 298 页的『限制 MFT 代理操作的用户权限』

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执行的 Managed File Transfer 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 301 页的『MFT 记录器的权限』

运行记录器的操作系统用户需要对记录器队列和 SYSTEM.FTE 主题具有特定 IBM MQ 权限。

限制特定于 MFT 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 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：

ULW 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
```

IBM i 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)
```

z/OS 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：

ULW 对于 UNIX、Linux 和 Windows 系统：

```
setmqaut -m agent_queue_manager -t qmgr -g FTEAGENT +connect +inq +setid
```

IBM i 对于 IBM i：

```
GRTMQMAUT OBJ('agent_queue_manager') OBJTYPE(*MQM) USER(FTEAGENT) AUT(*CONNECT)
```

z/OS 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 304 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：

ULW 对于 UNIX、Linux 和 Windows 系统：

```
setmqaut -m QM1 -n SYSTEM.FTE.COMMAND.agent_name -t queue -g FTEUSER +put
```

IBM i 对于 IBM i：

```
GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEUSER) AUT(*PUT)
MQMNAME('QM1')
```

z/OS 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：

ULW 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
```

IBM i 对于 IBM i：

```
GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*PUT)
MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*GET)
MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*SETID)
MQMNAME('QM1')
```

z/OS 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 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：

ULW 对于 UNIX、Linux 和 Windows 系统：

```
setmqaut -m QM1 -n SYSTEM.FTE.DATA.agent_name -t queue -g FTEAGENT +put +get +inq
```

IBM i 对于 IBM i：

```
GRTMQMAUT OBJ('SYSTEM.FTE.DATA.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*PUT)
MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.FTE.DATA.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*GET)
MQMNAME('QM1')
```

z/OS 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](#)。

用于运行命令和 IBM MQ Explorer 的权限

管理命令 (例如 **fteStartAgent** 命令) 和 IBM MQ Explorer 的 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：

ULW 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
```

IBM i 对于 IBM i:

```
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*Q) USER(FTEAGENT) AUT(*PUT) MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*Q) USER(FTEAGENT) AUT(*GET) MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEAGENT) AUT(*PUB) MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEAGENT) AUT(*SUB) MQMNAME('QM1')
```

z/OS 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：

ULW 对于 UNIX、Linux 和 Windows 系统:

```
setmqaut -m QM1 -n SYSTEM.FTE -t topic -g FTEUSER +sub
```

IBM i 对于 IBM i:

```
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEUSER) AUT(*SUB) MQMNAME('QM1')
```

z/OS 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：

ULW 对于 UNIX、Linux 和 Windows 系统:

```
setmqaut -m QM1 -n SYSTEM.DEFAULT.MODEL.QUEUE -t queue -g FTEUSER +dsp +put +get +browse
```

IBM i 对于 IBM i:

```
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(FTEUSER) AUT(*ADM DSP)
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')
GRTRMQAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(FTEUSER) AUT(*BROWSE)
MQMNAME('QM1')
```

z/OS 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：

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	本地主题	SUBSCRIBE	PUBLISH 和 SUBSCRIBE
SYSTEM.DEFAULT.MODEL.QUEUE (或 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 IBM 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 IBM 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），IBM 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，浏览，放置
- 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 304 页的『MFT 命令和进程与队列管理器的连接关系』

相关参考

第 298 页的『限制 MFT 代理操作的用户权限』

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执行的 Managed File Transfer 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 301 页的『MFT 记录器的权限』

运行记录器的操作系统用户需要对记录器队列和 SYSTEM.FTE 主题具有特定 IBM MQ 权限。


限制 MFT 代理操作的用户权限

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执行的 Managed File Transfer 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`。有关更多信息，请参阅第 460 页的『MFT `agent.properties` 文件』。
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。

 在 z/OS 平台上，运行代理的用户必须对可请求许可权的用户标识具有 ALT_USER 权限才能执行代理操作。


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：   从 IBM MQ 9.0.0 Fix Pack 3 和 IBM MQ 9.0.4 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 31: 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	Managed File Transfer 访问权限	权限队列	IBM MQ 访问权限 (多平台)	RACF 访问级别 (仅限 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
对此代理程序运行受管调用				
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	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
取消由启动传输的用户或组创建的传输或者由其他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用于启动传输的资源监视器或调度传输, 用户需要 **监视器** 或 **调度** 权限以及 **传输源** 和 **传输目标** 权限。

V 9.0.0.3 - **V 9.0.4** 从 IBM MQ 9.0.0 Fix Pack 3 和 IBM MQ 9.0.4 开始, 当代理收到取消文件传输的请求时执行权限检查的顺序已发生更改, 此时代理首先检查请求取消的用户是否为启动传输的用户。如果取消传输的用户并非请求传输的用户, 那么代理会检查取消的用户是否具有 SYSTEM.FTE.AUTHOPS1.agent_name 队列的浏览许可权。更改检查顺序可避免在请求文件传输的用户与请求取消的用户为同一用户的情况下, 代理和队列管理器错误日志中出现意外错误。

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 32: 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	Managed File Transfer 访问权限	权限队列	IBM MQ 访问权限 (多平台)	RACF 访问级别 (仅限 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 destination_agent	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

z/OS `setmqaut` 命令不能在 z/OS 系统上使用。对于 z/OS，使用 RACF 代替。请参阅在 [z/OS 上设置安全性](#) 以获取更多信息。

IBM i `setmqaut` 命令不能在 IBM i 系统上使用。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 690 页的『[针对安全性的 MFT 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。

有关更多信息，请参阅第 460 页的『MFT agent.properties 文件』。

相关参考

第 293 页的『限制特定于 MFT 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 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 301 页的『MFT 记录器的权限』

运行记录器的操作系统用户需要对记录器队列和 SYSTEM.FTE 主题具有特定 IBM MQ 权限。

第 445 页的『fteStopAgent』

使用 **fteStopAgent** 命令以受控方式停止 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。

MFT 记录器的权限

运行记录器的操作系统用户需要对记录器队列和 SYSTEM.FTE 主题具有特定 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 293 页的『限制特定于 MFT 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 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 298 页的『限制 MFT 代理操作的用户权限』

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执行的 Managed File Transfer 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。

用于访问敏感配置信息的 MFT 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库和密钥库文件。

如果某个 Managed File Transfer 进程检测到某个配置文件包含敏感信息，是密钥库或信任密钥库文件，并且具有系统范围的读、写或删除许可权，那么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更改，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 MFT 代理日志和状态消息的权限

Managed File Transfer 代理可发出在协调队列管理器上发布的各种日志、进度和状态消息。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[以保护开头的部分](#)。

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。消息使用又名为 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。

供 MFT 用于访问文件系统的权限

针对任何文件传输请求，Managed File Transfer 代理进程都需要其本地文件系统的某种访问权级别。

- 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 39 页的『MFT 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 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 213 页的『使用用户出口定制 MFT』。

commandPath MFT 属性

使用 `commandPath` 属性以指定 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 Managed File Transfer 的受管传输和受管调用功能运行。请参阅[程序调用](#)以了解更多信息。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，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
```

z/OS 或者，仅针对 z/OS 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。
- **z/OS** `command_directory_name_or_data_set_name_prefix` 是可以运行的命令的 z/OS UNIX 系统服务目录路径，或以 `//` 开头的数据集名称前缀。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。

UNIX 例如，在 UNIX 系统上，如果要运行位于目录 `/home/user/cmds1` 和 `/home/user/cmds2` 中的命令，请按如下设置 `commandPath` 代理属性：

```
commandPath=/home/user/cmds1:/home/user/cmds2
```

Windows 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 (`/`)。

z/OS 例如，在 z/OS 上，如果要运行满足以下条件的命令：

- 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.CMDS'` 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.CMDS()'
```

要点：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。

V9.0.0.1 您可以覆盖此行为以兼容以下发行版：

- IBM WebSphere MQ File Transfer Edition.
- IBM WebSphere MQ 7.5.0 Fix Pack 1 Managed File Transfer 组件（或更低版本）。
- 未设置安装属性 `enableFunctionalFixPack=7502` 的安装上的 IBM WebSphere MQ 7.5.0 Fix Pack 2 Managed File Transfer 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
```

V9.0.0.1 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。

相关参考

第 460 页的『MFT `agent.properties` 文件』

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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。

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
- IBM 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 IBM MQ Explorer Managed File Transfer 插件在运行时连接到多个队列管理器。

下表汇总了代理、记录器、命令和 IBM 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。

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
<code>fteAnt</code>				

表 33: 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
fteCancelTransfer		X		
fteChangeDefaultConfigurationOptions				
fteCleanAgent	X			
fteCreateAgent	X			
fteCreateBridgeAgent	X			
fteCreateCDAgent	X			
 fteCreate 环境				
fteCreateLogger				
fteCreateMonitor		X		
fteCreateTemplate			X	
fteCreateTransfer		X		
fteDefine				
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				
 fteSetAgentLog 级别				
fteSetAgentTraceLevel				
fteSetLoggerTraceLevel				
  fteSetProductID				
fteSetupCommands				
fteSetupCoordination				
fteShowAgentDetails			X	

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
fteShowLoggerDetails				
fteStartAgent				
fteStartLogger				
fteStopAgent		X		
fteStopLogger		X		

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 IBM 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](#)。

相关信息

[已安装的 MFT 命令集](#)

MFT 命令连接到哪个队列管理器的详细信息

有关 Managed File Transfer 命令连接到哪个队列管理器的更多信息。

本主题将展开 [第 304 页的『MFT 命令和进程与队列管理器的连接关系』](#) 中的信息以及一些插图。

连接到协调队列管理器的命令

以下命令连接到协调队列管理器:

- [fteCreateTemplate](#)
- [fteDeleteAgent](#)
- [fteDeleteTemplates](#)
- [fteListAgents](#)
- [fteListMonitors](#)
- [fteListScheduledTransfers](#)
- [fteListTemplates](#)
- [fteShowAgentDetails](#)

Managed File Transfer 拓扑的协调队列管理器是了解整个拓扑的中心集线器。协调队列管理器通过发送方和接收方通道连接到拓扑中的所有代理队列管理器。代理定期向协调队列管理器发布状态信息，并将其传输模板存储在其中。

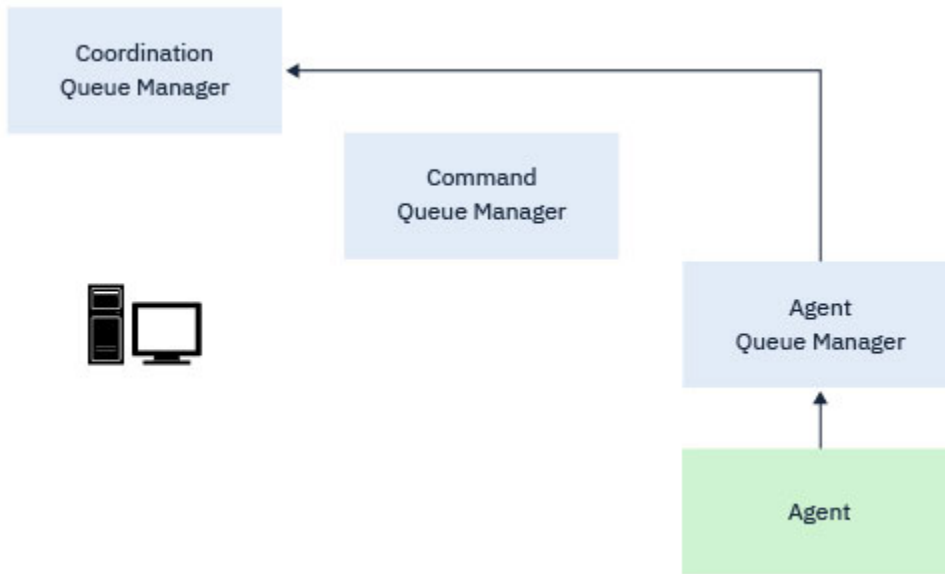


图 9: 代理程序在协调队列管理器上发布状态信息或存储传输模板

当运行任何连接到协调队列管理器的上述命令时，它们将直接连接到协调队列管理器，并执行以下任一操作：

- 创建或删除传输模板。
- 查询有关代理，监视器或已调度传输的状态信息，并向用户显示该信息。

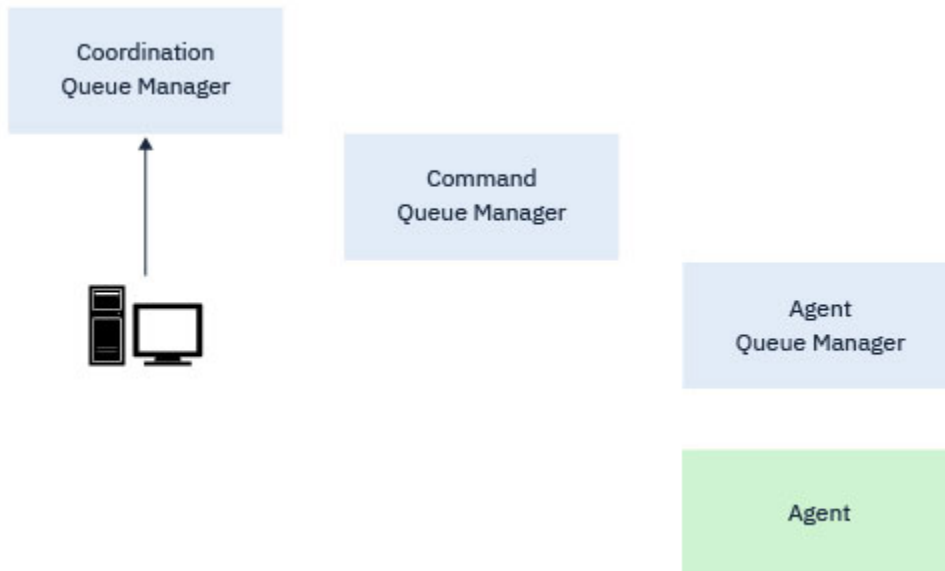


图 10: 然后，命令直接连接到协调队列管理器以获取相应的状态信息或使用传输模板。

连接到命令队列管理器的命令

以下命令连接到命令队列管理器：

- [fteCancelTransfer](#)
- [fteCreateMonitor](#)
- [fteCreateTransfer](#)
- [fteDeleteMonitor](#)

- [fteDeleteScheduledTransfer](#)
- [ftePingAgent](#)
- [fteStopAgent](#)

您可以将命令队列管理器视为 Managed File Transfer 拓扑中的一种网关形式。它使用发送方和接收方通道连接到代理队列管理器。

当运行任何连接到队列管理器的上述命令时，它们将执行下列操作：

- 连接到命令队列管理器。
- 创建临时应答队列。
- 将包含命令详细信息的信息发送到相应的代理程序。

该消息通过 IBM MQ 网络路由到代理队列管理器，在该队列管理器中，该消息由代理程序选取并进行处理。在代理处理该命令后，代理将应答发送回命令队列管理器，该命令将在该队列管理器中获取应答。

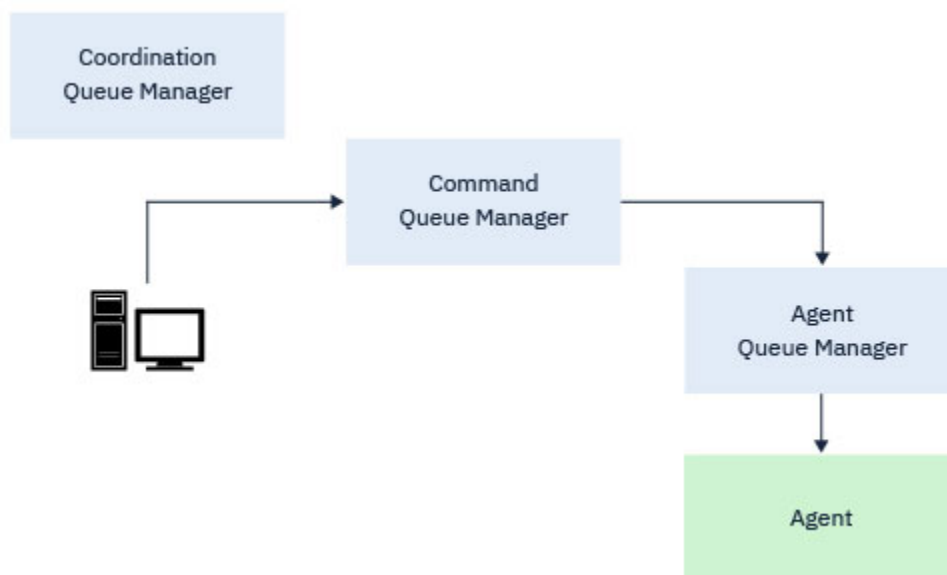


图 11: 这些命令连接到命令队列管理器。然后，包含该命令的消息将通过 IBM MQ 网络路由到正确的代理队列管理器，代理将在该队列管理器中选取该消息。

连接到代理队列管理器的命令

以下命令连接到代理队列管理器：

- [fteCleanAgent](#)
- [fteCreateAgent](#)
- [fteCreateBridgeAgent](#)
- [fteCreateCDAgent](#)
- [fteDeleteAgent](#)

每个代理都有自己的代理队列管理器。代理使用此队列管理器上托管的系统队列来维护状态信息和接收工作请求。

单个队列管理器可以充当多个代理的代理队列管理器。代理队列管理器连接到协调队列管理器，命令队列管理器以及使用发送方和接收方通道的其他代理队列管理器。

fteCreateAgent、**fteCreateBridgeAgent** 和 **fteCreateCDAgent** 命令将代理队列管理器名称作为自变量。

运行这些命令时，它们会连接到指定的队列管理器，并向协调队列管理器发送一条消息，指示已将代理添加到 Managed File Transfer 拓扑中。

同样，运行 **fteDeleteAgent** 时，它会连接到代理队列管理器，并向协调队列管理器发送一条消息，通知它现在已从 Managed File Transfer 拓扑中除去代理。

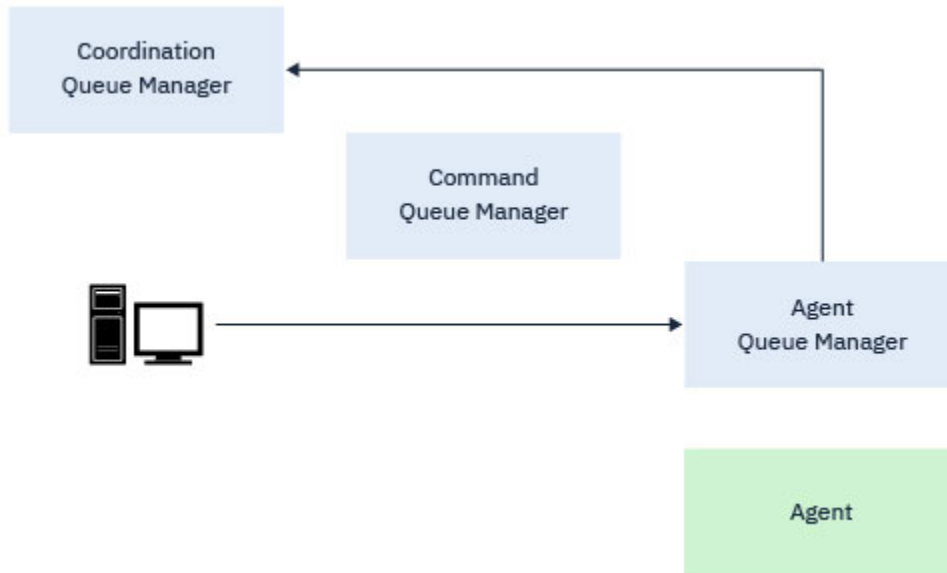



图 12: 各种 **fteCreate** 命令以及 **fteDeleteAgent** 命令连接到代理队列管理器，并向协调队列管理器发送一条消息，以从拓扑中注册或注销代理。

 **注意: fteCleanAgent** 连接到代理队列管理器，并从其系统队列中除去该代理的任何状态信息。运行此命令可能会影响整个拓扑。因此，您只能在 IBM 的指导下运行此命令。

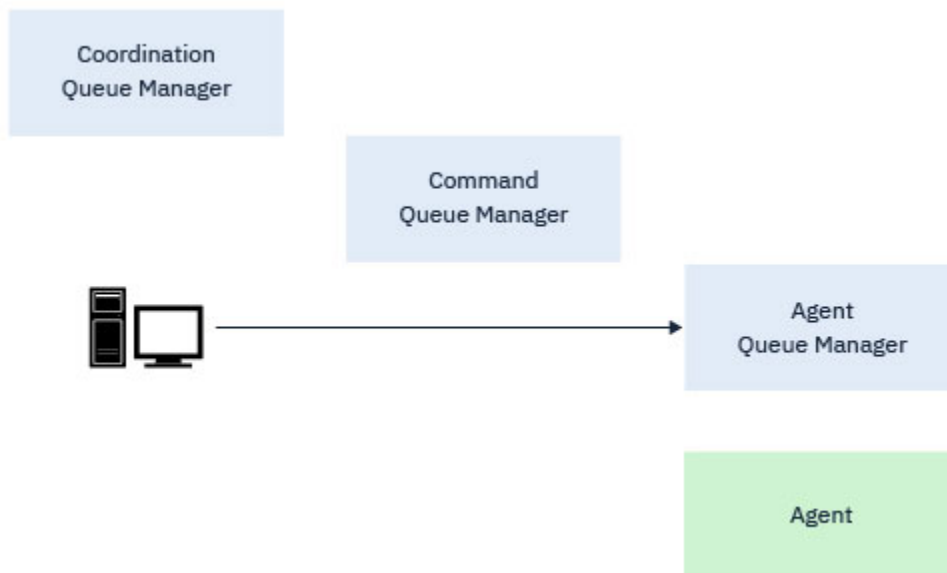


图 13: **fteCleanAgent** 命令直接连接到代理队列管理器，并从代理的系统队列中除去状态信息

相关信息

[已安装的 MFT 命令集](#)

MFT 命令

下表列出了所有 Managed File Transfer 命令以及指向其详细描述的连接。



表 35: Managed File Transfer 命令及其用途	
命令名	用途
迁移命令:	
fteMigrateAgent	将代理程序及其配置从 IBM WebSphere MQ File Transfer Edition 7.0.4 迁移到 Managed File Transfer 7.5 或更高版本
fteMigrateConfigurationOptions	将 IBM WebSphere MQ File Transfer Edition 7.0 配置迁移到 IBM WebSphere MQ 7.5 或更高版本中的 Managed File Transfer。
fteMigrateLogger	将独立数据库记录器的配置从 IBM WebSphere MQ File Transfer Edition 7.0.1 或更高版本迁移到 Managed File Transfer 7.5 或更高版本。
配置命令:	
fteChangeDefaultConfigurationOptions	更改希望 Managed File Transfer 使用的缺省配置选项
fteCreateAgent	创建 Managed File Transfer Agent
fteCreateBridgeAgent	创建 Managed File Transfer 协议网桥代理
fteCreateCDAgent	创建 Managed File Transfer Connect:Direct 网桥代理
 fteCreate 环境	为 Redistributable Managed File Transfer Agent 的文件配置和传输设置环境变量。
fteCreateLogger	创建 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 Managed File Transfer Agent
fteDeleteLogger	删除 Managed File Transfer 记录器
fteModifyAgent	仅适用于 Windows。将代理、Connect:Direct 网桥代理或协议网桥代理修改为作为 Windows 服务运行。
fteModifyLogger	仅适用于 Windows。将记录器修改为作为 Windows 服务运行。
fteSetupCommands	指定当您发出命令时连接到 IBM MQ 网络的队列管理器的详细信息
fteSetupCoordination	配置 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 IBM MQ Explorer 创建的所有 Managed File Transfer 传输。


表 35: Managed File Transfer 命令及其用途 (继续)	
命令名	用途
fteListTemplates	列出协调队列管理器的所有文件传输模板
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。
 fteSetProductID	设置 z/OS SCRT 记录产品标识
 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
 fteSetAgentLog 级别	对 Managed File Transfer 协议网桥代理与 FTP/SFTP/FTPS 文件服务器之间的文件传输启用或禁用诊断信息日志记录。
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


请参阅已安装的 MFT 命令集，以获取用于显示哪些命令与哪种 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](#)。

使用 MFT 命令的权限

如果您要发出 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 备用组的更多信息，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](#)。

可以使用 IBM MQ Explorer 来发出 Managed File Transfer 命令的子集。

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：

Managed File Transfer for 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。

针对 UNIX 系统的 Managed File Transfer

所有 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

z/OS

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 53 页的『创建 MFT 代理或记录器命令数据集』。

从 IBM i 平台发出命令

IBM i

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 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。

相关参考

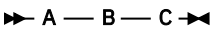
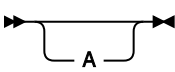
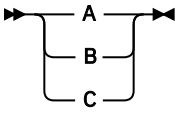
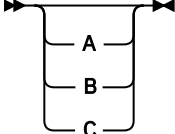
第 282 页的『MFT 的返回码』

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。

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：

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。

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 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：在启动代理时，跟踪将转至 *mft_config/logscoordination_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
```

相关参考

第 226 页的『对 MFT 进行故障诊断』

使用以下参考信息可帮助您诊断 Managed File Transfer 中的错误:

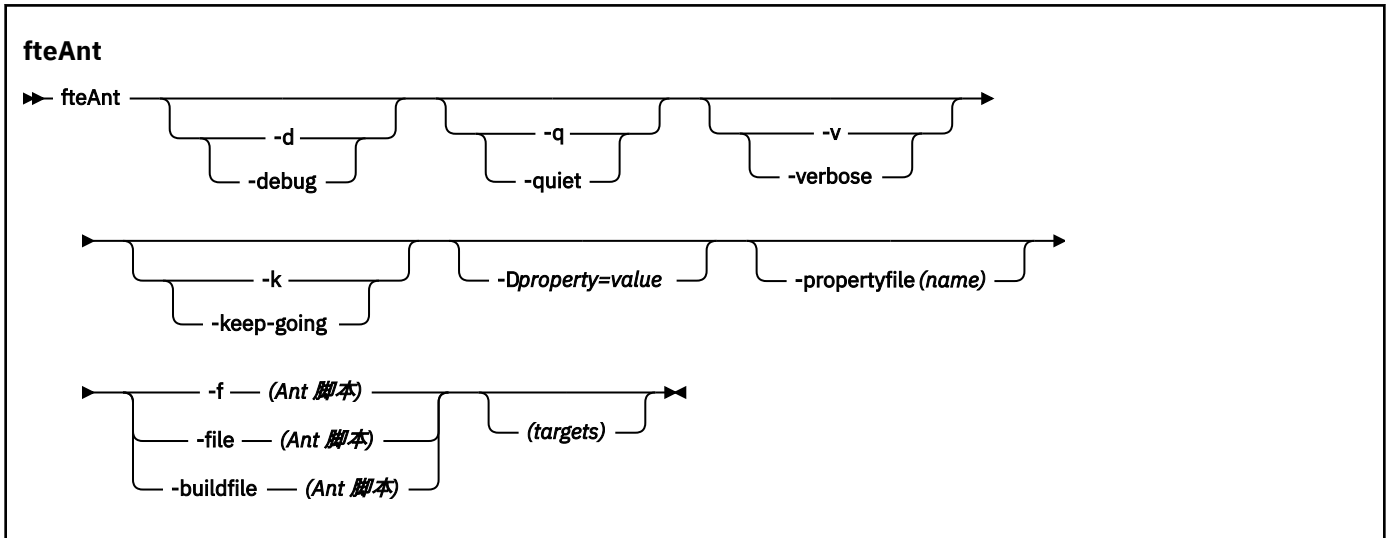
fteAnt: 在 MFT 中运行 Ant 任务

fteAnt 命令会在具有可用的 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 Managed File Transfer 的环境中运行 Ant 脚本。与标准 **ant** 命令不同, **fteAnt** 要求您定义脚本文件。

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 script)、-file (Ant script) 或 -buildfile (Ant script)

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 Managed File Transfer 命令和 Ant 版本。

-你说什么 或 -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。

z/OS fteBatch、fteCommon 和 ftePlatform

z/OS fteBatch、fteCommon 和 ftePlatform 是 Managed File Transfer 在 `MQ_INSTALLATION_PATH/bin` 目录中作为帮助脚本提供的脚本。并非每一个平台上都有这些脚本。

fteBatch 脚本 (仅限 z/OS)

z/OS fteBatch 是用于从 JZOS Batch Launcher 运行 Managed File Transfer 的帮助脚本。fteBatch 仅安装在 z/OS 上。通常 Managed File Transfer 使用提供的命令 shell 脚本启动，这些脚本会在启动适用于该功能的 Java 类之前执行某些环境配置。当使用 JZOS Batch Launcher 启动 Managed File Transfer 时，会直接从 Launcher 启动 Java 类。fteBatch 可作为启动程序设置的一部分加以调用，以将所需的类名放置到环境变量中，并在启动 Java 之前执行正常的命令 shell 脚本所执行的设置工作。这能够在您的作业与 Managed File Transfer 使用的内部类名之间提供一定程度的隔离。

对于 Managed File Transfer 8.0，不推荐使用 fteBatch 命令，因为您可以通过新的 PDSE 数据集命令来运行 Managed File Transfer。要了解更多信息，请参阅 [第 53 页的『创建 MFT 代理或记录器命令数据集』](#)。

fteCommon

fteCommon 是由其他 Managed File Transfer 命令脚本启动的辅助脚本，用于在启动 Java 之前执行常见的设置处理。

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

使用 **fteCancelTransfer** 命令可取消 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

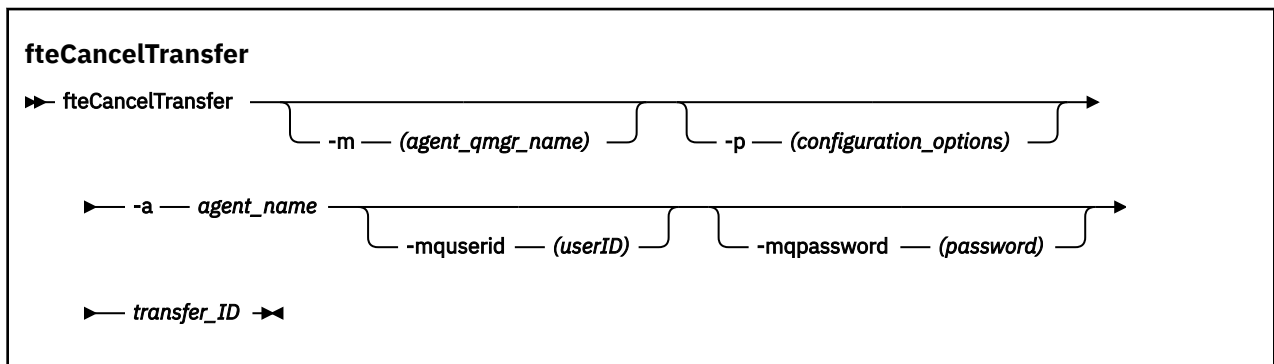
如果在传输正在进行时发出 **fteCancelTransfer** 命令，那么已在该传输过程中传输的任何文件都将保留在目标系统上而不会被删除。在该传输过程中部分传输的任何文件都将从目标系统中删除。传输为“已取消”的传输日志的目标端。

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。

您可以从可以连接到 IBM MQ 网络的任何系统运行 **fteCancelTransfer** 命令，然后路由到代理队列管理器。具体而言，要运行此命令，您必须已在该系统上安装了 Managed File Transfer，并将 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 (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。

-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，或者显示在 " IBM MQ 资源管理器传输日志 " 面板中。

-你说什么 或 -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 364 页的『[fteCreateTransfer: 启动新的文件传输](#)』



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

使用 **fteChangeDefaultConfigurationOptions** 命令来更改希望 Managed File Transfer 使用的缺省配置选项。配置选项的值定义 Managed File Transfer 使用的属性文件组。

要点:  在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理员（和 mqm 组成员）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

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。

您可以使用 **fteChangeDefaultConfigurationOptions** 命令来更改在 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 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 49 页的『[多平台上的 MFT 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。

-你说什么 或 -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 49 页的『多平台上的 MFT 配置选项』

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 MFT 代理

使用 **fteCleanAgent** 命令，可通过删除 Managed File Transfer Agent 使用的持久性和非持久性队列中的消息来清除该代理使用的队列。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 9.0.3 和 IBM MQ 9.0.0 Fix Pack 2 之前，如果仅运行带有 **agent_name** 参数的 **fteCleanAgent** 命令，那么缺省情况下，该命令会像提供了 **-all** 参数一样运行，这将导致清除代理上的所有调度传输，资源监视器和调度传输定义。

V 9.0.0.2 **V 9.0.3** 从 IBM MQ 9.0.3 和 IBM MQ 9.0.0 Fix Pack 2 开始，缺省情况下，必须通过将相应参数传递到 **fteCleanAgent** 命令以及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。如果在当前运行的代理上尝试运行此命令，那么您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，您必须拥有代理锁文件的写访问权，该文件位于

`MQ_DATA_PATH\mqft\logs\coordination_QMgr_name\agents\agent_name\agent.lck`

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`

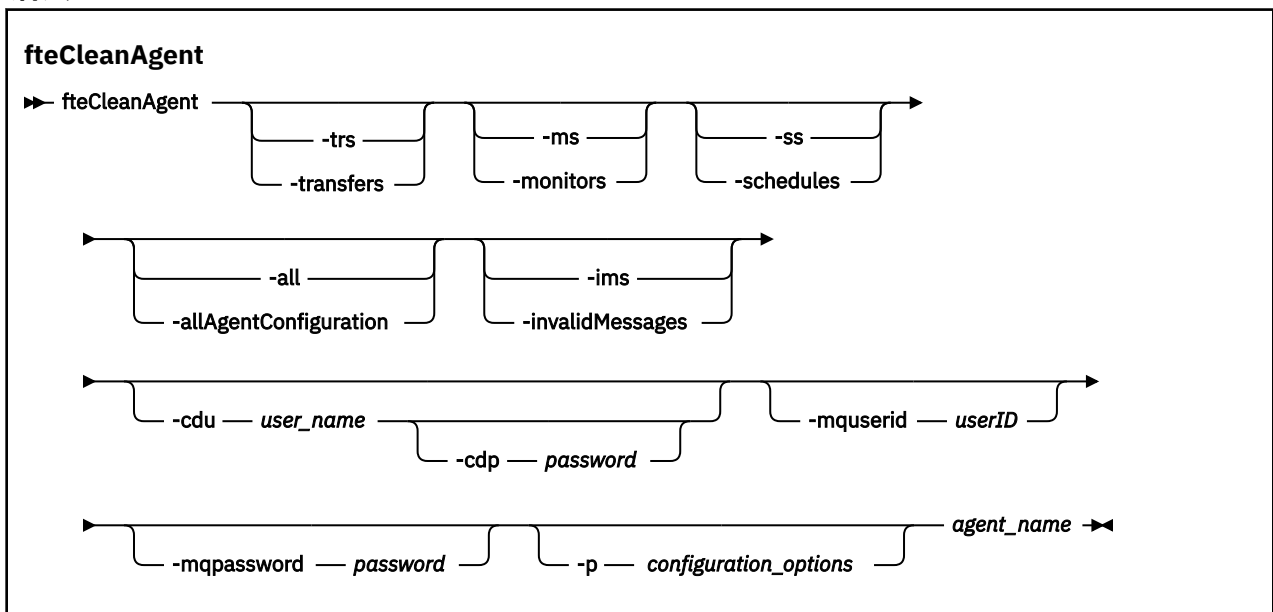
请参阅第 293 页的『限制特定于 MFT 的资源组权限』，以获取有关 FTEAGENT 组和限制组权限的更多信息。

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：在清除 Connect:Direct 网桥代理时，用于运行 **fteCleanAgent** 命令的用户标识必须具有 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。仅当要清除的代理为 Connect:Direct 网桥代理并且您指定了 **-cdu** 参数时才有效。如果您指定 **-cdp** 参数，那么该命令将使用提供的密码来建立到 Connect:Direct 网桥节点的连接，并且检索有关现有 Connect:Direct 进程的更多信息。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 (configuration_options)

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 Managed File Transfer 代理的名称。

-你说什么 或 **-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

Scheduler Queue Schedule Entries:

Warning - Invalid message found on the queue

BFGCL0149I: The agent 'AGENT2' has been cleaned.

在该示例中，将清除 Connect:Direct 网桥代理 AGENT_CD_BRIDGE 使用的传输队列：

```
C:\Documents and Settings\Administrator>fteCleanAgent -trs -cdu USER1 AGENT_CD_BRIDGE
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Enter Connect:Direct password:
```

All messages will be deleted from the state and command queues

State Queue Entries:

Transfer Identifier: 414d5120716d31202020202020202020202020202786de4d2048a703
Source Agent Name: AGENT2
Destination Agent Name: AGENT_CD_BRIDGE
Connect:Direct PNODE Name: CDNODE1
Connect:Direct SNODE Name: CDNODE2
Connect:Direct Current Processes: Name=FA34F8, Number=139

Command Queue New Transfer Entries:

BFGCL0149I: The agent 'AGENT_CD_BRIDGE' has been cleaned.

返回码

0

命令成功完成。

1

命令结束但未成功。

相关参考

第 445 页的『[fteStopAgent](#)』

使用 **fteStopAgent** 命令以受控方式停止 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 386 页的『[fteDeleteAgent](#)』

fteDeleteAgent 命令将删除 Managed File Transfer Agent 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。


第 457 页的『[MFT 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 Managed File Transfer 联系该队列管理器所需的信息。



fteCreateAgent (创建 MFT 代理)

fteCreateAgent 命令创建 Managed File Transfer Agent 及其关联的配置。

您可以控制对代理的访问。有关更多信息，请参阅 [限制用户对 MFT 代理程序操作的权限](#)。您必须使用 **-ac** 参数，并提供许可权来访问一些队列。

要点:  在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理员（和 mqm 组成员）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 **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`。

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 691 页的『MFT 凭证文件格式』。

要点:

在 UNIX 平台和 Linux Managed File Transfer 命令上，使用套接字文件与在同一主机上运行的代理程序进程进行通信。

这些套接字文件在代理程序的日志目录中创建，并在代理程序停止时删除。在 IBM MQ Managed File Transfer 安装中，将使用以下文件路径创建此套接字文件：`<MQ_DATA_PATH>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>`，其中缺省情况下 `MQ_DATA_PATH` 为 `/var/mqm`。

对于可再分发的代理程序，将在以下目录下创建此套接字文件：`<RE_DISTRIBUTABLE_DIRECTORY>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>`。

例如，如果代理名称为 `SRCAGENT`，代理队列管理器名称为 `SRCAGENTQM`，协调队列管理器名称为 `COORDQM`，并且可再分发代理正在从目录 `/home/myuser/mqmft-redis` 运行，那么此套接字文件的完整路径为：`/home/myuser/mqmft-redis/mqft/logs/COORDQM/agents/SRCAGENT/logs/SRCAGENT@SRCAGENTQM`

总文件路径长度为 85 个字符。

这些操作系统允许套接字文件的最大路径长度为 107 个字符。因此，在创建代理程序时，请注意确保套接字文件路径不超过 107 个字符。对于可重新分发的代理程序而言，这一点尤其重要，该代理程序的日志目录可以位于任意目录位置中。有关设置配置目录的详细信息，请参阅 `fteCreateEnvironment` 命令。

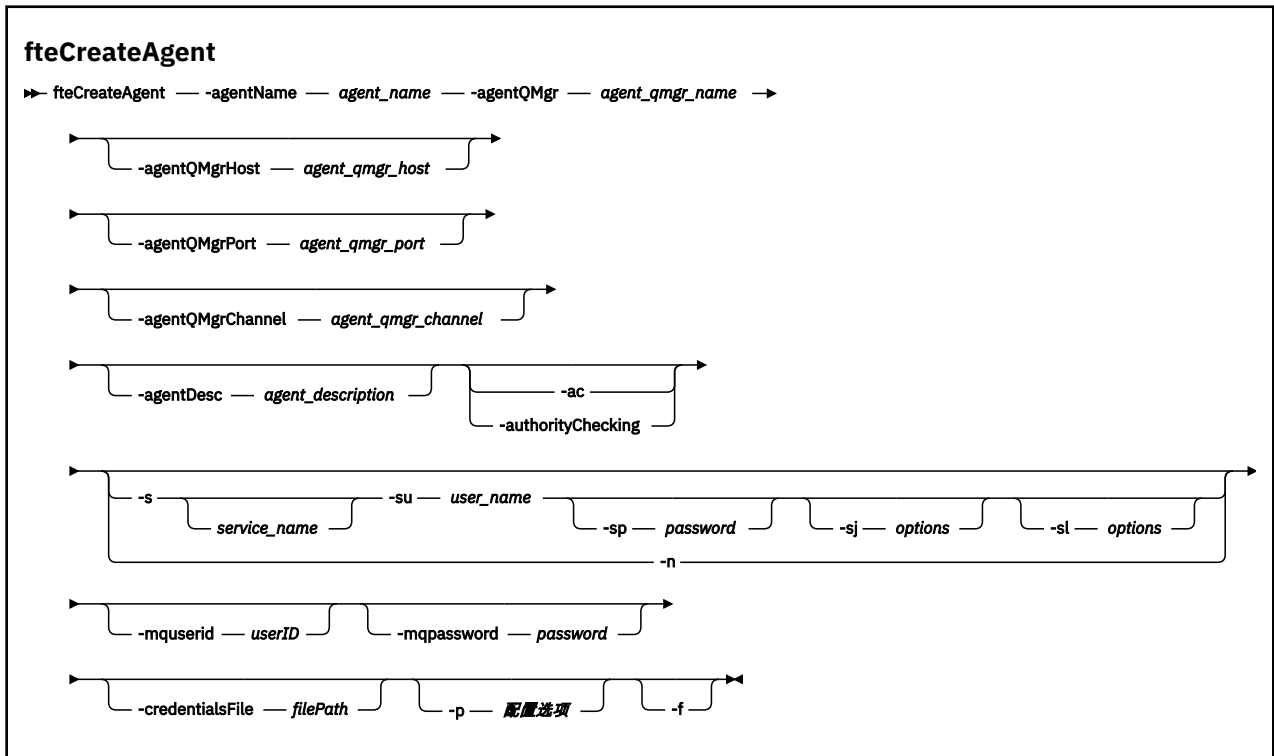
如果您启动代理程序，或者运行连接到代理程序的其他命令，并且路径长度超过 107 个字符，那么将收到以下消息：

```
BFGNV0159E: 尝试使用 FFDC 绑定到套接字文件失败
```

特殊字符

请谨慎使用包含特殊字符的参数值，从而避免命令 shell 以您不期望的方式解析字符。例如，包含空格、引号（单或双）、正斜杠或反斜杠之类字符的标准文件路径和名称可能由命令 shell 进行解释，而不是直接传递到命令本身。为避免命令 shell 解释字符，请将整个参数括在双引号/单引号中，或者使用命令 shell 的转义序列来转义特殊字符。

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。

-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[限制用户对 MFT 代理程序操作的权限](#)。

Windows -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此命令。如果不指定 *service_name*，那么服务将命名为 `mqmftAgentAGENTQMGR`，其中 *AGENT* 是代理名称；而 *QMGR* 是代理队列管理器名称。

服务的显示名称（显示在 Windows 服务窗口的名称列中）始终为 **Managed File Transfer Agent AGENT@QMGR**。

注: 如果可再发行代理将作为 Windows 服务运行, 那么 **BFG_DATA** 环境变量需要在系统环境中进行设置, 以便服务正常工作。

Windows -su (user_name)

可选 (仅限 Windows)。如果该代理将作为 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 **Log on as a service** 权限。有关如何授予此权限的信息, 请参阅 [有关将 MFT 代理程序或记录器作为 Windows 服务运行的指南](#)。

在指定 **-s** 时为必需项。

Windows -sp (password)

可选 (仅限 Windows)。

该参数仅在指定 **-s** 时才有效。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。

Windows -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。

Windows -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 服务存在问题, 那么此选项很有用。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。

Windows -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 Windows 进程。

-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, `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。

此命令支持将一组 IBM MQ 认证详细信息添加到指定的 Managed File Transfer 凭证文件中。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。

-你说什么 或 -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。

fteCreateBridgeAgent (创建并配置 MFT 协议网桥代理)

fteCreateBridgeAgent 命令将创建 Managed File Transfer 协议网桥代理及其关联配置。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: **ULW** 在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理员（和 mqm 组成员）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。

z/OS 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）。
- **V9.0.0.1** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V9.0.0.1** 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 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。
```

用户负责手动创建 ProtocolBridgeCredentials.xml 文件，该文件不再由 **fteCreateBridgeAgent** 命令创建。

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 478 页的『协议网桥凭证文件格式』和第 481 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（参数 -bt），那么该缺省服务器包含在 ProtocolBridgeProperties.xml 文件中且其主机名将用于服务器名称。使用 -bt 参数时，您必须指定以下参数：

- -bh
- -btz
- -bm
- -bsl
- -bfe
- -bts

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。

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。Advanced agent properties: Protocol bridge 和 Advanced agent properties: Protocol bridge agent logging 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 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。

要点：

在 UNIX 平台和 Linux Managed File Transfer 命令上，使用套接字文件与在同一主机上运行的代理程序进程进行通信。

这些套接字文件在代理程序的日志目录中创建，并在代理程序停止时删除。在 IBM MQ Managed File Transfer 安装中，将使用以下文件路径创建此套接字文件：<MQ_DATA_PATH>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>，其中缺省情况下 MQ_DATA_PATH 为 /var/mqm。

对于可再分发的代理程序，将在以下目录下创建此套接字文件：<RE_DISTRIBUTABLE_DIRECTORY>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>。

例如，如果代理名称为 SRCAGENT，代理队列管理器名称为 SRCAGENTQM，协调队列管理器名称为 COORDQM，并且可再分发代理正在从目录 /home/myuser/mqmftr-dist 运行，那么此套接字文件的完整路径为: /home/myuser/mqmftr-dist/mqmftr/logs/COORDQM/agents/SRCAGENT/logs/SRCAGENT@SRCAGENTQM

总文件路径长度为 85 个字符。

这些操作系统允许套接字文件的最大路径长度为 107 个字符。因此，在创建代理程序时，请注意确保套接字文件路径不超过 107 个字符。对于可重新分发的代理程序而言，这一点尤其重要，该代理程序的日志目录可以位于任意目录位置中。有关设置配置目录的详细信息，请参阅 **fteCreateEnvironment** 命令。

如果您启动代理程序，或者运行连接到代理程序的其他命令，并且路径长度超过 107 个字符，那么将收到以下消息：

BFGNV0159E: 尝试使用 FFDC 绑定到套接字文件失败

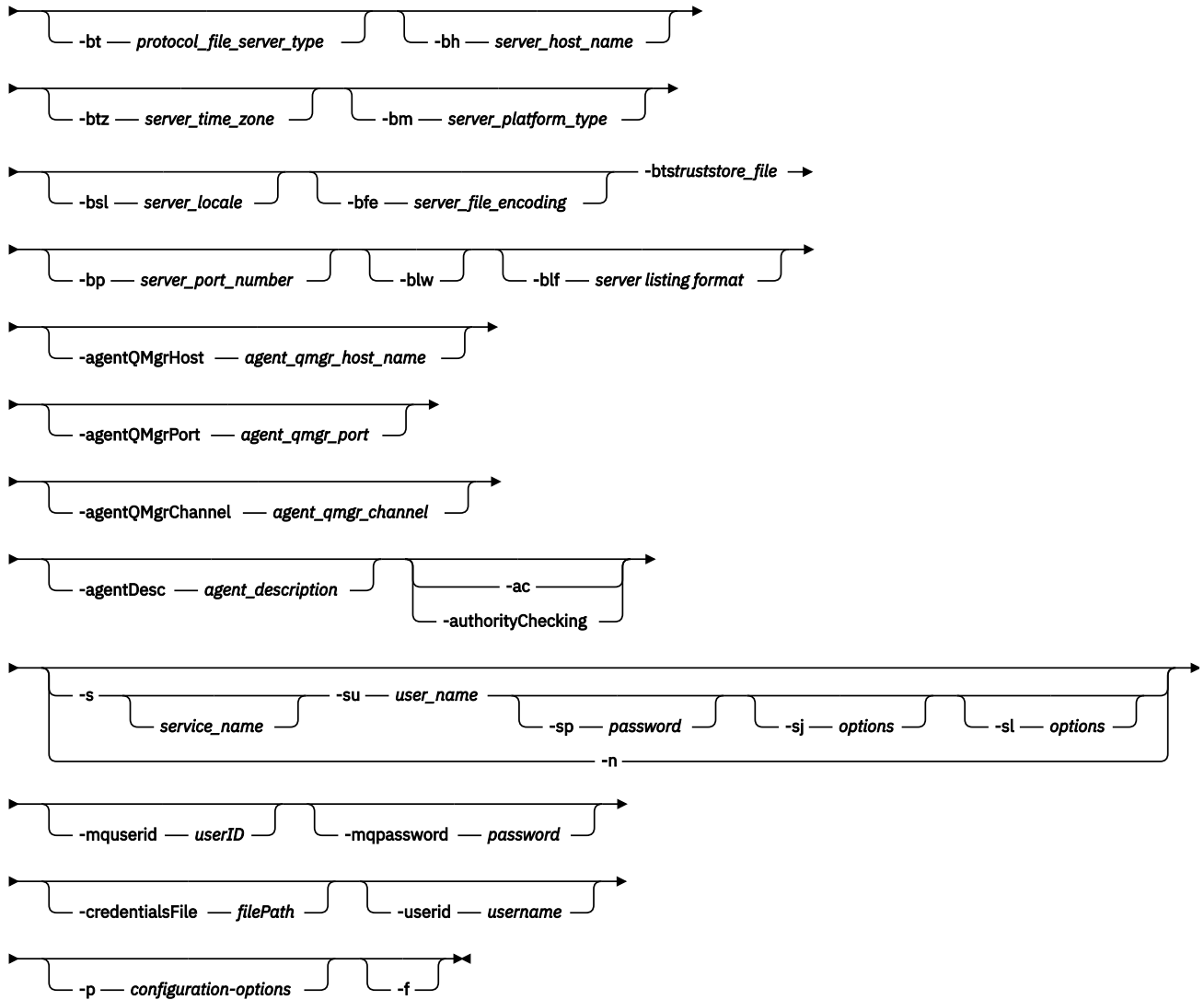
特殊字符

请谨慎使用包含特殊字符的参数值，从而避免命令 shell 以您不期望的方式解析字符。例如，包含空格、引号（单或双）、正斜杠或反斜杠之类字符的标准文件路径和名称可能由命令 shell 进行解释，而不是直接传递到命令本身。为避免命令 shell 解释字符，请将整个参数括在双引号/单引号中，或者使用命令 shell 的转义序列来转义特殊字符。

语法

fteCreateBridgeAgent

► fteCreateBridgeAgent — -agentName *agent_name* -agentQMgr *agent_qmgr_name* →



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

通用 UNIX 和 Linux 平台

Windows

通用 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，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

通用 UNIX 和 Linux 平台

Windows

通用 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 UNIX 显示以下类型的列表：

```
-rwxr-xr-x 2 userid groupId 4096 2009-07-23 09:36 filename
```

Windows Windows 显示以下类型的列表：

```
437,909 filename
```

IBM i IBM i 显示以下类型的列表：

```
OS400IFS -rwxrwsrwx 3 USERID 0 8192 Mar 7 08:33 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。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 [限制用户对 MFT 代理程序操作的权限](#)。

Windows -s (service_name)

可选（仅限 Windows）。指示该代理将作为 Windows 服务运行。如果不指定 *service_name*，那么服务将命名为 `mqmftAgentAGENTQMGR`，其中 *AGENT* 是代理名称；而 *QMGR* 是代理队列管理器名称。

服务的显示名称（显示在 Windows 服务窗口的名称列中）始终为 **Managed File Transfer Agent AGENT@QMGR**。

Windows -su (user_name)

可选（仅限 Windows）。如果该代理将作为 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 **Log on as a service** 权限。有关如何授予此权限的信息，请参阅 [有关将 MFT 代理程序或记录器作为 Windows 服务运行的指南](#)。

在指定 **-s** 时为必需项。

Windows -sp (password)

可选（仅限 Windows）。**-su** 参数设置的用户帐户的密码。

该参数仅在指定 **-s** 时才有效。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。

Windows **-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。

仅当指定了 **-s** 时，此参数才有效。

Windows **-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 服务存在问题，那么此选项很有用。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。

Windows **-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 Windows 进程。

-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[配置选项](#)，以获取更多信息。

-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。

此命令支持将一组 IBM MQ 认证详细信息添加到指定的 Managed File Transfer 凭证文件中。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。

-你说什么 或 -h

可选。显示命令语法。

不推荐使用的参数

以下参数已不推荐使用，并且在 IBM WebSphere MQ 7.5 或 IBM WebSphere MQ File Transfer Edition 7.0.2 或更高版本上不受支持。

-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
```

注：此代码不适用于 Managed File Transfer Agent 可再分发组件。

其他定制

如果已使用 -bt 参数（及需要的其他参数），那么 ProtocolBridgeProperties.xml 文件中将存在缺省服务器名称。

如果要添加其他 FTP 服务器或更改凭证文件的位置，请参阅 [使用 ProtocolBridgeProperties.xml 文件定义协议文件服务器的属性](#)。

返回码

0

命令成功完成。

1

命令结束但未成功。

使用 **fteStartAgent** 命令可启动协议网桥代理。有关更多信息，请参阅 [第 442 页的『fteStartAgent』](#)。

 另请参阅 [在 z/OS 上启动 MFT 代理](#)。

相关参考

[第 478 页的『协议网桥凭证文件格式』](#)

Managed File Transfer Agent 配置目录中的 ProtocolBridgeCredentials.xml 文件定义了协议网桥代理用于向协议服务器授权自身的用户名和凭证信息。

[第 481 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。

相关信息

[协议网桥](#)

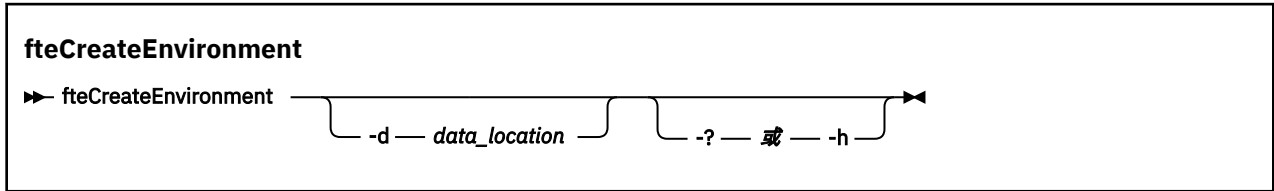
fteCreateEnvironment (为 Redistributable Managed File Transfer Agent 设置环境)

fteCreateEnvironment 命令为 Redistributable Managed File Transfer Agent 的文件配置和传输设置环境。

用途

使用 **fteCreateEnvironment** 命令可为使用 Redistributable Managed File Transfer Agent 设置环境。您可以运行具有 **-d** 参数的此命令，为 MFT 代理数据文件指定位置。如果未指定 **-d** 参数，该命令将在 Redistributable Managed File Transfer Agent 下载位置中创建数据文件，并设置数据路径。

语法



参数

-d (data_location)

可选。在设置环境时，此参数用于指定数据文件的位置。

如果未指定此参数，将在解压缩 Redistributable Managed File Transfer Agent 的位置创建数据目录（如果此目录尚未存在），并为此位置设置环境变量 (BFG_DATA)。

-你说什么 或 -h

可选。显示命令语法。

示例

在此示例中，在 Windows 上，**-d** 参数指定要在其中创建数据文件夹的位置：

```
fteCreateEnvironment -d C:\mftRedistributable\mftData
```

在 Linux 上，作为先决条件，此命令必须在 Bash shell 上运行。在 Bash shell 中，可通过多种方式来运行此命令，并且必须指定命令文件的来源：

```
source Path_of_MFTZipBin/fteCreateEnvironment
```

替代方法为：

```
. Path_of_MFTZipBin/fteCreateEnvironment
```

或者，如果从命令文件所在的目录中运行：

```
./fteCreateEnvironment
```



注意： 请注意第一个句点字符 (.) 后的空格

返回码

0

命令成功完成。

1

命令结束但未成功。

fteCreateCDAgent (创建 Connect:Direct 网桥代理)

fteCreateCDAgent 命令用于创建 Managed File Transfer Agent 及其相关联的配置，以与 Connect:Direct 网桥一起使用。

要点: `ULW` 在 IBM MQ for UNIX, Linux, and Windows 上, 只有 IBM MQ 管理员 (和 mqm 组成员) 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。

z/OS 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)。
- `V9.0.0.1` 是 BFG_GROUP_NAME 环境变量中指定的组 (如果指定了一个组) 的成员。
- `V9.0.0.1` 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 [Connect:Direct 网桥](#)。有关 Connect:Direct 网桥的受支持操作系统版本的详细信息, 请参阅 Web 页面 [System Requirements for IBM MQ](#)。

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`。

Managed File Transfer 提供帮助您配置代理的高级代理属性。[MFT 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, 您必须手动创建 `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 487 页的『[Connect:Direct 凭证文件格式](#)』。

要点:

在 UNIX 平台和 Linux Managed File Transfer 命令上，使用套接字文件与在同一主机上运行的代理程序进程进行通信。

这些套接字文件在代理程序的日志目录中创建，并在代理程序停止时删除。在 IBM MQ Managed File Transfer 安装中，将使用以下文件路径创建此套接字文件: `<MQ_DATA_PATH>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>`，其中缺省情况下 MQ_DATA_PATH 为 `/var/mqm`。

对于可再分发的代理程序，将在以下目录下创建此套接字文件: `<RE_DISTRIBUTABLE_DIRECTORY>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>`。

例如，如果代理名称为 SRCAGENT，代理队列管理器名称为 SRCAGENTQM，协调队列管理器名称为 COORDQM，并且可再分发代理正在从目录 `/home/myuser/mqmft-redis` 运行，那么此套接字文件的完整路径为: `/home/myuser/mqmft-redis/mqft/logs/COORDQM/agents/SRCAGENT/logs/SRCAGENT@SRCAGENTQM`

总文件路径长度为 85 个字符。

这些操作系统允许套接字文件的最大路径长度为 107 个字符。因此，在创建代理程序时，请注意确保套接字文件路径不超过 107 个字符。对于可重新分发的代理程序而言，这一点尤其重要，该代理程序的日志目录可以位于任意目录位置中。有关设置配置目录的详细信息，请参阅 **fteCreateEnvironment** 命令。

如果您启动代理程序，或者运行连接到代理程序的其他命令，并且路径长度超过 107 个字符，那么将收到以下消息：

```
BFGNV0159E: 尝试使用 FFDC 绑定到套接字文件失败
```

特殊字符

请谨慎使用包含特殊字符的参数值，从而避免命令 shell 以您不期望的方式解析字符。例如，包含空格、引号（单或双）、正斜杠或反斜杠之类字符的标准文件路径和名称可能由命令 shell 进行解释，而不是直接传递到命令本身。为避免命令 shell 解释字符，请将整个参数括在双引号/单引号中，或者使用命令 shell 的转义序列来转义特殊字符。

fteCreateCDAgent

► fteCreateCDAgent — *agent_name* — *agent_qmgr_name* →

► -cdNode — *cd_node_name* — *agent_qmgr_host* — *agent_qmgr_host* →

► -agentQMGrPort — *agent_qmgr_port* →

► -agentQMGrChannel — *agent_qmgr_channel* →

► -agentDesc — *agent_description* — *ac* — *authorityChecking* →

► -p — *configuration_options* — *f* — *cdNodeHost* — *cd_node_host* →

► -cdNodePort — *cd_node_port* — *cdTmpDir* — *cd_tmp_dir* →

► -s — *service_name* — *su* — *user_name* — *sp* — *password* — *sj* — *options* — *sl* — *options* — *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 [限制用户对 MFT 代理程序操作的权限](#)。

-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。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 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。

注: 如果运行 **fteCleanAgent** 命令，那么将删除该目录中的所有文件。

Windows -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此命令。如果不指定 *service_name*，那么服务将命名为 **mqmfAgentAGENTQMGR**，其中 **AGENT** 是代理名称；而 **QMGR** 是代理队列管理器名称。

服务的显示名称（显示在 Windows 服务窗口的名称列中）始终为 **Managed File Transfer Agent AGENT@QMGR**。

注: 如果可再发行代理将作为 Windows 服务运行，那么 **BFG_DATA** 环境变量需要在系统环境中进行设置，以便服务正常工作。

Windows -su (user_name)

可选（仅限 Windows）。如果该代理将作为 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 **Log on as a service** 权限。有关如何授予此权限的信息，请参阅 [有关将 MFT 代理程序或记录器作为 Windows 服务运行的指南](#)。

在指定 **-s** 时为必需项。

Windows -sp (password)

可选（仅限 Windows）。

该参数仅在指定 **-s** 时才有效。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。

Windows **-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。

Windows **-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 服务存在问题，那么此选项很有用。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。

Windows **-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 Windows 进程。

-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。

此命令支持将一组 IBM MQ 认证详细信息添加到指定的 Managed File Transfer 凭证文件中。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。与 Connect:Direct 之间传输的文件临时存储在目录 /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。

fteCreate 记录器 (创建 MFT 文件或数据库记录器)

使用 **fteCreateLogger** 命令可创建 Managed File Transfer 文件或数据库记录器。

要点:  在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理员（和 mqm 组成员）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。

z/OS 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）。
- **V9.0.0.1** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V9.0.0.1** 在运行命令时，在 BFG_GROUP_NAME 环境变量中未设置任何值。

IBM i 上的记录器

IBM i

IBM i 平台上不支持 Managed File Transfer 记录器。

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`

如果您以后希望删除该记录器，请使用 **fteDeleteLogger** 命令。

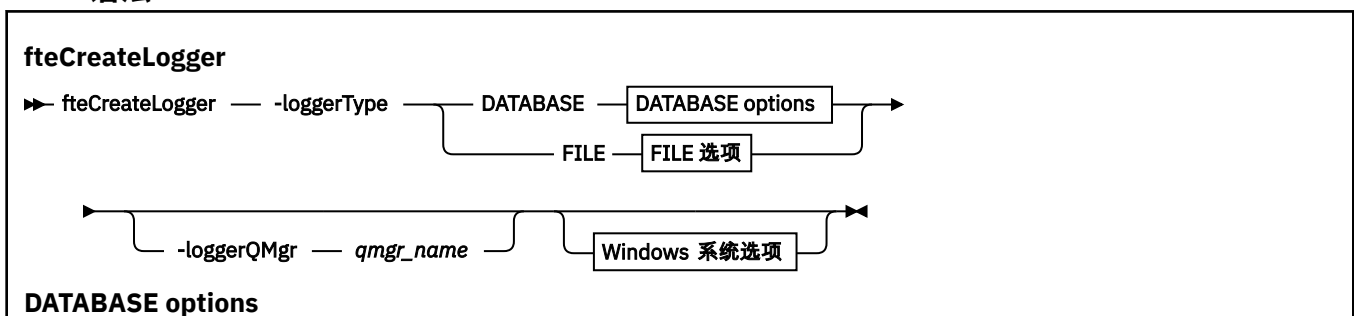
Managed File Transfer 提供可以帮助您配置记录器的高级记录器属性。请参阅 [MFT 记录器配置属性](#)

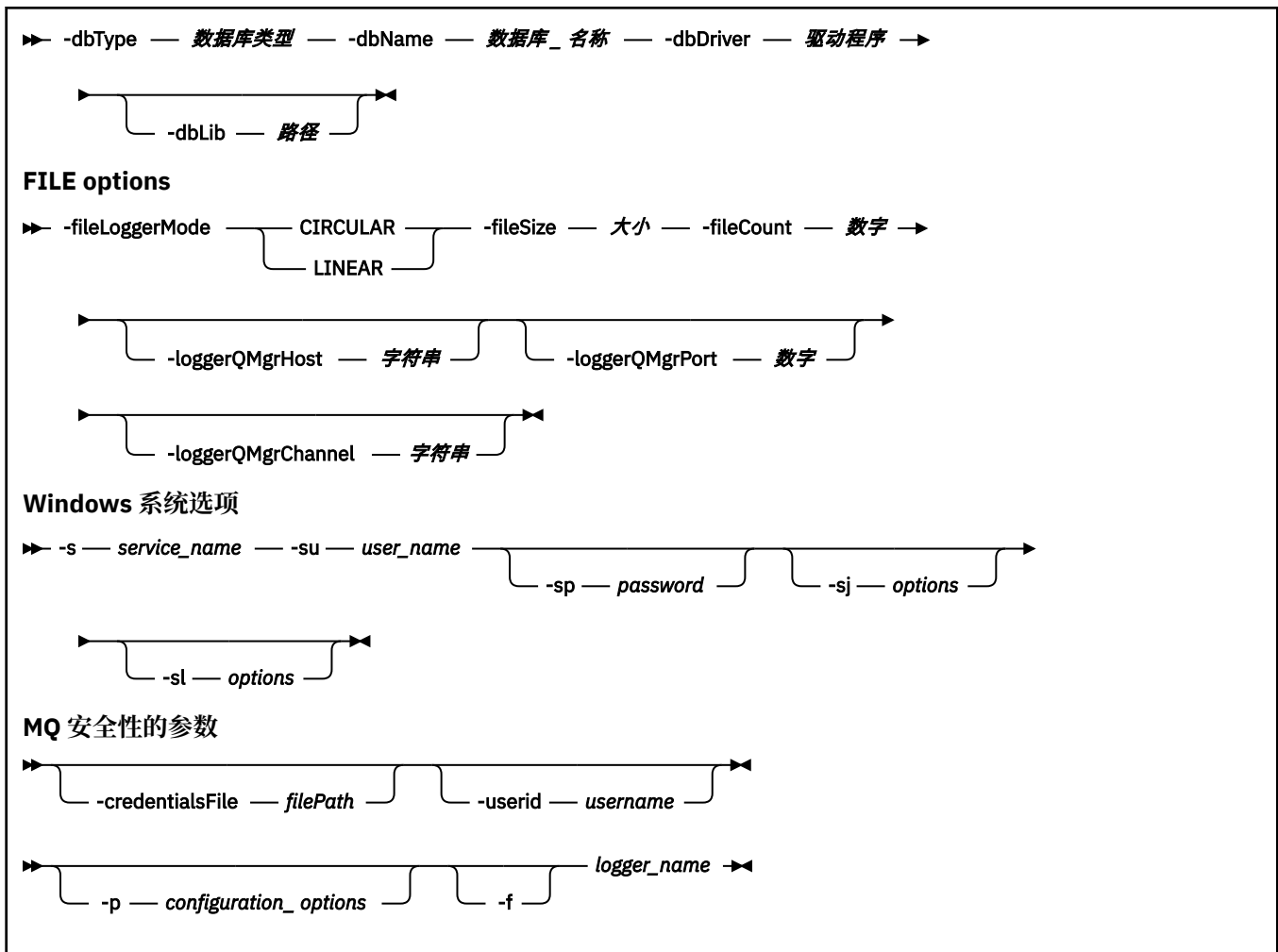
注：如果要创建的记录器是数据库记录器，并且它不是连接到本地 Db2 数据库，那么您将需要手动创建 MQMFTCredentials.xml 文件。此文件包含用于连接到数据库的用户名和密码。您应使用 logger.properties 文件中的属性文件 wmqfte.database.credentials 来指定 MQMFTCredentials.xml 文件的路径。此凭证文件的样本位于 `MQ_INSTALLATION_PATH/mqft/samples/credentials/` 中。

特殊字符

请谨慎使用包含特殊字符的参数值，从而避免命令 shell 以您不期望的方式解析字符。例如，包含空格、引号（单或双）、正斜杠或反斜杠之类字符的标准文件路径和名称可能由命令 shell 进行解释，而不是直接传递到命令本身。为避免命令 shell 解释字符，请将整个参数括在双引号/单引号中，或者使用命令 shell 的转义序列来转义特殊字符。

语法





参数

-loggerType (type)

必需。指定受管文件传输信息将记录到的位置。如果传输信息将记录到数据库，那么类型的选项是 DATABASE，如果该信息将记录到文件，那么为 FILE。

-loggerQMGr (qmgr_name)

可选。确定为了接收包含受管文件传输相关信息的信息而要连接到的队列管理器。该队列管理器必须位于与该记录器相同的系统上。如果未指定 **-loggerQMGr** 参数，那么为此记录器设置的配置选项所关联的协调队列管理器将用作缺省值。

-dbType (database_type)

-loggerType 是 DATABASE 时为必需项。指定用于存储受管文件传输信息的数据库管理系统的类型。选项是 db2 或 oracle

注：您需要使用 SQL 文件创建表。 .sql 文件可从 MQ_INSTALLATION_PATH_/mqft/sql 中获取：

- 对于 Db2 数据库: ftelog_tables_db2.sql
- 对于 Oracle 数据库: ftelog_tables_oracle.sql

-dbName (database_name)

-loggerType 是 DATABASE 时为必需项。受管文件传输信息所存储到的数据库的名称。该数据库必须使用 Managed File Transfer 日志表进行配置。

-dbDriver (driver)

-loggerType 是 DATABASE 时为必需项。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。

-dbLib (path)

-loggerType 是 DATABASE 时为可选。选定数据库驱动程序所需的任何本机库的位置。

-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 [MFT 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。

-loggerQMGrHost

运行记录器队列管理器的机器的主机名或 IP 地址。

缺省值是“无”。

如果未指定 -loggerQMGrHost 参数，那么将采用绑定方式创建记录器。

-loggerQMGrPort

记录器队列管理器正在侦听的端口号。

缺省值为 1414。

-loggerQMGrChannel

用于连接到记录器队列管理器的通道的名称。

缺省值为 SYSTEM.DEF.SVRCONN。

Windows -s (service_name)

可选（仅限 Windows 系统）。指示该记录器将作为 Windows 服务运行。如果未指定 service_name，那么服务将命名为 mqmftLoggerLOGGERQMGR，其中 LOGGER 是记录器名称，而 QMGR 是记录器队列管理器名称。

服务的显示名称（显示在 Windows 服务窗口的名称列中）始终为 **Managed File Transfer Logger
LOGGER@QMGR**。

Windows **-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 域用户帐户运行记录器，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 **Log on as a service** 权限。有关如何授予此权限的信息，请参阅 [有关将 MFT 代理程序或记录器作为 Windows 服务运行的指南](#)。

在指定 **-s** 时为必需项。

Windows **-sp (password)**

可选（仅限 Windows）。**-su** 参数设置的用户帐户的密码。

该参数仅在指定 **-s** 时才有效。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。

Windows **-sj (options)**

可选（仅限 Windows）。当记录器作为 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。

Windows **-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 服务存在问题，那么此选项很有用。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。

-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 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。

此命令支持将一组 IBM MQ 认证详细信息添加到指定的 Managed File Transfer 凭证文件中。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。

-你说什么 或 -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
```

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 Oracle 数据库：

```
fteCreateLogger -loggerType DATABASE -dbName FTEDB -dbType oracle
                -dbDriver "C:\app\oracle\product\12.1.0\dbhome_2\jdbc\lib\ojdbc7.jar" dblogger1
```

在此示例中，将使用主机名及缺省端口和通道来创建客户机方式文件记录器：

```
fteCreateLogger -loggerType FILE -loggerQMgr CORDQM -loggerQMgrHost cordqm.ibm.com
                -fileLoggerMode CIRCULAR -fileSize 10MB -fileCount 10 FL1
```

在此示例中，将使用主机名、端口和通道来创建客户机方式文件记录器：

```
fteCreateLogger -loggerType FILE -loggerQMgr CORDQM -loggerQMgrHost cordqm.ibm.com
                -loggerQMgrPort 4444 -loggerQMgrChannel LOGGER_CHANNEL -fileLoggerMode CIRCULAR -fileSize 10MB
                -fileCount 10 FL1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

fteCreateMonitor: 创建 MFT 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 Managed File Transfer 监视资源（例如，某个目录的内容），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 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，并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。此外，您可以使用 **-f** 和 **-c** 参数动态覆盖监视器配置。

注：对于可以在代理程序上创建的资源监视器数量没有限制，并且所有资源监视器都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。请参阅 [MFT 资源监视概念](#) 以获取更多信息。

fteCreateMonitor 命令在协议网桥代理上不受支持。

提示：您还可以使用 **fteListMonitors** 命令将资源监视器配置导出到 XML 文件：

- 将 **fteListMonitors** 命令与 **-ox** 配合使用将导出单个资源监视器的定义。
- **V9.0.5** 从 IBM MQ 9.0.5 开始，将 **fteListMonitor** 命令与 **-od** 配合使用，将多个资源监视器定义导出到指定的目录。您还可以使用 **-od** 选项将单个资源监视器定义导出到指定的目录。

有关 **fteListMonitors** 命令的更多信息，请参阅第 397 页的『[fteListMonitors: 列出 MFT 资源监视器](#)』。

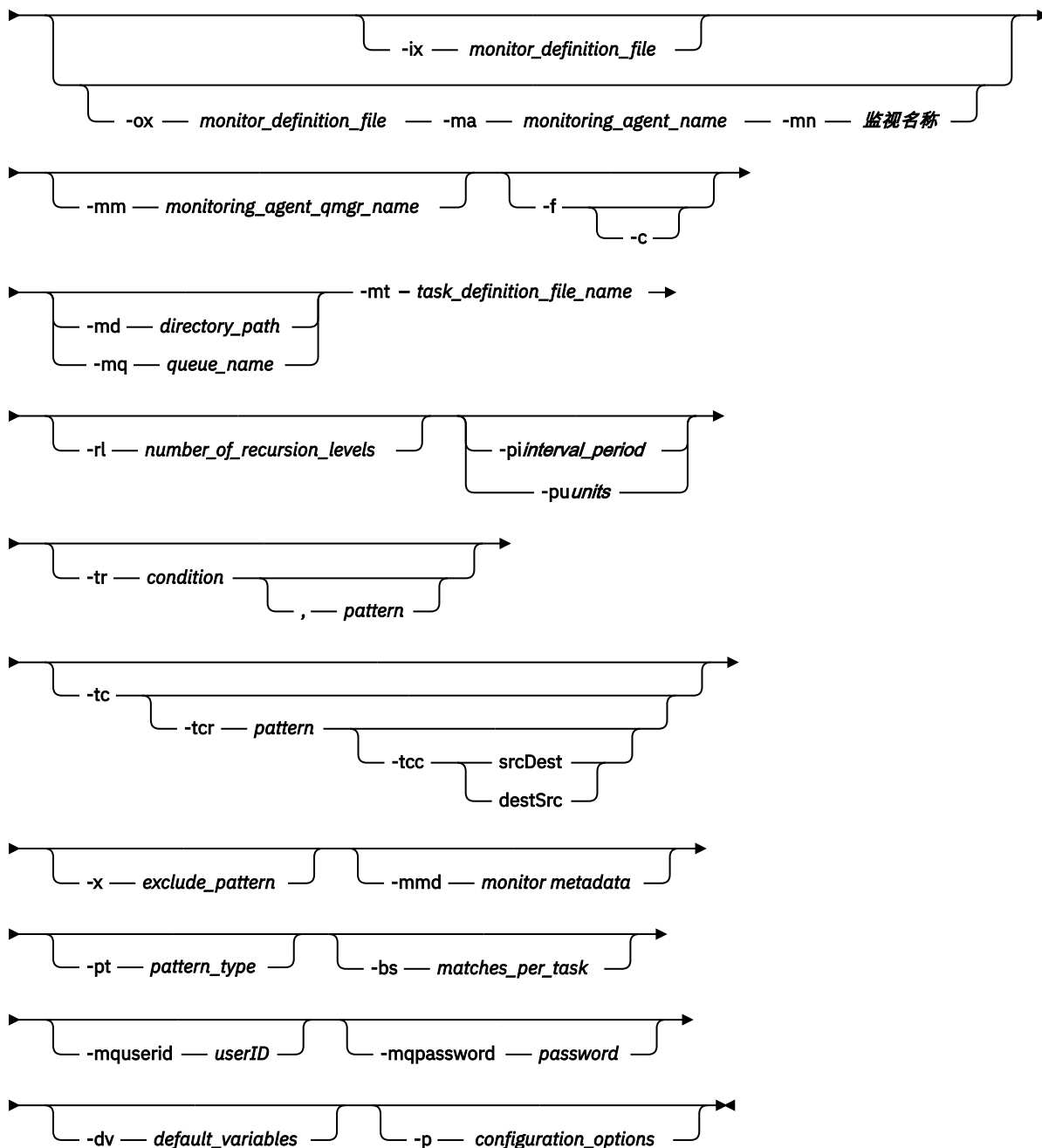
特殊字符

请谨慎使用包含特殊字符的参数值，从而避免命令 shell 以您不期望的方式解析字符。例如，包含空格、引号（单或双）、正斜杠或反斜杠之类字符的标准文件路径和名称可能由命令 shell 进行解释，而不是直接传递到命令本身。为避免命令 shell 解释字符，请将整个参数括在双引号/单引号中，或者使用命令 shell 的转义序列来转义特殊字符。

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 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，如果您选择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。此参数清除更新的历史记录，这使资源监视器重新检查触发条件。此参数只能与 **-f** 参数一起使用。

-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。该队列必须在监视代理队列管理器上。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)

必需。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。有关更多信息，请参阅第 133 页的『使用传输定义文件』。传输定义 XML 文档的路径必须在您从其运行 **fteCreateMonitor** 命令的本地文件系统中。如果未指定文件路径，那么该命令将在当前工作目录中查找此文件。除非您要使用 **-ix** 或 **-ox** 参数，否则 **-mt** 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。

V 9.0.1 在运行 **fteCreateMonitor** 命令时，也可以将传输恢复超时 **-rt** 参数与 **-gt** 参数一起使用。您可以设置时间量（以秒为单位），在此期间，源代理会一直重试恢复停止的传输。然后，恢复超时参数将包含在带有监视器使用的传输定义的 XML 文档。有关如何设置此参数的更多信息，请参阅 **fteCreateTransfer** 命令。

z/OS 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。

IBM i 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，在类似于以下示例的目录结构中（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**，Managed File Transfer 仅向下搜索到 C:\wmqfte\monitor\reports\2009 目录及其同级目录。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，如果匹配项是 *.go 且存在文件 LONDON.go 和 MANCHESTER.go，那么将为 LONDON.go 执行该任务，并为 MANCHESTER.go 执行另一任务。

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。

模式是使用通配符或 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** 可指定模式的格式。

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
```

仅当您还指定了 `-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 IBM MQ 消息，即表示满足该条件。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 IBM MQ 消息组是完整的，即表示满足该条件。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。

仅当您还指定了 `-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。将文件路径指定为 `source file path` 或 `source file path,destination file path`。此参数仅对目录监视触发器 `match` 和 `noSizeChange` 可用。

-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 156 页的『使用触发器文件』。

-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。

模式是使用通配符或 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。

-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。每个名称对由 `name=value` 组成。您可以在一条命令中多次使用 `-mmd` 参数。

-pt (pattern_type)

可选。由 `-tr` 和 `-x` 参数使用的模式的类型。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

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 549 页的『MFT 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 为 `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。此外，还必须指定 **-mquserid** 参数。如果指定了 **-mquserid**，但未指定 **-mqpassword**，那么系统会提示您提供相关的密码。将不显示此密码。

-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 149 页的『使用变量替换定制 MFT 资源监视器任务』](#)。仅当同时指定了 **-mq** 参数时，才能指定 **-dv** 参数。

-你说什么 或 -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，该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。在满足触发条件（即，在目录 `C:\wmqfte\monitors` 中存在大于 5MB 的文件）的情况下，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
```

在本例中，将创建使用代理 AGENT1 且名为 MONITOR1 的资源监视器，以传输大于 5 MB 的文件；并将该资源监视器导出到 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，触发器最初错误地设置为监视 `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。

fteCreateTemplate: 创建新的文件传输模板

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 IBM 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。

您可以从可以连接到 IBM MQ 网络的任何系统运行 **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 Managed File Transfer，并将 Managed File Transfer 组件配置为与 IBM MQ 网络进行通信。

此命令使用 `command.properties` 文件来连接到 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 526 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 363 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。该文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` 目录。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`。

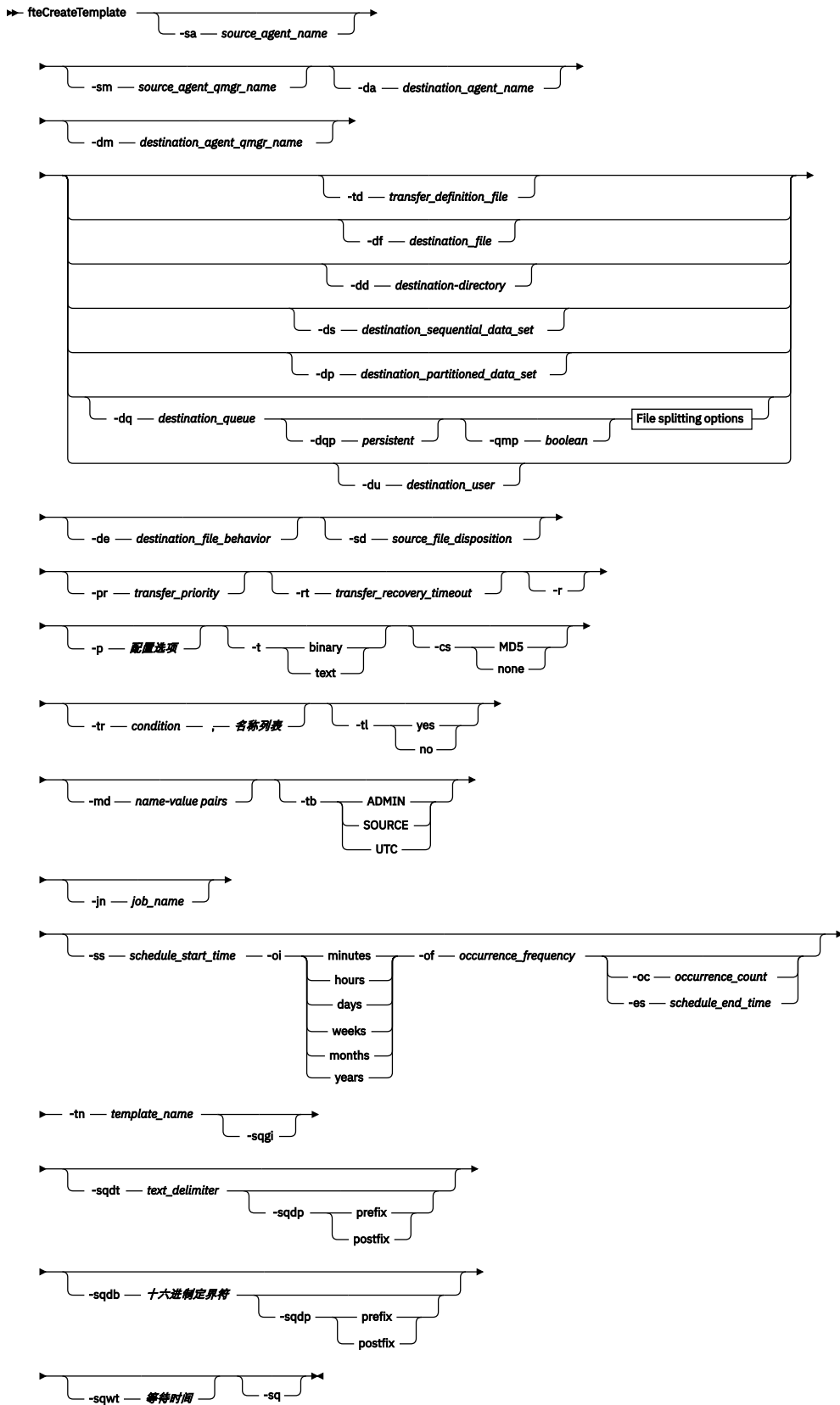
z/OS 在 z/OS 上，缺省情况下，会将当前运行代理所使用的用户名作为高级限定符前缀添加到非标准的数据集规范。例如：`//ABC.DEF`。要更改作为前缀添加到数据集名称的值，请在 `agent.properties` 文件中设置 `transferRootHLQ` 属性。该文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` 目录。将以下行添加到该文件：

```
transferRootHLQ=prepend_value
```

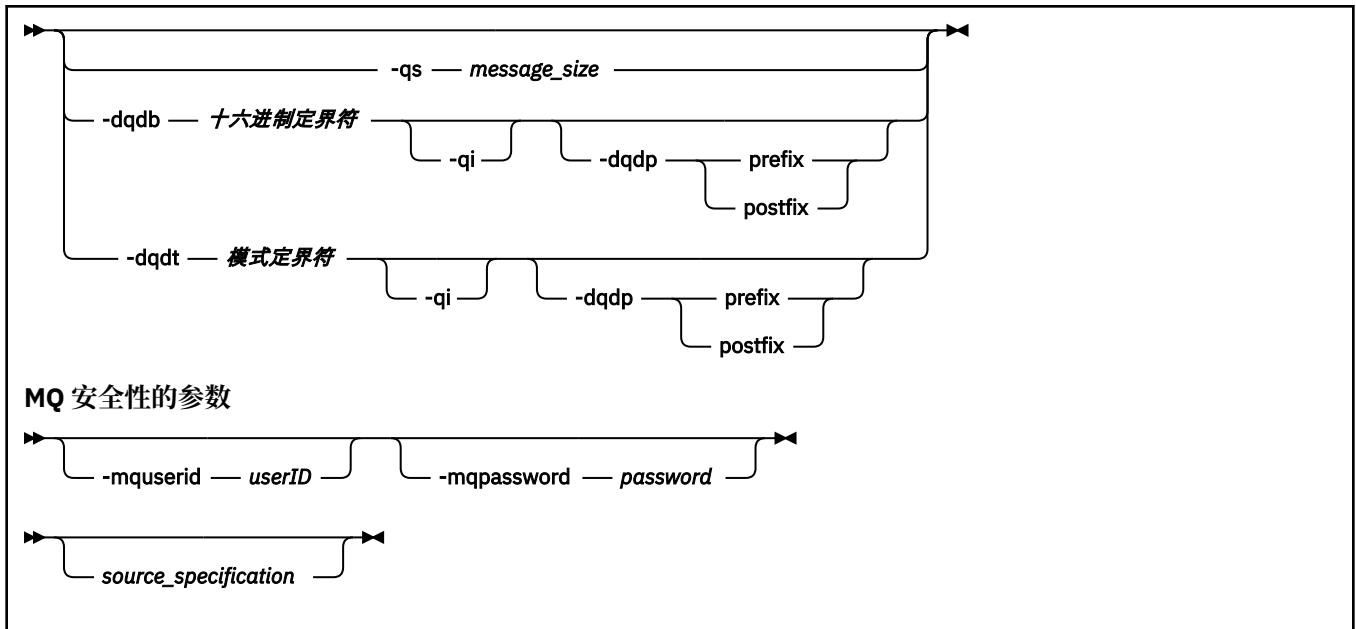
z/OS 然而，对于涉及 z/OS 系统上的 Connect:Direct 节点的传输，数据集规范将解释为标准名称。不会向数据集名称添加任何高级限定符。

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 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。

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。


z/OS 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。您无法将传输定义文件存储在 z/OS 顺序文件或 PDS 成员中。

IBM i 在 IBM i 上，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 destination_file

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 destination_directory

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，并且目标是 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 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。

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，那么还必须利用覆盖值指定 **-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 530 页的『与 Connect:Direct 节点之间传输数据集』。

对于仅涉及 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 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，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 460 页的『MFT 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 536 页的『不能与 MFT 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。

z/OS **-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。

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针对 **-ds** (*destination_sequential_data_set*) 参数的描述相同。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，那么您还必须使用覆盖值指定 **-de** 参数。

当目标代理为协议网桥代理时，不支持 **-dp** 参数。

-du destination_user

可选。将文件传输到其中的目标文件空间的用户的名称。

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 destination_queue

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 IBM 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 boolean

可选。指定由传输写入到目标队列的第一条消息是否已设置 IBM 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 564 页的『写入到目标队列的消息上由 MFT 设置的 MQ 消息属性』。

-qs message_size

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 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。消息的大小由 *message_size* 的值指定。*message_size* 的格式是长度单位，其中长度是正整数值，而单位是以下某个值：

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 hexadecimal_delimiter

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 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。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 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。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。此参数的值将求值为 Java 正则表达式。有关更多信息，请参阅第 549 页的『MFT 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 461 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 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，那么不能指定 **-td** 参数，因为这些参数是互斥的。

-sd source_file_disposition

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。

z/OS 在 z/OS 上，如果源是磁带数据集并且您指定了 delete 选项，那么将重新安装磁带以删除该数据集。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 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 199 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 transfer_priority

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 IBM 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 0，从而允许 IBM MQ 消息通信获得优先。

V 9.0.1 -rt transfer_recovery_timeout

可选。设置时间量（以秒为单位），在此期间，源代理会一直尝试恢复停止的文件传输。指定以下某个选项：

-1

代理继续尝试恢复停止的传输，直至传输完成。使用此选项相当于未设置属性时代理的缺省行为。

0

一旦进入恢复，代理将停止文件传输。

>0

在由指定正整数值设置的时间量（以秒为单位）内，代理继续尝试恢复停止的传输。例如，

```
-rt 21600
```

表示代理会在进入恢复后六小时内一直尝试恢复传输。此参数的最大值为 999999999。

通过这种方式指定传输恢复超时值是在每个传输的基础上进行设置。要为 Managed File Transfer 网络中的所有传输设置全局值，您可以在 [agent.properties](#) 文件中添加一个属性。

-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。当向 Managed File Transfer 提供通配符作为 *source_specification* 时，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 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。

z/OS 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。

z/OS 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，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。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。

 在 z/OS 上，空格表示定界符，因此您必须使用下划线来分隔值。例如，使用 `kw=text1_text2_text3` 而非 `kw="text1 text2 text3"`

-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 IBM MQ 组标识将消息分组。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。可将字符串字面值的 Java 转义序列包含在定界符中。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 545 页的『对 MFT 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

z/OS 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。

z/OS 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）。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。

-你说什么或 -h

可选。显示命令语法。

示例

在本示例中，创建称为 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
```

在本示例中，创建称为 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
```

在本示例中，创建称为 `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
```

z/OS 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
```

z/OS 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。

fteCreateTransfer: 启动新的文件传输

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** 命令可从 Managed File Transfer 代理创建并启动新的文件传输。

注: 只能在同一 Managed File Transfer 拓扑中的代理之间进行文件传输。

有关如何传输文件的指导信息，请参阅第 526 页的『文件传输准则』。对于 z/OS 平台，您可以传输文本文件，数据集和生成数据组 (GDG)。

您可以从可以连接到 IBM 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，要运行此命令，您必须在此系统上安装 Managed File Transfer 组件（服务或代理），并将此系统上的 Managed File Transfer 组件配置为与 IBM MQ 网络进行通信。

该命令使用名为 `command.properties` 的属性文件连接到 IBM 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，将生成错误。有关更多信息，请参阅第 457 页的『MFT `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 209 页的『将 Apache Ant 与 MFT 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

`ftCreateTransfer` 命令支持使用相对文件路径。对于以下平台，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：

- **Multi** 多平台
- **z/OS** z/OS 上的 UNIX 系统服务

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。该文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` 目录。将以下行添加到该文件：

```
transferRoot=directory_name
```

Windows 例如，将 `C:\TransferRoot` 指定为 `C:\\TransferRoot` 或 `C:/TransferRoot`。

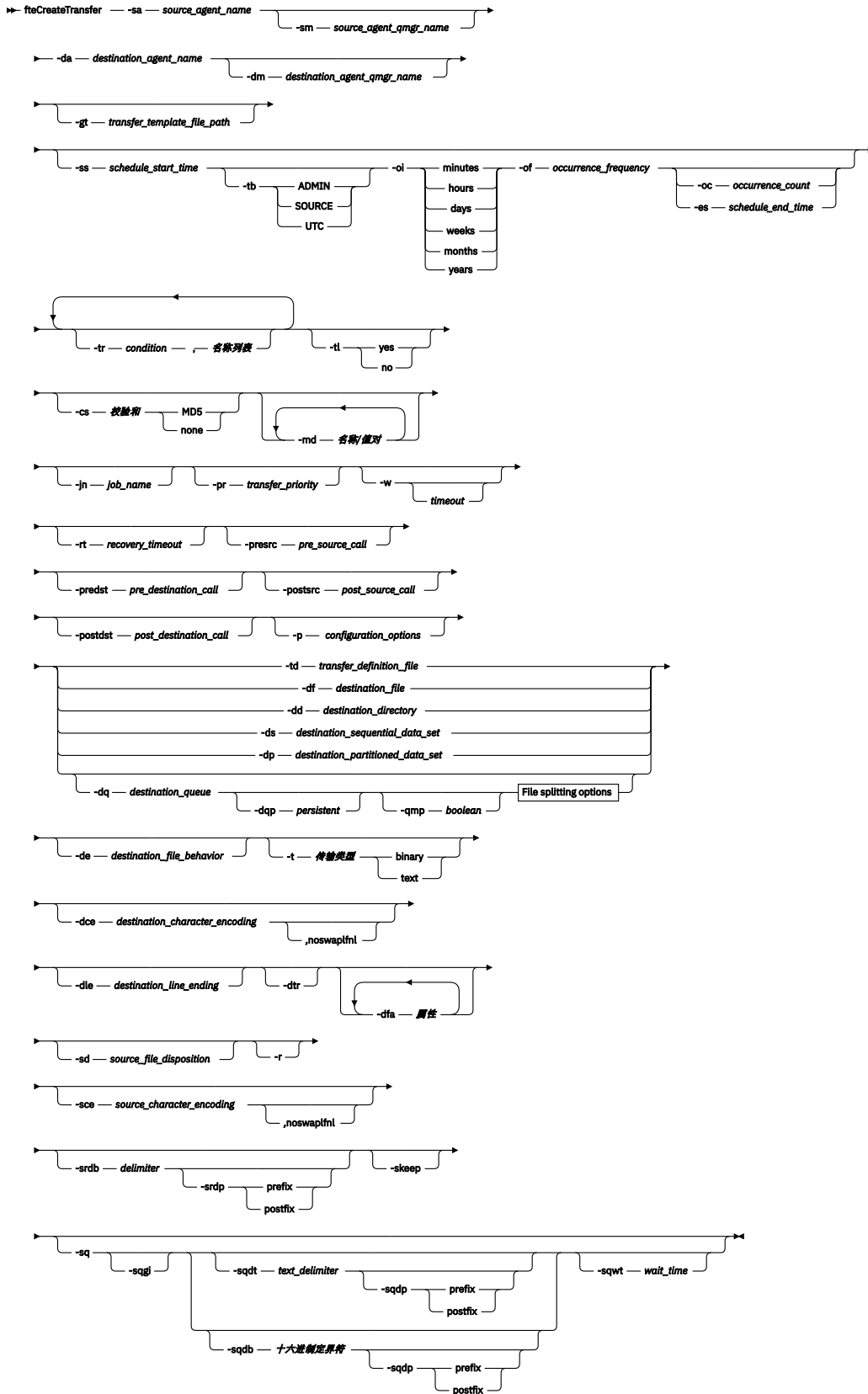
z/OS 在 z/OS 上，缺省情况下，会将运行代理所使用的用户名作为高级限定符前缀添加到非标准的数据集规范。例如：`//ABC.DEF`。要更改作为前缀添加到数据集名称的值，请在 `agent.properties` 文件中设置 `transferRootHLQ` 属性。该文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` 目录。将以下行添加到该文件：

```
transferRootHLQ=prepend_value
```

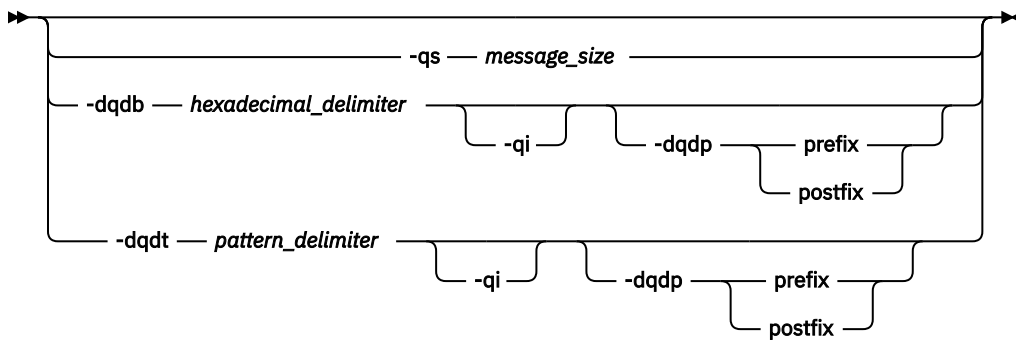
z/OS 然而，对于涉及 z/OS 系统上的 Connect:Direct 节点的传输，数据集规范将解释为标准名称。不会向数据集名称添加任何高级限定符。

语法

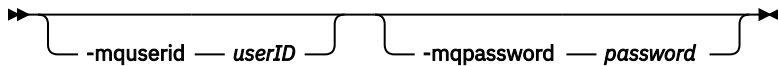
fteCreateTransfer



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。

z/OS 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。

如果您指定 **-td** 参数并且传输定义文件（文件中包含要用于传输的源代理），那么请勿指定 **-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，那么要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）。如果无法找到源代理程序的 `agent.properties` 文件，那么文件传输将失败。

-da destination_agent_name

必需。要将文件传输到的代理程序的名称。

如果您指定 **-td** 参数并且传输定义文件（文件中包含要用于传输的目标代理），那么请勿指定 **-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，那么要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）。如果无法找到目标代理的 `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 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

z/OS 在 z/OS 上，您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。您无法将传输模板文档存储在 z/OS 顺序文件或 PDS 成员中。

IBM i 在 IBM i 上，您必须将传输模板文档存储在集成文件系统中。

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** *destination_file*
- **-cs** 校验和
- **-de** *destination_file_behavior*
- **-dq** *destination_queue*
- **-t** 传输类型
- **-sd** *source_file_disposition*

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 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。

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，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：

```
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。

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 *job_name*

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 477 页的『`enableUserMetadataOptions`：支持的 MFT 用户定义的元数据键』。当 `enableUserMetadataOptions` 属性设置为 `true` 时，以 `com.ibm.wmqfte.` 开头的键名称不支持用户定义的法。

-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，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。

-pr *transfer_priority*

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 IBM 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 IBM MQ 消息通信获得优先。

-qmp boolean

可选。指定由传输写入到目标队列的第一条消息是否已设置 IBM 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 564 页的『写入到目标队列的消息上由 MFT 设置的 MQ 消息属性』。

-qs message_size

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 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。消息的大小由 *message_size* 的值指定。*message_size* 的格式是长度单位，其中长度是正整数值，而单位是以下某个值：

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。您只能指定 **-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 configuration_options

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 timeout

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。

V 9.0.1

-rt recovery_timeout

可选。设置时间量（以秒为单位），在此期间，源代理会一直尝试恢复停止的文件传输。指定以下某个选项：

-1

代理继续尝试恢复停止的传输，直至传输完成。使用此选项相当于未设置属性时代理的缺省行为。

0

一旦进入恢复，代理将停止文件传输。

>0

在由指定正整数值设置的时间量（以秒为单位）内，代理继续尝试恢复停止的传输。例如，

```
-rt 21600
```

表示代理会在进入恢复后六小时内一直尝试恢复传输。此参数的最大值为 999999999。

通过这种方式指定传输恢复超时值是在每个传输的基础上进行设置。要为 Managed File Transfer 网络中的所有传输设置全局值，您可以在 [agent.properties](#) 文件中添加一个属性。

调用程序的参数

有关如何从 Managed File Transfer 启动程序的更多信息，请参阅第 207 页的『指定要使用 MFT 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 723 页的『使用 `fteCreateTransfer` 来启动程序的示例』。

-presrc pre_source_call


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** 和 **jcl**。缺省值为 **executable**。

 **jcl** 值仅适用于以 z/OS 环境中代理为目标的情况下。在此情况下，该命令引用 ZFS 文件、QSAM 可读数据集或 PDS 成员。内容应为可提交的 JCL。

commandspec

必需。命令规范。请使用以下格式之一：

- 类型 **executable**: `command[(arg1,arg2,...)]`

如果自变量包含变量替换，例如 `${FilePath}` 或 `${FileName}`，那么这些变量将替换为传输请求中的第一个项。

例如，如果传输请求由文件 "reports01.csv, reports02.csv, reports03.csv" 组成，并且目标目录为 "/output"，那么以下传输请求：

```
fteCreateTransfer -sa 1 -da 2 -presrc "executable:archive(${FileName})"
-dd TargetDir "${FilePath}" -gt task.xml
```

替换为

```
fteCreateTransfer -sa 1 -da 2 -presrc "executable:archive(reports01.csv)"
-dd TargetDir "/ouptut" -gt task.xml
```

- 类型 **antscript**: `command[(name1=var1|target1,name2=var2|target2,...)]`
- 类型 **jcl**: `command`

其中：

命令

必需。要调用的程序的名称。

jcl 值仅适用于以 z/OS 环境中代理为目标的情况下。

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）。

successrc

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。

值

必需。有效整数。

-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。

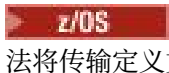
指定目标的参数


必须指定 **-td**、**-df**、**-dd**、**-ds**、**-dq** 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，包含受管传输请求的 XML 文档的名称（可能已由 **-gt** 参数生成）。如果指定 **-td** 参数并同时在命令行上指定任何其他参数，那么这些其他参数将覆盖来自于传输定义文件中的相应值。

fteCreateTransfer 命令查找与您的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。


 **z/OS** 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。您无法将传输定义文件存储在 z/OS 顺序文件或 PDS 成员中。

 **IBM i** 在 IBM i 上，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 destination_file

可选。目标文件的名称。

如果目标代理是 Connect:Direct 网桥代理，那么以 `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 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。
-  当目标代理位于 z/OS 上时，如果指定的文件以 // 开头，将假定此文件是 z/OS 分区数据集。

-dd destination_directory

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，并且目标是 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_server` 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 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。
-  当目标代理位于 z/OS 上时，如果指定的文件以 // 开头，将假定此文件是 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 526 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，那么还必须利用覆盖值指定 **-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 530 页的『与 Connect:Direct 节点之间传输数据集』。

对于仅涉及 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： **z/OS** 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 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 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 460 页的『MFT agent.properties 文件』。

z/OS 使用 **fteCreateTemplate** 命令、**fteCreateTransfer** 命令或 agent.properties 文件中的 **bpxwdynAllocAdditionalProperties** 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 536 页的『不能与 MFT 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。

如果想要调用针对代理定义的任何 Managed File Transfer 传输 I/O 用户出口，那么请勿在传输中指定 **-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 Managed File Transfer I/O。

z/OS **-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针对 **-ds (destination_sequential_data_set)** 参数的描述相同。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，那么您还必须使用覆盖值指定 **-de** 参数。

当目标代理为协议网桥代理时，不支持 **-dp** 参数。

如果想要调用针对代理定义的任何 Managed File Transfer 传输 I/O 用户出口，那么请勿在传输中指定 **-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 Managed File Transfer I/O。

-dq destination_queue

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。

如果想要调用针对代理定义的任何 Managed File Transfer 传输 I/O 用户出口，那么请勿在传输中指定 **-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 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 **-dqp** 参数。

-dqdb hexadecimal_delimiter

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 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。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。

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要使用的 Java 正则表达式。所有消息都具有相同的 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。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 549 页的『MFT 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` 代理属性来更改此行为。有关更多信息，请参阅第 461 页的『高级代理属性』。

只有在指定了 **-dq** 参数以及 **-t** 参数的 `text` 值时，您才可以指定 **-dqdt** 参数。您只能指定 **-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：

prefix

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** 参数指定用于转换的代码页和行尾。如果未指定 **-sce**、**-dce** 或 **-dle** 参数，那么执行的精确转换取决于源代理程序和目标代理程序的操作系统。

z/OS 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。

z/OS 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。

-dce destination_character_encoding

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 570 页的『MFT 的可用代码页』。

noswaplfnl

缺省情况下，Managed File Transfer 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。该选项仅适用于文本文件，因此还必须指定 **-t text** 参数。有效选项为：

LF

换行。它是下列平台的缺省值：

- **UNIX** UNIX 平台
- **z/OS** z/OS UNIX 系统服务文件

在将 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。 **Windows** 这是 Windows 的缺省值。

z/OS 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。当传输到在 4690 上运行的 IBM MQ 8.0 Managed File Transfer 代理时，此参数用于指定与传输中的目标文件相关联的文件属性的分号分隔列表。可以指定具有或没有值的 **-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)
```

您可以在一条命令中多次使用 **-dfa** 参数。

有关 4690 上的文件属性的更多信息，请参阅 IBM Documentation 中的 [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。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。

指定源的参数

-sd source_file_disposition

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您指定了 **delete** 选项，那么将重新安装磁带以删除该数据集。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 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 199 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。

-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。如果为 Managed File Transfer 提供通配符作为 *source_specification*，那么仅在指定 **-r** 参数时才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 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。

-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 570 页的『MFT 的可用代码页』。

noswaplfnl

缺省情况下，Managed File Transfer 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。

z/OS -skeep

可选。指定在文本方式传输过程中，在从面向记录的固定长度格式文件（例如，z/OS 数据集）读取的源记录上保留尾部空格。如果未指定该参数，将从源记录移除尾部空格。

z/OS -srdb delimiter

可选。对于面向记录的源文件（例如，z/OS 数据集），指定在将记录追加到二进制文件时要作为定界符插入的一个或多个字节值。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。

z/OS -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。

如果想要调用针对代理定义的任何 Managed File Transfer 传输 I/O 用户出口，那么请勿在传输中指定 **-sq** 参数。使用 **-sq** 参数将阻止针对源调用传输 I/O 用户出口，并且意味着将使用标准 Managed File Transfer I/O 代替。

-sqgi

可选。指定按 IBM MQ 组标识将消息分组。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 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。可将字符串字面值的 Java 转义序列包含在定界符中。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。

只有在指定了 **-sq** 参数以及 **-t** 参数的 **text** 值时，您才可以指定 **-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**。

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。只有在指定 **-sqdt** 和 **-sqdb** 参数之一时，才可以指定 **-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。

有关使用 **-sqwt** 参数的信息，请参阅第 569 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。

必需（如果指定 **-df**、**-dd**、**-dp**、**-dq** 或 **-ds** 参数之一）。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 545 页的『对 MFT 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。然而，如果针对 **-df** 或 **-ds** 参数指定多个源规范，并且还指定了 **-de overwrite**，那么目标将仅包含最后指定的源文件的数据。如果未指定 **-de overwrite**，那么传输仅可能部分成功。如果目标文件先前不存在，那么它将包含您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/***

z/OS 连续数据集

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。

z/OS 分区数据集

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)。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。

- 如果指定了 **-sq** 参数，那么 *source_specification* 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。

- **z/OS** 如果源代理在 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

-你说什么 或 -h

可选。显示命令语法。

示例

在此基本示例中，文件 **originalfile.txt** 从 AGENT1 传输到相同系统上的 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` 将从 AGENT1 传输到相同系统上的 AGENT2，直至目录 `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` 将从 AGENT1 的系统传输到 AGENT2 的系统。根据源代理系统的系统时间，文件传输安排在 09:00 进行，每两小时进行四次：

```
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` 将从 AGENT1 传输到 AGENT2，条件是 AGENT1 上存在文件 `A.txt`：

```
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
```

z/OS 在此示例中，文件 `originalfile.txt` 将从 AGENT1 的系统传输到 AGENT2 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
```

z/OS 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，将使用目标代理 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 的当前工作目录中以 `.txt` 结尾的所有文件都将传输到 AGENT2 上的目录 `C:\import` 中。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 的传输根目录中以 `.txt` 结尾的所有文件都将传输到 AGENT2 上的目录 `C:\import` 中。文件名保持不变。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -dd C:\import "*.txt"
```

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。

返回码	描述
21	在确定传输结果前, fteCreateTransfer 命令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。

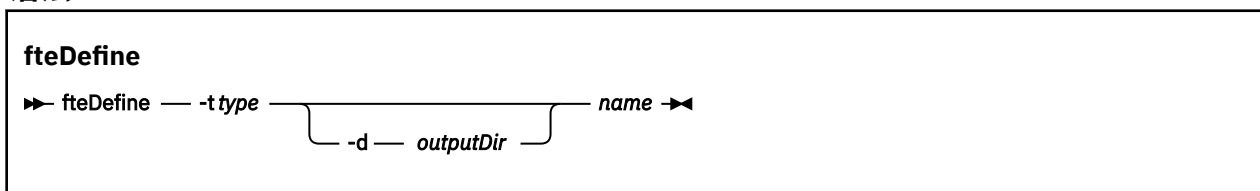
fteDefine (生成配置脚本)

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



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

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...*

-你说什么或 -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(99999999) +
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(99999999) +
SHARE +
NOTRIGGER +
USAGE(NORMAL) +
REPLACE
...
etc.
```

在此示例中，**fteDefine** 命令与 **-d outputDir** 参数和多个代理程序名称一起指定。

```
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 384 页的『**fteDelete** (生成用于除去配置脚本文本)』

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。

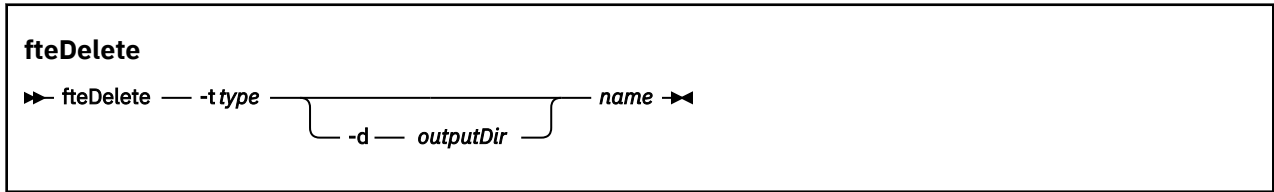
fteDelete (生成用于除去配置脚本文本)

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



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

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...`

-你点什么 或 -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 383 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

fteDeleteAgent 命令将删除 Managed File Transfer Agent 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。

如果您已将代理配置为作为 Windows 服务运行，那么运行 **fteDeleteAgent** 命令将删除服务定义。

V 9.0.4 从 IBM MQ 9.0.4 开始，将在删除代理时移除任何资源监视器和调度的传输。

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。

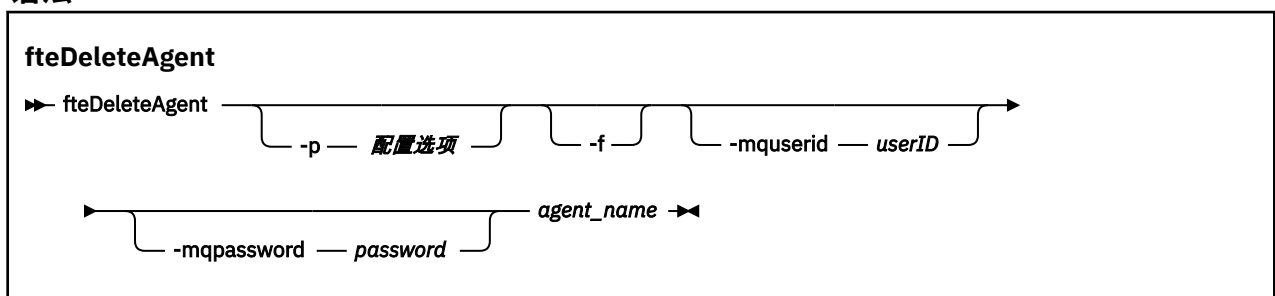
fteDeleteAgent 命令向您提供 MQSC 命令，您必须针对代理的队列管理器运行这些命令以清除和删除代理的系统队列。这些队列如下所示：

- 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

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 (configuration_options)

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 49 页的『多平台上的 MFT 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。

-你说什么 或 -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 445 页的『[fteStopAgent](#)』

使用 **fteStopAgent** 命令以受控方式停止 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 319 页的『[fteCleanAgent: 清除 MFT 代理](#)』

使用 **fteCleanAgent** 命令，可通过删除 Managed File Transfer Agent 使用的持久性和非持久性队列中的消息来清除该代理使用的队列。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 323 页的『[fteCreateAgent \(创建 MFT 代理\)](#)』

fteCreateAgent 命令创建 Managed File Transfer Agent 及其关联的配置。

第 442 页的『[fteStartAgent](#)』

fteStartAgent 命令从命令行启动 Managed File Transfer 代理程序。

fteDeleteLogger

使用 **fteDeleteLogger** 命令可删除 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: **ULW** 在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理员（和 mqm 组成员）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。

z/OS 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）。
- **V9.0.0.1** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V9.0.0.1** 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。

如果您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*

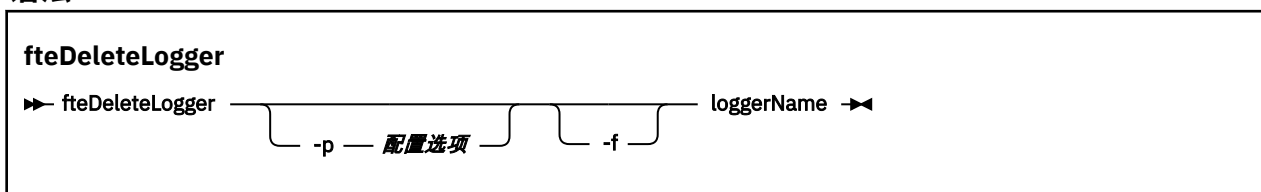
预订名称如下所示：

- 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。

-你说什么 或 -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 340 页的『[fteCreate 记录器 \(创建 MFT 文件或数据库记录器\)](#)』

使用 **fteCreateLogger** 命令可创建 Managed File Transfer 文件或数据库记录器。

第 444 页的『[fteStartLogger](#)』

fteStartLogger 命令启动 Managed File Transfer 日志记录应用程序。

第 447 页的『[fteStopLogger](#)』

fteStopLogger 命令停止 Managed File Transfer 记录器。

第 411 页的『[fteModify 记录器 \(将 MFT 记录器作为 Windows 服务运行\)](#)』

使用 **fteModifyLogger** 命令来修改 Managed File Transfer 记录器，以便可以将其作为 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 MFT 资源监视器

使用 **fteDeleteMonitor** 命令可通过命令行停止和删除现有 Managed File Transfer 资源监视器。针对资源监视代理发出此命令。

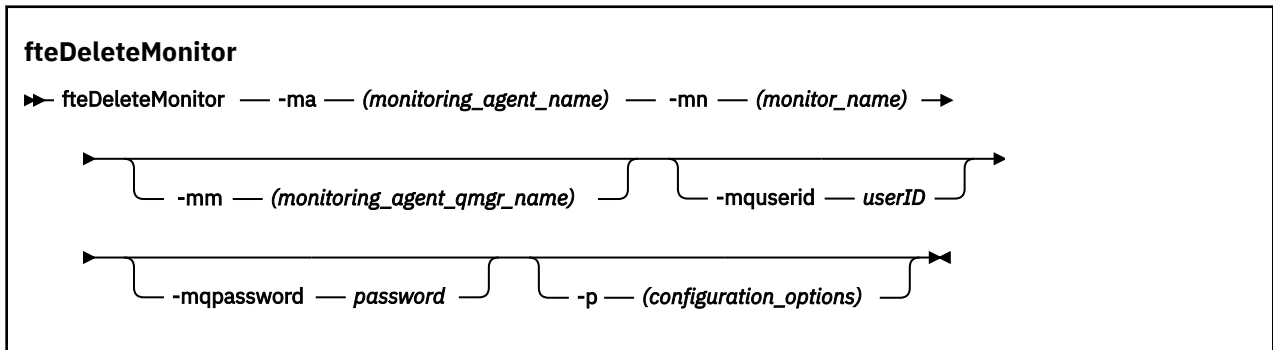
用途

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 IBM MQ 网络并随后路由到代理的队列管理器的任何系统运行 **fteDeleteMonitor** 命令。具体而言，要运行此命令，您必须已在该系统上安装了 Managed File Transfer 组件（服务或代理），并且将该系统的 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

-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 (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。

-你说什么 或 -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。

相关任务

第 141 页的『[监视 MFT 资源](#)』

您可以监视 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 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 345 页的『[fteCreateMonitor: 创建 MFT 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 Managed File Transfer 监视资源（例如，某个目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

第 397 页的『[fteListMonitors: 列出 MFT 资源监视器](#)』

通过命令行，使用 **fteListMonitors** 命令可列出 Managed File Transfer 网络中的所有现有资源监视器。

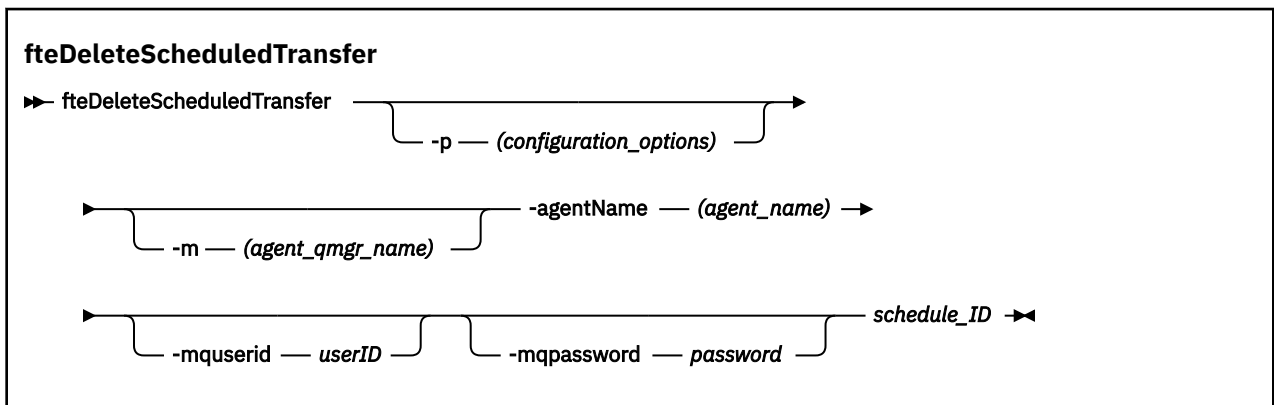
fteDeleteScheduledTransfer

用途

使用 **fteDeleteScheduledTransfer** 命令可删除先前使用命令行或 IBM MQ Explorer 创建的 Managed File Transfer 调度传输。

仅当您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 49 页的『[多平台上的 MFT 配置选项](#)』，以了解更多信息。

语法



参数

-p (configuration_options)

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。

-你说什么 或 -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 135 页的『[创建调度的文件传输](#)』

您可以从 IBM MQ Explorer 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 400 页的『[fteListScheduledTransfers](#)』

使用 `fteListScheduledTransfers` 命令可列出先前使用命令行或 IBM MQ Explorer 创建的所有 Managed File Transfer 传输。

fteDeleteTemplates

使用 `fteDeleteTemplates` 命令可从协调队列管理器中删除现有的 Managed File Transfer 模板。

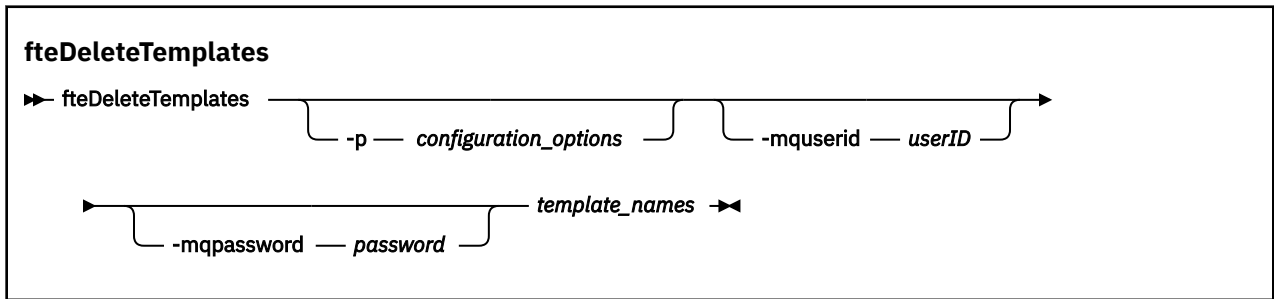
用途

`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。

您可以从可连接到 IBM MQ 网络并随后路由到协调队列管理器的任何系统运行 `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 Managed File Transfer，并将该系统的 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 (configuration_options)

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。

-你点什么 或 -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 162 页的『使用文件传输模板』

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。可以从命令行中使用 **fteCreateTemplate** 命令创建传输模板，也可以在 IBM MQ Explorer 中使用 **新建受管文件传输模板** 向导来创建传输模板，或者在创建文件传输时选中 **将传输设置保存为模板** 复选框来保存该模板。传输模板窗口显示在您的 Managed File Transfer 网络中已创建的所有传输模板。

第 163 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 351 页的『fteCreateTemplate: 创建新的文件传输模板』

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 401 页的『[fteListTemplates](#)』

fteListTemplates 命令可用于列出协调队列管理器上的可用 Managed File Transfer 传输模板。

fteDisplayVersion

使用 **fteDisplayVersion** 命令来显示您已安装的 Managed File Transfer 的版本。

用途

IBM 服务代表可能会要求您运行 **fteDisplayVersion** 命令以帮助确定问题。

语法




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。

 在 z/OS 上，**-v** 显示 **productId** 属性的值（如果已指定产品标识）。

-你说什么 或 -h

可选。显示命令语法。

未指定参数的示例

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，这是 IBM MQ 9.0 的输出：

```
5655-MFT, 5724-H72 Copyright IBM Corp. 2008, 2016. ALL RIGHTS RESERVED  
IBM MQ Components:  
Name:          IBM MQ Managed File Transfer  
Version:       9.0.0.0
```

指定了 -v 参数的示例

在本示例中，指定 **fteDisplayVersion** 命令时会指定 **-v** 参数。

```
fteDisplayVersion -v
```

此命令的输出包含有关产品版本的更多详细信息。例如：

```
5655-MFT, 5724-H72 Copyright IBM Corp. 2008, 2016. ALL RIGHTS RESERVED  
IBM MQ Components:  
Name:          IBM MQ Managed File Transfer
```

```
Version: 9.0.0.0
Level: p900-L160512.4
Platform: Windows 10 (10.0)
Architecture: amd64
JVM: JRE 1.8.0 Windows 10 amd64-64 Compressed References 20160427_301573 (JIT enabled, AOT
enabled)
J9VM - R28_Java8_SR3_20160427_1620_B301573
JIT - tr.r14.java.green.20160329_114288
GC - R28_Java8_SR3_20160427_1620_B301573_CMPRSS
J9CL - 20160427_301573
Product: C:\Program Files\IBM\MQ_1
Configuration: C:\ProgramData\IBM\MQ\mqft
```

```
Name: IBM MQ JMS Provider
Version: 9.0.0.0
Level: p900-L160512.4

Name: IBM MQLight Service for Bluemix JMS Provider
Version:
Level: p900-L160512.4

Name: Common Services for Java Platform, Standard Edition
Version: 9.0.0.0
Level: p900-L160512.4

Name: Java Message Service Client
Version: 9.0.0.0
Level: p900-L160512.4

Name: IBM MQ classes for Java Message Service
Version: 9.0.0.0
Level: p900-L160512.4

Name: IBM MQ classes for Java
Version: 9.0.0.0
Level: p900-L160512.4
```

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 MFT 代理

使用 **fteListAgents** 命令可列出向特定协调队列管理器注册的所有 Managed File Transfer 代理程序。

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 `bridge` 。
- 如果代理是 Connect:Direct 网桥代理，那么代理名称后会追加 (`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 453 页的『MFT coordination.properties 文件』](#)。

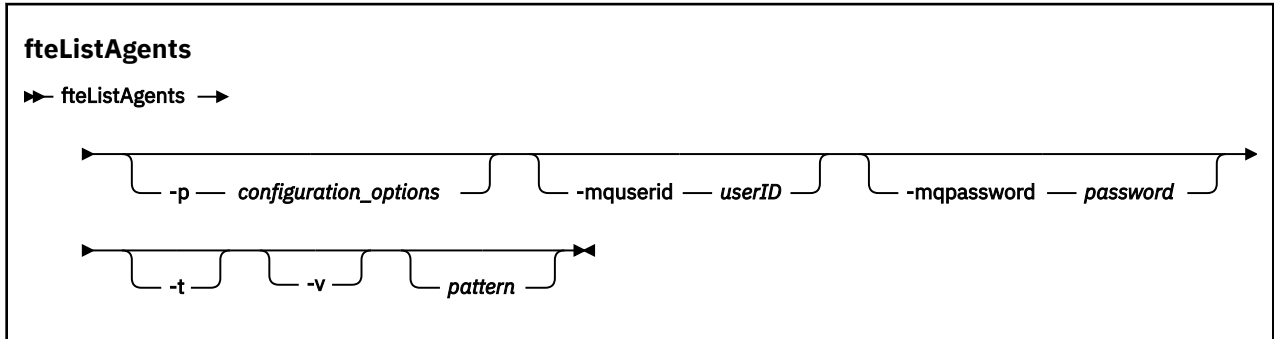
仅当希望使用不同于缺省集的配置选项集时，才对该命令指定可选的 **-p** 参数。有关更多信息，请参阅 [第 49 页的『多平台上的 MFT 配置选项』](#)。

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 620 页的『MFT 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。有关此属性的更多详细信息，请参阅主题 第 460 页的『MFT agent.properties 文件』。

语法



参数

-p (configuration_options)

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 620 页的『MFT 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。

-t

可选。指定简洁方式。对于 IBM MQ 9.0.2，缺省情况下输出包含 **Status Age** 列。如果不想看到 **Status Age** 信息，那么可使用 **-t** 参数发出命令以隐藏该列。有关更多信息，请参阅 **fteListAgents** 命令显示代理程序状态 UNKNOWN 时要执行的操作。

模式 (pattern)

可选。用于过滤 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，如果要特殊字符（如星号 (*) 和数字符号 (#）作为字面值处理，那么必须使用引号 (') 或双引号 (") 对其字符进行转义。如果未对这些字符进行转义，那么它们将根据其在特定 UNIX 或 Linux 系统上的含义进行解释。

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。

-你说什么 或 -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 521 页的『MFT 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 ( 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 176 页的『列出 MFT 代理』](#)

您可以使用命令行或 IBM MQ Explorer 列出向特定队列管理器注册的 Managed File Transfer 代理程序。

相关参考

[第 521 页的『MFT 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 433 页的『fteShowAgentDetails』](#)

使用 **fteShowAgentDetails** 命令可显示特定 Managed File Transfer Agent 的详细信息。这些是由代理的 Managed File Transfer 协调队列管理器存储的详细信息。

fteListMonitors：列出 MFT 资源监视器

通过命令行，使用 **fteListMonitors** 命令可列出 Managed File Transfer 网络中的所有现有资源监视器。

用途

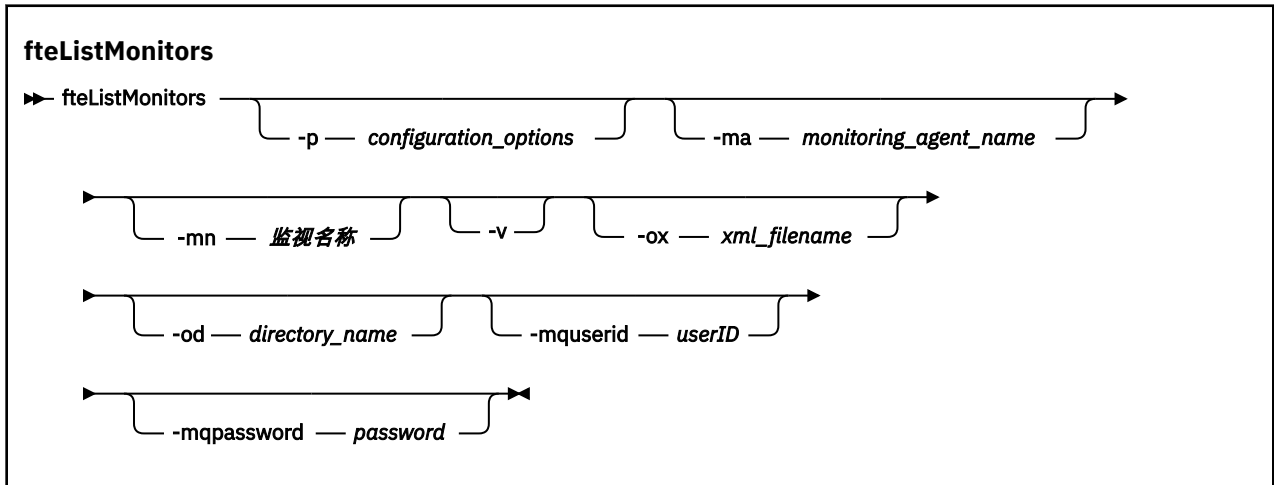
fteListMonitors 命令用于列出现有资源监视器。您可以通过指定代理名称和资源监视器名称来过滤命令输出。

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 453 页的『MFT coordination.properties 文件』。

您可以使用 `-ox` 参数将资源监视器导出到 XML 文件。有关如何使用此 XML 文件的更多信息，请参阅第 345 页的『fteCreateMonitor: 创建 MFT 资源监视器』。

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，请参阅配置选项。

语法



参数

`-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。

`-ma (monitoring_agent_name)`

可选。通过使用在输入时提供的模式，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。通过使用在输入时提供的模式，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。

`-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`，那么系统会提示您提供相关的密码。将不显示此密码。

`-v`

可选。生成包含监视器状态的其他相关信息的详细输出，其中包括监视器是已启动还是已停止、受监视的目录资源路径以及触发条件。

`-ox (xml_filename)`

可选。必须将此参数与 `-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` 命令和 `-ix` 参数来使用该文件。

V 9.0.5 `-ox` 参数不能与 `-od` 参数组合使用。

V 9.0.5 -od (directory_name)

可选。将多个资源监视器定义导出到指定的目录。每个资源监视器定义均保存到一个独立的 XML 文件，此文件的名称采用 *agent name.monitor name.xml* 格式。您必须为 XML 文件指定有效的目标目录，否则会显示一条错误消息。此参数不能与 **-ox** 参数组合使用。

-你说什么或 -h

可选。显示命令语法。

示例：列出资源监视器

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
```

示例：将一个资源监视器导出到 XML 文件

在此示例中，通过使用 **-ox** 参数指定 XML 文件名，将 AGENT1 上的一个资源监视器 MONITOR1 导出到 XML 文件 filename1.xml：

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -ox filename1.xml
```

示例：将一个资源监视器导出到指定的目录

V 9.0.5

在此示例中，将 AGENT1 上的一个资源监视器 MONITOR1 导出到由 **-od** 参数指定的目录。除了 XML 文件名格式的差异外，此示例与使用 **-ox** 参数类似。

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -od /usr/mft/resmonbackup
```

示例：将一批资源监视器导出到指定目录中的 XML 文件

V 9.0.5

在以下所有示例中，将资源监视器导出到由 **-od** 参数指定的目录。每个资源监视器定义都将保存到具有 *agent name.monitor name.xml* 格式的名称的单独 XML 文件中。

在此示例中，所有资源监视器都导出到指定的目录：

```
fteListMonitors -od /usr/mft/resmonbackup
```

在此示例中，AGENT1 上的所有资源监视器都导出到指定的目录：

```
fteListMonitors -ma AGENT1 -od /usr/mft/resmonbackup
```

您可以通过在指定用于匹配代理名称和/或监视器名称的模式时使用星号字符 (*)，以利用通配符匹配来定义要导出的资源监视器。

在此示例中，AGENT1 上名称与模式 MON* 匹配的所有资源监视器都导出到指定的目录：

```
fteListMonitors -ma AGENT1 -mn MON* -od /usr/mft/resmonbackup
```

在此示例中，名称与模式 AGEN* 匹配的代理上的所有资源监视器都导出到指定的目录：

```
fteListMonitors -ma AGEN* -od /usr/mft/resmonbackup
```

在此示例中，名称与模式 AGENT* 匹配的代理上名称与模式 MON* 匹配的所有资源监视器都导出到指定的目录：

```
fteListMonitors -ma AGENT* -mn MON* -od /usr/mft/resmonbackup
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关任务

[第 141 页的『监视 MFT 资源』](#)

您可以监视 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 Managed File Transfer 插件中的 **监视器** 视图来创建资源监视器。

[第 160 页的『备份和复原 MFT 资源监视器』](#)

对于您希望将来可用的资源监视器，可以通过将其定义导出到 XML 文件来备份这些监视器，以后可通过导入此 XML 文件来从备份创建新的资源监视器。

相关参考

[第 345 页的『fteCreateMonitor: 创建 MFT 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 Managed File Transfer 监视资源（例如，某个目录的内容），以便在满足触发条件时启动指定的任务（如文件传输）。

[第 389 页的『fteDeleteMonitor: 删除 MFT 资源监视器』](#)

使用 **fteDeleteMonitor** 命令可通过命令行停止和删除现有 Managed File Transfer 资源监视器。针对资源监视代理发出此命令。

fteListScheduledTransfers

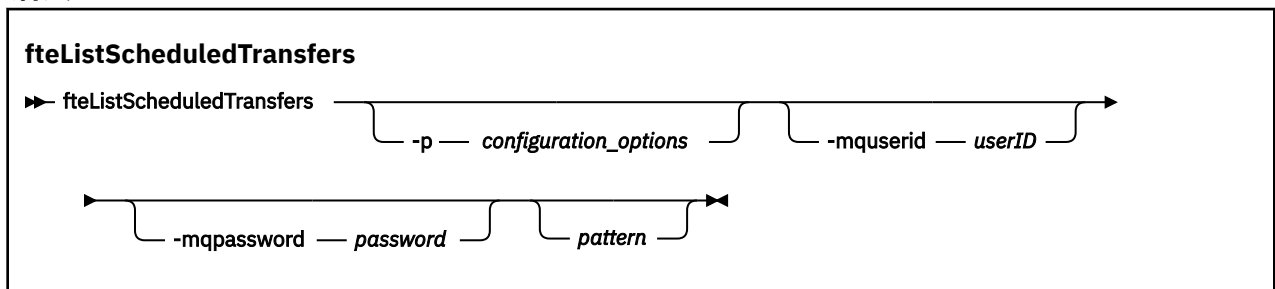
使用 **fteListScheduledTransfers** 命令可列出先前使用命令行或 IBM MQ Explorer 创建的所有 Managed File Transfer 传输。

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 49 页的『多平台上的 MFT 配置选项』](#)，以了解更多信息。

语法



参数

-p (configuration_options)

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。

模式 (pattern)

可选。用于过滤 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。

-你说什么 或 -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 135 页的『创建调度的文件传输』](#)

您可以从 IBM MQ Explorer 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 391 页的『fteDeleteScheduledTransfer』](#)

fteListTemplates

fteListTemplates 命令可用于列出协调队列管理器上的可用 Managed File Transfer 传输模板。

用途

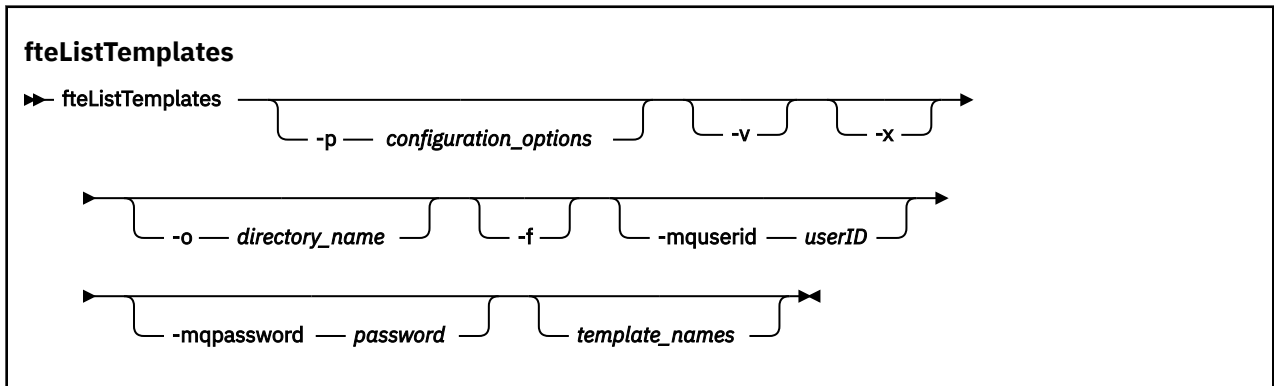
该命令列出所有模板名称或过滤的模板名称选择。列表的输出格式可以是以下任意一种：

- 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 453 页的『MFT 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，作为与零个或多个字符匹配的通配符。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。

-你说什么 或 -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。

fteMigrate 代理: 将 FTE 7.0 代理迁移到 MQ 7.5 或更高版本

如果要将现有代理及其关联配置从 IBM WebSphere MQ File Transfer Edition 7.0 迁移到 IBM WebSphere MQ 7.5 或更高版本，请使用 **fteMigrateAgent** 命令进行迁移。此命令可用于迁移标准代理、Connect:Direct 代理或协议网桥代理。该命令还可用于在单个请求中迁移多个代理。

V 9.0.0 从 IBM MQ 9.0 开始，Managed File Transfer 不支持 Web 代理。如果尝试使用 **fteMigrateAgent** 命令将 Web 代理从较低版本迁移到 IBM MQ 9.0，那么会显示错误消息来说明不支持迁移 Web 代理。

注：如果是从 IBM WebSphere MQ File Transfer Edition 7.0 或更高版本进行迁移，并且要继续使用 FTE_CONFIG 环境变量，那么可以执行此操作而不更改 FTE_CONFIG 值。可以执行标准迁移，但不得设置 BFG_DATA，并且必须按照 IBM WebSphere MQ 7.0 中的用法设置 FTE_CONFIG。

要点: **ULW** 在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理员（和 mqm 组成员）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。

z/OS 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）。
- **V 9.0.0.1** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V 9.0.0.1** 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。迁移完成后，请再次使用 **fteModifyAgent** 命令将新代理程序配置为 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。

必须先使用 **fteStopAgent** 命令停止要迁移的代理，然后才能运行 **fteMigrateAgent** 命令。

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：

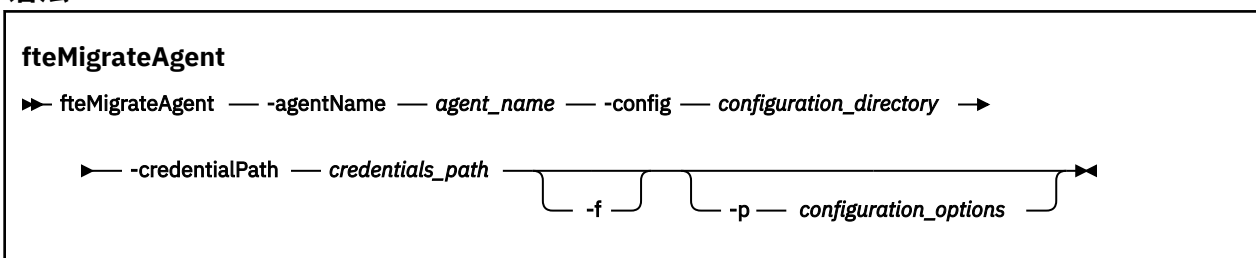
fteMigrateAgent 命令为每个代理迁移的文件的名称	信息
wmqfte.properties	wmqfte.properties 文件在 IBM WebSphere MQ 7.5 或更高版本中重命名为 installation.properties。
command.properties	
coordination.properties	
coordination_queue_manager.mqsc	
agent_name_create.mqsc	
agent_name_delete.mqsc	
exits 目录	该命令会复制 exits 目录中的所有文件。
仅适用于标准代理：	
UserSandboxes.xml	
仅适用于 Connect:Direct 网桥代理：	
ConnectDirectCredentials.xml	
ConnectDirectNodeProperties.xml	
ConnectDirectProcessDefinitions.xml	
仅适用于协议网桥代理：	
ProtocolBridgeCredentials.xml	

表 37: `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
ProtocolBridgeProperties.xml	此文件仅在 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本上存在。

fteMigrateAgent 命令用于迁移安装、协调和命令队列管理器的文件，并将这些文件复制到 IBM WebSphere MQ 7.5 或更高版本，前提是 IBM WebSphere MQ 7.5 或更高版本中尚不存在这些文件。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



参数

-agentName *agent_name*

必需。要迁移到 IBM WebSphere MQ 7.5 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)，其中可以有需要更新的现有成员，也可以没有现有成员，而需要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 Managed File Transfer 上的属性文件与 IBM WebSphere MQ 7.5 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。要了解更多信息，请参阅第 49 页的『多平台上的 MFT 配置选项』。

-你说什么 或 -h

可选。显示命令语法。

示例

在此示例中，AGENT3 及其在 /var/ibm/WMQFTE/config 中的配置迁移到 IBM WebSphere MQ 7.5 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 7.5 或更高版本。Windows 文件路径括在双引号 (") 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" -p "configurationOption" -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 282 页的『MFT 的返回码』。

fteMigrateConfigurationOptions: 将 FTE 7.0 配置迁移到 MQ 7.5 或更高版本

fteMigrateConfigurationOptions 命令从 IBM WebSphere MQ File Transfer Edition 7.0 迁移一组配置选项并将它们复制到 IBM WebSphere MQ 7.5 或更高版本，前提是目标版本中尚不存在相应文件。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：如果是从 IBM WebSphere MQ File Transfer Edition 7.0 进行迁移，并且要继续使用 FTE_CONFIG 环境变量，那么可以执行此操作而不更改 FTE_CONFIG 值。可以执行标准迁移，但不得设置 BFG_DATA，并且必须按照 IBM WebSphere MQ File Transfer Edition 7.0 中的用法设置 FTE_CONFIG。

要点: **ULW** 在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理员（和 mqm 组成员）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。

z/OS 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）。
- **V9.0.0.1** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V9.0.0.1** 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。

此参数用于将 `agent.properties`、`coordination.properties` 和 `command.properties` 文件中存在的 SSL/TLS 密钥库和信任库属性的密码属性从 IBM WebSphere MQ 7.5 之前的产品版本迁移到 7.5 或更高版本。

在 IBM WebSphere MQ 7.5 之前，密码属性 (例如 `coordinationSslTrustStorePassword`) 曾经存在于 `coordination.properties` 文件中，但已移至 IBM WebSphere MQ 7.5 中的 `MQMFTCredentials.xml` 文件。要将密码属性从 `coordination.properties` 文件移至 `MQMFTCredentials.xml` 文件，请使用 `-credentialsPath`。使用该选项时，`fteMigrateConfigurationOptions` 命令会在 `coordination.properties` 文件中查找 “`coordinationSslTrustStorePassword`”，如果存在此内容，那么会将该属性迁移到 `MQMFTCredentials.xml` 文件。

z/OS 对于 z/OS 平台，这可以是预先存在的扩展分区数据集 (PDSE)，其中可以有需要更新的现有成员，也可以没有现有成员，而需要针对这些凭证包含一个新成员。

注: 如果使用 PDSE，那么它必须是可变的。

-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 12 页的『MFT 的迁移注意事项』

如果计划迁移 Managed File Transfer，请查看以下信息。

第 403 页的『fteMigrate 代理: 将 FTE 7.0 代理迁移到 MQ 7.5 或更高版本』

如果要将现有代理及其关联配置从 IBM WebSphere MQ File Transfer Edition 7.0 迁移到 IBM WebSphere MQ 7.5 或更高版本，请使用 `fteMigrateAgent` 命令进行迁移。此命令可用于迁移标准代理、Connect:Direct 代理或协议网桥代理。该命令还可用于在单个请求中迁移多个代理。

第 407 页的『fteMigrate 记录器: 将 FTE 7.0 数据库记录器迁移到 MQ 7.5 或更高版本』

如果想要将现有独立数据库记录器的配置从 IBM WebSphere MQ File Transfer Edition 7.0.1 或更高版本迁移到 IBM WebSphere MQ 7.5 或更高版本，请使用 `fteMigrateLogger` 命令。

fteMigrate 记录器: 将 FTE 7.0 数据库记录器迁移到 MQ 7.5 或更高版本

如果想要将现有独立数据库记录器的配置从 IBM WebSphere MQ File Transfer Edition 7.0.1 或更高版本迁移到 IBM WebSphere MQ 7.5 或更高版本，请使用 `fteMigrateLogger` 命令。

您无法使用该命令来迁移 Java EE 数据库记录器: 请改为使用迁移 Java EE 数据库记录器中的信息。

注: 如果是从 IBM WebSphere MQ 7.0 或更高版本进行迁移，并且要继续使用 `FTE_CONFIG` 环境变量，那么可以执行此操作而不更改 `FTE_CONFIG` 值。可以执行标准迁移，但不得设置 `BFG_DATA`，并且必须按照 IBM WebSphere MQ 7.0 中的用法设置 `FTE_CONFIG`。

要点: **ULW** 在 IBM MQ for UNIX, Linux, and Windows 上, 只有 IBM MQ 管理员 (和 mqm 组成员) 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。

z/OS 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)。
- **V9.0.0.1** 是 BFG_GROUP_NAME 环境变量中指定的组 (如果指定了一个组) 的成员。
- **V9.0.0.1** 在运行命令时, 在 BFG_GROUP_NAME 环境变量中未设置任何值。

如果您已将独立的数据库记录器配置为作为 Windows 服务运行, 那么无法使用 **fteMigrateLogger** 命令迁移该记录器的配置。如果对配置为作为 Windows 服务运行的记录器运行 **fteMigrateLogger** 命令, 那么此命令会生成错误, 并且无法继续。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。

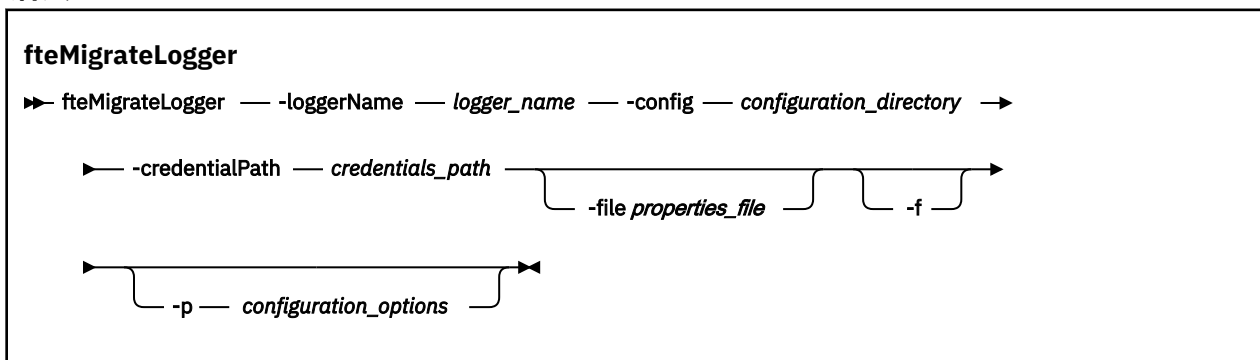
在运行 **fteMigrateLogger** 命令之前, 请停止要在 IBM WebSphere MQ File Transfer Edition 7.0 上迁移其配置 的数据库记录器。

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:

fteMigrateLogger 命令迁移的文件的名称	信息
wmqfte.properties	wmqfte.properties 文件基于 IBM WebSphere MQ 7.5 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 7.5 或更高版本中创建 logger.properties 文件。

fteMigrateLogger 命令用于迁移安装、协调和命令队列管理器的文件, 并将这些文件复制到 IBM WebSphere MQ 7.5 或更高版本, 前提是目标版本中尚不存在这些文件。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



参数

-loggerName logger_name

必需。您希望为 IBM WebSphere MQ 7.5 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 IBM WebSphere MQ 7.5 为新名称）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)，其中可以有需要更新的现有成员，也可以没有现有成员，而需要针对这些凭证包含一个新成员。

注: 如果使用 PDSE，那么它必须是可变的。

-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，如果在 IBM WebSphere MQ File Transfer Edition 上的数据库记录器属性文件与 IBM WebSphere MQ 7.5 或更高版本上的属性文件之间存在不匹配，那么指定 **-f** 参数意味着将忽略该不匹配情况。

-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 49 页的『多平台上的 MFT 配置选项』](#)。

-你说什么 或 -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配置将迁移到 IBM WebSphere MQ 7.5，并命名为 FTELOGGER1:

```
fteMigrateLogger -loggerName FTELOGGER1 -config /var/ibm/WMQFTE/config
-credentialPath /home/user1/FTELOGGER1
```

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 282 页的『MFT 的返回码』](#)。

运行 fteMigrateLogger 命令之后

要验证迁移，请在成功运行 **fteMigrateLogger** 命令之后，在 IBM WebSphere MQ 7.5 或更高版本上使用 [第 444 页的『fteStartLogger』](#) 命令启动已迁移其配置的数据库记录器。

相关参考

[第 12 页的『MFT 的迁移注意事项』](#)

如果计划迁移 Managed File Transfer，请查看以下信息。

[第 403 页的『fteMigrate 代理: 将 FTE 7.0 代理迁移到 MQ 7.5 或更高版本』](#)

如果要将现有代理及其关联配置从 IBM WebSphere MQ File Transfer Edition 7.0 迁移到 IBM WebSphere MQ 7.5 或更高版本，请使用 **fteMigrateAgent** 命令进行迁移。此命令可用于迁移标准代理、Connect:Direct 代理或协议网桥代理。该命令还可用于在单个请求中迁移多个代理。

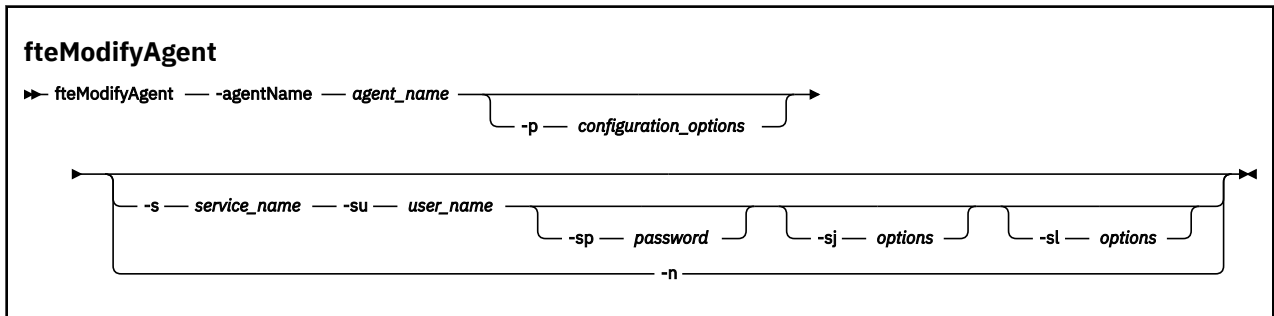
第 406 页的『[fteMigrateConfigurationOptions: 将 FTE 7.0 配置迁移到 MQ 7.5 或更高版本](#)』

fteMigrateConfigurationOptions 命令从 IBM WebSphere MQ File Transfer Edition 7.0 迁移一组配置选项并将它们复制到 IBM WebSphere MQ 7.5 或更高版本，前提是目标版本中尚不存在相应文件。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。

Windows **fteModify** 代理 (将 MFT 代理作为 Windows 服务运行)

fteModifyAgent 命令将修改现有代理，使其能够作为 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。

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。指示该代理将作为 Windows 服务运行。如果不指定 *service_name*，那么服务将命名为 `mqmftAgentAGENTQMGR`，其中 *AGENT* 是代理名称；而 *QMGR* 是代理队列管理器名称。

服务的显示名称（显示在 Windows 服务窗口的名称列中）始终为 **Managed File Transfer Agent AGENT@QMGR**。

注：如果可再发行代理将作为 Windows 服务运行，那么 **BFG_DATA** 环境变量需要在系统环境中进行设置，以便服务正常工作。

-su user_name

可选。如果该代理将作为 Windows 服务运行，那么该参数指定应作为该服务运行身份的帐户的名称。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 **Log on as a service** 权限。有关如何授予此权限的信息，请参阅 [有关将 MFT 代理程序或记录器作为 Windows 服务运行的指南](#)。

当指定 **-s** 时，该参数为必需参数。

-sp password

可选。该参数仅在指定 **-s** 时才有效。

-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。

该参数仅在指定 **-s** 时才有效。

有关 **fteModifyAgent** 命令处理 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 选项的指导信息](#)。

-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 服务存在问题，那么此选项很有用。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。

-n

可选。指示代理将作为正常进程运行。这与 **-s** 选项互斥。如果未指定 **-s** 或 **-n** 选项，那么会将代理程序配置为正常的 Windows 进程。

-你说什么 或 -h

可选。显示命令语法。

示例

在此示例中，AGENT1 将修改为作为 Windows 服务运行：

```
fteModifyAgent -agentName AGENT1 -s -su fteuser -sp ftepassword
```

在此示例中，AGENT1 将修改为移除 Windows 服务：

```
fteModifyAgent -agentName AGENT1
```

必须在运行 fteModifyAgent 命令前，使用 [fteStopAgent](#) 命令停止要修改的代理。

返回码

0

命令成功完成。

1

命令结束但未成功。

相关参考

[第 323 页的『fteCreateAgent \(创建 MFT 代理\)』](#)

fteCreateAgent 命令创建 Managed File Transfer Agent 及其关联的配置。

[第 411 页的『fteModify 记录器 \(将 MFT 记录器作为 Windows 服务运行\)』](#)

使用 **fteModifyLogger** 命令来修改 Managed File Transfer 记录器，以便可以将其作为 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。

相关信息

将 MFT 代理作为 [Windows 服务启动](#)

[用于将 MFT 代理或记录器作为 Windows 服务运行的指南](#)

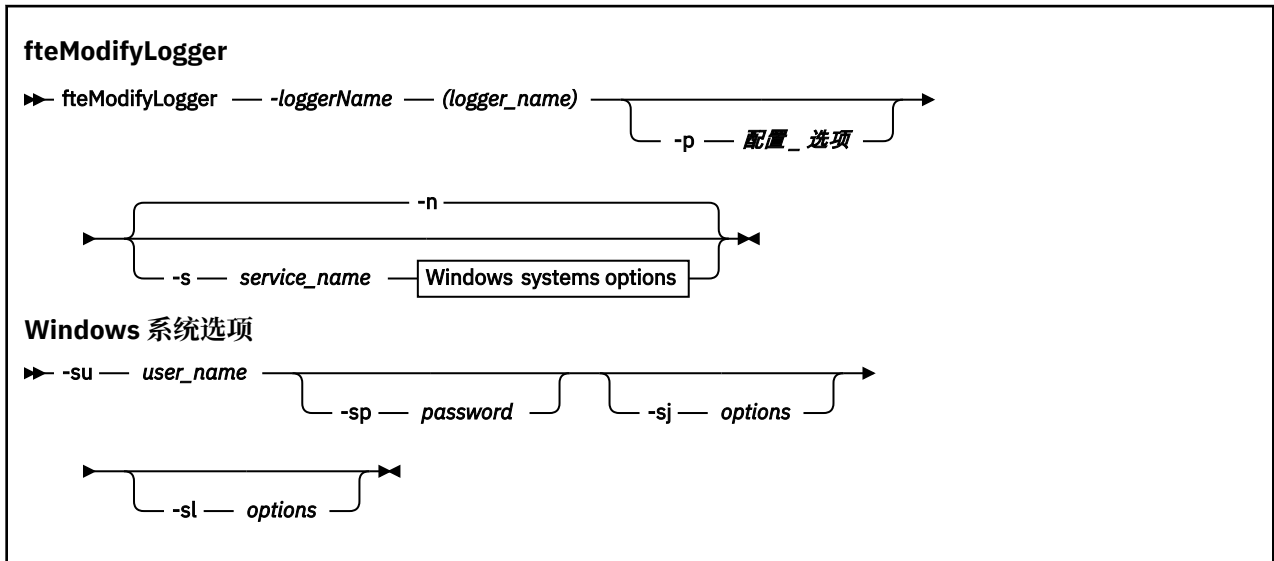
Windows fteModify 记录器 (将 MFT 记录器作为 Windows 服务运行)

使用 **fteModifyLogger** 命令来修改 Managed File Transfer 记录器，以便可以将其作为 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。

用途

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 “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。

语法



参数

-loggerName (logger_name)

必需。要修改的 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。指示该记录器将作为 Windows 服务运行。如果未指定 *service_name*，那么服务将命名为 *mqftLoggerLOGGERQMGR*，其中 *LOGGER* 是记录器名称，而 *QMGR* 是记录器队列管理器名称。

服务的显示名称（显示在 Windows 服务窗口的 **名称** 列中）始终为 **Managed File Transfer Logger *LOGGER@QMGR***。

-su user_name

当指定 **-s** 时必需。指定应作为 Windows 服务运行身份的帐户的名称。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。有关如何授予此许可权的信息，请参阅 [有关将 MFT 代理程序或记录器作为 Windows 服务运行的指南](#)。

-sp password

可选。仅当指定 **-s** 时才有效。**-su** 参数设置的用户帐户的密码。

如果未指定该参数，那么当指定 **-s** 参数时，系统将警告您必须首先使用 Windows“服务”工具设置密码，然后服务才能成功启动。

-sj options

可选。仅当指定 **-s** 时才有效。当记录器作为 Windows 服务启动时，此参数以 **-D** 或 **-X** 格式定义将传递至 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。

有关 **fteModifyLogger** 命令处理 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 选项的指导信息](#)。

-sl 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 服务存在问题，那么此选项很有用。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

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 **-s** 和 **-n** 选项均未指定，那么记录器将配置为正常 Windows 进程。

-你说什么 或 -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。该命令显示如何将记录器更改为作为 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。

相关参考

[第 444 页的『fteStartLogger』](#)

fteStartLogger 命令启动 Managed File Transfer 日志记录应用程序。

[第 447 页的『fteStopLogger』](#)

fteStopLogger 命令停止 Managed File Transfer 记录器。

相关信息

[将 MFT 代理作为 Windows 服务启动](#)

[用于将 MFT 代理或记录器作为 Windows 服务运行的指南](#)

fteObfuscate：对敏感数据进行加密

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。

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。

-你说什么 或 -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。

相关参考

第 36 页的『MFT 和 IBM MQ 连接认证』

连接认证允许将队列管理器配置为使用提供的用户标识和密码对应用程序进行认证。如果关联的队列管理器已启用安全性，并且需要凭证详细信息(用户标识和密码)，那么必须先启用连接认证功能，才能成功连接到队列管理器。可以在兼容性方式或 MQCSP 认证方式下运行连接认证。

第 691 页的『MFT 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。

ftePingAgent: 检查 MFT 代理是否处于活动状态

ftePingAgent 命令 ping Managed File Transfer 代理以确定代理是否可访问，如果可访问，则确定它是否能够响应简单查询。

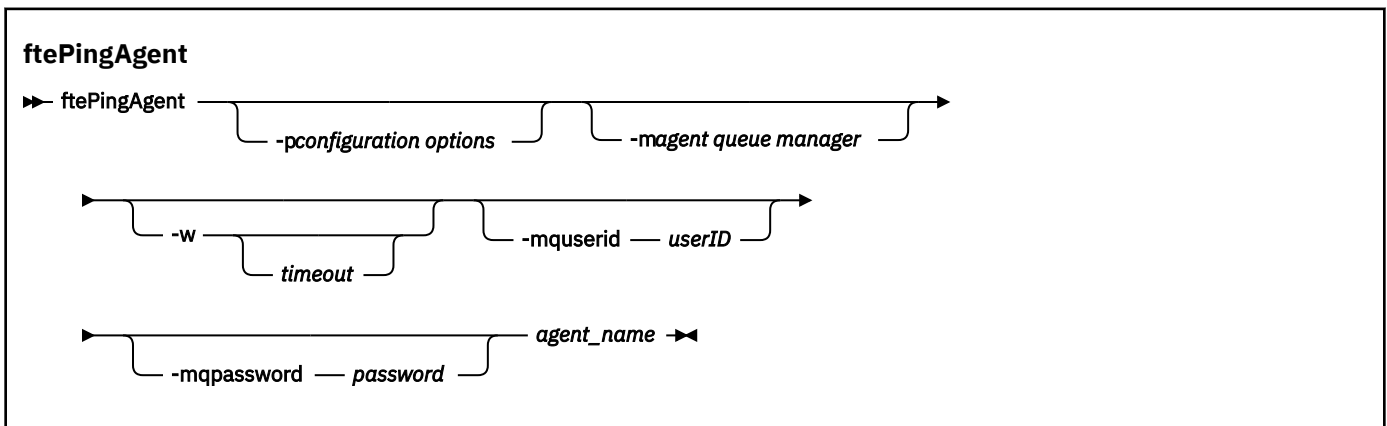
用途

使用 **ftePingAgent** 命令检查 Managed File Transfer 代理以确定代理是否可访问，如果可访问，则确定它是否能够响应类似 are you there? 这样的简单查询。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



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 Managed File Transfer 代理的名称。

-你说什么 或 -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 395 页的『[fteListAgents: 列出协调队列管理器的 MFT 代理](#)』

使用 **fteListAgents** 命令可列出向特定协调队列管理器注册的所有 Managed File Transfer 代理程序。

第 433 页的『[fteShowAgentDetails](#)』

使用 **fteShowAgentDetails** 命令可显示特定 Managed File Transfer Agent 的详细信息。这些是由代理的 Managed File Transfer 协调队列管理器存储的详细信息。

第 262 页的『[在您认为文件传输卡住时应该怎么办](#)』

在高负荷的系统上或者在源和目标代理之间发生网络问题时，传输可能会偶尔显示为卡住，进入排队或恢复状态。导致此情况的因素有很多。

fteRAS: 收集 MFT 故障诊断信息

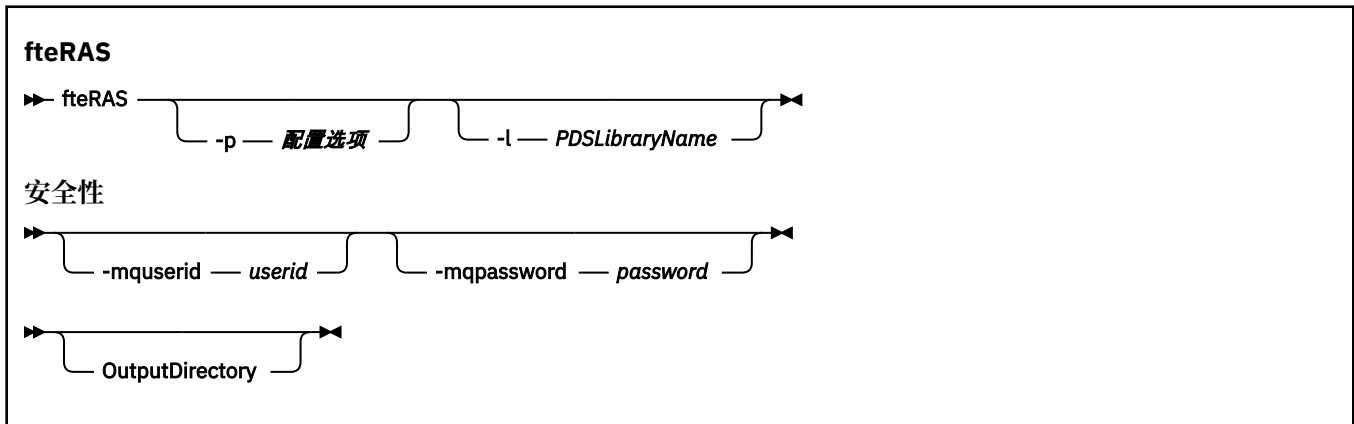
fteRAS 命令收集 Managed File Transfer 的故障诊断信息 (MustGather 数据)。**fteRAS** 收集的信息特定于运行程序的系统上的 Managed File Transfer 安装。

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。

z/OS -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

Linux **UNIX** 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
```

Windows 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](#)。

IBM i 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, 如以下示例中所示:

```
/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 226 页的『[对 MFT 进行故障诊断](#)』

使用以下参考信息可帮助您诊断 Managed File Transfer 中的错误:

V 9.0.0 fteSetAgentLogLevel: 设置 MFT 协议网桥代理日志级别

使用 **fteSetAgentLogLevel** 命令对 Managed File Transfer 协议网桥代理与 FTP/SFTP/FTPS 文件服务器之间的文件传输启用或禁用诊断信息日志记录。

用途

使用此命令可对有关 Managed File Transfer 协议网桥代理和 FTP/SFTP/FTPS 文件服务器之间的文件传输的诊断信息收集启用或禁用日志记录。Managed File Transfer 协议网桥代理日志记录更轻量级、与跟踪分隔但可与其并行运行。

由协议网桥代理发送到 FTP/SFTP/FTPS 文件服务器的命令以及接收到的响应会写入到日志文件中, 以对文件传输期间可能发生的问题进行简单诊断。

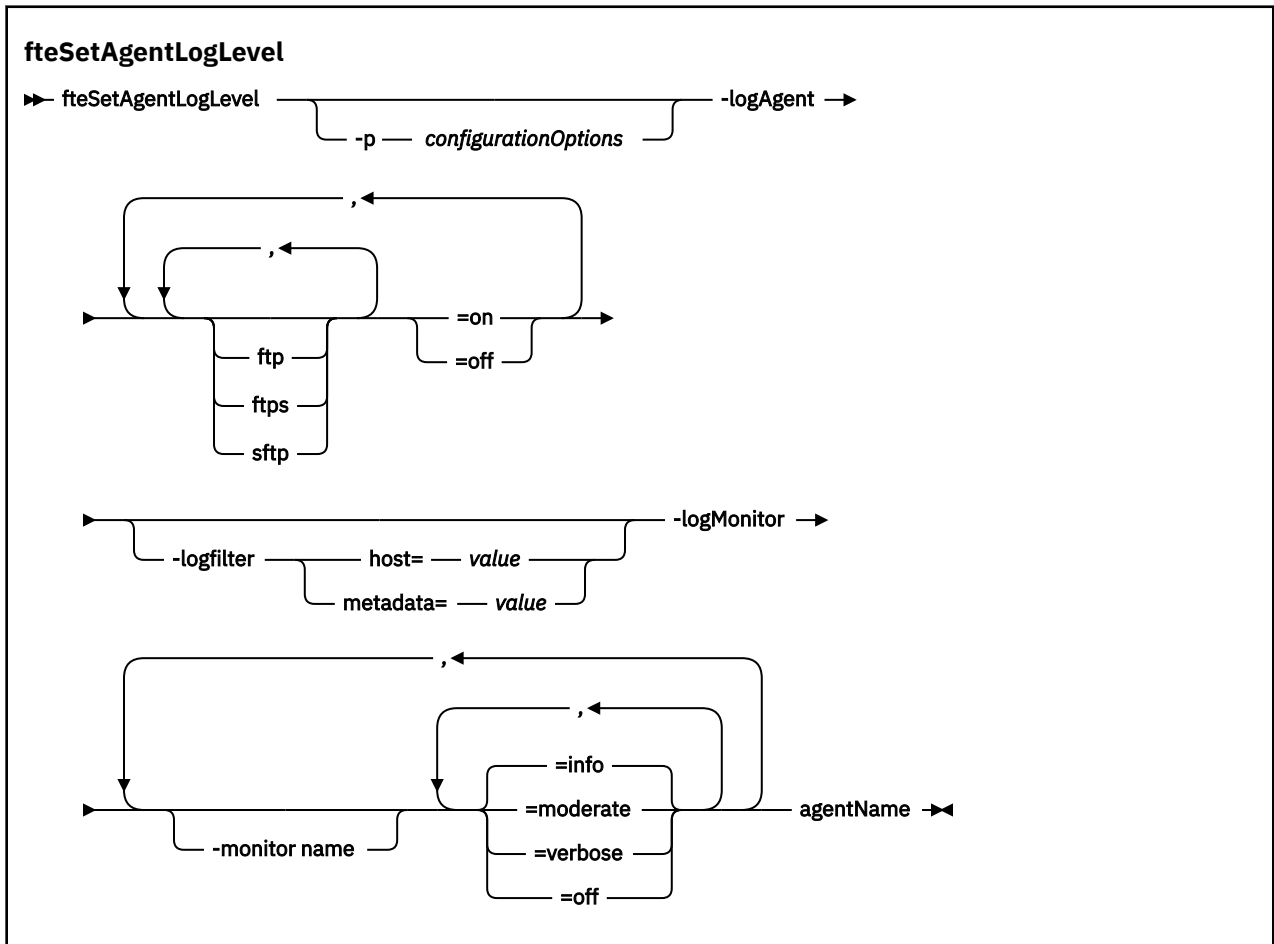
可以通过两种方式启用、禁用和设置所需的日志记录级别:

- 使用 **fteSetAgentLogLevel** 命令在协议网桥代理运行的同时启用或禁用日志记录。无需重新启动协议网桥代理, 即可开始日志记录。
- 使用 **agent.properties** 文件在代理启动时启用或禁用日志记录。如果使用此方法启用或禁用日志记录, 并且代理正在运行, 那么必须重新启动代理以开始日志记录。有关更多信息, 请参阅 [agent.properties](#) 文件。

协议网桥代理和文件服务器之间的交互被视为事件, 并且每个事件都写入到 **agentevent?.log** 文件。"?" 在日志名称中表示数字值。在其名称中包含 0 (零) 的文件会包含捕获的最新消息集。该日志文件存储在 **MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/** 文件夹中。

例如, 可以在 [agent.properties](#) 文件中设置更多日志记录属性, 以限制日志文件大小和保留的日志文件数。

语法



参数

`-logAgent component=operation`

必需。

V 9.0.3

要点: 只能选择 `logAgent` 和 `logMonitor` 中的一个。如果您尝试同时选择这两个选项，那么将收到错误消息。

为其启用或禁用日志记录的协议网桥代理的名称。可以为三个组件启用或禁用协议网桥代理日志记录：`ftp`、`ftps` 和 `sftp` 服务器。指定三种可能的服务器类型之一并添加操作值，以针对其连接到的协议网桥代理关闭或打开日志记录。

V 9.0.3

如果组件以加号 (+) 开头，那么将向目前正在记录的所有现有日志组件添加加号后面的组件列表。

有效日志级别操作选项如下：

关

对协议网桥代理禁用所有日志记录。这是缺省选项。

```
fteSetAgentLogLevel -logAgent =off PBA1
```

要对协议网桥代理连接到的指定组件禁用日志记录，请使用以下命令：

```
fteSetAgentLogLevel -logAgent ftp=off PBA1
```

```
fteSetAgentLogLevel -logAgent ftps=off PBA1
```

```
fteSetAgentLogLevel -logAgent sftp=off PBA1
```

开启

要对协议网桥代理连接到的全部三个可能的文件服务器组件启用日志记录，请使用以下命令：

```
fteSetAgentLogLevel -logAgent =on PBA1
```

要对协议网桥代理连接到的指定组件启用日志记录，请使用以下命令：

```
fteSetAgentLogLevel -logAgent ftp=on PBA1
```

```
fteSetAgentLogLevel -logAgent ftps=on PBA1
```

```
fteSetAgentLogLevel -logAgent sftp=on PBA1
```

有关其他配置选项，请参阅第 422 页的『示例 1』和第 422 页的『示例 2』。

-agentName

必需。为其启用或禁用日志记录的协议网桥代理的名称。

-logFilter filter=value

可选。用于将日志记录限于指定的过滤器条件。必须指定 **host** 或用户指定的文件传输 **metadata** 的值。

主机

指定要按其过滤的三个值之一：主机名、逗号分隔的主机名列表或 IP 地址。要记录 FTP 服务器 `ftpprod.ibm.com` 的命令和来自它的响应，请使用以下命令：

```
fteSetAgentLogLevel -logAgent ftp=on -logFilter host=ftpprod.ibm.com PBA1
```

要记录来自 IP 地址以 `9.182.*` 开头的所有 SFTP 服务器的命令和响应，请使用以下命令：

```
fteSetAgentLogLevel -logAgent sftp=on -logFilter host=9.182.* PBA1
```

元数据

以 `key=value` 格式按用户在传输创建期间的定义指定任何文本。例如，`metadata="BANK=WORLD BANK"`。要对连接到协议网桥代理 PBA1 的任何 FTP 服务器启用日志记录，并将文件传输过滤为仅当其包含 **metadata** 作为键且“BANK=WORLD BANK”作为值时才将这些文件传输包括在内，请使用以下命令：

```
fteSetAgentLogLevel -logAgent ftp=on metadata="BANK=WORLD BANK" PBA1
```

注：为按元数据过滤，必须在文件传输过程中在 `-md` 参数下指定进行过滤所依据的值。有关更多信息，请参阅第 364 页的『fteCreateTransfer：启动新的文件传输』。

V 9.0.3 -logMonitor monitor name=log level

必需。

要点：只能选择 **logAgent** 和 **logMonitor** 中的一个。如果您尝试同时选择这两个选项，那么将收到错误消息。

要启用或禁用日志记录的资源监视器的逗号分隔列表，其中：

monitor name

可选。要启用或禁用日志记录的资源监视器名称或名称列表。如果未指定名称，那么将对代理的所有资源监视器启用日志记录。

log level

必需。值为下列其中一项：

参考

开启参考级别日志记录。这是缺省值。

要对代理 AGENT1 的监视器 MON1 启用 info 级别的日志记录：

```
fteSetAgentLogLevel -logMonitor MON1=info AGENT1
```

moderate

开启适量级别日志记录。

要对代理 AGENT1 的监视器 MON1 和 MON2 启用 moderate 级别日志记录：

```
fteSetAgentLogLevel -logMonitor MON1, MON2=moderate AGENT1
```

详细

开启详细级别日志记录。

要对代理 AGENT1 的所有监视器启用 verbose 级别日志记录：

```
fteSetAgentLogLevel -logMonitor =verbose AGENT1
```

关

关闭日志记录。

要对代理 AGENT1 的监视器 MON1 和 MON2 关闭日志记录：

```
fteSetAgentLogLevel -logMonitor MON1, MON2=off AGENT1
```

要对监视器 MON1 关闭日志记录并对代理 AGENT1 的监视器 MON2 启用 info 级别日志记录：

```
fteSetAgentLogLevel -logMonitor MON1=off, MON2=info AGENT1
```

要对代理 AGENT1 的所有监视器关闭日志记录：

```
fteSetAgentLogLevel -logMonitor =off AGENT1
```

V 9.0.3 -p configurationOptions

可选。确定用于设置代理日志级别的配置选项集。将配置选项集的名称用作 -p 参数的值。

按照约定，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。

-你说什么或 -h

可选。显示命令语法。

V 9.0.3 如果在命令中重复使用同一个资源监视器名称，那么会将最后一次出现的组件视为有效组件。

例如：

```
fteSetAgentLogLevel -logMonitor MON1=info, MON2=off, MON1=off AGENT1
```

将对资源监视器 MON1 关闭日志记录。

V 9.0.3



注意：如果在命令中指定了不存在的资源监视器名称，那么控制台上不会显示错误。

示例 1

在此示例中，通过使用命令定界组在一个命令中指定多个组件。对于 FTP 和 SFTP 服务器启用了日志记录，但是对于 PBA1 协议网桥代理连接到的 FTPS 服务器禁用了日志记录。

```
fteSetAgentLogLevel -logAgent ftp=on,ftps=off,sftp=on PBA1
```

您还可以使用逗号分隔组件以实现相同的结果，例如：

```
fteSetAgentLogLevel -logAgent ftp,sftp=on,ftps=off PBA1
```

示例 2

在此示例中，在命令中重复了同一组件。*component=operation* 对的最后一个实例被视为有效。此示例对 PBA1 协议网桥代理连接到的 FTP 服务器禁用日志记录。

```
fteSetAgentLogLevel -logAgent ftp=on,ftp=off PBA1
```

上一个示例具有与此示例相同的效果：

```
fteSetAgentLogLevel -logAgent ftp=off PBA1
```

返回码

0

命令成功完成。

1

命令结束但未成功。

相关概念

[第 177 页的『协议网桥』](#)

通过协议网桥，Managed File Transfer (MFT) 网络可以访问您的 MFT 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 460 页的『MFT agent.properties 文件』](#)

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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 364 页的『fteCreateTransfer: 启动新的文件传输』](#)

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。

fteSetAgentTraceLevel

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：

Multi 在多平台上使用 IBM WebSphere MQ 7.5 或更高版本时，只有运行代理进程的用户才能运行 **fteSetAgentTraceLevel** 命令。

V 9.0.2 **z/OS** 对于 z/OS，针对 Continuous Delivery，从 IBM MQ 9.0.2 开始，**fteSetAgentTraceLevel** 命令可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 9.0.0.1 这也适用于来自 IBM MQ 9.0.0 Fix Pack 1 的 Long Term Support 发行版。

有关更多信息，请参阅第 460 页的『MFT agent.properties 文件』中的 **adminGroup** 属性。

在 IBM WebSphere MQ 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，仅在您的 IBM 服务代表要求时才启用跟踪。



注意：

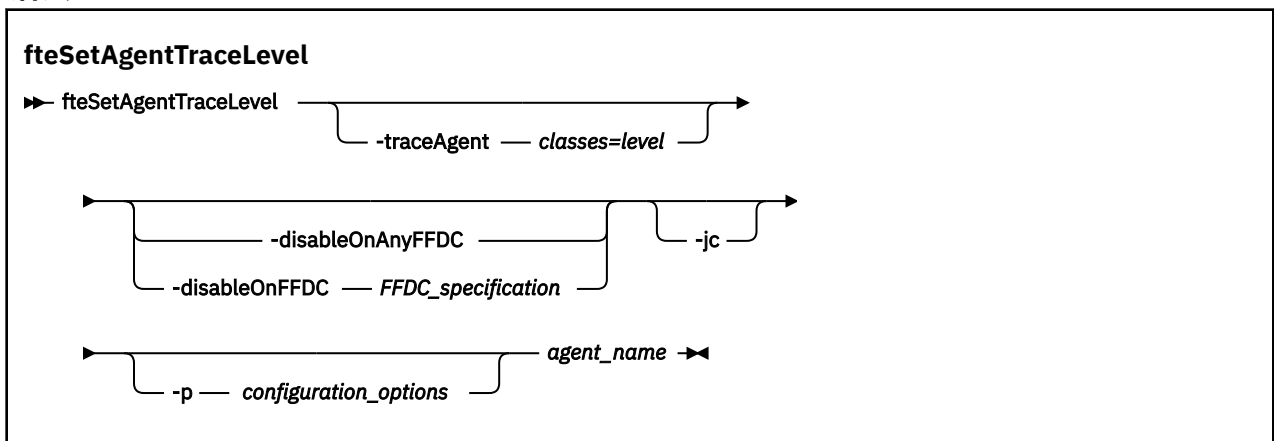
1. 必须在运行代理程序的系统上运行此命令。
2. 跟踪和日志记录不会在代理程序重新启动时持久存储。

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 460 页的『MFT 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类。

V 9.0.5 从 IBM MQ 9.0.5 开始，您可以指定逗号分隔的类规范列表。此选项使您能够在不同的级别设置不同类的跟踪。例如：

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.agent=all:com.ibm.wmqfte.cmdhandler=moderate
AGENT1
```

您仍可以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。请使用以下格式：

```
classes=level
```

例如：

```
com.ibm.wmqfte=all
```

您可以使用 Managed File Transfer 程序包名称来代替 *classes*，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 Managed File Transfer Agent 的名称。

-你说什么 或 -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，动态修改 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 WebSphere MQ 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，仅在您的 IBM 服务代表要求时才启用跟踪。



注意：

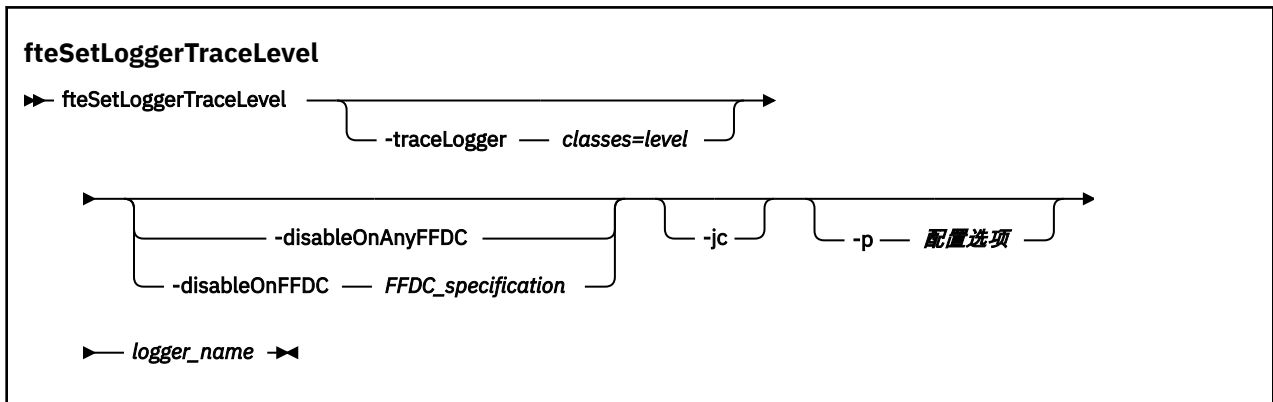
1. 必须在运行代理程序的系统上运行此命令。
2. 跟踪和日志记录不会在代理程序重新启动时持久存储。

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。要了解更多信息，请参阅第 98 页的『MFT 记录器配置属性』。

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。

V 9.0.5 从 IBM MQ 9.0.5 开始，您可以指定冒号分隔的类规范列表。此选项使您能够在不同的级别设置不同类的跟踪。例如：

```
fteSetLoggerTraceLevel -traceLogger com.ibm.wmqfte.logger=all:com.ibm.wmqfte.cmdhandler=moderate  
LOGGER1
```

您仍可以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。请使用以下格式：

```
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`。

您只能指定 **-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 configuration_options

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 Managed File Transfer Logger 的名称。

-你说什么 或 -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（以 `com.ibm.outer.inner` 开头的类除外）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。

fteSetProductId: 设置 z/OS SCRT 记录产品标识

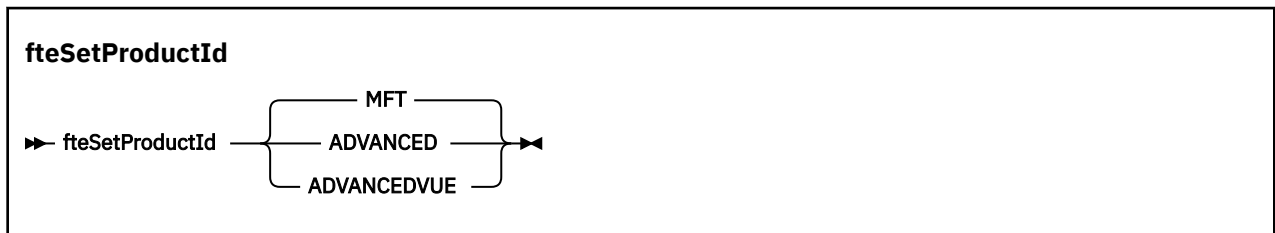
fteSetProductId 用于为安装设置要记录 Managed File Transfer 用法的产品类型。此命令仅在 z/OS 上有效。

用途

可以（在定义了至少一个协调队列管理器或者为安装创建了 [installation.properties](#) 文件之后）随时运行此命令。

请参阅 [报告产品信息](#)，以获取有关产品使用情况记录的更多信息。

语法



参数

用于用法记录的产品类型：

指定以下一项：

MFT

用法记录为独立 Managed File Transfer 产品，产品标识为 5655-MF9。

ADVANCED

用法记录为 IBM MQ Advanced for z/OS 产品的一部分，产品标识为 5655-AV9。

ADVANCEDVUE

用法记录为 IBM MQ Advanced for z/OS, Value Unit Edition 产品的一部分，产品标识为 5655-AV1。

返回码

0

命令成功完成。

1

命令失败，或者尚未将产品类型设置为请求的值。

相关任务

第 82 页的『[为 MFT 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。

fteSetupCommands: 创建 MFT command.properties 文件

fteSetupCommands 命令创建 Managed File Transfer `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。

要点: **ULW** 在 IBM MQ for UNIX, Linux, and Windows 上, 只有 IBM MQ 管理员 (和 mqm 组成员) 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。

z/OS 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:

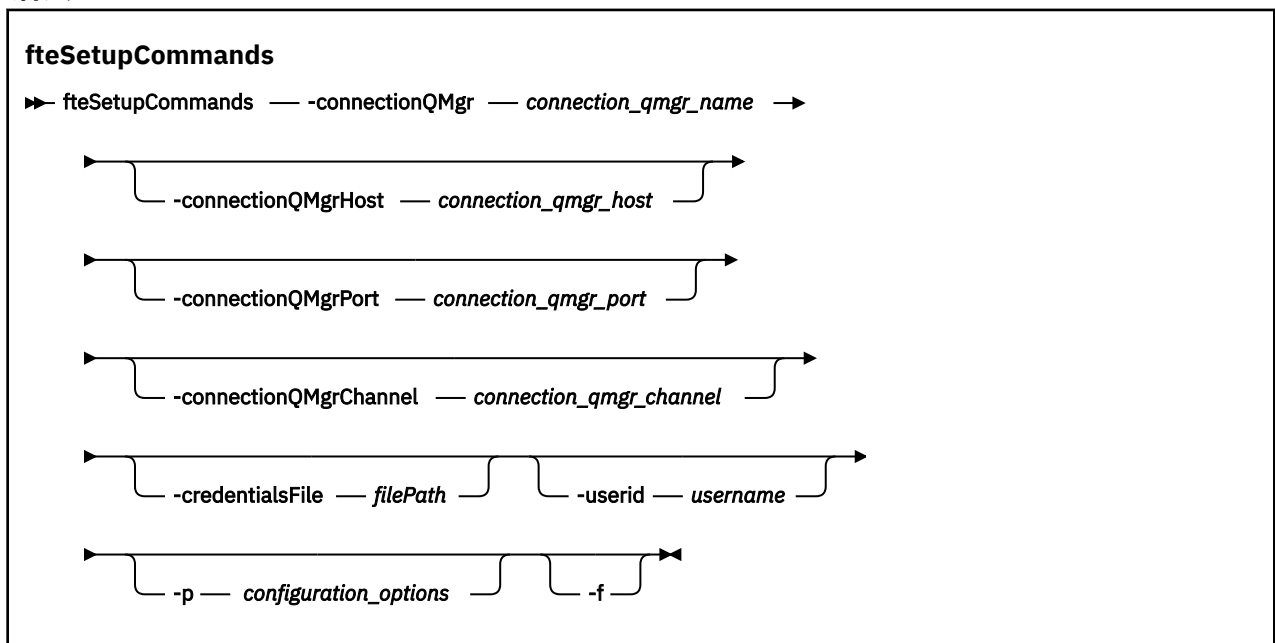
- 是 mqm 组的成员 (如果在系统上定义了 mqm 组)。
- **V9.0.0.1** 是 BFG_GROUP_NAME 环境变量中指定的组 (如果指定了一个组) 的成员。
- **V9.0.0.1** 在运行命令时, 在 BFG_GROUP_NAME 环境变量中未设置任何值。

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 457 页的『MFT `command.properties` 文件』。

语法



参数

-connectionQMgr (connection_qmgr_name)

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 (connection_qmgr_host)

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 (connection_qmgr_port)

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 (connection_qmgr_channel)

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 (configuration_options)

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。

此命令支持将一组 IBM MQ 认证详细信息添加到指定的 Managed File Transfer 凭证文件中。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。

-你说什么或 -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 457 页的『MFT 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 Managed File Transfer 联系该队列管理器所需的信息。


第 430 页的『fteSetupCoordination』

fteSetupCoordination 命令为 Managed File Transfer 创建属性文件和协调队列管理器目录。

fteSetupCoordination

fteSetupCoordination 命令为 Managed File Transfer 创建属性文件和协调队列管理器目录。

要点:  在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理员（和 mqm 组成员）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）。
- **V9.0.0.1** 是 BFG_GROUP_NAME 环境变量中指定的组（如果指定了一个组）的成员。
- **V9.0.0.1** 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 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 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。

z/OS 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：

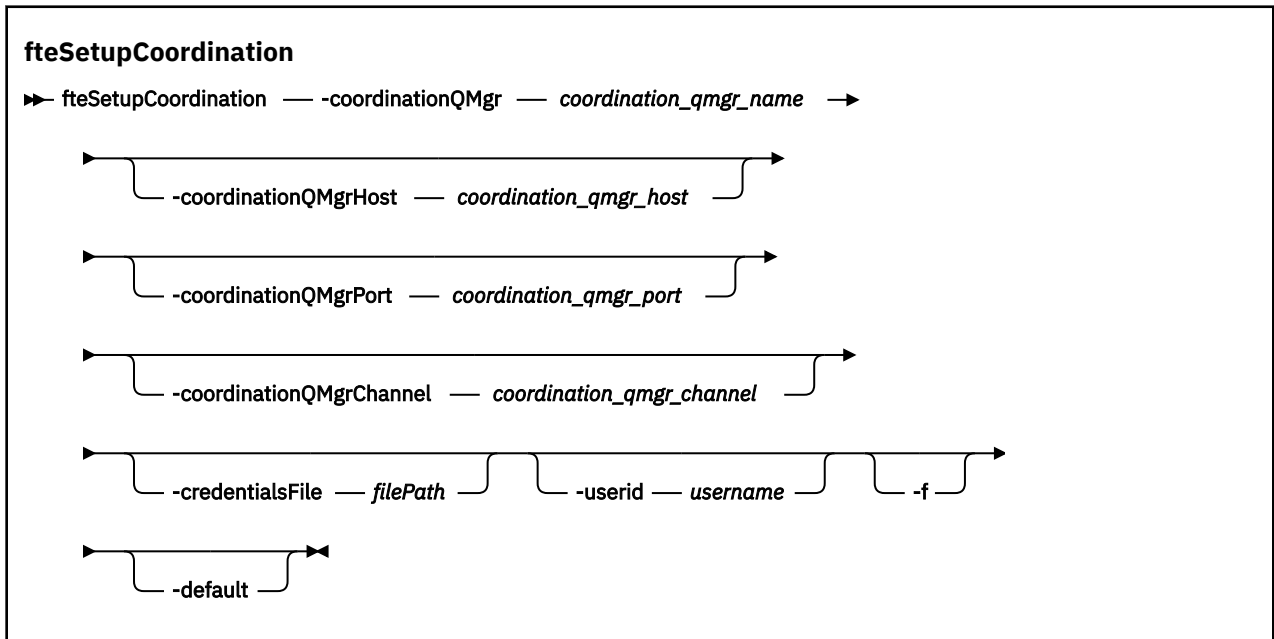
- SYSTEM.BROKER.DEFAULT.STREAM 队列
- SYSTEM.QPUBSUB.QUEUE.NAMELIST 名称列表
- SYSTEM.BROKER.DEFAULT.STREAM 和 SYSTEM.BROKER.ADMIN.STREAM 流

```
DEFINE TOPIC('SYSTEM.FTE') TOPICSTR('SYSTEM.FTE') REPLACE
ALTER TOPIC('SYSTEM.FTE') NPMSGDLV(ALLAVAIL) PMSGDLV(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](#)。

z/OS 如果使用的是 z/OS，那么可以使用根据 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 53 页的『创建 MFT 代理或记录器命令数据集』](#)。

语法



参数

-coordinationQMgr (*coordination_qmgr_name*)

必需。协调队列管理器的名称。此队列管理器必须是 IBM WebSphere MQ 7.0 或更高版本的队列管理器。

-coordinationQMgrHost (*coordination_qmgr_host*)

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 (*coordination_qmgr_channel*)

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。

此命令支持将一组 IBM MQ 认证详细信息添加到指定的 Managed File Transfer 凭证文件中。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。

-你说什么 或 -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 49 页的『多平台上的 MFT 配置选项』](#)

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 59 页的『在 z/OS 上配置 MQMFTCredentials.xml』](#)

如果 Managed File Transfer (MFT) 配置为启用安全性，那么连接认证需要所有 MFT 代理程序以及连接到队列管理器的命令来提供用户标识和密码凭证。

相关任务

[第 82 页的『为 MFT 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 460 页的『MFT agent.properties 文件』](#)

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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。

相关信息

[第 506 页的『coordination.properties 文件的 SSL 属性』](#)

fteShowAgentDetails

使用 **fteShowAgentDetails** 命令可显示特定 Managed File Transfer Agent 的详细信息。这些是由代理的 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。

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。要了解更多信息，请参阅 [第 49 页的『多平台上的 MFT 配置选项』](#)。

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 620 页的『MFT 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在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。有关更多信息，请参阅第 460 页的『MFT agent.properties 文件』。

对于 IBM WebSphere MQ 7.5.0 Fix Pack 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 9.0.2 **z/OS** **V 9.0.0.1** 对于 z/OS, 从 IBM MQ 9.0.2 和 IBM MQ 9.0.0 Fix Pack 1 开始, 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 460 页的『MFT agent.properties 文件』中的 **adminGroup** 属性。

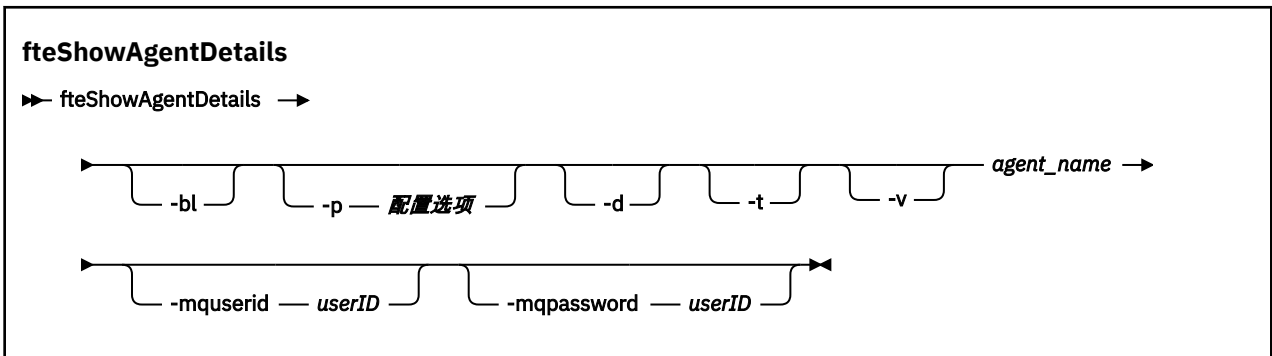
对于 IBM WebSphere MQ 7.5 或更高版本, 如果在代理所在的系统上运行该命令, 那么将会提供代理进程控制器和队列管理器的状态。您可以使用此信息来帮助确定问题。另外, 当您在代理所处的同一个系统上运行该命令时, 会为代理意外终止情况提供更多详细的代理状态信息。

要获取可能的代理状态值及其含义的列表, 请参阅第 521 页的『MFT 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 524 页的『MFT 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 228 页的『fteSetAgentTraceLevel』。

语法



参数

-bl

可选。额外输出代理的产品构建级别。

-p (configuration options)

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。

-t

可选。指定简洁方式。对于 IBM MQ 9.0.2, 缺省情况下输出包含 **Status Age** 信息。如果不想看到此信息, 那么可发出使用 **-t** 参数的命令。有关更多信息, 请参阅第 253 页的『如果代理被显示为处于 UNKNOWN 状态, 该怎么办』。

-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 620 页的『MFT 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 460 页的『MFT agent.properties 文件』。

agent_name

必需。要显示的 Managed File Transfer Agent 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。

-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。

-你说什么 或 -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
                    com.ibm.wmqfte.common:Any
                    com.ibm.wmqfte.Agent:1
  Trace FFDC:

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
```



```

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.
```

z/OS

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 395 页的『[fteListAgents: 列出协调队列管理器的 MFT 代理](#)』

使用 **fteListAgents** 命令可列出向特定协调队列管理器注册的所有 Managed File Transfer 代理程序。

第 521 页的『[MFT 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 524 页的『[MFT 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

使用 **fteShowLoggerDetails** 命令显示特定 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 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 49 页的『[多平台上的 MFT 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 524 页的『[MFT 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 525 页的『[MFT 记录器进程控制器状态值](#)』。

V 9.0.4

从 IBM MQ 9.0.4 开始，命令输出将显示记录器用于连接到队列管理器的连接信息。

V 9.0.4

如果在客户机方式下连接记录器，输出针对：

Last Status Reported

显示为 UNKNOWN

Status details

显示为 Information about the queue manager is not available because the logger has a client connection to queue manager.

语法

fteShowLoggerDetails

```
► fteShowLoggerDetails -p 配置选项 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 Managed File Transfer 记录器名称。

-你说什么 或 -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
```

```
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Logger Controller Information:
  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 524 页的『MFT 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 525 页的『MFT 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

fteStartAgent 命令从命令行启动 Managed File Transfer 代理程序。

用途

使用 **fteStartAgent** 命令来启动 Managed File Transfer 代理程序。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。

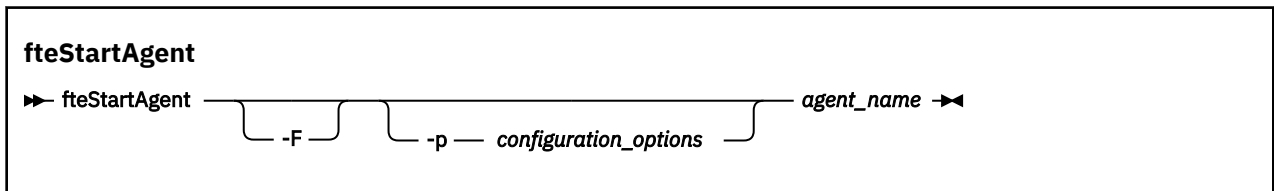
对于 IBM WebSphere MQ 7.5 或更高版本，代理进程控制器将管理代理的启动。但是，在某些情况下，例如当代理运行失败的频率比较高时，代理进程控制器可能会等待一段时间，然后再重新尝试启动代理。作为 IBM MQ 管理员，可以使用 **fteStartAgent** 命令来覆盖此等待时间值，并开始代理启动过程。如果代理进程控制器在等待队列管理器变为可用，那么该命令还将启动尝试重新连接到队列管理器的代理进程控制器。

Windows 如果已使用 **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 460 页的『MFT `agent.properties` 文件』，以了解更多信息。

语法



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，并且已使用 **fteCreateAgent** 或 **fteModifyAgent** 命令将代理程序配置为作为 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 Managed File Transfer 代理的名称。

-你说什么 或 -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（针对 UNIX 和 Linux 系统）中，将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 Managed File Transfer 冲突的附加库路径：

```
BFGCL0001E: An internal error has occurred. 异常为 : "CC=2;RC=2495;AMQ8568: "
找不到本地 JNI 库 "mqjbnd"。[3=mqjbnd]
```

如果 LD_LIBRARY_PATH 或 LIBPATH 环境变量设置为先引用 64 位版本的库，然后引用 32 位版本的库，而同时正在使用 32 位版本的 Java 运行代理（对于大部分平台，都是这种情况），那么会发生此错误。

要解决此问题，请设置 Managed File Transfer 代理属性 javaLibraryPath 以引用库的正确位置。例如，对于 AIX 上的 mqjbnd，请设置为：/usr/mqm/java/lib。对于 Linux 上的 mqjbnd，请设置为：/opt/mqm/java/lib

相关任务

[第 128 页的『将 MFT 代理作为 Windows 服务启动』](#)

您可以将代理作为 Windows 服务启动，以便在注销 Windows 时，代理继续运行并可接收文件传输。

[第 176 页的『列出 MFT 代理』](#)

您可以使用命令行或 IBM MQ Explorer 列出向特定队列管理器注册的 Managed File Transfer 代理程序。

[第 176 页的『停止 MFT 代理』](#)

您可以从命令行停止 Managed File Transfer 代理。在停止代理时，您将暂停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 131 页的『在 z/OS 上启动 MFT 代理』](#)

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

fteStartLogger 命令启动 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 89 页的『配置 MFT 记录器』主题。对于 IBM WebSphere MQ 7.5 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 MQ 管理员，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。

如果您已将记录器配置为使用 **fteModifyLogger** 命令作为 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。

仅当您要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 98 页的『MFT 记录器配置属性』

语法

fteStartLogger

```
► fteStartLogger -p 配置选项 -F logger_name ◄
```

参数

logger_name

必需。想要启动的 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。

-你说什么 或 -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 89 页的『配置 MFT 记录器』](#)

相关参考

[第 411 页的『fteModify 记录器 \(将 MFT 记录器作为 Windows 服务运行\)』](#)

使用 **fteModifyLogger** 命令来修改 Managed File Transfer 记录器，以便可以将其作为 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 447 页的『fteStopLogger』](#)

fteStopLogger 命令停止 Managed File Transfer 记录器。

[第 275 页的『MFT 记录器错误处理和拒绝』](#)

Managed File Transfer 记录器可识别两种类型的错误：每条消息的错误和通用错误。

fteStopAgent

使用 **fteStopAgent** 命令以受控方式停止 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。

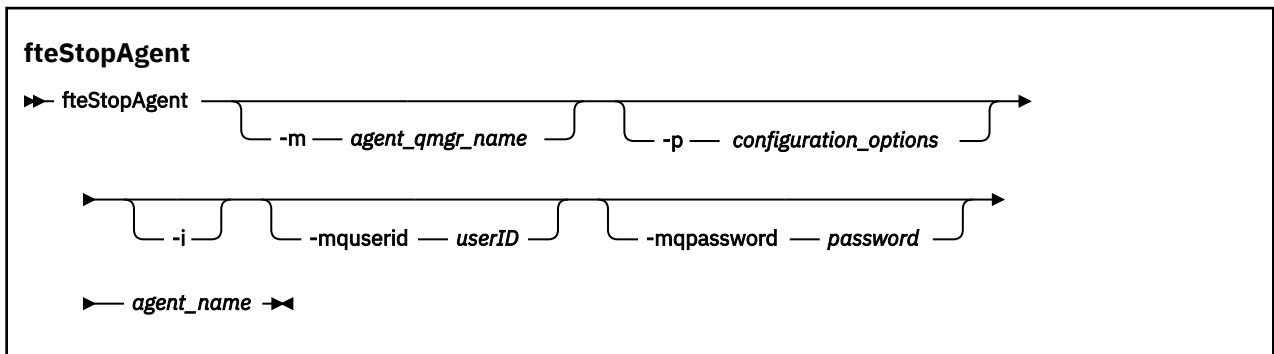
如果要停止的代理程序已连接到 IBM MQ 网络，那么可以从可以连接到 IBM MQ 网络并路由到代理队列管理器的任何系统运行 **fteStopAgent** 命令。具体而言，要运行此命令，您必须在此系统上安装和配置了 Managed File Transfer 组件（服务或代理）以与 IBM MQ 网络进行通信。如果没有可用的连接详细信息，那么将建立与本地系统上缺省队列管理器的绑定方式连接。如果 `command.properties` 不存在，那么将生成错误。

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，必须由运行该代理的相同用户来运行 **fteStopAgent** 命令。或者，如果代理正在 Windows 系统上运行，那么您可以采用管理员身份运行该命令。

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 460 页的『MFT `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 128 页的『将 MFT 代理作为 Windows 服务启动』。

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 293 页的『限制特定于 MFT 的资源的组权限』。

-p (configuration_options)

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 Managed File Transfer 代理的名称。

-你说什么 或 -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 176 页的『停止 MFT 代理』](#)

您可以从命令行停止 Managed File Transfer 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 442 页的『fteStartAgent』](#)

fteStartAgent 命令从命令行启动 Managed File Transfer 代理程序。

[第 176 页的『在 z/OS 上停止 MFT 代理』](#)

如果要从 JCL 将 Managed File Transfer Agent on z/OS 作为启动式任务运行，那么代理程序将接受 z/OS 操作程序命令 **MODIFY** 和 **STOP** 以及 **fteStopAgent** 命令。

fteStopLogger

fteStopLogger 命令停止 Managed File Transfer 记录器。

用途

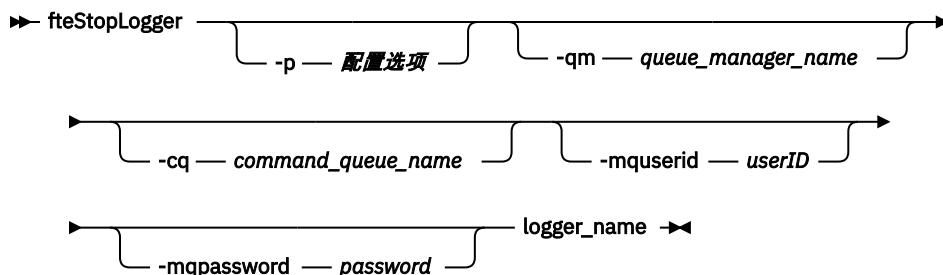
使用 **fteStopLogger** 命令以停止记录器。记录器可以是将受管文件传输活动的历史记录记录到文件中的文件记录器，也可以是将历史记录记录到数据库中的数据库记录器。

有关停止独立记录器进程的其他注意事项

如果记录器正在作为 Windows 服务运行，那么运行 **fteStopLogger** 命令将停止该 Windows 服务。

语法

fteStopLogger



参数

-p (configuration_options)

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 Managed File Transfer 记录器的名称。

-你说什么 或 -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 89 页的『配置 MFT 记录器』](#)

相关参考

[第 411 页的『fteModify 记录器 \(将 MFT 记录器作为 Windows 服务运行\)』](#)

使用 **fteModifyLogger** 命令来修改 Managed File Transfer 记录器，以便可以将其作为 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 444 页的『fteStartLogger』](#)

fteStartLogger 命令启动 Managed File Transfer 日志记录应用程序。

MFT 配置参考

在 MFT 属性中使用环境变量

在 IBM WebSphere MQ 7.5 中，可以在 Managed File Transfer 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

在本示例中，用户 `fteuser` 在 Windows 系统上使用环境变量 `USERPROFILE`：

```
wmqfte.database.credentials.file=%USERPROFILE%\logger\mqmftcredentials.xml
```

解析为以下文件路径：

```
C:\Users\fteuser\logger\mqmftcredentials.xml
```

在本示例中，用户 `fteuser` 在 UNIX 系统上使用环境变量 `HOME`：

```
transferRoot=$HOME/fte/
```

解析为以下文件路径：

```
/home/fteuser/fte/
```

MFT 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 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`。

V 9.0.1 对于 Redistributable Managed File Transfer Agent，将在您运行 **fteCreateEnvironment** 命令时设置数据路径。如果您运行此命令并使用 **-d** 参数指定您选择的位置，那么将为此位置设置数据路径。如果您不使用 **fteCreateEnvironment** 命令指定位置，将在 Redistributable Managed File Transfer Agent 解压缩到的根目录下创建目录 `mftdata`。Redistributable Managed File Transfer Agent 的 installation.properties 文件位于 `MQ_DATA_PATH\mqft\installations\MFTZipInstall` 目录中。

installation.properties 文件包含以下值：

表 39: 基本属性		
属性名	描述	缺省值
commandMessagePriority	<p>为 fteStopAgent、fteCancelTransfer 和 ftePingAgent 命令的内部消息和命令消息设置优先级。</p> <p>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。</p> <p>您还可以使用 <code>commandMessagePriority</code> 属性设置内部确认和期望确认消息的优先级。</p> <p>要将内部 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。</p> <p><code>commandMessagePriority</code> 属性的缺省值为 8。这意味着，如果代理命令队列上的 IBM MQ 属性 <code>DEFPRTY</code> (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 <code>DEFPRTY</code> 属性的值设置为 8 或 9 以保持 <code>commandMessagePriority</code> 属性的有效性，必须更改 <code>DEFPRTY</code> 或 <code>commandMessagePriority</code> 属性。</p>	<p>对于 IBM MQ 9.0.0.0，缺省值为 8。</p> <p>对于 APAR IT06213 之前的较早发行版，缺省值为 <code>MQPRI_PRIORITY_AS_Q_DEF</code> 常量，具有值 -1。</p>

表 39: 基本属性 (继续)

属性名	描述	缺省值
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	<p>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。</p> <p>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 Fix Pack 2 提供的函数，请将此属性设置为 8002 或 8.0.0.2。</p>	没有缺省值

表 39: 基本属性 (继续)

属性名	描述	缺省值
<p>     messagePublicationFormat </p>	<p> 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： </p> <p> messagePublicationFormat=mixed 在不使用 MQMD FORMAT (MQFMT_NONE) 的情况下发布消息，在 /LOG 主题树下发布（采用 MQMD 格式 MQFMT_STRING 发布）的消息除外。 </p> <p> messagePublicationFormat=MQFMT_NONE 在不使用 MQMD FORMAT 的情况下发布消息。 </p> <p> messagePublicationFormat=MQFMT_STRING 采用字符串格式发布消息。 </p> <p>  IBM MQ 9.0.2 中存在编码错误。在该版本中，当您指定 messagePublicationFormat=mixed 时，效果与指定 messagePublicationFormat=MQFMT_NONE 的效果相同。 </p> <p>  在 IBM MQ 9.0.3 中修正了此错误。 </p> <p> 在 IBM MQ 8.0 之前，MFT 代理程序将 XML 状态消息发布到 SYSTEM.FTE 主题 (MQFMT_STRING)。如果可以，必须更新先前使用 IBM WebSphere MQ 7.5 的应用程序，以采用 IBM MQ 8.0 或更高版本格式处理消息。如果无法更改应用程序，请将 messagePublicationFormat 属性设置为 MQFMT_STRING 以还原至 IBM WebSphere MQ 7.5 行为。 </p>	<p>messagePublicationFormat=mixed</p>
<p>   特定于 z/OS: </p>		

表 39: 基本属性 (继续)

属性名	描述	缺省值
 productId	要对其记录 MFT 用法的产品类型： <ul style="list-style-type: none"> • 独立 Managed File Transfer 产品。（MFT 为 productId）。 • IBM MQ Advanced 产品的一部分。（ADVANCED 为 productId）。 • IBM MQ Advanced for z/OS, Value Unit Edition 产品的一部分。（ADVANCEDVUE 为 productId）。 请参阅 报告产品信息，以获取有关产品使用情况记录的更多信息。  在 多平台 上忽略该属性。	MFT

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 49 页的『[多平台上的 MFT 配置选项](#)』

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 318 页的『[fteChangeDefaultConfigurationOptions](#)』

使用 **fteChangeDefaultConfigurationOptions** 命令来更改希望 Managed File Transfer 使用的缺省配置选项。配置选项的值定义 Managed File Transfer 使用的属性文件组。

MFT 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 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 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目录中。

coordination.properties 文件包含以下值：

表 40: 协调队列管理器属性		
属性名	描述	缺省值
coordinationQMgr	协调队列管理器的名称。	没有缺省值




属性名	描述	缺省值
coordinationQMGrHost	协调队列管理器的主机名或 IP 地址。	没有缺省值
coordinationQMGrPort	用于以客户机方式连接协调队列管理器的端口号。	1414
coordinationQMGrChannel	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。

属性名	描述	缺省值
代理属性:		
agentStatusJitterTolerance	<p>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。</p> <p>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。</p>	3000
代码页属性:		
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

表 41: 高级协调队列管理器属性 (继续)		
属性名	描述	缺省值
连接属性:		
javaLibraryPath	当以绑定方式连接到队列管理器时，Managed File Transfer 必须具有对 IBM MQ Java 绑定库的访问权。缺省情况下，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 。 如果要将特定 IBM MQ 前缀用于需要代理程序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.*
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 457 页的『 MFT command.properties 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 <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

表 41: 高级协调队列管理器属性 (继续)		
属性名	描述	缺省值
coordinationQMGrAuthenticationCredential sFile	包含用于连接到协调队列管理器的 MQ 连接凭证的文件的名称。	<p> 有关创建认证凭证文件的详细信息，请参阅 创建 MFT 凭证文件</p> <p> 请参阅 配置 MQMFTCredentials.xml，以获取有关此文件的位置和许可权的信息。</p> <p> 有关创建认证凭证文件的更多详细信息，请参阅 MFT 和 IBM MQ 连接认证</p>
预订属性:		
coordinationSubscriptionTopic	<p>使用此属性可指定 SYSTEM.FTE，以获取有关 IBM MQ 网络状态的出版物。所有工具仍将发布到 SYSTEM.FTE 主题，但是您可以更改 IBM MQ 拓扑以根据其内容将这些出版物分发到不同的主题。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。</p> <p>对于 IBM WebSphere MQ 7.5 和更高版本修订包，您需要针对 APAR IC96850 的临时修订，以便 IBM MQ Explorer 和 fteListMonitors 命令能够识别该属性。</p>	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 IBM MQ 队列管理器的名称。队列管理器 ERIS 是 Managed File Transfer 向其发送日志信息的队列管理器。

相关概念

第 49 页的『[多平台上的 MFT 配置选项](#)』

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 430 页的『[fteSetupCoordination](#)』

fteSetupCoordination 命令为 Managed File Transfer 创建属性文件和协调队列管理器目录。

第 504 页的『[MFT 的 SSL 属性](#)』

将 SSL 或 TLS 与 IBM MQ 和 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。

MFT 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 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 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 304 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 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目录中。

command.properties 文件包含以下值：

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。

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：		
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 hostname 和端口号，采用 IBM MQ CONNAME 格式。例如， <code>host_name(port_number)</code>	没有缺省值
安全属性：		

表 43: 高级命令队列管理器属性 (继续)

属性名	描述	缺省值
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 <i>None</i> 或 <i>java</i> 。	None
connectionQMGrAuthenticationCredentials 文件	包含用于连接到命令队列管理器的 MQ 连接凭证的文件的名称。	<p>z/OS 有关创建认证凭证文件的详细信息, 请参阅 创建 MFT 凭证文件</p> <p>ULW 请参阅 配置 MQMFTCredentials.xml, 以获取有关此文件的位置和许可权的信息。</p> <p>ULW 有关创建认证凭证文件的更多详细信息, 请参阅 MFT 和 IBM MQ 连接认证</p>
队列属性:		
dynamicQueuePrefix	<p>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。</p> <p>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。</p> <p>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。</p>	WMQFTE.*
modelQueueName	<p>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。</p> <p>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 453 页的『MFT coordination.properties 文件』。</p>	SYSTEM.DEFAULT.MODEL.QUEUE
连接属性:		

表 43: 高级命令队列管理器属性 (继续)		
属性名	描述	缺省值
javaLibraryPath	当以绑定方式连接到队列管理器时, Managed File Transfer 必须具有对 IBM MQ Java 绑定库的访问权。缺省情况下, 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	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 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 Managed File Transfer 命令 XML 消息可能会损坏。	false
代理属性		
    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 Managed File Transfer 命令连接的队列管理器。

相关概念

[第 49 页的『多平台上的 MFT 配置选项』](#)

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 504 页的『MFT 的 Java 系统属性』](#)

必须将一些 Managed File Transfer 命令和代理属性定义为 Java 系统属性, 因为它们为无法使用命令或代理属性机制的早期功能定义配置。

[第 504 页的『MFT 的 SSL 属性』](#)

将 SSL 或 TLS 与 IBM MQ 和 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 428 页的『[fteSetupCommands: 创建 MFT command.properties 文件](#)』

fteSetupCommands 命令创建 Managed File Transfer `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 319 页的『[fteCleanAgent: 清除 MFT 代理](#)』


使用 **fteCleanAgent** 命令，可通过删除 Managed File Transfer Agent 使用的持久性和非持久性队列中的消息来清除该代理使用的队列。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。

MFT agent.properties 文件

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，该文件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` 文件由安装程序或 **fteCreateAgent**、**fteCreateBridgeAgent** 或 **fteCreateCDAgent** 命令创建。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。

代理的 `agent.properties` 文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中。

 在 z/OS 上，`agent.properties` 文件位置为 `$BFG_CONFIG variable/mqft/config/coordination_qmgr_name/agents/agent_name`。

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 7.5 或更高版本，您可以在表示文件或目录位置的某些 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 448 页的『[在 MFT 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。

属性名	描述	缺省值
<code>agentName</code>	代理的名称。代理的名称必须符合 IBM MQ 对象命名约定。有关更多信息，请参阅第 520 页的『 MFT 对象命名约定 』。	没有缺省值
<code>agentDesc</code>	代理的描述 - 如果您选择创建描述。	没有缺省值
<code>agentQMGr</code>	代理队列管理器名称。	没有缺省值
<code>agentQMGrHost</code>	代理队列管理器的主机名或 IP 地址。	没有缺省值
<code>agentQMGrPort</code>	与代理队列管理器建立的客户机连接所使用的端口号。	1414
<code>agentQMGrChannel</code>	用于与代理队列管理器连接的 SVRCONN 通道名称。	SYSTEM.DEF.SVRCONN
<code>agentType</code>	代理类型： <ul style="list-style-type: none">• 标准非网桥代理 (STANDARD)• 协议网桥代理 (BRIDGE)• Connect:Direct 网桥代理 (CD_BRIDGE)• 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

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，请确保输入分隔符反斜杠 (\) 作为双反斜杠 (\\)，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 [Properties 类的 Javadoc](#)。




- [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](#)
-  [协议网桥代理日志属性](#)
- [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](#)
-  [传输恢复超时属性](#)
- [跟踪和记录属性](#)
- [传输限制属性](#)
- [用户出口例程属性](#)
- [IBM MQ 客户机压缩属性](#)
-  [特定于 z/OS 的属性](#)
- [其他属性](#)

表 45: 高级代理属性		
属性名	描述	缺省值
代理大小属性:		
agentCheckpointInterval	<p>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 Managed File Transfer 配置而言，无需修改其值。</p> <p>如果存在导致传输进入恢复状态的问题，那么传输只能恢复到检查点边界。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 Managed File Transfer 网络，增加该值可能很有益，这样可提高整体性能。</p>	1

表 45: 高级代理属性 (继续)		
属性名	描述	缺省值
agentChunkSize	<p>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 IBM MQ 消息大小。这是高级属性，对于大多数 Managed File Transfer 配置而言，无需修改其值。</p> <p>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。</p> <p>agentChunkSize 是一个整数值。例如：agentChunkSize=10240 将块大小设置为 10 KB。</p>	262144 字节（等于 256KB）
agentFrameSize	<p>传输帧的窗口数。这是高级属性，对于大多数 Managed File Transfer 配置而言，无需修改其值。</p> <p>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。</p> <p>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。</p> <p>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。</p>	5
agentWindowSize	<p>每个窗口的块数。这是高级属性，对于大多数 Managed File Transfer 配置而言，无需修改其值。</p> <p>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。</p> <p>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 IBM 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。</p> <p>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。</p>	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	<p>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）。</p> <p>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。</p> <p>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。</p>	10
连接属性：		
javaLibraryPath	当以绑定方式连接到队列管理器时，Managed File Transfer 必须具有对 IBM MQ Java 绑定库的访问权。缺省情况下，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：		

表 45: 高级代理属性 (继续)		
属性名	描述	缺省值
cdNode	<p>当您使用 Connect:Direct 网桥时所需的属性。</p> <p>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 191 页的『Connect:Direct 网桥』。</p>	没有缺省值
cdNodeHost	<p>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。</p> <p>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。</p> <p>如果您未设置 cdNode 属性, 那么会忽略该属性。</p>	本地系统的主机名或 IP 地址
cdNodePort	<p>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。</p> <p>如果您未设置 cdNode 属性, 那么会忽略该属性。</p>	1363
cdTmpDir	<p>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。</p> <p>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。</p> <p>如果您未设置 cdNode 属性, 那么会忽略该属性。</p> <p>此属性的值可以包含环境变量。</p> <p>请参阅第 448 页的『在 MFT 属性中使用环境变量』, 以了解更多信息。</p>	<pre>value_of_java.io.tmpdir /cdbridge-agentName</pre> <p>在 Windows 上,</p> <pre>value_of_java.io.tmpdir \cdbridge-agentName</pre>
cdTrace	<p>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。</p>	false
cdMaxConnectionRetries	<p>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。</p>	-1 (无数次尝试)
cdMaxPartialWorkConnectionRetries	<p>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。</p>	-1 (无数次尝试)
cdMaxWaitForProcessEndStats	<p>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。</p>	60000
cdAppName	<p>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。</p>	Managed File Transfer <i>current version</i> , 其中 <i>current version</i> 为产品的版本号。
cdNodeLocalPortRange	<p>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。</p> <p>缺省情况下, 操作系统选择本地端口号。</p>	None
cdNodeProtocol	<p>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:</p> <ul style="list-style-type: none"> • TCPIP • SSL • TLS 	TCPIP
cdNodeKeystore	<p>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。</p> <p>如果未将 cdNodeProtocol 属性设置为 SSL 或 TLS, 那么会忽略该属性。</p> <p>此属性的值可以包含环境变量。</p> <p>请参阅第 448 页的『在 MFT 属性中使用环境变量』, 以了解更多信息。</p>	None
cdNodeKeystoreType	<p>cdNodeKeystore 属性指定的密钥库的文件格式。以下值有效: jks 和 pkcs12。</p> <p>如果未将 cdNodeProtocol 属性设置为 SSL 或 TLS, 那么会忽略该属性。</p>	jks

表 45: 高级代理属性 (继续)		
属性名	描述	缺省值
cdNodeKeystoreCredentialsFile	包含 cdNodeKeystore 凭证的文件的路径。 此属性的值可以包含环境变量。 请参阅第 448 页的『在 MFT 属性中使用环境变量』，以了解更多信息。	 有关创建认证凭证文件的详细信息，请参阅 创建 MFT 凭证文件  请参阅 配置 MQMFTCredentials.xml ，以获取有关此文件的位置和许可权的信息。  有关创建认证凭证文件的更多详细信息，请参阅 MFT 和 IBM MQ 连接认证
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。 此属性的值可以包含环境变量。 请参阅第 448 页的『在 MFT 属性中使用环境变量』，以了解更多信息。	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 凭证的文件的路径。 此属性的值可以包含环境变量。 请参阅第 448 页的『在 MFT 属性中使用环境变量』，以了解更多信息。	请参阅 创建 MFT 凭证文件 和 配置 MQMFTCredentials.xml  有关创建认证凭证文件的更多详细信息也在 创建 MFT 凭证文件中  有关创建认证凭证文件的更多详细信息，请参阅 MFT 和 IBM MQ 连接认证
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。	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 IBM MQ 系统队列。系统队列以限定词 SYSTEM 为前缀。 注: IBM MQ、Managed File Transfer 和其他应用程序使用系统队列来传输重要信息。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 IBM MQ 集群队列。 注: 除 enableQueueInputOutput 属性外，还必须指定 enableClusterQueueInputOutput 代理属性。	false
maxDelimiterMatchLength	可以通过用于在“文件到消息”传输过程中将文本文件拆分为多条消息的 Java 正则表达式进行匹配的最大字符数。	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 Managed File Transfer 内部队列。有关更改该属性的信息，请参阅第 268 页的『用于设置与消息大小关联的 MQ 属性和 MFT 属性的指南』。	1048576

表 45: 高级代理属性 (继续)		
属性名	描述	缺省值
monitorGroupRetryLimit	<p>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。</p> <p>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。</p> <p>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。</p>	10
一般代理属性:		
agentStatusPublishRateLimit	<p>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)。</p> <p>如果将该属性设置为过小的值, 那么 IBM MQ 网络的性能可能会受到负面影响。</p>	30
agentStatusPublishRateMin	<p>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。</p>	300
enableMemoryAllocationChecking	<p>确定 Managed File Transfer Agent 是否会在接受传输之前检查是否有足够的内存可用于运行该传输。源和目标代理上都会执行该检查。如果没有足够的内存, 将拒绝此传输。</p> <p>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 Managed File Transfer 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。</p>	false
enableDetailedReplyMessages	<p>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 632 页的『文件传输日志消息格式』。</p> <p>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 663 页的『文件传输请求消息格式』。</p> <p>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。</p>	true
enableUserMetadataOptions	<p>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:</p> <ul style="list-style-type: none"> com.ibm.wmqfte.insertRecordLineSeparator com.ibm.wmqfte.newRecordOnLineSeparator com.ibm.wmqfte.convertLineSeparators <p>有关这些键含义的信息, 请参阅第 364 页的『fteCreateTransfer: 启动新的文件传输』。</p> <p>该属性的值可以为 true 或 false。</p>	false
failTransferOnFirstFailure	<p>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。</p> <p>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。</p> <p>如果将 failTransferOnFirstFailure 属性设置为 true, 那么代理将开始正常处理受管传输请求。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:</p> <ul style="list-style-type: none"> 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。 <p>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。</p>	false
itemsPerProgressMessage	<p>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。</p> <p>该属性可以设置的最大值为 1000。</p> <p>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。</p>	50

表 45: 高级代理属性 (继续)		
属性名	描述	缺省值
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 z/OS 系统上，该行为不会应用于连续数据集，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
enableMandatoryLocking	访问常规文件时，Managed File Transfer 对读取操作采取共享锁定，对写入操作采取互斥锁定。  在 Windows 上，只是建议文件锁定。将该属性设置为 true 时，Managed File Transfer 会强制文件锁定。在 Windows 上，这意味着如果一个应用程序打开一个文件，那么不会触发监视该文件，直至文件关闭为止。Managed File Transfer 传输使文件失败。  在 UNIX 类型的平台上，进程之间采用文件锁定。对于 UNIX 类型平台，设置此属性没有任何影响。 该属性仅适用于普通 Managed File Transfer 代理。Managed File Transfer 不支持文件服务器上的文件锁定机制。因此，该属性对于协议网桥代理无效，因为协议网桥代理在传输文件时不会锁定文件服务器上的文件。 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，但是如果并行传输数超过此限制，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	将此属性设置为 true 将使 Managed File Transfer Agent 能够通过多个 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： <ul style="list-style-type: none"> 第 475 页的『在集群中为多个通道配置 MFT 代理』 第 476 页的『为多个通道配置 MFT 代理：非集群』 此外，您还必须完成 Managed File Transfer 代理程序所需的标准 IBM MQ 配置步骤，详见第 80 页的『为首次使用配置 MFT』。 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:		

表 45: 高级代理属性 (继续)		
属性名	描述	缺省值
agentQMgrStandby	由 agentQMgr 定义的多实例代理队列管理器的备用实例在建立客户机连接时所使用的主机名和端口号, 采用 IBM MQ CONNAME 格式。例如, <code>host_name(port_number)</code> 当代理程序检测到连接中断错误 (例如 MQRC 2009) 时, 它会尝试连接到备用队列管理器。代理连接到备用队列管理器后, 代理将保持连接状态, 直到备用队列管理器变为不可用为止。	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
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。有关更多信息, 请参阅第 186 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 181 页的『ProtocolBridgePropertiesExit2: 查找协议文件服务器属性』。	没有缺省值
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> V 9.0.0 V 9.0.0 </div> <div> 协议网桥代理日志记录属性: </div> </div>		
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> V 9.0.0 V 9.0.0 </div> <div> agentLog </div> </div>	用于在协议网桥代理和 FTP/SFTP/FTPS 文件服务器之间启用或禁用 FTP 命令和响应的日志记录的键值对组件及操作。 例如: agentLog=on 打开所有组件的日志记录 agentLog=off 关闭所有组件的日志记录 agentLog=ftp=on, sftp=on, ftps=off 打开 FTP 和 SFTP 的日志记录, 并关闭 FTPS 的日志记录	没有缺省值
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> V 9.0.0 V 9.0.0 </div> <div> agentLogFileSize </div> </div>	定义捕获文件的最大大小 (兆字节)。与常规跟踪缺省文件大小的缺省值相同。	20
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> V 9.0.0 V 9.0.0 </div> <div> agentLogFiles </div> </div>	定义在废弃最旧的文件之前保留的最大捕获文件数。	10

表 45: 高级代理属性 (继续)

属性名	描述	缺省值
V 9.0.0 V 9.0.0 agentLogFilter	缺省情况下, 捕获与代理连接到的所有 FTP 服务器的通信。 例如: <ul style="list-style-type: none"> 对主机/IP 地址进行过滤 <pre>host=ftpprod.ibm.com, ftp2.ibm.com host=9.182.*</pre> 根据元数据进行过滤 <pre>metadata="outbound files to xyz corp"</pre> 	*
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 IBM 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 <code>\$FILEPATH</code> 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。值为 false 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 <code>fteCreateMonitor</code> 命令的匹配 <code>-bs</code> 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 45: 高级代理属性 (继续)

属性名	描述	缺省值
commandPath	<p>指定可调用命令的路径集，可以通过使用一下方法之一：</p> <ul style="list-style-type: none"> 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 Managed File Transfer Agent 命令 XML 模式（例如，managedCall 或 managedTransfer）。 <p>有关 commandPath 属性值的有效语法的信息，请参阅第 303 页的『commandPath MFT 属性』。</p> <p>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：</p> <ul style="list-style-type: none"> 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。 <p>此属性的值可以包含环境变量。</p> <p>请参阅第 448 页的『在 MFT 属性中使用环境变量』，以了解更多信息。</p> <p>V9.0.0.1 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：</p> <ul style="list-style-type: none"> IBM WebSphere MQ File Transfer Edition. IBM WebSphere MQ 7.5.0 Fix Pack 1 Managed File Transfer 组件（或更低版本）。 未设置安装属性 enableFunctionalFixPack=7502 的安装上的 IBM WebSphere MQ 7.5.0 Fix Pack 2 Managed File Transfer 组件（或更高版本）。 <p>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。</p>	None - 不能调用任何命令
<p>V9.0.0.1</p> <p>V9.0.0.1</p> <p>addCommandPathToSandbox</p>	<p>指定是否应将 commandPath 属性指定的目录（及其所有子目录）添加到：</p> <ul style="list-style-type: none"> 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。 <p>这可兼容以下发行版：</p> <ul style="list-style-type: none"> IBM WebSphere MQ File Transfer Edition. IBM WebSphere MQ 7.5.0 Fix Pack 1 Managed File Transfer 组件（或更低版本）。 未设置安装属性 enableFunctionalFixPack=7502 的安装上的 IBM WebSphere MQ 7.5.0 Fix Pack 2 Managed File Transfer 组件（或更高版本）。 <p>有关更多信息，请参阅第 303 页的『commandPath MFT 属性』。</p>	True
<p>V9.0.1</p> <p>V9.0.0.1</p> <p>V9.0.1</p> <p>V9.0.0.1</p> <p>additionalWildcardSandboxChecking</p>	<p>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。</p> <p>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罝，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罝而失败，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 43 页的『针对通配符传输的额外检查』）。</p> <p>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。</p>	None
sandboxRoot	<p>指定使用沙箱时包含和排除的根路径集。请参阅第 39 页的『使用 MFT 代理沙箱』以获取有关此功能的信息。</p> <p>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。</p> <p>sandboxRoot 属性在协议网桥代理上不受支持。</p> <p>您不能同时指定 sandboxRoot 属性和 userSandboxes 属性。</p> <p>此属性的值可以包含环境变量。</p> <p>请参阅第 448 页的『在 MFT 属性中使用环境变量』，以了解更多信息。</p>	None - 没有沙箱

表 45: 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。 此属性的值可以包含环境变量。 请参阅第 448 页的『在 MFT 属性中使用环境变量』，以了解更多信息。	启动代理进程的用户的根目录。
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 40 页的『使用 MFT 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 262 页的『当已调度的文件传输未运行或被延迟时该怎么做』。 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:		
agentQMGrAuthenticationCredentials File	包含 MQ 连接凭证的文件的名称。	 有关创建认证凭证文件的详细信息，请参阅 创建 MFT 凭证文件  请参阅 配置 MQMFTCredentials.xml ，以获取有关此文件的位置和权限的信息。  有关创建认证凭证文件的更多详细信息，请参阅 MFT 和 IBM MQ 连接认证
authorityChecking	指定是否启用第 298 页的『限制 MFT 代理操作的用户权限』中描述的安全功能。  从 IBM MQ 9.0.0 Fix Pack 3 和 IBM MQ 9.0.4 开始，inquire 在所有代理权限列表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 IBM MQ 的用户标识。如果指定了 java，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 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 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 45: 高级代理属性 (继续)		
属性名	描述	缺省值
xmlConfigReloadInterval	<p>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:</p> <ul style="list-style-type: none"> • ConnectDirectCredentials.xml • ConnectDirectNodeProperties.xml • ConnectDirectProcessDefinitions.xml • ProtocolBridgeCredentials.xml • ProtocolBridgeProperties.xml • UserSandboxes.xml 	30
跟踪和记录属性:		
javaCoreTriggerFile	<p>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。</p> <p>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:</p> <ul style="list-style-type: none"> • 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> <p>指定该属性后, 代理会在启动时输出以下消息:</p> <pre>BFGAG0092I The <insert_0> file will be used to request JVM diagnostic information.</pre> <p>此属性的值可以包含环境变量。</p> <p>请参阅第 448 页的『在 MFT 属性中使用环境变量』, 以了解更多信息。</p>	None
跟踪	<p>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>。</p>	None
outputLogFiles	保存的 <code>output.log</code> 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 <code>output.log</code> 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 <code>output.log</code> 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 <code>true</code> , 每当传输进入恢复状态时, 都会向 <code>output0.log</code> 文件中代理的事件日志报告诊断事件。	对于 IBM MQ 9.0.0.0, 缺省值为 <code>true</code>
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, 捕获的这些信息可能会很有帮助。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。	<code>false</code>
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	代理用于与消息的主题名称匹配的 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 45: 高级代理属性 (继续)

属性名	描述	缺省值
<p>V 9.0.3</p> <p>V 9.0.3</p> <p>resourceMonitorLog</p>	<p>用于开启或关闭日志记录的资源监视器和操作的键值对。</p> <p>可能的值为:</p> <ul style="list-style-type: none"> • 参考 • moderate • 详细 • 关 <p>例如:</p> <ul style="list-style-type: none"> • resourceMonitorLog=MON1,MON2=info:MON3=off 开启 MON1 和 MON2 的日志记录, 并关闭 MON3 的日志记录。 • resourceMonitorLog=info 对所有资源监视器开启 info 级别日志记录。 <p>资源监视器日志将写入名为 <code>resmoneventN.log</code> 的文件, 其中 <i>N</i> 代表数字; 例如, <code>resmonevent0.log</code>。</p> <p> 注意: 代理的所有资源监视器将写入同一个日志文件中。</p> <p>请参阅记录 MFT 资源监视器, 以获取更多信息。</p>	参考
<p>V 9.0.3</p> <p>V 9.0.3</p> <p>resourceMonitorLogFileSize</p>	<p>定义捕获文件的最大大小 (兆字节)。</p>	20
<p>V 9.0.3</p> <p>V 9.0.3</p> <p>resourceMonitorLogFiles</p>	<p>定义在废弃最旧的文件之前保留的最大捕获文件数。</p>	10
传输限制属性:		
maxDestinationTransfers	<p>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。</p> <p>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。</p> <p>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。</p>	<p>25 (对于 Connect:Direct 之外的所有代理)</p> <p>5 (对于 Connect:Direct 网桥代理)</p>
maxFilesForTransfer	<p>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。</p> <p>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。</p>	5000
maxSourceTransfers	<p>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。</p> <p>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。</p> <p>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。</p>	<p>25 (对于 Connect:Direct 网桥代理之外的所有代理)</p> <p>5 (对于 Connect:Direct 网桥代理)</p>
maxQueuedTransfers	<p>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。</p> <p>排队的传输请求的处理顺序取决于其优先级及排队的长短。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。</p> <p>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。</p>	1000
<p>V 9.0.1</p> <p>V 9.0.1</p>	传输恢复超时属性:	

表 45: 高级代理属性 (继续)

属性名	描述	缺省值
V 9.0.1 V 9.0.1 transferRecoveryTimeout	设置时间量（以秒为单位），在此期间，源代理会一直尝试恢复停止的文件传输。 未设置此属性时，代理的缺省行为是一直重试，直至成功恢复传输。您可以为传输恢复超时属性设置以下值： -1 代理继续尝试恢复停止的传输，直至传输完成。使用此选项相当于未设置属性时代理的缺省行为。 0 一旦进入恢复，代理将停止文件传输。 >0 在由指定正整数值设置的时间量（以秒为单位）内，代理继续尝试恢复停止的传输。例如，transferRecoveryTimeout=21600 表示代理会在进入恢复后六小时内一直尝试恢复传输。此参数的最大值为 99999999。	-1
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）。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。有关更多信息，请参阅第 749 页的『MFT 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。有关更多信息，请参阅第 186 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 216 页的『使用 MFT 传输 I/O 用户出口』。	没有缺省值。
IBM 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：		

表 45: 高级代理属性 (继续)		
属性名	描述	缺省值
      adminGroup	<p>安全管理器组。该组成员可以:</p> <ul style="list-style-type: none"> 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。 <p>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。</p> <pre>adminGroup=MFTADMIN</pre>	None
  bpxwdynAllocAdditionalOptions	<p>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, Managed File Transfer 可能确保安装数据设备 (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。</p> <p>使用 <code>agent.properties</code> 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 536 页的『不能与 MFT 一起使用的 BPXWDYN 属性』。</p>	缺省值如下: <ul style="list-style-type: none"> 对于 z/OS V1R8 和更高版本为 MOUNT
  armELEMTYPE	<p>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 ELEMTYPE 参数值。对于代理, 请将 ELEMTYPE 设置为 SYSBFAGAG。</p>	未设定
  armELEMENT	<p>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。</p>	未设定
其他属性:		
  legacyXMLMessageMQMDFormat	<p>通过代理生成的 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 Managed File Transfer XML 消息发送至 MQMD 格式字段为 MQSTR 的队列。</p> <p>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。</p> <p>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 Managed File Transfer 命令 XML 消息可能会损坏。</p>	false
 adjustScheduleTimeForDaylightSavings	<p>如果您的企业每天运行调度的传输, 因为调度的传输是使用以下命令创建的:</p> <ul style="list-style-type: none"> -oi 参数设置为 days, 并且 -tb 参数已设置为源 <p>例如, 在 <code>fteCreateTransfer</code> 命令上, 如果将此属性设置为 true, 那么当时钟向前移动 1 小时时, 将调度的传输时间向前移动 1 小时, 当时钟向后移动 1 小时时, 将调度的传输时间向后移动 1 小时。</p> <p>例如, 如果调度的传输将在凌晨 1:00 运行, 那么当时钟前进时, 传输将在凌晨 2:00 运行, 当时钟后退时, 传输将还原为凌晨 1:00。</p>	true

相关概念

第 49 页的『多平台上的 MFT 配置选项』

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 205 页的『恢复中文件传输的超时选项』

您可以设置时间量 (以秒为单位), 在此期间, 源代理会一直尝试恢复停止的文件传输。如果代理达到重试时间间隔的超时值时传输不成功, 则传输失败。

第 39 页的『MFT 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。

相关参考

第 504 页的『MFT 的 Java 系统属性』

必须将一些 Managed File Transfer 命令和代理属性定义为 Java 系统属性，因为它们为无法使用命令或代理属性机制的早期功能定义配置。

第 504 页的『MFT 的 SSL 属性』

将 SSL 或 TLS 与 IBM MQ 和 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 323 页的『fteCreateAgent (创建 MFT 代理)』

fteCreateAgent 命令创建 Managed File Transfer Agent 及其关联的配置。

第 327 页的『fteCreateBridgeAgent (创建并配置 MFT 协议网桥代理)』

fteCreateBridgeAgent 命令将创建 Managed File Transfer 协议网桥代理及其关联配置。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 335 页的『fteCreateCDAgent (创建 Connect:Direct 网桥代理)』

fteCreateCDAgent 命令用于创建 Managed File Transfer Agent 及其相关联的配置，以与 Connect:Direct 网桥一起使用。

第 448 页的『在 MFT 属性中使用环境变量』

在 IBM WebSphere MQ 7.5 中，可以在 Managed File Transfer 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。

在集群中为多个通道配置 MFT 代理

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。

除了第 80 页的『为首次使用配置 MFT』中列出的 Managed File Transfer 代理程序所需的标准 IBM MQ 配置步骤外，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 80 页的『为首次使用配置 MFT』

在首次使用 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 460 页的『MFT agent.properties 文件』

每个 Managed File Transfer Agent 都有自己的属性文件 agent.properties，该文件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。

为多个通道配置 MFT 代理：非集群

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。

除了第 80 页的『为首次使用配置 MFT』中列出的 Managed File Transfer 代理程序所需的标准 IBM MQ 配置步骤外，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 80 页的『为首次使用配置 MFT』

在首次使用 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 460 页的『MFT agent.properties 文件』

每个 Managed File Transfer Agent 都有自己的属性文件 agent.properties，该文件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。

enableUserMetadataOptions：支持的 MFT 用户定义的元数据键

如果代理属性 **enableUserMetadataOptions** 设置为值 true，那么支持为新传输请求指定以下用户定义的元数据键。

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 true，那么指定在读取面向记录的文件（如 z/OS 数据集）时，将在记录之间插入行分隔符。 如果该键设置为 false，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 true，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 false，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： 1. 当将用户定义的元数据键 com.ibm.wmqfte.newRecordOnLineSeparator 设置为 false 并且传输目标是面向记录的文件时。 2. 当将用户定义的元数据键 com.ibm.wmqfte.com.ibm.wmqfte.insertRecordLineSeparator 设置为 false 并且传输源是面向记录的文件时。	true

相关信息

第 461 页的『表 45』

[fteCreateTransfer -md 参数](#)

附加 MFT 代理配置文件

除 agent.properties 文件外，Managed File Transfer 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。

协议网桥凭证文件格式

Managed File Transfer Agent 配置目录中的 ProtocolBridgeCredentials.xml 文件定义了协议网桥代理用于向协议服务器授权自身的用户名和凭证信息。

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。

IBM WebSphere MQ 7.5 introduced a new <agent> element that contains the <server> or <serverHost> element for the named agent.

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- 7.5 或更高版本

以下模式描述了哪些元素在 IBM MQ 8 的 ProtocolBridgeCredentials.xml 文件中有效。

```
<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" 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。

The <server> element is not supported for 7.0.4 or earlier.

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	当 <server> 元素引用 FTPS 服务器时必需。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	当 <serverHost> 元素引用 FTPS 服务器时必需。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 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 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。

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 Managed File Transfer 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 WebSphere MQ 7.5 或者 IBM WebSphere MQ File Transfer Edition 7.0.2 及更高版本上不支持 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 7.5 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 448 页的『在 MFT 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： <ul style="list-style-type: none">• 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），请将该值设置为 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 <code>false</code> ，那么连接采用主动方式。如果您将该值设置为 <code>true</code> ，那么连接采用被动方式。缺省值为 <code>false</code> 。

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 <code>explicit</code> 。
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 <code>JKS</code> 。
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 <code>JKS</code> 。
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 <code>PBSZ/PROT</code> 命令之前还是之后向 FTPS 服务器发出 <code>USER/PASS</code> 命令。缺省值为 <code>false</code> ，这表示先发送 <code>USER/PASS</code> 命令，然后发送 <code>PBSZ/PROT</code> 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 <code>AUTH</code> 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 <code>TLS</code> 、 <code>SSL</code> 、 <code>TLS-C</code> 或 <code>TLS-P</code> ，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。 IBM WebSphere MQ 7.5 或者 IBM WebSphere MQ File Transfer Edition 7.0.2 及更高版本上不支持此属性。 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。 IBM WebSphere MQ 7.5 或者 IBM WebSphere MQ File Transfer Edition 7.0.2 及更高版本上不支持此属性。

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	<p>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。</p> <p>协议网桥代理支持的密码如下所示:</p> <ul style="list-style-type: none"> • blowfish-cbc • 3des-cbc • aes128-cbc • aes192-cbc • aes256-cbc • aes128-ctr • aes192-ctr • aes256-ctr • 3des-ctr • arcfour • arcfour128 • arcfour256 <p>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。</p>

Connect:Direct 凭证文件格式

Managed File Transfer Agent 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>

组元素，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。

<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	这是 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 <ul style="list-style-type: none"> wildcard - 使用通配符 regex - 使用 Java 正则表达式

<user>

提交传输请求的 IBM MQ 用户。

属性	描述
名	在 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 <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> wildcard - 使用通配符 regex - 使用 Java 正则表达式
cdUserId 或 cdUserIdCipher	Connect:Direct 网桥用于连接到其相关联的 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 文件传输期间充当辅助节点 (SNODE) 角色的 Connect:Direct 节点。

属性	描述
名	这是 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 <ul style="list-style-type: none"> 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

在该示例中，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 IBM 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 执行其传输部分时，它将使用用户名 pnodeuser 和密码 passw0rd1。

如果 Connect:Direct 传输中的辅助节点使用以前缀 FISH 开始的名称，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 传输中的辅助节点使用以前缀 CHIPS 开始的名称，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>
```

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 MFT 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 7.5 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 448 页的『在 MFT 属性中使用环境变量』。

节点

指定一个或多个 `Connect:Direct` 节点。

属性	描述
名	标识使用由节点元素指定的定义的 <code>Connect:Direct</code> 节点的名称的模式。模式匹配不区分大小写。
模式 (pattern)	指定用于 <code>name</code> 属性值的模式类型。 <code>pattern</code> 属性的有效值为： <ul style="list-style-type: none"> wildcard - 使用通配符 regex - 使用 Java 正则表达式 要了解有关 MFT 使用的正则表达式类型的信息，请参阅第 549 页的『MFT 使用的正则表达式』。
类型	指定与 <code>name</code> 属性给定的模式匹配的一个或多个 <code>Connect:Direct</code> 节点的操作系统类型。 <code>type</code> 属性的有效值为： <ul style="list-style-type: none"> Windows - 节点在 Windows 上运行

属性	描述
	<ul style="list-style-type: none"> 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 <p>该属性的值不区分大小写。</p>

示例

在此示例中，Connect:Direct 凭证文件名被指定为 ConnectDirectCredentials.xml。示例代码指定以下平台连接：

- 名称以“cdnodew”开头的所有 Connect:Direct 节点都在 Windows 平台上运行。
- 名称以“cdnodeu”开头的所有 Connect:Direct 节点都在 UNIX 平台上运行。
- 名称以“cdnodez”开头的所有 Connect:Direct 节点都在 z/OS 平台上运行。
- 所有其他 Connect:Direct 节点均在 UNIX 平台上运行。

Connect:Direct 网桥代理从文件开头到末尾搜索匹配项，并使用找到的第一个匹配项。

```
<?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>
```

Connect:Direct 进程定义文件格式

Connect:Direct 网桥代理配置目录中的 ConnectDirectProcessDefinitions.xml 文件指定用户定义的 Connect:Direct 进程，以在文件传输过程中启动。

ConnectDirectProcessDefinitions.xml 文件必须符合

ConnectDirectProcessDefinitions.xsd 模式。ConnectDirectProcessDefinitions.xsd 模式文档位于 MFT 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>

  <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。

条件 (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 550 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。

属性	描述
模式 (pattern)	指定用于 value 属性值的模式类型。 pattern 属性的有效值为 <ul style="list-style-type: none"> 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 550 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』。

进程

组元素，包含有关在找到匹配时调用的 Connect:Direct 进程所在位置的信息。

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 进程的文件名。该文件路径相对于 Connect:Direct 网桥代理配置目录。该属性是可选项，缺省值是使用由 MFT 生成的进程。对于 IBM WebSphere MQ 7.5 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 448 页的『在 MFT 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，那么它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:processSet>
</tns:cdprocess>
```

```

        </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>

```

使用 MFT 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` 属性会解释为 Java 正则表达式。有关更多信息，请参阅第 549 页的『MFT 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，那么代理会把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或数据集名称或队列名称的限定符中的零个或多个字符匹配。

?

问号与目录名称或数据集名称或队列名称的限定符中的一个字符完全匹配。

两个星号字符与零个或多个目录名匹配，或者与数据集名称或队列名称中的零个或多个限定符匹配。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
- `z/OS // '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。
- `*@QM1` 与队列管理器 `QM1` 上具有单个限定符的任何队列匹配。
- `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。
- `**@QM1` 与队列管理器 `QM1` 上的任何队列匹配。

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。

示例

此示例显示如何通过将以下 `<sandbox>` 元素添加到 `AGENT_JUPITER` 的配置目录中的文件 `UserSandboxes.xml`，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：

```
<?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

此示例说明了 MQMD 用户名为 account 且后跟单个数字的任何用户（例如，account4）如何完成以下操作：

- 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。

要允许具有 MQMD 用户名 account 的用户完成这些操作，请将以下 <sandbox> 元素添加到 AGENT_SATURN 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="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>

```

相关参考

[第 43 页的『针对通配符传输的额外检查』](#)

从 IBM MQ 9.0.1 (在 Continuous Delivery 中) 和 IBM MQ 9.0.0 Fix Pack 1 (在 Long Term Support 中) 开始，如果已使用用户或代理沙箱配置代理以限制代理可以将文件传输到的位置，那么可以指定要对该代理的通配符传输进行其他检查。

[第 40 页的『使用 MFT 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 460 页的『MFT agent.properties 文件』](#)

每个 Managed File Transfer Agent 都有自己的属性文件 agent.properties，该文件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。

MFT 记录器配置属性

Managed File Transfer 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 7.5 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 448 页的『在 MFT 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 `Properties` 类的 Javadoc。

绑定方式连接属性

表 47: 绑定方式的连接属性		
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。此参数是必需的, 并且是与队列管理器的绑定方式连接所需的所有参数。(有关用于连接到远程队列管理器的属性, 请参阅第 503 页的表 48。)	无缺省值
<code>wmqfte.message.source.type</code>	以下某个值: automatic subscription 缺省值。记录器在 <code>SYSTEM.FTE/Log/#</code> 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 <code>IBM MQ Explorer</code> 、 <code>MQSC</code> 或 <code>PCF</code>), 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 <code>IBM MQ</code> 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 <code>IBM MQ</code> , 以便队列接收通常由 <code>SYSTEM.FTE/Log/#</code> 。	<code>automatic subscription</code>
<code>wmqfte.message.source.name</code>	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








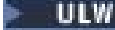
表 47: 绑定方式的连接属性 (继续)		
属性名	描述	缺省值
wmqfte.database.credentials.file	<p>包含用于连接至数据库的用户名和密码的文件。</p> <p>对于 IBM WebSphere MQ 7.5 或更高版本, 此属性的值可以包含环境变量。</p> <p>有关更多信息, 请参阅第 691 页的『MFT 凭证文件格式』。</p>	<p> 有关创建认证凭证文件的详细信息, 请参阅 创建 MFT 凭证文件</p> <p> 请参阅 配置 MQMFTCredentials.xml, 以获取有关此文件的位置和许可权的信息。</p> <p> 有关创建认证凭证文件的更多详细信息, 请参阅 MFT 和 IBM MQ 连接认证</p>
wmqfte.database.driver	<p>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, AIX 系统上 Db2 的类型 2 驱动程序需要文件 /opt/IBM/db2/V9.5/java/db2jcc.jar。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。</p> <p> 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>。</p> <p> 在 z/OS 系统上, 必须引用以下所有 JAR 文件:</p> <ul style="list-style-type: none"> • db2jcc.jar • db2jcc_license_cisuz.jar • db2jcc_javax.jar <p>如果数据库驱动程序由多个 JAR 文件组成 (例如, Db2 9.1 需要驱动程序 JAR 文件和许可证 JAR 文件), 请在此属性中包含所有这些 JAR 文件。通过针对平台使用类路径分隔符 (即, 在 Windows 系统上为分号字符 (;), 在其他平台上为冒号字符 (:)), 分隔多个文件名。</p>	无缺省值
wmqfte.database.exclude_duplicate_metadata	<p>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:</p> <ul style="list-style-type: none"> • 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 <p>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。</p>	true

表 47: 绑定方式的连接属性 (继续)		
属性名	描述	缺省值
wmqfte.database.host	<p>仅限 Db2:</p> <p>对于 IBM WebSphere MQ 7.5 或更高版本, 是指要使用 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, 那么也必须指定 <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。</p> <p>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。</p>	无缺省值
wmqfte.database.name	包含 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
wmqfte.database.port	<p>仅限 Db2:</p> <p>对于 IBM WebSphere MQ 7.5 或更高版本, 是指要使用 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, 那么也必须指定 <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。</p> <p>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。</p>	无缺省值
wmqfte.database.schema	包含 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	<p>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, AIX 系统上 Db2 的类型 2 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。</p> <p>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。</p>	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	<p>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。</p> <p><code>wmqfte.file.logger.fileSize=2d</code> 指定最大文件大小为 2 天的数据。</p>	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

表 47: 绑定方式的连接属性 (继续)

属性名	描述	缺省值
wmqfte.file.logger.mode	<p>正在使用的记录器方式: circular 或 linear。将该值设置为 CIRCULAR 或 LINEAR。</p> <p>CIRCULAR - 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。在该方式下写入了信息的文件的最大数量由使用 wmqfte.file.logger.fileCount 属性定义的值来控制。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), 那么最大大小将取决于这些时间段内系统中的日志消息吞吐量。在该方式下运行时使用的日志文件命名约定为: <i>logger_namenumbers-timestamp.log</i>, 其中:</p> <ul style="list-style-type: none"> • <i>logger_name</i> 是在 fteCreateLogger 命令中为记录器指定的名称。 • <i>number</i> 是文件在集合内的编号。 • <i>timestamp</i> 是创建文件时的时间戳记。 <p>例如, LOGGER1-20111216123430147.log</p> <p>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。在 linear 方式下运行时不会删除文件, 从而将忽略 wmqfte.file.logger.fileCount 属性, 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。在该方式下运行时使用的日志文件命名约定为: <i>logger_name-timestamp.log</i>, 其中:</p> <ul style="list-style-type: none"> • <i>logger_name</i> 是在 fteCreateLogger 命令中为记录器指定的名称。 • <i>timestamp</i> 是创建文件时的时间戳记。 <p>例如, LOGGER-20111216123430147.log</p>	无缺省值
wmqfte.max.retry.interval	<p>记录器遇到持续性错误时, 两次重试之间的最长时间 (秒)。</p> <p>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。</p> <p>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。</p>	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

表 47: 绑定方式的连接属性 (继续)		
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。	 有关创建认证凭证文件的详细信息, 请参阅 创建 MFT 凭证文件  请参阅 配置 MQMFTCredentials.xml , 以获取有关此文件的位置和许可权的信息。  有关创建认证凭证文件的更多详细信息, 请参阅 MFT 和 IBM MQ 连接认证
跟踪	<p>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。</p> <p>例如, com.ibm.wmqfte.databaselogger、com.ibm.wmqfte.databaselogger.operation=all。</p> <p>您可以在冒号分隔的列表中指定多个跟踪规范。例如,</p> <p>com.ibm.wmqfte.databaselogger=moderate: com.ibm.wmqfte.databaselogger.operation=all</p>	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

客户机方式连接属性

V 9.0.4

下表显示了支持到记录器队列管理器的客户机方式连接时需要的其他记录器属性。

表 48: 客户机方式的连接属性		
属性名	描述	缺省值
wmqfte.queue.manager.host	记录器队列管理器的主机名或 IP 地址。	无缺省值
wmqfte.queue.manager.port	记录器队列管理器正在侦听的端口。	1414
wmqfte.queue.manager.channel	记录器队列管理器上服务器连接通道的名称。	SYSTEM.DEF.SVRCONN
wmqfte.Ssl.CipherSuite	指定记录器和记录器队列管理器交换数据方式的 TLS 方面。 wmqfte.Ssl.CipherSuite 的值为 CipherSuite 名称。CipherSuite 名称将映射到记录器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。	无缺省值
wmqfte.Ssl.PeerName	指定必须与记录器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。	无缺省值
wmqfte.Ssl.TrustStore	指定记录器信任的证书的位置。 wmqfte.Ssl.TrustStore 的值为文件路径。如果文件路径是 Windows 文件路径, 那么反斜杠字符 (\) 必须使用另一个反斜杠字符进行转义 (\\)。 请注意, 此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.TrustStoreCredentialsFile	包含 wmqfte.Ssl.TrustStore 凭证的文件的 路径。 请注意, 此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.TrustStoreType	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
wmqfte.Ssl.KeyStore	指定记录器的专用密钥的位置。 wmqfte.Ssl.KeyStore 的值是文件路径。如果文件路径是 Windows 文件路径, 那么反斜杠字符 (\) 必须使用另一个反斜杠字符进行转义 (\\)。 请注意, 此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.KeyStore.CredentialsFile	包含 wmqfte.Ssl.KeyStore 凭证的文件的 路径。 请注意, 此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.KeyStoreType	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
wmqfte.Ssl.FipsRequired	指定要在记录器级别启用 FIPS 支持。该属性的值可以为 true 或 false。有关更多信息, 请参阅第 555 页的『MFT 中的 FIPS 支持』。	false

相关参考

第 448 页的『在 MFT 属性中使用环境变量』

在 IBM WebSphere MQ 7.5 中, 可以在 Managed File Transfer 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 504 页的『MFT 的 SSL 属性』

将 SSL 或 TLS 与 IBM MQ 和 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。




MFT 的 Java 系统属性

必须将一些 Managed File Transfer 命令和代理属性定义为 Java 系统属性，因为它们为无法使用命令或代理属性机制的早期功能定义配置。

通过定义环境变量 `BFG_JVM_PROPERTIES`，为要运行 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。

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 UNIX、Linux、Windows 和 z/OS，此长度没有限制。
<code>com.ibm.wmqfte.daemon.windows.windowsServiceLogFilesm</code>	（仅限 Windows。）指定要保存的 Windows 服务日志文件的最大数量。如果这些应用程序作为 Windows 服务运行，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，并包含关于服务启动和停止的消息。	5

相关概念

第 49 页的『多平台上的 MFT 配置选项』

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 227 页的『使用 MFT 的提示与技巧』

以下建议可帮助您最有效地使用 Managed File Transfer。

MFT 的 SSL 属性

将 SSL 或 TLS 与 IBM MQ 和 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。

以下 MFT 属性文件包括 SSL 属性：

- `agent.properties` 文件
- `coordination.properties` 文件
- `command.properties` 文件
-  `logger.properties` 文件

要了解有关将 SSL 与 Managed File Transfer 一起使用的信息，请参阅第 44 页的『为 MFT 配置 SSL 或 TLS 加密』。

对于 IBM WebSphere MQ 7.5 或更高版本，可以在表示文件或目录位置的某些 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 448 页的『在 MFT 属性中使用环境变量』。

agent.properties 文件的 SSL 属性

代理的 `agent.properties` 文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目录中。其属性包括以下 SSL 属性：

属性名	描述	缺省值
agentSslCipherSpec	<p>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。</p> <p>agentSslCipherSpec 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 Java 的 IBM MQ 类中的 SSL/TLS CipherSpecs 和 CipherSuites 和针对 JMS 的 IBM MQ 类中的 SSL/TLS CipherSpecs 和 CipherSuites 中。</p> <p>agentSslCipherSpec 类似于 agentSslCipherSuite。如果指定了 agentSslCipherSuite 和 agentSslCipherSpec，那么使用 agentSslCipherSpec 的值。</p>	None
agentSslCipherSuite	<p>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。</p> <p>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。</p> <p>agentSslCipherSuite 类似于 agentSslCipherSpec。如果指定了 agentSslCipherSuite 和 agentSslCipherSpec，那么使用 agentSslCipherSpec 的值。</p>	None
agentSslPeerName	<p>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。</p>	None
agentSslTrustStore	<p>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 (\) 必须进行转义 (\\)。</p> <p>对于 IBM WebSphere MQ 7.5 或更高版本，此属性的值可以包含环境变量。</p>	None
agentSslKeyStore	<p>指定代理程序专用密钥的位置。agentSslKeyStore 的值为文件路径。如果它是 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。</p> <p>对于 IBM WebSphere MQ 7.5 或更高版本，此属性的值可以包含环境变量。</p>	None
agentSslFipsRequired	<p>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 555 页的『MFT 中的 FIPS 支持』。</p>	false
agentSslKeyStoreType	<p>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。</p>	jks
agentSslKeyStoreCredentialsFile	<p>包含 agentSslKeyStore 凭证的文件的名称。</p> <p>对于 IBM WebSphere MQ 7.5 或更高版本，此属性的值可以包含环境变量。</p>	此属性的缺省值为 %USERPROFILE%\MQMFTCredentials.xml on Windows 和 \$HOME/MQMFTCredentials.xml on other 平台。
agentSslTrustStoreType	<p>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。</p>	jks

表 50: <i>agent.properties</i> 文件的 SSL 属性 (继续)		
属性名	描述	缺省值
agentSslTrustStoreCredentialsFile	包含 agentSslTrustStore 的凭证的文件的 对于 IBM WebSphere MQ 7.5 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 on Windows 和 \$HOME/ MQMFTCredentials.xml on other 平台。

coordination.properties 文件的 SSL 属性

coordination.properties 文件位于 *MQ_DATA_PATH/mqft/config/coordination_qmgr_name* 目录中。其属性包括以下 SSL 属性:

表 51: <i>coordination.properties</i> 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 Java 的 IBM MQ 类中的 SSL/TLS CipherSpecs 和 CipherSuites 和针对 JMS 的 IBM MQ 类中的 SSL/TLS CipherSpecs 和 CipherSuites 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 (\) 必须进行转义 (\\)。 对于 IBM WebSphere MQ 7.5 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 对于 IBM WebSphere MQ 7.5 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 on Windows 和 \$HOME/ MQMFTCredentials.xml on other 平台。

属性名	描述	缺省值
coordinationSslKeyStore	指定命令专用密钥的位置。coordinationSslKeyStore 的值为文件路径。如果它是 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 7.5 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
coordinationSslKeyStoreCredentialsFile	包含 coordinationSslKeyStore 凭证的文件的路径。 对于 IBM WebSphere MQ 7.5 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 on Windows 和 \$HOME/MQMFTCredentials.xml on other 平台。
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 555 页的『MFT 中的 FIPS 支持』。	false

command.properties 文件的 SSL 属性

command.properties 文件位于 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目录中。其属性包括以下 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。Java 的 IBM MQ 类中的 SSL/TLS CipherSpecs 和 CipherSuites 和针对 JMS 的 IBM MQ 类中的 SSL/TLS CipherSpecs 和 CipherSuites 中包含有效 CipherSpec 名称的列表。 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 (\) 必须进行转义 (\\)。 对于 IBM WebSphere MQ 7.5 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

表 52: *command.properties* 文件的 SSL 属性 (继续)

属性名	描述	缺省值
connectionSslTrustStoreCredentialsFile	包含 connectionSslTrustStore 凭证的文件的 路径。 对于 IBM WebSphere MQ 7.5 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 on Windows 和 \$HOME/MQMFTCredentials.xml on other 平 台。
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 7.5 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 7.5 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 路径。 对于 IBM WebSphere MQ 7.5 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 on Windows 和 \$HOME/MQMFTCredentials.xml on other 平 台。
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 555 页的『MFT 中的 FIPS 支持』。	false

logger.properties 文件的 SSL 属性

V 9.0.4

logger.properties 文件位于 *MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name* 目录中。支持到记录器队列管理器的客户机方式连接时需要的属性包括以下 SSL 属性：

表 53: *logger.properties* 文件的 SSL 属性

属性名	描述	缺省值
wmqfte.Ssl.CipherSuite	指定记录器和记录器队列管理器交换数据方式的 TLS 方面。 wmqfte.Ssl.CipherSuite 的值为 CipherSuite 名称。CipherSuite 名称将映射 到记录器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。	无缺省值
wmqfte.Ssl.PeerName	指定必须与记录器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。	无缺省值
wmqfte.Ssl.TrustStore	指定记录器信任的证书的位置。 wmqfte.Ssl.TrustStore 的值为文件路 径。如果文件路径是 Windows 文件路径，那 么反斜杠字符 (\) 必须使用另一个反斜杠字 符进行转义 (\\)。 请注意，此属性的值可以包含环境变量。	无缺省值

表 53: <i>logger.properties</i> 文件的 SSL 属性 (继续)		
属性名	描述	缺省值
wmqfte.Ssl.TrustStoreCredentialsFile	包含 wmqfte.Ssl.TrustStore 凭证的文件的路径。 请注意, 此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.TrustStoreType	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
wmqfte.Ssl.KeyStore	指定记录器的专用密钥的位置。 wmqfte.Ssl.KeyStore 的值是文件路径。如果文件路径是 Windows 文件路径, 那么反斜杠字符 (\) 必须使用另一个反斜杠字符进行转义 (\\)。 请注意, 此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.KeyStore.CredentialsFile	包含 wmqfte.Ssl.KeyStore 凭证的文件的路径。 请注意, 此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.KeyStoreType	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
wmqfte.Ssl.FipsRequired	指定要在记录器级别启用 FIPS 支持。该属性的值可以为 true 或 false。有关更多信息, 请参阅第 555 页的『MFT 中的 FIPS 支持』。	false

MFT 的 SHA-2 CipherSpec 和 CipherSuite

Managed File Transfer 支持 SHA-2 CipherSpec 和 CipherSuite。

在 IBM MQ 8.0 中, 要在代理和 IBM MQ 队列成员之间的连接上启用 SHA-2 CipherSpec 和 CipherSuite, 必须使用 IBM JRE 6.0 SR13 FP2、7.0 SR4 FP2 或更高版本。

要在 Managed File Transfer V8 中启用 SHA-2 CipherSpec 和 CipherSuite 以通过 FTPS 方式使用协议网桥连接到 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 554 页的『协议网桥提供的 FTPS 服务器支持』和第 481 页的『协议网桥属性文件格式』。

IBM i JVM 不支持在 MQ 8.0 中指定 CipherSpecs 中详细描述的最新密码。因此, 针对 IBM i 平台的 SHA-2 支持仅包含在 MQ 7.5 中指定 CipherSpecs 中详述的密码。

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 504 页的『MFT 的 SSL 属性』

将 SSL 或 TLS 与 IBM MQ 和 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。

MFT 文件记录器配置文件

除了 *logger.properties* 文件之外, Managed File Transfer 独立文件记录器还在其配置目录中具有 XML 配置文件。此配置文件称为 *FileLoggerFormat.xml*, 它定义由该文件记录器用于将消息写入日志文件的格式。该文件的内容必须符合 *FileLoggerFormat.xsd* 文件中定义的 XML 模式。

MFT 独立文件记录器缺省日志格式

Managed File Transfer 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="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>

```



```

    /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/@filesystem | destination/
queue/@filesystem</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 90 页的『MFT 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 514 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 90 页的『MFT 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 510 页的『MFT 独立文件记录器缺省日志格式』

Managed File Transfer 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 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 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 620 页的『MFT 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 622 页的『MFT 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 625 页的『MFT 调度列表消息格式』](#)。

SYSTEM.FTE/Templates

本主题包含描述 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 629 页的『MFT 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。IBM MQ Explorer 插件使用这些出版物来提供有关个别传输的进度信息。出版物的 XML 符合模式 *TransferStatus.xsd*。有关更多信息，请参阅 [第 630 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 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 632 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 653 页的『已调度的文件传输日志消息格式』](#)。
- 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 658 页的『MFT 监视器日志消息格式』](#)。

MFT 代理队列设置

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*

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*

参数	值 (如果适用)
DEFPRTY	0
DEFSOPT	SHARED

参数	值 (如果适用)
GET	ENABLED
MAXDEPTH	0
MAXMSGL	0
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
USAGE	正常
替换	

相关参考

第 323 页的『fteCreateAgent (创建 MFT 代理)』

fteCreateAgent 命令创建 Managed File Transfer Agent 及其关联的配置。

MFT 系统队列和系统主题

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。第 519 页的表 56 显示了每个队列上的消息类型：

队列名称	队列类型	用法
SYSTEM.FTE.AUTHAGT1.agent_name	权限	用于配置发送和接收传输请求的权限的队列。
SYSTEM.FTE.AUTHTRN1.agent_name	权限	用于配置启动和取消受管传输的权限的队列。还要启动受管调用。
SYSTEM.FTE.AUTHMON1.agent_name	权限	用于配置权限以允许用户创建或删除由同一用户创建的资源监视器的队列。
SYSTEM.FTE.AUTHOPS1.agent_name	权限	用于配置权限以删除由另一用户创建的资源监视器和调度传输的队列。
SYSTEM.FTE.AUTHSCH1.agent_name	权限	用于配置权限以创建或删除由同一用户创建的调度传输的队列。
SYSTEM.FTE.AUTHADM1.agent_name	权限	用于配置关闭代理程序的权限的队列，使用 fteStopAgent 命令上的 -m 选项。
SYSTEM.FTE.COMMAND.agent_name	操作	用于向代理程序发送命令请求的队列。
SYSTEM.FTE.DATA.agent_name	操作	目标代理用于保存源代理发送的数据的队列。

表 56: 队列名称, 类型和用法 (继续)

队列名称	队列类型	用法
SYSTEM.FTE.REPLY.agent_name	操作	用于从目标代理接收应答的队列。
SYSTEM.FTE.STATE.agent_name	操作	用于保存传输请求状态的队列。
SYSTEM.FTE.EVENT.agent_name	操作	用于保存资源监视器历史记录的队列。

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

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。

相关参考

第 298 页的『限制 MFT 代理操作的用户权限』

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执行的 Managed File Transfer 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。

MFT 对象命名约定

请针对 Managed File Transfer 对象使用以下命名约定:

- 代理程序和记录器名称:
 - 最多可包含 28 个字符, 并且不区分大小写。
 - 以小写或混合大小写形式输入将转换为大写
 - 必须符合标准 IBM MQ 对象命名约定。

下面详细描述了这些约定: [IBM MQ 对象的命名规则](#)。
- 除了 IBM MQ 对象命名约定外, 还有:
 - 不能在代理程序名称或记录器名称中使用正斜杠 (/) 字符
 - 不能在代理程序名称或记录器名称中使用百分比 (%) 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。
- 资源监视器和传输模板名称:
 - 不区分大小写
 - 以小写或混合大小写形式输入将转换为大写
 - 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:
 - 最小长度为 2 个字符, 但没有最大长度限制
 - 不区分大小写

- 必须符合标准 IBM MQ 对象命名约定。

下面详细描述了这些约定：[IBM MQ 对象的命名规则](#)。

IBM i 集成文件系统 (IFS) 中的文件

IBM i 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

z/OS 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 @。

MFT 管理参考

MFT 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，已突然关闭，或者可能正在运行，但遇到通信问题。有关更多信息，请参阅第 253 页的『如果代理被显示为处于 UNKNOWN 状态，该怎么办』。

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 460 页的『MFT agent.properties 文件』。

NO_INFORMATION

代理版本可以为 IBM WebSphere MQ File Transfer Edition 7.0.2 或更低版本。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。

V 9.0.2 对于 IBM MQ 9.0.2，在运行命令或查看连接到协调管理器的代理列表及其各个属性时，您可以看到代理的新 **Status Age** 值，该值显示自上次报告其状态以来的时间。有关更多信息，请参阅第 253 页的『如果代理被显示为处于 UNKNOWN 状态，该怎么办』。

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 395 页的『`fteListAgents`：列出协调队列管理器的 MFT 代理』

使用 `fteListAgents` 命令可列出向特定协调队列管理器注册的所有 Managed File Transfer 代理程序。

第 433 页的『`fteShowAgentDetails`』

使用 `fteShowAgentDetails` 命令可显示特定 Managed File Transfer Agent 的详细信息。这些是由代理的 Managed File Transfer 协调队列管理器存储的详细信息。

第 262 页的『在您认为文件传输卡住时应该怎么办』

在高负荷的系统上或者在源和目标代理之间发生网络问题时，传输可能会偶尔显示为卡住，进入排队或恢复状态。导致此情况的因素有很多。

第 253 页的『如果代理被显示为处于 UNKNOWN 状态，该怎么办』

您的代理正在运行并成功响应 `ftePingAgent` 命令，且正在正常传输项目。但是，`fteListAgents` 和 `fteShowAgentDetails` 命令以及 IBM MQ Explorer Managed File Transfer 插件报告代理处于 UNKNOWN 状态。

Multi MFT 进程控制器概述

IBM MQ Managed File Transfer (MFT) 进程控制器负责启动 MFT 代理程序，并在该进程因任何原因而结束时重新启动该进程。每个代理进程都有一个进程控制器。

注：进程控制器仅适用于 IBM MQ for Multiplatforms。

z/OS 在 IBM MQ for z/OS 上，代理进程由自动重新启动管理器 (ARM) 重新启动。有关此操作的更多信息，请参阅 [配置 MFT 以获取 z/OS 自动重新启动管理器 \(ARM\)](#)

进程控制器的工作方式

运行 `fteStartAgent` 命令时，它将启动该代理程序的进程控制器实例，然后该进程控制器将启动该代理程序进程。

运行 `fteStopAgent` 命令时，它会连接到该代理程序的进程控制器，并向其发送停止请求。进程控制器接收请求，停止代理进程，然后关闭自身。

进程控制器监视代理进程。如果代理进程意外停止，那么进程控制器会将其重新启动。

缺省情况下，如果代理程序进程在两分钟内停止了五次，那么进程控制器将自行关闭，并且不会再次尝试重新启动代理程序。在此情况下，您需要使用 **fteStartAgent** 命令手动重新启动代理程序。

您可以通过修改以下代理程序属性来更改此行为：

- **maxRestartCount**
- **maxRestartDelay**
- **maxRestartInterval**

如果您已将代理配置为使用 BINDINGS 传输连接到其代理队列管理器，那么进程控制器将在此队列管理器启动时创建与此队列管理器的连接。然后，进程控制器将监视此连接。

如果由于队列管理器变得不可用而导致连接中断，那么进程控制器将停止代理程序，然后尝试定期重新建立连接。

重新连接尝试之间的时间段由代理属性 **agentQMgrRetryInterval** 确定。一旦队列管理器再次可用并且进程控制器能够连接到该队列管理器，该进程控制器将重新启动代理进程。

注：当代理配置为使用 CLIENT 传输连接到其代理队列管理器时，如果该代理进程与队列管理器断开连接，那么该代理进程将保持活动状态。在这种情况下，代理进程会尝试定期重新连接自身。

有关此部分中提到的四个属性的更多信息，请参阅 *MFT agent.properties* 文件主题的 [Advanced agent properties: Process controller](#) 部分。

进程控制器日志文件

进程控制器将参考消息写入其事件日志。这是一个名为 `pceventN.log` 的文件，其中 N 是一个数字，可在以下目录中找到：`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/`

每个进程控制器事件日志文件的大小以及历史文件数由代理程序属性 **outputLogSize** 和 **outputLogFiles** 确定。

有关本节中提到的属性的更多信息，请参阅 *MFT agent.properties* 文件主题的 [高级代理程序属性: 跟踪和日志记录](#) 部分。

注：这些属性还用于确定代理程序日志文件 (称为 `outputN.log`) 以及进程控制器日志文件的大小和数量。

写入进程控制器事件日志的消息包括进程控制器的进程标识和代理进程的进程标识。下面显示了这些消息的一些示例：

```
[21/06/2022 16:17:40.000 GMT Daylight Time] 00000000000049e0
ProcessContro I BFGPC0003I: IBM MQ Managed File Transfer process controller started.
Log files located at: C:\ProgramData\IBM\MQ\mqft\logs\QM1\agents\AGENT1.

[21/06/2022 16:17:55.000 GMT Daylight Time] 00000000000049e0
ProcessContro I BFGPC0007I: IBM MQ Managed File Transfer process controller with process
identifier 18736 started AGENT1@QM1 with process identifier 1748.

[21/06/2022 16:19:20.000 GMT Daylight Time] 00000000000049e0
ProcessContro I BFGPC0027W: Process has ended with return code 1 and will be
restarted to attempt to recover the problem.

[21/06/2022 16:19:20.000 GMT Daylight Time] 00000000000049e0
ProcessContro I BFGPC0007I: IBM MQ Managed File Transfer process controller with process
identifier 18736 started AGENT1@QM1 with process identifier 1304.
```

此处，与代理程序 AGENT1 关联的进程控制器正在以进程标识 18736 运行。

最初，它启动了代理进程-此进程的进程标识为 1748。

代理程序启动后不久，进程控制器检测到它已意外停止，因此将其重新启动。重新启动后，代理进程的进程标识为 1304。

相关参考

第 525 页的『[MFT 进程控制器退出代码](#)』

如果 Managed File Transfer 进程控制器结束，那么将生成 BFGPC0004I 消息，其中的退出代码给出了进程控制器结束的原因。

MFT 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 433 页的『**fteShowAgentDetails**』

使用 **fteShowAgentDetails** 命令可显示特定 Managed File Transfer Agent 的详细信息。这些是由代理的 Managed File Transfer 协调队列管理器存储的详细信息。

MFT 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 98 页的『MFT 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 IBM WebSphere MQ File Transfer Edition 7.0.2 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 440 页的『fteShowLoggerDetails』

使用 **fteShowLoggerDetails** 命令显示特定 Managed File Transfer 记录器的详细信息。

MFT 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 440 页的『fteShowLoggerDetails』

使用 **fteShowLoggerDetails** 命令显示特定 Managed File Transfer 记录器的详细信息。

MFT 进程控制器退出代码

如果 Managed File Transfer 进程控制器结束，那么将生成 BFGPC0004I 消息，其中的退出代码给出了进程控制器结束的原因。

将显示以下消息，指示进程控制器已结束：

```
BFGPC0004I IBM MQ Managed File Transfer process controller ended with exit code reason_code.
```

其中，*reason_code* 显示进程控制器结束的原因。

注：来自进程控制器的退出代码主要反映标准操作系统退出代码，但是某些退出代码是针对特定用途定义的，并且在进程控制器日志文件中总是伴随特定消息。

原因码	描述
RC_SUCCESS = 0	进程控制器已成功结束。
RC_FAILURE = 1	常规进程控制器失败返回码（通常不应返回）。

表 57: 原因码值 (继续)	
原因码	描述
RC_EXIT = 2	已强制进程控制器退出（例如，诊断系统请求进程控制器终止）。
RC_ABEND = 70	进程控制器遇到不可恢复的问题且正在强制终止。
RC_QMUNAVAIL = 75	进程控制器无法继续，因为进程控制器的队列管理器不可用。
RC_CONFIG = 78	由于启动配置数据存在问题，进程控制器无法继续。

这些退出代码将写入到 pceventX.log，其中 X 可以是任何数字，例如，日志文件名可以为 pcevent0.log。

相关信息

[Managed File Transfer 诊断消息：BFGPC0001 - BFGPC9999](#)

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 527 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 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 528 页的『在 z/OS 上的数据集之间传输』](#)

您可以使用 Managed File Transfer 在 z/OS 数据集之间执行传输。仔细复审以下行为以确保正确传输数据集。

[第 530 页的『与 Connect:Direct 节点之间传输数据集』](#)

您可以使用 Connect:Direct 网桥在 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 531 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 536 页的『不能与 MFT 一起使用的 BPXWDYN 属性』](#)

使用 **fteCreateTemplate** 命令、**fteCreateTransfer** 命令或 agent.properties 文件中的 **bpxdynAllocAdditionalOptions** 属性时，不能指定某些 BPXWDYN 选项。

[第 536 页的『使用 MFT 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 Managed File Transfer 的文本文件传输行为。

[第 538 页的『在 Connect:Direct 和 MFT 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 Managed File Transfer Agent 和 Connect:Direct 节点之间的传输中的文本文件传输行为。

[第 539 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 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 539 页的『在 IBM i 系统上传输文件』](#)

如果您使用文本方式的 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 543 页的『在 IBM i 上传输 QSYS.LIB 中的保存文件』](#)

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。

第 544 页的『传输世代数据组 (GDG)』

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 545 页的『对 MFT 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 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。

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，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，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。如果将目标指定为目录而非文件，那么目标文件名是不包括高级限定词 (HLQ) 的数据集名称。
- **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 526 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 528 页的『在 z/OS 上的数据集之间传输』

您可以使用 Managed File Transfer 在 z/OS 数据集之间执行传输。仔细复审以下行为以确保正确传输数据集。

第 364 页的『fteCreateTransfer: 启动新的文件传输』

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。

在 z/OS 上的数据集之间传输

您可以使用 Managed File Transfer 在 z/OS 数据集之间执行传输。仔细复审以下行为以确保正确传输数据集。

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。

此外，目标代理正在运行的用户标识需要具有安装磁带的正确权限。请参阅企业正在使用的外部安全管理器的文档，以获取有关如何执行此操作的信息。

如果要从磁带传输到数据集。

为了访问磁带上的数据集，源代理正在运行的用户标识需要具有安装磁带的相应权限。请参阅企业正在使用的外部安全管理器的文档，以获取有关如何执行此操作的信息。

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 [常见 MFT 问题](#) 中的 z/OS 部分。
- **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)

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 526 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 544 页的『传输世代数据组 \(GDG\)』](#)

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 530 页的『与 Connect:Direct 节点之间传输数据集』

您可以使用 Connect:Direct 网桥在 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 527 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 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。

z/OS 与 Connect:Direct 节点之间传输数据集

您可以使用 Connect:Direct 网桥在 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 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 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 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 364 页的『fteCreateTransfer: 启动新的文件传输』和第 351 页的『fteCreateTemplate: 创建新的文件传输模板』。

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 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 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本
- 源代理为 Connect:Direct 网桥代理, 因此为 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本, 而目标代理为 IBM WebSphere MQ File Transfer Edition 7.0.3 或更低版本

如果目标代理为 Connect:Direct 网桥代理, 而源代理为 IBM WebSphere MQ File Transfer Edition 7.0.3 或更低版本, 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。

对于源代理为 IBM WebSphere MQ File Transfer Edition 7.0.4 或更高版本的传输，您还可以使用此语法作为常规 **-ds** 和 **-dp** 参数的替代方法。例如，如果您希望在所有情况中使用一致语法，而某些情况涉及的源代理为 IBM WebSphere MQ File Transfer Edition 7.0.3 或更低版本，请使用 **-df** 和 **-dd** 参数。

注: 如果传输目标为 PDS，并且目标代理为 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 Managed File Transfer 设置，并且作为参数传递给 Connect:Direct **COPY** 进程。您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 531 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 191 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 Managed File Transfer 组件的 Connect:Direct 网桥在 MFT 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，从 z/OS 上的 Managed File Transfer 代理将数据集传输到 z/OS 上的 Connect:Direct 节点。

相关参考

第 528 页的『在 z/OS 上的数据集之间传输』

您可以使用 Managed File Transfer 在 z/OS 数据集之间执行传输。仔细复审以下行为以确保正确传输数据集。

第 364 页的『fteCreateTransfer: 启动新的文件传输』

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 351 页的『fteCreateTemplate: 创建新的文件传输模板』

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 289 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 58: Connect:Direct COPY 语句的参数和 Managed File Transfer 使用的等效 BPXWDYN 关键字	
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 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 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 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. 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。
REPLACE NOREPLACE	无等效 BPXWDYN。当目标数据集已存在于目标系统中的行为由传输请求中的 -de (destination_file_behavior) 参数值定义。有关目标数据集已存在时的 Managed File Transfer 缺省行为的更多信息, 请参阅第 528 页的『在 z/OS 上的数据集之间传输』。
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

Connect:Direct COPY 语句的参数	BPXWDYN 关键字
DSNTYPE	DSNTYPE。为该关键字指定 PDS 值将覆盖由 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

DCB 参数的子参数	BPXWDYN 关键字
model-file-name	不存在任何映射
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 60: Connect:Direct **COPY From** 语句的 **DISP** 参数的子参数, 以及 Managed File Transfer 使用的等效 BPXWDYN 关键字

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。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。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。Managed File Transfer 将此子参数设置为 KEEP 。

表 61: Connect:Direct **COPY To** 语句的 **DISP** 参数的子参数, 以及 Managed File Transfer 使用的等效 BPXWDYN 关键字

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。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 。Managed File Transfer 不支持传输请求中提供的关键字 RPL 。
[KEEP CATLG]	[KEEP CATLOG] 或 PATHDISP	指定传输成功完成后数据集的状态。Managed File Transfer 将该子参数设置为 CATALOG 。
[KEEP CATLG DELETE]	[KEEP DELETE] 或 PATHDISP	指定在传输异常完成后数据集的状态。Managed File Transfer 将该子参数设置为 DELETE 。

表 62: Connect:Direct **COPY** 语句的 **LABEL** 参数的子参数, 以及 Managed File Transfer 使用的等效 BPXWDYN 关键字

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 63: *Connect:Direct COPY* 语句的 **VOL** 参数的子参数, 以及 *Managed File Transfer* 使用的等效 **BPXWDYN** 关键字

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 64: *Connect:Direct COPY* 语句的 **SYSOPTS** 参数的子参数, 以及 *Managed File Transfer* 使用的等效 **BPXWDYN** 关键字

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 <i>Managed File Transfer</i> 传输选项。有关更多信息, 请参阅第 536 页的『使用 MFT 传输文本文件』。
数据类型	不存在任何映射. <i>Managed File Transfer</i> 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。
XLATE	当 DATATYPE 的值为 TEXT 时, 不存在任何映射. <i>Managed File Transfer</i> 将此值设置为 NO。
STRIP.BLANKS	当 DATATYPE 的值为 TEXT 时, 不存在任何映射. <i>Managed File Transfer</i> 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 65: *Connect:Direct COPY* 语句的 **SPACE** 参数的子参数, 以及 *Managed File Transfer* 使用的等效 **BPXWDYN** 关键字

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 191 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 Managed File Transfer 组件的 Connect:Direct 网桥在 MFT 和 IBM Sterling Connect:Direct 之间传输文件。

相关任务

z/OS 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，从 z/OS 上的 Managed File Transfer 代理将数据集传输到 z/OS 上的 Connect:Direct 节点。

相关参考

与 Connect:Direct 节点传输数据集

您可以使用 Connect:Direct 网桥在 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。

z/OS 不能与 MFT 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 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。
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。
RECORG(LS)	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 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。

使用 MFT 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 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 364 页的『`fteCreateTransfer`: 启动新的文件传输』和第 133 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。

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
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 460 页的『MFT 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。

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 will 将行拆分为多条记录和块。	否

在运行 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 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，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 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。

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 MFT 相同
已纠正的块	单个空格	填充空格的记录	空行	与 MFT 相同
变量块	空行	空记录	空行	与 MFT 相同
变量块	单个空格	单个空格记录	单个空格	与 MFT 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 526 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 538 页的『在 Connect:Direct 和 MFT 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 Managed File Transfer Agent 和 Connect:Direct 节点之间的传输中的文本文件传输行为。


第 570 页的『MFT 的可用代码页』

本参考主题列出了 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 MFT 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 Managed File Transfer Agent 和 Connect:Direct 节点之间的传输中的文本文件传输行为。

有关 Managed File Transfer 中文本传输行为的信息，请参阅第 536 页的『使用 MFT 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。
 - 如果远程节点在 UNIX 或 Linux 系统上，请选择 UNIX。
 -  如果远程节点在 z/OS 系统上，那么请选择 OS/390。

Connect:Direct 网桥不支持到其他操作系统上的远程节点的传输。

- 确保对于作为文件传输目标或源的每个远程节点，您在 Connect:Direct 网桥代理配置目录下的 ConnectDirectNodeProperties.xml 文件中指定了远程 Connect:Direct 节点的操作系统类型。有关更多信息，请参阅第 120 页的『配置 ConnectDirectNodeProperties.xml 文件以包含有关远程 Connect:Direct 节点的信息』和 第 490 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 Managed File Transfer Agent，那么该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 536 页的『使用 MFT 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 Managed File Transfer 的文本文件传输行为。

第 288 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 MFT 代理和 Connect:Direct 节点之间传输文件时，会执行代码页转换和行尾字符转换。该传输使用 Connect:Direct 网桥节点的网络映射中的操作系统信息来确定远程节点的行尾字符。如果网络映射中的信息不正确，那么行尾字符转换可能未正确执行。

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 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 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。

确保了协议网桥代理外，在 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。

需要在 SFTP 文件服务器上创建新目录的受管传输请求

Managed File Transfer 协议网桥代理使用第三方 JSch 库通过 SFTP 协议与文件服务器进行通信。如果协议网桥代理尝试将文件传输到文件服务器上不存在的目录中，并且 JSch 无法执行所请求的 SFTP 操作来创建该目录，因为协议网桥代理登录到文件服务器的用户没有执行此操作的许可权，那么 JSch 会向协议网桥代理抛出异常。然后，协议网桥代理将受管传输标记为“失败”，并生成补充消息。如果 JSch 提供了有关故障的更多信息，那么协议网桥代理会在补充消息中包含此信息：

```
BFGTR0072E: 由于以下异常，传输未能完成：
BFGBR0119E: 网桥代理无法创建目录 目录名，因为来自 JSch 异常的消息
```

V 9.0.0.9 从 IBM MQ 9.0.0 Fix Pack 9 开始，如果 JSch 异常不包含有关故障的任何更多信息，那么协议网桥代理将生成以下补充消息：

```
BFGTR0072E: 由于以下异常，传输未能完成：
BFGBR0209E: 网桥代理无法创建目录 directory name
```

相关概念

第 177 页的『协议网桥』

通过协议网桥，Managed File Transfer (MFT) 网络可以访问您的 MFT 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。

IBM i 在 IBM i 系统上传输文件

如果您使用文本方式的 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，Managed File Transfer 会转换数据。但是，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。该属性的缺省值基于语言环境（但是您可以在 IBM i 系统上使用以下部分中描述的 `SystemDefault.properties` 文件来覆盖该缺省值：第 540 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，您可以使用 `WRKJVMJOB`（选项 7）来显示当前 Java 系统属性，以检查运行代理作业的 JVM 的文件编码值。（`WRKJVMJOB` 命令在 IBM i V5R4 系统上不存存在。）

如果您计划使用 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 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_STUDIO` 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_STUDIO') 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

传输文件时转换数据的另一种方式是创建传输定义以指定文件编码，或使用 `fteCreateTransfer` 命令的 `-sce` 和 `-dce` 参数。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 526 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 543 页的『在 IBM i 上传输 QSYS.LIB 中的保存文件』

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。

相关信息

在 IBM i 上安装 IBM MQ 服务器

IBM i 在 IBM i 上传输 QSYS.LIB 中的物理文件

Managed File Transfer 支持在两个 IBM i 系统间传输 QSYS.LIB 文件系统中的物理文件成员。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 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 MFT 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 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，如以下示例所示：

- ABCLIB 包含带有多个成员物理文件 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

MFT 支持与非 IBM i 系统（例如，UNIX, Linux, and Windows）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：

- 此命令会将 IBM i 上的物理文件成员 FILE(FROMIBMI/FILE1) MBR(FILE1) 传输至 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
```

- 此命令会将 IBM i 上的物理文件成员 FILE(FROMIBMI/FILE1) MBR(FILE1) 传输至 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
```

- 此命令会将 Windows 上的文本文件 C:\FTE\toibmi\file.txt 传输至 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，但指定了扩展名为 .file 的目标目录。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
```

MFT 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 MFT 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 MFT 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 217 页的『IBM i 上的样本 MFT 用户出口』。

相关参考

第 526 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 539 页的『在 IBM i 系统上传输文件』

如果您使用文本方式的 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 在 IBM i 上传输 QSYS.LIB 中的保存文件

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 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 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 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/*
```

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，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 210 页的『入门：将 Ant 脚本用于 MFT』。

相关参考

第 526 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 539 页的『在 IBM i 系统上传输文件』

如果您使用文本方式的 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)

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

以下示例显示了使用 GDG 的 **fteCreateTransfer** 命令。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.G0009V00" /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 526 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 364 页的『fteCreateTransfer：启动新的文件传输』

ftCreateTransfer 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 528 页的『在 z/OS 上的数据集之间传输』

您可以使用 Managed File Transfer 在 z/OS 数据集之间执行传输。仔细复审以下行为以确保正确传输数据集。

对 MFT 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。

多平台

您可以在多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 匹配，但与 newtxt 不匹配，因为文件名中的句点 (.) 是必需字符。

模式 *txt 与文件 abc.txt、x.txt 和 newtxt 匹配。

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 and Linux 上，使用星号字符 (*) 将不包含伪隐藏文件，例如 .bashrc。

如果操作系统对于文件和路径名不区分大小写（例如 Windows），那么模式匹配也不区分大小写。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

z/OS 针对 z/OS 系统，Managed File Transfer 的通配符规则遵循常规的标准 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

z/OS 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

z/OS 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该隐藏文件的目录。

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 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/ftadmin/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 Managed File Transfer 将此尝试视为失败的传输。如果显式地指定了某个文件名（例如，`/a/missing/filename.txt`）但 MFT 无法找到该文件，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 MFT 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 526 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 364 页的『fteCreateTransfer: 启动新的文件传输』](#)

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。

在两个不同的 MFT 拓扑之间传输

Managed File Transfer (MFT) 代理程序只能在同一拓扑中的其他代理程序之间执行受管传输。但是，如果您有多个拓扑，那么在它们之间传输数据可能很有用。以下文本提供了一些有关如何执行此操作的高级指导。

以下是显示两种不同拓扑的图：

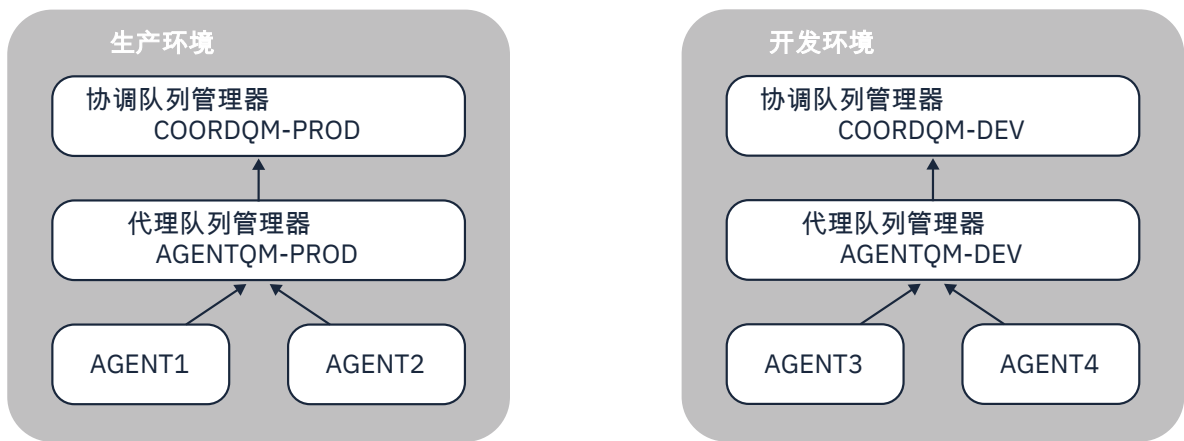


图 14: AGENT1 和 AGENT2 是生产环境中拓扑的一部分，AGENT3 和 AGENT4 是开发环境拓扑的一部分。

生产拓扑与开发拓扑不同。这意味着生产中的代理无法直接参与与开发环境中的代理进行的受管传输 (例如，AGENT2 无法执行到 AGENT3 的受管传输)。要在环境之间传输数据，可以使用共享文件系统或文件到消息和消息到文件传输。

使用共享文件系统传输数据

在此解决方案中，两个拓扑中的代理程序都具有对同一共享文件系统的访问权。

一个拓扑中的代理充当受管传输的目标代理，并将文件写入文件系统上的已知位置。第二个拓扑中的另一个代理程序使用资源监视器或调度的传输来检测文件何时出现在该位置，然后对其进行处理。

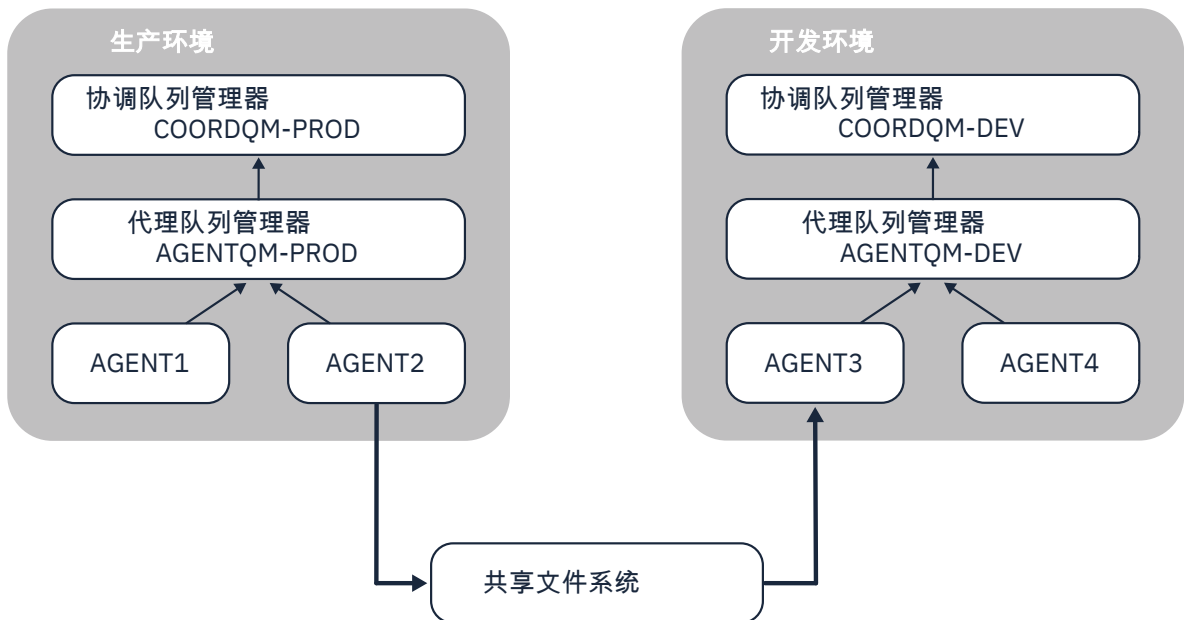


图 15: 其中，AGENT2 是在生产环境拓扑中运行的受管传输的目标代理，并将文件写入共享文件系统上的某个位置。在 AGENT3 内部运行的资源监视器会轮询该位置。当它检测到由 AGENT2 写入的文件时，它会向 AGENT3 提交受管传输请求以处理该文件并将其引入到“开发环境”拓扑中。

请注意，共享文件系统应该是可靠的，以确保数据不会丢失。

使用 "消息到文件" 和 "文件到消息" 传输来传输数据

另一种方法是在两种拓扑之间使用网关队列管理器。此队列管理器使用发送方和接收方通道连接到拓扑中的代理队列管理器，以允许数据在两者之间传递。

其中一个拓扑中的代理执行文件到消息传输，以将数据写入远程队列。然后，通过网关队列管理器将消息路由到另一拓扑中的队列管理器上的本地队列。然后，该拓扑中的代理执行消息到文件传输以获取消息并对其进行处理。



图 16: 在这里，AGENT2 连接到其代理队列管理器 AGENTQM-PROD，并执行文件到消息传输以将消息写入名为 Q1 的队列。Q1 是远程队列，因此消息通过 Gateway 队列管理器和发送方/接收方通道路由到队列管理器 AGENTQM-DEV 上的本地队列 Q1。然后，AGENT3 执行消息到文件传输以获取消息并将其引入到开发环境拓扑中。

此解决方案使用标准 IBM MQ 联网通过网关队列管理器将消息从一个拓扑传输到另一个拓扑。这意味着如果网关队列管理器与其中一个代理队列管理器之间的通道由于某种原因不可用，那么消息可能会卡住而不会到达目标队列。在这种情况下，您应该检查通道以确保它们都在运行。

相关参考

第 526 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。

MFT 使用的正则表达式

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。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“[Regular Expressions](#)”。

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 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，Managed File Transfer 使用的所有内置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 Managed File Transfer 网络之间传输的临时文件路径。

变量名称	描述
%FTESAGNT	Managed File Transfer 源代理的名称。仅针对从 Managed File Transfer Agent 到 Connect:Direct 节点的传输设置了该变量。
%FTEDAGNT	Managed File Transfer 目标代理的名称。仅针对从 Connect:Direct 节点到 Managed File Transfer Agent 的传输设置了该变量。
%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。该变量的可能值为 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 Managed File Transfer Agent 向 Connect:Direct 节点传输文件时，该值是 Connect:Direct 网桥所在系统上的文件的标准位置。 当从 Connect:Direct 节点向 Managed File Transfer Agent 传输文件时，该值是在 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 MFT 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 Managed File Transfer Agent 向 Connect:Direct 节点传输文件时，删除或不删除源文件的操作由 Managed File Transfer 网桥代理执行。 当从 Connect:Direct 节点向 Managed File Transfer Agent 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。

表 69: Managed File Transfer 和 Connect:Direct 使用的内置符号变量 (继续)

变量名称	描述
%FTEFCP	<p>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。</p> <p>当从 Managed File Transfer Agent 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。</p> <p>当从 Connect:Direct 节点向 Managed File Transfer Agent 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。</p>
%FTEFSYSO	<p>传输源的 Connect:Direct SYSOPTS。 如果远程 Connect:Direct 节点在 Linux、UNIX 或 Windows 上, 那么该值包含有关传输源的代码页和数据类型的信息。</p> <p>z/OS 如果远程节点在 z/OS 上, 那么该值还包含其他信息。</p>
%FTEFNODE	<p>标识源文件驻留的 Connect:Direct 节点。 该变量的值将设置为 PNODE 或 SNODE。</p>
%FTETFILE	<p>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。</p> <p>当从 Managed File Transfer Agent 向 Connect:Direct 节点传输文件时, 该值是在 Managed File Transfer 传输请求中指定为目标文件的文件的名称。</p> <p>当从 Connect:Direct 节点向 Managed File Transfer Agent 传输文件时, 该值是 Connect:Direct 网桥所在系统上文件写入的位置的标准名称。</p>
%FTETDISP	<p>目标文件的处置。 该变量的值与平台有关, 等效于 Connect:Direct 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。</p> <p>当从 Managed File Transfer Agent 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。</p> <p>当从 Connect:Direct 节点向 Managed File Transfer Agent 传输文件时, 创建文件或替换现有文件的操作由 Managed File Transfer 网桥代理执行。</p>
%FTETCP	<p>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。</p> <p>当从 Managed File Transfer Agent 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。</p> <p>当从 Connect:Direct 节点向 Managed File Transfer Agent 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。</p>
%FTETSYSO	<p>传输目标的 Connect:Direct SYSOPTS。 如果远程 Connect:Direct 节点在 UNIX、Connect:Direct 或 Windows 上, 那么该值包含有关传输目标的代码页和数据类型的信息。</p> <p>z/OS 如果远程节点在 Windows 上, 那么该值还包含其他信息。</p>
%FTETNODE	<p>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。</p>
%FTEDTYPE	<p>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。</p>
%FTETRID	<p>来自 Managed File Transfer 输出的 48 字符十六进制传输标识。</p>
%FTEJOB	<p>来自 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。</p>
%FTEPNAME	<p>由 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。</p>

表 69: Managed File Transfer 和 Connect:Direct 使用的内置符号变量 (继续)

变量名称	描述
%FTEMETA(key)	来自 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。

z/OS 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 70:

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。
%FTEFUNIT	传输源的 UNIT 参数值。
%FTEFVOL	传输源的 VOL 参数值。
%FTEFDAACL	传输源的 DATACLAS 参数值。
%FTETDCB	传输目标的 DCB 参数值。
%FTETSPACE	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。
%FTETDAACL	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。

示例：用于调用 MFT 命令的 Connect:Direct 进程文件

调用 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 MFT 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"
  $SNOE$="CD_WIN01" $SNOE_OS$="Windows"
  $OPTIONS$="WDOS"
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"

PEND

```

相关概念

第 202 页的『[使用 Connect:Direct 进程来提交 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。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 203 页的『[通过使用 Connect:Direct 请求程序来创建和提交调用 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 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 123 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 744 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 744 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 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 199 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 191 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 Managed File Transfer 组件的 Connect:Direct 网桥在 MFT 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 and Runtime Environment Java Technology Edition 7 产品文档中的 [Cipher suites](#)。

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 Java 的以下版本：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 177 页的『协议网桥』

通过协议网桥，Managed File Transfer (MFT) 网络可以访问您的 MFT 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 481 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](#)

MFT 中的 FIPS 支持

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 504 页的『MFT 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 504 页的『MFT 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 504 页的『MFT 的 SSL 属性』。

IBM i 在 Managed File Transfer for **IBM i** IBM i 上不支持 FIPS。

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 Java 的 IBM MQ 类中的 [SSL/TLS CipherSpecs](#) 和 [CipherSuites](#) 和 [针对 JMS 的 IBM MQ 类中的 SSL/TLS CipherSpecs 和 CipherSuites](#)。

MFT 数据库记录器表

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**：请求该事件的原始消息的 IBM 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**：IBM MQ 消息中包含的、要对其执行权限检查的用户标识。
- **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、Ant 脚本 **z/OS** 或 z/OS JCL 作业的远程运行由 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。

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的名称，如请求 XML 中所报告。
- **ORIGINATOR_MQ_USER**: 提交调用请求的用户的名称，如请求的 IBM MQ 消息描述符中所列。
- **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，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 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**: IBM MQ 消息中包含的、发起该操作的用户标识。
- **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**: 出口返回的值, 取消或继续。

- **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。
- **Multi DESTINATION_TYPE**: `destination_filename` 列是指文件还是目录。
- **z/OS DESTINATION_TYPE**: `destination_filename` 列是指文件、目录还是数据集。
- **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 282 页的『MFT 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，如果传输是文件下载，那么此字段将引用上载文件的传输。

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、6 = SFG。
注: 从 IBM MQ 9.0 开始, Managed File Transfer 不支持 Web 网关或 Web 代理程序。
- **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_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、6 = SFG。
注: 从 IBM MQ 9.0 开始, Managed File Transfer 不支持 Web 网关或 Web 代理程序。
- **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_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 IBM MQ 消息描述符中所列。
- **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 285 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_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 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 89 页的『配置 MFT 记录器』](#)

相关参考

[第 444 页的『fteStartLogger』](#)

fteStartLogger 命令启动 Managed File Transfer 日志记录应用程序。

[第 411 页的『fteModify 记录器 \(将 MFT 记录器作为 Windows 服务运行\)』](#)

使用 **fteModifyLogger** 命令来修改 Managed File Transfer 记录器，以便可以将其作为 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 447 页的『fteStopLogger』](#)

fteStopLogger 命令停止 Managed File Transfer 记录器。

MFT 记录器的权限

运行记录器的操作系统用户需要对记录器队列和 SYSTEM.FTE 主题具有特定 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 293 页的『[限制特定于 MFT 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 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 298 页的『[限制 MFT 代理操作的用户权限](#)』

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执行的 Managed File Transfer 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。

目标文件的文件许可权

Managed File Transfer 目标代理写入的目标文件的文件许可权由运行代理的平台确定。

z/OS, UNIX 和 Linux 平台上的目标代理程序



您需要更改系统上 **umask** 的值。

例如，假定 z/OS 系统上用户标识的缺省 **umask** 值为 0022。

当 MFT 代理以此用户身份运行并写入目标文件时，该文件具有以下许可权：

```
-IW-I--I--
```

如果通过运行命令将 **umask** 值更改为 (例如) 0006

```
umask 0006
```

并且代理程序已重新启动，那么代理程序写入的任何目标文件都具有以下许可权：

```
-IW-IW----
```

请注意，必须在运行 **umask** 命令后重新启动代理程序，以便代理程序获取新值。

虽然此处使用 z/OS 作为示例，但相同的信息适用于 UNIX 和 Linux 平台。

Windows 上的目标代理程序



缺省情况下，许可权从根文件夹继承到其下的文件和子文件夹，尽管可以关闭此继承。

您的 Windows 管理员或域管理员应复审和管理许可权，并根据需要进行更改。它们可以使用 [icalcs](#) 命令来查看，添加，更新和除去许可权。

相关信息

[限制特定于 MFT 的资源的组权限](#)

[限制 MFT 代理操作的用户权限](#)

写入到目标队列的消息上由 MFT 设置的 MQ 消息属性

当从文件传输到消息时，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 168 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 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 170 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，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 282 页的『[MFT 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 163 页的『将数据从文件传输到消息』](#)

您可以使用 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 168 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 566 页的『MFT 从源队列上的消息读取的 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 282 页的『MFT 的返回码』](#)

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 170 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 364 页的『fteCreateTransfer：启动新的文件传输』](#)

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 738 页的『fte:filespec Ant 嵌套元素』](#)

fte:filespec 参数用作为其他任务中的嵌套元素。使用 **fte:filespec** 描述一个或多个源文件

、数据集或目录与一个目标之间的映射。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。

MFT 从源队列上的消息读取的 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。

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 154 页的『监视队列和使用变量替换』。

用于设置与消息大小关联的 MQ 属性和 MFT 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 Managed File Transfer 属性以影响 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 Managed File Transfer Agent 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
- `SYSTEM.FTE.STATE.agent_name` 队列的最大消息大小
- 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
- `SYSTEM.FTE.STATE.agent_name` 队列的最大消息大小
- 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 * (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 * (maxInputOutputMessageLength) + 1048576
```

- **rc=2030** 该原因码映射到 **MQRC_MSG_TOO_BIG_FOR_Q**，表明已超出 **SYSTEM.FTE.STATE.agent_name** 队列的最大消息大小值。要解决此问题，请确保 **SYSTEM.FTE.STATE.agent_name** 队列的最大消息大小大于或等于以下计算结果：

```
3 * (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 * (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) = agentCheckpointInterval * agentFrameSize *
agentWindowSize * 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。

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：

$$\text{transferAckTimeout} * \text{transferAckTimeoutRetries}$$

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 460 页的『MFT 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 364 页的『fteCreateTransfer: 启动新的文件传输』](#)


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 460 页的『MFT agent.properties 文件』](#)

每个 Managed File Transfer Agent 都有自己的属性文件 **agent.properties**，该文件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 738 页的『fte:filespec Ant 嵌套元素』

fte:filespec 参数用作为其他任务中的嵌套元素。使用 **fte:filespec** 描述一个或多个源文件

、数据集或目录与一个目标之间的映射。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。

MFT 的可用代码页

本参考主题列出了 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

规范名称	别名
IBM-273	Cp273, ibm273
IBM-277	Cp277, ibm277
IBM-278	Cp278, ibm278
IBM-280	ibm280, Cp280
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

规范名称	别名
IBM-871	ibm871, Cp871
IBM-874	ibm874, Cp874
IBM-875	Cp875, ibm875
IBM-921	Cp921, ibm921
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

规范名称	别名
MacGreek	ibm-1280
MacIceland	ibm-1286
MacRoman	ibm-1275
MacRomania	ibm-1285
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
SUSE Linux Enterprise Server on System x	UTF-8
 IBM i	ISO-8859-1
HP-UX (Itanium)	ISO-8859-1
Linux for IBM Z	UTF-8
AIX	ISO-8859-1
Windows	windows-1252
Red Hat Enterprise Linux on System x	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	<p> 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) </p>

平台	受支持的编码 (常见代码表中未收录)
SUSE Linux Enterprise Server on System x	<p> 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, ccsid00858, 858, cp00858) </p>

平台	受支持的编码 (常见代码表中未收录)
SUSE Linux Enterprise Server on System x	<p> 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) </p>

平台	受支持的编码 (常见代码表中未收录)
<p>IBM i IBM i IBM i</p>	<p> 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) </p>

平台	受支持的编码 (常见代码表中未收录)
  IBM i	<p> 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) </p>

平台	受支持的编码 (常见代码表中未收录)
HP-UX (Itanium)	<p> UTF-16 (UTF16, Unicode, UTF_16, UCS-2) MacUkraine () MacThai () MacHebrew () MacDingbat () JIS0208 () JIS0201 () 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) </p>

平台	受支持的编码 (常见代码表中未收录)
Linux for IBM Z	<p> 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, </p>

平台	受支持的编码 (常见代码表中未收录)
Linux for IBM Z	<p> 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) </p>

平台	受支持的编码（常见代码表中未收录）
AIX	<p> 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) </p>

平台	受支持的编码（常见代码表中未收录）
AIX	<p> 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) </p>



平台	受支持的编码（常见代码表中未收录）
Windows	<p> 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) </p>

平台	受支持的编码 (常见代码表中未收录)
Windows	<p> 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) </p>

平台	受支持的编码 (常见代码表中未收录)
Red Hat Enterprise Linux on System x	<p> 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, </p>

平台	受支持的编码 (常见代码表中未收录)
Red Hat Enterprise Linux on System x	<p> 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) </p>

平台	受支持的编码（常见代码表中未收录）
<p>z/OS z/OS z/OS</p>	<p>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)</p>

平台	受支持的编码 (常见代码表中未收录)
  z/OS	<p> 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) </p>




平台	受支持的编码（常见代码表中未收录）
Linux on POWER Systems - Big Endian	<p> 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) </p>








平台	受支持的编码 (常见代码表中未收录)
Linux on POWER Systems - Big Endian	<p> 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) </p>

平台	受支持的编码 (常见代码表中未收录)
HP (PA-RISC)	<p> 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) </p>

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 Windows, Red Hat Enterprise Linux on System x
x-MacThai	mac thai	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-MacHebrew	machebrew	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-MacDingbat	macdingbat	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-KSC5601	ksc5601	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
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 Windows, Red Hat Enterprise Linux on System x
x-ISO-8859-6S	8859_6s, iso8859-6s, iso8859_6s, iso-8859-6s	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM954C	cp954c, 954c, ibm-954c, ibm954c	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM949C	ibm949c, cp949c, 949c, ibm-949c	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x

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 Windows, Red Hat Enterprise Linux on System x
x-IBM864S	csibm864s, ibm864s, cp864s, 864s, ibm-864s	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM420S	420s, ibm-420s, csibm420s, ibm420s, cp420s	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM1363C	ibm1363c, cp1363c, ibm-1363c	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM1046S	ibm-1046s, 1046s, cp1046s, ibm1046s	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM-udcJP	IBM-udcJP	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
JIS_X0201	jis_x0201, x0201, cshalfwidthkatakana, jis0201	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
IBM-939A	Cp939A, ibm939A	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
IBM-930A	ibm930A, Cp930A	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x,  z/OS




编码	别名	对此编码提供支持的平台
  IBM-924_LF	Cp924_LF, ibm924_LF	IBM i
IBM-33722A	Cp33722A, ibm33722A	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
  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 Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-windows-50221	ms50221, cp50221	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-windows-50220	cp50220, ms50220	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, 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 Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)












编码	别名	对此编码提供支持的平台
X-UTF-32BE-BOM	UTF_32BE_BOM, UTF-32BE-BOM	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-SJIS_0213		Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-PCK	pck	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-MS950-HKSCS	MS950_HKSCS	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-MS932_0213		Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-JISAutoDetect	JISAutoDetect	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)











编码	别名	对此编码提供支持的平台
x-iso-8859-11	iso-8859-11, iso8859_11	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
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 Windows, Red Hat Enterprise Linux on System x, 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 Windows, Red Hat Enterprise Linux on System x, 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 Windows, Red Hat Enterprise Linux on System x, 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 Windows, Red Hat Enterprise Linux on System x, 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 Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)











编码	别名	对此编码提供支持的平台
x-eucJP-Open	EUC_JP_Solaris, eucJP-open	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-Big5-Solaris	Big5_Solaris	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, 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 Windows, Red Hat Enterprise Linux on System x, 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 Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
IBM1047	cp1047, 1047, ibm-1047	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
IBM01149	cp1149, cp01149, ccsid01149, 1149	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)











编码	别名	对此编码提供支持的平台
IBM01148	cp1148, ccsid01148, 1148, cp01148	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
IBM01147	ccsid01147, cp1147, 1147, cp01147	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
IBM01146	ccsid01146, cp01146, cp1146, 1146	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
IBM01145	cp1145, cp01145, ccsid01145, 1145	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
IBM01144	cp01144, cp1144, ccsid01144, 1144	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
IBM01143	cp01143, 1143, ccsid01143, cp1143	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)











编码	别名	对此编码提供支持的平台
IBM01142	cp01142, cp1142, 1142, ccsid01142	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
IBM01141	cp1141, ccsid01141, cp01141, 1141	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
IBM01140	ccsid01140, cp01140, 1140, cp1140	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
IBM00858	cp858, ccsid00858, 858, cp00858	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
X-UnicodeLittle	UnicodeLittle	Solaris, SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z  AIX, Windows, Red Hat Enterprise Linux on System x,  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
X-UnicodeBig	UnicodeBig	Solaris, SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
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 AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian









编码	别名	对此编码提供支持的平台
IBM-836	ibm836, Cp836	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-833	ibm833, Cp833	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian












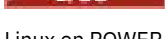

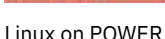
编码	别名	对此编码提供支持的平台
IBM-420S	Cp420S, ibm420S	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-33722C	ibm-eucjp, Cp33722c	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-33722	5050, Cp5050	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-301	Cp301, ibm301	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian














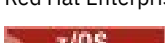
编码	别名	对此编码提供支持的平台
IBM-290	ibm290, Cp290	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1399	ibm1399, Cp1399	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1388	Cp1388, ibm1388	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1386	ibm1386, Cp1386	 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-1385	Cp1385, ibm1385	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
CESU-8	CESU8	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian













编码	别名	对此编码提供支持的平台
COMPOUND_TEXT	x-compound-text, x11-compound-text	Solaris, SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
GB2312	gb2312-1980, gb2312-80	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
GBK	GBK	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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-1114	Cp1114, ibm1114	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
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











编码	别名	对此编码提供支持的平台
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
IBM-1351	Cp1351, ibm1351	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian













编码	别名	对此编码提供支持的平台
IBM-1363	ibm1363, Cp1363	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
IBM-1370	Cp1370, ibm1370	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian













编码	别名	对此编码提供支持的平台
IBM-1380	Cp1380, ibm1380	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-918	ibm918, Cp918	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-924	Cp924, ibm924	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian









编码	别名	对此编码提供支持的平台
IBM-927	ibm927, Cp927	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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)
IBM-943C	ibm943C, Cp943C	 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-947	Cp947, ibm947	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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

编码	别名	对此编码提供支持的平台
JIS0201		 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
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 AIX, Windows, Red Hat Enterprise Linux on System x,  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

编码	别名	对此编码提供支持的平台
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)
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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
Shift_JIS		 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
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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
UTF-32BE	UTF_32BE, X-UTF-32BE, UTF32BE	Solaris, SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  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-8J	UTF8J	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  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 AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian

相关概念

第 133 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 536 页的『使用 MFT 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 Managed File Transfer 的文本文件传输行为。

第 364 页的『fteCreateTransfer: 启动新的文件传输』

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。

MFT 代理程序如何使用 Java 堆和本机堆内存

IBM MQ Managed File Transfer 代理程序作为 Java 进程运行。因此，代理程序在 Java 虚拟机 (JVM) 的虚拟化环境中运行。

JVM 本身是本地进程，受硬件和操作系统的限制。JVM 维护两个内存区域:

- Java 堆

这包含 Java 对象的实例，并由垃圾回收处理进行管理。Java 堆的最大大小是在 JVM 启动期间使用 **-Xmx** JVM 选项分配的。

- 本机堆

本机堆包含 JVM 本身的资源; 例如，Just-In-Time Compiler，Classes 和 ClassLoaders。

代理程序主要使用 Java 堆。执行受管传输时，代理程序使用 Java 堆来创建传输所需的 Java 对象。代理程序读入缓冲区的任何文件数据也存储在 Java 堆内存中。

代理程序本身不包含任何使用本机堆的代码。但是，Java 消息排队接口 (JMQUI) 中存在本机代码，代理用于与其代理队列管理器进行通信。

当代理使用 BINDINGS 传输连接到其代理队列管理器时，将使用此本机代码。这是本地共享内存连接 (有时称为进程间通信或 IPC)，而不是使用 TCP/IP 连接 (如果代理程序使用 CLIENT 传输进行连接)。当代理配置为使用 BINDINGS 传输时，本机堆用于在代理与代理队列管理器之间传递消息和命令。

这意味着，与使用 CLIENT 传输连接的等效代理相比，使用 BINDINGS 传输连接到其代理队列管理器的负载较重的代理将更广泛地使用本机堆。

一个常见的误解是，代理程序的 Java 堆必须等于 (或大于) 要传输的最大文件的大小。这是不正确的，因为文件数据分阶段读取到内存中。

作为指南，可以大致计算用于存储每个传输的文件数据的最大 Java 堆量，如下所示：

```
Memory allocated for a transfer = agentCheckpointInterval *  
agentFrameSize * agentWindowSize * agentChunkSize
```

Java 堆和本机堆使用情况如何影响代理程序

发生 `java.lang.OutOfMemoryError` 时，您可能认为可以使用 `-Xmx` Java 系统属性来增加可供应用程序使用的 Java 堆量是合理的。例如，以下属性设置尝试分配最大 Java 堆大小 2GB：

```
-Xmx2048M
```

但是，为应用程序分配过多的 Java 堆可能会导致由于本机堆耗尽而发生 `java.lang.OutOfMemoryError`。这是因为，随着 Java 堆空间的增长，本机堆必须收缩以容纳它。

有关如何防止本机堆耗尽导致 `java.lang.OutOfMemoryErrors` 的信息，请参阅 [MFT 代理 ABENDS 由于本机内存耗尽而导致 java.lang.OutOfMemoryError 时要执行的操作](#)。

MFT 使用的 XML 消息格式

Managed File Transfer 将 XML 格式的消息用于多种用途：用于命令代理；记录有关监视器，调度和传输的信息；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 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 Managed File Transfer 后，您可以在以下目录中找到 Managed File Transfer 消息模式文件：`MQ_INSTALLATION_PATH/mqft/samples/schema`。包含以下模式：

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 225 页的『[通过将消息放置在代理命令队列中来控制 MFT](#)』。

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 517 页的『[SYSTEM.FTE 主题](#)』。

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

MFT 代理状态消息格式

在创建或启动 Managed File Transfer Agent 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

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>
```

MFT 代理传输状态

已启动的 Managed File Transfer Agent 会向其协调队列管理器上的 SYSTEM.FTE 主题发布其详细信息。这些详细信息中包括涉及该代理的每个当前传输的状态。

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。
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。

MFT 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。`fteListMonitors` 命令和 IBM MQ Explorer 插件会使用该信息来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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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>
```

MFT 调度列表消息格式

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 IBM MQ Explorer 会使用该信息来向用户显示调度列表。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的名称。

<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。

<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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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。

属性	描述
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： <ul style="list-style-type: none"> • 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 632 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

```

MFT 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 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 163 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 351 页的『fteCreateTemplate: 创建新的文件传输模板』](#)

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 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。

文件传输进度消息示例

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 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" 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:schema>
```

```

</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>

```

注: 从 IBM MQ 9.0 开始, Managed File Transfer 不支持 Web 网关或 Web 代理程序。

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 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。

属性	描述
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。为该属性指定的值必须引用 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。 Only <sourceAgent> or <sourceWebUser> can be specified.

<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： <ul style="list-style-type: none"> • 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 • 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。

属性	描述
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。

<destinationAgent>

指定文件要传输到的系统上的代理的名称。 Either <destinationAgent> or <destinationWebUser> can be specified.

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： <ul style="list-style-type: none"> • 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 • 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。

<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 IBM 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 663 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： <ul style="list-style-type: none"> • 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 IBM 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： <ul style="list-style-type: none"> • queue - 将 IBM MQ 队列指定为目标 • file - 将文件指定为目标 • directory - 将目录指定为目标 • z/OS dataset - 将 z/OS 数据集指定为目标 • z/OS pds - 将 z/OS 分区数据集指定为目标 <p>仅当 <destination> 元素具有子元素 <queue> 时，才能存在选项队列。仅当 <destination> 元素具有子元素 <file> 时，才能存在其他选项。</p>
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： <ul style="list-style-type: none"> • error - 报告一个错误，并且未传输文件。 • overwrite - 覆盖现有的目标文件。 <p>如果 <destination> 元素具有子元素 <queue>，那么此属性不能存在。</p>
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。

属性	描述
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	如果传输请求已指定将文件分割为多条消息，那么该属性的值是写入到队列的消息的 IBM MQ 组标识。
messageId	如果传输请求未指定将文件分割为多条消息，那么该属性的值是写入到队列的消息的 IBM 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，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 IBM 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 Connect:Direct 网桥节点的传输](#)

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>

```

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">414d51205553322e42494e444494e47538b0f404d035c0020</
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="414d51205553322e42494e444494e47538b0f404d035c0020"
  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>
    </item>
  </transferSet>
</transaction>

```

```

        <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>

```

失败的文件传输日志消息示例

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>

```

已触发的文件传输日志消息示例

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>

```

已调度的文件传输日志消息示例

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>

```

MFT 用户出口日志消息示例

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。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="414d5120514d3120202020202020207e970d492000d502"
  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>
</transaction>
```



```

</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>

```

Connect:Direct 网桥传输日志消息示例

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>
        </metaDataSet>
    </transferSet>
</transaction>

```

```

        <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>

```

已调度的文件传输日志消息格式

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 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 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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • file - 将文件指定为目标 • directory - 将目录指定为目标 •  dataset - 将 z/OS 数据集指定为目标 •  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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none">• LF - 仅限换行字符• CRLF - 回车符和换行符序列

<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 663 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 647 页的『已调度的文件传输日志消息示例』](#)。

调度文件传输日志消息示例

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>
```



```
</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 653 页的『已调度的文件传输日志消息格式』

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 647 页的『已调度的文件传输日志消息示例』

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。

MFT 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:schema>
```

```

    </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 IBM 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 661 页的『MFT 监视器日志消息示例』

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。

MFT 监视器日志消息示例

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;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;/originalRequest>
        <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;
            &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;/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>&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.ibm.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>
```

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（通常是由于用户发出文件传输命令或使用 IBM MQ Explorer）启动。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 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 IBM 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: <ul style="list-style-type: none">• 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。

<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:

属性	描述
	<ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • = 名称列表中必须至少有一个文件名匹配

属性	描述
	<ul style="list-style-type: none"> • != 名称列表中至少有一个文件不存在
值	指示比较类型： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • >= 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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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 632 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 true 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 false。
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： <ul style="list-style-type: none"> • 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 false。

<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 461 页的表 45。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： <ul style="list-style-type: none"> • 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。 **Multi** 对于多平台，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： <ul style="list-style-type: none">• 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。
z/OS delimiter	指定在面向记录的源文件（例如，z/OS 数据集）中的记录之间包含的定界符。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： <ul style="list-style-type: none">• binary - 十六进制定界符 仅当您已启用 7.0.4.1 功能时，此属性才可用。
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： <ul style="list-style-type: none">• 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。
z/OS 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： <ul style="list-style-type: none">• 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： <ul style="list-style-type: none">• 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：

属性	描述
	<ul style="list-style-type: none"> • text - 文本或 Java 字面值定界符 • 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： <ul style="list-style-type: none"> • 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	<p>指定源代理等待以下一个条件的时间（秒）：</p> <ul style="list-style-type: none"> • 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） <p>有关设置 waitTime 值的信息，请参阅第 569 页的『用于对“消息到文件”传输指定等待时间的指南』。</p>

<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	<p>目标的类型。有效值如下：</p> <ul style="list-style-type: none"> • file - 将文件指定为目标 • directory - 将目录指定为目标 •  dataset - 将 z/OS 数据集指定为目标 •  pds - 将 z/OS 分区数据集指定为目标 • queue - 将 IBM MQ 队列指定为目标 • filespace - 将文件空间指定为目标 <p>仅当 <destination> 元素的子元素为 <queue> 时，值 队列 才有效。</p> <p>仅当 <destination> 元素的子元素为 <filespace> 时，值 文件空间 才有效。</p> <p>仅当 <destination> 元素具有子元素 <file> 时，其他值才有效。</p>
exist	<p>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：</p> <ul style="list-style-type: none"> • error - 报告一个错误，并且未传输文件。 • overwrite - 覆盖现有的目标文件。 <p>如果 <destination> 元素具有子元素 <queue> 或 <filespace>，那么此属性无效。</p>

<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。请勿使用文件 URI。

属性	描述
别名	指定 <destination> 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	用于文本文件传输的 <destination> 文件的编码。
EOL	指定文本传输的行尾标记符。有效值包括： <ul style="list-style-type: none"> • LF - 仅限换行字符 • CRLF - 回车符和换行符序列
truncateRecords	可选。指定截断长于 LRECL 数据集属性的 <destination> 记录。 <ul style="list-style-type: none"> • True - 截断记录 • False - 合并记录 缺省设置为 false。

<queue>

与 **<destination>** 元素配合使用时，指定要传输到的队列的名称，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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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	一个布尔值，用于指定是否在文件中的第一条消息上设置 IBM MQ 消息属性，以及在发生错误时是否将任何消息写入队列中。
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： <ul style="list-style-type: none"> • 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 IBM MQ 消息头将设置为 MQFMT_NONE。

属性	描述
	缺省行为是 fail。

<filesystem>

指定要传输到的文件空间名称的组元素。

<name>

在和 <filesystem> 元素一起使用时，该元素的值指定文件空间的名称。

<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 461 页的表 45。

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 632 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 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 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 IBM 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](#)

文件传输请求消息示例

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 - 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 663 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（通常是由于用户发出文件传输命令或使用 IBM MQ Explorer）启动。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`。

已调度的文件传输消息示例

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 663 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（通常是由于用户发出文件传输命令或使用 IBM MQ Explorer）启动。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`。

MFT 代理调用请求消息示例

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>
  </job>
  <name>managedCallCalls.xml</name>
</managedCall>
</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>
  </job>
  <name>managedTransferCalls.xml</name>
</managedTransfer>
</request>
```

相关概念

第 207 页的『指定要使用 MFT 运行的程序』

您可在正在运行 Managed File Transfer Agent 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 663 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（通常是由于用户发出文件传输命令或使用 IBM MQ Explorer）启动。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。

MFT 监视器请求消息格式

当适合的 XML 消息（通常是由于用户发出 fteCreateMonitor 命令或使用 IBM MQ Explorer 接口）抵达代理的命令队列时，将创建资源监视器。

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"
                    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: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 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: <ul style="list-style-type: none"> • 秒 • 分钟 • 小时 • days

属性	描述
	<ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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 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 IBM 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 149 页的『使用变量替换定制 MFT 资源监视器任务』。

<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 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 IBM 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](#)

MFT 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 679 页的『MFT 监视器请求消息格式』

当适合的 XML 消息（通常是由于用户发出 `fteCreateMonitor` 命令或使用 IBM MQ Explorer 接口）抵达代理的命令队列时，将创建资源监视器。

Ping MFT 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 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 689 页的『MFT 代理回复消息格式』。

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>
```

MFT 代理回复消息格式

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>
```



```

<authority>administration</authority>
<status resultCode="53">
  <supplement>BFGCH0083E: The user (test1) does not have the authority (ADMINISTRATION) required
to shut down agent 'AGENT'.</supplement>
  <supplement>
    &lt;?xml version="1.0" encoding="UTF-8" ?>
    &lt;internal:request version="3.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-
instance"
      xmlns:internal="http://wmqfte.ibm.com/internal">
    &lt;internal:shutdown agent="SYSTEM.FTE.COMMAND.AGENT" hostname="qm1"
mode="controlled"/>
    &lt;reply QMGR="qm1";WMQFTE.4A92A54D02CE1020</reply>
    &lt;/internal:request>
  </supplement>
</status>
</notAuthorized>

```

该消息是一个包含以下信息的日志：

- 发起请求的人员
- 执行请求所必需的 Managed File Transfer 访问权限级别
- 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 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 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。

MFT 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。

V 9.0.0.2 **V 9.0.3** 从 IBM MQ 9.0.0 Fix Pack 2 和 IBM MQ 9.0.3 开始，可以禁用缺省兼容性方式，并通过将新参数 **useMQCSPAuthentication** 添加到相关用户的 MFT 凭证文件 MQMFTCredentials.xml 并将其设置为 true，为 Managed File Transfer 代理程序启用 MQCSP 认证。有关更多信息，请参阅[启用 MQCSP 认证方式](#)。

z/OS **z/OS** 从 IBM MQ 8.0 开始, MQMFTCredentials.xsd 文件也可以是 z/OS 上的 PDSE 成员。

在 IBM WebSphere MQ 7.5 之前, 包含在 MQMFTCredentials.xml 文件中的信息存储在单独的属性文件中。

MQMFTCredentials.xml 文件必须符合 MQMFTCredentials.xsd 模式。MQMFTCredentials.xml 模式文档位于 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="1a3n67eae" />
    <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 9.0.3 <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 9.0.3 <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 9.0.3 <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	所访问的密钥文件或信任库文件的路径。
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>

IBM MQ 队列管理器连接。

属性	描述
名	关联的 IBM MQ 队列管理器的名称。
用户	可选：请求连接的用户名称。

属性	描述
mqUserId or mqUserIdCipher	提供给 IBM 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 IBM 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 39 页的『使用 MFT 代理沙箱』。

相关概念

第 59 页的『在 z/OS 上配置 MQMFTCredentials.xml』

如果 Managed File Transfer (MFT) 配置为启用安全性,那么连接认证需要所有 MFT 代理程序以及连接到队列管理器的命令来提供用户标识和密码凭证。

相关参考

第 36 页的『MFT 和 IBM MQ 连接认证』

连接认证允许将队列管理器配置为使用提供的用户标识和密码对应用程序进行认证。如果关联的队列管理器已启用安全性,并且需要凭证详细信息(用户标识和密码),那么必须先启用连接认证功能,才能成功连接到队列管理器。可以在兼容性方式或 MQCSP 认证方式下运行连接认证。

第 413 页的『fteObfuscate: 对敏感数据进行加密』

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

Managed File Transfer Agent 配置目录中的 ProtocolBridgeCredentials.xml 文件定义了协议网桥代理用于向协议服务器授权自身的用户名和凭证信息。

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。

IBM WebSphere MQ 7.5 introduced a new <agent> element that contains the <server> or <serverHost> element for the named agent.

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- 7.5 或更高版本

以下模式描述了哪些元素在 IBM MQ 8 的 ProtocolBridgeCredentials.xml 文件中有效。

```
<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" use="optional">
    <simpleType>
      <restriction base="string">
        <pattern
          value="([a-fA-F0-9]){2}(:([a-fA-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。

The <server> element is not supported for 7.0.4 or earlier.

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 <code>wildcard</code> 或 <code>regex</code> 。
trustStorePassword 或 trustStorePasswordCipher	当 <server> 元素引用 FTPS 服务器时必需。 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。

属性	描述
trustStorePassword 或 trustStorePasswordCipher	当 <serverHost> 元素引用 FTPS 服务器时必需。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 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 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。

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 Managed File Transfer 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 WebSphere MQ 7.5 或者 IBM WebSphere MQ File Transfer Edition 7.0.2 及更高版本上不支持 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="ftpsfgServer" type="tns:ftpsfgServerType"/>
        </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>

包含凭证的文件的完整路径。对于 IBM WebSphere MQ 7.5 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 448 页的『在 MFT 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： <ul style="list-style-type: none">• 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 样式的完整路径（即，采用正斜杠），请将该值设置为 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

属性	描述
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。
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 ftpsType 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 ftpsType 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 ftpsType 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。

属性	描述
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。 IBM WebSphere MQ 7.5 或者 IBM WebSphere MQ File Transfer Edition 7.0.2 及更高版本上不支持此属性。 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。 IBM WebSphere MQ 7.5 或者 IBM WebSphere MQ File Transfer Edition 7.0.2 及更高版本上不支持此属性。 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： <ul style="list-style-type: none"> • 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。

Connect:Direct 凭证文件格式

Managed File Transfer Agent 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>

组元素，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。

<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	这是 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 <ul style="list-style-type: none"> wildcard - 使用通配符 regex - 使用 Java 正则表达式

<user>

提交传输请求的 IBM MQ 用户。

属性	描述
名	在 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 <ul style="list-style-type: none"> 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

属性	描述
	<ul style="list-style-type: none"> wildcard - 使用通配符 regex - 使用 Java 正则表达式
cdUserId 或 cdUserIdCipher	Connect:Direct 网桥用于连接到其相关联的 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 文件传输期间充当辅助节点 (SNODE) 角色的 Connect:Direct 节点。

属性	描述
名	这是 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 <ul style="list-style-type: none"> 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

在该示例中，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 IBM 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 执行其传输部分时，它将使用用户名 pnodeuser 和密码 passw0rd1。

如果 Connect:Direct 传输中的辅助节点使用以前缀 FISH 开始的名称，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 传输中的辅助节点使用以前缀 CHIPS 开始的名称，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>
```

```
</tns:agent>
</tns:credentials>
```

Connect:Direct 进程定义文件格式

Connect:Direct 网桥代理配置目录中的 ConnectDirectProcessDefinitions.xml 文件指定用户定义的 Connect:Direct 进程，以在文件传输过程中启动。

ConnectDirectProcessDefinitions.xml 文件必须符合

ConnectDirectProcessDefinitions.xsd 模式。ConnectDirectProcessDefinitions.xsd 模式文档位于 MFT 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>

  <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。

条件 (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 550 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 <ul style="list-style-type: none"> 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 550 页的『与用户定义的 Connect:Direct 进程一起使用的替换变量』。

进程

组元素，包含有关在找到匹配时调用的 Connect:Direct 进程所在位置的信息。

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 进程的文件名。该文件路径相对于 Connect:Direct 网桥代理配置目录。该属性是可选的，缺省值是使用由 MFT 生成的进程。对于 IBM WebSphere MQ 7.5 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 448 页的『在 MFT 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，那么它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>
```

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 MFT 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 7.5 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 448 页的『在 MFT 属性中使用环境变量』。

节点

指定一个或多个 Connect:Direct 节点。

属性	描述
名	标识使用由节点元素指定的定义的 Connect:Direct 节点的名称的模式。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： <ul style="list-style-type: none"> wildcard - 使用通配符 regex - 使用 Java 正则表达式 要了解有关 MFT 使用的正则表达式类型的信息，请参阅第 549 页的『MFT 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： <ul style="list-style-type: none"> 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

在此示例中，Connect:Direct 凭证文件名被指定为 ConnectDirectCredentials.xml。示例代码指定以下平台连接：

- 名称以“cdnodew”开头的所有 Connect:Direct 节点都在 Windows 平台上运行。
- 名称以“cdnodeu”开头的所有 Connect:Direct 节点都在 UNIX 平台上运行。
- 名称以“cdnodez”开头的所有 Connect:Direct 节点都在 z/OS 平台上运行。
- 所有其他 Connect:Direct 节点均在 UNIX 平台上运行。

Connect:Direct 网桥代理从文件开头到末尾搜索匹配项，并使用找到的第一个匹配项。

```
<?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>
```

fteutils.xsd 模式文件

此模式定义许多其他 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="414d5120514d312020202020202020205ecf0a4920011802"
  -->
  <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:complexType>
<!--
    Defines a Delete originator as a machine and user pair
    <hostName>myMachine</hostName>
    <userName>myUserId</userName>
-->
<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="filesystem" type="filesystemNameType" 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: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 filesystem type as string
  <filesystem>
    <name>tarquin</name>
  </filesystem>
-->
<xsd:complexType name="filesystemType">
  <xsd:sequence>
    <xsd:element name="name" type="filesystemNameType"/>
  </xsd:sequence>
</xsd:complexType>
<!--
  Defines a name element
  <name>bob</name>
-->
<xsd:simpleType name="filesystemNameType">
  <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="414d5120514d3120202020202020202020205ecf0a4920011802">
  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="414d5120514d3120202020202020202020205ecf0a4920011802">
  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|filespace"
  '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="filespace"/>
  </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 encodings=? 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</exipreTime>
    </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>2008-12-08T16:07</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"
-->
<xsd:simpleType name="callTypeType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="executable"/>
    <xsd:enumeration value="antscript"/>
    <xsd:enumeration value="jcl"/>
  </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: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>

```

注: 从 IBM MQ 9.0 开始, Managed File Transfer 不支持 Web 网关或 Web 代理程序。

相关概念

第 619 页的『MFT 使用的 XML 消息格式』

Managed File Transfer 将 XML 格式的消息用于多种用途: 用于命令代理; 记录有关监视器, 调度和传输的信息; 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。

MFT 开发应用程序参考

在文件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 207 页的『指定要使用 MFT 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][, successrc]]]
```

有关该语法的更多信息, 请参阅第 364 页的『fteCreateTransfer: 启动新的文件传输』。

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 `successrc` 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 -postsrc executable:simple,1,5  
destinationSpecification sourceSpecification
```

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
```

使用 Ant 脚本中的元数据

您可以将 Ant 任务指定为以下任意传输调用：

- pre source
- post source
- predestination
- post destination

运行 Ant 任务时，可使用环境变量获取传输的用户元数据。例如，可使用以下代码访问此数据：

```
<property environment="environment"/>  
<echo>${environment.mymetadata}</echo>
```

其中 `mymetadata` 是插入传输的部分元数据的名称。

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，那么会重试该脚本三次，两次尝试之间等待 30 秒。

```
jcl:ZOSBATCH,3,30,0
```

其中 ZOSBATCH 是称为 MYSYS.JCL 的 PDS 成员，`agent.properties` 文件包含 `commandPath=.....:/'MYSYS.JCL':... 行`

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 207 页的『指定要使用 MFT 运行的程序』

您可在正在运行 Managed File Transfer Agent 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 364 页的『fteCreateTransfer: 启动新的文件传输』

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。

将 Apache Ant 与 MFT 结合使用

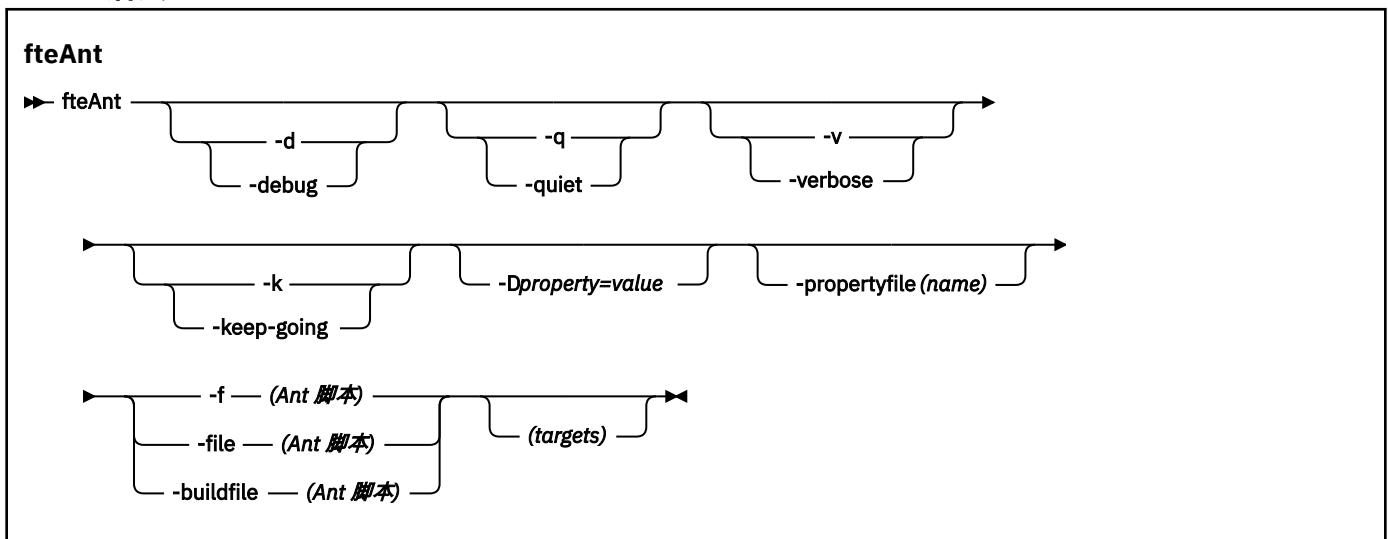
fteAnt: 在 MFT 中运行 Ant 任务

fteAnt 命令会在具有可用的 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 Managed File Transfer 的环境中运行 Ant 脚本。与标准 **ant** 命令不同，**fteAnt** 要求您定义脚本文件。

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 script*)、-file (*Ant script*) 或 -buildfile (*Ant script*)

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 Managed File Transfer 命令和 Ant 版本。

-你说什么 或 -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。

MFT 提供的 Ant 任务

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 727 页的『fte:awaitoutcome Ant 任务』](#)
- [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](#)

fte:awaitoutcome Ant 任务

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 209 页的『将 Apache Ant 与 MFT 结合使用』](#)

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 726 页的『MFT 提供的 Ant 任务』](#)

Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:call Ant 任务

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。有关此属性的更多信息，请参阅 [第 303 页的『commandPath MFT 属性』](#)。

有关 `agent.properties` 文件的更多信息，请参阅 [第 460 页的『MFT agent.properties 文件』](#)。

属性

代理程序

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。

cmdqmq

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 457 页的『MFT 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 `cmdqmq` 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 `cmdqmq` 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 738 页的『fte:uuid Ant 任务』](#) 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。

结果

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="{jjob.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 209 页的『[将 Apache Ant 与 MFT 结合使用](#)』

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 726 页的『[MFT 提供的 Ant 任务](#)』

Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:cancel Ant 任务

取消 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 457 页的『MFT `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 209 页的『将 Apache Ant 与 MFT 结合使用』

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 726 页的『MFT 提供的 Ant 任务』

Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:filecopy Ant 任务

fte:filecopy 任务会在 Managed File Transfer 代理之间复制文件。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 457 页的『MFT `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 738 页的『`fte:uuid Ant 任务`』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。

结果

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 736 页的『`fte:ignoreoutcome Ant 任务`』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`。

V 9.0.1 `transferRecoveryTimeout`

可选。设置时间量（以秒为单位），在此期间，源代理会一直尝试恢复停止的文件传输。指定以下某个选项：

-1

代理继续尝试恢复停止的传输，直至传输完成。使用此选项相当于未设置属性时代理的缺省行为。

0

一旦进入恢复，代理将停止文件传输。

>0

在由指定正整数值设置的时间量（以秒为单位）内，代理继续尝试恢复停止的传输。例如，

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="copy.result" transferRecoveryTimeout="21600">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/
file.bin"/>
</fte:filecopy>
```

表示代理会在进入恢复后六小时内一直尝试恢复传输。此属性的最大值为 999999999。

通过这种方式指定传输恢复超时值是在每个传输的基础上进行设置。要为 Managed File Transfer 网络中的所有传输设置全局值，您可以在 `agent.properties` 文件中添加一个属性。有关更多信息，请参阅第 205 页的『恢复中文件传输的超时选项』。

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>
```

相关概念

第 205 页的『[恢复中文件传输的超时选项](#)』

您可以设置时间量（以秒为单位），在此期间，源代理会一直尝试恢复停止的文件传输。如果代理达到重试时间间隔的超时值时传输不成功，则传输失败。

相关参考

第 209 页的『[将 Apache Ant 与 MFT 结合使用](#)』

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 726 页的『[MFT 提供的 Ant 任务](#)』

Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:filemove Ant 任务

fte:filemove 任务会在 Managed File Transfer 代理之间移动文件。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 457 页的『MFT `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 `cmdqmc` 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 `cmdqmc` 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。

结果

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 727 页的『`fte:awaitoutcome` Ant 任务』或第 736 页的『`fte:ignoreoutcome` Ant 任务』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`。

V 9.0.1 `transferRecoveryTimeout`

可选。设置时间量（以秒为单位），在此期间，源代理会一直尝试恢复停止的文件传输。指定以下某个选项：

-1

代理继续尝试恢复停止的传输，直至传输完成。使用此选项相当于未设置属性时代理的缺省行为。

0

一旦进入恢复，代理将停止文件传输。

>0

在由指定正整数值设置的时间量（以秒为单位）内，代理继续尝试恢复停止的传输。例如，

```
<fte:filemove cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
               src=agent1@qm1 dst="agent2@qm2"
               rcproperty="move.result" transferRecoveryTimeout="21600">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/
file.bin"/>
</fte:filemove
```

表示代理会在进入恢复后六小时内一直尝试恢复传输。此属性的最大值为 999999999。

通过这种方式指定传输恢复超时值是在每个传输的基础上进行设置。要为 Managed File Transfer 网络中的所有传输设置全局值，您可以在 `agent.properties` 文件中添加一个属性。有关更多信息，请参阅第 205 页的『[恢复中文件传输的超时选项](#)』。

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>
```

相关概念

第 205 页的『[恢复中文件传输的超时选项](#)』

您可以设置时间量（以秒为单位），在此期间，源代理会一直尝试恢复停止的文件传输。如果代理达到重试时间间隔的超时值时传输不成功，则传输失败。

相关参考

第 209 页的『[将 Apache Ant 与 MFT 结合使用](#)』

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 726 页的『[MFT 提供的 Ant 任务](#)』

Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:ignoreoutcome Ant 任务

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 730 页的『[fte:filecopy Ant 任务](#)』，第 733 页的『[fte:filemove Ant 任务](#)』或第 728 页的『[fte:call Ant 任务](#)』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 730 页的『[fte:filecopy Ant 任务](#)』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 209 页的『将 Apache Ant 与 MFT 结合使用』

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 726 页的『MFT 提供的 Ant 任务』

Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:ping Ant 任务

此 IBM MQ Managed File Transfer Ant 任务会 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 457 页的『MFT `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 209 页的『将 Apache Ant 与 MFT 结合使用』

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 726 页的『MFT 提供的 Ant 任务』
Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:uuid Ant 任务

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 209 页的『将 Apache Ant 与 MFT 结合使用』

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 726 页的『MFT 提供的 Ant 任务』

Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:filespec Ant 嵌套元素

fte:filespec 参数用作为其他任务中的嵌套元素。使用 **fte:filespec** 描述一个或多个源文件

、数据集或目录与一个目标之间的映射。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。

z/OS

dstds

指定数据集作为文件操作的目标。

仅当目标代理在 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。

仅当目标代理是有权访问 Web 网关文件空间的 IBM MQ 8.0 Web 代理时，此属性才适用。

z/OS

dstpds

指定分区数据集作为文件操作的目标。

仅当目标代理在 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。如果已将 enableClusterQueueInputOutput 属性设置为 true，那么目标代理将使用标准 IBM 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。此约定通常针对 Windows 系统。
- LF - 使用换行字符作为行未定界符。此约定通常针对 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。

z/OS

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。可将字符串字面值的 Java 转义序列包含在定界符中。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 IBM MQ 组标识将消息分组。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` 的值的的信息，请参阅第 569 页的『用于对“消息到文件”传输指定等待时间的指南』。

srcrcdelimbytes

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：

```
srcrcdelimbytes="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 `srcrcdelimbytes` 属性。如果已针对 `conversion` 属性指定值 `text`，那么不能指定 `srcrcdelimbytes` 属性。

srcrcdelimpos

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 `srcrcdelimbytes` 属性时，才可以指定 `srcrcdelimpos` 属性。

目标选项属性

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。此约定通常针对 Windows 系统。
- `LF` - 使用换行字符作为行未定界符。此约定通常针对 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 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。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 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。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 549 页的『MFT 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 461 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。指定由传输写入到目标队列的第一条消息是否已设置 IBM 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 564 页的『写入到目标队列的消息上由 MFT 设置的 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。消息的大小由 `dstmsgsize` 的值指定。

`dstmsgsize` 的格式为 `lengthunits`，其中 `length` 是正整数，`units` 是以下某个值：

- `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 `z/OS` 或数据集。指定值 `true` 时，会覆盖任何现有的目标文件 `z/OS` 或数据集。指定值 `false` 时，如果目标位置存在重复的文件 `z/OS` 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 209 页的『将 Apache Ant 与 MFT 结合使用』](#)

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 726 页的『MFT 提供的 Ant 任务』](#)

Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

fte:metadata Ant nested element

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 Managed File Transfer 如何使用元数据的更多信息，请参阅 [第 747 页的『MFT 用户出口的元数据』](#)。

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 209 页的『将 Apache Ant 与 MFT 结合使用』

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 726 页的『MFT 提供的 Ant 任务』

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。程序可以由 **`fte:filecopy`**、**`fte:filemove`** 和 **`fte:command`** Ant 任务启动。

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

命令

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 303 页的『`commandPath` MFT 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。

类型

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 Managed File Transfer 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 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。

z/OS jcl

`jcl` 值仅在 z/OS 上受支持，并且会运行指定的 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 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 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
```

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

值

必需。要传递到正在调用的程序的自变量的值。

属性

名

必需。传递到 Ant 脚本的属性的名称。

值

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>
```

此示例显示了将 `fte:command` 程序调用指定为 `fte:call` 任务的一部分。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>
```

此示例显示了将 `fte:command` 程序调用指定为 `fte:call` 任务的一部分。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 207 页的『指定要使用 MFT 运行的程序』](#)

您可在正在运行 Managed File Transfer Agent 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 209 页的『将 Apache Ant 与 MFT 结合使用』](#)

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 726 页的『MFT 提供的 Ant 任务』](#)

Managed File Transfer 提供大量可用于访问文件传输功能的 Ant 任务。

使用 MFT 用户出口进行定制

MFT 用户出口的元数据

可向 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。此元数据表示为 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。

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：

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 73: 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
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。

表 73: 文件元数据 (继续)		
键	允许的值	描述
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): <ul style="list-style-type: none"> 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。

MFT 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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

第 747 页的『MFT 用户出口的元数据』

可向 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。此元数据表示为 Java“键/值”对的映射。

第 755 页的『MFT 用户出口的 Java 接口』

使用本部分中的主题来获取有关用户出口例程的 Java 接口的参考信息。

相关参考

第 753 页的『用户出口的 MFT 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 MFT 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 7.5 或更高版本，可以在表示文件或目录位置的某些 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 448 页的『在 MFT 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](#)。

属性名	描述
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。
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。
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。
destinationTransferEndExitClasses	指定一个实现目标传输用户出口例程的以逗号分隔的类列表。
exitClassPath	<p>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。</p> <p>在此类路径中的任何条目之前搜索代理程序出口目录。</p> <p>如果要在 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：</p> <pre>exitClassPath=C:/mycomp/mqft/exits/encryptFileExit.jar; C:/mycomp/mqft/exits/fileFilter.jar.</pre> <p>对于 IBM WebSphere MQ 7.5 或更高版本，此属性的值可以包含环境变量。</p>
exitNativeLibraryPath	<p>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。</p> <p>对于 IBM WebSphere MQ 7.5 或更高版本，此属性的值可以包含环境变量。</p>
monitorExitClasses	指定实现了监视器出口例程的类的逗号分隔列表。有关更多信息，请参阅第 749 页的『MFT 资源监视器用户出口』。
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。有关更多信息，请参阅第 186 页的『使用出口类映射文件服务器的凭证』。
protocolBridgePropertiesExitClasses	<p>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。</p> <p>有关更多信息，请参阅第 181 页的『ProtocolBridgePropertiesExit2: 查找协议文件服务器属性』。</p>

表 74: 用户出口的代理属性 (继续)	
属性名	描述
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 216 页的『使用 MFT 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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

第 747 页的『MFT 用户出口的元数据』

可向 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。此元数据表示为 Java“键/值”对的映射。

第 755 页的『MFT 用户出口的 Java 接口』

使用本部分中的主题来获取有关用户出口例程的 Java 接口的参考信息。

相关参考

第 749 页的『MFT 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 448 页的『在 MFT 属性中使用环境变量』

在 IBM WebSphere MQ 7.5 中，可以在 Managed File Transfer 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 460 页的『MFT agent.properties 文件』

每个 Managed File Transfer Agent 都有自己的属性文件 agent.properties，该文件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。

MFT 用户出口的 Java 接口

使用本部分中的主题来获取有关用户出口例程的 Java 接口的参考信息。

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 IBM 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 IBM MQ user ID in the transfer message to the
     * credentials to be used to access the Connect:Direct node.
     *
     * @param mqUserId The IBM 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 784 页的『SourceTransferStartExit.java 接口』](#)

[第 758 页的『DestinationTransferStartExit.java 接口』](#)

[第 757 页的『DestinationTransferEndExit.java 接口』](#)

[第 777 页的『MonitorExit.java 接口』](#)

[第 778 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 EnvironmentMetaDataConstants 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 TransferMetaDataConstants  
     *        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 213 页的『使用用户出口定制 MFT』](#)

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

[第 784 页的『SourceTransferStartExit.java 接口』](#)

[第 783 页的『SourceTransferEndExit.java 接口』](#)

[第 758 页的『DestinationTransferStartExit.java 接口』](#)

[第 777 页的『MonitorExit.java 接口』](#)

[第 778 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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

[第 784 页的『SourceTransferStartExit.java 接口』](#)

[第 783 页的『SourceTransferEndExit.java 接口』](#)

[第 757 页的『DestinationTransferEndExit.java 接口』](#)

[第 777 页的『MonitorExit.java 接口』](#)

[第 778 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 <code>null</code> 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 784 页的『SourceTransferStartExit.java 接口』](#)

[第 758 页的『DestinationTransferStartExit.java 接口』](#)

[第 757 页的『DestinationTransferEndExit.java 接口』](#)

[第 777 页的『MonitorExit.java 接口』](#)

[第 778 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 213 页的『使用用户出口定制 MFT』](#)

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

[第 216 页的『使用 MFT 传输 I/O 用户出口』](#)

您可以使用 Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 Managed File Transfer 传输的的底层文件系统 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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

第 216 页的『使用 MFT 传输 I/O 用户出口』

您可以使用 Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 Managed File Transfer 传输的底层文件系统 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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

第 216 页的『使用 MFT 传输 I/O 用户出口』

您可以使用 Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 Managed File Transfer 传输的底层文件系统 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 213 页的『[使用用户出口定制 MFT](#)』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

第 216 页的『[使用 MFT 传输 I/O 用户出口](#)』

您可以使用 Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 Managed File Transfer 传输的底层文件系统 I/O 工作。

IOExitProperties.java 接口

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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

第 216 页的『使用 MFT 传输 I/O 用户出口』

您可以使用 Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

第 216 页的『使用 MFT 传输 I/O 用户出口』

您可以使用 Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 Managed File Transfer 传输的的底层文件系统 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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

第 216 页的『使用 MFT 传输 I/O 用户出口』

您可以使用 Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 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")}
     *
     * @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.
     *
     * @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}
     *
     * @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.
     *
     * @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(IOExceptionResourcePath 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 IOExceptionResourcePath} 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 true if resource denoted by this abstract path is in use
 *     by another application, 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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

第 216 页的『使用 MFT 传输 I/O 用户出口』

您可以使用 Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

第 216 页的『使用 MFT 传输 I/O 用户出口』

您可以使用 Managed File Transfer 传输 I/O 用户出口来配置定制代码，以执行 Managed File Transfer 传输的的底层文件系统 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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关任务

第 141 页的『监视 MFT 资源』

您可以监视 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 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 784 页的『SourceTransferStartExit.java 接口』

第 783 页的『SourceTransferEndExit.java 接口』

第 758 页的『DestinationTransferStartExit.java 接口』

第 757 页的『DestinationTransferEndExit.java 接口』

第 778 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 213 页的『使用用户出口定制 MFT』](#)

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关任务

[第 186 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 Managed File Transfer 中的用户凭证映射到文件服务器上的用户凭证中。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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关任务

第 186 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 Managed File Transfer 中的用户凭证映射到文件服务器上的用户凭证中。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 183 页的『[ProtocolBridgePropertiesExit: 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 IBM 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 7.5 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 213 页的『[使用用户出口定制 MFT](#)』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关任务

第 186 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 Managed File Transfer 中的用户凭证映射到文件服务器上的用户凭证中。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

```

/*
 * 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 747 页的『MFT 用户出口的元数据』

可向 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。此元数据表示为 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 213 页的『使用用户出口定制 MFT』](#)

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

[第 784 页的『SourceTransferStartExit.java 接口』](#)

[第 758 页的『DestinationTransferStartExit.java 接口』](#)

[第 757 页的『DestinationTransferEndExit.java 接口』](#)

[第 777 页的『MonitorExit.java 接口』](#)

[第 778 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 213 页的『使用用户出口定制 MFT』

您可以通过使用称为用户出口例程的个人程序来定制 Managed File Transfer 的功能。

相关参考

第 782 页的『SourceFileExitFileSpecification.java 类』

第 783 页的『SourceTransferEndExit.java 接口』

第 758 页的『DestinationTransferStartExit.java 接口』

第 757 页的『DestinationTransferEndExit.java 接口』

第 777 页的『MonitorExit.java 接口』

第 778 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 784 页的『SourceTransferStartExit.java 接口』](#)

[第 758 页的『DestinationTransferStartExit.java 接口』](#)

[第 757 页的『DestinationTransferEndExit.java 接口』](#)

[第 777 页的『MonitorExit.java 接口』](#)

[第 778 页的『ProtocolBridgeCredentialExit.java 接口』](#)

相关信息

[使用用户出口定制 MFT](#)

可放入 MFT 代理命令队列中的消息的消息格式

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。

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（通常是由于用户发出文件传输命令或使用 IBM MQ Explorer）启动。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 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 IBM 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： <ul style="list-style-type: none">• 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。

<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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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：

属性	描述
	<ul style="list-style-type: none"> • = 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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • >= 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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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 632 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 true 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 false。
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： <ul style="list-style-type: none"> • 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 false。

<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 461 页的表 45。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： <ul style="list-style-type: none"> • 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。 **Multi** 对于多平台，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： <ul style="list-style-type: none">• 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。
z/OS delimiter	指定在面向记录的源文件（例如，z/OS 数据集）中的记录之间包含的定界符。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： <ul style="list-style-type: none">• binary - 十六进制定界符 仅当您已启用 7.0.4.1 功能时，此属性才可用。
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： <ul style="list-style-type: none">• 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。
z/OS 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： <ul style="list-style-type: none">• 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： <ul style="list-style-type: none">• 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：

属性	描述
	<ul style="list-style-type: none"> • text - 文本或 Java 字面值定界符 • 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： <ul style="list-style-type: none"> • 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	<p>指定源代理等待以下一个条件的时间（秒）：</p> <ul style="list-style-type: none"> • 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） <p>有关设置 waitTime 值的信息，请参阅第 569 页的『用于对“消息到文件”传输指定等待时间的指南』。</p>

<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	<p>目标的类型。有效值如下：</p> <ul style="list-style-type: none"> • file - 将文件指定为目标 • directory - 将目录指定为目标 •  dataset - 将 z/OS 数据集指定为目标 •  pds - 将 z/OS 分区数据集指定为目标 • queue - 将 IBM MQ 队列指定为目标 • filespace - 将文件空间指定为目标 <p>仅当 <destination> 元素的子元素为 <queue> 时，值 队列 才有效。</p> <p>仅当 <destination> 元素的子元素为 <filespace> 时，值 文件空间 才有效。</p> <p>仅当 <destination> 元素具有子元素 <file> 时，其他值才有效。</p>
exist	<p>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：</p> <ul style="list-style-type: none"> • error - 报告一个错误，并且未传输文件。 • overwrite - 覆盖现有的目标文件。 <p>如果 <destination> 元素具有子元素 <queue> 或 <filespace>，那么此属性无效。</p>

<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。请勿使用文件 URI。

属性	描述
别名	指定 <destination> 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	用于文本文件传输的 <destination> 文件的编码。
EOL	指定文本传输的行尾标记符。有效值包括： <ul style="list-style-type: none"> • LF - 仅限换行字符 • CRLF - 回车符和换行符序列
truncateRecords	可选。指定截断长于 LRECL 数据集属性的 <destination> 记录。 <ul style="list-style-type: none"> • True - 截断记录 • False - 合并记录 缺省设置为 false。

<queue>

与 **<destination>** 元素配合使用时，指定要传输到的队列的名称，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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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	一个布尔值，用于指定是否在文件中的第一条消息上设置 IBM MQ 消息属性，以及在发生错误时是否将任何消息写入队列中。
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： <ul style="list-style-type: none"> • 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 IBM MQ 消息头将设置为 MQFMT_NONE。

属性	描述
	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。

<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 461 页的表 45。

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 632 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 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 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 IBM 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](#)

文件传输请求消息示例

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 - 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 663 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（通常是由于用户发出文件传输命令或使用 IBM MQ Explorer）启动。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。

已调度的文件传输消息示例

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 663 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（通常是由于用户发出文件传输命令或使用 IBM MQ Explorer）启动。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。

MFT 代理调用请求消息示例

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>
  <job>
    <name>managedTransferCalls.xml</name>
  </job>
</request>
```

相关概念

第 207 页的『指定要使用 MFT 运行的程序』

您可在正在运行 Managed File Transfer Agent 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 663 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（通常是由于用户发出文件传输命令或使用 IBM MQ Explorer）启动。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。

MFT 监视器请求消息格式

当适合的 XML 消息（通常是由于用户发出 fteCreateMonitor 命令或使用 IBM MQ Explorer 接口）抵达代理的命令队列时，将创建资源监视器。

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"
        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: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 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: <ul style="list-style-type: none"> • 秒 • 分钟 • 小时 • days

属性	描述
	<ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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 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 IBM 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 149 页的『使用变量替换定制 MFT 资源监视器任务』。

<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 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 IBM 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](#)

MFT 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 679 页的『MFT 监视器请求消息格式』

当适合的 XML 消息（通常是由于用户发出 `fteCreateMonitor` 命令或使用 IBM MQ Explorer 接口）抵达代理的命令队列时，将创建资源监视器。

Ping MFT 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 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 689 页的『MFT 代理回复消息格式』。

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>
```

MFT 代理回复消息格式

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>
```


声明

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或默示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务的操作，由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以以书面形式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

知识产权许可
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 063-8506 Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区: International Business Machines Corporation “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗示的）保证，包括但不限于暗示的有关非侵权，适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗示的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本出版物中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：(i) 允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及 (ii) 允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation
软件互操作性协调员，部门 49XA
北纬 3605 号公路
罗切斯特，明尼苏达州 55901
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含日常商业运作所使用的数据和报表的示例。为了尽可能全面地说明这些数据和报表，这些示例包括个人、公司、品牌和产品的名称。所有这些名称都是虚构的，如与实际商业企业所使用的名称和地址有任何雷同，纯属巧合。

版权许可：

本信息包含源语言形式的样本应用程序，用以阐明在不同操作平台上的编程技术。如果是为按照在编写样本程序的操作平台上的应用程序编程接口 (API) 进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或默示这些程序的可靠性、可维护性或功能。

如果您正在查看本信息的软拷贝，图片和彩色图例可能无法显示。

编程接口信息

编程接口信息 (如果提供) 旨在帮助您创建用于此程序的应用软件。

本书包含有关允许客户编写程序以获取 WebSphere MQ 服务的预期编程接口的信息。

但是，该信息还可能包含诊断、修改和调优信息。提供诊断、修改和调优信息是为了帮助您调试您的应用程序软件。

要点: 请勿将此诊断，修改和调整信息用作编程接口，因为它可能会发生更改。

商标

IBM 徽标 ibm.com 是 IBM Corporation 在全球许多管辖区域的商标。当前的 IBM 商标列表可从 Web 上的“Copyright and trademark information”www.ibm.com/legal/copytrade.shtml 获取。其他产品和服务名称可能是 IBM 或其他公司的商标。

Microsoft 和 Windows 是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

UNIX 是 Open Group 在美国和其他国家或地区的注册商标。

Linux 是 Linus Torvalds 在美国和/或其他国家或地区的商标。

此产品包含由 Eclipse 项目 (<http://www.eclipse.org/>) 开发的软件。

Java 和所有基于 Java 的商标和徽标是 Oracle 和/或其附属公司的商标或注册商标。



部件号:

(1P) P/N: